Análisis del perfil de expresión génica entre ratones tratados con linezolid o vancomicina en la sepsis por MRSA

(anaaliciaUOC?)

2025-01-05

Contents

1 Introducción y Objetivos

Staphylococcus aureus es una de las principales causas de infecciones nosocomiales y adquiridas en la comunidad. Hay evidencia que indica que ciertas toxinas bacterianas juegan un papel clave en la patogénesis de diversas infecciones por MRSA (*S. aureus* resistente a meticilina). (Sharma-Kuinkel et al. 2013)

Se ha demostrado que la inhibición de la síntesis de toxinas bacterianas mediante antibióticos que bloquean la síntesis de proteínas puede reducir la producción de toxinas y mejorar los resultados clínicos. Linezolid (LNZ), a diferencia de la vancomicina (VAN), es un potente inhibidor de la síntesis de proteínas bacterianas. Por otro lado, los agentes activos sobre la pared celular, como la vancomicina, tienden a estimular la producción de toxinas bacterianas. Estas diferencias sugieren que ambas clases de antibióticos tienen efectos opuestos sobre la expresión de genes de virulencia. (Sharma-Kuinkel et al. 2013)

El objetivo de este análisis es evaluar simultáneamente cómo afectan LNZ y VAN tanto al huésped como al patógeno en un modelo murino de sepsis por MRSA. Aquí se analiza la hipótesis de que LNZ genera un perfil de expresión génica diferente al de VAN en la sepsis por MRSA. Para ello realizaremos comparaciones en el perfil de expresión génico entre infectados y no infectados: sin tratamiento, en tratamiento con LNZ y en tratamiento con VAN.

2 Métodos

2.1 Metodología para la descarga y preparación de los datos

Siguiendo las directrices de la práctica los datos crudos en formato CELL han sido descargados de GEO Accesion viewer: https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE38531. Se ha customizado la descarga para no incluir las 5 muestras tomadas a las dos horas. Se ha guardado una carpeta llamada GSE38531_RAW en la carpeta del proyecto de R, junto con el archivo "allTargets.txt" proporcionado en el enunciado de la práctica. Los datos crudos han sido modelizados según el algoritmo RMA (Robust Multi-array Average) con el objetivo de ajustar el ruido de fondo, tomar logaritmos base 2 de cada intensidad, normalizar y estimar las intensidades de cada gen separadamente para cada conjunto de ondas.(Gautier et al. 2004) Se ha aplicado la función `filter_microarray` proporcionada en el enunciado para obtener un subconjunto filtrado y procesado de los datos basado en reglas específicas de dicha función.

2.2 Metodología del análisis exploratorio y control de calidad

Para el análisis exploratorio y control de calidad de los datos se han realizado visualizaciones de los datos en diagramas de caja o "boxplots", histogramas y gráficos de densidad, gráficos de varianza, gráficos para la visualización de componentes principales (PCA) y clustering. Además se ha realizado un informe complementario de calidad utilizando el paquete `arrayqualitymetrics`.(Kauffmann, Gentleman, and Huber 2009) Se ha utilizado el paquete `limma`, para procesar y limpiar los datos, y el paquete `umap` para representarlos de manera más comprensible y visual.[Ritchie et al. (2015)](Konopka 2023)

2.3 Metodología del filtrado de datos

Para el filtrado de los datos se ha calculado la desviación de cada fila de la matriz (sonda), los datos se han ordenado de mayor desviación a menor y se ha seleccionado el 10% empezando por la primera fila, obteniendo asi el 10% de las sondas con mayor variabilidad en la expresión genética a lo largo de las distintas muestras.

2.4 Metodología para la construcción de las matrices de diseño y de contrastes

Para realizar ambas matrices se utiliza el paquete `limma`.(Ritchie et al. 2015) Para la creación de la matriz de diseño aplicamos la función `model.matrix()` que describe el diseño del experimento y, para la matriz de contrastes `makeContrasts()` que define los contrastes entre las condiciones de interés: las comparaciones de infectados vs no infectados para diferentes tratamientos.

2.5 Metodología para la obtención de las listas de genes diferencialmente expresados para cada comparación

Se utilizó la función `lmFit()` del paquete `limma` para ajustar un modelo lineal sobre la matriz de expresión génica. Con la función `contrasts.fit()`, se aplicaron contrastes para comparar los diferentes grupos o condiciones especificadas en la matriz de contrastes. Se utilizó la función `eBayes()` para realizar un ajuste bayesiano de los estadísticos del modelo. Para cada comparación, se usó la función `topTable()` para obtener los genes más significativos ajustando los valores p con el método FDR (False Discovery Rate). Se aplicó la función `decideTests()` para decidir cuáles genes son significativamente diferentes entre las condiciones. Se ajustaron los valores p con el método FDR y se utilizó un umbral de 0.01 para identificar los genes significativos. Se filtraron los resultados con una restricción en el valor del cambio de expresión (log fold change, LFC = 1). Se utilizó la función `vennDiagram()` para crear un diagrama de Venn y visualizar los genes en común entre las condiciones seleccionadas.(Ritchie et al. 2015)

2.6 Metodología para la anotación de los genes

Se utilizó el paquete `biomart` para conectar y consultar la base de datos de `Ensembl`. Se conectó al dataset de Ensembl para el genoma de ratón: mmusculus_gene_ensembl. Para cada conjunto de genes, se realizó una consulta con los atributos: affy_mouse430_2, identificador de ENTREZ, Nombre del gen (HUGO), Identificador de Ensembl más una descripción funcional del gen. Los resultados se obtuvieron filtrando por los identificadores de sonda (affy_mouse430_2).@biomaRt

2.7 Metodología del análisis de la significación biológica

Se extraen las listas de genes anotados con identificadores ENTREZ desde los conjuntos de datos correspondientes a cada grupo. Se utiliza la función `enrichGO` del paquete `clusterProfiler` para realizar el análisis en los tres grupos de genes segun el tipo de tratamiento recibido.@clusterProfiler El ajuste de p-valor se realiza siguiendo el método de Benjamini-Hochberg (BH), los umbrales utilizados han sido adabtados segun el grupo. Los resultados se han visualizado en `dotplot` y `cnetplot`.@enrichplot

3 Resultados

Tras la preparación de los datos obtenemos 24 muestras, segun el tratamiento 3 grupos (sin tratar, LNZ y VAN) con 8 muestras cada uno de 4 ratones no infectados y 4 ratones infectados por la cepa *S. aureus* USA300, de estos a la mitad se les tomo la muestra en tiempo 0 mientras que a la otra mitad a las 24h.

Los gráficos de cajas y de distribución muestran que, tras la normalización, los valores de expresión presentan una distribución homogénea entre las diferentes muestras, evidenciado por la similitud en el tamaño y rango de las cajas y la superposición de las curvas en el gráfico de distribución (Fig 1.A y B).

El gráfico para la visualización de los PCA muestra dos grupos claramente diferenciados, ratones infectados y no infectados, lo que implica que la expresión génica captura diferencias biológicas significativas entre ambas condiciones (Fig 1.C). La representación de grupos jerarquizados muestra agrupaciones más definidas en las muestras que presentan mayor similitud, las muestras con menor distancia entre si corresponden a ratones sometidos a las mismas condiciones (Fig 1.D).

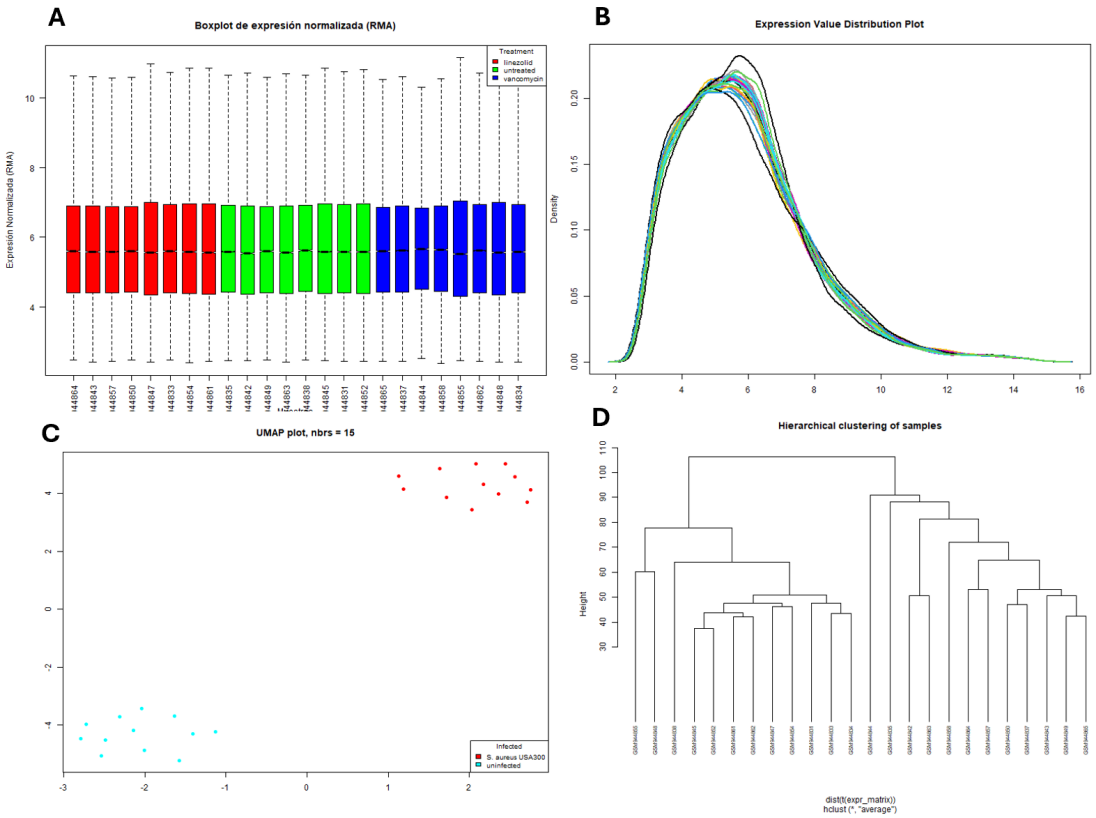

Fig 1. Representaciones para el análisis de calidad. (A) Gráfico de cajas de los valores de expresión coloreados según los grupos de tratamiento. (B) Gráfico de distribución de la expresión génica de todas las muestras. (C) Represetación gráfica de las PCA coloreadas según no infectados e infectados. (D) Representación de grupos jerarquizados de todas las muestras.

En un análisis adicional se muestran diversas figuras interesantes, como un mapa de calor donde se observan regiones representadas con el mismo color, hecho que podría indicar que hay grupos con características comunes (Fig 2.A). Después de la normalización y la transformación a una escala logarítmica de los datos, se espera que la variabilidad (desviación estándar) de las intensidades sea uniforme a lo largo del rango de las medias, es decir, no hay secciones del rango con desviaciones estándar desproporcionadamente altas o bajas. En la figura 2 (B), la línea roja es aproximadamente horizontal, lo que indica un buen equilibrio en los datos. Según la estadística *D* de Hoeffding, ninguna matriz tuvo valores Da>0,15, por lo que no se identificaron valores atípicos (Fig 2.C).

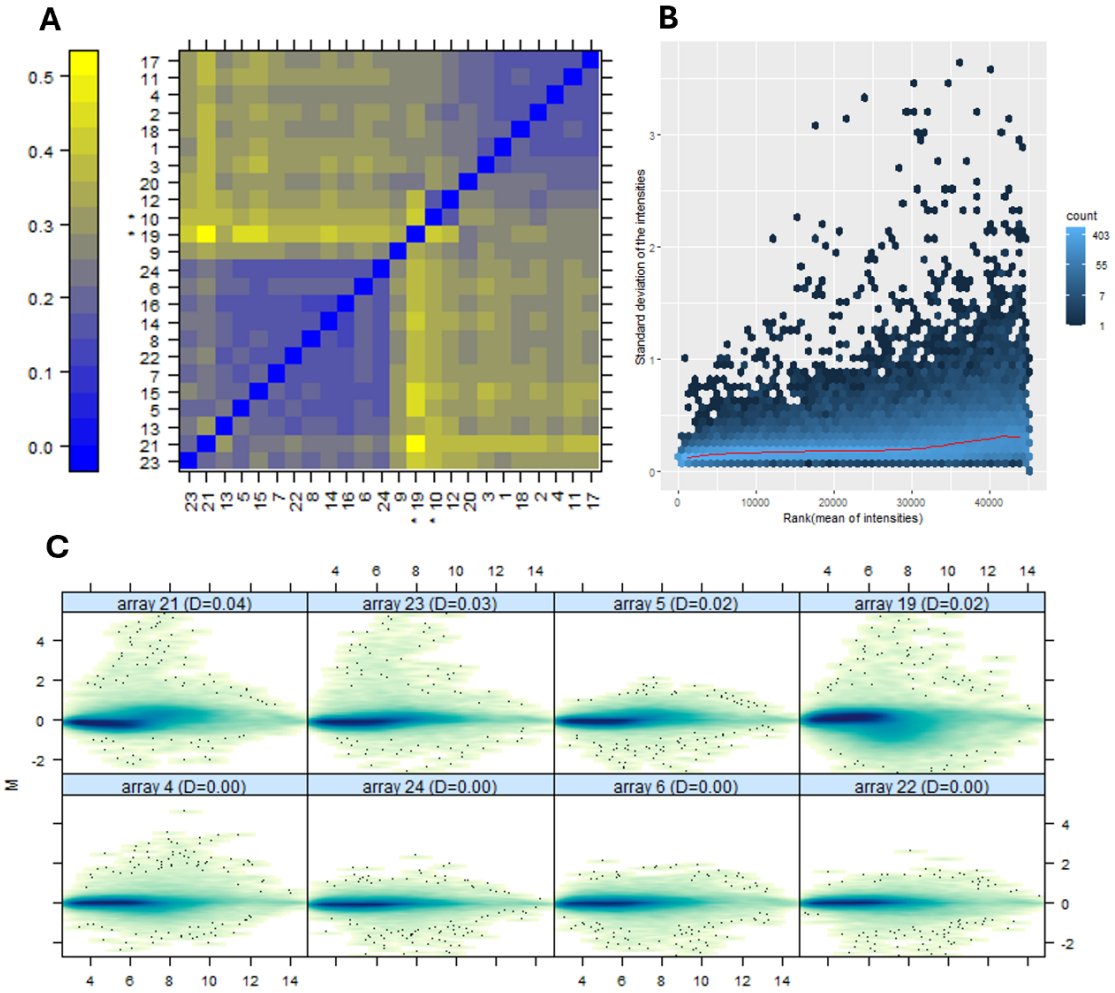

Fig 2. Representaciones para el análisis de calidad adicionales con `arrayqualitymetrics`. (A) Mapa de colores de las distancias entre los datos. (B) Gráfico de densidad con la desviación estándar de las intensidades en el eje y, frente al rango de su media en el eje x. Los puntos rojos, conectados por líneas, representan la mediana móvil de la desviación estándar. (C) Gráficos MA que muestran la relación entre M (log-ratio) y A (intensidad promedio logarítmica) para las matrices analizadas. Las 4 matrices con los valores más altos y más bajos de la estadística *D* de Hoeffding se presentan, con valores indicados en los encabezados.

Tras el filtrado de los datos, se obtienen el 10% de las sondas con mayor variabilidad lo que equivale a un total de 4.511 sondas. De entre los genes con mayor variabilidad observamos que según el tratamiento el grupo con mayor número de genes únicos es el de ratones que no han recibido tratamiento. Mientras que aquellos grupos que más genes comparten, por orden son: no tratados-LNZ, LNZ-VAN y no tratados-VAN (Fig 3).

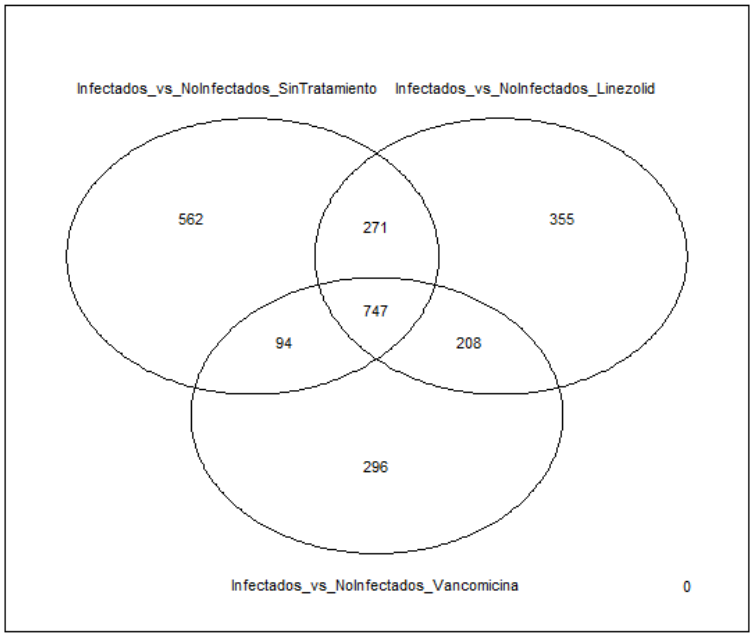

Fig 3. Diagrama de Venn que ilustra la superposición de genes entre las tres categorías seleccionadas. Los círculos representan los conjuntos de genes en cada categoría, y las áreas de intersección muestran los genes en común entre dos o más conjuntos. El número dentro de cada sección indica la cantidad de genes únicos o compartidos.

A continuación se visualiza el listado con los 10 genes con mayor variabilidad en su expresión según el grupo de tratamiento al que pertenecen (Tab 1).

affy_mouse430_2	Id ENTREZ	Id HUGO	Id ENSEMBL	Descripcion
UNTREATED				
1421262_at	16891	Lipg	ENSMUSG00000053846	lipase, endothelial
1450188_s_at	16891	Lipg	ENSMUSG00000053846	lipase, endothelial
1448213_at	16952	Anxa1	ENSMUSG00000024659	annexin A1
1422953_at	14289	Fpr2	ENSMUSG00000052270	formyl peptide receptor 2
1437060_at	380924	Olfm4	ENSMUSG00000022026	olfactomedin 4
1421366_at	23845	Clec5a	ENSMUSG00000029915	C-type lectin domain family 5
1449366_at	17394	Mmp8	ENSMUSG00000005800	matrix metallopeptidase 8
1427747_a_at	16819	Lcn2	ENSMUSG00000026822	lipocalin 2
1440865_at	213002	Ifitm6	ENSMUSG00000059108	interferon induced transmembrane protein 6
1418722_at	18054	Ngp	ENSMUSG00000032484	neutrophilic granule protein
LINEZOLID				
1421262_at	16891	Lipg	ENSMUSG00000053846	lipase, endothelial
1450188_s_at	16891	Lipg	ENSMUSG00000053846	lipase, endothelial
1422953_at	14289	Fpr2	ENSMUSG00000052270	formyl peptide receptor 2
1416301_a_at	13591	Ebf1	ENSMUSG00000057098	early B cell factor 1
1437060_at	380924	Olfm4	ENSMUSG00000022026	olfactomedin 4
1450826_a_at	20210	Saa3	ENSMUSG00000040026	serum amyloid A 3
1449366_at	17394	Mmp8	ENSMUSG00000005800	matrix metallopeptidase 8
1427747_a_at	16819	Lcn2	ENSMUSG00000026822	lipocalin 2
1440865_at	213002	Ifitm6	ENSMUSG00000059108	interferon induced transmembrane protein 6
1418722_at	18054	Ngp	ENSMUSG00000032484	neutrophilic granule protein
VANCOMICINA				
1421262_at	16891	Lipg	ENSMUSG00000053846	lipase, endothelial
1450188_s_at	16891	Lipg	ENSMUSG00000053846	lipase, endothelial
1416301_a_at	13591	Ebf1	ENSMUSG00000057098	early B cell factor 1
1437060_at	380924	Olfm4	ENSMUSG00000022026	olfactomedin 4
1419709_at	20863	Stfa3	ENSMUSG00000054905	stefin A3
1447806_s_at	56504	Srpk3	ENSMUSG00000002007	serine/arginine-rich protein specific kinase 3
1449366_at	17394	Mmp8	ENSMUSG00000005800	matrix metallopeptidase 8
1427747_a_at	16819	Lcn2	ENSMUSG00000026822	lipocalin 2
1440865_at	213002	Ifitm6	ENSMUSG00000059108	interferon induced transmembrane protein 6
1418722_at	18054	Ngp	ENSMUSG00000032484	neutrophilic granule protein

Tab 1. Obtención del listado de genes. Clasificados según su grupo de tratamiento. Untreated: sin tratar; Linezolid: tratados con el antibiótico linezolid; Vancomicina: tratados con el antibiótico vancomicina. Cada columna corresponte a un identificador distinto, más la columna Descripción que especifica el nombre completo del gen en cuestión. Resaltados en verde los genes de expresión única para cada grupo.

Tras realizar un análisis de los grupos de genes con mayor variabilidad en su expresión génica según el tratamiento recibido, obtenemos un conjunto de imágenes, donde se resaltan los procesos biológicos a los que pertenecen dichos genes (Fig 4). En el grupo que no ha recibido tratamiento la mayoria de los genes han estado implicados en la regulación de la respuesta inflamatoria (Fig 4 A).

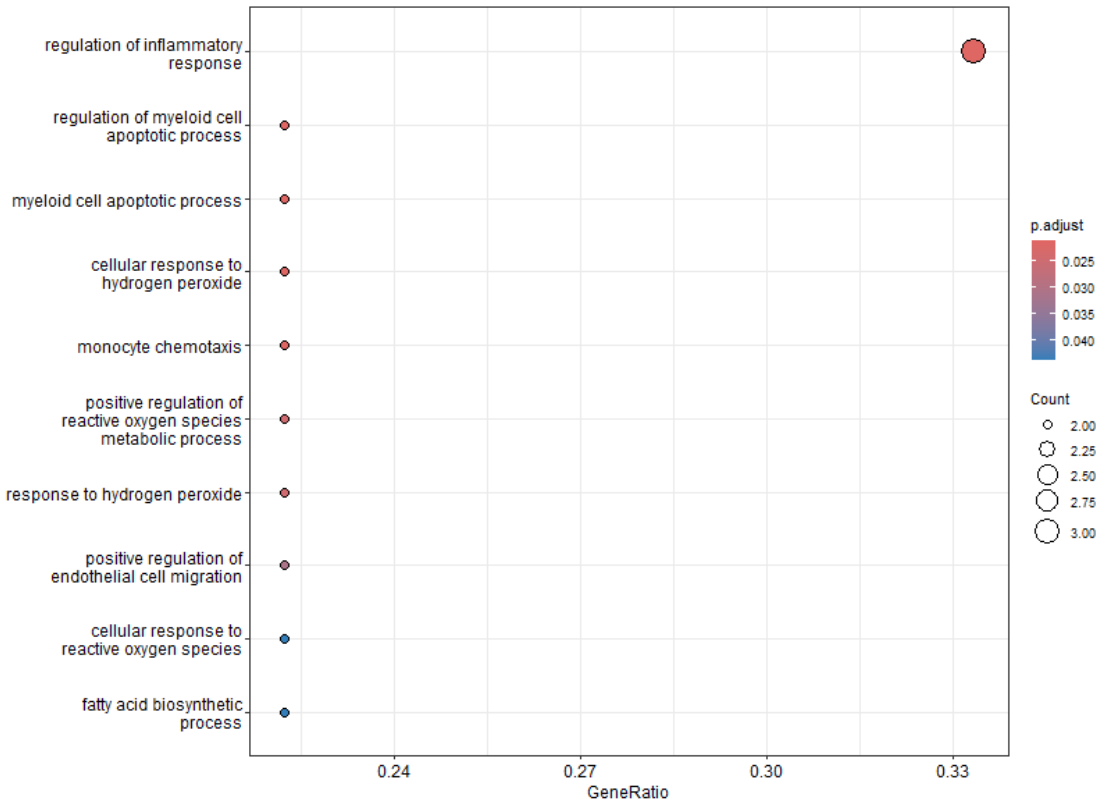

Fig 4 A. Procesos biológicos en los que están implicados los genes con mayor variabilidad en su expresión génica, grupo sin tratamiento.

En el grupo que han recibido tratamiento LNZ, la mayoria de los genes han estado implicados en la regulación del proceso metabólico de las especies reactivas de oxígeno y la señal de transducción canónica de NF-kappaB, aún que los valores p ajustados no han sido estrictamente significativos (Fig 4 B).

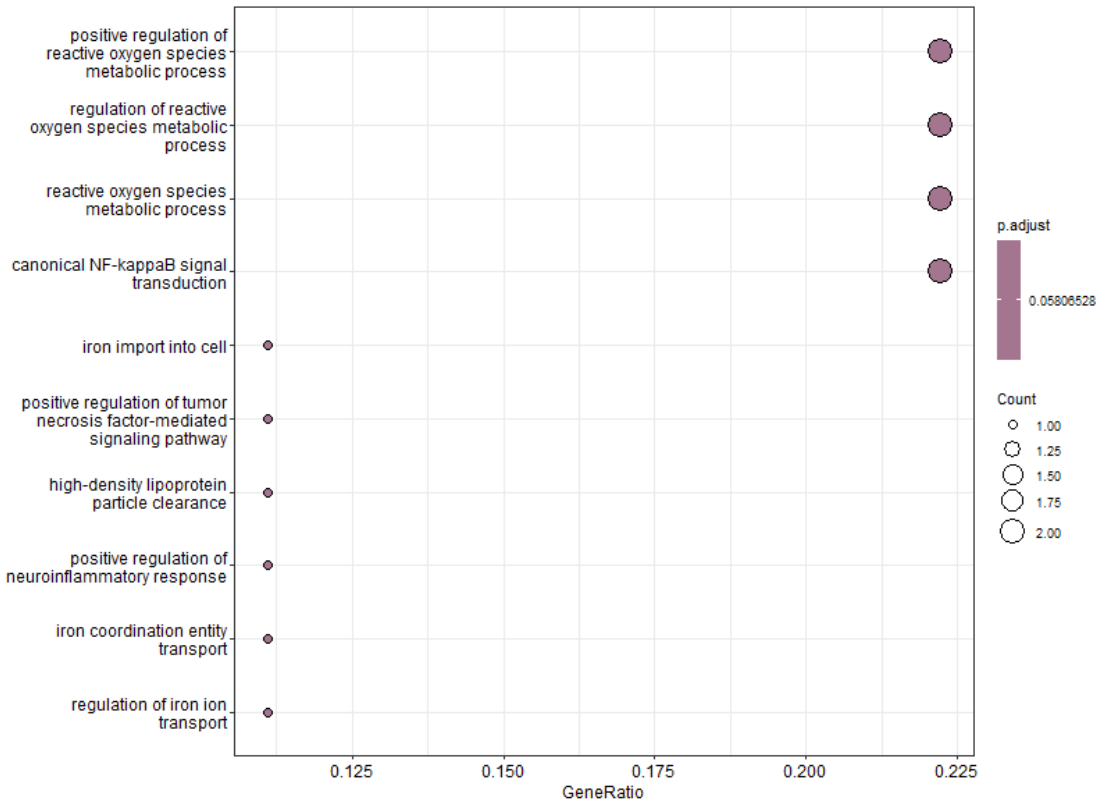

Fig 4 B. Procesos biológicos en los que están implicados los genes con mayor variabilidad en su expresión génica, grupo LNZ.

En el grupo que han recibido tratamiento VAN, la mayoria de los genes han estado implicados en la regulación negativa de la peptidasa, hidrolasa y proteólisis, aún que los valores p ajustados no han sido estrictamente significativos (Fig 4 C).

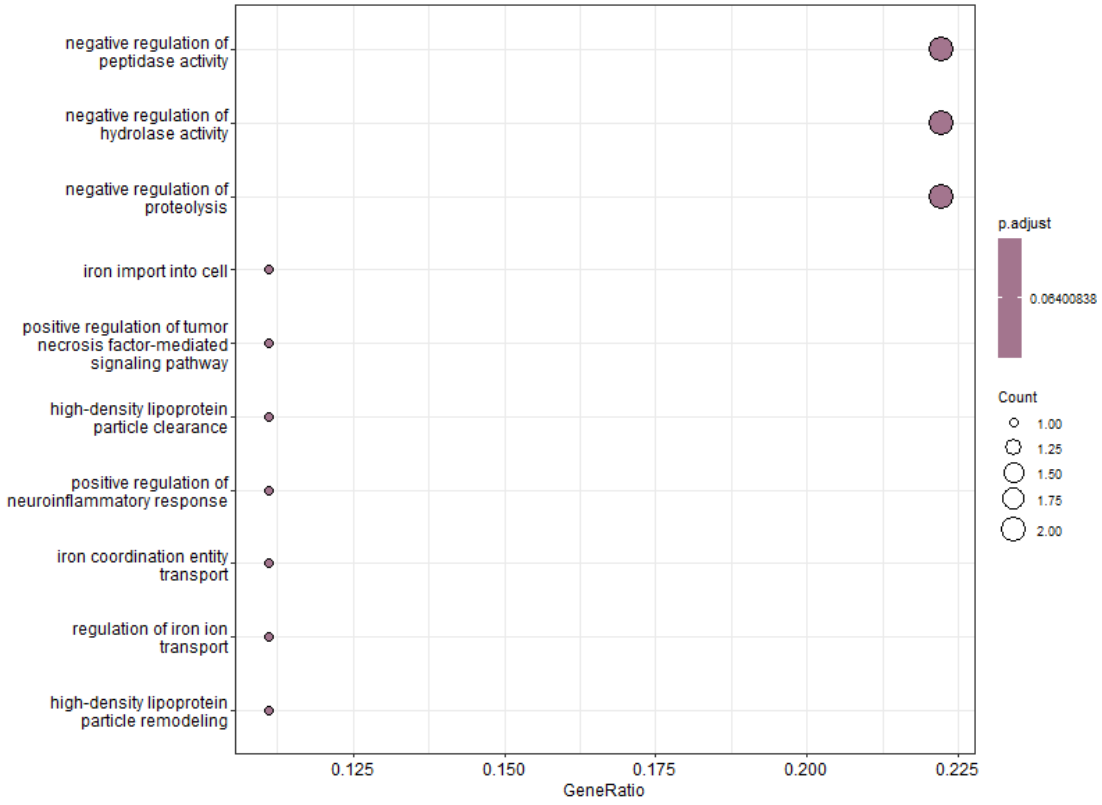

Fig 4 C. Procesos biológicos en los que están implicados los genes con mayor variabilidad en su expresión génica, grupo VAN.

4 Discusión

La normalización realizada con el método RMA fue efectiva en minimizar la variabilidad técnica entre las muestras. Bolstad et al. (2003) Según el análisis de los PCA las diferencias en la expresión génica más significativas están entre los grupos de ratones no infectados e infectados con la cepa MRSA. En general, los resultados obtenidos tras el análisis de calidad de los datos fueron positivos.

Respecto a los genes únicos de expresión alterada según su grupo de tratamiento, encontramos que la expresión del gen *Serum Amyloid A3* (Saa3) se encuentra alterada en el grupo de pacientes tratados con LNZ, esto es possible ya que LNZ inhibe la síntesis de proteínas bacterianas y la producción de toxinas, pudiendo reducir el estímulo inflamatorio en el huésped. Esto, a su vez, puede disminuir la activación de las vías de señalización que inducen la expresión de genes como Saa3. En otras palabras, al eliminar el estímulo infeccioso y reducir la inflamación sistémica, la expresión de Saa3 podría disminuir.@Yan2024

La expresión de los genes stefin A3 (Stfa3) y serine/arginine-rich protein-specific kinase 3 (Srpk3) se encuentra significativamente más alterada en el grupo de pacientes tratados con VAN. Si la VAN induce un aumento en la producción de toxinas bacterianas, es probable que se genere un entorno inflamatorio más intenso, lo que alteraría la expresión de genes del huésped relacionados con la inflamación, el estrés celular y la defensa tisular, como Stfa3 y Srpk3.@Evans2019

En cuanto a los procesos biológicos, la agrupación y representación de los genes implicados mediante un dotplot resulta una herramienta eficaz para interpretar lo que ocurre dentro de cada grupo de tratamiento. En el grupo sin tratamiento, es habitual observar una gran variabilidad en la expresión de genes relacionados con la regulación de procesos inflamatorios entre los individuos infectados y no infectados, lo cual se explica por el hecho de que algunos están desarrollando un proceso infeccioso, mientras que otros no.@Li2019

El hecho de que en el grupo tratado con LNZ la mayoría de los genes estén implicados en la regulación del proceso metabólico de las especies reactivas de oxígeno (ROS) y en la señal de transducción canónica de NF-kappaB tiene sentido debido a que LNZ impide la síntesis de proteínas bacterianas al bloquear la subunidad ribosomal 50S. Esto no solo reduce la proliferación bacteriana, sino también la liberación de toxinas que pueden exacerbar la respuesta inflamatoria en el huésped. Durante las infecciones bacterianas, los fagocitos (como macrófagos y neutrófilos) generan ROS como parte de su mecanismo para eliminar patógenos. En un entorno donde las toxinas bacterianas están reducidas, pero aún persisten estímulos inflamatorios, los genes relacionados con la regulación del metabolismo de ROS pueden activarse para equilibrar la producción y eliminación de estas moléculas y prevenir un daño oxidativo excesivo. La vía canónica de NF-kappaB es crucial en la respuesta inmune e inflamatoria. Se activa en presencia de estímulos como restos bacterianos, citoquinas inflamatorias (por ejemplo, TNF-α) o incluso daño tisular causado por la infección inicial. Aunque LNZ reduce la producción de toxinas bacterianas, el sistema inmune sigue respondiendo al daño previo o a los fragmentos bacterianos mediante la activación de NF-kappaB, que regula genes inflamatorios clave.@Zahedi2017

El hecho de que en el grupo tratado con VAN la mayoría de genes estén implicados en la regulación negativa de la peptidasa, hidrolasa y proteólisis, puede significar que el tratamiento con el antibiótico induce cambios en las vías del huésped destinadas a contrarrestar posibles daños tisulares asociados a la infección bacteriana. La regulación negativa de actividades como la de peptidasas, hidrolasas y procesos de proteólisis podría estar relacionada con un mecanismo protector para limitar la degradación excesiva de proteínas propias del huésped, un fenómeno que suele acompañar a respuestas inflamatorias exacerbadas.@Mühlberg2020

Tras el análisis realizado podríamos decir que LNZ no elimina inmediatamente todos los estímulos inflamatorios, más bien, al reducir la carga bacteriana, permite al sistema inmunológico manejar la infección de manera más controlada. La regulación de ROS y la activación moderada de NF-kappaB pueden ser señales de un equilibrio inmunológico, donde el huésped intenta resolver la inflamación sin desencadenar un daño excesivo a los tejidos. Aunque los genes con mayor variabilidad en el grupo tratado con LNZ no están directamente relacionados con la inhibición de la síntesis proteica de las toxinas de MRSA, su regulación parece reflejar un efecto

secundario positivo del tratamiento, enfocado en controlar la inflamación y minimizar el daño tisular causado por la infección.@Chavanet2013; Sharma-Kuinkel et al. (2013)

Tras el análisis realizado podríamos decir que VAN promueve una estrategia del organismo para mantener la integridad tisular y prevenir el daño asociado al exceso de enzimas proteolíticas, que son frecuentemente activadas o liberadas durante infecciones graves. Pero a su vez, esto podria implicar una tendencia a estimular la producción de toxinas de MRSA, debido a la regulación negativa de procesos proteolíticos que podrían haber degradado dichas toxinas, lo que coincide con la hipótesis inicial de que la vancomicina podria estar estimulando la producción de toxinas bacterianas.@Chavanet2013; Sharma-Kuinkel et al. (2013)

Finalmente, según los resultados obtenidos, podriamos confirmar la hipótesis planteada en el inicio de que LNZ genera un perfil de expresión génica diferente al de VAN en la sepsis por MRSA. Aún que tambíen podria resultar interesante evaluar en detalle si LNZ podría estar inhibiendo la síntesis de proteínas virulentas o toxinas, como la *leucocidina de Panton-Valentine* (PVL), mientras que el tratamiento con VAN podría estar estimulando su producción, para ello, sería ideal realizar un estudio detallado de la expresión génica de las sondas implicadas. Aunque el análisis realizado ha sido extenso, los filtros aplicados han priorizado las sondas con mayor variabilidad, lo que podría haber excluido algunas que, si bien no presentan un gran peso estadístico, podrían ser relevantes desde un punto de vista biológico. Porchera et al. (2024); Sharma-Kuinkel et al. (2013); Kato et al. (2021)

5 Apéndice

5.1 Apéndice. Código para la descarga y preparación de los datos

```
# Función para instalar paquetes si no están instalados
installifnot <- function(pckgName) {
  if (!requireNamespace(pckgName, quietly = TRUE)) {
    if (pckgName %in% rownames(available.packages())) {
      install.packages(pckgName)
    } else {
      if (!requireNamespace("BiocManager", quietly = TRUE)) {
        install.packages("BiocManager")
      }
      BiocManager::install(pckgName, ask = FALSE)
    }
  } else {
    message(paste("Package", pckgName, "is already installed."))
  }
}
```

```r
# Lista de paquetes
packages <- c(
  "Biobase", "affy", "arrayQualityMetrics", "genefilter",
  "limma", "hgu133a.db", "annotate", "annaffy", "hwriter",
  "gplots", "GOstats", "GSA", "umap", "biomaRt","clusterProfiler","org.Mm.eg.db")
for (pkg in packages) {
  installifnot(pkg)
}

require(affy)

# Leer los archivos .CEL desde la carpeta descomprimida
raw_data <- ReadAffy(celfile.path = "GSE38531_RAW")

# Renombrar las muestras en raw_data con los primeros 9 dígitos
sampleNames(raw_data) <- substr(sampleNames(raw_data), 1, 9)

# Normalización RMA
norm_data <- rma(raw_data)
```

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

```r
# Extraer la matriz de expresión normalizada
expr_matrix <- exprs(norm_data)

#Aplicar la función que se sugiere en el enunciado
filter_microarray <- function(allTargets, seed = 123) {
  # Configurar la semilla aleatoria
  set.seed(seed)

  # 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)
}


# 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("vancomyci
)

# Aplicar la función
result <- filter_microarray(allTargets, seed=76624687)
```

5.2 Apéndice. Código para el análisis exploratorio y control de calidad

```
require(limma)
require(umap)

# Reordenar las columnas de expr_matrix según el orden de result$sample
expr_matrix <- expr_matrix[, match(result$sample, colnames(expr_matrix))]
```

```r
# Eliminar columnas que no estén en result$sample
expr_matrix <- expr_matrix[, intersect(result$sample, colnames(expr_matrix))]

# Asignar colores según los tratamientos
treatments <- result$agent
unique_treatments <- unique(treatments)
treatment_colors <- rainbow(length(unique_treatments))
treatment_colors_assigned <- treatment_colors[match(treatments, unique_treatments)]

# Asignar colores según infectado o no
infected <- result$infection
unique_infected <- unique(infected)
infected_colors <- rainbow(length(unique_infected))
infected_colors_assigned <- infected_colors[match(infected, unique_infected)]

# Crear el directorio de salida para las figuras
output_dir <- "Created R figures"
if (!dir.exists(output_dir)) {
  dir.create(output_dir)
}

# Boxplot con colores de tratamiento
output_file1 <- file.path(output_dir, "boxplot.png")
title <- "Boxplot de expresión normalizada (RMA)"
dev.new(width = 3 + ncol(expr_matrix) / 6, height = 5)
par(mar = c(7, 4, 2, 1))
png(filename = output_file1, width = 800, height = 600)
boxplot(expr_matrix,
        boxwex = 0.7,
        notch = T,
        main = title,
        outline = FALSE,
        las = 2,
        col = treatment_colors_assigned,  #
        ylab = "Expresión Normalizada (RMA)",
        xlab = "Muestras")
legend("topright",
       legend = unique_treatments,
       fill = treatment_colors,
       title = "Treatment",
       cex = 0.8)
dev.off()
```

```
## pdf
##   2
```

```
# Gráfico de densidad de valores de expresión
output_file2 <- file.path(output_dir, "expression_value_distribution_plot.png")
par(mar = c(4, 4, 2, 1))
title <- "Expression Value Distribution Plot"
png(filename = output_file2, width = 800, height = 600)
plotDensities(expr_matrix, main = title, legend = F)
dev.off()
```

```
## pdf
##   2
```

```
# UMAP plot (multi-dimensional scaling) con colores por infección
output_file3 <- file.path(output_dir, "UMAP_plot_multi_dimensional_scaling.png")
expr_matrix_nodupli <- expr_matrix[!duplicated(expr_matrix), ]  # Eliminar duplicados
ump <- umap(t(expr_matrix_nodupli), n_neighbors = 15, random_state = 123)
png(filename = output_file3, width = 800, height = 600)
plot(ump$layout,
     main = "UMAP plot, nbrs = 15",
     xlab = "",
     ylab = "",
     pch = 20,
     cex = 1.5,
     col = infected_colors_assigned)
legend("bottomright",
       legend = unique_infected,
       fill = infected_colors,
       title = "Infected",
       cex = 0.8)
dev.off()
```

```
## pdf
##   2
```

```
# Crear el gráfico de clustering
output_file4 <- file.path(output_dir, "hierarchical_clustering.png")
# Calcular la distancia y realizar el clustering jerárquico
clust.euclid.average <- hclust(dist(t(expr_matrix)), method = "average")
png(filename = output_file4, width = 800, height = 600)
plot(clust.euclid.average, labels = colnames(expr_matrix),
```

```
     main = "Hierarchical clustering of samples", hang = -1, cex = 0.7)
dev.off()
```

```
## pdf
##   2
```

```
#análisis de calidad complementario con arrayqualitymetrics
require(Biobase)

colnames(expr_matrix) <- result$sample
rownames(result) <- result$sample
result_df <- AnnotatedDataFrame(data = result)
# Crear el ExpressionSet
eset <- ExpressionSet(assayData = expr_matrix, phenoData = result_df)

require(arrayQualityMetrics)

arrayQualityMetrics(expressionset = eset,
                    outdir = "Informe_de_calidad_para_los_datos_RAM",
                    force = TRUE)
```

5.3 Apéndice. Código para el filtrado de los datos

```
# Calcular la desviación estándar de cada fila
row_sd <- apply(expr_matrix, 1, sd)

# Determinar el 10% superior basado en la desviación estándar
top_10_percent_indices <- order(row_sd, decreasing = TRUE)[1:ceiling(0.1 * nrow(expr_mat

# Filtrar la matriz original
expr_matrix_hSD <- expr_matrix[top_10_percent_indices, ]
```

5.4 Apéndice. Código para la construcción de las matrices de diseño y de contrastes

```
library(limma)

#Creación de la matriz de diseño
```

```
result$Group <- interaction(result$infection, result$agent)
design <- model.matrix(~ 0 + Group, data = result)
colnames(design) <- levels(result$Group)

#Creación de la matriz de contraste
colnames(design) <- make.names(colnames(design))
contrast_matrix <- makeContrasts(
  Infectados_vs_NoInfectados_SinTratamiento = S..aureus.USA300.untreated - uninfected.unt
  Infectados_vs_NoInfectados_Linezolid = S..aureus.USA300.linezolid - uninfected.linezoli
  Infectados_vs_NoInfectados_Vancomicina = S..aureus.USA300.vancomycin - uninfected.vanco
  levels = design
)
```

5.5 Apéndice. Obtención de las listas de genes diferencialmente expresados para cada comparación

Utilizad limma para obtener una lista de genes diferencialmente expresados, siguiendo los ejemplos presentados en las notas y los casos resueltos. Las comparaciones entre las listas de genes la podéis hacer gráficamente o usando la función decideTests de limma.

```
require(limma)

fit <- lmFit(expr_matrix_hSD, design) #Ajustar el modelo lineal con la matriz de diseño
fit.main <- contrasts.fit(fit, contrast_matrix) #Aplicar los contrastes
fit.main <- eBayes(fit.main) #Ajustar los estadísticos con eBayes

# Obtener listas de genes para cada comparación (se aplica topTable para obtener aquello.
genes_untreated <- topTable(fit.main, coef = "Infectados_vs_NoInfectados_SinTratamiento"
genes_linezolid <- topTable(fit.main, coef = "Infectados_vs_NoInfectados_Linezolid", adju
genes_vancomicina <- topTable(fit.main, coef = "Infectados_vs_NoInfectados_Vancomicina",

res<-decideTests(fit.main, method="separate", adjust.method="fdr", p.value=0.01, lfc=1)
sum.res.rows<-apply(abs(res),1,sum)
res.selected<-res[sum.res.rows!=0,]

res<-decideTests(fit.main, method="separate", adjust.method="fdr", p.value=0.05, lfc=1)
sum.res.rows<-apply(abs(res),1,sum)
res.selected<-res[sum.res.rows!=0,]


output_file5 <- file.path(output_dir, "Genes in common 1.png")
```

```
png(filename = output_file5, width = 800, height = 600)
vennDiagram (res.selected[,1:3], main="Genes in common 1", cex=0.9)
```

5.6 Apéndice. Código para la anotación de los genes

El análisis con limma nos arroja listad de identificadores basados en los identificadores originales. Con estas listas debéis anotarlos, es decir asociarles algún identificador como "Symbol", "EntrezID" o "EnsemblID"

```
library(biomaRt)
library(annotate)
# Conexión al biomart para el genoma de ratón
ensembl <- useMart("ensembl", dataset = "mmusculus_gene_ensembl")

# Anotación de las sondas en ENTREZ, HUGO, y ENSEMBL
genes_untreated$ProbeID <- rownames(genes_untreated)
annotated_untreated <- getBM(
    attributes = c("affy_mouse430_2", "entrezgene_id", "external_gene_name", "ensembl_ge
    filters = "affy_mouse430_2",
    values = genes_untreated$ProbeID,
    mart = ensembl
)
# Renombrar las columnas de la tabla anotada para untreated
colnames(annotated_untreated) <- c(
    "affy_mouse430_2 (Id original)",
    "Id ENTREZ",
    "Id HUGO",  # HUGO es el atributo "external_gene_name"
    "Id ENSEMBL",
    "descripcion"
)
# Anotación de las sondas en ENTREZ, HUGO, y ENSEMBL (L)
genes_linezolid$ProbeID <- rownames(genes_linezolid)
annotated_linezolid <- getBM(
    attributes = c("affy_mouse430_2", "entrezgene_id", "external_gene_name", "ensembl_ge
    filters = "affy_mouse430_2",
    values = genes_linezolid$ProbeID,
    mart = ensembl
)
# Renombrar las columnas de la tabla anotada para L
colnames(annotated_linezolid) <- c(
    "affy_mouse430_2 (Id original)",
    "Id ENTREZ",
```

```
    "Id HUGO",  # HUGO es el atributo "external_gene_name"
    "Id ENSEMBL",
    "descripcion"
)
# Anotación de las sondas en ENTREZ, HUGO, y ENSEMBL (V)
genes_vancomicina$ProbeID <- rownames(genes_vancomicina)
annotated_vancomicina <- getBM(
    attributes = c("affy_mouse430_2", "entrezgene_id", "external_gene_name", "ensembl_ge
    filters = "affy_mouse430_2",
    values = genes_vancomicina$ProbeID,
    mart = ensembl
)
# Renombrar las columnas de la tabla anotada para V
colnames(annotated_vancomicina) <- c(
    "affy_mouse430_2 (Id original)",
    "Id ENTREZ",
    "Id HUGO",  # HUGO es el atributo "external_gene_name"
    "Id ENSEMBL",
    "descripcion"
)
```

5.7 Apéndice. Código para el análisis de la significación biológica

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

#para grupo untreated
gene_list_untreated <- subset(annotated_untreated, select = `Id ENTREZ`)
topGenes <- gene_list_untreated$`Id ENTREZ`
ego <- enrichGO(
  gene = as.integer(topGenes),
  keyType = "ENTREZID",  #Usamos el identificador "ENTREZID"
  OrgDb =  org.Mm.eg.db,  #Usamos la base de datos para Mus musculus
  ont = "BP",              #Usamos la categoría "Biological Process" (BP)
  pAdjustMethod = "BH",     #Método de ajuste de p-valor (Benjamini-Hochberg)
  qvalueCutoff = 0.25,
  readable = TRUE            #Poner nombres de genes en lugar de IDs
)
output_file6 <- file.path(output_dir, "dotplot bio (untreated).png")
```

```r
png(filename = output_file6, width = 800, height = 600)
dotplot(ego, showCategory=10)
dev.off()
```

```
## pdf
##   2
```

```r
output_file6a <- file.path(output_dir, "cnetplot bio (untreated).png")
png(filename = output_file6a, width = 800, height = 600)
cnetplot(ego)
dev.off()
```

```
## pdf
##   2
```

```r
# Para el grupo Linezolid
gene_list_linezolid <- subset(annotated_linezolid, select = `Id ENTREZ`)
topGenesL <- gene_list_linezolid$`Id ENTREZ`

# Realizar el análisis de enriquecimiento de GO
egoL <- enrichGO(
  gene = as.integer(topGenesL),
  keyType = "ENTREZID",
  OrgDb = org.Mm.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.06, #ajustado devido a que no hay términos con un pvalor ajustado inf
  qvalueCutoff = 0.25,  # Este es el umbral de q-value para el análisis inicial
  readable = TRUE
)

# Generar el gráfico dotplot con los primeros 10 términos más significativos
output_file7 <- file.path(output_dir, "dotplot bio (linezolid).png")
png(filename = output_file7, width = 800, height = 600)
dotplot(egoL, showCategory=10)
dev.off()
```

```
## pdf
##   2
```

```r
output_file7a <- file.path(output_dir, "cnetplot bio (linezolid).png")
png(filename = output_file7a, width = 800, height = 600)
cnetplot(egoL)
dev.off()
```

```
## pdf
##   2
```

```r
#para grupo v
gene_list_vancomicina <- subset(annotated_vancomicina, select = `Id ENTREZ`)
topGenesV <- gene_list_vancomicina$`Id ENTREZ`
egoV <- enrichGO(
  gene = as.integer(topGenesV),
  keyType = "ENTREZID",
  OrgDb =  org.Mm.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.07, #ajustado devido a que no hay términos con un pvalor ajustado inf
  qvalueCutoff = 0.25,
  readable = TRUE
)
# Generar el gráfico dotplot con los primeros 10 términos más significativos
output_file8 <- file.path(output_dir, "dotplot bio (vancomicina).png")
png(filename = output_file8, width = 800, height = 600)
dotplot(egoV, showCategory=10)
dev.off()
```

```
## pdf
##   2
```

```r
output_file8a <- file.path(output_dir, "cnetplot bio (vancomicina).png")
png(filename = output_file8a, width = 800, height = 600)
cnetplot(egoV)
dev.off()
```

```
## pdf
##   2
```

Referencias

Bolstad, B. M., R. A Irizarry, M. Åstrand, and T. P. Speed. 2003. "A Comparison of Normalization Methods for High Density Oligonucleotide Array Data Based on Variance and Bias." *Bioinformatics* 19 (2): 185–93. https://doi.org/10.1093/bioinformatics/19.2.185.

Gautier, Laurent, Leslie Cope, Benjamin M. Bolstad, and Rafael A. Irizarry. 2004. "Affy—Analysis of Affymetrix GeneChip Data at the Probe Level" 20. https://doi.org/10.1093/bioinformatics/btg405.

Kato, Hideo, Mao Hagihara, Nobuhiro Asai, Yuichi Shibata, Yusuke Koizumi, Yuka Yamagishi, and Hiroshige Mikamo. 2021. "Meta-Analysis of Vancomycin Versus Linezolid in Pneumonia with Proven Methicillin-Resistant Staphylococcus Aureus." *Journal of Global Antimicrobial Resistance* 24 (March): 98–105. https://doi.org/10.1016/j.jgar.2020.12.009.

Kauffmann, Audrey, Robert Gentleman, and Wolfgang Huber. 2009. "arrayQualityMetrics–a Bioconductor Package for Quality Assessment of Microarray Data" 25.

Konopka, Tomasz. 2023. "Umap: Uniform Manifold Approximation and Projection." https://CRAN.R-project.org/package=umap.

Porchera, Bruno Russo, Carolina Moraes da Silva, Rayssa Pinheiro Miranda, Antônio Rafael Quadros Gomes, Pedro Henrique dos Santos Fernandes, Camili Giseli Oliveira de Menezes, Paula do Socorro de Oliveira da Costa Laurindo, Maria Fani Dolabela, and Heliton Patrick Cordovil Brígido. 2024. "Linezolid and Vancomycin for Nosocomial Infections in Pediatric Patients: A Systematic Review." *Jornal de Pediatria* 100 (3): 242–49. https://doi.org/10.1016/j.jped.2023.08.011.

Ritchie, Matthew E, Belinda Phipson, Di Wu, Yifang Hu, Charity W Law, Wei Shi, and Gordon K Smyth. 2015. "{Limma} Powers Differential Expression Analyses for {RNA}-Sequencing and Microarray Studies" 43: e47. https://doi.org/10.1093/nar/gkv007.

Sharma-Kuinkel, Batu K., Yurong Zhang, Qin Yan, Sun Hee Ahn, and Vance G. Fowler. 2013. "Host Gene Expression Profiling and In Vivo Cytokine Studies to Characterize the Role of Linezolid and Vancomycin in Methicillin-Resistant Staphylococcus Aureus (MRSA) Murine Sepsis Model." Edited by Karsten Becker. *PLoS ONE* 8 (4): e60463. https://doi.org/10.1371/journal.pone.0060463.