

PAC1: Informe de análisis metabolómico

Manuel Quintana Montero

2024-10-31

Tabla de contenidos

Resumen ejecutivo	2
Objetivos del estudio	3
Materiales y métodos	4
Resultados	5
Carga de los datos	5
Creación del objeto de clase <i>SummarizedExperiment</i>	5
Análisis de los datos	7
Creación del repositorio <i>GitHub</i>	16
Discusión, limitaciones del estudio y conclusiones	17

Resumen ejecutivo

A continuación se presenta el análisis preliminar metabolómico de un conjunto de datos obtenido de *GitHub*.

Se utilizó el lenguaje de programación R para cargar los datos y construir un objeto *SummarizedExperiment*, integrando los datos obtenidos del estudio y los metadatos adicionales.

Se presenta un análisis preliminar de los datos, así como el código utilizado para realizar el ejercicio.

Finalmente se crea un repositorio *github* con los datos procesados.

Objetivos del estudio

Los objetivos del estudio son:

- a) Seleccionar un *dataset* de metabolómica
- b) Crear un contenedor del tipo *SummarizedExperiment* que contenga los datos y los metadatos.
- c) Realizar un análisis del *dataset*.
- d) Crear un repositorio de *GitHub* con los datos del estudio.

Materiales y métodos

Los datos metabolómicos fueron descargados desde *GitHub* (<https://github.com/nutrimetabolomics/metaboData/>), concretamente los contenidos en la carpeta *2023-UGrX-4MetaboAnalystTutorial*.

Para realizar el análisis en *RStudio* se descargaron los datos utilizando el paquete *readr* de *CRAN*.

Se obtuvo un archivo txt del cual se separó la parte de los datos del estudio de las partes que correspondían a los metadatos.

Se creó un objeto de clase *SummarizedExperiment* que incluía los datos y los metadatos del estudio. Este objeto se creó usando la librería del mismo nombre incluida en el paquete *Bioconductor*.

Posteriormente se utilizó *RStudio* para explorar la estructura de los datos. Se observó que la matriz de los datos consistía en una matriz de 12 columnas (muestras) y 142 líneas (variables). De las 12 muestras, 6 correspondían a muestras antes de trasplante y las otras 6 correspondían a muestras después del trasplante. Se realizaron cambios con *RStudio* en la matriz de los datos para señalar mejor esto último.

El análisis preliminar de los datos con *RStudio* mostró una gran asimetría hacia la derecha de las concentraciones de metabolitos estudiadas, por lo que se gracias al lenguaje *R* se realizó una transformación logarítmica de los datos para realizar los análisis posteriores.

Continuamos con *RStudio* para realizar un cálculo de las componentes principales. Al reducir los datos a solo 2 componentes (los que más variación explican), pudimos graficar todas las muestras en un plano (2D) y observar patrones o grupos.

A continuación, usamos un agrupamiento jerárquico para Visualizar patrones o relaciones entre las muestras.

Finalmente realizamos un análisis estadístico utilizando *ttest* para encontrar diferencias significativas en las concentraciones de metabolitos entre los grupos pre y post-trasplante.

Resultados

Carga de los datos

En primer lugar, cargamos en *RStudio* las librerías necesarias para nuestro trabajo

```
library(SummarizedExperiment)
library(readr)
```

y descargamos el *dataset* del repositorio de *github*.

```
url <- "https://raw.githubusercontent.com/nutrimetabolomics/metaboData/main/Datasets/2023-UGrX-4MetaboA
download.file(url, destfile = "dataset.txt")
```

A continuación leemos el archivo descargado

```
lines <- read_lines("dataset.txt")
head(lines)
```

```
## [1] "#METABOLOMICS WORKBENCH ofiehn_20130123_9589761_mwtab.txt DATATRACK_ID:34 STUDY_ID:ST000002 ANAL
## [2] "VERSION          \t1"
## [3] "CREATED_ON       \t2016-09-17"
## [4] "#PROJECT"
## [5] "PR:PROJECT_TITLE \tIntestinal Samples II pre/post transplantation"
## [6] "PR:PROJECT_TYPE  \tHuman intestinal samples"
```

Observamos que se trata de un archivo txt con 363 líneas. Dicho archivo contiene la matriz de datos así como información sobre el estudio.

Creación del objeto de clase *SummarizedExperiment*

Procedemos a separar estos bloques. Las primeras 70 líneas hacen referencia a la información del estudio, la separaremos en *study_info*.

```
study_info <- lines[1:70]
```

Los valores del estudio se encuentran en un formato rectangular a partir de la fila 72 y durante un total de 143 líneas. Los almacenaremos en el objeto *data_values*.

```
data_values <- read_delim("dataset.txt", skip = 71, n_max = 143, delim = "\t",
                          show_col_types = FALSE)
head(data_values)
```

```
## # A tibble: 6 x 13
##   Samples      LabF_684508 LabF_684512 LabF_684516 LabF_684520 LabF_684524
##   <chr>          <chr>          <chr>          <chr>          <chr>          <chr>
## 1 Factors      Transplant~ Transplant~ Transplant~ Transplant~ Transplant~
## 2 1-monoolein  6047.0000    2902.0000    1452.0000    3428.0000    2985.0000
## 3 1-monostearin 9771.0000    6521.0000    1302.0000    2781.0000    5789.0000
## 4 2-hydroxybutanoic~ 13238.0000  29774.0000  4134.0000    4419.0000    13334.0000
## 5 2-hydroxyglutaric~ 7160.0000   11501.0000  3202.0000    17238.0000   20376.0000
## 6 2-ketoisocaproic ~ 812.0000    2011.0000    738.0000     2550.0000     871.0000
## # i 7 more variables: LabF_684528 <chr>, LabF_684483 <chr>, LabF_684487 <chr>,
## #   LabF_684491 <chr>, LabF_684495 <chr>, LabF_684499 <chr>, LabF_684503 <chr>
```

Observamos que los datos consisten en una tabla de 13 columnas (una con el nombre de los metabolitos y 12 muestras) y 142 líneas (variables: un factor y 141 metabolitos). De las 12 muestras, según el factor, 6 correspondían a muestras antes de trasplante y las otras 6 correspondían a muestras después del trasplante.

```
colnames(data_values,)
```

```
## [1] "Samples"      "LabF_684508" "LabF_684512" "LabF_684516" "LabF_684520"
## [6] "LabF_684524" "LabF_684528" "LabF_684483" "LabF_684487" "LabF_684491"
## [11] "LabF_684495" "LabF_684499" "LabF_684503"
```

El *dataset* no acaba aquí, sino que a partir de la línea 219 tenemos una tabla con información de cada uno de los 141 metabolitos incluidos en los datos. Esta información la almacenamos en el objeto *info_metab*. Nótese que se ha añadido la instrucción **supressWarnings** al haber campos vacíos en dicha tabla).

```
info_metab <- suppressWarnings(read_delim("dataset.txt", skip = 218, n_max = 142,
                                          delim = "\t", show_col_types = FALSE))
```

A continuación, transformamos el bloque de datos en una matriz. Para ello, suprimimos la primera columna, que es la que incluye el nombre de las variables.

```
assay_data_values <- data_values
assay_data_values[] <- suppressWarnings(lapply(assay_data_values, function(x)
  as.numeric(as.character(x))))
assay_data <- as.matrix(assay_data_values[, -1])
```

No obstante, esos valores de la primera columna servirán para dar el nombre a cada una de las filas de nuestra matriz de datos.

```
rownames(assay_data) <- data_values[[1]]
```

Como nombre de las columnas, a los identificadores de las muestras les añadiremos una “A” o “B” delante en función de si son muestras de después (“*After*”) o antes (“*Before*”) del trasplante. Así, suprimiremos la línea correspondiente al factor, ya que lo hemos incorporado en el nombre de la muestra.

```
colnames(assay_data) <- paste0(ifelse(1:12 <= 6, "A", "B"), colnames(assay_data))
assay_data <- assay_data[-1, ]
```

Seguidamente creamos el objeto *col_data*. Esto es importante para construir el contenedor *SummarizedExperiment* en R, ya que sirve para almacenar información sobre los identificadores de las muestras.

```
col_data <- DataFrame(sampleID = colnames(assay_data))
head(col_data)
```

```
## DataFrame with 6 rows and 1 column
##      sampleID
##      <character>
## 1 ALabF_684508
## 2 ALabF_684512
## 3 ALabF_684516
## 4 ALabF_684520
## 5 ALabF_684524
## 6 ALabF_684528
```

Ya estamos en disposición de generar el objeto de clase *SummarizedExperiment* con la matriz *assay_data* y el objeto *col_data*.

```
se <- SummarizedExperiment(assays = list(counts = assay_data), colData = col_data)
```

Y le añadimos los metadatos:

```
metadata(se) <- list(study_info = study_info)
```

Y le asignamos la información de los metabolitos a las filas del objeto:

```
rowData(se) <- DataFrame(info_metab)
```

Con ello, ya tenemos creado el objeto “se” de clase *SummarizedExperiment*.

Análisis de los datos

```
se
```

```
## class: SummarizedExperiment
## dim: 142 12
## metadata(1): study_info
## assays(1): counts
## rownames(142): 1-monoolein 1-monostearin ... xanthine xylose
## rowData names(9): metabolite_name moverz_quant ... other_id
##   other_id_type
## colnames(12): ALabF_684508 ALabF_684512 ... BLabF_684499 BLabF_684503
## colData names(1): sampleID
```

Si examinamos el objeto *se*, trata de un objeto que contiene información de la concentración de 142 metabolitos en 12 muestras diferentes, las cuales empiezan por “A” o “B” en función de si son después (“*After*”) o antes (“*Before*”) de trasplante.

```
colnames(se)
```

```
## [1] "ALabF_684508" "ALabF_684512" "ALabF_684516" "ALabF_684520" "ALabF_684524"
## [6] "ALabF_684528" "BLabF_684483" "BLabF_684487" "BLabF_684491" "BLabF_684495"
## [11] "BLabF_684499" "BLabF_684503"
```

El listado de los 142 metabolitos. los cuales estaban ordenados por orden alfabético, se pueden listar con el comando *names*.

```
rownames(se)
```

```
## [1] "1-monoolein" "1-monostearin"
## [3] "2-hydroxybutanoic acid" "2-hydroxyglutaric acid"
## [5] "2-ketoisocaproic acid" "2-monopalmitin"
## [7] "2-monostearin NIST" "3-aminoisobutyric acid"
## [9] "3-hydroxybutanoic acid" "3-hydroxypropionic acid"
```

## [11]	"3-phenyllactic acid"	"5-aminovaleric acid"
## [13]	"acetohydroxamic acid"	"acetophenone NIST"
## [15]	"adipic acid"	"alanine"
## [17]	"alpha ketoglutaric acid"	"aminomalonate"
## [19]	"arabinose"	"arabitol"
## [21]	"arachidic acid"	"arachidonic acid"
## [23]	"arginine + ornithine"	"asparagine"
## [25]	"aspartic acid"	"behenic acid"
## [27]	"benzoic acid"	"benzylalcohol"
## [29]	"beta-alanine"	"beta-gentiobiose"
## [31]	"beta-sitosterol"	"caffeic acid"
## [33]	"cerotic acid"	"cholesterol"
## [35]	"citric acid"	"citrulline"
## [37]	"creatinine"	"cyano-L-alanine"
## [39]	"cysteine"	"cystine"
## [41]	"dihydroabietic acid"	"elaidic acid"
## [43]	"erythritol"	"ethanolamine"
## [45]	"fructose"	"fucose 1 + rhamnose 2"
## [47]	"fucose 2"	"fumaric acid"
## [49]	"GABA"	"galactose"
## [51]	"galacturonic acid"	"gamma-tocopherol"
## [53]	"gluconic acid"	"glucose"
## [55]	"glucuronic acid"	"glutamic acid"
## [57]	"glutamine"	"glutaric acid"
## [59]	"glyceric acid"	"glycerol"
## [61]	"glycerol-3-galactoside NIST"	"glycerol-alpha-phosphate"
## [63]	"glycine"	"glycolic acid"
## [65]	"glycyl proline"	"guanine"
## [67]	"guanosine"	"heptadecanoic acid"
## [69]	"homoserine"	"hydrocinnamic acid"
## [71]	"hydroxycarbamate NIST"	"hydroxylamine"
## [73]	"icosenoic acid"	"indole-3-acetate"
## [75]	"indole-3-lactate"	"inosine"
## [77]	"inositol-4-monophosphate"	"inositol myo-"
## [79]	"isoleucine"	"isolinoic acid NIST"
## [81]	"isothreononic acid"	"lactic acid"
## [83]	"lanosterol"	"lauric acid"
## [85]	"leucine"	"levanbiose"
## [87]	"levoglucosan"	"lignoceric acid"
## [89]	"linoleic acid"	"lysine"
## [91]	"lyxitol"	"malate"
## [93]	"maleimide"	"methanolphosphate"
## [95]	"methionine"	"methionine sulfoxide"
## [97]	"monomyristin NIST"	"monopalmitin-1-glyceride"
## [99]	"myristic acid"	"naphtalene"
## [101]	"N-methylalanine"	"nonadecanoic acid"
## [103]	"octadecanol"	"oleic acid"
## [105]	"ornithine"	"orotic acid"
## [107]	"oxoproline"	"palatinose"
## [109]	"palmitic acid"	"palmitoleic acid"
## [111]	"pelargonic acid"	"pentadecanoic acid"
## [113]	"phenylalanine"	"phosphoric acid"
## [115]	"proline"	"putrescine"
## [117]	"pyruvate"	"ribitol"


```
## [119] "ribose"           "serine"
## [121] "shikimic acid"    "sorbitol"
## [123] "spermidine"       "stearic acid"
## [125] "succinic acid"    "sucrose"
## [127] "taurine"          "threitol"
## [129] "threonic acid"    "threonine"
## [131] "tocopherol"       "trehalose"
## [133] "tryptophan"       "tyramine"
## [135] "tyrosine"         "uracil"
## [137] "urea"             "uric acid"
## [139] "uridine"          "valine"
## [141] "xanthine"         "xylose"
```

La **pregunta biológica** que podemos hacernos a partir de los datos es si existen diferencias en los niveles de metabolitos antes y después del trasplante en el tejido intestinal. Las diferencias moleculares encontradas podrían indicar cambios en el metabolismo debido al trasplante o identificar marcadores metabólicos para el monitoreo del proceso de recuperación.

El **diseño del estudio** se realizó dividiendo los datos en muestras pre- y post-trasplante, lo que permitió comparar e identificar las diferencias en las concentraciones de los metabolitos estudiados. Si bien es cierto, al comparar solo dos estados (antes y después del trasplante), las diferencias en los metabolitos podrían estar influenciadas por factores individuales no controlados.

Podemos echar un primer vistazo a los datos realizando un resumen básico de ellos:

```
apply(assay_data, 2, summary)
```

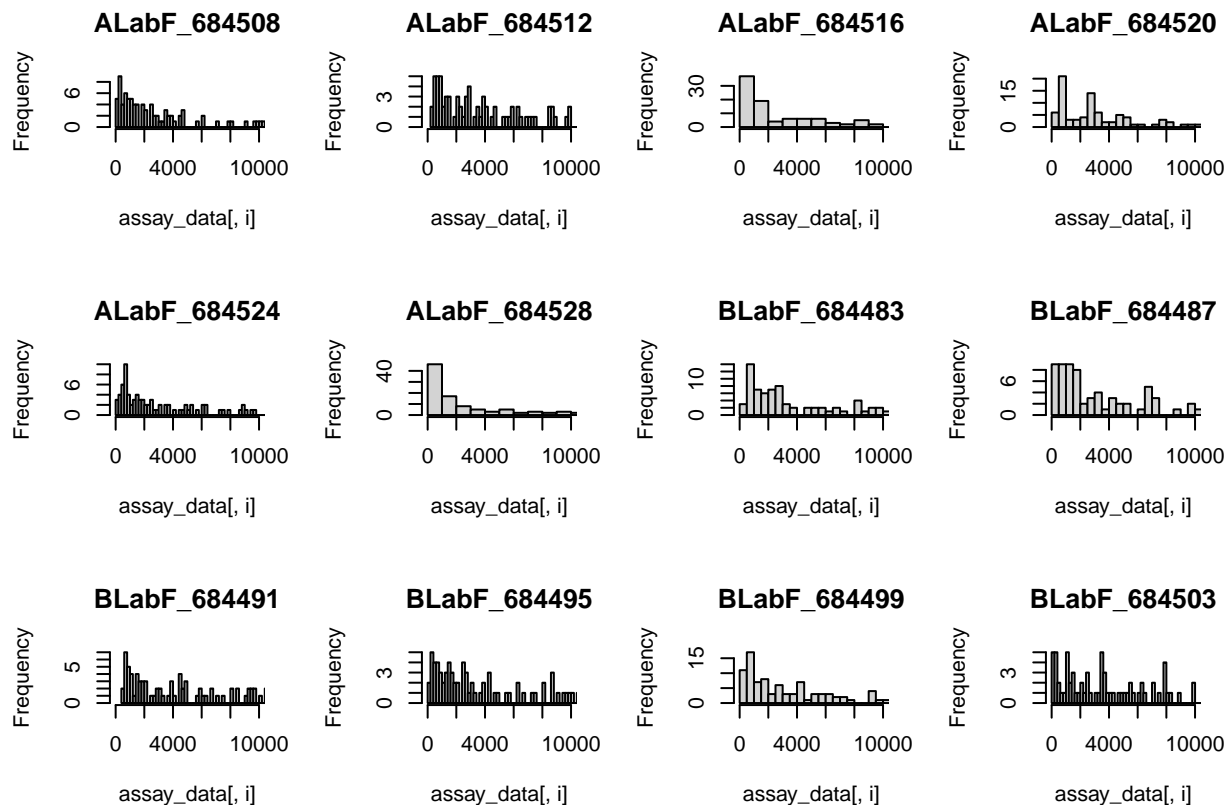
```
##           ALabF_684508 ALabF_684512 ALabF_684516 ALabF_684520 ALabF_684524
## Min.           95.00         336.00          98.00         186.0         114.00
## 1st Qu.       1261.25       2815.00         910.75        2214.0        1527.25
## Median        4728.50      10369.50        4877.00        5989.0        7428.00
## Mean         140977.96     141017.37      141063.10      140921.9      140911.18
## 3rd Qu.       52749.50      60511.25      36756.25      33838.0        67985.25
## Max.         1665633.00     2165933.00    7204190.00    4694846.0    2498885.00
##           ALabF_684528 BLabF_684483 BLabF_684487 BLabF_684491 BLabF_684495
## Min.           48.00         309.00         192.00         464.00          88.00
## 1st Qu.        592.00        2448.75        2051.25        3003.75        2449.00
## Median        3164.50       10899.50       12006.50       9611.00       10563.00
## Mean         140966.43     141038.38     141185.13     141186.82     140877.74
## 3rd Qu.       17145.75      41716.00      63356.50      81265.50      59357.75
## Max.         12543992.00    3937010.00    5370106.00    2458026.00    1515847.00
##           BLabF_684499 BLabF_684503
## Min.           164.00          67.00
## 1st Qu.        1592.00         3474.00
## Median        5836.00       11010.00
## Mean         140910.15     141294.26
## 3rd Qu.        67630.75      69076.75
## Max.         3434602.00     2754573.00
```

Observamos que existe una gran distancia entre los valores máximos y mínimos. Esto sugiere que existen metabolitos con concentraciones muy elevadas en algunas muestras, mientras que otros metabolitos presentan concentraciones bajas. La gran diferencia entre el mínimo y el máximo (más de un millón en algunos casos) podría corresponder a que haya valores atípicos o metabolitos que pueden estar altamente expresados en situaciones específicas.

Las medias son considerablemente más altas que las medianas en todas las muestras, lo cual indica una distribución asimétrica desviada a la derecha. Esto podría ser debido a la presencia de algunos valores extremadamente altos. Esto es importante para decidir si conviene aplicar transformaciones a los datos (por ejemplo, logarítmicas) para reducir la asimetría antes de realizar análisis estadísticos adicionales.

Podemos representar lo observado en forma de histograma para cada una de las muestras:

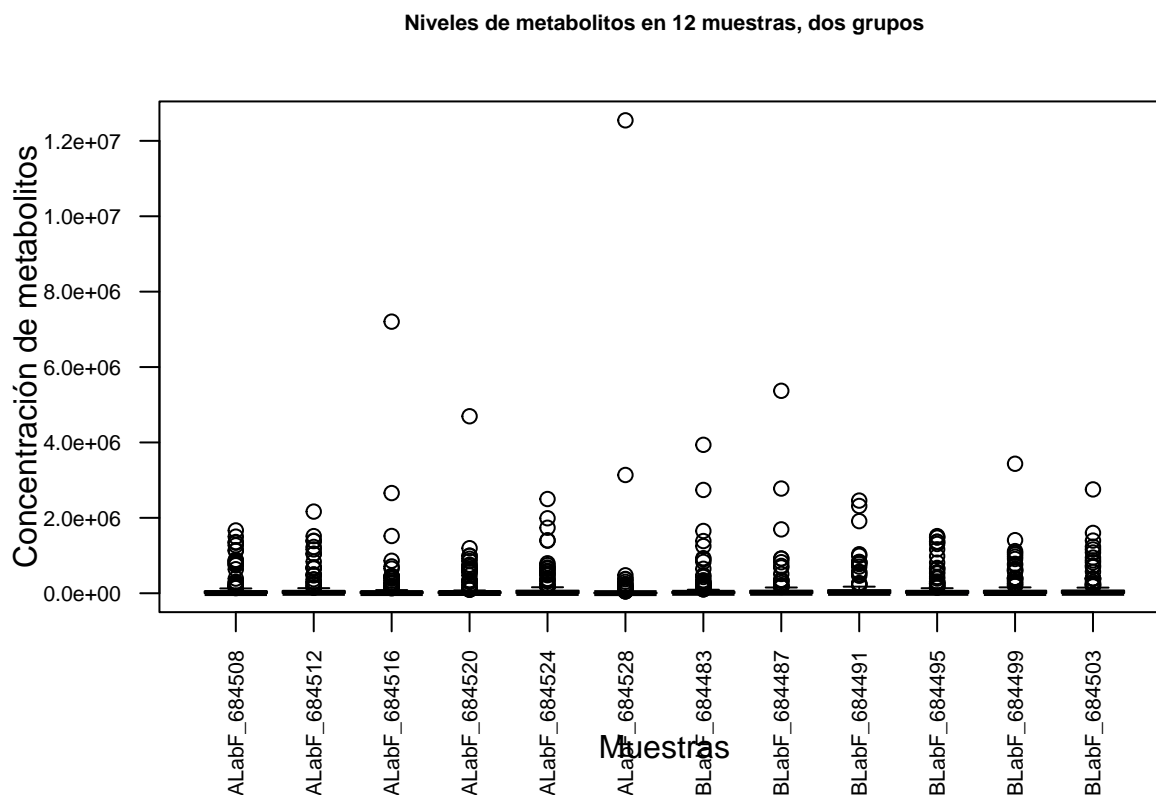
```
opt <- par(mfrow = c(3, 4))
for (i in 1:ncol(assay_data)) {
  hist(assay_data[, i], main = colnames(assay_data)[i], xlim = c(0, 1e4), breaks = 1e4)
}
```



```
par(opt)
```

O en forma de boxplot:

```
groupColors <- c(rep("red", 6), rep("blue", 6))
boxplot(assay_data, col=groupColors, main="Niveles de metabolitos en 12 muestras, dos grupos",
  xlab="Muestras",
  ylab="Concentración de metabolitos", las=2, cex.axis=0.7, cex.main=0.7)
```



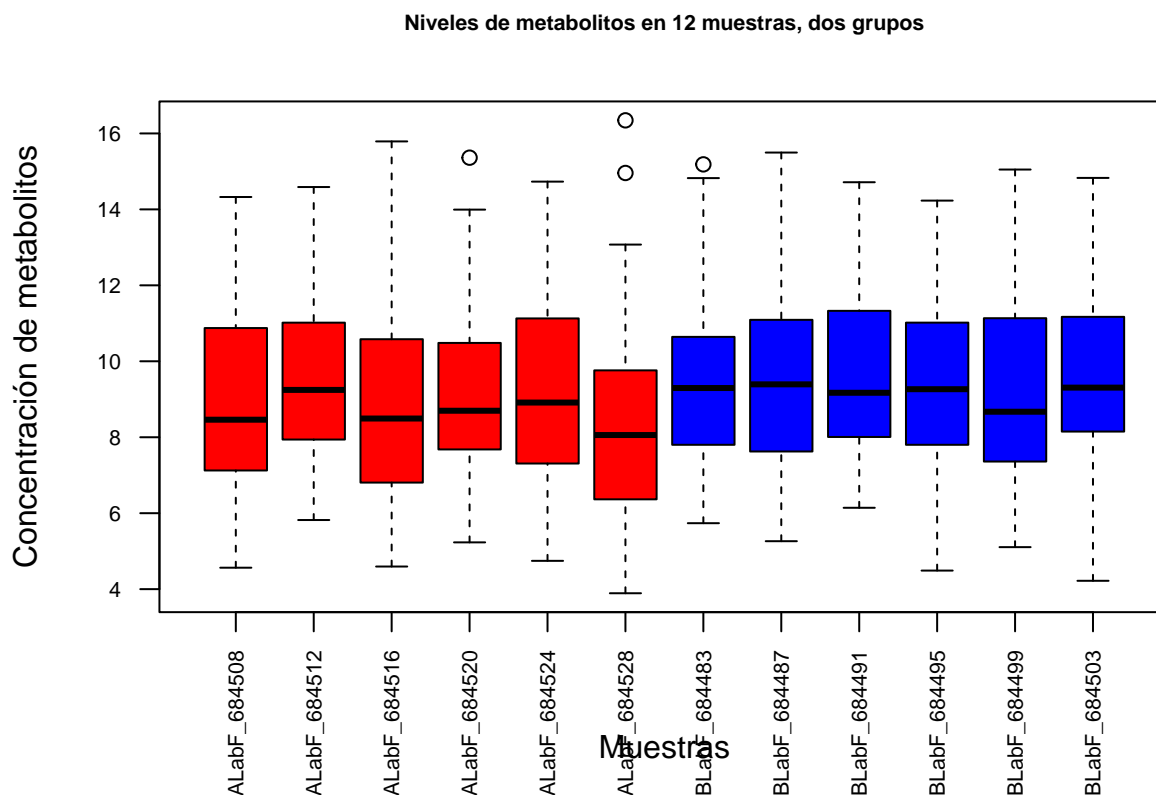
Aquí observamos la distribución asimétrica desviada a la derecha. Sería conveniente aplicar una transformación logarítmica a los datos para reducir esta asimetría (**preprocesamiento**).

Apliquemos la transformación logarítmica con el comando *log1p*:

```
log_assay_data <- log1p(assay_data)
```

Y a continuación realizamos el boxplot sobre los datos transformados:

```
groupColors <- c(rep("red", 6), rep("blue", 6))
boxplot(log_assay_data, col=groupColors, main="Niveles de metabolitos en 12 muestras, dos grupos",
  xlab="Muestras",
  ylab="Concentración de metabolitos", las=2, cex.axis=0.7, cex.main=0.7)
```



A simple vista no se observan diferencias hay una tendencia clara en los valores de los cuartiles y la media que diferencie visiblemente las muestras “A” y “B”. Sin embargo, un análisis más profundo (como un análisis de componentes principales o una comparación estadística) podría ayudar a determinar si hay patrones diferenciadores entre las muestras antes y después del trasplante.

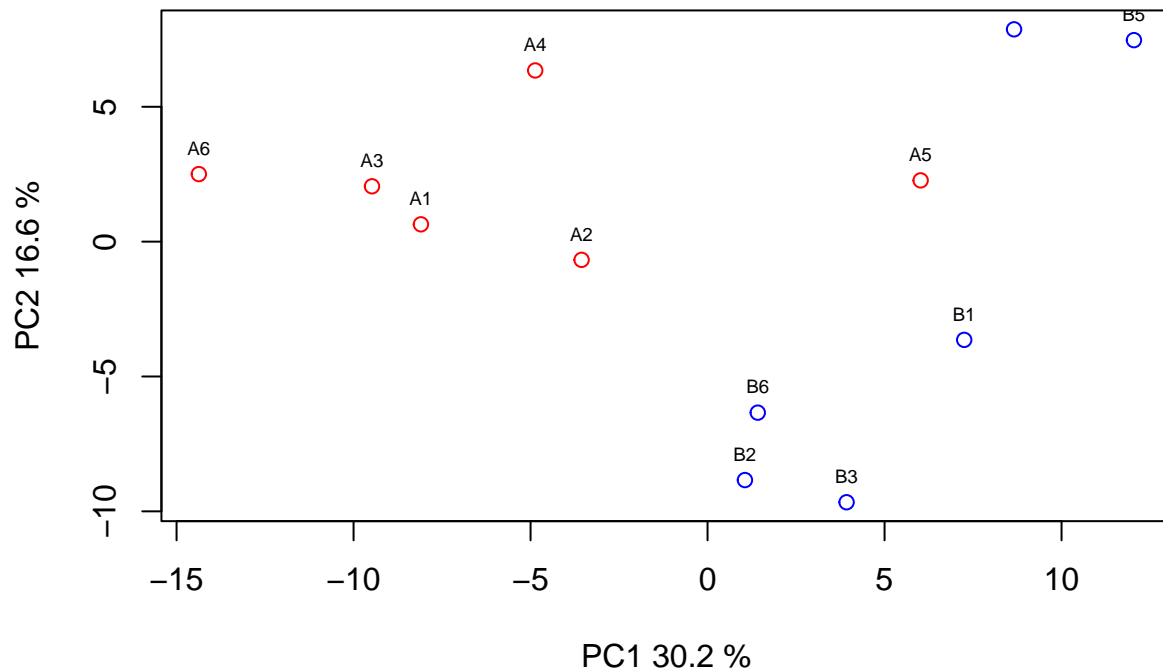
Hagamos el cálculo de los componentes principales y almacenamos los porcentajes de la varianza total explicada por cada componente principal en el objeto *loads*.

```
pc<-prcomp(t(log_assay_data), scale=FALSE)
loads<- round(pc$sdev^2/sum(pc$sdev^2)*100,1)
```

Y a continuación representamos gráficamente las dos primeras componentes:

```
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(pc$x[,1:2],xlab=xlab,ylab=ylab, col=groupColors,
      main ="Componentes principales")
names2plot<-paste0(rep(c("A", "B"), each = 6), 1:6)
text(pc$x[,1],pc$x[,2],names2plot, pos=3, cex=.6)
```

Componentes principales



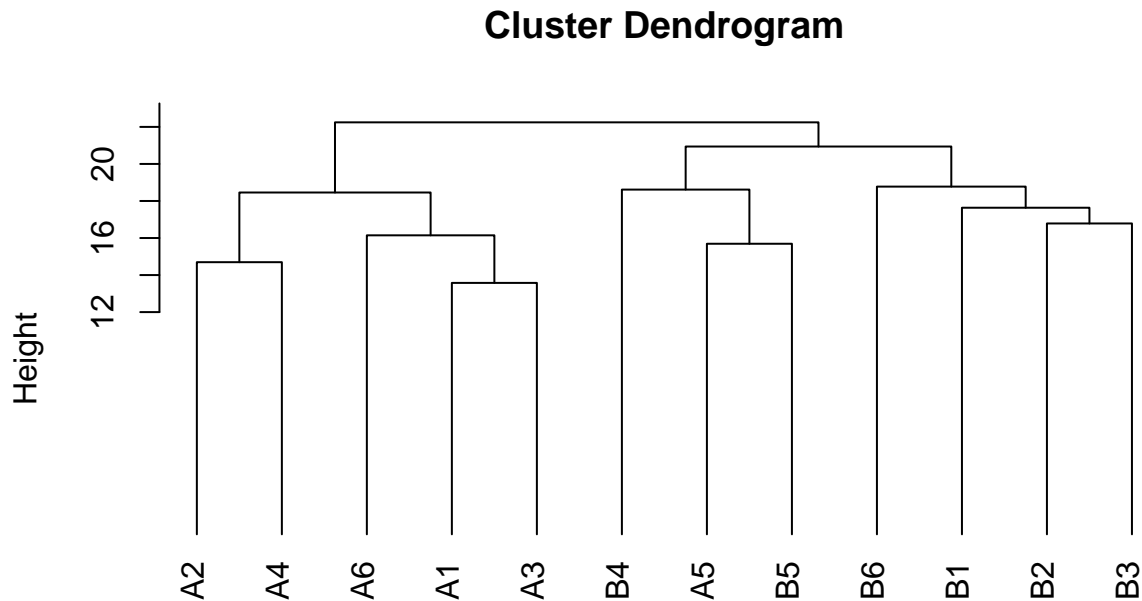
La componente 1 (PC1) explica el 30.2% de la varianza total de los datos, mientras que la 2 (PC2) explica el 16.6% de la varianza. En conjunto, estos dos componentes explican el 46.8% de la variabilidad, lo cual indica que aún hay bastante varianza explicada por otros componentes.

Hay una separación entre los puntos rojos y los azules a lo largo del eje X (PC1). Esto sugiere que PC1 está capturando una diferencia importante entre los dos grupos (antes y después del trasplante).

Dentro de cada grupo (rojo y azul), hay cierta dispersión, especialmente en el eje Y (PC2), lo que podría indicar diferencias individuales dentro de cada grupo.

A continuación realizamos un cluster jerárquico:

```
colnames(log_assay_data) <- names2plot
clust.euclid.average <- hclust(dist(t(log_assay_data)),method="average")
plot(clust.euclid.average, hang=-1)
```



```
dist(t(log_assay_data))
hclust (*, "average")
```

La estructura sugiere que podría haber diferencias metabólicas entre las muestras “A” (después del trasplante) y “B” (antes del trasplante), dado que algunos de estos grupos están claramente separados en el dendrograma.

Para el *análisis estadístico* comparamos la concentración de metabolitos entre los dos grupos con un test t. Para ello definimos una función, *ttest*, una función que toma como entrada un vector x de valores de concentración de metabolitos. Los 6 primeros valores (x[1:6]) representan las muestras post-trasplante, mientras que los 6 siguientes (x[7:12]) son las muestras antes del trasplante.

```
ttest=function(x){
  tt = t.test(x[1:6], x[7:12])
  return(c(tt$statistic, tt$p.value, tt$estimate[1] - tt$estimate[2]))
}
```

La función *t.test* realiza una prueba t de Student para comparar los dos grupos. Los valores devueltos son:
tt\$statistic : el valor de la estadística t. * *tt\$p.value*: el valor p. *tt\$estimate[1] - tt\$estimate[2]*: la diferencia en la media de concentración entre los dos grupos, también llamado *fold change*.

A continuación aplicamos la función *ttest* a cada fila de nuestra matriz de datos, donde cada fila representa un metabolito y cada columna representa una muestra.

```
ans <- apply(log_assay_data, 1, ttest)
ts <- ans[1,]
pvals <- ans[2,]
fc <- ans[3,]
```

El resultado (*ans*) es una matriz donde:

ts almacena los valores del estadístico *t* para cada metabolito. *pvals* almacena los valores *p* para cada metabolito. *fc* almacena los valores de *fold change* para cada metabolito.

A continuación podemos calcular el número de metabolitos que presentan un *p*-valor menor a diferentes puntos de corte:

```
for (i in c(0.05, 0.01))
  print(paste("El número de metabolitos con un p-valor inferior a", i, " es de", length(which(pvals < i))))

## [1] "El número de metabolitos con un p-valor inferior a 0.05 es de 24"
## [1] "El número de metabolitos con un p-valor inferior a 0.01 es de 7"
```

Así, existen hasta 24 genes que muestran diferencias significativas entre los dos grupos, siendo estos:

```
metabolitos_significativos <- rownames(assay_data)[which(pvals < 0.05)]
metabolitos_significativos
```

```
## [1] "2-monostearin NIST"      "alpha ketoglutaric acid"
## [3] "arabinose"              "erythritol"
## [5] "gluconic acid"          "glucuronic acid"
## [7] "glutamic acid"          "glycerol-3-galactoside NIST"
## [9] "glycolic acid"          "isothreonic acid"
## [11] "lactic acid"            "levanbiose"
## [13] "levoglucosan"           "methanolphosphate"
## [15] "orotic acid"            "oxoproline"
## [17] "phosphoric acid"        "pyruvate"
## [19] "ribitol"                "ribose"
## [21] "sorbitol"               "threonic acid"
## [23] "urea"                   "xanthine"
```

Se identificaron diferencias muy significativas en 7 de ellos:

```
metabolitos_muy_significativos <- rownames(assay_data)[which(pvals < 0.01)]
metabolitos_muy_significativos
```

```
## [1] "2-monostearin NIST" "lactic acid"      "levanbiose"
## [4] "methanolphosphate" "phosphoric acid"  "pyruvate"
## [7] "threonic acid"
```

Los metabolitos diferencialmente expresados podrían estar relacionados con el estrés metabólico o la respuesta al trasplante. Se puede considerar el uso de una base de datos para contextualizar estos metabolitos específicos encontrados.

En este caso, ácido láctico y piruvato son ambos metabolitos clave en la vía glucolítica. El piruvato es el producto final de la glucólisis, y puede ser convertido en lactato en condiciones de baja disponibilidad de oxígeno (como sucede en condiciones de estrés metabólico). Levanbiosa es un disacárido, y su presencia sugiere también alteraciones en el metabolismo de carbohidratos en condiciones de estrés.

El fosfato de metanol y ácido fosfórico están relacionados con la transferencia de grupos fosfato. La alteración de estas moléculas puede reflejar cambios en el estado de fosforilación/desfosforilación que podrían tener lugar debido al estrés del trasplante.

La 2-Monostearina es un monoglicérido importante en las membranas celulares.

El ácido treónico está asociado a respuestas antioxidantes. Su presencia podría reflejar un aumento en la demanda antioxidante, probablemente en respuesta al estrés post-trasplante.

Creación del repositorio *GitHub*

Para crear el repositorio, en primer lugar nos registramos en *GitHub*. Posteriormente, creamos el repositorio público “Quintana-Montero-Manuel-PEC1”.

Home

<> Start writing code

Start a new repository for manuelquintana78

A repository contains all of your project's files, revision history, and collaborator discussion.

Repository name *

Quintana-Montero-Manuel-PEC1

✔ Quintana-Montero-Manuel-PEC1 is available.

☒ **Public**
Anyone on the internet can see this repository

☐ **Private**
You choose who can see and commit to this repository

Create a new repository

Figure 1: Creando el repositorio

Y subimos los siguientes archivos:

- a) el informe en formato .Rmd y .pdf,
- b) el objeto contenedor con los datos y los metadatos en formato .Rda
- c) el código R para la exploración de los datos
- d) los datos en formato texto
- e) los metadatos acerca del dataset en un archivo markdown.

El objeto contenedor *se* lo guardamos en formato .Rda con el siguiente comando:

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

El repositorio se puede consultar en la siguiente dirección web: <https://github.com/manuelquintana78/Quintana-Montero-Manuel-PEC1>

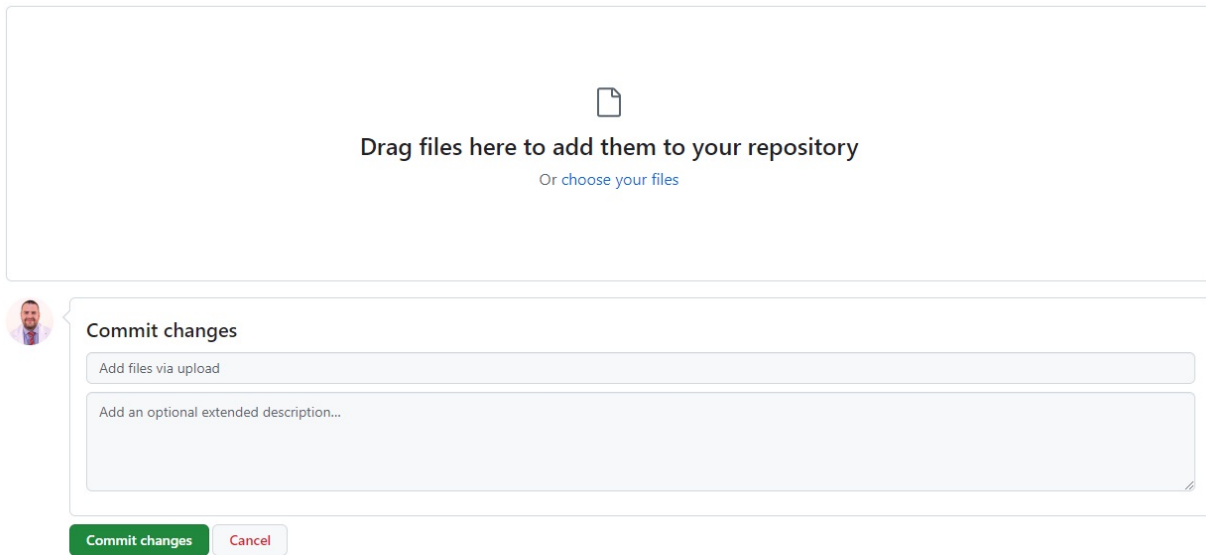


Figure 2: Subida de archivos al repositorio

Discusión, limitaciones del estudio y conclusiones

Este análisis preliminar del conjunto de datos metabolómicos permitió explorar la variabilidad de los metabolitos en muestras intestinales antes y después del trasplante, y demostró cómo algunas transformaciones (como la logarítmica) facilitan la interpretación de datos asimétricos.

Las técnicas estadísticas aplicadas, como el análisis de componentes principales (PCA), el dendograma y el t-test, han sido útiles para observar diferencias entre los dos grupos.

Sin embargo, se presentan algunas limitaciones en este estudio:

- El tamaño de muestra es limitado, lo cual puede afectar la generalización de los resultados.
- Únicamente hemos dividido las muestras en pre y post-trasplante, por lo que podemos no tener en cuenta otras variables, como por ejemplo la edad, los tratamientos seguidos en cada caso, etc.
- Faltaban datos más exhaustivos sobre los metabolitos en los metadatos, lo cual ha hecho que la interpretación biológica de los resultados del estudio haya sido más limitada.

En conclusión, este ejercicio sirvió como una introducción práctica al análisis de datos metabolómicos. Además, la creación de un repositorio de *GitHub* facilitó la organización y la documentación de los datos, metadatos y resultados.