ROBUST GENE SET ANALYSIS
AND ROBUST GENE EXPRESSION

By

NING TANG

A dissertation submitted to the

Graduate School – New Brunswick

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Statistics

written under the direction of

Professor Javier Cabrera

and approved by

_____

_____

_____

_____

New Brunswick, New Jersey

May 2014

ABSTRACT OF THE DISSERTATION

Robust Gene Set Analysis and Robust Gene Expression

by NING TANG

Dissertation Director:

Professor Javier Cabrera

This paper explores various methods of statistical analysis of DNA microarray data. First, we review the RMA method which produces estimates of gene expression from a microarray data and propose a new version of RMA that is not only resistant to outliers but also has high efficiency. To construct our new RMA estimator we rely upon M-estimator of location, including Tukey's biweight and Huber's M-estimator. We compare the performance of our robust version of RMA with median, the currently used one in the RMA method, as well as mean, which is a non-robust estimator of location. Second, we review the Gene Set Enrichment Analysis (GSEA) methodology. Currently, the GSEA method is performed at gene-level. This requires DNA microarray data be transformed from the raw probe-level data to the gene-level data. This process cannot avoid losing subtle but crucial information contained in the probe-level data. Inspired by the GSEA method, we extend its idea to the probe-level data. Finally, we develop a family of enrichment method - Enrichment Analysis using M-estimator (EAME), which, as implied by its name, uses robust M-estimator and take advantage of the idea of gene set enrichment. At the end of this paper, we use the R language as a tool to show some examples of DNA microarray analysis based on the methodologies discussed in this paper.

Acknowledgements

First I would like to express my deepest appreciation to all my committee members for their time and efforts devoted to advising and reviewing my dissertation.

I definitely owe a lot of thanks to my committee chair and dissertation advisor, Professor Javier Cabrera, who has been always willing to offer necessary support, invaluable suggestions and continuing encouragement. Without his abundant help, my goal of research could not be accomplished. I must also say many thanks to Dr. Dhammika Amaratunga, who has been giving generous assistance for my research. I undoubtedly received from him a great deal of advice, which greatly helped me overcome many obstacles in areas that I am not familiar with.

I am greatly indebted to Professor John Kolassa, who has offered me a lot of help both academically and financially. I would like to thank Professor Donald R. Hoover too for his generous support for my last semesters.

My thanks also go to all faculty of the Statistics Department of Rutgers for providing excellent academic training for all students.

At the end, I must express my love and gratitude to my beloved families for their indispensable support throughout the duration of my studies.

Dedication

To my family

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 An Introduction of DNA Microarray Technology

The Deoxyribonucleic Acid (DNA) microarray is undoubtedly a fast-developing technology in modern biomedical research. The traditional "one gene per experiment" approach is now becoming obsolete due to its incapability of producing a complete vision of gene functions and overall genome behaviors in a reasonable amount of time. Unlike the traditional molecular biology research approach, the DNA microarray technology offers biologists a much more efficient way to collect the expression information about thousands of genes at the same time.

Generally speaking, microarrays refer to a lot of distinct platforms that can assay large amounts of biological material using high-throughput screening methods on a solid support. It can produce thousands to tens of thousands of data points in one experiment. There are several types of microarrays, among which tissue microarrays, protein microarrays and DNA microarrays are the three major ones. Tissue microarrays relocate small amount of tissue from biopsies of multiple patients on to glass slides to allow multiplex histological analysis; protein microarrays immobilize peptides or intact proteins to track the interactions and activities of proteins, and to determine their function. For recent years, with the development of gene array detection technologies such as

Affymetrix, Illumina and Codelink, etc., DNA microarrays has caught a great amount of attention and there are a lot of literatures concentrating on DNA microarray analysis.

In this paper, we will focus on methods of statistical analysis of DNA microarrays. We give an introduction of current procedures for preprocessing oligonucleotide microarray in Chapter 2. In Chapter 3, methods of statistical analyses on individual genes are introduced. We then review the RMA method and discuss the application of robust statistics in the RMA method in Chapter 4. The use of robust statistics in RMA makes it possible for us to develop new methodologies that are robust and efficient. Next, Gene-level GSEA method and Probe-level GSEA method are introduced and compared in Chapter 5 and Chapter 6. In Chapter 7, we propose various EAME approaches and compare their performance of detection of differentially expressed gene sets. In Chapter 8, we use the R language as a tool to show some examples of DNA microarray analysis based on the methodologies discussed in this paper. In Chapter 9, we give some conclusions and remarks.

Before we move on to next chapter, an introduction of basic biology of gene expression is given in the following sections.

1.2 Basic Biology of Gene Expression

Genomics is an interesting and complex subject and it can simply occupy a multi-volume book just by itself. However, it is about life after all. The basic unit of life is the cell and a very crucial characteristic of life is the ability of reproduction. Single-celled organisms reproduce by self-duplication, while

multicellular organisms begin with a single cell and develop by the programmed division of cells.

During the process of reproduction, some information is carried from one generation to the next by some sort of a hereditary mechanism. This important feature of that mechanism results in that family members tend to exhibit similar characteristics. The existence of such hereditary units, now called *genes*, has been firmly established.

All living beings depend on genes because they specify all proteins. Proteins are involved in performing and regulate most of the important functions in a cell. A protein can be represented as a linear sequence of amino acids connected by peptide bonds. Each amino acid consists of a central carbon atom to which three chemical entities are joined. They are the amine (NH_2) group, the carboxyl (COOH) group, and the side chain (R). There are 20 commonly occurring types of side chains and thus 20 amino acids.

Proteins do not self-assemble. They are assembled based on information contained in genes, which are made of deoxyribonucleic acid (DNA). The assembly of proteins requires messenger ribonucleic acid (messenger RNA or mRNA) to act as an intermediate. Messenger RNA is synthesized using DNA as a template, and is then used for *translation* into protein. All RNA molecules consist of a sequence of nucleotides and each nucleotide in RNA contains a ribose sugar, a phosphate group and a nitrogenous base. The base is one of adenine (A), cytosine (C), guanine (G) and uracil (U).

A DNA molecule is also a polymer of nucleotides, in many ways similar to RNA. However, there are three major differences between them:

(1) A DNA molecule is a double-stranded helix, consisting of two long polymers of nucleotides, while RNA is a single-stranded molecule in many of its biological roles and has a much shorter chain of nucleotides.

(2) DNA molecules contain a deoxyribose sugar backbone, rather than a ribose sugar backbone as in RNA.

(3) In DNA, the base thymine (T) replaces uracil (U).

The double helix structure of DNA is stabilized by chemical bonds between pairs of complementary bases on the two strands. The adenines (A) only bind with thymines (T) and the guanines (G) only pair with the cytosines (C). The pairs so formed are called *base pairs* (shortly *bp*). This complementarity of bases is a crucial feature of DNA and it is the basis of both cell reproduction and gene expression.

When a gene is being expressed, the two strands of one DNA molecule unwind and one mRNA molecule is then synthesized using a segment of one of the two DNA strands as a template. This process is called *transcription*. The DNA segment used for mRNA synthesis corresponds roughly to a particular single gene. The sequence of bases along the mRNA strand being synthesized is identical to the sequence of bases along the inactive DNA strand, except that mRNA has uracil (U) where DNA has thymine (T).

1.3 Hybridization Assays

If the sequences from two single-stranded DNA molecules are complementary to each other, these two DNA molecules usually tend to bind together to form a single double-stranded DNA molecule. This process is known as *hybridization*. No matter if two DNA strands (or one DNA strand and one mRNA strand) come

from the same source or from two different sources, they will hybridize with each other as long as their base pair sequences match according to the complementary base-pairing rules. Even when the sequences on the two strands do not match exactly, some base pairing will still occur and a hybrid DNA molecule will be formed as long as there is sufficient similarity.

This property of DNA (also mRNA) strands provides biologists a useful tool to observe the current state of a cell at a given period. Because scientists are interested in knowing what subsets of an organism's genes are being expressed at a given time, and we know the expressed DNA sequences are transcribed into mRNA during the step of transcription in gene expression, it is reasonable to believe that from knowledge of what mRNAs are present in the cell and in what quantities, we can make some inferences about the current state of that particular cell. Thus, a biochemical experiment can be designed in a way as by utilizing a probe which consists of a grid or array of single-stranded DNA molecules, whose sequence is known, one applies an immobilized target of interest, which consists of a heterogeneous mixture of mRNA molecules of unknown composition, to the probe and try to measure the amount of presence of hybridization.

Usually a radioactive or fluorescent substance is used to help label the target. Because the probe will hybridize only to sequences complementary to its sequence, we can wash off all unhybridized sample probes and the intensity of fluorescence should be proportional to the number of molecules of target bound to the probe.

1.4 Microarray Technologies

There are two major types of microarrays, cDNA microarrays (Schena *et al*., 1995) and oligonucleotide microarrays (Lockhart *et al*., 1996). These two microarray technologies are the very earliest and are currently the most widely used. However some other competing technologies are emerging. A brief description of these two most popular microarray technologies is given in this section.

1.4.1 cDNA Microarrays

cDNA microarrays, also known as robotically spotted microarrays, were introduced into common use at Stanford University and first described by Schena *et al*. in 1995. It usually consists of probes of cDNA robotically printed on a microscope slide. To make the array, a robotic spotter mechanically picks up specific cDNA sequence and deposits them in specific locations in the grid on the glass slide to create specific probes.

There are several advantages and disadvantages of using cDNA microarrays. (1) The first advantage of using cDNA arrays is that one can customize cDNA microarray chips for a specific purpose by designing a layout and directing a robotic spotter to make these microarrays. However, this wonderful customizability may also lead to more possibilities for errors. (2) The second advantage of cDNA microarrays is that, because the cDNA probes are generally several hundred bases long, or even cover the entire cDNA sequences, stringent hybridization conditions can be achieved and the likelihood of cross-hybridization is reduced. However, Kohane *et al*. (2003) pointed out that, even though a long probing subsequence ensures a sufficiently confident representative substring of the original gene, it does not mean that hybridization

conditions will be fully and equally optimized for all species of cDNA subsequences, because the probe-sample probe hybridization rate is known to be a function of the guanine-cytosine (GC) content of a transcript. (3) The third advantage is that we can apply two different RNA samples, say one treatment and one control, on to one common cDNA microarray substrate at the same time.

1.4.2 Oligonucleotide Microarrays

Oligonucleotide microarrays have oligonucleotide probes lithographically synthesized directly on the array. The array in this case is a silicon chip instead of a glass slide (Fodor *et al*., 1991). Currently, the Affymetrix GeneChip is the most popular oligonucleotide expression array technology. In an oligonucleotide array, each unique gene is represented by a probe set, which consists of 11-20 probe pairs. Each probe pair includes one *perfect match* (PM) oligonucleotide probe and one corresponding *mismatch* (MM) probe. The perfect match probe consists of a 25-mer oligonucleotide that is exactly complementary to a 25-mer oligonucleotide sequence of an exon of the target gene. It is designed in such a way as to hybridize to different regions of the RNA that is corresponding to an expressed gene and act a series of multiple independent detectors for the gene.

The mismatch probe is identical to PM except that the middle base (the 13^{th} position) of the corresponding perfect match probe is changed to a different nucleotide. The purpose of a mismatch probe is to serve as an internal control of hybridization specificity peculiar to its particular hybridization site. We expect that the mismatch probe should not hybridize well to the target transcript but it should hybridize to other transcripts that the perfect match oligonucleotide cross-hybridizes to. Thus the hybridization to the gene by the perfect match probe should be stronger than any other nonspecific hybridization to the

mismatch probe. Theoretically, if the PM intensities are consistently larger than the corresponding MM intensities for a probe set (gene), it is more likely to be an indication of the actual presence of mRNA corresponding to that gene in the sample as opposed to being a random chance event.

The oligonucleotide microarray technology has its own advantage over the cDNA microarray technology. The oligonucleotide microarray allows for more genes to be screened or assayed on one single microarray chip due to its higher density of probe pairs. Therefore, it is unnecessary for scientists to restrict the number of genes that are to be scanned. However, due to the limitation of the current technology, only one experiment can be performed on one single microarray chip at one time. Instead, if one needs to compare two different RNA samples as would one do in cDNA microarray experiment, he or she has to use two separate oligonucleotide arrays. This will undoubtedly introduce variations which are not expression changes from a control to a treatment condition.

Furthermore, the oligonucleotide microarray requires that the entire sequence of the target be known. That is, if a scientist is interested in studying a specific species while no appropriate oligonucleotide microarray chip exists, then he or she cannot take advantage of this technology, because unlike cDNA microarray chips, oligonucleotide microarray chips cannot be customized at the user's end easily.

In this paper, we will mainly focus on oligonucleotide microarrays.

Chapter 2

Current Procedures for Preprocessing Oligonucleotide Microarray

2.1 Scanned Image

At the end of a DNA microarray experiment, that is, after hybridization of fluorescently labeled cDNA molecules to the microarray probes, a scanner (laser scanning confocal microscope) or a charge-coupled device (CCD) camera is used to obtain a series of images, which record the intensity of fluorescence at each pixel location on the microarray, and those images are to be stored in a 16-bit tagged image file (TIFF) (Simon *et al*., 2003). Before used for any analysis, the image has to be converted into spot intensities.

Converting scanned images to spot intensities usually requires an experimenter to go through three basic steps: *gridding*, *segmentation* and *quantification*. During gridding, the location of each spot in the microarray is defined by assigning coordinates to the center of each spot; in segmentation, we need to separate from the background the set of pixels corresponding to labeled cDNA which is hybridized to its complementary DNA sequence, while the background refers to the set of pixels that correspond to labeled cDNA hybridizing nonspecifically to the microarray; at the end, the quantification step assigns intensity values to every spot. More detailed information about this procedure is provided by Yang *et al*. (2000, 2001).

2.2 Affymetrix GeneChip Expression Measures

Once the raw oligonucleotide microarray images are converted into a quantitative data set, it is usual to take an appropriate procedure which can eventually give a measure of expression that represents the amount of the corresponding mRNA species in the microarray experiment. This procedure, which aims at obtaining one single expression value for one gene per array, instead of values for the 11-20 probe pairs (PM and MM probes) in a probe set, usually involves the following three steps: *background correction*, *normalization* and *summarization*.

2.2.1 Background Correction

Ideally, we would expect the intensities of those pixels, which are not corresponding to spots in the scanned microarray images, to be zero, but this practically never happens. In fact, those pixels often emit some nonspecific fluorescence because of various reasons such as nonspecific binding of the labeled sample to the array substrate and auto-fluorescence. Although the level of this emission may be low, it is not trivial to ignore anyway.

Because of the existence of this *background* fluorescence, it is more reasonable to assume that the observed spot intensity is really the result of the additive combination of the true spot intensity and the background one. Thus, it would be more proper if an adjustment that subtracts the background from the raw spot intensity values can be performed. Currently, there are various approaches to achieve background correction, and it has been shown that the choice of background correction method can have a large impact on the final output such as log ratios (Yang *et al*., 2002c; Jain *et al*., 2002).

Current methods of background correction include the followings:

(1) Global Background Correction

Global background correction uses the same constant value to represent the background for all spots. The constant could be calculated as the average intensity of all the pixels not belonging to the spots. Although this approach looks simple and easy to perform, it does not take into account the fact that the background variations over the entire microarray do exist.

(2) Spot Background Correction

The spot background correction, also called regional background correction, provides more flexibility than global background correction. One can subtract the spot background from the raw spot intensity value to yield a spot background-corrected spot intensity value. However, it is often found that spots with high intensity tend to have high spot local background, whereas spots with low intensity tend to have low spot background. This is because the segmentation process is usually imperfect and the spot background often contains a contribution from the signal.

(3) Smoothed Background Correction

Since the experimental effects such as hybridization artifacts, the washing process, and the scanning variation usually vary gradually across slides, we have reason to believe the true variation in background across an array should be smooth too. The background may be smoothed by running a simple smoothing procedure through the array. Yang *et al*. (2001) applied an algorithm called *morphological opening* for this purpose.

2.2.2 Normalization

Normalization is a crucial step for comparing the gene expression values between arrays in the analysis of DNA microarray experiment. Even though microarrays may be treated exactly in the same way during a DNA microarray experiment, scientists often find substantial differences in intensity measurements among microarrays. This annoying phenomenon has nothing to do with the samples' own biological features, but instead is due to a variety of systematic effects such as the concentration and amount of DNA placed on the microarrays, arraying equipment such as spotting pins that wear out over time, mRNA preparation, etc.

The objective of normalization is to remove the effects of any systematic sources of variation as much as possible by adjusting the gene expression values of all genes on the array, so that the genes that are not really differentially expressed have similar values across the arrays. Many studies show that the normalization step has a great impact on the final expression measures (Bolstad *et al*., 2002).

There are a large number of normalization methods that are being used in either academia or industry. Generally, they can be divided into two categories: *global* or *linear normalization*, and *intensity-based normalization*. Global or linear normalization schemes assume that the spot intensities on every pair of arrays that is being normalized are linearly related with no intercept. In intensity-based normalization, the transformed spot intensity data is normalized using a nonlinear normalization function $X \rightarrow f(x)$. For an intensity-based normalization, there must be a reference or baseline microarray to which all the microarrays are normalized.

2.2.3 Summarization

In oligonucleotide microarrays, each unique gene is represented by a probe set, which consists of 11-20 probe pairs. In order to finally obtain one single expression value for each gene, the summarization step has to be performed, in which various mathematical or statistical techniques are used.

2.2.3.1 The Average Difference

Since each probe pair includes one perfect match (PM) oligonucleotide probe and one corresponding mismatch (MM) probe, and if we let PM_{gi} and MM_{gi} denote the background-corrected spot intensity measurements for the ith perfect match probe and mismatch probe respectively for gene g, thus $Y_{gi} = PM_{gi} - MM_{gi}$ serves as a measure of the hybridization level of the ith probe of gene g, an intuitive estimate of the expression value for gene g would be simply the arithmetic mean of the intensity differences between PM and MM for all probe pairs. This is called the *Average Difference* ($AvgDiff$) and a mathematical formula is defined as

$$AvgDiff_g = \frac{1}{N_p}\sum_{i=1}^{N_p}\left(PM_{gi} - MM_{gi}\right) = \frac{1}{N_p}\sum_{i=1}^{N_p} Y_{gi} \qquad (2.1)$$

, where N_p is the number of probe pairs in a probe set g (gene g). A modified version of this formula was adopted by one of Affymetrix's early approaches, in which any extreme value of Y_{gi} (three standard deviations away from the mean) is trimmed off.

Although the average difference method looks simple and intuitively correct, it ignores the fact that MM probes may also measure some specific

binding (Wu *et al*., 2004). Therefore, an *ideal mismatch value*, denoted by IM_{gi}, should be obtained by adjusting the value of MM_{gi} before we can subtract it from its corresponding PM_{gi}. If the value of PM_{gi} is greater than MM_{gi}, the difference Y_{gi} represents a possible measure of the true hybridization level for the ith probe of probe set g, and IM_{gi} is usually set to MM_{gi}. However, if the value of PM_{gi} is less than MM_{gi}, the difference Y_{gi} is then negative, hence no longer represents a possible measure of the hybridization level. To solve this problem Affymetrix recommends using an algorithm they developed for calculating a value of IM_{gi} such that the value of IM_{gi}, which is then based on the behavior of the totality of probes in the gth probe set, will be between 0 and PM_{gi}. After obtaining the values of Y_{gi}, an average based on Y_{gi} is then calculated through their one-step biweight mean, and finally converted back to the original scale. This procedure is called the *weighted average difference*.

2.2.3.2 Model-based Approach

A model-based approach for summarizing gene expression for oligonucleotide arrays was proposed by Li and Wong (2001b). For each probe set in an array, let PM_{ij} and MM_{ij} denote the probe intensities for the perfect match and mismatch probe respectively, where i refers to the sample (array) index and j refers to the probe pair index. Suppose θ_i represents the true expression level of the probe set in the ith array, the *Li-Wong* model assumes that the observed measurements for PM_{ij} and MM_{ij} are linear functions of θ_i; and for a truly expressed gene, there exists a factor ϕ_j that makes the strength of the PM_{ij} versus θ_i relationship greater than the strength of the MM_{ij} versus θ_i relationship. The *Li-Wong* model can be defined mathematically as follows:

$$MM_{ij} = \nu_j + \theta_i \alpha_j + \varepsilon \tag{2.2}$$

$$PM_{ij} = \nu_j + \theta_i \alpha_j + \theta_i \phi_j + \varepsilon \tag{2.3}$$

, where ν_j is the baseline response of the jth probe pair due to the nonspecific hybridization, α_j is the rate of increase of the MM response of the jth probe pair, ϕ_j is the additional rate of increase in the corresponding PM response, and ε is a random error term.

The difference between the PM and MM based on the *Li-Wong* model gives an even simpler model as follows:

$$y_{ij} = PM_{ij} - MM_{ij} = \theta_i \phi_j + \varepsilon_{ij} \tag{2.4}$$

To make the model identifiable, we need to put a constraint on ϕ_j by simply letting $\sum_j \phi_j^2 = J$, where J stands for the number of probes in the probe set. The model parameters are then estimated using maximum likelihood.

2.2.3.3 Only Using Perfect Match Probes

Studies have shown that mismatch (MM) probes often contain too much target signal to function as a true measure of nonspecific hybridization. In fact, the MM intensities exceed the PM intensities for about one third of the probes (Irizarry *et al*., 2003). This results in a negative estimate of the differences between PM and MM probes, which is theoretically impossible. Furthermore, it also makes it impossible to apply to the data some types of transformation techniques that may be commonly used in statistics (e.g. log-transformation, square-root transformation, etc.). Many methods of gene expression summarization now turn to only using perfect match (PM) probes instead of utilizing PM and MM

altogether. For the average difference method, we may now calculate the gene expression only based on the PM values. Since the distribution of PM probes is usually skewed, instead of directly taking their arithmetic mean, it is often better to transform them into log scale.

Another method of calculating gene expression level for oligonucleotide arrays, which also only uses PM intensities, is the *Robust Multi-chip Average* (RMA) method. In the summarization step, RMA uses an additive model to fit the PM probes (after background correction, normalization and log-transformation) and applies a robust method called *median polish* to fit the additive model. More details will be given in section 2.3.3.

2.3 Current Popular Methods for Computing Gene Expression

As the interest in the development of preprocessing methodology for the Affymetrix GeneChip technology continues to boom, a great number of various procedures for computing gene expression become available to both the academia and the industry, and new ones are on the way coming out. Most of these preprocessing methods require going through three steps as discussed in section 2.2, while some of them may drop the background correction step for their own reason. Irizarry *et al*. (2006) provide a list of some popular methods for computing gene expression level. In this section, we will briefly introduce the following three well-known methods: dChip (Li and Wong, 2001), MAS 5.0 (Affymetrix, 2002), and RMA (Irizarry et al., 2003).

2.3.1 dChip

DNA-Chip (dChip) is a Windows software package for probe-level and high-level analysis of Affymetrix GeneChip Microarrays based on the Li-Wong model discussed in section 2.2.3.2. The background correction step for the dChip method is theoretically based on Equation (2.2) and (2.3), which include a model-based expression index (MBEI) θ_i as the measure of gene expression. The model is then reduced to a simpler formula as shown in Equation (2.4) by subtracting MM_{ij} from PM_{ij}. In the normalization step, dChip uses the invariant set method, which attempts to base normalization only on those probes that are not differentially expressed between chips. In the summarization step, Equation (2.4) is fit using a maximum likelihood method to obtain the estimate of θ_i.

2.3.2 MAS 5.0

The Affymetrix Microarray Suite (MAS) 5.0 method was developed by Affymetrix to replace its MAS 4.0 method for its GeneChip system. As discussed in section 2.2.3.1, MM probes may also measure some specific binding and as a result, the intensity of MM probes sometimes may be greater than the intensity of PM probes. This will give a negative difference value for $PM_{ij} - MM_{ij}$. MAS 5.0 uses a multistep process to ensure that no negative intensity will occur for the background correction step. For the normalization step, MAS 5.0 conducts a scale normalization that adjusts all chips to have the same mean intensity. For the summarization step, MAS 5.0 uses the one-step biweight mean $T_{biwt}\{X_{gi}\}$ on the log difference, as shown in Equation (2.5), between the PM and IM

$$X_{gi} = \log\left(Y_{gi}\right) = \log(PM_{gi} - IM_{gi}) \ , \tag{2.5}$$

where IM_{gi} stands for *ideal mismatch value* which is discussed in section 2.2.3.1. The log transformation reduces the skewness of distribution of Y_{gi} and the use of the one-step biweight mean reduces the influence of outliers on the final estimate.

2.3.3 RMA

The robust multi-chip average (RMA), which was proposed by Irizarry *et al*. (2003), has become a more and more popular gene expression measure. In RMA algorithm, only the values of PM probes are used. RMA method adopts a background plus signal model, which is shown in Equation (2.6), where PM_{gij},

$$PM_{gij} = bg_{gij} + s_{gij} \tag{2.6}$$

bg_{gij} and s_{gij} are the PM intensity, background signal and probe-specific signal respectively for the ith probe of the jth array in probe set g. The background signal bg_{gij} could be caused by optical noise and non-specific binding. In RMA method, the background correction step is accomplished by estimating the probe-specific intensities $E\left(S_{gij} \mid PM_{gij}\right)$. RMA uses the quantile normalization method (Bolstad *et al*., 2003), which is an aggressive form of normalization that makes the distribution of the spot intensities as similar as possible over all microarray chips. In quantile normalization, either a subset of quantiles or all the quantiles may be equated.

In RMA method, after background corrected and quantile normalized, the PM intensities will be also transformed into a logarithmic scale (usually $\log_2$), then an additive model (2.7) will be fit to compute the gene expression level.

$$y_{gij} = \mu_g + \alpha_{gi} + \beta_{gj} + \varepsilon_{gij} \tag{2.7}$$

In model (2.7), μ_g denotes the overall typical value for the probe set g, α_{gi} is the probe effect, and β_{gj} represents the array effect. Usually, we assume the error term ε_{gij} follows a normal distribution with mean 0 and variance σ^2. The gene expression for probe set g (i.e. gene g) is then computed by adding the estimate of μ_g and β_{gj} as shown in Formula (2.8).

$$S_g = \hat{\mu}_g + \hat{\beta}_{gj} \tag{2.8}$$

To obtain an additive fit in the form of model (2.7), the RMA method uses an iterative process, called *median polish*, which iteratively finds and subtracts row medians and column medians from the data matrix (Hoaglin *et al*., 2000). Median polish is a robust method of fitting an additive model that is in some ways analogous to analysis of variance (ANOVA). The iterative process can be stopped when all rows and columns have zero median. However, a small number of iterations are usually sufficient to obtain satisfactory estimates of the factor effects. In practice, as Hoaglin *et al*. pointed out, it is often good enough by using two iterations and usually no more than four iterations will be needed for most of the situations.

As a matter of fact, the three-step procedure of the RMA method resembles the *median-of-median* (MOM) method proposed by Amaratunga and Cabrera (2001a, 2001b), which uses a three-step process consisting of (i) background subtraction; (ii) standardization by quantiles (i.e. quantile normalization), and (iii) median-of-median (MOM) estimates. The third step in the MOM method uses a highly resistant estimator for μ_g as shown in (2.9)

$$S_g = \text{median}_j \text{median}_i \left(Y_{gij} \right). \tag{2.9}$$

It can also be calculated by a one-step median polish. Then one can proceed with statistical analyses at the gene level. Table 1 gives a comparison between RMA and MOM.

Table 1 Comparison of RMA and MOM

	RMA (Irizarry *et al*., 2003)	MOM (Amaratunga and Cabrera, 2001a, 2001b)
1	Background correction	Background subtraction
2	Quantile normalization	Quantile normalization
3	Median Polish	One-step Median Polish
4	Statistical analysis	Statistical analysis

Chapter 3
Statistical Analysis on Individual Genes

3.1 Introduction

In the previous chapter, we discussed the preprocessing procedures about the scanned image from a DNA microarray experiment. Once the scanned images from replicate samples have been converted, through a specific preprocessing method that usually involves background correction, normalization, and summarization (as discussed in Chapter 2), into a numerical data matrix that finally contains gene-level expressions for microarrays, a wide variety of investigations with a wide variety of objectives can be carried out through applying some particular statistical procedures. Many study objectives of interest can be classified into the following three categories:

(1) Class Comparison

Class comparison is a natural objective for DNA microarray experiments. The goal focuses on determining whether gene expression levels of a set of genes across two or more conditions are significantly different. The classes may represent different tissue types, or the same tissue type but under different experimental conditions or under different classes of individuals. For example, an experimenter might be interested in comparing the gene expression levels of several genes in a cancer study, which may involve healthy liver cells and

cancerous liver cells, or for the cancerous liver cells, the tissue taken before or after medical treatment or some kind of experimental intervention.

(2) Class Prediction

Unlike class comparison, class prediction emphasizes on developing a statistical model that can be used to categorize a new specimen (say a new tumor). Because it is likely that different genes are expressed in the cells of different tumor classes, it should be possible to differentiate among the tumor classes by studying which genes are informative for distinguishing the predefined classes, and apply class prediction or supervised classification techniques to develop a classification rule to discriminate them. It has important role for medical problems in diagnostic classification, prognostic prediction, and treatment selection (Simon *et al*., 2003)

(3) Class Discovery

Class discovery focuses on the identification of subtypes of specimens or genes within the same population. The idea is that important biological differences among specimens that are clinically and morphologically similar may be distinguishable at the molecular level. A popular approach to class discovery involves grouping similar genes or samples together using methods such as k-means or hierarchical clustering. Studies using class discovery may help develop improved medical treatments by uncovering biological features of diseases.

In this paper, we will only focus on statistical methods for comparing microarrays of two groups. The following sections will provide reviews on current statistical methods for two-group comparison of individual genes.

3.2 Two-group Comparison of Individual Genes

In a DNA microarray experiment, scientists are often interested in conducting experiments on two groups of mRNA specimens whose biological characteristics are different in some manner independent of the expression profiles, and using some statistical methodologies to identify genes that are significantly differentially expressed over the two groups. Considering a set of G genes in two groups of microarrays, denoted by Group 1 and Group 2 respectively, suppose there are n_1 microarrays in Group 1 and n_2 microarrays in Group 2, thus the total number of arrays is $N = n_1 + n_2$. Let x_{gj} represent the gene intensity of Group 1 for the gth gene and the jth microarray (in Group 1), and similarly let y_{gj} be the gene intensity of Group 2 for the gth gene and the jth microarray (in Group 2). That is, for each gene g, Group 1 contains gene intensities x_{g1} through x_{gn_1} and Group 2 contains gene intensities y_{g1} through y_{gn_2} (see Table 2). It is assumed that all data have been transformed and normalized. For example, in the RMA method, the data should be converted from the raw probe-level data to the gene-level data by going through background correction, quantile normalization, and, after log2-transformation, finally summarization using median polish.

Table 2 DNA Microarray data (gene-level)

Gene	**Group 1**			**Group 2**		
1	x_{11}	…	x_{1n_1}	y_{11}	…	y_{1n_2}
⋮	⋮	x_{gj_1}	⋮	⋮	y_{gj_2}	⋮
G	x_{G1}	…	x_{Gn_1}	y_{G1}	…	y_{Gn_2}

3.2.1 Fold Changes

The simplest and most common situation is to compare one gene over the two groups at a time. Our interest is to know, for one particular gene in the microarray data, whether the mean of expression level for Group 1 is significantly different from the mean of expression level for Group 2. If we let $\bar{x}$ denote the mean of the expression level for Group 1 and $\bar{y}$ the mean of the expression level for Group 2 (without confusion, we can omit the subscript *g* for genes), it is clear that large absolute difference of $\bar{x}$ and $\bar{y}$ may indicate the gene is differentially expressed. To give a quantified rule of determining whether a gene is differentially expressed, early analyses adopt a method which compares fold increase or fold decrease with a specified threshold. The *fold change* (or *log fold change*) is defined as $\bar{x} - \bar{y}$. In general, the decision rule that declares a gene to be differentially expressed over groups based on an *h*-fold (or greater) change is to see if $|\bar{x} - \bar{y}| > \log(h)$ holds. A problem with this approach is that it can lead to a high probability of declaring a gene to be differentially expressed when it is not (Miller *et al*., 2001) and the choice of fold is somewhat arbitrary. It utterly ignores the variability of gene intensities as an influential factor on determining whether a gene is significantly differentially expressed (Amaratunga and Cabrera, 2004).

3.2.2 Two-sample t-Test

The two-sample *t*-test is the most commonly used method that also takes into consideration the variability of gene expression for comparing two groups. It assumes that the gene intensities of the two groups follow normal distributions independently, that is

$$X_1, X_2, \ldots, X_{n_1} \sim i.i.d.\ \ Normal\left(\mu_1, \sigma_1^2\right) \tag{3.1}$$

and

$$Y_1, Y_2, \ldots, Y_{n_2} \sim i.i.d.\ \ Normal\left(\mu_2, \sigma_2^2\right). \tag{3.2}$$

The hypotheses are

$$H_0 : \mu_1 = \mu_2 \quad \text{vs.} \ \ H_1 : \mu_1 \neq \mu_2 . \tag{3.3}$$

If we assume the two samples have equal variances, that is $X_j \sim Normal\left(\mu_1, \sigma^2\right)$ and $Y_j \sim Normal\left(\mu_2, \sigma^2\right)$, we may define the *t*-test statistic *T* as

$$T = \frac{\overline{x} - \overline{y}}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}, \tag{3.4}$$

where

$$s_p^2 = \frac{\left(n_1 - 1\right) s_1^2 + \left(n_2 - 1\right) s_2^2}{n_1 + n_2 - 2} \tag{3.5}$$

is the pooled variance estimate, and s_1^2 and s_2^2 are the sample variances for the two groups respectively. Under the null hypothesis H_0, T follows a *t*-distribution with degrees of freedom $n_1 + n_2 - 2$. A rejection criterion is then based on the observed value of T, denoted by T_{obs}, such that a gene is declared significantly differentially expressed if

$$p = \Pr\left(|T| > T_{obs}\right) < \alpha, \tag{3.6}$$

where α is a pre-selected testing level (usually 0.05) and p is called the *p*-value.

A test based on Equation (3.4) is true only when it is reasonable to assume that the two populations, which are also assumed normally distributed, have equal variances. However, the assumption of equal variances is difficult to be substantiated considering the small sample sizes of microarray data. When such assumption is untenable, the traditional *t*-test may tend to have a higher false positive rate than expected (Amaradatunga and Cabrera, 2004).

To overcome this problem, a modified version of *t*-test with Welch correction (also called a Welch's test) can be used. The Welch correction is to provide a valid *t*-test in the presence of unequal population variances. The Welch's test statistic is defined as follows:

$$T = \frac{\overline{x} - \overline{y}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}}. \tag{3.7}$$

The *t*-statistic shown in Equation (3.7) follows approximately a *t*-distribution with a corrected number of degrees of freedom ν to assess the significance of the *t*-statistics, where ν is defined as the next smaller integer of the value obtained from the following equation:

$$\nu = \frac{\left(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right)^2}{\frac{\left(s_1^2/n_1\right)^2}{n_1 - 1} + \frac{\left(s_2^2/n_2\right)^2}{n_2 - 1}}. \tag{3.8}$$

When the variances of the two groups are equal, Equation (3.8) reduces to the usual degrees of freedom $n_1 + n_2 - 2$ if the two groups have equal numbers of observations; however, if the two groups have different numbers of observations,

Equation (3.8) gives a smaller value than $n_1 + n_2 - 2$, which makes the *t*-test using Welch correction too conservative.

3.2.3 SAM *t* and Conditional *t*

The original two-sample *t*-test should work fine for most of the situations that require comparison of two groups; however it may encounter some problems when it is applied to DNA microarray data. Because the sample size in a common DNA microarray experiment is usually very limited, the original two-sample *t*-test often shows low statistical power of detecting differentially expressed genes. The very small sample size makes it difficult to estimate the standard errors of microarray data well and small standard errors can appear completely by chance. This may result in a high false positive rate for genes that happen to have low variability and a high false negative rate for genes that happen to have high variability.

One solution to solve the abovementioned problem is to include a carefully chosen constant to the denominator of the *t*-statistic. A modified *t*-statistic then can be defined as follows:

$$T(c) = \frac{\bar{x} - \bar{y}}{s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} + c} \tag{3.9}$$

The idea is to choose an intermediate positive value of c such that the dependence of $T(c)$ on $s_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$ is minimized. This approach was suggested by Tusher *et al.* (2001) and the statistic $T(c)$ is often called the SAM *t*-statistic, where SAM means "significance analysis of microarrays".

Another solution to solve the problem discussed in the first paragraph of this subsection is proposed by Amaratunga and Cabrera (2003c), which addresses the dependence of T on $s = s_p\sqrt{\frac{1}{n_1}+\frac{1}{n_2}}$ by determining the critical value of T from the distribution of T conditioned on s.

3.3 Multiple Testing

Because of the nature of DNA microarray experiments, usually a very huge number of statistical analyses are performed simultaneously. These simultaneous tests could result in high false positive rate and thus high expected number of false positives. Suppose we have n many genes in one microarray experiment and we want to perform n many *t*-tests to compare arrays that are divided into two groups (phenotypes). Each test is done at level α. Then, if those tests are independent, the probability of making at least one false positive is

$$\Pr(\text{at least one false positive in } n \text{ tests}) = 1-(1-\alpha)^n \tag{3.10}$$

If n is large (and it is often the case in microarray experiments), Equation (3.10) would be very close to one, which implies one may definitely find at least one false positive in n many tests. Furthermore, if n is large, the expected number of false positives, $n\alpha$, could be an overwhelmingly large number and this makes it very difficult to discern the true significant genes.

Early methods to overcome this problem due to multiple testing focus on adjusting the *p*-values of each test so that the *familywise error rate* (FWER), the probability of having at least one false positive, is less than a specified level. For example, in Bonferroni adjustment, the *p*-value for the *k*th test is simply

$\tilde{p}_k = K \cdot p_k$, assuming there are K many tests and p_k is the original *p*-value for the *k*th test. Apparently, methods like Bonferroni adjustment would be too conservative in the sense that while they reduce the number of false positives, they also reduce the number of true discoveries.

Another way to solve the problem due to multiple testing is that, instead of controlling the FWER, one can control the ratio of the number of false discoveries in those tests that result in a discovery. This ratio, called False Discovery Rate (FDR), can be loosely defined as (Storey and Tibshirani, 2001)

$$FDR = E\left(\frac{V}{V+S}\right) = E\left(\frac{V}{R}\right), \tag{3.11}$$

where *V* stands for the number of false positives and S for the number of true positives, thus R would be the number of total discoveries. Benjamini and Hochberg (1995) defined the FDR more precisely as

$$FDR = E\left(\frac{V}{R}\middle| R > 0\right)\Pr\left(R > 0\right) \tag{3.12}$$

Storey (2001a) defined the *positive false discovery rate* (pFDR) as

$$\text{pFDR} = \text{E}\left[\frac{V}{R}\middle| R > 0\right]. \tag{3.13}$$

The pFDR emphasizes the fact that we need to do adjustment only if positive findings are present.

Chapter 4
Robust Estimators in RMA Method

4.1 Review of RMA Method

The RMA method was proposed by Irizarry *et al*. (2003), and has become a popular gene expression measure. In RMA algorithm, only the values of PM probes are used. The PM intensities are to be transformed into a logarithmic scale after the background correction and the quantile normalization steps, and then an additive model (2.7) is fit to compute the gene expression level.

The RMA method uses median polish to obtain an additive fit in the form of model (2.7), that is, it iteratively finds and subtracts row medians and column medians from the data matrix. One could either begin with the rows or begin with the columns, until the process results in no more change in rows or columns, that is, all rows and columns have zero median. Usually, the median polish procedure is done in one step. Figure 1 shows the diagram of the RMA procedure.

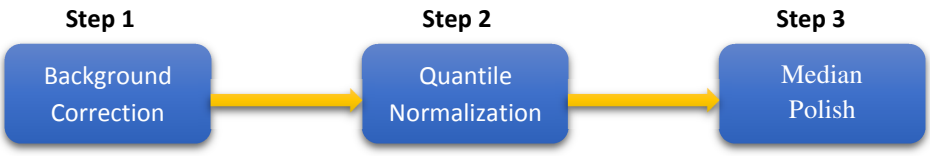


Figure 1 Three Steps of RMA

4.2 Use Sample Mean in RMA

As discussed in the previous section, the RMA method takes advantage of median polish, which iteratively removes current row and column medians, to estimate the row, column and overall effects of an additive model (2.7).

In fact, any measure of location can be applied in this manner to fit the additive model (2.7). A very natural choice of measure of location is the sample mean $\bar{X}$. The steps to obtain the row, column and overall effects using means are similar to the way used in median polish; however it does not need iteration. As a matter of fact, one can obtain the row, column and overall effects as well as the residuals in one step. First, we could compute the grand mean (i.e. the average of all values in the data matrix) and then subtract it from all observations. Second, the row means can be computed and subtracted from their corresponding rows, and then the column means be computed and subtracted from their corresponding columns in the same way. This process can be formularized as follows:

For an additive model (2.7), the estimates for the g-th probe set are:

$$\hat{\mu} = \frac{1}{IJ}\sum_i \sum_j y_{ij} \tag{4.1}$$

$$\hat{\alpha}_i = \frac{1}{J}\sum_i \left(y_{ij} - \hat{\mu} \right) \qquad i = 1,\ldots I \tag{4.2}$$

$$\hat{\beta}_i = \frac{1}{I}\sum_j \left(y_{ij} - \hat{\mu} \right) \qquad j = 1,\ldots J \tag{4.3}$$

, and the residuals are

$$r_{ij} = y_{ij} - \hat{\mu} - \hat{\alpha}_i - \hat{\beta}_j \qquad i = 1,\ldots I;\ j = 1,\ldots J\ . \tag{4.4}$$

The estimates in equations (4.1), (4.2) and (4.3) minimizes the sum of the squares of the residuals

$$SSR = \sum_i \sum_j r_{ij}^2\ . \tag{4.5}$$

However, the sample mean is not a resistant estimator and its performance in estimating the central location of a sample, especially for that with a small sample size, is greatly subject to the presence of outliers.

4.3 Use Other Robust Statistics in RMA

In the previous section, we pointed out that it is possible to adopt other estimators of location in lieu of the sample median used in the summarization step of the RMA method. Although the sample mean is one of the choices, its performance greatly depends on how well the underlying assumptions (e.g. normality, outliers, etc.) are satisfied. Considering that the DNA microarray data usually involves only a small number of experiments, and the complexity of these experiments can often introduce different types of variations along the laboratorial processes, a robust estimator of location, such as the sample median, is absolutely more preferred than the sample mean.

4.3.1 Huber's M-estimator

Besides the sample median, some other robust estimators are also currently available. One of them is *Huber's M-estimator*. As a matter of fact, median belongs to the family of *Huber's M-estimators*. M-estimators are a generalization of

maximum likelihood estimators (MLE's). For MLE's, we try to find an estimator that could maximize the likelihood function (4.6),

$$\prod_{i=1}^{n} f\left(x_i \mid \theta\right) \tag{4.6}$$

or equivalently, minimize (4.7).

$$\sum_{i=1}^{n} -\log f\left(x_i \mid \theta\right) \tag{4.7}$$

Huber (1964) proposed to generalize this by minimizing an objective function which utilizes some function $\rho(x)$ and sums over the sample (see Formula (4.8)).

$$\sum_{i=1}^{n} \rho\left(x_i \mid \theta\right) \tag{4.8}$$

Thus minimizing $\sum_{i=1}^{n} \rho\left(x_i \mid \theta\right)$ can usually be done by differentiating ρ and solving Equation (4.9)

$$\sum_{i=1}^{n} \psi\left(x_i \mid \theta\right) = 0 \tag{4.9}$$

where $\psi\left(x \mid \theta\right) = \frac{\partial}{\partial \theta} \rho\left(x \mid \theta\right)$.

The most familiar M-estimator is the sample mean, which is the least squares estimate of location. The ρ-function for this case is the square of residual (4.10)

$$\rho\left(x \mid \theta\right) = \left(x - \theta\right)^2 \tag{4.10}$$

Thus the ψ-function can be obtained by differentiating (4.10) with respect to θ: $-2(x-\theta)$. By dropping the leading constant 2 and summing over all samples, we can find the estimator by minimizing (4.11),

$$\sum_{i=1}^{n}(x_i-\theta)=0 \tag{4.11}$$

in which $\hat{\theta}=\bar{x}=\frac{\sum_{i=1}^{n}x_i}{n}$, the sample mean, turns out to solve this equation.

Another example of M-estimator is the sample median. For this case, the ρ-function is the absolute value of the residual (see Formula (4.12)),

$$\rho(x\,|\,\theta)=|x-\theta| \tag{4.12}$$

and the corresponding ψ-function is (4.13).

$$\psi(x\,|\,\theta)=\operatorname{sgn}(x-\theta) \tag{4.13}$$

The family of Huber M-estimators uses the following ρ- and ψ- functions given in (4.14) and (4.15) respectively.

$$\rho_k(x\,|\,\theta)=\begin{cases}\frac{1}{2}(x-\theta)^2 & \text{if} \quad |x-\theta|\le k\\ k|x-\theta|-\frac{1}{2}k^2 & \text{if} \quad |x-\theta|>k\end{cases} \tag{4.14}$$

$$\psi_k(x\,|\,\theta)=\begin{cases}x-\theta & \text{if} \quad |x-\theta|\le k\\ k\operatorname{sgn}(x-\theta) & \text{if} \quad |x-\theta|>k\end{cases} \tag{4.15}$$

The sample mean and the sample median are special cases of Huber M-estimators that are corresponding to the limit cases $k\to\infty$ and $k\to 0$ respectively (Maronna *et al.*, 2006).

4.3.2 Tukey's Biweight

Tukey's biweight estimator (also called *bisquare*) is also a popular M-estimator. It was proposed by Tukey in the 1970s. Its corresponding ψ-function is given in (4.16).

$$\psi(x \mid \theta) = \begin{cases} (x-\theta)\left[1-\left(\frac{x-\theta}{C}\right)^2\right]^2 & \text{if} \quad |x-\theta| \le C \\ 0 & \text{if} \quad |x-\theta| > C \end{cases} \tag{4.16}$$

The family of Tukey's biweight M-estimators is also known as redescending estimator because its corresponding ψ-function goes back to 0 when the absolute value of $x-\theta$ is greater than a specified positive number C. Hoaglin *et al.* (2000) suggest one-step Tukey's biweight which uses one step of iteration when iteratively reweighting data points starting with an estimator (e.g. the median).

4.4 Comparison of Mean, Median, Tukey's Biweight and Huber's M-estimators

All of the abovementioned estimators except Mean are considered as robust statistics in the sense they are not excessively affected by outliers or other small departures from model assumptions. A simulation study can be carried out to compare the performances of the three robust estimators, Median, Tukey's Biweight and Huber's M-estimators as well as the sample mean. Usually, one can generate a "contaminated" dataset under a mixture model, which may include a small amount of contamination. For example, we can use a mixture of 95% an underlying distribution and 5% the same distribution with the same mean but significantly higher standard deviation to represent outliers.

In the following example, we generated a microarray probe-level data with 1000 genes and 12 samples. Those 12 samples are divided into two groups, one as the control group containing 6 samples, and the other as the treatment group containing the other 6 samples. Each gene (also known as probe set) consists of 11 probes according to the general Affymetrix GeneChip microarray structure. The generated data matrix is thus 11000 by 12 in dimensions. The data matrix is generated from a normal distribution with mean 0, and with standard deviation 0.5 for 95% of the data points (randomly chosen) and standard deviation 1, 2, 3 and 4 respectively for each run for the rest 5% data points. Next a list of increasing difference values (in this example, 0, 0.1, 0.2 and 0.4) are added to all data points in the data matrix for one of the two groups, and then one of the location estimators is used in RMA to obtain the gene expression. We want to find out the overall patterns of the discovery rate of significant genes (i.e. the proportion of significant genes out of the 1000 genes) when using these estimators, by performing a series of two-sample *t*-test on each gene. Theoretically speaking, we should expect to see the discovery rate is around 5% when 0 is added to the treatment group (i.e. no differential expression for genes) for all estimators provided that we use 5%-significance level and an increasing discovery rate when the added difference value is increasing. Meanwhile, we also expect to see the discovery rate of significant genes decrease when we increase the value of standard deviation of the second normal distribution, as it represents the outliers in the "contaminated" data matrix.

Figure 2 shows the comparisons of the performances of detecting differentially expressed genes when using mean, median, Tukey's biweight and Huber's M-estimator as the location estimator in RMA. For all of the four figures, the x-axis represents the standard deviations in the second normal distribution in

the mixture, and the y-axis represents the proportion of discovered significant genes. Clearly, when no differential expression is present, all estimators perform relatively well by giving roughly 5% discovery rate, except that the discovery

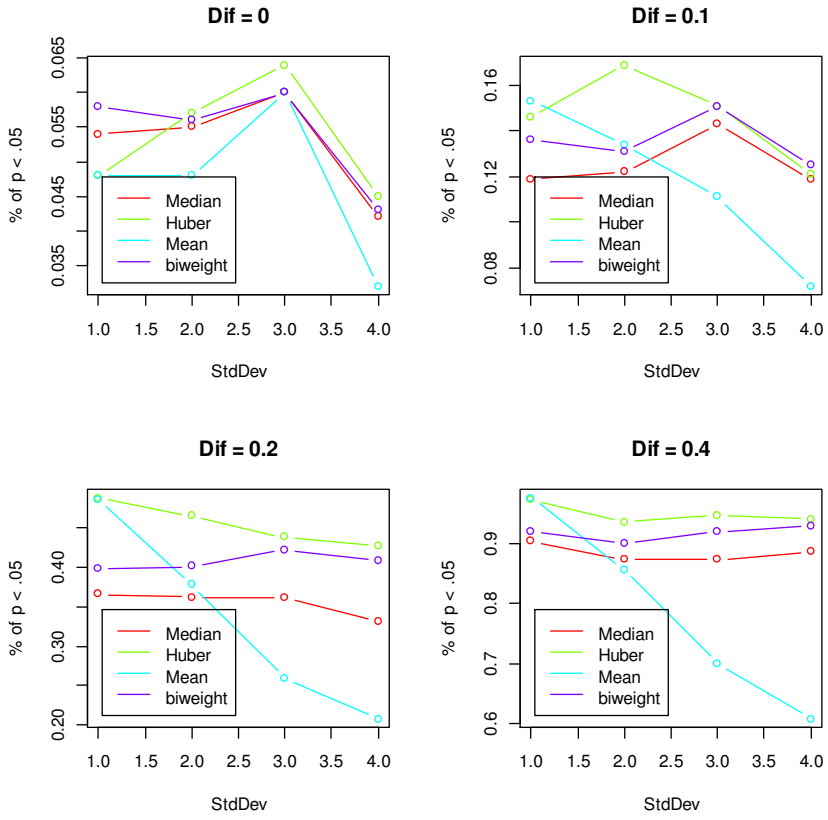


Figure 2 Comparisons of Estimators in RMA (5% contamination)

Simulation data generated from a mixture of 95% Normal(0, std = .5) + 5% Normal(0, std = 1, 2, 3, 4). A difference value (dif = 0, 0.1, 0.2, 0.4) added to the treatment group for each run.

a. Dif = 0 : all estimators perform relatively well by giving roughly 5% discovery rate, except that the discovery rate using mean drops to below 0.035 when the standard deviation is increased to 4 for the second normal distribution.
b. Dif = 0.1, 0.2, 0.4 : Discovery rate increasing as Dif increasing for all estimators.

Discovery rate drops as StdDev increasing for mean.

rate using mean drops to below 0.035 when the standard deviation is increased to 4 for the second normal distribution. It implies that the sample mean may not be a good choice when outliers are present. As the difference value increases, the

discovery rate of significant genes is, as is expected, increasing too. And it is obviously going down as the standard deviation is also increasing. This trend becomes less obvious when the difference value added to the treatment group becomes bigger for median, Tukey's biweight and Huber's M-estimator. This phenomenon confirms that the robust estimators can still perform well when a small amount of outliers exist in the data set. However, for mean, it keeps dropping significantly as the standard deviation increases, even if there is bigger difference between the control group and the treatment group, which is evidence that the sample mean is not robust under unfavorable conditions.

When the amount of outliers in the dataset increases, the overall performances of those estimators of location may change accordingly too. Usually, the estimators will "break down" when the proportion of outliers becomes too large. Hampel (1968) gives the concept of *breakdown point* of an estimator to describe the sensitivity of an estimator to the presence of outliers in a data set. The breakdown point is defined as the maximum fraction of observations in a sample that, without greatly changing the value of the location estimate, can be outliers.

Surely the higher the breakdown point of an estimator, the more robust it is. The breakdown point of the median is 0.5, and it is the biggest possible one for an estimator of location that treats observations on each side of the estimate symmetrically (Hoaglin *et al.*, 2000). For the mean, the breakdown point is 0 since a big change of a single observation in the data set may make the mean change greatly.

In Figure 3 and Figure 4, we illustrate the performance change of the various RMA methods, based on the mean, the median, Tukey's biweight and

Huber's M-estimators respectively, by adjusting the percentage of outliers in the simulated data set. Instead of having 5% outliers, we increase the proportion of outliers to 20% and 40%, and go through the same procedures as are mentioned above to compare how well the outliers are handled by the four location estimators.

It is not surprising to see the sample mean is still the worst candidate as a location estimator in the RMA method, since the breakdown point of the mean is, as we already know, zero. Its performance may be only acceptable when the standard deviation is small in the second normal distribution, or in another word,

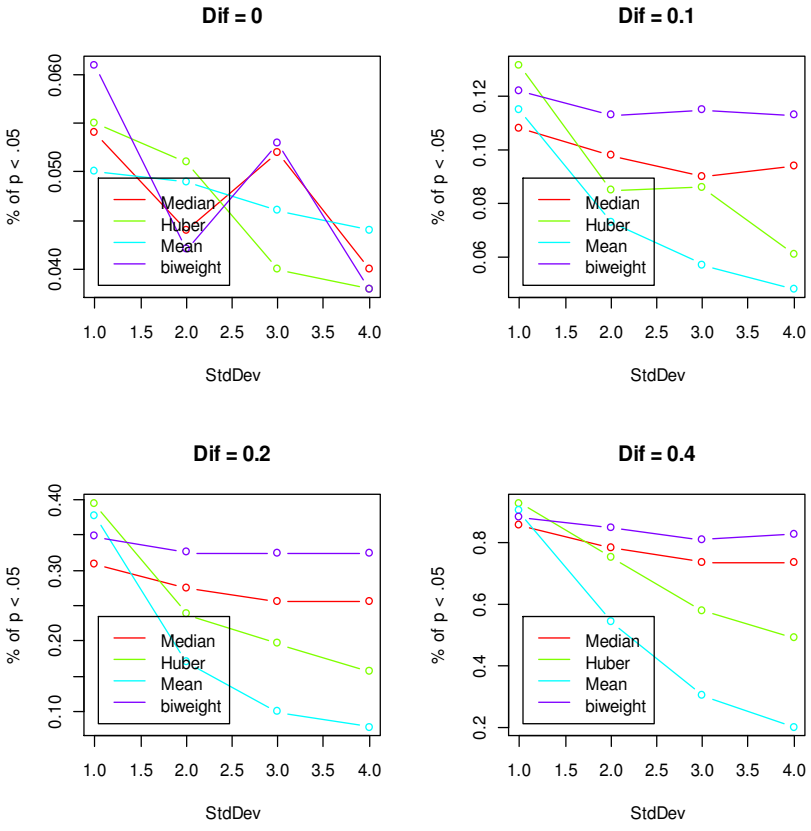


Figure 3 Comparisons of Estimators in RMA (20% contamination)

it can only perform as similarly well as other candidates do when few or even no outliers are present.

Although we have seen that the performance of Huber's M-estimator is the best choice among others when only a small proportion of outliers exist, it is no longer the performance leader after we increased the contamination rate from 5% to 20% and 40%. It is clear that, compared to the median and Tukey's biweight estimator, Huber's M-estimator becomes worse when the standard deviation of the second normal distribution in the mixture increases. Meanwhile, as the contamination rate gets close to 50% (Figure 5), the median becomes the best candidate among other estimators for the RMA method, and it is

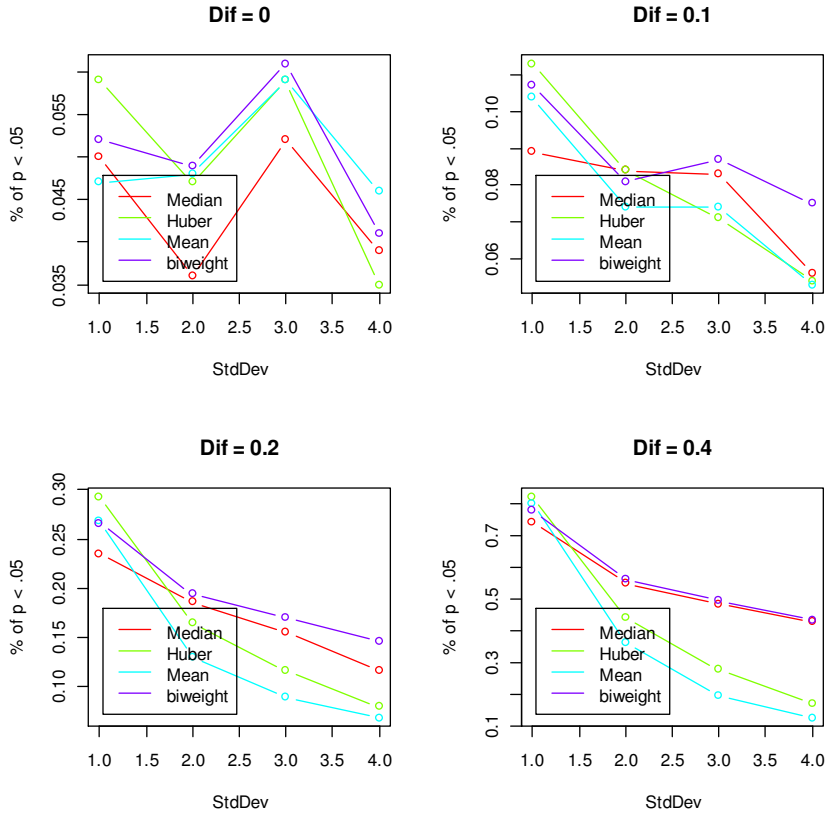


Figure 4 Comparisons of Estimators in RMA (40% contamination)

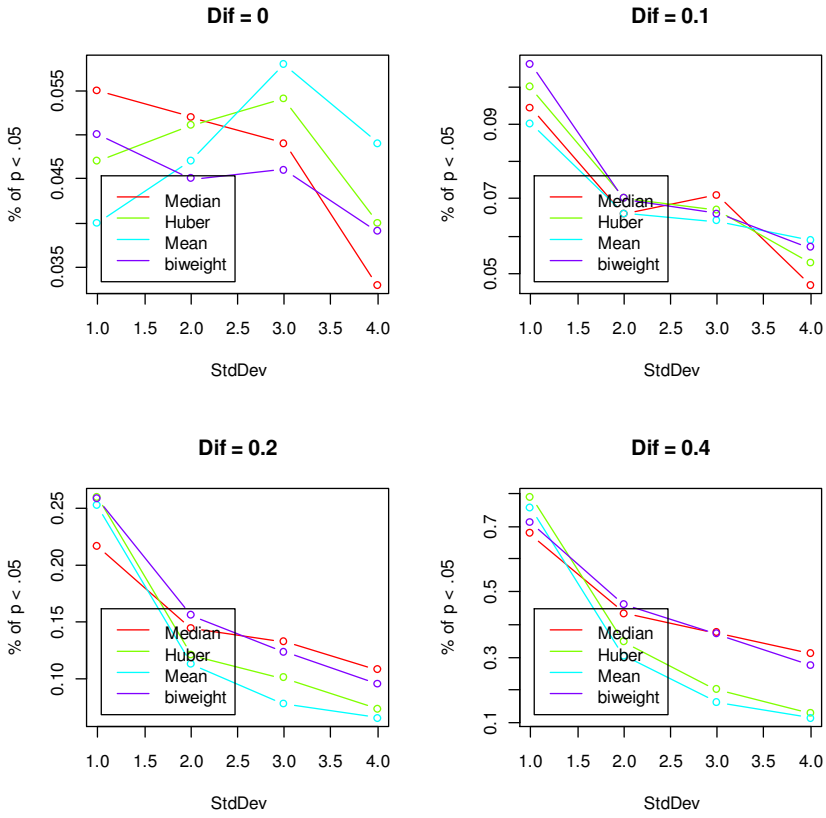


Figure 5 Comparisons of Estimators in RMA (50% contamination)

immediately followed by Tukey's biweight estimator.

4.5 Conclusion

In this chapter, we discussed the possibility of application of different types of location estimators, including the sample mean, the sample median, Tukey's biweight estimator and Huber's M-estimator, in the summarization step of the RMA procedure. The simulation results turn out to favor, as one can expect, the usage of robust estimators such as the median, Tukey's biweight and Huber's M-estimators over the sample mean. In fact, the performance of the sample mean for detecting differentially expressed genes drops significantly when outliers are more likely to be present. Overall, Huber's M-estimator outperforms all of the

other estimators in the sense its discovery rate of significance is higher than that of other estimators provided that the outliers in the data set account for only a relatively small proportion. Tukey's biweight estimator (one-step biweight) also does good jobs, and its performance is better than the sample median used in the regular median polish process under the same situation. When the proportion of outliers in the data set becomes larger, especially when it is close to 50%, the median turns out to beat Huber's M-estimator and the sample mean as well. It performs similarly as Tukey's biweight estimator though.

Huber's M-estimator works much better under the condition that there exist only a small proportion of outliers in the data set. And we can reasonably presume that it is very unlikely to find a large amount of outliers present in a DNA Microarray data set if the DNA Microarray experiments are prudentially carried out. We may conclude that the use of Huber's M-estimator during the summarization step of the RMA method shall be an optimal choice.

Chapter 5

Gene Set Enrichment Analysis (GSEA) on Gene-level Microarray Data

5.1 Issues of Analysis on Individual Genes

As discussed in previous chapters, traditional genome-wide Ribonucleic Acid (RNA) expression analysis, which usually aims at obtaining a list of significantly differentially expressed genes, is based on statistics computed from a collection of samples that belong to one of two or more groups of interest. For example, one may be interested in knowing whether one or more genes may show different expressing behaviors between healthy organisms and diseased ones (e.g. tumor).

Despite the fact that Deoxyribonucleic Acid (DNA) Microarray technology allows biologists to acquire the expression profiles of a large number of genes simultaneously, the statistics for comparing groups are still computed individually for each row in microarray data, treating the associated genes as different entities. Because microarray experiments often come with a considerably small number of samples due to either statistical illiteracy or more possibly financial burden on each microarray chip, the gene-by-gene analysis through the traditional methods of comparison unavoidably shows low statistical power of detecting differentially expressed genes. For example, one may find that no individual gene may be picked out according to the predetermined threshold for statistical significance because the variance among samples could be relatively high compared to the relevant biological differences.

Similarly, it is also possible to pick out too many statistically significant genes if the samples happen to have low variability. Under either circumstance, it could be difficult for biologists to give meaningful interpretation of the microarray data that is being analyzed.

Besides the issue of small sample size of microarray data, the gene-by-gene analysis also ignores the biological knowledge regarding how genes work in concert with each other. As Subramanian *et al.* (2005) pointed out, an increase of 20% in all genes encoding members of a metabolic pathway may be more important than a 20-fold increase in a single gene because it may dramatically alter the flux through the pathway.

5.2 Introduction of Gene Set Enrichment Analysis (GSEA)

Gene Set Enrichment Analysis (GSEA) method was recently proposed by Subramanian *et al*. (2005) to remedy those abovementioned problems. Instead of simply focusing on each individual gene (row) in a given microarray data set at a time, GSEA method tries to determine whether a predefined set of genes would statistically reveal concordant and significant differences between two groups of microarrays that belong to two different phenotypes. The gene sets are usually constructed according to prior biological information like known biochemical pathways, coexpression in previous experiments, etc. Annotation databases such as Gene Ontology (GO) (The Gene Ontology Consortium, 2005), Kyoto Encyclopedia of Genes and Genomes (KEGG) (Kanehisa and Goto, 2000), etc. can provide such information for GSEA.

5.3 Gene Set Enrichment Score

GSEA method can be performed basically in two steps. The first step involves the calculation of an Enrichment Score (ES) for each pre-defined gene set. To achieve this, suppose we have a collection of pre-defined gene sets $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K\}$, which group all (say N) genes in the microarray data into K many gene sets. Then for all genes, we compute a list of association scores $\{r_g, g = 1, 2, \dots, N\}$ for comparing the two means of the two different phenotypic groups. We should note that, since we want to take advantage of a measurement on the difference of that gene's expression in the two phenotypes to formulate the enrichment score, we actually have many choices (e.g. t-test, Wilcoxon rank sum test, etc.) to achieve this goal. We can follow the following procedures to calculate the association scores for each gene:

i. Calculate the mean or median for each gene under one phenotype and denote it with $\tilde{X}$.
ii. Calculate the mean or median for each gene under the other phenotype and denote it with $\tilde{Y}$.
iii. Calculate the difference between the two means or medians: $\tilde{X} - \tilde{Y}$.
iv. Calculate a measure of variability for the difference between the two means or medians and denote it with S.
v. Calculate the association statistic by dividing the difference between the two means or medians by the measure of variability of the difference. If we denote it with r, the formula for obtaining the association score should look like $r = \frac{\tilde{X}-\tilde{Y}}{S}$.

If we choose $\tilde{X}$ and $\tilde{Y}$ to be the sample means from the two phenotypes for each gene and S to be $S_{pooled}\sqrt{\frac{1}{n_x}+\frac{1}{n_y}}$, where S_{pooled} is the pooled standard deviation from the two samples, r will follow a student's t-distribution with $n_x + n_y - 2$ degrees of freedom and a list of such statistics for each gene can be used as the association scores in the following step to obtain the enrichment scores of the gene sets.

Based on the definition of each gene set, the association score (e.g. t-statistic) for each gene then can be linked to the corresponding gene sets. For example, if Gene $g, g \in [1, N]$ belongs to Gene Set $\mathcal{S}_k, k \in [1, K]$, then the association score r_g is included in a set of association scores whose corresponding genes are also in the same gene set. Intuitively, if the gene set $\mathcal{S}_k$ is biologically related to the phenotypes, then it is likely to observe higher association scores for genes linked to gene set $\mathcal{S}_k$ than those linked to other gene sets that are not related to the phenotypes. We can in fact produce a list of genes (let us call it L) based on their association scores and make the list ordered decreasingly with large association scores on the top of the list. It is then reasonable to expect near the top of the list L a large proportion of genes in $\mathcal{S}_k$ to show up and a small proportion of genes in other gene sets that are not related to the phenotypes. For a given position i in the list L, we can find out the proportion of genes out of the genes in Gene Set $\mathcal{S}_k$ that are before the position i and the proportion of genes out of all genes but *not* included in Gene Set $\mathcal{S}_k$ that are also before position i. The maximum difference between the two proportions for all possible position i's, called the Enrichment Score for Gene Set $\mathcal{S}_k$, gives the evidence whether a gene set is enriched or not. A large value of the enrichment score (either positive or negative) implies an enriched gene set because of the large proportion of genes

present near the top of the ordered gene list. If there is no enrichment of Gene Set $\mathcal{S}_k$, the enrichment score should be close to zero because in that case, we should see little difference between the proportion of genes in $\mathcal{S}_k$ that appear near the top of the list L and the proportion of other genes near the top of the list L. As a matter of fact, a Gene Set Enrichment Score S_k for Gene Set $k, k = 1, 2, \dots, K$ can be seen as a signed version of Kolmogorov-Smirnov statistic between the values of $\{r_g, g \in \mathcal{S}_k\}$ and their complement $\{r_g, g \notin \mathcal{S}_k\}$ (Efron and Tibshirani, 2007).

5.4 Mathematical Formulation

A mathematical formula can be given as follows:

For a list of N genes, $L = \{g_1, g_2, \dots, g_N\}$, which is decreasingly ordered based on the association scores $\{r_g, g = 1, 2, \dots, N\}$ between the two phenotypes, and a collection of pre-defined gene sets $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K\}$, let P_{hit} denote the proportion of genes out of the genes in Gene Set $\mathcal{S}_k$ that are before the position i and P_{miss} denote the proportion of genes out of all genes but *not* included in Gene Set $\mathcal{S}_k$ that are also before position i, that is

$$P_{hit}(\mathcal{S}_k, i) = \sum_{\substack{g_j \in \mathcal{S}_k \\ j \le i}} \frac{1}{N_R} \tag{5.1}$$

where N_R represents the number of genes in Gene Set $\mathcal{S}_k$ and it sums over all g_j's such that $j \le i$ (i.e. all $1/N_R$ terms that appear before position i) and

$$P_{miss}(\mathcal{S}_k, i) = \sum_{\substack{g_j \notin \mathcal{S}_k \\ j \le i}} \frac{1}{N - N_R} \tag{5.2}$$

where $N - N_R$ represents the number of genes not in Gene Set S_k and it sums over all g_j's such that $j \leq i$ (i.e. all $1/(N - N_R)$ terms that appear before position i). Then the Enrichment Score (ES) for the k-th gene set is defined as shown in (5.3)

$$ES_k = \max_i \left[P_{hit}(S_k, i) - P_{miss}(S_k, i) \right] \tag{5.3}$$

over all position i in the list L.

5.5 A Weighted Version of Enrichment Score

As stated in the previous section, a high enrichment score could be attained by a gene set in which genes are found to appear more frequently than expected near the top of the list L, however it could also be attained if genes in a gene set are found to show up more often than expected around the middle of the list L. This undesired behavior is because the original definition of an Enrichment Score ignores the fact that the association scores within the top of the list L show *more* evidence that genes are really differentially expressed than those around the middle of the list L. Hence a reasonable modification for the original definition of Enrichment Score should include a weight factor such that it will reduce the magnitude of enrichment scores for gene sets that are enriched around the middle of the list L, while still be able to maintain or boost the magnitude of enrichment scores for those that are really enriched near the top of the list L. Subramanian *et al*. (2007) proposed an improved version of their original definition. Instead of using equal weights on the running-sum statistic P_{hit} (5.1), the modified formula takes into consideration the association scores of each gene according to the two phenotypes and introduce a weighting factor p to control the manner how the association scores influence the running-sum statistic P_{hit}.

By introducing the use of the association scores r_j's and the weighting factor p, Equation (5.1) can be revised as follows:

$$P_{hit}(\mathcal{S}_k, i) = \sum_{\substack{g_i \in \mathcal{S}_k \\ j \le i}} \frac{|r_j|^p}{N_R} \tag{5.4}$$

where $N_R = \sum_{g_i \in \mathcal{S}_k} |r_j|^p$. Since N_R has a new definition, it does not necessarily mean the number of genes in Gene Set $\mathcal{S}_k$ anymore, so we instead use N_H to denote the number of genes in Gene Set $\mathcal{S}_k$ in the new method. Thus, Equation (5.2) can be rewritten as follows:

$$P_{miss}(\mathcal{S}_k, i) = \sum_{\substack{g_i \notin \mathcal{S}_k \\ j \le i}} \frac{1}{N - N_H} \tag{5.5}$$

If we choose the weighting factor $p = 0$, all $|r_j|^p$ terms reduce to 1 and N_H is still the number of genes in Gene Set $\mathcal{S}_k$, thus we have the same results as given by the original Enrichment Score formula; if p is chosen to be 1, the formula gives a weight to each gene in Gene Set $\mathcal{S}_k$ according to the absolute value of their association scores r_j.

5.6 Significance Level of Enrichment Score

Once we have obtained the enrichment scores ES_k's for all gene sets in the first step, in the next step, we want to find an appropriate way to estimate the significance level of them. The enrichment score measures the maximum difference between P_{hit} and P_{miss} for one gene set and it is supposed to be close to zero if that gene set is not enriched. However, if some or all of genes in one

gene set have unusual values of the association scores r_j, we should expect the enrichment score ES for that gene set to be large in absolute magnitude.

To determine if a gene set is statistically significant, we need to find a null distribution under which we assume that there is no significant difference between the two phenotypes among the samples, thus all gene sets should behave in a similar way and for each gene set, its corresponding enrichment score should be close to zero. However it is reasonable to expect some random fluctuation of the enrichment scores around zero, more or less, even if none of the gene sets is really enriched.

Since it is not easy to find the null distribution of the Enrichment Score analytically, GSEA method uses an empirical phenotype-based permutation test to obtain a sampling distribution instead. Basically, we can permute the sample labels (column permutation) and follow exactly the same way as abovementioned to compute a new set of gene set enrichment scores for each gene set based on the newly permuted dataset, and then repeat this process for a number of times (say 1000 times), which finally generates a null distribution for the enrichment score. And then the empirical *p*-values can be calculated by

$$p_k^* = \Pr\left(S_k > s_k \mid \hat{\mathrm{F}}_{null}\right) = \frac{\#\left\{\hat{\mathrm{F}}_{null} > s_k\right\}}{\#\left\{\hat{\mathrm{F}}_{null}\right\}} \tag{5.6}$$

, where $\hat{F}_{null}$ is the null distribution based on permutations for the enrichment score of Gene Set k, and s_k is the observed enrichment score for the same gene set. This formula is to be used under the assumption that the observed enrichment score s_k will appear in the positive portion of the empirical null distribution $\hat{F}_{null}$. However, if s_k shows up in the negative portion of the empirical null distribution,

we shall change the inequality sign in the formula from " $>$ " to " $<$ " in order to obtain the correct p-value. A preset significance level α (usually set at 0.05) can be used and a gene set is declared to be significantly different from other gene sets with regard to the gene expression patterns over the two phenotypes of interest if the observed empirical p-value p_k^* is less than α.

5.7 Issues with a small number of permutations

As discussed in Section 5.6, Gene Set Enrichment Analysis (GSEA) usually requires the use of an empirical permutation test procedure to help estimate the statistical significance (often set at 0.05). One can perform either column permutations or row permutations based on the method being used, in order to obtain the null distribution of Enrichment Scores (ES) for gene sets. In either case, a complete set of all possible permutations is undoubtedly desired, if we want to compute the exact p-values for enrichment scores of each gene set.

However, the nature of microarray data and the computation of required intermediate statistics may prohibit us from running complete permutations in an affordable way. A microarray data set often contains tens of thousands of genes, which could be assigned to tens of thousands of gene sets too. Even though the number of arrays for microarray data is generally small, a complete column permutation (permutation of arrays) and the subsequent computation of statistics (either at the gene level or at the probe level) may halt your computer for hours, perhaps days or even weeks. Furthermore, as is usually huge in size, microarray data inevitably needs much more room in computer memory when computation is in progress. It will not be surprising to see computations be stopped unexpectedly due to insufficient memory. It could be even worse for row permutations (permutation of genes or probes).

To avoid being stuck in such a dilemma, one needs to ease on the accuracy of empirical *p*-values obtained through permutations, by reducing the number of permutations to a lower level. The estimator of a proportion is given in Equation (5.7),

$$\hat{p} = \frac{X}{m} \tag{5.7}$$

where X represents the number of positive observations. When the observations are independent, the estimator $\hat{p}$ follows a binomial distribution, hence approximately a normal distribution with the same mean and variance as the binomial distribution has. Then the approximate 95% confidence interval for a proportion estimated to be $\hat{p}$ from m permutations is

$$\hat{p} \pm 2\sqrt{\frac{\hat{p}(1-\hat{p})}{m}} \approx \hat{p} \pm 2\sqrt{\frac{\hat{p}}{m}}, \tag{5.8}$$

when $\hat{p}$ is small. If we are required to estimate a *p*-value to two decimal places, which means the maximum length of the confidence interval would be 0.01, then we must have

$$4\sqrt{\frac{\hat{p}}{m}} < 0.01. \tag{5.9}$$

Solving Equation (5.9) for m, we have $m > 16 \times 10000 \times \hat{p}$. If $\hat{p}$ is to be 0.01, which is the smallest possible value of two decimal places, we should have $m > 1600$; if $\hat{p}$ is 0.05, then we need to have $m > 8000$. Since we usually do not care about *p*-values smaller than 0.01, a good rule of thumb is to have between 1,000 and 10,000 permutations.

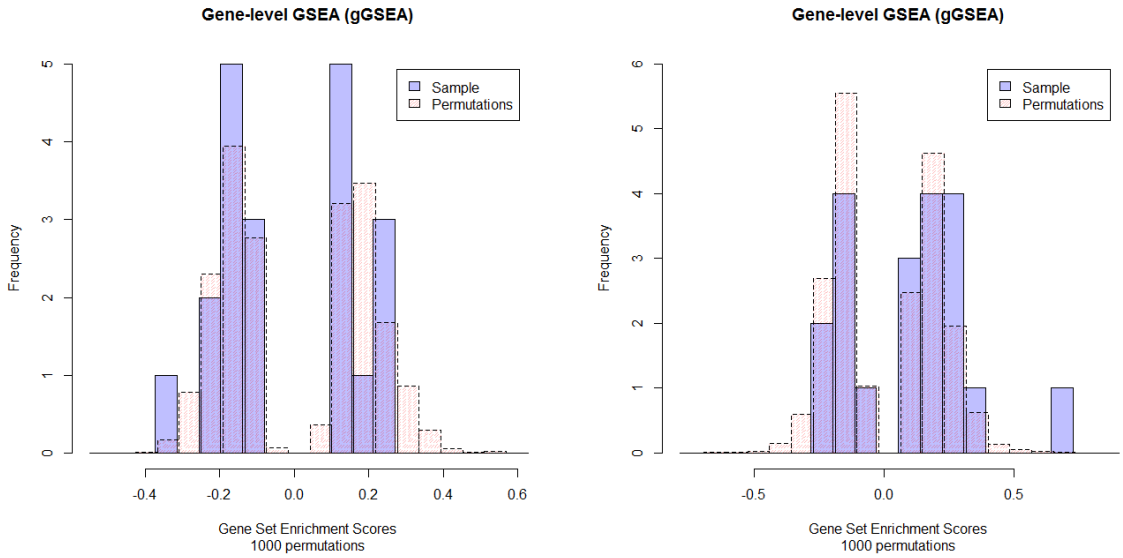
5.8 A Simulation Example

Here we give a simulation example to help understand the picture of the GSEA method. The simulation was performed in the following way. We generated a microarray probe-level data with 1000 genes and 12 samples. Each consecutive non-overlapping block of 50 genes is considered to be one gene set. Those 12 samples are divided into two groups, one as the control group containing 6 samples and the other as the treatment group containing the other 6 samples. Each gene (also known as probe set) consists of 11 probes according to the general Affymetrix GeneChip microarray structure. More detailed explanation of Affymetrix GeneChip microarray is given in Chapter 1.

Two simulations were done to show the picture of the GSEA method. For both of the two simulations, we first generated each data value x_{ijk} as $i.i.d.Normal(0,1)$, where i represents for probes and ranges from 1 to 11 in our example; j for arrays and ranges from 1 to 12; k for genes and ranges from 1 to 1000. Next, we kept the data intact for the first simulation, and for the second simulation a constant value of 0.6 was added to all probes in the first gene set for the treatment group, meanwhile all other data values remain unchanged. Then a gene expression summarization process using the median polish method was applied to obtain the gene-level data. Details about methods of preprocessing microarray probe-level data are given in Chapter 2. Finally, we obtained two microarray gene-level dataset with one 1000×12 matrix for each simulation respectively. For the first simulation, since all data are from the same normal distribution independently, there should be no average difference between the two groups (the control and treatment groups). In the meantime, for the second simulation, since we added a positive value to all probes in the first

gene set for the treatment group, it is reasonable to expect to see the first gene set shows higher average expressions for each gene in the treatment group and no average difference for other gene sets between the two groups.

Once we have the simulated microarray data ready, we can perform the GSEA procedure as described in Section 5.5 and carry out a number of permutations as described in Section 5.6 to obtain a null distribution of the gene set enrichment scores. In our example, we did 1000 permutations for each of the simulations. In Figure 6, the left panel shows the histogram (solid lines with blue shades) of the gene set enrichment scores for the 50 gene sets for the first simulation. It also shows the histogram (dashed lines with gray shades) of the empirical null distribution based on the permutations. Clearly, we can see no gene set stand out of the pattern of the null distribution, which implies that all gene sets may be expressed in a similar manner. Similarly, the right panel shows those histograms for the second simulation. One gene set stands out to the right side as expected with an ES score of 0.72. The results show that the GSEA method works reasonably well for microarray gene-level data.



Simulation 1
(no value added to treatment)

Simulation 2
(0.6 added to treatment)

Figure 6: A simulation example for the GSEA method on microarray gene-level data

a. left panel shows the histogram (solid lines with blue shades) of the gene set enrichment scores for the 50 gene sets for Simulation #1. No gene set stands out.

b. right panel shows the histogram (solid lines with blue shades) of the gene set enrichment scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out fairly clearly with an ES score of 1.

Chapter 6
Probe-Level GSEA

6.1 Introduction

In previous chapters, we introduced the basic structure of the microarray technology and several relevant analytical methods were discussed. Despite the differences among those statistical methodologies, all of them are currently carried out at the gene level, though the microarray technology actually gives the probe intensities rather than the gene intensities. As discussed in Chapter 2, one has to go through a preprocessing stage to obtain the gene intensities from the raw probe intensities. The preprocessing steps are important for microarray data analysis, because it can greatly enhance the quality of downstream analyses. For example, in the processing stage, the background correction step removes the nonspecific binding signals so that the true DNA expression level can be obtained; the normalization step eliminates the systematic effects so that the data can better describe the samples' own biological features, etc.

The last step in the preprocessing stage is the summarization step. Prior to any further analysis, the background corrected and normalized probe intensities are usually required to be summarized into a single measure of expression level for each gene (probe set). There are a variety of statistical algorithms being used in the summarization step. The advantage of summarizing the probe intensities into the gene intensities is that it greatly reduces the data size. For example, suppose there is a microarray dataset which contains 1,000 probe sets and each

probe set in that microarray dataset contains 11 probes, then for each array, the probe intensities would contain 11,000 data points. However, for gene intensities, it would only contain 1,000 data points. The data reduction is useful and preferred if the computing capability is limited because by reducing the size of the microarray dataset, it not only lowers the requirement on the computer's memory size, but on its computing speed as well. However the current rapid development of computer hardware as well as software has made fast computing easily accessible. The issues related to hardware bottleneck may have become unimportant and even nonexistent.

Despite the differences in summarization algorithms, the major drawback of summarization is that a substantial amount of probe level information is discarded. Many researches have been done on comparing the methods using probe level data and those conventional methods using gene level data. For example, Lemon *et al*. (2003) argued about this issue by comparing the coefficients of variation found at the probe level with the coefficients of variation found in the corresponding gene level expression, using the MAS5.0 and dChip expression algorithms, for the Affymetrix Latin Square dataset. It turns out that the gene level expression measures failed to reach the optimal efficiency predicted by sampling theory. This is one of the examples that show the current popular algorithms used in the summarization step may not be able to work perfectly as expected to capture all the crucial biological information held in each individual probe.

Without any doubt, efforts on improving the current algorithms and developing new methods for summarizing probe level measures have never been reduced. However, an alternative approach that directly takes advantage of

the probe level data could be another viable path to help researchers better understand and solve the problems in current bioinformatics research.

6.2 Extension of GSEA Method to Microarray Probe-level Data

In Chapter 5, we introduced the GSEA method, which is a knowledge-based approach for interpreting genome-wide profiles. The GSEA method currently is used at the gene level, which requires summarizing gene level expression measures from the probe level expression measures (after the background correction and the normalization steps). The first step of the GSEA method involves the calculation of an Enrichment Score (ES) for each pre-defined gene set, and the Enrichment Score depends on the values of a set of association scores (e.g. t-test, Wilcoxon rank sum test, etc.) that measures the difference of each gene's expression in the two phenotype for that gene set based on the gene level expression measures. If the gene level expression measures do not capture all of the information contained in the individual probes, then those association scores would become less powerful in detecting the difference level of genes using gene level intensities, which makes it difficult to believe that the subsequent steps in the GSEA method would work efficiently as is supposed to be.

6.2.1 GSEA at probe level

To overcome the problem of potentially losing information contained in the microarray data because of summarizing the probe level data into the gene level intensities, we may modify the GSEA method in a way such that it would allow us to directly take advantage of the probe level intensities. To achieve this, suppose we have a collection of pre-defined gene sets $\{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_K\}$, which group all N many probe sets (genes) in the microarray data into K many gene sets. Each

probe set contains N_g many probes, where g ranges from 1 through N and N_g may usually range from 11 to 20 in an Affymetrix GeneChip. The location of each probe can be represented by a pair of indexes (g, n_g). For example, the pair $(10,3)$ refers to the 3rd probe in the 10th probe set. Then for each probe across all arrays, we compute a list of association scores $\{r_{g,n_g}\}$ for comparing the two means of the two different phenotypic groups. Based on the definition of each gene set, the association score for each probe then can be linked to the corresponding gene sets. For example, if Gene $g, g \in [1, N]$ belongs to Gene Set $\mathcal{S}_k, k \in [1, K]$, then the association scores $\{r_{g,1}, r_{g,2}, \ldots r_{g,N_g}\}$ of all N_g probes in Gene g are included in a set of association scores whose corresponding probes (so that their corresponding genes) are also in the same gene set.

Once the association scores at probe level are obtained, similarly to the algorithm at gene level, we can produce a list L of probes based on their association scores and make the list ordered decreasingly with large association scores on the top of the list. Then for each gene set in the collection of pre-defined gene sets $\{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K\}$, the Enrichment Score ES_k for the kth gene set $\mathcal{S}_k$ would be the maximum difference between two proportions: the proportion of probes out of all probes in Gene Set $\mathcal{S}_k$ that are before some position i, and the proportion of probes out of all probes but not included in Gene Set $\mathcal{S}_k$ that are also before position i.

Similar to what is discussed in Section 5.5, we should note that a high enrichment score could be attained by a gene set in which probes of those genes in that gene set are found to appear more frequently than expected near the top of the list L, however it could also be attained if probes of those genes in a gene

set are found to show up more often than expected around the middle of the list L. Thus we should also modify the algorithm to introduce a weighting factor p to control the manner how the association scores influence the running-sum statistic (see Equation (6.1)).

After computing the Enrichment Scores for each gene set, the significance level of each gene set then can be obtained by performing a permutation test. This has been discussed in Section 5.6.

6.2.2 Mathematical Formulation for Probe Level Data

For a microarray probe level data set, suppose there are N many probe sets, and each probe set contains N_g many probes, hence the total number of probes in the data set would be $N_{probe} = \sum_{g=1}^{N} N_g$. If we assume all probe sets contain equal number of probes, then the total number of probes in the data set would be simply $N_{probe} = N \times N_g$. Let $L = \left\{ p_1, p_2, \ldots p_{N_{probe}} \right\}$ be the list of all probes, which is decreasingly ordered based on the association scores $\left\{ r_i = r_{g,n_g}, g = 1,2,\ldots,N, n_g = 1,2,\ldots N_g, i = 1,2,\ldots,N_{probe} \right\}$ between the two phenotypes, and a collection of pre-defined gene sets $\mathcal{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_K\}$, let P_{hit} denote the proportion of probes out of all probes in Gene Set $\mathcal{S}_k$ that are before some position i, and P_{miss} denote the proportion of probes out of all probes but not included in Gene Set $\mathcal{S}_k$ that are also before position i, that is

$$P_{hit}\left(\mathcal{S}_k, i\right) = \sum_{\substack{p_i \in \mathcal{S}_k \\ j \le i}} \frac{\left| r_j \right|^p}{N_R} \tag{6.1}$$

where $N_R = \sum_{p_i \in \mathcal{S}_k} |r_j|^p$. When $p = 0$, N_R represents the number of probes in Gene Set $\mathcal{S}_k$ and $P_{hit}(\mathcal{S}_k, i)$ sums over all p_j's such that $j \le i$ (i.e. all $1/N_R$ terms that appear before position i). For the same reason shown in Section 5.5, when $p > 0$, N_R does not necessarily mean the number of genes in Gene Set $\mathcal{S}_k$ anymore, so we use N_H instead to denote the number of genes in Gene Set $\mathcal{S}_k$ in the new method.

$$P_{miss}(\mathcal{S}_k, i) = \sum_{\substack{p_j \notin \mathcal{S}_k \\ j \le i}} \frac{1}{N - N_H} \tag{6.2}$$

where $N - N_H$ represents the number of probes not in Gene Set $\mathcal{S}_k$ and it sums over all p_j's such that $j \le i$ (i.e. all $1/(N - N_H)$ terms that appear before position i). Then the Enrichment Score (ES) for the k-th gene set is defined as

$$ES_k = \max_i \left[P_{hit}(\mathcal{S}_k, i) - P_{miss}(\mathcal{S}_k, i) \right] \tag{6.3}$$

over all position i in the list L.

6.2.3 A Simulation Example

To illustrate the picture of the GSEA method at the probe level, we can perform a simulation in a way exactly the same as what we did in Section 5.8, except that no summarization is needed for the probe-level GSEA. Details about how the probe-level intensities were generated can be found in Section 5.8. As a summary, the generated data set contains 1000 probe sets and 12 arrays. Each probe set contains 11 probes. Then the data set would be a 11000×12 numerical matrix. Each consecutive non-overlapping block of 50 probe sets (genes) is considered to be one gene set. Those 12 arrays are divided into two groups, one as the control

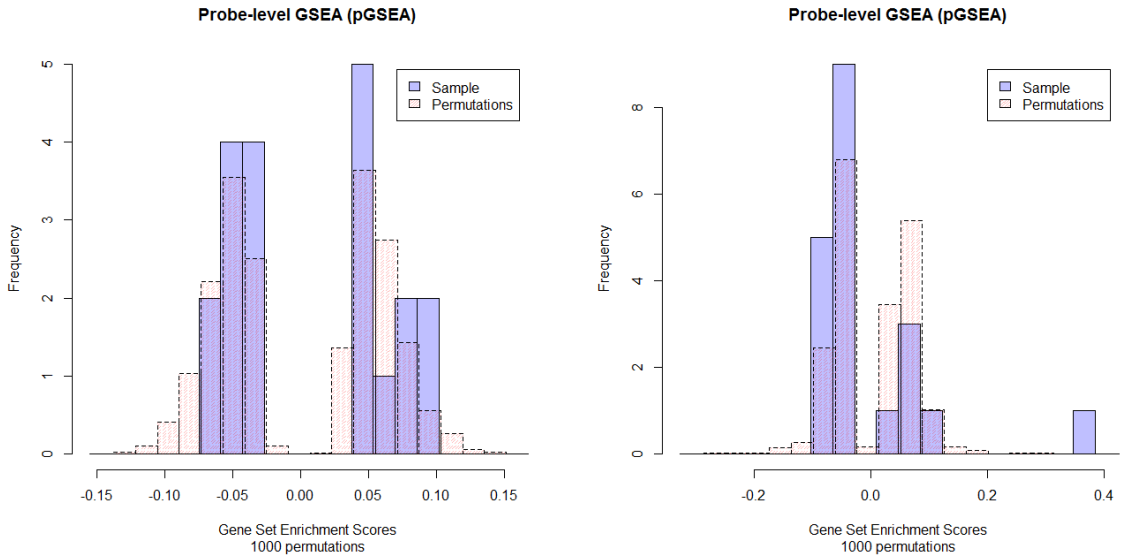
group containing 6 arrays and the other as the treatment group containing the other 6 arrays.

We still use two simulations to show the pictures of the probe-level GSEA. We kept the data intact for the first simulation, and for the second simulation a constant value of 0.6 was added to all probes in the first gene set for the treatment group, meanwhile all other data values remain unchanged. We expect to see no average difference between the control and the treatment groups for the first simulation because the data in the first simulation was independently generated from the same normal distribution. In the meantime, we expect to see the first gene set show higher average expressions for each probe in the treatment group and no average difference for other gene sets between the two groups.

After generating the probe-level intensities, we can follow the procedure outlined in Section 6.2.1 to perform the probe-level GSEA. In the example, 1000 permutations for each simulation were done to obtain a null distribution of the gene set enrichment scores.

The pictures of the observed GSEA enrichment scores (ES) and the null distribution for the two simulations are shown in Figure 7. In Figure 7, the left panel shows the histogram (solid lines with blue shades) of the gene set enrichment scores for the 50 gene sets from the first simulation. It also shows the histogram (dashed lines with gray shades) of the empirical null distribution based on the 1000 permutations. Obviously, no gene set stands out of the pattern of the null distribution and it indicates that all gene sets may be expressed in a similar manner. The right panel shows the two histograms from the second

simulation. One gene set stands out significantly to the right side as expected with an ES score close to 0.38.



Simulation 1
(no value added to treatment)

Simulation 2
(0.6 added to treatment)

Figure 7: A simulation example for the GSEA method on microarray gene-level data

a. left panel shows the histogram (solid lines with blue shades) of the gene set enrichment scores for the 50 gene sets for Simulation #1. No gene set stands out.
b. right panel shows the histogram (solid lines with blue shades) of the gene set enrichment scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out significantly with an ES score close to 1.

6.2.4 Comparison with the gene-level GSEA

We can compare the results from the probe-level GSEA (pGSEA) with the results from the gene-level GSEA (gGSEA). When there is no difference present between the control group and the treatment group, both pGSEA and gGSEA works reasonably well. However, if a difference between the two groups becomes present, the histogram from the pGSEA method looks much stronger than the histogram from the gGSEA method.

To compare testing methods, we usually look at the power of each testing method. However, an analytical approach to calculating the power of gGSEA or pGSEA methods is difficult to acquire. An alternative approach using simulations can be used instead to obtain the power of those methods. Basically, the power of a statistical test is the probability that the test will reject the null hypothesis when the alternative hypothesis is true. For the gGSEA and pGSEA methods, the null hypothesis would be that the interested gene set behaves similarly as other gene sets do, which implies no significant difference should be found across the treatment group and the control group for that gene set, and the alternative hypothesis would be the opposite. Therefore, we can create a series of simple alternative hypotheses such that in each of those alternative hypotheses, one value will be added to the probe intensities in the first gene set for the treatment group, then a number of simulations can be carried out for each simple alternative. The rejection rate, given in Equation (6.4) where d represents the difference value that would be added to the probes in the treatment group, can be used as the empirical power for the corresponding method under the particular alternative.

$$\beta_d = \frac{\#\{\text{rejections}\}}{\#\{\text{total simulations}\}} \tag{6.4}$$

Steps for calculating powers for gGSEA and pGSEA:

1. Create a list of simple alternative hypotheses (i.e. a list of difference values that would be added to the treatment group).
2. For each difference value, do a number of simulations based on the gGSEA and pGSEA methods.
3. Compute the rejection rate β_d at each difference value.

Figure 8 shows an example of the powers of the gGSEA and the pGSEA methods in the same graph, which is based on the following settings:

- Each data value x_{ijk} was generated as $i.i.d.Normal(0,1)$, where i represents for probes and ranges from 1 to 11 in the example; j for arrays and ranges from 1 to 12; k for genes and ranges from 1 to 1000.
- Difference values starting from 0 through 0.6 with an interval of 0.05 in length (i.e. 0, 0.05, 0.10, 0.15, … , 0.60) were added for each run respectively to all probes of the first gene set under the treatment group.
- 1000 simulations were done for each run to compute the rejection rate β_d .

Apparently, the picture in Figure 8 shows that the power of the pGSEA method is unanimously higher than the power of the gGSEA method at all times, which is evidence that the pGSEA method works better than the gGSEA method in detecting gene sets that are differentially expressed across the control and the treatment groups.

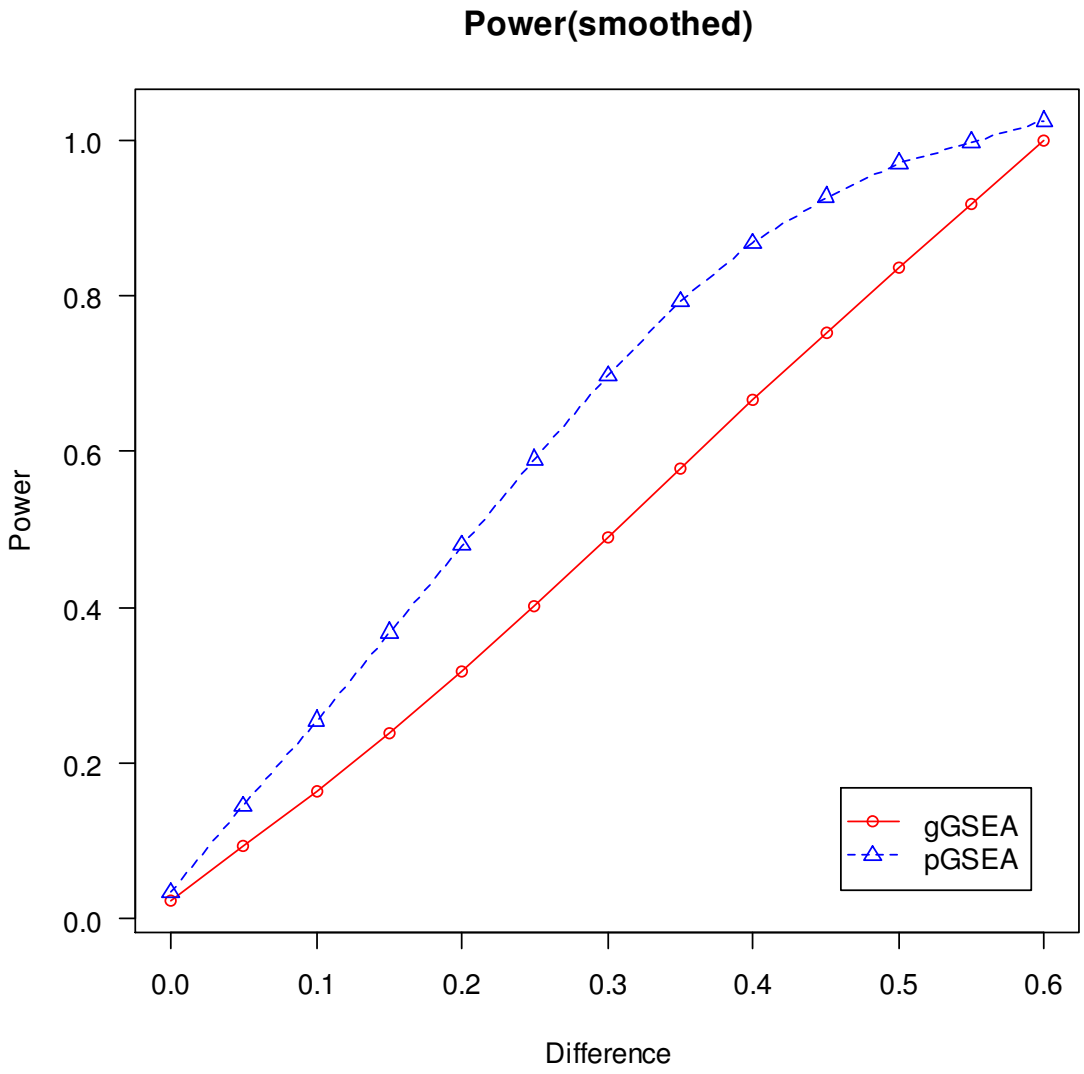


Figure 8: A simulation example for comparing the power of gGSEA and the power of pGSEA

a. The red solid line represents the powers of the gGSEA method, with circles representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

b. The blue dashed line represents the powers of the pGSEA method, with triangles representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

Table 3 Empirical Power (GSEA Methods)

Distance	gGSEA	pGSEA
0	0.052	0.066667
0.05	0.07	0.123333
0.1	0.136667	0.21
0.15	0.266667	0.426667
0.2	0.293333	0.463333
0.25	0.313333	0.493333
0.3	0.466667	0.753333
0.35	0.666667	0.893333
0.4	0.676667	0.903333
0.45	0.733333	0.973333
0.5	0.876667	0.996667
0.55	0.93	1
0.6	0.963333	1

Chapter 7
Enrichment Analysis with Robust M-estimators (EAME)

7.1 Why Use Robust Methods

Both the traditional gene-level GSEA method and the modified probe-level GSEA method employ the sample mean $\bar{X}$ and the sample standard deviation s as a vehicle to compare the difference between the treatment group and the control group, either based on the gene-level expression intensities after summarizing from the probe-level expression intensities, or directly using the probe-level expression intensities. These statistics are optimal in many sense if the underlying distribution of the sample, or equivalently the error term ε, follows a normal distribution. If such assumption is not valid, the result would become unreliable. Furthermore, due to the complexity of DNA microarray experiment, many levels of variation can be introduced at different stages of the experiments, and outliers are basically impossible to avoid.

For example, the `affydata` package (Gautier, 2011) from Bioconductor (http://www.bioconductor.org/) includes an example data set containing part of the data from a *Dilution* experiment. The data in Dilution is a small sample of probe sets from 2 sets of duplicate arrays hybridized with different concentrations of the same RNA. After performing the background correction and the normalization steps as well as log-transformation, we used the `rlm()`

function in the `MASS` package to fit a linear model (Equation (7.1)) by robust regression using an M-estimator on each probe set across arrays, and obtain the residual matrices.

$$y_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij} \tag{7.1}$$

Since traditionally y_{ij}'s, hence ε_{ij}'s, are assumed to follow a normal distribution, we should expect to see the *p*-values from a normality test on the residual matrices for each probe set follow roughly a uniform distribution. That is, if we are going to reject the null hypothesis in the normality test at the $\alpha = 0.05$ level, then we should expect to see the proportion of those *p*-values that are less than 0.05 is also around 0.05, since it is possible by chance to reject 5% of all of the null hypotheses in the normality test. In our example, we used the Shapiro–Wilk test

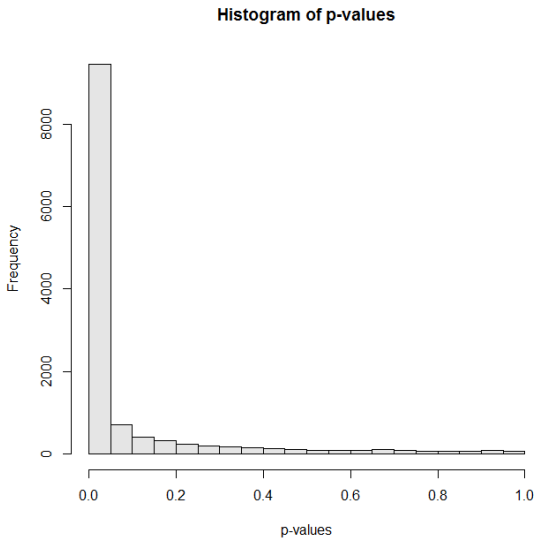


Figure 9: Histogram of p-values from the Shapiro–Wilk test for the residual matrix of each probe set

a. The histogram obviously shows a non-uniform pattern.
b. It also implies a large portion of the null hypotheses are rejected.

(Shapiro and Wilk, 1965) to test the normality assumption. Figure 9 shows the histogram of *p*-values from the Shapiro-Wilk test for the residual matrix of each probe set. It clearly illustrates a non-uniform pattern, which is against the traditional assumption that the error terms follow a normal distribution. Moreover, it also illustrates a large portion of the null hypotheses are rejected. As a matter of fact, the proportion of *p*-values that are less than 0.05 is 0.749, far from 0.05. It could be evidence that the normality assumption may not necessarily be valid for this example.

7.2 Robust Method - Median

To overcome the issues discussed in Section 7.1, we can choose *resistant* estimators, which does not require the assumption of a normal distribution and the influence of outliers is limited. The most reasonable resistant estimators of the mean μ and the standard deviation σ are the *median* $\tilde{X}$ (Equation (7.2)) and the *median absolute deviation from the median* (MAD) (Equation (7.3)).

$$\tilde{X} = \text{median}\{X_1, X_2, \ldots, X_n\} \tag{7.2}$$

$$\text{MAD} = \underset{i}{\text{median}}\left\{\left|X_i - \tilde{X}\right|\right\} \tag{7.3}$$

Based on the median and the MAD, we can construct a robust statistic that measures the difference between two samples. Equation (7.4) gives the formula for computing the robust statistic.

$$t^* = \frac{\tilde{X}_1 - \tilde{X}_2}{\frac{1}{2}\left(\text{MAD}_1 + \text{MAD}_2\right)} \tag{7.4}$$

7.2.1 A Robust Gene Set Score

Here we propose a new method for detecting differentially expressed gene sets based on the robust statistic given in Equation (7.4). After obtaining the robust statistic t^*'s for each probe in the first step, secondly we can compute the mean of all t^*'s within the same probe set for every probe set. Then thirdly we can take average over the absolute value of all the means computed for each probe set from the second step that are in the same gene set. This would in fact produce a gene set score GS_k for each gene set similar to the gene set enrichment score used in the GSEA method. Although taking median over all t^*'s within the same probe set is seemingly preferred, a numerical average in the second step as well as in the third step should be good enough because any extreme situation has been taken care of within the first step when the robust statistic t^* is computed using the medians and the MAD's. Another reason for using the mean instead of the median at these two steps is that it may underestimate the level of differential expression because it does not take into consideration of the magnitude of the actual values obtained from the first step or the second step. For example, if in the same probe set, 5 probes show big differences across the treatment and the control groups, and the other 6 probes do not show any differences (assuming 11 probes in the probe set), we would expect to see no evidence that the gene is differentially expressed if we are about to use the median. We will see the same situation in the third step because there will be no sign that the corresponding gene set is differentially expressed if, for example, only 24 out of 50 probe sets in one gene set are actually differentially expressed, and the other 26 probe sets are not. This is definitely not a desired phenomenon.

The steps to compute the robust gene set score are given as follows:

i. Compute the robust statistic t^*'s for each probe (Equation (7.4)).

ii. Compute the mean of all t^*'s within the same probe set for every probe set.

iii. For each gene set, compute the mean over the absolute value of all the means computed for each probe set in that gene set from the second step.

7.2.2 Mathematical Formulation

For a microarray probe level data set, suppose there are N many probe sets, and each probe set contains N_g many probes, hence the total number of probes in the data set would be $N_{probe} = \sum_{g=1}^{N} N_g$. Let t^*_{ijk} be the robust statistic defined as in Equation (7.4) for comparing probes across the treatment and the control groups, where i represents for probe sets which range from 1 through N ; j represents for probes in a probe set which range from 1 through N_g ; k represents for gene sets which range from 1 through K . The robust gene set score (GS) is then defined as in Equation (7.5):

$$\begin{aligned} GS_k &= \underset{i}{\text{mean}} \left\{ \left| \underset{j}{\text{mean}} \left(t^*_{ijk} \right) \right| \right\} \\ &= \underset{i}{\text{mean}} \left\{ \left| \underset{j}{\text{mean}} \left(\frac{\tilde{X}_{ijk,1} - \tilde{X}_{ijk,2}}{\frac{1}{2}\left(\text{MAD}_{ijk,1} + \text{MAD}_{ijk,2} \right)} \right) \right| \right\}, \end{aligned} \tag{7.5}$$

where $\tilde{X}_{ijk,1}$ and $\tilde{X}_{ijk,2}$ are the medians of probes, which correspond to the ith probe set (gene) and jth probe in the kth gene set, for the treatment and the control groups respectively.

7.2.3 Significance Level

Once the robust gene score GS_k's for all gene sets are obtained, we will use a permutation test, which permutes rows of the probe-level expression intensities, to estimate the significance level. Basically, we will permute the probe labels (row permutation) and follow the exactly the same way as discussed in Section 7.2.1 and 7.2.2 to compute a set of gene set scores for each gene set based on the newly permutated dataset, and then repeat this process for a number of times (say 1000 times), which finally generates a null distribution for the gene set scores. The null hypothesis under the row permutations is that the gene set $\mathcal{S}_k$ is chosen by random selection of $\sum_{g\in\mathcal{S}_k} N_g$ probes from the full set of $N_{probe} = \sum_{g=1}^{N} N_g$ probes. And then the empirical p-values can be calculated using Equation (5.6). A preset significance level α (usually set at 0.05) can be used and a gene set is declared to be significantly differentially expressed with regard to the gene expression patterns over the two phenotypes of interest if the observed empirical p-value p_k^* is less than α.

7.3 Huber M-estimator

In previous sections, we introduced a new method for detecting differentially expressed gene sets based on the robust statistic given in Equation (7.4), which is based on the medians and the MAD's across the treatment and the control groups.

As a matter of fact, median belongs to the family of *Huber M-estimators*. We have introduced Huber M-estimator in Chapter 4. Here we give a brief review. M-estimators are a generalization of *maximum likelihood estimators*

(MLE's). For MLE's, we try to find an estimator that could maximize the likelihood function (7.6),

$$\prod_{i=1}^{n} f\left(x_i \mid \theta\right) \tag{7.6}$$

or equivalently, minimize (7.7).

$$\sum_{i=1}^{0} -\log f\left(x_i \mid \theta\right) \tag{7.7}$$

Huber (1964) proposed to generalize this by minimizing an objective function which utilizes some function $\rho(x)$ and sums over the sample (7.8).

$$\sum_{i=1}^{n} \rho\left(x_i \mid \theta\right) \tag{7.8}$$

Thus minimizing $\sum_{i=1}^{n} \rho\left(x_i \mid \theta\right)$ can usually be done by differentiating ρ and solving Equation (7.9)

$$\sum_{i=1}^{n} \psi\left(x_i \mid \theta\right) = 0 \tag{7.9}$$

where $\psi\left(x \mid \theta\right) = \frac{\partial}{\partial \theta} \rho\left(x \mid \theta\right)$.

The most familiar M-estimator is the sample mean, which is the least squares estimate of location. The ρ-function for this case is the square of residual (7.10)

$$\rho\left(x \mid \theta\right) = \left(x - \theta\right)^2 \tag{7.10}$$

Thus the ψ-function can be obtained by differentiating (7.10) with respect to θ: $-2(x-\theta)$. By dropping the leading constant 2 and summing over all sample, we can find the estimator by minimizing (7.11),

$$\sum_{i=1}^{n}(x_i-\theta)=0 \tag{7.11}$$

in which $\hat{\theta}=\bar{x}=\frac{\sum_{i=1}^{n}x_i}{n}$, the sample mean, turns out to solve this equation.

Another example of M-estimator is the sample median. For this case, the ρ-function is the absolute value of the residual (7.12),

$$\rho(x\,|\,\theta)=|x-\theta| \tag{7.12}$$

and the corresponding ψ-function is (7.13).

$$\psi(x\,|\,\theta)=\operatorname{sgn}(x-\theta) \tag{7.13}$$

The family of Huber M-estimators uses the following ρ- and ψ- functions given in (7.14) and (7.15) respectively.

$$\rho_k(x\,|\,\theta)=\begin{cases}\frac{1}{2}(x-\theta)^2 & \text{if} \quad |x-\theta|\le k\\ k|x-\theta|-\frac{1}{2}k^2 & \text{if} \quad |x-\theta|>k\end{cases} \tag{7.14}$$

$$\psi_k(x\,|\,\theta)=\begin{cases}x-\theta & \text{if} \quad |x-\theta|\le k\\ k\operatorname{sgn}(x-\theta) & \text{if} \quad |x-\theta|>k\end{cases} \tag{7.15}$$

The sample mean and the sample median are special cases of Huber M-estimators that are corresponding to the limit cases $k\to\infty$ and $k\to 0$ (Maronna *et al.*, 2006).

The value k for the Huber estimators is called a *tuning constant*. There is a trade-off between robustness and efficiency. Smaller values of k produce more resistance to outliers, but at the expense of lower efficiency when errors are normally distributed. Maronna *et al*. (2006) gave an example to illustrate how different values of k can affect the asymptotic variances of Huber M-estimate for a *contaminated normal distribution* F (7.16)

$$F = (1-\varepsilon)G + \varepsilon H \tag{7.16}$$

with $\varepsilon = 0$, 0.05 and 0.10, where $G \sim N(0,1)$ and $H \sim N(0,10)$. It turns out when $k = 1.4$, the variance of the M-estimator at the normal is only 4.7% larger than that of the sample mean and much smaller than that of the sample median. And it clearly shows a smaller value than that of the sample mean and the sample median for the contaminated normal distributions with $\varepsilon = 0.05$ and 0.10.

Therefore, it is reasonable to consider Huber M-estimators as an alternative to the conventional sample mean and sample standard deviation used in Equation (7.5).

One approach is to use Huber estimators instead of taking sample means at the second and third steps (see page 72) to compute the robust gene set score. Then Equation (7.5) can be rewritten as

$$\begin{aligned} GS_k &= \underset{i}{\text{huber}_{\text{L}}} \left\{ \left| \underset{j}{\text{huber}_{\text{L}}} \left(t^*_{ijk} \right) \right| \right\} \\ &= \underset{i}{\text{huber}_{\text{L}}} \left\{ \left| \underset{j}{\text{huber}_{\text{L}}} \left(\frac{\tilde{X}_{ijk,1} - \tilde{X}_{ijk,2}}{\frac{1}{2}\left(\text{MAD}_{ijk,1} + \text{MAD}_{ijk,2} \right)} \right) \right| \right\} \end{aligned} \tag{7.17}$$

, where huber_{L} means the Huber M-estimator of location.

The second approach is to use Huber estimators for the location and the scale in Equation (7.4) instead of the median and the MAD. Then the gene set score (Equation (7.5)) can be rewritten as

$$\begin{aligned} GS_k &= \underset{i}{\text{mean}}\left\{\left|\underset{j}{\text{mean}}\left(t_{ijk}^{H}\right)\right|\right\} \\ &= \underset{i}{\text{mean}}\left\{\left|\underset{j}{\text{mean}}\left(\frac{\tilde{X}_{ijk,1}^{H}-\tilde{X}_{ijk,2}^{H}}{\frac{1}{2}\left(\hat{\sigma}_{ijk,1}^{H}+\hat{\sigma}_{ijk,2}^{H}\right)}\right)\right|\right\} \end{aligned} \tag{7.18}$$

, where $\tilde{X}^{H}$ and $\hat{\sigma}^{H}$ are the Huber location and scale estimators.

The third approach is to replace all the conventional estimators (mean, median and MAD) in Equation (7.5) with Huber M-estimators of location and scale. The revised equation is given as follows:

$$\begin{aligned} GS_k &= \underset{i}{\text{huber}_{\text{L}}}\left\{\left|\underset{j}{\text{huber}_{\text{L}}}\left(t_{ijk}^{H}\right)\right|\right\} \\ &= \underset{i}{\text{huber}_{\text{L}}}\left\{\left|\underset{j}{\text{huber}_{\text{L}}}\left(\frac{\tilde{X}_{ijk,1}^{H}-\tilde{X}_{ijk,2}^{H}}{\frac{1}{2}\left(\hat{\sigma}_{ijk,1}^{H}+\hat{\sigma}_{ijk,2}^{H}\right)}\right)\right|\right\} \end{aligned} \tag{7.19}$$

Since the process described in this chapter utilizes the idea of enrichment analysis and takes advantage of robust M-estimators, we will call it the method of *Enrichment Analysis with M-estimators* (EAME). Table 4 shows a summary of all above-mentioned EAME methods.

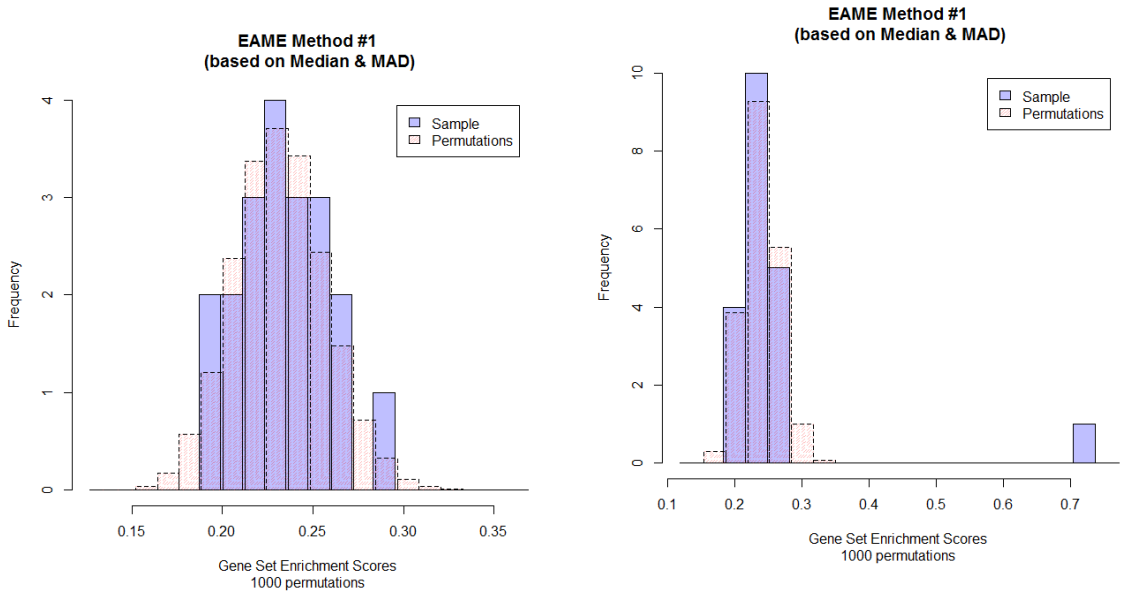
Table 4 Summary of EAME Methods

Method	Description of each Step	Formula
1	i. Use median and MAD ii. Use sample mean iii. Use sample mean	$GS_k = \underset{i}{\text{mean}}\left\{\left\|\underset{j}{\text{mean}}\left(t_{ijk}^{*}\right)\right\|\right\}$ $= \underset{i}{\text{mean}}\left\{\left\|\underset{j}{\text{mean}}\left(\frac{\tilde{X}_{ijk,1}-\tilde{X}_{ijk,2}}{\frac{1}{2}\left(\text{MAD}_{ijk,1}+\text{MAD}_{ijk,2}\right)}\right)\right\|\right\}$
2 (as a comparison to Method 1)	i. Use median and MAD ii. Use sample median iii. Use sample median	$GS_k = \underset{i}{\text{median}}\left\{\left\|\underset{j}{\text{median}}\left(t_{ijk}^{*}\right)\right\|\right\}$ $= \underset{i}{\text{median}}\left\{\left\|\underset{j}{\text{median}}\left(\frac{\tilde{X}_{ijk,1}-\tilde{X}_{ijk,2}}{\frac{1}{2}\left(\text{MAD}_{ijk,1}+\text{MAD}_{ijk,2}\right)}\right)\right\|\right\}$
3	i. Use median and MAD ii. Use Huber M-estimator of location iii. Use Huber M-estimator of location	$GS_k = \underset{i}{\text{huber}_{\text{L}}}\left\{\left\|\underset{j}{\text{huber}_{\text{L}}}\left(t_{ijk}^{*}\right)\right\|\right\}$ $= \underset{i}{\text{huber}_{\text{L}}}\left\{\left\|\underset{j}{\text{huber}_{\text{L}}}\left(\frac{\tilde{X}_{ijk,1}-\tilde{X}_{ijk,2}}{\frac{1}{2}\left(\text{MAD}_{ijk,1}+\text{MAD}_{ijk,2}\right)}\right)\right\|\right\}$
4	i. Use Huber M-estimators ii. Use sample mean iii. Use sample mean	$GS_k = \underset{i}{\text{mean}}\left\{\left\|\underset{j}{\text{mean}}\left(t_{ijk}^{H}\right)\right\|\right\}$ $= \underset{i}{\text{mean}}\left\{\left\|\underset{j}{\text{mean}}\left(\frac{\tilde{X}_{ijk,1}^{H}-\tilde{X}_{ijk,2}^{H}}{\frac{1}{2}\left(\hat{\sigma}_{ijk,1}^{H}+\hat{\sigma}_{ijk,2}^{H}\right)}\right)\right\|\right\}$
5	i. Use Huber M-estimators ii. Use Huber M-estimator of location iii. Use Huber M-estimator of location	$GS_k = \underset{i}{\text{huber}_{\text{L}}}\left\{\left\|\underset{j}{\text{huber}_{\text{L}}}\left(t_{ijk}^{H}\right)\right\|\right\}$ $= \underset{i}{\text{huber}_{\text{L}}}\left\{\left\|\underset{j}{\text{huber}_{\text{L}}}\left(\frac{\tilde{X}_{ijk,1}^{H}-\tilde{X}_{ijk,2}^{H}}{\frac{1}{2}\left(\hat{\sigma}_{ijk,1}^{H}+\hat{\sigma}_{ijk,2}^{H}\right)}\right)\right\|\right\}$

7.4 A Simulation Example

Here we give a simulation example to show the picture of the robust method with row permutations. With the same initial settings as used in Chapter 5 and Chapter 6, the generated data set contains 1000 probe sets and 12 arrays. Each probe set contains 11 probes. Each consecutive non-overlapping block of 50 probe sets (genes) is considered to be one gene set. Those 12 arrays are divided into two groups, one as the control group containing 6 arrays and the other as the treatment group containing the other 6 arrays.

Two simulations were carried out to illustrate how well the robust methods work. We kept the data intact for the first simulation, and for the second simulation a constant value of 0.6 was added to all probes in the first gene set for the treatment group, meanwhile all other data values remain unchanged. There should be no average difference between the control and the treatment groups for the first simulation because the data in the first simulation was independently generated from the same normal distribution. Meanwhile for the second simulation, we expect to see the first gene set show higher average expressions for each probe in the treatment group and no average difference for other gene sets between the two groups. 1000 permutations for each simulation were done to obtain a null distribution of the gene set scores.



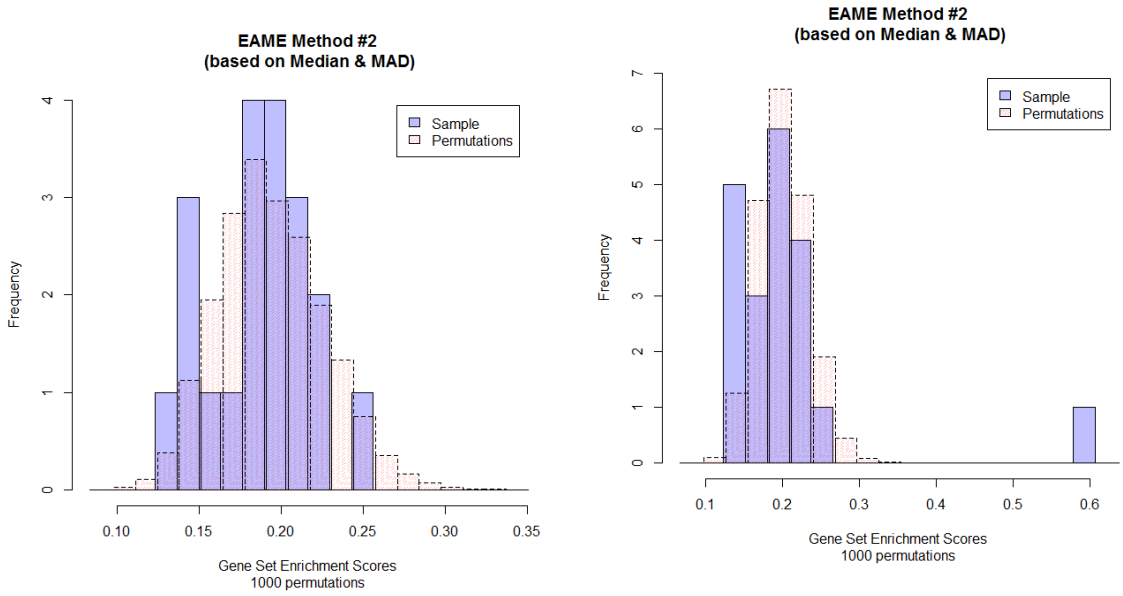
Simulation 1
(no value added to treatment)

Simulation 2
(0.6 added to treatment)

Figure 10: A simulation example for the EAME Method 1 based on Median and MAD (probe-level data)

a. left panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #1. No gene set stands out.

b. right panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out significantly with a GS score around 0.8.



Simulation 1
(no value added to treatment)

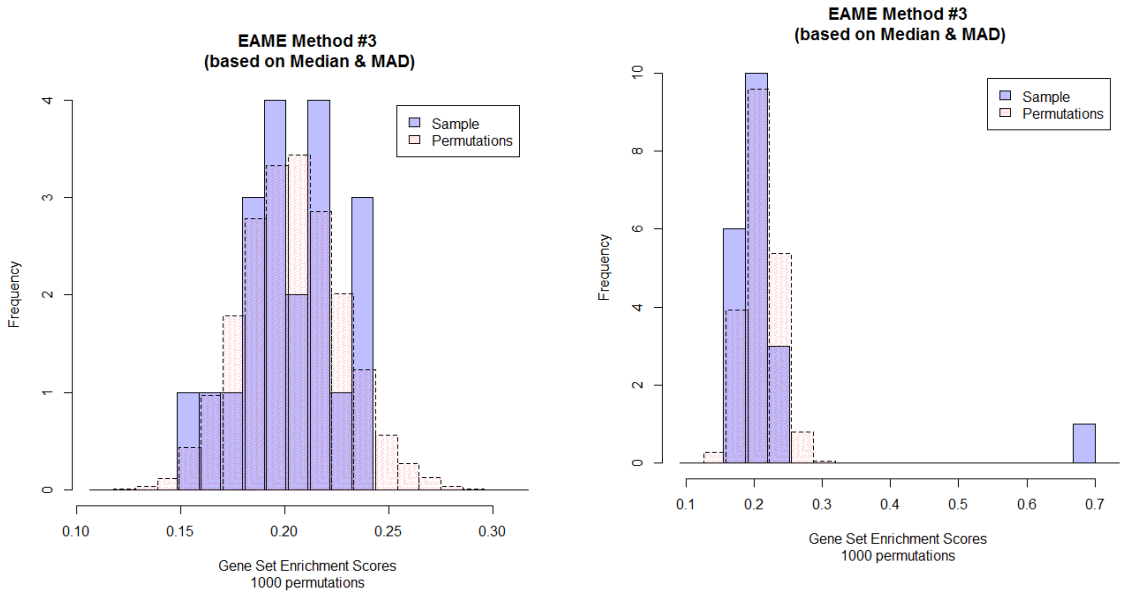
Simulation 2
(0.6 added to treatment)

Figure 11: A simulation example for the EAME Method 2 based on Median and MAD (probe-level data)

a. left panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #1. No gene set stands out.
b. right panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out significantly with a GS score around 0.6.

The pictures of the observed gene set scores (GS) and the null distribution for the two simulations based on Method 1 and 2 (using the median and the MAD (Equation (7.18))) are shown in Figure 10 and Figure 12. In Figure 10, the left panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets from the first simulation. It also shows the histogram (dashed lines with gray shades) of the empirical null distribution based on the 1000 permutations. Obviously, no gene set stands out of the pattern of the null distribution and it indicates that all gene sets may be expressed in a similar manner. The right panel shows the two histograms from the second simulation. One gene set stands out significantly to the right side as expected with a GS score

around 0.8. We have similar histograms shown in Figure 12 for Method 2. Although the GS score for the first gene set in Method 2 stands out clearly, it is smaller than the GS score obtained from Method 1.



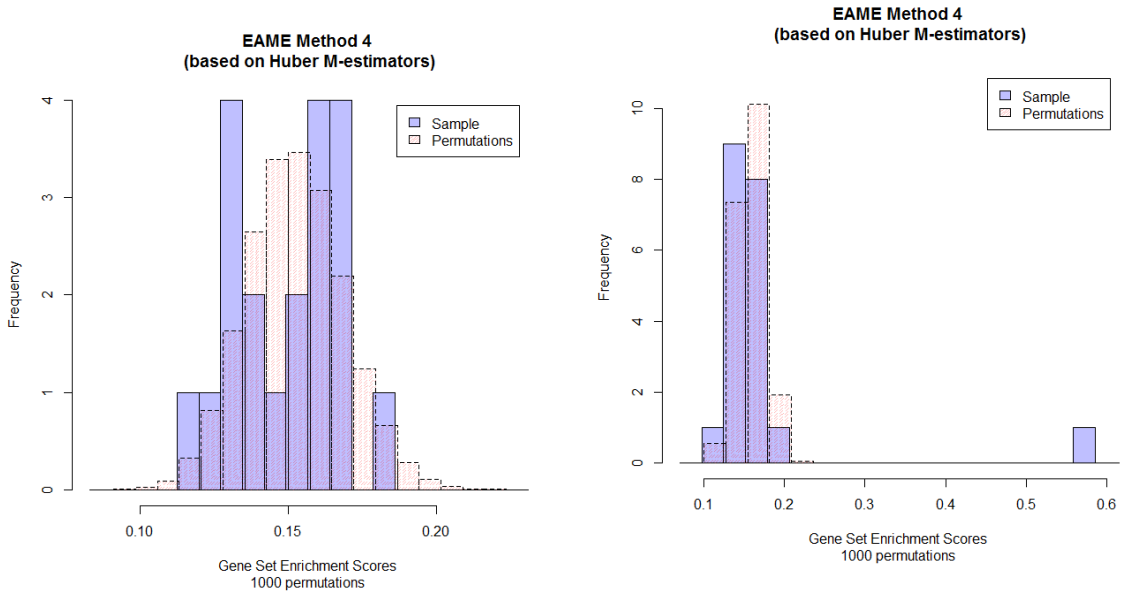
Simulation 1
(no value added to treatment)

Simulation 2
(0.6 added to treatment)

Figure 12: A simulation example for the EAME Method 3 based on Median and MAD (probe-level data)

a. left panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #1. No gene set stands out.
b. right panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out significantly with a GS score around 0.7.

Figure 12 shows the histograms for Method 3, which is based on the median and the MAD and uses Huber M-estimator of location at the second and third steps. As expected, there is no gene set standing out when no value was added to the treatment group, and one gene set stands out with a GS score around 0.7 when a value was added to the treatment group.



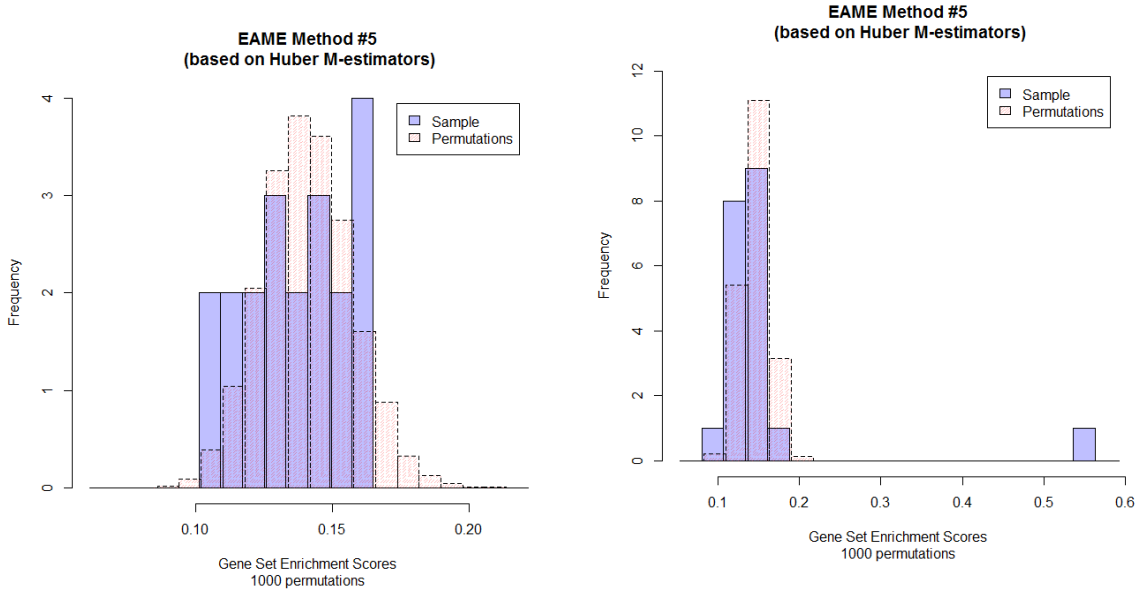
Simulation 1
(no value added to treatment)

Simulation 2
(0.6 added to treatment)

Figure 13: A simulation example for the EAME Method 4 based on Huber M-estimators (probe-level data)

a. left panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #1. No gene set stands out.

b. right panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out significantly with a GS score around 0.6.



Simulation 1
(no value added to treatment)

Simulation 2
(0.6 added to treatment)

Figure 14: A simulation example for the EAME Method 5 based on Huber M-estimators (probe-level data)

a. left panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #1. No gene set stands out.
b. right panel shows the histogram (solid lines with blue shades) of the gene set scores for the 50 gene sets for Simulation #2. Gene Set #1 stands out significantly with a GS score around 0.6.

The histograms of the observed gene set scores (GS) and the null distribution for the two simulations based on the Huber M-estimators are given in Figure 13 and Figure 14. Figure 13 shows the histograms for Method 4, which takes average of t_{ijk}^{H}'s over probes and probe sets at the second and third steps. Figure 14 shows the histograms for Method 5, which uses Huber M-estimator of location at the second and third steps. For both of the two methods, no gene set stands out of the pattern of the null distribution for Simulation #1, and it indicates that all gene sets may be expressed in a similar manner. For Simulation

#2, one gene set stands out, as expected, significantly to the right side with GS scores between 0.5 and 0.6.

In general, all of the above-mentioned EAME methods perform reasonably well and identify the differentially expressed gene set successfully.

7.5 Comparison of Methods

In this section, we will compare all of the seven methods that have been discussed in the previous chapters: Gene-level GSEA (gGSEA) (discussed in Chapter 5), Probe-level GSEA (pGSEA) (discussed in Chapter 6), and Enrichment Analysis with M-estimators (EAME Method 1-5) (discussed in Chapter 7).

The simulation examples, which are given in Section 5.8, Section 6.2.3 and Section 7.4, show overlays of the observed enrichment scores and the underlying permutated null distributions for the methods of gGSEA, pGSEA and EAME respectively. All of the seven methods seem to achieve the goal by successfully identifying the differentially expressed gene set from the simulations.

To further compare the seven methods, we can look at the power of each of them. As discussed in Section 6.2.4, we can use simulations to obtain an empirical power of a testing method without working out an analytical formula. Details about the steps of comparison are given in Section 6.2.4. As a summary, the data set was generated according to the following:

- Each data value x_{ijk} was generated as $i.i.d.Normal(0,1)$, where i represents for probes and ranges from 1 to 11 in the example; j for arrays and ranges from 1 to 12; k for genes and ranges from 1 to 1000.

- Difference values starting from 0 through 0.6 with an interval of 0.05 in length (i.e. 0, 0.05, 0.10, 0.15, …, 0.60) were added for each run respectively to all probes of the first gene set under the treatment group.
- 1000 simulations were done for each run to compute the rejection rate β_d (see Equation (6.4)).

In section 6.2.4, we compared the empirical power of the gGSEA method and the empirical power of the pGSEA method (see Figure 8). It turns out the pGSEA method is more powerful in detecting the differentially expressed gene set.

Figure 15 shows the comparison of the powers of the three EAME methods that are based on the median and the MAD (Method 1, 2, and 3). It is clear to see that the power of the EAME method #3, which adopts Huber M-estimators of location at the second and third steps, is unanimously higher than the other two methods. The EAME method #1, which takes numeric average at the second and third steps, follows the EAME method #3, and the least powerful method turns out to be the EAME method #2 as expected.

Figure 16 shows the comparison of the powers of the EAME method #4 and #5. These two methods are both based on Huber M-estimators of location and scale. The difference between these two methods is that Method #4 uses the sample mean at the second and third steps, while Method #5 uses a Huber M-estimator of location. The empirical power plot shows almost no difference between these two methods. This example may imply that the Huber M-estimators perform very well when dealing with small sample size, and it may be unnecessary to use robust estimators in the subsequent steps.

A grand comparison of the powers of all of the five EAME methods is given in Figure 17. Apparently, the EAME method #4 and #5, which are based on the Huber M-estimators of location and scale, are slightly more powerful than the EAME method #3, and undoubtedly outperform the other two EAME methods (#1 and #2) that are based on the median and the MAD. The result confirms that the Huber M-estimator is an optimal choice for the family of the EAME methods.

In Figure 18, we compare the GSEA methods (gene-level and probe-level) and the EAME methods (Method 1 - 5) altogether. Although the EAME method #4 and #5 are slightly more powerful than the EAME method #3, the graph shows that the empirical power of the EAME method #4 and #5 are the best among these gene set enrichment methods. Meanwhile the EAME method #1 and the pGSEA methods are roughly in a tie when the difference is small, and when it becomes large, the EAME method #1 gradually beat the pGSEA method. However, the pGSEA method is still more powerful than the EAME method #2 at all time, and it turns out that the gGSEA seems to be the least powerful method amongst all. Numerical results are shown in Table 5.

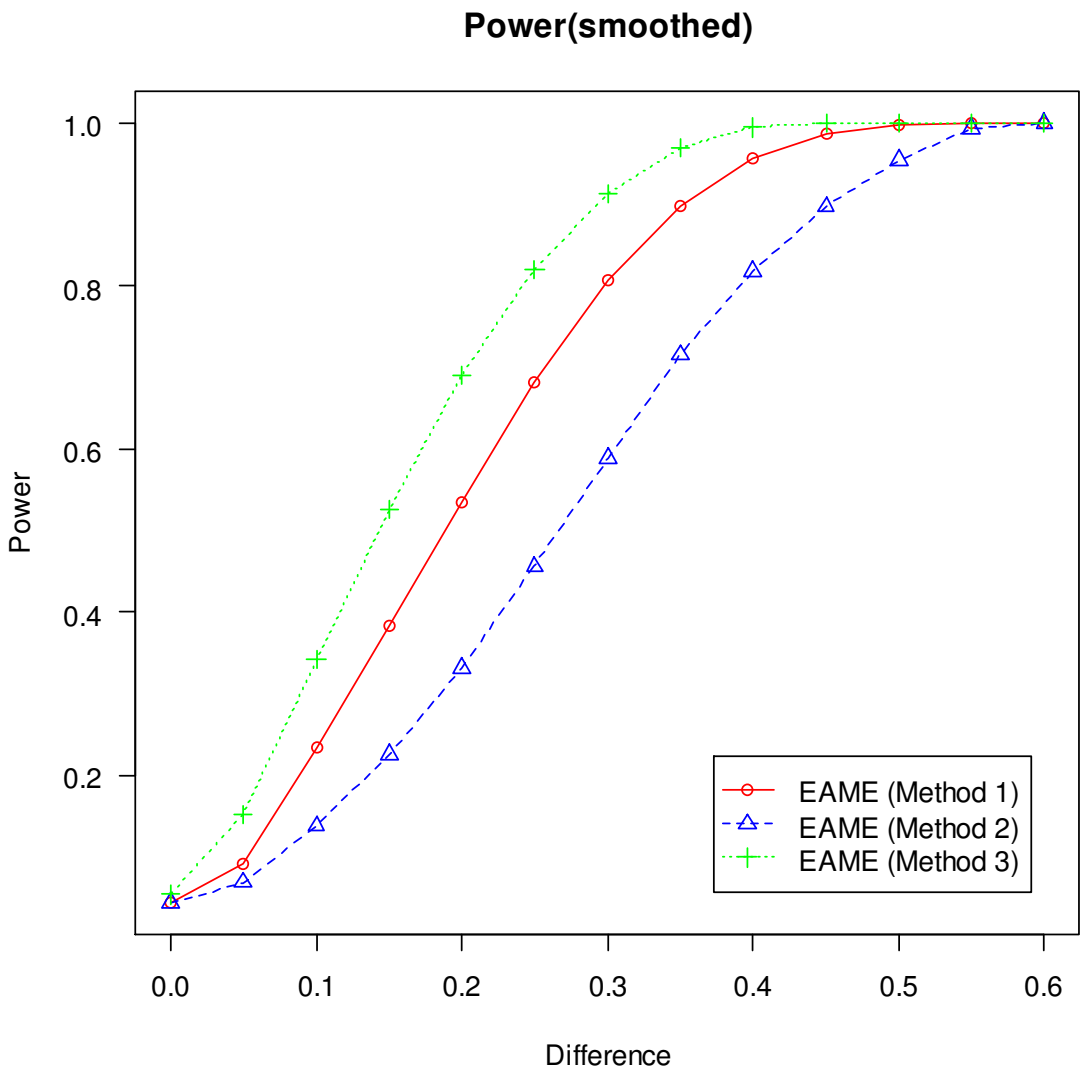


Figure 15: A simulation example for comparing the power of the EAME methods based on Median and MAD

a. The red solid line represents the powers of the EAME method #1, with circles representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

b. The blue dotted line represents the powers of the EAME method #2, with triangles representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

c. The green dashed line represents the powers of the EAME method #3, with plus signs representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

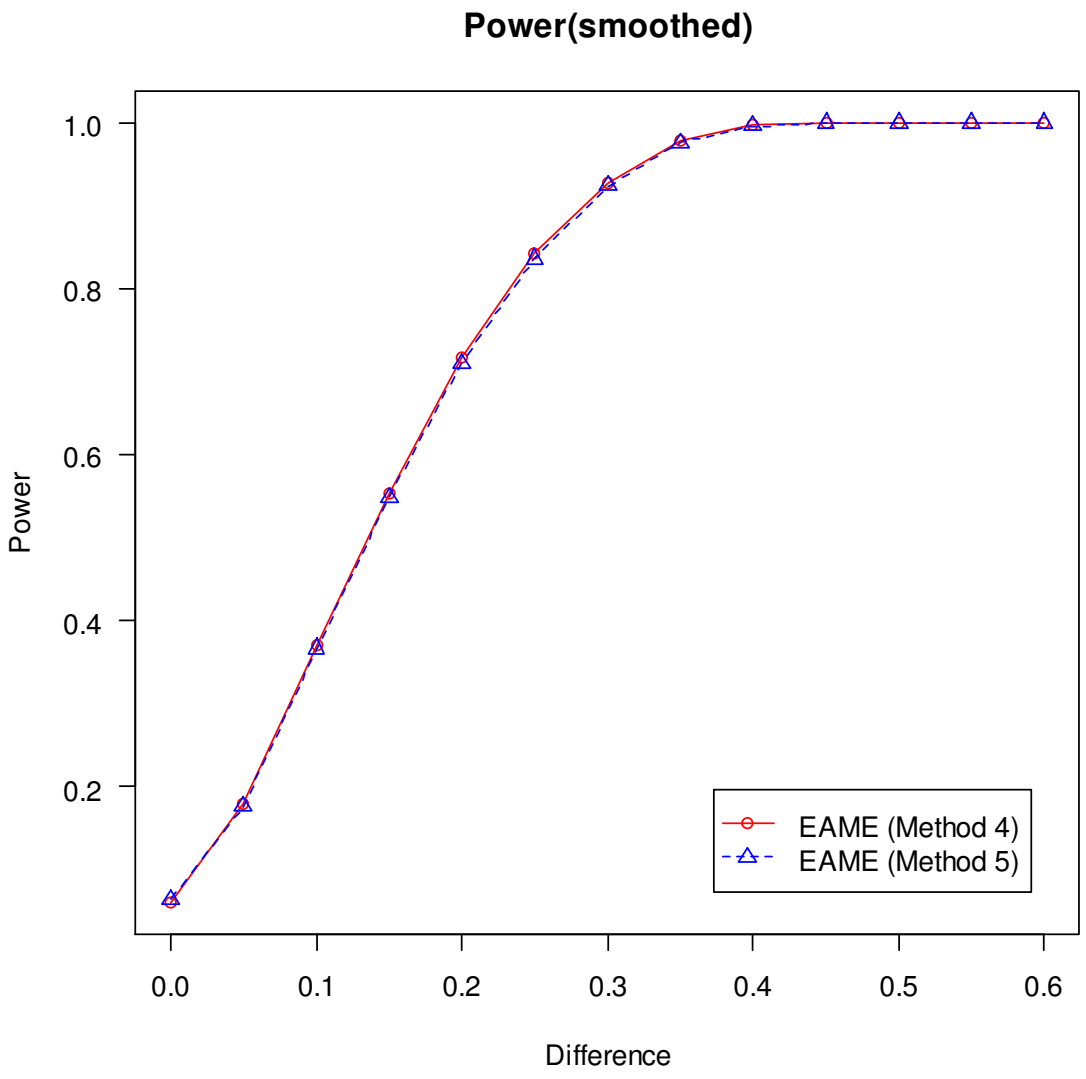


Figure 16: A simulation example for comparing the power of the EAME methods based on Huber M-estimators

a. The red solid line represents the powers of the EAME method #4, with circles representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

b. The blue dotted line represents the powers of the EAME method #5, with triangles representing the powers at each difference value being added to the probes in the first gene set under the treatment group.

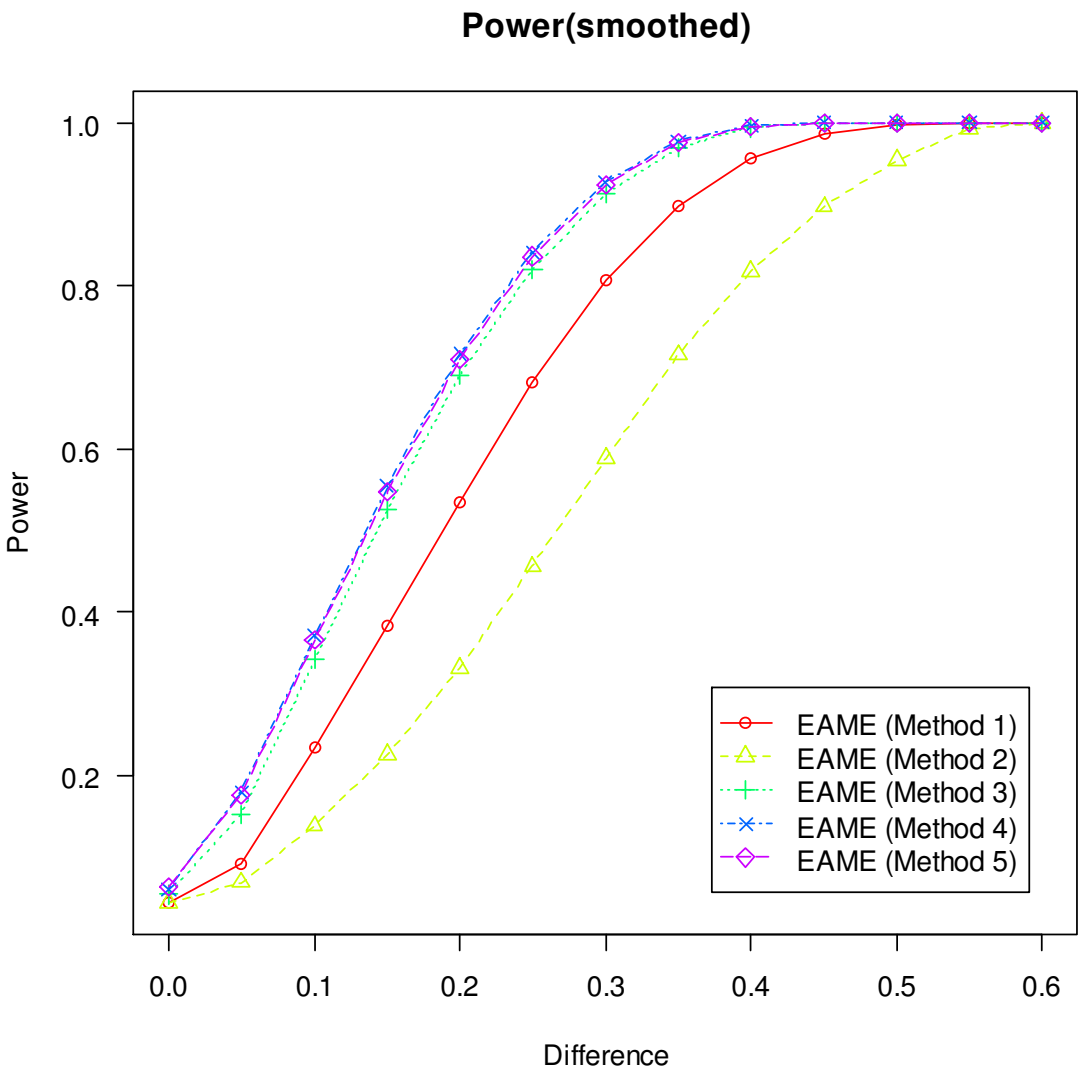


Figure 17: A simulation example for comparing the power of all of the EAME methods (Method 1 - 5)

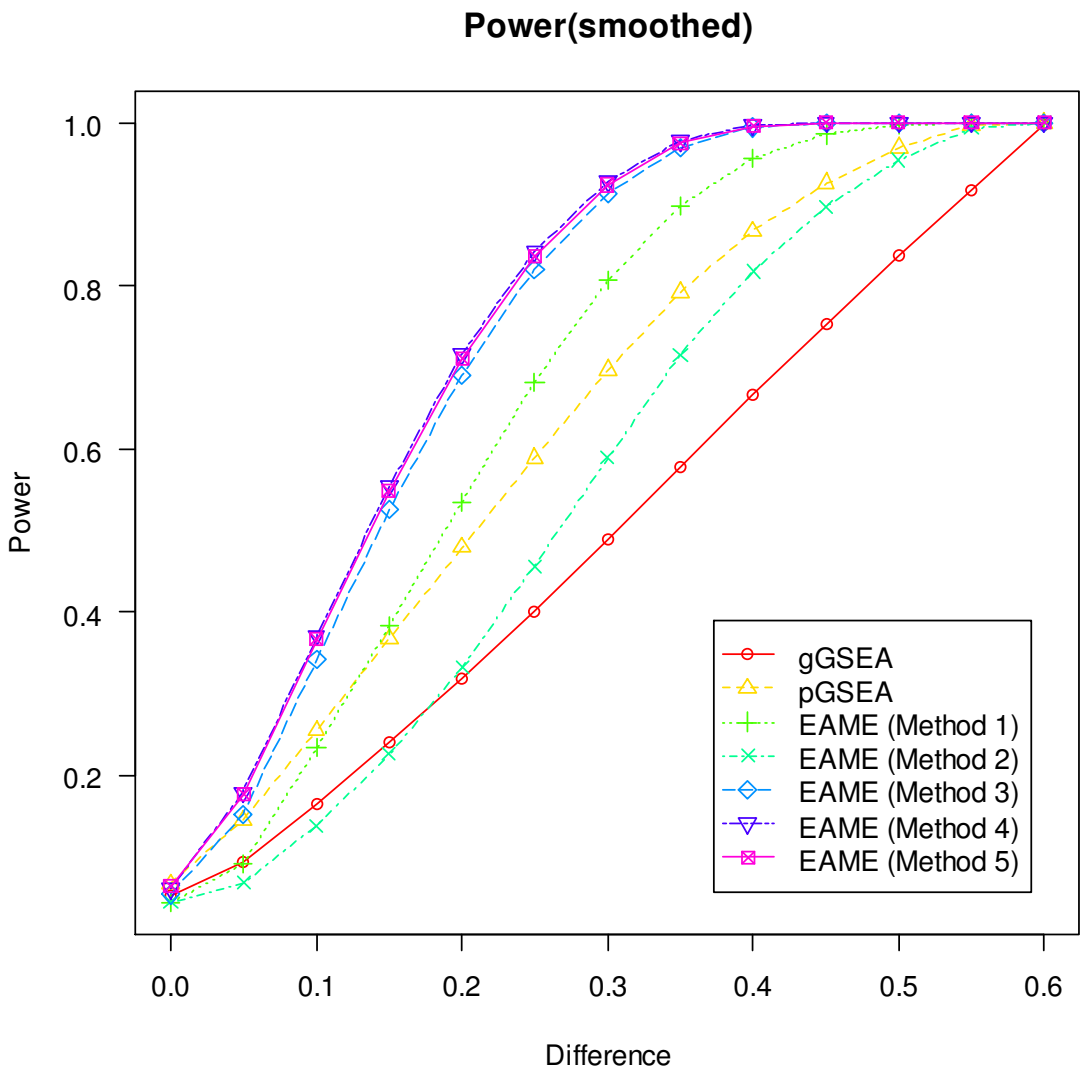


Figure 18: A simulation example for comparing the power of the GSEA methods and all of the EAME methods (Method 1 - 5)

Table 5 Empirical Power (All GSEA and EAME Methods)

Distance	gGSEA	pGSEA	EAME (Method 1)	EAME (Method 2)	EAME (Method 3)	EAME (Method 4)	EAME (Method 5)
0	0.052	0.066667	0.043333	0.043333	0.053	0.06	0.063333
0.05	0.07	0.123333	0.06	0.053333	0.07	0.09	0.09
0.1	0.136667	0.21	0.113333	0.08	0.23	0.236667	0.236667
0.15	0.266667	0.426667	0.306667	0.163333	0.503333	0.59	0.573333
0.2	0.293333	0.463333	0.533333	0.246667	0.883333	0.926667	0.906667
0.25	0.313333	0.493333	0.793333	0.406667	0.99	1	1
0.3	0.466667	0.753333	0.94	0.603333	1	1	1
0.35	0.666667	0.893333	0.99	0.85	1	1	1
0.4	0.676667	0.903333	1	0.876667	1	1	1
0.45	0.733333	0.973333	1	0.953333	1	1	1
0.5	0.876667	0.996667	1	0.996667	1	1	1
0.55	0.93	1	1	0.996667	1	1	1
0.6	0.963333	1	1	1	1	1	1

Chapter 8
An R package for Robust DNA Microarray Analysis and Examples

8.1 R Packages and Data

R is a programming language and a free software environment for statistical analysis and visual illustration. It was initially written by Ross Ihaka and Robert Gentleman at the Department of Statistics of the University of Auckland in Auckland, New Zealand, and currently developed by the *R Development Core Team*.

The base R distribution comes with a lot of commonly widely used statistical functionalities, and it can also be easily extended via *packages* when tasks cannot be fulfilled using the base R. A package in R is a user-created block of R codes (or scripts from other programming languages such as C) that includes specialized statistical techniques, graphical devices, import and/or export capabilities, reporting tools, etc. To handle Affymetrix DNA microarray data, we will need the *affy* package, which contains functions for the storage, management and analysis of Affymetrix probe-level data (Gautier *et al*., 2003). Another useful R package for DNA microarray analysis is the *DNAMR* package[1], developed and maintained by D. Amaratunga and J. Cabrera. This package

[1] The `DNAMR` package can be downloaded from
`http://www.rci.rutgers.edu/~cabrera/DNAMR/`

contains many useful routines for microarray data analysis. The advantage of using the DNAMR package instead of using the similar functions in the base R is that the computing speed of the former is usually much faster than the latter, especially for data sets with a huge amount of entries (e.g. DNA microarray data). To further meet the needs of analytical methods discussed in this paper, some modifications and supplements to the DNAMR package, and implementations of new methodologies discussed in this paper have been done and compiled as a new package called *DNARA*.

In the following section, we will use two real DNA microarray datasets to illustrate the usage of the abovementioned statistical tools. The data sets, called *RMA0* and *RMA18*, come from an experiment conducted to study whether mice whose Slc17A5 gene had been knocked out could be distinguished from wild-type mice at the gene expression level (Amaratunga, Cabrera and Lee, 2008). It recorded the gene expression measurements from 0-day-old (newborn) and 18-day-old mice using Affymetrix Mouse430_2 GeneChips.

8.2 An Example

In this section, we will use an example to illustrate the methodologies discussed in this paper. First, we need to make sure the `affy` package and the `DNARA` package have been installed properly. To install the `affy` package, start R and type in the R console the following codes:

```
source("http://bioconductor.org/biocLite.R")
biocLite("affy")
```

To install the `DNARA` package, you want to click the "Packages" tab on the menu of the R window, and click "install package(s) from local zip files…", then locate the zip file and click the "open" button.

The following code is used to read the raw Affymetrix DNA microarray data into R:

```
library(affy)
rawcelf  <- ReadAffy(filenames = celf)
# celf is a vector containing the filenames of the raw data
```

8.2.1 Obtaining Gene Expressions

To obtain gene expression from probe-level DNA microarray data using the regular RMA method, we need to go through the following steps: background correction, quantile normalization and summarization. This can be done by using the `rma()` function offered in the `affy` package:

```
gene <- rma(rawcelf)
```

, or we can compute the RMA gene expression step by step, by using `bg.correct()` and `normalize()` offered in the `affy` package, and then use `medpolish()`[1] offered in the base R distribution:

```
y        <- bg.correct(rawcelf, method = 'rma')     # background correction
y        <- normalize(z, method = 'quantiles')      # quantile normalization
ymat     <- intensity(y)                            # get intensity matrix
pindex   <- indexProbes(y, which = 'pm');           # get index of PM probes

# function to perform median polish on ONE probe set
doMedPolish      <- function(p.index, y.mat, method)
{
         Z        <- y.mat[p.index, ];
         obj     <- polish(z, method = method, maxiter = 40, trace.iter = FALSE);
         g        <- obj$overall + obj$col;
         return(g);
}

# take log2 on the intensity matrix
# after background correction and quantile normalization
log2y    <- log2(ymat);
```

[1] In the example shown below, a function called `polish()`, a modified version of `medpolish()`, was adopted. The new version takes one more parameter that tells which location estimator (i.e. mean, median, Tukey's biweight or Huber's M-estimator) is to be used.

```
# apply median polish for each probe set (gene)
gene     <- t(sapply(pindex, doMedPolish, log2y, "med"));
```

`doMedPolish()` performs central polish for one probe set (gene). It gives the same summarization result as `rma()` does, when using `method = "med"` option. As discussed in Chapter 4, we can also use the mean, Tukey's biweight estimator or Huber's M-estimator by submitting `method = "mean"`, `method = "biweight"`, or `method = "huber"` respectively.

To find out significantly differentially expressed genes, we can perform a two-sample *t*-test for every gene. The following graph (Figure 19) compares the

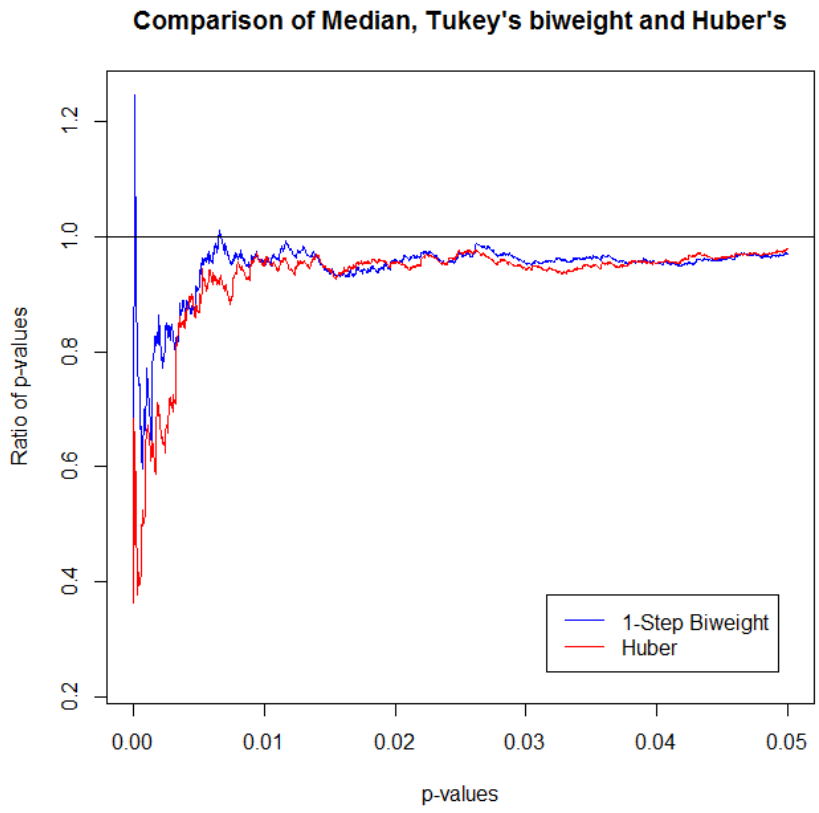


Figure 19 Comparison of Location Estimators in the summarization step of RMA

A two-sample *t*-test was performed on the gene expressions based on the RMA method using median, Tukey's biweight (1-step) and Huber's M-estimator respectively. The ratios of p-values / RMA p-values were drawn.

performances of Median, Tukey's biweight and Huber's M-estimator for the RMA18 data. We use the p-values from the regular RMA method (i.e. using median) as the baseline, and compute the ratio of p-values from Tukey's biweight and Huber's over the baseline. It is clear to see that both of Tukey's biweight and Huber's M-estimator perform better than median since the ratios are below 1, with an exception for Tukey's biweight when its p-values are near 0. Nonetheless, it beats median for most of the time. Huber's M-estimator shows the best performance among the three location estimators.

8.2.2 Gene Set Enrichment Analysis (GSEA) – Gene-level and Probe-level

In this section, we will introduce how to use R to perform Gene Set Enrichment Analysis on both gene-level and probe-level. We will still use the RMA0 and RMA18 datasets. After reading in the data set using `ReadAffy()` function, we can type the variable name (`rawcelf`) in the R console and a summary of the data set will be printed as follows:

```
>
> rawcelf
AffyBatch object
size of arrays=1002x1002 features (20 kb)
cdf=Mouse430_2 (45101 affyids)
number of samples=12
number of genes=45101
annotation=mouse4302
notes=
```

The summary shows some information about the DNA microarray data, including the number of samples, genes, and the annotation information of the data. This data set contains 12 samples and we know they are divided into two groups, one as control and the other as treatment.

To perform GSEA, we also need to acquire the information about gene sets. This can be done by installing and loading the relevant annotation data. For the RMA18 dataset, we need Affymetrix Mouse Genome 430 2.0 Array annotation data (chip mouse4302). To install it, type the following in R console:

```
source("http://bioconductor.org/biocLite.R")
biocLite("mouse4302.db")
```

, or we can obtain the annotation information and load the package automatically. The following code will determine which annotation data is going to be used, and will install (if not installed yet) and load the package automatically.

```
# get gene set information. GO term.
# "object" is the affy data
anno <- annotation(object);
pkg  <- paste(anno, ".db", sep = "");

# check if the required annotation pkg exists?
# DNARA package required
if(!is.PkgInstalled(pkg))
{
        source("http://bioconductor.org/biocLite.R");
        biocLite(pkg);
}

b_PkgLoaded <- require(pkg, character.only = TRUE, quietly = TRUE); # load the
package
```

The mapping between the gene sets (GO terms[1]) and genes (or probes) is stored in mouse4302GO, an R object. Since the number of gene sets described in mouse4302GO is too large, we will intentionally choose genes containing exactly 11 probes, and gene sets that contain exactly 50 such genes. By doing so, we reduce the number of gene sets to 14. The following code does the job as described above:

```
## -- control and treatment group
grp     <- rep(1:2, each = 6) # two groups
```

[1] For detailed information about Gene Ontology (GO), please visit http://www.geneontology.org/

```
permutations     <- col.permute(500, c(6, 6));# pre-create a permutation table

## -- GO info -- ##
n <- length(grp);
n.probes <- 11;
##############################
## --- choose probe sets --- ##
ip       <- indexProbes(object, 'pm');
ps       <- ip[sapply(ip, length) == n.probes];
ps.name  <- names(ps);
GO.info  <- paste(anno, "GO", sep = "");
GO       <- eval(parse(text = GO.info));
GO.table <- toTable(GO[ps.name]);  # create a gene-to-GO table

gn       <- unique(GO.table[,"probe_id"]); # gn is the genes in GO.table AND
that have #probes == 11 (or specified number in the parameter)
gsets    <- split(GO.table[, 'probe_id'], GO.table[, 'go_id']);
goid     <- names(gsets);
gsets2   <- gsets[goid[sapply(gsets, length) == 50]]
```

To perform GSEA at gene-level:

```
## -- gene-level GSEA -- ##
xg       <- exprs(rma(object))[ps.name, ];
zg       <- gsea(xg, grp, gsets2, permutations);
```

To perform GSEA at probe-level:

```
## -- probe-level GSEA -- #
xp       <- probes(object, which = 'pm');
zp       <- gsea(xp, grp, gsets2, permutations);
```

The `gsea()` function will return a list containing two objects. One is a vector containing the GSEA geneset scores of the sample, and the other is a matrix containing the GSEA geneset scores of the permutations, which will be served as the null distribution for p-value calculation.

The following code will draw histograms of the geneset scores for the sample and the permutations as well.

```
## -- draw histogram
hist.gsea(zg$es1, zg$es0)
```

```
hist.gsea(zp$es1, zp$es0)
```

The histograms are shown in Figure 20.

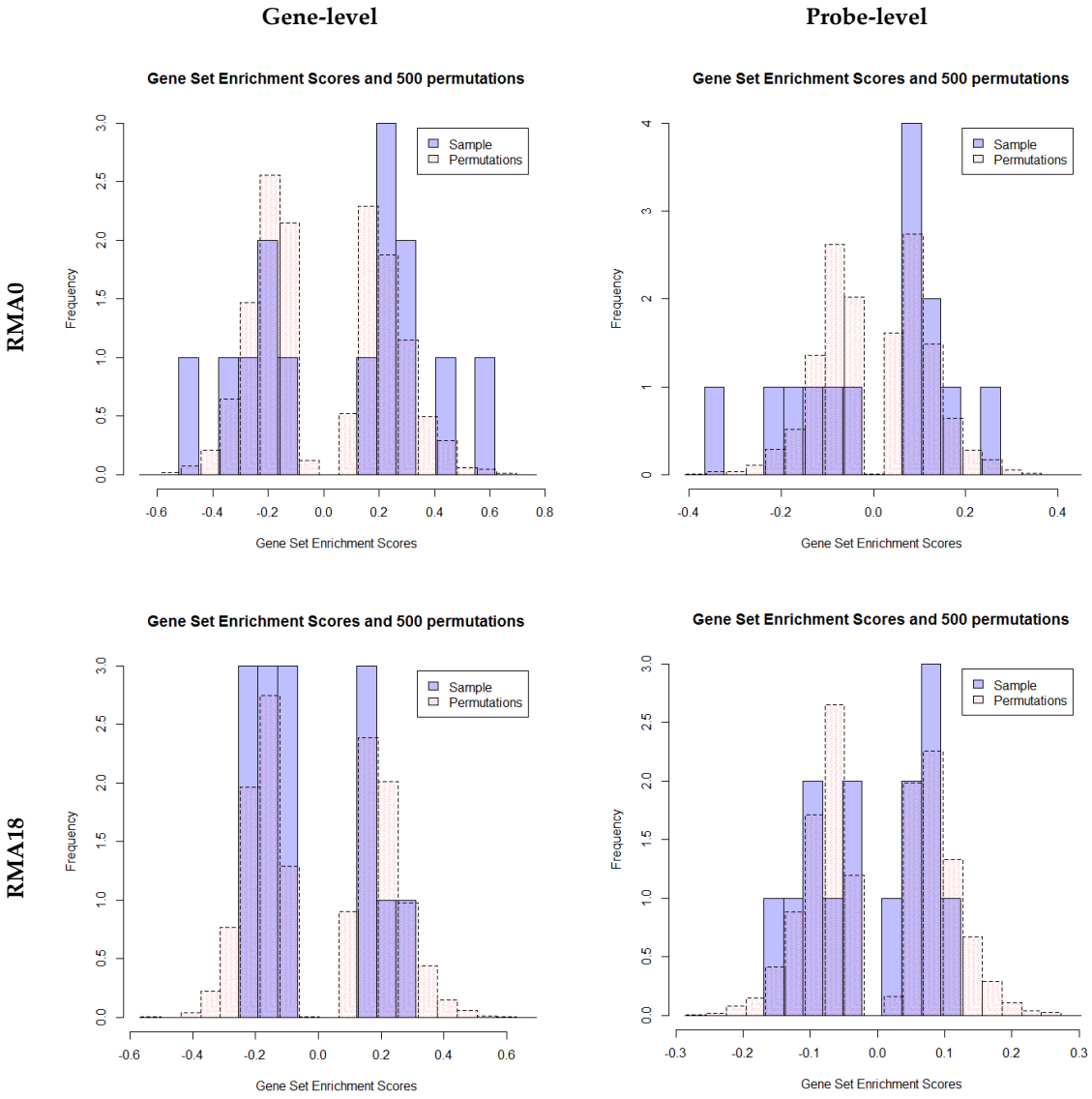


Figure 20 Histograms of GSEA geneset scores for RMA0 and RMA18 data sets at Gene-level and Probe-level

The following code will tell which gene sets are significant at 5% level:

```
pvg      <- p.value(zg$es1, zg$es0)
sum(pvg < 0.05)
which(pvg < 0.05)

pvp      <- p.value(zp$es1, zp$es0)
```

```
sum(pvp < 0.05)
which(pvp < 0.05)
```

We can also use the following code to find the terms associated with the gene sets after installing the GO.db package:

```
### use GO.db to find the Terms associated with those GOIDs
source("http://bioconductor.org/biocLite.R")
biocLite("GO.db")
library("GO.db")
Term(names(which(pvg < 0.05)))
```

The significant gene sets found by using GSEA method at gene-level as well as probe-level are shown in Table 6.

Table 6 Significant Gene Sets (GSEA method)

		GOID	Description
RMA0	Gene-level	GO:0005201	extracellular matrix structural constituent
		GO:0008652	cellular amino acid biosynthetic process
		GO:0019001	guanyl nucleotide binding
		GO:0070577	histone acetyl-lysine binding
	Probe-level	GO:0005201	extracellular matrix structural constituent
		GO:0017075	syntaxin-1 binding
		GO:0045880	positive regulation of smoothened signaling pathway
		GO:0070577	histone acetyl-lysine binding
RMA18	Gene-level	No significant gene set found.	
	Probe-level	GO:0005201	extracellular matrix structural constituent

8.2.3 Enrichment Analysis using M-estimators (EAME)

This section shows examples for EMME methods using R. Assuming raw DNA microarray data has been read and PM intensities have been extracted, we can use `eame()` function included in the `DNARA` package to perform EAME analysis. For each of the five different methods discussed in this paper, we can submit the `method=` option in the `eame()` function. For example, to perform EAME using method 3, we can use the following code:

```
e3       <- eame(xp, grp, gsets2, method = 3)
```

Similar to the `gsea()` function, `eame()` function will return a list containing two slots, one with the EAME geneset scores for the sample, and the other for the permutations. To draw the histograms of the geneset scores and find out the significant gene sets, we can use the following code:

```
e3       <- eame(xp, grp, gsets2, method = 3)
hist.gsea(e3$es1, e3$es0)
pv       <- p.value(e3$es1, e3$es0)
sum(pv < 0.05)
e3.goid  <- names(which(pv < 0.05))
```

The histograms are shown in Figure 21 through Figure 25.

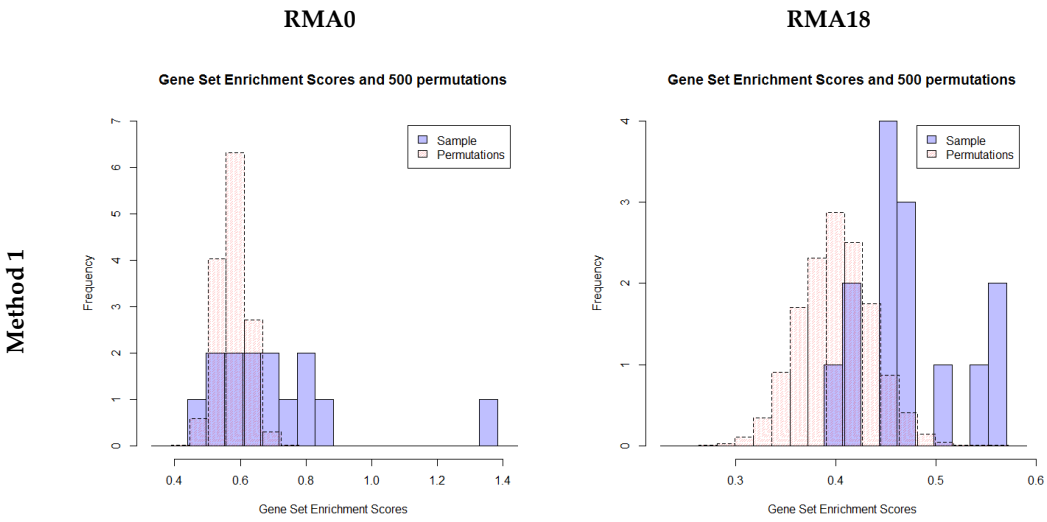


Figure 21 Histograms of EAME geneset scores for RMA0 and RMA18 data sets (Method 1)

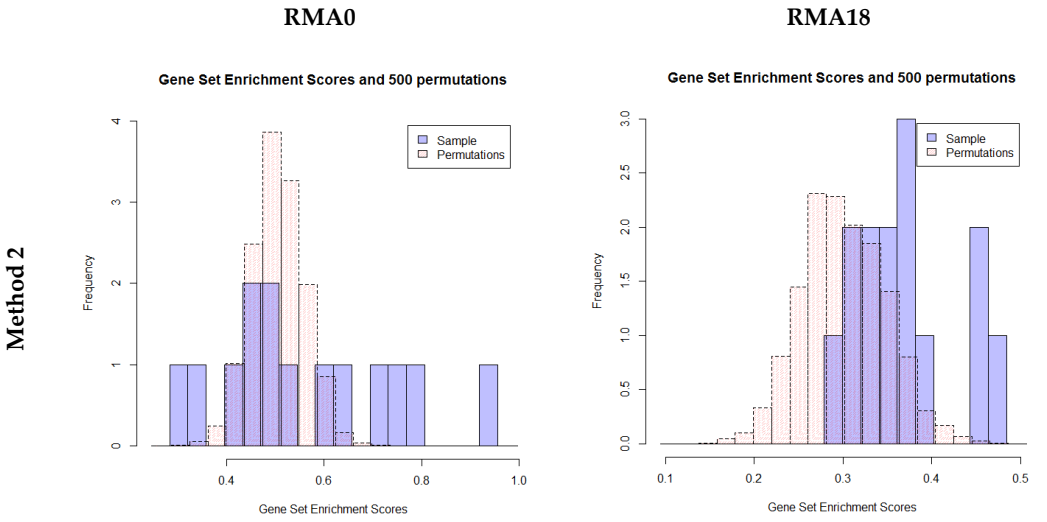


Figure 22 Histograms of EAME geneset scores for RMA0 and RMA18 data sets (Method 2)

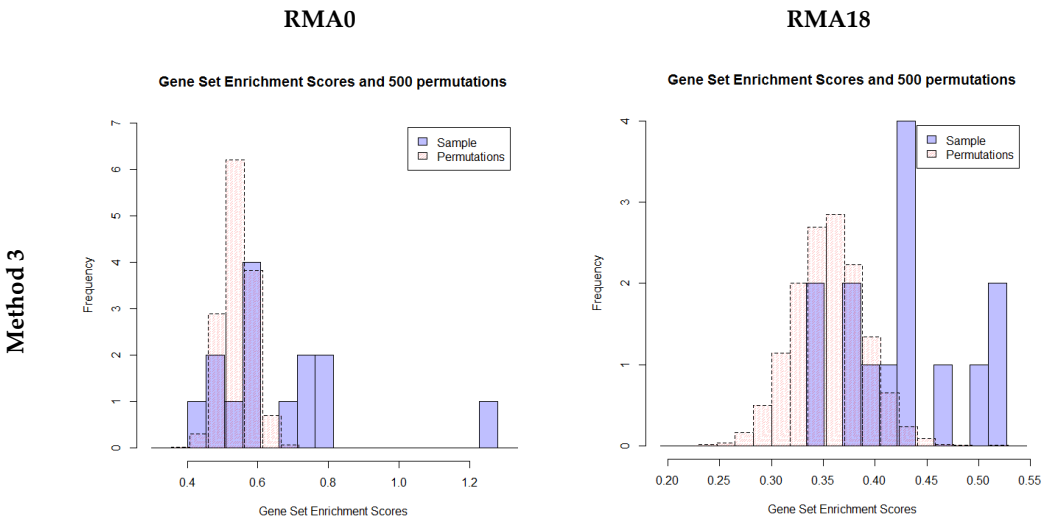


Figure 23 Histograms of EAME geneset scores for RMA0 and RMA18 data sets (Method 3)

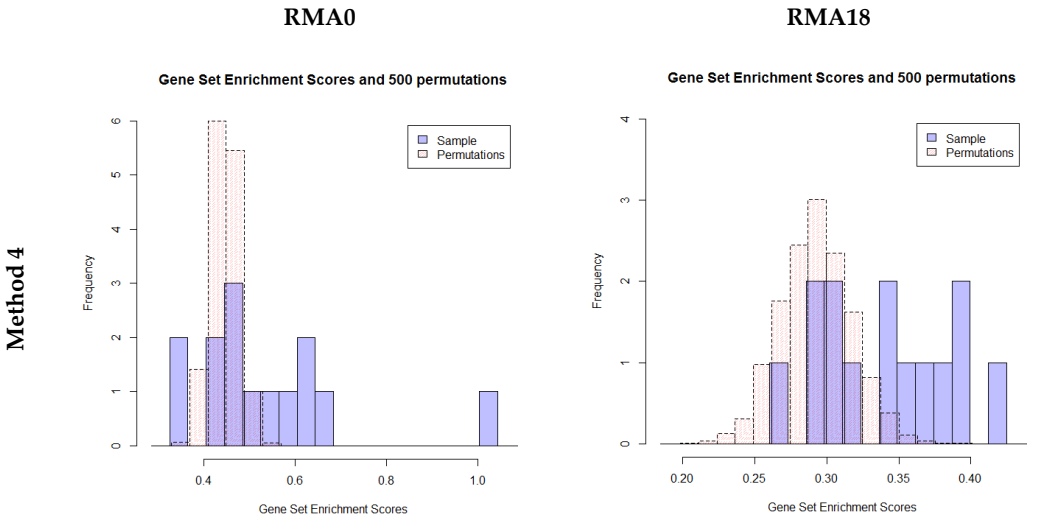


Figure 24 Histograms of EAME geneset scores for RMA0 and RMA18 data sets (Method 4)

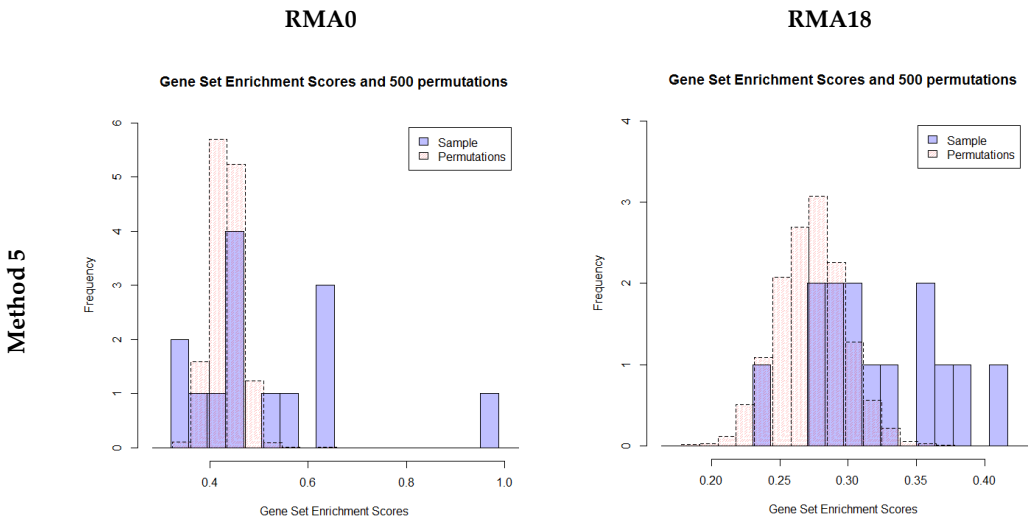


Figure 25 Histograms of EAME geneset scores for RMA0 and RMA18 data sets (Method 5)

In Table 7 and Table 8, there list significant gene sets found by various GSEA and EAME methods for the RMA0 and RMA18 data sets. For both the RMA0 and RMA18 data, GO:0005201 and GO:0070577 are found to be significant by most of the methods.

Table 7 Significant Gene Sets for RMA0 Data

Method	GO ID									
	GO:0005201	GO:0045880	GO:0048745	GO:0070577	GO:0071837	GO:2000134	GO:2001243	GO:0008652	GO:0019001	GO:0017075
GSEA (gene)	RMA0			RMA0				RMA0	RMA0	
GSEA (probe)	RMA0	RMA0		RMA0						RMA0
EAME (1)	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0			
EAME (2)	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0				
EAME (3)	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0				
EAME (4)	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0			
EAME (5)	RMA0	RMA0	RMA0	RMA0	RMA0	RMA0				

GO Terms :

GO:0005201 "extracellular matrix structural constituent"; GO:0045880 "positive regulation of smoothened signaling pathway"; GO:0048745 "smooth muscle tissue development"; GO:0070577 "histone acetyl-lysine binding"; GO:0071837 "HMG box domain binding"; GO:2000134 "negative regulation of G1/S transition of mitotic cell cycle"; GO:2001243 "negative regulation of intrinsic apoptotic signaling pathway"; GO:0008652 "cellular amino acid biosynthetic process"; GO:0019001 "guanyl nucleotide binding"; GO:0017075 "syntaxin-1 binding" **(significant gene sets are marked with RMA0)**

Table 8 Significant Gene Sets for RMA18 Data

Method	GO ID										
	GO:0005201	GO:0032809	GO:004301	GO:0045880	GO:0048745	GO:0070577	GO:0071837	GO:0008652	GO:0019001	GO:0042562	GO:2001243
GSEA (gene)											
GSEA (probe)	RMA18										
EAME (1)	RMA18	RMA18	RMA18	RMA18	RMA18	RMA18	RMA18				
EAME (2)	RMA18			RMA18		RMA18	RMA18	RMA18	RMA18		
EAME (3)	RMA18	RMA18	RMA18	RMA18	RMA18	RMA18	RMA18		RMA18	RMA18	
EAME (4)	RMA18	RMA18		RMA18	RMA18	RMA18	RMA18			RMA18	RMA18
EAME (5)	RMA18	RMA18		RMA18	RMA18	RMA18	RMA18			RMA18	

GO Terms :

GO:0005201 "extracellular matrix structural constituent"; GO:0032809 "neuronal cell body membrane"; GO:0043015 "gamma-tubulin binding"; GO:0045880 "positive regulation of smoothened signaling pathway"; GO:0048745 "smooth muscle tissue development"; GO:0070577 "histone acetyl-lysine binding"; GO:0071837 "HMG box domain binding"; GO:0008652 "cellular amino acid biosynthetic process"; GO:0019001 "guanyl nucleotide binding"; GO:0042562 "hormone binding"; GO:2001243 "negative regulation of intrinsic apoptotic signaling pathway" **(significant gene sets are marked with RMA18)**

Chapter 9
Conclusion and Remarks

In this paper, we first introduced and discussed several current methodologies that deal with DNA microarray data, and then we discussed in details about the application of robust estimators of location in the summarization step in the RMA method. We compared the performance of several candidates including mean, median (currently used in RMA), Tukey's biweight and Huber's M-estimator, and find that robust estimators beat the sample mean when outliers are present in the data. Also we found that Huber's M-estimator works much better than others under the condition that there exist only a small proportion of outliers in the data set.

In the following chapters, we focused on the methodologies that take advantage of the idea of Gene Set Enrichment. We talked about the Gene Set Enrichment Analysis (GSEA) method that is currently carried out at the gene-level, which is called gGSEA in this paper. The method requires the data set be converted from the raw probe-level data into the gene-level data before further analysis. This step is called the preprocessing step of DNA microarray data, and usually it contains a sequence of three major substeps: background correction, normalization and summarization.

Although there are a great number of various methods that have been developed to handle the three steps, and improvements and new methods are

being studied, it is inevitable to lose important information that is contained in the probe-level microarray data through the preprocessing stage of microarray data analysis. It is reasonable to believe that the analysis directly based on the probe-level data may give more reliable result. Therefore, we made several proposals to solve this issue.

First, we extended the gene-level GSEA method (gGSEA) by modifying the algorithm to allow us to apply the method directly onto the probe-level microarray data, which is called the probe-level GSEA method (pGSEA) in this paper. A simulation comparison shows that the pGSEA method works much better than the original gGSEA method because the empirical power of the pGSEA method for detecting differentially expressed gene sets is higher than that of the gGSEA method.

Secondly, we proposed another gene set enrichment method that utilizes the robust M-estimators. Since this method adopts the idea of enrichment analysis and takes advantage of robust M-estimators, we call it the *Enrichment Analysis with M-estimators* (EAME) method. We first used the median and the MAD, and then generalized it as Huber M-estimators, of which sample mean and sample median are both a special case. A simulation comparison was performed and it showed that the EAME methods (Method #4 and #5) with Huber M-estimators has the highest power for detecting the differentially expressed gene set, and is followed by the EAME methods (Method #1, #2 and #3) based on median and MAD.

When comparing the GSEA method and the EAME methods altogether, it turns out that the EAME method #4 and #5 outperform both of the gGSEA method and the pGSEA method, while the EAME method #1 performs roughly

as well as the pGSEA method when the difference value is small, and it gradually beats the pGSEA method when the difference value becomes larger and larger. However, the EAME method #2 is still not a competitor compared to the pGSEA method. Finally, it turns out that the least powerful method among all of the GSEA and the EAME methods is the gGSEA method.

References

1. Amaratunga, D. and J. Cabrera (2004), "Exploration and Analysis of DNA Microarray and Protein Array Data", *Wiley*.
2. Amaratunga, D. and J. Cabrera (2001a), "Outlier resistance, standardization and modeling issues for DNA microarray data", In L. T. Fernholz, S. Morgenthaler, and W. Stahel, eds., *Statistics and Genetics for the Envirnmental Sciences*, Basel: Birkhauser-Verlag.
3. Amaratunga, D. and J. Cabrera (2001b), "Statistical analysis of viral microchip data", *J. Am. Stat. Assoc*., 96, 1161-1170.
4. Amaratunga, D., J. Cabrera and Y.-S. Lee (2008), "Enriched random forests", *Bioinformatics*, 24:2010-2014.
5. Benjamini, Y. and Y. Hochberg (1995), "Controlling the False Discovery Rate: A practical and power ful approach to multiple testing", *Journal of the Royal Statistical Society*, B57, 289-300.
6. Bolstad, B. M., R. A. Irizarry, M. Astrand, and T. P. Speed (2003), "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Bias and Variance", *Bioinformatics*, 19(2):185-193
7. Carlson, M. (). *"GO.db: A set of annotation maps describing the entire Gene Ontology*". R package version 2.10.1.
8. Carlson, M. (). *"mouse4302.db: Affymetrix Mouse Genome 430 2.0 Array annotation data (chip mouse4302)*". R package version 2.10.1.
9. Efron, B. and R. Tibshirani (2007), "On Testing the Significance of Sets of Genes", *Annals of Applied Statistics*, Volume 1, Number 1, 107-129.
10. Fodor, S. P., J. L. Read, M. C. Pirrung, L. Stryer, A. T. Lu and D. Solas (1991), "Light-directed, spatially addressable parallel chemical synthesis", *Science*, 251:767-773.
11. Kanehisa, M. and S. Goto (2000), "KEGG: Kyoto Encyclopedia of Genes and Genomes", *Nucleic Acids Res.* 28, 27-30.
12. Kohane, I. S., A. T. Kho and A. J. Butte (2003), "Microarrays for an Integrative Genomics", *The MIT Press*.
13. Gautier, L. (2011). "affydata: Affymetrix Data for Demonstration Purpose". R package version 1.11.18.
14. Gautier, L., L. Cope, B. M. Bolstad and R. A. Irizarry (2004), "affy – analysis of Affymetrix GeneChip data at the probe level", *Bioinformatics*, 20(3):307-315.
15. Hampel, F. R. (1968), "Contributions to the theory of robust estimation", PhD thesis, *University of California, Berkeley*.

16. Hoaglin, D. C., F. Mosteller, and J. W. Tukey (2000), "Understanding Robust and Exploratory Data Analysis", *Wiley*.
17. Hornik, K. (2013), "The R FAQ", *http://CRAN.R-project.org/doc/FAQ/R-FAQ.html*
18. Huber, P. J. (1964). "Robust estimation of a location parameter", *Annals of Mathematical Statistics*, 35,73-101.
19. Irizarry, R. A., B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs and T. P. Speed (2003), "Summaries of Affymetrix GeneChip probe level data", *Nucleic Acids Research*, 31(4):e15.
20. Irizarry, R. A., B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf and T. P. Speed (2003), "Exploration, normalization, and summaries of high density oligonucleotide array probe level data", *Biostatistics*, 4(2):249-64.
21. Irizarry, R. A., Z. Wu and H. A. Jaffee (2006), "Comparison of Affymetrix GeneChip expression measures", *Bioinformatics*, ;22(7):789-94.
22. Lemon, W. J., S. Liyanarachchi, and M. You (2003), "A high performance test of differential gene expression for oligonucleotide arrays", *Genome Biology*, 4:R67.
23. Li, C. and W. H. Wong (2001a), "Model-based analysis of oligonucleotide arrays: model validation, design issues and standard error application", *Genome Biology*, 2(8):research 0032.1 – 0032.11
24. Li, C. and W. H. Wong (2001b), "Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection", *Proc. Nat. Acad. Sci.*, 98:31-36.
25. Lockhart, D., H. Dong, M. Byrne, M. Follettie, M. Gallo, M. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton and E. Brown (1996), "Expression monitoring by hybridization to high-density oligonucleotide arrays", *Nature Biotechnology*, 14, 1675-1680.
26. Maronna, R. A., R. D. Martin, and V. J. Yohai (2006), "Robust Statistics: Theory and Methods", *Wiley*, 26.
27. Miller, R. A, A. Galecki, and R. J. Shmookler-Reis (2001), "Interpretation, design, and analysis of gene array expression experiments", *Journal of Gerontology*, 56A:B52-B57.
28. Naef, F., D. A. Lim, N. Patil and M. O. Magnasco (2001), "From features to expression: High-density oligonucleotide array analysis revisited", *Proc DIMACS Workshop on Analysis of Gene Expression Data*.
29. Schena, M., D. Shalon, R. W. Davis and P. O. Brown (1995), "Quantitative monitoring of gene expression patterns with a complementary DNA microarray", *Science*, 270, 467-470.
30. Shapiro, S. S., and M. B. Wilk (1965). "An analysis of variance test for normality (complete samples)". *Biometrika*, 52 (3-4): 591–611.

31. Simon, R. M., E. L. Korn, L. M. McShane, M. D. Radmacher, G. W. Wright and Y. Zhao (2003), "Design and Analysis of DNA Microarray Investigations", *Springer*.
32. Storey, J. D. (2001), "The positive False Discovery Rate: A Bayesian interpretation and the *q*-value", *Technical Report of the Stanford University Department of Statistics*.
33. Storey, J. D. and R. Tibshirani (2001), "Estimating false discovery rates under dependence, with applications to DNA microarrays", *Technical Report of the Stanford University Department of Statistics*.
34. Subramanian, A., P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov (2005), "Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles", *Proc. Natl. Acad. Sci. USA*. 102(43): 15545–15550.
35. The Gene Ontology Consortium (2005), "Gene ontology: tool for the unification of biology." *Nature Genetics*, 25(1):25-9.
36. Wu, Z., R. A. Irizarry, R. Gentleman, F. M. Murillo and F. Spencer (2004), "A Model Based Background Adjustment for Oligonucleotide Expression Arrays", Johns Hopkins University, Dept. of Biostatistics Working Papers. Working Paper 1. http://biostats.bepress.com/jhubiostat/paper1
37. Yang, Y. H., M. J. Buckley, S. Dudoit and T. P. Speed (2000). "Comparison of methods for image analysis on cDNA microarray data", *Technical Report of the Department of Statistics, University of California at Berkeley*.