PEC1_cachexia_github

Gabriel Regueira Huguet

2025-03-27

1- SELECCIONEM UN DATASET DE METABOLÒMICA:

La *cachexia* és un síndrome metabòlic complex, que es mostra habitual en pacients amb càncer. Aquest síndrome es caracteritza per una pèrdua de massa musculari/o greixosa, inflamació sistèmica, alteracions hormonals i en el metabolisme enegètic. Aquesta pèrdua de massa muscular es tradueix a un catabolisme muscular accelerat, que proboca l'alliberament de aminoàcids com la valina, leucina, alanina, etc. Aquest síndrome proboca que el pacient es trobi en un estat de semi-fam metabòlica, que proboca que hi hagi una demanda energètica elevada i alguns metabòlits intermedis del metabolisme energètic s'acumulen (3-hydrpxybutyrate, pyroglutamate, glutamine).

```
#Importem directament les dades crues (raw) desde el repositori de github que proporciona l'enunciat:
url_github <- "https://raw.githubusercontent.com/nutrimetabolomics/metaboData/refs/heads/main/Datasets/2
human_cachexia_data <- read.csv(url_github, header = TRUE, sep = ",") #Separació per comes tai i com es
```

```
View(human_cachexia_data)
head(human_cachexia_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
```

Observem un dataset on les files són les mostres (pacients) i les columnes són els metabòlits analitzats. També podem observar que hi ha una variable que separa els pacients en dos grups, pacients amb *cachexia* i pacients control.

2- CREAR UN OBJECTE *SUMMARIZEDEXPERIMENT*

Per a crear un objecte *SummarizedExperiment* necessito una matriu de dades quantitatives (amb les concentracions de metabòlits per pacient) i un *colData* amb informació de les mostres, que en aquest cas és el tipus de grup que pertany cada pacient (*cachexia/control*).

```
#Matriu de dades numèriques
assay_data1_ <- human_cachexia_data[, -(1:2)] #Eliminem les dues primeres columnes
rownames(assay_data1_) <- human_cachexia_data$Patient.ID #Assignem com a nom de fila els identificadors
assay_data1 <- as.matrix(assay_data1_)
View(assay_data1) #Observem que tenim el nom de les files assignats als identificadors dels pacients (P
```

```
#colData: Informació sobre les mostres (pacients): Muscle.loss
col_data1 <- data.frame(
  Patient.ID = human_cachexia_data$Patient.ID, #Agafem els pacients,
  Muscle.loss = human_cachexia_data$Muscle.loss #I el grup al que pertanyen
)
rownames(col_data1) <- human_cachexia_data$Patient.ID #Assignem coma nom de fila els indentificadors co
View(col_data1)
```

Abans de fer el *SummarizedExperiment*, necessitem tenir una matriu numèrica on les mostres estiguin com a columnes en comptes de com a files, de la mateixa manera els metabòlits com a files en comptes de com a columnes:

```
#Trasposem la matriu per tenir metabolits (files) x pacients (columnes)
assay_data1t <- t(assay_data1)
View(assay_data1t)
colnames(assay_data1t) #Automàticament ja cambia rownames per colnames quan trasposem la matriu
```

```
##  [1] "PIF_178"      "PIF_087"      "PIF_090"      "NETL_005_V1"  "PIF_115"
##  [6] "PIF_110"      "NETL_019_V1"  "NETCR_014_V1" "NETCR_014_V2" "PIF_154"
## [11] "NETL_022_V1"  "NETL_022_V2"  "NETL_008_V1"  "PIF_146"      "PIF_119"
## [16] "PIF_099"      "PIF_162"      "PIF_160"      "PIF_113"      "PIF_143"
## [21] "NETCR_007_V1" "NETCR_007_V2" "PIF_137"      "PIF_100"      "NETL_004_V1"
## [26] "PIF_094"      "PIF_132"      "PIF_163"      "NETCR_003_V1" "NETL_028_V1"
## [31] "NETL_028_V2"  "NETCR_013_V1" "NETL_020_V1"  "NETL_020_V2"  "PIF_192"
## [36] "NETCR_012_V1" "NETCR_012_V2" "PIF_089"      "NETCR_002_V1" "PIF_179"
## [41] "PIF_114"      "NETCR_006_V1" "PIF_141"      "NETCR_025_V1" "NETCR_025_V2"
## [46] "NETCR_016_V1" "PIF_116"      "PIF_191"      "PIF_164"      "NETL_013_V1"
## [51] "PIF_188"      "PIF_195"      "NETCR_015_V1" "PIF_102"      "NETL_010_V1"
## [56] "NETL_010_V2"  "NETL_001_V1"  "NETCR_015_V2" "NETCR_005_V1" "PIF_111"
## [61] "PIF_171"      "NETCR_008_V1" "NETCR_008_V2" "NETL_017_V1"  "NETL_017_V2"
## [66] "NETL_002_V1"  "NETL_002_V2"  "PIF_190"      "NETCR_009_V1" "NETCR_009_V2"
```

```
## [71] "NETL_007_V1"  "PIF_112"      "NETCR_019_V2" "NETL_012_V1"  "NETL_012_V2"
## [76] "NETL_003_V1"  "NETL_003_V2"
```

Ara ja tenim la matriu de dades numèriques llesta per a fer *SummarizedExperiment*

```
library(SummarizedExperiment)
```

```
## Cargando paquete requerido: MatrixGenerics

## Cargando paquete requerido: matrixStats

##
## 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

##
## Adjuntando el paquete: 'IRanges'

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

## Cargando paquete requerido: GenomeInfoDb

## 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
```

```
cachexia_se <- SummarizedExperiment(
  assays = list(counts = assay_data1t), #Matriu de dades
  colData = col_data1 #Grup per pacient
)
cachexia_se
```

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

Una vegada creat el *SummarizedExperiment*, el guardarem en un arxiu en format .Rda com indica l'enunciat:

```
save(cachexia_se, file = "cachexia_se.rda") #Guardem l'arxiu al repositori
```

Diferències *ExpressionSet* i *SummarizedExperiment* :

ExpressionSet ha estat durant molt temps el format clàssic per analitzar dades de miacroarrays, només admet una única matriu de dades (*exprs*). És molt útil, però està pensat per un tipus específic de dades i no té tanta flexibilitat. En canvi l'objecte *SummarizedExperiment* és més potent, ja que és capaç de gestionar més tipus de dades (comptes, intensitats, etc) i pot contenir múltiples matrius (en *assays*) i és compatible amb dades més complexes, és l'OOP estàndard actual per a estudis RNa-seq, proteòmica i metabolòmica. Tots dos objectes són molt útils per organitzar les dades de manera integrada i sincronitzada, però *SummarizedExperiment* ho fa amb més flexibilitat i amb una estructura més moderna.

3- ANÀLISIS EXPLORATORI:

```
#S'ha fet un resum estadístic per a cada metabòlit (mínim, mitjana, màxim, etc), posem els 5 primer com
apply(assay(cachexia_se)[1:5, ], 1, summary) #Resum numèric dels metabòlits (5 primers)
```

```
##         X1.6.Anhydro.beta.D.glucose X1.Methylnicotinamide X2.Aminobutyrate
## Min.                        4.7100              6.42000          1.28000
## 1st Qu.                    28.7900             15.80000          5.26000
## Median                     45.6000             36.60000         10.49000
## Mean                      105.6304             71.57364         18.15974
## 3rd Qu.                   141.1700             73.70000         19.49000
## Max.                      685.4000           1032.77000        172.43000
##         X2.Hydroxyisobutyrate X2.Oxoglutarate
## Min.                 4.85000         5.5300
## 1st Qu.             15.80000        22.4200
## Median              32.46000        55.1500
## Mean                37.25065       145.0871
## 3rd Qu.             54.60000        92.7600
## Max.                93.69000      2465.1300
```

```
dim(cachexia_se)
```

```
## [1] 63 77
```

En aquest cas, es pot deduïr que les variables són 63 concentracions de metabòlits analitzats en la orina de 77 individus. Totes les variables, doncs, són numèriques menys la variables grup de *Muscle.loss*, que es la que determina quin pacient té *cachexia* i quin pertany al grup *control*.

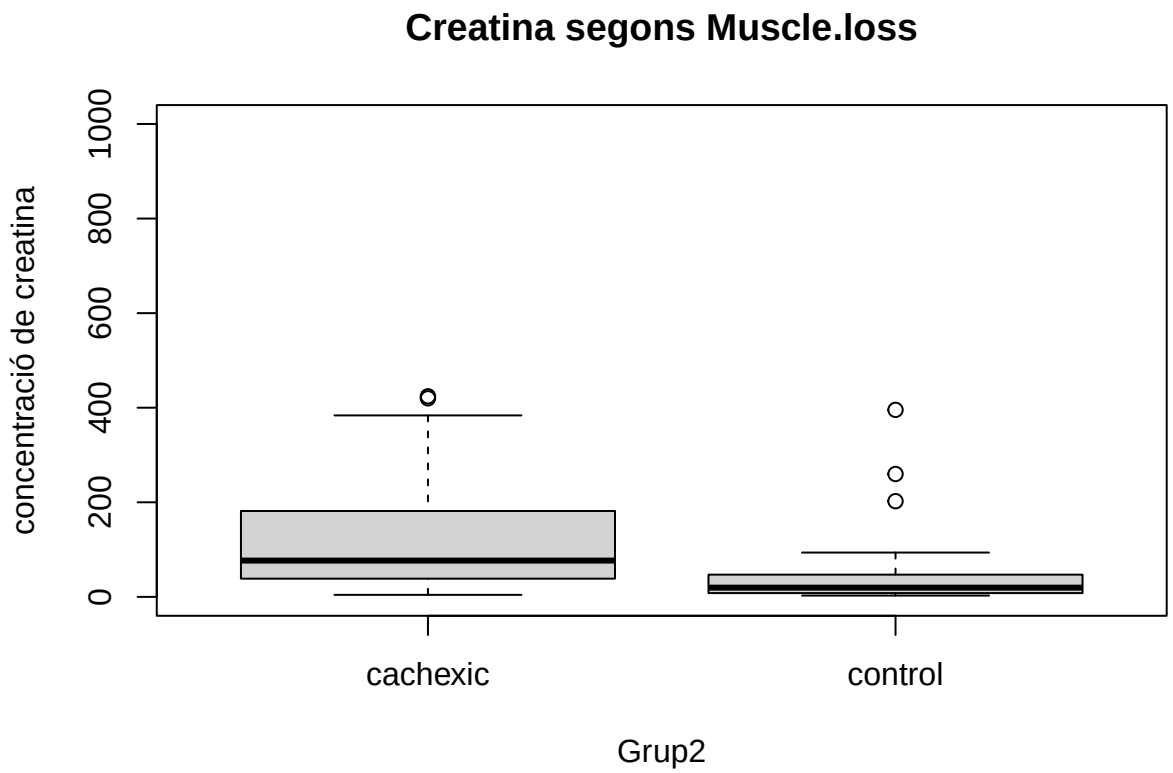
```r
#Comprovem que no hi hagi valors faltants (NA) en la matriu de dades
anyNA(assay(cachexia_se))
```

```
## [1] FALSE
```

Podem observar de forma general que aquestes dades consten d'un OOP *summarizedExperiment* on la matriu de dades està format per 77 pacients (columnes) els quals estàn dividits pel grup "Muscle.Loss" i 63 metabòlits (files) que són les concentracions de diferents metabòlits analitzades en les mostres d'orina dels pacients.

Creatina

```r
creatina <- assay(cachexia_se)["Creatine", ] #Extraiem les concentracions del metabòlit creatina
muscle_loss <- colData(cachexia_se)$Muscle.loss #Extraiem el grup Muscle.loss
boxplot(creatina ~ muscle_loss,
        main = "Creatina segons Muscle.loss",
        xlab = "Grup2",
        ylab = "concentració de creatina",
        ylim = c(0, 1000))
```




```r
t.test(creatina ~ muscle_loss)
```

```
##
##  Welch Two Sample t-test
##
## data:  creatina by muscle_loss
## t = 2.3988, df = 55.284, p-value = 0.01985
## alternative hypothesis: true difference in means between group cachexic and group control is not equa
```

```
## 95 percent confidence interval:
##   20.3217 226.4964
## sample estimates:
## mean in group cachexic  mean in group control
##              174.91340               51.50433
```

Mitjançant aquest anàlisi bàsic podem observar que el metabòlit creatina mostra una diferència significativa en la concentració entre els grups Muscle.loss. Observem que la mitjana en el grup que tenen *cachexia* (pèrdua constant de massa muscular) és significativament superior (174.91) a la del grup control (51.50) amb un interval de confiança de [20.32 - 226.50]. Aquests resultats poden indicar que la concentració de creatina en la orina podria estar relacionada amb l'estat de cachexia i, per tant, podria ser un potencial marcador per ajudar a diagnosticar aquesta malaltia. Això té certa coherència amb la fisiopatologia del síndrome, ja que la *cachexia* comporta un elevat catabolisme proteic i muscular que es pot traduïr a un augment de les concentracions extracel · lulars de creatina i, per tant, un augment en la concentració de creatina en la orina dels pacients amb *cachexia*.

Boxplot metabòlits més rellevants

Seguidament, seguirem amb l'anàlisis estadístic descriptiu mitjançant un boxplot múltiple. Com que no podem fer un boxplot dels 63 metabòlits, farem un t-test univariant per a cada metabòlit i seleccionarem els 4 metabòlits que tinguin p-valors més baixos (més significació).

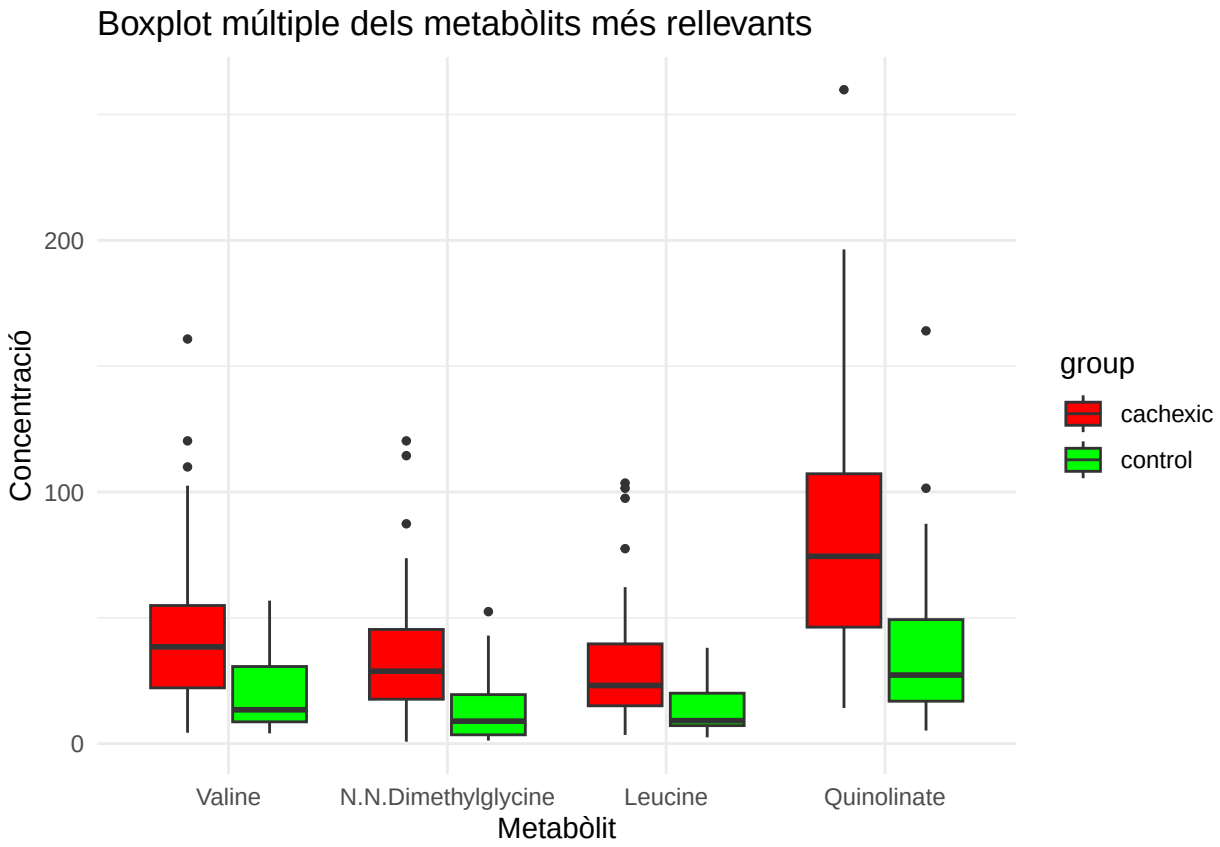
```
metabolits <- assay(cachexia_se) #Assignem metabolits
group <- colData(cachexia_se)$Muscle.loss #Assignem group a la variables Muscle.loss
#Fem un t-test per cada metabòlit i guardem els p-valors dels t-tests
p_valors <- apply(metabolits, 1, function(x) {
  tryCatch(t.test(x ~ group)$p.value, error = function(e) NA) #Agafem els p_valors dels t-tests de cada
})
#Ordenem els metabòlits segons els p-valors que hagin donat els t-tests
p_valors_ordenats <- sort(p_valors)
top_metabolits <- names(p_valors_ordenats) [1:4]
top_metabolits
```

```
## [1] "Valine"              "N.N.Dimethylglycine" "Leucine"
## [4] "Quinolinate"
```

Aquests són els metabòlits que han donat més nivell de significació fent el t-test segons la variable grup *Muscle.loss*. Per tant, haurien de ser els que tenen diferències més significatives de concentracions segons si els pacients tenen *cachexia* o no.

```
library(reshape2)
library(ggplot2)
#Seleccionem els metabòlits que ens han sortit
top_metabolits <- c("Valine", "N.N.Dimethylglycine", "Leucine", "Quinolinate")
#Extraiem la matriu només amb els metabòlits seleccionats
top_data <- assay(cachexia_se)[top_metabolits, ]
#Preparem les dades per fer ggplot2 (passem les mostres a les files en comptes de les columnes i anyadi
top_data_prep <- as.data.frame(t(top_data))
top_data_prep$group <- colData(cachexia_se)$Muscle.loss #Anyadim la columna grup
#Format compatible amb bloxplot
data_metabolits <- reshape2::melt(top_data_prep, id.vars = "group",
                                  variable.name = "metabòlit",
                                  value.name = "concentració")
#Amb les dades preparades, procedim a fer el boxploit múltiple
```

```r
ggplot(data_metabolits, aes( x= metabòlit, y = concentració, fill = group)) +
  geom_boxplot(outlier.size = 1) +
  labs(title = "Boxplot múltiple dels metabòlits més rellevants",
       x = "Metabòlit",
       y = "Concentració") +
  scale_fill_manual(values = c("cachexic" = "red", control = "green")) + #Separem grups per color
  theme_minimal()
```



Aquest gràfic mostra les diferències de concentracions dels 4 metabòlits que presenten més diferències significatives segons la variables categórica *Muscle.loss*. Tal i com s'observa en el gràfic, els 4 metabòlits tenen majors concentracions en els individus que presenten la malaltia *cachexia* que en els individus del grup control.

Pas 1: Anàlisi de Components Principals (PCA)

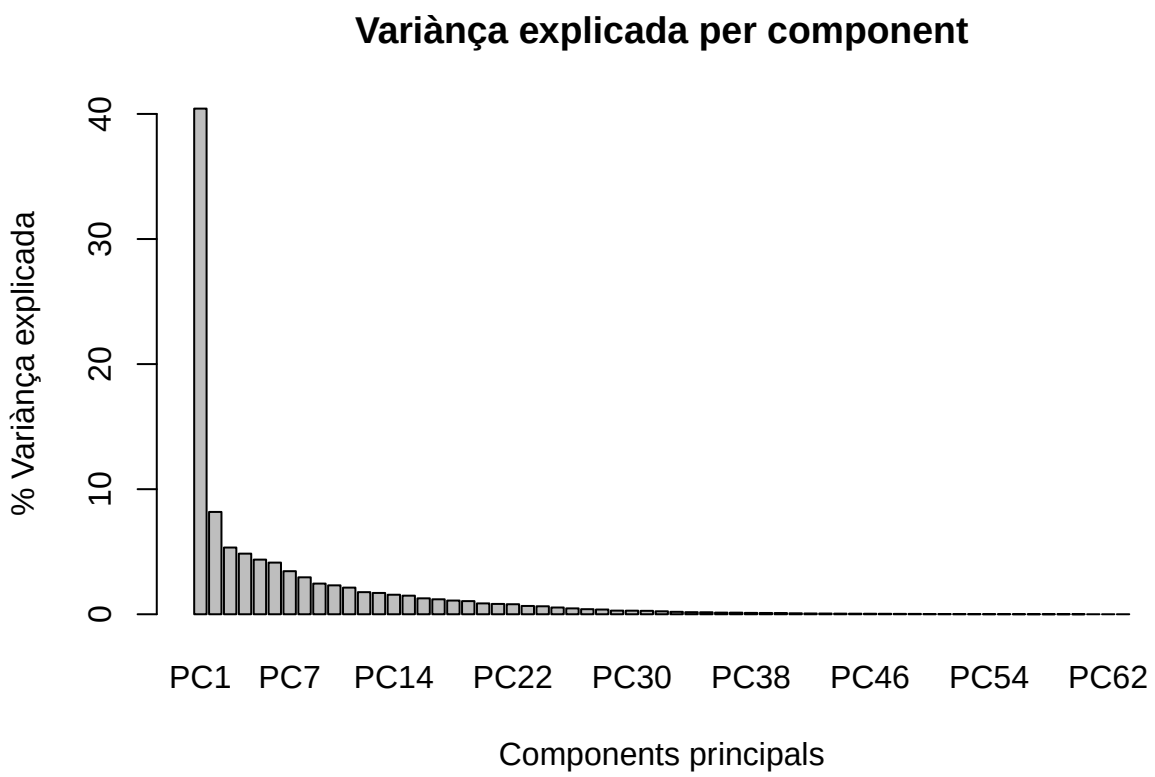
Mitjançant aquest tipus d'anàlisis, l'objectiu serà reduïr la dimensió de les dades i visualitzar si les mostres s'agrupen segons "Muscle.Loss" (*cachexia/control*) basant-se en els seus perfils metabolòmics:

```r
#Primerament, trasposem la matriu de l'objecte per tenir les mostres com a files i els metabòlits com a
t_data <- t(assay(cachexia_se))
#És recomanable escalar les variables quan estàn a diferents escales , en el nostre cas algun metabòlit
pca_resultats <- prcomp(t_data, scale. = TRUE)
summary(pca_resultats)$importance[, 1:5] #Mostrem els 5 components principals més importants (dels 63PC
```

```
##                             PC1      PC2      PC3      PC4      PC5
## Standard deviation     5.04667 2.270128 1.833107 1.747276 1.659056
## Proportion of Variance 0.40427 0.081800 0.053340 0.048460 0.043690
## Cumulative Proportion  0.40427 0.486070 0.539410 0.587870 0.631560
```

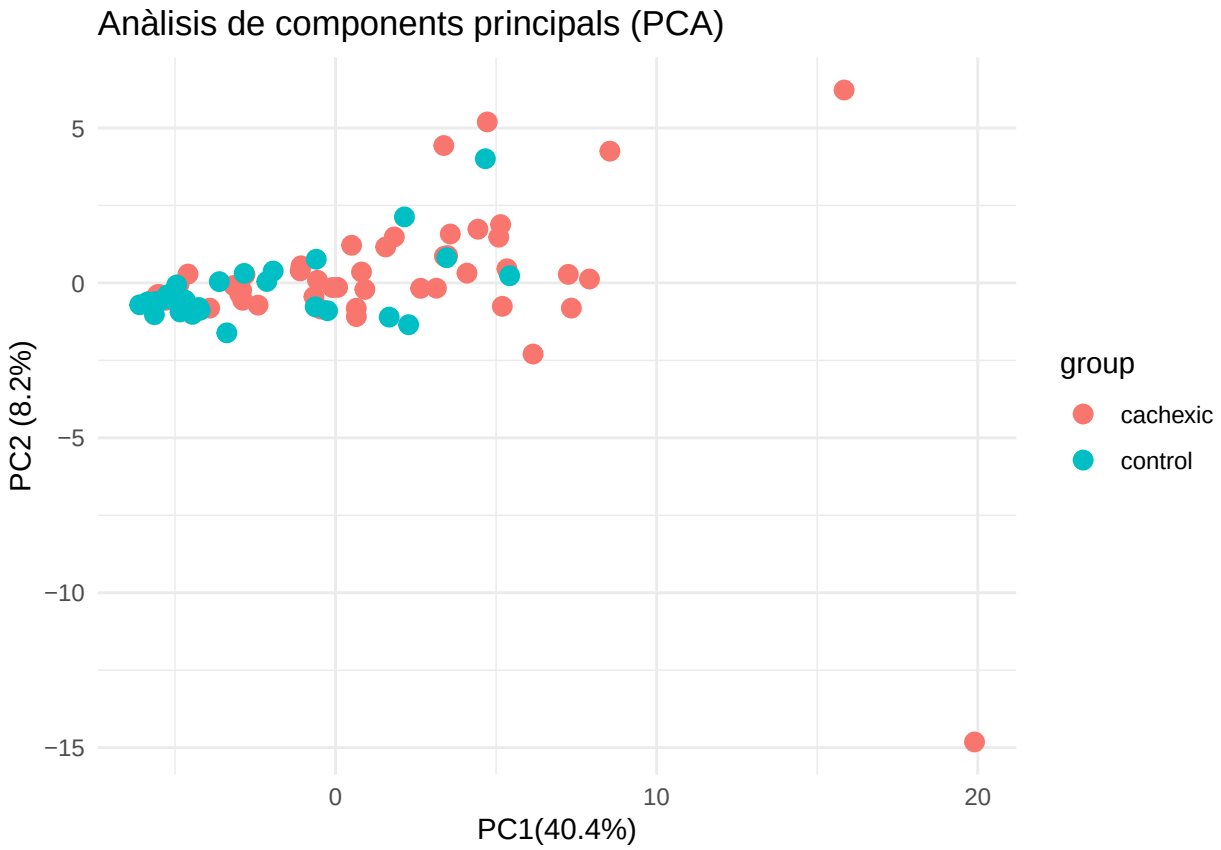
Observem en els resultats de l'anàlisis de components principals que els dos primers ja tenen una variabilitat del **48.61%**, que ja es considera bastant alta per ser dades òmiques.

```
var_explicada <- summary(pca_resultats)$importance[2, ] * 100 #Seleccionem PC1 i PC2
barplot(var_explicada,
        main = "Variança explicada per component",
        xlab = "Components principals",
        ylab = "% Variança explicada",
)
```



Observem que a partir de el PC2 ja la resta explica cada cop menys la variança de les dades. Per tant, decidim quedar-nos amb els dos primers components principals i utilitzarem aquests per a obtenir una representació de les dades en una dimensió reduïda:

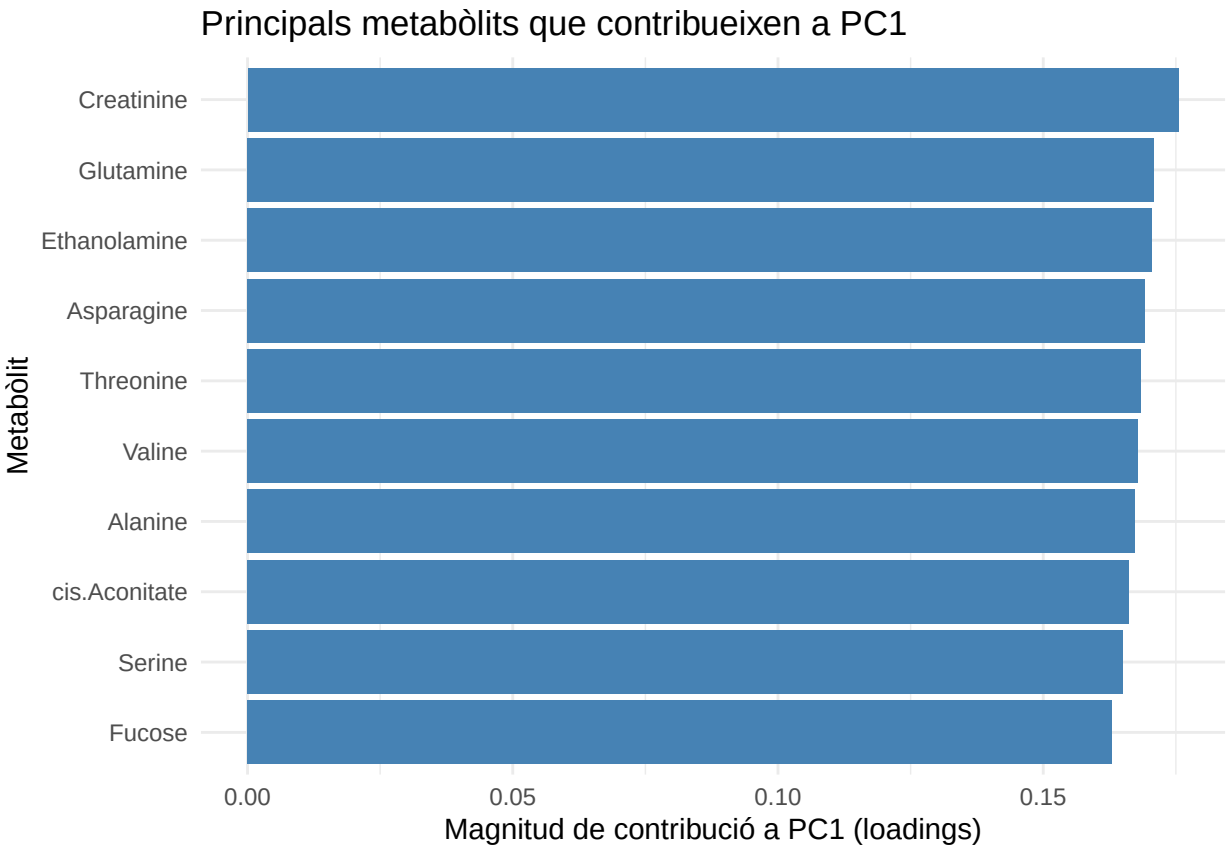
```
cach_control <- colData(cachexia_se)$Muscle.loss
pca_d <- as.data.frame(pca_resultats$x) #Cada fila representa una mostra i cada columna un PCA
pca_d$group <- cach_control #Afegim la classe de cada mostra
library(ggplot2)
ggplot(pca_d, aes(x = PC1, y = PC2, color = group)) + #Separem per grup segons el color
geom_point(size = 3) +
labs(
title = "Anàlisis de components principals (PCA)",
x = paste0("PC1(", round(var_explicada[1], 1), "%)"),
y = paste0("PC2 (", round(var_explicada[2], 1), "%)")
) + theme_minimal()
```



S'ha realitzar un anàlisi de components principals sobre la matriu de concentracions de metabòlits. Prèviament s'han centrat i escalat les dades per a evitar que les diferències d'escala entre les variables afectin l'anàlisi. Els dos primers components principals, com es pot observar, expliquen gairebé un 50% de la variança total (48.6%). La magnitud de la contribució de cada variable a les PC són els seus "loadings" en cada PC. Els autovectors (eigenvectors) associats a la matriu de covariança són els loadings, indiquen quina direcció prenen els nous components i quines variables (metabòlits) contribueixen més.

```
#creem un data frame amb els "loadings" (magnitud de contribució de cada metabòlit al component princip
pca_resultats <- prcomp(t_data, scale. = TRUE)
loadings_pca <- as.data.frame(pca_resultats$rotation) #assignem els loadings
loadings_pca$metabolit <- rownames(loadings_pca)
```

```
top_PC1 <- loadings_pca[order(abs(loadings_pca$PC1), decreasing =TRUE), ][1:10, ] #Agafem els 10 metabò
#Grafic
ggplot(top_PC1, aes(x = reorder(metabolit, PC1), y = PC1)) +
  geom_col(fill = "steelblue") +
  coord_flip() + #Cambiem els eixos per a millor visualització
  labs(title = "Principals metabòlits que contribueixen a PC1",
       x = "Metabòlit",
       y = "Magnitud de contribució a PC1 (loadings)") +
  theme_minimal()
```



Aquest gràfic mostra els 10 metabòlits que més contribueixen a la variança capturada pel primer component principal (PC1). Com hem pogut observar prèviament, el PC1 és el component principal del qual la seva direcció recull la major part de la variabilitat de les dades, i els valors *loading* indicarien quina força té cada metabòlit en definir aquesta direcció. Podem oobservar com la majoria de metabòlits contribueixen de forma gairebé equitativa a la PC1, destaquem la *Creatine*, que és la que contribueix més. Això té coherència amb l'anàlisi anterior, on ja havíem vist que aquest metabòlit mostrava diferències significatives segons el grup (*cachexia/control*).

CLUSTERING JERÀRQUIC

El clustering jeràquic és un potent recurs per a l'anàlisis exploratori de dades, porporcionant mètodes potents i flexibles per descubrir grups en les dades.

```
dades_s <- scale(t_data) #Matriu amb mostres com a files i metabòlits com a columnes (t) i escalada
dist_mostres <- dist(dades_s, method = "euclidean") #Calculqem la matriu de distàncies (eucledian)
hc <- hclust(dist_mostres, method = "complete") #Mètode de distància escollit: "Complete link", màxim d
grups <- colData(cachexia_se)$Muscle.loss #grups per etiquetal segons Muscle.loss
#Gràfic del dendograma
library(dendextend)
```

```
## Warning: package 'dendextend' was built under R version 4.4.3
```

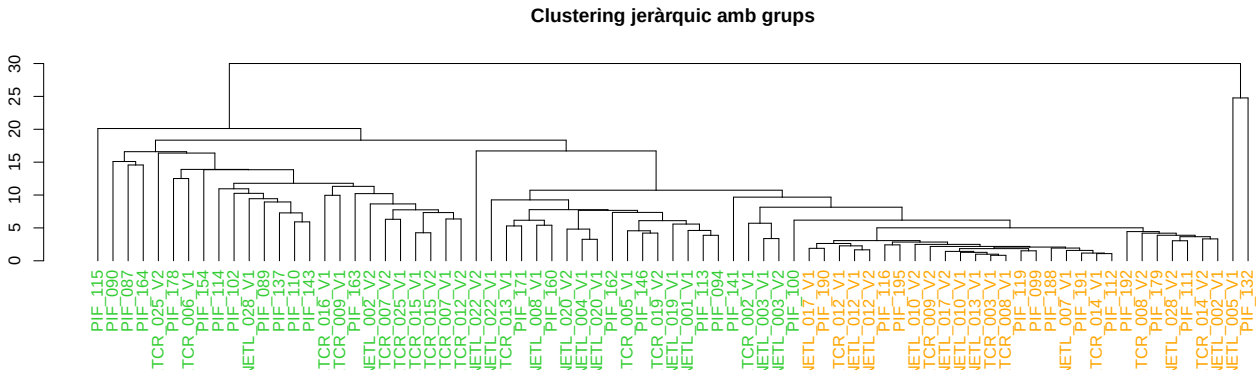
```
##
## ---------------------
## Welcome to dendextend version 1.19.0
## Type citation('dendextend') for how to cite the package.
##
## Type browseVignettes(package = 'dendextend') for the package vignette.
```

```
## The github page is: https://github.com/talgalili/dendextend/
##
## Suggestions and bug-reports can be submitted at: https://github.com/talgalili/dendextend/issues
## You may ask questions at stackoverflow, use the r and dendextend tags:
##   https://stackoverflow.com/questions/tagged/dendextend
##
##  To suppress this message use:  suppressPackageStartupMessages(library(dendextend))
## ----------------------
```

```
##
## Adjuntando el paquete: 'dendextend'
```

```
## The following object is masked from 'package:stats':
##
##     cutree
```

```
dend <- as.dendrogram(hc)
labels_colors(dend) <- ifelse(grups == "cachexic", "limegreen", "orange") #Dividim les mostres segons e
plot(dend,
     main = "Clustering jeràrquic amb grups",
     cex = 0.8)
```



Observem en el dendrograma que hi ha una separació visible entre els dos grups principals (*cachexia* i *control*), tot i que no està perfectament separada perquè algunes mostres *cachexia* (vermell) queden barrejades amb *control* (blau). Això pot indicar que pot haver efectes tècnics que desconeixem i/o que només alguns metabòlits separen clarament els dos grups (no tots).

Heatmap

Un heatmapa amb 63 variables (meatbòlits) seria massa sorollós i difícil d'interpretar. Per tant, abans de fer el heatmap farem una selecció prèvia dels 10 metabòlits (variables) més significatius.

```
#Com ho hem fet anteriorment, ja tenim els p-valors ordenats dels metabòlits
p_valors_ordenats <- sort(p_valors)
top_metabolits_heatmap <- names(p_valors_ordenats)[1:10] #Seleccionem els 10 metabòlits més significati
top_metabolits_heatmap
```

```
##  [1] "Valine"              "N.N.Dimethylglycine" "Leucine"
##  [4] "Quinolinate"         "Dimethylamine"       "Pyroglutamate"
##  [7] "X3.Hydroxybutyrate"  "Creatinine"          "Alanine"
## [10] "Glutamine"
```

```r
#extraim les dades només per als meta`+bolits més significatius
mat <- metabolits[top_metabolits_heatmap, ] #10 files (metabòlits) x 77 mostres (pacients)
#Escalem pels metabòlits (mitjana 0, desviació 1)
mat_scaled <- t(scale(t(mat))) #trasposem, escalem i tornem a transposar després
```

El heatmap mostratà les mostres (pacients) i els metabòlits, però no sap quin grup pertany cada mostra. Per tant, hem de fer que el mapa pugui caracteritzar les mostres segons el grup que pertany, li hem de donar la informació.

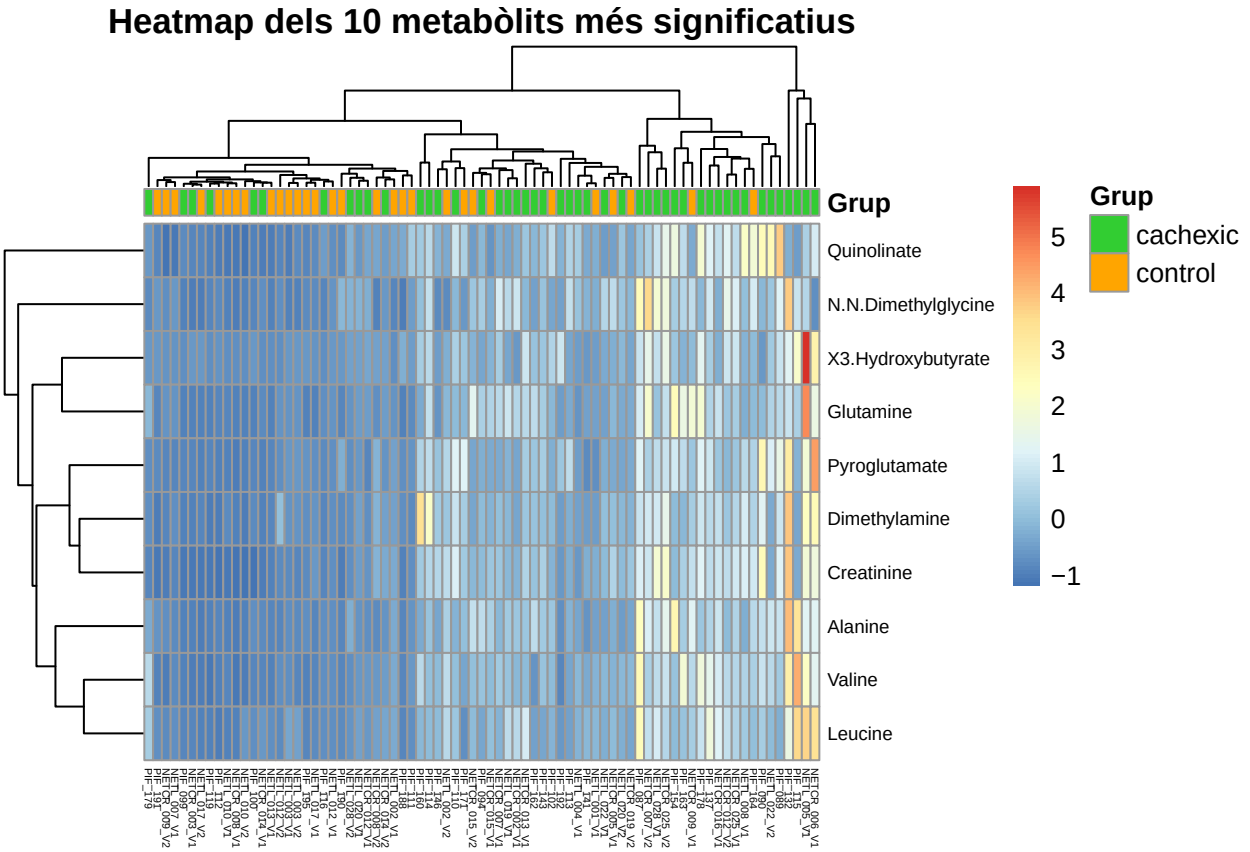
```r
anotacions <- data.frame(Grup = grups) #Creem un petit dataframe amb la columna grup (cachexia/control)
rownames(anotacions) <- colnames(mat_scaled) #Així el heatmap sabrà que x columna és la mostra PIC_XXX
head(anotacions)
```

```
##                 Grup
## PIF_178     cachexic
## PIF_087     cachexic
## PIF_090     cachexic
## NETL_005_V1 cachexic
## PIF_115     cachexic
## PIF_110     cachexic
```

```r
library(pheatmap)
```

```
## Warning: package 'pheatmap' was built under R version 4.4.3
```

```r
pheatmap(mat_scaled, #Matriu de dades de 10 meatbòlits per 77 mostres (pacients)
         annotation_col = anotacions, #Afegeix una linea de colors a dalt del mapa indicant si la mostr
         annotation_colors = list(
           Grup = c(cachexic = "limegreen", control  ="orange") #Definim el color per cada grup
         ),
         scale = "none", #None, ja hem escalat els valors manualment
         clustering_distance_rows = "euclidean", #Mètode per agrupar metabòlits
         clustering_distance_cols = "euclidean", #Mètode per agrupar mostres
         clustering_method = "complete", #clustering jeràrquic
         main = "Heatmap dels 10 metabòlits més significatius",
         fontsize_row = 7, #Tamany text dels metabòlits significatius
         fontsize_col = 4) #Tamany text de les mostres
```



Com que hem escalat la matriu de dades dels metabòlits, per cada fila de metabòlit la mitjana és 0 i la desviació estàndard és 1. D'aquesta manera la majoria de valors d'un metabòlit queden aprop del 0, però si hi ha algun valor molt alt comparat amb la mitjana, aquesta destacarà sobre la resta i mostrarà una coloració més llunyana del blau/blanc i s'aproparà al vermell. D'aquesta manera, amb el heatmap podem veure quins valors de metabòlits destaquen sobre la resta.

El dendrograma de dalt mostra com les mostres (pacients) s'agrupen segons la semblança dels seus perfils metabolòmics, d'aquesta manera veiem que les mostres de color verd que pertany al grup que té *cachexia* tendeixen a agrupar-se a la dreta, on es mostren valors dels metabòlits més elevats que les seves mitjanes. Mentre que les mostres de taronja que pertanyen als pacients control, tendeixen a agrupar-se a l'esquerra, amb perfils metabolòmics més propers a la mitjana (color blau).

Com ja havíem vist en els anàlisis anteriors, aquest patró reforça la idea que els pacients amb *cachexia* semblen presentar perfils metabolòmics diferenciats, amb concentracions més elevades en diversos metabòlits rellevants.

Els resultats observats al heatmap i la resta d'anàlisis són consistents amb la literatura científica sobre el síndroma *cachexia*. Ja que *cachexia* és un síndrome caracteritzat per una gran desregulació metabòlica, un augment de la degradació de proteïnes musculars i una activació de la gluconeogènesi i alteració de les vies energètiques (Evans et al., 2009; Argilés et al., 2014). Aquests processos catabòlics provoquen l'alliberament d'aminoàcids al torrent sanguini (valina, glutamina, leucina, alanina, etc) que podem veure reflectits en els pacients amb *cachexia* en el heatmap (majors concentracions, colors allunyats del blau). De la mateixa manera s'observa major presència d'intermedis com 3-hydroxybutyrate, pdoructe de l'oxidació de lípids en contextos de dèficit energètic. A més, observem un augment de quilonate que podria reflectir a l'activació de la via del triptòfan associada a l'estrès inflamatori i oxidatiu, habitual en pacients amb cachexia (Faeron et al., 2011).

Per tant, podem concluïr que els patrons observats en els resultats dels anàlisi no només tenen consistència estadística, sinó també una base fisiològica. Aquest conjunt d'evidències suggereix que el perfil metabolòmic

pot ser una eina molt útil per a identificar pacients amb *cachexia* i pot oferir un punt de partida per a futurs anàlisis de diagnòstic.

Referències

https://www.sthda.com/english/articles/31-principal-component-methods-in-r-practical-guide/117-hcpc-hierarchical-clustering-on-principal-components-essentials/

https://www.datanovia.com/en/blog/cluster-analysis-in-r-practical-guide/

Evans WJ, Morley JE, Argilés J, Bales C, Baracos V, Guttridge D, Jatoi A, Kalantar-Zadeh K, Lochs H, Mantovani G, Marks D, Mitch WE, Muscaritoli M, Najand A, Ponikowski P, Rossi Fanelli F, Schambelan M, Schols A, Schuster M, Thomas D, Wolfe R, Anker SD. Cachexia: a new definition. Clin Nutr. 2008 Dec;27(6):793-9. doi: 10.1016/j.clnu.2008.06.013. Epub 2008 Aug 21. PMID: 18718696.

Fearon K, Strasser F, Anker SD, Bosaeus I, Bruera E, Fainsinger RL, Jatoi A, Loprinzi C, MacDonald N, Mantovani G, Davis M, Muscaritoli M, Ottery F, Radbruch L, Ravasco P, Walsh D, Wilcock A, Kaasa S, Baracos VE. Definition and classification of cancer cachexia: an international consensus. Lancet Oncol. 2011 May;12(5):489-95. doi: 10.1016/S1470-2045(10)70218-7. Epub 2011 Feb 4. PMID: 21296615.

Argilés JM, Busquets S, Stemmler B, López-Soriano FJ. Cancer cachexia: understanding the molecular basis. Nat Rev Cancer. 2014 Nov;14(11):754-62. doi: 10.1038/nrc3829. Epub 2014 Oct 9. PMID: 25291291.