

PAC1

Silvia Arroita Jauregui Avilés

2024-11-05

Tabla de contenido

Abstract.	2
Objetivos del estudio.....	2
Materiales y métodos.	2
Materiales utilizados.....	2
Métodos.....	3
Creación del proyecto en R y del repositorio en GitHub.....	3
Análisis en R.....	3
Resultados.	4
Descarga de datos.....	4
Creación del objeto <i>Summarized Experiment</i>	4
Guardado del objeto <i>Summarized experiment</i>	6
Análisis del dataset.....	6
Resumen inicial del dataset.	8
Normalización de los datos.	11
Estudio de componentes principales.	11
Estudio de la distribución de las características.	13
Generación del <i>heatmap</i> y filtrado de metadatos.....	16
Filtrado de metabolitos.....	17
Discusión del estudio.....	19
Repositorio de GitHub.	19
APÉNDICE.	20

Abstract.

El presente trabajo tiene como objetivo realizar un acercamiento al análisis de datos ómicos utilizando el dataset *2018-MetabotypingPaper*, así como RStudio y GitHub.

En primer lugar, se genera un repositorio en GitHub conectado al proyecto en R y se crea el objeto *SummarizedExperiment* a partir del dataset.

Tras ello, se realiza una exploración del dataset que incluye normalización, análisis de componentes principales, filtrado de metabolitos y visualización de heatmaps.

Estos resultados parecen indicar patrones de correlación entre metabolitos y factores individuales. No obstante, son necesarios más estudios para poder concluir esto, pudiendo ser estos modelos predictivos, análisis longitudinales y análisis de rutas metabólicas.

Objetivos del estudio.

Los objetivos del presente trabajo son ejecutar un proceso simplificado de análisis de datos ómicos, específicamente:

- Generar un proyecto en R y conectarlo a un repositorio en GitHub.
- Crear un objeto Summarized Experiment a través del dataset escogido.
- Realizar una exploración del dataset para tener una visión general de este.

Materiales y métodos.

Materiales utilizados.

Los materiales utilizados son el dataset *2018-MetabotypingPaper* (extraído del repositorio de GitHub *metaboData*, al cual puede accederse a través de <https://github.com/nutrimetabolomics/metaboData/tree/main/Datasets/2018-MetabotypingPaper>).

Este dataset fue utilizado en el paper “*Metabotypes of response to bariatric surgery independent of the magnitude of weight loss*” y está conformado por:

- DataInfo_S013.csv: Este archivo corresponde a los metadatos.
- DataValues_S013.csv: Valores clínicos y metabolómicos de 39 pacientes en 5 puntos temporales.

- `AAInformation_S006.csv`: Información adicional de los metabolitos de “`DataValues_S013.csv`”.

Así mismo, el estudio se realizará utilizando RStudio, con la versión de R 4.4.1. Los paquetes utilizados se exploran en el apartado **Métodos**.

Finalmente, para poder cargar todos nuestros documentos, será necesario utilizar un repositorio de GitHub.

Métodos.

Creación del proyecto en R y del repositorio en GitHub.

En primer lugar, crearemos el repositorio en GitHub. Para ello, con la sesión iniciada, nos dirigiremos a *Repositories / New*.

En la página de creación del Repositorio, tendremos que indicar el nombre, una breve descripción y, en esta ocasión, lo haremos de dominio público. Tras ello, clicaremos en *Create repository*.

Esto nos creará el repositorio, el enlace del cual deberemos copiar en RStudio.

En RStudio, nos dirigimos a *File / New Project / Version Control / Git*. En la pantalla final, deberemos copiar la URL del repositorio creado, así como indicar la dirección local del proyecto en nuestro ordenador.

Finalmente, para sincronizar RStudio con GitHub, deberemos ir haciendo *Commit* en *Git* en las actualizaciones que generemos.

Y con un simple click en *Push* estos quedarán actualizados en GitHub:

Análisis en R.

Para generar el objeto *Summarized experiment* y realizar su posterior análisis, se utilizarán una serie de repositorios y librerías de R, a destacar el repositorio *Bioconductor* y su librería *Summarized Experiment*, además de otras que se verán en el apartado **Resultados**.

Tras obtener el objeto, se hará un primer análisis exploratorio de datos, continuando por normalizar el dataset, así como realizar el cálculo de PCA. También se harán estudios de la distribución de las variables y heatmaps para revisar la relación entre los metabolitos, estos últimos con los datos originales y con un posterior filtrado.

Resultados.

Descarga de datos.

Para la descarga de datos escogeremos los archivos DataValues_S013.csv y DataInfo_S013.csv:

```
# Indicamos Los enlaces de descarga del dataset:
link_DataValues <-
"https://raw.githubusercontent.com/nutrimetabolomics/metaboData/main/Data
sets/2018-MetabotypingPaper/DataValues_S013.csv"
link_DataInfo <-
"https://raw.githubusercontent.com/nutrimetabolomics/metaboData/main/Data
sets/2018-MetabotypingPaper/DataInfo_S013.csv"

# Ponemos nombre a Los archivos:
DataValues <- "DataValues_S013.csv"
DataInfo <- "DataInfo_S013.csv"

# Procedemos a La descarga:
download.file(link_DataValues, destfile = DataValues)
download.file(link_DataInfo, destfile = DataInfo)

# Finalmente, cargamos nuestros archivos:
data_values <- read.csv(DataValues, row.names = 1)
data_info <- read.csv(DataInfo, row.names = 1)
```

Creación del objeto *Summarized Experiment*.

Tras revisar los archivos, vemos que en primer lugar debemos transponer data_values, ya que es importante que en el assay tengamos las muestras en columnas y los metabolitos en filas:

```
data_values <- t(data_values)
```

Además, observamos, gracias a data_info, que los metadatos son las primeras nueve columnas de data_values (*Subjects, Surgery, Age, Gender, Group*, y 4 mediciones de metabolitos iniciales que tienen valores categóricos). Vamos a separarlos para que se identifiquen como metadatos:

```
metadata <- data_values[1:9,]
metadata <- t(metadata)
metadata <- as.data.frame(metadata)
assay_data <- data_values[-c(1:9), ]
```

Ponemos los sujetos como identificativo de fila en metadata y de las columnas de assay_data:

```
rownames(metadata) <- metadata$SUBJECTS
colnames(assay_data) <- metadata$SUBJECTS
```

Convertimos a numéricas las columnas de metadatos correspondientes para facilitar el posterior análisis:

```
metadata$SUBJECTS <- as.numeric(metadata$SUBJECTS)
metadata$AGE <- as.numeric(metadata$AGE)
metadata$MEDDM_T0 <- as.numeric(metadata$MEDDM_T0)
metadata$MEDCOL_T0 <- as.numeric(metadata$MEDCOL_T0)
metadata$MEDINF_T0 <- as.numeric(metadata$MEDINF_T0)
metadata$MEDHTA_T0 <- as.numeric(metadata$MEDHTA_T0)
str(metadata)

## 'data.frame': 39 obs. of 9 variables:
## $ SUBJECTS : num 1 2 3 4 5 6 7 8 9 10 ...
## $ SURGERY : chr "by pass" "by pass" "by pass" "by pass" ...
## $ AGE : num 27 19 42 37 42 24 33 55 40 47 ...
## $ GENDER : chr "F" "F" "F" "F" ...
## $ Group : chr "1" "2" "1" "2" ...
## $ MEDDM_T0 : num 0 0 0 0 0 0 0 0 0 0 ...
## $ MEDCOL_T0: num 0 0 0 0 0 0 0 0 0 0 ...
## $ MEDINF_T0: num 0 0 0 0 0 0 0 1 0 0 ...
## $ MEDHTA_T0: num 1 0 0 0 0 0 0 0 0 0 ...
```

Forzamos los metadatos a ser reconocidos como tal:

```
library(S4Vectors)

colData <- DataFrame(Subjects = metadata$SUBJECTS,
                     Surgery = metadata$SURGERY,
                     Age = metadata$AGE,
                     Gender = metadata$GENDER,
                     Group = metadata$Group,
                     MEDMM_T0 = metadata$MEDDM_T0,
                     MEDCOL_T0 = metadata$MEDCOL_T0,
                     MEDINF_T0 = metadata$MEDINF_T0,
                     MEDHTA_T0 = metadata$MEDHTA_T0)
```

En *assay_data*, detectamos que se están almacenando los datos como caracteres, convertimos también a numéricos y transformamos los NA por 0:

```
# Convertimos Los NA a 0:
assay_data[is.na(assay_data)] <- 0

# Transformamos assay_data en una matriz de 39 columnas y 686 filas:
assay_data <- as.matrix(assay_data)
assay_data <- matrix(as.numeric(assay_data), nrow = 686, ncol = 39)

# Indicamos que Los identificativos de Las filas serán Los metabolitos y
Los de Las columnas Los sujetos:
rownames(assay_data) <- rownames(data_values)[-c(1:9)]
colnames(assay_data) <- metadata$SUBJECTS
```

Hacemos un análisis inicial para comprobar que la generación del objeto *summarized_object* será correcta. Para ello, comprobamos que las dimensiones de *assay_data* y *colData* son compatibles.

También comprobamos que los nombres de las filas de *colData* sean los mismos que los nombres de las columnas de *assay_data*, lo que servirá para identificar de cada muestra sus metadatos.

```
print(dim(assay_data))
## [1] 686 39

print(dim(colData))
## [1] 39 9

print(all(rownames(colData) == colnames(assay_data)))
## [1] TRUE
```

Ahora tenemos que cargar la librería *SummarizedExperiment* para poder generar el objeto homónimo:

```
library(SummarizedExperiment)
```

Creamos el objeto *SummarizedExperiment*. Para ello, usaremos *assay_data* como datos y *colData* como metadatos:

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

Guardado del objeto *Summarized experiment*.

Guardamos el objeto *summarized_experiment* como archivo .RDA:

```
save(summarized_experiment, file = "summarized_experiment.RDA")
```

Análisis del dataset.

En primer lugar, revisamos el objeto *summarized_experiment*. Con esto, podemos confirmar las dimensiones (686 metabolitos y 39 muestras), que solo tiene un assay, en formato de concentración de metabolitos y que tiene metadatos en *colData* sobre cada muestra, formado por 9 variables.

```
summarized_experiment

## class: SummarizedExperiment
## dim: 686 39
## metadata(0):
## assays(1): counts
## rownames(686): GLU_T0 INS_T0 ... SM.C24.0_T5 SM.C24.1_T5
## rowData names(0):
```

```
## colnames(39): 1 2 ... 38 39
## colData names(9): Subjects Surgery ... MEDINF_T0 MEDHTA_T0
```

Si revisamos los metadatos de *colData*, podemos confirmar las 9 variables por muestra:

```
colData(summarized_experiment)

## DataFrame with 39 rows and 9 columns
##      Subjects      Surgery      Age      Gender      Group  MEDMM_T0
MEDCOL_T0
##      <numeric> <character> <numeric> <character> <character> <numeric>
<numeric>
## 1           1      by pass       27           F           1           0
0
## 2           2      by pass       19           F           2           0
0
## 3           3      by pass       42           F           1           0
0
## 4           4      by pass       37           F           2           0
0
## 5           5      tubular       42           F           1           0
0
## ...      ...      ...      ...      ...      ...      ...
...
## 35          35      tubular       39           M           2           0
0
## 36          36      tubular       35           M           1           0
0
## 37          37      by pass       46           M           2           0
0
## 38          38      tubular       41           M           1           0
0
## 39          39      by pass       26           M           1           0
0
##      MEDINF_T0 MEDHTA_T0
##      <numeric> <numeric>
## 1           0           1
## 2           0           0
## 3           0           0
## 4           0           0
## 5           0           0
## ...      ...      ...
## 35          0           1
## 36          0           0
## 37          1           0
## 38          0           0
## 39          0           0
```

Revisando las filas de *assay*, obtenemos un listado de los metabolitos:

```
head(rownames(assay(summarized_experiment)))
```

```
## [1] "GLU_T0"          "INS_T0"          "HOMA_T0"
## [4] "HBA1C_T0"        "HBA1C.mmol.mol_T0" "PESO_T0"
```

Resumen inicial del dataset.

Hacemos un primer resumen de los datos:

```
summary(assay(summarized_experiment))
```

```
##           1           2           3           4
## Min.      :-99.0000   Min.      : -9.00   Min.      :-99.000   Min.      :-99.00
## 1st Qu.:   0.4425   1st Qu.:   0.00   1st Qu.:   0.540   1st Qu.:   0.52
## Median :   3.3000   Median :   1.32   Median :   3.735   Median :   4.49
## Mean      :  36.7822   Mean      :  32.76   Mean      :  34.580   Mean      :  41.23
## 3rd Qu.:  33.0000   3rd Qu.:  24.93   3rd Qu.:  33.830   3rd Qu.:  41.60
## Max.      :876.0000   Max.      :951.00   Max.      :723.000   Max.      :987.00
##           5           6           7           8
## Min.      : -99.0000   Min.      : -99.0000   Min.      :  -9.00   Min.      : -
99.0000
## 1st Qu.:   0.3892   1st Qu.:   0.5525   1st Qu.:   0.00   1st Qu.:
0.4925
## Median :   2.7150   Median :   3.9950   Median :   1.23   Median :
4.0700
## Mean      :  38.7642   Mean      :  42.8119   Mean      :  31.10   Mean      :
39.3962
## 3rd Qu.:  41.1750   3rd Qu.:  38.4500   3rd Qu.:  18.70   3rd Qu.:
43.4500
## Max.      :1070.0000   Max.      :1320.0000   Max.      :1250.00   Max.
:926.0000
##           9          10          11          12
## Min.      : -99.00   Min.      : -99.00   Min.      : -99.000   Min.      : -99.000
## 1st Qu.:   0.00   1st Qu.:   0.00   1st Qu.:   0.000   1st Qu.:   0.920
## Median :   1.16   Median :   1.03   Median :   0.885   Median :   4.115
## Mean      :  32.96   Mean      :  32.74   Mean      :  30.104   Mean      :  40.668
## 3rd Qu.:  21.43   3rd Qu.:  24.75   3rd Qu.:  18.650   3rd Qu.:  39.475
## Max.      :1150.00   Max.      :954.00   Max.      :892.000   Max.      :746.000
##          13          14          15          16
## Min.      : -99.0000   Min.      : -99.00   Min.      :  -9.000   Min.      :  -
9.0000
## 1st Qu.:   0.8608   1st Qu.:   0.59   1st Qu.:   0.570   1st Qu.:
0.5225
## Median :   4.1100   Median :   4.14   Median :   4.805   Median :
3.9400
## Mean      :  38.6822   Mean      :  46.03   Mean      :  50.166   Mean      :
42.5622
## 3rd Qu.:  41.0000   3rd Qu.:  41.60   3rd Qu.:  56.025   3rd Qu.:
43.0500
## Max.      :654.0000   Max.      :1050.00   Max.      :1350.000   Max.
:833.0000
```


##	17	18	19	20
##	Min. : -99.00	Min. : -9.000	Min. : -9.0000	Min. : -99.0000
##	1st Qu.: 0.53	1st Qu.: 0.500	1st Qu.: 0.5308	1st Qu.: 0.5748
##	Median : 4.49	Median : 4.145	Median : 4.1900	Median : 5.0350
##	Mean : 45.91	Mean : 49.194	Mean : 42.2168	Mean : 45.2580
##	3rd Qu.: 43.67	3rd Qu.: 47.450	3rd Qu.: 39.9000	3rd Qu.: 46.9750
##	Max. :1040.00	Max. :1560.000	Max. :863.0000	Max. :1240.0000
##	21	22	23	24
##	Min. : -9.0000	Min. : -99.00	Min. : -99.000	Min. : -99.000
##	1st Qu.: 0.4625	1st Qu.: 0.00	1st Qu.: 0.350	1st Qu.: 0.645
##	Median : 3.2500	Median : 0.98	Median : 2.665	Median : 4.820
##	Mean : 35.3213	Mean : 36.12	Mean : 35.076	Mean : 51.008
##	3rd Qu.: 34.9750	3rd Qu.: 21.60	3rd Qu.: 32.000	3rd Qu.: 49.525
##	Max. :677.0000	Max. :1230.00	Max. :965.000	Max. :1280.000
##	25	26	27	28
##	Min. : -99.0000	Min. : -99.0000	Min. : -99.000	Min. : -99.000
##	1st Qu.: 0.4725	1st Qu.: 0.4325	1st Qu.: 0.450	1st Qu.: 0.000
##	Median : 3.4300	Median : 3.4200	Median : 3.155	Median : 0.018
##	Mean : 41.3190	Mean : 33.7009	Mean : 39.297	Mean : 20.098
##	3rd Qu.: 43.0000	3rd Qu.: 32.9000	3rd Qu.: 38.975	3rd Qu.: 4.662
##	Max. :935.0000	Max. :871.0000	Max. :907.000	Max. :713.000
##	29	30	31	32
##	Min. : -99.00	Min. : -99.000	Min. : -99.00	Min. : -9.000
##	1st Qu.: 0.00	1st Qu.: 0.720	1st Qu.: 0.00	1st Qu.: 0.380
##	Median : 1.23	Median : 5.615	Median : 1.24	Median : 2.735
##	Mean : 31.51	Mean : 56.929	Mean : 39.08	Mean : 38.141
##	3rd Qu.: 24.45	3rd Qu.: 58.750	3rd Qu.: 23.93	3rd Qu.: 35.975
##	Max. :846.00	Max. :1240.000	Max. :1140.00	Max. :921.000
##	33	34	35	36
##	Min. : -9.000	Min. : -99.0000	Min. : -9.0000	Min. : -9.000
##	1st Qu.: 0.390	1st Qu.: 0.0512	1st Qu.: 0.3925	1st Qu.: 0.390

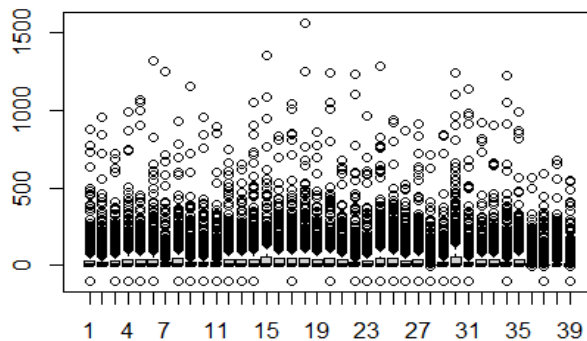
```

0.000
## Median : 2.425 Median : 1.5050 Median : 3.1000 Median :
0.000
## Mean : 36.208 Mean : 36.9961 Mean : 41.1227 Mean :
20.247
## 3rd Qu.: 36.475 3rd Qu.: 23.6250 3rd Qu.: 38.5850 3rd Qu.:
4.562
## Max. :903.000 Max. :1220.0000 Max. :990.0000 Max.
:562.000
## 37 38 39
## Min. : -99.000 Min. : -9.00 Min. : -99.000
## 1st Qu.: 0.000 1st Qu.: 0.00 1st Qu.: 0.000
## Median : 0.000 Median : 0.76 Median : 0.057
## Mean : 18.263 Mean : 26.98 Mean : 23.597
## 3rd Qu.: 4.157 3rd Qu.: 16.45 3rd Qu.: 5.085
## Max. :593.000 Max. :680.00 Max. :552.000

```

Vamos a reforzar estos resultados con un boxplot para poder revisar mejor los estadísticos:

```
boxplot(assay(summarized_experiment))
```



Con estos resultados, podemos observar lo siguiente:

- Vemos puntos mínimos en negativo, como -99 (muestras 1, 4, 5...) y -9, pudiendo ser valores faltantes o errores en la medición, dando lugar a datos inválidos.
- Se observan valores máximos muy altos (1560 en la muestra 18) en comparación con el resto, por lo que sería interesante estudiar la presencia de outliers que pudieran afectar el análisis.
- Vemos las medianas y los valores medios bastante distintos, por lo que los datos podrían ser asimétricos.

- El rango de los datos es muy amplio, como se ha observado con los puntos mínimos y los valores máximos.

Habiendo observado este primer resumen de los datos, vamos a eliminar los valores negativos:

```
assay(summarized_experiment)[assay(summarized_experiment) < 0] <- NA
```

Normalización de los datos.

A continuación, debido a que vemos mucha asimetría en los datos, vamos a proceder a realizar la normalización de estos. Para ello, creamos un objeto *DESeqDataSet*, utilizando los grupos como condición, siguiendo la naturaleza del estudio inicial:

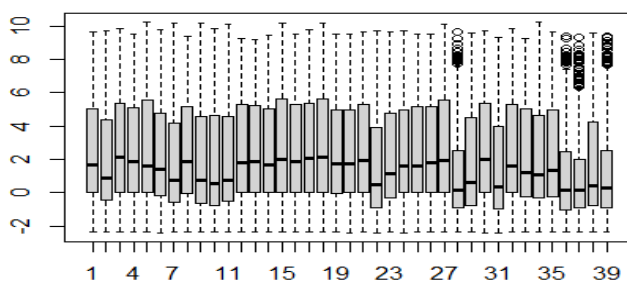
```
library(DESeq2)
sum(is.na(assay(summarized_experiment)))
## [1] 310
assay(summarized_experiment)[is.na(assay(summarized_experiment))] <- 0
assay(summarized_experiment) <- round(assay(summarized_experiment))
DESDataset <- DESeqDataSet(summarized_experiment, design = ~ Group)
```

Tras ello, aplicamos la normalización:

```
DES_vst <- rlog(DESDataset) assay(summarized_experiment) <- assay(DES_vst)
```

Revisamos la normalización:

```
boxplot(assay(summarized_experiment))
```



Obtenemos datos más homogéneos, se han eliminado los posibles outliers, disminuido el rango de muestra y, por tanto, la dispersión de los datos y la variabilidad.

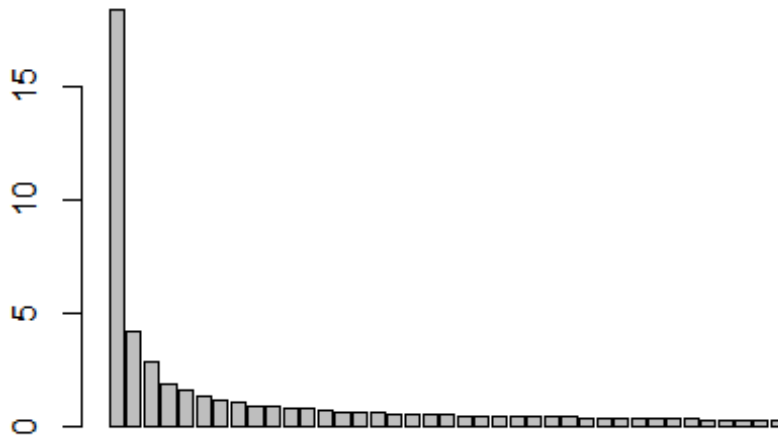
Estudio de componentes principales.

Ahora, vamos a revisar los componentes principales:

```
#Extraemos data.frame:
data_for_pcs <- assay(summarized_experiment)
pcs <- prcomp(data_for_pcs)
names(pcs)

## [1] "sdev"      "rotation" "center"   "scale"    "x"

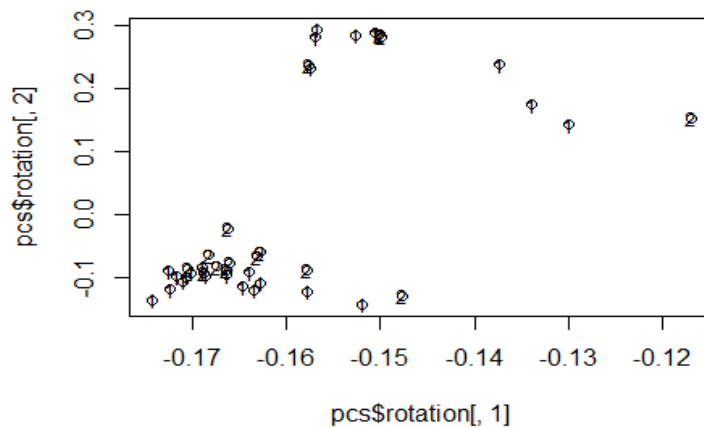
barplot(pcs$sdev)
```



Observamos que la mayoría de la variabilidad la explica el primer componente principal.

```
plot(pcs$rotation[, 1], pcs$rotation[, 2],
      main = "Representación de los dos primeros componentes principales")
text(pcs$rotation[, 1], pcs$rotation[, 2], metadata$Group, cex = 0.8, pps
= 3)
```

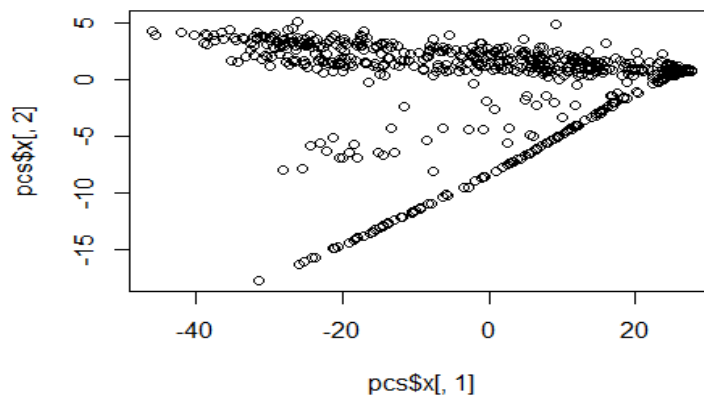
presentación de los dos primeros componentes prin



Se detecta una agrupación en la esquina inferior izquierda de variables, lo que podría indicar correlación entre ellas.

No obstante, el resto de puntos se encuentran dispersos, por lo que no parece que haya correlaciones muy altas entre las variables.

```
plot(pcs$x[,1], pcs$x[,2])
```



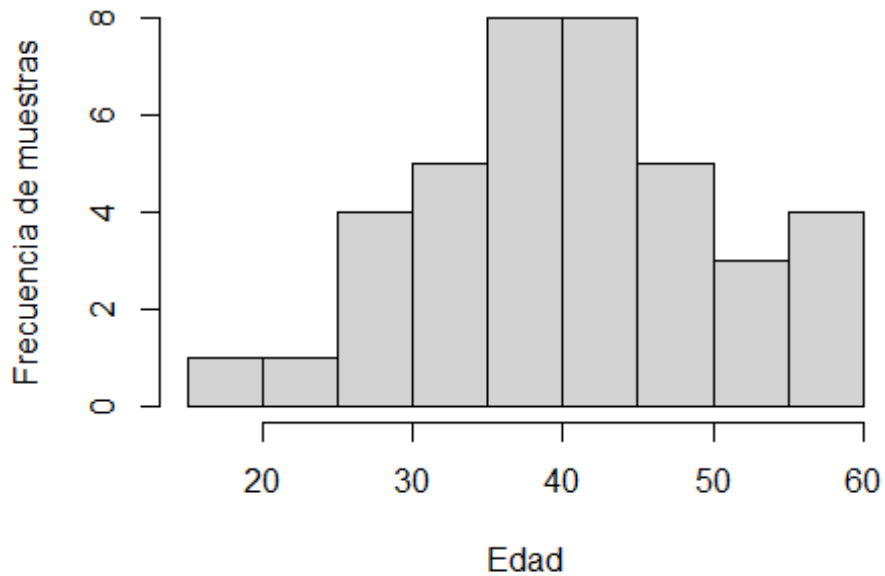
Hay una tendencia en la dispersión a la diagonal, lo que parece indicar una variabilidad en los dos componentes principales.

Estudio de la distribución de las características.

Vamos a realizar un breve estudio para revisar la distribución de las variables de los metadatos, empezando por la edad:

```
hist(colData(summarized_experiment)$Age, main = "Distribución de la  
edad", xlab = "Edad", ylab = "Frecuencia de muestras")
```

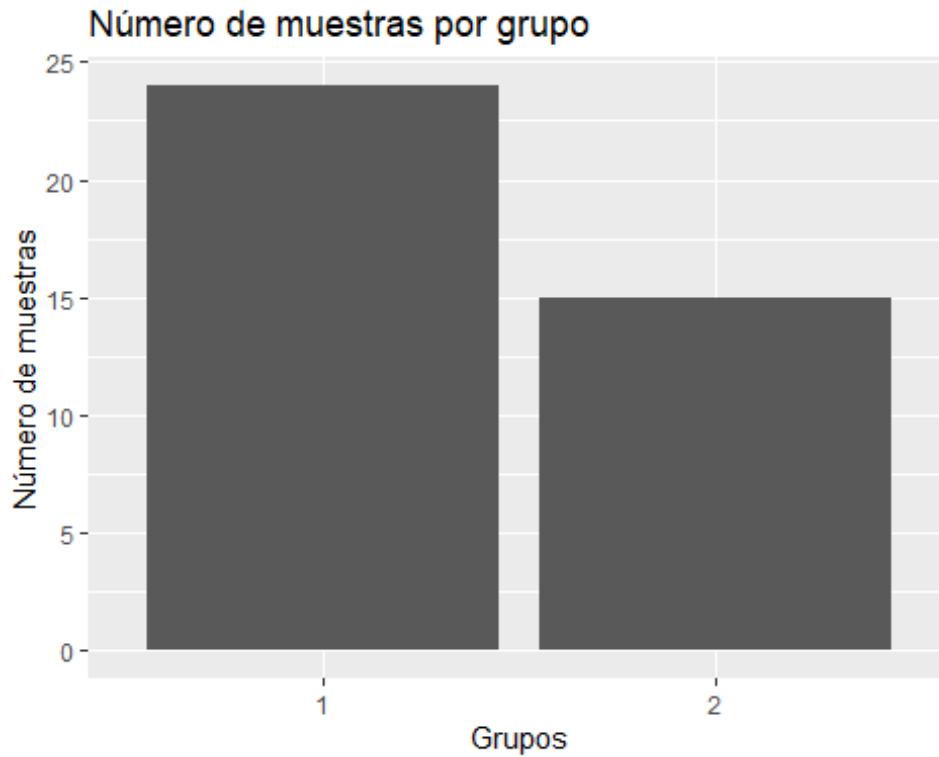
Distribución de la edad



Observamos que la media de la edad de los sujetos está entorno a los 40 años, yendo esta de los 15 a los 60 años.

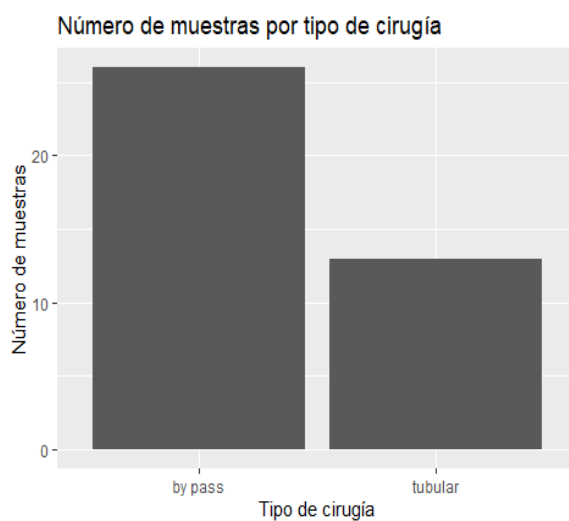
Referente a los grupos, vemos que la mayoría de los participantes corresponden al primer grupo:

```
library(ggplot2)
ggplot(as.data.frame(colData(summarized_experiment)), aes(x = Group)) +
  geom_bar() +
  xlab("Grupos") +
  ylab("Número de muestras") +
  ggtitle("Número de muestras por grupo")
```



Finalmente, vemos que referente al tipo de cirugía, la mayoría de los pacientes se sometieron a un by pass:

```
ggplot(as.data.frame(colData(summarized_experiment)), aes(x = Surgery)) +
  geom_bar() +
  xlab("Tipo de cirugía") +
  ylab("Número de muestras") +
  ggtitle("Número de muestras por tipo de cirugía")
```

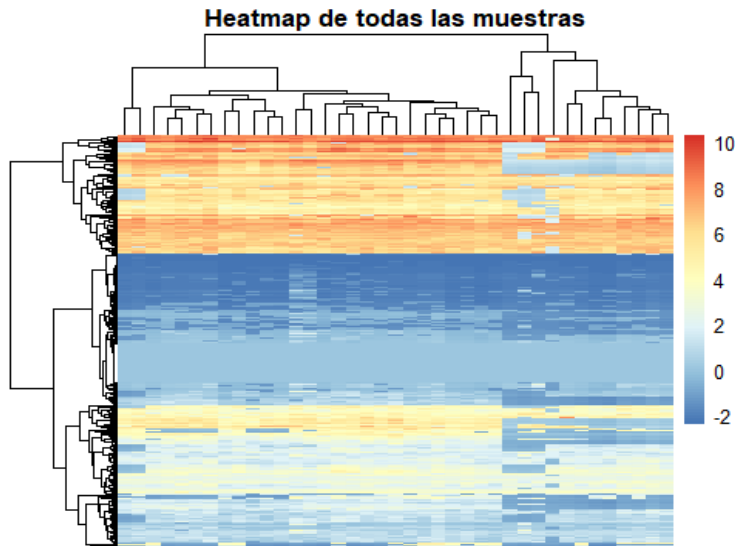


Generación del *heatmap* y filtrado de metadatos.

En primer lugar, vamos a generar un *heatmap* de todas las muestras:

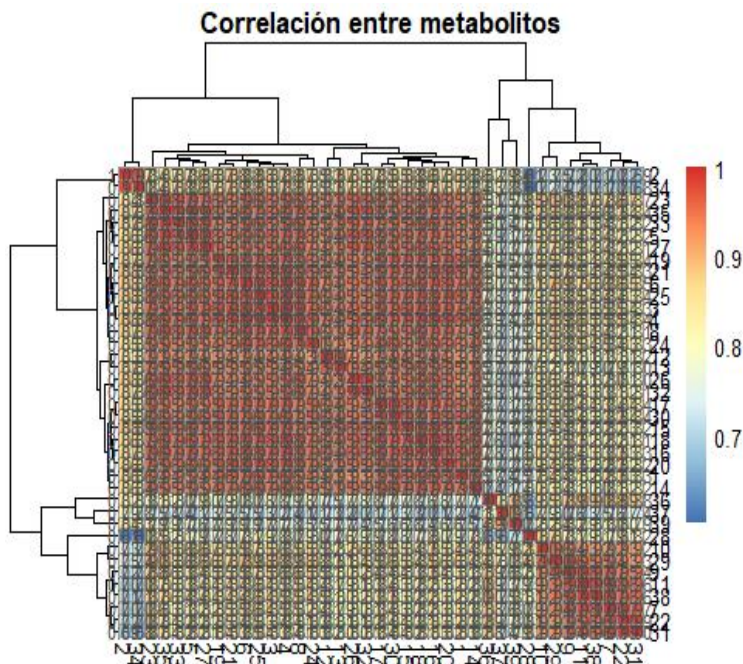
```
library(pheatmap)

pheatmap(assay(summarized_experiment), cluster_rows = TRUE, cluster_cols = TRUE,
         show_rownames = FALSE, show_colnames = FALSE, main = "Heatmap de todas las muestras")
```



Y también realizamos un *heatmap* de la correlación entre los metabolitos:

```
cor_matrix <- cor(assay(summarized_experiment))
pheatmap(cor_matrix, main = "Correlación entre metabolitos",
         display_numbers = TRUE)
```

Filtrado de metabolitos.

Para poder revisar con más facilidad estos *heatmaps*, vamos a filtrar los metadatos.

En primer lugar, filtraremos los metabolitos que tienen muchos valores NA:

```
valor_na <- 0.5 * ncol(summarized_experiment)
na_correctos <- rowSums(is.na(assay(summarized_experiment))) <= valor_na
```

También descartaremos los metabolitos con baja variabilidad entre las muestras, cogiendo de varianza < 0.1:

```
varianza <- apply(assay(summarized_experiment), 1, var, na.rm = TRUE)
var_correctos <- varianza > 0.1
```

Filtramos según los resultados anteriores:

```
metabolitos_filtrados <- na_correctos & var_correctos
summarized_experiment_filtered <-
summarized_experiment[metabolitos_filtrados, ]
```

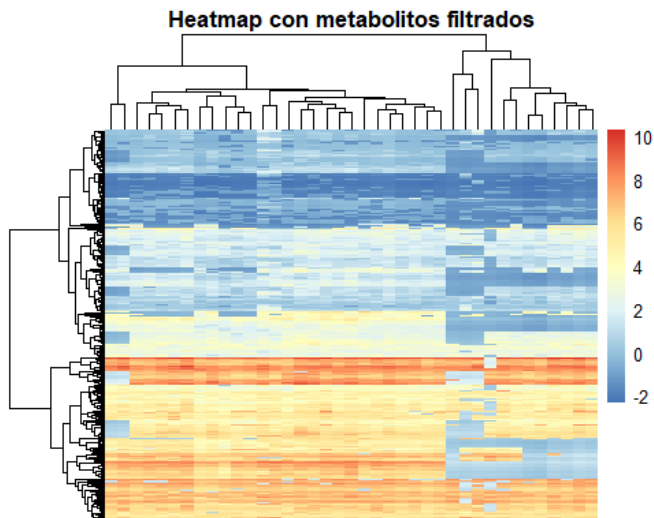
Vamos a revisar el número de metabolitos filtrados:

```
dim(summarized_experiment_filtered)
## [1] 513 39
```

Así pues, volveremos a hacer el heatmap con los 513 metabolitos resultantes:

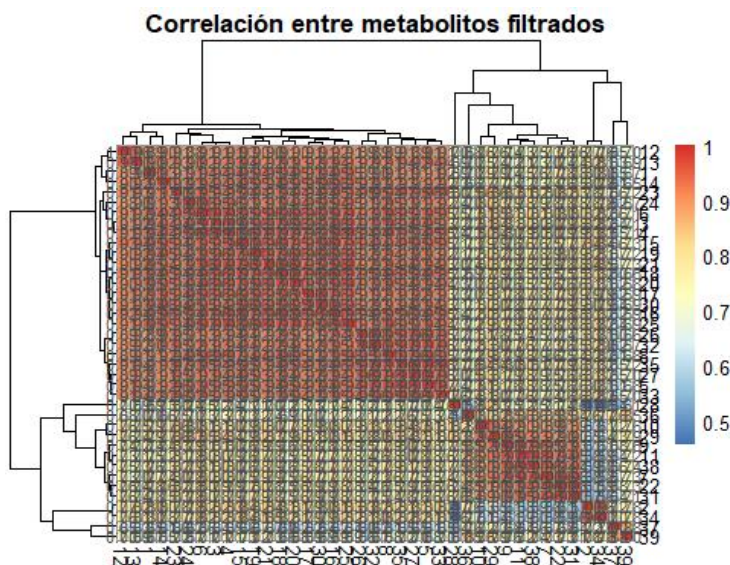
```
pheatmap(assay(summarized_experiment_filtered), cluster_rows = TRUE,
cluster_cols = TRUE,
```

```
show_rownames = FALSE, show_colnames = FALSE, main = "Heatmap  
con metabolitos filtrados")
```



Con este *heatmap* podemos observar que parece que tanto las muestras como los metabolitos se agrupan por clusteres. Esto podría indica funciones similares en los metabolitos y posibles factores experimentales que las expliquen.

```
cor_matrix_filtered <- cor(assay(summarized_experiment_filtered))  
pheatmap(cor_matrix_filtered, main = "Correlación entre metabolitos  
filtrados", display_numbers = TRUE)
```



Observamos una clara correlación entre los metabolitos de la parte superior izquierda, indicando nuevamente posibilidad de vías metabólicas similares. Esta información podría ampliarse utilizando los clústeres generados.

También observamos todo lo contrario, metabólicos con muy poca correlación, sobretodo en la parte inferior derecha, de color azul.

Discusión del estudio.

En el estudio original se identifican patrones metabólicos específicos que sirven para predecir los resultados metabólicos postquirúrgicos de los pacientes. Así pues, indican que la salud metabólica individual tiene una gran responsabilidad en las cirugías bariátricas.

En los primeros pasos de nuestro estudio hemos eliminado los valores negativos y posibles *outliers*, así como la normalización de las muestras, consiguiendo mejorar la calidad de estos.

Así mismo, el estudio de componentes principales nos enseña que, pese a que el primer componente explica la mayor variabilidad, no parece que hayan correlaciones altas entre las variables.

También se ha realizado un filtrado de los metabolitos para poder visualizar más cómodamente los heatmaps. Estos parecen indicar funciones similares entre metabolitos. Así mismo, las zonas de metabolitos sin correlación podría estar relacionada con el estudio original y la importancia de las individualidades metabólicas en las intervenciones de cirugía bariátrica.

Así pues, sería necesario seguir realizando más pruebas para llegar a las mismas conclusiones que el estudio principal, como podría ser modelos de regresión, análisis longitudinales y una identificación de las rutas metabólicas de los clústeres principales.

Repositorio de GitHub.

https://github.com/guachinwey/PCA1_Omics

APÉNDICE.

Capturas de creación de repositorio en GitHub y de vinculación con el proyecto en R:

