Análisis de datos ómicos (M0-157) PEC 1

Elisabet Nebot Colomer

Marzo 2025

https://github.com/ElisabetNebot/Nebot-Colomer-Elisabet-PEC1

1. Ejercicios

Ejercicio 1: Seleccionad y descargad un dataset de metabolómica.

Dentro del repositorio de Github de metaboData me he descargado el dataset de '2024-Cachexia' (Figura 1). Dentro de la carpeta se puede observar los datos en formato .csv y la descripción de los datos. Por otro lado, dentro de Github puedo descargar los metadatos de la descripción del estudio (Figura 2).

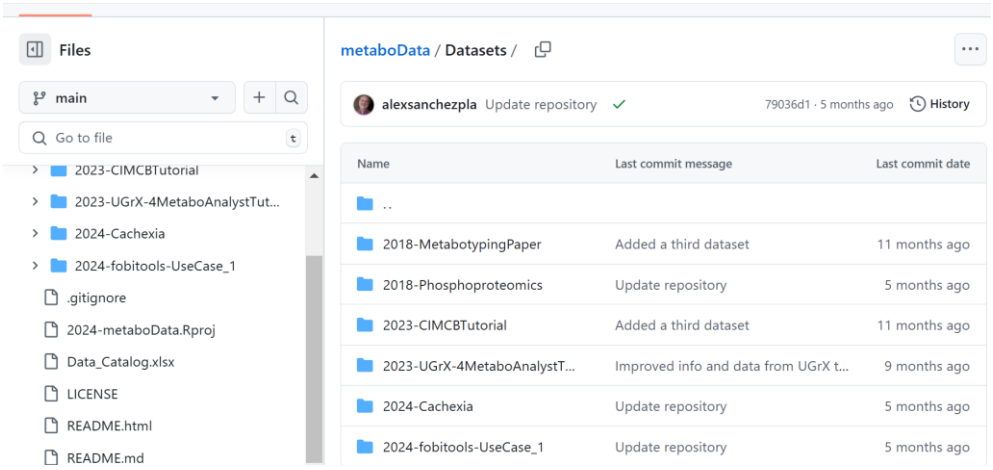

Figura 1. Github de datasets de metabolómica

Dataset	Samples	Features	Description
2018-MetabotypingPaper	39	690	Data used in the paper "Metabotypes of response to bariatric surgery independent of the magnitude of weight loss"
2018-Phosphoproteomics	12	1320	The acompanying dataset has been obtained from a phosphoproteomics experiment that was performed to analyze (3 + 3) PDX models of two different subtypes using Phosphopeptide enriched samples.
2023-CIMCBTutorial	140	149	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)
2023-UGrX-4MetaboAnalystTutorial	24	145	Data from MetabolomicsWorkbench (ID ST000002)
2024-fobitools-UseCase_1	45	1541	This dataset is used in the fobitools Bioconductor package, in one its vignettes, [Use Case ST000291] analyzing the data from Metabolomics Workbench Dataset
2024-Cachexia	77	63	Cachexia is a complex metabolic syndrome associated with an underlying illness (such as cancer) and characterized by loss of muscle with or without loss of fat mass (Evans et al., 2008). A total of 77 urine samples were collected being 47 of them patients with cachexia, and 30 control patients (from the "specmine.datasets" R package)

Figura 2. Metadatos descripción estudio

Una vez descargado el dataset lo cargo en R y hago una exploración del dataset 'cachexia' (Figura 3).

```
library(readr);library(SummarizedExperiment);library(tidyverse); library(xfun)
```

```
human_cachexia <- read_csv("Data/human_cachexia.csv")
```

```
## Rows: 77 Columns: 65
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (2): Patient ID, Muscle loss
## dbl (63): 1,6-Anhydro-beta-D-glucose, 1-Methylnicotinamide, 2-Aminobutyrate,...
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
str(human_cachexia)
```

Figura 3. Carga y estructura dataset 'cachexia'

Explicación y justificación: El dataset es un conjunto de datos de metabolómica, que mide la concentración de 63 metabolitos en 77 pacientes, diferenciados en dos grupos: pacientes con caquexia y controles sin la enfermedad. Cada fila representa un paciente y cada columna corresponde a una variable, incluyendo la identidad del paciente, la presencia o ausencia de pérdida muscular y los niveles de metabolitos en sus muestras biológicas.

La elección de este dataset se basa en el interés por comprender las alteraciones metabólicas asociadas a la caquexia, una condición caracterizada por una pérdida severa de masa muscular que afecta la calidad de vida y el pronóstico de los pacientes.

Preguntas biológicas:

- Los pacientes con caquexia presentan mayores niveles de expresión de metabolitos?
- Hay algún metabolito que se expresa mucho más en los pacientes con caquexia?

Ejercicio 2. Cread un objeto de clase SummarizedExperiment.

Primero de todo creo los parámetros del SummarizedExperiment tales como:

- assays: Contiene la matriz de datos, donde cada fila representa una muestra biológica (paciente) y cada columna representa la concentración de un metabolito.
- ColData: Contiene los metadatos de las columnas, es decir, información sobre los pacientes (si tienen caquexia o no).
- metadata: Contiene información general sobre el estudio y el dataset.

Y luego con la función 'SummarizedExperiment' creo el objeto 'se_object' (Figura 4).

```
# Se seleccionan las dos columnas que contienen la información de los pacientes (metadata)
metadata_cols <- 1:2

# Se crea la matriz de datos de expresión (excluyendo las columnas de metadata)
expression_data <- as.matrix(human_cachexia[, -(metadata_cols)])

# Se asignan los nombres de las filas con los IDs de los pacientes
rownames(expression_data) <- human_cachexia$`Patient ID`

# Se crea los metadatos de las muestras (colData), con los IDs de pacientes como nombres de fila
colData_metadata <- human_cachexia[, metadata_cols] %>%
  column_to_rownames(var = "Patient ID")

# Descripción del dataset (metadata general)
descripcion_estudio <- "Cachexia is a complex metabolic syndrome associated with an underlying illness (such as ca
ncer) and characterized by loss of muscle with or without loss of fat mass (Evans et al., 2008). A total of 77 uri
ne samples were collected, 47 from patients with cachexia and 30 from control patients (from the 'specmine.dataset
s' R package)."

# Creación del objeto SummarizedExperiment
se_object <- SummarizedExperiment(
  assays = list(counts = t(expression_data)),  # Matriz de datos
  colData = colData_metadata, # Metadatos de pacientes
  metadata = list(description = descripcion_estudio))  #  Descripción del estudio

se_object
```

Figura 4. Creación SummarizedExperiment

Una vez creado el objeto SummarizedExperiment este se puede guardar en formato .Rda.

```
## class: SummarizedExperiment
## dim: 63 77
## metadata(1): description
## assays(1): counts
## rownames(63): 1,6-Anhydro-beta-D-glucose 1-Methylnicotinamide ...
##   pi-Methylhistidine tau-Methylhistidine
## rowData names(0):
## colnames(77): PIF_178 PIF_087 ... NETL_003_V1 NETL_003_V2
## colData names(1): Muscle loss
```

```
save(se_object, file = "SummarizedExperiment_data.Rda")
```

Figura 5. Estructura de 'se_object' y guardado.

Ejercicio 3. Análisis exploratorios

Análisis univariantes

Con estos análisis descriptivos observamos que el objeto 'se_object' tiene 63 filas (metabolitos) y 77 columnas (pacientes). Además podemos ver en detalle el nombre de los metabolitos analizados y las estadística descriptiva de los niveles de metabolitos por paciente (Figura 6).

```
dim(se_object) # dimensiones
```

```
## [1] 63 77
```

```
names(se_object) # nombres de metabolitos que han sido medidos en cada muestra
```

```
##  [1] "1,6-Anhydro-beta-D-glucose" "1-Methylnicotinamide"
##  [3] "2-Aminobutyrate"            "2-Hydroxyisobutyrate"
##  [5] "2-Oxoglutarate"             "3-Aminoisobutyrate"
##  [7] "3-Hydroxybutyrate"          "3-Hydroxyisovalerate"
##  [9] "3-Indoxylsulfate"           "4-Hydroxyphenylacetate"
## [11] "Acetate"                    "Acetone"
## [13] "Adipate"                    "Alanine"
```

```
summary(assay(se_object, "counts")) # estadística descriptiva de los niveles de cada metabolito en los diferentes pacientes
```

```
##     PIF_178            PIF_087            PIF_090           NETL_005_V1
##  Min.   :    5.58   Min.   :    7.69   Min.   :    4.44   Min.   :   25.03
##  1st Qu.:   52.72   1st Qu.:   78.66   1st Qu.:   31.50   1st Qu.:  102.51
##  Median :  154.47   Median :  208.51   Median :  141.17   Median :  247.15
##  Mean   :  699.86   Mean   :  708.30   Mean   :  771.79   Mean   : 1021.28
##  3rd Qu.:  416.24   3rd Qu.:  412.10   3rd Qu.:  308.03   3rd Qu.:  673.71
##  Max.   :16481.60   Max.   :15835.35   Max.   :24587.66   Max.   :20952.22
```

Figura 6. Análisis descriptivos

Seguidamente para poder graficar con ggplot convierto la matriz en un 'data.frame' (Figura 7).

```r
# Seguidamente para poder graficar con ggplot convierto la matriz en un 'data.frame'
dataframe <- as.data.frame(t(assay(se_object, "counts"))) %>%
  rownames_to_column(var = "Patient_ID") %>%
  pivot_longer(-Patient_ID, names_to = "Metabolito", values_to = "Expresión")

dataframe <- dataframe  %>%
  left_join(as.data.frame(colData(se_object)) %>%
              rownames_to_column(var = "Patient_ID"), by = "Patient_ID") # se une la columna de 'Muscle.loss'

dim(dataframe) # se visualiza las dimensiones de este dataframe
```

```
## [1] 4851    4
```

```r
names(dataframe) # se visualiza los nombres de las columnas del dataframe
```

```
## [1] "Patient_ID"  "Metabolito"  "Expresión"   "Muscle.loss"
```

Figura 7. Creación y visualización data.frame para trabajar con ggplot

Para observar la frecuencia de expresión de metabolitos por paciente realizo un histograma, en este se observa mayor frecuencia con expresiones en niveles de expresión bajos, sin embargo como cada metabolito presenta una escala diferente, es mejor transformar los datos para poder comparar mejor (Figura 8 y 9).

```r
# Se crea histogramas por paciente
(histo <- ggplot(dataframe, aes(x = Expresión)) +
    geom_histogram(bins = 30, fill = "steelblue", alpha = 0.7) +
    facet_wrap(~ Patient_ID, scales = "free") +
    theme_bw() +
    labs(x = "Nivel de expresión", y = "Frecuencia") +
    theme(
      axis.title.x = element_text(size = 18),
      axis.title.y = element_text(size = 18),
      axis.text.x = element_text(size = 16),
      axis.text.y = element_text(size = 16)))
```

```r
# Se crea histogramas por paciente con transformación logaritmica
(histo_trans <- ggplot(dataframe, aes(x = log10(Expresión + 1))) +
    geom_histogram(bins = 30, fill = "steelblue", alpha = 0.7) +
    facet_wrap(~ Patient_ID, scales = "free") +
    theme_bw() +
    labs(x = "Nivel de expresión (log transformado)", y = "Frecuencia") +
    theme(
      axis.title.x = element_text(size = 18),
      axis.title.y = element_text(size = 18),
      axis.text.x = element_text(size = 16),
      axis.text.y = element_text(size = 16)))
```

Figura 8. Código de histograma con ggplot2

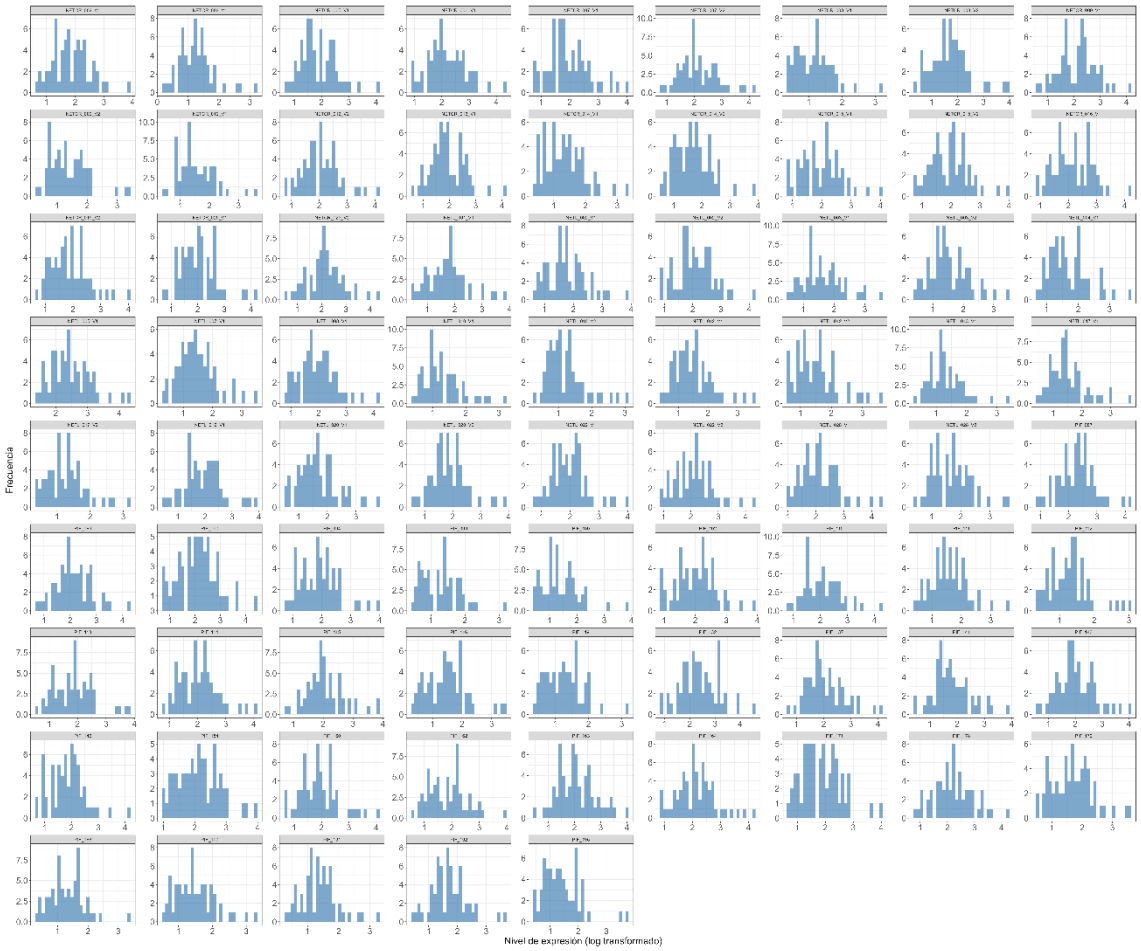

Figura 9. Código y histograma de la frecuencia de expresión por paciente transformado con base logarítmica

Para responder a las preguntas biológicas, se crean varios boxplot a partir del dataframe creado anteriormente. A nivel de paciente, se observa como hay algunos pacientes con caquexia con mayores niveles de expresión que otros (Figura 10), pero en general estos presentan mayores niveles de expresión que los pacientes 'control' (Figura 11).

```
# Se crea un boxplot tranformado para comparar los niveles de expresión por paciente
(boxplot1 <- ggplot(dataframe, aes(x = Patient_ID, y = log10(Expresión + 1), fill = `Muscle.loss`)) +
    geom_boxplot(outlier.shape = NA) +
    scale_fill_manual(values = c("cachexic" = "brown4", "control" = "steelblue")) +
    labs(x = "Paciente", y = "Nivel de expresión", fill = "Muscle.Loss")+
    theme_bw() +
    theme(
      axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1, size = 16),
      axis.title.x = element_text(size = 18),
      axis.title.y = element_text(size = 18),
      axis.text.y = element_text(size = 16),
      legend.text = element_text(size = 16),
      legend.title = element_text(size = 18)))
```

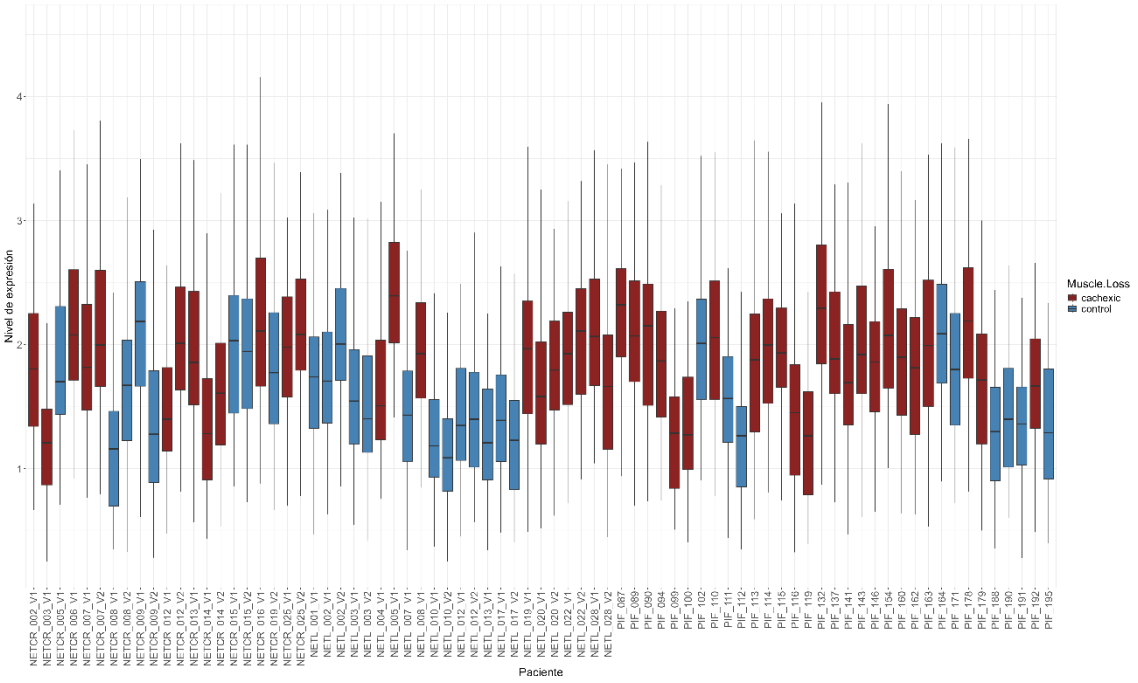

Figura 10. Código y boxplot de los niveles de expresión por pacientes.

```
# Se crea un boxplot tranformado para comparar los niveles de expresión por afección de caquexia o no
(boxplot3 <- ggplot(dataframe, aes(x = `Muscle.loss`, y = log10(Expresión + 1), fill = `Muscle.loss`)) +
    geom_boxplot(outlier.shape = NA) +
    scale_fill_manual(values = c("cachexic" = "brown4", "control" = "steelblue")) +
    labs(x = "Tratamiento", y = "Nivel de expresión (log transformado)", fill = "Muscle.Loss") +
    theme_bw() +
    theme(
      axis.text.x = element_text(angle = 45, vjust = 0.5, hjust = 0.5, size = 16),
      axis.title.x = element_text(size = 18),
      axis.title.y = element_text(size = 18),
      axis.text.y = element_text(size = 16),
      legend.text = element_text(size = 16),
      legend.title = element_text(size = 18)
    ))
```

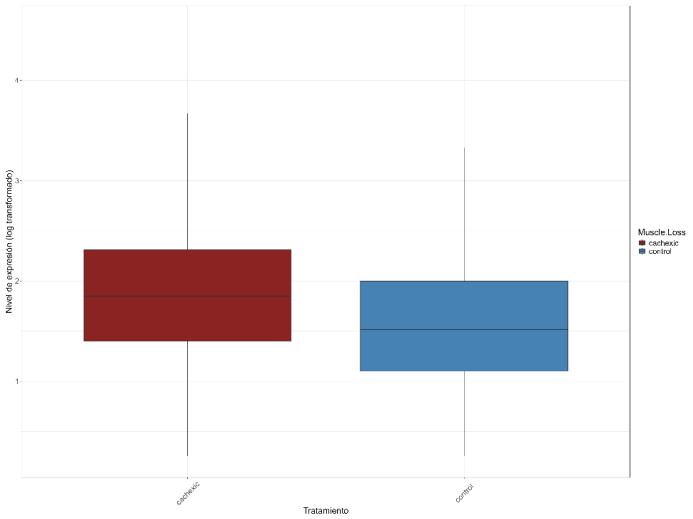

Figura 11. Código y boxplot de los niveles de expresión por tratamiento (‘caquexia’ y ‘control’).

Por otro lado a nivel de metabolito, como se ha visto anteriormente todos los niveles de expresión de los metabolitos de pacientes con caquexia son mayores que en los pacientes control. Aunque se puede observar mayor diferencia entre ambos tratamientos con el metabolito 'Creatine', 'Quinolinate' y 'Adipate'. Sin embargo se debería realizar un análisis estadístico para comprobar si existen diferencias significativas entre ambos niveles.

```
# Se crea un boxplot tranformado para comparar los niveles de expresión por metabolito
(boxplot2 <- ggplot(dataframe, aes(x = Metabolito, y = log10(Expresión + 1), fill = `Muscle.loss`)) +
    geom_boxplot(outlier.shape = NA) +
    scale_fill_manual(values = c("cachexic" = "brown4", "control" = "steelblue")) +
    labs(x = "Paciente", y = "Nivel de expresión", fill = "Muscle.Loss")+
    theme_bw() +
    theme(
      axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1, size = 16),
      axis.title.x = element_text(size = 18),
      axis.title.y = element_text(size = 18),
      axis.text.y = element_text(size = 16),
      legend.text = element_text(size = 16),
      legend.title = element_text(size = 18)))
```

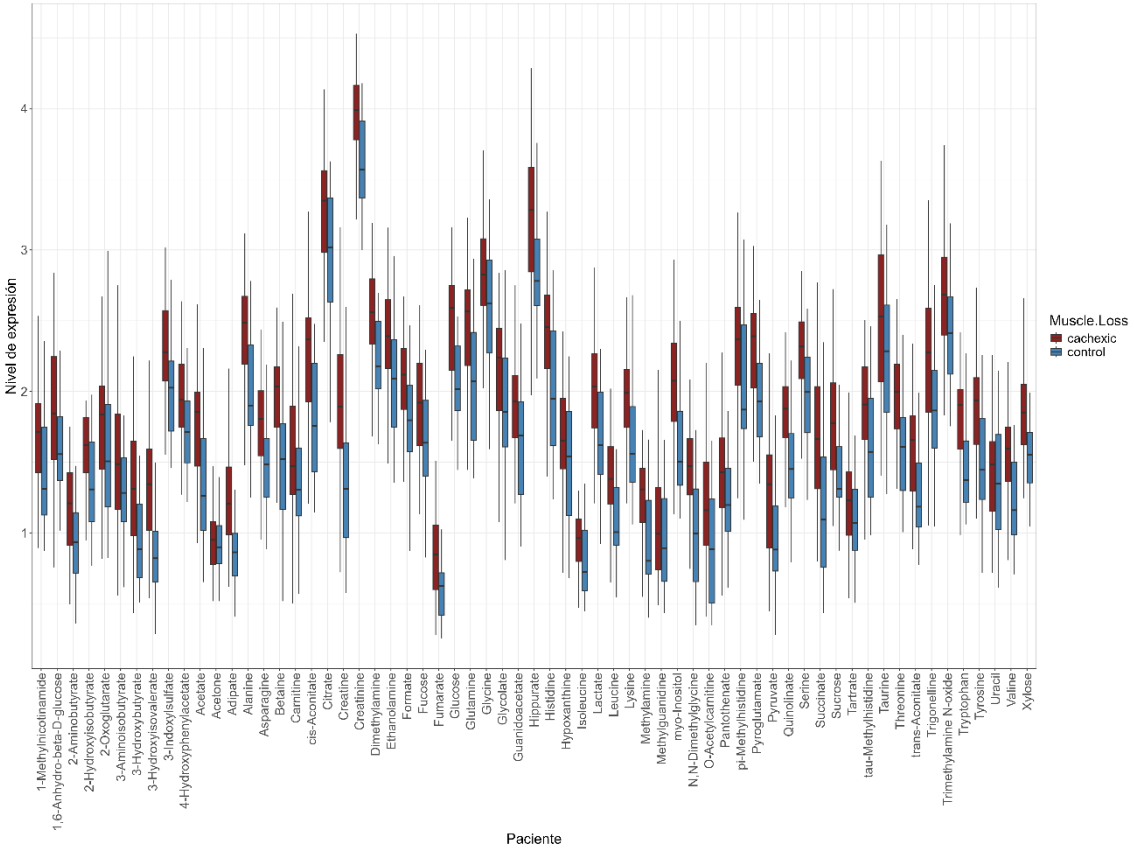

Figura 12. Código y boxplot de los niveles de expresión por metabolito.

Análisis multivariantes

Para visualizar como los pacientes se agrupan en función de la similitud en sus niveles de expresión de los metabolitos realizo un análisis de principales componentes (PCA en inglés). El objetivo es resumir la información que recogida en diversos componentes principales en dos y representarlo en un gráfico en 2D.

```
# Realizar PCA sobre los datos de expresión de metabolitos
log_counts <- log10(t(assay(se_object, "counts")) + 1) #transformo los datos con el logaritmo en base 10.

pca_res_log <- prcomp(log_counts, scale. = TRUE) # se obtiene las coordenadas de cada PC por metabolito

# Extraer las coordenadas de los primeros dos componentes principales (PC1 y PC2)
pca_df <- as.data.frame(pca_res_log$x)  # Esto extrae la x(coordenadas de PC) y convierte las coordenadas en dataf
rame
pca_df$Muscle.loss <- colData(se_object)$`Muscle loss`  # Se añade la columna de metadatos (Muscle.loss)

# Extraer las coordenadas de los metabolitos
pca_var <- as.data.frame(pca_res_log$rotation[, 1:2]) #  extrae las columnas del PC1 y PC2 y lo convierte en dataf
rame
pca_var$Metabolito <- rownames(pca_var)  # Añade una columna con los nombres de metabolitos


## Determinar el porcentaje de variación explicado por cada PC.
std_devs <- pca_res_log$sdev # extrae los desviación estándar del objeto pca_res_log
eigenvalues <- std_devs^2 # cálculo para obtener la varianza.
variance_explained <- eigenvalues / sum(eigenvalues)
cumulative_variance <- cumsum(variance_explained) # porcentaje de variación explicada por los PC.
```

```
(pca_plot_v1 <- ggplot(pca_df, aes(x = PC1, y = PC2, colour = Muscle.loss)) +
    geom_point(size = 3) +
    scale_colour_manual(values = c("cachexic" = "brown4", "control" = "steelblue")) +
    labs(x = "PC1 (58.53%)",y = "PC2 (4.45%)") +
    theme_bw() +
    theme(text = element_text(size = 16),
          legend.title = element_text(size = 18),
          legend.text = element_text(size = 16)))
```

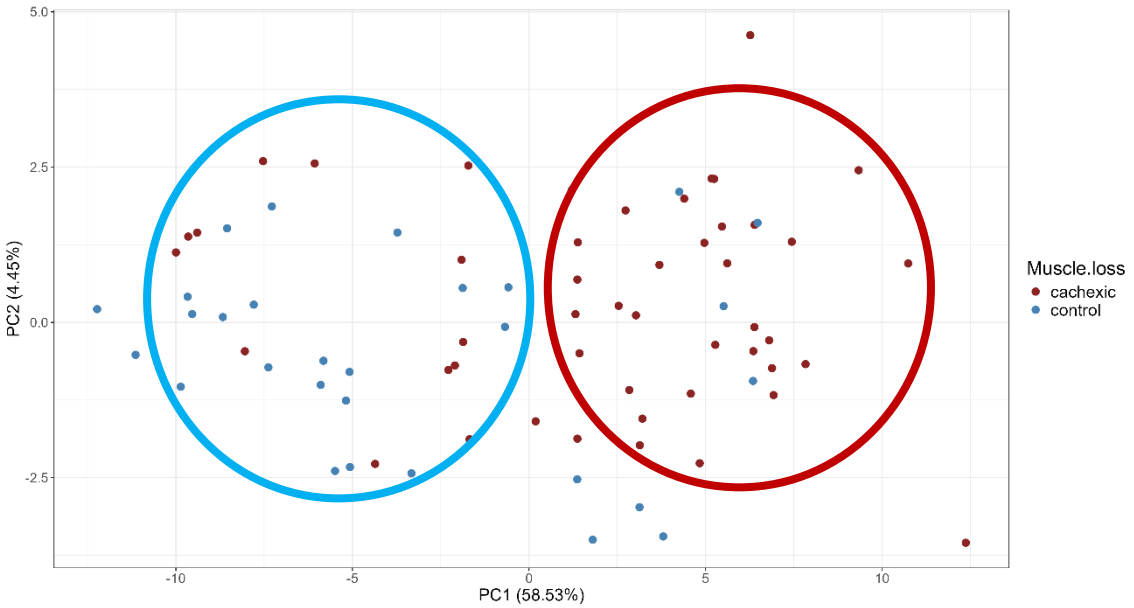

Figura 13. Código para realizar el PCA y representación gráfica de los dos PC.

El PC1 explica el 58.53% de la variación observada en datos, se observa una clara separación de los pacientes, los pacientes enfermos se agrupan más a la derecha y los pacientes control más a la izquierda (Figura 13). En concreto, los metabolitos influyen en la separación de las muestras (pacientes) hacia la derecha (Figura 14). Se puede observar como hay ciertos metabolitos con flechas más largas (ej. Acetate, Oxo-glutarate) lo que indica que tiene una mayor contribución a la varianza explicada por los PC.

```
# PCA añadiendo las coordenadas de los metabolitos
(pca_plot_v2 <- ggplot() +
    geom_point(data = pca_df, aes(x = PC1, y = PC2, colour = Muscle.loss), size = 3) +
    scale_colour_manual(values = c("cachexic" = "brown4", "control" = "steelblue")) +
    geom_segment(data = pca_var, aes(x = 0, y = 0, xend = PC1 * 5, yend = PC2 * 5),      # Flechas de los metabolit
os
                 arrow = arrow(length = unit(0.2, "cm")), color = "black", alpha = 0.7) +
    geom_text_repel(data = pca_var, aes(x = PC1 * 5, y = PC2 * 5, label = Metabolito),
                    size = 5, color = "black", max.overlaps = 20) +     # Etiquetas de los metabolitos
    labs(x = "PC1 (58.53%)",
         y = "PC2 (4.45%)") +
    theme_bw() +
    theme(text = element_text(size = 16),
          legend.title = element_text(size = 18),
          legend.text = element_text(size = 16)))
```

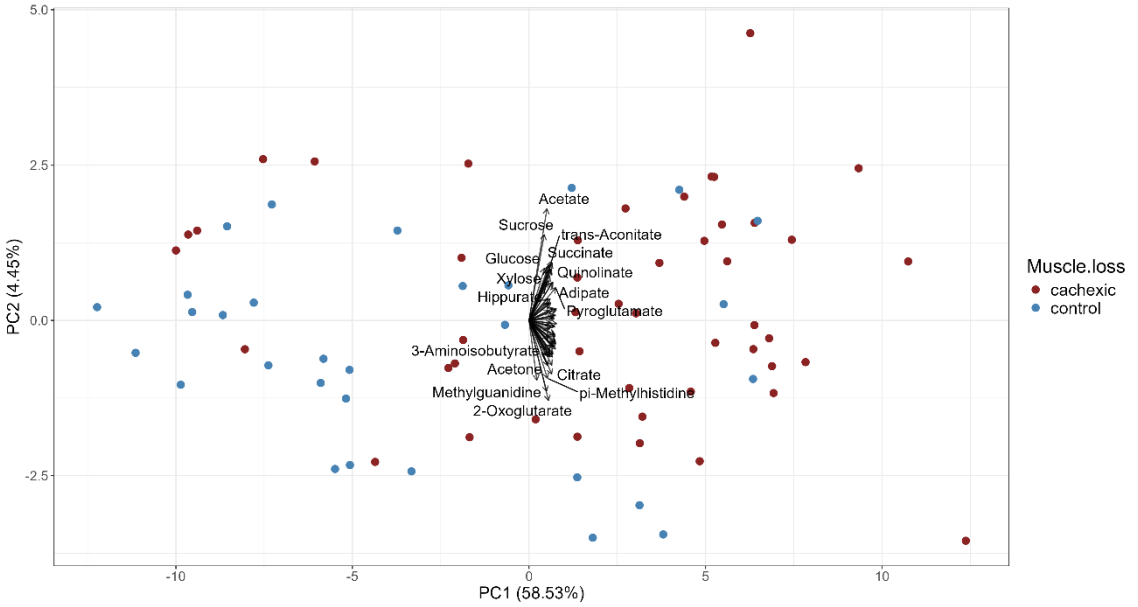

Figura 14. Código para realizar el PCA y representación gráfica de los dos PC y los metabolitos.

Respuesta preguntas biológicas:

Respecto la interpretación biológica de los datos, vemos claramente que los pacientes con caquexia muestran niveles más elevados de expresión de metabolitos. Sin embargo, no se observa un patrón claro que metabolito determina la caquexia.