ANÁLISIS DE DATOS ÓMICOS - PEC2

Bruno Martin Farcic Melo

Contents

1 INTRODUCCIÓN **1**

2 OBJETIVOS **2**

3 MATERIALES Y MÉTODOS **2**

4 ANÁLISIS **3**

4.1 CREAR EL EXPRESSION SET 3

4.2 EXPLORACIÓN DE LOS DATOS Y CONTROL DE CALIDAD 4

4.2.1 ARRAY QUALIY METRICS 8

4.3 NORMALIZACIÓN Y FILTRAJE 9

4.4 ANÁLISIS DE EXPRESIÓN DIFERENCIAL 10

4.4.1 MATRICES DE DISEÑO Y CONTRASTE 10

4.4.2 ANOTACIÓN DE GENES 13

4.4.3 COMPARACIÓN MÚLTIPLE 14

4.5 PERFILES DE EXPRESIÓN 18

4.6 ANÁLISIS DE SIGNIFICANCIA BIOLÓGICA 20

4.6.1 GENE SET ENRICHMENT 23

5 CONCLUSIONES **24**

6 APÉNDICE **25**

6.1 CÓDIGO COMPLETO 25

1 INTRODUCCIÓN

Dentro del análisis de datos ómicos, una de las principales herramientas que nos podemos encontrar son los microarrays, de los cuales se puede extraer información sobre la expresión de múltiples genes a la vez mediante el uso de sondas y la intereptación de sus intensidades.

Este informe pretende dar una visión global simplificada de un proceso de análisis de expresión sobre un dataset obtenido del repositorio público Gene Expression Omnibus (GEO), donde a partir de un modelo de ratón, se pretende encontrar diferencias entre tres grupos de tratamiento de antibiótico, el primero control,

el segundo tratado con linezolid y el último con vancomicina, bajo condiciones de infección. Este trabajo incluye todas las etapas de un análisis de datos ómicos, partiendo de un preprocesado y normalización de datos, hasta obtener las vías más alteradas diferencialmente entre los grupos de estudio.

2 OBJETIVOS

El principal objetivo que tiene este informe es la familiarización con los diferentes paquetes de R que se utilizan en un estudio de estas caracterísiticas, así como la interpretación de resultados de los mismos.

Más dentro del estudio realizado, el objetivo principal es encontrar qué vías se puedne ver alteradas al someter a infección a grupos tratados con diversos antibióticos, o en otras palabras, la utilidad de los antibióticos linezolid y vancomicina para la inmunomodulación durante la infección bacteriana.

3 MATERIALES Y MÉTODOS

Para el estudio se ha utilizado un dataset que podemos encontrar en GEO siguiendo este enlace:

- GSE38531.

Resumiendo un poco los grupos de estudio este repositorio contiene 35 muestras de ratón divididos en tres grupos de tratamiento e infectados con una cepa de *Staphylococcus aureus* resistente a meticilina, concretamente la cepa USA300:

- **Grupo control (untreated)**: Este grupo consta de 5 muestras obtenidas a la hora 0 de la infección, 5 muestras obtenidas a las 2 horas de infección y 5 muestras obtenidas a las 24 horas de la infección.
- **Grupo linezolid (linezolid)**: Aquí se encuentran 5 muestras del momento de la infección (hora 0), y 5 muestras obtenidas 24 horas después.
- **Grupo vancomicina (vancomycin)**: Este último grupo contiene las mismas muestras que el grupo tratado con linezolid, es decir, 5 de la hora 0 de infección y 5 depués de 24 horas.

Para simplficiar el análisis primero se ha procedido para eliminar las 5 muestras pertenecientes al grupo control tomadas 2 horas después de la infección, eliminado una muestra de los grupos resultantes y aleatorizado las muestras de estudio. De esta manera, los grupos resultantes constan de 24 muestras divididas en 3 grupos de tratamiento de la siguiente manera:

- 4 muestras tomadas a la hora 0 de infección (sin infectar).
- 4 muestras tomadas 24 horas después de la infección (infectados).

Para el procedimiento de análsis se ha generado un Expression Set con las muestras crudas y se las hecho un análsiis para determinar si era necesario normalizar los datos o aplicar alguna transformación. El siguiente análisis se ha hecho utilizando R y paquetes que se pueden encontrar en Bioconductor. Las representaciones gráficas utilizan paquetes de R como ggplot2, pheatmap o dendextend para hacer más visuales los datos resultantes.

4 ANÁLISIS

4.1 CREAR EL EXPRESSION SET

Primero se ha aleatorizado las muestras obteniendo como resultado el siguiente grupo de trabajo:

Tabla 1: Selección de muestras después de eliminar y aleatorizar.

```
##              sample        infection    time      agent
## GSM944850 GSM944850 S. aureus USA300 hour 24  linezolid
## GSM944843 GSM944843 S. aureus USA300 hour 24  linezolid
## GSM944836 GSM944836 S. aureus USA300 hour 24  linezolid
## GSM944857 GSM944857 S. aureus USA300 hour 24  linezolid
## GSM944861 GSM944861       uninfected  hour 0  linezolid
## GSM944854 GSM944854       uninfected  hour 0  linezolid
## GSM944847 GSM944847       uninfected  hour 0  linezolid
## GSM944840 GSM944840       uninfected  hour 0  linezolid
## GSM944863 GSM944863 S. aureus USA300 hour 24  untreated
## GSM944835 GSM944835 S. aureus USA300 hour 24  untreated
## GSM944856 GSM944856 S. aureus USA300 hour 24  untreated
## GSM944849 GSM944849 S. aureus USA300 hour 24  untreated
## GSM944859 GSM944859       uninfected  hour 0  untreated
## GSM944831 GSM944831       uninfected  hour 0  untreated
## GSM944838 GSM944838       uninfected  hour 0  untreated
## GSM944852 GSM944852       uninfected  hour 0  untreated
## GSM944837 GSM944837 S. aureus USA300 hour 24 vancomycin
## GSM944851 GSM944851 S. aureus USA300 hour 24 vancomycin
## GSM944858 GSM944858 S. aureus USA300 hour 24 vancomycin
## GSM944844 GSM944844 S. aureus USA300 hour 24 vancomycin
## GSM944834 GSM944834       uninfected  hour 0 vancomycin
## GSM944841 GSM944841       uninfected  hour 0 vancomycin
## GSM944855 GSM944855       uninfected  hour 0 vancomycin
## GSM944848 GSM944848       uninfected  hour 0 vancomycin
```

Para generar el Expression Set, se ha descomprimido los archivos .CEL obtenidos de GEO en una carpeta, donde mediante el comando read.celfiles() del paquete de Bioconductor oligo podremos extraer la información pertenecientes a las intensidades de las sondas. De esta manera, el Expression Set queda así:

```
## Cargando paquete requerido: pd.mouse430.2

## Cargando paquete requerido: RSQLite

## Cargando paquete requerido: DBI

## Platform design info loaded.

## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944850_2564_6914_32316_24h-lin-3_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944843_2564_6914_32309_24h-lin-2_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944836_2564_6914_32302_24h-lin-1_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944857_2564_6914_32323_24h-lin-4_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944861_2564_6914_32327_Ctl-Lin-5_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944854_2564_6914_32320_Ctl-Lin-4_Mouse430+2
```

```
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944847_2564_6914_32313_Ctl-Lin-3_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944840_2564_6914_32306_Ctl-Lin-2_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944863_2564_6914_32329_24h-5_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944835_2564_6914_32301_24h-1_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944856_2564_6914_32322_24h-4_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944849_2564_6914_32315_24h-3_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944859_2564_6914_32325_Ctl-5_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944831_2564_6914_32297_Ctl-1_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944838_2564_6914_32304_Ctl-2_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944852_2564_6914_32318_Ctl-4_Mouse430+2.CEL
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944837_2564_6914_32303_24h-Van-1_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944851_2564_6914_32317_24h-Van-3_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944858_2564_6914_32324_24h-Van-4_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944844_2564_6914_32310_24h-Van-2_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944834_2564_6914_32300_Ctl-Van-1_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944841_2564_6914_32307_Ctl-Van-2_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944855_2564_6914_32321_Ctl-Van-4_Mouse430+2
## Reading in : C:/Users/Bruno/Desktop/PEC2/GES38531_RAW/GSM944848_2564_6914_32314_Ctl-Van-3_Mouse430+2
```

```
## ExpressionFeatureSet (storageMode: lockedEnvironment)
## assayData: 1004004 features, 24 samples
##   element names: exprs
## protocolData
##   rowNames: GSM944850 GSM944843 ... GSM944848 (24 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: GSM944850 GSM944843 ... GSM944848 (24 total)
##   varLabels: sample infection time agent
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.mouse430.2
```

```
## [1] TRUE
```

Este Expression Set de los datos crudos contiene 1004004 sondas para nuestras 24 muestras seleccionadas. Dentro de los fenodatos incluimos información como la muestra, su estado de infección y el antibiótico utilizando durante su estudio.

4.2 EXPLORACIÓN DE LOS DATOS Y CONTROL DE CALIDAD

Una vez obtenido el Expression Set de los datos crudos, se puede proceder al análisis de estos. La exploración de datos y control de calidad no pretenden mostrar cómo se distribuyen las muestras, si hay valores "outliers", si siguen una distribución normal, o cómo de diferentes son entre si los grupos.

Para facilitar la exploración, se hac reado una nueva columna en los fenodatos, que se llama "group", donde dividimos las muestras dentro de sus grupos, lo que nos facilitará darles un color para los gráficos

Tabla 2: Grupos de las muestras

Tratamiento	Tiempo de infección	Grupo	Color
Control	0h	untreated_uninfected	Azul claro
Control	24h	untreated_infected	Azul
Linezolid	0h	linezolid_uninfected	Verde claro
Linezolid	24h	linezolid_infected	Verde oscuro
Vancomycin	0h	vancomycin_uninfected	Salmón
Vancomycin	24h	vancomycin_infected	Rojo

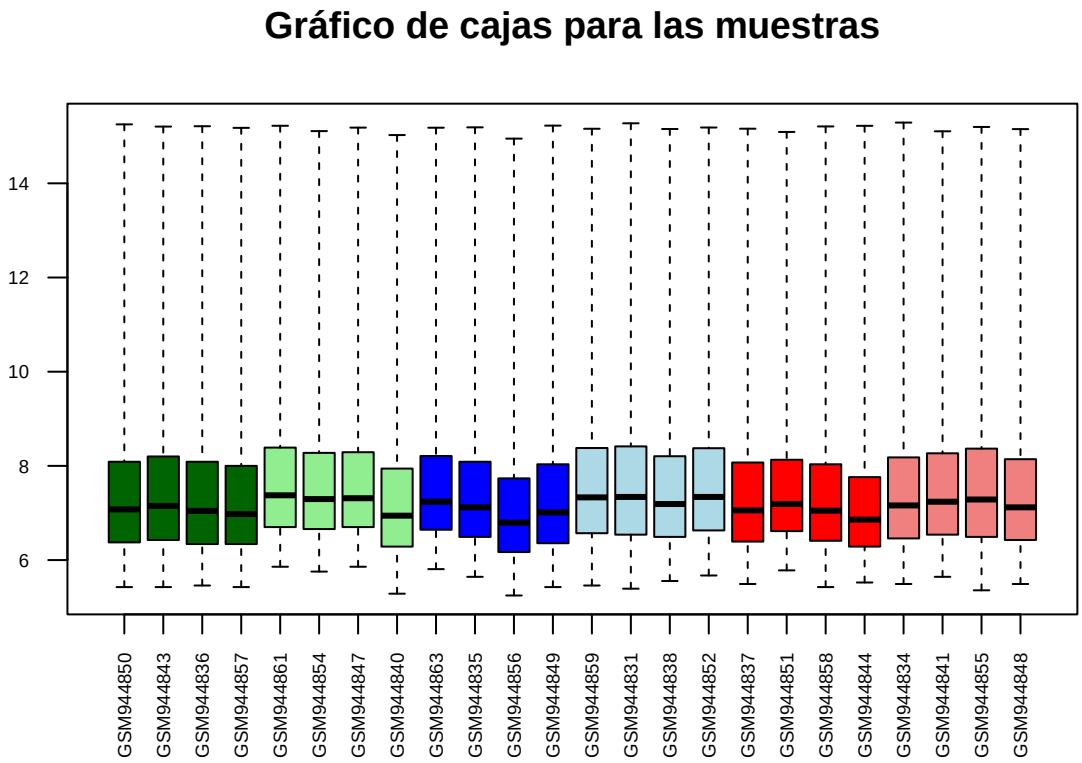

Gráfico 1: Gráfico de cajas para las muestras antes del normalizado.

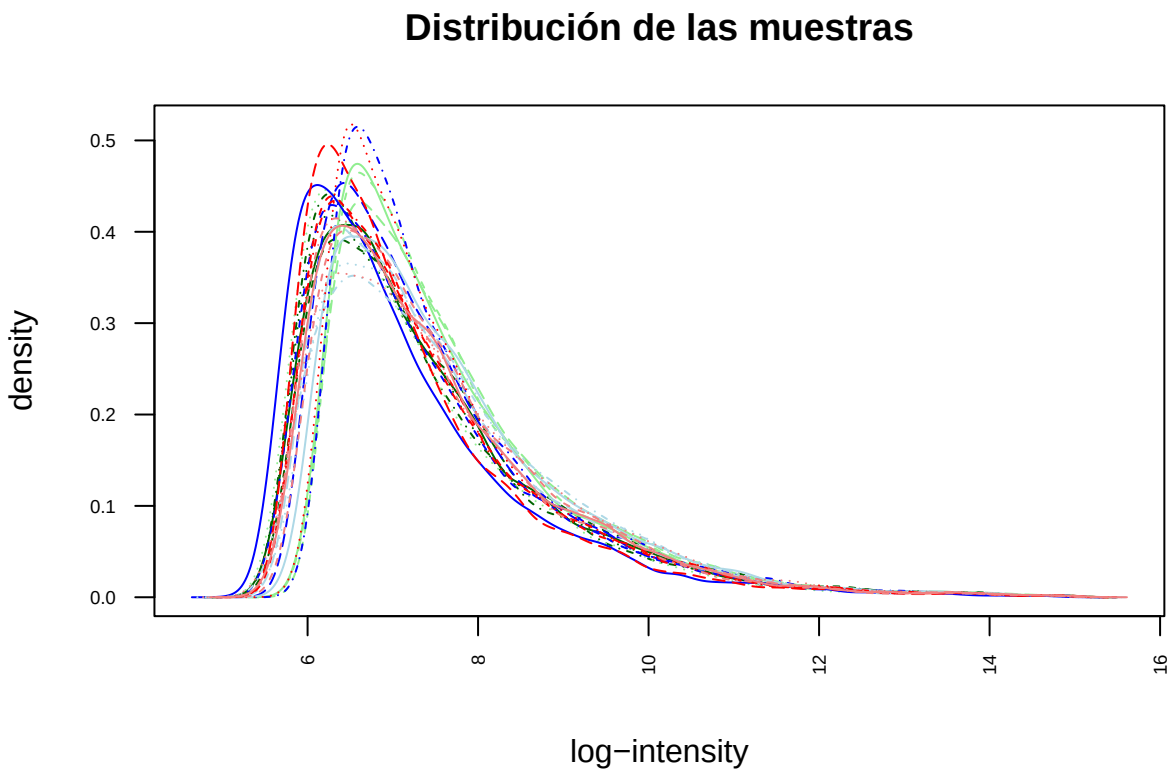

Gráfico 2: Gráfico de densidad para las muestras antes del normalizado.

Con el gráfico de cajas y el de densidad se puede observar que las diversas muestras tienen similitudes en la distribución aunque no están centradas, lo que nos sugiere que un centrado sería necesario para poder llevar a cabo un análsiis preciso.

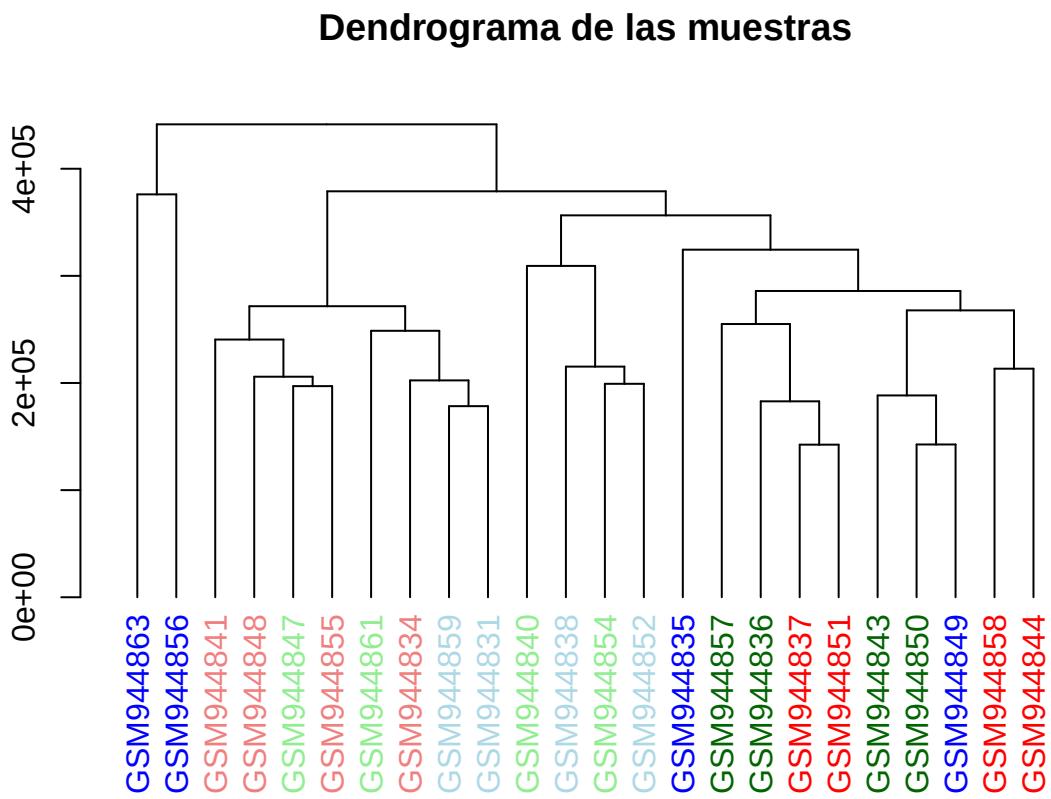

Gráfico 3: Dendograma de las muestrras.

El dendograma o "cluster" es un gráfico que permite identificar agrupaciones gracias a la similitud de sus perfiles. Las distancias entre las muestras indican qué tan similares o separados están, por lo que cabría esperar que los grupos sin infectar e infecatados se vieran claramente separados.

En este caso, se observa cómo dejando de lado dos muestras, los grupos son homogéneos, ya que se diferencian los grupos mencionados.

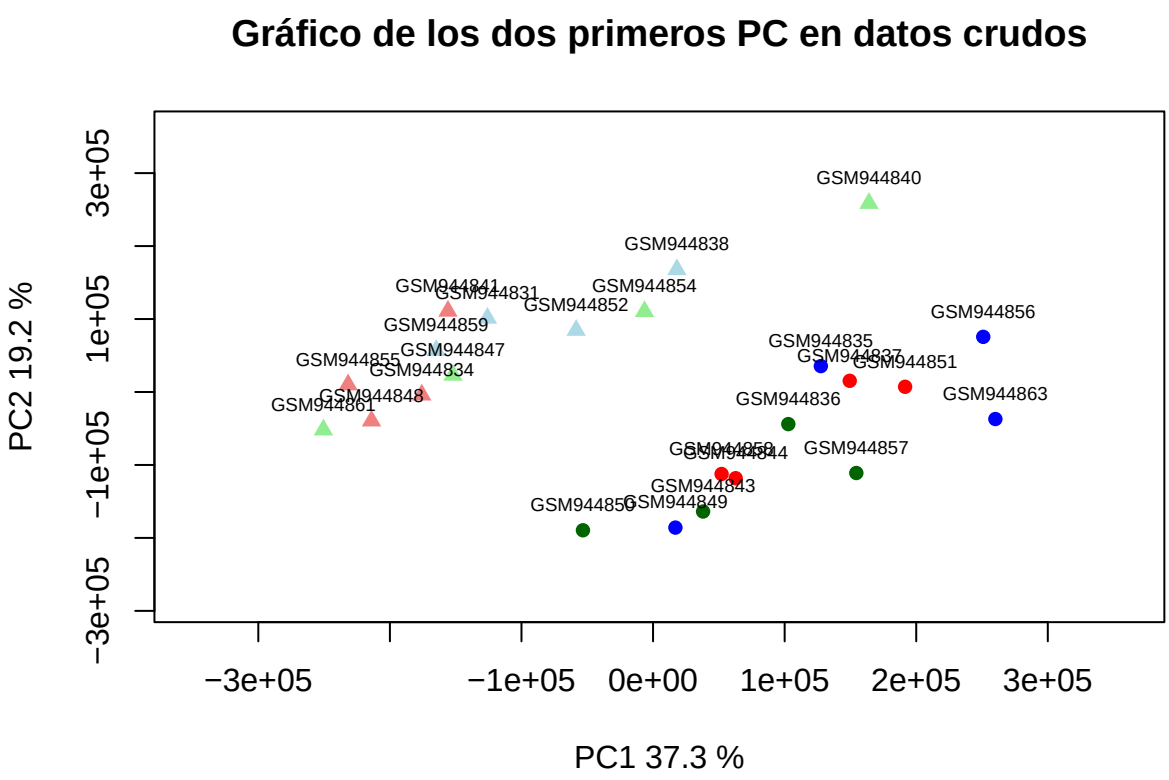

Gráfico 4: Análisis de los dos primeros componentes principales (PCA).

Un PCA o análsis de los componentes principales es un gráfico que de nuevo agrupa a las muestras según los componentes de la variabilidad. En este caso, vemos que el primer componente constituye el 37,3% de la variabilidad y el segundo el 19,2%, haciendo un total del 56,5%, o lo que viene a ser en otras palabras, que en estos dos componentes explicamos más de la mitad de la variabildad entre las muestras.

Lo primero que llama la atención es que se han agrupado perfectamente los grupos inectados y sin infectar a ambos lados de una diagonal. Esto indica que las muestras están bien separadas, y que con un solo componente no se puede explicar esta separación, ya que no hay una línea divisoria vertical u horizontal, por lo que se necesita como mínimo estos dos componentes para explicar dicha separración.

4.2.1 ARRAY QUALIY METRICS

Array quality metrics es un paquete de Bioconductor que nos lleva a cabo el control de calidad. Por si solo es un análisis importante, pero juntos a los gráficos que se han visto hasta ahora aporta relativamente poca información, ya que gran parte del análsisi que lleva a cabo viene dado por los gráficos que hemos visto. De todas maneras, hace estudios extra, como por ejemplo el estudio de muestras “outliers”, que en este estudio se destaca que la muestra número 11, correspondiente al ID GSM944856 lo es, por lo que eliminarlo podría ser una buena práctica para conseguir más precisión en el estudio.

El archivo HTML generado se puede consultar en el siguiente enlance:

- Array Quality Metrics

Para finalizar, el análsiis exploratorio de los datos crudos revela principalmente que las muestras de estudio siguen una distribución que debe ser normalizada, con semejanzas entre los grupos problema como se ha

observado en el dendgrama y PCA, determinando que la variabildad más grande es debida a la infección de los ratones.

4.3 NORMALIZACIÓN Y FILTRAJE

Como se ha determinado en el apartado de exploración de datos, una normalización sería conveniente para que nuestros datos se centraran y siguieran una distribución más normal. Para ello, se ha utilizado la función rma() del paquete oligo de Bioconductor.

RMA son las siglas de (Robust Multiarray Average), un método ampliamente utilizado gracias a sus capacidades y reproducibilidad y bastante robusto frente a "outliers", que ayudará a reducir el ruido de fondo y normalizar las muestras. Además, gracias a su implementación en Bioconductor, el objeto resultado de esta función es un Expression Set igual que el que teníamos con los valores dentro del apartado assayData normalizado.

```
## Background correcting
## Normalizing
## Calculating Expression
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 45101 features, 24 samples
##   element names: exprs
## protocolData
##   rowNames: GSM944850 GSM944843 ... GSM944848 (24 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: GSM944850 GSM944843 ... GSM944848 (24 total)
##   varLabels: sample infection ... group (5 total)
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.mouse430.2
```

Lo primero que se observa de este nuevo Expression Set normalizado, es que hemos pasado de tener 1004004 "features" a 45101. Esto se debe a que previamente el 1004004 se refería a las sondas utilizadas, pero, tras la normalización, rma() une las sondas que midan el mismo gen, por lo que terminamos con 45101 "features" que se refiere a los genes de estudio.

Como paso final se ha realizado un filtrado. El filtraje es una herramienta poderosa para discriminar resultados. En este caso, se ha decidido estudiar únicamente el 10% de los genes que tengan más variabilidad.

```
## $eset
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 2048 features, 24 samples
##   element names: exprs
## protocolData
##   rowNames: GSM944850 GSM944843 ... GSM944848 (24 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: GSM944850 GSM944843 ... GSM944848 (24 total)
##   varLabels: sample infection ... group (5 total)
```

```
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: mouse4302
##
## $filter.log
## $filter.log$numDupsRemoved
## [1] 16955
##
## $filter.log$numLowVar
## [1] 18428
##
## $filter.log$numRemoved.ENTREZID
## [1] 7657
##
## $filter.log$feature.exclude
## [1] 13

## NULL
```

Gracias a la utilización de la función nsFilter() perteneciente al paquete Gene Filter de Bioconductor se puede determinar las condiciones de filtraje, pero cabe recordar que el objeto resultante es de tipo lista, donde se guardan los datos del nuevo Expression Set filtrado y los resultados de los genes eliminados.

En las condiciones propuestas, observamos que se han eliminado los siguientes genes:

- **Genes duplicados**: 16955
- **Genes de baja variabilidad**:18428, es decir, no cumplen los requisitos del 10%
- **Genes sin identificador ENTREZ**: 7657 por no tener identificador ENTREZ. Esto es importante para la posterior anotación, ya que estos identificadores ayudan a saber a qué gen se asocia cada sonda.

Para visualizar las diferencias entre los datos crudos y los filtrados normalizados se podría hacer de nuevo una exploración de esos datos, donde se determina si el normalizado ha resultado existoso gracias a la distribución y si el ruido de fondo se ha visto reducido.

4.4 ANÁLISIS DE EXPRESIÓN DIFERENCIAL

El siguiente paso del estudio es obtener qué genes se expresna diferencialmente entre los grupos. Para ello, se debe construir primeramente la matriz de diseño y las matrices de contraste, posteriormente anotar los genes para finalizar con un análisis de enriquecimiento.

4.4.1 MATRICES DE DISEÑO Y CONTRASTE

Las matrices de diseño y contraste son las encargadas de generar el modelo lineal que representa las condiciones experimentales.

- **Matriz de diseño**: Establece la relación entre las muestras dentro de los grupos, es decir, asigna cada muestra a su grupo dependiendo en este caso del tratamiento aplicado y el estado de infección o no infección.

- **Matriz de contraste**: Esta matriz describe las comparaciones entre los grupos que nosotros le indiquemos..

Tabla 3: Matriz de diseño.

	linezolid_infected	linezolid_uninfected	untreated_infected	untreated_uninfected	vancomycin_infected	vancomycin_uninfected
1	1	0	0	0	0	0
2	1	0	0	0	0	0
3	1	0	0	0	0	0
4	1	0	0	0	0	0
5	0	1	0	0	0	0
6	0	1	0	0	0	0
7	0	1	0	0	0	0
8	0	1	0	0	0	0
9	0	0	1	0	0	0
10	0	0	1	0	0	0
11	0	0	1	0	0	0
12	0	0	1	0	0	0
13	0	0	0	1	0	0
14	0	0	0	1	0	0
15	0	0	0	1	0	0
16	0	0	0	1	0	0
17	0	0	0	0	1	0
18	0	0	0	0	1	0
19	0	0	0	0	1	0
20	0	0	0	0	1	0
21	0	0	0	0	0	1
22	0	0	0	0	0	1
23	0	0	0	0	0	1
24	0	0	0	0	0	0

```
## 24                    0                    0                    1
## attr(,"assign")
## [1] 1 1 1 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$`pData(datos_norm)$group`
## [1] "contr.treatment"
```

Como se observa, la matriz de diseño numera 1 o 0 dependiendo si se pertenece al grupo problema o no, y los grupos son los definidos previamente.

Tabla 4: Matriz de contrastes.

```
##                       Contrasts
## Levels                 untreated_infected_vs_uninfected
##   linezolid_infected                                  0
##   linezolid_uninfected                                0
##   untreated_infected                                  1
##   untreated_uninfected                               -1
##   vancomycin_infected                                 0
##   vancomycin_uninfected                               0
##                       Contrasts
## Levels                 linezolid_infected_vs_uninfected
##   linezolid_infected                                  1
##   linezolid_uninfected                               -1
##   untreated_infected                                  0
##   untreated_uninfected                                0
##   vancomycin_infected                                 0
##   vancomycin_uninfected                               0
##                       Contrasts
## Levels                 vancomycin_infected_vs_uninfected
##   linezolid_infected                                   0
##   linezolid_uninfected                                 0
##   untreated_infected                                   0
##   untreated_uninfected                                 0
##   vancomycin_infected                                  1
##   vancomycin_uninfected                               -1
```

Las matrices de contrastes aplican 0 si el grupo no entra en el contraste y 1 y -1 a los grupos que se enfrentan para comparar.

Para proceder con las comparaciones de las matrices de contrastes primero se deben crear el modelo lineal. lmFit() es una función del paquete limma de Bioconductor que lleva a cabo estas comparaciones y nos devuelve una tabla con los p-valores y log fold change que nosotros le indiquemos. Para este estudio se ha elegido un p-valor de 0.05 y log fold change de 2, esto sigfnica que devolverá genes con una significancia del 95% y cuyas tasas de variación sean de 4 veces, tanto sobreexpresados como por debajo.

Tabla 5: Primeros resultados de la top table para el grupo control.

```
##                 logFC   AveExpr        t      P.Value    adj.P.Val        B
## 1427747_a_at 6.376484 10.743356 22.54624 2.961703e-16 6.065569e-13 27.14222
## 1421262_at   7.308103  7.093060 19.95626 3.485765e-15 3.569423e-12 24.81830
## 1422953_at   3.901662 11.244048 17.77174 3.532025e-14 2.411196e-11 22.59246
## 1417290_at   4.835270 10.111054 16.57946 1.393132e-13 7.132834e-11 21.25699
## 1418722_at   5.904350 11.000080 15.33670 6.396561e-13 2.511037e-10 19.76140
## 1419681_a_at 5.269464  7.469765 15.22676 7.356553e-13 2.511037e-10 19.62361
```

```
## Genes diferenciales para el grupo control (infectado vs sin infectar): 245
```

Definidos los valores, se observa la tabla, donde primero se nos idndica el gen, posteriormente sus estadísticos junto al p-valor ajustado.

Tabla 6: Primeros resultados de la top table para el grupo tratado con linezolid.

```
##                 logFC   AveExpr        t      P.Value    adj.P.Val        B
## 1421262_at   6.284683  7.093060 17.16160 7.054963e-14 9.667405e-11 21.78274
## 1427747_a_at 4.782518 10.743356 16.91023 9.440826e-14 9.667405e-11 21.50802
## 1419681_a_at 5.113203  7.469765 14.77522 1.318318e-12 8.999721e-10 18.99194
## 1422953_at   2.841899 11.244048 12.94461 1.648843e-11 7.020602e-09 16.53830
## 1417290_at   3.767437 10.111054 12.91801 1.714014e-11 7.020602e-09 16.50038
## 1440865_at   3.609903 11.119953 12.74070 2.223043e-11 7.587988e-09 16.24578

## Genes diferenciales para linezolid (infectado vs sin infectar): 166
```

Tabla 7: Primeros resultados de la top table para el grupo tratado con vancomicina.

```
##                 logFC   AveExpr        t      P.Value    adj.P.Val        B
## 1421262_at   6.734936  7.093060 18.39111 1.787140e-14 3.660064e-11 23.12934
## 1427747_a_at 4.913839 10.743356 17.37456 5.528078e-14 5.660752e-11 22.06267
## 1419681_a_at 5.323759  7.469765 15.38365 6.027276e-13 4.114621e-10 19.77403
## 1419709_at   5.431222  6.958553 13.45209 7.963151e-12 4.077133e-09 17.26066
## 1418722_at   5.055732 11.000080 13.13239 1.256290e-11 5.145765e-09 16.81302
## 1440865_at   3.570470 11.119953 12.60153 2.731833e-11 7.992562e-09 16.04810

## Genes diferenciales para vancomycin (infectado vs sin infectar): 134
```

4.4.2 ANOTACIÓN DE GENES

Siguiendo el flujo de trabajo, debemos anotar los genes, es decir, ponerles los identificadores ENTREZ o cualquier otro identificador para poder reconocer los genes. Este paso es importante para poder identificar cada gen con su importancia biológica, vía metabólica o relevancia dentro del organismo. En este estudio se ha optado por el identificador ENTREZ, y su símbolo, aunque el identificador ENSEMBL también habría sido una buena elecci

```
## 'select()' returned 1:many mapping between keys and columns
```

Tabla 8: Top table anotada para el grupo control.

```
##       probeIds ENTREZID SYMBOL    logFC   AveExpr        t      P.Value
## 1 1427747_a_at    16819   Lcn2 6.376484 10.743356 22.54624 2.961703e-16
## 2   1421262_at    16891   Lipg 7.308103  7.093060 19.95626 3.485765e-15
## 3   1422953_at    14289   Fpr2 3.901662 11.244048 17.77174 3.532025e-14
## 4   1417290_at    76905   Lrg1 4.835270 10.111054 16.57946 1.393132e-13
## 5   1418722_at    18054    Ngp 5.904350 11.000080 15.33670 6.396561e-13
## 6 1419681_a_at    50501  Prok2 5.269464  7.469765 15.22676 7.356553e-13
##      adj.P.Val        B
## 1 6.065569e-13 27.14222
## 2 3.569423e-12 24.81830
## 3 2.411196e-11 22.59246
## 4 7.132834e-11 21.25699
## 5 2.511037e-10 19.76140
## 6 2.511037e-10 19.62361
```

Tabla 9: Top table anotada para el grupo tratado con linezolid.

```
##       probeIds ENTREZID SYMBOL    logFC   AveExpr        t      P.Value
## 1   1421262_at    16891   Lipg 6.284683  7.093060 17.16160 7.054963e-14
## 2 1427747_a_at    16819   Lcn2 4.782518 10.743356 16.91023 9.440826e-14
## 3 1419681_a_at    50501  Prok2 5.113203  7.469765 14.77522 1.318318e-12
## 4   1422953_at    14289   Fpr2 2.841899 11.244048 12.94461 1.648843e-11
## 5   1417290_at    76905   Lrg1 3.767437 10.111054 12.91801 1.714014e-11
## 6   1440865_at   213002 Ifitm6 3.609903 11.119953 12.74070 2.223043e-11
##      adj.P.Val        B
## 1 9.667405e-11 21.78274
## 2 9.667405e-11 21.50802
## 3 8.999721e-10 18.99194
## 4 7.020602e-09 16.53830
## 5 7.020602e-09 16.50038
## 6 7.587988e-09 16.24578
```

Tabla 10: Top table anotada para el grupo tratado con vancomicina.

```
##       probeIds ENTREZID SYMBOL    logFC   AveExpr        t      P.Value
## 1   1421262_at    16891   Lipg 6.734936  7.093060 18.39111 1.787140e-14
## 2 1427747_a_at    16819   Lcn2 4.913839 10.743356 17.37456 5.528078e-14
## 3 1419681_a_at    50501  Prok2 5.323759  7.469765 15.38365 6.027276e-13
## 4   1419709_at    20863  Stfa3 5.431222  6.958553 13.45209 7.963151e-12
## 5   1418722_at    18054    Ngp 5.055732 11.000080 13.13239 1.256290e-11
## 6   1440865_at   213002 Ifitm6 3.570470 11.119953 12.60153 2.731833e-11
##      adj.P.Val        B
## 1 3.660064e-11 23.12934
## 2 5.660752e-11 22.06267
## 3 4.114621e-10 19.77403
## 4 4.077133e-09 17.26066
## 5 5.145765e-09 16.81302
## 6 7.992562e-09 16.04810
```

4.4.3 COMPARACIÓN MÚLTIPLE

Otra manera para determinar lo que se observa gracias al modelo lineal, es mediante la función decideTests() del paquete limma. El resultado es una matriz de decisiones que representa cada fila un gen y cada columna un contraste.

Tabla 11: Matriz de decisiones.

```
##        untreated_infected_vs_uninfected linezolid_infected_vs_uninfected
## Down                                 86                               34
## NotSig                             1803                             1882
## Up                                  159                              132
##        vancomycin_infected_vs_uninfected
## Down                                  81
## NotSig                              1914
## Up                                    53
```

De esta tabla se interpreta el número de genes para cada condición que se encuentra sobreexpresado, subexpresado o los genes que no son significativamente diferentes en las condiciones propuesta de log fold change de 2.

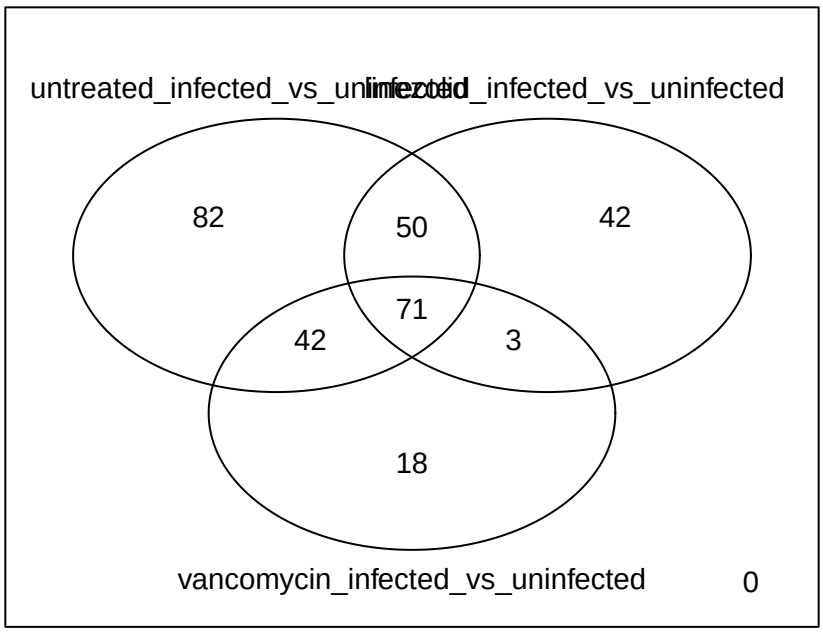

Gráfico 5: Diagrama de Venn.

Finalmente el diagrama de venn es una representación visual de los genes diferenciales y de las condiciones.

Se observan los siguientes datos:

- **Grupo control**
  - 82 genes expresados diferencialmente solo para este grupo.
  - 50 genes expresados diferencialmente compartidos con el grupo linezolid.
  - 42 genes expresados diferencialmente compartidos con el grupo vancomicina.
  - 71 genes expresados diferencialmente compartidos con los grupos linezolid y vancomicina.
- **Grupo linezolid**
  - 42 genes expresados diferencialmente solo para este grupo.
  - 3 genes compartidos con el grupo vancomicina.
- **Grupo vancomicina**
  - 18 genes expresados diferencialmente solo para este grupo.

Ahora, se puede realizar volcano plots, o gráficos de volcán para tener una representación mucho más visual de lo que se veía en el diagrama de Venn, es decir, esos genes que se comparten entre condiciones. Un volcano plot se intrepreta de manera que los puntos más altos y cerca de las esquinas son los que se expresan de manera más diferencial. En esta ocasión, además de señalizar los genes con su símbolo, se añaden las líneas correspondientes al log fold change 2 y -2, para que se visualice aquellos genes que cumplen esta condición

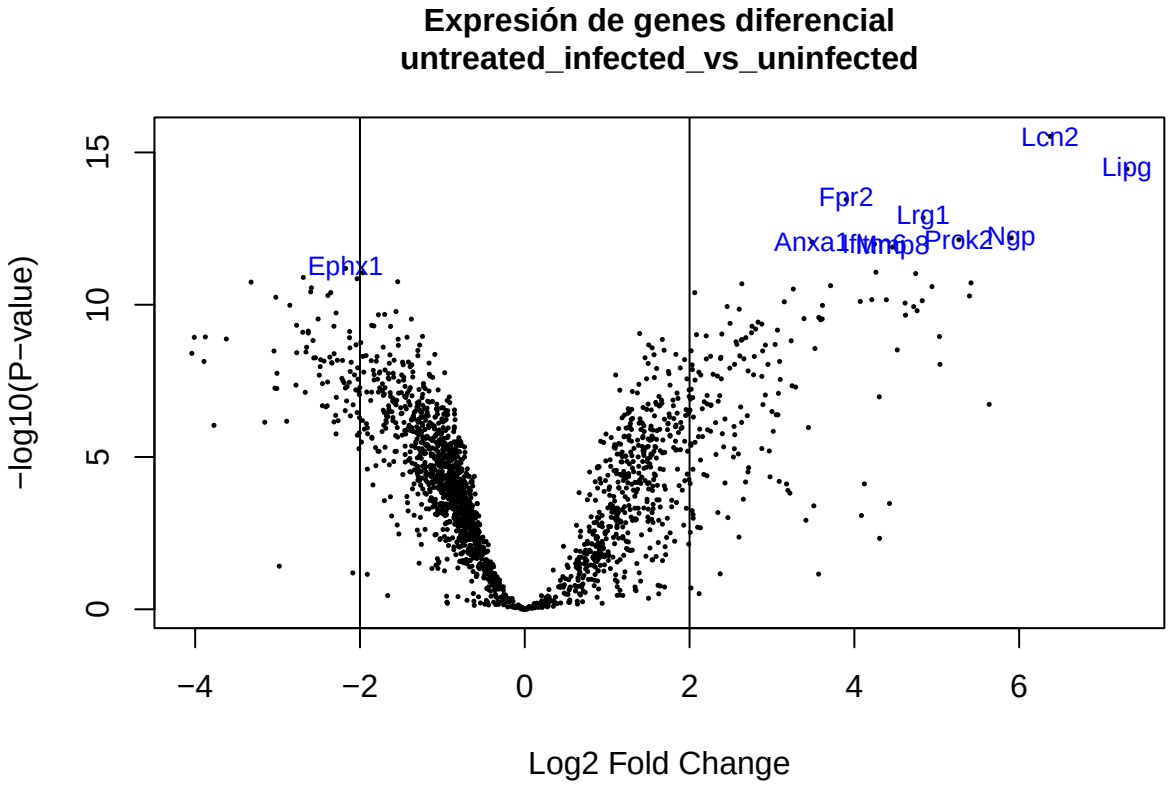

Gráfico 6: Volcano plot para el grupo control.

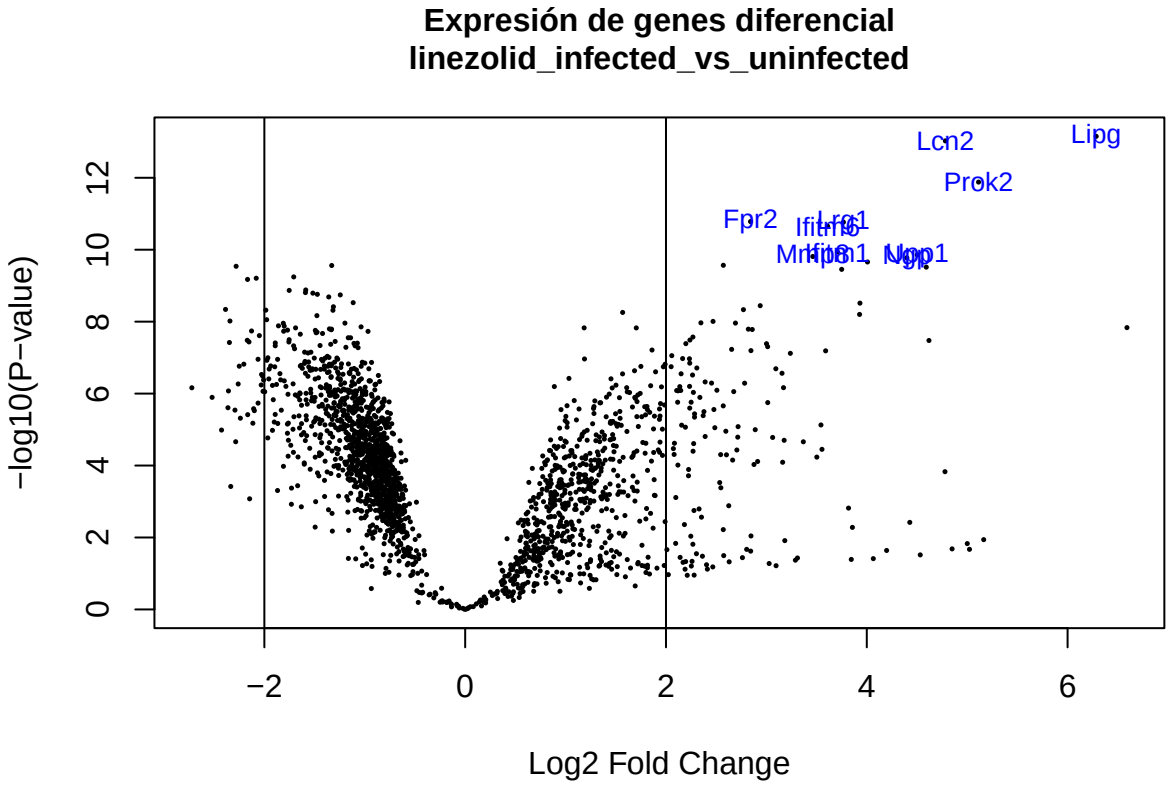

Gráfico 7: Volcano plot para el grupo linezolid.

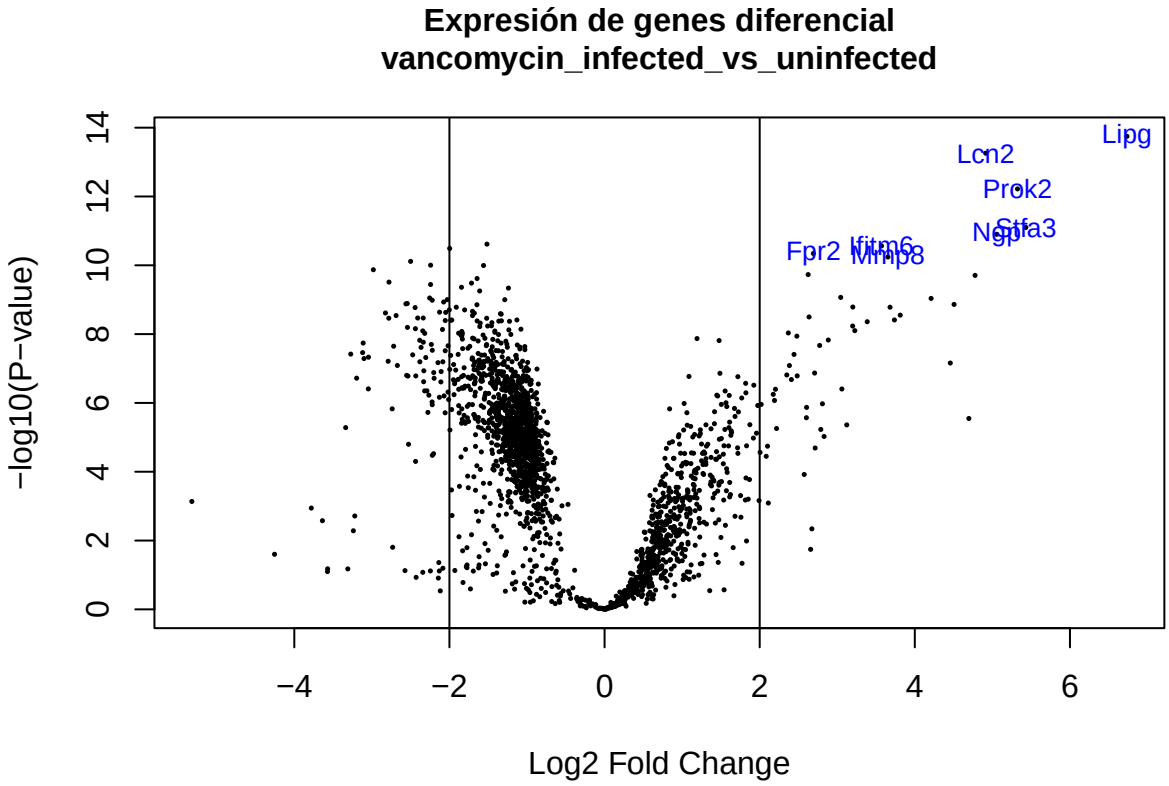

Gráfico 8: Volcano plot para el grupo vancomicina.

4.5 PERFILES DE EXPRESIÓN

Para visualizar los perfiles de expresión, lo más útil resulta un mapa de calor o heatmap.En estos se hace un cluster tanto para muestras como para los genes de estudio, y se colorean las celdas de tal manera que se representa la sobreexpresión como la subexpresión.

```
## 'select()' returned 1:1 mapping between keys and columns
```

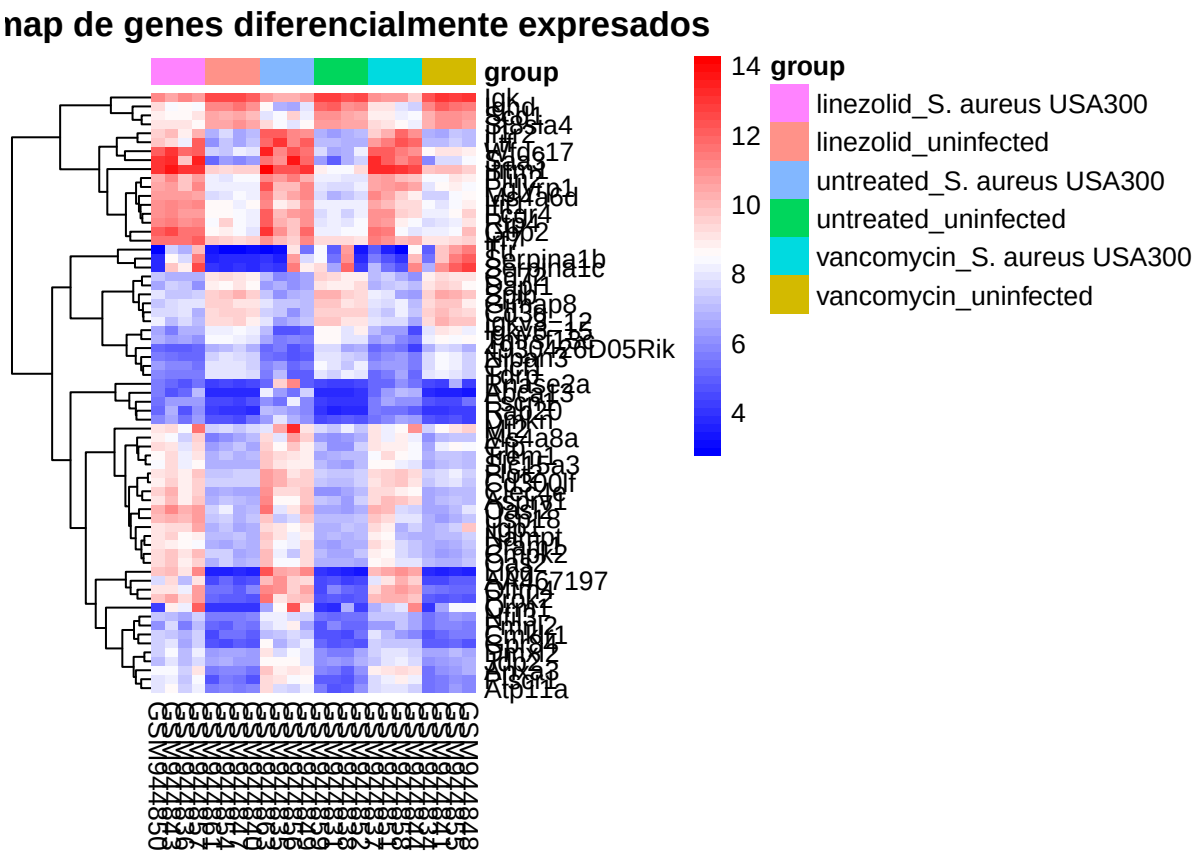

Gráfico 9: Heatmap para los genes expresados de manera diferencial.

El heatmap resultante nos muestra para los diferentes grupos experimentales los patrones de expresión. Como se puede observar, los genes diferenciales cambian sus patrones de expresión entre grupos no infectados a infectados, aunque las diferencias dentro de los grupos puede ser más complicada de ver. Para ello, se puede llevar a cabo otro heatmap, el cual realiza un cluster de las muestras para ver las agrupaciones que surgen de las expresiones, o lo que sería lo mismo, identificar esas muestras que comparten genes como veíamos en el diagrama de Venn.

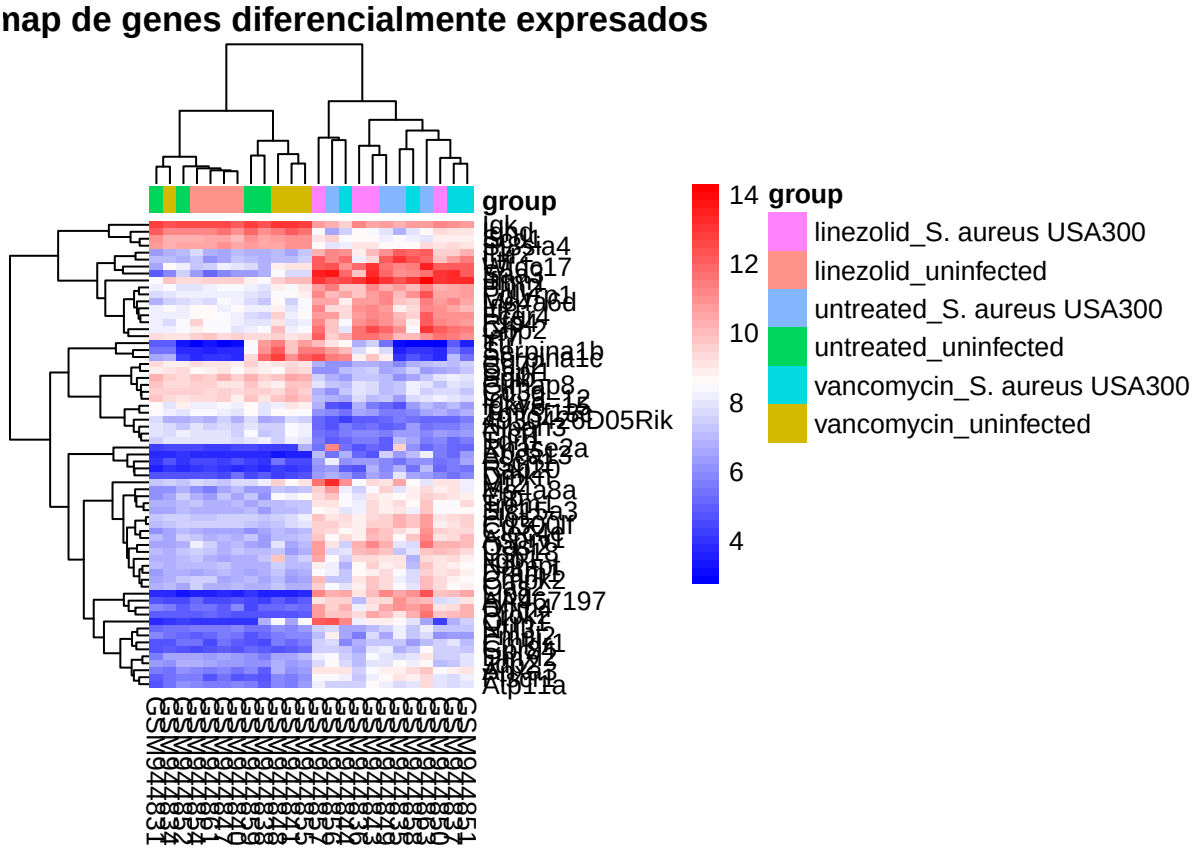

Gráfico 10: Heatmap para los genes expresados de manera diferencia. sin hacer un cluster

Una vez realizado el heatmap con el cluster previo, podemos ver que hay dos grupos muy diferenciados que parecen resultar ser los grupos sin infectar y los grupos infectados, donde podemos ver que hay expresión diferencial compartida entre estos.

4.6 ANÁLISIS DE SIGNIFICANCIA BIOLÓGICA

Finalmente hemos identificado los genes que están diferencialmente expresados, por lo que ahora se llevan a cabo las interpretaciones de estos resultados.

El análisis de significancia biológcica se puede realizar de varias maneras. En este estudio se ha optado por dos vías:

- **Gene Ontology Enrichment**: GO Enrichment, o análsis de enriquecimiento, está pensado para buscar dentro de los genes diferenciales sus ontologías, es decir, sus funciones, y agruparlos. Este análisis nos indicará qué proceso biológico (BP), función molecular (MF) o componente celular (CC), está más alterado según lo que nosotros le indiquemos. Para los fines de este estudio, lo más interesante puede ser ver el proceso biológico alterado, aunque un análisis de consonancia de las 3 variables sería lo idóneo para averiguar las vías afectadas por los genes.
- **Gene Set Enrichment Analysis**: GSEA utiliza la lista de genes ordenada, y dependiendo de log fold change que tenga cada gen interpreta si hay enriquecimiento o no. Esto ayuda a detectar los grupos de genes que se encuentran alterados.

Estos análisis se llevaran a cabo con las funciones enruchGO() y gseGO() del paquete clusterprofiler de Bioconductor. Este paquete ayuda a elegir los parámetros para el análisis como los p-valores o log fold change, además de dar como resultados objetos que se pueden graficar.

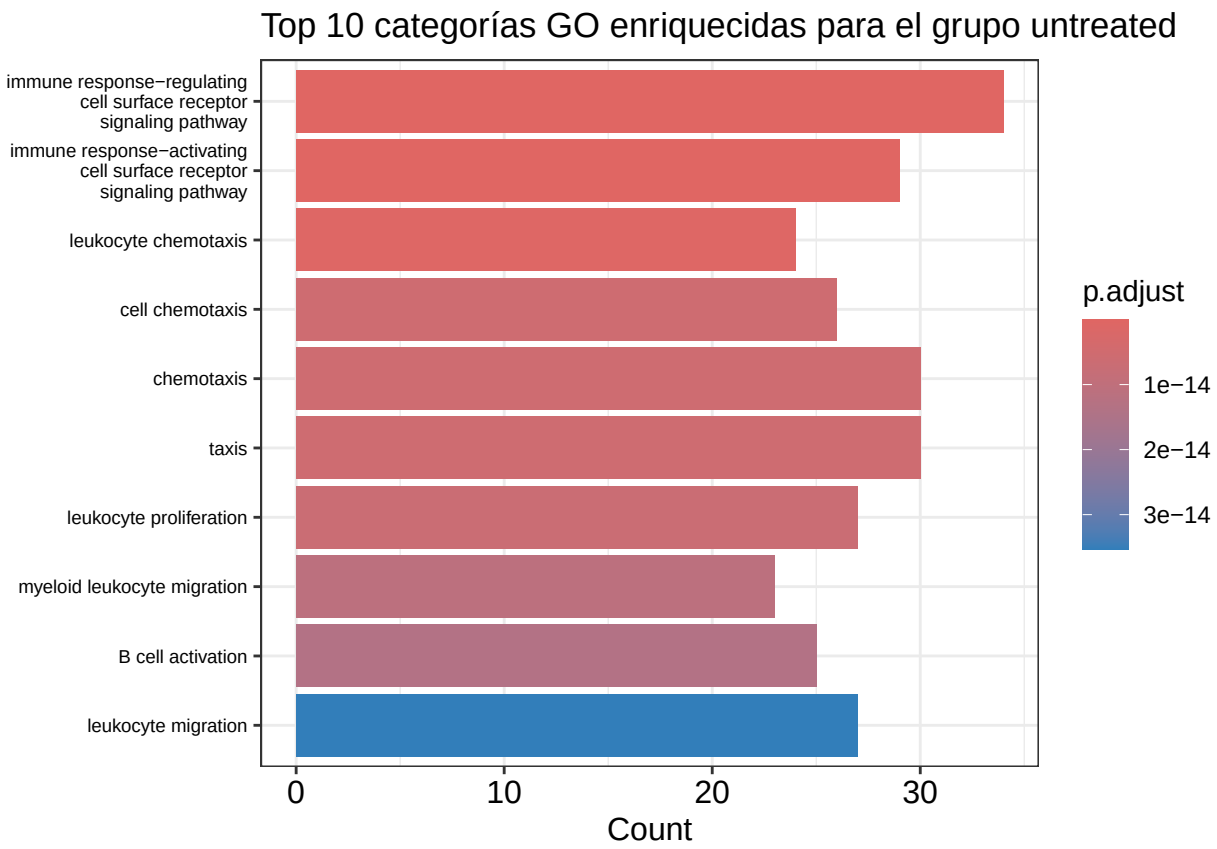

Gráfico 11: Gráfico de barras de las categorías GO enriquecidas para el grupo control.

Se puede observar que las ontolgías enriquecidas por estos genes diferenciales son las que tienen que ver con la respuesta inmune regulada, la quimiotaxis y con proliferación y migración de leucocitos. De esto podemos interpretar que las alteraciones tienen sobretodo cambios en el sistema inmunitario. Esto tiene sentido ya que los primeros cambios que deberíamos ver es la respuesta innata frente al patógeno. La quimiotaxis hace referencia al gradiente químico que ayuda a localizar la infección y a que las células B puedan llegar.

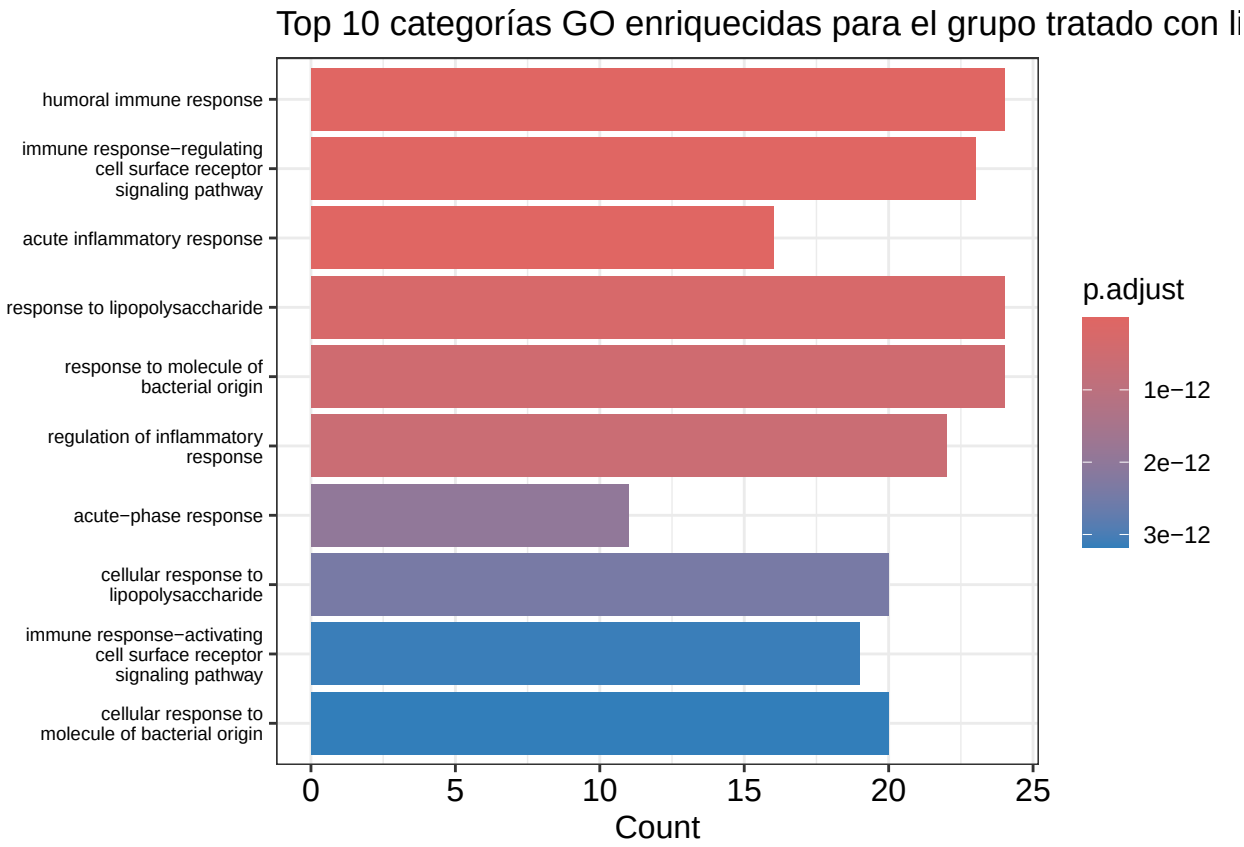

Gráfico 12: Gráfico de barras de las categorías GO enriquecidas para el grupo tratado con linezolid.

En el grupo tratado con linezolid vemos que las vías principales son la respuesta a bacterias, la respuesta humoral y a lipopolisacáridos, es decir, una respuesta más focalizada. La respuesta inmune humoral es la mediada por anticuerpos liberados por células B. Seguramente, se active la proliferación de células B productoras de igM para hacer frente a la infección, ya que el organismo ha detectado la presencia de sustancias bacterianas, por lo que es una respuesta mucho más centrada en el organismo invasor.

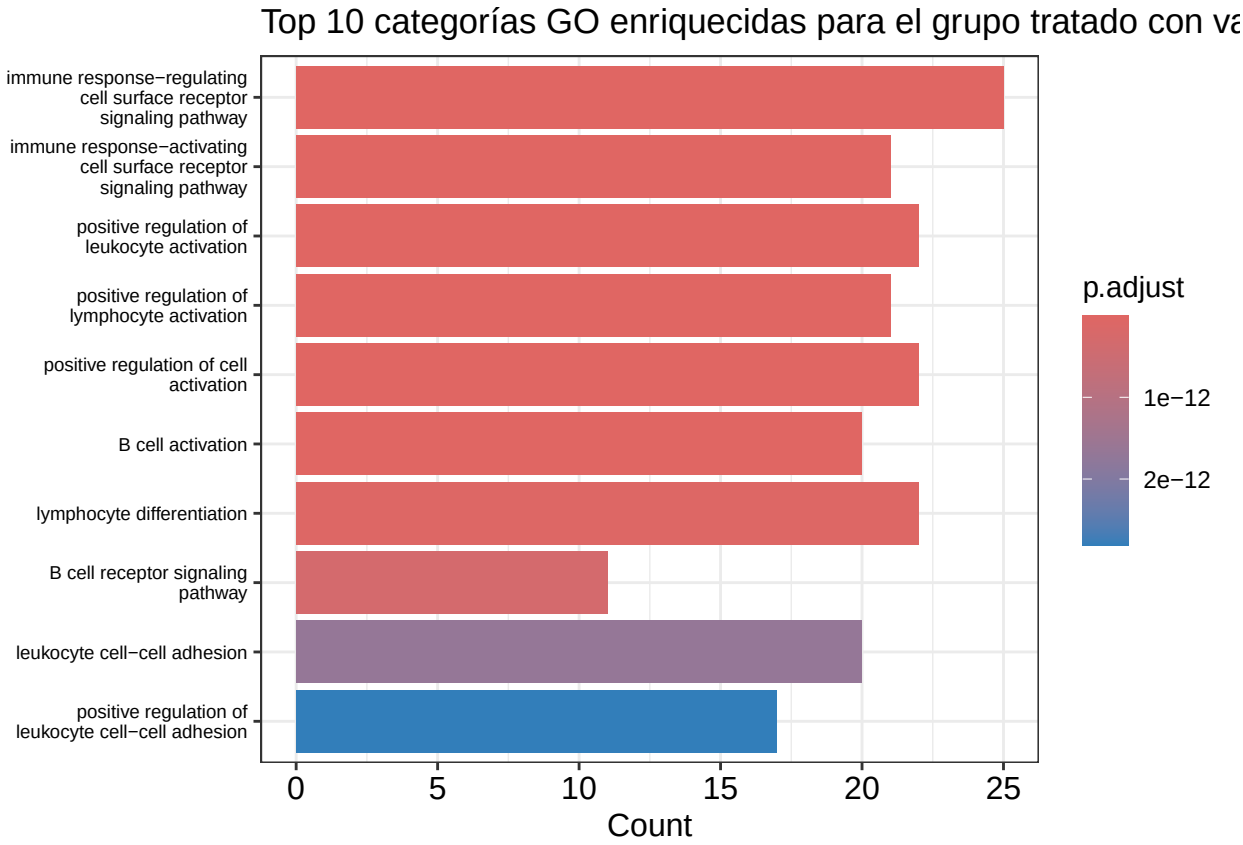

Gráfico 13: Gráfico de barras de las categorías GO enriquecidas para el grupo tratado con vancomicina.

Por último, el grupo tratado con vancomicina presenta unos GO enriquecidos muy similares al grupo control, con regulaciones positivas del sistema inmune, aunque aquí se ve mucha más activación y diferenciación de células B, ya que para que estas estén activas debe promoverse su diferenciación en los ganglios. De nuevo, es una reacción esperable, más directa a activar la respuesta de fase aguda que el grupo control pero menos focalizada como la que se observaba en el grupo linezolid.

4.6.1 GENE SET ENRICHMENT

Tabla 12: Resultados de GSEA para el grupo control.

```
##                    ID                                     Description setSize
## GO:0006950 GO:0006950                                response to stress     108
## GO:0042113 GO:0042113                                  B cell activation      25
## GO:0046649 GO:0046649                               lymphocyte activation      55
## GO:1901701 GO:1901701 cellular response to oxygen-containing compound      31
## GO:0050851 GO:0050851     antigen receptor-mediated signaling pathway      16
## GO:0006952 GO:0006952                                defense response      91
## GO:0042221 GO:0042221                              response to chemical     101
## GO:0071396 GO:0071396                 cellular response to lipid      21
## GO:0030183 GO:0030183                     B cell differentiation      12
## GO:0009607 GO:0009607                 response to biotic stimulus      88
```

Los resultados esta vez se visualizan en formato tabla, aunque podría elegirse otro gráfico como el de barras como hemos visto para los datos de GO. Aquí podemos ver el ID del "Gene Ontology", la descripción y los

genes dentro de los que se han estudiado que estan dentro de la ontología. Por eso, podemos ver que en el caso del grupo control lo que más se ve de nuevo un enriquecimiento de las vías de respuesta inmune, respuesta a estrés, activación y diferenciación de células B, en definitiva, la misma idea que para el GO enrichment.

Tabla 13: Resultados de GSEA para el grupo tratado con linezolid.

```
##                    ID                          Description setSize
## GO:0030098 GO:0030098       lymphocyte differentiation      14
## GO:0046649 GO:0046649            lymphocyte activation      25
## GO:0006950 GO:0006950               response to stress      96
## GO:0042113 GO:0042113                B cell activation      12
## GO:0006952 GO:0006952                 defense response      81
## GO:0045321 GO:0045321              leukocyte activation      31
## GO:1903131 GO:1903131 mononuclear cell differentiation      18
## GO:0042221 GO:0042221              response to chemical      83
## GO:0002252 GO:0002252          immune effector process      34
## GO:0006996 GO:0006996           organelle organization      13
```

Este grupo como habíamos visto previamente enriquecía las vías de diferenciación y activación B. Otra vez podemos ver que la diferenciación y activación leucocitaria se ve alterada, como respuesta a la alteración de las vías de respuesta a estrés y respuesta de defensa. De nuevo, recordemos que la respuesta aguda se centrará en presentar antígenos y generar células de respuesta.

Tabla 14: Resultados de GSEA para el grupo tratado con vancomicina.

```
##                    ID                          Description setSize
## GO:0006950 GO:0006950               response to stress      50
## GO:0006952 GO:0006952                 defense response      42
## GO:0046649 GO:0046649            lymphocyte activation      40
## GO:0042113 GO:0042113                B cell activation      20
## GO:0006954 GO:0006954            inflammatory response      23
## GO:0034097 GO:0034097             response to cytokine      20
## GO:1901652 GO:1901652              response to peptide      20
## GO:0042221 GO:0042221              response to chemical      49
## GO:0009605 GO:0009605    response to external stimulus      60
## GO:0080134 GO:0080134 regulation of response to stress      19
```

DE nuevo, como se venía viendo de los otros grupos la respuesta a estrés es la vía más alterada. La activación linfocitaria, respuesta B es común en todos los grupos gracias a la respuesta aguda, aunque en este grupo en concreto podemos ver que también hay enriquecimiento de vías inflamatorias y de respuesta a citoquinas. Las citoquinas son moléculas caracterísiticas de la respuesta inmune innata y adaptativa, encargadas de la señalización celular, que promueven proliferación y diferenciación de linfocitos, así como la quimiotaxis y la secreción de inmunoglobulinas. Basándonos en que la respuesta es reciente y a un microorganismo seguramente estemos hablando de interleucinas 1 o 2, las cuales a parte de ser proinflamatorias promueven la activación de linfocitos T helpers, responsables de presentar el antígeno a las células B en los ganglios, lo que puede llegar a indicar que la vancomicina está promoviendo una respuesta más fuerte que linezolid.

5 CONCLUSIONES

El estudio ha cumplido los objetivos principales, habiendo hecho un análisis estándar de microarry y de expresión diferencial.

Para comenzar con el propio estudio, se encuentra como primer incoveniente un número de muestras bastante bajo, siendo este de 4 muestras por grupo problema, lo que ofrece una reproducibilidad más baja de lo normal y susceptibilidad al efecto batch mayor. Podría haber sido interesante trabajar con todas las muestras para ver si los resultados variaban de alguna manera o se vería la expresión diferencial del grupo control sin infectar enfrentado al grupo control después de 2h de infección, y de nuevo este grupo con el grupo de 24h infectado.

Aún con las limitaciones que presentan la mayoría de estudios de este calibre, se ha podido entender y destacar las vías que potencialmente se ven afectadas por la infección bacteriana y si los antibióticos pueden modular la respuesta.

- El grupo control tiene importancia porque es el que nos permite comparar. Aunque no tenga nada fuera de lo común ya que las vías que vemos alteradas son las esperadas, es interesante ver cómo el organismo activa las respuestas de defensa innata, ya que se encuentra en las primeras horas de la infección, proliferando sobretodo la quimiotaxis para enviar los leucocitos a combatir el foco de infección.
- El grupo linezolid por su parte se centra en un potenciamiento de la respuesta inmune innata, más enfocada a la proliferación de células B, encargadas de la producción de los anticuerpos, los cuales aún en las primeras fases de la infección son importantes para la defensa gracias a la promiscuidad de la igM.
- El grupo vancomicina en cambio, se centra en una respuesta más brusca, proinflamatoria y de potenciamiento de linfocitos T helper, encargados de la presentación de antígeno a los linfocitos B para producir más rápidamente la respuesta adaptativa.

Ambos tratamientos abordan la respuesta inmune y su potenciamiento de maneras diferentes, pero no se puede determinar cuál es mejor o peor ya que cada organismo puede responder de manera diferente a los antibióticos, a la inflamación o al proliferamiento de la respuesta. Para realizar un estudio más en profundidad, se podría ver dentro de los Gene Ontology, las funciones moleculares y componentes celulares afectados para tener una idea más concreta de las alteraciones diferenciales.

Para concluir, cabe decir que este estudio no es definitivo, y que se necesitarían más estudios de laboratorio para determinar si los genes candidatos están o no diferencialmente expresados, pero aún así es una manera efectiva y potente para discriminar entre la inmensidad de genes que se observan en el microarray.

6 APÉNDICE

6.1 CÓDIGO COMPLETO

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager", repos = "https://cran.rstudio.com")

BiocManager::install("Biobase")
BiocManager::install("limma")
BiocManager::install("affy")
BiocManager::install("oligo")
BiocManager::install("arrayQualityMetrics")
BiocManager::install("genefilter")
BiocManager::install("mouse4302")
BiocManager::install("clusterProfiler")
BiocManager::install("enrichplot")
```

```r
BiocManager::install("org.Mm.eg.db")

install.packages("readr", repos = "https://cran.rstudio.com")
install.packages("dplyr", repos = "https://cran.rstudio.com")
install.packages("DT", repos = "https://cran.rstudio.com")
install.packages("kableExtra", repos = "https://cran.rstudio.com")
install.packages("pheatmap", repos = "https://cran.rstudio.com")
install.packages("dendextend", repos = "https://cran.rstudio.com")

library(affy)
library(Biobase)
library(oligo)
library(dendextend)
library(ggplot2)
library(arrayQualityMetrics)
library(genefilter)
library(limma)
library(mouse4302.db)
library(AnnotationDbi)
library(pheatmap)
library(clusterProfiler)
library(enrichplot)

filter_microarray <- function(allTargets, seed = 49933547) {
  # Configurar la semilla aleatoria
  set.seed(49933547)

  # Filtrar las filas donde 'time' no sea 'hour 2'
  filtered <- subset(allTargets, time != "hour 2")

  # Dividir el dataset por grupos únicos de 'infection' + 'agent'
  filtered$group <- interaction(filtered$infection, filtered$agent)

  # Seleccionar 4 muestras al azar de cada grupo
  selected <- do.call(rbind, lapply(split(filtered, filtered$group), function(group_data) {
    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 seleccionadas
  original_indices <- match(selected$sample, allTargets$sample)

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

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

```r
# Simular el dataset basado en la descripción proporcionada
allTargets <- data.frame(
  sample = c("GSM944831", "GSM944838", "GSM944845", "GSM944852", "GSM944859",
             "GSM944833", "GSM944840", "GSM944847", "GSM944854", "GSM944861",
             "GSM944834", "GSM944841", "GSM944848", "GSM944855", "GSM944862",
             "GSM944832", "GSM944839", "GSM944846", "GSM944853", "GSM944860",
             "GSM944835", "GSM944842", "GSM944849", "GSM944856", "GSM944863",
             "GSM944836", "GSM944843", "GSM944850", "GSM944857", "GSM944864",
             "GSM944837", "GSM944844", "GSM944851", "GSM944858", "GSM944865"),
  infection = c(rep("uninfected", 15), rep("S. aureus USA300", 20)),
  time = c(rep("hour 0", 15), rep("hour 2", 5), rep("hour 24", 15)),
  agent = c(rep("untreated", 5), rep("linezolid", 5), rep("vancomycin", 5),
            rep("untreated", 5), rep("untreated", 5), rep("linezolid", 5), rep("vancomycin", 5))
)

# Aplicar la función (cambiar 123 por vuestro ID de la UOC u otro número que podáis escribir en el docu
result <- filter_microarray(allTargets, seed=49933547)

# Partiendo de las muestras que hemos aleatorizado, nos quedamos con la columma samples que es la que c
seleccionadas <- result$sample

# Buscamos los archivos .CEL en la ruta donde los tenemos guardados
archivos_cel <- list.files(path = "C:/Users/Bruno/Desktop/PEC2/GES38531_RAW", pattern="\\.CEL$", full.na

# Seleccionamos aquellas muestras que coincidan en id con las que trabajaremos que se encuentran en la
muestras_seleccionadas <- sapply(seleccionadas, function(id) {
  matches <- grep(paste0("^", id), basename(archivos_cel), value = TRUE)
  for (m in matches) {
    if (length(matches) == 1) {
      return(file.path("C:/Users/Bruno/Desktop/PEC2/GES38531_RAW", m))
    }
  }
  return(NA)
})

# Leemos el archivo de los fenodatos
pdata_todo <- read.table("allTargets.txt", header = TRUE, sep = " ", stringsAsFactors = FALSE)

# Filtramos para aquellas muestras que nos interesa trabajar puesto que son las que hemos seleccionado
pdata_sel <- pdata_todo[pdata_todo$sample %in% seleccionadas, ]

# Ordenados los fenoadtos para que estén alineados
pdata_sel <- pdata_sel[match(seleccionadas, pdata_sel$sample), ]
rownames(pdata_sel) <- pdata_sel$sample

# Mostramos los fenodatos
print(pdata_sel)

# Creamos el data frame de los fenodatos
pheno_data <- AnnotatedDataFrame(pdata_sel)

# Generamos el Expression Set
datos_crudos <- read.celfiles(filenames=muestras_seleccionadas, phenoData=pheno_data)
```

```r
# Los nombres de los archivos CEL contienen información extra que eliminaremos para hacer más visual el
clean_names <- sub("_.*", "", sampleNames(protocolData(datos_crudos)))

# Asignamos los nuevos nombres limpios
sampleNames(protocolData(datos_crudos)) <- clean_names


colnames(exprs(datos_crudos)) <- rownames(pData(datos_crudos))

show(datos_crudos)

# Creamos una variable que contenga el nombre de las muestras
nombre_muestras <- colnames(exprs(datos_crudos))

# Creamos una columna combinada para los grupos
pData(datos_crudos)$group <- with(pData(datos_crudos), paste(agent, infection, sep = "_"))

# Definimos unos colores para cada grupo, para que posteriormente cuando hagamos gráficos el resultado
colores_grupos <- c("untreated_uninfected" = "lightblue",
                    "untreated_S. aureus USA300" = "blue",
                    "linezolid_uninfected" = "lightgreen",
                    "linezolid_S. aureus USA300" = "darkgreen",
                    "vancomycin_uninfected" = "lightcoral",
                    "vancomycin_S. aureus USA300" = "red")

# Asignar los colores a cada muestra
colores_muestras <- colores_grupos[pData(datos_crudos)$group]

# Generamos el boxplot para los datos
boxplot(datos_crudos, col = colores_muestras,las = 2, cex.axis = 0.6, main = "Gráfico de cajas para las

# Generamos el histograma para los datos
hist(datos_crudos, col = colores_muestras,las = 2, cex.axis = 0.6, main = "Distribución de las muestras

# Generamos el dendograma
cluster_datos_crudos<- hclust(dist(t(exprs(datos_crudos))), method = "average")

dend_crudo <- as.dendrogram(cluster_datos_crudos)

labels_colors(dend_crudo) <- colores_muestras[order.dendrogram(dend_crudo)]

plot(dend_crudo, main = "Dendrograma de las muestras", cex = 0.7)

# Para estudiar el PCA, primero creamos una función para representarlo
plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE, formapunts=NULL, myCex=0.8
{
  pcX <- prcomp(t(X), scale=scale)
  loads <- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
  xlab <- c(paste("PC1", loads[1],"%"))
  ylab <- c(paste("PC2", loads[2],"%"))
  if (is.null(colors)) colors=1
  plot(pcX$x[,1:2], xlab=xlab, ylab=ylab, col=colors, pch=formapunts, xlim=c(min(pcX$x[,1])-100000, max
  text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=myCex)
```

```
      title(paste("Gráfico de los dos primeros PC en", dataDesc, sep=" "), cex=0.8)
}

# Graficamos el PCA
plotPCA(exprs(datos_crudos), labels=nombre_muestras, dataDesc="datos crudos", colors=colores_muestras, i

# Realizamos el Array Quality Metrics y añadimos el rerun para no ejecutarlo más de una vez
rerun <- FALSE
if(rerun){
  arrayQualityMetrics(datos_crudos, reporttitle = "QC_Datos_Crudos", force = FALSE)
}

# Normalizamos los datos
datos_norm <- rma(datos_crudos)

datos_norm

probeIds <- rownames(exprs(datos_norm))

# Filtramos los datos, y nos quedamos con el cutoff de 0.9, para que nos devuelva el 10% de genes con m

annotation(datos_norm) <- "mouse4302"

datos_filtrados <- nsFilter(datos_norm, var.func = IQR, var.cutoff = 0.9, var.filter = TRUE, filterByQua

print(datos_filtrados)

print(datos_filtrados$datos_norm)

# Cambiamos los nombres de S. aureus USA300 a infected para hacer la presentación de las matrices más v

pData(datos_norm)$group <- gsub("S. aureus USA300", "infected", pData(datos_norm)$group)
# Creamos la matriz de diseño
design <- model.matrix(~ 0 + pData(datos_norm)$group)
colnames(design) <- gsub("pData\\(datos_norm\\)\\$group", "", colnames(design))

# Creamos las matrices de contrastes, haciendo las comparaciones entre los grupos
contrastes <- makeContrasts(
    untreated_infected_vs_uninfected = untreated_infected - untreated_uninfected,
    linezolid_infected_vs_uninfected = linezolid_infected - linezolid_uninfected,
    vancomycin_infected_vs_uninfected = vancomycin_infected - vancomycin_uninfected,
    levels = design
)

print(design)

# Visualizamos la matriz de contrastes
print(contrastes)

fit <- lmFit(exprs(datos_filtrados$eset), design)

# Aplicamos los contrastes
fit2 <- contrasts.fit(fit, contrastes)
```

```r
fit2 <- eBayes(fit2)

# Creamos las toptables
top_untreated <- topTable(fit2, coef = "untreated_infected_vs_uninfected", number = Inf, adjust="fdr", l

top_linezolid <- topTable(fit2, coef = "linezolid_infected_vs_uninfected", number = Inf, adjust="fdr", l

top_vancomycin <- topTable(fit2, coef = "vancomycin_infected_vs_uninfected", number = Inf, adjust="fdr"

# Mostramos los resultados de los contrastes para el grupo control
print(head(top_untreated))
cat("Genes diferenciales para el grupo control (infectado vs sin infectar):", nrow(top_untreated), "\n")

# Grupo linezolid
print(head(top_linezolid))
cat("Genes diferenciales para linezolid (infectado vs sin infectar):", nrow(top_linezolid), "\n")

# Grupo vancomicina
print(head(top_vancomycin))
cat("Genes diferenciales para vancomycin (infectado vs sin infectar):", nrow(top_vancomycin), "\n")

# Seleccionamos el paquete de anotación e indicamos las claves que deberá buscar y las columnas que que
anotaciones <- AnnotationDbi::select(mouse4302.db, keys = probeIds, columns = c("ENTREZID", "SYMBOL"))

# Generamos un pipeline para cada toptable donde añadiremos las anotaciones y las enseñaremos en las pr
top_untreated_anotado <- top_untreated %>% mutate(probeIds=rownames(top_untreated)) %>% merge(anotacion

top_linezolid_anotado <- top_linezolid %>% mutate(probeIds=rownames(top_linezolid)) %>% merge(anotacion

top_vancomycin_anotado <- top_vancomycin %>% mutate(probeIds=rownames(top_vancomycin)) %>% merge(anotac

# Visualizamos los resultados
head(top_untreated_anotado)

# Para el grupo linezolid
head(top_linezolid_anotado)

# Para el grupo vancomicina
head(top_vancomycin_anotado)

# Generamos la matriz de decisiones mediante decide test
resultados <- decideTests(fit2, method = "separate", adjust.method = "fdr", p.value = 0.05, lfc=2)

suma.resultados<-apply(abs(resultados),1,sum)
res.selected<-resultados[suma.resultados!=0,]
print(summary(resultados))

vennDiagram (res.selected[,1:3], main="Genes comunes", cex=0.9)

# A la hora de crear los volcano plots, primero seleccionamos los nombres de los genes para anotarlos e
gene_symbols_untreated <- top_untreated_anotado$SYMBOL[match(rownames(fit2$coefficients), top_untreated_

coef1 <- 1
```

```r
# Graficamos el volcano plot y le añadimos dos líneas para indicar el lfc=2
volcanoplot(fit2, highlight = 10, names = gene_symbols_untreated, coef=coef1, main=paste("Expresión de g
abline(v=c(-2,2))

# Para el grupo linezolid
gene_symbols_linezolid <- top_linezolid_anotado$SYMBOL[match(rownames(fit2$coefficients), top_linezolid_

coef2 <- 2
volcanoplot(fit2, highlight = 10, names = gene_symbols_linezolid, coef=coef2, main=paste("Expresión de g
abline(v=c(-2,2))

# Para el grupo vancomicina
gene_symbols_vancomycin <- top_vancomycin_anotado$SYMBOL[match(rownames(fit2$coefficients), top_vancomyc

coef3 <- 3
volcanoplot(fit2, highlight = 10, names = gene_symbols_vancomycin, coef=coef3, main=paste("Expresión de
abline(v=c(-2,2))

# Para generar el heatmap lo haremos desde la matriz creada para el decide test, ya que tiene los genes
genes_seleccionados <- rownames(res.selected)
genes_seleccionados.selected <- genes_seleccionados[suma.resultados!=0]
genes_validos <- intersect(genes_seleccionados.selected, rownames(exprs(datos_filtrados$eset)))

# Seleccionamos los genes en nuestro expresion set filtrado
matriz_heatmap <- exprs(datos_filtrados$eset)[genes_validos, ]

# Anotamos los símbolos para que nos sea más fácil la identificación
symbols <- AnnotationDbi::select(mouse4302.db, keys = rownames(matriz_heatmap), columns = "SYMBOL", key
rownames(matriz_heatmap) <- symbols$SYMBOL[match(rownames(matriz_heatmap), symbols$PROBEID)]

anotaciones_muestras <- data.frame(group = pData(datos_filtrados$eset)$group)

rownames(anotaciones_muestras) <- colnames(matriz_heatmap)

# Creamos el heatmap
colores_heatmap <- colorRampPalette(c("blue", "white", "red"))(50)

pheatmap(matriz_heatmap, color = colores_heatmap, border_color = NA, annotation_col = anotaciones_muest

# Heatmap por cluster
pheatmap(matriz_heatmap, color = colores_heatmap, border_color = NA, annotation_col = anotaciones_muest

# Filtramos por los genes diferencialmente expresados
# Para el grupo control
genes_significativos_untreated <- top_untreated_anotado[ top_untreated_anotado$adj.P.Val < 0.05 & abs(t

# Realizamos el análisis de sobre-representación para GO
go_enrich_untreat <- enrichGO(gene = genes_significativos_untreated, OrgDb = mouse4302.db, keyType = "E

# Para el grupo linezolid
genes_significativos_linezolid <- top_linezolid_anotado[ top_linezolid_anotado$adj.P.Val < 0.05 & abs(t
```

```r
go_enrich_linezolid <- enrichGO(gene = genes_significativos_linezolid, OrgDb = mouse4302.db, keyType = 

# Para el grupo vancomicina
genes_significativos_vancomycin <- top_vancomycin_anotado[ top_vancomycin_anotado$adj.P.Val < 0.05 & abs

go_enrich_vancomycin <- enrichGO(gene = genes_significativos_vancomycin, OrgDb = mouse4302.db, keyType =

# Creamos un gráfico para el grupo control
barplot(go_enrich_untreat, showCategory = 10, title = "Top 10 categorías GO enriquecidas para el grupo u

# Para el grupo linezolid
barplot(go_enrich_linezolid, showCategory = 10, title = "Top 10 categorías GO enriquecidas para el grupo

# Para el grupo vancomycin
barplot(go_enrich_vancomycin, showCategory = 10, title = "Top 10 categorías GO enriquecidas para el grup

# Ordenamos los resultados

top_filtrado_untreated <- top_untreated_anotado[order(-top_untreated_anotado$logFC), ]

# Creamos el vector que contiene los genes a analizar por gseGO
genes_untreated <- setNames(top_filtrado_untreated$logFC, top_filtrado_untreated$ENTREZID)

# Realizamos el análisis para el grupo control
gsea_go_untreated <- gseGO(geneList = genes_untreated, OrgDb = mouse4302.db, keyType = "ENTREZID", ont =

# Visualizamos los resultados
head(gsea_go_untreated[,1:3],10)

top_filtrado_linezolid <- top_linezolid_anotado[order(-top_linezolid_anotado$logFC), ]

# Creamos la lista de genes
genes_linezolid <- setNames(top_filtrado_linezolid$logFC, top_filtrado_linezolid$ENTREZID)

# Realizamos el gseGO para el grupo linezolid
gsea_go_linezolid <- gseGO(geneList = genes_linezolid,OrgDb = mouse4302.db,keyType = "ENTREZID",  ont =

head(gsea_go_linezolid[,1:3],10)

top_filtrado_vancomycin <- top_vancomycin_anotado[order(-top_vancomycin_anotado$logFC), ]

# Creamos la lista de genes
genes_vancomycin <- setNames(top_filtrado_vancomycin$logFC, top_filtrado_vancomycin$ENTREZID)

# Realizamos el gseGO para el grupo vancomycin
gsea_go_vancomycin <- gseGO(geneList = genes_vancomycin, OrgDb = mouse4302.db, keyType = "ENTREZID", ont


head(gsea_go_vancomycin[,1:3],10)
```