14 DE JUNIO DE 2020

PEC 2: ANÁLISIS DE DATOS ÓMICOS

MÁSTER INTERUNIVERSITARIO DE BIOESTADÍSTICA Y BIOINFORMÁTICA

ARCHIVOS SOBRE EL ESTUDIO DISPONIBLES EN:
https://github.com/mariarm23/PEC-2-Analisis-de-datos-omicos

MARÍA RODRÍGUEZ MERCHÁN

ÍNDICE.

1. ABSTRACT.

En este trabajo de evaluación continua se realiza un análisis de expresión génica comparando los niveles de expresión de un estudio obtenido del repositorio GTEx (Genotype-Tissue Expression). En el estudio se comparan muestras pertenecientes a un análisis de tiroides donde se comparan 292 muestras pertenecientes a tres grupos: Not infiltrated tissues (NIT), Small focal infiltrates (SFI) y Extensive lymphoid infiltrates (ELI).

2. OBJETIVOS.

El objetivo de este estudio es ilustrar el proceso de análisis de datos de ultrasecuenciación mediante la realización de un estudio, de principio a fin, tal como se llevará a cabo en una situación real.

3. MATERIALES Y MÉTODOS.

3.1. Información del análisis.

Se trata de una análisis de datos de ultrasecuenciación. Las técnicas de ultrasecuenciación, también denominadas Next-generation sequencing (NGS) son técnicas bioquímicas baratas y rápidas cuya finalidad es determinar el orden de los nucleótidos en un oligonucleótido de ADN, con el fin de estudiar, en la mayoría de los casos, enfermedades ligadas a dichos genes.

En este trabajo se analizarán muestras obtenidas de un estudio que utiliza NGS pertenecientes a un análisis de tiroides. Nosotros nos centraremos en los datos de expresión (RNA-seq) pertenecientes a un análisis del tiroides en donde se compara tres tipos de infiltración medido en un total de 292 muestras pertenecientes a tres grupos:

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

Debido al gran número de muestras a tratar de cada grupo, se optó por tomar 10 muestras aleatorias de cada uno de ellos para realizar nuestro análisis.

3.2. Procedimiento general de análisis y explicación del pipeline.

(NOTA: En este apartado se explica el proceso de análisis. La línea de Código de R asociada a ella está insertada en el apéndice del artículo. En la página de github señalada en la portada del documento se puede encontrar un archivo html

denominado PEC2 que contiene el Código de R y las explicaciones en conjunto, para facilitar la comprensión en el caso que fuese necesario).

3.2.1. Definición de las muestras.

En primer lugar, se cargan los archivos targets y counts. El archivo counts muestra los nombres de ENSEMBL en formato acabado en .#, el que Bioconductor no puede identificar, por lo que se antes de cargarlo en R se eliminaron dichos factores desde el archivo de Excel.

Una vez hecho, extraemos los grupos y seleccionamos las muestras y las juntamos en un único data.frame con la función rbind.

```
eli<- subset(targets, Group == "ELI")
sfi<- subset(targets, Group == "SFI")
nit<- subset(targets, Group == "NIT")

targets_eli<- eli[sample(nrow(eli), 10, replace = FALSE),]
targets_sfi<- sfi[sample(nrow(sfi), 10, replace = FALSE),]
targets_nit<- nit[sample(nrow(nit), 10, replace = FALSE),]

mytargets <- rbind(targets_eli, targets_nit, targets_sfi)
```

De la misma forma se observa que existen diferencias entre las muestras en el dataset count y el dataset target, por lo que los equiparamos y seleccionamos los datos de las muestras elegidas teniendo en cuenta la variable Sample_name de targets.

```
mytargets$Sample_Name <- gsub("-",".", mytargets$Sample_Name)
mycounts=counts[,c(mytargets$Sample_Name)]
rownames(mycounts)=counts$X
```

3.2.2. Procesado de los datos: filtraje y normalización.

Una vez se han definido, ordenado y preparado las muestras, se puede comenzar el análisis de expresión de los genes. Para ello, en este trabajo se utilizará DESeq2 de Bioconductor.

Para comenzar se realizará la matriz de conteo que contiene los conteos brutos para cada muestra y realizaremos los pasos de control de calidad. Lo primero será formar la matriz con la función DESeqDataSetFromMatrix.

Una vez que ya tenemos la matriz que puede leer nuestra función, filtramos los datos obviando aquellos que sean igual a 1, ya que se probó anulando solo los valores menores de 0, pero no fue suficiente para tener buenos resultados.

Una vez tenemos los valores filtrados, se puede pasar a la normalización. DESeq2 utiliza un método de normalización de medianas, donde se ajustan los recuentos sin procesas para el tamaño de la biblioteca y es resistente a grandes cantidades de genes

expresados diferencialmente. La función estimateSizeFactors reasigna los datos al propio dds.

Los recuentros brutos para cada muestra se dividen por el factor de tamaño específico de muestra para la normalización. Para ver los factores de tamaño hemos usado la función sizeFactors.

Una vez que los recuentos normalizados han sido calculados y agregados a DESeq2, los recuentos se pueden extraer de él a partir de la función counts.

3.2.3. Identificación de genes diferencialmente expresados.

Una vez que tenemos los datos normalizados, se puede continuar con el análisis de expresión diferencial. Para usar método de visualización, hay que transformar los recuentos normalizados mediante una transformación estabilizadora de varianza, que podemos ver en la función VST de DESeq2. Se trata de una transformación logarítmica que modera la varianza a través de la media.

Podemos utilizar un mapheat para ver cuán parecidas o diferentes son las muestras de los diferentes grupos. El mapa de calor se crea utilizando los valores de correlación de expresión génica para todas las combinaciones por pares de muestras, siendo 1 la correlación perfecta. Dicha correlación se muestra por los colores.

Para hacer un mapa, hay que transformar los valores normalizados añadidos a dds a matriz. Una vez transformados, se utiliza el paquete pheatmap para crear el mapa de color.

También realizamos un gráfico PCA para representar los agrupamientos de otra forma y observar valores atípicos.

Una vez hemos visto la calidad de las muestras se puede comenzar el análisis de expresión diferencial.

El primer paso que seguir será ajustar los recuentos sin procesar al modelo DESeq2 estimando los factores de tamaño y la variación en la expresión a través de repeticiones para cada gen. Para realizar los cálculos necesitamos dos funciones, una que ya hemos utilizado anteriormente, ya que queremos los datos sin procesar, al no haber eliminado ningún valor tras el mapa de color y la PCA. Ahora se utilizará DESeq para generar los pasos completos al completar los espacios de DESeq2.

Nuestro objeto final deberá tener todos los factores necesarios para realizar la prueba de expresión diferencial entre grupos de muestra específicos.

Antes de continuar con el análisis debemos ver si nuestros datos se ajustan bien al modelo. Para los datos de RNA-seq, se espera que la varianza aumente con la expresión media del gen. Para observarlo, se puede calcular la media y la varianza para cada gen de las muestras normales usando la función apply. Posteriormente se trata un ggplot2 utilizando escalas en logaritmo en base 10.

Para terminar de verificar el ajuste de nuestros datos al modelo DESeq2, puede ser útil observar las estimaciones de dispersión mediante la función plotDispEsts.

3.2.4. Anotación de los resultados.

Ahora que hemos explorado el ajuste nuestros datos al modelo, podemos extraer los resultados de la prueba de expresión diferencial. Los resultados se pueden extraer utilizando la función result. Para ello elegimos un alpha de 0.05 de significancia.

Además, se realiza un diagrama MA para ver la muestra de los recuentos normalizados frente a los genes probados

3.2.5. Búsqueda de patrones de expresión y agrupación de las muestras.

Ahora se explorarán los resultados de expresión diferencial realizando las comparaciones. Para obtener descripciones de las columnas en la tabla de resultados, se usa la función mcols.

Los genes filtrados por DESeq2 aparacen en las tablas representados por NA. Podemos utilizar summary para ver los genes expresados para nuestro nivel de significancia 0.05 e información sobre el número de genes filtrados.

Para descubrir a qué genes pertenecen los resultados, añadimos el valor de ID de genes ENSEMBL.

3.2.6. Análisis de significación biológica.

Ahora se visualizarán los resultados para obtener una visión general del análisis mediante algunos métodos de visualización estándar.

Un método de visualización es mediante la gráfica del volcán, que muestra los cambios de pliegue en relación con los p valores ajustados para todos los genes.

Para finalizar, guardamos los resultados en tablas y formato csv.

3.2.7. Información de la sesión.

Se generarán los archivos que guarden las tablas con la información obtenida de las tres comparaciones.

4. RESULTADOS Y DISCUSIÓN.

4.1. Definición de las muestras.

Se obtienen el archivo mycount y mytargets con las características necesarias para realizar el análisis. Se han seleccionado las diez muestras de cada uno de los grupos de filtraje en ambos archivos, se han ordenado y colocado en orden los hombres de las columnas de mytargets y las filas de mycounts y se puede observar cómo se han eliminado perfectamente el .# final de los nombres de ENSEMBL en el archivo mytargets.

	Experiment	SRA_Sample	Sample_Name	Grupo_analisis	body_site	molecular_data_type	sex	Group	ShortName
GTEX-14BMU-0226-SM-5S2QA	SRX568916	SRS627158	GTEX.14BMU.0226.SM.5S2QA	3	Thyroid	Allele-Specific Expression	female	ELI	14BMU_ELI
GTEX-R55G-0726-SM-2TC6J	SRX204036	SRS374975	GTEX.R55G.0726.SM.2TC6J	3	Thyroid	RNA Seq (NGS)	female	ELI	R55G-_ELI
GTEX-13NZ9-1126-SM-5MR37	SRX582762	SRS631169	GTEX.13NZ9.1126.SM.5MR37	3	Thyroid	RNA Seq (NGS)	male	ELI	13NZ9_ELI
GTEX-PLZ4-1226-SM-2I5FE	SRX199272	SRS333099	GTEX.PLZ4.1226.SM.2I5FE	3	Thyroid	RNA Seq (NGS)	female	ELI	PLZ4-_ELI
GTEX-13QJC-0826-SM-5RQKC	SRX601511	SRS638114	GTEX.13QJC.0826.SM.5RQKC	3	Thyroid	Allele-Specific Expression	female	ELI	13QJC_ELI
GTEX-ZYY3-1926-SM-5GZXS	SRX568364	SRS627095	GTEX.ZYY3.1926.SM.5GZXS	3	Thyroid	Allele-Specific Expression	female	ELI	ZYY3-_ELI
GTEX-TMMY-0826-SM-33HB9	SRX222429	SRS389623	GTEX.TMMY.0826.SM.33HB9	3	Thyroid	Allele-Specific Expression	female	ELI	TMMY-_ELI
GTEX-YJ89-0726-SM-5P9F7	SRX583148	SRS631283	GTEX.YJ89.0726.SM.5P9F7	3	Thyroid	RNA Seq (NGS)	male	ELI	YJ89-_ELI
GTEX-14AS3-0226-SM-5Q5B6	SRX607358	SRS639491	GTEX.14AS3.0226.SM.5Q5B6	3	Thyroid	RNA Seq (NGS)	female	ELI	14AS3_ELI
GTEX-111VG-0526-SM-5N9BW	SRX563960	SRS625636	GTEX.111VG.0526.SM.5N9BW	3	Thyroid	RNA Seq (NGS)	male	ELI	111VG_ELI
GTEX-11DXZ-0926-SM-5N9CG	SRX602081	SRS638199	GTEX.11DXZ.0926.SM.5N9CG	1	Thyroid	Allele-Specific Expression	male	NIT	11DXZ_NIT
GTEX-11TTK-0826-SM-5N9EG	SRX569988	SRS627421	GTEX.11TTK.0826.SM.5N9EG	1	Thyroid	RNA Seq (NGS)	female	NIT	11TTK_NIT
GTEX-13FHO-0926-SM-5N9EW	SRX560993	SRS624999	GTEX.13FHO.0926.SM.5N9EW	1	Thyroid	RNA Seq (NGS)	male	NIT	13FHO_NIT
GTEX-11P81-0126-SM-5HL5Y	SRX599506	SRS637827	GTEX.11P81.0126.SM.5HL5Y	1	Thyroid	RNA Seq (NGS)	female	NIT	11P81_NIT
GTEX-139YR-1226-SM-5IFEU	SRX560623	SRS624912	GTEX.139YR.1226.SM.5IFEU	1	Thyroid	Allele-Specific Expression	male	NIT	139YR_NIT
GTEX-13FTW-0626-SM-5IFEX	SRX576441	SRS629388	GTEX.13FTW.0626.SM.5IFEX	1	Thyroid	RNA Seq (NGS)	male	NIT	13FTW_NIT
GTEX-11DXX-0226-SM-5P9HL	SRX568083	SRS627069	GTEX.11DXX.0226.SM.5P9HL	1	Thyroid	Allele-Specific Expression	female	NIT	11DXX_NIT
GTEX-ZAB5-0726-SM-5P9JG	SRX594942	SRS636243	GTEX.ZAB5.0726.SM.5P9JG	1	Thyroid	Allele-Specific Expression	male	NIT	ZAB5-_NIT

Figura 1. Visualización de datos mycounts

	GTEX.14BMU.0226.SM.5S2QA	GTEX.R55G.0726.SM.2TC6J	GTEX.13NZ9.1126.SM.5MR37	GTEX.PLZ4.1226.SM.2I5FE	GTEX.13QJC.0826.SM.5RQKC
ENSG00000223972	2	3	0	5	0
ENSG00000227232	423	134	1002	489	825
ENSG00000243485	0	1	1	1	1
ENSG00000237613	0	2	0	3	0
ENSG00000268020	2	1	0	2	0
ENSG00000240361	1	0	1	1	1
ENSG00000186092	0	1	0	3	1
ENSG00000238009	18	3	15	7	10
ENSG00000233750	6	143	19	100	21
ENSG00000237683	325	366	602	523	853
ENSG00000268903	1	1	2	2	6
ENSG00000239906	4	5	32	3	16
ENSG00000241860	41	51	96	183	94
ENSG00000222623	0	0	0	0	0
ENSG00000241599	1	1	0	1	0

Figura 2. Visualización de datos mytargets

4.2. Procesado de los datos: filtraje y normalización.

Al realizar la matriz de DESeq2 y eliminar los datos con valores menores de uno se obtiene que se filtran alrededor de 13000 datos. Las muestras de estudio contenían 56202 muestras y nos quedamos con 43369 para proseguir con muestro análisis.

```
dds <- dds_wt [rowSums(counts((dds_wt))) > 1]
dim(dds_wt)

## [1] 56202    30

dim(dds)

## [1] 43369    30
```

Tras realizar la normalización se obtienen los siguientes valores:

```
dds <- estimateSizeFactors(dds)
sizeFactors(dds)

##  GTEX-YFC4-2626-SM-5P9FQ GTEX-14BMU-0226-SM-5S2QA GTEX-111VG-0526-SM-
5N9BW
##               1.6167599               0.7974216
0.8928268
## GTEX-13NZ9-1126-SM-5MR37 GTEX-11XUK-0226-SM-5EQLW GTEX-14ABY-0926-SM-
5Q5DY
##               1.2685293               0.8868170
1.1740473
## GTEX-13QJC-0826-SM-5RQKC GTEX-11NV4-0626-SM-5N9BR  GTEX-YJ89-0726-SM-
5P9F7
##               0.8868437               1.0254077
1.4753475
##  GTEX-PLZ4-1226-SM-2I5FE GTEX-131XE-0126-SM-5LZVC  GTEX-OIZG-0226-SM-
2TC5L
##               1.1922881               0.8567108
0.7348213
##  GTEX-OIZI-0726-SM-2XCEI  GTEX-WYBS-1926-SM-3NM8N  GTEX-XV7Q-0326-SM-
4BRVM
##               1.4392952               1.2902647
0.9880657
## GTEX-11DXZ-0926-SM-5N9CG GTEX-1192W-0126-SM-5EGGS  GTEX-XLM4-0726-SM-
4AT64
##               0.8347847               0.8544493
0.8889334
##  GTEX-T6MN-0626-SM-32PM9  GTEX-XBEW-0126-SM-4AT66 GTEX-11O72-2326-SM-
5BC7H
##               1.3042471               0.7108404
1.0519420
##  GTEX-R55E-0826-SM-2TC5M GTEX-131XF-1826-SM-5EGKG GTEX-12584-0826-SM-
5FQSK
##               0.8001468               0.7551029
```

```
0.9561208
##  GTEX-WYVS-0326-SM-3NM9V  GTEX-P78B-0526-SM-2I5F7  GTEX-RM2N-0526-SM-
2TF4N
##                1.4477608                1.1134809
0.8725489
## GTEX-12ZZX-1226-SM-5EGHS  GTEX-Q2AH-0726-SM-2I3EA GTEX-13FH7-0126-SM-
5KLZ1
##                0.7377582                0.9496596
1.1413465
```

Los resultados pueden observarse en un gráfico de puntos:

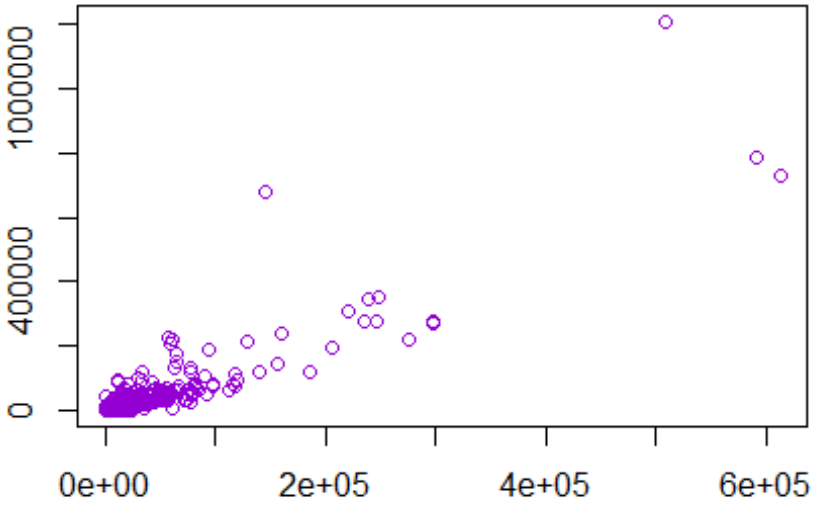

Figura 3. Gráfico de puntos datos normalizados

4.3. Identificación de genes diferencialmente expresados.

Para usar un método de observación de los datos normalizados se transforman los datos normalizados mediante estabilización de varianza con la función VST de DESeq2. El mapa de calor se crea utilizando los valores de correlación de expresión génica para todas las combinaciones por pares de muestras, siendo 1 la correlación perfecta. Dicha correlación se muestra por los colores. Por lo que calculamos la correlación. Usamos la función pheatmap y el resultado es el siguiente:

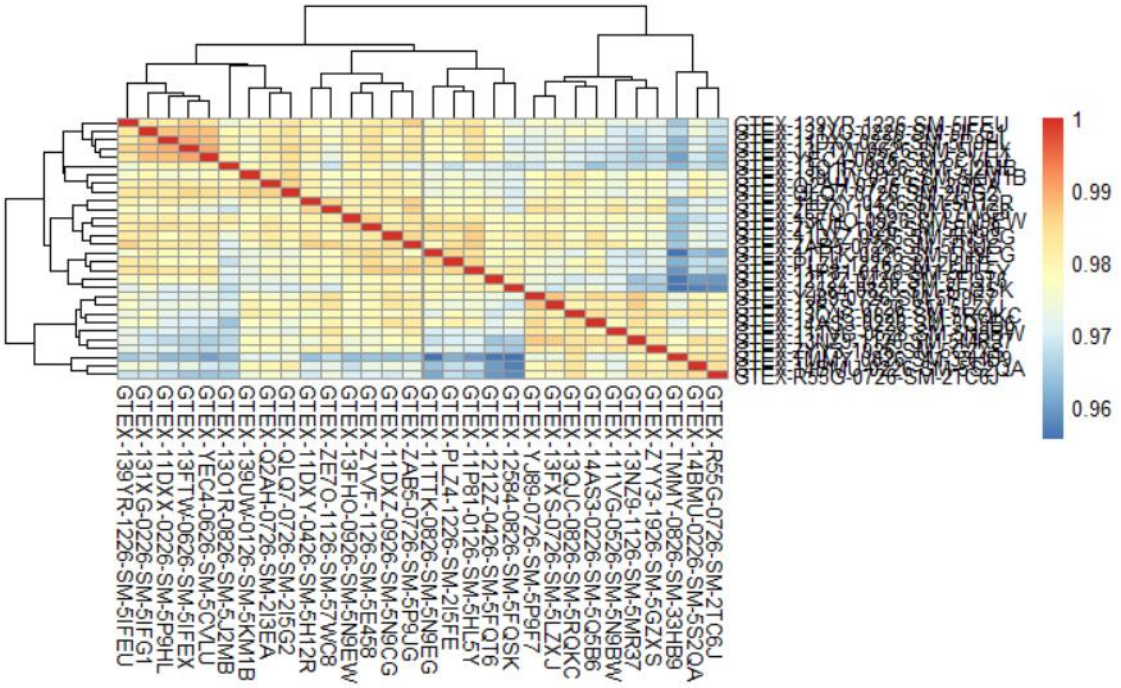

Figura 4. Mapa de color datos correlación

El argumento de anotación nos selecciona los factores de targets que se incluirán como barras de anotación, es decir, nos agrupará las muestras según los tres grupos iniciales. La salida del mapa de color muestra que las réplicas biológicas se agrupan y las condiciones se separan. Esto es alentador, ya que los genes expresados diferencialmente entre las condiciones es probable que impulsen dicha separación.

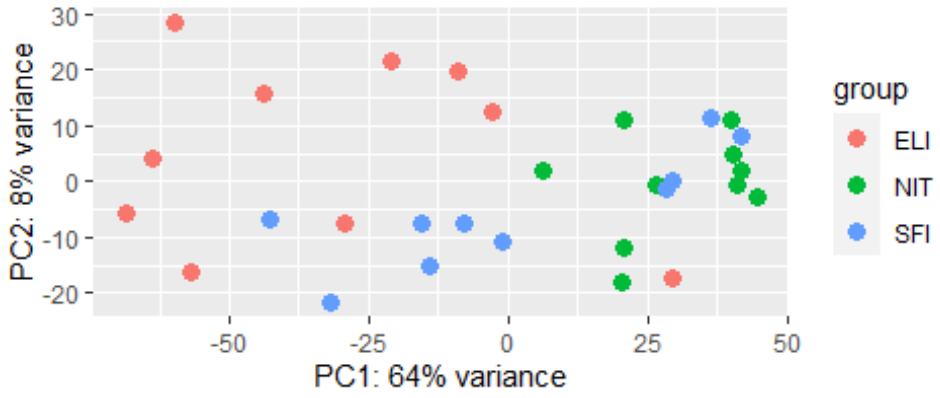

Figura 5. Gráfica PCA de las muestras de los tres grupos

El gráfico PCA obtenido con dichos valores muestra que la mayoría de los puntos pertenecen a PC1, se visualiza algún valor atípico, que probablemente se relacione con los puntos de baja correlación en el mapa de colores.

Ahora utilizamos DESeq para generar los pasos completos al completar los espacios de DESeq2:

```
dds <- DESeq(dds)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 211 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions

## fitting model and testing
```

Se realiza un gráfico con ggplot2 para ver si nuestros datos se ajustan al modelo DESeq2, viendo si la varianza aumenta con la expresión media del gen:

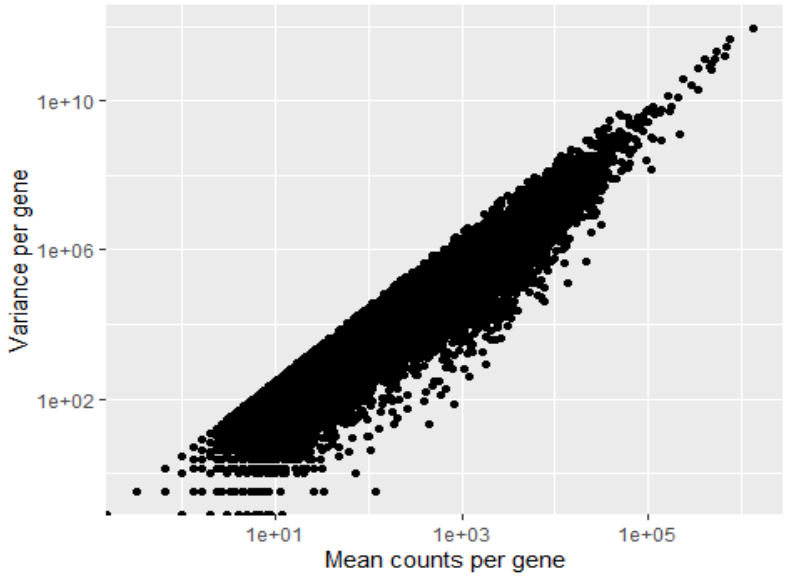

Figura 6. Gráfico varianza y media

Cada punto negro representa un gen. En el gráfico vemos que la varianza en la expresión génica aumenta con la media, es decir, se ajusta al modelo.

Para terminar de verificar el ajuste de nuestros datos al modelo DESeq2, puede ser útil observar las estimaciones de dispersión mediante la función plotDispEsts.

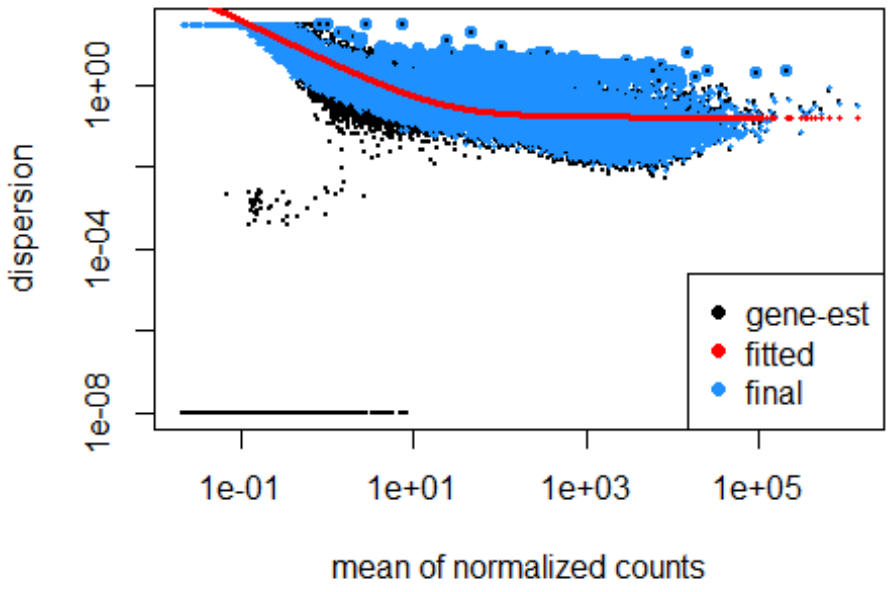

Figura 7. Gráfico dispersión

El valor de dispersión media calculado teniendo en cuenta todos los genes se refleja en la línea roja. Los puntos negros son las estimaciones iniciales, que normalmente no se ajustan del todo bien y se reducen hacia la curva para obtener estimaciones precisas que se muestran como puntos azules y se utilizan para determinar los genes expresados diferencialmente. Podemos ver que nuestros datos se ajustan muy bien a la curva y, por tanto, al modelo.

4.4. Anotación de los resultados.

Se extraen los datos de la prueba de expresión diferencial con un Alpha 0.05 y se generan diagramas MA para ver la muestra de los recuentos normalizados frente a los genes probados:

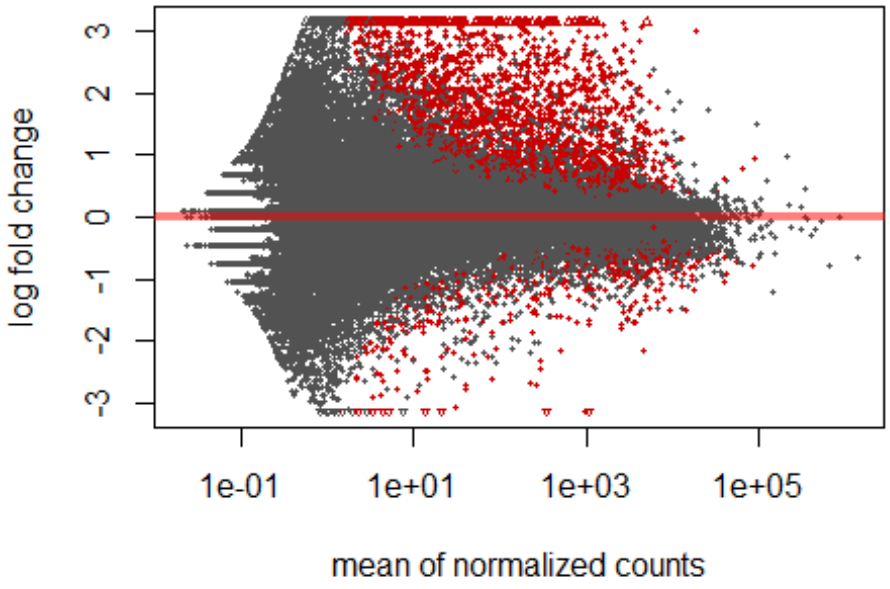

Figura 8. Diagrama dispersión comparación ELI-SFI

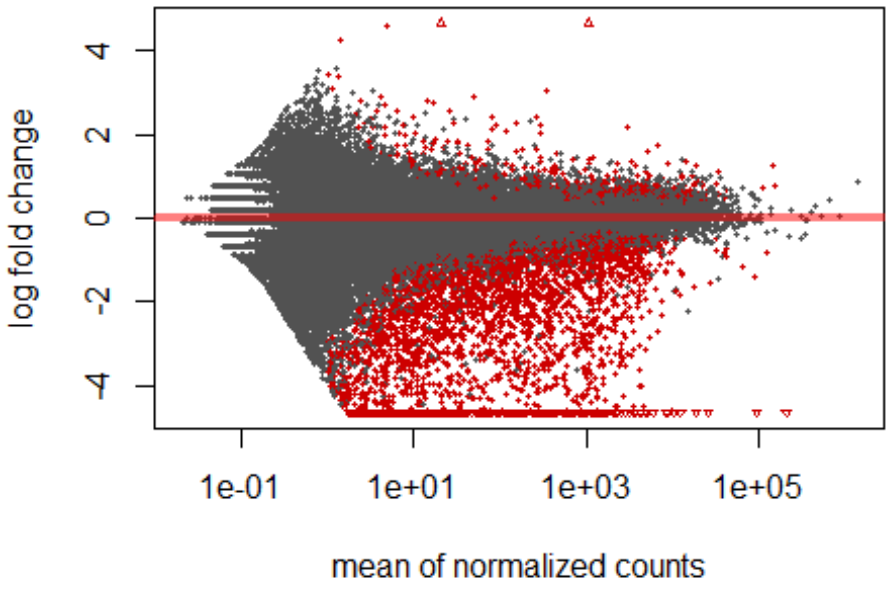

Figura 9. Diagrama dispersión comparación NIT-ELI

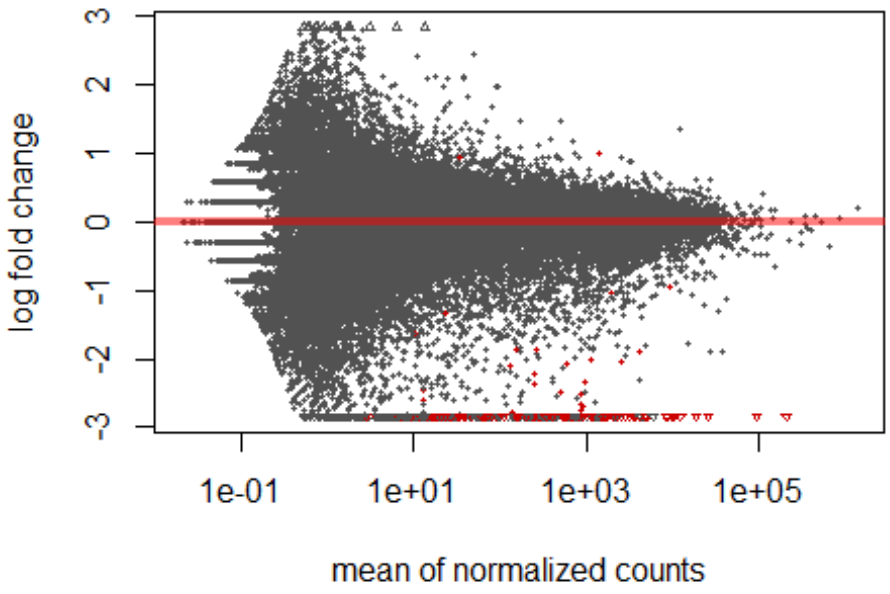

Figura 10. Diagrama de dispersión comparación NIT-SFI

Se puede observar que en los resultados de los grupos NIT y SFI hay menos puntos rojos, es decir, menos genes diferencialmente expresados.

4.5. Búsqueda de patrones de expresión y agrupación de las muestras.

Se obtienen las descripciones de las columnas en la tabla de resultados mediante la función mcols.

```
mcols(result_eli_sfi)
```

```
## DataFrame with 6 rows and 2 columns
##                         type                                 description
##                  <character>                                 <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MLE): Group ELI vs SFI
## lfcSE               results          standard error: Group ELI vs SFI
## stat                results          Wald statistic: Group ELI vs SFI
## pvalue              results       Wald test p-value: Group ELI vs SFI
## padj                results                        BH adjusted p-values
```

```
mcols(result_nit_eli)
```

```
## DataFrame with 6 rows and 2 columns
##                         type                                 description
##                  <character>                                 <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MLE): Group NIT vs ELI
## lfcSE               results          standard error: Group NIT vs ELI
## stat                results          Wald statistic: Group NIT vs ELI
## pvalue              results       Wald test p-value: Group NIT vs ELI
## padj                results                       BH adjusted p-values

mcols(result_nit_sfi)

## DataFrame with 6 rows and 2 columns
##                         type                                 description
##                  <character>                                 <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MLE): Group NIT vs SFI
## lfcSE               results          standard error: Group NIT vs SFI
## stat                results          Wald statistic: Group NIT vs SFI
## pvalue              results       Wald test p-value: Group NIT vs SFI
## padj                results                       BH adjusted p-values
```

En la primera columna vemos el valor medio de todas las muestras, seguido del log 2, cambios de plegado, error estándar de las estimaciones de cambio de plegado, la prueba de Wald, el pvalor para dicha prueba y el p ajustado de Benjamini-Hochberg. Los genes filtrados por DESeq2 aparacen en las tablas representados por NA.

Obtenemos los resultados de los genes expresados para el nivel de significancia 0.05:

```
summary(result_eli_sfi)

##
## out of 46140 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 2118, 4.6%
## LFC < 0 (down)     : 554, 1.2%
## outliers [1]       : 0, 0%
## low counts [2]     : 16803, 36%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

summary(result_nit_eli)

##
## out of 46140 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 573, 1.2%
## LFC < 0 (down)     : 3015, 6.5%
## outliers [1]       : 0, 0%
## low counts [2]     : 15035, 33%
```

```
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

summary(result_nit_sfi)

##
## out of 46140 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 5, 0.011%
## LFC < 0 (down)     : 152, 0.33%
## outliers [1]       : 0, 0%
## low counts [2]     : 16803, 36%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

En estos resultados, se puede ver la suma de los genes DE para cada grupo de comparación, aquellos que tienen un log2 fold que cambia menos de 0 y aquellos con fold que cambia más que 0.

A dichos valores se les añadió los valores ENSEMBL mediante mapIDs, como puede observarse en el siguiente ejemplo.

```
## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 6 rows and 8 columns
##                          baseMean     log2FoldChange             lfcSE
##                         <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292  -1.18869114663546 0.676752393645434
## ENSG00000227232  703.913194096269 0.0571869895502778 0.209263197949233
## ENSG00000243485  1.65379907479822  -2.16485573428325  0.74564989710915
## ENSG00000237613  1.05107039541176  -2.36700511544809  1.04940446228125
```

4.6. Análisis de significación biológica.

Se realiza un gráfico de volcán de cada uno de los resultados de cada comparación, que muestra los cambios de pliegue en relación con los p valores ajustados para todos los genes.

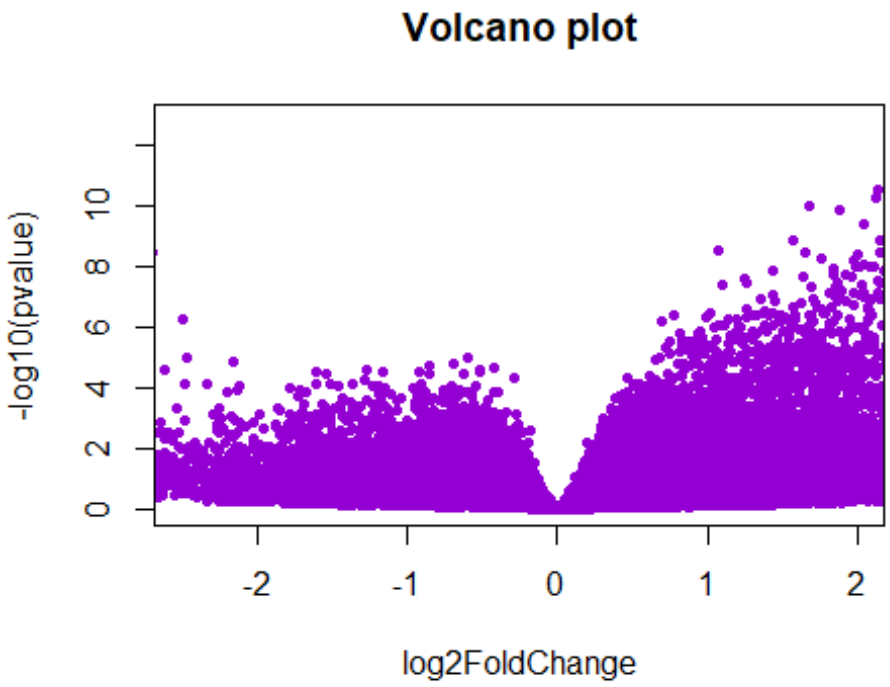

Figura 11. Gráfico de volcán comparación ELI-SFI

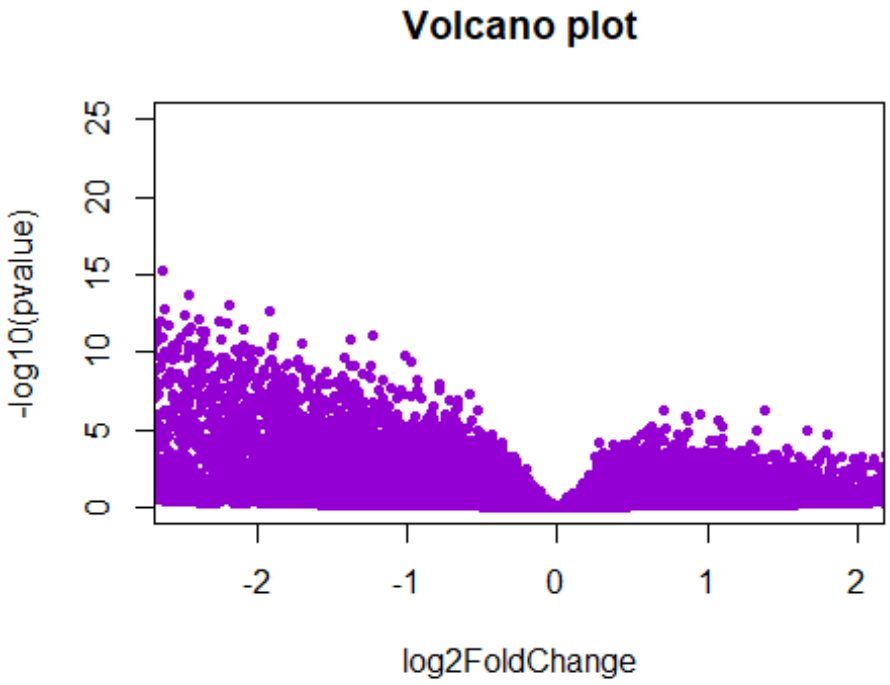

Figura 12. Gráfico de volcán comparación NIT-ELI

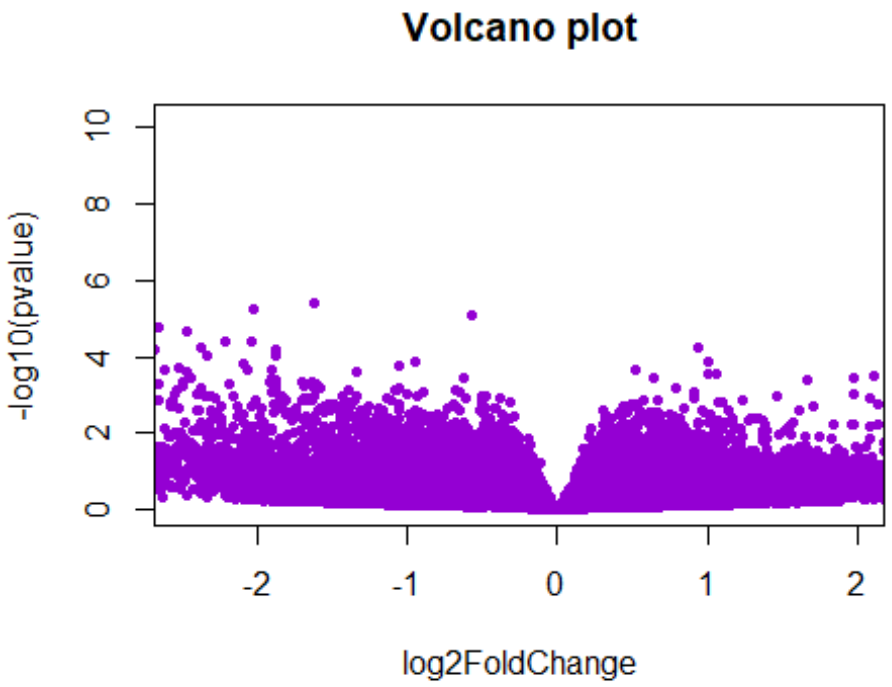

Figura 13. Gráfico volcán comparación NIT-SFI

4.7. Información de la sesión.

La información de la sesión es la siguiente:

```
sessionInfo()
```

```
## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.1252  LC_CTYPE=Spanish_Spain.1252
## [3] LC_MONETARY=Spanish_Spain.1252 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.1252
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils
datasets
## [8] methods   base
##
## other attached packages:
##  [1] org.Hs.eg.db_3.10.0         AnnotationDbi_1.48.0
##  [3] ggplot2_3.3.1               pheatmap_1.0.12
```

```
##  [5] DESeq2_1.26.0               SummarizedExperiment_1.16.1
##  [7] DelayedArray_0.12.3         BiocParallel_1.20.1
##  [9] matrixStats_0.56.0          Biobase_2.46.0
## [11] GenomicRanges_1.38.0        GenomeInfoDb_1.22.1
## [13] IRanges_2.20.2              S4Vectors_0.24.4
## [15] BiocGenerics_0.32.0         dplyr_1.0.0
## [17] readxl_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] bit64_0.9-7            splines_3.6.3          Formula_1.2-3
##  [4] latticeExtra_0.6-29    blob_1.2.1
GenomeInfoDbData_1.2.2
##  [7] cellranger_1.1.0       yaml_2.2.1             RSQLite_2.2.0
## [10] pillar_1.4.4           backports_1.1.7        lattice_0.20-41
## [13] glue_1.4.1             digest_0.6.25          RColorBrewer_1.1-2
## [16] XVector_0.26.0         checkmate_2.0.0        colorspace_1.4-1
## [19] htmltools_0.4.0        Matrix_1.2-18          XML_3.99-0.3
## [22] pkgconfig_2.0.3        genefilter_1.68.0      zlibbioc_1.32.0
## [25] purrr_0.3.4            xtable_1.8-4           scales_1.1.1
## [28] jpeg_0.1-8.1           tibble_3.0.1           htmlTable_1.13.3
## [31] annotate_1.64.0        farver_2.0.3           generics_0.0.2
## [34] ellipsis_0.3.0         withr_2.2.0            nnet_7.3-12
## [37] survival_3.1-12        magrittr_1.5           crayon_1.3.4
## [40] memoise_1.1.0          evaluate_0.14          foreign_0.8-76
## [43] tools_3.6.3            data.table_1.12.8      lifecycle_0.2.0
## [46] stringr_1.4.0          locfit_1.5-9.4         munsell_0.5.0
## [49] cluster_2.1.0          compiler_3.6.3         rlang_0.4.6
## [52] grid_3.6.3             RCurl_1.98-1.2         rstudioapi_0.11
## [55] htmlwidgets_1.5.1      labeling_0.3           bitops_1.0-6
## [58] base64enc_0.1-3        rmarkdown_2.2          gtable_0.3.0
## [61] DBI_1.1.0              R6_2.4.1               gridExtra_2.3
## [64] knitr_1.28             bit_1.1-15.2           Hmisc_4.4-0
## [67] stringi_1.4.6          Rcpp_1.0.4.6           geneplotter_1.64.0
## [70] vctrs_0.3.1            rpart_4.1-15           acepack_1.4.1
## [73] png_0.1-7              tidyselect_1.1.0       xfun_0.14
```

5. DISCUSIÓN DE PROBLEMAS.

Mis problemas a la hora de realizar este informe han sido los siguientes:

- Empezar a realizar el análisis: En un principio me sentí bastante perdida, pero posteriormente encontré un curso online gratuito que se expone en la bibliografía, el cual he seguido para realizar la PEC y me ha sido muy útil.

- Interpretar los resultados: La realización del informe no me parece compleja una vez tienes un buen tutorial para hacerlo por primera vez, pero analizar qué significa cada número o cada dato me parece bastante complejo.

6. CONCLUSIÓN.

Me resulta complejo obtener una conclusión del trabajo. Los análisis de datos de NGS tienen un número elevado de muestras, por lo que no siempre es sencillo trabajar con ellas, además de que suelen necesitar transformar o procesar los datos, lo que puede llevar a errores.

Parece observarse que la comparación NIT-SFI tienen menos genes expresados diferencialmente, mientras que ELI contiene una alta cantidad de genes expresados diferencialmente. Sería interesante ver a qué tipo de vías celulares afectan estos tres grupos, sobre todo ELI, pero no se ha podido realizar el mapa de rutas. Por tanto, se puede decir que se ven algunas conclusiones, pero que sería interesante hacer más pruebas que relacionasen los genes con las rutas y los tejidos a los que más afectan los genes.

7. BIBLIOGRAFÍA.

1. Apuntes de la asignatura Análisis de datos ómicos (Universitat Oberta de Catalunya).
2. Curso "Introduction to RNA-Seq" de DataCamp. (Disponible en https://campus.datacamp.com/courses/rna-seq-with-bioconductor-in-r/introduction-to-rna-seq-theory-and-workflow?ex=1)
3. González, Ignacio. Statistical analysis of RNA-Seq data – Tutorial. 2014. (Disponible en http://www.nathalievialaneix.eu/doc/pdf/tutorial-rnaseq.pdf)
4. Michael I. Love, Simon Anders, and Wolfgang Huber. Analyzing RNA-seq data with DESeq2. (Disponible en: https://bioconductor.org/packages/release/bioc/vignettes/DESeq2/inst/doc/DESeq2.html#theory)
5. MA-Plot From Base Means And Log Fold Changes. (Disponible en https://www.rdocumentation.org/packages/DESeq2/versions/1.12.3/topics/plotMA)
6. Paquete org.Hs.eg.db Bioconductor. (Disponible en: https://bioconductor.org/packages/release/data/annotation/html/org.Hs.eg.db.html)

8. APÉNDICE (Código generado en R).

8.1. Definición de las muestras.

```
library(readxl)
```

```
targets <- read.csv("/Users/maria/OneDrive/UOC/Análisis de datos
ómicos/PEC 2/data/targets.csv", header= TRUE)
counts <- read.csv("/Users/maria/OneDrive/UOC/Análisis de datos
ómicos/PEC 2/data//counts.csv", header= TRUE, sep = ";")
View(targets)

eli<- subset(targets, Group == "ELI")
sfi<- subset(targets, Group == "SFI")
nit<- subset(targets, Group == "NIT")

targets_eli<- eli[sample(nrow(eli), 10, replace = FALSE),]
targets_sfi<- sfi[sample(nrow(sfi), 10, replace = FALSE),]
targets_nit<- nit[sample(nrow(nit), 10, replace = FALSE),]

mytargets <- rbind(targets_eli, targets_nit, targets_sfi)
row.names(mytargets) <- mytargets$Sample_Name
head(mytargets, 5)

##                           Experiment SRA_Sample
Sample_Name
## GTEX-YFC4-2626-SM-5P9FQ   SRX615373  SRS644099  GTEX-YFC4-2626-SM-
5P9FQ
## GTEX-14BMU-0226-SM-5S2QA  SRX568916  SRS627158 GTEX-14BMU-0226-SM-
5S2QA
## GTEX-111VG-0526-SM-5N9BW  SRX563960  SRS625636 GTEX-111VG-0526-SM-
5N9BW
## GTEX-13NZ9-1126-SM-5MR37  SRX582762  SRS631169 GTEX-13NZ9-1126-SM-
5MR37
## GTEX-11XUK-0226-SM-5EQLW  SRX619829  SRS644736 GTEX-11XUK-0226-SM-
5EQLW
##                           Grupo_analisis body_site
molecular_data_type
## GTEX-YFC4-2626-SM-5P9FQ              3   Thyroid Allele-Specific
Expression
## GTEX-14BMU-0226-SM-5S2QA             3   Thyroid Allele-Specific
Expression
## GTEX-111VG-0526-SM-5N9BW             3   Thyroid              RNA Seq
(NGS)
## GTEX-13NZ9-1126-SM-5MR37             3   Thyroid              RNA Seq
(NGS)
## GTEX-11XUK-0226-SM-5EQLW             3   Thyroid              RNA Seq
(NGS)
##                              sex Group ShortName
## GTEX-YFC4-2626-SM-5P9FQ  female   ELI YFC4-_ELI
## GTEX-14BMU-0226-SM-5S2QA female   ELI 14BMU_ELI
## GTEX-111VG-0526-SM-5N9BW   male   ELI 111VG_ELI
## GTEX-13NZ9-1126-SM-5MR37   male   ELI 13NZ9_ELI
## GTEX-11XUK-0226-SM-5EQLW female   ELI 11XUK_ELI

library(stats)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

mytargets$Sample_Name <- gsub("-",".", mytargets$Sample_Name)
mycounts=counts[,c(mytargets$Sample_Name)]
rownames(mycounts)=counts$X

View(mycounts)
head(mycounts)

##                 GTEX.YFC4.2626.SM.5P9FQ GTEX.14BMU.0226.SM.5S2QA
## ENSG00000223972                       1                       2
## ENSG00000227232                    1472                     423
## ENSG00000243485                       1                       0
## ENSG00000237613                       0                       0
## ENSG00000268020                       0                       2
## ENSG00000240361                       1                       1
##                 GTEX.111VG.0526.SM.5N9BW GTEX.13NZ9.1126.SM.5MR37
## ENSG00000223972                        1                        0
## ENSG00000227232                      474                     1002
## ENSG00000243485                        1                        1
## ENSG00000237613                        0                        0
## ENSG00000268020                        1                        0
## ENSG00000240361                        1                        1
##                 GTEX.11XUK.0226.SM.5EQLW GTEX.14ABY.0926.SM.5Q5DY
## ENSG00000223972                        0                        1
## ENSG00000227232                      419                      775
## ENSG00000243485                        0                        2
## ENSG00000237613                        1                        0
## ENSG00000268020                        0                        0
## ENSG00000240361                        0                        0
##                 GTEX.13QJC.0826.SM.5RQKC GTEX.11NV4.0626.SM.5N9BR
## ENSG00000223972                        0                        3
## ENSG00000227232                      825                     1301
## ENSG00000243485                        1                        1
## ENSG00000237613                        0                        0
## ENSG00000268020                        0                        0
## ENSG00000240361                        1                        1
##                 GTEX.YJ89.0726.SM.5P9F7 GTEX.PLZ4.1226.SM.2I5FE
## ENSG00000223972                       4                       5
## ENSG00000227232                    1325                     489
## ENSG00000243485                       1                       1
## ENSG00000237613                       0                       3
```

```
## ENSG00000268020                       2                         2
## ENSG00000240361                       1                         1
##                 GTEX.131XE.0126.SM.5LZVC GTEX.OIZG.0226.SM.2TC5L
## ENSG00000223972                       1                         6
## ENSG00000227232                     623                       313
## ENSG00000243485                       0                         1
## ENSG00000237613                       0                         1
## ENSG00000268020                       0                         1
## ENSG00000240361                       0                         3
##                 GTEX.OIZI.0726.SM.2XCEI GTEX.WYBS.1926.SM.3NM8N
## ENSG00000223972                       0                         7
## ENSG00000227232                     523                       730
## ENSG00000243485                       2                         0
## ENSG00000237613                       3                         1
## ENSG00000268020                       0                         1
## ENSG00000240361                       3                         0
##                 GTEX.XV7Q.0326.SM.4BRVM GTEX.11DXZ.0926.SM.5N9CG
## ENSG00000223972                       2                         1
## ENSG00000227232                     837                       768
## ENSG00000243485                       0                         0
## ENSG00000237613                       0                         1
## ENSG00000268020                       1                         0
## ENSG00000240361                       0                         0
##                 GTEX.1192W.0126.SM.5EGGS GTEX.XLM4.0726.SM.4AT64
## ENSG00000223972                       3                         1
## ENSG00000227232                     573                       647
## ENSG00000243485                       1                         0
## ENSG00000237613                       0                         0
## ENSG00000268020                       0                         0
## ENSG00000240361                       3                         0
##                 GTEX.T6MN.0626.SM.32PM9 GTEX.XBEW.0126.SM.4AT66
## ENSG00000223972                       1                         1
## ENSG00000227232                     892                       473
## ENSG00000243485                       1                         1
## ENSG00000237613                       0                         2
## ENSG00000268020                       1                         1
## ENSG00000240361                       0                         1
##                 GTEX.11O72.2326.SM.5BC7H GTEX.R55E.0826.SM.2TC5M
## ENSG00000223972                       0                         3
## ENSG00000227232                     633                       533
## ENSG00000243485                       2                         1
## ENSG00000237613                       1                         0
## ENSG00000268020                       0                         0
## ENSG00000240361                       1                         0
##                 GTEX.131XF.1826.SM.5EGKG GTEX.12584.0826.SM.5FQSK
## ENSG00000223972                       5                         1
## ENSG00000227232                     656                      1064
## ENSG00000243485                       1                         2
## ENSG00000237613                       1                         0
## ENSG00000268020                       0                         2
```

```
## ENSG00000240361                       1                       2
##                 GTEX.WYVS.0326.SM.3NM9V GTEX.P78B.0526.SM.2I5F7
## ENSG00000223972                       6                       8
## ENSG00000227232                     820                     548
## ENSG00000243485                       0                      11
## ENSG00000237613                       1                       6
## ENSG00000268020                       0                       1
## ENSG00000240361                       4                       4
##                 GTEX.RM2N.0526.SM.2TF4N GTEX.12ZZX.1226.SM.5EGHS
## ENSG00000223972                       3                       2
## ENSG00000227232                     406                     679
## ENSG00000243485                       4                       2
## ENSG00000237613                       1                       4
## ENSG00000268020                       0                       0
## ENSG00000240361                       1                       1
##                 GTEX.Q2AH.0726.SM.2I3EA GTEX.13FH7.0126.SM.5KLZ1
## ENSG00000223972                       1                       5
## ENSG00000227232                     874                     576
## ENSG00000243485                       8                       4
## ENSG00000237613                       2                       3
## ENSG00000268020                       0                       0
## ENSG00000240361                       1                       1
```

8.2. Procesado de los datos: filtraje y normalización.

```
library(BiocGenerics)

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get,
```

```
grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which, which.max, which.min

library(DESeq2)

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:base':
##
##     expand.grid

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomicRanges

## Loading required package: GenomeInfoDb

## Loading required package: SummarizedExperiment

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

## Loading required package: DelayedArray

## Loading required package: matrixStats
```

```
##
## Attaching package: 'matrixStats'

## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians

## The following object is masked from 'package:dplyr':
##
##     count

## Loading required package: BiocParallel

##
## Attaching package: 'DelayedArray'

## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges

## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum

dds_wt <- DESeqDataSetFromMatrix(countData = mycounts, colData =
mytargets,design = ~ Group)

dds <- dds_wt [rowSums(counts((dds_wt))) > 1]
dim(dds_wt)

## [1] 56202    30

dim(dds)

## [1] 43369    30

dds <- estimateSizeFactors(dds)
sizeFactors(dds)

##  GTEX-YFC4-2626-SM-5P9FQ GTEX-14BMU-0226-SM-5S2QA GTEX-111VG-0526-SM-
5N9BW
##               1.6167599               0.7974216
0.8928268
## GTEX-13NZ9-1126-SM-5MR37 GTEX-11XUK-0226-SM-5EQLW GTEX-14ABY-0926-SM-
5Q5DY
##               1.2685293               0.8868170
1.1740473
## GTEX-13QJC-0826-SM-5RQKC GTEX-11NV4-0626-SM-5N9BR  GTEX-YJ89-0726-SM-
5P9F7
##               0.8868437               1.0254077
1.4753475
##  GTEX-PLZ4-1226-SM-2I5FE GTEX-131XE-0126-SM-5LZVC  GTEX-OIZG-0226-SM-
2TC5L
```

```
##                1.1922881                0.8567108
0.7348213
##  GTEX-OIZI-0726-SM-2XCEI  GTEX-WYBS-1926-SM-3NM8N  GTEX-XV7Q-0326-SM-
4BRVM
##                1.4392952                1.2902647
0.9880657
## GTEX-11DXZ-0926-SM-5N9CG GTEX-1192W-0126-SM-5EGGS  GTEX-XLM4-0726-SM-
4AT64
##                0.8347847                0.8544493
0.8889334
##  GTEX-T6MN-0626-SM-32PM9  GTEX-XBEW-0126-SM-4AT66 GTEX-11O72-2326-SM-
5BC7H
##                1.3042471                0.7108404
1.0519420
##  GTEX-R55E-0826-SM-2TC5M GTEX-131XF-1826-SM-5EGKG GTEX-12584-0826-SM-
5FQSK
##                0.8001468                0.7551029
0.9561208
##  GTEX-WYVS-0326-SM-3NM9V  GTEX-P78B-0526-SM-2I5F7  GTEX-RM2N-0526-SM-
2TF4N
##                1.4477608                1.1134809
0.8725489
## GTEX-12ZZX-1226-SM-5EGHS  GTEX-Q2AH-0726-SM-2I3EA GTEX-13FH7-0126-SM-
5KLZ1
##                0.7377582                0.9496596
1.1413465
```

```
normalized_counts <- counts(dds, normalized=TRUE)
```

```
View (normalized_counts)
plot(normalized_counts, xlab="", ylab="", col="dark violet")
```

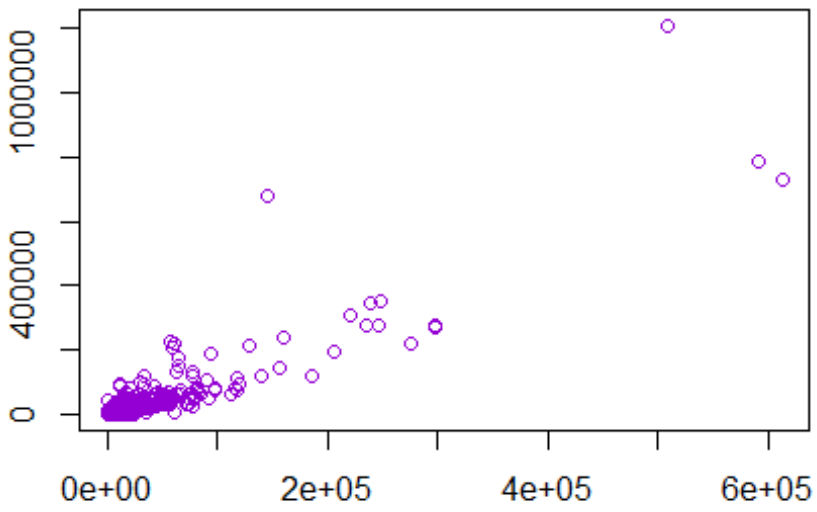

Figura 14. Gráfico de puntos datos normalizados

8.3. Identificación de genes diferencialmente expresados.

```
vsd <- vst(dds, blind=TRUE)

vsd_mat<- assay(vsd)

vsd_cor <- cor(vsd_mat)
View(vsd_cor)

library(pheatmap)
pheatmap(vsd_cor)
```

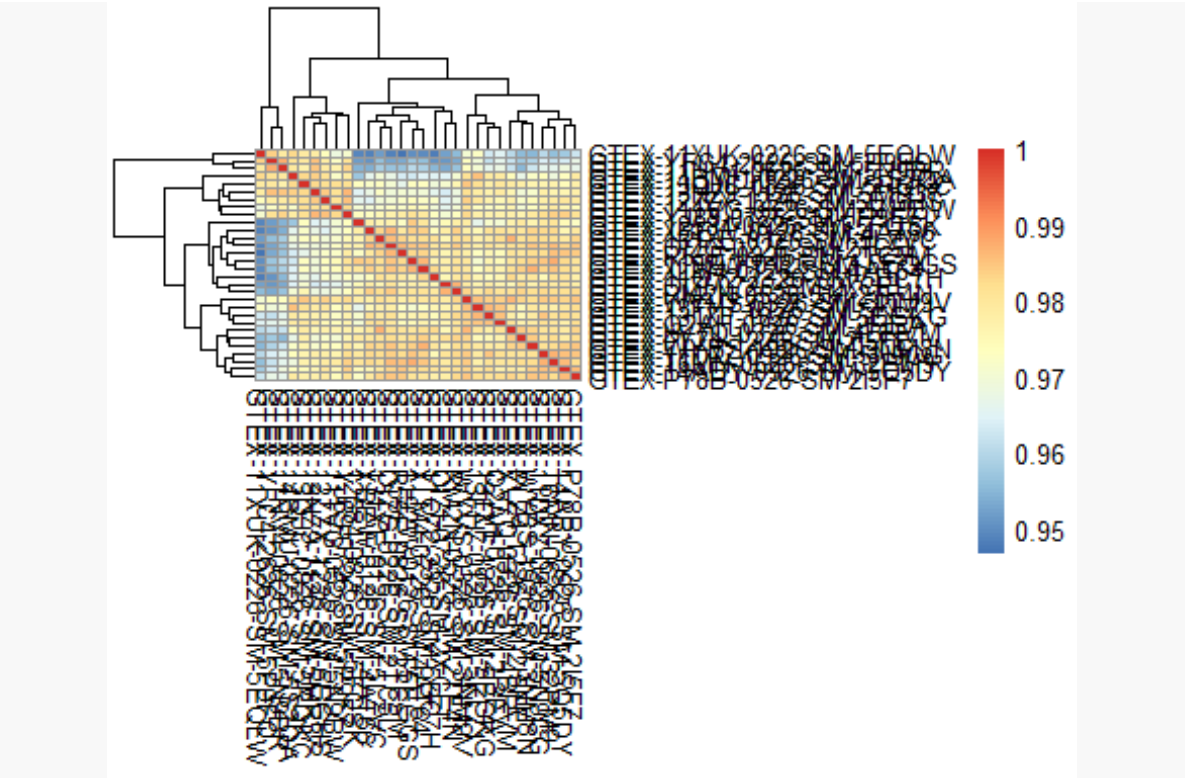

Figura 15. Mapa de colores correlación

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

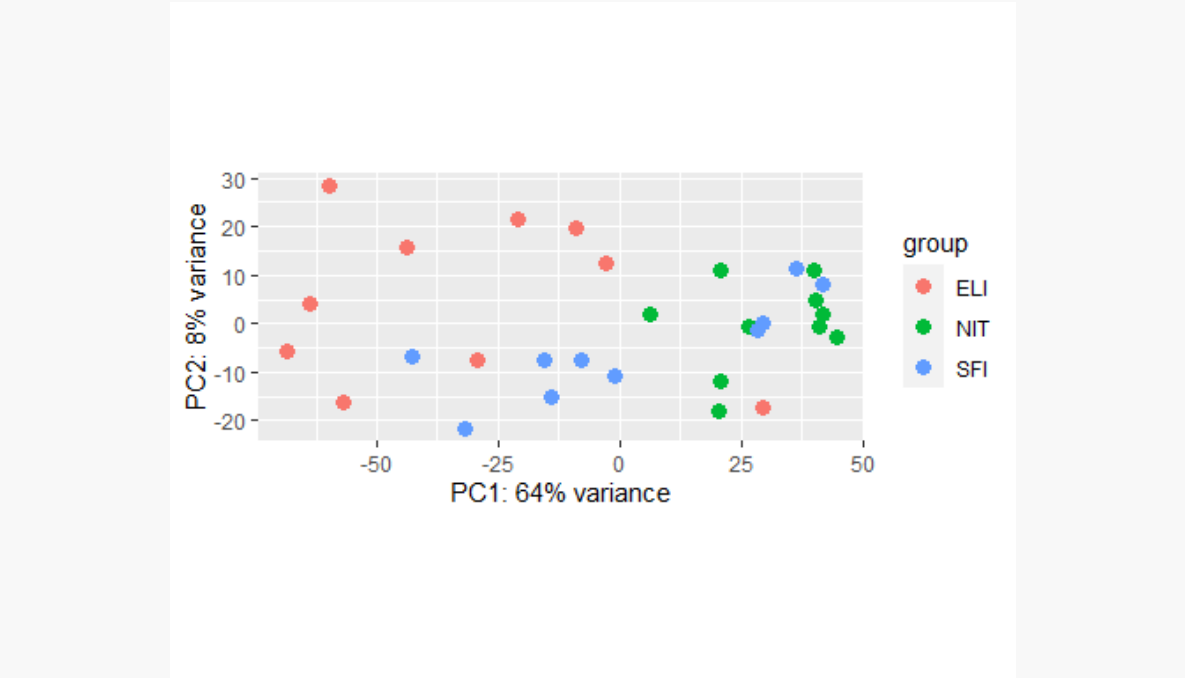

Figura 16. Gráfico PCA para los tres grupos

```
dds <- DESeqDataSetFromMatrix(countData = mycounts, colData = mytargets,
design = ~ Group)

dds <- DESeq(dds)

## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

## -- replacing outliers and refitting for 211 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)

## estimating dispersions
```

fitting model and testing

```
mean_counts <- apply(mycounts[, 1:3], 1, mean)
variance_counts <- apply(mycounts[,1:3], 1, var)

df <- data.frame(mean_counts, variance_counts)

library(ggplot2)
ggplot(df) +
        geom_point(aes(x=mean_counts, y=variance_counts)) +
        scale_y_log10() +
        scale_x_log10() +
        xlab("Mean counts per gene") +
        ylab("Variance per gene")

## Warning: Transformation introduced infinite values in continuous y-
axis

## Warning: Transformation introduced infinite values in continuous x-
axis
```

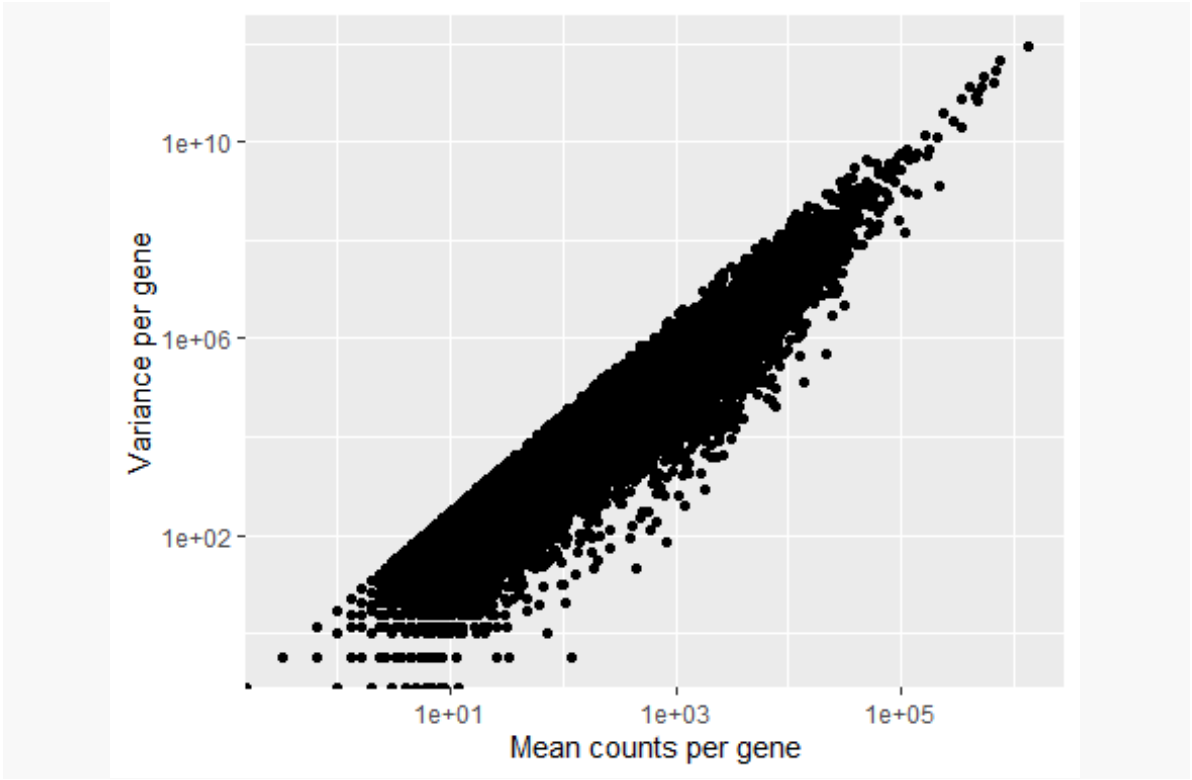

Figura 17. Gráfico varianza-media

```
plotDispEsts(dds)
```

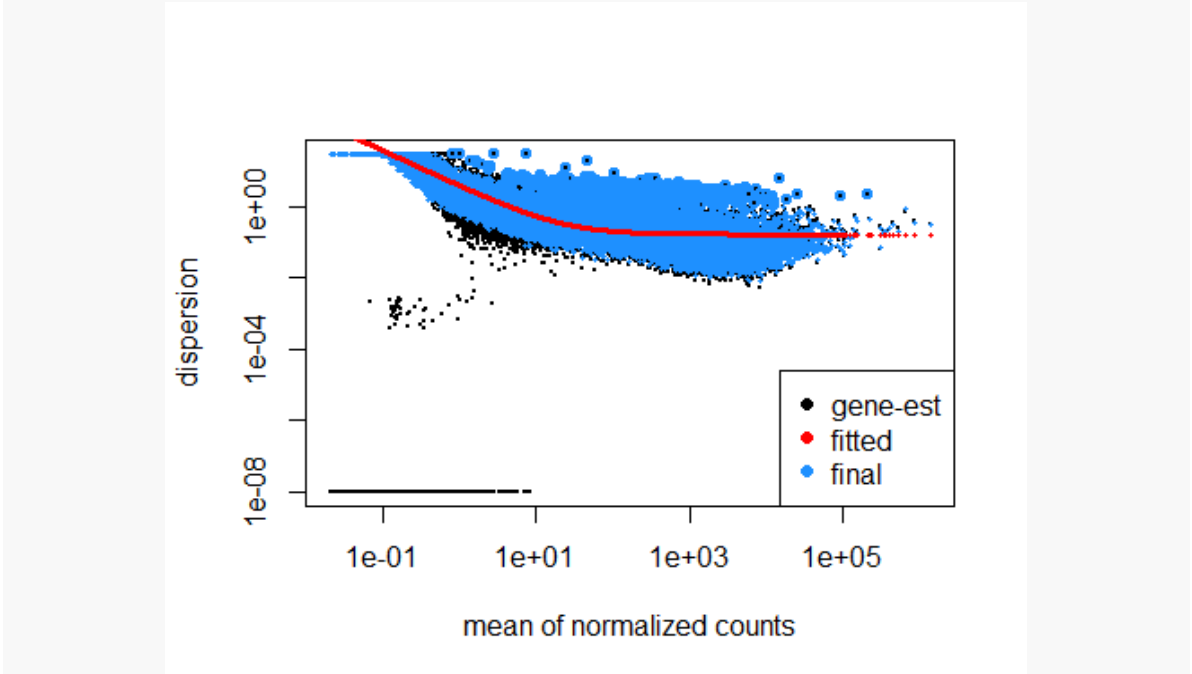

Figura 18. Gráfico de dispersión

8.4. Anotación de los resultados.

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

## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 56202 rows and 6 columns
##                          baseMean     log2FoldChange             lfcSE
##                         <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292 -0.506593549328503 0.649229336762504
## ENSG00000227232  703.913194096269 -0.109513559323702 0.209437857000647
## ENSG00000243485  1.65379907479822   -2.5783289351789  0.81642325585627
## ENSG00000237613  1.05107039541176  -1.27726670668977  0.98310765349482
## ENSG00000268020 0.499085044973889  0.564688910437072  1.34526171500069
## ...                           ...                ...               ...
## ENSG00000198695  42163.5862436026  -1.13239817562284 0.511299275780117
## ENSG00000210194   15.921378987448    -1.361430294971 0.857288610669824
## ENSG00000198727  401350.743812415 -0.132784145903131  0.31784480562689
## ENSG00000210195 0.919859172946447  0.693682237624121  1.15580481805625
## ENSG00000210196  2.10294725712943  -1.32305589012052  1.05327176073086
##                             stat             pvalue
padj
##                         <numeric>          <numeric>
<numeric>
## ENSG00000223972 -0.780299842663797  0.435214405402849
0.953291388478706
## ENSG00000227232 -0.522892856583055  0.601048815952088
0.98045612467988
## ENSG00000243485  -3.15807874982044 0.0015881265034332
0.149565827087083
## ENSG00000237613  -1.29921347082312  0.193870679803148
NA
## ENSG00000268020  0.419761377388771  0.674659782295886
NA
## ...                           ...                ...
...
## ENSG00000198695  -2.21474629295157 0.0267774834488893
0.58845365043088
## ENSG00000210194  -1.58806530032783  0.112271573793291
0.819590869251396
## ENSG00000198727 -0.417764089745745  0.676119608628538
0.988965659465185
## ENSG00000210195  0.600172474441408  0.548391295993563
NA
## ENSG00000210196  -1.25613914608559  0.209065525388696
0.891018804383927
```

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

## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 56202 rows and 6 columns
##                          baseMean     log2FoldChange             lfcSE
##                         <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292  0.682097597306958 0.696142710503623
## ENSG00000227232  703.913194096269 -0.166700548873979 0.209327862275152
## ENSG00000243485  1.65379907479822 -0.413473200895654 0.917820855420444
## ENSG00000237613  1.05107039541176   1.08973840875832   1.1216852644861
## ENSG00000268020 0.499085044973889 -0.332662683943004  1.29975894634954
## ...                           ...                ...               ...
## ENSG00000198695  42163.5862436026 -0.611563336273819 0.511300135645572
## ENSG00000210194   15.921378987448 -0.375226747031203 0.861117193646452
## ENSG00000198727  401350.743812415  0.155622618242794 0.317844837764405
## ENSG00000210195 0.919859172946447    2.2788491449908  1.23107967320807
## ENSG00000210196  2.10294725712943 -0.841630760699087  1.06161495771956
##                             stat             pvalue
padj
##                         <numeric>          <numeric>
<numeric>
## ENSG00000223972   0.97982437654701  0.327172817185538
0.630029254076678
## ENSG00000227232 -0.796361015022735    0.4258222308262
0.712696843015605
## ENSG00000243485 -0.450494449383857  0.652353955582283
0.856759168032771
## ENSG00000237613  0.971518877229418  0.331289957447704
0.634035983927653
## ENSG00000268020 -0.255941830504271  0.797995753993229
NA
## ...                           ...                ...
...
## ENSG00000198695   -1.1960946098746  0.231659640823698
0.528513618913861
## ENSG00000210194 -0.435744112183247  0.663022396720406
0.862119704578156
## ENSG00000198727  0.489618202823055  0.624404093229944
0.840800750101895
## ENSG00000210195   1.85109801955576 0.0641554529684609
NA
## ENSG00000210196 -0.792783442413985  0.427904004931932
0.714290844186617

result_eli_sfi <- results(dds, contrast = c ("Group", "ELI", "SFI"),
alpha=0.05)
result_eli_sfi
```

```
## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 56202 rows and 6 columns
##                          baseMean     log2FoldChange             lfcSE
##                         <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292  -1.18869114663546 0.676752393645434
## ENSG00000227232  703.913194096269 0.0571869895502778 0.209263197949233
## ENSG00000243485  1.65379907479822  -2.16485573428325  0.74564989710915
## ENSG00000237613  1.05107039541176  -2.36700511544809  1.04940446228125
## ENSG00000268020 0.499085044973889  0.897351594380076  1.31438998535319
## ...                            ...                ...               ...
## ENSG00000198695  42163.5862436026 -0.520834839349018 0.511295695491947
## ENSG00000210194   15.921378987448 -0.986203547939795 0.853033094090796
## ENSG00000198727  401350.743812415 -0.288406764145925 0.317844754892932
## ENSG00000210195 0.919859172946447  -1.58516690736668  1.26673208534181
## ENSG00000210196  2.10294725712943 -0.481425129421438  1.00674867664005
##                             stat              pvalue
padj
##                        <numeric>           <numeric>
<numeric>
## ENSG00000223972  -1.75646389698363  0.0790092307703652
0.292720270434619
## ENSG00000227232  0.273277815261867   0.784639667371412
0.91118708389076
## ENSG00000243485   -2.9033139314795 0.00369236307498618
0.0441084040134445
## ENSG00000237613  -2.25556989752318  0.0240975802159779
NA
## ENSG00000268020  0.682713353251052   0.494787990223506
NA
## ...                            ...                 ...
...
## ENSG00000198695  -1.01865680454809   0.308365926255845
0.601431310925325
## ENSG00000210194  -1.15611405321963   0.247634508527153
0.538656423086215
## ENSG00000198727 -0.907382486909612   0.364204567343912
0.65178710511791
## ENSG00000210195  -1.25138292912107   0.210794802408651
NA
## ENSG00000210196 -0.478197926247252   0.632509338372967
0.832477205763162

plotMA(result_eli_sfi)
```

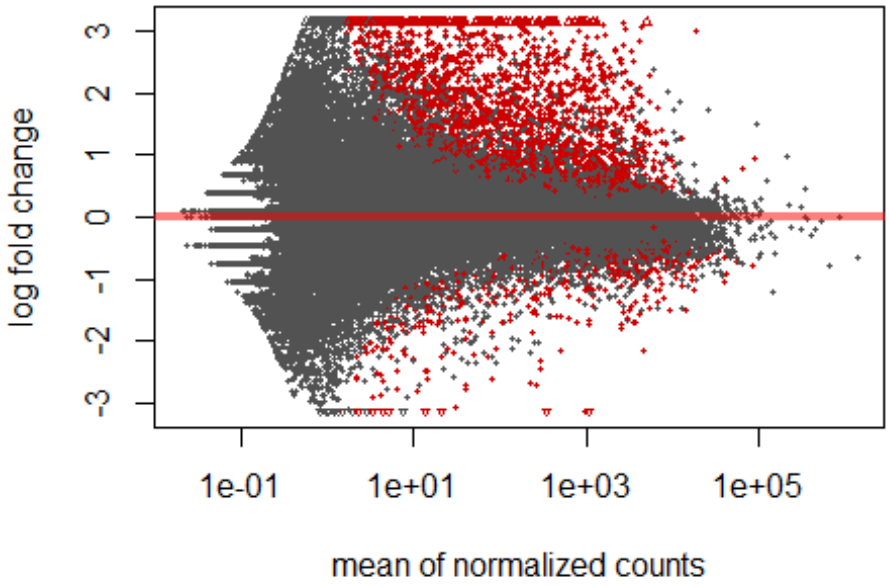

Figura 19. Diagrama MA comparación ELI-SFI

```
plotMA(result_nit_eli)
```

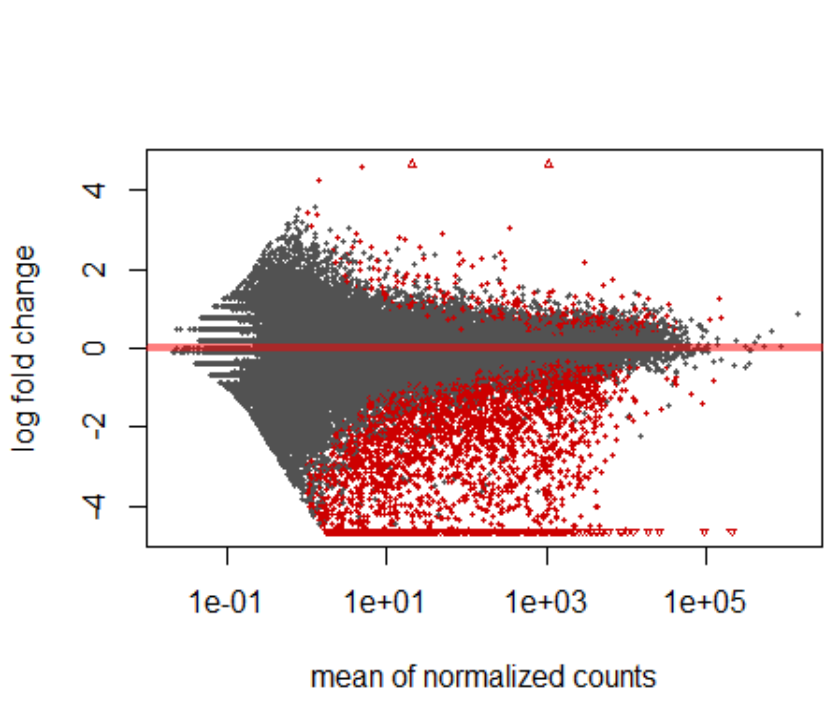

Figura 20. Diagrama MA comparación NIT-ELI

```
plotMA(result_nit_sfi)
```

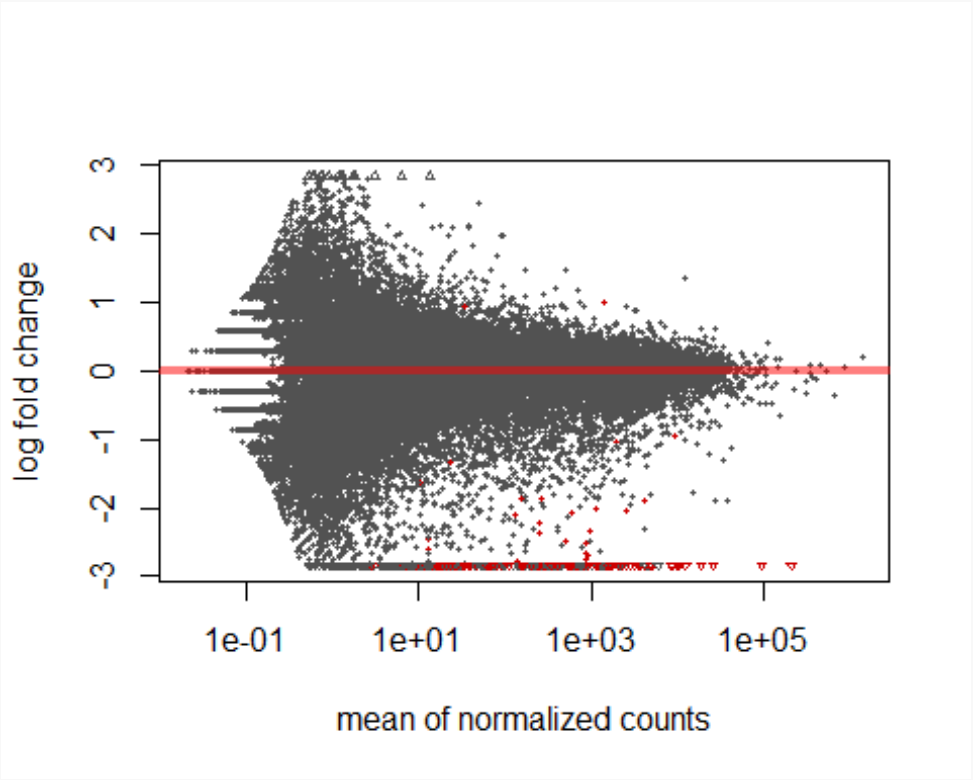

Figura 21. Gráfico dispersión NIT-SFI

8.5. Búsqueda de patrones de expresión y agrupación de las muestras.

```
mcols(result_eli_sfi)
```

```
## DataFrame with 6 rows and 2 columns
##                         type                                     description
##                  <character>                                     <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MLE): Group ELI vs SFI
## lfcSE               results          standard error: Group ELI vs SFI
## stat                results          Wald statistic: Group ELI vs SFI
## pvalue              results       Wald test p-value: Group ELI vs SFI
## padj                results                        BH adjusted p-values
```

```
mcols(result_nit_eli)
```

```
## DataFrame with 6 rows and 2 columns
##                         type                                     description
##                  <character>                                     <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MLE): Group NIT vs ELI
```

```
## lfcSE               results          standard error: Group NIT vs ELI
## stat                results          Wald statistic: Group NIT vs ELI
## pvalue              results       Wald test p-value: Group NIT vs ELI
## padj                results                        BH adjusted p-values

mcols(result_nit_sfi)

## DataFrame with 6 rows and 2 columns
##                         type                                 description
##                  <character>                                 <character>
## baseMean       intermediate mean of normalized counts for all samples
## log2FoldChange      results  log2 fold change (MLE): Group NIT vs SFI
## lfcSE               results          standard error: Group NIT vs SFI
## stat                results          Wald statistic: Group NIT vs SFI
## pvalue              results       Wald test p-value: Group NIT vs SFI
## padj                results                        BH adjusted p-values

head(result_eli_sfi, n=15)

## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 15 rows and 6 columns
##                            baseMean      log2FoldChange
lfcSE
##                           <numeric>           <numeric>
<numeric>
## ENSG00000223972   2.46823190608292   -1.18869114663546
0.676752393645434
## ENSG00000227232   703.913194096269  0.0571869895502778
0.209263197949233
## ENSG00000243485   1.65379907479822   -2.16485573428325
0.74564989710915
## ENSG00000237613   1.05107039541176   -2.36700511544809
1.04940446228125
## ENSG00000268020  0.499085044973889   0.897351594380076
1.31438998535319
## ...                            ...                 ...
...
## ENSG00000268903   3.23025737520392   0.705829411702717
0.614775506593162
## ENSG00000239906    10.557517315862 -0.0813488724790572
0.574628595570516
## ENSG00000241860   83.9431647633359   0.310156375153334
0.310394318430091
## ENSG00000222623 0.0606116738948831  -0.183084656405305
3.64968891400214
## ENSG00000241599   0.96352288046703   -1.41754248572977
0.979318445841682
##                                stat              pvalue
padj
##                           <numeric>           <numeric>
```

```
<numeric>
## ENSG00000223972   -1.75646389698363  0.0790092307703652
0.292720270434619
## ENSG00000227232   0.273277815261867   0.784639667371412
0.91118708389076
## ENSG00000243485    -2.9033139314795 0.00369236307498618
0.0441084040134445
## ENSG00000237613   -2.25556989752318  0.0240975802159779
NA
## ENSG00000268020   0.682713353251052   0.494787990223506
NA
## ...                             ...                 ...
...
## ENSG00000268903    1.14810919454833   0.250923488952598
0.54171550321259
## ENSG00000239906  -0.141567741504912   0.887421448799063
0.954156114131848
## ENSG00000241860   0.999233416133514   0.317681631764352
0.61137824823446
## ENSG00000222623 -0.0501644553054633   0.959991336431446
NA
## ENSG00000241599   -1.44747859263639   0.147762927876217
NA
```

```
head(result_nit_eli, n=15)
```

```
## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 15 rows and 6 columns
##                           baseMean     log2FoldChange
lfcSE
##                          <numeric>          <numeric>
<numeric>
## ENSG00000223972   2.46823190608292  0.682097597306958
0.696142710503623
## ENSG00000227232   703.913194096269 -0.166700548873979
0.209327862275152
## ENSG00000243485   1.65379907479822 -0.413473200895654
0.917820855420444
## ENSG00000237613   1.05107039541176   1.08973840875832
1.1216852644861
## ENSG00000268020  0.499085044973889 -0.332662683943004
1.29975894634954
## ...                             ...                ...
...
## ENSG00000268903   3.23025737520392   0.13588917429986
0.578867204078798
## ENSG00000239906    10.557517315862  0.247037506230939
0.57275077024091
## ENSG00000241860   83.9431647633359 -0.206655525098968
```

```
0.310104520871366
## ENSG00000222623 0.0606116738948831 -0.104503420559749
3.64968891400214
## ENSG00000241599   0.96352288046703  0.437959898758768
1.04628385463799
##                              stat            pvalue
padj
##                         <numeric>         <numeric>
<numeric>
## ENSG00000223972    0.97982437654701 0.327172817185538
0.630029254076678
## ENSG00000227232  -0.796361015022735   0.4258222308262
0.712696843015605
## ENSG00000243485  -0.450494449383857 0.652353955582283
0.856759168032771
## ENSG00000237613   0.971518877229418 0.331289957447704
0.634035983927653
## ENSG00000268020  -0.255941830504271 0.797995753993229
NA
## ...                            ...               ...
...
## ENSG00000268903   0.234750169542101 0.814402645911773
0.931864357242728
## ENSG00000239906   0.431317632496645 0.666237431227453
0.863874047024124
## ENSG00000241860  -0.666406037932902 0.505151604217333
0.770123061896645
## ENSG00000222623 -0.0286335145329287 0.977156882304159
NA
## ENSG00000241599     0.4185861196437  0.67551863552443
0.868751372385417
```

```
head(result_nit_sfi, n=15)
```

```
## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 15 rows and 6 columns
##                           baseMean     log2FoldChange
lfcSE
##                          <numeric>          <numeric>
<numeric>
## ENSG00000223972   2.46823190608292 -0.506593549328503
0.649229336762504
## ENSG00000227232   703.913194096269 -0.109513559323702
0.209437857000647
## ENSG00000243485   1.65379907479822   -2.5783289351789
0.81642325585627
## ENSG00000237613   1.05107039541176  -1.27726670668977
0.98310765349482
## ENSG00000268020  0.499085044973889  0.564688910437072
```

```
1.34526171500069
## ...                             ...                ...
...
## ENSG00000268903   3.23025737520392  0.841718586002577
0.616276608059853
## ENSG00000239906    10.557517315862  0.165688633751882
0.573926017387686
## ENSG00000241860   83.9431647633359  0.103500850054366
0.311433364124079
## ENSG00000222623 0.0606116738948831 -0.287588076965054
3.64973752994955
## ENSG00000241599   0.96352288046703 -0.979582586971003
0.953498509362182
##                                stat             pvalue
padj
##                           <numeric>          <numeric>
<numeric>
## ENSG00000223972  -0.780299842663797  0.435214405402849
0.953291388478706
## ENSG00000227232  -0.522892856583055  0.601048815952088
0.98045612467988
## ENSG00000243485   -3.15807874982044 0.0015881265034332
0.149565827087083
## ENSG00000237613   -1.29921347082312  0.193870679803148
NA
## ENSG00000268020   0.419761377388771  0.674659782295886
NA
## ...                             ...                ...
...
## ENSG00000268903    1.36581297260731  0.171997673589929
0.871153449182669
## ENSG00000239906   0.288693365925524  0.772816039874319
0.995337565395347
## ENSG00000241860   0.332337064609204  0.739634755323864
0.993723847744319
## ENSG00000222623 -0.0787969202182682  0.937194153742128
NA
## ENSG00000241599    -1.0273561807939  0.304252776992182
NA

summary(result_eli_sfi)

##
## out of 46140 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 2118, 4.6%
## LFC < 0 (down)     : 554, 1.2%
## outliers [1]       : 0, 0%
## low counts [2]     : 16803, 36%
## (mean count < 1)
```

```
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

summary(result_nit_eli)

##
## out of 46140 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 573, 1.2%
## LFC < 0 (down)     : 3015, 6.5%
## outliers [1]       : 0, 0%
## low counts [2]     : 15035, 33%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

summary(result_nit_sfi)

##
## out of 46140 with nonzero total read count
## adjusted p-value < 0.05
## LFC > 0 (up)       : 5, 0.011%
## LFC < 0 (down)     : 152, 0.33%
## outliers [1]       : 0, 0%
## low counts [2]     : 16803, 36%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

library(AnnotationDbi)

##
## Attaching package: 'AnnotationDbi'

## The following object is masked from 'package:dplyr':
##
##     select

library(org.Hs.eg.db)

##

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

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

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

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

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

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

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

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

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

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

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

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

head(result_eli_sfi)

## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 6 rows and 8 columns
##                          baseMean     log2FoldChange             lfcSE
##                         <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292  -1.18869114663546 0.676752393645434
## ENSG00000227232  703.913194096269 0.0571869895502778 0.209263197949233
## ENSG00000243485  1.65379907479822  -2.16485573428325  0.74564989710915
## ENSG00000237613  1.05107039541176  -2.36700511544809  1.04940446228125
## ENSG00000268020 0.499085044973889  0.897351594380076  1.31438998535319
## ENSG00000240361  1.12125694215739  -1.11711918488111 0.830885380734276
##                             stat              pvalue
padj
##                         <numeric>           <numeric>
<numeric>
## ENSG00000223972 -1.75646389698363  0.0790092307703652
0.292720270434619
## ENSG00000227232 0.273277815261867   0.784639667371412
0.91118708389076
## ENSG00000243485  -2.9033139314795 0.00369236307498618
0.0441084040134445
## ENSG00000237613 -2.25556989752318  0.0240975802159779
NA
## ENSG00000268020 0.682713353251052   0.494787990223506
NA
## ENSG00000240361  -1.3444925266273   0.178789171475878
NA
##                      symbol          GO
```

```
##                 <character> <character>
## ENSG00000223972     DDX11L1          NA
## ENSG00000227232          NA          NA
## ENSG00000243485          NA          NA
## ENSG00000237613     FAM138A          NA
## ENSG00000268020          NA          NA
## ENSG00000240361          NA          NA
```

```
head(result_nit_eli)
```

```
## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 6 rows and 8 columns
##                         baseMean     log2FoldChange             lfcSE
##                        <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292  0.682097597306958 0.696142710503623
## ENSG00000227232  703.913194096269 -0.166700548873979 0.209327862275152
## ENSG00000243485  1.65379907479822 -0.413473200895654 0.917820855420444
## ENSG00000237613  1.05107039541176   1.08973840875832   1.1216852644861
## ENSG00000268020 0.499085044973889 -0.332662683943004  1.29975894634954
## ENSG00000240361  1.12125694215739  0.533250745977481 0.870835493451701
##                             stat            pvalue              padj
##                        <numeric>         <numeric>         <numeric>
## ENSG00000223972   0.97982437654701 0.327172817185538 0.630029254076678
## ENSG00000227232 -0.796361015022735   0.4258222308262 0.712696843015605
## ENSG00000243485 -0.450494449383857 0.652353955582283 0.856759168032771
## ENSG00000237613  0.971518877229418 0.331289957447704 0.634035983927653
## ENSG00000268020 -0.255941830504271 0.797995753993229                NA
## ENSG00000240361   0.61234383530218 0.540310293062119 0.791832294651999
##                      symbol          GO
##                 <character> <character>
## ENSG00000223972     DDX11L1          NA
## ENSG00000227232          NA          NA
## ENSG00000243485          NA          NA
## ENSG00000237613     FAM138A          NA
## ENSG00000268020          NA          NA
## ENSG00000240361          NA          NA
```

```
head(result_nit_sfi)
```

```
## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 6 rows and 8 columns
##                         baseMean     log2FoldChange             lfcSE
##                        <numeric>          <numeric>         <numeric>
## ENSG00000223972  2.46823190608292 -0.506593549328503 0.649229336762504
## ENSG00000227232  703.913194096269 -0.109513559323702 0.209437857000647
## ENSG00000243485  1.65379907479822   -2.5783289351789  0.81642325585627
## ENSG00000237613  1.05107039541176  -1.27726670668977  0.98310765349482
## ENSG00000268020 0.499085044973889  0.564688910437072  1.34526171500069
## ENSG00000240361  1.12125694215739 -0.583868438903633 0.796643658284814
```

```
##                              stat             pvalue
padj
##                         <numeric>          <numeric>
<numeric>
## ENSG00000223972 -0.780299842663797  0.435214405402849
0.953291388478706
## ENSG00000227232 -0.522892856583055  0.601048815952088
0.98045612467988
## ENSG00000243485  -3.15807874982044 0.0015881265034332
0.149565827087083
## ENSG00000237613  -1.29921347082312  0.193870679803148
NA
## ENSG00000268020  0.419761377388771  0.674659782295886
NA
## ENSG00000240361 -0.732910420903507  0.463613066813478
NA
##                      symbol          GO
##                 <character> <character>
## ENSG00000223972     DDX11L1          NA
## ENSG00000227232          NA          NA
## ENSG00000243485          NA          NA
## ENSG00000237613     FAM138A          NA
## ENSG00000268020          NA          NA
## ENSG00000240361          NA          NA
```

8.6. Análisis de significación biológica.

```
with(result_eli_sfi, plot(log2FoldChange, -log10(pvalue), pch=20,
main="Volcano plot", xlim=c(-2.5,2), col="dark violet"))
```

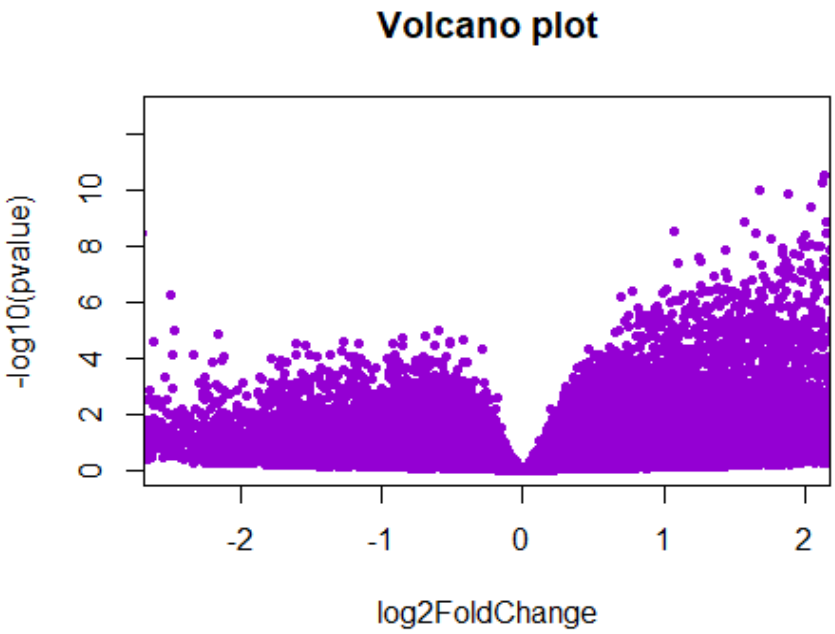

Figura 22. Gráfico de volcán comparación ELI-SFI

```
with(result_nit_eli, plot(log2FoldChange, -log10(pvalue), pch=20,
main="Volcano plot", xlim=c(-2.5,2), col="dark violet"))
```

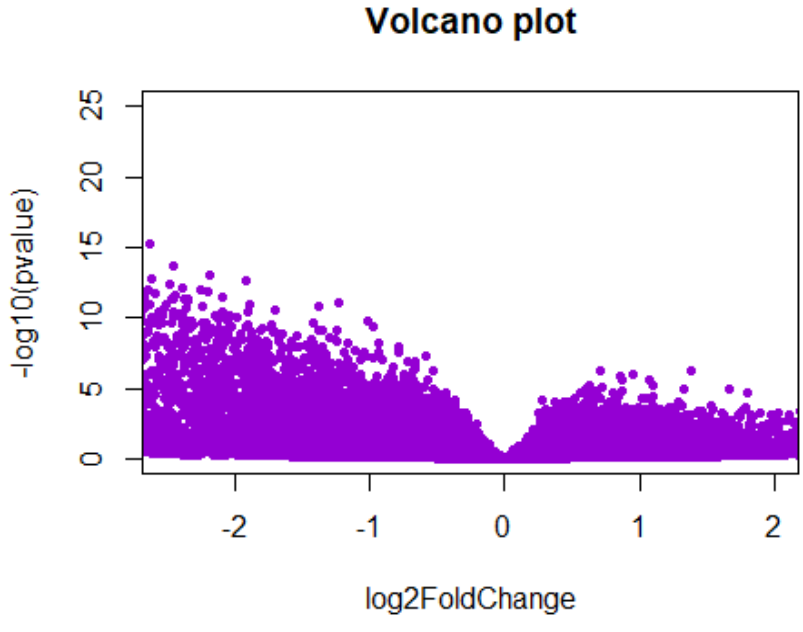

Figura 23. Gráfico de volcán comparación NIT-ELI

```
with(result_nit_sfi, plot(log2FoldChange, -log10(pvalue), pch=20,
main="Volcano plot", xlim=c(-2.5,2), col="dark violet"))
```

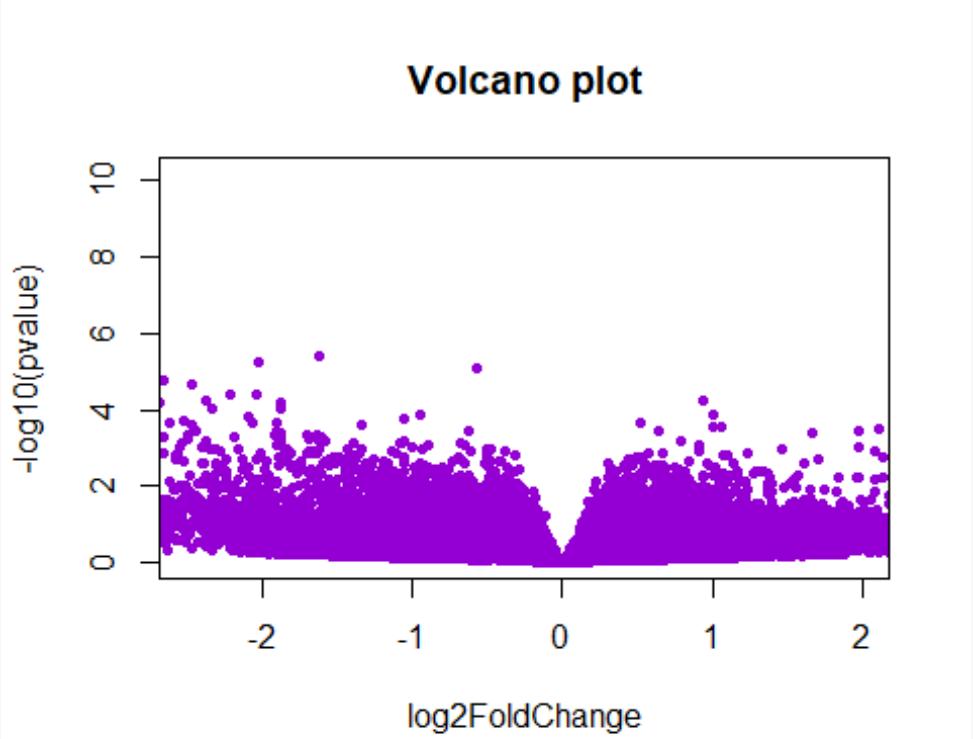

Figura 24. Gráfico volcán comparación NIT-SFI

```
result_eli_sfi <- as.data.frame(result_eli_sfi)
write.csv(result_eli_sfi[1:20,], file="/Users/maria/OneDrive/UOC/Análisis
de datos ómicos/PEC 2/data/result_eli_Sfi.csv")

result_nit_eli <- as.data.frame(result_nit_eli)
write.csv(result_nit_eli[1:20,], file="/Users/maria/OneDrive/UOC/Análisis
de datos ómicos/PEC 2/data/result_nit_eli.csv")

result_nit_sfi <- as.data.frame(result_nit_sfi)
write.csv(result_nit_sfi[1:20,], file="/Users/maria/OneDrive/UOC/Análisis
de datos ómicos/PEC 2/data/result_nit_Sfi.csv")
```

8.7. Información de la sesión.

```
sessionInfo()

## R version 3.6.3 (2020-02-29)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 18362)
##
```

```
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.1252  LC_CTYPE=Spanish_Spain.1252
## [3] LC_MONETARY=Spanish_Spain.1252 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.1252
##
## attached base packages:
## [1] stats4    parallel  stats     graphics  grDevices utils
datasets
## [8] methods   base
##
## other attached packages:
##  [1] org.Hs.eg.db_3.10.0         AnnotationDbi_1.48.0
##  [3] ggplot2_3.3.1               pheatmap_1.0.12
##  [5] DESeq2_1.26.0               SummarizedExperiment_1.16.1
##  [7] DelayedArray_0.12.3         BiocParallel_1.20.1
##  [9] matrixStats_0.56.0          Biobase_2.46.0
## [11] GenomicRanges_1.38.0        GenomeInfoDb_1.22.1
## [13] IRanges_2.20.2              S4Vectors_0.24.4
## [15] BiocGenerics_0.32.0         dplyr_1.0.0
## [17] readxl_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] bit64_0.9-7             splines_3.6.3           Formula_1.2-3
##  [4] latticeExtra_0.6-29     blob_1.2.1
GenomeInfoDbData_1.2.2
##  [7] cellranger_1.1.0        yaml_2.2.1              RSQLite_2.2.0
## [10] pillar_1.4.4            backports_1.1.7         lattice_0.20-41
## [13] glue_1.4.1              digest_0.6.25           RColorBrewer_1.1-2
## [16] XVector_0.26.0          checkmate_2.0.0         colorspace_1.4-1
## [19] htmltools_0.4.0         Matrix_1.2-18           XML_3.99-0.3
## [22] pkgconfig_2.0.3         genefilter_1.68.0       zlibbioc_1.32.0
## [25] purrr_0.3.4             xtable_1.8-4            scales_1.1.1
## [28] jpeg_0.1-8.1            tibble_3.0.1            htmlTable_1.13.3
## [31] annotate_1.64.0         farver_2.0.3            generics_0.0.2
## [34] ellipsis_0.3.0          withr_2.2.0             nnet_7.3-12
## [37] survival_3.1-12         magrittr_1.5            crayon_1.3.4
## [40] memoise_1.1.0           evaluate_0.14           foreign_0.8-76
## [43] tools_3.6.3             data.table_1.12.8       lifecycle_0.2.0
## [46] stringr_1.4.0           locfit_1.5-9.4          munsell_0.5.0
## [49] cluster_2.1.0           compiler_3.6.3          rlang_0.4.6
## [52] grid_3.6.3              RCurl_1.98-1.2          rstudioapi_0.11
## [55] htmlwidgets_1.5.1       labeling_0.3            bitops_1.0-6
## [58] base64enc_0.1-3         rmarkdown_2.2           gtable_0.3.0
## [61] DBI_1.1.0               R6_2.4.1                gridExtra_2.3
## [64] knitr_1.28              bit_1.1-15.2            Hmisc_4.4-0
## [67] stringi_1.4.6           Rcpp_1.0.4.6            geneplotter_1.64.0
## [70] vctrs_0.3.1             rpart_4.1-15            acepack_1.4.1
## [73] png_0.1-7               tidyselect_1.1.0        xfun_0.14
```