PEC1-Analisis de Datos Omicos

Julia Fernandez Reyes

29/04/2021

1. Abstract

El síndrome de Turner es una enfermedad genética causada por la pérdida total o parcial de una de las copias del cromosoma X, afectando únicamente a mujeres, pudiendo clasificar a las pacientes en los grupos 45Xm o 45Xp, de herencia materna o paterna respectivamente. Así, se ha demostrado que existen diferencias en la severidad del fenotipo de las pacientes según el cromosoma heredado, incluyendo diferente prevalencia para patologías cardiovasculares.

2. Objetivos

El objetivo principal de este estudio es estudiar el impacto del cromosoma X heredado en el fenotipo del síndrome de Turner. Para ello, se realizaron estudios por microarray de expresión procedentes de pacientes con genotipo 45Xm y 45Xp, comparadas con muestras de genotipo 46XX analizadas en paralelo, identificando también cambios en el patrón de expresión génica entre pacientes y genotipos sanos.

3. Materiales y métodos

Naturaleza de los datos:

En este trabajo se analizan muestras de humano (homo sapiens) procedentes tanto de pacientes de síndrome de Turner como individuos control. Las muestras proceden de células mononucleares de sangre periférica (PBMC), habiendo un total de 10 muestras con genotipo 46XX, 16 de genotipo 45Xm y 10 con genotipo 45Xp.

El array utilizado en este estudio es el Affymetrix Human Genome U133 Plus 2.0 Array, que se trata de un único array en el que se representan mas de 14.000 genes humanos bien caracterizados, de tal manera que se ha utilizado para explorar o identificar diferencias de expresión entre procesos fisiológicos propios de la biología humana y procesos patológicos.

Análisis realizado:

El procedimiento general del análisis llevado a cabo se resume en los siguientes pasos:

1) Descarga y preparación de los datos. Se va a trabajar con un set de datos de 18 muestras aleatorias.
2) Exploración de los datos y control de calidad: gráficamente se va a estudiar la estructura de nuestro set de datos para ver si requieren alguna corrección y de un paso de normalización. En este caso no se realizará filtraje previo, ya que puede quitar información de los datos.
3) Normalización de los datos con RMA: se utiliza el algoritmo Robust Multiarray Average (RMA), que descarta las sondas MM (mismatched match) ya que añaden más variación y mezclan fondo y señal de hibridación. Ajusta las intensidades del fondo, normaliza por cuantiles los PM (perfect match) y transforma los niveles de expresión a log2.

4) Exploración de los datos normalizados: con gráficos como boxplot, dendograma y gráfico de densidad, para comprarlos con los gráficos realizados en el paso 2 y poder ver el efecto de la normalización de los datos.

5) Matriz de diseño y matriz de contraste: en el que indicamos las tres condiciones que vamos a comparar.

6) Identificación de genes diferencialmente expresados: tras ajustar el modelo lineal con limma se seleccionan los genes con un p-valor corregido (FDR) menor de 0.05. Este grupo de genes será utilizado posteriormente para un análsis más detallado.

7) Visualización de los resultados: con una representación de volcano plot en la que se señala el top 10 de los genes diferencialmente expresados. También se va a realizar un heatmap, una matriz de colores para visualizar los genes diferencialmente expresados entre los controles y los casos.

8) Anotación de resultados: de los genes seleccionados previamente según su valor de FDR. Se va a utilizar la base de datos correspondiente a: Affymetrix Human Genome U133 Plus 2.0 Array. De forma resumida se crea una tabla con el identificador de cada gen, el valor de FoldChange, su p-valor, el p-valor corregido (FDR) y los resultados de la anotación: símbolo del gen, identificador de Entrez y la función con la que dicho gen está relacionado.

9) Análisis de significación biológica: utilizando todo el set de genes, se buscan los términos GO enriquecidos.

4 Resultados

4.1 Preparación de los datos

Primero cargamos librerias que se van a usar para poder hacer el estudio de los datos

```
library(BiocManager)
library(Biobase)
```

```
## Loading required package: BiocGenerics
```

```
## Warning: package 'BiocGenerics' was built under R version 4.0.5
```

```
## 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.max, which.min

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

```
library(knitr)
```

```
## Warning: package 'knitr' was built under R version 4.0.5
```

```
library(gplots)
```

```
##
## Attaching package: 'gplots'

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

```
library(limma)
```

```
##
## Attaching package: 'limma'

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

```
library(affy)
library(affyPLM)
```

```
## Loading required package: gcrma

## Loading required package: preprocessCore
```

```
#BiocManager::install("BiocGenerics")
library(org.Rn.eg.db)
```

```
## Loading required package: AnnotationDbi
```

```
## Loading required package: stats4

## Loading required package: IRanges

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

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

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

##
## Attaching package: 'IRanges'

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

##
```

```
library("BiocGenerics")
library("hgu133plus2.db")
```

```
## Loading required package: org.Hs.eg.db

##

##
```

```
library(annotate)
```

```
## Loading required package: XML
```

```
library(GOstats)
```

```
## Loading required package: Category

## Loading required package: Matrix

##
## Attaching package: 'Matrix'
```

```
## The following object is masked from 'package:S4Vectors':
##
##     expand

## Loading required package: graph

##
## Attaching package: 'graph'

## The following object is masked from 'package:XML':
##
##     addNode

##


##
## Attaching package: 'GOstats'

## The following object is masked from 'package:AnnotationDbi':
##
##     makeGOGraph
```

```
#BiocManager::valid()
```

```
#BiocManager::install("XVector")
#BiocManager::install("S4Vectors")
#BiocManager::install("matrixStats")
#BiocManager::install("limma")
#BiocManager::install("gcrma")
#BiocManager::install("hgu133plus2.db")
#BiocManager::install("IRanges")
#BiocManager::install("genefilter")
#BiocManager::install("Biostrings")
#BiocManager::install("AnnotationDbi")

library(limma)
library(gcrma)
library(AnnotationDbi)
library(BiocGenerics)
library(hgu133plus2.db)
```

Cargamos la información del set de datos generado aleatoriamente y obtenemos los ficheros CEL

```
selectSamples<- function (myID){
  set.seed(myID)
  selected <- c(sample(1:10, 6),11, sample(12:26, 5), sample(27:36,6))
  selected <- sort(selected)
}

mySelected <- selectSamples(1084613)
targetsAll <-read.csv(file="C:/Users/julia/Documents/MASTER/3 Cuatrimestre/Analisis de Datos Omicos/PEC
```

```
myTargets <- targetsAll[mySelected,]

#gse <- getGEO("GSM1134018", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134020", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134022", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134023", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134026", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134028", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134031", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134033", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134035", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134038", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134043", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134044", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134047", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134048", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134050", GSEMatrix = TRUE)
#gse <- getGEO("GSM1134051", GSEMatrix = TRUE)

#filePaths = getGEOSuppFiles("GSM1134016")
#filePaths = getGEOSuppFiles("GSM1134016")
#filePaths = getGEOSuppFiles("GSM1134017")
#filePaths = getGEOSuppFiles("GSM1134018")
#filePaths = getGEOSuppFiles("GSM1134020")
#filePaths = getGEOSuppFiles("GSM1134022")
#filePaths = getGEOSuppFiles("GSM1134023")
#filePaths = getGEOSuppFiles("GSM1134026")
#filePaths = getGEOSuppFiles("GSM1134028")
#filePaths = getGEOSuppFiles("GSM1134031")
#filePaths = getGEOSuppFiles("GSM1134033")
#filePaths = getGEOSuppFiles("GSM1134035")
#filePaths = getGEOSuppFiles("GSM1134038")
#filePaths = getGEOSuppFiles("GSM1134043")
#filePaths = getGEOSuppFiles("GSM1134044")
#filePaths = getGEOSuppFiles("GSM1134047")
#filePaths = getGEOSuppFiles("GSM1134048")
#filePaths = getGEOSuppFiles("GSM1134050")
#filePaths = getGEOSuppFiles("GSM1134051")


celFiles <- list.celfiles("C:/Users/julia/Documents/MASTER/3 Cuatrimestre/Analisis de Datos Omicos/PEC1/
sampleInfo <- read.AnnotatedDataFrame(file.path("C:/Users/julia/Documents/MASTER/3 Cuatrimestre/Analisis
                                      header = TRUE, row.names = 1, sep="\t")
rawData <- read.affybatch(filenames=celFiles,
                          phenoData=sampleInfo)



head(pData(sampleInfo))
```

```
##              title karyotype
## GSM1134016 XX_rep1      46XX
## GSM1134017 XX_rep2      46XX
```

```
## GSM1134018 XX_rep3      46XX
## GSM1134020 XX_rep5      46XX
## GSM1134022 XX_rep7      46XX
## GSM1134023 XX_rep8      46XX
```

```
expMat = exprs(rawData)
dim(expMat)
```

```
## [1] 1354896      18
```

```
head(expMat, 10)
```

```
##    GSM1134016 GSM1134017 GSM1134018 GSM1134020 GSM1134022 GSM1134023 GSM1134026
## 1          88        116        133        179        137        158        100
## 2        9954       7769       9658      12048      11534      10767       5107
## 3         175        173        190        217        214        183         85
## 4       11354       8560      10681      12201      11616      10738       5386
## 5         160        119        172        137        140        125         80
## 6         100        106        105         99        118        110         73
## 7       10122       8012      10307      12388      11723      10197       5339
## 8         124        121        111        128        139        124         87
## 9        9548       7146       9474      11930      10935      10298       5231
## 10        132        118        111        140        139        117         81
##    GSM1134028 GSM1134031 GSM1134033 GSM1134035 GSM1134038 GSM1134043 GSM1134044
## 1          73        254        329        197        271         60        232
## 2        4965      14022      13523      10671      15414       4226      11902
## 3          70        244        330        347        427         78        300
## 4        5056      14376      13543      10821      15093       4323      12416
## 5          65        179        172        234        255         65        223
## 6          63        171        221        192        245         55        211
## 7        4957      13816      13470      10700      15196       4037      11819
## 8          71        244        276        235        269         60        261
## 9        4908      13529      13718      11024      15050       4125      11889
## 10         70        199        316        242        274         50        242
##    GSM1134047 GSM1134048 GSM1134050 GSM1134051
## 1         312        265        304        206
## 2       12060      13745      12751      14694
## 3         399        305        433        235
## 4       12234      14123      12520      15172
## 5         208        273        251        160
## 6         232        206        273        182
## 7       11895      14021      12407      14370
## 8         349        276        319        195
## 9       11843      13715      12536      14150
## 10        320        260        291        219
```

4.2 Control de calidad y exploración de los datos

Análisis exploratorio de los datos crudos, con la representación de diferentes gráficos:

GRÁFICO DE DENSIDAD:

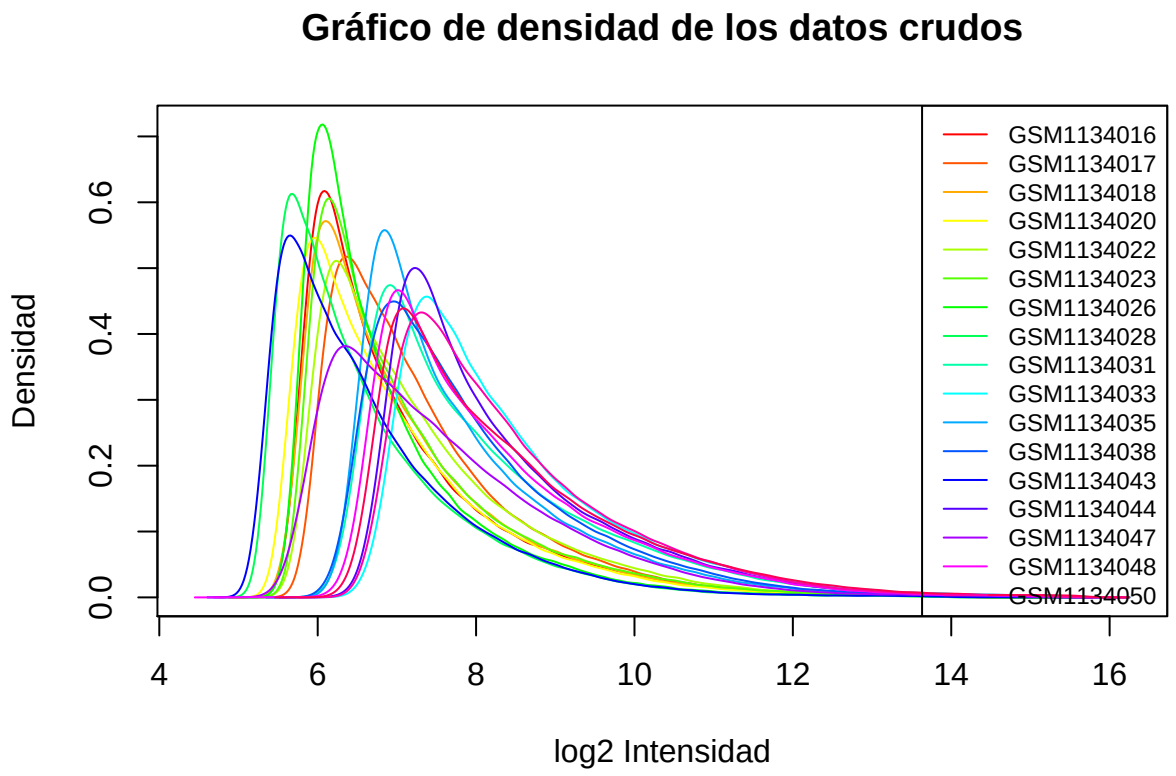
```
color <- rainbow(length(sampleNames(rawData)))
hist(rawData, main="Gráfico de densidad de los datos crudos",
     xlab = "log2 Intensidad", ylab= "Densidad",     col=color, lty=1)
```

```
## Warning: replacing previous import 'AnnotationDbi::tail' by 'utils::tail' when
## loading 'hgu133plus2cdf'
```

```
## Warning: replacing previous import 'AnnotationDbi::head' by 'utils::head' when
## loading 'hgu133plus2cdf'
```

```
##
```

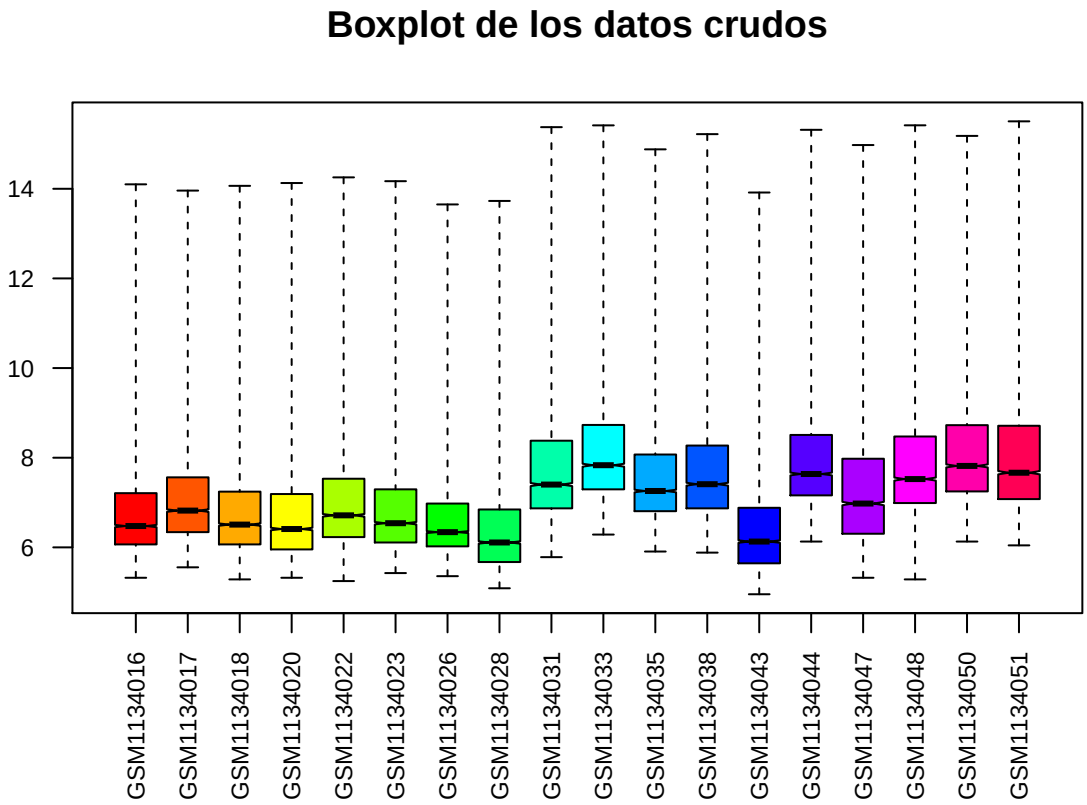
```
legend (x="topright", legend=sampleNames(rawData) ,
        col=color, lty=1, cex=0.75)
```



Las curvas de densidad no son muy distintas, pero los datos requieren una normalización.

BOXPLOT:

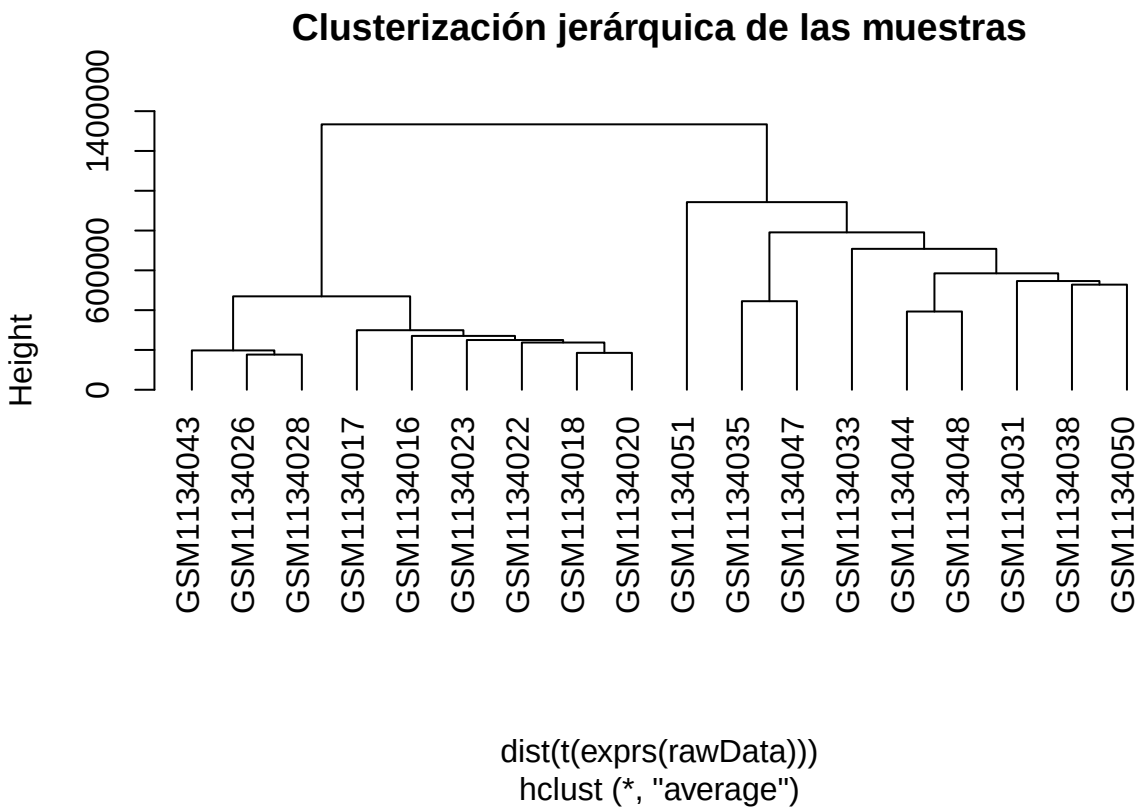
```
boxplot(rawData, main= "Boxplot de los datos crudos",
        cex.axis=0.75, col=color, las=2, names=sampleNames(rawData),
        notch=TRUE)
```



No hay mucha diferencia en la media de los datos, pero también nos indica la falta de normalización

DENDOGRAMA:

```
clust.euclid.average <- hclust(dist(t(exprs(rawData))),method="average")
plot(clust.euclid.average, labels=sampleNames(rawData),
     main="Clusterización jerárquica de las muestras", hang=-1)
```



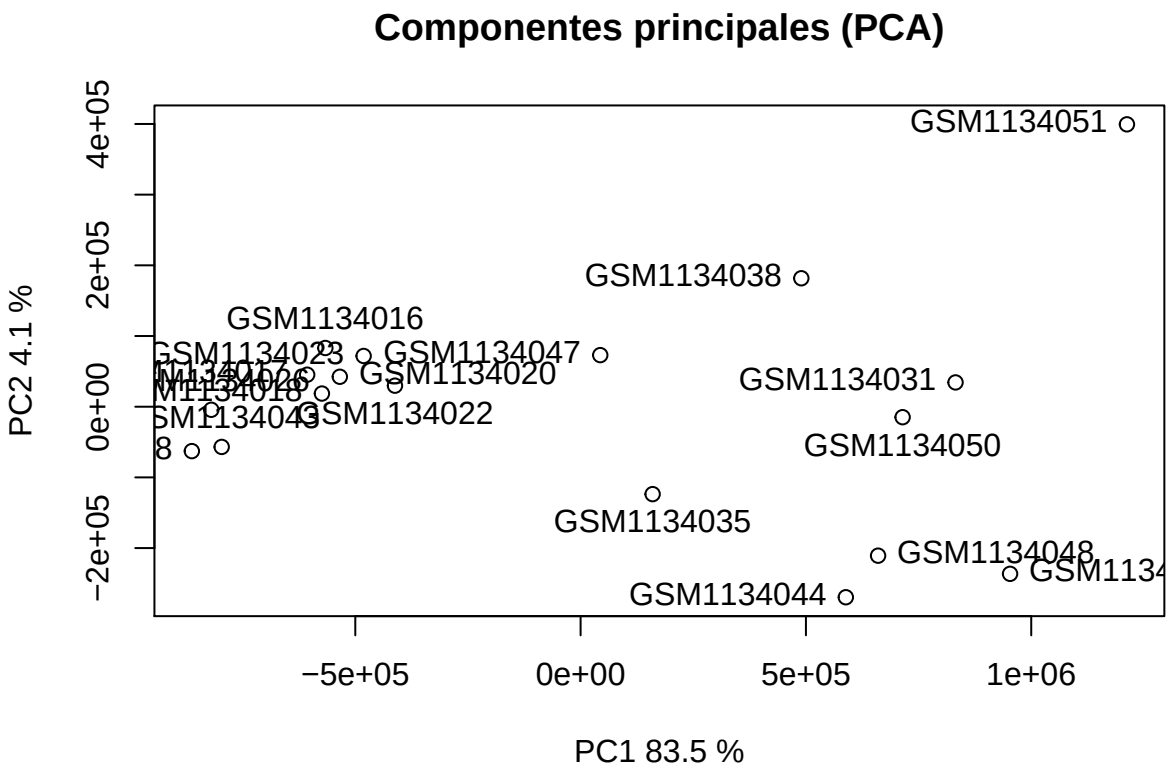
Esta figura soporta la premisa de una normalización requerida de los datos.

ANÁLISIS DE COMPONENTES PRINCIPALES:

Siguiendo con las representaciones de clusterización de los datos, se realiza un análisis de los componentes principales.

```
pcX<-prcomp(t(expMat), scale=FALSE)
loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))

plot(pcX$x[,1:2],xlab=xlab,ylab=ylab)
title("Componentes principales (PCA)")
text(pcX$x[,1],pcX$x[,2],colnames(rawData),pos=c(3,2,2,4,1,2))
```



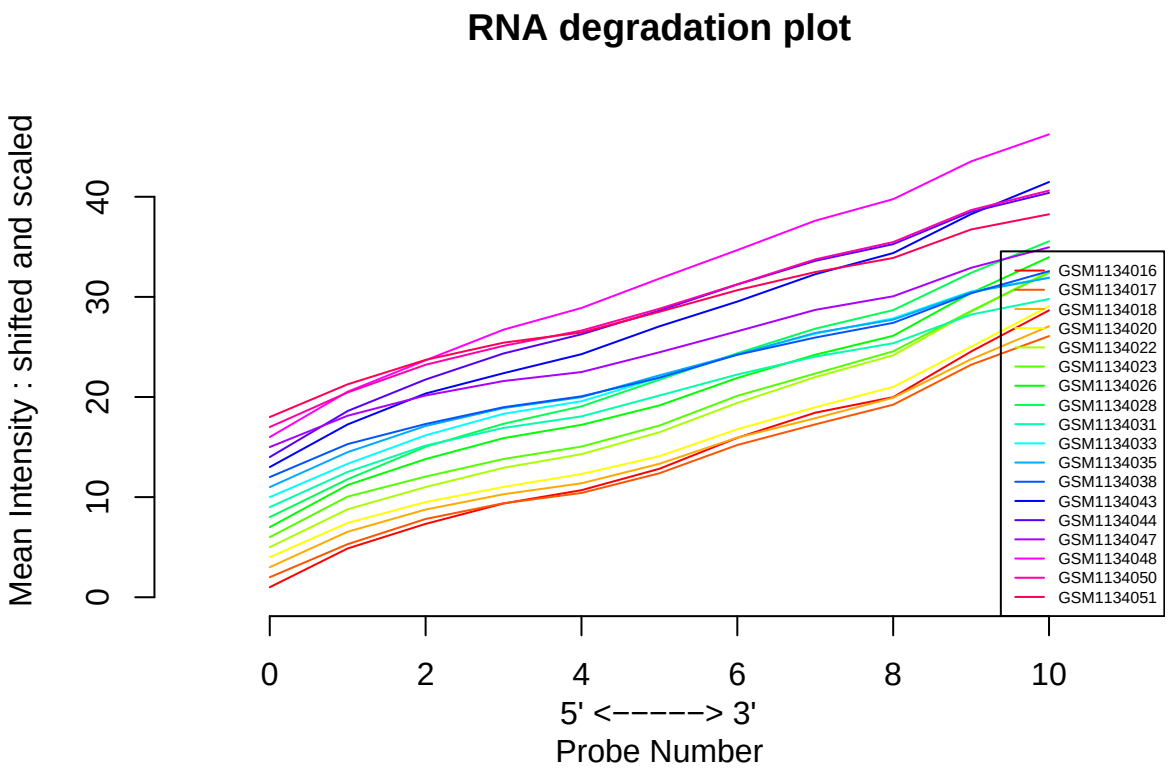
Al igual que en el dendograma, no hay una agrupación clara de las pacientes y los controles.

PLOT DE DEGRADACIÓN DEL ARN:

```
deg <- AffyRNAdeg(rawData, log.it=F)
summaryAffyRNAdeg(deg)
```

```
##        GSM1134016 GSM1134017 GSM1134018 GSM1134020 GSM1134022 GSM1134023
## slope    2.53e+00   2.24e+00   2.20e+00   2.28e+00   2.54e+00   2.42e+00
## pvalue   2.33e-09   1.35e-09   3.04e-09   1.06e-08   3.16e-09   3.45e-09
##        GSM1134026 GSM1134028 GSM1134031 GSM1134033 GSM1134035 GSM1134038
## slope    2.45e+00   2.60e+00   1.96e+00   2.13e+00   2.00e+00   1.92e+00
## pvalue   1.47e-09   1.26e-11   1.31e-10   3.19e-11   7.06e-11   6.82e-11
##        GSM1134043 GSM1134044 GSM1134047 GSM1134048 GSM1134050 GSM1134051
## slope    2.67e+00   2.50e+00   1.88e+00   2.90e+00   2.27e+00   1.93e+00
## pvalue   2.67e-11   1.62e-10   9.17e-11   2.17e-12   5.88e-12   4.42e-11
```

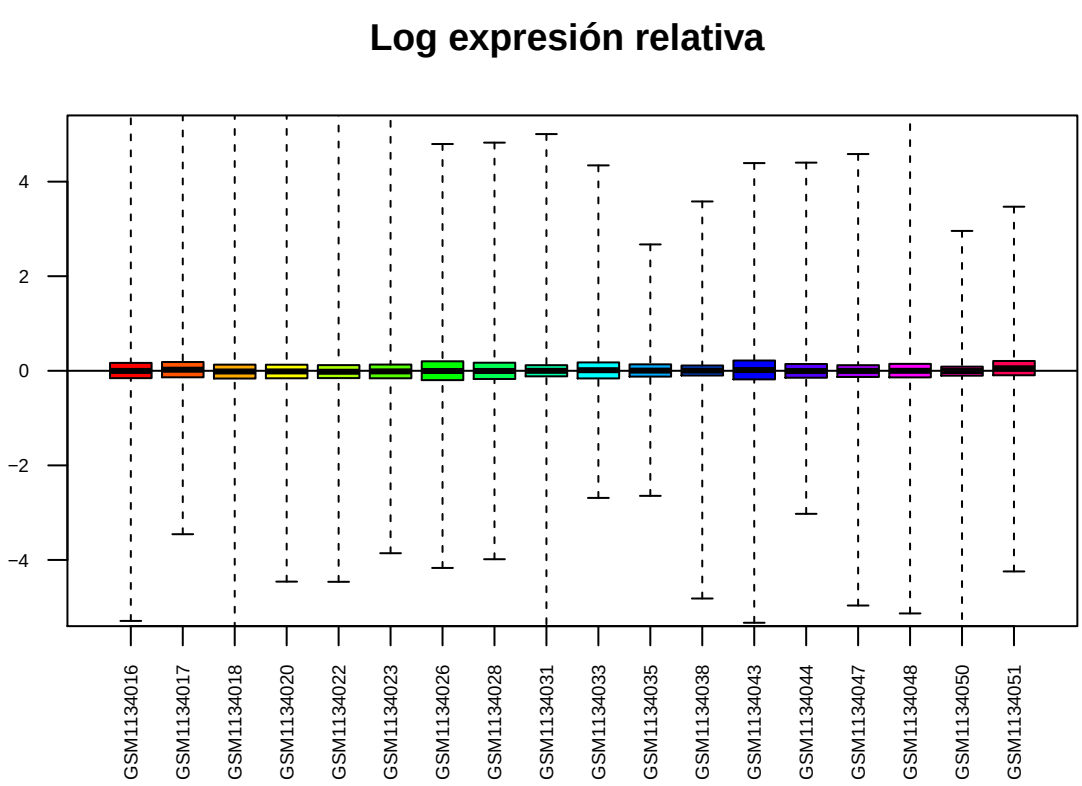
```
plotAffyRNAdeg(deg, lwd=2, col=color)
legend(x="bottomright", legend=sampleNames(rawData), col=color, lty=1, cex=0.5)
```



El perfil que sigue la degradación del RNA en todas las muestras es bastante similar, por lo que podemos descartar un problema con el RNA.

EXPRESIONES RELATIVAS (RLE) Y NUSE:

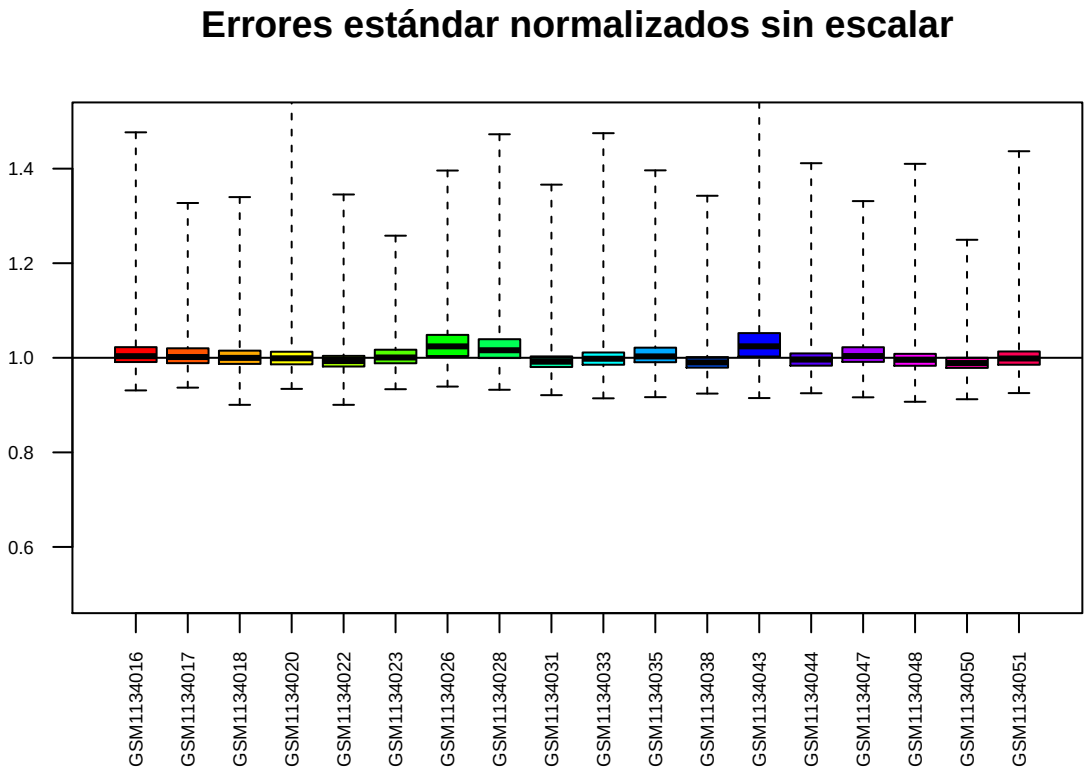
```
require(affyPLM)
Pset <- fitPLM(rawData)
RLE(Pset, main = "Log expresión relativa ", names=sampleNames(rawData),
    las=2, col=color, cex.axis=0.6,ylim=c(-5,5))
```



Los valores de RLE están cerca de 0, por lo que no existe ninguna muestra que se desvíe de lo esperado.

```
NUSE(Pset, main = "Errores estándar normalizados sin escalar",
     las=2, names=sampleNames(rawData), las=2, col=color, cex.axis=0.6, ylim=c(0.5,1.5))
```

```
## Warning in bxp(list(stats = structure(c(0.931055619436107, 0.99104744647757, :
## Duplicated argument las = 2 is disregarded
```



Todas las muestras siguen una distribución parecida, por lo que ningún array tiene una calidad baja con respecto a otros.

4.3 Normalización

Tras hacer una exploración extensa de los datos de forma gráfica concluimos que es necesaria la normalización de los datos, que nos permitirá comparar intensidades entre las distintas sondas del microarray

```
eset_rma <- rma(rawData)
```

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

```
eset_rma_matrix = exprs(eset_rma)
head(eset_rma_matrix)
```

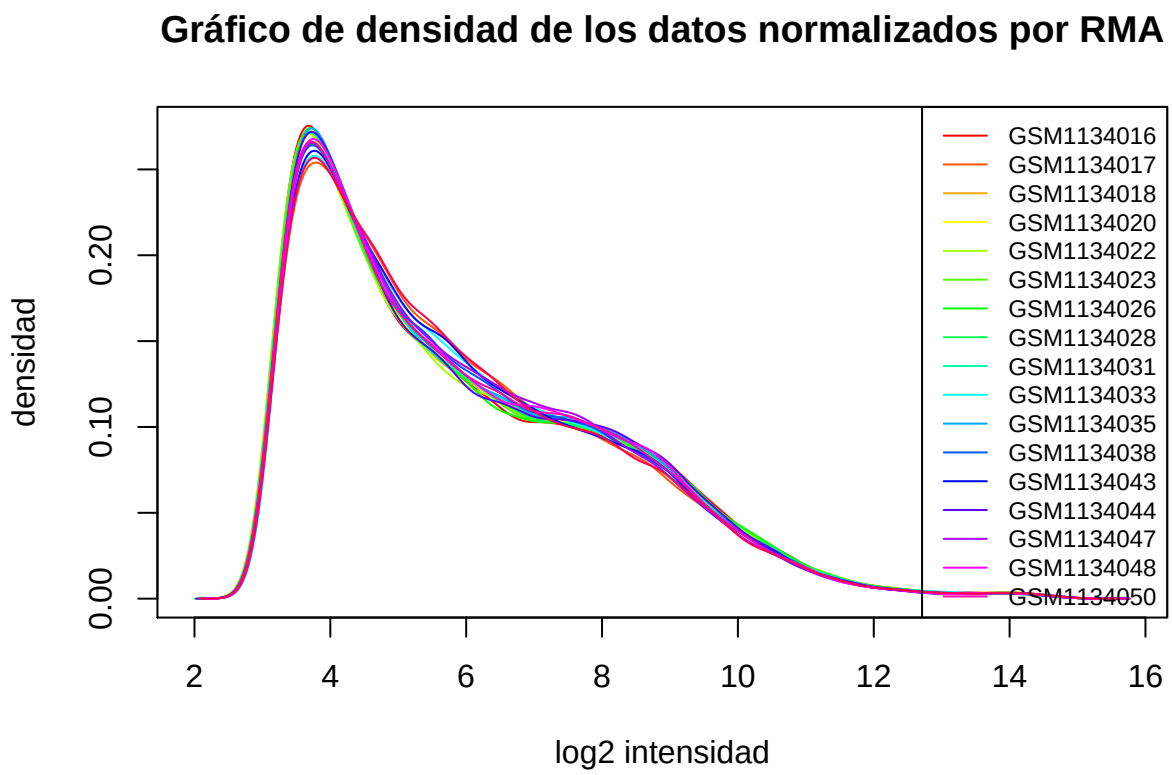
```
##           GSM1134016 GSM1134017 GSM1134018 GSM1134020 GSM1134022 GSM1134023
## 1007_s_at   7.318557   6.962393   6.657510   7.068992   6.887998   6.710283
## 1053_at     7.367662   7.192689   7.393529   7.393683   7.186786   7.448110
## 117_at      8.922991   8.279600   8.562964   8.218422   8.354296   7.898491
## 121_at      7.716635   8.020565   7.643451   7.917911   7.317513   7.744926
## 1255_g_at   3.491745   3.453112   3.400646   3.490880   3.238129   3.262528
## 1294_at     9.536167   9.237145   9.190361   8.967841   9.450060   9.245102
```

```
##           GSM1134026 GSM1134028 GSM1134031 GSM1134033 GSM1134035 GSM1134038
## 1007_s_at   6.714438   6.879812   6.885531   6.729946   6.665933   6.977679
## 1053_at     6.904110   6.572204   7.088911   6.845647   6.944125   7.158309
## 117_at      8.181790   9.377974   9.182782   9.331359   8.621798   7.803624
## 121_at      7.533464   7.487910   7.631438   7.643984   7.878171   7.787041
## 1255_g_at   3.437897   3.380564   3.632769   3.523146   3.399879   3.472889
## 1294_at     9.650040   9.773987   9.678793   9.816962   9.701489   9.669438
##           GSM1134043 GSM1134044 GSM1134047 GSM1134048 GSM1134050 GSM1134051
## 1007_s_at   6.706525   7.477870   6.844218   7.360736   7.202858   6.608907
## 1053_at     6.734386   6.696210   6.849293   6.625462   7.052863   7.167760
## 117_at      8.340028   8.675084   8.362341   8.236700   8.394988   7.852003
## 121_at      7.551392   7.157046   7.832003   7.420923   7.926811   7.808581
## 1255_g_at   3.383473   3.410122   3.709201   3.579567   3.373177   3.635814
## 1294_at     9.791451  10.159404   9.716137   9.930339   9.732139   9.751603
```

4.4 Exploracion de los datos normalizados

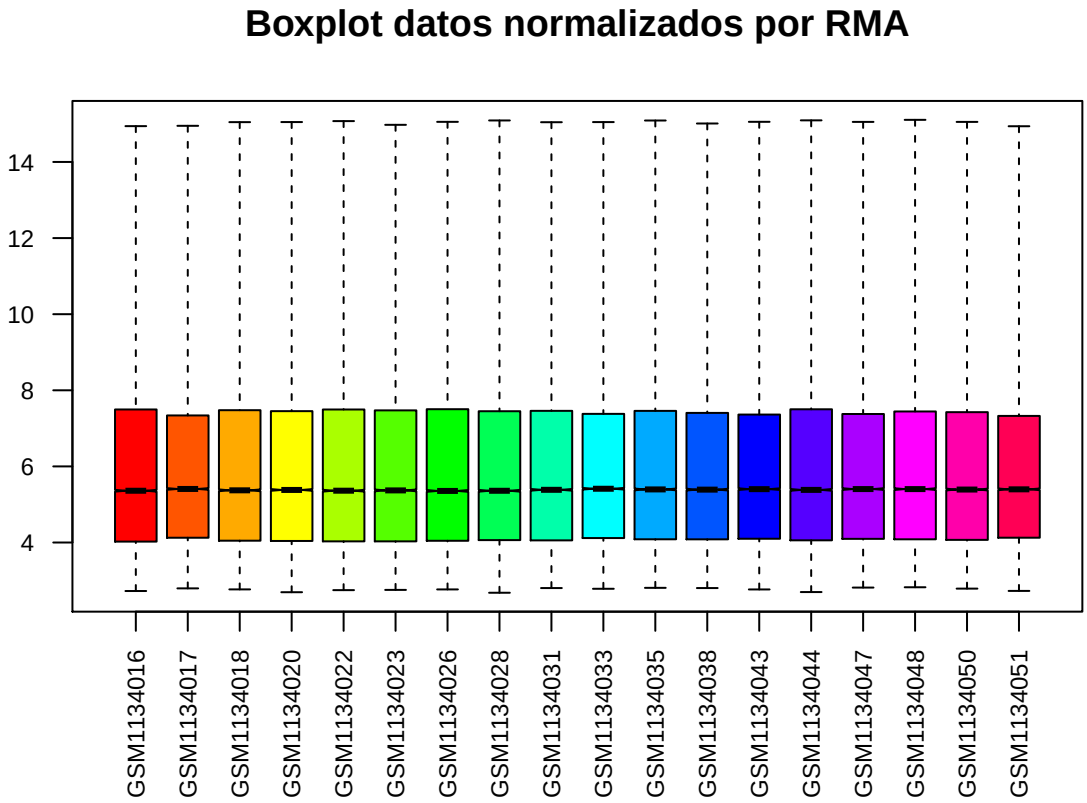
```
hist(eset_rma, main="Gráfico de densidad de los datos normalizados por RMA",
     xlab = "log2 intensidad", ylab= "densidad", col=color, lty=1)

legend ("topright", legend=sampleNames(rawData) , col=color, lty=1, cex=0.75)
```



Se puede observar que, las curvas de densidad, que antes no eran muy diferentes entre sí, ahora siguen un patrón similar y se asemejan tras realizar la normalización de los datos.

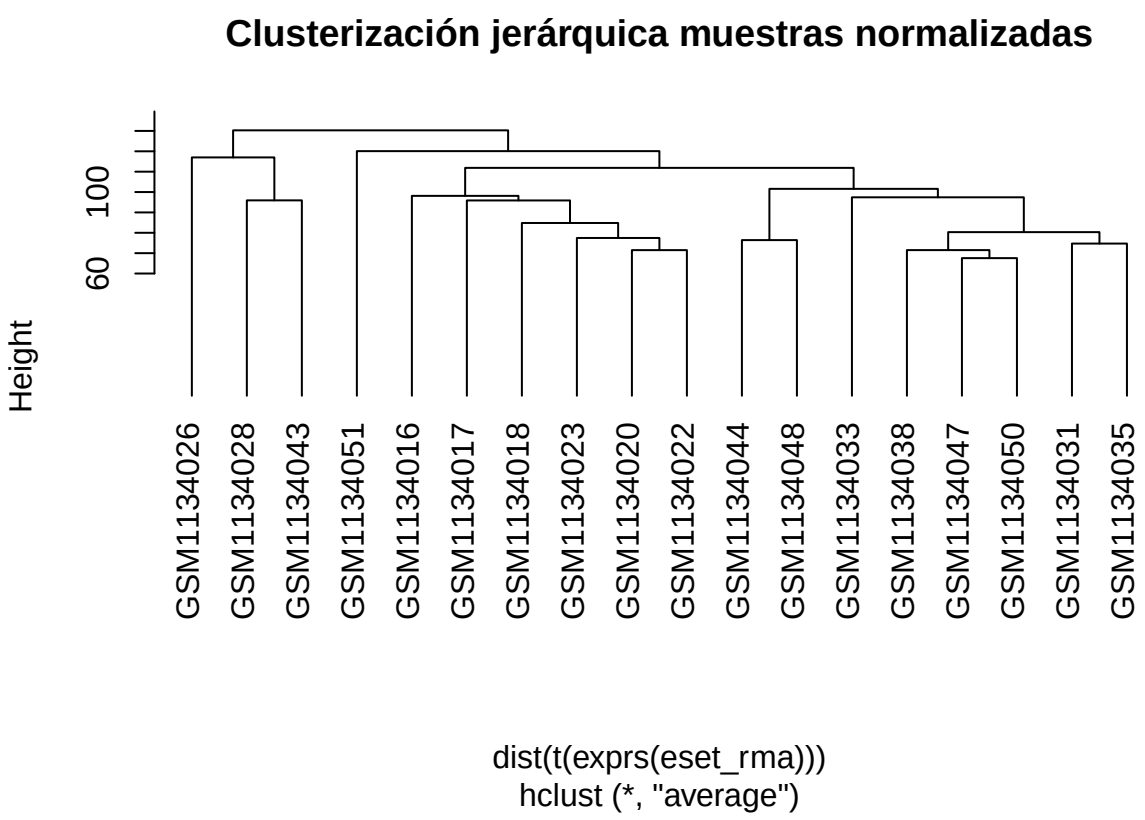
```
boxplot(eset_rma, main= "Boxplot datos normalizados por RMA",
        cex.axis=0.75, col=color, las=2, names=sampleNames(rawData), notch=TRUE)
```



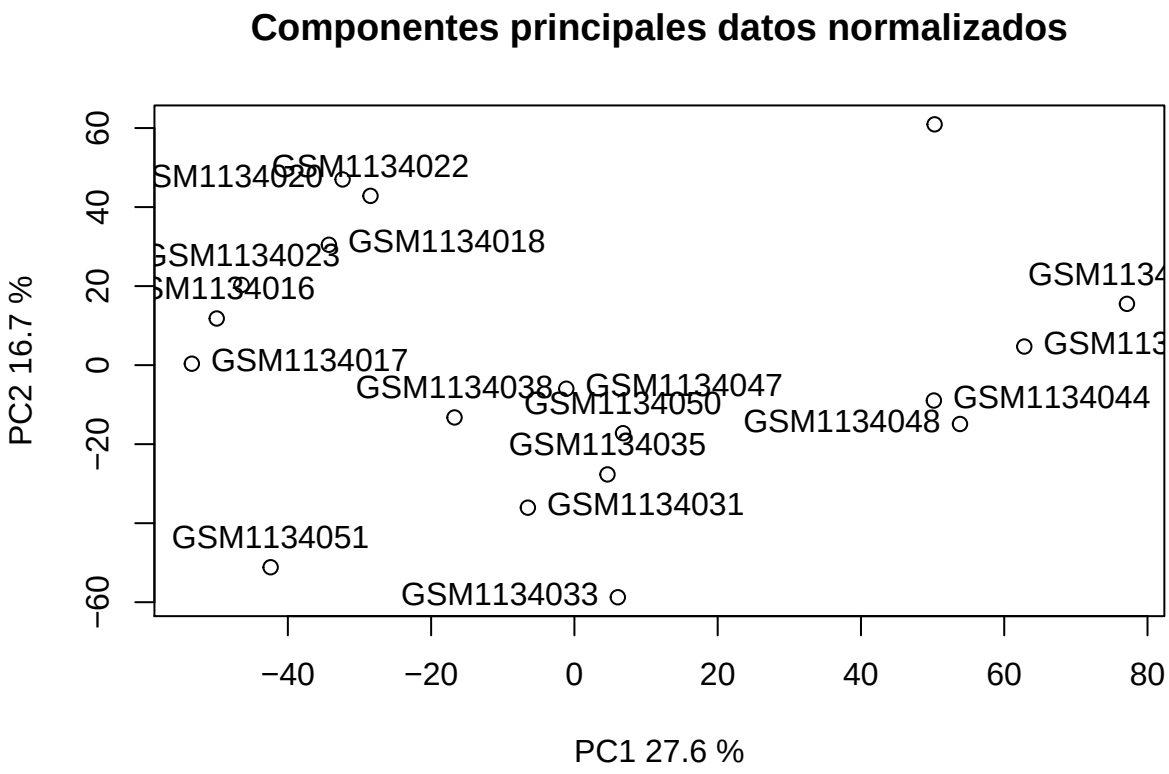
También se puede observar el efecto de la normalización en los diagramas de caja, donde la mediana y los cuartiles de los datos normalizados son muy similares.

```
clust.euclid.average <- hclust(dist(t(exprs(eset_rma))), method="average")

plot(clust.euclid.average, labels=sampleNames(rawData),
     main="Clusterización jerárquica muestras normalizadas", hang=-1)
```




```
pcX<-prcomp(t(eset_rma_matrix), scale=FALSE)
loads<- round(pcX$sdev^2/sum(pcX$sdev^2)*100,1)
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(pcX$x[,1:2],xlab=xlab,ylab=ylab)
title("Componentes principales datos normalizados")
text(pcX$x[,1],pcX$x[,2],colnames(rawData), pos=c(3,4,4,2,3,3))
```



Tanto en este gráfico como en el anterior, se puede observar que no se produce una agrupación de las muestras.

4.5 Matriz de diseño y matriz de contraste

Vamos a crear la matriz de diseño del modelo con la función pData y le vamos a dar el nombre que hemos utilizado para las muestras a lo largo de todo el análisis.

```
targets <- pData(eset_rma)
lev <- factor(targets$karyotype, levels=unique(targets$karyotype))
design <- model.matrix(~ 0+lev)
colnames(design)<-c("xx","xm","xp")
rownames(design) <-sampleNames(rawData)
print(design)
```

```
##            xx xm xp
## GSM1134016  1  0  0
## GSM1134017  1  0  0
## GSM1134018  1  0  0
## GSM1134020  1  0  0
## GSM1134022  1  0  0
## GSM1134023  1  0  0
## GSM1134026  0  1  0
## GSM1134028  0  1  0
## GSM1134031  0  1  0
```

```
## GSM1134033  0  1  0
## GSM1134035  0  1  0
## GSM1134038  0  1  0
## GSM1134043  0  0  1
## GSM1134044  0  0  1
## GSM1134047  0  0  1
## GSM1134048  0  0  1
## GSM1134050  0  0  1
## GSM1134051  0  0  1
## attr(,"assign")
## [1] 1 1 1
## attr(,"contrasts")
## attr(,"contrasts")$lev
## [1] "contr.treatment"
```

```
contrasts <- c("xx-xm-xp")
cont.matrix <- makeContrasts(contrasts=contrasts, levels=c("xx","xm","xp"))
cont.matrix
```

```
##       Contrasts
## Levels xx-xm-xp
##     xx        1
##     xm       -1
##     xp       -1
```

4.6 Selección de genes diferencialmente expresados

Ajustamos el modelo con limma

```
fit<-lmFit(eset_rma, design)
fit.main<-contrasts.fit(fit, cont.matrix)
fit.main<-eBayes(fit.main)
```

A las tablas vamos a añadir el p-valor ajustado por FDR

```
topTableControl_Caso <- topTable(fit.main, number=nrow(fit.main),
adjust="fdr")
head(topTableControl_Caso)
```

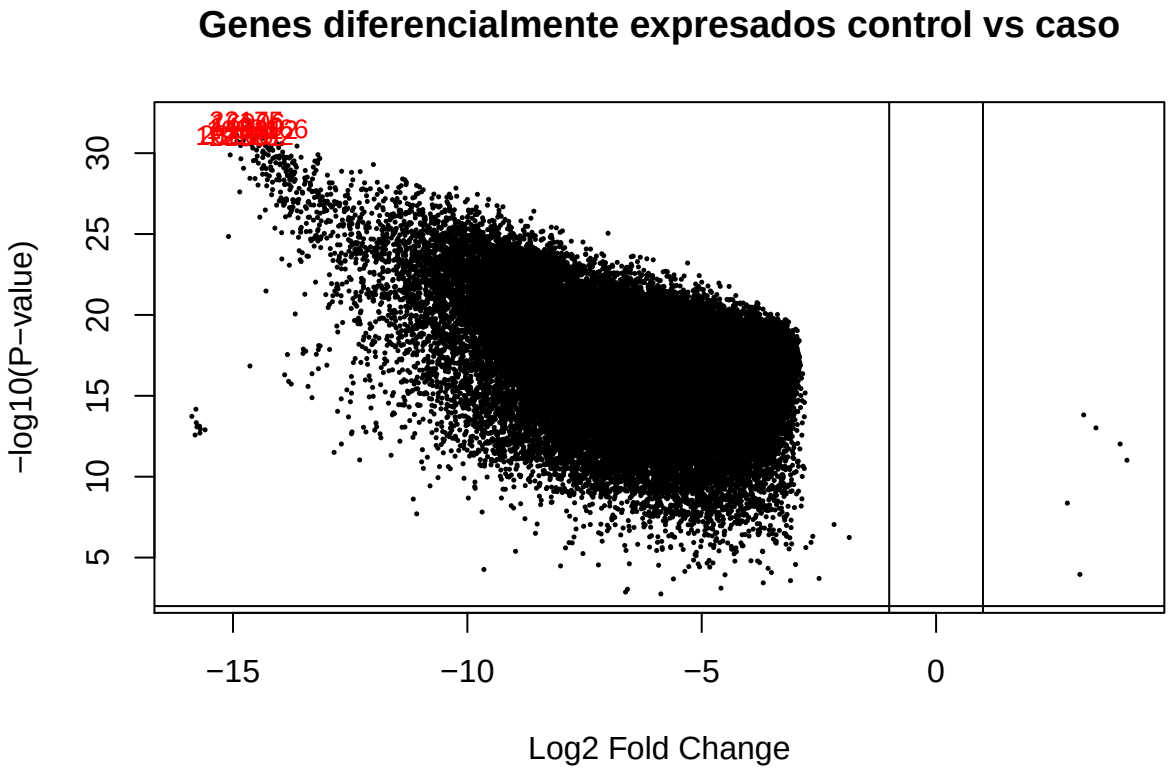
```
##                 logFC  AveExpr         t      P.Value    adj.P.Val        B
## 212869_x_at -14.70982 14.64459 -230.1165 1.037911e-32 3.434885e-28 53.17009
## 206559_x_at -14.67312 14.68429 -227.6885 1.256474e-32 3.434885e-28 53.12750
## 204892_x_at -14.71138 14.76546 -219.3748 2.455740e-32 3.529815e-28 52.97244
## 200717_x_at -14.17948 14.26369 -217.0719 2.969933e-32 3.529815e-28 52.92679
## 201492_s_at -14.77557 14.85488 -214.7308 3.610657e-32 3.529815e-28 52.87910
## 200763_s_at -14.39592 14.31991 -213.8946 3.873596e-32 3.529815e-28 52.86175
```

En esta tabla tenemos todos los estadísticos necesarios para seleccionar los genes diferencialmente expresados: logFC, p-valor y el p-valor ajustado por FDR calculado en el paso anterior.

4.7 Visualizacion de resultados

VOLCANO PLOT:

```
volcanoplot(fit.main, coef=1, highlight = 10, hl.col = "red",
names=fit.main$ID,
main="Genes diferencialmente expresados control vs caso")
abline(v=c(-1,1))
abline(h=2)
```



Vamos a hacer el heatmap de los genes que hemos filtrado por un p-valor corregido <0.05. Se representan en rojo los genes sobreexpresados, en azul los infraexpresados y en blanco los que tienen una expresión neutra.

HEATMAP:

```
FDR <- topTableControl_Caso[,5]
topTab.05 <- subset(topTableControl_Caso, FDR < 0.05)

probes.05 <- rownames(topTab.05)
exprs2cluster <- eset_rma_matrix[probes.05,]
#memory.limit(size = 8000)
#heatmap.2(exprs2cluster, col=bluered(75), scale="row", key=TRUE, symkey=FALSE,
#density.info="none", trace="none", cexCol=1, Rowv=TRUE, Colv=TRUE,
#labRow=TRUE, dendogram="both",
#main= "Heatmap de casos vs controles (FDR < 0.05)")
```

Debido al tamaño, no se puede visualizar la imagen. He estado mirando soluciones pero parece que es debido a un problema de RAM cedida a RStudio.

4.8 Anotacion de resultados

```
eset_rma@annotation
```

```
## [1] "hgu133plus2"
```

```
print(paste("genes with p-values smaller than 0.05:",length(which(FDR < 0.05))))
```

```
## [1] "genes with p-values smaller than 0.05: 54675"
```

```
topFC <- topTab.05$logFC
toppv <- topTab.05$P.Value
topFDR <- topTab.05$adj.P.Val

id <- rownames(topTab.05)

symbolTable <- toTable(hgu133plus2SYMBOL)
rownames(symbolTable) <- symbolTable$probe_id
geneSymbol.05 <-symbolTable[id, 2]
entrezTable <- toTable(hgu133plus2ENTREZID)
rownames(entrezTable) <- entrezTable$probe_id
geneEntrez.05 <- entrezTable[id,2]
titleTable <- toTable(hgu133plus2GENENAME)
rownames(titleTable) <- titleTable$probe_id
geneTitle.05 <- titleTable[id,2]

results <- data.frame(as.vector(id),FD=topFC, pvalue=toppv,
FDR=topFDR, as.vector(geneSymbol.05),
as.vector(geneEntrez.05), as.vector(geneTitle.05))
header <- c("ID", "FoldChange", "pvalue", "FDR", "GeneSymbol", "EntrezID", "GeneTitle")
colnames(results) <- header
head(results)
```

```
##            ID FoldChange       pvalue          FDR GeneSymbol EntrezID
## 1 212869_x_at  -14.70982 1.037911e-32 3.434885e-28       TPT1     7178
## 2 206559_x_at  -14.67312 1.256474e-32 3.434885e-28     EEF1A1     1915
## 3 204892_x_at  -14.71138 2.455740e-32 3.529815e-28     EEF1A1     1915
## 4 200717_x_at  -14.17948 2.969933e-32 3.529815e-28       RPL7     6129
## 5 201492_s_at  -14.77557 3.610657e-32 3.529815e-28      RPL41     6171
## 6 200763_s_at  -14.39592 3.873596e-32 3.529815e-28      RPLP1     6176
##                                          GeneTitle
## 1        tumor protein, translationally-controlled 1
## 2 eukaryotic translation elongation factor 1 alpha 1
## 3 eukaryotic translation elongation factor 1 alpha 1
## 4                             ribosomal protein L7
## 5                            ribosomal protein L41
## 6         ribosomal protein lateral stalk subunit P1
```

4.9 Análisis de significación biológica

Este tipo de análisis son más fiables cuanto mayor sea el número de genes con el que se trabaja, por lo que se va a utilizar la tabla creada incialmente no filtrada por el FDR.

```
entrezUniverse <- mappedkeys(org.Hs.egGO)
geneIds <- unique(getEG(x=id,data="hgu133plus2.db"))

GOparams <- new("GOHyperGParams", geneIds=geneIds, universeGeneIds=entrezUniverse, annotation="org.Hs.eg
ontology="BP", pvalueCutoff=0.001, conditional=FALSE, testDirection="over")
```

```
## Warning in makeValidParams(.Object): removing geneIds not in universeGeneIds
```

```
GOhyper <- hyperGTest(GOparams)
htmlReport(GOhyper, file="GO_results.html", summary.args=list("htmlLinks"=TRUE))
head(summary(GOhyper, categorySize=10))[c(1,2,7)]
```

```
##       GOBPID        Pvalue                                          Term
## 1 GO:0044260 1.565860e-171      cellular macromolecule metabolic process
## 2 GO:0071840 7.953808e-168 cellular component organization or biogenesis
## 3 GO:0016043 2.084703e-159               cellular component organization
## 4 GO:1901564 3.984381e-145     organonitrogen compound metabolic process
## 5 GO:0032502 3.060442e-124                         developmental process
## 6 GO:0048856 4.225822e-122              anatomical structure development
```

```
nrow(summary(GOhyper))
```

```
## [1] 2001
```

```
GOhyper
```

```
## Gene to GO BP  test for over-representation
## 15930 GO BP ids tested (2001 have p < 0.001)
## Selected gene set size: 15644
##     Gene universe size: 18866
##     Annotation package: org.Hs.eg
```

Se han encontrado 2001 términos GO.

En esta última tabla se resume la cabecera de los resultados obtenidos, señalando las columnas de interés: el ID del término GO, el p-valor asociado al mismo, y el nombre del proceso biológico (biological process).

5. Discusión

Este tipo de estudios de microarrays tienen limitaciones, como hemos visto previamente. Una de ellas, bastante patente en este análisis en concreto es el tamaño muestral. A priori, en nuestro caso dispondriamos de un tamaño muestral relativamente alto (N=18). No obstante, al dividirlo en tres grupos (46XX, 46Xm, 46Xp), vemos que se reduce el numero de réplicas que tendríamos, lo que podría afectar a nuestros resultados finales. Además, el estudio original se compone de 36 muestras, mientras que en este solo se ha trabajado con 18. Esto implica que de haber utilizado todas las muestras o un set de datos diferente, los resultados podrían variar.

Además, el método elegido para la normalización ha sido RMA, pero sabemos que existen muchos otros métodos y que los resultados obtenidos con estos pueden llegar a ser diferentes. La elección del método de normalización, por tanto, sesgará el análisis posterior. En este caso, por problemas con la visualización del heatmap, no se puede decir a ciencia cierta cuales son las diferencias en la expresión génica según los métodos aplicados. De la misma manera, los criterios que hemos establecido para la selección de genes diferencialmente expresados o el cut-off que hemos impuesto a la hora de seleccionar los términos GO diferencialmente presentes. El conjunto de estas decisiones "objetivas" determina el resultado del análisis.

Una limitación es que esta técnica debe de ser comprobada con algún otro método, como podrían ser amplificaciones de ácidos nucleicos mediante distintas variantes de la técnica PCR. En mi caso, no he podido acceder al estudio completo, por lo que no se puede saber si las variaciones identificadas se han podido comprobar.