PEC1_Analisis_Omicos

Francesc Torrents Torre

2024-11-06

Primera prueba de evaluación contínua (PEC1)

Presentación y objetivos

Esta PEC completa la introducción a las ómicas meidante un ejercicio de repaso y ampliación que nos permite trabajar con algunas de las herramientas de este curso, en concreto, Bioconductor y la exploración multivariante de datos

Github: https://github.com/Frantt96/Torrents-Torre-Francesc-PEC1.git

Descripción de la PEC

El objetivo de esta PEC es que planifiquéis y ejecutéis una versión simplificada del proceso de análisis de datos ómicos, a la vez que practicáis con algunas de las herramientas y métodos que hemos trabajado

Materiales y Métodos

Para la realización de esta PEC es necesario la utilización de un dataset de metabolómica que obtendremos del siguiente repositorio de github: https://github.com/nutrimetabolomics/metaboData/

En mi caso, se ha escogido el dataset human_cachexia. Este viene en un solo documento .csv del cual tendremos que obtener los metadatos para poder realizar la clase SummarizedExperiment. Los datos proporcionados sobre este dataset son los siguientes:

- "Samples are not paired"
- "2 groups were detected in samples"
- "All data values are numeric"
- "A total of 0 (0%) missing values were detected"

Este está compuesto por una tabla de concentraciones de metabolitos provenientes de dos grupos de muestra de orina humana.

Resultados

Para la utilización de este Dataset es necesario descargado, ya sea copiando directamente el repositorio de Github o descargando el documento .csv (que ha sido el método utilizado).

Una vez descargado lo cargamos a RStudio con el siguiente codigo.

```r
data <- read.csv("human_cachexia.csv", header = TRUE, stringsAsFactors = FALSE)
```

Este cargará los datos de human_cachexia desde el directorio actual. Se indica que la priemra fila del archivo contiene los nombres de las columnas y no datos. Tambíen nos aseguramos de que las columnnas de texto se lean como cadenas de carácteres y no como factores para evitar de esta manera problemas posteriores.

Utilizaremos head(data) y str(data) para poder visualizar el tipo de datos que tenemos en el DataFrame y confirmar que la importanción es correcta. También observamos que las variables estan bien generadas.

```r
head(data)
```

```
##    Patient.ID Muscle.loss X1.6.Anhydro.beta.D.glucose X1.Methylnicotinamide
## 1     PIF_178    cachexic                       40.85                 65.37
## 2     PIF_087    cachexic                       62.18                340.36
## 3     PIF_090    cachexic                      270.43                 64.72
## 4 NETL_005_V1    cachexic                      154.47                 52.98
## 5     PIF_115    cachexic                       22.20                 73.70
## 6     PIF_110    cachexic                      212.72                 31.82
##   X2.Aminobutyrate X2.Hydroxyisobutyrate X2.Oxoglutarate X3.Aminoisobutyrate
## 1            18.73                 26.05           71.52              1480.30
## 2            24.29                 41.68           67.36               116.75
## 3            12.18                 65.37           23.81                14.30
## 4           172.43                 74.44         1199.91               555.57
## 5            15.64                 83.93           33.12                29.67
## 6            18.36                 80.64           47.94                17.46
##   X3.Hydroxybutyrate X3.Hydroxyisovalerate X3.Indoxylsulfate
## 1              56.83                 10.07            566.80
## 2              43.82                 79.84            368.71
## 3               5.64                 23.34            665.14
## 4             175.91                 25.03            411.58
## 5              76.71                 69.41            165.67
## 6              31.82                 35.16            183.09
##   X4.Hydroxyphenylacetate Acetate Acetone Adipate Alanine Asparagine Betaine
## 1                  120.30  126.47    9.49   38.09  314.19     159.17  109.95
## 2                  432.68  212.72   11.82  327.01  871.31     157.59  244.69
## 3                  292.95  314.19    4.44  131.63  464.05      89.12  116.75
## 4                  214.86   37.34  206.44  144.03  589.93     273.14  278.66
## 5                   97.51  407.48   44.26   15.03 1118.79      42.52  391.51
## 6                  132.95   81.45   14.44   25.28  237.46     157.59   66.69
##   Carnitine  Citrate Creatine Creatinine Dimethylamine Ethanolamine Formate
## 1    265.07  3714.50   196.37   16481.60        632.70       645.48  441.42
## 2    120.30  2617.57   212.72   15835.35        607.89       487.85  252.14
## 3     25.03   862.64   221.41   24587.66        735.10       407.48  249.64
## 4    200.34 13629.61    85.63   20952.22       1064.22       820.57  468.72
## 5     84.77   854.06   105.64    6768.26        242.26       365.04  114.43
## 6     40.04  1958.63   200.34   15677.78        614.00       459.44  314.19
##   Fucose Fumarate Glucose Glutamine Glycine Glycolate Guanidoacetate Hippurate
## 1 336.97     7.69  395.44    871.31 2038.56    685.40         154.47   4582.50
## 2 198.34    18.92 8690.62    601.85 1107.65    651.97         109.95   1737.15
## 3 186.79     7.10 1352.89    301.87  620.17    141.17         183.09   4315.64
## 4 407.48    96.54  862.64   1685.81 5064.45     70.81         102.51    757.48
## 5  26.05    19.69 6836.29    432.68  395.44     26.58          52.98   1152.86
## 6 123.97     5.05  512.86    298.87  482.99    428.38          57.97   3568.85
```

```
##   Histidine Hypoxanthine Isoleucine Lactate Leucine Lysine Methylamine
## 1    925.19        97.51       5.58  106.70   42.10 146.94       52.46
## 2    845.56        82.27       8.17  368.71   77.48 284.29       23.57
## 3    284.29       114.43       9.30  749.95   31.50  97.51       18.73
## 4   1043.15       223.63      37.71  368.71  103.54 290.03       48.91
## 5    327.01        66.69      40.04 3640.95  101.49 122.73       27.94
## 6    459.44        62.80       8.17  113.30   28.79 120.30       36.97
##   Methylguanidine N.N.Dimethylglycine O.Acetylcarnitine Pantothenate
## 1            9.97               23.34             52.98        25.79
## 2            7.69               87.36             50.40       186.79
## 3            4.66               24.53              5.58       145.47
## 4          141.17               40.04            254.68        42.52
## 5            5.31               46.06             45.60        74.44
## 6           43.38               24.29             13.46        35.52
##   Pyroglutamate Pyruvate Quinolinate  Serine Succinate Sucrose Tartrate Taurine
## 1        437.03    21.12      165.67  284.29    154.47   45.15    97.51 1919.85
## 2        437.03    36.97       72.97  391.51    244.69  459.44    32.79 1261.43
## 3        713.37    29.37      192.48  295.89    142.59  160.77    16.28 4272.69
## 4        566.80    64.07       86.49 1248.88    144.03  111.05   837.15 1525.38
## 5        184.93    12.30       38.09  206.44     68.72   75.19     4.53  468.72
## 6        432.68    32.79      112.17  387.61     33.45  336.97    24.05 2059.05
##   Threonine Trigonelline Trimethylamine.N.oxide Tryptophan Tyrosine Uracil
## 1    184.93       943.88               2121.76     259.82   290.03 111.05
## 2    198.34       208.51                639.06      83.10   167.34  46.99
## 3    109.95       192.48               1152.86      82.27    60.34  31.50
## 4    376.15       992.27               1450.99     235.10   323.76  30.57
## 5     64.07        86.49                172.43     103.54   142.59  44.26
## 6    105.64       862.64                880.07     239.85   127.74  29.67
##   Valine  Xylose cis.Aconitate myo.Inositol trans.Aconitate pi.Methylhistidine
## 1  86.49   72.24        237.46       135.64           51.94             157.59
## 2 109.95  192.48        333.62       376.15          217.02             307.97
## 3  59.15 2164.62        330.30        86.49           58.56             145.47
## 4 102.51  125.21       1863.11       247.15           75.94             249.64
## 5 160.77  186.79        101.49       749.95           98.49              84.77
## 6  36.97   89.12        287.15       129.02          121.51             399.41
##   tau.Methylhistidine
## 1              160.77
## 2              130.32
## 3               83.93
## 4              254.68
## 5               79.84
## 6               68.72
```

```
str(data)
```

```
## 'data.frame':    77 obs. of  65 variables:
##  $ Patient.ID                : chr  "PIF_178" "PIF_087" "PIF_090" "NETL_005_V1" ...
##  $ Muscle.loss               : chr  "cachexic" "cachexic" "cachexic" "cachexic" ...
##  $ X1.6.Anhydro.beta.D.glucose: num  40.9 62.2 270.4 154.5 22.2 ...
##  $ X1.Methylnicotinamide     : num  65.4 340.4 64.7 53 73.7 ...
##  $ X2.Aminobutyrate          : num  18.7 24.3 12.2 172.4 15.6 ...
##  $ X2.Hydroxyisobutyrate     : num  26.1 41.7 65.4 74.4 83.9 ...
##  $ X2.Oxoglutarate           : num  71.5 67.4 23.8 1199.9 33.1 ...
##  $ X3.Aminoisobutyrate       : num  1480.3 116.8 14.3 555.6 29.7 ...
```

```
##  $ X3.Hydroxybutyrate         : num  56.83 43.82 5.64 175.91 76.71 ...
##  $ X3.Hydroxyisovalerate      : num  10.1 79.8 23.3 25 69.4 ...
##  $ X3.Indoxylsulfate          : num  567 369 665 412 166 ...
##  $ X4.Hydroxyphenylacetate    : num  120.3 432.7 292.9 214.9 97.5 ...
##  $ Acetate                    : num  126.5 212.7 314.2 37.3 407.5 ...
##  $ Acetone                    : num  9.49 11.82 4.44 206.44 44.26 ...
##  $ Adipate                    : num  38.1 327 131.6 144 15 ...
##  $ Alanine                    : num  314 871 464 590 1119 ...
##  $ Asparagine                 : num  159.2 157.6 89.1 273.1 42.5 ...
##  $ Betaine                    : num  110 245 117 279 392 ...
##  $ Carnitine                  : num  265.1 120.3 25 200.3 84.8 ...
##  $ Citrate                    : num  3714 2618 863 13630 854 ...
##  $ Creatine                   : num  196.4 212.7 221.4 85.6 105.6 ...
##  $ Creatinine                 : num  16482 15835 24588 20952 6768 ...
##  $ Dimethylamine              : num  633 608 735 1064 242 ...
##  $ Ethanolamine               : num  645 488 407 821 365 ...
##  $ Formate                    : num  441 252 250 469 114 ...
##  $ Fucose                     : num  337 198.3 186.8 407.5 26.1 ...
##  $ Fumarate                   : num  7.69 18.92 7.1 96.54 19.69 ...
##  $ Glucose                    : num  395 8691 1353 863 6836 ...
##  $ Glutamine                  : num  871 602 302 1686 433 ...
##  $ Glycine                    : num  2039 1108 620 5064 395 ...
##  $ Glycolate                  : num  685.4 652 141.2 70.8 26.6 ...
##  $ Guanidoacetate             : num  154 110 183 103 53 ...
##  $ Hippurate                  : num  4582 1737 4316 757 1153 ...
##  $ Histidine                  : num  925 846 284 1043 327 ...
##  $ Hypoxanthine               : num  97.5 82.3 114.4 223.6 66.7 ...
##  $ Isoleucine                 : num  5.58 8.17 9.3 37.71 40.04 ...
##  $ Lactate                    : num  107 369 750 369 3641 ...
##  $ Leucine                    : num  42.1 77.5 31.5 103.5 101.5 ...
##  $ Lysine                     : num  146.9 284.3 97.5 290 122.7 ...
##  $ Methylamine                : num  52.5 23.6 18.7 48.9 27.9 ...
##  $ Methylguanidine            : num  9.97 7.69 4.66 141.17 5.31 ...
##  $ N.N.Dimethylglycine        : num  23.3 87.4 24.5 40 46.1 ...
##  $ O.Acetylcarnitine          : num  52.98 50.4 5.58 254.68 45.6 ...
##  $ Pantothenate               : num  25.8 186.8 145.5 42.5 74.4 ...
##  $ Pyroglutamate              : num  437 437 713 567 185 ...
##  $ Pyruvate                   : num  21.1 37 29.4 64.1 12.3 ...
##  $ Quinolinate                : num  165.7 73 192.5 86.5 38.1 ...
##  $ Serine                     : num  284 392 296 1249 206 ...
##  $ Succinate                  : num  154.5 244.7 142.6 144 68.7 ...
##  $ Sucrose                    : num  45.1 459.4 160.8 111 75.2 ...
##  $ Tartrate                   : num  97.51 32.79 16.28 837.15 4.53 ...
##  $ Taurine                    : num  1920 1261 4273 1525 469 ...
##  $ Threonine                  : num  184.9 198.3 110 376.1 64.1 ...
##  $ Trigonelline               : num  943.9 208.5 192.5 992.3 86.5 ...
##  $ Trimethylamine.N.oxide     : num  2122 639 1153 1451 172 ...
##  $ Tryptophan                 : num  259.8 83.1 82.3 235.1 103.5 ...
##  $ Tyrosine                   : num  290 167.3 60.3 323.8 142.6 ...
##  $ Uracil                     : num  111 47 31.5 30.6 44.3 ...
##  $ Valine                     : num  86.5 110 59.1 102.5 160.8 ...
##  $ Xylose                     : num  72.2 192.5 2164.6 125.2 186.8 ...
##  $ cis.Aconitate              : num  237 334 330 1863 101 ...
##  $ myo.Inositol               : num  135.6 376.1 86.5 247.2 750 ...
```

```
##  $ trans.Aconitate            : num  51.9 217 58.6 75.9 98.5 ...
##  $ pi.Methylhistidine         : num  157.6 308 145.5 249.6 84.8 ...
##  $ tau.Methylhistidine        : num  160.8 130.3 83.9 254.7 79.8 ...
```

Ahora, separamos los datos en una matriz de expresión (los metabolits) y en metadatos (el ID del paciente y la pérdida de músculo) y nos aseguramos que las filas tengan los mismos nombres.

```
expression_data <- as.matrix(data[, 3:ncol(data)])
rownames(expression_data) <- data$Patient.ID
```

Seleccionamos desde la tercera columna hasta la última (metabolitos) del dataframe. Las dos primeras columnas son las que contiene el IDpaciente y pérdida de músculo.

Ahora creamos los metadatos para las filas y columnas. Las filas contendran la información del paciente (IDpaciente y la pérdida de masa muscular) y las columnas contendran los nombres de los metabolitos

```
row_metadata <- data.frame(Patient_ID = data$Patient.ID, Muscle_loss = data$Muscle.loss)
col_metadata <- data.frame(Metabolite = colnames(expression_data))
```

Una vez preparado todos los datos, creamos el objeto SummarizedExperiment

```
se <- SummarizedExperiment(assays = list(counts = expression_data),
                                rowData = row_metadata,
                                colData = col_metadata)

se
```

```
## class: SummarizedExperiment
## dim: 77 63
## metadata(0):
## assays(1): counts
## rownames(77): PIF_178 PIF_087 ... NETL_003_V1 NETL_003_V2
## rowData names(2): Patient_ID Muscle_loss
## colnames(63): X1.6.Anhydro.beta.D.glucose X1.Methylnicotinamide ...
##   pi.Methylhistidine tau.Methylhistidine
## colData names(1): Metabolite
```

Los resultados indican lo siguiente:

- Class: SummarizedExperiment
- Dim: contiene 77 filas y 63 columnas
- Metadata(0): No contiene metadatos adicionales para el objeto.
- Assays(1): counts. Existe un conjunto de datos que contiene los valores de expresión de los metabolitos, almacenados en expression_data.
- RowNames: Son los nombres de las filas correspondientes a los identificadores de los pacientes.
- RowData names: Son los metadatos asociados a las filas.
- Colnames: Representan los metabolitos. Son los nombres de las columnas.
- colData names: Describre las columnas.

Una vez tenemos el SummarizedExperiment lo normalizamos para que los resultados del análisis estadístico sea más representativo. Utilizaremos el paquete POMA, que ofrece un conjunto de herramientas integral

diseñado para el análisis de datos ómicos. Este paquete aprovechará la clase estandarizada SummarizedExperiment de Bioconductor, garantizando la integración y compatibilidad con las herramientas de este.

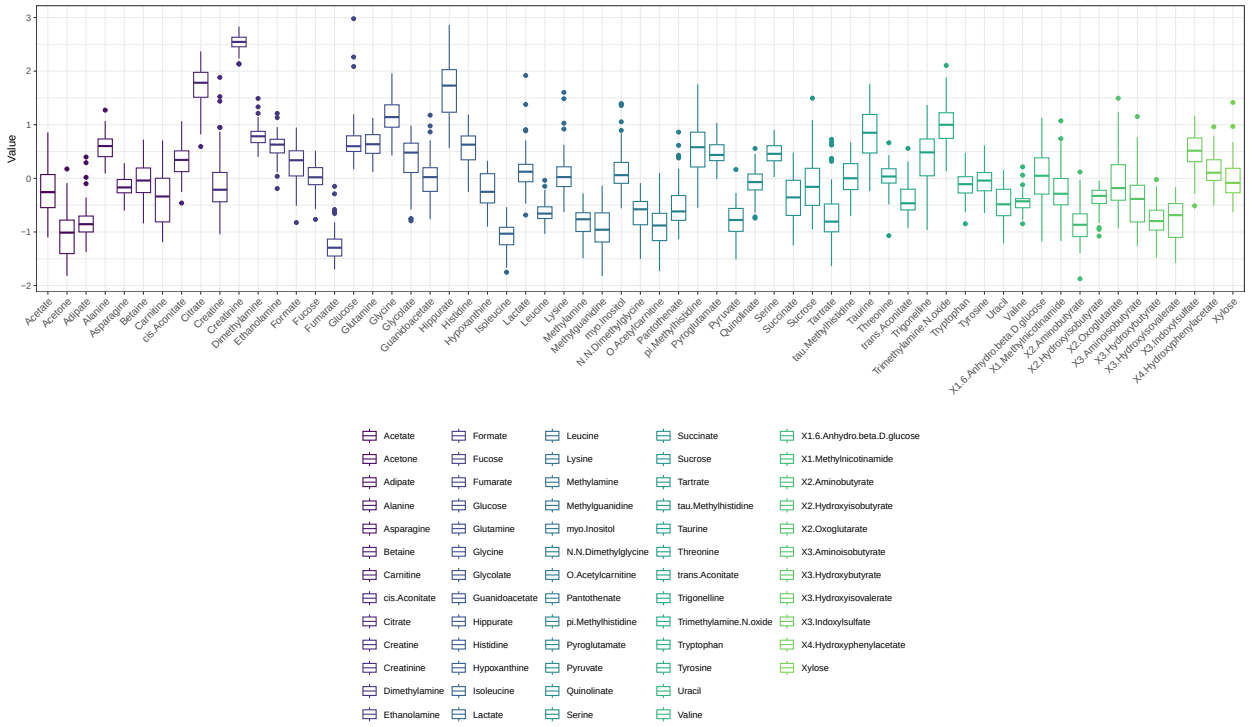
El método de normalización "log_pareto" es específico para el análisis de datos de metabolómica. Se puede utilizar cuando los datos tienen características de distribución sesgada y escalas muy variadas, lo que es bastante común en datos de metabolitos.

```
normalized <- se %>%
  PomaNorm(method = "log_pareto")
```

Podemos representar diferentes gráficos para observar los datos una vez normalizados.

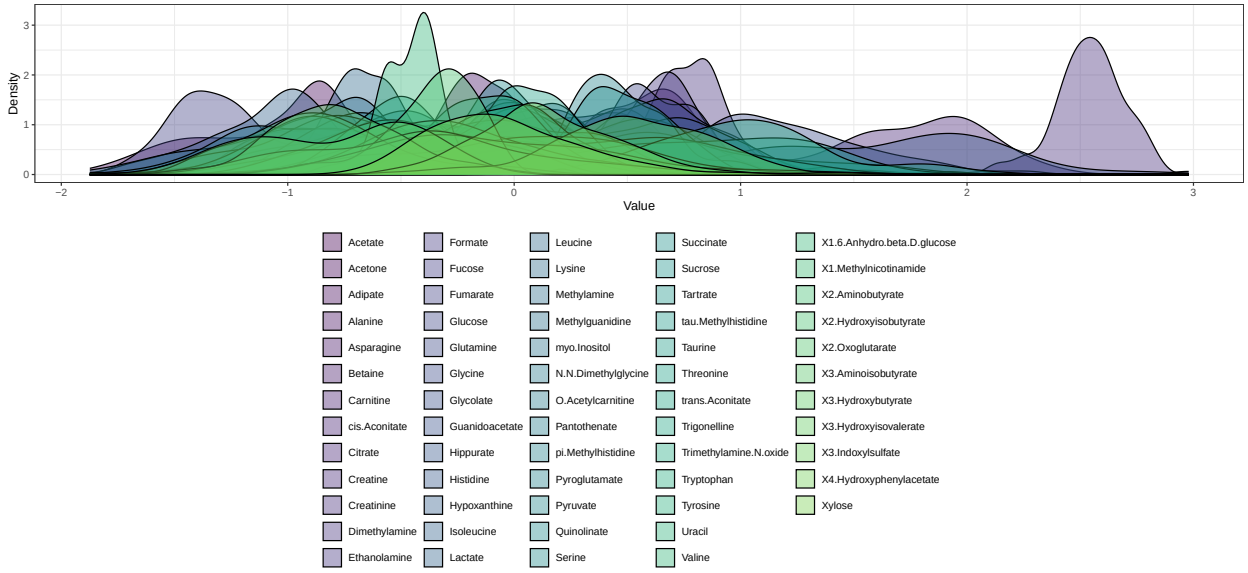
Este gráfico nos muestra boxplots para cada metabolito en el conjunto de datos normalizados. Resulta útil para observar la distribución de los valores de cada metabolito.

```
PomaBoxplots(normalized)
```



Este gráfico de densidad muestra la distribución de los valores normalizados de los metabolitos en el conjunto de datos. Permite observar cómo estan distribuidos de forma continua y verificar la homogeneidad en las distribuciones

```
PomaDensity(normalized)
```

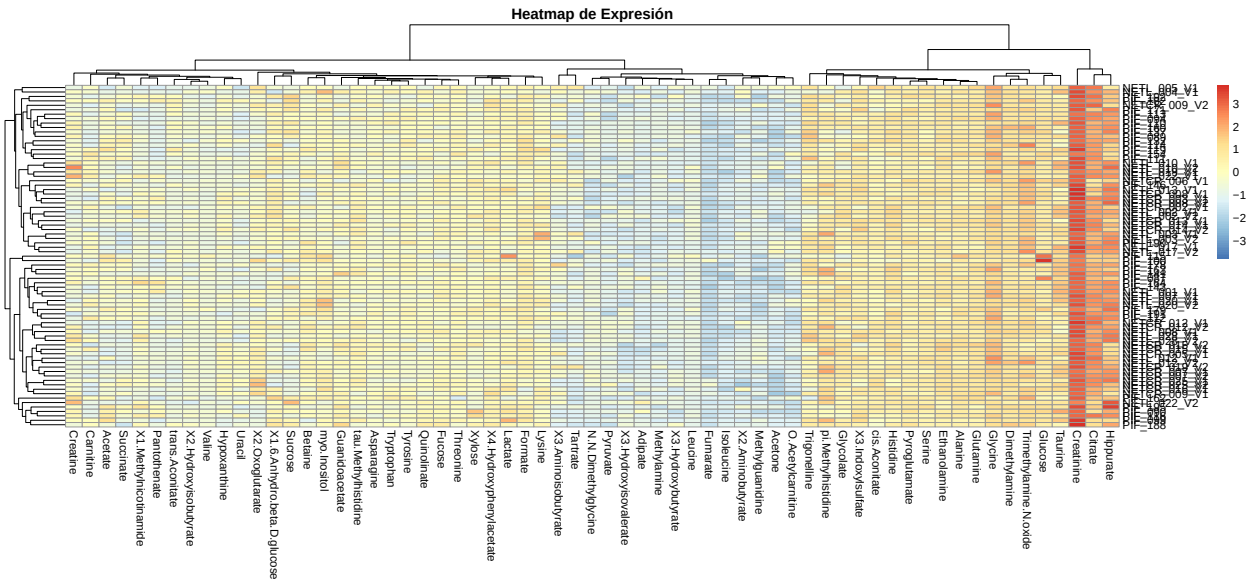


Un gráfico heatmap es útil para poder visualizar la expresión de metabolitos a traves de muestras para observar patrones de agrupamiento entre las muestras y las características (en este caso metabolitos).

Los colores identifican los niveles de expresión. En este caso, colores más cálidos (mas saturados) indican un nivel alto de expresión mientras que colores mas frios (menos saturado) indican niveles bajos de expresión.

Los dendrogramas ayudan a la visualización de agrupaciones jerárquicas de muestras o metabolitos. Cuanto mas cerca estan dos muestras en el dendrograma, mas similares son en términos de sus perfiles de expresion de metabolitos.

```
pheatmap(assay(normalized), scale = "row", clustering_distance_rows = "euclidean",
clustering_distance_cols = "euclidean", main = "Heatmap de Expresión")
```

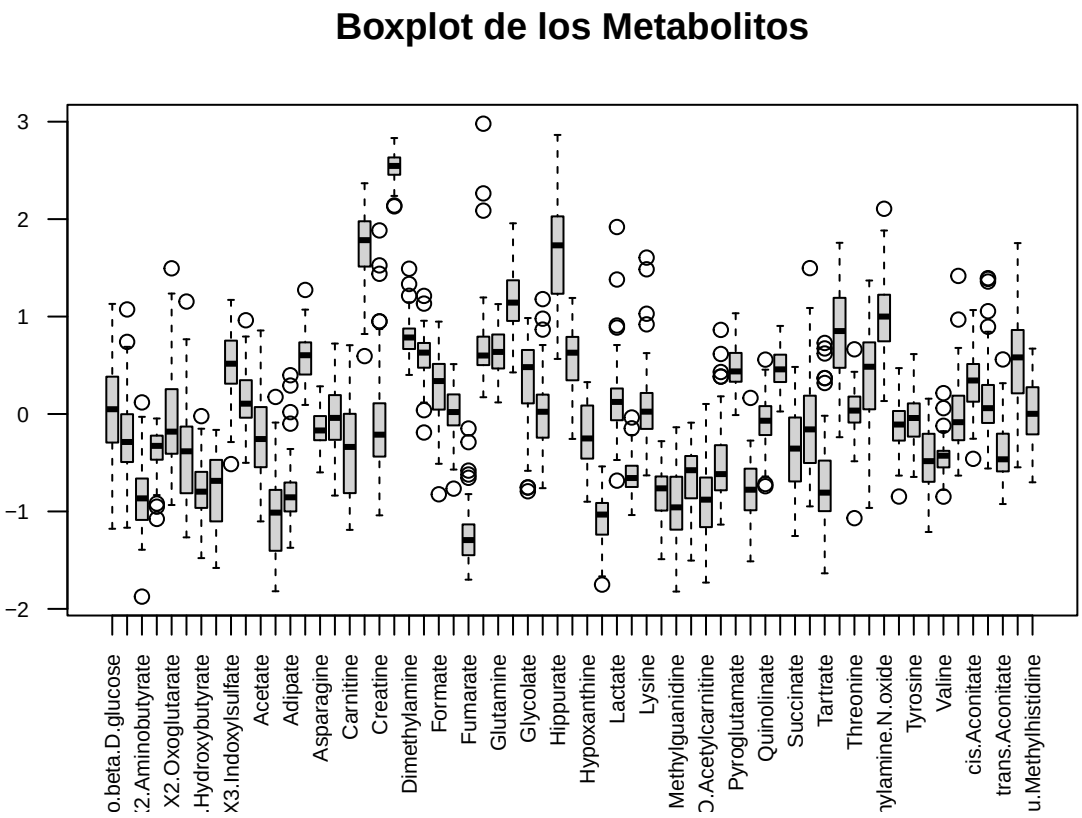


En el gráfico boxplot, cada caja representa la distribución de valores de cada metabolito para todas las muestras. En el eje de las Y se muestra los valores de expresión de los metabolitos despues de la normalización (rango de -2 a 2)

Este gráfico nos proporciona una visión general de la distribución de cada metabolito. Los que la caja esté centrada alrededor de 0 están cerca del valor promedio de la mayoria de las muestras. En cambio, las que estan descentradas tienen un sesgo en su abundancia.

Metabolitos con muchos outliers sugieren variabilidad en las muestras y que estas son muy diferentes con el resto.

```
boxplot(assay(normalized), main = "Boxplot de los Metabolitos", las = 2, cex.axis = 0.7)
```



El heatmap de la Matriz de Correlación de Metabolitos representa la magnitud y la dirección de la correlación entre los metabolitos a partir de los colores. De manera general, los colores oscuros (tonos mas rojizos) indican correlaciones fuertes y los colores claros (tonos mas suaves) indican correlaciones débiles o cercanas a 0.

Las correlaciones fuertes sugieren que los metabolitos tienden a cambiar de manera simiar en todas las muestras. Este hecho da a entender que podria existir una relación biológica o funcional.

Las correlaciones débiles sugieren que los metabolitos no estan relacionados entre si y no presentan variación conjunta significativa en las muestras que se han analizado.

Los bloques de color a lo largo de la diagonal representan grupos de metabolitos que están altamente correlacionados.

```
cor_matrix <- cor(assay(normalized))
heatmap(cor_matrix, main = "Matriz de Correlación de Metabolitos")
```

Matriz de Correlación de Metabolitos