

# PEC2

Rubén Jové Nieto

8/6/2020

Enlace al repositorio de github.

[https://github.com/rbenjn/PEC\\_02](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. Búsqueda 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.

```
## Warning: Missing column names filled in: 'X1' [1]

## Parsed with column specification:
## cols(
##   .default = col_double(),
##   X1 = col_character()
## )

## See spec(...) for full column specifications.

## Parsed with column specification:
## cols(
##   Experiment = col_character(),
##   SRA_Sample = col_character(),
##   Sample_Name = col_character(),
##   Grupo_analisis = col_double(),
##   body_site = col_character(),
##   molecular_data_type = col_character(),
##   sex = col_character(),
##   Group = col_character(),
##   ShortName = col_character()
## )
```

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’.

```
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)
```

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

```
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]
```

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.

```
dds <- DESeqDataSetFromMatrix(countData = countdata, colData = targets_10_g, design = ~Group)
```

```
## converting counts to integer mode
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
## design formula are characters, converting to factors
```

**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.

```
nrow(dds)
```

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

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

```
## [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)

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

```
##           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
```

```
colData(vsd)
```

```
## 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().

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

```
## rlog() may take a few minutes with 30 or more samples,  
## vst() is a much faster transformation
```

```
head(assay(rld), 3)
```

```
##           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:

```
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"))
```

```
## Warning: `as_data_frame()` is deprecated as of tibble 2.0.0.  
## Please use `as_tibble()` instead.  
## The signature and semantics have changed, see `?as_tibble`.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last_warnings()` to see where this warning was generated.
```

```
colnames(df)[1:2] <- c("x", "y")
```

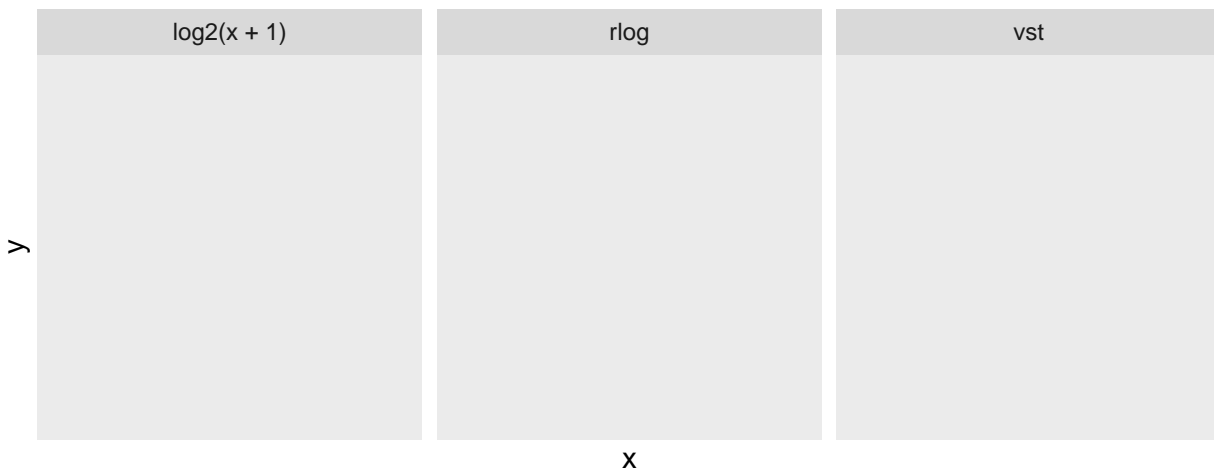
```
ggplot(df, aes(x = x, y = y)) + geom_hex(bins = 80) +  
  coord_fixed() + facet_grid( . ~ transformation)
```

```
## Warning: Computation failed in `stat_binhex()`:
```

```
## Package `hexbin` required for `stat_binhex`.
## Please install and try again.

## Warning: Computation failed in `stat_binhex()`:
## Package `hexbin` required for `stat_binhex`.
## Please install and try again.

## Warning: Computation failed in `stat_binhex()`:
## Package `hexbin` required for `stat_binhex`.
## Please install and try again.
```



Utilizando los resultados de VST, es decir el objeto ‘vst’ 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.

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

```
##           1.1           1.2           1.3           1.4           1.5           1.6           1.7
## 1.2  122.91043
## 1.3  103.06203 135.69252
## 1.4  103.73411 133.81523 111.55542
## 1.5  119.02867 136.63502 117.19267 125.76381
## 1.6  106.50627 125.44629 108.40843 101.77091 120.77517
```

```

## 1.7 135.95673 125.52461 139.04018 141.58821 132.17251 134.30872
## 1.8 97.82843 133.59801 109.10060 106.74988 122.45512 115.93692 138.66525
## 1.9 125.87081 139.38958 128.78547 133.73684 114.96069 133.88622 125.72504
## 1.10 125.15642 117.55104 125.77388 128.44760 147.19765 120.08176 138.94726
## 2.1 145.74360 148.88860 155.58994 137.14828 158.78926 143.17134 145.50889
## 2.2 138.58070 136.64917 138.99921 127.99990 142.65618 129.83290 139.41702
## 2.3 161.90843 153.59093 173.21075 164.09264 188.47190 167.14992 169.93220
## 2.4 124.45308 151.49586 127.36301 132.43215 141.75937 131.25095 147.65926
## 2.5 127.33593 132.28582 128.36571 123.01669 143.55130 121.42442 133.61318
## 2.6 141.85846 156.37124 148.39924 142.45793 164.40156 149.20903 162.71154
## 2.7 136.09527 126.09378 149.76006 147.70050 144.29568 144.99484 134.42696
## 2.8 125.78980 119.22113 133.93536 136.43342 142.52892 133.93012 146.33093
## 2.9 138.08641 168.01200 135.28439 141.27695 160.42524 144.97056 172.10743
## 2.10 109.80940 141.99726 108.16059 109.86312 125.78904 105.86894 140.19311
## 3.1 169.11868 188.71033 181.57983 169.99800 193.92096 177.33470 185.86952
## 3.2 160.89673 166.06306 169.77965 164.26416 186.52386 167.66285 165.52070
## 3.3 194.36547 196.03220 199.66171 197.31171 213.62862 199.00014 199.32490
## 3.4 159.01867 160.41867 168.63543 159.95971 184.46820 161.83281 174.22319
## 3.5 130.77726 114.39174 137.25315 137.22466 140.04700 141.26564 136.47202
## 3.6 180.27601 199.68984 186.59840 176.32965 197.62776 185.24033 195.22600
## 3.7 135.57267 129.24158 146.27463 137.64244 145.84366 135.04024 113.59466
## 3.8 156.47218 164.23669 164.56659 148.80228 175.27419 150.88515 158.98183
## 3.9 202.81538 204.81669 207.64087 199.79003 220.00174 203.48525 201.74725
## 3.10 155.46560 149.97144 163.97209 154.42711 170.27093 152.53889 141.30125
##      1.8      1.9      1.10      2.1      2.2      2.3      2.4
## 1.2
## 1.3
## 1.4
## 1.5
## 1.6
## 1.7
## 1.8
## 1.9 137.21723
## 1.10 131.25162 147.68764
## 2.1 142.37267 154.00769 146.24878
## 2.2 142.13646 136.66863 120.22514 127.69969
## 2.3 157.03574 192.81159 166.68533 142.06507 176.16697
## 2.4 112.20625 159.84074 149.59265 143.41166 152.01220 150.42653
## 2.5 122.95809 144.22094 123.64053 128.45391 126.06740 150.27107 129.88154
## 2.6 140.15148 170.37700 153.62237 129.32231 151.49236 137.67810 127.94202
## 2.7 138.45255 155.75510 150.07172 136.28180 157.91800 138.98610 140.40120
## 2.8 128.21536 159.65072 136.49860 142.93031 150.26261 136.35470 133.48869
## 2.9 130.39091 175.74012 149.54918 155.80970 156.56954 174.06122 136.99888
## 2.10 118.35574 133.33953 139.81535 142.24882 138.32786 159.93106 119.86244
## 3.1 158.43400 202.49138 194.47680 152.53157 193.21207 144.64931 147.79465
## 3.2 151.96250 188.17999 173.97808 144.21697 184.07977 121.31915 137.84034
## 3.3 185.96249 221.33543 202.26392 169.48220 208.17433 138.99559 169.66826
## 3.4 153.01465 190.55293 167.44659 138.36132 171.87819 119.58384 144.88157
## 3.5 135.75546 141.65288 135.62504 141.13534 142.43646 135.08207 143.08261
## 3.6 168.30981 202.11324 202.60287 151.58140 190.92560 154.13195 162.93435
## 3.7 137.62527 130.89009 138.62168 124.93369 124.48130 158.74505 143.51438
## 3.8 146.02126 176.27262 166.77170 119.48109 159.27133 130.51477 134.19621
## 3.9 190.12474 224.87930 211.23087 166.54714 211.89527 139.90687 172.89232
## 3.10 153.31682 167.60963 165.71903 132.92108 167.57729 123.78656 143.59001

```

##	2.5	2.6	2.7	2.8	2.9	2.10	3.1
## 1.2							
## 1.3							
## 1.4							
## 1.5							
## 1.6							
## 1.7							
## 1.8							
## 1.9							
## 1.10							
## 2.1							
## 2.2							
## 2.3							
## 2.4							
## 2.5							
## 2.6	145.73600						
## 2.7	139.61595	142.92688					
## 2.8	137.66597	135.78315	114.75674				
## 2.9	137.56714	140.09493	164.24052	153.17637			
## 2.10	128.21502	136.51478	145.50338	136.46634	146.54112		
## 3.1	166.67285	157.11913	166.35388	169.05663	179.51473	167.55894	
## 3.2	151.28714	143.12003	147.32592	151.56696	172.52120	157.81401	130.87644
## 3.3	186.07885	163.81184	173.83700	173.16568	194.87246	187.87478	143.41178
## 3.4	146.75578	144.99635	151.72498	139.09480	168.08291	157.59597	136.84620
## 3.5	138.48739	133.39322	126.22468	115.77175	161.21947	134.67095	173.43196
## 3.6	180.30817	160.23518	182.52070	180.19800	183.36399	171.84632	128.84350
## 3.7	118.46957	153.56200	138.85967	145.11298	169.12595	138.84824	172.72682
## 3.8	140.94539	134.69775	149.22390	148.86733	167.75653	144.46322	130.28678
## 3.9	182.37852	173.77593	181.98087	186.34558	200.94495	191.55256	132.07951
## 3.10	138.24090	144.87621	133.20658	144.05905	178.56041	144.40589	148.36993
##	3.2	3.3	3.4	3.5	3.6	3.7	3.8
## 1.2							
## 1.3							
## 1.4							
## 1.5							
## 1.6							
## 1.7							
## 1.8							
## 1.9							
## 1.10							
## 2.1							
## 2.2							
## 2.3							
## 2.4							
## 2.5							
## 2.6							
## 2.7							
## 2.8							
## 2.9							
## 2.10							
## 3.1							
## 3.2							
## 3.3	118.53059						
## 3.4	127.44952	143.32583					



```

## 3.5  147.10058 167.27568 147.52599
## 3.6  142.42321 137.58741 139.43764 176.08653
## 3.7  163.23625 198.81752 157.26786 140.33742 184.21882
## 3.8  124.22609 144.18662 122.58459 153.01344 129.45863 141.61654
## 3.9  115.70949 106.46253 139.12332 179.80637 127.81914 191.99301 135.78703
## 3.10 120.04325 151.89658 135.95268 134.24649 160.33332 134.68814 121.70859
##      3.9
## 1.2
## 1.3
## 1.4
## 1.5
## 1.6
## 1.7
## 1.8
## 1.9
## 1.10
## 2.1
## 2.2
## 2.3
## 2.4
## 2.5
## 2.6
## 2.7
## 2.8
## 2.9
## 2.10
## 3.1
## 3.2
## 3.3
## 3.4
## 3.5
## 3.6
## 3.7
## 3.8
## 3.9
## 3.10 145.21190

```

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.

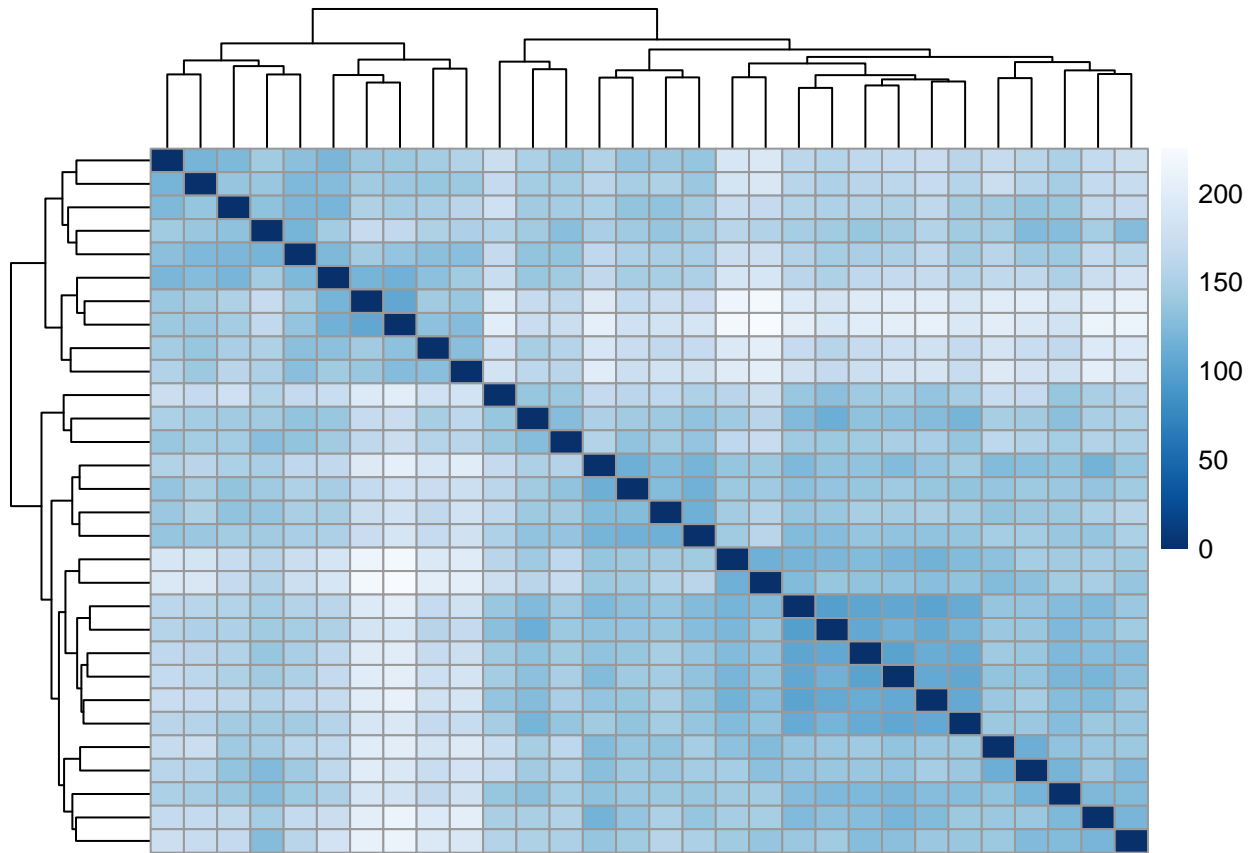
Lo visualizamos en el siguiente mapa de calor:

```

library("pheatmap")
library("RColorBrewer")

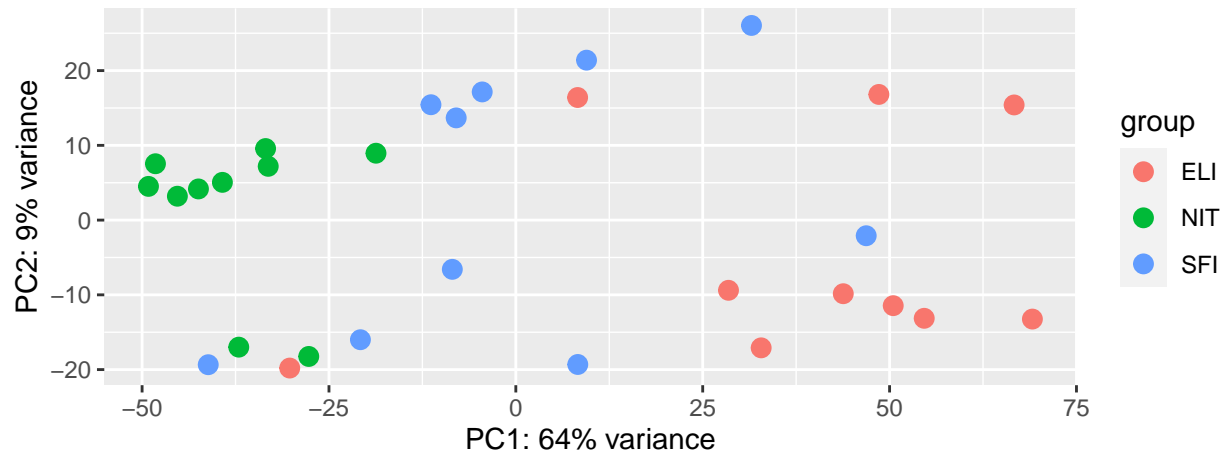
sampleDistMatrix <- as.matrix( sampleDists )
rownames(sampleDistMatrix) <- paste( vsd$dex, vsd$cell, sep = " - " )
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
          clustering_distance_rows = sampleDists,
          clustering_distance_cols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

```



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

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



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.

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

## using pre-existing size factors

## estimating dispersions

## gene-wise dispersion estimates: 6 workers

## mean-dispersion relationship

## final dispersion estimates, fitting model and testing: 6 workers

## -- replacing outliers and refitting for 240 genes
## -- DESeq argument 'minReplicatesForReplace' = 7
## -- original counts are preserved in counts(dds)
```

```
## estimating dispersions
```

```
## fitting model and testing
```

```
# Realizamos las distintas comparaciones usando 'contrast' para cada grupo, con su respectivo factor:
```

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

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

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

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

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

```
## [1] 8318
```

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

```
## [1] 1608
```

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

```
## [1] 3670
```

NIT vs ELI: 8318 genes significativos

NIT vs SFI: 1608 genes significativos

ELI vs SFI: 3670 genes significativos

Posteriormente podemos identificar aquellos que están down-regulated:

```
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)
```

```
head(res_NIT_vs_ELI_Sig[ order(res_NIT_vs_ELI_Sig$log2FoldChange), ])
```

```
## 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				

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

```
## 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
```

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

```
## 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:

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

```
## 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
```

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

```
## 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
```

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

```
## 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 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")
```

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

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

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

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

```
head(res_NIT_vs_ELI_Ordered)
```

```
## 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
## ENSG0000009790.10 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
```

```
# 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")
```

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

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

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

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

```
## 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
```

```
# ELI vs SFI

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

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

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

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

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

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

head(res_ELI_vs_SFI_Ordered)
```

```
## 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:

El Diagrama de Venn nos muestra los genes comunes entre las 2 comparaciones. Un total de 822

Los Mapas de Calor como el siguiente permiten visualizar aquellos genes que se expresan de forma diferencial. Los colores que encontramos en la gráfica permite resaltar valores distintos según su expresión. El agrupamiento jerárquico permite encontrar genes con patrones comunes de variación y así asociarlo a los 3 grupos que estamos analizando (N.N, I.S y I.L)

```
# Con el paquete 'ClusterProfiler'

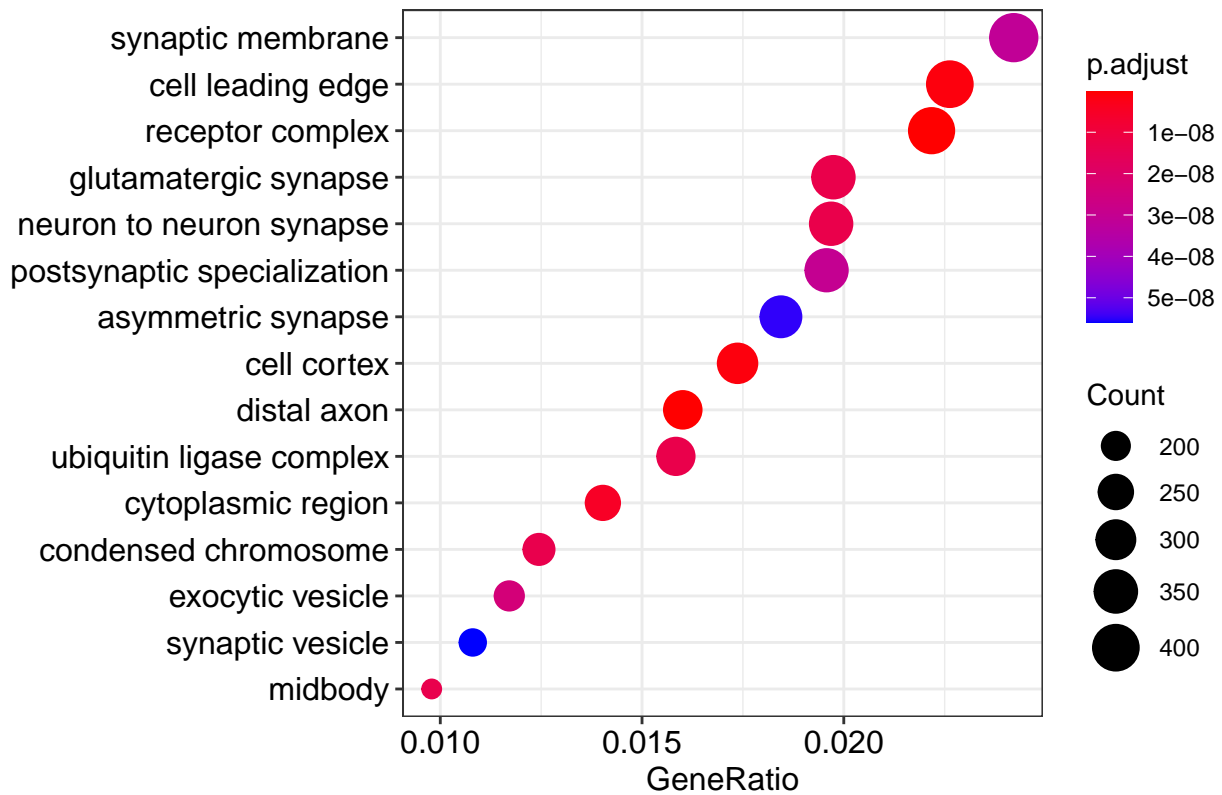
ego_NIT_vs_ELI <- enrichGO(gene          = tmp_NIT_vs_ELI,
                           OrgDb        = org.Hs.eg.db,
                           keyType       = 'ENSEMBL',
                           ont           = "CC",
                           pAdjustMethod = "BH",
                           pvalueCutoff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

## 9. Análisis de significación biológica (“Gene Enrichment Analysis”)

```
## 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
## G0:0150034 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 ENSG00000067606/ENSG00000171680/ENSG00000162493/ENSG00000142627/ENSG00000088280/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
```

```
dotplot(ego_NIT_vs_ELI, showCategory = 15)
```



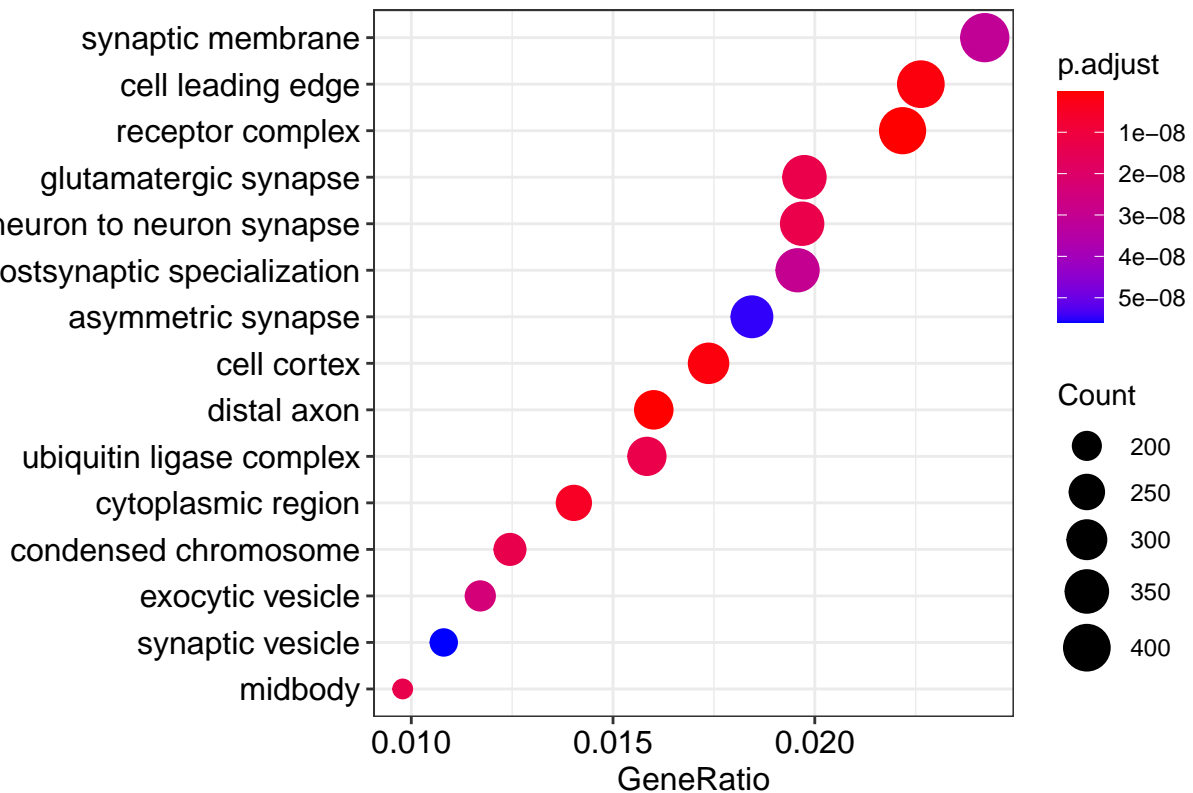
```
ego_NIT_vs_SFI <- enrichGO(gene          = tmp_NIT_vs_SFI,
  OrgDb          = org.Hs.eg.db,
  keyType        = 'ENSEMBL',
  ont            = "CC",
  pAdjustMethod  = "BH",
  pvalueCutoff   = 0.01,
  qvalueCutoff   = 0.05)
head(summary(ego_NIT_vs_SFI))
```

```
## 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
## GO:0150034 GO:0150034      distal axon 283/17677 296/21659 4.632806e-13
## GO:0043235 GO:0043235      receptor complex 392/17677 419/21659 8.502792e-13
## GO:0031252 GO:0031252      cell leading edge 400/17677 430/21659 6.022935e-12
## GO:0005938 GO:0005938      cell cortex 307/17677 325/21659 7.625450e-12
## GO:0099568 GO:0099568      cytoplasmic region 248/17677 260/21659 3.127931e-11
## GO:0098978 GO:0098978      glutamatergic synapse 349/17677 375/21659 1.152573e-10
##           p.adjust      qvalue
## GO:0150034 3.413871e-10 2.295754e-10
## GO:0043235 3.413871e-10 2.295754e-10
## GO:0031252 1.530809e-09 1.029436e-09
## GO:0005938 1.530809e-09 1.029436e-09
## GO:0099568 5.023457e-09 3.378165e-09
```

```
## G0:0098978 1.324241e-08 8.905235e-09
##
## G0:0150034
## G0:0043235
## G0:0031252 ENSG00000067606/ENSG00000171680/ENSG00000162493/ENSG00000142627/ENSG00000088280/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
```

```
dotplot(ego_NIT_vs_SFI, showCategory = 15)
```



Preparamos lista de genes analizados

```
## NIT_vs_ELI NIT_vs_SFI ELI_vs_SFI
## 43699 43699 43699
```

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
hsa04010	hsa04010 MAPK signaling pathway	292/7483	294/7984	8.443451e-07
hsa04144	hsa04144 Endocytosis	246/7483	247/7984	1.558988e-06
hsa04510	hsa04510 Focal adhesion	200/7483	200/7984	1.982885e-06
hsa05010	hsa05010 Alzheimer disease	362/7483	367/7984	2.948692e-06
hsa04024	hsa04024 cAMP signaling pathway	215/7483	216/7984	1.085583e-05
hsa04015	hsa04015 Rap1 signaling pathway	209/7483	210/7984	1.575858e-05

	p.adjust	qvalue
hsa04010	0.0002134906	0.0001294093
hsa04144	0.0002134906	0.0001294093
hsa04510	0.0002134906	0.0001294093
hsa05010	0.0002381069	0.0001443307
hsa04024	0.0007012866	0.0004250914
hsa04015	0.0008483367	0.0005142272

```
##
## hsa04010
## hsa04144
## hsa04510
## hsa05010 5330/80326/5293/489/7124/841/7471/51384/778/5682/10383/5702/8851/5707/4702/84790/3028/3551/
## hsa04024
## hsa04015
```

	Count
hsa04010	292
hsa04144	246
hsa04510	200
hsa05010	362
hsa04024	215
hsa04015	209

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

```
## Comparison: NIT_vs_SFI
```

ID	Description	GeneRatio	BgRatio	pvalue
hsa04010	hsa04010 MAPK signaling pathway	292/7483	294/7984	8.443451e-07
hsa04144	hsa04144 Endocytosis	246/7483	247/7984	1.558988e-06
hsa04510	hsa04510 Focal adhesion	200/7483	200/7984	1.982885e-06
hsa05010	hsa05010 Alzheimer disease	362/7483	367/7984	2.948692e-06
hsa04024	hsa04024 cAMP signaling pathway	215/7483	216/7984	1.085583e-05
hsa04015	hsa04015 Rap1 signaling pathway	209/7483	210/7984	1.575858e-05

	p.adjust	qvalue
hsa04010	0.0002134906	0.0001294093
hsa04144	0.0002134906	0.0001294093
hsa04510	0.0002134906	0.0001294093
hsa05010	0.0002381069	0.0001443307
hsa04024	0.0007012866	0.0004250914
hsa04015	0.0008483367	0.0005142272

```
##
## hsa04010
## hsa04144
```

```

## hsa04510
## hsa05010 4311/1454/80326/4702/10382/79861/7471/4540/30849/1020/4709/4512/6391/3710/355/2081/54539/45
## hsa04024
## hsa04015
##          Count
## hsa04010    292
## hsa04144    246
## hsa04510    200
## hsa05010    362
## hsa04024    215
## hsa04015    209

## #####
## Comparison:  ELI_vs_SFI
##          ID          Description GeneRatio  BgRatio      pvalue
## hsa04010 hsa04010 MAPK signaling pathway 292/7483 294/7984 8.443451e-07
## hsa04144 hsa04144      Endocytosis 246/7483 247/7984 1.558988e-06
## hsa04510 hsa04510      Focal adhesion 200/7483 200/7984 1.982885e-06
## hsa05010 hsa05010      Alzheimer disease 362/7483 367/7984 2.948692e-06
## hsa04024 hsa04024 cAMP signaling pathway 215/7483 216/7984 1.085583e-05
## hsa04015 hsa04015 Rap1 signaling pathway 209/7483 210/7984 1.575858e-05
##          p.adjust      qvalue
## hsa04010 0.0002134906 0.0001294093
## hsa04144 0.0002134906 0.0001294093
## hsa04510 0.0002134906 0.0001294093
## hsa05010 0.0002381069 0.0001443307
## hsa04024 0.0007012866 0.0004250914
## hsa04015 0.0008483367 0.0005142272
##
## hsa04010
## hsa04144
## hsa04510
## hsa05010 5293/5330/489/8851/1536/84617/27123/841/7475/7124/5702/1452/10023/1131/22926/79861/10376/10
## hsa04024
## hsa04015
##          Count
## hsa04010    292
## hsa04144    246
## hsa04510    200
## hsa05010    362
## hsa04024    215
## hsa04015    209

```

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

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

## Resultados

Obtenemos resultados significativos de ambas comparaciones.

## Muestra Normal vs Muestra Pulpitis

El Gene Set Enrichment Analysis mostró genes que se expresan de manera diferencial. Con una mayor expresión en la muestra de Pulpitis de genes asociados con la activación de la respuesta inmunitaria.

**Muestra Dolor Leve vs Dolor Severo** Entre las personas que referían dolor (3 severo y 3 leve), encontramos genes expresados de forma diferencial entre los 2 grupos. Con especial atención a aquellos genes relacionados con el sistema inmune adaptativo y la interacción citoquina-citoquina.

## Discusión

Aunque el uso del análisis de microarrays es una herramienta poderosa para estudiar la expresión simultánea de varios genes, hay algunas limitaciones.

Por ejemplo, factores técnicos como el rango limitado y la hibridización cruzada. Además el perfil celular de las pulpas inflamadas difiere de las normales. En las inflamadas se caracterizan por tener una afluencia ya por encima de lo normal de células inmunes.

Es decir, que los resultados pueden ser en parte debido a diferencias en la composición celular.

## Apéndice

El código R usado en este documento R Markdown:

## Bibliografía

- (1) Galicia, J. C., Henson, B. R., Parker, J. S., & Khan, A. A. (2016). Gene expression profile of pulpitis. *Genes and immunity*, 17(4), 239–243. <https://doi.org/10.1038/gene.2016.14>
- (2) 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>.
- (3) 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>.