Análisis de datos ómicos: PEC1

Berta Laudo

2024-11-06

Contents

Abstract

La caquexia es un síndrome caracterizado por la pérdida de peso, masa muscular y grasa corporal, comúnmente asociado a enfermedades crónicas como el cáncer, insuficiencia cardíaca, EPOC y enfermedades renales. Este trastorno no solo implica una disminución de la masa corporal, sino que afecta el equilibrio metabólico, la función inmune y la calidad de vida. La alteración metabólica en la caquexia refleja un desajuste entre el gasto y la producción de energía, lo cual se traduce en cambios en los perfiles de metabolitos. En este estudio, se analiza la concentración de un grupo de metabolitos entre un grupo de pacientes y un grupo control, para ver cómo afecta esta enfermedad a los merfiles metabólicos relacionados. El resultado de este análisis abre la puerta al desarrollo de nuevas estrategias terapéuticas y la identificación de biomarcadores diagnósticos y pronósticos, esenciales para mejorar el tratamiento y seguimiento de los pacientes.

Objetivos del estudio

El objetivo principal de este estudio es estudiar las diferencias en la concentración de metabolitos entre los pacientes que presentan caquexia en comparación con los controles.

Materiales

Los datos utilizados en este estudio se han obtenido del repositorio de github en formato de tabla separada por comas. Todo el proceso de anásis, incluído la descarga de datos, se ha realizado en RStudio.

Métodos y Resultados

Preparación de los datos

Descarga y carga de la base de datos

Observamos que el archivo se trata de un *dataframe* con 65 columnas y 77 observaciones. Las dos primeras columnas corresponden a la metadata, es decir, la información sobre los datos. En este caso, la primera columna contiene un código identificativo de los pacientes y la segunda la condición del paciente (caquexia o control). Las demás 63 columnas corresponden a diferentes metabolitos y cada fila registra la concentración del metabolito en el paciente.

Preparación de datos

Para poder trabajar mejor con los datos, extraeremos las columnas de metadata en una nueva variable para poder trabajar independientemente con las concentraciones y con la metadata.

```
# Extraemos las dos primera columnas en una nueva variable
metadata <- as.data.frame(caxdb[,2])

# Asignamos los ID como rownames
rownames(metadata) <- caxdb[,1]
names(metadata)[1] <- "Muscle.loss"
metadata$Muscle.loss <- factor(metadata$Muscle.loss, levels = c("control", "cachexic"))

# Eliminamos las dos primeras columnas de las concentraciones
cax.conc <- as.matrix(caxdb[, 3:65])
# Asignamos los ID como rownames
rownames(cax.conc) <- caxdb[,1]
# Trasponemos la matriz para generar el SE
cax.conc <- t(cax.conc)
```

Creación del contenedor SummarizedExperiment

```
library(SummarizedExperiment)

# Creamos el objeto de clase SummarizedExperiment
cax.se <- SummarizedExperiment(
  assays = list(metabolites = cax.conc),
  colData = metadata)

# Inspeccionamos el objecto SummarizedExperiment creado
cax.se
```

```
class: SummarizedExperiment
dim: 63 77
metadata(0):
assays(1): metabolites
rownames(63): X1.6.Anhydro.beta.D.glucose X1.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
```

Podemos decir pues, que los datos están formados por 77 muestras, 53 metabolitos, 1 covariable y 2 grupos experimentales (Controles y Caquexia).

Pre-procesado de los datos

Antes del análisis estadístico, es importante pre-procesar los datos.

Eliminación de los *Missing Values*

Debido a errores biólogicos o tecnológicos, algunos valores pueden no ser cuantificados correctamente en algunas muestras y aparecen como *missing values* o valores NA, y es importante controlarlos para el posterior análisis estadístico.

Aunque en el repositorio de github de donde descargamos los datos nos detalla que no hay valores NA en el archivo, lo confirmaremos con la siguiente función.

```
anyNA(assay(cax.se, "metabolites"))
```

```
[1] FALSE
```

Por lo tanto, no hay ningún valor NA en nuestros datos.

Normalización de los datos

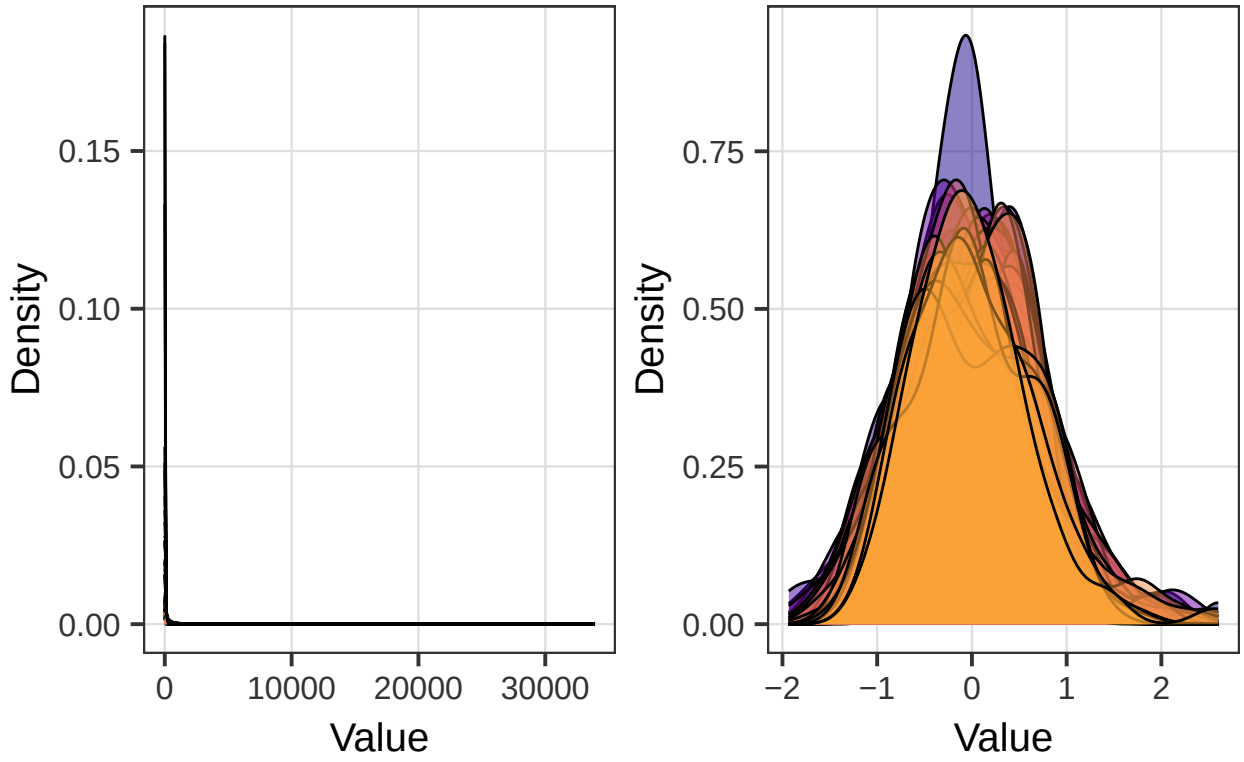
La normalización de los datos es un punto clave en el análisis de ciertos tipos de datos, como los de metabolómica, en los que la variabilidad de los resultados puede tener una influencia crítica en los cálculos estadísticos.

Para controlar esta variablidad en los valores de concentración de metabolitos, es común normalizar los valores usando una transformación logarítmica, que podemos aplicar con el paquete POMA.

```
library(POMA)

# Aplicamos la transformación logarítmica
log.cax.se <- PomaNorm(cax.se, method = "log_pareto")
assayNames(log.cax.se) <- "metabolites"
```

Podemos comparar la distribución de nuestros datos antes y después de la normalización con un gráfico de densidad, que podemos generar con la función PomaDensity() del paquete POMA.

```
dens.notnorm <- PomaDensity(cax.se,
                            x = "features") +
  ggtitle("Sin Normalizar")  +
  theme(legend.position = "none")

dens.norm <- PomaDensity(log.cax.se,
             x = "features") +
  ggtitle("Normalizados")  +
  theme(legend.position = "none")

grid.arrange(dens.notnorm, dens.norm, nrow = 1)
```



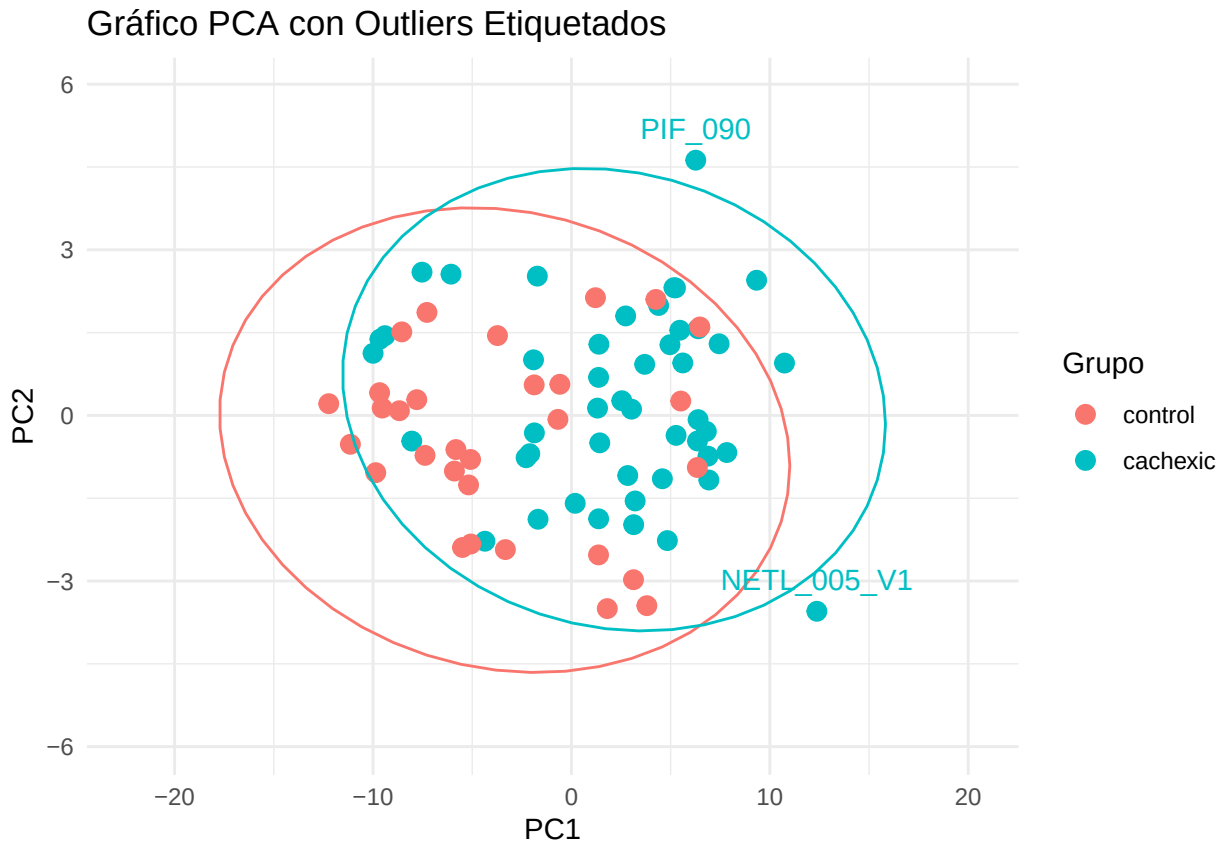
Vemos que después de normalizar los datos, estos se distribuyen alrededor de un valor central, en este caso 0, a diferencia de los datos sin normalizar.

Detección de outliers

Para la detección de outliers, realizaremos un gráfico de PCA combinado con un cálculo de la distancia de las distancias de Mahalanobis, ya que con un análisis de componentes principales permite visualizar la distribución de las muestras y nos permite detectar muestras que se alejen del resto, indicando posibles outliers. Además, la distancia de Mahalanobis calcula cómo de lejos está una muestra del centro de la distribución, por lo que muestras que sobrepasen un cierto umbral de esta distancia, nos confirmarían que son outliers.

```
# Extraemos la matriz de metabolitos
metabolites_data <- assay(log.cax.se)
# Realizamos el PCA
pca_result <- prcomp(t(metabolites_data), center = TRUE, scale. = TRUE)
pca_data <- as.data.frame(pca_result$x)
# Añadir los grupos de la metadata
pca_data$Grupo <- colData(log.cax.se)$Muscle.loss
# Calculamos las distancias de Mahalanobis para identificar outliers
pc_scores <- pca_data[, c("PC1", "PC2")]
center <- colMeans(pc_scores)
cov_matrix <- cov(pc_scores)
mahalanobis_dist <- mahalanobis(pc_scores, center, cov_matrix)
# Definimos el umbral para los outliers (95% de confianza)
threshold <- qchisq(0.95, df = 2)
pca_data$outlier <- mahalanobis_dist > threshold
```

```r
# Añadimos los nombres de los outliers
pca_data$label <- ifelse(pca_data$outlier, rownames(pca_data), NA)

# Generamos el gráfico
ggplot(pca_data, aes(x = PC1, y = PC2, color = Grupo)) +
  geom_point(size = 3) +
  stat_ellipse(type = "norm", level = 0.95, show.legend = FALSE) +
  geom_text(aes(label = label), vjust = -1, hjust = 0.5, na.rm = TRUE, show.legend = FALSE) +
  labs(title = "Gráfico PCA con Outliers Etiquetados") +
    scale_x_continuous(expand = expansion(mult = 0.2)) +
  scale_y_continuous(expand = expansion(mult = 0.2)) +
  theme_minimal()
```



A partir del gráfico, podemos confirmar que las muestras *PIF_090* y *NETL_005_V1* son outliers, ya que se alejan del resto de las muestra de manera significativa con un intervalo de confianza del 95%.

Por lo tanto, para el análisis estadístico eliminaremos estas dos muestras.

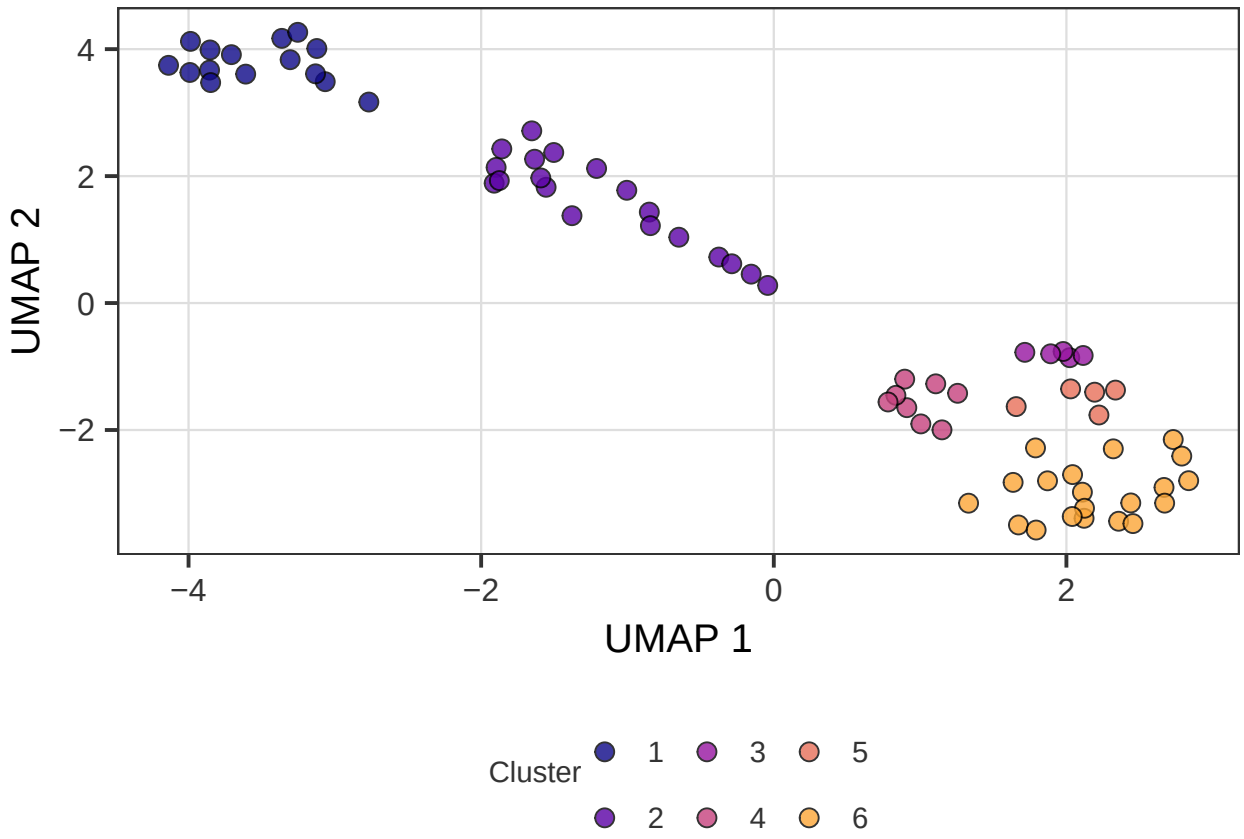
```r
outliers <- c("PIF_090", "NETL_005_V1")
# Filtramos los datos
cax.se <- cax.se[, !colnames(cax.se) %in% outliers]
log.cax.se <- log.cax.se[, !colnames(log.cax.se) %in% outliers]
```

Análisis Estadístico Multivariante

Gráfico UMAP

Para complementar el PCA, realizaremos un gráfico UMAP (Uniform Manifold Approximation and Projection), que reduce la dimensionalidad de nuestros datos y nos ayuda a detectar patrones. Para ello, volveremos a utilizar el paquete POMA.

```
log.cax.se_umap <- PomaUMAP(log.cax.se)
log.cax.se_umap$umap_plot
```



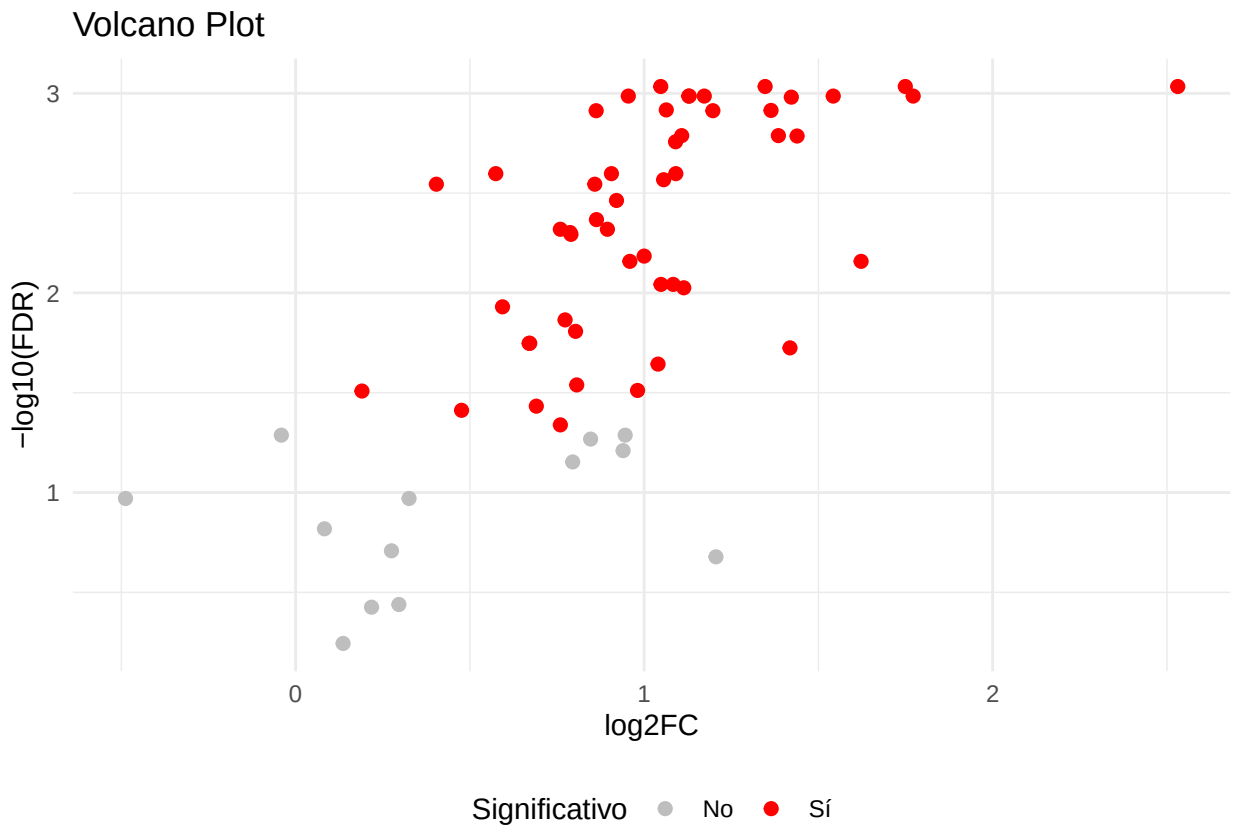
A partir del gráfico podemos observar como los datos se separan en grupos distintos.

Diferencias de metabolitos entre grupos

Para poder estudiar en profundidad las diferencias en las concentraciones de metabolitos entre los 2 grupos de pacientes podemos realizar un test estadístico de Wilcoxon, con el paquete POMA. A partir de los resultados comparativos, podemos visualizarlos generando un volcano plot para ver los metabolitos que se expresan significativamente más en un grupo o en otro. Usaremos los datos sin normalizar para evitar valores de FC negativos.

```
# cachexic vs control
cax.se_mann <- as.data.frame(PomaUnivariate(cax.se,
                                            method = "mann")$result)
cax.se_mann$log2FC <- log2(cax.se_mann$fold_change)
# Añadimos una columna para etiquetar los metabolitos significativos
cax.se_mann$Significativo <- ifelse(cax.se_mann$adj_pvalue < 0.05, "Sí", "No")

# Creamos el volcano plot
```

```r
ggplot(cax.se_mann, aes(x = log2FC, y = -log10(adj_pvalue), color = Significativo)) +
  geom_point(size = 2) +
  scale_color_manual(values = c("No" = "gray", "Sí" = "red")) +
  theme_minimal() +
  labs(title = "Volcano Plot", x = "log2FC", y = "-log10(FDR)") +
  theme(legend.position = "bottom")
```



En el volcano plot podemos ver todos los metabolitos, en rojo los significativos (FDR < 0.05) y además, como la comparación que hemos hecho es cachexic vs control y los puntos significativos tienen valores de log2FC positivo, significa que todos los puntos rojos corresponden a metabolitos con una concentración significativamente mayor en el grupo cachexic.

Discusión y limitaciones y conclusiones del estudio

En este estudio hemos podido analizar las diferencias en las concentraciones de metabolitos entre un grupo de pacientes con caquexia y un grupo control, además de analizar la distribución y agrupación de los datos. Hemos visto que hay ciertos metabolitos con una concentración significativamente más alta en los pacientes que en los controles, por lo que sería muy interesante estudiar en profundidad estos metabolitos y ver cómo se relacionan con la enfermedad y si se podrían considerar como posibles biomarcadores.

Una de las limitaciones de los datos ha sido tener que generar la tabla de metadata a partir de la tabla descargada, ya que era crucial que la tabla con la información de las muestras tuviera un formato adecuado para trabajar a posteriori con el contenedor SummarizedExperiment. Otra limitación importante ha sido la utilización del paquete POMA, que aunque facilita muchas funciones de análisis, la documentación del paquete y sus funciones no está actualizada en muchas fuentes y no parece que haya mucha comunidad científica utilizándolo, por lo que costaba resolver los problemas y errores que pudieran surgir.

En resumen, este estudio ha sido un muy buen ejercicio para trabajar con los contenedores de tipo SummarizedExperiment y poder aplicarles distintas funciones para realizar un análisis de los datos.

Repositorio github de la entrega