Informe de práctica I
Métodos no supervisados
Clustering

Ana Medina García

16 de noviembre de 2014

Introducción

Análisis de datos de expresión diferencial

Uno de los objetivos prioritarios en el análisis de datos de expresión o de microarrays es identificar los cambios (o ausencia de ellos) en los niveles de expresión de genes y correlacionar estos cambios para identificar conjuntos de genes con perfiles similares. Los biológos intentan agrupar genes basándose en el patrón temporal de sus niveles de expresión. Mientras el uso de la agrupación jerárquica ha sido el más común en los estudios de microarrays, existen muchos otros algoritmos que podrían ser utilizados para obtener resultados en esta campo. En este estudio se ha seleccionado un método algo más complejo llamado SOM (Self-Organizing Maps, Kohonen, 1997), además del clustering jerárquico, para realizar agrupación de genes basada en perfiles de expresión, y se ha evaluado la robustez de ambos métodos y su rendimiento sobre un conocido conjunto de datos de esporulación de levadura.

Sistemas y métodos

Datos de esporulación de levadura

Consideramos los datos de microarrays del proceso de esporulación de levadura incipiente, recogidos y analizados por Chu *et al.*(1998). Al conjunto de datos se puede acceder publicamente en `http://cmgm.stanford.edu/pbrow/sporulation`. Utilizaron microarrays de ADN con el 97 % de los genes que se sabe o se predice que están involucrados en el proceso, 6118 en total. Los niveles de mRNA fueron medidos en siete instantes de tiempo durante el proceso de esporulación. Pueden encontrarse más detalles sobre el experimento en el artículo de Chu *et al.* En primer lugar, cargamos los datos de esporulación de levadura leyendo el archivo tabulado y analizamos el contenido de los mismos.

```
> #IMPORTACIÓN DE DATOS
> workingDir <- "C:/Users/usuario/Desktop/Curso14-15/Aprendizaje_Computacional/Practica1"
```

```
> setwd(workingDir)
> datos <- read.delim2("spospread.txt", header=T)
```

Selección de datos de expresión diferencial

Se ha seleccionado un subconjunto de **474 genes**, de los 6118 iniciales, eliminando aquellos genes cuyos niveles de expresión no se han modificado significativamente, determinado a través del error cuadrático medio (RMSE) de los ratios log2-transformados, utilizando un umbral de 1.6. Cargamos los datos seleccionados y guardamos el subconjunto de las columnas que nos interesan de la matriz, es decir, las de los ratios.

```
> datos.filt <- read.delim2("sporulation-filtered.txt", header=T)
> datos.filt.num <- as.matrix(datos.filt[1:474,2:8])
> dimnames(datos.filt.num)[[1]] <- datos.filt[1:474,1]
> str(datos.filt.num)
```

```
 num [1:474, 1:7] 1.164 0.956 1.455 1.699 1.303 ...
 - attr(*, "dimnames")=List of 2
  ..$ : chr [1:474] "YAL025C" "YAL036C" "YAL040C" "YDL037c" ...
  ..$ : chr [1:7] "t0" "t0.5" "t2" "t5" ...
```

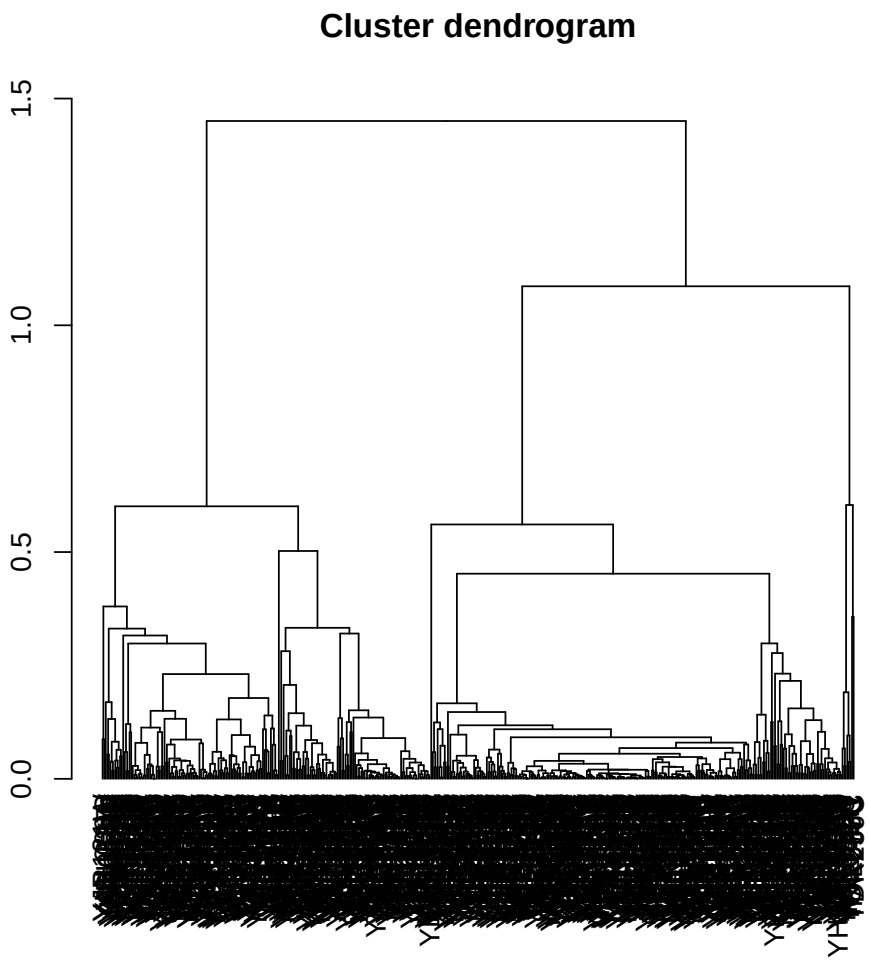
Métodos de agrupación

Para intentar conseguir una agrupación de los genes que se expresan de forma diferencial en el experimento, con el fin de establecer posibles niveles de correlación entre ellos, se han considerado técnicas de agrupamiento basadas en algoritmos no supervisados. Las siguientes técnicas de agrupamiento han sido implementadas en R, utilizando los paquetes **hclust** y **kohonen**.

Clustering jerárquico

Este algoritmo produce una jerarquía de grupos, en lugar de un determinado conjunto de grupos fijado previamente. Se ha propuesto el método aglomerativo, en el que, en un primer nivel, cada uno de los datos forma su propio cluster, y en cada uno de los subsiguientes niveles, los dos clusters más 'cercanos' se combinan para formar una más grande. Pueden utilizarse varios métodos para medir la distancia entre los clusters. El método 'average' mide la media de las distancias entre los puntos de un cluster y el otro. El método 'complete' mide la distancia entre los dos puntos más lejanos. Las distancias entre los genes se han considerado utilizando la correlación estadística entre los perfiles de expresión (coeficiente de correlación de Pearson). Visualizamos a continuación en un dendrograma una primera agrupación jerárquica de los datos utilizando el método 'average'.

```
> #CLUSTERING JERÁRQUICO (HC):
> misdatos.hc<-t(scale(t(datos.filt.num)))
> hr<-hclust(as.dist(1-cor(t(misdatos.hc),method="pearson")),method="average")
> plot(as.dendrogram(hr), main="Cluster dendrogram")
```



Fijamos un número de clusters K=7 y observamos el número de genes clasificados por grupo:

```
> mycl7<-cutree(hr,k=7)
> hc.n7.clust<-as.integer(summary(as.factor(mycl7)))
> hc.n7.clust

[1]  95 111   6 258   2   1   1
```

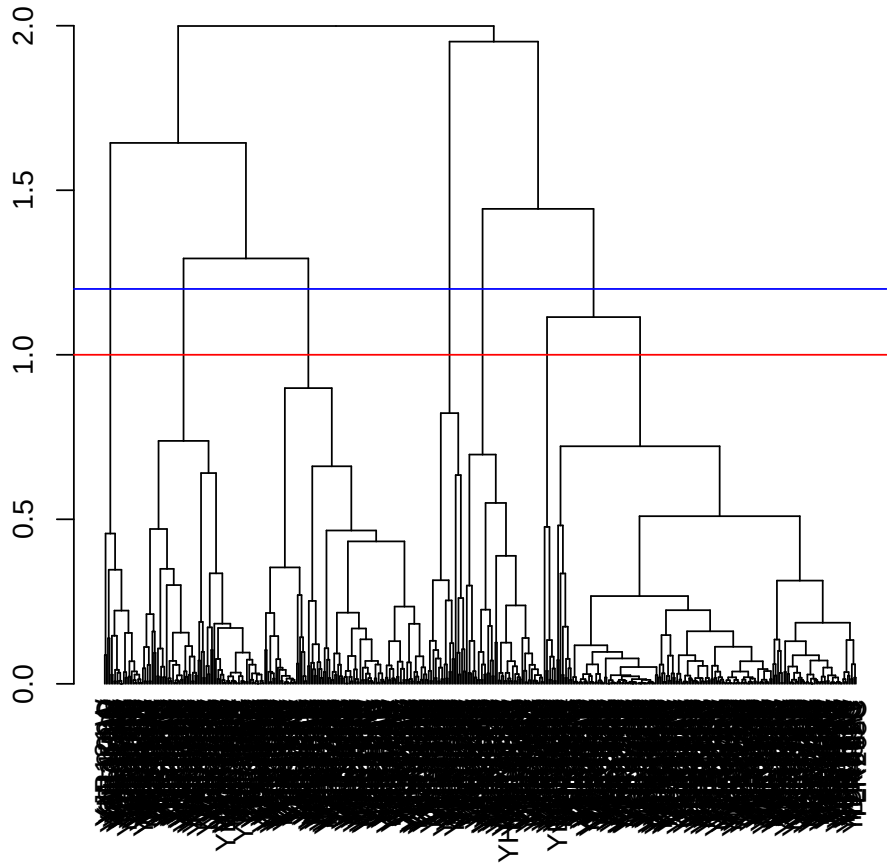
Analizando el dendrograma y los datos de genes clasificados en cada cluster, observamos que la agrupación es muy desequilibrada (hay cuatro grupos con muy pocos genes), por lo que variamos el valor K=4:

```
> mycl4<-cutree(hr,k=4)
> hc.n4.clust<-as.integer(summary(as.factor(mycl4)))
> hc.n4.clust

[1] 207   6 259   2
```

Comprobamos que al variar el número de clusters la agrupación sigue siendo muy desequilibrada, por lo que probamos con otro método diferente de aglomeración, en este caso "completo", y variando el valor de K:

```
> hc1<-hclust(as.dist(1-cor(t(misdatos.hc),method="pearson")),method="complete")
> #para k=7:
> mycl1.7<-cutree(hc1,k=7)
> hc1.n7.clust<-as.integer(summary(as.factor(mycl1.7)))
> hc1.n7.clust


[1]  77   8 103  24  24 189  49


> #para k=6:
> mycl1.6<-cutree(hc1,k=6)
> hc1.n6.clust<-as.integer(summary(as.factor(mycl1.6)))
> hc1.n6.clust


[1]  77 197 103  24  24  49


> #para k=5:
> mycl1.5<-cutree(hc1,k=5)
> hc1.n5.clust<-as.integer(summary(as.factor(mycl1.5)))
> hc1.n5.clust


[1] 180 197  24  24  49
```

Observando la distribución de las agrupaciones para los tres valores de K, podemos ver que para K=5, ya empiezan a unirse grupos de gran tamaño, lo cual consideramos que probablemente empobrezca los resultados en cuanto a su interpretación biólogica. Sin embargo cuando utilizamos K=6, un pequeño grupo de gran diferencia de tamaño con los demás, formado por tan solo 8 genes, se une al grupo más cercano. Para observar esto, pintamos el dendrograma para el cluster jerárquico con método de aglomeración "completo", marcando con una línea roja el "corte" del árbol para K=7, y con una línea azul para K=6:

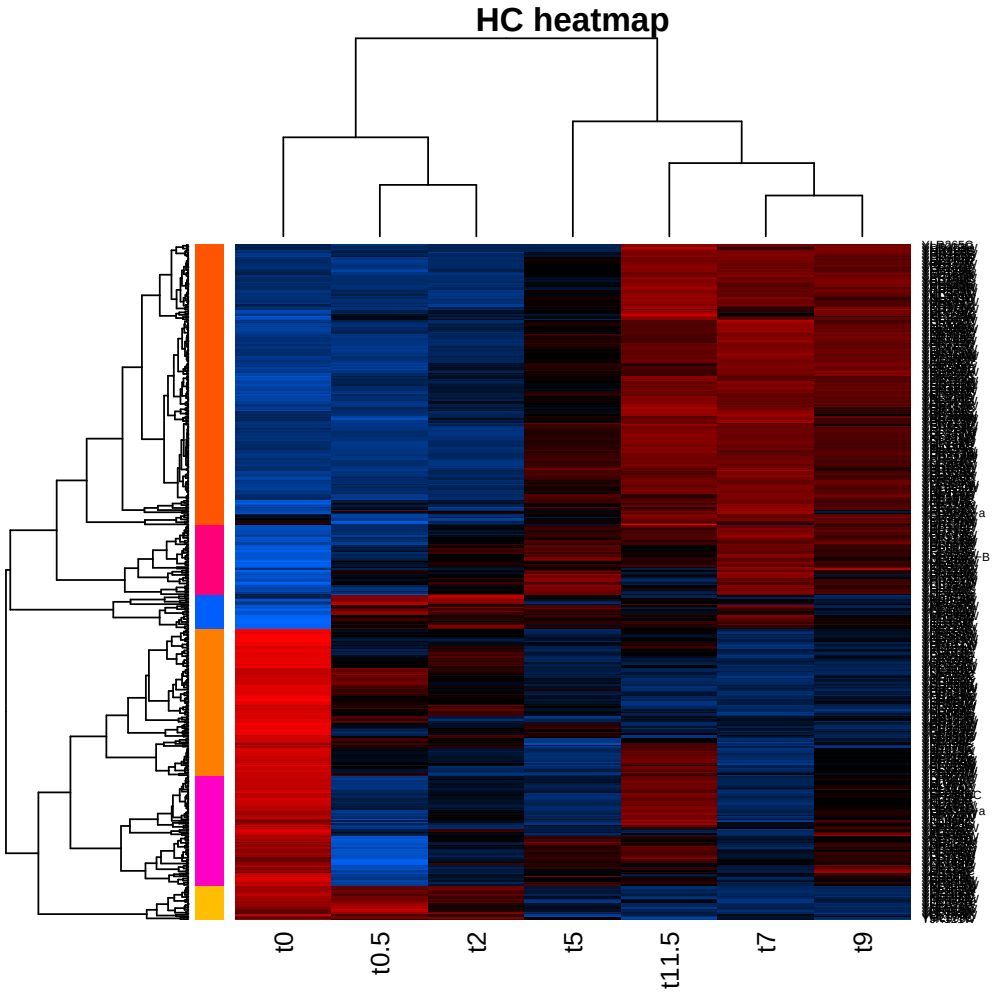


Según lo discutido anteriormente y ayudádonos de la visualización del dendrograma, consideramos que K=6 sería la elección más apropiada para este método. Para visualizar la agrupación de otra forma, utilizaremos un **heatmap**, en el que podemos observar los 6 grupos formados representados en la franja de 6 colores junto al dendrograma:

```
> #función para los colores del heatmap
> my.colorFct <- function(n=50,low.col=0.45,high.col=1,saturation=1) {
+   if(n<2)stop("n must be greater than 2")
+   n1<-n%/%2
+   n2<-n-n1
+   c(hsv(low.col,saturation,seq(1,0,length=n1)),hsv(high.col,saturation,
+                                                    seq(0,1,length=n2)))
+ }
> #heatmap del clustering jerárquico con método "complete" y K=6 en un
> mycolhc<-sample(rainbow(256))
> mycolhc<-mycolhc[as.vector(mycl1.6)]
> hc<-hclust(as.dist(1-cor(misdatos.hc,method="spearman")),method="complete")
> heatmap(datos.filt.num,Rowv=as.dendrogram(hc1),Colv=as.dendrogram(hc),
+         col=my.colorFct(low.col=0.6,high.col=1,saturation=1),scale="row",
+         RowSideColors=mycolhc,verbose=TRUE, main="HC heatmap")
```

```
layout: widths =  1 0.2 4 , heights =  1.2 4 ; lmat=
     [,1] [,2] [,3]
```

```
[1,]    0    0    4
[2,]    3    1    2
```



Redes auto-organizadas

En primer lugar realizamos el agrupamiento por red auto-organizada (SOM – Self-Organised Map), utilizando un valor de K=7 y visualizamos la disctribución de los genes en los siete grupos creados

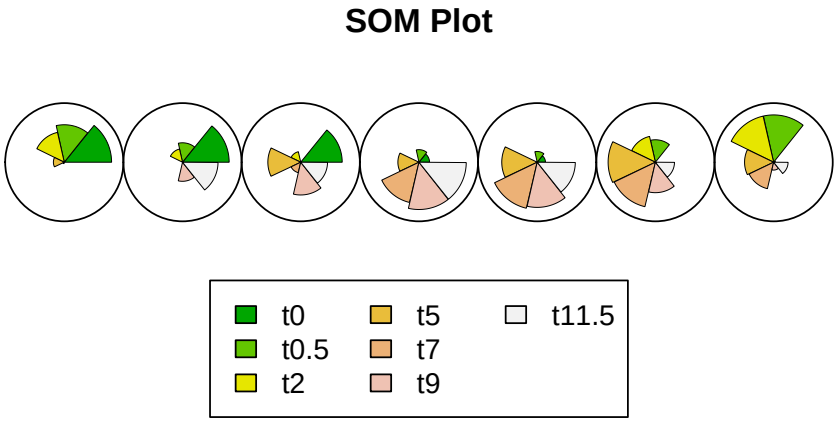
```
> #CLUSTERING POR RED AUTO-ORGANIZADA (SOM):
> library(kohonen)
> datos.filt.sc <- t(scale(t(datos.filt.num)))
> datos.som <- som(data=datos.filt.sc, grid=somgrid(xdim=7, ydim=1))
> #cuántos genes se han clasificado en cada cluster (vector de int):
> som.n.clust <- as.integer(summary(as.factor(datos.som$unit.classif)))
> som.n.clust
```

```
[1]  94  68  44  69 130  49  20
```

Como podemos observar en el vector creado, la distribución del número de genes por grupo parece bastante coherente a simple vista, teniendo en cuenta además que los estudios anteriores sobre los

datos aprobaban un valor de K cercano a 7. Por lo tanto, en este caso nos quedaremos con la agrupación en 7 clusters. A continuación mostramos los resultados en un gráfico donde cada círculo representa uno de los grupos, cada medida temporal está identificada con un color, y la variación en la expresión de los genes de cada grupo en cada instante t, se representa con el tamaño de la porción del color correspondiente y en círculo concreto.

```
> plot(datos.som, main="SOM Plot")
```



Algoritmos e implementación

Se han implementado ambas técnicas de agrupamiento utilizando R para los datos de esporulación. Chu *et al.* (1998) defendían la agrupación de los genes expresados en siete clases temporales por motivos biológicos. Siguiendo su artículo, el número de clusters se ha considerado cercano a siete para cada método. Como se esperaba, hay diferencias entre los resultados de los dos algoritmos. Para evaluar la robustez de los algoritmos, se han utilizados las medidas de estabilidad descritas a continuación.

Validación

Medidas de estabilidad

La idea detrás del enfoque de validación es que un algoritmo debe ser recompensado por la coherencia. En nuestra configuración se recogen los datos de expresión sobre todos los genes en estudio en varios instantes de tiempo (digamos l instantes). En el caso de los datos de esporulación, K será alrededor de 7 y l=7. Para cada i=1,2,...,l, se repiten los algoritmos de clustering para el conjunto de datos obtenido de eliminar la observación en el instante de tiempo i. Para poner en práctica esta idea, se han implementado funciones que realicen los métodos de agrupamiento anteriormente desarrollados pero eliminando una columna de datos -correspondiente a una medida temporal- que se le pase a la función como parámetro. Después se hacen llamadas a estas funciones eliminando en cada iteración una columna diferente. Los datos obtenidos de realizar cada método de clustering 7 veces, eliminando en cada una de ellas una columna, se guardan en una matriz que contendrá: en la primera columna, la agrupación inicial obtenida (como referencia); en las 7 columnas siguientes, las agrupaciones de validación.

```
> hc.func <- function(datos, t){
+   datos.val <- datos[,-t]
+   datos.val.sc<-t(scale(t(datos.val)))
+   hr.val<-hclust(as.dist(1-cor(t(datos.val.sc),method="pearson")),
+                  method="complete")
+   mycl.val<-cutree(hr.val,k=6)
+   return(mycl.val)
+ }
> #llamada a la función eliminando una columna en cada iteración.
> #Resultado en una matriz (la primera columna contiene el clustering
> #inicial sin eliminar instantes de t, para comparar)
> hc.val.mat<-matrix(c(mycl1.6), nrow=474, ncol=1)
> for(i in 1:7) {
+   hc.val.mat<-cbind(hc.val.mat,hc.func(datos.filt.num, i))
+ }
```

```
> #función que realice SOM eliminando una columna de tiempo
> som.func <- function(datos, t) {
+   datos.val <- datos[,-t]
+   datos.val.sc <- t(scale(t(datos.val)))
+   datos.val.som <- som(data=datos.val.sc, grid=somgrid(xdim=7, ydim=1))
+   return(datos.val.som$unit.classif)
+ }
> #llamada a la función eliminando una columna en cada iteración.
> #Resultado en una matriz (la primera columna contiene el clustering
> #inicial sin eliminar instantes de t, para comparar)
> som.val.mat<-matrix(c(datos.som$unit.classif),nrow=474,ncol=1)
> for(i in 1:7) {
+   som.val.mat<-cbind(som.val.mat,som.func(datos.filt.num, i))
+ }
```

Medida de la proporción media de no superposición (APN)

Esta medida calcula la proporción media de genes que no son clasificados en el mismo cluster por el método de agrupación cuando se eliminan las observaciones en uno de los instantes. Es decir, mide los genes que cambian de cluster. Tendrá un valor entre 0 y 1, considerándose un mejor resultado cuanto más cercano sea este valor a 0.
A modo de ejemplo de implementación, mostramos la función de la medidad APN y los valores de robustez obtenidos utilizando la misma.

```
> #función para la validación APN
> apn.func <- function (val.mat) {
+   apn<-0
+   sum<-0
+   for(j in 2:dim(val.mat)[2]) {
+     for(i in 1:dim(val.mat)[1]) {
+       if(val.mat[i,j]!=val.mat[i,1]){
+         sum<-sum+1
+       }
+     }
+   }
+   apn <- sum/length(val.mat[,2:8])
+   return(apn)
+ }
> #llamada a la función APN con los datos del método hc:
> hc.apn <- apn.func(hc.val.mat)
> #llamada a la función APN con los datos del método som:
> som.apn <- apn.func(som.val.mat)
```

Se han obtenido unos valores de APN=0.533 para el cluster jerárquico y APN=0.584 para redes auto-organizadas. Estos valores de robustez no se consideran óptimos para ninguno de los dos algoritmos.

Medida de la distancia media entre los centros (ADM)

Esta medida calcula la distancia media entre los coeficientes de expresión promedio de todos los genes que son clasificados dentro del mismo cluster cuando se eliminan las observaciones de uno de los instantes y cuando se tienen todos los datos. Es decir, mide cómo cambian los centros de los clusters creados.

Medida de la distancia media (AD)

Esta medda calcula la distancia media entre los niveles de expresión de todos los genes que son calsificados en en mismo cluster. Es decir, mide como cambian las distancias entre los genes de los clusters.

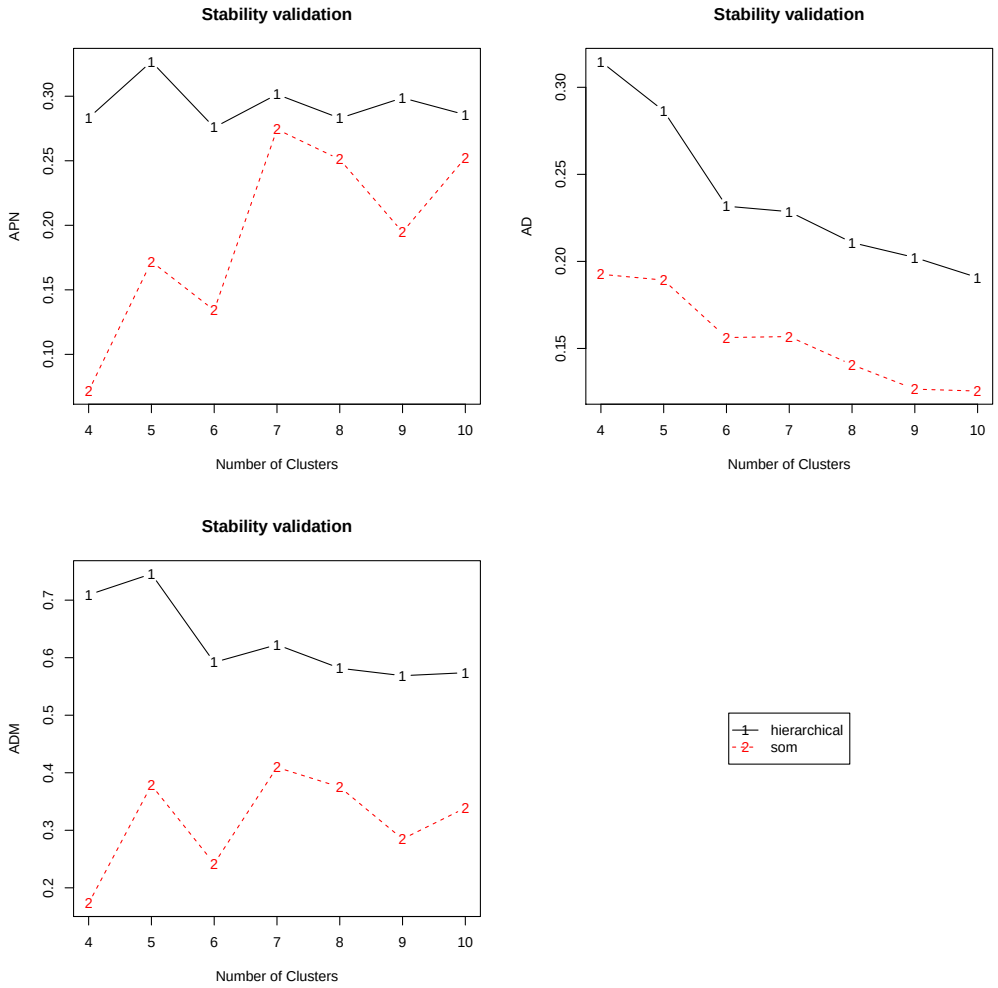
Discusión

Para analizar los resultados que obtenemos utilizando los métodos de agrupamiento antes descritos, utilizaremos el paquete **clValid**, gracias al cual podemos analizar tanto la robustez de los algoritmos (validación de estabilidad) como el rendimiento de los mismos (validación interna), teniendo en cuenta además la variación en el número de clusters.

En cuanto a la robustez, tendremos en cuenta las medidas de estabilidad descritas en la sección anterior.

```
> #VALIDACIÓN UTILIZANDO EL PAQUETE clValid:
> library(clValid)
> #Estabilidad (APN, AD, ADM):
> stab.valid <- clValid(datos.filt.num, 4:10, clMethods=c("hierarchical","som"),
+                       validation="stability", metric="correlation",
+                       method="complete")
> summary(stab.valid)


Clustering Methods:
 hierarchical som

Cluster sizes:
 4 5 6 7 8 9 10

Validation Measures:
                        4      5      6      7      8      9     10

hierarchical APN  0.2832 0.3268 0.2758 0.3014 0.2830 0.2987 0.2857
             AD   0.3148 0.2864 0.2317 0.2286 0.2108 0.2022 0.1907
             ADM  0.7093 0.7457 0.5919 0.6226 0.5817 0.5687 0.5739
             FOM  0.5552 0.5425 0.5290 0.5263 0.5136 0.5080 0.4953
som          APN  0.0716 0.1720 0.1341 0.2744 0.2511 0.1948 0.2522
             AD   0.1926 0.1893 0.1563 0.1568 0.1407 0.1266 0.1255
             ADM  0.1730 0.3785 0.2421 0.4094 0.3751 0.2848 0.3388
             FOM  0.4437 0.4280 0.4100 0.4104 0.3982 0.3731 0.3678

Optimal Scores:

    Score  Method Clusters
APN 0.0716 som    4
AD  0.1255 som    10
ADM 0.1730 som    4
FOM 0.3678 som    10
```

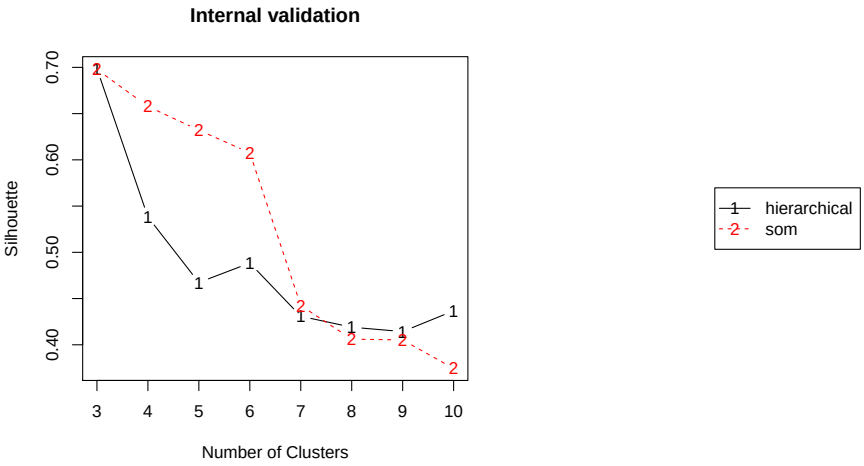
A continuación mostramos gráficamente los valores de las medidas de estabilidad (APN, AD, ADM) para ambos métodos de agrupamiento con una variación de K=2,...,12.



Como podemos observar en los gráficos, el algoritmo de SOM obtiene, en general, mejores valores de medida de robustez. En cuanto a la elección del número de clusters, considerando las tres medidas de estabilidad y los dos métodos estudiados, podríamos considerar K=6 la mejor elección.

Para analizar el rendimiento de los algoritmos sobre el grupo de estudio, utilizaremos la medida de validación interna "Silhouette Width", mostrando un gráfico con los valores para ambos métodos de agrupamiento y con una variación de K=4,...,10. Esta forma de validación combina medidas de cohesión y de separación de clusters. "Silhouette Width" es la media entre los valores de Silhouette de cada observación. El valor Silhouette mide el grado de confianza en una agrupación y toma valores en el intervalo [-1,1], donde las observaciones bien agrupadas tendrán un valor cercano a 1, y las observaciones mal agrupadas tendrán un valor cercano a -1.

```
> #Interna (Silhuette):
> intern.valid <- clValid(datos.filt.num, 3:10, clMethods=c("hierarchical","som"),
+                         validation="internal", metric="correlation", method="complete")
> par(mfrow=c(1,2))
> plot(intern.valid, measures=measNames(intern.valid)[3],legend=FALSE)
> plot(nClusters(stab.valid),measures(stab.valid,"APN")[,,1],type="n",
+      axes=F, xlab="",ylab="")
> legend("center", clusterMethods(stab.valid), col=1:9, lty=1:9, pch=paste(1:9))
```



Como podemos observar en el gráfico, para ambos métodos el rendimiento parece empeorar cuando aumenta el número de clusters. Teniendo en cuenta estos valores y los de robustez anteriormente estudiados podríamos considerar K=6 un número apropiado de clusters para la agrupación de nuestro estudio.