

Pipeline corresponent a nualart_oriol_ADO_PEC2.pdf

Oriol Nualart Mundó

14/06/2020

CONTINGUTS

- 3.3.1. Preparació de les dades
- 3.3.2. Control de qualitat, filtratge i visualització preliminar de les dades
- 3.3.3. Normalització
- 3.3.4. Anàlisi d'expressió diferencial
- 3.3.5. Anotació dels resultats
- 3.3.6. Comparació entre comparacions
- 3.3.7. Anàlisi de significació biològica
- 4. Resultats

3.3.1. Preparació de les dades

- Càrrega dels arxius *targets* i *counts* com a *data frames* i correcció del nom de les mostres de *targets_big*

```
targets_big <- read.csv2("./data/targets.csv", header = TRUE, sep = ",")  
  
counts_big <- read.csv2("./data/counts.csv", header = TRUE, sep = ";")  
  
library(stringr)  
  
sample_names <- targets_big$Sample_Name  
sample_names <- str_replace_all(sample_names, "-", ".")  
targets_big$Sample_Name <- sample_names
```

- Llavor pseudoaleatòria

```
set.seed(123)
```

- Selecció aleatòria dels números de fila de *targets_big*

```
library(dplyr)  
  
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##     filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##     intersect, setdiff, setequal, union  
  
samplNIT <- sample(which(targets_big$Group == "NIT"), 10)  
samplSFI <- sample(which(targets_big$Group == "SFI"), 10)  
samplELI <- sample(which(targets_big$Group == "ELI"), 10)  
  
samples <- c(samplNIT, samplSFI, samplELI)
```

- Creació del data frame *targets* reduït amb les mostres seleccionades

```
targets <- targets_big[samples,]
```

- Creació del data frame *counts* reduït amb les mostres seleccionades

```
columns <- targets$Sample_Name  
counts <- counts_big[columns]
```

- Correcció del nom de les files i columnes de *counts*

```
X <- c()  
  
for (i in 1:length(counts_big$X)) {  
  X[i] <- substring(counts_big$X[i], 1, 15)  
}  
  
row.names(counts) <- X  
colnames(counts) <- targets$ShortName
```

- Construcció del *DESeqDataSet*

```

library(DESeq2)

## Loading required package: S4Vectors
## Loading required package: stats4
## Loading required package: BiocGenerics
## Loading required package: parallel
##
## Attaching package: 'BiocGenerics'

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

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

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

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which, which.max, which.min

##
## Attaching package: 'S4Vectors'

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

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

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

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

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

## Loading required package: GenomicRanges

```

```

## Loading required package: GenomeInfoDb
## Loading required package: SummarizedExperiment
## Loading required package: Biobase
## Welcome to Bioconductor
##
##      Vignettes contain introductory material; view with
##      'browseVignettes()'. To cite Bioconductor, see
##      'citation("Biobase")', and for packages 'citation("pkgname")'.
## Loading required package: DelayedArray
## Loading required package: matrixStats
##
## Attaching package: 'matrixStats'
## The following objects are masked from 'package:Biobase':
## 
##     anyMissing, rowMedians
## The following object is masked from 'package:dplyr':
## 
##     count
## Loading required package: BiocParallel
##
## Attaching package: 'DelayedArray'
## The following objects are masked from 'package:matrixStats':
## 
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
## The following objects are masked from 'package:base':
## 
##     aperm, apply, rowsum
dds <- DESeqDataSetFromMatrix(countData = counts,
                               colData = targets,
                               design = ~ sex + Group)
dds$Group <- factor(dds$Group, levels = c("NIT", "SFI", "ELI"))
dds

## class: DESeqDataSet
## dim: 56202 30
## metadata(1): version
## assays(1): counts
## rownames(56202): ENSG00000223972 ENSG00000227232 ... ENSG00000210195
##     ENSG00000210196
## rowData names(0):
## colnames(30): S7SE-_NIT ZC5H-_NIT ... 13QJC_ELI YJ89-_ELI
## colData names(9): Experiment SRA_Sample ... Group ShortName

```

3.3.2. Control de qualitat, filtratge i visualització preliminar de les dades

- Transformació de les dades en *pseudocounts*

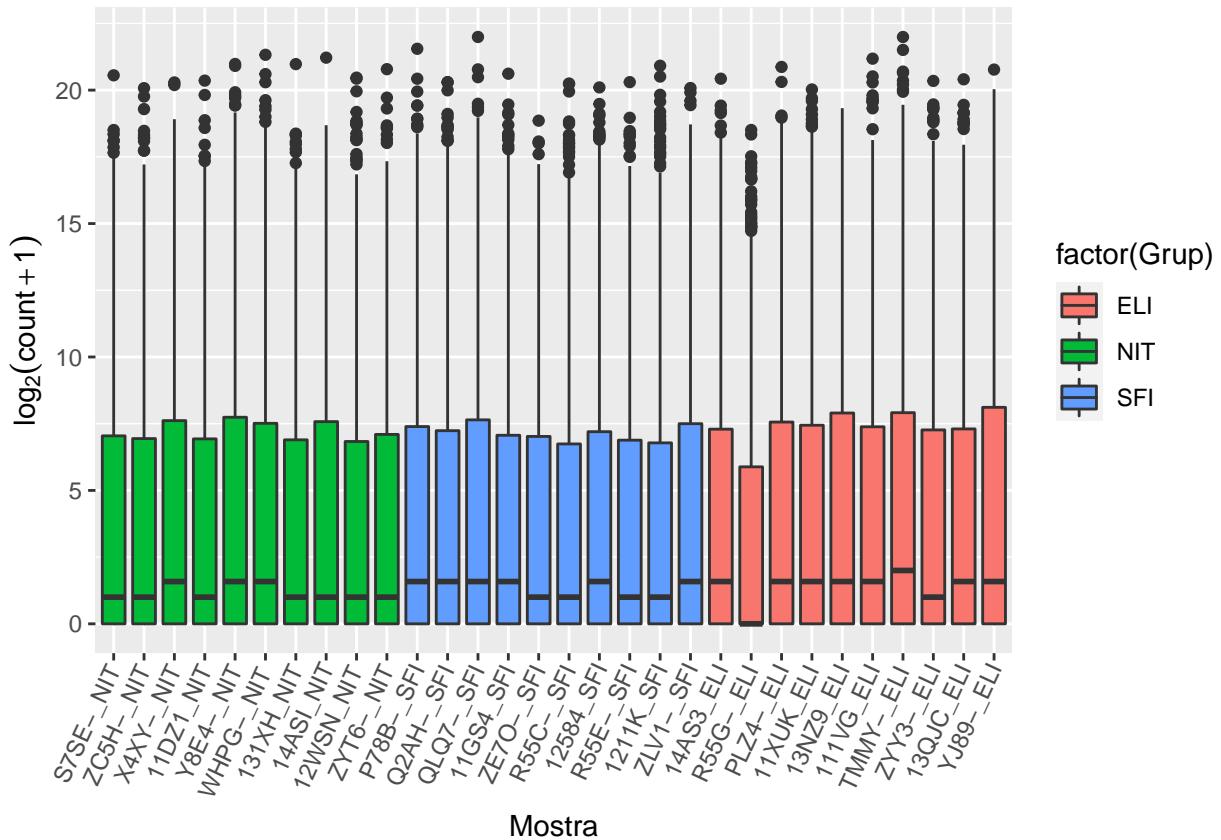
```
pseudoCounts <- log2(counts + 1)
```

- Boxplot de comparació entre mostres

```
library(reshape2)
library(ggplot2)
df <- melt(pseudoCounts, variable.name = "Samples")
```

```
## No id variables; using all as measure variables
```

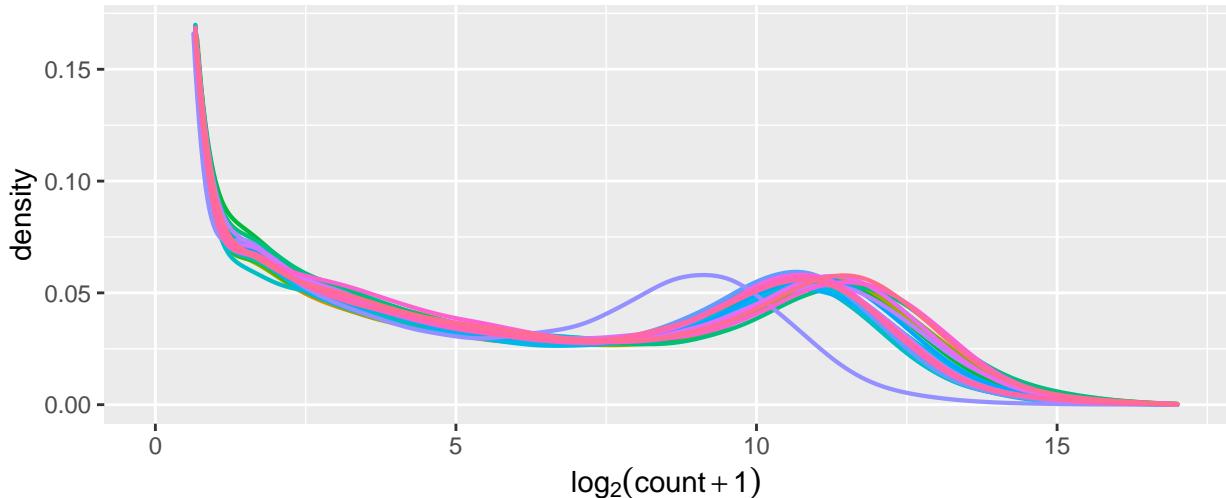
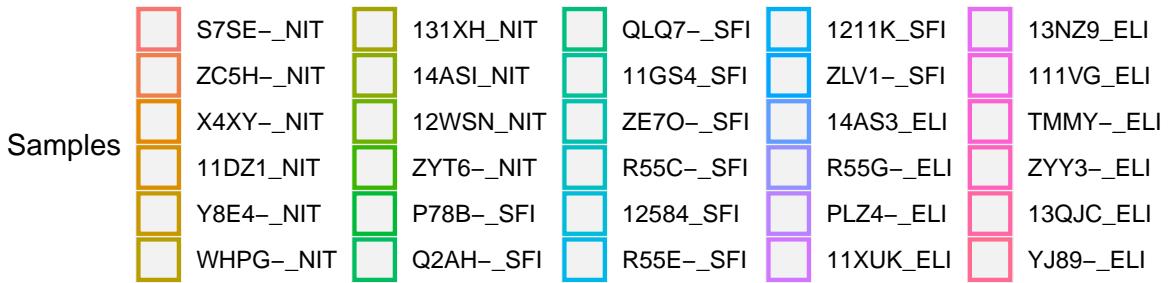
```
df$Grup <- c(rep("NIT", 562020), rep("SFI", 562020), rep("ELI", 562020))
ggplot(df, aes(x = Samples, y = value)) + geom_boxplot(aes(fill = factor(Grup))) + xlab("Mostra") +
ylab(expression(log[2](count + 1))) + theme(axis.text.x = element_text(angle = 65, hjust = 1.2, vjust = 0.5))
```



- Histograma de comparació entre mostres

```
ggplot(df, aes(x = value, colour = Samples)) + ylim(c(0, 0.17)) + xlim(c(0, 17)) +
geom_density(alpha = 0.2, size = 0.75) +
theme(legend.position = "top") + xlab(expression(log[2](count + 1)))
```

```
## Warning: Removed 556 rows containing non-finite values (stat_density).
```



- Filtratge de les dades

```
nrow(dds)
```

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

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

```
## [1] 43364
```

- Estabilització de la variança

```
rld <- rlog(dds)
```

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

- Heatmap de distància entre mostres

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

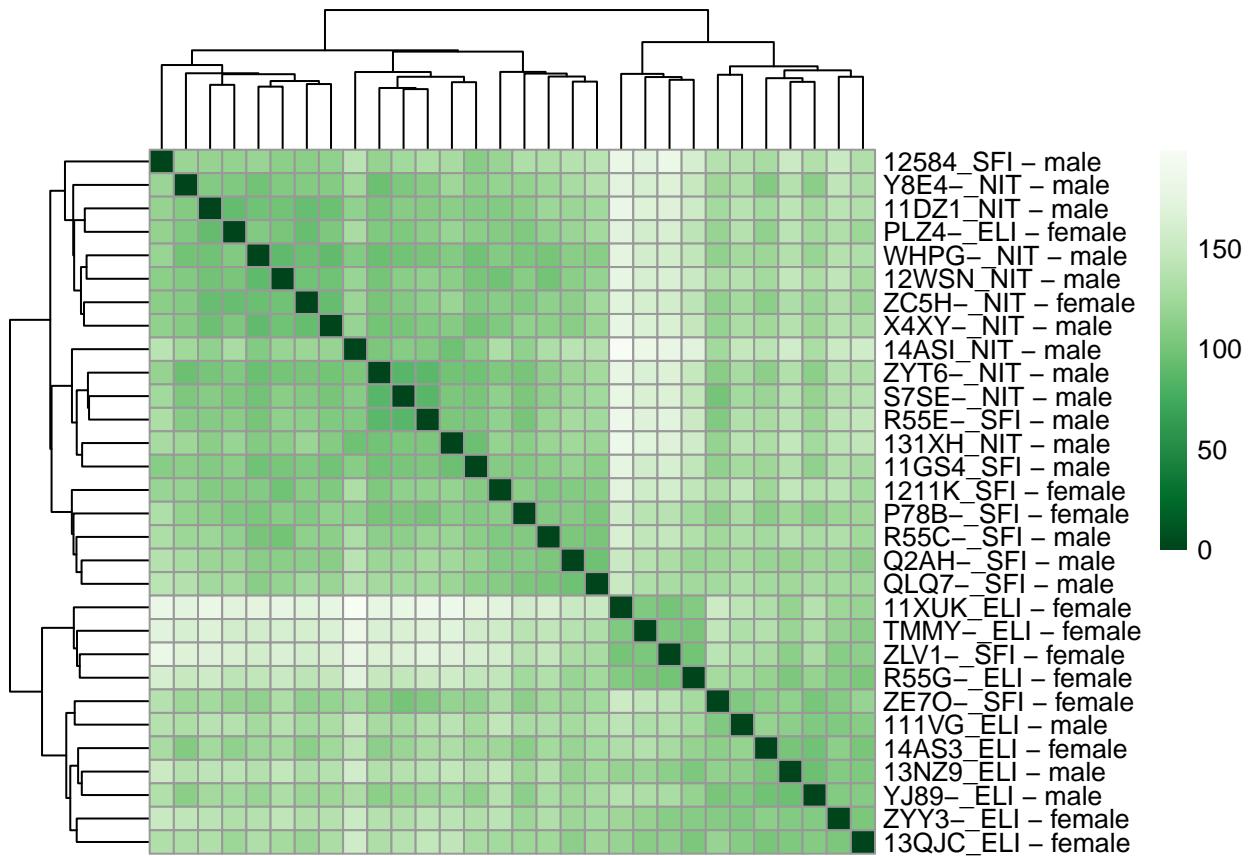
```
library("pheatmap")  
library("RColorBrewer")
```

```
sampleDistMatrix <- as.matrix(sampleDists)  
rownames(sampleDistMatrix) <- paste( rld$ShortName, rld$sex, sep = " - " )  
colnames(sampleDistMatrix) <- NULL  
colors <- colorRampPalette( rev(brewer.pal(9, "Greens")) )(255)  
pheatmap(sampleDistMatrix,
```

```

clustering_distance_rows = sampleDists,
clustering_distance_cols = sampleDists,
col = colors)

```

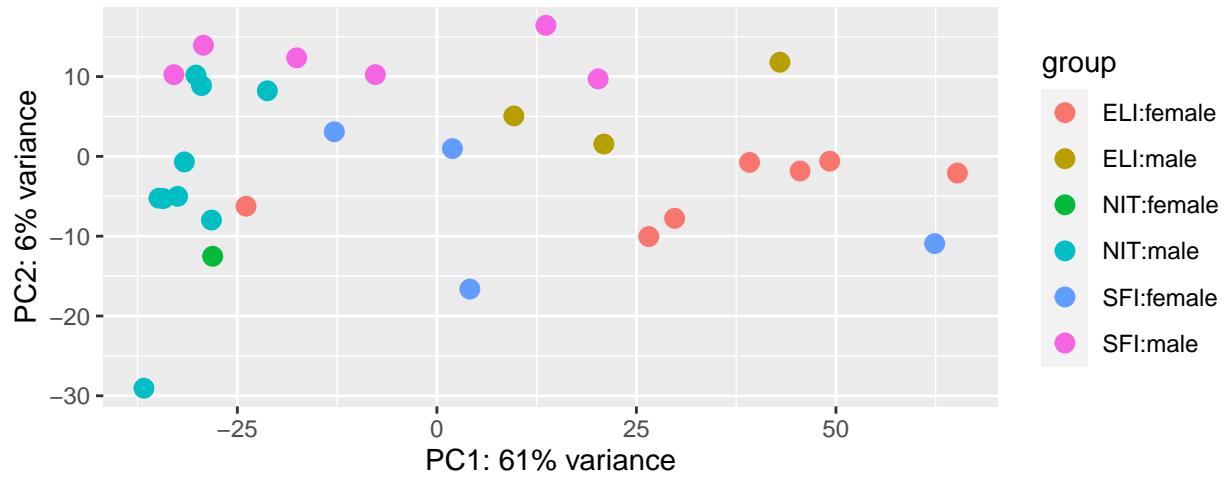


- Anàlisi de components principals

```

plotPCA(rld, intgroup = c("Group", "sex"))

```

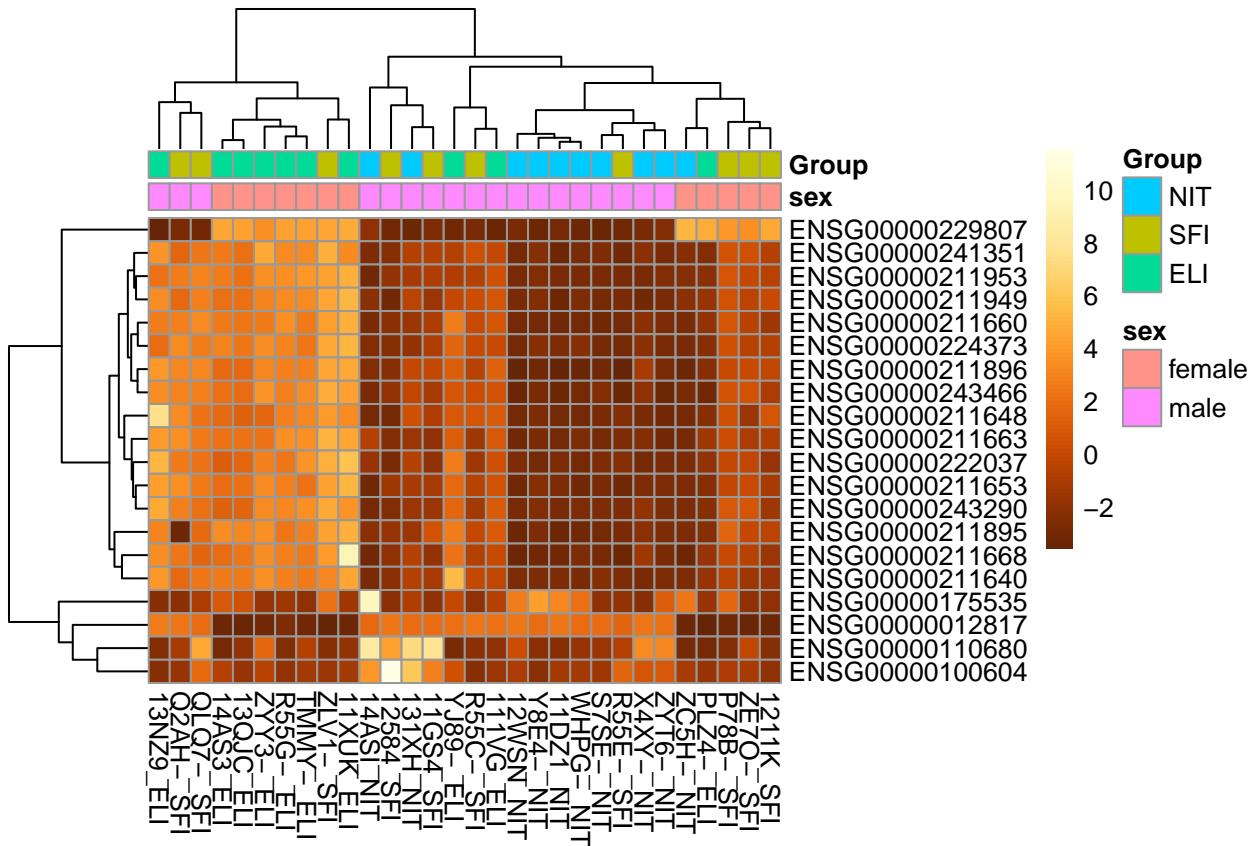


- Heatmap de la variança dels 20 gens amb expressió més variable

```
library("genefilter")

##
## Attaching package: 'genefilter'
## The following objects are masked from 'package:matrixStats':
##       rowSds, rowVars
topVarGenes <- head(order(rowVars(assay(rld))), decreasing = TRUE), 20)

mat  <- assay(rld)[topVarGenes, ]
mat  <- mat - rowMeans(mat)
anno <- as.data.frame(colData(rld)[, c("sex","Group")])
pheatmap(mat, annotation_col = anno, color = colorRampPalette(rev(brewer.pal(n = 9, name =
"YlOrBr")))(100))
```



3.3.3. Normalització

- Normalització per *Relative Log Expression*

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

3.3.4. Anàlisi d'expressió diferencial

- Estimació dels paràmetres de dispersió de les dades

```
dds <- estimateDispersions(dds)
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

- Aplicació dels tests negatius binomials

```
dds <- nbinoWaldTest(dds)
```

- Resum dels resultats del contrast ELI vs NIT

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

```
summary(resELIvsNIT)
```

```

## 
## out of 43364 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 881, 2%
## LFC < 0 (down)    : 2154, 5%
## outliers [1]       : 191, 0.44%
## low counts [2]     : 15096, 35%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

• Valors associats als gens amb un diferencial d'expressió positiu i negatiu més gran en el contrast ELI vs NIT

resSigELIvsNIT <- subset(resELIvsNIT, padj < 0.1)
valSigELIvsNIT <- resSigELIvsNIT[c("log2FoldChange", "pvalue", "padj")]
head(valSigELIvsNIT[ order(valSigELIvsNIT$log2FoldChange, decreasing = TRUE), ], 3)

## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 3 rows and 3 columns
##           log2FoldChange          pvalue          padj
##           <numeric>          <numeric>          <numeric>
## ENSG00000108849  4.9071129578193  0.00128138864454107  0.0244578850936638
## ENSG00000170419  4.74493205228696  5.57176867461188e-06  0.000402155653154441
## ENSG00000079689  4.67587739452871  2.44597214624369e-05  0.0012836553261698

head(valSigELIvsNIT[ order(valSigELIvsNIT$log2FoldChange), ], 3)

## log2 fold change (MLE): Group NIT vs ELI
## Wald test p-value: Group NIT vs ELI
## DataFrame with 3 rows and 3 columns
##           log2FoldChange          pvalue          padj
##           <numeric>          <numeric>          <numeric>
## ENSG00000211654 -22.36784577138  1.90368126580899e-18  1.06899317800238e-14
## ENSG00000170054 -21.8165051466117  5.01323261897087e-34  1.40756532242845e-29
## ENSG00000253742 -21.6755614741358  1.09220599710142e-33  1.53329338903082e-29

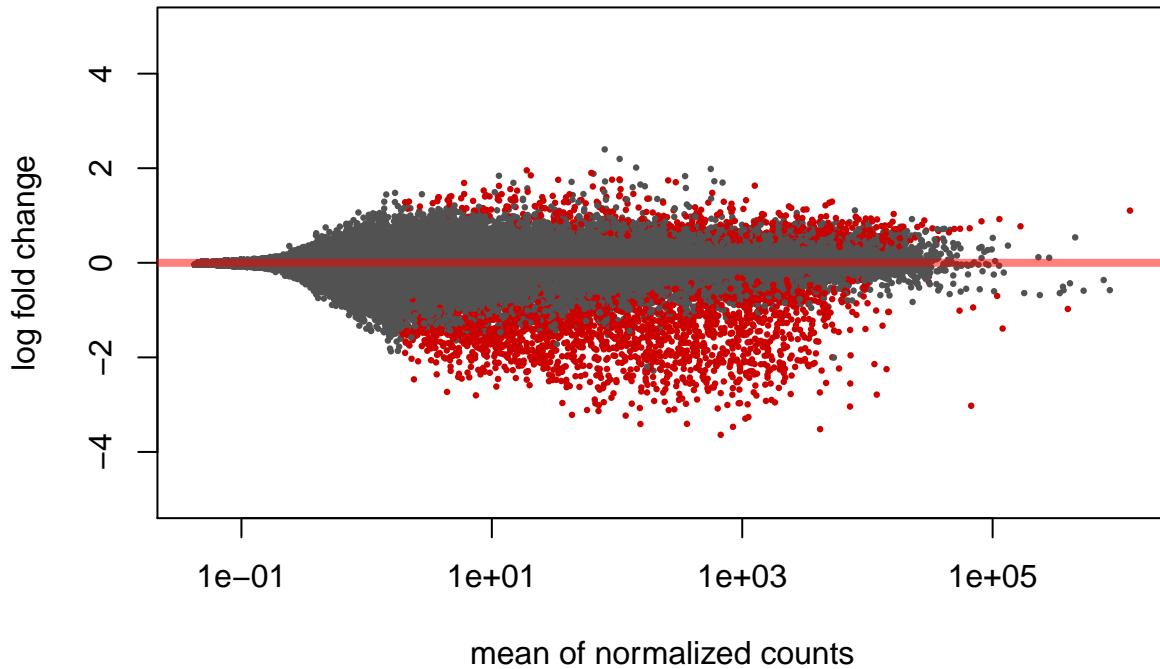
• MA plot del contrast ELI vs NIT

shrELIvsNIT <- lfcShrink(dds, contrast=c("Group", "NIT", "ELI"))

## using 'normal' for LFC shrinkage, the Normal prior from Love et al (2014).
##
## Note that type='apeglm' and type='ashr' have shown to have less bias than type='normal'.
## See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
## Reference: https://doi.org/10.1093/bioinformatics/bty895

plotMA(shrELIvsNIT, ylim = c(-5, 5))

```



- Resum dels resultats del contrast SFI vs NIT

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

```
summary(resSFIvsNIT)
```

```
##  
## out of 43364 with nonzero total read count  
## adjusted p-value < 0.1  
## LFC > 0 (up)      : 55, 0.13%  
## LFC < 0 (down)    : 277, 0.64%  
## outliers [1]       : 191, 0.44%  
## low counts [2]     : 13432, 31%  
## (mean count < 1)  
## [1] see 'cooksCutoff' argument of ?results  
## [2] see 'independentFiltering' argument of ?results
```

- Valors associats als gens amb un diferencial d'expressió positiu i negatiu més gran en el contrast SFI vs NIT

```
resSigSFIvsNIT <- subset(resSFIvsNIT, padj < 0.1)
```

```
valSigSFIvsNIT <- resSigSFIvsNIT[c("log2FoldChange", "pvalue", "padj")]
```

```
head(valSigSFIvsNIT[ order(valSigSFIvsNIT$log2FoldChange, decreasing = TRUE), ], 3)
```

	log2FoldChange	pvalue	padj
## log2 fold change (MLE): Group NIT vs SFI			
## Wald test p-value: Group NIT vs SFI			
## DataFrame with 3 rows and 3 columns			

```

## <numeric> <numeric> <numeric>
## ENSG00000216522 3.48987029993807 0.000422348293258026 0.050854496314927
## ENSG00000248560 3.48229581665463 0.000258269176448894 0.0350738976108062
## ENSG00000213642 3.26712295520834 9.38743594434013e-05 0.0158821115886208
head(valSigSFIvsNIT[ order(valSigSFIvsNIT$log2FoldChange), ], 3)

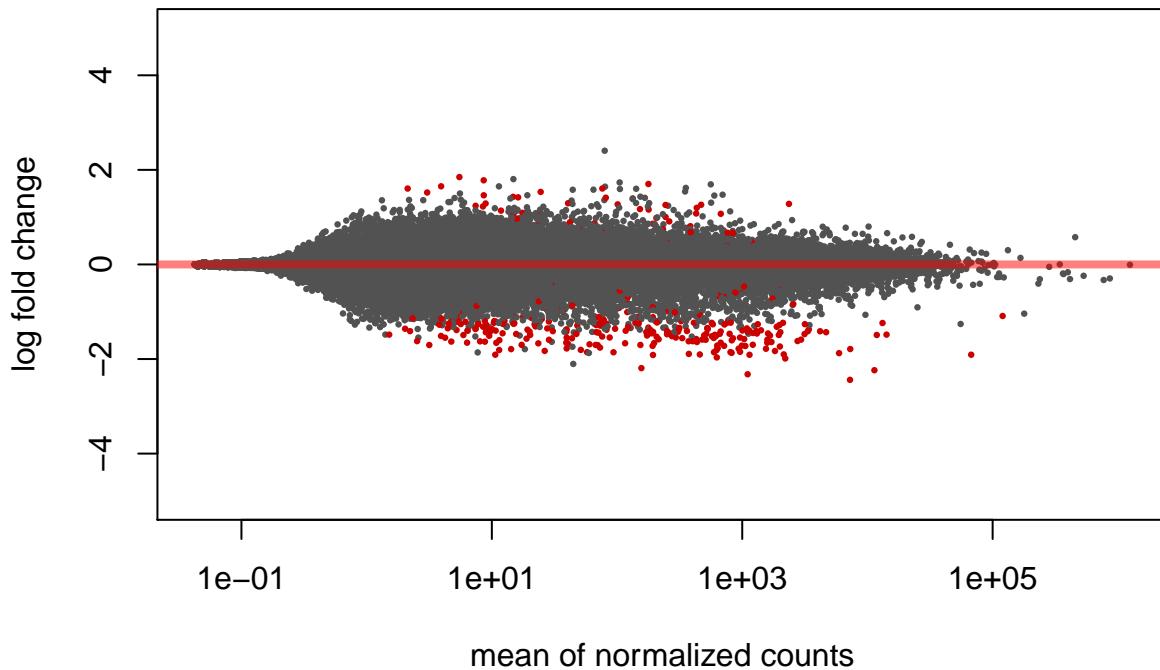
## log2 fold change (MLE): Group NIT vs SFI
## Wald test p-value: Group NIT vs SFI
## DataFrame with 3 rows and 3 columns
##          log2FoldChange           pvalue          padj
## <numeric> <numeric> <numeric>
## ENSG00000211654 -21.9099968985243 2.76331922069389e-21 1.64367753885314e-17
## ENSG00000253131 -20.4081759057272 2.0070185733501e-28 2.98453696950027e-24
## ENSG00000253742 -20.3181310036864 2.27831159641197e-35 6.77592651888885e-31

• MA plot del contrast SFI vs NIT

shrSFIvsNIT <- lfcShrink(dds, contrast=c("Group", "NIT", "SFI"))

## using 'normal' for LFC shrinkage, the Normal prior from Love et al (2014).
##
## Note that type='apeglm' and type='ashr' have shown to have less bias than type='normal'.
## See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
## Reference: https://doi.org/10.1093/bioinformatics/bty895
plotMA(shrSFIvsNIT, ylim = c(-5, 5))

```



- Resum dels resultats del contrast ELI vs SFI

```

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

summary(resELIvsSFI)

##
## out of 43364 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 1320, 3%
## LFC < 0 (down)    : 1512, 3.5%
## outliers [1]       : 191, 0.44%
## low counts [2]     : 17598, 41%
## (mean count < 4)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

• Valors associats als gens amb un diferencial d'expressió positiu i negatiu més gran en el contrast ELI vs SFI

resSigELIvsSFI <- subset(resELIvsSFI, padj < 0.1)
valSigELIvsSFI <- resSigELIvsSFI[c("log2FoldChange", "pvalue", "padj")]
head(valSigELIvsSFI[ order(valSigELIvsSFI$log2FoldChange, decreasing = TRUE), ], 3)

## log2 fold change (MLE): Group SFI vs ELI
## Wald test p-value: Group SFI vs ELI
## DataFrame with 3 rows and 3 columns
##           log2FoldChange          pvalue          padj
##           <numeric>          <numeric>          <numeric>
## ENSG00000181092 4.09926430328123 3.28763222370361e-05 0.00587980378470069
## ENSG00000089225 3.98119016224604 6.54329566980265e-05 0.00804542244015398
## ENSG00000164326 3.72378454936897 4.99181610057772e-05 0.00734885079009114
head(valSigELIvsSFI[ order(valSigELIvsSFI$log2FoldChange), ], 3)

## log2 fold change (MLE): Group SFI vs ELI
## Wald test p-value: Group SFI vs ELI
## DataFrame with 3 rows and 3 columns
##           log2FoldChange          pvalue          padj
##           <numeric>          <numeric>          <numeric>
## ENSG00000171195 -9.36895738684711 1.95805691553369e-05 0.00447118800131912
## ENSG00000254029 -6.23480756553029 0.000938714138255819 0.0277865903771905
## ENSG00000259721 -4.40925845029186 1.50308895511771e-05 0.00392260204358524

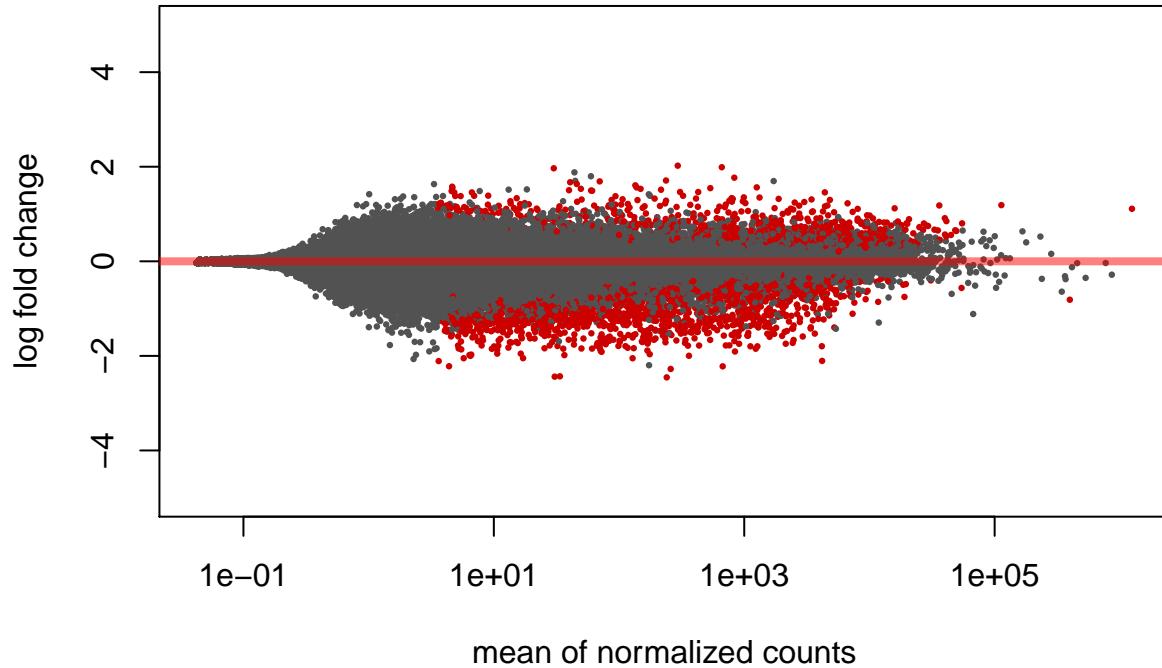
• MA plot del contrast ELI vs SFI

shrELIvsSFI <- lfcShrink(dds, contrast=c("Group", "SFI", "ELI"))

## using 'normal' for LFC shrinkage, the Normal prior from Love et al (2014).
##
## Note that type='apeglm' and type='ashr' have shown to have less bias than type='normal'.
## See ?lfcShrink for more details on shrinkage type, and the DESeq2 vignette.
## Reference: https://doi.org/10.1093/bioinformatics/bty895

```

```
plotMA(shrELIvsSFI, ylim = c(-5, 5))
```



3.3.5. Anotació dels resultats

- Addició del *gene symbol*, de l'*Entrez ID* i del *gene name* als llistats de gens expressats diferencialment

```
library("org.Hs.eg.db")
```

```
## Loading required package: AnnotationDbi
##
## Attaching package: 'AnnotationDbi'
## The following object is masked from 'package:dplyr':
##   select
##
library("AnnotationDbi")

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

## 'select()' returned 1:many mapping between keys and columns
resSigELIvsNIT$entrez  <- mapIds(org.Hs.eg.db, keys=row.names(resSigELIvsNIT),
                                    column="ENTREZID", keytype="ENSEMBL",
```

```

                    multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigELIvsNIT$genename <- mapIds(org.Hs.eg.db, keys=row.names(resSigELIvsNIT),
                                    column="GENENAME", keytype="ENSEMBL",
                                    multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigSFIvsNIT$symbol <- mapIds(org.Hs.eg.db, keys=row.names(resSigSFIvsNIT),
                                    column="SYMBOL", keytype="ENSEMBL", multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigSFIvsNIT$entrez <- mapIds(org.Hs.eg.db, keys=row.names(resSigSFIvsNIT),
                                    column="ENTREZID", keytype="ENSEMBL",
                                    multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigSFIvsNIT$genename <- mapIds(org.Hs.eg.db, keys=row.names(resSigSFIvsNIT),
                                    column="GENENAME", keytype="ENSEMBL",
                                    multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigELIvsSFI$symbol <- mapIds(org.Hs.eg.db, keys=row.names(resSigELIvsSFI),
                                    column="SYMBOL", keytype="ENSEMBL", multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigELIvsSFI$entrez <- mapIds(org.Hs.eg.db, keys=row.names(resSigELIvsSFI),
                                    column="ENTREZID", keytype="ENSEMBL",
                                    multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
resSigELIvsSFI$genename <- mapIds(org.Hs.eg.db, keys=row.names(resSigELIvsSFI),
                                    column="GENENAME", keytype="ENSEMBL",
                                    multiVals="first")

## 'select()' returned 1:many mapping between keys and columns
• Ordenació dels resultats i conservació en arxius csv
resOrdELIvsNIT <- resSigELIvsNIT[order(resSigELIvsNIT$padj),]
resOrdSFIvsNIT <- resSigSFIvsNIT[order(resSigSFIvsNIT$padj),]
resOrdELIvsSFI <- resSigELIvsSFI[order(resSigELIvsSFI$padj),]

dfELIvsNIT <- as.data.frame(resOrdELIvsNIT)
write.csv(dfELIvsNIT, file = "results/resultsELIvsNIT.csv")

dfSFIvsNIT <- as.data.frame(resOrdSFIvsNIT)
write.csv(dfSFIvsNIT, file = "results/resultsSFIvsNIT.csv")

dfELIvsSFI <- as.data.frame(resOrdELIvsSFI)
write.csv(dfELIvsSFI, file = "results/resultsELIvsSFI.csv")

```

3.3.6. Comparació entre comparacions

- Creació de la matriu necessària per fer el diagrama de Venn

```
library(limma)

## 
## Attaching package: 'limma'

## The following object is masked from 'package:DESeq2':
## 
##     plotMA

## The following object is masked from 'package:BiocGenerics':
## 
##     plotMA

selectedELIvsNIT <- subset(dfELIvsNIT, abs(log2FoldChange) > 1)
selectedELIvsNIT <- row.names(selectedELIvsNIT)

selectedSFIvsNIT <- subset(dfSFIvsNIT, abs(log2FoldChange) > 1)
selectedSFIvsNIT <- row.names(selectedSFIvsNIT)

selectedELIvsSFI <- subset(dfELIvsSFI, abs(log2FoldChange) > 1)
selectedELIvsSFI <- row.names(selectedELIvsSFI)

selectednames <- unique(c(selectedELIvsNIT, selectedSFIvsNIT, selectedELIvsSFI))

selected <- matrix(rep(0, length(selectednames)*3), ncol = 3)
colnames(selected) <- c("ELI vs NIT", "SFI vs NIT", "ELI vs SFI")
row.names(selected) <- selectednames

for (i in 1:length(selectednames)) {
    if (selectednames[i] %in% selectedELIvsNIT) {
        selected[i,1] <- 1
    }
}

for (i in 1:length(selectednames)) {
    if (selectednames[i] %in% selectedSFIvsNIT) {
        selected[i,2] <- 1
    }
}

for (i in 1:length(selectednames)) {
    if (selectednames[i] %in% selectedELIvsSFI) {
        selected[i,3] <- 1
    }
}

head(selected, 10)

##          ELI vs NIT SFI vs NIT ELI vs SFI
## ENSG00000170054      1         1         0
## ENSG00000253742      1         1         0
## ENSG00000237638      1         1         0
```

```

## ENSG00000253131      1      1      0
## ENSG00000211654      1      1      0
## ENSG00000211685      1      1      0
## ENSG00000132465      1      1      0
## ENSG00000170476      1      1      0
## ENSG00000249096      1      1      1
## ENSG00000167483      1      1      1

```

- Diagrama de Venn

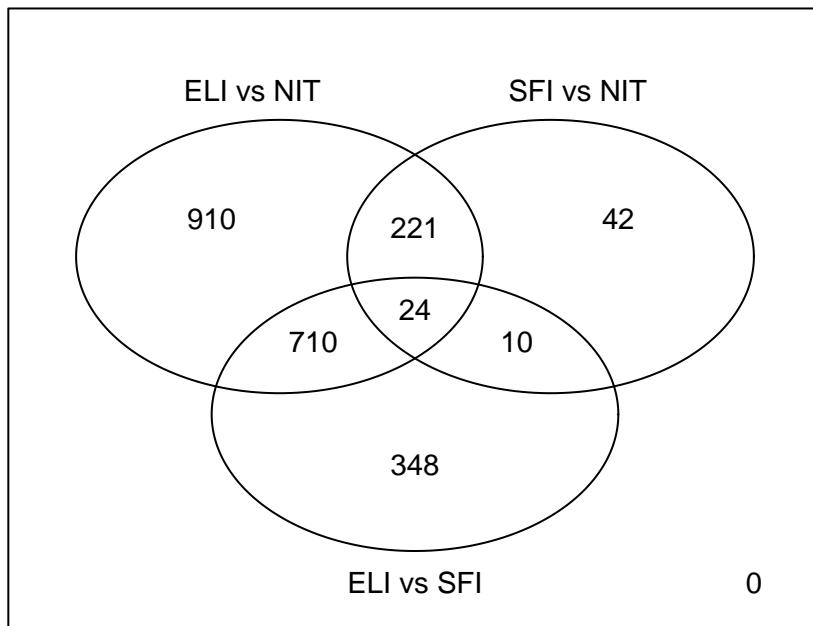
```

selectedCounts <- vennCounts(selected)

vennDiagram(selectedCounts, cex=0.9)
title("Gens diferencialment expressats segons cada comparació.")

```

Gens diferencialment expressats segons cada comparació.



3.3.7. Anàlisi de significació biològica

- Llistat d'identificadors *Entrez* dels gens

```

selectedIDs <- list(dfELIvsNIT$entrez, dfSFIvsNIT$entrez, dfELIvsSFI$entrez)
names(selectedIDs) <- c("ELIvsNIT", "SFIvsNIT", "ELIvsSFI")

```

- Generació del llistat de gens humans coneguts a partir de les anotacions del paquet *org.Hs.eg.db*

```

library(AnnotationDbi)
mapped_genes <- mappedkeys(org.Hs.egPATH)

```

- Identificació de *pathways* i generació dels arxius de resultats i dels gràfics

```

library(clusterProfiler)

## 
## Registered S3 method overwritten by 'enrichplot':
##   method           from
##   fortify.enrichResult DOSE

## clusterProfiler v3.14.3 For help: https://guangchuangyu.github.io/software/clusterProfiler
## 
## If you use clusterProfiler in published research, please cite:
## Guangchuang Yu, Li-Gen Wang, Yanyan Han, Qing-Yu He. clusterProfiler: an R package for comparing bio
## 
## Attaching package: 'clusterProfiler'

## The following object is masked from 'package:DelayedArray':
## 
##     simplify

listOfData <- selectedIDs[1:3]
comparisonsNames <- names(listOfData)

for (i in 1:length(listOfData)){
  genesIn <- listOfData[[i]]
  comparison <- comparisonsNames[i]

  enrich.KEGG <- enrichKEGG(gene = genesIn,
                             organism = 'hsa',
                             pvalueCutoff = 0.1,
                             pAdjustMethod = "BH",
                             universe = mapped_genes)

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

    png(file=paste0("./results/","enrichKEGGBarplot.",comparison,".png"),
        width = 800)
    print(barplot(enrich.KEGG, showCategory = 15, font.size = 8,
                  title = paste0("Anàlisi d'enriquiment de pathways KEGG per ", comparison))
    dev.off()

    png(file = paste0("./results/","enrichKEGGcnetplot.",comparison,".png"))
    print(cnetplot(enrich.KEGG, categorySize = "geneNum",
                  schowCategory = 15, vertex.label.cex = 0.75))
    dev.off()
  }
}

```

4. Resultats

- Imatges dels gràfics de resultats

```
library(knitr)
include_graphics("results/enrichKEGGBarplot.ELIvsNIT.png")
```

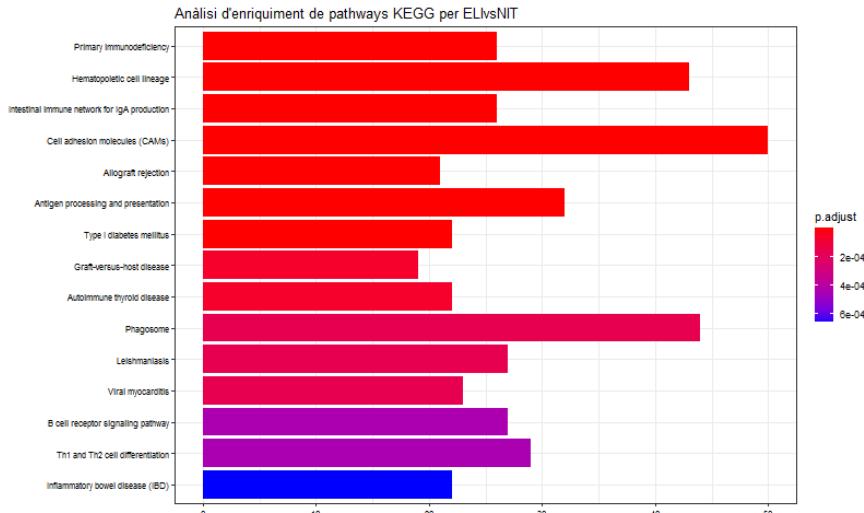


Figure 1: Gràfic de barres dels pathways per ELI vs NIT

```
include_graphics("results/enrichKEGGnetplot.ELIvsNIT.png")
include_graphics("results/enrichKEGGBarplot.SFIvsNIT.png")
include_graphics("results/enrichKEGGnetplot.SFIvsNIT.png")
include_graphics("results/enrichKEGGBarplot.ELIvsSFI.png",)
include_graphics("results/enrichKEGGnetplot.ELIvsSFI.png")
```

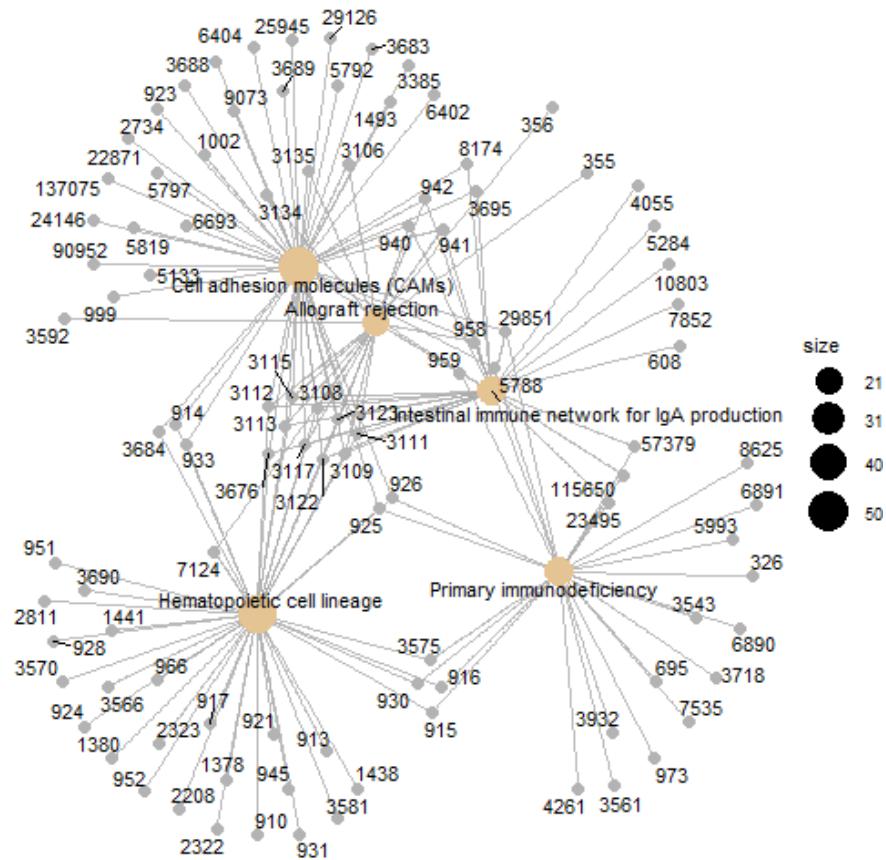


Figure 2: Gràfic de ret dels pathways per ELI vs NIT

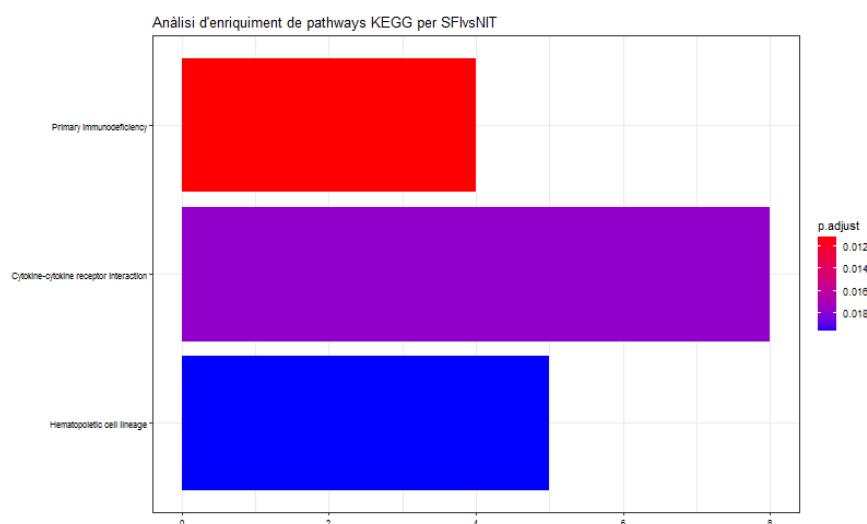


Figure 3: Gràfic de barres dels pathways per SFI vs NIT

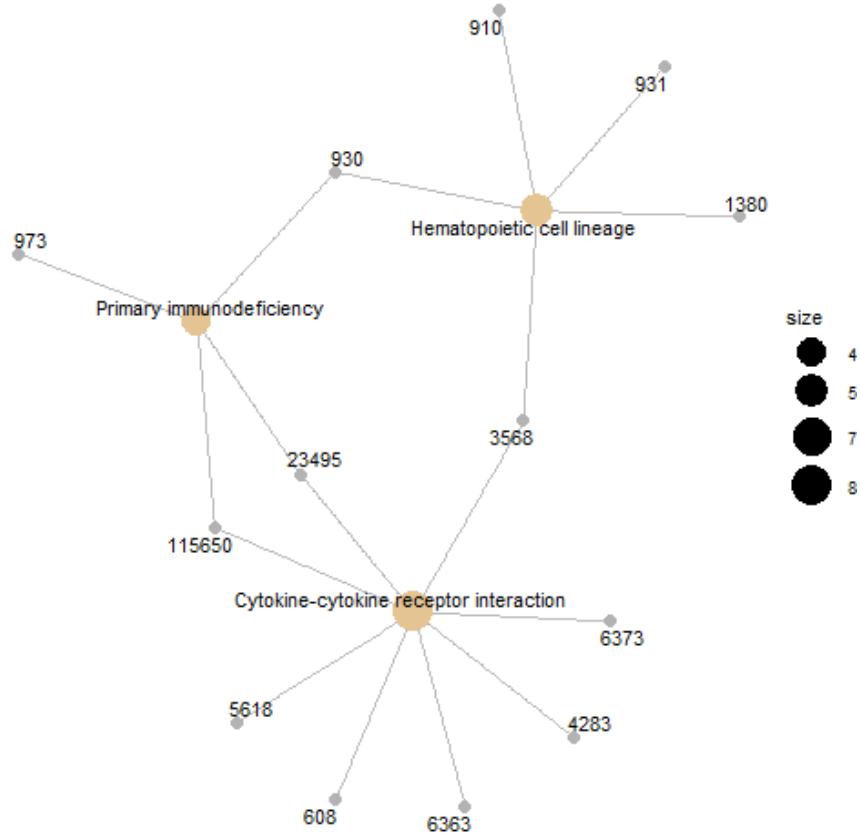


Figure 4: Gràfic de ret dels pathways per SFI vs NIT

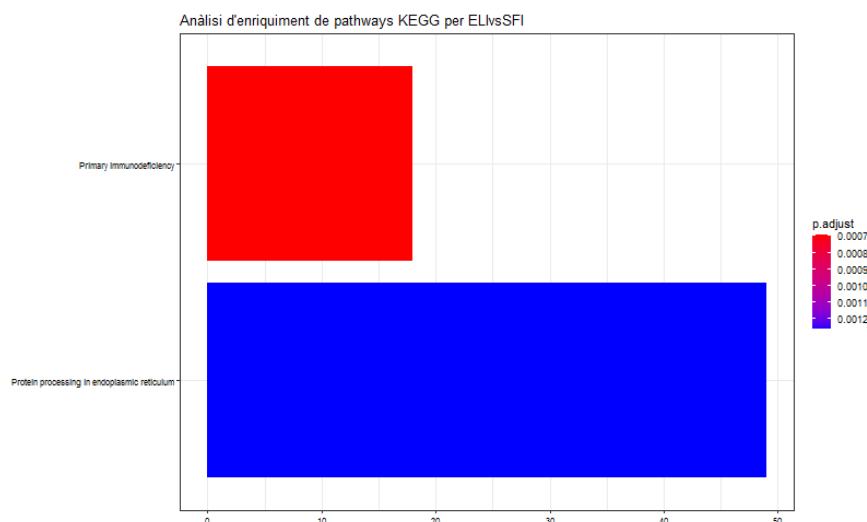


Figure 5: Gràfic de barres dels pathways per ELI vs SFI

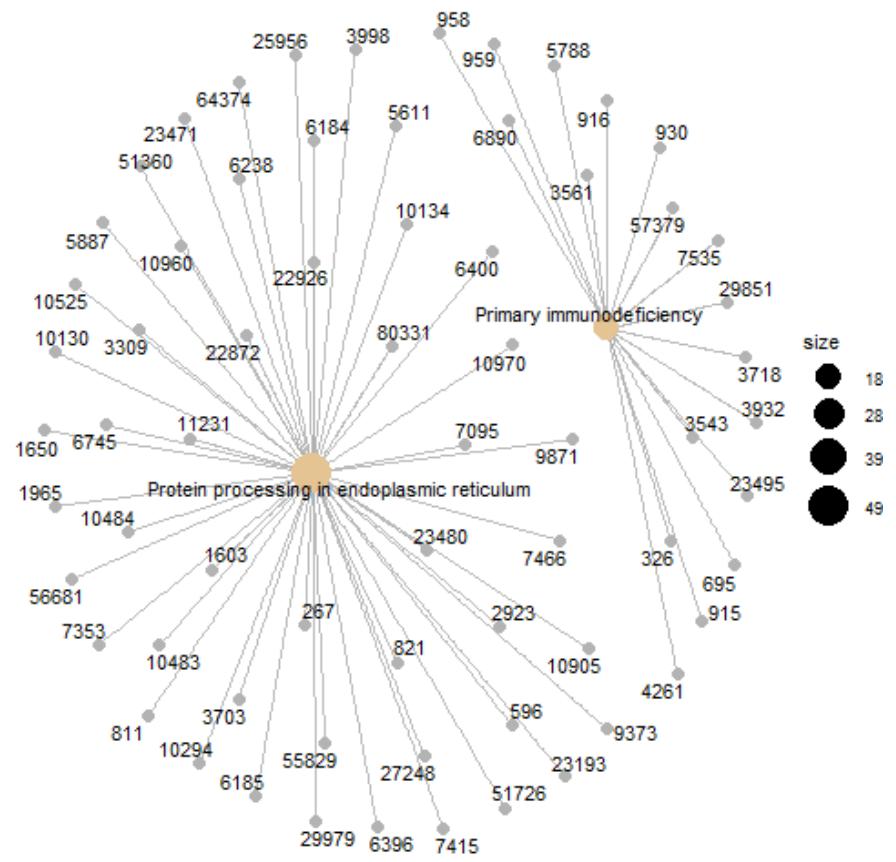


Figure 6: Gràfic de ret dels pathways per ELI vs SFI