Análisis de datos ómicos - Segunda prueba de evaluación continua

Aràntzazu Alonso Carrasco

2023-05-21

Índice

Introducción y objetivos

Para este informe, analizaremos el conjunto de datos `GDS2107` con la serie `GSE3311`, titulado *Long-term ethanol consumption effect on pancreas*. Este estudio se llevó a cabo utilizando muestras de rata común (*Rattus norvegicus*). El conjunto de datos proporciona información detallada sobre los perfiles de expresión génica en el páncreas de ratas sometidas a consumo prolongado de etanol. A través de este análisis, se busca comprender los cambios moleculares y los procesos biológicos involucrados en la respuesta del páncreas al consumo crónico de etanol. Se sabe que el consumo de etanol a largo plazo no causa pancreatitis aguda, pero sensibiliza el páncreas a agresiones posteriores. Para determinar si las alteraciones en la expresión génica pancreática podrían participar en la sensibilización mediada por etanol, se realizaron análisis de perfiles génicos.

Con todo, el objetivo de este estudio fue investigar los efectos del consumo crónico de etanol en el tejido pancreático.

Métodos

Este estudio, los animales se separaron en dos grupos en función del tipo de alimentación: dieta Lieber-DeCarli que contenía etanol o dieta control. Después de 8 semanas, se analizó la expresión del ARN pancreático utilizando Affymetrix GeneChips.

Para el análisis de los datos obtenidos a partir de los archivos .CEL, se utilizó el lenguaje de programación R y los paquetes especializados de Bioconductor. Estos paquetes proporcionan herramientas y funciones específicas para el procesamiento y análisis de datos de expresión génica.

Los archivos .CEL fueron cargados en R y se realizó una serie de pasos de preprocesamiento, incluyendo la normalización y filtrado de los datos. Para la normalización, se usó la función `rma()` del paquete `affy`, que convierte un objeto `AffyBatch` en un objeto `ExpressionSet` usando la medida de expresión media robusta multiarray (RMA). Para el filtrado, se utilizó la función `nsFilter` del paquete `genefilter`. A continuación, se llevó a cabo un análisis de expresión diferencial para identificar los genes cuya expresión se veía afectada por la dieta que contenía etanol en comparación con la dieta control.

Además, se realizó un análisis de enriquecimiento génico para identificar las vías biológicas y funciones que estaban sobre-representadas entre los genes diferencialmente expresados. Se utilizaron bases de datos y ontologías específicas, como Gene Ontology (GO) y bases de datos de anotaciones génicas, para llevar a cabo este análisis.

Por último, se realizaron visualizaciones gráficas para presentar de manera clara y concisa los resultados obtenidos, como gráficos de expresión diferencial, gráficos de enriquecimiento génico y redes de interacción génica.

Resultado

En este estudio se empleó un diseño de bloques aleatorizados, considerando dos tratamientos: control y etanol. Las ratas utilizadas en el estudio pertenecían a una única cepa. El diseño en bloques aleatorizados permite asignar de forma aleatoria los tratamientos a los animales, lo que ayuda a minimizar el efecto de posibles variables de confusión.

La matriz de diseño, por tanto, es la siguiente:

```
##        control_diet ethanol_diet
## 93_C              1            0
## 94_C              1            0
## 95_C              1            0
## 96_Eth            0            1
## 97_Eth            0            1
## 98_Eth            0            1
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$lev
## [1] "contr.treatment"
```

Como podemos ver, hay tres animales que pertenecen al grupo control y tres animales que pertenecen al grupo alimentado con etanol. Por tanto, la pregunta que debemos hacernos es *cuál es el efecto del etanol*, que de forma paramétrica quedaría así:

$$\alpha_1 - \alpha_2 = 0$$

Donde $\alpha_1 = \mathrm{E}(ethanol)$ y $\alpha_2 = \mathrm{E}(control)$. Por tanto, la matriz de constrastes es la siguiente:

```
##               Contrasts
## Levels         Ethanol.vs.control
##   control_diet                 -1
##   ethanol_diet                  1
```

Una vez definidos los contrastes que se pretende investigar, se procedió a evaluar la calidad de los datos utilizando los archivos .CEL, los cuales fueron cargados en un objeto "ExpressionFeatureSet" denominado "rawData". En la Figura 1 se presentan tres gráficos: un boxplot, un clúster jerárquico y un análisis de componentes principales.

En el boxplot, se observa que no existen valores atípicos (outliers) en los datos, ya que no se identifican muestras con medias muy diferentes al resto. Esto indica que no hay observaciones extremas que puedan afectar el análisis posterior.

En el clúster jerárquico, se aprecia una tendencia a agrupar las muestras del grupo control por un lado y las del grupo etanol por otro. Este resultado es consistente con las diferencias esperadas entre los dos tratamientos y respalda la selección de los grupos para el estudio.

En el análisis de componentes principales, se observa que las muestras del grupo control tienen una distribución similar entre sí, mientras que las del grupo etanol presentan una mayor dispersión. No obstante, no se identifican patrones preocupantes en este análisis.

En resumen, los gráficos de calidad de los datos (boxplot, clúster jerárquico y análisis de componentes principales) indican que los datos obtenidos a partir de los archivos .CEL presentan una calidad adecuada para realizar el análisis de expresión génica diferencial. No se detectaron valores atípicos que puedan sesgar los resultados y se observaron patrones esperados en relación con los grupos de tratamiento. Esto proporciona confianza en la calidad y fiabilidad de los datos para su posterior análisis.

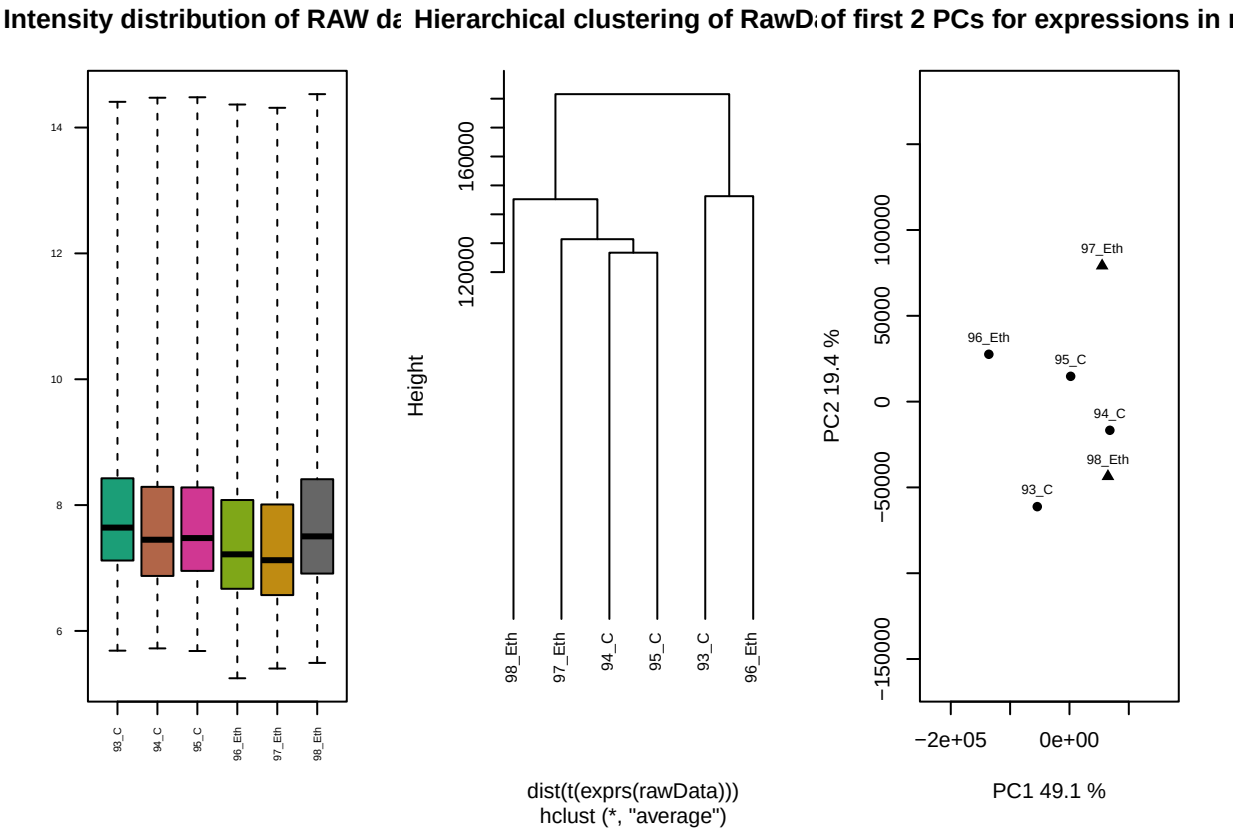


Figura 1: Gráficos del control de calidad

Por lo tanto, se procedió a realizar la normalización y filtrado de los datos para obtener el objeto "filteredData", el cual se utilizó para ajustar un modelo lineal. A partir de este modelo, se utilizó la función "topTable" del paquete "limma" para seleccionar únicamente los genes con un "log-fold-change" mayor a 1 y un p-valor ajustado menor a 0.05.

Para poder realizar el análisis de los genes seleccionados, se realizó una anotación utilizando el paquete "rae230a.db", el cual contiene los datos de anotación del array RAE230A de Affymetrix utilizado en este estudio. Específicamente, se utilizó la anotación "SYMBOL" para asignar nombres de genes a los resultados

obtenidos.

En la Figura 2 se presenta un gráfico de tipo “volcano plot” que muestra los genes diferencialmente expresados entre el grupo control y el grupo etanol. En el eje horizontal se representa el logaritmo del “fold change” o cambio entre los grupos, lo cual está relacionado con la significación biológica del cambio. En el eje vertical se muestra el p-valor del test en una escala logarítmica negativa, donde los p-valores más pequeños se encuentran en la parte superior del gráfico. Este eje refleja la evidencia estadística o la confiabilidad del cambio observado. En relación a estos datos, se pueden destacar los genes “Reg3a” y “Mt2A” como ejemplos de genes diferencialmente expresados entre los grupos control y etanol.

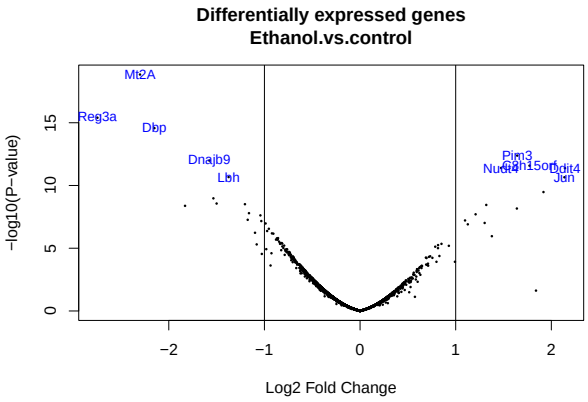


Figura 2: Volcano Plot de los genes diferencialmente expresados

Además, en la Figura 3 se muestra un HeatMap de los datos, el cual permite visualizar las expresiones de cada gen agrupándolas para resaltar los genes que están simultáneamente regulados al alza (upregulated) o a la baja (downregulated), formando perfiles de expresión. Podemos observar que los perfiles de expresión de los animales 98_Eth y 97_Eth son prácticamente idénticos, al igual que ocurre con los tres animales del grupo control.

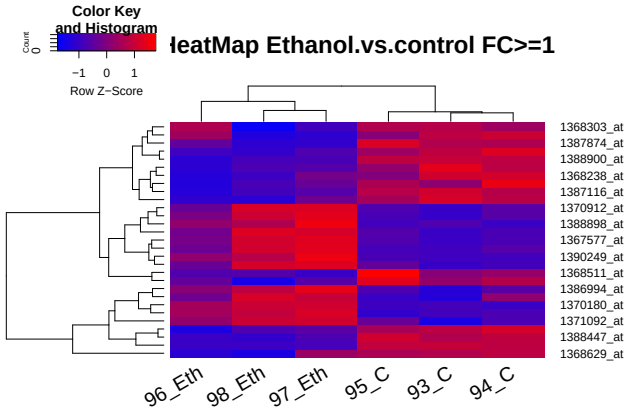


Figura 3: HeatMap del grupo etanol contra el grupo control usando un fold change de 1

Para concluir, se realizó un análisis de significación biológica utilizando el paquete `GOstats`. En la Figura 4 se muestra un dotplot que presenta los diez términos enriquecidos más significativos ordenados según su p-valor ajustado, donde los términos ubicados en la parte superior del gráfico son los más significativos. Cada punto en el gráfico representa un término, y su tamaño está proporcionalmente relacionado con el número de genes asociados a dicho término. En este análisis, se observó que los términos más significativos fueron *rhythmic process*, *cellular response to ROS* y *response to hydrogen peroxide*. El primer término está relacionado con

la periodicidad de los procesos biológicos, mientras que los dos últimos están relacionados con respuestas celulares ante la presencia de especies reactivas de oxígeno y peróxido de hidrógeno, respectivamente.

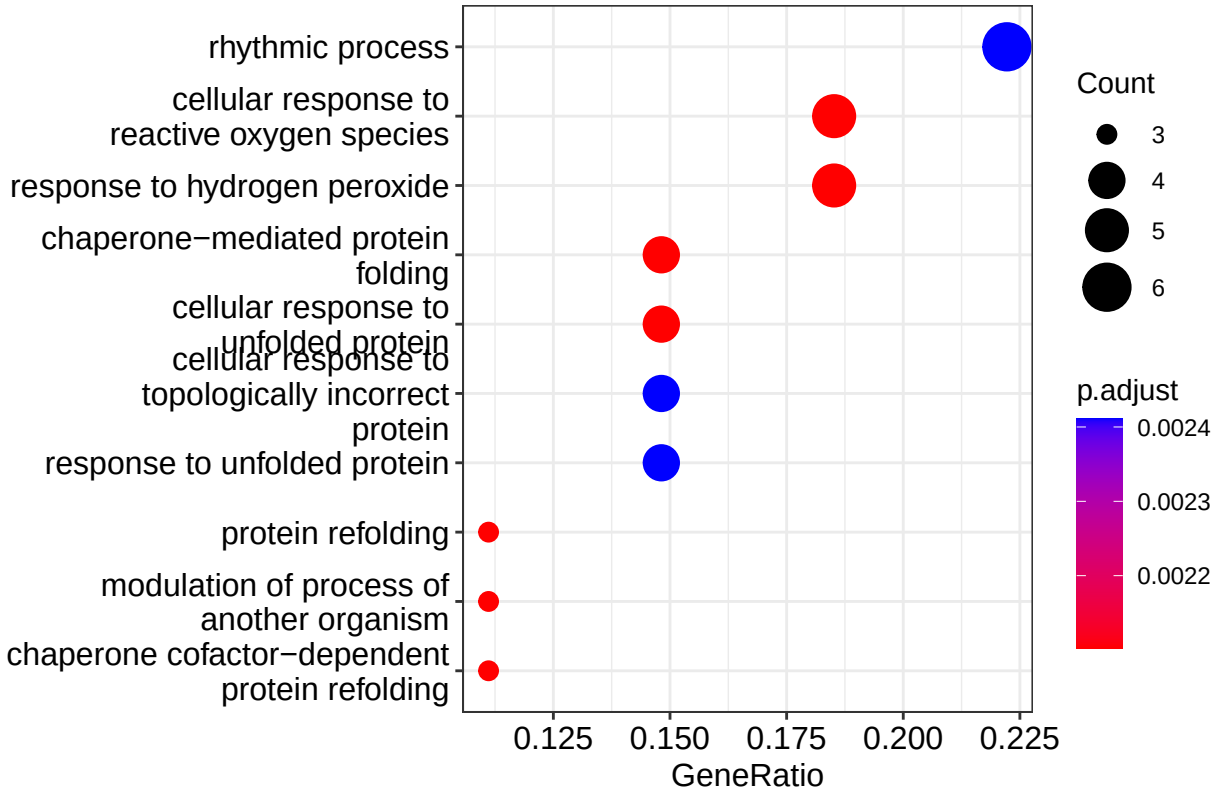


Figura 4: Dotplot de los diez términos más enriquecidos

En la figura 5 se presenta una visualización de los términos enriquecidos de Gene Ontology (GO). Cada término GO se representa como un nodo en el gráfico, y las líneas que los conectan indican las relaciones jerárquicas entre ellos. Los nodos de color rojo representan los términos con un p-valor ajustado más bajo, lo que indica su mayor significancia. En este caso, se observa que los términos *rhythmic process* y *response to oxygen-containing compound* continúan destacándose como términos significativos.

En la figura 6 se muestra una representación visual de la red génica asociada a los términos enriquecidos en el análisis de enriquecimiento génico. En este gráfico, los nodos representan genes y las líneas entre ellos indican las interacciones o relaciones entre los genes en la red. Mediante esta representación, es posible identificar grupos de genes que están estrechamente relacionados entre sí y que participan en procesos biológicos comunes. Esta visualización puede resultar útil para identificar genes candidatos para estudios posteriores o para comprender mejor las vías biológicas implicadas en la condición experimental. En el caso particular de este análisis, se destaca el gen *Hspb1*, el cual está relacionado con términos como *chaperone-mediated protein folding*, *protein-refolding*, *response to hydrogen peroxide* y *cellular response to ROS*.

Para terminar, la Figura 7 muestra un mapa de enriquecimiento que agrupa los términos enriquecidos por similitud medida por solapamiento de genes entre términos. Los nodos más grandes indican términos más significativos, y los nombres de los términos se muestran dentro de los nodos. Este tipo de gráfico puede ayudar a identificar grupos de términos que están relacionados y que pueden estar implicados en procesos biológicos similares o que pueden ser regulados por los mismos genes. Así vemos que, por ejemplo, los procesos rítmicos están relacionados con la respuesta celular a las especies reactivas del oxígeno.

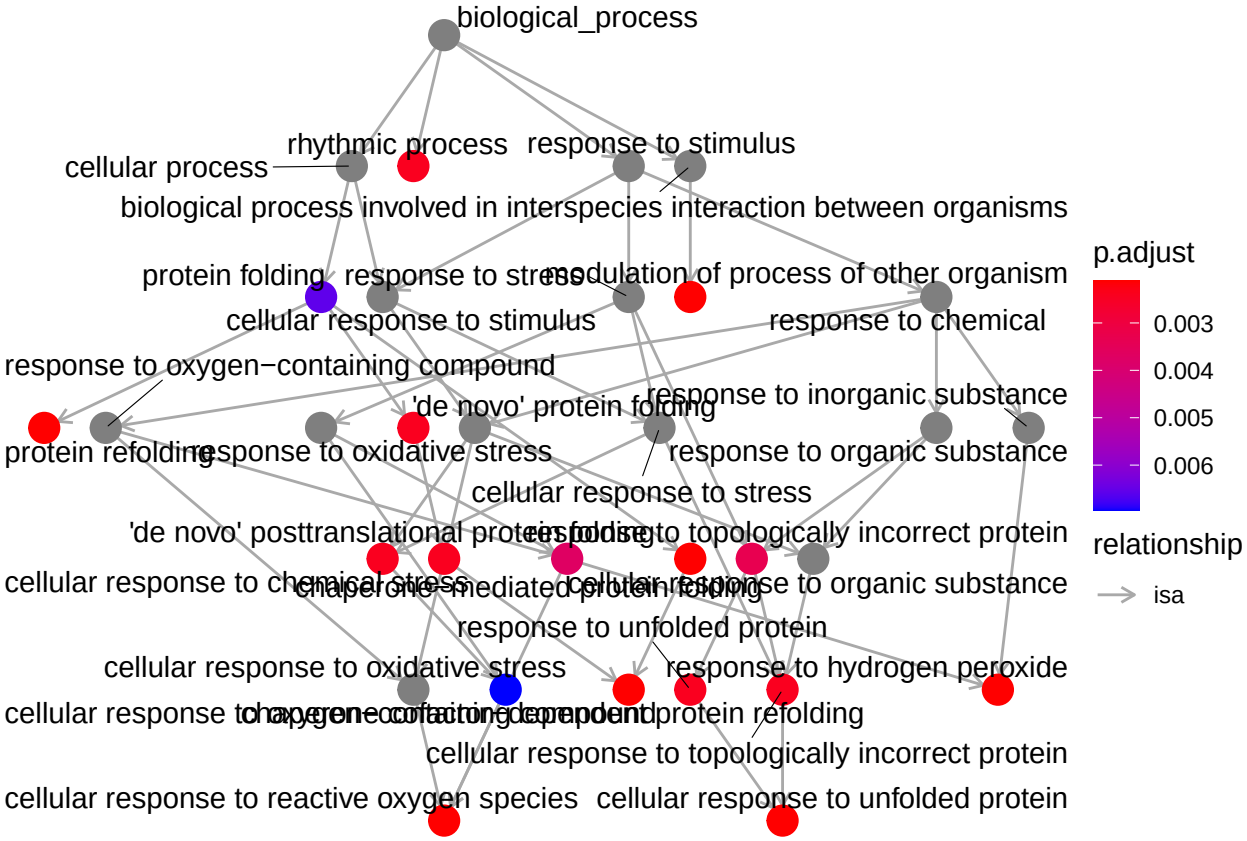


Figura 5: Términos Gene Ontology enriquecidos

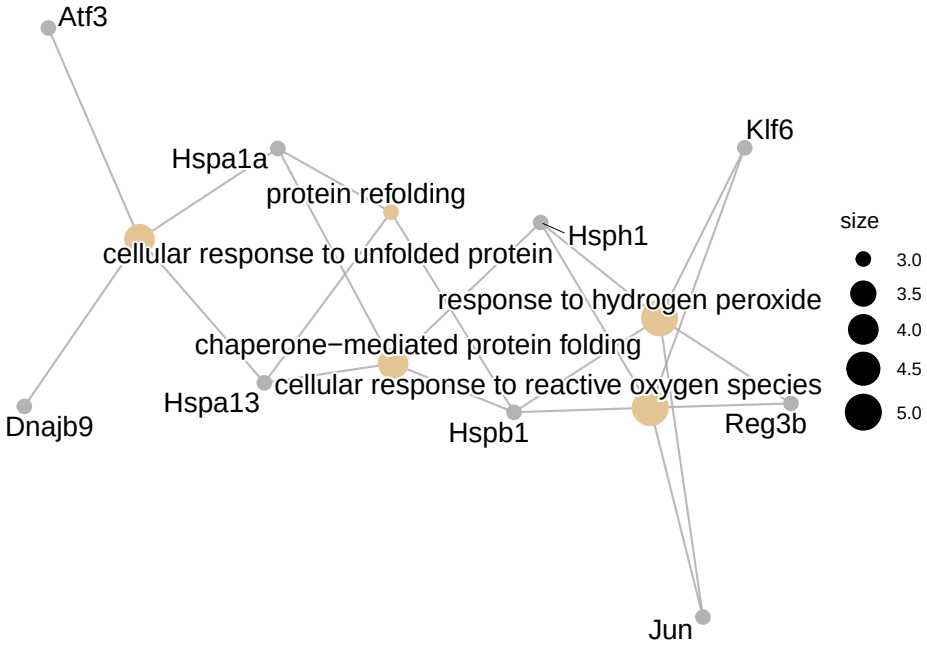


Figura 6: Red génica asociada a los términos enriquecidos

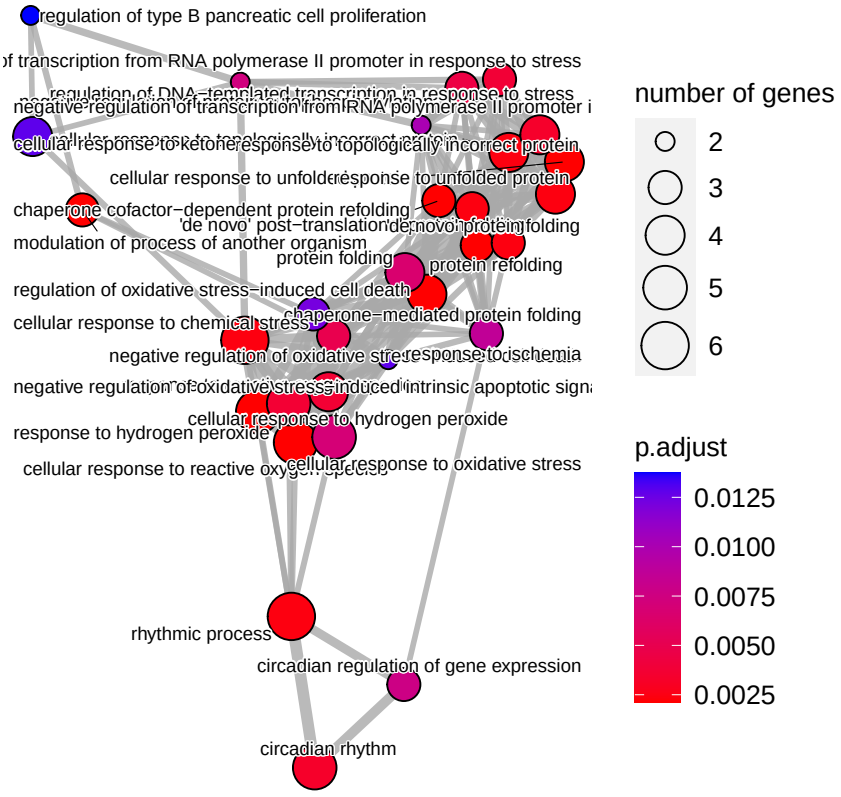


Figura 7: Mapa de enriquecimiento

Discusión

El objetivo de este estudio fue investigar los posibles cambios en la expresión génica de las células acinares pancreáticas durante la alimentación prolongada con etanol. Mediante el análisis de los datos de expresión génica y el enriquecimiento funcional, se han identificado términos enriquecidos significativos que proporcionan información sobre los procesos biológicos alterados en esta condición experimental.

En el análisis de enriquecimiento, se observó la significancia de los procesos rítmicos y las respuestas celulares a las especies reactivas de oxígeno o al peróxido de hidrógeno. Estos hallazgos sugieren la implicación del estrés celular en la respuesta pancreática al consumo prolongado de etanol. Es importante destacar que se identificaron genes específicos, como Hspb1, que están relacionados con estos procesos. Estos genes podrían convertirse en candidatos de interés para investigaciones futuras enfocadas en comprender mejor los mecanismos subyacentes y las vías biológicas involucradas en esta condición.

Referencias

Kubisch, C. H., Gukovsky, I., Lugea, A., Pandol, S. J., Kuick, R., Misek, D. E., ... Logsdon, C. D. (2006). Long-term Ethanol Consumption Alters Pancreatic Gene Expression in Rats. Pancreas, 33(1), 68–76. doi: 10.1097/01.mpa.0000226878.81377.94

Apéndice

```r
#-------------------------------------------------------
# Instalación de paquetes necesarios
#--------------------------------------------------------

if (!require(BiocManager)) install.packages("BiocManager")

installifnot <- function (pkg){
  if (!require(pkg, character.only=T)){
    BiocManager::install(pkg)
  }
}
installifnot("pd.mogene.1.0.st.v1")
installifnot("mogene10sttranscriptcluster.db")
installifnot("oligo")
installifnot("limma")
installifnot("Biobase")
installifnot("arrayQualityMetrics")
installifnot("genefilter")
installifnot("annotate")
installifnot("xtable")
installifnot("gplots")
installifnot("GOstats")
installifnot("gplots")
installifnot("GEOquery")
installifnot("rae230a.db")
```

```r
workingDir <-getwd()
dataDir <- file.path(workingDir, "dades")
resultsDir <- file.path(workingDir, "results")
```

```r
library(Biobase)
#TARGETS
targets <-read.csv(file=file.path(dataDir,"targets.csv"), header = TRUE, sep=";")
#DEFINE SOME VARIABLES FOR PLOTS
sampleNames <- as.character(targets$ShortName)
# Creamos un objeto AnnotatedDataFrame
targets <- AnnotatedDataFrame(targets)
```

```r
CELfiles <- targets$fileName
rawData <- read.celfiles(file.path(dataDir,CELfiles), phenoData=targets)
```

```
## Platform design info loaded.
```

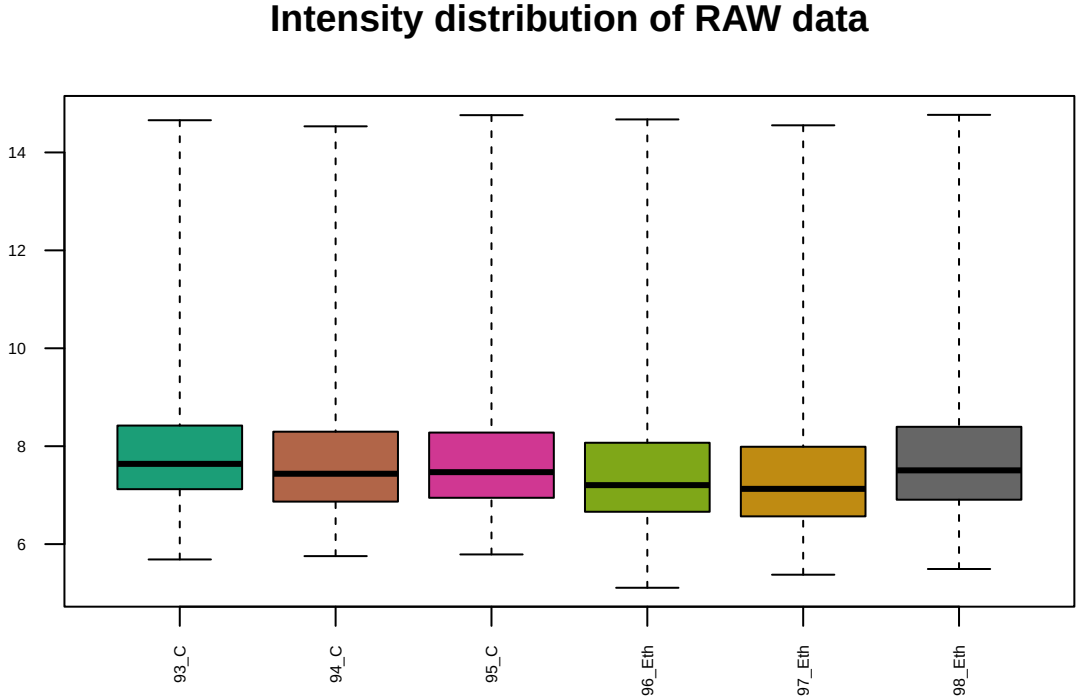
```
## Reading in : G:/Bioinformàtica/Curs 2022-2023/2n semestre/Anàlisi de dades òmiques/PAC2/Analisis_de_
## Reading in : G:/Bioinformàtica/Curs 2022-2023/2n semestre/Anàlisi de dades òmiques/PAC2/Analisis_de_
## Reading in : G:/Bioinformàtica/Curs 2022-2023/2n semestre/Anàlisi de dades òmiques/PAC2/Analisis_de_
## Reading in : G:/Bioinformàtica/Curs 2022-2023/2n semestre/Anàlisi de dades òmiques/PAC2/Analisis_de_
## Reading in : G:/Bioinformàtica/Curs 2022-2023/2n semestre/Anàlisi de dades òmiques/PAC2/Analisis_de_
## Reading in : G:/Bioinformàtica/Curs 2022-2023/2n semestre/Anàlisi de dades òmiques/PAC2/Analisis_de_
```

```
## Warning in read.celfiles(file.path(dataDir, CELfiles), phenoData = targets):
## 'channel' automatically added to varMetadata in phenoData.
```

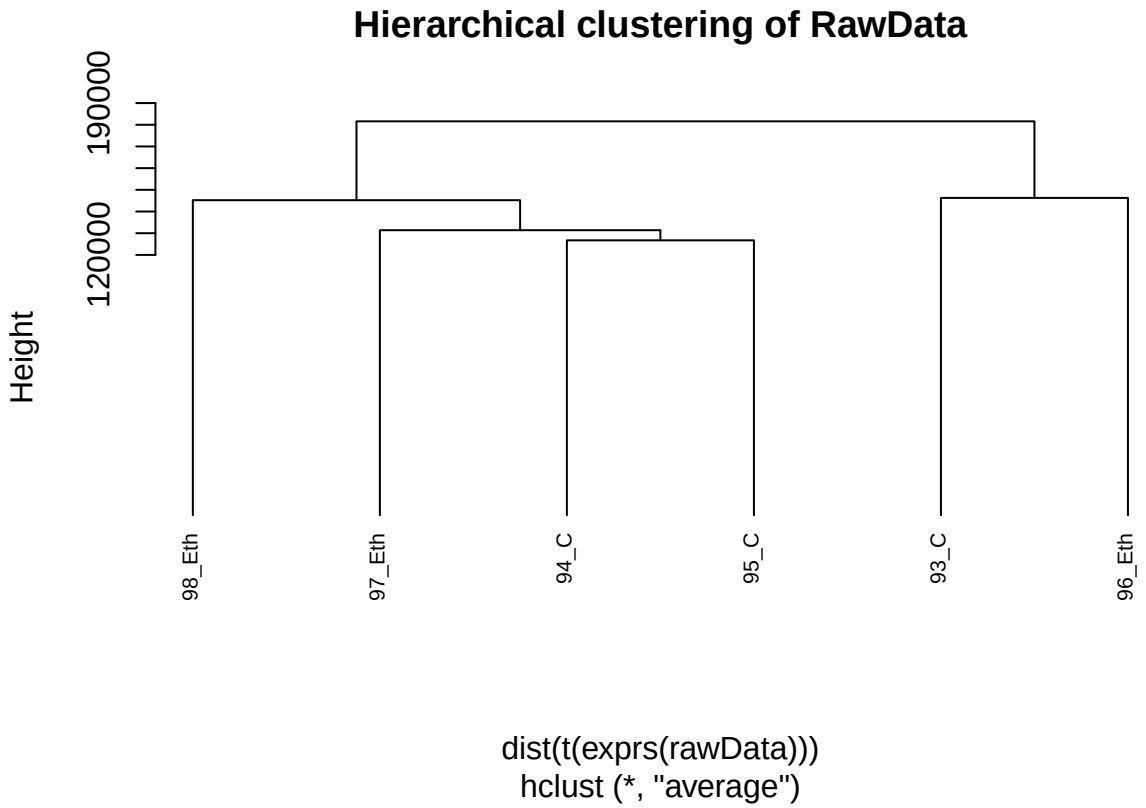
```
rawData
```

```
## ExpressionFeatureSet (storageMode: lockedEnvironment)
## assayData: 362404 features, 6 samples
##   element names: exprs
## protocolData
##   rowNames: 1 2 ... 6 (6 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: 1 2 ... 6 (6 total)
##   varLabels: fileName grupos ShortName
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.rae230a
```

```
#BOXPLOT
boxplot(rawData, which="all",las=2, main="Intensity distribution of RAW data",
        cex.axis=0.5, names=sampleNames)
```

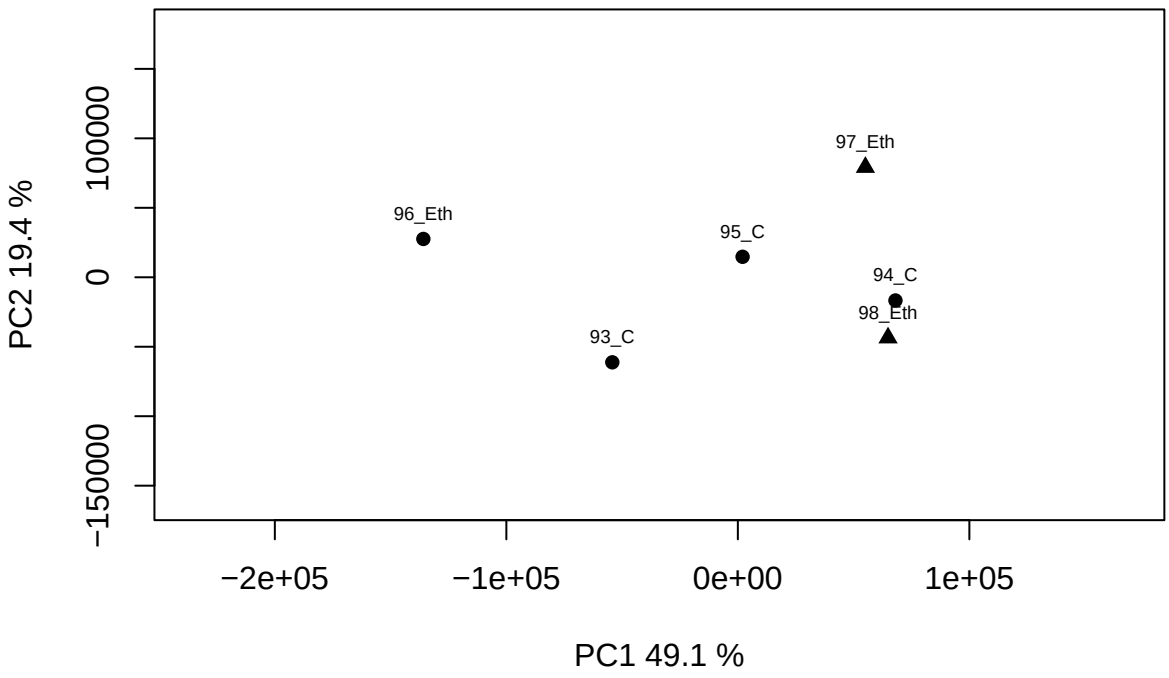


```
#HIERARQUICAL CLUSTERING
clust.euclid.average <- hclust(dist(t(exprs(rawData))),method="average")
plot(clust.euclid.average, labels=sampleNames, main="Hierarchical clustering of RawData", cex=0.7,  hang
```




```
#PRINCIPAL COMPONENT ANALYSIS
plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE,
                      formapunts=NULL, myCex=0.8,...)
{
  pcX<-prcomp(t(X), scale=scale) # o prcomp(t(X))
  loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
  xlab<-c(paste("PC1",loads[1],"%"))
  ylab<-c(paste("PC2",loads[2],"%"))
  if (is.null(colors)) colors=1
  plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=colors, pch=formapunts,
       xlim=c(min(pcX$x[,1])-100000, max(pcX$x[,1])+100000),
       ylim=c(min(pcX$x[,2])-100000, max(pcX$x[,2])+100000))
  text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=myCex)
  title(paste("Plot of first 2 PCs for expressions in", dataDesc, sep=" "), cex=0.8)
}

plotPCA(exprs(rawData), labels=sampleNames, dataDesc="raw data",
        formapunts=c(rep(16,4),rep(17,4)), myCex=0.6)
```




```
# Avoid re-running it each time  the script is executed.
rerun <- FALSE
if(rerun){
  arrayQualityMetrics(eset,  reporttitle="QC_RawData", force=TRUE)
}
```

```
# Normalización
eset<-rma(rawData)
```

```
## Background correcting
## Normalizing
## Calculating Expression
```

```
write.exprs(eset, file.path(resultsDir, "NormData.txt"))
eset
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 15923 features, 6 samples
##   element names: exprs
## protocolData
##   rowNames: 1 2 ... 6 (6 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: 1 2 ... 6 (6 total)
```

```
##   varLabels: fileName grupos ShortName
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.rae230a
```

```
# Filtrado
library(genefilter)
library(rae230a.db)
annotation(eset) <- "rae230a.db"
eset_filtered <- nsFilter(eset, var.func=IQR,
         var.cutoff=0.75, var.filter=TRUE, require.entrez = TRUE,
         filterByQuantile=TRUE)
#NUMBER OF GENES REMOVED
print(eset_filtered)
```

```
## $eset
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 2690 features, 6 samples
##   element names: exprs
## protocolData
##   rowNames: 1 2 ... 6 (6 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: 1 2 ... 6 (6 total)
##   varLabels: fileName grupos ShortName
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: rae230a.db
##
## $filter.log
## $filter.log$numDupsRemoved
## [1] 2537
##
## $filter.log$numLowVar
## [1] 8070
##
## $filter.log$numRemoved.ENTREZID
## [1] 2620
##
## $filter.log$feature.exclude
## [1] 6
```

```
#NUMBER OF GENES IN
print(eset_filtered$eset)
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 2690 features, 6 samples
##   element names: exprs
## protocolData
##   rowNames: 1 2 ... 6 (6 total)
```

```
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
##   rowNames: 1 2 ... 6 (6 total)
##   varLabels: fileName grupos ShortName
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: rae230a.db
```

```
filteredEset <- eset_filtered$eset
filteredData <- exprs(filteredEset)
colnames(filteredData) <- pData(eset_filtered$eset)$ShortName
```

```
# Matriz de diseño
library(limma)
treat <- pData(filteredEset)$grupos
lev <- factor(treat, levels = unique(treat))
design <-model.matrix(~0+lev)
colnames(design) <- levels(lev)
rownames(design) <- sampleNames
print(design)
```

```
##        control_diet ethanol_diet
## 93_C              1            0
## 94_C              1            0
## 95_C              1            0
## 96_Eth            0            1
## 97_Eth            0            1
## 98_Eth            0            1
## attr(,"assign")
## [1] 1 1
## attr(,"contrasts")
## attr(,"contrasts")$lev
## [1] "contr.treatment"
```

```
#COMPARISON
cont.matrix1 <- makeContrasts(
        Ethanol.vs.control = ethanol_diet-control_diet,
        levels = design)
comparisonName <- "Efecto del etanol"
print(cont.matrix1)
```

```
##               Contrasts
## Levels         Ethanol.vs.control
##   control_diet                 -1
##   ethanol_diet                  1
```

```
#MODEL FIT
fit1 <- lmFit(filteredData, design)
fit.main1 <- contrasts.fit(fit1, cont.matrix1)
fit.main1 <- eBayes(fit.main1)
```

```
topTab <-  topTable (fit.main1, number=nrow(fit.main1), coef="Ethanol.vs.control", adjust="fdr",lfc=1, p
dim(topTab)
```

```
## [1] 29  6
```

```
head(topTab)
```

```
##                logFC   AveExpr          t      P.Value    adj.P.Val        B
## 1388271_at -2.301132 10.478841 -14.652405 1.437301e-19 3.866340e-16 33.84419
## 1387930_at -2.746280  7.733064 -11.929641 3.976415e-16 5.348278e-13 26.33449
## 1387874_at -2.150355  7.631634 -11.316954 2.694968e-15 2.416488e-12 24.50182
## 1367725_at  1.643399  8.988216   9.770816 4.165251e-13 2.801131e-10 19.64728
## 1387116_at -1.575883  6.899841  -9.500717 1.035018e-12 5.568396e-10 18.76720
## 1390249_at  1.771694  6.002867   9.226933 2.625876e-12 1.177268e-09 17.86615
```

```
# Anotar
library(AnnotationDbi)
keytypes(rae230a.db)
```

```
##  [1] "ACCNUM"       "ALIAS"        "ENSEMBL"      "ENSEMBLPROT"  "ENSEMBLTRANS"
##  [6] "ENTREZID"     "ENZYME"       "EVIDENCE"     "EVIDENCEALL"  "GENENAME"
## [11] "GENETYPE"     "GO"           "GOALL"        "IPI"          "ONTOLOGY"
## [16] "ONTOLOGYALL"  "PATH"         "PFAM"         "PMID"         "PROBEID"
## [21] "PROSITE"      "REFSEQ"       "SYMBOL"       "UNIPROT"
```

```
valid_keys <- keys(org.Rn.eg.db)
anotaciones <- AnnotationDbi::select(rae230a.db, keys = rownames(filteredData), columns = c("ENTREZID",
```

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

```
library(dplyr)
topTabAnotada <- topTab %>%
  mutate(PROBEID=rownames(topTab)) %>%
  left_join(anotaciones) %>%
  arrange(P.Value) %>%
  select(7,8,9, 1:6)
```

```
## Joining with `by = join_by(PROBEID)`
```

```
head(topTabAnotada)
```

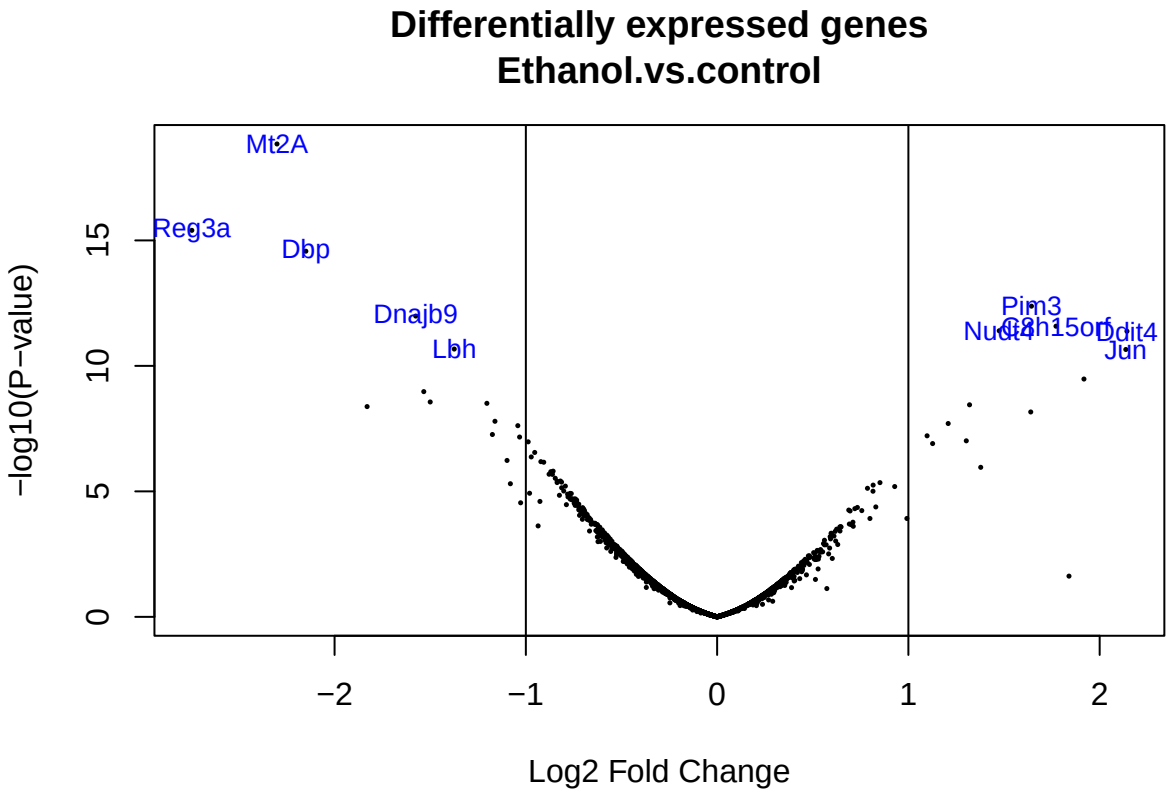
```
##      PROBEID ENTREZID     SYMBOL     logFC   AveExpr          t      P.Value
## 1 1388271_at   689415       Mt2A -2.301132 10.478841 -14.652405 1.437301e-19
## 2 1387930_at   171162      Reg3a -2.746280  7.733064 -11.929641 3.976415e-16
## 3 1387874_at    24309        Dbp -2.150355  7.631634 -11.316954 2.694968e-15
## 4 1367725_at    64534       Pim3  1.643399  8.988216   9.770816 4.165251e-13
## 5 1387116_at    24908     Dnajb9 -1.575883  6.899841  -9.500717 1.035018e-12
## 6 1390249_at   315702 C8h15orf39  1.771694  6.002867   9.226933 2.625876e-12
##      adj.P.Val        B
```

```
## 1 3.866340e-16 33.84419
## 2 5.348278e-13 26.33449
## 3 2.416488e-12 24.50182
## 4 2.801131e-10 19.64728
## 5 5.568396e-10 18.76720
## 6 1.177268e-09 17.86615
```

```
genenames <- AnnotationDbi::select(rae230a.db,
                    rownames(fit.main1), c("SYMBOL"))$SYMBOL
```

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

```
volcanoplot(fit.main1, highlight=10, names=genenames,
            main = paste("Differentially expressed genes", colnames(cont.matrix1), sep="\n"))
abline(v = c(-1, 1))
```

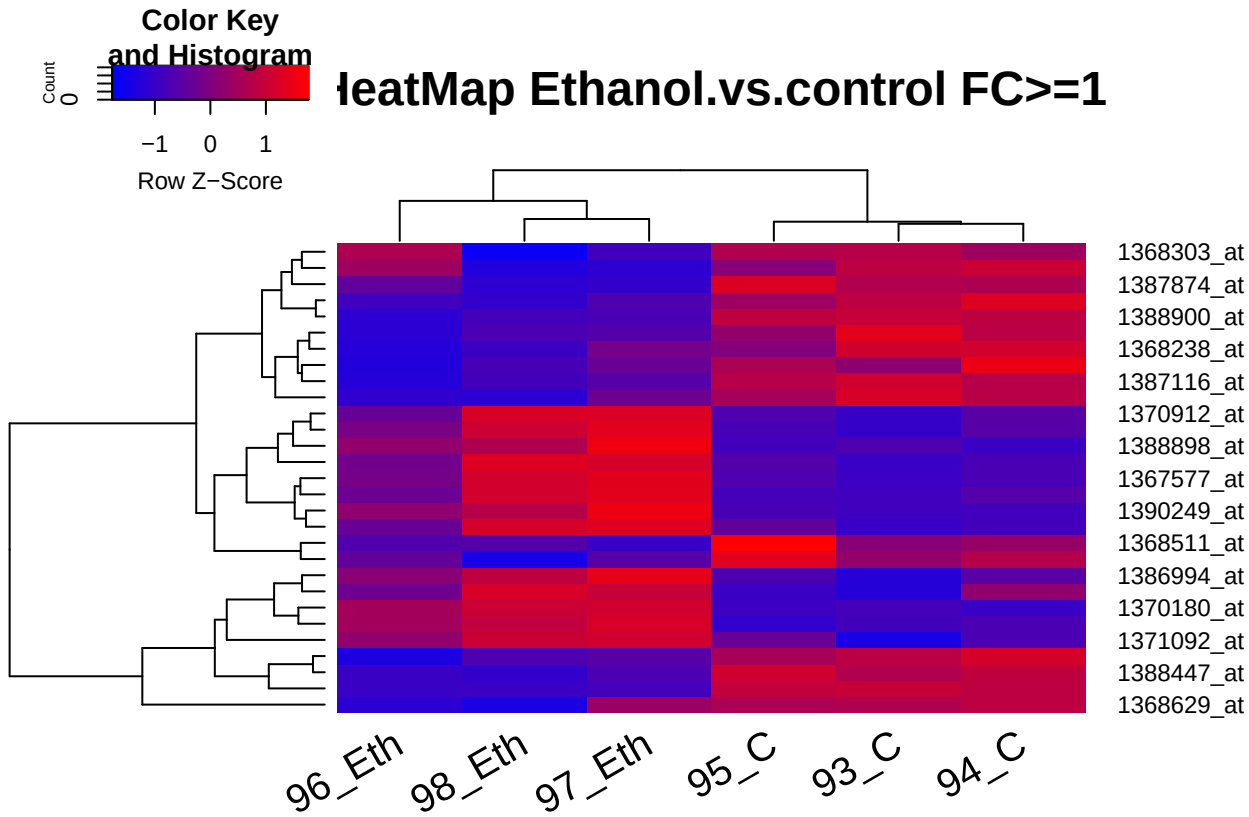



```
pdf(file.path(resultsDir,"Volcanos.pdf"))
volcanoplot(fit.main1, highlight = 10, names = genenames,
            main = paste("Differentially expressed genes", colnames(cont.matrix1), sep = "\n"))
abline(v = c(-1, 1))
dev.off()
```

```
## pdf
##   2
```

```
selectedRows <- rownames(filteredData) %in% rownames(topTab)
selectedData <- filteredData[selectedRows,]

#HEATMAP PLOT
my_palette <- colorRampPalette(c("blue", "red"))(n = 2690)
library(gplots)
heatmap.2(selectedData,
          Rowv=TRUE,
          Colv=TRUE,
          main="HeatMap Ethanol.vs.control FC>=1",
          scale="row",
          col=my_palette,
          sepcolor="white",
          key=TRUE,
          keysize=1.5,
          density.info="histogram",
          tracecol=NULL,
          srtCol=30)
```




```
pdf(file.path(resultsDir,"Heatmap.pdf"))
heatmap.2(selectedData,
          Rowv=TRUE,
          Colv=TRUE,
          main="HeatMap Induced.vs.WT FC>=3",
          scale="row",
```

```
          col=my_palette,
          sepcolor="white",
          sepwidth=c(0.05,0.05),
          cexRow=0.5,
          cexCol=0.9,
          key=TRUE,
          keysize=1.5,
          density.info="histogram",
          tracecol=NULL,
          srtCol=30)
dev.off()
```

```
## pdf
##   2
```

```
library(rae230a.db)
probesUniverse <- rownames(filteredData)
entrezUniverse<- AnnotationDbi::select(rae230a.db, probesUniverse, "ENTREZID")$ENTREZID
```

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

```
topProbes <-   rownames(selectedData)
entrezTop<- AnnotationDbi::select(rae230a.db, topProbes, "ENTREZID")$ENTREZID
```

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

```
# Eliminamos posibles duplicados

topGenes <- entrezTop[!duplicated(entrezTop)]
entrezUniverse <- entrezUniverse[!duplicated(entrezUniverse)]
```

```
library(GOstats)

# This parameter has an "ontology" argument. It may be "BP", "MF" or "CC"
# Other arguments are taken by default. Check the help for more information.

GOparams = new("GOHyperGParams",
    geneIds=topGenes, universeGeneIds=entrezUniverse,
    annotation="rae230a.db", ontology="BP",
    pvalueCutoff=0.01)
```

```
GOhyper = hyperGTest(GOparams)
```

```
head(summary(GOhyper))
```

```
##       GOBPID       Pvalue  OddsRatio   ExpCount Count Size
## 1 GO:0044278 1.125900e-06        Inf 0.03241297     3    3
## 2 GO:0035821 1.109721e-05 154.375000 0.05402161     3    5
## 3 GO:0034614 1.122031e-04  13.149351 0.50780312     5   47
## 4 GO:0031953 1.124549e-04        Inf 0.02160864     2    2
```

```
## 5 GO:0042542 1.515707e-04  12.257576 0.54021609     5   50
## 6 GO:0071310 1.519131e-04   4.482839 5.98559424    15  554
##                                                 Term
## 1           cell wall disruption in another organism
## 2          modulation of process of another organism
## 3       cellular response to reactive oxygen species
## 4 negative regulation of protein autophosphorylation
## 5                       response to hydrogen peroxide
## 6             cellular response to organic substance
```

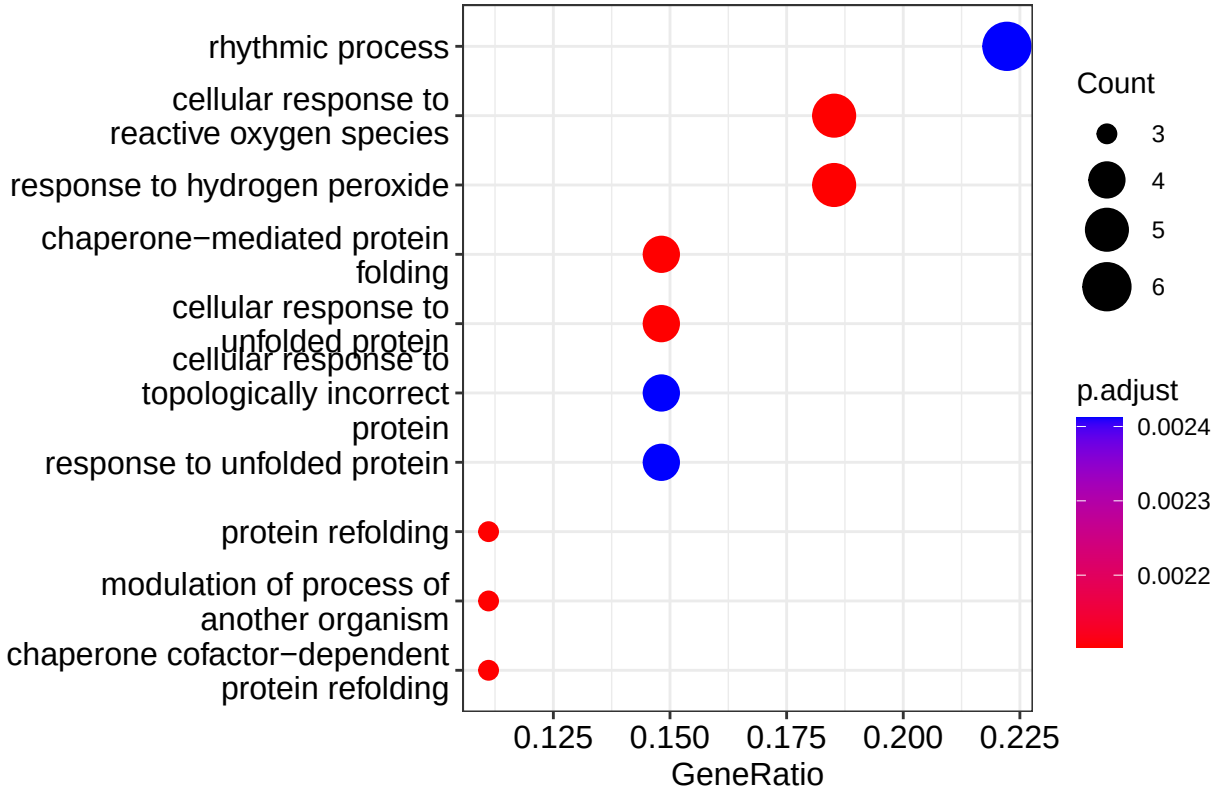
```
dim(summary(GOhyper))
```

```
## [1] 162   7
```

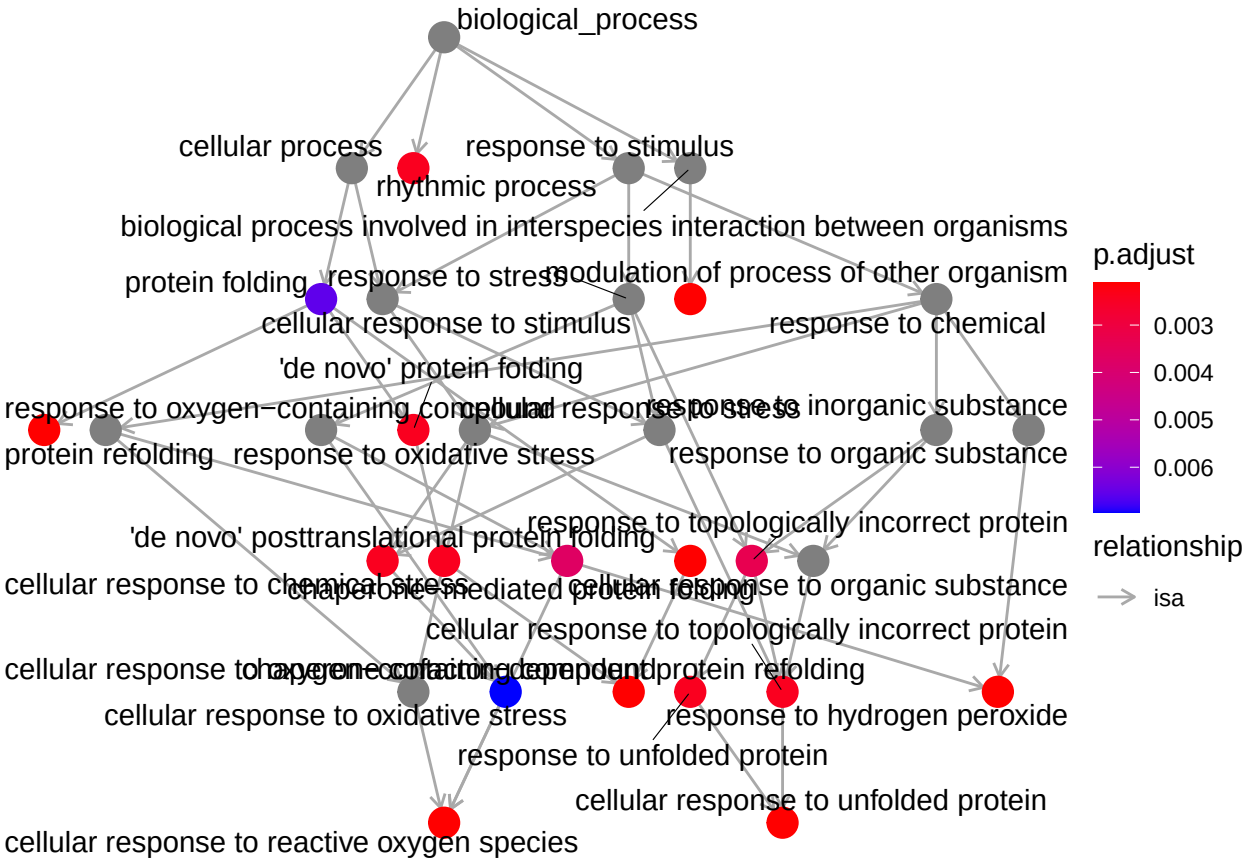
```
# Creamos un informe html con los resultados
GOfilename =file.path(resultsDir, "GOResults.html")
htmlReport(GOhyper, file = GOfilename, summary.args=list("htmlLinks"=TRUE))
```

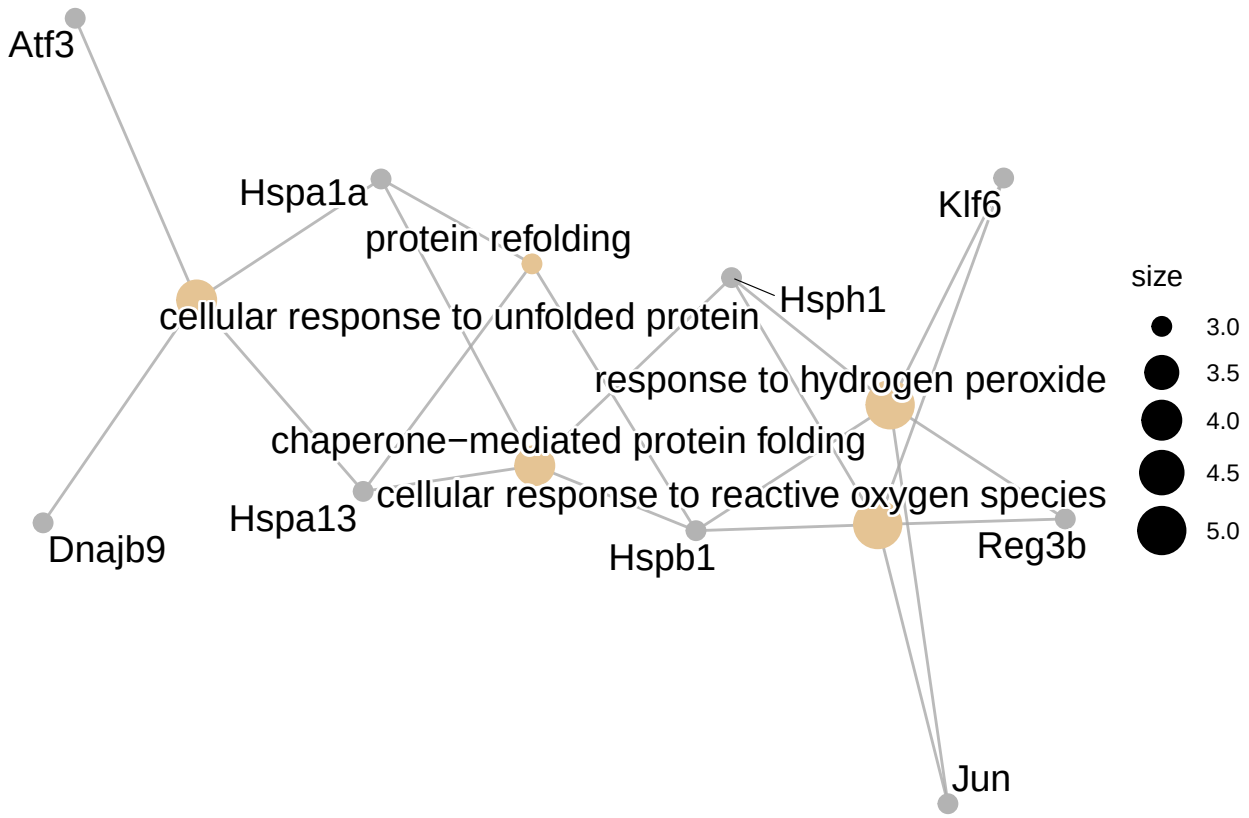
```
library(clusterProfiler)
ego <- enrichGO(gene = as.integer(topGenes), # selgenes,
                # universe = entrezUniverse, # universe = all_genes,
                keyType = "ENTREZID",
                OrgDb = rae230a.db,
                ont = "BP",
                pAdjustMethod = "BH",
                qvalueCutoff = 0.25,
                readable = TRUE)
```

```
dotplot(ego, showCategory=10)
```

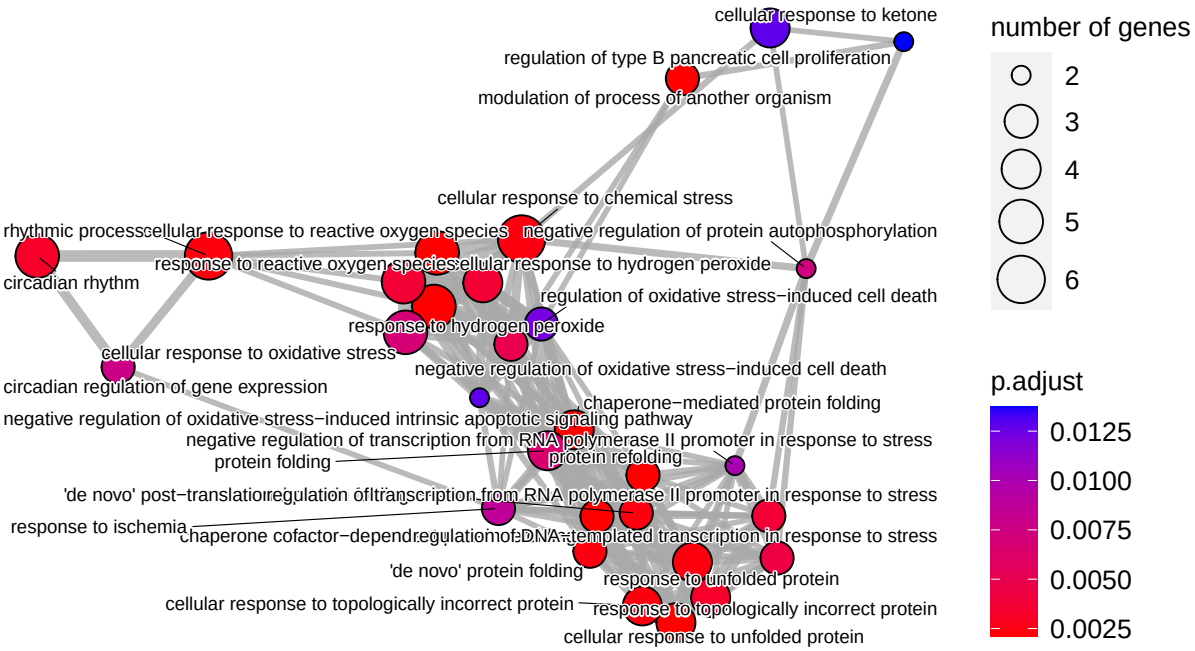



```
goplot(ego, showCategory=10)
```


biological_process
cellular process
response to stimulus
rhythmic process
biological process involved in interspecies interaction between organisms
protein folding
response to stress
modulation of process of other organism
cellular response to stimulus
response to chemical
'de novo' protein folding
response to oxygen−containing compound
response to inorganic substance
protein refolding
response to oxidative stress
response to organic substance
response to topologically incorrect protein
'de novo' posttranslational protein folding
cellular response to chemical stress
chaperone−mediated protein folding
cellular response to organic substance
cellular response to topologically incorrect protein
cellular response to oxygen−containing compound
oxygen−dependent protein refolding
cellular response to oxidative stress
response to hydrogen peroxide
response to unfolded protein
cellular response to unfolded protein
cellular response to reactive oxygen species
p.adjust
0.003
0.004
0.005
0.006
relationship
isa




```
## Warning in emapplot.enrichResult(x, showCategory = showCategory, ...): Use 'cex.params = list(categor
##  The cex_label_category parameter will be removed in the next version.
```


cellular response to ketone
regulation of type B pancreatic cell proliferation
modulation of process of another organism
cellular response to chemical stress
rhythmic process
cellular response to reactive oxygen species
negative regulation of protein autophosphorylation
response to reactive oxygen species
cellular response to hydrogen peroxide
circadian rhythm
regulation of oxidative stress−induced cell death
response to hydrogen peroxide
cellular response to oxidative stress
negative regulation of oxidative stress−induced cell death
circadian regulation of gene expression
chaperone−mediated protein folding
negative regulation of oxidative stress−induced intrinsic apoptotic signaling pathway
negative regulation of transcription from RNA polymerase II promoter in response to stress
protein folding
protein refolding
'de novo' post−translational protein folding
regulation of transcription from RNA polymerase II promoter in response to stress
response to ischemia
chaperone cofactor−dependent protein refolding
regulation of DNA−templated transcription in response to stress
'de novo' protein folding
response to unfolded protein
cellular response to topologically incorrect protein
response to topologically incorrect protein
cellular response to unfolded protein
number of genes
2
3
4
5
6
p.adjust
0.0125
0.0100
0.0075
0.0050
0.0025