Fuertes-Perez-Sandra-PEC1

Sandra Fuertes Perez

2024-11-04

1. Seleccionar un dataset de metabolómica que podéis obtener de

- Este repositorio de github: https://github.com/nutrimetabolomics/metaboData/
- Si lo preferís podéis usar algún dataset del repositorio metabolomicsWorkbench

2. Una vez descargados los datos cread un contenedor del tipo SummarizedExperiment que contenga los datos y los metadatos (información acerca del dataset, las filas y las columnas). La clase SummarizedExperiment es una extensión de ExpressionSet y muchas aplicaciones o bases de datos (como metabolomicsWorkbench) lo utilizan en vez de usar expressionSet.

3. Llevad a cabo una exploración del dataset que os proporcione una visión general del mismo en la línea de lo que hemos visto en las actividades

4. Elaborad un informe que describa el proceso que habéis realizado, incluyendo la descarga de los datos, la creación del contenedor, la exploración de los datos y la reposición de los datos en github. El nombre del repositorio tiene que ser el siguiente: APELLIDO1-Apellido2-Nombre-PEC1. Por ejemplo, en mi caso el repositorio se llamaría: "Sanchez-Pla-Alex-PEC1"

5. Cread un repositorio de github que contenga

- el informe
- el objeto contenedor con los datos y los metadatos en formato binario (.Rda)
- el código R para la exploración de los datos
- los datos en formato texto
- los metadatos acerca del dataset en un archivo markdown

La dirección (url) del repositorio deberá estar incluida en la última sección del informe de forma clara. Observad que, para entregar vuestra PEC tenéis que entregar únicamente el informe. El resto de entregables de vuestra PEC deberán quedar en el repositorio de github.

//

Estructura de los resultados

La distribucion de los archivos es la siguiente:

- El archivo "README.md" contiene las instrucciones de la actividad
- Este informe "Fuertes-Perez-Sandra-PEC1.pdf" contiene la explicacion del proceso llevado a cabo en el analisis exploratorio de los datos

- El archivo "Fuertes-Perez-Sandra-PEC1.Rmd consiste en el archivo markdown que ha generado este informe, y el codigo utilizado durante la actividad
- La carpeta RawData contiene los archivos con los datos a utilizar descargados del repositorio de github sin modificacion alguna
- Los archivos "data.csv", "colData.csv" y "rowData.csv" contienen los datos extraidos con los que se ha generado el objeto SummarizedExperiment (sin pre-procesar)
- El archivo "metadata.md" contiene los metadatos sobre el estudio
- El archivo "summarized_experiment.rda" contiene el objeto SummarizedExperiment creado (antes del pre-procesado)

Eleccion de dataset y descarga de datos

Se ha escogido el database "2023-CIMCBTutorial".

Samples: 140

Features: 149

Description: NMR data from a gastric cancer study used in a metabolomics data analysis tutorial ("Basic Metabolomics Data Analysis Workflow" (https://cimcb.github.io/MetabWorkflowTutorial/Tutorial1.html)

Se ha creado inicialmente un repositorio en GitHub dedicado a almacenar todos los archivos de trabajo de la PEC1. Además, se ha descargado dicho repositorio para trabajar de forma local desde RStudio.

Para la obtencion de los datos a utilizar, se ha ejecutado el siguiente codigo para la descarga directa del archivo excel, que contiene todos los datos, y el archivo ".md", que contiene la descripcion del estudio. Los datos se descargan en el directorio *RawData* dentro del directorio de trabajo, que en este caso es el que alberga al repositorio de la PEC1.

*Nota. La descarga utilizando la URL en la barra del navegador generaba archivos vacios o con el contenido HTML. En su lugar, se utilizan las URL "raw" para acceder al contenido de los archivos.

```
# instalacion de paquetes requeridos
if (!require(BiocManager)) install.packages("BiocManager", dep=TRUE)
if (!require(SummarizedExperiment)) BiocManager::install("SummarizedExperiment")
if (!require(POMA)) BiocManager::install("POMA")
if (!require(magrittr)) install.packages("magrittr")
if (!require(ggtext)) install.packages("ggtext")
if (!require(gplots)) install.packages("gplots")
if (!require(dplyr)) install.packages("dplyr")
if (!require(dplyr)) install.packages("readxl")
```

```
# url de cada archivo
xlsx_file <- "https://raw.githubusercontent.com/nutrimetabolomics/metaboData/main/Datasets/2023-CIMCBTu
description_file <- "https://raw.githubusercontent.com/nutrimetabolomics/metaboData/main/Datasets/2023-

# crea un directorio de almacenaje de los datos raw
dir.create("RawData", showWarnings=FALSE)

# descarga de archivos
download.file(xlsx_file, destfile = "RawData/GastricCancer_NMR.xlsx", mode="wb")
download.file(description_file, destfile = "RawData/description.md")
```

Los datos numericos los cargamos de las dos pestañas disponibles en el archivo excel *Gastric-Cancer_NMR.xlsx*. Utilizamos la funcion *read_excel()* del paquete *readxl*.

```
# carga de datos del archivo excel
main_data <- readxl::read_excel("RawData/GastricCancer_NMR.xlsx")
peak_data <- readxl::read_excel("RawData/GastricCancer_NMR.xlsx", sheet="Peak")
```

Creacion de objeto SummarizedExperiment

Para utilizar la clase *SummarizedExperiment*, cargamos la libreria previamente instalada del mismo nombre, contenida en Bioconductor.

```
# carga de libreria
library(SummarizedExperiment)
```

Para generar nuestro objeto *SummarizedExperiment* necesitamos un total de 3 matrices, una para los datos de muestras y variables, una para la informacion sobre las columnas, y una para la informacion sobre las filas.

La principal matriz de datos en este caso estará compuesta por 140 muestras, que estaran distribuidas en las columnas, y 149 variables, o en este caso metabolitos medidos, que se distribuiran en las filas. Estos datos podemos obtenerlos de la matriz *main_data*, extrayendo todas las filas, y las columnas 5 a 153 (metabolitos M1-M149). Además, transpondremos filas y columnas para dejar las "probes" o variables medidas como filas, y las muestras como columnas. Nuestra matriz de datos es *data_matrix*.

```
# creamos nuestra matriz principal de datos
data_matrix <- t(main_data[,c(5:ncol(main_data))])
colnames(data_matrix) <- main_data$SampleID
```

La matriz que contiene la informacion sobre las columnas (*coldata*), es decir, las muestras, podemos obtenerlo tambien de la matriz *main_data*, de las columnas iniciales, con informacion sobre el ID de la muestra, el tipo de muestra (quality control QC o muestra de estudio) y la clase, que indica el resultado clinico observado (GC = cancer gastrico, BN = tumor benigno, HE = control sano). Asi, podemos extraer dicha informacion de las columnas 2 a 4. En esta nueva matriz, las filas contienen las muestas, y las columnas las variables descriptivas de las mismas.

Realizamos un paso similar para la matriz de informacion de las filas (*rowdata*), pero en este caso extraemos los datos de la matriz *peak_data*. Obtenemos los metabolitos como filas, y en las columnas los descriptores, incluyendo el nombre unico del metabolito (Label), porcentaje de muestras sin medida para este metabolito o NAs (Perc_missing), y una puntuacion de calidad que representa la variacion asociada a la cuantificacion del metabolito en las muestras (QC_RSD).

```
# creamos metadatos para filas y columnas
rowdata <- peak_data[,3:ncol(peak_data)]
rownames(rowdata) <- peak_data$Name
coldata <- main_data[,3:4]

# convertimos las variables "Class" y "SampleType" a factor
coldata$Class <- factor(coldata$Class)
coldata$SampleType <- factor(coldata$SampleType)

rownames(coldata) <- main_data$SampleID
```

Creamos el objeto *SummarizedExperiment* proveyendo a la funcion con la matriz de datos, y las dos matrices de metadatos para filas y columnas.

```
# creamos SummarizedExperiment object
se <- SummarizedExperiment(assays=list(metabolites=data_matrix), colData=coldata, rowData=rowdata)
```

Los metadatos sobre el estudio estan contenidos en el archivo *description.md*. Junto con informacion que se puede obtener de la publicacion, podemos rellenar una serie de campos explicativos sobre el estudio y la procedencia de los datos, y almacenarlos en el campo *metadata* del objecto *SummarizedExperiment*. Ademas, podemos tambien incluir informacion sobre los valores de las variables.

```
# extremos el contenido del archivo 'description.md'
raw_description = paste(readLines("RawData/description.md"), sep='\n')

# creamos el campo metadata del objeto se
metadata(se) <- list(raw_description = raw_description,
                     data = "Quantification of 149 metabolites (M1-M149) in 140 patient samples",
                     coldata = "SampleID\nSampleType: quality control QC or patient sample Sample\nClass
                     rowdata = "Name: column header of metabolite\nLabel: unique name for metabolite\nP
                     database_name = "2023-CIMCBTutorial",
                     description = "NMR data from a gastric cancer study used in a metabolomics data ana
                     publication_title = "1H-NMR urinary metabolomic profiling for diagnosis of gastric
                     authors = "Chan, A., Mercier, P., Schiller, D. et al.",
                     publication_DOI = "https://doi.org/10.1038/bjc.2015.414",
                     journal = "British Journal of Cancer",
                     issue = "114",
                     pages = "59-62",
                     year = "2016",
                     annotated_data_file = "Metabolomics Workbench data repository\nProject ID PR000699
                     project_DOI = "10.21228/M8B10B",
                     centre = "Canada's National High Field Nuclear Magnetic Resonance Centre (NANUC)",
                     equipment = "600 MHz Varian Inova spectrometer")
```

Guardado de datos

Los datos distribuidos en las tres matrices/dataframes se guardan como archivos de texto *.csv*. El objeto SummarizedExperiment se guarda entero junto con todos sus campos como un archivo *.rda*.

```
# guardamos nuestros datos y el summarizedexperiment object
save(se, file = "summarized_experiment.rda")
write.csv(data_matrix, file="data.csv")
write.csv(coldata, file="colData.csv")
write.csv(rowdata, file="rowData.csv")
```

Los metadatos del archivo se guardaran en un archivo markdown.Podemos extraer los metadatos con sus encabezados del objeto *SummarizedExperiment* y copiarlos en un archivo *metadata.md*.

```
# extraemos metadatos en un archivo .md
metadata <- c("")
for (name in names(metadata(se))) {
  metadata <- append(metadata, paste(c(paste(name, ':'), metadata(se)[[name]], '\n')))
```

```
}

cat(paste(metadata, collapse="\n"), file="metadata.md")
```

Exploracion del dataset

```
# cargamos librerias para la exploracion de datos
library(POMA)
library(magrittr)
library(ggtext)
```

En primer lugar, podemos llevar a cabo una exploracion simple, meramente de la distribucion de los datos.

Observamos en la distribucion de los datos (valores minimo, maximo, media y mediana) que las escalas para cada metabolito no son, en muchos casos, comparables. Seria adecuado normalizar los valores antes de proceder con algun analisis.

```
# con la funcion str() podemos visualizar la cantidad de datos contenidos en el objeto se
str(se)
```

```
## Formal class 'SummarizedExperiment' [package "SummarizedExperiment"] with 5 slots
##   ..@ colData        :Formal class 'DFrame' [package "S4Vectors"] with 6 slots
##   .. .. ..@ rownames       : chr [1:140] "sample_1" "sample_2" "sample_3" "sample_4" ...
##   .. .. ..@ nrows          : int 140
##   .. .. ..@ elementType    : chr "ANY"
##   .. .. ..@ elementMetadata: NULL
##   .. .. ..@ metadata       : list()
##   .. .. ..@ listData       :List of 2
##   .. .. .. ..$ SampleType: Factor w/ 2 levels "QC","Sample": 1 2 2 2 2 2 2 2 2 1 ...
##   .. .. .. ..$ Class     : Factor w/ 4 levels "BN","GC","HE",..: 4 2 1 3 2 1 2 3 2 4 ...
##   ..@ assays         :Formal class 'SimpleAssays' [package "SummarizedExperiment"] with 1 slot
##   .. .. ..@ data:Formal class 'SimpleList' [package "S4Vectors"] with 4 slots
##   .. .. .. .. ..@ listData       :List of 1
##   .. .. .. .. .. ..$ metabolites: num [1:149, 1:140] 90.1 491.6 202.9 35 164.2 ...
##   .. .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. .. .. .. ..$ : chr [1:149] "M1" "M2" "M3" "M4" ...
##   .. .. .. .. .. .. .. ..$ : chr [1:140] "sample_1" "sample_2" "sample_3" "sample_4" ...
##   .. .. .. .. ..@ elementType    : chr "ANY"
##   .. .. .. .. ..@ elementMetadata: NULL
##   .. .. .. .. ..@ metadata       : list()
##   ..@ NAMES          : chr [1:149] "M1" "M2" "M3" "M4" ...
##   ..@ elementMetadata:Formal class 'DFrame' [package "S4Vectors"] with 6 slots
##   .. .. ..@ rownames       : NULL
##   .. .. ..@ nrows          : int 149
##   .. .. ..@ elementType    : chr "ANY"
##   .. .. ..@ elementMetadata: NULL
##   .. .. ..@ metadata       : list()
##   .. .. ..@ listData       :List of 3
##   .. .. .. ..$ Label       : chr [1:149] "1_3-Dimethylurate" "1_6-Anhydro- -D-glucose" "1_7-Dimethylx
##   .. .. .. ..$ Perc_missing: num [1:149] 11.429 0.714 5 8.571 1.429 ...
```

```
##   .. .. .. ..$ QC_RSD      : num [1:149] 32.21 31.18 34.99 12.8 9.37 ...
##   ..@ metadata       :List of 17
##   .. ..$ raw_description    : chr [1:15] "Dataset used in the CIMBC tutorial on [\"Basic Metabolomics
##   .. ..$ data               : chr "Quantification of 149 metabolites (M1-M149) in 140 patient samples
##   .. ..$ coldata            : chr "SampleID\nSampleType: quality control QC or patient sample Sample\
##   .. ..$ rowdata            : chr "Name: column header of metabolite\nLabel: unique name for metaboli
##   .. ..$ database_name      : chr "2023-CIMCBTutorial"
##   .. ..$ description        : chr "NMR data from a gastric cancer study used in a metabolomics data a
##   .. ..$ publication_title  : chr "1H-NMR urinary metabolomic profiling for diagnosis of gastric canc
##   .. ..$ authors            : chr "Chan, A., Mercier, P., Schiller, D. et al."
##   .. ..$ publication_DOI    : chr "https://doi.org/10.1038/bjc.2015.414"
##   .. ..$ journal            : chr "British Journal of Cancer"
##   .. ..$ issue              : chr "114"
##   .. ..$ pages              : chr "59-62"
##   .. ..$ year               : chr "2016"
##   .. ..$ annotated_data_file: chr "Metabolomics Workbench data repository\nProject ID PR000699"
##   .. ..$ project_DOI        : chr "10.21228/M8B10B"
##   .. ..$ centre             : chr "Canada's National High Field Nuclear Magnetic Resonance Centre (NA
##   .. ..$ equipment          : chr "600 MHz Varian Inova spectrometer"
```

```r
# realizamos un resumen por filas (por metabolitos)
head(apply(assay(se, i=1), 1, summary))
```

```
## $M1
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.40   29.82   60.35  101.07  133.38  909.90      16
##
## $M2
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     3.1   140.9   270.2   642.0   480.9 26195.8       1
##
## $M3
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     0.1    53.6   105.1   146.4   198.8   862.5       7
##
## $M4
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.10   18.77   35.70   43.83   51.33  242.50      12
##
## $M5
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##     1.3    67.0   160.3   231.1   253.1  2503.0       2
##
## $M6
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.20   15.50   25.90   41.63   48.60  339.40       7
```

Imputación de NAs

Entre la informacion que podemos extraer del resumen, es que hay varios missing datapoint en la mayoria de los metabolitos de la base de datos. Si extraemos una tabla resumen del porcentaje de missing data contenidos en los metadatos de las filas (*rowData()*), observamos que para algunos metabolitos el porcentaje de datos perdidos supera el 30%.

```
# resumen del porcentaje de datos missing de los metabolitos
summary(rowData(se)$Perc_missing)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.7143  2.1429  5.1246  6.4286 31.4286
```

Un 20% de muestras sin valor para un metabolito se considera criterio suficiente para eliminarlo del analisis (Wei, Wang, Su, et al., 2018). Eliminamos los metabolitos que no cumplan dicho criterio e imputamos el resto. Hay que considerar que la funcion *PomaImpute()* no almacena el rowData del objeto original, de forma que debe ser manualmente añadido despues.

```
# guarda el rowData original
se_rowdata <- rowData(se)

# elimina NAs por encima del cutoff e imputa el resto de datos
se_imputed <- se[rowData(se)$Perc_missing <= 20]
se_imputed <- se_imputed %>% PomaImpute(method="knn", zeros_as_na=TRUE, remove_na=FALSE)

# integramos de nuevo el rowData en el nuevo objeto SE
rowData(se_imputed) <- se_rowdata[rownames(se_imputed),]

# resumen de la distribucion de NAs en el objeto
summary(rowData(se_imputed)$Perc_missing)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.5357  2.1429  3.8367  5.0000 20.0000
```

Observamos que ya no hay missing data presentes en los datos.

```
# resumen numerico de los 5 primeros metabolitos
head(apply(assay(se_imputed[1:5,], i=1), 1, summary))
```

```
##               M1         M2       M3        M4        M5
## Min.      0.4000     3.1000   0.1000   0.10000    1.3000
## 1st Qu.  31.2250   141.4500  47.6625  18.70000   67.0000
## Median   62.5250   269.8500 101.1000  34.85500  162.2000
## Mean    103.2469   639.0788 141.0978  42.91316  233.1621
## 3rd Qu. 134.3750   475.6250 195.1250  51.15000  253.6750
## Max.    909.9000 26195.8000 862.5000 242.50000 2503.0000
```

Control de calidad

Tambien podemos evaluar la puntuacion de calidad para cada metabolito contenida en los metadatos de las filas (*rowData*). La puntuacion esta basada en la desviacion estandar relativa (RSD), que establece si la desviacion estandar de cada metabolito es una cantidad pequeña o grande comparada con la media. Un criterio comun de aceptacion para el RSD es <20% (Sangster, 2016; Dunn, 2011b). En el resumen vemos que hay datos que pueden alcanzar un RSD de hasta 133. Podemos eliminar dichos metabolitos al no poder confiar en su cuantificacion.

```
# resumen de la puntuacion de calidad de los metabolitos
summary(rowData(se_imputed)$QC_RSD)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.432  12.968  23.945  27.378  38.803 112.592
```

```
# eliminar metabolitos con QC > 20
se_quality <- se_imputed[rowData(se_imputed)$QC_RSD <= 20]

# resumen de la puntuacion de calidad tras el filtrado
summary(rowData(se_quality)$QC_RSD)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   2.432   5.582   9.436  10.145  14.490  19.146
```

Tras todo el proceso de eliminacion de datos de poca calidad, mantenemos 52 metabolitos.

```
# dimension de la base de datos
dim(se_quality)
```

```
## [1]  52 140
```

Normalizacion

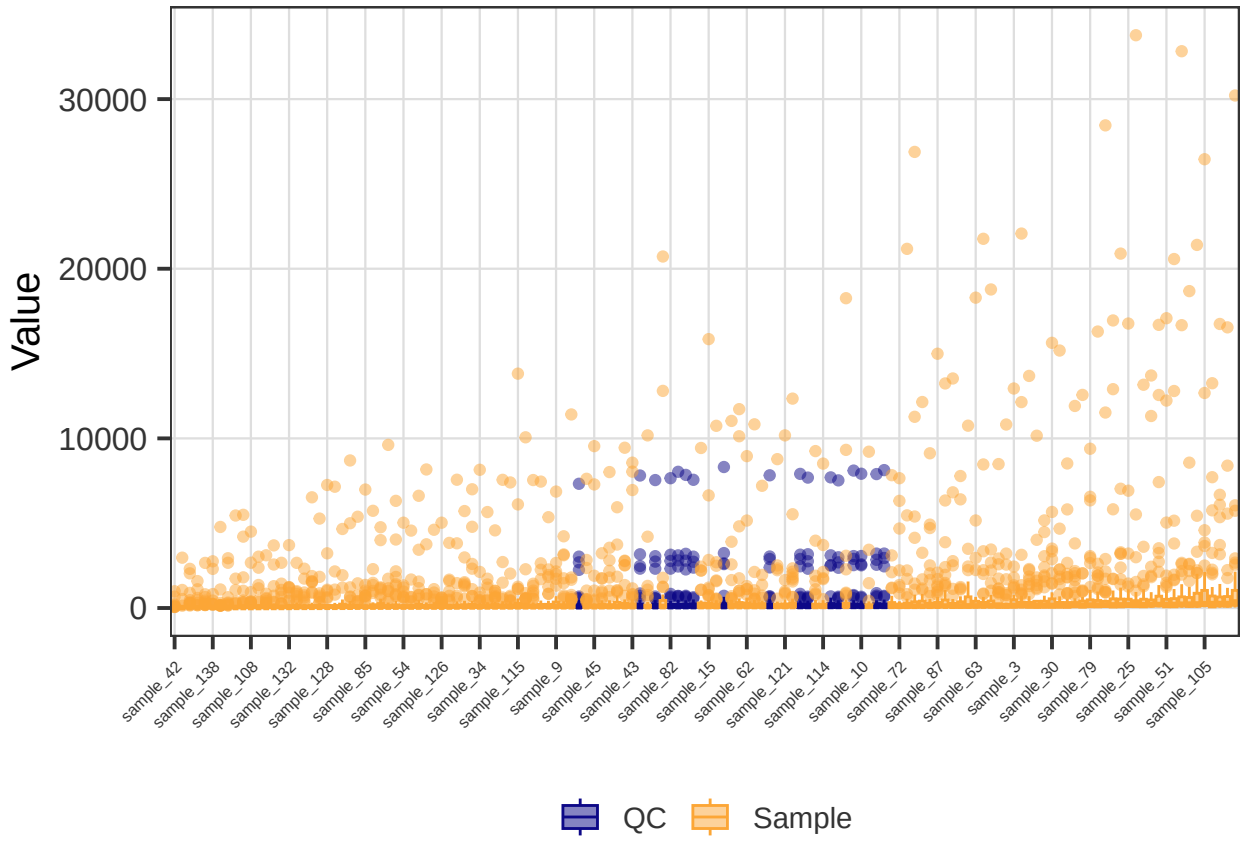
Aplicamos una normalizacion de los valores

```
# normalizacion de los datos
se_normalized <- se_quality %>% PomaNorm(method = "log_pareto")

# integramos de nuevo el rowData en el nuevo objeto SE
rowData(se_normalized) <- se_rowdata[rownames(se_normalized),]

apply(assay(se_normalized[1:5,], i=1), 1, summary)
```

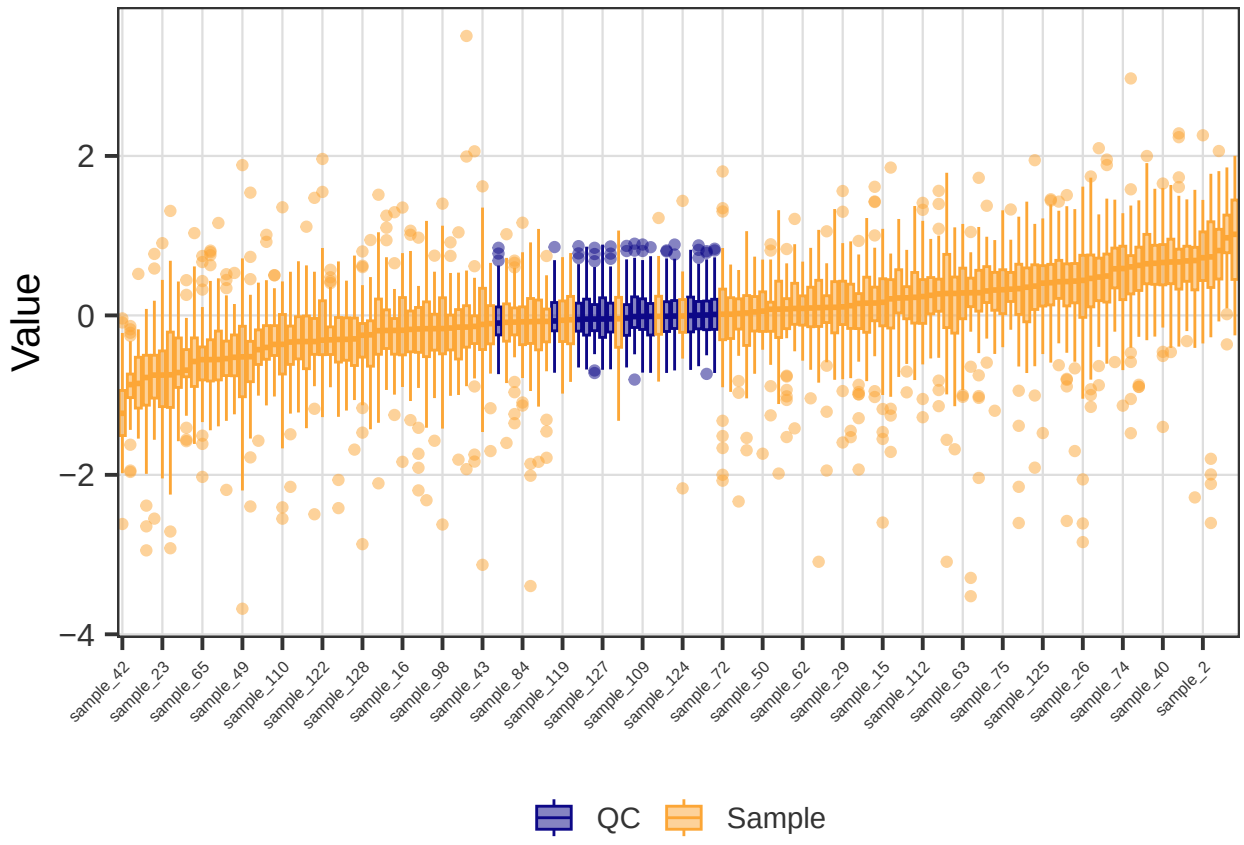
```
##                    M4            M5            M7            M8           M11
## Min.    -2.334253e+00 -2.623074e+00 -1.352527e+00 -1.326854e+00 -2.578499e+00
## 1st Qu. -3.245605e-01 -4.538011e-01 -4.847002e-01 -2.776447e-01 -3.733797e-01
## Median   9.255835e-02  1.069382e-01 -3.504414e-02 -3.012138e-02 -2.814049e-02
## Mean     1.730356e-16 -7.872232e-17  1.152654e-16 -7.078006e-17 -1.514987e-16
## 3rd Qu.  3.535104e-01  3.920257e-01  2.697219e-01  3.030387e-01  4.089981e-01
## Max.     1.426853e+00  1.856096e+00  1.626286e+00  1.823215e+00  2.002075e+00
```

El paquete POMA pone a disposicion funciones para la visualizacion de los datos en formar de boxplots. Podemos representar el efecto de la normalizacion visualizando los datos antes y despues de llevarla a cabo. Nuestro datos estan ahora centrados alrededor de una media 0. Podemos observar la distribucion aparentemente aleatoria de los tipos de muestras, con la excepcion de la localizacion de las muestras de control de calidad en el centro, con poca variacion de valor como se esperaria, que indican consistencia. Las muestras antes de la normalizacion muestran signos de skewness, con valores para algunas de ellas muy extremos que, tras la normalizacion, son suavizados.

```r
# boxplots de las muestras antes y despues de la normalizacion
library(ggplot2)

# pre-normalizacion
plot1 <- PomaBoxplots(se_quality, x="samples")
plot1 + scale_x_discrete(breaks = function(x) x[seq(1, length(x), by = 5)]) + theme(axis.text.x = eleme
```

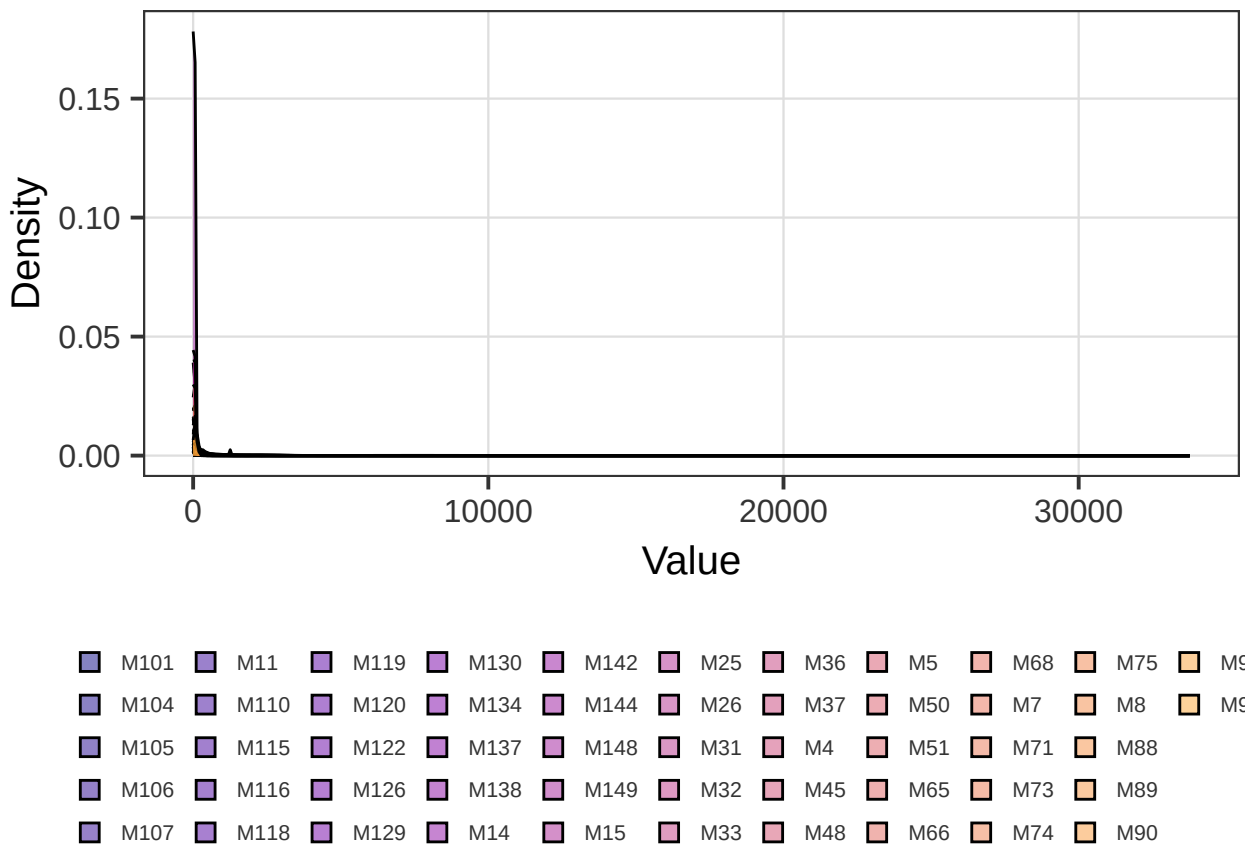



```r
# post-normalizacion
plot2 <- PomaBoxplots(se_normalized, x="samples")
plot2 + scale_x_discrete(breaks = function(x) x[seq(1, length(x), by = 5)]) + theme(axis.text.x = eleme
```

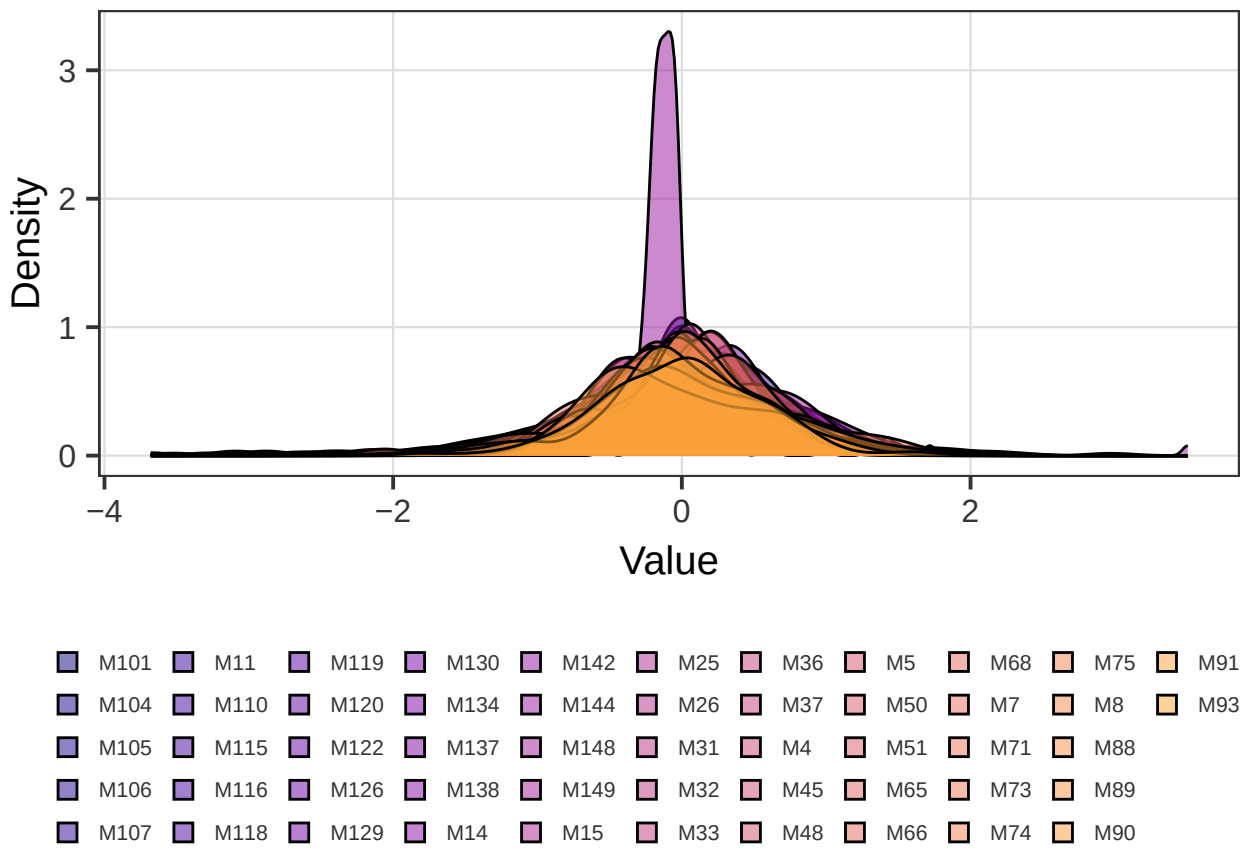


Tambien es posible observar las curvas de densidad de valores para las variables (metabolitos). Observamos una clara diferencia entre los valores con mayor densidad pre-normalizacion, agrupados en los valores bajos y, aunque no visibles, outliers muy extremos en valores altos. Por otra parte al normalizar los valores, observamos que los picos de casi todos los metabolitos se solapan. Uno de los picos parece de mayor altura que el resto. Podria indicar un feature dominante, o podria indicar la presentcia de outliers extremos en la medida de dicho metabolito.

```
# creacion de curva de densidad de valores antes de la normalizacion
plot1 <- PomaDensity(se_quality, x="features")
plot1 + theme(
  legend.text = element_text(size=8),
  legend.key.size = unit(3, "mm")) + guides(fill=guide_legend(ncol=12))
```




```
# creacion de curva de densidad de valores tras la normalizacion
plot2 <- PomaDensity(se_normalized, x="features")
plot2 + theme(
  legend.text = element_text(size=8),
  legend.key.size = unit(3, "mm")) + guides(fill=guide_legend(ncol=12))
```



Es posible extraer el indice del metabolito que crea el pico de mayor altura para poder estar atentos a él.

```
# extraccion del pico de mayor altura
library(dplyr)
p <- plot2$data
highest_peak_feature <- p %>%
  group_by(name) %>%
  summarize(max_density = max(value)) %>%
  slice_max(max_density, n = 1) %>%
  pull(name)

highest_peak_feature
```

```
## [1] "M144"
```

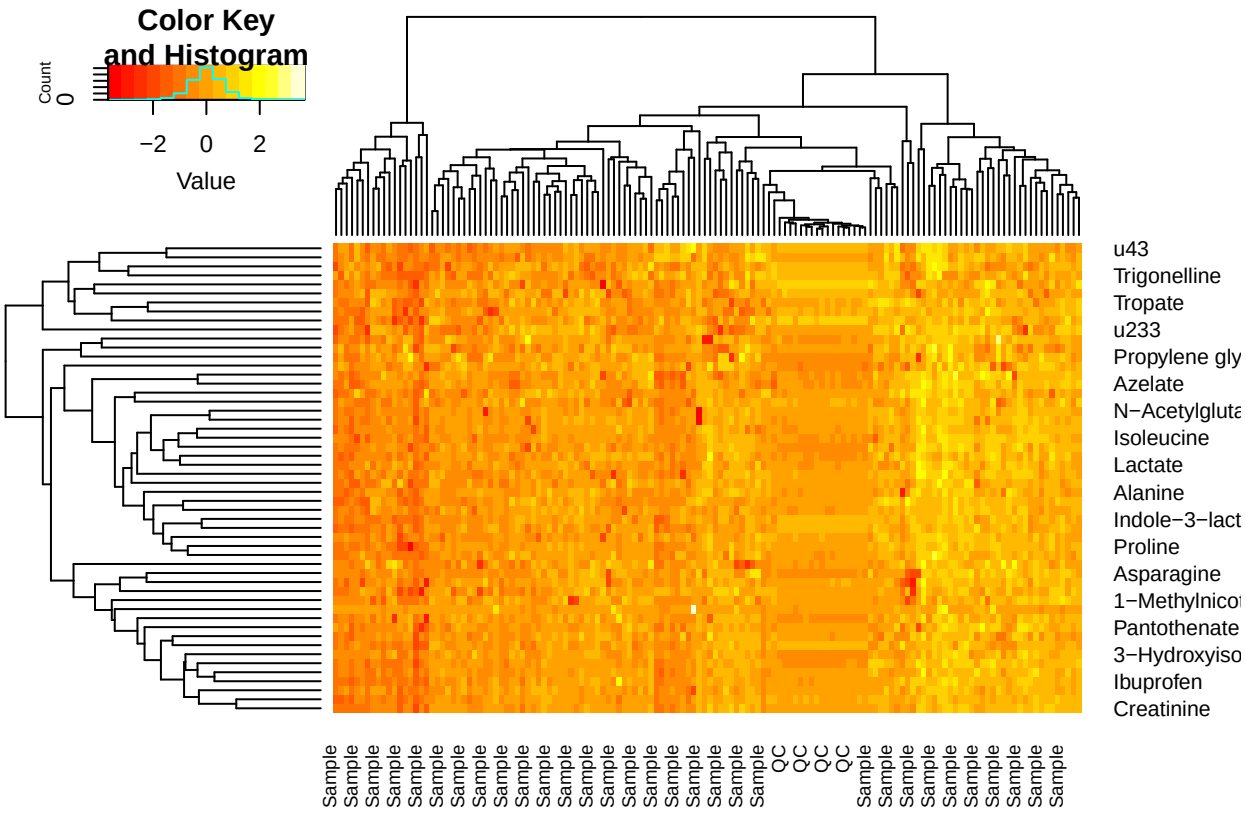
```
cat("El metabolito que produce el pico de mayor altura que el resto es ", rowData(se)[highest_peak_featu
```

```
## El metabolito que produce el pico de mayor altura que el resto es  u87
```

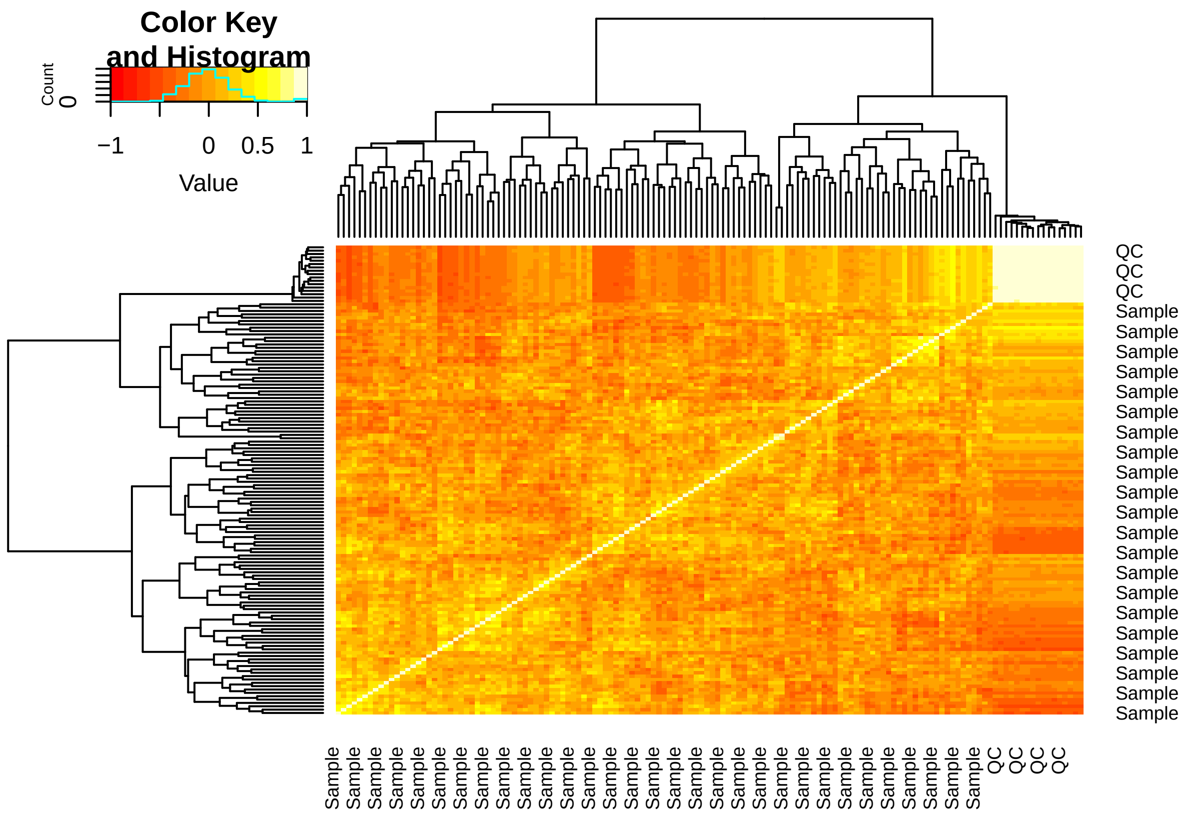
Relacion y correlacion

Con la libreria gplots es posible visualizar tanto los dendrogramas de muestras y features juntos, como la matriz de correlacion tanto de muestras como de features. Observamos que en la representacion de correlacion de las muestras, aquellas uqe pertenecen al control de calidad estan claramente diferenciadas del resto, lo cual es buena señal.

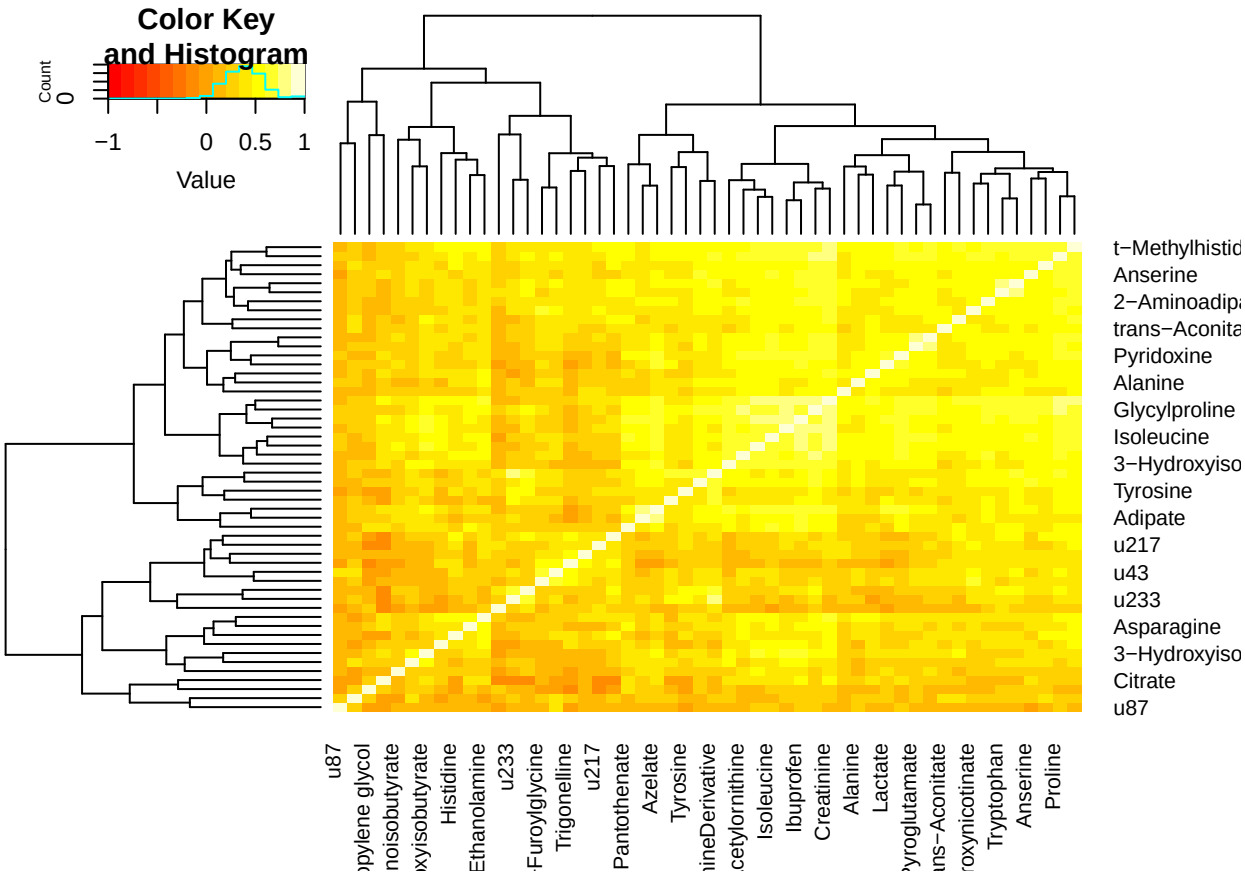
```r
# creacion de heatmaps para la relacion entre muestras y metabolitos, y la correlacion entre muestras, 
library(gplots)
data_heatmap <- assay(se_normalized)
rownames(data_heatmap) <- rowData(se_normalized)$Label
colnames(data_heatmap) <- colData(se_normalized)$SampleType

heatmap_result <- heatmap.2(data_heatmap, trace="none", dendrogram="both")
```




```r
heatmap_result <- heatmap.2(cor(data_heatmap), trace="none")
```



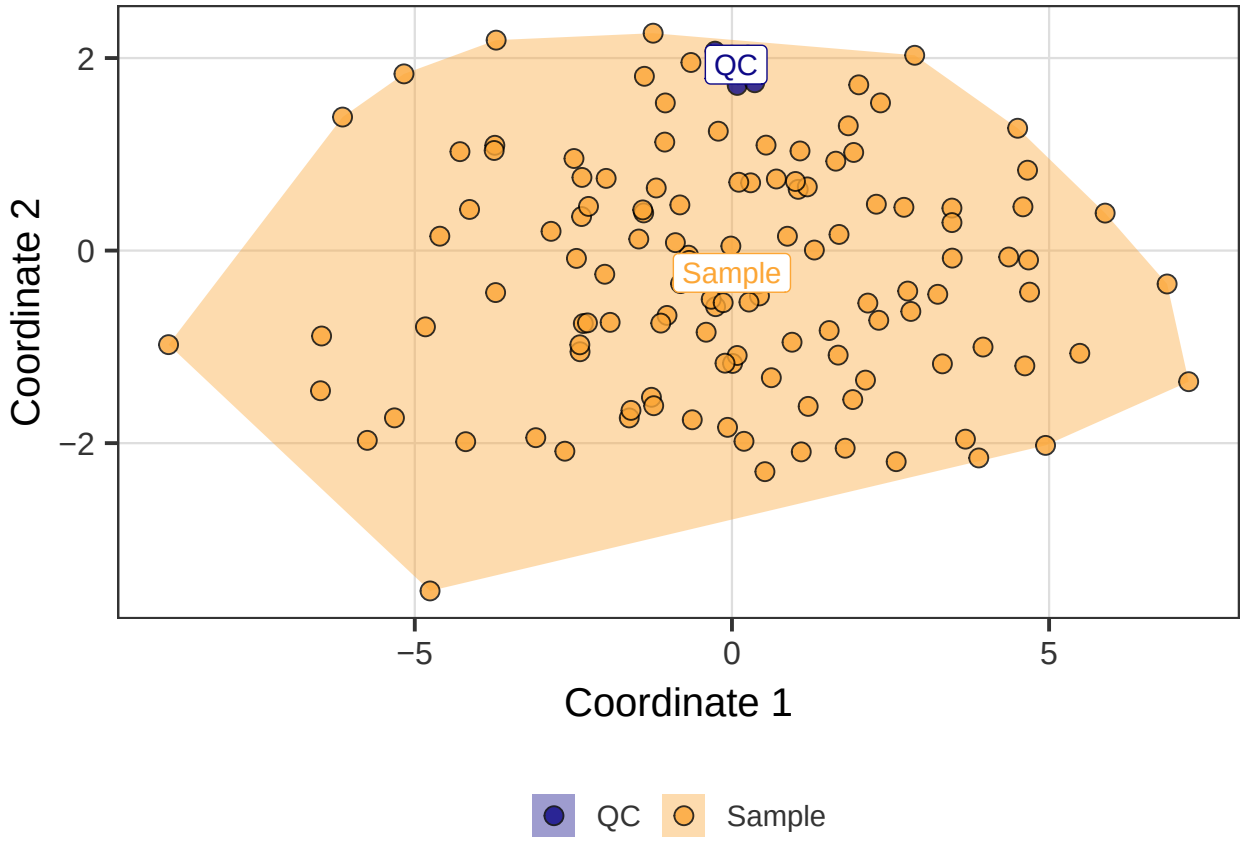

```r
heatmap_result <- heatmap.2(cor(t(data_heatmap)), trace="none")
```



Outliers

Con el siguiente codigo, tambien utilizando la libreria POMA, podemos eliminar y visualizar outliers en nuestros datos, donde vemos, de nuevo, que las muestras del control de calidad forman un cluster concentrado, indicando muy poca variabilidad. Tal y como se espera, la dispersion de las muestras es mucho mayor.

```
# extract the summarizedexperiment object without ouliers
se_outliers <- PomaOutliers(se_normalized)$data

# represent outliers detection
PomaOutliers(se_normalized)$polygon_plot
```



PCA

Podemos realizar un analisis de PCA de los datos. De nuevo, se agrupan en muestras de QC y biologicas. Las primeras deberian formar un cluster compacto comparadas con el resto de muestras.

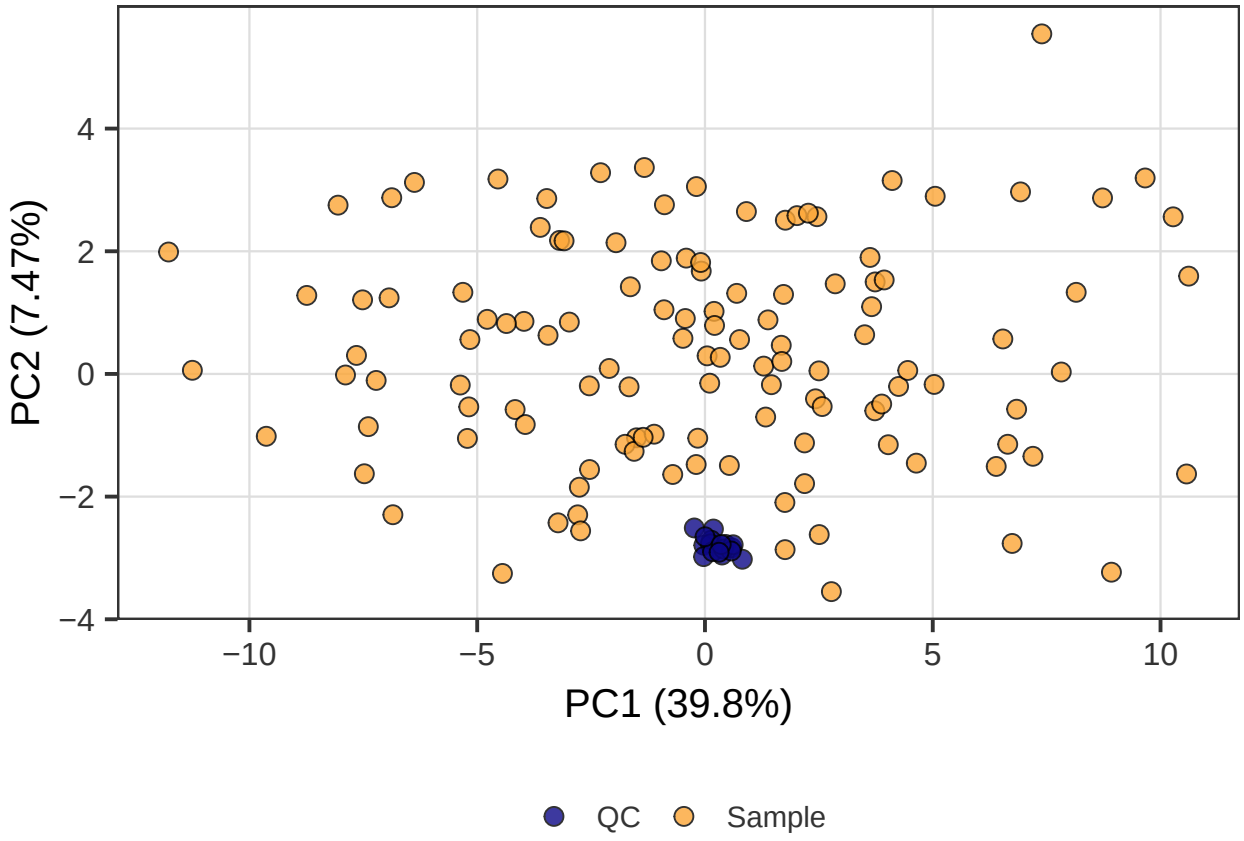
```
# perform PCA analysis of the data
PCA <- PomaPCA(se_outliers)

# extract eigenvalues
PCA$eigenvalues
```

```
## # A tibble: 4 x 2
##   comp  var_exp
##   <chr>   <dbl>
## 1 PC1     71.0
## 2 PC2     13.3
## 3 PC3      9.06
## 4 PC4      6.56
```

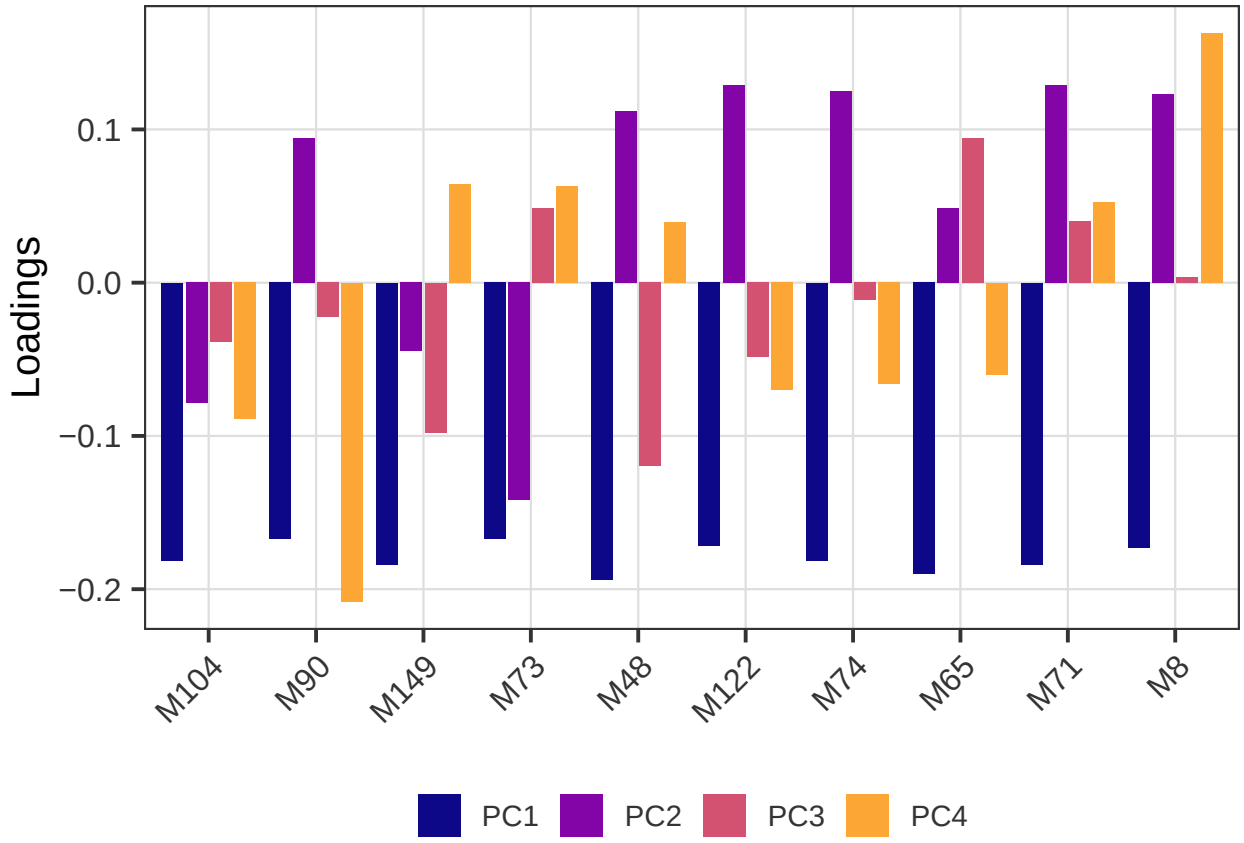
```
# extract loadings and the most contributing features for each PC
loadings <- PCA$loadings
max_load_pc1 <- loadings[which.max(abs(loadings$PC1)),]$feature
max_load_pc2 <- loadings[which.max(abs(loadings$PC2)),]$feature
rowdata[c(max_load_pc1, max_load_pc2), 1]
```

```
## # A tibble: 2 x 1
##   Label
##   <chr>
## 1 Creatinine
## 2 Citrate
```

```
# distribucion de las muestras en el espacio de las dos primeras PCs
PCA$factors_plot
```




```
# peso de cada metabolito a cada PC
PCA$loadings_plot
```



La componente principal contiene un 71% de la varianza, seguida de la segunda componente que engloba un 13% de la misma. Mas del 80% de la varianza esta explicada por las dos primeras componentes. Parece que los features que, de manera absoluta, tienen un peso mayor para las componentes primera y segunda son la creatinina y el citrato, respectivamente. La distribucion de las muestras en el plano de las dos primeras componentes muestra la dispersion de las muestras biologicas frente a la aglomeracion de las muestas control. Finalmente, el grafico de contribucion de cada metabolito muestra que las contribuciones de la mayoria de metabolitos a la primera componente son bastante similares, y es en el resto de componentes cuando parece que hay mayor variacion de la contribucion. Podria indicarse que no hay un unico metabolito que destaque en su contribucion a la mayoria de la variabilidad.

PCA entre muestras sanas y cancerosas

Podemos repetir el proceso anterior pero unicamente considerando aquellas muestras que sean o bien de tejido sano, o bien de tejido canceroso. En este caso, la PC1 y PC2 tambien engloban la mayoria de la variabilidad, con un 74% perteneciendo a la primera. En la grafica de factores podemos ver que podria haber una cierta separacion entre las ambos tipos de muestras. En la grafica de contribuciones vemos que de nuevo hay una contribucion bastante similar de multiples metabolitos al mismo tiempo.

```
# recuperamos el SummarizedExperiment
se_HvC <- se[,colData(se)$Class %in% c("HE", "GC")]
colData(se_HvC) <- colData(se_HvC)[,c("Class", "SampleType")]

# imputacion de datos
imputed <- se_HvC[rowData(se_HvC)$Perc_missing <= 20]
imputed <- imputed %>% PomaImpute(method="knn", zeros_as_na=TRUE, remove_na=FALSE)
```

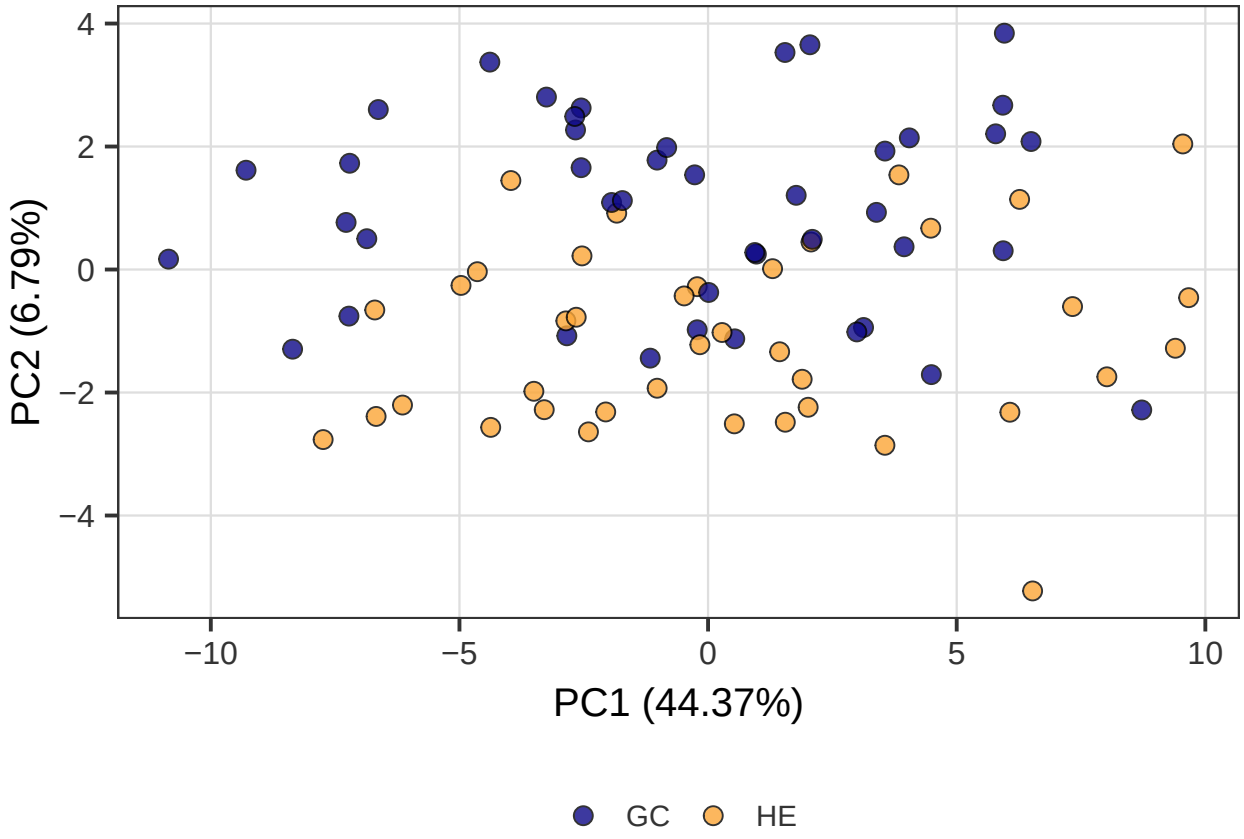
```r
rowData(imputed) <- se_rowdata[rownames(imputed),]

# eliminacion de metabolitos de baja calidad
quality <- imputed[rowData(imputed)$QC_RSD <= 20]

# normalizacion
normalized <- quality %>% PomaNorm(method = "log_pareto")
rowData(normalized) <- se_rowdata[rownames(normalized),]

# eliminacion de outliers
outliers <- PomaOutliers(normalized)$data

# analisis PCA en base a la clase de la muestra
PCA <- PomaPCA(outliers)

# extract eigenvalues
PCA$eigenvalues
```

```
## # A tibble: 4 x 2
##   comp  var_exp
##   <chr>   <dbl>
## 1 PC1     74.6
## 2 PC2     11.4
## 3 PC3      7.98
## 4 PC4      5.95
```

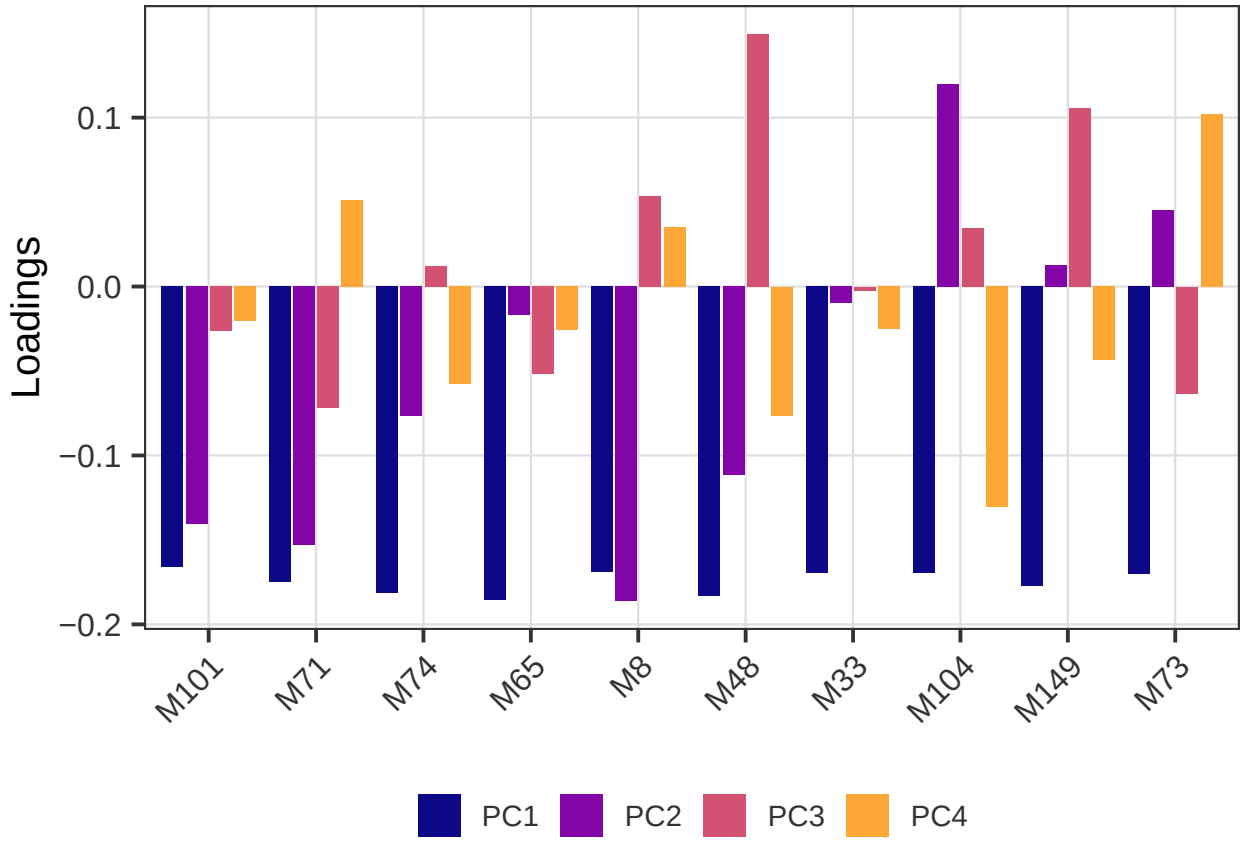
```r
# extract loadings and the most contributing features for each PC
loadings <- PCA$loadings
max_load_pc1 <- loadings[which.max(abs(loadings$PC1)),]$feature
max_load_pc2 <- loadings[which.max(abs(loadings$PC2)),]$feature
rowdata[c(max_load_pc1, max_load_pc2), 1]
```

```
## # A tibble: 2 x 1
##   Label
##   <chr>
## 1 Glycylproline
## 2 Trigonelline
```

```r
# distribucion de las muestras en el espacio de las dos primeras PCs
PCA$factors_plot
```




```
# peso de cada metabolito a cada PC
PCA$loadings_plot
```




```
# pesos de cada metabolito
summary(PCA$loadings[,"PC1"])
```

```
##       PC1
##  Min.   :-0.18551
##  1st Qu.:-0.16160
##  Median :-0.13839
##  Mean   :-0.13508
##  3rd Qu.:-0.11597
##  Max.   :-0.06789
```

URL al repositorio de GitHub

https://github.com/Chirimoyia/Fuertes-Perez-Sandra-PEC1.git

References

1. Sangster, T., Major, H., Plumb, R., Wilson, A. J., & Wilson, I. D. (2006). A pragmatic and readily implemented quality control strategy for HPLC-MS and GC-MS-based metabonomic analysis. Analyst, 131(10), 1075–1078.
2. Dunn, W. B., Broadhurst, D., Begley, P., Zelena, E., Francis-McIntyre, S., Anderson, N., et al. (2011b). Procedures for large-scale metabolic profiling of serum and plasma using gas chromatography and liquid chromatography coupled to mass spectrometry. Nature Protocols, 6(7), 1060–1083.

3. Wei, R., Wang, J., Su, M. et al. Missing Value Imputation Approach for Mass Spectrometry-based Metabolomics Data. Sci Rep 8, 663 (2018). https://doi.org/10.1038/s41598-017-19120-0