Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data

Juan Pablo Acosta
Industrial Engineer
Code: 1.077.032.887

Universidad Nacional de Colombia
Faculty of Sciences
Statistics Department
Bogotá, D.C.
May, 2015

Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data

Juan Pablo Acosta
Industrial Engineer
Code: 1.077.032.887

Thesis to qualify for the degree
Master of Statistics

Advisor
Liliana López–Kleine, Ph.D.
Mathematics

Research line
Statistics

Universidad Nacional de Colombia
Faculty of Sciences
Statistics Department
Bogotá, D.C.
May, 2015

Title in English

Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data.

Título en español

Estrategia para la Identificación Multivariada de Genes Diferencialmente Expresados en Datos de Expresión Génica.

Abstract: Microarray technology has become one of the most important tools in understanding genetic expression in biological processes. As microarrays contain measurements of thousands of genes' expression levels across multiple conditions, identification of differentially expressed genes will necessarily involve data mining or large scale multiple testing procedures. To the date, advances in this regard have either been multivariate but descriptive, or inferential but univariate.
In this work, we present a new multivariate inferential analysis method for detecting differentially expressed genes in microarray data. It estimates the *positive false discovery rate* ($pFDR$) using artificial components close to the data's principal components, but with an exact interpretation in terms of differential gene expression. Our method works best under very common assumptions and gives way to a new understanding of genetic differential expression in microarray data. We provide a methodology to analyse time course microarray experiments and some guidelines for assessing whether the required assumptions hold. We illustrate our method on two publicly available microarray data sets.

Resumen: Los microarreglos de ADN se han convertido en una de las herramientas más importantes para entender la expresión genética en procesos biológicos. Como cada microarreglo contiene mediciones del nivel de expressión de miles de genes en múltiples condiciones, la identificación de genes diferencialmente expresados involucrará necesariamente minería de datos o pruebas de hipótesis múltiples a gran escala. Hasta hoy, avances en este campo han sido o bien multivariados pero descriptivos, o bien inferenciales pero univariados.
En este trabajo, presentamos un nuevo método inferencial y multivariado para identificar genes diferencialmente expresados en microarreglos de ADN. Estimamos la *tasa positiva de falsos positivos* ($pFDR$) utilizando componentes artificiales cercanos a los componentes principales de los datos, pero con una interpretación exacta en términos de expresión génica diferencial. Nuestro método funciona mejor bajo algunos supuestos muy comunes y da lugar a un nuevo entendimiento de la expresión diferencial en datos de microarreglos. Planteamos una metodología para analizar microarreglos con múltiples puntos en el tiempo y damos guías heurísticas para determinar si los supuestos necesarios se cumplen en una determinada base de datos. Ilustramos nuestro método con dos bases de datos públicas de microarreglos de ADN.

Keywords: Microarrays, false discovery rate, principal components analysis, bootstrap.

Palabras clave: Microarreglos de ADN, tasa de falsos positivos, análisis en componentes principales, bootstrap.

Acceptation Note

Thesis Work
AP

Jury
José Rafael Tovar Cuevas

Jury
Leonardo Trujillo Oyola

Advisor
Liliana López–Kleine

Bogotá, D.C., May 2015

Acknowledgements

Special thanks go to Silvia Restrepo, Thibaut Jombart, Rosa Montaño and Francisco Torres for valuable comments. To Liliana for her guidance and encouragement. To my family for reminding me that fun is just as important as work. And to the Consulado Popular for granting me so many free hours to get this done.

Contents

Contents **III**

List of Tables **V**

List of Figures **VII**

1. Introduction **1**

2. Theoretical Framework **3**

2.1 Background on DNA expression and microarrays 3

2.2 General probability model for microarray data 5

2.3 Principal Components Analysis 6

2.3.1 PCA mechanics 6

2.3.2 A word on interpretation 8

2.4 Bootstrap 8

2.4.1 Bootstrap estimates: One sample case 8

2.4.2 Bootstrap confidence intervals 10

2.4.2.1 Percentile intervals 10

2.4.2.2 BCa intervals 11

2.4.3 Permutation and bootstrap hypothesis tests: Two sample case 12

2.4.3.1 Permutation tests 13

2.4.3.2 Bootstrap hypothesis tests 14

2.4.3.3 Selection of the test statistic 14

2.5 Multiple hypothesis testing 15

2.5.1 Some definitions 15

2.5.1.1 Family-Wise Error Rate 16

2.5.1.2 False Discovery Rate (FDR) 17

2.5.1.3 Positive False Discovery Rate (pFDR) . . . 18
2.5.2 Estimation of the pFDR under independence . . . 19
2.5.3 The q–value . . . 21
2.5.4 A word on dependence . . . 22
2.5.5 Choice of the null hypothesis . . . 23
2.6 A word on scale . . . 23

3. Proposed Strategy **25**
3.1 Artificial components . . . 25
3.2 Single time point analysis . . . 26
3.2.1 Estimation . . . 27
3.2.2 Assumptions . . . 28
3.2.3 Further assessments . . . 29
3.3 Time course analysis . . . 30
3.3.1 Active vs. supplementary time points . . . 30
3.3.2 Groups conformation through time . . . 31
3.4 Microarray data sets . . . 32
3.4.1 Tomato inoculated with *P. Infestans* (PI) in the field . . . 32
3.4.2 *Arabidopsis thaliana* inoculated with *A. tumefaciens* (AT) . . . 32

4. Results **35**
4.1 Tomato plants inoculated with *P. infestans* . . . 35
4.1.1 Differentially expressed genes . . . 37
4.1.2 Time course analysis . . . 38
4.1.3 Comparison with other methods . . . 40
4.2 *Arabidopsis thaliana* inoculated with *A. tumefaciens* . . . 41
4.2.1 Differentially expressed genes . . . 42
4.2.2 Comparison with other methods . . . 44
4.3 A word of caution . . . 45

5. Conclusions and Future Perspectives **47**

A. Appendix **51**
A.1 Differentially expressed genes in the PI data set . . . 51
A.2 Functions in R . . . 57

Bibliography **61**

List of Tables

2.1 Possible outcomes when testing n hypothesis simultaneously. 16

3.1 Biological, technical and probabilistic assumptions. 28
3.2 Data 60 hai from tomato plants inoculated with *P. infestans*. 32
3.3 Data 48 hai from *Arabidopsis thaliana* inoculated with *A. tumefaciens*. . . . 33

4.1 Inertia ratios for the PI data set. 36
4.2 Comparison with Cai et al. (2013) for PI microarray data set 60 hai. 40
4.3 Inertia ratios for the AT data set. 41
4.4 Comparison with Ditt et al. (2006) for AT microarray data set 48 hai. 44
4.5 Heuristic guidelines for assessing Biological Scenario 2. 45

A.1 Down regulated genes in the PI data set 60 hai. 51
A.2 Up regulated genes in the PI data set 36 hai. 55
A.3 Up regulated genes in the PI data set 60 hai. 56

List of Figures

2.1 Scanned image of a microarray after hybridization. 4

4.1 Inertia ratios $R(\mathbf{v}_2)$ for the PI microarray data set. 36
4.2 Artificial components for the PI microarray data set 60 hai. 36
4.3 Estimated $pFDR$ for the PI microarray data set 60 hai. 37
4.4 Differentially expressed genes in the PI microarray data set 60 hai. 37
4.5 Active vs supplementary time points analysis for the PI microarray data set. 38
4.6 Group conformation through time for the PI microarray data set. 39
4.7 Comparison with Cai et al. (2013) for PI microarray data 60 hai. 40
4.8 Inertia ratios $R(\mathbf{v}_2)$ for the AT microarray data set. 42
4.9 Artificial components for the AT microarray data set 48 hai. 42
4.10 Estimated $pFDR$ for the AT microarray data set 48 hai. 43
4.11 Differentially expressed genes in the AT microarray data set 48 hai. 43
4.12 Comparison with Ditt et al. (2006) for AT microarray data 48 hai. 44

CHAPTER 1

Introduction

Microarray technology has become one of the most important tools in understanding gene expression in biological processes (Yuan & Kendziorski, 2006). Since its development in the 1990s, an enormous amount of data has become available and new statistical methods are needed to cope with its particular nature and to approach genomic problems in a sound statistical manner (Simon et al., 2003). This work focuses on the identification of differentially expressed genes in multiple slide microarray experiments.

As microarrays contain measurements of thousands of gene expression levels across multiple biological conditions, statistical analysis of a microarray experiment necessarily involve data mining or large scale multiple testing procedures. To the date, advances in this regard have either been multivariate but descriptive, or inferential but univariate.

Multivariate approaches developed until now include cluster analysis (Alizadeh et al., 2000; Ross et al., 2000), condition-related directions in the data's principal component space (Ospina & López-Kleine, 2013), permutation validated principal components analysis (Landgrebe et al., 2002) and discriminant analysis on principal components (Jombart et al., 2010). However, the statistical significance of the clusters or sets of genes identified as differentially expressed remains an open question –partly because the nature of the data does not easily allow distributional assumptions, and partly because unsupervised classification methods are not guaranteed to identify clusters that actually correspond to differentially expressed genes.

On the other hand, inferential approaches developed so far are based either on parametric models as ANOVA (Kerr et al., 2000) and Hidden Markov Models (Yuan & Kendziorski, 2006), or in non–parametric multiple testing procedures controlling the *family wise error rate* (Shaffer, 1995; Benjamini & Hochberg, 1995; Benjamini & Yekutieli, 2001; Dudoit et al., 2002) or the *false discovery rate* (Efron et al., 2001; Tusher et al., 2001; Storey & Tibshirani, 2001; Storey, 2002; Storey, 2003; Taylor et al., 2005). These methods, however, rely on a gene–by–gene approach in which the multivariate structure of the data is not taken into account.

Thus, multivariate–descriptive and univariate–inferential methods are the two pieces still to be assembled into an integral strategy for the identification of differentially expressed genes in microarray experiments.

In this work, we present a new strategy that combines a gene–by–gene multiple testing procedure and a multivariate descriptive approach into a multivariate inferential method suitable for microarray data. It is based mainly on the work of Storey & Tibshirani (2001) for the estimation of the $pFDR$ and on the construction of two artificial components –close to the data's principal components, but with an exact interpretation in terms of overall and differential gene expression.

Our method works best under some very common biological and technical assumptions and gives way to a new understanding of gene differential expression. We also provide a methodology to analyse time course microarray experiments and some guidelines for assessing whether the biological and technical assumptions required are likely to hold in a given data set.

We applied our method in two real microarray data sets previously analysed by Cai et al. (2013) and Ditt et al. (2006), respectively. In the first data set, appropriate biological and technical conditions were met and our method proved to be more useful than traditional approaches in that it identified new differentially expressed genes, it offered valuable insights regarding the time course behaviour of the differential expression process, and it avoided wrongly classifying non–expressed genes as differentially expressed. In the second data set, based on the results of our method, we were able to determine that the required biological and technical conditions were not met and thus to conclude that, in such cases, more traditional methods should be preferred.

As a rule, univariate oriented methods identified much more genes as being differentially expressed. These discrepancies arise from differences in the biological assumptions that underlie each method and the corresponding implied definitions of differential expression, and, thus, are not indicative of any method's greater power as a multiple testing procedure. Moreover, when the aim of the study is to perform an intervention upon differentially expressed genes, our method may prove very valuable as it prevents it from being done upon genes with no expression whatsoever.

As our method constitutes a first multivariate inferential approach to identifying differentially expressed genes in microarray data, many questions remain open for further investigation. These include statistical assessments of the extent to which our method's required probabilistic, biological and technical assumptions hold, extensions for when this is not the case, and further applications in biological studies.

This work is constructed as follows. In Chapter 2 we present the theoretical foundations that constitute the cornerstones of our methodology, including the basic aspects of microarray experiments, principal components analysis, bootstrap and permutation estimation methods, multiple testing procedures and Type I Error measures. In Chapter 3 we present our method for the identification of differentially expressed genes in microarray experiments along with some additional assessments, and introduce two real microarray data sets. In Chapter 4 we apply our method to those data sets and compare our results with previous studies. In Chapter 5 we present our conclusions and outline some open questions and future perspectives.

CHAPTER 2

Theoretical Framework

In this chapter, we present the theoretical foundations that underlie our methodology. We begin by introducing the basics of gene expression and microarray technology to better understand the nature of microarray data sets, and propose a general probability model for representing this type of data. Then, we introduce the basis of Principal Component Analysis (PCA) and show how to derive a gold–standard for assessing factors or components that capture desirable features when identifying differentially expressed genes in microarray experiments. Next, we give a brief overview of bootstrap estimation methods and permutation tests. Finally, we reformulate the problem of identifying differentially expressed genes as one of multiple hypothesis testing and present some measures to control Type I Error and false positives in the process. Estimation of those measures is also considered.

As it will be seen, the methods presented in this chapter (included those from Benjamini & Yekutieli, 2001; Tusher et al., 2001; Dudoit et al., 2002; Storey et al., 2004; Dudoit & Van Der Laan, 2008) begin by adopting a gene–by–gene single hypothesis testing approach by means of univariate test statistics and, afterwards, correct for multiple testing. Because the experimental units of a microarray experiment are the replicates (the genes being the variables measured over each replicate), this gene–by–gene approach implies a univariate point of view in which important features of the data remain unaccounted for. In Chapter 3, we propose a method that preserves the multivariate scale of the data by applying the multiple hypothesis testing procedure for control of the $pFDR$ presented towards the end of this chapter using a test statistic similar to the principal components in a PCA.

2.1 Background on DNA expression and microarrays

[1] Gene expression is the process by which different proteins are synthesized within a cell. It consists of the *transcription* of a gene or a segment of DNA into a complementary segment (or transcript) of mRNA and its subsequent *translation* into a protein specific for that gene. Each DNA segment is a double-stranded polymer composed of four basic molecular units or nucleotides –adenine (A), cytosine (C), guanine (G) and thymine (T)– that follow a unique pairing pattern. When the transcription stage takes place, the two

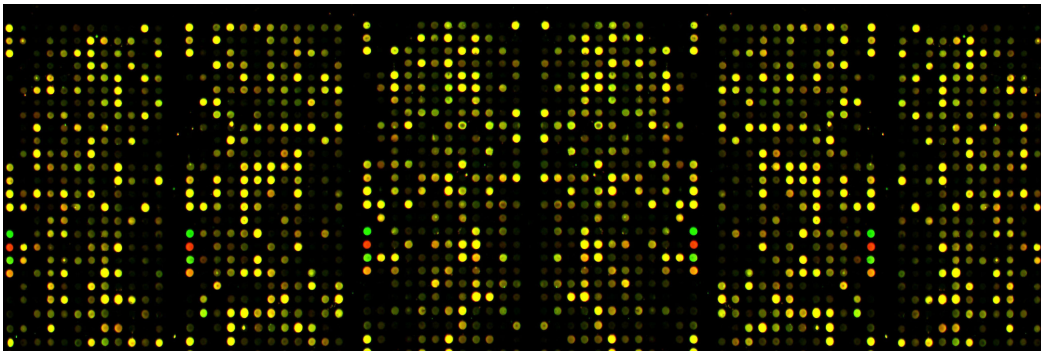
[1] This section is based on Dudoit et al. (2002) and Simon et al. (2003).

strands in the DNA segment split and a corresponding segment of mRNA is made by copying one of the strands and replacing the T nucleotide by a uracil (U) base. In the translation stage, this mRNA segment travels to a ribosome in the cytoplasm and directs the synthesis of a molecule of the corresponding protein.

Microarray technology aims at measuring the expression levels of each gene in the genome of a given organism by ways of quantifying the number of mRNA transcripts contained in the cytoplasm of a sample of cells. Because, in theory, there is a one to one correspondence between proteins, mRNA transcripts and its parent genes, and because a single transcript produces a single molecule of its respective protein, mRNA abundance in the transcription stage is assumed to constitute a good measure of gene expression in terms of protein production.

A microarray consists of thousands of genes' single strands printed on a microscope slide in a high density array (see Figure 2.1). Each spot on the microarray corresponds to a single gene or expressed sequence tag (EST) and contains many copies of the gene or *probes* printed within. In a microarray experiment, two mRNA samples are reverse transcribed into complementary single strands of DNA (or cDNA) and labeled using fluorescent dyes (usually red and green). The samples are mixed in equal proportions and placed on the array. Then, a competitive hybridization takes place in which transcripts attach themselves to matching probes printed on the microarray. Finally, the microarray is scanned and the red and green intensities are stored in a high resolution image file (see Figure 2.1) from which each gene's relative amount of mRNA is obtained.

FIGURE 2.1. Scanned image of a microarray after hybridization.



Taken from `http://www.bionivid.com`.

Several designs for microarray experiments –in which the number of microarrays and the relations between the treatment/control conditions and the reference/target samples vary– are available (see Simon et al., 2003, Chapter 3). We focus on multiple–slide experiments. The basic output of these experiments are the target samples' intensities in the scanned images after hybridization. If a different microarray is used for each individual, a common reference mRNA sample (usually pooled from the control replicates) can be compared against a each individual's target sample in each microarray. In this case, another output of the experiment consists of the target vs reference samples' intensity ratios. If more than one microarray per individual is available, dye swaps and more elaborate settings are frequently used (Simon et al., 2003).

Because of the technical complexities of microarray experiments –that involve microarray printing, mRNA sampling and hybridization, image analysis, etc.–, there are many

potential sources of systematic variation that may have to be dealt with via quality control and normalization procedures before any further analysis. Rigorous treatment of such procedures, however, is beyond the scope of this work and we will assume here on that normalization procedures, if needed, have been applied as a previous step to our method for the identification of differentially expressed genes[2].

2.2 General probability model for microarray data

We will assume the following probabilistic model holds for microarray data. Denote n the number of genes and $p = p_1 + p_2$ the number of replicates, with p_1 replicates for treatment and p_2 for control, respectively. Let X_{ij} be the random variable in a general probability space $(\Omega, \mathcal{F}, P)$ that represents the expression level of gene i at replicate j as measured in microarray experiments, with realization x_{ij} and cumulative distribution function F_{ij}. Then, the column random vector $\mathbf{X}_{\cdot j}$ represents all the gene expression levels for replicate j whereas the row random vector $\mathbf{X}_{i\cdot}$ represents the expression levels for gene i across all replicates; as before, $\mathbf{x}_{\cdot j}$, $F_{\cdot j}$, $\mathbf{x}_{i\cdot}$ and $F_{i\cdot}$ denote the realizations and the cumulative distribution functions of the corresponding random vectors. Finally, define the random matrix $\mathbf{X}$ as $(\mathbf{X}_{\cdot 1}, \ldots, \mathbf{X}_{\cdot p})$ with realization $X = (\mathbf{x}_{\cdot 1}, \ldots, \mathbf{x}_{\cdot p})$, and joint cumulative distribution function F.

In order to provide a general framework for gene expression data, we make the following assumptions:

1. The random column vectors $\mathbf{X}_{\cdot 1}, \ldots, \mathbf{X}_{\cdot p}$ are mutually independent.
2. Let $F_{\cdot j}$ denote the joint cumulative distribution function of $\mathbf{X}_{\cdot j}$ for $j = 1, \ldots, p$. Then, $F_{\cdot 1} = \cdots = F_{\cdot p_1} = F_{Tr}$ and $F_{\cdot p_1+1} = \cdots = F_{\cdot p_1+p_2} = F_C$, where treatment and control distributions, F_{Tr} and F_C, may be different. This implies $F_{i1} = \cdots = F_{ip_1} = F_{iTr}$ and $F_{i(p_1+1)} = \cdots = F_{ip} = F_{iC}$, for $i = 1, \ldots, n$.
3. The random row vectors $\mathbf{X}_{1\cdot}, \ldots, \mathbf{X}_{n\cdot}$ are not mutually independent, nor are they identically distributed.

The first assumption simply states that the replicates are mutually independent, which arises naturally due to experimental conditions. The second imposes identical joint and marginal distributions for treatment replicates and for control replicates separately, but not necessarily equal for all replicates. For this assumption to be plausible, we will assume that the data has been standardized column–wise so that each column has, at least, zero mean and unit variance. The third assumption copes with the fact that different genes have different metabolic functions and therefore express themselves in a different way at different moments. Also, given that genes usually work in groups (Dudoit & Van Der Laan, 2008), dependence between some of the genes' expression levels is expected and, so, is included in the model.

Note that this model is very general and that, given the nature of microarray data, further assumptions concerning the dependence structure of the genes or the funcional form of the distributions would be difficult to sustain. In this setup, it will be convenient to think of the genes as the *individuals* and the replicates as the *variables* of the analysis (measured over each individual).

[2]See Section 2.6 for more on this subject.

2.3 Principal Components Analysis

One of the main objectives of our methodology is to capture and take advantage of the multivariate nature of microarray data throughout the entire analysis. This is not only limited to the correlation structures between genes, already captured by some univariate–approach methods (Benjamini & Hochberg, 1995; Benjamini & Yekutieli, 2001; Tusher et al., 2001; Storey, 2002; Storey & Tibshirani, 2003; Taylor et al., 2005), but also to preserve the relative scale of expression levels among all genes and, so, to be able to answer questions like which genes have higher (lower) expression levels, what does 'higher' means in terms of the expression levels in a given microarray data set, which differences in expression levels are sufficiently large in this scale, etc. For this, we find that Principal Components Analysis (PCA) is extremely useful. In order to maintain the general probability model for gene expression data of section 2.2 without adding further probabilistic assumptions, we focus on the descriptive stream of standardized PCA and follow Lebart et al. (1995) in its presentation.

2.3.1 PCA mechanics

Let $\mathbf{W}$ be a $n \times p$ data matrix with elements w_{ij} representing p measurements or variables for n individuals. One of the aims of PCA is to find the axes or directions in $\mathbb{R}^p$ that capture most of the variability in $\mathbf{W}$. For these directions to capture actual variability and not variability due to scale and location, the first step is to standardize $\mathbf{W}$ column–wise. Let $\mathbf{X}$ be $n \times p$ matrix with elements

$$x_{ij} = \frac{w_{ij} - \bar{w}_{\cdot j}}{se(\mathbf{w}_{\cdot j})}, \qquad i = 1, \ldots, n, \quad j = 1, \ldots, p, \tag{2.1}$$

where $\bar{w}_{\cdot j} = n^{-1} \sum_{i=1}^{n} w_{ij}$ and $se(\mathbf{w}_{\cdot j}) = n^{-1} \sum_{i=1}^{n} (w_{ij} - \bar{w}_{\cdot j})^2$. Now, every column of $\mathbf{X}$ has zero mean and unit variance. Note that this makes Assumption 2 of the General Probability Model from Section 2.2 plausible.

Following Lebart et al. (1995), we will capture the variability in $\mathbf{X}$ in terms of the inertia of the rows of $\mathbf{X}$ upon each column[3]. For this, let d be a convenient metric on $\mathbb{R}^p$ represented by the $p \times p$ symmetric matrix $\mathbf{D}$ with elements d_j in the diagonal and zero otherwise, such that the distance between two vectors $\mathbf{x}$ and $\mathbf{y}$ in $\mathbb{R}^p$ is $d(\mathbf{x}, \mathbf{y}) = [\sum_{j=1}^{p} d_j (x_j - y_j)^2]^{1/2}$. Also, let the $n \times n$ symmetric matrix $\mathbf{M}$ be the weights' matrix, with elements m_i in the diagonal and zero otherwise, m_i representing the weight of the i-th row of $\mathbf{X}$.

Because $\mathbf{X}$ is standardized column–wise, its center of gravity is the origin and the total inertia of $\mathbf{X}$ with respect to the origin may be calculated as

$$In = \sum_{i=1}^{n} m_i \mathrm{d}^2 (\mathbf{x}_{i\cdot}, \mathbf{0}) = \sum_{i=1}^{n} m_i \sum_{j=1}^{p} d_j x_{ij}^2 = tr \left(\mathbf{X}' \mathbf{M} \mathbf{X} \mathbf{D} \right),$$

where $\mathbf{x}_{i\cdot}$ is the i-th row of $\mathbf{X}$, $\mathbf{X}'$ is $\mathbf{X}$ transposed and $\mathrm{d}^2 (\mathbf{x}_{i\cdot}, \mathbf{0})$ represents the squared distance between $\mathbf{x}_{i\cdot}$ and the origin under the metric d.

[3]This constitutes the "individuals' cloud analysis" part of the PCA as explained by Lebart et al. (1995). The dual part of the PCA, the "variables' cloud analysis", will not be needed farther on and, thus, will be omitted from our presentation. We refer the interested reader to Section 1.2 of Lebart et al. (1995).

Now, let $\mathbf{u} \in \mathbb{R}^p$ be a unitary vector representing any direction in the metric space $(\mathbb{R}^p, d)$. The coordinates of the orthogonal projections of the n individuals upon the direction $\mathbf{u}$ are

$$\boldsymbol{\varphi} = \mathbf{XDu}, \qquad \text{where} \qquad \mathbf{u'Du} = 1, \tag{2.2}$$

and the inertia projected on $\mathbf{u}$ is

$$In(\mathbf{u}) = \boldsymbol{\varphi}'\mathbf{M}\boldsymbol{\varphi} = \mathbf{u'DX'MXDu}. \tag{2.3}$$

Note that, $\boldsymbol{\varphi}$ is centered, but does not necessarily have unit variance.

PCA consists of finding the set of orthogonal unitary vectors in $(\mathbb{R}^p, d)$, $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, $k \leq p$, that maximizes the inertia projected upon them. Because $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$ is an orthogonal set, the inertia projected on the space generated by them is just $In(\mathbf{u}_1, \ldots, \mathbf{u}_k) = In(\mathbf{u}_1) + \cdots + In(\mathbf{u}_k)$. Now, define the first k Principal Components of $\mathbf{X}$, $\{\mathbf{u}_1, \ldots, \mathbf{u}_k\}$, as the solution to the following succesive maximization problems:

$$\begin{aligned} &\text{Max}_{\mathbf{u}_1} : \ \mathbf{u_1'DX'MXDu_1} && \text{s.t.} \quad \mathbf{u_1'Du_1} = 1, \\ &\text{Max}_{\mathbf{u}_s : s=2,\ldots,k} : \ \mathbf{u_s'DX'MXDu_s} && \text{s.t.} \quad \mathbf{u_s'Du_r} = \begin{cases} 1, & s = r \\ 0, & s \neq r \end{cases}, \quad r = 1, \ldots, s. \end{aligned} \tag{2.4}$$

Maximization via Lagrange Multipliers yields the solution

$$\mathbf{X'MXDu_s} = \lambda_s \mathbf{u}_s, \qquad s = 1, \ldots, k. \tag{2.5}$$

Then, it is easy to see that $\lambda_1, \ldots, \lambda_k$ and $\mathbf{u}_1, \ldots, \mathbf{u}_k$ are the first k eigenvalues and corresponding k eigenvectors of $\mathbf{X'MXD}$. Also, premultiplying (2.5) by $\mathbf{u}_s'D$ and replacing into (2.3) and (2.4), we get $In(\mathbf{u}_s) = \lambda_s$, $s = 1, \ldots, k$.

Should we have supplementary rows of data in a $(n^+ \times p)$ matrix $\mathbf{W}^+$ (like individuals that where not included in the first analysis or the centers of gravity of groups of individuals), the way to standardize them and project them onto the principal components of the previous PCA is:

$$x_{ij}^+ = \frac{w_{ij}^+ - \bar{w}_{\cdot j}}{se(\mathbf{w}_{\cdot j})}, \qquad \boldsymbol{\varphi}_s^+ = \mathbf{X}^+\mathbf{Du_s}, \tag{2.6}$$

for $i = 1, \ldots, n^+$, $j = 1, \ldots, p$ and $s = 1, \ldots, k$.

Finally, to give a familiar statistical meaning to the previous results, we set $\mathbf{D} = \mathbf{I}_p$, the identity $(p \times p)$ matrix, and $m_i = 1/n$ so that $\mathbf{M} = n^{-1}\mathbf{I}_n$. Then, as the columns in $\mathbf{X}$ have zero mean and unit variance, $\mathbf{X'MXD} = n^{-1}\mathbf{X'X}$ is the correlation matrix of $\mathbf{X}$ and the total inertia $In = tr(n^{-1}\mathbf{X'X}) = p$. Moreover, λ_s and $\mathbf{u}_s$, $s = 1, \ldots, k$, are the eigenvalues and eigenvectors of the correlation matrix of $\mathbf{X}$ and the iniertia projected on any $\mathbf{u}_s$, $s = 1, \ldots, k$, takes the form of the variance of $\boldsymbol{\varphi}_s$ from (2.2), that is:

$$In(\mathbf{u}_s) = \lambda_s = \boldsymbol{\varphi}_s' \left(\frac{1}{n} I_n\right) \boldsymbol{\varphi}_s = \frac{1}{n}\sum_{i=1}^{n} \varphi_{is}^2 = \text{Var}(\boldsymbol{\varphi}_s), \qquad s = 1, \ldots, k.$$

2.3.2 A word on interpretation

Because the principal components are the eigenvectors of the standardized data's correlation matrix, they are entirely determined by $\mathbf{X}$. Moreover, $-\mathbf{u}_s$, $s = 1, \ldots, k$, are also a solution of (2.4), so the principal components will vary between data sets and their interpretation will be difficult more often than not.

As will be the case in Chapter 3, for detecting differentially expressed genes in a microarray experiment, we need factors or directions that capture two very specific characteristics of the data: a gene's overall expression level and the extent to which it is differentially expressed, both measured with respect to all of the genes' expression levels. By the nature of PCA, there is no guarantee for a set of microarray data that any of its principal components will capture any of those two features of gene expression.

However, the principal components of the data do provide with a practical gold–standard in terms of captured variability for comparing directions that do refer to a specific set of characteristics (as overall expression levels and differential expression) in a given microarray data set. For example, let $\mathbf{v} \in \mathbb{R}^p$ be an unitary vector in $(\mathbb{R}^p, d)$ different from the principal components of $\mathbf{X}$. The inertia projected onto $\mathbf{v}$, $In(\mathbf{v})$, can be computed from (2.3). Define the inertia ratio of $\mathbf{v}$ as

$$R(\mathbf{v}) = \frac{In(\mathbf{v})}{In(\mathbf{u}_1)} = \frac{In(\mathbf{v})}{\lambda_1}. \tag{2.7}$$

If a large part of the variability in $\mathbf{X}$ relates to direction $\mathbf{v}$, $R(\mathbf{v})$ will be close to one, for λ_1 is the maximum inertia that can be projected onto any single direction.

Also, if we have access to the same variables measured at different time points, say, $\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(L)}$, a practical way to compare the amount of information or variability captured by a set of orthogonal directions $\{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$ between time points (without additional probabilistic assumptions) is to compare the ratios

$$R^{(l)}(\mathbf{v}_1, \ldots, \mathbf{v}_k) = \frac{In^{(l)}(\mathbf{v}_1, \ldots, \mathbf{v}_k)}{In^{(l)}(\mathbf{u}_1, \ldots, \mathbf{u}_k)} = \frac{In^{(l)}(\mathbf{v}_1, \ldots, \mathbf{v}_k)}{\lambda_1^{(l)} + \cdots + \lambda_k^{(l)}}, \quad \text{for } l = 1, \ldots, L. \tag{2.8}$$

2.4 Bootstrap

In this section we present the basics of the bootstrap estimation method introduced by Efron (1979), following the presentation made by Efron & Tibshirani (1993). The usefulness of this method in identifying differentially expressed genes will become clear when simulating null distributions and dealing with multiple hypothesis testing farther on. We address bootstrap methods regarding point estimation, confidence intervals and hypothesis testing. For simplicity, throughout this section, we use a different notation from the one in the General Probability Model of Section 2.2.

2.4.1 Bootstrap estimates: One sample case

Suppose we have a data set $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ that has been generated by random sampling from a probability distribution F; that is, $\mathbf{x}$ is a realization of the random vector $\mathbf{X} = (X_1, \ldots, X_n)$, where $X_i \sim F$, $i = 1, \ldots, n$, are iid random variables. As in Efron &

Tibshirani (1993)'s notation, we denote this by $F \to \mathbf{x}$. Now suppose there is a parameter of interest $\theta = t(F)$ and an estimator of θ, say $\hat{\theta}(\mathbf{X}) = s(\mathbf{X})$ for some measurable function s, with realization or estimate $\hat{\theta}(\mathbf{x}) = s(\mathbf{x})$ and distribution $G = F \circ s^{-1}$.

Let $\phi = t'(G)$ be a parameter of interest from G (usual examples are the variance, the bias or the mean squared error of $\hat{\theta}(\mathbf{X})$) to which we may refer to as a "level 2" parameter of interest (Athreya & Lahiri, 2006). Also, let $\hat{\phi} = s'(\mathbf{y})$ be a good estimation of ϕ for any distribution G, should we have access to observations $\mathbf{y} = (y_1, \ldots, y_n) \leftarrow G$. In theory, we could calculate ϕ from F or estimate it from $\mathbf{x}$, but, unless F is known and s is very simple, there usually is no mathematical formula for this. The bootstrap offers an alternative way of estimating ϕ for which F and G do not need to be known.

Let $\tilde{F}$ be the empirical distribution of $\mathbf{x}$, where $\tilde{F}$ is the probability measure that assigns probability $1/n$ to each x_i in $\mathbf{x}$. In other words, if a random variable $X \sim \tilde{F}$, then $P_{\tilde{F}}(X = x) = \#\{x_i = x\}/n$. The plug-in estimates of θ and ϕ are $\tilde{\theta}(\mathbf{x}) = t(\tilde{F})$ and $\tilde{\phi} = t'(\tilde{G}) = t'(\tilde{F} \circ s^{-1})$, respectively[4]. Note that, although computation of $\tilde{\theta} = t(\tilde{F})$ is usually straightforward, in general there is no explicit way of calculating $\tilde{\phi}$ [5].

Given $\mathbf{x}$, define a bootstrap sample of $\mathbf{x}$ or, equivalently, a bootstrap sample from $\tilde{F}$, as $\mathbf{x}^* = (x_{i_1}, \ldots, x_{i_n})$, where $(i_1, i_2, \ldots, i_n)$ is a random sample of size n drawn with replacement from $\{1, 2, \ldots, n\}$. Note that, by construction, $\tilde{F} \to \mathbf{x}^*$. Define a bootstrap replication of $\hat{\theta}(\mathbf{x})$ as $\hat{\theta}^* = s(\mathbf{x}^*)$. The calculation of the bootstrap estimate of ϕ given $\mathbf{x}$, $\hat{\phi}_B$, is presented in Algorithm 2.4.1.

Algorithm 2.4.1: Computation of bootstrap estimates in the one sample case.

1. Draw a large number B of independent bootstrap samples from $\mathbf{x}$, $\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*$.
2. Calculate B bootstrap replications of $\hat{\theta}(\mathbf{x})$, $\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*$, from the bootstrap samples in the previous step.
3. Estimate ϕ as $\hat{\phi}_B = s'(\hat{\theta}_1^*, \ldots, \hat{\theta}_B^*)$.

When ϕ can be expressed as the expected value of a function f of $\hat{\theta}(\mathbf{X})$ and F (like the bias $E_F(\hat{\theta} - \theta)$, or the distribution function $E_F(I\{\hat{\theta} \leq x\})$ where $I\{\cdot\}$ is the indicator function), step 3 in Algorithm 2.4.1 can be reformulated as

3′. Estimate ϕ as $\hat{\phi}_B = B^{-1} \sum_{b=1}^{B} f(\hat{\theta}_b^*)$.

In such case (Hall, 1992, p. 288), $\hat{\phi}_B$ is an unbiased consistent estimator of $\tilde{\phi}$, given $\mathbf{x}$ [6]. That is:

$$E_{\tilde{F}}(\hat{\phi}_B) = \tilde{\phi} \quad \text{and} \quad \lim_{B \to \infty} \hat{\phi}_B = \tilde{\phi},$$

where the randomness in $\hat{\phi}_B$ given $\mathbf{x}$ comes from the random sampling from $\mathbf{x}$. In other words, the bootstrap estimate $\hat{\phi}_B$ is itself an estimate of $\tilde{\phi}$, whose accuracy increases with B. Also, note that in the special case $\phi = t'(G) = G$, the bootstrap estimate of

[4] It can be proven that $\tilde{F}$ is a sufficient statistic for F (Efron & Tibshirani, 1993, p.32) and that $\tilde{\theta}$ and $\tilde{\phi}$ are consistent estimators for θ and ϕ (Bickel & Doksum, 2001, p. 302). In this sense, the plug-in estimate $t(\tilde{F})$ is a very good choice for estimating $t(F)$ if n is large and there is no parametrical assumptions or additional information about F other than the data $\mathbf{x}$.

[5] For example, for $X \sim F$, estimating $E(X)$ with the sample mean, if $\phi = Var(\bar{X})$, we know that $\phi = Var(X)/n$, and $\tilde{\phi} = \frac{1}{n} \sum_{i=1}^{n} \frac{(x_i - \bar{x})^2}{n}$. This may be very difficult for s other than the mean.

[6] For more asymptotic properties of bootstrap estimators, see Hall (1992) and Athreya & Lahiri (2006).

the distribution of $\hat{\theta}$ given $\mathbf{x}$ becomes the empirical distribution of the bootstrap sample $\hat{\theta}^*_1, \ldots, \hat{\theta}^*_B, \tilde{G}_B$.

Summarizing, for large B, the bootstrap estimate $\hat{\phi}_B$ is a good approximation of the plug-in estimate $\tilde{\phi}$, which, in turn, is a very good estimate of ϕ, when no information is available about F other than the sample $\mathbf{x}$. Note, however, that, even as $B \to \infty$, there remains uncertainty in $\hat{\phi}_B$ as we are still estimating F by $\tilde{F}$ and $\lim_{B\to\infty} \hat{\phi}_B$ remains the plug-in estimate of ϕ.

2.4.2 Bootstrap confidence intervals

There are multiple ways of computing approximate confidence intervals for a parameter of interest θ using bootstrap methods[7]. Here, we briefly present two such methods and some of their properties.

2.4.2.1 Percentile intervals

In normal theory, the bounds of the usual $(1-\alpha)$ confidence interval for the mean, $\bar{x} \pm z_{1-\alpha/2}\frac{\sigma}{\sqrt{n}}$, are also estimates of the $\alpha/2$ and $1-\alpha/2$ percentiles of $\bar{X}$'s distribution. This property arises from the facts that $\bar{X}$ is an unbiased estimator of μ, that $\mathrm{Var}(\bar{X})$ is known and constant for all values of μ, and that $\sqrt{n}(\bar{X}-\mu)/\sigma$ is a pivotal statistic (i.e. its distribution does not depend on unknown parameters). The same principle can be applied to bootstrap estimates as follows.

Under the previous assumptions of unbiasedness and constant variance for $\hat{\theta}$ with respect to θ, a $(1-\alpha)$ confidence interval for θ would be $[G^{-1}(\alpha/2), G^{-1}(1-\alpha/2)]$, where $G^{-1}(\alpha)$ is the α percentile of $\hat{\theta}$'s distribution. G being unknown, one can approximate such interval by ways of the bootstrap estimate of G. Then, for B large enough, an approximate $(1-\alpha)$ confidence interval for θ can be computed as

$$[\theta_{lo}, \theta_{up}] = \left[\hat{\theta}^{*(B\times\alpha/2)}, \hat{\theta}^{*(B\times(1-\alpha/2))}\right] \tag{2.9}$$

where $\hat{\theta}^{*(B\times\alpha)}$ denotes the α percentile of the bootstrap replications $\hat{\theta}^{*(1)}, \ldots, \hat{\theta}^{*(B)}$. Note that $\hat{\theta}^{*(B\times\alpha)}$ is the actual α percentile of $\tilde{G}_B$ and, by means of the plugin principle, it is itself an estimate of the α percentile of $\hat{\theta}$'s distribution G.

A practical advantage of the percentile interval in (2.9) is that it is *transformation-respecting*: if m is a monotone function defined in the parameter space (Efron & Tibshirani, 1993, p. 175-177), the percentile interval for $m(\theta)$ is just $[m(\theta_{lo}), m(\theta_{up})]$.

Despite the previous properties, use of the percentile intervals is not justified when $\hat{\theta}$ is biased for θ or when $\mathrm{Var}(\hat{\theta})$ depends on the true value of θ. Moreover, the percentile interval (2.9) is only first order accurate, that is

$$P\left(\hat{\theta}^{*(B\times\alpha/2)} \leq \theta \leq \hat{\theta}^{*(B\times(1-\alpha/2))}\right) = 1 - \alpha + O(n^{-1/2}).$$

These limitations are what motivates the construction of BCa intervals presented in the next section.

[7] For a more theoretical treatment of such intervals and their asymptotic properties, the reader is referred to Hall (1992) and DiCiccio & Efron (1996).

2.4.2.2 BCa intervals

In these section, we outline the construction of BCa intervals for upper confidence bounds. The extension for lower confidence bounds and two-sided confidence intervals is straightforward and is therefore omitted.

Construction of BCa intervals for θ (BCa standing for "bias corrected and accelerated") is based on the following model (Efron & Tibshirani, 1993, p. 326). Suppose there exists an increasing function m such that, for $\phi = m(\theta)$ and $\hat{\phi} = m(\hat{\theta})$,

$$\frac{\hat{\phi} - \phi}{\sigma_\phi} \sim N(-z_0, 1), \qquad \sigma_\phi = \sigma_{\phi_0} \left[1 + a\left(\phi - \phi_0\right)\right] \tag{2.10}$$

where σ_{ϕ_0} denotes the standard error of $\hat{\phi}$ when $\phi = \phi_0$, for some ϕ_0. Here, z_0 is the bias of $\hat{\phi}$ with respect to ϕ and a represents the rate of change of $\hat{\phi}$'s standard error with respect to changes in ϕ.

If (2.10) holds exactly, an exact upper $1 - \alpha$ confidence bound for ϕ is

$$\phi[1-\alpha] = \hat{\phi} + \sigma_{\hat{\phi}} \frac{z_0 + z_{1-\alpha}}{1 - a(z_0 + z_{1-\alpha})},$$

where z_α is the α percentile of a standard normal distribution. Let G denote the cumulative distribution function of $\hat{\theta}$ and recall that m is increasing; the exact upper $1-\alpha$ confidence bound for θ is then

$$\theta[1-\alpha] = G^{-1}\left(\Phi\left[z_0 + \frac{z_0 + z_{1-\alpha}}{1 - a\left(z_0 + z_{1-\alpha}\right)}\right]\right), \tag{2.11}$$

where Φ is the standard normal cumulative distribution function[8]. Note that the function m need not be known and that BCa intervals are also transformation-respecting.

When (2.10) does not hold exactly, the error in the approximation will typically be of order $O(n^{-1})$, which implies that BCa intervals constructed as in (2.11) are second order accurate (Efron & Tibshirani, 1993, p. 325-326), that is, $P(\theta \leq \theta[1-\alpha]) = 1-\alpha+O(n^{-1})$. Moreover, BCa intervals are also second order correct (DiCiccio & Efron, 1996).

In practice, we obtain approximate upper confidence bounds for θ replacing G, z_0 and a by their estimates $\tilde{G}$, $\hat{z}_0$ and $\hat{a}$, according to Efron (1987) and Efron & Tibshirani (1993, Chapters 14 and 22). First, we estimate G with $\tilde{G} \approx \tilde{G}_B$, for large B. For z_0, note that if (2.10) holds, then $\hat{\phi}$ has the same distribution as $\phi + \sigma_\phi(Z - z_0)$, where $Z \sim N(0,1)$. Then,

$$P(\hat{\theta} \leq \theta) = P(\hat{\phi} \leq \phi) = P(Z \leq z_0) = \Phi(z_0).$$

Then, $z_0 = \Phi^{-1}[P(\hat{\theta} \leq \theta)]$ and a bootstrap estimate for z_0 is

$$\hat{z}_0 = \Phi^{-1}\left(\frac{\#\{\hat{\theta}_b^* < \hat{\theta}\}}{B}\right). \tag{2.12}$$

[8] For details, see Efron (1987).

For the constant a, Efron (1987) shows that a good estimate is

$$\hat{a} = \frac{\sum_{i=1}^{n}(\hat{\theta}_{jack} - \hat{\theta}_{(i)})^3}{6\left[\sum_{i=1}^{n}(\hat{\theta}_{jack} - \hat{\theta}_{(i)})^2\right]^{3/2}}, \tag{2.13}$$

where $\hat{\theta}_{(i)} = s(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ is the i-th *jackknife value* of $\hat{\theta}$ and $\hat{\theta}_{jack} = n^{-1}\sum_{i=1}^{n}\hat{\theta}_{(i)}$.

2.4.3 Permutation and bootstrap hypothesis tests: Two sample case

In this section we describe two nonparametric procedures for hypothesis testing in the two sample case. Although here we deal mainly with a single hypothesis being tested, the value of these two methods for detecting differentially expressed genes lies on the fact that they provide a useful way of estimating a test statistic's null distribution under the general probability model of section 2.2. The general framework is as follows[9].

Let $\mathbf{X} = (X_1, \ldots, X_p)$ a random vector with realization $\mathbf{x} = (x_1, \ldots, x_p)$, whose first p_1 components are a random sample from a probability distribution F_1, say the treatment group, and whose last p_2 components are a random sample from a (possibly different) probability distribution F_2, say the control group, where $p = p_1 + p_2$. Note that $\mathbf{X}$ may represent the behaviour of a single gene (a random row vector) in the general probability model in Section 2.2. Now, our interest may lie in determining whether $F_1 = F_2$ or whether both F_1 and F_2 share a common feature, i.e., $t(F_1) = t(F_2)$, for some function t. We concentrate on the former.

Define a test statistic $s(\mathbf{X}) \in \mathbb{R}$, where s is a measurable function, such that large values of $s(\mathbf{x})$ are evidence against the null hypothesis $H : F_1 = F_2$, and in favour of some alternative hypothesis K. Now, for a fixed rejection region $[t, \infty)$, define the decision rule δ_t such that

$$\delta_t(\mathbf{x}) = \begin{cases} 1, & s(\mathbf{x}) \in [t,\infty) \quad \longrightarrow \quad \text{We reject H,} \\ 0, & s(\mathbf{x}) \notin [t,\infty) \quad \longrightarrow \quad \text{We don't reject H.} \end{cases} \tag{2.14}$$

The significance level of the test δ_t is defined as $\alpha_t = P_H(\delta_t(\mathbf{X}) = 1)$ and depends on the rejection region we choose. Here, P_H is the probability measure on $(\Omega, \mathcal{F})$ when H is true. For fixed $\mathbf{x}$, the achieved significance level (ASL) or p–value of the test is $ASL = \inf_t \{\alpha_t : \delta_t(\mathbf{x}) = 1\}$. As α_t is decreasing in t, when α_t is also continuous, the ASL may be computed as

$$ASL = \alpha_{s(\mathbf{x})} = P_H\left(\delta_{s(\mathbf{x})}(\mathbf{X}) = 1\right) = P_H\left(s(\mathbf{X}) \geq s(\mathbf{x})\right) = 1 - G_H(s(\mathbf{x})), \tag{2.15}$$

where G_H denotes $s(\mathbf{X})$'s cumulative distribution function under H. Note that $ASL \leq \alpha_t$ iif $\delta_t(\mathbf{x}) = 1$ iif $s(\mathbf{x}) \geq t$. So $ASL(\mathbf{X}) = \alpha_{s(\mathbf{X})}$ is, itself, a test statistic. Now the question arises of how to estimate a test statistic's null distribution without further parametric assumptions.

[9] For a thorough presentation of hypothesis tests, see Chapter 4 of Bickel & Doksum (2001).

2.4.3.1 Permutation tests

[10]In permutation tests, we need the order representation of the data. Let $\mathbf{g} = (g_1, \ldots, g_p)$ be a vector consisting of p_1 ones and p_2 twos, such that $g_i = 1$ if x_i belongs to the treatment group, and $g_i = 2$ if x_i belongs to the control group. If H is true, then $X_1, \ldots, X_p$ are exchangeable and the vector $\mathbf{g}$ has probability $1/\binom{p}{p_1}$ of taking any of its possible $\binom{p}{p_1}$ values[11].

Now, write $s(\mathbf{x}) = s(\mathbf{x}, \mathbf{g})$ as a function of both $\mathbf{x}$ and $\mathbf{g}$ and define a permutation replication of $s(\mathbf{x}, \mathbf{g})$ as $s^* = s(\mathbf{x}, \mathbf{g}^*)$, where $\mathbf{g}^*$ is any one of the $\binom{p}{p_1}$ possible vectors consisting of p_1 ones and p_2 twos, keeping $\mathbf{x}$ fixed. In other words, $\mathbf{g}^*$ is a random sample of size p taken *without replacement* from $\mathbf{g}$.

Under H, $s(\mathbf{x}, \mathbf{g}^*)$ is a random variable with $\binom{p}{p_1}$ possible values, each occurring with probability $1/\binom{p}{p_1}$, and with realization $s(\mathbf{x}, \mathbf{g})$. We define the permutation distribution of s^*, G_{perm} , as the distribution that assigns probability $1/\binom{p}{p_1}$ to each possible permutation replication of $s(\mathbf{x}, \mathbf{g})$. Then, the permutation ASL is defined as

$$ASL_{perm} = P\left(s\left(\mathbf{x}, \mathbf{g}^*\right) \geq s\left(\mathbf{x}, \mathbf{g}\right)\right) = \frac{\#\left\{s\left(\mathbf{x}, \mathbf{g}^*\right) \geq s\left(\mathbf{x}, \mathbf{g}\right)\right\}}{\binom{p}{p_1}}, \tag{2.16}$$

where $\mathbf{x}$ remains fixed and the random element is $\mathbf{g}^*$.

To relate the ASL_{perm} to the test's ASL in (2.15), it can be shown (Efron & Tibshirani, 1993, p. 210) that for any $0 < \alpha < 1$,

$$P_H(ASL_{perm} < \alpha) \approx \alpha,$$

where the approximation is due only to the discreteness of G_{perm}. It follows that for small values of α (as is usually the case) and relatively large values of p (≥ 10), by rejecting H when $ASL_{perm} \leq \alpha$, one (approximately) achieves the desired test's significance level.

In practice, for large p, ASL_{perm} can be approximated by Monte Carlo Methods as in Algorithm 2.4.2. Note that this is equivalent to computing the bootstrap estimate in Algorithm 2.4.1, with $f(s(\mathbf{x}, \mathbf{g})) = I\{s(\mathbf{x}, \mathbf{g}^*) \geq s(\mathbf{x}, \mathbf{g})\}$ in step $3'$, and taking samples without replacement (instead of bootstrap samples) in step 1. As with bootstrap estimates, $\widehat{ASL}_{perm}$ is unbiased and consistent for ASL_{perm} as $B \to \infty$.

Algorithm 2.4.2: Estimation of the ASL_{perm} in permutation tests.

1. Draw a large number B of independent samples $\mathbf{g}_1^*, \ldots, \mathbf{g}_B^*$ from G_{perm}.
2. Compute the corresponding permutation replications of $s(\mathbf{x}, \mathbf{g})$, $s_1^*, \ldots, s_B^*$, where $s_b^* = s(\mathbf{x}, \mathbf{g}_b^*)$, $b = 1, \ldots, B$.
3. Estimate ASL_{perm} as $\widehat{ASL}_{perm} = \#\left\{s_b^* \geq s\left(\mathbf{x}, \mathbf{g}\right)\right\}/B$.

[10]We follow closely the presentation of the subject in Chapter 15, Efron & Tibshirani (1993).
[11]This is the "Permutation Lemma" in (Efron & Tibshirani, 1993, p. 207).

2.4.3.2 Bootstrap hypothesis tests

[12]In permutation tests, we estimated G_H by G_{perm}, that is, maintaining $\mathbf{x}$ fixed and varying the group vector $\mathbf{g}$. The approach in bootstrap hypothesis tests is somewhat different in that it estimates G_H using the plug-in principle. If H is true, then $x_1, \ldots, x_p$ where generated from a common distribution, say, F_H, whose plug-in estimate, $\tilde{F}_H$, assigns $1/(p_1+p_2)$ probability to each single element in $\mathbf{x}$. Then, under H and keeping $\mathbf{g}$ fixed, we can compute a bootstrap version of the test's ASL using the plug-in estimate of $s(\mathbf{X}, \mathbf{g})$'s cumulative distribution function $\tilde{G}_H$ as follows:

$$ASL_{boot} = P_H\left(s\left(\mathbf{x}^*, \mathbf{g}\right) \geq s\left(\mathbf{x}, \mathbf{g}\right)\right) = 1 - \tilde{G}_H\left(s\left(\mathbf{x}, \mathbf{g}\right)\right), \tag{2.17}$$

where $\mathbf{x}$ and $\mathbf{g}$ are fixed and $\mathbf{x}^* \leftarrow \tilde{F}_H$ is a bootstrap sample of $\mathbf{x}$. Here, ASL_{boot} is the plug-in estimate of the test's ASL in (2.15).

In practice, we estimate ASL_{boot} with Monte Carlo methods as shown in Algorithm 2.4.3. Once again, $\widehat{ASL}_{boot}$ is unbiased and consistent for ASL_{boot} as $B \to \infty$, and consistent for ASL as $p \to \infty$ and $B \to \infty$. It follows that for large values of p and B, by rejecting H when $ASL_{boot} \leq \alpha$ one (approximately) achieves the desired test's significance level.

Algorithm 2.4.3: Estimation of the ASL_{boot} in bootstrap tests.

1. Draw a large number B of independent bootstrap samples from $\mathbf{x}$, $\mathbf{x}_1^*, \ldots, \mathbf{x}_B^*$.
2. Compute the corresponding bootstrap replications of $s(\mathbf{x}, \mathbf{g})$, $s_1^*, \ldots, s_B^*$, where $s_b^* = s(\mathbf{x}_b^*, \mathbf{g})$, $b = 1, \ldots, B$.
3. Estimate ASL_{boot} as $\widehat{ASL}_{boot} = \#\left\{s_b^* \geq s\left(\mathbf{x}, \mathbf{g}\right)\right\}/B$.

2.4.3.3 Selection of the test statistic

The choice of the test statistic to be used in the previous hypothesis tests depends mainly on the test's alternative hypothesis K. If we keep the general hypothesis $K : F_1 \neq F_2$, the power of a given test will increase if the two distributions differ in some feature that the test statistic captures well. For instance, if the true distributions F_1 and F_2 differ only in their expected value and $s(\mathbf{x}) = \bar{x}_1 - \bar{x}_2$, the test will have good power of detecting the alternative. However, if F_1 and F_2 have equal means but unequal variances, the probability of detecting K using $s(\mathbf{x})$ would be very low (Efron & Tibshirani, 1993, Chapter 15).

For the detection of differentially expressed genes in microarray experiments, one is usually more interested in detecting differences in mean expression levels between treatment and control, than in detecting differences in variances or in other functionals of the distributions (Dudoit & Van Der Laan, 2008). We then want a test statistic that captures differences in the mean expression levels so that if H is rejected, we know (up to a certain significance level) that gene's expression levels following F_1 and F_2 differ at least in their expected values. Such considerations will be addressed in Chapter 3.

[12]We follow closely the presentation of the subject in Chapter 16, Efron & Tibshirani (1993).

2.5 Multiple hypothesis testing

When identifying differentially expressed genes in microarray data, one necessarily encounters simultaneous multiple hypothesis tests: the null hypotheses being those of no differential expression for each one of the thousands of genes under study. Generalizations of the single hypothesis test paradigm have become necessary and different approaches have been taken in this regard.

In this section, we present two such approaches[13]: one controlling the family–wise error rate (FWER), the other based on the false discovery rate (FDR). We discuss the advantages and limitations of each in the context of genetic differential expression. In what follows, we return to the notations and assumptions of the general probability model from Section 2.2.

2.5.1 Some definitions

Lets assume we want to test if a single gene, say gene i, is differentially expressed between treatment and control conditions. Now the null hypothesis would be that of no differential expression and we can express it as $H_i : F_{iTr} = F_{iC}$ versus an alternative hypothesis $K_i : \mu_{iTr} \neq \mu_{iC}$, where differences in treatment and control expected values are supposed to imply differential expression[14].

Let $s(\mathbf{X}_{i\cdot})$ be a statistic with realization $s(\mathbf{x}_{i\cdot})$ and cumulative distribution function G_i, for which large values imply strong evidence against H_i and in favour of K_i. Also, let δ_t be a decision rule such that

$$\delta_t(\mathbf{x}_{i\cdot}) = \begin{cases} 1, & s(\mathbf{x}_{i\cdot}) \in [t, \infty) \quad \longrightarrow \quad \text{We reject } H_i, \\ 0, & s(\mathbf{x}_{i\cdot}) \notin [t, \infty) \quad \longrightarrow \quad \text{We don't reject } H_i. \end{cases} \tag{2.18}$$

As usual, a Type I Error consists in rejecting H_i when it is true, and a Type II Error consists in not rejecting H_i when K_i is true. Finally, the significance of the test using δ_t is

$$\alpha_t = P(\text{“Type I Error”}) = P_{H_i}\left(s\left(\mathbf{X}_{i\cdot}\right) \geq t\right) = 1 - G_{iH_i}(t),$$

where P_{H_i} and G_{iH_i} refer to the probability measure in $(\Omega, \mathcal{F})$ and the cumulative distribution function of $s(\mathbf{X}_{i\cdot})$ when H_i is true. In the single hypothesis paradigm, one sets a desired significance level α^* and chooses t so that $\alpha_t \leq \alpha^*$.

When detecting differentially expressed genes, we deal with testing $H_1, \ldots, H_n$ simultaneously. Let $\mathcal{G} = \{1, \ldots, n\}$ be the set of genes and $\mathcal{H} = \{i \in \mathcal{G} : H_i \text{ is true}\}$, with cardinality n_0, be the set of genes for which the null hypothesis is true, i.e., the set of non differentially expressed genes. Also, for fixed t and rejection region $[t, \infty)$, let $\mathcal{R}_t = \{i \in \mathcal{G} : s(\mathbf{X}_{i\cdot}) \geq t\}$, with cardinality $R(t) = R$, be the set of genes for which the null hypothesis is rejected, and let $\mathcal{V}_t = \{i \in \mathcal{G} : s(\mathbf{X}_{i\cdot}) \geq t,\ H_i \text{ is true}\} = \mathcal{R}_t \cap \mathcal{H}$, with cardinality $V(t) = V$, be the set of false positives, that is, the set of genes for which the null hypothesis is rejected despite being true. Note that R and V are random variables with realizations $r = \#\{i \in \mathcal{G} : s(\mathbf{x}_{i\cdot}) \geq t\}$ and $v = \#\{i \in \mathcal{G} : s(\mathbf{x}_{i\cdot}) \geq t,\ H_i \text{ is true}\}$,

[13] For a recount of other approaches to multiple hypothesis testing and a thorough theoretical treatment see Dudoit & Van Der Laan (2008).

[14] Note that other location parameters might be used as well.

respectively. The possible outcomes of testing $H_1, \ldots, H_n$ for fixed t are depicted in Table 2.1. Here, $W = n - R$ and $U = n_0 - V$.

TABLE 2.1. Possible outcomes when testing n hypothesis simultaneously.

Hypothesis	Accept	Reject	Total
Null true	U	V	n_0
Alternative true	$W - U$	$R - V$	$n - n_0$
Total	W	R	n

Adapted from Storey (2002).

Now, the ideal (though generally unattainable) outcome in multiple hypothesis tests is $W \equiv U$ and $V \equiv 0$, so that all true alternative hypothesis are detected (no Type II Errors) and no true null hypothesis are rejected (no Type I Errors). When detecting differentially expressed genes, one is more concerned with false positives, hence priority is given to control of Type I Errors. Type II Error reduction may then be achieved by a judicious choice of the test statistic (Efron & Tibshirani, 1993, p. 211).

2.5.1.1 Family-Wise Error Rate

The family-wise error rate ($FWER$) is the most stringent measure for controlling Type I Errors in multiple hypothesis tests. It is defined as (Dudoit et al., 2002):

$$FWER = P(\text{“At least one Type I Error”}) = P(V \geq 1). \tag{2.19}$$

Most methods that control $FWER$ when detecting differentially expressed genes test for $H_i : E(X_1) = E(X_2)$ versus $K_i : E(X_1) \neq E(X_2)$, where $X_1 \sim F_1$ and $X_2 \sim F_2$, and use Welch t-statistics and their p–values to derive ‘adjusted p–values’ as their test statistics (Dudoit et al., 2002). The best known (though most stringent) of such methods is the Bonferroni procedure:

Let p_i be the (unadjusted) p–value of the i-th gene's single hypothesis test using Welch t-statistic, for $i = 1, \ldots, n$. Define the adjusted p–values $\tilde{p}_i = \min\{np_i, 1\}$ and reject the null hypothesis for those genes with $\tilde{p}_i \leq \alpha^*$. For this procedure, $FWER \leq \alpha^*$.

There are two downsides with this approach. First, it is too conservative: for very large n (as is usually the case in microarray experiments) very few, if any, null hypotheses will be rejected. Second, for small p, one has to know (or make some strong assumptions about) the Welch t-statistics' exact joint distributions to compute the adjusted and unadjusted p–values.

To achieve more power, several modifications have been made to the Bonferroni Procedure, among which is worth mentioning the Westfall & Young (1993) *step-down miniP adjusted p–value* and *step-down maxT adjusted p–value* procedures, for they take into account the dependence structures among the genes[15].

As to the second downside, Dudoit et al. (2002) extend Westfall & Young (1993) procedures by estimating the adjusted and unadjusted p–values with permutation distributions,

[15]When independence among the tests is a reasonable assumption, the reader is referred to the multiple testing procedures presented in Shaffer (1995).

just as permutation tests in Section 2.4.3.1. Dudoit & Van Der Laan (2008) extend this idea by estimating adjusted and unadjusted p–values with bootstrap achieved significance levels as in Section 2.4.3.2.

Despite such improvements, methods that control $FWER$ remain too conservative and, loose power as n increases. Additionally, in the context of identifying differentially expressed genes, control of the $FWER$ produces a somewhat inappropriate measure of Type I Error. Given a group of genes identified as differentially expressed, more than the probability of whether any Type I Error was made ($FWER$), real interest lies in the number of falsely rejected hypothesis and the proportion of false positives among the group of identified genes (Storey, 2002).

2.5.1.2 False Discovery Rate (FDR)

The previous considerations led to the definition of the false discovery rate (FDR) by Benjamini & Hochberg (1995) as the expected proportion of falsely rejected null hypothesis. For this, define the random variable $Q = V/R$ if $R > 0$ and $Q = 0$ otherwise. Then, the FDR is computed as

$$FDR = E(Q) = E\left(\frac{V}{R}\middle| R > 0\right) P(R > 0), \tag{2.20}$$

where the expectation is taken under the true distribution P instead of the complete null distribution P_H, H being the complete null hypothesis $\bigcap_{i=1}^{n} H_i$. Two important properties arise from this definition:

1. If H is true (all null hypothesis are true), then $FDR = FWER$. In this case, $V = R$ so $FDR = E(1)P(R > 0) = P(V \geq 1) = FWER$.

2. If H is not true (not all null hypothesis are true), then $FDR \leq FWER$. In this case $V \leq R$ so $I\{V \geq 1\} \geq Q$. Taking expectations on both sides, we get $FWER = P(V \geq 1) \geq E(Q) = FDR$.

In general, then, $FDR \leq FWER$ so while controlling $FWER$ amounts to controlling FDR, the reverse is not true, and, thus, controlling only the FDR results in less strict procedures and increased power (Benjamini & Hochberg, 1995).

Benjamini & Hochberg (1995) proposed a procedure for controlling the FDR that consists of fixing a desired FDR level α^* and determining a suitable rejection region based on the test statistics' p–values. More specifically, let $p_{[1]} \leq p_{[2]} \leq \cdots \leq p_{[n]}$ be the ordered p–values of the test statistics for testing $H_{[1]}, \ldots, H_{[n]}$, respectively. Now, reject $H_{[i]}$ for $i = 1, \ldots, k$, where k is the largest integer for which $np_{[k]} \leq k\alpha^*$. They proved that if the test statistics are independent, then for the above procedure $FDR \leq \alpha^*$ for every $n_H = 0, 1, \ldots, n$ [16]. Benjamini & Yekutieli (2001) extended this procedure to the case when the test statistics present positive regression dependence.

In practice, the statistics' distributions are unknown, and, thus, the p–values and the cutoff k have to be estimated from the data. In this sense, Benjamini & Hochberg (1995)'s procedure amounts to fixing a desired FDR level and estimating a rejection region (determined by $\hat{k}$) that approximately achieves the desired FDR. Storey & Tibshirani (2001),

[16] See Theorem 1 in Benjamini & Hochberg (1995).

Storey (2002) and Storey & Tibshirani (2003) applied this approach to the identification of differentially expressed genes, estimating the statistics' distributions and corresponding p–values via permutation and bootstrap distributions and extended it for general statistics other than p–values.

Despite the advantages of Benjamini & Hochberg (1995)'s FDR over the $FWER$, the FDR still presents some downsides for identifying differentially expressed genes (Storey & Tibshirani, 2001; Storey, 2002; Storey, 2003). First, if H is true, every rejected hypothesis H_i constitutes a Type I Error and one would expect the FDR to be 1 instead of being equal to the $FWER$. Second, being an expectation, controlling the FDR only controls the ratio V/R in the long run and the actual ratio v/r may well be above the desired level α^*. Third, if after performing the test one or more null hypothesis were rejected, that is, conditioning on $R > 0$, the expected proportion of false positives is only being controlled at level $\alpha^*/P(R > 0)$.

2.5.1.3 Positive False Discovery Rate (pFDR)

The positive false discovery rate ($pFDR$) was introduced by Storey (2002) as a modification to Benjamini & Hochberg (1995)'s FDR, motivated by the previous considerations. It consists of the expected proportion of false positives conditioning on $R > 0$: that is,

$$pFDR = E\left(\frac{V}{R}\middle| R > 0\right). \tag{2.21}$$

Some properties arise from this definition:

1. For t such that[17] $P(s(\mathbf{X}_{i\cdot}) \geq t) > 0$, $i = 1, \ldots, n$, $\lim_{n\to\infty} P(R > 0) = 1$. Then, for large n, there is little harm in assuming $R > 0$. Also, $\lim_{n\to\infty} FDR = pFDR$ (Storey et al., 2004).

2. If after performing multiple tests no null hypothesis was rejected, then there is no possibility of making a Type I Error and, therefore, the expected proportion of false positives (conditioning on $R > 0$ or not) is of no interest and doesn't have to be defined for this particular case (Storey, 2003).

3. Note that $pFDR \geq FDR$, so controlling the $pFDR$ implies control of the FDR.

4. If all null hypothesis are true ($V = R$), then $pFDR = 1$, as desired.

Therefore, $pFDR$ is a more appropriate measure than FDR and $FWER$ for controlling Type I Errors in multiple hypothesis testing when n is large (Storey & Tibshirani, 2001; Storey, 2002; Storey, 2003; Storey & Tibshirani, 2003).

The last property, however, makes it impossible to apply the usual multiple hypothesis testing paradigm to control the $pFDR$. Indeed, if H is true, then $pFDR \equiv 1$ and there is no rejection region such that $pFDR \leq \alpha^* < 1$ (Storey, 2003). As a result, to perform multiple tests controlling the $pFDR$, Storey (2002) proposed to choose a rejection region and, then, estimate its $pFDR$; instead of fixing a desired $pFDR$ and estimating a suitable rejection region. Storey et al. (2004) proved that in terms of the control for FDR and $pFDR$, both types of procedures are asymptotically equivalent.

[17] Note that this is a very reasonable condition: if it doesn't hold, the test has no power and is of no use.

As the FDR, the $pFDR$ is also an expectation and, therefore, controlling the $pFDR$ only controls the ratio V/R in the long run, while the actual realization v/r may exceed the desired level α^*. Hence, Storey & Tibshirani (2001) recommend to compute upper confidence bounds for $pFDR$ to asses the possible magnitude of the discrepancy. Fortunately (Storey, 2003, Theorem 4), under some general conditions, the FDR, the $pFDR$ and the realized $v/\max\{r, 1\}$ converge to the same limit as $n \to \infty$; so, for large n, these discrepancies will be small. The remainder of this chapter focuses on estimation and multiple hypothesis testing when controlling for the $pFDR$.

2.5.2 Estimation of the pFDR under independence

Storey & Tibshirani (2001) proposed a well behaved finite sample estimator for the $pFDR$, given a rejection region $[t, \infty)$ and when $s(\mathbf{X}_{i\cdot})$ are independent and have identical null distributions. The following Theorem from Storey (2003) is required:

Theorem 1. *Suppose n identical hypothesis tests are performed with the statistics $S_i = s(\mathbf{X}_{i\cdot}), i = 1, \ldots, n$, and rejection region $[t, \infty)$. Assume that (S_i, H_i) are iid random variables with $S_i|H_i \sim (1 - H_i)G_0 + H_iG_1$ for some null distribution G_0 and alternative distribution G_1, and $H_i \sim$ Bernoulli(π_1), for $i = 1, \ldots, n$. Then*

$$pFDR(t) = P(H = 0|S \geq t), \tag{2.22}$$

where $\pi_0 = 1 - \pi_1$ is the implicit prior probability used in the above posterior probability.

Here, the null hypothesis for gene i being true depends on the random variable H_i being zero. From (2.22), we obtain

$$pFDR(t) = \frac{\pi_0 P(S \geq t|H = 0)}{P(S \geq t)} = \frac{\pi_0 P_H(S \geq t)}{P(S \geq t)} = \frac{\pi_0 \alpha_t}{P(S \geq t)}, \tag{2.23}$$

where α_t is the significance of a single test using $[t, \infty)$ as rejection region. For large n and under some general conditions (see Theorem 2 in Storey & Tibshirani (2001)), (2.23) holds approximately, even when the H_i are not random and the test statistics S_i are dependent in finite blocks.

The formula (2.23) gives a natural way of estimating $pFDR(t)$ when the tests statistics are independent and have the same null distribution. The plug-in estimator of $P(S \geq t)$ is $R(t)/n$.

On the other hand, $\alpha_t = P_H(S \geq t) = 1 - G_H(t)$, where G_H can be approximated by G_{perm} or $\tilde{G}_B$ via permutation or bootstrap methods from Sections 2.4.3.1 and 2.4.3.2. More specifically, for B bootstrap or permutation replications $(s^*_{1b}, \ldots, s^*_{nb})$, $b = 1, \ldots, B$, the estimate of α_t is

$$\hat{\alpha}_t = \frac{1}{B}\sum_{b=1}^{B} \frac{\#_{i=1,\ldots,n}\{s^*_{ib} \geq t\}}{n} = \frac{1}{n}\left[\frac{1}{B}\sum_{b=1}^{B} r^*_b(t)\right] = \frac{\hat{E}_H(R(t))}{n},$$

where $r^*_b(t) = \#\{s^*_{ib} \geq t : i = 1, \ldots, n\}$.

Regarding π_0, let $\Gamma_\alpha = [t_\alpha, \infty)$ be the rejection region for which the significance of a single test $P_H(S(\mathbf{X}_{i\cdot}) \geq t_\alpha) = \alpha$, and note that $\{\Gamma_\alpha\}$ is a nested set of rejection regions, that is, if $\alpha < \alpha'$ then $\Gamma_\alpha \subset \Gamma_{\alpha'}$. Now, for a well chosen $0 < \lambda < 1$, we expect $n_0(1 - \lambda)$ of

the tests to have a p–value in the interval $(\lambda, 1]$, and, therefore, we also expect $n_0(1-\lambda)$ of the test statistics to fall outside Γ_λ. Note also that, for n large enough, $\pi_0 = n_0/n$. Then, for a well chosen λ, we can estimate π_0 as (Storey, 2002):

$$\hat{\pi}_0(\lambda) = \frac{\#\{S_i < t_\lambda\}}{n(1-\lambda)} = \frac{n - R(t_\lambda)}{n(1-\lambda)} = \frac{W(t_\lambda)}{n(1-\lambda)}.$$

Although identifying Γ_λ exactly for a given λ may not be straight forward, we can use $t_\lambda = G_H^{-1}(1-\lambda)$ to estimate t_λ using the bootstrap or permutation estimates of G_H, $\tilde{G}_B$ and G_{perm}.

Storey & Tibshirani (2001) proved that $\hat{\pi}_0(\lambda)$ is conservatively biased for $0 < \lambda < 1$ and that there is a tradeoff between bias and variance as λ varies: increases in λ produce larger variances but smaller bias. Storey (2002) proposed a method for finding the value of λ that minimizes $\hat{\pi}_0(\lambda)$'s mean squared error for independent test statistics and Storey & Tibshirani (2001) generalized it for dependent test statistics. However, to ease the computational burden of our method, we will follow Storey et al. (2004), Taylor et al. (2005) and Li & Tibshirani (2013) in setting $\lambda = 0.5$.

Replacing into (2.23), we obtain the following estimator for $pFDR$ (Storey, 2002):

$$\hat{Q}_\lambda(t_\alpha) = \frac{\hat{\pi}_0 \hat{\alpha}}{\hat{P}(S \geq t_\alpha)} = \frac{\hat{\pi_0} \hat{E}_H(R(t_\alpha))}{R(t_\alpha)}. \tag{2.24}$$

Storey (2002) proved that $E(\hat{Q}_\lambda(t)) \geq pFDR(t)$ for all $t \in \mathbb{R}$, and π_0, so $\hat{Q}_\lambda(t)$ offers strong conservative control of the $pFDR$ for finite samples. Also, if conditions of Theorem 1 hold, $\hat{Q}_\lambda(t)$ is a maximum likelihood estimator of

$$\frac{\pi_0 + \pi_1[1 - g(\lambda)]/(1-\lambda)}{\pi_0} pFDR(t) \geq pFDR, \tag{2.25}$$

where $g(\lambda) = P_K(S \geq t_\lambda)$ [18] (Storey, 2002, Theorem 5). As a result, $\hat{Q}_\lambda(t)$ is a consistent estimator for the left value of (2.25) and is, therefore, consistently conservative for the $pFDR$. The steps for computing $\hat{Q}_\lambda(\Gamma)$ are depicted in Algorithm 2.5.1, adapted from Storey & Tibshirani (2001) and Storey (2002).

For small sample sizes, $R(t)$ can be zero with positive probability and (2.24) needs some modifications to be well defined (Storey & Tibshirani, 2001; Storey, 2002). On the contrary, if n is large (as it tends to be the case in microarray experiments) under conditions of Theorem 1 and for non trivial tests[19], we can safely assume $P(R(t) = 0) = 0$.

[18] If K is composite, then g is formed as an appropriate mixture of alternative distributions (Storey, 2002).
[19] Choosing t so that $P(S \geq t) > 0$. As a result, $\lim_{n\to\infty} P(R(t) > 0) = 1$.

Algorithm 2.5.1: Estimation of the $pFDR$ when testing n hypothesis simultaneously, for fixed $[t, \infty)$ and λ.

1. For large B, compute the bootstrap or permutation replications of $s(\mathbf{x}_{1\cdot}), \ldots, s(\mathbf{x}_{n\cdot})$, obtaining $s^*_{1b}, \ldots, s^*_{nb}$ for $b = 1, \ldots, B$.
2. Compute $\hat{E}_H(R(t))$ as
$$\hat{E}_H(R(t)) = \frac{1}{B}\sum_{b=1}^{B} r^*_b(t),$$
where $r^*_b(t) = \#\{s^*_{ib} \geq t : i = 1, \ldots, n\}$.
3. Set $[t_\lambda, \infty)$ as rejection region with t_λ as the $(1-\lambda)$-th percentile of the bootstrap or permutation replications of step 1. Estimate π_0 by
$$\hat{\pi}_0(\lambda) = \frac{w(t_\lambda)}{n(1-\lambda)},$$
where $w(t_\lambda) = \#\{s(\mathbf{x}_{i\cdot}) < t_\lambda : i = 1, \ldots, n\}$.
4. Estimate $pFDR_\lambda(t)$ as
$$\hat{Q}_\lambda(t) = \frac{\hat{\pi}_0 \hat{E}_H(R(t))}{r(t)},$$
where $r(t) = \#\{s(\mathbf{x}_{i\cdot}) \geq t : i = 1, \ldots, n\}$.

Adapted from Storey & Tibshirani (2001) and Storey (2002).

2.5.3 The q–value

In addition to the $pFDR$, Storey (2002) proposed the *q–value* as the analogue of the p–value when controlling the $pFDR$ in multiple hypothesis testing. For an observed statistic $s(\mathbf{x}_{i\cdot})$, the q–value is defined as the minimum $pFDR$ that can occur when rejecting all hypothesis for which $s(\mathbf{x}_{i'\cdot}) \geq s(\mathbf{x}_{i\cdot}), i' = 1, \ldots, n$. More specifically:

$$\text{q–value}(s(\mathbf{x}_{i\cdot})) = \inf_t \{pFDR(t) : s(\mathbf{x}_{i\cdot}) \geq t\}. \tag{2.26}$$

Also, under the conditions of Theorem 1, Storey (2003) shows that

$$\text{q–value}(s(\mathbf{x}_{i\cdot})) = P(H = 0 | S \geq s(\mathbf{x}_{i\cdot})),$$

so the q–value can be interpreted as the posterior probability of making a Type I Error when testing n hypothesis with rejection region $[s(\mathbf{x}_{i\cdot}), \infty)$.

As $\hat{Q}_\lambda(t)$ is not necessarily decreasing in t, we estimate the q–values following Algorithm 2.5.2, adapted from Storey (2002).

Algorithm 2.5.2: Estimation of the q–values for testing n hypothesis simultaneously.

1. Let $s_{[1]} \leq s_{[2]} \leq \cdots \leq s_{[n]}$ be the ordered statistics for $s_{[i]} = s(\mathbf{x}_{[i]\cdot})$, for $i = 1, \ldots, n$.
2. Set $\hat{q}(s_{[1]}) = \hat{Q}_\lambda(s_{[1]})$.
3. Set $\hat{q}(s_{[i]}) = \min\{\hat{Q}_\lambda(s_{[i]}), \hat{q}(s_{[i-1]})\}$ for $i = 2, \ldots, n$.

Adapted from Storey (2002).

2.5.4 A word on dependence

When the test statistics are not independent, the following theorem from Storey & Tibshirani (2001, Theorem 2) is required:

Theorem 2. *Suppose that*

$$\frac{V_n(t)}{\#\mathcal{H}_n} \to P_H(S \geq t) \qquad \textit{and} \qquad \frac{R_n(t) - V_n(t)}{n - \#\mathcal{H}_n} \to P_K(S \geq t),$$

in probability for some rejection region $[t, \infty)$ *with* $P(S \geq t) > 0$. *Then*

$$\lim_{n\to\infty} pFDR_n(t) = \frac{\pi_0 P_H(S \geq t)}{P(S \geq t)},$$

where $pFDR_n(t)$ *is the* $pFDR$ *of* $[t, \infty)$ *resulting from the first* n *statistics.*

If the conditions in Theorem 2 hold, we also have (Storey & Tibshirani, 2001)

$$\lim_{n\to\infty} \hat{Q}_\lambda(t) = \frac{\pi_0 + \pi_1[1 - g(\lambda)]/(1 - \lambda)}{\pi_0} pFDR(t) \geq pFDR(t),$$

where $g(\lambda)$ is a single test's power when using $[t_\lambda, \infty)$ as rejection region[20]. Hence, $\hat{Q}_\lambda(t)$ is still conservatively consistent.

Also (Storey, 2003, Theorem 4),

$$\lim_{n\to\infty} \left| \frac{V_n(t)}{\max\{1, R_n(t)\}} - \frac{\pi_0 P_H(S \geq t)}{P(S \geq t)} \right| = 0, \quad \text{almost surely.}$$

Therefore, for large n, the $pFDR$ and the actual (realized) ratio $v/\max\{1, r\}$ are very close and control of the former amounts to control of the latter. More specifically, if $P_H(S \geq t)$ and $P_K(S \geq t)$ in the conditions of Theorem 2 above are continuous functions of t, then, for every $\Delta < \infty$,

$$\liminf_{n\to\infty} \inf_{t\leq\Delta} \left\{ \hat{Q}_\lambda(t) - pFDR(t) \right\} \geq 0, \quad \liminf_{n\to\infty} \inf_{t\leq\Delta} \left\{ \hat{Q}_\lambda(t) - \frac{V_n(t)}{\max\{1, R_n(t)\}} \right\} \geq 0, \quad (2.27)$$

with probability 1 (Storey et al., 2004, Theorem 6). This means that, for n large, the previous results hold for all t simultaneously. Thus $\hat{Q}_\lambda(t)$ is conservatively consistent for the $pFDR$ and for the realized $v/\max\{1, r\}$, for all rejection regions of the form $[t, \infty)$,

[20] Again, if K is composite, then g is formed as an appropriate mixture of alternative distributions (Storey, 2002).

$t < \infty$. As a result, using a rejection region of the form $[t^*, \infty)$ such that $\hat{Q}_\lambda(t^*) \leq \alpha^*$ amounts to controlling the $pFDR$ at level α^*. Note that this is the case for every $0 < \lambda < 1$, as long as conditions of Theorem 2 hold. Also, remember that $pFDR \geq FDR$, so $\hat{Q}_\lambda(t)$ is also conservatively consistent for the FDR.

When detecting differentially expressed genes, the dependence structures between genes (and between test statistics) arise from groups of corregulated genes and occurs in finite blocks. Storey & Tibshirani (2001) assert that when this happens, conditions of Theorem 2 are satisfied. Then, it makes sense to use the $pFDR$ estimator of Algorithm 2.5.1 to identify differentially expressed genes in microarray data.

2.5.5 Choice of the null hypothesis

The previous estimators for the $pFDR$ require estimating G_H by bootstrap sampling or permutations of the treatment/control tags in the microarray dataset. This procedure implies that we are assuming the complete null hypothesis to be $H : F_{Tr} = F_C$, which can be too restrictive some times. In observational studies, for example, when testing equality of means between two populations in the presence of many unobserved covariates, one is not usually willing to assume equal variances under the simpler null hypothesis $\mu_1 = \mu_2$. Efron (2004) shows that under this circumstances, the use of permutation and bootstrap estimates of G_H is not entirely justified and proposes an empirical Bayes' method instead.

However, in the context of microarray experiments where experimental conditions are controlled and the only (allegedly) significant covariate is the treatment/control factor, $H : F_{Tr} = F_C$ is a reasonable null assumption and estimates of G_H via bootstrap or permutation methods are, indeed, justified (Dudoit & Van Der Laan, 2008). Finally, note that column–wise standardization of the data set (see Sections 2.2 and 2.3) enforces the plausibility of $H : F_{Tr} = F_C$ up to the distributions' second moments.

2.6 A word on scale

The estimation method of the $pFDR$ presented in Section 2.5 has been applied to various sets of microarray data to detect differentially expressed genes (Storey & Tibshirani, 2001; Taylor et al., 2005; Cai et al., 2013). The widely used Significance Analysis of Microarrays (as presented in Tusher et al., 2001) is a specially conservative version of the former that sets $\hat{\pi}_0 = 1$. However, as a rule, applications for microarray data use test statistics of the form

$$s(\mathbf{X}_{i\cdot}) = \frac{\bar{X}_{iTr} - \bar{X}_{iC}}{s.e.(\bar{X}_{iTr} - \bar{X}_{iC}) + c_0}, \tag{2.28}$$

where c_0 is a convenient constant (usually zero), or monotone functions of s as p–values. Using this type of statistics, it is plausible to assume that they all have the same null distribution, making the conditions of Theorem 2 more likely to hold.

Yet, dividing by the standard error in (2.28), one looses the inherent gene expression scale that lies within the data. Think of the following two biological scenarios for gene expression data:

Biological Scenario 1: All genes among all replicates have true positive expression levels when the sample is taken. Therefore, the major differences in scale between genes in a microarray data set are a due to external sources of variation.

Biological Scenario 2: Only a small proportion of the genes in each replicate has true positive expression levels when the tissue sample is taken and no systematic sources of variation other than control/treatment effects are present in the experiment. Therefore, the major differences in scale between genes in a microarray data set are due to whether a gene was actively producing proteins when the sample was taken.

If Scenario 1 holds, there is no relevant information in the differences between the scales in the rows of the data, and row standardization is in order. If, on the contrary, Scenario 2 seems more appropriate, the information contained in the differences between the scales of the rows is relevant for it allows to asses which genes had actual positive expression levels when the sample was taken.

On the other hand, the data for the genes that were not expressing themselves when the sample was taken is only the result of external sources of variation. Those genes, having true zero expression levels in both treatment and control replicates, cannot be classified as differentially expressed for they are not expressed to begin with. However, because n is very large and there might be systematic sources of variation (changes in printing tips, background intensity or other aspects of microarray technology), it is very likely that a considerable number of those genes with no expression will be classified as differentially expressed when using statistics of the form (2.28)[21].

We strongly believe that the first condition of Biological Scenario 2 is more reasonable in the context of microarray experiments. Additionally, there are several methods for normalization of microarray data that remove sources of variation other than the control/treatment effect without performing any kind of row standardization (Dudoit et al., 2002; Simon et al., 2003) that can be applied beforehand to ensure the technical part of Biological Scenario 2. Therefore, row standardization should not be performed, in order to avoid classifying genes with no expression as being differentially expressed.

[21] For an illustrative example, see Section 4.1.3.

CHAPTER 3

Proposed Strategy

In this chapter we present our proposed strategy for the identification of differentially expressed genes in microarray data using a multivariate approach. First, we present the construction of two artificial components (similar to principal components) that capture the multivariate structure and the inherent scale of the data for genetic differential expression. Then, we present the application of the methodology from Storey & Tibshirani (2001) using one of the artificial components as the test statistic, and we extend the analysis for when data sets are available for multiple time point. We close the chapter by introducing the two microarray data sets to be analysed in Chapter 4.

3.1 Artificial components

As was shown in Section 2.3.2, Principal Components Analysis constitutes a powerful tool for capturing multivariate structures in large data sets. In a general way, if Biological Scenario 2 from Section 2.6 holds, the first principal component will *mainly* represent overall expression, and the second will *mainly* represent differences in expression levels between conditions. Unfortunately, this interpretation of the principal components is, at best, approximate. For this reason, inferential procedures for differential expression using principal components as test statistics are not appropriate.

Therefore, in order to perform multiple tests regarding genetic differential expression, we need new components that capture (exactly) the genes' overall and differential expression levels. We call these components *artificial* because they do not arise naturally as the solution of a maximization problem. Instead, they are constructed deliberately to capture specific features of the data and, thus, have an exact interpretation. Their construction is as follows.

For $i = 1, \ldots, n$, let the overall, the treatment and the control means for gene i be

$$\bar{\mathbf{x}}_{i\cdot} = \frac{1}{p}\sum_{j=1}^{p} x_{ij}, \qquad \bar{\mathbf{x}}_{iTr} = \frac{1}{p_1}\sum_{j=1}^{p_1} x_{ij}, \qquad \bar{\mathbf{x}}_{iC} = \frac{1}{p_2}\sum_{j=p_1+1}^{p} x_{ij}.$$

Define

$$\psi_1(\mathbf{x}_{i\cdot}) = \psi_{1i} = \sqrt{p} \times \bar{\mathbf{x}}_{i\cdot}, \qquad \psi_2(\mathbf{x}_{i\cdot}) = \psi_{2i} = \frac{\sqrt{p_1 p_2}}{\sqrt{p_1 + p_2}} \left(\bar{\mathbf{x}}_{iTr} - \bar{\mathbf{x}}_{iC}\right). \tag{3.1}$$

Now, ψ_{1i} is just a multiple of the mean expression level of gene i across both conditions, so it captures its overall expression level. As the data has not been standardized by rows, $\psi_{1i} > \psi_{1i'}$ implies that gene i has a higher overall expression level than gene i' and, thus, $\boldsymbol{\psi}_1 = (\psi_{11}, \ldots, \psi_{1n})$ provides a natural scale for comparing expression levels between the genes in the microarray. In PCA's vocabulary (Lebart et al., 1995), $\boldsymbol{\psi}_1$ is a *size component*.

On the other hand, ψ_{2i} is a multiple of the difference between treatment and control mean expression levels, so it captures the extent to which gene i is differentially expressed. We call $\boldsymbol{\psi}_2 = (\psi_{21}, \ldots, \psi_{2n})$ a *differential expression component*. Large positive (negative) values of ψ_2 indicate high (low) expression levels in the treatment replicates and low (high) expression levels in the control replicates.

The multiplicative constants in (3.1) are defined so that $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are the result of an orthogonal projection via unit projection vectors as in the PCA framework. Note that $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ can be computed as:

$$\boldsymbol{\psi}_1 = X\mathbf{v}_1, \qquad \boldsymbol{\psi}_2 = X\mathbf{v}_2, \tag{3.2}$$

where $\mathbf{v}_1 = (1, \ldots, 1)/\sqrt{p}$ and $\mathbf{v}_2 = (p_2, \cdots, p_2, -p_1, \cdots, -p_1)/\sqrt{p_1 p_2 p}$, with p_1 positive entries and p_2 negative entries, are orthogonal and have unit norm. In particular, if $p_1 = p_2$, $\mathbf{v}_2 = (1, \ldots, 1, -1, \ldots, -1)/\sqrt{p}$ and $\psi_{2i} = (\bar{\mathbf{x}}_{iTr} - \bar{\mathbf{x}}_{iC})\sqrt{p}/2$.

Plotting the genes in the *artificial plane* ($\boldsymbol{\psi}_1$ vs. $\boldsymbol{\psi}_2$) gives useful insight into the structure of the microarray data and the general behaviour of the genes. Because $\boldsymbol{\psi}_1$ and $\boldsymbol{\psi}_2$ are centered, genes near the origin characterize the *mean behaviour* of the data. Genes to the right (left) of the plane have high (low) overall expression levels with respect to that mean behaviour, and genes in the top (bottom) of the plane will be over (under) expressed with respect to that mean behaviour. Moreover, if Biological Scenario 2 from Section 2.6 holds, one would expect most of the genes to be gathered around the origin and only a few to the right of the plane. Differentially expressed genes should be near the right-top and right-bottom corners of the plane. Also, no genes should lie far to the left of the plane.

Finally, assessment of the amount of useful information in the data regarding differential expression can be made via the inertia projected onto $\mathbf{v}_2$ from (2.3) and the corresponding inertia ratios from (2.7) and (2.8).

3.2 Single time point analysis

In this section we present our method for identifying differentially expressed genes for a single time point. It consists of a specific application of Storey & Tibshirani (2001)'s methodology, by using a statistic that is well suited for microarray data when Biological Scenario 2 from Section 2.6 holds, thus controlling both FDR and $pFDR$ while maintaining a multivariate point of view. The statistic in question is $|\psi_2(\mathbf{X}_{i\cdot})|$ as defined in (3.1).

3.2.1 Estimation

Our method for identifying differentially expressed genes in microarray data for a single time point is presented in Algorithm in 3.2.1. Functions in the `R` programming language (R Core Team, 2014) for performing Algorithm 3.2.1 are presented in the Appendix.

Algorithm 3.2.1: Identification of differentially expressed genes in microarray data for a single time point.

1. Compute $\psi_{2i} = \psi_2(\mathbf{x}_{i\cdot})$ for $i = 1, \ldots, n$ from (3.1).
2. For each t in $\mathcal{T} = \{|\psi_{21}|, \ldots, |\psi_{2n}|\}$ and B large enough, compute $\hat{Q}_\lambda(t)$ as in Algorithm 2.5.1 with $\lambda = 0.5$ and using $s(\cdot) = |\psi_2(\cdot)|$ from (3.1).
3. Set a desired $pFDR$ level α^* and compute $t^* = \inf\left\{t \in \mathcal{T} : \hat{Q}_{0.5}(t) \leq \alpha^*\right\}$.
4. Identify the set of differentially expressed genes as:

$$\mathcal{R}_{t^*} = \{i : |\psi_2(\mathbf{x}_{i\cdot})| \geq t^*\}.$$

   The down-regulated and up-regulated sets of genes are:

$$\mathcal{D}_{t^*} = \{i : \psi_2(\mathbf{x}_{i\cdot}) \leq -t^*\}, \qquad \mathcal{U}_{t^*} = \{i : \psi_2(\mathbf{x}_{i\cdot}) \geq t^*\}.$$

5. Compute the corresponding q–values using Algorithm 2.5.2.

In Section 2.5, we presented two distinct approaches in multiple hypothesis testing for controlling the $pFDR$: one fixating a desired FDR level and estimating a rejection region, the other fixating a rejection region and estimating its $pFDR$ (note that, for large n, $pFDR$ and FDR are equivalent). Storey et al. (2004) showed that this two approaches are asymptotically equivalent. Specifically, define

$$t_\alpha(\hat{Q}_\lambda) = \inf\left\{t : \hat{Q}_\lambda(t) \leq \alpha\right\}, \qquad 0 \leq \alpha \leq 1.$$

Then, rejecting all null hypothesis with $s(\mathbf{X}_{i\cdot}) \geq t_\alpha(\hat{Q}_\lambda)$ amounts to controlling the $pFDR$ at level α, for n large enough (see Section 2.5.4). Thus, the procedure in Algorithm 3.2.1 controls the $pFDR$ at level α^*.

Finally, the following observations about Algorithm 3.2.1 are in order:

1. We only estimate the $pFDR$ for $t \in \mathcal{T}$ because those are the values of t for which the number of rejected hypothesis actually changes. More specifically, let $t_{[1]}, \ldots, t_{[n]}$ be the ordered values of $\mathcal{T}$. Then, using $[t_{[n-k+1]}, \infty)$ as the rejection region, produces k genes identified as differentially expressed, for $k = 1, \ldots, n$.

2. For computational ease, we set $\lambda = 0.5$, following Storey et al. (2004), Taylor et al. (2005) and Li & Tibshirani (2013). However, a more suitable λ in terms of the Mean Square Error of $\hat{Q}_\lambda(t)$ can be computed via bootstrap methods as shown in Storey & Tibshirani (2001, Section 6).

3. The estimation of $\hat{Q}_{0.5}(t)$ in step 2 may be done by using permutation or bootstrap estimates of the statistics' null distribution (see Section 2.5). Though permutation methods are more popular (Li & Tibshirani, 2013), we favor bootstrap estimates

of the null distribution for ease of interpretation (see Sections 2.4.3.1–2.4.3.2 and Dudoit & Van Der Laan, 2008, p. 65). In any case, for large B, the results should be very similar (Efron & Tibshirani, 1993).

4. $B = 100$ should be enough for obtaining accurate and stable estimates in step 2 (Efron & Tibshirani, 1993). However, depending on the data and the shape of the FDR, small changes in the estimated FDR may produce large changes in t^* and, hence, a much larger value of B may be needed to guarantee stability of the groups of up and down regulated genes.

3.2.2 Assumptions

Until now, we have made several probabilistic, technical and biological assumptions that are necessary for procedure in Algorithm 3.2.1 to effectively control the $pFDR$. We recall them in Table 3.1.

TABLE 3.1. Biological, technical and probabilistic assumptions.

No.	Type	Assumption
1.	Biological	Only a small proportion of the genes have true positive expression levels when the sample is taken (first condition in Biological Scenario 2, Section 2.6).
2.	Technical	All systematic sources of variation in the microarray data other than the control/treatment effects have been removed (second condition in Biological Scenario 2, Section 2.6).
3.	Probabilistic	Assumptions of the General Probability Model of Section 2.2 hold.
4.	Probabilistic	Assumptions of Theorem 2 from Section 2.5.4 hold and the limits of $P_H(S \geq t)$ and $P_K(S \geq t)$ as $n \to \infty$ are continuous functions of t.
5.	Probabilistic	n is large enough for the asymptotic results of Section 2.5.4 to hold.

As discussed in Section 2.6, we consider Assumption 1 to be reasonable in the context of microarray data. Assumptions 3 and 5, being very general, seem also reasonable in the context of differential genetic expression and will not be examined any further.

Regarding Assumption 2, there are several methods for normalizing microarray data in order that no systematic sources of variation remain unaccounted for (Dudoit et al., 2002; Dudoit & Van Der Laan, 2008; Simon et al., 2003). However, if those methods perform some sort of row–standardization, the inherent scale of the genes' expression levels may be lost and we advise against it (see Section 2.6). Note, however, that column–wise standardization (as required by our method) already removes some of the systematic sources of variation between the replicates.

As for Assumption 4, Storey & Tibshirani (2001) and Storey et al. (2004) argue that it is likely to hold under gene dependence in finite blocks and certain mixing of distributions for the statistics under the null and alternative hypotheses. In the context of differential expression in microarray data this seems a reasonable scenario. Actual verification of Assumption 4, however, is beyond the scope of this work.

3.2.3 Further assessments

As was seen in Section 2.5.4, as B, n and p grow, $\hat{Q}_\lambda$ approaches from above both the $pFDR$ and the realized proportion of false positives among all rejected null hypothesis. In practice, however, because B, n and p are finite, the control achieved using $\hat{Q}_\lambda$ is only approximate and some additional assessments are needed.

Storey & Tibshirani (2001) suggested the use of a bootstrap percentile confidence upper bound for the $pFDR$ to provide a somewhat more precise notion of the actual control achieved, but concluded that percentile upper bounds were not appropriate as they under estimated the actual confidence upper bound. We overcome this limitation by computing a BCa upper confidence bound for the $pFDR$ as shown in Algorithm 3.2.2. We find plots of $\hat{Q}_\lambda(t)$ and the $pFDR$'s upper confidence bound vs. t to be very informative as to the $pFDR$ control actually achieved. Functions in the `R` programming language (R Core Team, 2014) for performing Algorithm 3.2.2 are presented in the Appendix.

Algorithm 3.2.2: Computation of a BCa upper confidence bound for the $pFDR$.

1. Compute $\hat{Q}_\lambda$ by applying steps 1 and 2 of Algorithm 3.2.1 to X.
2. Compute a large number R of independent bootstrap samples from X, $X_1^*, \ldots, X_R^*$, where $X_r^* = (\mathbf{x}_{\cdot j_1}, \ldots, \mathbf{x}_{\cdot j_{p_1}}, \mathbf{x}_{\cdot k_1}, \ldots, \mathbf{x}_{\cdot k_{p_2}})$, with $(j_1, \ldots, j_{p_1})$ being random sample with replacement from $\{1, \ldots, p_1\}$ and $(k_1, \ldots, k_{p_2})$ being a random sample with replacement from $\{p_1 + 1, \ldots, p_1 + p_2\}$.
3. Compute bootstrap replicates of $\hat{Q}_\lambda$, $\hat{Q}_\lambda^{*(1)}, \ldots, \hat{Q}_\lambda^{*(R)}$, by applying steps 1 and 2 of Algorithm 3.2.2 to X_r^*, $r = 1, \ldots, R$, using the set $\mathcal{T}$ from step 1.
4. For each t in $\mathcal{T}$ and the desired confidence level γ:

   4.1 Compute $z_0(t)$ from (2.12) as

   $$z_0(t) = \Phi^{-1}\left(\frac{\#\{\hat{Q}_\lambda^{*(r)}(t) < \hat{Q}_\lambda(t)\}}{R}\right)$$

   4.2 Compute $\hat{a}(t)$ from (2.13) as

   $$\hat{a}(t) = \frac{\sum_{j=1}^{p}[\hat{Q}_{jack}(t) - \hat{Q}_{(j)}(t)]^3}{6\{\sum_{j=1}^{p}[\hat{Q}_{jack}(t) - \hat{Q}_{(j)}(t)]^2\}^{3/2}},$$

   where $\hat{Q}_{(j)}(t)$ is the mean of the bootstrap replicates $\hat{Q}_\lambda^{*(r)}(t)$ for which the bootstrap indexes $(j_1, \ldots, j_{p_1}, k_1, \ldots, k_{p_2})$ in step 2 do not contain j, and $\hat{Q}_{jack}(t)$ is just $p^{-1}\sum_{j=1}^{p}\hat{Q}_{(j)}(t)$.

   4.3 Compute the upper γ confidence bound for the $pFDR$ from (2.11) as

   $$Q_t[\gamma] = \tilde{G}_t^{-1}\left(\Phi\left[z_0(t) + \frac{z_0(t) + z_\gamma}{1 - \hat{a}(t)\,(z_0(t) + z_\gamma)}\right]\right),$$

   where $\tilde{G}_t$ is the empirical cumulative distribution function of $\hat{Q}_\lambda^{*(1)}(t), \ldots, \hat{Q}_\lambda^{*(R)}(t)$, and z_γ is the γ percentile of a standard normal distribution.

Technically, $Q_t[\gamma]$ is a BCa upper γ confidence bound for $E[\hat{Q}_\lambda(t)] \geq pFDR(t)$. Because $\hat{Q}_\lambda(t)$ is conservatively biased, $Q_t[\gamma]$ is a γ^* confidence upper bound for $pFDR(t)$ with $\gamma^* \geq \gamma$, and we say that $Q_t[\gamma]$ is a *conservative* γ confidence upper bound for both $pFDR(t)$ and FDR (since $pFDR \geq FDR$). Moreover, if n is large, $pFDR \approx v/\max\{1, r\}$, so $Q_t[\gamma]$ is also a conservative γ confidence upper bound for the realized proportion of false positives.

However, $Q_t[\gamma]$ is only second order accurate (see Section 2.4.2), so, unless assumptions in (2.10) hold exactly, $P(E[\hat{Q}_\lambda(t)] \leq Q_t[\gamma]) = \gamma + O(p^{-1})$, $Q_t[\gamma]$ being the random variable and p the number of replicates in the microarray experiment. As p is usually small in microarray data, the approximation error must be kept in mind when analysing both $\hat{Q}_\lambda(t)$ and $Q_t[\gamma]$. Fortunately, the fact that $\hat{Q}_\lambda(t)$ and $Q_t[\gamma]$ are conservatively biased compensates, to some extent, this approximation error. Naturally, as p increases, the power of the multiple testing procedure, the precision of $\hat{Q}_\lambda(t)$ and the accuracy of $Q_t[\gamma]$ will improve.

Finally, the following observations about Algorithm 3.2.2 are in order:

1. In steps 1 and 3, $\hat{Q}_\lambda$ and $\hat{Q}^*_\lambda$ are functions of t defined for $t \in \mathcal{T}$, where $\mathcal{T}$ is the set of values for t in step 1.

2. In step 2, $X^*_r \leftarrow \tilde{F}$, where $\tilde{F}$ is the empirical distribution of X under the General Probability Model of Section 2.2.

3. $\hat{Q}_{(j)}(t)$ in step 4.2 is a bit different from (2.13). For large R, $\hat{Q}_{(j)}(t)$ is very close to the form (2.13) and it can save a considerable amount of computational effort (Efron & Tibshirani, 1993, p. 277).

4. The number of computations in Algorithm 3.2.2 is in the order of $R \times B \times n$ so a compromise must be made between R and B for obtaining comfortable computation times. For accurate bootstrap confidence intervals, $R = 1000$ should be enough (Efron & Tibshirani, 1993), so we recommend setting $B = 100$ and $R = 1000$. As n is usually very large, Algorithm 3.2.2 may require considerable computational effort.

3.3 Time course analysis

It is often the case in microarray experiments to have samples taken at different time points for analysing the genetical behaviour of the replicates in different stages of a disease or factor of interest. We propose two complementary extensions to our method in the single time point case for analysing time course microarray data. For the rest of this section, suppose we have L data sets $X^{(1)}, \ldots, X^{(L)}$ taken at time points $1, \ldots, L$. In Chapter 4 we present an example of this analysis.

3.3.1 Active vs. supplementary time points

The first approach consists of supposing that there is a single group of genes that, at some time point, become differentially expressed. The questions of interest, then, become which time point is the more suitable for detecting the group of differentially expressed genes and how do those genes behave through time.

In this setup one would expect that, as time passes, differential expression becomes more acute and easy to identify. However, different experimental conditions may occur between different time points and the signal to noise ratio may be lower in latter time points (Yuan & Kendziorski, 2006), so a more quantitative assessment is needed. Presently, we can use inertia ratios as defined in (2.8).

The amount of information about differential expression at a given time point l can be measured by $In^{(l)}(\mathbf{v}_2) = \mathrm{Var}(\boldsymbol{\psi}_2^{(l)})$, where $\boldsymbol{\psi}_2^{(l)}$ is the result of applying (3.2) to $X^{(l)}$, $l = 1, \ldots, L$. For this measure to be comparable between time points, we divide it by the maximum inertia that can be captured by a single direction as in (2.3), obtaining the inertia ratios:

$$R^{(l)}(\mathbf{v}_2) = \frac{In^{(l)}(\mathbf{v}_2)}{\lambda_1^{(l)}}, \qquad l = 1, \ldots, L, \tag{3.3}$$

where $\mathbf{v}_2 = (p_2, \cdots, p_2, -p_1, \cdots, -p_1)/\sqrt{p_1 p_2 p}$ as in (3.2) and $\lambda_1^{(l)}$ is the inertia projected onto the first principal component of $X^{(l)}$. Then, the data set that contains more information concerning differential expression, say $X^{(l^*)}$, is the one that maximizes $R^{(l)}(\mathbf{v}_2)$. We call l^* the *active* time point in the analysis.

Once l^* has been determined, Algorithms 3.2.1 and 3.2.2 can be applied to data set $X^{(l^*)}$ obtaining the respective groups of up and down regulated genes, $\mathcal{U}^{(l^*)}$ and $\mathcal{D}^{(l^*)}$. Then, plots of $\boldsymbol{\psi_1}^{(l)}$ vs. $\boldsymbol{\psi_2}^{(l)}$ can be made for each time point, coloring the genes in each group and, if needed, projecting the groups' means or centers of gravity as supplementary rows using (2.6) and the projection vectors $\mathbf{v}_1 = (1, \ldots, 1)/\sqrt{p}$ and $\mathbf{v}_2$ from above.

3.3.2 Groups conformation through time

The other approach supposes that there may be different genes with differential expression at different time points. The analysis here consists simply of applying Algorithms 3.2.1 and 3.2.2 to each data set $X^{(1)}, \ldots, X^{(L)}$ and comparing the groups of up and down regulated genes detected at each time point. As before, $\boldsymbol{\psi_1}^{(l)}$ vs. $\boldsymbol{\psi_2}^{(l)}$ plots, coloring of differentially expressed genes and projection of groups' centers of gravity are also very useful here.

In practice, we have found both approaches to work well and to provide complementary and useful insights. If one expects to have a single group of up regulated and a single group of down regulated genes at the end of the analysis, we recommend taking $\mathcal{U}^{(l^*)}$ and $\mathcal{D}^{(l^*)}$ from the first approach as reference, and assessing their behaviour through time using the second approach. If one is interested in analysing the changes in the groups of differentially expressed genes through time, the second approach is in order, and the first one can be used to get an idea of the intensity of the differential expression process at each time point.

3.4 Microarray data sets

In Chapter 4, we apply the previous method in two microarray data sets. We now present those data sets and the main characteristics of their respective microarray experiments.

3.4.1 Tomato inoculated with *P. Infestans* (PI) in the field

The first microarray data set was obtained from the Tomato Expression Database website (http://ted.bti.cornell.edu/), experiment E022 (Restrepo et al., 2005). Throughout the experiment, 8 tomato plants (line IL6-2) in field conditions where inoculated with *Phytophthora infestans*, and 8 control plants where mock-inoculated with sterile water. Leaf tissue samples from each replicate were taken at 12 hours before and 12, 36 and 60 hours after inoculation (hai). We refer to 12 hours before inoculation as the 0 hai time point.

mRNA was extracted from each sample and then hybridized on a cDNA microarray[1] (for more details of the experimental design and conditions of the study see Cai et al. (2013)). Expression levels were obtained for 13,440 genes. A portion of the microarray data at 60 hai is presented in Table 3.2. In the remainder of this work, we will refer to this data set as the PI data set.

TABLE 3.2. Data 60 hai from tomato plants inoculated with *P. infestans*.

	Inoculated (I)					Non-inoculated (NI)				
Gene	I1	I2	I3	$\cdots$	I8	NI1	NI2	NI3	$\cdots$	NI8
1	35	30	43	$\cdots$	29	34	30	55	$\cdots$	25
2	300	158	159	$\cdots$	82	640	602	246	$\cdots$	187
3	39	31	37	$\cdots$	27	40	31	47	$\cdots$	25
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
13,440	64	49	152	$\cdots$	38	58	63	81	$\cdots$	39

3.4.2 *Arabidopsis thaliana* inoculated with *A. tumefaciens* (AT)

The second microarray data set was obtained from Ditt et al. (2006)[2]. Throughout the experiment, mRNA samples were taken from 8 *Arabidopsis thaliana* cell cultures, 4 of which were inoculated with *Agrobacterium tumefaciens*, and 4 of which were mock–inoculated. mRNA samples were taken at 4, 12, 24 and 48 hai and hybridized to a cDNA microarray (for more details, see Ditt et al. (2006)).

Expression levels were obtained for 26,474 genes. Here, the reference sample was taken from a large pool of control cell cultures and the genes' expression levels correspond to the $\log_2$ of the ratios between the target and the reference samples respective intensities. The data was further normalized using the *lowess* method (see Dudoit et al., 2002). A portion of the data at 48 hai is presented in Table 3.3. In what follows, we will refer to this data set as the AT data set.

[1] Using the TOM1 chip available at http://ted.bti.cornell.edu.

[2] Available at http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE4116.

TABLE 3.3. Data 48 hai from *Arabidopsis thaliana* inoculated with *A. tumefaciens*.

	Inoculated (I)				Non-inoculated (NI)			
Gene	I1	I2	I3	I4	NI1	NI2	NI3	NI4
1	0.24	0.14	-0.06	0.55	0.36	0.3	0.45	0.65
2	-0.06	-0.07	-0.48	-0.43	-0.02	-0.21	-0.57	-0.28
3	0.34	0.23	0.06	0.43	-0.03	0.07	-0.2	0.18
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
26,474	0.01	-0.16	0.11	0.12	-0.03	0	0.06	0.11

CHAPTER 4

Results

In this chapter we apply the methods presented in Chapter 3 to identify differentially expressed genes in the PI and AT microarray data sets. The results are consistent with those obtained with traditional methods, but offer more useful insights regarding the behaviour of up and down regulated genes when Biological Scenario 2 from Section 2.6 seems more reasonable.

The PI data set constitutes a perfect case study in which Biological Scenario 2 holds and our method is most powerful. Its analysis was carried out following the guidelines in Chapter 3. We compare our results with those of Cai et al. (2013) and illustrate the advantages of our method when Biological Scenario 2 holds.

On the other hand, the AT data set does not exhibit a behaviour consistent with Biological Scenario 2 and, thus, the applicability of our method here is limited. We choose an active time point as in Section 3.3 and perform the single time point analysis from Chapter 3. We compare our results with those of Ditt et al. (2006) and illustrate the drawbacks of our method when Biological Scenario 2 does not hold.

As it will be seen, it is of paramount importance to determine if Biological Scenario 2 holds in order to choose the appropriate method for identifying differentially expressed genes in microarray experiments. Throughout the analysis of both data sets we highlight some hints and ad hoc rules for this assessment and summarize them in the form of heuristic guidelines towards the end of the chapter.

4.1 Tomato plants inoculated with *P. infestans*

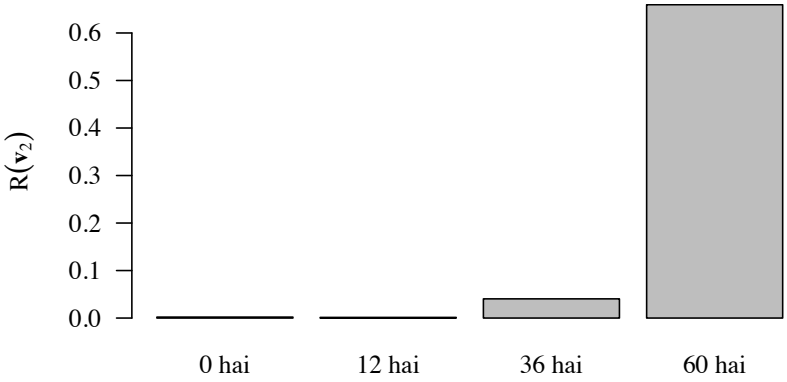
The inertia ratios for the PI data set (Section 3.4.1) are presented in Table 4.1 and Figure 4.1. Based on the ratio $R(\mathbf{v}_2)$, we see that 60 hai is the time point at which differential expression is most clear and that there are signs of differential expression taking place at 36 hai (see Restrepo et al. (2005)). Data sets at 0 and 12 hai have practically no information regarding differential expression.

The inertia ratios $R(\mathbf{v}_1)$ and $R(\mathbf{v}_1, \mathbf{v}_2)$ in Table 4.1 are very close to 1, so the artificial components are very close to the principal components of the data. More specifically, they capture 98.2%, 97.5%, 99.8% and 99.5% as much information, respectively.

TABLE 4.1. Inertia ratios for the PI data set.

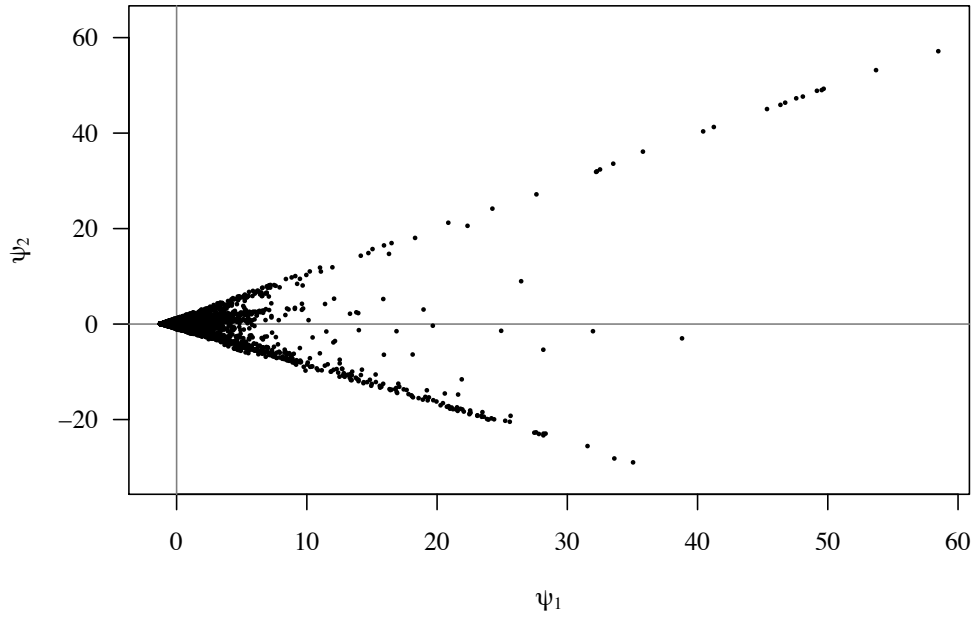
Time point	0 hai	12 hai	36 hai	60 hai
$R(\mathbf{v}_1)$	1.000	1.000	1.000	0.997
$R(\mathbf{v}_2)$	0.002	0.001	0.040	0.660
$R(\mathbf{v}_1, \mathbf{v}_2)$	0.982	0.975	0.998	0.995

FIGURE 4.1. Inertia ratios $R(\mathbf{v}_2)$ for the PI microarray data set.



The plot of the artificial components for the PI data set at 60 hai is presented in Figure 4.2. The genes' distribution in the plane is consistent with the expected behaviour when Biological Scenario 2 holds. Indeed, most genes are close to the origin and only a small proportion are far towards the right side of the plot, indicating that only a small proportion of the genes were actually expressing themselves when the samples were taken. Note, also, that there are no genes far to the left of the plane.

FIGURE 4.2. Artificial components for the PI microarray data set 60 hai.



4.1.1 Differentially expressed genes

To identify differentially expressed genes in the PI data set, we estimated the $pFDR$ and its 95% upper confidence bound according to Algorithms 3.2.1 and 3.2.2. We used $\lambda = 0.5$, $R = 1000$ and $B = 100$. The results are displayed in Figure 4.3.

Controlling the $pFDR$ at level $\alpha^* = 5\%$ gives the rejection region $[t^*, \infty)$ for $|\psi_2(\mathbf{x}_{i\cdot})|$ with $t^* = 10.49$. The BCa upper confidence bound for the $pFDR$ at t^* is 8.6%, so good control is actually achieved. With this setup, 32 up regulated and 94 down regulated genes were identified. These are presented in Figure 4.4 and a list is given in the Appendix.

FIGURE 4.3. Estimated $pFDR$ for the PI microarray data set 60 hai.

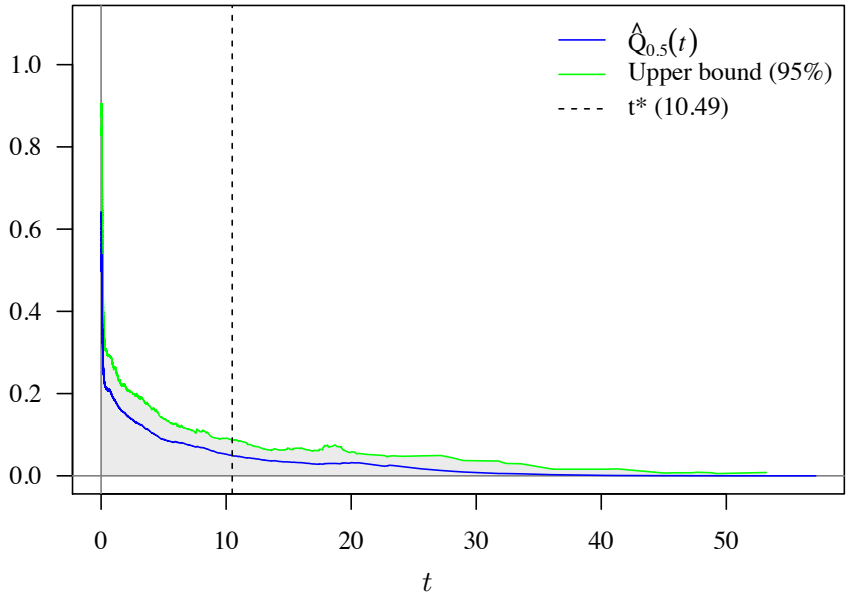
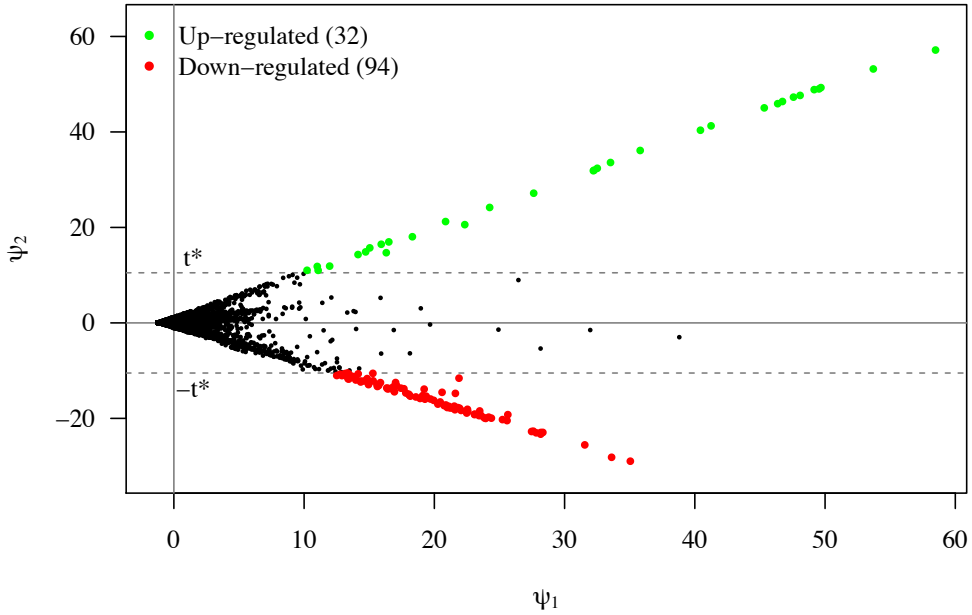


FIGURE 4.4. Differentially expressed genes in the PI microarray data set 60 hai.



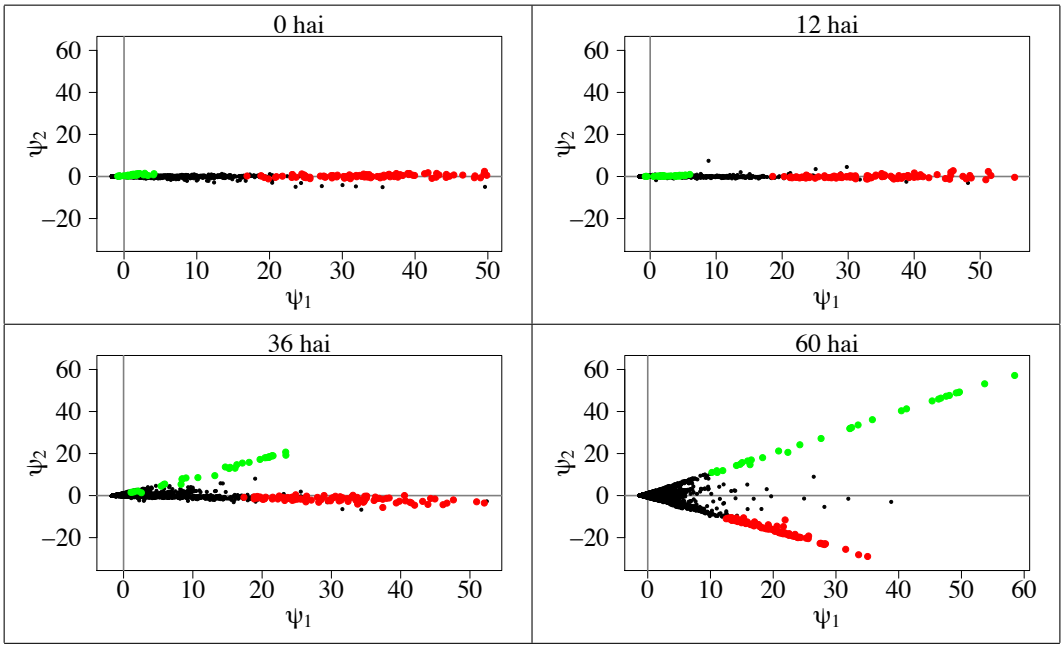
Because the identified groups of genes are small, it is fairly straight forward to locate each group's center of gravity and, so, we omit them from the plot. However, should the identified groups of genes have considerable sizes, direct visual analysis would be cumbersome, and projecting each group's center of gravity upon the artificial plane with (2.6) could be very informative as to their general behaviour.

4.1.2 Time course analysis

The behaviour throughout the time course of the up and down regulated genes identified at 60 hai is presented in Figure 4.5. Between 0 and 12 hai, the up regulated genes (in green) lie at the origin of the artificial plane so they have low or zero expression levels in all replicates. Between 12 and 36 hai, they move towards the top right corner and move even farther between 36 and 60 hai, presenting high expression levels only in the inoculated replicates. This behaviour is consistent with that of defence genes that would normally have low expression levels but become highly expressed as a reaction to the pathogen.

On the other hand, the down regulated genes (in red) lie far to the right in the artificial plane and very near to the horizontal axis between 0 and 36 hai, so they have high expression levels for both inoculated and non inoculated replicates. Between 36 and 60 hai, the down regulated genes' expression levels drop drastically only in the inoculated replicates. This behaviour is consistent with that of genes associated with primary metabolic functions that would normally have high expression levels but fail to function as a result of the inoculation with the pathogen.

FIGURE 4.5. Active vs supplementary time points analysis for the PI microarray data set.

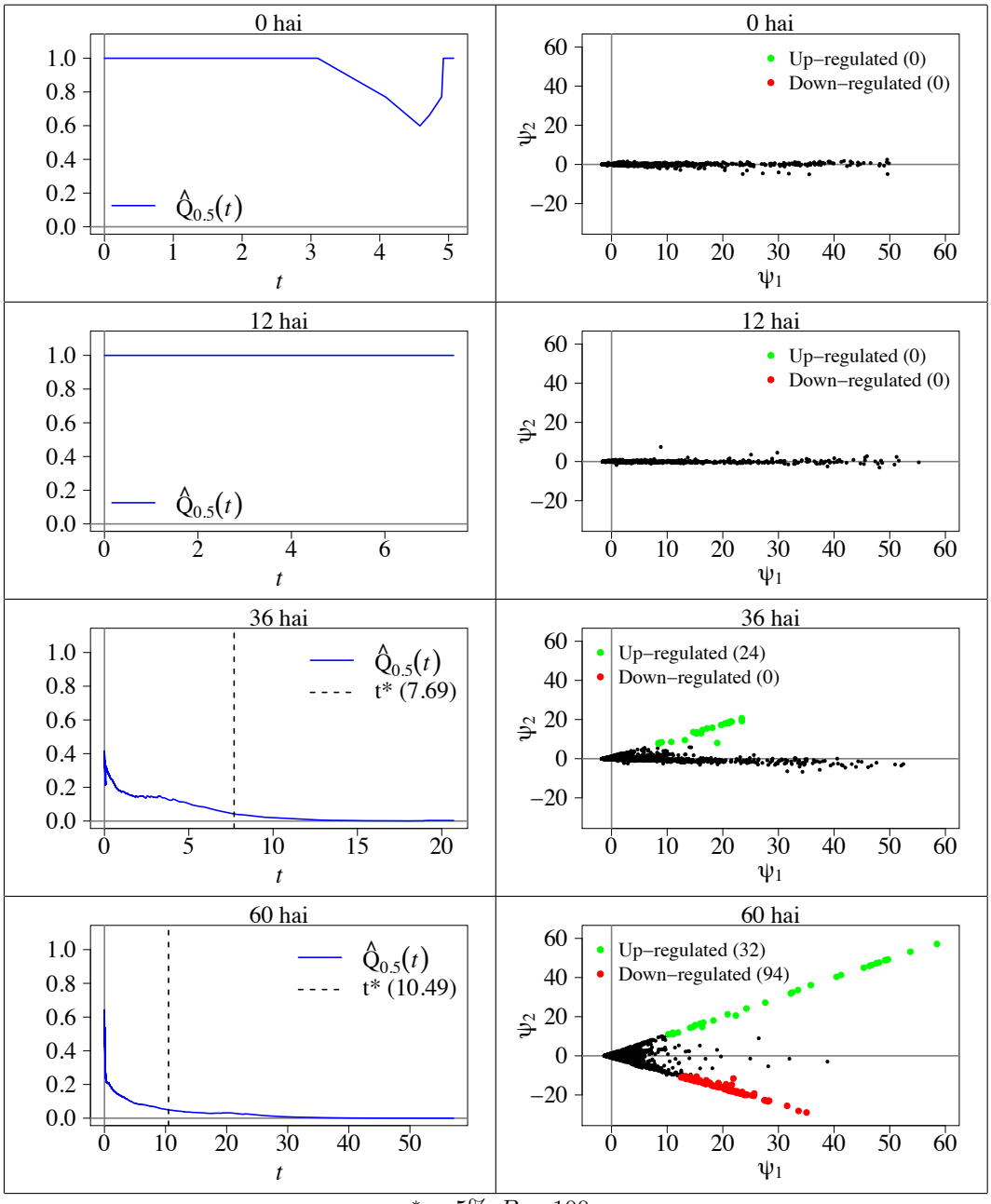


Active time point 60 hai, $\alpha^* = 5\%, B = 100$.

In Figure 4.6, we present the estimated $pFDR$ and the corresponding groups of up and down regulated genes detected when the single time point analysis is performed for each time point separately. As expected, there are no differentially expressed genes at 0 and

12 hai. At 36 hai, 24 up regulated genes and 0 down regulated genes are identified. At 60 hai, the remaining 8 up regulated and 94 down regulated genes become differentially expressed. In other words, the process of differential expression begins between 12 and 36 hai with 24 defence related genes increasing their expression levels and it attains its full dimension at 60 hai with 32 defence related genes having high expression levels and 94 primary metabolic functions related genes being shut down in the inoculated replicates.

FIGURE 4.6. Group conformation through time for the PI microarray data set.



$\alpha^* = 5\%, B = 100$.

Note that the estimated curves of the $pFDR$ at each time point are very informative regarding this timeline for the differential expression process. At 0 and 12 hai, for example, it is clear that there are no differentially expressed genes to be detected, whereas at 36

and 60 hai it is possible to attain reasonable $pFDR$ levels, which indicates the existence of differentially expressed genes.

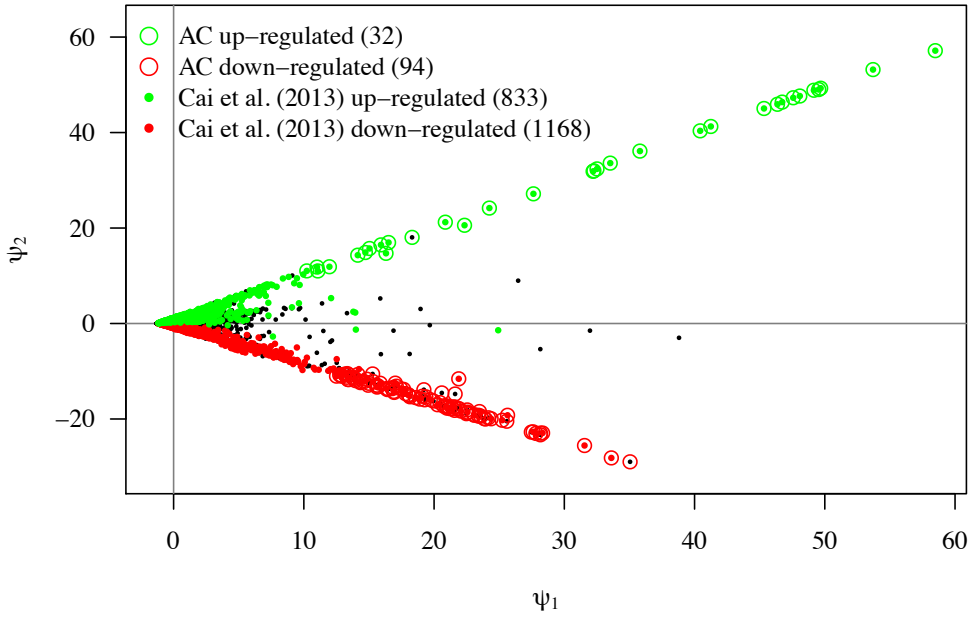
4.1.3 Comparison with other methods

Cai et al. (2013) applied SAM and a two factor (cultivar and time point) ANOVA to identify differentially expressed genes between the IL6-2 tomato plants in the PI data set and another near isogenic tomato line (M82). Although their main objectives were different from ours, it is possible to obtain the groups of up and down regulated genes only for the IL6-2 tomato line in the presence of *P. infestans* from their analysis[1]. We compare their results with our findings in Table 4.2 and Figure 4.7. We refer to our method as AC for it is based on the construction of artificial components.

TABLE 4.2. Comparison with Cai et al. (2013) for PI microarray data set 60 hai.

AC	Cai et al. (2013)			Total
	Up regulated	Down regulated	No diff. expr.	
Up regulated	30	0	2	32
Down regulated	0	41	53	94
No diff. expr.	803	1127	11384	13314
Total	833	1168	11439	13440

FIGURE 4.7. Comparison with Cai et al. (2013) for PI microarray data 60 hai.



While our method found 2 up regulated and 53 down regulated genes unidentified by Cai et al. (2013), they still identified a much larger number of genes as being differentially expressed. This is a consequence of row standardization as required by ANOVA and SAM and their corresponding univariate point of view (see Section 2.6). Indeed, when row

[1]Tables S2 and S3, and clusters 1, 2, 5–10 from table S1 in Cai et al. (2013).

standardization is performed, the inherent scale of genetic expression in the microarray is lost for further analysis.

To see this, note that a very large number of the genes identified by Cai et al. (2013) lie very close to the origin of the artificial plane in Figure 4.7, which means that their overall expression levels are close to the average overall expression level in the microarray experiment. If Biological Scenario 2 holds, this is an important mistake because genes with no expression (those near the origin) are being identified as differentially expressed. Thus, the value of a multivariate point of view and the reason why our method should be preferred when Biological Scenario 2 is more likely to hold.

Finally, note that the fact that SAM and ANOVA identify more genes as differentially expressed than our method does not imply that they have greater power as a multiple testing procedure. Instead, these discrepancies arise from differences in the biological assumptions that underlie each method (see Section 2.6) and the corresponding implied definitions of differential expression.

4.2 *Arabidopsis thaliana* inoculated with *A. tumefaciens*

The inertia ratios for the AT data set (Section 3.4.2) are presented in Table 4.3 and Figure 4.8. Based on the ratio $R(\mathbf{v}_2)$, we see that 48 hai is the time point with more information regarding differential expression. However differences in terms of inertia between the time points for the AT data set are not as large as they were for the PI data set.

Moreover, the inertia ratios are very low in all of the time points, which means that the differential expression signal in this data set is relatively week –as may have been the actual process of differential expression in the cell cultures. Yet, this could also indicate that there are stronger signals in the data not related with differential expression.

The inertia ratios $R(\mathbf{v}_1)$ in Table 4.1 are very close to 1, so, in this case, the first artificial component is very close to the first principal component of the data. Additionally, the ratios $R(\mathbf{v}_1, \mathbf{v}_2)$ are relatively close to 1, so the artificial components capture a good proportion of the information captured by the first two principal components. In particular, the artificial plane captures 96.7% of the information captured by the first two principal components at 48 hai.

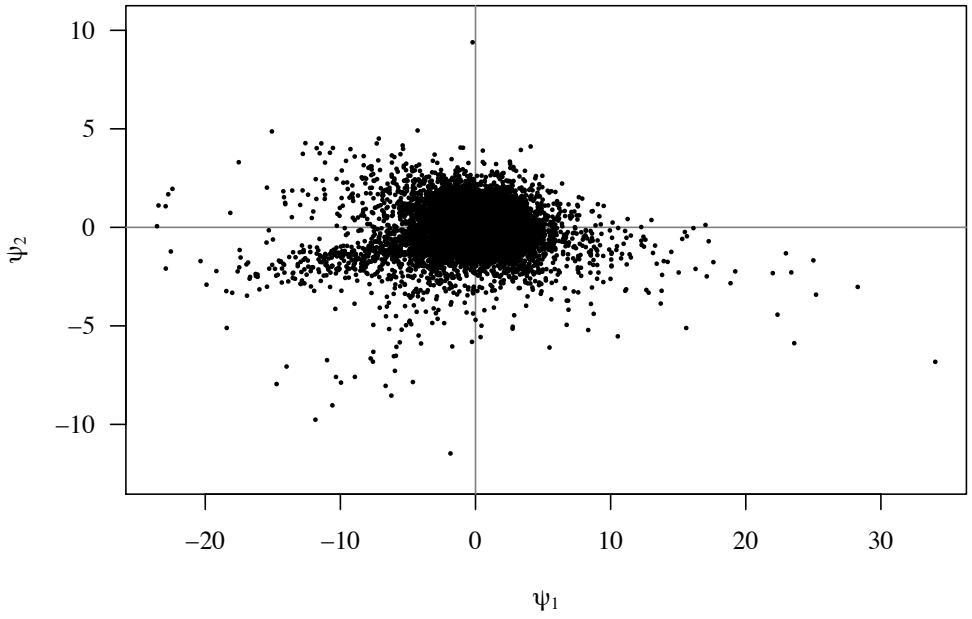
TABLE 4.3. Inertia ratios for the AT data set.

Time point	0 hai	12 hai	24 hai	48 hai
$R(\mathbf{v}_1)$	0.992	0.998	0.987	0.995
$R(\mathbf{v}_2)$	0.044	0.061	0.063	0.124
$R(\mathbf{v}_1, \mathbf{v}_2)$	0.840	0.897	0.885	0.967

The plot of the artificial components for the AT data set at 48 hai is presented in Figure 4.9. The distribution of the genes in the plane is not quite consistent with the expected behaviour when Biological Scenario 2 holds. Although most genes lie close to the origin and only a small proportion are far towards the right side of the plane, there is indeed large number of genes far away to the left of the plane. With this behaviour, Biological Scenario 2 does not seem so likely to hold and there may be systematic sources of variation other than inoculation with *A. tumefaciens* affecting the data.

FIGURE 4.8. Inertia ratios $R(\mathbf{v}_2)$ for the AT microarray data set.

FIGURE 4.9. Artificial components for the AT microarray data set 48 hai.



4.2.1 Differentially expressed genes

To identify differentially expressed genes in the AT data set, we estimated the $pFDR$ and its 95% upper confidence bound according to Algorithms 3.2.1 and 3.2.2. We used $\lambda = 0.5$, $R = 1000$ and $B = 100$. The results are displayed in Figure 4.10.

Controlling the $pFDR$ at level $\alpha^* = 5\%$ gives the rejection region $[t^*, \infty)$ for $|\psi_2(\mathbf{x}_{i\cdot})|$ with $t^* = 7.59$. The BCa upper confidence bound for the $pFDR$ at t^* is 11.87%, so reasonable control of the $pFDR$ is actually achieved. With this setup, 1 up regulated and 10 down regulated genes were identified. These are presented in Figure 4.11.

In this case, however, the position of the genes in the artificial plane is not consistent with the definition of differential expression according to Biological Scenario 2. Indeed, the 12 genes identified have less than the average overall expression level in the microarray. If Biological Scenario 2 were to hold, then these genes would be among the non–expressed ones and, thus, should not be classified as being differentially expressed. Moreover, if

FIGURE 4.10. Estimated $pFDR$ for the AT microarray data set 48 hai.

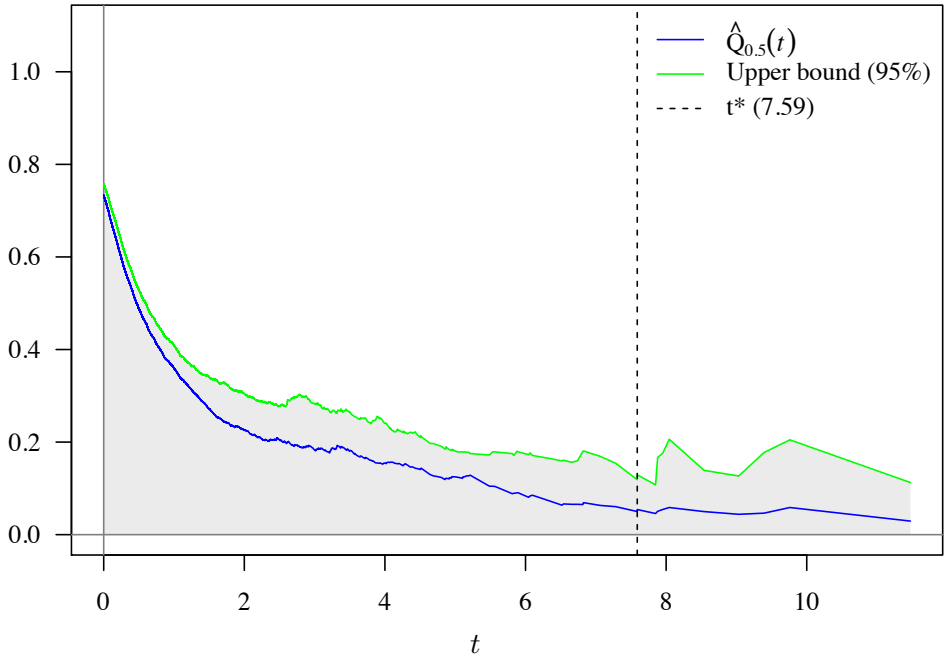
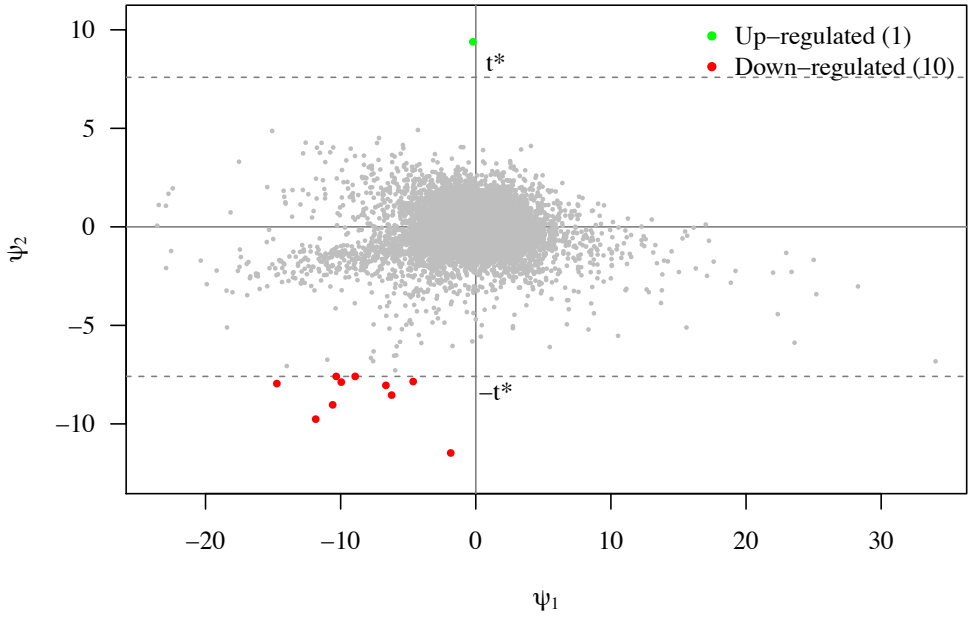


FIGURE 4.11. Differentially expressed genes in the AT microarray data set 48 hai.



Biological Scenario 2 were to hold, given the distribution of the genes in Figure 4.9, there would be no differentially expressed genes in the microarray.

Now, considering the aim of the experiment, the low inertia ratio $R(\mathbf{v}_2)$ and the distribution of genes in Figure 4.9, it is more reasonable to assume that Biological Scenario 2 does not hold in the AT microarray data set and that either most genes had true expression levels when the samples were taken, or there are strong systematic sources of

variation other than treatment/control effects. Either way, row standardization is in order and univariate oriented methods as SAM would be more convenient.

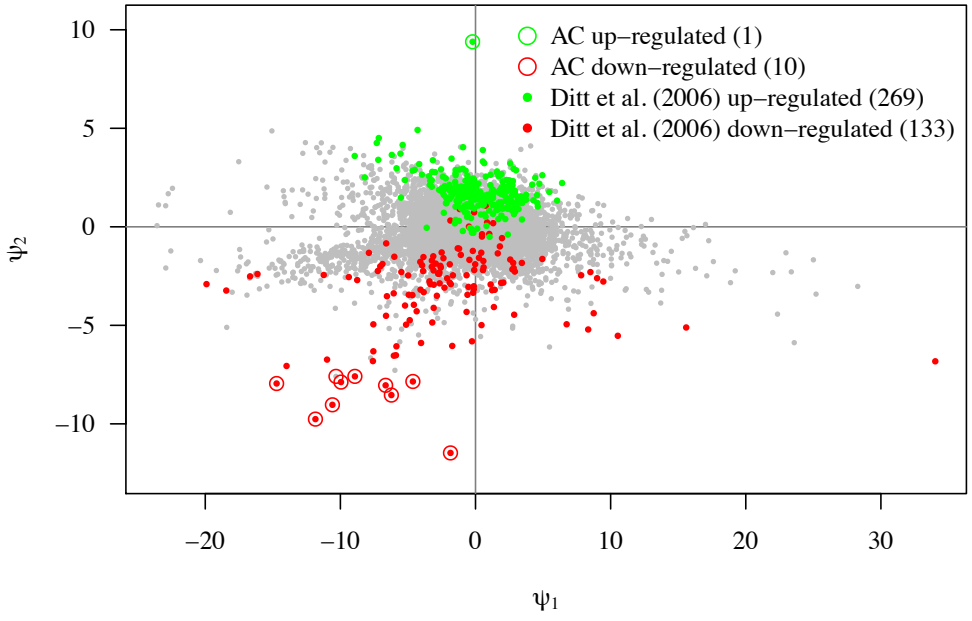
4.2.2 Comparison with other methods

Ditt et al. (2006) analyzed the AT microarray data set and found that differential expression occurs between 24 and 48 hai. Using SAM, they found 133 down regulated and 269 up regulated genes at 48 hai. We compare their results with our findings in Table 4.4 and Figure 4.12.

TABLE 4.4. Comparison with Ditt et al. (2006) for AT microarray data set 48 hai.

AC	Ditt et al. (2006)			Total
	Up regulated	Down regulated	No diff. expr.	
Up regulated	1	0	0	1
Down regulated	0	9	1	10
No diff. expr.	268	124	26071	26462
Total	269	133	26072	26474

FIGURE 4.12. Comparison with Ditt et al. (2006) for AT microarray data 48 hai.



Most differentially expressed genes identified by Ditt et al. (2006) lie close to the origin of the artificial plane and, thus, have overall expression levels close to the average in the microarray. Moreover, most down regulated genes lie to the left of the plane and, thus, have overall expression levels below the average. Naturally, in order to assert that these genes lying to the left or close to the origin are indeed differentially expressed, one must be willing to accept that Biological Scenario 1 from Section 2.6 holds.

Furthermore, the differential expression phenomenon detected in this way refers to differential expression as implicitly defined by univariate oriented methods in which overall

expression levels are not taken into account[2]. Clearly, this interpretation of differential expression is strongly related to Biological Scenario 1 and it is justified only when these biological and technical assumptions are likely to hold.

4.3 A word of caution

Throughout the analysis we have seen that, as a rule, our method identifies a much smaller number of differentially expressed genes than univariate oriented methods such as SAM. However, differences in the number of detected genes are not the result of lesser power in multiple testing, but a consequence of different understandings of the differential expression phenomenon, understandings that arise from quite opposite biological and technical assumptions about the microarray experiment under analysis.

In this regard, it is of paramount importance to be able to asses which of the biological scenarios in Section 2.6 is more likely to hold and, thus, which approach, univariate or multivariate, should be applied for a given microarray data set. Although a rigorous test is beyond the scope of this work, we propose the following heuristic guidelines based on the analysis of the PI and AT data sets to help in this endeavour:

TABLE 4.5. Heuristic guidelines for assessing Biological Scenario 2.

Biological Scenario 2 is likely to hold and our method should be preferred if:
1. $R(\mathbf{v}_2) \geq 25\%$ and $R(\mathbf{v}_1, \mathbf{v}_2) \geq 90\%$ in at least one time point.
2. Only a few genes lie far to the right from the origin in the artificial plane and no genes lie far to the left.
3. All genes detected as differentially expressed lie to the right of the vertical axis in the artificial plane.
Otherwise, Biological Scenario 2 is not likely to hold and univariate oriented methods should be preferred.

[2]This is due, partly, to row standardization (see Section 2.6).

CHAPTER 5

Conclusions and Future Perspectives

Throughout this work, we presented a multivariate approach for the identification of differentially expressed genes in microarrays. While resting on a very general probabilistic model, the applicability of our method lies upon the key and yet common biological and technical assumptions for microarray data summarized in Biological Scenario 2 from Section 2.6. If these assumptions hold, as is generally the case in microarray data, a multivariate approach is needed in order to avoid identifying non–expressed genes as being differentially expressed. So far, no multivariate inferential approach appropriate for Biological Scenario 2 had been proposed.

Our method is based on the work of Storey & Tibshirani (2001) for the estimation of the $pFDR$ and on the construction of two artificial components –close to the data's principal components, but with an exact interpretation in terms of overall and differential genetic expression– that provide useful insights regarding the extent to which Biological Scenario 2 holds and the behaviour of the differential expression process. Also, comparison of inertia ratios and estimated false discovery rates between different time points proved to be very valuable in this regard.

Additional assessments were proposed in order to gain more statistical assurance for the results obtained with our method. These were the complementary approaches for time course analysis, the computation of a BCa upper confidence bound for the $pFDR$ and the heuristic guidelines derived from real microarray data sets to assess whether Biological Scenario 2 holds. These additional assessments constitute the final piece of an integral strategy for the identification of differentially expressed genes in microarray data.

We applied our method in two real microarray data sets. Our analysis of the PI data set (the *case study*) resulted in 32 defence related genes identified as up regulated and 94 primary metabolic function related genes identified as down regulated. We found that the process of differential expression began between 12 and 36 hai with 24 defence related genes rising their expression levels only in the inoculate replicates, and attained its full dimension between 36 and 60 hai with all 126 differentially expressed genes. The genes' distribution on the artificial plane and the corresponding inertia ratios supported the assumptions in Biological Scenario 2 for this data set.

After comparison with Cai et al. (2013), a large number of the genes identified as differentially expressed by more traditional methods lied close to the origin of the artificial

plane. It then became clear that when applying methods based upon univariate statistics, the inherent genetic expression scale in the data is lost and genes with true zero expression levels may be wrongly identified as being differentially expressed. Thus, the value of a multivariate point of view and the reason why our method should be preferred when Biological Scenario 2 is more likely to hold.

Our analysis of the AT data set (the *counter example*) resulted in 1 up regulated and 11 down regulated genes at 48 hai. However, the identified genes lied in the left half of the artificial plane and, thus, their behaviour was not consistent with the assumptions in Biological Scenario 2. The genes' distribution on the plane and the small inertia ratios confirmed that Biological Scenario 2 was not likely to hold in this data set. Here, either most genes had true expression levels when the samples were taken or there were strong systematic sources of variation other than treatment/control effects. Either way, row standardization is in order and univariate oriented methods as SAM would be more convenient for this data set.

As a rule, univariate oriented methods identified much more genes as being differentially expressed. These discrepancies arise from differences in the biological assumptions that underlie each method and the corresponding implied definitions of differential expression, and, thus, are not indicative of any method's greater power as a multiple testing procedure. Moreover, when the aim of the study is to perform an intervention upon differentially expressed genes, our method may prove very valuable as it prevents it from being done upon genes with no expression whatsoever.

Finally, as our method constitutes a first multivariate inferential approach to identifying differentially expressed genes in microarray data, many questions remain open for further investigation. We end this work by presenting some future perspectives along with various hints that, we hope, will shed some light on the road ahead.

Biological and technical assumptions: Determining if Biological Scenario 2 holds is of paramount importance in order to choose the appropriate method for identifying differentially expressed genes in microarray experiments. As a part of our strategy, we proposed some heuristic guidelines to help in this respect based mainly on inertia ratios and the genes' distribution on the artificial plane. However, it still remains to design formal hypothesis tests that give statistical assurance to these heuristic procedures. Factor Analysis tests may provide a starting point in this regard.

Probabilistic assumptions and assessments: However general the probabilistic assumptions of our method, its validity still depends on the conditions of Theorem 2. Storey & Tibshirani (2001) affirm that these conditions are likely to hold in microarray data and that they may be verified in practice, but do not give any further hints. This practical verification as well as determining the asymptotic properties of $\hat{Q}_\lambda(t)$ when $|\psi_2|$ is used as the test statistic constitute important work still to be done.

Extensions: Our methodology was based on the metric and projection matrices of Principal Components Analysis and its applicability required the assumptions in Biological Scenario 2. However, a parallel may be drawn between microarray data and contingency tables; thus, the metric and projection procedures used in Correspondence Analysis may provide genetic expression components suitable for extending our method for data sets in which Biological Scenario 1 is more likely to hold. More obvious, yet challenging, extensions of our method consist in analyzing microarray data sets with more than two conditions of interest or in including of continuous covariates.

Applications to biological studies: As was previously seen, our method can be very valuable in applications where expensive interventions take place (gene silencing, etc.) for it prevents them to be performed upon non–expressed genes. Aside this natural application, our method could provide a good filter in the construction of genetic networks. Using the size factor ψ_1 as the test statistic, one could identify which genes had actual positive expression levels when the samples were taken and exclude non–expressed genes from further analyses. If a genetic network is to be constructed upon differentially expressed genes only, application of our method for identifying those genes may also be very useful.

APPENDIX A

Appendix

A.1 Differentially expressed genes in the PI data set

The down regulated genes identified in the PI data set at 60 hai are presented in Table A.1. The up regulated genes identified in the PI data set at 36 and 60 hai are presented in Tables A.2 and A.3, respectively.

TABLE A.1. Down regulated genes in the PI data set 60 hai.

Probe ID	Unigene	Description	ψ_1	ψ_2	Q–Value
1-1-8.1.14.16	No re-sequence	No description.	35.05	-28.97	0.01
1-1-1.3.14.17	SGN-U225521	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	33.61	-28.15	0.01
1-1-2.4.7.6	SGN-U225521	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	31.55	-25.55	0.02
1-1-1.3.3.6	No re-sequence	No description.	27.47	-22.74	0.02
1-1-2.2.10.8	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	27.6	-22.69	0.02
1-1-2.3.10.20	Potential chimera	No description.	27.82	-23.02	0.02
1-1-4.1.14.7	No re-sequence	No description.	28.33	-22.94	0.02
1-1-8.1.7.14	Potential chimera	No description.	28.17	-22.95	0.02
1-1-8.2.12.19	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	28.16	-23.30	0.02
1-1-1.1.14.14	SGN-U225521	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	25.64	-19.21	0.03
1-1-1.3.11.2	SGN-U219365	F-box family protein [Populus trichocarpa].	20.99	-17.72	0.03
1-1-2.4.5.12	No re-sequence	No description.	25.23	-20.25	0.03
1-1-3.1.10.21	No re-sequence	No description.	23.48	-18.44	0.03
1-1-3.2.7.9	No re-sequence	No description.	22.57	-18.56	0.03
1-1-4.2.9.2	SGN-U213815	Thylakoid membrane phosphoprotein 14 kDa, chloroplast precursor, putative [Ricinus communis].	22.05	-18.33	0.03

(continues...)

Probe ID	Unigene	Description	ψ_1	ψ_2	Q–Value
1-1-5.2.7.9	SGN-U225521	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	24.38	-19.95	0.03
1-1-5.2.12.2	SGN-U225519	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	21.53	-17.44	0.03
1-1-5.4.1.10	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	23.94	-20.01	0.03
1-1-5.4.12.4	No re-sequence	No description.	22.51	-18.7	0.03
1-1-5.4.12.14	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	21.94	-17.87	0.03
1-1-6.4.14.8	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	21.12	-17.48	0.03
1-1-6.4.14.14	No re-sequence	No description.	23.59	-19.40	0.03
1-1-7.2.7.10	SGN-U225521	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	25.58	-20.46	0.03
1-1-7.2.10.13	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	22.54	-18.10	0.03
1-1-7.2.12.15	SGN-U225527	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	23.84	-19.89	0.03
1-1-7.3.16.8	No re-sequence	No description.	20.93	-17.44	0.03
1-1-7.3.18.2	No re-sequence	No description.	23.09	-19.21	0.03
1-1-7.4.16.8	No re-sequence	No description.	21.80	-17.74	0.03
1-1-7.4.18.2	No re-sequence	No description.	21.22	-17.83	0.03
1-1-8.1.11.1	SGN-U215688	WRKY transcription factor, putative [Ricinus communis].	24.17	-19.74	0.03
1-1-8.2.14.13	SGN-U213219	NAC-domain containing protein 29 , putative [Solanum demissum].	23.43	-19.42	0.03
1-1-8.2.16.7	No re-sequence	No description.	22.48	-18.88	0.03
1-1-8.3.14.14	No re-sequence	No description.	21.45	-17.86	0.03
1-1-8.3.16.8	No re-sequence	No description.	23.41	-19.22	0.03
1-1-8.3.18.2	No re-sequence	No description.	21.58	-18.19	0.03
1-1-8.4.16.8	SGN-U213250	40S ribosomal protein S11, putative [Ricinus communis].	23.10	-19.11	0.03
1-1-5.4.10.10	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	20.75	-17.31	0.03
1-1-8.4.18.2	No re-sequence	No description.	20.88	-17.19	0.03
1-1-5.2.10.13	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	20.27	-17.01	0.03
1-1-6.4.7.9	No re-sequence	No description.	20.45	-16.56	0.03
1-1-5.4.8.4	No re-sequence	No description.	19.95	-16.22	0.03
1-1-4.1.12.13	No re-sequence	No description.	19.71	-15.95	0.03
1-1-6.4.5.7	No re-sequence	No description.	19.28	-15.98	0.03
1-1-2.2.8.2	SGN-U223767	catalytic, putative [Ricinus communis].	18.91	-15.82	0.03
1-1-7.2.8.19	No re-sequence	No description.	18.58	-15.51	0.03
1-1-5.4.7.9	SGN-U235358	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	19.37	-15.36	0.03
1-1-5.4.15.19	SGN-U219781	No description.	18.15	-15.34	0.03
1-1-7.4.7.5	No re-sequence	No description.	19.10	-15.24	0.03
1-1-7.3.14.14	No re-sequence	No description.	18.07	-15.09	0.03
1-1-2.2.12.2	No re-sequence	No description.	18.03	-14.93	0.03

(continues. . .)

Probe ID	Unigene	Description	ψ_1	ψ_2	Q–Value
1-1-4.1.4.20	SGN-U212623	Putative acyl-CoA synthetase [Capsicum annuum].	21.62	-14.76	0.03
1-1-8.4.7.13	SGN-U225508	Putative DNA/RNA binding protein [Solanum tuberosum].	17.81	-14.67	0.03
1-1-4.1.10.15	No re-sequence	No description.	20.59	-14.54	0.03
1-1-4.4.5.1	SGN-U215316	Pectin methylesterase [Nicotiana tabacum].	16.94	-14.43	0.03
1-1-6.2.2.17	SGN-U225527	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	16.90	-14.23	0.03
1-1-1.2.12.13	SGN-U225512	Phospho-2-dehydro-3-deoxyheptonate aldolase 2, chloroplastic; Phospho-2-keto-3-deoxyheptonate aldolase 2; 3-deoxy-D-arabino-heptulosonate 7-phosphate synthase 2; DAHP synthetase 2; Precursor.	19.22	-13.87	0.04
1-1-1.4.16.6	No re-sequence	No description.	17.44	-13.67	0.04
1-1-4.2.14.7	SGN-U222659	Sn-1 [Capsicum annuum].	16.35	-13.63	0.04
1-1-4.4.12.12	SGN-U212939	Ribulose bisphosphate carboxylase/oxygenase activase, chloroplastic; Precursor.	16.78	-13.64	0.04
1-1-5.2.10.18	SGN-U225512	Phospho-2-dehydro-3-deoxyheptonate aldolase 2, chloroplastic; Phospho-2-keto-3-deoxyheptonate aldolase 2; 3-deoxy-D-arabino-heptulosonate 7-phosphate synthase 2; DAHP synthetase 2; Precursor.	17.66	-13.79	0.04
1-1-5.4.7.11	No re-sequence	No description.	16.42	-13.83	0.04
1-1-6.4.12.14	No re-sequence	No description.	16.50	-13.73	0.04
1-1-6.4.12.20	No re-sequence	No description.	16.85	-13.39	0.04
1-1-3.3.12.5	SGN-U213389	Photosystem I reaction center subunit N, chloroplast precursor, putative [Ricinus communis].	15.62	-13.31	0.04
1-1-8.2.18.1	SGN-U222358	No description.	15.72	-13.12	0.04
1-1-8.4.7.14	SGN-U225512	Phospho-2-dehydro-3-deoxyheptonate aldolase 2, chloroplastic; Phospho-2-keto-3-deoxyheptonate aldolase 2; 3-deoxy-D-arabino-heptulosonate 7-phosphate synthase 2; DAHP synthetase 2; Precursor.	17.10	-13.12	0.04
1-1-5.3.4.20	SGN-U213815	Thylakoid membrane phosphoprotein 14 kDa, chloroplast precursor, putative [Ricinus communis].	15.60	-12.93	0.04
1-1-6.4.10.21	SGN-U225538	Ribulose bisphosphate carboxylase small chain 1, chloroplastic; LESS 17; Precursor.	14.94	-12.92	0.04
1-1-1.4.10.17	No re-sequence	No description.	17.02	-12.48	0.04
1-1-3.3.1.13	SGN-U230677	Oxygen evolving enhancer 3 (PsbQ) family protein [Arabidopsis thaliana].	14.99	-12.52	0.04
1-1-6.2.7.14	No re-sequence	No description.	15.86	-12.51	0.04
1-1-4.4.2.15	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	14.35	-12.38	0.04

(continues. . .)

Probe ID	Unigene	Description	ψ_1	ψ_2	Q–Value
1-1-4.3.9.6	No re-sequence	No description.	14.37	-12.18	0.04
1-1-6.1.1.20	SGN-U214397	High mobility group protein [Solanum tuberosum].	14.63	-12.16	0.04
1-1-6.2.5.20	No re-sequence	No description.	15.30	-12.14	0.04
1-1-4.1.1.10	SGN-U212939	Ribulose bisphosphate carboxylase/oxygenase activase, chloroplastic; Precursor.	13.97	-11.96	0.04
1-1-1.3.9.8	SGN-U213763	Amino acid binding protein, putative [Ricinus communis].	13.44	-11.75	0.04
1-1-6.3.14.15	SGN-U213322	40S ribosomal protein S29 [Zea mays].	14.84	-11.66	0.04
1-1-7.4.7.13	SGN-U225545	Ribulose bisphosphate carboxylase small chain 1, chloroplastic; LESS 17; Precursor.	13.38	-11.72	0.04
1-1-8.4.5.19	No re-sequence	No description.	13.97	-11.66	0.04
1-1-3.4.7.4	No re-sequence	No description.	21.89	-11.58	0.04
1-1-4.3.10.13	SGN-U225539	ADP-glucose pyrophosphorylase small subunit.	13.84	-11.46	0.04
1-1-5.4.9.1	SGN-U217904	Chitinase, class II [Solanum lycopersicum].	13.30	-11.26	0.05
1-1-7.4.3.1	No re-sequence	No description.	13.76	-11.23	0.05
1-1-1.1.9.9	SGN-U225505	Ribulose bisphosphate carboxylase small chain 1, chloroplastic; LESS 17; Precursor.	12.51	-11.02	0.05
1-1-5.3.12.20	No re-sequence	No description.	12.90	-11.04	0.05
1-1-6.3.4.21	No re-sequence	No description.	12.86	-10.82	0.05
1-1-6.4.9.6	SGN-U212885	Putative chloroplast thiazole biosynthetic protein [Nicotiana tabacum].	14.15	-10.67	0.05
1-1-7.4.2.13	SGN-U225512	Phospho-2-dehydro-3-deoxyheptonate aldolase 2, chloroplastic; Phospho-2-keto-3-deoxyheptonate aldolase 2; 3-deoxy-D-arabino-heptulosonate 7-phosphate synthase 2; DAHP synthetase 2; Precursor.	13.45	-10.74	0.05
1-1-7.4.7.4	SGN-U225521	Ribulose bisphosphate carboxylase small chain 3B, chloroplastic; Precursor.	13.18	-10.69	0.05
1-1-8.2.7.10	No re-sequence	No description.	12.75	-10.76	0.05
1-1-8.3.12.20	No re-sequence	No description.	13.13	-10.62	0.05
1-1-7.2.7.4	No re-sequence	No description.	15.28	-10.56	0.05
1-1-8.4.5.20	No re-sequence	No description.	13.33	-10.49	0.05

TABLE A.2. Up regulated genes in the PI data set 36 hai.

Probe ID	Unigene	Description	ψ_1	ψ_2	Q–Value
1-1-1.1.14.7	No re-sequence	No description.	23.43	19.21	0.00
1-1-1.3.14.10	No re-sequence	No description.	20.91	18.35	0.00
1-1-1.3.19.19	No re-sequence	No description.	21.58	18.93	0.00
1-1-1.4.19.19	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	21.42	18.91	0.00
1-1-2.3.14.10	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	20.67	18.08	0.00
1-1-2.3.16.4	No re-sequence	No description	21.06	18.51	0.00
1-1-2.3.19.19	No re-sequence	No description	23.4	20.70	0.00
1-1-2.4.16.4	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	21.47	18.95	0.00
1-1-2.4.19.19	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	20.41	18.00	0.00
1-1-6.3.4.17	SGN-U213053	Phosphoribulokinase, chloroplastic; Phosphopentokinase; Precursor.	20.99	18.16	0.00
1-1-1.4.16.4	No re-sequence	No description.	19.70	17.21	0.00
1-1-1.3.16.4	No re-sequence	No description.	18.13	15.82	0.00
1-1-4.2.9.3	SGN-U212923	Pathogenesis-related leaf protein 4; Precursor.	17.15	15.48	0.00
1-1-5.1.19.9	SGN-U218404	Transcription factor, putative [Ricinus communis].	16.35	14.70	0.00
1-1-5.2.16.3	SGN-U212923	Pathogenesis-related leaf protein 4; Precursor.	14.69	13.61	0.00
1-1-2.3.12.16	No re-sequence	No description.	15.39	13.44	0.00
1-1-6.2.5.13	No re-sequence	No description.	16.09	13.03	0.01
1-1-6.2.7.7	No re-sequence	No description.	15.24	12.88	0.01
1-1-7.3.7.10	No re-sequence	No description.	13.17	9.49	0.02
1-1-4.3.3.6	SGN-U216579	FtsH-like protein precursor [Solanum lycopersicum].	10.73	8.55	0.03
1-1-5.3.5.2	SGN-U226562	67kD chloroplastic RNA-binding protein, P67.1 [Raphanus sativus].	9.01	8.40	0.03
1-1-4.1.5.21	No re-sequence	No description.	18.99	8.05	0.04
1-1-4.2.19.9	SGN-U213628	39 kDa EF-Hand containing protein [Solanum tuberosum].	8.47	8.03	0.04
1-1-1.1.12.13	No re-sequence	No description.	8.44	7.69	0.04

TABLE A.3. Up regulated genes in the PI data set 60 hai.

Probe ID	Unigene	Description	ψ_1	ψ_2	Q–Value
1-1-1.1.14.7	No re-sequence	No description.	58.48	57.15	0.00
1-1-1.4.19.19	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	53.70	53.19	0.00
1-1-1.3.16.4	No re-sequence	No description.	49.68	49.29	0.00
1-1-1.4.16.4	No re-sequence	No description.	49.53	49.01	0.00
1-1-2.3.14.10	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	47.58	47.29	0.00
1-1-2.3.16.4	No re-sequence	No description.	49.16	48.86	0.00
1-1-2.3.19.19	No re-sequence	No description.	48.08	47.65	0.00
1-1-1.3.14.10	No re-sequence	No description.	46.36	45.92	0.00
1-1-2.4.16.4	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	46.73	46.38	0.00
1-1-2.4.19.19	SGN-U212922	Pathogenesis-related leaf protein 6; Ethylene-induced protein P1; Precursor.	45.33	45.04	0.00
1-1-1.3.19.19	No re-sequence	No description.	41.25	41.28	0.00
1-1-4.2.9.3	SGN-U212923	Pathogenesis-related leaf protein 4; Precursor.	40.43	40.36	0.00
1-1-5.2.16.3	SGN-U212923	Pathogenesis-related leaf protein 4; Precursor.	35.80	36.12	0.00
1-1-5.1.19.9	SGN-U218404	transcription factor, putative [Ricinus communis].	33.53	33.60	0.00
1-1-2.3.12.16	No re-sequence	No description.	32.51	32.39	0.01
1-1-6.2.7.7	No re-sequence	No description.	32.21	31.88	0.01
1-1-6.3.4.17	SGN-U213053	Phosphoribulokinase, chloroplastic; Phosphopentokinase; Precursor.	32.27	31.99	0.01
1-1-6.2.5.13	No re-sequence	No description.	27.63	27.17	0.01
1-1-1.1.12.13	No re-sequence	No description.	24.25	24.18	0.02
1-1-4.2.19.9	SGN-U213628	39 kDa EF-Hand containing protein [Solanum tuberosum].	20.86	21.23	0.03
1-1-4.3.3.6	SGN-U216579	FtsH-like protein precursor [Solanum lycopersicum].	18.31	18.05	0.03
1-1-7.3.7.10	No re-sequence	No description.	22.34	20.58	0.03
1-1-5.1.9.9	SGN-U213790	Acidic 26 kDa endochitinase; Precursor.	16.5	16.97	0.03
1-1-5.3.5.2	SGN-U226562	67kD chloroplastic RNA-binding protein, P67.1 [Raphanus sativus].	15.92	16.48	0.03
1-1-1.2.9.4	SGN-U213790	Acidic 26 kDa endochitinase; Precursor.	15.04	15.72	0.03
1-1-2.1.5.18	No re-sequence	No description.	14.72	14.87	0.03
1-1-2.4.14.10	SGN-U225826	Chloroplast phosphate transporter precursor [Solanum tuberosum].	16.31	14.71	0.03
1-1-2.3.17.13	Potential chimera	No description.	14.14	14.32	0.03
1-1-5.1.10.18	No re-sequence	No description.	11.96	11.9	0.04
1-1-8.3.14.8	SGN-U213790	Acidic 26 kDa endochitinase; Precursor.	11.01	11.83	0.04
1-1-4.4.7.10	No re-sequence	No description.	11.08	11.01	0.05
1-1-6.3.14.8	SGN-U213790	Acidic 26 kDa endochitinase; Precursor.	10.23	11.03	0.05

A.2 Functions in R

Estimation of the pFDR: Function that estimates the $pFDR$ and identifies differentially expressed genes in microarray data for a single time point, according to Algorithm 3.2.1.

```
# Arguments:
 # data<numeric>:  a matrix with genes in rows and replicates in columns for a
     single timepoint.  Treatment columns should go to the left:
     Tr_1, ..., Tr_{p_1}, C_1, ..., C_{p_2}.
 # design<list>:  a list with attributes tr (number of treatments) and c (number
     of controls).
 # des.pFDR<numeric>:  Number in [0,1] for the desirable pFDR.
 # B<numeric>:  Number of bootstrap samples or permutations for estimating
     pFDR.
 # replacement<logical>:  if TRUE, the null distribution is estimated via
     bootstrap.  If FALSE, it is estimated via permutations.
 # lambda<numeric>:  parameter for estimating pi_0.
 # t<numeric>:  threshold values T (optional).
 # parallel<logical>:  If TRUE, parallel computation is performed.
 # '...':  Further arguments including 'cores' for the mclapply() function from
     'parallel' package, if parallel computation is to be performed.
# Value:
 # pFDR<numeric>:  Estimated pFDR as a function of the threshold values in t.
 # groups<numeric>:  array with each gene's classification (1 for no diff.
     expr., 2 for down-regulated, 3 for up-regulated).
 # q.values<numeric>:  array with each gene's estimated q-value.
 # tstar<numeric>:  smallest threshold that produces an estimated pFDR as close
     to the desirable pFDR as possible; i.e.  t*.
 # pFDRstar<numeric>:  estimated pFDR using t*.
 # psi<numeric>:  Gene's coordinates on the artificial components (psi_1, psi_2).
 # t<numeric>:  set of thresholds for estimating pFDR(t).
 # pi0<numeric>:  estimated pi_0.
pFDR <- function(data, design, des.pFDR=0.05, B=100,
    replacement=TRUE, lambda=0.5, t=NULL, parallel=FALSE, ...){
  # Loads required libraries
  require(ade4)

  # Definitions.
  p1 <- design$tr; p2 <- design$c; p <- p1 + p2
  n <- nrow(data)

  # Columnwise standardization.
  W.std <- dudi.pca(data, scannf=FALSE, nf=p)$tab
  v1 <- rep(1/sqrt(p), p)
  v2 <- c(rep(p2, p1), rep(-p1, p2)) / sqrt(p1*p2*p)
  v <- cbind(v1, v2)

  # Computation of artificial components (psi_1, psi_2).
  psi <- as.matrix(W.std) %*% v
  psi1 <- psi[,1]; psi2 <- psi[,2]

  # Computation of psi_2 bootstrap/permutation replicates under the complete null
     distribution.
  psi2.H <- matrix(rep(0, n*B), ncol=B)
```

```
  for(b in 1:B) {
    psi2.H[,b] <- as.matrix(W.std[,sample(1:p,
       replace=replacement)]) %*% v[,2]
  }

  # Computation of Ê_H(R(t)).
  t <- abs(psi2)[order(abs(psi2), decreasing=FALSE)] # Threshold
     values T.
  R <- rep(0, n) # Realized R(t)
  R.H <- matrix(rep(0, n*B), ncol=B) # Bootstrap/permutation replicates
     of R(t).
  if(parallel){
    require(parallel)
    R.t <- function(ti, A) apply(X=A, MARGIN=2, FUN=function(a,
       par.x) sum(abs(a) >= par.x), par.x=ti)
    R.H <- matrix(unlist(mclapply(X=t, FUN=R.t, A=psi2_H, ...)),
       byrow=TRUE, nrow=n)
  }
  for(i in 1:n){
    R[i] <- sum(abs(psi2) >= t[i])
    if(!parallel) R_H[i,] <- apply(X=psi2_H, MARGIN=2,
       FUN=function(a, x) sum(abs(a) >= x), x=t[i])
  }
  E.R.H <- rowMeans(R.H) # Ê_H(R(t))

  # Computation of π̂_0
  perc <- quantile(abs(psi2.H), probs=(1-lambda))
  ind <- t <= perc
  t.lambda <- max(t[ind])
  W.lambda <- sum(abs(psi2) < t.lambda) # n − R(t_λ)
  pi0 <- max(min(W.lambda/(n*(1-lambda)), 1), 0)

  # Computation of Q̂_λ(t)
  pFDR <- rep(1,n)
  pFDR <- pi0*(E.R.H/R); pFDR <- pFDR*(pFDR<=1)+(pFDR>1)

  # Q-Values
  q <- rep("NC", n)
  if(q.val){
    o <- order(abs(psi2), decreasing=FALSE)
    q <- rep(1, n)
    q[o[1]] <- pFDR[t==abs(psi2[o[1]])]
    for(i in 2:n){
      q[o[i]] <- min(pFDR[t==abs(psi2[o[i]])], q[o[i-1]])
    }
  }

  # Groups of differentially expressed genes.
  pFDRstar <- max(des.pFDR, min(pFDR))
  groups <- rep(1, n)
  if(des.pFDR >= min(pFDR)){
    tstar <- min(abs(t[pFDR <= des.pFDR]))
    groups[psi2 <= -tstar] <- 2 # Down regulated
    groups[psi2 >=  tstar] <- 3 # Up regulated
  }
```

```
  else tstar <- t[order(pFDR)[1]]

  # Results
  list(pFDR=pFDR, groups=groups, q.values=q, tstar=tstar,
       pFDRstar=pFDRstar, psi=psi, t=t, pi0=pi0)
}
```

BCa Upper bound for the pFDR: Function that computes a BCa upper confidence bound for the $pFDR$ as in Algorithm 3.2.2.

```
# Arguments:
 # data<numeric>: a matrix with genes in rows and replicates in columns for a
     single timepoint. Treatment columns should go to the left:
     Tr_1, ..., Tr_{p_1}, C_1, ..., C_{p_2}.
 # design<list>: a list with attributes tr (number of treatments) and c (number
     of controls).
 # conf<numeric>: Desired confidence for the BCa upper bound, γ.
 # des.pFDR<numeric>: Number in [0,1] for the desirable pFDR.
 # R<numeric>: Number of bootstrap samples or permutations for estimating the
     BCa upper bound for pFDR.
 # B<numeric>: Number of bootstrap samples or permutations for estimating pFDR
     in each of the R samples.
 # replacement<logical>: argument for function pFDR().
 # lambda<numeric>: parameter for estimating π_0.
 # parallel<logical>: If TRUE, parallel computation is performed.
 # '...': Further arguments including 'cores' for the mclapply() function from
     'parallel' package, if parallel computation is to be performed.
# Value:
 # res.pFDR<numeric>: object as returned by function pFDR().
 # BCa<numeric>: BCa upper confidence bound for the pFDR.
 # z0<numeric>: Estimated z_0 for each t ∈ T.
 # a<numeric>: Estimated a for each t ∈ T.
BCa.pFDR.upper.bound <- function(data, design, conf=0.95,
    des.pFDR=0.05, R=1000, B=100, replacement=TRUE, lambda=0.5,
    parallel=FALSE, ...){
  # Computes Q̂_λ(t)
  res.pFDR <- pFDR(data, design, des.pFDR=des.pFDR, B=B,
      replacement=replacement, lambda=lambda)

  # Definitions.
  t <- res.pFDR$t # Threshold values T.
  tstar <- res.pFDR$tstar # t*.
  est.pFDR <- res.pFDR$pFDR # Q̂_λ(t).
  n <- length(t)
  p1 <- design$tr; p2 <- design$c; p <- p1 + p2

  # Bootstrap/permutation samples of X.
  boot.pFDR <- matrix(rep(0, n*R), ncol=R)
  boot.index <- matrix(rep(0, p*R), ncol=R)
  for(b in 1:R){
    boot.index[,b] <- c(sample(1:p1,
       replace=TRUE),(p1+sample(1:p2, replace=TRUE)))
    data.b <- data
    data.b <- data.b[,boot.index[,b]]
```

```
35      aux <- pFDR(data.b, design, des.pFDR=des.pFDR, B=B,
            replacement=replacement, lambda=lambda, t=t,
            parallel=parallel, ...)
36      boot.pFDR[,b] <- aux$pFDR
37    }
38
39    # Computation of z_0.
40    probs <- rep(0,n)
41    for(i in 1:n){
42      probs[i] <- sum(boot.pFDR[i,] < est.pFDR[i]) / R
43    }
44    z0 <- qnorm(probs); z0[is.infinite(z0)] <- max(is.finite(z0))
45
46    # Computation of a.
47    jack.pFDR <- matrix(rep(0,n*p), ncol=p)
48    for(j in 1:p){
49      cols <- (rowSums(boot.index==j) == 0)
50      jack.pFDR[,j] <- rowMeans(boot.pFDR[,cols])
51    }
52    jack.mean <- rowMeans(jack.pFDR)
53    jack.cent <- matrix(rep(jack.mean, p), ncol=p) - jack.pFDR
54    num <- rowSums(jack.cent^3)
55    den <- 6*(rowSums(jack.cent^2))^(3/2)
56    a <- (num / den); a[is.infinite(a)] <- max(a[is.finite(a)])
57
58    # Computation of the BCa upper confidence bound, Q_t[gamma].
59    z.conf <- rep(qnorm(conf), n)
60    pr <- z0+(z0+z.conf)/(1-a*(z0+z.conf))
61    BCa <- rep(0,n)
62    for(i in 1:n){
63      BCa[i] <- quantile(boot.pFDR[i,], probs=pnorm(pr[i]))
64    }
65    BCa <- BCa*(BCa<=1)+(BCa>1)
66
67    # Results
68    list(res.pFDR=res.pFDR, BCa=BCa, z0=z0, a=a)
69  }
```

Bibliography

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X. et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling, *Nature* **403**(6769): 503–511.

Athreya, K. & Lahiri, S. (2006). Probability Theory, *Texts and Readings in Mathematics (TRIM) Series, Hindustan Book Agency* **41**.

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

Benjamini, Y. & Yekutieli, D. (2001). The Control of the False Discovery Rate in Multiple Testing under Dependency, *Annals of Statistics* pp. 1165–1188.

Bickel, P. J. & Doksum, K. A. (2001). *Mathematical Statistics: Basic Ideas and Selected Topics. Vol I.*, 2nd edn, Prentice-Hall.

Cai, G., Restrepo, S., Myers, K., Zuluaga, P., Danies, G., Smart, C. & Fry, W. (2013). Gene profiling in partially resistant and susceptible near-isogenic tomatoes in response to late blight in the field, *Molecular Plant Pathology* **14**(2): 171–184.

DiCiccio, T. J. & Efron, B. (1996). Bootstrap Confidence Intervals, *Statistical Science* pp. 189–212.

Ditt, R. F., Kerr, K. F., de Figueiredo, P., Delrow, J., Comai, L. & Nester, E. W. (2006). The Arabidopsis thaliana transcriptome in response to Agrobacterium tumefaciens, *Molecular Plant-Microbe Interactions* **19**(6): 665–681.

Dudoit, S. & Van Der Laan, M. J. (2008). *Multiple Testing Procedures with Applications to Genomics*, Springer.

Dudoit, S., Yang, Y. H., Callow, M. J. & Speed, T. P. (2002). Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments, *Statistica Sinica* **12**(1): 111–140.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife, *The Annals of Statistics* pp. 1–26.

Efron, B. (1987). Better Bootstrap Confidence Intervals, *Journal of the American Statistical Association* **82**(397): 171–185.

Efron, B. (2004). Large-Scale Simultaneous Hypothesis Testing: The Choice of a Null Hypothesis, *Journal of the American Statistical Association* **99**(465).

Efron, B. & Tibshirani, R. (1993). *An Introduction to the Bootstrap*, Vol. 57, Chapman & Hall/CRC.

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.

Hall, P. (1992). *The Bootstrap and Edgeworth Expansion*, Springer.

Jombart, T., Devillard, S. & Balloux, F. (2010). Discriminant analysis of principal components: a new method for the analysis of genetically structured populations, *BMC Genetics* **11**(1): 94.

Kerr, M. K., Martin, M. & Churchill, G. A. (2000). Analysis of Variance for Gene Expression Microarray Data, *Journal of Computational Biology* **7**(6): 819–837.

Landgrebe, J., Wurst, W. & Welzl, G. (2002). Permutation-validated principal components analysis of microarray data, *Genome Biology* **3**(4): 1–11.

Lebart, L., Morineau, A. & Piron, M. (1995). *Statistique exploratoire multidimensionnelle*, Vol. 3, Dunod Paris.

Li, J. & Tibshirani, R. (2013). Finding consistent patterns: A nonparametric approach for identifying differential expression in RNA-Seq data, *Statistical Methods in Medical Research* **22**(5): 519–536.

Ospina, L. & López-Kleine, L. (2013). Identification of differentially expressed genes in microarray data in a principal component space, *SpringerPlus* **2**(1): 60.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*

Restrepo, S., Cai, G., Fry, W. E. & Smart, C. D. (2005). Gene expression profiling of infection of tomato by Phytophthora infestans in the field, *Phytopathology* **95**(S88).

Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C., Spellman, P., Iyer, V., Jeffrey, S. S., Van de Rijn, M., Waltham, M. et al. (2000). Systematic variation in gene expression patterns in human cancer cell lines, *Nature Genetics* **24**(3): 227–235.

Shaffer, J. P. (1995). Multiple hypothesis testing, *Annual Review of Psychology* **46**(1): 561–584.

Simon, R. M., Korn, E. L., McShane, L. M., Radmacher, M. D., Wright, G. W. & Zhao, Y. (2003). *Design and Analysis of DNA Microarray Investigations*, Springer.

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.

Storey, J. D. (2003). The positive false discovery rate: A Bayesian interpretation and the q-value, *Annals of Statistics* pp. 2013–2035.

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

Storey, J. D. & Tibshirani, R. (2001). Estimating False Discovery Rates Under Dependence, with Applications to DNA Microarrays, *Technical report, Department of Statistics, Stanford University* .

Storey, J. D. & Tibshirani, R. (2003). Statistical Significance for Genomewide Studies, *Proceedings of the National Academy of Sciences* **100**(16): 9440–9445.

Taylor, J., Tibshirani, R. & Efron, B. (2005). The "miss rate" for the analysis of gene expression data, *Biostatistics* **6**(1): 111–117.

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.

Westfall, P. H. & Young, S. S. (1993). *Resampling-Based Multiple Testing: Examples and Methods for P-value Adjustment*, Vol. 279, John Wiley & Sons.

Yuan, M. & Kendziorski, C. (2006). Hidden Markov Models for Microarray Time Course Data in Multiple Biological Conditions, *Journal of the American Statistical Association* **101**(476): 1323–1332.