PEC 1

Jordi Sansó Sastre

2024-10-30

Contents

Selecció del *dataset* de metabolòmica

Per tal de començar amb la PEC, es selecciona el *dataset* amb títol **2023-UGrX-4MetaboAnalystTutorial** que es troba en el repositori de github metaboData.

Per tal d'obtenir les dades del *dataset* seleccionat, es descarrega l'arxiu *ST000002_AN000002.txt* pitjant sobre el botó *Dowload raw file*. A més, en la carpeta on es troba aquest arxiu, trobem un arxiu anomenat *description.md* que ens indica informació sobre el *dataset* i alguns canvis que s'han de fer per preparar les dades per després poder dur a terme l'anàlisi.

Estructura de l'arxiu *ST000002_AN000002.txt*

Tal com indica l'arxiu *description.md* que es troba en la carpeta d'on s'obtenen les dades (2023-UGrX-4MetaboAnalystTutorial), les dades es troben en un arxiu .txt que està estructurat en dos blocs:

- Primer bloc (files 1:70): conté informació sobre l'estudi i les mostres.
- Segon bloc (files 71:363). Conté les dades en format rectangular i informació sobre els metabolits.

Preparació de les dades per l'anàlisi

Per tal d'obtenir les dades per l'anàlisi, es tenen en compte diverses consideracions que es comenten el l'arxiu *description.md*. Aquestes consideracions són:

- Eliminació del primer bloc de l'arxiu (files 1:71).
- Eliminació de la informació sobre els metabolits (files 216:363)
- Canvi de les etiquetes dels factos a *Before* i *After*.
- Afegir la lletra B o A als noms de les mostres per indicar si pertanyen al grup *Before* o *After*, respectivament.

Creació del contenidor del tipus `SummarizedExperiment`

La classe `SummarizedExperiment` es la classe més important de `Bioconductor` per dades experimentals en forma de matriu. Aquesta classe, pot emmagatzemar múltiples matrius de dades experimentals de dimensions indèntiques, amb metadades associades sobre les files/gens/transcripcions/o altres mesures (`rowData`), columnes/fenotips de mostres o dades clíniques (`colData`) y l'experiment en general (`metadata`).

Abans de començar amb la creació de l'objecte `SummarizedExperiment`, es carrega l'arxiu .txt que s'ha descarregat anteriorment que conté la informació sobre el conjunt de dades.

```
dades <- read.table("ST000002_AN000002.txt", header=TRUE, sep="\t", row.names=1, nrows=143, skip=71)
```

Tal com podem observar, `dades` només conté les 143 files de l'arxiu .txt posteriors a les 71 primeres línies, les quals, corresponen a les dades de l'experiment. Una veagda obtingudes les dades de l'experiment, es duen a terme es canvis que es mencionen anteriorment, els quals, corresponen a l'etiquetatge de les mostres.

```
factors <- dades[1,]
metabolites <- rownames(dades)[-1]
samples <- colnames(dades)

n <- length(samples)
sampleNames <- vector("character", n)
factorLabels <- vector("character", n)

for (i in 1:n) {

  if (grepl("After", factors[i])) {
    sampleNames[i] <- gsub(pattern="LabF", replacement="A", x=samples[i])
    factorLabels[i] <- c("After")
  }

  if (grepl("Before", factors[i])) {
    sampleNames[i] <- gsub(pattern="LabF", replacement="B", x=samples[i])
    factorLabels[i] <- c("Before")
  }
}

colnames(dades) <- sampleNames
factorLabels <- sapply(factorLabels, function(x) as.factor(x))
dades[1,] <- factorLabels
```

Es poden extreure les dades en un arxiu .txt utilitzant la funció `write.table()`.

```
write.table(dades, file="dades.txt", sep="\t", row.names=FALSE)
```

Assay - Dades experimentals

L'*assay* corrspon a una matriu amb valors en format numèric on les files representen els metabolits i les columnes les mostres. Cada valor de la matriu representa l'altura de pic del metabolit en la mostra corresponent.

```
exprs <- dades[-1,] %>% sapply(function(x) as.numeric(as.character(x)))

colnames(exprs) <- sampleNames
rownames(exprs) <- metabolites
```

colData - Informació de les mostres

El *colData* és un `data.frame` que conté informació sobre les mostres. Es crea utilitzant el vector amb la informació sobre a quin grup pertany cada mostra i el vector amb el nom de les mostres.

```
colData <- data.frame(sampleNames, factorLabels)
```

rowData - Informació de les files

El *rowData* conté informació sobre els metabolits. Per tal de crear l'objecte `rowData`, es carreguen les files de l'arxiu .txt que contenen la informació sobre els metabolits tal com es mostra a continuació:

```
rowData <- read.table("ST000002_AN000002.txt", header=TRUE, sep="\t", nrows=142, skip=218)
```

Creació de l'objecte `SummarizedExperiment` combinant les tres parts

Abans de crear l'objecte de la classe `SummarizedExperiment`, es comprova que l'ordre de les files i les columnes en els objectes `rowData` i `colData` es corresponguin a l'ordre de les files i les columnes de la matriu de dades, respectivament.

```
stopifnot(rownames(exprs) == rowData$metabolite_name)
stopifnot(colnames(exprs) == colData$sampleNames)
```

Un cop feta la comprovació, es crea l'objecte de classe `SummarizedExperiment` tal com es mostra en la següent línia de codi:

```
se <- SummarizedExperiment(assay=list("exprs"=exprs),
                           colData=colData,
                           rowData=rowData)
```

Finalment, es pot guardar l'objecte `se` en un arxiu .Rda utilitzant la funció `save()`.

```
save(se, file="SummarizedExperiment.Rda")
```

Addicionalment, podem guardar les metadades en un archiu markdown (.md), el qual, es construeix a partir dels objectes `colData` i `rowData`. Per tal de fer-ho, utilitzem les funcions `file()`, `writeLines()` i `close()`.

```
metadades <- file("metadades.md", "w")

writeLines("# Metadades ST000002_AN000002\n\n", metadades)

writeLines("## Informació sobre les mostres\n\n", metadades)
writeLines(as.character(kable(colData, format = "markdown")), metadades)
```

```
writeLines("\n\n", metadades)

writeLines("## Informació sobre els metabolits\n\n", metadades)
writeLines(as.character(kable(rowData, format = "markdown")), metadades)

close(metadades)
```

Obtenció de l'objecte `SummarizedExperiment` utilitzant `metabolicWorkbenchR`

Utilitzant el paquet `metabolomicssWorkbenchR`, es poden descarregar directament els objectes de classe `SummarizedExperimnt` dels *datsets* presents en la base de dades Metabolomics Workbench. Per tal de fer-ho, es pot utilitzar la funció `do_query()`.

```
se1 <- do_query(context="study",
                input_item="study_id",
                input_value="ST000002",
                output_item="SummarizedExperiment")
```

Exploració del *dataset*

Processament i normalització de les dades

En primer lloc, observem les dimensions del conjunt de dades i comprovem que no hi hagi *missing values* (NAs). Per tal de fer-ho utilitzem les funcions `dim()` i `anyNA()`, respectivament.

```
dim(assay(se))
```

```
## [1] 142  12
```

Tal com podem observar, el nostre conjunt de dades, conté 142 files i 12 columnes.

Utilitzant la funció `anyNA()` comprovem que no hi hagi valors faltants.

```
anyNA(assay(se))
```

```
## [1] FALSE
```

Tal com podem observar, no hi ha NAs en el conjunt de dades.

Una vegada comprovat que no hi ha NAs, podem obtenir un resum estadístic de les diferents mostres i dels diferents metabolits per tal d'observar els valors màxims i mínims, el primer i tercer quartil i la mitjana i la mediana. Per tal de fer-ho, s'utilitza la funció `summary()`.

```
summary(assay(se))
```

```
##     A_684508          A_684512          A_684516          A_684520
##  Min.   :     95   Min.   :    336   Min.   :     98   Min.   :    186
##  1st Qu.:   1261   1st Qu.:   2815   1st Qu.:    911   1st Qu.:   2214
##  Median :   4728   Median :  10370   Median :   4877   Median :   5989
##  Mean   : 140978   Mean   : 141017   Mean   : 141063   Mean   : 140922
```

```
##  3rd Qu.:  52750   3rd Qu.:  60511   3rd Qu.:  36756   3rd Qu.:  33838
##  Max.   :1665633   Max.   :2165933   Max.   :7204190   Max.   :4694846
##     A_684524          A_684528          B_684483          B_684487
##  Min.   :    114   Min.   :      48   Min.   :    309   Min.   :    192
##  1st Qu.:   1527   1st Qu.:     592   1st Qu.:   2449   1st Qu.:   2051
##  Median :   7428   Median :    3164   Median :  10900   Median :  12006
##  Mean   : 140911   Mean   :  140966   Mean   : 141038   Mean   : 141185
##  3rd Qu.:  67985   3rd Qu.:   17146   3rd Qu.:  41716   3rd Qu.:  63356
##  Max.   :2498885   Max.   :12543992   Max.   :3937010   Max.   :5370106
##     B_684491          B_684495          B_684499          B_684503
##  Min.   :    464   Min.   :     88   Min.   :    164   Min.   :     67
##  1st Qu.:   3004   1st Qu.:   2449   1st Qu.:   1592   1st Qu.:   3474
##  Median :   9611   Median :  10563   Median :   5836   Median :  11010
##  Mean   : 141187   Mean   : 140878   Mean   : 140910   Mean   : 141294
##  3rd Qu.:  81266   3rd Qu.:  59358   3rd Qu.:  67631   3rd Qu.:  69077
##  Max.   :2458026   Max.   :1515847   Max.   :3434602   Max.   :2754573
```

Com podem observar, la mitjana és molt major a la mediana en totes les mostres, la qual cosa inidica que la distribució de les dades és asimètrica cap a la dreta.

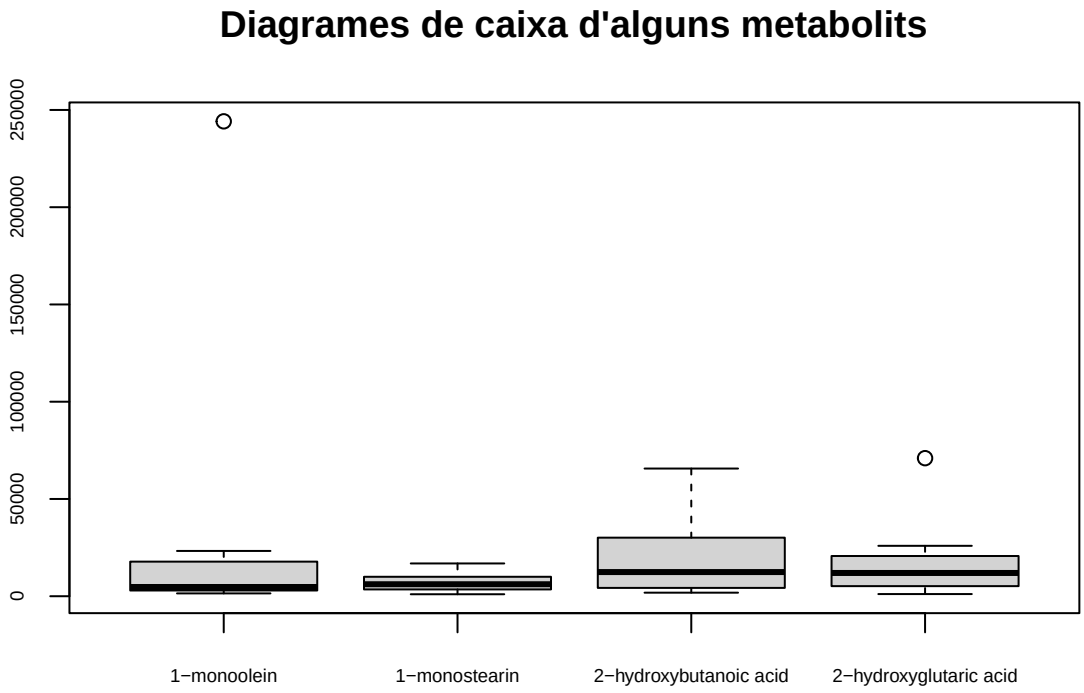
També es mostra el resum estadístic dels quatre primers metabolits del conjunt de dades.

```
summary(t(assay(se)))[,1:4]
```

```
##   1-monoolein     1-monostearin   2-hydroxybutanoic acid 2-hydroxyglutaric acid
##  Min.   :  1452   Min.   : 1013   Min.   : 1823          Min.   : 1109
##  1st Qu.:  2964   1st Qu.: 3838   1st Qu.: 4348          1st Qu.: 6170
##  Median :  4738   Median : 6155   Median :12412          Median :11952
##  Mean   : 27643   Mean   : 6939   Mean   :17779          Mean   :16819
##  3rd Qu.: 17058   3rd Qu.: 9880   3rd Qu.:29937          3rd Qu.:20523
##  Max.   :244142   Max.   :16848   Max.   :65635          Max.   :70972
```

En aquest cas, també observem que les mitjanes són majors a les medianes. A través dels diagrames de caixa, podem observar la presència de valors atípics. Utilitzem la funció `boxplot()` per fer un diagrama de caixa dels quatre primers metabolits del conjunt de dades.

```
boxplot(t(assay(se)[1:4,]), cex.axis=0.6, main="Diagrames de caixa d'alguns metabolits")
```



En els diagrames de caixa, podem observar tant l'asimetria de les dades com la presència de valors atípics en 1-monoolein i 2-hydroxyglutaric acid.

Una vegada feta l'exploració de les dades, es veu que és necessari tractar les dades abans de dur a terme un anàlisi de components principals (PCA). Per això, es normalitzen les dades.

Per tal de normalitzar les dades, en primer lloc, es centren les dades utilitzant la normalització per suma i tot seguit, s'aplica l'escalat per Pareto, Per tal de fer-ho, es creen les funcions `SumNorm()` i `ParetoNorm()`.

```
SumNorm <- function(x) {
  1000*x/sum(x)
}
```

La funció `SumNorm()` normalitza un vector dividint cada element d'aquest per la suma total dels elements del vector. Es multiplica per 1000 per escalar les dades a un rang més manejable.

```
ParetoNorm <- function(x) {
  (x - mean(x))/sqrt(sd(x))
}
```

La funció `ParetoNorm()` aplica la normalització de tipus Pareto ($\tilde{x}_{ij} = \frac{x_{ij}-\bar{x}}{\sqrt{s_i}}$ $s_i = \sqrt{\frac{\sum_{j=1}^{J}(x_{ij}-\bar{x})^2}{J-1}}$).

Una vegada creades les funcions, aquestes ja es poden utilitzar per normalitzar les dades.

En primer lloc, s'expreu la matriu de dades de l'objecte de la classe `SummarizedExperiment` utilitzant la funció `assay()`.

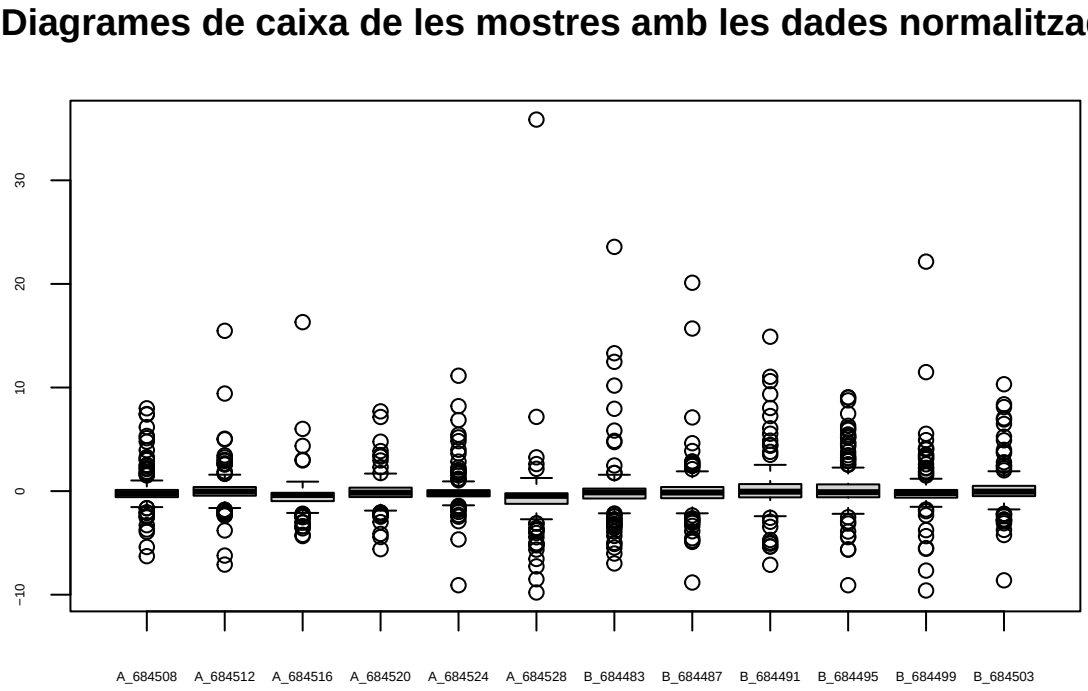
```
X <- assay(se)
```

A continuació, es du a terme la normalització de les dades utilitzant les funcions creades anteriorment i la funció `apply()`.

```
X_centrat <- apply(X, MARGIN=2, SumNorm)
X_norm <- t(apply(X_centrat, MARGIN=1, ParetoNorm))
```

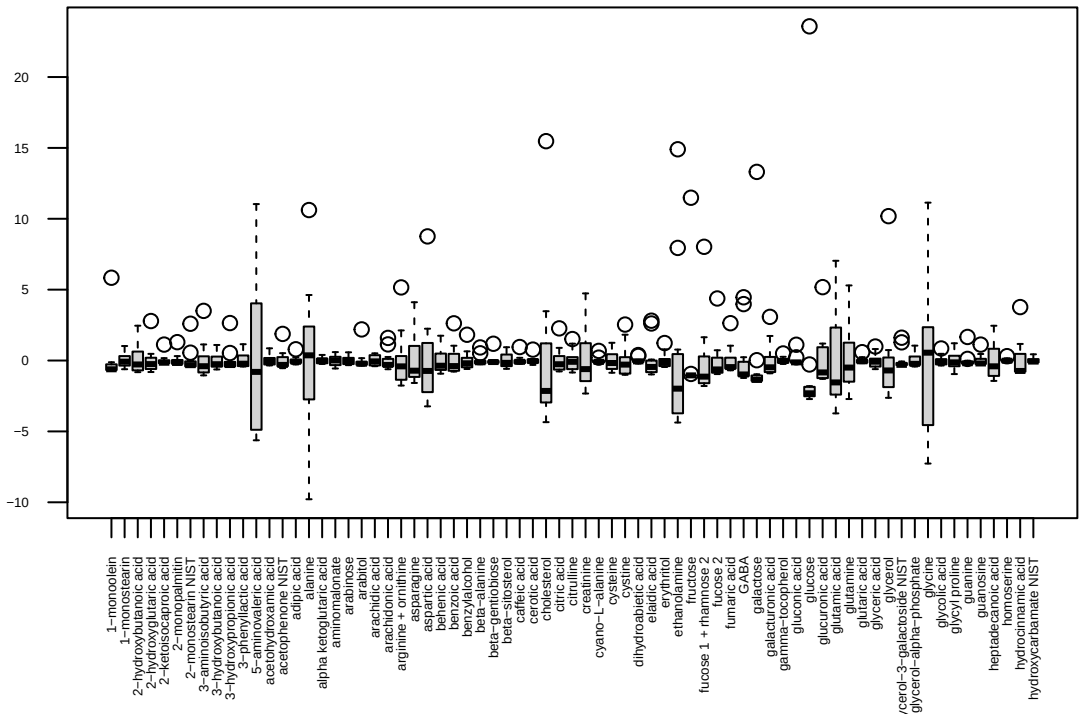
Un cop s'han normalitzat les dades, podem representar-les utilitzant diagrames de caixa de les mostres i dels metabolits.

```
boxplot(X_norm, cex.axis=0.4, main="Diagrames de caixa de les mostres amb les dades normalitzades")
```



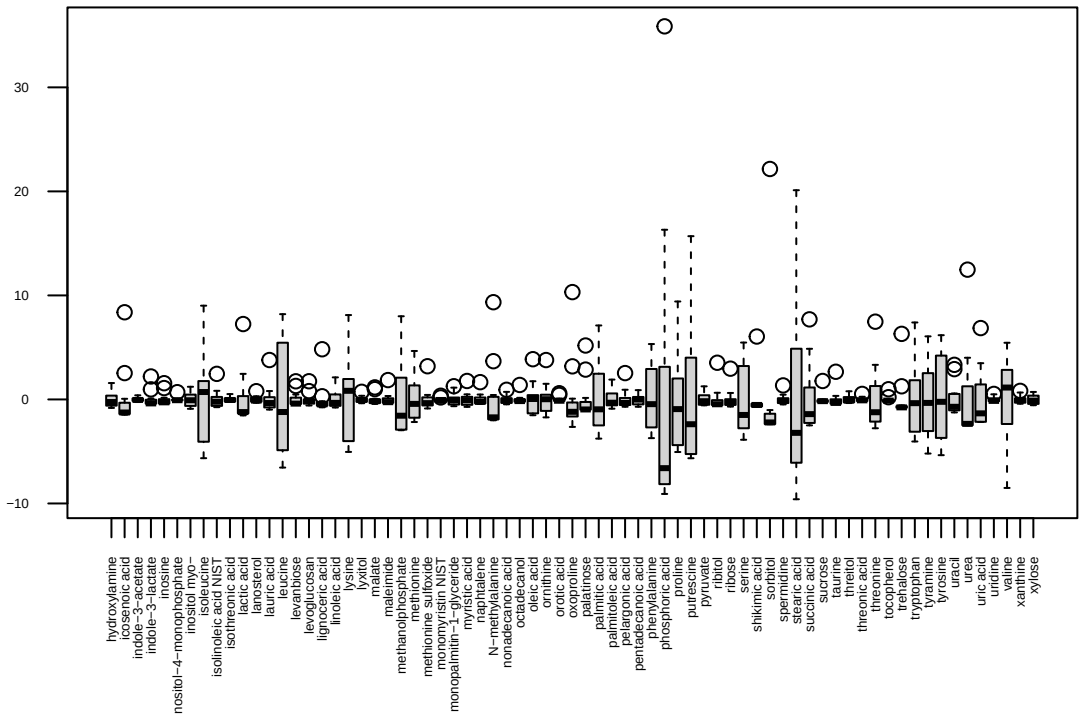

```
boxplot(t(X_norm[1:71,]), cex.axis=0.4, las=2, main="Diagrames de caixa dels metabolits amb les dades n
```

Diagrames de caixa dels metabolits amb les dades normalitzades




```
boxplot(t(X_norm[72:142,]), cex.axis=0.4, las=2, main="Diagrames de caixa dels metabolits amb les dades
```

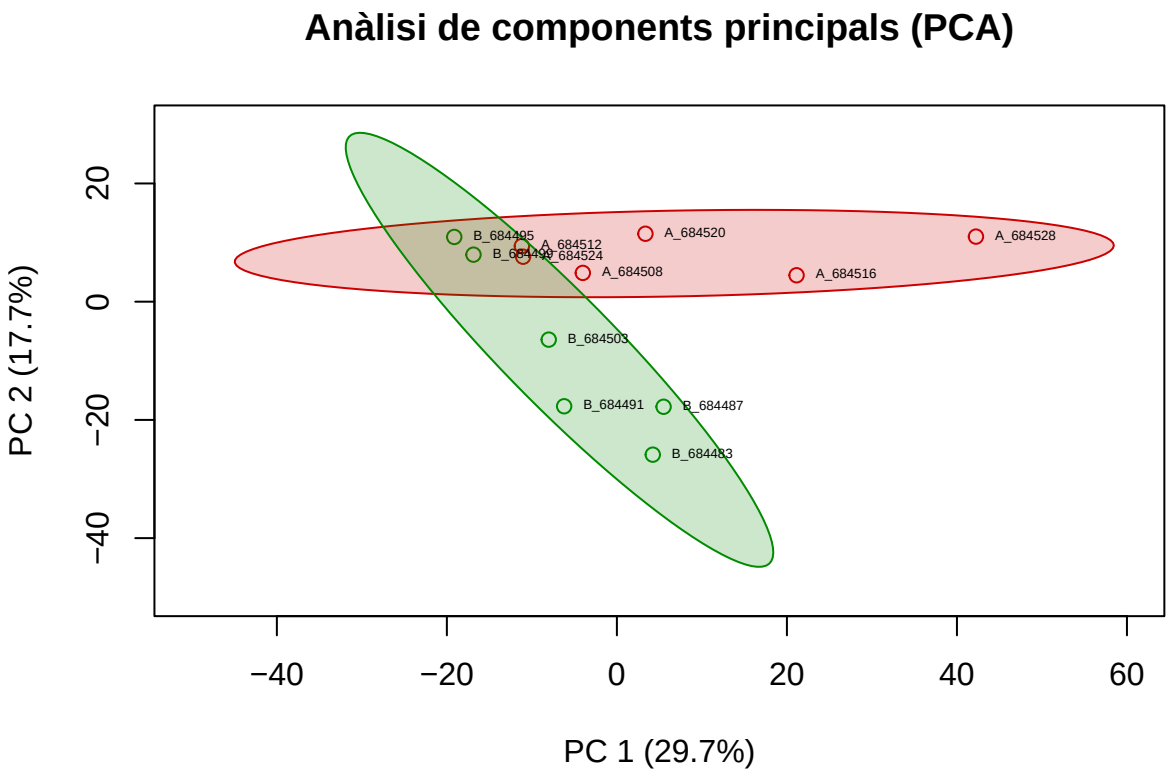
Diagrames de caixa dels metabolits amb les dades normalitzades



Per seguir amb l'anàlisi de les dades, es pot dur a terme un anàlisi de components principals (PCA). Per tal de fer-ho, s'utilitza la funció `prcomp()`.

```
pcX <- prcomp(t(X_norm))
loads <- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
```

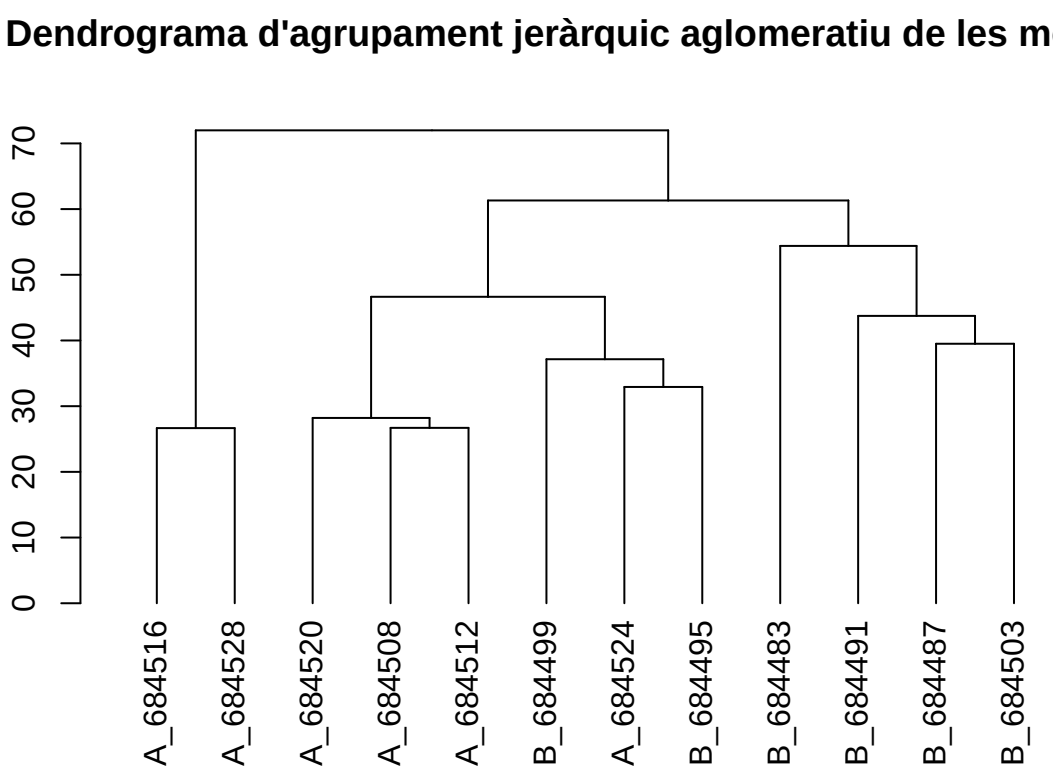
```
xlab <- c(paste0("PC 1 (", loads[1],"%)", sep=""))
ylab <- c(paste0("PC 2 (", loads[2],"%)", sep=""))
colorGrups <- c(rep("red3",6), rep("green4",6))

plot(pcX$x[,1:2], xlab=xlab, ylab=ylab, col=colorGrups, main="Anàlisi de components principals (PCA)", i

for (i in unique(colorGrups)) {
    coordGrup <- pcX$x[colorGrups == i, 1:2]
    matriuCov <- cov(coordGrup)
    centre <- colMeans(coordGrup)
    elipse <- ellipse(matriuCov, centre=centre, level=0.95)

    polygon(elipse, col=adjustcolor(i, alpha=0.2), border=i)
}

text(pcX$x[,1],pcX$x[,2],colnames(X_norm), pos=4, cex=0.4)
```



Hi ha diferència entre els dos grups d'estudi (`Before` i `After`). Tal com es pot observar al gràfic, el primer component permet explicar el 29.7% de la variabilitat de les dades, mentre que el segon component, permet explicar el 17.7% de la variabilitat. També podem observar com no hi ha cap mostra fora dels llindars de confiança. Tot i això, podem observar que les mostres `B_684495`, `B_684499`, `A_684512` i `A_684524` estan molt properes entre elles, la qual cosa ens pot fer pensar que no hi ha gaire diferència entre aquestes mostres d'abans i després del transplantament.

També es pot dur a terme un anàlisi basat en distàncies, el qual, ens permet observar si totes les mostres s'agrupamen correctament. S'espera que les mostres etiquetades com a `Before` siguin més properes entre elles i les mostres etiquetades com a `After`, siguin més properes entre elles. Per tal de dur a terme l'anàlisi, s'utilitzen les funcions `dist()` i `hclust()`

```
dist <- dist(t(X_norm), method="euclidean")
hc <- hclust(dist, method="ward.D2")
plot(as.dendrogram(hc), main="Dendrograma d'agrupament jeràrquic aglomeratiu de les mostres")
```



Tal com podem observar en el dendrograma, les mostres d'abans i després del transplant, s'agrupen entre elles a excepció de les mostres `B_684499` i `A_684524`. S'ha vist en la PCA que aquestes dues mostres estan presenten una gran similitud.

Per observar si hi ha diferències en la quantitat dels metabolits entre les dues condicions (abans i després del transplantament) es duen a terme proves t simples. Per tal de fer-ho, es crea la funció `ttest()`, en la qual, es considera la variància dels dos grups igual.

```
ttest <- function(x) {

  tt <- t.test(x[1:6],x[7:12],var.equal=TRUE)
  return(c(tt$statistic,
           tt$p.value,
           tt$estimate[2]/tt$estimate[1]))
}
```

Cal destacar que per tal de comparar les mostres d'abans (`Before`) respecte a les de després del tractament (`After`), s'especifica en la funció `ttest()` que es divideixi la mitjana del segon grup (`Before`) per la mitjana del primer grup (`After`).

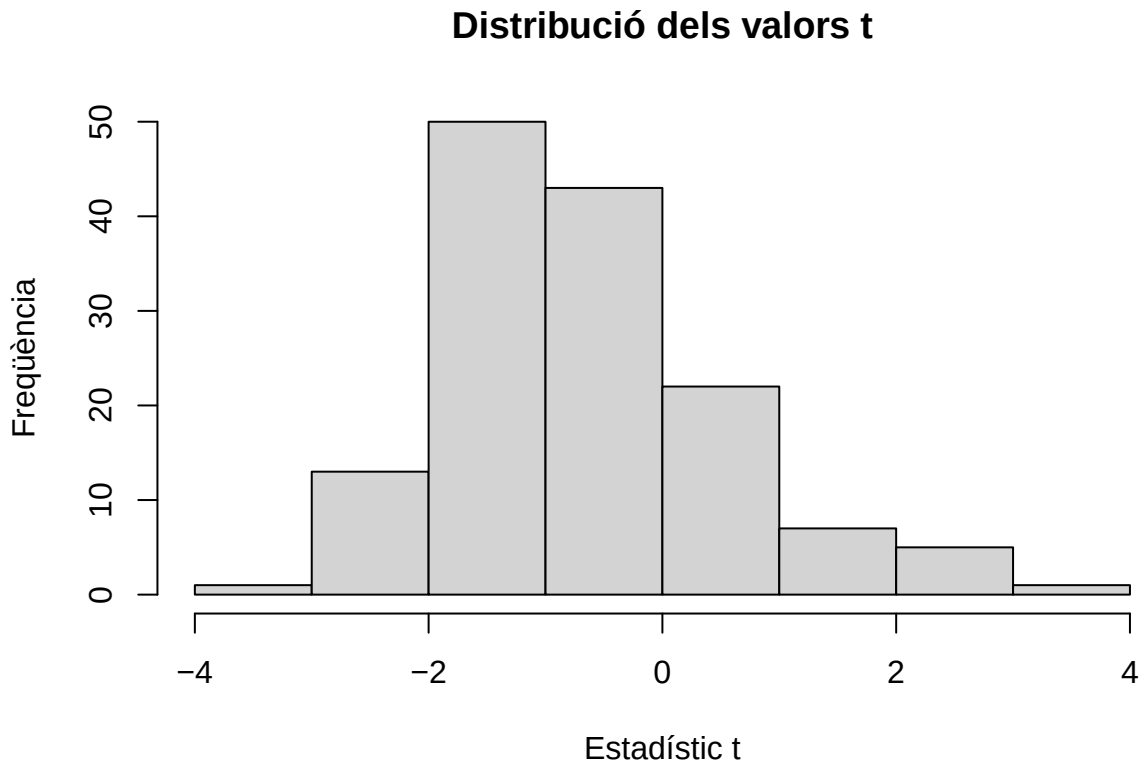
Un cop creada la funció, s'aplica a la matriu de dades sense normalitzar utilitzant la funció `apply()` i, a més, es calcula el `-log10()` del *p-value* i el `log2()` del *Fold Change*.

```
T.test <- apply(X, MARGIN=1, ttest)
T.statistic <- T.test[1,]
T.pvalue <- T.test[2,]
T.logpV <- -log10(T.pvalue)
```

```
T.fc <- T.test[3,]
T.logFc <- log2(T.fc)

resultatTtests <- data.frame(t.stat=T.statistic,
                             pvalue=T.pvalue,
                             logp=T.logpV,
                             FC=T.fc,
                             log2FC=T.logFc)
```

Podem estudiar la distribució de l'estadístic t obtingut per cada metabolit a través d'un histograma tal com es mostra a continuació:

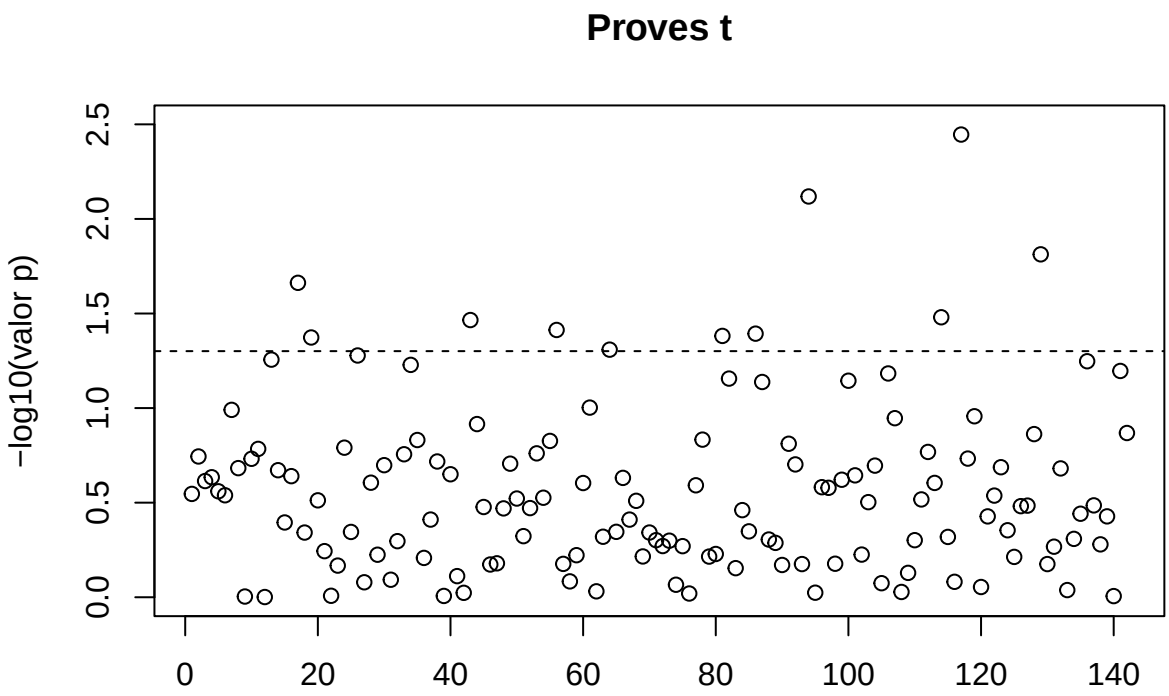
```
hist(T.statistic, main="Distribució dels valors t", xlab="Estadístic t", ylab="Freqüència")
```



Com podem observar hi ha una petita asimetria en les dades, la qual cosa pot suggerir que hi ha alguns metabolits amb un major pic d'intensitat entre els dos grups de mostres.

Podem visualitzar els resultats de les proves t al gràfic que es mostra a continuació:

```
plot(c(1:142), T.logpV, xlab="", ylab="-log10(valor p)", main="Proves t", ylim=c(0,2.5))
abline(h=-log10(0.05), lty=2)
```



La línia discontiínua representa el llindar de significació, de tal manera que els punts que es troben per sobre, representen metabolits amb una altura de pic diferent entre els dos grups. L'eix de les x representa la fila de cada metabolit en la matriu de dades de tal manera que la posició 1 reperesenta el *1-monoolein*, la posició 2, el *1-monostrearin* i així successivament.

A continuació, podem observar quants de metabolits tenen pics d'intensitat significativament diferent entre grups en funció del nivell de significació escollit:

```
for (i in c(0.05, 0.01, 0.001)) {

  print(paste("metabolits amb p valor menor que ", i, ": ", length(which(T.pvalue < i)), sep=""))
}
```

```
## [1] "metabolits amb p valor menor que 0.05: 11"
## [1] "metabolits amb p valor menor que 0.01: 2"
## [1] "metabolits amb p valor menor que 0.001: 0"
```

A continuació es mostra una taula amb els metabolits significativament diferents en funció del nivell de significació:

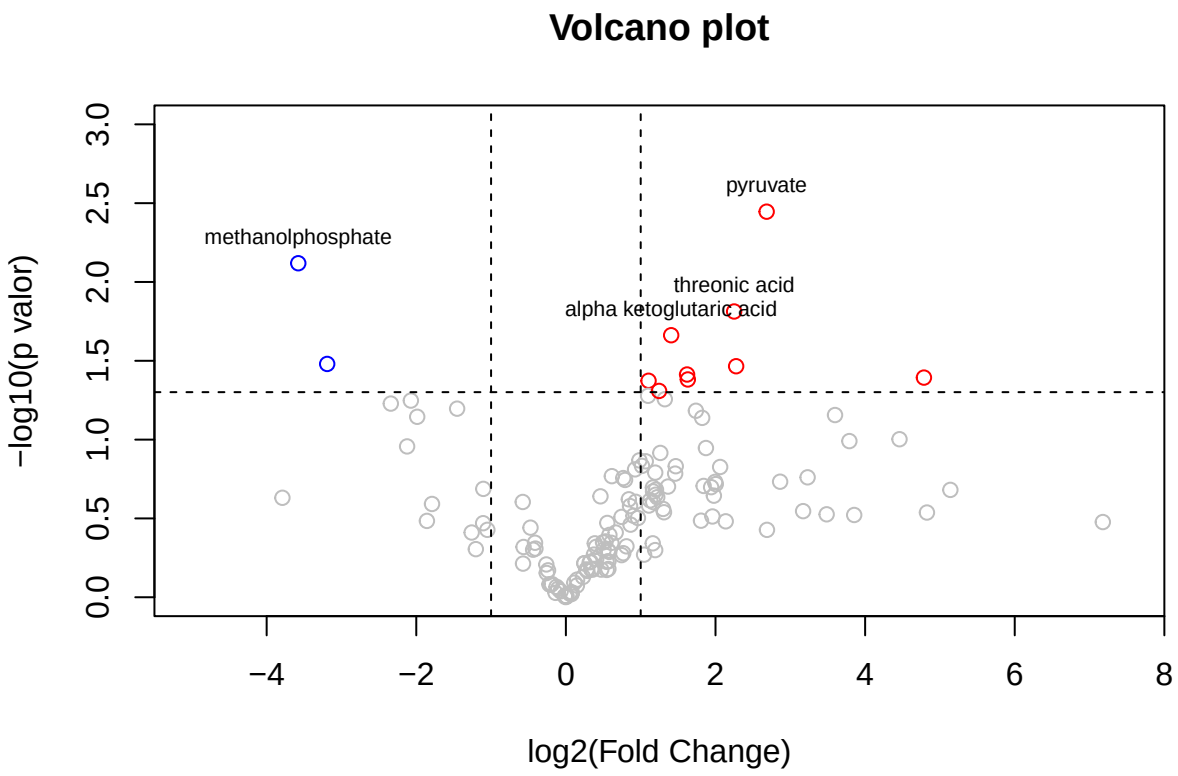
```
metabol5 <- rownames(X)[which(resultatTtests$pvalue < 0.05)]
metabol1 <- rownames(X)[which(resultatTtests$pvalue < 0.01)]

if (length(metabol1) < length(metabol5)) {
  metabol1 <- c(metabol1, rep("-", length(metabol5) - length(metabol1)))
}
```

```r
taula <- data.frame(
  ns5 = metabol5,
  ns1 = metabol1
)

colnames(taula) <- c("Nivell de significació 0.05", "Nivell de significació 0.01")
kable(taula)
```

Nivell de significació 0.05	Nivell de significació 0.01
alpha ketoglutaric acid	methanolphosphate
arabinose	pyruvate
erythritol	-
glutamic acid	-
glycolic acid	-
isothreonic acid	-
levanbiose	-
methanolphosphate	-
phosphoric acid	-
pyruvate	-
threonic acid	-

Finalment, per visualitzar si els metabolits estan regulats positivament o negativament entre els dos grups, es pot dur a terme un *volcano plot* utilitzant el paquet `ggplot2`.

```r
n <- nrow(X)
colors <- vector("character", n)

for (i in 1:n) {

  if (T.logpV[i] < -log10(0.05)) {
    colors[i] <- "grey"
  }
  else if (T.logpV[i] > -log10(0.05) && T.logFc[i] < -log2(2)) {
    colors[i] <- "blue"
  }
  else if (T.logpV[i] > -log10(0.05) && T.logFc[i] > -log2(2)) {
    colors[i] <- "red"
  }
}

plot(T.logFc, T.logpV, col=colors, main="Volcano plot", xlab="log2(Fold Change)", ylab="-log10(p valor)
abline(h=-log10(0.05), lty=2)
abline(v=c(-log2(2),log2(2)), lty=2)

for (i in 1:n) {
  if (T.logpV[i] > 1.5) {
    text(T.logFc[i], T.logpV[i], labels = rownames(X)[i], pos = 3, cex = 0.7)
  }
}
```



En el gràfic anterior, es mostren en color blau els metabolits amb un menor pic en el grup `After` respecte al grup `Before`, mentre que en color vermell, es mostren els metabolits amb un major pic en el grup `After` respecte al grup `Before`. A més, en el gràfic podem observar el nom dels metabolits amb el valor negatiu de logaritme en base de 10 del valor p major a 1.5.

Repositori de *GitHub*

Una vegada realitzada la PEC, es crea un repositori de *GitHub* que conté els diferents arxius que es demanen.

Per tal de crear el repositori, en primer lloc, es crea una compte a *GitHub*. A continuació, es selecciona *Repositories* > *New* on s'especifica el nom del repositori. Un cop fet això, es crea el repositori (*Create repository*) i finalment, es penjen els arxius seleccionant la opció *uploading an existing file*.

Es pot accedir al repositori de la PEC 1 a través del següent enllaç: https://github.com/jordi0ss/Sanso-Sastre-Jordi-PEC1

Referències

- RNA-Seq analysis with R and Bioconductor
- Mini-Tutorial: Análisis de Datos Metabolómicos con MeraboAnalyst 5.0 | BioINfoGRX
- MetaboAnalystR: An R Package for Comprehensive Analysis of Metabolomics Data