MÁSTER UNIVERSITARIO EN BIOINFORMÁTICA Y BIOESTADÍSTICA

PRUEBA de EVALUACIÓN CONTINUA 2
Análisis de datos de ultrasecuenciación

Análisis de Datos Ómicos
Alexandre Sánchez Pla

Nombre y Apellidos: **Judith Guitart Matas**

INFORME CIENTÍFICO-TÉCNICO

El proyecto *Genotype-Tissue Expression* (GTEx) fue lanzado en 2010 por *National Insitutes of Health* (NIH) con el objetivo de estudiar cambios en la expresión de los genes que puedan ser los causantes a presentar cierta susceptibilidad a determinadas enfermedades humanas [1]. Supone un repositorio abierto a toda la comunidad científica con el objetivo de ampliar el conocimiento sobre la correlación entre el genotipo y la expresión génica en un total de 54 tejidos humanos y más de 25.000 muestras biológicas.

El código y los datos del análisis desarrollado en el presente informe, centrado en los datos de expresión pertenecientes al tejido tiroideo, puede descargarse del repositorio de Github en la siguiente url [2]:

https://github.com/judithguitart/guitart_judith_ADO_PEC2

TABLA DE CONTENIDOS

1. ABSTRACT

La tiroides es una glándula endocrina que regula el metabolismo del cuerpo y la sensibilidad a otras hormonas. La infiltración de células en la glándula tiroides provoca su deterioro progresivo y puede ser causada por una enfermedad autoinmune conocida como la tiroiditis de Hashimoto. El presente informe usa datos del repositorio GTEx para la realización de un análisis de expresión diferencial sobre tejidos tiroideos pertenecientes a pacientes con distintos tipos de infiltración.

2. OBJETIVOS

La finalidad de este análisis de ultrasecuenciación es examinar la presencia de genes diferencialmente expresados entre los datos pertenecientes al tejido tiroideo del proyecto GTEx. El conjunto de datos de secuenciación de ARN consta de un total de 292 muestras clasificadas en tres grupos según el grado de infiltración en el tejido. Para el análisis presentado en este informe se han seleccionado 10 muestras aleatorias de cada uno de los grupos, obteniendo un total de 30 muestras.

Así, el objetivo principal de este análisis es determinar los genes diferencialmente expresados entre los distintos grupos con distintos niveles de infiltración en la glándula tiroides para investigar si determinados genes o procesos biológicos tendrían algún efecto en el grado de infiltración en la tiroides de los distintos pacientes.

3. MATERIALES Y MÉTODOS

3.1 NATURALEZA DE LOS DATOS

Los datos usados para este análisis forman parte del proyecto GTEx y han sido proporcionados por el profesorado en un formato de dos archivos con la extensión *.csv*:

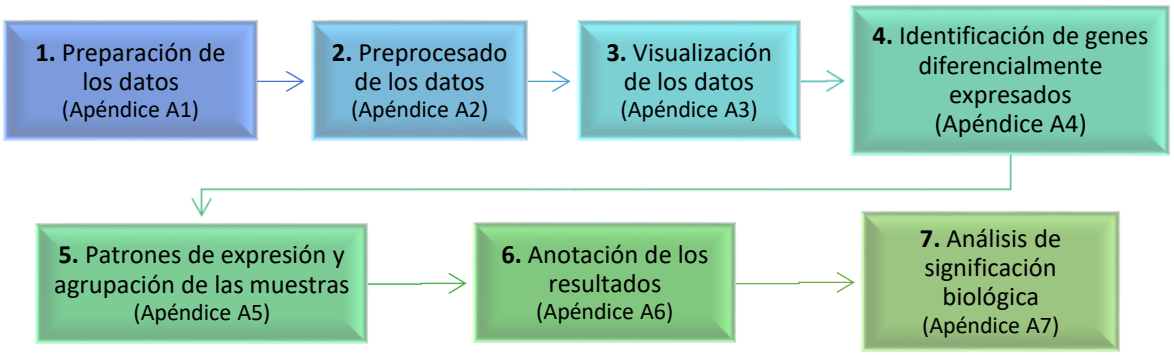
- El archivo *targets.csv* contiene información sobre las distintas muestras, como el grupo de análisis al que pertenecen o el sexo del paciente.
- El archivo *counts.csv* contiene los datos preprocesados de los niveles de expresión de los tránscritos analizados en cada una de las muestras, definidos por sus códigos *Ensembl*.

El tipo de estudio que permiten estos datos es la comparación directa entre grupos experimentales previamente identificados para la selección de genes diferencialmente expresados. Las muestras de este estudio están clasificadas en tres grupos según el tipo de infiltración medido en las 292 muestras de tejido tiroideo:

- *Not infiltrated tissues* (**NIT**): tejidos no infiltrados, en un total de 236 muestras.
- *Small focal infiltrates* (**SFI**): tejidos con pequeños infiltrados, en un total de 42 muestras.
- *Extensive lymphoid infiltrates* (**ELI**): tejidos extensamente infiltrados de linfoides, en un total de 14 muestras.

De cada uno de estos grupos se han extraído 10 muestras de manera aleatoria, siendo el tamaño muestral del análisis de un total de 30 muestras.

3.2 PROCEDIMIENTO GENERAL DE ANÁLISIS



3.3 SOFTWARE

Para la realización de este análisis se ha utilizado el lenguaje de programación estadístico *R* en el entorno de *Bioconductor* por sus herramientas para el análisis y comprensión de datos genómicos. Para el análisis de expresión diferencial se ha utilizado el paquete *DESeq2*, que permite analizar los conteos obtenidos en ensayos de secuenciación como el RNA-seq [3].

3.4 MÉTODOS

3.4.1 PREPARACIÓN DE LOS DATOS

Para preparar los datos para el análisis, primeramente, se leen los archivos *targets.csv* y *counts.csv* en R, los cuales se encuentran en la carpeta '*Datos*' dentro del directorio principal de este análisis.

A continuación, del archivo *targets.csv* se extraen 10 muestras aleatorias de cada uno de los grupos (`NIT`, `SFI` y `ELI`) mediante funciones del paquete `dplyr`. Para generar el subconjunto del archivo *counts.csv* con las mismas 30 muestras seleccionadas, inicialmente, se modifican los nombres de las columnas de este último para coincidir en formato con los identificadores de la variable `Sample_Name` de *targets.csv*. Ambos grupos de datos se guardan en dos variables nombradas `targets.pec` y `counts.pec`, respectivamente, y se comprueba que las filas de la variable `Sample_Name` de `targets.pec` coincidan con las columnas de `counts.pec`.

Dado que se dispone directamente de la matriz de conteos (`counts.pec`) y de la tabla con la información de las muestras (`targets.pec`), es posible construir directamente el objeto *DESeqDataSet* con la función `DESeqDataSetFromMatrix`. La fórmula de diseño tiene en cuenta la variable `Group` de `targets.pec` como diseño experimental, ya que el objetivo de este análisis es analizar la expresión diferencial entre los distintos grupos `NIT`, `SFI` y `ELI` (ver códigos en *Apéndice A1*).

3.4.2 PREPROCESADO DE LOS DATOS

La matriz de conteos del objeto *DESeqDataSet*, definido como `ddsM`, contiene filas con valor cero de expresión en todas las muestras, o bien con valores muy bajos. Estas filas pueden ser eliminadas con el objetivo de reducir el tamaño del objeto e incrementar la velocidad de las funciones a realizar sobre el mismo. Con un total de 30 muestras, se ha decidido eliminar todas aquellas filas que contienen 10 o menos conteos en el total de las muestras, que correspondería a unos 0,5 conteos por millón. Este proceso supone el **pre-filtrado** de los datos y genera un nuevo objeto `dds`.

A continuación, se realizan dos **transformaciones** independientes sobre los datos para eliminar la dependencia de la varianza sobre la media, pues muchos métodos estadísticos requieren homocedasticidad. Para este paso, se usan dos funciones del paquete `DESeq2` que realizan transformaciones en conteos de RNA-seq para estabilizar la varianza, sin establecer diferencias entre los distintos grupos: la función `vst` (*variance stabilizing transformation* [4]) y la función `rlog` (*regularized log transformation* [3]). Finalmente, para mostrar el efecto de las transformaciones, se representa la primera muestra contra la segunda transformadas mediante la función $\log_2$ y las dos funciones del paquete `DESeq2`. Finalmente, se estudia la distribución de los conteos mediante un diagrama de cajas de los datos no normalizados y normalizados representados en conteos por millón (cpm) (ver códigos en *Apéndice A2*).

3.4.3 VISUALIZACIÓN DE LOS DATOS

Con el objetivo de analizar la semejanza entre las distintas muestras, se calcula la distancia entre las mismas. Estas distancias se han representado en este informe mediante un mapa de calor con la función `pheatmap`, que a su vez permite agrupar las muestras en clústeres. Estas distancias también pueden visualizarse a partir de los datos transformados por *VST* mediante un análisis de componentes principales (PCA), que permite detectar si las muestras de grupos experimentales se agrupan de forma 'natural', o mediante gráficos de escala multidimensional (MDS), que usa matrices de distancias (ver códigos en *Apéndice A3*).

3.4.4 ANÁLISIS DE EXPRESIÓN DIFERENCIAL Y PATRONES DE EXPRESIÓN

El paquete `DESeq2` realiza análisis de expresión diferencial mediante la función `DESeq` [3]. De forma general, esta función ejecuta tres pasos independientes sobre los datos crudos: `estimateSizeFactors()`, que supone una normalización para controlar las diferencias entre las muestras, `estimateDispersions()` para cada uno de los genes, y `nbinomWaldTest()`, que establece el modelo lineal generalizado (GLM). La salida de esta función es un objeto *DESeqDataSet* a partir del cual es posible extraer resultados de cada una de las comparaciones con la función `results`. Como la variable `Group` tiene tres niveles distintos, es posible realizar tres comparaciones dos a dos, que en este informe se ven definidas con las siguientes variables: `res_NITvsSFI`, `res_NITvsELI`, `res_SFIvsELI`.

El análisis de significación biológica entre comparaciones ha sido ejecutado con un valor de significación (`alpha`) de 0.05 con el objetivo de disminuir la probabilidad de obtener falsos positivos. Finalmente, se ordenan los resultados de cada una de las comparaciones para mostrar los genes con un p-valor ajustado inferior a 0.05 que se encuentran más infra y sobreexpresados.

Para visualizar los resultados obtenidos en el apartado anterior se realiza un gráfico MA de cada una de las comparaciones, que permite observar la distribución de los coeficientes estimados. En estos gráficos se remarca el gen con un p-valor ajustado más pequeño para cada una de las comparaciones. Adicionalmente, se genera un histograma de los p-valores de cada comparación (ver código en *Apéndice A4*).

3.4.5 AGRUPACIÓN DE LAS MUESTRAS

Para observar la agrupación de las muestras se realiza un mapa de calor que representa una agrupación jerárquica de las muestras para los veinte genes con más varianza a partir de los datos transformados por *VST*. Y finalmente, se realiza una **comparación entre las distintas comparaciones** que se representa mediante un diagrama de Venn (ver código en *Apéndice A5*).

3.4.6 ANOTACIÓN DE LOS RESULTADOS

El proceso de anotación permite asociar los identificadores *Ensembl* de cada uno de los genes identificados en el análisis de expresión diferencial con identificadores con nombres más informativos para su interpretación: `SYMBOL`, `ENTREZID` y `GENENAME`. Para ello, es necesario eliminar la versión situada al final de cada uno de los identificadores *Ensembl* con la función `gsub` y cargar el paquete que contiene las anotaciones para *Homo sapiens* (`org.Hs.eg.db`), para posteriormente poder ejecutar la función `mapIDs` y así añadir estas columnas a la tabla de resultados de cada comparación, con el primer resultado disponible. Estos resultados han sido también representados mediante *volcano plots* (ver códigos en *Apéndice A6*).

3.4.7 ANÁLISIS DE SIGNIFICACIÓN BIOLÓGICA

El análisis de significación biológica utiliza bases de datos de anotación funcional para realizar análisis de enriquecimiento. En este análisis se ha usado el paquete de anotación de *Gene Ontology* (GO) con el objetivo de establecer si determinados procesos biológicos están presentes en mayor o menor medida en la lista de genes seleccionados diferencialmente expresados en cada una de las comparaciones. Para realizar este análisis se ha empleado la función `enrichGO` del paquete `clusterProfiler` que toma como entrada los identificadores de `ENTREZ` de la lista de genes seleccionada, utiliza las anotaciones *OrgDb* disponibles por Bioconductor, permite ajustar los *p*-valores para controlar la tasa de falsos positivos y finalmente, proporciona la lista de categorías del sistema de clasificación de GO más representadas en cada comparación (ver código en *Apéndice A7*) [5].

4. RESULTADOS

4.1 LOS DATOS

En el apartado *3.1* de los *Métodos* de este informe se ha explicado el proceso de obtención de 30 muestras aleatorias del archivo *targets.csv* y la posterior obtención de los conteos de estas muestras del archivo *counts.csv*. Una vez se ha comprobado que las muestras de ambas nuevas variables son coincidentes, definidas como `targets.pec` y `counts.pec`, tal y como puede comprobarse en el *Apéndice A1* de este informe, o bien, visualizando los archivos exportados en el repositorio Github [2], el objeto *DESeqDataSet* ha sido creado y definido como `ddsM`. A continuación, se muestran las características de este objeto:

```
ddsM <- DESeqDataSetFromMatrix(countData = counts.pec, colData = targe
ts.pec, design = ~ Group)

ddsM

## class: DESeqDataSet
## dim: 56202 30
## metadata(1): version
## assays(1): counts
## rownames(56202): ENSG00000223972.4 ENSG00000227232.4 ...
##   ENSG00000210195.2 ENSG00000210196.2
## rowData names(0):
## colnames(30): GTEX-11ZTT-1026-SM-5EQKF GTEX-11GSO-0626-SM-5A5LW ...
##   GTEX-YJ89-0726-SM-5P9F7 GTEX-13NZ9-1126-SM-5MR37
## colData names(8): SRA_Sample Sample_Name ... Group ShortName
```

4.2 PREPROCESADO DE LOS DATOS

El proceso de pre-filtrado, que implica la eliminación de los genes que contienen 10 o menos conteos en la suma total de las muestras, elimina 20308 filas, que no aportan información de expresión diferencial al análisis, y genera un nuevo objeto con un total de 35894 filas, definido como `dds`:

```
nrow(ddsM)

## [1] 56202

dds <- ddsM[rowSums(counts(ddsM)) >= 10, ]
nrow(dds)

## [1] 35894
```

En los siguientes gráficos de dispersión se muestra el efecto de las tres transformaciones realizadas sobre los conteos de nuestros datos para estabilizar la varianza de la primera muestra contra la segunda. A la izquierda se observa la transformación `log2` de los datos normalizados, la transformación `rlog` en el medio y la transformación `vst` a la derecha. Esta última transformación muestra una escala distinta respecto a las dos anteriores en el caso de los genes con conteos bajos. Asimismo, se observa que las funciones del paquete `DESeq2` (`rlog` y `vst`) son capaces de eliminar de forma más eficiente la elevada varianza que suele observarse en los conteos bajos, en comparación con la escala logarítmica ordinaria (`log2`) (Figura 1).

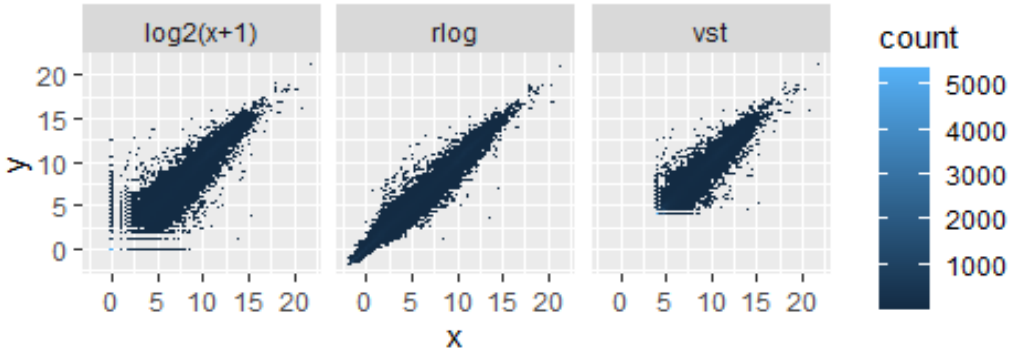


Figura 1. Efecto de las transformaciones `log2`, `rlog` y `vst`, de izquierda a derecha, sobre los datos.

La representación de los conteos por millón mediante diagramas de caja permite visualizar y comparar la distribución de los conteos en el total de las muestras. A continuación, se representa esta distribución de los datos crudos (Figura 2A) y de los datos normalizados (Figura 2B). Se observa que esta transformación de los datos genera valores de mediana y varianza más parecidos entre las muestras, siendo la media más cercana a cero. Este es el primer paso que se realiza en el análisis de expresión diferencial por la función `DESeq`:

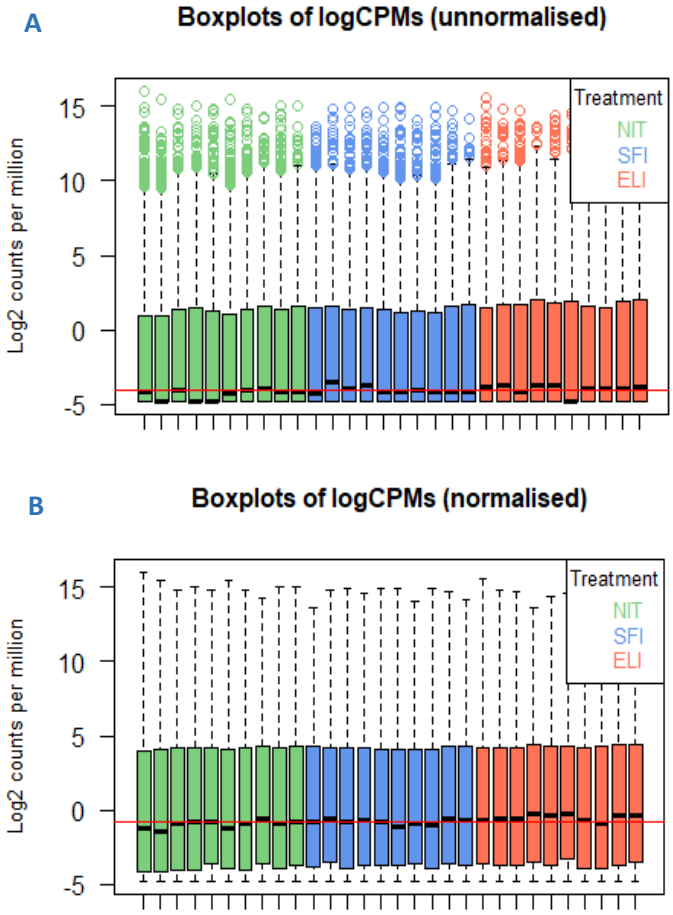


Figura 2. Efecto de la normalización en los conteos por millón de cada una de las muestras.
A. Datos no normalizados. B. Datos normalizados.

4.3 VISUALIZACIÓN DE LOS DATOS

El mapa de calor que se muestra a continuación agrupa las muestras según la distancia calculada entre las mismas. Cuanto más oscuro es el cuadro que representa la distancia entre dos muestras, menos distancia hay entre ambas muestras y, por lo tanto, éstas son más similares. Así, aunque no se observa un agrupamiento claro de los tipos distintos de infiltración, se observa un agrupamiento del grupo `NIT`, mayormente (arriba a la izquierda), seguidos de un contraste con muestras del grupo `ELI`, que se agrupan entre ellas y se representan en el medio del mapa de calor:

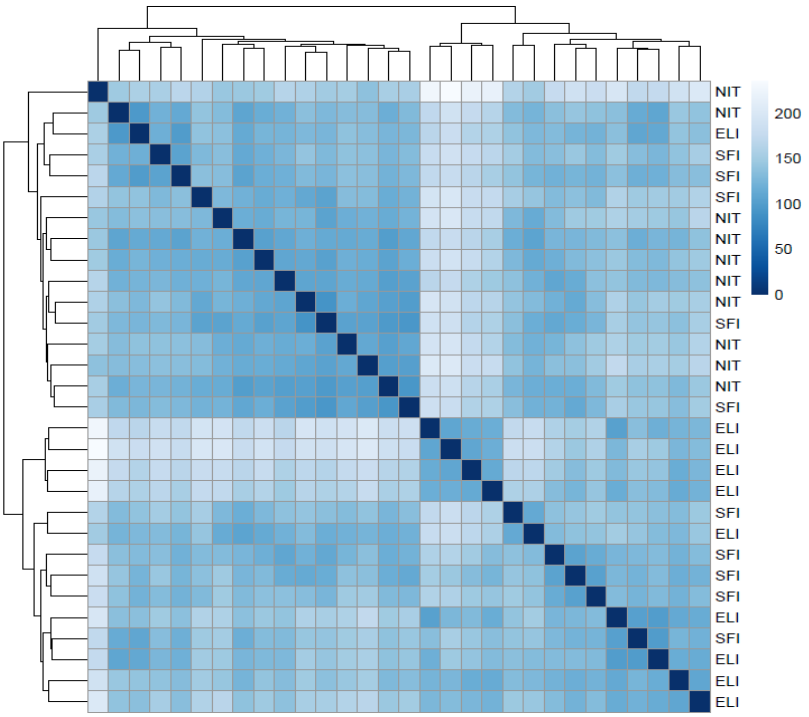


Figura 3. Mapa de calor representando las distancias entre las distintas muestras.

Seguidamente, se visualizan estas distancias calculadas a partir de los datos transformados por *VST* mediante un análisis de componentes principales (PCA, izquierda) y un gráfico de escala multidimensional (MDS, derecha):

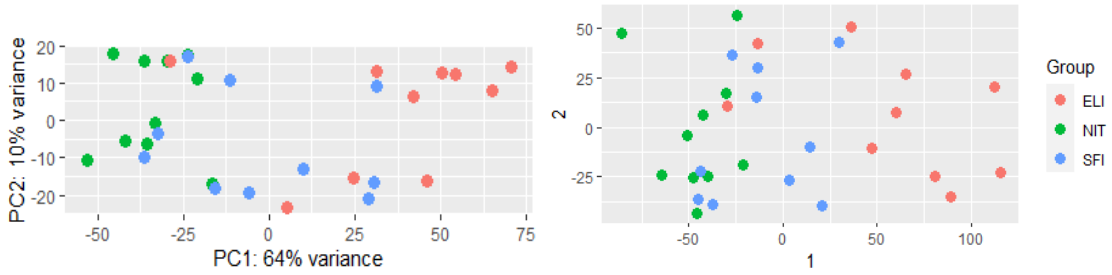


Figura 4. Gráfico PCA (izquierda) y gráfico MDS (derecha) a partir de los datos transformados por *VST*.

El análisis de componentes principales cuenta con un 64% de la variabilidad total de las muestras, contribuida principalmente por la condición de grupo. De manera general, tanto en el gráfico PCA como en el gráfico MDS, que usa matrices de distancias, las muestras del grupo `NIT` se agrupan a la izquierda (verde), las muestras del grupo `SFI` en el centro (azul) y las muestras del grupo `ELI` a la derecha (rojo), observando un leve solapamiento entre algunas de las muestras.

4.4 ANÁLISIS DE EXPRESIÓN DIFERENCIAL

```
dds <- DESeq(dds, parallel = TRUE)
```

Una vez realizada la función `DESeq` sobre los datos, los resultados de expresión diferencial referentes a cada una de las comparaciones generan un objeto de datos con las siguientes variables para cada uno de los genes: `baseMean` (es la media de conteos en el total de las muestras), `log2FoldChange` (estimación del efecto de expresión diferencial), `lfcSE` (error estándar asociado al efecto), `stat` (valor estadístico), `pvalue` (significación estadística del efecto) y `padj` (significación estadística ajustada por el método de Benjamini-Hochberg [6]).

En este análisis, la variable `Group` de estudio contiene tres niveles (`NIT`, `SFI` y `ELI`), de manera que pueden compararse dos a dos, obteniendo un total de tres comparaciones:

- `res_NITvsSFI`: expresión diferencial entre pacientes con pequeños infiltrados en el tejido tiroideo y pacientes sin infiltrados.
- `res_NITvsSFI`: expresión diferencial entre pacientes con muchos infiltrados en el tejido tiroideo y pacientes sin infiltrados.
- `res_SFIvsELI`: expresión diferencial entre pacientes con muchos infiltrados en el tejido tiroideo y pacientes con pequeños infiltrados.

A continuación, se muestra para cada una de las comparaciones: el código para la obtención de los resultados y su resumen, una tabla con los cinco genes más infra y sobreexpresados con un p-valor ajustado inferior a 0.05 y, finalmente, un gráfico MA que muestra la distribución de los coeficientes y el histograma de los p-valores.

EXPRESIÓN DIFERENCIAL ENTRE LOS GRUPOS `NIT` Y `SFI`:

```
res_NITvsSFI <- results(dds, contrast=c("Group","NIT","SFI"),
                              alpha=0.05)
summary(res_NITvsSFI)

##
## out of 35890 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 16, 0.045%
## LFC < 0 (down)     : 285, 0.79%
## outliers [1]       : 0, 0%
## low counts [2]     : 7658, 21%
## (mean count < 2)
```

En la comparación `res_NITvsSFI`, que compara pacientes sin infiltrados en el tejido tiroideo con pacientes con pequeños infiltrados, se observan 16 genes sobreexpresados y 285 genes infraexpresados. Teniendo en cuenta los genes con un p-valor ajustado inferior al 5%, es posible ordenar los genes de mayor a menor, y viceversa, según el efecto de la expresión diferencial y así obtener los genes más sobreexpresados y más infraexpresados:

```
resSig_NITvsSFI <- subset(res_NITvsSFI, padj <= 0.05)

head(resSig_NITvsSFI[ order(resSig_NITvsSFI$log2FoldChange), ])

head(resSig_NITvsSFI[ order(resSig_NITvsSFI$log2FoldChange, decreasing
= TRUE), ])
```

Tabla 1. Genes diferencialmente expresados en la comparación `NITvsSFI`.

	Base Mean	log2 FoldChange	lfcSE	stat	pvalue	padj
	Genes más infraexpresados					
ENSG00000211930.1	18,1191	-8,3203	1,5900	-5,2328	1,6694E-07	5,1798E-05
ENSG00000211670.2	228,1669	-8,2464	1,1414	-7,2247	5,0228E-13	1,7728E-09
ENSG00000223648.2	80,7691	-7,9195	1,2751	-6,2108	5,2729E-10	3,1678E-07
ENSG00000211951.2	224,7878	-7,9054	1,1968	-6,6057	3,9563E-11	3,8603E-08
ENSG00000254395.1	73,6506	-7,8643	1,1205	-7,0185	2,2422E-12	5,7556E-09
	Genes más sobreexpresados					
ENSG00000110680.8	162,5485	4,3231	1,0103	4,2792	1,8755E-05	0,00320946
ENSG00000163501.6	139,2227	3,7789	0,8382	4,5085	6,5293E-06	0,00128028
ENSG00000240707.2	3,8690	2,9066	0,7890	3,6839	0,00022967	0,02604387
ENSG00000267131.1	9,8838	2,8145	0,5492	5,1245	2,9839E-07	8,7765E-05
ENSG00000247311.2	36,9696	2,3812	0,6851	3,4756	0,00050971	0,04862261

A continuación, se observan los resultados obtenidos para esta comparación representados en un gráfico MA, en el cual se remarca en azul el gen con un p-valor más pequeño (Figura 5A). Adicionalmente, en un histograma, se representan los p-valores de los genes que presentan una media de conteos del total de las muestras superior a 1 (Figura 5B).

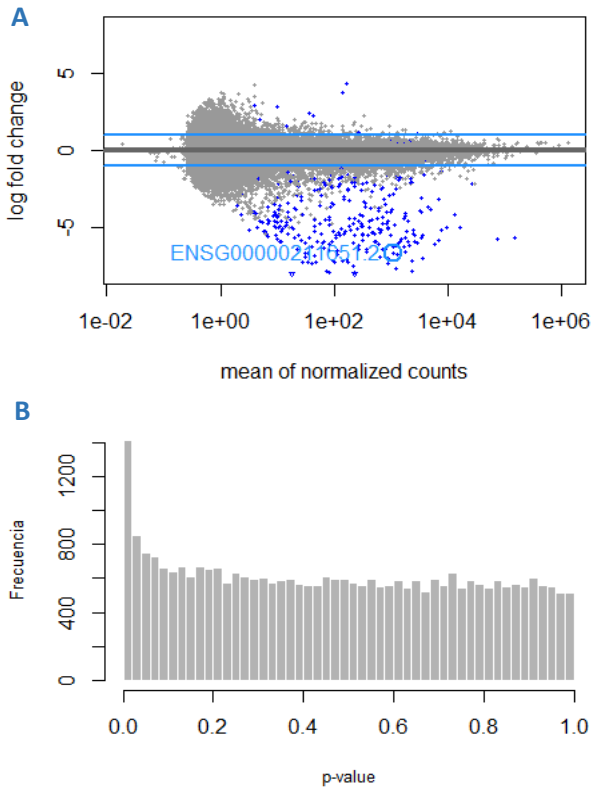


Figura 5. A. Gráfico MA de la distribución de los coeficientes estimados de la comparación `NITvsSFI`. B. Histograma de los p-valores obtenidos de los resultados de la comparación `NITvsSFI`.

Los resultados para las comparaciones `NITvsELI` y `SFIvsELI` se encuentran a continuación de forma más abreviada.

EXPRESIÓN DIFERENCIAL ENTRE LOS GRUPOS NIT Y ELI:

```
res_NITvsELI <- results(dds, contrast=c("Group","NIT","ELI"), alpha =
0.05)
summary(res_NITvsELI)

##
## out of 35890 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 1525, 4.2%
## LFC < 0 (down)     : 3405, 9.5%
## outliers [1]       : 0, 0%
## low counts [2]     : 5571, 16%
## (mean count < 1)
```

En la comparación `res_NITvsELI`, que compara pacientes sin infiltrados en el tejido tiroideo con pacientes con muchos infiltrados, se observan 1525 genes sobreexpresados y 3405 genes infraexpresados.

```
resSig_NITvsELI <- subset(res_NITvsELI, padj <= 0.05)

head(resSig_NITvsELI[ order(resSig_NITvsELI$log2FoldChange), ])

head(resSig_NITvsELI[ order(resSig_NITvsELI$log2FoldChange, decreasing
= TRUE), ])
```

Tabla 2. Genes diferencialmente expresados en la comparación `NITvsELI`.

	Base Mean	log2 FoldChange	lfcSE	stat	pvalue	padj
	Genes más infraexpresados					
ENSG00000211650.2	179,4440	-10,3362	1,3917	-7,4271	1,1101E-13	1,5953E-11
ENSG00000222037.5	898,6659	-9,4911	1,0018	-9,4738	2,698E-21	1,0226E-17
ENSG00000211640.3	1252,9245	-9,3287	1,1876	-7,8549	4,0025E-15	9,5565E-13
ENSG00000211649.2	219,9166	-9,1502	1,2123	-7,5479	4,4248E-14	6,952E-12
ENSG00000211951.2	224,7878	-8,9805	1,1965	-7,5053	6,1272E-14	9,2436E-12
	Genes más sobreexpresados					
ENSG00000215323.4	4,6904	5,2450	1,9200	2,7318	0,00629842	0,04114323
ENSG00000110680.8	162,5485	4,1505	1,0099	4,1098	3,9593E-05	0,00071166
ENSG00000079689.9	79,8719	3,9251	0,9699	4,0469	5,1902E-05	0,00089473
ENSG00000213215.1	1,7428	3,7877	1,1238	3,3704	0,00075065	0,00791988
ENSG00000163501.6	139,2227	3,6438	0,8379	4,3486	1,3701E-05	0,00028281

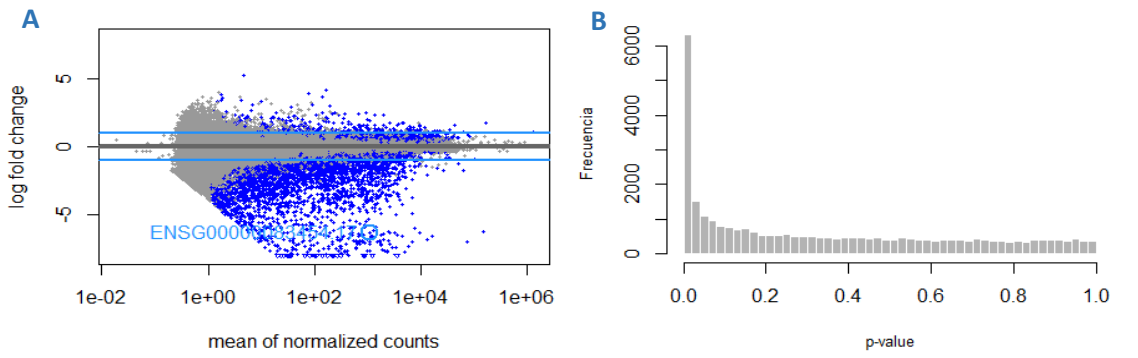


Figura 6. A. Gráfico MA de la distribución de los coeficientes estimados de la comparación `NITvsELI`. B. Histograma de los p-valores obtenidos de los resultados de la comparación `NITvsELI`.

EXPRESIÓN DIFERENCIAL ENTRE LOS GRUPOS SFI Y ELI:

```
res_SFIvsELI <- results(dds, contrast=c("Group","NIT","ELI"), alpha =
0.05)
summary(res_SFIvsELI)

##
## out of 35890 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 877, 2.4%
## LFC < 0 (down)     : 2317, 6.5%
## outliers [1]       : 0, 0%
## low counts [2]     : 6963, 19%
## (mean count < 2)
```

En la comparación `res_SFIvsELI`, que compara pacientes con pequeños infiltrados en el tejido tiroideo con pacientes con muchos infiltrados, se observan 877 genes sobreexpresados y 2317 genes infraexpresados.

```
resSig_SFIvsELI <- subset(res_SFIvsELI, padj <= 0.05)

head(resSig_SFIvsELI[ order(resSig_SFIvsELI$log2FoldChange), ])

head(resSig_SFIvsELI[ order(resSig_SFIvsELI$log2FoldChange, decreasing
= TRUE), ])
```

Tabla 3. Genes diferencialmente expresados en la comparación `NITvsELI`.

	Base Mean	log2 FoldChange	lfcSE	stat	pvalue	padj
	Genes más infraexpresados					
ENSG00000160505.11	13,5083	-6,3646	1,3794	-4,6141	3,95E-06	0,0003
ENSG00000170054.10	39,5494	-6,0115	1,4472	-4,1538	3,27E-05	0,0012
ENSG00000162897.10	6,8567	-5,5946	1,0444	-5,3565	8,48E-08	1,46E-05
ENSG00000221971.3	5,4651	-5,3438	0,8902	-6,0031	1,94E-09	1,13E-06
ENSG00000224610.1	2,1316	-5,0891	1,0792	-4,7154	2,41E-06	0,0002
	Genes más sobreexpresados					
ENSG00000250033.1	58,7463	4,2993	1,0105	4,2546	2,09E-05	0,0009
ENSG00000151012.9	2025,8787	4,0472	0,7439	5,4406	5,31E-08	1,05E-05
ENSG00000079689.9	79,8719	3,9729	0,9697	4,0968	4,19E-05	0,0014
ENSG00000174417.2	8,7877	3,7326	0,9890	3,7742	0,0002	0,0039
ENSG00000100604.8	94,1263	3,3162	1,0319	3,2138	0,0013	0,0181

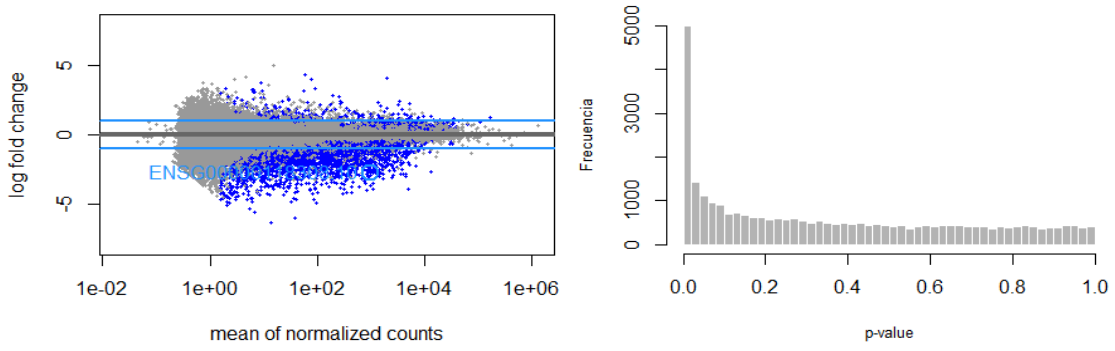


Figura 7. A. Gráfico MA de la distribución de los coeficientes estimados de la comparación `SFIvsELI`. B. Histograma de los p-valores obtenidos de los resultados de la comparación `SFIvsELI`.

4.5 AGRUPACIÓN DE MUESTRAS

A continuación, en la Figura 8, se observan representados los veinte genes con más varianza en cada una de las muestras en un mapa de calor. Las muestras y los genes se agrupan en dendogramas y se observa una clara agrupación de las muestras de los grupos `NIT` y `ELI`, aunque las muestras del grupo del `SFI` no se observan agrupadas y dispersadas entre las muestras de los grupos `NIT` y `ELI`.

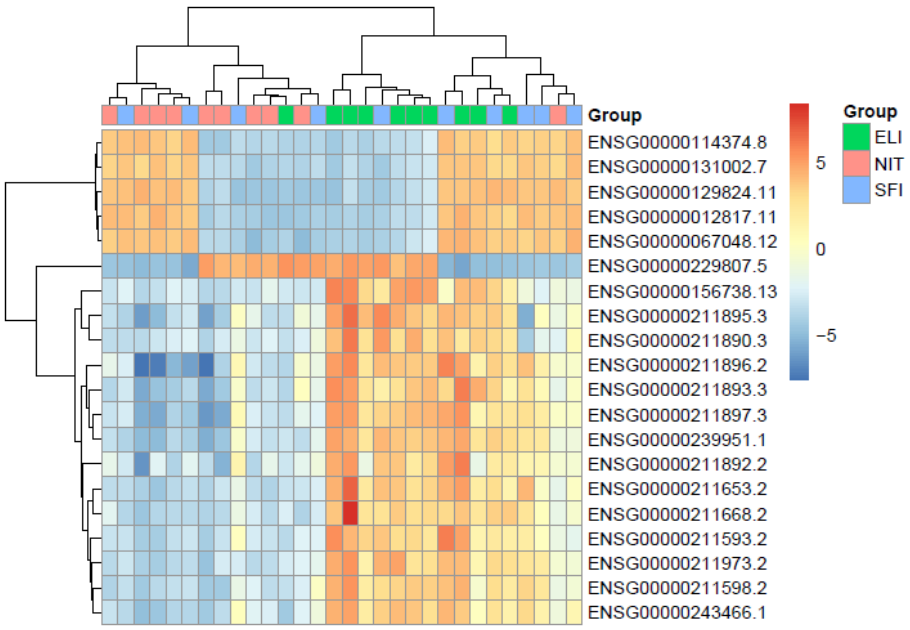


Figura 8. Mapa de calor de los veinte genes con más varianza en las 30 muestras del estudio agrupadas en dendogramas. El mapa de calor con el nombre completo de las muestras puede encontrarse en el repositorio de Github de este estudio [2].

En la figura anterior, se observan algunos de los genes con más varianza agrupados en niveles de expresión similar. Existen patrones de expresión que no se agrupan según los distintos grupos de tipos de infiltrados en la tiroides, sin embargo, el grupo de genes infraexpresados que se observa abajo a la izquierda (azul) parece estar correlacionado con el grupo `NIT`, es decir, con el tipo de tejidos que no muestra infiltrados.

El diagrama de Venn que se muestra a continuación representa una comparación entre las distintas comparaciones. Se observa que la comparación que presenta un mayor número de genes diferencialmente expresados es la comparación `NITvsELI`, que compara los tejidos sin infiltrados con los tejidos extensivamente infiltrados, siendo 2334 genes diferencialmente expresados exclusivos de esta comparación.

Cabe destacar los 4630 genes diferencialmente expresados que son comunes de las comparaciones `NITvsELI` y `SFIvsELI`, más los 201 genes diferencialmente expresados comunes entre las tres comparaciones. También es interesante observar que de los genes diferencialmente expresados en la comparación `NITvsSFI`, tan solo 14 son exclusivos de esta comparación y 428 son comunes con la comparación `NITvsELI`.

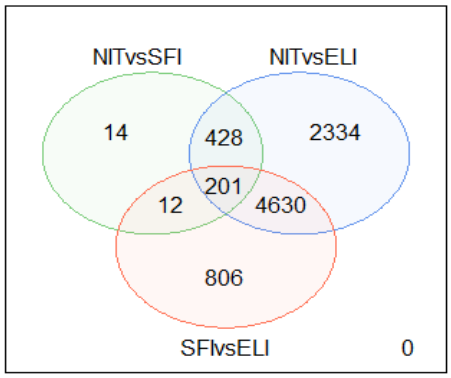


Figura 9. Diagrama de Venn de los genes diferencialmente expresados comunes entre las tres comparaciones estudiadas.

4.6 ANOTACIÓN DE LOS RESULTADOS

Este paso permite asociar los genes diferencialmente expresados determinados en el análisis de expresión diferencial a los identificadores `SYMBOL`, `ENTREZID` y `GENENAME`, que facilitan la interpretación de los resultados. A continuación, se muestran los seis primeros resultados obtenidos de cada una de las comparaciones ordenados de menor a mayor p-valor y excluyendo los valores `NA` (*not available*), seguidos de su representación mediante un *volcano plot*.

GENES DIFERENCIALMENTE EXPRESADOS ANOTADOS DE LA COMPARACIÓN `NITvsSFI`:

Tabla 4. Genes diferencialmente expresados anotados de la comparación `NITvsSFI`.

	log2FC	pvalue	symbol	entrez	genename
ENSG00000132465	-5,14045	1,43E-10	JCHAIN	3512	joining chain of multimeric IgA and IgM
ENSG00000105369	-5,28248	4,08E-10	CD79A	973	CD79a molecule
ENSG00000170476	-4,86043	5,89E-10	MZB1	51237	marginal zone B and B1 cell specific protein
ENSG00000110777	-4,59036	6,54E-09	POU2AF1	5450	POU class 2 homeobox associating factor 1
ENSG00000117322	-5,07437	9,89E-08	CR2	1380	complement C3d receptor 2
ENSG00000143297	-4,39256	1,22E-07	FCRL5	83416	Fc receptor like 5

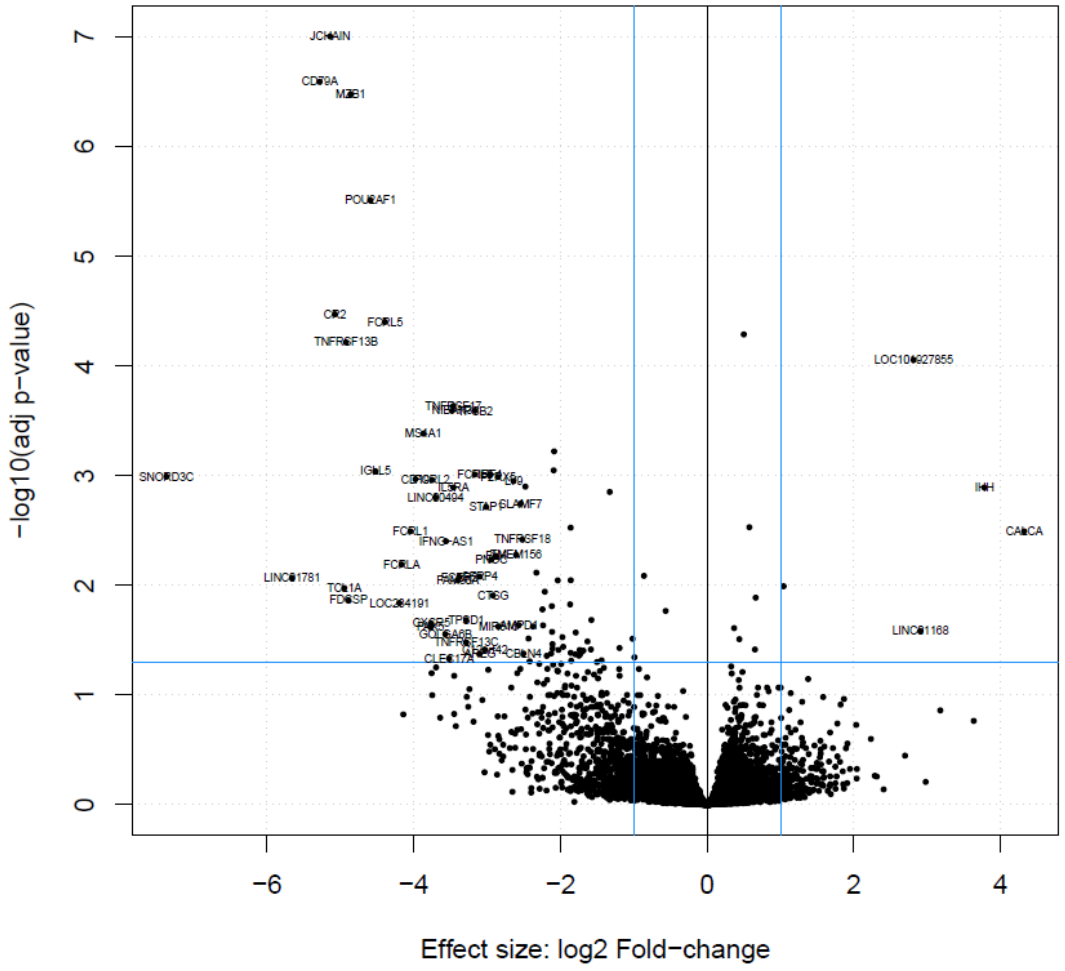


Figura 10. Representación de los genes diferencialmente expresados anotados de la comparación `NITvsSFI`. Mayormente, se observan genes infraexpresados en el grupo `SFI` respecto al grupo `NIT`.

GENES DIFERENCIALMENTE EXPRESADOS ANOTADOS DE LA COMPARACIÓN `NITvsELI`:

Tabla 5. Genes diferencialmente expresados anotados de la comparación `NITvsELI`.

	log2FC	pvalue	symbol	entrez	genename
ENSG00000083454	-6,31926	3,79E-24	P2RX5	5026	purinergic receptor P2X 5
ENSG00000136573	-6,79868	1,18E-22	BLK	640	BLK proto-oncogene, Src family tyrosine kinase
ENSG00000167483	-6,94637	1,25E-22	NIBAN3	199786	niban apoptosis regulator 3
ENSG00000156738	-7,76238	7,95E-22	MS4A1	931	membrane spanning 4-domains A1
ENSG00000035720	-6,48581	1,33E-21	STAP1	26228	signal transducing adaptor family member 1
ENSG00000173200	-4,59398	2,17E-21	PARP15	165631	poly(ADP-ribose) polymerase family member 15

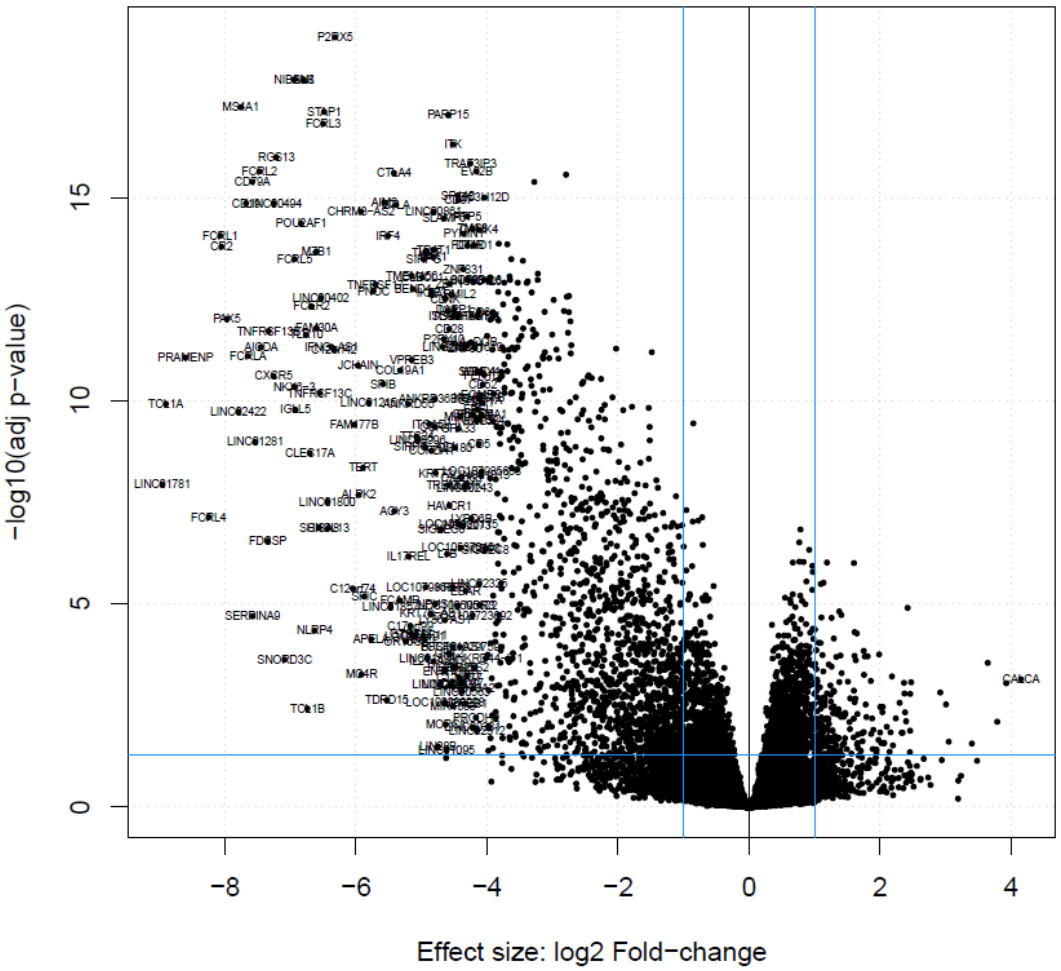


Figura 11. Representación de los genes diferencialmente expresados anotados de la comparación `NITvsELI`. En esta comparación, se observan una gran cantidad de genes infraexpresados en el grupo `ELI` respecto al grupo `NIT`, respecto a la comparación anterior, y con un p-valor ajustado muy menor.

GENES DIFERENCIALMENTE EXPRESADOS ANOTADOS DE LA COMPARACIÓN SFIvsELI:

Tabla 6. Genes diferencialmente expresados anotados de la comparación SFIvsELI.

	log2FC	pvalue	symbol	entrez	genename
ENSG00000118308	-2,76216	1,8E-14	LRMP	4033	lymphoid restricted membrane protein
ENSG00000069493	-2,25105	3,13E-13	CLEC2D	29121	C-type lectin domain family 2 member D
ENSG00000074966	-3,02665	9,8E-13	TXK	7294	TXK tyrosine kinase
ENSG00000196172	-1,16607	3,49E-12	ZNF681	148213	zinc finger protein 681
ENSG00000111913	-2,47621	3,67E-12	RIPOR2	9750	RHO family interacting cell polarization regulator 2
ENSG00000068831	-2,87925	4,41E-12	RASGRP2	10235	RAS guanyl releasing protein 2

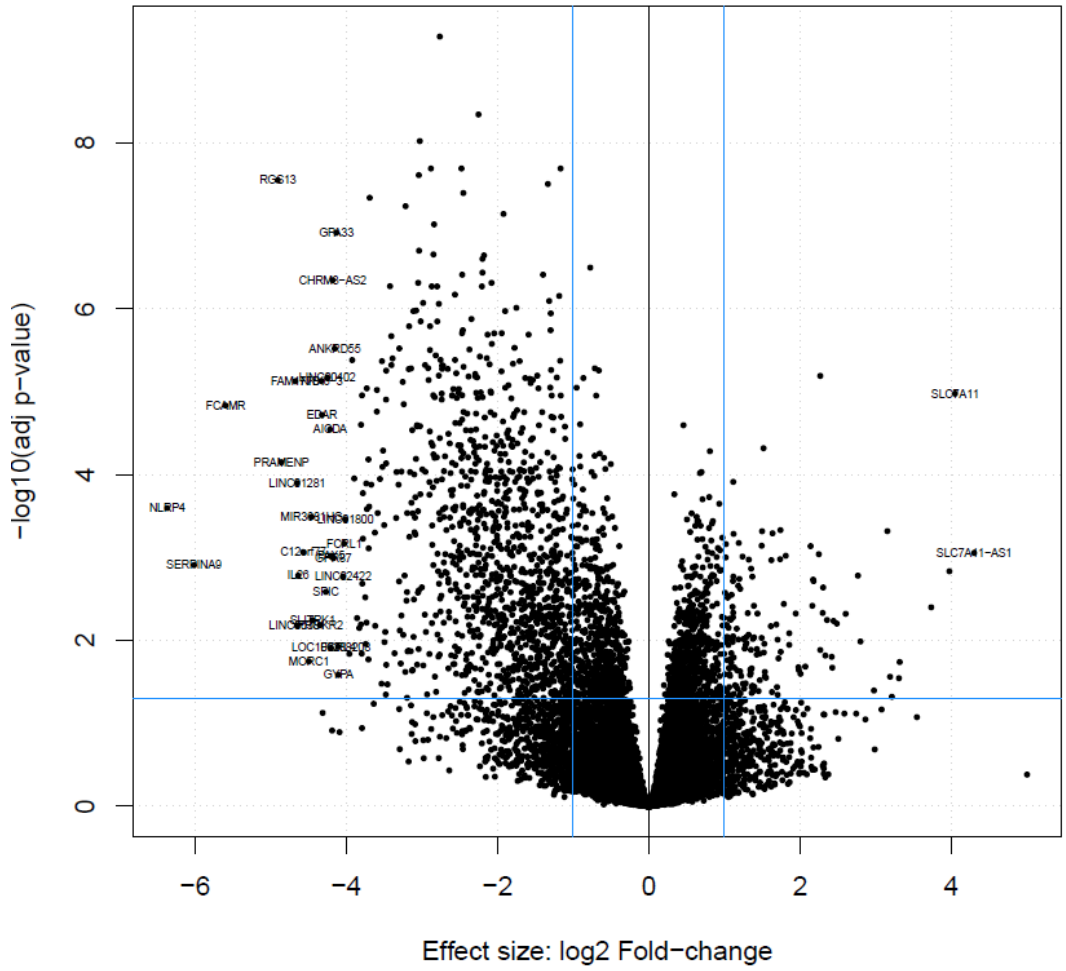


Figura 12. Representación de los genes diferencialmente expresados anotados de la comparación SFIvsELI. En esta comparación, se observan principalmente genes infraexpresados en el grupo ELI respecto al grupo SFI, con p-valores ajustados más similares a la comparación NITvsSFI.

4.7 ANÁLISIS DE SIGNIFICACIÓN BIOLÓGICA

El análisis de significación biológica proporciona para cada comparación un archivo *.csv* y otro archivo *.xlsx* con un resumen de las vías metabólicas enriquecidas, un diagrama de puntos `dotplot` y un gráfico `cnetplot` con las vías metabólicas enriquecidas y una red `emapplot` con los procesos funcionales enriquecidos. Estos archivos y gráficos pueden encontrarse en el repositorio Github correspondiente a este análisis [2].

A continuación, se muestran los seis primeros resultados del análisis de las vías metabólicas enriquecidas ordenadas por el p-valor y su representación mediante *dotplots* para cada una de las comparaciones:

ANÁLISIS DE ENRIQUECIMIENTO DE LA COMPARACIÓN `NITvsSFI`:

Tabla 7. Procesos biológicos (BP) diferencialmente expresados en la comparación `NITvsSFI`.

ID	Description	pvalue	p.adjust	Count
GO:0030098	lymphocyte differentiation	3,1172E-12	4,8098E-09	16
GO:0042113	B cell activation	6,3164E-12	4,8731E-09	15
GO:0042100	B cell proliferation	1,2832E-11	6,6E-09	10
GO:0050853	B cell receptor signaling pathway	2,7659E-10	1,0669E-07	10
GO:0070661	leukocyte proliferation	6,5459E-10	2,0201E-07	13
GO:0050851	antigen receptor-mediated signaling pathway	1,3373E-09	3,439E-07	13

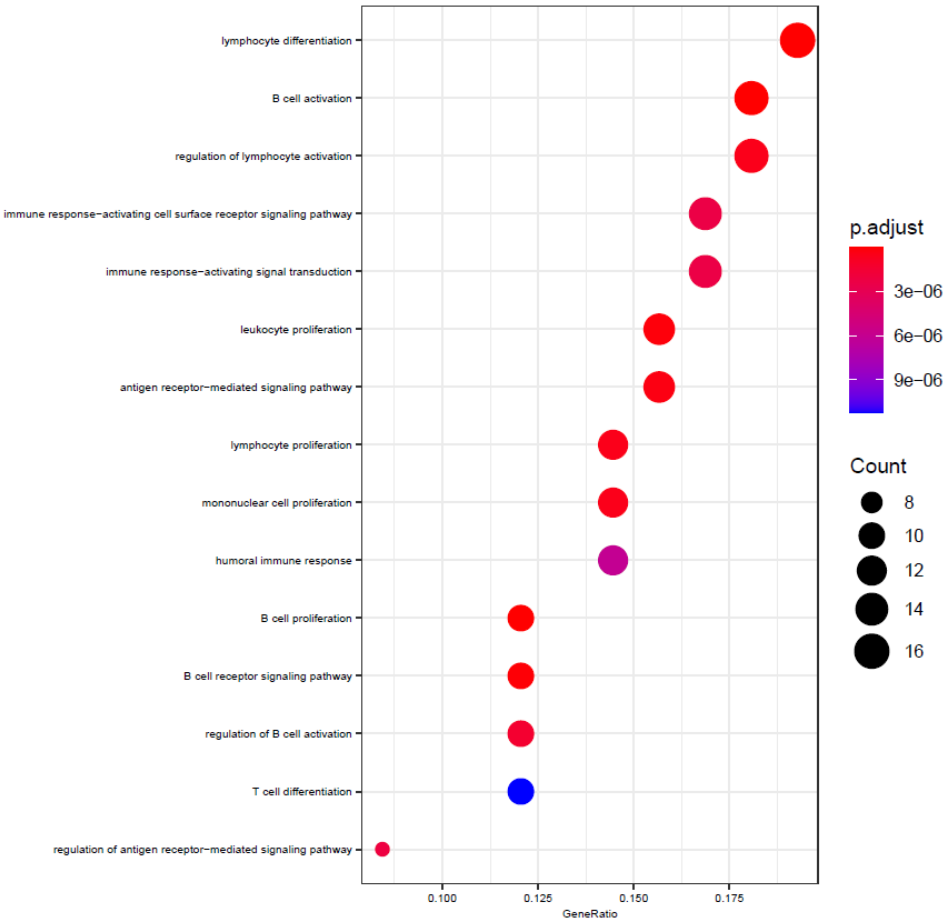


Figura 13. Visualización de las vías metabólicas enriquecidas en la comparación `NITvsSFI`.

ANÁLISIS DE ENRIQUECIMIENTO DE LA COMPARACIÓN NITvsELI:

Tabla 8. Procesos biológicos (BP) diferencialmente expresados en la comparación NITvsELI.

ID	Description	pvalue	p.adjust	Count
GO:0042110	T cell activation	1,2667E-40	7,9383E-37	196
GO:0030098	lymphocyte differentiation	2,023E-33	6,3391E-30	153
GO:0050863	regulation of T cell activation	6,9351E-33	1,4488E-29	141
GO:0051249	regulation of lymphocyte activation	7,2284E-29	1,1325E-25	180
GO:0007159	leukocyte cell-cell adhesion	2,3066E-28	2,8911E-25	140
GO:0022407	regulation of cell-cell adhesion	1,4326E-27	1,4964E-24	156

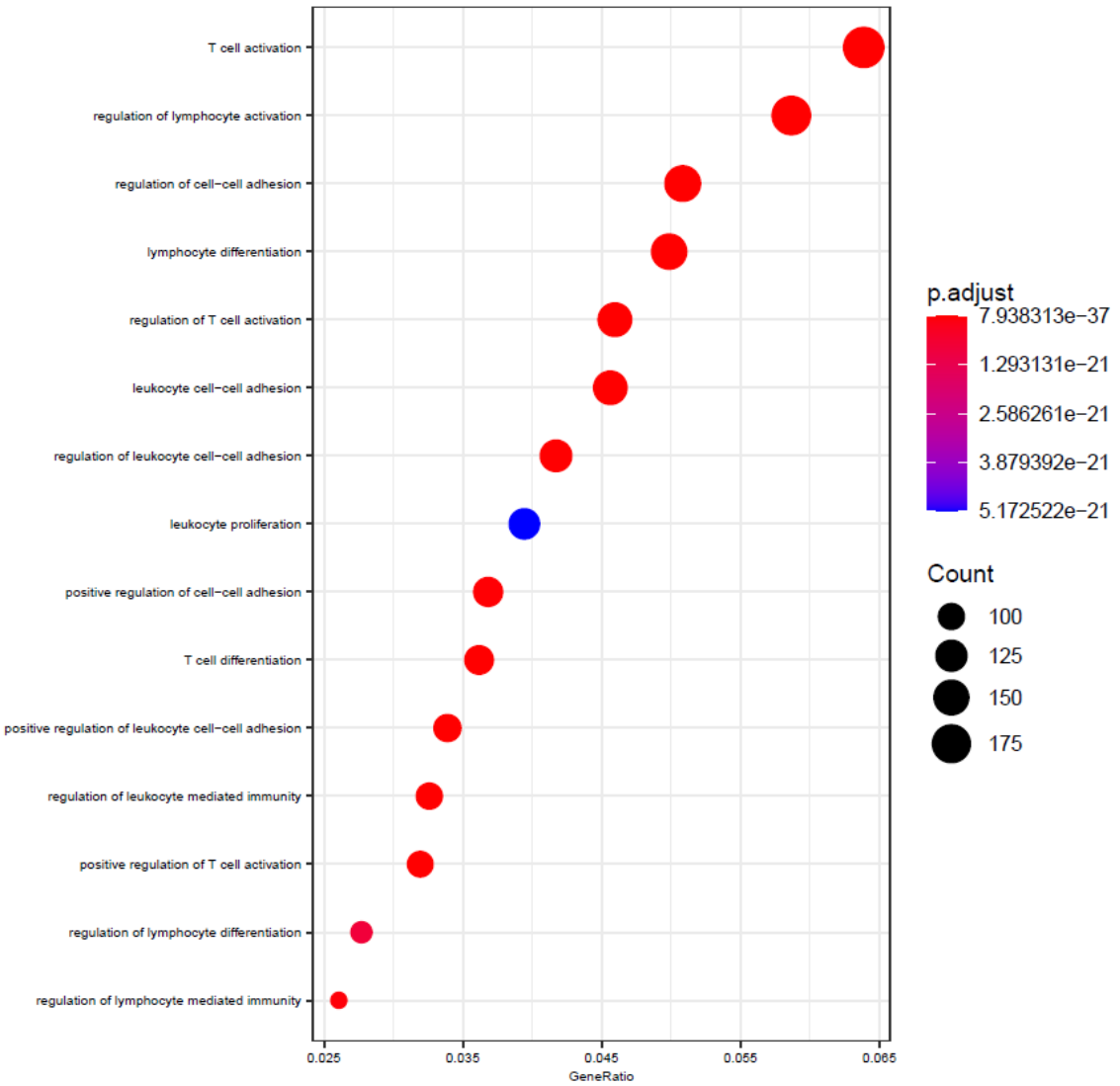


Figura 14. Visualización de las vías metabólicas enriquecidas en la comparación NITvsELI.

ANÁLISIS DE ENRIQUECIMIENTO DE LA COMPARACIÓN SFIvsELI:

Tabla 9. Procesos biológicos (BP) diferencialmente expresados en la comparación SFIvsELI.

ID	Description	pvalue	p.adjust	Count
GO:0042110	T cell activation	8,2116E-34	4,9409E-30	146
GO:0030098	lymphocyte differentiation	6,4017E-29	1,926E-25	116
GO:0050863	regulation of T cell activation	5,6301E-27	1,1292E-23	105
GO:0030217	T cell differentiation	5,665E-24	8,5216E-21	85
GO:0007159	leukocyte cell-cell adhesion	4,9109E-23	5,9098E-20	103
GO:0051249	regulation of lymphocyte activation	5,4665E-21	5,1888E-18	126

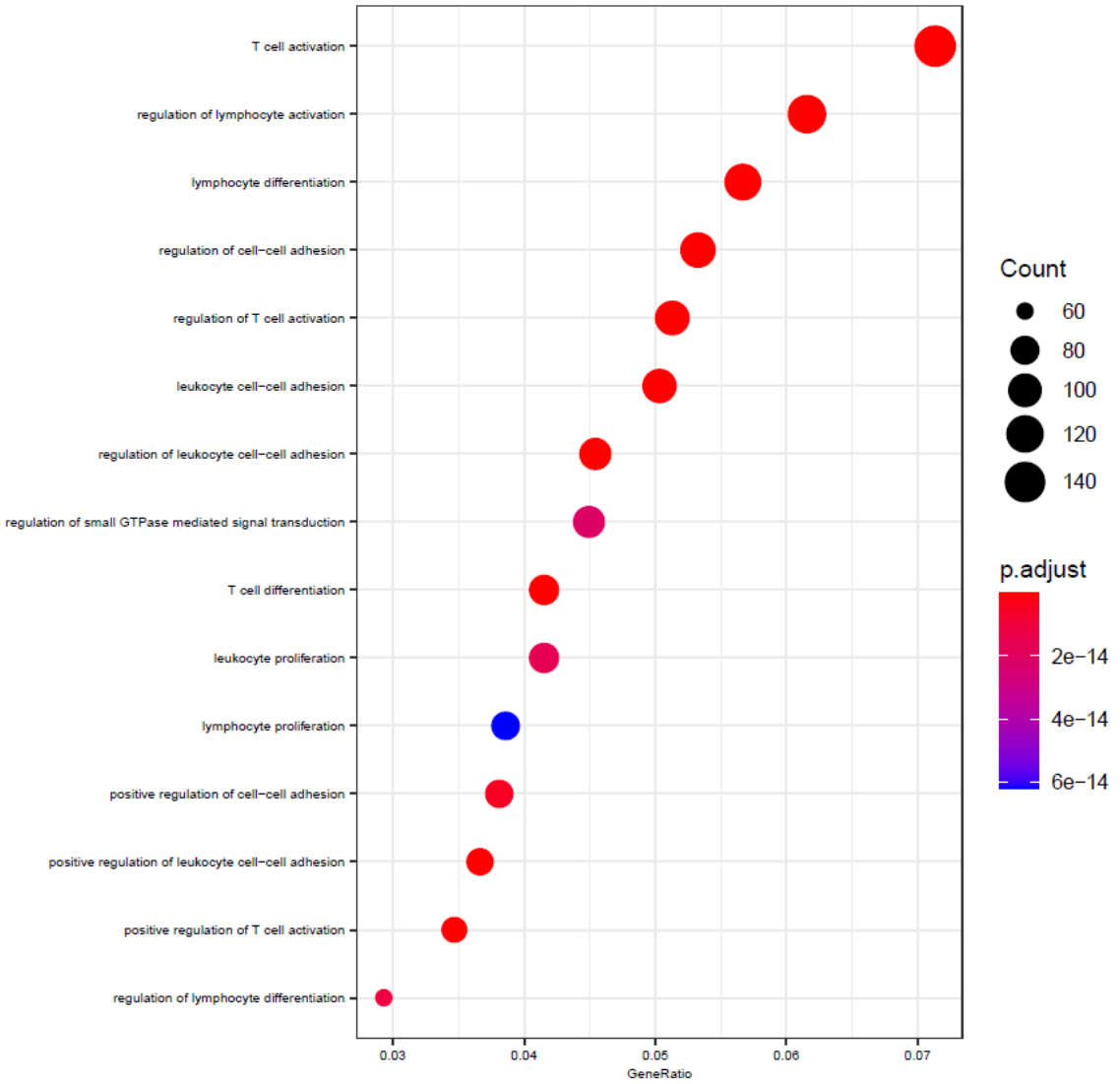


Figura 15. Visualización de las vías metabólicas enriquecidas en la comparación SFIvsELI.

5. DISCUSIÓN

La secuenciación del ARN es una tecnología basada en técnicas de secuenciación masiva que, por su capacidad de determinar la cantidad de ARN en una muestra biológica en condiciones concretas, se ha convertido en una de las técnicas más relevantes para el análisis del transcriptoma y la identificación de genes diferencialmente expresados.

Actualmente, existen muchos métodos y software diferentes para realizar análisis de expresión diferencial a partir de datos de RNA. Algunos de los software considerados más óptimos por su precisión y sensibilidad son los paquetes `DESeq2`, `Limma` y `NOIseq` [7]. En este estudio, el análisis ha sido realizado mediante `DESeq2`, basado en una distribución binomial negativa, con el objetivo de incrementar el conocimiento de este paquete por la estudiante y entender así su aplicabilidad cuando se dispone de matrices de conteos a partir de datos de ultrasecuenciación.

La infiltración de células en la glándula tiroidea puede ocurrir como una anormalidad puntual o debido a la manifestación de una enfermedad generalizada. En este informe se realiza un análisis de expresión diferencial entre tres tipos distintos de tejido tiroideo según el nivel de infiltración.

Primeramente, del total de muestras obtenidas por el proyecto GTEx, diez muestras de cada grupo han sido escogidas de manera aleatoria. Siendo el número total de muestras de cada grupo muy diferente (236 grupo `NIT`, 42 grupo `SFI` y 14 grupo `ELI`), este paso condiciona significativamente los resultados del presente análisis. La variación del grupo `ELI` es mucho menor al realizar el análisis con muestras aleatorias distintas, mientras que la variación de los resultados con muestras aleatorias distintas del grupo `NIT` puede implicar resultados muy diversos. No obstante, este paso permite la reducción del tamaño de los objetos y el incremento en velocidad para la realización de los distintos pasos del análisis.

En referencia a la visualización inicial de las muestras, llama la atención que las muestras no se agrupan claramente por grupos en los dendogramas de los mapas de calor (Figura 3). Generalmente, el grupo `SFI` es el que se encuentra más disperso entre los grupos `NIT` y `ELI`, como también se observa en los gráficos PCA y MDS (Figura 4). Esta observación del grupo `SFI` puede venir causada por ser el grupo con un nivel intermedio de infiltración, entre los tejidos no infiltrados `NIT` y los tejidos infiltrados extensivamente `ELI`.

El análisis de expresión diferencial entre las distintas comparaciones proporciona un relevante mayor número de genes diferencialmente expresados en la comparación `NITvsELI` respecto a las demás comparaciones. Esta observación puede determinarse claramente del diagrama de Venn y los *volcano plots* (Figuras 9-12). La explicación más razonable a estos resultados yace en que esta comparación comprende los dos grupos extremos, es decir, las muestras de tejidos sin infiltrados y las muestras de tejidos extensivamente infiltrados.

El efecto de la infiltración en los tejidos tiroideos se interpreta a nivel biológico mediante los análisis de enriquecimiento. Por un lado, este análisis muestra que en los tejidos con pequeños infiltrados aparece diferencialmente expresada la activación de la respuesta inmune primaria mediante la diferenciación y proliferación de linfocitos y células B. Por otro lado, en los tejidos extensivamente infiltrados prevalece la activación de la respuesta inmune secundaria al determinar procesos biológicos diferencialmente expresados relacionados con la activación y diferenciación de células T, tanto en comparación con los tejidos que no tienen infiltrados como en comparación con los tejidos con pequeños infiltrados. En resumen, se observa una evolución de la respuesta inmune a medida que aumenta el nivel de infiltración.

Futuros estudios podrían incluir un grupo con niveles medios de infiltración en el tejido tiroideo y adicionalmente, el análisis podría incorporar un número más representativo de muestras. Finalmente, sería interesante determinar genes diferencialmente expresados en los tejidos con pequeños infiltrados comunes en los tejidos extensivamente infiltrados potenciales de ser indicadores precoces del deterioro de la glándula tiroidea con la finalidad de obtener un diagnóstico temprano.

6. REFERENCIAS

1. Lonsdale, John, et al. "The genotype-tissue expression (GTEx) project." *Nature genetics* 45.6 (2013): 580. https://doi.org/10.1038/ng.2653.

2. Repositorio Github del análisis realizado en este informe: https://github.com/judithguitart/guitart_judith_ADO_PEC2

3. Love, Michael I., Wolfgang Huber, and Simon Anders. "Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2." *Genome biology* 15.12 (2014): 550. http://dx.doi.org/10.1186/s13059-014-0550-8.

4. Anders, Simon, and Wolfgang Huber. "Differential expression analysis for sequence count data." *Nature Precedings* (2010): 1-1. https://doi.org/10.1186/gb-2010-11-10-r106.

5. Yu, Guangchuang, et al. "clusterProfiler: an R package for comparing biological themes among gene clusters." *Omics: a journal of integrative biology* 16.5 (2012): 284-287. https://doi.org/10.1089/omi.2011.0118.

6. Benjamini, Yoav, and Yosef Hochberg. "Controlling the false discovery rate: a practical and powerful approach to multiple testing." *Journal of the Royal statistical society: series B (Methodological)* 57.1 (1995): 289-300. https://www.jstor.org/stable/2346101.

7. Costa-Silva, Juliana, Douglas Domingues, and Fabricio Martins Lopes. "RNA-Seq differential expression analysis: An extended review and a software tool." *PloS one* 12.12 (2017). https://doi.org/10.1371/journal.pone.0190152.

7. APÉNDICE: R CODE

Nota: Algunos comentarios del archivo RMarkdown (##) han sido descartados en este informe con el objetivo de facilitar la visualización del código. Pueden encontrarse en el código completo cargado en el repositorio Github de este análisis en formato *word* y *html* [2].

A1. Preparación de los datos

Primeramente, se leen los archivos targets.csv y counts.csv guardados en la carpeta 'Datos' dentro del directorio principal.

```
targets <- read.csv("./Datos/targets.csv", header = TRUE, sep =
",", row.names=1)
counts <- read.csv("./Datos/counts.csv", header = TRUE, sep = ";
", row.names = 1)
```

A continuación, se extraen 10 muestras aleatorias de cada uno de los grupos NIT, SFI y ELI del archivo '*targets.csv*' mediante funciones del paquete `dplyr` y se observa que la salida sea la deseada:

```
library(dplyr)

set.seed(3333)
targets.pec <- targets %>% group_by(Group) %>% sample_n(10) %>%
arrange(Grupo_analisis)
```

Se modifican los nombres de las columnas del archivo *counts.csv* con el fin que coincidan con los nombres de la variable `Sample_Name` de *targets.csv* mediante la función `gsub`, sustituyendo los puntos por guiones, y finalmente se genera un subconjunto del archivo *counts.csv* que contenga las mismas muestras seleccionadas aleatoriamente del archivo *targets.csv*:

```
names(counts) <- gsub(x=names(counts), pattern = "\\.",
replacement = "-")

counts.pec <- counts %>% select(one_of(as.character(targets.pec$
Sample_Name)))
```

Estos conjuntos de datos se definen con las variables `targets.pec` y `counts.pec`, que pueden encontrarse en el repositorio Github de este análisis en formato *.csv* [2]. Finalmente, se comprueba que las filas de la variable `Sample_Name` de `targets.pec` coincidan con las columnas seleccionadas de `counts.pec`:

```
all(rownames(targets.pec$Sample_Name) == colnames(counts.pec))

## [1] TRUE
```

Seguidamente, y después de instalar los paquetes necesarios, ya es posible construir el objeto *DESeqDataSet*, definido como `ddsM`:

```
library("DESeq2")

ddsM <- DESeqDataSetFromMatrix(countData = counts.pec, colData =
targets.pec, design = ~ Group)
```

A2. Preprocesado de los datos

El resultado obtenido del pre-filtrado de los datos se encuentra en el apartado *4.2* de los *Resultados* de este informe. A continuación, se muestran los códigos para las transformaciones con `vst` y `rlog` y para su representación de la primera muestra frente a la segunda. También se ha incluido el código para la representación de la normalización de los datos con factor `log2`, realizada mediante la función `estimateSizeFactors`:

```
vsd <- vst(dds, blind=FALSE)
rld <- rlog(dds, blind=FALSE)

library("ggplot2")
library("hexbin")
ddsN <- estimateSizeFactors(dds)
df <- bind_rows(
  as_data_frame(log2(counts(ddsP, normalized=TRUE)[, 1:2]+1)) %>% m
utate(transformation = "log2(x+1)"),
  as_data_frame(assay(vsd)[, 1:2]) %>% mutate(transformation = "vst
"), as_data_frame(assay(rld)[, 1:2]) %>% mutate(transformation = "r
log")
)

colnames(df)[1:2] <- c("x","y")
ggplot(df, aes(x = x, y = y)) + geom_hex(bins = 80) + coord_fixed()
+ facet_grid( . ~transformation)
```

El gráfico comparativo de transformaciones se encuentra en el apartado *4.2* de los *Resultados* de este análisis.

Seguidamente, se muestran los códigos para estudiar la distribución de los conteos de las distintas muestras mediante diagramas de caja de los datos no normalizados y normalizados:

```
library("edgeR")

logcounts <- cpm(counts.pec, log=TRUE)
boxplot(logcounts, names = c(rep(" ",30)), col = "gray90", outcol =
c(rep("palegreen3",10),rep("cornflowerblue",10),rep("coral1",10)),
ylab = "Log2 counts per million", las=2, cex.lab = 0.8)
abline(h=median(logcounts), col="red")
title("Boxplots of logCPMs (unnormalised)", cex.main = 1)
legend("topright",legend=c("NIT","SFI","ELI"), text.col=c("palegree
n3","cornflowerblue","coral1"), cex=0.8, title = "Treatment", title
.col = "black")

ddsN = estimateSizeFactors(dds)
sizeFactors(ddsN)

normcounts <- cpm(ddsN, log = TRUE)
boxplot(normcounts, names = c(rep(" ",30)), col = c(rep("palegreen3
",10),rep("cornflowerblue",10),rep("coral1",10)), ylab = "Log2 coun
ts per million", las=2, cex.lab = 0.8)
abline(h=median(normcounts), col="red")
title("Boxplots of logCPMs (normalised)", cex.main = 1)
legend("topright",legend=c("NIT","SFI","ELI"), text.col=c("palegree
n3","cornflowerblue","coral1"), cex=0.8, title = "Treatment", title
.col = "black")
```

A3. Visualización de los datos

A continuación, se muestra el código para calcular la distancia entre las muestras a partir de los datos transformados por *VST* y para la posterior representación mediante un mapa de calor:

```
sampleDists <- dist(t(assay(vsd)))

library("pheatmap")
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(vsd$Group, sep = " - ")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette(rev(brewer.pal(30, "Blues")) )(255)

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

Seguidamente se muestra el código para generar los gráficos PCA y MDS:

```
plotPCA(vsd, intgroup = c("Group"))

mds <- as.data.frame(colData(vsd)) %>% cbind(cmdscale(sampleDistMat
rix))
ggplot(mds, aes(x = `1`, y = `2`, color = Group)) + geom_point(size
=3) + coord_fixed() + ggtitle("MDS con los datos VSD")
```

A4. Patrones de expresión

A continuación, se muestra el código para la visualización de los patrones de expresión del análisis de expresión diferencial para cada una de las comparaciones. **MA-plots:**

```
DESeq2::plotMA(res_NITvsSFI, ylim = c(-8,8))
topGene1 <- rownames(res_NITvsSFI)[which.min(res_NITvsSFI$padj)]
with(res_NITvsSFI[topGene1, ], {
  points(baseMean, log2FoldChange, col = "dodgerblue", cex = 2, lwd
= 2) text(baseMean, log2FoldChange, topGene1, pos = 2, col = "dodge
rblue")
})
abline(h=c(-1,1), col="dodgerblue", lwd=2)
title("MA-plot NITvsSFI")

DESeq2::plotMA(res_NITvsELI, ylim = c(-8,8))
topGene2 <- rownames(res_NITvsELI)[which.min(res_NITvsELI$padj)]
with(res_NITvsELI[topGene2, ], {
  points(baseMean, log2FoldChange, col = "dodgerblue", cex = 2, lwd
= 2) text(baseMean, log2FoldChange, topGene2, pos = 2, col = "dodge
rblue")
})
abline(h=c(-1,1), col="dodgerblue", lwd=2)
title("MA-plot NITvsELI")

DESeq2::plotMA(res_SFIvsELI, ylim = c(-8,8))
topGene3 <- rownames(res_SFIvsELI)[which.min(res_SFIvsELI$padj)]
with(res_SFIvsELI[topGene3, ], {
  points(baseMean, log2FoldChange, col = "dodgerblue", cex = 2, lwd
= 2) text(baseMean, log2FoldChange, topGene3, pos = 2, col = "dodge
rblue")
})
abline(h=c(-1,1), col="dodgerblue", lwd=2)
title("MA-plot SFIvsELI")
```

Histograma de los p-valores:

```
hist(res_NITvsSFI$pvalue[res_NITvsSFI$baseMean > 1], breaks = 0:50/
50, col = "grey70", border = "white", main = "Histograma de p-valor
es NITvsSFI", cex.main = 1, ylab = "Frecuencia", xlab = "p-value",
cex.lab = 0.8)

hist(res_NITvsELI$pvalue[res_NITvsELI$baseMean > 1], breaks = 0:50/
50, col = "grey70", border = "white", main = "Histograma de p-valor
es NITvsELI", cex.main = 1, ylab = "Frecuencia", xlab = "p-value",
cex.lab = 0.8)

hist(res_SFIvsELI$pvalue[res_SFIvsELI$baseMean > 1], breaks = 0:50/
50, col = "grey70", border = "white", main = "Histograma de p-valor
es SFIvsELI", cex.main = 1, ylab = "Frecuencia", xlab = "p-value",
cex.lab = 0.8)
```

A5. Agrupación de las muestras

Código para la generación del **mapa de calor** de los veinte genes que presentan una mayor varianza agrupando las muestras jerárquicamente:

```
library("genefilter")

topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE),
20)
topVarGenes

##  [1] 35151 24617 35801 35849 35828 24614 24618 35826 35846 24609
4539 33862
## [13] 33899 19750 24748  4517  4520  4524 24608 24603

mat <- assay(vsd)[topVarGenes, ]
mat <- mat - rowMeans(mat)
anno <- as.data.frame(colData(vsd)["Group"])
pheatmap1 <- pheatmap(mat, annotation_col = anno)
```

Código para comparar las distintas comparaciones y para su representación mediante un **diagrama de Venn**:

```
res_NITvsSFI.genes <- row.names(resSig_NITvsSFI)
res_NITvsELI.genes <- row.names(resSig_NITvsELI)
res_SFIvsELI.genes <- row.names(resSig_SFIvsELI)

comb <- c(res_NITvsSFI.genes,res_NITvsELI.genes,res_SFIvsELI.genes)

res_NITvsSFI.genes.2 <- comb %in% res_NITvsSFI.genes
res_NITvsELI.genes.2 <- comb %in% res_NITvsELI.genes
res_SFIvsELI.genes.2 <- comb %in% res_SFIvsELI.genes

venn_counts <- cbind(res_NITvsSFI.genes.2, res_NITvsELI.genes.2, re
s_SFIvsELI.genes.2)
venn_counts_results <- vennCounts(venn_counts)
vennDiagram(venn_counts_results, cex = 1, names=c("NITvsSFI","NITvs
ELI","SFIvsELI"), circle.col=c("palegreen3","cornflowerblue","coral
1"))
```

A6. Anotación de los resultados

Inicialmente se cargan los paquetes necesarios y se elimina la versión de los identificadores Ensemble que aparece después del último punto, el cual también se elimina:

```
library("AnnotationDbi")
library("org.Hs.eg.db")

row.names(res_NITvsSFI) <- gsub(x=row.names(res_NITvsSFI), pattern
= "\\..*", replacement = "")
row.names(res_NITvsELI) <- gsub(x=row.names(res_NITvsELI), pattern
= "\\..*", replacement = "")
row.names(res_SFIvsELI) <- gsub(x=row.names(res_SFIvsELI), pattern
= "\\..*", replacement = "")
```

A continuación, se añaden las columnas `SYMBOL`, `ENTREZID` y `GENENAME` a cada una de las tablas de resultados:

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

res_NITvsSFI$entrez <- mapIds(org.Hs.eg.db, keys=row.names(res_NITv
sSFI), column = "ENTREZID", keytype="ENSEMBL", multiVals = "first")

res_NITvsSFI$genename <- mapIds(org.Hs.eg.db, keys=row.names(res_NI
TvsSFI), column = "GENENAME", keytype="ENSEMBL", multiVals = "first
")

res_NITvsSFI_Ordered <- res_NITvsSFI[order(res_NITvsSFI$pvalue),]
res_NITvsSFI_Annot <- res_NITvsSFI_Ordered[which(res_NITvsSFI_Order
ed$symbol != "NA"), ]
```

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

res_NITvsELI$entrez <- mapIds(org.Hs.eg.db, keys=row.names(res_NITv
sELI), column = "ENTREZID", keytype="ENSEMBL", multiVals = "first")

res_NITvsELI$genename <- mapIds(org.Hs.eg.db, keys=row.names(res_NI
TvsELI), column = "GENENAME", keytype="ENSEMBL", multiVals = "first
")

res_NITvsELI_Ordered <- res_NITvsELI[order(res_NITvsELI$pvalue),]
res_NITvsELI_Annot <- res_NITvsELI_Ordered[which(res_NITvsELI_Order
ed$symbol != "NA"), ]
```

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

res_SFIvsELI$entrez <- mapIds(org.Hs.eg.db, keys=row.names(res_SFIv
sELI), column = "ENTREZID", keytype="ENSEMBL", multiVals = "first")

res_SFIvsELI$genename <- mapIds(org.Hs.eg.db, keys=row.names(res_SF
IvsELI), column = "GENENAME", keytype="ENSEMBL", multiVals = "first
")

res_SFIvsELI_Ordered <- res_SFIvsELI[order(res_SFIvsELI$pvalue),]
res_SFIvsELI_Annot <- res_SFIvsELI_Ordered[which(res_SFIvsELI_Order
ed$symbol != "NA"), ]
```

Código para la representación mediante ***volcano plots***:

```
genesymbols1 <- res_NITvsSFI_Annot$symbol
plot(res_NITvsSFI_Annot$log2FoldChange, -log10(res_NITvsSFI_Annot$p
adj), panel.first = grid(), main = "Volcano plot NITvsSFI", xlab="E
ffect size: log2 Fold-change", ylab="-log10(adj p-value)", pch=20,
cex=0.6)
abline(v=0)
abline(v=c(-1,1), col="dodgerblue")
abline(h=-log10(0.05), col="dodgerblue")
gn.selected <- abs(res_NITvsSFI_Annot$log2FoldChange) > 2.5 & res_N
ITvsSFI_Annot$padj < 0.05
text(res_NITvsSFI_Annot$log2FoldChange[gn.selected],
     -log10(res_NITvsSFI_Annot$padj)[gn.selected],
     lab=genesymbols1[gn.selected], cex=0.4)
```

```
genesymbols2 <- res_NITvsELI_Annot$symbol
plot(res_NITvsELI_Annot$log2FoldChange, -log10(res_NITvsELI_Annot$p
adj), panel.first = grid(), main = "Volcano plot NITvsELI", xlab="E
ffect size: log2 Fold-change", ylab="-log10(adj p-value)", pch=20,
cex=0.6)
abline(v=0)
abline(v=c(-1,1), col="dodgerblue")
abline(h=-log10(0.05), col="dodgerblue")
gn.selected <- abs(res_NITvsELI_Annot$log2FoldChange) > 4 & res_NIT
vsELI_Annot$padj < 0.05
text(res_NITvsELI_Annot$log2FoldChange[gn.selected],
     -log10(res_NITvsELI_Annot$padj)[gn.selected],
     lab=genesymbols2[gn.selected], cex=0.4)
```

```
genesymbols3 <- res_SFIvsELI_Annot$symbol
plot(res_SFIvsELI_Annot$log2FoldChange, -log10(res_SFIvsELI_Annot$p
adj), panel.first = grid(), main = "Volcano plot SFIvsELI", xlab="E
ffect size: log2 Fold-change", ylab="-log10(adj p-value)", pch=20,
cex=0.6)
abline(v=0)
abline(v=c(-1,1), col="dodgerblue")
abline(h=-log10(0.05), col="dodgerblue")
gn.selected <- abs(res_SFIvsELI_Annot$log2FoldChange) > 4 & res_SFI
vsELI_Annot$padj < 0.05
text(res_SFIvsELI_Annot$log2FoldChange[gn.selected],
     -log10(res_SFIvsELI_Annot$padj)[gn.selected],
     lab=genesymbols3[gn.selected], cex=0.4)
```

A7. Análisis de significación biológica

Para realizar el análisis de significación, se ha creado una función para aplicar el análisis a las tres comparaciones realizadas en este estudio y a su vez, exportar los resultados en formato *.csv* y *.xlsx*, y en dos gráficos; un *doplot* y un *emapplot*:

```
library(clusterProfiler)

GO_NITvsSFI <- as.data.frame(res_NITvsSFI_Annot)
GO_NITvsELI <- as.data.frame(res_NITvsELI_Annot)
GO_SFIvsELI <- as.data.frame(res_SFIvsELI_Annot)

library("dplyr")
library("xlsx")
universe_prova <- list(NITvsSFI = GO_NITvsSFI$entrez, NITvsELI = GO
_NITvsELI$entrez, SFIvsELI = GO_SFIvsELI$entrez)
func <- function(x) {
  x %>% filter(padj < 0.05, !is.na(entrez)) %>% pull(entrez)
}
sigGenes_prova <- list(NITvsSFI = GO_NITvsSFI, NITvsELI = GO_NITvsE
LI, SFIvsELI = GO_SFIvsELI) %>% lapply(func)
comparisonsNames <- names(sigGenes_prova)

for (i in 1:length(sigGenes_prova)){
  genesIn <- sigGenes_prova[[i]]
  comparison <- comparisonsNames[i]
  enrich.result <- enrichGO(gene = genesIn, OrgDb = org.Hs.eg.db, o
nt = "ALL", pAdjustMethod = "BH", pvalueCutoff = 0.05, universe = u
niverse_prova, readable = TRUE)

cat("###################################")
 cat("\nComparison: ", comparison,"\n")
 print(head(enrich.result))

if (length(rownames(enrich.result@result)) != 0) {
 write.csv(as.data.frame(enrich.result),
            file =paste0("./Resultados/","Enrich.Results.",comparis
on,".csv"),
            row.names = FALSE)

write.xlsx(as.data.frame(enrich.result),
            file =paste0("./Resultados/","Enrich.Results.",comparis
on,".xlsx"),
            row.names = FALSE)

 pdf(file=paste0("./Resultados/","Enrich.Dotplot.",comparison,".pdf
"))
   print(dotplot(enrich.result, showCategory = 15, font.size = 6,
           title = paste0("EnrichGO Pathway Analysis for ", compari
son,". Dotplot")))
 dev.off()

 pdf(file = paste0("./Resultados/","EnrichGOemapplot.",comparison,"
.pdf"))
   print(emapplot(enrich.result, categorySize = "geneNum", schowCat
egory = 15,
          vertex.label.cex = 0.75))
 dev.off()

 }
}
```