

PEC 2 - Análisis de datos ómicos

Alexandre Arredondo Campaña

25/5/2020

Contents

Abstract	1
Objetivos	1
Materiales y métodos	2
Naturaleza de los datos	2
Métodos y herramientas utilizados en el análisis	2
Resultados	2
Discusión	9
Apéndice	10
Apéndice 1	10
Apéndice 2	10
Apéndice 3	11
Apéndice 4	11
Apéndice 5	12

Enlace al repositorio de github: https://github.com/alexarredondodentaid/PEC2_ADO.git

Abstract

Con el objetivo de implementar un *pipeline* de análisis de datos Ref-Seq y asimilar los conocimientos necesarios para ello, en esta PEC se analizaron 30 muestras de tejido tiroidal, repartidas en 3 grupos de 10 muestras cada una: *Not infiltrated tissues* (NIT), *Small focal infiltrates* (STI) y *Extensive lymphoid infiltrates* (ELI). Se buscaron diferencias significativas en la expresión genética de los distintos grupos utilizando el paquete DESeq2 de Bioconductor, encontrando genes sobre y subexpresados en todas las comparaciones.

Objetivos

El objetivo de esta PEC es el de comparar los perfiles de expresión genética de 3 tipos de muestra de tejido tiroidal a partir de unos archivos targets.csv y counts.csv ya suministrados en el enunciado.

Materiales y métodos

Naturaleza de los datos

Los datos provienen del repositorio GTEx (Genotype-Tissue Expression project), el cual pone a disposición pública datos de expresión genética de 54 tipos de tejidos sanos obtenidos de cerca de 1000 sujetos. En esta PEC se emplearon datos de tejido tiroidal, partiendo de un dataset en el que había 292 muestras de 3 grupos distintos:

- *Not infiltrated tissues* (NIT): 236 muestras.
- *Small focal infiltrates* (SFI): 42 muestras.
- *Extensive lymphoid infiltrates* (ELI): 14 muestras.

Sin embargo, para simplificar los cálculos y para igualar el número de muestras de los distintos grupos, en este estudio se pidió utilizar 10 muestras aleatorias de cada grupo, lo cual se realizó mediante un script que se desarrolla en el siguiente apartado.

Métodos y herramientas utilizados en el análisis

Para realizar el análisis de expresión se ha utilizado el software RStudio junto con el paquete DESeq2, el cual proporciona funciones integradas para este tipo de estudios.

Para obtener 10 muestras aleatorias de cada tipo, se utilizó el código del apéndice 1, en el que en primer lugar se separaron del archivo targets.csv las muestras según el grupo (ELI, SFI o NIT) y después se aleatorizaron 10 muestras de cada uno utilizando la función `sample`. Después se unieron bajo un mismo data frame para tener las 30 muestras deseadas. Para seleccionar los contajes correspondientes a las muestras, se utilizó la variable `Sample_Name` como filtro, quedándonos con las 30 columnas de counts.csv que nos interesaban.

Una vez seleccionados los datos, se utilizó el paquete DESeq2 para filtrar aquellos contajes que mostraran un nivel global de expresión muy bajo (< 1 en el total de las muestras). Posteriormente, mediante la función `vst` se normalizaron los datos según el concepto de la transformación estabilizadora de la varianza. El código puede consultarse en el apéndice 2.

Para agrupar las muestras según su expresión diferencial genética se utilizó el código disponible en el apéndice 3. En este análisis se hizo un agrupamiento según la distancia de los datos que se muestra en un mapa de calor y en un agrupamiento PCA (Principal Component Analysis).

Para determinar cuantitativamente la diferencia de expresión de genes concretos en las distintas comparaciones, se utilizó la función `results` del paquete DESeq2, introduciendo los contrastes NIT vs SFI, ELI vs SFI y ELI vs NIT (apéndice 4). La cantidad de genes sobre y subexpresados fueron resumidos, primero filtrando por aquellos con un p valor < 0.05 y después ordenando de manera que se mostraran los 20 genes más sobreexpresados y los 20 genes más subexpresados de cada comparación. Se realizó un mapa de calor de estas comparaciones mediante la función `pheatmap`, mostrando las agrupaciones más significativas encontradas.

Finalmente, se realizó un análisis de significación biológica de los genes elegidos, añadiendo el código GO y SYMBOL a las tablas de expresión diferencial. Dichas tablas fueron exportadas a formato `.csv` (apéndice 5) y, manualmente, se añadió la función biológica de cada código GO.

Resultados

En el proceso de filtraje se observó una reducción en el número de genes de 12838, los cuales correspondían a genes que mostraron muy poca o ninguna expresión diferencial.

La normalización de los datos mediante la función `vst` permitió visualizar los resultados obtenidos mediante un gráfico (figura 1), el cual confirmaba que los datos estaban bien normalizados y que el procedimiento podía continuar.

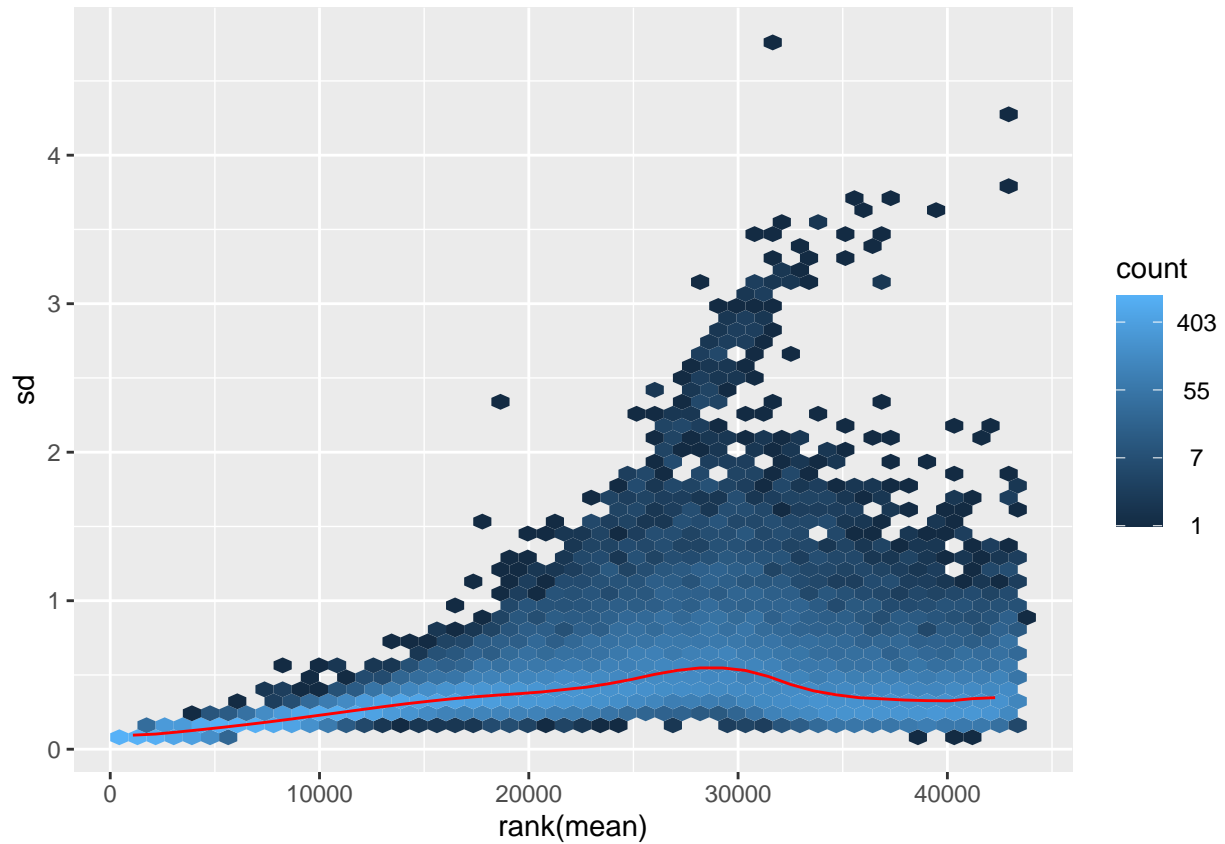


Figura 1. Gráfico que muestra la desviación estándar en función de la media. Como se puede ver, y a pesar de que hay algunos outliers, la desviación estándar es suficientemente baja como para aceptar la transformación realizada

Se hizo un análisis para determinar como se agrupaban los datos. Los resultados se pueden observar en las figuras 2 y 3.

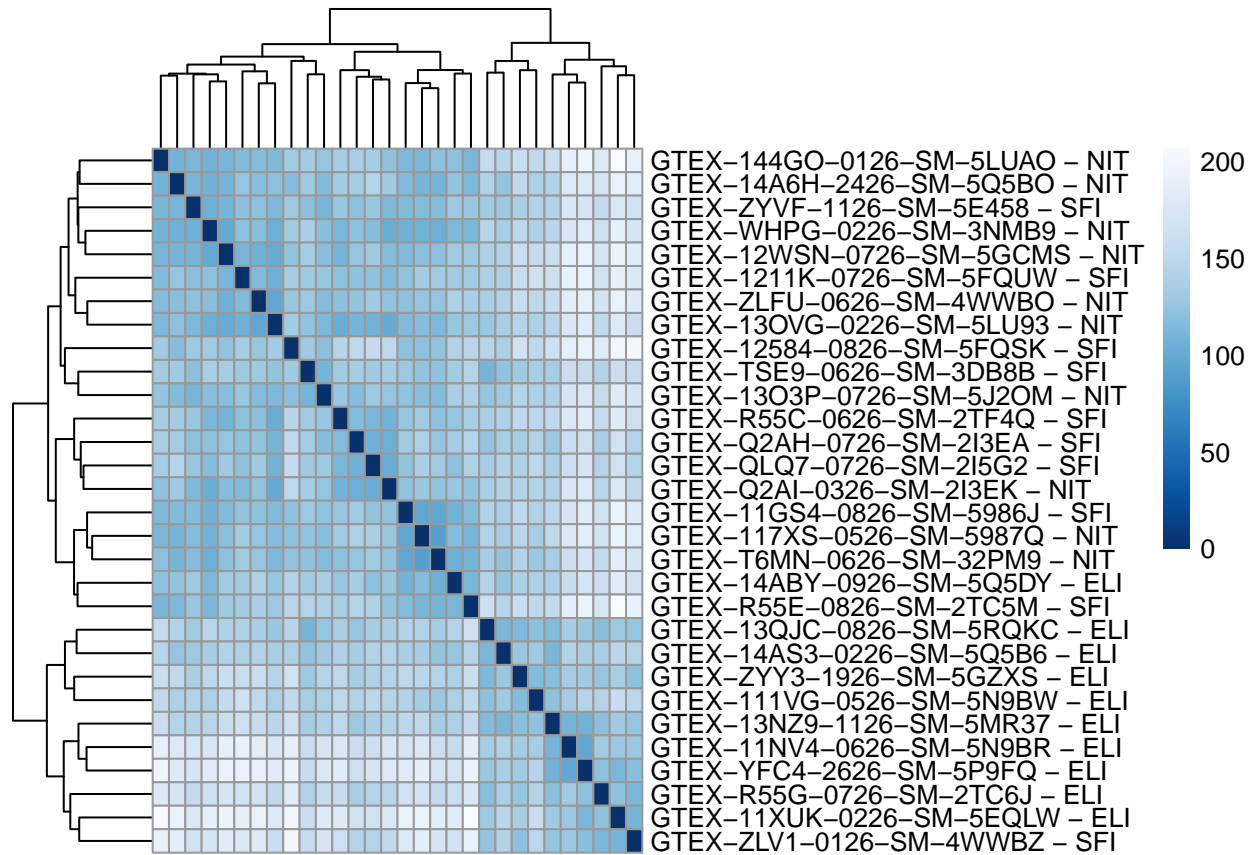


Figura 2. Mapa de calor en el que se muestran las similitudes entre muestras a partir de los datos obtenidos de la función `vst`. Se puede observar dos agrupamientos: las muestras ELI por un lado y las NIT y SFI por el otro.

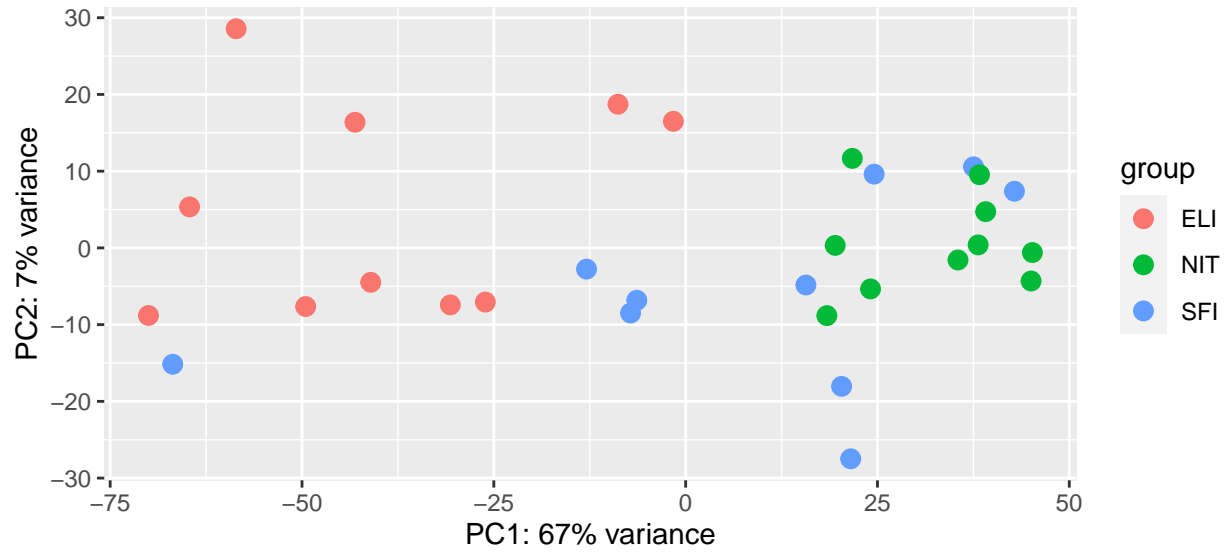


Figura 3. Gráfico PCA a partir de los datos obtenidos con la función `vst`. Se confirman los resultados mostrados en la figura 2, en la que se agrupan las muestras ELI por un lado y las NIT y SIF por el otro.

Para observar si se daba agrupamiento entre la expresión diferencial de ciertos genes y ciertas muestras, se realizó un mapa de calor (figura 4), en el que se observaron agrupamientos claros entre la sobre y subexpresión de algunos genes en ciertas muestras.

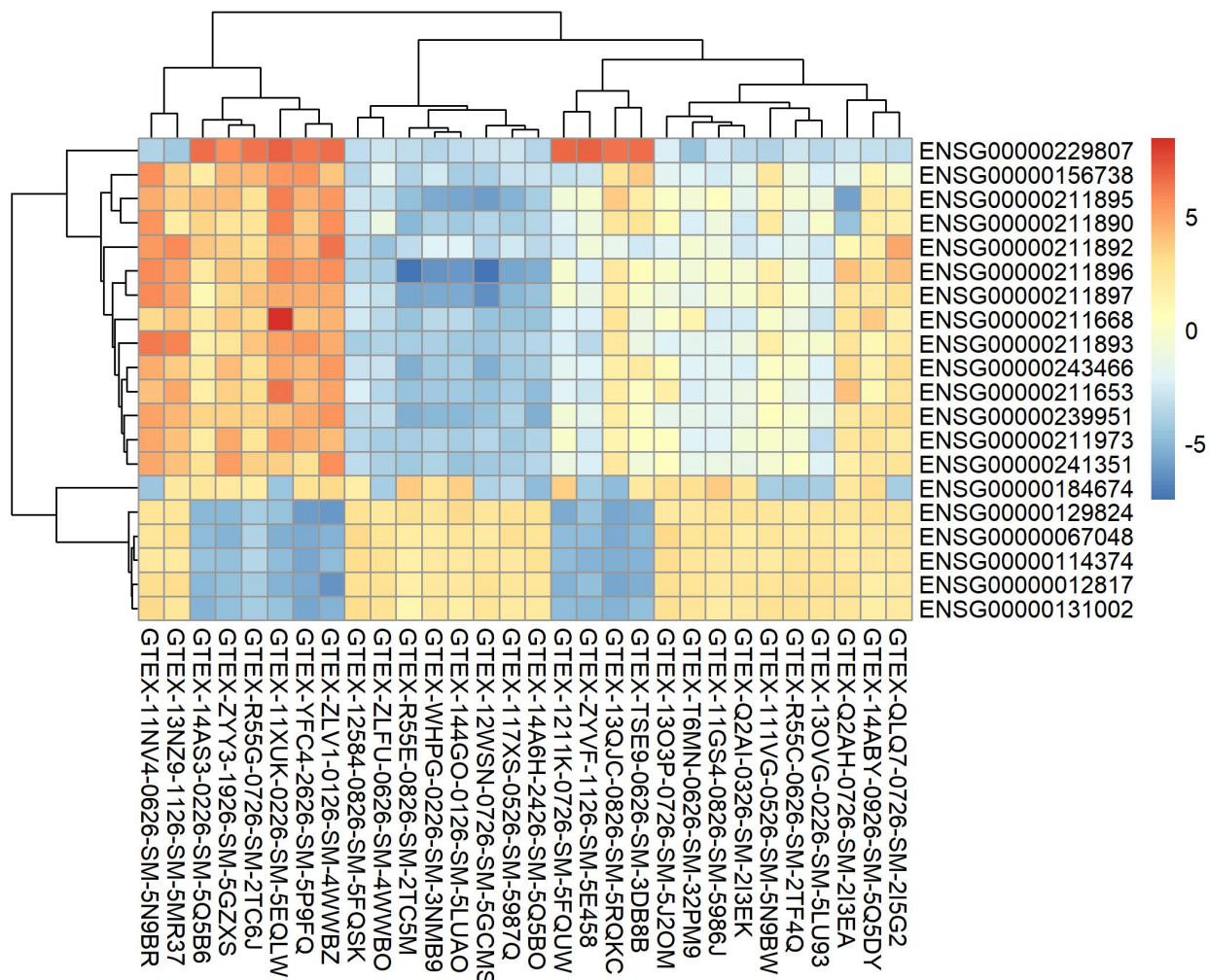


Figura 4. Mapa de calor en el que se agrupan los genes que han mostrado una mayor expresión diferencial junto con las muestras en las que se ha observado dicha expresión. En el eje X se disponen las muestras, mientras que en el eje Y los genes diferencialmente expresados.

El listado de los 20 genes más sobreexpresados y subexpresados de cada comparación, con el cambio expresado en logaritmo en base 2 (columna Dif.), se muestran en las siguientes tablas:

Genes sobreexpresados en la comparación ELI vs NIT

```
## # A tibble: 20 x 3
##   ...1      Dif. Function
##   <chr>      <dbl> <chr>
## 1 ENSG00000229807 10.2 dosage compensation by inactivation of X chromosome
## 2 ENSG00000170054  9.15 serine-type endopeptidase inhibitor activity
## 3 ENSG00000253274  8.76 NA
## 4 ENSG00000260303  8.39 NA
## 5 ENSG00000211599  8.37 NA
## 6 ENSG00000255760  8.27 NA
## 7 ENSG00000222037  8.17 NA
## 8 ENSG00000257275  8.16 NA
## 9 ENSG00000100721  8.12 protein binding
## 10 ENSG00000253497  8.06 NA
```

```
## 11 ENSG00000242534 7.92 NA
## 12 ENSG00000254029 7.90 NA
## 13 ENSG00000253818 7.86 NA
## 14 ENSG00000243063 7.86 NA
## 15 ENSG00000235896 7.85 NA
## 16 ENSG00000181617 7.76 extracellular region
## 17 ENSG00000242580 7.74 NA
## 18 ENSG00000211648 7.73 NA
## 19 ENSG00000211667 7.70 NA
## 20 ENSG00000253234 7.62 NA
```

Genes subexpresados en la comparación ELI vs NTI

```
## # A tibble: 20 x 3
##   ...1      Dif. Function
##   <chr>      <dbl> <chr>
## 1 ENSG00000230663 -4.58 NA
## 2 ENSG00000110680 -4.39 monocyte chemotaxis
## 3 ENSG00000079689 -4.29 calcium ion binding
## 4 ENSG00000249694 -4.08 NA
## 5 ENSG00000250387 -3.48 NA
## 6 ENSG00000156269 -3.44 peptide alpha-N-acetyltransferase activity
## 7 ENSG00000259129 -3.17 NA
## 8 ENSG00000234025 -3.11 NA
## 9 ENSG00000237859 -3.08 NA
## 10 ENSG00000137558 -3.03 extracellular space
## 11 ENSG00000228411 -3.02 NA
## 12 ENSG00000242097 -3.01 NA
## 13 ENSG00000184735 -2.98 RNA binding
## 14 ENSG00000242307 -2.98 NA
## 15 ENSG00000251676 -2.89 NA
## 16 ENSG00000120332 -2.89 osteoblast development
## 17 ENSG00000089225 -2.88 negative regulation of transcription by RNA polymerase-
## 18 ENSG00000258011 -2.87 NA
## 19 ENSG00000255986 -2.86 NA
## 20 ENSG00000230902 -2.77 NA
```

Genes sobreexpresados en la comparación ELI vs SFI

```
## # A tibble: 20 x 3
##   ...1      Dif. Function
##   <chr>      <dbl> <chr>
## 1 ENSG00000254~ 7.36 NA
## 2 ENSG00000231~ 5.60 NA
## 3 ENSG00000229~ 5.42 NA
## 4 ENSG00000165~ 5.31 RNA polymerase II cis-regulatory region sequence-specifi-
## 5 ENSG00000162~ 5.16 adaptive immune response
## 6 ENSG00000228~ 4.84 NA
## 7 ENSG00000170~ 4.83 serine-type endopeptidase inhibitor activity
## 8 ENSG00000181~ 4.52 extracellular region
## 9 ENSG00000223~ 4.45 NA
## 10 ENSG00000154~ 4.43 virus receptor activity
## 11 ENSG00000261~ 4.41 NA
```

```
## 12 ENSG00000164~ 4.31 cytokine receptor activity
## 13 ENSG00000180~ 4.29 DNA-binding transcription factor activity, RNA polymeras~
## 14 ENSG00000211~ 4.24 NA
## 15 ENSG00000255~ 4.24 G protein-coupled receptor activity
## 16 ENSG00000104~ 4.18 positive regulation of humoral immune response mediated ~
## 17 ENSG00000253~ 4.15 NA
## 18 ENSG00000224~ 4.14 NA
## 19 ENSG00000166~ 4.10 diet induced thermogenesis
## 20 ENSG00000264~ 4.09 NA
```

Genes subexpresados en la comparación ELI vs SFI

```
## # A tibble: 20 x 3
##   ...1      Dif. Function
##   <chr>      <dbl> <chr>
## 1 ENSG00000152266 -21.9 negative regulation of transcription by RNA polymeras~
## 2 ENSG00000110680 -5.86 monocyte chemotaxis
## 3 ENSG00000134443 -5.38 signaling receptor binding
## 4 ENSG00000243137 -4.96 extracellular region
## 5 ENSG00000178082 -4.03 NA
## 6 ENSG00000079689 -3.95 calcium ion binding
## 7 ENSG00000114638 -3.52 structural molecule activit
## 8 ENSG00000234025 -3.40 NA
## 9 ENSG00000250033 -3.40 NA
## 10 ENSG00000100604 -3.38 regulation of the force of heart contraction
## 11 ENSG00000230663 -3.18 NA
## 12 ENSG00000180673 -3.16 NA
## 13 ENSG00000130720 -3.14 protein binding
## 14 ENSG00000120332 -3.10 osteoblast development
## 15 ENSG00000243584 -3.09 NA
## 16 ENSG00000128564 -3.08 ovarian follicle development
## 17 ENSG00000089225 -3.04 negative regulation of transcription by RNA polymeras~
## 18 ENSG00000232762 -2.77 NA
## 19 ENSG00000136695 -2.67 negative regulation of cytokine-mediated signaling pa~
## 20 ENSG00000140465 -2.66 response to hypoxia
```

Genes sobreexpresados en la comparación NIT vs SFI

```
## # A tibble: 20 x 3
##   ...1      Dif. Function
##   <chr>      <dbl> <chr>
## 1 ENSG000002165~ 3.47 NA
## 2 ENSG000002540~ 2.95 NA
## 3 ENSG000002601~ 2.59 NA
## 4 ENSG0000000073~ 2.27 transketolase activity
## 5 ENSG000001064~ 1.88 positive regulation of receptor internalization
## 6 ENSG000001658~ 1.57 extracellular region
## 7 ENSG000002260~ 1.22 NA
## 8 ENSG000001868~ -1.49 positive regulation of leukocyte migration
## 9 ENSG000001727~ -1.53 GTPase activity
## 10 ENSG000002717~ -1.60 NA
## 11 ENSG000000622~ -1.77 2-acylglycerol 0-acyltransferase activity
## 12 ENSG000001168~ -1.79 membrane raft polarization
```



```
## 13 ENSG000000282~ -1.88 RNA polymerase II cis-regulatory region sequence-specif~
## 14 ENSG000001597~ -1.92 ruffle
## 15 ENSG000001596~ -2.02 G protein-coupled receptor activity
## 16 ENSG000001242~ -2.04 DNA binding
## 17 ENSG000001020~ -2.06 G1/S transition of mitotic cell cycle
## 18 ENSG000001379~ -2.07 plasma membrane
## 19 ENSG000001635~ -2.07 activation of innate immune response
## 20 ENSG000001670~ -2.08 cell
```

Genes subexpresados en la comparación NIT vs SFI

```
## # A tibble: 20 x 3
##   ...1      Dif. Function
##   <chr>    <dbl> <chr>
## 1 ENSG000002165~ 3.47 NA
## 2 ENSG000002540~ 2.95 NA
## 3 ENSG000002601~ 2.59 NA
## 4 ENSG000000073~ 2.27 transketolase activity
## 5 ENSG000001064~ 1.88 positive regulation of receptor internalization
## 6 ENSG000001658~ 1.57 extracellular region
## 7 ENSG000002260~ 1.22 NA
## 8 ENSG000001868~ -1.49 positive regulation of leukocyte migration
## 9 ENSG000001727~ -1.53 GTPase activity
## 10 ENSG000002717~ -1.60 NA
## 11 ENSG000000622~ -1.77 2-acylglycerol O-acyltransferase activity
## 12 ENSG000001168~ -1.79 membrane raft polarization
## 13 ENSG000000282~ -1.88 RNA polymerase II cis-regulatory region sequence-specif~
## 14 ENSG000001597~ -1.92 ruffle
## 15 ENSG000001596~ -2.02 G protein-coupled receptor activity
## 16 ENSG000001242~ -2.04 DNA binding
## 17 ENSG000001020~ -2.06 G1/S transition of mitotic cell cycle
## 18 ENSG000001379~ -2.07 plasma membrane
## 19 ENSG000001635~ -2.07 activation of innate immune response
## 20 ENSG000001670~ -2.08 cell
```

Para una mejor visualización de los datos ampliados consultar los archivos `.xlsx` disponibles en la carpeta `results` del repositorio de github.

Discusión

La principal limitación de este estudio se ha dado al analizar la significación biológica. Si bien es cierto que se han proporcionado los datos correspondientes a los 20 genes más sobreexpresados y subexpresados, lo cual ya cubre una parte importante de la información necesaria, también es cierto que el investigador que quiera analizar la totalidad de los datos agradecería tener los datos de significación biológica de todas las muestras. Esto no fue posible debido a cierta compatibilidad de librerías encontrada en RStudio, por lo que se optó por anotar los genes manualmente, lo cual *imposibilita* anotar más de 40.000 genes.

Hay que destacar que una parte nada desdeñable de los genes con mayor expresión diferencial no pudieron ser anotados debido a que no tenían un código GO asociado. Se podría haber optado por eliminar estos genes y proporcionar información únicamente acerca de aquellos genes con código GO. Sin embargo, se optó por añadirlos al informe debido a su *posible* significación biológica. Por ejemplo, al comparar la expresión genética entre muestras ELI y NIT, se observó que el gen ENSG00000253274 mostraba una expresión diferencial de

8.76 log2 y sin embargo no se encontró un código GO asociado. Es de esperar que el investigador que reciba un informe de estas características quiera conocer los genes con expresión diferencial tengan o no un código GO asociado, y por lo tanto se decidió incluirlo en el informe.

Apéndice

Apéndice 1

Código empleado para la obtención de las 30 muestras aleatorias.

```
# Leemos los archivos facilitados para el estudio
targets <- read.csv("./data/targets.csv")
counts <- read.csv("./data/counts.csv", sep = ";")

# Extraemos los 3 grupos (NIT, SFI y ELI) y seleccionamos 10 muestras aleatorias de cada
set.seed(123)
subsample_eli <- subset(targets, Group == "ELI")
subsample_sfi <- subset(targets, Group == "SFI")
subsample_nit <- subset(targets, Group == "NIT")
random_eli <- subsample_eli[sample(nrow(subsample_eli), 10),]
random_sfi <- subsample_sfi[sample(nrow(subsample_sfi), 10),]
random_nit <- subsample_nit[sample(nrow(subsample_nit), 10),]

random_targets <- rbind(random_eli, random_sfi, random_nit)
row.names(random_targets) <- random_targets$Sample_Name

# Puesto que existe una diferencia entre el nombre de la muestra en el dataset counts vs
# el dataset targets, realizamos una modificación para equipararlos:
colnames(counts) <- gsub("\\\\.", "-", colnames(counts))

# Ahora seleccionamos los datos del dataset counts basándonos en las muestras elegidas
random_names <- c(as.character(random_targets$Sample_Name))

# Modificamos el nombre de la referencia Ensembl para evitar problemas más adelante
# del pipeline:
random_df <- counts[random_names]
Ensembl <- counts$X
Ensembl <- gsub("\\\\.*", "", Ensembl, fixed = FALSE)
row.names(random_df) <- Ensembl
```

Apéndice 2

Código empleado para el filtraje de genes con poca expresión y la normalización de los datos.

```
dds <- DESeqDataSetFromMatrix(countData = random_df, colData = random_targets,
                             design = ~ Group)
nrow(dds)

# Filtramos (eliminamos aquellas filas que solo tienen 0s o 1s)
dds <- dds[rowSums(counts(dds)) > 1,]
nrow(dds)
```

```

# Restando las filas obtenidas anteriormente y las actuales, calculamos cuantos genes
# no han mostrado expresión diferencial.

# Normalizamos según el concepto de la transformación estabilizadora de la varianza:
vsd <- vst(dds, blind = FALSE)
head(assay(vsd))

# Comprobamos los efectos en un gráfico:
meanSdPlot(assay(vsd))

```

Apéndice 3

Código para las figuras 2 y 3

```

# Calculamos como de distintas son las muestras entre si:
sampleDists <- dist(t(assay(vsd)))
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$Sample_Name, vsd$Group, sep = " - ")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(9,"Blues")))(255)

pheatmap(sampleDistMatrix, clustering_distance_rows = sampleDists,
          clustering_distance_cols = sampleDists, col = colors)

# Hacemos un gráfico PCA para representar los agrupamientos de otro modo:
plotPCA(vsd, intgroup = c("Group"))

```

Apéndice 4

Código para la selección de aquellos genes más sobre y subexpresados de cada comparación y para la realización de la figura 4.

```

# Empezamos el análisis de expresión diferencial
dds <- DESeq(dds, parallel = TRUE)

# Realizamos primero el contraste entre las comparaciones de los grupos:
res_nit_sfi <- results(dds, contrast = c("Group", "NIT", "SFI"))
res_eli_sfi <- results(dds, contrast = c("Group", "ELI", "SFI"))
res_eli_nit <- results(dds, contrast = c("Group", "ELI", "NIT"))

# Ahora comprobamos los (como máximo) 20 genes más sobreexpresados y los 20 genes
# más subexpresados en cada comparación usando un padj < 0.05
res_nit_sfi_adj <- subset(res_nit_sfi, padj < 0.05)
res_eli_sfi_adj <- subset(res_eli_sfi, padj < 0.05)
res_eli_nit_adj <- subset(res_eli_nit, padj < 0.05)

# Subexpresados:
head(res_nit_sfi_adj[order(res_nit_sfi_adj$log2FoldChange),], 20)
head(res_eli_sfi_adj[order(res_eli_sfi_adj$log2FoldChange),], 20)
head(res_eli_nit_adj[order(res_eli_nit_adj$log2FoldChange),], 20)

```

```

# Sobreexpresados:
head(res_nit_sfi_adj[order(res_nit_sfi_adj$log2FoldChange, decreasing = TRUE),], 20)
head(res_eli_sfi_adj[order(res_eli_sfi_adj$log2FoldChange, decreasing = TRUE),], 20)
head(res_eli_nit_adj[order(res_eli_nit_adj$log2FoldChange, decreasing = TRUE),], 20)

# Hacemos una representación gráfica de los clusters de genes más diferenciados
topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 20)

mat <- assay(vsd)[topVarGenes, ]
mat <- mat - rowMeans(mat)

anno <- as.data.frame(colData(vsd)[,c("Group")])
pheatmap(mat, anotacion_col = anno)

```

Apéndice 5

Código para la adición de la información biológica de los genes.

```

# Anotamos y exportamos los resultados (significación biológica)

# Añadimos el nombre del gen "SYMBOL" y "GO"
res_nit_sfi_adj$symbol <- mapIds(org.Hs.eg.db, keys=row.names(res_nit_sfi_adj),
                                column = "SYMBOL", keytype = "ENSEMBL", multiVals = "first")
res_nit_sfi_adj$GO <- mapIds(org.Hs.eg.db, keys=row.names(res_nit_sfi_adj),
                             column = "GO", keytype = "ENSEMBL", multiVals = "first")

res_eli_sfi_adj$symbol <- mapIds(org.Hs.eg.db, keys=row.names(res_eli_sfi_adj),
                                column = "SYMBOL", keytype = "ENSEMBL", multiVals = "first")
res_eli_sfi_adj$GO <- mapIds(org.Hs.eg.db, keys=row.names(res_eli_sfi_adj),
                             column = "GO", keytype = "ENSEMBL", multiVals = "first")

res_eli_nit_adj$symbol <- mapIds(org.Hs.eg.db, keys=row.names(res_eli_nit_adj),
                                column = "SYMBOL", keytype = "ENSEMBL", multiVals = "first")
res_eli_nit_adj$GO <- mapIds(org.Hs.eg.db, keys=row.names(res_eli_nit_adj),
                             column = "GO", keytype = "ENSEMBL", multiVals = "first")

# Exportamos las tablas de los sobreexpresados
res_nit_sfi_adjDF <- as.data.frame(res_nit_sfi_adj)
res_nit_sfi_adjDFSobre <- res_nit_sfi_adjDF[order(res_nit_sfi_adj$log2FoldChange,
                                                  decreasing = TRUE),]
write.csv(res_nit_sfi_adjDFSobre[1:20,], file="./results/resultats_NITvsSFIsofre.csv")

res_eli_sfi_adjDF <- as.data.frame(res_eli_sfi_adj)
res_eli_sfi_adjDFSobre <- res_eli_sfi_adjDF[order(res_eli_sfi_adj$log2FoldChange,
                                                  decreasing = TRUE),]
write.csv(res_eli_sfi_adjDFSobre[1:20,], file="./results/resultats_ELIVsSFIsofre.csv")

res_eli_nit_adjDF <- as.data.frame(res_eli_nit_adj)
res_eli_nit_adjDFSobre <- res_eli_nit_adjDF[order(res_eli_nit_adj$log2FoldChange,
                                                  decreasing = TRUE),]
write.csv(res_eli_nit_adjDFSobre[1:20,], file="./results/resultats_ELIVsNITsofre.csv")

```

```

# Exportamos las tablas de los subexpresados
res_nit_sfi_adjDFsub <- res_nit_sfi_adjDF[order(res_nit_sfi_adj$log2FoldChange),]
write.csv(res_nit_sfi_adjDFsub[1:20,], file="./results/resultats_NITvsSFIsobresub.csv")

res_eli_sfi_adjDFsub <- res_eli_sfi_adjDF[order(res_eli_sfi_adj$log2FoldChange),]
write.csv(res_eli_sfi_adjDFsub[1:20,], file="./results/resultats_ELIVsSFIsobresub.csv")

res_eli_nit_adjDFsub <- res_eli_nit_adjDF[order(res_eli_nit_adj$log2FoldChange),]
write.csv(res_eli_nit_adjDFsub[1:20,], file="./results/resultats_ELIVsNITsobresub.csv")

# Leemos las nuevas tablas modificadas manualmente:
eli_nit_sobre <- read_excel("./results/ELIVsNITsobre.xlsx")
eli_nit_sub <- read_excel("./results/ELIVsNITsub.xlsx")
eli_sfi_sobre <- read_excel("./results/ELIVsSFIsobre.xlsx")
eli_sfi_sub <- read_excel("./results/ELIVsSFIsobresub.xlsx")
nit_sfi_sobre <- read_excel("./results/NITvsSFIsobre.xlsx")
nit_sfi_sub <- read_excel("./results/NITvsSFIsobresub.xlsx")

```