Informe PEC2

Maria Luisa Reyes Conde

Contents

1 Resumen

En este informe presentamos un análisis sobre la expresión génica diferencial usando datos transcriptómicos de sangre periférica de pacientes que se infectaron por SARS-CoV-2 en comparación con personas con infecciones bacterianas respiratorias (como son el coronavirus estacional, la gripe y la neumonía bacteriana) y con controles sanos. Para llevarlo a cabo, hemos obtenido los datos del repositorio GEO (GSE161731). En el análisis, tras un completo preprocesamiento de los datos, usamos el *pipeline voom+limma*. Gracias a este, identificamos genes que están diferencialmente expresados en las comparaciones de pacientes con COVID-19 y los sanos y de aquellos con infecciones bacterianas respiratorias y sanos. La sobre-representación que se analizó nos proporciona algunos procesos biológicos que se ven enriquecidos al activarse la respuesta inmunitaria de aquellas personas que padecieron COVID-19. Los resultados se visualizaron mediante gráficos de PCA, Diagramas de Venn y análisis funcional con GO:BP.

2 Objetivos

Los objetivos para obtener un buen análisis sobre la respuesta inmunológica del huésped a la infección por el SARS-CoV-2 en comparación con las infecciomnes bacterianas respiratorias y con los controles sanos son los siguientes:

- Identificar aquellos genes que estén diferenciamente expresados para poder asociar perfiles moleculares específicos en moemntos de la enfermedad dada.
- Evaluar la calidad de los datos, eliminar los datos que no nos aportan informción o que nos complican el análisis para posteriormente ver si las muestras que tenemos siguen algún patrón. Además, esto nos proporciona información sobre si existen variables confusoras para poder ajustar los modelos y que no influyan de manera "errónea" en la expresión génica.
- Mediante análisis funcionales, buscar qué procesos y funciones biológicas están más implicados en cada condición que se estudia.
- Comparar los perfiles trasncriptómicos entre los pacientes con las distintas condiciones.
- Visualizar de manera interpretativa los resultados obtenidos, de modo que nos ayuden a proponer posibles biomarcadores para distintos escenarios como sería la propuesta de biomarcadores diagnósticos

3 Métodos

Dividimos esta sección para poder comentar con claridad los métodos usdos en este análisis.

3.1 Origen y naturaleza de los datos

Los datos usados para este análisis corresponden a la expresión génica obtenidos mediante secuenciación ARN (es decir, *RNA-seq*) de muestras de sangre periférica de individuos incluidos en el estudio GSE161731, que se obtuvieron del repositorio público *Gene Expression Omnibus* (GEO). Este estudio, titulado "*Dysregulated transcriptional responses to SARS-CoV-2 in the periphery support novel diagnostic approaches*", compara los perfiles transcriptómicos de pacientes con SARS-Cov-2 (COVID-19), infecciomnes bacterianas respiratorias y controles sanos.
Estos datos fueron descargados en R usando el paquete `GEOquery` y la función "`getGEOSuppFiles`" para poder extraer los archivos correspondientes y poder usar la matriz de expresión y los metadatos clínicos de los pacientes del estudio.

3.2 Preprocesamiento y selección de muestras

A partir de nuestro conjunto de datos completo, se seleccionaron tres cohortes: *COVID-19*, *Bacterial* y *Healthy* para posteriormente, poder realizar un análisis más específico en nuestro método de estudio. Una vez fueron seleccionadas estas cohortes, se eliminaron todas aquellas entradas que se encontraban duplicadas y nos aseguramos que entre los metadatos y la matriz de expresión existiera una correspondencia entre las muestras.
Una vez conseguido esto, seleccionamos 75 muestras de forma aleatoria usando una semilla personalizada, siguiendo el siguiente código:

```
myseed <- sum(utf8ToInt("marialuisareyesconde"))
set.seed(myseed)
muestras_seleccionadas <- sample(rownames(metadatos), 75)
```

Una vez nos quedamos con las 75 muestras con las que haremos el análisis, las volvemos a filtrar para eliminar cualquier gen con expresión baja: por ejemplo, nos quedamos con los genes con al menos 10 counts en 10% de las muestras.
Posteriormente, cargando la librería `limma`, pudimos aplicar una transformación `voom`, lo que nos

permite adaptar el análisis lineal a los datos de conteo con estimación de la variabilidad dependiente de la media. Esta transformación proporciona pesos de precisión para cada gen, que son incorporados en el modelo lineal general posterior. Los valores transformados fueron almacenados como una nueva matriz dentro del objeto `SummarizedExperiment`.

3.3 Análisis

Para visualizar la estructura de los datos y detectar si hay algún outlier se aplicó un análisis exploratorio mediante análisis de componentes principales (PCA) y de escalamiento multidimensional (MDS). Para la visualización de los gráficos de estos análisis se usa el paquete `ggplot2`. Gracias a estos gráficos, se pudo hacer una identificación de variables confusoras, que son las que posteriormente se añaden a la matriz de diseño del análisis diferencial junto con variable de estudio (*cohort*). Gracias al enfoque `voom + limma`, que se determinó usando el siguiente código

```
set.seed(myseed)
metodos <- sample(c("edgeR", "voom+limma", "DESeq2"), size = 1)
```

se realizaron comparaciones de expresión génica entre los pacientes con COVID-19 vs los controles sanos y entre los pacientes con infecciomnes bacterianas respiratorias vs los sanos. El modelo lineal se ajusta para cada gen y los contrastes específicos se definen con el comando `makeContrasts()`. Se consideraron que los genes significativos son aquellos que superan un umbral de log2FC de 1.5 y que poseen un ajuste de p-valor FDR superior a 0.05. Por ejemplo, el código para el contraste COVID-19 vs controles sería:

```
#logFC > 1.5 como umbral)
res_covid <- topTable(ajuste2, coef = "COVID19_vs_healthy", number = Inf,
                      adjust = "fdr", lfc = log2(1.5))
#significativos
sig_covid <- subset(res_covid, adj.P.Val < 0.05)
```

Por último, una vez obtenidos los genes significativos para cada contraste, se compararon dichos contrastes para detectar los posibles solapamientos y las diferencias que existan entre las respuestas al COVID-19 y a las diferentes infecciones bacterianas. Para visualizar estos contrastes, se usa un diagrama de Venn, graficado gracias a la librería `VennDiagram`.

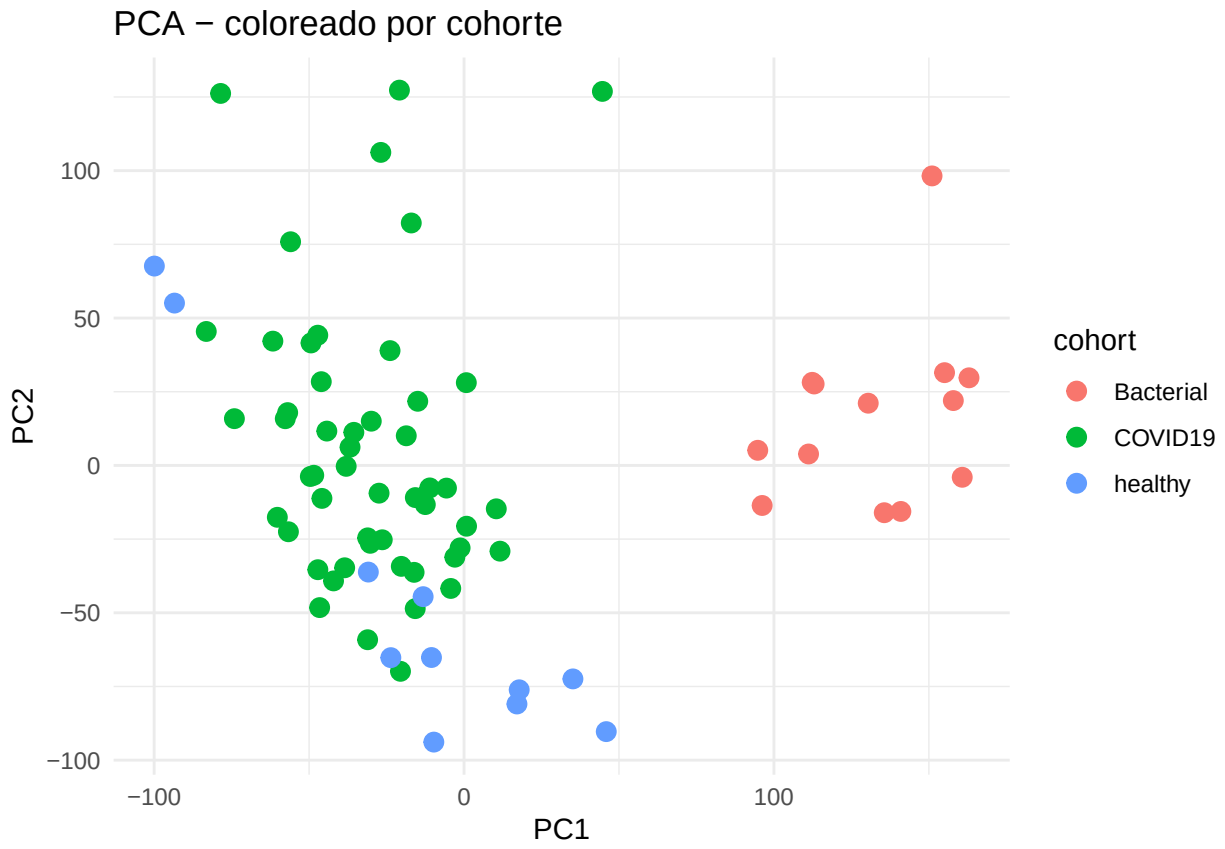
3.4 Enriquecimiento funcional

Una vez realizado todo el análisis y determinados los los genes sobreexpresados en el contraste COVID-19 vs controles sanos, se realizó un análisis de sobre-representación (ORA) para identificar las funciones enriquecidas entre los genes sobreexpresados en pacientes de SARS-CoV-2. Se cargaron las librerías `clusterProfiler`, para realizar el análisis *GO Biological Process* (GO:BP), y `org.Hs.eg.db` para poder mapear los ID de los genes a términos biológicos. La visualización de estos resultados se realizó mediante un `barplot` o gráfico de barras.

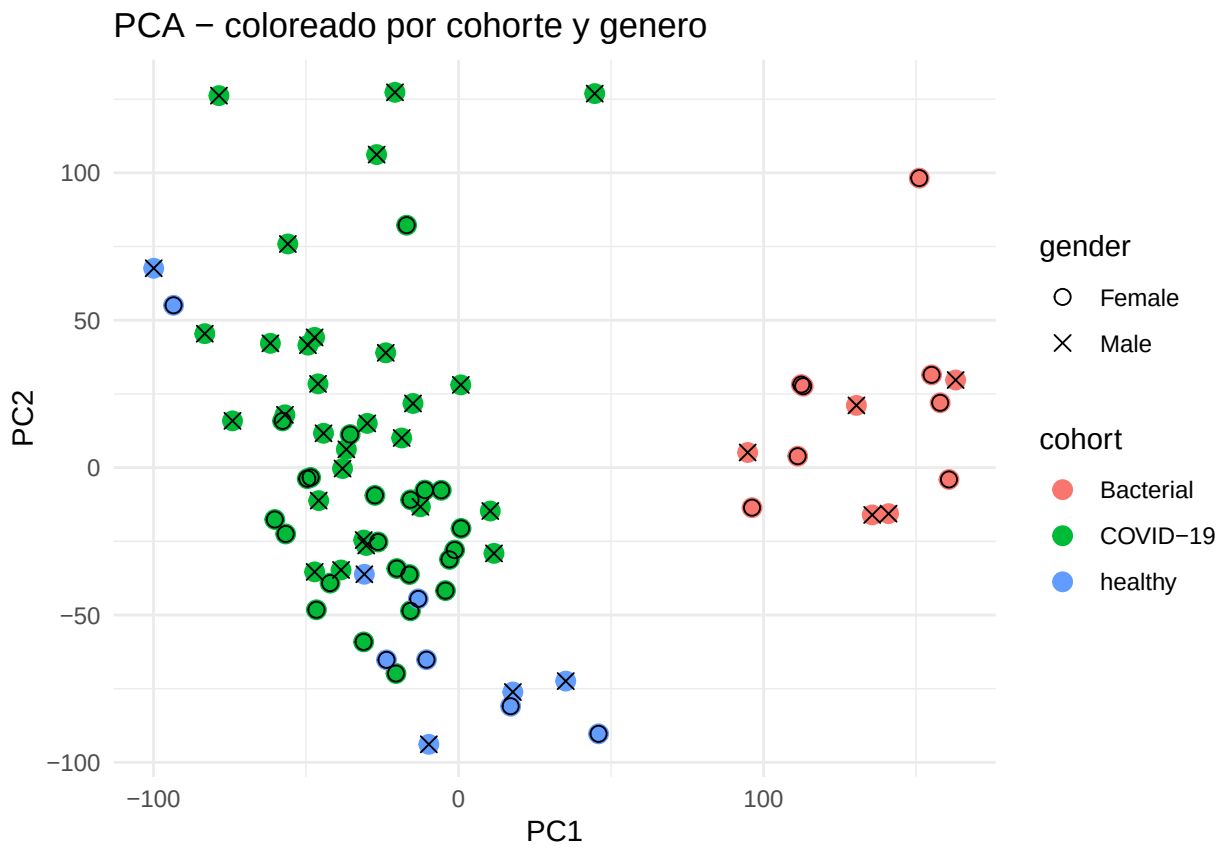
4 Resultados

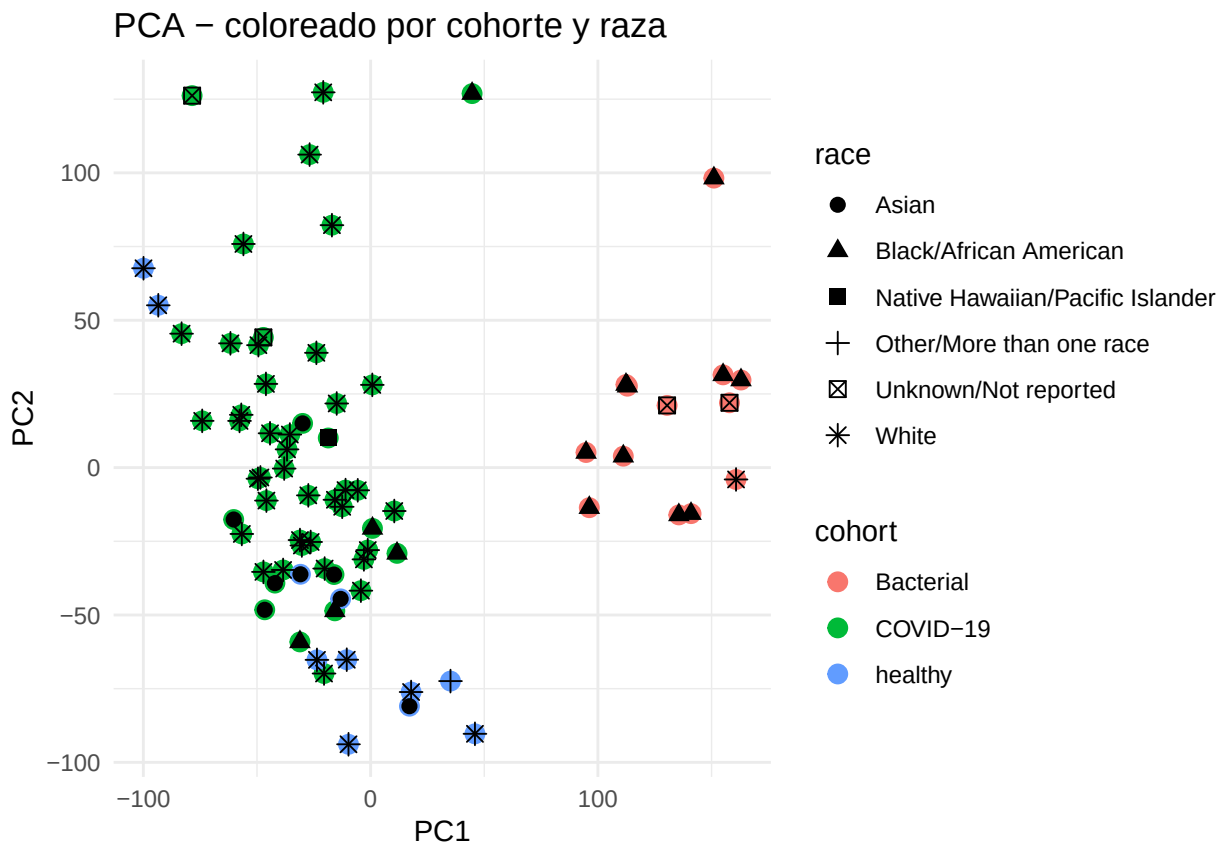
Como se ha comentado en el apartado de "Métodos", del dataset original se seleccionaron 75 muestras de forma aleatoria de las 3 cohortes que se escogieron (*COVID-19*, *Bacterial* y *Healthy*). Además se eliminaron los genes con baja expresión, y aún así nos quedamos con una gran cantidad

de genes para el posterior análisis.
Para llevar a cabo el análisis exploratoro sobre los datos filtrados, transformados y normalizados, se realizó un análisis de componentes principales, obteniendo la siguiente gráfica del PCA:



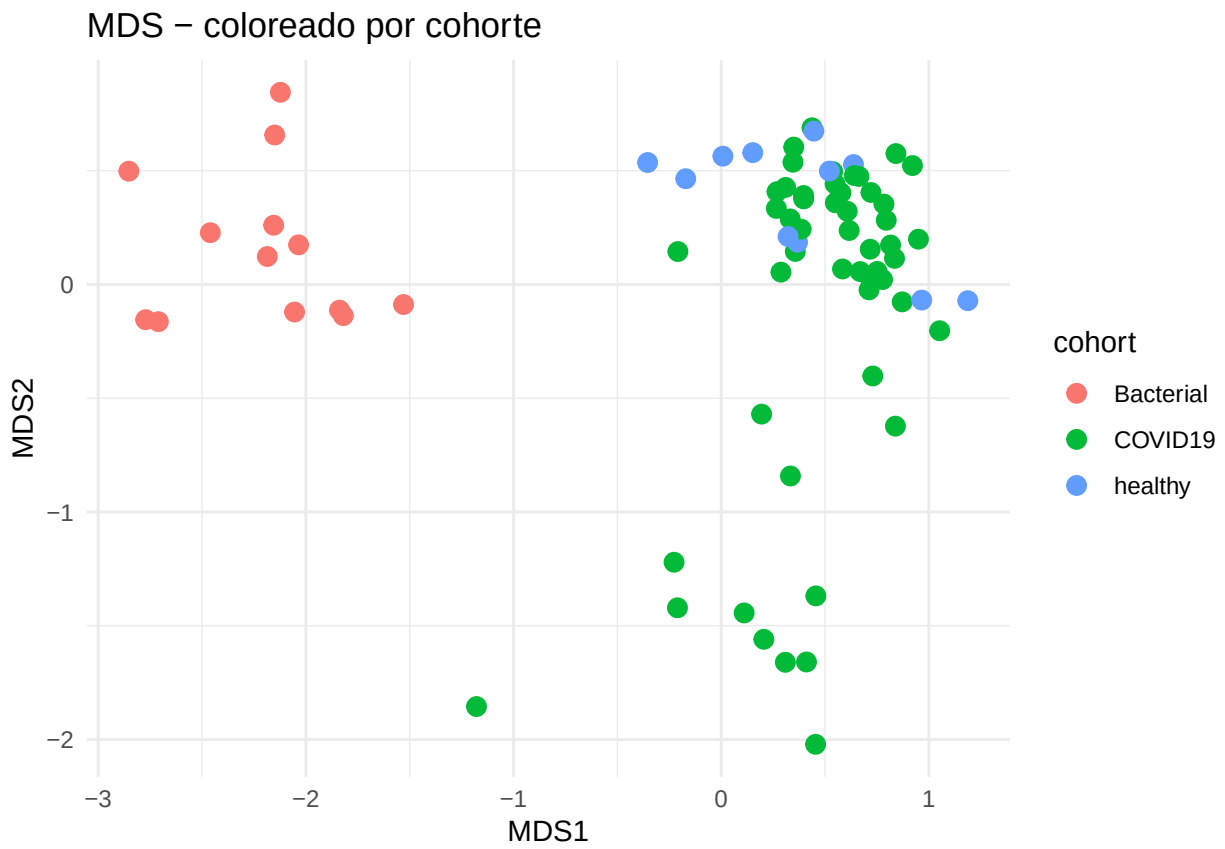
Como podemos observar, hay una separación clara entre los perfiles biológicos de las muestras de los pacientes que sufren de infecciones bacterianas respiratorias con respecto a los de COVID-19 y a los controles sanos. Aunque los controles sanos también parezcan un poco separados, se observa que hay superposición con las muestras de COVID-19, lo que nos indica que sus perfiles son más similares, al menos en los componentes principales 1 y 2.
Además de ver el PCA "simple", comparamos las cohortes con otras variables de nuestro dataset, como son el género y la raza:


PCA − coloreado por cohorte y genero
PC2
PC1
100
50
0
−50
−100
−100
0
100
gender
Female
Male
cohort
Bacterial
COVID−19
healthy



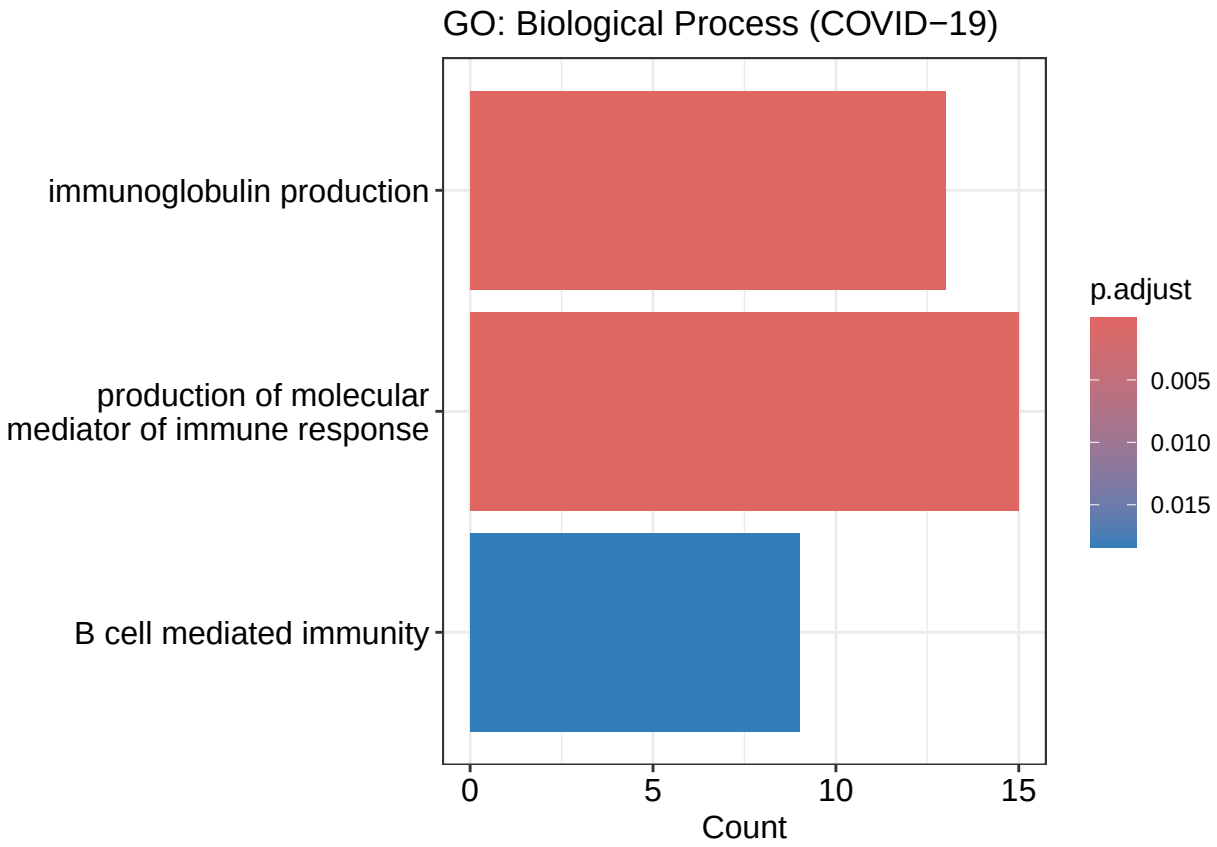
En el PCA cohorte-género vemos que no hay una distribución clara, es decir, las muestras de cada tipo de cohorte son tanto de hombres como de mujeres. En el caso del PCA cohorte-raza, observamos que la mayoría de los casos recogidos de COVID-19 y controles sanos son de personas "blancas". En cambio, en el caso de las muestras de infecciones bacterianas respiratorias, parece que destaca el caso de personas asiáticas. Estas dos variables (género y raza) las incluimos como variables confusoras en la matriz de diseño.

Además de un PCA, para el análisis exploratorio se graficó un plot de MDS (escalamiento multidimensional), en el que se observa, al igual que en el PCA, una clara separación entre las muestras bacterianas y las de COVID-19 y controles sanos, teniendo estas dos algunas muestras superpuestas:



Posterior al análisis exploratorio, creamos la matriz de diseño y las matrices de contrastes adecuadas para evaluar la expresión génica diferencial en las comparaciones *Bacterial vs healthy* y *COVID-19 vs healthy* y se realiza un análisis de expresión diferencial. Una vez que se realizaron estos contrastes, se usó un DIagrama de Venn para observar los genes significativos en ambos contrastes y los "exclusivos" de cada uno:

Como observamos, hay 2646 genes que se diferencian de manera significativa en ambos contrastes. Esto se puede deber a la respuesta inmune compartida que existe sobre el COVID-19 y las infecciones bacterianas respiratorias, que spoiblemente están relacionadas con respuestas generales a patógenos. Además de los genes compartidos, vemos que 8416 son exclusivos de la cohorte *Bacterial* y 1577 lo son de la cohorte *COVID-19*. Que en las ifnecciones bacterianas haya más genes significativamente expresados a la hora de compararlo con los controles sanos nos puede indicar que la respuesta transcriptómica es más amplia o distinta. Esto tiene sentido si observamos el PCA, ya que vemos que hay una diferencia más clara en los componentes principales del grupo *Bacterial* respecto del grupo de los controles sanos (*Healthy*). Los genes exclusivos de cada grupo nos pueden servir como biomarcadores específicos para poder distinguir entre la neumonía bacteriana y el SARS-CoV-2. Con los genes compartidos se pueden entender mecanismos comunes entre estas enfermedades.
Por último, una vez se hicieron las comparaciones, se realizó un análisis de sobrerepresentación para identificar las funciones enriquecidas entre los genes sobreexpresados en pacientes con COVID-19 en comparación con los controles sanos. Para ello se usa el análisis GO:BP (*GO Biological Process*) y representamos los resultados mediante un gráfico de barras en el que el en Eje Y nos encontramos los términos GO relaciondos con procesos biológicos relevantes en los que se encuentran invlocurados los genesque estudiamos y en el Eje X, el número de genes sobreexpresados a estos términos.



Observando este gráfico, podemos determinar las siguientes conclusiones:

- Hay aproximadamente 13 genes implicados en la síntesis de anticuerpos (producción de inmunoglobulinas), lo cual es una respuesta clave del sistema inmunológico humoral (la respuesta inmune mediada por anticuerpos). La sobreexpresión de los genes en este aspecto nos sugiere que hay una fuerte activación de a respuesta inmune humoral que posiblemente se deba a un mecanismo de defensa contra el COVID-19.
- Hay 15 genes expresados en la prodcciónde moléculas mediadoras de la respuesta inmune (como las citoquinas o las quimioquinas). Esto es importante, ya que la sobreproducción de estas moléculas puede llegar sa agravar el estado clínico del paciente con SARS-CoV-2.
- Hay unos 9 genes sobreexpresados en la respuesta inmune mediada por linfocitos B, lo que incluye a la activación de estas células y su diferenciación en células plasmáticas que son productoras de anticuerpos.
- Con los colores determinamos la confianza estadística en que los procesos biológicos que se han idetificado no son aleatorios. En los dos primeros casos tenemos un p-valor muy bajo y por lo tanto son muy significativos. En el último, aunque el p-valor sea un poco mayor, es menos signficativo aunque sigue siendo un proceso muy relevante.

En conclusión, los genes sobreexpresados en pacientes con COVID-19 están significativamente dotados en procesos inmunológico, sobre todo en la producción de mediadores inmunes, en la producción de anticuerpos y en la inmunidad mediada por las células B (que son muy importantes en la memoria inmunológica y en la defensa específica).

5 Discusión

Con este estudio hemos podido comparar los perfiles de expresióon génica de pacientes con SARS-CoV-2, con neumonía bacteriana y controles sanos a partir de los datos transcriptómicos de muestras de sangre periférica. Gracias a un análisis completyo, que incluye desde la descarga de los datos hata el análisis de expresión diferencial y enriquecimiento funcional, hemos sido capaces de identificar las diferencias clave en la activación de la inmunidad que son específicas en cada tipo de enfermedad estudiada. Uno de los principales hallazgos ha sido la activación de la producción de mediadores inmunes y anticuerpos y en la inmunidad mediada por las células B.
Aún así, los hallazgos han sido descubiertos tras haber filtrado los datos originales a 3 cohortes y 75 muestras, lo cual hace que el dataset usado sea mucho más reducido que el original. Esto puede limitar la generalización de los resultados. Además, las muestras provienen de datos de sangre periférica, lo que puede limitar también los resutlados, ya que sería interesante conocer la respuesta inmune específica del tejido afectado por estas enfermedades, es decir, por el pulmón. También es necesario tener en cuenta las variables confusoras que se incluyeron en el modelo del estudio, que son género y raza. Sería conveniente considerar otros factores que no están presnetes en estos datos, como por ejemplo el tratamiento recibido por el paciente, que puede ser influyente en los perfiles de expresión génica.
Por último cabe destacar que, aunque se hayan obtenido resultados relevantes en el análisis de sobreexpresión de los genes, sería ideal realizar estudios funcionales o proteómicos que complementen dicho análisis y que confirmen que los procesos inmunológicos destacados son reales y correctos en el contexto clínico.

6 Conclusiones

Gracias al análisis de expresión génica diferencial hemos podido identificar un conjunto de genes diferencialmente expresados en pacientes con COVID-19 comparándolos con controles sanos, detacando sobre todo la respuesta inmunitaria humoral y la mediada por las células B. En comparación con este perfil, la respuesta inmunitaria de pacientes con infecciones bacterianas mostraron un perfil más enfocado en procesos inflamatorios específicos.
Cabe destacar la eficacia de la metodología usada, ya que gracias al RNA-seq y al análisis GO:BP ha sido sencillo caracterizar las respuestas inmunes en los tipos de infecciones respirstorias estudiadas. Los resutlados obtenidos aportan bastante información y nos pueden ser de gran ayuda para entender las bases moleculares de la respuesta inmune del COVID-19 y en un futuro, para mejorar los diagnósticos e identificar posibles biomarcadores.
Sin embargo y auqnue el análisis sea relevante, sería interesante realizar algún estudio futuro en el que se incluyan muestras de otro tipo y no solo de sangre perfiérica, más de 75 muestras de pacientes o incluyendo más variables confusoras a la hora del modelo. Así, podríamos obtener un estudio más amplio y que corrobore la información obtenida al realziar los análisis de este estudio.

7 Referencias

- Repositorio de GitHub: [PEC2-Análisis de Datos Ómicos - María Luisa Reyes Conde](https://github.com/marisireyes/Reyes-Conde-MariaLuisa-PEC2) [1]
- R Packages: `GEOquery, limma, edgeR, clusterProfiler, SummarizedExperiment` [2]
- https://www.lifeder.com/inmunidad-humoral/ [3]

8 Anexo

Código completo de R:

```
### PEC 2 - ANALISIS DE DATOS OMICOS - Msria Luisa Reyes Conde
## APARTADO 1
# librerias q nos serviran
library(GEOquery)
library(EnsDb.Hsapiens.v86)
library(SummarizedExperiment)
# Descarga la matriz de expresión correspondiente y los metadatos, y cárgalos en R.
#getGEOSuppFiles("GSE161731", makeDirectory = TRUE) #para extraer los archivos de GEO
# cargamos la matriz de expresiones
matriz_expr <- as.matrix(read.csv("GSE161731/GSE161731_counts.csv", row.names = 1, check.names
mode(matriz_expr) <- "numeric"
# metadatos
metadatos <- read.csv("GSE161731/GSE161731_key.csv", stringsAsFactors = FALSE)
rownames(metadatos)<-metadatos$rna_id

# Construye un objeto SummarizedExperiment que contenga ambos.
# Agrega también las coordenadas génicas como rowRanges. Ten en cuenta que necesitas tener mue
# y la matriz de expresión antes de crear el objeto SummarizedExperiment, así como genes en co
# la anotación (por ejemplo, EnsDb.Hsapiens.v86).

# como tenemos que asegurarnos de que las muestras coincidan entre los metadatos y la expresio
# que están tanto en la matriz de expresión como en los metadatos y "filtramos" estos dos conj
# comunes de la matriz y de los metadatos)
muestras_comunes <- intersect(colnames(matriz_expr), rownames(metadatos))
matriz_expr <- matriz_expr[, muestras_comunes]
metadatos <- metadatos[muestras_comunes, ]

# para obtener las coordenadas genicas de estas muestras usamos, como nos indica, la funcion "
coord_gen <- genes(EnsDb.Hsapiens.v86)
# para asegurarnos de que los genes de nuestra matriz de expresion estan incluidos en "coord_g
# nos quedamos con los genes que sean comunes:
genes_comunes <- intersect(rownames(matriz_expr), names(coord_gen))
matriz_expr <- matriz_expr[genes_comunes, ]
coord_gen <- coord_gen[genes_comunes]


# creamos ahora el summarizedExperiment
se <- SummarizedExperiment(
  assays = list(counts = matriz_expr),
  colData = metadatos,
  rowRanges = coord_gen
)
se
```

```r
## APARTADO 2
# Limpia los metadatos y selecciona solamente tres cohortes: COVID19, Bacterial y healthy.
cohortes <- c("COVID-19", "Bacterial", "healthy")
metadatos <- metadatos[metadatos$cohort %in% cohortes, ]

#  Sugerencias: (i) elimina individuos duplicados (conserva solo la primera entrada cuando hay
metadatos <- metadatos[!duplicated(rownames(metadatos)), ]

# (ii) asegúrate de que las variables sean del tipo adecuado (por ejemplo, la edad debe ser nu
metadatos$age <- as.numeric(metadatos$age)
metadatos$batch <- as.factor(metadatos$batch) #solo toma vslores 1 y 2


# Importante: una vez hecho esto, selecciona exclusivamente 75 muestras de manera aleatoria, u
# proporciona a continuación (basada en tu primer nombre, primer y segundo apellidos sin tilde
myseed <- sum(utf8ToInt("marialuisareyesconde"))
set.seed(myseed)
# seleccionamos las muestras ocn la semilla seleccionada y filtramos los metadatos
muestras_selec <- sample(rownames(metadatos), 75)
metadatos <- metadatos[muestras_selec, ]

# filtramos tambien la matriz y el se
matriz_expr <- matriz_expr[, muestras_selec]
se <- se[, muestras_selec]

# eliminamos guiones, barras y espacios

metadatos[] <- lapply(metadatos, function(x) {
  if (is.character(x) || is.factor(x)) {
    x <- as.character(x)           # convertir factor a carácter
    gsub("[- /]", "", x)            # eliminar guiones y espacios
  } else {
    x
  }
})

## APARTADO 3
library(DESeq2)
# creamos un objeto DESeqDataSet para poder hacer el analisis de expresion diferencial
dds <- DESeqDataSet(se, design = ~ cohort)
# Lleva a cabo el preprocesado inicial de los datos (eliminación de genes con baja expresión,
genes_buenos <- rowSums(counts(dds) >= 10) >= (0.1 * ncol(dds))
dds <- dds[genes_buenos, ]
# y la transformación/normalización que consideres apropiada.
vst_datos <- varianceStabilizingTransformation(dds, blind = TRUE)
vst_matriz <- assay(vst_datos) #matriz normalizada
```

```r
## APARTADO 4
library(ggplot2)
library(pheatmap)
library(limma)
# Realiza un análisis exploratorio sobre los datos transformados/normalizados.
metadatos <- as.data.frame(colData(se)) # metadatos transformados/normalizados
#Puedes usar PCA, MDS, clustering y/o heatmaps.
# PCA:
pca <- prcomp(t(vst_matriz), scale. = TRUE)
# agregamos los pc a los metadatos para graficarlo
metadatos$PC1 <- pca$x[,1]
metadatos$PC2 <- pca$x[,2]
# grafico coloreado x cohorte
p_cohort<-ggplot(metadatos, aes(x = PC1, y = PC2, color = cohort)) +
  geom_point(size = 3) +
  theme_minimal()
# MDS
mds <- plotMDS(vst_matriz, plot = FALSE)
metadatos$MDS1 <- mds$x
metadatos$MDS2 <- mds$y
# grafico coloreado x cohorte
ggplot(metadatos, aes(x = MDS1, y = MDS2, color = cohort)) +
  geom_point(size = 3) +
  theme_minimal() +
  labs(title = "MDS - coloreado por cohorte")
```

```r
# Identifica y elimina muestras atípicas (outliers).
# Ejemplo: eliminar muestras fuera de +/- 3 SDs en PC1 o PC2
outlier_idx <- which(abs(metadatos$PC1) > 3 * sd(metadatos$PC1) |
                       abs(metadatos$PC2) > 3 * sd(metadatos$PC2))

# Nombres de muestras atípicas
outliers <- rownames(metadatos)[outlier_idx]

# Eliminar outliers
if (length(outliers) > 0) {
  se <- se[, !colnames(se) %in% outliers]
  metadatos <- as.data.frame(colData(se))
  vst_matriz <- assays(se)$vst
}

# De acuerdo con los resultados de estos análisis, identifica qué variables
# en los metadatos pueden considerarse variables confusoras y por tanto deberían incluirse en

# grafico coloreado x cohorte + genero
p1<-p_cohort +
  geom_point(aes(shape = gender), color = "black", size = 2.5) +
```

```
  scale_shape_manual(values = c("Female" = 1, "Male" = 4)) +
  labs(title = "PCA - coloreado por cohorte y genero")
  theme_minimal()
    # grafico coloreado x cohorte + raza
p2<-p_cohort +
  geom_point(aes(shape = race), color = "black", size = 2.5) +
  #scale_shape_manual(values = c("Female" = 1, "Male" = 4)) +
  labs(title = "PCA - coloreado por cohorte y raza")
  theme_minimal()
  # grafico coloreado x cohorte + hospitalizado
 p3<- p_cohort +
    geom_point(aes(shape = hospitalized), color = "black", size = 2.5) +
    scale_shape_manual(values = c("Yes" = 1, "No" = 4)) +
    labs(title = "PCA - coloreado por cohorte y raza")
  theme_minimal()

# Añadiria genero y raza


## APARTADO 5
# Construye la matriz de diseño y las matrices de contrastes adecuadas para evaluar la expresi
# en las comparaciones Bacterial vs healthy y COVID19 vs healthy y realiza un análisis de expr
# semilla y determinar metodo
set.seed(myseed)
metodos <- sample(c("edgeR", "voom+limma", "DESeq2"), size = 1)
metodos # = voom+limma
library(edgeR)
dge <- DGEList(counts = assay(se, "counts"))
# Crear matriz de diseño
matriz_dis <- model.matrix(~ 0 + cohort + race + gender, data = metadatos)  # Añade aquí tus v
# Aplicar voom
v <- voom(dge, matriz_dis, plot = TRUE)
```

```
metadatos[] <- lapply(metadatos, function(x) {
  if (is.character(x) || is.factor(x)) {
    x <- as.character(x)           # convertir factor a carácter
    gsub("[- /]", "", x)            # eliminar guiones y espacios
  } else {
    x
  }
})
colnames(matriz_dis) <- gsub("[- /]", "", colnames(matriz_dis))
# hacemos los contrastes para comparar:
matriz_contrs <- makeContrasts(
  Bacterial_vs_healthy = cohortBacterial - cohorthealthy,
  COVID19_vs_healthy = cohortCOVID19 - cohorthealthy,
  levels = matriz_dis
```

```
)

# por ultimo ajustamos con limma
ajuste <- lmFit(v, matriz_dis)
ajuste2 <- contrasts.fit(ajuste, matriz_contrs)
ajuste2 <- eBayes(ajuste2)

# resultados (logFC > 1.5 como umbral)
res_bacterial <- topTable(ajuste2, coef = "Bacterial_vs_healthy", number = Inf, adjust = "fdr"
res_covid <- topTable(ajuste2, coef = "COVID19_vs_healthy", number = Inf, adjust = "fdr", lfc =

# filtramos tb solo significativos
sig_bacterial <- subset(res_bacterial, adj.P.Val < 0.05)
sig_covid <- subset(res_covid, adj.P.Val < 0.05)



## APARTADO 6
# Compara los resultados de ambos contrastes (Bacterial vs healthy y COVID19 vs healthy).
# vamos a usar un diagrama de VEnn
library(VennDiagram)

# Obtenemos los genes diferencisdos d cada uno
genes_bact <- rownames(sig_bacterial)
genes_covid <- rownames(sig_covid)

# Diagrama de Venn
grid.newpage()
diag_venn <- venn.diagram(
  list(Bacterial = genes_bact, COVID19 = genes_covid),
  filename = NULL,
  fill = c("red", "blue"),
  alpha = 0.5,
  cex = 1.5,
  cat.cex = 1.5,
  main = "Genes diferencialmente expresados"
)

grid::grid.draw(diag_venn)
```

```
## APARTADO 7
# Realiza un análisis de sobrerepresentación para identificar las funciones enriquecidas entre
# genes sobreexpresados en pacientes con COVID19 en comparación con los controles sanos.
# Utiliza solo el dominio de Gene Ontology "Biological Process".
library(clusterProfiler) # realiza el analisis go
library(org.Hs.eg.db) # para mapear los id de genes a terminos biologicos
```

```r
# Convertir Ensembl a Entrez ID
genes_up_covid <- rownames(subset(sig_covid, logFC > 0))  # cgemos solo los genes sobreexpresa
entrez_ids <- mapIds(org.Hs.eg.db,
                     keys = genes_up_covid,
                     keytype = "ENSEMBL",
                     column = "ENTREZID",
                     multiVals = "first")
entrez_ids <- na.omit(entrez_ids)

# GO Biological Process:  test de hipergeométrica para evaluar si los genes de nuestra lista
# están sobre-representados en ciertas categorías GO
ego_bp <- enrichGO(
  gene = entrez_ids,
  OrgDb = org.Hs.eg.db,
  ont = "BP",
  keyType = "ENTREZID",
  pAdjustMethod = "fdr",
  qvalueCutoff = 0.05,
  readable = TRUE
)

# visualizacion (sin revigo)
barplot(ego_bp, showCategory = 20, title = "GO: Biological Process (COVID19)")
```

```r
dotplot(ego_bp, showCategory = 20)
```