lmdme: Linear Models on Designed Multivariate Experiments in R

Cristóbal Fresno
Universidad Católica de Córdoba

Mónica G Balzarini
Universidad Nacional de Córdoba

Elmer A Fernández
Universidad Católica de Córdoba

Abstract

This introduction to linear model on designed multivariate experiments of the R package **lmdme** is a (slightly) modified version of **?**, published in the *Journal of Statistical Software*.

The **lmdme** package (**?**) decomposes analysis of variance (ANOVA) through linear models on designed multivariate experiments, allowing ANOVA-principal component analysis (APCA) and ANOVA-simultaneous component analysis (ASCA) in R (**?**). It also extends both methods with the application of partial least squares (PLS) through the specification of a desired output matrix. The package is freely available on the Bioconductor website (**?**), licensed under GNU general public license.

ANOVA decomposition methods for designed multivariate experiments are becoming popular in "omics" experiments (transcriptomics, metabolomics, etc.), where measurements are performed according to a predefined experimental design (**?**), with several experimental factors or including subject-specific clinical covariates, such as those present in current clinical genomic studies. ANOVA-PCA and ASCA are well-suited methods for studying interaction patterns on multidimensional datasets. However, current R implementation of APCA is only available for *Spectra* data in **ChemoSpec** package (**?**), whereas ASCA (**?**) is based on average calculations on the indices of up to three design matrices. Thus, no statistical inference on estimated effects is provided. Moreover, ASCA is not available in R package format.

Here, we present an R implementation for ANOVA decomposition with PCA/PLS analysis that allows the user to specify (through a flexible `formula` interface), almost any linear model with the associated inference on the estimated effects, as well as to display functions to explore results both of PCA and PLS. We describe the model, its implementation and one high-throughput *microarray* example applied to interaction pattern analysis.

Keywords: linear model, ANOVA decomposition, PCA, PLS, designed experiments, R.

1. Introduction

Current "omics" experiments (proteomics, transcriptomics, metabolomics or genomics) are multivariate in nature. Modern technology allows us to explore the whole genome or a big subset of the proteome, where each gene/protein is in essence a variable explored to elucidate its relationship with an outcome. In addition, these experiments are including an increasing number of experimental factors (time, dose, etc.) from design or subject-specific information, such as age, gender and linage, and are available for analysis. Hence, to decipher experimental design or subject-specific patterns, some multivariate approaches should be applied, with principal component analysis (PCA) and partial least squares regression (PLS) being the most common. However, it is known that working with raw data might mask information of interest. Therefore, analysis of variance (ANOVA)-based decomposition is becoming popular to split variability sources before applying such multivariate approaches. Seminal works on genomics were that of **?** on ANOVA-PCA (APCA) and of **?** on ANOVA-SCA (ASCA) models. However, to the best of our knowledge R implementation of APCA is only available for *Spectra* data, **ChemoSpec** R package by **?**. Regarding ASCA, as there is no R package for this model, it can only be used by uploading script-function files resulting from a MATLAB code translation (**?**). In addition, ASCA only accepts up to three design matrices, which limits its use and makes it difficult. Moreover, coefficient estimations are based on average calculations using binary design matrices, without any statistical inference over them.
Here, we provide a flexible linear model-based decomposition framework. Almost any model can be specified, according to the experimental design, by means of a flexible `formula` interface. Because coefficient estimation is carried out by means of maximum likelihood, statistical significance is naturally given. The framework also provides both capacity to perform PCA and PLS analysis on appropriate ANOVA decomposition results, as well as graphical representations. The implementation is well-suited for direct analysis of gene expression matrices (variables on rows) from high-throughput data such as *microarray* or *RNA-seq* experiments. Below we provide an examples to introduce the user to the package applications, through the exploration of interaction patterns on a microarray experiment.

2. The model

A detailed explanation of ANOVA decomposition and multivariate analysis can be found in **?** and **?**. Briefly and without the loss of generality, let us assume a *microarray* experiment where the expression of $(G_1,\ G_2,\ \ldots\ ,\ G_g)$ genes are arrayed in a chip. In this context, let us consider an experimental design with two main factors: A, with a levels $(A_1,\ A_2,\ \ldots\ ,\ A_i,\ \ldots\ ,\ A_a)$ and B, with b levels $(B_1,\ B_2,\ \ldots\ ,\ B_j,\ \ldots\ ,\ B_b)$, with replicates $R_1,\ R_2,\ \ldots\ ,\ R_k,\ \ldots\ ,\ R_r$ for each $A \times B$ combination levels. After preprocessing steps are performed as described in (**?**), each chip is represented by a column vector of gene expression measurements of $g \times 1$. Then, the whole experimental data is arranged into a $g \times n$ expression matrix (X), where $n = a \times b \times r$. In this data scheme, single gene measurements across the different treatment combinations $(A_i \times B_j)$ are presented in a row on the X matrix, as depicted in Figure **??**. An equivalent X matrix structure needs to be obtained for *2D-DIGE* or *RNA-seq* experiments and so forth.

Regardless of data generation, the ANOVA model for each gene (row) in X can be expressed

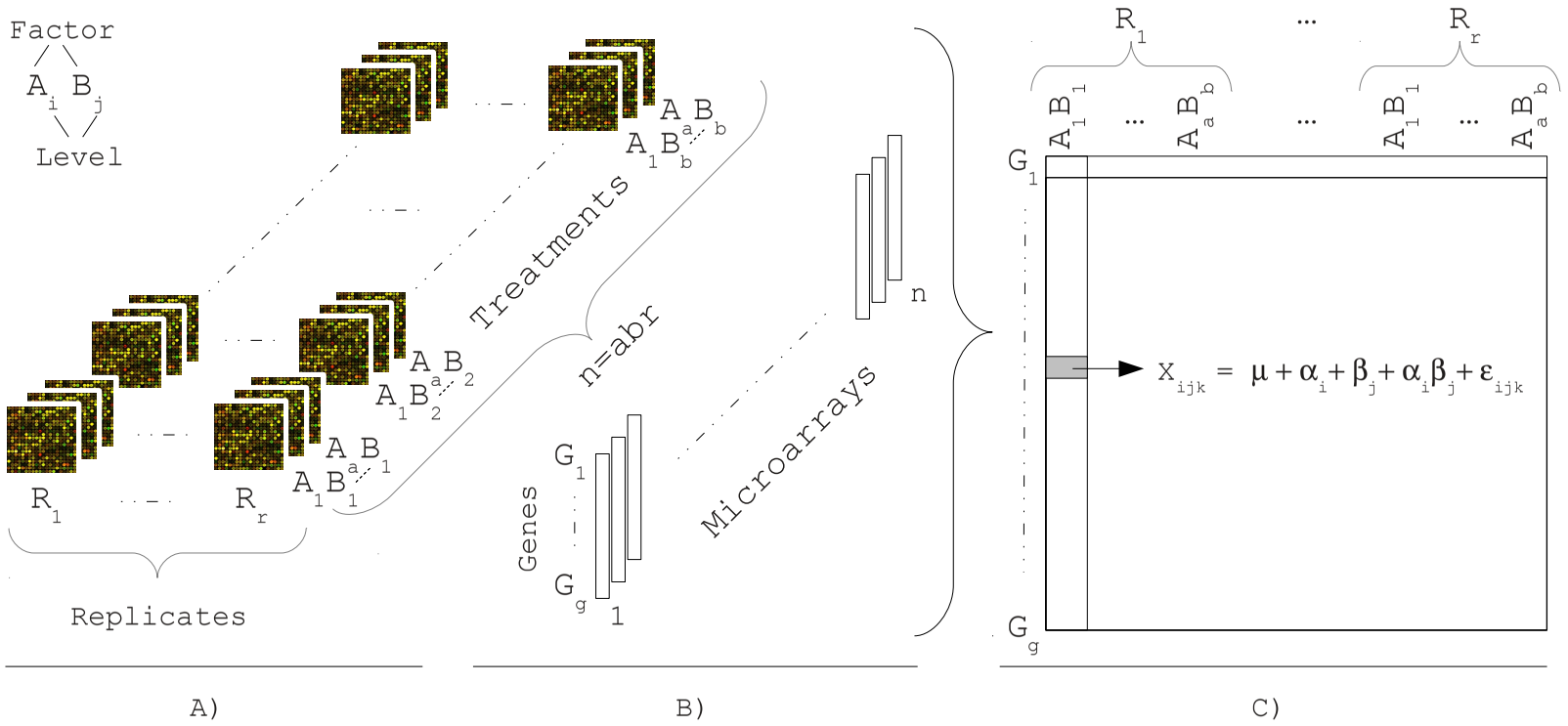


Figure 1: Data representation of microarray gene expression. A) Genes are spotted on the chip. Then, expression levels for each combination of treatment factor levels A_iB_j and their replicates R_k can be measured on the chips, yielding a total of $n = a \times b \times r$ microarrays. B) Gene expression of each chip (microarray) is then interpreted as a column vector of expression levels. C) Then, these column vectors will be combined by columns producing the experiment gene expression matrix X. Expression measurements under all treatment combinations for a gene are represented by the X matrix rows. Thus, measurements on a row are subjected to the ANOVA model of (**??**).

as (**??**):

$$x_{ijk} = \mu + \alpha_i + \beta_j + \alpha_i \times \beta_j + \varepsilon_{ijk} \tag{1}$$

Where x_{ijk} is the measured expression for "some" gene, at combination "ij" of factors A and B for the k replicate; μ is the overall mean; α, β and $\alpha \times \beta$ are the main and interaction effects respectively; and the error term $\varepsilon_{ijk} \sim N(0, \sigma^2)$. In addition, (**??**) can also be expressed in matrix form for all genes into (**??**):

$$X = X_\mu + X_\alpha + X_\beta + X_{\alpha\beta} + E = \sum_{l \in \{\mu,\alpha,\beta,\alpha\beta\}} X_l + E \tag{2}$$

Where X_l, E matrices are of dimension $g \times n$ and contain the level means of the corresponding $l - th$ term and the random error respectively. However, in the context of linear models X_l can also be written as a linear combination of two matrix multiplications in the form of (**??**):

$$X = \sum_{l \in \{\mu,\alpha,\beta,\alpha\beta\}} X_l + E = \sum_{l \in \{\mu,\alpha,\beta,\alpha\beta\}} B_l Z_l^T + E = B_\mu Z_\mu^T + \ldots + B_{\alpha\beta} Z_{\alpha\beta}^T + E = \mu 1^\top + B_\alpha Z_\alpha^T + \ldots + B_{\alpha\beta} Z_{\alpha\beta}^T + E \tag{3}$$

Where B_l and Z_l are referenced in the literature as *coefficient* and *model* matrices of dimensions $g \times m_{(l)}$ and $n \times m_{(l)}$, respectively, and $m_{(l)}$ is the number of levels of factor l. The first term is usually called *intercept*, with $B_\mu = \mu$ and $Z_\mu = \mathbf{1}$ being of dimension $g \times 1$ and $n \times 1$, respectively. In this example, all Z_l are binary matrices, identifying whether a measurement belongs ("1") or not ("0") to the corresponding factor.
In the implementations provided by **?** and **?**, the estimation of the coefficient matrices is based on calculations of *averages* using the design matrix (up to three design matrices $Z_{\alpha,\beta,\alpha\beta}$), to identify the average samples. In theory, these authors fully decompose the original matrix as shown in (**??**). On the contrary, in this package the model coefficients are estimated, iteratively, by the *maximum likelihood* approach, using the `lmFit` function provided by **limma** package (**?**). Consequently, three desirable features are also incorporated:

1. *Flexible* `formula` *interface* to specify any potential model. The user only needs to provide: i) The gene expression `matrix` (X), ii) The experimental `data.frame` (`design`) with treatment structure, and iii) The model in a `formula` style, just as in an ordinary `lm` R function. Internal `model.matrix` call, will automatically build the appropriate Z matrices, overcoming the constraint on factorial design size, and tedious model matrix definitions.

2. *Hypothesis tests* on coefficient B_l matrices. A T test is automatically carried out for the $s - th$ gene model, to test whether or not the $o - th$ coefficient is equal to zero, i.e., $H_0 : b_{so} = 0$ vs $H_1 : b_{so} \neq 0$. In addition, an F test is performed to simultaneously determine whether or not all b_{so} are equal to zero.

3. *Empirical Bayes correction* can also be achieved through the `eBayes` **limma** function. It uses an empirical Bayes method to shrink the row/gene-wise sample variances towards a common value and to augment the degrees of freedom for the individual variances (**?**).

By contrast, **?** estimate the main and interaction effects by overall mean subtraction. Hence, genes need to be treated as an additional factor. Meanwhile, in **?** and **?** implementations,

the estimations are obtained on a gene-by-gene basis, as in (**??**). Therefore, in a two-way factor experiment, such as *time* × *oxygen*, De Haan's model includes two additional double interactions and a triple interaction, because genes are treated as a factor, unlike the models of Smilde and Nueda.

2.1. The decomposition algorithm

The ANOVA model (**??**) is decomposed iteratively using (**??**), where in each step the $l-th$ coefficients $\hat{B}_l$, $\hat{E}_l$ matrices and $\hat{\sigma}^2_l$ are estimated. Then, the particular term contribution matrix $\hat{X}_l = \hat{B}_l Z_l^\top$ is subtracted from the preceding residuals to feed the next model, as depicted in (**??**):

$$\begin{aligned}
X &= X_\mu + X_\alpha + X_\beta + X_{\alpha\beta} + E = \sum_{l\in\{\mu,\alpha,\beta,\alpha\beta\}} X_l + E \\
step\ \mu: \quad X &= X_\mu + E_\mu \Rightarrow X = \hat{B}_\mu Z_\mu^\top + \hat{E}_\mu \Rightarrow \hat{E}_\mu = X - \hat{B}_\mu Z_\mu^\top \\
step\ \alpha: \quad E_\mu &= X_\alpha + E_\alpha \Rightarrow \hat{E}_\mu = \hat{B}_\alpha Z_\alpha^\top + \hat{E}_\alpha \Rightarrow \hat{E}_\alpha = \hat{E}_\mu - \hat{B}_\alpha Z_\alpha^\top \\
&\vdots \quad \vdots \\
step\ l: E_{l-1} &= X_l + E_l \Rightarrow \hat{E}_{l-1} = \hat{B}_l Z_l^\top + \hat{E}_l \Rightarrow \hat{E}_l = \hat{E}_{l-1} - \hat{B}_l Z_l^\top \\
&\vdots \quad \vdots \\
step\ \alpha\beta: \quad E_\beta &= X_{\alpha\beta} + E \Rightarrow \hat{E}_\beta = \hat{B}_{\alpha\beta} Z_{\alpha\beta}^\top + \hat{E} \Rightarrow \hat{E} = \hat{E}_\beta - \hat{B}_{\alpha\beta} Z_{\alpha\beta}^\top
\end{aligned} \tag{4}$$

Where the hat ("^") denotes estimated coefficients. In this implementation, the first step always estimates the *intercept* term, i.e., `formula=~1` in R style, with $\hat{B}_\mu = \hat{\mu}$ and $Z_\mu = 1$. The following models will only include the $l-th$ factor without the intercept, i.e., `formula=~lth_term-1`, where lth_term stands for α, β or $\alpha\beta$ in this example. This procedure is quite similar to the one proposed by **?**.

2.2. PCA and PLS analyses

These methods explain the variance/covariance structure of a set of observations (e.g., genes) through a few linear combinations of variables (e.g., experimental conditions). Both methods can be applied to the $l-th$ ANOVA decomposed step of (**??**) to deal with different aspects:

- PCA concerns with the *variance* of a single matrix, usually with the main objectives of reducing and interpreting data. Accordingly, depending on the matrix to which it is applied, there are two possible methods: ASCA, when PCA is applied to *coefficient* matrix, $\hat{B}_l$, (**?**); and APCA when PCA is calculated on the *residual*, $\hat{E}_{l-1}$. The latter is conceptually an ASCA and is usually applied to, $X_l + E$, i.e., the mean factor matrix X_l, plus the error of the fully decomposed model E of (**??**), as in **?**.

- PLS not only generalizes but also combines features from PCA and regression to explore the *covariance* structure between input and some output matrices, as described by **?** and **?**. It is particularly useful when one or several dependent variables (outputs - O) must be predicted from a large and potentially highly correlated set of independent variables (inputs). In our implementation, the input can be either the *coefficient* matrix

$\hat{B}_l$ or the *residual* $\hat{E}_{l-1}$. According to the choice, the respective output matrix will be a diagonal `O=diag(nrow(`$\hat{B}_l$`))` or design matrix $O = Z_l$. In addition, users can specify their own output matrix, O, to verify a particular hypothesis. For instance, in functional genomics it could be the Gene Ontology class matrix as used in gene set enrichment analysis (GSEA) by **?**.

When working with the *coefficient* matrix, the user will not have to worry about the expected number of components in X (rank of the matrix, given the number of replicates per treatment level), as suggested by **?**, because the components are directly summarized in the coefficient $\hat{B}_l$ matrix. In addition, for both PCA/PLS, the **lmdme** package (**?**) also offers different methods to visualize results, e.g., `biplot`, `loadingplot` and `screeplot` or `leverage` calculation, in order to filter out rows/genes as in **?**.

3. Example

In this section we provide an overview of **lmdme** package by **?**. The example consists of an application of the analysis of gene expression interaction pattern, where we address: How to define the model, undertake ANOVA decomposition, perform PCA/PLS analysis and visualize the results.
From here onwards, some outputs were removed for reasons of clarity and the examples were performed with `options(digits=4)`.

3.1. Package overview

The original data files for the first example are available at Gene Expression Omnibus (**?**), with accession GSE37761 and **stemHypoxia** package (**?**) on the Bioconductor website. In this dataset, **?** studied differentiation of human embryonic stem cells under hypoxia conditions. They measured gene expression at different time points under controlled oxygen levels. This experiment has a typical two-way ANOVA structure, where factor A stands for "*time*" with $a = 3$ levels $\{0.5,\ 1,\ 5\ days\}$, factor B stands for "*oxygen*" with $b = 3$ levels $\{1,\ 5,\ 21\%\}$ and $r = 2$ replicates, yielding a total of 18 samples. The remainder of the dataset was excluded in order to have a balanced design, as suggested by **?** to fulfil orthogonality assumptions in ANOVA decomposition.
First, we need to load **stemHypoxia** package to access R objects calling `data("stemHypoxia")`, which will then load the experimental `design` and gene expression intensities `M`.

```
R> library("stemHypoxia")
R> data("stemHypoxia")
```

Now we manipulate the `design` object to maintain only those treatment levels which create a balanced dataset. Then, we change `rownames(M)` of each gene in `M`, with their corresponding `M$Gene_ID`.

```
R> timeIndex<-design$time %in% c(0.5, 1, 5)
R> oxygenIndex<-design$oxygen %in% c(1, 5, 21)
R> design<-design[timeIndex & oxygenIndex, ]
```

```
R> design$time<-as.factor(design$time)
R> design$oxygen<-as.factor(design$oxygen)
R> rownames(M)<-M$Gene_ID
R> M<-M[, colnames(M) %in% design$samplename]
```

Now we can explore microarray gene expression data present on the `M` matrix, with $g = 40736$ rows (individuals/genes) and $n = 18$ columns (samples/microarrays). In addition, the experimental `design` data.frame contains main effect columns (e.g., *time* and *oxygen*) and the sample names (`samplename`). A brief summary of these objects is shown using `head` function:

```
R> head(design)
```

```
  time oxygen samplename
3  0.5      1    12h_1_1
4  0.5      1    12h_1_2
5  0.5      5    12h_5_1
6  0.5      5    12h_5_2
7  0.5     21   12h_21_1
8  0.5     21   12h_21_2
```

```
R> head(M)[, 1:3]
```

```
             12h_1_1 12h_1_2 12h_5_1
A_24_P66027    7.182   7.512   8.225
A_32_P77178    6.385   6.035   6.440
A_23_P212522   9.562   9.390   9.211
A_24_P934473   6.288   6.397   6.265
A_24_P9671    12.007  11.995  12.282
A_32_P29551   10.176   9.273   9.360
```

Once the preprocessing of the experiment data is completed, `library("lmdme")` should be loaded. This instruction will automatically load the required packages: **limma** (**?**) and **pls** (**?**). Once the data are loaded, the ANOVA decomposition of Section **??** can be carried out using (**??**) calling `lmdme` function with the `model` formula, actual `data` and experimental `design`.

```
R> library("lmdme")
R> fit<-lmdme(model=~time*oxygen, data=M, design=design)
R> fit
```

```
lmdme object:
Data dimension:  40736  x  18
Design (head):
  time oxygen samplename
3  0.5      1    12h_1_1
4  0.5      1    12h_1_2
5  0.5      5    12h_5_1
```

```
6  0.5      5    12h_5_2
7  0.5     21   12h_21_1
8  0.5     21   12h_21_2

Model:~time * oxygen
Model decomposition:
  Step       Names            Formula CoefCols
1    1 (Intercept)                ~ 1        1
2    2        time         ~ -1 + time        3
3    3      oxygen       ~ -1 + oxygen        3
4    4 time:oxygen ~ -1 + time:oxygen        9
```

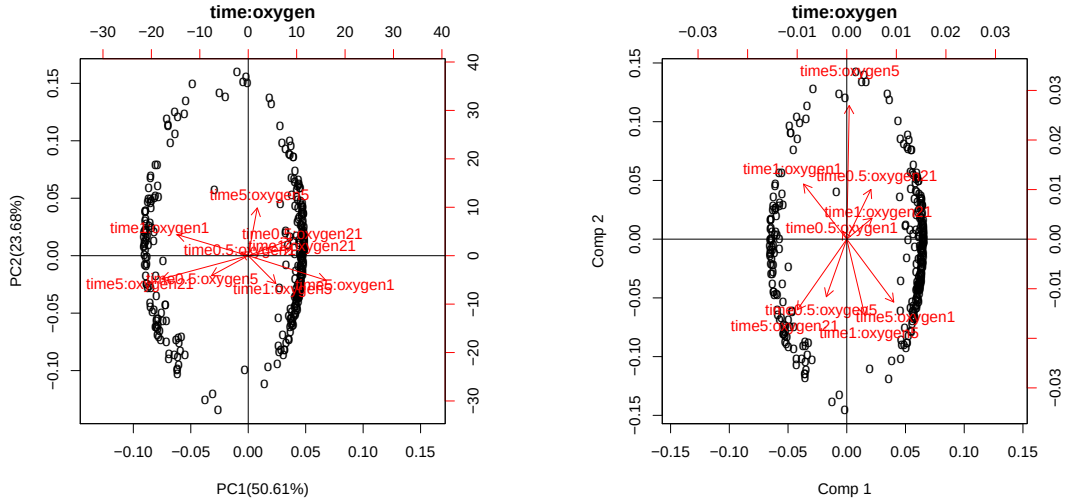
The results of `lmdme` will be stored inside the `fit` object, which is an S4 R class. By invoking the `fit` object, a brief description of the *data* and *design* used are shown as well as the `Model` applied and a summary of the decomposition. This `data.frame` describes the applied `Formula` and `Names` for each `Step`, as well as the amount of estimated coefficients for each gene (`CoefCols`).
At this point, we can choose those subjects/genes in which at least one interaction coefficient is statistically different from zero (F test on the coefficients) with a threshold p value of 0.001 and perform ASCA on the interaction *coefficient* `term`, and PLS against the identity matrix (default option).

```
R> id<-F.p.values(fit, term="time:oxygen")<0.001
R> decomposition(fit, decomposition="pca", type="coefficient",
+    term="time:oxygen", subset=id, scale="row")
R> fit.plsr<-fit
R> decomposition(fit.plsr, decomposition="plsr", type="coefficient",
+    term="time:oxygen", subset=id, scale="row")
```

These instructions will perform ASCA and PLS `decomposition` over the `scale="row"` version of the 305 selected subjects/genes (`subset=id`) on `fit` and `fit.plsr` object, respectively. The results will be stored inside these objects. In addition, we have explicitly indicated the decomposition `type="coefficient"` (default value) in order to apply it to the `coefficient` matrix, on `"time:oxygen"` interaction `term` ($\hat{B}_{\alpha\beta}$).
Now, we can visualize the associated `biplots` (see Figure **??** **??** and **??**).

```
R> biplot(fit, xlabs="o", expand=0.7)
R> biplot(fit.plsr, which="loadings", xlabs="o",
+    ylabs=colnames(coefficients(fit.plsr, term="time:oxygen")),
+    var.axes=TRUE)
```

For visual clarity, `xlabs` are changed with the `"o"` symbol, instead of using the `rownames(M)` with manufacturer ids, and second axis with the `expand=0.7` option to avoid cutting off loading labels. In addition, PLS `biplot` is modified from the default **pls** behavior to obtain a graph similar to ASCA output (`which="loadings"`). Accordingly, `ylabs` is changed to match the corresponding `coefficients` of the interaction `term` and `var.axes` is set to `TRUE`.



(a) ANOVA simultaneous component analysis

(b) ANOVA partial least squares

Figure 2: Biplot on the decomposed interaction coefficients ($time \times oxygen$) on genes satisfying the F test with p value < 0.001. Notice that the interaction matrix in the ASCA model is of rank 9. Thus, 9 arrows are expected and the score of the 305 selected subjects are projected onto the space spanned by the first two principal components in Figure **??**.

The ASCA biplot of the first two components (see Figure **??**), explain over 70% of the coefficient variance. The genes are arranged in an elliptical shape. Thus, it can be observed that some genes tend to interact with different combinations of time and oxygen. A similar behavior is observed in PLS biplot of Figure **??**.

The interaction effect on the `fit` object can also be displayed using the `loadingplot` function (see Figure **??**). For every combination of two consecutive levels of factors (time and oxygen), the figure shows an interaction effect on the first component, which explains 50.61% of the total variance of the "time:oxygen" term.

```
R> loadingplot(fit, term.x="time", term.y="oxygen")
```

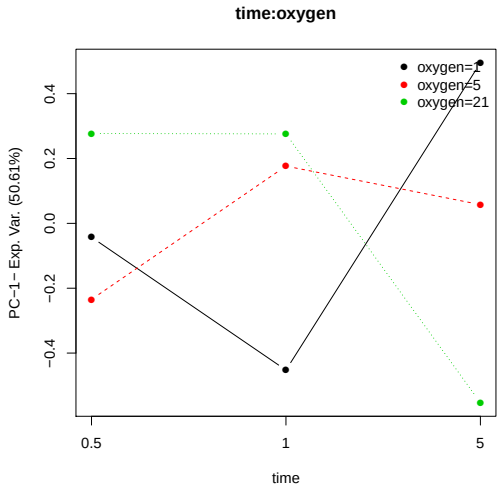


Figure 3: ANOVA simultaneous component analysis `loadingplot` on genes satisfying the F test with p value < 0.001 on the interaction coefficients ($time \times oxygen$).

In the case of an ANOVA-PCA/PLS analysis, the user only needs to change the `type="residuals"` parameter in the `decomposition` function and perform a similar exploration.

Acknowledgements

Funding: This work was supported by the National Agency for Promoting Science and Technology, Argentina (PICT00667/07 to E.A.F. and PICT 2008-0807 BID to E.A.F.), Córdoba Ministry of Science and Technology, Argentina (PID2008 to E.A.F and PIP2009 to M.G.B.), Catholic University of Córdoba, Argentina and National Council of Scientific and Technical Research (CONICET), Argentina.

Session Info

```
R> sessionInfo()

R version 3.3.0 RC (2016-04-26 r70550)
Platform: x86_64-apple-darwin13.4.0 (64-bit)
Running under: OS X 10.9.5 (Mavericks)

locale:
[1] C/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods
[7] base

other attached packages:
[1] lmdme_1.14.0      pls_2.5-0         stemHypoxia_1.7.0

loaded via a namespace (and not attached):
[1] limma_3.28.0 tools_3.3.0
```

Affiliation:

Cristóbal Fresno & Elmer A Fernández
Bioscience Data Mining Group
Faculty of Engineering
Universidad Católica de Córdoba
X5016DHK Córdoba, Argentina
E-mail: cfresno@bdmg.com.ar, efernandez@bdmg.com.ar
URL: http://www.bdmg.com.ar/

Mónica G Balzarini
Biometry Department
Faculty of Agronomy
Universidad Nacional de Córdoba
X5000JVP Córdoba, Argentina
E-mail: mbalzari@gmail.com
URL: http://www.infostat.com.ar/