TP: Classification non supervisée avec R

Résumé

Travaux pratiques sur la classification non supervisée (CAH, kmeans) de données transcriptomiques avec R.

Introduction

Les données sont toujours celles concernant les régimes de 40 souris réparties selon deux facteurs :

- Génotype (2 modalités) : les souris sont soit de type sauvage (wt) soit génétiquement modifiées (PPAR) ; 20 souris dans chaque cas.
- Régime (5 modalités) : les 5 régimes alimentaires sont notés `ref`, `efad`, `dha`, `lin`, `tournesol` ; 4 souris de chaque génotype sont soumises à chaque régime alimentaire.

L'ACP a mis en évidence les gènes dont les différences d'expression distinguent les génotypes et l'AFD a permis de distinguer les régimes. L'objectif est maintenant de regrouper les gènes en classes homogènes afin d'aider le biologiste à les associer à des fonctions ou métabolismes de la lipogénèse.

Attention, les gènes sont considérés comme des variables et donc la distance à prendre en compte entre ceux-ci peut être adaptée en utilisant, en plus de celle euclidienne classique entre vecteurs, celles considérant la corrélation. En conséquence, la représentation des classes doit se faire alors dans les graphiques d'un MDS adapté à ces distances. En revanche, dans le cas de la distance euclidienne usuelle, ACP et MDS donnent des représentation équivalentes.

1 Classification ascendante hiérarchique

```
# Calcul des distances entre gènes
rS = cor(Exprs)
dS=as.dist(1-rS) # corrélation
dS2=as.dist(sqrt(1-rS**2)) # carré de la corrélation
# distance euclidienne usuelle entre gènes
d=dist(t(Exprs))
dN=dimnames(Exprs)[[2]] # noms des gènes

# CAH avec la corrélation linéaire
hc.ds <- hclust(dS,method="ward")
plot(hc.ds)  # dendogramme
# choix du nombre de classes
plot(hc.ds$height[118 :100],type="b")

# CAH avec le carré de la corrélation
hc.ds2 <- hclust(dS2,method="ward")
plot(hc.ds2)  # dendogramme
# choix du nombre de classes
plot(hc.ds2$height[118 :100],type="b")

# CAH avec la distance euclidienne
hc.d <- hclust(d,method="ward")
plot(hc.d)  # dendogramme
# choix du nombre de classes
plot(hc.d$height[118 :100],type="b")
```

Tout ce qui suit est calculé pour la CAH sur les distances euclidiennes mais peut être refait sur les autres classifications. Ce choix est fait afin de comparer cette classification avec celle obtenue par réallocation dynamique.

```
# découpage de l'arbre en 6 classes
classif.6G <- cutree(hc.d,k=6)
# répartition des gènes en classes
sort(classif.6G)
# gènes de la 2ème classe
names(classif.6G[classif.6G==2])
```

Représentation de la classification dans les coordonnées du MDS. Comme la distance utilisée est celle euclidienne, la représentation obtenue est équivalente à celle fournie par une ACP.

```
coul = classif.6G
mds=cmdscale(d,k=2)
plot(mds, type="n", xlab="Dimension 1",
      ylab="Dimension 2", main="CAH euclid")
text(mds,dN,col=coul)
```

Malgré le nombre de gènes, cette représentation est plus lisible ou au moins complémentaire au dendrogramme. Remarquer que la classification est essentiellement construite sur les valeurs prises par les gènes sur le premier axe. Comparer avec une classification obtenus avec un autre critère :

```
classif.7G = cutree(hc.ds2,k=7)
coul1 = classif.7G
mds2=cmdscale(dS2,k=2)
plot(mds2, type="n", xlab="Dimension 1",
     ylab="Dimension 2", main="CAH corr2")
text(mds2,dN,col=coul1)
```

Une autre dimension serait peut-être nécessaire afin de mieux discriminer les classes.

2 Réallocation dynamique ou `kmeans`

Attention, cette technique de classification s'applique uniquement à des variables quantitatives, pas à un tableau de distances. La distance considérée entre les gènes est alors celle euclidienne classique.

```
km.genes=kmeans(t(Exprs),centers=6)
# comparaison des deux classifications
table(classif.6G,km.genes$cluster,
     dnn=c("tree","kmeans"))
```

Remarquer la bonne cohérence entre les deux classifications. Ce n'est pas une règle générale. Cette cohérence peut être améliorée en initialisant `kmeans` par les barycentres des classes obtenues par CAH. C'est une pratique très courante qui revient à améliorer une CAH par un algorithme de réallocation dynamique.

```
mat.init.km.genes=matrix(nrow=6,ncol=40)
for (i in 1 :6) # calcul des barycentres des classes
mat.init.km.genes[i,]=apply(t(Exprs)
     [classif.6G==i,],2,mean)
# kmeans après initialisation par les barycentres
km.genes.init=kmeans(t(Exprs),centers=mat.init.km.genes)
# comparaisons des classifications
table(classif.6G,km.genes.init$cluster,
     dnn=c("tree","kmeans"))
coul1 = km.genes$cluster
représentation dans les axes du MDS
plot(mds, type="n", xlab="Dimension 1",
     ylab="Dimension 2", main="Kmeans euclid")
text(mds,dN,col=coul)
```

3 Double classification

Enfin, il est d'usage sur ce type de données de proposer une double classification hiérarchiques des lignes et des colonnes. Pas toujours très simple à interpréter mais cela fait de jolis graphiques en couleur :

```
# définition de la fontion utilisée pour la
# classification, il est dommage que cela
# soit la même pour lignes et colonnes
lf = function(d) hclust(d, method="ward")
# double classification sur données brutes
heatmap(as.matrix(Exprs),hclustfun=lf)
# la même en changeant la normalisation
heatmap(scale(as.matrix(Exprs),scale=FALSE),
        hclustfun=lf)
```