Genómica Funcional y Análisis de Microarrays

PEC 2- Segunda Prueba de Evaluación Continua

Author: Ramón Tamarit Agusti

Análisis de cluster no supervisados. Aplicaciones en la búsqueda y visualización de perfiles de expresión en datos de microarrays.

Resumen

Existen multitud de técnicas para resolver el problema de la determinación de los patrones de expresión a partir de los datos de microarrays. Cada una de las técnicas dispone igualmente de distintos parámetros o formas de medida, y en cada caso pueden obtenerse resultados distintos. El objetivo de este trabajo es presentar de forma sencilla una comparativa de las siguientes técnicas de análisis no supervisado:

- *HC, Cluster jerárquico,*
- *PCA, Análisis de componentes principales,*
- *PAM, Clusters partitivos,*
- *SOM, Mapas autoorganizativos,*
- *MDS, escalado multidimensional*

Para comparar las técnicas se usa el conjunto de datos del experimento:" Arabidopsis thaliana gene expression in response to IAA challenge", *publicado en GEO (http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE1110)..*

Introducción

Los métodos de agrupación o clustering se pueden clasificar en:

- Métodos supervisados: Se emplean básicamente para encontrar una *firma molecular* o un conjunto reducido de genes cuyo perfil de expresión permita clasificar una muestra, es decir partimos de patrón de expresión génica determinado. Una aplicación típica es clasificar una muestra de un paciente con una determinada dolencia en alguno de los grupos ya establecidos.
- Métodos no supervisados: El objetivo principal es determinar que elementos ya sean genes o muestras presentan un patrón similar. La aplicación de los métodos no supervisados es descubrir los patrones de expresión que posteriormente podrán usarse en análisis supervisados, en detectar genes corregulados.

Para construir los grupos de genes o muestras con perfiles de expresión similares se tiene que utilizar una medida de distancia. Las medidas de distancia más usadas son la *euclidiana* y la correlación de *Pearson y de Sperman*. En el caso de los métodos de agrupamiento jerárquicos hay que además definir el método para determinar distancias entre conjuntos de genes.

Los métodos de agrupamiento por lo general no necesitan de una información de partida sobre los clusters, sino que son los algoritmos los que agrupan las muestras basándose en el grado de similitud entre los perfiles de expresión de los genes en estudio. El método de agrupamiento más empleado en datos de microarreglos es el agrupamiento jerárquico. Este método no supervisado deriva una serie de particiones de los datos; en este caso, cada dato será el perfil de expresión de una muestra o gen. Existen varios tipos de métodos de agrupamiento jerárquicos, tales como el aglomerativo y el divisivo, los divisivos funcionan mejor para dividir los datos en pocos grupos de varios elementos. El resultado de estos métodos es una estructura de árbol o dendograma.

Como alternativa a los métodos jerárquicos están los métodos partitivos. El método *k-Means* es el más usado. Tiene la desventaja de que requiere como entrada el número de grupos en que se considera estén separados los datos. La estimación de *k (número de grupos)* es un problema conocido, siempre que se desea encontrar el mapeo de cualquier estructura de datos a una estructura de grupos, especialmente estudiado en datos de expresión de genes. Un criterio muy usado propone seleccionar a *k* como el número de grupos a partir del cual se observan pocas variaciones de las ordenadas del gráfico FOM (Figure of Merit). Otros métodos se basan en evaluar la estabilidad de los grupos.

Hay que destacar que el análisis por grupos resuelve directamente el problema de predicción y comparación de clases. Los análisis de cluster no supervisados no aportan información cuantitativa válida desde el punto de vista estadístico sobre cuáles genes se expresan diferencialmente entre clases, y hay que tomarlos como un método exploratorio previo.

El tipo de técnica a utilizar depende del objetivo de la investigación o del problema. En general los métodos jerárquicos son prefereribles cuando no tenemos una idea precisa de los patrones de respuesta que podemos encontrar y cuando podemos encontrar puntos muy separados entre si. Los métodos de partición pueden ser interesantes cuando ya conocemos en una primera aproximación que agrupaciones son posibles o esperables. Por ejemplo podemos partir en una primera aproximación de con un método jerárquico y utilizar el resultado como punto de partida para el método partitivo.

Descripción del experimento

Microarrays

Los microarrays son de la marca Affymetrix, en concreto el modelo utilizado es el Affymetrix Arabidopsis ATH1 Genome Array una descripción se encuentra en (http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GPL198)

Platform GPL198

Status

Public on Jul 18, 2002

Title

[ATH1-121501] Affymetrix Arabidopsis ATH1 Genome Array

Description

The current release has 22810 entries and was indexed 26-Jun-2003.
Annotation data from TAIR, Gene Ontology Consortium and TIGR were mapped to the Arabidopsis ATH1 Array probe sets. The AGI (Arabidopsis Genome Initiative) ID (e.g. AT5G23000) corresponding to the gene represented on the array was used to map annotation data obtained from TAIR, Gene Ontology and TIGR databases. Similarly, gene title and gene symbol, as well as the EC annotations for the AGI ID were extracted from the TIGR database. These annotation mappings were validated by a two-pronged approach. First, probe sets were randomly selected and manually curated to check for consistency between Gene Ontology terms, gene title and protein domain associations. Furthermore, associations within and between data sets from different public databases were also used to check for consistencies. For example, consider the association of gene ontology terms and InterPro IDs.

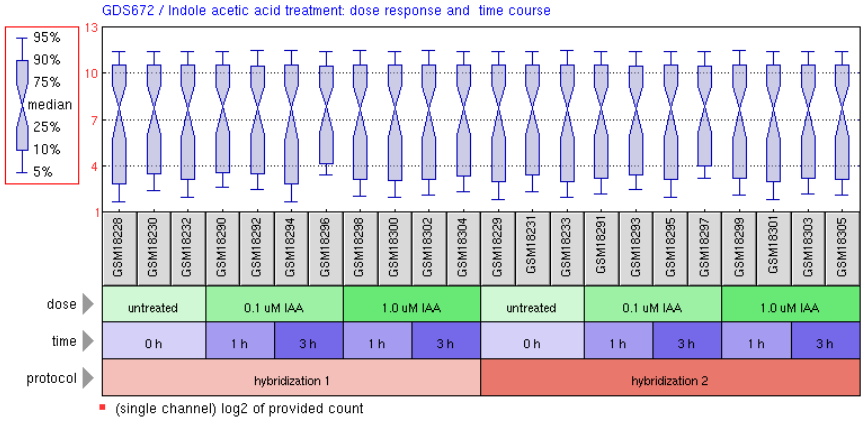
Several Gene Onoltogy terms have InterPro ID(s) curated as supporting evidence for assigning the term to an AGI locus. This relationship was used to validate the consistency of the ontology terms from Gene Onotlogy and InterPro domain annotations from TAIR. The GeneChip® Arabidopsis ATH1 Genome Array contains more than 22,500 probe sets representing approximately 24,000 genes. Sequences used in the design were selected and clustered in collaboration with TIGR and were derived from TIGR's ATH1-121501 Database. Oligoneuclotide probes are synthesized in situ to each corresponding sequence. Eleven pairs of oligoneuclotide probes are used to measure the level of transcription of each sequence represented on the GeneChip Arabidopsis ATH1 Genome Array. Most sequences represented on the previous generation GeneChip® Arabidopsis Genome Array are also represented on the ATH1 array. Due to the dynamic nature of public databases, probe sets for these sequences will not be identical and in some cases will be represented by a completely new probe set. As a result, data generated with different versions of the Arabidopsis array may not always produce concordant results. The probe arrays are for research use only and not intended for use in diagnosis of diseases.

http://www.affymetrix.com/support/technical/byproduct.affx?product=arab
http://www.affymetrix.com/analysis/index.affx

Diseño experimental

El diseño experimental (http://www.ncbi.nlm.nih.gov/projects/geo/gds/profileGraph.cgi?gds=672) consta de 20 arrays, con el siguiente de diseño experimental: Dos replicas biológicas (hibridación 1 y 2), cada replica se compone de una muestra de control con tres replicas técnicas y un diseño factorial de 2x2, tratamiento con Acido Acetico y tiempo de exposición.

Title: GDS672 / Indole acetic acid treatment: dose response and time course / Arabidopsis thaliana
Summary: Analysis of 10-12 day Columbia-0 seedlings treated with either 0.1 uM or 1.0 uM indole acetic acid (IAA) auxin for 1 and 3 hours. Treated seedlings analyzed in reference to untreated control. Part of a study assessing the performance of the ATH1 array.



Graph caption help

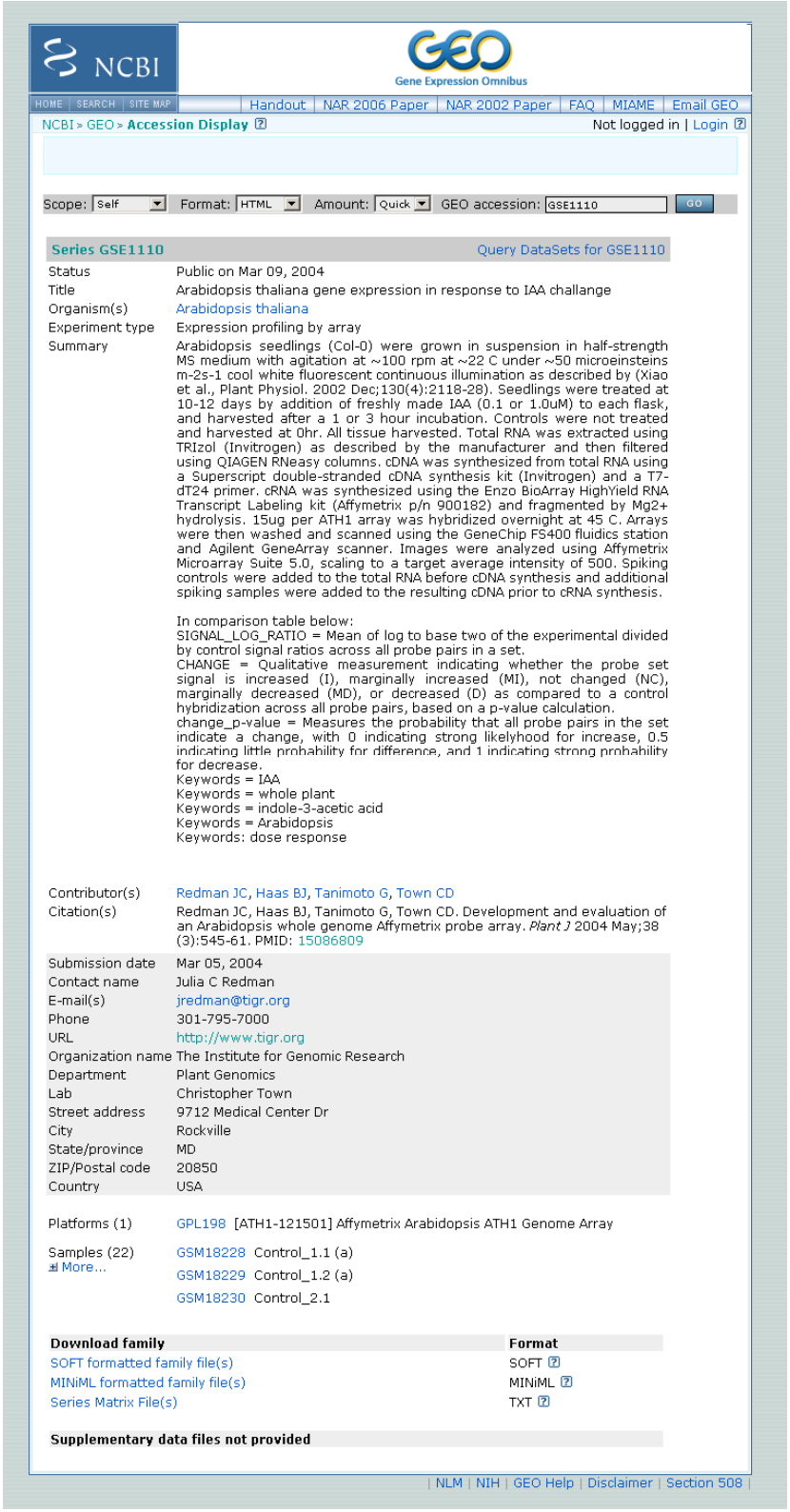
Samples:

GSM18228: Control_1.1 (a)
GSM18230: Control_2.1
GSM18232: Control_3.1
GSM18290: 0.1uM_IAA_1h_1.1
GSM18292: 0.1uM_IAA_1h_2.1
GSM18294: 0.1uM_IAA_3h_1.1
GSM18296: 0.1uM_IAA_3h_2.1
GSM18298: 1.0uM_IAA_1h_1.1
GSM18300: 1.0uM_IAA_1h_2.1
GSM18302: 1.0uM_IAA_3h_1.1
GSM18304: 1.0uM_IAA_3h_2.1
GSM18229: Control_1.2 (a)
GSM18231: Control_2.2
GSM18233: Control_3.2
GSM18291: 0.1uM_IAA_1h_1.2
GSM18293: 0.1uM_IAA_1h_2.2
GSM18295: 0.1uM_IAA_3h_1.2
GSM18297: 0.1uM_IAA_3h_2.2
GSM18299: 1.0uM_IAA_1h_1.2
GSM18301: 1.0uM_IAA_1h_2.2
GSM18303: 1.0uM_IAA_3h_1.2
GSM18305: 1.0uM_IAA_3h_2.2

Información experimental.

La información experimental esta disponible que GEO en la dirección:
http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE1110

La citación en PubMed es: http://www.ncbi.nlm.nih.gov/sites/entrez?Db=Pubmed&term=15086809[UID]



NCBI

Gene Expression Omnibus

HOME | SEARCH | SITE MAP | Handout | NAR 2006 Paper | NAR 2002 Paper | FAQ | MIAME | Email GEO

NCBI » GEO » **Accession Display** Not logged in | Login

Scope: Self Format: HTML Amount: Quick GEO accession: GSE1110 GO

Series GSE1110 Query DataSets for GSE1110

Status	Public on Mar 09, 2004
Title	Arabidopsis thaliana gene expression in response to IAA challange
Organism(s)	Arabidopsis thaliana
Experiment type	Expression profiling by array
Summary	Arabidopsis seedlings (Col-0) were grown in suspension in half-strength MS medium with agitation at ~100 rpm at ~22 C under ~50 microeinsteins m-2s-1 cool white fluorescent continuous illumination as described by (Xiao et al., Plant Physiol. 2002 Dec;130(4):2118-28). Seedlings were treated at 10-12 days by addition of freshly made IAA (0.1 or 1.0uM) to each flask, and harvested after a 1 or 3 hour incubation. Controls were not treated and harvested at 0hr. All tissue harvested. Total RNA was extracted using TRIzol (Invitrogen) as described by the manufacturer and then filtered using QIAGEN RNeasy columns. cDNA was synthesized from total RNA using a Superscript double-stranded cDNA synthesis kit (Invitrogen) and a T7-dT24 primer. cRNA was synthesized using the Enzo BioArray HighYield RNA Transcript Labeling kit (Affymetrix p/n 900182) and fragmented by Mg2+ hydrolysis. 15ug per ATH1 array was hybridized overnight at 45 C. Arrays were then washed and scanned using the GeneChip FS400 fluidics station and Agilent GeneArray scanner. Images were analyzed using Affymetrix Microarray Suite 5.0, scaling to a target average intensity of 500. Spiking controls were added to the total RNA before cDNA synthesis and additional spiking samples were added to the resulting cDNA prior to cRNA synthesis. In comparison table below: SIGNAL_LOG_RATIO = Mean of log to base two of the experimental divided by control signal ratios across all probe pairs in a set. CHANGE = Qualitative measurement indicating whether the probe set signal is increased (I), marginally increased (MI), not changed (NC), marginally decreased (MD), or decreased (D) as compared to a control hybridization across all probe pairs, based on a p-value calculation. change_p-value = Measures the probability that all probe pairs in the set indicate a change, with 0 indicating strong likelyhood for increase, 0.5 indicating little probability for difference, and 1 indicating strong probability for decrease. Keywords = IAA Keywords = whole plant Keywords = indole-3-acetic acid Keywords = Arabidopsis Keywords: dose response
Contributor(s)	Redman JC, Haas BJ, Tanimoto G, Town CD
Citation(s)	Redman JC, Haas BJ, Tanimoto G, Town CD. Development and evaluation of an Arabidopsis whole genome Affymetrix probe array. *Plant J* 2004 May;38 (3):545-61. PMID: 15086809
Submission date	Mar 05, 2004
Contact name	Julia C Redman
E-mail(s)	jredman@tigr.org
Phone	301-795-7000
URL	http://www.tigr.org
Organization name	The Institute for Genomic Research
Department	Plant Genomics
Lab	Christopher Town
Street address	9712 Medical Center Dr
City	Rockville
State/province	MD
ZIP/Postal code	20850
Country	USA
Platforms (1)	GPL198 [ATH1-121501] Affymetrix Arabidopsis ATH1 Genome Array
Samples (22)	GSM18228 Control_1.1 (a) GSM18229 Control_1.2 (a) GSM18230 Control_2.1 More...

Download family	Format
SOFT formatted family file(s)	SOFT
MINiML formatted family file(s)	MINiML
Series Matrix File(s)	TXT

Supplementary data files not provided

NLM | NIH | GEO Help | Disclaimer | Section 508

Metodología y flujo de análisis

Carga de los datos

El proceso de carga de los datos desde GEO lo realizamos con el siguiente código. Los valores de expresión los guardamos en un fichero de texto para posterior uso, visualización y/o modificación.

```
################################################
##### CARGA DE LOS DATOS DESDE GEO  #############
################################################
library(GEOquery)
library(Biobase)
myGEOdata <-getGEO("GSE1110")
class(myGEOdata); names(myGEOdata); class(myGEOdata[[1]])
mydata <-exprs(myGEOdata[[1]])
save(mydata, file= "GSE1110_series_matrix.Rda")
##GUARDAMOS LOS VALORES DE EXPRESIÓN EN UN FICHERO DE TEXTO
write.table(mydata, "expresion_orig.txt",col.names=TRUE, sep="\t")
head(mydata)
################################################
##### VISUALIZACIÓN DE LOS DATOS LEIDOS #########
################################################
class(exprs)
dim(exprs)
colnames(exprs)
head(exprs)
```

El fichero de texto en donde hemos guardado los valores de expresión se pude usar posteriormente para no tener que volver a repetir la descarga desde GEO.

```
##RECUPERAMOS LOS VALORES DE EXPRESIÓN EN UN FICHERO DE TEXTO
exprs <- as.matrix(read.table("expresion_orig.txt", header = TRUE, sep = "\t",
+ row.names = 1, as.is = TRUE))
```

Filtrado de los datos

Mediante el filtrado de lo datos seleccionamos los que presentan mayores valores de intensidad y mayor variabilidad o niveles de expresión.

```
#FILTRADO.
mydata <- mydata[apply(mydata>100, 1, sum)/length(mydata[1,])>0.5 & apply(log2(mydata),
+ 1, IQR)>1.5,]
summary(mydata)
```

Para mejorar este proceso, o buscando otros objetivos podríamos emplear otras herramientas estadísticas con el objeto de obtener un conjunto de datos que nos sea significativo. Después del filtrado el conjunto de datos se reduce a 123 probes, suficientes para el objetivo de este estudio.

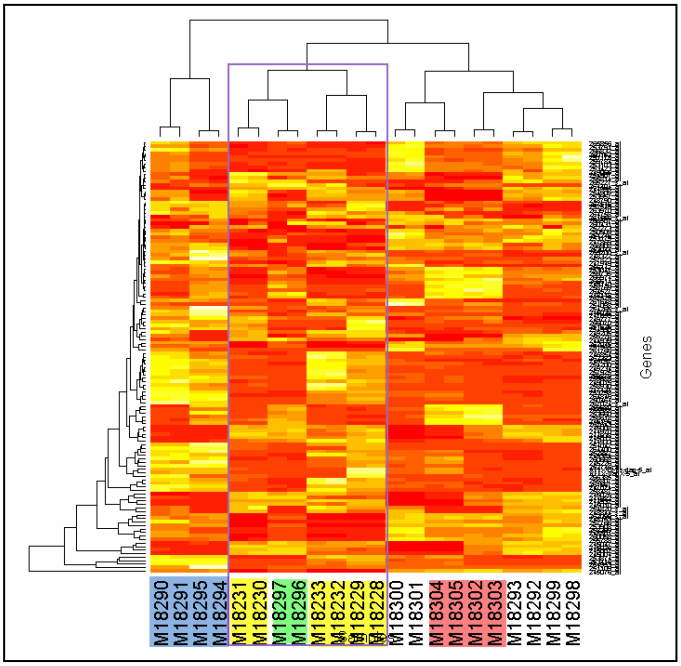
Exploración de los datos mediante Heat-Map y clustering jerarquico.

Exploración preliminar

Antes de comenzar con los análisis de cluster realizaremos un simple Heat-Map de los datos mediante la función `heatmap()`, de esta forma comprobamos cual es el perfil de los datos sin ordenar.

```
### Visualización con heat-map
heatmap (mydata, cexRow=0.5, cexCol=1.5,xlab="Samples", ylab="Genes")
```

GSM18228	Control_1.1	Hibridación 1	0 uM	0 horas
GSM18230	Control_2.1			
GSM18232	Control_3.1			
GSM18290	0.1uM_IAA_1h_1.1		0.1 uM	1 hora
GSM18292	0.1uM_IAA_1h_2.1			
GSM18294	0.1uM_IAA_3h_1.1			3 horas
GSM18296	0.1uM_IAA_3h_2.1			
GSM18298	1.0uM_IAA_1h_1.1		1.0 uM	1 hora
GSM18300	1.0uM_IAA_1h_2.1			
GSM18302	1.0uM_IAA_3h_1.1			3 horas
GSM18304	1.0uM_IAA_3h_2.1			
GSM18229	Control_1.2	Hibridación 2	0 uM	0 horas
GSM18231	Control_2.2			
GSM18233	Control_3.2			
GSM18291	0.1uM_IAA_1h_1.2		0.1 uM	1 hora
GSM18293	0.1uM_IAA_1h_2.2			
GSM18295	0.1uM_IAA_3h_1.2			3 horas
GSM18297	0.1uM_IAA_3h_2.2			
GSM18299	1.0uM_IAA_1h_1.2		1.0 uM	1 hora
GSM18301	1.0uM_IAA_1h_2.2			
GSM18303	1.0uM_IAA_3h_1.2			3 horas
GSM18305	1.0uM_IAA_3h_2.2			



Por el momento nos vamos a fijar únicamente en como han quedado distribuidas las muestras. He marcado con colores algunas de las distribuciones que son significativas y se agrupan como podríamos esperar. En concreto las seis muestras de control (dos pares de tres replicas) deberían estar todas en la misma rama, pero no es así, al mismo nivel se incluyen dos, una de una hora y 0.1 uM, y otra de 0.1 uM y tres horas.

Una consideración a tener en cuenta es que la función heatmap() calcula las distancias entre genes y muestras usando un modelo euclideo, esta no es la forma más apropiada para tratar los datos de microarrays, especialmente en los experimentos de dosificación y evolución temporal. En general para este tipo de experimentos se ha demostrado mejores resultados con distancias tipo pearson o spearson

Análisis mediante cluster jerárquico.

En el caso de los métodos jerárquicos los datos se ordenan en niveles de manera que los niveles superiores contienen a los inferiores. La jerarquía construida permite obtener también una partición de los datos en grupos. Se utiliza la matriz de distancias o similitudes entre los elementos de la matriz original los de datos.

Los algoritmos jerárquicos pueden ser de dos tipos: De división y de Aglomeración. El algoritmo de división asume que en un primer paso todos los datos conforman un solo conglomerado. Este cluster se va dividiendo sucesivamente en conglomerados más pequeños de acuerdo a algún criterio seleccionado previamente. El resultado de este procedimiento se representa por el dendograma.

En el algoritmo de aglomeración cada observación inicialmente es un conglomerado y en cada paso se asocian los conglomerados mas similares hasta llegar a un solo cluster.

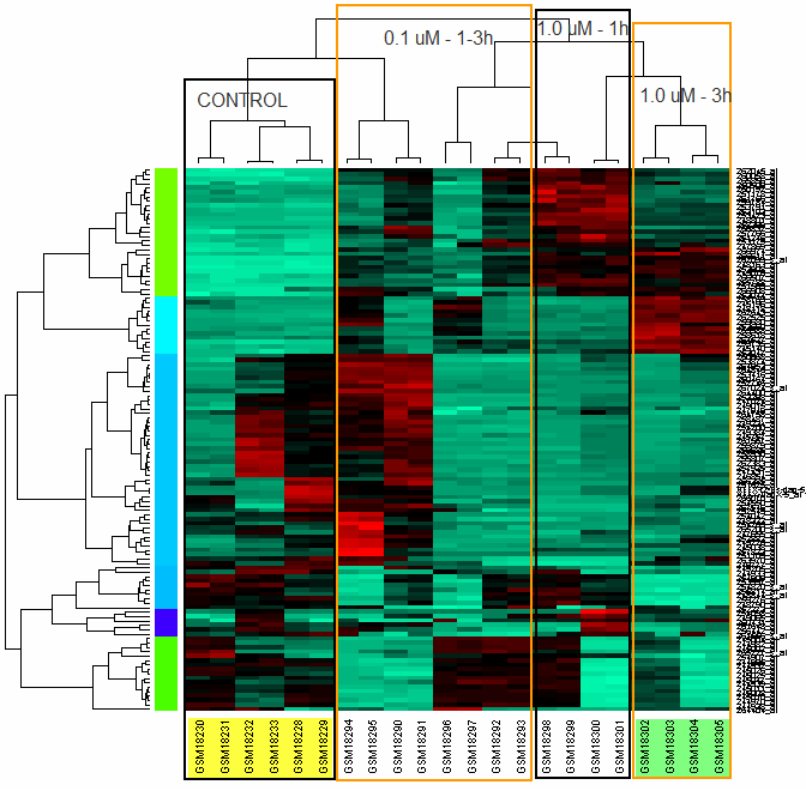
En el dendograma la escala vertical representa la distancia. La distancia entre dos conglomerados que se calcula según un algoritmo predeterminado. El algoritmo de cluster jerárquico pueden ser.

- Linkage promedio: promedio de las distancias de las observaciones en cada cluster.
- Linkage simple: la menor distancia entre las observaciones de cada cluster
- Linkage completo: la mayor distancia entre las observaciones de cada cluster.

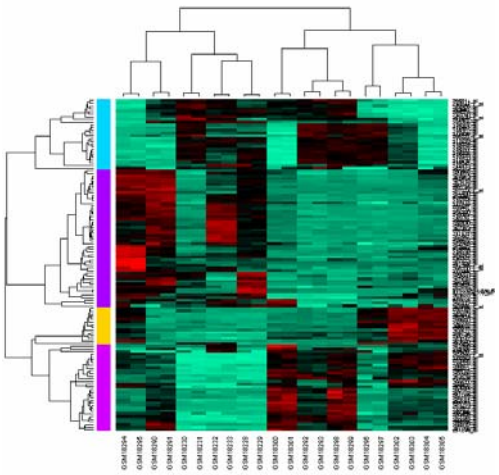
La implementación hclust de R (http://sekhon.berkeley.edu/stats/html/hclust.html) utiliza el método Lance-Williams que calcula y actualiza en cada paso la disimilaridad entre clusters, este método es aglomerativo.

Si cortamos el dendograma a un nivel de distancia dado, obtenemos una clasificación del número de grupos existentes a ese nivel y los elementos que los forman.

```
#######################################################
##### ANALISIS MEDIANTE CLUSTER JERARQUICO #########
#######################################################
# DESCARGA DEL PATRON DE COLOR DE GIRKE.
source("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/my.colorFct.R")
# Import an alternative color scheme for the heatmap function.

# CENTRADO Y ESCALADO DE LOS DATOS.
mydatascale <- t(scale(t(mydata)))
## GUARDAMOS LOS FICHEROS PARA ANALISIS POSTERIOR
write.table(mydata, "expresion_orig_filtered_RND_PRC.txt",col.names=TRUE, sep="\t")
write.table(mydatascale, "expresion_orig_scaled.txt_RND_PRC",col.names=TRUE, sep="\t")

hr <- hclust(as.dist(1-cor(t(mydatascale), method="pearson")), method="complete")
# Cluster rows by Pearson correlation.

hc <- hclust(as.dist(1-cor(mydatascale, method="spearman")), method="complete")
# Clusters columns by Spearman correlation.

x11()
heatmap(mydata, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col=my.colorFct(), scale="row")
# Plot the data table as heatmap and the cluster results as dendrograms.

mycl <- cutree(hr, h=max(hr$height)/1.5); mycolhc <- sample(rainbow(256));
mycolhc <- mycolhc[as.vector(mycl)];
heatmap(mydata, Rowv=as.dendrogram(hr), Colv=as.dendrogram(hc), col=my.colorFct(), scale="row",
RowSideColors=mycolhc)
# Cut the tree at specific height and color the corresponding clusters in the
# heatmap color bar.
```



Podemos comprobar (a la izquierda) que a nivel de muestras se pueden observar ya clusters biológicamente significativos, por lo que hemos mejorado la ordenación mediante la metrica sperman. El método “complete” por otra parte puede no ser el más adecuado. Si repetimos el cálculo para las muestras usando como parámetros pearson y average, no obtenemos la ordenación biológica que seria de esperar, lo que mejoramos en unas agrupaciones lo perdemos en otras (heat-map inferior)



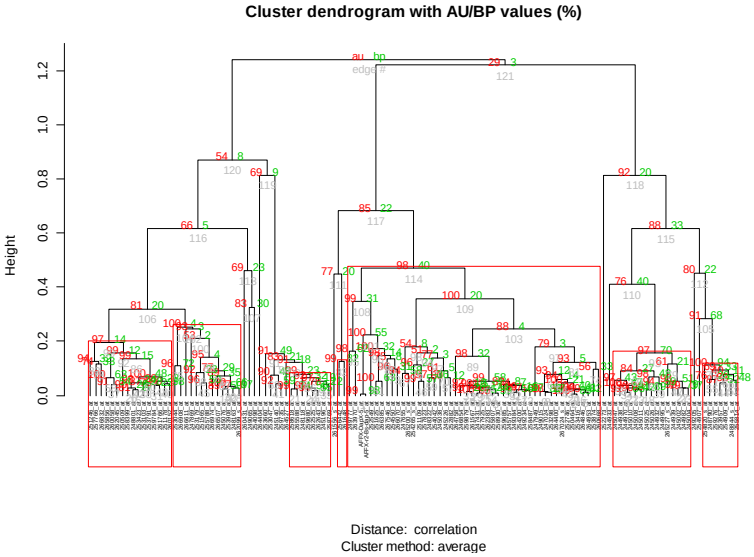
La agrupación de genes es en ambos casos similar, y tan solo en los genes cercanos parece que se incluyen en uno u otro grupo.

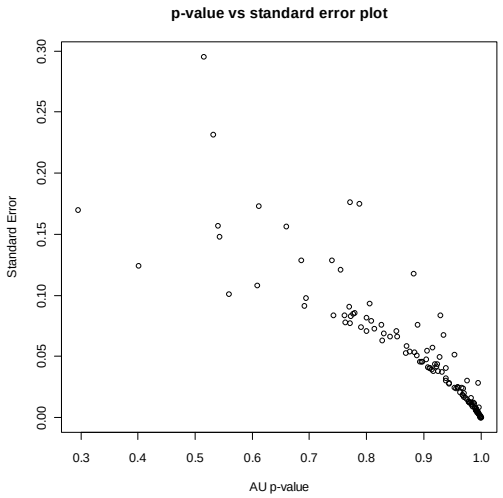
Evaluación de la incertidumbre de los cluster jerárquicos mediante bootstraping

El paquete pvcluster permite evaluar la incertidumbre de los cluster mediante un proceso iterativo http://bioinformatics.oxfordjournals.org/cgi/content/full/22/12/1540. Los valores de p-valor obtenidos nos permiten establecer un punto de corte para los clusters significativos, y evaluar la significación de los mismos.

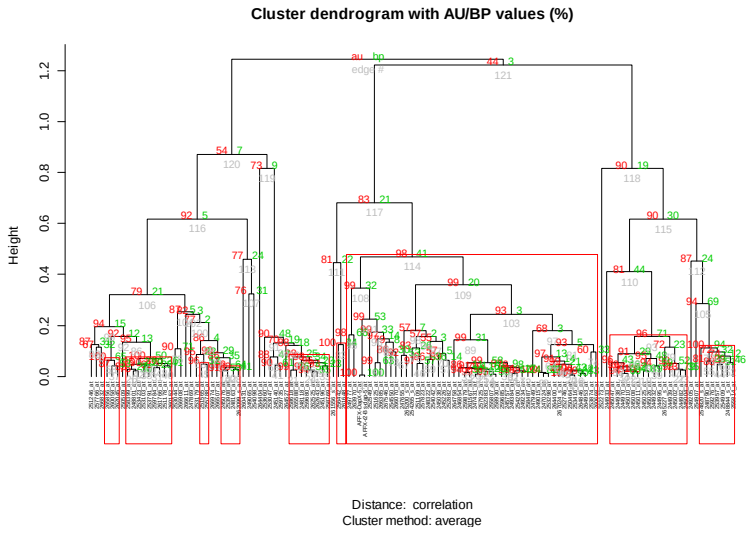
```
#############################################################
##### EVALUACIÓN DEL CLUSTER JERARQUICO CON PVCLUST #########
#############################################################
library(pvclust)
library(gplots)
# Loads the required pvclust package.

pv <- pvclust(scale(t(mydata)), method.dist="correlation", method.hclust="average", nboot=100)
#pv <- pvclust(scale(t(mydata)), method.dist="correlation", method.hclust="average",
nboot=1000)
# Perform the hierarchical cluster analysis.
# Due to time resrictions, we are using here only 10 bootstrap repetitions.
# Usually, one should use at least 1000 repetitions.

##GRAFICO SEPLOT DE LAS INCERTIDUMBRES
X11()
seplot.(pv)

x11()
plot(pv, hang=-1, cex=0.4);
pvrect(pv, alpha=0.95)
# Plots result as a dendrogram where the significant clusters
# are highlighted with red rectangles.

clsig <- unlist(pvpick(pv, alpha=0.95, pv="au", type="geq", max.only=TRUE)$clusters)
# Retrieve members of significant clusters.

source("http://faculty.ucr.edu/~tgirke/Documents/R_BioCond/My_R_Scripts/dendroCol.R")
# Import tree coloring function.
x11()
dend colored <- dendrapply(as.dendrogram(pv$hclust), dendroCol, keys=clsig, xPar="edgePar",
bgr="black", fgr="red", pch=20, cex=0.4)
# Create dendrogram object where the significant clusters are labeled in red.

heatmap(mydata, Rowv=dend colored, Colv=as.dendrogram(hc), col=my.colorFct(), scale="row",
RowSideColors=mycolhc)
# Plot the heatmap from above, but with the significant clusters in red
# and the cluster bins from the tree cutting step in the color bar.

x11(height=12);
heatmap.2(mydata, Rowv=dend colored, Colv=as.dendrogram(hc), col=my.colorFct(), scale="row",
trace="none", RowSideColors=mycolhc)
# Plot heatmap with heatmap.2()
# function which scales better for many entries.

mydatasort <- mydata[pv$hclust$labels[pv$hclust$order], hc$labels[hc$order]]
# Sort rows in data table by 'dend_colored' and its colums by 'hc'.
```

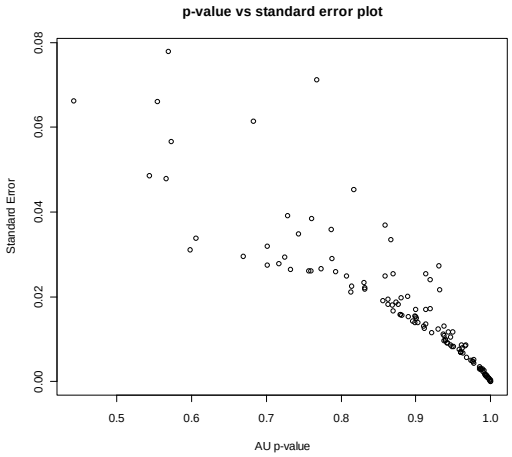
Abajo tenemos el dendrograma para los genes obtenido usando distancias pearson y método average, con 100 iteraciones.





Si aumentamos el número de iteraciones a 1000, comprobamos como los clusters a la izquierda se desdoblan en varios, y en el grafico seplot comprobamos que los errores estándar han disminuido drásticamente.





Análisis mediante clusters partitivos y comparación con los HC.

La función pam de la librería cluster encuentra los conglomerados usando el particionamiento alrededor de medoides. Las medoides, son instancias representativas de los clusters que se quieren formar. Para un pre-especificado número de clusters K, el procedimiento PAM está basado en la búsqueda iterativa de los K medoides, M = (m1, . . . ,mK) de todas las observaciones a clasificar

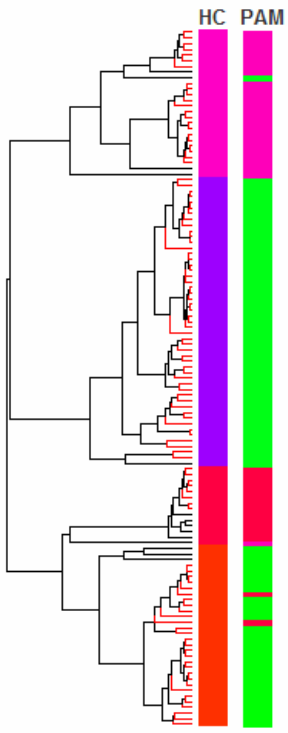
Para encontrar M hay que minimizar la suma de las distancias de las observaciones al Medoide mas cercano.

$$M^* = \arg\min_M \sum_i \min_k d(x_i, m_k)$$

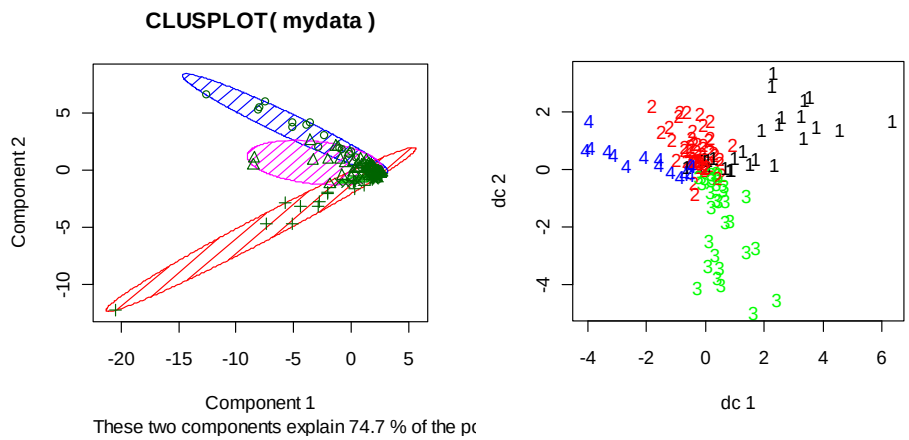
En donde d es una medida de disimilaridad

El código R para hacer los análisis es:

```
#############################################################
##### ANALISIS COM PAM #########
#############################################################

library(cluster)
# Loads required library.
mydist <- t(scale(t(mydata)))
# Center and scale data.
mydist <- as.dist(1-cor(t(mydist), method="pearson"))
# Generates distance matrix using Pearson correlation as distance method.
pamy <- pam(mydist, max(mycl))
# Clusters distance matrix into as many clusters
# as obtained by tree cutting step (6).
mycolkm <- sample(rainbow(256));
mycolkm <- mycolkm[as.vector(pamy$clustering)];
heatmap(mydata, Rowv=dend colored, Colv=as.dendrogram(hc), col=my.colorFct(), scale="row",
RowSideColors=mycolkm)
# Compare PAM clustering results with hierarchical clustering
# by labeling it in heatmap color bar.
### OTRAS OPCIONES DE VISUALIZACIÓN ######
# vary parameters for most readable graph
library(cluster)
clusplot(mydata , pamy$clustering, color=TRUE, shade=TRUE,
+ lines=0)
# Centroid Plot against 1st 2 discriminant functions
library(fpc)
plotcluster(mydata,pamy$clustering)
```



En la imagen de la derecha podemos observar que los clusteres obtenidos con HC y PAM son casi idénticos y únicamente se encuentran diferencias entre 4 probes (señaladas en distinto color en el cluster PAM). Con las funciones clustplot() y plotcluster(), podemos obtener una visualización “reducida” de los cuatro clusteres.



Mediante la funcion cluster.stats() del paquete fpc tenemos un mecanismo para comparar la similitud de los clusters entre dos métodos, en este caso HC y PAM

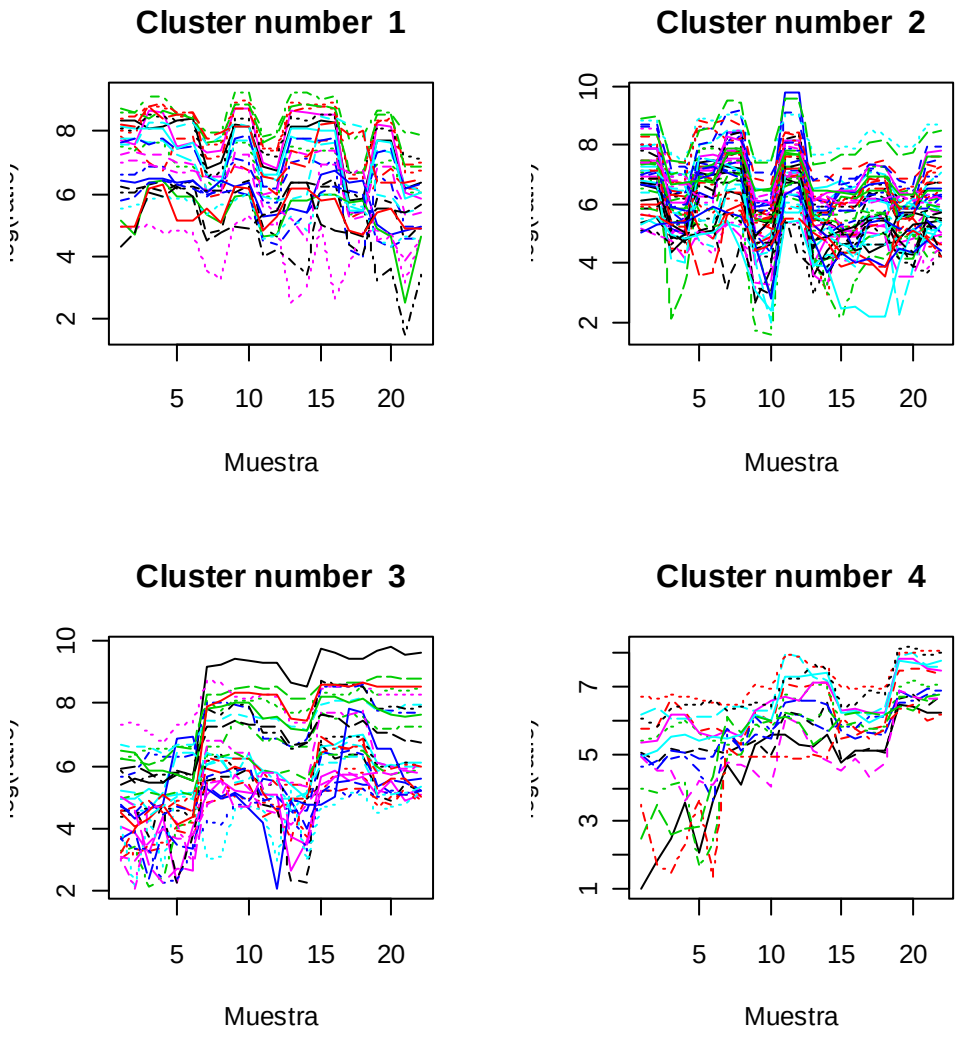
```
> library(fpc)
> cluster.stats(mydist , mycl, pamy$clustering)
$n
[1] 123

$cluster.number
[1] 4

$cluster.size
[1] 26 51 32 14

$diameter
[1] 0.9895297 1.0407853 1.0037479 1.1116232

$average.distance
[1] 0.4135617 0.3578642 0.3050080 0.2074796

$median.distance
[1] 0.4613607 0.3374755 0.2396606 0.1135572

$separation
[1] 0.25027344 0.25027344 0.09697078 0.09697078

$average.toother
[1] 1.248612 1.222880 1.174506 1.153643

$separation.matrix
          [,1]      [,2]       [,3]       [,4]
[1,] 0.0000000 0.2502734 0.44910791 0.54827204
[2,] 0.2502734 0.0000000 0.28850369 0.51617137
[3,] 0.4491079 0.2885037 0.00000000 0.09697078
[4,] 0.5482720 0.5161714 0.09697078 0.00000000

$average.between
[1] 1.205798

$average.within
[1] 0.3478962

$n.between
[1] 5316

$n.within
[1] 2187

$within.cluster.ss
[1] 10.45751

$clus.avg.silwidths
        1         2         3         4
0.5993657 0.6601984 0.5816209 0.7397117

$avg.silwidth
[1] 0.6359468

$q2
NULL

$q3
NULL

$hubertgamma
[1] 0.8226678

$dunn
[1] 0.0872335

$entropy
[1] 1.291176

$wb.ratio
[1] 0.2885196

$corrected.rand
[1] 0.9354988

$vi
[1] 0.2470952
```

Este estadístico nos puede permitir testear fácilmente las diferencias reales entre soluciones.

El algoritmo PAM es similar al K-means, pero hay un par de diferencias significativas, la más importante es que con K-means se evalúa el perfil de las similitudes con la distancia euclideana mientras que PAM puede usar cualquier medida de distancia. Basándose en la agrupación de correlación de Pearson en lugar de la distancia euclídea se debería poner más atención a la forma de perfil de expresión en lugar del Fold. change. Además de la de la matriz de distancia, el único parámetro que se tiene que proporcionar es número de agrupaciones, por lo que es fácil de lograr buenos resultados.

Veamos como es la imagen de los clusters para cerciorarnos de esto:

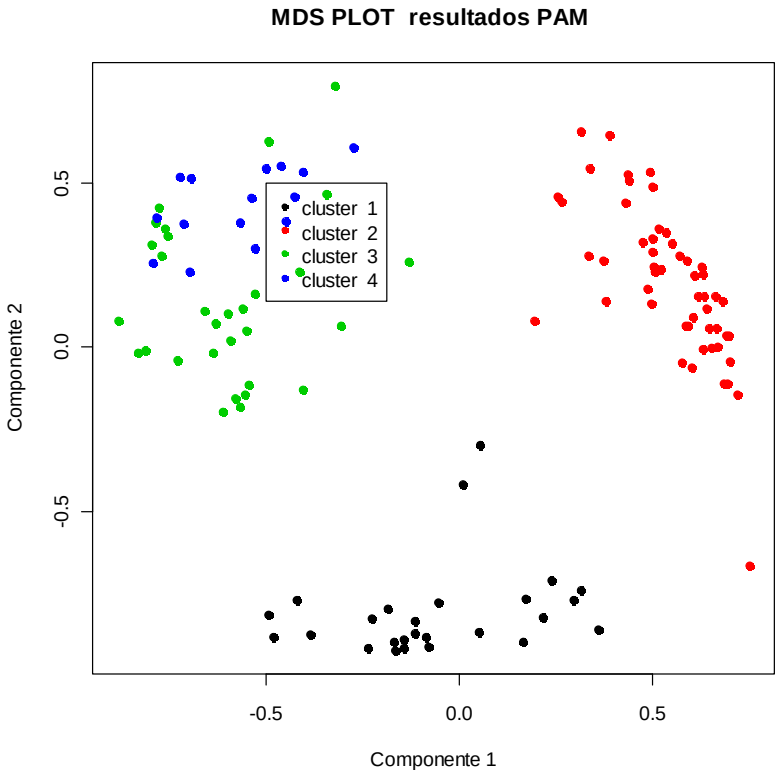
```
####Visualización de los perfiles de expresión.
number_clusters<-max(mycl)
par(mfrow=c(2,2))
x11(par)
for (loop in 1:number_clusters) {
matplot(t(log(mydata[pamy$clustering==loop,])), type="l",
      ylog=TRUE, ylab="log(ratio)", xlab="Muestra",
      main=paste("Cluster number ",loop))
}
```



Como vemos en la figura de arriba los perfiles de expresión de los cuatro clusters construidos tienen un dibujo similar para los genes dentro de un cluster. Esta es una forma visual de comprobar si la elección del numero de clusters es la adecuada y de paso comprobamos si tienen sentido biológico.

El MDS plot (que veremos más adelante) es muy práctico para evaluar la eficacia del algoritmo, así como mostrar visualmente si existe "solapamiento" entre los cluster. El siguiente código superpone los resultados de los Clusters PAM con la matriz de correlación pearson de los genes:

```
mds_pea <- cmdscale (mydist, eig = TRUE)
x11()
plot (mds_pea$points, col = pamy$clustering, xlab="Componente 1",
      ylab="Componente 2", pch=19)
title(main="MDS PLOT \ resultados PAM ")
legend_names <- paste("cluster ", as.character (seq(1,number_clusters,1)))
legend(legend=legend_names, col=1:number_clusters, x=-0.5, y=0.5,
      pch=20)
```



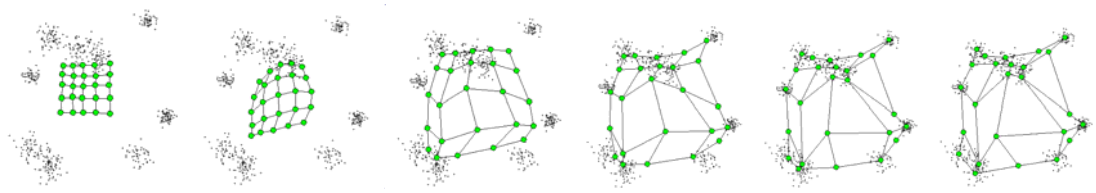
Observamos con claridad que tenemos 3 clusters muy bien definidos. Los clusters 3 y 4 tienden a mezclarse en unos solo…. ¿Seria adecuado calcular únicamente tres clusters?.

Análisis con Mapas auto-organizativos (SOM) y comparación con los HC.

Una forma más sofisticada de particionado o agrupamiento es usar mapas autoorganizativos (SOM), ya que tiene la ventaja de mostrar además las relaciones entre todos los subgrupos.

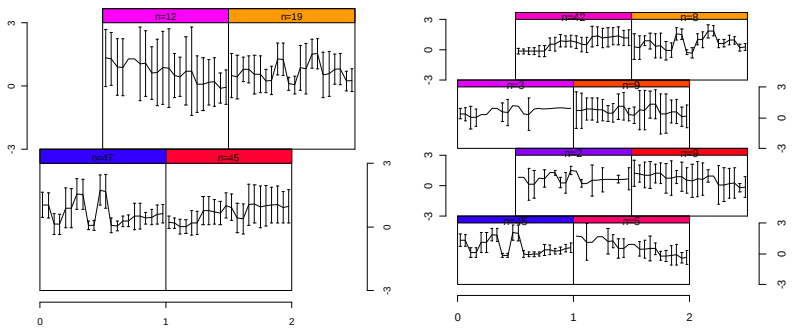
Los SOM proporcionan una técnica de visualización de datos que ayuda a entender visualmente los perfiles de expresión , sobre todo muestran so potencial en conjuntos de datos grandes y con dimensionalidades altas. Se puede decir que los SOM reducen las dimensiones de los datos y a la vez muestra las similitudes entre ellos.

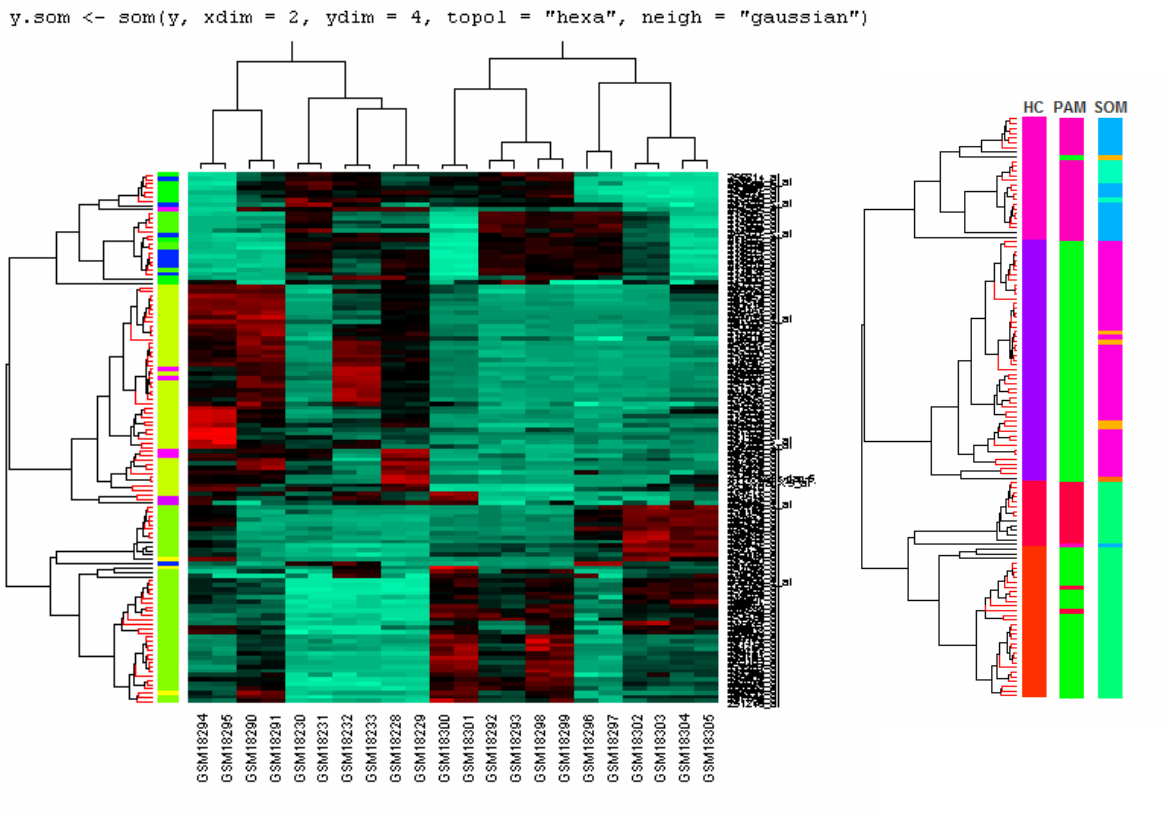
SOM es un proceso iterativo basado en redes neuronales y un proceso de entrenamiento. El input de SOM requiere como entradas la matriz de distancias, el número de nodos del gris, y una geometría del grid. El siguiente conjunto de imágenes ilustra el proceso.



```
# Mapas auto-organizativos (SOM) y comparación con clusters jerárquicos.
#################################################################
##### ANALISIS CON SOM #########
#################################################################
library(som) # Loads required library.
y <- t(scale(t(mydata))) # Center and scale data.
y.som <- som(y, xdim = 2, ydim = 3, topol = "hexa", neigh = "gaussian")
# Performs SOM clustering.
x11()
plot(y.som) # Plots results.
pdf("som.pdf");
plot(y.som);
dev.off() # Save plot to PDF: 'som.pdf'.
somclid <- as.numeric(paste(y.som$visual[,1], y.som$visual[,2], sep=""))+1
# Returns SOM cluster assignment in order of input data.
mycolsom <- sample(rainbow(256));
mycolsom <- mycolsom[somclid];
x11()
heatmap(mydata, Rowv=dend colored, Colv=as.dendrogram(hc), col=my.colorFct(), scale="row",
RowSideColors=mycolsom)
# Compare SAM clustering results with hierarchical clustering
# by labeling it in heatmap color bar.
```

Como resultado obtenemos un conjunto de cajas que representan el grid obtenido. El color de la parte superior de las cajas nos indica la similitud de un grupo frente a otro. En el interior de cada caja tenemos el perfil de expresión del cluster y las barras de error del mismo. En la imagen inferior vemos el cálculo para una geometría 2x2 (cuatro clusters) y para 3x2 (seis clusters). En el caso de 6 clusters podemos comprobar como hay cajas con similares, indicativo de que esos clusters son parecidos. En nuestro caso SOM no puede ofrecernos toda su potencia, y está más indicado para conjuntos de datos de mayor tamaño.



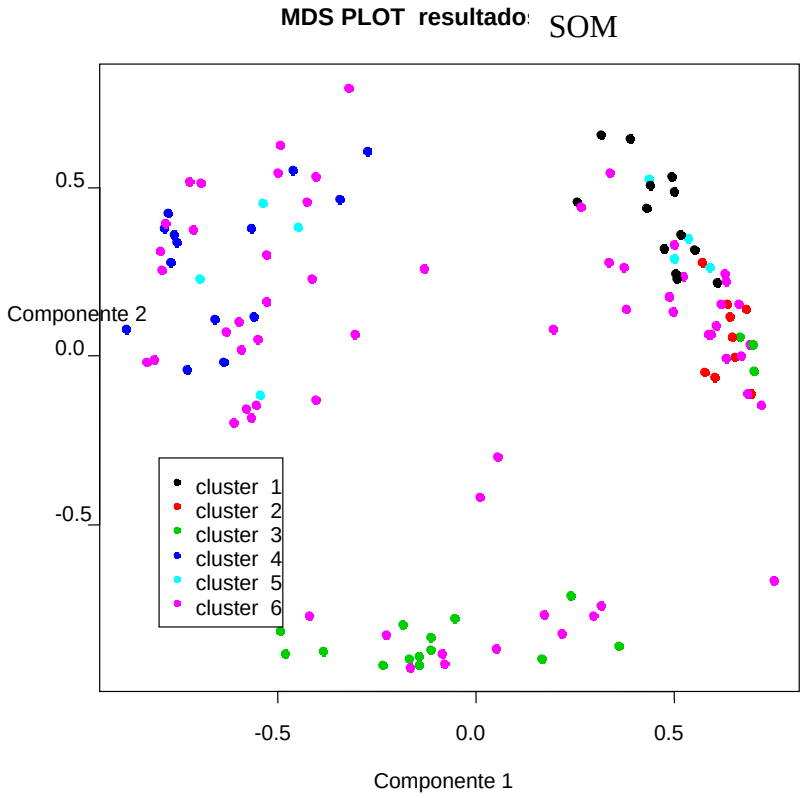


En los heat-map superior comprobamos que ocurre al aumentar el número de dimensiones a 8, y la comparación con HC y PM de SOM con un grid de 3x2

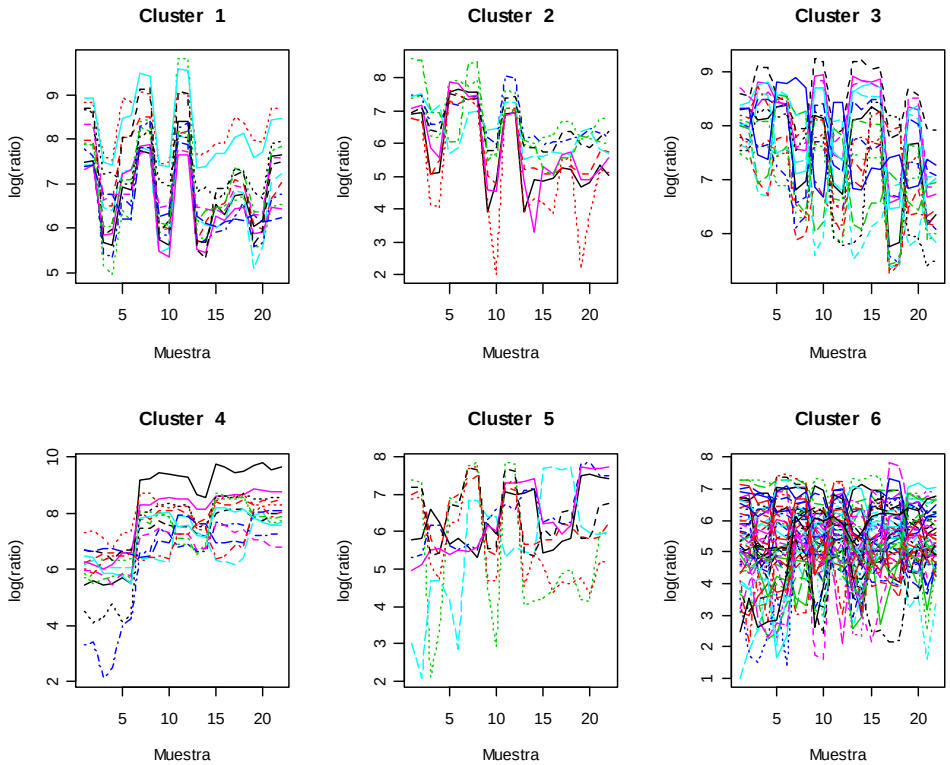
Otra opción más sencilla para hacer los cluster con SOM desde R es:

```
###GENERAMOS 6 CLUSTERS CON SOM########
som_grid <- somgrid(xdim=3, ydim=2, topo = "hexagonal")
som_results <- batchSOM (mydata, som_grid, radii=1)
som_clusters <- as.numeric (knn1 (som_results$code,mydata,0:5 ))
##VISUALIZACIÓN DEL RESULTADO##
number_clusters<-6
par(mfrow=c(2,3))
x11(par)
for (loop in 1:number_clusters) {
matplot(t(log(mydata[som_clusters==loop,])), type="l",
      ylog=TRUE, ylab="log(ratio)", xlab="Muestra",
      main=paste("Cluster ",loop))
}
mds_pea <- cmdscale (mydist, eig = TRUE)
x11()
plot (mds_pea$points, col = som_clusters, xlab="Componente 1",
      ylab="Componente 2", pch=19)
title(main="MDS PLOT \ resultados PAM ")
legend_names <- paste("cluster ", as.character (seq(1,number_clusters,1)))
legend(legend=legend_names, col=1:number_clusters, x=-0.8, y=-0.3,pch=20)
```

De esta forma podemos emplear el grafico MDS para comparar los resultados obtenidos con PAM. Lo primero que deducimos del gráfico MDS es que 6 clusters es demasiado, y como habíamos previsto con 4 o cuatro nos sobraría.



Igualmente en los perfiles de expresión observamos que SOM tiene mayor sensibilidad es decir los patrones de expresión de los primeros clusters hasta el 4 son más limpios, el sexto es ya un garabato sin sentido.



Análisis de componentes Principales

En el análisis de microarrays el PCA se puede usar con dos intenciones: Para identificar los perfiles de expresión de genes comunes (con similar perfil de expresión, básicamente como el MDS) o para comprobar los resultados de otros métodos de cluster, como hemos hecho con el MDS.

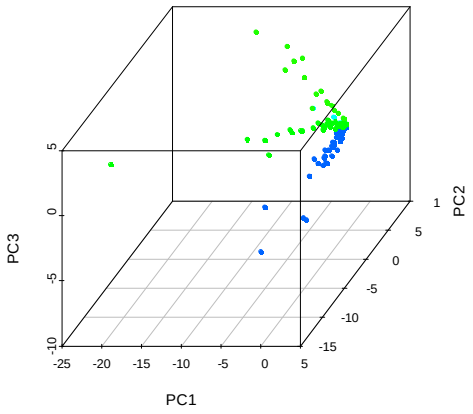
Los estadísticos dirán que El PCA es una técnica de reducción de la dimensionalidad, pero intuitivamente veremos que nos puede servir para determinar el número de factores que se esconden detrás de los datos y que explican la variabilidad de los mismos.

Matemáticamente el PCA busca una proyección por la cual el ajuste de mínimos cuadrados sea satisfactorio. Las nuevas cooredenadas (componentes) son una combinación lineal de las componentes originales. Las componentes son progresivas, es decir la primera es la que acumula mayor variabilidad y después la segunda y así sucesivamente.

El PCA tiene sentido con datos con "baja dimensionalidad" es decir cuando entre el 80% de la variabilidad puede ser explicada mediante 2 o 3 componentes. Con más componentes perdemos su capacidad visualizadora.

```
# PCA y comparación con los mapas auto-organizativos
pca <- prcomp(mydata, scale=T)
# Performs principal component analysis after scaling the data.
summary(pca) # Prints variance summary for all principal components.
library(scatterplot3d) # Loads 3D library.
scatterplot3d(pca$x[,1:3], pch=20, color=mycolsom)
# Plots PCA result in 3D. The SOM clusters are highlighted in their color.
```

La utilidad de la técnica la vemos en el grafico tridimensional de la derecha. Se ve claramente como son tres las componentes principales que explican el 91% de la variabilidad de nuestros datos, en fin esta técnica nos ayuda a "ver" que en realidad con tres clusters tenemos bastante.




```
> summary(pca) # Prints variance summary for all principal
components.
Importance of components:
                          PC1    PC2    PC3     PC4     PC5
PC6     PC7     PC8
Standard deviation     3.497 2.050 1.918 0.8824 0.6296 0.5825 0.38041 0.30930
Proportion of Variance 0.556 0.191 0.167 0.0354 0.0180 0.0154 0.00658 0.00435
Cumulative Proportion  0.556 0.747 0.914 0.9496 0.9676 0.9830 0.98957 0.99392
                           PC9     PC10     PC11     PC12     PC13     PC14     PC15
Standard deviation     0.25078 0.15818 0.12663 0.08790 0.07084 0.06050 0.05860
Proportion of Variance 0.00286 0.00114 0.00073 0.00035 0.00023 0.00017 0.00016
Cumulative Proportion  0.99678 0.99791 0.99864 0.99899 0.99922 0.99939 0.99955
                          PC16    PC17     PC18     PC19     PC20     PC21     PC22
Standard deviation     0.05068 0.0459 0.04198 0.03603 0.03046 0.02709 0.02445
Proportion of Variance 0.00012 0.0001 0.00008 0.00006 0.00004 0.00003 0.00003
Cumulative Proportion  0.99966 0.9998 0.99984 0.99990 0.99994 0.99997 1.00000
```

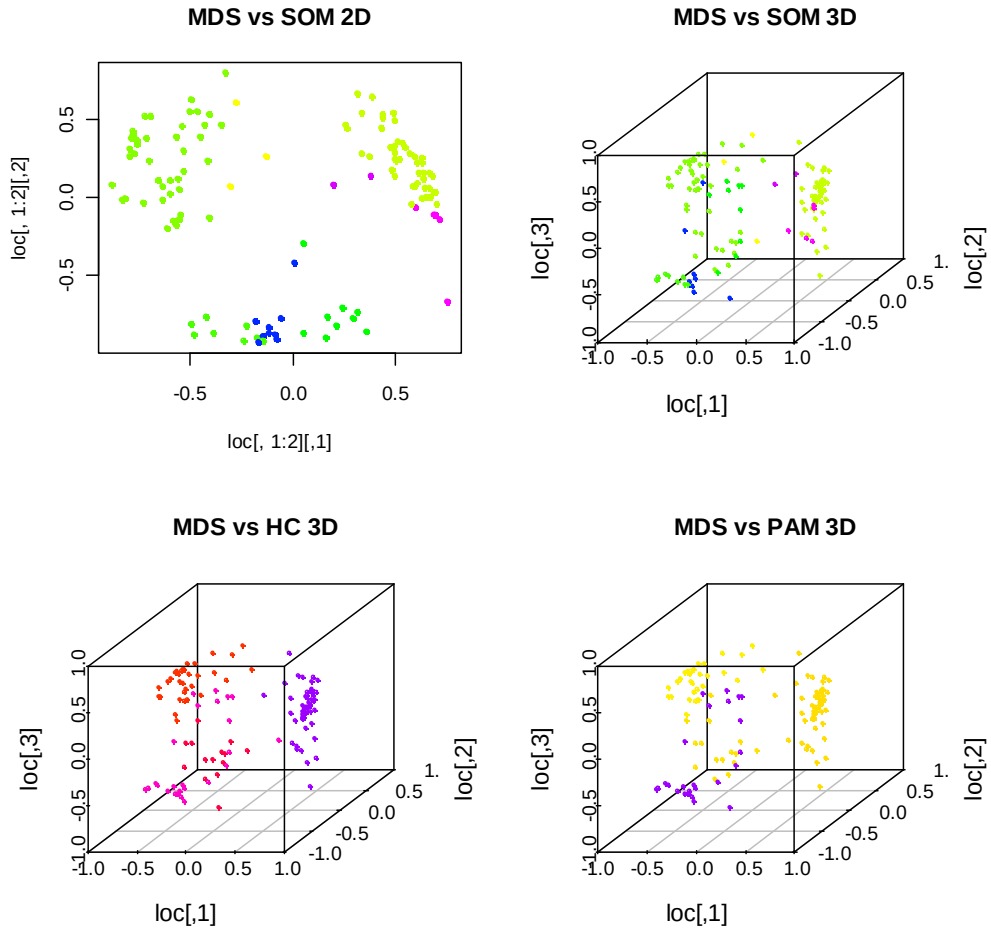
Análisis mediante escalado multidimensional. Comparamos todas las técnicas

Durante este documento, hemos ido utilizando el MDS para comparar y representar resultados. La principal aplicación del MDS es a) ayudar a otras técnicas en la representación, b) obtener la dimensión adecuada del modelo. En si mismo es muy similar al PCA, la principal diferencia es que PCA trabaja sobre las matrices de covarianza (regresión), mientras que el MDS se construye directamente sobre las matrices de distancias.

Matemáticamente los componentes se calculan "girando" los ejes de coordenadas hasta obtener la dimensionalidad reducida que minimice las distancias entre los puntos.

```
# MDS y comparación con HC, SOM y  PAM

loc <- cmdscale(mydist, k = 3)
# Performs MDS analysis and returns results for three dimensions.
x11(height=8, width=8, pointsize=12); par(mfrow=c(2,2))
# Sets plotting parameters.
plot(loc[,1:2], pch=20, col=mycolsom, main="MDS vs SOM 2D")
# Plots MDS-SOM comparison in 2D.
# The SOM clusters are highlighted in their color.
scatterplot3d(loc, pch=20, color=mycolsom, main="MDS vs SOM 3D")
# Plots MDS-SOM comparison in 3D.
scatterplot3d(loc, pch=20, color=mycolhc, main="MDS vs HC 3D")
# Plots MDS-HC comparison.
scatterplot3d(loc, pch=20, color=mycolkm, main="MDS vs PAM 3D")
# Plots MDS-KM comparison.
```

Este es el resumen en 3D de nuestros datos con HC, SOM, yPAM.



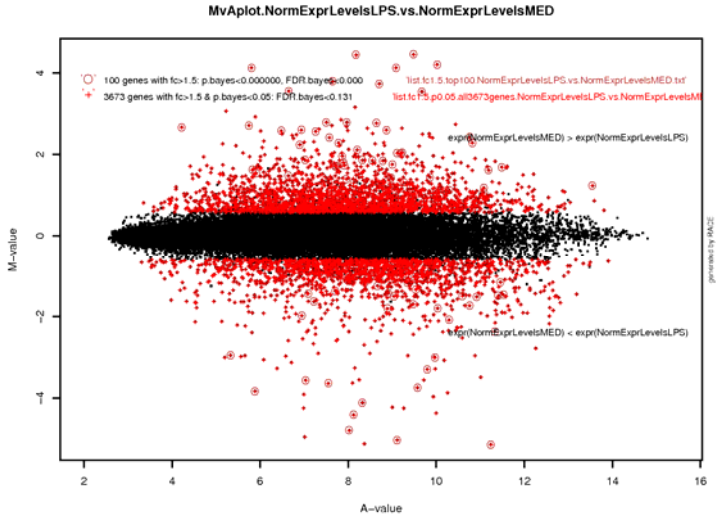
Aplicación de las técnicas de cluster a un conjunto de datos del ejemplo de la PEC1

Del conjunto de datos de:

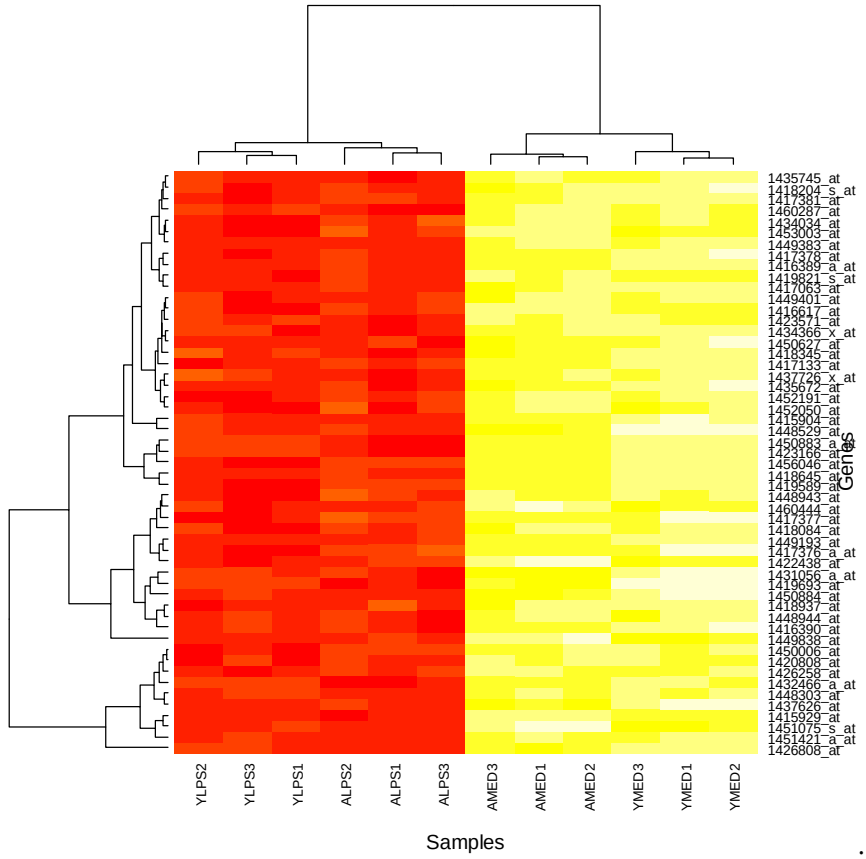
Molecular basis of age-associated cytokine dysregulation in LPS-stimulated macrophages
R. Lakshman Chelvarajan, Yushu Liu,‡ Diana Popa, Marilyn L. Getchell,
Thomas V. Getchell,,¶ Arnold J. Stromberg, and Subbarao Bondada.

Selecciono de los top100 genes los up-regulated (52 datos) según el diseño experimental de la tabla (el realizado en la PEC1):

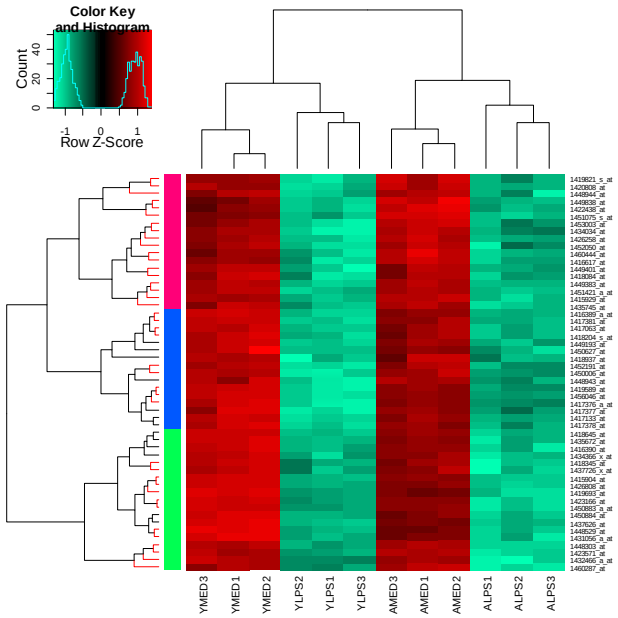
Factor Trat	
LPS	MED
4	4

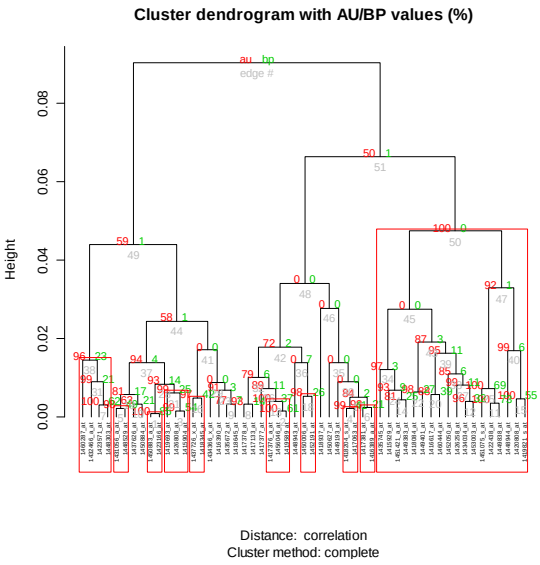


El heatmap original de los datos (función heatmap()), ya nos muestra como se separan correctamente por muestras.

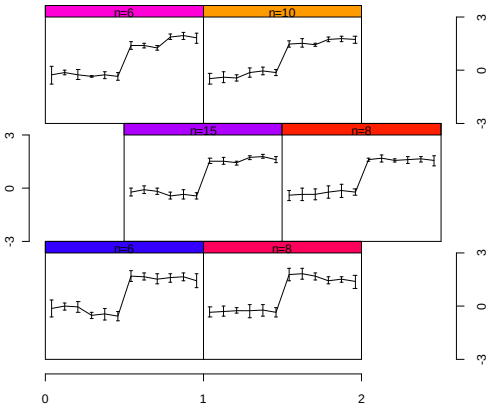


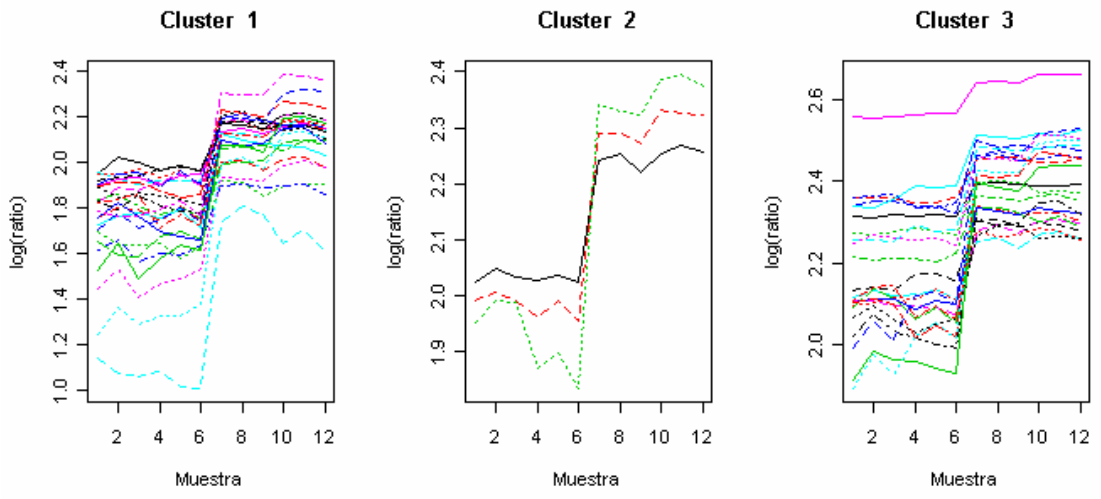
Aplicando el HC como en el ejemplo anterior, observamos la formación de tres clusters



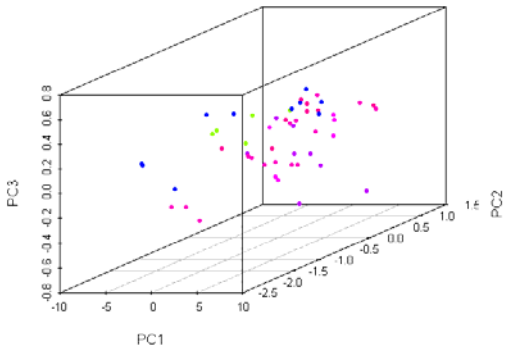


Confirmamos con MAS la validez de los tres clusters

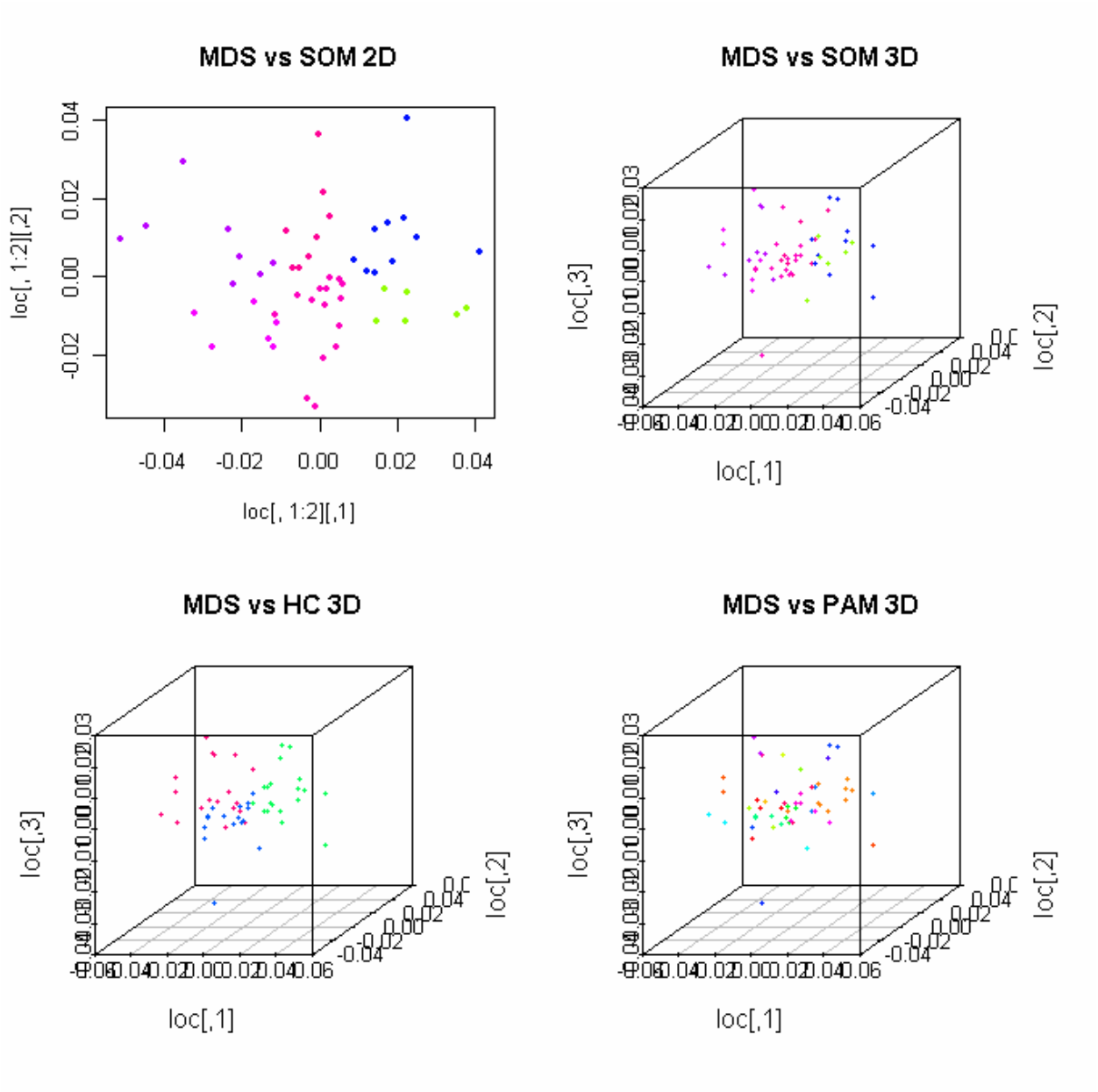




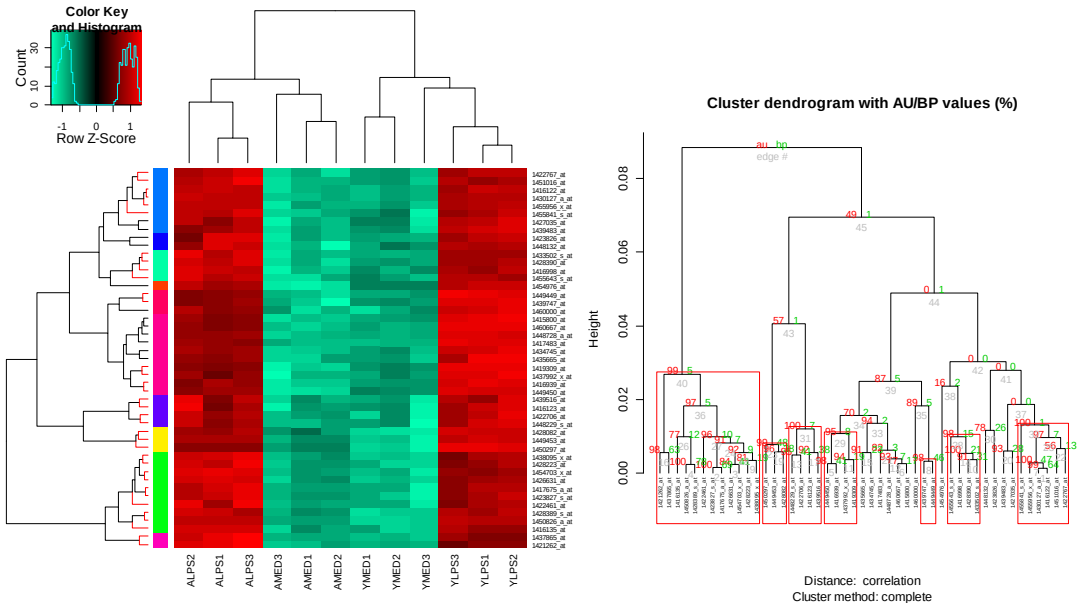
Con PCA no tenemos muy buena visualización:



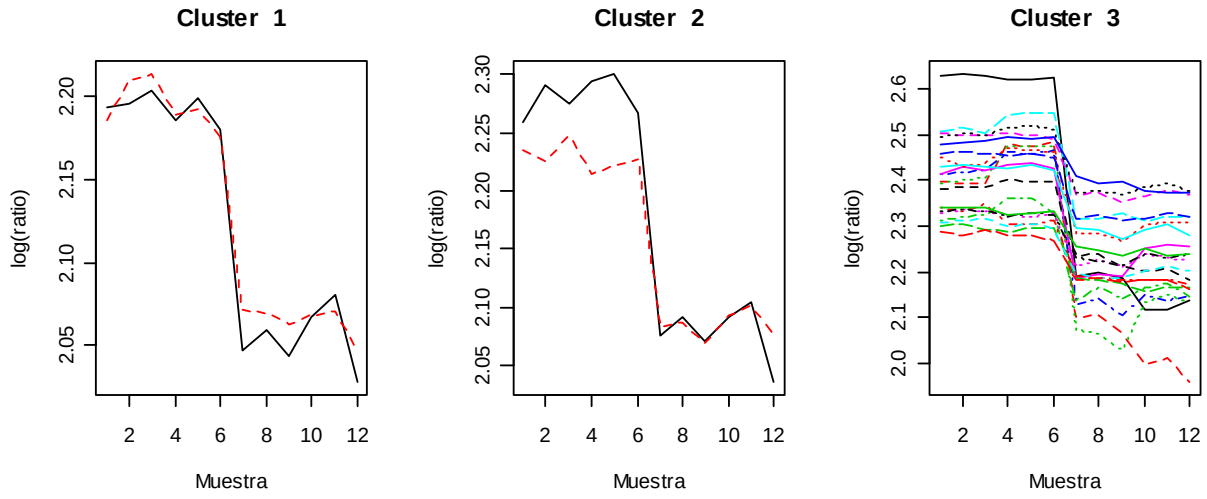
Usando MDS vemos como se visualizan correctamente en MDS vs SOM·D



Si probamos con los down-regulated, los patrones de expresión son más claros



Los patrones con MDS



Referencias interesantes.

http://www.pnas.org/content/95/25/14863.full.pdf+html

Cluster analysis and display of genome-wide expression patterns

1. Michael B. Eisen*,
2. Paul T. Spellman*,
3. Patrick O. Brown†, and
4. David Botstein*,‡

http://elfosscientiae.cigb.edu.cu/PDFs/BA/2008/25/4/BA0025RV290-300.pdf

Análisis de datos de microarreglos de ADN.
Parte II: Cuantificación y análisis de la expresión génica
Jamilet Miranda, Ricardo Bringas

Relación entre escalamiento multidimensional métrico y análisis de componentes principales

- **Autores:** María del Rosario Martínez Arias, Teresa Rivas
- **Localización:** Psicothema, ISSN 0214-9915, Vol. 3, Nº. 2, 1991 , pags. 443-451

http://blog.peltarion.com/2007/06/13/the-self-organized-gene-part-2/