TP MISO: analyse différentielle de données transcriptomiques

Guillemette Marot

1 Installation d'un package de Bioconductor

La procédure d'installation de tout package Bioconductor est disponible sur la page https://www.bioconductor.org/install/

Les packages de base de Bioconductor s'installent en utilisant les commandes suivantes:

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install()
```

Le package `limma` s'installe ensuite en utilisant la commande:

```
BiocManager::install("limma")
```

Pour les curieux, un petit manuel de programmation S4 est disponible à https://cran.r-project.org/doc/contrib/Genolini-PetitManuelDeS4.pdf

Quelques commandes à connaître:

```
class(nomobjet)
class ?nomclasse
new(Class=nomclasse)
slotNames #donne le nom des attributs
getSlots #donne le nom des attributs et leur type
getClass #donne le nom des attributs et leur type, mais aussi les héritiers et les ancêtres
showMethods(class="nomclasse")
getMethod("nommethode","nomclasse")
```

2 Utilisation de limma

Un tutoriel complet de `limma` est disponible en lançant les commandes suivantes:

```
library(limma)
limmaUsersGuide()
```

Le package `limma` effectue des t-tests modérés, où la modélisation de la variance au dénominateur repose sur une approche bayésienne empirique. Cela permet de détecter plus de gènes différentiellement exprimés dans les études où il y a peu d'individus, tout en limitant le nombre de faux positifs.

Principes:

- théorème de Bayes
$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}$$
- empirique: a priori calculés à partir des données observées

$$\tilde{\theta_g} = \widehat{\theta_c} + b(\widehat{\theta_g} - \widehat{\theta_c})$$

avec θ_c estimateur commun, θ_g estimateur gène à gène, b facteur de "shrinkage".

Reprenons l'exemple du jeu de données de puces à ADN traité dans le TP "tests multiples".

```
microarray.data.url  <- "http://pedagogix-tagc.univ-mrs.fr/courses/ASG1/data/marrays"
expr.url <- file.path(microarray.data.url, "GSE13425_Norm_Whole.txt")
pheno.url<-file.path(microarray.data.url,"phenoData_GSE13425.tab")
expr.matrix<- read.table(expr.url,sep="\t", header = T, row.names = 1)
head(expr.matrix)
pheno.matrix <-  read.table(pheno.url,sep="\t", header = T, row.names = 1)
head(pheno.matrix)
```

L'analyse différentielle avec limma s'effectue de la façon suivante:

```
library(limma)
groups <- pheno.matrix$sample.labels
class(groups)#on vérifie que c'est un facteur
design.matrix<-model.matrix(~0+groups)
colnames(design.matrix)<-levels(groups)
design.matrix
fit<-lmFit(expr.matrix,design.matrix)
contrast.matrix<-makeContrasts(Bh-Bo,levels=groups)
fit.contrast<-contrasts.fit(fit, contrast.matrix)
fit2 <- eBayes(fit.contrast)
names(fit2)
```

3 Histogramme des p-values brutes

Si les hypothèses du modèle statistique sont vérifiées, alors l'histogramme des p-values brutes doit montrer une distribution uniforme pour les gènes non différentiellement exprimés. En fonction du nombre de gènes différentiellement exprimés, on observe un pic plus ou moins grand à gauche.

Tracer un histogramme des p-values brutes puis ajuster les p-values brutes avec la méthode de Benjamini-Hochberg. Combien de gènes sont différentiellement exprimés, avec un FDR de 5%?

```
hist(fit2$p.value,nclass=100)
adjpval<-p.adjust(fit2$p.value,method="BH")
length(which(adjpval<=0.05))
```

4 Volcano plot

Le volcano plot permet de repérer des gènes qui présentent une différence d'expression à la fois statistiquement significative et importante en pratique.

```
de<-rep("NO",length(adjpval))
de[which(fit2$coefficients>=1&adjpval<=0.05)]<-"UP"
de[which(fit2$coefficients<=-1&adjpval<=0.05)]<-"DOWN"
res<-data.frame(fit2$coefficients,fit2$p.value,de)
colnames(res)<-c("LogFC","PValue","DE")
library(ggplot2)
g <- ggplot(data<-res,
```

```
            aes(x=LogFC,y=-log10(PValue),col=DE)) +geom_point()+theme_minimal()
g + scale_color_manual(values=c("blue", "black", "red"))
```