PEC 1

María Núñez León

2 de abril, 2025

Contents

RESUMEN

Se llevó a cabo un análisis bioinformático donde los datos fueron organizados en un objeto SummarizedExperiment (SE).

El análisis exploratorio incluyó un análisis de conglomerados, seleccionando un subconjunto de variables con mayor variabilidad. Se representaron gráficos de barras para visualizar diferencias entre los grupos identificados, destacando la concentración de hipurato, mayor en los sujetos que consumieron zumo de arándanos, salvo dos excepciones.

El análisis de conglomerados reveló dos grupos diferenciados:

- Uno compuesto mayoritariamente por sujetos que consumieron zumo de arándanos.
- Otro con el resto de las muestras del experimento.

Los resultados sugieren un posible efecto diferenciador de las procianidinas en el metabolismo.

OBJETIVOS Y JUSTIFICACIÓN

Los objetivos que persigue el siguiente trabajo se resumen en el siguiente apartado, así como la justificación del dataset seleccionado.

Se llevará a cabo la creación y exploración del objeto de clase SummarizedExperiment (SE) usando un dataset de metabolómica obtenido del siguiente repositorio, así como una explicación de las principales diferencias con la clase ExpressionSet (ES).

El dataset seleccionado, con identificador *2024-fobitools-UseCase_1*, consta de tres archivos que deben incorporarse al objeto de clase SummarizedExperiment. Esto representa una oportunidad para desarrollar habilidades en su manipulación durante la creación del objeto SE, garantizando una integración adecuada para una gestión sincronizada posterior.

Los datos también pueden encontrarse en el siguiente repositorio de Metabolomics Workbench con ID ST000291.

Por otro lado, se realizará un análisis exploratorio del dataset de modo que nos permita esclarecer si existe algún tipo de relación entre el consumo de alimentos ricos en procianidinas y los cambios producidos en el metabolismo. Esto abre la posibilidad de aplicar análisis de conglomerados (Cluster Analysis) para identificar posibles patrones o agrupaciones dentro de los datos y así observar si existen cambios significativos en el metabolismo.

MÉTODOS

Para el análisis bioinformático, se utilizaron los paquetes Biobase, BiocManager, SummarizedExperiment y tidyverse, junto con funciones base de R.

En primer lugar, se cargaron los tres archivos de datos obtenidos del repositorio mencionado previamente para construir un objeto SummarizedExperiment (SE) de la librería BiocManager. Se organizaron los datos en base al orden presente en el archivo cargado como assay_data. También se renombraron las filas de los archivos row_data y col_data para poder generar el objeto SE. Esto permitió la eliminación de datos faltantes en cada ranura, asegurando que todos los elementos estuvieran sincronizados.

El análisis exploratorio se realizó mediante un análisis de conglomerados, utilizando funciones base de R. Dada la gran cantidad de variables representadas, se seleccionó un subconjunto de aquellas con mayor variabilidad, por considerarse las más representativas.

Posteriormente, se generaron gráficos de barras para visualizar las diferencias entre los dos grupos identificados en el análisis de conglomerados. Se representaron cuatro pares de muestras:

- Un par compuesto por una muestra del grupo que consumió zumo de arándanos y otra del grupo en ayunas.
- Un par compuesto por una muestra del grupo que consumió zumo de arándanos y otra del grupo que consumió zumo de manzana.
- Dos pares adicionales, formados por muestras que, atendiendo al tratamiento, no deberían agruparse juntas en el cluster ya que no es la tendencia general observada.

Esta representación gráfica permitió obtener una visión general de las variaciones entre los grupos identificados en el clúster.

RESULTADOS

La clase SummarizedExperiment almacena matrices de datos experimentales, típicas en estudios de secuenciación y microarrays. Permite gestionar múltiples ensayos con dimensiones iguales y mantiene sincronizados los datos y metadatos al crear subconjuntos, de forma que al incluir una muestra se puede hacer para los datos y metadatos en una sola operación. Su flexibilidad en la representación de filas, ya sea con GRanges (estructura para trabajar con coordenadas genómicas y datos asociados) o DataFrames, la hace ideal para experimentos como RNA-Seq y ChIP-Seq, diferenciándola del antiguo ExpressionSet que está más orientado a microarrays.

La estructura de los datos de la clase SummarizedExperiment (1):

- colData: metadatos de las muestras
- rowData: metadatos de las características analizadas, en este caso metabolitos
- metadata: información general sobre el experimento

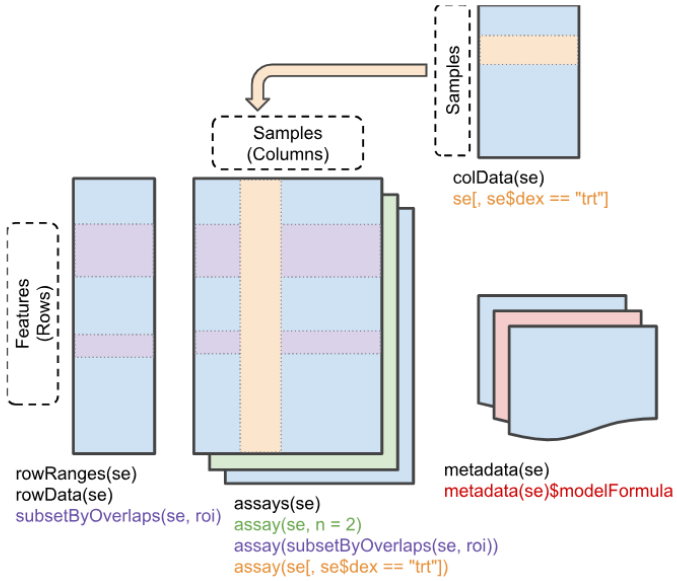


Figure 1: Estructura ExpressionSet

Para la creación del objeto SE se inspecciona la estructura de los datos cargados.

En el archivo assay_data, las columnas representan las observaciones y las filas corresponden a los metabolitos.

El archivo col_data tiene dos columnas: la primera contiene los ID de las observaciones, y la segunda ofrece una descripción del tratamiento aplicado a cada observación.

Por último, el archivo row_data tiene como filas los metabolitos y las columnas contienen diferentes nombres o identificadores para cada metabolito.

A continuación se realizan las modificaciones pertinentes para crear el objeto SE y se incorpora a los metadatos la descripción del experimento.

```
# Comprobamos que las muestras esten en el mismo orden en la matriz de
# datos de conteo y en la de información de las muestras y lo mismo para
# los datos de metabolitos.

#stopifnot(rownames(assay_data) == row_data$PubChem)
stopifnot(colnames(assay_data) == col_data$ID)

# Como no coinciden vamos a ordenar row_data en base a la matriz de datos
pubCherm <- rownames(assay_data)

row_data_ordered <- row_data[order(match(row_data$PubChem, pubCherm)), ]

stopifnot(rownames(assay_data) == row_data_ordered$PubChem)
```

```r
# Los nombres de las filas han de coincidir con los nombres de las filas
# del assay_data por lo que definimos el nombre de las filas correctamente

rownames(row_data_ordered) <- row_data_ordered$PubChem

# Realizamos lo propio para los nombres de las filas que han de coincidir
# con los nombres de las columnas del assay_data

rownames(col_data) <- col_data$ID

# Crear el objeto SE

se <- SummarizedExperiment(assays = list(counts = assay_data),
                           colData = col_data,
                           rowData = row_data_ordered)


# Añadimos metadatos
metadata(se) <- list(
  Experiment_Description = "Dieciocho estudiantes universitarias sanas
  (21-29 años, IMC normal) participaron en el estudio. Se les pidió evitar
  alimentos ricos en procianidinas (arándanos, uvas, chocolate, etc.) antes
  del experimento. El día 7, se recolectaron muestras de orina y sangre en
  ayunas. Luego, fueron divididas en dos grupos (n=9) para consumir jugo de
  arándano rojo o de manzana durante tres días (mañana y noche). El día 10,
  se tomaron nuevas muestras tras el ayuno nocturno y 30 minutos después de
  ingerir jugo. Tras dos semanas, se intercambiaron los regímenes y se
  repitió el procedimiento. Tres participantes fueron excluidas por falta de
  muestras. El objetivo era analizar los cambios metabólicos inducidos por las
  procianidinas mediante LCMS. Todas las muestras se almacenaron a -80°C hasta
  su análisis."
)

# Objeto SE
se
```

```
## class: SummarizedExperiment
## dim: 1541 45
## metadata(1): Experiment_Description
## assays(1): counts
## rownames(1541): 443489 107754 ... 53297445 11954209
## rowData names(3): names PubChem KEGG
## colnames(45): b1 b10 ... c8 c9
## colData names(2): ID Treatment
```

```r
#Guardar el objeto SE
save(se, file="SummarizedExperiment_data.Rda")

# Dimensiones del objeto
dim(se)
```

```
## [1] 1541   45
```

Análisis exploratorio de los datos mediante clustering con R.

Primero se lleva a cabo la busqueda de valores faltantes en los datos y eliminación de todas aquellas filas que contienen valores NA.

Para ello generamos una función que introduzca en un vector las coordenadas de las filas donde aparecen los valores NA y os devuelva un objeto con las filas eliminadas.

Comprobamos que las dimensiones son iguales en las distintas ranuras y que nuestro objeto funciona correctamente.

```
options(scipen = 999)

sum(is.na(assay(se)))
```

```
## [1] 8190
```

```
# arr.ind==TRUE devuelve las cordenadas por fila-columna en una matriz.
# En caso contrario nos devuelve la posicion que ocupa cada NA dentro de
# la matriz pero realizando un conteo del numero de datos recorriendo las
# columnas.


delete_na <- function(objeto_se){
  filas_na <- unique(which(is.na(assay(objeto_se)), arr.ind = TRUE)[,1])
  objeto_se <- objeto_se[-filas_na[[1]] : -filas_na[[length(filas_na)]] ,]
}

se <- delete_na(se)

dim(se)
```

```
## [1] 1359   45
```

```
dim(assay(se))
```

```
## [1] 1359   45
```

```
dim(rowData(se))
```

```
## [1] 1359    3
```

```
dim(colData(se))
```

```
## [1] 45  2
```

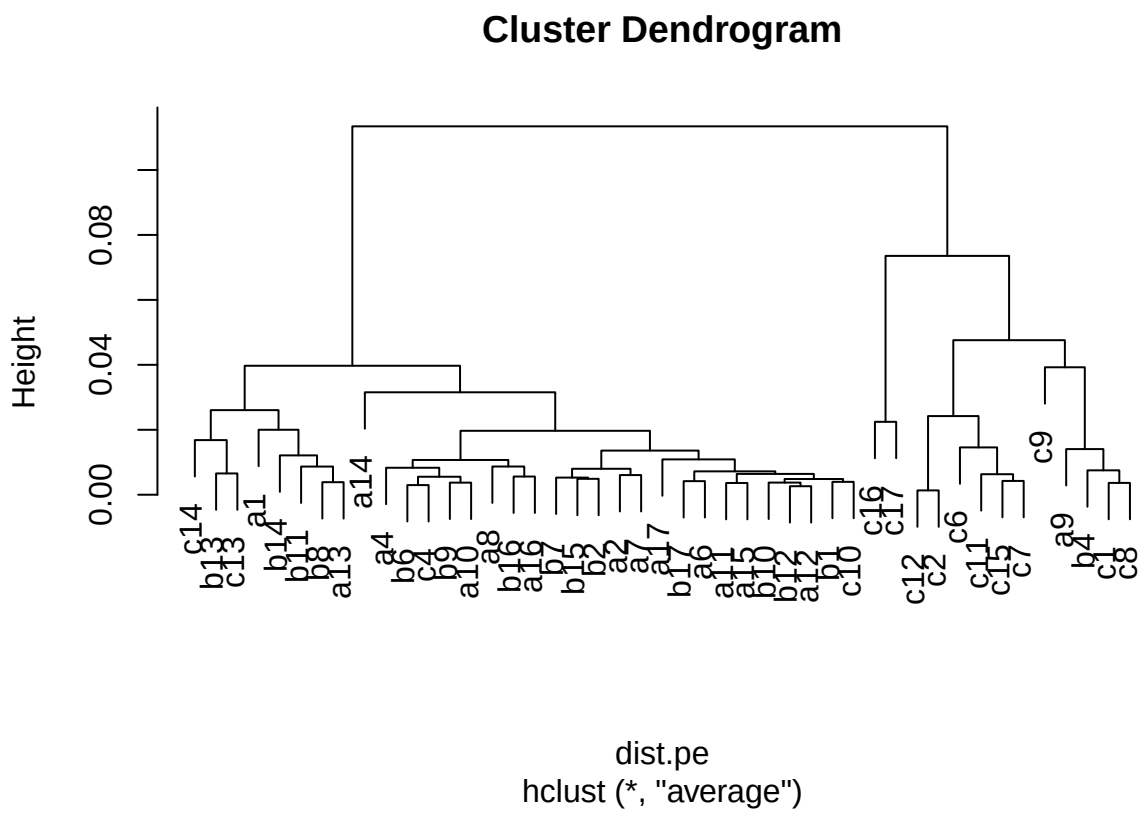
Realizamos un clustering para comprobar si existen grupos en los datos.

Primero seleccionamos aquellos metabolitos que presentan variabilidad en las muestras, aquellos con las desviaciones estándar más altas y posteriormente normalizamos los datos.

```r
# Obtener los metabolitos que presentan el umbral más alto de
# desviación estándar

datos_normalizados <-  scale(assay(se))
percentage <- c(0.975)
sds <- apply(datos_normalizados, MARGIN=1, FUN="sd")

sel <- (sds>quantile(sds,percentage))

# el vector lógico se usa para filtrar los valores de sd más altos
dat_sel <- datos_normalizados[sel, ]
dim(dat_sel)
```

```
## [1] 34 45
```

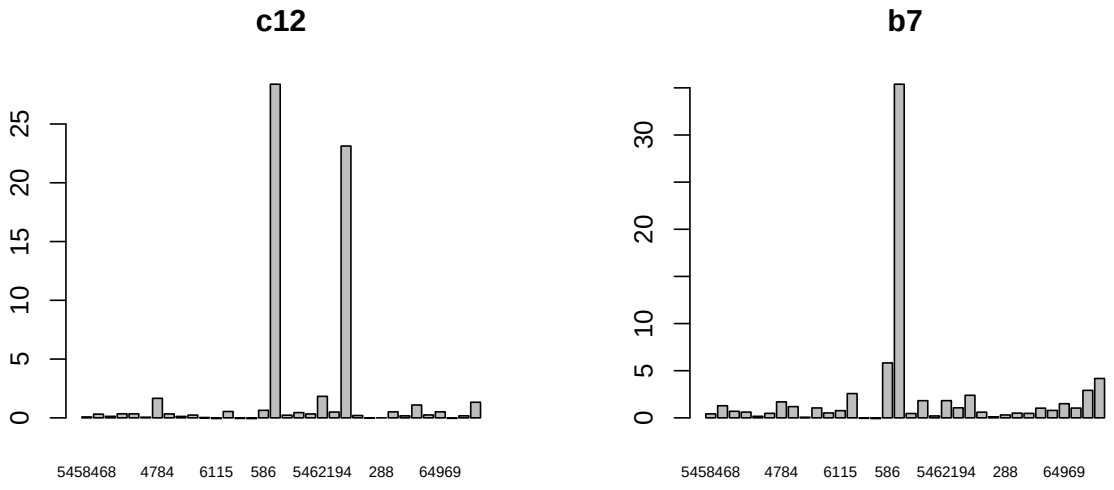
```r
# Correlación de Pearson

cor.pe <- cor(as.matrix(dat_sel), method=c("pearson"))

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

hc.cor <- hclust(dist.pe, method="average")

plot(hc.cor)
```



```r
# vamos a comparar como varia los metabolitos mas influyentes en una muestra
# de cada grupo del dendrograma


col_names <- colnames(dat_sel)

# seleccionamos un conjunto de muestras de cada grupo para compararlas
col_repr <- c("c12","b7","c7","a2","a9","c10","b4","c14")

vec_pos <- c()
for (pos in col_repr) {
  vec_pos<-c(vec_pos,which(col_names == pos))
}

# vector de posicion de las columnas seleccionadas
vec_pos
```

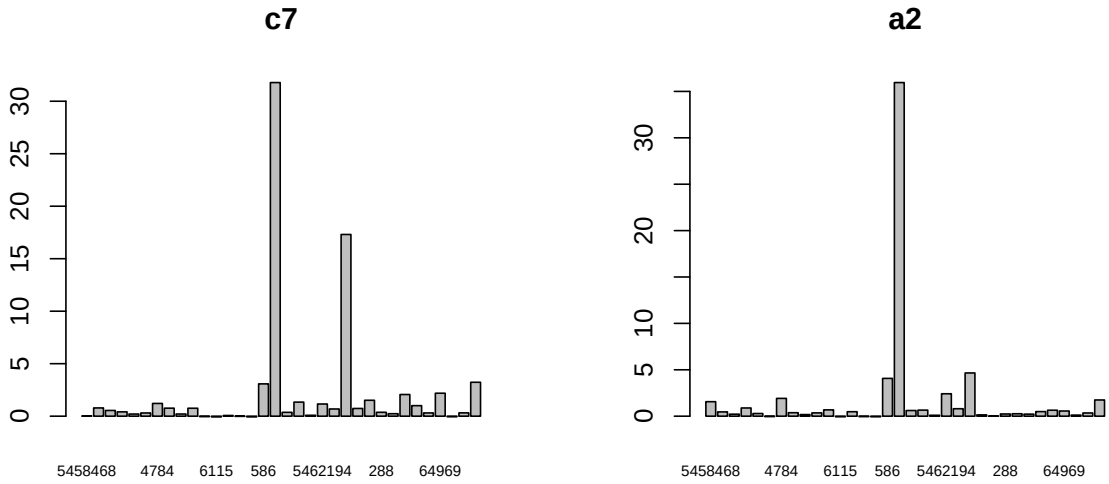
```
## [1] 34 13 43 25 30 32 11 36
```

```r
# Establecer dos graficas en la misma linea
par(mfrow = c(1, 2))

# Gráfico de barras para la columna 34
barplot(dat_sel[, vec_pos[1]], main = colnames(dat_sel)[vec_pos[1]], cex.names = 0.6)

# Gráfico de barras para la columna 13
barplot(dat_sel[, vec_pos[2]], main = colnames(dat_sel)[vec_pos[2]], cex.names = 0.6)
```

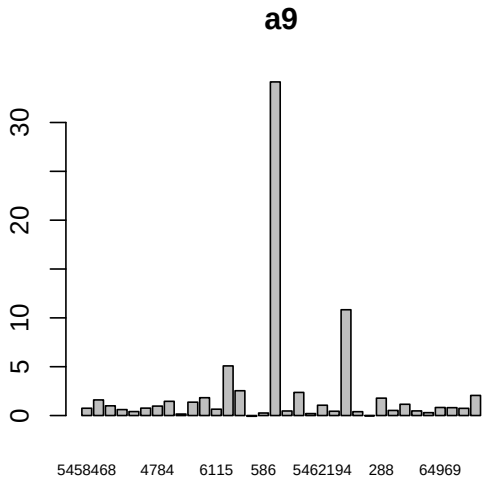


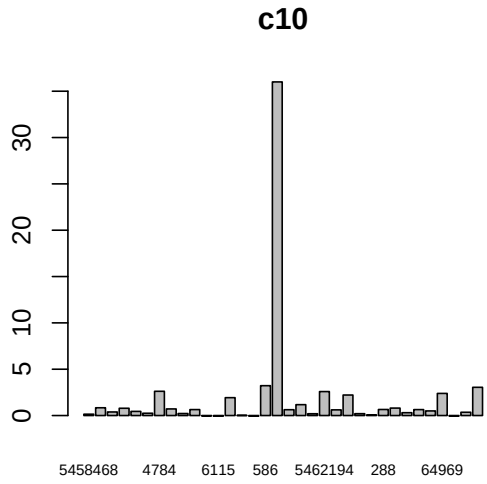

```r
# Restablecer la configuración de gráficos
par(mfrow = c(1, 1))
```

```r
#---
par(mfrow = c(1, 2))

barplot(dat_sel[, vec_pos[3]], main = colnames(dat_sel)[vec_pos[3]], cex.names = 0.6)

barplot(dat_sel[, vec_pos[4]], main = colnames(dat_sel)[vec_pos[4]], cex.names = 0.6)
```

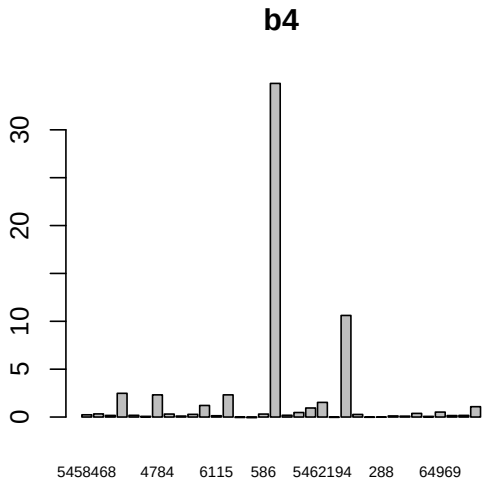


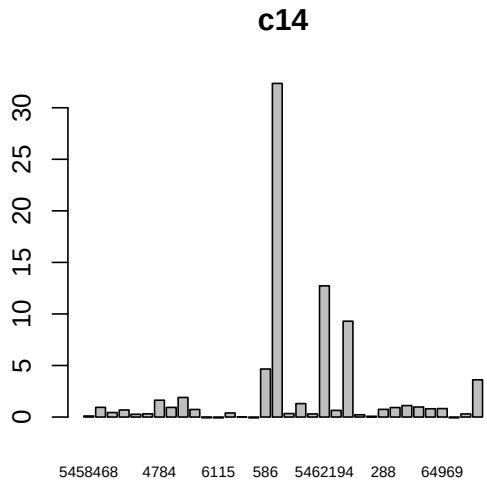

```r
par(mfrow = c(1, 1))

#---
par(mfrow = c(1, 2))

barplot(dat_sel[, vec_pos[5]], main = colnames(dat_sel)[vec_pos[5]], cex.names = 0.6)

barplot(dat_sel[, vec_pos[6]], main = colnames(dat_sel)[vec_pos[6]], cex.names = 0.6)
```






```
par(mfrow = c(1, 1))

#---
par(mfrow = c(1, 2))

barplot(dat_sel[, vec_pos[7]], main = colnames(dat_sel)[vec_pos[7]], cex.names = 0.6)

barplot(dat_sel[, vec_pos[8]], main = colnames(dat_sel)[vec_pos[8]], cex.names = 0.6)
```






```
par(mfrow = c(1, 1))
```

DISCUSIÓN

El análisis realizado en este estudio ha permitido explorar el posible impacto del consumo de alimentos ricos en procianidinas en el metabolismo. El uso de herramientas bioinformáticas y la construcción del objeto SummarizedExperiment (SE) ha sido efectivo para organizar, sincronizar y visualizar los datos pero existen diversas limitaciones que deben ser consideradas.

En primer lugar, el tamaño de la muestra es limitado y presenta un sesgo en el sexo de los participantes, esto podría llevar a una generalización en los resultados. Si bien se especifica que las participantes estaban sanas existen otros factores biológicos y ambientales como el ciclo menstrual, la actividad física y factores genéticos que pueden influir en un análisis de datos metabólicos.

Por otro lado, la selección de las variables que presentaban mayor variabilidad nos permitie reducir la complejidad del análisis pero introducen sesgo en el análisis.

En conjunto, estas limitaciones subrayan la necesidad de un análisis complementario que incorpore una muestra más diversa, considere factores adicionales y explore metodologías alternativas para la selección de variables.

CONCLUSIONES

La representación de los metabolitos con mayor variación destaca el Hipurato, donde se observa una mayor concentración en los grupos que consumieron zumo de arándanos, a excepción de dos sujetos que podrían haberlo obtenido mediante el consumo de otros alimentos. El hippurato es generalmente excretado en la orina como un metabolito del ácido benzoico, que se encuentra comúnmente en ciertos alimentos como las bayas, las frutas y algunos vegetales.

El análisis de conglomerados revela dos grupos claramente definidos:

- Grupo donde se encuentran 11 de las 15 muestras de sujetos que tomaron zumo de arándanos, una muestra perteneciente al grupo baseline y otra al que grupo que consumió zumo de manzana.
- Grupo con el resto de muestras del experimento.

Parece existir un efecto diferenciador en el metabolismo tras el consumo de procinidinas. Sin embargo, aunque el análisis de conglomerados ha permitido identificar posibles patrones de agrupación, es necesario un estudio más profundo para determinar su relevancia biológica.

REFERENCIAS