Identification of Differentially Expressed Genes with Artificial Components – the acde Package

Juan Pablo Acosta
Universidad Nacional de Colombia

Liliana López-Kleine
Universidad Nacional de Colombia

Abstract

Microarrays and RNA Sequencing have become the most important tools in understanding genetic expression in biological processes. With measurments of thousands of genes' expression levels across several 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 method for detecting differentially expressed genes in gene expression data implemented in the **acde** package for R (R Core Team 2014). It estimates the FDR using artificial components close to the data's principal components, but with an exact interpretation in terms of differential genetic expression. Our method works best under the most common gene expression data structure and gives way to a new understanding of genetic differential expression. We present the main features of the **acde** package and illustrate its functionality on a publicly available data set.

Keywords: differential expression, false discovery rate, principal components analysis, multiple hypothesis tests.

Contents

1. Introduction 2
2. Materials and Methods 3
   2.1. Artificial components for differential expression 3
   2.2. A word on scale 4
   2.3. Multiple hypothesis testing 5
   2.4. Single time point analysis 7
   2.5. Time course analysis 11
   2.6. Data – *Phytophthora infestans* 12
3. An R session with **acde**: detecting differentially expressed genes 13
   3.1. Single time point analysis – function `stp` 13
   3.2. Time course analysis – function `tc` 16
   3.3. Comparison with other methods 20
4. Conclusions 21
5. Session Info 22
References 24

A. Appendix: Technical details in **acde** 24
A.1. Other functions in **acde** 24
A.2. Parallel computation 24
A.3. Construction of this vignette 25
A.4. Future perspectives 25

1. Introduction

Microarrays and RNA Sequencing have become the most important tools in understanding genetic expression in biological processes (Yuan and Kendziorski 2006). Since their development, 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, Korn, McShane, Radmacher, Wright, and Zhao 2003).

With measurements of thousands of genes' expression levels across several conditions, statistical analysis of a microarray experiment necessarily involves data mining or large scale multiple testing procedures. To the date, advances in this regard have either been multivariate but descriptive (Alizadeh, Eisen, Davis, Ma, Lossos, Rosenwald, Boldrick, Sabet, Tran, Yu *et al.* 2000; Ross, Scherf, Eisen, Perou, Rees, Spellman, Iyer, Jeffrey, Van de Rijn, Waltham *et al.* 2000; Landgrebe, Wurst, and Welzl 2002; Jombart, Devillard, and Balloux 2010), or inferential but univariate (Kerr, Martin, and Churchill 2000; Yuan and Kendziorski 2006; Benjamini and Yekutieli 2001; Dudoit, Yang, Callow, and Speed 2002; Tusher, Tibshirani, and Chu 2001; Storey and Tibshirani 2001; Taylor, Tibshirani, and Efron 2005)[1].

As a result, 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 gene expression data.

In this document 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 gene expression data. It is based mainly on the work of Storey and Tibshirani (2001, 2003) for the estimation of the FDR 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. Although it was concieved as a tool for analysing microarray data, this strategy can be applied to other gene expression data such as RNA sequencing because no assumptions on variable distributions are required (Xiong *et al.* 2014).

The remainder of this work is organized as follows. In Section 2, we introduce the construction of artificial components related to genetic expression, we give a brief overview of multiple hypothesis tests and false discovery rates, and we present our strategy for the identification of differentially expressed genes. Also, towards the end of Chapter 2, we introduce the `phytophthora` data set (Restrepo, Cai, Fry, and Smart 2005), included in the package. In Section 3, we present the main functionalities of **acde** and illustrate step by step on the `phytophthora` data set. In Section 4 we present the conclusions of our analysis. Finally, we adress some technical aspects of the package in the Appendix.

[1]Recently, Xiong, Brown, Boley, Bickel, and Huang (2014) developed a method for testing gene differential expression that represents the expression profile of a gene by a functional curve based in a Functional Principal Components (FPC) space and tests by comparing FPC coefficients between two groups of samples. However, being based on functional statistics, this method requires a very high number of replicates.

2. Materials and Methods

In this section, we present the theoretical baseis of our strategy for identifying differentially expressed genes in gene expression data. We begin by introducing the multivariate piece of the puzzle: artificial components. We follow with a brief overview of multiple hypothesis tests, define the false discovery rate (FDR) according to Benjamini and Hochberg (1995), and provide the inferential piece of the puzzle with Storey and Tibshirani (2001)'s procedure for estimating the FDR. Putting these two pieces together, we get the single time point analysis from which derives most of **acde**'s functionality.

In this section, we also discuss the biological and technical assumptions required for our method to work, we provide additional assessments both for the FDR control achieved and for the validity of these assumptions, and we extend the single time point analysis for time course experiments. We end this section presenting the `phytophthora` data set (included in **acde**) from Restrepo *et al.* (2005).

2.1. Artificial components for differential expression

Let Z represent a $n \times p$ matrix where the rows correspond to the genes and the columns to the replicates in a gene expression data set, z_{ij} representing gene i expression level in replicate j. Also, let $\mathcal{C}$ be the columns and $\mathcal{G}$ be the rows of Z, genes in $\mathcal{G}$ being treated as the individuals of the analysis. We first standardize Z with respect to its column's means and variances for obtaining a new matrix X suitable for a Principal Components Analysis (PCA) as follows:

$$x_{ij} = \frac{z_{ij} - \bar{\mathbf{z}}_j}{s.e.(\mathbf{z}_j)}, \qquad i \in \mathcal{G}, \qquad j \in \mathcal{C},$$

where $\mathbf{z}_j$ is the j-th column of Z.

Usually, in a PCA of microarray data, the first principal component will mainly explain overall gene expression and the second one will mainly explain differential expression between conditions. However, in order to perform multiple tests regarding genetic differential expression, we need new components that exactly capture the genes' overall and differential expression levels. We call these components *artificial* because they do not arise naturally from solving a maximization problem as do the principal components in a PCA. 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

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

for p_1 treatment and p_2 control replicates with $p = p_1 + p_2$. Define

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

Now, ψ_{1i} is proportional to the mean expression level of gene i across both conditions, so it captures its overall expression level. Because 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 experiment. In PCA's vocabulary (Lebart, Morineau, and Piron 1995), $\boldsymbol{\psi}_1$ is a *size component*.

On the other hand, ψ_{2i} is proportional to the difference between treatment and control mean expression levels, so it captures the amount 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 (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 also be computed as:

$$\boldsymbol{\psi}_1 = X\mathbf{v}_1, \qquad \boldsymbol{\psi}_2 = X\mathbf{v}_2, \tag{2}$$

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

Finally, interpretation of ψ_1 and ψ_2 is quite straightforward: large (small) values of ψ_1 indicate high (low) overall expression levels; large positive (large negative) values of ψ_2 indicate high (low) expression levels only in the treatment replicates. Genes near the horizontal axis in the artificial plane ($\boldsymbol{\psi}_1$ vs $\boldsymbol{\psi}_2$) are those with no differential expression, and genes near the origin have near–average overall expression levels.

2.2. A word on scale

As a rule, methods that control the FDR in microarray data (Benjamini and Yekutieli 2001; Storey and Tibshirani 2001; Tusher *et al.* 2001) use test statistics of the form

$$s(\mathbf{x}_{i\cdot}) = \frac{\bar{\mathbf{x}}_{iTr} - \bar{\mathbf{x}}_{iC}}{s.e.(\bar{\mathbf{x}}_{iTr} - \bar{\mathbf{x}}_{iC}) + c_0}, \tag{3}$$

or monotone functions of s as p-values; c_0 being a convenient constant, usually zero. However, when dividing by the standard error in (3), we lose the inherent genetic expression scale that lies within the data. Consider 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 the genes are 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 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 the genes are due to whether a gene was actively transcribing when the sample was taken.

If Biological Scenario 1 holds, there is no relevant information in the differences between the scales of the rows in the data, and row standardization is in order. If, on the contrary, Biological 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.

Also, if Biological Scenario 2 holds, the data for the genes that were not expressing themselves is merely the result of external sources of variation. Those genes, having true zero expression levels in both treatment and control replicates, cannot be classified as being differentially expressed. However, because n is very large and there might be systematic sources of variation (depending on the technology used for obtaining the data), it is very likely that a considerable number of those genes with no expression will be identified as differentially expressed when using statistics of the form (3).

We strongly believe that the first condition of Biological Scenario 2 is more reasonable in the context of gene expression data. Additionally, there are several methods for normalization of gene expression data that remove systematic sources of variation other than control/treatment effects, without performing any kind of row standardization (Dudoit *et al.* 2002; Simon *et al.* 2003). These normalization procedures can be applied beforehand to assure the validity of the technical part of Biological Scenario 2.

Summing up, row standardization should not be performed, in order to avoid identifying genes with no expression as differentially expressed; this standardization eliminates natural gene differences making all gene expression levels very similar. If a scenario in between seems more likely, Biological Scenario 2 should be favoured.

Finally, it is of paramount importance to be able to asses which of the former biological scenarios is more likely to hold for a given data set. Although a rigorous test is beyond the scope of this work, we propose the following heuristic guidelines based on the results of Acosta (2015):

Table 1: Heuristic guidelines for assessing the validity of Biological Scenario 2.

Biological Scenario 2 is likely to hold if:

1. The ratios $\text{Var}(\boldsymbol{\psi}_2)/\lambda_1 \geq 0.25$ and $[\text{Var}(\boldsymbol{\psi_1}) + \text{Var}(\boldsymbol{\psi_2})]/(\lambda_1 + \lambda_2) \geq 0.9$ in at least one time point, where λ_1 and λ_2 are the eigenvalues of the first two principal components of X (see inertia ratios, Eq. (5), in Section 2.5.1).
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.

2.3. Multiple hypothesis testing

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 $K_i : t(F_{iTr}) \neq t(F_{iC})$, F_{iTr} and F_{iC} being the cumulative probability distributions for gene i in treatment and control groups, respectively. Naturally, differences in the parameter $t(F)$ are

supposed to imply differential expression.

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

$$\delta_t(\mathbf{x}_{i\cdot}) = \begin{cases} 1, & s(\mathbf{x}_{i\cdot}) \geq t \longrightarrow \text{We reject } H_i, \\ 0, & s(\mathbf{x}_{i\cdot}) < t \longrightarrow \text{We don't reject } H_i. \end{cases}$$

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

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

where P_{H_i} refers to the probability measure in $(\Omega, \mathcal{F})$ and G_{H_i} to the cumulative distribution function of $s(\mathbf{X}_{i\cdot})$ when H_i is true. In the single hypothesis paradigm, one fixates 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. Now, 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 $v = \#\{i \in \mathcal{G} : s(\mathbf{x}_{i\cdot}) \geq t,\ H_i \text{ is true}\}$, respectively. The possible outcomes of testing $H_1, \ldots, H_n$ for fixed t are depicted in Table 2. Here, $W = n - R$ and $U = n_0 - V$.

Table 2: 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 the 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, and, hence, priority is given to control of Type I Errors. However, Type II Error reduction may then be achieved by a judicious choice of the test statistic (see Efron and Tibshirani 1994, p. 211).

False Discovery Rate

Benjamini and Hochberg (1995) defined the *false discovery rate* (FDR) as the expected proportion of falsely rejected null hypothesis in the previous setup. For this, define the random variable $Q = V/R$ if $R > 0$ and $Q = 0$ otherwise. Then, the FDR is defined as

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

where the expectation is taken under the true distribution P instead of the complete null distribution P_H, where H : all null hypothesis are true. Two important properties arise from this definition (Benjamini and Hochberg 1995):

1. If H is true, then FDR equals the family wise error rate ($FWER$). 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 the $FWER$ amounts to controlling the FDR, the opposite is not true. Then, controlling only the FDR results in less strict procedures and increased power (Benjamini and Hochberg 1995).

2.4. Single time point analysis

Our method for identifying differentially expressed genes for a single time point, embeded in the `stp` function of **acde**, consists of a specific application of Storey and Tibshirani (2001)'s methodology, using a statistic that is well suited for gene expression data when Biological Scenario 2 holds. It is based on the large sample estimator for the FDR, $\hat{Q}_\lambda(t)$, presented in Storey and Tibshirani (2001) for multiple testing procedures, but uses $|\psi_2(\mathbf{X}_{i\cdot})|$ as the test statistic. Our method is presented in Algorithm 1.

Now, there are two distinct approaches in multiple hypothesis testing for controlling the FDR: one fixating a desired FDR level and estimating a rejection region, the other fixating a rejection region and estimating its FDR. Storey, Taylor, and Siegmund (2004) showed that these two approaches are asymptotically equivalent. More 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 FDR at level α, for n large enough (Storey *et al.* 2004). Consequently, the procedure in Algorithm 1 controls the FDR at level α^*.

Finally, the following observations about Algorithm 1 are in order:

1. $\hat{Q}_\lambda(t)$ is conservatively biased, i.e., $E[\hat{Q}_\lambda(t)] \geq FDR$ (Storey and Tibshirani 2001).

2. We only estimate the FDR 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 $[t_{[k]}, \infty)$ as the rejection region, produces k genes identified as differentially expressed, for $k = 1, \ldots, n$.

3. For computational ease, we set $\lambda = 0.5$, following Storey *et al.* (2004), Taylor *et al.* (2005) and Li and 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 and Tibshirani (2001, Section 6).

Algorithm 1: Identification of differentially expressed genes for a single time point.

1. Compute $\psi_{2i} = \psi_2(\mathbf{x}_{i\cdot})$ for $i = 1, \ldots, n$ from (1).
2. For each t in $\mathcal{T} = \{|\psi_{21}|, \ldots, |\psi_{2n}|\}$ and B large enough, compute $\hat{Q}_\lambda(t)$ as in Storey and Tibshirani (2001, p. 6) with $\lambda = 0.5$ and using the test statistic $s(\cdot) = |\psi_2(\cdot)|$ from (1). That is:
   2.1. Compute B bootstrap or permutation replications of $s(\mathbf{x}_{1\cdot}), \ldots, s(\mathbf{x}_{n\cdot})$, obtaining $s^*_{1b}, \ldots, s^*_{nb}$ for $b = 1, \ldots, B$.
   2.2. Compute $\hat{E}_H(R(t))$ as
   $$\hat{E}_H(R(t)) = \frac{1}{B}\sum_{b=1}^{B} r^*_b(t),$$
   where $r^*_b(t) = \#\{s^*_{ib} \geq t : i = 1, \ldots, n\}$.
   2.3. 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 $\pi_0 = n_0/n$ by
   $$\hat{\pi}_0(\lambda) = \frac{w(t_\lambda)}{n(1-\lambda)},$$
   where $w(t_\lambda) = \#\{s(\mathbf{x}_{i\cdot}) < t_\lambda : i = 1, \ldots, n\}$.
   2.4. Estimate $FDR_\lambda(t)$ as
   $$\hat{Q}_\lambda(t) = \frac{\hat{\pi}_0 \hat{E}_H(R(t))}{r(t)},$$
   where $r(t) = \#\{s(\mathbf{x}_{i\cdot}) \geq t : i = 1, \ldots, n\}$.
3. Set a desired FDR level α and compute $t^* = \inf\left\{t \in \mathcal{T} : \hat{Q}_{0.5}(t) \leq \alpha\right\}$.
4. Idenfity 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:
   $$\mathcal{D}_{t^*} = \{i : \psi_2(\mathbf{x}_{i\cdot}) \leq -t^*\}, \qquad \mathcal{U}_{t^*} = \{i : \psi_2(\mathbf{x}_{i\cdot}) \geq t^*\},$$
   respectively.
5. Estimate each gene's q-value as in Algorithm 3 (see Section 2.4.2).

4. 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 (Dudoit and Van Der Laan 2008). Though permutation methods are more popular (Li and Tibshirani 2013), we favor bootstrap estimates of the null distribution for ease of interpretation (Dudoit and Van Der Laan 2008, p. 65). In any case, for large B, the results should be very similar (Efron and Tibshirani 1994).

5. $B = 100$ should be enough for obtaining accurate and stable estimates of the FDR in step 2 (Efron and Tibshirani 1994). 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.

Further assessments

As B, n and p grow, $\hat{Q}_\lambda$ approaches from above both the FDR and the actual or realized proportion of false positives among all rejected null hypothesis (Storey and Tibshirani 2001). In practice, however, because B, n and p are finite, the control achieved using $\hat{Q}_\lambda$ is only approximate and, so, additional assessments are needed.

Storey and Tibshirani (2001) suggested the use of a bootstrap percentile confidence upper bound for the FDR 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 *bias-corrected and acceleerated* (BCa) upper confidence bound (Efron and Tibshirani 1994) for the FDR as shown in Algorithm 2. We find plots of $\hat{Q}_\lambda(t)$ and the FDR's upper confidence bound vs. t to be very informative as to the actual FDR control achieved.

Now, technically, $Q_t[\gamma]$ is a BCa upper γ confidence bound for $E[\hat{Q}_\lambda(t)] \geq FDR(t)$. Because $\hat{Q}_\lambda(t)$ is conservatively biased, $Q_t[\gamma]$ is a γ^* confidence upper bound for $FDR(t)$ with $\gamma^* \geq \gamma$, and we say that $Q_t[\gamma]$ is a *conservative* γ confidence upper bound for $FDR(t)$. Moreover, if n is large, $FDR \approx v/\max\{1, r\}$, so $Q_t[\gamma]$ is also a conservative γ confidence upper bound for the actual proportion of false positives (Storey and Tibshirani 2001).

Still, $Q_t[\gamma]$ is only second order accurate (Efron and Tibshirani 1994), that is: $P(E[\hat{Q}_\lambda(t)] \leq Q_t[\gamma]) = \gamma + O(p^{-1})$, $Q_t[\gamma]$ being the random variable and p the number of replicates in the experiment. As p is usually small in gene expression 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 $Q_t[\gamma]$, all improve.

Finally, the following observations about Algorithm 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 computed in step 1.

2. In step 2, $X^*_r \sim \tilde{F}$, where $\tilde{F}$ is the empirical distribution of X.

3. The number of computations in Algorithm 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 and Tibshirani 1994), so we recommend setting $B = 100$ and $R = 1000$. As n is usually very large, Algorithm 2 may take require considerable computational effort.

Algorithm 2: Computation of a BCa upper confidence bound for the FDR.

1. Compute $\hat{Q}_\lambda$ by applying steps 1 and 2 of Algorithm 1.
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 a random sample with replacement from $\{1, \ldots, p_1\}$ and $(k_1, \ldots, k_{p_2})$ being a random sample with replacement from $\{p_1 + 1, \ldots, p\}$.
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 1 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 $z_0(t)$ as

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

   4.2 Compute $\hat{a}(t)$ 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 FDR as (Efron and Tibshirani 1994):

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

   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.

The q-value

For an observed statistic $s(\mathbf{x}_{i\cdot})$, the q-value is defined as the minimum FDR that can ocurr when rejecting all hypothesis for which $s(\mathbf{x}_{i'\cdot}) \geq s(\mathbf{x}_{i\cdot}), i' = 1, \ldots, n$ (Storey 2002). More specifically:

$$\text{q-value}(s(\mathbf{x}_{i\cdot})) = \inf_t\{FDR(t) : s(\mathbf{x}_{i\cdot}) \geq t\}. \tag{4}$$

Storey (2003) showed that the q-value can be interpreted as the posterior probability of making a Type I Error when testing n hypothesis with rejection region $[s(\mathbf{x}_{i\cdot}), \infty)$; and so it is the analogue of the p-value when controlling the FDR in multiple hypothesis tests.

As $\hat{Q}_\lambda(t)$ is neither smooth nor necessarily decreasing in t, we estimate the q-values following Algorithm 3 adapted from Storey (2002).

Algorithm 3: 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. Time course analysis

It is often the case in gene expression data to have samples taken at different time points for analysing the genetical behaviour of the replicates in different stages of the disease or factor of interest. We propose two complementary extensions to the single time point analysis for time course gene expression data. These two approaches are: *active vs. supplementary time points* and *groups conformation through time*. For the rest of this section, suppose we have L data sets $X^{(1)}, \ldots, X^{(L)}$ taken at time points $1, \ldots, L$.

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, so a more quantitative assessment is needed. Presently, we can use inertia ratios as follows.

The amount of information about differential expression at time point l can be measured by the inertia[2] projected on the differential expression component, $\text{Var}(\boldsymbol{\psi}_2^{(l)})$, where $\boldsymbol{\psi}_2^{(l)}$ is the result of applying (2) to $X^{(l)}$, $l = 1, \ldots, L$. For it to be comparable between time points, we divide it by the maximum inertia that can be captured by a single direction as in PCA, obtaining the inertia ratios:

$$IR^{(l)} = \frac{\text{Var}[\psi_2^{(l)}]}{\lambda_1^{(l)}}, \qquad l = 1, \ldots, L, \tag{5}$$

where $\lambda_1^{(l)}$ 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 $IR^{(l)}$. We call l^* the *active* time point in the analysis.

Once l^* has been determined, Algorithms 1 and 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 $\boldsymbol{\psi_1}^{(l)}$

[2] According to PCA terminology (Lebart *et al.* 1995).

vs. $\boldsymbol{\psi_2}^{(l)}$ can be made for each time point, coloring the genes in each group to see how the differential expression process evolves through time.

Groups conformation through time

The second approach supposes that there may be different genes with differential expression at different time points. The analysis here consists simply of applying Algorithms 1 and 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. Here, $\boldsymbol{\psi_1}^{(l)}$ vs. $\boldsymbol{\psi_2}^{(l)}$ plots are also very useful.

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 as the final output 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 approach can be used to get an additional idea of the intensity of the differential expression process at each time point via inertia ratios.

2.6. Data – *Phytophthora infestans*

The **acde** package comes with the `phytophthora` data set, taken from the Tomato Expression Database website (`http://ted.bti.cornell.edu/`), experiment E022 (Restrepo *et al.* 2005)[3]. This data set contains raw measurements of 13440 tomato genes for eight plants inoculated with *Phytophthora infestans* and eight plants mock-inoculated with sterile water at four different time points (0, 12, 36 and 60 hours after inoculation).

This data set is a list with four matrices of 13440×16, one for each of the time points in the experiment. A portion of data at 60 hai (`phytophthora[[4]]`) is presented in Table 3.

Table 3: 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
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
13440	64	49	152	$\cdots$	38	58	63	81	$\cdots$	39

[3]RNA was extracted from each sample and then hybridized on a cDNA microarray, using the TOM1 chip available at `http://ted.bti.cornell.edu`. For more details of the experimental design and conditions of the study, see Cai, Restrepo, Myers, Zuluaga, Danies, Smart, and Fry (2013).

3. An R session with acde: detecting differentially expressed genes

The **acde** package is designed to identify differentially expressed genes between two conditions (treatment vs control, inoculated vs non- inoculated, etc.). All its functionality resides within two functions, `stp` and `tc`, that perform single time point analysis and time course analysis, respectively. For a correct usage, some care is needed in the organization of the data-related arguments for both functions.

Here is the code that will be used in the remainder of this work. Before running it, please consider that it may require a considerable time (see Appendix A.3). For a quick run, set argument `BCa` in function `stp` to `FALSE` in order to skip lengthy BCa computations (note that, in this case, Figure 1 won't have the BCa confidence upper bound).

```
> # Analysis of the phytophthora data set with acde
> library(acde)

> ## Single time point analysis (36 hai)
> dat <- phytophthora[[3]]
> des <- c(rep(1,8), rep(2,8))
> ### For a quick run, set BCa=FALSE
> stpPI <- stp(dat, des, BCa=TRUE)

> stpPI
> plot(stpPI)

> ## Time course analysis
> desPI <- vector("list",4)
> for(tp in 1:4) desPI[[tp]] <- c(rep(1,8), rep(2,8))
> tcPI <- tc(phytophthora, desPI)

> summary(tcPI)
> tcPI
> plot(tcPI)
```

In the next sections, we present the usage of the **acde** functionality step by step, explaining the results returned by the various functions in the previous code and interpreting them in terms of genetic differential expression. We first describe the use of the `stp` function and present a simple example with the `phytophthora` data set at 36 hai. Then, we describe and apply the `tc` function to the whole time course of the `phytophthora` data set. For further details on the internal workings of the package, please refer to the Appendix.

3.1. Single time point analysis – function `stp`

The function `stp` performs the single time point analysis presented in Section 2.4. It's main arguments are `Z` and `design`. These arguments represent the gene expression data to be analysed. The first argument, `Z`, is a matrix of $n \times p$ that represents gene expression levels with n genes in the rows and p replicates in the columns, p_1 of which correspond to the treatment or the condition of interest (inoculation, presence of disease, etc.) and the rest being control

replicates (mock-inoculated, healthy tissue, etc.). The second argument, `design`, is a vector of length p with p_1 ones indicating the position of the treatment columns in `Z`, and twos otherwise.

For performing a single time point analysis of the `phytophthora` data set at 36 hai, we use the following code. Note that the first eight columns of `phytophthora[[.]]` correspond to treatment replicates, and the remaining eight to control replicates – which gives the construction of argument `des` in this example.

```
> library(acde)
> dat <- phytophthora[[3]]
> des <- c(rep(1,8), rep(2,8))
> stpPI <- stp(dat, des, BCa=TRUE)
```

In the single time point analysis, the artificial components of the genes, ψ_1 and ψ_2, are computed via the `ac` function (same arguments `Z` and `design`). Arguments `alpha`, `lambda` and `B` correspond to the respective parameters in Algorithm 1. The argument `PER` specifies whether permutation (`PER=TRUE`) or bootstrap (`PER=FALSE`, the default) replications are to be used in step 2.1 in Algorithm 2.4 (see observation 4 in page 8). Argument `th` refers to the set $\mathcal{T}$ of threshold values to be evaluated (the default is as specified in Algorithm 1).

Arguments `BCa`, `gamma` and `R` refer to the parameters of Algorithm 2 for the additional assessments in the single time point analysis. If `BCa=TRUE`, a BCa `gamma`–confidence upper bound for the FDR is computed. Due to the computational effort required, the default is `BCa=FALSE`.

Function `stp` returns an object of (S3) class '`STP`', which is a list with various components storing the results of the single time point analysis. Of special interest are `$dgenes` (classification of the genes), `$astar` (actual FDR control achieved $\hat{Q}_\lambda(t^*)$), `$tstar` (t^*) and `$qvalues`.

Package **acde** comes with the respective `print` and `plot` methods for class '`STP`'. For printing the basic results of the previous example, just type `stpPI`. In this case, 18 genes were identified as up regulated, achieving an FDR of 1.4%. Note that the BCa 95% confidence upper bound for the FDR is 3.7%, so good control is actually achieved[4].

```
> stpPI

Single time point analysis for detecting differentially
expressed genes in microarray data.

Achieved FDR: 1.4%.
95% BCa upper bound for the FDR: 3.7%.
Warnings in BCa computation: 7398.

Inertia ratio: 4.04%.
tstar: 12.88, pi0: 1, B: 100.
```

[4]The item `Warnings in BCa computation` in the printed results refers to the number of values in `th` for which the function `boot.ci` from package **boot** returned a warning during BCa computations. These can be seen by typing `stpPI$bca$warnings` or `plot(stpPI, WARNINGS=TRUE)`.

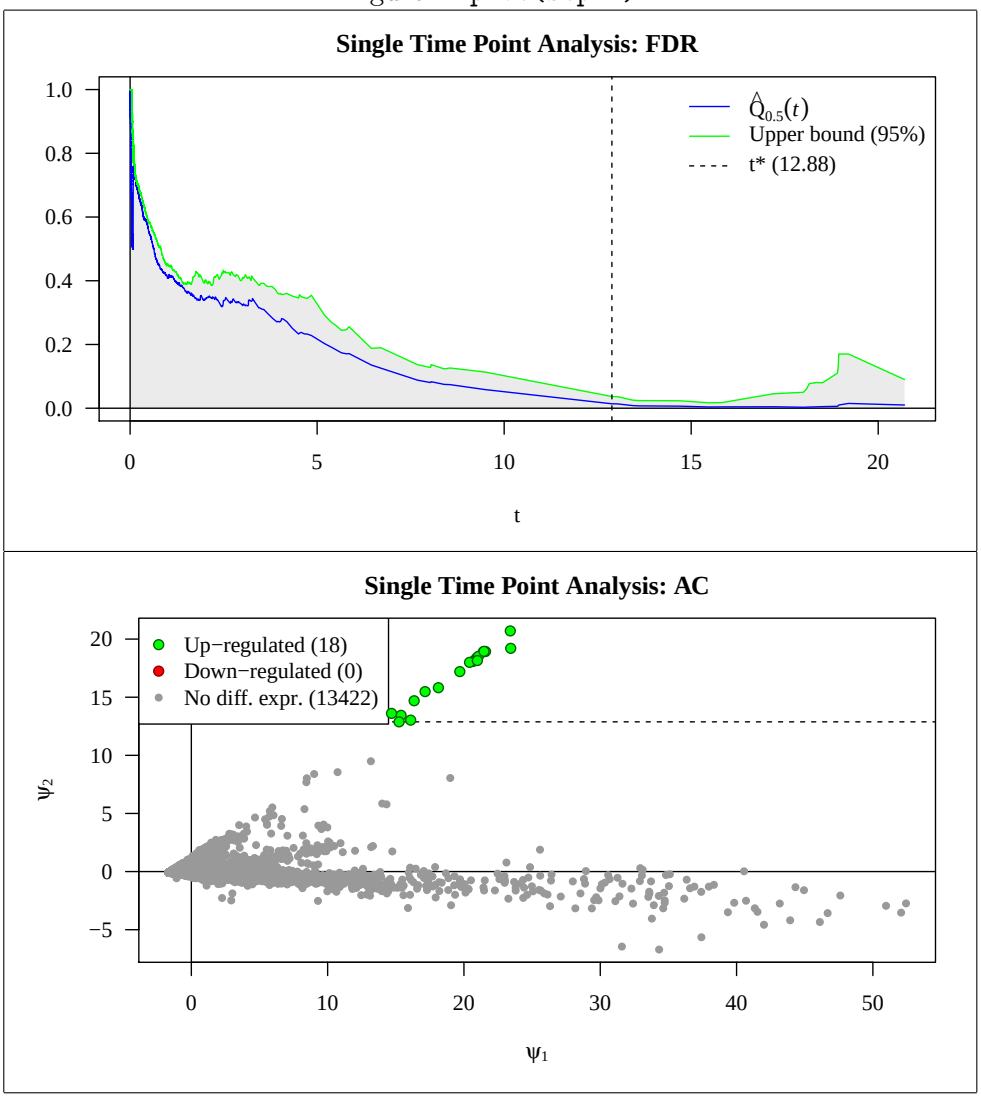
```
Differentially expressed genes:
no-diff.  up-reg.
   13422       18

Results:
                psi1   psi2 Q-value Diff. expr.
1-1-1.1.14.7  23.430 19.206   0.003     up-reg.
1-1-1.3.14.10 20.913 18.346   0.003     up-reg.
1-1-1.3.19.19 21.580 18.931   0.003     up-reg.
1-1-1.4.19.19 21.423 18.910   0.003     up-reg.
1-1-2.3.14.10 20.670 18.081   0.003     up-reg.
1-1-2.3.16.4  21.055 18.514   0.003     up-reg.
1-1-2.3.19.19 23.401 20.704   0.003     up-reg.
1-1-2.4.16.4  21.471 18.947   0.003     up-reg.
1-1-2.4.19.19 20.409 17.999   0.003     up-reg.
1-1-6.3.4.17  20.986 18.157   0.003     up-reg.
1-1-1.3.16.4  18.127 15.819   0.004     up-reg.
1-1-1.4.16.4  19.696 17.206   0.004     up-reg.
1-1-4.2.9.3   17.150 15.481   0.004     up-reg.
1-1-5.1.19.9  16.347 14.696   0.006     up-reg.
1-1-5.2.16.3  14.686 13.610   0.007     up-reg.
1-1-2.3.12.16 15.386 13.444   0.008     up-reg.
1-1-6.2.5.13  16.092 13.030   0.014     up-reg.
1-1-6.2.7.7   15.242 12.880   0.014     up-reg.
1-1-7.3.7.10  13.171  9.492   0.058    no-diff.
1-1-4.3.3.6   10.728  8.554   0.074    no-diff.
1-1-5.3.5.2    9.011  8.396   0.075    no-diff.
1-1-4.1.5.21  18.994  8.048   0.080    no-diff.
1-1-4.2.19.9   8.472  8.030   0.080    no-diff.
1-1-1.1.12.13  8.437  7.688   0.088    no-diff.
1-1-2.3.1.13  34.317 -6.708   0.126    no-diff.
1-1-6.2.20.15 31.587 -6.451   0.135    no-diff.
1-1-5.2.9.1   14.002  5.852   0.171    no-diff.
1-1-7.4.9.9   14.318  5.787   0.171    no-diff.
...

*More results are available in the objects:
$ac, $qvalues and $dgenes.
```

The instruction `plot(stpPI)` produces the two plots shown in Figure 1. The top plot shows the estimated FDR as a function of multiple threshold values (with the corresponding BCa confidence upper bounds when argument `BCa=TRUE`). The bottom plot shows the differentially expressed genes on the artificial plane ($\boldsymbol{\psi}_1$ vs $\boldsymbol{\psi}_2$).

Here, the genes' distribution in the plane is consistent with the expected behaviour when Biological Scenario 2 holds (see Table 1). 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 1: `plot(stpPI)`



3.2. Time course analysis – function `tc`

The function `tc` performs the time course analysis presented in Section 2.5. The two principal arguments for the `tc` function are `data` and `designs`. These are lists of the same length with the corresponding `Z` or `design` arguments for function `stp` at each time point. In other words, `data` and `designs` are lists such that typing `stp(data[[tp]], designs[[tp]])` performs the single time point analysis for time point `tp`. Also, note that time point names are taken from `names(data)`. Arguments `alpha`, `lambda`, `B`, `PER`, `BCa`, `gamma` and `R` are the same as in the `stp` function.

The `method` argument specifies which of the *active vs supplementary time points* and *groups conformation through time* approaches are to be performed (the default is both). If the former

approach is performed, the default is to select the active time point via inertia ratios as in Section 2.5. However, the user may set the active time point via the `activeTP` argument.

To perform a time course analysis for the `phytophthora` data set, use the following code.

```
> desPI <- vector("list",4)
> for(tp in 1:4) desPI[[tp]] <- c(rep(1,8), rep(2,8))
> tcPI <- tc(phytophthora, desPI)
```

The `tc` function returns an object of (S3) class '`TC`', which is a list containing the results from the selected methods in the `method` argument. Object `$act` contains the results from the *active vs supplementary time points* approach; Object `$gct` contains the results from the *groups conformation through time* approach.

Package **acde** comes with the respective `print`, `plot` and `summary` methods for class '`TC`'. The `summary` method is the most concise way of printing a time course analysis' results, as:

```
> summary(tcPI)

Time course analysis for detecting differentially
expressed genes in microarray data.

Inertia ratios (%):
  h0  h12  h36   h60
 0.2 0.13 4.04 65.95


Active vs complementary time points analysis:

Active timepoint: h60

Achieved FDR: 5 %.

Differentially expressed genes:
down-reg.  no-diff.   up-reg.
       94     13314        32


Groups conformation through time analysis:

Differentially expressed genes:
 h36 vs h60
           down-reg. no-diff. up-reg.   Sum
  no-diff.        94    13314      14 13422
  up-reg.          0        0      18    18
  Sum             94    13314      32 13440
```

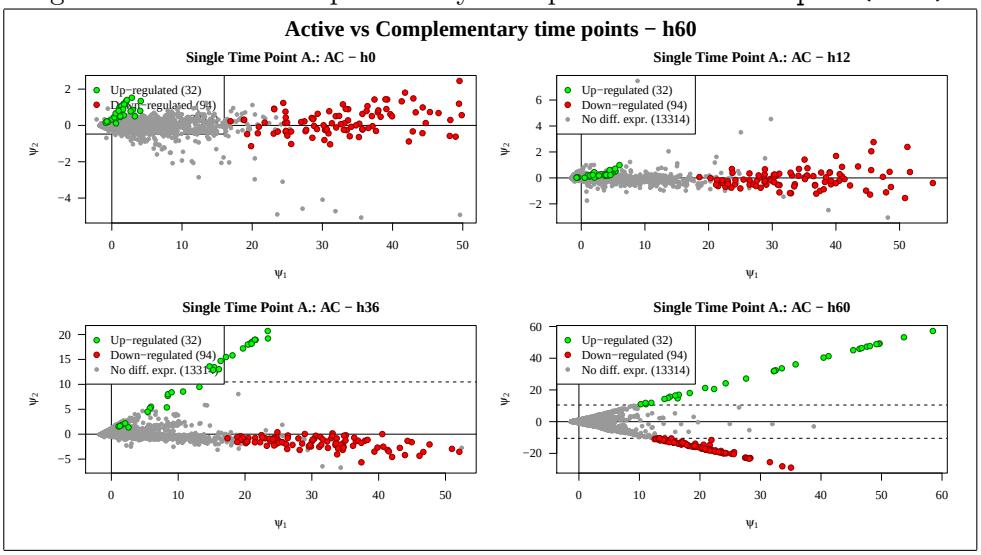
Plots of the results from both approaches (via the `plot(tcPI)` command) are displayed in Figures 2, 3 and 4. We now analyse the results of the time course analysis for the `phytohpthora` data set.

Active vs complementary time points

The results of this approach on the `phytophthora` data set are presented in Figure 2. The interpretation is quite straightforward: between 0 and 12 hai, the up regulated genes (green) lie at the origin of the artificial plane so they have low or zero expression levels in all replicates[5]. 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 (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 2: Active vs complementary time points results from `plot(tcPI)`.



Groups conformation through time

In Figures 3 and 4, we present the estimated FDR 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, 18 up regulated genes and 0 down regulated genes are identified. At 60 hai, the remaining 14 up regulated and 94 down regulated genes become differentially expressed.

[5] Assuming that Biological Scenario 2 holds.

Note that the estimated functions FDR(t) at each time point (Figure 3) are very informative regarding this timeline for the differential expression process (as are the inertia ratios at different timepoints in the output from `summary(tcPI)` above). 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 FDR levels, which indicates the presence of differentially expressed genes.

Figure 3: Groups conformation through time results from `plot(tcPI)`. Estimated FDRs.

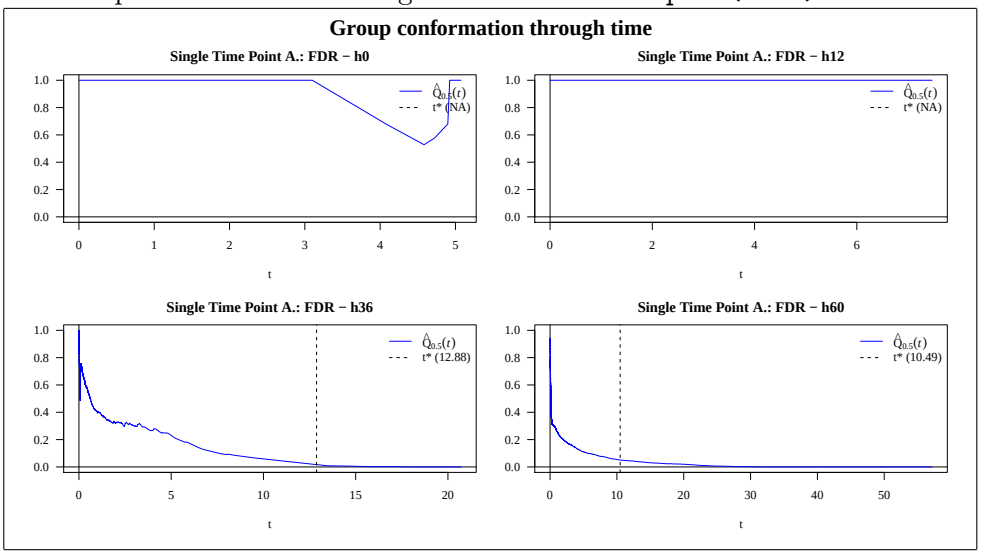
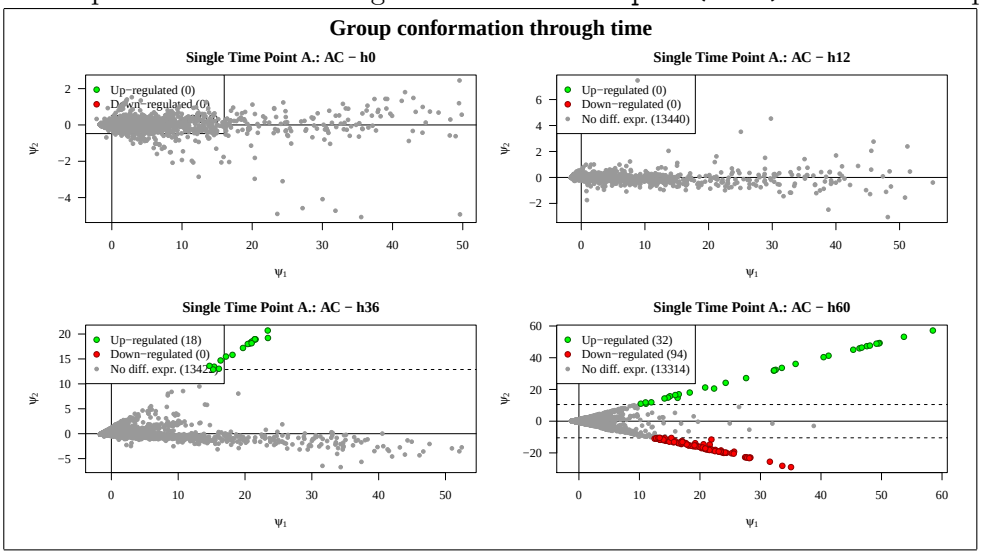


Figure 4: Groups conformation through time results from `plot(tcPI)`. Artificial components.



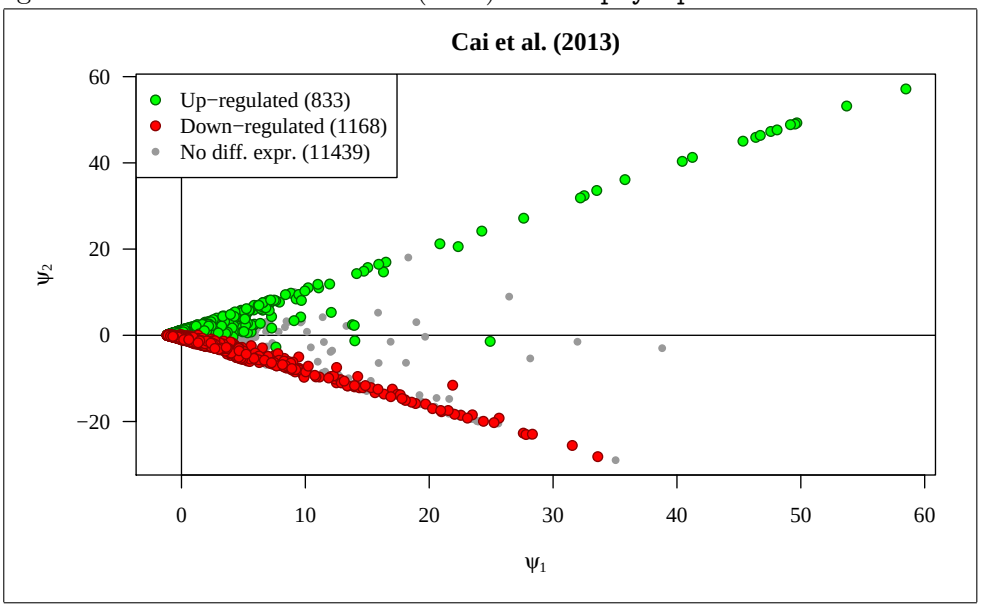
3.3. Comparison with other methods

Cai *et al.* (2013) applied SAM[6] and a two factor (cultivar and time point) ANOVA to identify differentially expressed genes between the tomato plants in the `phytophthora` data set and another near isogenic tomato line (M82). Although their main objectives were different from ours, it is possible to extract the groups of up and down regulated genes only for the `phytophthora` data set at 60 hai from their analysis[7]. We compare their results with our findings in Table 4 and Figure 5.

Table 4: Comparison with Cai et al. (2013) for the `phytophthora` data set 60 hai.

acde	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 5: Results from Cai *et al.* (2013) for the `phytophthora` data set 60 hai.



While our method found 2 up regulated and 53 down regulated genes previously unidentified by Cai *et al.* (2013), they still identified a much larger number of genes. This is a consequence of row standardization as required by ANOVA and SAM and their corresponding univariate point of view (see Section 2.2). Indeed, when row standardization is performed, the inherent scale of genetic expression in the data 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

[6]Significance Analysis of Microarray, see Tusher *et al.* (2001).
[7]Tables S2 and S3, and clusters 1, 2, 5–10 from table S1 in Cai *et al.* (2013).

close to the origin of the artificial plane in Figure 5, which means that their overall expression levels are very close to the average overall expression level in the 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, the fact that SAM and ANOVA identify more genes as differentially expressed than **acde** 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 2.2) and the corresponding implied definitions of differential expression.

4. Conclusions

Throughout this work, we presented a multivariate inferential method for the identification of differentially expressed genes in gene expression data and its implementation in the **acde** package for R. While resting on a very general probabilistic model, the applicability of our method lies upon the key biological and technical assumptions summarized in Biological Scenario 2 from Section 2.2. If these assumptions hold, as is generally the case in gene expression data, a multivariate approach is needed in order to avoid identifying non–expressed genes as differentially expressed. Until now, no multivariate inferential approach appropriate for Biological Scenario 2 had been proposed.

Our method is based on the work of Storey and Tibshirani (2001, 2003) for the estimation of the FDR and on the construction of two artificial components 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 FDRs between different time points proved to be very valuable in this regard.

Additional assessments were proposed in order to gain more statistical assurance with respect to the results obtained with our method. These were the complementary approaches for time course analysis and the computation of a BCa confidence upper bound for the FDR. These additional assessments constitute the final pieces of an integral strategy for the identification of differentially expressed genes.

Our analysis of the `phytophthora` data set resulted in 32 defence related genes identified as up regulated and 94 primary metabolic function related genes identified as down regulated at 60 hai. After comparison with previous results (Cai *et al.* 2013), a large number of 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, genes with true zero expression levels may be wrongly identified as being differentially expressed.

Finally, as a rule, univariate oriented methods will identify 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. Therefore, these 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.

5. Session Info

- R version 3.2.0 (2015-04-16), `x86_64-apple-darwin13.4.0`
- Locale: `en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: BiocInstaller 1.18.1
- Loaded via a namespace (and not attached): tools 3.2.0

References

Acosta JP (2015). *Strategy for Multivariate Identification of Differentially Expressed Genes in Microarray Data*. Statistics MS Thesis, Universidad Nacional de Colombia. Unpublished.

Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, 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.

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.

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.

Canty A, Ripley B (2015). *boot: Bootstrap R (S-Plus) Functions*. R package version 1.3-14.

Chambers J (2008). *Software for data analysis: programming with R*. Springer Science & Business Media.

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

Dudoit S, Yang YH, Callow MJ, Speed TP (2002). "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments." *Statistica sinica*, **12**(1), 111–140.

Efron B, Tibshirani R (1994). *An Introduction to the Bootstrap*, volume 57. Chapman & Hall/CRC.

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 MK, Martin M, Churchill GA (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*, volume 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.

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 WE, Smart CD (2005). "Gene expression profiling of infection of tomato by Phytophthora infestans in the field." *Phytopathology*, **95**(S88).

Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, Iyer V, Jeffrey SS, 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.

Simon RM, Korn EL, McShane LM, Radmacher MD, Wright GW, Zhao Y (2003). *Design and Analysis of DNA Microarray Investigations*. Springer.

Storey JD (2002). "A direct approach to false discovery rates." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **64**(3), 479–498.

Storey JD (2003). "The Positive False Discovery Rate: A Bayesian Interpretation and the q-value." *Annals of statistics*, pp. 2013–2035.

Storey JD, Taylor JE, 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 JD, Tibshirani R (2001). "Estimating False Discovery Rates Under Dependence, with Applications to DNA Microarrays." *Technical report, Department of Statistics, Stanford University*.

Storey JD, 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 VG, 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.

Xiong H, Brown J, Boley N, Bickel P, Huang H (2014). *Statistical Analysis of Next Generation Sequencing Data*. Springer.

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.

A. Appendix: Technical details in acde

A.1. Other functions in acde

While the main functionalities of the **acde** package reside in two functions, `stp` and `tc`, the package includes several other functions for internal use and for additional or preliminary assesments. Following the *Prime Directive* for developping thrustworthy software in Chambers (2008), all functions (internal included) are exported in the namespace of the package and, thus, are directly accessible to the user. Here, we briefly explain this other functions and their role in the computations for the single time point and time course analyses.

The `ac` and `ac2` functions compute the artificial components of gene expression data and are used by functions `tc`, `stp`, `fdr` and `bcaFDR`. Their arguments, `Z` and `design`, are the same as in the `stp` function. As a preliminary analysis, function `ac` may be useful for assessing if Biological Scenario 2 holds, without the lengthy computations of `stp` and `tc` functions. A simple way to do this is to verify that the genes' distribution on the artificial plane is consistent with Biological Scenario 2 following the heuristic guidelines in 1. To obtain a plot of the artificial plane, just type `plot(ac(Z, design))`. Also, the inertia ratio for data set `Z` can be directly computed with `var(ac2(Z, des)) / eigen(cor(Z))$values[1]`.

Function `fdr` performs step 2 of Algorithm 1 and returns estimates of the FDR and the parameter π_0. Function `bcaFDR` performs steps 2 to 4 in Algorithm 2 to compute the BCa confidence upper bound for the FDR. Both functions are called upon by function `stp`. Their arguments are the same as in the `stp` function. Note that these functions are for internal use, and, so, do not check for the validity of the arguments they recieve. These verifications are made within the `stp` and `tc` functions when needed. Finally, the function `qval` computes the genes' Q-Values following Algorithm 3. Its arguments are the estimated FDRs (as object `$Q` returned by `fdr` or `stp`) and the second artificial component (as returned by `ac2`).

With this in mind, the `tc` function makes calls to the `stp` function one or several times (depending on the approach specified in the `method` argument) and the `stp` function, in turn, makes calls to the `ac`, `ac2`, `fdr`, `bcaFDR` and `qval` functions as needed.

A.2. Parallel computation

As both `stp` and `tc` functions use the `boot` function (package **boot**, Canty and Ripley 2015), parallel computation is fairly straight forward for single time point and time course analyses. The argument '`...`' in both `stp` and `tc` functions refers specifically to parallel computation related arguments in the `boot` function. A simple setup for computation in two clusters is `parallel="multicore", ncpus=2`. For more details regarding parallel computation and possible arguments specification, please refer to the help page of the `boot` function in R.

A.3. Construction of this vignette

Because of the lengthy computations for obtaining the BCa confidence upper bound in Figure 1, the compilation of the code presented in this document requires considerable time. In order to guarantee confortable installation times, this vignette[8] is constructed loading a workspace session from the file `resVignette.rda` in the directory `acde/vignettes` in the source of the **acde** package. However, the results can be replicated using the following code for obtaining objects `stpPI` and `tcPI`.

```
> ## Object stpPI
> set.seed(73, kind="Mersenne-Twister")
> des <- c(rep(1,8), rep(2,8))
> stpPI <- stp(phytophthora[[3]], desPI[[3]], BCa=TRUE)

> ## Object tcPI
> set.seed(27, kind="Mersenne-Twister")
> desPI <- vector("list",4)
> for(tp in 1:4) desPI[[tp]] <- c(rep(1,8), rep(2,8))
> tcPI <- tc(phytophthora, desPI)
```

When running this code with different random seeds, because of the shape of the estimated FDRs (slopes approaching zero as t increases in Figure 3), small changes in its estimates (vertical displacements of the plots) can produce large changes in t^* and, subsequently, in the number of differentially expressed genes identified. While setting $B = 100$ is enough for obtaining stable estimates of the FDR, a much larger B is needed in order to obtain stable groups of differentially expressed genes.

A.4. Future perspectives

In future versions of the **acde** package, we hope to extend the previous analysis to include a version of artificial components similar to the factors obtained from the correspondence analysis framework (Lebart *et al.* 1995) instead of PCA, and which may be more suitable when Biological Scenario 1 holds. Meanwhile, please enjoy the package and let us know of any comments or suggestions for future improvements!

Affiliation:

Liliana López-Kleine
Department of Statistics
Faculty of Sciences
Universidad Nacional de Colombia
111321 Bogotá, Colombia
E-mail: llopezk@unal.edu.co
URL: http://www.docentes.unal.edu.co/llopezk/

[8] See the `acde.Rwd` file in source.