Análisis de Datos Ómicos - PEC2

Juan A. Martínez Velasco

Junio 2020

Tabla de Contenidos

Resumen

Este documento presenta el proceso de análisis de datos de ultrasecuenciación obtenidos a partir de las muestras disponibles en el repositorio GTEx, que contiene datos de distintos tejidos. El estudio se centra en los datos de expresión pertenecientes a un análisis del tiroides en donde se comparan tres tipos de infiltración medidos en un total de 292 muestras pertenecientes a tres grupos: (i) Not infiltrated tissues (NIT) - 236 muestras; (ii) Small focal infiltrates (SFI) - 42 muestras; (iii) Extensive lymphoid infiltrates (ELI) - 14 muestras. El documento presenta el flujo de trabajo implantado para analizar, filtrar y normalizar los datos, realizar un estudio de expresión diferencial, anotar los resultados, buscar patrones de expresión y realizar un análisis de expresión biológica. El estudio ha sido realizado utilizando las prestaciones de la librería *edgeR* disponible en *Bioconductor*. Se ha creado un repositorio

en Github (https://github.com/juan-antonio1952/ADO-PEC2.git) con toda la información necesaria para poder reproducir los resultados presentados en este documento.

Introducción

Los objetivos finales del presente trabajo son la realización de un estudio de expresión diferencial de genes y un análisis de expresión biológica a partir de los datos recibidos en dos archivos en formato *csv*: *targets* y *counts*.

Estos archivos contienen datos de ultrasecuenciación de 54 tejidos distintos obtenidos a partir de las muestras disponibles en el repositorio GTEx. El estudio se centra en los datos de expresión pertenecientes a un análisis del tiroides en donde se comparan tres tipos de infiltración medidos en un total de 292 muestras pertenecientes a tres grupos: (i) Not infiltrated tissues (NIT) - 236 muestras; (ii) Small focal infiltrates (SFI) - 42 muestras; (iii) Extensive lymphoid infiltrates (ELI) - 14 muestras.

El trabajo a realizar se basará en una selección de 10 muestras de cada uno de los tres grupos (NIT, SFI y ELI).

El flujo de trabajo (*workflow*) ha sido organizado de la siguiente foma:

1. Descarga, organización y análisis de los datos.
2. Preprocesado de los datos: filtraje y normalización.
3. Identificación de genes diferencialmente expresados.
4. Anotación y enriquecimiento de genes.
5. Busca de patrones de expresión y agrupación de las muestras.
6. Análisis de significación biológica.

El trabajo ha sido realizado utilizando las prestaciones de la librería *edgeR* (Empirical Analysis of Digital Gene Expression Data in R), desarrollada por investigadores del Garvan Institute of Medical Research y The Walter and Eliza Hall Institute of Medical Research (Australia), y disponible en *Bioconductor* [1].

Esta librería es de manejo relativamente fácil para los objetivos de este trabajo. Por otro lado, existe una abundante bibliografía [2], [3], [4], [5], que facilita la aplicación de las prestaciones necesarias para estudios de expresión diferencial y significación biológica. Por lo que respecta al flujo de trabajo, las referencias [6], [7] y [8] han sido de gran utilidad; especialmente la primera. Otros documentos que han sido utilizados en este trabajo se encuentran en las referencias [9] y [10].

Existen una abundante literatura sobre tecnologías de ultrasecuenciación (NGS, *Next generation Sequencing*) y los procedimientos a seguir para analizar los datos; ver referencias [11], [12], [13], [14], [15], [16], [17] y [18].

En la última sección de este documento se han añadido los enlaces de otras páginas web que han sido consultadas durante la realización del presente estudio.

Entrada y preparación de datos

Se leen los dos archivos cuyos datos se han de utilizar en este trabajo: *targets* y *counts*; los dos en formato `csv`.

- Archivo *targets*

  Una vez descargado, hay que seleccionar de forma aleatoria 10 muestras de cada uno de los tres grupos existentes (ELI, NIT, SFI); estas muestras serán las que servirán para la realización del estudio. El procedimiento a seguir para conseguir esto es evidente: (i) se lee/descarga el archivo; (ii) se realiza una selección aleatoria de 10 muestras para cada grupo; (iii) finalmente, para facilitar el estudio, se renombran las muestras utilizando las referencias abreviadas, disponibles en el mismo archivo.

  NOTA: En la información disponible en el archivo `targets` aparecen muestras obtenidas mediante ultra-secuenciación y otras técnicas. Se considera que ambas son válidas y se utilizan todas las seleccionadas.

```
Archivo targets

[1] "data.frame"

'data.frame':   292 obs. of  8 variables:
 $ SRA_Sample         : Factor w/ 292 levels "SRS333004","SRS333021",..: 110
234 98 102 86 92 168 281 82 157 ...
 $ Sample_Name        : Factor w/ 292 levels "GTEX-111CU-0226-SM-5GZXC",..: 1
2 3 4 5 6 7 8 9 10 ...
 $ Grupo_analisis     : int  1 1 3 1 1 1 1 1 2 1 ...
 $ body_site          : Factor w/ 1 level "Thyroid": 1 1 1 1 1 1 1 1 1 1 ...
 $ molecular_data_type: Factor w/ 2 levels "Allele-Specific Expression",..: 1
2 2 1 2 1 2 1 2 1 ...
 $ sex                : Factor w/ 2 levels "female","male": 2 2 2 2 1 1 1 2 2
2 ...
 $ Group              : Factor w/ 3 levels "ELI","NIT","SFI": 2 2 1 2 2 2 2 2
3 2 ...
 $ ShortName          : Factor w/ 292 levels "111CU_NIT","111FC_NIT",..: 1 2
3 4 5 6 7 8 9 10 ...

 [1] 177  70 260 272 274 192  77 121 101 124 162  14  99  52 243 188  49  79
203
[20] 232   3 251 100 149 186  40 290 146 119 253

 [1] QEL4-_NIT 132AR_NIT ZDYS-_NIT ZTPG-_NIT ZTX8-_NIT RU1J-_NIT 139T6_NIT
 [8] 13S86_NIT 13NZA_NIT 13VXU_NIT OXRP-_SFI 11DXY_SFI 13NZ8_SFI 12WSG_SFI
[15] Y5V6-_SFI RM2N-_SFI 12BJ1_SFI 139UW_SFI SIU8-_SFI XMK1-_SFI 111VG_ELI
[22] YFC4-_ELI 13NZ9_ELI 14BMU_ELI R55G-_ELI 11XUK_ELI ZYY3-_ELI 14ABY_ELI
[29] 13QJC_ELI YJ89-_ELI
292 Levels: 111CU_NIT 111FC_NIT 111VG_ELI 111YS_NIT 1122O_NIT ... ZZPU-_NIT
```

- Archivo *counts*

  Se procede como con el archivo *targets*. Una vez descargado el nuevo archivo se seleccionan las mismas muestras que en *targets*.

```
 [1] "GTEX.QEL4.0726.SM.3GIJ5"  "GTEX.132AR.1126.SM.5P9GA"
 [3] "GTEX.ZDYS.0626.SM.5J2N5"  "GTEX.ZTPG.0826.SM.5DUVC"
 [5] "GTEX.ZTX8.0626.SM.59HKC"  "GTEX.RU1J.0226.SM.2TF5Y"
 [7] "GTEX.139T6.0326.SM.5J2LY" "GTEX.13S86.1126.SM.5RQJX"
 [9] "GTEX.13NZA.1026.SM.5MR48" "GTEX.13VXU.0826.SM.5KLZ2"
[11] "GTEX.OXRP.0326.SM.33HBJ"  "GTEX.11DXY.0426.SM.5H12R"
[13] "GTEX.13NZ8.0226.SM.5J2OK" "GTEX.12WSG.0226.SM.5EGIF"
[15] "GTEX.Y5V6.0526.SM.4VBRV"  "GTEX.RM2N.0526.SM.2TF4N"
[17] "GTEX.12BJ1.0426.SM.5FQSO" "GTEX.139UW.0126.SM.5KM1B"
[19] "GTEX.SIU8.0626.SM.2XCDN"  "GTEX.XMK1.0626.SM.4B65A"
[21] "GTEX.111VG.0526.SM.5N9BW" "GTEX.YFC4.2626.SM.5P9FQ"
[23] "GTEX.13NZ9.1126.SM.5MR37" "GTEX.14BMU.0226.SM.5S2QA"
[25] "GTEX.R55G.0726.SM.2TC6J"  "GTEX.11XUK.0226.SM.5EQLW"
[27] "GTEX.ZYY3.1926.SM.5GZXS"  "GTEX.14ABY.0926.SM.5Q5DY"
[29] "GTEX.13QJC.0826.SM.5RQKC" "GTEX.YJ89.0726.SM.5P9F7"

Archivo counts

[1] 56202    30
```

	QEL4-_NIT	132AR_NIT	ZDYS-_NIT	ZTPG-_NIT	ZTX8-_NIT
ENSG00000223972.4	4	0	5	1	3
ENSG00000227232.4	511	907	637	524	586
ENSG00000243485.2	2	1	3	1	0
ENSG00000237613.2	2	1	4	0	2
ENSG00000268020.2	4	0	1	0	2

  Se observa que el archivo final tiene datos de 30 muestras correspondientes a 56202 tránscritos.

Para completar el archivo de trabajo, se crea un tercer objeto; *genes*, con los nombres de las filas del archivo original. En esta primera versión, es un archivo con una sola columna que se nombra *RefENSMBL*, y en la que los nombres de los genes se han reducido, excluyendo el número de la versión. Se le añade una segunda columna con el identificador *Entrez* de los genes.

RefENSMBL
ENSG00000223972
ENSG00000227232
ENSG00000243485
ENSG00000237613

ENSG00000268020
ENSG00000240361

	RefENSMBL	entrezIDs
ENSG00000223972	ENSG00000223972	100287102
ENSG00000227232	ENSG00000227232	NA
ENSG00000243485	ENSG00000243485	NA
ENSG00000237613	ENSG00000237613	645520
ENSG00000268020	ENSG00000268020	NA
ENSG00000240361	ENSG00000240361	NA

Se observa que hay un elevado número de valores perdidos, NAs.

```
Número de NAs en código ENTREZ = 31704
```

La librería *edgeR* trabaja con una tabla de las lecturas que hay en el objeto *counts*, con filas correspondientes a los genes y columnas correspondientes a las muestras. Esta librería almacena los datos en un simple objeto de clase `DGEList` basado en listas.

La última tarea en esta primera etapa consiste en convertir los datos descargados en un objeto de clase `DGEList`, quitar las filas de los genes con todas las entradas a 0 y renombrar los grupos del objeto creado según los acrónimos asignados a los grupos de las muestras de tejidos analizadas (NIT, SFI, ELI). La estructura inicial tiene tres objetos: `counts`, `samples` y `genes`.

```
Archivo - Clase DGEList

[1] "counts"  "samples" "genes"
```

Se muestra una selección del contenido de los tres. En el objeto `samples` se renombran la variable grupo según el grupo de muestras (NIT, SIF, ELI) y se presenta una selección que cubre los tres grupos.

```
Objeto counts

[1] 46387    30
```

	QEL4-_NIT	132AR_NIT	ZDYS-_NIT
ENSG00000223972.4	4	0	5
ENSG00000227232.4	511	907	637
ENSG00000243485.2	2	1	3
ENSG00000237613.2	2	1	4
ENSG00000268020.2	4	0	1

```
Objeto samples
```

```
[1] 30  3
```

```
Antes de renombrar los grupos
```

	group	lib.size	norm.factors
QEL4-_NIT	1	38646199	1
132AR_NIT	1	70926853	1
ZDYS-_NIT	1	60210708	1
OXRP-_SFI	1	89131381	1
11DXY_SFI	1	53087541	1
13NZ8_SFI	1	59535746	1
111VG_ELI	1	52199752	1
YFC4-_ELI	1	81226878	1
13NZ9_ELI	1	61447691	1

```
Después de renombrar los grupos
```

	group	lib.size	norm.factors
QEL4-_NIT	NIT	38646199	1
132AR_NIT	NIT	70926853	1
ZDYS-_NIT	NIT	60210708	1
OXRP-_SFI	SIF	89131381	1
11DXY_SFI	SIF	53087541	1
13NZ8_SFI	SIF	59535746	1
111VG_ELI	ELI	52199752	1
YFC4-_ELI	ELI	81226878	1
13NZ9_ELI	ELI	61447691	1

```
Objeto genes
```

	RefENSMBL	entrezIDs
ENSG00000223972.4	ENSG00000223972	100287102
ENSG00000227232.4	ENSG00000227232	NA
ENSG00000243485.2	ENSG00000243485	NA
ENSG00000237613.2	ENSG00000237613	645520
ENSG00000268020.2	ENSG00000268020	NA

El archivo final que se analizará a continuación tiene 46387 tránscritos provenientes de las 30 muestras seleccionadas.

Análisis de los datos

Los valores de las cuentas (*reads*) de un estudio de ultra-secuenciación están muy sesgadas. Para analizar y visualizar datos, es muy útil trabajar con una versión de datos transformados en la que se utiliza el logaritmo de base 2

$$y = \log_2(K + k)$$

donde K es el número de las cuentas y k una constante positiva (aquí se usa 0.5), que evitará problemas cuando el valor de las cuentas sea 0.

Se presentan algunos resultados después de realizar la transformación logarítmica.

```
Datos transformados
```

	QEL4-_NIT	132AR_NIT	ZDYS-_NIT	ZTPG-_NIT	ZTX8-_NIT
ENSG00000223972.4	2.169925	-1.0000000	2.4594316	0.5849625	1.807355
ENSG00000227232.4	8.998590	9.8257538	9.3162815	9.0347990	9.195987
ENSG00000243485.2	1.321928	0.5849625	1.8073549	0.5849625	-1.000000
ENSG00000237613.2	1.321928	0.5849625	2.1699250	-1.0000000	1.321928
ENSG00000268020.2	2.169925	-1.0000000	0.5849625	-1.0000000	1.321928

A continuación, se presentan algunos resultados gráficos; como paso previo, se escogen los colores que serán utilizados en algunas figuras, teniendo en cuenta los factores (NIT, SFI, ELI) de las muestras a analizar.

La Figura 1 compara la distribución de una muestra antes y después de transformar los datos. Se puede comprobar fácilmente el efecto de la transformación logarítmica por el cambio en la distribución de las frecuencias que aparecen en cada versión.

Un diagrama de cajas puede ser muy útil para visualizar la distribución de valores entre muestras, contrastar la distribución de valores de expresión a nivel de genes en distintas muestras, o comprobar el efecto del filtrado. La Figura 2 muestra los diagramas de cajas para el caso en estudio antes de filtrar. Se pueden observar las diferencias entre las muestras de cada grupo y entre las muestras de diferentes grupos; por otro lado, es evidente el efecto de la transformación logarítmica que reduce de forma significativa las diferencias.

Un gráfico tipo MA presenta valores M (el logaritmo de la relación entre las cuentas de cada gen entre dos muestras) frente a valores A (el nivel medio de cuentas para cada gen entre dos muestras). Este tipo de gráfico se utiliza para presentar la reproductibilidad entre muestras de un experimento y permite deducir si es necesaria la normalización de los datos originales: en un gráfico MA los genes con una expresión similar en dos muestras se sitúan en la recta horizontal, $y = 0$. La Figura 3 muestra este tipo de gráfico para las dos primeras muestras del estudio.

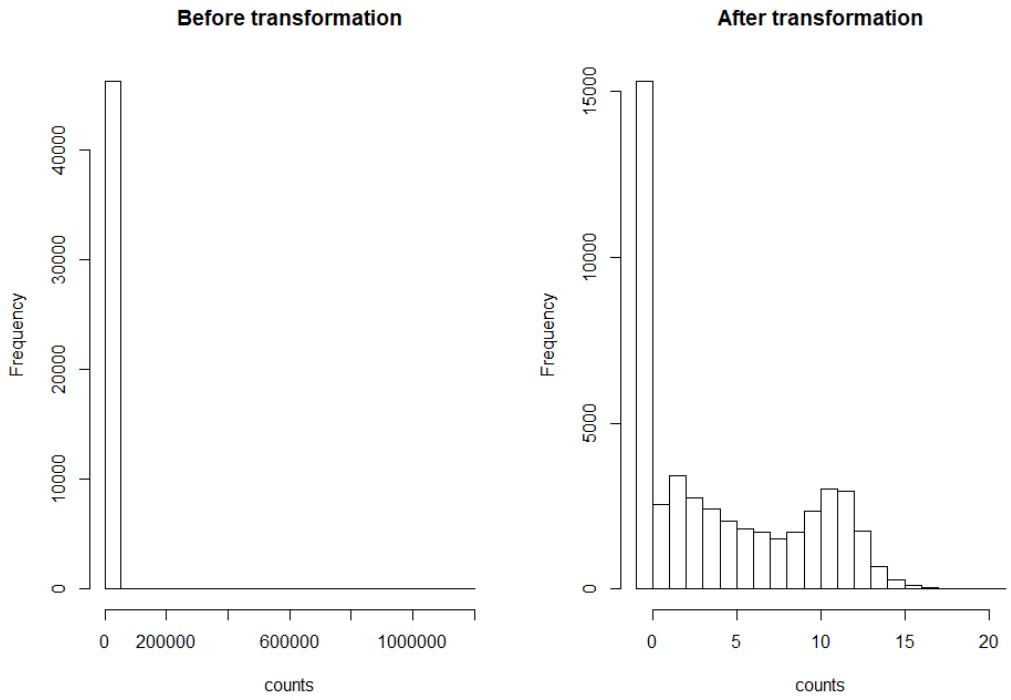

Figura 1. Histogramas - Muestra 12.

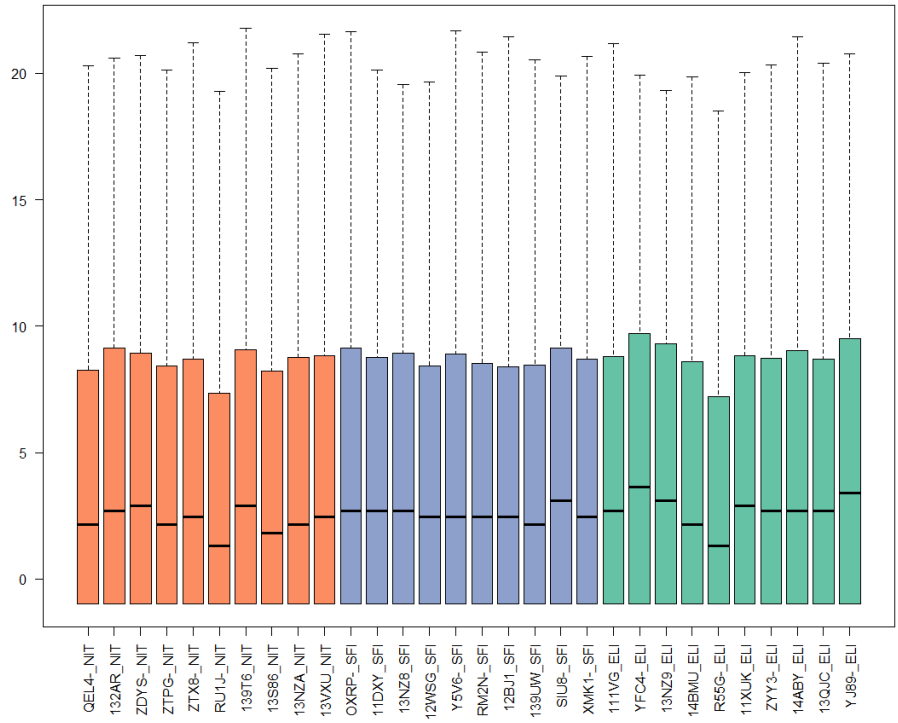

Figura 2. Diagramas de cajas - Muestras transformadas.

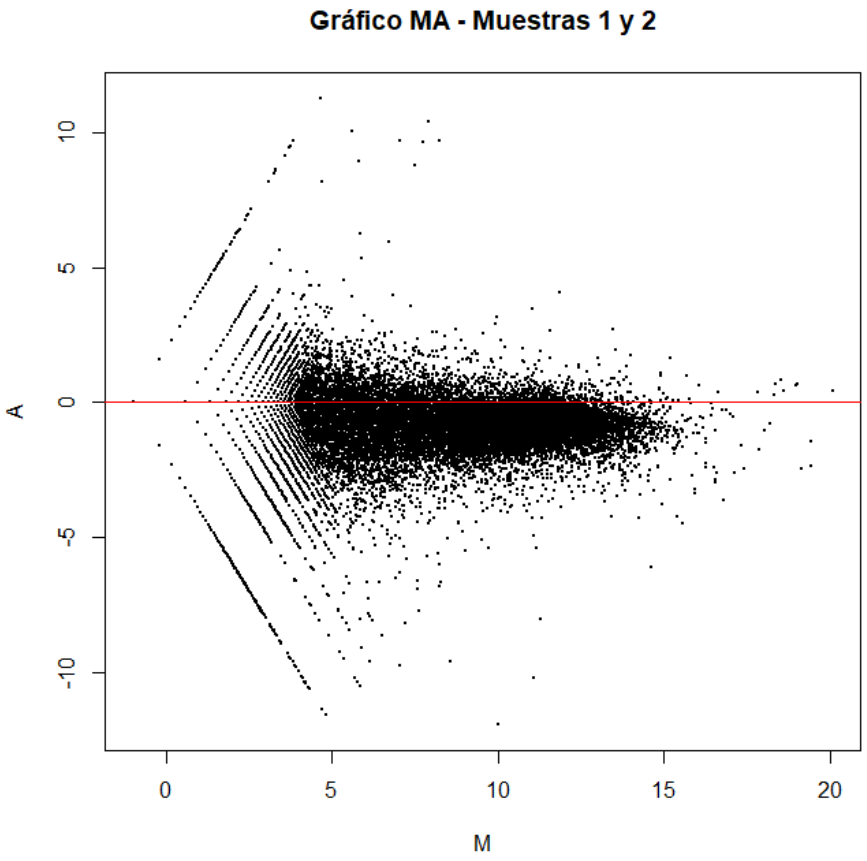

Figura 3. Gráfico MA - Muestras 1-2.

El objetivo de una escalado multidimensional (MDS en inglés) es presentar un patrón de similaridad entre un conjunto de objetos. En general, la similaridad entre cada par de muestras se mide con la distancia euclídea entre las cuentas de los genes. La Figura 4 muestra el escalado multidimensional de los datos transformados. Esta figura ha sido obtenida con la opción `geneselection = "common"` que escoge los genes que presentan una máxima desviación entre muestras. El resultado con la opción mencionada muestra una clara división entre dos conglomerados en cada uno de los cuales se encuentran muestras de los tres grupos.

Dendrogramas y mapas de colores son dos opciones gráficas que permiten visualizar de forma relativamente rápida la división de muestras entre grupos o conglomerados utilizando las distancias entre muestras. Un mapa de colores es un gráfico bidimensional en el que los datos se presentan con un determinado color que tiene en cuenta su situación en la matriz de distancias: las filas y columnas se disponen de acuerdo con una organización jerárquica en la que filas y columnas similares se colocan unas cerca de la otras. Las Figuras 5 and 6 muestran el dendrograma y el mapa de color que resultan con los datos transformados.

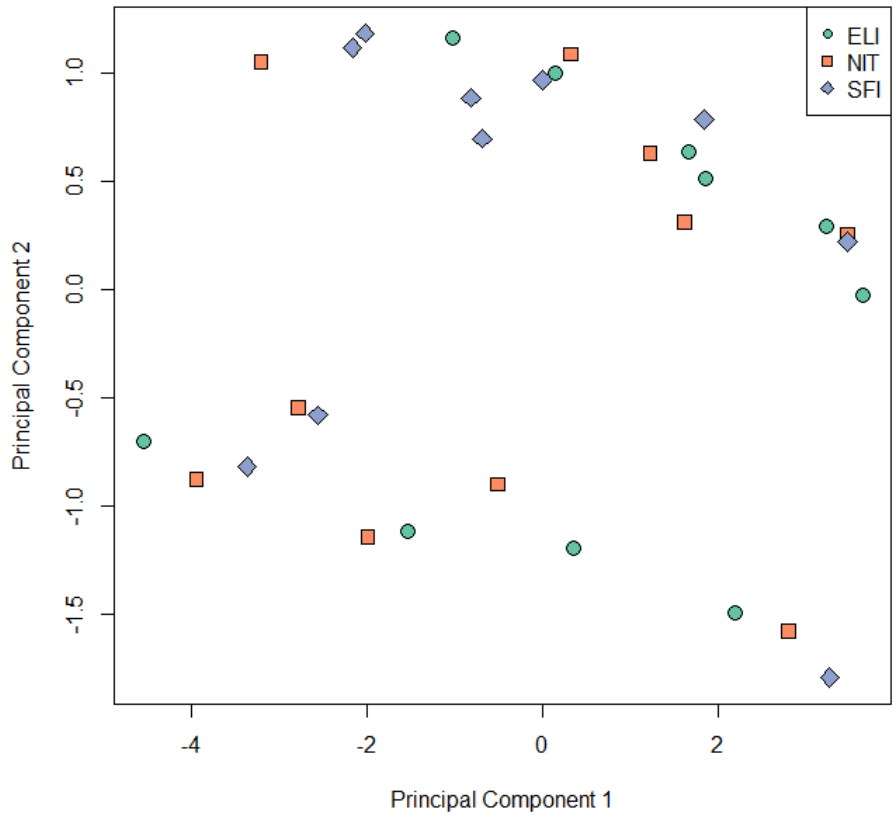

Figura 4. Escalado multidimensional.

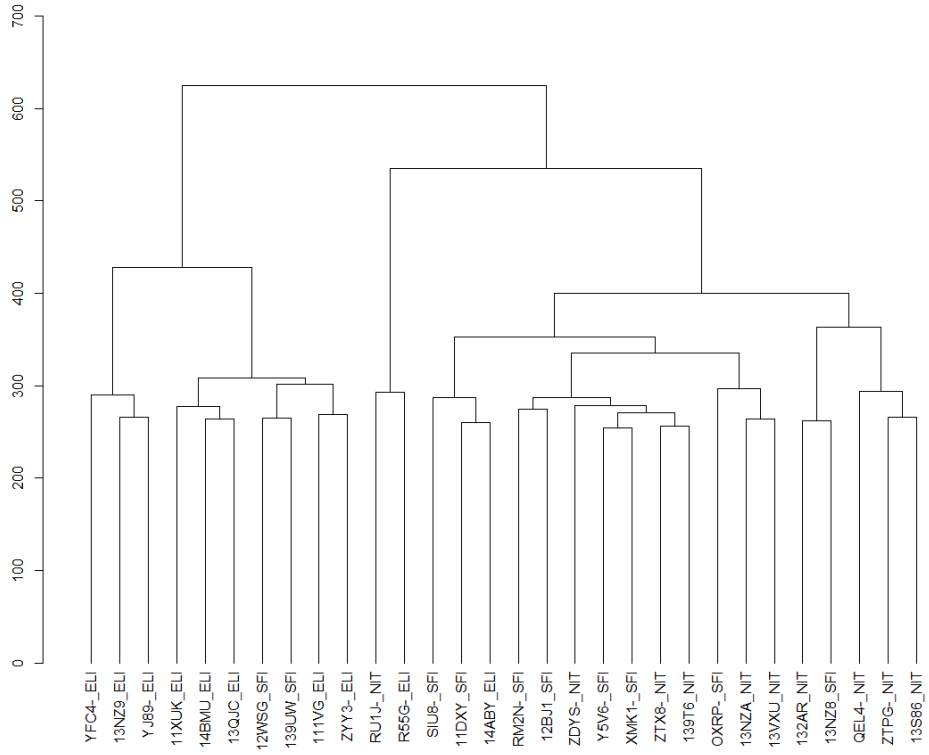

Figura 5. Dendrograma de las muestras.

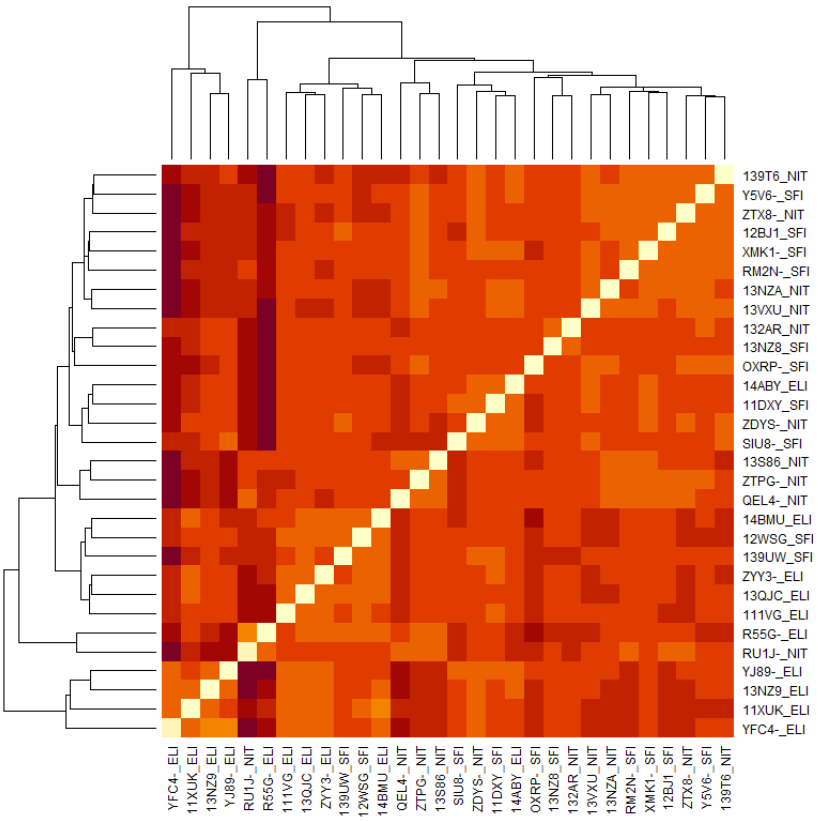

Figura 6. Mapa de color de las muestras.

Tanto el dendrograma como el mapa de colores presentan la organización de las muestras según un determinado criterio que tiene en cuenta las distancias. Los resultados no son iguales: se puede comprobar que el dendrograma de la Figura 5 no es el mismo que los que aparecen en la Figura 6. En ambos casos se observa que algunas distancias entre muestras de diferentes grupos son más cortas que las que hay entre las muestras de un mismo grupo.

Filtrado y normalización de los datos

Filtrado

Los genes con cuentas muy bajas son poco útiles para un estudio de expresión diferencial. Un primer filtrado se puede realizar teniendo en cuenta este hecho: se eliminan los genes que no alcanzan un mínimo nivel. Aunque el valor umbral lo fija el usuario, se suele aceptar que no son necesarios para el estudio los genes que no tengan más de 5-10 cuentas por millón. En general, el filtrado se suele realizar usando el concepto CPM (count-per-millon).

La librería *edgeR* dispone de varias opciones para realizar un filtrado. Una de estas opciones es `filterByExpr`, que mantiene la filas (genes) con un mínimo número de genes y realiza el filtrado independientemente del grupo al que pertenece la muestra. Aquí se aplica con las opciones por defecto.

El resultado es el siguiente:

```
Dimensiones antes del filtrado

[1] 46387    30

Dimensiones después del filtrado

[1] 22185    30
```

El número de tránscritos se había reducido de 56202 a 46387 al eliminar las filas con todos sus valores a cero; con el filtrado el tamaño (número de filas) se ha reducido a 22185.

Normalización

El proceso de normalización se aplica para identificar y eliminar diferencias técnicas entre muestras, y asegurar que los sesgos técnicos tienen un mínimo impacto sobre los resultados. Se han propuesto varios métodos de normalización, y aunque no existe consenso sobre cuál es el mejor, todos se basan en el mismo principio: multiplicar las cuentas de una muestra por un mismo factor (diferente entre muestras) de forma que el resultado final sea un número del mismo orden en todas las muestras.

La librería *edgeR* dispone de la función `calcNormFactors`, que calcula los factores de escalado que minimizan las diferencias entre muestras. La función se puede aplicar utilizando distintos métodos; aquí se aplica el `TMM` (*trimmed mean of M-values*).

	group	lib.size	norm.factors
QEL4-_NIT	NIT	38620898	1.0579607
132AR_NIT	NIT	70883275	0.9777699
ZDYS-_NIT	NIT	60169331	1.0058236
ZTPG-_NIT	NIT	49770457	0.9384294
ZTX8-_NIT	NIT	57162862	1.0008164

Se presentan algunos resultados gráficos con datos filtrados y normalizados.

La Figura 7 muestra el diagrama de cajas que resulta una vez los datos han sido filtrados y después de aplicar la transformación logarítmica comentada en la sección anterior. Se puede comprobar que hay diferencias tanto en la forma como en los valores del nuevo gráfico con respecto al diagrama anterior mostrado en la Figura 2.

Cuando se ajusta un modelo binomial, la variabilidad dentro de un grupo se representa con la dispersión. El hecho de tener datos de varias muestras con los mismos genes puede ayudar en la inferencia de cada gen en particular; por ejemplo, asumiendo que la dispersión es la misma en todos los genes. Una extensión de esta hipótesis sería asumir una tendencia media en la varianza, que también se podría suponer la misma para todos los genes. Sin embargo, los niveles de expresión entre genes no son iguales por lo que se hace necesario suponer una dispersión específica para cada gen.

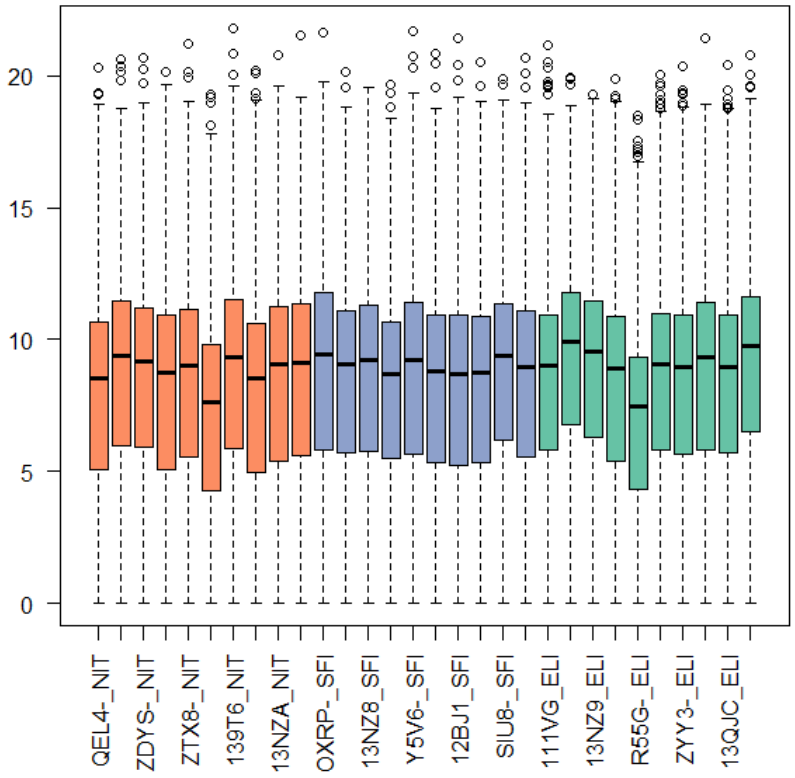

Figura 7. Diagramas de cajas de los datos filtrados.

La Figura 8 presenta un gráfico tipo BCV (*genewise biological coefficient variation*) de los datos filtrados. El valor BCV es la raíz cuadrada de la dispersión binomial negativa. Este tipo de gráfico presenta la variación del BCV frente a la abundancia de genes (expresada en el logaritmo de las cuentas por millón), y presenta una estimación de la dispersión de cada gen con la relación de dispersión media. Para poder obtener el gráfico tipo BCV es necesario aplicar la función `estimateCommonDisp`, que maximiza la verosimilitud común condicional binomial para estimar la dispersión en todos los genes, y después la función `estimateTagwiseDisp`, que estima los valores de dispersión mediante un método empírico de Bayes basado en una verosimilitud máxima condicional ponderada [6].

```
Disp = 0.23596 , BCV = 0.4858

[1] "DGEList"
attr(,"package")
[1] "edgeR"
```

El filtrado solo afecta al objeto `counts`, mientras que la normalización afecta solo al objeto `samples`; puesto que la normalización se realiza con los datos ya filtrados, el gráfico BCV presenta los dos efectos.

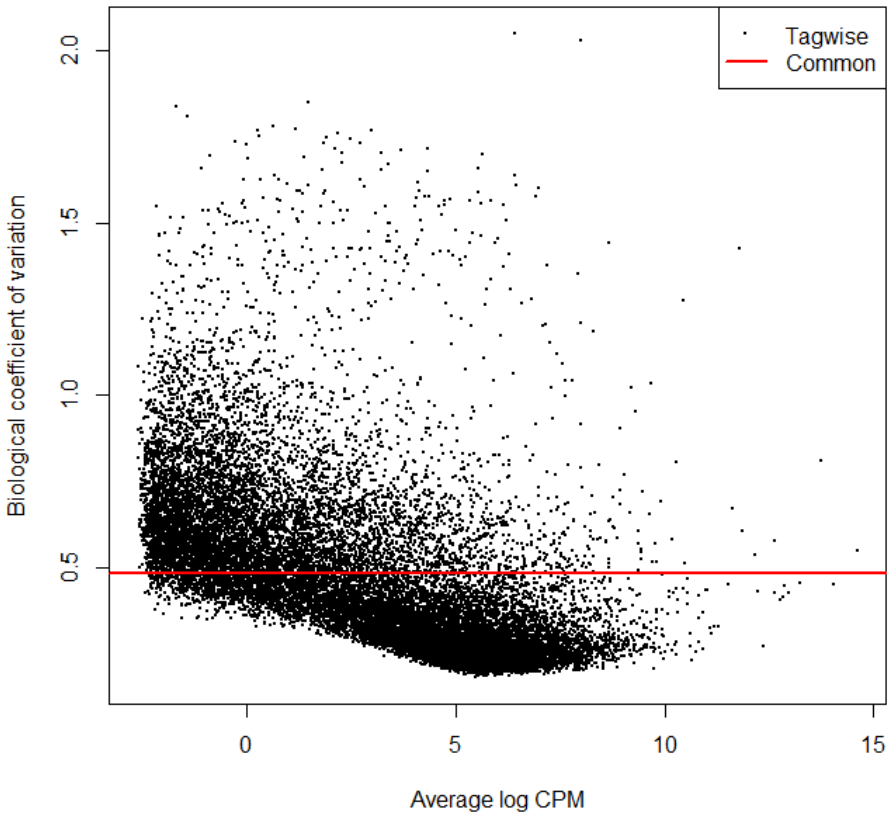

Figura 8. Gráfico BCV de los datos filtrados y normalizados.

Expresión diferencial

Las prestaciones de `edgeR` permiten realizar la identificación de genes diferencialmente expresados de varias formas. En este trabajo se aplicarán tres.

Identificación de genes diferencialmente expresados

Obtener los genes expresados diferencialmente es inmediato una vez que se han ajustado los modelos binomiales de cada gen y se han estimado las dispersiones. La librería `edgeR` permite contrastar dos condiciones con la función `exactTest`, que calcula, entre otros, los valores p o las diferencias entre los valores medios de dos grupos de cuentas distribuidas binomialmente.

A continuación, se presentan los resultados de las tres comparaciones (NIT-ELI, SIF-NIT y ELI-SIF). Para visualizar los resultados se dispone de varias opciones: gráficos MA, gráficos tipo volcán, o dendrogramas.

Un gráfico MA presenta una visión general de la comparación entre dos grupos. Cada gen se presenta con un punto cuyas coordenadas son el promedio del logaritmo de cuentas por millón (CPM) y el logaritmo base 2 del FC (fold change). Los genes con un valor p por debajo de un determinado umbral (threshold, cutoff) se suelen colorear: la función `smear` selecciona aquellos que tienen una concentración estimada mínima en uno de los grupos, especialmente los que tienen una cuenta igual a cero. Las Figuras 9, 10 y 11 muestran los gráficos MA,

obtenidos con la función `plotSmear`, que resultan de las tres comparaciones (NIT-ELI, SIF-NIT, ELI-SIF).

```
Comparación NIT-ELI
```

	NIT-ELI
Down	2469
NotSig	19004
Up	712

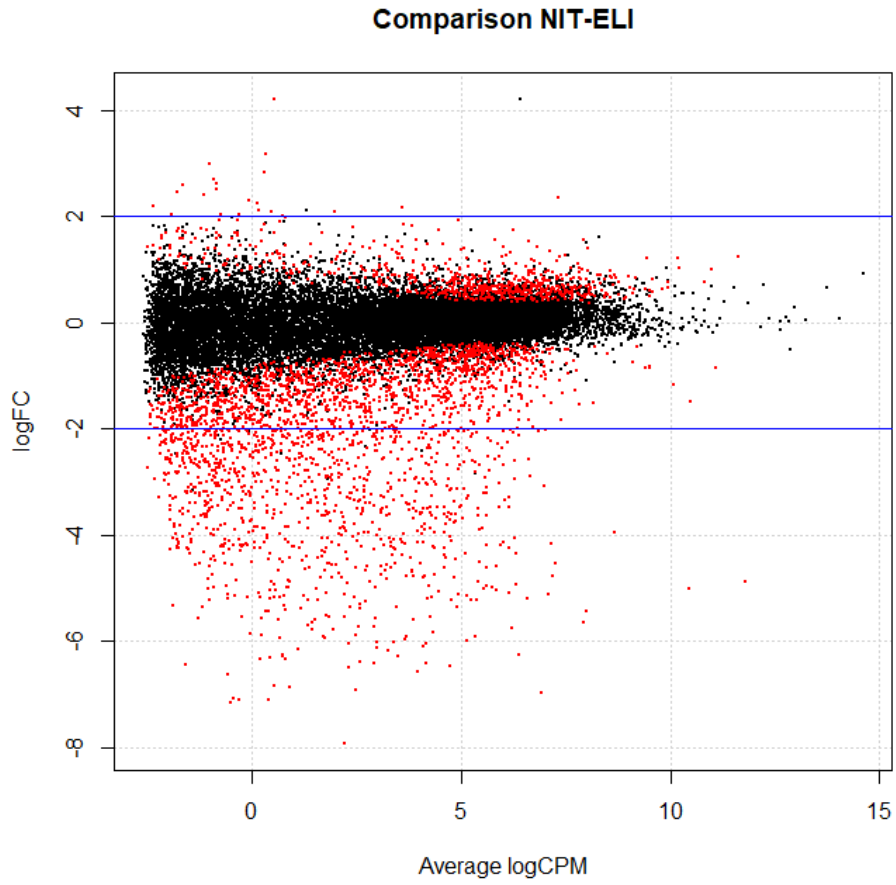

Figura 9. Gráfico MA - Comparación NIT-ELI.

```
Comparación SIF-NIT
```

	SIF-NIT
Down	6
NotSig	22139
Up	40

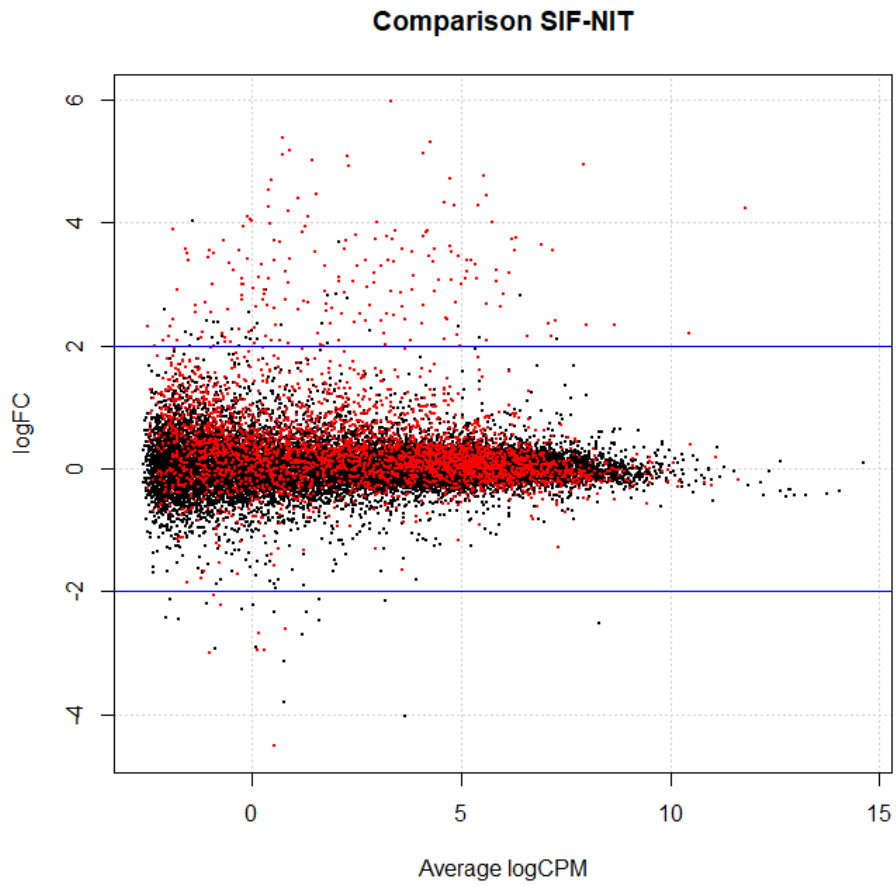

Figura 10. Gráfico MA - Comparación SIF-NIT.

```
Comparación ELI-SIF
```

	ELI-SIF
Down	505
NotSig	19965
Up	1715

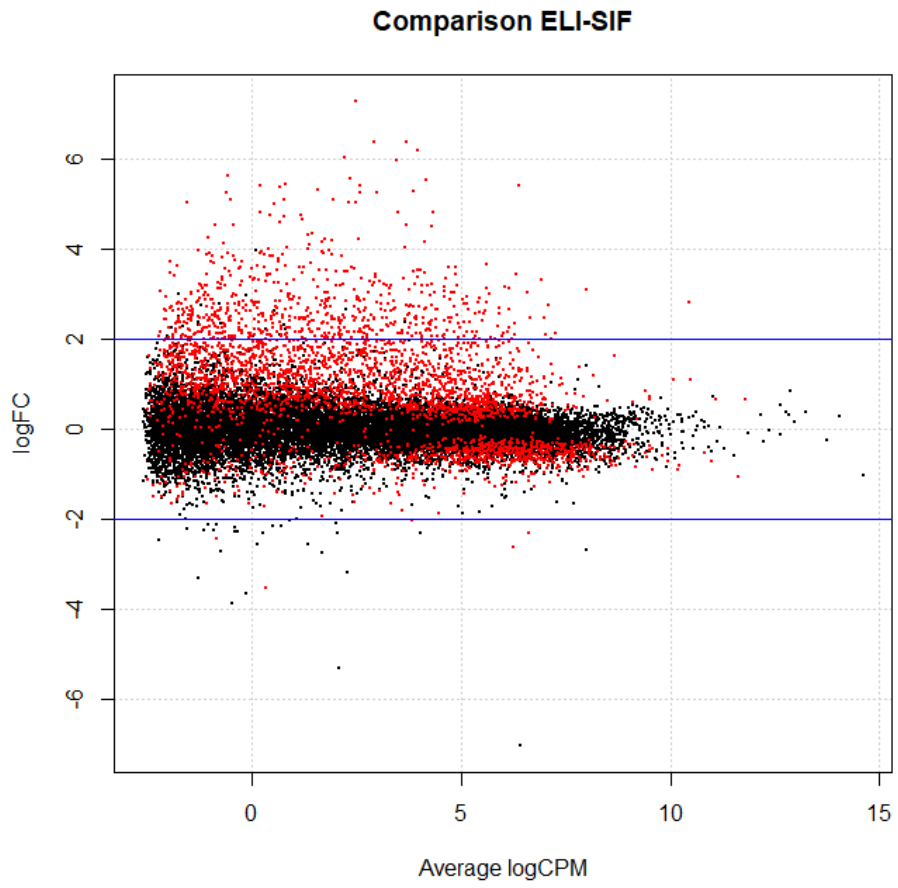

Figura 11. Gráfico MA - Comparación ELI-SIF.

Un resultado más detallado de una comparación entre dos grupos se puede obtener con la opción `topTags`, que presenta produce la siguiente información en la que los genes van ordenados de menor a mayor valor *p*:

```
Comparison of groups:  NIT-ELI
                        RefENSMBL entrezIDs     logFC   logCPM       PValue
ENSG00000247982.2  ENSG00000247982    283663 -3.615155 4.874458 2.188263e-21
ENSG00000205744.5  ENSG00000205744     79958 -2.770081 4.935932 4.106224e-21
ENSG00000068831.14 ENSG00000068831     10235 -3.343204 6.045012 6.010669e-21
ENSG00000069493.10 ENSG00000069493     29121 -2.933036 4.279504 1.946473e-20
ENSG00000104894.7  ENSG00000104894       951 -4.193925 5.595951 1.786147e-19
                             FDR
ENSG00000247982.2  4.444889e-17
ENSG00000205744.5  4.444889e-17
ENSG00000068831.14 4.444889e-17
ENSG00000069493.10 1.079563e-16
ENSG00000104894.7  7.925134e-16
```

El significado de las columnas es el siguiente:

- logFC: the logarithm (to basis 2) of the fold change;

- logCPM: the average log2 counts-per-million (CPM) for each tag;
- PValue: p value for the statistical significance of this change;
- FDR: p value adjusted for multiple testing with the Benjamini-Hochberg procedure, which controls false discovery rate (FDR).

Un *volcano plot* es una manera efectiva de presentar un test estadístico. Se trata de un gráfico de dispersión con el log2 de la diferencia de medias (el numerador del test t o *fold change*) en el eje de las x y los valores p en el eje de las y. El gráfico termina adoptando la forma de volcán si se transforman los valores p aplicando la función -log10(p). Para destacar genes, se añaden dos rectas verticales y una recta horizontal. Teniendo en cuenta lo que se presenta en cada eje, si se quieren destacar los genes con un valor p por debajo de 0.01 se coloca la recta horizontal en 2: los genes destacados son lo que quedan por encima de la recta. El log2 del FC (*fold change*) se usa para mostrar los genes equidistantes del centro que quedan arriba (*up*) y abajo (*down*).

Las Figuras 12, 13 y 14 muestran los gráficos resultantes para las tres comparaciones en las que se destacan (en color rojo) los genes con un valor p inferior a 0.05 y un valor absoluto del logaritmo base 2 del FC superior a 2.

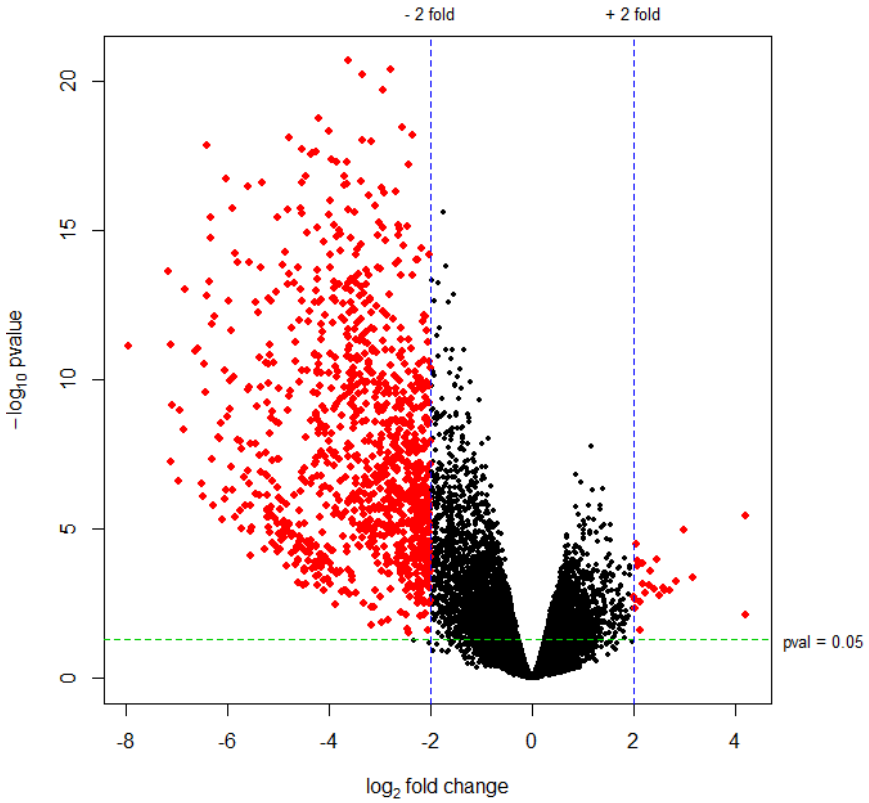

Figura 12. Volcano plot - Comparación NIT-ELI.

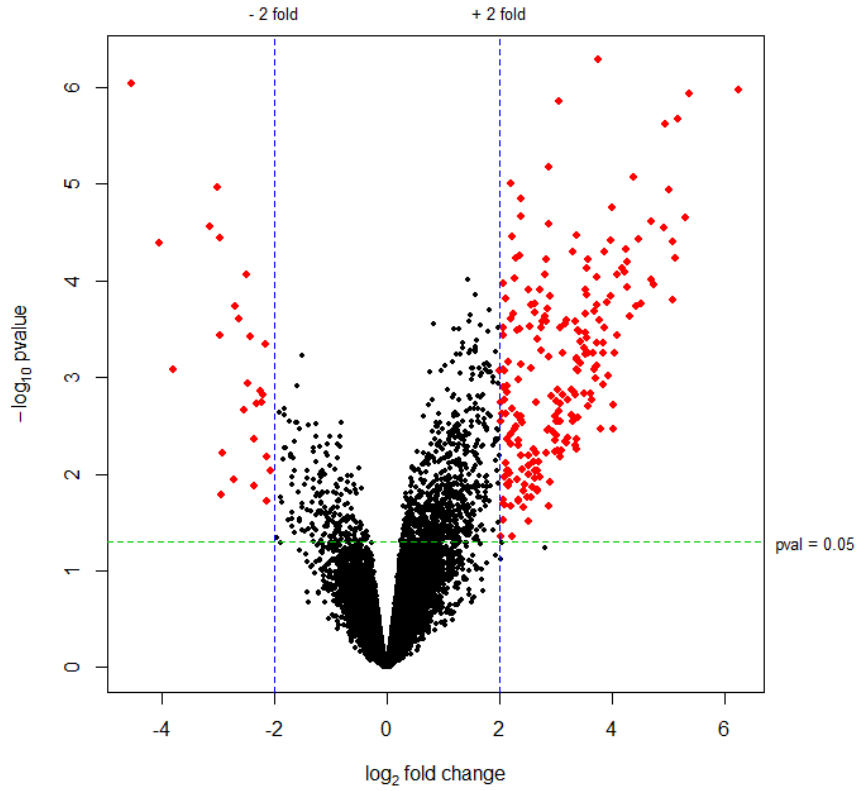

Figura 13. Volcano plot - Comparación SIF-NIT.

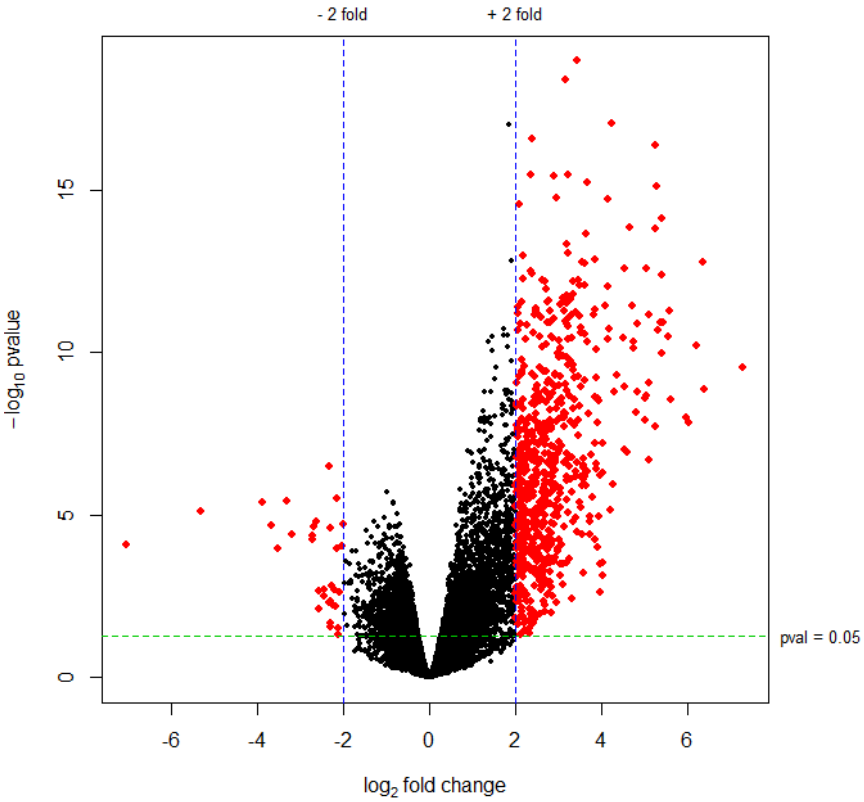

Figura 14. Volcano plot - Comparación ELI-SIF.

Aplicación de un modelo lineal generalizado (GLM)

El primer paso en la aplicación de esta opción será crear la matriz de diseño para poder obtener toda la información experimental deseada. En el presente estudio se trata de comparar los tres tipos de infiltración medidos: NIT, SFI, y ELI. La matriz se puede generar teniendo en cuenta los tres factores y utilizando la función `model.matrix`.

```
Matriz de diseño
```

	GELI	GNIT	GSIF
QEL4-_NIT	0	1	0
132AR_NIT	0	1	0
ZDYS-_NIT	0	1	0
ZTPG-_NIT	0	1	0
ZTX8-_NIT	0	1	0
RU1J-_NIT	0	1	0
139T6_NIT	0	1	0
13S86_NIT	0	1	0
13NZA_NIT	0	1	0
13VXU_NIT	0	1	0
OXRP-_SFI	0	0	1
11DXY_SFI	0	0	1
13NZ8_SFI	0	0	1
12WSG_SFI	0	0	1
Y5V6-_SFI	0	0	1
RM2N-_SFI	0	0	1
12BJ1_SFI	0	0	1
139UW_SFI	0	0	1
SIU8-_SFI	0	0	1
XMK1-_SFI	0	0	1
111VG_ELI	1	0	0
YFC4-_ELI	1	0	0
13NZ9_ELI	1	0	0
14BMU_ELI	1	0	0
R55G-_ELI	1	0	0
11XUK_ELI	1	0	0
ZYY3-_ELI	1	0	0
14ABY_ELI	1	0	0
13QJC_ELI	1	0	0

```
YJ89-_ELI       1       0       0
```

El ajuste de un modelo lineal con *edgeR* requiere de varios pasos: ajustar la dispersión común, la tendencia (*trend*) de la dispersión, y la dispersión particular de cada muestra (*tagwise dispersion*). La dispersión de un gen es utilizada para representar la varianza en los valores de las cuentas y se mide con el cuadrado del coeficiente de variación biológica.

Para aplicar este procedimiento con la librería `edgeR` se utiliza la función `estimateDisp` basada en una estrategia empírica de verosimilitud ponderada.

La información creada en el nuevo objeto está organizada de la siguiente forma:

```
Objetos creados con la función estimateGLMCommonDisp

 [1] "counts"             "samples"            "genes"
 [4] "common.dispersion"  "pseudo.counts"      "pseudo.lib.size"
 [7] "AveLogCPM"          "prior.df"           "prior.n"
[10] "tagwise.dispersion" "span"
```

Se presentan

- la dispersión común

```
Dispersión común

[1] 0.233578
```

- los valores mínimo y máximo de la dispersión

```
Valores mínimo y máximo de la dispersión común
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0331264	0.0770956	0.1634847	0.2644558	0.3317239	4.204681

El valor de dispersión común es 0.23, mientras que los valores mínimo y máximo son, respectivamente, 0.03 y 4.20.

A continuación, se aplican las funciones `estimateGLMTrendedDisp` (con las opciones por defecto) y `estimateGLMTagwiseDisp`.

El resultado del proceso completo se resume en la Figura 15 que muestra un gráfico BCV. Si se compara con el de la Figura 8 se puede observar que ahora aparece la tendencia de la variación. En general, la tendencia de la dispersión es la de decrecer con un incremento de las cuentas en los genes.

Finalmente, antes de proceder a realizar los contrastes se obtiene un modelo lineal general con la función `glmFit`.

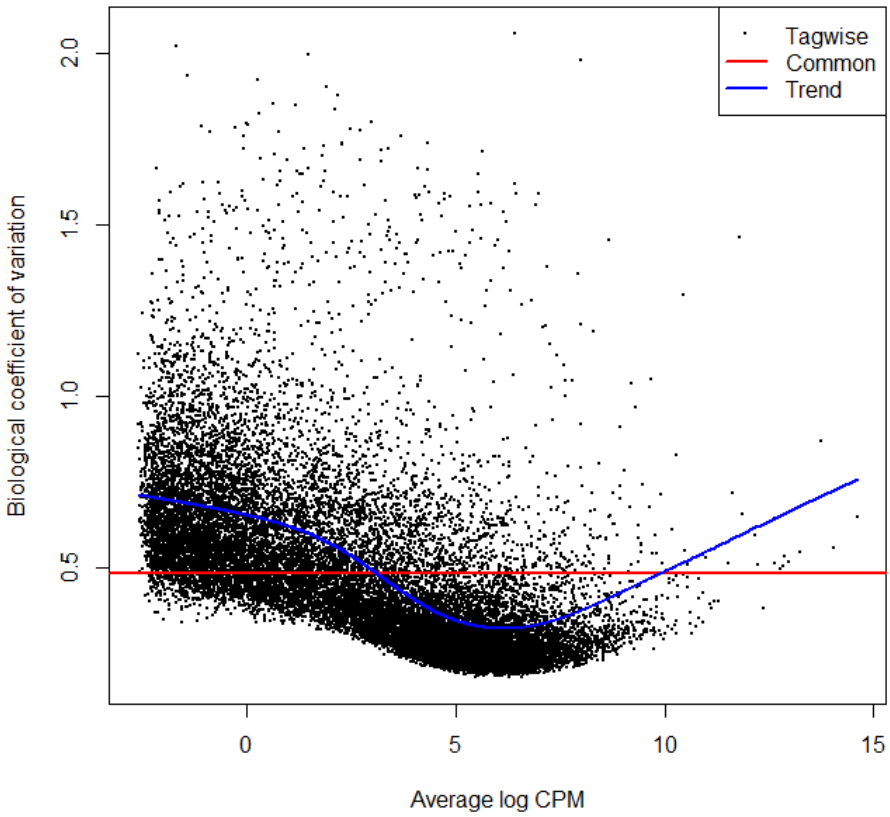

Figura 15. Gráfico BCV - modelo lineal generalizado (GLM).

La lista de objetos creados con este modelo es la siguiente:

```
Objetos del modelo lineal generalizado (GLM)

 [1] "coefficients"          "fitted.values"         "deviance"
 [4] "method"                "counts"                "unshrunk.coefficients"
 [7] "df.residual"           "design"                "offset"
[10] "dispersion"            "prior.count"           "samples"
[13] "genes"                 "prior.df"              "AveLogCPM"
```

Se realizan los contrastes utilizando la función `glmLRT`. El resultado es clasificado (up, down, not significant) con la función `decideTestsDGE`. Por otro lado, este resultado se puede presentar de forma gráfica utilizando de nuevo la función `plotSmear`.

Se presentan los resultados de las tres comparaciones y el gráfico correspondiente a la primera comparación.

```
Coefficient:  -1*GELI 1*GNIT
                         RefENSMBL entrezIDs     logFC   logCPM       LR
ENSG00000247982.2  ENSG00000247982    283663 -3.615172 4.874449 92.04949
ENSG00000205744.5  ENSG00000205744     79958 -2.770079 4.935931 90.20475
ENSG00000068831.14 ENSG00000068831     10235 -3.343206 6.045008 89.74622
ENSG00000069493.10 ENSG00000069493     29121 -2.933049 4.279499 87.07762
ENSG00000104894.7  ENSG00000104894       951 -4.193930 5.595947 82.16668
```

```
                         PValue          FDR
ENSG00000247982.2  8.453581e-22 1.875427e-17
ENSG00000205744.5  2.147455e-21 2.002230e-17
ENSG00000068831.14 2.707545e-21 2.002230e-17
ENSG00000069493.10 1.043444e-20 5.787199e-17
ENSG00000104894.7  1.250810e-19 5.549846e-16
```

	-1*GELI 1*GNIT
Down	2497
NotSig	18947
Up	741

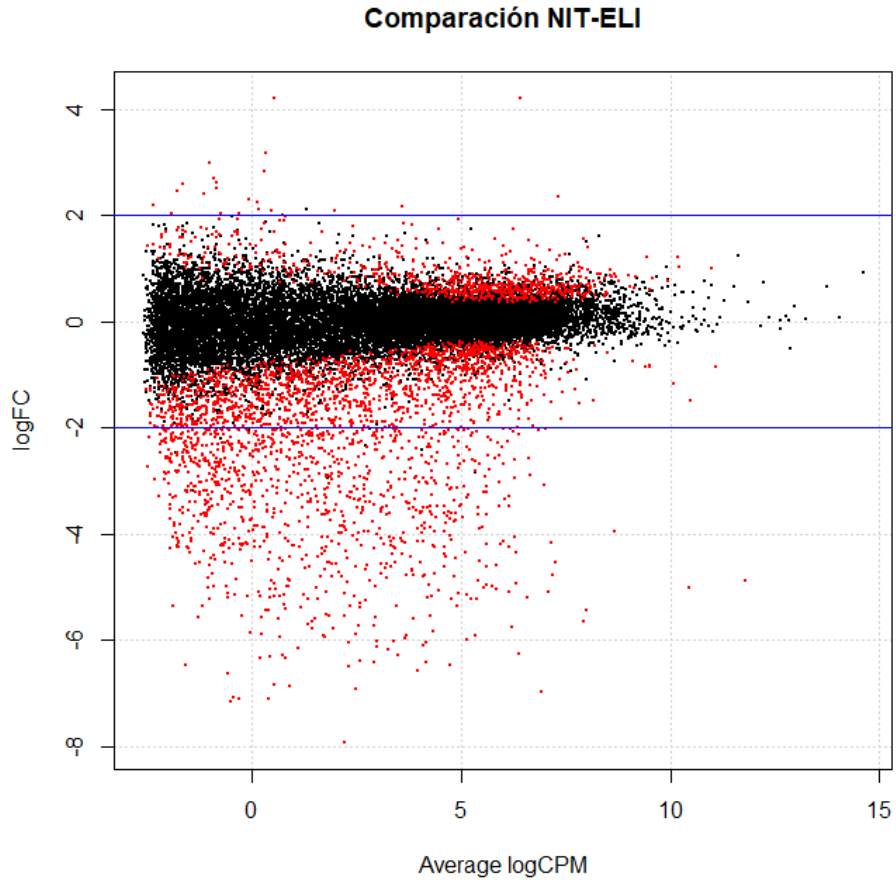

Figura 16. Gráfico MA - Comparación NIT-ELI.

```
Coefficient:  -1*GNIT 1*GSIF
                        RefENSMBL entrezIDs     logFC    logCPM       LR
ENSG00000230006.3  ENSG00000230006    645784  3.768073 4.0967374 25.71194
ENSG00000259536.1  ENSG00000259536      <NA> -4.518121 0.5212098 24.80882
ENSG00000253998.2  ENSG00000253998      <NA>  5.381362 0.7388529 24.17572
ENSG00000241294.1  ENSG00000241294      <NA>  6.268500 3.3316791 24.17287
ENSG00000088340.11 ENSG00000088340     80307  3.076271 4.7684863 23.72528
                         PValue         FDR
ENSG00000230006.3  3.963664e-07 0.004884262
```

```
ENSG00000259536.1  6.330713e-07 0.004884262
ENSG00000253998.2  8.793372e-07 0.004884262
ENSG00000241294.1  8.806422e-07 0.004884262
ENSG00000088340.11 1.111132e-06 0.004930092
```

	-1*GNIT 1*GSIF
Down	6
NotSig	22125
Up	54

```
Coefficient:  1*GELI -1*GSIF
                        RefENSMBL entrezIDs    logFC   logCPM       LR
ENSG00000247982.2  ENSG00000247982    283663 3.426912 4.874449 84.20921
ENSG00000068831.14 ENSG00000068831     10235 3.155403 6.045008 81.31540
ENSG00000111679.12 ENSG00000111679      5777 1.838655 5.902308 74.20609
ENSG00000175857.4  ENSG00000175857    202309 4.228112 1.719548 73.89576
ENSG00000175463.7  ENSG00000175463    374403 2.369243 5.188073 72.10976
                          PValue          FDR
ENSG00000247982.2  4.450913e-20 9.874351e-16
ENSG00000068831.14 1.924217e-19 2.134438e-15
ENSG00000111679.12 7.037289e-18 4.567510e-14
ENSG00000175857.4  8.235311e-18 4.567510e-14
ENSG00000175463.7  2.035542e-17 9.031698e-14
```

	1*GELI -1*GSIF
Down	534
NotSig	19909
Up	1742

Aunque los resultados no son los mismos que se obtuvieron con el método anterior, las cantidades son muy similares para los tres contrastes.

Aplicación de la matriz de contrastes

Si se desea que las comparaciones sean las mismas que se han realizado con los dos métodos anteriores, la matriz de contrastes debería ser la siguiente:

```
Matriz de contrastes
```

	NE	NS	SE
GELI	-1	0	-1
GNIT	1	1	0
GSIF	0	-1	1

El procedimiento implantado aquí empieza con un filtrado mediante la función `filterByExpr`, y continua con la aplicación de la función `voom`. Esta función calcula el logaritmo base 2 de las cuentas por millon (logCPM), estima la relación media-varianza y usa estos resultados para calcular los pesos (*weigths*) adecuados. Los datos obtenidos se ajustan mediante un modelo lineal con la librería `limma` [19], [20], [21]: aplicación de la función `lmFit` seguida de la aplicación de `eBayes` con la matriz de contrastes definida anteriormente.

Inicialmente, se presenta el resultado obtenido con la función `voom`; al final, el obtenido con la función `eBAyes`.

```
[1] 22185
```

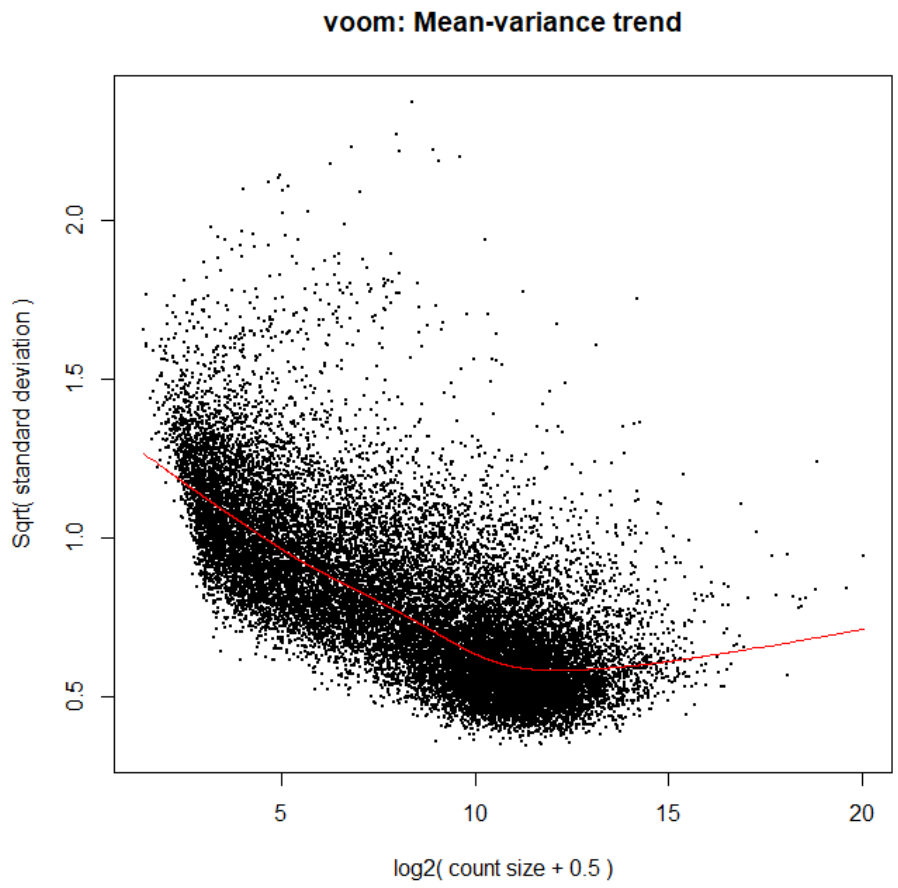

Figura 17. Resultado función voom.

	NE	NS	SE
Down	2895	0	2218
NotSig	18736	22185	19435
Up	554	0	532

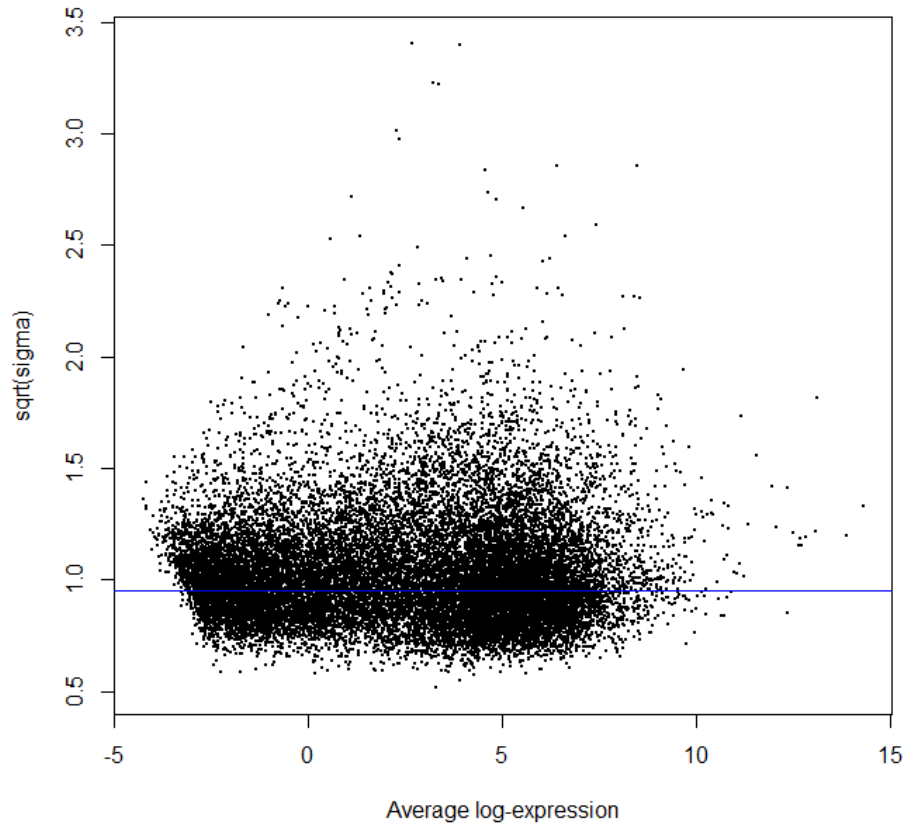

Figura 18. Modelo final - Tendencia media-varianza.

El resultado final con esta opción es el siguiente:

	NE	NS	SE
Down	2734	0	2058
NotSig	18991	22185	19722
Up	460	0	405

```
Contraste NIT-ELI
```

	RefENSMBL	entrezIDs	logFC	AveExpr	t	P.Value	adj.P.Val
ENSG00000253701.1	ENSG00000253701	NA	-6.717939	-2.861594	-8.746349	0	4.8e-06
ENSG00000255760.1	ENSG00000255760	105369723	-6.991255	-2.753587	-8.676958	0	4.8e-06
ENSG00000197549.5	ENSG00000197549	649179	-6.274792	-3.887990	-8.504340	0	4.8e-06
ENSG00000156738.13	ENSG00000156738	931	-7.420992	2.860954	-8.512021	0	4.8e-06
ENSG00000105492.11	ENSG00000105492	946	-5.279643	-3.831006	-8.445261	0	4.8e-06
ENSG00000026950.12	ENSG00000026950	11119	-2.369660	5.450454	-8.370007	0	4.8e-06

```
Contraste NIT-SIF
```

	RefENSMBL	entrezIDs	logFC	AveExpr	t	P.Value	adj.P.Val
ENSG00000260128.2	ENSG00000260128	100288380	-1.665716	-2.1179667	-3.714732	0.0006138	0.9996224
ENSG00000155011.4	ENSG00000155011	27123	-1.993939	-0.0276161	-3.600281	0.0008772	0.9996224
ENSG00000141682.11	ENSG00000141682	5366	-1.407069	0.2584565	-3.491810	0.0010973	0.9996224

ENSG00000236750.1	ENSG00000236750	NA	1.751329	-3.0787208	3.507375	0.0011046	0.9996224
ENSG00000234336.2	ENSG00000234336	NA	-2.103900	-2.6429525	-3.483860	0.0012217	0.9996224
ENSG00000106018.9	ENSG00000106018	7434	-1.944994	-1.4436247	-3.394416	0.0015374	0.9996224

```
Contraste SIF-ELI
```

	RefENSMBL	entrezIDs	logFC	AveExpr	t	P.Value	adj.P.Val
ENSG00000156738.13	ENSG00000156738	931	-6.317136	2.860954	-8.595334	0	1.52e-05
ENSG00000026950.12	ENSG00000026950	11119	-2.270357	5.450454	-8.164767	0	1.52e-05
ENSG00000197549.5	ENSG00000197549	649179	-6.017994	-3.887990	-8.153502	0	1.52e-05
ENSG00000175463.7	ENSG00000175463	374403	-2.385179	4.583275	-8.012376	0	1.52e-05
ENSG00000111679.12	ENSG00000111679	5777	-1.780226	5.565027	-7.898603	0	1.52e-05
ENSG00000167895.10	ENSG00000167895	147138	-3.925196	4.711413	-7.940925	0	1.52e-05

Se puede comprobar que en el segundo contraste todos los genes tienen su valor p en 1.

La Figura 19 muestra un gráfico tipo MD (log-FCs vs average log-CPM) del contraste NIT-ELI a partir de los resultados del último modelo lineal.

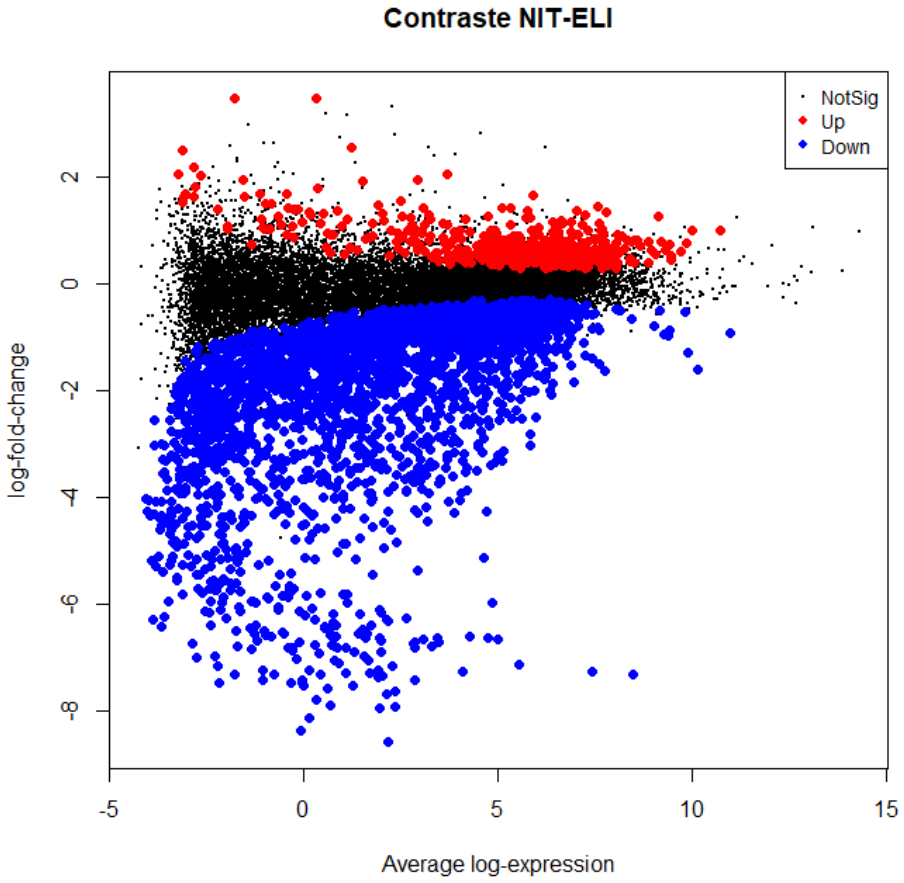

Figura 19. Modelo MD - Contraste NIT-ELI.

Los resultados obtenidos con las comparaciones realizadas se pueden resumir con un diagrama de Venn.

La Figura 20 muestra el número de genes diferencialmente expresados que se ha obtenido en cada comparación y los comunes a cada grupo, con el segundo método.

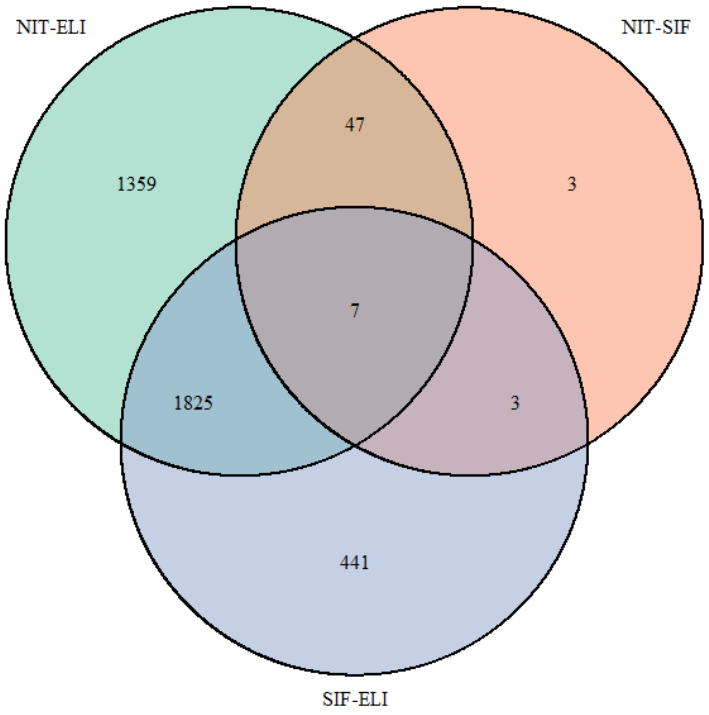

Figura 20. Diagrama de Venn con las tres comparaciones – Segundo método.

Con el filtrado y normalización realizadas, el número de genes es relativamente elevado en las comparaciones NIT-ELI y SIF-ELI, respecto al tercer grupo NIT-SIF. El número de genes comunes a dos grupos solo es elevado en las comparaciones (NIT-ELI)/(SIF-ELI). Finalmente, el número de genes involucrados en los tres grupos es 7.

La Figura 21 muestra el resultado con el tercer método. Se observa que, aunque los valores no son los mismos, la pauta en los valores es muy similar a la obtenida con los métodos anteriores.

Análisis de enriquecimiento biológico

El siguiente paso tiene como objetivo interpretar los resultados de expresión diferencial (obtenidos anteriormente) en un contexto biológico [22], [23]. El procedimiento más común utiliza una ontología genética disponible en bases como GO (Gene Ontology) o KEGG (Kyoto Encyclopedia of Genes and Genomes) en las que los genes se agrupan en términos o categorías con una propiedad biológica común. El estudio tiene como objetivo, a partir de los conjuntos de genes expresados bajo un cierto contraste, encontrar que términos GO/KEGG están sobre- o sub-representados; así, por ejemplo, los términos GO que aparecen frecuentemente en una lista de genes expresados diferencialmente se dicen que están sobre-representados o enriquecidos.

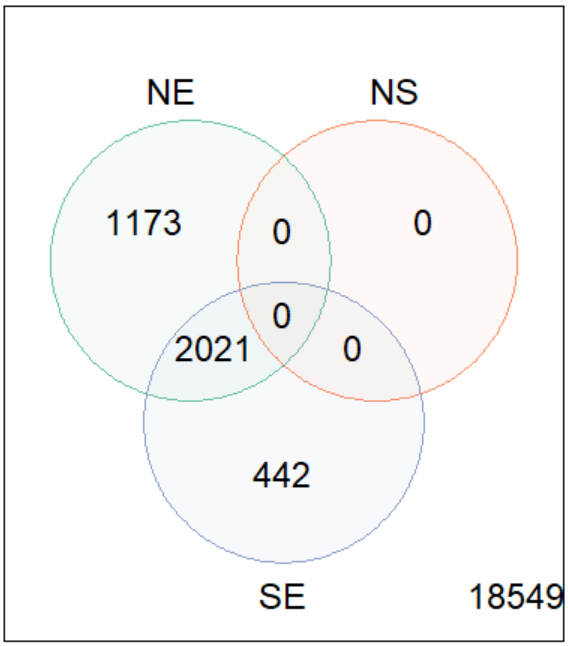

Figura 21. Diagrama de Venn con las tres comparaciones – Tercer método.

La librería `edgeR` tiene varias prestaciones que serán utilizadas para realizar este análisis [4], [19], [23]. La función `goana` extrae genes diferencialmente expresados de un objeto `tr` (obtenido con la función `glmQLFTest`) y realiza las pruebas necesarias para obtener los términos relacionadas de la base GO. La función usa las anotaciones NCBI RefSeq y los identificadores de genes *Entrez*. Por defecto, usa un valor umbral (cutoff) para una FDR del 5%. Los resultados se presentan en forma de tabla con la función `topGO` que los ordena en función de los valores *p*.

Se inicia el estudio obteniendo los términos GO para el contraste NIT-ELI. La selección se realiza para la especie *Homo Sapiens* - *Hs*.

	Term	Ont	N	DE	P.DE
GO:0005515	protein binding	MF	11807	5933	0
GO:0003674	molecular_functio…	MF	17697	7680	0
GO:0005488	binding	MF	15267	6986	0
GO:0050896	response to stimu…	BP	9324	4867	0
GO:0010033	response to organ…	BP	3302	2176	0
GO:0048518	positive regulati…	BP	6254	3491	0
GO:0070887	cellular response…	BP	3281	2117	0
GO:0042221	response to chemi…	BP	4797	2800	0
GO:0051716	cellular response…	BP	7617	4016	0
GO:0071310	cellular response…	BP	2707	1793	0

Las dimensiones del archivo GO son las siguientes:

```
[1] 22911     5
```

El significado de los resultados obtenidos con goana es el siguiente:

- Term: término descriptivo en la base GO.
- Ont: ontologia a la que pertenece cada término GO (BP es "biological process", CC es "cellular component", MF es molecular function).
- N: número de genes en el término GO.
- DE: número de genes en el conjunto de genes expresados diferencialmente.
- P.DE: valor *p* para sobre-representación del término GO en el conjunto.

Se comprueban los tipos de ontologías, se seleccionan los términos correspondientes a una de ellas (BP - *biological process*), y se averigua su número.

```
Tipos de ontologías

[1] "BP" "CC" "MF"

Número de términos GO-BP 16227
```

Se obtienen los términos de la ontología "BP" con menos valor *p* y con mayor número de genes.

	Term	Ont	N	DE	P.DE
GO:0050896	response to stimulus	BP	9324	4867	0
GO:0010033	response to organic substance	BP	3302	2176	0
GO:0048518	positive regulation of biological process	BP	6254	3491	0
GO:0070887	cellular response to chemical stimulus	BP	3281	2117	0
GO:0042221	response to chemical	BP	4797	2800	0
GO:0051716	cellular response to stimulus	BP	7617	4016	0

	Term	Ont	N	DE	P.DE
GO:0071704	organic substance metabolic process	BP	12093	5462	0
GO:0008152	metabolic process	BP	12552	5634	0
GO:0050789	regulation of biological process	BP	12093	5660	0
GO:0065007	biological regulation	BP	12777	5957	0
GO:0009987	cellular process	BP	16968	7282	0
GO:0008150	biological_process	BP	18670	7779	0

Se comprueba que el término con un *p* valor más pequeño es el GO:0050896 y el término con mayor número de genes es "GO:0008150".

Aunque en este trabajo solo se analizarán los resultados obtenidos de la base GO, se repite el proceso con la base KEGG y se muestran algunos resultados. Las dimensiones del archivo KEGG son las siguientes:

```
[1] 337   4
```

	Pathway	N	DE	P.DE
path:hsa05200	Pathways in cancer	531	435	0
path:hsa04080	Neuroactive ligand-rec…	340	278	0
path:hsa05169	Epstein-Barr virus inf…	201	176	0
path:hsa04010	MAPK signaling pathway	294	242	0
path:hsa04151	PI3K-Akt signaling pat…	354	282	0
path:hsa05163	Human cytomegalovirus …	225	188	0
path:hsa05166	Human T-cell leukemia …	219	181	0
path:hsa04024	cAMP signaling pathway	216	178	0
path:hsa05170	Human immunodeficiency…	212	175	0
path:hsa05205	Proteoglycans in cance…	205	170	0

La salida es similar a la que se obtiene con GO, y el significado de cada variable es el mismo.

Para realizar el siguiente paso se seleccionan algunos términos GO de la lista obtenida. La selección se puede llevar a cabo analizando el significado de cada término que aparece en la segunda columna de la tabla obtenida. Aquí se ha hecho buscando en la base GO términos relacionados con el término *tiroides*. Se ha obtenido una lista con los siguientes términos: "GO:2000823", "GO:0045893", "GO:0070324", "GO:0042403", "GO:0006590", "GO:0002154", "GO:0030375", "GO:0003713".

El siguiente paso será comprobar si los genes seleccionados están diferencialmente expresados con el contraste utilizado. Inicialmente, se crea una lista con todos los términos GO seleccionados anteriormente. Cada término será utilizado para definir un conjunto con los genes que han sido anotados para ese término: cada código GO se convierte en un vector (*Entrez Gene ID*) con los genes correspondientes a ese término.

GOID	TERM
GO:2000823	regulation of androgen receptor activity
GO:0045893	positive regulation of transcription, DNA-templated
GO:0070324	thyroid hormone binding
GO:0042403	thyroid hormone metabolic process
GO:0006590	thyroid hormone generation
GO:0002154	thyroid hormone mediated signaling pathway
GO:0030375	thyroid hormone receptor coactivator activity
GO:0003713	transcription coactivator activity

Los identificadores de genes se convierten en una lista de índices con la función `ids2indices`.

La librería `edgeR` dispone de varias prestaciones para analizar este tipo de listas. Se empieza con la función `roast`, en una versión adecuada para tests múltiples.

	NGenes	PropDown	PropUp	Direction	PValue	FDR	PValue.Mixed	FDR.Mixed
GO:0006590	17	0.0588235	0.6470588	Up	0.0063	0.05000	0.0014	0.0027000
GO:0042403	21	0.0952381	0.5714286	Up	0.0237	0.09460	0.0004	0.0009333
GO:2000823	4	0.0000000	0.5000000	Up	0.0715	0.14330	0.0563	0.0750000
GO:0070324	7	0.2857143	0.1428571	Down	0.0717	0.14330	0.0022	0.0034400
GO:0003713	292	0.1780822	0.2397260	Up	0.2433	0.38920	0.0001	0.0002000
GO:0002154	5	0.0000000	0.2000000	Up	0.4621	0.50355	0.5933	0.5933000
GO:0030375	4	0.2500000	0.0000000	Down	0.4669	0.50355	0.2303	0.2631429
GO:0045893	1320	0.2234848	0.2325758	Down	0.5036	0.50360	0.0001	0.0002000

El resultado es un archivo en el que cada fila corresponde al conjunto de genes relacionados con un término GO común. La información de cada columna es como sigue:

- NGenes: número de genes en cada término GO.
- PropDown and PropUp: proporción de genes en el término que están sub- y sobre-representados, respectivamente, con valores de *logfFC* más grandes que la raíz cuadrada de 2.
- Direction: dirección neta del cambio.
- PValue: evidencia sobre si la mayoría de genes en el término están diferencialmente expresados en la dirección especificada.
- PValue.Mixed: valores *p* válidos para cualquier dirección.
- FDRs (false discovery rates): valores calculados a partir de los correspondientes valores *p* en todos los conjuntos.

El mismo estudio se puede realizar con la función `fry`, que proporciona los siguientes resultados:

	NGenes	Direction	PValue	FDR	PValue.Mixed	FDR.Mixed
GO:0003713	292	Down	0	0	0.00e+00	0.00e+00
GO:0045893	1320	Down	0	0	0.00e+00	0.00e+00
GO:0030375	4	Down	0	0	1.92e-05	2.19e-05
GO:0070324	7	Down	0	0	0.00e+00	0.00e+00
GO:2000823	4	Down	0	0	2.27e-05	2.27e-05
GO:0002154	5	Down	0	0	1.20e-06	1.60e-06
GO:0042403	21	Down	0	0	0.00e+00	0.00e+00
GO:0006590	17	Down	0	0	0.00e+00	0.00e+00

Una tercera opción disponible en `edgeR` es `camera` que proporciona los siguientes resultados:

	NGenes	Direction	PValue	FDR
GO:0003713	292	Down	0.0000000	0.0000000
GO:0042403	21	Up	0.0001187	0.0004748
GO:0006590	17	Up	0.0008239	0.0021970
GO:0045893	1320	Down	0.0040947	0.0081894
GO:2000823	4	Down	0.0224412	0.0359060
GO:0002154	5	Down	0.0694777	0.0926370
GO:0030375	4	Down	0.1059418	0.1210763
GO:0070324	7	Up	0.4324753	0.4324753

Se comprueba que los resultados son distintos con cada opción.

Se repite el proceso con el contraste NIT-SIF. Se recuerda que los términos GO se ordenan de menor a mayor valor p. Los resultados obtenidos con la función `roast` serían los siguientes:

	NGenes	PropDown	PropUp	Direction	PValue	FDR	PValue.Mixed	FDR.Mixed
GO:0045893	1320	0.0666667	0.0393939	Down	0.3450	0.8052667	0.6500	0.9377714
GO:0070324	7	0.1428571	0.0000000	Down	0.3583	0.8052667	0.7998	0.9377714
GO:0002154	5	0.0000000	0.0000000	Down	0.4378	0.8052667	0.6360	0.9377714
GO:0030375	4	0.2500000	0.2500000	Down	0.5223	0.8052667	0.1369	0.9377714
GO:0006590	17	0.0000000	0.1176471	Up	0.5333	0.8052667	0.8206	0.9377714
GO:0042403	21	0.0000000	0.0952381	Up	0.6040	0.8052667	0.9393	0.9393000
GO:0003713	292	0.0582192	0.0308219	Down	0.8620	0.9119500	0.5661	0.9377714
GO:2000823	4	0.0000000	0.0000000	Up	0.9120	0.9120000	0.7175	0.9377714

Para visualizar los resultados de cada conjunto de genes se puede utilizar la opción `barcodeplot` que presenta los genes correspondientes al término GO seleccionado ordenados de izquierda a derecha con un valor creciente de logFC.

La Figura 22 muestra el resultado para el término GO:0006590, que aparece en el quinto lugar de la selección de términos GO realizada anteriormente. A partir de la figura, se comprueba que los genes de este término tienden a estar sobre-representados en el grupo NIT con respecto al grupo SIF.

La opción `barcodeplot` permite comparar dos términos. La Figura 23 muestra la comparación de GO:2000823 y GO:0045893. De esta figura se deduce que el primer conjunto de genes está sobre-representado en el grupo NIT sobre el grupo SIF, mientras que con el segundo grupo no se obtiene una tendencia clara, ya que las desviaciones respecto al valor neutral no son grandes.

Finalmente, la Figura 24 presenta la comparación de los códigos cuarto (GO:0006590) y quinto (GO:0030375), que se encuentran sobre- y sub-representados, respectivamente, según el contraste NIT-SIF. El segundo grupo de genes ya fue analizado por separado y muestra la misma tendencia que se vio en la Figura 22, mientras que con el primer grupo la tendencia es la misma.

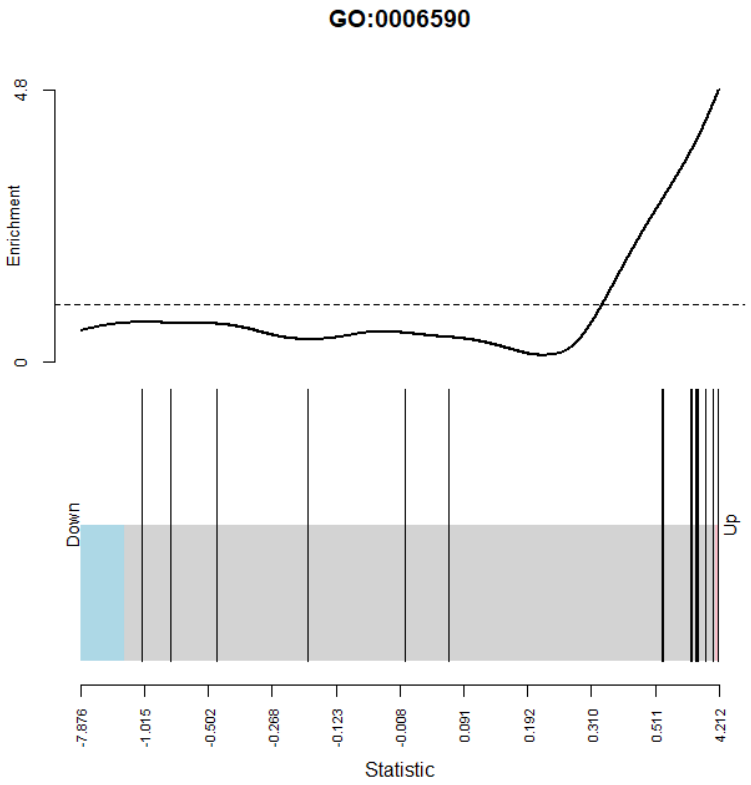

Figura 22. Gráfico código de barras para el término GO:0006590.

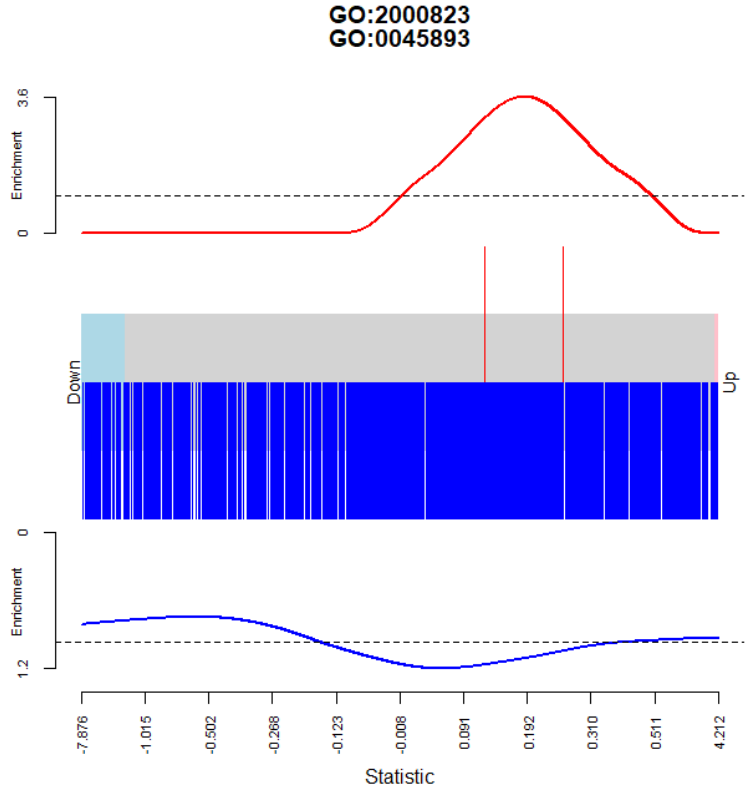

Figura 23. Gráfico código de barras para GO:2000823 y GO:0045893.

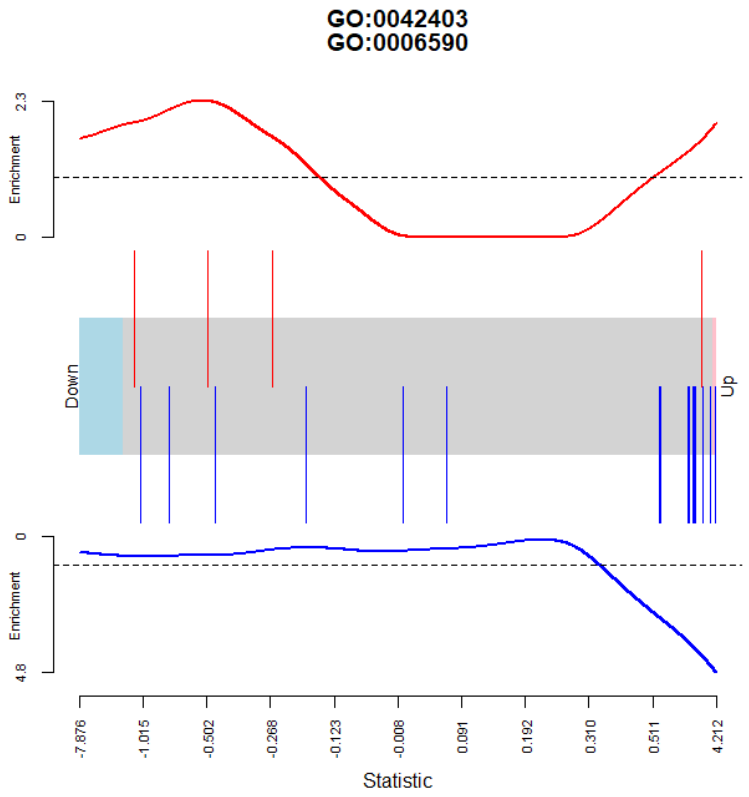

Figura 24. Gráfico código de barras para GO:0006590 y GO:0030375.

Análisis de significación biológica

Gene Set Enrichment Analysis (GSEA) es un método computacional que tiene como objetivo determinar si un conjunto de genes muestra diferencias estadísticamente significativas entre dos estados biológicos.

Esta parte del estudio se realizará con la librería `EGESEA` diseñada especialmente para trabajar con resultados de la librería `edgeR` [23], [24], [25], [26].

La instalación de `EGSEA` debe ir acompañada de la instalación de otra librería `EGSEAdata` que incluye varias colecciones (MSidDB, KEGG, GeneSetDB) de versiones de genes para *homo sapiens* y *mus musculus* (ratón).

La instalación de `EGSEAdata` para *homo sapiens* genera el siguiente mensaje:

```
The following databases are available in EGSEAdata for Homo sapiens:

Database name: KEGG Pathways
Version: NA
Download/update date: 07 March 2017
Data source: gage::kegg.gsets()
Supported species: human, mouse, rat
Gene set collections: Signaling, Metabolism, Disease
Related data objects: kegg.pathways
Number of gene sets in each collection for  Homo sapiens :
```

```
    Signaling: 132
    Metabolism: 89
    Disease: 71

Database name: Molecular Signatures Database (MSigDB)
Version: 5.2
Download/update date: 07 March 2017
Data source: http://software.broadinstitute.org/gsea
Supported species: human, mouse
Gene set collections: h, c1, c2, c3, c4, c5, c6, c7
Related data objects: msigdb, Mm.H, Mm.c2, Mm.c3, Mm.c4, Mm.c5, Mm.c6, Mm.c7
Number of gene sets in each collection for  Homo sapiens :
    h Hallmark Signatures: 50
    c1 Positional Gene Sets: 326
    c2 Curated Gene Sets: 3754
    c3 Motif Gene Sets: 836
    c4 Computational Gene Sets: 858
    c5 GO Gene Sets: 6166
    c6 Oncogenic Signatures: 144
    c7 Immunologic Signatures: 1888

Database name: GeneSetDB Database
Version: NA
Download/update date: 15 January 2016
Data source: http://www.genesetdb.auckland.ac.nz/
Supported species: human, mouse, rat
Gene set collections: gsdbdis, gsdbgo, gsdbdrug, gsdbpath, gsdbreg
Related data objects: gsetdb.human, gsetdb.mouse, gsetdb.rat
Number of gene sets in each collection for  Homo sapiens :
    GeneSetDB Drug/Chemical: 7032
    GeneSetDB Disease/Phenotype: 5425
    GeneSetDB Gene Ontology: 2431
    GeneSetDB Pathway: 1535
    GeneSetDB Gene Regulation: 215

Type ?<data object name> to get a specific information
                about it, e.g., ?kegg.pathways.
```

`EGSEA` es una potente herramienta para análisis de significación biológica que utiliza cálculo paralelo, manipula archivos de clases compatibles varias librerías, como `limma` y `edgeR`, y realiza operaciones disponibles en estas librerías [25], [26]. En este trabajo se ha realizado una selección que permita ilustrar el funcionamiento de esta librería.

El archivo de trabajo será el mismo que fue utilizado en el estudio de expresión diferencial; se trata por tanto de la colección de genes que se tenía después del filtrado. Por lo que respecta a las anotaciones se utilizará exclusivamente la colección *c5* (GO Gene Sets) de la base de datos *GeneSetDB*.

En el primer paso, antes de descargar anotaciones y realizar el análisis de significación biológica, se prepara el archivo de trabajo; concretamente, se modifica el objeto *genes* del que

solo se utilizará la columna de identificadores *Entrez* y a la que se añadirá una columna (simbólica) de símbolos que se basará en el número de fila.

```
Objeto genes
```

	entrezIDs	Symbol
ENSG00000223972.4	100287102	1
ENSG00000227232.4	NA	2
ENSG00000243485.2	NA	3
ENSG00000237613.2	645520	4
ENSG00000268020.2	NA	5
ENSG00000240361.1	NA	6

```
Objeto counts
```

	QEL4-_NIT	132AR_NIT	ZDYS-_NIT	ZTPG-_NIT	ZTX8-_NIT
ENSG00000223972.4	4	0	5	1	3
ENSG00000227232.4	511	907	637	524	586
ENSG00000243485.2	2	1	3	1	0
ENSG00000237613.2	2	1	4	0	2
ENSG00000268020.2	4	0	1	0	2

```
Objeto samples
```

	group	lib.size	norm.factors
QEL4-_NIT	NIT	38646199	1
132AR_NIT	NIT	70926853	1
ZDYS-_NIT	NIT	60210708	1
ZTPG-_NIT	NIT	49792267	1
ZTX8-_NIT	NIT	57198726	1
RU1J-_NIT	NIT	22156453	1

En el siguiente paso se eliminan todas las filas con identificador *Entrez* NA y las filas que tengan un código *Entrez* repetido.

```
Antes de eliminar NAs y duplicados

Dimensiones del objeto genes

[1] 46387     2

Dimensiones del objeto counts
```

```
[1] 46387    30

Después de eliminar NAs y duplicados

Dimensiones del objeto genes

[1] 22976     2

Dimensiones del objeto counts

[1] 22976    30
```

Se normaliza el objeto *samples* y se aplica la función voom.

El nuevo archivo tiene los siguientes objetos:

```
[1] "genes"   "targets" "E"       "weights" "design"
```

con el siguiente número final de genes:

```
Número final de genes = 17451
```

Tal como se ha dicho anteriormente, para realizar las anotaciones se escoge la opción *Molecular Signatures Database* (MSigDB) y la colección c5.

```
[1] "Loading MSigDB Gene Sets ... "
[1] "Loaded gene sets for the collection c5 ..."
[1] "Indexed the collection c5 ..."
[1] "Created annotation for the collection c5 ..."
```

El archivo de anotaciones tiene las siguientes características

```
$c5
An object of class "GSCollectionIndex"
    Number of gene sets: 6166
    Annotation columns: ID, GeneSet, BroadUrl, Description, PubMedID, NumGene
s, Contributor, Ontology, GOID
    Total number of indexing genes: 17451
    Species: Homo sapiens
    Collection name: c5 GO Gene Sets
    Collection uniqe label: c5
    Database version: 5.2
    Database update date: 07 March 2017
```

y contiene los siguientes objetos

```
[1] "original"   "idx"        "anno"       "featureIDs" "species"
[6] "name"       "label"      "version"    "date"
```

Por ejemplo, el objeto *featureIDs* contiene los identificadores *Entrez* que había en el archivo de trabajo.

```
 [1] "100996442" "105378580" "400728"    "79854"     "643837"    "100130417"
 [7] "148398"    "26155"     "339451"    "84069"     "84808"     "57801"
```

```
[13] "9636"      "375790"    "100288175" "54991"     "100506376" "254173"
[19] "8784"      "7293"
```

Se crea el mapa de símbolos que contiene los identificadores *Entrez* y los símbolos escogidos.

Finalmente, se escogen los métodos que se utilizarán en el análisis de significación biológica. Con `EGSEA` se pueden utilizar hasta 12 métodos distintos; aquí se utilizarán los siguientes:

```
 [1] "camera"     "safe"       "gage"       "padog"      "plage"
 [6] "zscore"     "gsva"       "ssgsea"     "globaltest" "ora"
```

La lista de ordenaciones disponibles en `EGSEA` incluye las siguientes opciones:

```
 [1] "p.value"       "p.adj"         "vote.rank"     "avg.rank"
 [5] "med.rank"      "min.pvalue"    "min.rank"      "avg.logfc"
 [9] "avg.logfc.dir" "direction"     "significance"  "camera"
[13] "roast"         "safe"          "gage"          "padog"
[17] "plage"         "zscore"        "gsva"          "ssgsea"
[21] "globaltest"    "ora"           "fry"
```

mientras que la lista de opciones para combinar valores *p* individuales incluye

```
[1] "fisher"    "wilkinson" "average"   "logitp"    "sump"      "sumz"
[7] "votep"     "median"
```

La opción utilizada por defecto es “wilkinson”.

La función disponible en `EGSEA` para realizar el análisis de significación biológica es `gsa`. En este trabajo se aplica esta función con el método de ordenación “med.rank” y la opción por defecto (“wilkinson”) para combinar valores. Por último, se especifica 3 para realizar los cálculos en paralelo; después de varias pruebas, este parece ser el número que termina seleccionando `gsa`.

```
##------ Fri Jun 12 15:11:42 2020 ------##
##------ Fri Jun 12 15:35:40 2020 ------##
```

La ejecución de esta función genera un archivo con las siguientes características:

```
> show(gsa)

An object of class "EGSEAResults"
    Total number of genes: 17451
    Total number of samples: 30
    Contrasts: NE, NS, SE
    Base GSE methods: camera (limma:3.42.2), safe (safe:3.26.0), gage (gage:2
.36.0), padog (PADOG:1.28.0), plage (GSVA:1.34.0), zscore (GSVA:1.34.0), gsva
(GSVA:1.34.0), ssgsea (GSVA:1.34.0), globaltest (globaltest:5.40.0), ora (sta
ts:3.6.3)
    P-values combining method: wilkinson
    Sorting statistic: med.rank
    Organism: Homo sapiens
    HTML report generated: No
```

```
    Tested gene set collections:
        c5 GO Gene Sets (c5): 6166 gene sets - Version: 5.2, Update date: 07
March 2017
    EGSEA version: 1.14.0
    EGSEAdata version: 1.14.0
Use summary(object) and topSets(object, ...) to explore this object.
```

y los siguientes objetos:

```
 [1] "results"           "limmaResults"      "contr.names"
 [4] "contrast"          "sampleSize"        "gs.annots"
 [7] "baseMethods"       "baseInfo"          "combineMethod"
[10] "sort.by"           "symbolsMap"        "logFC"
[13] "logFC.calculated"  "sum.plot.axis"     "sum.plot.cutoff"
[16] "report"            "report.dir"        "egsea.version"
[19] "egseaData.version"
```

Si se solicita un resumen, el programa presenta la información correspondiente a los 10 primeros conjuntos de genes:

```
**** Top 10 gene sets in the c5 GO Gene Sets collection ****
** Contrast NE **
GO_B_CELL_RECEPTOR_SIGNALING_PATHWAY | GO_T_CELL_RECEPTOR_COMPLEX
GO_REGULATION_OF_B_CELL_RECEPTOR_SIGNALING_PATHWAY | GO_POSITIVE_REGULATION_O
F_CELL_KILLING
GO_REGULATION_OF_LEUKOCYTE_MEDIATED_IMMUNITY | GO_REGULATION_OF_INTERFERON_GA
MMA_SECRETION
GO_REGULATION_OF_LYMPHOCYTE_MEDIATED_IMMUNITY | GO_POSITIVE_REGULATION_OF_CEL
L_CELL_ADHESION
GO_REGULATION_OF_CELL_KILLING | GO_REGULATION_OF_ANTIGEN_RECEPTOR_MEDIATED_SI
GNALING_PATHWAY

** Contrast NS **
GO_GROOMING_BEHAVIOR | GO_DIACYLGLYCEROL_KINASE_ACTIVITY
GO_REGULATION_OF_CELL_COMMUNICATION_BY_ELECTRICAL_COUPLING | GO_SARCOPLASMIC_
RETICULUM_MEMBRANE
GO_CAMP_CATABOLIC_PROCESS | GO_3_5_CYCLIC_AMP_PHOSPHODIESTERASE_ACTIVITY
GO_REGULATION_OF_RYANODINE_SENSITIVE_CALCIUM_RELEASE_CHANNEL_ACTIVITY | GO_NE
GATIVE_REGULATION_OF_CYTOSOLIC_CALCIUM_ION_CONCENTRATION
GO_CYCLIC_NUCLEOTIDE_CATABOLIC_PROCESS | GO_REGULATION_OF_CALCIUM_ION_TRANSME
MBRANE_TRANSPORT

** Contrast SE **
GO_POSITIVE_REGULATION_OF_CELL_CELL_ADHESION | GO_POSITIVE_REGULATION_OF_T_CE
LL_PROLIFERATION
GO_REGULATION_OF_LEUKOCYTE_MEDIATED_IMMUNITY | GO_REGULATION_OF_LYMPHOCYTE_ME
DIATED_IMMUNITY
GO_B_CELL_RECEPTOR_SIGNALING_PATHWAY | GO_T_CELL_RECEPTOR_COMPLEX
GO_POSITIVE_REGULATION_OF_IMMUNE_EFFECTOR_PROCESS | GO_POSITIVE_REGULATION_OF
_ALPHA_BETA_T_CELL_ACTIVATION
GO_REGULATION_OF_LYMPHOCYTE_DIFFERENTIATION | GO_REGULATION_OF_HOMOTYPIC_CELL
```

```
_CELL_ADHESION

** Comparison analysis **
GO_POSITIVE_REGULATION_OF_CELL_CELL_ADHESION | GO_REGULATION_OF_LEUKOCYTE_MED
IATED_IMMUNITY
GO_POSITIVE_REGULATION_OF_T_CELL_PROLIFERATION | GO_REGULATION_OF_LYMPHOCYTE_
MEDIATED_IMMUNITY
GO_B_CELL_RECEPTOR_SIGNALING_PATHWAY | GO_T_CELL_RECEPTOR_COMPLEX
GO_REGULATION_OF_HOMOTYPIC_CELL_CELL_ADHESION | GO_REGULATION_OF_LYMPHOCYTE_D
IFFERENTIATION
GO_REGULATION_OF_ALPHA_BETA_T_CELL_ACTIVATION | GO_POSITIVE_REGULATION_OF_IMM
UNE_EFFECTOR_PROCESS
```

Se puede comprobar que en la información de cada gen se incluye el contraste con el que ha sido obtenido y los términos GO correspondientes.

La información generada también se puede consultar con la opción `topSets` que mostrará el número de genes especificados de la colección seleccionada (aquí solo puede ser *c5*) y con el método de ordenación deseado (aquí se selecciona "ora").

El resultado es una extensa tabla con un número de columnas que depende de los métodos aplicados y cuyas filas coinciden con los términos GO prioritarios. Para poder analizar esta información se ha partido la tabla en secciones con las columnas consecutivas. La información corresponde al primer contraste NIT-ELI.

NOTA: No hay resultados correspondientes al contraste NIT-SIF (ver Figura 21).

	Rank	p.value	p.adj	vote.rank	avg.rank
GO_IMMUNE_SYSTEM_PROCESS	1	0	0	5	820.2
GO_IMMUNE_RESPONSE	2	0	0	5	761.5
GO_REGULATION_OF_IMMUNE_SYSTEM_PROCESS	3	0	0	5	847.1
GO_LEUKOCYTE_ACTIVATION	4	0	0	5	103.6
GO_REGULATION_OF_IMMUNE_RESPONSE	5	0	0	5	794.1
GO_LYMPHOCYTE_ACTIVATION	6	0	0	10	69.9
GO_POSITIVE_REGULATION_OF_IMMUNE_SYSTEM_PROCESS	7	0	0	10	824.1
GO_DEFENSE_RESPONSE	8	0	0	10	890.1
GO_REGULATION_OF_CELL_ACTIVATION	9	0	0	45	87.9
GO_CELL_ACTIVATION	10	0	0	110	740.3

	med.rank	min.pvalue	min.rank	avg.logfc
GO_IMMUNE_SYSTEM_PROCESS	309.0	0	1	1.945975
GO_IMMUNE_RESPONSE	222.5	0	2	2.191080
GO_REGULATION_OF_IMMUNE_SYSTEM_PROCESS	309.0	0	3	1.937206
GO_LEUKOCYTE_ACTIVATION	78.0	0	2	2.317512
GO_REGULATION_OF_IMMUNE_RESPONSE	260.5	0	4	2.007627
GO_LYMPHOCYTE_ACTIVATION	75.0	0	5	2.342952
GO_POSITIVE_REGULATION_OF_IMMUNE_SYSTEM_PROCESS	288.5	0	6	2.094694

GO_DEFENSE_RESPONSE	372.5	0	8	1.863689
GO_REGULATION_OF_CELL_ACTIVATION	68.0	0	4	2.178098
GO_CELL_ACTIVATION	205.0	0	10	2.181104

Si como opción de contraste se escoge “comparison” el resultado es el siguiente:

	Rank	p.value	p.adj	vote.rank	avg.rank
GO_POSITIVE_REGULATION_OF_CELL_CELL_ADHESION	1	0	0	20	819.4333
GO_REGULATION_OF_LEUKOCYTE_MEDIATED_IMMUNITY	2	0	0	35	845.1000
GO_POSITIVE_REGULATION_OF_T_CELL_PROLIFERATION	3	0	0	45	636.7333
GO_REGULATION_OF_LYMPHOCYTE_MEDIATED_IMMUNITY	4	0	0	5	627.1333
GO_B_CELL_RECEPTOR_SIGNALING_PATHWAY	5	0	0	5	424.6000
GO_T_CELL_RECEPTOR_COMPLEX	6	0	0	30	916.1000
GO_REGULATION_OF_HOMOTYPIC_CELL_CELL_ADHESION	7	0	0	15	555.0333
GO_REGULATION_OF_LYMPHOCYTE_DIFFERENTIATION	8	0	0	60	807.1333
GO_REGULATION_OF_ALPHA_BETA_T_CELL_ACTIVATION	9	0	0	65	726.1333
GO_POSITIVE_REGULATION_OF_IMMUNE_EFFECTOR_PROCESS	10	0	0	70	1029.7333

	med.rank	min.pvalue	min.rank	avg.logfc
GO_POSITIVE_REGULATION_OF_CELL_CELL_ADHESION	39.0	0	11	1.607440
GO_REGULATION_OF_LEUKOCYTE_MEDIATED_IMMUNITY	42.5	0	7	1.508118
GO_POSITIVE_REGULATION_OF_T_CELL_PROLIFERATION	42.5	0	3	1.671690
GO_REGULATION_OF_LYMPHOCYTE_MEDIATED_IMMUNITY	48.5	0	2	1.512958
GO_B_CELL_RECEPTOR_SIGNALING_PATHWAY	53.5	0	1	2.130717
GO_T_CELL_RECEPTOR_COMPLEX	55.0	0	1	1.849038
GO_REGULATION_OF_HOMOTYPIC_CELL_CELL_ADHESION	73.5	0	10	1.515611
GO_REGULATION_OF_LYMPHOCYTE_DIFFERENTIATION	73.5	0	34	1.412907
GO_REGULATION_OF_ALPHA_BETA_T_CELL_ACTIVATION	74.0	0	21	1.415518
GO_POSITIVE_REGULATION_OF_IMMUNE_EFFECTOR_PROCESS	76.5	0	29	1.538169

La función `showSetByname` presenta los detalles correspondientes a los conjuntos de genes para los que se desea información adicional. Aquí se presentan los tres primeros de la lista generada en el paso anterior.

```
ID: M14725
GeneSet: GO_POSITIVE_REGULATION_OF_CELL_CELL_ADHESION
BroadUrl: http://www.broadinstitute.org/gsea/msigdb/cards/GO_POSITIVE_REGULAT
ION_OF_CELL_CELL_ADHESION.html
Description: Any process that activates or increases the rate or extent of ce
ll adhesion to another cell.
PubMedID:
NumGenes: 215/243
Contributor: Gene Ontology
Ontology: BP
GOID: GO:0022409

ID: M14544
```

```
GeneSet: GO_REGULATION_OF_LEUKOCYTE_MEDIATED_IMMUNITY
BroadUrl: http://www.broadinstitute.org/gsea/msigdb/cards/GO_REGULATION_OF_LE
UKOCYTE_MEDIATED_IMMUNITY.html
Description: Any process that modulates the frequency, rate, or extent of leu
kocyte mediated immunity.
PubMedID:
NumGenes: 143/156
Contributor: Gene Ontology
Ontology: BP
GOID: GO:0002703

ID: M11398
GeneSet: GO_POSITIVE_REGULATION_OF_T_CELL_PROLIFERATION
BroadUrl: http://www.broadinstitute.org/gsea/msigdb/cards/GO_POSITIVE_REGULAT
ION_OF_T_CELL_PROLIFERATION.html
Description: Any process that activates or increases the rate or extent of T
cell proliferation.
PubMedID:
NumGenes: 82/95
Contributor: Gene Ontology
Ontology: BP
GOID: GO:0042102
```

La representación gráfica puede ser una manera eficaz de mostrar resultados de significación biológica sobre patrones de expresión genética entre y dentro de conjuntos de genes.

Las prestaciones de `EGSEA` permiten explorar estos resultados con un mapa de colores utilizando la función `plotHeatmap` o un "pathway map" con la función `plotPathway`, que aprovecha las prestaciones de la librería `pathview`. La segunda opción sólo se puede usar con la colección KEGG que no ha sido seleccionada en este trabajo.

La Figura 25 el mapa de colores que resulta de seleccionar la colección "GO_T_CELL_RECEPTOR_COMPLEX" y el contraste "comparison".

La función `plotMethods` genera un gráfico con escalado multidimensional en el que se combinan los resultados obtenidos mediante los métodos empleados con la función `gsa`. Los métodos que generen similares conjuntos de genes similares aparecen muy cerca unos de otros. La Figura 26 presenta los resultados de un escalado multidimensional con los métodos seleccionados en este trabajo.

La función `plotSummary` presenta cada conjunto de genes como una burbuja cuyas coordenadas corresponden a −log10(p-value) (eje X) y el valor medio absoluto de logFC (eje Y). Los conjuntos de genes más relevantes son los que aparecen en la esquina inferior izquierda. `EGSEA` genera dos tipos de gráficos: un resumen direccional que colorea las burbujas teniendo en cuenta la dirección de regulación del conjunto de genes y un resumen de rangos que colorea las burbujas teniendo en cuenta el ranking de los conjuntos de genes para una determinada colección. La Figuras 27 y 28 muestran los dos gráficos obtenidos con el contraste NIT-SFI.

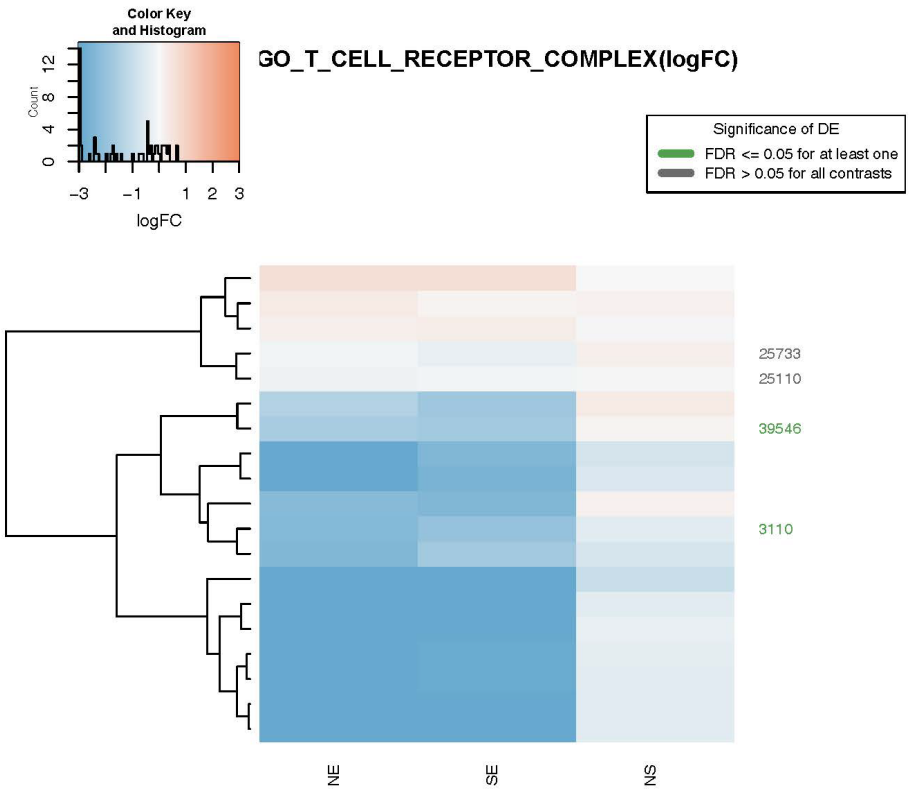

Figura 25. Mapa de colores - GO_T_CELL_RECEPTOR_COMPLEX - Comparison.

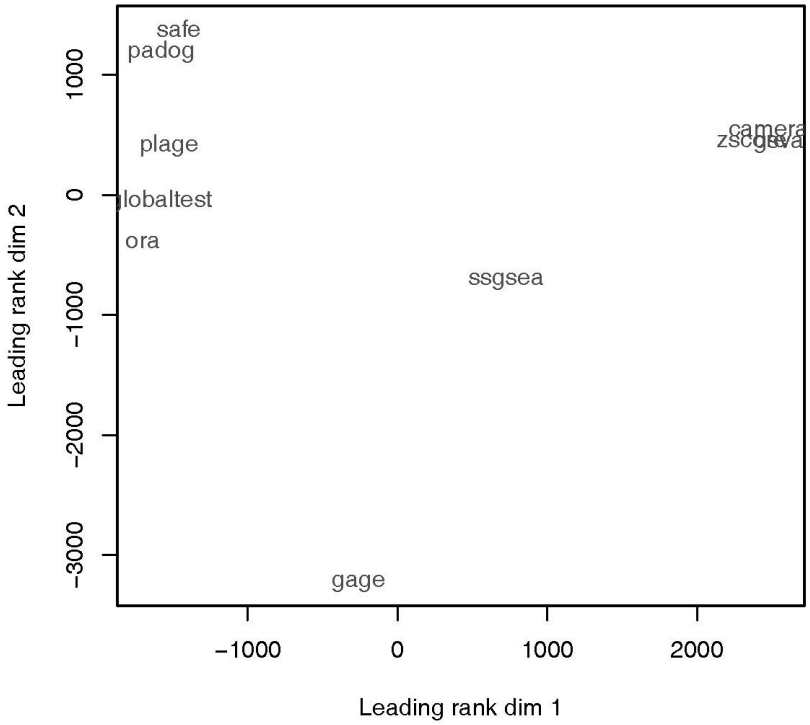

Figura 26. Mapa de colores - GO_T_CELL_RECEPTOR_COMPLEX - Comparison.

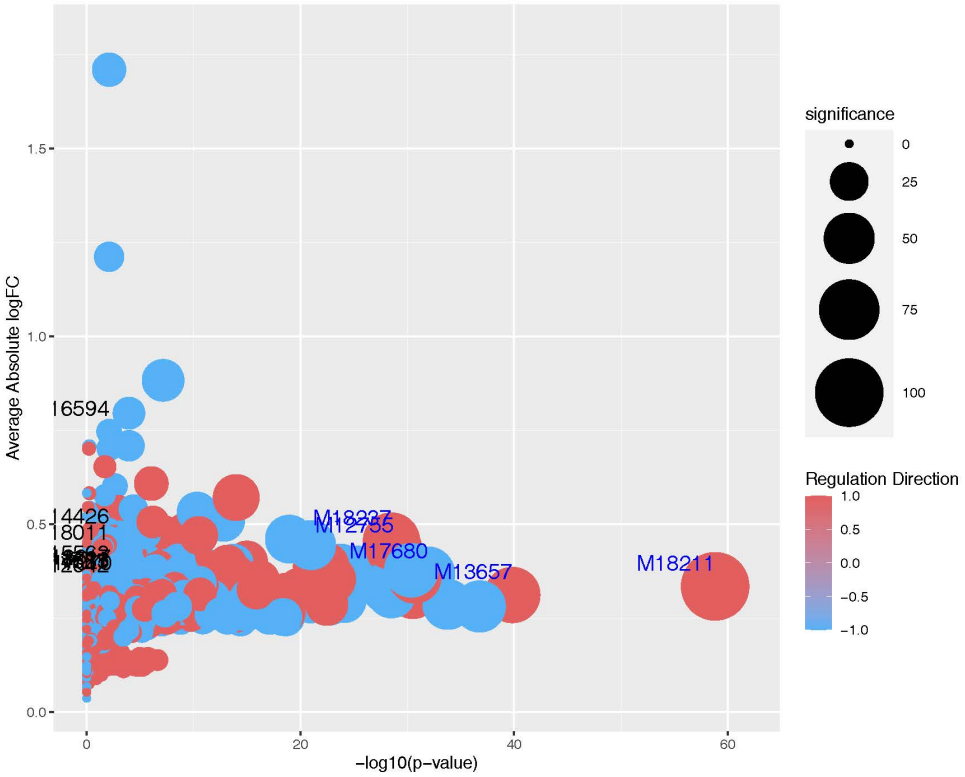

Figura 27. Significado de los conjuntos de genes: Resumen direccional.

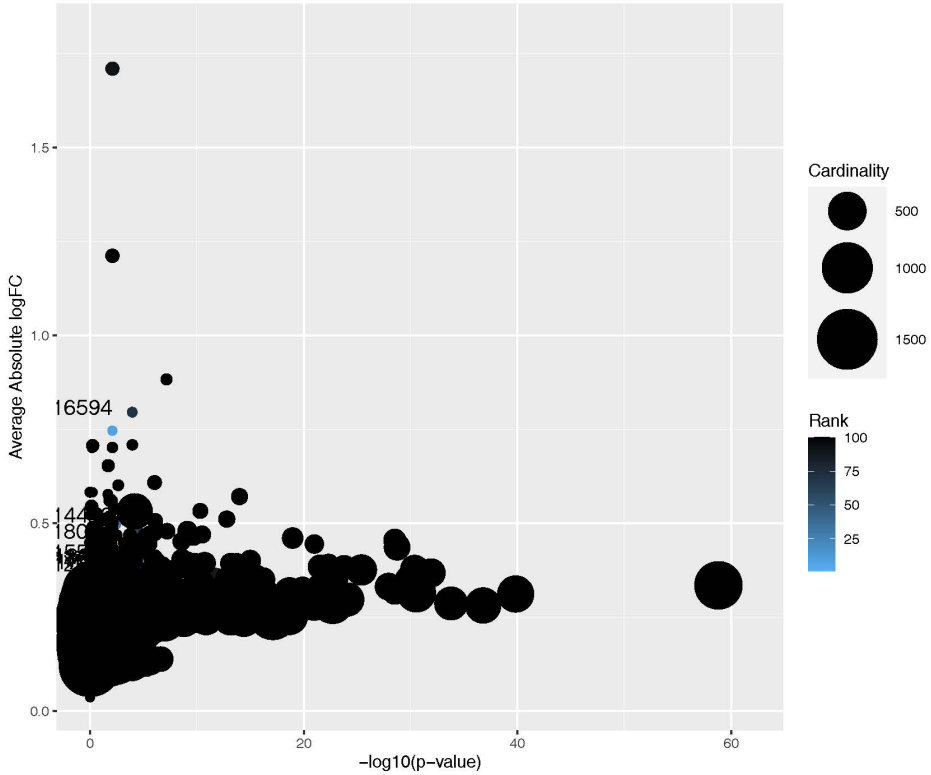

Figura 28. Significado de los conjuntos de genes: Resumen de rangos.

Las opciones de esta función también permiten realizar una selección de los conjuntos de genes especificando un valor umbral (cutoff), seleccionar otras variables para los ejes, o presentar los conjuntos de genes que resultan de una comparación.

La función `plotGOGraph` aprovecha la información recopilada con `gsa` de la base de datos GO para mostrar cómo se relacionan entre si los conjuntos de genes. Las Figuras 29, 30 y 31 presentan los resultados del contraste NIT-ELI ordenados con la opción "avg.rank". Cada figura muestra las relaciones entre los 5 primeros conjuntos de genes para las tres ontologías (MF, CC y BP).

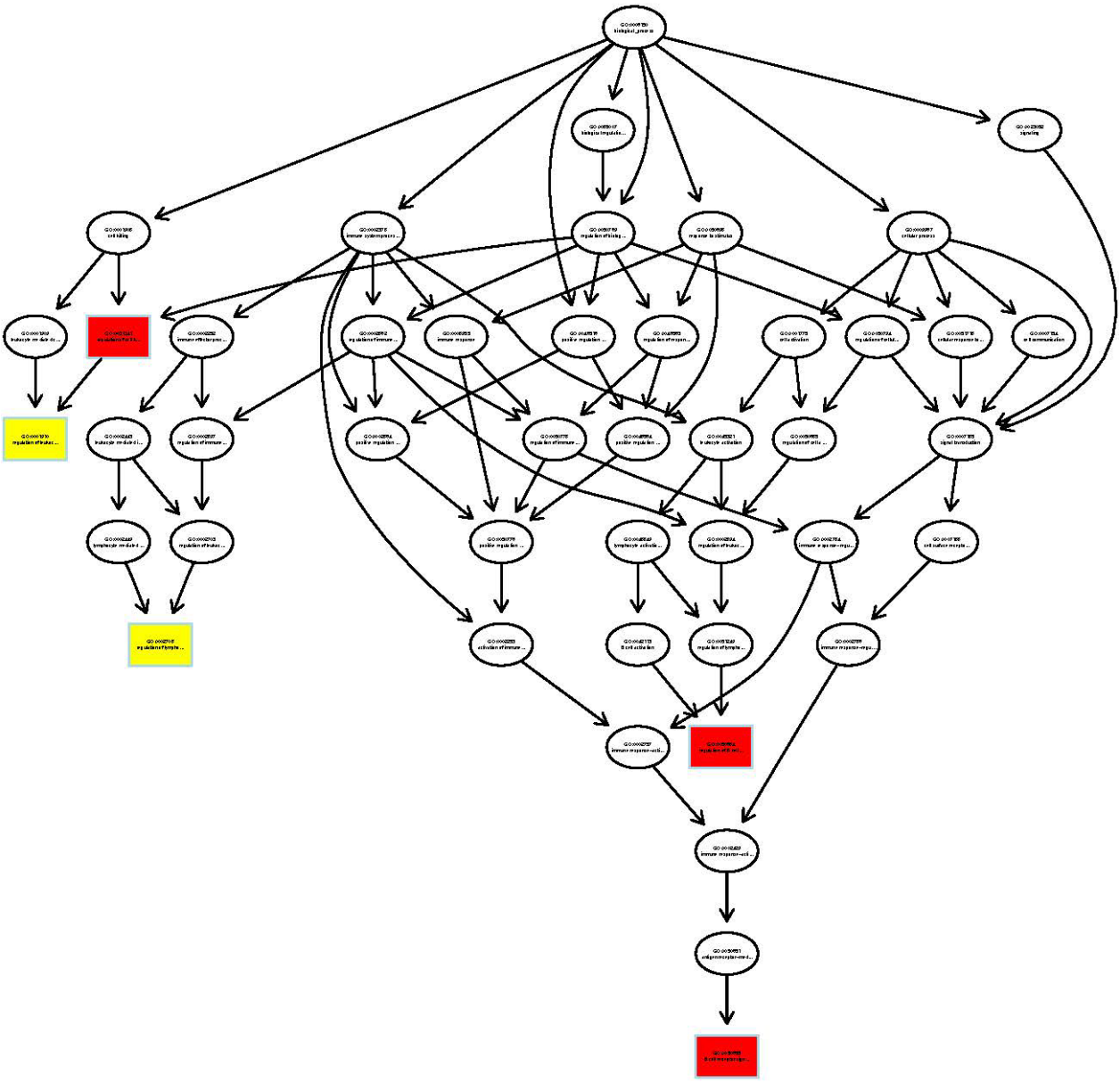

Figura 29. Gráfico GO - Ontología BP.

Se comprueba que los cinco conjuntos están coloreados: el color varía del rojo (los conjuntos más significativos) a amarillo (menos significativos). El número de conjuntos de genes que aparecen en las figuras es, por defecto, 5; el usuario puede cambiar este número.

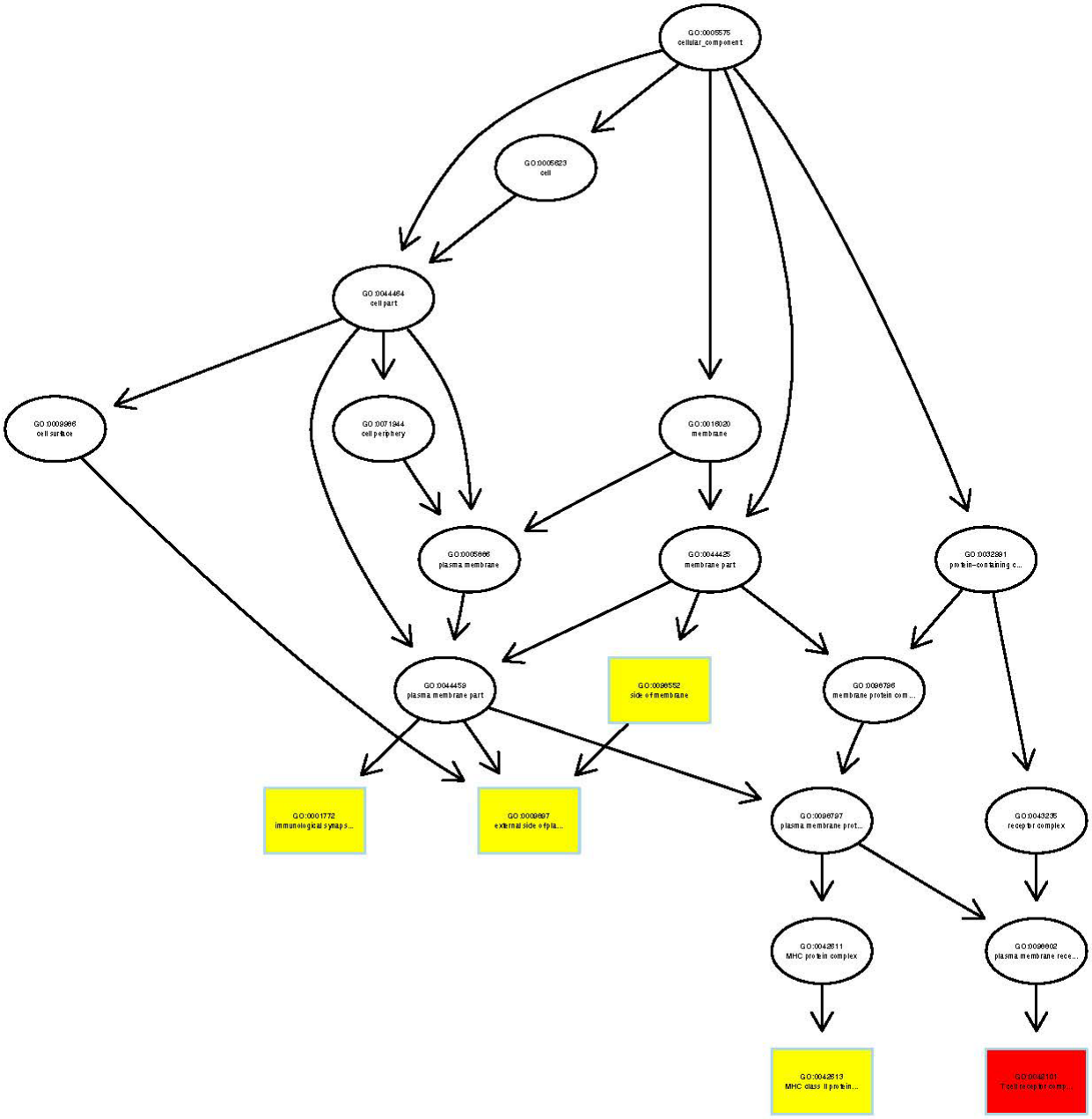

Figura 30. Gráfico GO - Ontología CC.

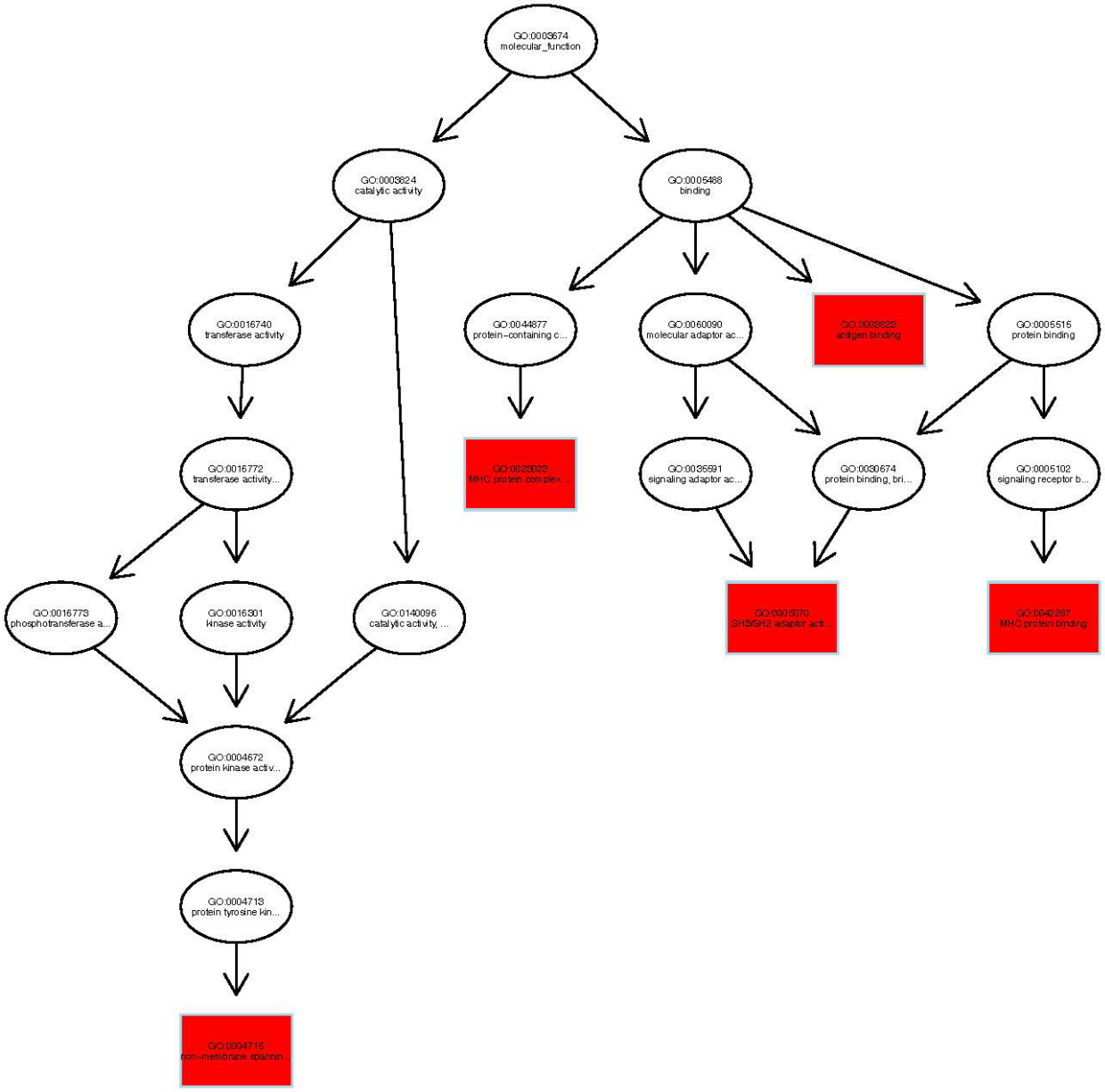

Figura 31. Gráfico GO - Ontología MF.

La función `plotBars` presenta un gráfico de barras con los primeros conjuntos de genes para un contraste particular o múltiple. La Figura 32 presenta el gráfico de barras utilizando la opción "comparison". El color de las barras está basado en la dirección de cada conjunto de genes: rojo para sobre-regulados, púrpura para neutrales (contrastes que muestran comportamientos opuestos), azul para sub-regulados.

Finalmente, se presenta la función `plotSummaryHeatmap` que permite generar un mapa de colores que compara los primeros conjuntos de genes (por defecto los 20 primeros). La figura 33 presenta el resultado que corresponde al presente estudio en el que solo se ha considerado la etiqueta "c5" correspondiente a la clasificación de la base GO.

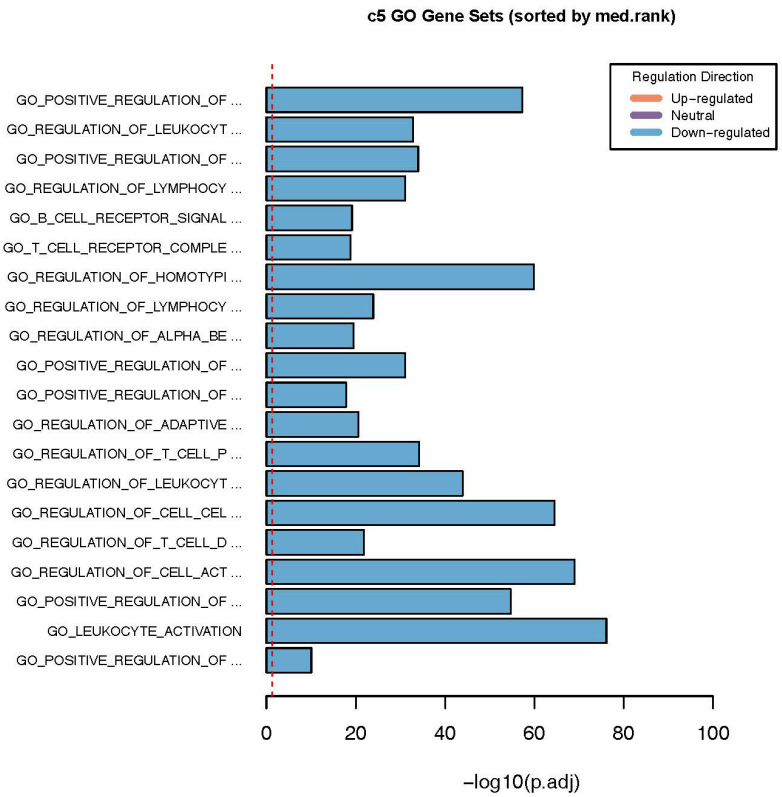

Figura 32. Gráfico de barras - Opción="comparison".

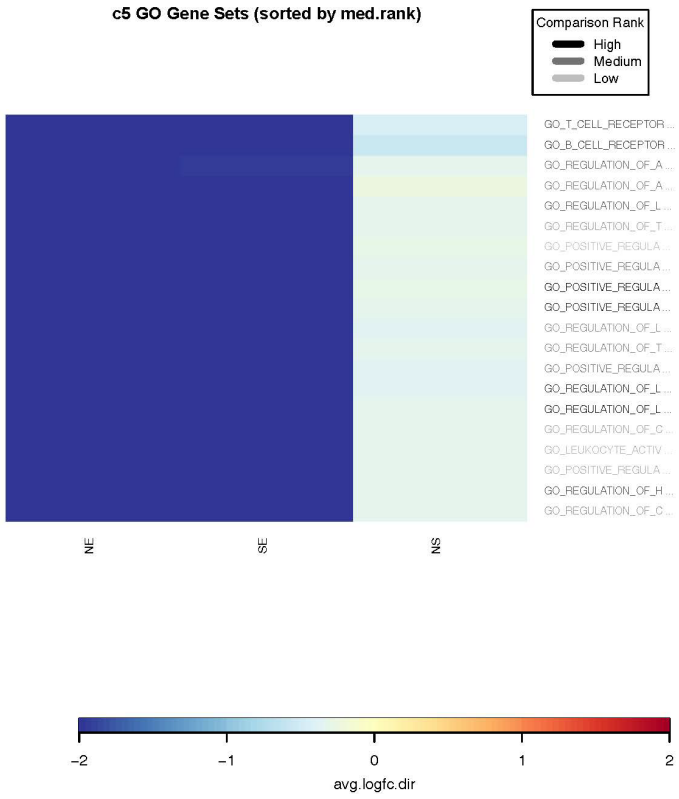

Figura 33. Mapa de colores resumen.

Notas

1 - La identificación de genes diferencialmente expresados se ha realizado aplicando tres procedimientos distintos. Los dos primeros han proporcionado resultados muy parecidos, mientras estos han sido distintos con el tercero. De todas formas, se advierte la misma tendencia en las tres comparaciones con los tres métodos. Es muy probable que una razón para estas discrepancias se encuentre en los parámetros utilizados en cada procedimiento. Por otro lado, es importante tener en cuenta que algunos de los datos y resultados obtenidos con el tercer método, basado en la matriz de contrastes, ha sido la base para el análisis de significación biológica.

2 - El análisis de significación biológica se ha realizado con una lista de símbolos de genes inapropiada; obviamente, esto no ha representado ningún problema para obtener los resultados presentados en este trabajo. Sin embargo, ha impedido añadir algún resultado (mapa de colores, informe) para los que era crucial una lista adecuada de símbolos.

3 - Los resultados del análisis de significación biológica presentados en este trabajo son solo una pequeña parte de los que se pueden presentar ya sea aprovechando la información generada en este trabajo o utilizando todas las prestaciones de las librerías `EGSEA` y `EGSEAdata`. Por ejemplo, por lo que respecta a las anotaciones sólo se ha utilizado la categoría *c5* de la colección *Molecular Signatures Database* (MSigDB), que corresponde a los códigos GO (*Gene Ontology*). Y por lo que respecta a métodos, se ha prescindido de algunos disponibles en `EGSEA` para no repetir algunos cálculos anteriores y limitar el tiempo de cálculo. Aunque la librería `EGSEA` utiliza cálculo paralelo, alguno de sus métodos (p.e. `ssgsea`) requiere un tiempo de cálculo que dobla al de todos los restantes métodos. Para acabar, conviene mencionar que los códigos/anotaciones disponibles en las bases de datos que usa `EGSEAdata` son limitados. Por ejemplo, solo dispone de 6166 códigos GO.

4 - Para la realización de este trabajo se han consultado, en algún caso de forma exhaustiva, una gran parte de las referencias que se listan al final de este trabajo. Otros trabajos que han sido de alguna utilidad se encuentran en los siguientes enlaces:

https://web.stanford.edu/class/bios221/labs/rnaseq/lab_4_rnaseq.html

https://davetang.org/muse/2011/01/24/normalisation-methods-for-dge-data/

http://girke.bioinformatics.ucr.edu/systemPipeR/mydoc_systemPipeRIBOseq_01.html

http://bioinf.wehi.edu.au/edgeR/

http://www.nathalievialaneix.eu/doc/

https://learn.gencore.bio.nyu.edu/

http://manuals.bioinformatics.ucr.edu/home/ht-seq

https://melbournebioinformatics.github.io/MelBioInf_docs/tutorials/rna_seq_dge_basic/rna_seq_basic_tutorial/

https://rnaseq.uoregon.edu/

https://www.bioconductor.org/packages/devel/workflows/vignettes/rnaseqGene/inst/doc/rnaseqGene.html

Resumen de la sesión

Se solicita un resumen con la información de la sesión.

```
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=C

attached base packages:
 [1] grid      parallel  stats4    stats     graphics  grDevices utils
 [8] datasets  methods   base

other attached packages:
 [1] Rgraphviz_2.30.0     EGSEAdata_1.14.0     EGSEA_1.14.0
 [4] pathview_1.26.0      topGO_2.38.1         SparseM_1.78
 [7] graph_1.64.0         gage_2.36.0          GO.db_3.10.0
[10] VennDiagram_1.6.20   futile.logger_1.4.3  RColorBrewer_1.1-2
[13] org.Hs.eg.db_3.10.0  AnnotationDbi_1.48.0 IRanges_2.20.2
[16] S4Vectors_0.24.4     Biobase_2.46.0       BiocGenerics_0.32.0
[19] edgeR_3.28.1         limma_3.42.2         printr_0.1
[22] knitr_1.28

loaded via a namespace (and not attached):
  [1] sn_1.6-2                 lazyeval_0.2.2           GSEABase_1.48.0
  [4] splines_3.6.3            ggplot2_3.3.1            TH.data_1.0-10
  [7] digest_0.6.25            foreach_1.5.0            htmltools_0.4.0
 [10] gdata_2.18.0             magrittr_1.5             memoise_1.1.0
 [13] org.Rn.eg.db_3.10.0      Biostrings_2.54.0        annotate_1.64.0
 [16] KEGGdzPathwaysGEO_1.24.0 matrixStats_0.56.0       sandwich_2.5-1
 [19] jpeg_0.1-8.1             colorspace_1.4-1         blob_1.2.1
 [22] xfun_0.14                dplyr_1.0.0              crayon_1.3.4
 [25] RCurl_1.98-1.2           jsonlite_1.6.1           org.Mm.eg.db_3.10.0
 [28] survival_3.1-12          zoo_1.8-8                iterators_1.0.12
 [31] glue_1.4.1               gtable_0.3.0             zlibbioc_1.32.0
 [34] XVector_0.26.0           R2HTML_2.3.2             hgu133a.db_3.2.3
 [37] KEGG.db_3.2.3            scales_1.1.1             futile.options_1.0.1
 [40] mvtnorm_1.1-0            DBI_1.1.0                rngtools_1.5
 [43] bibtex_0.4.2.2           Rcpp_1.0.4.6             plotrix_3.7-8
 [46] metap_1.3                viridisLite_0.3.0        xtable_1.8-4
 [49] bit_1.1-15.2             GSVA_1.34.0              DT_0.13
```

```
 [52] htmlwidgets_1.5.1       httr_1.4.1              hgu133plus2.db_3.2.3
 [55] gplots_3.0.3            TFisher_0.2.0           ellipsis_0.3.1
 [58] farver_2.0.3            pkgconfig_2.0.3         XML_3.99-0.3
 [61] locfit_1.5-9.4          labeling_0.3            tidyselect_1.1.0
 [64] rlang_0.4.6             later_1.1.0.1           munsell_0.5.0
 [67] tools_3.6.3             generics_0.0.2          RSQLite_2.2.0
 [70] globaltest_5.40.0       HTMLUtils_0.1.7         evaluate_0.14
 [73] stringr_1.4.0           fastmap_1.0.1           yaml_2.2.1
 [76] bit64_0.9-7             caTools_1.18.0          purrr_0.3.4
 [79] KEGGREST_1.26.1         nlme_3.1-148            doRNG_1.8.2
 [82] mime_0.9                formatR_1.7             KEGGgraph_1.46.0
 [85] compiler_3.6.3          shinythemes_1.1.2       plotly_4.9.2.1
 [88] png_0.1-7               tibble_3.0.1            statmod_1.4.34
 [91] geneplotter_1.64.0      stringi_1.4.6           highr_0.8
 [94] Glimma_1.14.0           lattice_0.20-38         Matrix_1.2-18
 [97] multtest_2.42.0         vctrs_0.3.1             mutoss_0.1-12
[100] pillar_1.4.4            lifecycle_0.2.0         GSA_1.03.1
[103] Rdpack_0.11-1           PADOG_1.28.0            data.table_1.12.8
[106] bitops_1.0-6            gbRd_0.4-11             httpuv_1.5.4
[109] R6_2.4.1                hwriter_1.3.2           promises_1.1.0
[112] KernSmooth_2.23-16      codetools_0.2-16        lambda.r_1.2.4
[115] MASS_7.3-51.5           gtools_3.8.2            safe_3.26.0
[118] mnormt_1.5-6            multcomp_1.4-13         tidyr_1.1.0
[121] rmarkdown_2.2           numDeriv_2016.8-1.1     shiny_1.4.0.2
```

Referencias

[1] M.D. Robinson, D.J. McCarthy, and G.K. Smyth, "edgeR: a Bioconductor package for differential expression analysis of digital gene expression data", *Bioinformatics*, 26(1):139-40, 2010. doi: 10.1093/bioinformatics/btp616.

[2] Y. Chen, A.T.L. Lun, and G.K. Smyth, "Differential Expression Analysis of Complex RNA-seq Experiments Using edgeR," Chapter 3 of *Statistical Analysis of Next Generation Sequencing Data*, S. Datta and D. Nettleton (eds.), Springer, 2014, DOI 10.1007/978-3-319-07212-8__3. https://rd.springer.com/chapter/10.1007%2F978-3-319-07212-8_3.

[3] Y. Chen et al., "Package edgeR – Reference Manual," Disponible en el enlace https://bioconductor.org/packages/release/bioc/manuals/edgeR/man/edgeR.pdf

[4] Y. Chen, D. McCarthy, M. Ritchie, M. Robinson, and G. Smyth, "edgeR: differential analysis of sequence read count data User's Guide," April 2020. Disponible en el enlace https://bioconductor.org/packages/release/bioc/vignettes/edgeR/inst/doc/edgeRUsersGuide.pdf.

[5] A. Moisan, I. González, and N. Villa-Vialaneix, "Practical statistical analysis of RNA-Seq data - edgeR," Vignette disponible en http://www.nathalievialaneix.eu/doc/html/solution-edgeR-rnaseq.html.

[6] I. González, "Tutorial - Statistical analysis of RNA-Seq data," Plateforme Bioinformatique-INRA Toulouse, Plateforme Biostatistique-IMT Universite Toulouse III,

Toulouse, November 2014. Existe una versión anterior en Disponible en http://www.nathalievialaneix.eu/doc/pdf/TP-rnaseq-answers.pdf.

[7] Y. Chen, A.T.L. Lun, and G.K. Smyth, "From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline," *F1000Res*, 5: 1438, 201. doi: 10.12688/f1000research.8987.2.

[8] *RNA Sequencing Analysis*, Vignette disponible en https://www.pathwaycommons.org/guide/primers/data_analysis/rna_sequencing_analysis/.

[9] *Bioconductor for High Throughput Sequence Analysis*, Vignette disponible en http://www.bioconductor.org/help/course-materials/2015/useR/bioc-for-sequence-analysis.html.

[10] M. Morgan and N. Delhomme, *R/Bioconductor for High-Throughput Sequence Analysis*, October 2012. Disponible en https://www.ebi.ac.uk/sites/ebi.ac.uk/files/content.ebi.ac.uk/materials/2013/131021_HTS/practical-n.delhomme.pdf.

[11] A. Conesa et al., "A survey of best practices for RNA-seq data analysis," *Genome Biology* 17:13, 2016. DOI 10.1186/s13059-016-0881-8.

[12] S. Goodwin, J.D. McPherson, and W.R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," *Nat Rev Genet*, 17(6):333-51, 2016. doi: 10.1038/nrg.2016.49.

[13] A. McDermaid, B. Monier, J. Zhao, B. Liu, and Q. Ma, "Interpretation of differential gene expression results of RNA-seq data: review and integration", *Briefings in Bioinformatics*, 20(6), 2044–2054, 2019. doi: 10.1093/bib/bby067.

[14] X. Wang, *Next-Generation Sequencing Data Analysis*, CRC Press, Boca Raton, FL-USA, 2016. https://doi.org/10.1201/b19532. ISBN: 978-1-4822-1789-6.

[15] E. Korpelainen et al., *RNA-seq Data Analysis-A Practical Approach*, CRC Press, Boca Raton, FL-USA, 2015. https://doi.org/10.1201/b17457. ISBN: 978-1-4665-9501-9.

[16] S. Datta and D. Nettleton (eds.), *Statistical Analysis of Next Generation Sequencing Data*, Springer, 2014. DOI:978-3-319-07212-8. ISBN 978-3-319-07211-1.

[17] D.J. McCarthy, Y. Chen, and G.K. Smyth, "Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation," *Nucleic Acids Research*, 40(10), 4288-4297, 2012. doi: 10.1093/nar/gks042.

[18] F. Emmert-Streib and G.V. Glazko, "Pathway Analysis of Expression Data: Deciphering Functional Building Blocks of Complex Diseases," *PLoS Comput Biol* 7(5): e1002053, 2011. doi:10.1371/journal.pcbi.1002053.

[19] C.W. Law, M. Alhamdoosh, S. Su, X. Dong, L. Tian, G.K. Smyth, and M.E. Ritchie, "RNA-seq analysis is easy as 1-2-3 with limma, Glimma and edgeR," *F1000Res* 5: ISCB Comm J-1408, 2016. doi: 10.12688/f1000research.9005.3.

[20] G.K. Smyth, M. Ritchie, N. Thorne, J. Wettenhall, W. Shi, and Y. Hu, "limma: Linear Models for Microarray and RNA-Seq Data-User's Guide," April 2020. Disponible en https://www.bioconductor.org/packages/release/bioc/vignettes/limma/inst/doc/usersguide.pdf.

[21] G. Smyth et al., "Package limma," June 2020. Disponible en https://bioconductor.org/packages/release/bioc/manuals/limma/man/limma.pdf

[22] E. Korpelainen et al., "Annotating the Results," Chapter 10 of *RNA-seq Data Analysis-A Practical Approach*, CRC Press, Boca Raton, FL-USA, 2015. https://doi.org/10.1201/b17457. ISBN: 978-1-4665-9501-9.
https://www.taylorfrancis.com/books/9780429169205/chapters/10.1201/b17457-14.

[23] A.T.L. Lun, Y. Chen, and G.K. Smyth, "It's DE-licious: A Recipe for Differential Expression Analyses of RNA-seq Experiments Using Quasi-Likelihood Methods in edgeR," Chapter 19 of *Statistical Genomics: Methods and Protocols*, E. Mathé and S. Davis (eds.), Springer, New York, 2016. DOI 10.1007/978-1-4939-3578-9_19.

[24] M. Alhamdoosh1, M. Ng, N.J. Wilson, J.M. Sheridan, H. Huynh, M.J. Wilson, and M.E. Ritchie, "Combining multiple tools outperforms individual methods in gene set enrichment analyses," *Bioinformatics*, 33(3), 414-424, 2017.
https://doi.org/10.1093/bioinformatics/btw623.
https://academic.oup.com/bioinformatics/article/33/3/414/2875813.

[25] M. Alhamdoosh, L. Tian, M. Ng and M. Ritchie, "Package EGSEA," June 2020. Disponible en
https://www.bioconductor.org/packages/release/bioc/manuals/EGSEA/man/EGSEA.pdf

[26] M. Alhamdoosh1, L. Tian, M. Ng, and M. Ritchie, "Ensemble of Gene Set Enrichment Analyses," April 27, 2020. Disponible en
http://www.bioconductor.org/packages/release/bioc/vignettes/EGSEA/inst/doc/EGSEA.pdf.

[27] M. Alhamdoosh, C.W. Law, L. Tian, J.M. Sheridan, M. Ng, and M.E. Ritchie, "Easy and efficient ensemble gene set testing with EGSEA," *F1000Research*, 6:2010, 2017. doi: 10.12688/f1000research.12544.1. https://f1000research.com/articles/6-2010/v1.

Anexo - Código implantado en R

Se adjunta el código implantado en **R** para la realización de este trabajo.

```
```{r setup, include=FALSE}
library(knitr)
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE,
                      comment = NA, prompt = TRUE, tidy = FALSE,
                      fig.width = 7, fig.height = 7, fig_caption = TRUE,
                      cache=FALSE)
Sys.setlocale("LC_TIME", "C")
```

```{r echo=FALSE}
if(!(require(printr))) {
  install.packages(
    'printr',
    type = 'source',
    repos = c('http://yihui.name/xran', 'http://cran.rstudio.com'))
}
```

```{r echo=FALSE, message=FALSE, eval=FALSE}
# La lista de packages es mas larga de lo realmente necesario para la PEC
if(!require(BiocManager)) install.packages("BiocManager")
if(!require(airway)) BiocManager::install("airway")
if(!require(Rsamtools)) BiocManager::install("Rsamtools")
if(!require(GenomicFeatures)) BiocManager::install("GenomicFeatures")
if(!require(DESeq2)) BiocManager::install("DESeq2")
if(!require(apeglm)) BiocManager::install("apeglm")
if(!require(BiocParallel)) BiocManager::install("BiocParallel")
if(!require(genefilter)) BiocManager::install("genefilter")
if(!require(org.Hs.eg.db)) BiocManager::install("org.Hs.eg.db")
if(!require(AnnotationDbi)) BiocManager::install("AnnotationDbi")
if(!require(ReportingTools)) BiocManager::install("ReportingTools")
if(!require(RUVSeq)) BiocManager::install("RUVSeq")
if(!require(sva)) BiocManager::install("sva")
if(!require(Gviz)) BiocManager::install("Gviz")
if(!require(magrittr)) install.packages("magrittr", dep=TRUE)
if(!require(dplyr)) install.packages("dplyr", dep=TRUE)
if(!require(ggplot2)) install.packages("ggplot2", dep=TRUE)
if(!require(pheatmap)) install.packages("pheatmap", dep=TRUE)
if(!require(RColorBrewer)) install.packages("RColorBrewer", dep=TRUE)
if(!require(ggbeeswarm)) install.packages("ggbeeswarm", dep=TRUE)
if(!require(ggbeeswarm)) install.packages("EGSEA", dep=TRUE)
if(!require(ggbeeswarm)) install.packages("EGSEAdata", dep=TRUE)

#if (!requireNamespace("BiocManager", quietly = TRUE))
#install.packages("BiocManager")
#BiocManager::install("edgeR")
#BiocManager::install("EGSEA")
#BiocManager::install("EGSEAdata")
```

```{r echo=FALSE}
library(edgeR)
```
```

Entrada y preparación de datos

```{r echo=FALSE}
targets <- read.csv("targets.csv", row.names = 1)
cat("Archivo targets", "\n")
class(targets)
str(targets)
```

```{r echo=FALSE}
set.seed(12345)
GNIT <- which(targets$Group=="NIT")
NITsamples <- targets[sample(GNIT, 10), 2]
GSFI <- which(targets$Group=="SFI")
SFIsamples <- targets[sample(GSFI, 10), 2]
GELI <- which(targets$Group=="ELI")
ELIsamples <- targets[sample(GELI, 10), 2]
```

```{r echo=FALSE}
(samplest <- c(NITsamples, SFIsamples, ELIsamples))
targets$ShortName[c(as.matrix(samplest))]
```

Archivo *counts*

```{r echo=FALSE}
tabla <- read.csv("counts.csv", sep=";", header=TRUE)
tabla <- as.data.frame(tabla)
rown <- tabla[,1]        # Se copian los nombres de la primera columna
tabla <- tabla[,-1]      # Se quita la primera columna
rownames(tabla) <- rown  # Se nombran las filas
```

```{r echo=FALSE}
counts <- as.matrix(tabla[, samplest])
colnames(counts)
colnames(counts) <- targets$ShortName[c(as.matrix(samplest))]
rownames(counts) <- rown
```

```{r echo=FALSE}
cat("Archivo counts", "\n")
dim(counts)
counts[1:5,1:5]
```

```{r echo=FALSE}
genes <- as.data.frame(substring(c(rownames(tabla2)), 1, 15))
colnames(genes) <- "RefENSMBL"
head(genes)
library(org.Hs.eg.db)
symbols2 <- as.character(genes[,1])
entrezIDs <- mapIds(org.Hs.eg.db, symbols2, 'ENTREZID', 'ENSEMBL')
genes <- as.data.frame(cbind(genes,entrezIDs))
head(genes)
```

```
counts2 <- cbind(counts2, genes)
```

```{r echo=FALSE}
na_count <-sapply(genes[,2], function(y) sum(length(which(is.na(y)))))
lNAs <- length(which(na_count==1))
cat(paste("Número de NAs en código ENTREZ =",lNAs,"\n"))
```

Se genera un archivo de clase `DGEList`
Se quitan las filas de los genes con todas las entradas a 0
Se renombran los grupos del objeto

```{r echo=FALSE}
countsFull <- DGEList(counts=counts2[,1:30], genes=counts2[,c(31,32)],
                      remove.zeros = TRUE)
names(countsFull)
```

```{r echo=FALSE}
cat("Archivo - Clase DGEList", "\n")
names(countsFull)
# summary(countsFull)
```

```{r echo=FALSE}
cat("Objeto counts","\n")
dim(countsFull$counts)
countsFull$counts[1:5,1:3]
```

```{r echo=FALSE}
cat("Objeto samples","\n")
dim(countsFull$samples)
cat("Antes de renombrar los grupos","\n")
countsFull$samples[c(1:3,11:13,21:23),]
cat("\n")
cat("Después de renombrar los grupos","\n")
countsFull$samples$group <- c(replicate(10,"NIT"),
                              replicate(10,"SIF"),
                              replicate(10,"ELI"))
countsFull$samples[c(1:3,11:13,21:23),]
```

```{r echo=FALSE}
cat("Objeto genes","\n")
countsFull$genes[1:5,]
```

Análisis de los datos

transformación logarítmica.

```{r echo=FALSE}
cat("Datos transformados", "\n")
cat("","\n")
pseudoCounts <- log2(countsFull$counts + 0.5)
```

```
head(pseudoCounts)[1:5,1:5]
```

Se presentan algunos resultados gráficos

```{r echo=FALSE}
library(RColorBrewer)
labels_counts <- as.factor(countsFull$samples$group)
levels_counts <- levels(labels_counts)
ncolors <- length(levels_counts)
colors <- brewer.pal(ncolors, "Set2")
colors_counts <- colors[unname(labels_counts)]
```

```{r fig.cap='Figura 1. Histogramas - Muestra 12.', fig.width=10, echo=FALSE}
par(mfrow=c(1,2))
hist(countsFull$counts[, 12],
     main = "Before transformation", xlab = "counts")
hist(pseudoCounts[ ,12],
     main = "After transformation", xlab = "counts")
```

```{r fig.cap='Figura 2. Diagramas de cajas - Muestras transformadas.', fig.height=10, fig.width=12, echo=FALSE}
par(mar=c(10,4,2,4))
boxplot(pseudoCounts, col=colors_counts, las=2,
        cex.names=0.2, cex.lab=2)
```

```{r fig.cap='Figura 3. Gráfico MA - Muestras 1-2.', echo=FALSE}
limma::plotMA(pseudoCounts[ ,1:2], xlab = "M", ylab = "A",
              main = "Gráfico MA - Muestras 1 y 2")
abline(h = 0, col = "red")
```

```{r fig.cap='Figura 4. Escalado multidimensional.', echo=FALSE}
plotMDS(pseudoCounts, pch=c(21:23), bg = colors, cex=1.5,
        gene.selection = "common")
legend("topright", legend=levels(targets$Group),
       pch=c(21:23), pt.bg=brewer.pal(ncolors, "Set2"))
```

```{r fig.cap='Figura 5. Dendrograma de las muestras.', fig.height=10, fig.width=12, echo=FALSE}
sampleDists <- as.matrix(dist(t(pseudoCounts)))
cs  <- hclust(dist(t(pseudoCounts)), method = "ward.D2")
dmd <- as.dendrogram(cs)
plot(dmd, ylim=c(0,700))
```

```{r fig.cap='Figura 6. Mapa de color de las muestras.',fig.height=8, fig.width=12, echo=FALSE}
par(mar=c(18,4,2,4))
heatmap(sampleDists)
```

Filtrado y normalización de los datos

```
### Filtrado

```{r fig.height=10, fig.width=12, echo=FALSE}
filtered.group <- filterByExpr(countsFull)
filtered.group <- countsFull[filtered.group, keep.lib.sizes=FALSE]
```

```{r echo=FALSE, eval=FALSE}
cat("Antes del filtrado","\n")
countsFull$samples[1:5,]
cat("\n")
cat("Después del filtrado","\n")
filtered.group$samples[1:5,]
```

```{r echo=FALSE}
cat("Dimensiones antes del filtrado", "\n")
dim(countsFull)
cat("\n")
cat("Dimensiones después del filtrado", "\n")
dim(filtered.group)
```

Normalización

```{r echo=FALSE}
filtered.group <- calcNormFactors(filtered.group, method="TMM")
filtered.group$samples[1:5,]
```

Algunos resultados gráficos con datos filtrados y normalizados

```{r fig.cap='Figura 7. Diagramas de cajas de los datos filtrados.',
echo=FALSE}
filter.pseudoCounts <- log2(filtered.group$counts + 1)
par(mar=c(10,4,2,4))
boxplot(filter.pseudoCounts, col=colors_counts, las = 2, cex.names = 1)
```

```{r fig.cap='Figura 8. Gráfico BCV de los datos filtrados y normalizados.',
echo=FALSE}
filtered.group$samples$group <- as.factor(filtered.group$samples$group)
filtered.group2 <- estimateCommonDisp(filtered.group, verbose=TRUE)
filtered.group2 <- estimateTagwiseDisp(filtered.group2)
class(filtered.group2)
plotBCV(filtered.group2)
```

Expresión diferencial

Identificación de genes diferencialmente expresados

```{r echo=FALSE}
et12 <- exactTest(filtered.group2, pair=c(1,2)) # compare groups 1 and 2
de1 <- decideTestsDGE(et12, adjust.method="BH", p.value=0.05)
cat("Comparación NIT-ELI","\n")
```
```

```
summary(de1)
```

```{r fig.cap='Figura 9. Gráfico MA - Comparación NIT-ELI.', echo=FALSE}
cat("\n")
de1tags12 <- rownames(filtered.group2)[as.logical(de1)]
plotSmear(et12, de.tags=de1tags12, main="Comparison NIT-ELI")
abline(h = c(-2, 2), col = "blue")
```

```{r echo=FALSE}
et23 <- exactTest(filtered.group2, pair=c(2,3)) # compare groups 2 and 3
de2 <- decideTestsDGE(et23, adjust.method="BH", p.value=0.05)
cat("Comparación SIF-NIT","\n")
summary(de2)
```

```{r fig.cap='Figura 10. Gráfico MA - Comparación SIF-NIT.', echo=FALSE}
cat("\n")
de1tags23 <- rownames(filtered.group2)[as.logical(de1)]
plotSmear(et23, de.tags=de1tags12, main="Comparison SIF-NIT")
abline(h = c(-2, 2), col = "blue")
```

```{r echo=FALSE}
et31 <- exactTest(filtered.group2, pair=c(3,1)) # compare groups 3 and 1
de3 <- decideTestsDGE(et31, adjust.method="BH", p.value=0.05)
cat("Comparación ELI-SIF","\n")
summary(de3)
```

```{r fig.cap='Figura 11. Gráfico MA - Comparación ELI-SIF.', echo=FALSE}
cat("\n")
de1tags31 <- rownames(filtered.group2)[as.logical(de1)]
plotSmear(et31, de.tags=de1tags12, main="Comparison ELI-SIF")
abline(h = c(-2, 2), col = "blue")
```

opción ``topTags``

```{r echo=FALSE}
topTags(et12, n=5)[,2:5]
```

```{r echo=FALSE}
lfc  <- 2
pval <- 0.05
```

```{r fig.cap='Figura 12. Volcano plot - Comparación NIT-ELI.', echo=FALSE}
par(mar = c(4, 4, 4, 4))
taget12 <- topTags(et12, n=22185)
taget12 <- taget12[,-c(1:2)]
tab12 <- data.frame(logFC=taget12$table[, 1],
                    negLogPval=-log10(taget12$table[, 3]))
plot(tab12$logFC, tab12$negLogPval, pch = 16, cex = 0.6,
     xlab = expression(log[2]~fold~change),
```

```
     ylab = expression(-log[10]~pvalue))
signGenes = (abs(tab12$logFC) > lfc & tab12$negLogPval > -log10(pval))
points(tab12[signGenes, ], pch = 16, cex = 0.8, col = "red")
abline(h = -log10(pval), col = "green3", lty = 2)
abline(v = c(-lfc, lfc), col = "blue", lty = 2)
mtext(paste("pval =", pval), side = 4, at = -log10(pval),
      cex = 0.8, line = 0.5, las = 1)
mtext(c(paste("-", lfc, "fold"), paste("+", lfc, "fold")),
      side = 3, at = c(-lfc, lfc), cex = 0.8, line = 0.5)
```

```{r fig.cap='Figura 13. Volcano plot - Comparación SIF-NIT.', echo=FALSE}
par(mar = c(4, 4, 4, 4))
taget23 <- topTags(et23, n=22185)
taget23 <- taget23[,-c(1:2)]
tab23 <- data.frame(logFC=taget23$table[, 1],
                    negLogPval=-log10(taget23$table[, 3]))
plot(tab23, pch = 16, cex = 0.6,
     xlab = expression(log[2]~fold~change),
     ylab = expression(-log[10]~pvalue))
signGenes = (abs(tab23$logFC)>lfc & tab23$negLogPval>-log10(pval))
points(tab23[signGenes, ], pch = 16, cex = 0.8, col = "red")
abline(h = -log10(pval), col = "green3", lty = 2)
abline(v = c(-lfc, lfc), col = "blue", lty = 2)
mtext(paste("pval =", pval), side = 4, at = -log10(pval),
      cex = 0.8, line = 0.5, las = 1)
mtext(c(paste("-", lfc, "fold"), paste("+", lfc, "fold")),
      side = 3, at = c(-lfc, lfc), cex = 0.8, line = 0.5)
```

```{r fig.cap='Figura 14. Volcano plot - Comparación ELI-SIF.', echo=FALSE}
par(mar = c(4, 4, 4, 4))
taget31 <- topTags(et31, n=22185)
taget31 <- taget31[,-c(1:2)]
tab31 <- data.frame(logFC=taget31$table[, 1],
                    negLogPval=-log10(taget31$table[, 3]))
plot(tab31, pch = 16, cex = 0.6,
     xlab = expression(log[2]~fold~change),
     ylab = expression(-log[10]~pvalue))
signGenes = (abs(tab31$logFC)>lfc & tab31$negLogPval>-log10(pval))
points(tab31[signGenes, ], pch = 16, cex = 0.8, col = "red")
abline(h = -log10(pval), col = "green3", lty = 2)
abline(v = c(-lfc, lfc), col = "blue", lty = 2)
mtext(paste("pval =", pval), side = 4, at = -log10(pval),
      cex = 0.8, line = 0.5, las = 1)
mtext(c(paste("-", lfc, "fold"), paste("+", lfc, "fold")),
      side = 3, at = c(-lfc, lfc), cex = 0.8, line = 0.5)
```

Aplicación de un modelo lineal generalizado (GLM)

Se crea la matriz de diseño

```{r echo=FALSE}
G <- as.factor(countsFull$samples$group)
design.matrix  <- model.matrix(~ 0+G)
labels <- colnames(countsFull$counts)
```

```
rownames(design.matrix) <- labels
cat("Matriz de diseño","\n","\n")
design.matrix
```

Se calcula la dispersión común y se aplica el modelo lineal

```{r echo=FALSE}
group1 <- estimateGLMCommonDisp(filtered.group2, design.matrix)
```

```{r echo=FALSE}
cat("Objetos creados con la función estimateGLMCommonDisp","\n","\n")
names(group1)
```

Dispersión común y los valores mínimo y máximo de la dispersión

```{r echo=FALSE}
cat("Dispersión común", "\n","\n")
group1$common.dispersion
```

```{r echo=FALSE}
cat("Valores mínimo y máximo de la dispersión común", "\n","\n")
summary(group1$tagwise.dispersion)
```

Se aplican ``estimateGLMTrendedDisp`` y ``estimateGLMTagwiseDisp``

```{r echo=FALSE}
group1 <- estimateGLMTrendedDisp(group1, design.matrix)
group1 <- estimateGLMTagwiseDisp(group1, design.matrix)
```

Gráfico BCV

```{r fig.cap='Figura 15. Gráfico BCV - modelo lineal generalizado (GLM).', echo=FALSE}
plotBCV(group1)
```

Se realizan los contrastes

```{r echo=FALSE}
fit <- glmFit(group1, design.matrix)
```

```{r echo=FALSE}
cat("Objetos del modelo lineal generalizado (GLM)", "\n","\n")
names(fit)
```

```{r echo=FALSE}
dgeLRTtest1 <- glmLRT(fit, contrast=c(-1,1,0))
topTags(dgeLRTtest1, n=5)
de1 <- decideTestsDGE(dgeLRTtest1, adjust.method="BH", p.value = 0.05)
```

```
summary(de1)
de2tags1 <- rownames(group1)[as.logical(de1)]
```

```{r fig.cap='Figura 16. Gráfico MA - Comparación NIT-ELI.', echo=FALSE}
plotSmear(dgeLRTtest1, de.tags=de2tags1, main="Comparación NIT-ELI")
abline(h = c(-2, 2), col = "blue")
```

```{r echo=FALSE}
dgeLRTtest2 <- glmLRT(fit, contrast=c(0,1,-1))
topTags(dgeLRTtest2, n=5)
de2 <- decideTestsDGE(dgeLRTtest2, adjust.method="BH", p.value = 0.05)
summary(de2)
de2tags2 <- rownames(group1)[as.logical(de2)]
```

```{r echo=FALSE}
dgeLRTtest3 <- glmLRT(fit, contrast=c(-1,0,1))
topTags(dgeLRTtest3, n=5)
de3 <- decideTestsDGE(dgeLRTtest3, adjust.method="BH", p.value = 0.05)
summary(de3)
de2tags3 <- rownames(group1)[as.logical(de3)]
```

Aplicación de la matriz de contrastes

```{r echo=FALSE}
contrast_levels <- colnames(design.matrix)
contrast.matrix <- makeContrasts(NE = GNIT - GELI,
                                 NS = GNIT - GSIF,
                                 SE = GSIF - GELI,
                                 levels = contrast_levels)
cat("Matriz de contrastes","\n","\n")
contrast.matrix
```

Se inicia con ``filterByExpr`` y continua con ``voom``

```{r fig.cap='Figura 17. Resultado función voom.', echo=FALSE}
keep <- filterByExpr(countsFull, design.matrix)
length(which(keep==TRUE))
v <- voom(countsFull[keep,], design.matrix, plot=TRUE)
id=rownames(v)
```

```{r echo=FALSE}
vfitx <- limma::lmFit(v, design.matrix)
vfitx <- contrasts.fit(vfitx, contrasts=contrast.matrix)
efitx <- eBayes(vfitx)
# head(efitx)
summary(decideTests(efitx))
```

```{r fig.cap='Figura 18. Modelo final - Tendencia media-varianza.', echo=FALSE}
plotSA(efitx) #main="Final model: Mean-variance trend")
```

```{r echo=FALSE}
tfitx <- treat(vfitx, lfc=0.05)
dtx <- decideTests(tfitx)
summary(dtx)
```

```{r echo=FALSE}
cat("\n")
cat("Contraste NIT-ELI","\n")
head(topTreat(tfitx, coef=1, n=Inf))
cat("\n")
cat("Contraste NIT-SIF","\n")
head(topTreat(tfitx, coef=2, n=Inf))
cat("\n")
cat("Contraste SIF-ELI","\n")
head(topTreat(tfitx, coef=3, n=Inf))
```

Gráfico tipo MD

```{r fig.cap='Figura 19. Modelo MD - Contraste NIT-ELI.', echo=FALSE}
plotMD(tfitx, column=1, status=dtx[,1], main="Contraste NIT-ELI")
```

Diagramas de Venn

```{r fig.height=6, fig.width=6, fig.cap='Figura 20. Diagrama de Venn con las tres comparaciones - Métodos 1 y 2.', echo=FALSE}
library(VennDiagram)
#par(mar=c(4,30,4,30))
vd <- venn.diagram(x = list("NIT-ELI" = de2tags1,
                            "NIT-SIF" = de2tags2,
                            "SIF-ELI" = de2tags3),
                   fill = brewer.pal(3, "Set2")[1:3],
                   filename = NULL)
grid.draw(vd)
```

```{r fig.height=6, fig.width=6, fig.cap='Figura 21. Diagrama de Venn con las tres comparaciones - Método 3.', echo=FALSE}
vennDiagram(dtx[,1:3], circle.col=brewer.pal(3, "Set2")[1:3])
```

Análisis de enriquecimiento biológico

Términos GO para el contraste NIT-ELI

```{r echo=FALSE}
fitT <- glmQLFit(group1, design.matrix, robust=TRUE)
con <- makeContrasts(GNIT - GELI, levels=design.matrix)
qlf <- glmQLFTest(fitT, contrast=con)
go <- goana(c(qlf$genes[,1]), species = "Hs")
```

```{r echo=FALSE}
topGO(go, n=10, truncate=20)
```

```
dim(go)
```

Selecciónde términos BP

```{r echo=FALSE}
cat("Tipos de ontologías","\n")
levels(as.factor(go$Ont))
cat("\n")
```

```{r echo=FALSE}
goBPl <- which(go$Ont=="BP")
cat("Número de términos GO-BP", length(goBPl),"\n")
```

```{r echo=FALSE}
goBP <- subset(go, go$Ont=="BP")
head(goBP[order(goBP[,5]),])
cat("\n","\n")
tail(goBP[order(goBP[,4]),])
```

se repite el proceso con la base KEGG

```{r echo=FALSE}
keg <- kegga(c(qlf$genes[,1]), species="Hs")
dim(keg)
topKEGG(keg, n=10, truncate=25)
```

```{r echo=FALSE}
library(GO.db)
cyt.go <- c("GO:2000823", "GO:0045893", "GO:0070324",
            "GO:0042403", "GO:0006590", "GO:0002154",
            "GO:0030375", "GO:0003713")
(term <- select(GO.db, keys=cyt.go, columns="TERM"))
```

Los identificadores con la función ``ids2indices``

```{r echo=FALSE}
# rm(org.Hs.egGO2ALLEGS)
Rkeys(org.Hs.egGO2ALLEGS) <- cyt.go
# length(as.list(org.Hs.egGO2ALLEGS)[[1]])
# length(as.list(org.Hs.egGO2ALLEGS)[[8]])
ind <- ids2indices(as.list(org.Hs.egGO2ALLEGS), group1$genes[,2])
```

función ``roast``

```{r echo=FALSE}
y   <- estimateDisp(group1, design.matrix)
con <- makeContrasts(GNIT-GELI, levels=design.matrix)
rst <- mroast(y, index=ind, design=design.matrix, nrot=9999,
              contrast=con)
rst
```

```
```

Función ``fry``

```{r echo=FALSE}
fry(y, index=ind, design=design.matrix)
```

Función ``camera``

```{r echo=FALSE}
camera(y, ind, design.matrix)
```

Se repite el proceso con el contraste NIT-SIF

```{r echo=FALSE}
con <- makeContrasts(GNIT-GSIF, levels=design.matrix)
rst <- mroast(y, index=ind, design=design.matrix, nrot=9999,
              contrast=con)
rst
```

Visuallización de resultados con ``barcodeplot``

```{r fig.cap='Figura 22. Gráfico código de barras para el término GO:0006590.',
echo=FALSE}
barcodeplot(qlf$table$logFC, ind[[5]], main=names(ind)[5])
```

```{r fig.cap='Figura 23. Gráfico código de barras para GO:2000823 y
GO:0045893.', echo=FALSE}
barcodeplot(qlf$table$logFC, ind[[1]], ind[[2]], main=names(ind)[c(1,2)])
```

```{r fig.cap='Figura 24. Gráfico código de barras para GO:0006590 y
GO:0030375.', echo=FALSE}
barcodeplot(qlf$table$logFC, ind[[4]], ind[[5]], main=names(ind)[4:5])
```


Análisis de significación biológica

Instalación de ``EGSEA`` y ``EGSEAdata``

```{r echo=FALSE}
library(EGSEA)
library(EGSEAdata)
```

```{r  echo=FALSE}
# info <- egsea.data("human", returnInfo = TRUE)
# names(info)
egsea.data("human")
```

Anotaciones

Se prepara el archivo de trabajo

```{r echo=FALSE}
countsB <- countsFull
nsym <- dim(countsB$genes)[1]
Symbol <- as.data.frame(c(seq(1,nsym,1)))
colnames(Symbol) <- "Symbol"
countsB$genes <- cbind(countsFull$genes, Symbol)
countsB$genes <- countsB$genes[,-1]
```

```{r echo=FALSE}
cat("Objeto genes","\n","\n")
head(countsB$genes)
cat("Objeto counts","\n","\n")
countsB$counts[1:5,1:5]
cat("Objeto samples","\n","\n")
head(countsB$samples)
```

Se eliminan las filas (identificador *Entrez* NA y código *Entrez* repetido)

```{r echo=FALSE}
cat("Antes de eliminar NAs y duplicados","\n")
cat("\n")
cat("Dimensiones del objeto genes","\n")
dim(countsB$genes)
cat("Dimensiones del objeto counts","\n")
dim(countsB$counts)
cat("\n")
narows <- which(is.na(countsB$genes$entrezIDs), arr.ind=TRUE)
countsB$genes <- countsB$genes[-c(narows),]
countsB$counts <- countsB$counts[-c(narows),]
rownames(countsB$counts) <- countsB$genes[,1]
dup <- duplicated(countsB$genes[,1])
dupr <- which(dup=="TRUE")
countsB$genes <-countsB$genes[-dupr,]
countsB$counts <-countsB$counts[-dupr,]
cat("Después de eliminar NAs y duplicados","\n")
cat("Dimensiones del objeto genes","\n")
dim(countsB$genes)
cat("Dimensiones del objeto counts","\n")
dim(countsB$counts)
```

Se normaliza el objeto *samples* y se aplica la función ``voom``

```{r echo=FALSE}
countsB <- calcNormFactors(countsB, method="TMM")
keepv <- filterByExpr(countsB, design.matrix)
v2 <- voom(countsB[keepv,], design.matrix, plot=FALSE)
rownames(v2$genes) <- c(seq(1, dim(v2$genes)[1],1))
```

```{r echo=FALSE}
names(v2)
```

```

```{r echo=FALSE}
cat("Número final de genes =",dim(v2$genes)[1],"\n")
```

Anotaciones ``MSigDB`` y la colección ``c5``

```{r echo=FALSE}
genes_n <- as.character(v2$genes$entrezIDs)
dd <- as.numeric(genes_n)
annotaciones <- buildMSigDBIdx(entrezIDs=dd,
                               species = "Homo sapiens",
                               geneSets = "c5", go.part = FALSE,
                               min.size = 1)
```

```{r echo=FALSE}
print(annotaciones)
```

```{r echo=FALSE}
slotNames(annotaciones$c5)
```

```{r echo=FALSE}
annotaciones$c5$featureIDs[1:20]
# class(annotaciones$c5$featureIDs)
```

Mapa de símbolos

```{r echo=FALSE}
Map = v2$genes[, c(1, 2)]
Map = cbind(v2$genes[,1], v2$genes[,2])
colnames(Map) = c("FeatureID", "Symbol")
```

```{r echo=FALSE}
(Methods = egsea.base()[-c(2, 12)])
```

```{r echo=FALSE}
egsea.sort()
```

```{r echo=FALSE}
egsea.combine()
```

Se realiza el análisis de significación biológica con ``gsa``

```{r echo=FALSE}
gsa = egsea(voom.results=v2, contrasts=contrast.matrix,
            gs.annots=annotaciones, symbolsMap=Map,
            baseGSEAs=Methods, sort.by="med.rank",
            num.threads=3, report=FALSE)
```
```

```

```{r}
show(gsa)
slotNames(gsa)
summary(gsa)
```

```{r echo=FALSE}
t = topSets(gsa, contrast=1, gs.label="c5", sort.by="ora",
            number = 10, names.only=FALSE)
t[,1:5]
cat("\n","\n")
t[,6:10]
cat("\n")
```

```{r echo=FALSE}
t = topSets(gsa, contrast="comparison", gs.label="c5",
            number = 10, names.only=FALSE)
t[,1:5]
cat("\n","\n")
t[,6:9]
cat("\n")
```

```{r echo=FALSE}
showSetByName(gsa, "c5", rownames(t)[1:3])
```

Algunos resultados gráficos

```{r echo=FALSE}
plotHeatmap(gsa, gene.set="GO_T_CELL_RECEPTOR_COMPLEX", gs.label="c5",
            contrast = "comparison",
            file.name = "heatmap_GO_T_CELL_RECEPTOR_COMPLEX")
```

```{r fig.cap='Figura 25. Mapa de colores - GO_T_CELL_RECEPTOR_COMPLEX - Comparison.', echo=FALSE}
knitr::include_graphics('heatmap_GO_T_CELL_RECEPTOR_COMPLEX.png')
```

```{r echo=FALSE}
plotMethods(gsa, gs.label = "c5", contrast = "NE", file.name = "MDS_NIT-ELI")
```

```{r fig.cap='Figura 26. Mapa de colores - GO_T_CELL_RECEPTOR_COMPLEX - Comparison.', echo=FALSE}
knitr::include_graphics('MDS_NIT-ELI.jpg')
```

```{r echo=FALSE}
plotSummary(gsa, gs.label = 1, contrast = "NS", file.name = "Resumen NIT-SFI")
```
```

```{r fig.cap='Figura 27. Significado de los conjuntos de genes: Resumen direccional.', echo=FALSE}
knitr::include_graphics('Resumen NIT-SFI-dir.jpg')
```

```{r fig.cap='Figura 28. Significado de los conjuntos de genes: Resumen de rangos.', echo=FALSE}
knitr::include_graphics('Resumen NIT-SFI-rank.jpg')
```

```{r echo=FALSE}
plotGOGraph(gsa, gs.label="c5", file.name="Contraste_NIT-ELI",
            sort.by="avg.rank")
```

```{r fig.cap='Figura 29. Gráfico GO - Ontología BP.', echo=FALSE}
knitr::include_graphics('Contraste_NIT-ELIBP.jpg')
```

```{r fig.cap='Figura 30. Gráfico GO - Ontología CC.', echo=FALSE}
knitr::include_graphics('Contraste_NIT-ELICC.jpg')
```

```{r fig.cap='Figura 31. Gráfico GO - Ontología MF.', echo=FALSE}
knitr::include_graphics('Contraste_NIT-ELIMF.jpg')
```

```{r echo=FALSE}
plotBars(gsa, gs.label = "c5", contrast = "comparison", file.name="Comparacion_Barras")
```

```{r fig.cap='Figura 32. Gráfico de barras - Opción="comparison".', echo=FALSE}
knitr::include_graphics('Comparacion_Barras.jpg')
```

```{r echo=FALSE}
plotSummaryHeatmap(gsa, gs.label="c5", hm.vals = "avg.logfc.dir",
                   file.name="Resumen_HeatMap")
```

```{r fig.cap='Figura 33. Mapa de colores resumen.', echo=FALSE}
knitr::include_graphics('Resumen_HeatMap.jpg')
```

Resumen de la sesión

```{r echo=FALSE}
sessionInfo()
```