Fernández_López_David_PEC1_informe

David Fernández

2024-11-06

Abstract

En aquest treball es presenta l'anàlisi d'un dataset de dades metabolòmiques 2023-CIMBCTutorial descarregat del repositori github: (*https://github.com/nutrimetabolomics/metaboData/*). En aquest es duen a terme alguns anàlisis que permeten observar les diferències entre els metabòlits de les mostres segons el grup i també per veure com es relacionen entre sí.

Objectius de l'estudi

L'objetiu d'aquest treball és explorar les dades d'un dataset per poder dur a terme una visió general de com s'estructuren les dades i si es poden trobar algunes diferències entre els grups de mostres a partir d'anàlisis multivariants com l'anàlisi de components principals.

A part, també es preten familiaritzar-se amb la creació i manipulació de contenidors de tipus SummarizedExperiment i també la creació de repositoris Github així com també treballar en el propi entorn de R.

Materials i mètodes

Selecció del dataset

D'entre els datasets proporcionats en el repositori de Github, es treballarà amb el Dataset usat al tutorial CIMBC *"Basic Metabolomics Data Analysis Workflow"*.

Primer entrem al repositori de github (*https://github.com/nutrimetabolomics/metaboData/*) i ens trobem amb el votó verd de "code" d'on podem agafar l'enllaç. Llavors, obrim R i podem clonar-lo accedint a File > New Project i un cop aquí ens surt una pantalla on seleccionem "Version Control" i llavors seleccionem "Git" on enganxarem el link que hem copiat del repositori.

Ara, a la barra lateral ens surten tots els arxius disponibles. Podem veure una carpeta que diu "Datasets" i dins en trobem diferents.

Aquestes dades corresponen a les concentracions de metabòlits en diferents tipus de mostres (individus) que es poden categoritzar per classes, com sans i amb càncer gàstric, entre altres.

Les dades de l'estudi es troben en un fitxer .xlsx que conté dues fulles. En la primera fulla (Data) podem trobar els valors de les concentracions de diferents metabòlits per les diferents mostres, que també tenen altres dades com el tipus o classe de mostra i el seu identificador. En la segona (Peak) podem trobar la metadata associada a les pròpies variables, com per exemple el nom complet de cada metabòlit

Eines informàtiques

El treball es realitza en un markdown de R studio. Per poder manipular les dades se'ns demana poder crear un contenidor de tipus SummarizedExperiment de Bioconductor. Per fer-ho agafem el codi que podem trobar a la web de Bioconductor: *https://bioconductor.org/packages/release/bioc/html/SummarizedExperiment.html*

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")

BiocManager::install("SummarizedExperiment")
```

```
## Bioconductor version 3.20 (BiocManager 1.30.25), R 4.4.1 (2024-06-14 u
crt)
```

```
## Warning: package(s) not installed when version(s) same as or greater t
han current; use
##   `force = TRUE` to re-install: 'SummarizedExperiment'
```

```
## Installation paths not writeable, unable to update packages
##   path: C:/Program Files/R/R-4.4.1/library
##   packages:
##     boot, foreign, MASS, Matrix, nlme, survival
```

```
## Old packages: 'curl', 'xfun'
```

Per altra banda, també necessitem altres paquets per poder treballar, per exemple amb gràfics o bé per llegir les dades:

Un cop hem descarregat el paquet, necessitem carregar-lo:

```
library(SummarizedExperiment)
```

```
## S'està carregant el paquet requerit: MatrixGenerics
```

```
## S'està carregant el paquet requerit: matrixStats
```

```
##
## S'està adjuntant el paquet: 'MatrixGenerics'
```

```
## Els següents objectes estan emmascarats des de '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

## S'està carregant el paquet requerit: GenomicRanges

## S'està carregant el paquet requerit: stats4

## S'està carregant el paquet requerit: BiocGenerics

##
## S'està adjuntant el paquet: 'BiocGenerics'

## Els següents objectes estan emmascarats des de 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## Els següents objectes estan emmascarats des de '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

## S'està carregant el paquet requerit: S4Vectors

##
## S'està adjuntant el paquet: 'S4Vectors'

## L'objecte següent està emmascarat per 'package:utils':
##
##     findMatches

## Els següents objectes estan emmascarats des de 'package:base':
##
##     expand.grid, I, unname

## S'està carregant el paquet requerit: IRanges

##
## S'està adjuntant el paquet: 'IRanges'
```

```
## L'objecte següent està emmascarat per 'package:grDevices':
##
##     windows

## S'està carregant el paquet requerit: GenomeInfoDb

## S'està carregant el paquet requerit: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## S'està adjuntant el paquet: 'Biobase'

## L'objecte següent està emmascarat per 'package:MatrixGenerics':
##
##     rowMedians

## Els següents objectes estan emmascarats des de 'package:matrixStats':
##
##     anyMissing, rowMedians

# També carreguem readxl per llegir el fitxer excel
library(readxl)

# I el dplyr per fer algunes operacions (com el pipe %>% o el select)
library(dplyr)

##
## S'està adjuntant el paquet: 'dplyr'

## L'objecte següent està emmascarat per 'package:Biobase':
##
##     combine

## Els següents objectes estan emmascarats des de 'package:GenomicRanges'
:
##
##     intersect, setdiff, union

## L'objecte següent està emmascarat per 'package:GenomeInfoDb':
##
##     intersect

## Els següents objectes estan emmascarats des de 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## Els següents objectes estan emmascarats des de 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union

## Els següents objectes estan emmascarats des de 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union

## L'objecte següent està emmascarat per 'package:matrixStats':
##
##     count

## Els següents objectes estan emmascarats des de 'package:stats':
##
##     filter, lag

## Els següents objectes estan emmascarats des de 'package:base':
##
##     intersect, setdiff, setequal, union
```

Creació del contenidor de tipus SummarizedExperiment:

En escollir el dataset "2023-CIMCBTutorial" podem veure com dins hi ha un arxiu .xlsx que haurem de llegir mitjançant read_excel (carregant primer el readxl). Podem veure tota la informació sobre aquestes dades al següent enllaç: (*https://cimcb.github.io/MetabWorkflowTutorial/Tutorial1.html*)

El document té dues fulles, la primera conté la concentració de 149 metabòlits (columnes) i 4 columnes inicials: índex, SampleID, SampleType (si són mostres reals o Pooled QC) i la classe (el diagnòstic dels individus, GC = Gastric Cancer , BN = Benign Tumor , HE = Healthy Control).

Per fer el SummarizedExperiment necessitem una "expression matrix" (**assay**), una taula per descriure les mostres (**sample metadata slot o ColData**) i la taula que descrigui les dades (gens, metabòlits,...) (**features metadata o RowData**). Podem trobar-ne un exemple a: *https://uclouvain-cbio.github.io/bioinfo-training-02-rnaseq/summarizedexperiments.html*

```
# Importem l'arxiu excel:

library(readxl)
GastricCancer_NMR <- read_excel("metaboData-main/metaboData-main/Datasets
/2023-CIMCBTutorial/GastricCancer_NMR.xlsx")
View(GastricCancer_NMR)



# Com ens diu en la web que descriu dataset, carreguem cadascun dels dos
fulls d'excel en dues variables diferents (però ho fem en R):
```

```r
# Carreguem la primera fulla (Data):
data <- read_excel("metaboData-main/metaboData-main/Datasets/2023-CIMCBTu
torial/GastricCancer_NMR.xlsx", sheet = "Data")

# Carreguem la segona fulla (Peak):
peak <- read_excel("metaboData-main/metaboData-main/Datasets/2023-CIMCBTu
torial/GastricCancer_NMR.xlsx", sheet = "Peak")
```

Creació de la "Expression Matrix"

Visualitzant l'arxiu veiem que ens interessen les columnes de la 5 a la 153 (ja que són les que contenen els metabòlits). Seleccionem de "data" aquestes columnes i, amb %>% ho passem a as.matrix() per tenir-lo en forma de matriu necessària.

```r
# Construïm la matriu a partir del dataframe "data" selecccionant les col
umnes que ens interessen:
count_matrix <- data[, 5:153] %>%
  as.matrix()

# Transposem la matriu, això ho fem perquè a la matriu les columnes corre
sponen a cada mostra mentre que al data frame la disposició era al revès,
 amb les mostres ordeandes per files.
count_matrix = t(count_matrix)

# Podem afecgir el nom de les files agafant el propi Sample ID de "data"
i així els fem coincidir.
colnames(count_matrix) <- data$SampleID

# Mirem uns quants registres per veure si l'estructura de la matriu és la
 que esperem
count_matrix[1:10, 1:10]
```

```
##     sample_1 sample_2 sample_3 sample_4 sample_5 sample_6 sample_7 sam
ple_8
## M1      90.1     43.0    214.3     31.6     81.9    196.9     45.5
 91.0
## M2     491.6    525.7  10703.2     59.7    258.7    128.2    190.4
231.9
## M3     202.9    130.2    104.7     86.4    315.1    862.5     32.0
212.5
## M4      35.0       NA     46.8     14.0      8.7     18.7       NA
 18.2
## M5     164.2    694.5    483.4     88.6    243.2    200.1    362.7
 72.5
## M6      19.7    114.5    152.3     10.3     18.4      4.7     35.7
  6.7
## M7      41.0     37.9    110.1    170.3    349.4     37.3     59.6
 15.3
## M8      46.5    125.7     85.1     23.9     61.1    243.7     51.3
 37.1
```

```
## M9      17.3     57.8    238.3       NA     12.2    293.3       NA
 22.7
## M10    106.8       NA     48.0       NA     72.9    113.1     60.1
 47.8
##     sample_9 sample_10
## M1     480.6      62.2
## M2     470.3     181.5
## M3      60.7      75.5
## M4       8.4      36.0
## M5     270.2     203.4
## M6      57.4      18.7
## M7     213.8      44.4
## M8      65.6      48.6
## M9      59.5      47.2
## M10    148.9     153.8

# Com podem veure, tenim tots els metabòlits organitzats en files i les m
ostres són les columnes.
```

Creació de la taula amb les dades de la mostra, el que seria el ColData:

Ara necessitem les dades addicionals sobre les mostres, és a dir, les descripcions d'aquestes que es troben a les columnes SampleID, SampleType i Class.

```
# Seleccionem del data.frame "data" les columnes que necessitem emprant %
>% i la funció select():
sample_metadata <- data %>%
  select(SampleID, SampleType, Class)

rownames(sample_metadata) <- data$SampleID

## Warning: Setting row names on a tibble is deprecated.

# Veiem com queda la taula (que només ha de tenir 3 columnes)
head(sample_metadata)

## # A tibble: 6 × 3
##   SampleID SampleType Class
##   <chr>    <chr>      <chr>
## 1 sample_1 QC         QC
## 2 sample_2 Sample     GC
## 3 sample_3 Sample     BN
## 4 sample_4 Sample     HE
## 5 sample_5 Sample     GC
## 6 sample_6 Sample     BN

# Com podem comprovar, les files d'aquesta taula corresponen a les column
es de la matriu anterior.
```

Creació de la taula amb les dades de les característiques (RowData):

Podem trobar les dades de les característiques per obtenir la part RowData del SummarizedExperiment al full dos del fitxer .xlsx (Peak), on tenim les columnes: Label (el nom de cada metabòlit), Name (el nom dels metabòlits donat a la capçalera de la taula "Data", és a dir M1, M2, etc.), Perc_missing (% de missing data de cada metabòlit en el conjunt de mostres), QC_RSD (quality score dels metabòlits comparats entre totes les mostres). Fem com hem fet abans:

```
#sleccionem totes les columnes (menys la primera, índex, que no aporta re
s)
metabolite_metadata <- peak %>%
  select(Label, Name, Perc_missing, QC_RSD) %>%
  as.data.frame()

# Fem que les files tinguin el mateix nom que a la matriu (és a dir el Na
me)
rownames(metabolite_metadata) <- metabolite_metadata$Name

# Consultem com queda la taula
head(metabolite_metadata)
```

```
##                       Label Name Perc_missing    QC_RSD
## M1       1_3-Dimethylurate   M1   11.4285714 32.208005
## M2 1_6-Anhydro-β-D-glucose   M2    0.7142857 31.178028
## M3    1_7-Dimethylxanthine   M3    5.0000000 34.990605
## M4    1-Methylnicotinamide   M4    8.5714286 12.804201
## M5          2-Aminoadipate   M5    1.4285714  9.372664
## M6         2-Aminobutyrate   M6    5.0000000 46.977149
```

```
# Com podem veure, les files d'aquesta taula corresponen a les files de l
a matriu.

head(rownames(metabolite_metadata))
```

```
## [1] "M1" "M2" "M3" "M4" "M5" "M6"
```

```
head(rownames(count_matrix))
```

```
## [1] "M1" "M2" "M3" "M4" "M5" "M6"
```

```
head(rownames(sample_metadata))
```

```
## [1] "sample_1" "sample_2" "sample_3" "sample_4" "sample_5" "sample_6"
```

Verifiquem que les mides siguen correctes:

```
dim(count_matrix)
```

```
## [1] 149 140
```

```
dim(sample_metadata)
```

```
## [1] 140   3

dim(metabolite_metadata)

## [1] 149   4
```

Creació del contenidor SummarizedExperiment:

Ara, amb tots tres elements podem fer l'objecte SummarizedExperiment en combinar-los de la següent manera:

```
# Seguint la idea que les columnes de la matriu corresponen a les files d
el Coldata (sample_metadata) i que les files de la matriu corresponen a l
es files del Rowdata (metabolite_metadata) fem la següen verificació aban
s de continuar:
stopifnot(all(colnames(count_matrix) == rownames(sample_metadata)))
stopifnot(all(rownames(count_matrix) == rownames(metabolite_metadata)))

# Construïm el contenidor amb els tres elements:
se <- SummarizedExperiment(assays = list(counts = count_matrix),
                           colData = sample_metadata,
                           rowData = metabolite_metadata)

# Visualitzem les característiques del contenidor
se

## class: SummarizedExperiment
## dim: 149 140
## metadata(0):
## assays(1): counts
## rownames(149): M1 M2 ... M148 M149
## rowData names(4): Label Name Perc_missing QC_RSD
## colnames(140): sample_1 sample_2 ... sample_139 sample_140
## colData names(3): SampleID SampleType Class

# El podem veure amb detall emprant diferents funcions:
#head(assay(se))
#colData(se)
#head(rowData(se))
```

Guardar el contenidor en format binari .Rda:

Seguin l'exemple d'aquest enllaç podem guardar el contenidor en un arxiu a part: ### Guardar el contenidor en format .Rda

https://www.bioconductor.org/packages//release/bioc/vignettes/recountmethylation/inst/doc/exporting_saving_data.html

```
save(se, file = "C:/Users/dvd93/OneDrive/Escritorio/MÁSTER 2/ANÀLISI DE D
ADES ÒMIQUES/PAC 1/contenidor.rda")
```

Creació del repositori a Github:

Resultats

Ara que ja tenim les dades podem començar a treballar-hi. Podem

Anàlisi estadístic de les dades

Ara amb les dades guardades ja hi podem treballar. En aquest cas, per centrar-me en un cas més concret, m'interessa veure la relació entre el grup d'individus amb càncer gàstric (els que tenen la classe "GC") en relació als sans (amb la classe "HE"). Per fer-ho, podem crear un subset del SummarizedExperiment, una de les avantages que té aquest tipus d'objecte. L'anomenem "se_gc":

Subset i processat de les dades:

```
# Generem un subset del SummarizedExperiment complet "se" per quedar-nos amb els individus de càncer gàstric i els sans, per analitzar-los.

se_gc <- se[, se$Class %in% c("GC", "HE")]

#amb %in% podem seleccionar els que compleixin que a Class tenen els valors indicats https://rsanchezs.gitbooks.io/rprogramming/content/chapter9/filter.html



# Tal com s'indica en la web del dataset, se'ns recomana eliminar tots els metabòlits amb % de missing values elevat (no interessen els >10%) i tots els que tinguin una valor de qualitat QC_RSD major a 20. Per eliminar aquests metabòlits hem d'anar al rowData per filtrar a partir del que seria la taula de "features metadata".

se_gc <- se_gc[rowData(se_gc)$Perc_missing < 10 & rowData(se_gc)$QC_RSD < 20, ]

# Per últim, hem d'eliminar tots el valors nuls que hi poden haver ja que ens poden afectar als anàlisis i provocar errors. Per fer-ho eliminem les mostres (columnes) amb valors nuls. La idea és accedir a la matriu, mirar amb is.na si hi ha valors nuls i comptar-los amb colSum(). Si n'hi ha serà diferent a 0 i es posarà False i si no n'hi ha serà True. Així tindrem un vector amb Trues i Falses i el podem fer servir per filtrar el subset ja que només ens quedarem amb les columnes amb True. La idea es pot consultar en el següent tutorial: https://stackoverflow.com/questions/25188051/using-is-na-in-r-to-get-column-names-that-contain-na-values


columnes_utils <- colSums(is.na(assay(se_gc))) == 0
se_gc <- se_gc[, columnes_utils]
```

```r
#podem veure com queda accedint a les dades per veure que només tenim els
 que ens interessen:
#head(assay(se_gc))
se_gc  #veiem que se'ns queden 52 metabòlits i 40 mostres i podem treball
ar de forma més senzilla i neta
```

```
## class: SummarizedExperiment
## dim: 52 40
## metadata(0):
## assays(1): counts
## rownames(52): M4 M5 ... M148 M149
## rowData names(4): Label Name Perc_missing QC_RSD
## colnames(40): sample_8 sample_9 ... sample_137 sample_139
## colData names(3): SampleID SampleType Class
```

```r
head(colData(se_gc))
```

```
## DataFrame with 6 rows and 3 columns
##              SampleID  SampleType       Class
##           <character> <character> <character>
## sample_8     sample_8      Sample          HE
## sample_9     sample_9      Sample          GC
## sample_12   sample_12      Sample          HE
## sample_15   sample_15      Sample          HE
## sample_18   sample_18      Sample          HE
## sample_20   sample_20      Sample          GC
```

```r
head(rowData(se_gc))
```

```
## DataFrame with 6 rows and 4 columns
##                     Label        Name Perc_missing    QC_RSD
##               <character> <character>    <numeric> <numeric>
## M4  1-Methylnicotinamide          M4      8.57143  12.80420
## M5        2-Aminoadipate          M5      1.42857   9.37266
## M7       2-Furoylglycine          M7      2.85714   5.04916
## M8  2-Hydroxyisobutyrate          M8      0.00000   5.13234
## M11   3-Aminoisobutyrate         M11      5.00000  15.47616
## M14 3-Hydroxyisobutyrate         M14      2.14286   8.90571
```

Visualització de les dades i resum

```r
# Per algunes funcions de R necessitem que les variables estiguin en les
columnes, una cosa que en SummarizedExperiment és al revés. Podem transpo
sar la matriu amb t() accedint a la matriu amb assay(). La guardem en una
 variable:

matriu = assay(se_gc) # accedim a la matriu amb assay i la guardem en una
 variable
matriu_t = t(matriu)  #ara això és la matriu transposada (columnes = meta
bòlits)

#Podem fer un petit anàlisi amb str. M'he adonat que no donava el resulta
```

```
t esperar i és perquè amb el SummarizedExperiment tenim una matriu, no un
 dataframe. Podem generar un dataframe per treballar en alguns casos:

se_df <- as.data.frame(matriu_t)  #ara això és un dataframe de la matriu
transposta per usar-la en algunes funcions

# Fem dos anàlisis genèrics de les dades amb str i summary
str(se_df)
```

```
## 'data.frame':    40 obs. of  52 variables:
##  $ M4  : num  18.2 8.4 45 70.6 13.4 23.7 51.4 58.5 10.3 13.8 ...
##  $ M5  : num  72.5 270.2 62.6 65.4 51.2 ...
##  $ M7  : num  15.3 213.8 42.4 26.2 23.6 ...
##  $ M8  : num  37.1 65.6 68 81.2 27.9 ...
##  $ M11 : num  54.1 92.9 100.7 73.7 58.2 ...
##  $ M14 : num  30.3 61.9 45.5 95.8 25.4 ...
##  $ M15 : num  19.2 54.2 60.8 48.8 24.8 67.4 157 70.7 51.8 19.1 ...
##  $ M25 : num  6.6 39.6 14.2 13.5 5.3 17.1 42.8 21.2 12.5 11.8 ...
##  $ M26 : num  29.3 20.8 25 39.9 2 13.8 36 30.3 4.1 11.1 ...
##  $ M31 : num  9.9 67.7 62.5 1.2 3.6 ...
##  $ M32 : num  38.7 444.9 81.5 177.9 81.2 ...
##  $ M33 : num  250 324 171 331 136 ...
##  $ M36 : num  4 6.7 13.4 49.3 4.5 ...
##  $ M37 : num  13.9 150.5 121.2 48.6 5.3 ...
##  $ M45 : num  676 1978 4205 6639 1093 ...
##  $ M48 : num  2665 6864 10177 15850 4778 ...
##  $ M50 : num  61.6 0.2 131.9 143.4 11.3 ...
##  $ M51 : num  8.8 354.3 289.6 654.4 135.6 ...
##  $ M65 : num  160 331 204 754 139 ...
##  $ M66 : num  1436 2155 1080 1110 161 ...
##  $ M68 : num  266.1 105.6 91.4 753.5 0.1 ...
##  $ M71 : num  14 28.3 38.8 34.1 7.9 ...
##  $ M73 : num  37.1 84.7 78 121.9 16 ...
##  $ M74 : num  6.2 29.1 21.6 18.7 5.2 25.5 84.3 34.8 17.7 6.3 ...
##  $ M75 : num  66.9 491.5 74.6 159.5 37.7 ...
##  $ M88 : num  67.6 155.4 156.6 166.3 54.5 ...
##  $ M89 : num  78.4 696.9 353.3 451.7 89.6 ...
##  $ M90 : num  31.2 59.4 32.2 157.8 30 ...
##  $ M91 : num  20.7 34.9 85.9 110.7 28.5 ...
##  $ M93 : num  35.4 30 33.9 91.9 33.4 ...
##  $ M101: num  6.8 26.5 43.3 38.6 0.1 ...
##  $ M104: num  143.1 371 328.4 269 52.6 ...
##  $ M105: num  66.9 38.7 176.3 289.8 11 ...
##  $ M106: num  20.5 59.9 88.1 21.1 31.9 ...
##  $ M107: num  71.5 350.2 356.4 341.3 85.2 ...
##  $ M110: num  123.6 44.5 108.5 9.3 28.5 ...
##  $ M115: num  60.7 392.5 352.1 26.9 11.6 ...
##  $ M116: num  3.7 38.7 35.3 47 26.6 ...
##  $ M118: num  82.3 45.5 166.4 194.8 6.2 ...
##  $ M119: num  14.4 57.8 83.4 71.4 21 ...
```

```
##  $ M120: num  7.5 97.8 31.7 83.7 28 ...
##  $ M122: num  8.2 53.7 17.2 26.5 10.6 17.7 41.7 26.6 28.6 20.9 ...
##  $ M126: num  20 51.5 46.3 50.6 9.1 ...
##  $ M129: num  1068 2658 1303 2833 300 ...
##  $ M130: num  57 19.2 12.9 2.5 5.3 ...
##  $ M134: num  888 1720 764 1204 154 ...
##  $ M137: num  260.8 368.4 617.4 99.2 180.4 ...
##  $ M138: num  89.3 1317.2 101.8 2.1 31.3 ...
##  $ M142: num  7.1 54 22.8 3.7 4.6 10 16.6 8 3.9 2.3 ...
##  $ M144: num  29.7 29.3 25.9 26.5 25.6 26.4 29.4 25.3 29.1 24.9 ...
##  $ M148: num  18 106 159 405 1 ...
##  $ M149: num  81.6 197.2 185.6 129.5 47.9 ...
```

```
summary(se_df)
```

```
##        M4                M5                M7                M8        
##  Min.   :  1.60   Min.   :   2.7   Min.   :  8.60   Min.   : 12.00  
##  1st Qu.: 15.65   1st Qu.:  56.0   1st Qu.: 16.52   1st Qu.: 32.02  
##  Median : 25.45   Median : 126.8   Median : 34.85   Median : 50.80  
##  Mean   : 35.90   Mean   : 243.2   Mean   : 89.22   Mean   : 59.49  
##  3rd Qu.: 50.73   3rd Qu.: 286.0   3rd Qu.: 81.67   3rd Qu.: 72.95  
##  Max.   :141.80   Max.   :2503.0   Max.   :492.60   Max.   :207.40  
##       M11               M14               M15               M25        
##  Min.   :   0.7   Min.   : 12.90   Min.   : 13.00   Min.   :  3.90  
##  1st Qu.:  48.3   1st Qu.: 34.58   1st Qu.: 27.60   1st Qu.: 11.05  
##  Median : 104.9   Median : 52.40   Median : 46.50   Median : 16.70  
##  Mean   : 178.8   Mean   : 76.05   Mean   : 57.03   Mean   : 25.89  
##  3rd Qu.: 187.5   3rd Qu.: 92.72   3rd Qu.: 68.22   3rd Qu.: 31.27  
##  Max.   :1688.2   Max.   :295.80   Max.   :195.60   Max.   :171.80  
##       M26               M31               M32               M33        
##  Min.   :  2.00   Min.   :  1.20   Min.   :  0.60   Min.   : 102.1  
##  1st Qu.: 12.38   1st Qu.: 11.53   1st Qu.: 90.03   1st Qu.: 222.3  
##  Median : 23.45   Median : 31.60   Median :174.55   Median : 313.1  
##  Mean   : 37.75   Mean   : 55.52   Mean   :224.46   Mean   : 389.9  
##  3rd Qu.: 35.33   3rd Qu.: 67.03   3rd Qu.:313.93   3rd Qu.: 496.4  
##  Max.   :374.60   Max.   :264.90   Max.   :874.20   Max.   :1070.4  
##       M36               M37               M45               M48        
##  Min.   :  4.00   Min.   :  4.30   Min.   :   69.8   Min.   : 2043  
##  1st Qu.: 13.97   1st Qu.: 23.00   1st Qu.: 1431.0   1st Qu.: 5799  
##  Median : 30.00   Median : 62.65   Median : 2548.6   Median : 8111  
##  Mean   : 52.16   Mean   :100.64   Mean   : 3943.4   Mean   :10579  
##  3rd Qu.: 76.60   3rd Qu.:131.38   3rd Qu.: 5128.7   3rd Qu.:13392  
##  Max.   :241.50   Max.   :520.70   Max.   :16673.9   Max.   :32822  
##       M50               M51               M65               M66        
##  Min.   :   0.2   Min.   :   8.8   Min.   : 111.6   Min.   :  161.0  
##  1st Qu.: 116.8   1st Qu.: 189.7   1st Qu.: 224.6   1st Qu.:  636.2  
##  Median : 191.2   Median : 421.3   Median : 354.1   Median : 1111.0  
##  Mean   : 343.2   Mean   : 445.9   Mean   : 521.4   Mean   : 2207.6  
##  3rd Qu.: 286.4   3rd Qu.: 643.2   3rd Qu.: 702.4   3rd Qu.: 2228.8  
##  Max.   :4037.5   Max.   :1264.2   Max.   :2138.8   Max.   :16544.5
```

```
##       M68              M71              M73              M74        
##  Min.   :  0.10   Min.   :  5.50   Min.   : 16.00   Min.   :  0.10  
##  1st Qu.: 46.33   1st Qu.: 16.15   1st Qu.: 47.27   1st Qu.: 10.53  
##  Median :141.00   Median : 37.50   Median : 94.10   Median : 20.55  
##  Mean   :182.96   Mean   : 48.38   Mean   : 99.07   Mean   : 32.90  
##  3rd Qu.:231.30   3rd Qu.: 63.35   3rd Qu.:138.05   3rd Qu.: 35.08  
##  Max.   :753.50   Max.   :190.70   Max.   :243.70   Max.   :171.00  
##       M75              M88              M89              M90        
##  Min.   : 34.70   Min.   :  0.50   Min.   :  56.6   Min.   :  4.00  
##  1st Qu.: 67.28   1st Qu.: 88.25   1st Qu.: 262.1   1st Qu.: 33.62  
##  Median :109.85   Median :152.75   Median : 510.6   Median : 71.00  
##  Mean   :169.08   Mean   :189.46   Mean   : 713.2   Mean   :109.59  
##  3rd Qu.:199.32   3rd Qu.:266.38   3rd Qu.: 994.8   3rd Qu.:144.45  
##  Max.   :650.70   Max.   :718.90   Max.   :2402.9   Max.   :514.20  
##       M91              M93              M101             M104       
##  Min.   : 10.90   Min.   : 10.30   Min.   :  0.10   Min.   :  17.5  
##  1st Qu.: 43.27   1st Qu.: 35.70   1st Qu.: 18.43   1st Qu.: 176.3  
##  Median : 65.00   Median : 55.25   Median : 33.15   Median : 277.7  
##  Mean   : 82.34   Mean   : 74.59   Mean   : 42.22   Mean   : 363.7  
##  3rd Qu.:110.88   3rd Qu.: 97.47   3rd Qu.: 43.08   3rd Qu.: 429.3  
##  Max.   :254.90   Max.   :202.20   Max.   :218.50   Max.   :1579.2  
##       M105             M106             M107             M110       
##  Min.   :   0.1   Min.   :  7.50   Min.   :   0.4   Min.   :  2.30  
##  1st Qu.:  36.2   1st Qu.: 31.82   1st Qu.: 177.0   1st Qu.: 35.00  
##  Median :  63.8   Median : 52.00   Median : 337.9   Median : 57.55  
##  Mean   : 200.7   Mean   : 63.69   Mean   : 407.8   Mean   : 68.94  
##  3rd Qu.: 177.6   3rd Qu.: 73.92   3rd Qu.: 568.6   3rd Qu.: 86.42  
##  Max.   :2182.2   Max.   :257.50   Max.   :1350.7   Max.   :280.90  
##       M115             M116             M118             M119       
##  Min.   :   6.90   Min.   :  3.70   Min.   :   4.60   Min.   : 13.70  
##  1st Qu.:  42.12   1st Qu.: 14.05   1st Qu.:  63.55   1st Qu.: 37.50  
##  Median :  80.55   Median : 21.95   Median : 153.45   Median : 77.70  
##  Mean   : 242.59   Mean   : 30.15   Mean   : 215.25   Mean   : 83.47  
##  3rd Qu.: 279.90   3rd Qu.: 37.05   3rd Qu.: 279.77   3rd Qu.:107.38  
##  Max.   :2134.50   Max.   :156.00   Max.   :1005.00   Max.   :221.20  
##       M120             M122             M126             M129       
##  Min.   :  0.40   Min.   :  2.80   Min.   :  2.50   Min.   : 300.1  
##  1st Qu.: 37.40   1st Qu.: 13.10   1st Qu.: 23.95   1st Qu.:1034.8  
##  Median : 66.30   Median : 23.35   Median : 44.70   Median :1730.5  
##  Mean   : 88.33   Mean   : 29.11   Mean   : 71.47   Mean   :2182.9  
##  3rd Qu.:127.53   3rd Qu.: 34.65   3rd Qu.: 65.12   3rd Qu.:2691.7  
##  Max.   :317.90   Max.   :158.60   Max.   :609.40   Max.   :8038.2  
##       M130             M134             M137             M138       
##  Min.   :   0.20   Min.   : 113.5   Min.   :  67.9   Min.   :   2.1  
##  1st Qu.:  19.50   1st Qu.: 655.0   1st Qu.: 259.2   1st Qu.: 113.3  
##  Median :  38.95   Median :1076.0   Median : 464.1   Median : 745.0  
##  Mean   : 102.06   Mean   :1591.8   Mean   : 806.8   Mean   :1067.8  
##  3rd Qu.:  68.12   3rd Qu.:2068.9   3rd Qu.:1135.6   3rd Qu.:1612.5  
##  Max.   :1188.70   Max.   :8567.8   Max.   :5394.5   Max.   :4476.4  
##       M142             M144             M148             M149       
```

```
##  Min.   :  0.10   Min.   : 21.90   Min.   :   1.0   Min.   : 28.9
##  1st Qu.:  3.85   1st Qu.: 25.45   1st Qu.:  77.0   1st Qu.:104.0
##  Median : 10.05   Median : 26.80   Median : 157.2   Median :167.2
##  Mean   : 24.69   Mean   : 34.10   Mean   : 345.3   Mean   :176.2
##  3rd Qu.: 23.95   3rd Qu.: 29.15   3rd Qu.: 495.6   3rd Qu.:221.4
##  Max.   :182.30   Max.   :191.40   Max.   :2560.3   Max.   :401.7
```

Matriu de covariàncies

```
#Escalem les variables de la matriu centrant a cada columna en la seva mi
tjana.
matriu_t_scale <- scale(matriu_t, center = TRUE, scale = FALSE)

# Calcula la media de cada columna (es decir, de cada variable):
# Podem fer na.rm = True per obviar els valors nuls) https://www.datacamp
.com/tutorial/na-rm-in-r
apply(matriu_t_scale, 2, mean)
```

```
##            M4            M5            M7            M8           M11
## -9.769529e-16 -4.277481e-15  5.348152e-16  2.487160e-15  0.000000e+00
##           M14           M15           M25           M26           M31
##  1.595946e-15  2.931422e-15  1.065207e-15 -1.686628e-15 -2.487333e-15
##           M32           M33           M36           M37           M45
## -1.136487e-14 -1.776357e-14  1.155066e-15  7.112366e-16  6.813994e-14
##           M48           M50           M51           M65           M66
## -3.637313e-13  3.270578e-14 -6.403211e-15 -4.973799e-14 -1.079447e-13
##           M68           M71           M73           M74           M75
## -7.810072e-15 -2.841997e-15 -4.617660e-15 -3.375685e-15  1.313810e-14
##           M88           M89           M90           M91           M93
## -1.385524e-14 -4.833772e-14 -5.329071e-15 -5.863712e-15 -5.152476e-15
##          M101          M104          M105          M106          M107
## -8.881784e-16  1.491723e-14 -1.065814e-14 -2.841390e-15  4.971717e-15
##          M110          M115          M116          M118          M119
##  6.571566e-15 -1.065814e-14 -1.021969e-15 -4.973105e-15  1.420739e-15
##          M120          M122          M126          M129          M130
##  4.795643e-15 -7.996642e-16 -6.217249e-15 -1.534189e-13 -1.776357e-15
##          M134          M137          M138          M142          M144
##  7.105427e-14 -4.829748e-14 -5.969253e-14  6.207708e-16 -2.840675e-15
##          M148          M149
##  1.990699e-14  3.904169e-15
```

Ara, amb les dades centrades de matriu_t_scale podem fer la matriu de covariàncies. Primer necessitem calcular el valor de n, que correspon al número de mostres (com hem vist abans quedava a 40 després d'eliminar els valors nuls).

```
dim(se_df)
```

```
## [1] 40 52
```

```
dim(matriu_t_scale)
```

```
## [1] 40 52
```

```
# calculem la matriu de variàncies:

n<- dim(se_df)[1]
S<-cov(matriu_t_scale)*(n-1)/n

#Donat a la grandària de la matriu, comento la línia per no donar la sort
ida massa llarga.
#show(S)
```

Matriu de correlacions

Ara, podem, amb les mateixes dades, calcular la matriu de correlacions per veure com es relacionen els diferents metabòlits entre sí.

```
R<-cor(matriu_t_scale)

#De la mateixa forma, comento la línia per evitar que es vegi una matriu
massa llarga a l'informe.
#show(R)
```

Anàlisi de les components principals (PCA)

Calculem les components principals a partir de diagonalització de la matriu de covariàncies

```
EIG <- eigen(S)
#Ho podem veure tot amb show pero comento la línia per evitar una sortida
 massa llarga
#show(EIG)

#Individualment podem veure els valors:
EIG$values
```

```
##  [1]  6.122632e+07  1.173102e+07  5.587031e+06  3.113594e+06  1.389738
e+06
##  [6]  7.599520e+05  5.655010e+05  3.701926e+05  1.729711e+05  1.309849
e+05
## [11]  1.114495e+05  6.014413e+04  4.170365e+04  3.989831e+04  2.385131
e+04
## [16]  1.886574e+04  1.625106e+04  1.387419e+04  9.562762e+03  5.300002
e+03
## [21]  4.440317e+03  4.068069e+03  2.575680e+03  2.041130e+03  1.662123
e+03
## [26]  1.597503e+03  1.108346e+03  7.370915e+02  6.452660e+02  4.682692
e+02
## [31]  3.319043e+02  2.630437e+02  2.089284e+02  1.528592e+02  1.021077
e+02
## [36]  5.245173e+01  3.132611e+01  2.075954e+01  1.158857e+01  1.672325
e-09
## [41]  7.381722e-10  4.446287e-10  2.784981e-10  2.306224e-10  1.339107
e-10
## [46]  7.563635e-11  1.768730e-12 -2.971044e-12 -3.129654e-12 -2.407277
```

```
e-11
## [51] -6.968600e-10 -7.795020e-10

# I els vectors (la primera part amb head)
head(EIG$vectors)

##              [,1]         [,2]          [,3]          [,4]          [,
5]
## [1,] -0.001849139  0.000874341  1.513072e-05  0.0001876913  0.00095906
59
## [2,] -0.024073646 -0.076802129  2.311585e-02  0.0306903239 -0.06943175
69
## [3,] -0.002380213 -0.020823581 -2.377972e-02 -0.0105759588 -0.02793380
13
## [4,] -0.004183835 -0.001502140 -2.579751e-03 -0.0028807005  0.00400570
08
## [5,] -0.006171962 -0.006885990  2.453004e-02  0.0001007958 -0.08542273
86
## [6,] -0.004895268 -0.002229540 -3.162800e-03 -0.0178140958  0.00871696
68
##              [,6]          [,7]         [,8]         [,9]       [,10]
## [1,]  0.007315491  0.0045741010  0.001055426  0.008270285 -0.02466562
## [2,]  0.035877588 -0.0509376562 -0.052611227  0.030695270 -0.25106997
## [3,] -0.002913067  0.0349154255 -0.028889435  0.087301610 -0.04231847
## [4,]  0.003713499  0.0003015685  0.008785354 -0.011908268  0.02545399
## [5,]  0.044030066  0.0096785881  0.020416385 -0.196006230  0.03716488
## [6,]  0.014487527  0.0033890253  0.014243173 -0.015205102  0.04185587
##             [,11]        [,12]       [,13]         [,14]        [,15]
## [1,]  0.007492366  0.014686718  0.01318060 -0.0105090859 -0.032915117
## [2,] -0.256939609 -0.476454721 -0.16291603  0.1263110560 -0.216922930
## [3,]  0.069654675  0.038065549 -0.09233076 -0.0249108057  0.087924611
## [4,]  0.009240819  0.006783907 -0.02003166 -0.0008999676  0.003751899
## [5,]  0.515319920 -0.192453750  0.62767587 -0.1530487299 -0.086925007
## [6,]  0.045194475  0.012445542 -0.00492740  0.0196850730 -0.074958249
##             [,16]        [,17]       [,18]       [,19]       [,20]
   [,21]
## [1,] -0.013107730 -0.021568690  0.03962182 -0.01314202 -0.07422748  0.
087473644
## [2,]  0.181223050 -0.525888693  0.03411531 -0.06791068 -0.17579698 -0.
002628218
## [3,]  0.085271239 -0.007224626 -0.15291432  0.17854302 -0.09312202  0.
017904381
## [4,] -0.001392149  0.016899544 -0.01767003 -0.04094654  0.01716744 -0.
064092567
## [5,] -0.009721318 -0.231507504 -0.25775584 -0.10638946 -0.04008074 -0.
080517810
## [6,]  0.005951642 -0.007420448 -0.01603781 -0.07448143  0.10672295  0.
040778938
##            [,22]       [,23]       [,24]       [,25]       [,26]
 [,27]
```

```
## [1,] -0.03930646  0.16782817  0.16307818 -0.14085678  0.09202955  0.03
210141
## [2,]  0.24449008  0.08826929 -0.20234766 -0.02870572  0.10206101  0.05
514636
## [3,]  0.07526151 -0.33711324  0.18968874 -0.17910436 -0.09593612 -0.02
349277
## [4,]  0.03481085  0.02087417  0.08716117 -0.04192650 -0.02990527  0.07
455502
## [5,] -0.01450009 -0.08760029 -0.12941184  0.11765154 -0.03205080  0.02
931587
## [6,]  0.05242390  0.03884469 -0.03191996  0.13682498  0.01218402 -0.06
604631
##            [,28]       [,29]        [,30]       [,31]       [,32]
   [,33]
## [1,]  0.29496019 -0.08951940 -0.194049439 -0.01440107 -0.35303683  0.3
45527272
## [2,]  0.06224917  0.17098097  0.029570294  0.04859468  0.02456818  0.0
16074046
## [3,]  0.04198340  0.11440505  0.283012659  0.59247743  0.03184258  0.0
05763044
## [4,]  0.04831148  0.02776174 -0.008031263  0.10890224 -0.03303052  0.0
52412976
## [5,]  0.16959022  0.04387287  0.012543891  0.03267261 -0.02029137 -0.0
06747733
## [6,] -0.19689675  0.16076034 -0.180725610  0.20175255  0.47374249  0.0
28242046
##            [,34]       [,35]        [,36]       [,37]       [,38]
  [,39]
## [1,]  0.20033668 -0.04789807 -0.010910706  0.09585859  0.08499615  0.2
3951891
## [2,]  0.08945789 -0.03482779  0.012240638  0.05910811 -0.02950106 -0.0
5348779
## [3,]  0.23181465 -0.06376476  0.197295906 -0.03326974 -0.13933141 -0.0
4008033
## [4,]  0.04996052  0.33056650  0.006787411  0.08896157 -0.23688503 -0.3
0183795
## [5,] -0.01043933 -0.02559847 -0.022353761 -0.02414959 -0.01527384  0.0
2221961
## [6,] -0.23581910  0.28583073  0.083026019  0.14189415  0.02860303  0.1
3807104
##              [,40]       [,41]        [,42]        [,43]       [,44]
## [1,]  0.0000000000  0.00000000 -0.351678660  0.000000000  0.00000000
## [2,] -0.0002763435  0.01794052 -0.044662506  0.003192191 -0.04466347
## [3,] -0.0771883436 -0.16880020  0.039074497 -0.059903830  0.22486930
## [4,]  0.3846252310 -0.03745707  0.288238832  0.172833271 -0.58234782
## [5,]  0.0279846303 -0.02659986  0.006111274  0.016838814  0.01176455
## [6,] -0.4212814405 -0.07923619 -0.251182215  0.075950349 -0.19107907
##             [,45]         [,46]       [,47]        [,48]        [,49]
## [1,]  0.000000000  0.0000000000  0.00000000  0.000000000  0.000000000
## [2,]  0.003630406  0.0245152352  0.01580932 -0.020478595  0.021437553
```

```
## [3,] -0.101167365 -0.0038139564  0.04632083  0.022706613  0.036566190
## [4,] -0.040861756 -0.0740259603 -0.08591906  0.047752719 -0.007554777
## [5,] -0.053356362 -0.0150365266 -0.01670032  0.002560428  0.018139378
## [6,] -0.034600878 -0.0006715634  0.07617720  0.033498242 -0.016792208
##            [,50]       [,51]        [,52]
## [1,]  0.00000000  0.00000000  0.528682705
## [2,] -0.04127329  0.00376128 -0.059639712
## [3,]  0.04835248  0.03581842  0.144458184
## [4,]  0.04323118 -0.04703151  0.257287918
## [5,] -0.02580105  0.03227295  0.005800841
## [6,]  0.03463313  0.14015467  0.275089486
```

Ara, tenim els 52 vectors, que corresponen a les components. Aquests vectors corresponen a les coordenades de les compoments principals i podem usar-los per multiplicar la matriu original amb les dades ja centrades (en el nostre cas anomenada matriu_t_scale) per fer la transformació asociada a les components principals:

```
#accedim als vectors Eigen i els emmagatzamem en una nova variable:
eigenVectors <- EIG$vectors

#transformem la matriu original multiplicant-hi els vectors:
PCAS1 <- matriu_t_scale %*% eigenVectors

# Mirem com ha quedat:
#head(PCAS1)
```

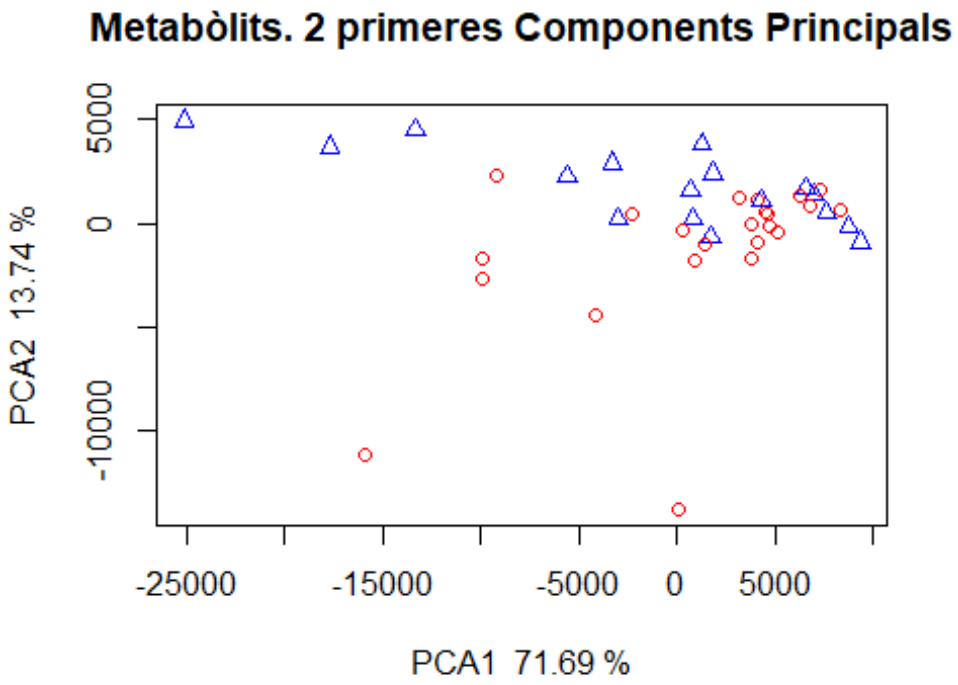
Podem representar com es relacionen les mostres en funció de les dues primeres components (és a dir, les que tenen major impacte en la variabilitat). Podem fer un plot on veiem la posició dels punts sobre els eixos de cada component. Podem afegir més informació visual si calculem el percentatge de la variabilitat explicada per cada component així com si separem per colors les mostres de cada grup (GC i HE).

Podem saber el percentatge de variabilitat explicat per cada component si accedim als valors de l'objecte EIG (de l'anàlisi Eigen) i dividim cadascun per la suma del total:

```
#accedim als valors i en fem els %:
eigenValors <- EIG$values/sum(EIG$values)

#arrodonim a 3 decimals:
round(eigenValors, 3)
```

```
##  [1] 0.717 0.137 0.065 0.036 0.016 0.009 0.007 0.004 0.002 0.002 0.001
 0.001
## [13] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000
## [25] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
 0.000
## [37] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
```

```
 0.000
## [49] 0.000 0.000 0.000 0.000

#podem veure com el primer component explica el 71.7% de la variabilitat
mentre que els segon ho fa en un 13.7%.


## VISUALITZACIÓ DEL PLOT DELS 2 PRIMERS PCs:

# Podem fer servir aquests valors per afegirlos al gràfic com etiquetes d
e cada eix (que afegirem després al plot:

xlabel = paste("PCA1 ", round(eigenValors[1]*100, 2),"%" )
ylabel = paste("PCA2 ", round(eigenValors[2]*100,2), "%")

# Ara volem separar les mostres pels dos grups (GC i HE) amb colors difer
ents:

#Seleccionem les dues classes que podem obtenir del colData del contenido
r:
classe <- colData(se_gc)$Class

colClass <- ifelse(classe == "GC", "red", "blue")
pchClass <- ifelse(classe == "GC", 1, 2)


#Expressem el plot per veure els diferents punts de cada grup (el vermell
 per GC i el blau pel HE)
plot(PCAS1[, 1], PCAS1[, 2], main = "Metabòlits. 2 primeres Components Pr
incipals", xlab=xlabel, ylab=ylabel, pch = pchClass, col = colClass, bg =
 colClass)
```



Interpretació de les components:

Un cop hem fet l'anàlisi de les components a partir de la diagonalització de la matriu de covariàncies, hem obtingut els diferents vectors. Quan fem EIG <- eigen(S) i després show(EIG) podem veure tots els vectors en forma de matriu. La primera columna correspon a la primera component principal, la segona a la segona component i així successivament. Per altra banda, cada fila correspondrà als valors dels coeficients per a cada variable, en el mateix ordre (els metabòlits M4, M5, M7, etc.). Llavors, podem descriure les compoments segons les equacions que queden de cada producte de coeficient* variable.

Podem calcular les compoments principals amb la funció prcomp (pricomp no em deixa perquè tenim més variables que mostres i no ho accepta):

```
# funció prcomp per calcular les components principals:
PCAS2 <- prcomp(matriu_t_scale)
names(PCAS2)

## [1] "sdev"     "rotation" "center"   "scale"    "x"

# podem mirar algunes dades com la desviació estandar:
PCAS2$sdev

##  [1] 7.924407e+03 3.468691e+03 2.393802e+03 1.787017e+03 1.193889e+03
##  [6] 8.828578e+02 7.615780e+02 6.161856e+02 4.211963e+02 3.665290e+02
## [11] 3.380935e+02 2.483672e+02 2.068163e+02 2.022902e+02 1.564062e+02
## [16] 1.391024e+02 1.291037e+02 1.192893e+02 9.903515e+01 7.372856e+01
```

```
## [21] 6.748460e+01 6.459395e+01 5.139770e+01 4.575442e+01 4.128852e+01
## [26] 4.047795e+01 3.371595e+01 2.749529e+01 2.572569e+01 2.191520e+01
## [31] 1.845033e+01 1.642524e+01 1.463849e+01 1.252113e+01 1.023357e+01
## [36] 7.334619e+00 5.668275e+00 4.614308e+00 3.447566e+00 7.867653e-13

#Mirem els scores que estan a x:
#head(PCAS2$x)
```

Amb rotation podem accedir a cada columna o vector i veure els coeficients per a cada variable, és a dir, el seu pes. Com més gran sigui el seu valor absolut major és el seu pes en explicar la variabilitat.

```
# Accedim a la primera component (primer vector o primera columna):
PCAS2$rotation[,1]
```

```
##            M4            M5            M7            M8           M11
## -0.0018491393 -0.0240736461 -0.0023802130 -0.0041838346 -0.0061719622
##           M14           M15           M25           M26           M31
## -0.0048952677 -0.0036494008 -0.0018053686 -0.0017481173 -0.0049281803
##           M32           M33           M36           M37           M45
## -0.0126200015 -0.0191225235 -0.0038111030 -0.0092267487 -0.3752120587
##           M48           M50           M51           M65           M66
## -0.9012062036 -0.0080282389 -0.0326116856 -0.0389602671 -0.1438895642
##           M68           M71           M73           M74           M75
## -0.0126579192 -0.0039409311 -0.0046887313 -0.0033804021 -0.0114864183
##           M88           M89           M90           M91           M93
## -0.0134131280 -0.0309296324 -0.0107933623 -0.0047345040 -0.0039738512
##          M101          M104          M105          M106          M107
## -0.0037321535 -0.0268114333 -0.0267764476 -0.0031026671 -0.0333448962
##          M110          M115          M116          M118          M119
##  0.0001454401 -0.0084341228 -0.0026611138 -0.0072209045 -0.0039984012
##          M120          M122          M126          M129          M130
## -0.0049200563 -0.0017672567 -0.0063434262 -0.1039481576 -0.0068894377
##          M134          M137          M138          M142          M144
## -0.0450986650 -0.0491638984 -0.0488879883 -0.0012338493 -0.0014941681
##          M148          M149
## -0.0127601573 -0.0088694743
```

```
coeficients = PCAS2$rotation[,1]

#ordenem decreixentment en valor absolut per veure els coeficients més al
ts i trobar-los
ordenats <- coeficients[order(abs(coeficients), decreasing = TRUE)]
cat("\n coeficients de major a menor", ordenats)
```

```
##
##  coeficients de major a menor -0.9012062 -0.3752121 -0.1438896 -0.1039
482 -0.0491639 -0.04888799 -0.04509867 -0.03896027 -0.0333449 -0.03261169
 -0.03092963 -0.02681143 -0.02677645 -0.02407365 -0.01912252 -0.01341313
-0.01276016 -0.01265792 -0.01262 -0.01148642 -0.01079336 -0.009226749 -0.
008869474 -0.008434123 -0.008028239 -0.007220905 -0.006889438 -0.00634342
```

```
6 -0.006171962 -0.00492818 -0.004920056 -0.004895268 -0.004734504 -0.0046
88731 -0.004183835 -0.003998401 -0.003973851 -0.003940931 -0.003811103 -0
.003732153 -0.003649401 -0.003380402 -0.003102667 -0.002661114 -0.0023802
13 -0.001849139 -0.001805369 -0.001767257 -0.001748117 -0.001494168 -0.00
1233849 0.0001454401
```

Podem veure com els metabòlits M48 (amb un coeficient de -0.9012062036), el M45 (amb -0.3752120587) i el M66 (0.1438895642) són els que més expliquen la primera component.

```
# Accedim a la segona component(:
PCAS2$rotation[,2]
```

```
##           M4           M5           M7           M8          M11
    M14
## -0.000874341  0.076802129  0.020823581  0.001502140  0.006885990  0.00
2229540
##          M15          M25          M26          M31          M32
    M33
##  0.006360350  0.003261146  0.012100332  0.004726398  0.022340329  0.03
3519065
##          M36          M37          M45          M48          M50
    M51
##  0.002101158  0.013660880 -0.418359274 -0.015294798  0.032142416  0.00
9824127
##          M65          M66          M68          M71          M73
    M74
##  0.058067222  0.840941034  0.001287133  0.003282470  0.005979298  0.00
3652064
##          M75          M88          M89          M90          M91
    M93
##  0.015740892  0.014734261  0.066736697  0.012068351  0.006376666  0.00
2463251
##         M101         M104         M105         M106         M107
   M110
##  0.003438738  0.043473683  0.013179306  0.004415253  0.022208744  0.00
7933248
##         M115         M116         M118         M119         M120
   M122
##  0.027255903  0.003523003  0.032132164  0.005348696  0.002961047  0.00
2889537
##         M126         M129         M130         M134         M137
   M138
##  0.004887510  0.216154290  0.032012675  0.148835017  0.018607561  0.15
0701858
##         M142         M144         M148         M149
##  0.004570863  0.004615770  0.043670540  0.007083789
```

```
coeficients2 = PCAS2$rotation[,2]

#ordenem decreixentment per veure els coeficients més alts i trobar-los
```

```r
ordenats2 <- coeficients2[order(abs(coeficients2), decreasing = TRUE)]
cat("\n coeficients de major a menor", ordenats2)
```

```
##
##  coeficients de major a menor 0.840941 -0.4183593 0.2161543 0.1507019
0.148835 0.07680213 0.0667367 0.05806722 0.04367054 0.04347368 0.03351906
 0.03214242 0.03213216 0.03201267 0.0272559 0.02234033 0.02220874 0.02082
358 0.01860756 0.01574089 -0.0152948 0.01473426 0.01366088 0.01317931 0.0
1210033 0.01206835 0.009824127 0.007933248 0.007083789 0.00688599 0.00637
6666 0.00636035 0.005979298 0.005348696 0.00488751 0.004726398 0.00461577
 0.004570863 0.004415253 0.003652064 0.003523003 0.003438738 0.00328247 0
.003261146 0.002961047 0.002889537 0.002463251 0.00222954 0.002101158 0.0
0150214 0.001287133 -0.000874341
```

En quant a la segona component la que més pes té és la M66 (amb un coeficient de 0.840941034), seguit de M45 (-0.418359274) o de M129 (0.216154290).

Discussió

L'anàlisi principal d'aquest estudi ha sigut un anàlisi de les components principals (PCA). L'avantatge d'aquest tipus d'anàlisi és que permet estudiar alhora conjunts de dades multivariants i permet reduir la dimensionalitat (un dels problemes de les dades òmiques) ja que redueix la complexitat en centrar-se un les components que millor expliquen la variabilitat de les dades.

En aquest cas s'han mirat les dues primeres components que juntes expliquen el 85.43% de la variabilitat de les dades de les mostres (71.96% i 13.74% respectivament).

A més, com es pot graficar, ens permet veure si hi ha agrupaments entre les diferents classes. En el nostre cas, treballàvem amb les classes GC (càncer gàstric) i HE (sans). Podem veure, en el gràfic, com hi ha algunes agrupacions similars en la mateixa regió, cosa que indica que alguns metabòlits no mostren diferències entre els dos grups (sans i amb càncer). No obstant, també podem veure punts de colors diferents (grups diferents), completament separats, el que indica que alguns metabòlits sí que tenen un impacte gran en la variabilitat dels dos grups de mostres de l'estudi.

Això es pot veure posteriorment amb les dades dels coeficients que s'obtenen del rotation de la funció prcomp de l'anàlisis PCA. Podem veure com per a la primera component el metabòlit M48 (Creatinina), el M45 (citrat) i en menor grau el M66 (Hippurat) poden jugar un paper clau en les diferències observats entre els grups dels pacients de càncer i el sans. En la segona component, menys explicativa, hi torna a aparèixer entre els valors més alts el M45 amb signe negatiu igualment però el més alt és el M66 (amb signe positiu). A part, apareix d'entre els més alts el M129 (u11).

No obstant, les diferències entre les dues components pot indicar que els patrons que hi ha darrera d'aquestes dades són complexos. No obstant, tenim diversos

metabòlits que poden ser d'interès de cara a explicar les diferències com la Creatinina, el citrat i el Hippurat.

Caldria fer més anàlisis partint d'aquests metabòlits o dels que tenen majors impactes en la variabilitat per estudiar millor les diferències entre el grup de càncer gàstric i el grup d'individus sans.

Es podrien fer a més proves estadístiques com ANOVa o t-test per valorar les diferències significatives entre els grups diferents.I acompanyar les dades inicials amb gràfics que ajudin a visualitzar millor les distribució de les dades (que m'ha faltat per temps).

Per altra banda, cal comentar un aspecte important i és que donat al fet que hi havia molts valors nuls, durant el processat de les dades s'han eliminat nombroses mostres quedant-nos amb 40. Això té un efecte molt important sobre els anàlisis i s'ha de tenir en consideració ja que s'ha perdut molta informació i pot alterar les valoracions finals. Una bona pràctica, en comptes d'eliminar els valors que faltaven, hagués sigut imputar-los, tal com s'aconsella al propi web del dataset *https://cimcb.github.io/MetabWorkflowTutorial/Tutorial1.html*).

Enllaç per accedir al repositori Github

https://github.com/davidfernandez9390/Fernandez-Lopez-David-PEC1.git