

PAC 2 Joan Martinez Sancho

18/05/25

Contents

1. Introducció	2
2. Materials i Mètodes	2
2.1. Dades	2
2.2. Processament de les dades	2
2.3. Filtratge individus	2
2.4. Filtratge de gens i normalització	2
2.5. Anàlisis estadístic	2
3. Resultats	3
3.1. Distribució dels valors expressats	3
3.2. Similitud entre mostres	4
3.3. Agrupament jeràrquic	5
3.4. Escala multidimensional	5
4. Anàlisis d'expressió diferencial	6
4.1. Descriptiva de les dades	6
4.2. Anàlisis diferencial	7
4.3 Comparacions múltiples i visualització dels resultats	9
5. Anàlisis de significació biològica	10
5.1 Visualització dels resultats de l'anàlisis d'enriquiment	10
6. Discussió	11
7. Conclusions	11
8. Referències	12
9. Material Suplementari	12
10. Annexe	13

1. Introducció

Les infeccions respiratòries constitueixen una de les principals causes de morbiditat i mortalitat a escala mundial. La capacitat de distingir entre etiologies víriques i bacterianes en pacients simptomàtics és essencial per a una gestió clínica eficient, la reducció de l'ús innecessari d'antibiòtics i el control de brots. En aquest context, la transcriptòmica permet caracteritzar les respuestes immunitàries específiques davant diferents patògens, oferint una via prometedora per al desenvolupament de biomarcadors diagnòstics.

Aquest estudi té com a objectiu explorar els patrons d'expressió gènica en mostres de sang perifèrica de pacients amb COVID-19, infeccions bacterianes i individus sans. Mitjançant l'anàlisi d'un subconjunt de dades RNA-Seq de la cohort GSE161731, s'apliquen tècniques de preprocessament, normalització i comparació per identificar signatures transcriptòmiques diferencials entre els grups.

2. Materials i Mètodes

2.1. Dades

S'utilitzen dades públiques del GEO (GSE161731), que inclouen una matriu de comptatges de gens i metadades clíniques associades a 198 mostres. Les metadades contenen informació sobre l'edat, el gènere, la raça, la cohort clínica, així com altres variables com l'hospitalització i el lot experimental.

2.2. Processament de les dades

Es realitza una harmonització dels noms de les columnes del fitxer de comptatges per poder fusionar les dades transcriptòmiques i les clíniques. Per fer-ho, se substitueixen els caràcters estranys de les dues bases de dades per “_”. Es defineix la tipologia correcta de les variables clíniques. Abans de la creació de l'objecte **SummarizedExperiment**, es construeix una base de dades amb els gens disponibles i les seves anotacions. Amb els tres data frames (comptatges, metadades i anotacions), es crea l'objecte **SummarizedExperiment**.

2.3. Filtratge individus

Es filtra l'objecte SummarizedExperiment, seleccionant només les mostres de les cohorts COVID-19, Bacterial o sans. S'eliminen les mostres repetides (mateix subject_id), conservant únicament la primera entrada. Finalment, es selecciona una mostra de 75 individus.

2.4. Filtratge de gens i normalització

Es transformen els comptatges a valors CPM (Counts per Million) utilitzant la llibreria edgeR. Es conserven únicament els gens amb CPM > 1 en almenys el 20% de les mostres (> 14). També s'eliminen les tres mostres amb les lectures d'expressió més baixes.

Després del filtratge, s'aplica una normalització TMM (Trimmed Mean of M-values) i una transformació log2. Es crea un objecte DGE i les dades normalitzades i transformades es guarden de nou a l'objecte **SummarizedExperiment**.

2.5. Anàlisis estadístic

Es comparen les variables clíniques entre cohorts per detectar possibles variables confusores. Les variables numèriques es resumeixen amb la mediana i el rang interquartílic. Les categòriques s'expressen en freqüències i proporcions. S'aplica la prova U de Mann-Whitney per comparar l'edat, i la prova de la Xi-Quadrat o de Fisher per a les categòriques.

En segon lloc, per avaluar la qualitat global i la coherència dels valors d'expressió entre mostres, es porta a terme una anàlisi exploratòria utilitzant valors de comptatge normalitzats i transformats en escala logarítmica. La distribució dels comptatges normalitzats es visualitza mitjançant boxplots de valors log2-CPM, tant abans com després de la transformació. Això permet identificar mostres amb comportaments atípics i avaluar la consistència general entre condicions experimentals. Per avaluar la similitud global entre mostres, es calcula una matriu de distàncies euclidianes entre els perfils d'expressió utilitzant la funció dist() de base R sobre els

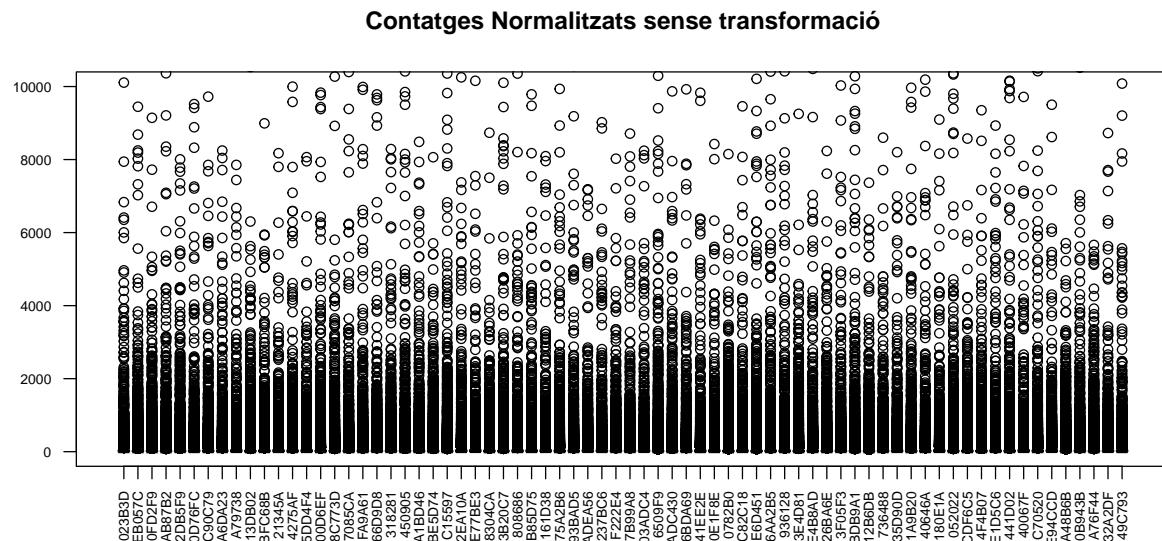
valors log2CPM. A partir de la matriu de distàncies, es realitza un agrupament jeràrquic, visualitzant-lo amb un dendrograma, representant la proximitat transcripcional entre mostres. Finalment s'aplica un escalament multidimensional clàssic (MDS) sobre la matriu de distàncies, mitjançant la funció cmdscale(), per projectar les mostres en un espai de dues dimensions.

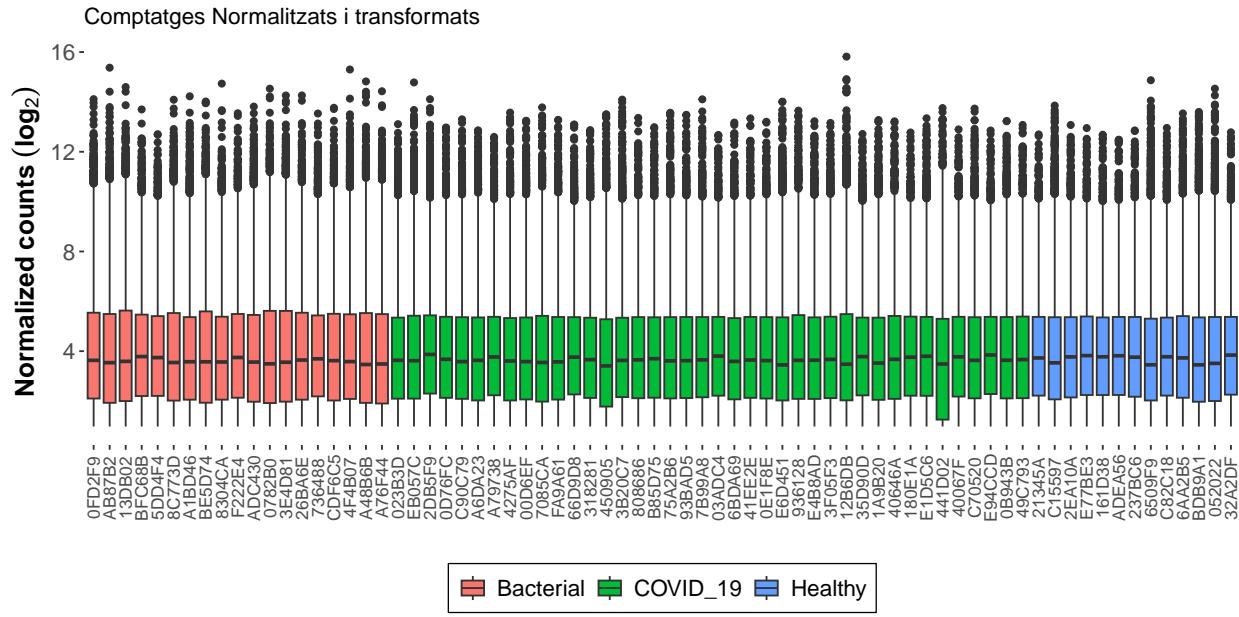
Finalment, s'implementen models lineals generalitzats utilitzant el paquet *limma* d'R amb la intenció d'obtenir els gens diferencialment expressats entre dues cohorts (Bacterial vs Healthy i COVID19 vs Healthy). Com a criteri d'expressió diferencial s'ha utilitzat un llindar de log2FC de 1,5. I per corregir els p valors obtinguts s'ha emprat la correcció de Benjamini Hochberg.

Per a totes les anàlisis estadístiques s'ha considerat un nivell de significació del 0,05.

3.Resultats

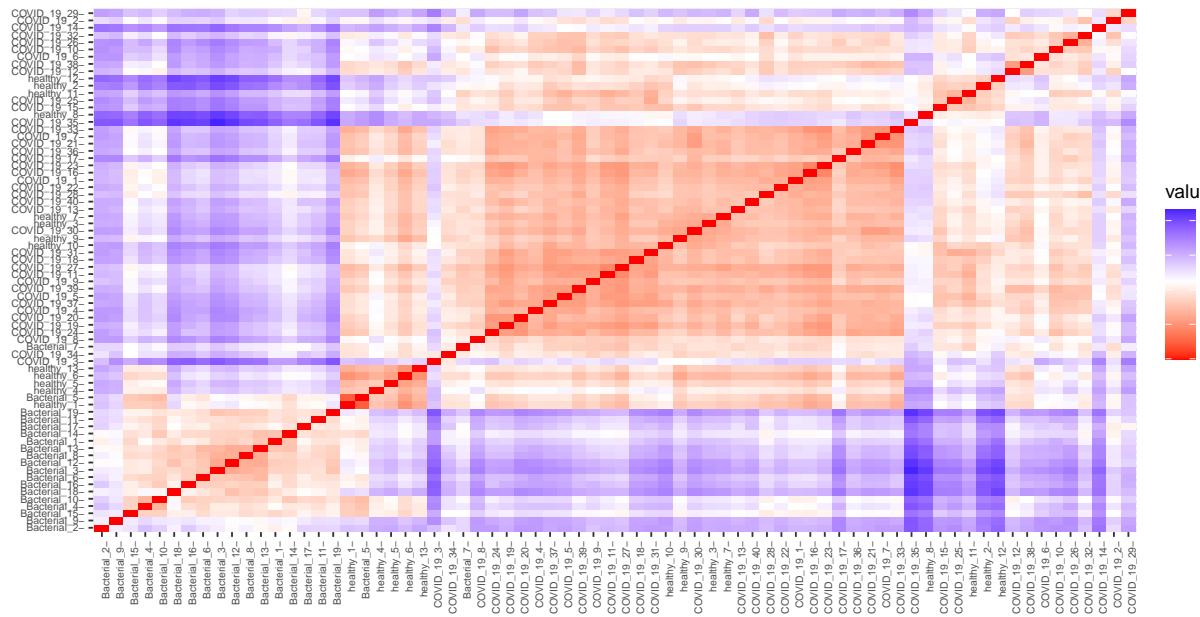
3.1. Distribució dels valors expressats





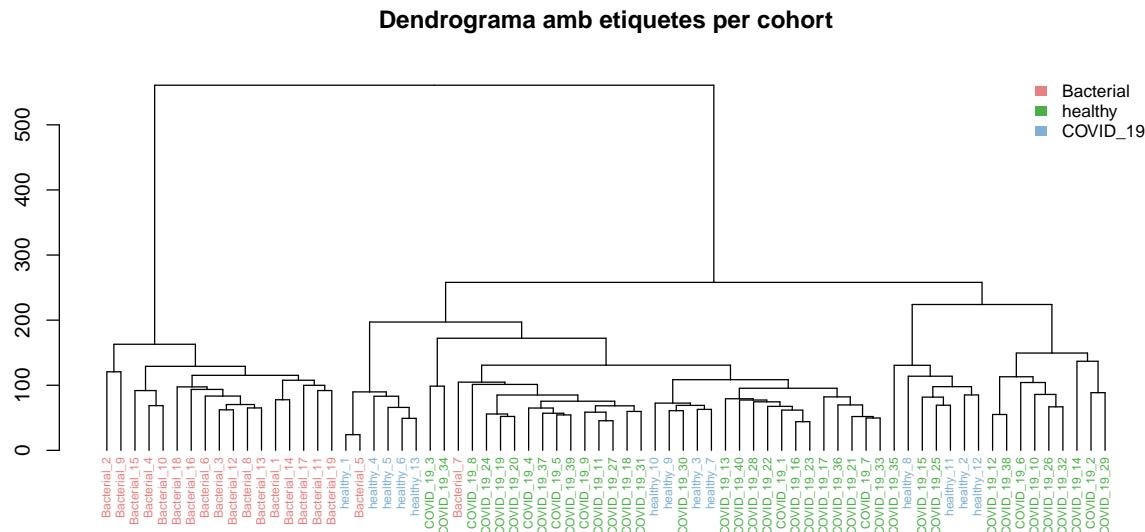
Els valors de log2(CPM) mostren distribucions simètriques dins de cada grup. Hi ha certa heterogeneïtat a les mostres COVID-19 però sense valors extrems rellevants. Les mostres Healthy presenten menor dispersió, coherent amb un estat basal no inflamatori.

3.2. Similitud entre mostres



Es calcula la matriu de distàncies euclidianes i s'observa una agrupació per cohort. Algunes mostres (p. ex., Bacterial_7) es desvien de la resta. També hi ha entremescla entre individus Healthy i COVID-19.

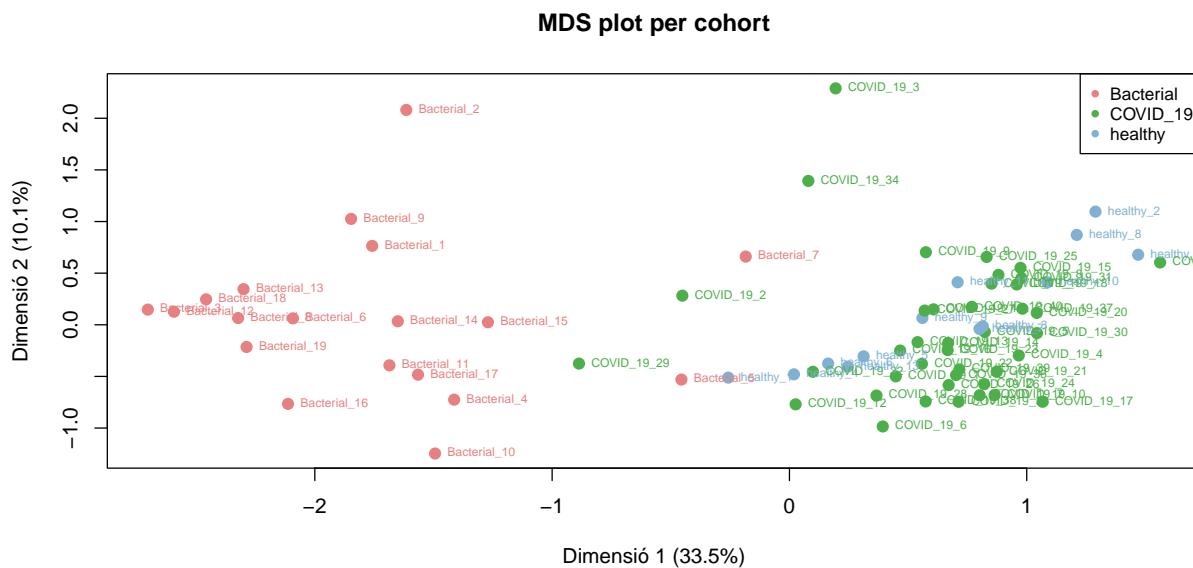
3.3. Agrupament jeràrquic



El dendrograma mostra agrupaments similars als de la matriu de distàncies. Les cohorts tendeixen a agrupar-se amb algunes excepcions.

3.4. Escala multidimensional

Realitzem un escalament multidimensional el qual ens proporcionarà una representació en dimensió reduïda que descriu amb relativa fidelitat les diferències i similituds entre mostres.



Les mostres del grup Healthy apareixen àmpliament distribuïdes entre les de COVID-19, fet que suggereix una baixa separabilitat transcriptòmica entre aquests dos grups en l'espai de components principals. Això podria indicar una similitud en les signatures d'expressió basals i/o una resposta immune menys específica en

els individus sans.

D'altra banda, s'observen diversos individus amb un comportament clarament atípic dins del seu grup. Concretament, les mostres COVID19_3, COVID19_29 i COVID19_2 presenten posicions allunyades respecte del nucli principal del grup COVID-19. De manera similar, les mostres Bacterial_2, Bacterial_5 i Bacterial_7 també es desvien notablement respecte del patró general del seu grup.

Tot i que la meva inclinació és mantenir aquestes mostres, ja que poden reflectir variabilitat biològica real i no necessàriament error experimental, un dels objectius específics d'aquesta pràctica és identificar i eliminar possibles outliers. Per tant, i de manera justificada, es decideix excluir les sis mostres esmentades de les ànàlisis posteriors per tal de millorar la cohesió dels grups i la interpretabilitat dels resultats.

4. Anàlisis d'expressió diferencial

4.1. Descriptiva de les dades

Table 1: Comparació Bivariada

	Bacterial N=16	COVID_19 N=37	healthy N=13	p.overall
age	64.5 [48.0;76.8]	33.0 [29.0;56.0]	18.0 [18.0;19.0]	<0.001
gender:				0.962
Female	8 (50.0%)	17 (45.9%)	6 (46.2%)	
Male	8 (50.0%)	20 (54.1%)	7 (53.8%)	
race:				<0.001
Asian	0 (0.00%)	6 (16.2%)	6 (46.2%)	
Black_African_American	12 (75.0%)	5 (13.5%)	0 (0.00%)	
Native_Hawaiian_Pacific_Islander	0 (0.00%)	1 (2.70%)	0 (0.00%)	
Other_More_than_one_race	0 (0.00%)	0 (0.00%)	1 (7.69%)	
Unknown_Not_reported	2 (12.5%)	0 (0.00%)	0 (0.00%)	
White	2 (12.5%)	25 (67.6%)	6 (46.2%)	
time_since_onset:				.
early	0 (.)	12 (32.4%)	0 (.)	.
late	0 (.)	5 (13.5%)	0 (.)	.
middle	0 (.)	20 (54.1%)	0 (.)	.
hospitalized:				.
No	0 (.)	30 (81.1%)	0 (.)	.
Yes	0 (.)	7 (18.9%)	0 (.)	.

L'ànàlisi bivariat de les variables clíniques revela diferències notables en l'edat entre els tres grups estudiats. Sense realitzar encara comparacions estadístiques detallades, s'observa que els individus del grup Healthy tendeixen a ser més joves, mentre que els pacients amb COVID-19 corresponen principalment a edats joves-adultes, i els de la cohort Bacterial presenten edats mitjanes més altes. Aquest patró suggereix que l'edat pot actuar com a variable confusora en les comparacions transcriptòmiques, especialment en aquelles associades a la resposta immunitària.

Pel que fa al sexe, la seva distribució és equilibrada entre els tres grups, cosa que a priori no fa necessària la seva inclusió com a covariable ajustadora. Tot i això, tenint en compte la importància creixent d'una perspectiva de gènere en la recerca biomèdica i el reconeixement de possibles diferències en la resposta immune entre sexes, es decideix mantenir el sexe com a variable de control en les ànàlisis posteriors.

Pel que fa a la variable raça, existeix un nombre molt reduït d'observacions en algunes de les seves categories. Aquesta manca d'equilibri pot comprometre la consistència estadística dels models i augmentar la probabilitat

de resultats espuris. Per aquest motiu, es decideix no incloure la raça com a covariable en les anàlisis principals.

Finalment, les variables time_since_onset i hospitalized només estan disponibles per als individus de la cohort COVID-19. Per tant, aquestes variables només resulten útils en cas de dur a terme subanàlisis específics dins d'aquesta cohort, però no s'inclouen en les comparacions entre grups principals.

4.2. Anàlisis diferencial

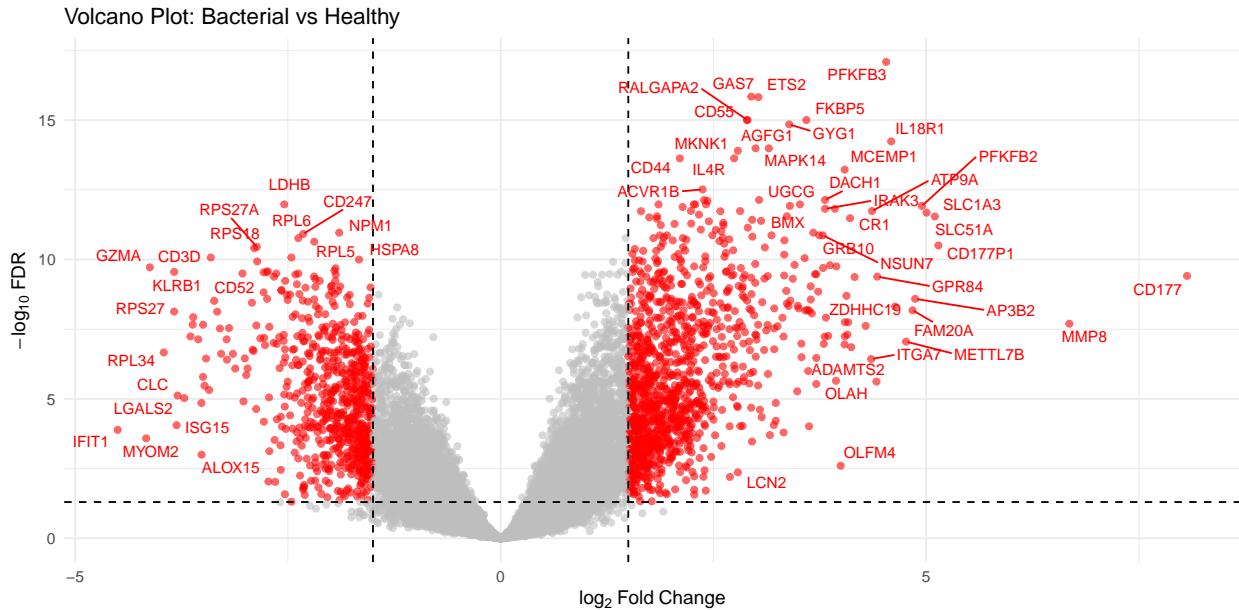
```
##   grouphealthy groupBacterial groupCOVID_19 age genderMale
## 1          0          0          1   50          0
## 2          0          1          0   52          0
## 3          0          0          1   30          1
## 4          0          0          1   32          1
## 5          0          0          1   33          1
## 6          0          0          1   35          0

##               Contrasts
## Levels           BacterialVsHealthy COVID19VsHealthy
## grouphealthy      -1              -1
## groupBacterial     1              0
## groupCOVID_19      0              1
## age                0              0
## genderMale         0              0

## [1] "voom+limma"
```

Table 2: Resum dels gens diferencialment expressats en la comparació Bacterial vs Healthy

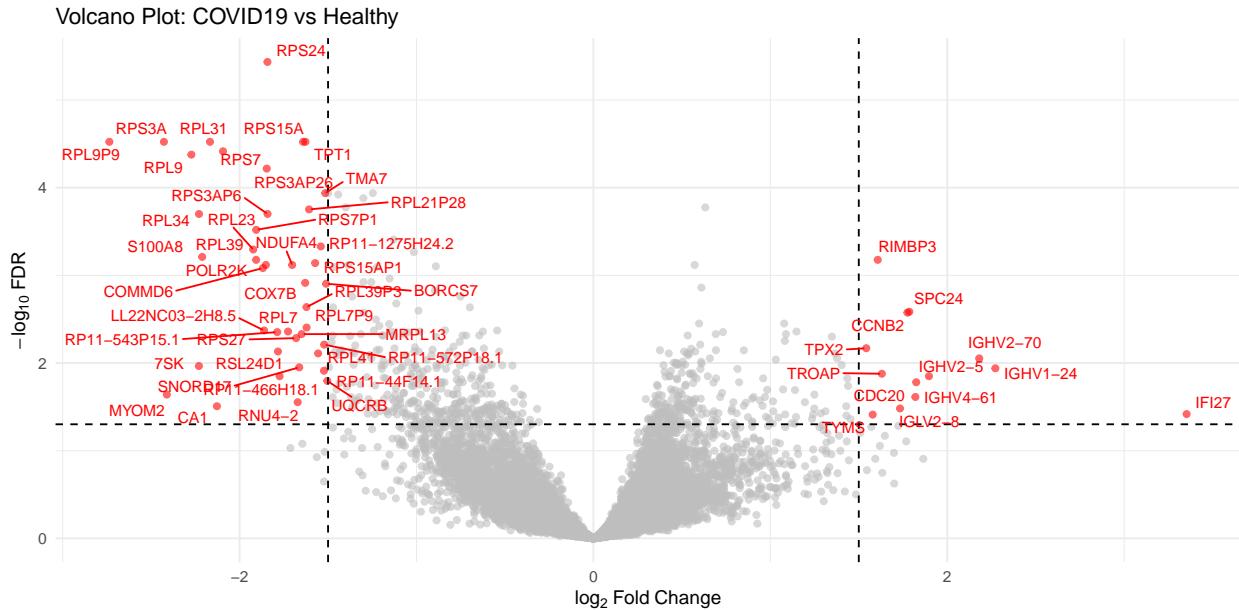
Estat.dels.gens	Nombre.de.gens
Gens diferencialment expressats ($\log FC > 1.5 \text{ & } FDR < 0.05$)	1796
- Sobreeexpressats en Bacterial ($\log FC > 1.5$)	1132
- Infraexpressats en Bacterial ($\log FC < -1.5$)	664
Gens no diferencialment expressats	13815
Total de gens analitzats	15611



De 15611 gens analitzats el 11.50% (n=1796) són diferencialment expressats entre els individus de la cohort Bacterial i de la cohort Healthy. D'aquests el 63.03% (n=1132) estan sobreexpressats i la resta, un 36.97% (n=664), infraexpressats.

Table 3: Resum dels gens diferencialment expressats en la comparació COVID19 vs Healthy

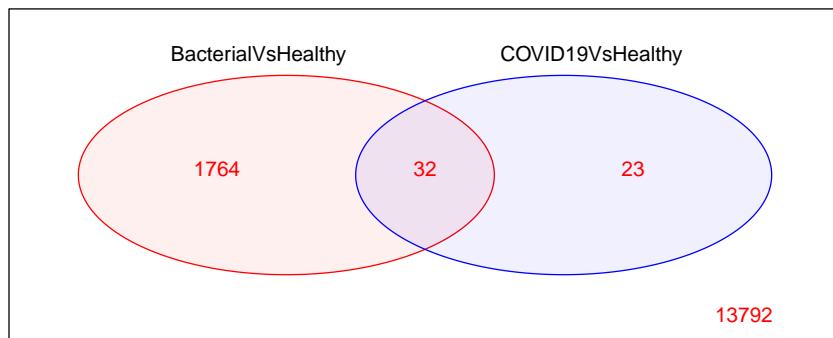
Estat.dels.gens	Nombre.de.gens
Gens diferencialment expressats	55
- Sobreexpressats en COVID19 (logFC > 1.5)	13
- Infraexpressats en COVID19 (logFC < -1.5)	42
Gens no diferencialment expressats	15556
Total de gens analitzats	15611



De 15611 gens analitzats el 0.35% ($n=55$) són diferencialment expressats entre els individus de la cohort COVID19 i de la cohort Healthy. D'aquests el 23.64% ($n=13$) estan sobreexpressats i la resta, un 76.36% ($n=42$), infraexpressats.

4.3 Comparacions múltiples i visualització dels resultats

Comparem les dues comparacions realitzades. Com a criteri de significació estadística hem considerat un pvalor ajustat inferior a 0.05 i el log2FC de 1.5 per una expressió diferencial.

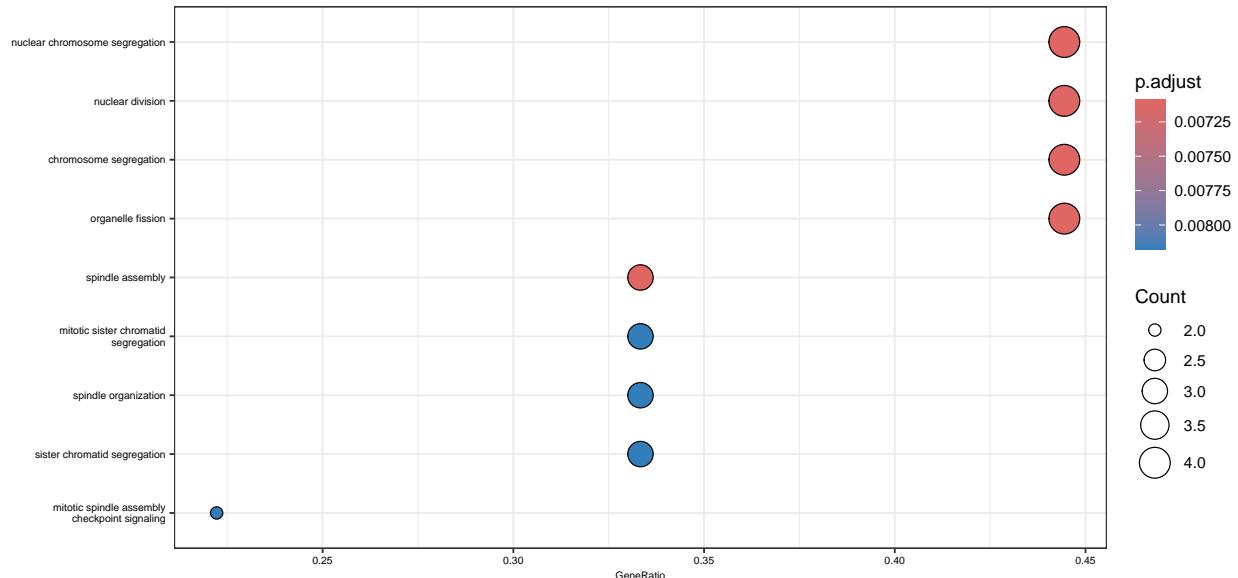


Observem que el 58.18% ($n=32$) de gens diferencialment expressats entre COVID19 i Healthy també estàn diferencialment expressats entre Bacterial i Healthy.

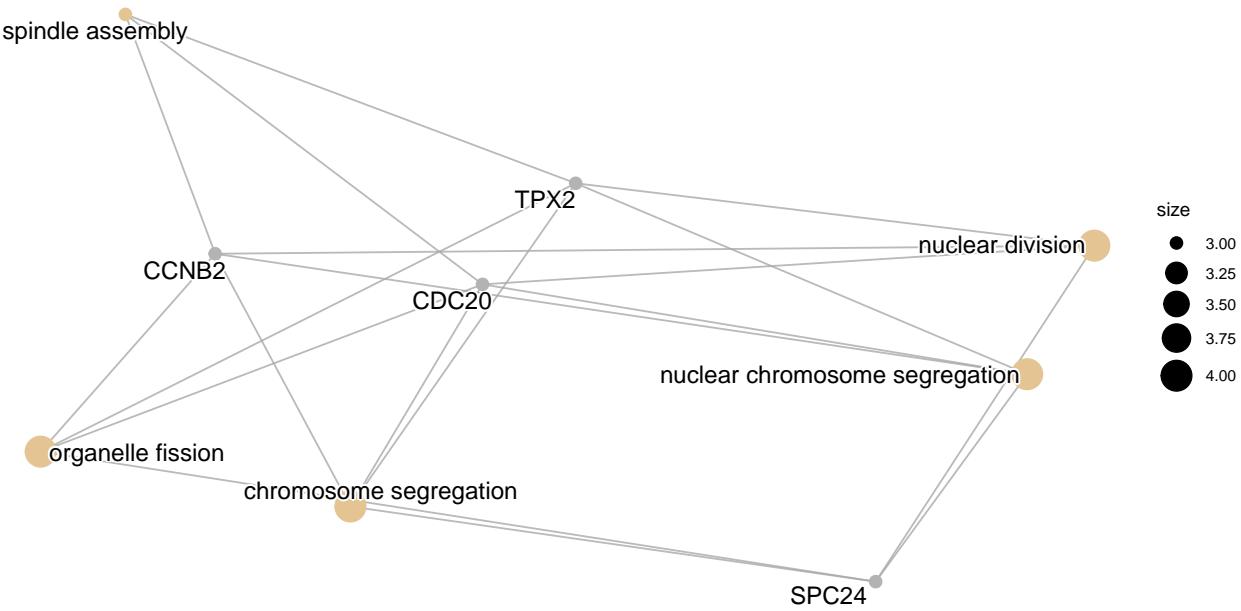
5. Anàlisis de significació biològica

5.1 Visualització dels resultats de l'anàlisis d'enriquiment

```
##           ID          Description GeneRatio   BgRatio
## GO:0098813 GO:0098813 nuclear chromosome segregation 4/9 261/11312
## GO:0000280 GO:0000280      nuclear division    4/9 327/11312
## GO:0051225 GO:0051225      spindle assembly   3/9 119/11312
## GO:0007059 GO:0007059      chromosome segregation 4/9 361/11312
## GO:0048285 GO:0048285      organelle fission    4/9 366/11312
## GO:0000070 GO:0000070 mitotic sister chromatid segregation 3/9 172/11312
##          pvalue     p.adjust      qvalue      geneID Count
## GO:0098813 3.185579e-05 0.007092393 0.003795681 SPC24/CCNB2/TPX2/CDC20 4
## GO:0000280 7.700748e-05 0.007092393 0.003795681 SPC24/CCNB2/TPX2/CDC20 4
## GO:0051225 9.105212e-05 0.007092393 0.003795681 CCNB2/TPX2/CDC20 3
## GO:0007059 1.131869e-04 0.007092393 0.003795681 SPC24/CCNB2/TPX2/CDC20 4
## GO:0048285 1.194006e-04 0.007092393 0.003795681 SPC24/CCNB2/TPX2/CDC20 4
## GO:0000070 2.712880e-04 0.008178168 0.004376762 SPC24/TPX2/CDC20 3
```



L'anàlisis de significació dels gens sobreexpressats ens indica que estan fortament relacionats amb el control del cicle cel·lular, específicament amb **mecanismes mitòtics**. Per tant, la condició de tenir COVID19 podria estar afectant en la regulació del cicle cel·lular.



L'anàlisis de xarxa funcional mostra que els gens sobreexpressats de la cohort COVID19 estan fortament associats a processos de divisió nuclear i segregació cromosòmica. Gens com TPX2, CCNB2, CDC20 i SPC24, es troben al centre d'aquestes vies, suggerint que la COVID19 afecta directament la regulació del cicle cel·lular.

6. Discussió

Aquest estudi explora l'expressió gènica en sang perifèrica de pacients amb infeccions respiratòries bacterianes, pacients amb COVID-19 i individus sans, amb l'objectiu de distingir signatures transcriptòmiques diferencials que puguin ajudar a la discriminació clínica entre etiologies víriques i bacterianes. Aquesta diferenciació és clau per reduir l'ús innecessari d'antibiòtics i per millorar la presa de decisions diagnòstiques i terapèutiques.

Tot i això, l'estudi presenta diverses limitacions importants. En primer lloc, la grandària de la mostra final (66 individus després del filtratge) pot limitar el poder estadístic, especialment en la detecció de canvis subtils d'expressió entre grups. Aquesta limitació és especialment rellevant per al grup COVID-19, en què la variabilitat interna observada pot haver diluït senyals diferencials, afavorint una aparent similitud amb el grup sa.

També cal destacar que les dades transcriptòmiques provenen exclusivament de sang perifèrica, fet que pot limitar la capacitat de capturar respostes immunes específiques que es manifesten a nivell local, com ara en el teixit pulmonar. Així, no es pot descartar que la resposta immunitària més marcada en COVID-19 es produueixi en altres compartiments biològics no evaluats en aquest estudi.

Finalment, cal fer autocrítica sobre el propi procés analític. L'ús de llindars de filtratge (CPM, % de mostres) i el valor de $\log_{2}FC \geq 1.5$ són decisions que poden afectar fortament el conjunt de gens identificats com diferencials. La sensibilitat dels resultats a aquests paràmetres posa de manifest la importància de validar les troballes amb altres mètodes o cohorts independents.

7. Conclusions

Els resultats de l'anàlisi diferencial d'expressió són consistents amb les observacions prèvies derivades de les tècniques d'escalament multidimensional i agrupament jeràrquic, on ja es va evidenciar una separació clara del grup Bacterial respecte als altres dos. En concret, un 11.50% dels gens, fet que indica una activació transcripcional significativa en els pacients amb infecció bacteriana.

En canvi, la comparació entre COVID-19 i Healthy mostra un perfil molt més similar: només un 0.35% dels

gens compleixen els criteris de significació. Aquest escàs nombre de gens diferencials reforça la forta solapació observada entre mostres COVID-19 i Healthy en els gràfics de similitud, suggerint que, almenys a nivell transcriptòmic basal, la resposta immunitària en COVID-19 pot ser més propera a l'estat sa que no pas a la resposta inflamatòria característica de les infeccions bacterianes.

Mentre que la resposta transcriptòmica a la infecció bacteriana és molt marcada i diferenciada, la resposta en pacients amb COVID-19 presenta un perfil global molt més proper al dels individus sans. No obstant això, l'anàlisi de significació funcional dels pocs gens diferencialment sobreexpressats en COVID-19 revela que aquests estan fortament associats a processos de regulació del cicle cel·lular, especialment a mecanismes mitòtics i de divisió nuclear.

Així, la integració dels resultats transcriptòmics amb l'anàlisi funcional reforça la idea que la COVID-19, tot i generar una resposta globalment moderada a nivell d'expressió gènica, pot afectar de manera específica processos biològics essencials per al funcionament i la viabilitat cel·lular.

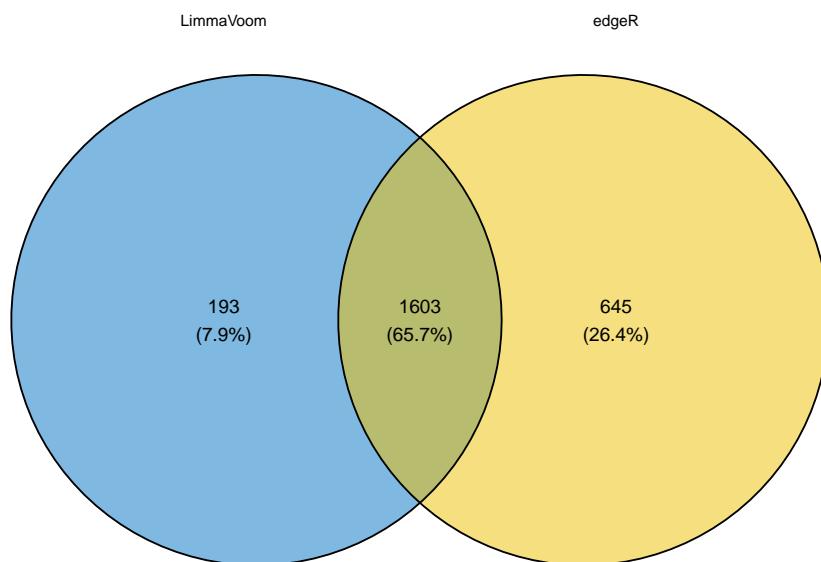
8. Referències

1. https://aspteaching.github.io/Analisis_de_datos_omicos-Ejemplo_2-RNASeq/WorkflowBasico_de_RNASeq.html#5_Preprocesado_de_los_datos
2. McClain, M. T., Constantine, F. J., Henao, R., Liu, Y., Tsalik, E. L., Burke, T. W., Steinbrink, J. M., Petzold, E., Nicholson, B. P., Rolfe, R., Kraft, B. D., Kelly, M. S., Saban, D. R., Yu, C., Shen, X., Ko, E. M., Sempowski, G. D., Denny, T. N., Ginsburg, G. S., & Woods, C. W. (2021). Dysregulated transcriptional responses to SARS-CoV-2 in the periphery. *Nature Communications*, 12(1), 1079. <https://doi.org/10.1038/s41467-021-21289-y>
3. https://github.com/ASPTeaching/Analisis_de_RNASeq_para_Infografia/tree/main
4. <https://github.com/jmsancho18/Martinez-Sancho-Joan-PEC2>

9. Material Suplementari

Aquí tenim la comparació entre el mètode Edge i Limma+Voom

Comparació Gens Diferencialment Expressats Voom i Edge Bacterial



10. Annexe

```
##### Processament de dades
library(dplyr)
library(tidyverse)
#BiocManager::install("EnsDb.Hsapiens.v86")
library(EnsDb.Hsapiens.v86)
## Lectura de la matriu de counts.
counts <- read.csv("GSE161731_counts.csv", row.names = 1)
## Lectura de la matriu de metadades.
metadades <- read.csv("GSE161731_counts_Key.csv")

## S'han de canviar una mica els noms de les columnes de "counts" per a poder "emparellar les dades
colnames(counts) <- gsub("\\.", "_", colnames(counts))
colnames(counts) <- gsub("X", "", colnames(counts))

#De les metadades canviem els signes estranys per a "_"
metadades <- metadades %>%
  mutate(across(where(is.character), ~ str_replace_all(., "[\\s\\-/]", "_")))

## Si fem la intersecció entre les columnes de "counts" i rna_id
## de les metadades
## veiem que ens quadren els 198 individus que hi tocaria haver
## Ens quedem només amb les metadades i counts de les mostres en comú
## entre la matriu de expressió i les metadades

counts_metadades <- intersect(colnames(counts), metadades$rna_id)
metadades <- metadades[metadades$rna_id %in% counts_metadades,]
counts <- counts[,counts_metadades]

str(metadades)
#Abans de crear el SummarizedExperiment el que farem serà definir bé les covariables
## la variable edat té un >89, que s'haurà de considerar NA, ja que no tenim
## l'edat i no la podem
## tractar com a numèrica si posem >89.
## El Gender, Race, cohort, time_since_onset i hospitalized seran factors
metadades <- metadades %>%
  mutate(
    age = as.numeric(age), # edat com a numèrica
    gender = as.factor(gender), # sexe com a factor
    cohort = as.factor(cohort),
    race = as.factor(race),
    time_since_onset = as.factor(time_since_onset),
    hospitalized = as.factor(hospitalized),
    batch = as.factor(batch)
  )
# Converteix metadata en DataFrame per a SummarizedExperiment
colData <- as.data.frame(metadades)
rownames(colData) <- colData$rna_id

## Obtenim la base de dades Anotacions dels gens
gens <- genes(EnsDb.Hsapiens.v86, return.type = "GRanges")
```

```

# Alinea gens (files) entre matriu i anotació
## Mirem quins gens de la matriu "counts" trobem anotats en la base de dades
## de Ensembl i ens quedem només amb els que tinguin anotació (coordenades)

matriu_gens <- intersect(rownames(counts), names(gens))
counts <- counts[matriu_gens, ]
gens <- gens[matriu_gens]

## Creem l'objecte SummarizedExperiment
library(SummarizedExperiment)
se <- SummarizedExperiment(
  assays = list(counts = as.matrix(counts)),
  rowRanges = gens,
  colData = colData
)

## Ara volem treballar només amb les cohorts COVID19, BActerial, i Healthy
coldata <- as.data.frame(colData(se))
coldata <- coldata %>%
  dplyr::filter(cohort %in% c("COVID_19", "Bacterial", "healthy"))

## Volem eliminar els individus repetits
coldata <- coldata %>%
  distinct(subject_id, .keep_all = TRUE)

## Treballem només amb el se que no té individus repetits i amb els filters de cohort utilitzats
## De fet podríem ficar com a colnames de se el id del subjecte.
se <- se[, colnames(se) %in% coldata$rna_id]
nom_sub <- coldata$subject_id[match(colnames(se), coldata$rna_id)]
colnames(se) <- nom_sub

## Segons l'enunciat, seleccionem una mostra de 75 individus de manera aleatoria
myseed <- sum(utf8ToInt("joanmartinezsanche"))
set.seed(myseed)
mostres_random <- sample(colnames(se), 75)
se_75 <- se[,mostres_random]

library(edgeR)
## Expresssem els contatges com CPMs es a dir, counts per milió.
## Podem guardar els contatges "bruts" per si mai es volen recuperar
counts.CPM <- cpm(assay(se_75))

assays(se_75)$countCPM <- counts.CPM

## Un cop tenim les dades com CPMs podem filtrar. Com a mètode de filtratge utilitzarem el filtre que
## realitza l'article de referència que hem escollit per obtenir les dades
## Genes with counts per million greater than 1 in fewer than 20% of samples

```

```

## were dropped along with three samples with a high proportion
## of lowly expressed reads.
## És a dir, s'eliminem els gens amb recomptes per milló superior
## a 1 en menys del 20% de les mostres.
## Per tant, ens quedem amb aquells gens que tenen recomptes per milió superior
## a 1 en almenys el 20% de mostres. Per tant, en el nostre cas ens quedariem
## amb aquells gens que tenen un recompte per milió superior a 1
## en almenys 15 (0.2*75mostres) mostres

gens_keep <- rowSums(counts.CPM>1) >= (0.2*ncol(se_75))

# Eliminar 3 mostres amb més baixa expressió, els quals són les mostres amb id 896282, 5835FC i 9C7138
low_expr_samples <- order(colSums(counts.CPM>1))[1:3]
#colSums(counts.CPM>1)[c(20,43,54)]

## Tenim el nou SummarizedExperiment amb les dades filtrades
se_75_filtrat <- se_75[gens_keep,-low_expr_samples]

## Normalització
## Per fer-ho primer guardarem les dades de recomptes a un objecte dgeList
dgeObj <- DGEList(counts = assays(se_75_filtrat)$countCPM,
                   samples = colData(se_75_filtrat),
                   genes = rowData(se_75_filtrat),group = se_75_filtrat$cohort,
                   lib.size = colSums(assays(se_75_filtrat)$countCPM),
                   remove.zeros = FALSE,
                   norm.factors = rep(1,ncol(assays(se_75_filtrat)$countCPM)))

#show(dgeObj)

##Además de estandarizar los contajes, es importante eliminar otros sesgos de
##composición entre librerías. Esto puede hacerse aplicando la normalización
##por el método TMM que genera un conjunto de factores de normalización,
##tal que producto de estos factores y los tamaños de librería (el número
##de secuencias de cada muestra) definen el tamaño efectivo de dichas
##muestras, es decir el peso real que se les asignará en las
##comparaciones posteriores.
## Així com indica l'apartat de mètodes de l'article de referència
## realitzarem una normalització pel mètode TMM i llavors aplicarem
## el logaritme en base 2

dgeObj_norm <- calcNormFactors(dgeObj, method = "TMM")
head(dgeObj_norm$samples, 10)

## Aquestes transformacions cerquen compensar la mida diferent de
## les llibreries o la diferent composició
## d'aquestes, pero les distribucions en cada mostra són asimètriques

boxplot(dgeObj_norm$counts, col = dgeObj_norm$samples$group,
         las = 2, cex.axis = 0.7,
         main = "Contajes normalizados",ylim = c(0, 10000))

log2count_norm <- cpm(dgeObj_norm, log = TRUE)

```

```

orden <- order(dgeObj_norm$samples$group)
log2count_ordenat <- log2count_norm[, orden]
cols_ordenats <- dgeObj_norm$samples$group[orden]
grups_ordenats <- dgeObj_norm$samples$group[orden]
##grups_ordenats <- dgeObj_norm$samples$subject_id[orden]
boxplot(log2count_ordenat,
        col = cols_ordenats,
        las = 2,
        cex.axis = 0.7,
        main = "Contajes normalizados (log2)",
        names = grups_ordenats)

assays(se_75_filtrat)$log2count_nom <- log2count_norm

ordered_sample_names <- rep(colnames(log2count_ordenat), each = nrow(log2count_ordenat))
group_order <- rep(c(rep("Bacterial", 19), rep("COVID_19", 40), rep("Healthy", 13)),
                    each = nrow(log2count_norm))

d <- tibble(
  y = as.numeric(log2count_ordenat),
  x = factor(rep(colnames(log2count_ordenat), each = nrow(log2count_ordenat))),
  co = factor(c(rep("Bacterial", each = nrow(log2count_norm) * 19),
               rep("COVID_19", each = nrow(log2count_norm) * 40),
               rep("Healthy", each = nrow(log2count_norm) * 13)))
)

# Reordena x segons co
d$x <- factor(d$x, levels = unique(d$x[order(d$co)]))

gg1 <- ggplot(d,aes(x,y,fill=co)) + geom_boxplot() +
  xlab('') + ylab(expression(bold('Normalized'~'counts'~(log[2])))) +
  theme(axis.text.x = element_text(angle=90,vjust = 0.5, size=10),
        axis.text.y = element_text(size=13),
        axis.title.y = element_text(size=15),
        axis.title = element_text(face='bold'),
        legend.title = element_blank(),
        legend.text = element_text(size=13),
        legend.position = 'bottom',
        plot.background = element_rect(fill = "transparent", color = NA),
        panel.background = element_rect(fill = "transparent"),
        legend.background = element_rect(fill = "transparent"), # get rid of legend bg
        legend.box.background = element_rect(fill = "transparent"))+
  ggtitle("Comptatges Normalitzats i transformats")

boxplot(dgeObj_norm$counts, col = dgeObj_norm$samples$group, las = 2, cex.axis = 0.7,
        main = "Contatges Normalitzats sense transformació",ylim = c(0, 10000))

gg1

```

```

#####
#####
#####

### Anàlisis de similitud
## Per poder visualitzar millor, el que podem fer es definir el nom de les mostres com
## COVID19_1, COVID19_2, etc, Bacterial_1, Bacterial_2, etc...
cohorts <- as.character(colData(se_75_filtrat)$cohort)
# Crear un vector nuu amb números únics enumerats per cohort
# Agrupar per cohort i enumerar cada element
cohort_counts <- ave(cohorts, cohorts, FUN = seq_along)
new_names <- paste0(cohorts, "_", cohort_counts)

colnames(se_75_filtrat) <- new_names

sampleDists <- dist(t(assays(se_75_filtrat)$log2count_nom))
#round(sampleDists, 1)

#Visualització heatmap
library(factoextra)
fviz_dist(sampleDists,
          show_labels = TRUE) +
  theme(axis.text.x = element_text(angle = 90, size = 6),
        axis.text.y = element_text(size = 6))

library(ggdendro)
library(dendextend)

## Calculem l'agrupament jeràrquic
hc <- hclust(sampleDists,method = 'ward.D2')

dend <- as.dendrogram(hc)

cohort_colors <- c(
  "Bacterial" = "#E78181", # vermell suau
  "COVID_19" = "#4DAF4A", # verd mitjà
  "healthy" = "#80B1D3" # blau clar
)
# Assignar colors per cohort
ordered_cohorts <- cohorts[match(labels(dend), colnames(se_75_filtrat))]

label_colors <- scales::col_factor(palette = "Dark2",
                                      domain = unique(ordered_cohorts))(ordered_cohorts)

# 5. Assignem els colors als textos del dendrograma
dend <- dend %>%
  set("labels_col", cohort_colors[cohorts[match(labels(dend), new_names)]])) %>%
  set("labels_cex", 0.6)

# 6. Dibuixem el dendrograma

```

```

plot(dend, main = "Dendrograma amb etiquetes per cohort")
legend("topright", legend = unique(ordered_cohorts),
       fill = unique(cohort_colors), border = NA, bty = "n", cex = 0.8)

library(limma)
library(RColorBrewer)

mds <- limma:::plotMDS(assays(se_75_filtrat)$log2count_nom, main = "Status", cex = 0.7)

# 4. Fer el plot amb colors per grup
plot(mds$x, mds$y,
      col = cohort_colors[cohorts],
      pch = 16,
      cex = 1.3,
      xlab = paste("Dimensió 1 (", round(mds$var.explained[1]*100, 1), "%)", sep = ""),
      ylab = paste("Dimensió 2 (", round(mds$var.explained[2]*100, 1), "%)", sep = ""),
      main = "MDS plot per cohort")
legend("topright", legend = names(cohort_colors),
       col = cohort_colors, pch = 16, cex = 0.8)
text(mds$x, mds$y,
      labels = colnames(se_75_filtrat),
      pos = 4, # posició: 3 = sobre del punt
      cex = 0.6,
      col = cohort_colors[cohorts])

mostres_eliminar <- c("Bacterial_2", "Bacterial_5", "Bacterial_7",
                      "COVID_19_2", "COVID_19_3", "COVID_19_29")

noms_mostres_eliminar <- colData(se_75_filtrat)$subject_id[which(colnames(se_75_filtrat) %in% mostres_eliminari)]
se_75_filtrat <- se_75_filtrat[, !colnames(se_75_filtrat) %in% mostres_eliminar]

#####
#####

## Comparació

library(compareGroups)
t <- compareGroups(cohort~age+gender+race+time_since_onset+hospitalized,
                     data=as.data.frame(colData(se_75_filtrat)), method=NA)
t <- createTable(t)
export2md(t, caption = "Comparació Bivariada")

```

```

### Anàlisi d'expressió diferencial

## Treballarem amb l'objecte dgeObj_norm que havíem creat
## al principi, per fer-ho haurem
## d'eliminar les 6 mostres que hem eliminat del objecte summarizedExperiment
dgeObj_norm <- dgeObj_norm[, !(colnames(dgeObj_norm) %in% noms_mostres_eliminar)]


group <- relevel(factor(colData(se_75_filtrat)$cohort), ref = "healthy")
age <- colData(se_75_filtrat)$age
gender <- factor(colData(se_75_filtrat)$gender)

## Creem la matriu de disseny
design <- model.matrix(~ 0+group + age + gender)
#colnames(design)
head(design)
## Estem interessants en les diferències entre els grups, necessitem especificar quines comparacions volen
cont.matrix <- makeContrasts(BacterialVsHealthy=groupBacterial - grouphealthy,
                               COVID19VsHealthy=groupCOVID_19 - grouphealthy, levels=design)
cont.matrix

## voom+limma
library(edgeR)
library(limma)
set.seed(myseed)
library(kableExtra)
sample(c("edgeR", "voom+limma", "DESeq2"), size = 1)
## Ens ha tocat el mètode voom+limma
## Transformem les dades amb voom
voomObj <- voom(dgeObj_norm, design)

## Realitzem el model amb la matriu de disseny, ens quedarà ajustat per edat i sexe
## A més acabem el procés amb la regularització del estimador del error utilitzant la funció
## eBayes
fit <- lmFit(voomObj)
fit2 <- contrasts.fit(fit, cont.matrix)
fit2 <- eBayes(fit2)

## Per Bacterial vs Healthy
res_bact <- topTable(fit2,coef=1,sort.by="p", number=nrow(fit2))
res_bact$significant <- with(res_bact, abs(logFC) > 1.5 & adj.P.Val < 0.05)
## Com a threshold posem el límit en el pvalo <0.05 i abs(logFC) > 1.5
##head(res_bact)

# Total de gens analitzats
total_genes <- nrow(res_bact)

# Gens diferencialment expressats
num_DEGs <- sum(res_bact$significant)

# Sobreeexpressats en Bacterial (logFC > 1.5)

```

```

num_upregulated <- sum(res_bact$logFC > 1.5 & res_bact$adj.P.Val < 0.05)

# Infraexpressats en Bacterial (logFC < -1.5)
num_downregulated <- sum(res_bact$logFC < -1.5 & res_bact$adj.P.Val < 0.05)

# Gens no diferencialment expressats
num_non_DEGs <- total_genes - num_DEGs

# Mostrar taula resum
summary_table <- data.frame(
  `Estat dels gens` = c(
    "Gens diferencialment expressats (|logFC| > 1.5 & FDR < 0.05)",
    " - Sobreexpressats en Bacterial (logFC > 1.5)",
    " - Infraexpressats en Bacterial (logFC < -1.5)",
    "Gens no diferencialment expressats",
    "Total de gens analitzats"
  ),
  `Nombre de gens` = c(
    num_DEGs,
    num_upregulated,
    num_downregulated,
    num_non_DEGs,
    total_genes
  )
)

kable(summary_table, caption = "Resum dels gens diferencialment
expressats en la comparació Bacterial vs Healthy") %>%
kable_styling(latex_options = c("striped", "hold_position"))

library(ggrepel)
ggplot(res_bact, aes(x = logFC, y = -log10(adj.P.Val), color = significant)) +
  geom_point(alpha = 0.6) +
  scale_color_manual(values = c("grey", "red")) +
  geom_vline(xintercept = c(-1.5, 1.5), linetype = "dashed", color = "black") +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "black") +
  geom_text_repel(data = subset(res_bact, significant),
                  aes(label = symbol), size = 3, max.overlaps = 15) +
  labs(title = "Volcano Plot: Bacterial vs Healthy",
       x = expression(log[2]^Fold^Change),
       y = expression(-log[10]^FDR)) +
  theme_minimal() +
  theme(legend.position = "none")

## COVID19 vs Healty
res_covid <- topTable(fit2, coef=2, sort.by="p", number=nrow(fit2))
res_covid$significant <- with(res_covid, abs(logFC) > 1.5 & adj.P.Val < 0.05)

# Total de gens analitzats

```

```

total_genes <- nrow(res_covid)

# Gens diferencialment expressats
num_DEGs <- sum(res_covid$significant)

# Sobreexpressats en Bacterial (logFC > 1.5)
num_upregulated <- sum(res_covid$logFC > 1.5 & res_covid$adj.P.Val < 0.05)

# Infraexpressats en Bacterial (logFC < -1.5)
num_downregulated <- sum(res_covid$logFC < -1.5 & res_covid$adj.P.Val < 0.05)

# Gens no diferencialment expressats
num_non_DEGs <- total_genes - num_DEGs

# Mostrar taula resum
summary_table <- data.frame(
  `Estat dels gens` = c(
    "Gens diferencialment expressats (|logFC| > 1.5 & FDR < 0.05)",
    " - Sobreexpressats en COVID19 (logFC > 1.5)",
    " - Infraexpressats en COVID19 (logFC < -1.5)",
    "Gens no diferencialment expressats",
    "Total de gens analitzats"
  ),
  `Nombre de gens` = c(
    num_DEGs,
    num_upregulated,
    num_downregulated,
    num_non_DEGs,
    total_genes
  )
)

kable(summary_table, caption = "Resum dels gens diferencialment
expressats en la comparació COVID19 vs Healthy") %>%
  kable_styling(latex_options = c("striped", "hold_position"))

ggplot(res_covid, aes(x = logFC, y = -log10(adj.P.Val), color = significant)) +
  geom_point(alpha = 0.6) +
  scale_color_manual(values = c("grey", "red")) +
  geom_vline(xintercept = c(-1.5, 1.5), linetype = "dashed", color = "black") +
  geom_hline(yintercept = -log10(0.05), linetype = "dashed", color = "black") +
  geom_text_repel(data = subset(res_covid, significant),
                  aes(label = symbol), size = 3, max.overlaps = 15) +
  labs(title = "Volcano Plot: COVID19 vs Healthy",
       x = expression(log[2]~Fold~Change),
       y = expression(-log[10]~FDR)) +
  theme_minimal() +
  theme(legend.position = "none")

##Cream una taula similar a la que crea decideTests però amb els pvalors ajustats

```

```

fdr_cutoff <- 0.05
logFC_cutoff <- 1.5
result_matrix <- sapply(colnames(fit2$coefficients), function(contrast) {
  tt <- topTable(fit2, coef = contrast, number = Inf, sort.by = "none")
  sign <- ifelse(tt$adj.P.Val < fdr_cutoff & tt$logFC > logFC_cutoff, 1,
                 ifelse(tt$adj.P.Val < fdr_cutoff & tt$logFC < -logFC_cutoff, -1, 0))
  return(sign)
})
## Aquesta taula ens dona 1 si el gene esta sobreexpresat, 0 si no es significatiu i -1 si està
## Infraexpresat
rownames(result_matrix) <- rownames(fit2$coefficients)
result_matrix <- as.data.frame(result_matrix)

summa.fit <- data.frame(BacterialVsHealthy=c(table(result_matrix$BacterialVsHealthy)[names(table(result_matrix$BacterialVsHealthy))], table(result_matrix$BacterialVsHealthy)[names(table(result_matrix$BacterialVsHealthy))], table(result_matrix$BacterialVsHealthy)[names(table(result_matrix$BacterialVsHealthy))]), COVID19VsHealthy=c(table(result_matrix$COVID19VsHealthy)[names(table(result_matrix$COVID19VsHealthy))], table(result_matrix$COVID19VsHealthy)[names(table(result_matrix$COVID19VsHealthy))], table(result_matrix$COVID19VsHealthy)[names(table(result_matrix$COVID19VsHealthy))]))

vc<- vennCounts(result_matrix)
vennDiagram(vc, include=c("Up", "Down"),
            counts.col=c("red", "blue"),
            circle.col = c("red", "blue", "green3"), cex=c(1,1,1))

### Anàlisis de significació

#BiocManager::install("clusterProfiler")
#BiocManager::install("org.Hs.eg.db")

res_covid <- topTable(fit2,coef=2,sort.by="p", number=nrow(fit2))

library(clusterProfiler)
library(org.Hs.eg.db)
topTab<- res_covid

allEntrezs <- as.vector(as.character(topTab$entrezid))
selectedEntrezsUP <- as.vector(as.character(subset(topTab,
                                                       (logFC> 1.5) & (adj.P.Val < 0.05))$entrezid))
#length(allEntrezs)
#length(selectedEntrezsUP)

ego_up <- enrichGO(gene      = selectedEntrezsUP,
                    universe   = allEntrezs,
                    OrgDb     = org.Hs.eg.db,
                    keyType   = "ENTREZID",
                    ont       = "BP",
                    pAdjustMethod = "BH",

```

```

        qvalueCutoff = 0.05,
        pvalueCutoff = 0.05,
        readable      = TRUE)

#head(ego_up)
ego_results <- data.frame(ego_up)
#write.csv(ego_results, "clusterProfiler_ORAresults_UpGO.csv")

head(ego_up)
dotplot(ego_up, showCategory=9, font.size=6)

cnetplot(ego_up, font.size=1)

#####
#####

### Comparació amb edgeR

library(edgeR)

# Eliminar les mostres atípiques
dgeObj_edge <- dgeObj_norm[, !(colnames(dgeObj_norm) %in% noms_mostres_eliminar)] 

# Factors d'interès
group <- relevel(factor(colData(se_75_filtrat)$cohort), ref = "healthy")
age <- colData(se_75_filtrat)$age
gender <- factor(colData(se_75_filtrat)$gender)

# Matriu de disseny (ajustat per edat i sexe)
design <- model.matrix(~ 0 + group + age + gender)
colnames(design) <- make.names(colnames(design))

# Crear objecte DGEList amb la informació de disseny
dge <- estimateDisp(dgeObj_edge, design)

# Ajust del model amb GLM
fit <- glmQLFit(dge, design)

# Definició dels contrastos
cont.matrix <- makeContrasts(
  BacterialVsHealthy = groupBacterial - grouphealthy,
  COVID19VsHealthy = groupCOVID_19 - grouphealthy,
  levels = design
)

```

```

# Test de quasi-likelihood per Bacterial vs Healthy
qlf_bact <- glmQLFTTest(fit, contrast = cont.matrix[, "BacterialVsHealthy"])

# Resultats ordenats i anotació de significatius
res_bact_edge <- topTags(qlf_bact, n = nrow(dge$counts))$table
res_bact_edge <- subset(res_bact_edge, abs(logFC) > 1.5 & FDR < 0.05)

res_bact <- topTable(fit2, coef=1, sort.by="p", number=nrow(fit2))
res_bact <- subset(res_bact, abs(logFC) > 1.5 & adj.P.Val < 0.05)

topGenes_voom <- rownames(res_bact)
topGenes_edge <- rownames(res_bact_edge)
library(ggvenn)
x = list(LimmaVoom = topGenes_voom, edgeR = topGenes_edge)
ggvenn(x, fill_color = c("#0073C2FF", "#EFC000FF"), stroke_size = 0.5, set_name_size = 3) +
  ggtitle("Comparacio Gens Diferencialment Expressats Voom i Edge Bacterial")

library(edgeR)

# Eliminar les mostres atípiques
dgeObj_edge <- dgeObj_norm[, !(colnames(dgeObj_norm) %in% noms_mostres_eliminar)]

# Factors d'interès
group <- relevel(factor(colData(se_75_filtrat)$cohort), ref = "healthy")
age <- colData(se_75_filtrat)$age
gender <- factor(colData(se_75_filtrat)$gender)

# Matriu de disseny (ajustat per edat i sexe)
design <- model.matrix(~ 0 + group + age + gender)
colnames(design) <- make.names(colnames(design))

# Crear objecte DGEList amb la informació de disseny
dge <- estimateDisp(dgeObj_edge, design)

# Ajust del model amb GLM
fit <- glmQLFit(dge, design)

# Definició dels contrastos
cont.matrix <- makeContrasts(
  BacterialVsHealthy = groupBacterial - grouphealthy,
  COVID19VsHealthy = groupCOVID_19 - grouphealthy,
  levels = design
)

# Test de quasi-likelihood per Bacterial vs Healthy
qlf_bact <- glmQLFTTest(fit, contrast = cont.matrix[, "COVID19VsHealthy"])

# Resultats ordenats i anotació de significatius
res_cov_edge <- topTags(qlf_bact, n = nrow(dge$counts))$table
res_cov_edge <- subset(res_cov_edge, (logFC > 1.5) & (FDR < 0.05))

```

```
res_covid <- topTable(fit2,coef=2,sort.by="p", number=nrow(fit2))
res_covid <- subset(res_covid, abs(logFC) > 1.5 & adj.P.Val < 0.05)

topGenes_voom <- rownames(res_covid)
topGenes_edge <- rownames(res_cov_edge)
library(ggvenn)
x = list(LimmaVoom = topGenes_voom, edgeR = topGenes_edge)
ggvenn(x, fill_color = c("#0073C2FF", "#EFC000FF"), stroke_size = 0.5, set_name_size = 3) +
  ggtitle("Comparacio Gens Diferencialment Expressats Voom i Edge COVID")
```