

Alumna: Consuelo Marín-Vicente

Asignatura: Análisis de datos ómicos (ADO)

PEC 1: datos completos se encuentran en el repositorio:

https://github.com/cmarinvic/MARIN_VICENTE_CONSUELO_PEC1_1

1. Tabla de contenidos:

- 2. Resumen—
- 3. Objetivos—
- 4. Métodos—
- 5. Resultados—
- 6. Discusión—
- 7. Conclusiones—
- 8. Referencias—

2. Resumen:

Las tecnologías ómicas tales como transcriptómica, proteómica o metabolómica, incluyen un gran número de datos que se deben manejar con las herramientas bioinformáticas adecuadas. En este trabajo inspeccionamos un dataset de metabolómica. En este data set se analizan las expresiones de los diferentes metabolitos adquiridos mediante espectrometría de masas. Concretamente se pretende estudiar el impacto de la curcumina sobre el metabolismo. Para hacer esta inspección se han usado herramientas de Bioconductor. Igualmente, aunamos todos los procesos seguidos para depositarlos en un repositorio público de github con el fin de compartir la información elaborada.

3. Objetivos:

En este trabajo planteamos los siguientes objetivos:

- Extraer los datos de metabolómica de la página Metabolomics Workbench.
- Crear un objeto de clase SummarizedExperiment para poder evaluar el dataset.
- Realizar una inspección o análisis exploratorio del dataset para evaluar calidad, distribuciones y outliers.
- Interpretar los datos obtenidos en el apartado anterior.
- Compartir los procedimientos y resultados en un repositorio de github.

4. Métodos:

Utilizamos la página Metabolomics Workbench para extraer el dataset objeto de estudio. Utilizamos Bioconductor (Gentleman et al., 2003) para generar un objeto del tipo SummarizedExperiment (Fig.1)

Figura 1: Creación de objeto SummarizedExperiment.

```
# Crear el objeto SummarizedExperiment
se_object <- SummarizedExperiment(
  assays = list(counts = data_matrix), # Los datos de expresión
```

```

# No hay metadatos de las columnas (o puedes añadirlos si los tienes
)
)

# Verifica el objeto SummarizedExperiment
se_object

## class: SummarizedExperiment
## dim: 9814 144
## metadata(0):
## assays(1): counts
## rownames(9814): 413.858@0.70853347 440.0761@0.6910459 ...
## 402.2583@1.114 975.1623@5.697
## rowData names(0):
## colnames(144): X17may15_004.r001 X17may15_004.r002 ...
## X18may15_040.r001 X18may15_040.r002
## colData names(0):

```

Al inspeccionar el archivo veo que los metadatos están incluidos en el mismo archivo descargado. En este archivo se incluyen en las columnas los metabolitos identificados con la masa del precursor analizado así como la anotación al metabolito concreto si es el caso.

En las primera y segunda filas se incluyen los nombres de las muestras y la descripción de cada una de ellas, respectivamente.

Me quedo con la información de la primera fila en el archivo a estudiar, y la información de la primera columna. Igualmente, extraigo los metadatos en archivo adicional (dos primeras filas y la primera columna: metadata.xlsx).

A nivel metodológico es importante destacar la diferencia entre ExpressionSet y SummarizedExperiment, tal como se destaca a continuación:

- **ExpressionSet:** La clase ExpressionSet es parte del paquete **Biobase** de Bioconductor. Es una clase de datos más antigua que se utiliza principalmente para almacenar datos de expresión génica, como las matrices de expresión, junto con metadatos de las filas (genes) y las columnas (muestras). Se asocia principalmente con datos de arrays y algunos tipos de datos de expresión. En cuanto a su estructura, los datos de expresión se almacenan principalmente en una matriz de expresión (gen x muestra). Los metadatos relacionados con las filas (genes) y las columnas (muestras) se almacenan como objetos AnnotatedDataFrame. Es relativamente rígido en su estructura y no soporta de manera eficiente múltiples tipos de datos dentro del mismo objeto. En cuanto a compatibilidad con nuevos paquetes, aunque muchos paquetes antiguos y enfoques de análisis se basan en ExpressionSet, está un poco más limitado en comparación con el estándar actual de clases en Bioconductor.
- **SummarizedExperiment:** La clase SummarizedExperiment, que pertenece al paquete **SummarizedExperiment**, es más moderna y flexible que ExpressionSet. Está diseñada para ser más general y se utiliza para almacenar y manejar datos de cualquier tipo de experimento omico. Es una clase más versátil y extensible que

permite manejar datos más complejos. Tiene una estructura más flexible: el objeto puede almacenar **varios "assays"** (por ejemplo, matrices de expresión para diferentes condiciones o tecnologías) dentro de una lista. Además, los metadatos de las filas y las columnas se almacenan en **rowData** y **colData** como objetos **DataFrame**. Esto permite una mayor flexibilidad en la gestión de diferentes tipos de datos y metadatos. Está diseñado para ser la clase de elección en nuevos desarrollos en Bioconductor.

Utilizamos las herramientas en R para generar los datos exploratorios que describimos en el apartado de Resultados.

5. Resultados:

5.1. Descripción biológica del dataset:

El set de datos utilizado es ST000567, que incluye un único archivo disponible para ser descargado. En este proyecto se estudia la influencia de la curcumina en el metabolismo del organismo como factor aminorador del riesgo cardiovascular.

La edad es uno de los mayores factores de riesgo en enfermedades cardiovasculares. Dos parámetros que contribuyen a este riesgo son la rigidez de las arterias y el desarrollo de disfunciones endoteliales que se indican mediante EDD (impaired nitric oxide induced endothelium dependent dilation. Disfunción en la dilatación del endotelio causada por óxido nítrico). La reducción de la rigidez en arterias y el incremento de la función endotelial vascular mediante el incremento en la disponibilidad del NO reducen el riesgo de enfermedad. Estudios preliminares han mostrado que el uso de curcumina en la dieta ayuda a reducir dicho riesgo. El objetivo del proyecto seleccionado es el estudio del metaboloma plasmático tras la administración oral de curcumina.

5.2. Análisis exploratorio y resultados:

Un análisis exploratorio de datos normalmente incluye varios pasos, como la inspección de las dimensiones del conjunto de datos (Figura 2), la visualización de las distribuciones de las variables (como la distribución de las intensidades de los metabolitos, Figura 3, o la visualización de las intensidades por muestra (Figura 4), así como de las intensidades estandarizadas (Figura 5)), la evaluación de la calidad de los datos y la detección de posibles outliers (Figura 6 y 7). Esto lo vamos a hacer con funciones de exploración sobre el objeto creado tal cual se muestra en el R script y en el código destacado más abajo (ver R script: MARINVICENTE_CONSUELO_ADO_PEC1_3.R y en archivo MARIN_VICENTE_CONSUELO_ADO_PEC1_RCODE_AND_RESULTS.pdf)

Los box plots nos muestran una similitud en la distribución de la señal por muestra. La distribución de las muestras en base a esta señal deja ver en el PCA una acumulación de las muestras en un cluster común mientras que hay algunas muestras que son outliers de este agrupamiento. El análisis de variabilidad nos indica cuáles son los metabolitos más abundantes a lo largo del ensayo.

Figure 2. Análisis exploratorio sobre la estructura de los datos

#ANÁLISIS EXPLORATORIO:

Ver el objeto SummarizedExperiment
se_object

```
## class: SummarizedExperiment
## dim: 9814 144
## metadata(0):
## assays(1): counts
## rownames(9814): 413.858@0.70853347 440.0761@0.6910459 ...
## 402.2583@1.114 975.1623@5.697
## rowData names(0):
## colnames(144): X17may15_004.r001 X17may15_004.r002 ...
## X18may15_040.r001 X18may15_040.r002
## colData names(0):
```

Obtener dimensiones de la matriz de expresión
dim(se_object)

```
## [1] 9814 144
```

Ver Los nombres de Las filas (metabolitos)
head(rownames(se_object))

```
## [1] "413.858@0.70853347" "440.0761@0.6910459" "542.0147@0.6911163"
"
## [4] "565.0907@0.69125646" "418.0785@0.7215334" "667.0658@0.7038285"
"
```

Ver Los nombres de Las columnas (muestras)
head(colnames(se_object))

```
## [1] "X17may15_004.r001" "X17may15_004.r002" "X17may15_005.r001"
## [4] "X17may15_005.r002" "X17may15_006.r001" "X17may15_006.r002"
```

#Para calcular la media:

Acceder a la matriz de datos

```
data_matrix <- assays(se_object)$counts
```

Calcular la media y desviación estándar por fila excluyendo la primera columna

```
row_means <- rowMeans(data_matrix[, 2:ncol(data_matrix)], na.rm = TRUE
)
```

```
row_sds <- apply(assays(se_object)$counts, 1, sd)
```

```

# Ver Las primeras medias calculadas
head(row_means)

## 413.858@0.70853347 440.0761@0.6910459 542.0147@0.6911163 565.090
7@0.69125646
## 545.042 24545.615 10704.748
15148.420
## 418.0785@0.7215334 667.0658@0.7038285
## 3826.483 53671.280

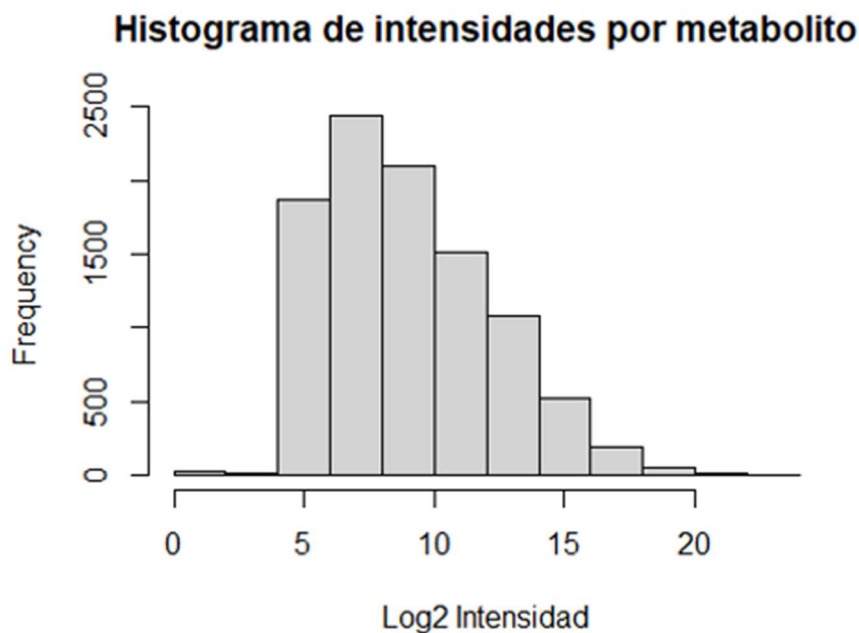
# Calcular La media y desviación estándar por muestra (columna)
col_means <- colMeans(assays(se_object)$counts)
col_sds <- apply(assays(se_object)$counts, 2, sd)

#Visualización de datos:

# Histograma de Las intensidades para todos Los metabolitos (filas)
hist(log2(row_means + 1), main = "Histograma de intensidades por metab
olito", xlab = "Log2 Intensidad")

```

Figura 3: Distribución de intensidades de metabolitos (gráfico y código inmediatamente anterior).



```

# Boxplot de Las intensidades para Las muestras (columnas)
boxplot(assays(se_object)$counts, main = "Boxplot de las intensidades
por muestra", las = 2)

```

Boxplot de las intensidades por muestra

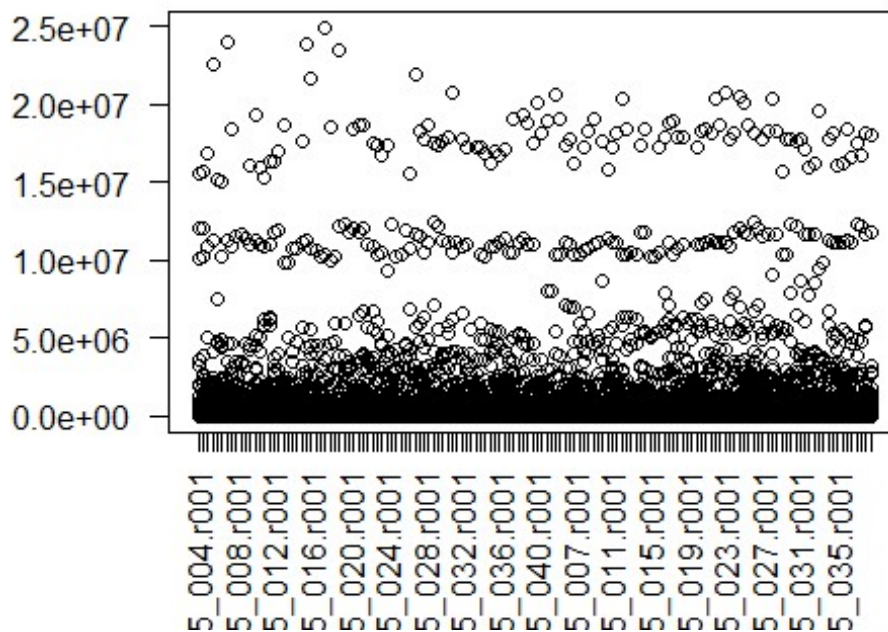


Figura 4: Intensidades por muestra (gráfico y código inmediatamente anterior).

```
#Escalado y normalización de Los datos:
```

```
# Log-transformación de Los datos de expresión
```

```
log_counts <- log2(assays(se_object)$counts + 1)
```

```
# Boxplot después de log-transformar Los datos
```

```
boxplot(log_counts, main = "Boxplot de las intensidades log-transformadas por muestra", las = 2)
```

oxplot de las intensidades log-transformadas por mi

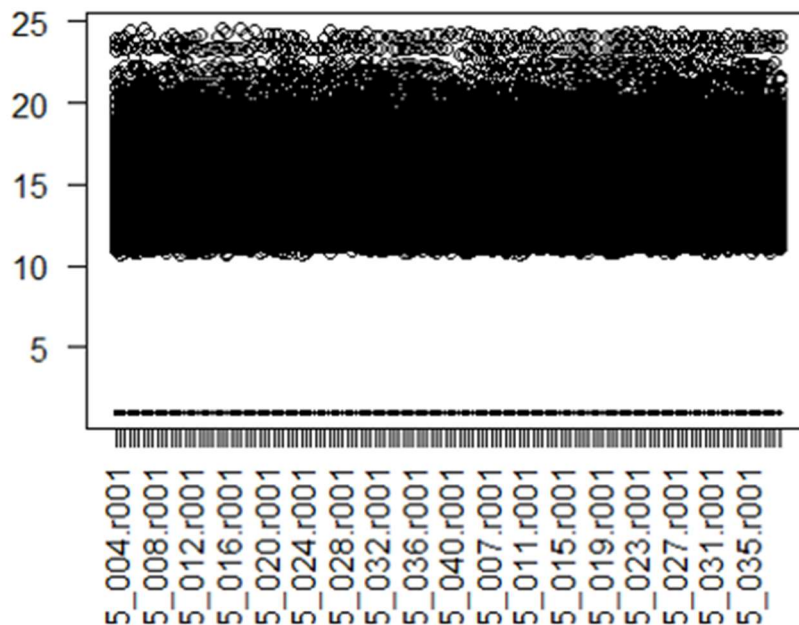


Figura 5: Intensidades estandarizadas por muestra (gráfico y código inmediatamente anterior).

```
# Realizar PCA sobre los datos
pca_result <- prcomp(t(log_counts), center = TRUE, scale. = TRUE)

# Visualizar los resultados del PCA
library(ggplot2)
pca_data <- data.frame(pca_result$x)

# Graficar las primeras dos componentes principales
ggplot(pca_data, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = colData(se_object)$condition)) + # Cambia
  'condition' por el nombre de la variable de interés
labs(title = "PCA de las Muestras", x = "PC1", y = "PC2") +
  theme_minimal()
```

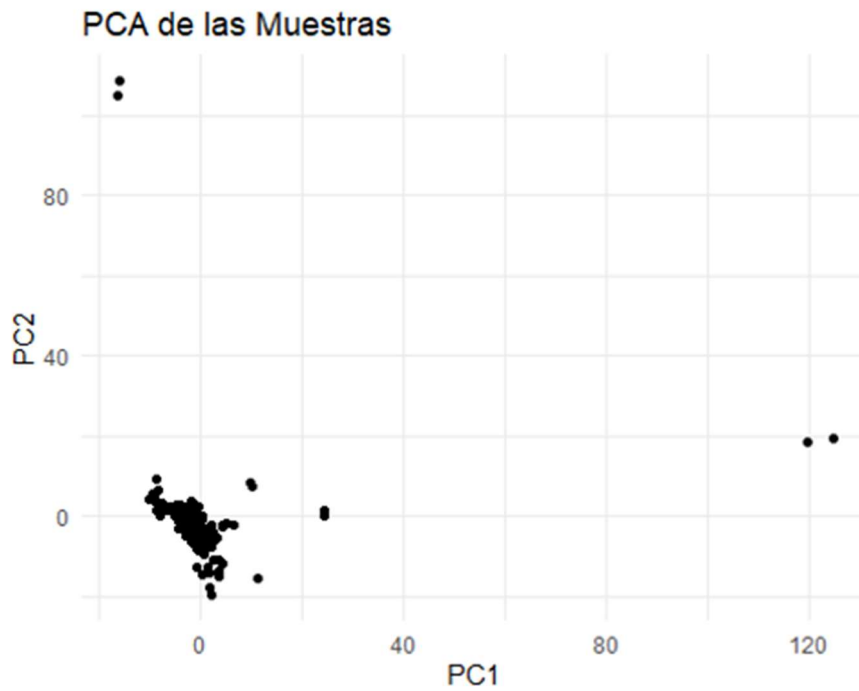


Figura 6: PCA para evaluar dispersión de datos y outliers (código inmediatamente anterior)

```
#Estudiar variabilidad:
```

```
# Calcular la varianza por metabolito
```

```
var_per_metabolite <- apply(assays(se_object)$counts, 1, var)
```

```
# Visualizar los metabolitos con mayor varianza
```

```
top_var_metabolites <- order(var_per_metabolite, decreasing = TRUE)[1:  
10]
```

```
barplot(var_per_metabolite[top_var_metabolites], main = "Metabolitos c  
on mayor variabilidad")
```

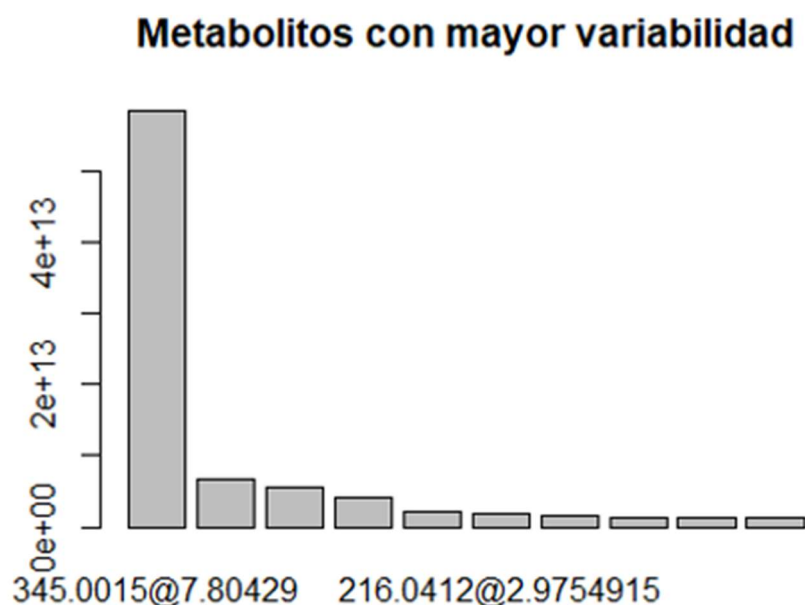



Figura 7: Estudio de metabolitos con mayor variabilidad y posibles outliers (código inmediatamente anterior)

6. Discusión:

Las herramientas bioinformáticas son de gran ayuda para inspeccionar los datasets de experimentos ómicos. Cientos de miles de espectros en el caso de la espectrometría de masas pueden ser aglutinados en las métricas usadas para analizar la distribución de la población. De gran importancia es la estandarización de los datos y evaluar la calidad de los mismos mediante herramientas de análisis multivariante como el PCA. En metabolómica, gran cantidad de metabolitos están definidos con la masa de su precursor, análisis de anotación son esenciales para establecer a qué metabolito corresponde cada masa y así poder interpretar los datos obtenidos a nivel biológico.

7. Conclusiones: Las herramientas bioinformáticas como bioconductor son esenciales para gestionar datos ómicos y poder evaluar la calidad de los mismos.

8. References:

Gentleman R, Carey VJ. Visualization and annotation of genomic experiments, in Parmigiani G, Garrett ES, Irizarry RA, Zeger SL, eds. The Analysis of Gene Expression Data, NY: Springer, 2003