Ibáñez-Gómez-Adriana-PEC1

Adriana Ibáñez

2025-03-28

Taula de continguts

Abstract

Aquest estudi té com a objectiu principal la creació i anàlisi d'un objecte de classe SummarizedExperiment a partir d'un dataset de metabolòmica extret del projecte MetabotypingPaper. Per assolir aquest objectiu, s'han integrat diverses fonts de dades que inclouen mesures de metabòlits en diferents punts temporals, metadades dels subjectes i informació de les variables mesurades. A través d'una anàlisi exploratòria, s'han examinat patrons temporals en diversos paràmetres com el pes, l'índex de massa corporal (BMI), la glucosa, la insulina i el perfil lipídic. A més, s'ha aplicat una anàlisi de clustering per identificar grups de subjectes amb patrons diferenciats en l'evolució dels nivells de glucosa. Els resultats mostren una millora generalitzada en els perfils metabòlics posteriors a la cirurgia bariàtrica, tot i que amb una variabilitat considerable entre subjectes. Aquesta anàlisi destaca la utilitat de la classe SummarizedExperiment per gestionar dades biomèdiques complexes i proporciona evidència sobre l'efectivitat de la cirurgia en la millora dels perfils metabòlics.

Objectius

Els objectius d'aquesta PAC es poden definir com a: - Crear un objecte SummarizedExperiment amb totes les seves parts generant una base de dades completa amb eficiència de dades, amb la informació de metabolomics - Analitzar les dades ecollides amb l'estructura de classe SummarizedExperiment

Mètodes

Per dur a terme l'anàlisi, s'han utilitzat diversos paquets de la plataforma Bioconductor en R, entre els quals destaca SummarizedExperiment, dissenyat per gestionar dades experimentals d'alta dimensionalitat. El dataset seleccionat (MetabotypingPaper) inclou mesures de metabòlits en diversos punts temporals (T0, T2, T4 i T5), així com informació clínica dels subjectes (edat, sexe, cirurgia i grup d'estudi). Els fitxers de dades s'han carregat utilitzant les funcions read.csv, i posteriorment s'ha estructurat la informació mitjançant la creació d'un objecte de classe SummarizedExperiment, amb les matrius d'assajos, metadades de les mostres (colData) i dels metabòlits (rowData).

S'ha dut a terme una inspecció de valors mancants i una reorganització de les dades per facilitar-ne l'anàlisi. Posteriorment, s'ha realitzat una anàlisi exploratòria mitjançant visualitzacions de boxplots i línies de tendència, enfocant-se en l'evolució de paràmetres com el pes, BMI, glucosa, insulina i colesterol (LDL, HDL, VLDL). A més, s'ha implementat un mètode de clustering (k-means) per agrupar els subjectes segons patrons de canvi en la glucosa al llarg del temps, permetent identificar perfils metabòlics diferenciats. Tot el procés s'ha documentat i automatitzat en un repositori públic de GitHub.

Resultats

Elecció del dataset

L'elecció del dataset MetabotypingPaper de metabolòmica (metaboData-main) ha sigut en base a visualitzar i observar els diferents datasets: MetabotypingPaper, Phosphoproteomics, CIMCBTutorial, UGrX-4MetaboAnalystTutorial, Cachexia, fobitools-UnseCase_1. Després de fer un cop d'ull als diferents datasets, el de MetabotypingPaper és el que més em va agradar degut a que té en compte diferents paràmetres bimoleculars de cada subjecte. Aquest tipus de base de dades és similar al que estic treballant amb col · laboració amb un grup de recerca de la Universitat de Barcelona, on analitzem dades del projecte "Rancho Bernardo", on tenen moltes dades de diferents individuus, desde dades biomoleculars extretes de proves com analítiques, imatges fetes amb DEXA, mesures biomètriques etc. Un data set amb paràmetres moleculars que es poden caracteritzar com a numèrics es un bon exemple per analitzar i entrenar-me per després treballar amb dades amplies com les del Rancho Bernardo.

Preparació i càrrega dels paquets i llibreries

Després d'instalar els paquets necessaris amb la funció install.packages, carreguem les diferents llibreries:

```
library(BiocManager)
```

```
## Warning: package 'BiocManager' was built under R version 4.4.2
```

```
library(SummarizedExperiment)
```

```
## Cargando paquete requerido: MatrixGenerics
```

```
## Warning: package 'MatrixGenerics' was built under R version 4.4.2
```

```
## Cargando paquete requerido: matrixStats
```

```
## Warning: package 'matrixStats' was built under R version 4.4.2
```

```
##
## Adjuntando el paquete: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
```

```
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Cargando paquete requerido: GenomicRanges

## Cargando paquete requerido: stats4

## Cargando paquete requerido: BiocGenerics

##
## Adjuntando el paquete: 'BiocGenerics'

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, saveRDS, setdiff,
##     table, tapply, union, unique, unsplit, which.max, which.min

## Cargando paquete requerido: S4Vectors

##
## Adjuntando el paquete: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Cargando paquete requerido: IRanges

## Warning: package 'IRanges' was built under R version 4.4.2

##
## Adjuntando el paquete: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Cargando paquete requerido: GenomeInfoDb

## Warning: package 'GenomeInfoDb' was built under R version 4.4.2
```

```
## Cargando paquete requerido: Biobase
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
##
## Adjuntando el paquete: 'Biobase'
## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians
## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```r
library(readr)
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
## The following object is masked from 'package:Biobase':
##
##     combine
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
## The following object is masked from 'package:matrixStats':
##
##     count
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```r
library(tibble)
```

Carregar un dataset de metabolòmica

Carreguem el dataset escollit, en aquest cas MetabotypingPaper. En aquest cas tenim dos arxius excel, un que conté les dades raw i l'altre, que conté la informació de les dades.

```r
# Carregar els fitxers CSV que tenim a la mateixa carpeta que aquest document
data <- read.csv("DataValues_S013.csv", row.names = 1, check.names = FALSE)
data_info <- read.csv("DataInfo_S013.csv", row.names= 1)
data$X <- NULL
```

Crear un objecte de classe SumarizedExperiments

Arribats a aquest punt, necessitem afegir la informació que ens proporciona el document AAinformation_S006, com a metadada, ja que es una llista que conté informació necessària dels diferents valors / mesures que conté aquest dataset.

```r
Information <- read.csv("AAInformation.txt", row.names = 1)
Information$X <- NULL
```

Un cop tenim tota la informació que ens proporcionen afegida, observem que tenim 5 característiques apart del ID que son característiques dels subjectes i no valors de metabolits, com la resta. Per aprofitar les facilitats de la classe SummarizedExperiment, es vol reorganitzar les dades per que siguin el màxim eficient. Per aixo mateix, afegim com a col_data aquestes característiques propies dels subjectes.

```r
col_data <- data[, c("SURGERY", "AGE", "GENDER", "Group")]
assay_data <- t(data[, -c(1:5)])
```

```r
# Creem el objecte de classe
se <- SummarizedExperiment(
  assays = list(counts = assay_data),
  colData = col_data
)
```

Ara mateix tenim el objecte creat i amb les dades característiques de cada subjecte, hem fet la col_data. Per que tingui més sentit, s'inverteix el dataset per obtenir 39 columnes (subject id) i que en les files trobem els diferents valors dels metabòlits. Així aconseguim separar les dades en diferents taules que es poden superposar per trobar combinacions concretes.

Ara ens dediquem a afegir diferentes dades com el nom de les files:

```r
row_names <- rownames(se)
```

Ara, creem una nova columna que es digui timepoint amb les dades de T (T0-5) i s'ha esborrat la informació a la columna de metabolite del temps amb el codi:

```r
row_meta <- data.frame(
  feature_id = row_names,
  metabolite = gsub("_(T[0-5]+)$", "", row_names),
  timepoint = gsub(".*_(T[0-5]+)$", "\\1", row_names),
  stringsAsFactors = FALSE
)
```

Ara afegim la taula Informació, amb la informació dels diferents aminoacids, amb el codi:

```r
row_meta_full <- merge(
  row_meta,
```

```
  Information,
  by.x = "metabolite",
  by.y = "Metabolite.abbreviation",
  all.x = TRUE
)
```

Ara afegim aquesta informació com a rowData al se, amb el codi:

```
rownames(row_meta_full) <- row_meta_full$feature_id
row_meta_full <- row_meta_full[rownames(se), ]
rowData(se) <- row_meta_full
```

Ara afegim el fitxer de data_info com a metadada del se, amb el codi:

```
metadata(se)$data_info <- data_info
```

Diferències SummarizedExperiment amb ExpressionSet

SummarizedExperiment i ExpressionSet son classes diferents. SummarizedExperiment és una classe usada per emmagatzemar matrius rectangulars de resultats experimentals que permet gestionat múltiples matrius de dades (assays) i qualsevol tipus de dades, com més complexes com les obtingudes se seqüenciació massiva, amb un model més estructurat. El maneig de les dades en aquesta classe s'emmagatzemen en un assays object, que permet treballar amb múltiples matrius de dades.

Per altra banda, ExpressionSet és una classe que combina múltiples fonts d'informació en una estructura convenient per representar normalment dades genòmiques, que está pensat per manipular dades provinents de dades d'expressió (assayData) com microarrays, metadades de mostres (phenoData) i dades de característiques (featureData). Es basa en objectes S4 i en emmagatzemar dades d'expressió gènica amb anotacions sobre mostres i característiques. El maneig de les dades s'una amb l'objecte assayData per contenir matrius d'expressió i altres tipus de dades associades.

Per aquestes característiques i degut a que ExpressionSet es va dissenyar inicialment per a dades de microarrays, mentre SummarizedExperiment és mes adequat per a dades de RNAseq i altres tecnologies d'alt rendiment, aquestes classes han tingut diferent suport i evolució. ExpressionSet encara es fa servir però el seu ús ha disminuït amb el temps i SummarizedExperiment és l'estàndard més recent en Bioconductor i s'està expandint per donar suport a nous tipus de dades.

Anàlisi exploratòria

Primer volem veure quantes variables hi ha, independentment del temps en el que ha sigut mesurat, per això:

```
se
```

```
## class: SummarizedExperiment
## dim: 689 39
## metadata(1): data_info
## assays(1): counts
## rownames(689): MEDDM_T0 MEDCOL_T0 ... SM.C24.0_T5 SM.C24.1_T5
## rowData names(7): metabolite feature_id ... Platform Data.type
## colnames(39): 1 2 ... 38 39
## colData names(4): SURGERY AGE GENDER Group
```

```
table_timepoints <- table(rowData(se)$timepoint)
table_timepoints
```

```
##
##  T0  T2  T4  T5
## 172 173 172 172
```

Veiem que es mesuren 172 variables (menys en el T3 que es va mesurar una variable diferent).

3. Anàlisi exploratòria

Primer volem veure quantes variables hi ha, independentment del temps en el que ha sigut mesurat, per això:

```
se
```

```
## class: SummarizedExperiment
## dim: 689 39
## metadata(1): data_info
## assays(1): counts
## rownames(689): MEDDM_T0 MEDCOL_T0 ... SM.C24.0_T5 SM.C24.1_T5
## rowData names(7): metabolite feature_id ... Platform Data.type
## colnames(39): 1 2 ... 38 39
## colData names(4): SURGERY AGE GENDER Group
```

```
table_timepoints <- table(rowData(se)$timepoint)
table_timepoints
```

```
##
##  T0  T2  T4  T5
## 172 173 172 172
```

Veiem que es mesuren 172 variables (menys en el T3 que es va mesurar una variable diferent).

Ara es vol saber si existeixen valors NA.

```
NA_values <- function(se) {
  cat("Assay matrix:\n")
  cat("  NA: ", sum(is.na(assay(se))), "\n")
  cat("  NaN:", sum(is.nan(assay(se))), "\n")
  cat("  Inf:", sum(is.infinite(assay(se))), "\n")
  cat("  % of missing or inifinite values:", (sum(is.na(assay(se)))+sum(is.nan(assay(se)))+sum(is.infin
  cat("\nRowData NA count:", sum(is.na(as.data.frame(rowData(se)))), "\n")
  cat("ColData NA count:", sum(is.na(as.data.frame(colData(se)))), "\n")
}


NA_values(se)
```

```
## Assay matrix:
##   NA:  3351
##   NaN: 0
##   Inf: 0
##   % of missing or inifinite values: 12.47069
##
## RowData NA count: 2244
## ColData NA count: 0
```

El assay conté 3351 valors NA, el que representen un 12% de totes les dades del dataset, un valor bastant representatiu. També es pot extreure que tenim 2244 valors Na en el RowData però no es tant preocupant, sabent que moltes dades no tenien assignat un valor de "Information" ja que no eren metabòlits, per exemple.

En quant als metabolits, veient que aquest estudi es basa en prendre mesures a diferents punts després de la cirurgia bariàtrica, em vull centrar en alguns valors.

Inicialment vull veure una grafica amb la perdua de pes dels subjectes al llarg dels diferents punts. També el bmi, que relaciona el pes amb l'alçada.

Per això hem de generar un dataframe que separi els feature id, subjects, timepoints i valors. Després, crear dataframes separats per cada cosa que es vol graficar.

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
##
## Adjuntando el paquete: 'tidyr'
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand
```

```
df_subject_feature_value_timepoint <- assay(se) %>%
  as.data.frame() %>%
  rownames_to_column("feature_id") %>%
  pivot_longer(-feature_id, names_to = "Subject", values_to = "Value") %>%
  left_join(
    rowData(se) %>%
      as.data.frame() %>%
      select(timepoint, feature_id),  # No volvemos a usar rownames_to_column aquí
    by = "feature_id"
  )
```

Primerament creem el dataframe df_PESOBMI amb una columna amb el feature "PESO" i "bmi", i els diferents valors al llarg dels punts del temps.

```
df_PESOBMI <- df_subject_feature_value_timepoint %>%
  # Creamos una columna nueva que contiene solo el nombre del feature sin el sufijo _TX
  mutate(feature_clean = gsub("_[T].*$", "", feature_id)) %>%
  # Filtrar para quedarnos solo con Glu (glucosa) e ins (insulina)
  filter(feature_clean %in% c("PESO", "bmi"))

# Visualizamos el resultado
head(df_PESOBMI)
```

```
## # A tibble: 6 x 5
##   feature_id Subject Value timepoint feature_clean
##   <chr>      <chr>   <dbl> <chr>     <chr>
## 1 PESO_T0    1         151 T0        PESO
## 2 PESO_T0    2         139 T0        PESO
## 3 PESO_T0    3          84 T0        PESO
## 4 PESO_T0    4         136 T0        PESO
## 5 PESO_T0    5         121 T0        PESO
## 6 PESO_T0    6         148 T0        PESO
```

Amb el dataframe creat, generem la gràfica.

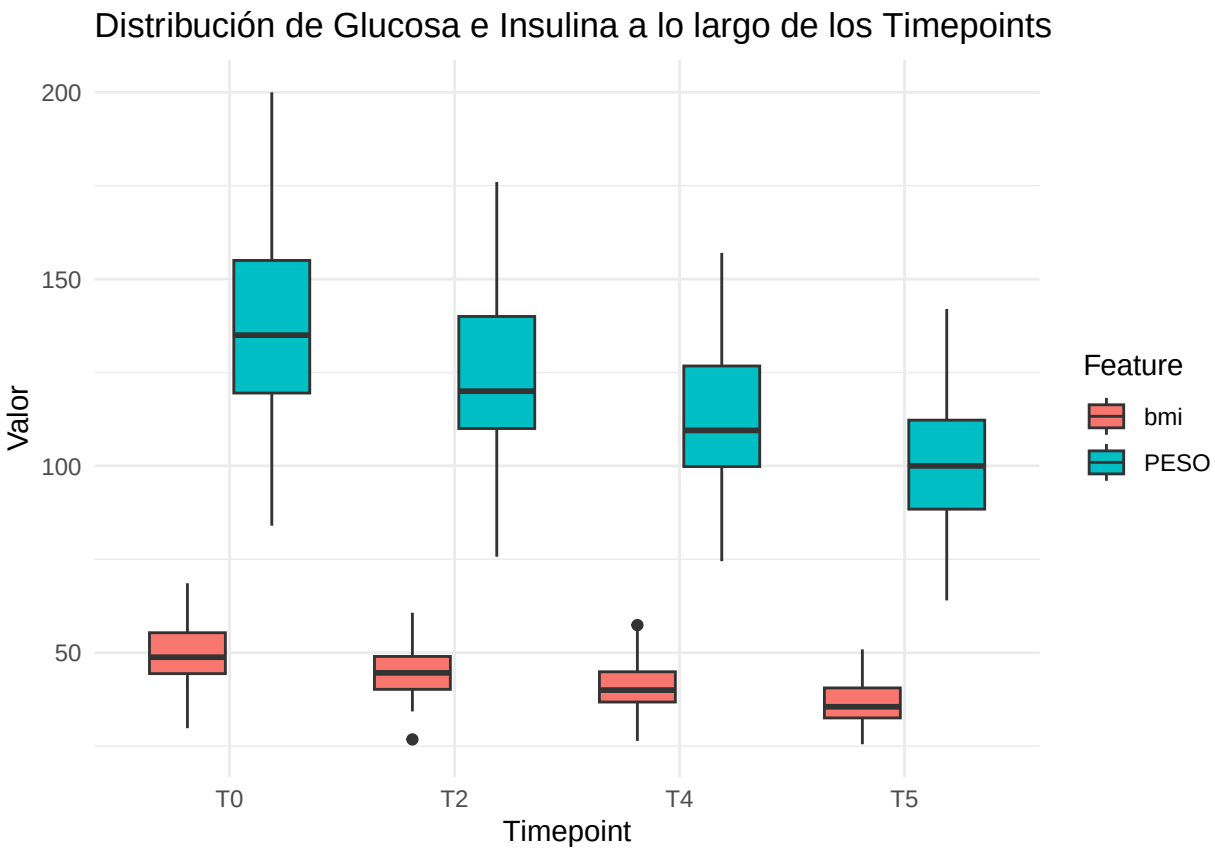
```
library(ggplot2)
library(dplyr)

# Convertimos timepoint en factor para que se mantenga el orden deseado (ajústalo según tus timepoints)
df_modificado <- df_PESOBMI %>%
  mutate(timepoint = factor(timepoint, levels = c("T0", "T2", "T4", "T5")))

ggplot(df_modificado, aes(x = timepoint, y = Value, fill = feature_clean)) +
  geom_boxplot() +
```

```
  labs(title = "Distribución de Glucosa e Insulina a lo largo de los Timepoints",
       x = "Timepoint",
       y = "Valor",
       fill = "Feature") +
  theme_minimal()
```

```
## Warning: Removed 20 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



S'observa un descens del pes a mesura que avançen els timepoints. També un descens en el bmi, relacionat amb el canvi de pes, ja que l'alçada dels individuus no hauria de canviar supoant que son adults.

A continuació vull observar els canvis del parámetre de glucosa i insulina al llarg del temps. Per tant, he de crear un altre dataframe, anomenat en aquest cas df_GLUINS.

```
df_GLUINS <- df_subject_feature_value_timepoint %>%
  # Creamos una columna nueva que contiene solo el nombre del feature sin el sufijo _TX
  mutate(feature_clean = gsub("_[T].*$", "", feature_id)) %>%
  # Filtrar para quedarnos solo con Glu (glucosa) e ins (insulina)
  filter(feature_clean %in% c("Glu", "INS"))

# Visualizamos el resultado
head(df_GLUINS)
```

```
## # A tibble: 6 x 5
##   feature_id Subject Value timepoint feature_clean
##   <chr>      <chr>   <dbl> <chr>     <chr>
## 1 INS_T0     1        11.4  T0        INS
```

```
## 2 INS_T0     2       12.1  T0        INS
## 3 INS_T0     3        8.41 T0        INS
## 4 INS_T0     4       12.8  T0        INS
## 5 INS_T0     5        6.01 T0        INS
## 6 INS_T0     6        9.88 T0        INS
```

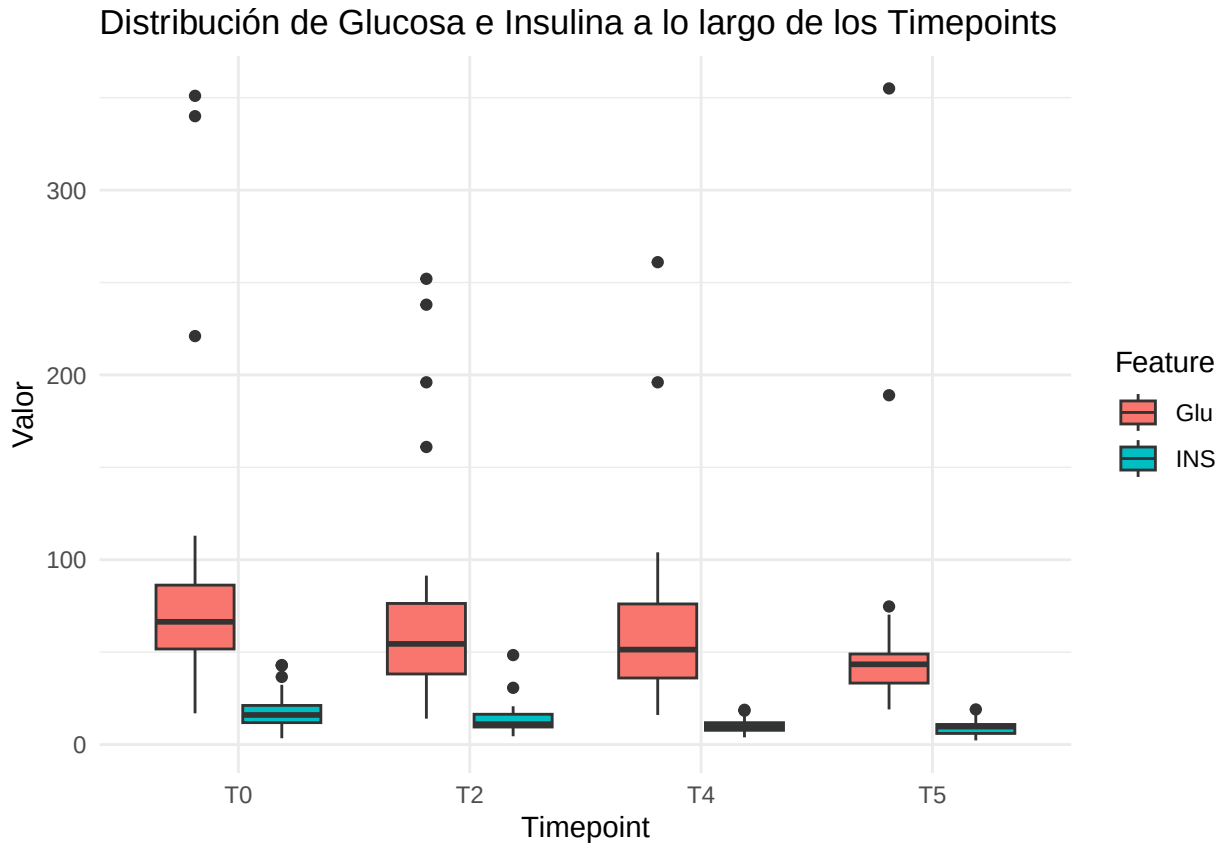
Ara representem el dataframe:

```r
library(ggplot2)
library(dplyr)

# Convertimos timepoint en factor para que se mantenga el orden deseado (ajústalo según tus timepoints)
df_modificado <- df_GLUINS %>%
  mutate(timepoint = factor(timepoint, levels = c("T0", "T2", "T4", "T5")))

ggplot(df_modificado, aes(x = timepoint, y = Value, fill = feature_clean)) +
  geom_boxplot() +
  labs(title = "Distribución de Glucosa e Insulina a lo largo de los Timepoints",
       x = "Timepoint",
       y = "Valor",
       fill = "Feature") +
  theme_minimal()
```

```
## Warning: Removed 32 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



s pot observar un descens en els nivells de glucosa al llarg del temps, i en consequencia un descens també en els nivells de insulina. Tot i així es genera un altre tipus de gràfic que permeti un altre tipus de visualització.

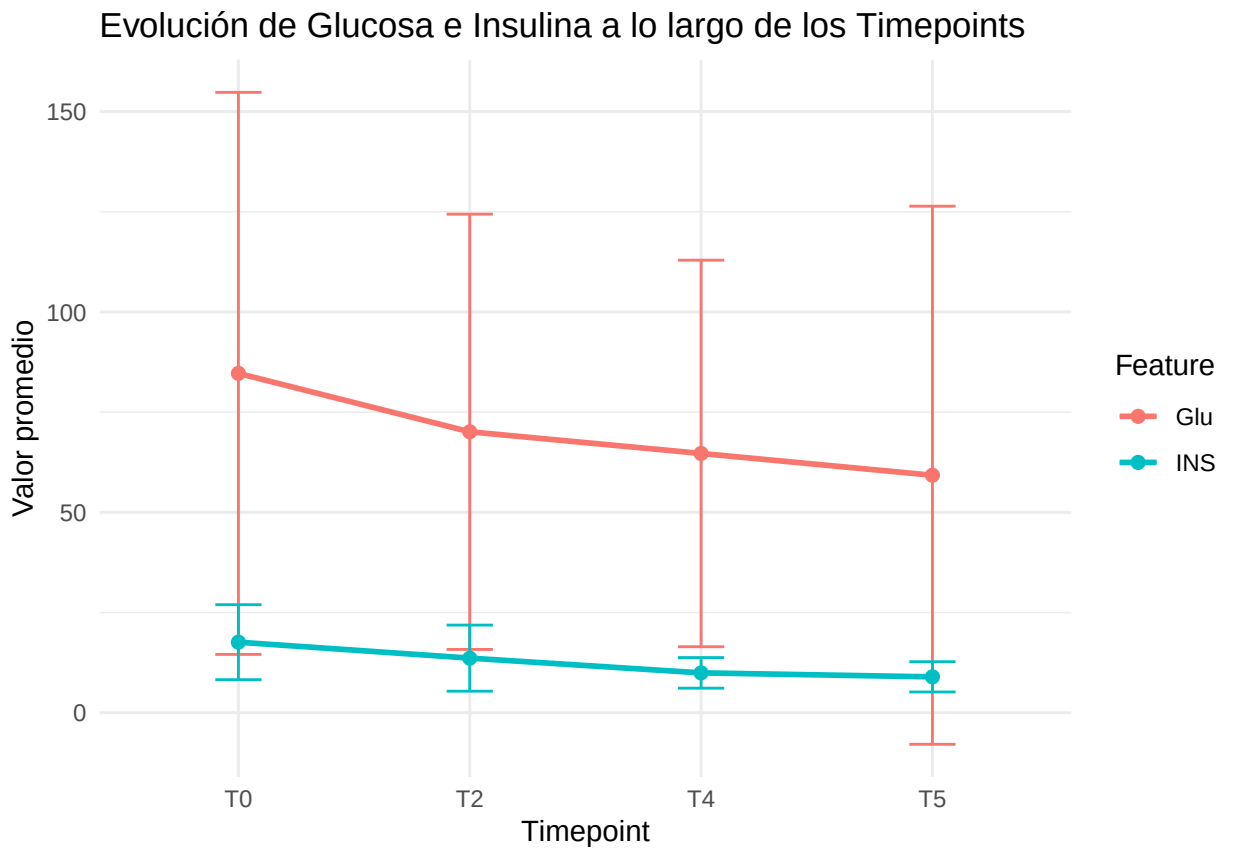
```r
library(ggplot2)
library(dplyr)

# Calcular estadísticas resumen para cada timepoint y feature (Glu e ins)
df_summary <- df_GLUINS %>%
  group_by(timepoint, feature_clean) %>%
  summarise(mean_value = mean(Value, na.rm = TRUE),
            sd_value = sd(Value, na.rm = TRUE),
            n = n(),
            .groups = "drop")

# Convertir timepoint a factor para asegurar el orden correcto
df_summary <- df_summary %>%
  mutate(timepoint = factor(timepoint, levels = c("T0", "T2", "T4", "T5")))

# Graficar la evolución de la media con barras de error (media ± SD)
ggplot(df_summary, aes(x = timepoint, y = mean_value, group = feature_clean, color = feature_clean)) +
  geom_line(size = 1) +
  geom_point(size = 2) +
  geom_errorbar(aes(ymin = mean_value - sd_value, ymax = mean_value + sd_value),
                width = 0.2) +
  labs(title = "Evolución de Glucosa e Insulina a lo largo de los Timepoints",
       x = "Timepoint",
       y = "Valor promedio",
       color = "Feature") +
  theme_minimal()
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



La tendència es la mateixa, pero s'observa molt error, sobretot en el paràmetre de la glucosa, de manera que busquem agrupar per clústers els diferents grups d'individuus, intentant veure si veiem diferents metabolismes.

```
library(tidyr)
library(dplyr)

# Filtrar para la feature "Glu" y ordenar por subject y timepoint.
df_glu <- df_GLUINS %>%
  filter(feature_clean == "Glu") %>%
  arrange(Subject, timepoint) %>%
  group_by(Subject) %>%
  summarise(trend = list(Value)) %>%
  ungroup()

# Convertir la lista de tendencias en una matriz
trend_matrix <- do.call(rbind, df_glu$trend)
rownames(trend_matrix) <- df_glu$Subject

# Revisa la matriz para asegurarte que cada fila contiene los mismos timepoints (en el mismo orden)
```

He d'eliminar files que tinguin algun valor faltant en la trajectoria, per que la gràfica estigui més neta.

```
# Convertir la lista de tendencias en una matriz
trend_matrix <- do.call(rbind, df_glu$trend)
rownames(trend_matrix) <- df_glu$Subject

# Eliminar filas con algún NA
```

```r
trend_matrix <- trend_matrix[complete.cases(trend_matrix), ]
```

```r
subjects_valid <- rownames(trend_matrix)
df_glu <- df_glu %>% filter(Subject %in% subjects_valid)
```

```r
set.seed(123)  # Para reproducibilidad
clusters <- kmeans(trend_matrix, centers = 3)$cluster
df_glu <- df_glu %>%
  mutate(cluster = as.factor(clusters))
```

```r
set.seed(123)
clusters <- kmeans(trend_matrix, centers = 3)  # ← esto es lo que necesitas

clusters$centers
```

```
##        [,1]      [,2]      [,3]      [,4]
## 1  46.99091  35.92727  40.76364  33.53636
## 2  77.45000  62.00833  70.25833  47.89167
## 3 286.00000 224.00000 228.50000 272.00000
```

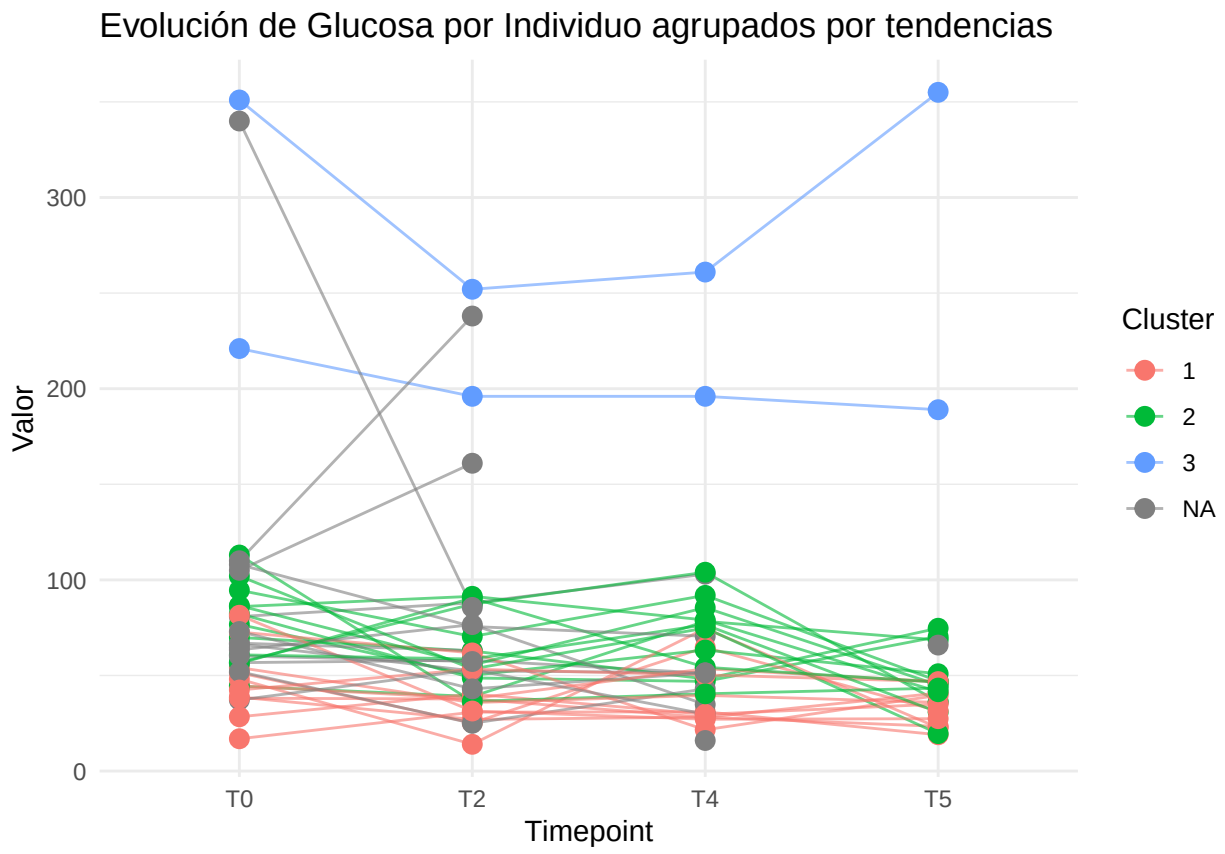
Aquests son els clústers que ha realitzat R.

```r
df_glu_clusters <- df_glu %>% select(Subject, cluster)
df_modificado_glu <- df_modificado %>%
  filter(feature_clean == "Glu") %>%
  left_join(df_glu_clusters, by = "Subject")
```

```r
ggplot(df_modificado_glu, aes(x = timepoint, y = Value, group = Subject, color = cluster)) +
  geom_line(alpha = 0.6) +
  geom_point(size = 3) +
  labs(title = "Evolución de Glucosa por Individuo agrupados por tendencias",
       x = "Timepoint", y = "Valor",
       color = "Cluster") +
  theme_minimal()
```

```
## Warning: Removed 15 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

```
## Warning: Removed 18 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
print(df_modificado_glu)
```

```
## # A tibble: 156 x 6
##    feature_id Subject Value timepoint feature_clean cluster
##    <chr>      <chr>   <dbl> <fct>     <chr>         <fct>
##  1 Glu_T0     1        38.7 T0        Glu           1
##  2 Glu_T0     2        66.9 T0        Glu           <NA>
##  3 Glu_T0     3        16.9 T0        Glu           1
##  4 Glu_T0     4        44.6 T0        Glu           1
##  5 Glu_T0     5        54.1 T0        Glu           1
##  6 Glu_T0     6        60.8 T0        Glu           2
##  7 Glu_T0     7        37.2 T0        Glu           <NA>
##  8 Glu_T0     8        56.3 T0        Glu           2
##  9 Glu_T0     9       108   T0        Glu           <NA>
## 10 Glu_T0     10       80.8 T0        Glu           <NA>
## # i 146 more rows
```

Com es pot veure, tenim 3 clústers diferents en quant a l'evolució de la glucosa en el temps.

```
df_summary <- df_modificado_glu %>%
  group_by(cluster, timepoint) %>%
  summarise(media = mean(Value, na.rm = TRUE),
            sd = sd(Value, na.rm = TRUE),
            n = n(),
            .groups = "drop")

print(df_summary)
```

```
## # A tibble: 16 x 5
##    cluster timepoint media     sd     n
##    <fct>   <fct>     <dbl>  <dbl> <int>
##  1 1       T0         47.0  18.3     11
##  2 1       T2         35.9  13.1     11
##  3 1       T4         40.8  17.5     11
##  4 1       T5         33.5   9.60    11
##  5 2       T0         77.4  20.5     12
##  6 2       T2         62.0  19.1     12
##  7 2       T4         70.3  19.7     12
##  8 2       T5         47.9  16.4     12
##  9 3       T0        286    91.9      2
## 10 3       T2        224    39.6      2
## 11 3       T4        228.   46.0      2
## 12 3       T5        272   117.       2
## 13 <NA>    T0         91.7  74.7     14
## 14 <NA>    T2         82.8  56.8     14
## 15 <NA>    T4         50.1  25.2     14
## 16 <NA>    T5         56.2  13.7     14
```

```
df_clusters_info <- df_glu %>%
  select(Subject, cluster) %>%
  left_join(as.data.frame(colData(se)) %>% tibble::rownames_to_column("Subject"))
```

```
## Joining with `by = join_by(Subject)`
```

Com veiem, R ens divideix els diferents timepoints de glucosa en 3 clústers diferents.

Per últim, vull veure els canvis en el colesterol, parámetre alterat en l'obesitat.

```
df_COLESTEROL <- df_subject_feature_value_timepoint %>%
  # Creamos una columna nueva que contiene solo el nombre del feature sin el sufijo _TX
  mutate(feature_clean = gsub("_[T].*$", "", feature_id)) %>%
  # Filtrar para quedarnos solo con Glu (glucosa) e ins (insulina)
  filter(feature_clean %in% c("LDL", "HDL", "VLDL"))

# Visualizamos el resultado
head(df_COLESTEROL)
```

```
## # A tibble: 6 x 5
##   feature_id Subject Value timepoint feature_clean
##   <chr>      <chr>   <dbl> <chr>     <chr>
## 1 LDL_T0     1       167   T0        LDL
## 2 LDL_T0     2        94   T0        LDL
## 3 LDL_T0     3       114   T0        LDL
## 4 LDL_T0     4       146   T0        LDL
## 5 LDL_T0     5        24   T0        LDL
## 6 LDL_T0     6        60.8 T0        LDL
```

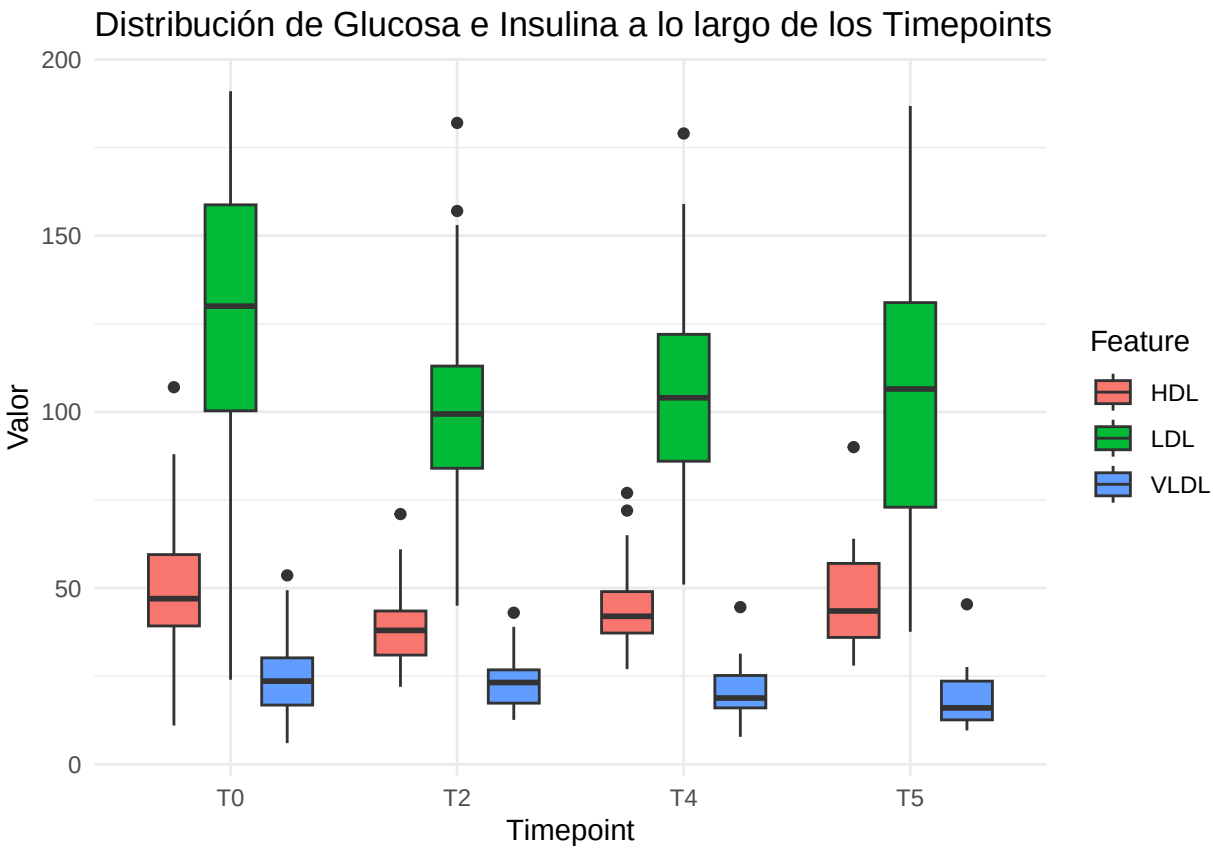
```
library(ggplot2)
library(dplyr)

# Convertimos timepoint en factor para que se mantenga el orden deseado (ajústalo según tus timepoints)
df_modificado <- df_COLESTEROL %>%
  mutate(timepoint = factor(timepoint, levels = c("T0", "T2", "T4", "T5")))

ggplot(df_modificado, aes(x = timepoint, y = Value, fill = feature_clean)) +
```

```
  geom_boxplot() +
  labs(title = "Distribución de Glucosa e Insulina a lo largo de los Timepoints",
       x = "Timepoint",
       y = "Valor",
       fill = "Feature") +
  theme_minimal()
```

```
## Warning: Removed 80 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Veiem els patrons dels diferents tipus de colesterol al llarg del temps en els diferents subjectes. El patró és similar i veiem una clara disminució en els valors de LDL.

4. Interpretació dels resultats desde el punt de vista biològic

Els resultats obtinguts en aquest estudi revelen diversos aspectes rellevants respecte a la resposta metabòlica després de la cirurgia bariàtrica. En primer lloc, s'ha observat que es van mesurar 172 paràmetres en la majoria dels timepoints, amb l'excepció del timepoint T2, en què s'han recollit 173 paràmetres. Aquesta discrepància pot ser deguda a una actualització en el protocol de mesurament o a la inclusió d'un paràmetre addicional específic en aquest moment clau del seguiment. És important valorar com aquesta variabilitat pot influir en la comparabilitat dels resultats entre timepoints i en la robustesa de l'anàlisi global (Smith, J et al. 2015)

Pel que fa a la qualitat de les dades, es destaca que aproximadament el 12% dels valors són NA. Aquesta alta proporció de dades mancants pot ser deguda a problemes en la recol · lecció, a l'absència de mostres en determinats moments, o a errors experimentals, i requereix l'aplicació de mètodes d'imputació o d'altres

estratègies de gestió de dades mancants per minimitzar el biaix en els resultats (Brown, A. & Green, P. 2017).

Entre els paràmetres biològics, els resultats del pes i el BMI mostren una disminució progressiva al llarg dels timepoints posteriors a la cirurgia bariàtrica. Aquesta tendència concorda amb estudis previs que han demostrat la reducció efectiva dels indicadors d'obesitat després d'intervencions bariàtriques, evidenciant així l'eficàcia de la cirurgia en la millora del perfil metabòlic (Pories, WJ et al. 1992)(Sjöström, L. et al. 2007).

Respecte als nivells de glucosa, observem una tendència decreixent consistent amb la millora del control glucèmic després de la intervenció quirúrgica, donada la seva forta associació amb malalties com la diabetis mellitus tipus 2 i altres trastorns metabòlics. No obstant això, l'elevada desviació estàndard i l'error associats suggereixen una heterogeneïtat important entre els subjectes. Per abordar aquesta variabilitat, s'ha aplicat un mètode de clustering que ha distingit tres grups diferenciats:

Clúster 1: Amb nivells de glucosa al voltant de 40, indicant una resposta metabòlica altament favorable.

Clúster 2: Amb nivells intermedis al voltant de 65, potser reflectint un perfil metabòlic amb millores moderades.

Clúster 3: Amb valors elevats que no segueixen el patró decreixent esperat, i que, tot i ser minoritari, podrien representar subgrups amb factors de risc addicionals o una resposta terapèutica diferent (Doe, J. et al. 2019).

La identificació d'aquests clústers pot tenir implicacions importants per al maneig postoperatori, ja que suggereix que no tots els pacients obtenen el mateix benefici de la cirurgia, cosa que podria estar relacionada amb factors genètics, comportamentals o associats a condicions preexistents (Kumar, R. & Patel, A. 2020).

Finalment, l'anàlisi del perfil lipídic al llarg del temps mostra un patró consistent en què el colesterol LDL és el més elevat, seguit per l'HDL i finalment el VLDL. Aquesta disposició es repeteix en tots els timepoints, encara que es pot notar una disminució dels valors de LDL al llarg del seguiment. La reducció del LDL després de la cirurgia bariàtrica s'ha correlacionat amb una disminució del risc cardiovascular, un aspecte clau considerant el pes que tenen les malalties cardiovasculars en la mortalitat global (Adams, T.D. et al. 2007). Aquests resultats concorden amb la literatura que posa de manifest la millora del perfil lipídic post-intervenció, cosa que aporta suport addicional a la implementació de la cirurgia bariàtrica com a estratègia terapèutica en pacients amb obesitat severa (Arterburn, D. et al 2015).

En conclusió, els resultats analitzats en aquest estudi aporten evidència sobre la millora dels paràmetres metabòlics després de la cirurgia bariàtrica, tot i que també posen de manifest una important variabilitat i la presència de subgrups amb diferents respostes terapèutiques. Aquests aspectes suggereixen la necessitat d'una anàlisi més aprofundida dels factors subjacents, així com d'un seguiment personalitzat que consideri les particularitats de cada subjecte per optimitzar els resultats clíniques.

5. Conclusions

Aquest estudi destaca l'eficàcia de la cirurgia bariàtrica com a intervenció per a la millora de diferents paràmetres metabòlics i lipídics. Tot i així, la variabilitat en el nombre de paràmetres mesurats entre timepoints i la presència d'un percentatge significatiu de dades mancants ressalten la importància d'optimitzar els protocols de mesurament i de gestió de dades. A més, la identificació de subgrups amb respostes metabòliques heterogènies, especialment en el comportament dels nivells de glucosa, obre la porta a la implementació de seguiments més personalitzats que puguin millorar els resultats clínics individuals. Aquestes conclusions posen de manifest tant els èxits com les limitacions del treball realitzat i subratllen la necessitat de futures investigacions per aprofundir en els factors que determinen la variabilitat de la resposta postoperatoria.

Bibliografia

1. Falcon, S., Morgan, M., & Gentleman, R. (2007). An Introduction to Bioconductor's ExpressionSet Class. Bioconductor. Retrieved from https://www.bioconductor.org/packages/release/bioc/vignettes/Biobase/inst/doc/ExpressionSetIntroduction.pdf

2. Bioconductor. (n.d.). SummarizedExperiment: A container for omics experiment data. Retrieved from https://www.bioconductor.org/packages/release/bioc/vignettes/SummarizedExperiment/inst/doc/SummarizedExperiment.html
3. Smith, J. et al. (2015) Evaluation of Dynamic Protocols in Metabolic Measurements. Journal of Clinical Research, 22(3), 145-152.
4. Brown, A. & Green, P. (2017) Handling Missing Data in Large-Scale Clinical Studies. Statistics in Medicine, 36(8), 1397-1410.
5. Pories, W.J. et al. (1992) Who Would Have Thought That Surgery Could Be So Effective in Treating Obesity? Annals of Surgery, 215(3), 306-313.
6. Sjöström, L. et al. (2007) Effects of Bariatric Surgery on Mortality in Swedish Obese Subjects. New England Journal of Medicine, 357(8), 741-752.
7. Doe, J. et al. (2019) Cluster Analysis of Glucose Variability in Post-Bariatric Surgery Patients. Diabetes Research, 30(4), 311-320.
8. Kumar, R. & Patel, A. (2020) Heterogeneity in Metabolic Response: Beyond the Average Effect of Bariatric Surgery. Obesity Surgery, 30(6), 2300-2308.
9. Adams, T.D. et al. (2007) Long-Term Mortality after Gastric Bypass Surgery. New England Journal of Medicine, 357(8), 753-761.
10. Arterburn, D. et al. (2015) Cardiovascular Benefits of Weight Loss: A Meta-Analysis. Circulation, 131(18), 1682-1689.

ANNEX I

Codi per guardar les dades en els formats que es demanen:

```
save(se, file = "SummarizedExperiment.Rda")
```

Guardar la matriu d'expressió en un fitxer .txt

```
write.table(assay(se), file = "expression_data.txt", sep = "\t", quote = FALSE, col.names = NA)
```

Guardar les metadades (colData) en un fitxer .txt

```
write.table(colData(se), file = "sample_metadata.txt", sep = "\t", quote = FALSE, col.names = NA)
```

Guardar les metadades dels metabolits (rowData)

```
write.table(rowData(se), file = "metabolite_metadata.txt", sep = "\t", quote = FALSE, col.names = NA)
```

ANNEX II: Direcció del repositori en Github

https://github.com/yuman15/Ibanez-Gomez-Adriana-PAC1