

# PEC 2

Ànnia Castillo Niell

## PEC 2

## TAULA DE CONTINGUTS

- INTRODUCCIÓ
- INFORME DE L'ANÀLISI
- INTRODUCCIÓ I OBJECTIUS
- MÈTODES
- RESULTATS
- DISCUSSIÓ
- REFERÈNCIES
- APÈNDIX
- PREPARACIÓ DE LES DADES
  - TREBALLAR AMB GSE38531\_SERIES\_MATRIX.TXT
  - DESCÀRREGA DELS FITXERS CEL A L'ENTORN LOCAL
- ANÀLISI EXPLORATÒRIA I CONTROL DE QUALITAT
  - DESCRIPCIÓ I DISTRIBUCIÓ DE LES DADES
- FILTRATGE DE DADES
- MATRIUS DE DISSENY I CONTRASTS
- ANOTACIÓ
- ANÀLISI DE SIGNIFICACIÓ BIOLÒGICA

## INTRODUCCIÓ

Aquest és el document generat per la PEC2. Està vinculat al repositori de github <https://github.com/acniell/PEC2.git>. Es mostren tots els bloc de codi que s'han fet servir, tot i que no tots tindran sortida en el document. Els gràfics no són la generació directa dels chunks de codi per evitar sobrecarregar el document i la RAM, són imatges adjuntes de menor resolució i es troben a l'apartat d'apèndix. Per millorar la visualització en cas que es vulgui veure'ls amb més claredat, caldria correr el codi per a generar la imatge.

# INFORME DE L'ANÀLISI

## INTRODUCCIÓ I OBJECTIUS

Les bactèries multiresistents són un dels grans problemes del sistema sanitari, i que està en augment en l'actualitat, cosa que justifica la investigació pel desenvolupament d'antibiòtics per poder continuar tractant les infeccions. Una de les primeres bactèries que es van identificar va ser *Staphylococcus aureus*, un coc gram positiu, que amb el desenvolupament de les penicil·lines, ràpidament es va fer resistent a aquestes. El que presenten fenotip de resistència s'anomenen MRSA (methicillin resistant *staphylococcus aureus*) i es va aïllar el 1960 a nivell hospitalari. I tot i que no està present en totes les comunitats, cada vegada és més comú el MRSA comunitari (1980-1990 ja es detecta a Austràlia i EEUU). Per tant, s'han hagut d'anar desenvolupant antibiòtics per poder anar tractant les diferents bactèries multiresistents que van sorgint, i dos dels antibiòtics considerats fins ara de primera línia contra MRSA són el linezolid i la vancomicina (actualment també s'ha comercialitzat la ceftarolina que també és efectiva contra aquest i està disponible als hospitals ICS).

L'estudi original d'on hem extret les dades està disponible a <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE38531>, i es van analitzar ratolins infectats i no infectats per MRSA (específicament MRSA USA-300), i no tractats i tractats (amb vanco o line). En aquests es va mirar la producció de toxines bacterianes i citocines, així com les diferències en l'expressió gènica dels diferents grups per intentar identificar quin antibiòtic és més efectiu en el tractament de la infecció per MRSA.

En el nostre cas, volem comparar l'efecte de no tractar vs tractar, i dins d'aquest segon grup el tractament amb line o vanco. L'efecte del tractament el mirarem segons l'expressió gènica que presenta cada grup i quines diferents vies implicades i hi ha, per així poder inferir si un dels dos antibiòtics és més efectiu que l'altre. Utilitzarem només 24 mostres de les recollides a l'estudi mencionat anteriorment i es crearan 3 comparacions:

- Ratolins infectats vs. no infectats sense tractament.
- Ratolins infectats vs. no infectats tractats amb linezolid.

- Ratolins infectats vs. no infectats tractats amb vancomicina.

## MÈTODES

Com s'ha mencionat anteriorment, les dades utilitzades són les pertanyents a l'estudi GSE38531

(<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE38531>), de les quals s'ha aplicat un procés de selecció aleatòria per obtenir 24 mostres. L'anàlisi per tant s'ha aplicat a aquest subgrup de 24 mostres.

S'ha realitzat un procés d'anàlisi de qualitat, normalització i filtratge de les dades per poder realitzar les diferents comparacions entre grups. De les mostres, s'han seleccionat els gens amb més variabilitat (el 10% amb més variabilitat) que han estat un total de 4511. Per la comparació de grups, s'ha construït la matriu de disseny i així crear els grups de comparació mencionats anteriorment. Per a cada contrast s'han analitzat els gens diferencialment expressats i s'han seleccionat els que es consideren significatius amb els criteris de  $p$  valor ajustat  $<0.05$  i  $\log_{2}FC > 1$ .

Per poder interpretar els resultats obtinguts, s'ha aplicat un procés d'anotació utilitzant Gene Ontology i associant els resultats especialment als identificadors d'ENTREZ i descriptors.

Finalment s'ha realitzar l'anàlisi de significació biològica per identificar les vies biològiques implicades en aquests gens diferencialment expressats i s'han comparat els resultats entre els diferents grups.

## RESULTATS

Després de l'anàlisi realitzat hem trobat que: amb el subgrup de mostres analitzades, no hi ha expressió diferencial de gens entre els grups de no infectat i infectat que **NO** han rebut tractament. **SÍ** que hem trobat diferències en els grups que han estat tractats amb vancomicina i linezolid. D'aquests grups, el que crida més l'atenció és que d'aquests gens diferencialment expressats, tenen tendència a estar infra expressats i són molt similars les troballes entre la vancomicina i el linezolid (tot i que no idèntiques).

Això suggereix que l'efecte biològic d'aquests antibiòtics en els ratolins és similar, cosa que lògicament també és coherent ja que són dos fàrmacs efectius contra la infecció per MRSA.

Les vies que es veuen sobretot implicades són aquelles relacionades amb el sistema immunitari: regulació de la resposta innata, adhesió cel·lular, activació cel·lular i leucocitària, diferenciació leucocitària... i són comunes tant a la vancomicina com el linezolid.

## DISCUSSIÓ

Amb els resultats obtinguts hi ha varies coses que criden l'atenció:

- En primer lloc, crida l'atenció que no hi hagi expressió diferencial de gens en ratolins infectats i no infectats. El sentit comú diu que davant una infecció hi ha d'haver certes vies de resposta immunitària destinades a defensar l'organisme que haurien d'estar molt més activades en els ratolins infectats, ja que és el nostre sistema de defensa. Per tant, no trobar diferències en aquest grup de comparació és molt estrany. Podria ser que hi hagués un error en els càlculs realitzats o bé que el resultat trobat no sigui extrapolable perquè hem realitzat l'anàlisi amb un subgrup de mostres i no el dataset sencer.
- En segon lloc s'ha evidenciat que el fet de rebre tractament antibiòtic modula de forma clara la resposta immunitària a l'hoste. Tot i així, destaca però la GRAN similitud dels resultats obtinguts amb linezolid i vancomicina.  
El linezolid és una oxazolidinona i la vancomicina és un glucopèptid, dues famílies diferents d'antibiòtics i que són efectives contra la infecció per MRSA però per vies diferents. Per tant, també sorprèn molt que presentant mecanismes d'acció diferents tinguin un resultat tan similar. El resultat final però d'aquestes vies està clar i és a disminuir la càrrega bacteriana a l'hoste.
- Per últim, destacar que el que veiem és una disminució en l'activació d'aquestes vies, no una sobreactivació. Tot i que el primer que es pot reflexionar és que hauria de ser a l'inversa, ja que l'organisme ha de lluitar contra la infecció, en la sèpsia està descrit que la resposta immune pot ser excessiva, passant a ser perjudicial. Podria ser que precisament l'administració d'antibiòtic equilibrés les respostes immunes, evitant una sobreinflamació en l'hoste, i per tant traduïnt-se a infraexpressió gènica.

## REFERÈNCIES

- Guies mèdiques de Surviving sepsis campaign. Maneig òptim del codi sèptic en els humans: <https://www.sccm.org/clinical-resources/guidelines/guidelines/surviving-sepsis-guidelines-2021>
- Història de MRSA: <https://www.health.state.mn.us/diseases/staph/mrsa/basics.html>
- III update d'infeccions multiresistents (MRSA) organitzat per SOCMIC el 11 de desembre del 2014 a l'Hospital Dr. Josep Trueta de Girona: presentació power point de la sessió cedida per Dra. Lopez de Arbina.
- Documentació de l'assignatura anàlisi de dades òmiques UOC. Semestre hivern 2024-2025.

## APÈNDIX

### PREPARACIÓ DE LES DADES

Carreguem l'arxiu de allTargets per fer la selecció.

```
allTargets <- read.table("allTargets.txt", header = TRUE, se
```

Apliquem el codi cedit:

```
filter_microarray <- function(allTargets, seed =123) {  
  set.seed(seed)  
  filtered <- subset(allTargets, time != "hour 2")  
  
  # Dividir el dataset por grupos únicos de 'infection' + 'a  
  filtered$group <- interaction(filtered$infection, filtered  
  
  # Seleccionar 4 muestras al azar de cada grupo  
  selected <- do.call(rbind, lapply(split(filtered, filtered  
    if (nrow(group_data) > 4) {  
      group_data[sample(1:nrow(group_data), 4), ]  
    } else {  
      group_data  
    }  
  }  
}))
```

```

# Obtener los índices originales como nombres de las filas
original_indices <- match(selected$sample, allTargets$samp

# Modificar los rownames usando 'sample' y los índices ori
rownames(selected) <- paste0(selected$sample, ".", origina

# Eliminar la columna 'group' y devolver el resultado
selected$group <- NULL
return(selected)
}

```

Fem la selecció aleatòria i la guardem en un nou fitxer per si el necessitem a posteriori. En aquest cas farem servir el número de DNI per l'aleatorització.

```

# Aplicar la función (cambiar 123 por vuestro ID de la UOC u
resultat_mostra <- filter_microarray(allTargets, seed=415774
print(resultat_mostra)

### Aprofitarem i ajustarem ara els noms, perquè no s'havia

resultat_mostra$infection <- gsub("S\\. aureus USA300", "aur
resultat_mostra$time <- gsub("hour ", "", resultat_mostra$ti
resultat_mostra$agent <- gsub("linezolid", "line", gsub("van
resultat_mostra$nom <- paste0(
  sub("GSM944", "", resultat_mostra$sample), "_",
  resultat_mostra$infection, "_",
  resultat_mostra$time, "_",
  resultat_mostra$agent
)

write.table(resultat_mostra, file = file.path(dataDir, "most
          sep = "\t", row.names = FALSE)

```

Ja tenim una selecció de 24 mostres aleatòries del dataset amb les que treballar.

Ara tenim dues opcions: o descarreguem de la pàgina de GEO el supplementary File amb els arxius CEL a dins i seleccionem els que havien sortit amb la selecció aleatòria o bé treballem amb el fitxer

"series matrix" disponible a la pàgina de GEO. Mostrem primer com es faria amb la segona opció, però treballarem amb la primera (descàrrega a l'entorn local del fitxers CEL).

## TREBALLAR AMB GSE38531\_SERIES\_MATRIX.TXT

```
BiocManager::install("GEOquery")
library(GEOquery)
GSE38531<-getGEO("GSE38531", GSEMatrix = TRUE, parseCharacter
```

Ara accedirem als diferents blocs d'informació per veure que estigui tot correcte a primera vista:

```
ES<- GSE38531[[1]]
matriu_expressio<-exprs(ES)
head(matriu_expressio)
fenodata<-pData(ES)
head(fenodata)
```

Ara volem agafar només la informació les mostres que hem seleccionat amb l'aleatorització.

```
mostres <- read.table(file.path(dataDir, "mostra_seleccionad
                        header = TRUE, sep = "\t", stringsAsFa
noms_mostra<-mostres$nom #llista dels que volem.
```

```
sampleNames(ES)->noms_ES
coincidencies<-intersect(noms_mostra, noms_ES)
coincidencies #veiem que efectivament hi ha 24 elements a la
ES2<-ES[,coincidencies]
ES2
```

Ja tenim creat el nostre expressionSet amb la mostra que hem seleccionat aleatòriament.

```
#ens el guardem per si el necessitem més tard naïve  
write.csv(exprs(ES2), file = file.path(dataDir, "ES2.csv"))
```

## DESCÀRREGA DELS FITXERS CEL A L'ENTORN LOCAL

A la pàgina web de GEO

(<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE38531>) veiem que hi ha un arxiu anomenat Supplementary File on tenim tots els arxius CEL. Una vegada descarregats, eliminem els que no ens interessin després de fer la selecció aleatòria, ens quedem els que sí, i els hi ajustem el nom:



En aquest cas farem servir la llibreria oligo (els passos estan basats en el document cedit per la UOC "Statistical Analysis of Microarray data" [https://aspteaching.github.io/Omics\\_Data\\_Analysis-Case\\_Study\\_1-Microarrays/Case\\_Study\\_1-Microarrays\\_Analysis.html](https://aspteaching.github.io/Omics_Data_Analysis-Case_Study_1-Microarrays/Case_Study_1-Microarrays_Analysis.html) ).

```
library(oligo) #pels elements CEL  
library(Biobase)  
arxius_CEL <- list.celfiles("/Users/annia/Library/CloudStorage  
phenoData<-AnnotatedDataFrame(data = mostres)  
rawData<-read.celfiles(arxius_CEL, phenoData=phenoData) #cre
```

Ja hem llegit els arxius CEL i hem associat la informació de l'element mostres, que té la selecció aleatòria que hem fet anteriorment.

## ANÀLISI EXPLORATÒRIA I CONTROL DE QUALITAT

Prèviament a la normalització, anem a veure la qualitat de les nostres dades i que no hi hagi cap outlier que hàgim de gestionar. Això ho fem amb el paquet arrayQualityMetrics. Els documents generats amb aquesta funció, es poden trobar al repositori github.

```
BiocManager::install("arrayQualityMetrics")  
library(arrayQualityMetrics)
```



```
arrayQualityMetrics(rawData, force=FALSE)
```

El document generat ens aporta un informe on podem identificar a través de boxplot i PCA si tenim outliers. En el document mencionat anteriorment de la UOC recomanen revisar els outliers que s'identifiquin com a tal en  $> 0 = 3$  proves diferents dins la realització d'un estudi de qualitat de les dades. En el nostre cas veiem que es detecta un outlier en dues proves, pel que de moment el mantindrem dins del nostre dataset.

Seguirem per normalitzar les dades que hem generat anteriorment amb RMA tal com es proposa:

```
ESet <- rma(rawData)
rawData
```

Una vegada creat el nostre expressionSet, assegurarem que el nom de les mostres sigui el de la columna sample del fitxer mostres i farem una consulta ràpida dels elements per veure que estiguin bé:

```
mostres$nom
sampleNames(ESet) <- mostres$nom
sampleNames(ESet)
pData(ESet)
head(ESet)
```

## DESCRIPCIÓ I DISTRIBUCIÓ DE LES DADES

Una vegada realitzada la normalització, i ens hem assegurat que les dades estan correctament creades, anem a fer una exploració bàsica de les dades que tenim:

```
matriu2 <- exprs(ESet)
dim(matriu2)
```

```
## [1] 45101 24
```

```
str(matriu2)
```

```
## num [1:45101, 1:24] 7.38 10.14 11.05 6.9 8.37 ...
## - attr(*, "dimnames")=List of 2
## ..$ : chr [1:45101] "1415670_at" "1415671_at" "1415672_
## ..$ : chr [1:24] "850_aureus_24_line" "857_aureus_24_li
```

```
head(matriu2)
```

```
##          850_aureus_24_line 857_aureus_24_line 843_aureus_24_line
## 1415670_at          7.377539          7.661332
## 1415671_at          10.140773          10.267031
## 1415672_at          11.054640          10.602395
## 1415673_at           6.896972           6.942270
## 1415674_a_at         8.368597           8.377307
## 1415675_at           7.882178           7.772171
##          864_aureus_24_line 861_uninfected_0_line 840_uninfected_0_line
## 1415670_at           7.438351           7.423553
## 1415671_at          10.389609          10.391099
## 1415672_at          11.401611          10.406412
## 1415673_at           7.019661           6.552072
## 1415674_a_at         8.294469           8.784424
## 1415675_at           7.772919           8.287480
##          847_uninfected_0_line 854_uninfected_0_line
## 1415670_at           6.664764           7.503319
## 1415671_at          10.618936          10.625470
## 1415672_at          10.951508          11.548634
## 1415673_at           6.391413           6.987974
## 1415674_a_at         8.287116           8.294712
## 1415675_at           7.419386           7.957054
##          849_aureus_24_untreated 842_aureus_24_untreated
## 1415670_at           7.636685           7.512
## 1415671_at          10.301996          10.271
## 1415672_at          11.211378          11.166
## 1415673_at           7.049801           7.066
## 1415674_a_at         8.434156           8.435
## 1415675_at           7.991979           7.891
##          835_aureus_24_untreated 863_aureus_24_untreated
## 1415670_at           7.294563           7.203
## 1415671_at          10.550219          10.534
## 1415672_at          10.985514          11.249
## 1415673_at           6.866069           6.526
## 1415674_a_at         8.463667           8.491
```

```

## 1415675_at          7.876645          7.895
##                845_uninfected_0_untreated 859_uninfected_0_
## 1415670_at          7.110719
## 1415671_at          10.678616
## 1415672_at          11.045148
## 1415673_at          6.740297
## 1415674_a_at        8.400129
## 1415675_at          7.850291
##                852_uninfected_0_untreated 838_uninfected_0_
## 1415670_at          7.372845
## 1415671_at          10.214536
## 1415672_at          11.112127
## 1415673_at          7.173377
## 1415674_a_at        8.407017
## 1415675_at          7.859973
##                844_aureus_24_vanco 851_aureus_24_vanco 858_
## 1415670_at          7.274382          7.092627
## 1415671_at          10.548459          10.481792
## 1415672_at          11.126907          10.772960
## 1415673_at          6.287330          6.773963
## 1415674_a_at        8.477683          8.212788
## 1415675_at          7.743458          7.761416
##                865_aureus_24_vanco 848_uninfected_0_vanco 8
## 1415670_at          7.441754          7.317117
## 1415671_at          10.506133          10.325506
## 1415672_at          11.613814          11.296011
## 1415673_at          7.022686          7.009234
## 1415674_a_at        8.257270          8.349285
## 1415675_at          7.927027          7.955926
##                862_uninfected_0_vanco 855_uninfected_0_vanc
## 1415670_at          7.124336          7.08962
## 1415671_at          10.386109          10.60282
## 1415672_at          10.925898          11.24381
## 1415673_at          6.588123          6.89883
## 1415674_a_at        8.534151          8.49530
##  - - - - -

```

```
summary(matriu2)
```

```

## 850_aureus_24_line 857_aureus_24_line 843_aureus_24_line
## Min. : 2.416      Min. : 2.468      Min. : 2.447
## 1st Qu.: 4.391    1st Qu.: 4.411    1st Qu.: 4.435
## Median : 5.593    Median : 5.593    Median : 5.627

```

##	Mean : 5.866	Mean : 5.864	Mean : 5.869
##	3rd Qu.: 6.951	3rd Qu.: 6.917	3rd Qu.: 6.931
##	Max. :15.103	Max. :15.147	Max. :15.111
##	861_uninfected_0_line	840_uninfected_0_line	847_uninfected_0_line
##	Min. : 2.443	Min. : 2.428	Min. : 2.4
##	1st Qu.: 4.362	1st Qu.: 4.405	1st Qu.: 4.5
##	Median : 5.546	Median : 5.587	Median : 5.6
##	Mean : 5.861	Mean : 5.859	Mean : 5.8
##	3rd Qu.: 6.910	3rd Qu.: 6.900	3rd Qu.: 6.8
##	Max. :15.084	Max. :15.079	Max. :15.1
##	854_uninfected_0_line	849_aureus_24_untreated	842_aureus_24_untreated
##	Min. : 2.436	Min. : 2.421	Min. : 2
##	1st Qu.: 4.386	1st Qu.: 4.347	1st Qu.: 4
##	Median : 5.593	Median : 5.561	Median : 5
##	Mean : 5.870	Mean : 5.870	Mean : 5
##	3rd Qu.: 6.978	3rd Qu.: 7.004	3rd Qu.: 7
##	Max. :15.073	Max. :15.091	Max. :15
##	835_aureus_24_untreated	863_aureus_24_untreated	845_uninfected_24_untreated
##	Min. : 2.480	Min. : 2.493	Min. : 2
##	1st Qu.: 4.410	1st Qu.: 4.409	1st Qu.: 4
##	Median : 5.601	Median : 5.599	Median : 5
##	Mean : 5.861	Mean : 5.861	Mean : 5
##	3rd Qu.: 6.887	3rd Qu.: 6.892	3rd Qu.: 7
##	Max. :15.106	Max. :15.105	Max. :15
##	859_uninfected_0_untreated	852_uninfected_0_untreated	
##	Min. : 2.423	Min. : 2.405	
##	1st Qu.: 4.382	1st Qu.: 4.378	
##	Median : 5.595	Median : 5.579	
##	Mean : 5.865	Mean : 5.870	
##	3rd Qu.: 6.958	3rd Qu.: 6.975	
##	Max. :15.068	Max. :15.100	
##	838_uninfected_0_untreated	844_aureus_24_vanco	851_aureus_24_vanco
##	Min. : 2.434	Min. : 2.459	Min. : 2
##	1st Qu.: 4.311	1st Qu.: 4.409	1st Qu.: 4
##	Median : 5.525	Median : 5.591	Median : 5
##	Mean : 5.871	Mean : 5.855	Mean : 5
##	3rd Qu.: 7.056	3rd Qu.: 6.880	3rd Qu.: 7
##	Max. :15.115	Max. :15.089	Max. :15
##	858_aureus_24_vanco	865_aureus_24_vanco	848_uninfected_0_untreated
##	Min. : 2.421	Min. : 2.466	Min. : 2.476
##	1st Qu.: 4.360	1st Qu.: 4.367	1st Qu.: 4.412
##	Median : 5.571	Median : 5.569	Median : 5.613
##	Mean : 5.862	Mean : 5.857	Mean : 5.866
##	3rd Qu.: 6.987	3rd Qu.: 6.968	3rd Qu.: 6.941
##	Max. :15.079	Max. :15.078	Max. :15.095

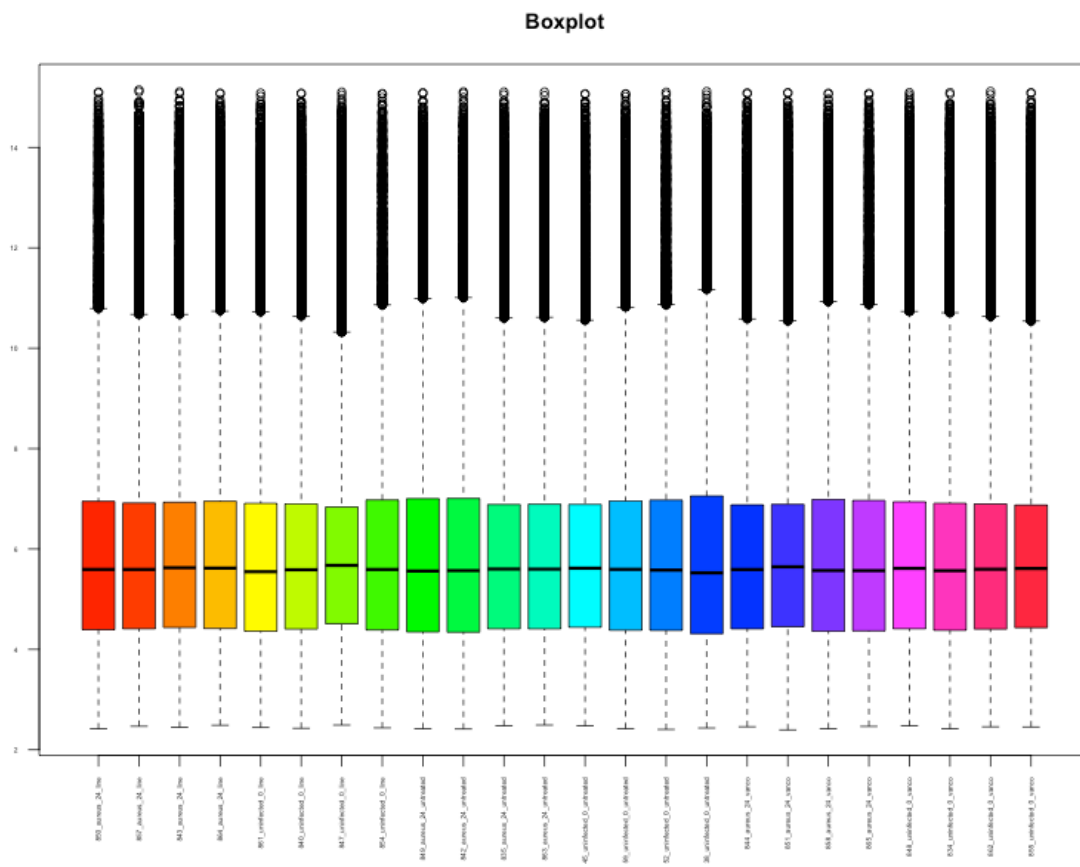
```
## 834_uninfected_0_vanco 862_uninfected_0_vanco 855_uninfe
## Min. : 2.416 Min. : 2.458 Min. : 2
## 1st Qu.: 4.379 1st Qu.: 4.403 1st Qu.: 4
## Median : 5.567 Median : 5.598 Median : 5
## Mean : 5.863 Mean : 5.860 Mean : 5
## 3rd Qu.: 6.911 3rd Qu.: 6.897 3rd Qu.: 6
##
```

```
sum(is.na(matriu2))
```

```
## [1] 0
```

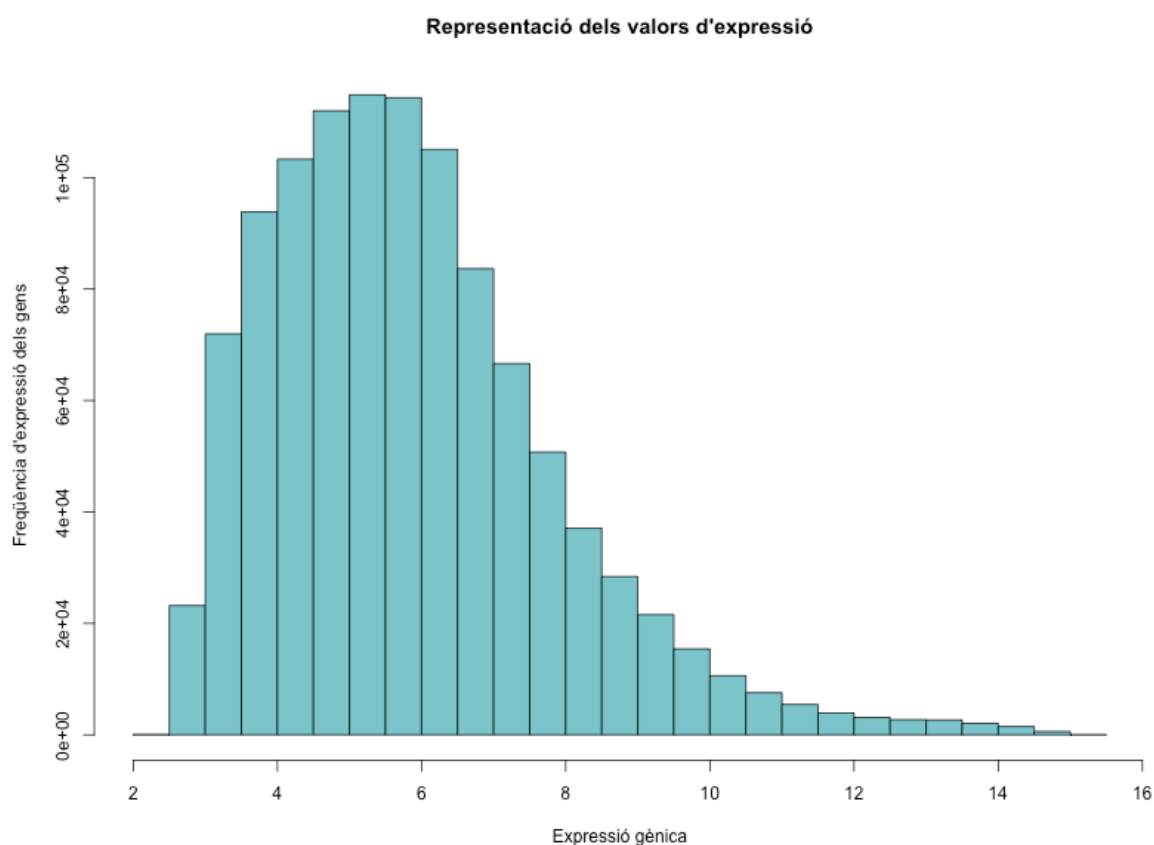
Tot seguit farem una representació gràfica per veure com es distribueixen de forma visual:

```
output_file <- file.path(resultsDir, "boxplot.png")
png(file = output_file, width = 1000, height = 800)
boxplot(matriu2, main="Boxplot", col = rainbow(ncol(matriu2))
dev.off()
```



Veiem que globalment les medianes estan alineades i les distribucions són similars. No s'observa cap mostra que sobresurti més que la resta.

```
output_file <- file.path(resultsDir, "histograma_expressio.p
png(file = output_file, width = 800, height = 600)
hist(as.numeric(as.matrix(matriu2)),
     main="Representació dels valors d'expressió",
     xlab="Expressió gènica",
     ylab="Freqüència d'expressió dels gens",
     col="cadetblue3", breaks=35)
dev.off()
```

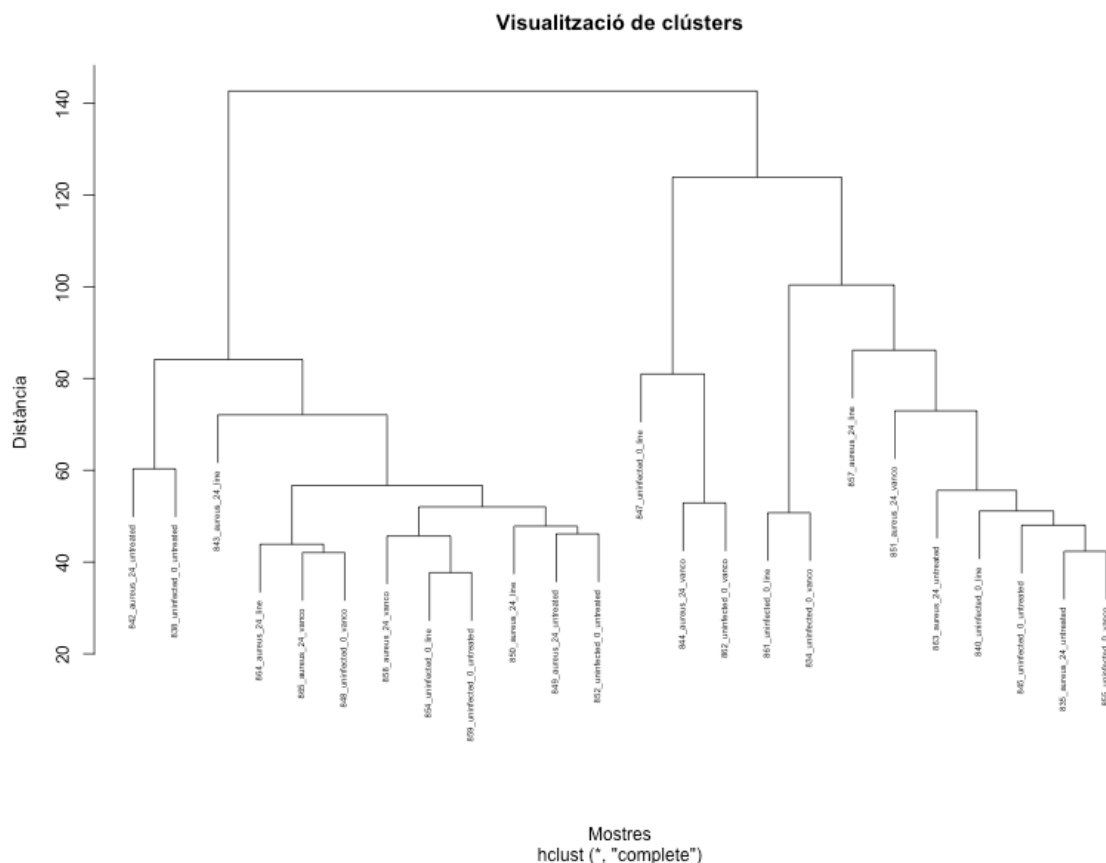


Amb aquest histograma veiem clarament una cua a la dreta, cosa que pot suggerir que les dades hagin passat per una transformació tipus logarítmica.

Continuem amb un dendrograma per veure com s'agrupen les mostres.

```
output_file <- file.path(resultsDir, "dendograma.png")
png(file = output_file, width = 800, height = 600)
distancia <- dist(t(matriu2))
clustering <- hclust(distancia)
```

```
plot(clustering,
     main="Visualització de clústers",
     xlab="Mostres", ylab="Distància",
     cex=0.5)
dev.off()
```



No veiem que hi hagi una diferenciació clara entre tractament rebut (res, linezolid o vancomicina) però a simple vista podria ser que hi hagia més mostres de les 24 a l'esquerra i de 0h a la dreta. Tot i així, no m'impresiona que hi hagi un efecte batch que no hàgim detectat.

Procedim a fer una anàlisi de components principals:

```
pca <- prcomp(t(matriu2), scale.=TRUE)
summary(pca)
```

```
## Importance of components:
##
##          PC1      PC2      PC3      PC4
## Standard deviation 111.3073 76.1261 57.1432 50.18384
## Proportion of Variance 0.2747 0.1285 0.0724 0.05584
## Cumulative Proportion 0.2747 0.4032 0.4756 0.53144
##
##          PC7      PC8      PC9      PC10
```

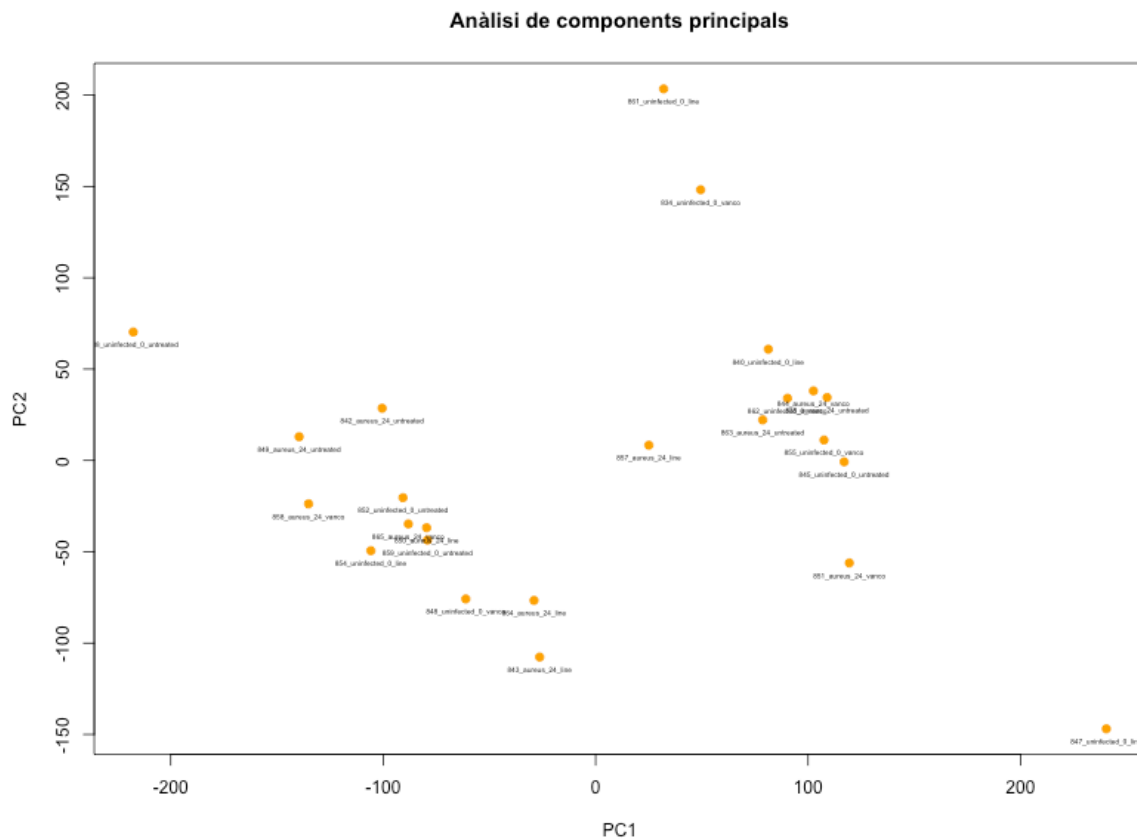
```
## Standard deviation      38.2284 36.62065 35.79419 35.59101
## Proportion of Variance  0.0324 0.02973 0.02841 0.02809
## Cumulative Proportion  0.6426 0.67236 0.70077 0.72886
##                          PC13    PC14    PC15    PC16
## Standard deviation      32.48063 31.7124 31.60874 30.70697
## Proportion of Variance  0.02339 0.0223 0.02215 0.02091
## Cumulative Proportion  0.80378 0.8261 0.84824 0.86914
##                          PC19    PC20    PC21    PC2
## Standard deviation      30.05388 29.45554 28.18123 27.5331
## Proportion of Variance  0.02003 0.01924 0.01761 0.0168
```

Amb aquests resultats inicials veiem que la primera component ens explica el 27,47% i la segona 18,25%.

```
output_file <- file.path(resultsDir, "plotPCA.png")
png(file = output_file, width = 800, height = 600)
plot(pca$x[,1], pca$x[,2],
     xlab="PC1",
     ylab="PC2",
     main="Anàlisi de components principals",
     pch=19, col="orange")

text(pca$x[,1], pca$x[,2], labels=colnames(matriu2), pos=1,
     dev.off())
```





Veiem que algunes mostres sí que es veuen aïllades de la resta pel que hi pot haver diferències en l'expressió gènica.

## FILTRATGE DE DADES

Per veure primer quins són els gens amb més variabilitat, mirarem les seves distribucions i les representarem per visualitzar-ho millor.

```
desv_std<-apply(matriu2, 1, sd)
sorted_desv_std<-sort(desv_std)
head(sorted_desv_std)
```

```
## 1428361_x_at 1416624_a_at 1416642_a_at 1451675_a_at 14394
## 0.02222959 0.02269365 0.02534559 0.02560192 0.0
```

```
output_file <- file.path(resultsDir, "plotSD.png")
png(file = output_file, width = 800, height = 600)
plot(1:length(sorted_desv_std), sorted_desv_std,
     main="Variabilitat dels gens",
     xlab="Índex de variabilitat",
     ylab="SD",
```

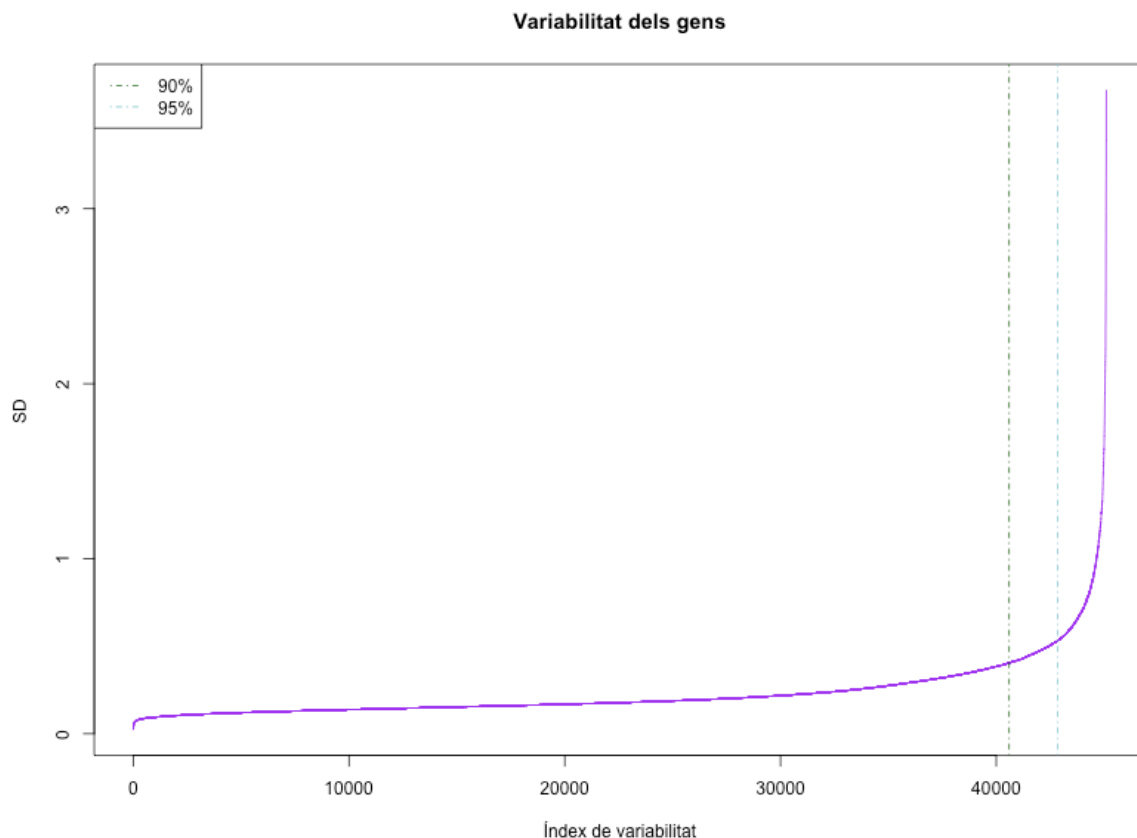
```

type="l", col="purple", lwd=1)

# Afegir línies verticals als percentils 90% i 95%
abline(v=length(sorted_desv_std)*c(0.9, 0.95), col=c("darkgr

# Llegenda
legend("topleft", legend=c("90%", "95%"),
      col=c("darkgreen", "cadetblue3"), lwd=1, lty=4)
dev.off()

```



Una vegada hem vist això anirem a buscar, tal com se suggereix, el 10% més variable.

```

limit90<- quantile(desv_std, 0.9)
pool10 <- matriu2[desv_std >= limit90, ]
dim(pool10)

```

```
## [1] 4511 24
```

Finalment, ens quedem doncs com a mostra amb 4511 gens de 24 mostres.

## MATRIUS DE DISSENY I CONTRASTS

Ja tenim anteriorment generada una taula amb la fenodata que necessitem, anem a refrescar-la:

```
str(mostres)
```

```
## 'data.frame':    24 obs. of  5 variables:
## $ sample      : chr  "GSM944850" "GSM944857" "GSM944843" "G
## $ infection: chr  "aureus" "aureus" "aureus" "aureus" ..
## $ time        : int   24 24 24 24 0 0 0 0 24 24 ...
## $ agent       : chr  "line" "line" "line" "line" ...
## $ nom        : chr  "850_aureus_24_line" "857_aureus_24_li
```

```
mostres$infection <- factor(mostres$infection, levels=c("uni
mostres$agent <- factor(mostres$agent, levels=c("untreated",
#trec les variable sque no necessitaré per les comparacions.
mostres <- mostres[, !(colnames(mostres) %in% c("time", "nom
#str(mostres)
```

En les comparacions que ens demanen, en cap moment es parla de línia temporal, pel que de moment, agafarem les mostres que tenim, però obviarem la línia temporal (és a dir les hores). Agruparem les mostres independentment de si és 0h o 24h; així aconseguirem dos grans grups: infectats i no infectats. I tot seguit els tres subgrups de tractament: **no tractament, vancomicina o linezolid**.

```
disseny <- model.matrix(~ 0 + infection * agent, data = most
colnames(disseny) <- gsub(":", "_", colnames(disseny))
disseny
```

```
##      infectionuninfected infectionaureus agentline agentvan
## 1              0              1          1
## 2              0              1          1
## 3              0              1          1
```

## 4	0	1	1
## 5	1	0	1
## 6	1	0	1
## 7	1	0	1
## 8	1	0	1
## 9	0	1	0
## 10	0	1	0
## 11	0	1	0
## 12	0	1	0
## 13	1	0	0
## 14	1	0	0
## 15	1	0	0
## 16	1	0	0
## 17	0	1	0
## 18	0	1	0
## 19	0	1	0
## 20	0	1	0
## 21	1	0	0
## 22	1	0	0
## 23	1	0	0
## 24	1	0	0
##	infectionaureus_agentline		infectionaureus_agentvanco
## 1	1		0
## 2	1		0
## 3	1		0
## 4	1		0
## 5	0		0
## 6	0		0
## 7	0		0
## 8	0		0
## 9	0		0
## 10	0		0
## 11	0		0
## 12	0		0
## 13	0		0
## 14	0		0
## 15	0		0
## 16	0		0
## 17	0		1
## 18	0		1
## 19	0		1
## 20	0		1
## 21	0		0
## 22	0		0
## 23	0		0

```
## 24                                0                                0
## attr(,"assign")
## [1] 1 1 2 2 3 3
## attr(,"contrasts")
## attr(,"contrasts")$infection
## [1] "contr.treatment"
##
## attr(,"contrasts")$agent
```

```
colnames(disseny)
```

```
## [1] "infectionuninfected"      "infectionaureus"
## [3] "agentline"                "agentvanco"
## [5] "infectionaureus_agentline" "infectionaureus_agentva
```

I tal com ens han proposat, dissenyarem els tres contrastos:

- Infectats vs no infectats, en el grup *sense tractament* (seria grup base).
- Infectats vs no infectats, en el grup de tractament amb *linezolid*.
- Infectats vs no infectats, en el grup de tractament amb *vancomicina*.

Això ho farem amb el paquet limma.

```
library(limma)
contrasts <- makeContrasts(
  Infectats_vs_NoInfectats_Untreated = infectionaureus - inf
  Infectats_vs_NoInfectats_Linezolid = infectionaureus_agent
  Infectats_vs_NoInfectats_Vancomycin = infectionaureus_agen
  levels = disseny
)

print(contrasts)
```

```
##                                Contrasts
## Levels                        Infectats_vs_NoInfectats_Unt
##   infectionuninfected
```

```
## infectionaureus
## agentline
## agentvanco
## infectionaureus_agentline
## infectionaureus_agentvanco
##
## Contrasts
## Levels Infectats_vs_Noinfectedats_Lin
## infectionuninfected
## infectionaureus
## agentline
## agentvanco
## infectionaureus_agentline
## infectionaureus_agentvanco
##
## Contrasts
## Levels Infectats_vs_Noinfectedats_Van
## infectionuninfected
## infectionaureus
## agentline
## agentvanco
## infectionaureus_agentline
... ..
```

Finalment queda fer l'estimació del model, ja que volem detectar quins gens tenen expressió diferencial a través de la comparació de cada gen en les dues situacions experimentals (infectat vs no infectat amb un mateix tractament en comú (no tractament, line o vanco).

Els resultats generats els guardarem per si ens fessin falta més endavant.

```
fit <- lmFit(pool10, design = disseny)
fit2 <- contrasts.fit(fit, contrasts)
fit2 <- eBayes(fit2)

# Obtenir els resultats per a cada comparació
resultats_untreated <- topTable(fit2, coef="Infectats_vs_Noinfectedats_Lin")
head(resultats_untreated)
```

```
##          logFC AveExpr      t    P.Value adj
## 1420357_s_at -1.3255585 5.107748 -2.695146 0.01285520 0.9
## 1454685_at  -0.7207441 7.339178 -2.425748 0.02344250 0.9
```

```
## 1427883_a_at 1.1634450 5.447073 2.355059 0.02733148 0.9
## 1427820_at -1.2187198 8.183081 -2.286833 0.03163894 0.9
## 1436998_at -0.9657128 5.586410 -2.146802 0.04247593 0.9
... ..
```

```
resultats_line<- topTable(fit2, coef="Infectats_vs_Noinfected")
head(resultats_line)
```

```
##          logFC AveExpr      t      P.Value
## 1450678_at -13.11502 12.12619 -22.16088 3.972424e-17 9.
## 1434376_at -13.23014 12.32611 -21.96531 4.836924e-17 9.
## 1448752_at -13.72267 13.49913 -21.53819 7.477793e-17 9.
## 1418625_s_at -13.72472 12.98242 -21.25842 9.989881e-17 9.
## 1460218_at -12.86859 12.33489 -20.92354 1.419425e-16 9.
## 1449574_a_at -12.10868 11.08532 -20.89386 1.464659e-16 9.
```

```
resultats_vanco<- topTable(fit2, coef="Infectats_vs_Noinfected")
head(resultats_vanco)
```

```
##          logFC AveExpr      t      P.Value
## 1450678_at -12.93818 12.126189 -21.86206 5.370289e-17 8
## 1450753_at -11.44555 9.446085 -21.42778 8.379830e-17 8
## 1452954_at -13.27552 12.959169 -21.40992 8.536051e-17 8
## 1448752_at -13.49689 13.499127 -21.18382 1.079822e-16 8
## 1434376_at -12.73214 12.326108 -21.13851 1.132219e-16 8
## 1418625_s_at -13.63340 12.982418 -21.11696 1.158059e-16 8
```

```
write.csv(resultats_untreated, file.path(resultsDir, "resultats_untreated.csv"))
write.csv(resultats_line, file.path(resultsDir, "resultats_line.csv"))
write.csv(resultats_vanco, file.path(resultsDir, "resultats_vanco.csv"))
```

A les taules resultants, sobretot ens interessa fixar-nos en:

- La columna de logFC, on veurem l'efecte dels gens. Quan és negatiu podem interpretar que està infra expressat, i el contrari quan és positiu.
- La columna de p valor i p valor ajustat que ens ajudaran a veure quin gens tenen canvis significatius.

## ANOTACIÓ

A les taules generades, veiem que la primera columna és le gen però si no sabem res més, ens aporta poca informació realment ja que no en podem interpretar res amb aquell nom. Per tant, ara procedirem a fer el pas que s'anomena anotació. Ho farem en base a l'exemple de [https://aspteaching.github.io/Omics\\_Data\\_Analysis-Case\\_Study\\_1-Microarrays/Case\\_Study\\_1-Microarrays\\_Analysis.html#environment-preparation](https://aspteaching.github.io/Omics_Data_Analysis-Case_Study_1-Microarrays/Case_Study_1-Microarrays_Analysis.html#environment-preparation), tal i com hem anat mencionant.

Haurem de mirar a la pàgina de GEO la plataforma utilitzada per així saber quin array s'ha utilitzat i saber quina base de dades hem de fer servir pel procés d'anotació (això també es podria fer a través de R, si li demanem que ens retori l'objecte rawData veiem a ens diu en un dels apartats que Annotation: pd.mouse430.2). En aquest cas és [Mouse430\_2] Affymetrix Mouse Genome 430 2.0 Array.

```
library(annotate)
library(mouse4302.db)

annotatedTopTable <- function(topTab, anotPackage) {
  topTab <- cbind(PROBEID=rownames(topTab), topTab) #les eti
  myProbes <- rownames(topTab)
  thePackage <- eval(parse(text = anotPackage))
  geneAnots <- select(thePackage, myProbes, c("SYMBOL", "ENT
  annotatedTopTab <- merge(x=geneAnots, y=topTab, by.x="PROB
  return(annotatedTopTab)
}
```

```
annotated_untreated <- annotatedTopTable(resultats_untreated
                                         anotPackage="mouse4
head(annotated_untreated)
```



```
##          PROBEID  SYMBOL  ENTREZID
## 1  1415677_at  Dhrs1    52585      dehydrogenas
## 2  1415682_at   Xpo7    65246
## 3  1415703_at  Huwe1    59026  HECT, UBA and WWE domain
## 4  1415708_at   Tug1    544752      taurine upreg
## 5 1415716_a_at  Rps27    57294      ribosoma
## 6 1415716_a_at Rps27rt 100043813      ribosomal protein S
##          logFC  AveExpr      t    P.Value adj.P.Val
## 1  0.3080049  8.054020  0.9584020 0.3477295 0.9494925 -4.
## 2 -0.3225181 10.269323 -0.6964638 0.4930492 0.9494925 -4.
## 3 -0.2629589  8.223032 -0.8202812 0.4203937 0.9494925 -4.
## 4 -0.1906593  9.328072 -0.5861006 0.5634599 0.9494925 -4.
## 5 -0.0949502 12.795259 -0.2619349 0.7956803 0.9523177 -4.
## 6 -0.0949502 12.795259 -0.2619349 0.7956803 0.9523177 -4.
```

```
annotated_line <- annotatedTopTable(resultats_line,
                                     anotPackage="mouse4302.d
head(annotated_line)
```

```
##          PROBEID  SYMBOL  ENTREZID
## 1  1415677_at  Dhrs1    52585      dehydrogenas
## 2  1415682_at   Xpo7    65246
## 3  1415703_at  Huwe1    59026  HECT, UBA and WWE domain
## 4  1415708_at   Tug1    544752      taurine upreg
## 5 1415716_a_at  Rps27    57294      ribosoma
## 6 1415716_a_at Rps27rt 100043813      ribosomal protein S
##          logFC  AveExpr      t    P.Value  adj.P.Va
## 1 -9.179144  8.054020 -15.26716 1.324135e-13 1.949263e-1
## 2 -10.061860 10.269323 -11.61419 3.726509e-11 1.417393e-1
## 3 -7.270085  8.223032 -12.12217 1.578354e-11 7.063449e-1
## 4 -8.685557  9.328072 -14.27177 5.484386e-13 5.380307e-1
## 5 -11.736561 12.795259 -17.30630 9.026950e-15 3.607011e-1
## 6 -11.736561 12.795259 -17.30630 9.026950e-15 3.607011e-1
```

```
annotated_vanco <- annotatedTopTable(resultats_vanco,
                                     anotPackage="mouse4302.
```

```
head(annotated_vanco)
```

```
##          PROBEID  SYMBOL  ENTREZID
## 1  1415677_at    Dhrs1    52585
## 2  1415682_at    Xpo7    65246
## 3  1415703_at    Huwe1    59026 HECT, UBA and WWE domain
## 4  1415708_at    Tug1    544752 taurine upreg
## 5  1415716_a_at  Rps27    57294 ribosoma
## 6  1415716_a_at  Rps27rt 100043813 ribosomal protein S
##          logFC  AveExpr      t      P.Value  adj.P.Val
## 1  -8.645923  8.054020 -14.38028 4.679231e-13 5.013780e-1
## 2  -9.503992 10.269323 -10.97025 1.152033e-10 3.466860e-1
## 3  -7.381677  8.223032 -12.30824 1.159891e-11 5.462172e-1
## 4  -8.725414  9.328072 -14.33726 4.982675e-13 5.252946e-1
## 5 -12.448652 12.795259 -18.35632 2.513045e-15 2.380574e-1
## 6 -12.448652 12.795259 -18.35632 2.513045e-15 2.380574e-1
```

A continuació farem una representació gràfica dels nostres resultats. En aquest cas farem servir com a límits un p valor  $<0.05$  i  $\log FC > 1$ . Com bé es menciona en l'assignatura, no existeix un límit establert per fer aquest punt de tall, sinó que es basa en sentit comú i plausibilitat biològica. És per aquest motiu que farem servir aquest valor i  $\log FC$ .

```
library(ggplot2)
#per crear el gràfic, com que ho haurem de fer 3 vegades, ho

create_volcano <- function(resultats, titol) {
  resultats$Significant <- with(resultats,
    ifelse(adj.P.Val < 0.05 & abs(logFC) > 1, "Significatiu"
  ) #amb això analitzem cada element i determinem si és o no

  ggplot(resultats, aes(x = logFC, y = -log10(P.Value), color =
    ifelse(adj.P.Val < 0.05 & abs(logFC) > 1, "orange", "purple"))) +
    geom_point(alpha = 0.5, size = 1.5, shape=17) +
    scale_color_manual(values = c("orange", "purple")) +
    labs(
      title = titol,
      x = "logFC",
      y = "P valor transformat"
    ) +
    theme(
```

```

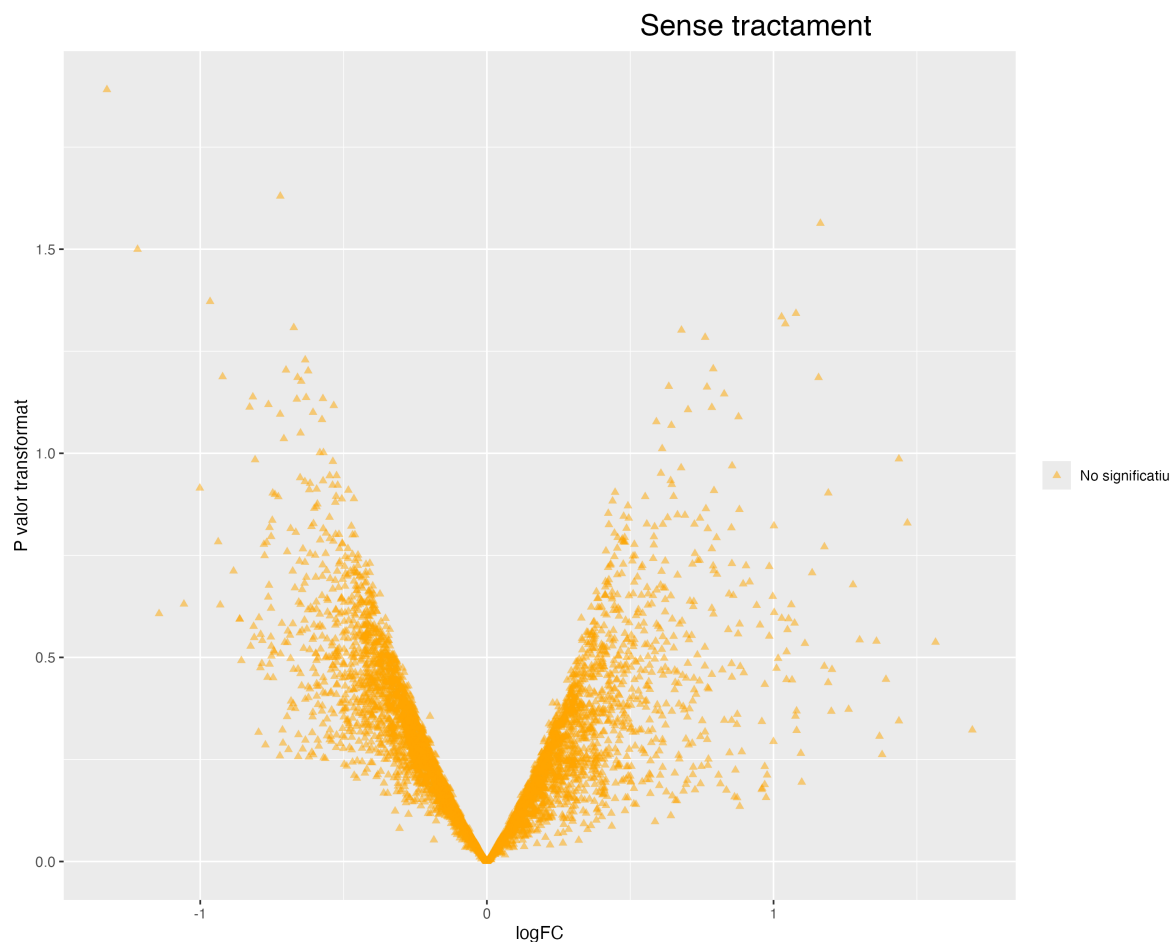
    plot.title = element_text(hjust = 0.8, size = 18),
    legend.title = element_blank()
  )
}

save_volcano <- function(plot, filename) {
  ggsave(
    filename = file.path(resultsDir, filename),
    plot = plot,
    width = 10, height = 8, dpi = 300
  )
}

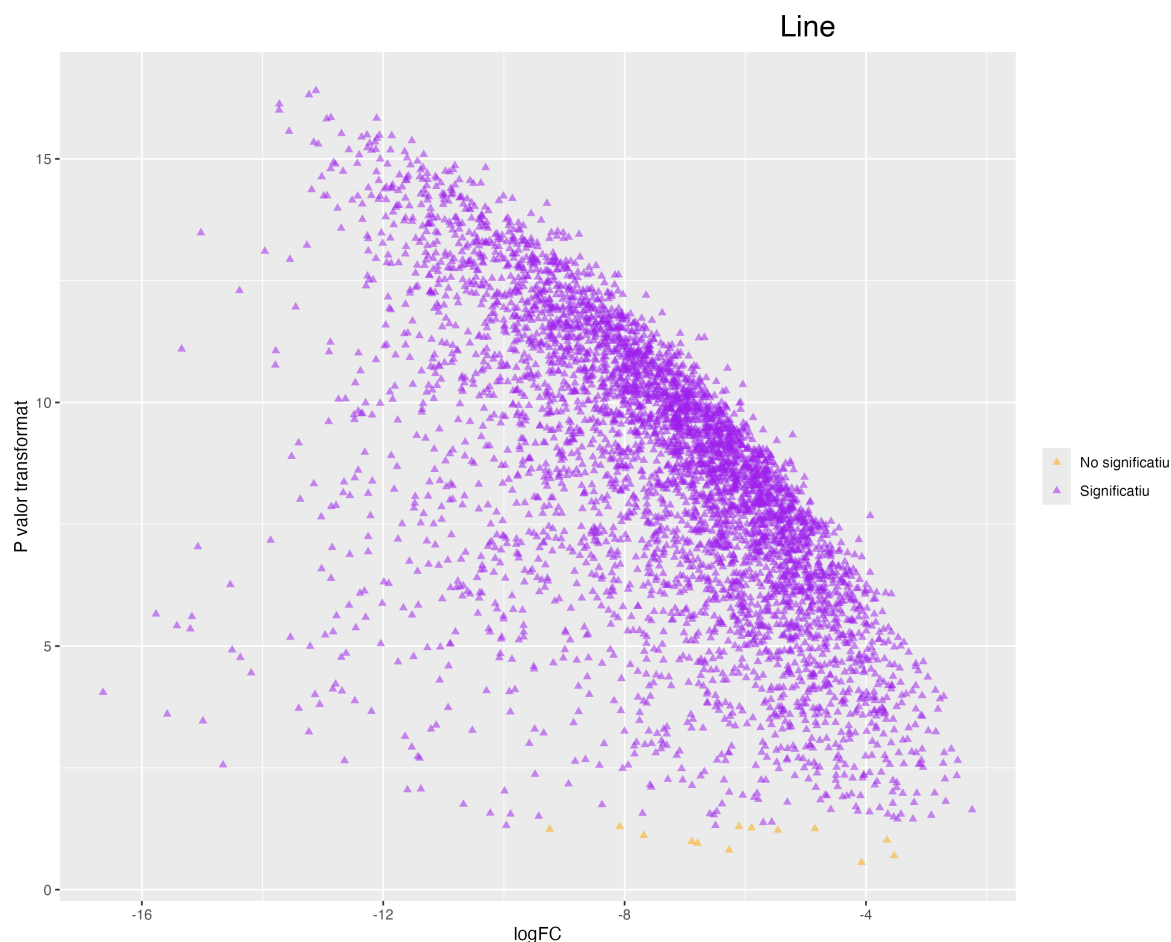
# Crear Volcano Plots per a cada comparació
volcano_untreated <- create_volcano(resultats_untreated, "Sense tractament")
volcano_line <- create_volcano(resultats_line, "Line")
volcano_vanco <- create_volcano(resultats_vanco, "Vanco")

save_volcano(volcano_untreated, "volcano_untreated.png")
save_volcano(volcano_line, "volcano_line.png")
save_volcano(volcano_vanco, "volcano_vanco.png")

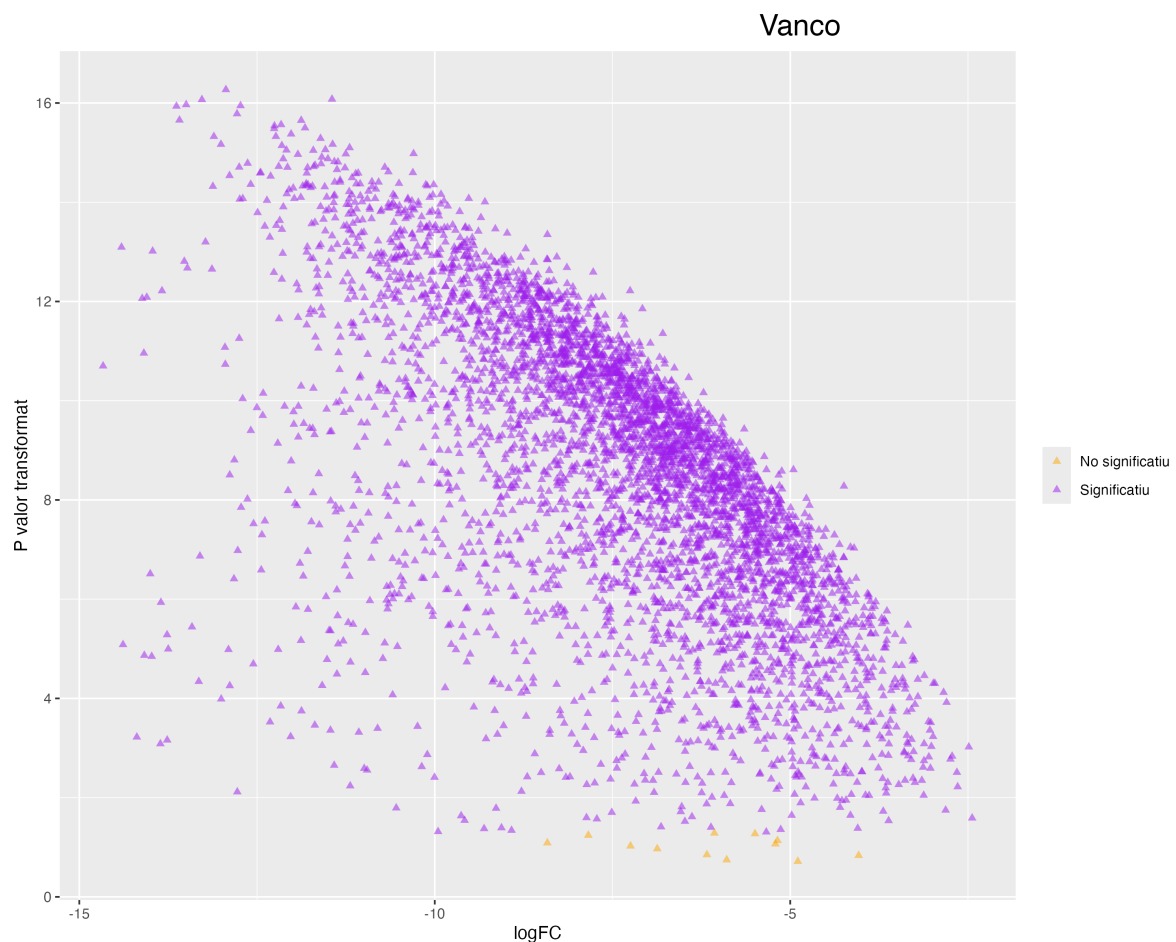
```



Al primer gràfic veiem que tots els punts són NO SIGNIFICATIUS. Per tant no hi ha hagut diferències singifcatives en l'expressió gènica entre els infectats i no infectats que no han rebut tractament.



En segon lloc tenim el gràfic del tractament amb linezolid: veiem a l'eix que les X que la gran majoria dels punts estan a per sota el 0 (són valors negatius). Com dèiem abans, això es tradueix amb gens que estan infra expressats en aquest cas. A l'eix Y veiem el p valor transformat. Tot i que hi ha una cua de valors que queda per sota el valor 5, la gran majoria queda per sobre, i es considerarien estadísticament significatius.



Per últim, la situació de la vancomicina és similar a la del linezolid, tot i que té un volum una mica més elevat de punts que no serien significatius.

A continuació compararem els gens diferencialment expressats entre les diferents situacions clíniques: els no tractat, i el que sí, amb linezolid i vancomicina respectivament. En aquest cas, agafem els gens que tinguin un p-valor associat  $<0.05$ . Crearem 3 objectes per emmagatzemar el llistat de gens que compleixin el criteri.

```
seleccio <- decideTests(
  fit2,
  method = "separate",
  adjust.method = "fdr",
  p.value = 0.05,
  lfc = 1
)

print(summary(seleccio))
```

```
##          Infectats_vs_Noinfectedats_Untreated Infectats_vs_No
## Down                                0
## NotSig                             4511
## Up                                0
##          Infectats_vs_Noinfectedats_Vancomycin
## Down                             4499
## NotSig                           12
## Up                                0
```

```
# Filtrar només els gens diferencialment expressats
sum.seleccio.rows <- apply(abs(seleccio), 1, sum)
seleccio.selected <- seleccio[sum.seleccio.rows != 0, ]
```

Ja tenim la taula resum comparant-nos els dos estats (infectat vs no infectat) de cada grup i com es comporten els gens implicats. Agafem per exemple el primer grup Infectats\_vs\_Noinfectedats\_Untreated: com veiem, igual que al gràfic anterior, no tenim cap gen que s'expressi de forma diferencial (tots són no significatius). En canvi en els grups d'antibiòtic, sorprenentment veiem que hi ha el mateix nombre de gens que estan diferencialment expressats (negativament en aquest cas), i és més, dels gens que hem seleccionat en tot el procés analític, únicament 13 (12 en el cas de la vancomicina) no són significatius.

## ANÀLISI DE SIGNIFICACIÓ BIOLÒGICA

A continuació fem l'anàlisi de significació biològica i ho farem a través de GO Gene Ontology. Primer agafarem els identificadors dels gens significatius, en aquest cas els ENTREZID.

```
# Seleccionem els ENTREZ ID
ID_untreated <- annotated_untreated$ENTREZID[annotated_untreated$adj.P.Val < 0.05]
ID_line <- annotated_line$ENTREZID[annotated_line$adj.P.Val < 0.05]
head(ID_line)
```

```
## [1] "52585" "65246" "59026" "544752" "5729"
```

```
ID_vanco <- annotated_vanco$ENTREZID[annotated_vanco$adj.P.V
head(ID_vanco)
```

```
## [1] "52585"      "65246"      "59026"      "544752"     "5729"
```

No fem l'anàlisi del grup sense tractament perquè ja hem vist que no té gens significatius.

```
library(clusterProfiler)
library(org.Mm.eg.db)

go_enrich_line <- enrichGO(
  gene = ID_line, #els gens significatius dels que han rebut
  OrgDb = org.Mm.eg.db,
  ont = "BP", #a triar entre BP, MF, CC.
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  readable = TRUE
)

go_enrich_vanco <- enrichGO(
  gene = ID_vanco, #els gens significatius dels que han rebu
  OrgDb = org.Mm.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.05,
  readable = TRUE
)
```

Fet l'anàlisi, en fem la representació gràfica amb un dotplot per visualitzar-ho. Crida MOLT l'atenció, que si més no els primers gens diferencialment expressats, són pràcticament idèntics en ambdós grups.

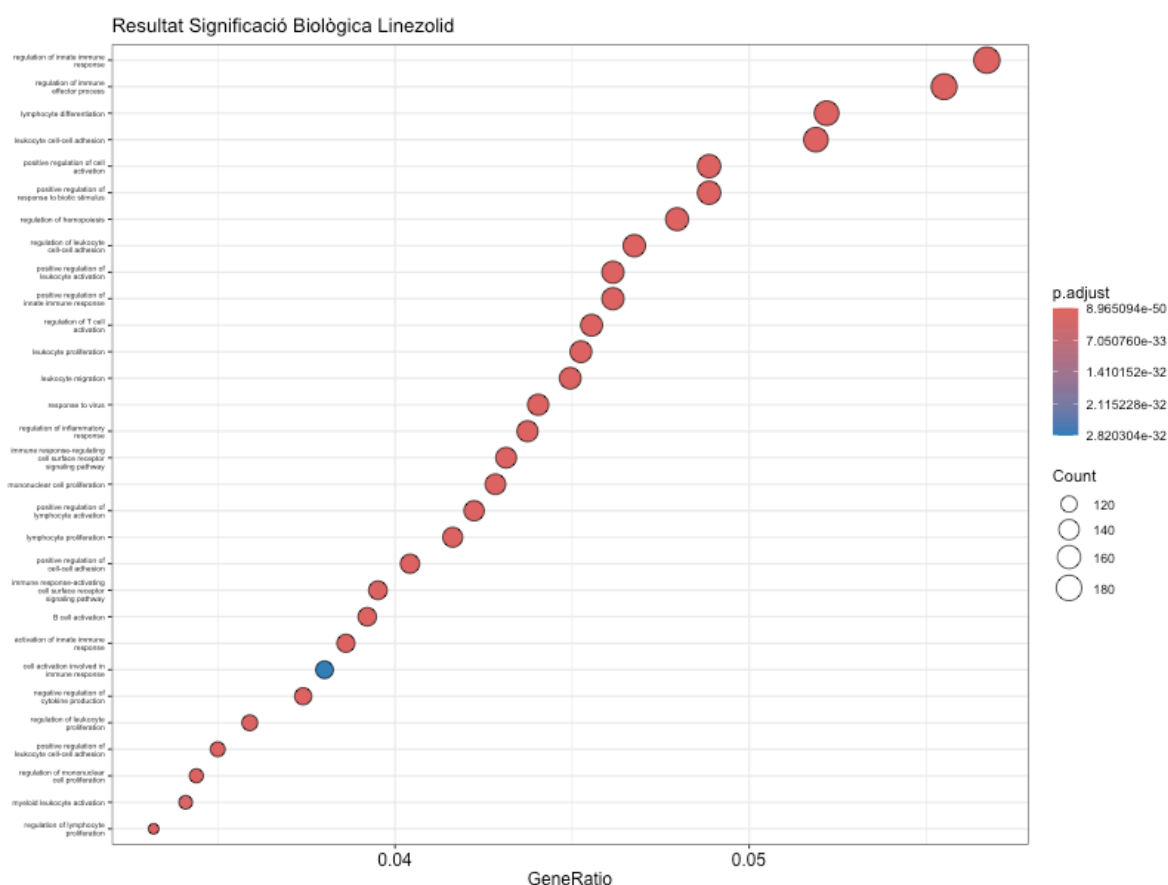
Alguna diferència hi ha (es comprova en el següent bloc de codi) entre els dos resultats, però sobretot el que veiem és que les vies que s'activen són les de resposta immune. Això és molt coherent ja que

estem treballant amb una infecció per MRSA i els seus tractaments de primera línia (vancomicina i linezolid).

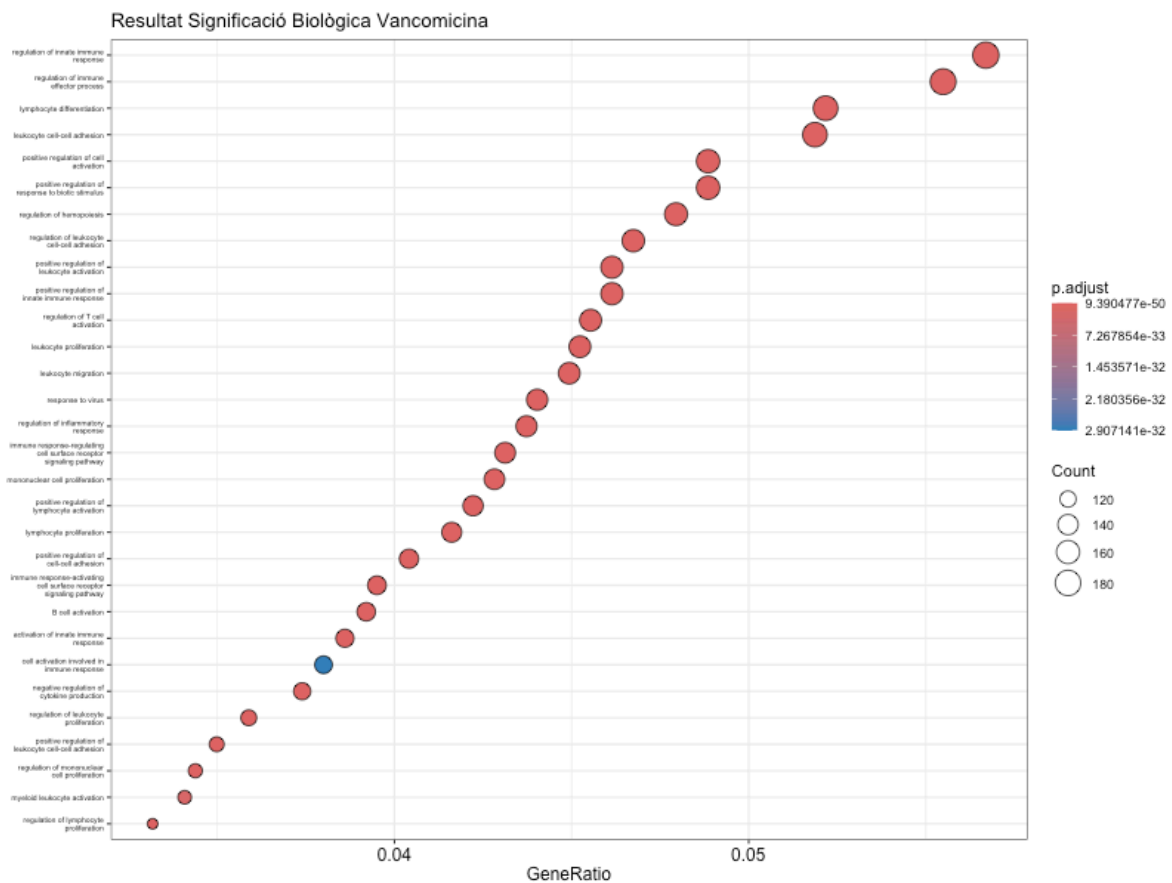
Són dos antibiòtics molt diferents, és a dir, no pertanyen a la mateixa família d'antibiòtics; la vancomicina és un glicopèptid i el linezolid una oxazolidinona. Tot i així, la resposta que generen a l'hoste és similar: actiavació de les vies immunes i de la proliferació cel·lular. En resum, tot encarat en lluitar contra una infecció.

```
output_file <- file.path(resultsDir, "dotplot_line.png")
png(file = output_file, width = 800, height = 600)
library(ggplot2)
dotplot(go_enrich_line, showCategory = 30, title = "Resultat

output_file <- file.path(resultsDir, "dotplot_vanco.png")
png(file = output_file, width = 800, height = 600)
dotplot(go_enrich_vanco, showCategory = 30, title = "Resultat
dev.off()
```







```
llista_vanco<- as.data.frame(go_enrich_vanco)
llista_line<- as.data.frame(go_enrich_line)
```

El bloc de codi anterior, es permet veure en una llista les diferents vies identificades en el procés d'anàlisi biològica. Majoritàriament són vies relacionades amb el sistema immunològic.

```
if (identical(llista_vanco, llista_line)) {
  print("Les dues llistes són idèntiques.")
} else {
  print("Les dues llistes NO són idèntiques.")
}
```

```
## [1] "Les dues llistes NO són idèntiques."
```