PEC1

Álvaro Linacero

2024-11-01

```
if (!requireNamespace("SummarizedExperiment", quietly = TRUE)) {
  install.packages("BiocManager")
  BiocManager::install("SummarizedExperiment")
}
if (!requireNamespace("readxl", quietly = TRUE)) {
  install.packages("readxl")
}
library(SummarizedExperiment)
```

```
## Cargando paquete requerido: MatrixGenerics
```

```
## Cargando paquete requerido: matrixStats
```

```
##
## Adjuntando el paquete: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars
```

```
## Cargando paquete requerido: GenomicRanges
```

```
## Cargando paquete requerido: stats4
```

```
## Cargando paquete requerido: BiocGenerics
```

```
##
## Adjuntando el paquete: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, table,
##     tapply, union, unique, unsplit, which.max, which.min

## Cargando paquete requerido: S4Vectors

##
## Adjuntando el paquete: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Cargando paquete requerido: IRanges

##
## Adjuntando el paquete: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Cargando paquete requerido: GenomeInfoDb

## Cargando paquete requerido: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Adjuntando el paquete: 'Biobase'
```

```
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library(readxl)
```

Extraer los conjuntos de datos del archivo en formato excel y preparar los conjuntos de datos que formarán el objeto SE.

```
Excel<- "C:/Users/Usuario/OneDrive - UNIVERSIDAD SAN JORGE/Escritorio/Bioinfor y Bioest/Análisis de dat
M <- as.matrix(read_excel(Excel, sheet = "Data", col_names = TRUE))
rownames(M) <- M[, 2]  # Define la primera columna como nombres de fila (M)
colData<- M[,2:4]
colData<- data.frame(colData)
colData$Class<- as.factor(colData$Class)
M_data <- M[, 5:153]
M_data <- apply(M_data, 2, function(x) as.numeric(trimws(x)))
M_data<- t(M_data)
M_peak <- as.matrix(read_excel(Excel, sheet = "Peak", col_names = TRUE))
rownames(M_peak)<- M_peak[, 2]
M_peak <- M_peak[, 3:5]
```

A continuación se crea el objeto Summaried Experiment, introduciendo el conjunto de "Datos" que contiene los valores corresponientes de concentración de cada metabolito para cada muestra como assay. El conjunto de datos "colData" será añadidio como colData y es un conjunto de datos que almacena metadatos de las muestras. "M_peak" se añade como rowData y almacena características de de cada metabolito.

```
se <- SummarizedExperiment(assays = list(counts = M_data), colData = colData, rowData = M_peak)
se
```

```
## class: SummarizedExperiment
## dim: 149 140
## metadata(0):
## assays(1): counts
## rownames(149): M1 M2 ... M148 M149
## rowData names(3): Label Perc_missing QC_RSD
## colnames(140): sample_1 sample_2 ... sample_139 sample_140
## colData names(3): SampleID SampleType Class
```

```
save(se, file = "SE.Rda")
Datos<- assay(se)
Caracteristicas<-colData(se)
Peak<- rowData(se)
#Generar conjuntos de datos en formato csv y txt
write.csv(as.data.frame(Datos), "Datos.csv", row.names = TRUE)
write.table(as.data.frame(Datos), "Datos.txt", sep = "\t", row.names = TRUE)
write.csv(as.data.frame(Caracteristicas), "Caracteristicas.csv", row.names = TRUE)
write.table(as.data.frame(Caracteristicas), "Caracteristicas.txt", sep = "\t", row.names = TRUE)
write.csv(as.data.frame(Peak), "Peak.csv", row.names = TRUE)
write.table(as.data.frame(Peak), "Peak.txt", sep = "\t", row.names = TRUE)
```

El objeto SE es el que debemos subir a hithub y del que partiremos para acceder a los datos.

```
dim(se)
```

```
## [1] 149 140
```

```
str(Datos)
```

```
##  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" ...
```

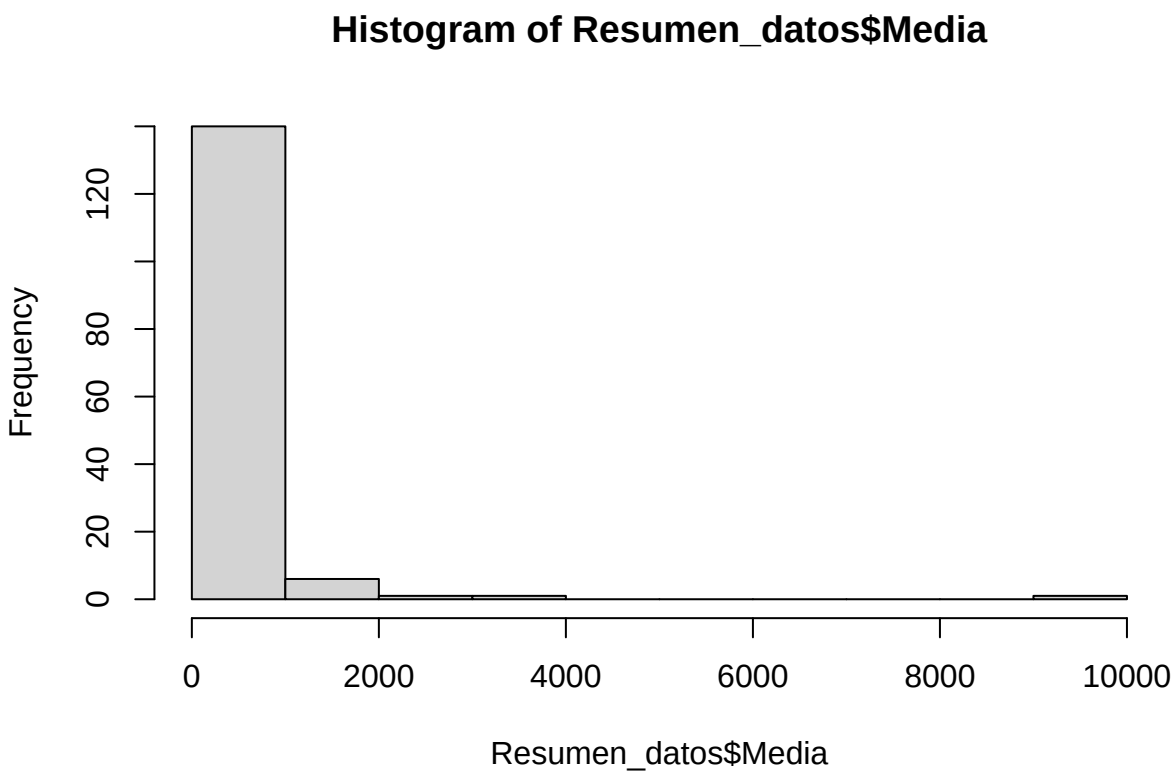
```
Resumen_datos <- data.frame(Media = rowMeans(Datos, na.rm = TRUE), Mediana = apply(Datos, 1, median, na
head(Resumen_datos)
```

```
##        Media Mediana Minimo  Maximo Desviacion
## M1 101.07097   60.35    0.4   909.9  123.61378
## M2 641.99784  270.20    3.1 26195.8 2397.53563
## M3 146.36692  105.10    0.1   862.5  131.85017
## M4  43.83359   35.70    0.1   242.5   39.05195
## M5 231.10797  160.35    1.3  2503.0  337.54214
## M6  41.63383   25.90    0.2   339.4   48.40078
```

Vamos a hacer un análisis descriptivo de las concenctraciones de metabolitos.

El metabolito con mayor concentración media en las diistintas muestras es el 48 con 9989.2464286, el metabolito cuya mediana de concentración fué mayor es 48 con 7963.95y el que mayor desviación 60 con \ensuremath{1.4293948\times 10^{4}}, el metabolito que tuvo la concentración media menor fué 3 con0.1.

```
hist(Resumen_datos$Media)
```




```
#La mayoría de datos de concentraciones están en el rango de 0 a 1000, se observa que hay algún metabol
max(Resumen_datos$Media)
```

```
## [1] 9989.246
```

```
which.max(Resumen_datos$Media)
```

```
## [1] 48
```

```
max(Resumen_datos$Mediana)
```

```
## [1] 7963.95
```

```
which.max(Resumen_datos$Mediana)
```

```
## [1] 48
```

```
max(Resumen_datos$Desviacion)
```

```
## [1] 14293.95
```

```r
which.max(Resumen_datos$Desviacion)
```

```
## [1] 60
```

```r
min(Resumen_datos$Minimo)
```

```
## [1] 0.1
```

```r
which.min(Resumen_datos$Minimo)
```

```
## [1] 3
```

```r
max(Resumen_datos$Maximo)
```

```
## [1] 160844.7
```

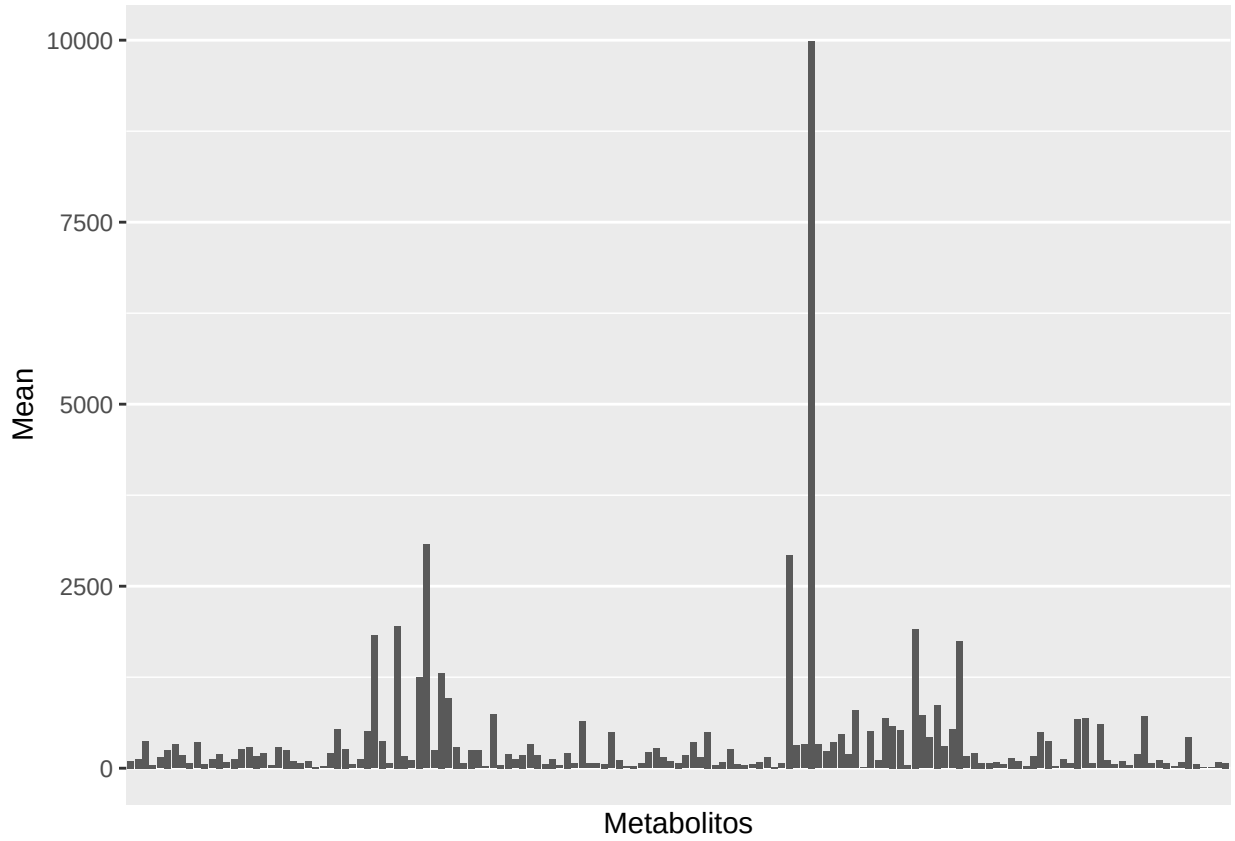
```r
which.max((Resumen_datos$Maximo))
```

```
## [1] 60
```

```r
if (!requireNamespace("ggplot2", quietly = TRUE)) {
  install.packages("ggplot2")
}
library(ggplot2)

# Graficar la media de cada metabolito en formato histograma para hacernos una idea de las concentracio

df<- data.frame((Metabolitos = rownames(Datos)), Mean = Resumen_datos$Media)
ggplot(df, aes(x = Metabolitos, y = Mean)) + geom_bar(stat = "identity") + scale_x_discrete(breaks = 1:
```




```
nrow(Caracteristicas)
```

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

```
nrow(Datos)
```

```
## [1] 149
```

```
nrow(Peak)
```

```
## [1] 149
```

```
head(Caracteristicas)
```

```
## DataFrame with 6 rows and 3 columns
##             SampleID  SampleType    Class
##          <character> <character> <factor>
## sample_1    sample_1          QC       QC
## sample_2    sample_2      Sample       GC
## sample_3    sample_3      Sample       BN
## sample_4    sample_4      Sample       HE
## sample_5    sample_5      Sample       GC
## sample_6    sample_6      Sample       BN
```

```r
# Dividir las muestras en función del factor 'Grupo'
Datos_grupos <- split(seq_len(ncol(Datos)), colData(se)$Class)
# Crear una lista de assays por grupo
Datos_por_grupos <- lapply(Datos_grupos, function(cols) Datos[, cols])
Medias_por_grupos <- lapply(Datos_por_grupos, rowMeans, na.rm = TRUE)
df_Datos_por_grupos<- data.frame(Datos_por_grupos)
```

Comprobar si al separar por grupos de pacientes benignos, enfermos, sanos y QC había cambio en el metabolito más expresado. Encontramos que M48 es el mas expresado en todos los grupos.

```r
max(Medias_por_grupos$BN)
```

```
## [1] 10394.27
```

```r
which.max(Medias_por_grupos$BN)
```

```
## M48
##  48
```

```r
max(Medias_por_grupos$GC)
```

```
## [1] 8838.94
```

```r
which.max(Medias_por_grupos$GC)
```

```
## M48
##  48
```

```r
max(Medias_por_grupos$HE)
```

```
## [1] 11747.39
```

```r
which.max(Medias_por_grupos$HE)
```

```
## M48
##  48
```

```r
max(Medias_por_grupos$QC)
```

```
## [1] 7809.059
```

```r
which.max(Medias_por_grupos$QC)
```

```
## M48
##  48
```

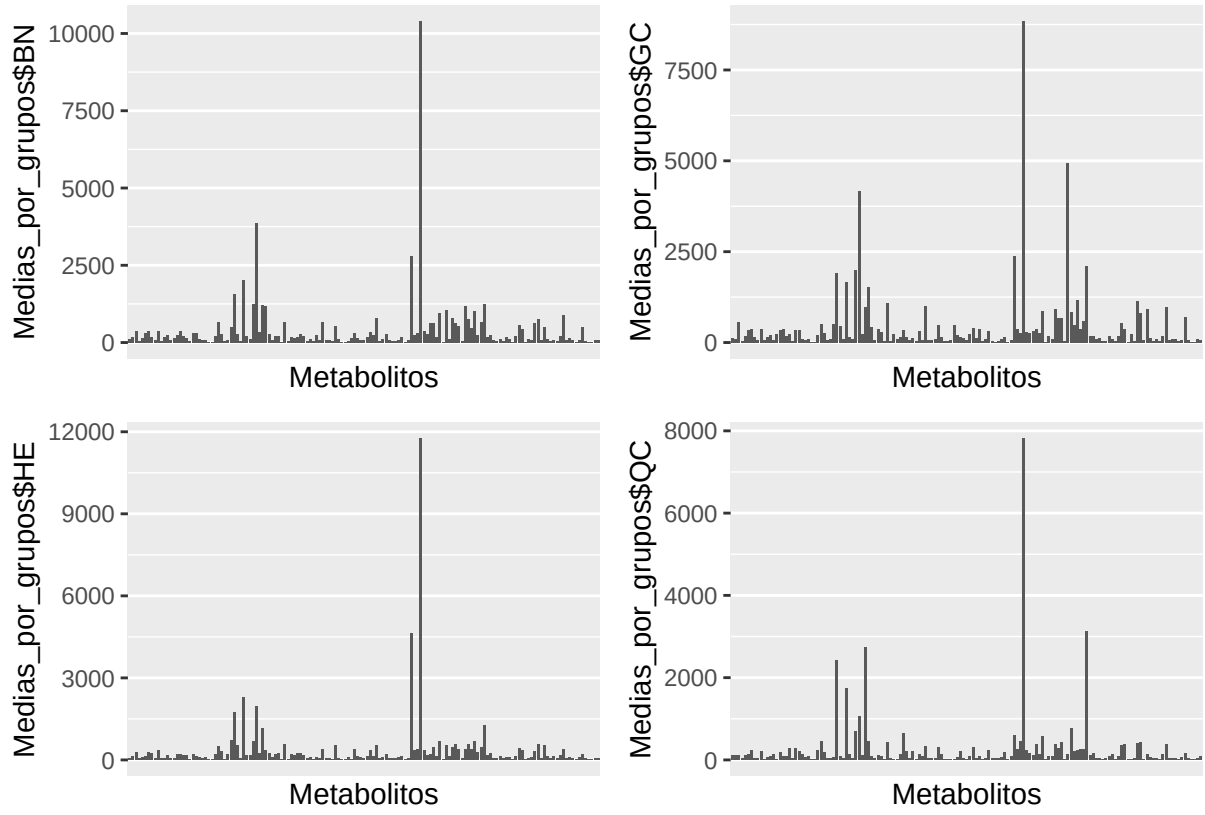
Visualizar la distribución de concentraciones de metabolitos por grupos. Observamos que M48 es el mas abundante en todos y que los demás presentan diferencias de expresión en función del grupo al que pertenecen.

```
install.packages("patchwork")
```

```
## Installing package into 'C:/Users/Usuario/AppData/Local/R/win-library/4.4'
## (as 'lib' is unspecified)
```

```
## package 'patchwork' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\Usuario\AppData\Local\Temp\RtmpKMHZi5\downloaded_packages
```

```
library(patchwork)
df_medias <- data.frame(Medias_por_grupos)
BN_plot<- ggplot(df_medias, aes(x = Metabolitos , y = Medias_por_grupos$BN)) + geom_bar(stat = "identity
GC_plot<- ggplot(df_medias, aes(x = Metabolitos , y = Medias_por_grupos$GC)) + geom_bar(stat = "identity
HE_plot<- ggplot(df_medias, aes(x = Metabolitos , y = Medias_por_grupos$HE)) + geom_bar(stat = "identity
QC_plot<- ggplot(df_medias, aes(x = Metabolitos , y = Medias_por_grupos$QC)) + geom_bar(stat = "identity
combined_plot <- (BN_plot | GC_plot) / (HE_plot | QC_plot)
combined_plot
```



Enlace para acceder al repositorio GitHub

https://github.com/VaroLG/Linacero_Gracia_-lvaro-PEC1.git