PEC 1. ANÁLISIS DE DATOS ÓMICOS

Helena Ortiz Rivero

Contents

1. ABSTRACT

En este proyecto se analiza el conjunto de datos *"Metabotypes of response to bariatric surgery independent of the magnitude of weight loss"*, cuyo propósito es la identificación de perfiles metabólicos o "metabotypes" en pacientes sometidos a cirugía bariátrica. Esta cirugía tiene la finalidad de propiciar la reducción de peso en pacientes con obesidad grave, provocando en consecuencia cambios metabólicos que pueden variar significativamente entre pacientes.

En este estudio, se busca no solo comprender la variabilidad en la respuesta clínica y metabólica de los pacientes sino también desarrollar perfiles que ayuden a predecir la respuesta postquirúrgica independientemente de la cantidad de peso perdido. Esto permitiría diseñar intervenciones terapéuticas individualizadas que optimicen los beneficios en cada paciente.

El análisis incluye la creación de un contenedor de datos mediante SummarizedExperiment en R para la estructuración y visualización de datos clínicos y metabolómicos. Este enfoque permite una organización óptima para el análisis exploratorio y la interpretación preliminar de los datos, sentando las bases para análisis más complejos en el futuro.

---

2. OBJETIVOS

El objetivo de este análisis es:

- Preparar y cargar los datos de un dataset de metabolómica.
- Crear un contenedor de tipo `Summarized Experiment` para almacenar y organizar los datos y metadatos.
- Realizar un análisis exploratorio de los datos, evaluando su calidad y su distribución.
- Crear un repositorio en GitHub para presentar el análisis de los datos.

---

3. MATERIALES Y MÉTODOS

3.1. Selección y descarga del dataset

El dataset seleccionado incluye datos clínicos y metabolómicos de 39 pacientes con obsesidad severa sometidos a cirugía bariátrica. La monitorización se llevó a cabo en cuatro momentos distintos (timepoints) antes y después de la intervención, lo que permite capturar tanto el impacto inmediato como los cambios a largo plazo. Este enfoque longitudinal ofrece una visión integral de cómo la cirugía afecta al metabolismo en diversas etapas de recuperación, más allá de la simple reducción de peso.

El dataset está disponible en el repositorio de la asignatura a través del siguiente enlace: GitHub y se compone de los tres archivos siguientes:

- DataInfo_S013.csv: metadatos con la descripción de cada columna en el archivo de valores de datos.
- DataValues_S013.csv: valores clínicos y metabolómicos de 39 pacientes en cuatro timepoints.
- AAInformation_S006.csv: información adicional sobre los metabolitos.

3.2. Creación de un contenedor de tipo SummarizedExperiment

`SummarizedExperiment` permite organizar de manera estructurada y coherente datos experimentales, lo que facilita el análisis de datos ómicos complejos al incluir tanto los valores de los datos como de sus metadatos.

A continuación, se detallan los pasos seguidos para la creación del contenedor.

3.2.1. Instalación y carga de las bibliotecas y paquetes necesarios:

```
# Instalación y carga de BiocManager y SummarizedExperiment
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install(c("SummarizedExperiment", "ggplot2"), force = TRUE)

suppressMessages({
  library(SummarizedExperiment)
  library(readr)
  library(dplyr)
  library(ggplot2)
  library(tidyr)
  })
```

3.2.2. Carga del dataset y preparación de los datos:

Cargamos los archivos necesarios para crear el contenedor, incluyendo el archivo principal de datos, los metadatos y la información adicional de los metabolitos.

```
data_info <- read_csv("/Users/helenaortiz/Downloads/DataInfo_S013.csv")
```

```
## New names:
## Rows: 695 Columns: 4
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): ...1, VarName, varTpe, Description
## 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.
## * `` -> `...1`
```

```
data_values <- read_csv("/Users/helenaortiz/Downloads/DataValues_S013.csv")
```

```
## New names:
## Rows: 39 Columns: 696
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (2): SURGERY, GENDER dbl (693): ...1, SUBJECTS, AGE, Group, MEDDM_T0,
## MEDCOL_T0, MEDINF_T0, MEDHT... lgl (1): X
## 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.
## * `` -> `...1`
```

```
# Visualizamos las estructuras de los datasets cargados
head(data_info, 5)
```

```
## # A tibble: 5 x 4
##   ...1     VarName  varTpe    Description
##   <chr>    <chr>    <chr>     <chr>
## 1 SUBJECTS SUBJECTS integer   dataDesc
## 2 SURGERY  SURGERY  character dataDesc
## 3 AGE      AGE      integer   dataDesc
## 4 GENDER   GENDER   character dataDesc
## 5 Group    Group    integer   dataDesc
```

```
head(data_values[, 1:10], 5)
```

```
## # A tibble: 5 x 10
##    ...1 SUBJECTS SURGERY   AGE GENDER Group MEDDM_T0 MEDCOL_T0 MEDINF_T0
##   <dbl>    <dbl> <chr>   <dbl> <chr>  <dbl>    <dbl>     <dbl>     <dbl>
## 1     1        1 by pass    27 F          1        0         0         0
## 2     2        2 by pass    19 F          2        0         0         0
## 3     3        3 by pass    42 F          1        0         0         0
## 4     4        4 by pass    37 F          2        0         0         0
## 5     5        5 tubular    42 F          1        0         0         0
## # i 1 more variable: MEDHTA_T0 <dbl>
```

```r
# Preparamos los datos de expresión
datos_expresion <- as.matrix(data_values[, -1]) # Excluímos la primera columna
rownames(datos_expresion) <- data_values$SUBJECTS
head(datos_expresion[, 1:10], 5)
```

```
##   SUBJECTS SURGERY   AGE  GENDER Group MEDDM_T0 MEDCOL_T0 MEDINF_T0 MEDHTA_T0
## 1 " 1"     "by pass" "27" "F"    "1"   " 0"     " 0"      " 0"      " 1"
## 2 " 2"     "by pass" "19" "F"    "2"   " 0"     " 0"      " 0"      " 0"
## 3 " 3"     "by pass" "42" "F"    "1"   " 0"     " 0"      " 0"      " 0"
## 4 " 4"     "by pass" "37" "F"    "2"   " 0"     " 0"      " 0"      " 0"
## 5 " 5"     "tubular" "42" "F"    "1"   " 0"     " 0"      " 0"      " 0"
##   GLU_T0
## 1 " 85"
## 2 " 78"
## 3 " 75"
## 4 " 71"
## 5 " 82"
```

```r
# Conversión de los metadatos a data frame y establecemos nombres de fila
metadatos <- as.data.frame(data_info)
rownames(metadatos) <- metadatos$VarName
```

Verificamos que las dimensiones de `datos_expresion` y `metadatos` coincidan para despues proceder a crear el objeto `SummarizedExperiment`.

```r
# Verificar las dimensiones
cat("Dimensiones de datos_expresion:", dim(datos_expresion)[1], "\n")
```

```
## Dimensiones de datos_expresion: 39
```

```r
cat("Dimensiones de metadatos:", dim(metadatos)[1], "\n")
```

```
## Dimensiones de metadatos: 695
```

```r
# Verificar nombres de las filas
cat("Nombres de filas de datos_expresion:\n")
```

```
## Nombres de filas de datos_expresion:
```

```r
print(head(rownames(datos_expresion), 10))
```

```
##  [1] "1"  "2"  "3"  "4"  "5"  "6"  "7"  "8"  "9"  "10"
```

```r
cat("Nombres de filas de metadatos:\n")
```

```
## Nombres de filas de metadatos:
```

```r
print(head(rownames(metadatos), 10))
```

```
##  [1] "SUBJECTS"  "SURGERY"   "AGE"       "GENDER"    "Group"     "MEDDM_T0"
##  [7] "MEDCOL_T0" "MEDINF_T0" "MEDHTA_T0" "GLU_T0"
```

```r
# Comprobar si coinciden
if (!all(rownames(datos_expresion) %in% rownames(metadatos))) {
  cat("Algunos sujetos en datos_expresion no están en metadatos.\n")
}
```

```
## Algunos sujetos en datos_expresion no están en metadatos.
```

```r
dim(datos_expresion)
```

```
## [1]  39 695
```

```r
dim(metadatos)
```

```
## [1] 695   4
```

3.2.3. Creación del objeto SummarizedExperiment para crear el contenedor de datos:

Como ya hemos mencionado, los objetos `SummarizedExperiment` son contenedores de datos especializados para análisis ómicos en R. Estos permiten almacenar los datos de expresión (matriz de datos de metabolitos) y los metadatos de las muestras (información adicional, como la condición experimental).

```r
# Creamos el objeto SummarizedExperiment
se <- SummarizedExperiment(
  assays = list(counts = datos_expresion),
  colData = metadatos
)
se
```

```
## class: SummarizedExperiment
## dim: 39 695
## metadata(0):
## assays(1): counts
## rownames(39): 1 2 ... 38 39
## rowData names(0):
## colnames(695): SUBJECTS SURGERY ... SM.C24.0_T5 SM.C24.1_T5
## colData names(4): ...1 VarName varTpe Description
```

```r
# Guardamos el objeto en formato .Rda
save(se, file = "Summarized_Experiment.Rda")
```

Siguiendo estos pasos habríamos creado de manera correcta el contenedor y lo tendríamos guardado en el directorio asignado en formato .Rda .

4. RESULTADOS

4.1. EXPLORACIÓN DE LOS DATOS

Para asegurar la integridad de los datos cargados en el objeto `SummarizedExperiment`, revisamos un resumen estadístico de `se` y de los `metadatos`.

```
# Verificamos que los datos contenidos en SummarizedExperiment son correctos
summary(se)
```

```
## [1] "SummarizedExperiment object of length 39 with 0 metadata columns"
```

```
summary(metadatos)
```

```
##      ...1             VarName             varTpe          Description
##  Length:695         Length:695         Length:695         Length:695
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
rowData(se)
```

```
## DataFrame with 39 rows and 0 columns
```

```
colData(se[, 1:10], 5)
```

```
## DataFrame with 10 rows and 4 columns
##                    ...1     VarName      varTpe Description
##             <character> <character> <character> <character>
## SUBJECTS      SUBJECTS    SUBJECTS     integer    dataDesc
## SURGERY       SURGERY     SURGERY   character    dataDesc
## AGE               AGE         AGE     integer    dataDesc
## GENDER         GENDER      GENDER   character    dataDesc
## Group           Group       Group     integer    dataDesc
## MEDDM_T0     MEDDM_T0    MEDDM_T0     integer    dataDesc
## MEDCOL_T0   MEDCOL_T0   MEDCOL_T0     integer    dataDesc
## MEDINF_T0   MEDINF_T0   MEDINF_T0     integer    dataDesc
## MEDHTA_T0   MEDHTA_T0   MEDHTA_T0     integer    dataDesc
## GLU_T0         GLU_T0      GLU_T0     integer    dataDesc
```

4.1.1. Resumen estadístico inicial

Procedemos a realizar un resumen estadístico de las principales variables numéricas y categóricas del dataset (incluyendo la media, la mediana y los rangos intercuartílicos), para así comprobar que los datos se han cargado correctamente.

```
# Mostramos el resumen de las primeras 5 columnas de datos_expresión
# (para reducir el ruido)
apply(as.matrix(datos_expresion[, 1:5]), 2, summary)
```

```
##        SUBJECTS    SURGERY     AGE         GENDER      Group
## Length "39"        "39"        "39"        "39"        "39"
## Class  "character" "character" "character" "character" "character"
## Mode   "character" "character" "character" "character" "character"
```

```
# Realizamos el resumen estadístico de algunas variables
summary(data_values$AGE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   19.00   35.00   41.00   40.79   46.00   59.00
```

```
summary(data_values$bmi_T0)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   29.80   44.40   48.80   50.52   55.35   68.60
```

```
summary(data_values$bmi_T2)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   26.80   40.20   44.60   45.09   49.00   60.70       2
```

```
summary(data_values$bmi_T4)
```

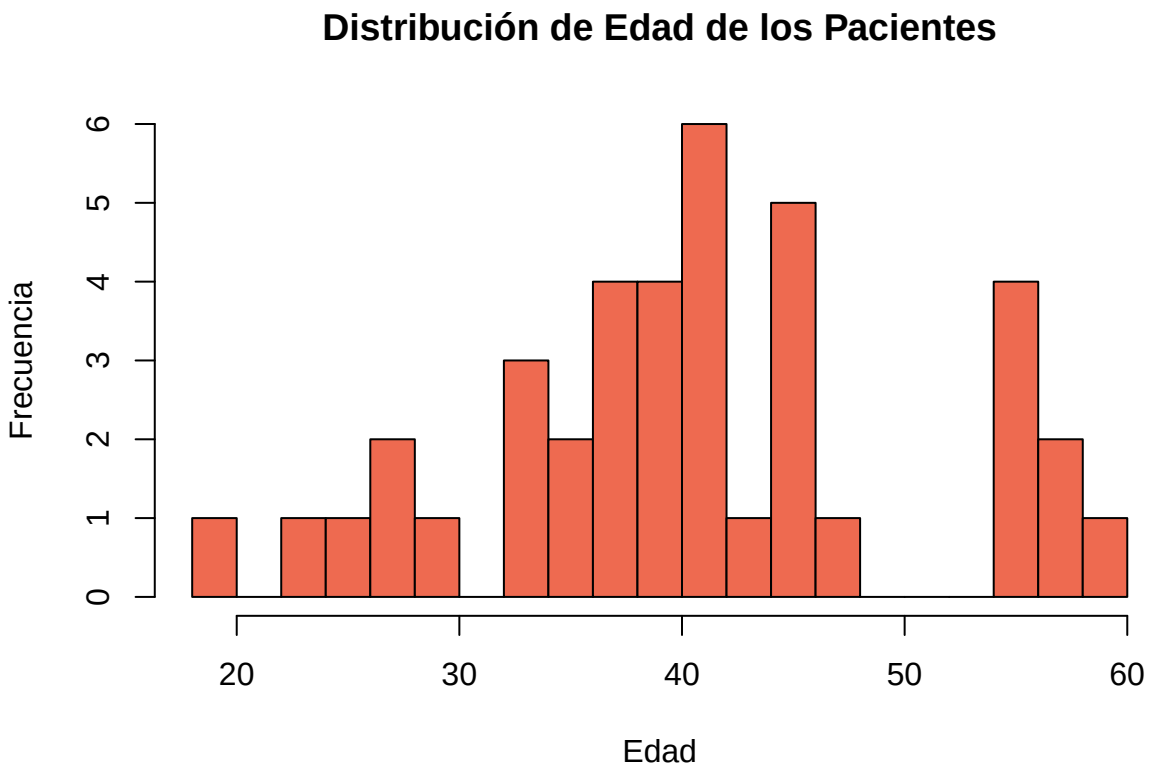
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   26.40   36.80   40.00   41.28   44.90   57.40       1
```

```
summary(data_values$bmi_T5)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   25.50   32.55   35.53   36.42   40.58   50.90       7
```

4.1.2. Distribución de las variables AGE y bmi

```
# Distribución de la edad
edad <- as.numeric(as.character(data_values$AGE))
hist(edad, breaks = 20, main = "Distribución de Edad de los Pacientes", xlab = "Edad", ylab = "Frecuenci
```

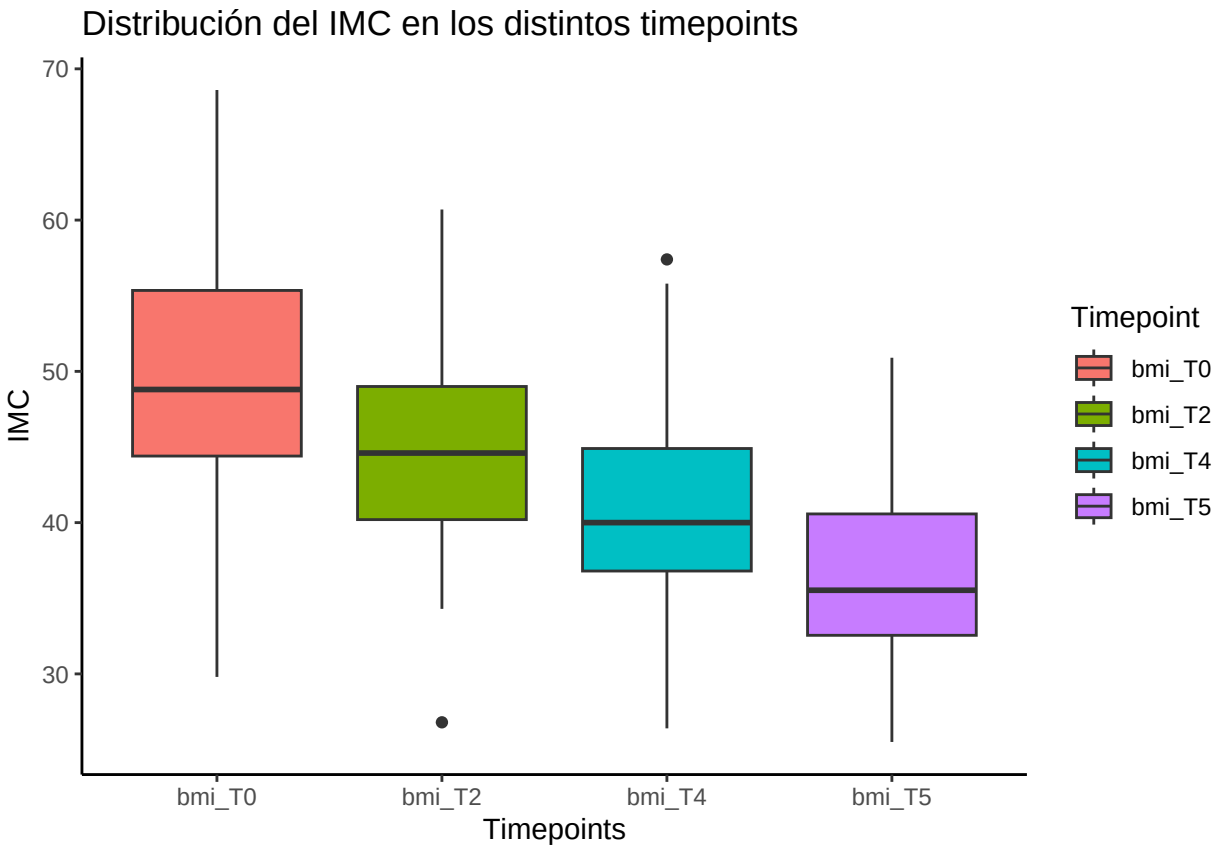


La visualización básica de la variable de AGE revela la distribución de este factor en la población de estudio. Aunque simple, esta exploración preliminar es clave para comprender las características de la muestra y proporciona una primera aproximación a los datos.

```
bmi_all <- data_values %>%
  pivot_longer(cols = starts_with("bmi_T"),
               names_to ="Timepoint",
               values_to = "BMI")

# Representación gráfica en un boxplot:
ggplot(bmi_all, aes(x = Timepoint, y = BMI, fill = Timepoint)) +
    geom_boxplot() +
    labs(title = "Distribución del IMC en los distintos timepoints",
         x = "Timepoints",
         y = "IMC") +
    theme_classic()
```

```
## Warning: Removed 10 rows containing non-finite outside the scale range
## (‘stat_boxplot()‘).
```



Con la visualización de las variables `bmi_T0`, `bmi_T2`, `bmi_T4`y `bmi_T5` podemos observar cómo disminuye el índice de masa corporal de los individuos de la muestra en los distintos timepoints donde se mide.

```
write.table(data_values, file = "Datos.txt", sep = "\t", row.names = FALSE)
```

4.1.3. Guardar datos en formato .txt:

```
writeLines(c(
  "# Metadatos del Análisis de Datos Ómicos",
  "Este archivo contiene información relevante sobre el dataset utilizado en el análisis.",

  "## Descripción de los archivos",
  "- **DataValues_S013.txt**: Contiene valores clínicos y metabolómicos de 39 pacientes en cuatro momen
  "- **DataInfo_S013.csv**: Contiene descripciones de cada columna del archivo de datos.",
  "- **AAInformation_S006.csv**: Información adicional sobre los metabolitos presentes en el dataset.",

  "## Variables importantes",
  "- **AGE**: Edad del paciente en años.",
  "- **BMI_T0, BMI_T2, BMI_T4, BMI_T5**: Índice de masa corporal en diferentes momentos.",
  "- **SUBJECTS**: Identificador de cada paciente.",
```

```
  "## Enlaces útiles",
  "Para más información, visita [SummarizedExperiment en Bioconductor](https://bioconductor.org/package
), con = "metadatos.md")
```

4.1.4. Guardar metadatos en formato .md:

---

5. CONCLUSIONES

El análisis exploratorio de este dataset ha permitido organizar y revisar los datos clínicos y metabolómicos de los pacientes en un formato estructurado usando SummarizedExperiment, un contenedor de datos específico para análisis ómicos en R. Esta estructuración facilita un procesamiento coherente de los datos y garantiza que la información esté correctamente organizada para futuras evaluaciones.

En futuros análisis, se podrán implementar técnicas avanzadas, como el análisis de componentes principales (PCA) y otros métodos multivariantes, para identificar patrones en los perfiles metabólicos. Estos patrones podrían ofrecer una visión sobre cómo los cambios en el metabolismo tras la cirugía bariátrica reflejan la respuesta del organismo a nivel molecular y clínico, independientemente de la pérdida de peso lograda.

Este análisis preliminar, por lo tanto, constituye un paso inicial sólido para profundizar en la identificación de "metabotipos" y abre la posibilidad de desarrollar intervenciones terapéuticas personalizadas basadas en el perfil metabólico de los pacientes, optimizando los resultados de la cirugía bariátrica en función de las características metabólicas individuales.

---

6. REPOSITORIO DE GITHUB

El repositorio con todos los archivos de este análisis, incluyendo el informe, el objeto `SummarizedExperiment`, el código en R y los archivos de datos y metadatos se encuentran en el siguiente enlace:

Repositorio GitHub

https://github.com/helenaortizz/Ortiz-Rivero-Helena-PEC1

7. BIBLIOGRAFÍA

Además de los materiales proporcionados por la asignatura, se ha hecho uso de los siguientes recursos:

- Bioconductor. (2023). SummarizedExperiment: SummarizedExperiment container (Version 1.30.1) [Software]. Disponible en https://bioconductor.org/packages/release/bioc/html/SummarizedExperiment.html
- Huber, W., et al. (2015). Orchestrating high-throughput genomic analysis with Bioconductor. Nature Methods, 12(2), 115–121. doi:10.1038/nmeth.3252
- Arora, T., & Sharma, R. (2019). Metabolic effects of bariatric surgery: clinical insights. Experimental and Clinical Endocrinology & Diabetes, 127(3), 171–180. doi:10.1055/s-0043-120927
- Rao, S. R. (2016). In search of the optimal diet after bariatric surgery: a systematic review. Surgery for Obesity and Related Diseases, 12(5), 929–935. doi:10.1016/j.soard.2016.01.019

- Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York. doi: 10.1007/978-3-319-24277-4
- Biotecnika. (2020). Understanding Summarized Experiment - Biological data analysis using R Bioconductor [Video]. YouTube. https://www.youtube.com/watch?v=UsOBxRZH8j4
