

PEC nº01. Análisis de Datos Ómicos

Ana Gabriela Ponce González

8 de abril, 2025

Índice

Resumen.....	1
Objetivos.....	1
Métodos.....	2
Resultados.....	4
Discusión.....	7
Conclusiones.....	9
Referencias.....	10
Anexos.....	11

Resumen

En esta primera prueba de evaluación continua se ha realizado un análisis exploratorio de un conjunto de datos de metabolómica obtenidos del repositorio GitHub NutriMetabolomics Lab (Lab 2024). El dataset seleccionado, MetabotypingPaper, cuenta con perfiles metabólicos y metadatos clínicos de pacientes sometidos a diferentes tipos de cirugías bariátricas. Los datos fueron preprocesados imputando los valores NA y luego se procedió a la normalización de los mismos.

Se emplearon diversas técnicas de análisis exploratorio, incluyendo gráficos de caja, distribución de densidades y análisis de componentes principales... Para todo se ha hecho uso del paquete POMA de R, estructura de Bioconductor que internamente contiene un objeto de clase SummarizedExperiment, requerido en la evaluación.

Por último, se llevaron a cabo análisis de correlación y análisis estadístico diferencial que permitieron identificar metabolitos con diferencias significativas entre los grupos quirúrgicos y obtener una visión general de la estructura de datos.

Objetivos

- **Objetivo general.** Aplicación de técnicas bioinformáticas y bioestadísticas para llevar a cabo un análisis exploratorio completo de los datos de metabolómica seleccionados.

- **Objetivos específicos.** Seleccionar un conjunto de datos metabolómicos; construir un objeto de clase “SummarizedExperiment” que integre los datos de expresión y los metadatos clínicos; realizar el preprocesamiento de los datos; aplicar técnicas de análisis exploratorio; evaluar posibles correlaciones entre metabolitos y detectar diferencias estadísticamente significativas entre los grupos clínicos; e interpretar los resultados obtenidos desde un punto de vista biológico.

Métodos

Conjunto de datos:

El conjunto de datos utilizado en este trabajo proviene del repositorio de GitHub proporcionado por los responsables de la asignatura. Concretamente se ha seleccionado el dataset denominado *MetabotypingPaper*. Este dataset contiene perfiles metabolómicos obtenidos mediante espectrometría de masas en muestras biológicas humanas, junto con metadatos clínicos de pacientes sometidos a diferentes tipos de cirugía bariátrica.

Justificación en la selección del conjunto de datos:

El dataset fue seleccionado por contener datos reales de metabolómica humana, junto con información clínica detallada sobre los pacientes. Además, su estructura lo hace especialmente adecuado para realizar el análisis exploratorio y estadístico. Por último, mencionar que, si se estuviese haciendo un análisis a escala real, el contexto clínico en el que se recogen los datos permitiría una interpretación biológica muy relevante.

Creación del objeto “SummarizedExperiment”:

Los datos fueron cargados en R y reorganizados. Posteriormente, se construyó un objeto mediante la función “PomaCreateObject()” del paquete “POMA” (Pol Castellano-Escuder 2021), por considerarlo de mayor eficiencia; que genera un objeto de clase “POMA” basado internamente en la clase “SummarizedExperiment”. Este objeto permite almacenar y gestionar de forma estructurada los datos ómicos y sus metadatos asociados.

En primer lugar cargamos el dataset elegido. Este archivo contine datos y metadatos juntos:

```
features <- readr::read_csv("DataValues_S013.csv", col_names = TRUE)

## 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`
```

```

# Vamos a realizar una exploración básica de los mismos para ver cómo
están organizados:
head(features, 10)
str(features)

# Después de haber visto como se organizan, las primeras variables, la
presencia de valores nulos, el tipo de datos... vamos a proceder a
limpiar los datos. Para ello, eliminamos la primera columna y la fila 26,
ya que los metadatos no pueden tener valores nulos (NA):
features <- features[-1]
target <- features[-26,1:9]

# Luego, vamos a modificar el nombre de la observación "bypass" por
"by_pass":
target$SURGERY[target$SURGERY == "by pass"] = "by_pass"

features <- features[-26,-c(1:9)]

# Como ya la base de datos está acomodada, podemos crear el objeto. Como
comentamos anteriormente, hemos optado por el uso de POMA para crear la
clase SummarizedExperiment (SE), ya que la función PomaCreateObject() del
paquete POMA genera internamente un objeto de clase POMA que encapsula un
SE
objeto <- PomaCreateObject(metadata = target, features = features)

```

Guardar el objeto: (Alboukadel Kassambara 2024b)

```

save(objeto, file = "SummarizedExperiment_objeto.Rda")

```

Las principales diferencias entre la clase ExpressionSet y el objeto SummarizedExperiment son que la primera está orientada al análisis de datos en microarrays, mientras que SummarizedExperiment es una clase más reciente que responde a la necesidad de desarrollar herramientas para trabajar con los datos masivos actuales. Aquí, en lugar de usar assayData, como en ExpressionSet, se usan los assays (listas con matrices o arrays). Por último, a diferencia de la clase ExpressionSet, SummarizedExperiment permite también el almacenamiento de los metadatos (Martin Morgan, Valerie Obenchain, Jim Hester, Hervé Pagès 2023) (Alboukadel Kassambara 2024a).

Flujo de análisis y herramientas utilizadas:

El análisis se llevó a cabo en R utilizando los paquetes “POMA”, “tidyverse”, “ggplot2”, “magrittr” y “ggtext”. La metodología seguida constó de los siguientes pasos:

1. **Imputación de valores faltantes** mediante el método kNN (“PomaImpute()”), considerando valores cero como NA y eliminando variables con más del 20% de datos faltantes.
2. **Normalización de los datos** utilizando escalado z-score (“PomaNorm(method =”auto_scaling”)”).
3. **Análisis exploratorio** mediante gráficos de caja y densidades para evaluar la distribución y homogeneidad de los datos.

4. **Análisis de componentes principales (PCA)** para visualizar la variabilidad entre muestras.
5. **Análisis de correlación** con la finalidad de estudiar las relaciones entre metabolitos ("PomaCorr()").
6. **Análisis estadístico diferencial** utilizando test de Mann-Whitney ("PomaUnivariate()") (POMA Package 2020b) y modelos lineales ajustados con "LIMMA" ("PomaLimma()") (POMA Package 2020a).

Resultados

Análisis exploratorio:

Código en R completo y resto del material requerido en la evaluación disponible en [el repositorio de GitHub habilitado para esta PEC](#)

```
# Para poder hacer el análisis exploratorio, iniciamos con la imputación de los NA empleando el método kNN:
imputed <- objeto %>%
  PomaImpute(method = "knn", zeros_as_na = TRUE,
             remove_na = TRUE, cutoff = 20)

## 31 features removed.

# Ahora, aplicamos la normalización por auto-escalado (z-score), que centra y escala cada variable para que tenga media cero y ds uno. De esta forma podemos comparar las variables sin que la escala afecte al resto del análisis:
normalized <- imputed %>%
  PomaNorm(method = "auto_scaling")

# A continuación, vamos a generar gráficos de caja por muestra y variable para detectar outliers y ver el efecto de la normalización:

# Boxplot por muestras antes de normalizar:
PomaBoxplots(imputed, x = "samples") +
  ggtitle("No Normalizado") +
  theme(legend.position = "none")

# Boxplot por variables antes de normalizar (computacionalmente muy demandante):
PomaBoxplots(imputed, x = "features") +
  ggtitle("No Normalizado") +
  theme(legend.position = "none")

# Boxplot por muestras después de normalizar:
PomaBoxplots(normalized, x = "samples") +
  ggtitle("Normalizado") +
  theme(legend.position = "none")
```

```
# Boxplot por variables después de normalizar (computacionalmente muy
demandante):
PomaBoxplots(normalized, x = "features") +
  ggtitle("Normalizado") +
  theme(legend.position = "none")

# Por otro lado, PomaDensity nos muestra la distribución de todas las
características antes y después del proceso de normalización. La
normalización debería homogeneizar la dispersión:
PomaDensity(imputed) +
  ggtitle("No normalizado") +
  theme(legend.position = "none")

PomaDensity(normalized) +
  ggtitle("Normalizado") +
  theme(legend.position = "none")

# Ahora eliminamos posibles muestras outlier con la función PomaOutliers.
De esta forma, trabajamos con un conjunto más robusto para el análisis
multivariante:
pre_processed <- PomaOutliers(normalized)$data

# Analisis de componentes principales:
poma_pca <- PomaPCA(pre_processed, load_length = 0)

# Ahora, representamos la varianza explicada por cada componente. Con
esta primera se explican aprox el 60% de la varianza de los datos:
plot(poma_pca$eigenvalues_plot)

# Esta segunda muestra la distribución de las muestras de acuerdo a su
varianza:
plot(poma_pca$biplot)
```

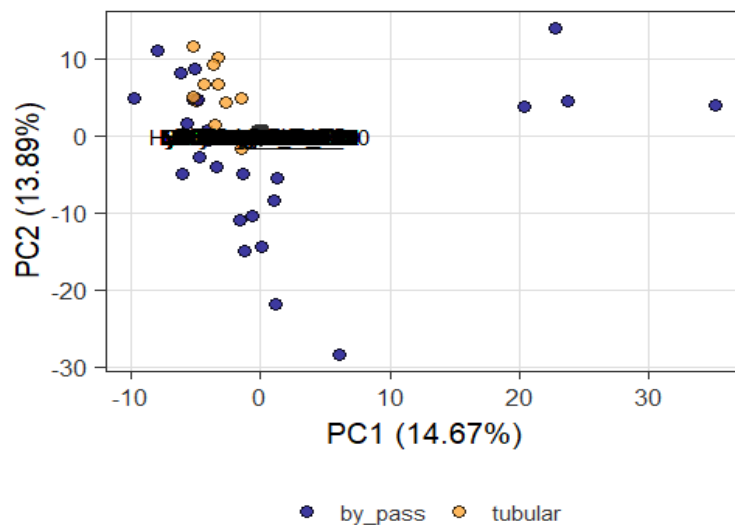


Figura 8. Biplot del PCA. Se observan tendencias de separación entre los grupos quirúrgicos by pass y tubular en el espacio de las componentes principales.

```
# A continuación estudiamos las correlaciones entre metabolitos
utilizando la matriz de correlaciones de Pearson:
poma_cor <- PomaCorr(pre_processed, method = "pearson",
                     cluster = TRUE, sig_level = 0.6)
# Visualizamos las correlaciones entre las variables:
plot(poma_cor$corrplot)

# Ahora, con un análisis univariante identificamos los metabolitos
diferencialmente expresados entre los grupos quirúrgicos. Empleamos el
test de Mann-Whitney (no paramétrico) que es adecuado para datos sin
distribución normal:
results <- pre_processed %>%
  PomaUnivariate(method = "mann") %>%
  dplyr::select(feature, fold_change, pvalue)
```

Referencia: (POMA Package 2020c)

```
# Representamos los resultados mediante un volcano plot que muestra la
relación entre la magnitud del cambio (log2 Fold Change) y la
significancia estadística: (-log10 p-value):
```

```
results %>%
  PomaVolcano(pval_cutoff = 0.05,
              log2fc_cutoff = NULL,
              labels = TRUE,
              x_label = "log2 (Fold Change)",
              y_label = "-log10 (P-value)")
```

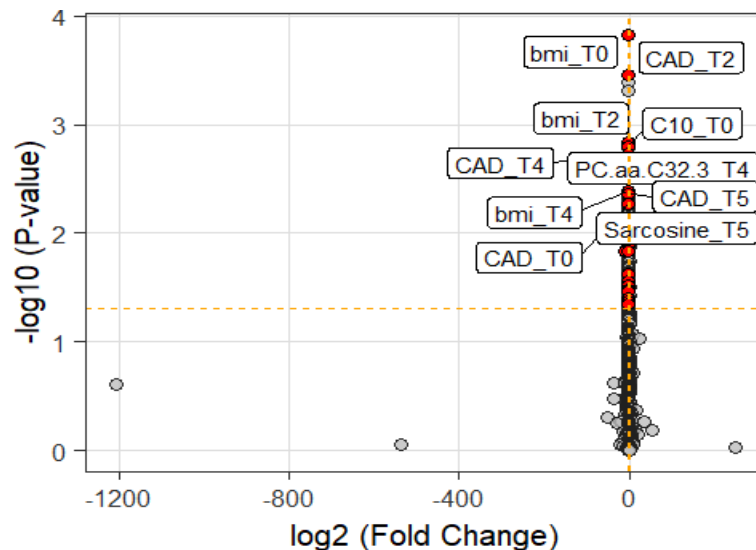


Figura 10. Volcano plot del análisis univariante (Mann-Whitney). Se identifican metabolitos diferencialmente expresados entre grupos quirúrgicos.

```
# Finalmente, aplicamos un modelo lineal mediante el método LIMMA,
mejorando la detección de diferencias significativas entre grupos al
modelar mejor la varianza. Definimos el contraste entre los grupos
"tubular" y "by_pass":
```

```

results <- pre_processed %>%
  PomaLimma(contrast = "tubular - by_pass") %>%
  dplyr::select(feature, log2FC, pvalue)

# Representamos de nuevo Los resultados con un volcano plot:
results %>%
  PomaVolcano(pval_cutoff = 0.05,
              log2fc_cutoff = NULL,
              labels = TRUE,
              x_label = "log2 (Fold Change)",
              y_label = "-log10 (P-value)")

```

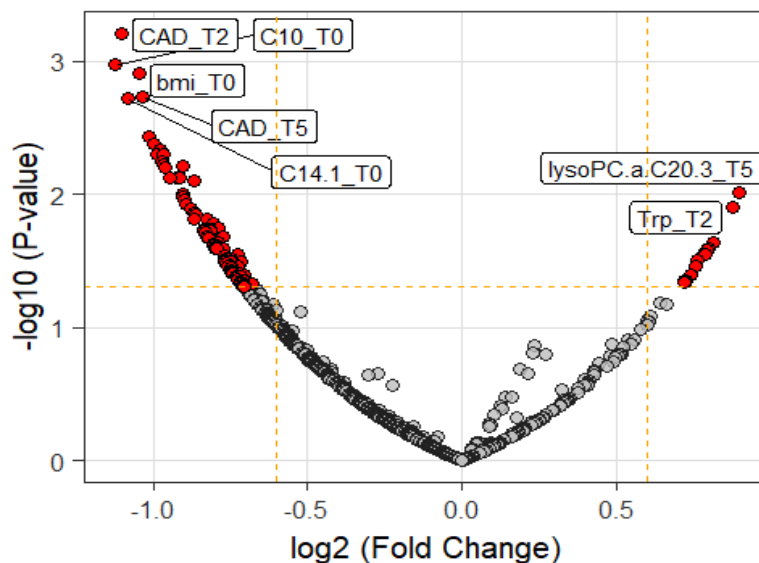


Figura 11. Volcano plot del análisis multivariante (LIMMA). Se destacan metabolitos relevantes tras modelar la varianza entre muestras.

Discusión

A través de nuestro análisis exploratorio podemos identificar patrones relevantes. El análisis se dividió en diferentes etapas:

Imputación de valores perdidos:

El preprocesamiento de la base de datos comenzó con la imputación de los valores NA mediante el método de los k vecinos (kNN), que básicamente estima los valores faltantes a partir de las muestras similares. En este caso, imputar nos ha permitido evitar eliminar metabolitos con valores NA, lo que es especialmente relevante debido a la alta dimensionalidad del dataset (696 variables) y el número limitado de pacientes (n = 39).

Normalización y visualización con boxplots:

Posteriormente se graficaron boxplots por muestra y variable antes de la normalización (Fig.1 y Fig2, véase en el anexo), y pudimos ver una gran dispersión de los valores, con múltiples

outliers: algunas variables presentan concentraciones desproporcionalmente elevadas respecto al resto.

Tras aplicar normalización por autoescalado (z-score), las distribuciones se homogenizaron (Fig 3, véase en el anexo). Las muestras quedaron centradas en cero y con varianza unitaria, lo que nos permitió comparar perfiles sin que la escala de las variables influya. Desde un punto de vista biológico, esto significa que las diferencias observadas entre grupos quirúrgicos se pueden atribuir a variaciones reales en los perfiles metabólicos, y no factores técnicos.

Distribuciones de densidad:

Con los gráficos de densidad reforzamos esta observación. En el conjunto de datos sin normalizar (Fig 5, véase en el anexo), la mayoría de los metabolitos muestran distribuciones asimétricas con colas largas, indicando presencia de valores extremos. Tras la normalización (Fig 6, véase en el anexo), las distribuciones se aproximan más a la normalidad, lo que facilita la aplicación de los métodos estadísticos posteriores y reduce el sesgo. Detección de outliers

Posteriormente eliminamos los outliers mediante PomaOutliers. Estas muestras, al presentar patrones metabólicos aberrantes, podrían corresponderse a errores técnicos o a pacientes con características clínicas distintas (comorbilidades no recogidas, efectos adversos, etc.). Al eliminarlos podemos enfocarnos en los efectos generales de la cirugía bariátrica sobre el metabolismo.

Análisis de Componentes Principales (PCA):

El análisis de componentes principales mostró que las dos primeras componentes explican el 60 % de la variabilidad del dataset (Fig 7, véase en el anexo). En el biplot (Figura 8), se observan tendencias a la separación entre los grupos quirúrgicos by pass y tubular, principalmente a lo largo de la componente PC1. Desde el punto de vista biológico, esta diferenciación tiene sentido ya que se sabe que no todas las cirugías generan la misma respuesta metabólica. Dichos cambios pueden estar asociados a variaciones en la absorción intestinal (algunos tipos de cirugía implican mayor o menor resección de superficie de absorción), secreción de hormonas entéricas, inflamación sistémica y metabolismo hepático.

Correlación entre metabolitos:

El mapa de correlaciones vemos que está muy sobrecargado debido, nuevamente, a la magnitud de la base de datos. Sin embargo, en él se pueden apreciar (Fig. 9, véase en el anexo) agrupaciones de metabolitos con comportamiento similar, lo cual sugiere la activación coordinada de ciertas rutas metabólicas. Por ejemplo, se observan agrupaciones que podrían corresponder a acilcarnitinas de cadena media, aminoácidos de cadena ramificada o metabolitos lipídicos relacionados. Lo que tendría sentido si pensamos en que luego de someterse a un procedimiento quirúrgico de este calibre el cuerpo sufre ciertas adaptaciones metabólicas a esos cambios fisiológicos. Estas agrupaciones podrían ser un reflejo de este hecho.

Análisis univariante (Mann-Whitney):

Por otro lado, análisis univariante empleando el test de Mann-Whitney nos permitió identificar metabolitos diferencialmente expresados entre los grupos (Figura 9). Entre los más destacados están: **C10_T0 y C14.1_T0** (acilcarnitinas implicadas en el metabolismo mitocondrial de ácidos grasos. Su diferencia podría implicar un cambio en la oxidación lipídica temprana tras la intervención), **BMI_T0 y BMI_T2** (reflejan el cambio esperado en el índice de masa corporal a lo largo del tiempo, que varía según el tipo de cirugía) y **CAD_T0-T5** (están asociados al riesgo cardiovascular, podrían estar modulados por la pérdida de peso y cambios inflamatorios derivados del procedimiento).

Análisis multivariante (LIMMA):

El modelo multivariante basado en el método LIMMA identificó diferencias más robustas al tener en cuenta la variabilidad entre muestras. En el volcano plot correspondiente (Fig. 10), destacan metabolitos como el **LysoPC.a.C20.3_T5** (lisofosfatidilcolina involucrada en señalización lipídica e inflamación, cuya regulación podría reflejar una respuesta inmunometabólica a medio plazo) y **Trp_T2** (triptófano, un aminoácido esencial implicado en la síntesis de serotonina, vinculado tanto al estado nutricional como a la modulación de la inflamación).

Conclusiones

En este trabajo hemos evaluado los perfiles metabolómicos de los pacientes sometidos a cirugía bariátrica empleando un análisis que partió en el preprocesamiento de los datos y culminó con la interpretación biológica de los resultados obtenidos.

Desde el punto de vista metodológico, gracias a la imputación de los valores ausentes evitamos la pérdida de variables potencialmente relevantes. Luego, con la normalización, la eliminación de los outliers y la transformación de las distribuciones aseguramos que las aplicaciones estadísticas fueran robustas. Por último, debemos destacar que la estrategia analítica seguida ha permitido abordar la base de datos desde múltiples perspectivas, lo que es especialmente importante a la hora de validar los resultados obtenidos.

En cuanto a la base de datos seleccionada, su interés radica en la información que recoge, ya que como se comentó anteriormente, si estuviésemos evaluando esto a gran escala y en un contexto más real, estos datos serían perfectos para ver cómo distintos enfoques terapéuticos pueden modular el metabolismo humano.

Finalmente, desde el punto de vista biológico y clínico, los resultados muestran que el tipo de cirugía bariátrica induce diferencias metabólicas significativas que se manifiestan desde etapas tempranas y se mantienen a lo largo del seguimiento. A lo largo del estudio se han identificado metabolitos diferencialmente expresados que refuerzan la idea de que este tipo de procedimientos afectan más allá de la simple pérdida de peso. Si se evalúan en un contexto más amplio, podrían identificarse también como marcadores de pronóstico clínico, por ejemplo.

Referencias

1. Ponce González, Ana Gabriela. 2025. «PEC1 - Análisis de Datos Ómicos». <https://github.com/AGabrielaPonceGlez/Ponce-Gonzalez-AnaGabriela-PEC1.git>
2. Alboukadel Kassambara. 2024a. «ExpressionSet and SummarizedExperiment». <https://www.sthda.com/english/wiki/expressionset-and-summarizedexperiment>
3. ———. 2024b. «Saving Data into R Data Format (RDS and RData)». <https://www.sthda.com/english/wiki/saving-data-into-r-data-format-rds-and-rdata>
4. Lab, Nutrimetabolomics. 2024. «metaboData: Clinical metabolomics data from bariatric surgery patients». <https://github.com/nutrimetabolomics/metaboData>
5. Martin Morgan, Valerie Obenchain, Jim Hester, Hervé Pagès. 2023. «SummarizedExperiment vignette». <https://www.bioconductor.org/packages/devel/bioc/vignettes/SummarizedExperiment/inst/doc/SummarizedExperiment.html>
6. Pol Castellano-Escuder. 2021. «POMA: User-friendly Workflow for Pre-processing and Statistical Analysis of Omics Data». <https://www.bioconductor.org/packages/release/bioc/html/POMA.html>
7. POMA Package. 2020a. «PomaLimma function documentation». <https://rdrr.io/bioc/POMA/man/PomaLimma.html>.
8. ———. 2020b. «PomaUnivariate function documentation». <https://rdrr.io/github/pcastellanoescuder/POMA/man/PomaUnivariate.html>
9. ———. 2020c. «PomaVolcano function documentation». <https://rdrr.io/bioc/POMA/man/PomaVolcano.html>

Anexos

Análisis exploratorio inicial:

Vamos a realizar una exploración básica de los mismos para ver cómo están organizados:

```
head(features,10)
```

```
str(features)
```

```
## # A tibble: 10 × 686
```

```
##   GLU_T0 INS_T0 HOMA_T0 HBA1C_T0 HBA1C.mmol.mol_T0 PESO_T0 bmi_T0  
CC_T0 CINT_T0
```

```
##   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
<dbl> <dbl>
```

```
## 1 85 11.4 2.4 NA NA 151 62.9  
0.7 116
```

```
## 2 78 12.1 2.32 NA NA 139 47 NA  
NA
```

```
## 3 75 8.41 1.56 5.4 35.5 84 29.8  
0.7 90
```

```
## 4 71 12.8 2.25 5.1 32.2 136 53.1 1  
157
```

```
## 5 82 6.01 1.22 5.6 37.7 121 46.6  
0.9 123
```

```
## 6 71 9.88 1.73 5.1 32.2 148 48.8  
0.7 110
```

```
## 7 80 9.2 1.82 5.6 37.7 109 43.7  
0.9 122
```

```
## 8 90 3.4 0.76 5.5 36.6 109 41.8  
0.9 124
```

```
## 9 92 5.43 1.23 5.7 38.8 114 44  
0.9 136
```

```
## 10 84 6.98 1.45 5.5 36.6 120 40.6 NA  
NA
```

```
## # i 677 more variables: CAD_T0 <dbl>, TAD_T0 <dbl>, TAS_T0 <dbl>,  
TG_T0 <dbl>,
```

```
## # COL_T0 <dbl>, LDL_T0 <dbl>, HDL_T0 <dbl>, VLDL_T0 <dbl>, PCR_T0  
<dbl>,
```

```
## # LEP_T0 <dbl>, ADIPO_T0 <dbl>, GOT_T0 <dbl>, GPT_T0 <dbl>, GGT_T0  
<dbl>,
```

```
## # URICO_T0 <dbl>, CREAT_T0 <dbl>, UREA_T0 <dbl>, HIERRO_T0 <dbl>,
```

```
## # TRANSF_T0 <dbl>, FERR_T0 <dbl>, Ile_T0 <dbl>, Leu_T0 <dbl>, Val_T0  
<dbl>,
```

```
## # Ala_T0 <dbl>, Pro_T0 <dbl>, Gly_T0 <dbl>, Ser_T0 <dbl>, Trp_T0  
<dbl>,
```

```
## # Phe_T0 <dbl>, Met_T0 <dbl>, Orn_T0 <dbl>, Arg_T0 <dbl>, His_T0  
<dbl>, ...
```

```
## tibble [38 × 686] (S3: tbl_df/tbl/data.frame)
```

```
## $ GLU_T0 : num [1:38] 85 78 75 71 82 71 80 90 92 84 ...
```

```
## $ INS_T0 : num [1:38] 11.4 12.1 8.41 12.8 6.01 9.88 9.2 3.4
```

```

5.43 6.98 ...
## $ HOMA_T0      : num [1:38] 2.4 2.32 1.56 2.25 1.22 1.73 1.82
0.76 1.23 1.45 ...
## $ HBA1C_T0     : num [1:38] NA NA 5.4 5.1 5.6 5.1 5.6 5.5 5.7 5.5
...
## $ HBA1C.mmol.mol_T0: num [1:38] NA NA 35.5 32.2 37.7 ...
## $ PESO_T0      : num [1:38] 151 139 84 136 121 148 109 109 114
120 ...
## $ bmi_T0       : num [1:38] 62.9 47 29.8 53.1 46.6 48.8 43.7 41.8
44 40.6 ...
## $ CC_T0        : num [1:38] 0.7 NA 0.7 1 0.9 0.7 0.9 0.9 0.9 NA
...
## $ CINT_T0      : num [1:38] 116 NA 90 157 123 110 122 124 136 NA
...
## $ CAD_T0       : num [1:38] 167 NA 126 162 132 148 141 136 148 NA
...
## $ TAD_T0       : num [1:38] 125 NA 79 73 84 74 65 75 91 NA ...
## $ TAS_T0       : num [1:38] 174 NA 111 127 122 131 121 128 128 NA
...
## $ TG_T0        : num [1:38] 147 150 45 109 30 61 75 53 164 145
...
## $ COL_T0       : num [1:38] 256 180 211 205 102 121 192 181 154
220 ...
## $ LDL_T0       : num [1:38] 167 94 114 146 24 60.8 96 99.4 67.2
160 ...
## $ HDL_T0       : num [1:38] 60 56 88 37 72 48 81 71 54 31 ...
## $ VLDL_T0      : num [1:38] 29.4 30 9 21.8 6 12.2 15 10.6 32.8 29
...
## $ PCR_T0       : num [1:38] 10.2 9 3.05 8.89 NA 1.6 4.52 NA NA NA
...
## $ LEP_T0       : num [1:38] 155 84 27 46 NA 38 61 NA NA NA ...
## $ ADIPO_T0     : num [1:38] 8.15 7.94 16.7 4.68 NA 11.8 6.51 NA
NA NA ...
## $ GOT_T0       : num [1:38] 21 16 24 21 42 11 14 50 21 15 ...
## $ GPT_T0       : num [1:38] 33 40 39 37 68 26 25 24 11 30 ...
## $ GGT_T0       : num [1:38] 22 25 20 20 18 12 16 36 16 52 ...
## $ URICO_T0     : num [1:38] 5.7 NA 2.7 4.9 5.2 3.7 3.8 5.6 5.1 5
...
## $ CREAT_T0     : num [1:38] 0.8 0.8 0.8 0.7 0.8 0.7 0.6 0.7 0.7 1
...
## $ UREA_T0      : num [1:38] 33 29 28 22 29 14 26 49 24 24 ...
## $ HIERRO_T0    : num [1:38] 77 113 142 64 94 102 72 13 36 113 ...
## $ TRANSF_T0    : num [1:38] NA 290 251 289 252 200 NA 300 288 235
...
## $ FERR_T0      : num [1:38] 53 25 15 18 39.4 25 24 23 48 101 ...
## $ Ile_T0       : num [1:38] 53.9 95.7 57.8 84.5 72 109 73.3 91.7
83.8 113 ...
## $ Leu_T0       : num [1:38] 105 188 119 180 140 205 147 191 183
225 ...
## $ Val_T0       : num [1:38] 192 317 183 296 247 376 261 340 320

```

```

325 ...
## $ Ala_T0          : num [1:38] 498 558 194 439 267 654 351 426 443
389 ...
## $ Pro_T0          : num [1:38] 160 249 153 216 156 498 155 163 222
255 ...
## $ Gly_T0          : num [1:38] 270 288 252 298 348 586 288 298 236
293 ...
## $ Ser_T0          : num [1:38] 161 174 143 144 184 310 125 165 188
134 ...
## $ Trp_T0          : num [1:38] 52.4 72 42.6 74.4 68.6 123 61.1 52.2
62.3 68.1 ...
## $ Phe_T0          : num [1:38] 62.1 98.3 50.8 94.8 75.4 121 66.7
89.7 100 72 ...
## $ Met_T0          : num [1:38] 19.7 36.6 23.3 28.3 36.8 53 24 28.1
23 30 ...
## $ Orn_T0          : num [1:38] 89.2 104 90.4 104 139 152 62.8 140
75.6 85.9 ...
## $ Arg_T0          : num [1:38] 110 166 101 142 133 204 132 107 179
133 ...
## $ His_T0          : num [1:38] 81.6 106 73.5 95.2 108 167 85.1 85
94.6 95.8 ...
## $ Asn_T0          : num [1:38] 40.8 50.4 36.7 46.1 58.7 82.1 45.7
45.3 51.9 66.3 ...
## $ Asp_T0          : num [1:38] 21 21.3 7.54 17.9 15.6 28.3 17 28.2
52.5 9.19 ...
## $ Glu_T0          : num [1:38] 38.7 66.9 16.9 44.6 54.1 60.8 37.2
56.3 108 80.8 ...
## $ Gln_T0          : num [1:38] 631 845 625 750 900 1320 666 696 597
844 ...
## $ Cit_T0          : num [1:38] 30.3 37.9 22.9 35 45.8 54.3 25.8 25.5
19.8 28.1 ...
## $ Tyr_T0          : num [1:38] 68.9 114 50.1 92.8 108 140 62 119
65.5 79.6 ...
## $ Thr_T0          : num [1:38] 150 203 107 107 144 280 113 133 210
132 ...
## $ Lys_T0          : num [1:38] 296 434 271 378 350 549 347 369 298
397 ...
## $ Creatinine_T0   : num [1:38] 66.9 75.3 39.1 79.3 66.4 113 51.7
57.7 68.5 100 ...
## $ Kynurenine_T0   : num [1:38] 2.55 3.3 1.77 3.16 3.57 6.51 1.61 3.6
2.92 3.24 ...
## $ Putrescine_T0    : num [1:38] 0.1 0.29 0 -99 0.279 0.209 0.159 -99
0.341 -99 ...
## $ Sarcosine_T0     : num [1:38] 7.7 11.7 2.05 8.11 3.77 12.2 6.61
5.63 2.67 7.49 ...
## $ Serotonin_T0     : num [1:38] 0.65 0.64 0.4 0.345 0.441 0.894 0.132
0.749 0.665 0.436 ...
## $ Taurine_T0       : num [1:38] 107 90.9 64.3 125 97.1 181 98 179 170
41 ...
## $ SDMA_T0         : num [1:38] 0.88 1.64 0.93 1.36 1.08 1.85 1.15

```

```

1.25 1.09 1.62 ...
## $ C0_T0 : num [1:38] 34.1 45.2 23.9 48.6 51.2 79.3 40.8
51.2 21.6 43.3 ...
## $ C2_T0 : num [1:38] 5.25 8.51 6 8.94 5.18 9.89 5.37 7.27
4.64 8.23 ...
## $ C3.OH_T0 : num [1:38] 0.26 0.27 0.24 0.23 0.28 0.37 0.18
0.17 0.26 0.15 ...
## $ C6..C4.1.DC._T0 : num [1:38] 0 0.15 0 -9 0.179 0.086 -9 0.086
0.096 -9 ...
## $ C5.DC..C6.OH._T0 : num [1:38] 0.05 0.07 0 0.044 0.076 -9 0.038
0.045 0.044 -9 ...
## $ C7.DC_T0 : num [1:38] 0.04 0.06 0.04 -9 0.071 -9 0.036 -9
0.042 -9 ...
## $ C8_T0 : num [1:38] 0.5 0.6 0.44 0.429 0.807 0.402 0.334
0.485 0.514 0.338 ...
## $ C10_T0 : num [1:38] 0.52 0.7 0.35 0.39 0.33 0.83 0.41
0.34 0.43 0.54 ...
## $ C10.1_T0 : num [1:38] 0.18 0.35 0.2 0.17 0.16 0.42 0.15
0.21 0.2 0.28 ...
## $ C10.2_T0 : num [1:38] 0.18 0.21 0.12 0.13 0.14 0.25 0.11
0.14 0.11 0.13 ...
## $ C14.1_T0 : num [1:38] 0.17 0.28 0.09 0.18 0.12 0.3 0.12
0.15 0.11 0.2 ...
## $ C14.2_T0 : num [1:38] 0.12 0.21 0.07 0.09 0.1 0.19 0.06
0.08 0.08 0.13 ...
## $ C16.1_T0 : num [1:38] 0.1 0.18 0.07 0.09 0.08 0.17 0.08
0.08 0.06 0.08 ...
## $ C16.2_T0 : num [1:38] 0.03 0.06 0.03 0.03 0.04 0.06 0.03
0.03 0.04 0.04 ...
## $ C16.2.OH_T0 : num [1:38] 0.04 0.06 0.04 0.04 0.04 0.05 0.03 0
0 0.03 ...
## $ C18.1_T0 : num [1:38] 0.18 0.22 0.1 0.23 0.12 0.25 0.12
0.15 0.08 0.13 ...
## $ C18.1.OH_T0 : num [1:38] 0.05 0.07 0.04 0.05 0.05 0.07 0.03
0.03 0.04 0.04 ...
## $ C18.2_T0 : num [1:38] 0.07 0.13 0.06 0.11 0.09 0.17 0.07
0.09 0.07 0.1 ...
## $ lysoPC.a.C16.0_T0: num [1:38] 167 149 82.2 187 86.1 228 141 109 153
137 ...
## $ lysoPC.a.C16.1_T0: num [1:38] 6.14 3.92 2.15 5.28 2.48 6.8 4.6 3.81
4.13 3.12 ...
## $ lysoPC.a.C17.0_T0: num [1:38] 3.05 1.69 1.91 3.26 1.13 3.58 2.32
2.11 1.42 1.27 ...
## $ lysoPC.a.C18.0_T0: num [1:38] 48.9 45.6 32 71.8 32.6 88.5 54.9 48.2
32.6 39.4 ...
## $ lysoPC.a.C18.1_T0: num [1:38] 42.6 28.4 21.7 50.8 22.4 58.2 39.1
29.1 19.4 26.8 ...
## $ lysoPC.a.C18.2_T0: num [1:38] 39 46 39.9 47.7 35.8 109 61.1 35.4
23.6 35.1 ...
## $ lysoPC.a.C20.3_T0: num [1:38] 4.84 3.51 2.54 8.33 1.57 7.5 4.94

```

```

3.78 4.75 3.03 ...
## $ lysoPC.a.C20.4_T0: num [1:38] 16 12.9 7.23 16.9 4.5 19.6 12.1 7.17
5.52 12.2 ...
## $ lysoPC.a.C24.0_T0: num [1:38] 0.47 0.35 0.41 0.54 0.46 0.56 0.48
0.35 0.34 0.46 ...
## $ lysoPC.a.C26.0_T0: num [1:38] 0.49 0.4 0.4 0.3 0.4 0.41 0.44 0.25
0.51 0.41 ...
## $ lysoPC.a.C26.1_T0: num [1:38] 0.35 0.46 0.33 0.19 0.3 0.5 0.4 0.14
0.27 0.3 ...
## $ lysoPC.a.C28.0_T0: num [1:38] 0.41 0.35 0.4 0.3 0.3 0.44 0.36 0.34
0.24 -9 ...
## $ lysoPC.a.C28.1_T0: num [1:38] 0.62 0.58 0.38 0.52 0.36 0.56 0.62
0.39 0.31 0.34 ...
## $ PC.aa.C24.0_T0 : num [1:38] 0.24 0.21 0.18 0.14 0.2 0.27 0.22
0.16 0.16 0.23 ...
## $ PC.aa.C28.1_T0 : num [1:38] 3.23 4.09 2.46 3.75 2.2 4.77 3.58
2.91 2.85 2.44 ...
## $ PC.aa.C30.0_T0 : num [1:38] 1.37 1.29 1.88 2.2 1.5 1.58 2.28 2.43
3.4 1.43 ...
## $ PC.aa.C32.0_T0 : num [1:38] 9.87 8.09 8.19 10.5 9.71 12.1 10.1
9.58 12.8 12.1 ...
## $ PC.aa.C32.1_T0 : num [1:38] 13 10.2 5.8 11.8 5.86 8.63 16.7 15
25.3 10.3 ...
## $ PC.aa.C32.3_T0 : num [1:38] 0.24 0.25 0.16 0.21 0.15 0.34 0.26
0.3 0.26 0.16 ...
## $ PC.aa.C34.1_T0 : num [1:38] 136 138 89.2 162 77.6 136 166 136 143
148 ...
## $ PC.aa.C34.2_T0 : num [1:38] 233 394 258 257 276 466 366 303 389
315 ...
## $ PC.aa.C34.3_T0 : num [1:38] 9.73 11 6.13 10.2 5.37 14.9 15.9 10.8
14.2 6.17 ...
## $ PC.aa.C34.4_T0 : num [1:38] 0.65 0.81 0.39 0.81 0.23 0.81 1.12
0.77 1.04 0.31 ...
## $ PC.aa.C36.0_T0 : num [1:38] 0.95 1.3 1.23 1.16 0.8 1.3 2 2.4 1.84
1.58 ...
## [list output truncated]

```

Parte de la limpieza inicial no incluida en el informe principal:

Ahora eliminamos posibles muestras outlier con la función PomaOutliers. De esta forma, trabajamos con un conjunto más robusto para el análisis multivariante:

```
pre_processed <- PomaOutliers(normalized)$data
```

Analisis de componentes principales:

```
poma_pca <- PomaPCA(pre_processed, load_length = 0)
```

Gráficos:

Figura n°01. Boxplot por muestras antes de la normalización.

No Normalizado

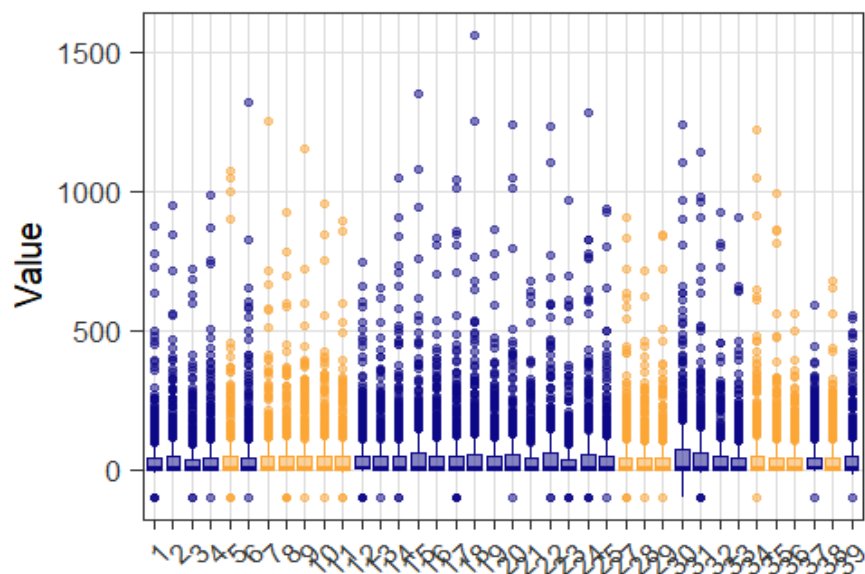


Figura n°02. Boxplot por variables antes de la normalización: se observa una amplia heterogeneidad en escalas y dispersión entre metabolitos, lo que justifica la necesidad de normalización (no podemos hacer comparaciones **directas**).

No Normalizado

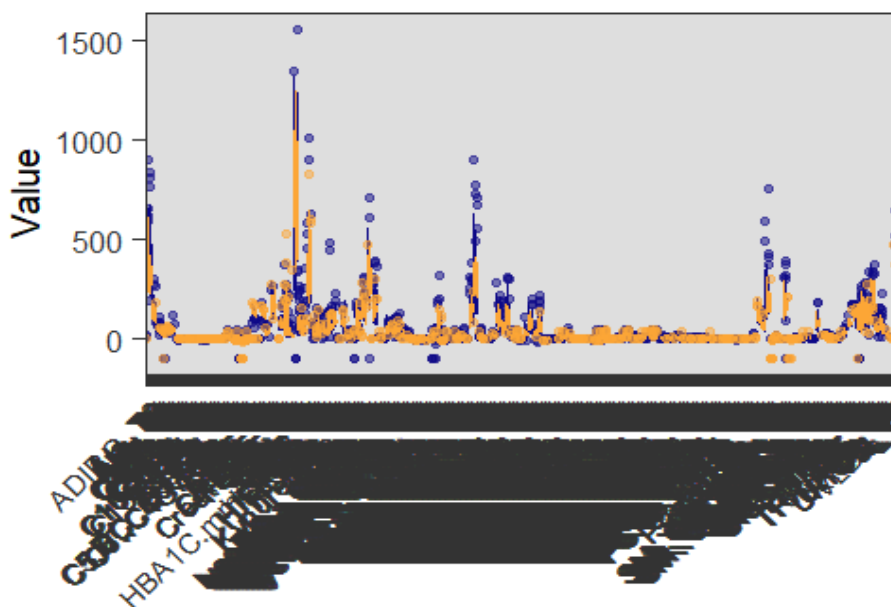


Figura n°03. Boxplot por muestras después de la normalización: las muestras muestran perfiles centrados y comparables, lo que permite interpretar la variabilidad como

diferencias biológicas reales y no como artefactos de escala.

Normalizado

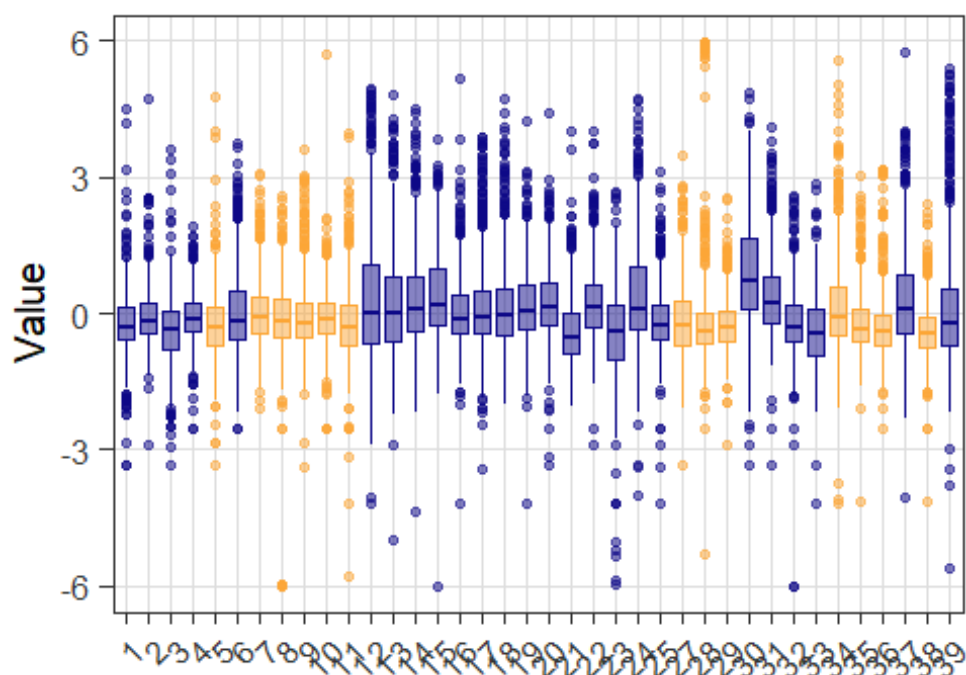


Figura n°04. Boxplot por variables después de la normalización: la escala común permite comparar directamente todas las variables, destacando la variabilidad real entre grupos quirúrgicos.

Normalizado

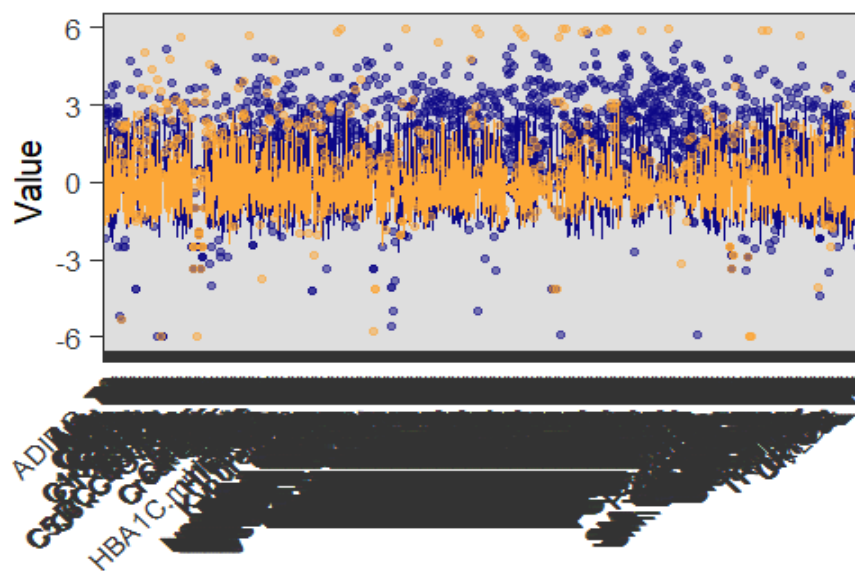


Figura nº05. *Densidad de metabolitos antes de la normalización:* se observa una fuerte asimetría y presencia de valores extremos, lo que justifica la necesidad de normalizar el dataset.

No normalizado

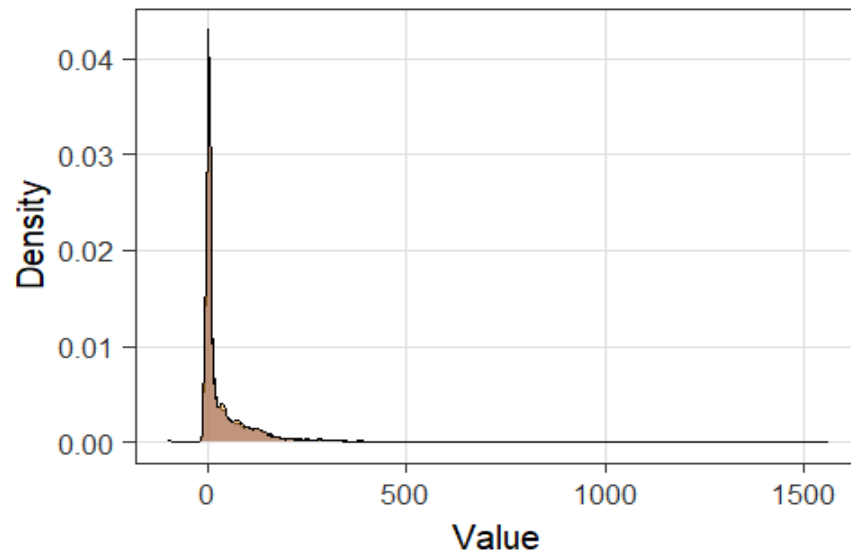


Figura nº06. *Densidad de metabolitos después de la normalización:* las variables muestran una forma más simétrica y centrada, lo que permite comparaciones más justas entre metabolitos y grupos.

Normalizado

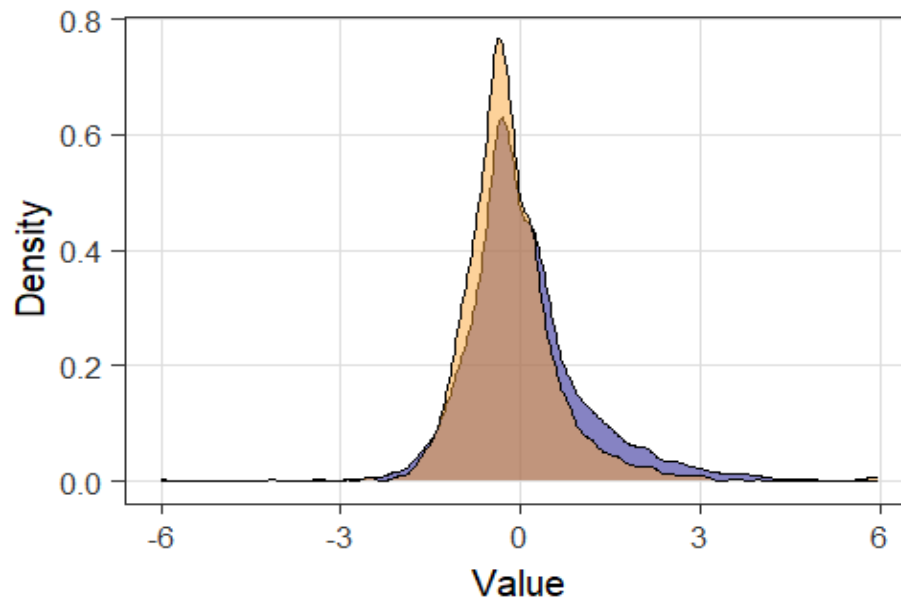


Figura n°07. Varianza explicada por el PCA: PC1 y PC2 juntas resumen el 60% de la variabilidad del conjunto de datos.

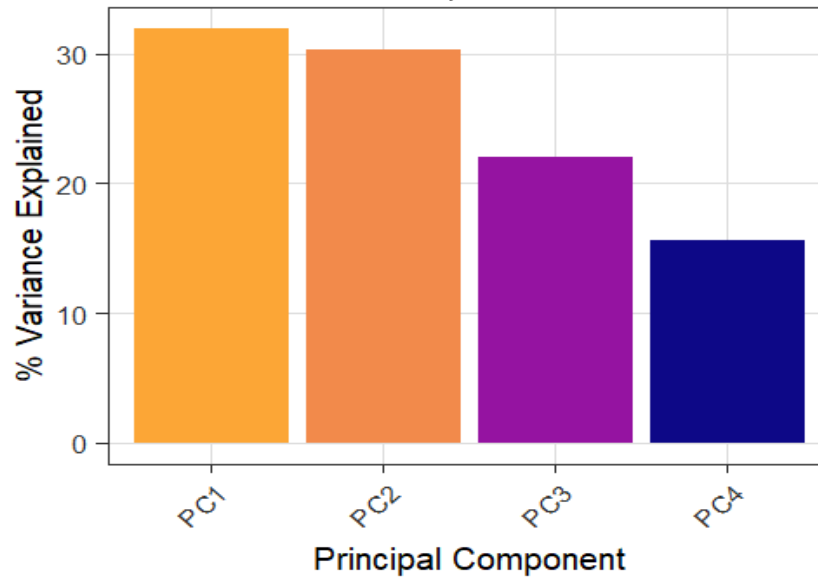


Figura n°08. Biplot del PCA: se observa una tendencia a la separación entre los grupos quirúrgicos by-pass y tubular a lo largo de PC1.

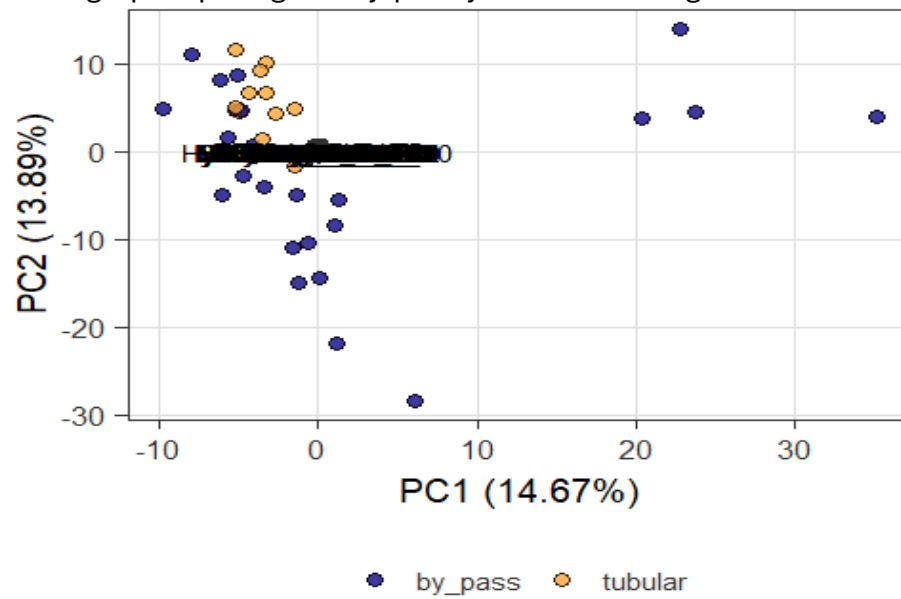


Figura nº09. Mapa de correlación entre metabolitos: se observan agrupaciones que indican regulación coordinada entre compuestos.

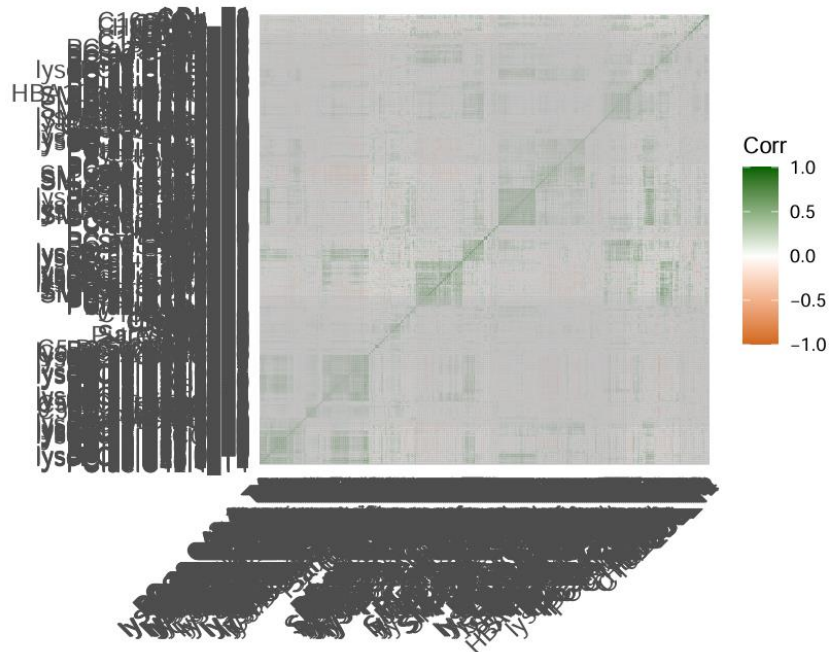


Figura nº10. Volcano plot del análisis univariante (Mann-Whitney): se muestran los metabolitos diferencialmente expresados entre grupos quirúrgicos, destacando compuestos relacionados con metabolismo lipídico, riesgo cardiovascular e índice de masa corporal.

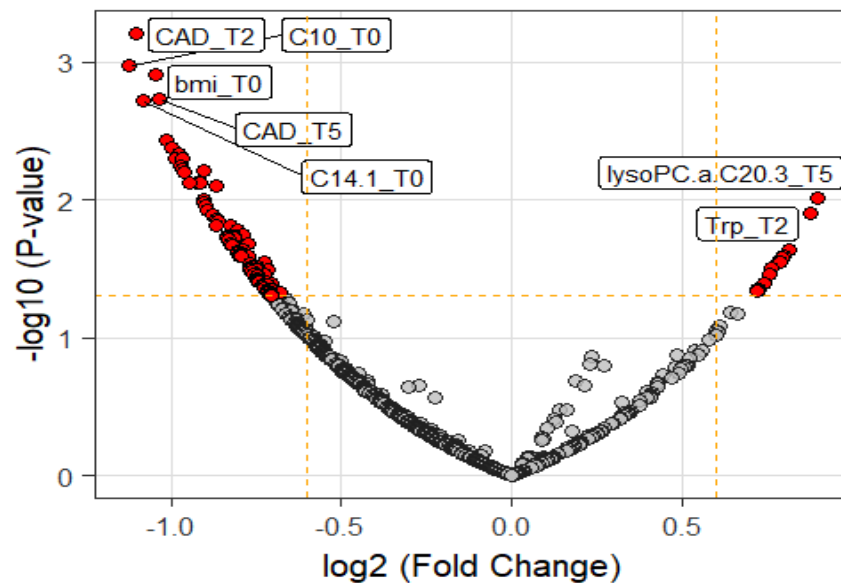


Figura nº11. Volcano plot del análisis multivariante (LIMMA): se muestran metabolitos diferencialmente expresados según significancia estadística y log2 Fold Change. Se destacan compuestos implicados en el metabolismo lipídico e inflamatorio.

