PEC1 - Análisis de datos ómicos

Aitana Vázquez Fernández

2025-03-20

Table of Contents

Abstract

La sarcopenia, una condición caracterizada por la pérdida progresiva de masa y función muscular, ha sido relacionada con la microbiota intestinal y su perfil metabolómico. En este trabajo se emplean técnicas de análisis metabolómico para explorar dicha relación utilizando datos obtenidos del repositorio Metabolomics Workbench. Los datos se organizaron en un objeto SummarizedExperiment y fueron analizados mediante el POMA Workflow y análisis estadísticos exploratorios. Los resultados principales sugieren que existen metabolitos significativamente asociados con la sarcopenia, aunque la variabilidad explicada por los componentes principales es relativamente baja. Además, el análisis de correlación muestra altas correlaciones entre muchos de los metabolitos analizados, y el análisis de agrupamiento jerárquico muestra cierta separación entre grupos, aunque no completamente definida. Estos hallazgos resaltan que la relación entre la microbiota y la sarcopenia empleando metabolitos para su estudio es compleja y es necesaria mayor investigación para comprender los mecanismos que intervienen en esta relación.

Objetivos

El objetivo general de este trabajo es realizar las tareas propuestas en la PEC1 y generar un informe con los resultados, utilizando para ello los conocimientos adquiridos a lo largo de este reto sobre tecnologías ómicas, el uso de de la herramienta de control de versiones Git y los repositorios en GitHub, el paquete de Bioconductor y sus clases, y por último herramientas estadísticas para la exploración de los datos. Como objetivos específicos se plantean los siguientes:

-Crear un repositorio de GitHub para llevar a cabo el control de versiones del código de R y volcar los archivos resultados del trabajo.

-Seleccionar un dataset con el que trabajar y presentarlo en formato clase SummarizedExperiment

-Llevar a cabo un análisis exploratorio de los datos del dataset empleando para ello las distintas técnicas vistas. Para ello se utilizará tanto el POMA Workflow como los métodos analíticos vistos en el reto.

Métodos

En primer lugar, se ha creado un repositorio en GitHub (1) que contendrá todos los elementos asociados a este trabajo. El código R para la exploración de los datos se encuentra debidamente comentado, y se realiza el control de versiones del mismo utilizando Git.

El dataset de metabolómica seleccionado para llevar a cabo este trabajo proviene del repositorio Metabolomics Workbench (2). El dataset pertenece al estudio "The role of gut microbiota in muscle mitochondria function, colon health, and sarcopenia: from clinical to bench (2)" (3). Brevemente este estudio pretende investigar en humanos (Homo Sapiens) cómo la microbiota se relaciona con la sarcopenia, puesto que podrían encontrarse potencialmente asociadas. Para investigar el papel de la microbiota en la sarcopenia se lleva a cabo la comparación de la microbiota intestinal y la composición de metabolitos entre individuos mayores con y sin sarcopenia. La razón por la que se ha elegido este dataset para el desarrollo del trabajo se debe a la relevancia que ha cobrado el estudio de la microbiota y sus metabolitos en los últimos años, puesto que parece que podría relacionarse con multitud de enfermedades o procesos patológicos, algunos de ellos asociados al envejecimiento como por ejemplo, aunque no únicamente, la sarcopenia.

Para poder trabajar con el dataset seleccionado este debe importarse a R, lo cual puede realizarse utilizando el paquete MetabolomicsWorkbenchR (4). Con este paquete, incluido dentro de Bioconductor, es posible acceder directamente a los dataset contenidos en el repositorio. Adicionalmente, es posible importar estos dataset como un objeto de clase SummarizedExperiment directamente en R, que contenga los datos y metadatos del dataset. Para obtener otra información como los objetivos del estudio, el plan de trabajo o los análisis y resultados obtenidos, puede consultarse más información en el repositorio Metabolomics Workbench (3).

La clase SummarizedExperiment es ampliamente utilizada en análisis ómicos (5,6). Una vez se dispone del dataset como objeto de la clase SummarizedExperiment puede comenzarse con la preparación de los datos para su posterior análisis.

Para llevar a cabo el análisis exploratorio de los datos se utilizará inicialmente el POMA Workflow (7, 8) ya que resulta útil para explorar datos contenidos en una clase Summarized Experiment. El POMA Workflow puede dividirse en tres pasos que consisten en la preparación de los datos, pre-procesamiento de los datos y por último, análisis estadístico. La preparación de los datos consiste en almacenar estos en un objeto de tipo SummarizedExperiment. En el pre-procesado de los datos comprende la imputación de valores missing, la normalización de los datos y la detección de outliers. Una vez se haya realizado el pre-procesado de los datos se continuará realizando el análisis estadístico univariado de los datos, con la prueba t. Posteriormente, se realizará análisis multivariado para continuar con la exploración de los datos, empleando análisis de componentes principales (PCA) y calcular la correlación entre variables.

Ya sin utilizar el POMA Workflow, se continuará el análisis exploratorio de los datos siguiendo las indicaciones proporcionadas en el reto (9, 10). Para poder realizar estos análisis debe almacenarse la matriz de datos de la clase Summarized Experiment como matriz datos de R para poder trabajar con ella. Por la forma en la que se encuentran dispuestos los datos en la matriz, es necesario trasponerla previamente para tener las muestras en las filas de la matriz en lugar de en las columnas. Se seguirá el paso a paso necesario para realizar análisis de componentes principales (PCA) de los datos (aunque ya se ha realizado el PCA mediante el paquete POMA), lo que permitirá utilizar otro enfoque y también comparar resultados y ver si ambos enfoques son equivalentes.

Otra posible opción sería llevar a cabo un análisis para detectar si se ha dado el efecto batch en nuestros datos. Este efecto puede afectar a los datos cuando muestras que se han producido en un mismo lote se parecen más entre ellas que las producidas en otros lotes. Puede controlarse con un diseño experimental adecuado, pero también puede controlarse en los análisis (10). Sin embargo, en el caso de este dataset en particular no es posible analizar la presencia de este efecto puesto que no se dispone información sobre el lote del que proceden las muestras.

Por último, se realizará un cluster analysis o análisis de conglomerados, que permite agrupar las muestras en distintos grupos. En este caso se realizará una agrupación jerárquica de las muestras del dataset, para lo cual la estructura utilizada será el dendrograma. Se realizará la agrupación jerárquica basada en las distancias euclidianas y "average linkage", y también basadas en la correlación de Pearson tanto para las muestras del dataset como los metabolitos.

Resultados

Se importa el dataset a R.

```
# Importar el dataset desde Metabolomics Workbench
se = do_query(
  context = 'study',
  input_item = 'study_id',
  input_value = 'ST003002',
  output_item = 'SummarizedExperiment'
)
```

Este SummarizedExperiment contiene dos assays (o matrices de datos experimentales). Para este trabajo se ha seleccionado únicamente una de ellas, AN004930.

El código completo y output de los comandos se muestra en el anexo, aquí solo se presentarán los resultados más relevantes (debido a su extensión).

Se explora la información contenida en el SummarizedExperiment:

```
# Comprobar el contenido de la matriz de expresión
head(assay(se[["AN004930"]]))
#Dimensiones de la matriz
dim(assay(se[["AN004930"]]))
# Comprobar los metadatos de las muestras
colData(se[["AN004930"]])
metadata(se[["AN004930"]])
```

Para acceder a la información general del estudio y procedencia de este dataset, puede hacerse uso de la función genérica metadata. Este objeto almacena la información sobre el diseño del estudio u otros detalles relevantes para caracterizarlo.

Este dataset se encuentra almacenado en la clase SummarizedExperiment, que es una estructura de datos ampliamente utilizada en las ómicas. La matriz de expresión (assay) tiene 974 filas (metabolitos) y 51 columnas (muestras). Los metadatos de las muestras, contenidos en el objeto colData incluyen información como el identificador de la muestra, del estudio, el origen de las muestras y el status del individuo, es decir, la presencia o ausencia de sarcopenia. Los metadatos de los metabolitos, en el objeto rowData contienen información sobre el nombre de los metabolitos, su identificador, o su nombre de referencia. Por último, los metadatos del estudio nos proporcionan información sobre la procedencia del dataset.

Las principales diferencias entre la clase SummarizedExperiment y la clase ExpressionSet son que la primera utiliza assays para contener los datos, mientras que ExpressionSet los almacena en una única matriz y se utiliza exprs() para acceder a ellos. Además, en la clase SummarizedExperimentla información de los genes, metabolitos u otras moléculas, se almacena en elementos rowData, y los datos de las muestras en colData. En cambio, para ExpressionSet se almacenan en featureData y phenoData, respectivamente. Los metadatos generales se almacenan en un elemento metadata para SummarizedExperiment y en experimentData para ExpressionSet. Además de las diferencias en los elementos que contienen los distintos tipos de datos, SummarizedExperiment ofrece una mayor flexibilidad puesto que puede manejar múltiples matrices de datos dentro de assays, mientras que ExpressionSet solo puede manejar una matriz principal. La clase SummarizedExperiment es una estructura más moderna porque se ha introducido más recientemente y permite el uso de múltiples formatos de datos, por otro lado, ExpressionSet es una estructura más antigua del paquete Bioconductor.

El POMA Workflow se divide en tres pasos que consisten en la preparación de los datos, pre-procesamiento de los datos y por último, análisis estadístico. La preparación de los datos consiste en almacenar estos en un objeto de tipo SummarizedExperiment, algo que ya se ha realizado en los datos empleados para este trabajo.

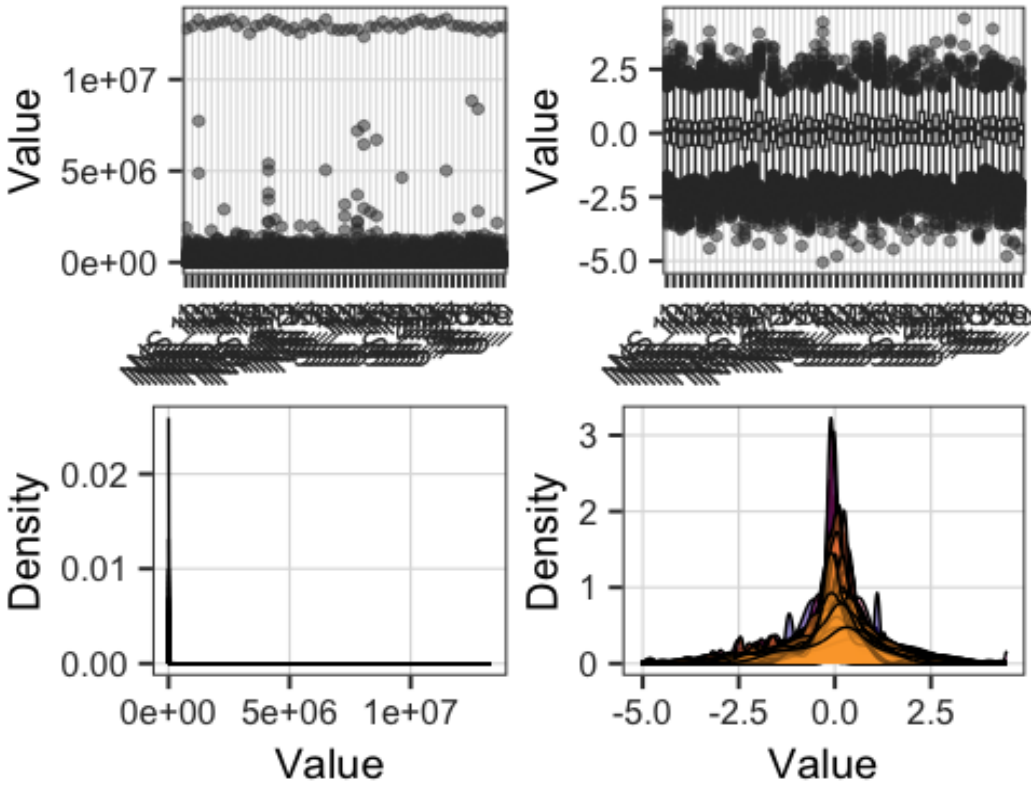
En el pre-procesado de los datos comprende la imputación de valores missing, la normalización de los datos y la detección de outliers. La presencia de valores missing en un dataset puede deberse a distintas razones tanto biológicas como técnicas. El paquete POMA ofrece distintos métodos de imputación que pueden llevarse a cabo para tratar estos valores faltantes. Por tanto, el primer paso del pre-procesado de los datos será tratar los missing, si existen.

```
## No missing values detected

## 0 features removed.
```

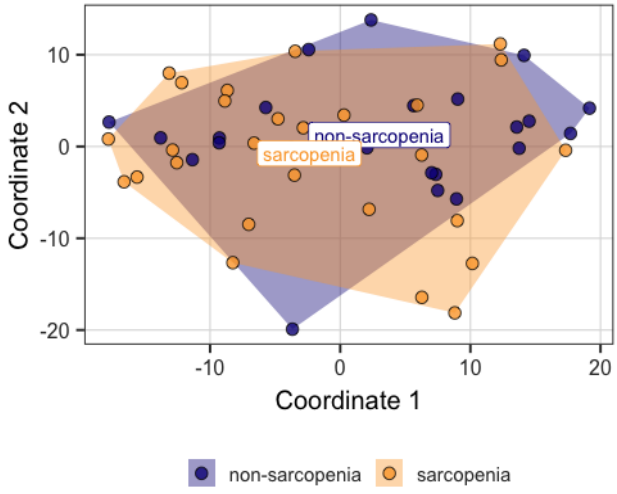
En la imputación de valores de valores missing (NA) en este dataset se observa que no existe ninguno, por lo tanto no se ha eliminado ninguna característica (feature) del mismo.

El siguiente paso consiste en la normalización de los datos.Esto es debido a que algunos factores pueden introducir variabilidad en algunos datos metabolómicos teniendo una gran influencia en el resultado final de los análisis estadísticos que se lleven a cabo. Ya que en el paso anterior no se ha detectado (ni imputado) ningún valor missing, puede llevarse a cabo la normalización de los datos sobre el dataset completo.

Se puede comprobar cuál ha sido el efecto de la normalización de los datos, llevando a cabo una comparación gráfico de los mismos antes y después de su normalización.



El último paso del pre-procesado de los datos consiste en la detección de outliers. Los outliers son valores que destacan por su valor muy distinto al de la mayoría de los demás valores restantes. Estos outliers pueden tener gran influencia en los resultados de los análisis que se lleven a cabo posteriormente. Conocer si existen valores outliers en los datos y decidir como tratarlos (incluirlos en los análisis, eliminarlos…) es un aspecto clave del pre-procesado de los datos. Respecto a la representación de los datos, se ha llevado a cabo la representación de los mismos en función de su 'Status' es decir si son muestras de individuos con sarcopenia o sin sarcopenia.

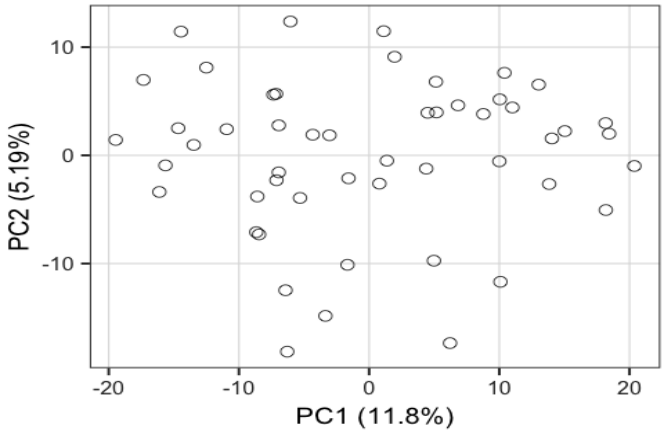


Se comprueba que no existe ningún valor que se haya considerado como outlier por el paquete POMA, y por tanto ningún dato ha sido eliminado.

Con el pre-procesamiento de los datos finalizado, se comenienza con el análisis exploratorio de los mismos, realizando análisis univariado, siguiendo el POMA Workflow.

Tras aplicar la prueba t (ttest), para comparar los niveles de los metabolitos entre los individuos con y sin sarcopenia, se encontrar diferencias estadísticamente significativas en múltiples metabolitos, lo que indica que efectivamente algunos de estos se asocian de forma diferente entre los individuos con y sin la patología. Aunque la prueba t requiera el cumplimiento de varias condiciones para poder aplicarse, los datos se han normalizado en pasos anteriores y el número de muestras aunque no muy elevado (n = 51) es suficiente para aplicar esta prueba, en lugar de una no parámetrica como la de Mann-Whitney.

Una vez se ha realizado el anális univariado, se lleva a cabo el análisis multivariado. El Workflow POMA permite llevar a cabo análisis de componentes principales (PCA) de forma relativamente sencilla, por lo que se utilizan los datos (normalizados) para realizar el PCA.



Los resultados PCA, permiten explorar la estructura de los datos y reducir dimensionalidad. Los resultados gráficos permiten observar como los primeros dos componentes principales explican una variabilidad baja/moderada de los datos. El PC1 explica un 11.8% de la variabilidad de los datos y el PC2 explica un 5.19%. De esta forma, se observa que parece que las muestras de individuos con sarcopenia podrían tender a agruparse de manera diferente a las de individuos sin la enfermedad, aunque la variabilidad explicada por los 2 PCs no es muy elevada.

Otra opción del paquete POMA es calcular las correlaciones entre variables. Esto permite además, obtener las correlaciones ordenadas de las variables más correlacionadas a las menos correlacionadas.

El análisis de correlación entre las variables, calculándose por defecto los coeficientes de Pearson, muestra que existen metabolitos fuertemente correlacionados entre sí.

Continuando con los análisis, se llevó a cabo PCA por otra vía en lugar de utilizando la herramienta de POMA Workflow. De esta manera, aunque ya se ha realizado el PCA mediante el paquete POMA, puede realizarse por otra vía, y permitirá también comparar resultados y ver si ambos enfoques son equivalentes.

El primer paso consiste en escalar los datos para poder trabajar con la matriz de covarianzas de los datos centrados. En los siguientes outputs solo se mostrará parta del output con las funciones head() o cat("parte del output"), puesto que resulta demasiado extenso para mostrarse completamente.

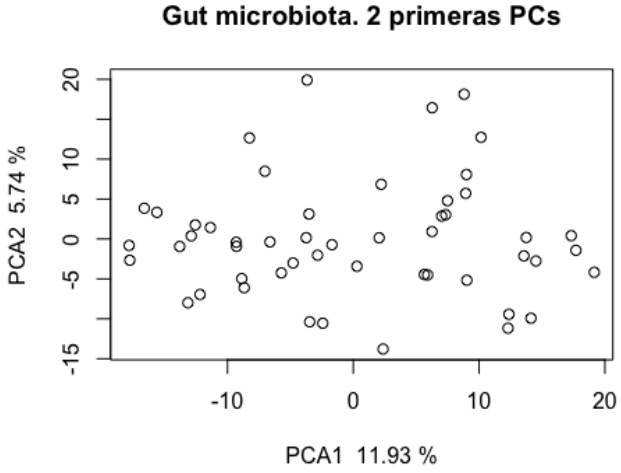
A continuación, se calcula la matriz de varianzas ajustada dividiendo entre n, y posteriormente se calcula la matriz de correlaciones.

Tras escalar los datos, la matriz de varianzas ajustada (S) y la matriz de correlaciones (R) muestran las relaciones entre los metabolitos.

Se comienza con el análisis PCA, empezando por el cálculo de las componentes principales. Los eigen$vectors son las coordenadas de las componentes principales.

Para obtener la transformación de los datos asociada a las componentes principales se obtiene multiplicando la matriz original por la matriz de vectores propio, y de esta forma llevar a cabo la representación de los componentes.

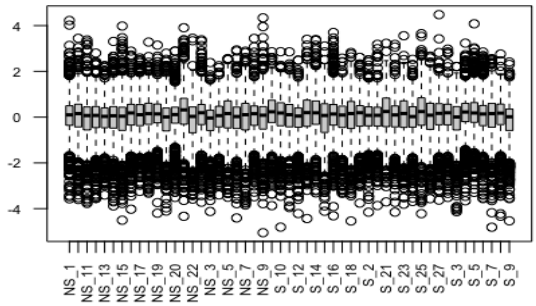
Los valores propios (eigenvalues) y los vectores propios (eigenvectors) de la matriz de covarianzas se obtenien para posteriomente determinar los componentes principales.



La representación gráfica muestra que los dos primeros componentes principales (PCA1 y PCA2) explicaron el 11.93% y 5.75% de la varianza total de los datos, en cada caso. Es posible observar que los resultados obtenidos en la realización de este PCA son muy similares a los resultados que se obtenían realizando el PCA con el paquete POMA Workflow, y las diferencias se deben a diferencias en los métodos empleados para realizar los cálculos.

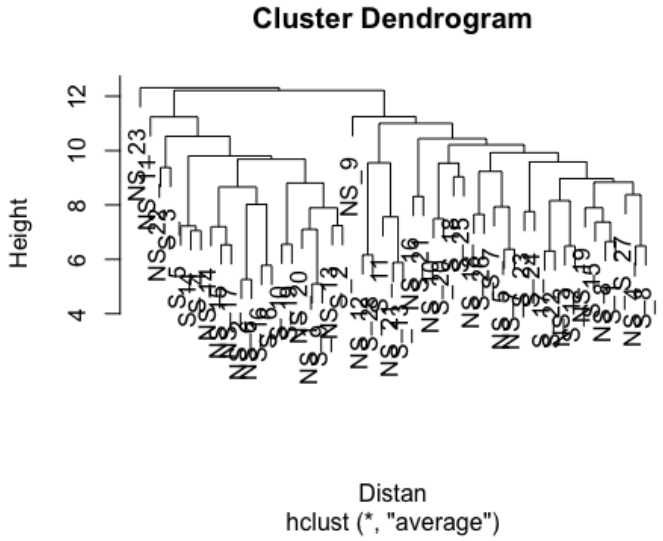
Por último, se realiza un cluster analysis o análisis de conglomerados, que permite agrupar las muestras en distintos grupos. En este caso se realizará una agrupación jerárquica de las muestras del dataset, para lo cual la estructura utilizada será el dendrograma.

Se comprueba mediante un diagrama de cajas (nuevamente), que efectivamente los datos se encuentran normalizados.



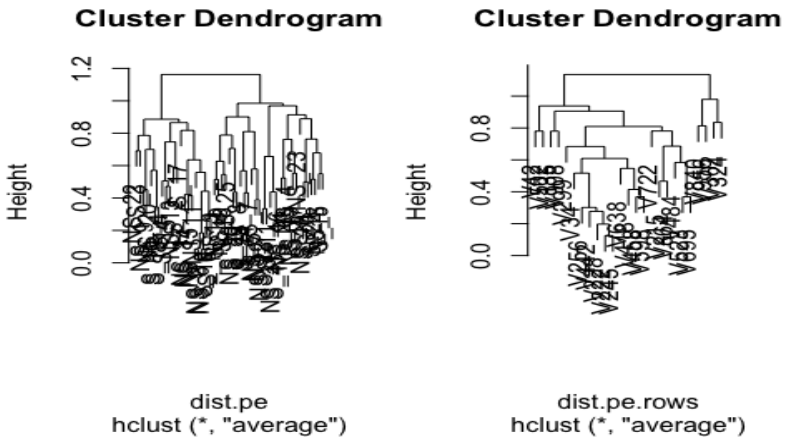
Se seleccionan solo aquellos metabolitos que posean el 1% de las desviaciones estándar más altas, ya que es una práctica habitual de los estudios de clasificación puesto que se considera que son los individuos con más variabilidad los que son más relevantes para poder realizar las agrupaciones.

Como aproximación inicial se realiza una agrupación jerárquica de los datos, basada en distancia euclídeas y “average linkage”.



El dendrograma muestra una división de las muestras en dos grupos o clusters principales, que a su vez se subdividen después en más grupos. Esta agrupación pueden reflejar diferencias subyacentes en las características de las distintas muestras en función de en qué grupo son agrupadas.

Por último, también es posible agrupar tanto las muestras del dataset como los metabolitos en función de los coeficientes de correlación. En este caso, se calculará la correlación de Pearson puesto que es la que se ha estado trabajando a lo largo de estos análisis, aunque también podría realizarse la agrupación jerárquica basada en la correlación de Spearman.



El dendrograma obtenido por el método de Pearson, muestra una estructura similar al obtenido previamente, ya que se observa que también hay dos agrupaciones o clusterings principales de las muestras, aunque presente algunas diferencias con el anterior. En este dendrograma también pueden observarse múltiples divisiones o subgrupos.
Por último, el dendrograma de los metabolitos también identifica dos o podrían considerarse tres clusters principales.

Discusión

El análisis realizado ha permitido explorar de manera detallada los datos metabolómicos de este estudio sobre la relación entre la microbiota intestinal y la sarcopenia. Durante el pre-procesamiento de los datos, se confirmó que no existían valores missing, lo que indica que la calidad del dataset parece ser alta. La normalización de los datos permitió reducir la variabilidad de los metabolitos y mejorar la comparabilidad entre muestras. Además, no se detectaron outliers significativos, lo que sugiere que la distribución de los datos es relativamente homogénea.

El análisis univariado utilizando la prueba t permitió identificar a los metabolitos con diferencias significativas entre los dos grupos de estudio, individuos con y sin sarcopenia, lo que apoyaría la hipótesis del estudio de que la microbiota intestinal y su composición metabólica podrían estar relacionadas con la sarcopenia.

El análisis de componentes principales (PCA) mostró que la variabilidad de los datos puede explicarse utilizando algunos componentes principales, sin embargo, la variabilidad explicada por estos se considera relativamente baja. Esto puede deberse a una gran hetorogeneidad entre las muestras comparadas, donde son múltiples factores los que influyen en la variabilidad de los datos y no únicamente el estado de sarcopenia o no sarcopenia de las muestras. El hecho de que la variabilidad explicada sea relativamente baja significa que hay mucha variabilidad en los datos que no se está capturando en los dos primeros componentes principales del análisis. Sin embargo, las similitudes en los resultados obtenidos realizando el PCA con el paquete POMA o el realizado paso a paso, refuerzan la robustez de los resultados. El POMA Workflow puede ser más sencillo y directo de implementar cuando se dispone de los datos contenidos en un objeto de clase SummarizedExperiment, puesto que el paquete está diseñado para trabajar con este tipo de objetos.

La correlación entre las variables permitió identificar metabolitos fuertemente correlacionados, al igual que también resulta relevante conocer si la correlación entre otros

metabolitos es baja y por qué. Los resultados obtenidos de las correlaciones entre metabolitos muestran que muchos de ellos se encuentran altamentente correlacionado, lo cual es esperable al tratarse de metabolitos procedentes del mismo tipo de muestras, pese a que pertenezcan a grupos de individuos diferentes. Además, ya se observaba en los resultados del PCA que aunque los componentes principales permiten explicar parte de la variabilidad observada en los datos, es probable que existan otros factores que estén influyendo en la variabilidad de los mismos, y no solo el hecho de que sean muestran de indivdiuos con y sin sarcopenia.

El análisis de agrupamiento jerárquico mostró la separación de las muestras en función de distintos perfiles. Aunque puede apreciarse una cierta agrupación o clustering tanto de las muestras, como de los metabolitos en uno de los dendrogramas obtenidos por el método de Pearson, la separación no es completamente definida, lo que refuerza la hipótesis de que existen otros factores que podrían influir en la agrupación de las muestras y metabolitos detectados en las muestras analizadas, aunque sí parecen observarse dos grandes grupos o clusters desde los que se subdividen las demás agrupaciones.

Conclusiones

En conclusión, en este estudio se han empleado diversas técnicas de análisis bioinformático y estadístico para explorar la relación entre la microbiota y la sarcopenia. La clase SummarizedExperiment permite trabajar de forma relativamente sencilla con datos ómicos contieniendo toda la información relevante del dataset. Los resultados muestran que el dataset utilizado presenta bastante calidad desde el inicio (no contiene valores faltantes o outliers). La consistencia de los dos análisis PCA demuestra robustez en la metodología utilizada, y el cluster análisis ha demostrado patrones de agrupación entre los datos analizados. Estos resultados respaldan la relevancia de los metabolitos analizados en la sarcopenia y proporcionan una base para futuras investigaciones. Sin embargo, la complejidad de la enfermedad requiere un enfoque integrativo que combine datos metabolómicos con otras tipos de información biológica para obtener una visión más completa de sus mecanismos.

Referencias

1. Mi repositorio de GitHub

2. Repositorio Metabolomics Workbench

3. The role of gut microbiota in muscle mitochondria function, colon health, and sarcopenia: from clinical to bench (2)

4. MetabolomicsWorkbenchR

5. SummarizedExperiment (apuntes de clase)

6. Información adicional sobre la clase SummarizedExperiment

7. POMA Workflow

8. POMA Workflow actualizado

9. Introduction to microarray data exploration and analysis with basic R functions

10. Casos y Ejemplos de Análisis Multivariante con R

Anexo

```
library(metabolomicsWorkbenchR)
library(SummarizedExperiment)

# Importar el dataset desde Metabolomics Workbench
se = do_query(
  context = 'study',
  input_item = 'study_id',
  input_value = 'ST003002',
  output_item = 'SummarizedExperiment'
)

se
```

```
## $AN004930
## class: SummarizedExperiment
## dim: 974 51
## metadata(8): data_source study_id ... description subject_type
## assays(1): ''
## rownames(974): ME776807 ME777597 ... ME776877 ME777611
## rowData names(3): metabolite_name metabolite_id refmet_name
## colnames(51): NS_1 NS_10 ... S_8 S_9
## colData names(6): local_sample_id study_id ... raw_data Status
##
## $AN004931
## class: SummarizedExperiment
## dim: 1077 51
## metadata(8): data_source study_id ... description subject_type
## assays(1): ''
## rownames(1077): ME778214 ME778659 ... ME778167 ME777709
## rowData names(3): metabolite_name metabolite_id refmet_name
## colnames(51): NS_1 NS_10 ... S_8 S_9
## colData names(6): local_sample_id study_id ... raw_data Status
```

```
# Comprobar el contenido de la matriz de expresión
head(assay(se[["AN004930"]]))

#Dimensiones de la matriz
dim(assay(se[["AN004930"]]))
```

```
## [1] 974  51
```

```
# Comprobar los metadatos de las muestras
colData(se[["AN004930"]])
```

```
## DataFrame with 51 rows and 6 columns
##       local_sample_id    study_id sample_source mb_sample_id    raw_data
##           <character> <character>   <character>  <character> <character>
## NS_1             NS_1    ST003002         Feces     SA326704   NS_1.mzML
## NS_10           NS_10    ST003002         Feces     SA326719  NS_10.mzML
## NS_11           NS_11    ST003002         Feces     SA326717  NS_11.mzML
```

```
## NS_12           NS_12    ST003002         Feces     SA326713  NS_12.mzML
## NS_13           NS_13    ST003002         Feces     SA326707  NS_13.mzML
## ...               ...         ...           ...          ...         ...
## S_5               S_5    ST003002         Feces     SA326737    S_5.mzML
## S_6               S_6    ST003002         Feces     SA326735    S_6.mzML
## S_7               S_7    ST003002         Feces     SA326736    S_7.mzML
## S_8               S_8    ST003002         Feces     SA326741    S_8.mzML
## S_9               S_9    ST003002         Feces     SA326742    S_9.mzML
##               Status
##             <factor>
## NS_1  non-sarcopenia
## NS_10 non-sarcopenia
## NS_11 non-sarcopenia
## NS_12 non-sarcopenia
## NS_13 non-sarcopenia
## ...               ...
## S_5       sarcopenia
## S_6       sarcopenia
## S_7       sarcopenia
## S_8       sarcopenia
## S_9       sarcopenia
```

```
library(POMA)
library(ggplot2)
library(ggraph)
library(plotly)

# Imputar los valores missing con POMA
imputed <- se[["AN004930"]] %>%
  PomaImpute(method = "knn", zeros_as_na = TRUE, remove_na = TRUE, cutoff =
20)
```

```
## No missing values detected

## 0 features removed.
```

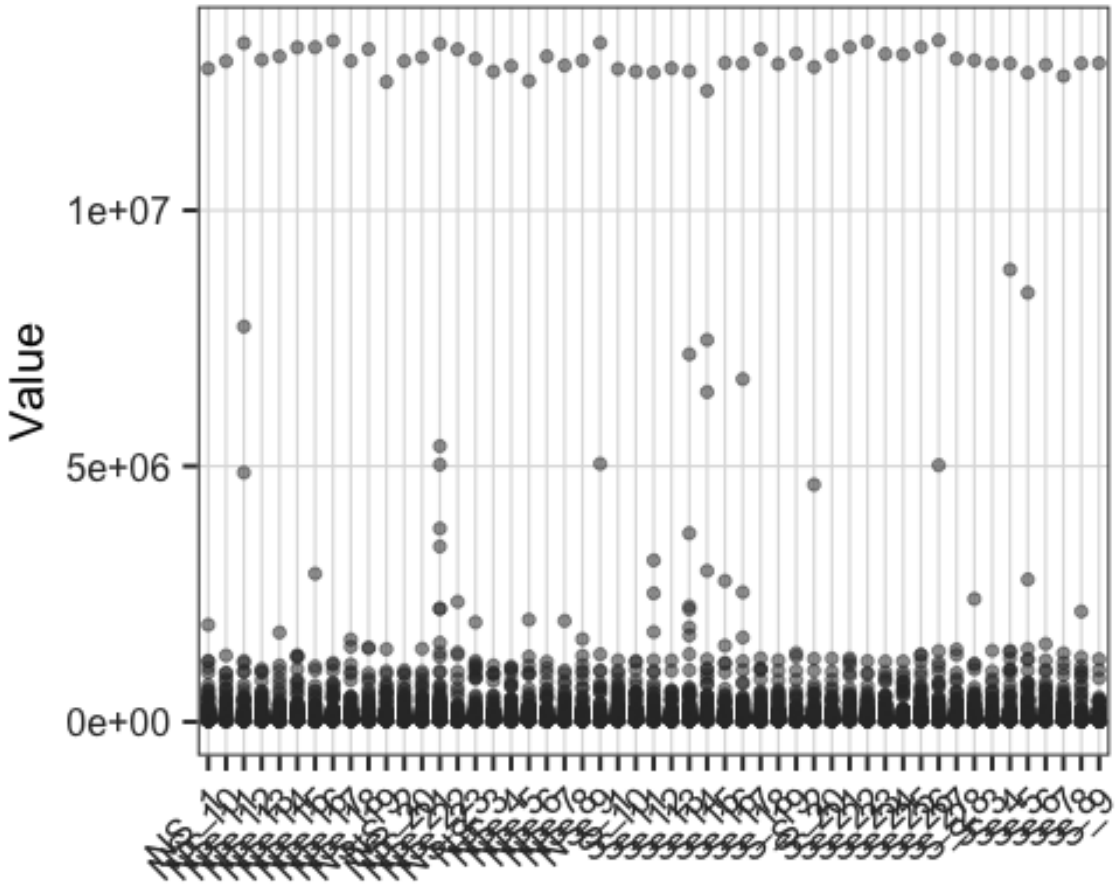
```
imputed
```

```
## class: SummarizedExperiment
## dim: 974 51
## metadata(0):
## assays(1): ''
## rownames(974): V1 V2 ... V973 V974
## rowData names(0):
## colnames(51): NS_1 NS_10 ... S_8 S_9
## colData names(6): local_sample_id study_id ... raw_data Status
```

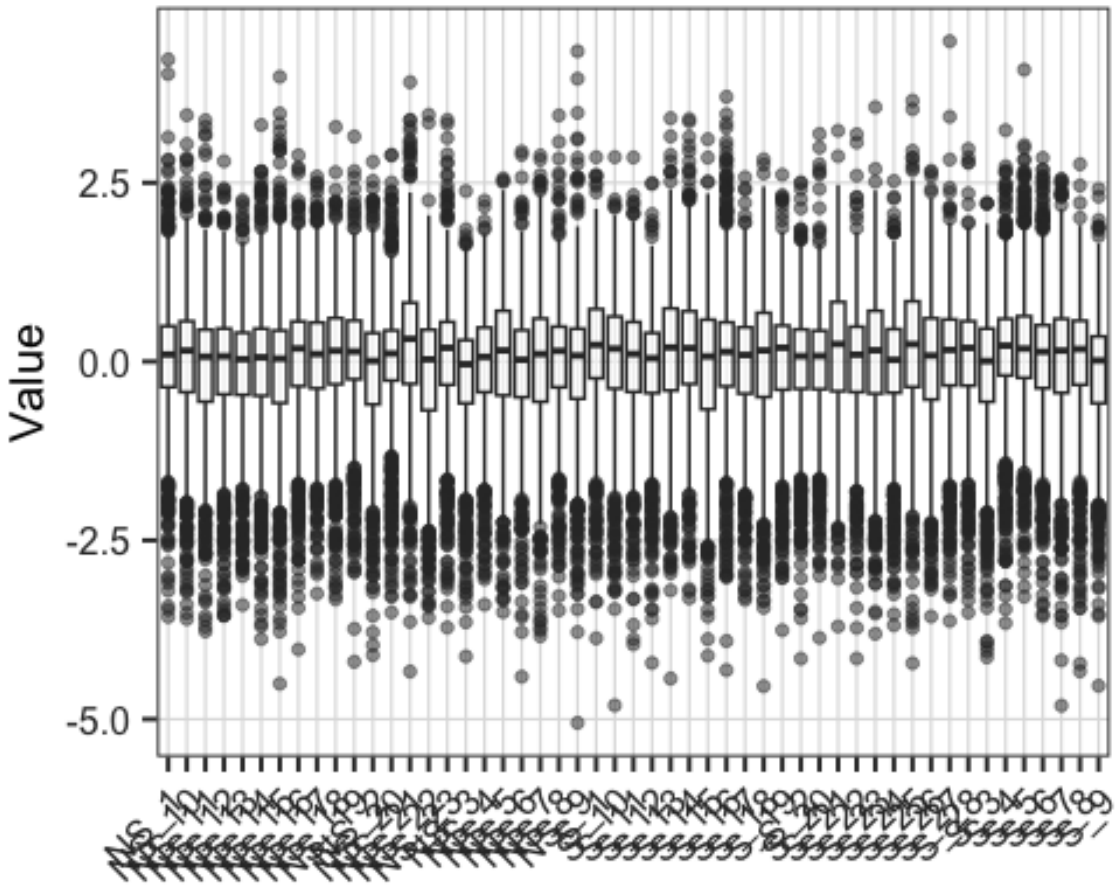
```
# Normalización de los datos
normalized <- se[["AN004930"]] %>%
  PomaNorm(method = "log_pareto")

normalized
```

```
## class: SummarizedExperiment
## dim: 974 51
## metadata(0):
## assays(1): ''
## rownames(974): V1 V2 ... V973 V974
## rowData names(0):
## colnames(51): NS_1 NS_10 ... S_8 S_9
## colData names(6): local_sample_id study_id ... raw_data Status
```

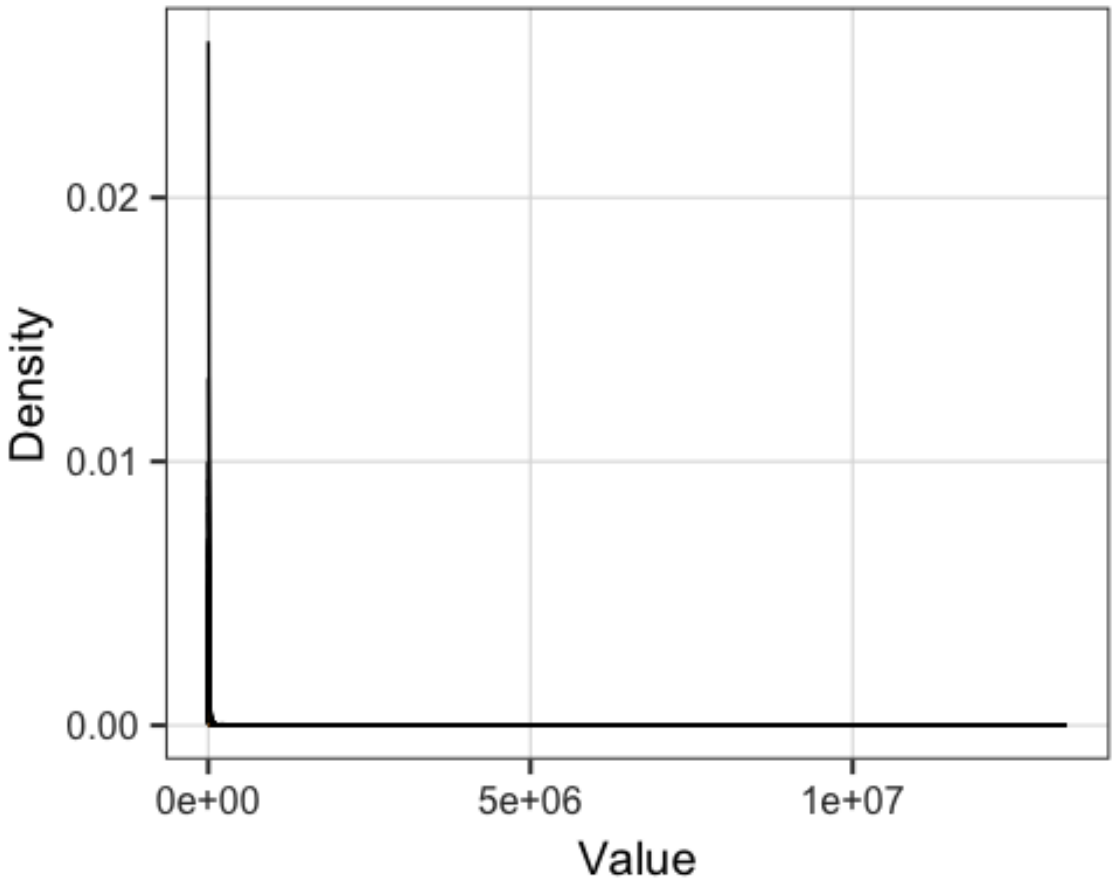
```
# Representación de los datos antes y después de la normalización
PomaBoxplots(se[["AN004930"]], x = "samples")
```



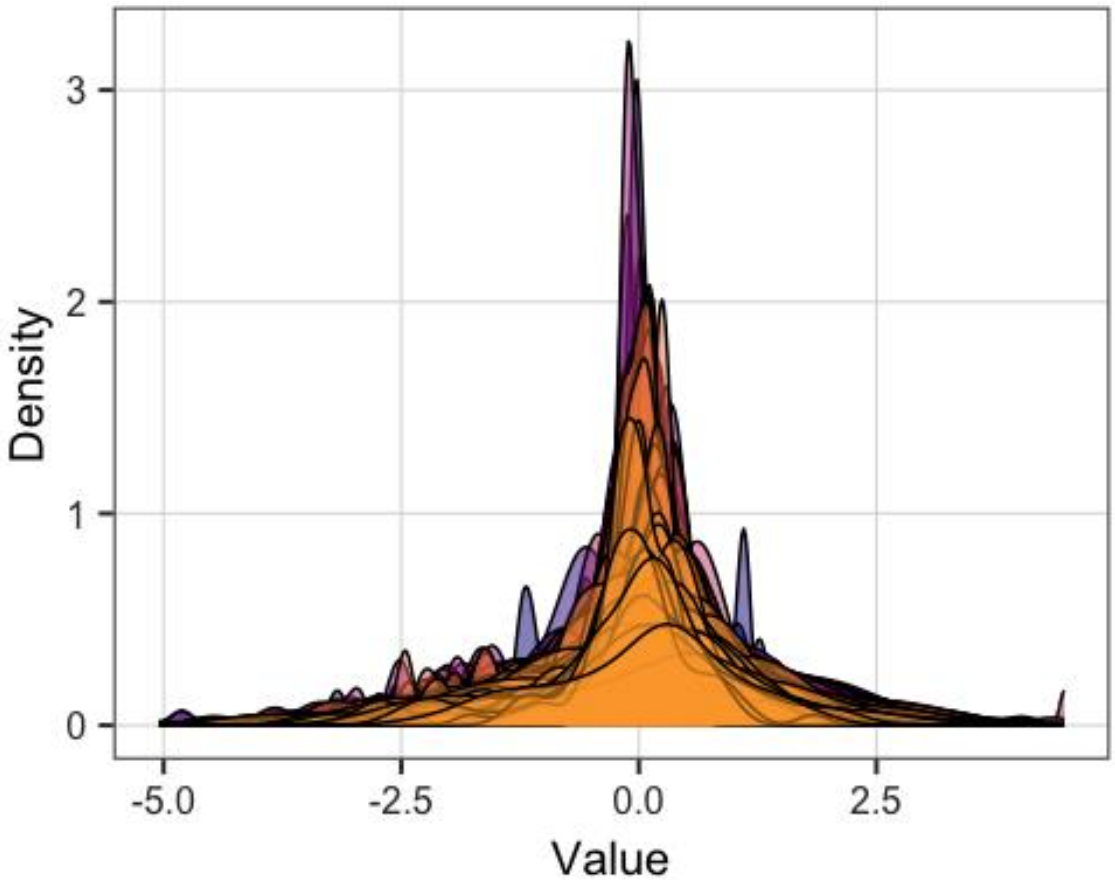

```
PomaBoxplots(normalized, x = "samples")
```



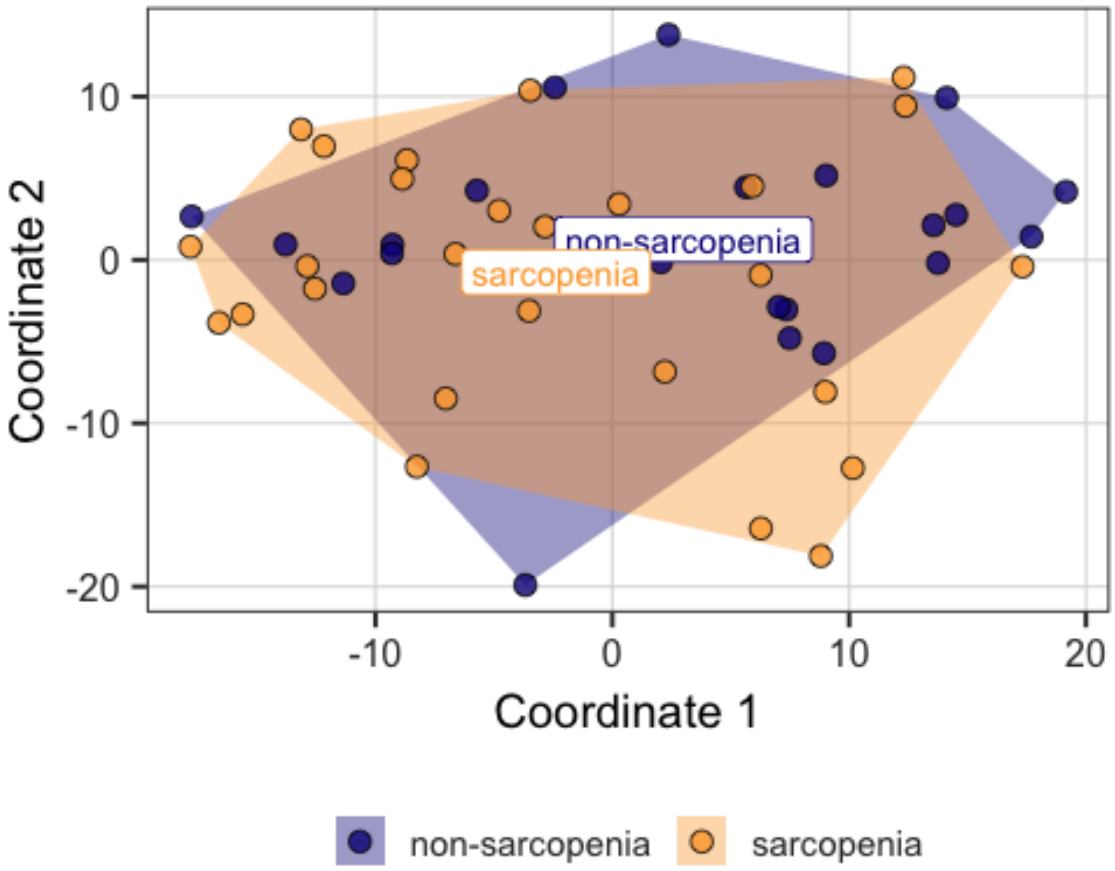

```
PomaDensity(se[["AN004930"]], x = "features", theme_params = list(legend_pos
ition = "none"))
```



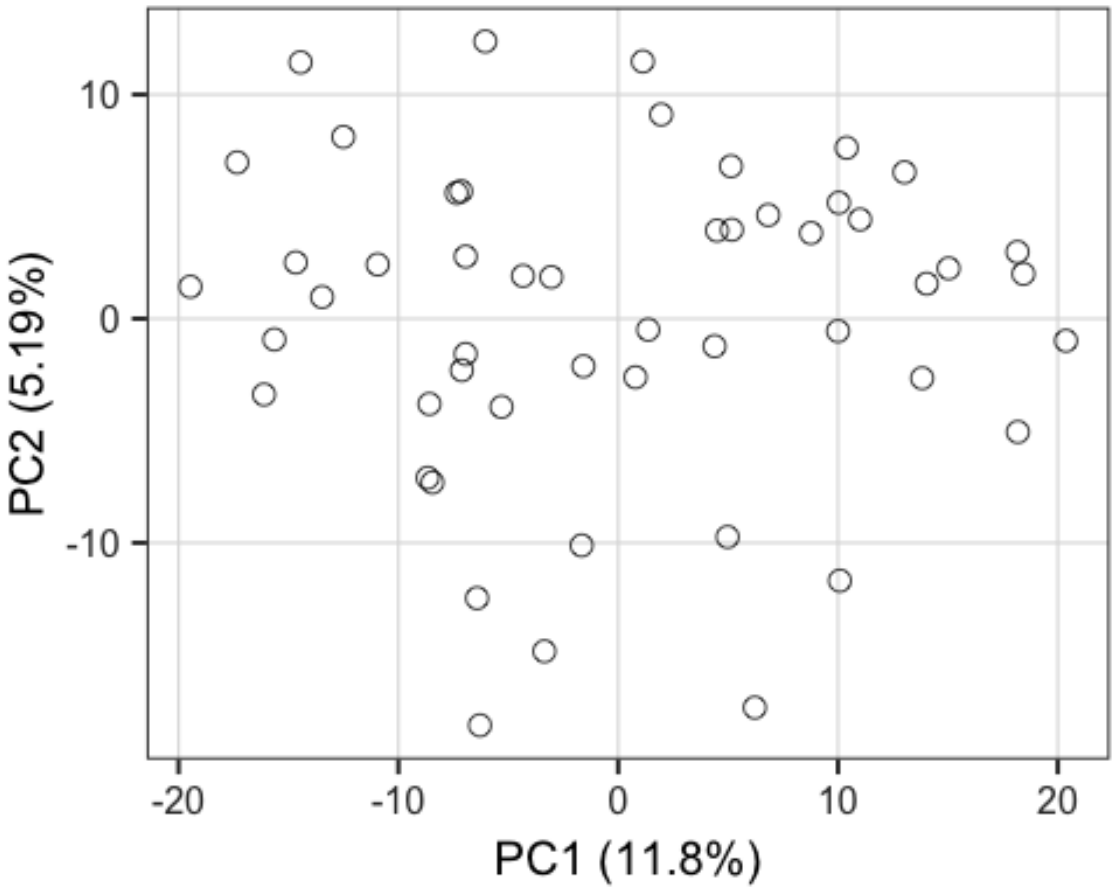

```
PomaDensity(normalized, x = "features", theme_params = list(legend_position
= "none"))
```




```
# Detección y eliminación (si se dan) de outliers
outlier_results <- normalized %>%
  PomaOutliers(method = "euclidean",
               type = "median",
               outcome = "Status",
               coef = 2,
               labels = FALSE)
outlier_results$polygon_plot
```




```
# Guardar los datos una vez se ha finalizado el pre-procesado
pre_processed <- outlier_results$data
pre_processed

## class: SummarizedExperiment
## dim: 974 51
## metadata(0):
## assays(1): ''
## rownames(974): V1 V2 ... V973 V974
## rowData names(0):
## colnames(51): NS_1 NS_10 ... S_8 S_9
## colData names(6): local_sample_id study_id ... raw_data Status

# Es necesario establecer "Status" que diferencia entre ambos grupos
# como variable que se empleará para realizar el ttest
pre_processed@colData <- pre_processed@colData[, c("Status", setdiff(colname
s(pre_processed@colData), "Status"))]

head(PomaUnivariate(pre_processed, method = "ttest"))

# Análisis de componentes principales
pca <- normalized %>%
  PomaPCA(outcome = "Status")

pca$factors_plot
```




```
# Cálculo de las correlaciones
poma_cor <- PomaCorr(normalized)
head(poma_cor$correlations)

# Extraer la matriz de datos del objeto SummarizedExperiment
data <- assay(normalized)
data <- as.matrix(data)
# Transponer la matriz si es necesario para tener muestras como filas
data <- t(data)

# Se escalan los datos
data <- scale(data, center = TRUE, scale=FALSE)
head(apply(data,2, mean))
```

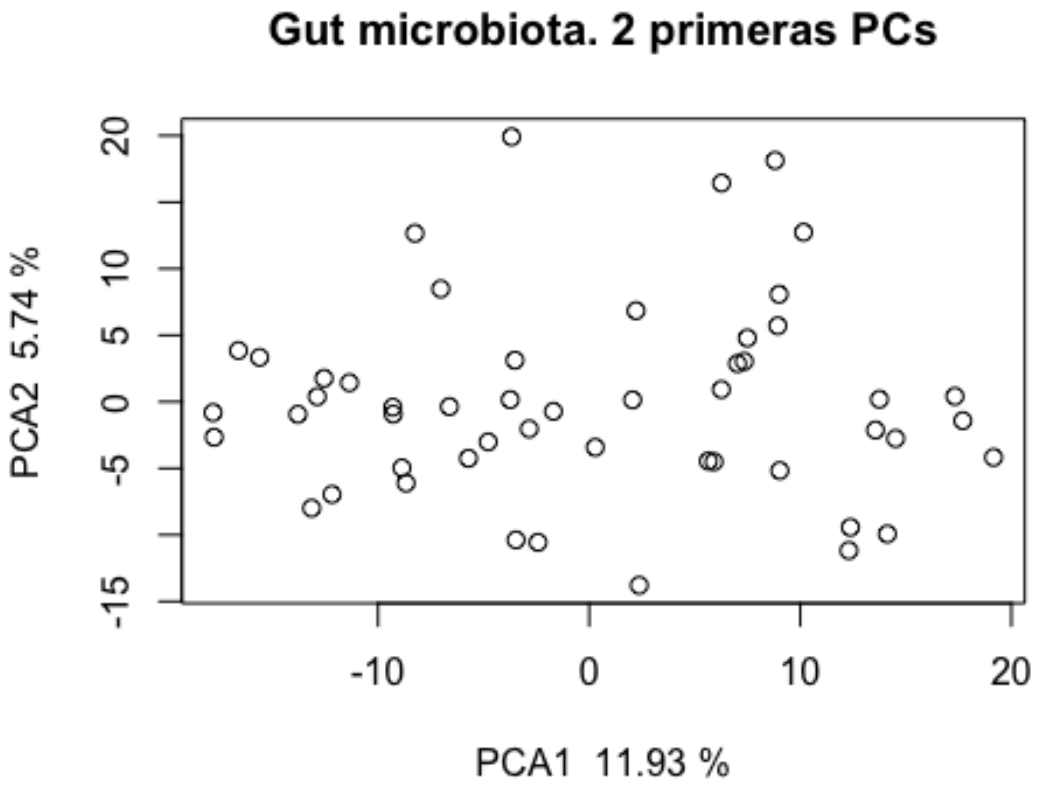
```
##            V1            V2            V3            V4            V5
## -3.537475e-17  3.700743e-17  1.023147e-16 -4.353816e-18 -7.619178e-18
##            V6
## -1.741526e-17
```

```
n<- dim(data)[1]
S<-cov(data)*(n-1)/n
cat("S output:", "\n", S[1:50], "\n")
```

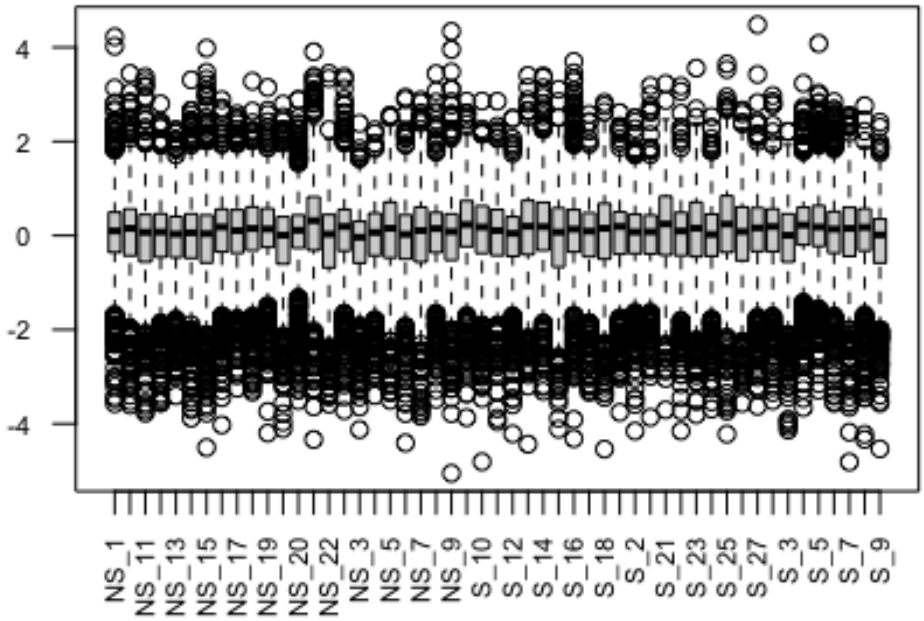
```
## S output:
##  0.6215589 0.1685287 0.09616762 0.1385614 -0.008686219 0.1623288 0.065245
```

```
23 -0.002377196 0.05195161 -0.06323925 -0.004156408 0.1764273 0.055548 0.210
3433 0.1715333 0.1362992 -0.1460973 0.09363502 -0.03579246 0.2632476 -0.0285
136 0.04477671 0.06771515 0.06039502 0.0222127 0.3083303 -0.06202652 0.00568
9491 0.3082154 -0.1089464 0.1173765 0.2740763 -0.11528 -0.4202177 -0.0313987
1 0.07348976 0.09083193 -0.02395873 -0.1816343 0.1409483 0.07011942 -0.08010
289 0.2664105 -0.1492261 -0.06964595 0.1983322 0.05263289 -0.09456732 0.0545
6628 0.2691832

R<-cor(data)
cat("R output:", "\n", R[1:50], "\n")

## R output:
##  1 0.2015466 0.1212662 0.1665394 -0.01840967 0.1930914 0.1097258 -0.00365
898 0.07049502 -0.3764305 -0.00743884 0.2325936 0.1004609 0.2769123 0.207676
1 0.1767037 -0.1613795 0.2052782 -0.0526778 0.3232281 -0.06751062 0.07165762
0.0906841 0.09453264 0.02273994 0.4091805 -0.08764793 0.007177866 0.3797849
-0.1142226 0.1462512 0.4219932 -0.1283162 -0.3606074 -0.03988987 0.1314802 0
.112209 -0.05012447 -0.2737054 0.1837426 0.09617038 -0.06381151 0.2866999 -0
.2433538 -0.100416 0.2840993 0.06943709 -0.1168723 0.07503135 0.3721475

EIG <- eigen(S)
cat("EIG values:", "\n", EIG$values[1:50], "\n")

## EIG values:
##  108.6132 52.25161 39.41451 35.20149 30.69882 27.92547 25.51061 24.51945
23.59994 21.87927 21.23715 20.11639 19.3583 19.2452 18.88684 18.19637 17.929
61 17.08755 16.77549 16.37455 15.95067 15.71153 15.4061 14.7662 14.48493 14.
07965 14.02 13.72091 13.21718 12.72404 12.16979 12.16355 11.562 11.36488 10.
79713 10.62909 10.51667 10.30696 10.18334 9.745231 9.63705 8.990235 8.673932
8.55046 8.276738 8.09029 7.96546 7.421631 7.194045 7.005242

cat("EIG vectors:", "\n", EIG$vectors[1:50], "\n")

## EIG vectors:
##  -0.03315405 0.02131121 -0.0358158 -0.0275213 0.01058974 -0.03685189 -0.0
2128026 -0.01231748 0.003299507 0.01009784 0.007643407 -0.05279755 -0.000791
6123 -0.03020251 -0.03772826 -0.005515258 0.02417714 0.003701881 0.01317098
-0.05842984 0.01265857 -0.01802887 0.006790654 -0.01111855 0.01164509 -0.045
90376 0.03918178 -0.01405944 -0.05435076 -0.01799435 0.008538575 -0.04666882
0.02139294 0.09299927 -0.02583058 -0.006999293 0.02424966 -0.008913902 0.034
69287 -0.01451502 0.01100556 0.05070926 -0.08889575 0.01318519 0.03161999 -0
.006610066 -0.002468012 0.03243447 -0.01145613 -0.06325945

eigenVecs1 <- EIG$vectors
PCAS1 <- data %*% eigenVecs1
cat("PCAS1", "\n", PCAS1[1:50], "\n")

## PCAS1
##  8.935337 -9.282765 13.56248 -5.718601 5.644357 17.69651 19.15574 -3.7428
89 7.491551 -9.302382 -11.35671 9.029378 7.345755 -3.672583 14.12392 7.02441
5 2.369753 -2.424687 -17.76365 14.51718 -13.79813 2.056127 13.75031 -7.02886
9 -1.692549 2.218208 -3.463768 -8.253219 8.804924 -6.621312 10.15494 6.25990
8 -12.88068 0.2877072 5.9136 -8.672998 -16.61277 -3.517125 -17.81625 -12.179
```

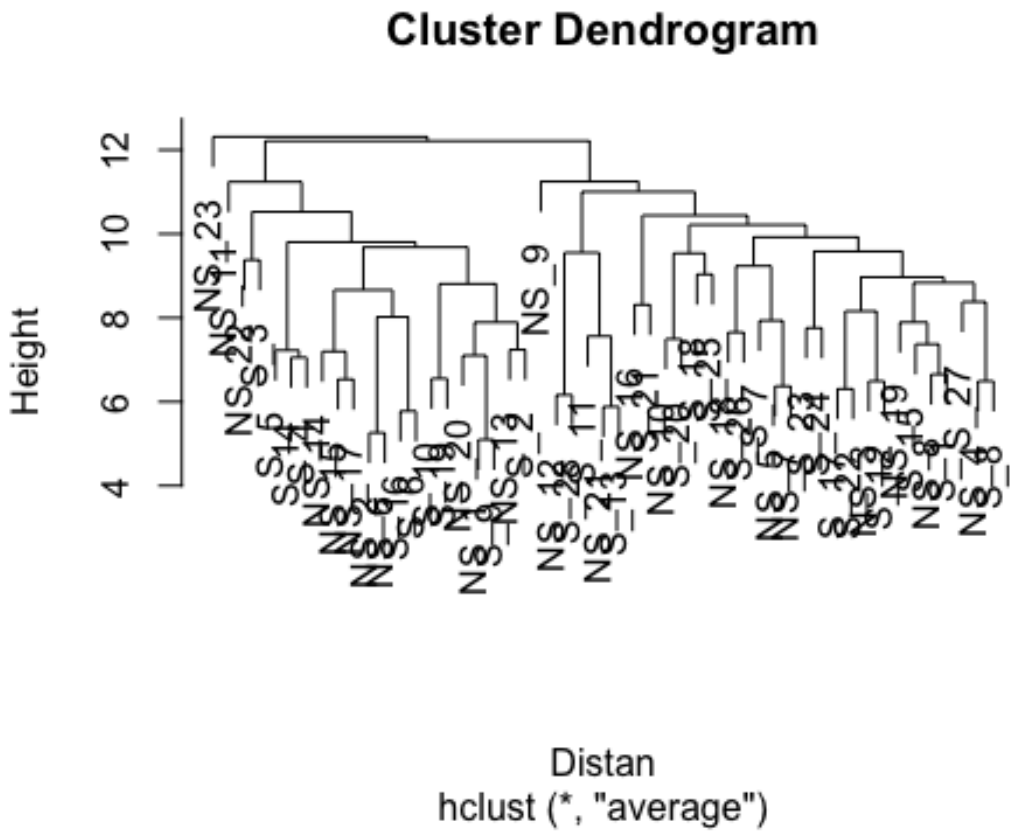
```
95 -15.61028 -12.55757 -4.783647 -8.868138 12.30145 8.994889 6.271645 17.322
08 -13.14391 -2.839614

vars1 <- EIG$values / sum(EIG$values)
invisible(round(vars1, 3))

#  Representación resultados PCA
xlabel <- paste("PCA1 ", round(vars1[1]*100, 2),"%" )
ylabel <- paste("PCA2 ", round(vars1[2]*100,2),"%" )
plot(PCAS1[,1], PCAS1[,2], main = "Gut microbiota. 2 primeras PCs",
     xlab=xlabel, ylab=ylabel)
```



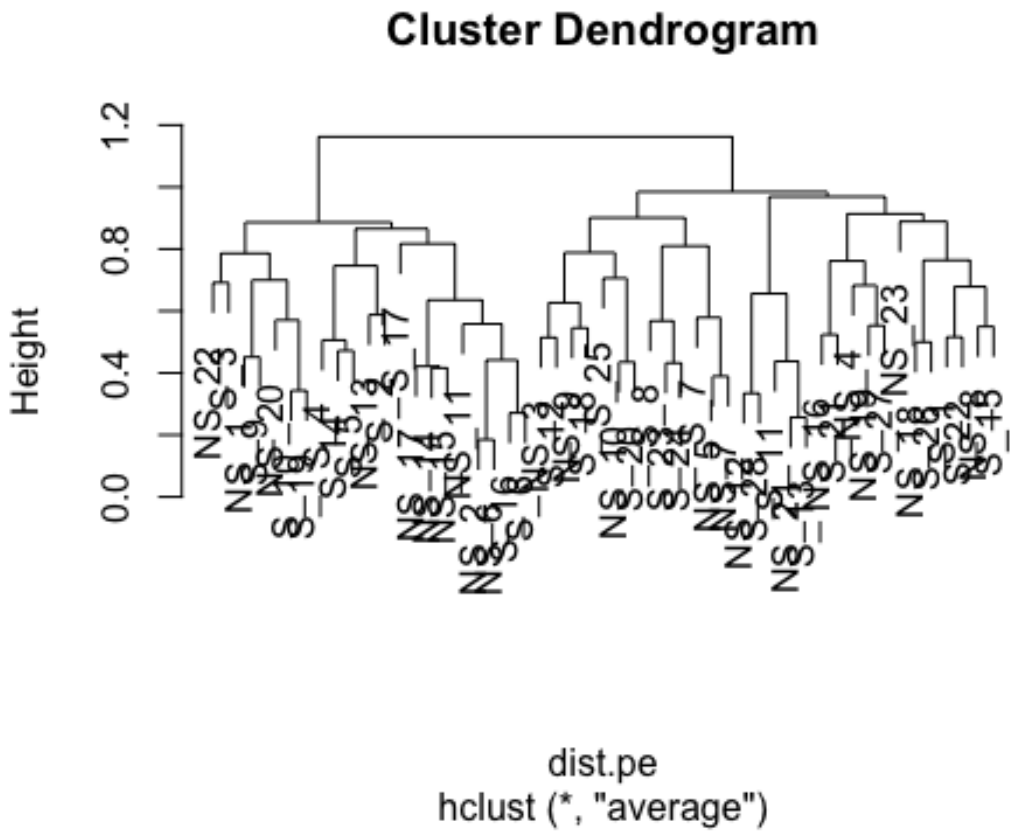

```
# Comprobación de datos normalizados
data <- assay(normalized)
data <- as.matrix(data)

boxplot(data, las=2, cex.axis=0.7)
```



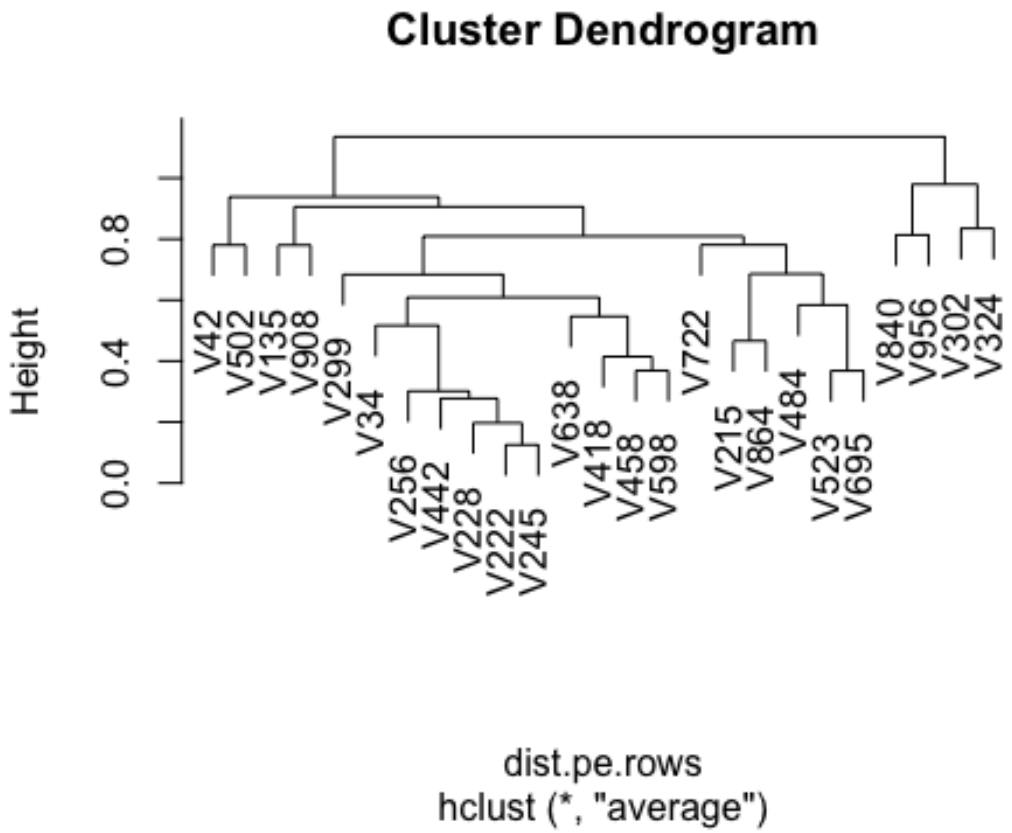

```
percentage <- c(0.975)
sds <- apply(data, MARGIN=1, FUN="sd")
sel <- (sds>quantile(sds,percentage))
data.sel <- data[sel, ]
dim(data.sel)

## [1] 25 51

# Agrupación jerárquica con distancia euclidiana
distmeth <- c("euclidian")
Distan <- dist(t(data.sel), method=distmeth)
treemeth <- c("average")
hc <- hclust(Distan, method=treemeth)
plot(hc)
```




```
# Agrupación jerárquica con el método de Pearson
cor.pe <- cor(as.matrix(data.sel), method=c("pearson"))
cor.pe.rows <- cor(t(as.matrix(data.sel)), method=c("pearson"))

dist.pe <- as.dist(1-cor.pe)
dist.pe.rows <- as.dist(1-cor.pe.rows)

hc.cor <- hclust(dist.pe, method=treemeth)
plot(hc.cor)
```




```
hc.cor.rows <- hclust(dist.pe.rows, method=treemeth)
plot(hc.cor.rows)
```


Cluster Dendrogram
Height
0.0
0.4
0.8
V42
V502
V135
V908
V299
V34
V256
V442
V228
V222
V245
V638
V418
V458
V598
V722
V215
V864
V484
V523
V695
V840
V956
V302
V324
dist.pe.rows
hclust (*, "average")