

PROBA

Gabriel Regueira Huguet

2025-03-22

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 muscular/o greixosa, inflamació sistèmica, alteracions hormonals i en el metabolisme enèrgic. Aquesta pèrdua de massa muscular es tradueix a un catabolisme muscular accelerat, que provoca l'alliberament de aminoàcids com la valina, leucina, alanina, etc. Aquest síndrome provoca que el pacient es trobi en un estat de semi-fam metabòlica, que provoca que hi hagi una demanda energètica elevada i alguns metabòlits intermedis del metabolisme energètic s'acumulen (3-hydroxybutyrate, pyroglutamate, glutamine).

S'ha descobert que el dataset de *cachexia* ja es troba en un paquet d'R anomenat *specmine.datasets*:

```
library(specmine.datasets)
data("cachexia")
```

```
class(cachexia)
```

```
## [1] "list"
```

Com podem observar, *cachexia* és una llista personalitzada utilitzada per el paquet *specmine.datasets*. Al executar `View(cachexia)` no surt una taula de dades convencional, sinó una vista estructurada dels components del dataset *cachexia*.

Observem en el dataset de *cachexia* que aquest té diferents elements:

- data: matriu 63 x 77 (metabolites x mostres)
- metadata: data.frame (77 x 1) amb informació sobre cada mostra (grup), indica a quin grup pertany cada mostra (*Muscle.loss*)
- description: petita descripció sobre les dades

2- CREAR UN OBJECTE SUMMARIZED EXPERIMENT:

Ara procedim a crear el *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, colAvgPerRowSet, 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, rowAvgPerColSet,
##   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
```

```
#Convertim cachexia$data a una matriu R per tal que SummarizedExperiment accepti el format i formi part
assay_data <- as.matrix(cachexia$data)
#Agafem cachexia$metadata i ens assegurem que els noms de les files coincideixin amb le nom de les columnes
col_metadata <- cachexia$metadata
rownames(col_metadata) <- colnames(assay_data)
#Creem l'objecte SummarizedExperiment amb la matriu de dades empaquetada en una llista (counts) i les m
se <- SummarizedExperiment(
  assays = list(counts = assay_data),
  colData = col_metadata
)
```

```
se
```

```
## class: SummarizedExperiment
## dim: 63 77
## metadata(0):
## assays(1): counts
```

```
## rownames(63): 1.6-Anhydro-beta-D-glucose 1-Methylnicotinamide ...
##   pi-Methylhistidine tau-Methylhistidine
## rowData names(0):
## colnames(77): PIF_178 PIF_087 ... NETL_003_V1 NETL_003_V2
## colData names(1): Muscle.loss
```

```
View(assay(se))
View(colData(se))
colData(se)
```

```
## DataFrame with 77 rows and 1 column
##           Muscle.loss
##           <factor>
## PIF_178      cachexic
## PIF_087      cachexic
## PIF_090      cachexic
## NETL_005_V1   cachexic
## PIF_115      cachexic
## ...          ...
## NETCR_019_V2  control
## NETL_012_V1   control
## NETL_012_V2   control
## NETL_003_V1   control
## NETL_003_V2   control
```

```
colnames(colData(se)) #Grup cachexia/control
```

```
## [1] "Muscle.loss"
```

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

```
save(se, file = "se_cachexia.rda")
```

Diferències *ExpressionSet* i *SummarizedExperiment* :

ExpressionSet ha estat durant molt temps el format clàssic per analitzar dades de microarrays, 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:

```
load("se_cachexia.rda")
```

```
apply(assay(se), 1, summary)
```

```
##           1.6-Anhydro-beta-D-glucose 1-Methylnicotinamide 2-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			
##	2-Hydroxyisobutyrate	2-Oxoglutarate	3-Aminoisobutyrate				
##	Min.	4.85000	5.5300	2.61000			
##	1st Qu.	15.80000	22.4200	11.70000			
##	Median	32.46000	55.1500	22.65000			
##	Mean	37.25065	145.0871	76.75636			
##	3rd Qu.	54.60000	92.7600	56.26000			
##	Max.	93.69000	2465.1300	1480.30000			
##	3-Hydroxybutyrate	3-Hydroxyisovalerate	3-Indoxylsulfate				
##	Min.	1.70000	0.92000	27.6600			
##	1st Qu.	5.99000	5.26000	82.2700			
##	Median	11.70000	12.55000	144.0300			
##	Mean	21.71701	21.64779	218.8792			
##	3rd Qu.	29.96000	30.27000	333.6200			
##	Max.	175.91000	164.02000	1043.1500			
##	4-Hydroxyphenylacetate	Acetate	Acetone	Adipate	Alanine		
##	Min.	15.490	3.49000	2.29000	1.55000	16.7800	
##	1st Qu.	41.680	16.28000	4.95000	6.11000	78.2600	
##	Median	70.110	39.65000	7.10000	10.18000	194.4200	
##	Mean	112.021	66.14143	11.42701	24.75636	273.5623	
##	3rd Qu.	145.470	86.49000	10.49000	19.11000	399.4100	
##	Max.	796.320	411.58000	206.44000	327.01000	1312.9100	
##	Asparagine	Betaine	Carnitine	Citrate	Creatine	Creatinine	
##	Min.	6.69000	2.29000	2.18000	59.740	2.7500	1002.250
##	1st Qu.	20.49000	28.79000	14.44000	788.400	17.6400	3498.190
##	Median	42.10000	64.72000	23.81000	1790.050	44.2600	7631.200
##	Mean	62.28364	90.32468	52.08506	2235.346	126.8319	8733.972
##	3rd Qu.	89.12000	127.74000	60.95000	3071.740	117.9200	12332.580
##	Max.	273.14000	391.51000	487.85000	13629.610	1863.1100	33860.350
##	Dimethylamine	Ethanolamine	Formate	Fucose	Fumarate	Glucose	
##	Min.	41.2600	16.1200	6.420	5.70000	0.79000	26.8400
##	1st Qu.	142.5900	86.4900	53.520	29.37000	2.23000	80.6400
##	Median	304.9000	204.3800	95.580	61.56000	4.10000	210.6100
##	Mean	358.1661	276.2604	147.403	88.66883	8.44013	559.8445
##	3rd Qu.	454.8600	407.4800	167.340	123.97000	7.85000	407.4800
##	Max.	1556.2000	1436.5500	1480.300	407.48000	96.54000	8690.6200
##	Glutamine	Glycine	Glycolate	Guanidoacetate	Hippurate	Histidine	
##	Min.	23.3400	38.0900	5.4200	7.03000	92.760	14.1500
##	1st Qu.	113.3000	262.4300	50.9100	33.78000	492.750	66.6900
##	Median	225.8800	528.4800	130.3200	64.72000	1224.150	174.1600
##	Mean	306.8716	880.7174	187.9894	86.37052	2286.838	292.6375
##	3rd Qu.	445.8600	1096.6300	267.7400	108.85000	2921.930	419.8900
##	Max.	1685.8100	5064.4500	720.5400	561.16000	19341.340	1863.1100
##	Hypoxanthine	Isoleucine	Lactate	Leucine	Lysine	Methylamine	
##	Min.	3.78000	1.790000	7.3200	2.51000	10.4900	1.51000
##	1st Qu.	20.70000	3.900000	35.5200	9.12000	30.2700	5.26000
##	Median	40.04000	7.170000	81.4500	19.11000	69.4100	14.73000
##	Mean	61.09766	8.709091	158.4565	24.36364	108.7942	17.37623
##	3rd Qu.	83.93000	11.250000	139.7700	31.19000	121.5100	24.05000
##	Max.	265.07000	40.040000	3640.9500	103.54000	788.4000	52.46000
##	Methylguanidine	N.N-Dimethylglycine	O-Acetylcarnitine	Pantothenate			

```
## Min.          1.70000          0.79000          1.23000          2.59000
## 1st Qu.       4.26000          7.03000          3.94000         11.13000
## Median       7.85000         21.98000         11.47000         22.65000
## Mean        15.32455         26.34961         19.73338         44.88377
## 3rd Qu.     19.30000         40.04000         20.91000         41.26000
## Max.       141.17000        120.30000        254.68000        692.29000
##      Pyroglutamate Pyruvate Quinolinolate Serine Succinate Sucrose
## Min.          21.3300  0.90000    5.21000    16.1200  1.72000    6.4900
## 1st Qu.       68.7200  4.85000   26.58000    83.1000  8.58000   19.3000
## Median      157.5900 13.46000   51.42000   142.5900 30.88000   40.8500
## Mean       211.4478 21.29442   66.43948  197.6869 60.22909  113.2278
## 3rd Qu.     301.8700 29.08000   87.36000  270.4300 74.44000   94.6300
## Max.     1064.2200 184.93000  259.82000 1248.8800 589.93000 2079.7400
##      Tartrate  Taurine Threonine Trigonelline Trimethylamine N-oxide
## Min.          2.20000   17.8100    8.2500    10.0700          55.7000
## 1st Qu.       6.89000   99.4800   31.8200    53.5200         175.9100
## Median      12.94000  249.6400   64.0700   114.4300         383.7500
## Mean       40.00403  525.1235   95.3574   270.4361         652.1569
## 3rd Qu.     25.79000  665.1400  137.0000   340.3600         735.1000
## Max.     837.15000 4272.6900  450.3400  2252.9600        5486.2500
##      Tryptophan Tyrosine  Uracil  Valine  Xylose cis-Aconitate
## Min.          8.67000   4.22000   3.10000   4.10000  10.0700    12.9400
## 1st Qu.      21.33000  23.57000  11.94000  12.18000  29.9600    36.2300
## Median      46.99000  60.34000  27.39000  33.12000  50.4000    129.0200
## Mean       66.24312  81.75727  35.55766  35.66701 100.9334    204.2197
## 3rd Qu.     96.54000 113.30000  44.26000  50.40000  89.1200    254.6800
## Max.     259.82000 539.15000 179.47000 160.77000 2164.6200   1863.1100
##      myo-Inositol trans-Aconitate pi-Methylhistidine tau-Methylhistidine
## Min.          11.5900          4.90000          11.3600          8.00000
## 1st Qu.       30.2700          12.43000          67.3600          27.39000
## Median       78.2600          26.84000          162.3900          68.72000
## Mean       135.3975          40.63039          370.2883          89.68688
## 3rd Qu.     167.3400          57.40000          387.6100          130.32000
## Max.       854.0600          217.02000          2697.2800          317.35000
```

```
dim(se) #Nombre de files i columnes
```

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

En aquest cas, es pot deduir que les variables són 63 concentracions de metabòlits analitzats en la orina de 77 individus. Totes les variables, doncs, són numèriques menys la variable grup de *Muscle.loss*.

```
anyNA(assay(se)) #No hi ha valors faltants (NA) en la matriu de dades
```

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

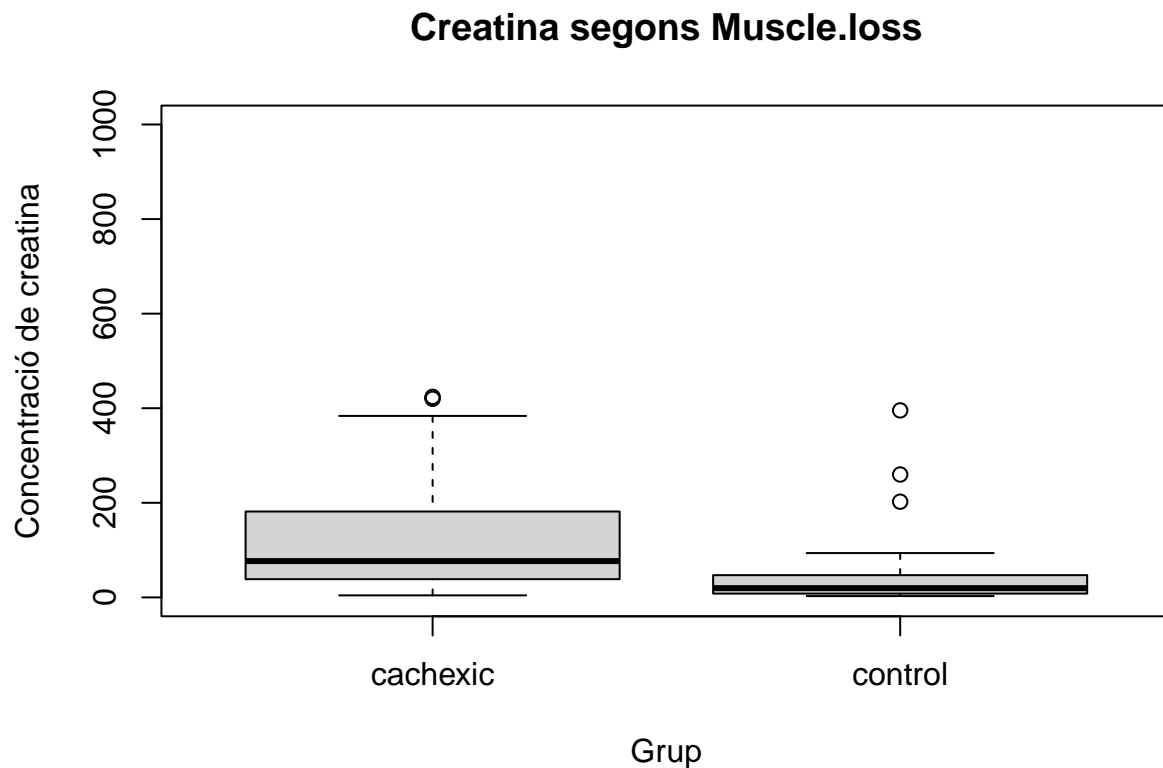
```
colData(se)
```

```
## DataFrame with 77 rows and 1 column
##      Muscle.loss
##      <factor>
## PIF_178      cachexic
```

```
## PIF_087      cachexic
## PIF_090      cachexic
## NETL_005_V1  cachexic
## PIF_115      cachexic
## ...          ...
## NETCR_019_V2 control
## NETL_012_V1  control
## NETL_012_V2  control
## NETL_003_V1  control
## NETL_003_V2  control
```

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 estan dividits pel grup "Muscle.Loss" i 63 metabòlits (files) que són les concentracions de diferents metabòlits analitzades en les mostres d'orina proporcionades pels pacients.

```
Creatina <- assay(se)["Creatine", ] #Extraim le concentracions del metabòlit "creatine"
muscle_loss <- colData(se)$Muscle.loss #ExtraIm el grup Muscle.loss
boxplot(Creatina ~ muscle_loss,
        main = "Creatina segons Muscle.loss",
        xlab = "Grup",
        ylab = "Concentració de creatina",
        ylim = c(0, 1000))
```



```
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 equal to 0
## 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 la *cachexia* comporta un elevat catabolisme proteic i muscular que es pot traduir 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*.

Seguidament, seguirem amb l'anàlisi 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).

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

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

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)
#Seleccióem els metabòlits
top_metabolits <- c("Valine", "N.N-Dimethylglycine", "Leucine", "Quinolate")
#Extraïm la matriu només amb els metabòlits seleccionats
top_data <- assay(se)[top_metabolits, ]
```

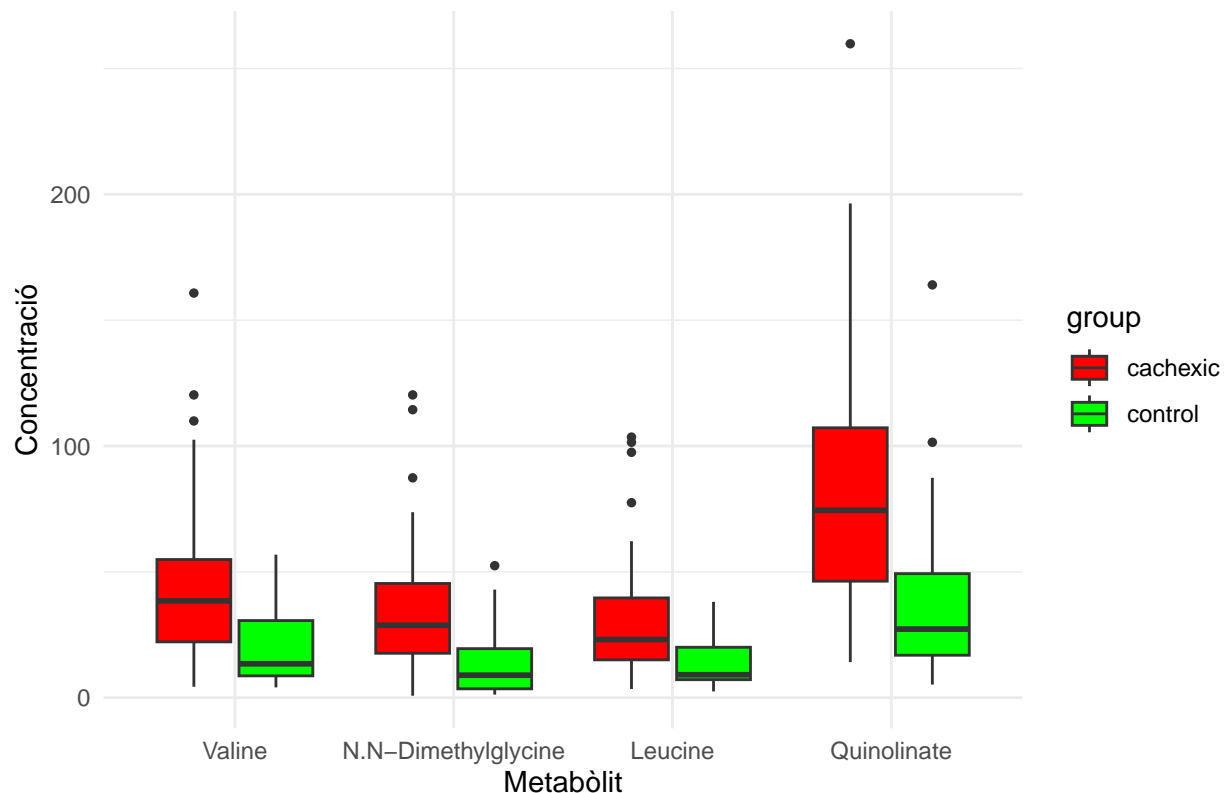


```

#Preparem les dades per ggplot2 (passem les mostres a les files en comptes de les columnes i anyadim la
top_data_prep <- as.data.frame(t(top_data))
top_data_prep$group <- colData(se)$Muscle.loss #Anyadim la columna group
#Format compatible amb boxplot
data_met <- reshape2::melt(top_data_prep, id.vars = "group",
                           variable.name = "metabòlit",
                           value.name = "concentració")
#Amb les dades preparades, procedim a fer el boxplot múltiple
ggplot(data_met, 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 amb colors (M
  theme_minimal()

```

Boxplot múltiple dels metabòlits més rellevants



Aquest gràfic mostra les diferències de concentracions dels 4 metabòlits que presenten més diferències significatives segons la variable categòrica *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'anàlisi, l'objectiu serà reduir la dimensió de les dades i visualitzar si les mostres s'agrupen segons "Muscle.Loss" (*cachexia/control*) basant-se en els seus perfils metabolòmics:

*Matriu de covariança FACER**

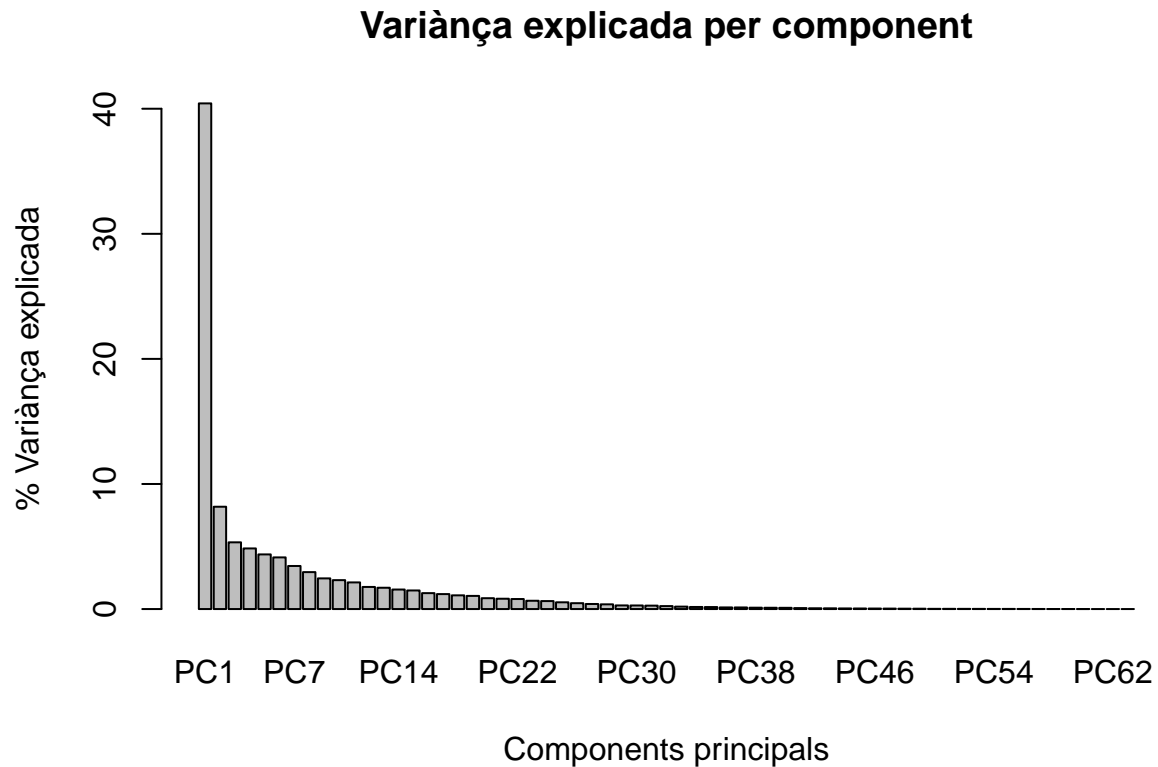
```
#Trasposem la matriu per tenir les mostres com a files i els metabòlits com a columnes
t_data <- t(assay(se))
cach_control <- colData(se)$Muscle.loss
#És recomanable centrar i escalar les variables quan estan en diferents escales, en el nostre cas algun
pca_resultats <- prcomp(t_data, scale. = TRUE) #Calcula internament la matriu de covariàncies i centra
summary(pca_resultats)
```

```
## Importance of components:
##
##          PC1      PC2      PC3      PC4      PC5      PC6      PC7
## Standard deviation  5.0467 2.2701 1.83311 1.74728 1.65906 1.6130 1.47304
## Proportion of Variance 0.4043 0.0818 0.05334 0.04846 0.04369 0.0413 0.03444
## Cumulative Proportion 0.4043 0.4861 0.53941 0.58787 0.63156 0.6729 0.70730
##
##          PC8      PC9      PC10     PC11     PC12     PC13     PC14
## Standard deviation  1.36403 1.24275 1.20650 1.1584 1.05503 1.03620 0.9914
## Proportion of Variance 0.02953 0.02451 0.02311 0.0213 0.01767 0.01704 0.0156
## Cumulative Proportion 0.73683 0.76135 0.78445 0.8057 0.82342 0.84046 0.8561
##
##          PC15     PC16     PC17     PC18     PC19     PC20     PC21
## Standard deviation  0.96773 0.89551 0.86788 0.83041 0.8133 0.73918 0.72112
## Proportion of Variance 0.01487 0.01273 0.01196 0.01095 0.0105 0.00867 0.00825
## Cumulative Proportion 0.87093 0.88366 0.89562 0.90656 0.9171 0.92573 0.93399
##
##          PC22     PC23     PC24     PC25     PC26     PC27     PC28
## Standard deviation  0.71053 0.64606 0.63389 0.5830 0.5442 0.50539 0.48743
## Proportion of Variance 0.00801 0.00663 0.00638 0.0054 0.0047 0.00405 0.00377
## Cumulative Proportion 0.94200 0.94863 0.95500 0.9604 0.9651 0.96916 0.97293
##
##          PC29     PC30     PC31     PC32     PC33     PC34     PC35
## Standard deviation  0.42674 0.42427 0.41483 0.38653 0.35092 0.32424 0.31646
## Proportion of Variance 0.00289 0.00286 0.00273 0.00237 0.00195 0.00167 0.00159
## Cumulative Proportion 0.97582 0.97867 0.98141 0.98378 0.98573 0.98740 0.98899
##
##          PC36     PC37     PC38     PC39     PC40     PC41     PC42
## Standard deviation  0.2867 0.28435 0.26060 0.25353 0.24800 0.21896 0.19537
## Proportion of Variance 0.0013 0.00128 0.00108 0.00102 0.00098 0.00076 0.00061
## Cumulative Proportion 0.9903 0.99158 0.99266 0.99368 0.99465 0.99541 0.99602
##
##          PC43     PC44     PC45     PC46     PC47     PC48     PC49
## Standard deviation  0.18914 0.1767 0.16864 0.1580 0.15287 0.1380 0.13101
## Proportion of Variance 0.00057 0.0005 0.00045 0.0004 0.00037 0.0003 0.00027
## Cumulative Proportion 0.99659 0.9971 0.99753 0.9979 0.99830 0.9986 0.99888
##
##          PC50     PC51     PC52     PC53     PC54     PC55     PC56
## Standard deviation  0.10759 0.10374 0.09853 0.08760 0.08258 0.08049 0.06927
## Proportion of Variance 0.00018 0.00017 0.00015 0.00012 0.00011 0.00010 0.00008
## Cumulative Proportion 0.99906 0.99923 0.99939 0.99951 0.99962 0.99972 0.99979
##
##          PC57     PC58     PC59     PC60     PC61     PC62     PC63
## Standard deviation  0.05937 0.05673 0.05088 0.04001 0.02972 0.02789 0.01876
## Proportion of Variance 0.00006 0.00005 0.00004 0.00003 0.00001 0.00001 0.00001
## Cumulative Proportion 0.99985 0.99990 0.99994 0.99997 0.99998 0.99999 1.00000
```

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

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

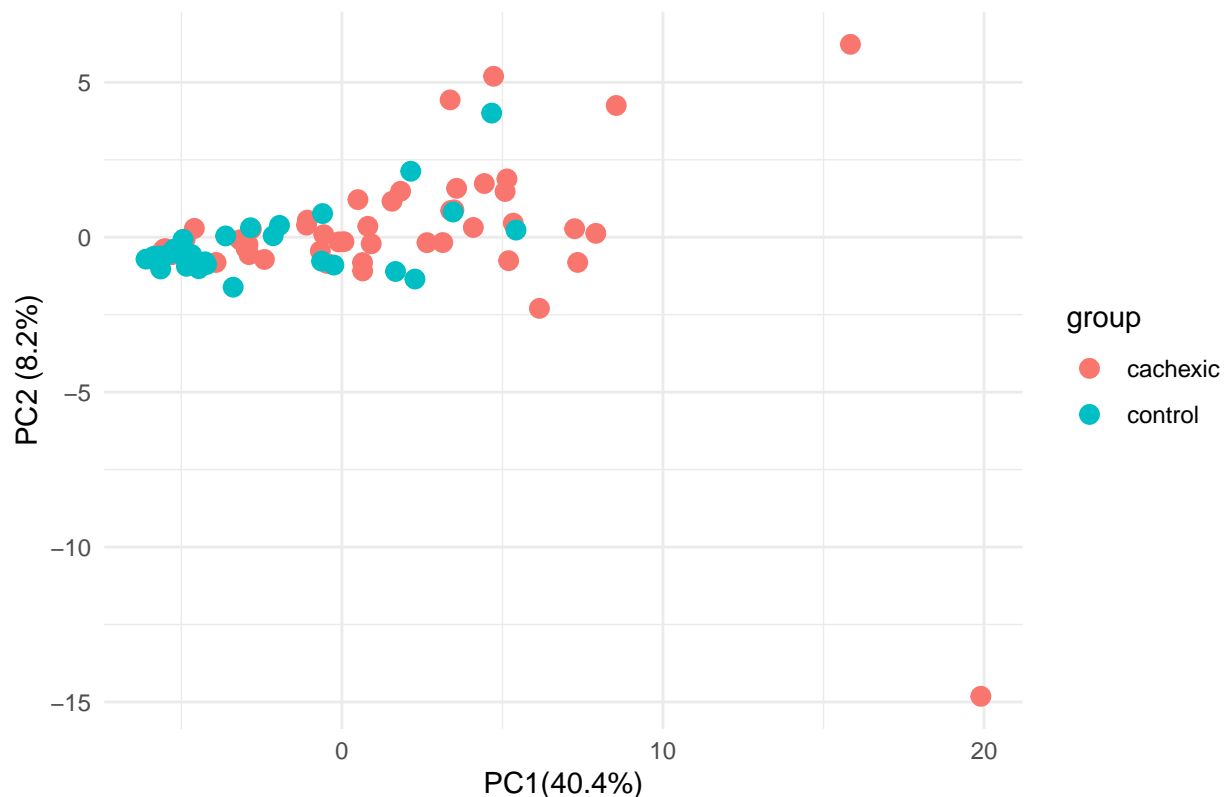
```
)
  ylab = "% Variànça explicada",
```



Ara ut8ilitzarem els valors dels primers components principals per a obtenir una representació de les dades en una dimensió reduïda.

```
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()
```

Anàlisi de components principals (PCA)



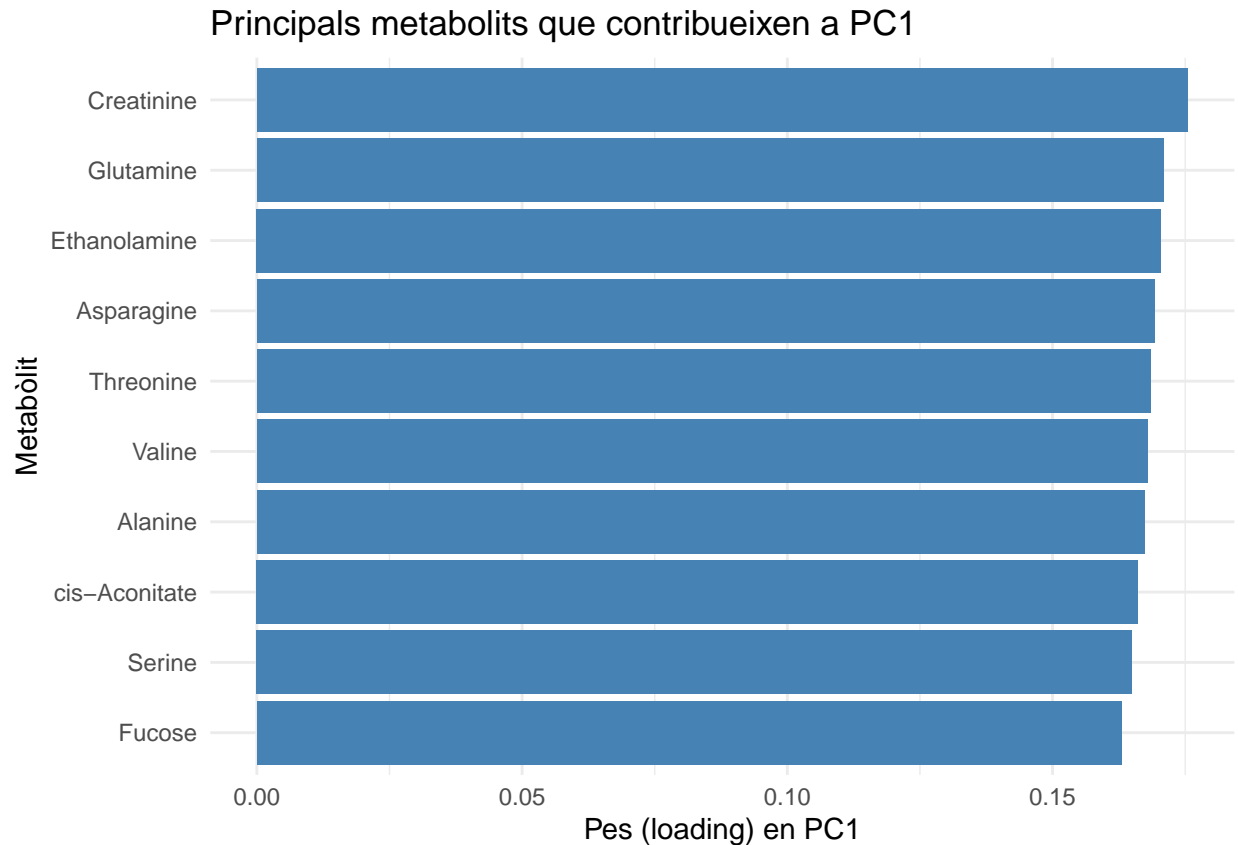
S'ha realitzat un anàlisi de components principals sobre la matriu de concentracions de metabòlits. Prèviament s'han centrat i escalat les dades per a evitar que les diferències d'escala entre les variables afectin l'anàlisi. Els dos primers components principals, com es pot observar, expliquen gairebé un 50% de la variància 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 covariància són els loadings, indiquen quina direcció prenen els nous components i quines variables (metabòlits) contribueixen més.

```
#Creem un data frame amb els loadings
loadings_pca <- as.data.frame(pca_resultats$rotation)
loadings_pca$metabolit <- rownames(loadings_pca)

top_PC1 <- loadings_pca[order(abs(loadings_pca$PC1), decreasing = TRUE), ][1:10, ]

ggplot(top_PC1, aes(x = reorder(metabolit, PC1), y = PC1)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(title = "Principals metabòlits que contribueixen a PC1",
       x = "Metabòlit",
       y = "Pes (loading) en PC1") +
  theme_minimal()
```



CLUSTERING JERÀRQUIC

El clustering jeràquic és un potent recurs per a l'anàlisi exploratori de dades, porporcionant mètodes potents i flexibles per descobrir 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") #Calculgem la matriu de distàncies (euclidian)
hc <- hclust(dist_mostres, method = "complete") #Mètode de distància escollit: "Complete link", màxim d
grups <- colData(se)$Muscle.loss #grups per etiquetar 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
##
```

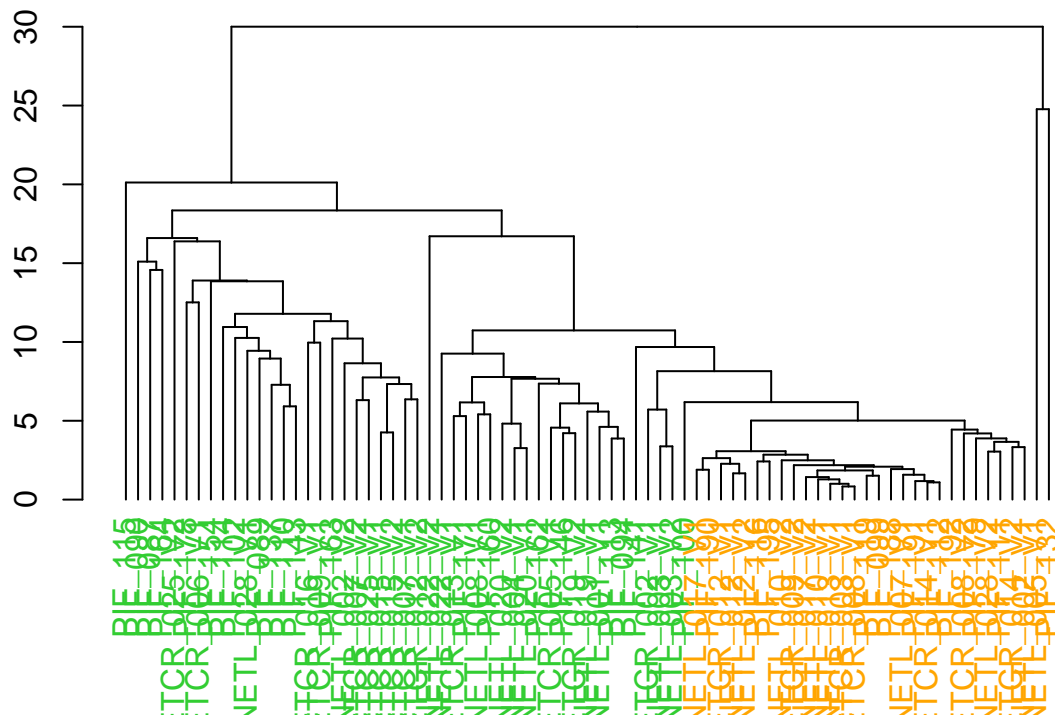
```
## To suppress this message use: suppressPackageStartupMessages(library(dendextend))
## -----
```

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

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

```
dend <- as.dendrogram(hc)
labels_colors(dend) <- ifelse(grups == "cachexia", "limegreen", "orange") #Dividim les mostres segons e
plot(dend,
     main = "Clustering jeràrquic amb grups",
     cex = 0.6)
```

Clustering jeràrquic amb grups



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 desconexim i/o que només alguns metabòlits separen clarament els dos grups (no tots).

Heatmap

Un heatmap amb 63 variables (metabòlits) seria massa sorollós i difícil d'interpretar. Per tant, abans de fer el heatmap farem una selecció prèvia dels 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] "Quinolate" "Dimethylamine" "Pyroglutamate"
## [7] "3-Hydroxybutyrate" "Creatinine" "Alanine"
## [10] "Glutamine"
```

```
#extraim les dades només per als meta`bolits més significatius
mat <- metab[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 mostrarà 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.

```
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 a columna és la mostra PIF_xxx
head(anotacions)
```

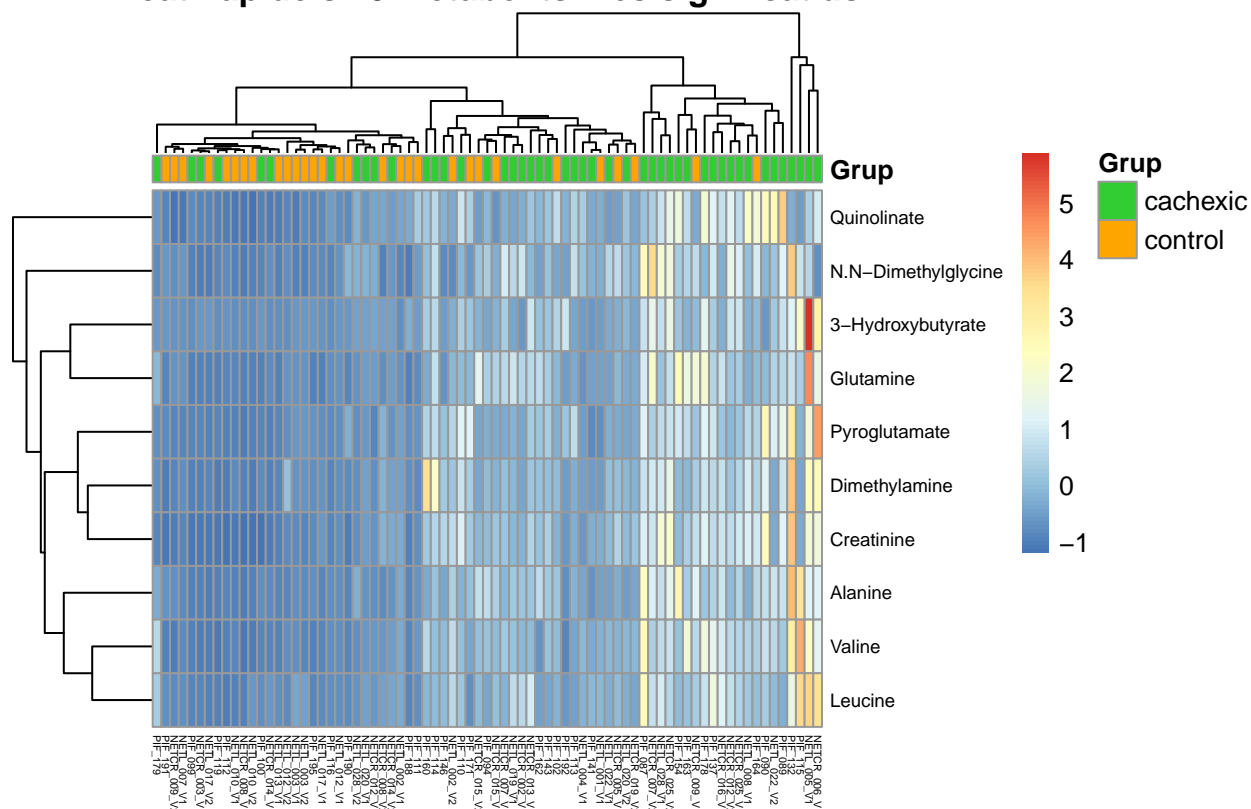
```
##          Grup
## PIF_178    cachexic
## PIF_087    cachexic
## PIF_090    cachexic
## NETL_005_V1 cachexic
## PIF_115    cachexic
## PIF_110    cachexic
```

```
library(pheatmap)
```

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

```
pheatmap(mat_scaled, #Matriu de dades de 10 metabòlits per 77 mostres (pacients)
  annotation_col = anotacions, #Afegeix una línia de colors a dalt del mapa indicant si la mostra
  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
```

Heatmap dels 10 metabòlits més significatius



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 síndrome *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, producte de l'oxidació de lípids en contextos de dèficit energètic. A més, observem un augment de quinolate 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 concloure que els patrons observats en els resultats dels anàlisis 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.