Tutorial in Exploratory Data Analysis of Genomics Data

Aedín Culhane

October 24, 2011

Contents

1 Introduction to the dataset for this tutorial

For the first part of this tutorial we will use a subset of the primate fibroblast gene expression from Karaman et al., Genome Research 2003. This study examines 3 groups,

human, bonobo and gorilla expression profiles on Affymetrix HG_U95Av2 chips (1). This dataset contains 46 chips and is available in the Bioconducor library fibroEset (MAS5.0 data), and the web site `http://hacialab.usc.edu/supplement/karaman_etal_2003/index.html` (raw cel files).

In this tutorial we will look at 9 chips which have been normalised using vsn. For information I have included details of how I normalised these, at the end of the tutorial. Download the normalized gene expression profiles from the web site (or Course wiki). The data are stored as a comma separated file, which is readable by MSExcel.

2 Task 1. Initial data Exploration

As we will be examining Affymetrix data, load the package *affy*. For exploratory analysis and ordination, we will use the package *made4*.

```
> library(affy)
> library(made4)
> library(scatterplot3d)
> library(gplots)
> library(limma)
> library(annaffy)
```

made4 accepts gene expression data is a wide variety of input formats, including Bioconductor formats, *AffyBatch*, *ExpressionSet*, *marrayRaw*, and `data.frame` or `matrix`.

In this case the *vsn* normalised data are provided as a comma separated file. To load in R:

```
> data.vsn <- read.csv("data.vsn.csv", as.is = TRUE,
+     row.names = 1)
> dim(data.vsn)
```

```
[1] 12625     9
```

The package *made4* contains a simple wrapper function, `overview` which will draw a dendrogram of hierarchical cluster analysis (1- Pearson Correlation distance metric, average linkage) of the samples (2), a `boxplot` and histogram showing the distribution of the data.

```
> overview(data.vsn, labels = substring(colnames(data.vsn),
+     1, 5))
```

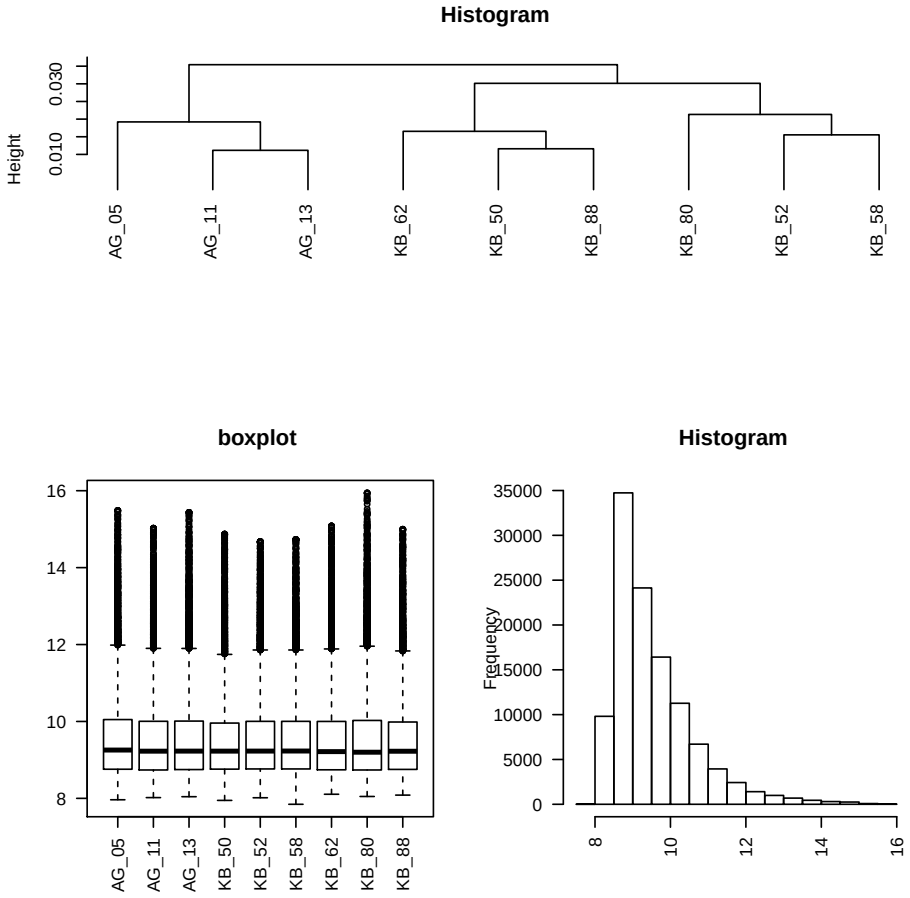


Figure 1: Overview of Fibroblast data. A) dendrogram showing results of average linkage clustering, B) boxplot and C) histrogram. The 9 Samples are labelled with their colnames (array filenames), however substring was used to reduce the length of the colnames for clarity

3 Task 2: Interpretation - labelling with covariates

Overview shows that we have 2 major(possibly 3) groups or clusters within the data. To interpret these exploratory data clustering, sample information is required. Read a text file (tab delimited) with sample information into R. The sample annotations are in the file annt.txt, which is on the course webpage/wiki.

```
> annt <- read.table("annt.txt", header = TRUE)
> annt[1:2, ]
```

```
             Cels short.names Donor Age Gender  DT
1 AG_05414_AS.cel    AG_05414   Hsa  73      M 2.3
2 AG_11745_AS.cel    AG_11745   Hsa  43      F 1.8
  estb.same
1         D
2         D
```

`read.table` reads in a table as a data.frame. The column heading are:

```
> colnames(annt)
```

```
[1] "Cels"        "short.names" "Donor"       "Age"
[5] "Gender"      "DT"          "estb.same"
```

This file contains the cel filenames (Cels), shorter names for the arrays (short.names), information about the Donor (Gorilla, Bonobo, Human), Age (years), Gender (male/female), doubling time (DT) of the cell lines, and information about whether cells where established from the same cell lines (estb.same). To view the data in a column in the `data.frame`, use the $ symbol and the column label. `table` can also be used to tabulate a summary of a categorical vector.

```
> annt$Donor
```

```
[1] Hsa Hsa Hsa Ggo Ppa Ppa Ggo Ppa Ggo
Levels: Ggo Hsa Ppa
```

```
> table(annt$Donor)
```

```
Ggo Hsa Ppa
  3   3   3
```

```
> table(annt$Gender)
```

```
F M
5 4
```

Redraw the overview plot, but add label information about the Donor.

```
> overview(data.vsn, label = annt$Donor, classvec = annt$Donor)
```

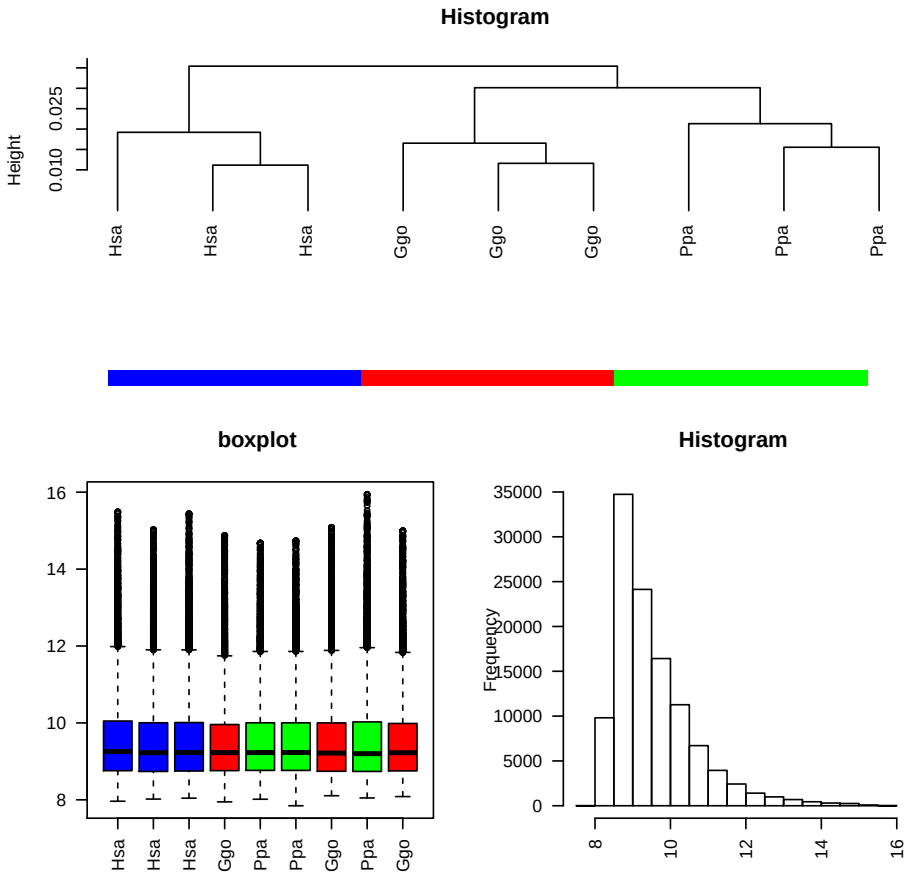


Figure 2: Overview of Fibroblast data. A) dendrogram showing results of average linkage clustering, B) boxplot and C) histrogram. The 9 Samples are labelled by Donor.

This is easier to interpret. It is be seen that humans are cluster distinctly from other primates. But BEFORE we go ahead and search for genes distinguishing these. CHECK the other co variants:

- Is there a confounding co-variate?
- Do the samples also group by Age, Gender, DT, estb.same ?
- What do you think of the experimental design? How could it be improved?
- Have a look at the different plots, What do you think?

4 Task 3: Ordination

4.1 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. To run a correspondence analysis (3) on this dataset.

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

Have a look at data.coa. The ordination results are in $ord. The row, column coordinates are $li and $co respectively. The eigenvalues are in $eig.

```
> data.coa$ord
```

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

$nf: 8 axis-components saved
$rank: 8
eigen values: 0.0001076 7.389e-05 4.039e-05 3.143e-05 2.057e-05 ...
  vector length mode    content
1 $cw    9      numeric column weights
2 $lw    12625  numeric row weights
3 $eig   8      numeric eigen values

  data.frame nrow  ncol content
1 $tab       12625 9    modified array
2 $li        12625 8    row coordinates
3 $l1        12625 8    row normed scores
4 $co        9     8    column coordinates
5 $c1        9     8    column normed scores
other elements: N
```

```
> data.coa$ord$co[1:2, 1:3]
```

```
                      Comp1        Comp2        Comp3
AG_05414_AS.cel -0.01483693 -0.002726216  0.004016784
AG_11745_AS.cel -0.01280013 -0.004886536 -0.001854585
```

In a COA analysis the total $eig will be equivalent to the total chi-sq of the table. To get the % of variance explained by each axis.

```
> data.coa$ord$eig * 100/sum(data.coa$ord$eig)

[1] 33.685712 23.130858 12.645416  9.838039  6.439921
[6]  5.834663  4.662855  3.762536
```

The cumulative variance is given by

```
> cumsum(data.coa$ord$eig * 100/sum(data.coa$ord$eig))

[1]  33.68571  56.81657  69.46199  79.30003  85.73995
[6]  91.57461  96.23746 100.00000
```

Therefore almost 57% of the variance is captured by the first 2 components.

4.2 Correspondence Analysis- Visualization of Results

There are many functions in *made4* for visualizing results from ordination analysis. The simplest way to view results from `ord` is to use the function `plot`. This will draw a plot of the eigenvalues, along with plots of the variables (genes) and a plot of the cases (microarray samples).

```
> plot(data.coa, classvec = annt$Donor)
```

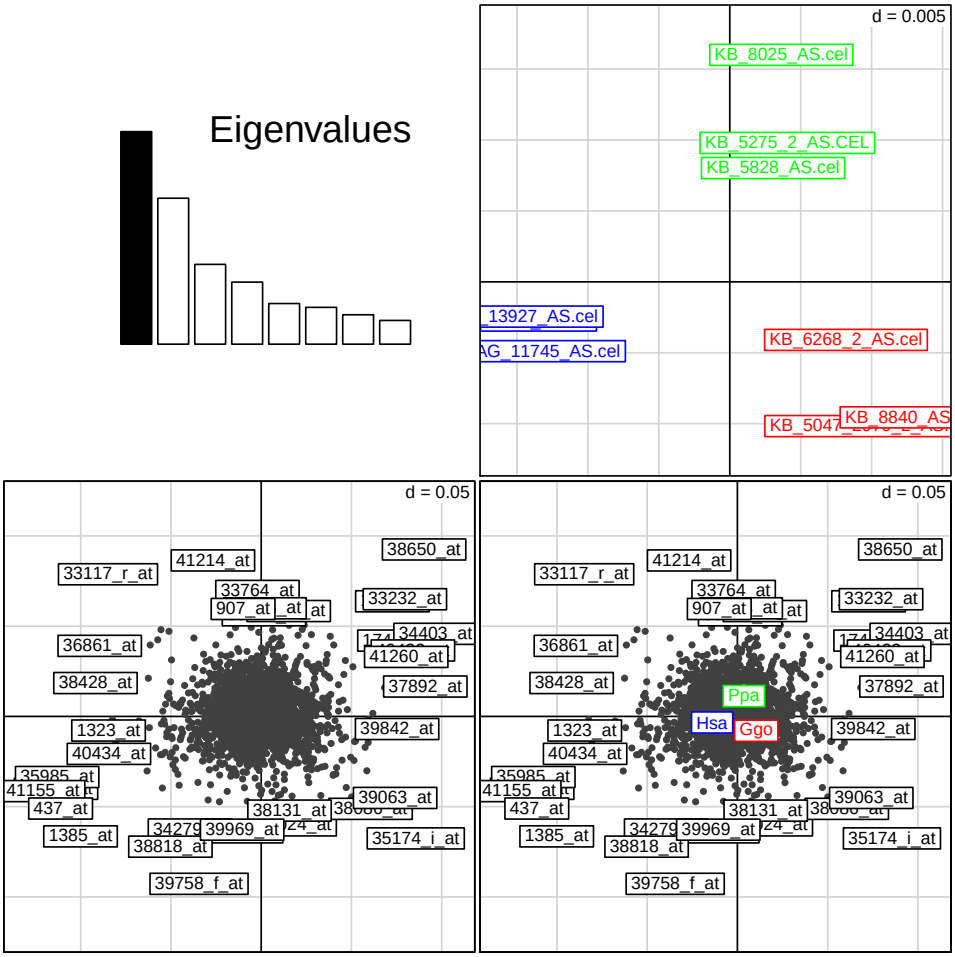


Figure 3: Correspondence analysis plot. A. plot of the eigenvalues, B. projection of microarray samples (colored by Donor) C. projection of genes (gray filled circles) and D. biplot showing both genes and samples. Samples and genes with a strong associated are projected in the same direction from the origin. The greater the distance from the origin the stronger the association

The distinction between species is captured on the first 2 eigenvectors. Principal component 1 (horizontal) defines the human versus the other primates, and PC2 captures the difference between the Bonobo (Ppa) and the primates, human and gorilla.

A heatmap can be used to visualize the weights (or contributions) of genes or arrays to each principal component (or axis).

```
> heatplot(data.coa$ord$co, dend = "none", labRow = annt$Donor)

[1] "Data (original) range:  -0.01 0.02"
[1] "Data (scale) range:  -2.13 1.86"
[1] "Data scaled to range:  -2.13 1.86"
```

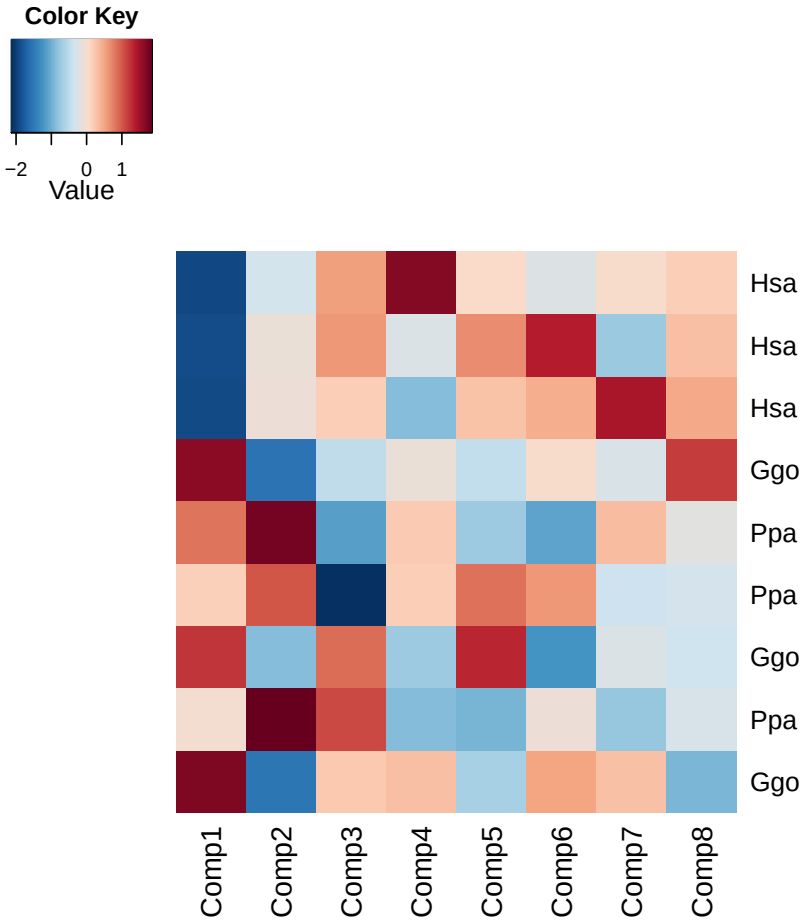


Figure 4: Heatmap of sample loadings in the new projection space. These allow easy visualization of which samples contribute to the variance on each new axes (or principal component).

To plot the arrays projections from the COA.

```
> plotarrays(data.coa$ord$co, classvec = annt$Donor)
> plotarrays(data.coa$ord$co, classvec = annt$Gender)
```

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(data.coa, n = 5, col = "red")
```

Sometimes R may put an X in front of row names if they start with a number. Hence the names in ax1 don't agree with data. If you see this it is easy to remove the "X" in the names,

```
ax1<-sub("X", "", ax1)
```

To extract a list of variables with greatest loadings or weights on an axes, (ie those at the end of an axes), use `topgenes`. For example, to get a list of the 5 genes at the negative and postive ends of axes 1.

```
> ax1 <- topgenes(data.coa, axis = 1, n = 5)
```

To only the a list of the genes at the positive end of the first axes

```
> genes.ax1 <- topgenes(data.coa, end = "pos", n = 5)
> genes.ax1

[1] "34403_at"   "37892_at"   "38650_at"   "35174_i_at"
[5] "40422_at"
```

Two lists can be compared using `comparelists`.

It is useful to use boxplots to visualize the gene expression distributions of a gene in different sample groups. The distrbibutions will be plotted using the order of levels(factor). In this example the order of annt$Donor is Ggo, Hsa, Ppa. It would be more useful to plot Hsa, Ggo and then Ppa. Therefore reorder the levels of the factor.

```
> annt$Donor

[1] Hsa Hsa Hsa Ggo Ppa Ppa Ggo Ppa Ggo
Levels: Ggo Hsa Ppa

> spec = factor(annt$Donor, levels = c("Hsa", "Ggo",
+     "Ppa"))
```

```
> par(mfrow = c(2, 1))
> gene.pos1 <- topgenes(data.coa, end = "pos", n = 1)
> df.PosGenes <- t(data.vsn[gene.pos1, ])
> boxplot(df.PosGenes ~ spec, col = getcol(3), main = paste(gene.pos1,
+     "has greatest loading on positive end of ax1"))
> gene.neg1 <- topgenes(data.coa, end = "neg", n = 1)
> df.NegGenes <- t(data.vsn[gene.neg1, ])
> boxplot(df.NegGenes ~ spec, col = getcol(3), main = paste(gene.neg1,
+     "has greatest loading on negative end of ax1"))
```

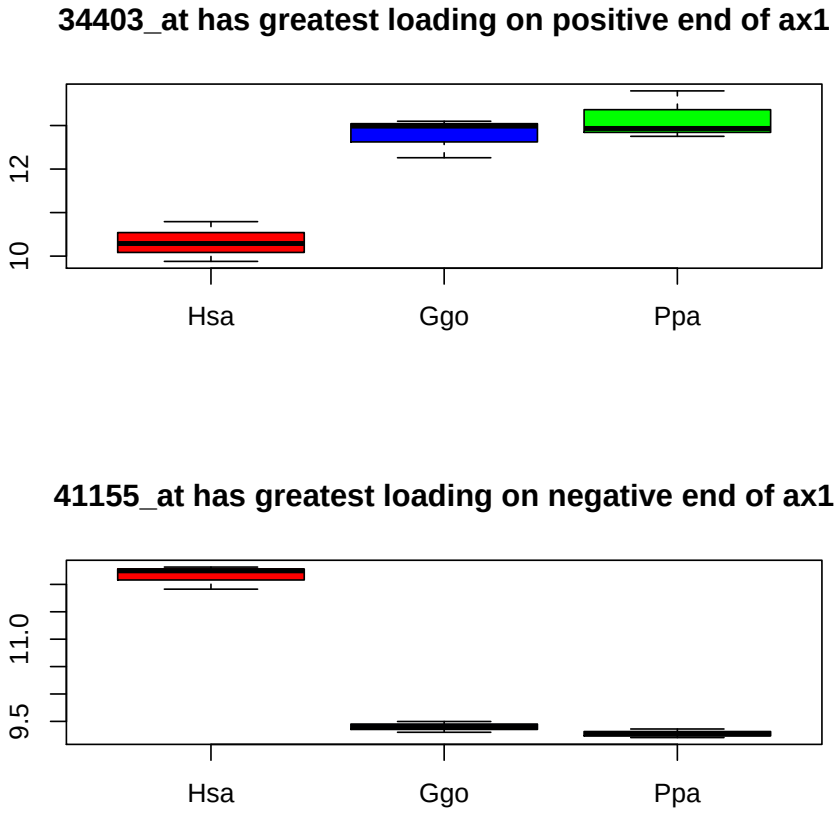


Figure 5: Heatmap of gene expression profiles of probesets with greatest loadings on the positive and negative ends of axes 1

Make a heatmap and perform a cluster analysis of gene expression profiles of the 10 genes with highest weights (neg and pos) on axis 1. In Fig 6 we see, while the human versus non-human primates difference is captured, the difference between the non-human primates is not well defined by axis 1.

```
> gene.pos.neg <- topgenes(data.coa, end = "both",
+     n = 5)
> heatplot(data.vsn[gene.pos.neg, ], labCol = as.character(annt$Donor))
```

```
[1] "Data (original) range:  8.62 14.53"
[1] "Data (scale) range:  -1.73 1.8"
[1] "Data scaled to range:  -1.73 1.8"
```

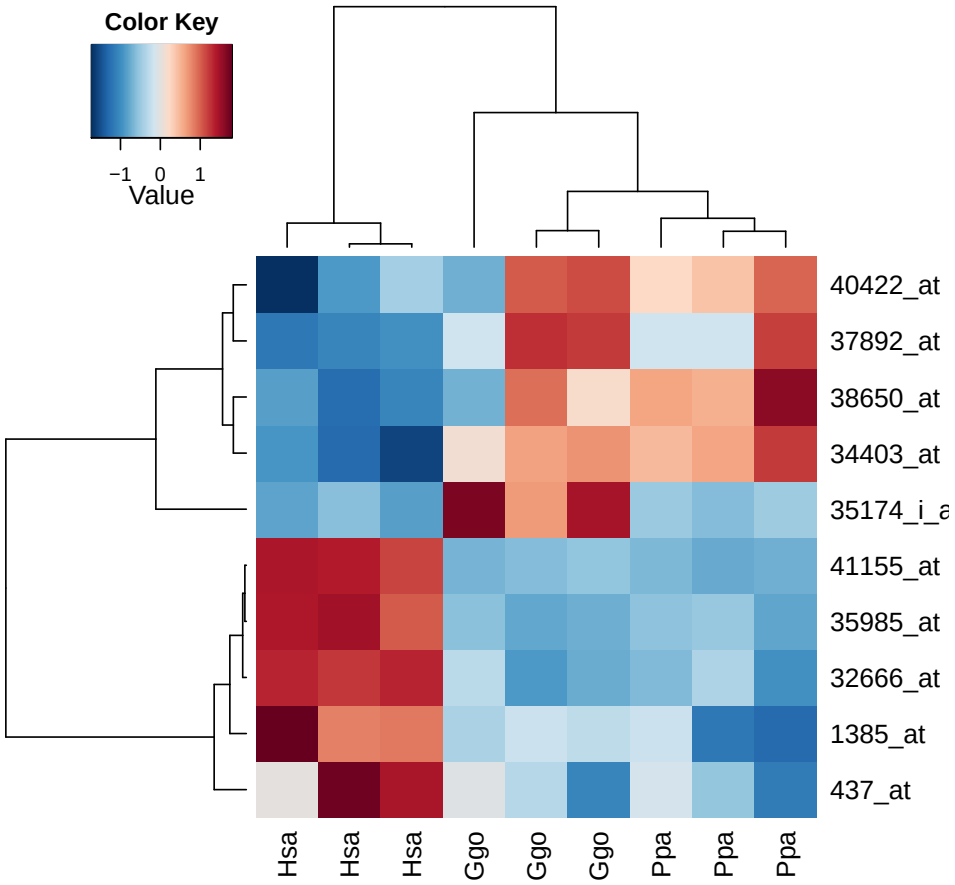


Figure 6: Heatmap of gene expression profiles of genes with greatest loadings on the negative end of axes 1

There are several ways to save an active plot or a plot you have just drawn, for

example look at `dev.copy`. MSwindows users can also use the function (savePlot)

```
savePlot("heatplot_COA")
```

4.3 PCA

We have run a Correspondence Analysis, Compare these results to a PCA

```
> data.pca <- ord(data.vsn, type = "pca")
> data.pca$ord

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

$nf: 8 axis-components saved
$rank: 9
eigen values: 8.764 0.08128 0.05529 0.02619 0.02274 ...
  vector length mode    content
1 $cw    9      numeric column weights
2 $lw    12625  numeric row weights
3 $eig   9      numeric eigen values

  data.frame nrow  ncol content
1 $tab       12625 9    modified array
2 $li        12625 8    row coordinates
3 $l1        12625 8    row normed scores
4 $co        9     8    column coordinates
5 $c1        9     8    column normed scores
other elements: cent norm
```

- Compare the difference between the results from PCA and COA.
- How much variance is capture by each approach?
- Examine and compare plots from PCA and COA?
- In the PCA plots, do arrays segregate by Donor, Age or Gender?

```
> plotarrays(data.pca$ord$co, classvec = annt$Donor)
> plotgenes(data.pca)
```

At this stage.. we need to get gene information in order to fully interpret our exploratory data analysis

5 Task 4: Annotating the plots with gene information

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. Plots can be easily re-labeled. It is often useful to labels genes with their HUGO gene symbols. We find the Bioconductor *annotate* and *annaffy* annotation packages are very useful for this. Alternatively we also use *biomaRt* or *Resourcerer* or the Stanford Source database.

For this practical we will use *annaffy*, to get the Gene Symbol for all genes. We can then used these in plots

```
> library(annaffy)
```

To get a list of the Unigene, LocusLink or descriptors for these genes, we can use the following. Remember help on annaffy can always be assessed by using ? and the command name or opening help in a web browser by typing `help.start()`.

```
> affy.id <- rownames(data.vsn)
> aafUniGene(affy.id[1:10], "hgu95av2.db")
> aafLocusLink(affy.id[1:10], "hgu95av2.db")
> aafDescription(affy.id[1:10], "hgu95av2.db")
```

These commands return a *list*, but to make these into a character vector use the function `getText`

```
> getText(aafLocusLink(affy.id[1:10], "hgu95av2.db"))
```

```
 [1] "5875" "5595" "7075" "1557" "643"  "643"  "1843" "4319"
 [9] "780"  "5610"
```

Get the list of all offical (HUGO) gene symbols and re-plot the COA results.

```
> affy.id <- rownames(data.vsn)
> affy.symbols <- aafSymbol(affy.id, "hgu95av2.db")
> affy.symbols <- getText(affy.symbols)
> plotgenes(data.coa, genelabels = affy.symbols,
+     col = "red", n = 10)
```

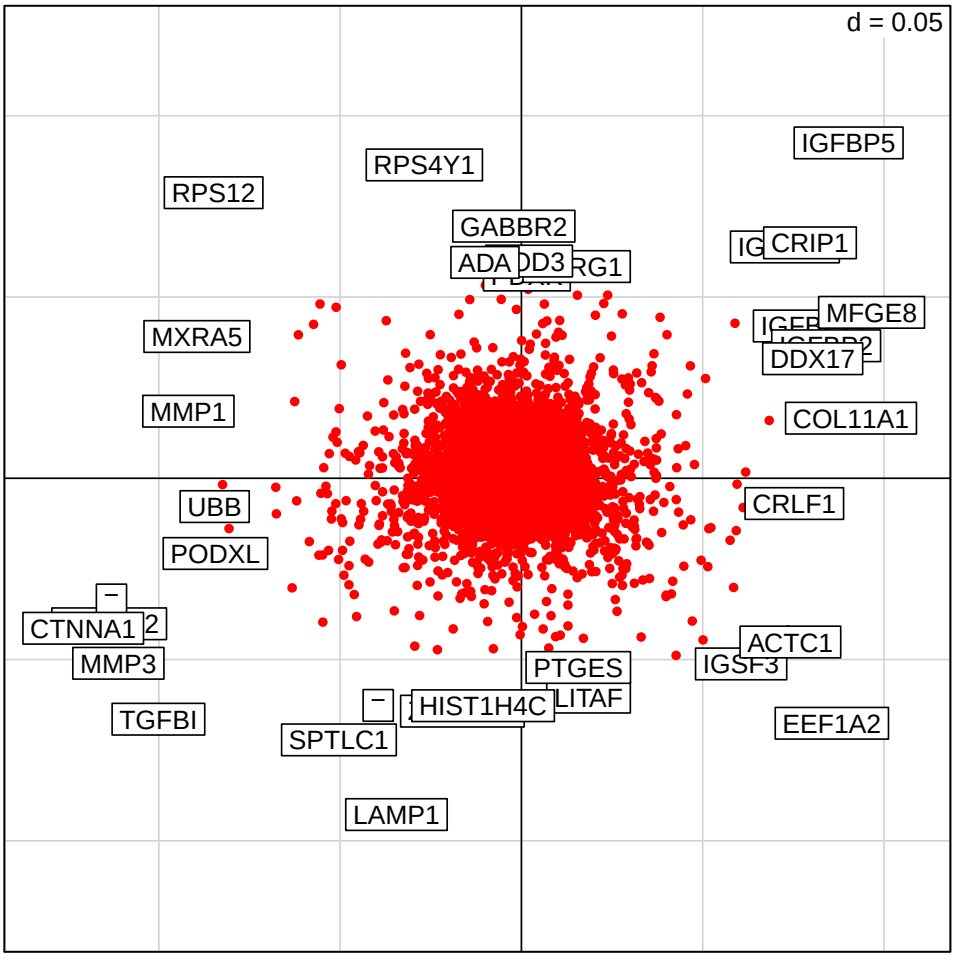


Figure 7: Projection of genes (filled circles) in Correspondence analysis. The genes at the ends of each of the axes are labelled with HUGO gene symbols.

- Get the gene symbols for the `topgenes` from the first axes which were highly expressed in human but not the other primates.
- Which genes are highly expressed in each of the other primates.
- Are any of these genes also expressed in males or females (gender)?
- Redraw the heatmap but add gene symbols.

```
> topgenes(data.coa, labels = affy.symbols, end = "neg",
+     n = 5)

[1] "CTNNA1" "CXCL12" "-"      "MMP3"   "TGFBI"
```

To obtain a browsable html table of gene annotation:

```
> anncols <- aaf.handler()
> anncols
> anntable <- aafTableAnn(ax1, "hgu95av2.db", anncols)
> saveHTML(anntable, "example1.html", title = "Example")
```

Have a look at copy of this output.

- Which genes are part of the apoptosis pathway?
- How many genes are found on Chromosome 2?
- How many publications are there in PubMed on IGFBP2?

6 Advanced Tasks

If you have time, there are extra tasks. The code may also be useful to you in your own data analysis.

6.1 Advanced Task 1: 3D Plots

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(data.coa$ord$co, classvec = annt$Donor, cex.symbols = 3)
> rotate3d(data.coa$ord$co, classvec = annt$Donor)
> html3D(data.coa$ord$co, classvec = annt$Donor,
+     writehtml = TRUE)
```

html3D produces a plot which can be rotated using chime or jmol. For an example see the course website.

6.2 Advanced Task 2: Use limma to select genes and examine these using ordination and clustering

Several feature selection methods are available. We provide a empirical comparision of these in our paper (4), in which we recommend *limma* or Rank Products (available in the package *RankProd* for feature selection.

To perform a feature selection using limma, first generate a class vector, with 2 classes (eg human v other primate).

```
> modelDonor <- model.matrix(~Donor, annt)
> lm.out <- lmFit(data.vsn, modelDonor)
> lm.out <- eBayes(lm.out)
> geneHsa <- topTable(lm.out, coef = 2)
> geneHsa
```

```
              ID      logFC   AveExpr         t
11262   41155_at  2.7574198 10.279341  26.32489
348      1323_at  2.5287692 12.635851  18.88615
3147  33117_r_at  3.3403499 13.021890  17.82386
1150   2036_s_at  1.7707213 10.659906  16.94543
6043    35985_at  2.5864848 10.064852  16.54727
8164    38086_at -1.5946226  9.294861 -16.24335
9460    39370_at  1.6575902 10.298392  15.51091
```

```
2689    32664_at -1.5600419 10.324557 -14.79767
1959  31941_s_at -0.9947943  8.914654 -14.32894
8902    38817_at  1.0199253 10.161200  13.82985
          P.Value    adj.P.Val         B
11262 6.223955e-10 7.857743e-06 11.755021
348   1.231552e-08 7.774171e-05  9.852284
3147  2.064207e-08 8.686871e-05  9.471719
1150  3.236202e-08 9.912720e-05  9.129020
6043  3.996864e-08 9.912720e-05  8.964595
8164  4.710996e-08 9.912720e-05  8.835035
9460  7.086836e-08 1.278162e-04  8.507590
2689  1.073555e-07 1.694203e-04  8.166605
1959  1.424970e-07 1.998916e-04  7.929708
8902  1.944918e-07 2.407189e-04  7.665521

> genePpa <- topTable(lm.out, coef = 3)
> genePpa

             ID      logFC   AveExpr         t      P.Value
9851 39758_f_at -2.5243528 12.395711 -30.77351 1.515265e-10
3147 33117_r_at  3.2717919 13.021890  17.45804 2.482823e-08
1150  2036_s_at  1.7412527 10.659906  16.66342 3.756306e-08
8164   38086_at -1.5294406  9.294861 -15.57938 6.816251e-08
8055   37978_at -1.6536445 10.920988 -13.25057 2.827657e-07
1959 31941_s_at -0.9129768  8.914654 -13.15045 3.021147e-07
2612 32588_s_at -1.6751211 11.388069 -12.98728 3.368583e-07
8903   38818_at -1.3861127  9.863251 -12.35339 5.204210e-07
9247   39159_at  0.8904531 10.218632  12.04530 6.476456e-07
2603   32579_at  1.0850460 10.078867  11.62180 8.820629e-07
        adj.P.Val         B
9851 1.913022e-06 11.867838
3147 1.567282e-04  9.056947
1150 1.580779e-04  8.761991
8164 2.151379e-04  8.321296
8055 6.075479e-04  7.196830
1959 6.075479e-04  7.142202
2612 6.075479e-04  7.051938
8903 8.212894e-04  6.686141
9247 9.085028e-04  6.499228
2603 1.033557e-03  6.231948

> comparelists(geneHsa[, 1], genePpa[, 1])
```

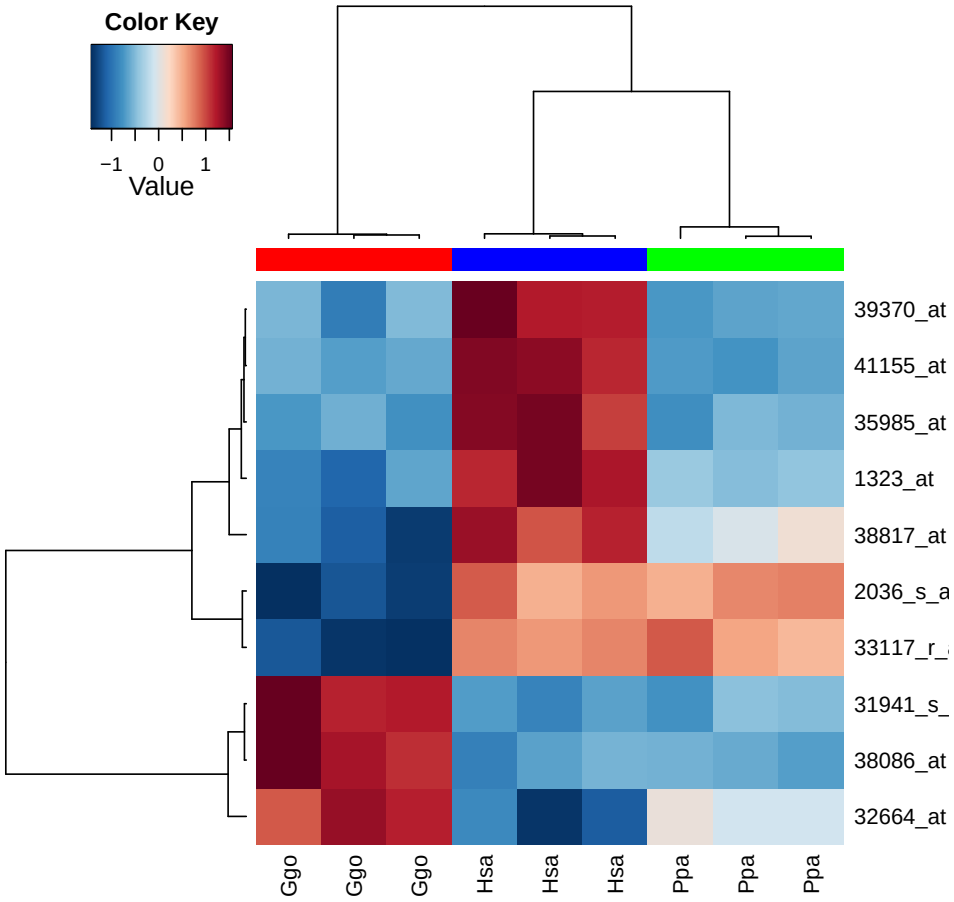
```
Items in X: 10
Items in Y: 10

No of vecX in vecY 4
No of vecY in vecX 4

Intersection of sets is 4
Difference in sets is 6
```

```
> heatplot(data.vsn[geneHsa[, 1], ], classvec = annt$Donor,
+     labCol = annt$Donor)

[1] "Data (original) range:  8.49 14.56"
[1] "Data (scale) range:  -1.43 1.56"
[1] "Data scaled to range:  -1.43 1.56"
     Class Color
[1,] "Ggo" "red"
[2,] "Hsa" "blue"
[3,] "Ppa" "green"
```


Color Key
−1
0
1
Value
39370_at
41155_at
35985_at
1323_at
38817_at
2036_s_a
33117_r_
31941_s_
38086_at
32664_at
Ggo
Ggo
Ggo
Hsa
Hsa
Hsa
Ppa
Ppa
Ppa

6.3 Advanced Task 3: Comparing datasets (meta-analysis) using Coinertia Analysis

Coinertia analysis has been applied to the cross-platform comparison (meta-analysis) of microarray gene expression datasets (9). 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 R package *ade4*.

Lets examine two gene expression datasets of the same 60 cell lines. The NCI60 cells lines are a set of 60 cell lines with different tumour phenotypes (eg Breast, Colon, Leukemia, Prostate, CNS, lung cancer, ovarian, renal cancer etc). The gene expression of these cell lines have been examined by a number of groups (10),(11).

The same 60 cell lines were analysed by different labs on differnt microarray platforms. We will compare one from Affymetrix (Staunton et al., 2001) and one that was obtained using Stanford spotted cDNA arrays (Ross et al., 2000) using `cia`. These 2 datasets were analyzed using `cia` by Culhane et al., 2003 (9).

These 2 datasets are available in the *made4* data package `NCI60`. 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.

```
> data(NCI60)
> summary(NCI60)

        Length Class      Mode
Ross     60    data.frame list
Affy     60    data.frame list
classes 120    -none-     character
Annot     4    data.frame list

> names(NCI60)

[1] "Ross"    "Affy"    "classes" "Annot"

> NCI60$classes[1:3, ]

     Sample          Class
[1,] "BREAST_BT549"  "BREAST"
[2,] "BREAST_HS578T" "BREAST"
[3,] "BREAST_MCF7"   "BREAST"

> table(NCI60$classes[, 2])
```

```
  BREAST      CNS    COLON     LEUK    MELAN    NSCLC
       8        6        7        6        8        9
    OVAR PROSTATE    RENAL
       6        2        8
```

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

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

```
> coin$coinertia
```

```
Coinertia analysis
call: coinertia(dudiX = t.dudi(coa1), dudiY = t.dudi(coa2), scannf = cia.scan,
    nf = cia.nf)
class: coinertia dudi

$rank (rank)     : 59
$nf (axis saved) : 2
$RV (RV coeff)   : 0.7859656

eigen values: 2.266e-05 9.904e-06 4.342e-06 2.335e-06 1.576e-06 ...

  vector length mode    content
1 $eig   59     numeric eigen values
2 $lw    144    numeric row weigths (crossed array)
3 $cw    144    numeric col weigths (crossed array)

   data.frame nrow ncol content
1  $tab       144  144  crossed array (CA)
2  $li        144  2    Y col = CA row: coordinates
3  $l1        144  2    Y col = CA row: normed scores
4  $co        144  2    X col = CA column: coordinates
5  $c1        144  2    X col = CA column: normed scores
6  $lX        60   2    row coordinates (X)
7  $mX        60   2    normed row scores (X)
8  $lY        60   2    row coordinates (Y)
9  $mY        60   2    normed row scores (Y)
10 $aX        2    2    axis onto co-inertia axis (X)
11 $aY        2    2    axis onto co-inertia axis (Y)
```

The RV coefficient $RV which is 0.786 in this instance, is a measure of "global" similarity between the datasets. The closer to 1, in the scale 0-1 the greater the correlation between the two datasets.

```
> coin$coinertia$RV

[1] 0.7859656
```

To visually examine the cell lines that have similar or different gene expression profiles in these datasets, use `plotarrays`.

```
> plotarrays(coin, classvec = NCI60$classes[, 2],
+     lab = "", cpoint = 3)
```

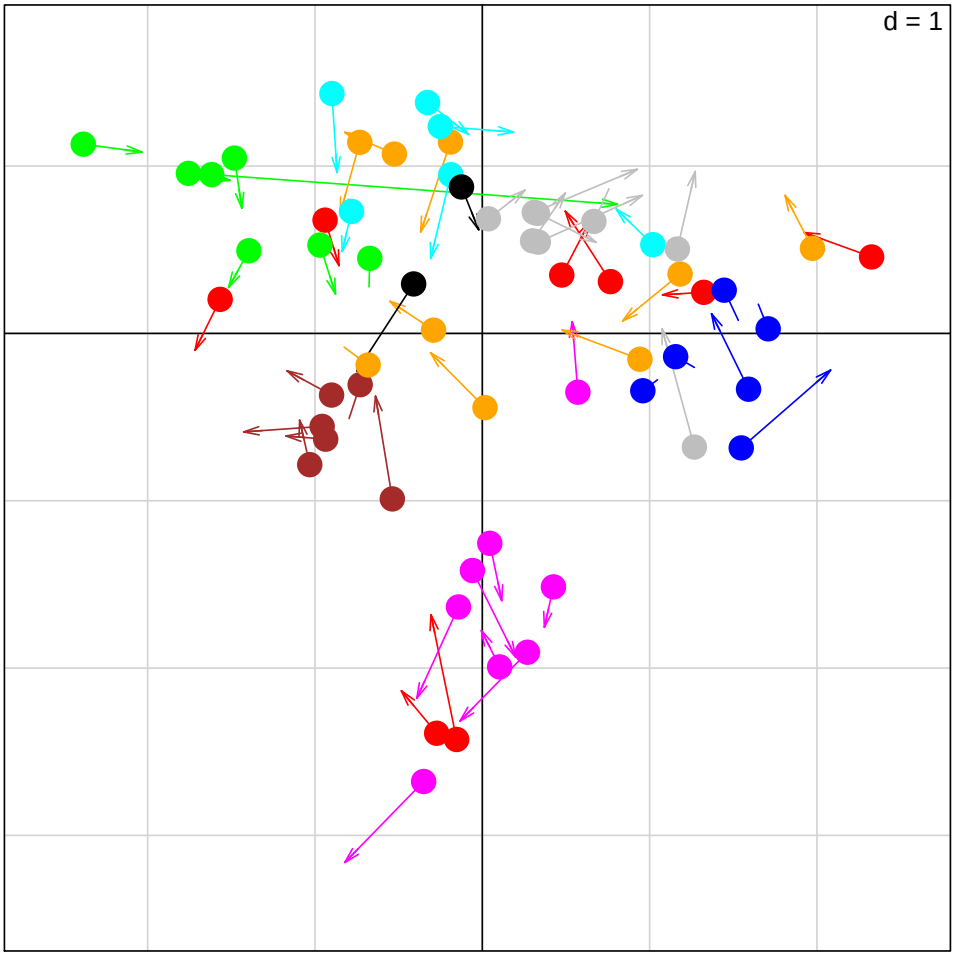


Figure 8: Coinertia analysis of NCI 60 cell line Spotted and Affymetrix gene expression dataset. Each cell lines is colored by its phenotype (eg colon are green,breast are red, melanoma are pink etc). For each of the 60 cell lines, there are two coordinates ($coinertia$mX and $coinertia$mY). On the plot, these are visually shown as a closed circle and an arrow. These are joined by a line. If the profiles are similar they will be projected close together in the new space (ie joined by a short line). For more information see Culhane et al., BMC bioinformatics 2003.

If `plot` is used, the above plot together with the plots of the gene projections from each dataset can be visualized.

```
> plot(coin, classvec = NCI60$classes[, 2])
```

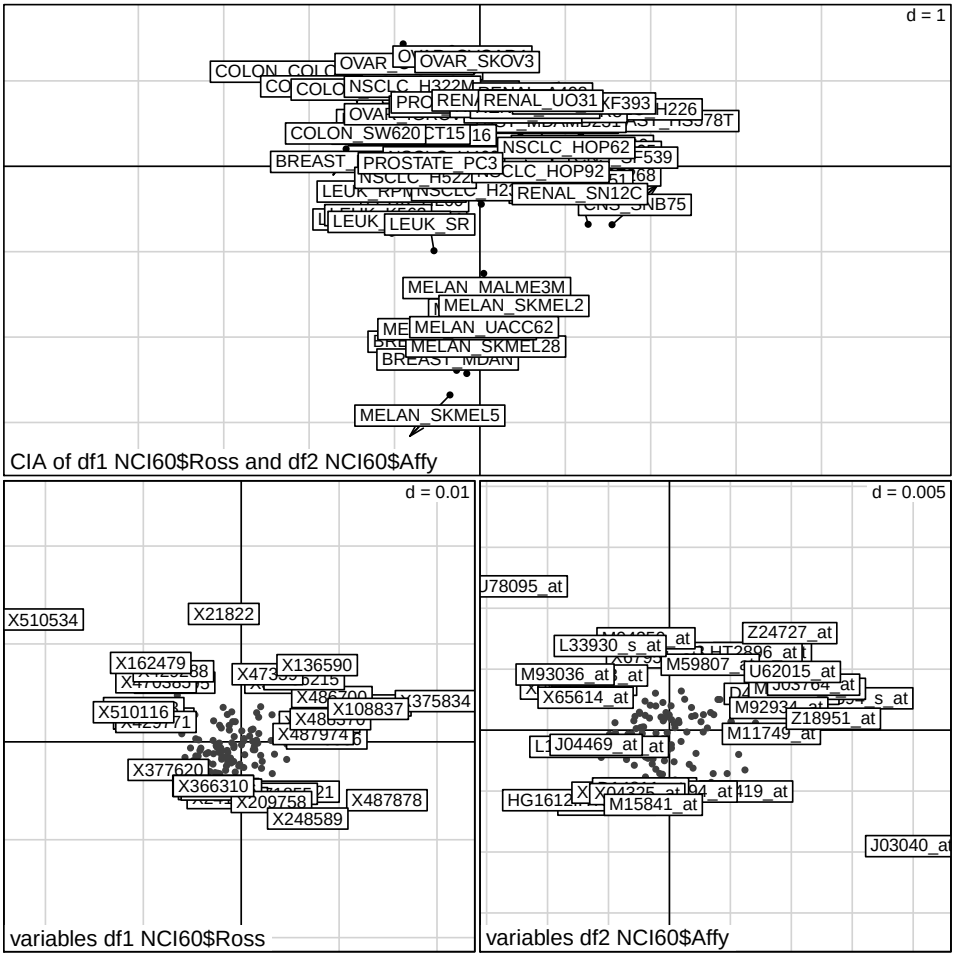


Figure 9: Coinertia analysis of NCI 60 cell line Spotted and Affymetrix gene expression dataset. A) shows a plot of the 60 microarray samples projected onto the one space. The 60 circles represent dataset 1 (Ross) and the 60 arrows represent dataset 2 (affy). Each circle and arrow are joined by a line, 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.

Coinertia analysis be applied to other types of data including the integration of gene expression and transcription factor binding site data (12) or to the analysis of gene and protein expression data (13).

7 vsn normalization of data.. for information only

For information only, please don't repeat in this today. *vsn* was used to normalize the Affymetrix data. To produce the normalized data, the cel files were downloaded, and then in R, use File -> Change directory (or `setwd` to select the directory containing the cel files. Then

```
getwd()
dir()

library(affy)
library(vsn)
cels <- list.celfiles()
data <- ReadAffy(filenames= cels)
normalize.AffyBatch.methods <- c(normalize.AffyBatch.methods, "vsn")

data.vsn <- es1 = expresso(data, bg.correct = FALSE,
normalize.method = "vsn", pmcorrect.method = "pmonly",
summary.method   = "medianpolish")

exprs2excel(data.vsn, file="data.vsn.csv")
```

A tab delimited text file could also be saved using

```
write.exprs(data.vsn, file="data.rma.txt")
```

8 Further help

More information about *made4* is available at `http://www.bioconductor.org`.

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 (6). The *ade4* homepage is `http://pbil.univ-lyon1.fr/ADE-4`.

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

For more examplez and information on *made4*, please see:

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

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.

Information about this session:

```
> sessionInfo()

R version 2.13.1 (2011-07-08)
Platform: i386-pc-mingw32/i386 (32-bit)

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

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

other attached packages:
 [1] hgu95av2.db_2.5.0    org.Hs.eg.db_2.5.0
 [3] annaffy_1.24.0       KEGG.db_2.5.0
 [5] GO.db_2.5.0          RSQLite_0.9-4
 [7] DBI_0.2-5            AnnotationDbi_1.14.1
 [9] limma_3.8.2          made4_1.26.0
[11] scatterplot3d_0.3-33 gplots_2.8.0
[13] caTools_1.12         bitops_1.0-4.1
[15] gdata_2.8.2          gtools_2.6.2
[17] RColorBrewer_1.0-5   ade4_1.4-17
[19] affy_1.30.0          Biobase_2.12.2

loaded via a namespace (and not attached):
[1] affyio_1.20.0         preprocessCore_1.14.0
[3] tools_2.13.1
```

References

[1] Karaman MW, Houck ML, Chemnick LG, Nagpal S, Chawannakul D, Sudano D, Pike BL, Ho VV, Ryder OA, Hacia JG Comparative analysis of gene-expression patterns in human and African great ape cultured fibroblasts. *Genome Res.* **13(7)**:1619-30.2003.

[2] Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. Cluster analysis and

display of genome-wide expression patSwterns. *Proc Natl Acad Sci U S A* **95**: 14863-14868. 1998.

[3] 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.

[4] Jeffery IB, Higgins DG, Culhane AC. Comparison and evaluation of microarray feature selection methods. *BMC Bioinformatics* **7**:359. 2006.

[5] Tusher VG, Tibshirani R, Chu G. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A*.**98(9)**:5116-21. 2001.

[6] 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.

[7] 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.

[8] Khan,J., Wei,J.S., Ringner,M., Saal,L.H., Ladanyi,M., Westermann,F., Berthold,F., Schwab,M., Antonescu,C.R., Peterson,C. et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.* **7**:673-679.

[9] Culhane AC, Perriere G, Higgins DG. Cross platform comparison and visualisation of gene expression data using co-inertia analysis. *BMC Bioinformatics*.**4**:59. 2003.

[10] Ross DT, Scherf U, Eisen MB, Perou CM, Rees C, Spellman P, Iyer V, Jeffrey SS, Van de Rijn M, Waltham M, Pergamenschikov A, Lee JC, Lashkari D, Shalon D, Myers TG, Weinstein JN, Botstein D, Brown PO. Systematic variation in gene expression patterns in human cancer cell lines. *Nat Genet* **24**:227-235 2000,

[11] Staunton JE, Slonim DK, Coller HA, Tamayo P, Angelo MJ, Park J, Scherf U, Lee JK, Reinhold WO, Weinstein JN, Mesirov JP, Lander ES, Golub TR: Chemosensitivity prediction by transcriptional profiling. *Proc Natl Acad Sci U S A* **98**:10787-10792. 2001

[12] Jeffery IB,Madden SF,McGettigan PA,Perriere G, Culhane AC, Higgins DG Integrating transcription factor binding site information with gene expression datasets. *Bioinformatics* **23(3)**:298-305.2006.

[13] Fagan A, Culhane AC, Higgins DG. A multivariate analysis approach to the integration of proteomic and gene expression data. *Proteomics* Jun 5 2007.