A COMPARISON OF METHODS TAKING INTO ACCOUNT ASYMMETRY WHEN EVALUATING DIFFERENTIAL EXPRESSION IN GENE EXPRESSION EXPERIMENTS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Seguy Tchakounte-Wakem

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

May 2018

Fargo, North Dakota

North Dakota State University
Graduate School

Title

A COMPARISON OF METHODS TAKING INTO ACCOUNT ASYMMETRY WHEN EVALUATING DIFFERENTIAL EXPRESSION IN GENE EXPRESSION EXPERIMENTS

By

Seguy Tchakounte-Wakem

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Megan Orr
Chair

Dr. Rhonda Magel

Dr. Diomo Motuba

Approved:

June 21, 2018
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

Gene expression technologies allow expression levels to be compared across treatments for thousands of genes simultaneously. Asymmetry in the empirical distribution of the test statistics from the analysis of a gene expression experiment is often observed. Statistical methods exist for identifying differentially expressed (DE) genes while controlling multiple testing error while taking into account the asymmetry of the distribution of the effect sizes. This paper compares three statistical methods (Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate) used to identify differentially expressed (DE) genes that take into account such patterns while controlling false discovery rate (FDR). The results of the simulation studies performed suggest that the Modified Q-values outperforms the other methods most of the time and also better controls the FDR.

ACKNOWLEDGEMENTS

I thank God for guiding me through my master’s education. I would like to also express my sincere gratitude to my advisor, Dr. Megan Orr for her continuous support, motivation, enthusiasm, and knowledge. She guides, helped me all the time of research and writing of this paper. Having only one professor in our department with knowledge in this area was frustrating at first, but I quickly realize that she alone was enough to guide me.

Beside Dr. Orr, I would like to thank the rest of my committee: Dr. Magel and Dr. Diomo, for their encouragement, insight and comments. I also thank Dr. Emmanuel Nojang for words of encouragement and motivation.

I wish to thank my wife Nadine Tchakounte, my brothers and sisters, my mother in-law and friend and class mates Muhammed Saho, Asare Mbouampon for their encouragement, love, time and prayers. I owe them everything and wish I could show them how much I love and appreciate them.

DEDICATION

I dedicate this Thesis to my wife, my kids and my late mother

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

EM	Expectation maximization
DRG	Down regulated Genes
URG	Up regulated Genes
DE	Differentially expressed.
EE	Equivalently expressed
FWER	Familywise error rate
FDR	False discovery rate
SAM	Significance analysis of microarray
DDE	Declare differentially expressed
*l*FDR	local false discovery rate
CDF	Cumulative Distribution Function

CHAPTER 1: INTRODUCTION

1.1. Background

Microarray technology is a popular gene expression platform used in the field of genetic, biological and medical research (Macgregor et al., 2002; Petricoin et al., 2002). DNA microarrays allow researchers to simultaneously measure the expression levels of thousands of genes from a single biological sample (Brown and Botstein, 1999) and provide information on each gene. Microarrays help identify genes that are differentially expressed between healthy and non-healthy cells and helps understand the evolution of gene regulation in different organisms (Baldi and Hatfield, 2002; Passador-Gurgel et al.,2007). Also, gene expression technologies are used frequently in molecular biology research to gain a snapshot of transcriptional activity in different tissues or population cells. These techniques also help identify new genes, their expression levels under many conditions. Results using these technologies can be found in pharmaceutical research where it is used to identify drugs candidates, to carry forensic analysis, or evaluating germline mutation in individuals or somatic mutation in cancers. Many types of microarrays have been proposed, including synthetized microarrays (Fodor et al., 1991), spotted microarrays (DeRisi et al., 1996), and oligonucleotide microarrays (Lockhart et al, 1996).

In many experiments, researchers are interested in comparing the gene expressions of multiple treatments to identify genes that are differentially expressed (DE), i.e., genes that exhibit different mean levels across treatments. Statistical methods used by researchers to analyze data from these types of experiments usually do not account for asymmetry in the test statistics (see Storey, J. D. 2002 and Storey, J. D. 2003, for example). Although many of these methods provide good results, methods that takes into account the asymmetry of the effects of the distribution gives better results when asymmetry is present (Megan Orr et al., 2014; Kotoka, 2017). As demonstrated by the results of Orr et *al*. (2014) and Kotoka (2017), methods that into account such pattern can result in higher power for detecting differential expression while still controlling a desired error rate at the nominal level, resulting in a more reliable set of genes for making important biological conclusions. Our objective of this research is to compare methods that take into account the asymmetry of the test statistics when finding DE genes and ultimately make recommendations as to which method to use in different experimental scenarios. This comparison will give scientists guidance when carrying out gene expression analysis when test statistics are observed to have an asymmetric distribution. To our knowledge, research similar to this has never been conducted and, thus, our results can be a meaningful contribution to the science of gene expression analysis.

1.2. Research Objectives and Organization

In the present research document, we carry out differential expression analysis on gene expression data resulting from with two class experiments. The purposes of this research are to:

(1) Develop a local false discovery rate method for analyzing microarray data that takes into account asymmetry in the distribution of the test statistics.

(2) Evaluate and compare the performances of three methods (modified Q-value, Modified SAM and the proposed Asymmetric Local False Discovery Rate) used to determine DE genes when the distribution of the test statistics is asymmetric using simulation studies and analysis of a real microarray data set.

The rest of this thesis is organized as follows. In Chapter 2, we review multiple testing procedures, and statistical methods used in differential expression analysis. Chapter 3 describes the methods used throughout our analysis. Results of the simulation study and real data analysis are covered in Chapter 4. Finally, in Chapter 5 we provide a conclusion to our research and make recommendations for future work.

CHAPTER 2: LITERATURE REVIEW

2.1. Gene Testing

A gene consists of a segment of DNA which codes for a particular protein, the ultimate expression of the genetic information. DNA is converted to messenger RNA (mRNA) through the process of transcription and microarray technology is used to measure the abundance of mRNA, or gene expression, in an organism. Gene testing, also known as differential expression analysis, is a procedure that researchers used to identify genes with differences in expression levels between experimental units in different conditions, groups, or treatments. This process can help identify how some diseases are transmitted (Guillermo Lay-Son et al., 2014: Agatino Battaglia et al., 2013) and to carry out pharmaceuticals research for new drugs. There are two main statistical methods use to perform gene testing: parametric and nonparametric methods.

2.1.1. Parametric methods

When parametric statistical methods are implemented, the data are assumed to follow a specific probability distribution with fixed, but unknown, parameters. The two-sample t-test, which assumes that the data are normally distributed, and its variations are a commonly used parametric methods for detecting DE genes between experimental units in two conditions. P-values for each gene are calculated and a gene is declared to be differentially expressed (DDE) if the corresponding p-values is less than a significance cutoff.

Bayesian and empirical Bayesian methods are other types of parametric statistical methods used to perform gene testing in which parameters are considered random and also follow a probability distribution. P. Baldi and A.D. Long (2001) developed a Bayesian framework for the analysis of microarray expression data based on the t-test. Another hierarchical model was proposed by Newton et al. (2001). Which assumed that the variances for all genes follow one distribution, a Gamma distribution, for example. In this case, a gene is DDE if the calculated posterior odd of change is either significantly large or significantly small. Smyth (2004) proposed an empirical Bayesian method which modified the two-sample t-test called the moderated t-test. He also proposed fitting a linear model to the expression level of every gene for more complicated study designs. In this method, the gene-wise variances are assumed to have a prior Gamma distribution.

2.1.2. Nonparametric methods

Contrary to parametric methods, nonparametric methods do not have distributional on the data. Several nonparametric methods are used for gene testing. A commonly used method is the Wilcoxon rank sum test proposed by Wilcoxon. (1945). The procedure is based on the ranks of the data as opposed to the original observed data. Similar to many parametric statistical methods, the p-values are calculated based on the test statistics and genes are DDE if their corresponding p-value is less than the significance threshold.

Significance Analysis of Microarrays (SAM) is another famous nonparametric method used to test gene from microarray data set developed by Tusher *et al*. (2001). The SAM method measures the strength between gene and the response variable by using repeated permutation of the data. The procedure assigns scores to each gene relative to the standard deviation of repeated measurements. The procedure continues by permuting the scores and for each permutation, calculate the score of each gene in order to create a baseline of scores. The expected null scores are calculated from the permuted data sets. Then, a gene is DDE if the absolute difference between the original score of the data and the expected null score is larger than a specified threshold. Other nonparametric methods have been proposed for gene testing, including the nonparametric t-test and the heuristic method based on high Pearson correlation (Olga G. *et al.* (2002).

2.2. Multiple Hypothesis Testing

The field of genomics has revived interested in multiple testing procedures by raising new methodological and computational challenges (Yongchao Ge et al., 2003). Microarray experiments generate large multiplicity problems in which thousands of hypotheses are tested simultaneously. When performing thousands of hypotheses test simultaneously, errors are committed, and a major concern is to control the rate of errors made. Statistically, there are two types of errors that research can make when performing a hypothesis test: a Type I error which is the incorrect rejection of the true null hypothesis (also known as “false positive” finding) and a Type II error which is incorrectly retaining a false null hypothesis (also known as “false negative” finding) (Peck, Roxy and Jay L. Devore, 2011). See Table 2.1 below. In genomic testing, and in most multiple testing settings in general, researches are more interested in controlling the rate at which Type I errors occur than that of Type II errors.

Table 2.1. *Error Types for multiple testing problem*

Table of error types		Null Hypothesis (H_0) is	
		TRUE	FALSE
Decision About Null Hypothesis (H_0)	Fail to reject	Correct inference (True positive)	Type II error (False Negative)
	Reject	Type I error (False Positive)	Correct inference (True Negative)

Table 2.2. *Microarray data for two independent class experiments (Bentil, 2017)*

	Treatment1				Treatment2			
Gene	Experiment 1	Experiment 2	…	Experiment n	Experiment 1	Experiment 2	…	Experiment n
1	21.0	2.5	…	2.6	1.0	1.5	…	2.6
2	4.8	12.6	…	2.86	23.8	62.6	…	0.86
…	…	…	…	2.98	…	…	…	1.98
m	3.9	29.7	…	5.9	7.9	9.7	…	2.9

The traditional approach to the multiplicity problem calls for controlling the familywise error rate (FWER). The Bonferroni procedure (SIMES, 1986) is the most well-known method for controlling the FWER. Other methods, like Holm's method (Holm, 1979), also control the FWER and sometimes result in more power than the Bonferroni procedure. However, when testing thousands of hypotheses simultaneously, the FWER generally results in extremely low power for identifying DE genes (Benjamini and Hochberg, 1995). The False Discovery rate (FDR) was introduced by Benjamini and Hochberg in 1995 to improve the power of detecting DE genes while still controlling the Type I error rate when simultaneously performing thousands of hypothesis tests.

2.2.1. False discovery rate

Consider the problem of testing *m* null hypotheses, of which m_0 are true (number of true null hypotheses). Additionally, **R** is the number of hypotheses rejected, **V** is the number of null hypotheses rejected from EE genes, and **S** is the number of null hypotheses rejected from DE genes. The following table give a summary of the situation in a simple form.

Table 2.3. *Random Variables Corresponding to the number of Error Committed when testing m hypothesis*

	Declared non-significant	Declared significant	Total
True null Hypotheses	**U**	**V**	m_0
Non-True null Hypotheses	**T**	**S**	$m - m_0$
	$m - \mathbf{R}$	**R**	m

- **U**: Number of true non-discoveries ("true negatives")
- **V:** Number of false discoveries / Type I errors ("false positives")
- **T:** Number of false non-discoveries / Type II errors ("false negatives")
- **S:** Number of true discoveries ("true positives")
- $\boldsymbol{m - m_0}$**:** Number of DE genes or number of False Null Hypotheses
- $m - \mathbf{R}$ **:** Number of non-discoveries ("negatives")

As previously mentioned, Benjamini and Hochberg (1995) introduced the FDR, which controls the proportion of false discoveries (Type I errors) among all discoveries (rejected null hypotheses). Formally, FDR is defined as

$$FDR = E\left(\frac{V}{\max(R, 1)}\right) \tag{2.1}$$

The procedure proposed by Benjamini and Hochberg for testing m null hypotheses while controlling FDR is as follows:

Consider that we want to test $H_{01}, H_{02}, \ldots, H_{0m}$ based on the corresponding p-values $P_1, P_2, \ldots,$ P_m (one p-value / test for each gene). Let $P_{(1)} \leq P_{(2)} \leq \ldots \leq P_{(m)}$ be the ordered p-values and denote $H_{(0i)}$ the null hypothesis corresponding to $P_{(i)}$. Let *k* be the largest *i* for which

$$P_{(i)} \leq \frac{i}{m} q^* \tag{2.2}$$

Reject all $H_{(0i)}$ i = 1, 2, …, k. For independent test statistics from genes with true null hypotheses and for any configuration of false null hypotheses, the above procedure controls the FDR at q^*.

The limit of controlling the FDR was presented by John D. Storey (2002). Instead of controlling the FDR which involves the sequential p-values rejection method based on the observed data, he proposed the positive False Discovery rate (pFDR) which involve the q-value method. The q-value is analogue to the p-values, but it eliminates the need to set the error rate beforehand (Storey, 2002). Storey (2002) proved that the pFDR yields more power than the FDR proposed by (BH) when controlling the Type I error rate. The pFDR is defined as follows:

$$\mathrm{pFDR{=}E}\left(\frac{\mathrm{V}}{\mathrm{R}}|\mathrm{R}{>}0\right). \tag{2.3}$$

The q-value calculation, which estimates the pFDR for each gene, proposed by Storey is as follows:

For each p-value, the corresponding q-value is

$$q_{(i)} = min\left\{\frac{P_{(k)}\widehat{m_0}}{k} : k = i, \dots, m\right\}, \tag{2.4}$$

where:

- $P_{(k)}\widehat{m_0}$: represents the estimate number of false discoveries
- $q_{(i)}$: denote the q-value that corresponds to the ith smallest p-value $p_{(i)}$.
- *k:* represent the total number of genes declared to be DE if genes with p-values less than or equal to P_k are declared to be DE
- $\widehat{m_0}$: represents the estimate of the number of EE genes in a data set. An estimate proposed by Storey (2003) is as follows:

$$\widehat{\mathrm{m}}_0(\lambda) = \frac{\sum_{\mathrm{j}=1}^{\mathrm{m}}\{\mathrm{P_j} > \lambda\}}{(1-\lambda)}, \tag{2.5}$$

where λ is an element of the interval (0, 1). If (2.5) is used to estimate m_0 for any fixed λ in (0, 1), then using q-values to generate lists of significant results will strongly control FDR.

2.2.2. The local false discovery rate

Benjamini and Hochberg's (1995) paper introduced FDR, a particular useful approach to multiple testing. A variant approach of FDR, the local false discovery rate, was proposed by Efron et al., (2001) and Efron and Tibshirani (2002). The local false discovery rate (*l*FDR) is an empirical Bayes technique used to determine the genes that are DE when controlling the number of Type I errors through the FDR.

<u>The setup of the Empirical Bayes techniques:</u>

Suppose we have m null hypotheses to consider simultaneously, each with a corresponding test statistic. The test statistics are calculated using the two-sample t-statistics. For convenience of the Bayesian approach, we transform the t-values to z-values using the following transformation

$$\boldsymbol{z_i = \varphi^{-1}(F_s(t_i))} \tag{2.6}$$

where φ^{-1} is the standard normal cumulative distribution function (cdf)**,** F_s is the cdf of the standard t variable with *s* degree of freedom, t_i the i$^{\text{th}}$ t-value associate with the H_{0i} null hypotheses.

<u>The Bayesian Approach:</u>

Lee et al. (2000), Newton et al. (2001), Efron et al. (2001), underlines the theory: we suppose that the *m* hypotheses are divided into two groups: the genes are either null or non-null and occur with prior probabilities π_0 or $\pi_1 = 1 - \pi_0$ with z-values having density either $f_0(z)$ (which represent the standard normal distribution *N (0, 1)* or $f_1(z)$ (which can be a longer-tailed density yielding z-values further away from 0). The prior probabilities and their associated density of test statistic are given bellow.

$$\pi_0 = Pr\{\text{null}\} \qquad\qquad f_0(z) \text{ density if null}$$

$\pi_1 = Pr\{\text{non-null}\}$ $f_1(z)$ density if non-null

Then the mixture density is:

$$f(z) = \pi_0 f_0(z) + \pi_1 f_1(z) \tag{2.7}$$

By the definition the *l*FDR is:

$$lFDR = Pr\{\text{null} \mid \text{z}\} = \pi_0 f_0(z) \,/\, f(z), \tag{2.8}$$

and the estimated $lFDR$ is given by:

$$\widehat{lFDR} = \frac{\hat{\pi}_0 f_0(z)}{\widehat{f(z)}}. \tag{2.9}$$

Below shows a histogram of z-values for a data set described in Efron (2005).

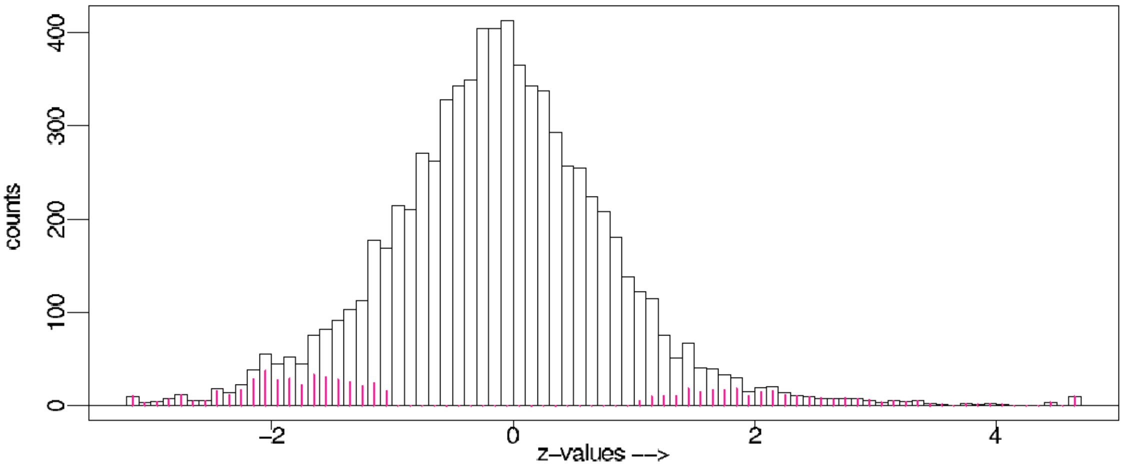


Figure 2.1. *Histogram of the 7680 z-values from HIV microarray experiment. Short vertical lines are estimated "thinned counts" of non-null genes (Efron 2005).*

These z-values are the transformed t-values. The histogram of z-values is shown in Figure 2.1. The normal-shaped central peak presumably charts the largest majority of "null" genes. The long tails reveal "non-null" genes. The method of the local false discovery describes by Efron (2005) is used to provide those thinned counts and estimate the histogram of the non-null z-values. They key assumption of the lFDR estimation is the smoothness of the z-value mixture density $f(z)$ (Efron, 2005). He assumes that, it is useful that the approximation for the distribution of z-values, null or non-null follow a normal distribution with mean μ and standard deviation σ_{μ}^{3}

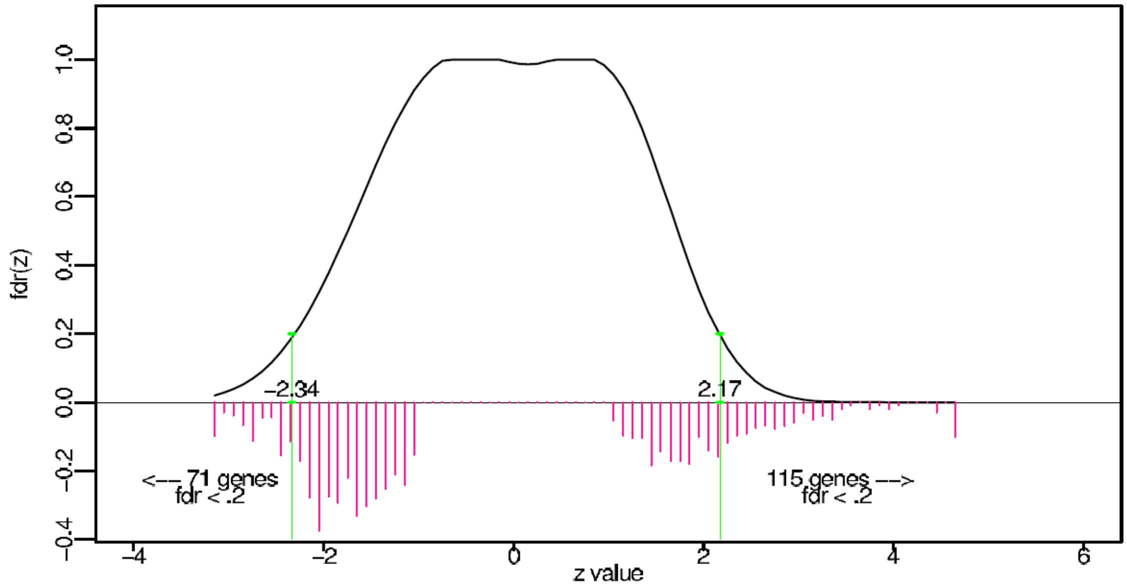


Figure 2.2. *Heavy curve of the local false discovery rate estimated from HIV data (Efron 2005)*

The local false discovery rate is a variant of Benjamini and Hochberg's (1995) false discovery rate. The local nature of FDR is an advantage in interpreting results for individual cases. Figure 2.2 displays the thinned counts from figure 2.1 it also shows the estimated local false discovery rate based on the empirical Bayes method. From figure 2.2 we see that 186 genes having *l*FDR ≤ 0.2. Most of non-null cases lie well within the 0.2 *l*FDR cutoff limits. Efron (2005) prove that the same result is held using Benjamini and Hochberg's FDR procedure with cutoff $q = 0.1$. Large scale testing gives the opportunity of local inference in which gene are judged on their own term and not with respect to the hypothetical possibility of more extreme results. That is the idea of the *l*FDR.

Efron (2005) established the relationship between *l*FDR and FDR. The figure bellow shows the geometrical relationship between FDR and *l*FDR

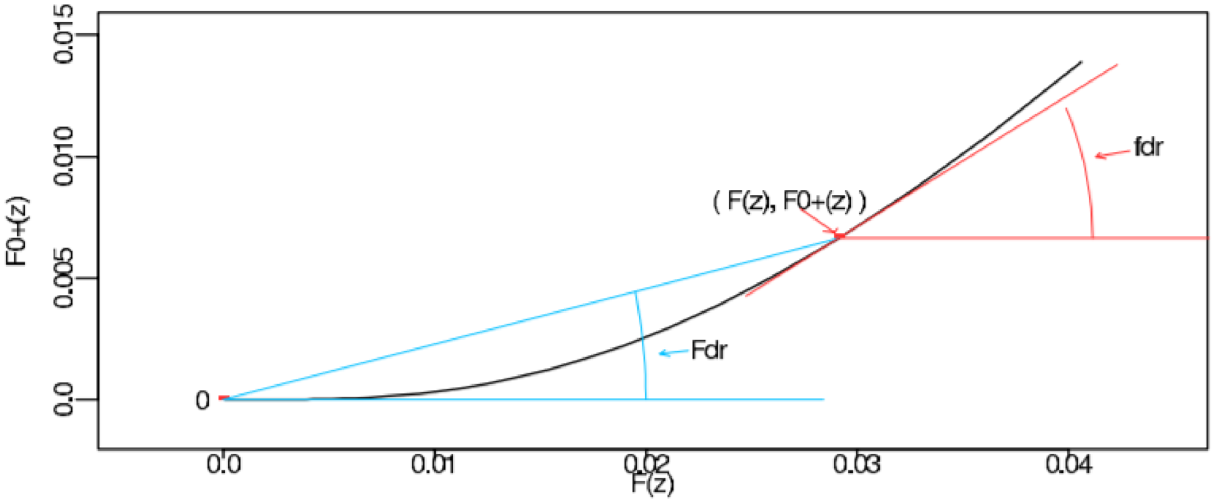


Figure 2.3. *Geometrical relationship of FDR (or Fdr on the graph) to lFDR (or fdr); heavy curve plots* $F_0^+(z)$ *versus F(z) (Efron 2005).*

*l*FDR(z) is the slope of tangent and FDR(z) is the slope of secant. $F_0^+(z)$ *and* $F(z)$ are the cdf's corresponding to $f_0^+(z) = \pi_0 f_0(z)$ and $f(z)$.

CHAPTER 3: METHODS AND MATERIALS

3.1. Methods of Gene Testing

We will consider only the analysis of two group gene expression experiments. For each gene, we want to test the null hypothesis

$$H_p : \mu_{1p} = \mu_{2p} \tag{3.1}$$

against the two-sided alternative. In equation (3.1) μ_{ip} represents the population mean expression value for the p^{th} gene (p = *1, 2*, …, m) in the i^{th} treatment group (i=*1, 2*).

Gene p is EE if H_p is true and if H_p is false, gene p is DE. If we reject, H_p , then the gene p is declared to be differentially expressed (DDE). Using the moderated t-test proposed by Smith (2004) we calculate the test statistic for each gene and the corresponding p-values. This method is used because it borrows information across all genes to more accurately estimate the error variance for each individual gene.

Each gene can be modeled as follows:

$y_{ipl} = \mu_{ip} + \varepsilon_{ipk}$ for $i = 1,2;\ \ p = 1, \dots, m;\ \ and\ \ \ k = 1, \dots, n_i,$

where y_{ipk}, and ε_{ipk} are the expression value and random error for the p^{th} gene from the k^{th} experimental unit in the i^{th} treatment, respectively. Also, ε_{ipk} follows a normal distribution with mean 0 and standard deviation σ_p. The posterior distribution of the population gene-wise variances is given as follows:

$$(\frac{1}{\sigma_p^2} \left| s_p^2, d_0, s_0^2 \right) \sim Gamma\left(\frac{d_0 + d}{2}\ , \frac{d s_p^2 + d_0 s_0^2}{2}\right), \tag{3.2}$$

where d is the degree of freedom associated with s_p^2, the sample pooled variance for gene *p*. Smyth proposed the estimator of σ_p^2 as:

$$\tilde{s}_p^2 = \frac{ds_p^2 + d_0 s_0^2}{2}, \tag{3.3}$$

where d_0 and s_0^2 are the hyper parameters representing prior degrees of freedom and common prior variance, respectively, and are estimated using empirical Bayesian methods

3.1.1. The modified q-values

The q-value method was proposed by Storey (2002) and recently modified by Orr et al. (2014) to take into account the asymmetry of the distribution of the test statistics. Orr et al. proposed that, if the distribution of test statistics in a two class gene expression experiments is asymmetric, the estimation of FDR using the q-value method is improved if this asymmetry is taken into consideration. The method is as follow. For a two class experiment, consider m null hypotheses to be tested as described in Section 3.1. We then calculate the test statistics for each gene with their corresponding two-sided p-value using the moderated t-test. The p-value are then partitioned into two subsets based on the signs of the corresponding test statistics: $\{P_p^1 : p = 1, \ldots , m^{neg}\}$ is the subset of p-values corresponding to the m^{neg} genes with negative test statistics and $\left\{P_p^2 : p = 1, \ldots , m^{pos}\right\}$ is the subset of p-values corresponding to the m^{pos} genes with positive test statistics. Then the corresponding q-values are calculated separately for each gene in each subset as follows:

$$q_p^{(1)} = min\left\{\frac{P_r^{(1)}\,\widehat{m_0}/2}{r} : r = p, \ldots m^{neg}\right\} \tag{3.4}$$

and

$$q_p^{(2)} = min\left\{\frac{P_r^{(1)}\,\widehat{m_0}/_2}{r} : r = p, \dots m^{pos}\right\}. \tag{3.5}$$

Orr et al. showed that this method performs better than the traditional q-value method for detecting DE genes while controlling the FDR when the distribution of the test statistics is asymmetric.

3.1.2. The modified SAM

SAM is one of the common non-parametric methods used to analyze microarray data. This method uses a permutation resampling technique. This method was proposed by Tusher *et al.*, (2001) for determining whether changes in gene expression between classes are statistically significant. Repeated permutations of the data are used to determine if the expression of any gene is significantly related to the class. The use of permutation-based analysis accounts for correlations between genes and avoids parametric assumption about the distribution of expressions of individual genes. SAM estimates FDR for user chosen significance cutoffs to find genes that are DE. This method (SAM) does not account for asymmetry in the distribution of the test statistics.

Motivated by the result of Orr et al. (2014) discussed above, Bentil (2017) proposed a modified SAM method that takes into account the asymmetry of the test statistics when determining DE genes. The method was developed for two class experiment.

An overview of the SAM method for estimating the FDR for a two-class experiment were proposed by Tusher et al. (2001). The modified SAM proposed by Bentil (2017) is outlined by the following steps for estimating FDR for a user defined significance cutoff Δ.

(1) Denote the expression level for the pth gene from the ith experimental unit as $x_{pj}, p = 1,2, \dots m$; $j = 1,2, \dots n$. Compute the test statistic for the pth gene as:

$$d_p = \frac{r_p}{s_p + s_0}, \tag{3.6}$$

where $r_p = \bar{x}_{p2} - \bar{x}_{p1}$, $\bar{x}_p = \sum_j x_{pj}/n$, s_p is the standard error from a traditional pooled two-sample t-test, s_0 is an exchangeability factor, and $\bar{x}_{pi}$ is the sample mean for the *p* gene in treatment *i*.

(2) Sort the test statistics from (1) to get the order statistics, $d_{(1)} \le d_{(2)} \dots \le d_{(m)}$.

(3) Permute the data from the n experimental units B times. For each permutation b compute statistics d_p^{*b} and corresponding order statistics $d_{(1)}^{*b} \le d_{(2)}^{*b} \dots \le d_{(m)}^{*b}$ using the procedure described in steps (1) and (2).

(4) Estimate the expected order statistics by

$$\bar{d}_{(p)} = \frac{1}{B}\sum\nolimits_b d_{(p)}^{*b} \; p = 1,2, \dots m. \tag{3.7}$$

(5) Divide the test statistics into two groups based on the sign of the test statistics. For genes with positive test statistics, i.e. $d_p \ge 0$, for a given Δ^+, genes are said to be significant positive if $d_{(p)} - \bar{d}_{(p)} > \Delta^+$. Define $cut_{up}(\Delta^+)$ as the minimum value of the test statistics $d_{(p)}$ among all significant positive genes.

(6) Calculate the number of falsely positively called genes for each of the B sets of permuted and ordered test statistics:

$$FC^b(\Delta^+) = \sum_{p=1}^{m} I\{d_{(p)}^{*b} > cut_{up}(\Delta^+)\} \tag{3.8}$$

This is the number of EE genes among significant positive genes. We also estimate the median number of falsely positively called genes as

$$med\{FC^b(\Delta^+)\} = median\{FC^b(\Delta^+); b = 1,2, \dots B\}. \tag{3.9}$$

(7) Estimate π_0, the proportion of EE genes in the data set as

$$\hat{\pi}_0 = \frac{\sum_j I\{d_i \in (q25, q75)\}}{0.5m}, \tag{3.10}$$

where $q25$ and $q75$ are the 25th and the 75th percentiles of the permuted d values (there m such values). Note that if $\hat{\pi}_0 > 1$ from (3.10), then it is set to 1. The proportion of up-regulated genes with $d_p \geq 0$ that are EE is:

$$\hat{\pi}_0^+ = m\frac{\hat{\pi}_0/2}{m^{pos}}, \tag{3.11}$$

Where m and m^{pos} represent the total number of genes and the number of gene with positive test statistics, respectively.

(8) The estimate of FDR for genes with positive test statistics is

$$FDR(\Delta^+) = \frac{med\{FC^b(\Delta^+)\}\hat{\pi}_0^+}{Number\ of\ significant\ positive\ genes(\Delta^+)}. \tag{3.12}$$

Steps (5) through (8) are repeated for genes with negative test statistics, i.e. $d_p < 0$. Genes are considered significant negative if $\bar{d}_{(p)} - d_{(p)} > \Delta^-$. The $cut_{low}(\Delta^-)$ is the maximum value of the test statistics $d_{(p)}$ among all significant negative genes. For each of the B sets of permuted and ordered test statistics, the number of falsely negative called genes is calculated as

$$FC^b(\Delta^-) = \sum_{p=1}^{m} I\{d_{(p)}^{*b} < cut_{low}(\Delta^-)\}, \tag{3.13}$$

and the median number of falsely negative called genes is

$$med\{FC^b(\Delta^-)\} = median\{FC^b(\Delta^-); b = 1,2, \dots B\}. \tag{3.14}$$

The estimated proportion of gene with negative test statistics that are EE is

$$\hat{\pi}_0^- = m\frac{\hat{\pi}_0/2}{m^{neg}}, \tag{3.15}$$

where m^{neg} in the number of genes with a negative test statistic. Finally, the estimated FDR in this case is

$$FDR(\Delta^{-}) = \frac{med\{FC^{b}(\Delta^{-})\}\hat{\pi}_0^{-}}{Number\ of\ significant\ negative\ genes(\Delta^{-})}. \qquad (3.16)$$

By taking into account the asymmetry in the test statistics, the original SAM method is modified by setting two values of Δ: Δ^{-} and Δ^{+} and estimate the FDR separately for genes with negative test statistics and genes with negative test statistics.

3.1.3. Asymmetric local false discovery rate

Local false discovery rates, Efron et al. (2001), Efron and Tibshirani (2002), are a variant of Benjamini and Hochberg's (1995) "tail area" false discovery rates. The local false discovery rate as discussed in 2.2.2. assumes the distribution of the test statistic follow a mixture of two distributions: one is the normal distribution with mean 0 and standard deviation equal to 1, the second distribution is also normal but non-null. This method itself does not consider asymmetry of the distribution of the test statistic. In this research, we propose the asymmetric local false discovery rate which takes into account the asymmetry of the test statistics when determining DE genes. We assume the distribution of the test statistic follow a mixture of three normal distributions. Down regulated genes (DRG) have negative test statistics and up regulated genes (URG) have positive test statistics. We use the Expectation Maximization (EM) (Arthur et al. 1977) algorithm to find the (local) maximum likelihood parameter of the mixture model. It is an iterative method to find maximum likelihood or maximum posteriori estimates of parameters. The EM algorithm alternates between performing an expectation (E) step which creates a function for the expectation of the log-likelihood, and the maximization (M) step computes parameters maximizing the expected log-likelihood found in the E step. The distribution of the test statistics can then be expressed as follow:

$$f(z) = \pi_0 f_0(z) \; + \pi_{up} f_{up}(z) + \; \pi_{dw} f_{dw}(z) \tag{3.17}$$

This is simply an extension of the density function proposed by Efron et al. (2001), where π_{up}, and f_{up} represent the proportion and the density of up regulated genes and π_{dw} and f_{dw} the proportion and density of down regulated genes and $f(z)$ is the density of the mixture distribution. These proportion are also estimated during the EM algorithm process. Good estimates of parameters are found when the algorithm converges. The local false discovery rate of the p^{th} gene having a *z-score* is then estimate using the idea proposed by Efron as

$$\widehat{lFDR}_p = \frac{\hat{\pi}_0 f_0(z)}{\widehat{f(z)}} \tag{3.18}$$

From the *l*FDR we estimate the FDR of the p^{th} gene as

$$\widehat{FDR}_p = Mean\{\widehat{lFDR}_p(z)\}. \tag{3.19}$$

where $\widehat{lFDR}_p(z)$ is the set of estimated local false discovery rates less than or equal to $\widehat{lFDR}_p$. A gene is then DDE if the corresponding false discovery rate is less than a predetermined cutoff.

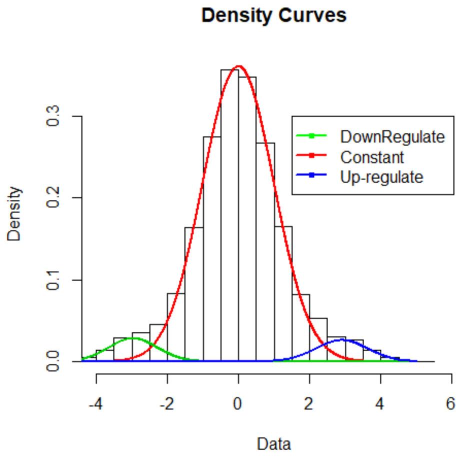


Figure 3.1. *Density Curves of the mix distribution when n=6,m_0=9000 π_A=0.5*

This clearly show the distribution of up regulated gene (blue distribution), down regulated genes (green distribution) and the red distribution which represent EE genes. The algorithm converges after only 65 iterations in this case. Also, when the mean of up-regulated genes and down-regulated genes are close to zero, we consider that normal distribution component as part of the distribution of EE genes.

3.2. Description of Simulation Studies

Two simulation studies were performed to test and compare the three methods.

3.2.1. Simulation using independent normal dataset

To compare the performance of the three methods for identifying DE genes while controlling FDR, gene expression data sets will be simulated using independent normally distributed data (50 data sets in this study). For each dataset, 10,000 genes expression values will be randomly drawn from n experimental units in each of two class treatment, with up and down regulated expression values. The expression values of the pth gene of the kth experimental unit in class i is simulated as

$$y_{ipk} \sim N(\mu_{ip}, \sigma_p^2) \tag{3.20}$$

and

$$\sigma_p^2 \sim Inv\mathbb{\Gamma}(a, b). \tag{3.21}$$

The variance σ_p^2 for each gene was randomly selected from an inverse gamma distribution. The parameters of the inverse gamma distribution were calculated from the dataset of an experiment described in Hannenhalli *et al.* (2006) using the methods proposed by Smyth (2004). The expression values from EE genes were generate from a normal distribution with mean 0 and standard deviation σ_p^2. Thus, if gene p was EE, then $\mu_{1p} = \mu_{2p} = 0$. Expression values from up-regulated genes were randomly generated from $N(\mu_\delta \sigma_p, \sigma_p^2)$ with probability π_{up} and down-regulated genes were randomly generated from $N(-\mu_\delta \sigma_p, \sigma_p^2)$ with probability π_{dw} where π_{up} represents the proportion of genes that are up-regulated and δ represent the effect size. In this first simulation study, $\mu_\delta = 2$. Simulations will be made under different conditions in order to assess the performance of each method. Simulations will be performed with sample sizes of n=4, 6, 10, 12, and 20 and a number of EE genes of m_0= 5000, 7000, and 9000 out of 10000. The proportion of up regulated genes among all DE genes is defined as π_A= $\pi_{up}/(\pi_{up} + \pi_{dw})$. The proportions used in the simulation study are π_A= 0.5, 0.7, and 0.9. This result in 45 different simulation settings.

Note that in this simulation study, it is only necessary to perform simulations when $\pi_A \geq 0.5$ because π_{up} and π_{dw} can be switched without changing the results (i.e., which genes are declared DE) of the gene expression analysis by switching which group is considered the "first group" and which group is considered the "second group".

3.2.2. Simulation using microarray genes expression data set

After simulating data from normal distribution (where each gene values are independent), the second simulation study used real microarray data. This microarray data is from heart tissue of 108 human subjects suffering from idiopathic dilated cardiomyopathy. This data is described by Hannenhalli *et al.* (2006) and is available at the Gene Expression Omnibus with accession number GSE5406 (Hannenhalli *et al* 2006). From this dataset, which contains data from 22283 genes, we randomly select $m = 10000$ genes for analysis in the simulation study. For each simulated data set, two n (n=4, 6, 10, 12, and 20) subjects were randomly drawn from the microarray dataset. At this point, the population group means are equal for each gene because the data from both treatments were randomly drawn from the same population ($\mu_{1p} = \mu_{2p}$). If gene p was EE, the data follow a distribution with a common mean and standard deviation equal S_p^2 meaning that the data are not altered. If gene p was DE, then the effect size δ_p, was randomly chosen from the mixture model $\pi_A N(\delta; \mu_\delta s_{p,} \, s_p^2) + (1 - \pi_A) N(\delta; -\mu_\delta s_{p,} \, s_p^2)$ and this effect size was added to all gene expressions from gene p in experimental units in the second treatment group. In this second simulation study we replace σ_p with s_p because these values are based on a real data set. The data has a complex correlation structure than the first simulation study. The idea is to assess the performance of the methods when the assumptions of normality of the data are not met and when the data have a complex correlation structure. As in the first simulation, we also generate 50 data sets per setting, and perform 45 different simulations for each method. The setting parameters remain the same to assess the performance of each method with no bias.

3.3. Description and analysis of Real data set: Thale cress seedlings

A real microarray data set analyzed by the three methods. This analyzed data is describe in Jang *et al.* (2014). Data were generated from a gene expression experiment analysis of atsf1-2 mutant seedlings for pre-mRNA splicing defect. Genes in thale cress seedling were compare between two genotypes, wild-type and mutant. A total of 6 samples consisting of two treatments: 3 samples each genotype. For each sample, a total of $m = 22810$ gene expression values were recorded. The data set from this experiment is also available at the Gene Expression Omnibus (GEO) with accession number GSE48114.

3.4. Statistical Analysis

For each simulated data set, gene testing was performed using each of the three methods to identify DE genes. Then, for each method, we calculate the mean of the number of DE genes that are DDE (mean of S) and the mean of the proportion of EE genes that are DDE (mean of

V/R). If no genes are DDE, then V/R is set to 0. All analyses were performed in the statistical software R. In both the simulation studies as well as the real data analysis, FDR was controlled at $\alpha = 0.10$ using each method. Thus, if the estimated FDR for a gene was ≤ 0.10, it was DDE.

CHAPTER 4: RESULTS OF SIMULATIONS AND REAL DATA ANALYSIS

4.1. Results

In this chapter we compare the performance of the Modified Q-value, Asymmetric local false discovery rate and the Modified SAM. We also analyze a real microarray data set to compare the number of DDE genes found for each method.

4.1.1. Results of simulation study using independent normal data

Fifty genes expression data sets were randomly generated for each simulation setting. For each dataset, the three methods were used to identify DE genes while controlling FDR at 0.1. For the Modified SAM method, threshold deltas were found corresponding to an estimated FDR closest to but no greater than 0.1 (or 10%). For the other two methods, FDRs were estimated for each gene, and genes with estimated values less than or equal to 0.1 were DDE.

For each simulation setting, for each of the fifty datasets, S (the number of DE genes DDE) was determined and the mean of all the fifty values of S were evaluated when controlling the FDR at 0.1. To determine if all the three methods controlled the FDR at the 10% significance level, the observed FDR, V/R which is the proportion of EE genes among all DDE genes was calculated for each dataset. The mean over the 50 datasets was then taken. This process was executed for all methods.

The table below presents the mean of S and mean of V/R for each simulation setting with corresponding standard errors in parentheses. A bolded value of mean S indicates that the method has a significant mean S. In other words, a bolded value of mean S indicates that, for all the 50 datasets, the method declares a significant amount of DE genes, and if a method outperforms both of the other methods, its mean S is underlined.

Table 4.1. *The mean S for the Modified Q-values, Modified SAM and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for normal simulations data.*

n	m_0	π_A	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
4	9000	0.5	***195.120 (4.583)***	***95.760 (3.293)***	*16.160 (1.424)*
		0.7	***250.120 (4.643)***	***34.980 (10.737)***	*34.080 (1.839)*
		0.9	***357.90 (3.918)***	***104.280 (20.671)***	*79.320 (2.964)*
	7000	0.5	***1842 (6.53)***	***971.960 (4.653)***	*768.940 (23.685)*
		0.7	***1903.200 (7.329)***	*0.000(0.000)*	*1049.680(18.746)*
		0.9	***2123.780(5.563)***	*419.540(102.206)*	***1700.580(13.552)***
	5000	0.5	***4083.400 (7.112)***	*2279.560(3.325)*	***3017.36(93.169)***
		0.7	***4127.040(7.594)***	*851.500(205.222)*	***3385.720(47.165)***
		0.9	***4296.700(5.631)***	*332.040(142.57)*	***4191.140(7.423)***
6	9000	0.5	***643.300(3.462)***	*319.380(1.739)*	***453.160(17.838)***
		0.7	***660.200(3.231)***	*374.380(29.887)*	***523.560(17.887)***
		0.9	***715.860(2.663)***	*456.40 (43.329)*	***612.280(15.511)***
	7000	0.5	***2631.200(3.668)***	*1381.600(2.313)*	***2340.860(34.001)***
		0.7	***2646.360(3.478)***	*1818.600(61.343)*	***2493.900(30.789)***
		0.9	***2703.820(2.811)***	*2304.880(88.01)*	***2673.940(16.135)***

Note: For each simulation setting, the higher mean *S* value at 10% significance level is bolded and the mean S is underlined indicating this method outperformed the other methods.

Table 4.1. *The mean S for the Modified Q-values, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for normal simulations data (continued).*

n	m_0	π_A	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
6	5000	0.5	***4767.680(3.024)***	*2637.340(3.108)*	***4553.900(30.384)***
		0.7	***4768.960(3.12)***	*3515.240(61.243)*	***4638.880(21.094)***
		0.9	***4796.940(2.561)***	*4253.860(125.53)*	***4741.180(10.229)***
10	9000	0.5	***947.460(1.158)***	*479.600 (0.846)*	***947.600(1.182)***
		0.7	***949.960(1.001)***	*0.000(0.000)*	***950.280(1.087)***
		0.9	***956.280(1.035)***	*163.540(49.866)*	***958.520(0.949)***
	7000	0.5	***2966.500(0.93)***	*1551.600(1.196)*	***2969.520(0.875)***
		0.7	***2967.240(0.852)***	*2142.820(1.193)*	***2971.220(0.839)***
		0.9	***2968.940(0.907)***	*2711.740(0.967)*	***2973.160(0.845)***
	5000	0.5	***4984.880(0.545)***	*2759.080(2.118)*	***4985.340(0.557)***
		0.7	***4984.100(0.64)***	*3580.720(16.475)*	***4987.080(0.648)***
		0.9	***4981.640(0.641)***	*4593.740(1.304)*	***4983.140(0.645)***

Table 4.1. *The mean S for the Modified Q-values, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for normal simulations data (continued)*

n	m_0	π_A	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
12	9000	0.5	**982 (0.658)**	495.700(0.538)	**980.900(0.718)**
		0.7	**981.520(0.712)**	0.000(0.00)	**980.400(0.677)**
		0.9	**982.820(0.597)**	36.300(25.405)	**982.440(0.597)**
	7000	0.5	**2990.40(0.472)**	1563.220(0.803)	**2991.580(0.42)**
		0.7	**2990.120(0.512)**	2153.500(1.057)	**2991.340(0.448)**
		0.9	**2989.320(0.459)**	2720.860(0.755)	**2989.960(0.456)**
	5000	0.5	**4996.920(0.23)**	2766.380(1.307)	**4997.240(0.222)**
		0.7	**4996.160(0.272)**	3690.240(9.612)	**4995.080(0.451)**
		0.9	**4994.780(0.41)**	4594.900(1.088)	**4994.360(0.335)**
20	9000	0.5	**645.700(2.92)**	0.000(0.000)	**622.240(2.83)**
		0.7	**654.580(2.72)**	0.000(0.000)	**631.900(2.676)**
		0.9	**704.640 (2.706)**	0.000(0.000)	**696.600(2.848)**
	7000	0.5	**4657.300(3.466)**	1569.07(0.000)	**4653.160(3.263)**
		0.7	**4654.520(3.237)**	2158.893(1.102)	**4658.600(3.477)**
		0.9	**4715.1(2.452)**	2723.786(0.833)	**4748.900(2.561)**
	5000	0.5	**2551.240(3.451)**	**2774.857(0.835)**	2532.440(3.393)
		0.7	**2569.240(3.826)**	**3730.179(1.923)**	2556.680(3.801)
		0.9	**2634.940(3.692)**	**4599.429(1.403)**	2674.100(3.179)

Table 4.2. *The mean V/R for the Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parenthesis for normal simulations data.*

n	m_0	π_A	***Mean V/R***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
4	9000	0.5	0.105 (0.003)	0.088 (0.004)	0.053 (0.008)
		0.7	0.099 (0.003)	0.018 (0.005)	0.037 (0.004)
		0.9	0.103 (0.002)	0.033 (0.007)	0.034 (0.003)
	7000	0.5	0.099 (0.001)	.058 (0.001)	.033 (0.001)
		0.7	0.098 (0.001)	**0.000 (0.000)**	0.044 (0.001)
		0.9	0.099 (0.001)	0.014 (0.004)	0.060 (0.001)
	5000	0.5	0.100 (0.001)	0.001 (0.000)	0.049 (0.003)
		0.7	0.098 (0.001)	0.010 (0.002)	0.066 (0.002)
		0.9	0.098 (0.001)	0.003 (0.001)	0.110 (0.001)
6	9000	0.5	0.088 (0.002)	0.088 (0.002)	0.058 (0.005)
		0.7	0.068 (0.006)	0.060 (0.006)	0.068 (0.005)
		0.9	0.098 (0.002)	0.061 (0.006)	0.073 (0.005)
	7000	0.5	0.100 (0.001)	0.057 (0.001)	0.066 (0.004)
		0.7	0.099 (0.001)	0.064 (0.003)	0.086 (0.005)
		0.9	0.099 (0.001)	0.072 (0.004)	0.116 (0.005)
	5000	0.5	0.101 (0.001)	0.001 (0.000)	0.072 (0.004)

Table 4.2. *The mean V/R for the Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for normal simulations data (continued).*

n	m_0	π_A	***Mean V/R***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
6	*5000*	*0.7*	*0.099 (0.001)*	*0.037 (0.001)*	*0.093 (0.004)*
		0.9	*0.100 (0.001)*	*0.064 (0.003)*	*0.135 (0.003)*
10	*9000*	*0.5*	*0.100 (0.001)*	*0.087 (0.002)*	*0.103 (0.001)*
		0.7	*0.098 (0.001)*	**0.000 (0.000)**	*0.102 (0.001)*
		0.9	*0.099 (0.002)*	*0.074 (0.022)*	*0.110 (0.001)*
	7000	*0.5*	*0.100 (0.001)*	*0.055 (0.001)*	*0.108 (0.001)*
		0.7	*0.100 (0.001)*	*0.072 (0.001)*	*0.118 (0.001)*
		0.9	*0.101 (0.001)*	*0.088(0.001)*	*0.143 (0.001)*
	5000	*0.5*	*0.099 (0.001)*	*0.001 (0.000)*	*0.101 (0.001)*
		0.7	*0.100 (0.001)*	*0.021 (0.002)*	*0.128 (0.001)*
		0.9	*0.101 (0.001)*	*0.077 (0.000)*	*0.164 (0.001)*
12	*9000*	*0.5*	*0.100 (0.001)*	*0.087 (0.001)*	*0.101 (0.001)*
		0.7	*0.099 (0.001)*	**0.000 (0.000)**	*0.102 (0.001)*
		0.9	*0.100 (0.002)*	*0.016 (0.011)*	*0.109 (0.002)*

Table 4.2. *The mean V/R for the Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for normal simulations data (continued).*

n	m_0	π_A	***Mean V/R***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
12	*7000*	*0.5*	*0.098 (0.001)*	*0.056 (0.001)*	*0.107 (0.001)*
		0.7	*0.100 (0.001)*	*0.073 (0.001)*	*0.116 (0.001)*
		0.9	*0.099 (0.001)*	*0.089 (0.00)*	*0.139 (0.001)*
	5000	*0.5*	*0.098 (0.001)*	***0.000 (0.000)***	*0.104 (0.001)*
		0.7	*0.100 (0.001)*	*0.033 (0.002)*	*0.119 (0.002)*
		0.9	*0.100 (0.001)*	*0.078 (0.000)*	*0.159 (0.001)*
20	*9000*	*0.5*	*0.099 (0.001)*	***0.000 (0.000)***	*0.097 (0.002)*
		0.7	*0.099 (0.001)*	***0.000 (0.000)***	*0.101 (0.001)*
		0.9	*0.098 (0.001)*	***0.000 (0.000)***	*0.107 (0.001)*
	7000	*0.5*	*0.101 (0.001)*	*0.057 (0.001)*	*0.103 (0.001)*
		0.7	*0.098 (0.001)*	*0.074 (0.001)*	*0.113 (0.001)*
		0.9	*0.098 (0.001)*	*0.091 (0.000)*	*0.141 (0.001)*
	5000	*0.5*	*0.101 (0.001)*	*0.001(0.000)*	*0.101 (0.001)*
		0.7	*0.101 (0.001)*	*0.04 (0.000)*	*0.108 (0.001)*
		0.9	*0.101 (0.001)*	*0.0800 (0.000)*	*0.129 (0.001)*

For these simulations, the FDR was controlled at the 10% significance level. We see that as the sample size increased, the power of detecting DE genes increased for each of the three methods. Also, the number of DE genes detected increased as the number of EE genes decreased.

From Table 4.1. we see that for $n = 4$ and $n = 6$, the Modified Q-value performed better than both the Asymmetric Local False Discovery and the Modified SAM in all 18 simulation settings regarding mean S. For $n = 10$ the Modified SAM performed better than both the Modified Q-value and the Asymmetric Local False Discovery Rate in all 9 simulation settings regarding mean S. However, the results from the Modified Q-value are very similar to those of the Modified SAM. Finally, for $n = 12$ and $n = 12$, the Modified Q-value perform better than the other two methods in 9 simulations following by the Modified with 5 simulations and the Asymmetric Local False Discovery rate with 4 simulations out of 18. In total, out of 45 simulations, the Modified Q-value perform better than other methods in 27 simulations followed by the Modified SAM with 14 simulations and the Asymmetric Local False Discovery Rate with 4 simulations.

As shown in Table 4.2. the observed FDR (mean V/R) was comparable among the three methods for each simulation. The mean V/R shows that the observed FDR was controlled at or close to 10% for all methods. Moreover, the Modified Q-value most closely controlled the FDR at 10% in most simulation settings. So, in this first simulation the Modified Q-values perform the best in term of controlling the FDR at 10%. Even though the Asymmetric Local False discovery generally controlled the FDR at 10%, it is very conservative when $m_0 = 5000$ for any value of sample size. For example when $(n = 4,\ m_0 = 5000,\ \pi_A = 0.5\,;0.7)$; $(n = 6,\ m = 5000,$ $\pi_A = 0.5\,;0.7)$; $(n = 10,\ m_0 = 5000,\ \pi_A = 0.5\,;0.7)$; $(n = 20,\ m_0 = 5000,\ \pi_A = 0.5;0.7)$ the mean of V/R is ranged between 0.001 and 0.04. This resulted in fewer DDE genes. On the other hand, the Modified SAM is anti-conservative with mean V/R exceeding the 0.12 in many settings. For example, when $n = 10,\ m_0 = 5000,\ \pi_A = 0.9$ we have V/R equal to 0.143 also when $n = 10,\ m_0 = 7000,\ \pi_A = 0.9$ V/R is equal to 0.164. These two cases show how anti-conservative the Modified SAM is for some setting.

4.1.2. Results of the simulations using real microarray dataset

The results of the simulations using microarray data are shown in the following tables.

Table 4.3. *The mean S for the Modified Q-values, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulations data.*

n	m_0	π_A	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
4	*9000*	*0.5*	***251.960(9.593)***	*57.880 (15.436)*	***134.780 (15.265)***
		0.7	***277.420(7.862)***	*147.240 (28.342)*	***139.500 (11.44)***
		0.9	***332.380(7.264)***	*241.78 (38.465)*	***198.300 (11.77)***
	7000	*0.5*	***1478.00(24.562)***	*802.160(21.564)*	***1071.820 (57.404)***
		0.7	***1515.380(23.575)***	*780.280(81.253)*	***1184.660 (58.29)***
		0.9	***1661.100 (21.318)***	*561.420(105.437)*	***1479.74 (53.078)***
	5000	*0.5*	***3136.06(27.658)***	*1720.820(30.809)*	***2924.020(73.618)***
		0.7	***3198.780(26.678)***	*1811.680(168.06)*	***3154.3400(64.852)***
		0.9	***3404.440(23.745)***	*1035.020(215.714)*	***3719.840(19.454)***

Table 4.3. *The mean S for the Modified Q-values, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulations data (continued).*

n	m_0	π_A	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
6	*9000*	*0.5*	***544.720 (5.328)***	*134.440(19.827)*	***562.180 (12.716)***
		0.7	***545.300 (6.747)***	*132.620 (29.679)*	***561.660 (14.48)***
		0.9	***581.780 (6.737)***	*196.000 (41.516)*	***600.280 (14.468)***
	7000	*0.5*	***2050.140 (11.191)***	*1083.54 (23.053)*	***2063.480 (32.936)***
		0.7	***2076.960 (11.197)***	*1103.16 (99.019)*	***2117.780 (29.297)***
		0.9	***2172.140 (11.086)***	*1151.480 (140.067)*	***2248.080 (22.58)***
	5000	*0.5*	***3880.680 (12.689)***	*2097.560 (29.21)*	***3950.800(13.03)***
		0.7	***3911.660 (13.452)***	*2016.280 (192.341)*	***4007.140 (14.551)***
		0.9	***4031.440 (13.855)***	*2124.100 (258.673)*	***4231.460 (13.939)***

Note: π_A's in these Simulation tables represent the proportion of DE genes that are up-regulated. So, $\pi_A = 0.7$ mean that 70% of DE genes are up-regulated and the remaining 30% are down-regulated. In our simulation study we consider three possible value of π_A for each value of m_0. The different possible values of π_A are 0.5, 0.7, and 0.9.

Table 4.3. *The mean S for the Modified Q-values, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulations data (continued).*

$\boldsymbol{n}$	$\boldsymbol{m_0}$	$\boldsymbol{\pi_A}$	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
6		*0.9*	***4031.440(13.855)***	*2124.000(258.673)*	***4231.460(13.939)***
10	*9000*	*0.5*	***746.540 (2.938)***	*182.440(26.655)*	***766.580 (6.079)***
		0.7	***752.900 (2.955)***	*74.140 (26.692)*	***764.46 (7.118)***
		0.9	***768.240 (3.095)***	*118.140(38.695)*	***781.220 (6.901)***
	7000	*0.5*	***2473.040(5.75)***	*1281.120 (8.981)*	***2474.740 (15.02)***
		0.7	***2482.600 (5.779)***	*1141.260(116.159)*	***2499.880(13.603)***
		0.9	***2530.720(5.887)***	*1315.400(160.007)*	***2564.740(11.06)***
	5000	*0.5*	***4365.220(7.286)***	*2330.000 (52.402)*	***4388.780(5.625)***
		0.7	***4382.400(7.235)***	*1961.960(209.381)*	***4420.400(5.07)***
		0.9	***4438.840(6.774)***	*2387.14(279.161)*	***4536.12 (6.484)***
12	*9000*	*0.5*	***787.500(2.76)***	*103.280(24.934)*	***793.060(6.43)***
		0.7	***796.280(2.407)***	*197.060(39.316)*	***810.360 (4.506)***
		0.9	***807.080(2.115)***	*270.080(51.51)*	***840.740(4.008)***
	7000	*0.5*	***2563.720(5.529)***	*1254.200(69.773)*	***2573.600(7.957)***
		0.7	***2561.960(4.423)***	*1414.560(108.26)*	***2570.560(9.281)***
		0.9	***2597.920(4.829)***	*1887.100(216.568)*	***2633.920(6.036)***

Table 4.3. *The mean S for the Modified Q-values, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulations data (continued).*

n	m_0	π_A	***Mean S***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
12	*5000*	*0.5*	***4446.380(6.834)***	*2315.540(54.858)*	***4460.660(5.611)***
		0.7	***4461.16(7.852)***	*2735.375(288.401)*	***4494.380(6.221)***
		0.9	***4497.660(5.705)***	*3200.880(235.598)*	***4580.400(5.011)***
20	*9000*	*0.5*	***869.100 (1.611)***	*78.160(18.887)*	***876.540(3.207)***
		0.7	***871.120 (1.922)***	*110.460 (26.811)*	***880.860(3.121)***
		0.9	***877.820 (1.918)***	*161.260 (37.332)*	***887.200 (2.929)***
	7000	*0.5*	***2712.920 (3.231)***	*1036.020 (48.076)*	***2717.760(5.322)***
		0.7	***2717.940 (3.428)***	*1354.440 (72.085)*	***2726.420(5.417)***
		0.9	***2735.140 (3.33)***	*1575.660(111.805)*	***2751.760(5.424)***
	5000	*0.5*	***4632.460(4.869)***	*2088.740(31.296)*	***4642.860(4.386)***
		0.7	***4638.740(4.42)***	*2800.420(74.754)*	***4657.980(4.118)***
		0.9	***4658.880(4.367)***	*3308.660(160.713)*	***4711.78(4.155)***

Table 4.4. *The mean V/R for the Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulation data.*

n	m_0	π_A	***Mean V/R***		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
4	*9000*	*0.5*	*0.084 (0.018)*	*0.054 (0.022)*	*0.068 (0.017)*
		0.7	*0.062 (0.011)*	*0.137 (0.031)*	*0.035 (0.009)*
		0.9	*0.067 (0.011)*	*0.116 (0.025)*	*0.035 (0.009)*
	7000	*0.5*	*0.094 (0.010)*	*0.061 (0.007)*	*0.062 (0.011)*
		0.7	*0.095 (0.010)*	*0.044 (0.008)*	*0.071 (0.012)*
		0.9	*0.097 (0.009)*	*0.028 (0.007)*	*0.09 (0.012)*
	5000	*0.5*	*0.082 (0.008)*	*0.001 (0.000)*	*0.074 (0.008)*
		0.7	*0.082 (0.007)*	*0.022 (0.003)*	*0.090 (0.008)*
		0.9	*0.082 (0.006)*	*0.015 (0.003)*	*0.136 (0.007)*
6	*9000*	*0.5*	*0.095 (0.016)*	*0.043 (0.011)*	*0.115 (0.018)*
		0.7	*0.0988 (0.020)*	*0.084 (0.021)*	*0.130 (0.023)*
		0.9	*0.100 (0.019)*	*0.080 (0.023)*	*0.138 (0.023)*
	7000	*0.5*	*0.077 (0.10)*	*0.053 (0.011)*	*0.095 (0.012)*
		0.7	*0.080 (0.01)*	*0.036 (0.007)*	*0.101 (.012)*
		0.9	*0.082 (0.009)*	*0.024 (0.004)*	*0.112 (0.011)*
	5000	*0.5*	*0.088 (0.008)*	*0.003 (0.001)*	*0.098 (0.008)*
		0.7	*0.089 (0.008)*	*0.018 (0.003)*	*0.105 (0.008)*
		0.9	*0.090 (0.007)*	*0.022 (0.003)*	*0.135 (0.008)*

Table 4.4. *The mean V/R for the Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulation data (continued).*

n	m_0	π_A	*Mean V/R*		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
10	9000	0.5	0.116 (0.019)	0.067 (0.018)	0.158 (0.021)
		0.7	0.120 (0.02)	0.051 (0.019)	0.152 (0.021)
		0.9	0.123 (0.020)	0.054 (0.019)	0.156 (0.021)
	7000	0.5	0.096 (0.011)	0.047 (0.005)	0.109 (0.012)
		0.7	0.094 (0.011)	0.024 (0.004)	0.125 (0.013)
		0.9	0.108 (0.010)	0.026 (0.004)	0.139 (0.012)
	5000	0.5	0.116 (0.008)	0.003 (0.001)	0.123 (0.008)
		0.7	0.116(0.008)	0.014 (0.002)	0.131 (0.008)
		0.9	0.115 (0.007)	0.026 (0.004)	0.162 (0.008)
12	9000	0.5	0.087 (0.015)	0.048 (0.017)	0.133 (0.019)
		0.7	0.113 (0.019)	0.142 (0.029)	0.156 (0.021)
		0.9	0.114 (0.018)	0.093 (0.019)	0.153 (0.019)
	7000	0.5	0.119 (0.015)	0.049 (0.023)	0.131 (0.015)
		0.7	0.090 (0.01)	0.030 (0.003)	0.106 (0.011)
		0.9	0.115 (0.013)	0.035 (0.005)	0.148 (0.014)
	5000	0.5	0.094 (0.009)	0.002 (0.001)	0.099 (0.008)
		0.7	0.095 (0.010)	0.016 (0.002)	0.11 (0.01)
		0.9	0.093 (0.007)	0.032 (0.003)	0.137 (0.008)

Table 4.4. *The mean V/R for the Modified Q-value, Modified SAM, and Asymmetric Local False Discovery Rate methods with associated standard errors in parentheses for microarray simulation data (continued).*

n	m_0	π_A	*Mean V/R*		
			Modified Q-value	***Asy. Local FDR***	***Modified SAM***
20	*9000*	*0.5*	*0.150 (0.021)*	*0.053 (0.016)*	*0.136 (0.018)*
		0.7	*0.085 (0.017)*	*0.093 (0.023)*	*0.127 (0.018)*
		0.9	*0.087 (0.017)*	*0.057 (0.015)*	*0.132 (0.018)*
	7000	*0.5*	*0.111 (0.013)*	*0.059 (0.016)*	*0.126 (0.014)*
		0.7	*0.112 (0.013)*	*0.046 (0.006)*	*0.129 (0.014)*
		0.9	*0.112 (0.012)*	*0.038 (0.004)*	*0.143 (0.014)*
	5000	*0.5*	*0.102 (0.010)*	*0.003 (0.001)*	*0.109 (0.009)*
		0.7	*0.100 (0.009)*	*0.027 (0.002)*	*0.115 (0.009)*
		0.9	*0.099 (0.008)*	*0.044 (0.004)*	*0.144 (0.008)*

The Modified Q-value and Modified SAM have high mean S for all 45 simulation settings. Moreover, the Modified SAM outperformed the Modified Q-value in 36 simulations out of 45 settings when $n = 6, 10, 12, and\ 20$. For $n = 4$, the Modified Q-value outperformed the Modified SAM. The Asymmetric Local False Discovery Rate poorly performed in all simulation settings compared to the other two methods. The inability of the Asymmetrical Local Discovery Rate to detect DE gene is due to the fact that the method is parameter sensitive, meaning that if the effect size is high, the normalEM algorithm will not converge regularly even though the E and M steps of the algorithm are executed, thus will not detect DE genes.

As shown in Table 4.4. the observed FDR (mean V/R) was comparable among the three methods for each simulation. The mean V/R shows that the observed FDR was controlled at or close to 10% for all methods. As in the first simulation study, we see that the Modified Q-values control the FDR better than the other methods, and the Asymmetric Local False Discovery Rate remains very conservative and the Modified SAM anti-conservative with values of V/R close to 0.001 and higher than 0.12, respectively.

The results from both simulations shows that the Modified Q-value and the Modified SAM have higher mean S when controlling the FDR at 10% compared to the Asymmetric Local False Discovery Rate. For sample size $n = 4,$ the Modified Q-value performed better in both simulation studies. Moreover, the Modified Q-value performed better than the other two methods when the data were simulated from independent normal distributions. But in the case of microarray dataset, the Modified SAM outperformed the other two methods in terms of mean S when the sample size is greater than 4 but did not adequately control FDR.

4.2. Real Data

4.2.1. Results of real data analysis

In this section we analyze the data described in Section 3.3 using the Modified Q-value, the Modified SAM and the Asymmetric Local False Discovery Rate. The data set is divided into two classes (Wild-type and Mutant); with a total of 3 samples for each class. The first class has 3 samples and the second has 3 as well. The dataset contains $m = 22810$ genes. Figure 4.1 shows the empirical distribution of the test statistics values for the 22810 genes.

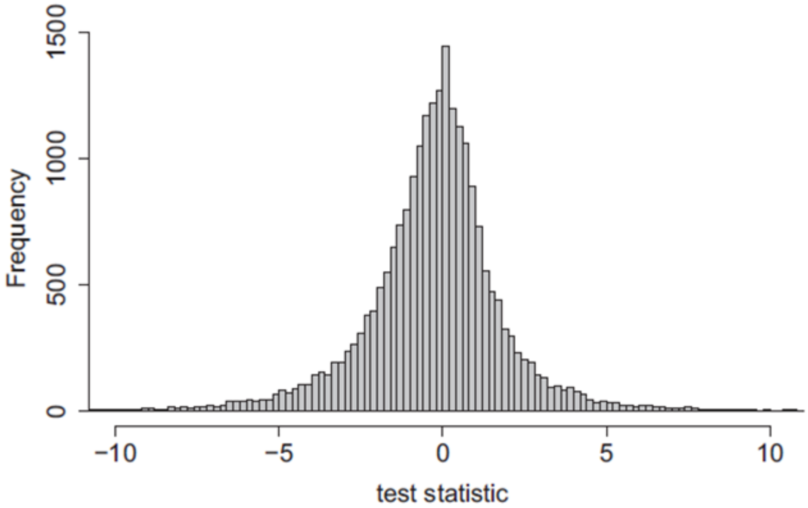


Figure 4.1. *Histogram of the test statistics from an experiment describe in Jang et al. (2014) in which microarray genes where used to examine the transcriptome profile in the atsf1-2 mutant and identified genes of which transcript levels were changed significantly*.

The Histogram of these test statistics does not clearly indicate an asymmetry in the distribution of test statistics. But when calculating the number of positive and negative test statistics, it is visible that there are more genes with negative test statistics than genes with positive test statistics, more precisely, there are $m^{neg} = 12355$ genes with negative test statistics and $m^{pos} = 10455$ genes with positive test statistics.

For Modified Q-value method, when using Storey and Tibshirani's (2003) natural cubic spline method, the estimated number of EE genes in this experiment is $\widehat{m}_0 = 17968.98$. We expect the EE genes to have the same number of positive and negative test statistics. Thus, the estimate number of EE genes with positive test statistics is $\widehat{m}_0$/2 = 8984.49 and the estimate number of EE genes with negative test statistics in also $\widehat{m}_0$/2 = 8984.49. We then estimate the number of DE genes with positive effect size as $10455 - 8984 = 1471$ genes and the number of DE genes with negative effect size as $12355 - 8984 = 3371$ genes. This result to $1471/(1471 + 3371) \simeq$ 30.38% of DE genes with positive effect sizes and 69.61 % of DE genes with negative effect sizes. So, to find genes that are declared to be DE, we partition the q-values in two subsets as follow:

$p = 1, \dots, 12355$

$$q_p^{(1)} = min\left\{\frac{P_r^{(1)}(8984.49)}{r} : r = p, \dots 12355\right\} \qquad (4.1)$$

for genes with negative effect size and

$p = 1, \dots, 10455$

$$q_p^{(2)} = min\left\{\frac{P_r^{(2)}(8984.49)}{r} : r = p, \dots 10455\right\} \qquad 4.2)$$

for genes with positive effect size.

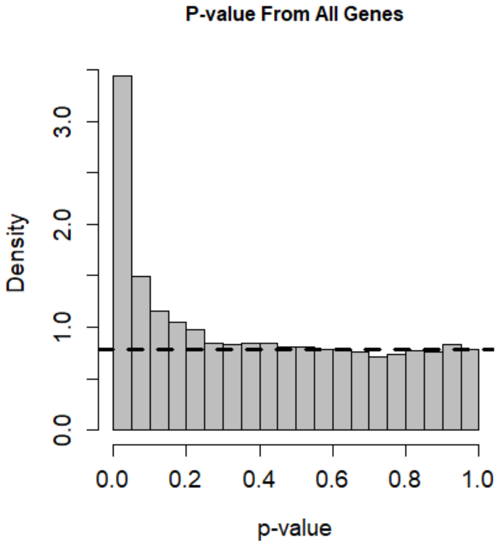
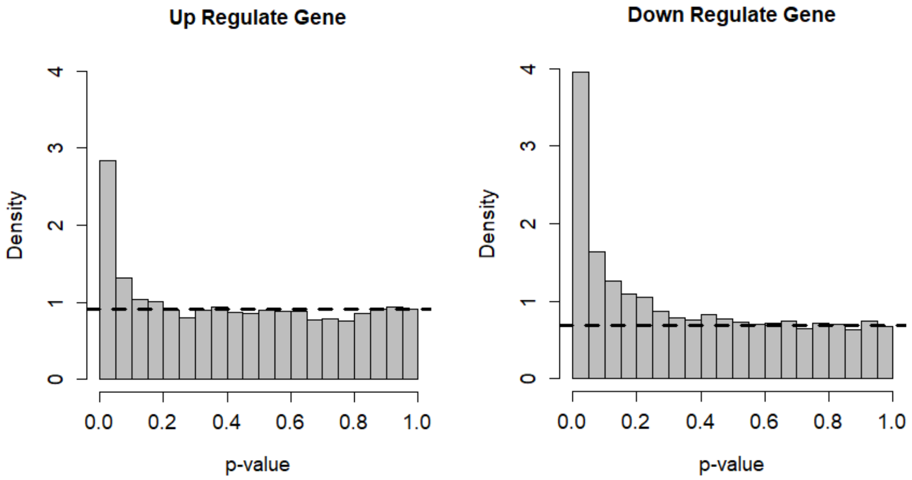


Figure 4.2. *Distribution of p-values.*



(a) (b)

Figure 4.3. *Distribution of p-value for up-regulated genes (a) and down-regulated genes (b)*

Finally, the number of gene declared to be DE by the Modified Q-value is: 1684 genes at 10% significance level. Figure 4.3 shows the distribution of subset of p-value when the gene are up-regulated and down-regulate. In the Modified SAM method, when using the method describe in section 3.1.2, we found that 1919 gene are DDE. The Asymmetric local False discovery rate uses the test statistics calculated by the Storey method and we consider the one side p-value to get z-values. Below is the distribution of the mix distribution.

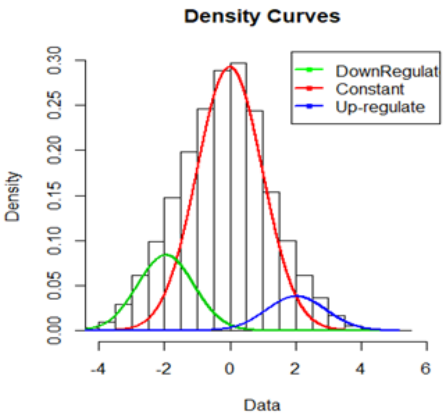


Figure 4.4. *Density Curves of the mix distribution for the real microarray dataset.*

The number of gene declared to be DE using Modified q-value, Asymmetric Local False Discovery Rate and the Modified SAM at 10% significant level are summarized in Figure 4.5. For all three methods combine there were 908 genes that were declared DE. Additionally, 153 genes were DDE by the Modified Q-values method and Asymmetric Local False Discovery rate. Also, there are 1807 genes DDE by Asymmetric Local False Discovery Rate. Finally, there are 70 genes DDE by the Modified SAM and the Modified Q-values.

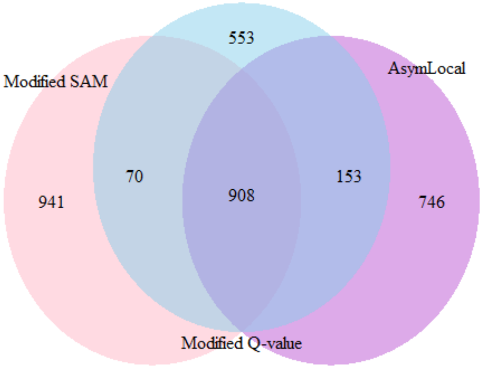


Figure 4.5. *Venn diagram of genes declared to be DE for the Modified SAM, Modified Q-values and The Asymmetric Local False Discovery Rate method.*

The analysis was performed on real gene expression data and not on simulated data set, which make it impossible to determine which genes are EE and which genes are DE. Which make it difficult to evaluate the FDR associate with each method. But from the results of both simulation study, the FDR is being adequately controlled at 10%.

CHAPTER 5: CONCLUSION RECOMMENDATION AND FUTURE WORK

5.1. Conclusion

The primary focus of this paper was to compare the performance of the Modified q-value (Orr *et al.* 2014), Modified SAM (Bentil 2017), and the proposed Asymmetric Local False Discovery rate in terms of detecting DE genes and controlling the FDR. The performance of these methods was evaluated using simulated and real microarray datasets with two independent treatments. All three methods consider the asymmetry in the distribution of the test statistics.

The Modified q-value method outperformed the Modified SAM and the Asymmetric local False discovery rate in terms of mean S in the first simulation study for $n = 4, 6$. When the sample size increased, the Modified SAM outperformed the Modified q-value in a few simulation settings. The Asymmetric Local false discovery rate exhibited a lower power for detecting DE genes in general but outperformed the Modified q-value when the sample size was equal to 20. Overall, the Modified q-value declare more genes DE than other methods in the first simulation study.

When simulating data sets from real microarray data, we saw that for sample size $n = 4$, the Modified q-value outperformed the other two methods. However, as the sample size increased, the Modified SAM outperformed the Asymmetric Local False discovery rate and the Modified q-value in terms of power, but not FDR control. As in the first simulation study, the Asymmetric Local False discovery rate showed a low power of detecting genes that are DE and conservative control of FDR.

In the case of real gene expression data analysis (in this study the sample size is 3) the Modified SAM declared more gene to be DE than the Modified Q-value and the Asymmetric Local False Discovery Rate at 10% significance level.

5.2. Recommendations

After the analysis of simulated data sets and data from a real microarray gene expression experiment, the following recommendations are hereby made.

(1) The Modified q-value is recommended for analysis in gene expression experiment with small sample sizes, say $n \leq 10$ because it had comparable power compared to modified SAM, Asymmetric Local False Discovery Rate and better FDR control.

(2) Except in the cases where the sample size in less than 10, and regarding the results of our simulation, it is recommended to use the Modified SAM to analyze gene expression, also when there is a noticeable high correlation among the measurements of each gene.

5.3. Future Work

(1) Given that the Asymmetric Local false discovery had generally low power, and regarding the fact that it is sensitive to parameter values when generating data, investigation into the properties of this method are recommended.

(2) Develop other methods for identifying DE genes that take into account asymmetry.

(3) RNAseq is a new technology used in gene expression analysis. Comparing methods that takes into account asymmetry of the distribution of the test statistics for RNAseq experiments.

REFERENCES

Efron, B., Tibshirani, R., Storey, J. D., & Tusher, V. (2001). Empirical Bayes Analysis of a Microarray Experiment. *Journal of the American Statistical Association,96*(456), 1151-1160. doi:10.1198/016214501753382129

Baldi, P. a. (2002). DNA Microarrays and Gene Expression: From Experiment to data Analysis and Modeling. Cambrige: Cambridge University Press.

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

Birsoy K, B. R. (2011 Nov: 138 (21):). Analysis of gene networks in white adipose tissue development reveals a role for ETS2 in adipogenesis. *Development* , 4709-19.

Brown, P. O. (1999). Exploring the new world of the genome with DNA microarrays. *Nat Genet, 21(1Suppl),* 33-37.

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

Derisi, J. L. (1997). Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale. *Science,278*(5338), 680-686. doi:10.1126/science.278.5338.680.

Efron, B. (2005). Local False Discovery Rates. *Large-Scale Inference,*70-88. doi:10.1017/cbo9780511761362.006

Ekua, F. K. (2017). Identification of Differentially Expressed Genes when the distribution of effect sizes is asymmetric in two class Experiments. 15-21.

Pascale F. Macgregor, Jeremy A. Squire, (2002). Application of Microarrays to the Analysis of Gene Expression in Cancer. *Clinical Chemistry, 48(8),* 1170-1177.

Fodor, S., Read, J., Pirrung, M., Stryer, L., Lu, A., & Solas, D. (1991). Light-directed, spatially addressable parallel chemical synthesis. *Science,251*(4995), 767-773. doi:10.1126/science.1990438.

Lay-Son, G., Espinoza, K., Vial, C., Rivera, J. C., Guzmán, M. L., & Repetto, G. M. (2015). Chromosomal microarrays testing in children with developmental disabilities and congenital anomalies. *Jornal De Pediatria (Versão Em Português),91*(2), 189-195. doi:10.1016/j.jpedp.2014.07.007.

Lockhart, D. J., Dong, H., Byrne, M. C., Follettie, M. T., Gallo, M. V., Chee, M. S., . . . Brown, E. L. (1996). Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology,14*(13), 1675-1680. doi:10.1038/nbt1296-1675.

Hannenhalli, S. (2006). Transcriptional Genomics Associates FOX Transcription Factors With Human Heart Failure. *Circulation,114*(12), 1269-1276. doi:10.1161/circulationaha.106.632430.

Newton, M., Kendziorski, C., Richmond, C., Blattner, F., & Tsui, K. (2001). On Differential Variability of Expression Ratios: Improving Statistical Inference about Gene Expression Changes from Microarray Data. *Journal of Computational Biology,8*(1), 37-52. doi:10.1089/106652701300099074

Troyanskaya, O. G., Garber, M. E., Brown, P. O., Botstein, D., & Altman, R. B. (2002). Nonparametric methods for identifying differentially expressed genes in microarray data. *Bioinformatics,18*(11), 1454-1461. doi:10.1093/bioinformatics/18.11.1454.

Orr, M., Liu, P., & Nettleton, D. (2012). Estimating the Number of Genes That Are Differentially Expressed in Both of Two Independent Experiments. *Journal of Agricultural, Biological, and Environmental Statistics,17*(4), 583-600. doi:10.1007/s13253-012-0108-8.

Orr, M., Liu, P., & Nettleton, D. (2014). An improved method for computing q-values when the distribution of effect sizes is asymmetric. *Bioinformatics,30*(21), 3044-3053. doi:10.1093/bioinformatics/btu432.

Passador-Gurgel, G., Hsieh, W., Hunt, P., Deighton, N., & Gibson, G. (2007). Quantitative trait transcripts for nicotine resistance in Drosophila melanogaster. *Nature Genetics,39*(2), 264-268. doi:10.1038/ng1944.

Peck R., Devore J. (2011). Statistics : The exploration & analysis of Data. Nelson Education.

Petricoin, E. F., Hackett, J. L., Lesko, L. J., Puri, R. K., Gutman, S. I., Chumakov, K., . . . Sistare, F. D. (2002). Medical applications of microarray technologies: A regulatory science perspective. *Nature Genetics,32*(Supp), 474-479. doi:10.1038/ng1029.

Pounds, S., & Morris, S. W. (2003). Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics,19*(10), 1236-1242. doi:10.1093/bioinformatics/btg148.

Simes, R. J. (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika,73*(3), 751-754. doi:10.1093/biomet/73.3.751.

Smyth, G. K. (2004). Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. *Statistical Applications in Genetics and Molecular Biology,3*(1), 1-25. doi:10.2202/1544-6115.1027.

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genomewide studies. *Proceedings of the National Academy of Sciences,100*(16), 9440-9445. doi:10.1073/pnas.1530509100.

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B (Statistical Methodology),64*(3), 479-498. doi:10.1111/1467-9868.00346.

Efron, B., & Tibshirani, R. (2002). Empirical bayes methods and false discovery rates for microarrays. *Genetic Epidemiology,23*(1), 70-86. doi:10.1002/gepi.1124

Tusher, V. G., Tibshirani, R., & Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences,98*(9), 5116-5121. doi:10.1073/pnas.091062498.

Wilcoxon, F. (1945). Individual Comparisons by Ranking Methods. *Biometrics Bulletin,1*(6), 80. doi:10.2307/3001968.

Ge, Y., Sealfon, S. C., & Speed, T. P. (2009). Multiple testing and its applications to microarrays. *Statistical Methods in Medical Research,18*(6), 543-563. doi:10.1177/0962280209351899.

APPENDIX. R CODE

```
source(″https://bioconductor.org/biocLite.R″)
biocLite(″limma″)
biocLite(″samr″)
library(limma)
library(qvalue)
library(pscl)
library(mixtools)
library(samr)
library(impute)
ni       ### number of samples per treatment
m0      ## number of EE genes
m      ## total number of genes
m1     ## number of DE genes
piup  ### proportion of up regulated genes
pi0   #proportion of EE gene
### Defining the variance
d0=3.658156
s20=0.04039051
Sds <- sqrt(rigamma(n = m, alpha = d0/2, beta = (d0*s20)/2))
nup <- round(m1*piup) ##number of up-regulated DE genes
ndn <- m1 - nup ##number of down-regulated DE genes
#######################################################################
```

```
##############      Simulating the data sets         #################
######################################################################


Patern <- function(x){

dat <- matrix(NA, nrow = m0, ncol = 2*ni) #matrix of EE genes

for(i in 1 : m0) {

sdi <- Sds[i]

dati <- rnorm(n =2*ni, mean = 0, sd = sdi)

dat[i,] <- dati

}

dat1 <- matrix(NA, nrow = m1, ncol = 2*ni)   ### matrix of DE genes

means <- c(rep(2,nup),rep(-2,ndn))                   ### defining means for up
and down regulated genes.

for(i in 1 : m1) {

sdi <- Sds[m0 + i]

dat1i <- rnorm(n = ni, mean = 0, sd = sdi)  # data for

dat2i <- rnorm(n = ni, mean = means[i]*sdi, sd = sdi)  # data for down
regulated genes

datai <- c(dat1i, dat2i)

dat1[i,] <- datai

}

Data <- rbind(dat,dat1)#### THis contains the first 9000 EE genes data and the
second 1000 (up and down regulated gene)

}

dataset <- lapply(1:50,patern)### Number of simulated data sets
```

```
#####################################################################
#############    MODIFIED Q-VALUE METHOD       ######################
#####################################################################

RVSVR1 <- data.frame()

for(i in 1 : length(dataset[i])) {

dataset = dat77

TS <- as.data.frame(dataset)

trt <- as.factor(c(rep(1,ni), rep(2,ni))) ### User define (depending on the
number of treatments per group)


design <- model.matrix(~trt+0)           ##design matrix

colnames(design)=c("t1","t2")             ## Name the colums of the design
matrix

contr.mat <- makeContrasts(t2-t1, levels=design) ##contrasts of interest

##Perform moderated t-test (Smyth, 2004)

fit1 <- lmFit(TS,design)

fit2 <- contrasts.fit(fit1,contr.mat)

fit3 <- eBayes(fit2)

ts <- fit3$t                     ### test statistics ( these are positiv and
negetives)

ps <- fit3$p.value[,1]           ### p-values

Zvalue <- scale(ps, center = TRUE, scale = TRUE) ### getting the z-value from
the p-values

source("estimate_qvalues_asymmetric.R")

qvsnew <- qval_asymm(ts, ps)
```

```
R1 <-sum(qvsnew <= 0.1)                  ### Number of genes DDE

V1 <- sum(qvsnew[1:m0] <= 0.1)          ### Number of EE genes DDE

S1 <- sum(qvsnew[(m0 + 1) : m] <= 0.1) ### Number of DE genes DDE

VR1 <- V1 / max(R1,1)

RVSVR1 <- rbind(RVSVR1,c(S1, VR1))      ### S and VR using ... (2 subsets of
p-values)

colnames(RVSVR1) <- c("S", "VR")

}

## The mean and standard errors of S and V/R

MStd1 <- round(apply(RVSVR1, 2, function(x) c(mean(x), sqrt(var(x) /
length(x)))),digits = 3)

MStd1

#######################################################################

############## Asymmtric local false discovery rate method ##########

#######################################################################

RVSVR2 <- data.frame()

#for(i in 1 :length(dataset)) {

TS <- as.data.frame(dataset)

trt <- as.factor(c(rep(1,ni), rep(2,ni))) ### User define (depending on the
number of treatments per group)

design <- model.matrix(~trt+0)           ##design matrix

colnames(design)=c("t1","t2")             ## Name the colums of the design
matrix

contr.mat <- makeContrasts(t2-t1, levels=design) ##contrasts of interest

##Perform moderated t-test (Smyth, 2004)
```

```
fit1 <- lmFit(TS,design)

fit2 <- contrasts.fit(fit1,contr.mat)

fit3 <- eBayes(fit2)

ts <- fit3$t                      ### test statistics ( these are positiv and
negetives)

ps <- fit3$p.value[,1]           ### p-values

#### spliting the p-values for one side p-values

pNeg <- ps[ts <= 0]/2

pPos <- 1-ps[ts > 0]/2

Pvalue <- c(pNeg,pPos)

Z1.znorm <- qnorm(Pvalue)  ### transforming the p-values to z-scores

hist(Z1.znorm)

#### Check at the mixdristribution and estimating the parameters or the
proportion.

set.seed(104)                                          # Settint the seed at
104

mixmdl = normalmixEM(Z1.znorm, k=3, arbvar=TRUE, mean.constr = c(0,″a″, ″-c″),
sigma = 1,

sd.constr = c(1, ″b″,″d″),  arbmean = TRUE, maxit = 30000)   ## Looking at the
distribut

plot(mixmdl, which=2)

mixmdl[c(″lambda″,″mu″,″sigma″)]                          ## Estimating the
proportion and the parameters of Mix model

#print(TAT)

summary(mixmdl)                                      ## Summary of normalmixEM
```

```
P_pi <- mixmdl$lambda                              ## vector containing the
proportions estimate

Means <- mixmdl$mu                                 ## Vector containing the
means of the mixed distribution

Sigmas <- mixmdl$sigma                             ## Vector containing the
standard deviations

## Estimate of fo and f

f0est1 <- P_pi[1]*dnorm(Z1.znorm)

f1est1 <- P_pi[2]*dnorm(Z1.znorm, Means[2], Sigmas[2])

f2est1 <- P_pi[3]*dnorm(Z1.znorm, Means[3], Sigmas[3])

if (abs(Means[2])<=0.1 |abs(Means[3]<= 0.1)) {

f0est = f0est1 + f1est1 + f2est1

} else if (abs(Means[2])<=0.1 | abs(Means[3])> 0.1) {

f0est = f0est1 + f1est1


} else if (abs(Means[3]<= 0.1) | abs(Means[2]> 0.1)){

f0est = f0est1 + f2est1


} else {

f0est = f0est1

}

Fest  <- f0est1 + f1est1 + f2est1

lcfdr <- f0est/Fest

func1 <- function(v){

FDR1 <- c()
for(i in 1:length(v)){
FDR1[i] <- mean(v[v<=v[i]])
}
FDR1
}
FDR = func1(lcfdr)
#FDR
R2 <-sum(FDR <= 0.1)                  ### Number of genes DDE
V2 <- sum(FDR[1:m0half] <= 0.1)          ### Number of EE genes DDE
S2 <- sum(FDR[(m0half + 1) : m] <= 0.1) ### Number of DE genes DDE
VR2 <- V2 / max(R2,1)
RVSVR2 <- rbind(RVSVR2,c(S2, VR2))      ### S and VR using ...  (2 subsets of
p-values)
colnames(RVSVR2) <- c(″S″, ″VR″)
}
# The mean and standard errors of S and V/R
MStd1 <- round(apply(RVSVR1, 2, function(x) c(mean(x), sqrt(var(x) /
length(x)))),digits = 3)
MStd1
##########################################################
#############   SAMseq Method   ##########################
##########################################################
# Estimation of s0 and calculation of the test statistics di
Sam.fdr = function(dataset){
```

```
Cut.lw.f = c()

Cut.up.f = c()

delta.neg = c()

delta.pos = c()

for(i in 1 :length(dataset)) {

dat= dataset[[i]]

test.func <- function(dat, ni){

dat1 <- dat[,1:ni]       # data for the first grroup

dat2 <- dat[,(ni+1):(2*ni)]  # data for the secomd group

# Sample means for each group

X1bar <- apply(dat1, 1, mean)   # mean of each gene from the first group

X2bar <- apply(dat2, 1, mean)

ri = X1bar - X2bar

S21 = apply(dat1, 1, var)        # Sample variances for each group ( down and
up regulated)

S22 = apply(dat2, 1, var)

Si = sqrt(((ni-1)*S21 + (ni-1)*S22)/(2*ni-2))*sqrt(2/ni)   # Pooled standard
errors

# Computation of S0

# 1) Let Sa be alpha percentile of the Si values. Let dia = ri/(Si + Sa)

S0s = quantile(Si, probs = seq(0, 1, by = 0.05))  ### this is the 100
quantiles of the Si(i denote q1,...q100)

Siord = order(Si)

lowcut = seq(1, 9901, 100)

hicut = seq(100, 10000, 100)
CVs = rep(NA, 21)

for (p in 1:21) {

S0p = S0s[p]

dip = ri/(Si+S0p)

madp = rep(NA, 100)

for(b in 1:100){

inds = Siord[lowcut[b]:hicut[b]]

dipb = dip[inds]

medpb = median(dipb)

madp[b] = median(abs(dipb-medpb))/ 0.64

}

CVs[p] = sd(madp)/mean(madp)

minCVind = order(CVs)

S0 = S0s[order(CVs)[1]]

di = ri/(Si + S0) # test statistic

or.di <- sort(di,decreasing=FALSE)

rk.di <- rank(di)

return(list(di = di, ordered.di = or.di, rank.di= rk.di, S0 = S0))

      }

}

testS = test.func(dat, ni)

S0 = testS$S0

di = testS$di
```

```
or.di = testS$ordered.di

rank.di=testS$rank.di

######### Step 2 Compute the order statistics #######################

y <- c(rep(1,ni),rep(2,ni))

#############permuted test statistics###################

insert.value <- function(vec, newval, pos) {    # this function is used to
insert a new value in the vector if necessary

if (pos == 1)                              # if the position of the new value
is the lowest then it will be placed at the bigining

return(c(newval, vec))

lvec <- length(vec)

if (pos > lvec)

return(c(vec, newval))

return(c(vec[1:pos - 1], newval, vec[pos:lvec]))

}

# Compute the matrix of permutations

permutes <- function(elem) {

# generates all perms of the vector elem

if (!missing(elem)) {

if (length(elem) == 2)

return(matrix(c(elem, elem[2], elem[1]), nrow = 2))

last.matrix <- permute(elem[-1])                          # function permute (
need to know what package has it)

dim.last <- dim(last.matrix)

new.matrix <- matrix(0, nrow = dim.last[1] * (dim.last[2] +
1), ncol = dim.last[2] + 1)
for (row in 1:(dim.last[1])) {
for (col in 1:(dim.last[2] + 1)) new.matrix[row +
(col - 1) * dim.last[1], ] <- insert.value(last.matrix[row,
], elem[1], col)
            }
return(new.matrix)
      }
else cat("Usage: permute(elem)\n\twhere elem is a vector\n")
}
getperms <- function(y, B) {              # nperms or B is the number of
permutations requested to estimate the false discovery rate
total.perms = factorial(length(y))
if (total.perms <= B) {
perms = permutes(1:length(y))
all.perms.flag = 1                         # Where all possible permutation
are used
B.act = total.perms                   # Number of permutation actually used.
Will be < nperms  : nperms.act
}
if (total.perms > B) {
perms = matrix(NA, nrow = B, ncol = length(y))   #
for (i in 1:B) {
perms[i, ] = sample(1:length(y), size = length(y))  #
}
```

```
all.perms.flag = 0
B.act = B
}
return(list(perms = perms, all.perms.flag = all.perms.flag,
B.act = B.act))
}
perm = getperms(y,100)
perms=perm[['perms']]
nperms = dim(perms)[1]
di.mat = matrix(NA, nrow = nperms, ncol = m)
test.p.func <- function(dat, ni) {
for (p in 1:100) {
datp = dat[, perm[['perms']][p, ]]  # permuted data
dat1p <- datp[, 1:ni]       # data for the first grroup
dat2p <- datp[, (ni + 1):(2 * ni)]  # data for the secomd group
#dat2
# Sample means for each group
X11bar <- apply(dat1p, 1, mean)   # mean of each gene from the first group
X22bar <- apply(dat2p, 1, mean)
rip = X11bar - X22bar
# Sample variances for each group ( down and up regulated)
S21p = apply(dat1p, 1, var)
S22p = apply(dat2p, 1, var)
```

```
# Pooled standard errors

Sip = sqrt(((ni - 1) * S21p + (ni - 1) * S22p) / (2 * ni - 2)) * sqrt(2 / ni)

## test statistics is di

dips = rip / (Sip + S0) # test statistic

#print(length(dip))

di.mat[p,] = sort(dips, decreasing = FALSE)

            }

return(di.mat = di.mat)

      }

##################  ordered permuted test statistis ################

                  testS.p.ord = di.mat =test.p.func(dat, ni)

####################### Step 3 continue ##########################

# From the B permutations ,estimate the expected order statistics

# expected ordered statistics

#di.bar1 <- apply(or.testS.p, 1, mean)

#di.bar1 <- di.bar1[length(di.bar1):1]

di.bar1 <- apply(testS.p.ord, 2, mean)

################## step5 ################

# plot di versus di.bar

plot(di, di.bar1)

############### step 6  find the possible deltas value  ####################

di.ord = sort(di)

res.mat <- data.frame(di.ord = di.ord, evo = di.bar1, dif1 = di.ord - di.bar1,
dif2 = di.bar1-di.ord) # this is a data frame with di, dibar and dif
```

```
res.pos <- res.mat[res.mat$di.ord > 0, ] # this is a data frame with positiv
di and the corresponding dif

res.neg <- res.mat[res.mat$di.ord < 0, ] # this is a data frame with negative
di and the corresponding dif

################ Up regulated genes ################################

################ estimate pi0s   ######################################

pis <- function(di.mat, di, m){

qq <- quantile(di.mat, c(0.25, 0.75))

pi0h <- sum((di >= qq[1]) & (di <= qq[2]))/(0.5 * length(di))

npos.di <- sum(di >= 0)  ##number of genes with positive test statistic (up
regulated)

nneg.di <- sum(di < 0)  ##number of genes with negative test statistic (down
regulated)

pi0hpos <- (pi0h*m/2)/npos.di  ##estimate of proportion of EE genes with
positive test statistics

pi0hneg <- (pi0h*m/2)/nneg.di  ##estimate of proportion of EE genes with
negative test statistics

return(list(pi0h = pi0h, pi0hpos = pi0hpos, pi0hneg = pi0hneg))

}

# Estimated pis

pis <- pis(di.mat, di, m)

print(pis)

pi0h = pis$pi0h

m0hat = m*pi0h

print(m0hat)
```

```
del.pos = matrix(seq(0.5, 2, by=0.01), nrow=1, ncol = 151)     ## the best value of delta so far is 0.8

#del.pos

FDR.fun = function(del.pos){

fdr1 = c()

for (l in 1:151){

sig.pos = res.pos$dif1 > del.pos[,l]      # Significan positive genes

Nsig.pos = sum(sig.pos)                   # number of significan possitive up regulated genes

Cut.up = min(res.pos$di.ord[sig.pos == TRUE])     #Cut up

nfc.pos = rep(NA, nperms)             # number of false called  positive gene

for (k in 1:nperms) {

dips.pos = di.mat[k,]

nfc.pos[k] = sum(dips.pos >= Cut.up)

}

nfc.pos

med.nfc.up = median(nfc.pos)

fdr.pos = (pis$pi0hpos)*med.nfc.up / Nsig.pos

fdr1[l] = fdr.pos

}

return(fdr1)

}

########### Down regulated genes #######################
```

```
del.neg = matrix(seq(0.5, 2, by=0.01), nrow=1, ncol = 151)     ## the best
value of delta so far is 0.8

#del.neg

FDR.fun.1 = function(del.neg){

fdr2 = c()

for (t in 1:151){

sig.neg = res.neg$dif2 > del.neg[,t]         # Significan positive genes

Nsig.neg = sum(sig.neg)                     # number of significan negative or
down  regulated genes

Cut.lw = max(res.neg$di.ord[sig.neg == TRUE])     #Cut low

nfc.neg = rep(NA, nperms)             # number of false called  negative gene

for (k in 1:nperms) {

dips.neg = di.mat[k,]

nfc.neg[k] = sum(dips.neg <= Cut.lw)

}

nfc.neg

med.nfc.lw = median(nfc.neg)

fdr.neg = (pis$pi0hneg)*med.nfc.lw / Nsig.neg

fdr2[t] = fdr.neg

}

return(fdr2)

}

fdr.pos = FDR.fun(del.pos)

delta.pos[i] <- min(del.pos[fdr.pos <= 0.1 & is.na(fdr.pos)== FALSE])
```

```
#delta.pos

sig.pos.f = res.pos$dif1 > delta.pos[i]      # final Significan positive genes

#Nsig.pos.f = sum(sig.pos.f)                   # number of significan
possitive up regulated genes

Cut.up.f[i] = min(res.pos$di.ord[sig.pos.f == TRUE])     #Cut up

fdr.neg = FDR.fun.1(del.neg)

delta.neg[i] <- min(del.neg[fdr.neg <= 0.1 & is.na(fdr.neg)== FALSE])

#delta.neg

sig.neg.f = res.neg$dif2 > delta.neg[i]         # Significan positive genes

#Nsig.neg.f = sum(sig.neg.f)                   # number of significan negative
or down  regulated genes

Cut.lw.f[i] = max(res.neg$di.ord[sig.neg.f == TRUE]) #Cut low

Delta.c = cbind(delta.pos, delta.neg, Cut.lw.f, Cut.up.f)

}

return(Delta.c)

Delta = Sam.fdr(dataset)

Delta

######################################################################

##############   Calculation of S, V, V/max(R,1)   ###################

######################################################################

#### Calculating the test statistics and put them in a list ###########

S.VR.fun = function(dataset){

DSi <- list(c())

for(i in 1 :length(dataset)) {
```

```
dat= dataset[[i]]

test.func <- function(dat, ni){

dat1 <- dat[,1:ni]       # data for the first grrOup

dat2 <- dat[,(ni+1):(2*ni)]  # data for the secomd group

# Sample means for each group

X1bar <- apply(dat1, 1, mean)   # mean of each gene from the first group

X2bar <- apply(dat2, 1, mean)

ri = X1bar - X2bar

# Sample variances for each group ( down and up regulated)

S21 = apply(dat1, 1, var)

S22 = apply(dat2, 1, var)

# pooled standard errors

Si = sqrt(((ni-1)*S21 + (ni-1)*S22)/(2*ni-2))*sqrt(2/ni)

# Computation of S0

# 1) Let Sa be alpha percentile of the Si values. Let dia = ri/(Si + Sa)

S0s = quantile(Si, probs = seq(0, 1, by = 0.05))  ### this is the 100
quantiles of the Si(i denote q1,...q100)

Siord = order(Si)

lowcut = seq(1, 9901, 100)

hicut = seq(100, 10000, 100)

CVs = rep(NA, 21)

for (p in 1:21) {

S0p = S0s[p]

dip = ri/(Si+S0p)
madp = rep(NA, 100)

for(b in 1:100){

inds = Siord[lowcut[b]:hicut[b]]

dipb = dip[inds]

medpb = median(dipb)

madp[b] = median(abs(dipb-medpb))/ 0.64

}

CVs[p] = sd(madp)/mean(madp)

minCVind = order(CVs)

S0 = S0s[order(CVs)[1]]

## test statistics is di

di = ri/(Si + S0) # test statistic

or.di <- sort(di,decreasing=FALSE)

rk.di <- rank(di)

}

return(di)

}

testS = test.func(dat, ni)

di = testS

DSi[[i]] <- di

}

return(DSi)

}
```

```
DS = S.VR.fun(dataset)

sam <- function(di, Cut.lw.f, Cut.up.f){


di1 <- di[1:m0]

di2 <- di[(m0+1):10000]

DDE1 <- di1 <= Cut.lw.f | di1 >= Cut.up.f

DDE2 <- di2 <= Cut.lw.f | di2 >= Cut.up.f

V1 <- sum(DDE1)

S1 <- sum(DDE2)

R1 = S1 + V1

VR1 = V1 / max(R1, 1)

return(cbind(S1, VR1))

}

RVS = t(sapply(c(1:dim(Delta)[1]), function(i){

return(sam(DS[[i]], Delta[i,][″Cut.lw.f″], Delta[i,][″Cut.up.f″]))}))

colnames(RVS) = c(″S″,″VR″)

RVS

#S = mean(RVS[,1])     # Number of DE gene DDE

#VR = mean(RVS[,2])

MStd3 <- round(apply(RVS, 2, function(x) c(mean(x), sqrt(var(x) /
length(x)))),digits = 3)

MStd3
```