Journal of Statistical Software

January 2014, Volume 56, Issue 7. http://www.jstatsoft.org/

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

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 from **Bioconductor** and licensed under the 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, currently an R implementation of APCA is only available for *Spectra* data in the **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 an R package.

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 two high-throughput *microarray* examples: one applied to interaction pattern analysis and the other to quality assessment.

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, which are then 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 (PLS) regression 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 Haan, Wehrens, Bauerschmidt, Piek, Schaik, and Buydens (2007) on ANOVA-PCA (APCA) and of Smilde, Jansen, Hoefsloot, Lamers, Greef, and Timmerman (2005) on ANOVA-SCA (ASCA) models. However, to the best of our knowledge an R (R Core Team 2013) implementation of APCA is only available for *Spectra* data in the R package **ChemoSpec** by Hanson (2012). 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 (The MathWorks, Inc. 2011) code translation (Nueda *et al.* 2007). In addition, ASCA only accepts up to three design matrices, which limits its use. Moreover, coefficient estimations are based on average calculations using binary design matrices, without any statistical inference available for 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 the 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 two examples to introduce the user to the application of the package, through the exploration of interaction patterns and assessment of microarray experiment quality.

2. The model

A detailed explanation of ANOVA decomposition and multivariate analysis can be found in Smilde *et al.* (2005) and Zwanenburg, Hoefsloot, Westerhuis, Jansen, and Smilde (2011). 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 Smyth (2004), 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 1. An equivalent X matrix structure needs to be obtained for *2D-DIGE* or *RNA-seq* experiments and so forth.

Regardless of the data generation, the ANOVA model for each gene (row) in X can be expressed as (1):

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

where x_{ijk} is the measured expression for "some" gene, at combination "ij" of factors A and B for replicate k; 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, (1) can also be expressed in matrix form for all genes:

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

where the matrices X_l and E are of dimension $g \times n$ and contain the level means of the corresponding lth 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 (3):

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

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 Smilde *et al.* (2005) and Nueda *et al.* (2007), 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 (1). 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 (Smyth *et al.* 2011). 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 using a `formula` interface, just as in a call to the R function `lm`. Internal a `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 matrices B_l. A T test is automatically carried out for the sth gene model, to test whether or not the oth 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.

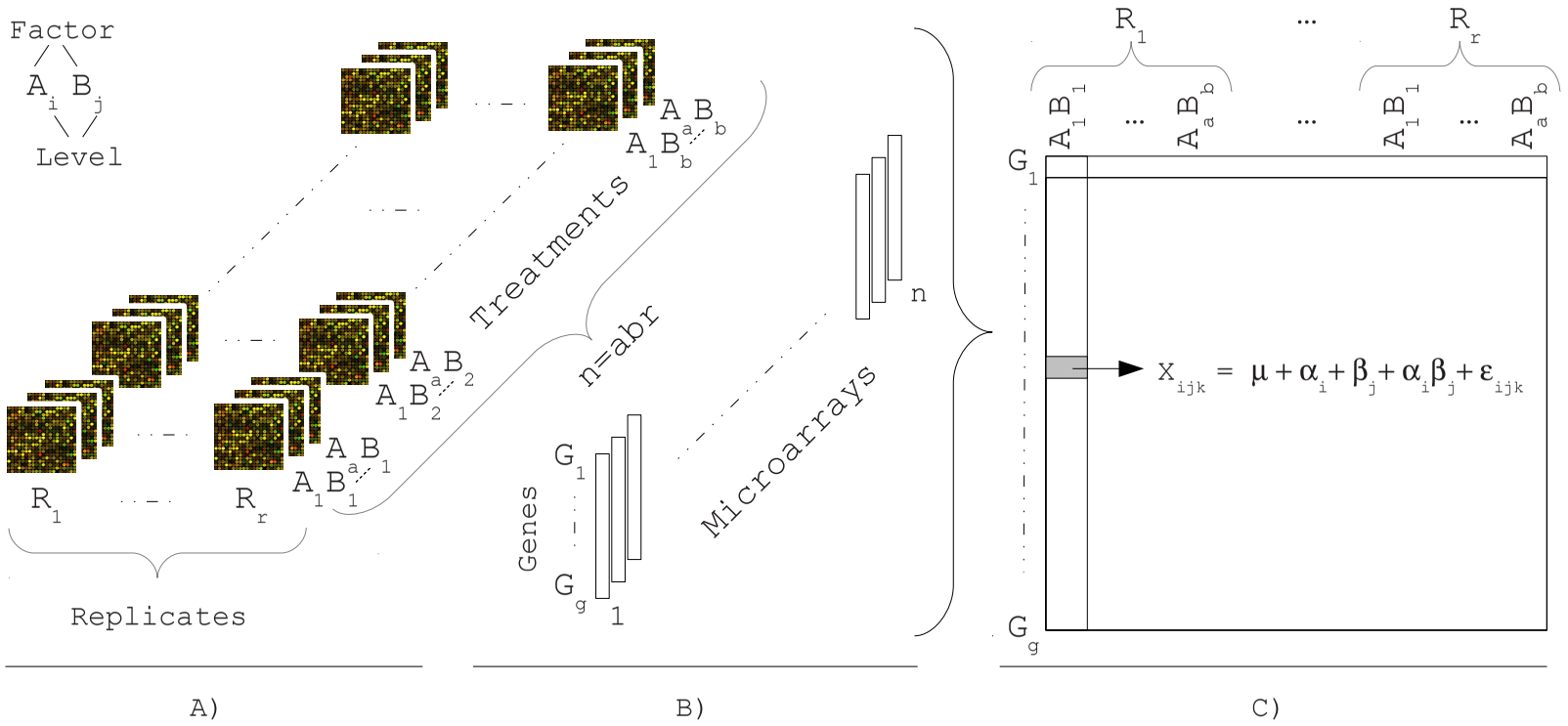


Figure 1: Data representation of microarray gene expression. A) Genes are spotted on the chip. Then, expression levels for each combination of treatment factor levels A_iB_j and their replicates R_k can be measured on the chips, yielding a total of $n = a \times b \times r$ microarrays. B) Gene expression of each chip (microarray) is then interpreted as a column vector of expression levels. C) Then, these column vectors are 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 (1).

3. *Empirical Bayes correction* can also be achieved through the `eBayes` function in package **limma**. 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 (Smyth 2004).

By contrast, Haan *et al.* (2007) estimate the main and interaction effects by overall mean subtraction. Hence, genes need to be treated as an additional factor. Meanwhile, in the implementations by Smilde *et al.* (2005) and Nueda *et al.* (2007), the estimations are obtained on a gene-by-gene basis, as in (1). 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 by Smilde *et al.* (2005) and Nueda *et al.* (2007).

2.1. The decomposition algorithm

The ANOVA model (2) is decomposed iteratively using (3), where in each step the lth coefficients $\hat{B}_l$, $\hat{E}_l$ matrices and $\hat{\sigma}_l^2$ are determined. 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 (4):

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

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 lth 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 de B. Harrington, Vieira, Espinoza, Nien, Romero, and Yergey (2005).

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 lth ANOVA decomposed step of (4) 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 the *coefficient*

matrix, $\hat{B}_l$, (Smilde *et al.* 2005); 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 (1), as in Haan *et al.* (2007).

- 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 Abdi and Williams (2010) and Shawe-Taylor and Cristianini (2004). 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 Subramanian, Tamayo, Mootha, Mukherjee, Ebert, Gillette, Paulovich, Pomeroy, Golub, Lander, and Mesirov (2005).

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 Smilde *et al.* (2005), because the components are directly summarized in the coefficient $\hat{B}_l$ matrix. In addition, for both PCA/PLS, the **lmdme** package (Fresno and Fernández 2013a) 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 Tarazona, Prado-López, Dopazo, Ferrer, and Conesa (2012).

3. Examples

In this section we provide an overview of the **lmdme** package (Fresno and Fernández 2013a), using two examples. The package is freely available on the **Bioconductor** website (Gentleman *et al.* 2004), licensed under the GNU General Public License. The first 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. In the second example, the method is applied to assess the quality of high-throughput microarray data.

From here onwards, some outputs were removed for reasons of clarity and the examples were performed with `options(digits = 4)`.

3.1. Example 1: Package overview

The original data files for the first example are available at Gene Expression Omnibus (Edgar, Domrachev, and Lash 2002), with accession GSE37761 and in the **stemHypoxia** package (Fresno and Fernández 2013b) on the **Bioconductor** website. In this dataset, Prado-Lopez *et al.* (2010) 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 Smilde *et al.* (2005) to fulfil orthogonality assumptions in the ANOVA decomposition.

First, we load the data, which consists of the experimental `design` and gene expression intensities `M`.

```
R> data("stemHypoxia", package = "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 in the `M` matrix, with $g = 40736$ rows (individuals/genes) and $n = 18$ columns (samples/microarrays). In addition, the experimental design data frame `design` contains main effect columns (e.g., *time* and *oxygen*) and the sample names (`samplename`). A brief summary of these objects is shown using the `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, package **lmdme** needs to be loaded. This instruction will automatically load the required packages, i.e., **limma** (Smyth *et al.* 2011) and **pls** (Mevik, Wehrens, and Liland 2011). Once the data are loaded, the ANOVA decomposition of Section 2.1 can be carried out using (4) by calling `lmdme` function with the `model` formula, the actual `data` and the experimental `design`.

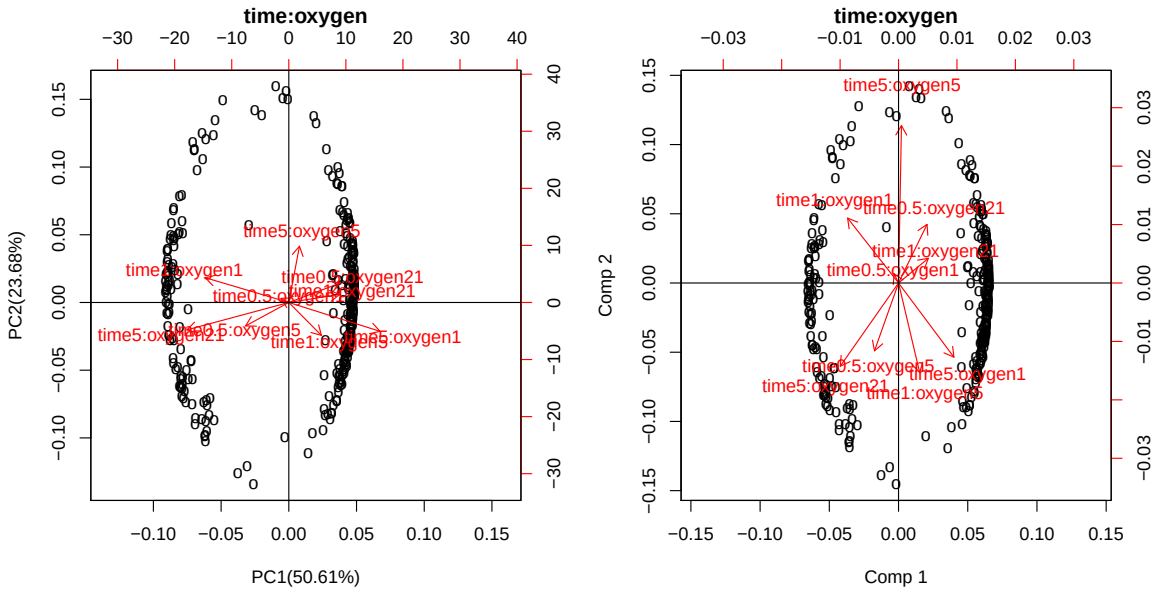
```
R> library("lmdme")
R> fit <- lmdme(model = ~time * oxygen, data = M, design = design)
R> fit

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 class. By printing 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 the '`fit`' and '`fit.plsr`' object,



(a) ANOVA simultaneous component analysis (b) ANOVA PLS

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 2(a).

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 2 (a) and (b)).

```
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 the second axis is printed 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`.

The ASCA biplot of the first two components (see Figure 2(a)), 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 the PLS biplot in Figure 2(b).

The interaction effect on the '`fit`' object can also be displayed using the `loadingplot` function (see Figure 3). For every combination of two consecutive levels of factors (time and oxygen),

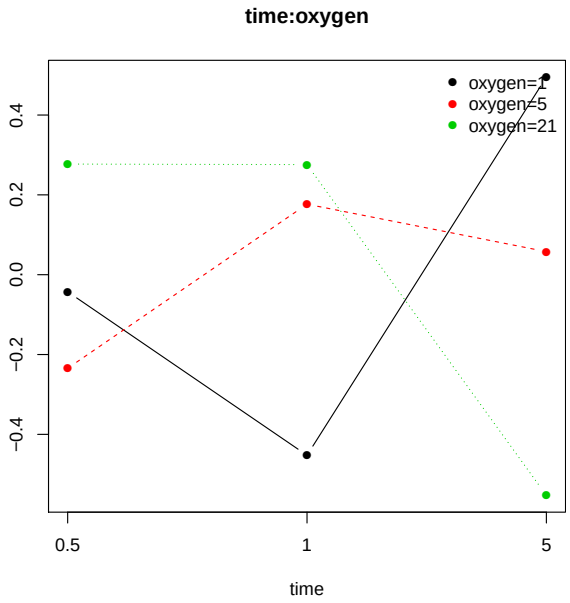


Figure 3: ANOVA simultaneous component analysis `loadingplot` on genes satisfying the F test with p value < 0.001 on the interaction coefficients (*time* $\times$ *oxygen*).

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")
```

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, as will be shown in Section 3.2.

3.2. Example 2: Application to quality assessment

In this example we use two-color *microarray* technology to explore gene expression profiles (data available as supplementary material and at http://www.bdmg.com.ar/). Expression intensity at different time points under diverse substrate growing conditions (protein concentration) on melanoma cell lines was measured. This experiment also has a two-way ANOVA structure, where factor A stands for "*time*" with $a = 3$ levels $\{0.5,\ 4,\ 12\ hours\}$, factor B stands for "*concentration*" with $b = 3$ levels $\{0,\ 1,\ 10\ units\}$ and $r = 3$ replicates, yielding a total of 27 samples. Data owners are particularly interested in finding genes with a differential expression using an F test with a p value < 0.05 for the *time* $\times$ *concentration* interaction term, which they have already confirmed in previous experiments. Preliminary results on differential expression analysis using **limma** did not show any interaction pattern. Here we show that, by means of the **lmdme** approach, we were able to identify unexpected technical effects that could bias biological interpretation and demonstrated how to remove this unexpected artefact through package **lmdme**.

Once again, we need to load the **lmdme** package and experimental data, which were previously stored on file. Using `load(file = "example2.RData")` the experimental `design` and gene

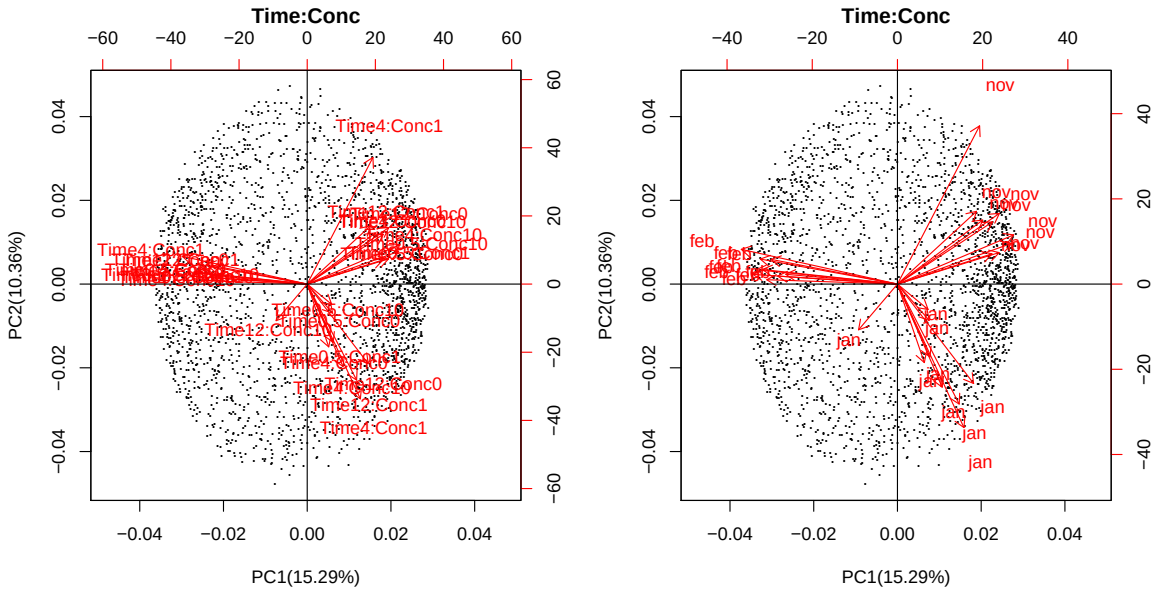
expression intensities, M, will be loaded. It is always recommended to explore these objects, to check if they were properly loaded, using the head function, as we did in the previous example.

```
R> library("lmdme")
R> load(file = "example2.RData")
R> head(design)

  Time Conc SampleName HybridDate
1  0.5    0 221732.gpr        nov
2  0.5    0 338515.gpr        jan
3  0.5    0 339577.gpr        feb
4  0.5    1 221678.gpr        nov
5  0.5    1 338514.gpr        jan
6  0.5    1 339576.gpr        feb

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

     221732.gpr 338515.gpr 339577.gpr
[1,]     0.1287     0.1181    0.72294
[2,]    -0.1653    -0.1080    0.10825
[3,]    -0.5227    -0.2300   -0.29959
[4,]     0.3142     0.5636    0.07366
[5,]     0.1519     0.2008   -1.10059
[6,]     0.2542    -0.1083   -0.40284
```

The dimension of matrix M is $g = 2520$ rows (individuals/genes) and $n = 27$ columns (samples/microarrays). In addition, the experimental design data frame design contains main effect columns (i.e., *Time* and *Conc* for concentration), the SampleName and the date when the chips were hybridized (HybridDate).

Using the lmdme function we can fit model = ~ Time * Conc using empirical Bayes = TRUE correction and verbose = TRUE to give the user feedback about the progress of the ANOVA decomposition. In addition, we can check if the results obtained by the data owners about non-differently expressed gene for the interaction term were correct.

```
R> fit <- lmdme(model = ~ Time * Conc, data = M, design = design,
+    Bayes = TRUE, verbose = TRUE)

testing:   ~ 1
testing:   ~ Time -1
testing:   ~ Conc -1
testing:   ~ Time:Conc -1

R> id.fit <- F.p.values(fit, term = "Time:Conc") < 0.05
R> sum(id.fit)

[1] 0
```



(a) Original `biplot` (b) Using `ylabs = design$HybridDate`

Figure 4: ANOVA-PCA `biplot` on the interaction residuals (*time* × *concentration*).

The result of `sum(id.fit)` which is equal to 0 promotes further exploration of the data. In this context the APCA approach can be applied to the 'fit' object to get a visual exploration on the `biplot` on `term = "Time:Conc"` (see Figure 4(a)).

```
R> decomposition(fit, "pca", scale = "row", type = "residual")
R> biplot(fit, term = "Time:Conc", xlabs = ".", expand = 0.9)
```

Some strange, uncontrolled variability source pattern seems to cluster the chips into three groups (see Figure 4(a)). By inspecting the data frame `design`, we decided to label `HybridData` (Hybridization Date) to explore a possible relationship between the observed `biplot` clusters.

```
R> biplot(fit, term = "Time:Conc", ylabs = design$HybridDate, xlabs = ".",
+     expand = 0.8)
```

Figure 4(b) shows that the cluster structure may be associated with hybridization date, an unconsidered variability source.

Given this evidence, we can use PLS with a user-defined `Omatrix` using the `model.matrix` function with `~HybridDate - 1` with the `design` object and ask whether or not the data cope or not with this structure.

```
R> decomposition(fit, "plsr", scale = "row", type = "residual",
+     term = "Time:Conc",
+     Omatrix = model.matrix(~ HybridDate - 1, design))
R> biplot(fit, term = "Time:Conc", which = "loadings", xlabs = ".",
+     var.axes = TRUE)
```

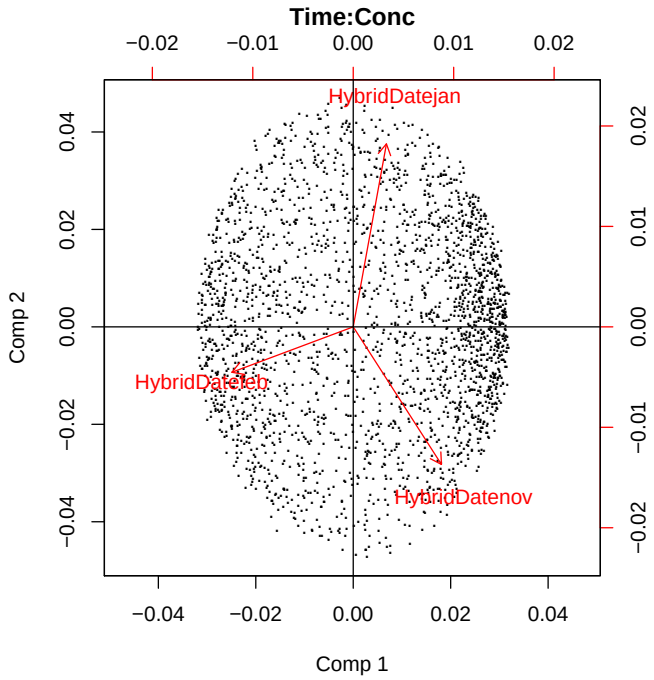


Figure 5: Biplot of PLS regression on the interaction residuals (*time* × *concentration*) using the hybridization date as output matrix.

In addition, visual exploration of the resulting `biplot` of Figure 5 proved our assumption. The data owners explained to us that, the original deployment was planned to hybridize the three replicates on the same day. But, due to custom constraints, it had to be modified to hybridize one replicate per shipment reception: the first in November (`nov`), the second in January (`jan`) and the last one in February (`feb`). The confirmation of our data exploration with the constraint in randomization suggests that `HybridDate` should be included in the model:

```
R> fit.date <- lmdme(model = ~ HybridDate + Time * Conc, data = M,
+    design = design, Bayes = TRUE)
R> id.fit.date <- F.p.values(fit.date, term = "Time:Conc") < 0.05
R> sum(id.fit.date)

[1] 13
```

By including `HybridDate` in the model, we were able to estimate and remove this effect. Then, the statistical inference about the individuals/genes has been modified, showing 13 candidates affected by *time* × *concentration* levels. In addition, the corresponding APCA `biplot` of Figure 6 shows that the previous pattern of Figure 4(a) was removed.

```
R> decomposition(fit.date, "pca", scale = "row", type = "residual")
R> biplot(fit.date, term = "Time:Conc", xlabs = ".", expand = 0.8)
```

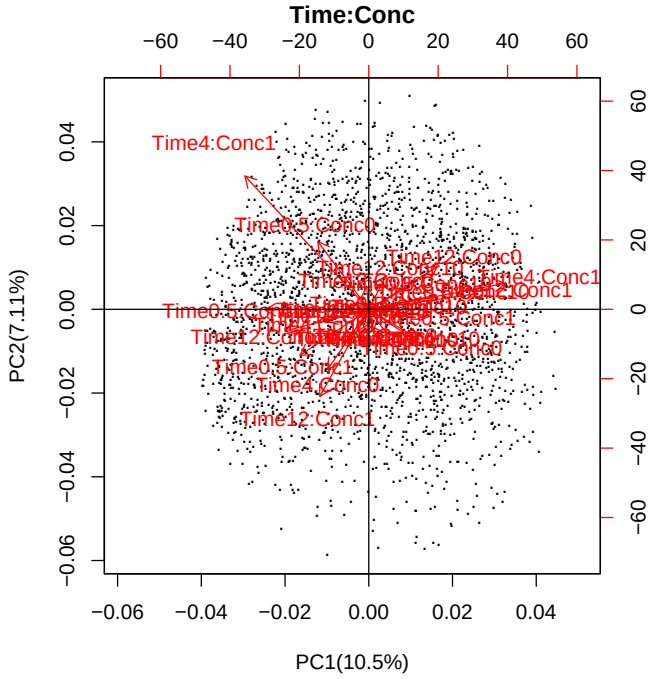


Figure 6: ANOVA-PCA `biplot` on the interaction residuals (*time* × *concentration*) including hybridization date in the model.

Acknowledgments

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.

Data: Thanks to Osvaldo Podhajcer and Edgardo Salvatierra from the Laboratory of Molecular and Cellular Therapy at Leloir Institute, Buenos Aires, Argentina for letting us use their yet *unpublished data* and making it freely available from `http://www.jstatsoft.org/`.

References

Abdi H, Williams LJ (2010). "Principal Component Analysis." *Wiley Interdisciplinary Reviews: Computational Statistics*, **2**(4), 433–459.

de B Harrington P, Vieira NE, Espinoza J, Nien JK, Romero R, Yergey AL (2005). "Analysis of Variance – Principal Component Analysis: A Soft Tool for Proteomic Discovery." *Analytica Chimica Acta*, **544**(1), 118–127.

Edgar R, Domrachev M, Lash AE (2002). "Gene Expression Omnibus: NCBI Gene Expression and Hybridization Array Data Repository." *Nucleic Acids Research*, **30**(1), 207–210.

Fresno C, Fernández EA (2013a). ***lmdme**: Linear Model Decomposition for Designed Multivariate Experiments*. R package version 1.2.1, URL http://www.Bioconductor.org/packages/release/bioc/html/lmdme.html.

Fresno C, Fernández EA (2013b). ***stemHypoxia**: Differentiation of Human Embryonic Stem Cells Under Hypoxia Gene Expression Dataset by Prado-Lopez et al. (2010)*. R package version 0.99.3, URL http://www.Bioconductor.org/packages/release/data/experiment/html/stemHypoxia.html.

Gentleman RC, Carey VJ, Bates DM, Bolstad B, Dettling M, Dudoit S, Ellis B, Gautier L, Ge Y, Gentry J, Hornik K, Hothorn T, Huber W, Iacus S, Irizarry R, Leisch F, Li C, Maechler M, Rossini AJ, Sawitzki G, Smith C, Smyth G, Tierney L, Yang JYH, Zhang J (2004). "**Bioconductor**: Open Software Development for Computational Biology and Bioinformatics." *Genome Biology*, **5**(10), R80. URL http://genomebiology.com/2004/5/10/R80.

Haan JRD, Wehrens R, Bauerschmidt S, Piek E, Schaik RCV, Buydens LMC (2007). "Interpretation of ANOVA Models for Microarray Data Using PCA." *Bioinformatics*, **23**(2), 184–190.

Hanson BA (2012). ***ChemoSpec**: Exploratory Chemometrics for Spectroscopy*. R package version 1.51-2, URL http://CRAN.R-project.org/package=ChemoSpec.

Mevik BH, Wehrens R, Liland KH (2011). ***pls**: Partial Least Squares and Principal Component Regression*. R package version 2.3-0, URL http://CRAN.R-project.org/package=pls.

Nueda MJ, Conesa A, Westerhuis JA, Hoefsloot HCJ, Smilde AK, Talón M, Ferrer A (2007). "Discovering Gene Expression Patterns in Time Course Microarray Experiments by ANOVA-SCA." *Bioinformatics*, **23**(14), 1792–1800.

Prado-Lopez S, Conesa A, Armiñán A, Martínez-Losa M, Escobedo-Lucea C, Gandia C, Tarazona S, Melguizo D, Blesa D, Montaner D, Sanz-González S, Sepúlveda P, Götz S, O'Connor JE, Moreno R, Dopazo J, Burks DJ, Stojkovic M (2010). "Hypoxia Promotes Efficient Differentiation of Human Embryonic Stem Cells to Functional Endothelium." *Stem Cells*, **28**(3), 407–418.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Shawe-Taylor J, Cristianini N (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Smilde AK, Jansen JJ, Hoefsloot HCJ, Lamers RJAN, Greef JVD, Timmerman ME (2005). "ANOVA-Simultaneous Component Analysis (ASCA): A New Tool for Analysing Designed Metabolomics Data." *Bioinformatics*, **21**(13), 3043–3048.

Smyth GK (2004). "Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments." *Statistical Applications in Genetics and Molecular Biology*, **3**(1), Article 3.

Smyth GK, Ritchie M, Silver J, Wettenhall J, Thorne N, Langaas M, Ferkingstad E, Davy M, Pepin F, Choi D, McCarthy D, Wu D, Oshlack A, de Graaf C, Hu Y, Shi W, Phipson B (2011). ***limma**: Linear Models for Microarray Data*. R package version 3.12.1, URL http://www.Bioconductor.org/packages/release/bioc/html/limma.html.

Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, Mesirov JP (2005). "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles." *Proceedings of the National Academy of Sciences of the United States of America*, **102**(43), 15545–15550.

Tarazona S, Prado-López S, Dopazo J, Ferrer A, Conesa A (2012). "Variable Selection for Multifactorial Genomic Data." *Chemometrics and Intelligent Laboratory Systems*, **110**(1), 113–122.

The MathWorks, Inc (2011). *MATLAB – The Language of Technical Computing, Version R2011b*. The MathWorks, Inc., Natick, Massachusetts. URL http://www.mathworks.com/products/matlab/.

Zwanenburg G, Hoefsloot HCJ, Westerhuis JA, Jansen JJ, Smilde AK (2011). "ANOVA-Principal Component Analysis and ANOVA-Simultaneous Component Analysis: A Comparison." *Journal of Chemometrics*, **25**(10), 561–567.

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/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 56, Issue 7 *Submitted:* 2012-10-04
January 2014 *Accepted:* 2013-06-26