Introduction to Multivariate Analysis of Microarray Gene Expression Data using MADE4

Aedín Culhane

May 18, 2005

Contents

1 Introduction

The package *made4* facilitates multivariate analysis of microarray gene expression data. The package provides a set of functions that utilise and extend multivariate statistical and graphical functions available in *ade4*, (1). *made4* accepts gene expression data is a wide variety of input formats, including Bioconductor formats, *AffyBatch*, *exprSet*, *marrayRaw*, and `data.frame` or `matrix`.

1.1 Installation

made4 requires that *ade4* is installed. *made4* also calls *scatterplot3d*. These can be installed, using `install.packages()`. To install *made4*

```
install.packages("made4",
    contriburl = "http://bioinf.ucd.ie/people/aedin/R/current")
```

1.2 Further help

More information about *made4* is available at `http://bioinf.ucd.ie/people/aedin/R`. This document provides an overview of *made4* functions. These are described in more detail in the other vingettes that accompany this package.

Extensive tutorials, examples and documentation on multivariate statistical methods are available from the ade4 website `http://pbil.univ-lyon1.fr/ADE-4` and *ade4* user support is available through the ADE4 mailing list. The *ade4* homepage is `http://pbil.univ-lyon1.fr/ADE-4`.

This tutorial assumes a basic knowledge of R, but we have found that Emmanuel Paradis's **R for Beginners** is a very good guide to those unfamiliar with R. This is available at `http://cran.r-project.org/doc/contrib/rdebuts_en.pdf`.

This documents assumes that data is normalised and preprocessed. Please refer to the Bioconductor packages *affy*, *arrayMagic* and *limma*, for input and initial pre-processing of microarray data. The Bioconductor project website is `http://www.bioconductor.org`.

2 Quickstart

We will very briefly demonstrate some of the functions in *made4*. To do this we will use a small dataset that is available in the Bioconductor package *factDesign*. This is a dataset of gene expression levels for 500 genes from Affymetrix HGU95av2 chips for eight samples from a breast cancer cell line. Load the necessary R packages and estrogen dataset.

```
> library(affy)
> library(factDesign)
> library(made4)
> library(ade4)

> data(estrogen)
```

This experiment studied the effect of estrogen on the gene expression in estrogen receptor positive breast cancer cells over time. After serum starvation, samples were exposed to estrogen, and mRNA was harvested at two time points (10 or 48 hours). The control samples were not exposed to estrogen and were harvested at the same time points. Table 1 shows the experiemental design, and corresponding samples names. The full data set (12,625 probes, 32 samples) and its analysis are discussed in Scholtens, et al. Analyzing Factorial Designed Microarray Experiments. Journal of Multivariate Analysis. (To appear). The gene expression values were calculated using the robust multichip average `rma` method (7) after quantile normalization using the *affy* package. The expression values are reported in log base 2 scale.

Table 1: Experimental Conditions for estrogen dataset available in *factDesign*

time	estrogen	
	absent	present
10 hours	et1	Et1
	et2	Et2
48 hours	eT1	ET1
	eT2	ET2

```
> estrogen

Expression Set (exprSet) with
        500 genes
        8 samples
                 phenoData object with 2 variables and 8 cases
         varLabels
                ES: presence or absence of estrogen
                TIME: length of exposure to treatment (hours)

> pData(estrogen)

        ES TIME
et1.CEL  A  10h
et2.CEL  A  10h
Et1.CEL  P  10h
Et2.CEL  P  10h
eT1.CEL  A  48h
eT2.CEL  A  48h
ET1.CEL  P  48h
ET2.CEL  P  48h
```

2.1 Overview

The *made4* function `overview()` provides a quick way to get an overview or feel for data. `overview()` will draw a boxplot, histogram and dendrogram of a hierarchical analysis. Hierarchical clustering is produced using average linkage clustering with a Pearson correlation measure of similarity (5) This gives a quick first glance at the data.

```
> overview(estrogen)
```

Often labelling the samples using a covariate of interest, in this case, the presence of estrogen (ES) or timepoint (TIME) is useful.

```
> overview(estrogen, label = estrogen$TIME)
```

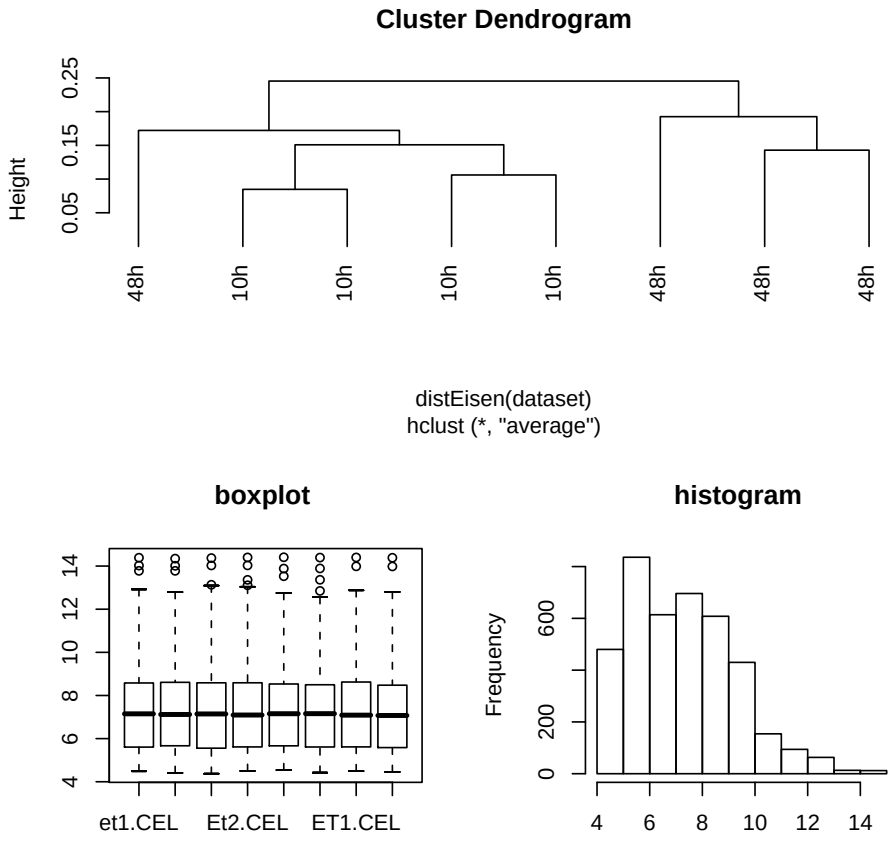

Figure 1: Overview of estrogen data. A) dendrogram showing results of average linkage clustering, B) boxplot and C) histrogram.

2.2 Correspondence Analysis

The function `ord` simplifies the running of ordination methods such as principal component, correspondence or non-symmetric correspondence analysis. It provides a wrapper which can call each of these methods in *ade4*. To run a correspondence analysis (6) on this dataset.

```
> estrogen.coa <- ord(estrogen, type = "coa")
```

Output from `ord` is a list of length 2, containing the ordination results ($ord) and a factor ($fac) if input. The ordination results (estrogen.coa$ord) contain a list of results

(of length 12) which includes the eigenvalues ($eig), and the projected coordinations of the variables ($li, 500 genes) and cases ($co, 8 microarray samples).

```
> names(estrogen.coa)

[1] "ord" "fac"

> estrogen.coa$ord

Duality diagramm
class: coa dudi
$call: dudi.coa(df = data.tr, scannf = FALSE, nf = ord.nf)

$nf: 7 axis-components saved
$rank: 7
eigen values: 0.0007448 0.0003107 0.000109 6.419e-05 4.691e-05 ...
  vector length mode    content
1 $cw    500    numeric column weights
2 $lw    8      numeric row weights
3 $eig   7      numeric eigen values

  data.frame nrow ncol content
1 $tab       8    500  modified array
2 $li        8    7    row coordinates
3 $l1        8    7    row normed scores
4 $co        500  7    column coordinates
5 $c1        500  7    column normed scores
other elements: N
```

2.3 Visualising Results

There are many functions in *ade4* and *made4* for visualising results from ordination analysis. The simplest way to view the results produced by `ord` is to use `plot`. `plot(estrogen.ord)` will draw a plot of the eigenvalues, along with plots of the variables (genes) and a plot of the cases (microarray samples). In this example Microarray samples are colour-coded using the `classvec estrogen$ES`.

```
> plot(estrogen.coa, classvec = estrogen$ES, arraycol = c("green",
+     "blue"), genecol = "pink")
```

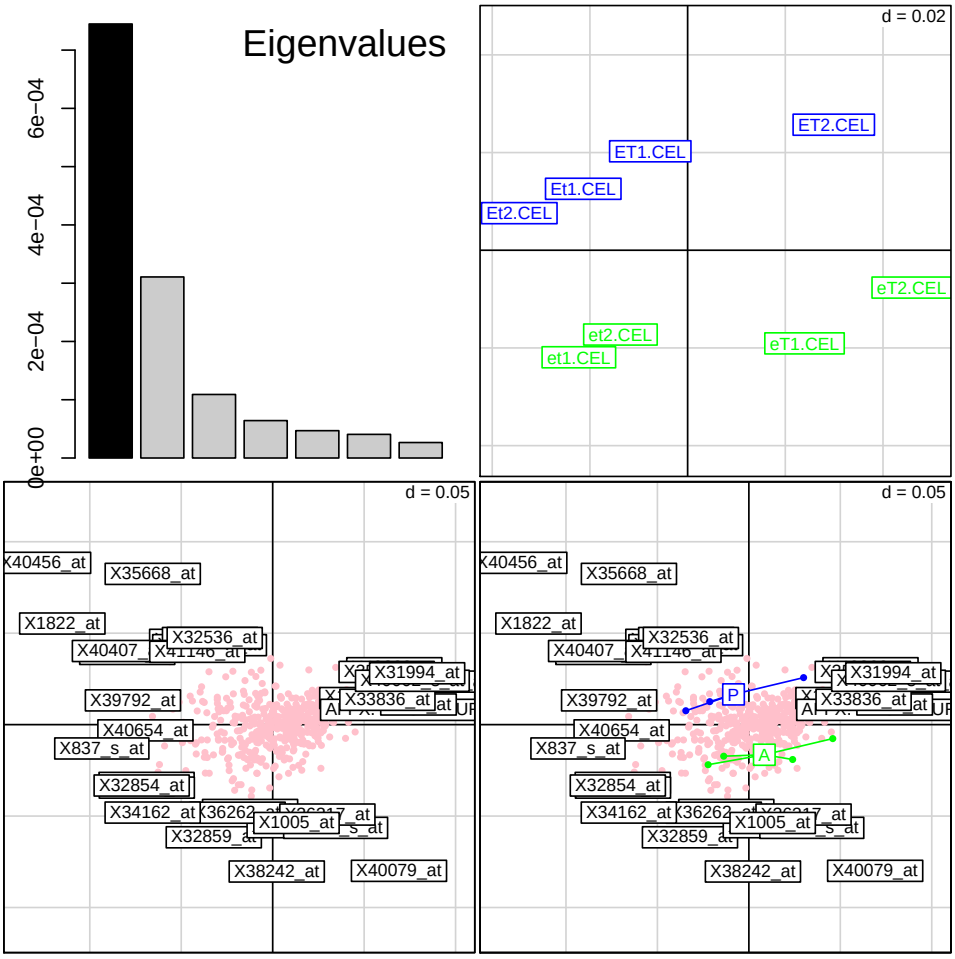

Figure 2: Correspondence analysis of estrogen dataset. A. plot of the eigenvalues, B. projection of microarray samples in which samples incubated in the absence (green squares) or presence (blue squares) of estrogen, C. projection of genes (pink filled circles) and D. biplot showing both genes and samples. Samples and genes with a strong associated are projected in the same direction from the origin. The greater the distance from the origin the stronger the association

Equally, samples could be coloured by time.

```
> plot(estrogen.coa, classvec = estrogen$TIME)
```

Genes and array projections can also be plotted using `s.var` and `s.groups`. The function `s.groups` required a class vector (classvec), and allowed groups to be coloured in different colours. For example, to plot microarray samples (cases),

```
> s.var(estrogen.coa$ord$li)
```

To plot microarray samples, colour by group (estrogen presence) as specified by estrogen$ES

```
> s.groups(estrogen.coa$ord$li, estrogen$ES)
```

Plot gene projections without labels (clab=0). Typically there are a large number of genes, thus it is not feasible to label all of these. The function plotgenes is more useful to use if you wish to add labels when there are lots of variables (genes)

```
> s.var(estrogen.coa$ord$co, clab = 0)
```

The gene projections can be also visualised with `plotgenes`. The number of genes that are labelled at the end of the axis can be defined. The default is 10.

```
> plotgenes(estrogen.coa$ord$co, n = 5, col = "red")
```

By default the variables (genes) are labelled with the rownames of the matrix. Typically these are spot IDs or Affymetrix accession numbers which are not very easy to interpret. But these can be easily labeled by gene symbols, using the *annaffy* annotation package. To retrieve the gene symbols for all of the affymetrix features on the HGU95av2 chip and label genes by gene symbol:

```
> library(annaffy)
> symbs <- aafSymbol(geneNames(estrogen), "hgu95av2")
> gene.symbs <- getText(symbs)
> plotgenes(estrogen.coa$ord$co, n = 10, col = "red",
+     varlabels = gene.symbs)
```

To get a list of variables at the end of an axes, use `topgenes`. For example, to get a list of the 5 genes at the negative and postive end of axes 1.

```
> topgenes(estrogen.coa$ord$co, axis = 1, n = 5)
```

To only the a list of the genes (default 10 genes) at the negative end of the first axes

```
> topgenes(estrogen.coa$ord$co, labels = gene.symbs,
+     end = "neg")

 [1] "SLC39A8" ""        "ME1"     "KPNA2"   "FDPS"
 [6] "HNRPR"   "FBXW11"  "PEX7"    ""        "PJA2"
```

Two lists can be compares using `comparelists`.

```
> plotgenes(estrogen.coa$ord$co, n = 10, col = "red",
+     varlabels = gene.symbs)
```

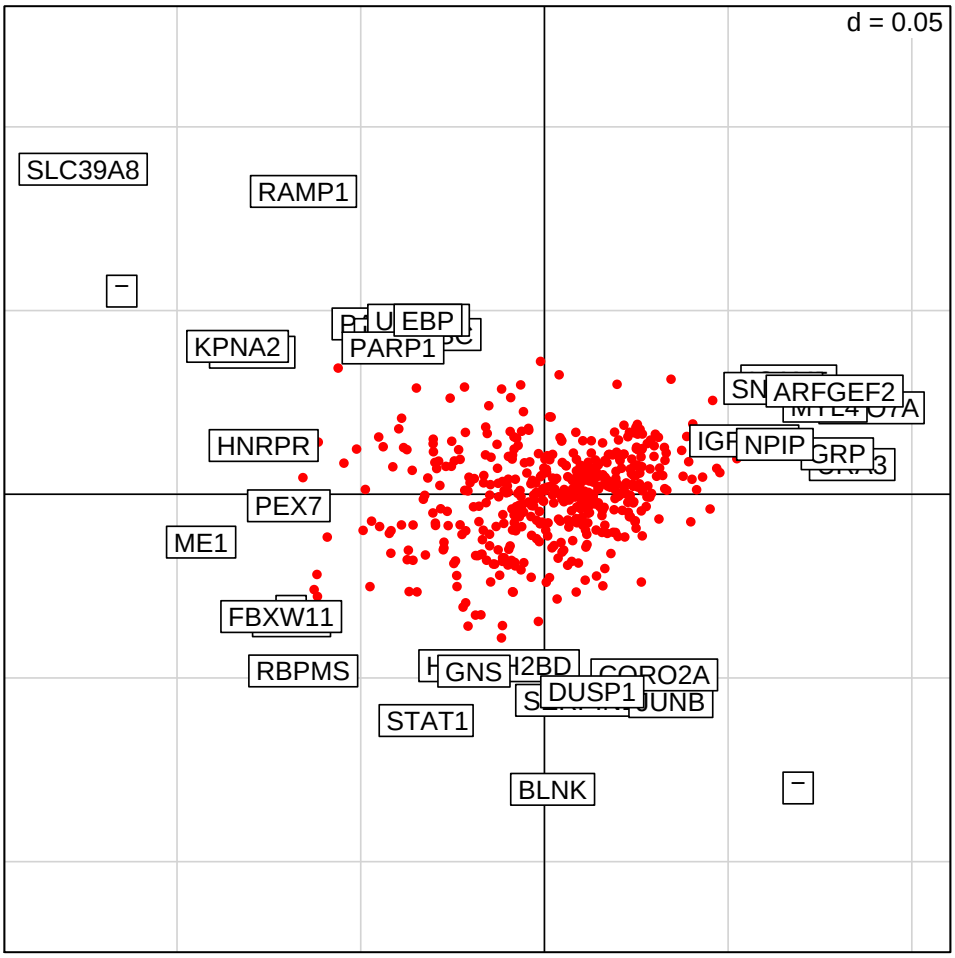

Figure 3: Projection of genes (filled circles) in Correspondence analysis of estrogen dataset. The genes at the ends of each of the axes are labelled with HUGO gene symbols.

To visualise the arrays (or genes) in 3D either use `do3d` or `html3d`. `do3d` is a wrapper for `scatterplot3d`, but is modified so that groups can be coloured. `html3d` produces a "pdb" output which can be visualised using rasmol or chime. Rasmol provides a free and very useful interface for colour, rotating, zooming 3D graphs.

```
> do3d(estrogen.coa$ord$li, classvec = estrogen$TIME,
+     cex.symbols = 3)
> html3D(estrogen.coa$ord$li, estrogen$TIME, writehtml = TRUE)
```

2.4 Classification and Class Prediction using Between Group Analysis

Between Group Analysis (BGA) is a supervised classification method (3). The basis of BGA is to ordinate the groups rather than the individual samples. In tests on two microarray gene expression datasets, BGA performed comparably to supervised classification methods, including support vector machines and artifical neural networks (2). To train a dataset, use `bga`, the projection of test data can be assessed using `suppl`. One leave out cross validation can be performed using `bga.jackknife`. See the BGA vignette for more details on this method.

```
> estrogen.bga <- bga(estrogen, type = "coa", estrogen$TIME)
```

2.5 Meta-analysis of microarray gene expression

Coinertia analysis `cia` (4) has been successfully applied to the cross-platform comparison (meta-analysis) of microarray gene expression datasets (8). CIA is a multivariate method that identifies trends or co-relationships in multiple datasets which contain the same samples. That is either the rows or the columns of a matrix must be "matchable". CIA can be applied to datasets where the number of variables (genes) far exceeds the number of samples (arrays) such is the case with microarray analyses. `cia` calls `coinertia` in the *ade4* package. See the CIA vignette for more details on this method.

```
> data(NCI60)
> coin <- cia(NCI60$Ross, NCI60$Affy)
> names(coin)

[1] "call"      "coinertia" "coa1"      "coa2"

> coin$coinertia$RV

[1] 0.7859656
```

The RV coefficient $RV which is 0.786 in this instance, is a measure of global similarity between the datasets. The greater (scale 0-1) the better.

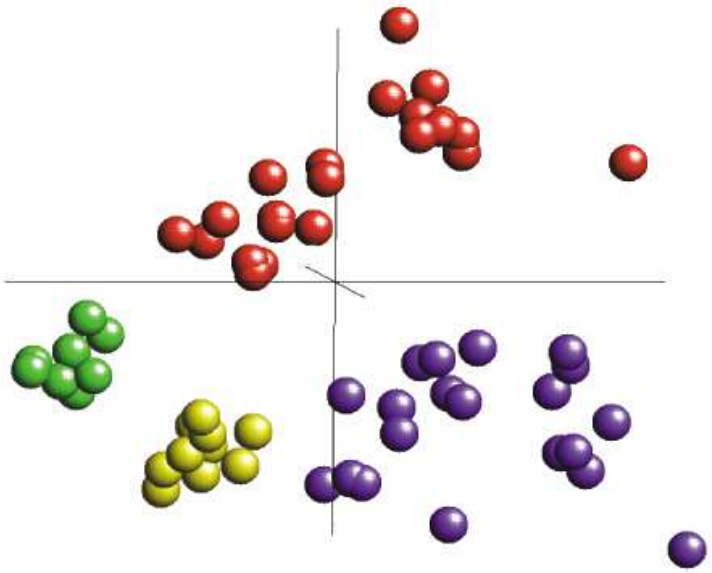

Figure 4: Output from html3D, which can be rotated and visualised on web browsers that can support chime (IE or Netscape on MS Windows or Mac).

```
> plot(estrogen.bga, genelabels = gene.symbs)

 [1]   4  67 173 238 240 279 286 296 306 314 325 372 406 411
[15] 412 415 445 455 464 470
```

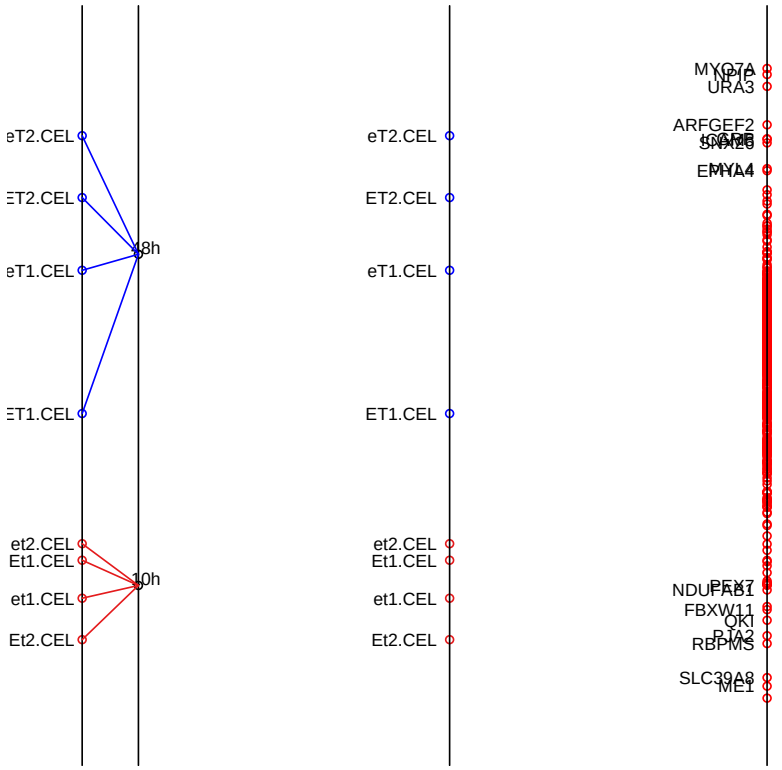

Figure 5: Between group analysis of Estrogen dataset. A. Between.graph of the microarray samples, showing their separation on the discriminating BGA axes, B. graph1D of microarray samples, coloured by their class, C. graph of positions of genes on the same axis. Genes at the ends of the axis are most discriminating for that group

```
> plot(coin, classvec = NCI60$classes[, 2], clab = 0,
+     cpoint = 3)
```

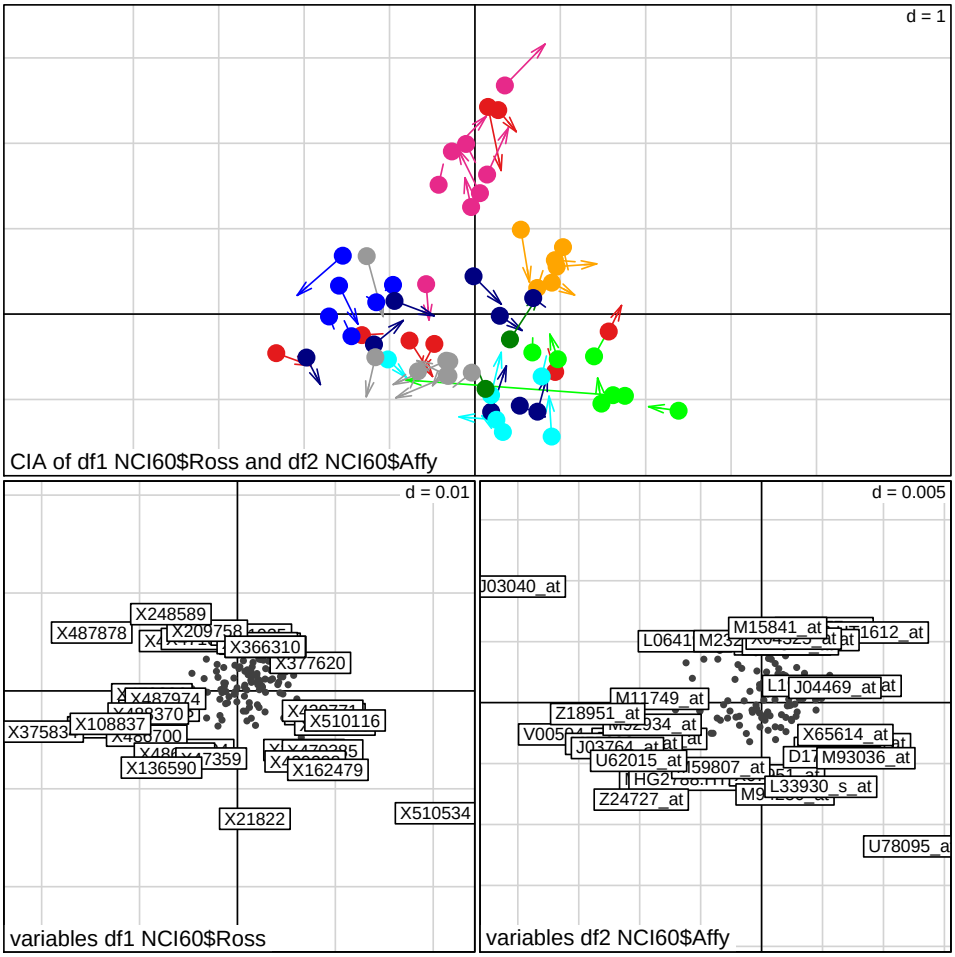

Figure 6: Coinertia analysis of NCI 60 cell line Spotted and Affymetrix gene expression dataset. The same 60 cell lines were analysed by two different labs on a spotted cDNA array (Ross) and an affymetrix array (Affy). The Ross dataset contains 1375 genes, and the affy dataset contains 1517. There is little overlap betwen the genes represented on these platforms. CIA allows visualisation of genes with similar expression patterns across platforms. A) shows a plot of the 60 microarray samples projected onto the one space. The 60 circles represent dataset 1 (Ross) and the 60 arrows represent dataset 2 (affy). Each circle and arrow are joined by a line, the length of which is proportional to the divergence between that samples in the two datasets. The samples are coloured by cell type. B)The gene projections from datasets 1 (Ross), C) the gene projections from dataset 2 (Affy). Genes and samples projected in the same direction from the origin show genes that are expressed in those samples. See vingette for more help on interpreting these plots.

3 Functions in made4

Data Input

array2ade4	Converts matrix, data.frame, exprSet, marrayRaw microarray gene expression data input data into a data frame suitable for analysis in ADE4. The rows and columns are expected to contain the variables (genes) and cases (array samples)
overview	Draw boxplot, histogram and hierarchical tree of gene expression data. This is useful only for a *brief first glance* at data.

Example datasets provides with made4

khan	Microarray gene expression dataset from Khan et al., 2001
NCI60	Microarray gene expression profiles of the NCI 60 cell lines

Classification and class prediction using Between Group Analysis

bga	Between group analysis
bga.jackknife	Jackknife between group analysis
randomiser	Randomly reassign training and test samples
bga.suppl	Between group analysis with supplementary data projection
suppl	Projection of supplementary data onto axes from a between group analysis
plot.bga	Plot results of between group analysis
between.graph	Plot 1D graph of results from between group analysis

Meta analysis of two or more datasets using Coinertia Analysis

cia	Coinertia analysis: Explore the covariance between two datasets
plot.cia	Plot results of coinertia analysis

Graphical Visualisation of results: 1D Visualisation

graph1D	Plot 1D graph of axis from multivariate analysis
between.graph	Plot 1D graph of results from between group analysis
commonMap	Highlight common points between two 1D plots
heatplot	Draws heatmap with dendrograms (of eigenvalues)

Graphical Visualisation of results: 2D Visualisation

plotgenes	Graph xy plot of variable (gene) projections from PCA or COA. Only label variables at ends of axes
s.var	Graph xy plot of variables (genes or arrays). Derived from ADE4 graphics module s.label.
s.groups	Graph xy plot of groups of variables (genes or arrays) and colour by group. Derived from ADE4 graphics module s.class
s.match.col	Graph xy plot of 2 sets of variables (normally genes) from CIA. Derived from ADE4 graphics module s.match
plot.bga	Plot results of between group analysis using plotgenes, s.groups and s.var
plot.cia	Plot results of coinertia analysis showing s.match.col, and plotgenes

Graphical Visualisation of results: 3D Visualisation

do3d	Generate a 3D xyz graph using scatterplot3d
rotate3d	Generate multiple 3D graphs using do3d in which each graph is rotated
html3D	Produce web page with a 3D graph that can be viewed using Chime web browser plug-in, and/or a pdb file that can be viewed using Rasmol

Interpretation of results

topgenes	Returns a list of variables at the ends (positive, negative or both) of an axis
sumstats	Summary statistics on xy co-ordinates, returns the slopes and distance from origin of each co-ordinate
comparelists	Return the intersect, difference and union between 2 vectors
print.comparelists	Prints the results of comparelists

References

[1] Thioulouse,J., Chessel,D., Dolédec,S., and Olivier,J.M ADE-4: a multivariate analysis and graphical display software. *Stat. Comput.*, **7**, 75-83. 1997.

[2] Culhane, A.C., Perriere, G., Considine, E.C., Cotter, T.G., and Higgins, D.G. Between-group analysis of microarray data. *Bioinformatics* **18**: 1600-1608. 2002.

[3] Dolédec, S., and Chessel, D. Rhythmes saisonniers et composantes stationelles en milieu aquatique I- Description d'un plan d'observations complet par projection de variables. *Acta Oecologica Oecologica Generalis* **8**: 403-426.1987.

[4] Dolédec, S., and Chessel, D. Co-inertia analysis: an alternative method for studying species-environment relationships. *Freshwater Biology* **31**: 277-294. 1994.

[5] Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci U S A* **95**: 14863-14868. 1998.

[6] Fellenberg, K., Hauser, N.C., Brors, B., Neutzner, A., Hoheisel, J.D., and Vingron, M. Correspondence analysis applied to microarray data. *Proc Natl Acad Sci U S A* **98**: 10781-10786. 2001.

[7] Irizarry, R. A., Bolstad, B. M.,Collin, F., Cope, L. M., Hobbs, B., Speed, T. P Summaries of Affymetrix GeneChip probe level data *Nucleic Acids Res* **31**:4 15. 2003.

[8] Culhane AC, et al., Cross platform comparison and visualisation of gene expression data using co-inertia analysis. *BMC Bioinformatics*.**4**:59. 2003.