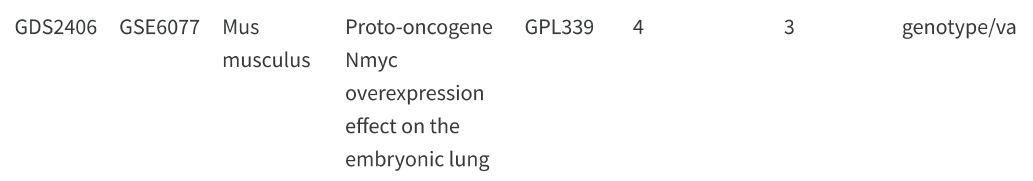
Análisis de datos Ómicos - PAC 2

Maria Lucas

2023-05-14

# Selección y carga del dataset

He seleccionado el set de datos con GEO serie “GSE6077”. Éste es un estudio de sobreexpresión de un proto-oncogen llamado Nmyc.



Primeramente, examinamos la phenoData para poder determinar las variables de interés. Las muestras con las que trabajamos son muestras de pulmón de ratón (Mus Musculus). Se ha realizado una extración de RNA total con RNeasy y un análisis de la expresión con array de Affymetrix (array de un color).

El objeto del estudio es determinar las diferencias de expresión entre dos líneas celulares, una wild type (muestras GSM140827.CEL y GSM140863.CEL) y una con nmyc sobreexpresado (muestras GSM140864.CEL y GSM140865.CEL). Con ésta información podemos crear el objeto targets como un AnnotatedDataFrame con la variable de interés (WT o OE), en mi caso he decido marcarla con dummy coding con 1 para WT y 0 para OE.

# Install and load GEOquery package  
if (!requireNamespace("BiocManager", quietly = TRUE))  
 install.packages("BiocManager")  
BiocManager::install("GEOquery")

## Bioconductor version 3.17 (BiocManager 1.30.20), R 4.3.0 (2023-04-21 ucrt)

## Warning: package(s) not installed when version(s) same as or greater than current; use  
## `force = TRUE` to re-install: 'GEOquery'

## Installation paths not writeable, unable to update packages  
## path: C:/Program Files/R/R-4.3.0/library  
## packages:  
## class, KernSmooth, MASS, nnet

library(GEOquery)

## Loading required package: Biobase

## Loading required package: BiocGenerics

##   
## Attaching package: 'BiocGenerics'

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

## The following objects are masked from 'package:base':  
##   
## anyDuplicated, aperm, 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.max, which.min

## Welcome to Bioconductor  
##   
## Vignettes contain introductory material; view with  
## 'browseVignettes()'. To cite Bioconductor, see  
## 'citation("Biobase")', and for packages 'citation("pkgname")'.

## Setting options('download.file.method.GEOquery'='auto')

## Setting options('GEOquery.inmemory.gpl'=FALSE)

# Set GEO accession number  
gse = "GSE6077"  
  
# Download and load series matrix file and extract phenoData  
gse\_data = getGEO(gse, GSEMatrix = TRUE)

## Found 1 file(s)

## GSE6077\_series\_matrix.txt.gz

pheno = pData(phenoData(gse\_data[[1]]))  
print(pheno[8:12]) # Print some of the more relevant data

## source\_name\_ch1 organism\_ch1  
## GSM140827 whole lung E18 normal Mus musculus  
## GSM140863 whole lung E18 normal Mus musculus  
## GSM140864 whole lung E18 sftpc-nmyc Mus musculus  
## GSM140865 whole lung E18 sftpc-nmyc Mus musculus  
## characteristics\_ch1  
## GSM140827 normal embryonic day 18 lungs  
## GSM140863 normal lung  
## GSM140864 lungs were over expressing nmyc-gfp fusion protein using the lung surfactant c promoter.  
## GSM140865 whole lungs overexpressing an nmyc-gfp fusion protein via the lung surfactant c promoter  
## molecule\_ch1 extract\_protocol\_ch1  
## GSM140827 total RNA RNeasy kit from QIagen  
## GSM140863 total RNA RNeasy kit from QIagen  
## GSM140864 total RNA RNeasy kit from QIagen  
## GSM140865 total RNA RNeasy kit from QIagen

# Install and load fastDummies package  
if (!requireNamespace("fastDummies", quietly = TRUE))  
 install.packages("fastDummies")  
library(fastDummies)  
  
# Create targets object by refining the phenoData  
  
# Create the dataframe of metadata to create the AnnotatedDataFrame  
metadata = data.frame(labelDescription = character(0), stringsAsFactors = FALSE)  
metadata[1, "labelDescription"] <- "Cell line: Wild type as 1, Overexpressed as 0"  
  
# Create the target object as an AnnotatedDataFrame  
targets = AnnotatedDataFrame( data = pheno[8], varMetadata = metadata) # Copy 8th column of phenoData  
colnames(targets) = "WT" # Change name column to WT  
targets$WT[targets$WT == 'whole lung E18 normal'] <- '1' # Normal = 1  
targets$WT[targets$WT == 'whole lung E18 sftpc-nmyc'] <- '0' # Overexpressed = 0  
print(targets)

## An object of class 'AnnotatedDataFrame'  
## rowNames: GSM140827 GSM140863 GSM140864 GSM140865  
## varLabels: WT  
## varMetadata: labelDescription

Ahora que tenemos correctamente creado el objeto targets, podemos pasar a la carga de datos RAW. Descargamos los datos con el número de serie correspondiente buscando en NCBI (<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE6077>). Extraemos los archivos CEL en la carpeta correspondiente y procedemos a la carga de datos RAW mediante el uso del paquete affy.

# BiocManager::install("affy")  
library(affy)  
  
setwd("D:/Antiguos estudios/MASTER2/Sem2/Ómica/PAC2/OMICA-2")  
workingDir = getwd()  
celfilesDir = file.path(workingDir,"celfiles")  
  
filenames = c('GSM140827.CEL', 'GSM140863.CEL', 'GSM140864.CEL', 'GSM140865.CEL')  
rawData = read.affybatch(filenames=file.path(celfilesDir, filenames), phenoData = targets, compress = FALSE)  
  
print(rawData)

## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when  
## loading 'moe430acdf'

## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail' when  
## loading 'moe430acdf'

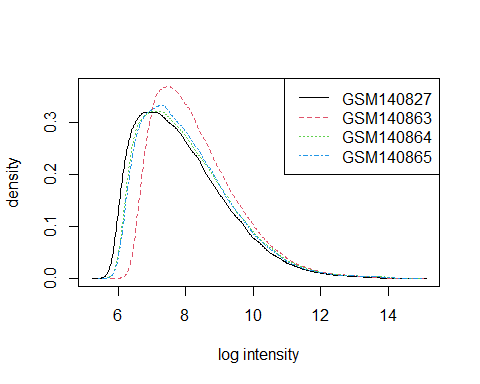
##

## AffyBatch object  
## size of arrays=712x712 features (19 kb)  
## cdf=MOE430A (22690 affyids)  
## number of samples=4  
## number of genes=22690  
## annotation=moe430a  
## notes=

# Control de calidad

### Histograma

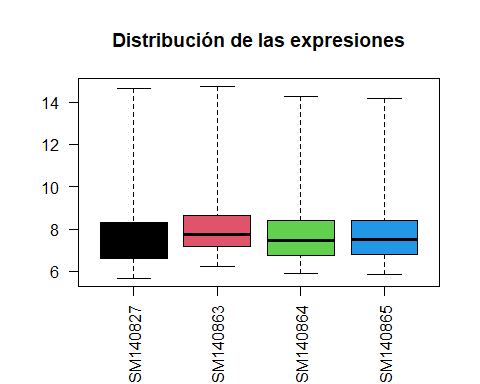
affySampleNames <- rownames(pData(rawData))  
affyColores <- c(1,2,3,4)  
affyLineas <- c(1,2,3,4)  
hist(rawData)  
legend (x="topright", legend=affySampleNames , col=affyColores, lty=affyLineas)



Como podemos ver, los 4 arrays dibujan una distribución de similar forma y posición. Esto sugiere que las expresiones no han sido afectadas por ningún aspecto técnico, y que la ligera variación que vemos en GSM140863 se debe a efectos biológicos.

### Boxplot

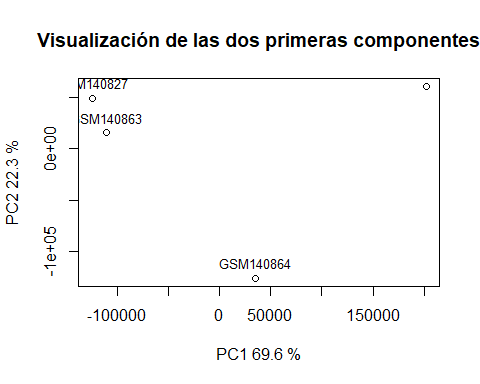
boxplot(rawData, main="Distribución de las expresiones", col=affyColores, las=2)



Nuevamente, el boxplot nos da una idea de la intensidad de la expresión, que es muy similar en todos los arrays.

### Análisis de componentes principales

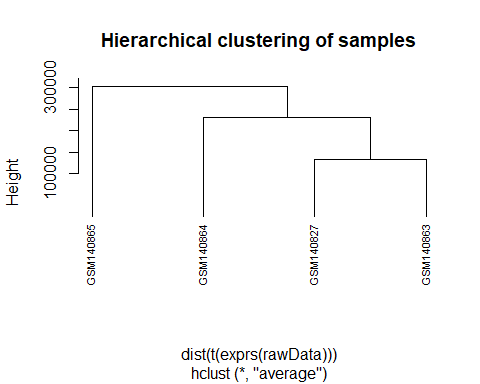
plotPCA <- function ( X, labels=NULL, colors=NULL, dataDesc="", scale=FALSE)  
{  
 pcX<-prcomp(t(X), scale=scale) # o prcomp(t(X))  
 loads<- round(pcX$sdev^2/sum(pcX$sdev^2)\*100,1)  
 xlab<-c(paste("PC1",loads[1],"%"))  
 ylab<-c(paste("PC2",loads[2],"%"))  
 if (is.null(colors)) colors=1  
 plot(pcX$x[,1:2],xlab=xlab,ylab=ylab, col=colors,   
 xlim=c(min(pcX$x[,1])-10, max(pcX$x[,1])+10),  
 ylim=c(min(pcX$x[,2])-10, max(pcX$x[,2])+10),  
 )  
 text(pcX$x[,1],pcX$x[,2], labels, pos=3, cex=0.8)  
 titulo <- ifelse(dataDesc=="", "Visualización de las dos primeras componentes", dataDesc)  
 title(titulo, cex=0.8)  
}  
  
plotPCA(exprs(rawData), labels=affySampleNames)



El análisis de componentes principales nos permite detectar si las muestras se agrupan de forma “natural”. En este caso vemos como las líneas “Wild Type” parecen estar agrupadas, mientras que las que presentan sobreexpresión se alejan del resto. Esto no tiene porque ser indicativo de la existencia de un problema, pero sí deberemos estar al cuidado de un posible efecto batch. Éste sucede cuando se observan diferencias entre muestras que fueron procesadas y analizadas separadamente, podría deberse a problemas técnicos en la hibridación, preparación o escaneo de las muestras. Por este tipo de efectos, es conveniente normalizar las muestras para así eliminar las diferencias sistemáticas entre muestras.

### Cluster Jerárquico

clust.euclid.average <- hclust(dist(t(exprs(rawData))),method="average")  
plot(clust.euclid.average, labels=colnames(exprs(rawData)), main="Hierarchical clustering of samples", hang=-1, cex=0.7)



Al igual que en el análisis de componentes principales, vemos como las muestras Wild Type son más similares entre ellas, mientras que las Sobreexpresadas difieren más. Una posible explicación de ésta variación sería que efecto de las condiciones experimentales afecte tan sólo a un reducido número de genes.

# Normalización

# Random BS GO

<https://support.bioconductor.org/p/41945/> <https://support.bioconductor.org/p/64888/> <https://support.bioconductor.org/p/46513/> <https://rdrr.io/bioc/Biobase/man/phenoData.html> <https://support.bioconductor.org/p/64888/>

Ultimate <https://aspteaching.github.io/Analisis_de_datos_omicos-Materiales_para_un_curso/exploraci%C3%B3n-de-los-datos-control-de-calidad-y-preprocesado.html>