Introduction to Multivariate Analysis of Microarray Gene Expression Data using MADE4

Aedín Culhane

October 4, 2007

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*, *ExpressionSet*, *marrayRaw*, and `data.frame` or `matrix`.

2 New in Current Release (version 1.12)

In *made4* release version 1.12, the functions `plotarrays`, `plotgenes` and `heatplot` have been improved. Each will accept a classvector and will automatically color samples by group.

There is a new function `pretty.dend` which is provides a nice visualization of results from a hierarchical cluster analysis, with a color-bar beneath the dendrogram representing different sample covariates.

3 Installation

made4 requires the package *ade4*. *made4* also calls *scatterplot3d*. These should be installed automatically when you install *made4*. To install *made4*, start R, and source biocLite from bioconductor

```
source("http://www.bioconductor.org/biocLite.R")
biocLite("made4")
```

3.1 Further help

This document provides an overview of *made4* functions.Further examples are described in detail in the RNews newletter, December 2006:

Culhane AC and Thioulouse J. (2006) A multivariate approach to integrating datasets using made4 and ade4.*R News*, **6(5)** 54-58. `http://cran.r-project.org/doc/Rnews/Rnews_2006-5.pdf`

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`. (although it helps to speak French)

We also have a tutorial on ordination using *made4* available online at `http://compbio.dfci.harvard.edu/courses/bioconductor/`.

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`.

3.2 Citing

We are delighted if you use this package. Please do email us if you find a bug or have a suggestion. We would be very grateful if you could cite:

Culhane AC, Thioulouse J, Perriere G, Higgins DG.(2005) MADE4: an R package for multivariate analysis of gene expression data. *Bioinformatics* **21(11):**2789-90.

4 Quickstart

In this vignette, we will demonstrate some of the functions in *made4*. To do this we will use a small dataset that is available in *made4*. This dataset `Khan` contains gene expression profiles of four types of small round blue cell tumours of childhood (SRBCT) published by Khan et al. (2001).

Note the data in `khan` contains gene expression levels for ONLY 306 genes for 64 patient samples. This is a subset of the published dataset. Load the necessary R packages and dataset.

```
> library(made4)
> library(ade4)
```

```
> data(khan)
```

This experiment studied gene expression in patient with four types of SRBCT. These were neuroblastoma (NB), rhabdomyosarcoma (RMS), Burkitt lymphoma, a subset of non-Hodgkin lymphoma (BL), and the Ewing family of tumours (EWS). Gene expression profiles from both tumour biopsy and cell line samples were obtained and are contained in this dataset. In this study data were divided into a training set of 64 samples, and a blind test dataset. These 2 dataset are called khan$train and khan$test. Have a look at the data. For this example we will just examine the training dataset.

```
> names(khan)
```

```
[1] "train"                "test"
[3] "train.classes"        "test.classes"
[5] "annotation"           "gene.labels.imagesID"
```

```
> k.data <- khan$train
> k.class <- khan$train.classes
```

5 Exploratory Data Analysis, Overview, Cluster Analysis

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 of the distance measures 1-Pearson correlation similarity (5) This gives a quick first glance at the data.

```
> overview(k.data)
```

Often its useful to label the samples using a class vector or covariate of interest, in this case, the tumour type (EWS, BL, NB or RMS).

```
> overview(k.data, labels = k.class)
```

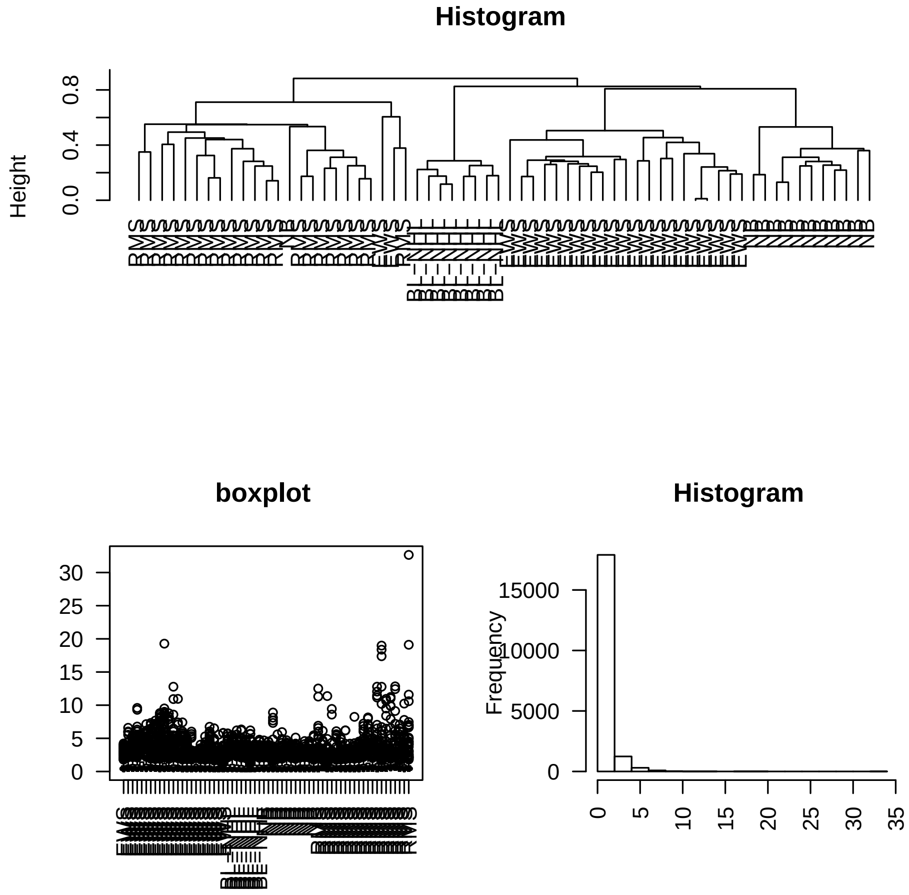

Figure 1: Overview of Khan data. A) dendrogram showing results of average linkage clustering, B) boxplot and C) histrogram.

Often one will known classes in the data (eg Normal v Treatment, or different tumor types). We can insert a class colourbar under the dendrogram, and colour the boxplot.

```
> overview(k.data, classvec = k.class, labels = k.class)
```

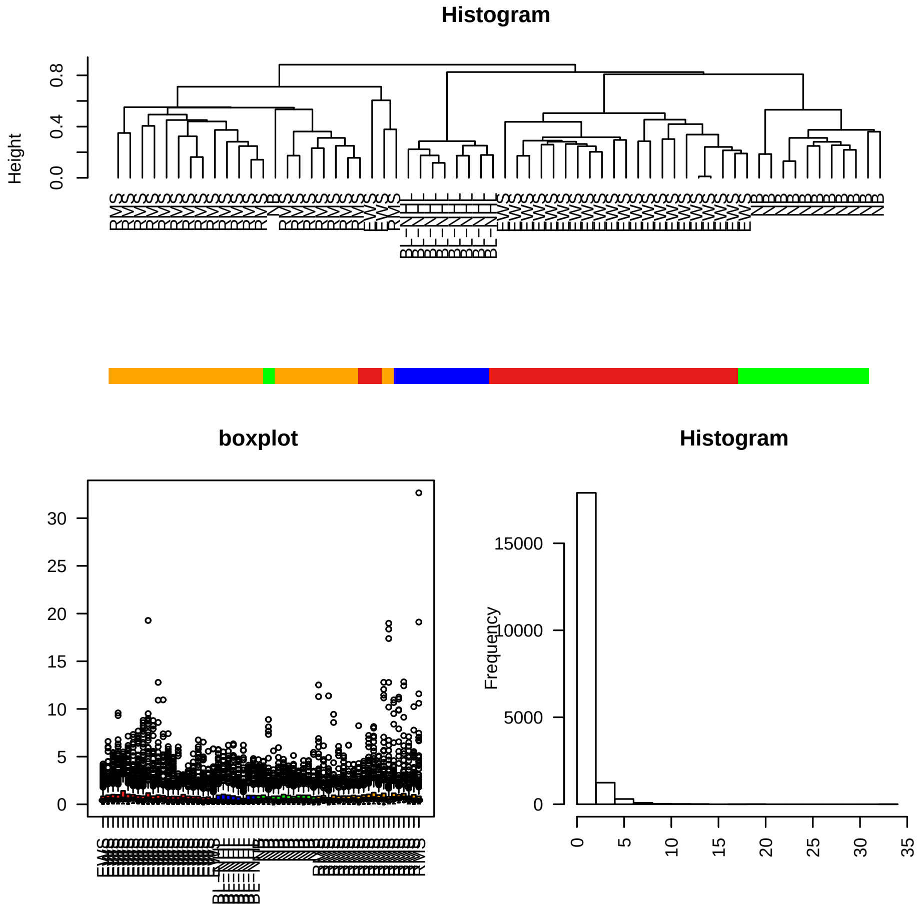

Figure 2: Overview of Khan data. A) dendrogram showing results of average linkage clustering, B) boxplot and C) histrogram. In this case we have added a vector of class (classvec) to color the overview by class membership

5.1 Pretty Dendrogram

A new function was introduced in *made4* version 1.12 called `pretty.dend`. This places color bars beneath a dendrogram from a hierarchical cluster analysis.

Create a data.frame containing factors or character vector which defines different co-variates about the samples (or genes). For example create a factor of khan samples type (cell lines,tissue).

```
> cellType = sapply((strsplit(colnames(khan$train),
+     "\\.")), function(x) substr(x[[2]], 1, 1))
> khanAnnot = cbind(class = as.character(khan$train.classes),
+     cellType = cellType)
> print(khanAnnot[1:2, ])
```

```
     class cellType
[1,] "EWS" "T"
[2,] "EWS" "T"
```

Also see the section on `heatplot`. It can be used to visualize the results of a hierarchical cluster analysis.

```
> pretty.dend(k.data, classvec = khanAnnot, covars = c(1,
+     2), labels = khan$train.classes, title = "Dendrogram using pretty.dend")
```

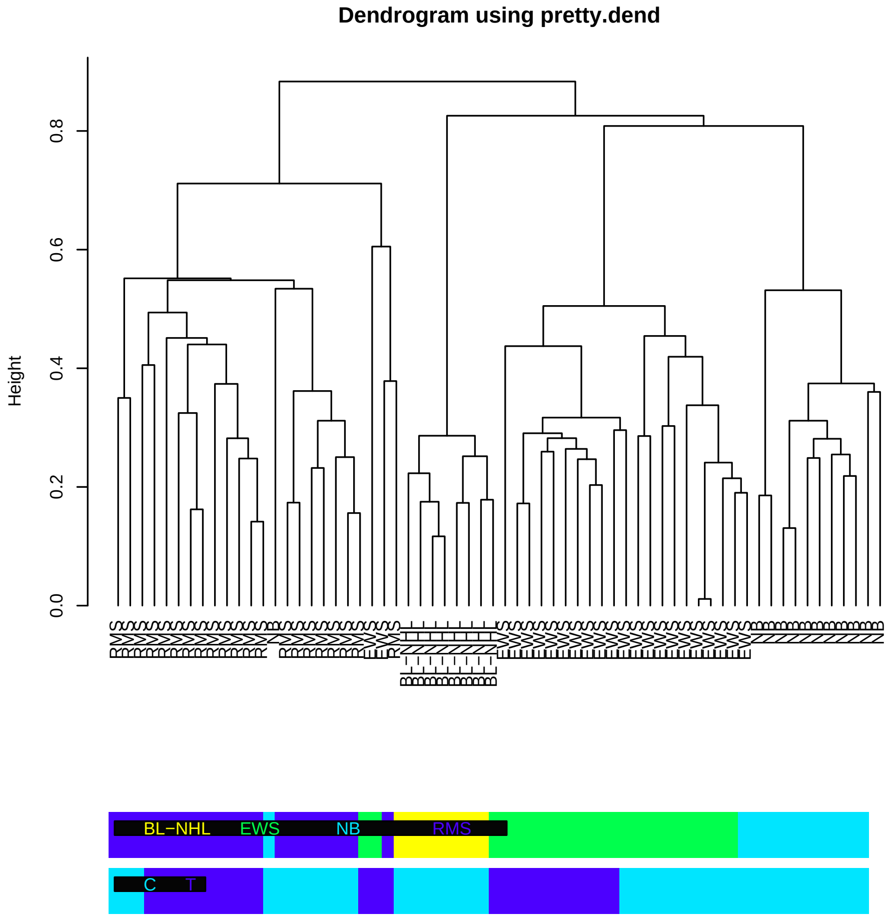

Figure 3: Dendrogram of hierarchical cluster analysis of Khan data plotted using function pretty.dend. The cluster analysis joins pairs of arrays based on their 1- Pearson correlation distance using average linkage clustering. Covariates associated with the arrays are shown in color bar beneath the dendrogram

6 Ordination and 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.

```
> k.coa <- ord(k.data, 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 (k.coa$ord) contain a list of results (of length 12) which includes the eigenvalues ($eig) and the new column coordinates ($co) and the row (line) coordinatein $li. Hence we can visualise the projected coordinations of the genes ($li, 306 genes) and array samples ($co, 64 microarray samples).

```
> names(k.coa)

[1] "ord" "fac"

> k.coa$ord

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

$nf: 63 axis-components saved
$rank: 63
eigen values: 0.1713 0.1383 0.1032 0.05995 0.04965 ...
  vector length mode    content
1 $cw    64     numeric column weights
2 $lw    306    numeric row weights
3 $eig   63     numeric eigen values

  data.frame nrow ncol content
1 $tab       306  64   modified array
2 $li        306  63   row coordinates
3 $l1        306  63   row normed scores
4 $co        64   63   column coordinates
5 $c1        64   63   column normed scores
other elements: N
```

7 Visualising ordination 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(k.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 a factor or character vector, `classvec khan$train.classes` which is saved as `k.class`.

```
> k.class[1:4]

[1] EWS EWS EWS EWS
Levels: EWS BL-NHL NB RMS

> table(k.class)

k.class
   EWS BL-NHL     NB    RMS 
    23      8     12     21 
```

```
> plot(k.coa, classvec = k.class, arraycol = c("red",
+     "blue", "yellow", "green"), genecol = "grey3")
```

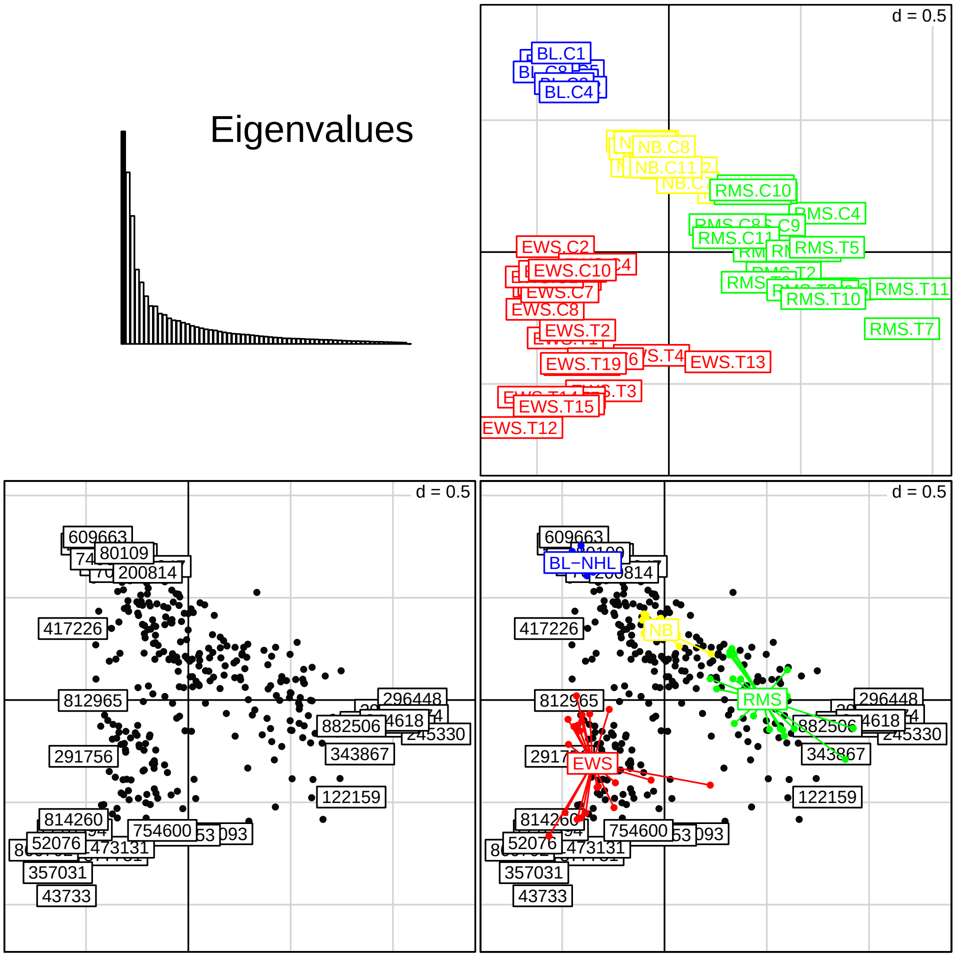

Figure 4: Correspondence analysis of Khan dataset. A. plot of the eigenvalues, B. projection of microarray samples from patient with tumour types EWS (red), BL(blue), NB (yellow)or RMS (green), C. projection of genes (gray filled circles) and D. biplot showing both genes and samples. Samples and genes with a strong associated are projected in the same direction from the origin. The greater the distance from the origin the stronger the association

7.1 Plots: plotarrays, plotgenes

Genes and array projections can also be plotted using `plotgenes` and `plotarrays`.

```
> plotgenes(k.coa)
> plotarrays(k.coa)
```

If you give the function `plotarrays` a class vector (classvec), it colors the arrays groups, defaulting to a ”groups” plot. To plot microarray samples,and colour by group (tumour type) as specified by khan$train.classes

```
> plotarrays(k.coa, classvec = k.class)
```

There are many graphs that (plotarrays) can plot, these include ”groups”, ”simple”, ”labels”.

The default plot is ”labels” if no classvec is given, but ”groups” if a classvec is provided.

”simple” plots a simple plot of the points without labels.

The different principal components of an ordination can be specified using axis1 and axis2.

```
> par(mfrow = c(3, 2))
> plotarrays(k.coa)

[1] "Need to specify groups"

> plotarrays(k.coa, graph = "labels")
> plotarrays(k.coa, classvec = k.class)
> plotarrays(k.coa, graph = "groups", classvec = k.class,
+     axis2 = 3)
> plotarrays(k.coa, graph = "simple", classvec = k.class,
+     axis2 = 3)
> plotarrays(k.coa, graph = "labels", classvec = k.class,
+     axis1 = 2, axis2 = 3)
```

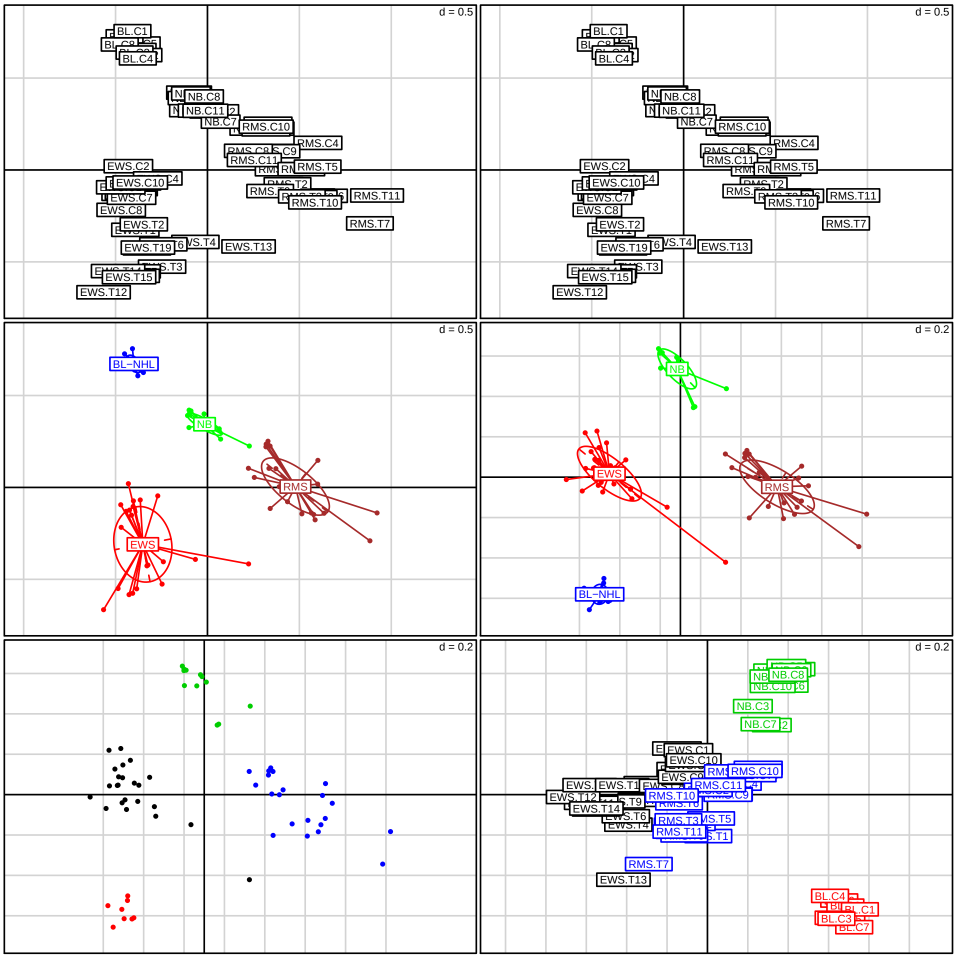

Figure 5: Visualization of arrays from a correspondence analysis of Khan dataset

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.

To plot gene projections without any labels set n=0.

Typically there are a large number of genes, thus it is not feasible to label all of these.

```
> plotgenes(k.coa, 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 your own labels. For example its often useful to labels using HUGO gene symbols. We find the Bioconductor *annotate* and *annaffy* annotation package s are very useful for this. Alternatively we also use *biomaRt* or *Resourcerer* or the Stanford Source database.

In this example we provide annotation from the Source database in khan$annotation. The gene symbol are in the column khan$annotation$Symbol

```
> gene.symbs <- khan$annotation$Symbol
> plotgenes(k.coa, n = 10, col = "red", genelabels = gene.symbs)
```

7.2 Creating a heatmap using the function heatplot

Sometime its useful to get a feel for the group separation on each of the principal components. The function (heatplot) is useful for this. The sample weight are k.coa$ord$co (co, columns) and the gene weight are k.coa$ord$li (li, lines or rows)

From this plot it can be seen, that the first component separates the RMS from othe samples. The RMS samples (brown on color bar) have positive weight on the first component whereas the remaining samples have negative loadings. On the second component, the BL samples (blue color bar), the NB (green color bar) and some of the RMS (brown on color bar) are distinguished from the EWS samples (red on color bar). The 3rd component separated the NB and other samples. The 2D and 3D plots to confirm this. It is clear that all 4 cancer types are clearly distinguished using correspondence analysis.

```
> plotgenes(k.coa, n = 10, col = "red", genelabels = gene.symbs)
```

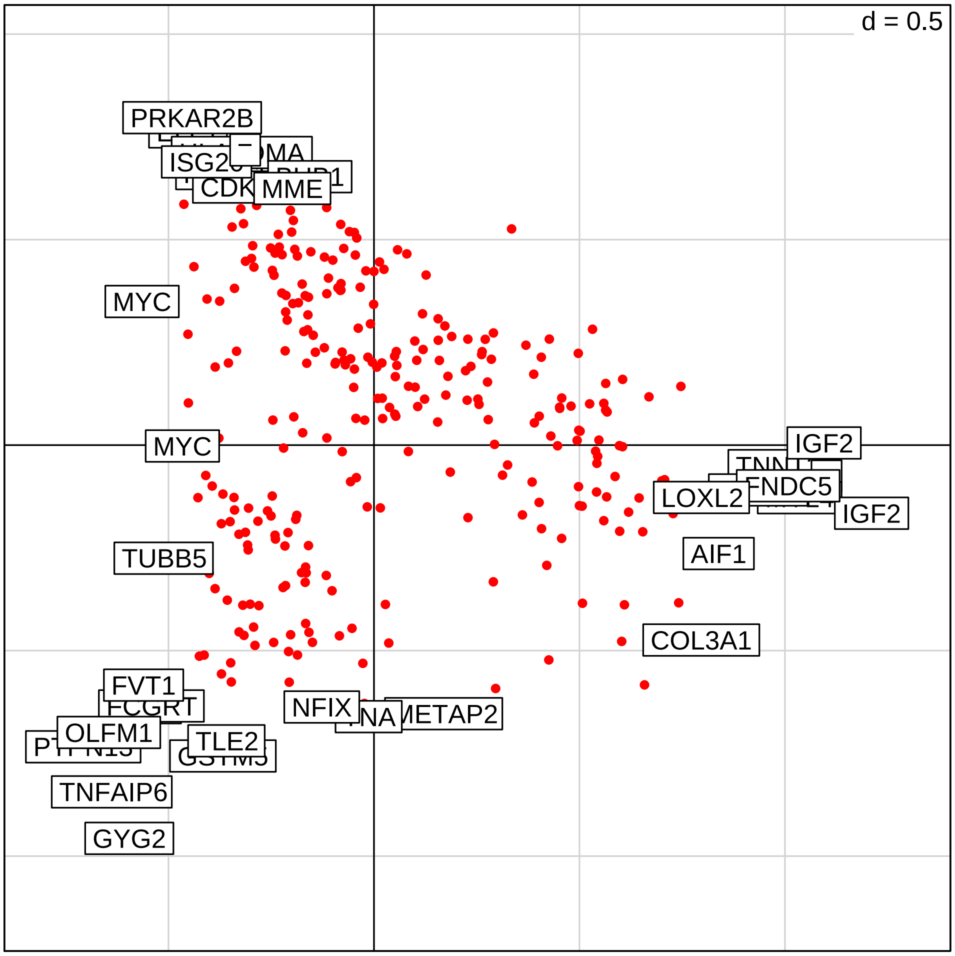

Figure 6: Projection of genes (filled circles) in Correspondence analysis of Khan dataset. The genes at the ends of each of the axes are labelled with HUGO gene symbols.

```
> heatplot(k.coa$ord$co[, 1:10], dend = "none",
+     classvec = k.class, scale = "none")
```

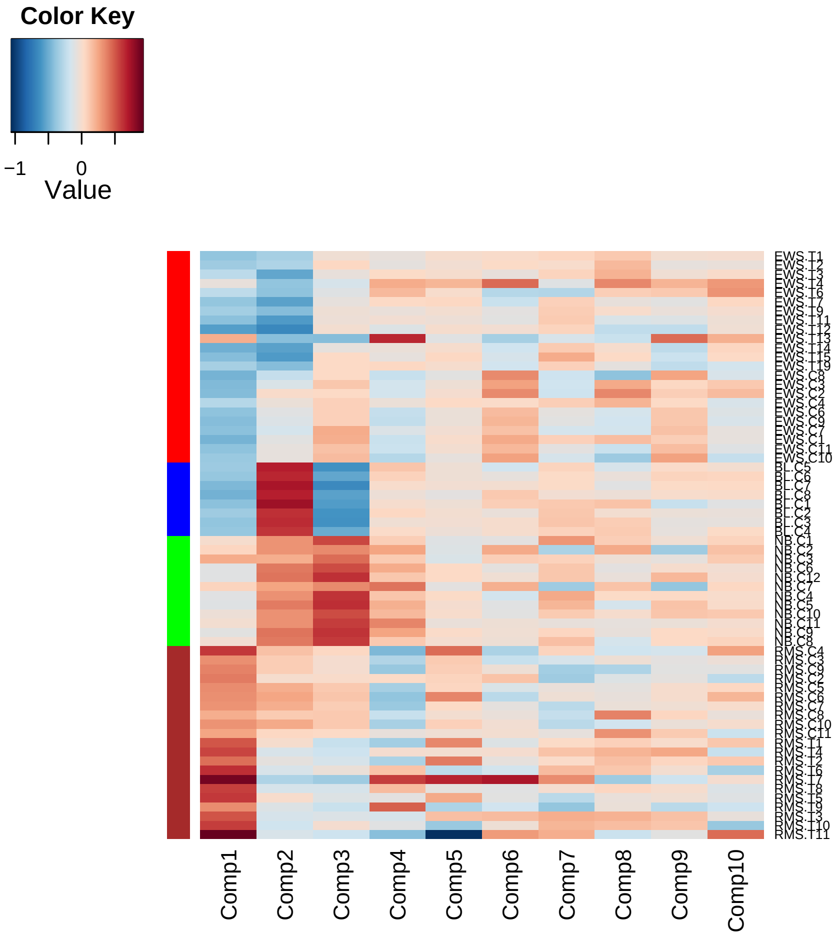

Figure 7: Visualization of weights or loadings on each of the first 10 components in gene space (projection of samples) of a correspondence analysis of Khan dataset

7.3 Plots: further information. Extracting top genes

COA or ordination can also be run where a class vector is given to (ord). In this case subsequent plots will be automatically colored using this class vector

```
> k.coa2 <- ord(k.data, classvec = k.class)
> plot(k.coa2)
```

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(k.coa, axis = 1, n = 5)
```

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

```
> topgenes(k.coa, labels = gene.symbs, end = "neg")
```

```
 [1] "PTPN13"  "OLFM1"   "TNFAIP6" "GYG2"    "CAV1"
 [6] "MYC"     "FVT1"    "FCGRT"   "TUBB5"   "MYC"
```

Two lists can be compares using `comparelists`.

7.4 Plots 3D visualisation and html output

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(k.coa$ord$co, classvec = k.class, cex.symbols = 3)
> html3D(k.coa$ord$co, k.class, writehtml = TRUE)
```

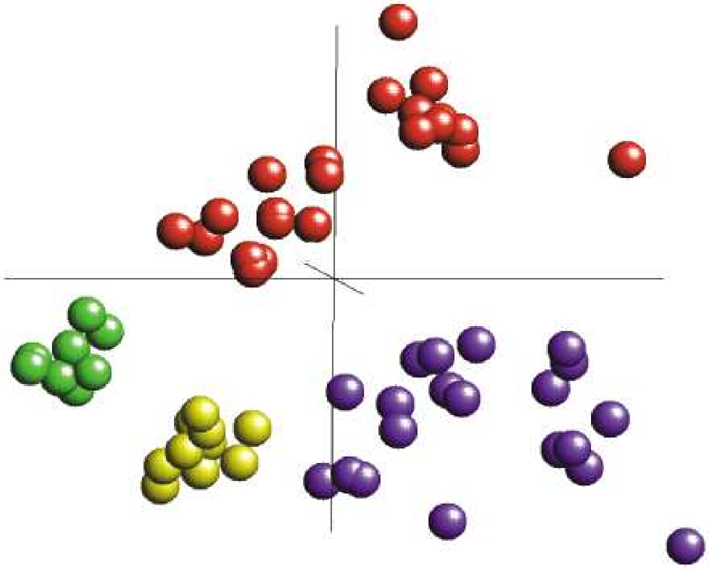

Figure 8: Output from html3D, which can be rotated and visualised on web browsers that can support chime (IE or Netscape on MS Windows or Mac).

8 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.

```
> k.bga <- bga(k.data, type = "coa", classvec = k.class)
```

Sometimes its useful to visualise 1 axes of an analysis. To do this use `graph1D` or `between.graph`. The latter function is specifically for visualising results from a bga as it shows the separation of classes achieved.

```
> between.graph(k.bga, ax = 1)
> plot(k.bga, genelabels = gene.symbs)
```

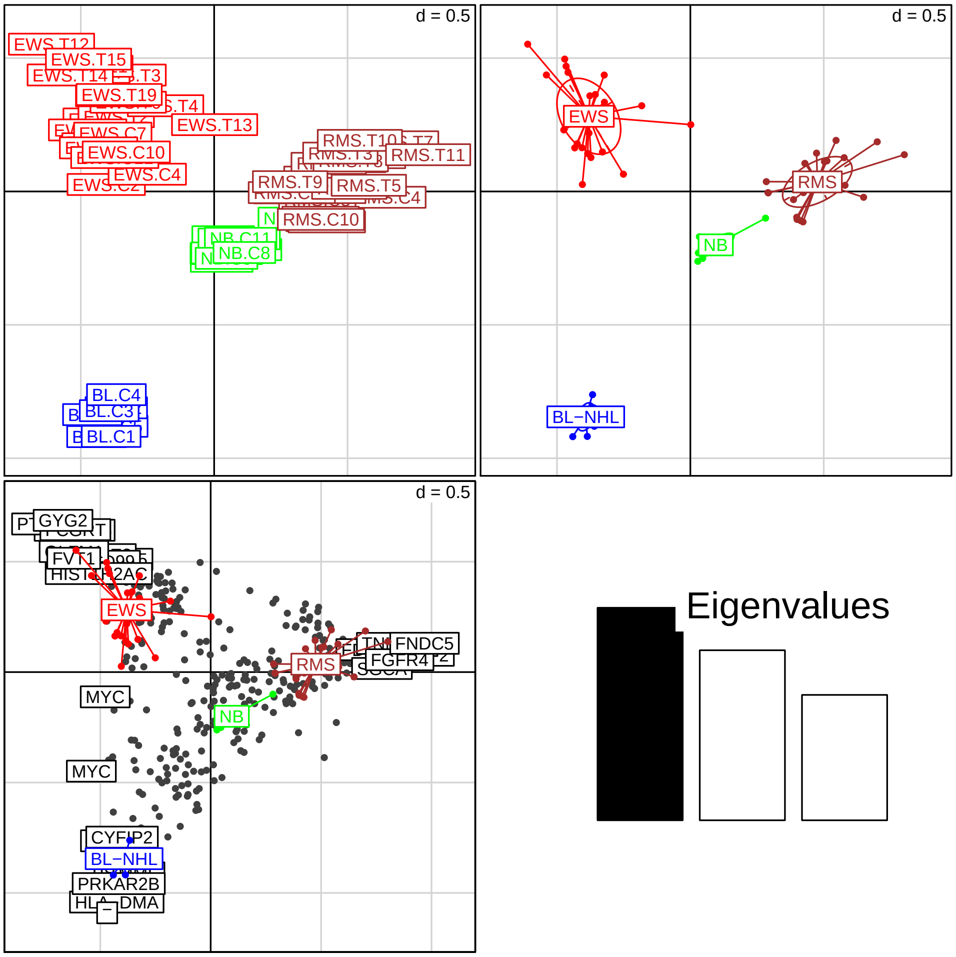

Figure 9: Between group analysis of Khan dataset. A. Between.graph of the microarray samples, showing their separation on the discriminating BGA axes, B. Scatterplot of the first 2 axes of microarray samples, coloured by their class, C. graph of positions of genes on the same axis. Genes at the ends of the axis are most discriminating for that group

9 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.

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

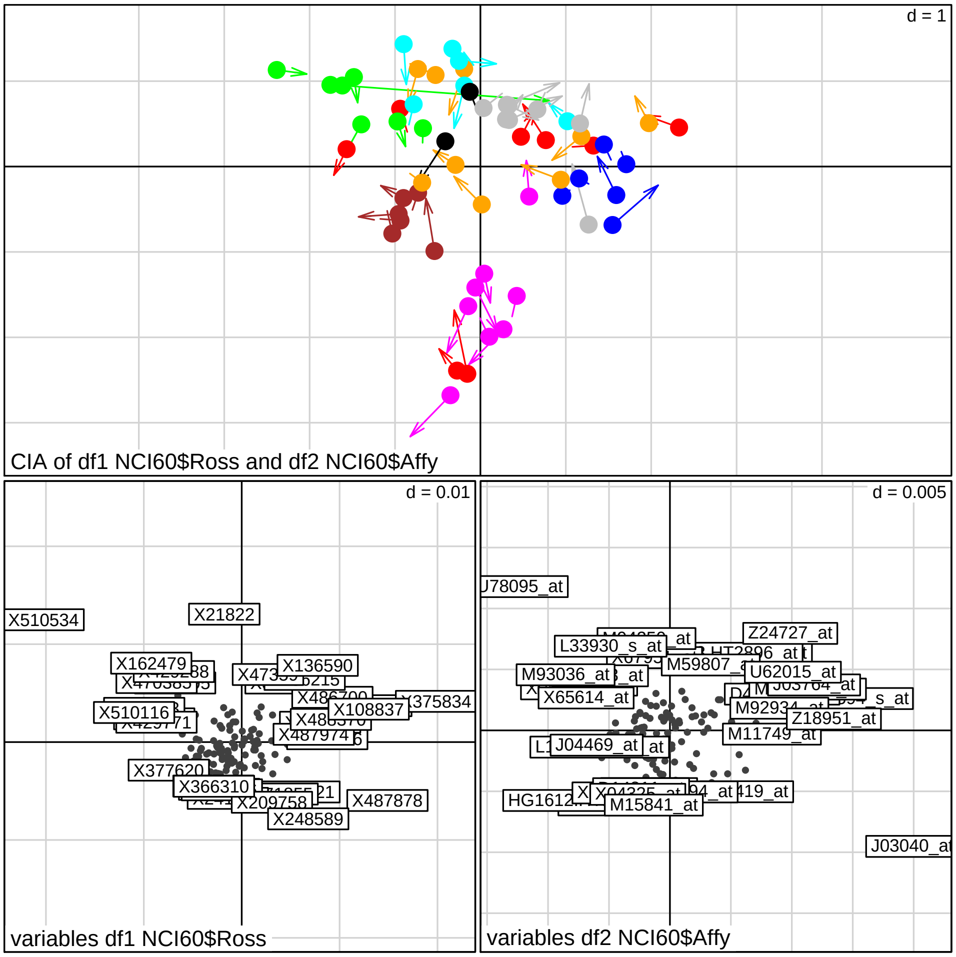

Figure 10: Coinertia analysis of NCI 60 cell line Spotted and Affymetrix gene expression dataset. The same 60 cell lines were analysed by two different labs on a spotted cDNA array (Ross) and an affymetrix array (Affy). The Ross dataset contains 1375 genes, and the affy dataset contains 1517. There is 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.

10 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 or row) projections from PCA or COA. Only label variables at ends of axes
plotarrays	Graph xy plot of the arrays (column) projections from an ordination analysis. There are several graph types, these are "groups", "simple", "labels", "groups2", "coinertia","coinertia2".

Graphical Visualisation of results:Cluster Analysis

overview	Graph of dendrogram from hierarchical cluster analysis (1-Pearson correlation coefficient distance metric, average linkage joining), boxplot and histrogram of data
pretty.dend	Graph of dendrogram from hierarchical cluster analysis (1-Pearson correlation coefficient distance metrix, average linkage joining). If a class vector or many class vectors (covariates) are given in a data.frame or factor,these are represented by a color bar is drawn underneath the dendrogram.

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.