PEC1

Juan Carlos Prieto Prieto

09 de abril de 2020

# Abstract:

Los ciclos de luz y oscuridad ejercen una fuerte influencia sobre los procesos metabólicos y fisiológicos de las plantas. Para estudiar esta influencia se utilizó como organismo modelo la planta Arabidopsis thaliana, ecotipo Columbia-0 (Col-0), cultivada en períodos de 12 horas de luz y 12 horas de oscuridad. Se tomaron muestras de RNA con 3 réplicas biológicas durante un día (24 horas) a intervalos de 4 horas (ZT0, ZT4, ZT8, ZT12, ZT16 y ZT20)

# Objetivos:

Se pretende estudiar los cambios inducidos por los niveles de homocisteína en la expresión genética de células embrionarias de pollo (Gallus Gallus).

# Materiales y métodos:

## 1:

Se tomaron muestras de RNA con 3 réplicas biológicas durante un día (24 horas) a intervalos de 4 horas (ZT0, ZT4, ZT8, ZT12, ZT16 y ZT20). El periodo de luz va desde ZT0 a ZT12 y el de oscuridad de ZT12 a ZT0. Usando placas de microarray de affymetrix ATH1-121501.

## 2:

Comenzamos cargando los datos necesarios para el estudio desde los directorios dode se almacena.

setwd("C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel")  
library(affy)

## Warning: package 'affy' was built under R version 3.4.2

## Loading required package: BiocGenerics

## Warning: package 'BiocGenerics' was built under R version 3.4.2

## 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, cbind, colMeans,  
## colnames, colSums, do.call, duplicated, eval, evalq, Filter,  
## Find, get, grep, grepl, intersect, is.unsorted, lapply,  
## lengths, Map, mapply, match, mget, order, paste, pmax,  
## pmax.int, pmin, pmin.int, Position, rank, rbind, Reduce,  
## rowMeans, rownames, rowSums, sapply, setdiff, sort, table,  
## tapply, union, unique, unsplit, which, which.max, which.min

## Loading required package: Biobase

## Warning: package 'Biobase' was built under R version 3.4.2

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

microarrays <- ReadAffy(verbose=TRUE)

## 1 reading C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77055.cel ...instantiating an AffyBatch (intensity a 506944x18 matrix)...done.  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77055.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77056.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77057.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77058.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77059.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77060.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77061.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77062.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77063.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77064.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77065.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77066.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77067.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77068.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77069.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77070.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77071.cel  
## Reading in : C:/Users/Antonio/Desktop/Medicina/Master/Datos Ómicos/PEC1/gsm77055.cel/gsm77072.cel

microarrays

## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail'  
## when loading 'ath1121501cdf'

## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head'  
## when loading 'ath1121501cdf'

##

## AffyBatch object  
## size of arrays=712x712 features (23 kb)  
## cdf=ATH1-121501 (22810 affyids)  
## number of samples=18  
## number of genes=22810  
## annotation=ath1121501  
## notes=

Y realizamos un análisis preeliminar de los datos; podemos observar la distribución mediante el histograma y el boxplot, así como una imagen general (solo de dos de los genes estudiados) y el control de calidad mediante el quality plot.

library(affy)  
library(simpleaffy)

## Warning: package 'simpleaffy' was built under R version 3.4.2

## Loading required package: genefilter

## Warning: package 'genefilter' was built under R version 3.4.2

## Loading required package: gcrma

## Warning: package 'gcrma' was built under R version 3.4.2

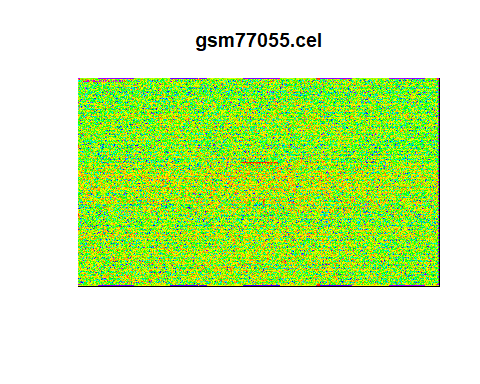
library(affyPLM)

## Warning: package 'affyPLM' was built under R version 3.4.2

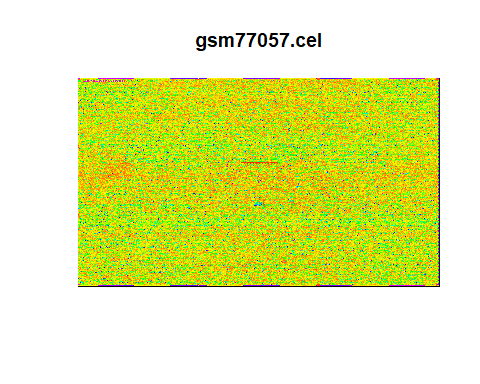
## Loading required package: preprocessCore

## Warning: package 'preprocessCore' was built under R version 3.4.2

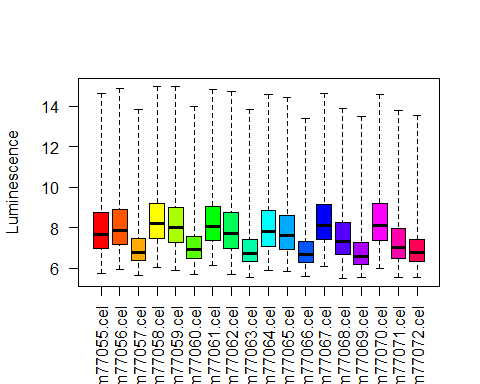
image(microarrays[,1],col=rainbow(100))



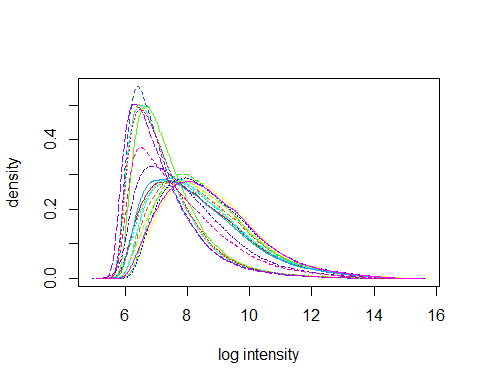
image(microarrays[,3],col=rainbow(100))



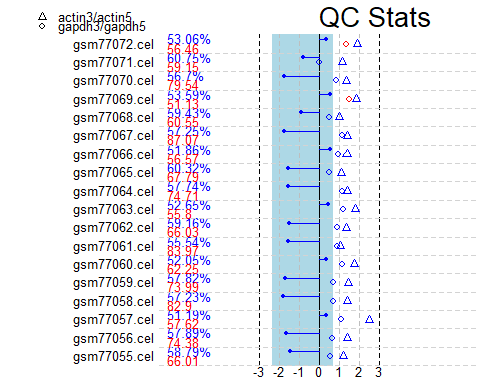
boxplot(microarrays,col=rainbow(18),las=2,ylab="Luminescence")



hist(microarrays,col=rainbow(18))



quality.control <- qc(microarrays)  
plot(quality.control)

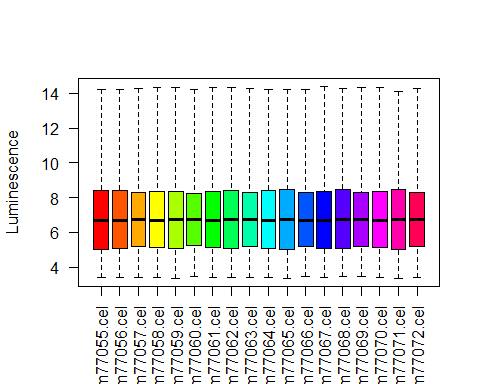


Pasamos a realizar la normalización mediante el modelo RMA. Y analizar visualmente los resultados tras esto.

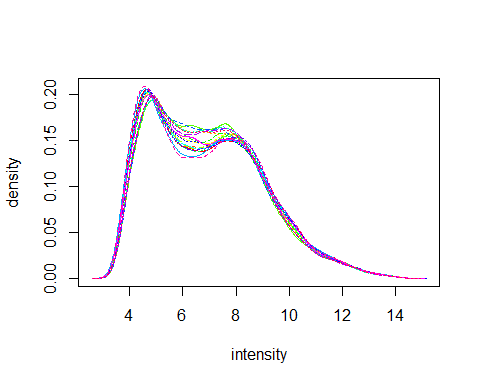
microarray.data <- rma(microarrays)

## Background correcting  
## Normalizing  
## Calculating Expression

boxplot(microarray.data,col=rainbow(18),las=2,ylab="Luminescence")



hist(microarray.data,col=rainbow(18))



Y extraemos los niveles de expresión de la misma matriz generada:

expression.level <- exprs(microarray.data)  
probe.names <- rownames(expression.level)  
sampleID <- c("ZT0\_1", "ZT0\_2", "ZT0\_3", "ZT4\_1", "ZT4\_2", "ZT4\_3", "ZT8\_1", "ZT8\_2", "ZT8\_3", "ZT12\_1", "ZT12\_2", "ZT12\_3", "ZT16\_1", "ZT16\_2", "ZT16\_3", "ZT20\_1", "ZT20\_2", "ZT20\_3")  
colnames(expression.level) <- sampleID  
head(expression.level)

## ZT0\_1 ZT0\_2 ZT0\_3 ZT4\_1 ZT4\_2 ZT4\_3 ZT8\_1  
## 244901\_at 4.947166 4.862576 4.802593 4.820768 4.573299 4.683383 4.805028  
## 244902\_at 5.187948 5.370610 5.358824 5.242519 5.169807 5.142875 5.083032  
## 244903\_at 7.787926 7.823443 8.080861 7.972121 7.376041 7.588188 7.636445  
## 244904\_at 6.106362 6.185612 7.026833 6.104678 6.203739 6.780890 5.897576  
## 244905\_at 5.343746 5.337174 5.765976 5.440447 5.477690 5.859851 5.430369  
## 244906\_at 7.307836 7.329513 7.363519 7.032156 6.931119 7.195222 7.101720  
## ZT8\_2 ZT8\_3 ZT12\_1 ZT12\_2 ZT12\_3 ZT16\_1 ZT16\_2  
## 244901\_at 4.621146 4.612337 5.183763 4.605071 4.814971 5.032732 4.756518  
## 244902\_at 5.152346 5.313793 5.133865 5.343667 5.147884 5.441206 5.100305  
## 244903\_at 7.219912 7.680730 8.399629 7.056816 7.698338 8.454468 6.870980  
## 244904\_at 5.955663 6.601336 6.179191 6.122243 6.455301 5.908197 5.655960  
## 244905\_at 5.502199 6.079341 5.357539 5.332196 5.956530 5.494657 5.179534  
## 244906\_at 6.875890 6.994368 7.175088 7.038077 7.282623 7.203951 6.972775  
## ZT16\_3 ZT20\_1 ZT20\_2 ZT20\_3  
## 244901\_at 4.957445 4.844900 4.787144 5.037835  
## 244902\_at 5.224320 5.221313 5.418099 5.113096  
## 244903\_at 8.529674 7.812527 7.169449 8.426873  
## 244904\_at 6.533285 5.815083 6.197616 6.468704  
## 244905\_at 5.950101 5.387417 5.178941 5.773538  
## 244906\_at 7.577668 7.127551 7.373099 7.280320

Para realizar la selección de genes expresados diferencialmente utilizaremos el paquete limma, realizando una serie de métodos estadísticos. Primero creamos una matriz con nuestros resultados y luego lo modelamos con la función lmfit:

library(limma)

## Warning: package 'limma' was built under R version 3.4.2

##   
## Attaching package: 'limma'

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

experiment.design <- model.matrix(~ -1+factor(c(1,1,1,2,2,2,3,3,3,4,4,4,5,5,5,6,6,6)))  
colnames(experiment.design) <- c("ZT00","ZT04","ZT08","ZT12","ZT16","ZT20")  
fit <- lmFit(expression.level, experiment.design)

Para obtener los contrastes utilizaremos la función makecontrast realizando algún cambio en los datos:

contrast.matrix <- makeContrasts(ZT00-ZT04,ZT00-ZT08,ZT00-ZT12,ZT00-ZT16,ZT00-ZT20,ZT04-ZT08,ZT04-ZT12,ZT04-ZT16,ZT04-ZT20,ZT08-ZT12,ZT08-ZT16,ZT08-ZT20,ZT12-ZT16,ZT12-ZT20,ZT16-ZT20,levels=c("ZT00","ZT04","ZT08","ZT12","ZT16","ZT20"))

Y ahora pasamos a ajustar el modelo lineal según las comparaciones que hemos realizado antes:

fit2 <- contrasts.fit(fit, contrast.matrix)  
ebayes <- eBayes(fit2)

DEGs.ZT00.ZT04 <- topTable(ebayes, number=22810,coef=1)  
head(DEGs.ZT00.ZT04)

## logFC AveExpr t P.Value adj.P.Val  
## 257628\_at -3.310220 7.224584 -29.04041 5.995718e-15 7.919203e-11  
## 266720\_s\_at -3.487090 4.928624 -28.76228 6.943624e-15 7.919203e-11  
## 258225\_at 3.034219 8.353683 27.17749 1.646457e-14 1.251856e-10  
## 260770\_at 4.337987 7.313356 25.75633 3.724996e-14 2.124179e-10  
## 245906\_at 2.301023 10.190624 24.29599 9.026879e-14 4.118062e-10  
## 252367\_at 6.279045 9.336706 23.41869 1.574694e-13 5.986462e-10  
## B  
## 257628\_at 23.59676  
## 266720\_s\_at 23.48080  
## 258225\_at 22.78757  
## 260770\_at 22.11526  
## 245906\_at 21.36904  
## 252367\_at 20.89123

En esta tabla nos interesan los valores de P.value y el valor de p ajustado al nivel de significación que queramos usar. Para realizar la selección usaré el fold-change, con valores de expresión superiores a 99% (de significación)

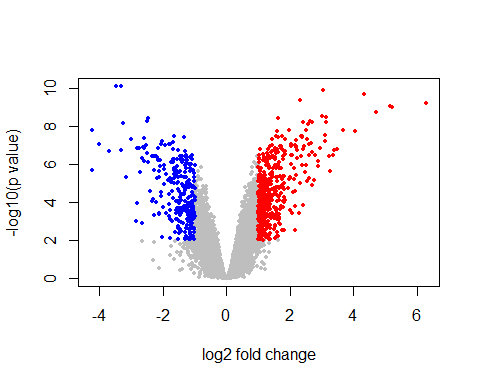
fold.change.ZT00.ZT04 <- DEGs.ZT00.ZT04[["logFC"]]  
log.p.value.ZT00.ZT04 <- -log10(DEGs.ZT00.ZT04[["adj.P.Val"]])  
probe.names <- DEGs.ZT00.ZT04[["ID"]]

Determinamos como genes activados de forma diferencial activated.ZT00.ZT04 aquellos cuyo fold-change es superior a 1 con una significancia mayor de 99% (p-valor menor 0.01 por lo tanto el correspondiente exponente obtenido como - log10 debe ser mayor de 2). De forma análoga determinamos como genes inhibidos de forma diferencial inhibited.ZT00.ZT04 aquellos cuyo fold.change es inferior a -1 (esto corresponde a genes que ven su expresión disminuida en menos de la mitad) con una significancia mayor del 99%:

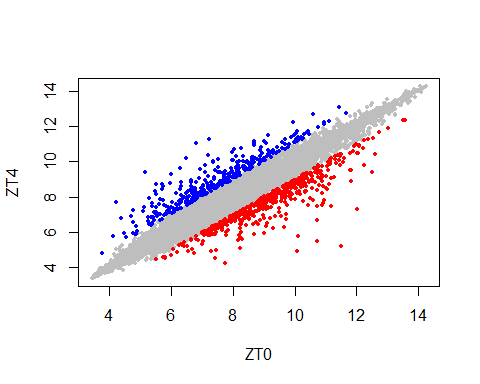
activated.ZT00.ZT04 <- (fold.change.ZT00.ZT04 > 1) & (log.p.value.ZT00.ZT04 > 2)  
inhibited.ZT00.ZT04 <- (fold.change.ZT00.ZT04 < - 1) & (log.p.value.ZT00.ZT04 > 2)

Y pasamos a representar gráficamente los genes medianteun volcano plot y un gráfico de dispersión:

plot(fold.change.ZT00.ZT04,log.p.value.ZT00.ZT04,pch=19,cex=0.5,col="grey",ylab="-log10(p value)",xlab="log2 fold change")  
points(fold.change.ZT00.ZT04[activated.ZT00.ZT04],log.p.value.ZT00.ZT04[activated.ZT00.ZT04],pch=19,cex=0.5,col="red")  
points(fold.change.ZT00.ZT04[inhibited.ZT00.ZT04],log.p.value.ZT00.ZT04[inhibited.ZT00.ZT04],pch=19,cex=0.5,col="blue")



cond0h <- (expression.level[,"ZT0\_1"] + expression.level[,"ZT0\_2"] + expression.level[,"ZT0\_3"])/3  
cond4h <- (expression.level[,"ZT4\_1"] + expression.level[,"ZT4\_2"] + expression.level[,"ZT4\_3"])/3  
cond8h <- (expression.level[,"ZT8\_1"] + expression.level[,"ZT8\_2"] + expression.level[,"ZT8\_3"])/3  
cond12h <- (expression.level[,"ZT12\_1"] + expression.level[,"ZT12\_2"] + expression.level[,"ZT12\_3"])/3  
cond16h <- (expression.level[,"ZT16\_1"] + expression.level[,"ZT16\_2"] + expression.level[,"ZT16\_3"])/3  
cond20h <- (expression.level[,"ZT20\_1"] + expression.level[,"ZT20\_2"] + expression.level[,"ZT20\_3"])/3  
mean.expression <- matrix(c(cond0h,cond4h,cond8h,cond12h,cond16h,cond20h), nrow=6,ncol=length(cond0h),byrow=TRUE)  
colnames(mean.expression) <- names(cond0h)  
rownames(mean.expression) <- c("ZT0", "ZT4", "ZT8", "ZT12", "ZT16", "ZT20")  
plot(mean.expression["ZT0",],mean.expression["ZT4",],pch=19,cex=0.5,col="grey",xlab="ZT0",ylab="ZT4")  
points(mean.expression["ZT0",(cond0h-cond4h)>1],mean.expression["ZT4",(cond0h-cond4h) > 1],pch=19,cex=0.5,col="red")  
points(mean.expression["ZT0",(cond0h-cond4h) < -1],mean.expression["ZT4",(cond0h-cond4h) < -1],pch=19,cex=0.5,col="blue")



# Discusión:

Al analizar los datos obtenidos en un estudio de microarrays es importante tener en cuenta cómo se han obtenido los datos (el número de repeticiones, el número de genes estudiados, y los procesos estadísticos realizados)