Exploracin de los datos - PEC1

Adrián Parrila Mesas

```
library(metabolomicsWorkbenchR)
library(SummarizedExperiment)
library(pheatmap)
library(RColorBrewer)
library(ggplot2)
library(patchwork)
```

El repositorio del estudio se encuentra en: https://www.metabolomicsworkbench.org/data/DRCCMetadata.php?Mode=Study&StudyID=ST002797&StudyType=MS&ResultType=1

Cargo los datos del estudio en formato *SummarizedExperiment* utilizando la libreria metabolomicsWorkbenchR.

```
df <- do_query(context = 'study',
               input_item = 'study_id',
               input_value = 'ST002797',
               output_item = 'SummarizedExperiment')

df <- df[[1]]
```

df es una lista de dos objetos "SummarizedExperiment" que corresponden a dos análisis distintos. Para la exploración posterior escojo solamente el primero.

Exploro los metadatos del estudio, contenidos en el slot ColData.

```
df@colData[1:8,]
```

```
## DataFrame with 8 rows and 6 columns
##         local_sample_id    study_id  sample_source mb_sample_id
##             <character> <character>    <character>  <character>
## C10_215         C10_215    ST002797 Amniotic fluid     SA300456
## C1_160           C1_160    ST002797 Amniotic fluid     SA300454
## C2_171           C2_171    ST002797 Amniotic fluid     SA300457
## C3_183           C3_183    ST002797 Amniotic fluid     SA300462
## C4_186           C4_186    ST002797 Amniotic fluid     SA300461
## C5_189           C5_189    ST002797 Amniotic fluid     SA300463
## C6_190           C6_190    ST002797 Amniotic fluid     SA300460
## C7_190           C7_190    ST002797 Amniotic fluid     SA300458
##                   raw_data    Group
##                <character> <factor>
## C10_215 C10_215.mzdata.xml  Control
## C1_160   C1_160.mzdata.xml  Control
## C2_171   C2_171.mzdata.xml  Control
## C3_183   C3_183.mzdata.xml  Control
```

```
## C4_186   C4_186.mzdata.xml  Control
## C5_189   C5_189.mzdata.xml  Control
## C6_190   C6_190.mzdata.xml  Control
## C7_190   C7_190.mzdata.xml  Control
```

El numero de muestras del estudio y la distribución de los grupos es:

```
table(df@colData$Group)
```

```
##
## Control    TTTS
##      10      22
```

Se observa que hay el doble de individuos en el grupo TTTS que en el control.

Exploro ahora la distribución de los datos.

La lista de todos los metabolitos identificados se puede acceder con rowData()

```
rowData(df)[1:10,]
```

```
## DataFrame with 10 rows and 3 columns
##                 metabolite_name metabolite_id            refmet_name
##                     <character>   <character>            <character>
## ME725430 10(E)-Heptadecenoic ..      ME725430
## ME725432 "10(Z),13(Z)-Nonadec..      ME725432
## ME725436 10(Z)-Heptadecenoic ..      ME725436 10Z-Heptadecenoic acid
## ME725425     11-Dodecenoic Acid      ME725425
## ME725435  11(E)-Eicosenoic Acid      ME725435     trans-Gondoic acid
## ME725431 (11E)-Octadecenoic a..      ME725431    trans-Vaccenic acid
## ME725428    12-Tridecenoic Acid      ME725428
## ME725427 "12(Z),15(Z)-Heneico..      ME725427
## ME725438 "1,3-Dipalmitoylglyc..      ME725438
## ME725423       13 Retinoic Acid      ME725423
```

Miro el número de assays que tiene el objeto df

```
df@assays
```

```
## An object of class "SimpleAssays"
## Slot "data":
## List of length 1
```

Como solo contiene 1 assay, se puede acceder directamente a los datos con el metodo assay(). Guardo los datos en una nueva variable para acceder mejor a ellos.

```
assay(df)[1:8,]
```

```
##       C10_215      C1_160     C2_171     C3_183     C4_186     C5_189
##         <num>       <num>      <num>      <num>      <num>      <num>
## 1: 3335628.51 14274066.31 7228407.05 6665808.23 3441411.38 6370853.21
## 2:         NA          NA   15987.69   45216.32         NA         NA
```

```
## 3: 3327976.45 14263540.89 7221078.59 6656569.68 3431481.81 6360071.48
## 4:   44502.46          NA         NA         NA   67963.91         NA
## 5:   72145.94    28889.79   59580.27   79567.52   40750.05   64308.62
## 6:  459632.81  2460093.24 1158917.83 1072366.23  515275.10  742939.28
## 7:  563787.71  2344928.51 1855778.31 1853539.00  613105.74 2161283.37
## 8:   10119.58    10119.58   10119.58   10119.58   10119.58   10119.58
##        C6_190     C7_190     C8_194     C9_213   T10_175R   T11_181R   T12_184R
##         <num>      <num>      <num>      <num>      <num>      <num>      <num>
## 1: 3376648.10 6276305.90 2983504.01 6617537.47 6317255.37 7168472.08 4506512.49
## 2:         NA   11600.19  115819.97   39185.17   52429.42   49095.99   39476.34
## 3: 3366183.12 6261733.31 2971964.70 6611861.33 6280994.96 7128147.81 4476850.17
## 4:   47423.09         NA   31284.19         NA   70994.39   28985.76  101733.19
## 5:   88358.28   62726.53   68169.21  113559.15  990453.52  995825.67  967012.75
## 6:  341852.16  513685.31  324848.92  911701.38   42476.34  147385.64   28905.23
## 7:  966733.75 2236157.56  877860.12 1642000.77  891045.17  825495.94  801319.71
## 8:   10119.58   10119.58   10119.58   10119.58  369623.97  238038.30  220885.14
##      T13_231R   T14_253R     T1_48D   T15_258D    T16_262D   T17_281D   T18_287D
##         <num>      <num>      <num>     <num>       <num>     <num>      <num>
## 1: 6075918.67 3990135.83 5206353.59 8544742.1 8016432.203 7174683.7 4267426.27
## 2:   14558.53   75800.89   31164.01        NA   27872.238   31711.7         NA
## 3: 6064046.33 3976803.76 5195284.21 8529782.7 7974480.821 7137454.5 4252914.62
## 4:         NA         NA   99701.36        NA          NA        NA         NA
## 5:   69339.62   75951.72   23304.98  805423.7  716528.720  938153.4   56644.41
## 6:  885715.37  653312.26  769789.91  512767.0    6183.842   46018.4  509837.21
## 7: 1619326.68  882084.76 1137977.68  834633.6 1620032.970  732821.6 1165783.97
## 8:   20239.16         NA         NA  209915.8  272422.174  214537.1   29794.46
##       T19_317R   T20_322R   T21_360R   T22_421D      T2_52D   T3_106DR
##          <num>      <num>      <num>      <num>       <num>      <num>
## 1: 5802176.072 2168401.31 8721270.83 6768974.80 11606337.63 7588719.94
## 2:   27044.140         NA   35166.69   51605.99    26526.11   57501.88
## 3: 5774062.273 2156002.51 8677625.31 6729927.38 11567983.90 7556486.42
## 4:          NA         NA   58868.07         NA    53166.18         NA
## 5:  743450.094   56897.05  968604.68  939118.02   828383.52 1005857.96
## 6:    6212.303  255460.32   23955.00   35995.67   872359.17   63862.10
## 7: 1127194.830  441732.59  953484.10  866327.38   885074.76  718234.42
## 8:  164549.487         NA  177572.40  118431.45   276558.58  413940.34
##       T4_109R    T5_114R   T6_118DR   T7_121DR    T8_161R    T9_164D
##         <num>      <num>      <num>      <num>      <num>      <num>
## 1: 3230822.48 7566412.25 6063113.44 7762695.30 5813751.87 7320237.49
## 2:         NA   63398.77   62974.91   31721.96   80360.88   39269.82
## 3: 3217472.69 7527577.96 6031568.74 7719748.69 5787201.60 7278496.78
## 4:   39924.47         NA         NA   66884.64   44946.43   46417.27
## 5:   87752.68 1044434.45 1013192.86 1057581.35 1086446.31  854167.77
## 6:  490211.89   46709.93   69987.14  114734.83   76990.67   33314.27
## 7:  673806.27  801187.92  621366.19  610295.96  590995.71 1026098.36
## 8:         NA  472077.65  341716.69  246361.23  280734.40  371780.49
```

```
data <- as.data.frame(assay(df))
```

Cuantifico el número de NAs y los imputo por 1 para evitar problemas durante la normalización.

```
sum(is.na(data))
```

```
## [1] 376
```

```r
data[is.na(data)] <- 1
```

```r
dim(data)
```

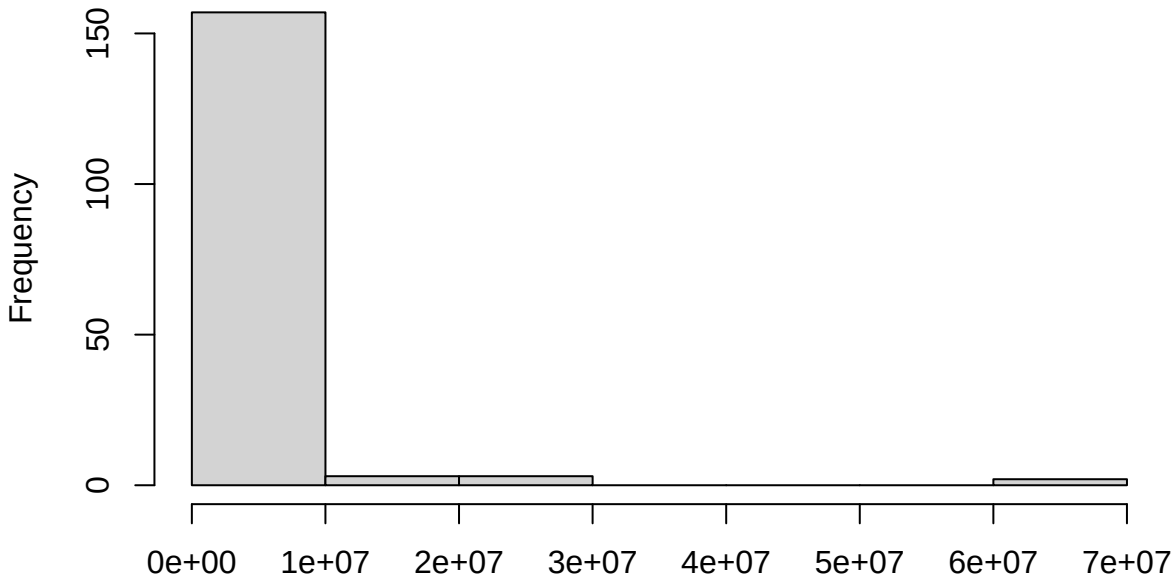
```
## [1] 165  32
```

```r
str(data)
```

```
## 'data.frame':    165 obs. of  32 variables:
##  $ C10_215 : num  3335629 1 3327976 44502 72146 ...
##  $ C1_160  : num  14274066 1 14263541 1 28890 ...
##  $ C2_171  : num  7228407 15988 7221079 1 59580 ...
##  $ C3_183  : num  6665808 45216 6656570 1 79568 ...
##  $ C4_186  : num  3441411 1 3431482 67964 40750 ...
##  $ C5_189  : num  6370853 1 6360071 1 64309 ...
##  $ C6_190  : num  3376648 1 3366183 47423 88358 ...
##  $ C7_190  : num  6276306 11600 6261733 1 62727 ...
##  $ C8_194  : num  2983504 115820 2971965 31284 68169 ...
##  $ C9_213  : num  6617537 39185 6611861 1 113559 ...
##  $ T10_175R: num  6317255 52429 6280995 70994 990454 ...
##  $ T11_181R: num  7168472 49096 7128148 28986 995826 ...
##  $ T12_184R: num  4506512 39476 4476850 101733 967013 ...
##  $ T13_231R: num  6075919 14559 6064046 1 69340 ...
##  $ T14_253R: num  3990136 75801 3976804 1 75952 ...
##  $ T1_48D  : num  5206354 31164 5195284 99701 23305 ...
##  $ T15_258D: num  8544742 1 8529783 1 805424 ...
##  $ T16_262D: num  8016432 27872 7974481 1 716529 ...
##  $ T17_281D: num  7174684 31712 7137454 1 938153 ...
##  $ T18_287D: num  4267426 1 4252915 1 56644 ...
##  $ T19_317R: num  5802176 27044 5774062 1 743450 ...
##  $ T20_322R: num  2168401 1 2156003 1 56897 ...
##  $ T21_360R: num  8721271 35167 8677625 58868 968605 ...
##  $ T22_421D: num  6768975 51606 6729927 1 939118 ...
##  $ T2_52D  : num  11606338 26526 11567984 53166 828384 ...
##  $ T3_106DR: num  7588720 57502 7556486 1 1005858 ...
##  $ T4_109R : num  3230822 1 3217473 39924 87753 ...
##  $ T5_114R : num  7566412 63399 7527578 1 1044434 ...
##  $ T6_118DR: num  6063113 62975 6031569 1 1013193 ...
##  $ T7_121DR: num  7762695 31722 7719749 66885 1057581 ...
##  $ T8_161R : num  5813752 80361 5787202 44946 1086446 ...
##  $ T9_164D : num  7320237 39270 7278497 46417 854168 ...
```

Exploro una de las variables para valorar si hay que hacer alguna transformación.

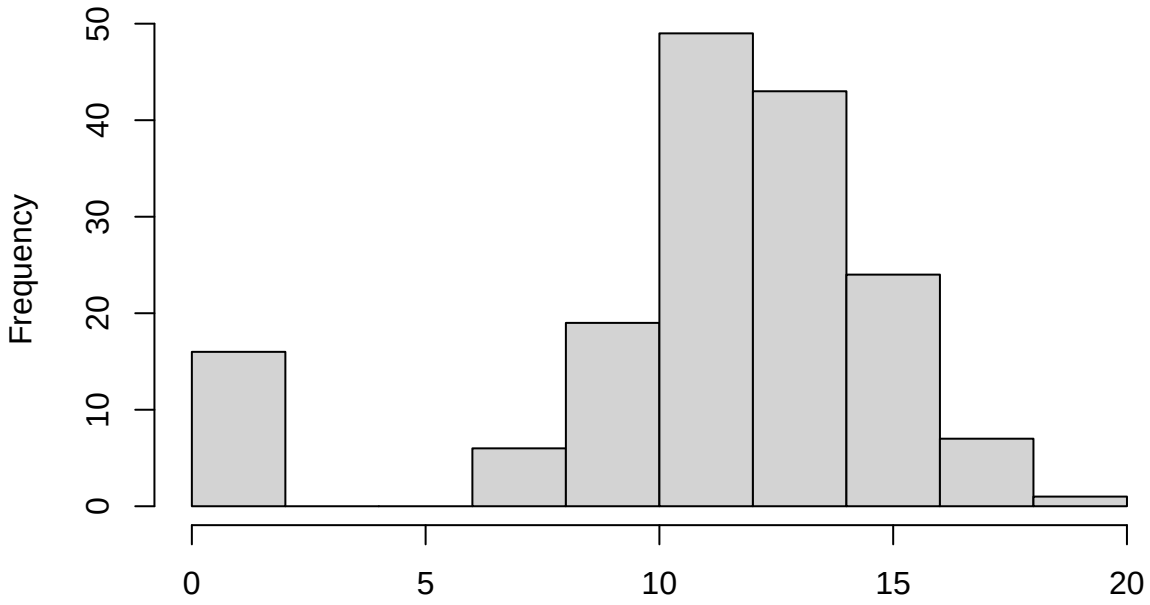
```r
hist(data$C10_215, main = '', xlab = '')
```



Como el rango de los datos es muy grande, les aplico una transformación logaritmica.

```
data_norm <- log(data)
```

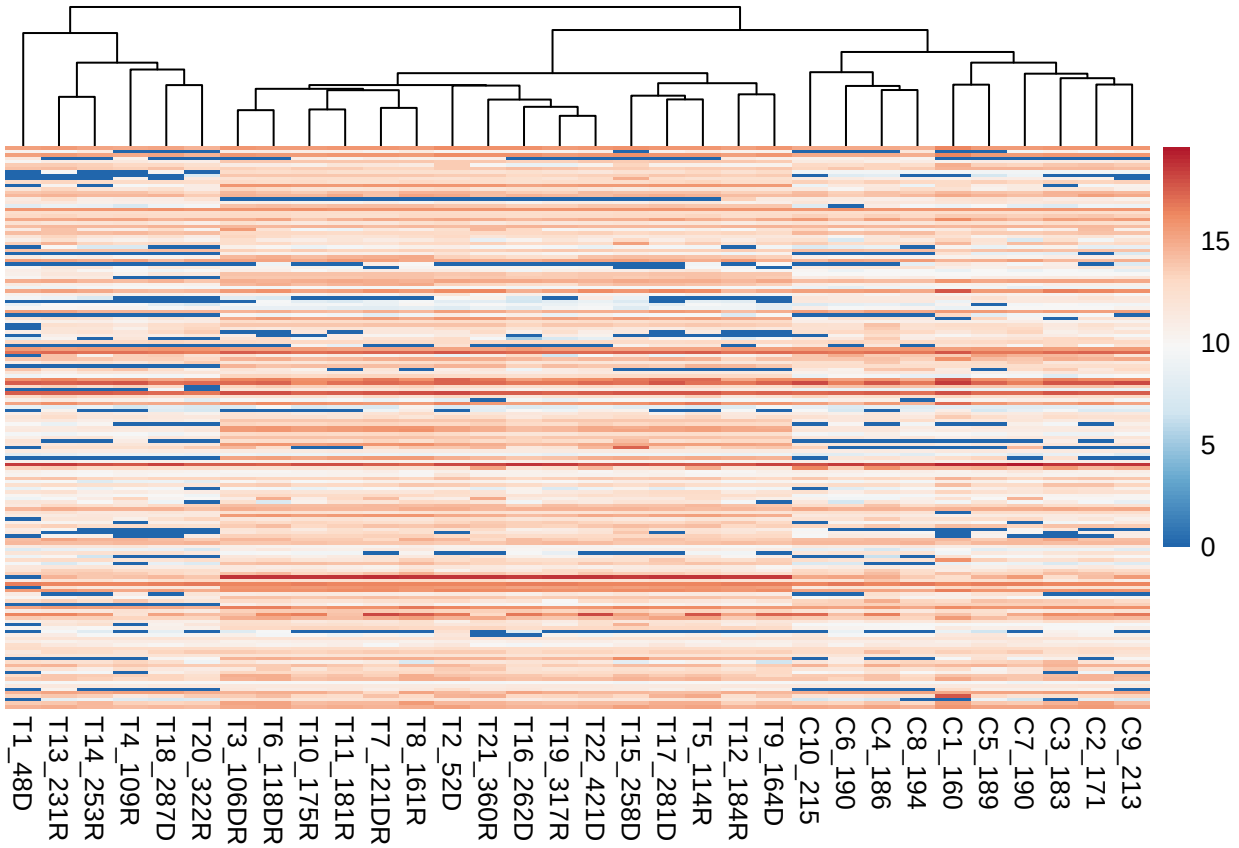
```
hist(data_norm$C10_215, main = '', xlab = '')
```



Realizo un heatmap con agrupamiento para tener una visión general de los resultados. Como cada fila representa un metabolito, agrego los nombres de los metabolitos identificados como rownames del dataset.

```
rownames(data_norm) <- rowData(df)$metabolite_name
```

```
pheatmap(data_norm,
         show_rownames = F,
         cluster_rows = F,
         cluster_cols = T,
         clustering_method = "complete",
         color = colorRampPalette(rev(brewer.pal(n = 7, name = "RdBu")))(100),
         )
```



El clustering del heatmap muestra una separación clara entre ambos grupos. Para verificarlo, realizo un análisis de componentes principales.

Primero transpongo la matriz de los datos para tener las muestras como filas y la variables de columnas.

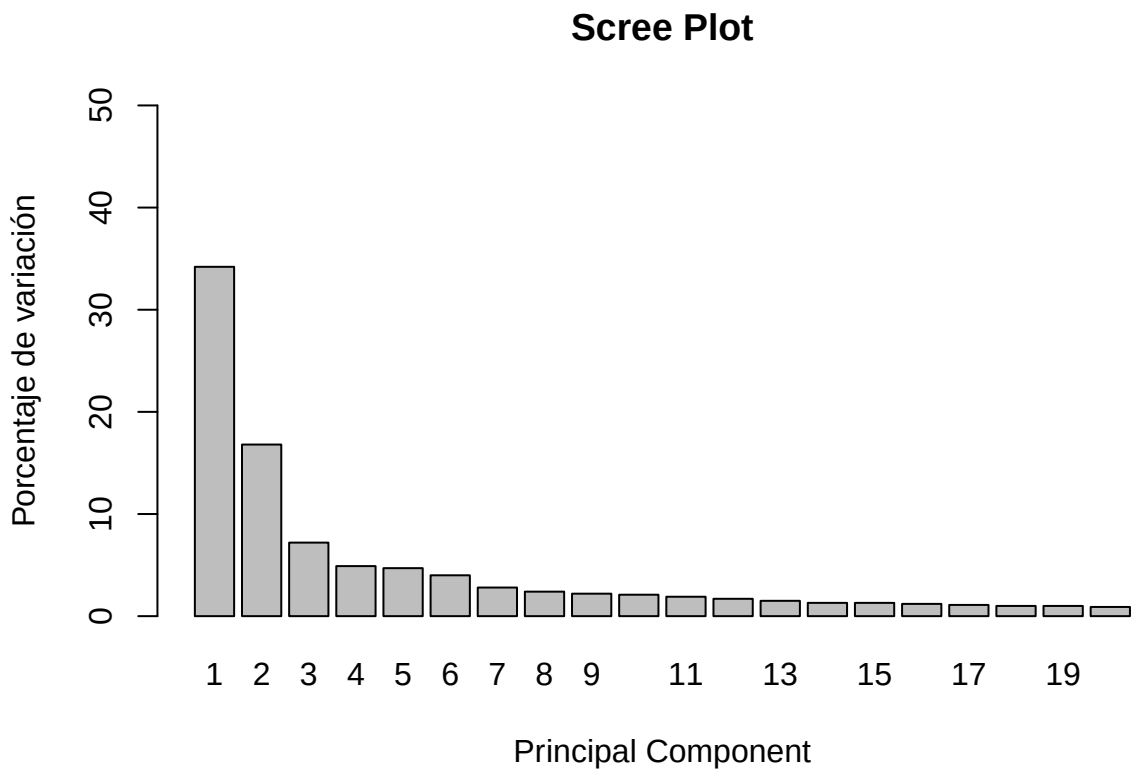
```
data_norm_pca <- t(data_norm)

pca <- prcomp(data_norm_pca, scale = TRUE)
```

Calculo la varianza y el porcentaje que explica cada componente.

```
pca.var <- pca$sdev^2
pca.var.per <- round(pca.var/sum(pca.var)*100, 1)
```

```
scree_plot <- barplot(pca.var.per[1:20], main="Scree Plot", xlab="Principal Component", ylab="Porcentaj
```



Creo un dataframe con los resultados de los 3 primero componentes del PCA y el grupo al que pertenecen las muestras.

```
group <- c(rep(c("Control"), 10), rep("TTTS", 22))
```

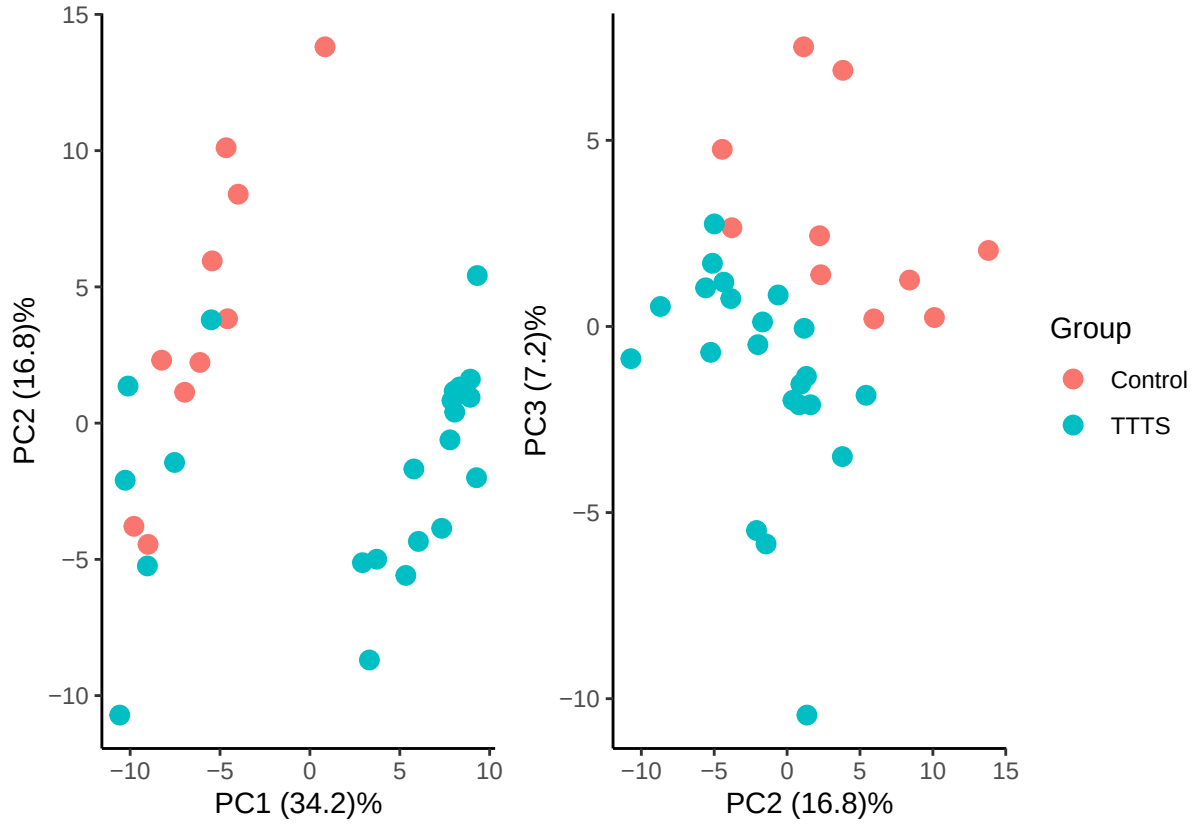
```
pca.data <- data.frame(Sample=rownames(pca$x),
                       Group= group,
                       PC1=pca$x[,1], #PCA1
                       PC2=pca$x[,2], #PCA2
                       PC3=pca$x[,3]) #PCA3

pca.data$Group <- as.factor(pca.data$Group)
```

```
p1 <- ggplot(data=pca.data, aes(x=PC1, y=PC2)) +
  geom_point(aes(color= Group), size = 3)+
  xlab(paste("PC1 (", pca.var.per[1],")%", sep="")) +
  ylab(paste("PC2 (", pca.var.per[2],")%", sep="")) +
  theme_classic() +
  theme(legend.position="none")

p2 <- ggplot(data=pca.data, aes(x=PC2, y=PC3)) +
  geom_point(aes(color= Group), size = 3)+
  xlab(paste("PC2 (", pca.var.per[2],")%", sep="")) +
  ylab(paste("PC3 (", pca.var.per[3],")%", sep="")) +
  theme_classic()
```

```
p1+p2
```



Se observa una clara separación entre ambos grupos el la primera componente principal. Obtengo los loadings para ver que metabolítos están influyendo más en esta separación.

```
loading_scores <- pca$rotation[,1]

metab_scores <- abs(loading_scores) # me quedo con las magnitudes
metab_scores_ranked <- sort(metab_scores, decreasing=TRUE)

top_10_metabs <- names(metab_scores_ranked[1:10])

pca$rotation[top_10_metabs,1]
```

```
##         4-Imidazoleacetic Acid             Hydrocinnamic Acid
##                     -0.1284947                      0.1271937
##           Docosahexaenoic Acid                  Nervonic Acid
##                      0.1247929                      0.1247638
##        3-Aminoisobutyric Acid Eicosatrienoic Acid (20:3 N-3)
##                     -0.1246159                      0.1245886
##    Dihomo-Gamma-Linolenic Acid           Mead Acid (20:3 N-9)
##                      0.1245746                      0.1245331
##              Retinyl Palmitate   "5,8,11-Eicosatriynoic Acid"
##                      0.1244252                      0.1241605
```

Se observa que el 4-Imidazoleacetic Acid es el que guía la separación del grupo control hacia la izquierda, aunque no hay una gran diferencia respecto al resto de moleculas en el top 10.