

PEC 2: ANÁLISIS DE DATOS DE RNAseq PROCEDENTES DE MUESTRAS DE INFILTRACIÓN EN TIROIDES

Github: https://github.com/sofiasofia2208/PEC2_SZ.git

TABLA DE CONTENIDOS

1. Abstract	p. 1
2. Objetivos	p. 1
3. Material.....	p. 2
4. Métodos y Resultados	p. 2
4.1. Preparación de los datos.....	p. 2
4. 2. Preprocesado de los datos: filtraje y normalización.....	p. 3
4.3. Identificación de genes diferencialmente expresados.....	p.6
4.4. Agrupación de los genes más diferencialmente expresados.....	p.10
4.5. Anotación de genes y exportación de resultados.....	p.10
4.6. Estudio de enriquecimiento de genes (GO) y rutas metabólicas (KEGG).....	p.11
5. Discusión	p.17
6. Conclusión	p.17
6. Bibliografía	p.17
Anexo: Código R utilizado	p.17

1. ABSTRACT

Se ha estudiado mediante análisis bioinformáticos si existen diferencias en la expresión génica en muestras de tejido tiroideo de pacientes sin, con pequeña o con gran infiltración en este órgano. Se han observado genes diferencialmente expresados que están sobre todo implicados en la respuesta inmune, siendo las mayores diferencias aquellas encontradas en los tejidos con gran infiltración que en otras comparaciones.

2. OBJETIVOS

El objetivo es hallar si existen genes que se expresan diferencialmente entre las combinaciones por pares de los tipos de infiltración en la tiroides en humanos: SFI versus NIT, SFI versus ELI y ELI versus SFI.

Asimismo, otro objetivo que subyace del anterior es conocer si existe enriquecimiento de genes implicados en distintos procesos biológicos o de rutas metabólicas cuando se comparan datos de expresión de distintos grupos.

3. MATERIAL

Los datos utilizados en este estudio proceden de un dataset proporcionado por el profesor en la asignatura Análisis de Datos Ómicos, cuyo origen es del repositorio del Genotype-Tissue Expression (GTEx) Project.

Las muestras estudiadas (n=30) proceden de la extracción de RNA de tejido de tiroides de tres tipos de infiltración ("Group"):

- Not infiltrated tissues (NIT): 10 muestras
- Small focal infiltrates (SFI): 10 muestras
- Extensive lymphoid infiltrates (ELI): 10 muestras

Todas las muestras de estudios contienen datos de expresión génica (RNAseq). En el presente trabajo se ha comparado la expresión génica entre dichos grupos: SFI-NIT, ELI-NIT y ELI-SFI.

Así, este experimento sería de tipo comparativo, donde se compara por pares la expresión génica entre las 3 categorías (niveles) del grupo "Group". Dentro de cada grupo tendríamos n=10 réplicas procedentes de tejido de distintos sujetos que han sido asignadas aleatoriamente desde el fichero inicial que contenía datos de más pacientes.

4. MÉTODOS Y RESULTADOS:

El software usado para el análisis de los datos ha sido R v.4.0.0. para Windows. Para ello, se han requerido distintos paquetes tanto de R como Bioconductor, recogidos al principio del Anexo: Código R utilizado. Asimismo, debido a los problemas con el paquete ClusterProfiler, se ha decidido realizar el análisis de enriquecimiento utilizando el recurso online DAVID v.6.8 (<https://david.ncifcrf.gov/>).

En el presente trabajo, se ha decidido agrupar los apartados "métodos y resultados" para explicar mejor el proceso de obtención de los resultados. Así, se recoge cómo se ha hecho en cada uno de los pasos del análisis y qué información se ha obtenido en cada caso.

4. 1. Preparación de los datos

Para escoger 30 muestras del dataset al azar (10 de cada categoría, NIT, ELI y SFI), lo primero que se ha hecho es cargar el archivo targets.csv, cuya apariencia se muestra en la Figura 1.

Experiment	SRA_Sample	Sample_Name	Grupo_analisis	body_site	molecular_data_type	sex	Group	ShortName
SRX567480	SR5626942	GTEX-111CU-0226-SM-5GZXC	1	Thyroid	Allele-Specific Expression	male	NIT	111CU_NIT
SRX615964	SR5644174	GTEX-111FC-1026-SM-5GZX1	1	Thyroid	RNA Seq (NGS)	male	NIT	111FC_NIT
SRX563960	SR5625636	GTEX-111VG-0526-SM-5N9BW	3	Thyroid	RNA Seq (NGS)	male	ELI	111VG_ELI
SRX564185	SR5625665	GTEX-111YS-0726-SM-5GZY8	1	Thyroid	Allele-Specific Expression	male	NIT	111YS_NIT
SRX559141	SR5624025	GTEX-1122O-0226-SM-5N9DA	1	Thyroid	RNA Seq (NGS)	female	NIT	1122O_NIT
SRX561718	SR5625313	GTEX-1128S-0126-SM-5H125	1	Thyroid	Allele-Specific Expression	female	NIT	1128S_NIT
SRX588467	SR5633874	GTEX-113JC-0126-SM-5EGJW	1	Thyroid	RNA Seq (NGS)	female	NIT	113JC_NIT
SRX634479	SR5648886	GTEX-117XS-0526-SM-5987Q	1	Thyroid	Allele-Specific Expression	male	NIT	117XS_NIT
SRX557750	SR5623875	GTEX-117YW-0126-SM-5EGCN	2	Thyroid	RNA Seq (NGS)	male	SFI	117YW_SFI
SRX580666	SR5629989	GTEX-117YX-1226-SM-5H115	1	Thyroid	Allele-Specific Expression	male	NIT	117YX_NIT

Figura 1. Contenido del archivo targets.csv.

De este archivo, seleccionamos 30 muestras, n=10 NIT, n=10 SFI y n=10 ELI, de forma aleatoria. Estas serán las que estudiaremos su expresión. A la hora de trabajar con sus datos, hemos creado otro fichero, targets_30, cuyo contenido se muestra en la Figura 2.

Experiment	SRA_Sample	Sample_Name	Grupo_analisis	body_site	molecular_data_type	sex	Group	ShortName
SRX405750	SR5524284	GTEX-XBEW-0126-SM-4AT66	1	Thyroid	Allele-Specific Expression	male	NIT	XBEW_NIT
SRX640007	SR5650067	GTEX-YEC3-0826-SM-4WWFP	1	Thyroid	Allele-Specific Expression	male	NIT	YEC3_NIT
SRX564506	SR5625717	GTEX-Y3IK-0526-SM-4WWE3	1	Thyroid	Allele-Specific Expression	female	NIT	Y3IK_NIT
SRX567491	SR5626943	GTEX-ZVZP-1026-SM-5CICI	1	Thyroid	RNA Seq (NGS)	male	NIT	ZVZP_NIT
SRX605094	SR5639215	GTEX-133LE-0326-SM-5P9G4	1	Thyroid	RNA Seq (NGS)	female	NIT	133LE_NIT
SRX198142	SR5333268	GTEX-P4QT-2626-SM-2I3FM	1	Thyroid	Allele-Specific Expression	female	NIT	P4QT_NIT
SRX634070	SR5648837	GTEX-146FR-0326-SM-5S18U	1	Thyroid	Allele-Specific Expression	female	NIT	146FR_NIT
SRX222293	SR5389583	GTEX-T6MN-0626-SM-32PM9	1	Thyroid	Allele-Specific Expression	male	NIT	T6MN_NIT
SRX632747	SR5648687	GTEX-146FO-0726-SM-5LU47	1	Thyroid	Allele-Specific Expression	male	NIT	146FO_NIT
SRX579411	SR5629803	GTEX-11GSO-0626-SM-5A5LW	1	Thyroid	RNA Seq (NGS)	male	NIT	11GSO_NIT

1-10 of 30 rows

Previous 1 2 3 Next

Figura 2. Contenido del archivo targets_30.

Por otro lado, hemos seleccionado a mano los counts de las 30 muestras seleccionadas al azar y creado un nuevo archivo a partir del fichero counts.csv llamado counts_30.csv (Figura 3). En este fichero, las muestras han sido ordenadas según aparecían en "targets_30".

	GTEX.XBEW.0126.SM.4AT66	GTEX.YEC3.0826.SM.4WWFP	GTEX.Y3IK.0526.SM.4WWE3	GTEX.ZVZP.1026.SM.5CICI	GTEX.133LE.0326.SM.5P9G4
ENSG00000223972.4	1	4	5	5	4
ENSG00000227232.4	473	705	1372	529	1175
ENSG00000243485.2	1	2	1	2	2
ENSG00000237613.2	2	0	1	1	3
ENSG00000268020.2	1	1	1	1	2
ENSG00000240361.1	1	1	2	1	0
ENSG00000186092.4	3	15	1	2	4
ENSG00000238009.2	5	8	6	6	4
ENSG00000233750.3	17	26	20	17	37
ENSG00000237683.5	1071	272	571	391	2739

1-10 of 56,202 rows | 1-6 of 31 columns

Previous 1 2 3 4 5 6 ... 100 Next

Figura 3. Contenido del archivo counts_30.csv.

A continuación, creamos un objeto de clase DESeqDataSetMatrix (Figura 4) con los datos de expresión de las 30 muestras que han sido escogidas al azar.

```
some variables in design formula are characters, converting to factors
class: DESeqDataSet
dim: 56202 30
metadata(1): version
assays(1): counts
rownames(56202): ENSG00000223972 ENSG00000227232 ... ENSG00000210195 ENSG00000210196
rowData names(0):
colnames(30): GTEX.XBEW.0126.SM.4AT66 GTEX.YEC3.0826.SM.4WWFP ... GTEX.11NV4.0626.SM.5N9BR GTEX.PLZ4.1226.SM.2I5FE
colData names(9): Experiment SRA_Sample ... Group ShortName
```

Figura 4. DESeqDataSetMatrix con datos de las 30 muestras incluidas en el estudio.

4. 2. Preprocesado de los datos: filtraje y normalización

Lo primero que hacemos es prefiltrar: eliminamos los genes que tienen cero o bajo número de reads. Así, de 56202 genes que teníamos nos quedamos con 43507.

A continuación, vamos a hacer dos transformaciones: transformación estabilizadora de la varianza (vst) y transformación logarítmica regularizada (rlog).

Y el efecto de las transformaciones lo veremos en el gráfico que aparece a continuación (Figura 5). Hemos usado la transformación log2 de los "counts" normalizados.

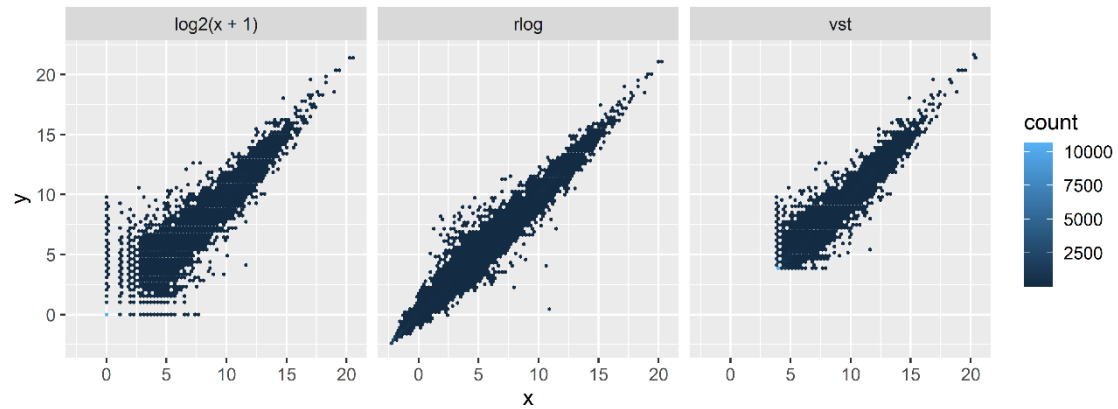


Figura 5. Efectos de la transformación \log_2 de los "counts" normalizados cuando se compara con la transformación logarítmica regularizada (rlog) y estabilizadora de la varianza (vst).

A continuación, se ha estudiado la distancias entre las muestras incluidas en el estudio. Para ver mejor la similitud entre muestras hacemos una matriz de distancias con las 30 muestras y la representamos con un heatmap (Figura 6).

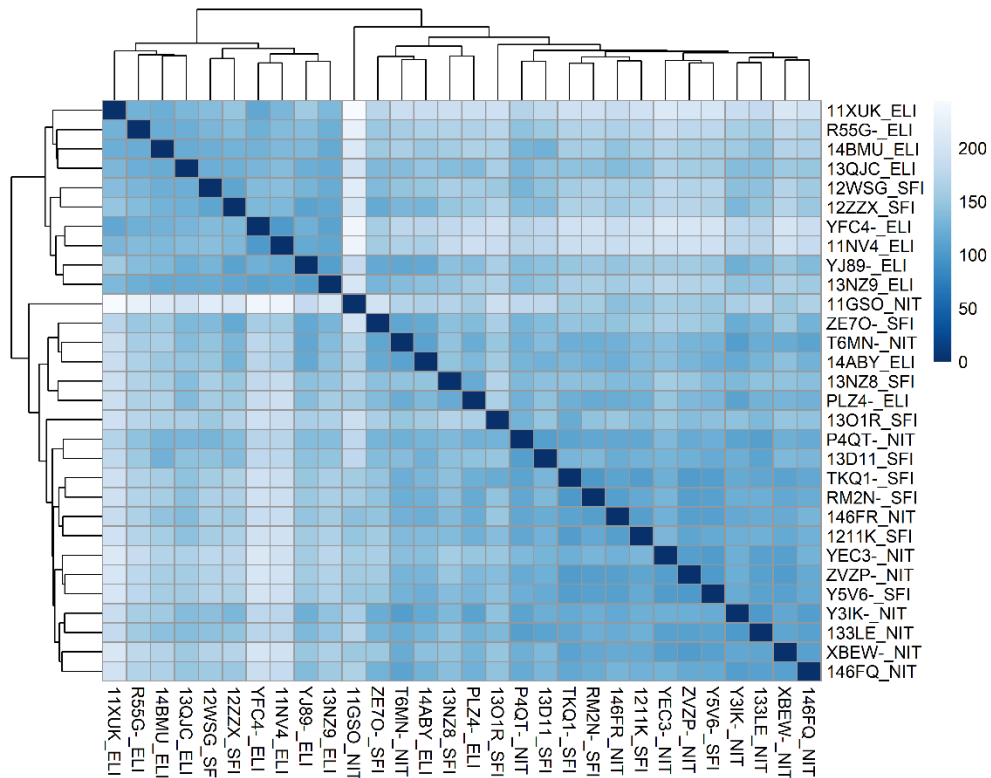


Figura 6. Heatmap con las distancias entre las muestras estudiadas.

Como era de esperar, la menor distancia entre muestras (0, azul más oscuro) es cuando enfrentamos a una muestra consigo misma.

Asimismo, hemos hecho un Análisis de Componentes Principales (PCA) con los datos de vst para ver cómo se agrupan las 30 muestras (Figura 7).

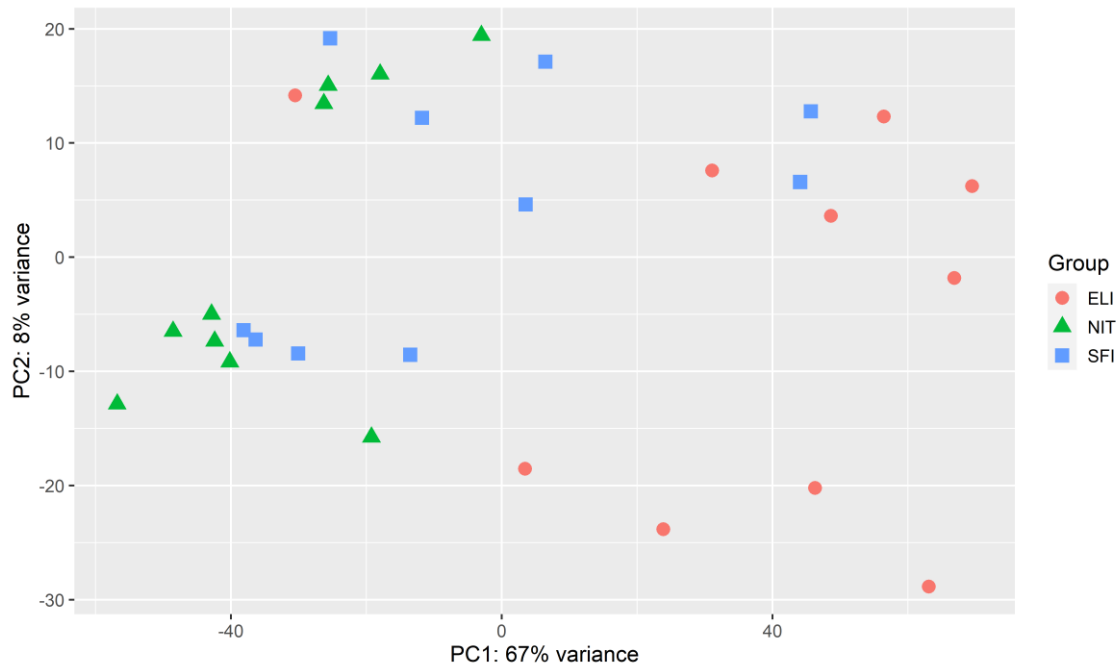


Figura 7. Análisis de componentes principales con los datos transformados (vst).

El primer componente, PC1, acumula el 67% de la varianza, mientras que el Segundo componente, PC2, acumula el 8% de la varianza de los datos normalizados (vst). Lo que podemos observar en primer lugar es que las muestras de ELI se agrupan claramente en sentido positivo del PC1, separándose de las dos categorías restantes. La agrupación de las otras dos categorías, SFI y NIT, no es tan clara, aunque sí parece que la expresión génica de las muestras de NIT difiere más de las de ELI que las de SFI.

Asimismo, hemos representando también las muestras con un MDS plot usando los datos de la matriz de distancias (Figura 8).

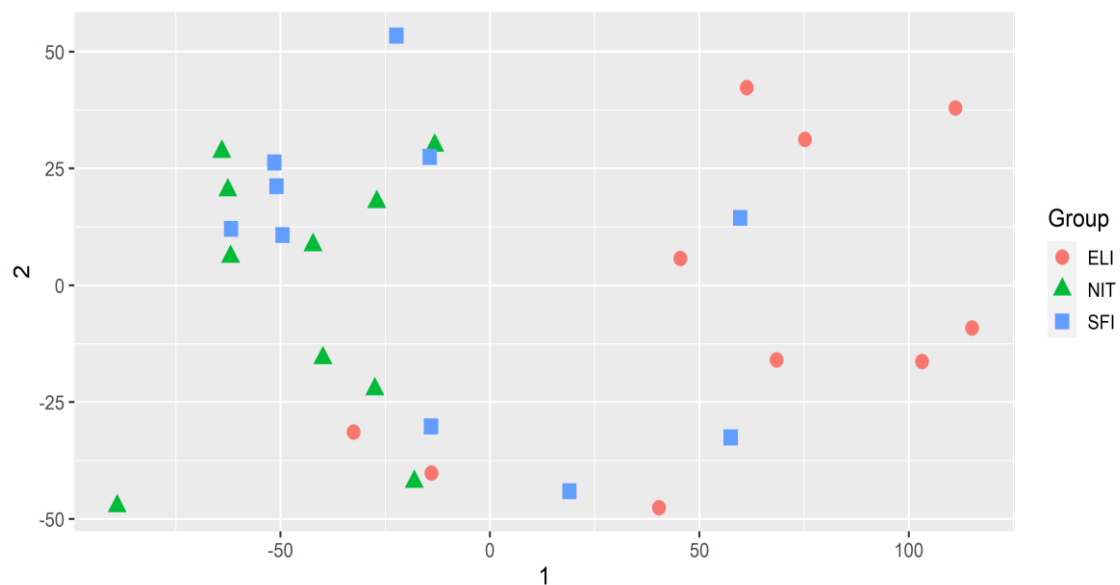


Figura 8. MDS plot con los datos de distancias entre muestras.

Como podemos observar en el gráfico, siguen las muestras del grupo ELI separándose de las de NIT y SFI (que solapan mucho), no obstante, esta separación es menos evidente que en el PCA.

Por tanto, en nuestro caso es más recomendable usar el PCA a la hora de explorar nuestros datos y buscar similitudes en la expresión génica de las distintas muestras.

Finalmente, se ha utilizado el paquete sva para eliminar los posibles efectos del Batch. Y utilizado el paquete RUVSeq para eliminar de nuestros datos de RNAseq la variación no deseada.

4.3. Identificación de genes diferencialmente expresados

Lo primero que hacemos es obtener la tabla de results de cada una de las tres comparaciones usando DESeq2 (Figuras 9, 10 y 11). Cada "result" contiene la información: baseMean, log2FoldChange, lfcSE, stat, pvalue y padj.

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

out of 43507 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)      : 433, 1%
LFC < 0 (down)    : 139, 0.32%
outliers [1]      : 0, 0%
low counts [2]    : 17714, 41%
(mean count < 3)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Figura 9. Tabla "results" de la comparación SFI versus NIT.

Vemos que en esta comparación hay 433 genes upregulados poniendo como punto de corte un p.adjusted value de 0.1 y 139 genes downregulados al mismo nivel de significación.

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

out of 43507 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)      : 1032, 2.4%
LFC < 0 (down)    : 2373, 5.5%
outliers [1]      : 0, 0%
low counts [2]    : 14340, 33%
(mean count < 2)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results
```

Figura 10. Tabla "results" de la comparación SFI versus ELI.

Vemos que en esta comparación hay 1032 genes upregulados poniendo como punto de corte un p.adjusted value de 0.1 y 2373 genes downregulados al mismo nivel de significación.

```

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

out of 43507 with nonzero total read count
adjusted p-value < 0.1
LFC > 0 (up)      : 2180, 5%
LFC < 0 (down)    : 4141, 9.5%
outliers [1]      : 0, 0%
low counts [2]    : 11809, 27%
(mean count < 1)
[1] see 'cooksCutoff' argument of ?results
[2] see 'independentFiltering' argument of ?results

```

Figura 11. Tabla “results” de la comparación NIT versus ELI.

Vemos que en esta comparación hay 2180 genes upregulados poniendo como punto de corte un p.adjusted value de 0.1 y 4141 genes downregulados al mismo nivel de significación.

A continuación, hacemos las anotaciones de los resultados para poder saber a qué gen corresponde cada valor de los mencionados utilizando los paquetes de Bioconductor: AnnotationDbi y EnsDb.Hsapiens.v86.

Ahora vamos a proceder a estudiar los genes más downregulados y upregulados en cada una de las tres comparaciones usando como criterio el p-adjusted value (padj) a un nivel de significación del 0.1 y ordenando los genes de acuerdo al valor de log2 fold change.

- SFI versus NIT

Genes más downregulados (Figura 12) y más upregulados de esta comparación (Figura 13):

```

log2 fold change (MLE): Group SFI vs NIT
Wald test p-value: Group SFI vs NIT
DataFrame with 6 rows and 7 columns
      baseMean log2FoldChange      lfcSE      stat      pvalue      padj      symbol
<numeric>    <numeric> <numeric> <numeric> <numeric> <numeric> <character>
ENSG00000206192    3.73139    -4.31292  1.165424 -3.70073  2.14982e-04  0.02594531 ANKRD20A9P
ENSG00000238245    14.06260    -4.11703  1.105814 -3.72308  1.96810e-04  0.02464231 MYO5BP2
ENSG00000179031    14.14204    -4.01829  0.878716 -4.57291  4.80992e-06  0.00204389 LLOXNC01-131B10.2
ENSG00000227195     4.40263    -3.40124  1.034478 -3.28788  1.00946e-03  0.06472606 MIR663AHG
ENSG00000132972    12.45005    -3.37806  0.795244 -4.24783  2.15852e-05  0.00518524 RNF17
ENSG00000182489   138.19044    -2.77512  0.645920 -4.29639  1.73605e-05  0.00447778 XKRX

```

Figura 12. Top genes downregulados entre SFI y NIT. En la columna “symbol” aparece el nombre de cada gen, mientras que en las columnas restantes podemos los distintos parámetros de “results”.

```

log2 fold change (MLE): Group SFI vs NIT
Wald test p-value: Group SFI vs NIT
DataFrame with 6 rows and 7 columns
      baseMean log2FoldChange      lfcSE      stat      pvalue      padj      symbol
<numeric>    <numeric> <numeric> <numeric> <numeric> <numeric> <character>
ENSG00000235896    71.5273     6.89698  1.49908  4.60082  4.20834e-06  1.93832e-03 NA
ENSG00000254709   121.2319     6.86538  1.22639  5.59806  2.16767e-08  9.78839e-05 IGLL5
ENSG00000225523    36.9674     6.77961  1.41561  4.78919  1.67456e-06  1.19310e-03 IGKV6D-21
ENSG00000211619    85.6582     6.75833  1.56359  4.32231  1.54407e-05  4.14857e-03 NA
ENSG00000242534    56.6406     6.53579  1.40302  4.65839  3.18694e-06  1.67756e-03 IGKV2D-28
ENSG00000211930    11.7811     6.07342  1.28104  4.74100  2.12668e-06  1.21896e-03 IGH3D-3

```

Figura 13. Top genes upregulados entre SFI y NIT. En la columna “symbol” aparece el nombre de cada gen, mientras que en las columnas restantes podemos los distintos parámetros de “results”.

- SFI versus ELI

Genes más downregulados (Figura 14) y más upregulados de esta comparación (Figura 15):

```
log2 fold change (MLE): Group SFI vs ELI
Wald test p-value: Group SFI vs ELI
DataFrame with 6 rows and 7 columns
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<character>
ENSG00000170054	56.4815	-7.63803	1.225518	-6.23249	4.59082e-10	2.15968e-07	SERPINA9
ENSG00000162897	73.4408	-7.43446	1.374219	-5.40995	6.30406e-08	1.10765e-05	FCAMR
ENSG00000100721	419.5701	-7.30202	0.741863	-9.84282	7.36209e-23	2.14730e-18	TCL1A
ENSG00000260303	22.3110	-7.23547	1.289081	-5.61289	1.98973e-08	4.23609e-06	RP11-203B7.2
ENSG00000181617	376.9526	-6.97258	1.530487	-4.55579	5.21884e-06	3.71263e-04	FDCSP
ENSG00000213231	7.1569	-6.77969	1.264302	-5.36240	8.21247e-08	1.38399e-05	TCL1B

Figura 14. Top genes downregulados entre SFI y ELI. En la columna “symbol” aparece el nombre de cada gen, mientras que en las columnas restantes podemos los distintos parámetros de “results”.

```
log2 fold change (MLE): Group SFI vs ELI
Wald test p-value: Group SFI vs ELI
DataFrame with 6 rows and 7 columns
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<character>
ENSG00000110680	5301.0646	10.01503	1.293628	7.74182	9.80030e-15	4.90650e-11	CALCA
ENSG00000134443	60.2432	9.31088	1.782182	5.22443	1.74695e-07	2.53498e-05	GRP
ENSG00000100604	254.1483	5.82009	0.950454	6.12349	9.15486e-10	3.56026e-07	CHGA
ENSG00000157005	17.4636	5.06488	0.950109	5.33084	9.77614e-08	1.59296e-05	SST
ENSG00000128564	64.6900	4.79649	0.771381	6.21806	5.03354e-10	2.28209e-07	VGf
ENSG00000105388	54.0163	4.40205	1.067834	4.12241	3.74930e-05	1.79862e-03	CEACAM5

Figura 15. Top genes upregulados entre SFI y ELI. En la columna “symbol” aparece el nombre de cada gen, mientras que en las columnas restantes podemos los distintos parámetros de “results”.

- NIT versus ELI

Genes más downregulados (Figura 16) y más upregulados de esta comparación (Figura 17):

```
log2 fold change (MLE): Group NIT vs ELI
Wald test p-value: Group NIT vs ELI
DataFrame with 6 rows and 7 columns
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<character>
ENSG00000100721	419.5701	-8.90411	0.774675	-11.49399	1.41431e-30	4.48309e-26	TCL1A
ENSG00000211619	85.6582	-8.80505	1.562537	-5.63510	1.74958e-08	1.10255e-06	NA
ENSG00000254709	121.2319	-8.68505	1.225596	-7.08639	1.37656e-12	3.18498e-10	IGLL5
ENSG00000163518	31.2660	-8.45334	0.923175	-9.15682	5.34505e-20	8.91723e-17	FCRL4
ENSG00000257275	30.6219	-8.43458	1.044623	-8.07428	6.78736e-16	4.21854e-13	RP11-164H13.1
ENSG00000254029	21.7654	-8.37490	2.200776	-3.80543	1.41557e-04	2.40594e-03	IGLC4

Figura 16. Top genes downregulados entre NIT y ELI. En la columna “symbol” aparece el nombre de cada gen, mientras que en las columnas restantes podemos los distintos parámetros de “results”.

```
log2 fold change (MLE): Group NIT vs ELI
Wald test p-value: Group NIT vs ELI
DataFrame with 6 rows and 7 columns
```

	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj	symbol
	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<numeric>	<character>
ENSG00000110680	5301.06461	5.08515	1.29378	3.93046	8.47843e-05	0.00158554	CALCA
ENSG00000244155	3.39625	4.59798	1.68780	2.72425	6.44470e-03	0.04474024	CYP4F34P
ENSG00000134443	60.24323	4.55345	1.79075	2.54276	1.09982e-02	0.06571538	GRP
ENSG00000233491	4.12512	4.46261	1.59238	2.80247	5.07128e-03	0.03764623	AC010091.1
ENSG00000164796	4.70990	4.18576	1.15233	3.63242	2.80776e-04	0.00418774	CSMD3
ENSG00000243961	3.11298	4.01195	1.57893	2.54093	1.10559e-02	0.06597567	RP5-839B4.8

Figura 17. Top genes upregulados entre NIT y ELI. En la columna “symbol” aparece el nombre de cada gen, mientras que en las columnas restantes podemos los distintos parámetros de “results”.

Finalmente, vamos a representar gráficamente los resultados en MAplots (Figuras 18, 19 y 20). En rojo, podemos observar los genes significativamente más diferencialmente expresados entre cada una de las comparaciones.

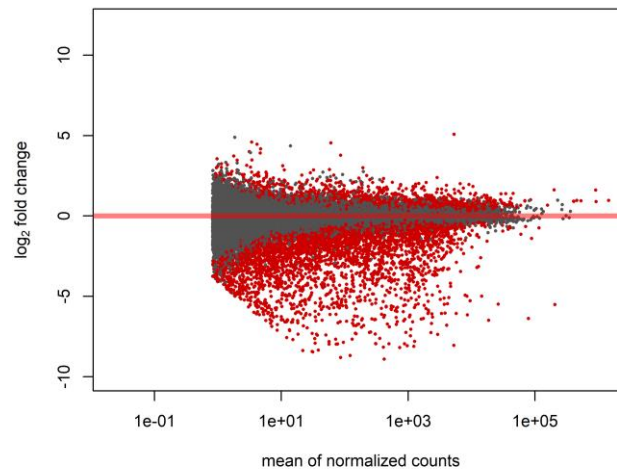


Figura 18. MA plot con el log2 fold change de los genes diferencialmente expresados entre SFI versus NIT. En rojo, genes con diferencias de expresión significativas.

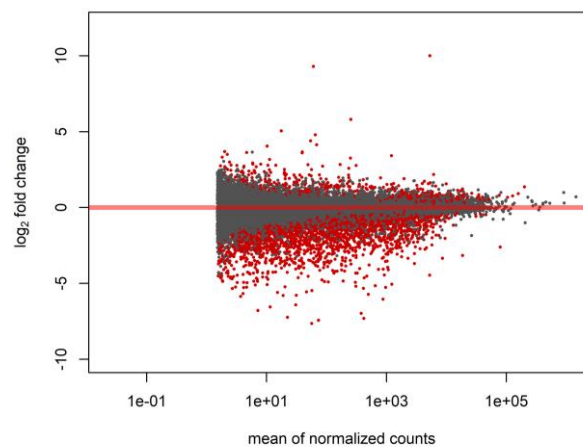


Figura 19. MA plot con el log2 fold change de los genes diferencialmente expresados entre SFI versus ELI. En rojo, genes con diferencias de expresión significativas.

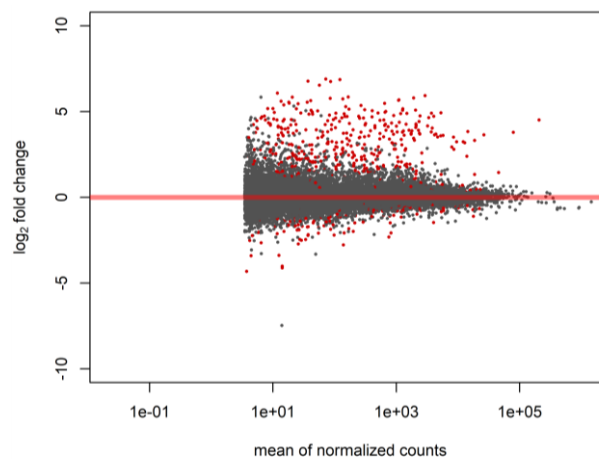


Figura 20. MA plot con el log2 fold change de los genes diferencialmente expresados entre NIT versus ELI. En rojo, genes con diferencias de expresión significativas.

4.4. Agrupación de los genes más diferencialmente expresados

Lo primero que hacemos es seleccionar los 20 genes con las varianzas más altas entre las 30 muestras. A continuación representaremos la expresión de estos genes mediante un heatmap (Figura 21) para ver cómo se comportan entre las diferentes muestras. Asimismo, se ha hecho un clustering tanto de los genes (filas) como de las muestras (columnas).

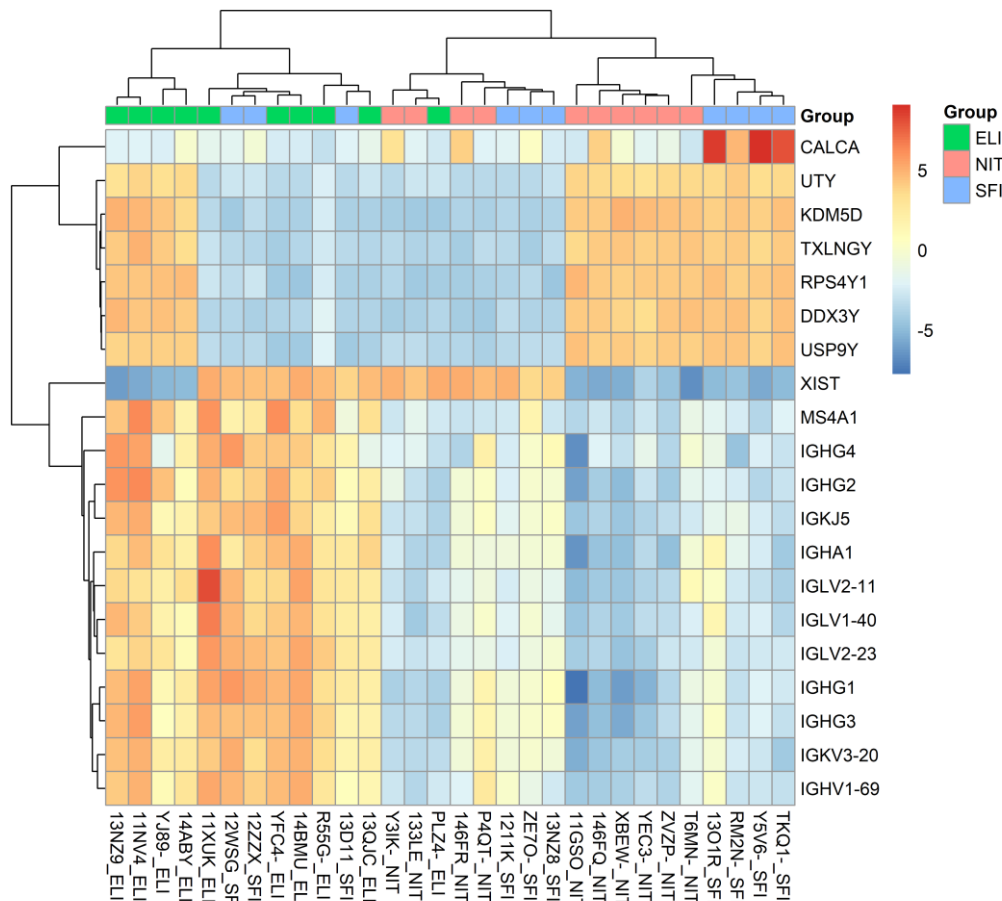


Figura 21. Heatmap con los top 20 genes de mayor varianza.

Vemos que hay genes que presentan varianzas muy altas en alguna de las muestras, como es el caso de *CALCA* en varias muestras de SFI, o *IGHG4*, *IGHG2*, *IGHG3* e *IGH4* entre otros en varias muestras de NIT. También se observan ciertos "comportamientos" en las varianzas de los genes entre las distintas muestras que permiten agruparlas por su varianza, aunque de forma general resulta bastante complicado de interpretar.

4.5. Anotación de genes y exportación de resultados

El siguiente paso es anotar los genes que nos han salido diferencialmente expresados utilizando el paquete de Bioconductor para el genoma humano "org.Hs.eg.db" como referencia. Una vez realizada la anotación podremos realizar el análisis de enriquecimiento de genes y rutas metabólicas. Los genes diferencialmente expresados junto a sus datos de expresión (results) de cada una de las dos comparaciones se han exportado a un archivo .csv.

Para un mayor detalle de cómo se ha hecho la anotación ver el Anexo: Código de R utilizado.

4.6. Estudio de enriquecimiento de genes (GO) y rutas metabólicas (KEGG)

El análisis de enriquecimiento se intentó realizar con el paquete de Bioconductor “clusterProfiler”. A continuación, en las Figuras 22 a 25 se muestran los resultados obtenidos tanto para el enriquecimiento de genes implicados en funciones concretas (GO) como de rutas metabólicas (KEGG) en la comparación SFI versus NIT. El problema, que no he sabido identificar, es que en las 3 comparaciones obtengo los mismos resultados.

ID ~CHP~	Description	GeneRatio ~CHP~	BgRatio ~CHP~	pvalue ~GO~	p.adjust ~GO~	qvalue ~GO~
GO:0002446	neutrophil mediated immunity	494/16854	499/18670	1.493469e-16	5.776905e-13	4.003702e-13
GO:0044282	small molecule catabolic process	442/16854	445/18670	1.967275e-16	5.776905e-13	4.003702e-13
GO:0007409	axonogenesis	464/16854	468/18670	2.686100e-16	5.776905e-13	4.003702e-13
GO:0002283	neutrophil activation involved in immune response	483/16854	488/18670	4.245706e-16	6.848323e-13	4.746252e-13
GO:0043312	neutrophil degranulation	480/16854	485/18670	5.642380e-16	7.280927e-13	5.046070e-13
GO:0042119	neutrophil activation	492/16854	498/18670	1.524152e-15	1.638971e-12	1.135894e-12

Figura 22. Listado de funciones biológicas (GO) que se encuentran enriquecidas en la comparación SFI versus NIT.

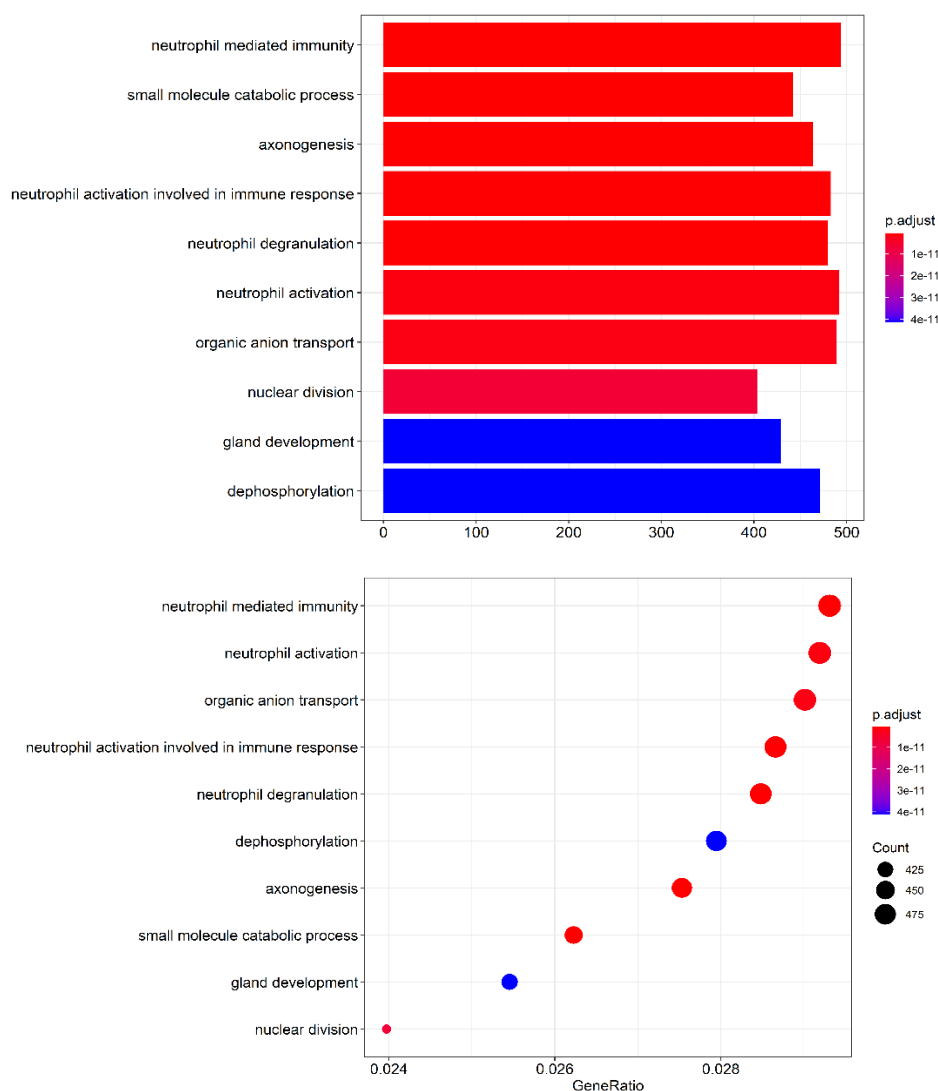


Figura 23. Distintos plots donde se pueden observar las funciones biológicas (GO) que se encuentran enriquecidas en la comparación SFI versus NIT.

ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
hsa05010	Alzheimer disease	365/7514	369/8031	2.864666e-07	8.552847e-05	5.128642e-05
hsa04010	MAPK signaling pathway	292/7514	294/8031	5.295880e-07	8.552847e-05	5.128642e-05
hsa05016	Huntington disease	303/7514	306/8031	1.900518e-06	1.880593e-04	1.127682e-04
hsa05165	Human papillomavirus infection	326/7514	330/8031	2.765005e-06	1.880593e-04	1.127682e-04
hsa04714	Thermogenesis	230/7514	231/8031	2.911135e-06	1.880593e-04	1.127682e-04
hsa04151	PI3K-Akt signaling pathway	349/7514	354/8031	3.622454e-06	1.950088e-04	1.169354e-04

Figura 24. Listado de rutas metabólicas (KEGG) que se encuentran enriquecidas en la comparación SFI versus NIT.

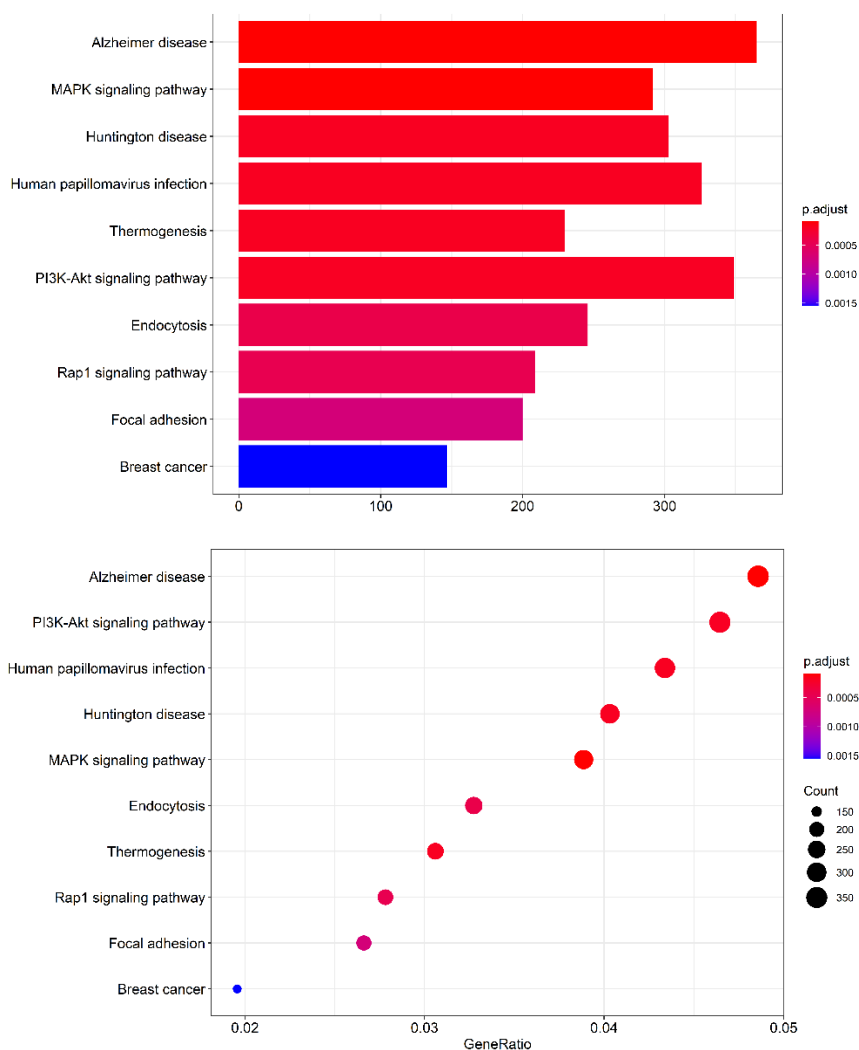


Figura 25. Distintos plots donde se pueden observar las rutas metabólicas (KEGG) que se encuentran enriquecidas en la comparación SFI versus NIT.

Dado este problema, se ha decidido realizar el estudio de enriquecimiento para las tres comparaciones utilizando la herramienta DAVID v.6.8. La elección de esta plataforma es por que, de acuerdo a Geistlinger et al. (2020) es la herramienta para este análisis más utilizada en la bibliografía.

Para ello, se ha cogido la lista de los top100 genes diferencialmente expresados entre cada comparación y se ha introducido en DAVID de la forma que aparece en la Figura 26, seleccionando los parámetros que se muestran.

Figura 25. Introducción de la lista de genes y parámetros seleccionados en DAVID.

Esta plataforma te permite dividir el estudio de enriquecimiento (GO) en las tres categorías existentes: GO BP (procesos biológicos), GO CC (componentes celulares) y GO MF (funciones moleculares) además de ofrecerte el análisis de las rutas (KEGG). Así, a continuación se presenta toda esta información para cada una de las tres comparaciones.

- SFI versus NIT

En la Figura 26 vemos el top 10 de GO procesos biológicos que están enriquecidos cuando comparamos los datos procedentes de muestras SFI contra los de NIT. Entre ellas destacan aquellas relacionadas con la respuesta inmune y en concreto con la activación y proliferación de linfocitos B.

Term	RT	Genes	Count	%	P-Value	Benjamini
adaptive immune response	RT		4	0,1	3,0E-3	5,8E-1
response to insulin	RT		3	0,1	7,6E-3	6,7E-1
cellular heat acclimation	RT		2	0,0	7,8E-3	5,3E-1
protein refolding	RT		2	0,0	2,9E-2	8,8E-1
B cell activation	RT		2	0,0	5,5E-2	9,6E-1
insulin secretion	RT		2	0,0	5,9E-2	9,5E-1
B cell proliferation	RT		2	0,0	6,1E-2	9,3E-1
cellular response to heat	RT		2	0,0	7,0E-2	9,3E-1
response to unfolded protein	RT		2	0,0	7,9E-2	9,3E-1
response to cAMP	RT		2	0,0	8,7E-2	9,3E-1
cation transmembrane transport	RT		2	0,0	9,0E-2	9,2E-1

Figura 26. Top 10 GO procesos biológicos enriquecidos en SFI versus NIT.

Y en la Figura 27 los componentes celulares -entre los que destaca la membrana-, mientras que en la Figura 28 se muestran las funciones moleculares enriquecidas cuando comparamos SFI versus NIT. No aparecen diez porque no fueron obtenidos tantos como en el GO BP.






Term	RT	Genes	Count	%	P-Value	Benjamini
secretory granule	RT		4	0,1	4,5E-4	3,1E-2
external side of plasma membrane	RT		4	0,1	9,1E-3	2,7E-1
blood microparticle	RT		3	0,1	3,8E-2	5,9E-1
membrane raft	RT		3	0,1	6,5E-2	6,9E-1
transport vesicle membrane	RT		2	0,0	7,4E-2	6,6E-1

Figura 27. Top 5 GO componentes celulares enriquecidos en SFI versus NIT.




Term	RT	Genes	Count	%	P-Value	Benjamini
ATPase activity, coupled	RT		2	0,0	2,1E-2	8,4E-1
heat shock protein binding	RT		2	0,0	7,2E-2	9,6E-1
SH3/SH2 adaptor activity	RT		2	0,0	9,5E-2	9,4E-1

Figura 28. Top 3 GO funciones moleculares enriquecidas en SFI versus NIT.

Finalmente, en la Figura 29 vemos la ruta metabólica enriquecida en las muestras SFI cuando la comparamos con NIT, en este caso hablamos de genes implicados en la inmunodeficiencia primaria.


Term	RT	Genes	Count	%	P-Value	Benjamini
Primary immunodeficiency	RT		2	0,0	7,2E-2	8,9E-1

Figura 29. Top rutas metabólicas enriquecidas en SFI versus NIT.

- SFI versus ELI

En la Figura 30 vemos el top 10 de GO procesos biológicos que están enriquecidos cuando comparamos los datos procedentes de muestras SFI contra los de ELI. Entre ellas destacan aquellos relacionados con la respuesta inmune y la proliferación celular.










Term	RT	Genes	Count	%	P-Value	Benjamini
cell division	RT		11	0,1	3,8E-6	1,9E-3
mitotic nuclear division	RT		9	0,1	1,6E-5	3,9E-3
immune response	RT		11	0,1	1,9E-5	3,1E-3
sister chromatid cohesion	RT		6	0,1	1,0E-4	1,3E-2
mitotic sister chromatid segregation	RT		4	0,0	1,9E-4	1,9E-2
chromosome segregation	RT		5	0,0	2,5E-4	2,1E-2
protein localization to kinetochore	RT		3	0,0	8,9E-4	6,0E-2
B cell receptor signaling pathway	RT		4	0,0	1,9E-3	1,1E-1
adaptive immune response	RT		5	0,0	4,5E-3	2,2E-1
cell differentiation	RT		8	0,1	4,9E-3	2,1E-1

Figura 30. Top 10 GO procesos biológicos enriquecidos en SFI versus ELI.

Y en la Figura 31 los componentes celulares -entre los que destacan aquellos implicados en la división-, mientras que en la Figura 32 se muestran las funciones moleculares enriquecidas cuando comparamos SFI versus ELI. Aquí vemos cómo también destacan las funciones relacionadas con transcripción y con la división celular, en concreto con los microtúbulos.

Term	RT	Genes	Count	%	P-Value	Benjamini
midbody	RT		7	0,1	1,9E-5	2,2E-3
chromosome, centromeric region	RT		5	0,0	1,0E-4	6,0E-3
spindle	RT		6	0,1	1,7E-4	6,6E-3
condensed chromosome kinetochore	RT		5	0,0	5,3E-4	1,5E-2
integral component of plasma membrane	RT		15	0,1	2,5E-3	5,7E-2
spindle midzone	RT		3	0,0	2,9E-3	5,6E-2
kinetochore	RT		4	0,0	5,1E-3	8,1E-2
microtubule	RT		6	0,1	1,1E-2	1,4E-1
chromosome passenger complex	RT		2	0,0	2,1E-2	2,4E-1
kinesin complex	RT		3	0,0	2,2E-2	2,2E-1

Figura 31. Top 10 GO componentes celulares enriquecidos en SFI versus ELI.

Term	RT	Genes	Count	%	P-Value	Benjamini
ATP binding	RT		16	0,1	1,2E-3	1,4E-1
microtubule motor activity	RT		4	0,0	4,6E-3	2,6E-1
protein binding	RT		49	0,4	5,6E-3	2,2E-1
transmembrane signaling receptor activity	RT		5	0,0	1,3E-2	3,4E-1
RNA polymerase II core promoter proximal region sequence-specific DNA binding	RT		6	0,1	1,7E-2	3,6E-1
carbohydrate binding	RT		4	0,0	5,0E-2	6,7E-1
microtubule binding	RT		4	0,0	5,7E-2	6,7E-1
receptor binding	RT		5	0,0	6,1E-2	6,5E-1
protein kinase activity	RT		5	0,0	6,4E-2	6,2E-1
protein serine/threonine kinase activity	RT		5	0,0	7,4E-2	6,3E-1

Figura 32. Top 10 GO funciones moleculares enriquecidas en SFI versus ELI.

Finalmente, en la Figura 33 vemos las rutas metabólicas enriquecidas en las muestras SFI cuando la comparamos con ELI, en este caso hablamos de genes implicados en la vía de señalización de los receptores de los linfocitos B y en la interacción de citoquinas con sus receptores entre otras.

Term	RT	Genes	Count	%	P-Value	Benjamini
B cell receptor signaling pathway	RT		6	0,1	7,2E-6	3,1E-4
Hematopoietic cell lineage	RT		5	0,0	3,9E-4	8,3E-3
Cytokine-cytokine receptor interaction	RT		4	0,0	7,5E-2	6,7E-1
Epstein-Barr virus infection	RT		3	0,0	8,7E-2	6,3E-1

Figura 33. Top rutas metabólicas enriquecidas en SFI versus ELI.

- NIT versus ELI

En la Figura 34 vemos el top 10 de GO procesos biológicos que están enriquecidos cuando comparamos los datos procedentes de muestras NIT contra los de ELI. Entre ellas destacan aquellos relacionados con la respuesta inmune, los linfocitos B y T y la respuesta inflamatoria.

Term	RT	Genes	Count	%	P-Value	Benjamini
immune response	RT		12	0,1	2,0E-7	5,6E-5
adaptive immune response	RT		6	0,1	1,7E-4	2,3E-2
B cell receptor signaling pathway	RT		4	0,0	9,0E-4	7,9E-2
humoral immune response	RT		4	0,0	1,1E-3	7,0E-2
B cell proliferation	RT		3	0,0	5,6E-3	2,7E-1
transcription from RNA polymerase II promoter	RT		7	0,1	9,0E-3	3,4E-1
inflammatory response	RT		6	0,1	1,1E-2	3,4E-1
cell surface receptor signaling pathway	RT		5	0,1	1,6E-2	4,2E-1
innate immune response	RT		6	0,1	1,7E-2	4,1E-1
T-helper 17 cell lineage commitment	RT		2	0,0	1,7E-2	3,8E-1

Figura 34. Top 10 GO procesos biológicos enriquecidos en NIT versus ELI.

Y en la Figura 35 los componentes celulares -entre los que destacan aquellos de la membrana plasmática y de los receptores de los linfocitos B-, mientras que en la Figura 36 se muestran las funciones moleculares enriquecidas cuando comparamos NIT versus ELI. Aquí vemos cómo destacan las funciones relacionadas con transcripción y de actividad de los receptores.

Term	RT	Genes	Count	%	P-Value	Benjamini
external side of plasma membrane	RT		7	0,1	1,4E-4	9,9E-3
plasma membrane	RT		28	0,3	1,1E-3	3,9E-2
integral component of plasma membrane	RT		14	0,2	2,0E-3	4,5E-2
extrinsic component of cytoplasmic side of plasma membrane	RT		4	0,0	2,1E-3	3,6E-2
B cell receptor complex	RT		2	0,0	1,1E-2	1,5E-1
integral component of membrane	RT		28	0,3	2,8E-2	2,8E-1

Figura 35. Top GO componentes celulares enriquecidos en SFI versus ELI.

Term	RT	Genes	Count	%	P-Value	Benjamini
transmembrane signaling receptor activity	RT		6	0,1	8,3E-4	7,5E-2
sequence-specific DNA binding	RT		6	0,1	3,2E-2	7,9E-1
receptor binding	RT		5	0,1	3,3E-2	6,5E-1
transcriptional activator activity, RNA polymerase II core promoter proximal region sequence-specific binding	RT		4	0,0	4,7E-2	6,8E-1

Figura 36. Top GO funciones moleculares enriquecidas en NIT versus ELI.

Finalmente, en la Figura 37 vemos las rutas metabólicas enriquecidas en las muestras NIT cuando la comparamos con ELI, son prácticamente las mismas que en la comparación anterior.

Term	RT	Genes	Count	%	P-Value	Benjamini
Hematopoietic cell lineage	RT		5	0,1	2,1E-4	8,5E-3
B cell receptor signaling pathway	RT		4	0,0	1,7E-3	3,4E-2
Primary immunodeficiency	RT		3	0,0	6,1E-3	8,0E-2
Cytokine-cytokine receptor interaction	RT		4	0,0	5,1E-2	4,2E-1
Epstein-Barr virus infection	RT		3	0,0	6,7E-2	4,3E-1

Figura 37. Top rutas metabólicas enriquecidas en NIT versus ELI.

DISCUSIÓN

Una de las principales limitaciones de este estudio ha sido el problema encontrado a la hora de estudiar el enriquecimiento utilizando clusterProfiler. Sin embargo, ha intentado solucionarse usando otra herramienta conocida para dicho fin.

Dentro de los resultados obtenidos, parecen señalar todos a cambios en la respuesta inmune y en concreto aquella relacionada con los linfocitos B y las citoquinas cuando existen infiltrados en el tejido tiroideo. Resulta interesante señalar que en el trabajo de Kuo et al. (2017), donde estudiaron cáncer tiroideo con infiltración de linfocitos, los autores comentaron “Papillary thyroid cancer with tumor-infiltrating lymphocytes is associated with an upregulation of immune response and cytokine production”. Este hecho es muy similar a lo hallado en nuestro trabajo.

Finalmente, señalar que las muestras de ELI (Extensive lymphoid infiltrates), tal y como vemos en la Figura 7 y en los análisis de genes up/down regulados, son las que más se distinguen del resto de grupos.

CONCLUSIÓN

- Cuando se comparan por pares la expresión de genes entre distintos tipos de muestras de tiroides, sin, con poca y con extendida infiltración, se hallan genes diferencialmente expresados que están sobre todo implicados en la respuesta inmune.
- Las muestras de ELI, es decir, aquellas de tejido tiroideo con infiltración extendida, tienen un perfil de expresión génica que difiere en mayor grado de las otras dos categorías (NIT y SFI) que entre las otras combinaciones a la hora de comparar muestras.

BIBLIOGRAFÍA

Geistlinger, L., Csaba, G., Santarelli, M., Ramos, M., Schiffer, L., Law, C., ... & Zimmer, R. (2020). Towards a gold standard for bench marking gene set enrichment analysis. *Briefings in Bioinformatics*, 1-12.

Kuo, C. Y., Liu, T. P., Yang, P. S., & Cheng, S. P. (2017). Characteristics of lymphocyte-infiltrating papillary thyroid cancer. *Journal of Cancer Research and Practice*, 4(3), 95-99.

ANEXO: Código R utilizado

```
library("dplyr")
library("DESeq2")
library("ggplot2")
library("pheatmap")
library("RColorBrewer")
library("sva")
library("RUVSeq")
library("DESeq")
library("AnnotationDbi")
library("EnsDb.Hsapiens.v86")
library("clusterProfiler")
library("org.Hs.eg.db")

### PREPARACIÓN DE LOS DATOS
##### Cargamos los datos de los ficheros "targets" y "counts"
```{r, include = FALSE}
```

```

targets=read.csv("D:/BIOESTADÍSTICA-INFORMÁTICA/Análisis de datos
ómicos/PEC2/targets.csv",header=TRUE,sep=",") #Cargamos los datos del fichero targets.csv
targets
####
Seleccionamos 30 muestras, n=10 NIT, n=10 SFI y n=10 ELI, de forma aleatoria
```{r, include = FALSE}
set.seed(679987)
targets_NIT<-subset(targets, targets$Group=="NIT")
targets_SFI<-subset(targets, targets$Group=="SFI")
targets_ELI<-subset(targets, targets$Group=="ELI")
library("dplyr")
NIT <- sample_n(targets_NIT, size = 10, replace=FALSE)
SFI <- sample_n(targets_SFI, size = 10, replace=FALSE)
ELI <- sample_n(targets_ELI, size = 10, replace=FALSE)
targets_30<-rbind(NIT, SFI, ELI)
targets_30
####
##### Cargamos los datos del fichero counts_30.csv que contiene los datos de counts de las 30 muestras
que arriba hemos elegido aleatoriamente y que previamente han sido ordenadas las muestras según
aparecían en "targets_30".
```{r, include = FALSE}
counts_30=read.csv("D:/BIOESTADÍSTICA-INFORMÁTICA/Análisis de datos
ómicos/PEC2/counts_30_ord.csv",header=TRUE,sep=",")
counts_30
####
```{r, echo = FALSE}
library("DESeq2")
####
```{r, echo = FALSE}
tmp <- gsub("\\.\"", "", counts_30[,1])
row.names(counts_30)<-tmp
counts_30<-counts_30[,-1]
####
Creamos un objeto de clase DESeqDataSetMatrix con los datos de expresión de las 30 muestras
que han sido escogidas al azar
```{r}
dds <- DESeqDataSetFromMatrix(countData = counts_30, colData = targets_30, design = ~ Group)
dds
####
### PREPROCESADO DE LOS DATOS: FILTRAJE Y NORMALIZACIÓN
```{r}
nrow(dds)
dds<- dds[rowSums(counts(dds)) > 1,]
nrow(dds) #De 56202 genes nos quedamos con 43507
####
```{r, include = FALSE}
# Transformación estabilizadora de la varianza (vst)
vsd <- vst(dds, blind = FALSE)
head(assay(vsd), 3)
colData(vsd)
####
```{r, include = FALSE}
Transformación logarítmica regularizada (rlog)
rld <- rlog(dds, blind = FALSE)
head(assay(rld), 3)
####
```{r, echo = FALSE}
dds<- estimateSizeFactors(dds)
df <- bind_rows(
  as_data_frame(log2(counts(dds, normalized=TRUE)[, 1:2]+1)) %>%
    mutate(transformation = "log2(x + 1)"),
  as_data_frame(assay(vsd)[, 1:2]) %>% mutate(transformation = "vst"),
  as_data_frame(assay(rld)[, 1:2]) %>% mutate(transformation = "rlog")
)
colnames(df)[1:2] <- c("x", "y")
ggplot(df, aes(x = x, y = y)) + geom_hex(bins = 80) +
  coord_fixed() + facet_grid( . ~ transformation)
```

```

```

```{r include = FALSE}
png("log_vst.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
ggplot(df, aes(x = x, y = y)) + geom_hex(bins = 80) +
  coord_fixed() + facet_grid( . ~ transformation)
dev.off()
```

Distancias
```{r, include = FALSE}
sampleDists <- dist(t(assay(vsd)))
sampleDists

```

```{r, echo = FALSE}
library("pheatmap")
library("RColorBrewer")
sampleDistMatrix <- as.matrix( sampleDists )
rownames(sampleDistMatrix) <- paste(dds$ShortName, sep = " - ")
colnames(sampleDistMatrix) <- dds$ShortName
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
  clustering_distance_rows = sampleDists,
  clustering_distance_cols = sampleDists,
  col = colors)
```

```{r include = FALSE}
png("heatmap30.png", width = 20, height = 16,
    units = "cm", res = 600, pointsize = 10)
pheatmap(sampleDistMatrix,
  clustering_distance_rows = sampleDists,
  clustering_distance_cols = sampleDists,
  col = colors)
dev.off()
```

ACP
```{r, echo = FALSE}
library(ggplot2)
data <- plotPCA(vsd, intgroup = c("Group"), returnData=TRUE)
percentVar <- round(100 * attr(data, "percentVar"))
ggplot(data, aes(PC1, PC2, color=Group, shape=Group)) + geom_point(size=3) + xlab(paste0("PC1:
",percentVar[1],"% variance")) + ylab(paste0("PC2: ",percentVar[2],"% variance"))
```

```{r include = FALSE}
png("pca30.png", width = 20, height = 12,
    units = "cm", res = 600, pointsize = 10)
ggplot(data, aes(PC1, PC2, color=Group, shape=Group)) + geom_point(size=3) + xlab(paste0("PC1:
",percentVar[1],"% variance")) + ylab(paste0("PC2: ",percentVar[2],"% variance"))
dev.off()
```

MDS plot usando los datos de la matriz de distancias.
```{r, echo = FALSE}
mds <- as.data.frame(colData(vsd)) %>%
  cbind(cmdscale(sampleDistMatrix))
ggplot(mds, aes(x = `1`, y = `2`, color = Group, shape = Group)) +
  geom_point(size = 3) + coord_fixed()
```

```{r include = FALSE}
png("mds30.png", width = 20, height = 16,
    units = "cm", res = 600, pointsize = 10)
ggplot(mds, aes(x = `1`, y = `2`, color = Group, shape = Group)) +
  geom_point(size = 3) + coord_fixed()
dev.off()
```

ANÁLISIS DE EXPRESIÓN DIFERENCIAL
```{r, include = FALSE}
dds <- DESeq(dds, parallel =TRUE)
```

Tabla results de cada comparación

```

```
SFI versus NIT
```{r}
res_SFI_NIT<-results(dds, contrast = c("Group", "SFI", "NIT"))
mcols(res_SFI_NIT, use.names = TRUE)
summary(res_SFI_NIT)
```

SFI versus ELI
```{r}
res_SFI_ELI<-results(dds, contrast = c("Group", "SFI", "ELI"))
mcols(res_SFI_ELI, use.names = TRUE)
summary(res_SFI_ELI)
```

NIT versus ELI
```{r}
res_NIT_ELI<-results(dds, contrast = c("Group", "NIT", "ELI"))
mcols(res_NIT_ELI, use.names = TRUE)
summary(res_NIT_ELI)
```

```{r, include = FALSE}
library(AnnotationDbi)
library(EnsDb.Hsapiens.v86)
```

```{r, include = FALSE}
res_SFI_NIT$symbol<-mapIds(EnsDb.Hsapiens.v86,
                           keys=row.names(res_SFI_NIT),
                           column="SYMBOL",
                           keytype="GENEID",
                           multiVals="first")
res_SFI_ELI$symbol<-mapIds(EnsDb.Hsapiens.v86,
                           keys=row.names(res_SFI_ELI),
                           column="SYMBOL",
                           keytype="GENEID",
                           multiVals="first")
res_NIT_ELI$symbol<-mapIds(EnsDb.Hsapiens.v86,
                           keys=row.names(res_NIT_ELI),
                           column="SYMBOL",
                           keytype="GENEID",
                           multiVals="first")
```

Genes más downregulados y upregulados en cada una de las tres comparaciones usando como
criterio el p-adjusted value (padj) a un nivel de significación del 0.1 y ordenando los genes de acuerdo al
valor de log2 fold change.
SFI versus NIT
Genes más downregulados:
```{r}
res_SFI_NIT_Sig <- subset(res_SFI_NIT, padj < 0.1)
head(res_SFI_NIT_Sig[ order(res_SFI_NIT_Sig$log2FoldChange), ])
```

Genes más upregulados:
```{r}
head(res_SFI_NIT_Sig[ order(res_SFI_NIT_Sig$log2FoldChange, decreasing = TRUE), ])
```

SFI versus ELI
Genes más downregulados:
```{r}
res_SFI_ELI_Sig <- subset(res_SFI_ELI, padj < 0.1)
head(res_SFI_ELI_Sig[ order(res_SFI_ELI_Sig$log2FoldChange), ])
```

Genes más upregulados:
```{r}
head(res_SFI_ELI_Sig[ order(res_SFI_ELI_Sig$log2FoldChange, decreasing = TRUE), ])
```

NIT versus ELI
Genes más downregulados:
```{r}
res_NIT_ELI_Sig <- subset(res_NIT_ELI, padj < 0.1)
head(res_NIT_ELI_Sig[ order(res_NIT_ELI_Sig$log2FoldChange), ])
```

```

```

...
Genes más upregulados:
```{r}
head(res_NIT_ELI_Sig[ order(res_NIT_ELI_Sig$log2FoldChange, decreasing = TRUE), ])
```

MAplots
```{r, echo = FALSE}
plotMA(as.data.frame(res_SFI_NIT),ylim=c(-10,10)) #SFI versus NIT
plotMA(as.data.frame(res_SFI_ELI),ylim=c(-10,12)) #SFI versus ELI
plotMA(as.data.frame(res_NIT_ELI),ylim=c(-10,10)) #NIT versus ELI
```

```{r include = FALSE}
png("MAPlot1.png", width = 14, height = 12,
    units = "cm", res = 600, pointsize = 10)
plotMA(as.data.frame(res_SFI_NIT),ylim=c(-10,10))
dev.off()
```

```{r include = FALSE}
png("MAPlot2.png", width = 14, height = 12,
    units = "cm", res = 600, pointsize = 10)
plotMA(as.data.frame(res_SFI_ELI),ylim=c(-10,12))
dev.off()
```

```{r include = FALSE}
png("MAPlot3.png", width = 14, height = 12,
    units = "cm", res = 600, pointsize = 10)
plotMA(as.data.frame(res_NIT_ELI),ylim=c(-10,12))
dev.off()
```

AGRUPACIÓN DE LOS GENES MÁS DIFERENCIALMENTE EXPRESADOS
Heatmap con 20 genes con las varianzas más altas entre las 30 muestras.
```{r, echo = FALSE}
library("genefilter")
topVarGenes <- head(order(rowVars(assay(vsd)), decreasing = TRUE), 20)
mat <- assay(vsd)[topVarGenes, ]
mat <- mat - rowMeans(mat)
colnames(mat) = dds$ShortName
row.names(mat) = mapIds(EnsDb.Hsapiens.v86,
    keys=row.names(mat),
    column="SYMBOL",
    keytype="GENEID",
    multiVals="first")
anno <- as.data.frame(colData(vsd)[,"Group"])
row.names(anno) = colnames(mat)
colnames(anno) = "Group"
pheatmap(mat, annotation_col = anno)
```

```{r include = FALSE}
png("topVarGenes_phenommap.png", width = 20, height = 18,
    units = "cm", res = 600, pointsize = 10)
pheatmap(mat, annotation_col = anno)
dev.off()
```

ANOTACIÓN DE GENES Y EXPORTACION DE RESULTADOS
```{r, include = FALSE}
library("org.Hs.eg.db")
library("AnnotationDbi")

##### Anotación para la comparación SFI vs NIT y almacenamiento
```{r}
res_SFI_NIT$symbol <- mapIds(org.Hs.eg.db,
 keys=row.names(res_SFI_NIT),
 column="SYMBOL",
 keytype="ENSEMBL",
 multiVals="first")
res_SFI_NIT$entrez <- mapIds(org.Hs.eg.db,
 keys=row.names(res_SFI_NIT),

```

```

 column="ENTREZID",
 keytype="ENSEMBL",
 multiVals="first")
res_SFI_NIT_ord<- res_SFI_NIT[order(res_SFI_NIT$pvalue),]
head(res_SFI_NIT_ord)
res_SFI_NIT_ord_df<-as.data.frame(res_SFI_NIT_ord)
write.csv(res_SFI_NIT_ord_df, file = "results_SFI_NIT.csv")
````
````{r, include = FALSE}
head(res_SFI_NIT_ord_df$symbol, 100)
````

##### Anotación de los resultados de la comparación SFI vs ELI
````{r}
res_SFI_ELI$symbol <- mapIds(org.Hs.eg.db,
 keys=row.names(res_SFI_ELI),
 column="SYMBOL",
 keytype="ENSEMBL",
 multiVals="first")
res_SFI_ELI$entrez <- mapIds(org.Hs.eg.db,
 keys=row.names(res_SFI_ELI),
 column="ENTREZID",
 keytype="ENSEMBL",
 multiVals="first")
res_SFI_ELI_ord<- res_SFI_ELI[order(res_SFI_ELI$pvalue),]
head(res_SFI_ELI_ord)
res_SFI_ELI_ord_df<-as.data.frame(res_SFI_ELI_ord)
write.csv(res_SFI_ELI_ord_df, file = "results_SFI_ELI.csv")
````
````{r, include = FALSE}
head(res_SFI_ELI_ord_df$symbol, 100)
````

##### Anotación de los resultados de la comparación NIT vs ELI
````{r, include = FALSE}
res_NIT_ELI$symbol <- mapIds(org.Hs.eg.db,
 keys=row.names(res_NIT_ELI),
 column="SYMBOL",
 keytype="ENSEMBL",
 multiVals="first")
res_NIT_ELI$entrez <- mapIds(org.Hs.eg.db,
 keys=row.names(res_NIT_ELI),
 column="ENTREZID",
 keytype="ENSEMBL",
 multiVals="first")
res_NIT_ELI_ord<- res_NIT_ELI[order(res_NIT_ELI$pvalue),]
res_NIT_ELI_ord_df<-as.data.frame(res_NIT_ELI_ord)
write.csv(res_NIT_ELI_ord_df, file = "results_NIT_ELI.csv")
````

````{r}
head(res_NIT_ELI_ord_df$symbol, 100)
````

#### ESTUDIO DE ENRIQUECIMIENTO DE GENES (GO) Y RUTAS METABÓLICAS (KEGG)
````{r}
library(clusterProfiler)
library("org.Hs.eg.db")
OrgDb <- org.Hs.eg.db
````

##### SFI vs NIT
````{r}
geneList1<-as.vector(res_SFI_NIT_ord_df$log2FoldChange)
names(geneList1)<- res_SFI_NIT_ord_df$entrez
gene1<- na.omit(res_SFI_NIT_ord_df$entrez)
go1 <- clusterProfiler::enrichGO(gene = gene1,
 OrgDb = OrgDb,
 ont = "BP",
 pAdjustMethod = "BH",
 pvalueCutoff = 0.05,

```



```

 qvalueCutoff = 0.05,
 readable = TRUE)

 {r}
 head(summary(go1)[-10])
 barplot(go1, showCategory=10)
 dotplot(go1, showCategory=10)

 {r include = FALSE}
 png("barplot_go1.png", width = 26, height = 16,
 units = "cm", res = 600, pointsize = 10)
 barplot(go1, showCategory=10)
 dev.off()

 {r include = FALSE}
 png("dotplot_go1.png", width = 26, height = 16,
 units = "cm", res = 600, pointsize = 10)
 dotplot(go1, showCategory=10)
 dev.off()

 {r}
 kegg1<- clusterProfiler::enrichKEGG(gene= gene1,
 organism = 'hsa',
 pAdjustMethod = "BH",
 pvalueCutoff = 0.05,
 qvalueCutoff = 0.05)
 head(summary(kegg1)[-10])
 barplot(kegg1, showCategory=10)
 dotplot(kegg1, showCategory=10)

 {r include = FALSE}
 png("barplot_ke1.png", width = 26, height = 16,
 units = "cm", res = 600, pointsize = 10)
 barplot(kegg1, showCategory=10)
 dev.off()

 {r include = FALSE}
 png("dotplot_ke1.png", width = 26, height = 16,
 units = "cm", res = 600, pointsize = 10)
 dotplot(kegg1, showCategory=10)
 dev.off()

SFI vs ELI
 {r}
 geneList2<-as.vector(res_SFI_ELI_ord_df$log2FoldChange)
 names(geneList2)<- res_SFI_ELI_ord_df$entrez
 gene2<- na.omit(res_SFI_ELI_ord_df$entrez)
 go2 <- clusterProfiler::enrichGO(gene = gene2,
 OrgDb = OrgDb,
 ont = "BP",
 pAdjustMethod = "BH",
 pvalueCutoff = 0.05,
 qvalueCutoff = 0.05,
 readable = TRUE)

 {r}
 head(summary(go2)[-10])
 barplot(go2, showCategory=10)
 dotplot(go2, showCategory=10)

 {r}
 kegg2<- clusterProfiler::enrichKEGG(gene= gene2,
 organism = 'hsa',
 pAdjustMethod = "BH",
 pvalueCutoff = 0.05,
 qvalueCutoff = 0.05)
 head(summary(kegg2)[-10])

```

```

barplot(kegg2, showCategory=10)
dotplot(kegg2,showCategory=10)
....

NIT vs ELI
####{r}
geneList3<-as.vector(res_NIT_ELI_ord_df$log2FoldChange)
names(geneList3)<- res_NIT_ELI_ord_df$entrez
gene3<- na.omit(res_NIT_ELI_ord_df$entrez)
go3 <- clusterProfiler::enrichGO(gene = gene3,
 OrgDb = OrgDb,
 ont = "BP",
 pAdjustMethod = "BH",
 pvalueCutoff = 0.05,
 qvalueCutoff = 0.05,
 readable = TRUE)
....

####{r}
go3.df<- as.data.frame(go3)
head(summary(go3)[-10])
barplot(go3, showCategory=10)
dotplot(go3, showCategory=10)
....

####{r}
kegg3<- clusterProfiler::enrichKEGG(gene= gene3,
 organism = 'hsa',
 pAdjustMethod = "BH",
 pvalueCutoff = 0.05,
 qvalueCutoff = 0.05)
head(summary(kegg3)[-10])
barplot(kegg3, showCategory=10)
dotplot(kegg3,showCategory=10)
....

```