

MO. 157 Análisis de datos ómicos. PEC 2

Rubén Jové Nieto

14/6/2020

Enlace al repositorio de github.

https://github.com/rbenjn/PEC_02

Abstract

A partir de unas muestras de un estudio obtenido del repositorio GTEx (Genotype-Tissue Expression), nos centramos en los datos de expresión (RNA-seq) pertenecientes a un análisis del tiroides. En él, se comparan a 3 tipos distintos de grupo según infiltración, de un total de 292: NIT (tejidos no infiltrados), SFI (pequeñas infiltraciones focales) y ELI (infiltraciones extensas linfoides). Obteniendo una muestra de estudio aleatoria de 30 (10 para cada grupo) se procede a su comparación. Obteniendo

Objetivos

El objetivo de este análisis es comparar y obtener las diferencias significativas resultado de realizar un análisis de expresión diferencia. También así poder constatar las posibles diferencias entre los 3 grupos a partir de los datos de expresión.

Materiales y métodos

Para obtener la muestra de $n = 30$ de manera aleatoria se realizó un pequeño script mediante R. A través de los 2 archivos: 'targets' y 'counts', se construyó la muestra de estudio.

Mediante R y BioConductor se realizó el análisis de datos. Utilizando el paquete DESeq2 para el análisis de los datos de expresión (RNA-seq).

El análisis de significación biológica se hizo mediante el enriquecimiento del conjunto de genes con Cluster-Profiler.

Se realizarán 3 comparaciones: -NIT vs SFI -SFI vs ELI -ELI vs NIT

“Pipeline” análisis

Los pasos o “pipeline” seguido para el análisis han sido:

1. Identificar que grupos hay y a qué grupo pertenece cada muestra.
2. Preprocesado de los datos: filtraje y normalización
3. Identificación de genes diferencialmente expresados
4. Anotación de los resultados
5. Busca de patrones de expresión y agrupación de las muestras(Comparación entre distintas comparaciones)
6. Análisis de significación biológica (“Gene Enrichment Analysis”)

“Pipeline” ampliada

1. Identificar que grupos hay y a qué grupo pertenece cada muestra. En primer lugar, importamos los 2 distintos archivos: ‘targets.csv’ y ‘counts.csv’ a R.

A continuación, creamos un script que nos permita extraer 10 muestras de cada grupo (NIT, SFI y ELI). Para ello, utilizamos la información de la columna ‘Group’ en el archivo ‘targets’. Dicho script nos aseguraremos que sea reproducible y aleatorio mediante ‘set.seed’.

Así quedaría el dataframe con las muestras escogidas.

```
## # A tibble: 6 x 9
##   Experiment SRA_Sample Sample_Name Grupo_analisis body_site molecular_data_~
##   <chr>      <chr>      <chr>          <dbl> <chr>      <chr>
## 1 SRX624616 SRS646000 GTEX-ZVP2~      1 Thyroid  RNA Seq (NGS)
## 2 SRX589853 SRS634351 GTEX-145MH~      1 Thyroid  RNA Seq (NGS)
## 3 SRX572499 SRS627895 GTEX-13061~      1 Thyroid  RNA Seq (NGS)
## 4 SRX597647 SRS637302 GTEX-11I78~      1 Thyroid  RNA Seq (NGS)
## 5 SRX222627 SRS389690 GTEX-QV31~      1 Thyroid  Allele-Specific~
## 6 SRX559141 SRS624025 GTEX-11220~      1 Thyroid  RNA Seq (NGS)
## # ... with 3 more variables: sex <chr>, Group <chr>, ShortName <chr>
```

Una vez tenemos las filas escogidas, 10 muestras para cada grupo, procedemos a ‘cruzar’ o ‘subsetear’ las columnas escogidas en el archivo ‘counts.csv’.

Obtenemos entonces, el dataframe ‘countdata’ con 30 variables, 10 para cada grupo, de manera aleatoria.

Una vez tenemos la dataframe, podemos ya construir el objeto DESeqDataSet.

2. Preprocesado de los datos: filtraje y normalización Procedemos a filtrar los datos ya que nuestro DESeqDataSet tiene filas con 0. Lo hacemos con el fin de reducir el tamaño del ‘dds’ y así agilizar las funciones.

```
## [1] 56202
```

```
## [1] 43699
```

Pasando tener de 56.202 filas a 43.699.

Para poder estabilizar la varianza de los datos, usaremos la función VST (para medios-grandes datasets, $n > 30$)

```
##           1.1      1.2      1.3      1.4      1.5      1.6
## ENSG00000223972.4 4.475310 4.363455 4.229991 4.372187 4.385326 4.255178
## ENSG00000227232.4 9.234145 9.273780 9.460894 9.209663 9.094404 9.800575
## ENSG00000243485.2 4.168691 3.628672 4.055396 3.628672 4.499164 4.073405
##           1.7      1.8      1.9      1.10      2.1      2.2
## ENSG00000223972.4 4.239974 4.286512 4.659221 4.034320 4.197054 4.211554
## ENSG00000227232.4 8.933064 8.679066 8.807334 9.833603 10.141167 9.725608
## ENSG00000243485.2 3.628672 4.688274 4.152096 3.628672 3.958237 4.300234
##           2.3      2.4      2.5      2.6      2.7      2.8
## ENSG00000223972.4 4.297062 4.047944 4.242781 4.263899 4.269980 3.628672
## ENSG00000227232.4 9.879822 9.882746 9.933560 9.099115 9.587032 9.487858
## ENSG00000243485.2 4.297062 4.787338 4.064539 3.628672 4.153718 3.628672
```

```

##          2.9      2.10      3.1      3.2      3.3      3.4
## ENSG00000223972.4 4.328987 4.223164 4.902760 3.628672 4.322605 4.687723
## ENSG00000227232.4 8.907394 8.997709 8.919048 9.661885 10.332883 10.216479
## ENSG00000243485.2 3.628672 4.050517 4.379890 3.990921 4.031874 4.069089
##          3.5      3.6      3.7      3.8      3.9      3.10
## ENSG00000223972.4 4.006336 4.222429 4.457829 3.628672 3.950132 3.628672
## ENSG00000227232.4 9.420341 9.487709 8.765157 9.895897 9.864630 9.982187
## ENSG00000243485.2 4.161267 4.222429 4.003551 4.062134 3.950132 4.073130

## DataFrame with 30 rows and 10 columns
##      Experiment  SRA_Sample      Sample_Name Grupo_analisis
##      <character> <character>      <character>      <numeric>
## 1.1  SRX624616  SRS646000  GTEX-ZVP2-0426-SM-57WC2          1
## 1.2  SRX589853  SRS634351  GTEX-145MH-0426-SM-5LU8T          1
## 1.3  SRX572499  SRS627895  GTEX-13061-0226-SM-5KM52          1
## 1.4  SRX597647  SRS637302  GTEX-11I78-0526-SM-5986A          1
## 1.5  SRX222627  SRS389690  GTEX-QV31-0726-SM-3GAEG          1
## ...      ...      ...      ...      ...
## 3.6  SRX222429  SRS389623  GTEX-TMMY-0826-SM-33HB9          3
## 3.7  SRX199272  SRS333099  GTEX-PLZ4-1226-SM-2I5FE          3
## 3.8  SRX601511  SRS638114  GTEX-13QJC-0826-SM-5RQKC          3
## 3.9  SRX615373  SRS644099  GTEX-YFC4-2626-SM-5P9FQ          3
## 3.10 SRX607358  SRS639491  GTEX-14AS3-0226-SM-5Q5B6          3
##      body_site      molecular_data_type      sex      Group      ShortName
##      <character>      <character> <character> <factor> <character>
## 1.1  Thyroid      RNA Seq (NGS)      male      NIT      ZVP2-_NIT
## 1.2  Thyroid      RNA Seq (NGS)      male      NIT      145MH_NIT
## 1.3  Thyroid      RNA Seq (NGS)      male      NIT      13061_NIT
## 1.4  Thyroid      RNA Seq (NGS)      female     NIT      11I78_NIT
## 1.5  Thyroid Allele-Specific Expression      male      NIT      QV31-_NIT
## ...      ...      ...      ...      ...
## 3.6  Thyroid Allele-Specific Expression      female     ELI      TMMY-_ELI
## 3.7  Thyroid      RNA Seq (NGS)      female     ELI      PLZ4-_ELI
## 3.8  Thyroid Allele-Specific Expression      female     ELI      13QJC_ELI
## 3.9  Thyroid Allele-Specific Expression      female     ELI      YFC4-_ELI
## 3.10 Thyroid      RNA Seq (NGS)      female     ELI      14AS3_ELI
##      sizeFactor
##      <numeric>
## 1.1  1.140662
## 1.2  0.915051
## 1.3  0.917387
## 1.4  0.893225
## 1.5  0.861796
## ...      ...
## 3.6  1.411864
## 3.7  1.190652
## 3.8  0.888879
## 3.9  1.621658
## 3.10 0.845116

```

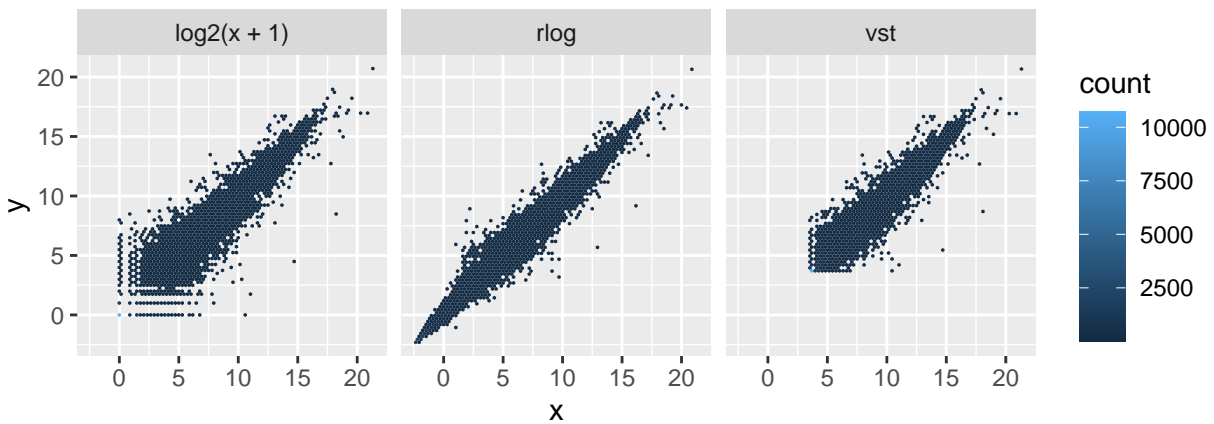
También podríamos realizar la transformación con ‘rlog’, que funciona mejor con muestras pequeñas ($n < 30$). De hecho así nos lo muestra el mensaje que aparece en la consola, recomendando vst().

```
## rlog() may take a few minutes with 30 or more samples,
```

```
## vst() is a much faster transformation
```

```
##          1.1      1.2      1.3      1.4      1.5      1.6
## ENSG00000223972.4 1.465206 1.3497010 1.2271724 1.3576602 1.3695933 1.2494382
## ENSG00000227232.4 9.244778 9.2748751 9.4169384 9.2263860 9.1394694 9.6767440
## ENSG00000243485.2 0.570240 0.3171452 0.4809149 0.3184021 0.8764663 0.4945403
##          1.7      1.8      1.9      1.10     2.1      2.2
## ENSG00000223972.4 1.2360223 1.278669 1.6700358 1.0701847 1.194665 1.2083355
## ENSG00000227232.4 9.0182543 8.828270 8.9232865 9.7022270 9.940065 9.6193553
## ENSG00000243485.2 0.3187547 1.087587 0.5578084 0.3118624 0.406815 0.6954525
##          2.3      2.4      2.5      2.6      2.7      2.8
## ENSG00000223972.4 1.2859746 1.081953 1.2385051 1.2570987 1.2632292 0.9239201
## ENSG00000227232.4 9.7376322 9.740030 9.7791573 9.1430631 9.5131781 9.4374740
## ENSG00000243485.2 0.6760885 1.181256 0.4878439 0.3229192 0.5576887 0.3148990
##          2.9      2.10     3.1      3.2      3.3      3.4
## ENSG00000223972.4 1.3180682 1.2211035 1.7888488 0.9035137 1.3121753 1.6726987
## ENSG00000227232.4 8.9988475 9.0666693 9.0107554 9.5704655 10.0888800 9.9981716
## ENSG00000243485.2 0.3121827 0.4772097 0.6969553 0.4316937 0.4630059 0.4912835
##          3.5      3.6      3.7      3.8      3.9      3.10
## ENSG00000223972.4 1.045901 1.2185806 1.4483669 0.9295448 0.9971832 0.9335508
## ENSG00000227232.4 9.386057 9.4373607 8.8923010 9.7501269 9.7262836 9.8166640
## ENSG00000243485.2 0.564019 0.6204204 0.4413517 0.4860236 0.4006927 0.4943332
```

Mostramos en plot la comparativa de ambos resultados de transformación de los datos:

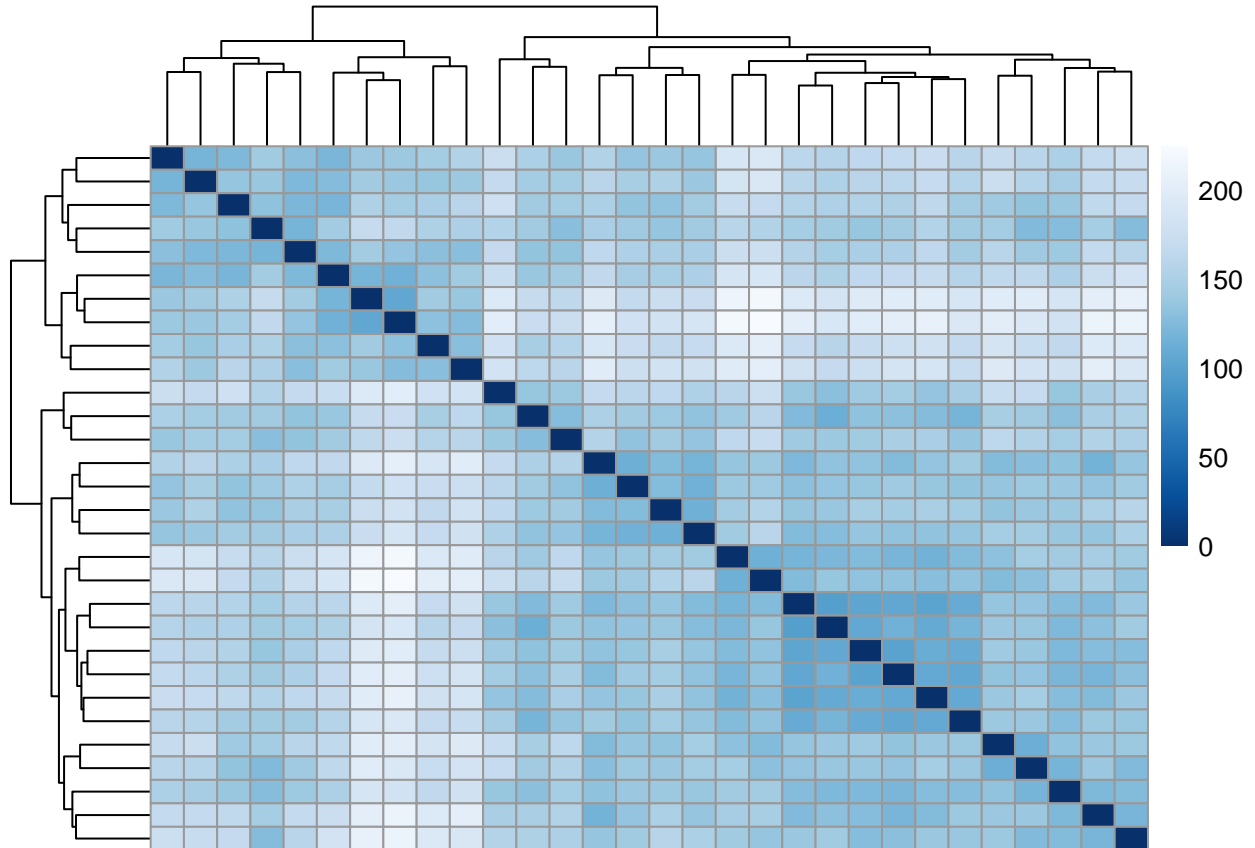


Utilizando los resultados de VST, es decir el objeto ‘vsd’ en el análisis, podemos evaluar la similitud general entre muestras. Podemos evaluar si son similares entre ellas o diferentes, así poder hacernos una primera

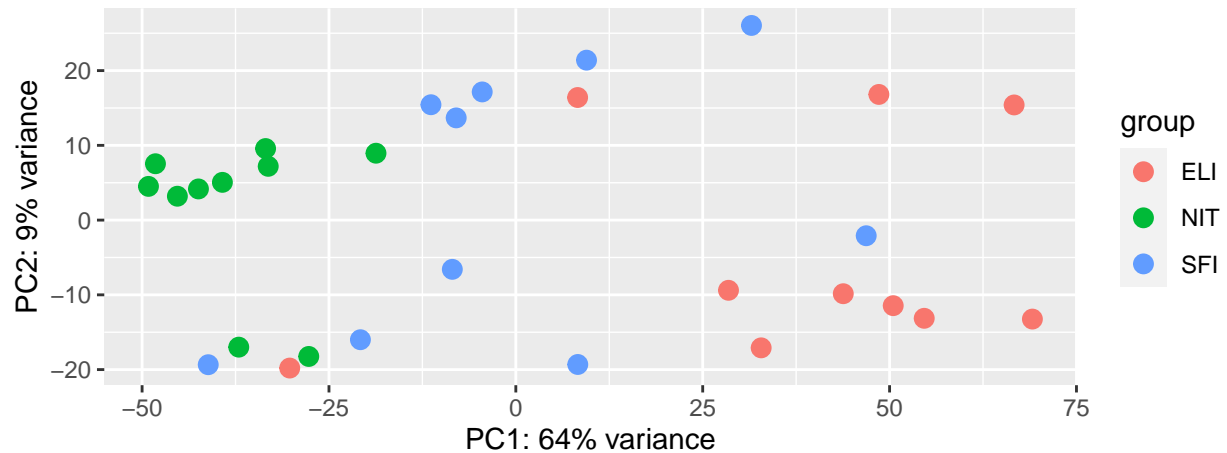
idea si cumple con lo esperado con el análisis. La función 'sampleDists' para calcular las distancias entre las muestras nos ayuda a ello.

Podemos observar una mayor distancia en muestras de distintos grupos, es decir entre las del grupo 1 NIT (1.1-1.10) hay una menor distancia, en cambio esta aumenta al compararse con las del grupo 3 ELI por ejemplo.

Lo visualizamos en el siguiente mapa de calor:



Realizamos el gráfico PCA para cada grupo con los datos obtenidos de VST también:



Podemos observar cada color para cada grupo y su varianza, surgiendo ciertos patrones similares según grupo.

3. Identificación de genes diferencialmente expresados Utilizamos la función ‘DESeq’ en nuestro objeto creado ‘dds’, para así luego poder de este obtener los resultados que nos interesan a partir de la opción ‘contrast’, indicando el factor ‘Group’ que es donde obtenemos los 3 distintos grupos según infiltración tiroídea y las consecuentes comparaciones.

Mostraremos un resumen de los resultados para cada comparación.

NIT vs ELI

```
##
## out of 43697 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 3171, 7.3%
## LFC < 0 (down)    : 5147, 12%
## outliers [1]      : 0, 0%
## low counts [2]     : 11863, 27%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

NIT vs SFI

```
##
## out of 43697 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 430, 0.98%
## LFC < 0 (down)    : 1178, 2.7%
## outliers [1]      : 0, 0%
## low counts [2]    : 16099, 37%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

ELI vs SFI

```
##
## out of 43697 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 2501, 5.7%
## LFC < 0 (down)    : 1169, 2.7%
## outliers [1]      : 0, 0%
## low counts [2]    : 15251, 35%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Con los datos obtenidos, podemos proceder a extraer y aquellos significativos. Teniendo en cuenta un posible 10% de falsos positivos, para cada comparación obtenemos:

NIT vs ELI: 8318 genes significativos

NIT vs SFI: 1608 genes significativos

ELI vs SFI: 3670 genes significativos

Posteriormente podemos identificar y mostrar aquellos que están down-regulated para cada comparación:

```
## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 6 rows and 6 columns
##      baseMean log2FoldChange    lfcSE    stat    pvalue
##      <numeric> <numeric> <numeric> <numeric> <numeric>
## ENSG00000181617.5 381.3621    -9.90648 1.649666 -6.00514 1.91166e-09
## ENSG00000222037.5 797.3926    -9.79812 0.960263 -10.20358 1.91095e-24
## ENSG00000223350.2 136.8381    -9.69551 1.286652 -7.53546 4.86624e-14
## ENSG00000170054.10 55.1950    -9.25324 1.314936 -7.03702 1.96389e-12
## ENSG00000253274.1 69.1401    -9.06199 1.219109 -7.43328 1.05934e-13
## ENSG00000117215.10 138.4381    -8.98663 1.188865 -7.55900 4.06169e-14
##      padj
##      <numeric>
## ENSG00000181617.5 9.67562e-08
## ENSG00000222037.5 1.71518e-20
## ENSG00000223350.2 7.17229e-12
## ENSG00000170054.10 1.94169e-10
## ENSG00000253274.1 1.41703e-11
## ENSG00000117215.10 6.07080e-12
```

```
## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000211946.2  119.3200      -10.18393  1.256589  -8.10442  5.29972e-16
## ENSG00000222037.5   797.3926      -8.34657  0.960361  -8.69107  3.59026e-18
## ENSG00000235896.2    59.0700      -8.33345  1.418771  -5.87371  4.26141e-09
## ENSG00000242371.1  2169.5790      -8.02743  0.905127  -8.86884  7.39089e-19
## ENSG00000253274.1    69.1401      -7.86274  1.219781  -6.44602  1.14823e-10
## ENSG00000211619.2    65.4272      -7.81824  1.406149  -5.56004  2.69717e-08
##           padj
##           <numeric>
## ENSG00000211946.2  2.99401e-12
## ENSG00000222037.5  4.95456e-14
## ENSG00000235896.2  1.50788e-06
## ENSG00000242371.1  2.03988e-14
## ENSG00000253274.1  6.46756e-08
## ENSG00000211619.2  7.09057e-06

## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000108342.8  108.06374      -3.81297  0.865798  -4.40400  1.06272e-05
## ENSG00000236136.1    2.89710      -3.43413  1.308220  -2.62504  8.66378e-03
## ENSG00000089225.15  253.92977      -2.98068  0.728853  -4.08955  4.32216e-05
## ENSG00000241717.1    9.57786      -2.86229  0.933100  -3.06751  2.15852e-03
## ENSG00000225893.1    3.14491      -2.79170  0.801341  -3.48378  4.94380e-04
## ENSG00000228036.1    2.19718      -2.76013  1.084950  -2.54402  1.09585e-02
##           padj
##           <numeric>
## ENSG00000108342.8  0.000581392
## ENSG00000236136.1  0.078070097
## ENSG00000089225.15  0.001766620
## ENSG00000241717.1  0.031109474
## ENSG00000225893.1  0.010944981
## ENSG00000228036.1  0.090231074
```

Y los que están up-regulated para cada comparación:

```
## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000110680.8  381.06314      5.45038  1.130294  4.82209  1.42063e-06
## ENSG00000149968.7  185.57747      5.33343  1.046885  5.09458  3.49524e-07
## ENSG00000266670.1    1.49666      4.33003  1.276611  3.39182  6.94311e-04
## ENSG00000079689.9   64.52303      4.21381  0.748237  5.63166  1.78487e-08
## ENSG00000108688.7    2.20990      3.85760  1.173768  3.28651  1.01437e-03
## ENSG00000230663.1    2.48142      3.80534  1.404951  2.70852  6.75837e-03
```



```

##                                padj
##                                <numeric>
## ENSG00000110680.8 3.58377e-05
## ENSG00000149968.7 1.03898e-05
## ENSG00000266670.1 6.51764e-03
## ENSG00000079689.9 7.31313e-07
## ENSG00000108688.7 8.77305e-03
## ENSG00000230663.1 3.74581e-02

## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000110680.8 381.06314      5.08776  1.128884  4.50689 6.57838e-06
## ENSG00000265933.1  4.67583      4.75570  1.281357  3.71145 2.06071e-04
## ENSG00000264940.2 52.91903      4.62055  1.384957  3.33624 8.49189e-04
## ENSG00000106302.5  7.98894      3.36257  1.134534  2.96383 3.03833e-03
## ENSG00000149968.7 185.57747      3.19173  1.039319  3.07098 2.13356e-03
## ENSG00000250433.1  2.93975      3.12168  0.973571  3.20642 1.34396e-03
##                                padj
##                                <numeric>
## ENSG00000110680.8 0.000825288
## ENSG00000265933.1 0.012205093
## ENSG00000264940.2 0.032128935
## ENSG00000106302.5 0.067937030
## ENSG00000149968.7 0.055368985
## ENSG00000250433.1 0.042295683

## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 6 rows and 6 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000257275.2  23.9942      8.63619  1.59285  5.42186 5.89836e-08
## ENSG00000170054.10 55.1950      8.22001  1.29274  6.35862 2.03577e-10
## ENSG00000260303.1  22.9468      7.99439  1.71672  4.65678 3.21190e-06
## ENSG00000100721.6 344.8526      7.23801  1.02300  7.07530 1.49128e-12
## ENSG00000264940.2  52.9190      6.88895  1.38284  4.98176 6.30099e-07
## ENSG00000181617.5 381.3621      6.02588  1.59868  3.76929 1.63712e-04
##                                padj
##                                <numeric>
## ENSG00000257275.2 1.19855e-05
## ENSG00000170054.10 1.68640e-07
## ENSG00000260303.1 2.33688e-04
## ENSG00000100721.6 5.30301e-09
## ENSG00000264940.2 6.66359e-05
## ENSG00000181617.5 4.84627e-03

```

4. Anotación de los resultados Utilizando la base de datos de Homo Sapiens (org.Hs.eg.db), y teniendo en cuenta el factor ENSEMBL, buscaremos los Entrez ID y el gene symbol obtenidos para cada comparación. Tendremos que quitar los puntos de hay después de cada ENSEMBL en nuestra lista con tal que puedan coincidir con los de la base de datos:

Para los datos significativos

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

```
## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000211895.3  60711.813      -7.31672  0.699848 -10.45473  1.39399e-25
## ENSG00000009790.10  1347.972      -4.26265  0.413001 -10.32118  5.65243e-25
## ENSG00000222037.5   797.393      -9.79812  0.960263 -10.20358  1.91095e-24
## ENSG00000132465.6   6764.172      -6.42659  0.630558 -10.19190  2.15502e-24
## ENSG00000159618.11   301.654      -4.40145  0.435089 -10.11620  4.68250e-24
## ENSG00000160856.16   942.398      -6.28575  0.641352  -9.80078  1.11719e-22
##           padj      symbol      entrez
##           <numeric> <character> <character>
## ENSG00000211895.3  4.43791e-21      NA      NA
## ENSG00000009790.10  8.99754e-21      TRAF3IP3      80342
## ENSG00000222037.5  1.71518e-20      NA      NA
## ENSG00000132465.6  1.71518e-20      JCHAIN      3512
## ENSG00000159618.11  2.98144e-20      ADGRG5      221188
## ENSG00000160856.16  5.24460e-19      FCRL3      115352
```

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

```
## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000242371.1  2169.579      -8.02743  0.905127 -8.86884  7.39089e-19
## ENSG00000222037.5   797.393      -8.34657  0.960361 -8.69107  3.59026e-18
## ENSG00000132465.6   6764.172      -5.19656  0.630573 -8.24101  1.70757e-16
## ENSG00000211946.2   119.320     -10.18393  1.256589 -8.10442  5.29972e-16
## ENSG00000211895.3  60711.813      -5.66991  0.699851 -8.10160  5.42394e-16
## ENSG00000211959.2  1642.976      -6.98970  0.874866 -7.98944  1.35548e-15
##           padj      symbol      entrez
##           <numeric> <character> <character>
## ENSG00000242371.1  2.03988e-14      NA      NA
## ENSG00000222037.5  4.95456e-14      NA      NA
## ENSG00000132465.6  1.57096e-12      JCHAIN      3512
## ENSG00000211946.2  2.99401e-12      NA      NA
## ENSG00000211895.3  2.99401e-12      NA      NA
## ENSG00000211959.2  6.23521e-12      NA      NA
```

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

```
## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
```

```
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000204282.3    937.022      2.59434  0.338017   7.67517 1.65200e-14
## ENSG00000007312.8    917.110      3.65081  0.479826   7.60861 2.77060e-14
## ENSG00000263264.1    103.177      3.64957  0.502417   7.26404 3.75709e-13
## ENSG00000009790.10  1347.972      2.99022  0.412168   7.25487 4.02047e-13
## ENSG00000227507.2     763.947      4.29290  0.594871   7.21653 5.33303e-13
## ENSG00000122986.9     307.411      2.21316  0.307446   7.19854 6.08587e-13
##           padj      symbol      entrez
##           <numeric> <character> <character>
## ENSG00000204282.3  3.94090e-10  TNRC6C-AS1  100131096
## ENSG00000007312.8  3.94090e-10      CD79B      974
## ENSG00000263264.1  2.85936e-09      NA      NA
## ENSG00000009790.10 2.85936e-09  TRAF3IP3    80342
## ENSG00000227507.2  2.88551e-09      LTB      4050
## ENSG00000122986.9  2.88551e-09  HVCN1      84329
```

No significativos

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

```
## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000211895.3   60711.813     -7.31672  0.699848 -10.45473 1.39399e-25
## ENSG00000009790.10  1347.972     -4.26265  0.413001 -10.32118 5.65243e-25
## ENSG00000222037.5    797.393     -9.79812  0.960263 -10.20358 1.91095e-24
## ENSG00000132465.6   6764.172     -6.42659  0.630558 -10.19190 2.15502e-24
## ENSG00000159618.11   301.654     -4.40145  0.435089 -10.11620 4.68250e-24
## ENSG00000160856.16   942.398     -6.28575  0.641352  -9.80078 1.11719e-22
##           padj      symbol      entrez
##           <numeric> <character> <character>
## ENSG00000211895.3  4.43791e-21      NA      NA
## ENSG00000009790.10  8.99754e-21  TRAF3IP3    80342
## ENSG00000222037.5  1.71518e-20      NA      NA
## ENSG00000132465.6  1.71518e-20  JCHAIN      3512
## ENSG00000159618.11 2.98144e-20  ADGRG5      221188
## ENSG00000160856.16 5.24460e-19  FCRL3      115352
```

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

```
## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 6 rows and 8 columns
##           baseMean log2FoldChange      lfcSE      stat      pvalue
##           <numeric>      <numeric> <numeric> <numeric>      <numeric>
## ENSG00000242371.1   2169.579     -8.02743  0.905127  -8.86884 7.39089e-19
## ENSG00000222037.5    797.393     -8.34657  0.960361  -8.69107 3.59026e-18
```

```

## ENSG00000132465.6 6764.172 -5.19656 0.630573 -8.24101 1.70757e-16
## ENSG00000211946.2 119.320 -10.18393 1.256589 -8.10442 5.29972e-16
## ENSG00000211895.3 60711.813 -5.66991 0.699851 -8.10160 5.42394e-16
## ENSG00000211959.2 1642.976 -6.98970 0.874866 -7.98944 1.35548e-15
##
##          padj          symbol      entrez
##          <numeric> <character> <character>
## ENSG00000242371.1 2.03988e-14      NA      NA
## ENSG00000222037.5 4.95456e-14      NA      NA
## ENSG00000132465.6 1.57096e-12      JCHAIN      3512
## ENSG00000211946.2 2.99401e-12      NA      NA
## ENSG00000211895.3 2.99401e-12      NA      NA
## ENSG00000211959.2 6.23521e-12      NA      NA

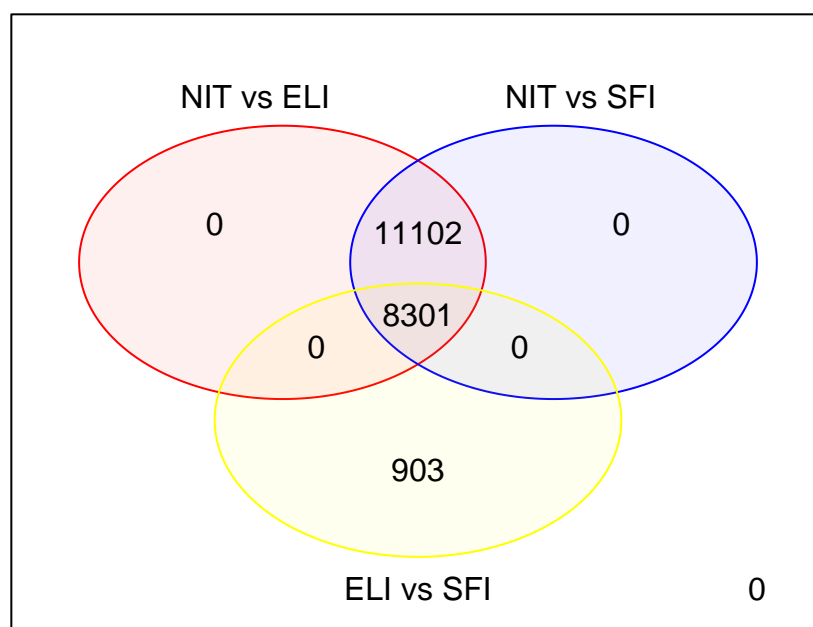
## 'select()' returned 1:many mapping between keys and columns
## 'select()' returned 1:many mapping between keys and columns

## log2 fold change (MLE): Group ELI vs SFI
## Wald test p-value: Group ELI vs SFI
## DataFrame with 6 rows and 8 columns
##
##          baseMean log2FoldChange      lfcSE      stat      pvalue
##          <numeric>      <numeric> <numeric> <numeric> <numeric>
## ENSG00000204282.3      937.022      2.59434 0.338017 7.67517 1.65200e-14
## ENSG00000007312.8      917.110      3.65081 0.479826 7.60861 2.77060e-14
## ENSG00000263264.1      103.177      3.64957 0.502417 7.26404 3.75709e-13
## ENSG00000009790.10    1347.972      2.99022 0.412168 7.25487 4.02047e-13
## ENSG00000227507.2      763.947      4.29290 0.594871 7.21653 5.33303e-13
## ENSG00000122986.9      307.411      2.21316 0.307446 7.19854 6.08587e-13
##
##          padj          symbol      entrez
##          <numeric> <character> <character>
## ENSG00000204282.3 3.94090e-10  TNRC6C-AS1 100131096
## ENSG00000007312.8 3.94090e-10      CD79B      974
## ENSG00000263264.1 2.85936e-09      NA      NA
## ENSG00000009790.10 2.85936e-09  TRAF3IP3      80342
## ENSG00000227507.2 2.88551e-09      LTB      4050
## ENSG00000122986.9 2.88551e-09  HVCN1      84329

```

5. Busca de patrones de expresión y agrupación de las muestras(Comparación entre distintas comparaciones) Realizamos las múltiples comparaciones. Utilizaremos para ello y hacerlo de manera visual, un Diagrama de Venn, adaptando los datos para su análisis y comparar las 3 comparaciones. Se podrá así ver patrones de expresión y agrupaciones de la muestra.

Genes en común en las 3 comparaciones



El Diagrama de Venn nos muestra los genes comunes entre las 3 comparaciones. Un total de 8301

6. Análisis de significación biológica (“Gene Enrichment Analysis”) Utilizamos el paquete ‘ClusterProfiler’ para el análisis de significación biológica. Lo haremos desde 2 perspectivas: por un lado con el workflow de la referencia (3) de Guangchuang Yu. Y por otro lado, con el pathway de ReactomePA adaptado a ClusterProfiler.

ClusterProfiler según referencia (3)

NIT vs ELI

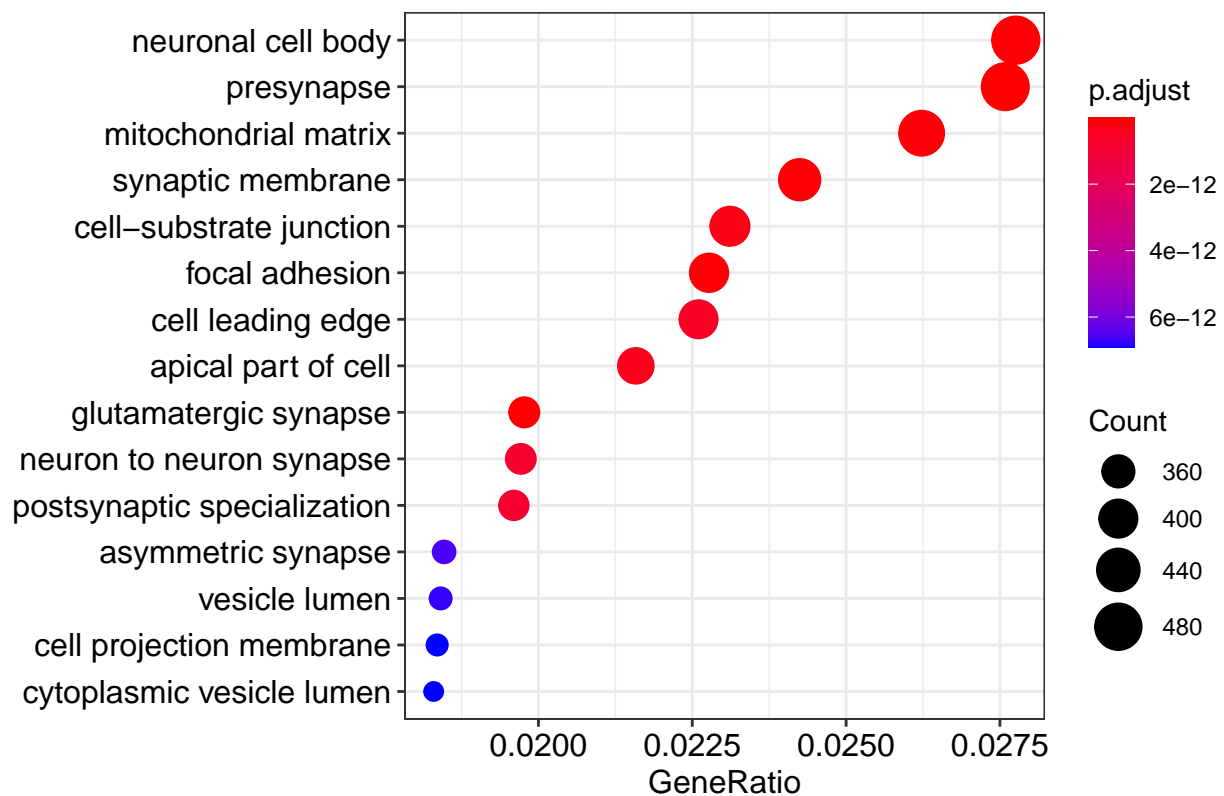
```
## Warning in summary(ego_NIT_vs_ELI): summary method to convert the object to
## data.frame is deprecated, please use as.data.frame instead.
```

##	ID	Description	GeneRatio	BgRatio	pvalue	
##	G0:0098793	G0:0098793	presynapse	487/17653	491/19717	6.818448e-19
##	G0:0098978	G0:0098978	glutamatergic synapse	349/17653	349/19717	1.207379e-17
##	G0:0097060	G0:0097060	synaptic membrane	428/17653	432/19717	3.266557e-16
##	G0:0043025	G0:0043025	neuronal cell body	490/17653	497/19717	3.751104e-16
##	G0:0005925	G0:0005925	focal adhesion	402/17653	405/19717	4.351175e-16
##	G0:0005759	G0:0005759	mitochondrial matrix	463/17653	469/19717	7.383140e-16
##			p.adjust		qvalue	
##	G0:0098793		5.222931e-16		2.935521e-16	
##	G0:0098978		4.624260e-15		2.599042e-15	

```

## G0:0097060 6.666000e-14 3.746591e-14
## G0:0043025 6.666000e-14 3.746591e-14
## G0:0005925 6.666000e-14 3.746591e-14
## G0:0005759 9.425809e-14 5.297727e-14
##
## G0:0098793 5579/5368/23157/3738/5023/8938/9143/5027/8514/594855/10590/1731/135/29993/5874/10890/7466
## G0:0098978
## G0:0097060
## G0:0043025 653361/3676/5368/3783/1269/30009/57471/57699/2185/7074/1233/933/5027/17
## G0:0005925
## G0:0005759 7015/5163/10667/844/28957/2235/11232/3028/5091/8803/2936/11315/27349/11222/88
##
## Count
## G0:0098793 487
## G0:0098978 349
## G0:0097060 428
## G0:0043025 490
## G0:0005925 402
## G0:0005759 463

```



NIT vs ELI con los datos significativos

```

## Warning in summary(ego_NIT_vs_ELI_Sig): summary method to convert the object to
## data.frame is deprecated, please use as.data.frame instead.

```

```

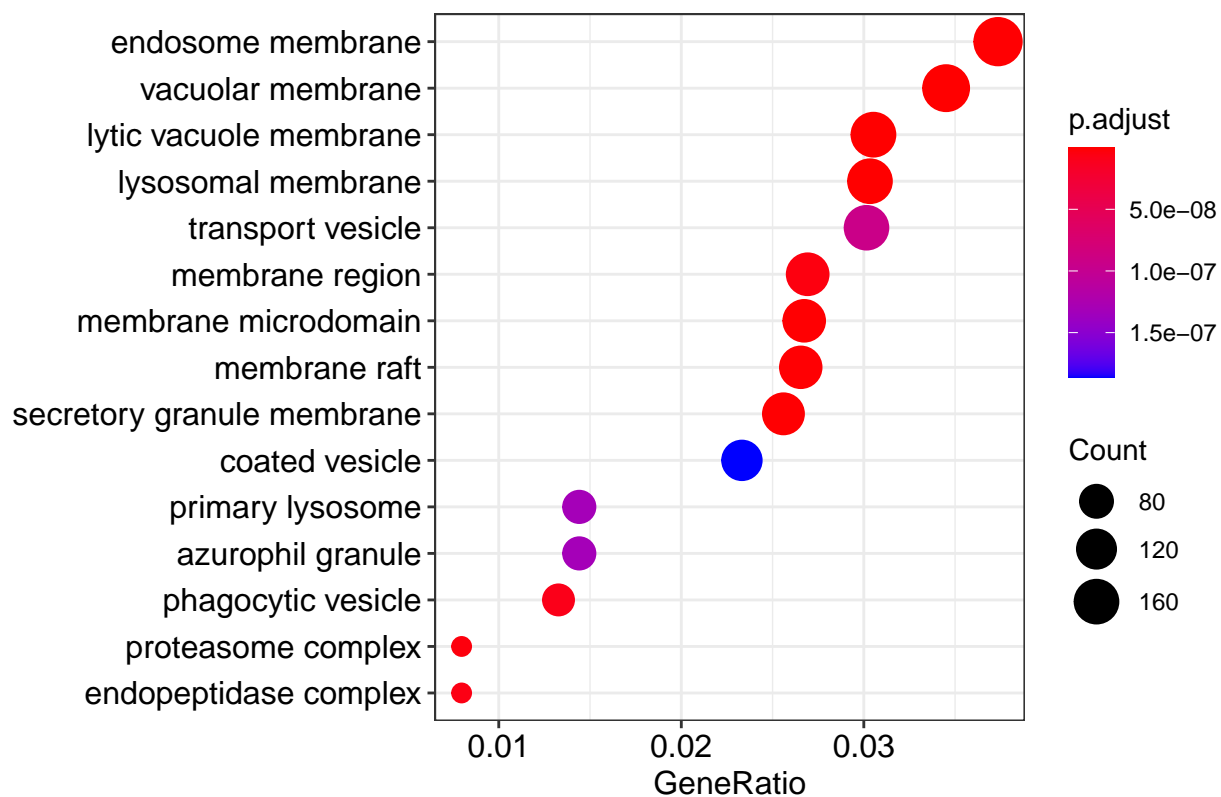
## ID Description GeneRatio BgRatio

```

```

## G0:0005774 G0:0005774      vacuolar membrane 182/5274 412/19717
## G0:0098852 G0:0098852      lytic vacuole membrane 161/5274 355/19717
## G0:0005765 G0:0005765      lysosomal membrane 160/5274 354/19717
## G0:0030667 G0:0030667 secretory granule membrane 135/5274 298/19717
## G0:0010008 G0:0010008      endosome membrane 197/5274 479/19717
## G0:0098857 G0:0098857      membrane microdomain 141/5274 316/19717
##          pvalue      p.adjust      qvalue
## G0:0005774 1.115544e-14 8.522758e-12 6.200077e-12
## G0:0098852 2.472360e-14 9.444417e-12 6.870560e-12
## G0:0005765 4.265344e-14 1.086241e-11 7.902112e-12
## G0:0030667 3.338066e-12 5.610022e-10 4.081140e-10
## G0:0010008 3.671480e-12 5.610022e-10 4.081140e-10
## G0:0098857 4.460296e-12 5.679443e-10 4.131643e-10
##
## G0:0005774
## G0:0098852
## G0:0005765
## G0:0030667
## G0:0010008 9744/3112/124460/255231/911/84174/8938/926/4948/29887/4689/8992/3134/55652/302/3115/8766/
## G0:0098857
##          Count
## G0:0005774    182
## G0:0098852    161
## G0:0005765    160
## G0:0030667    135
## G0:0010008    197
## G0:0098857    141

```



NIT vs SFI

```
## Warning in summary(ego_NIT_vs_SFI): summary method to convert the object to
## data.frame is deprecated, please use as.data.frame instead.
```

ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
G0:0098793	presynapse	487/17653	491/19717	6.818448e-19	5.222931e-16	2.935521e-16
G0:0098978	glutamatergic synapse	349/17653	349/19717	1.207379e-17	4.624260e-15	2.599042e-15
G0:0097060	synaptic membrane	428/17653	432/19717	3.266557e-16	6.666000e-14	3.746591e-14
G0:0043025	neuronal cell body	490/17653	497/19717	3.751104e-16	6.666000e-14	3.746591e-14
G0:0005925	focal adhesion	402/17653	405/19717	4.351175e-16	6.666000e-14	3.746591e-14
G0:0005759	mitochondrial matrix	463/17653	469/19717	7.383140e-16	9.425809e-14	5.297727e-14

##

G0:0098793 5023/79772/143425/3738/5142/5582/4311/23767/10590/1006/5368/5414/116/1020/10874/55737/127

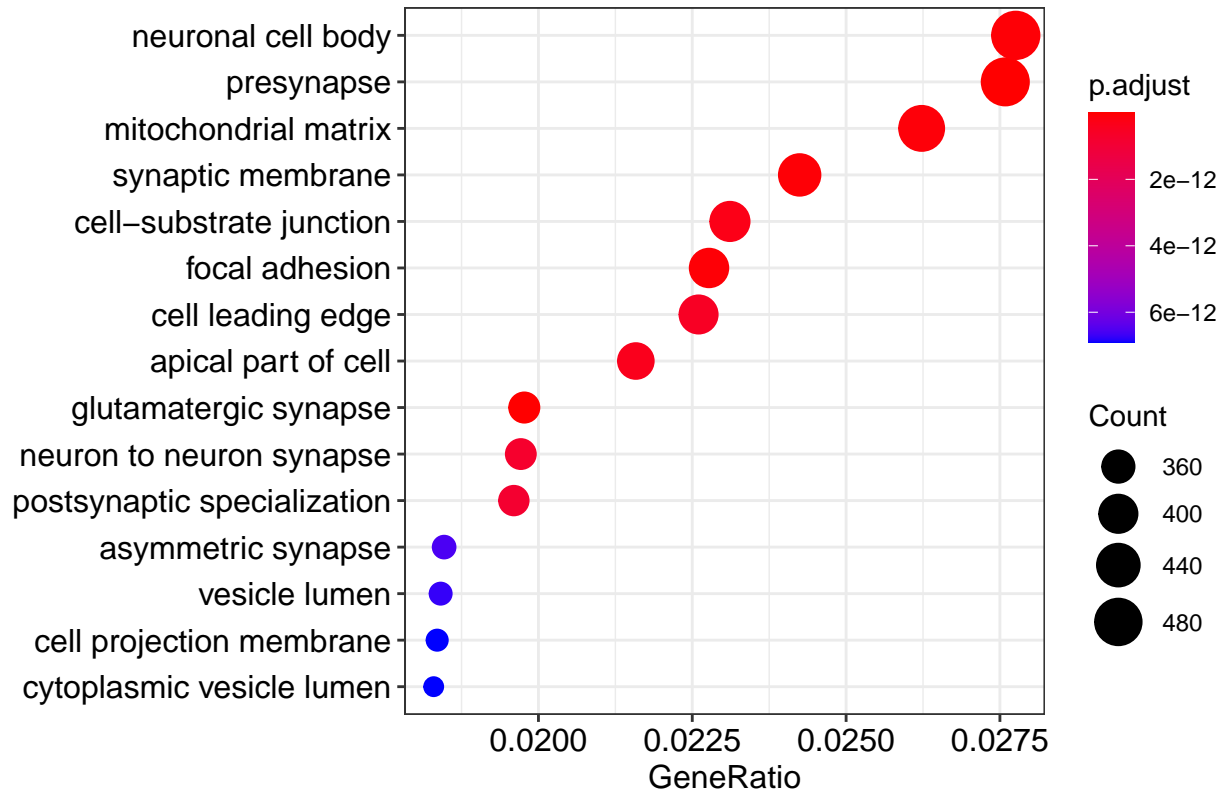
G0:0098978

G0:0097060

G0:0043025 8874/3676/4311/1233/10382/3065/57125/25828/5368/7043/79791/2806/29850/1

G0:0005925


```
## G0:0005759          92483/6648/9801/5163/65080/84274/54931/51642/55006/10667/686/51318/51103/2582
##                      Count
## G0:0098793         487
## G0:0098978         349
## G0:0097060         428
## G0:0043025         490
## G0:0005925         402
## G0:0005759         463
```

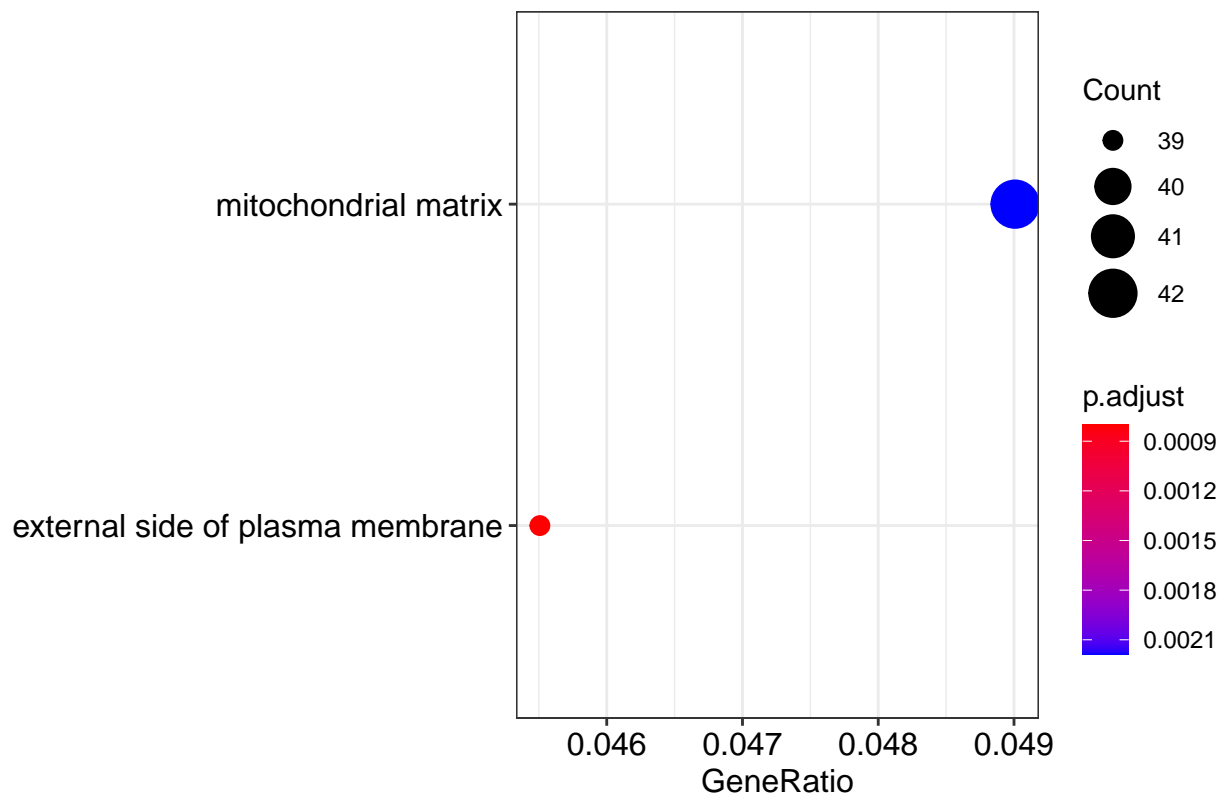


NIT vs SFI con los datos significativos

```
## Warning in summary(ego_NIT_vs_SFI_Sig): summary method to convert the object to
## data.frame is deprecated, please use as.data.frame instead.
```

```
##                      ID                      Description GeneRatio  BgRatio
## G0:0009897 G0:0009897 external side of plasma membrane  39/857 393/19717
## G0:0005759 G0:0005759          mitochondrial matrix    42/857 469/19717
##                      pvalue    p.adjust    qvalue
## G0:0009897 1.479241e-06 0.000798790 0.0007769907
## G0:0005759 8.098322e-06 0.002186547 0.0021268751
##
## G0:0009897          4283/100423062/931/973/5023/1493/940/3561/29851/10859
## G0:0005759 92483/6648/9801/5163/65080/84274/54931/51642/55006/10667/686/51318/51103/25828/122961/509
##                      Count
## G0:0009897         39
```

```
## G0:0005759    42
```

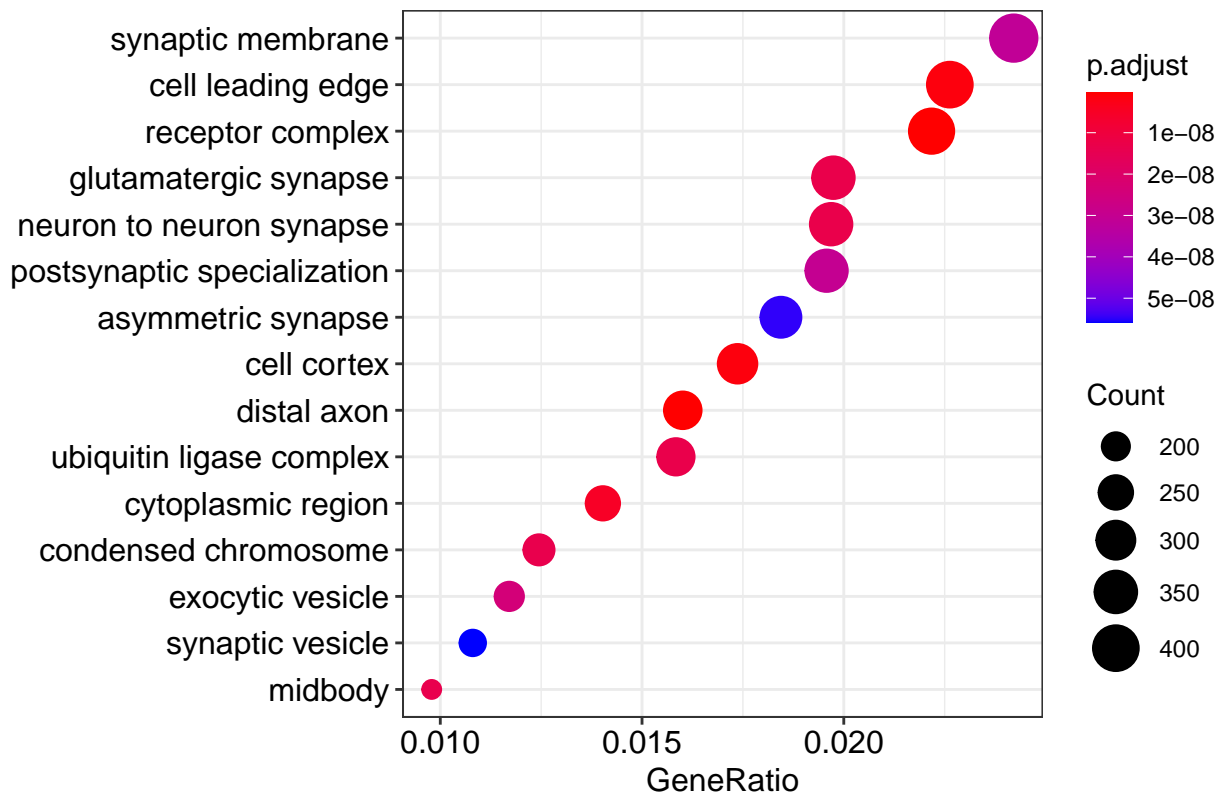


ELI vs SFI

```
## Warning in summary(ego_ELI_vs_SFI): summary method to convert the object to
## data.frame is deprecated, please use as.data.frame instead.
```

```
##          ID          Description GeneRatio  BgRatio      pvalue
## G0:0150034 G0:0150034      distal axon 283/17677 296/21659 4.632806e-13
## G0:0043235 G0:0043235      receptor complex 392/17677 419/21659 8.502792e-13
## G0:0031252 G0:0031252      cell leading edge 400/17677 430/21659 6.022935e-12
## G0:0005938 G0:0005938          cell cortex 307/17677 325/21659 7.625450e-12
## G0:0099568 G0:0099568      cytoplasmic region 248/17677 260/21659 3.127931e-11
## G0:0098978 G0:0098978      glutamatergic synapse 349/17677 375/21659 1.152573e-10
##          p.adjust      qvalue
## G0:0150034 3.413871e-10 2.295754e-10
## G0:0043235 3.413871e-10 2.295754e-10
## G0:0031252 1.530809e-09 1.029436e-09
## G0:0005938 1.530809e-09 1.029436e-09
## G0:0099568 5.023457e-09 3.378165e-09
## G0:0098978 1.324241e-08 8.905235e-09
##
## G0:0150034
## G0:0043235
## G0:0031252 ENSG00000161929/ENSG00000068831/ENSG00000102879/ENSG00000111913/ENSG00000159753/ENSG00000
```

```
## G0:0005938
## G0:0099568
## G0:0098978
##          Count
## G0:0150034  283
## G0:0043235  392
## G0:0031252  400
## G0:0005938  307
## G0:0099568  248
## G0:0098978  349
```



ELI vs SFI con los datos significativos

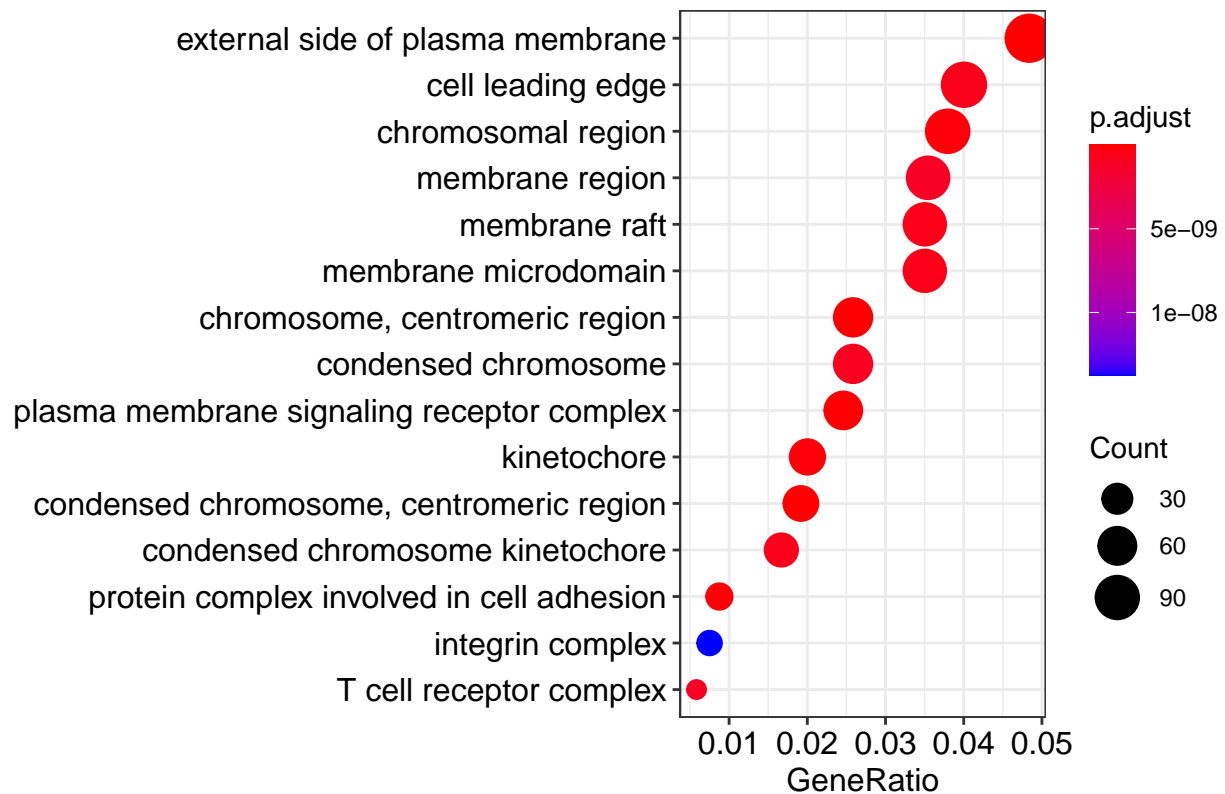
```
## Warning in summary(ego_ELI_vs_SFI_Sig): summary method to convert the object to
## data.frame is deprecated, please use as.data.frame instead.
```

```
##          ID          Description GeneRatio
## G0:0009897 G0:0009897 external side of plasma membrane 116/2398
## G0:0000779 G0:0000779 condensed chromosome, centromeric region 46/2398
## G0:0000775 G0:0000775 chromosome, centromeric region 62/2398
## G0:0098802 G0:0098802 plasma membrane signaling receptor complex 59/2398
## G0:0000776 G0:0000776 kinetochore 48/2398
## G0:0098687 G0:0098687 chromosomal region 91/2398
##          BgRatio      pvalue      p.adjust      qvalue
## G0:0009897 429/21659 1.748115e-20 1.232421e-17 1.034148e-17
```

```

## G0:0000779 127/21659 7.623932e-14 2.687436e-11 2.255079e-11
## G0:0000775 209/21659 1.624130e-13 3.816704e-11 3.202669e-11
## G0:0098802 198/21659 5.049480e-13 8.899708e-11 7.467915e-11
## G0:0000776 145/21659 1.061955e-12 1.497356e-10 1.256460e-10
## G0:0098687 385/21659 1.352512e-12 1.589202e-10 1.333530e-10
##
## G0:0009897 ENSG00000007312/ENSG00000112486/ENSG00000104921/ENSG00000096996/ENSG00000198851/ENSG00000000000
## G0:0000779
## G0:0000775
## G0:0098802
## G0:0000776
## G0:0098687
##
## Count
## G0:0009897 116
## G0:0000779 46
## G0:0000775 62
## G0:0098802 59
## G0:0000776 48
## G0:0098687 91

```



Pathway de ReactomePA con ClusterProfiler Preparamos lista de genes analizados

```

## NIT_vs_ELI NIT_vs_SFI ELI_vs_SFI
## 8318 1608 3670

```

Los genes a tener en cuenta, tienen al menos una anotación en Gene Ontology.

Con el paquete “clusterProfiler” y siguiendo el Pathway de ReactomePA, realizamos el análisis de la significación biológica.

```
## Reading KEGG annotation online:
##
## Reading KEGG annotation online:

## #####
## Comparison:  NIT_vs_ELI
##           ID           Description GeneRatio  BgRatio      pvalue
## hsa03050 hsa03050       Proteasome    37/2370   46/7984 1.331696e-12
## hsa05340 hsa05340 Primary immunodeficiency 30/2370   37/7984 1.302171e-10
## hsa05012 hsa05012       Parkinson disease 117/2370  249/7984 3.820165e-09
## hsa05330 hsa05330       Allograft rejection 28/2370   37/7984 9.376841e-09
## hsa05010 hsa05010       Alzheimer disease 159/2370  367/7984 9.720778e-09
## hsa05416 hsa05416       Viral myocarditis 38/2370   59/7984 3.271742e-08
##           p.adjust      qvalue
## hsa03050 4.301379e-10 3.364285e-10
## hsa05340 2.103006e-08 1.644848e-08
## hsa05012 4.113044e-07 3.216981e-07
## hsa05330 6.279622e-07 4.911551e-07
## hsa05010 6.279622e-07 4.911551e-07
## hsa05416 1.755777e-06 1.373265e-06
##
## hsa03050
## hsa05340
## hsa05012
## hsa05330
## hsa05010 5330/80326/5293/489/7124/841/7471/51384/778/5682/10383/5702/8851/5707/4702/84790/3028/3551/
## hsa05416
##           Count
## hsa03050      37
## hsa05340      30
## hsa05012     117
## hsa05330      28
## hsa05010     159
## hsa05416      38

## #####
## Comparison:  NIT_vs_SFI
##           ID           Description GeneRatio
## hsa05340 hsa05340       Primary immunodeficiency 13/396
## hsa04672 hsa04672 Intestinal immune network for IgA production 10/396
## hsa04640 hsa04640       Hematopoietic cell lineage 15/396
## hsa04660 hsa04660       T cell receptor signaling pathway 15/396
## hsa04146 hsa04146       Peroxisome 13/396
## hsa04658 hsa04658       Th1 and Th2 cell differentiation 13/396
##           BgRatio      pvalue      p.adjust      qvalue
## hsa05340 37/7984 1.085716e-08 3.224577e-06 3.165720e-06
## hsa04672 47/7984 7.988542e-05 8.558234e-03 8.402023e-03
## hsa04640 98/7984 8.644680e-05 8.558234e-03 8.402023e-03
## hsa04660 103/7984 1.539506e-04 1.143083e-02 1.122219e-02
```

```
## hsa04146 83/7984 2.007277e-04 1.192322e-02 1.170559e-02
## hsa04658 92/7984 5.651121e-04 2.797305e-02 2.746247e-02
##
##                                     geneID
## hsa05340 973/7374/3561/29851/115650/5788/915/7535/6890/930/23495/3932/3718
## hsa04672 608/940/3112/29851/115650/7852/3676/23495/9020/8174
## hsa04640 931/3112/1440/924/3568/3676/4311/921/915/930/960/951/952/1441/3566
## hsa04660 1493/10125/3458/940/29851/3702/5788/915/7535/84433/9402/919/3932/9020/3937
## hsa04146 54677/5193/6648/1891/3417/847/5052/10478/23600/8800/6647/8504/3295
## hsa04658 3458/3561/3112/915/6775/7535/3560/6772/919/3932/30009/3718/3566
## Count
## hsa05340 13
## hsa04672 10
## hsa04640 15
## hsa04660 15
## hsa04146 13
## hsa04658 13
```

```
## #####
```

```
## Comparison: ELI_vs_SFI
```

```
## ID Description GeneRatio BgRatio
## hsa04640 hsa04640 Hematopoietic cell lineage 46/1084 98/7984
## hsa05340 hsa05340 Primary immunodeficiency 23/1084 37/7984
## hsa04514 hsa04514 Cell adhesion molecules (CAMs) 50/1084 147/7984
## hsa04659 hsa04659 Th17 cell differentiation 37/1084 107/7984
## hsa04670 hsa04670 Leukocyte transendothelial migration 37/1084 113/7984
## hsa04810 hsa04810 Regulation of actin cytoskeleton 57/1084 213/7984
## pvalue p.adjust qvalue
## hsa04640 9.449124e-16 2.995372e-13 2.377201e-13
## hsa05340 8.359132e-12 1.324922e-09 1.051491e-09
## hsa04514 1.603551e-10 1.694419e-08 1.344732e-08
## hsa04659 2.390240e-08 1.894265e-06 1.503335e-06
## hsa04670 1.213424e-07 7.693109e-06 6.105439e-06
## hsa04810 1.787808e-07 8.624406e-06 6.844539e-06
##
```

```
## hsa04640 1380/2208
## hsa05340
## hsa04514 6402/3385/3683/6693/3695/3112/5133/914/933/5817/
## hsa04659
## hsa04670
## hsa04810 3681/3683/5880/3695/5293/7409/7074/623/81/9138/5305/3676/3682/9564/3071/9459/81624/3687/898
## Count
## hsa04640 46
## hsa05340 23
## hsa04514 50
## hsa04659 37
## hsa04670 37
## hsa04810 57
```

Mostramos para cada comparación, los genes más expresados diferencialmente y con la descripción según su función biológica (“Gene Ontology”):

NIT vs ELI

ID	Description	GeneRatio	BgRatio	pvalue	p.adjust	qvalue
hsa03050	Proteasome	37/2370	46/7984	1.33169618139426e-12	4.30137866590346e-10	3.364285
hsa05340	Primary immunodeficiency	30/2370	37/7984	1.30217113758605e-10	2.10300638720148e-08	1.644847
hsa05012	Parkinson disease	117/2370	249/7984	3.82016457986877e-09	4.11304386432538e-07	3.216980
hsa05330	Allograft rejection	28/2370	37/7984	9.37684061322095e-09	6.27962242520048e-07	4.911550
hsa05010	Alzheimer disease	159/2370	367/7984	9.72077774798836e-09	6.27962242520048e-07	4.911550
hsa05416	Viral myocarditis	38/2370	59/7984	3.2717420650491e-08	1.75577715875078e-06	1.373265

NIT vs SFI

Table 2: Ta

ID	Description	GeneRatio	BgRatio	pvalue	p.adjust
hsa05340	Primary immunodeficiency	13/396	37/7984	1.08571630893433e-08	3.2245774375
hsa04672	Intestinal immune network for IgA production	10/396	47/7984	7.98854193337924e-05	0.0085582335
hsa04640	Hematopoietic cell lineage	15/396	98/7984	8.64468034059993e-05	0.0085582335
hsa04660	T cell receptor signaling pathway	15/396	103/7984	0.000153950578032668	0.0114308304
hsa04146	Peroxisome	13/396	83/7984	0.00020072765592283	0.0119232227
hsa04658	Th1 and Th2 cell differentiation	13/396	92/7984	0.000565112106842106	0.0279730492

NIT vs ELI

ID	Description	GeneRatio	BgRatio	pvalue	p.adjust
hsa04640	Hematopoietic cell lineage	46/1084	98/7984	9.44912362283061e-16	2.9953721884373e-13
hsa05340	Primary immunodeficiency	23/1084	37/7984	8.35913200088488e-12	1.32492242214025e-09
hsa04514	Cell adhesion molecules (CAMs)	50/1084	147/7984	1.60355126228152e-10	1.69441916714414e-08
hsa04659	Th17 cell differentiation	37/1084	107/7984	2.39023985222337e-08	1.89426508288702e-06
hsa04670	Leukocyte transendothelial migration	37/1084	113/7984	1.21342409863878e-07	7.69310878536984e-06
hsa04810	Regulation of actin cytoskeleton	57/1084	213/7984	1.78780799852586e-07	8.62440550239862e-06

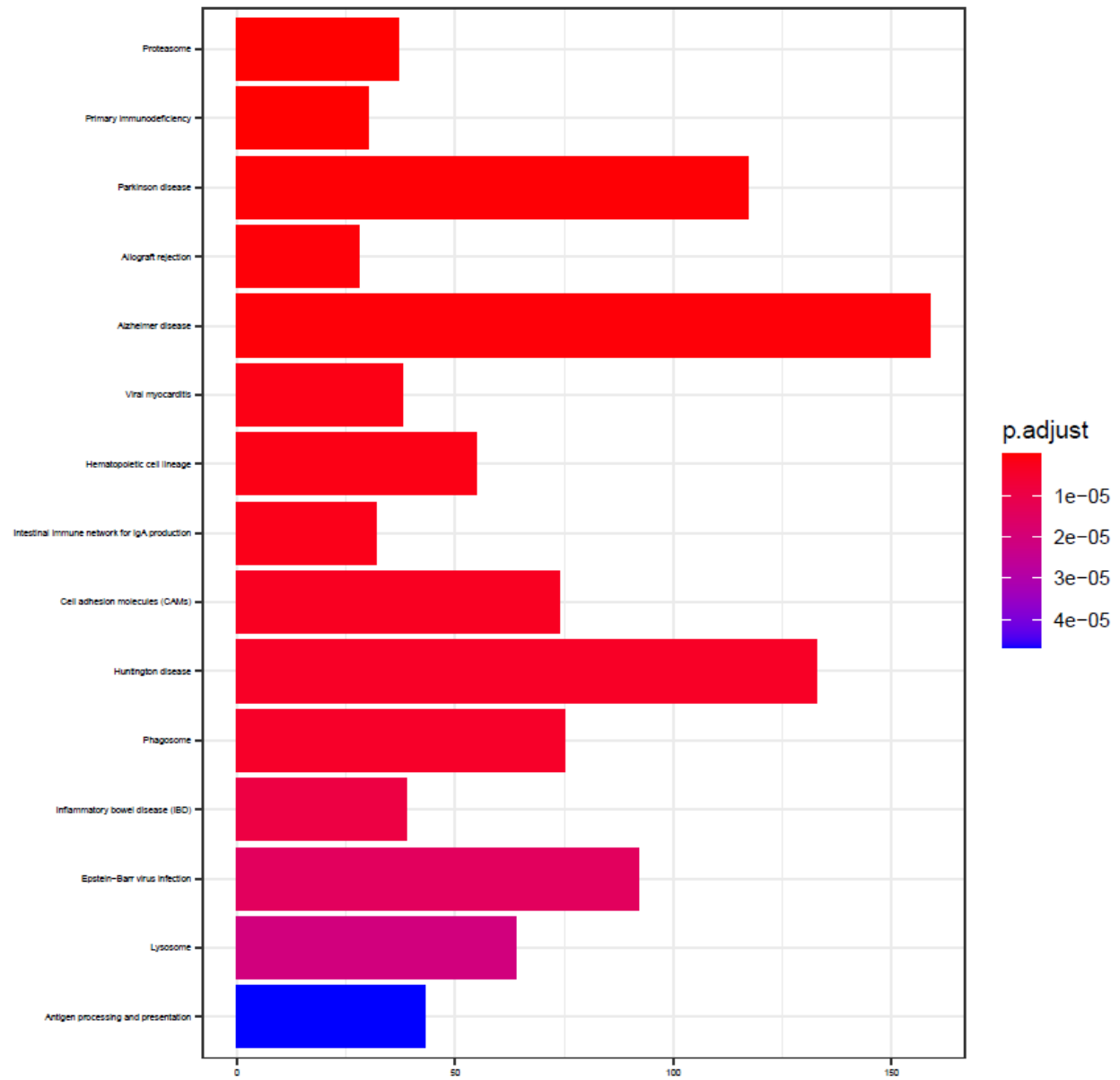
Los datos al completo los encontramos almacenados en los respectivos archivos .csv para cada comparación.
Resultados

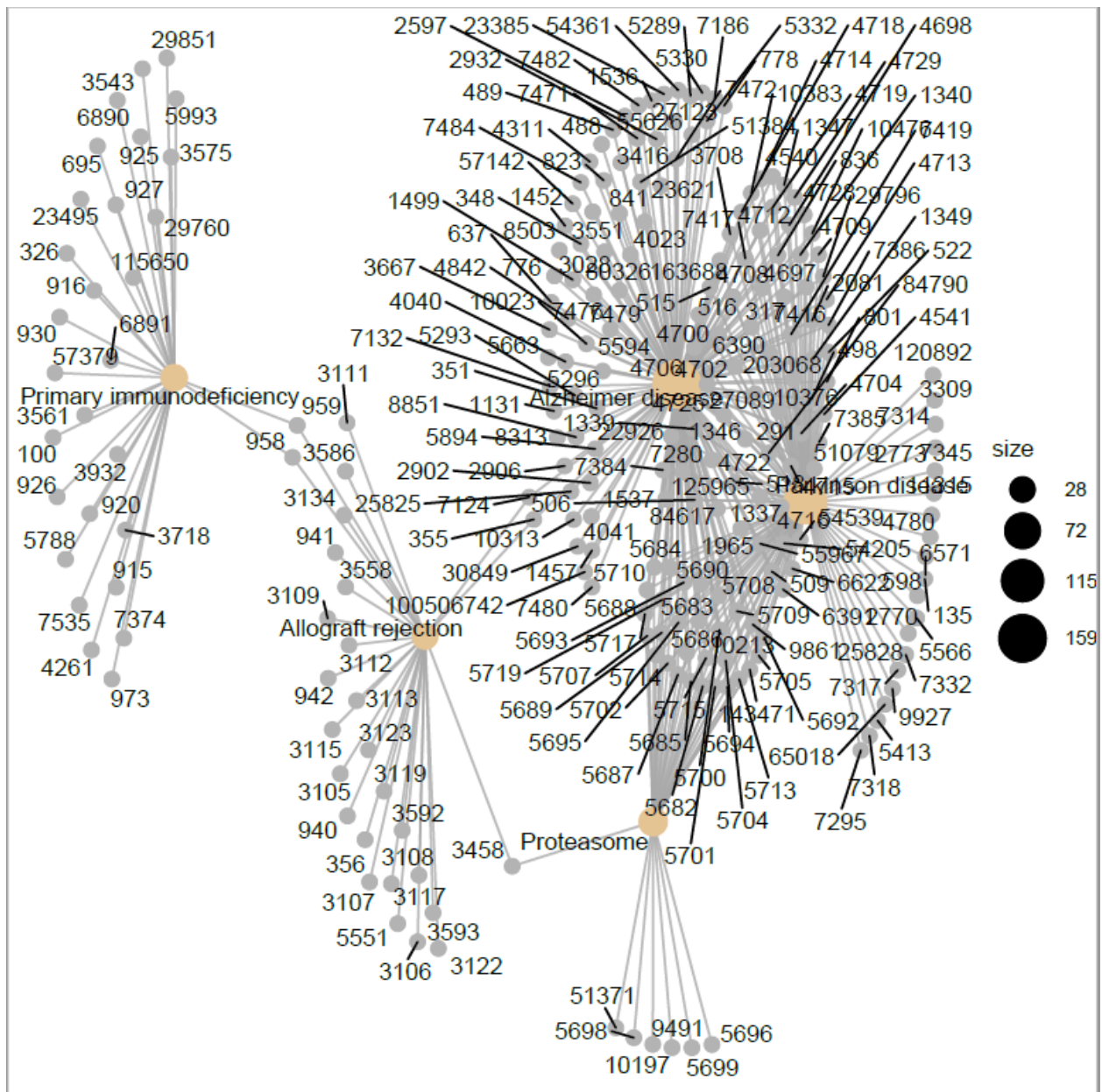
Obtenemos resultados significativos al realizar el análisis con ambos procedimientos y según la comparación entre grupos.

El Gene Set Enrichment Analysis mostró genes que se expresan de manera diferencial según grupo y comparación.

Grupo NIT vs Grupo ELI Es la comparación donde encontramos una mayor expresión diferencial. Posiblemente debido al origen distinto de los grupos: NIT (tejidos no infiltrados) vs ELI (infiltraciones extensas). Están mayormente expresados de manera diferencial genes relacionados con el ciclo celular y el rechazo a producido ante un transplante o infiltración como en este caso (Allograft rejection). También hay una expresión diferencial para genes relacionados con patologías y enfermedades como Parkinson, Alzheimer o Miocarditis.

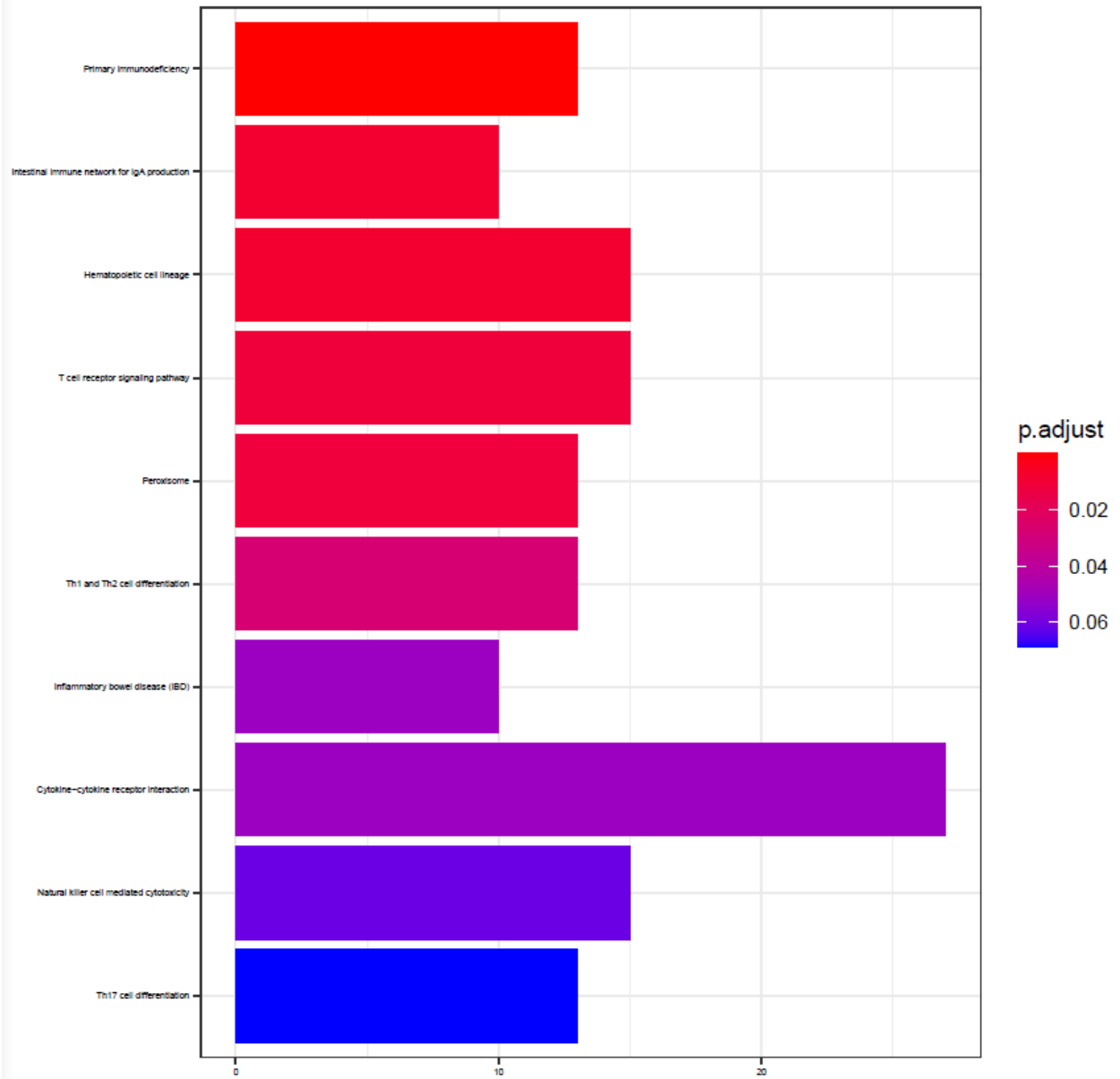
Cluster Pathway Analysis for NIT_vs_ELI. Barplot

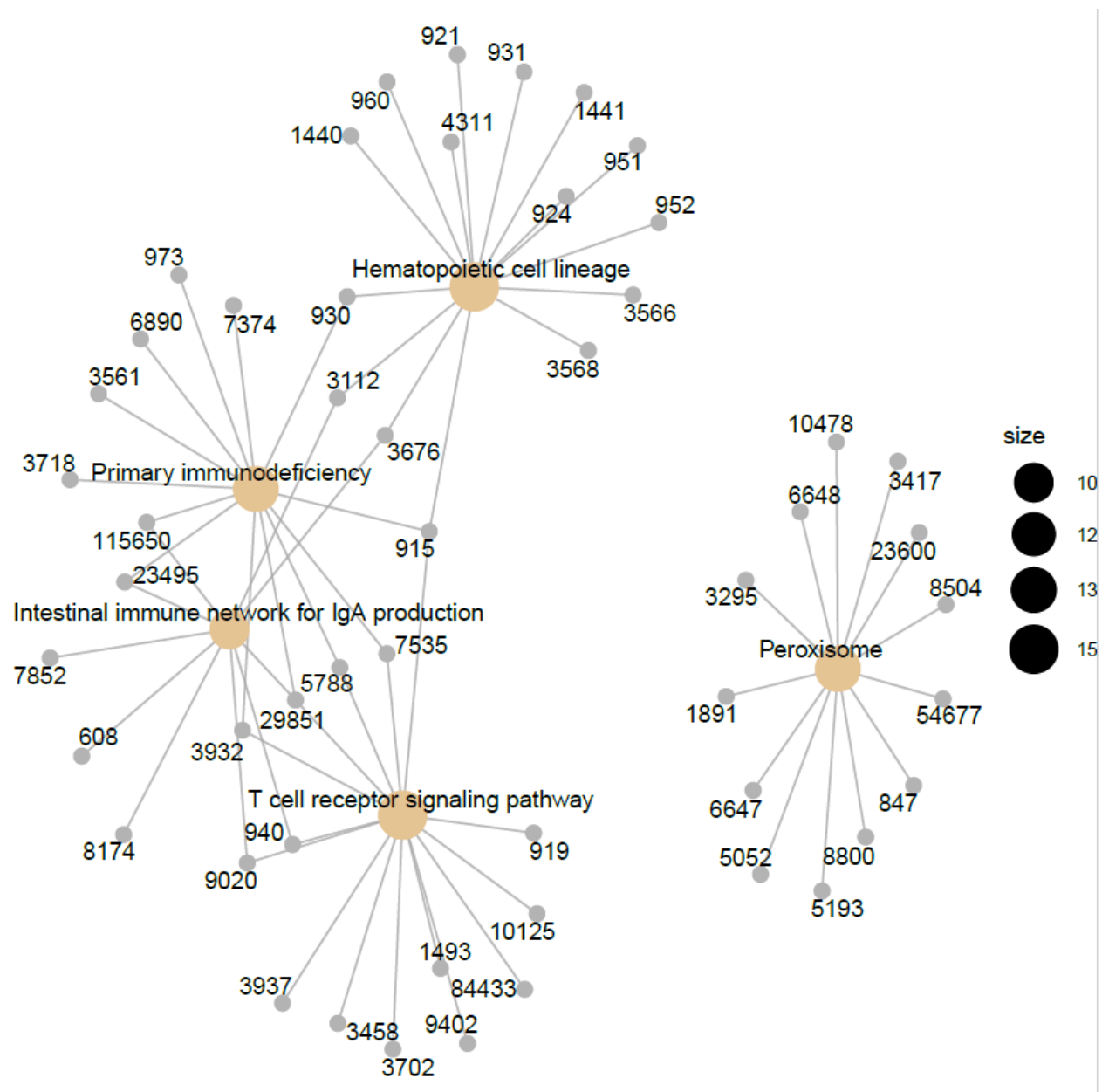




Grupo NIT vs Grupo SFI Entre el grupo NIT y SFI encontramos genes expresados de forma diferencial. Con especial atención a aquellos genes relacionados con el sistema inmune y su respuesta humoral.

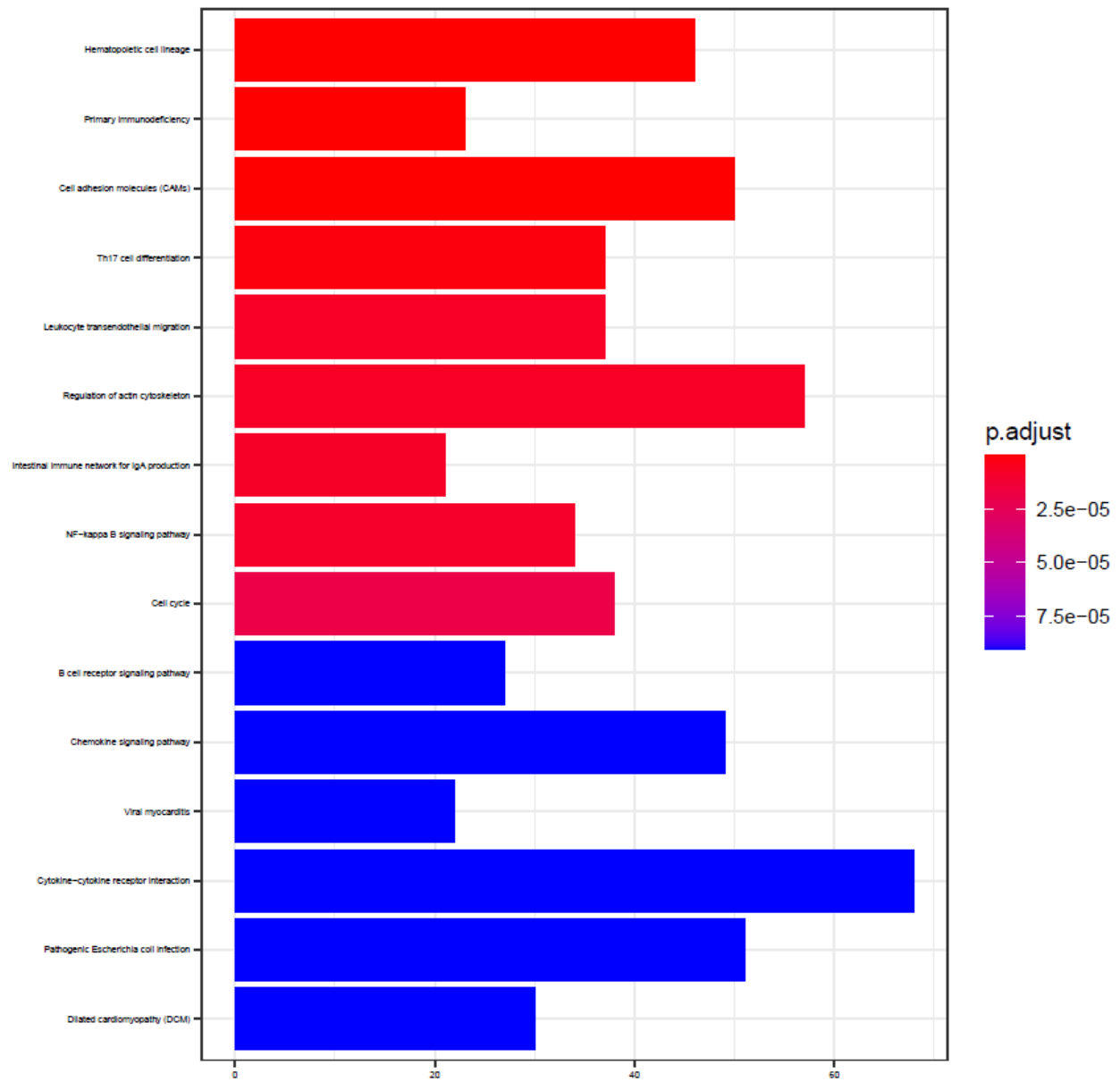
Cluster Pathway Analysis for NIT_vs_SFI. Barplot

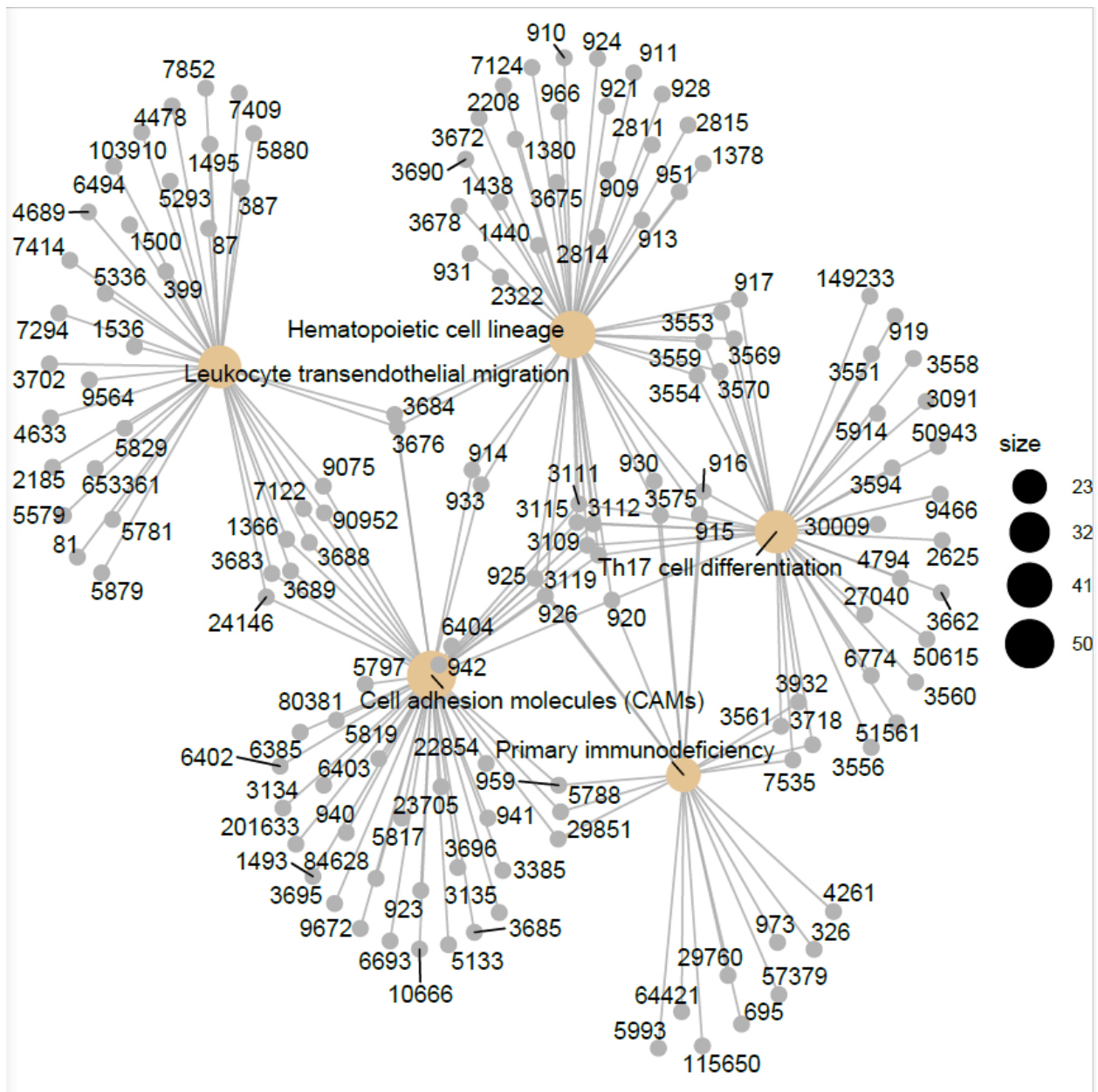




Grupo ELI vs Grupo SFI Finalmente, entre el grupo ELI y SFI encontramos diferencias a nivel de respuesta inmune y proceso inflamatorio, aunque no tantas como en los casos anteriores donde comparábamos tejido no infiltrado con distintos tejidos sí infiltrados (ELI y SFI).

Cluster Pathway Analysis for ELI_vs_SFI. Barplot





Discusión

El análisis de los datos de expresión de RNA es una herramienta útil y poderosa.

Mediante el análisis con R y los paquetes de BioConductor, se puede obtener y realizar distintos procesos. Desde el filtraje de los datos, transformación de estos, visualización en gráficos y su significación biológica, entre otras. Permitiendo realizar un completo análisis de los datos.

No obstante, hay algunas limitaciones vistas durante este análisis.

Por ejemplo, factores técnicos como el elevado NA en el mapeado al realizar la anotación aunque no ha sido un impedimento para el proceso. También el conteo bajo en algunos genes.

De todas maneras, el análisis se puede concluir como positivo. Ayudando a confirmar hipótesis iniciales acerca de qué diferencias de expresión se pueden encontrar en distintos tejidos según infiltración tiroidea y

que genes tiene expresados.

Apéndice

El código R usado en este documento R Markdown:

```
## ----setup, include=FALSE-----
knitr::opts_chunk$set(echo = TRUE)

## ----4.1, include=FALSE-----
setwd(".")
dir.create("data")
dir.create("results")

## ----include=FALSE-----
library(knitr)
library(readr)
library(colorspace)
library(gplots)
library(ggplot2)
library(DESeq2)
library(AnnotationDbi)
library(org.Hs.eg.db)
library(hugene21sttranscriptcluster.db)
library(clusterProfiler)
library(dplyr)
library(limma)
library(hexbin)
library(pheatmap)
library(RColorBrewer)

## ----include=FALSE-----
counts <- read_delim("C:/Users/rubenuoc/Desktop/UOC/M0157/PEC2/PEC_02/data/counts.csv",
                    ";", escape_double = FALSE, trim_ws = TRUE)

targets <- read_csv("C:/Users/rubenuoc/Desktop/UOC/M0157/PEC2/PEC_02/data/targets.csv")

## ----include=FALSE-----
set.seed(280520)

targets_10_g <- lapply(split(targets, targets$Grupo_analisis),
                      function(subdf) subdf[sample(1:nrow(subdf), 10),])

targets_10_g <- do.call('rbind', targets_10_g)
```

```

## ----echo=FALSE-----
head(targets_10_g)

## ----include=FALSE-----
df <- counts[, targets_10_g$Sample_Name]

# Para que podamos realizar con mayor falicidad el análisis posterior, cambiaremos el nombre de cada fi

# Añadimos la columna X1 a nuestro dataframe

df <- cbind(df, X1 = counts$X1)

# Cambiamos orden para tenerla primera y poder cambiar el

df <- df[, c(31, 1:30)]

countdata <- df[, -1]
rownames(countdata) <- df[, 1]

## ----include=FALSE-----
dds <- DESeqDataSetFromMatrix(countData = countdata, colData = targets_10_g, design = ~Group)

## ----echo=FALSE-----
nrow(dds)

dds <- dds[ rowSums(counts(dds)) > 1, ]
nrow(dds)

## ----echo=FALSE-----
vsd <- vst(dds, blind = FALSE) # blind = FALSE para evitar que las variables en el diseño del objeto (G
head(assay(vsd), 3)

## ----echo=FALSE-----
colData(vsd)

## ----echo=FALSE-----
rld <- rlog(dds, blind = FALSE)
head(assay(rld), 3)

## ----include=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"))

## ----echo=FALSE-----
colnames(df)[1:2] <- c("x", "y")

ggplot(df, aes(x = x, y = y)) + geom_hex(bins = 80) +
  coord_fixed() + facet_grid( . ~ transformation)

## ----include=FALSE-----
sampleDists <- dist(t(assay(vsd)))
sampleDists

## ----echo=FALSE-----

sampleDistMatrix <- as.matrix( sampleDists )
rownames(sampleDistMatrix) <- paste( vsd$dex, vsd$cell, sep = " - " )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
          clustering_distance_rows = sampleDists,
          clustering_distance_cols = sampleDists,
          col = colors)

## ----echo=FALSE-----
plotPCA(vsd, intgroup = c("Group"))

## ----include=FALSE-----
dds <- DESeq(dds)

## ----echo=FALSE-----
# Realizamos las distintas comparaciones usando 'contrast' para cada grupo, con su respectivo factor:

# NIT vs ELI

res_NIT_vs_ELI <- results(dds, contrast = c("Group", "NIT", "ELI"))

NIT_vs_ELI_rownames <- row.names(res_NIT_vs_ELI)

summary(res_NIT_vs_ELI)

## ----echo=FALSE-----
# NIT vs SFI

```



```

res_NIT_vs_SFI <- results(dds, contrast = c("Group", "NIT", "SFI"))

NIT_vs_SFI_rownames <- row.names(res_NIT_vs_SFI)

summary(res_NIT_vs_SFI)

## ----echo=FALSE-----
# ELI vs SFI

res_ELI_vs_SFI <- results(dds, contrast = c("Group", "ELI", "SFI"))

ELI_vs_SFI_rownames <- row.names(res_ELI_vs_SFI)

summary(res_ELI_vs_SFI)

## ----include=FALSE-----
sum(res_NIT_vs_ELI$padj < 0.1, na.rm=TRUE)

sum(res_NIT_vs_SFI$padj < 0.1, na.rm=TRUE)

sum(res_ELI_vs_SFI$padj < 0.1, na.rm=TRUE)

## ----include=FALSE-----
res_NIT_vs_ELI_Sig <- subset(res_NIT_vs_ELI, padj < 0.1)

res_NIT_vs_SFI_Sig <- subset(res_NIT_vs_SFI, padj < 0.1)

res_ELI_vs_SFI_Sig <- subset(res_ELI_vs_SFI, padj < 0.1)

## ----echo=FALSE-----
head(res_NIT_vs_ELI_Sig[ order(res_NIT_vs_ELI_Sig$log2FoldChange), ])

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

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

## ----echo=FALSE-----
head(res_NIT_vs_ELI_Sig[ order(res_NIT_vs_ELI_Sig$log2FoldChange, decreasing = TRUE), ])

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

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

## ----include=FALSE-----
columns(org.Hs.eg.db)

```

```

## ----echo=FALSE-----
# NIT vs ELI Sig

tmp_NIT_vs_ELI_Sig=gsub("\\\\.*","",row.names(res_NIT_vs_ELI_Sig)) # borrar '.' de ENSEMBL

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

res_NIT_vs_ELI_Sig$entrez <- mapIds(org.Hs.eg.db,
                                   keys=tmp_NIT_vs_ELI_Sig,
                                   column="ENTREZID",
                                   keytype="ENSEMBL",
                                   multiVals="first")

res_NIT_vs_ELI_Sig_Ordered <- res_NIT_vs_ELI_Sig[order(res_NIT_vs_ELI_Sig$pvalue),]

head(res_NIT_vs_ELI_Sig_Ordered)

# NIT vs SFI Sig

tmp_NIT_vs_SFI_Sig=gsub("\\\\.*","",row.names(res_NIT_vs_SFI_Sig)) # borrar '.' de ENSEMBL

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

res_NIT_vs_SFI_Sig$entrez <- mapIds(org.Hs.eg.db,
                                   keys=tmp_NIT_vs_SFI_Sig,
                                   column="ENTREZID",
                                   keytype="ENSEMBL",
                                   multiVals="first")

res_NIT_vs_SFI_Sig_Ordered <- res_NIT_vs_SFI_Sig[order(res_NIT_vs_SFI_Sig$pvalue),]

head(res_NIT_vs_SFI_Sig_Ordered)

# ELI vs SFI Sig

tmp_ELI_vs_SFI_Sig=gsub("\\\\.*","",row.names(res_ELI_vs_SFI_Sig)) # borrar '.' de ENSEMBL

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

res_ELI_vs_SFI_Sig$entrez <- mapIds(org.Hs.eg.db,
                                   keys=tmp_ELI_vs_SFI_Sig,

```

```

        column="ENTREZID",
        keytype="ENSEMBL",
        multiVals="first")

res_ELI_vs_SFI_Sig_Ordered <- res_ELI_vs_SFI_Sig[order(res_ELI_vs_SFI_Sig$pvalue),]

head(res_ELI_vs_SFI_Sig_Ordered)

## -----echo=FALSE-----
# Para obtener las keys

# NIT vs ELI

tmp_NIT_vs_ELI=gsub("\\..*", "", row.names(res_NIT_vs_ELI)) # borrar '.' de ENSEMBL

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

res_NIT_vs_ELI$entrez <- mapIds(org.Hs.eg.db,
                               keys=tmp_NIT_vs_ELI,
                               column="ENTREZID",
                               keytype="ENSEMBL",
                               multiVals="first")

res_NIT_vs_ELI_Ordered <- res_NIT_vs_ELI[order(res_NIT_vs_ELI$pvalue),]

head(res_NIT_vs_ELI_Ordered)

# NIT vs SFI

res_NIT_vs_SFI_DF <- as.data.frame(res_NIT_vs_SFI)

tmp_NIT_vs_SFI=gsub("\\..*", "", row.names(res_NIT_vs_SFI)) # borrar '.' de ENSEMBL

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

res_NIT_vs_SFI$entrez <- mapIds(org.Hs.eg.db,
                               keys=tmp_NIT_vs_SFI,
                               column="ENTREZID",
                               keytype="ENSEMBL",
                               multiVals="first")

res_NIT_vs_SFI_Ordered <- res_NIT_vs_SFI[order(res_NIT_vs_SFI$pvalue),]

head(res_NIT_vs_SFI_Ordered)

```

```

# ELI vs SFI

res_ELI_vs_SFI_DF <- as.data.frame(res_ELI_vs_SFI)

tmp_ELI_vs_SFI=gsub("\\\\.\\.*", "", row.names(res_ELI_vs_SFI_DF)) # borrar '.' de ENSEMBL

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

res_ELI_vs_SFI$entrez <- mapIds(org.Hs.eg.db,
                                keys=tmp_ELI_vs_SFI,
                                column="ENTREZID",
                                keytype="ENSEMBL",
                                multiVals="first")

res_ELI_vs_SFI_Ordered <- res_ELI_vs_SFI[order(res_ELI_vs_SFI$pvalue),]

head(res_ELI_vs_SFI_Ordered)

## ----Significativos, include=FALSE-----
# Exportamos los resultados significativos

# NIT_vs_ELI

res_NIT_vs_ELI_Sig_Ordered_DF <- as.data.frame(res_NIT_vs_ELI_Sig_Ordered)

write.csv(res_NIT_vs_ELI_Sig_Ordered, file = "./results/NIT_vs_ELI_Sig.csv")

# NIT_vs_SFI

res_NIT_vs_SFI_Sig_Ordered_DF <- as.data.frame(res_NIT_vs_SFI_Sig_Ordered)

write.csv(res_NIT_vs_SFI_Sig_Ordered, file = "./results/NIT_vs_SFI_Sig.csv")

# ELI_vs_SFI

res_ELI_vs_SFI_Sig_Ordered_DF <- as.data.frame(res_ELI_vs_SFI_Sig_Ordered)

write.csv(res_ELI_vs_SFI_Sig_Ordered, file = "./results/ELI_vs_SFI_Sig.csv")

## ----include=FALSE-----
# Exportamos los resultados

# NIT_vs_ELI

res_NIT_vs_ELI_Ordered_DF <- as.data.frame(res_NIT_vs_ELI_Ordered)

```

```

write.csv(res_NIT_vs_ELI_Ordered, file = "./results/NIT_vs_ELI.csv")

# NIT_vs_SFI

res_NIT_vs_SFI_Ordered_DF <- as.data.frame(res_NIT_vs_SFI_Ordered)

write.csv(res_NIT_vs_SFI_Ordered, file = "./results/NIT_vs_SFI.csv")

# ELI_vs_SFI

res_ELI_vs_SFI_Ordered_DF <- as.data.frame(res_ELI_vs_SFI_Ordered)

write.csv(res_ELI_vs_SFI_Ordered, file = "./results/ELI_vs_SFI.csv")

## ----echo=FALSE-----
# Combinamos las 3 comparaciones para el posterior Diagrama de Venn

comb <- c(rownames(res_NIT_vs_ELI_Sig_Ordered_DF), rownames(res_NIT_vs_ELI_Sig_Ordered_DF), rownames(res_NIT_vs_ELI_Sig_Ordered_DF))

res_NIT_vs_ELI_venn <- comb %in% rownames(res_NIT_vs_ELI_Sig_Ordered_DF)
res_NIT_vs_SFI_venn <- comb %in% rownames(res_NIT_vs_ELI_Sig_Ordered_DF)
res_ELI_vs_SFI_venn <- comb %in% rownames(res_ELI_vs_SFI_Sig_Ordered_DF)

counts_venn <- cbind(res_NIT_vs_ELI_venn, res_NIT_vs_SFI_venn, res_ELI_vs_SFI_venn)

results_venn <- vennCounts(counts_venn)

vennDiagram (results_venn, cex=1, names = c("NIT vs ELI", "NIT vs SFI", "ELI vs SFI"), circle.col = c("red", "green", "blue"),
title("Genes en común en las 3 comparaciones")

## ----echo=FALSE-----
# Con el paquete 'ClusterProfiler'

tmp_NIT_vs_ELI=gsub("\\\\.\\.", "", row.names(res_NIT_vs_ELI_Ordered_DF)) # borrar '.' de ENSEMBL

ego_NIT_vs_ELI <- enrichGO(gene          = res_NIT_vs_ELI_Ordered_DF$entrez,
  OrgDb          = org.Hs.eg.db,
  keyType        = 'ENTREZID',
  ont            = "CC",
  pAdjustMethod  = "BH",
  pvalueCutoff   = 0.01,
  qvalueCutoff   = 0.05)
head(summary(ego_NIT_vs_ELI))

dotplot(ego_NIT_vs_ELI, showCategory = 15)

## ----echo=FALSE-----
ego_NIT_vs_ELI_Sig <- enrichGO(gene          = res_NIT_vs_ELI_Sig_Ordered_DF$entrez,

```

```

        OrgDb          = org.Hs.eg.db,
        keyType        = 'ENTREZID',
        ont             = "CC",
        pAdjustMethod  = "BH",
        pvalueCutoff   = 0.01,
        qvalueCutoff   = 0.05)
head(summary(ego_NIT_vs_ELI_Sig))

dotplot(ego_NIT_vs_ELI_Sig, showCategory = 15)

## ----echo=FALSE-----
tmp_NIT_vs_SFI=gsub("\\\\.*", "", row.names(res_NIT_vs_SFI_Ordered_DF)) # borrar '.' de ENSEMBL

ego_NIT_vs_SFI <- enrichGO(gene          = res_NIT_vs_SFI_Ordered_DF$entrez,
        OrgDb          = org.Hs.eg.db,
        keyType        = 'ENTREZID',
        ont             = "CC",
        pAdjustMethod  = "BH",
        pvalueCutoff   = 0.01,
        qvalueCutoff   = 0.05)
head(summary(ego_NIT_vs_SFI))

dotplot(ego_NIT_vs_SFI, showCategory = 15)

## ----echo=FALSE-----
ego_NIT_vs_SFI_Sig <- enrichGO(gene          = res_NIT_vs_SFI_Sig_Ordered_DF$entrez,
        OrgDb          = org.Hs.eg.db,
        keyType        = 'ENTREZID',
        ont             = "CC",
        pAdjustMethod  = "BH",
        pvalueCutoff   = 0.01,
        qvalueCutoff   = 0.05)
head(summary(ego_NIT_vs_SFI_Sig))

dotplot(ego_NIT_vs_SFI_Sig, showCategory = 15)

## ----echo=FALSE-----
# En este caso utilizaremos como keyType 'ENSEMBL' en lugar de ENTREZID, para ver si hay diferencias.

tmp_ELI_vs_SFI=gsub("\\\\.*", "", row.names(res_ELI_vs_SFI_Ordered_DF)) # borrar '.' de ENSEMBL

ego_ELI_vs_SFI <- enrichGO(gene          = tmp_ELI_vs_SFI,
        OrgDb          = org.Hs.eg.db,
        keyType        = 'ENSEMBL',
        ont             = "CC",
        pAdjustMethod  = "BH",
        pvalueCutoff   = 0.01,
        qvalueCutoff   = 0.05)
head(summary(ego_ELI_vs_SFI))

```

```

dotplot(ego_ELI_vs_SFI, showCategory = 15)

## ----echo=FALSE-----
tmp_ELI_vs_SFI_Sig=gsub("\\\\.\\.*", "", row.names(res_ELI_vs_SFI_Sig_Ordered_DF)) # borrar '.' de ENSEMBL

ego_ELI_vs_SFI_Sig <- enrichGO(gene          = tmp_ELI_vs_SFI_Sig,
                               OrgDb         = org.Hs.eg.db,
                               keyType       = 'ENSEMBL',
                               ont           = "CC",
                               pAdjustMethod = "BH",
                               pvalueCutoff  = 0.01,
                               qvalueCutoff  = 0.05)
head(summary(ego_ELI_vs_SFI_Sig))

dotplot(ego_ELI_vs_SFI_Sig, showCategory = 15)

## ----echo=FALSE-----

listOfTables <- list(NIT_vs_ELI = res_NIT_vs_ELI_Sig_Ordered_DF, NIT_vs_SFI = res_NIT_vs_SFI_Sig_Ordered_DF)

listOfSelected <- list()

for (i in 1:length(listOfTables)){

  topTab <- listOfTables[[i]]

  whichGenes<-topTab$padj<0.1
  selectedIDs <- gsub("\\\\.\\.*", "", row.names(topTab)[whichGenes])
  EntrezIDs <- topTab$entrez
  listOfSelected[[i]] <- EntrezIDs
  names(listOfSelected)[i] <- names(listOfTables)[i]
}

sapply(listOfSelected, length)

## ----include=FALSE-----

mapped_genes2GO <- mappedkeys(org.Hs.egGO)
mapped_genes2KEGG <- mappedkeys(org.Hs.egPATH)
mapped_genes <- union(mapped_genes2GO , mapped_genes2KEGG)

## ----echo=FALSE-----

library(clusterProfiler)

listOfData <- listOfSelected[1:3]

```

```

comparisonsNames <- names(listOfData)
universe <- mapped_genes

for (i in 1:length(listOfData)){
  genesIn <- listOfData[[i]]
  comparison <- comparisonsNames[i]
  enrich.result <- enrichKEGG(gene = genesIn,
                              pvalueCutoff = 0.10,
                              pAdjustMethod = "BH",
                              organism = "human",
                              universe = universe)

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

  if (length(rownames(enrich.result@result)) != 0) {
    write.csv(as.data.frame(enrich.result),
              file =paste0("./results/", "cluster.Results.", comparison, ".csv"),
              row.names = FALSE)

    pdf(file=paste0("./results/", "clusterBarplot.", comparison, ".pdf"))
    print(barplot(enrich.result, showCategory = 15, font.size = 4,
                  title = paste0("Cluster Pathway Analysis for ", comparison, ". Barplot")))
    dev.off()

    pdf(file = paste0("./results/", "clustercnetplot.", comparison, ".pdf"))
    print(cnetplot(enrich.result, categorySize = "geneNum", showCategory = 15,
                   vertex.label.cex = 0.75))
    dev.off()
  }
}

## ----echo=FALSE-----
tabla_NIT_vs_ELI <- read.csv2("./results/cluster.Results.NIT_vs_ELI.csv", header = TRUE, sep = ",")

knitr::kable(head(tabla_NIT_vs_ELI), booktabs = TRUE, caption = 'Tabla NIT_vs_ELI')

## ----echo=FALSE-----
tabla_NIT_vs_SFI <- read.csv2("./results/cluster.Results.NIT_vs_SFI.csv", header = TRUE, sep = ",")

knitr::kable(head(tabla_NIT_vs_SFI), booktabs = TRUE, caption = 'Tabla NIT_vs_SFI')

## ----echo=FALSE-----
tabla_ELI_vs_SFI <- read.csv2("./results/cluster.Results.ELI_vs_SFI.csv", header = TRUE, sep = ",")

knitr::kable(head(tabla_ELI_vs_SFI), booktabs = TRUE, caption = 'Tabla ELI_vs_SFI')

## ----echo=FALSE-----

```



```
knitr::include_graphics("./results/barplot_NIT_vs_ELI.png")

knitr::include_graphics("./results/cnetplot_NIT_vs_ELI.png")

## ----echo=FALSE-----
knitr::include_graphics("./results/barplot_NIT_vs_SFI.png")

knitr::include_graphics("./results/cnetplot_NIT_vs_SFI.png")

## ----echo=FALSE-----
knitr::include_graphics("./results/barplot_ELI_vs_SFI.png")

knitr::include_graphics("./results/cnetplot_ELI_vs_SFI.png")
```

Bibliografia

- (1) Yu, Guangchuang, and Qing-Yu He. 2016. "ReactomePA: An R/Bioconductor Package for Reactome Pathway Analysis and Visualization." *Molecular BioSystems* 12 (2): 477–79. <https://doi.org/10.1039/C5MB00663E>.
- (2) Yu, Guangchuang, Li-Gen Wang, Guang-Rong Yan, and Qing-Yu He. 2015. "DOSE: An R/Bioconductor Package for Disease Ontology Semantic and Enrichment Analysis." *Bioinformatics* 31 (4): 608–9. <https://doi.org/10.1093/bioinformatics/btu684>.
- (3) Yu, G., Wang, L. G., Han, Y., & He, Q. Y. (2012). clusterProfiler: an R package for comparing biological themes among gene clusters. *Omics : a journal of integrative biology*, 16(5), 284–287. <https://doi.org/10.1089/omi.2011.0118>