PEC1\_JGuitart\_GSE55662

Judith Guitart

10/4/2020

### A1. Identificación de las muestras y los grupos

Se ha escogido el estudio GSE55662 y se han descargado los datos y guardado en una carpeta en la asignatura con el nombre de ‘PEC1’. .

En este directorio vamos a crear carpetas para los análisis, los datos y los resultados.

dir.create("MicroArrayAnalisis")

## Warning in dir.create("MicroArrayAnalisis"): 'MicroArrayAnalisis' already exists

dir.create("Datos")

## Warning in dir.create("Datos"): 'Datos' already exists

dir.create("Resultados")

## Warning in dir.create("Resultados"): 'Resultados' already exists

Guardamos los archivos .CEL descargados de la base de datos ‘Gene Expression Omnibus’ (GEO) en la nueva carpeta ‘Datos’.

A continuación, pasamos a crear y leer el archivo *‘targets’*, que nos servirá para retener toda la información relevante para el análisis de los archivos .CEL individuales en un solo arhivo.

targets.pec <- read.csv2("./Datos/targets.csv", header = TRUE, sep = ";")  
knitr::kable(targets.pec, booktabs = TRUE, caption = "Contenido del archivo targets usado para este análisis")

Contenido del archivo targets usado para este análisis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FileName | Group | Genotype | Treatment | ShortName |
| GSM1341344\_01\_E\_coli\_2\_ | WT\_ntCPR | WT | Not-treated | wt\_nt.1 |
| GSM1341345\_02\_E\_coli\_2\_ | WT\_ntCPR | WT | Not-treated | wt\_nt.2 |
| GSM1341346\_03\_E\_coli\_2\_ | WT\_CPR | WT | CPR | wt\_cpr.1 |
| GSM1341347\_04\_E\_coli\_2\_ | WT\_CPR | WT | CPR | wt\_cpr.2 |
| GSM1341348\_07\_E\_coli\_2\_ | FUR\_ntCPR | fur | Not-treated | fur\_nt.1 |
| GSM1341349\_08\_E\_coli\_2\_ | FUR\_ntCPR | fur | Not-treated | fur\_nt.2 |
| GSM1341350\_09\_E\_coli\_2\_ | FUR\_CPR | fur | CPR | fur\_cpr.1 |
| GSM1341351\_10\_E\_coli\_2\_ | FUR\_CPR | fur | CPR | fur\_cpr.2 |

Finalmente, procedemos a leer los archivos .CEL con la función *list.celfiles* del paquete *oligo*:

library(oligo)

## 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: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

## Loading required package: oligoClasses

## Welcome to oligoClasses version 1.48.0

## 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: Biostrings

## Loading required package: S4Vectors

## Warning: package 'S4Vectors' was built under R version 3.6.2

## Loading required package: stats4

##   
## Attaching package: 'S4Vectors'

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

## Loading required package: IRanges

## Warning: package 'IRanges' was built under R version 3.6.2

##   
## Attaching package: 'IRanges'

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

## Loading required package: XVector

##   
## Attaching package: 'Biostrings'

## The following object is masked from 'package:base':  
##   
## strsplit

## No methods found in package 'RSQLite' for request: 'dbListFields' when loading 'oligo'

## ================================================================================

## Welcome to oligo version 1.50.0

## ================================================================================

archivosCEL <- list.celfiles("./Datos", full.names = TRUE)

A continuación, guardamos el archivo ‘tagets.csv’ generado anteriormente en una nueva variable para asociar la información de los archivos .CEL al mismo. Utilizamos la función *read.AnnotatedDataFrame* del paquete *Biobase*.

library(Biobase)  
mis.targets <- read.AnnotatedDataFrame(file.path("./Datos", "targets.csv"), header = TRUE, row.names = 1, sep=";")  
rawData <- read.celfiles(archivosCEL, phenoData = mis.targets)

## Loading required package: pd.e.coli.2

## Loading required package: RSQLite

## Loading required package: DBI

## Platform design info loaded.

## Reading in : ./Datos/GSM1341344\_01\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341345\_02\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341346\_03\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341347\_04\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341348\_07\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341349\_08\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341350\_09\_E\_coli\_2\_.CEL  
## Reading in : ./Datos/GSM1341351\_10\_E\_coli\_2\_.CEL

## Warning in read.celfiles(archivosCEL, phenoData = mis.targets): 'channel'  
## automatically added to varMetadata in phenoData.

Así, obtenemos el *ExpressionSet* y podemos cambiar el nombre largo de los archivos .CEL por el nombre corto establecido en el archivo ‘targets.csv’:

mis.targets@data$ShortName -> rownames(pData(rawData))  
colnames(rawData) <- rownames(pData(rawData))  
head(rawData)

## ExpressionFeatureSet (storageMode: lockedEnvironment)  
## assayData: 1 features, 8 samples   
## element names: exprs   
## protocolData  
## rowNames: wt\_nt.1 wt\_nt.2 ... fur\_cpr.2 (8 total)  
## varLabels: exprs dates  
## varMetadata: labelDescription channel  
## phenoData  
## rowNames: wt\_nt.1 wt\_nt.2 ... fur\_cpr.2 (8 total)  
## varLabels: Group Genotype Treatment ShortName  
## varMetadata: labelDescription channel  
## featureData: none  
## experimentData: use 'experimentData(object)'  
## Annotation: pd.e.coli.2

Por lo tanto, de las 8 muestras, existen 4 grupos distintos con dos réplicas cada uno, en el siguiente orden: WT sin tratamiento, WT tratado con ciprofloxacina, -fur sin tratamiento, -fur tratado con ciprofloxacina.

### A2. Control de calidad de los datos crudos

Una vez hemos cargado los datos crudos con el nombre ‘rawData’ podemos pasar a determinar la calidad de estos datos antes de normalizar los datos.

library(arrayQualityMetrics)

A continuación observamos el archivo *index.html* creado con un resumen del análisis: **(Afegir imatge index.html de la carpeta)**

Observamos que todas las muestras han sido marcadas almenos alguna vez y la muestra 6 dos veces.

Podemos realizar este análisis también mediante un análisis de componentes principales.

library(ggplot2)

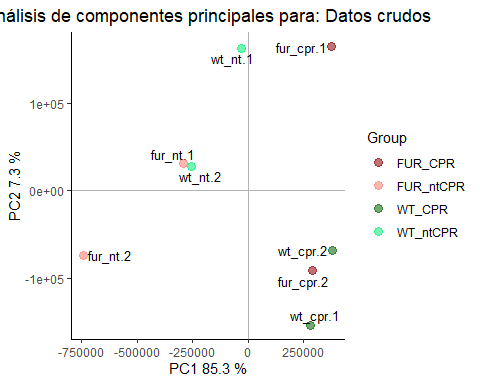
## Warning: package 'ggplot2' was built under R version 3.6.3

library(ggrepel)

## Warning: package 'ggrepel' was built under R version 3.6.3

plotPCA3 <- function (datos, labels, factor, title, scale, colores, size = 1.5, glineas = 0.25) {  
 data <- prcomp(t(datos),scale=scale)  
 dataDf <- data.frame(data$x)  
 Group <- factor  
 loads <- round(data$sdev^2/sum(data$sdev^2)\*100,1)  
 p1 <- ggplot(dataDf,aes(x=PC1, y=PC2)) +  
 theme\_classic() +  
 geom\_hline(yintercept = 0, color = "gray70") +  
 geom\_vline(xintercept = 0, color = "gray70") +  
 geom\_point(aes(color = Group), alpha = 0.55, size = 3) +  
 coord\_cartesian(xlim = c(min(data$x[,1])-5,max(data$x[,1])+5)) +  
 scale\_fill\_discrete(name = "Group")  
 p1 + geom\_text\_repel(aes(y = PC2 + 0.25, label = labels),segment.size = 0.25, size = size) +   
 labs(x = c(paste("PC1",loads[1],"%")),y=c(paste("PC2",loads[2],"%"))) +   
 ggtitle(paste("Análisis de componentes principales para:",title,sep=" "))+   
 theme(plot.title = element\_text(hjust = 0.5)) +  
 scale\_color\_manual(values=colores)  
}

plotPCA3(exprs(rawData), labels = targets.pec$ShortName, factor = targets.pec$Group,   
 title="Datos crudos", scale = FALSE, size = 3.5,   
 colores = c("darkred", "salmon", "darkgreen", "springgreen2"))



Cada color pertenece a cada uno de los cuatro grupos del estudio.

Observamos que el análisis de componentes principales cuenta con un 85.3% del total de la variabilidad de las muestras. Seguramente, esta variabilidad es contribuida principalmente por la condición del tratamiento, ya que observamos tanto la cepa salvaje como la mutante no tratadas a la izquierda y las tratadas con el antibiótico a la derecha.

Podemos guardar este gráfico en el formato *.tiff*, creando una nueva carpeta en el directorio que vamos a nombrar *‘Figuras’*:

dir.create("Figuras")

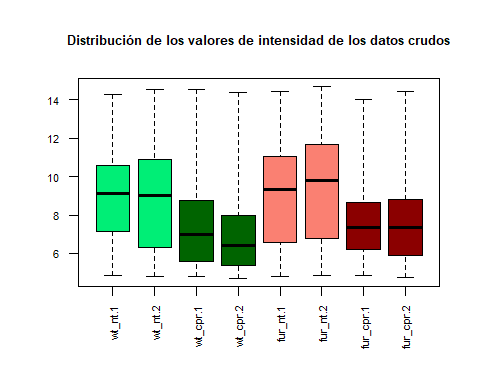
## Warning in dir.create("Figuras"): 'Figuras' already exists

tiff("./Figuras/PCA\_RawData.tiff", res = 200, width = 4.5, height = 4, units = 'in')  
plotPCA3(exprs(rawData), labels = targets.pec$ShortName, factor = targets.pec$Group,   
 title="Raw data", scale = FALSE, size = 2,   
 colores = c("darkred", "salmon", "darkgreen", "springgreen2"))  
dev.off()

## png   
## 2

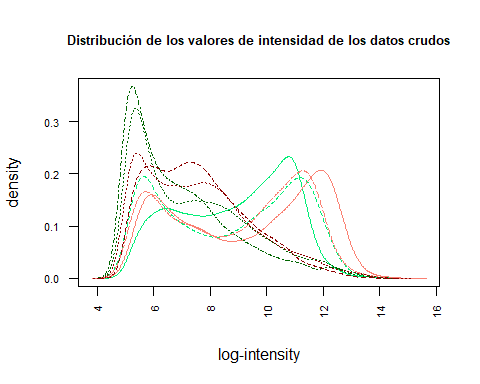
También podemos visualizar la distribución de la intensidad de los *arrays* con diagramas de cajas:

boxplot(rawData, cex.axis=0.7, las=2, which="all", col = c(rep("springgreen2", 2), rep("darkgreen", 2), rep("salmon", 2), rep("darkred", 2)), main = "Distribución de los valores de intensidad de los datos crudos", cex.main = 0.8)

 Se observa una gran variabilidad en la intensidad de los arrays, lo que ya es esperable en los datos crudos.

**Opcional**: Exploración y visualización

hist(rawData, cex.axis=0.7, las=2, which="all", col = c(rep("springgreen2", 2), rep("darkgreen", 2), rep("salmon", 2), rep("darkred", 2)), main = "Distribución de los valores de intensidad de los datos crudos", cex.main = 0.8)



Para observar si las muestras se agrupan por condiciones experimentales:

*Función dendograma falta*

### 3. Normalización

Para hacer los arrays comparables entre ellos y así, reducir al máximo la variabilidad en las muestras que no es debida a razones biológicas, realizaremos un proceso de normalización. Refleja la expresión diferencial de los genes, que no es debida a efectos técnicos. Consiste de tres pasos: corrección del ruido de fondo o *background*, normalización y sumarización.

Usaremos el método Robust Multichip Analysis (RMA), (explicar una mica):

eset\_rma.pec <- rma(rawData)

## Background correcting  
## Normalizing  
## Calculating Expression

Implica una normalización por cuantiles que fuerza la homogeneidad de los arrays.

### 4. Control de calidad de los datos normalizados

Para saber como después de este proceso se observa la calidad de los datos, realizaremos un nuevo control de los mismos:

arrayQualityMetrics(eset\_rma.pec, outdir = file.path("./Resultados", "QCDir.Norm"), force = TRUE)

## The report will be written into directory './Resultados/QCDir.Norm'.

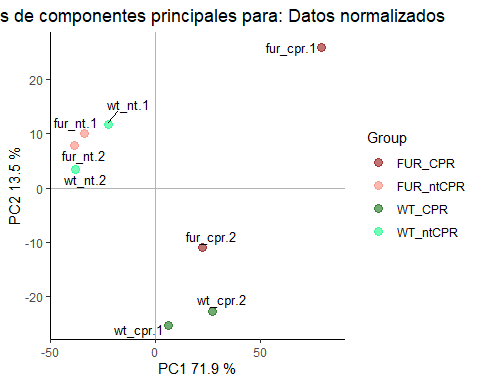
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value  
  
## Warning in svgStyleAttributes(style, svgdev): Removing non-SVG style attribute  
## name(s): subscripts, group.number, group.value

## (loaded the KernSmooth namespace)

Ahora observamos que tan solo una de las muestras está marcada, lo que significa que los problemas potenciales seran pequeños y podemos continuar con el análisis de este array.

Realizamos también PCA de los datos normalizados:

plotPCA3(exprs(eset\_rma.pec), labels = targets.pec$ShortName, factor = targets.pec$Group,   
 title="Datos normalizados", scale = FALSE, size = 3.5,   
 colores = c("darkred", "salmon", "darkgreen", "springgreen"))

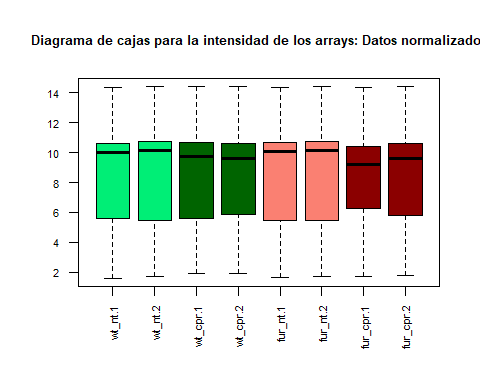


Seguimos viendo la separación de ambos grupos y ahora también la aglomeración de los dos grupos no tratados en un mismo componente.

Ahora cuenta con un 71.9% de la variabilidad total. Es importante destacar una de las muestras que se encuentra lejos de la otra réplica, lo que puede ser debido a algun outlier.

Ahora podemos también observar la distribución de las intensidades con los datos normalizados, que en este caso, las cajas del diagrama se espera que sean muy similares.

boxplot(eset\_rma.pec, cex.axis=0.7, las=2, which="all",   
 col = c(rep("springgreen2", 2), rep("darkgreen", 2), rep("salmon", 2), rep("darkred", 2)),  
 main="Diagrama de cajas para la intensidad de los arrays: Datos normalizados", cex.main = 0.8)



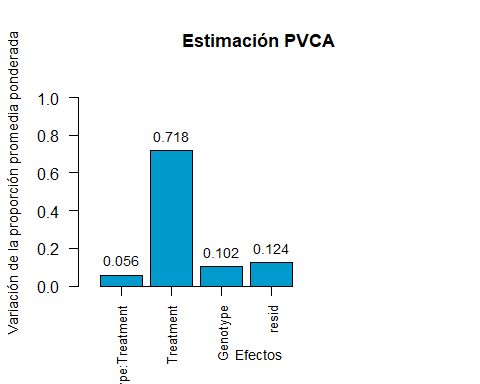
Ahora podemos observar que las intensidades de los arrays son mucho más parecidas.

Para identificar y eliminar los efectos de los distintos lotes, realizaremos un Análisis de componente de variación combate y principal (PVCA):

library(pvca)  
pData(eset\_rma.pec) <- targets.pec  
pct\_threshold <- 0.6  
batch.factors <- c("Genotype", "Treatment")  
pvcaObj <- pvcaBatchAssess(eset\_rma.pec, batch.factors, pct\_threshold)

## boundary (singular) fit: see ?isSingular  
## boundary (singular) fit: see ?isSingular  
## boundary (singular) fit: see ?isSingular

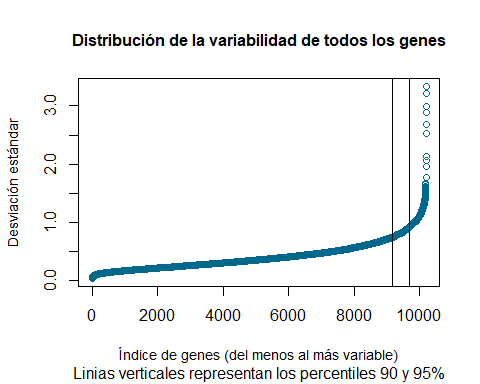
bp <- barplot(pvcaObj$dat, xlab = "Efectos", ylab = "Variación de la proporción promedia ponderada", ylim = c(0,1.1), col=c("deepskyblue3"), las=2, main="Estimación PVCA", xlim = c(0,8), cex.main = 1.1, cex.lab = 0.9)  
axis(1, at = bp, labels = pvcaObj$label, cex.axis = 0.8, las=2)  
values = pvcaObj$dat  
new\_values = round(values, 3)  
text(bp, pvcaObj$dat, labels = new\_values, pos =3, cex = 0.9)



Este gráfico muestra que la mayor fuente de variación es la condición de tratamiento, lo que ya observamos previamente en los análisis de PCA.

### A5. Filtraje no específico

sds <- apply(exprs(eset\_rma.pec), 1, sd)  
sds0 <- sort(sds)  
plot(1:length(sds0), sds0, main="Distribución de la variabilidad de todos los genes", sub="Linias verticales representan los percentiles 90 y 95%", xlab= "Índice de genes (del menos al más variable)", ylab="Desviación estándar", col ="deepskyblue4", cex.main = 1, cex.lab = 0.9)  
abline(v=length(sds)\*c(0.9,0.95))



Filtraje de los genes menos variables, aquellos que pueden atribuirse a variabilidad aleatoria, que no se consideran diferencialmente expresados, lo que permite reducir el nombre de tests, incrementando así la potencia.

Basándonos en un lindar de variabilidad podemos eliminar estos genes con la función *nsFilter* del paquete *genefitler*: Permite eliminar los genes que varían poco entre condiciones o que deseamos quitar por otras razones, con la función *nsFilter*. Escogemos el lindar 0.5 default.

library(genefilter)  
BiocManager::install("ecoli2.db")

## Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.1 (2019-07-05)

## Installing package(s) 'ecoli2.db'

## installing the source package 'ecoli2.db'

## Installation path not writeable, unable to update packages: class

## Old packages: 'backports', 'broom', 'dbplyr', 'DelayedArray', 'dplyr', 'ff',  
## 'glue', 'lattice', 'purrr', 'raster', 'Rcpp', 'RCurl', 'reshape2',  
## 'S4Vectors', 'sparklyr', 'survival', 'tibble'

BiocManager::install("AnnotationDbi")

## Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.1 (2019-07-05)

## Installing package(s) 'AnnotationDbi'

## package 'AnnotationDbi' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Judith\AppData\Local\Temp\RtmpSCAd53\downloaded\_packages

## Installation path not writeable, unable to update packages: class

## Old packages: 'backports', 'broom', 'dbplyr', 'DelayedArray', 'dplyr', 'ff',  
## 'glue', 'lattice', 'purrr', 'raster', 'Rcpp', 'RCurl', 'reshape2',  
## 'S4Vectors', 'sparklyr', 'survival', 'tibble'

BiocManager::install("org.EcK12.eg.db")

## Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.1 (2019-07-05)

## Installing package(s) 'org.EcK12.eg.db'

## installing the source package 'org.EcK12.eg.db'

## Installation path not writeable, unable to update packages: class

## Old packages: 'backports', 'broom', 'dbplyr', 'DelayedArray', 'dplyr', 'ff',  
## 'glue', 'lattice', 'purrr', 'raster', 'Rcpp', 'RCurl', 'reshape2',  
## 'S4Vectors', 'sparklyr', 'survival', 'tibble'

library(ecoli2.db)

## Loading required package: AnnotationDbi

## Loading required package: org.EcK12.eg.db

##

##   
## ecoli2.db is providing annotations for only one of the species that are  
## supported by this platform. You may want to get other annotations  
## from other sources/packages in order to cover all the species that  
## are represented by probes on this platform.

annotation(eset\_rma.pec) <- "ecoli2.db"  
filtered.pec <- nsFilter(eset\_rma.pec, require.entrez = TRUE, remove.dupEntrez = TRUE, var.filter = TRUE, var.func = IQR, var.cutoff = 0.5, filterByQuantile = TRUE, feature.exclude = "^AFFX")  
print(filtered.pec$filter.log)

## $numDupsRemoved  
## [1] 179  
##   
## $numLowVar  
## [1] 2066  
##   
## $numRemoved.ENTREZID  
## [1] 5897

eset\_filtered.pec <- filtered.pec$eset  
eset\_filtered.pec

## ExpressionSet (storageMode: lockedEnvironment)  
## assayData: 2066 features, 8 samples   
## element names: exprs   
## protocolData  
## rowNames: wt\_nt.1 wt\_nt.2 ... fur\_cpr.2 (8 total)  
## varLabels: exprs dates  
## varMetadata: labelDescription channel  
## phenoData  
## rowNames: 1 2 ... 8 (8 total)  
## varLabels: FileName Group ... ShortName (5 total)  
## varMetadata: labelDescription channel  
## featureData: none  
## experimentData: use 'experimentData(object)'  
## Annotation: ecoli2.db

Después del filtraje observamos que nos quedamos con 2066 genes y que hemos guardado estos en la variable *eset\_filtered.pec*.

Procedemos a guardar los datos normalizados y filtrados:

write.csv(exprs(eset\_rma.pec), file="./Resultados/Datos.normalizados.csv")  
write.csv(exprs(eset\_filtered.pec), file="./Resultados/Datos.normalizados.filtrados.csv")  
save(eset\_rma.pec, eset\_filtered.pec, file="./Resultados/Datos.normalizados.Rda")

### A6. Identificación de genes diferencialmente expresados

Ahora podemos proceder a generar la matriz de diseño para seleccionar los genes diferencialmente expresados para compara la expresión génica entre grupos. Usaremos el modelo de Smyth del paquete *limma*, como se ha estudiado en los recursos de la asignatura.

**La matriz de diseño es una tabla que describe cada muestra a un grupo o condición experimental:**

if(!exists("eset\_filtered.pec")) load(file="./Resultados/Datos.normalizados.Rda")

library(limma)

##   
## Attaching package: 'limma'

## The following object is masked from 'package:oligo':  
##   
## backgroundCorrect

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

Matdesign <- model.matrix(~0+Group, pData(eset\_filtered.pec))  
colnames(Matdesign) <- c("WT\_ntCPR", "WT\_CPR", "FUR\_ntCPR", "FUR\_CPR")  
print(Matdesign)

## WT\_ntCPR WT\_CPR FUR\_ntCPR FUR\_CPR  
## 1 0 0 0 1  
## 2 0 0 0 1  
## 3 0 0 1 0  
## 4 0 0 1 0  
## 5 0 1 0 0  
## 6 0 1 0 0  
## 7 1 0 0 0  
## 8 1 0 0 0  
## attr(,"assign")  
## [1] 1 1 1 1  
## attr(,"contrasts")  
## attr(,"contrasts")$Group  
## [1] "contr.treatment"

**La matriz de contrastes permite realizar comparaciones entre los parámetros del modelo.**

(Preguntas de estudio)

1. Efectos del tratamiento en la cepa salvaje.
2. Efectos de la deleción del gen fur.
3. Efectos del tratamiento entre la cepa salvaje y la cepa mutante.

Matcontrast <- makeContrasts(nt.WTvsFUR=FUR\_ntCPR-WT\_ntCPR, WT.ntvsCPR=WT\_CPR-WT\_ntCPR, FUR.ntvsCPR=FUR\_CPR-FUR\_ntCPR, levels = Matdesign)  
print(Matcontrast)

## Contrasts  
## Levels nt.WTvsFUR WT.ntvsCPR FUR.ntvsCPR  
## WT\_ntCPR -1 -1 0  
## WT\_CPR 0 1 0  
## FUR\_ntCPR 1 0 -1  
## FUR\_CPR 0 0 1

**Estimación del modelo y selección de genes:**

Ahora podemos proceder a estimar el modelo y los contrastes y generar los test de significancia para determinar si los genes estan diferencialmente expresados o no.

El paquete *limma* utiliza los modelos de Bayes empíricos determinados como fold-change, aunque para controlar el porcentaje de falsos positivos, podemos ajustar los p-valores utilizando el método de Benjamini y Hochberg.

library(limma)  
fit.pec <- lmFit(eset\_filtered.pec, Matdesign)  
fit.main.pec <- contrasts.fit(fit.pec, Matcontrast)  
fit.main.pec <- eBayes(fit.main.pec)  
class(fit.main.pec)

## [1] "MArrayLM"  
## attr(,"package")  
## [1] "limma"

Ahora ya podemos obtener las listas de los genes diferencialmente expresados con la función *topTable* ordenados por el p-value.

tt\_nt.WTvsFUR <- topTable(fit.main.pec, number=nrow(fit.main.pec), coef = "nt.WTvsFUR", adjust = "fdr")  
head(tt\_nt.WTvsFUR)

## logFC AveExpr t P.Value adj.P.Val B  
## 1766589\_s\_at -1.783351 11.493399 -16.22278 6.155247e-07 0.0005031329 6.860600  
## 1767506\_s\_at -1.967892 11.125541 -15.84490 7.264911e-07 0.0005031329 6.711526  
## 1759831\_at -2.530968 12.090743 -15.83223 7.305899e-07 0.0005031329 6.706442  
## 1765071\_s\_at 5.925855 9.012632 14.61249 1.281955e-06 0.0005754665 6.190534  
## 1765838\_s\_at -1.839593 11.851474 -14.44043 1.392707e-06 0.0005754665 6.113259  
## 1762671\_s\_at -1.771619 11.175903 -13.70120 2.010004e-06 0.0006921115 5.767503

library(xlsx)  
write.xlsx(tt\_nt.WTvsFUR, file = "tt\_nt.WTvsFUR.xlsx")

tt\_WT.ntvsCPR <- topTable(fit.main.pec, number=nrow(fit.main.pec), coef = "WT.ntvsCPR", adjust = "fdr")  
head(tt\_WT.ntvsCPR)

## logFC AveExpr t P.Value adj.P.Val B  
## 1764904\_s\_at -2.478467 12.129216 -17.56172 3.519295e-07 0.0006118159 7.309760  
## 1764659\_s\_at -1.745830 11.663741 -15.51783 8.411388e-07 0.0006118159 6.550895  
## 1763947\_s\_at -1.990319 11.623403 -15.39744 8.884065e-07 0.0006118159 6.501913  
## 1767285\_s\_at -2.086288 11.943222 -14.41766 1.408159e-06 0.0006396866 6.083315  
## 1761763\_s\_at -1.924742 12.104443 -14.22351 1.548128e-06 0.0006396866 5.995927  
## 1763792\_at 1.571757 9.334934 12.33891 4.159624e-06 0.0010005909 5.061069

write.xlsx(tt\_WT.ntvsCPR, file = "tt\_WT.ntvsCPR.xlsx")

tt\_FUR.ntvsCPR <- topTable(fit.main.pec, number=nrow(fit.main.pec), coef="FUR.ntvsCPR", adjust = "fdr")  
head(tt\_FUR.ntvsCPR)

## logFC AveExpr t P.Value adj.P.Val B  
## 1764904\_s\_at -2.544557 12.12922 -18.03002 2.922195e-07 0.0004641344 7.498647  
## 1763947\_s\_at -2.105803 11.62340 -16.29084 5.976549e-07 0.0004641344 6.877074  
## 1761763\_s\_at -2.167163 12.10444 -16.01496 6.739609e-07 0.0004641344 6.769956  
## 1764659\_s\_at -1.662496 11.66374 -14.77711 1.185259e-06 0.0006121863 6.256939  
## 1766450\_s\_at -1.483903 11.05486 -12.75760 3.301221e-06 0.0009813559 5.289097  
## 1760213\_s\_at -1.705236 10.48187 -12.73879 3.335180e-06 0.0009813559 5.279211

write.xlsx(tt\_FUR.ntvsCPR, file = "tt\_FUR.ntvsCPR.xlsx")

La primera columna de cada tabla contiene el ID del array de Affymetrix que cada probeset.

### A7. Anotación de los resultados

Permite asociar los identificadores del apartado anterior con identificadores con nombres familiares, por ejemplo con el Gene Symbol. Creamos una función para facilitar la generación de estas tablas de anotaciones.

annotatedTopTable <- function(topTab, anotPackage)  
{  
 topTab <- cbind(PROBEID=rownames(topTab), topTab)  
 myProbes <- rownames(topTab)  
 thePackage <- eval(parse(text = anotPackage))  
 geneAnots <- select(thePackage, myProbes, c("SYMBOL", "ENTREZID", "GENENAME"))  
 annotatedTopTab<- merge(x=geneAnots, y=topTab, by.x="PROBEID", by.y="PROBEID")  
return(annotatedTopTab)  
}

topAnotados\_nt.WTvsFUR <- annotatedTopTable(tt\_nt.WTvsFUR, anotPackage = "ecoli2.db")

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

topAnotados\_WT.ntvsCPR <- annotatedTopTable(tt\_WT.ntvsCPR, anotPackage = "ecoli2.db")

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

topAnotados\_FUR.ntvsCPR <- annotatedTopTable(tt\_FUR.ntvsCPR, anotPackage = "ecoli2.db")

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

write.csv(topAnotados\_WT.ntvsCPR, file="./Resultados/topAnotados\_WT.ntvsCPR.csv")  
write.csv(topAnotados\_nt.WTvsFUR, file="./Resultados/topAnotados\_nt.WTvsFUR.csv")  
write.csv(topAnotados\_FUR.ntvsCPR, file="./Resultados/topAnotados\_FUR.ntvsCPR.csv")

### A8. Comparación entre distintas comparaciones

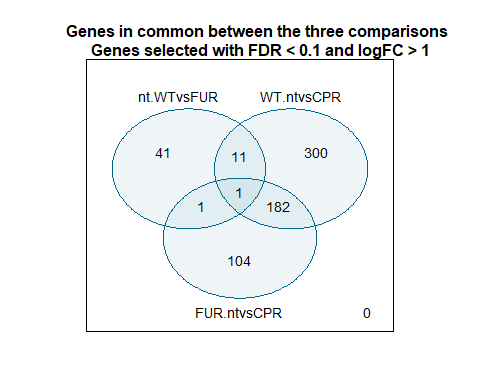
library(limma)  
res.pec <- decideTests(fit.main.pec, method ="separate", adjust.method = "fdr", p.value = 0.1, lfc = 1)

Tabla de anotaciones sencilla:

sum.res.rows.pec <- apply(abs(res.pec),1,sum)  
res.selected.pec <- res.pec[sum.res.rows.pec!=0,]  
print(summary(res.pec))

## nt.WTvsFUR WT.ntvsCPR FUR.ntvsCPR  
## Down 42 70 141  
## NotSig 2012 1572 1778  
## Up 12 424 147

vennDiagram(res.selected.pec[,1:3], cex = 0.9, circle.col = "deepskyblue4")  
title("Genes in common between the three comparisons \n Genes selected with FDR < 0.1 and logFC > 1", cex.main = 1)



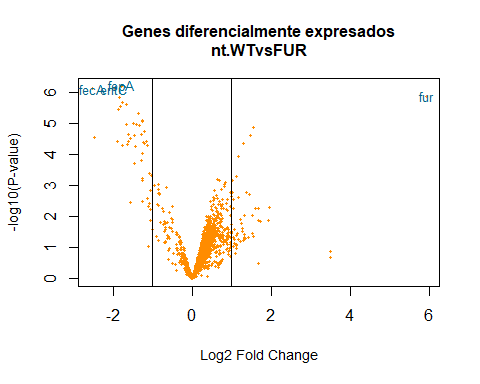
### A9. Visualización de la expresión diferencial con volcano plots

Nos permiten observar si hay muchos o pocos genes con elevado cambio en la expresión y la significación de la expresión.

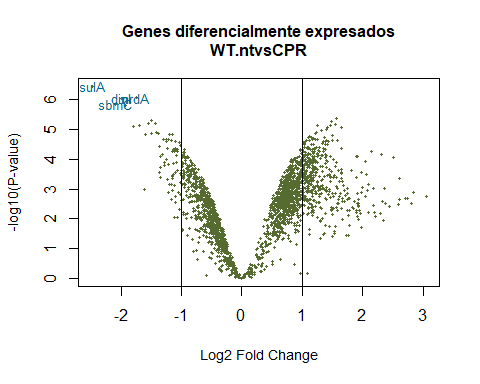
library(ecoli2.db)  
geneSymbols <- select(ecoli2.db, rownames(fit.main.pec), c("SYMBOL"))

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

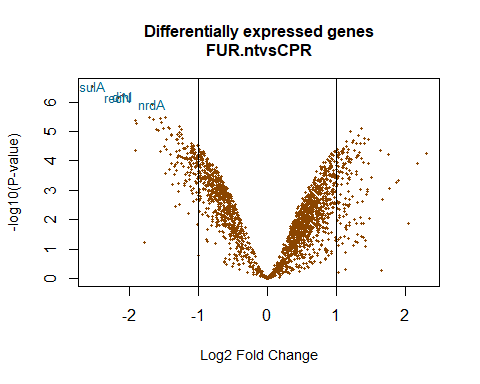
SYMBOLS <- geneSymbols$SYMBOL  
volcanoplot(fit.main.pec, coef=1, highlight=4, names=SYMBOLS, main=paste("Genes diferencialmente expresados", colnames(Matcontrast)[1], sep="\n"), hl.col="deepskyblue4", col=c("darkorange"), cex.main = 1, cex.lab = 0.9)  
abline(v=c(-1,1))



volcanoplot(fit.main.pec, coef=2, highlight=4, names=SYMBOLS, main=paste("Genes diferencialmente expresados", colnames(Matcontrast)[2], sep="\n"), hl.col="deepskyblue4", col=c("darkolivegreen"), cex.main = 1, cex.lab = 0.9)  
abline(v=c(-1,1))



volcanoplot(fit.main.pec, coef=3, highlight=4, names=SYMBOLS, main=paste("Differentially expressed genes", colnames(Matcontrast)[3], sep="\n"), hl.col="deepskyblue4", col=c("darkorange4"), cex.main = 1, cex.lab = 0.9)  
abline(v=c(-1,1))



**Heatmaps:**

probesInHeatmap.pec <- rownames(res.selected.pec)  
DatosHM <- exprs(eset\_filtered.pec)[rownames(exprs(eset\_filtered.pec)) %in% probesInHeatmap.pec,]  
  
geneSymbols <- select(ecoli2.db, rownames(DatosHM), c("SYMBOL"))

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

SYMBOLS<- geneSymbols$SYMBOL  
rownames(DatosHM) <- SYMBOLS  
write.csv(DatosHM, file = file.path("./Resultados/DatosHM.csv"))

Primero sin dendograma

my\_palette <- colorRampPalette(c("lightcyan", "deepskyblue4"))(n = 299)  
library(gplots)

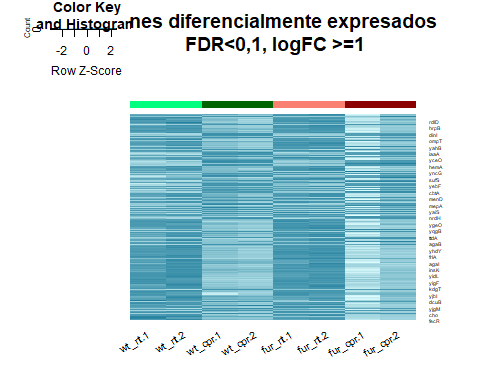
##   
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':  
##   
## space

## The following object is masked from 'package:S4Vectors':  
##   
## space

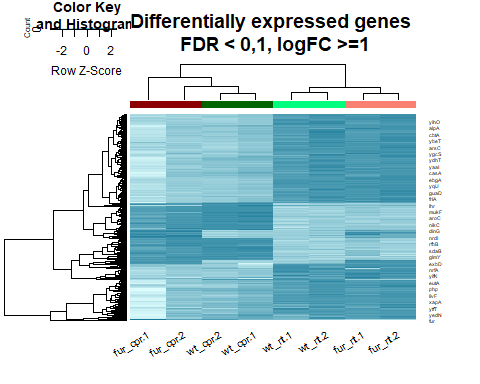
## The following object is masked from 'package:stats':  
##   
## lowess

heatmap.2(DatosHM,  
 Rowv = FALSE,  
 Colv = FALSE,  
 main = "Genes diferencialmente expresados \n FDR<0,1, logFC >=1",   
 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",  
 ColSideColors = c(rep("springgreen",2),rep("darkgreen",2), rep("salmon",2), rep("darkred",2)),  
 tracecol = NULL,  
 dendrogram = "none",  
 srtCol = 30)



Agrupando genes y muestras por su similaridad:

heatmap.2(DatosHM,  
 Rowv = TRUE,  
 Colv = TRUE,  
 dendrogram = "both",  
 main = "Differentially expressed genes \n FDR < 0,1, logFC >=1", cex.main = 0.8,  
 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",  
 ColSideColors = c(rep("springgreen",2),rep("darkgreen",2), rep("salmon",2), rep("darkred",2)),  
 tracecol = NULL,  
 srtCol = 30)



png("heatmap.png")  
heatmap.2(DatosHM,  
 Rowv = TRUE,  
 Colv = TRUE,  
 dendrogram = "both",  
 main = "Differentially expressed genes \n FDR < 0,1, logFC >=1", cex.main = 0.8,  
 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",  
 ColSideColors = c(rep("springgreen",2),rep("darkgreen",2), rep("salmon",2), rep("darkred",2)),  
 tracecol = NULL,  
 srtCol = 30)  
dev.off()

## png   
## 2

### A10. Análisis de significación biológica (“Gene enrichment analysis”)

ListaTablas.pec <- list(nt.WTvsFUR = tt\_nt.WTvsFUR, WT.ntvsCPR = tt\_WT.ntvsCPR, FUR.ntvsCPR = tt\_FUR.ntvsCPR)  
ListaSeleccionados.pec <- list()  
for (i in 1:length(ListaTablas.pec)){  
 # select the toptable  
 topTab <- ListaTablas.pec[[i]]  
 # select the genes to be included in the analysis  
 whichGenes<-topTab["adj.P.Val"]<0.15  
 selectedIDs <- rownames(topTab)[whichGenes]  
 # convert the ID to Entrez  
 EntrezIDs<- select(ecoli2.db, selectedIDs, c("ENTREZID"))  
 EntrezIDs <- EntrezIDs$ENTREZID  
 ListaSeleccionados.pec[[i]] <- EntrezIDs  
 names(ListaSeleccionados.pec)[i] <- names(ListaTablas.pec)[i]  
}

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

sapply(ListaSeleccionados.pec, length)

## nt.WTvsFUR WT.ntvsCPR FUR.ntvsCPR   
## 115 1893 1753

Como es E. coli no podemos utilizar el package Reactome PA, utilizaremos la función enrichGO del paquete *clusterProfiler*.

BiocManager::install("AnnotationDbi")

## Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.1 (2019-07-05)

## Installing package(s) 'AnnotationDbi'

## Installation path not writeable, unable to update packages: class

## Old packages: 'backports', 'broom', 'dbplyr', 'DelayedArray', 'dplyr', 'ff',  
## 'glue', 'lattice', 'purrr', 'raster', 'Rcpp', 'RCurl', 'reshape2',  
## 'S4Vectors', 'sparklyr', 'survival', 'tibble'

BiocManager::install("clusterProfiler")

## Bioconductor version 3.10 (BiocManager 1.30.10), R 3.6.1 (2019-07-05)

## Installing package(s) 'clusterProfiler'

## package 'clusterProfiler' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Judith\AppData\Local\Temp\RtmpSCAd53\downloaded\_packages

## Installation path not writeable, unable to update packages: class

## Old packages: 'backports', 'broom', 'dbplyr', 'DelayedArray', 'dplyr', 'ff',  
## 'glue', 'lattice', 'purrr', 'raster', 'Rcpp', 'RCurl', 'reshape2',  
## 'S4Vectors', 'sparklyr', 'survival', 'tibble'

library(AnnotationDbi)  
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 biological themes among gene clusters. OMICS: A Journal of Integrative Biology. 2012, 16(5):284-287.

library(igraph)

##   
## Attaching package: 'igraph'

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

## The following object is masked from 'package:oligo':  
##   
## normalize

## The following object is masked from 'package:Biostrings':  
##   
## union

## The following object is masked from 'package:IRanges':  
##   
## union

## The following object is masked from 'package:S4Vectors':  
##   
## union

## The following objects are masked from 'package:BiocGenerics':  
##   
## normalize, path, union

## The following objects are masked from 'package:stats':  
##   
## decompose, spectrum

## The following object is masked from 'package:base':  
##   
## union

ListaDatos.pec <- ListaSeleccionados.pec[1:3]  
comparisonsNames <- names(ListaDatos.pec)  
  
for (i in 1:length(ListaDatos.pec)){  
 genesIn <- ListaDatos.pec[[i]]  
 comparison <- comparisonsNames[i]  
 enrich.result <- enrichGO(gene = genesIn, OrgDb = "org.EcK12.eg.db", ont = "ALL", pAdjustMethod ="BH", pvalueCutoff = 0.05, readable = TRUE)  
   
 cat("##################################")  
 cat("\nComparison: ", comparison,"\n")  
 print(head(enrich.result))  
  
 if (length(rownames(enrich.result@result)) != 0) {  
 write.csv(as.data.frame(enrich.result),   
 file =paste0("./Resultados/","Enrich.Results.",comparison,".csv"),   
 row.names = FALSE)  
 write.xlsx(as.data.frame(enrich.result),   
 file =paste0("./Resultados/","Enrich.Results.",comparison,".xlsx"),   
 row.names = FALSE)  
   
 pdf(file=paste0("./Resultados/","Enrich.Barplot.",comparison,".pdf"))  
 print(barplot(enrich.result, showCategory = 15, font.size = 6,   
 title = paste0("EnrichGO Pathway Analysis for ", comparison,". Barplot")))  
 dev.off()  
  
 pdf(file = paste0("./Resultados/","EnrichGOcnetplot.",comparison,".pdf"))  
 print(cnetplot(enrich.result, categorySize = "geneNum", schowCategory = 15,   
 vertex.label.cex = 0.75))  
 dev.off()  
  
 }  
}

## ##################################  
## Comparison: nt.WTvsFUR   
## ONTOLOGY ID Description  
## GO:0009237 BP GO:0009237 siderophore metabolic process  
## GO:0009712 BP GO:0009712 catechol-containing compound metabolic process  
## GO:0018958 BP GO:0018958 phenol-containing compound metabolic process  
## GO:0019184 BP GO:0019184 nonribosomal peptide biosynthetic process  
## GO:0006826 BP GO:0006826 iron ion transport  
## GO:0044550 BP GO:0044550 secondary metabolite biosynthetic process  
## GeneRatio BgRatio pvalue p.adjust qvalue  
## GO:0009237 10/82 10/2754 3.144096e-16 9.935344e-14 8.505608e-14  
## GO:0009712 9/82 10/2754 1.153969e-13 1.823272e-11 1.560896e-11  
## GO:0018958 9/82 11/2754 6.194912e-13 4.893980e-11 4.189717e-11  
## GO:0019184 9/82 11/2754 6.194912e-13 4.893980e-11 4.189717e-11  
## GO:0006826 11/82 20/2754 1.116166e-12 7.054170e-11 6.039047e-11  
## GO:0044550 9/82 12/2754 2.418646e-12 1.273820e-10 1.090512e-10  
## geneID Count  
## GO:0009237 entC/fur/ybdZ/entB/fes/entE/entA/entF/entH/entD 10  
## GO:0009712 entC/ybdZ/entB/fes/entE/entA/entF/entH/entD 9  
## GO:0018958 entC/ybdZ/entB/fes/entE/entA/entF/entH/entD 9  
## GO:0019184 entC/fur/ybdZ/entB/entE/entA/entF/entH/entD 9  
## GO:0006826 fepA/fecA/feoA/exbB/feoB/yoeA/fepB/fepG/fepD/fepE/cirA 11  
## GO:0044550 entC/fur/ybdZ/entB/entE/entA/entF/entH/entD 9

## ##################################  
## Comparison: WT.ntvsCPR   
## ONTOLOGY ID Description GeneRatio  
## GO:0009432 BP GO:0009432 SOS response 19/1197  
## GO:0043711 BP GO:0043711 pilus organization 31/1197  
## GO:0071806 BP GO:0071806 protein transmembrane transport 15/1197  
## GO:0015031 BP GO:0015031 protein transport 23/1197  
## GO:0042886 BP GO:0042886 amide transport 49/1197  
## GO:0045184 BP GO:0045184 establishment of protein localization 27/1197  
## BgRatio pvalue p.adjust qvalue  
## GO:0009432 22/2754 3.911309e-05 0.0233515 0.02283571  
## GO:0043711 42/2754 5.816065e-05 0.0233515 0.02283571  
## GO:0071806 17/2754 1.714647e-04 0.0327898 0.03206554  
## GO:0015031 30/2754 2.115914e-04 0.0327898 0.03206554  
## GO:0042886 77/2754 2.398783e-04 0.0327898 0.03206554  
## GO:0045184 37/2754 2.450048e-04 0.0327898 0.03206554  
## geneID  
## GO:0009432 sulA/dinI/recN/dinG/yebG/umuD/recA/ruvA/cho/polB/umuC/dinB/ruvB/lexA/recX/ssb/rimK/dinD/uvrD  
## GO:0043711 htrE/yadN/yadC/yadV/yraI/yadK/yraJ/yqiG/ycbV/ybgD/ybjN/yhcD/ygiL/ybgQ/sfmA/yadM/elfD/sfmC/sfmD/elfC/yraH/ydeS/yfcV/ycbU/yehC/yfcU/ydeR/ydeT/hofN/sfmF/yehB  
## GO:0071806 gspJ/gspF/gspE/gspI/tatC/secA/tatE/hofC/secE/csgG/tatB/csgE/tatA/secY/secG  
## GO:0015031 gspJ/napD/gspF/gspE/gspI/yhjG/ffh/lolA/tatC/secD/secA/tatE/hofC/secE/csgG/tolA/ftsY/tatB/csgE/asmA/tatA/secY/secG  
## GO:0042886 gspJ/napD/gspF/gspE/gspI/exbB/cydD/yhjG/yejB/mdtO/ffh/mdtN/lolA/mdtP/dtpC/tatC/secD/secA/tatE/hofC/secE/yeeO/csgG/gsiA/tolA/yejE/ydeE/ftsY/dtpA/tolR/tatB/panF/yejF/csgE/bioP/nikD/asmA/tonB/oppB/tatA/secY/dtpB/oppC/gsiB/secG/bcr/dtpD/fiu/yojI  
## GO:0045184 gspJ/napD/gspF/gspE/gspI/bamE/yhjG/ffh/lolA/tatC/secD/secA/tatE/hofC/secE/csgG/tolA/ftsY/tatB/csgE/bamB/asmA/tatA/secY/yidC/secG/bamA  
## Count  
## GO:0009432 19  
## GO:0043711 31  
## GO:0071806 15  
## GO:0015031 23  
## GO:0042886 49  
## GO:0045184 27

## ##################################  
## Comparison: FUR.ntvsCPR   
## ONTOLOGY ID Description GeneRatio  
## GO:0009432 BP GO:0009432 SOS response 19/1136  
## GO:0045184 BP GO:0045184 establishment of protein localization 27/1136  
## GO:0006886 BP GO:0006886 intracellular protein transport 12/1136  
## GO:0034613 BP GO:0034613 cellular protein localization 20/1136  
## GO:0070727 BP GO:0070727 cellular macromolecule localization 20/1136  
## GO:0008104 BP GO:0008104 protein localization 29/1136  
## BgRatio pvalue p.adjust qvalue  
## GO:0009432 22/2754 1.595970e-05 0.01281564 0.01246537  
## GO:0045184 37/2754 8.390117e-05 0.03042732 0.02959569  
## GO:0006886 13/2754 1.894702e-04 0.03042732 0.02959569  
## GO:0034613 26/2754 2.237408e-04 0.03042732 0.02959569  
## GO:0070727 26/2754 2.237408e-04 0.03042732 0.02959569  
## GO:0008104 42/2754 2.273523e-04 0.03042732 0.02959569  
## geneID  
## GO:0009432 sulA/dinI/recN/ruvA/dinG/recA/yebG/umuD/cho/polB/ruvB/rimK/dinB/umuC/lexA/recX/dinD/uvrD/ssb  
## GO:0045184 napD/ffh/secA/secD/bamE/gspJ/gspI/gspF/gspE/lolA/tatC/bamB/tatE/tolA/tatB/secE/ftsY/secY/yhjG/yidC/secG/asmA/hofC/tatA/bamA/csgE/skp  
## GO:0006886 ffh/secA/tatC/tatE/tatB/secE/ftsY/secY/yhjG/secG/asmA/tatA  
## GO:0034613 ffh/secA/bamE/lolB/lolC/lolA/tatC/bamB/tatE/tatB/secE/ftsY/secY/yhjG/yidC/secG/asmA/tatA/bamA/skp  
## GO:0070727 ffh/secA/bamE/lolB/lolC/lolA/tatC/bamB/tatE/tatB/secE/ftsY/secY/yhjG/yidC/secG/asmA/tatA/bamA/skp  
## GO:0008104 napD/ffh/secA/secD/bamE/gspJ/lolB/gspI/gspF/gspE/lolC/lolA/tatC/bamB/tatE/tolA/tatB/secE/ftsY/secY/yhjG/yidC/secG/asmA/hofC/tatA/bamA/csgE/skp  
## Count  
## GO:0009432 19  
## GO:0045184 27  
## GO:0006886 12  
## GO:0034613 20  
## GO:0070727 20  
## GO:0008104 29

cnetplot(enrich.result, categorySize = "geneNum", schowCategory = 15,   
 vertex.label.cex = 0.75)

