Microarray_skin_cells

Guadalupe Rivera Torruco

4/16/2021

Analysis of Expression profiling by array

Series: GSE10433 Organism: Human Tissue: Human skin Overall design: 6 baseline/before isotretinoin and 6 after 1-week isotretinoin treatment. Platform: Affymetrix Human Genome U133A 2.0 Array GPL571

```
options(warn=-1)
```

```
packages <- c("BiocManager",
              "grid",
              "gridExtra",
              "RColorBrewer")

for (i in packages){
  if(!is.element(i, .packages(all.available = TRUE))){
    install.packages(i)
  }
  library(i, character.only = TRUE)
}
```

```
packages_bioconductor <- c("limma",
                           "affy",
                           "genefilter",
                           "oligo",
                           "GEOquery",
                           "hgu133a.db")

#Check Bioconductor packages
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install(packages_bioconductor, force = TRUE, ask = FALSE)
```

```
## package 'limma' successfully unpacked and MD5 sums checked
## package 'affy' successfully unpacked and MD5 sums checked
## package 'genefilter' successfully unpacked and MD5 sums checked
## package 'oligo' successfully unpacked and MD5 sums checked
## package 'GEOquery' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\lupit\AppData\Local\Temp\RtmpK4YOD4\downloaded_packages
##
##   There are binary versions available but the source versions are later:
##       binary source needs_compilation
## cli    3.2.0  3.3.0              TRUE
## dplyr  1.0.8  1.0.9              TRUE
```

```
## igraph  1.3.0  1.3.1              TRUE
## tibble  3.1.6  3.1.7              TRUE
```

```
multi_library <- function(packages){
  for(i in packages){
    library(i, character.only = TRUE)
  }
}

multi_library(packages_bioconductor)
```

```
knitr::opts_chunk$set(fig.width=12, fig.height=8, fig.align = "center") #check it is not working
knitr::opts_knit$set(global.par = TRUE) #check it is not working
```

Affimetryx platform outputd .CEL files

```
dir_files <- "C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/"
celfiles <-list.celfiles(dir_files)
celfiles
```

```
##  [1] "control_1.CEL" "control_2.CEL" "control_3.CEL" "control_4.CEL"
##  [5] "control_5.CEL" "control_6.CEL" "trat_1.CEL"    "trat_2.CEL"
##  [9] "trat_3.CEL"    "trat_4.CEL"    "trat_5.CEL"    "trat_6.CEL"
```

Once having the file names, we create the absolute paths joining the link to .cel files (dir_files) and its names (celfiles), then we read it with the Affy function read.celfiles(link)

```
rawData <- read.celfiles(paste(dir_files, celfiles, sep=""))
```

```
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/control_1.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/control_2.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/control_3.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/control_4.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/control_5.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/control_6.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/trat_1.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/trat_2.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/trat_3.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/trat_4.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/trat_5.CEL
## Reading in : C:/Users/lupit/OneDrive/Microarray_skin_cells/Data/trat_6.CEL
```

```
#Inspect class
class(rawData)
```

```
## [1] "ExpressionFeatureSet"
## attr(,"package")
## [1] "oligoClasses"
```

```
rawData
```

```
## ExpressionFeatureSet (storageMode: lockedEnvironment)
## assayData: 535824 features, 12 samples
##   element names: exprs
## protocolData
##   rowNames: control_1.CEL control_2.CEL ... trat_6.CEL (12 total)
##   varLabels: exprs dates
##   varMetadata: labelDescription channel
## phenoData
```

```
##   rowNames: control_1.CEL control_2.CEL ... trat_6.CEL (12 total)
##   varLabels: index
##   varMetadata: labelDescription channel
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: pd.hg.u133a.2
```

```
head(intensity(rawData))
```

```
##   control_1.CEL control_2.CEL control_3.CEL control_4.CEL control_5.CEL
## 1            76            72           110           106            96
## 2          9210         10506         12693         19832         14205
## 3            81            99           138           140           114
## 4         10195         11216         13127         20022         14584
## 5            92            89           160           128           111
## 6            57            68            98           107            78
##   control_6.CEL trat_1.CEL trat_2.CEL trat_3.CEL trat_4.CEL trat_5.CEL
## 1           113         87         57         85        101         89
## 2         17550       6080         75      15048      17102       5674
## 3           141         69      12109        127        150         93
## 4         17413       6075        103      16772      18087       6061
## 5           136        115      12815        114        143        260
## 6           107         59         74        102         81         61
##   trat_6.CEL
## 1        113
## 2      13041
## 3        124
## 4      14267
## 5        302
## 6         71
```

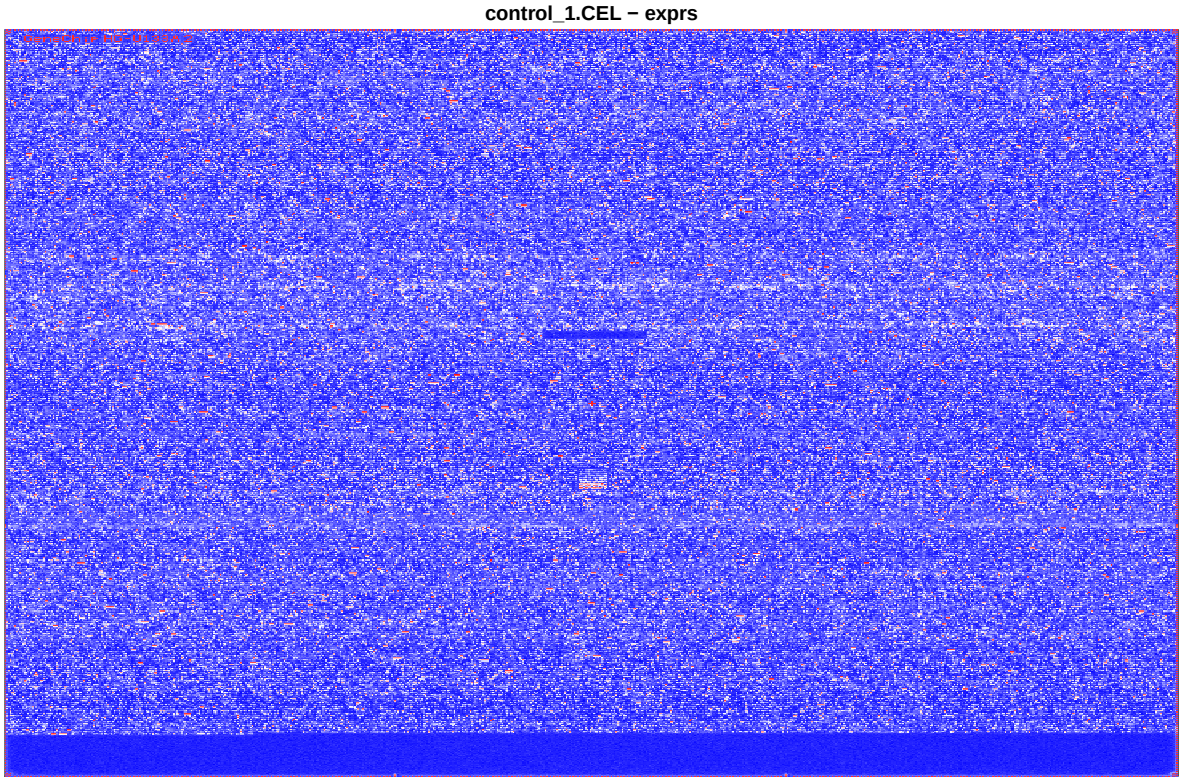
```
ph = rawData@phenoData
ph@data[,1] = c("control_1", "control_2", "control_3", "control_4", "control_5", "control_6", "tx_1", "
#Names of the columns in varLabels: index
ph$index
```

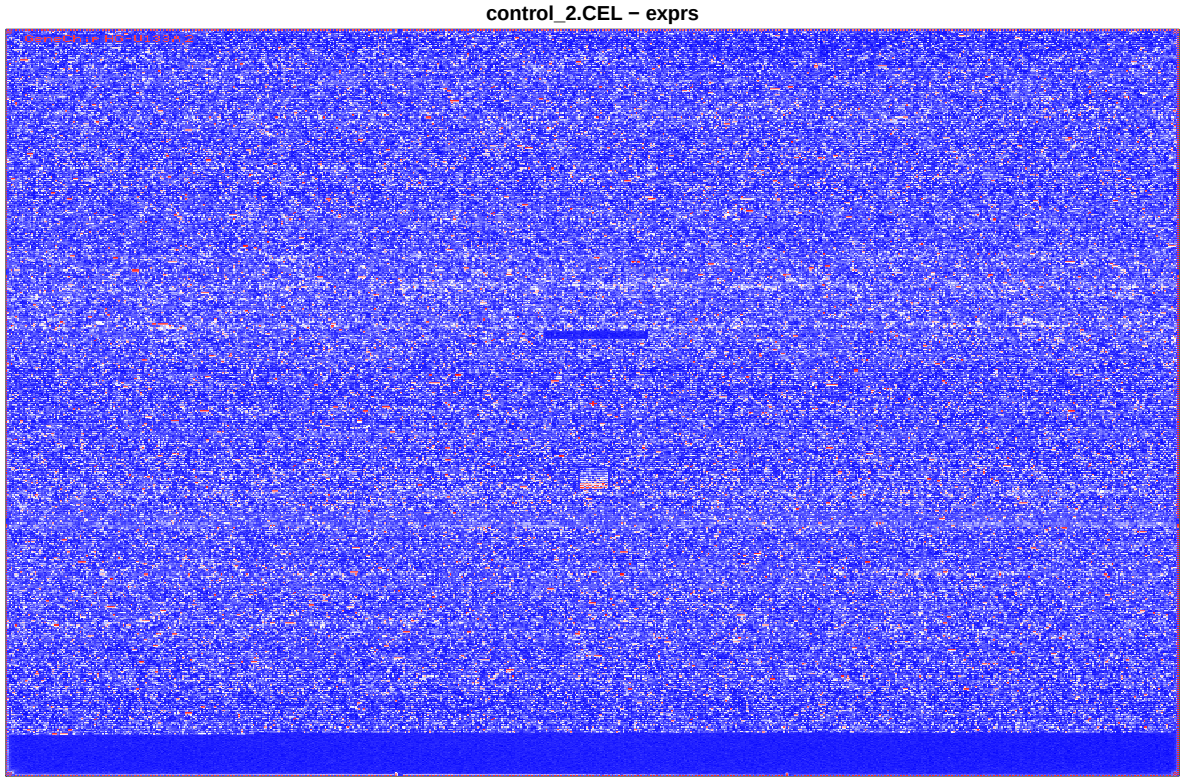
```
##  [1] "control_1" "control_2" "control_3" "control_4" "control_5" "control_6"
##  [7] "tx_1"      "tx_2"      "tx_3"      "tx_4"      "tx_5"      "tx_6"
```

```
#Look all data in the dataframe
ph@data
```

```
##                   index
## control_1.CEL control_1
## control_2.CEL control_2
## control_3.CEL control_3
## control_4.CEL control_4
## control_5.CEL control_5
## control_6.CEL control_6
## trat_1.CEL         tx_1
## trat_2.CEL         tx_2
## trat_3.CEL         tx_3
## trat_4.CEL         tx_4
## trat_5.CEL         tx_5
## trat_6.CEL         tx_6
```

```
op = par(mfrow = c(1,2)) #NOT WORKINGG!!!
for (i in 1:2){oligo::image(rawData,which=i,main=ph@data$index[i])}
```

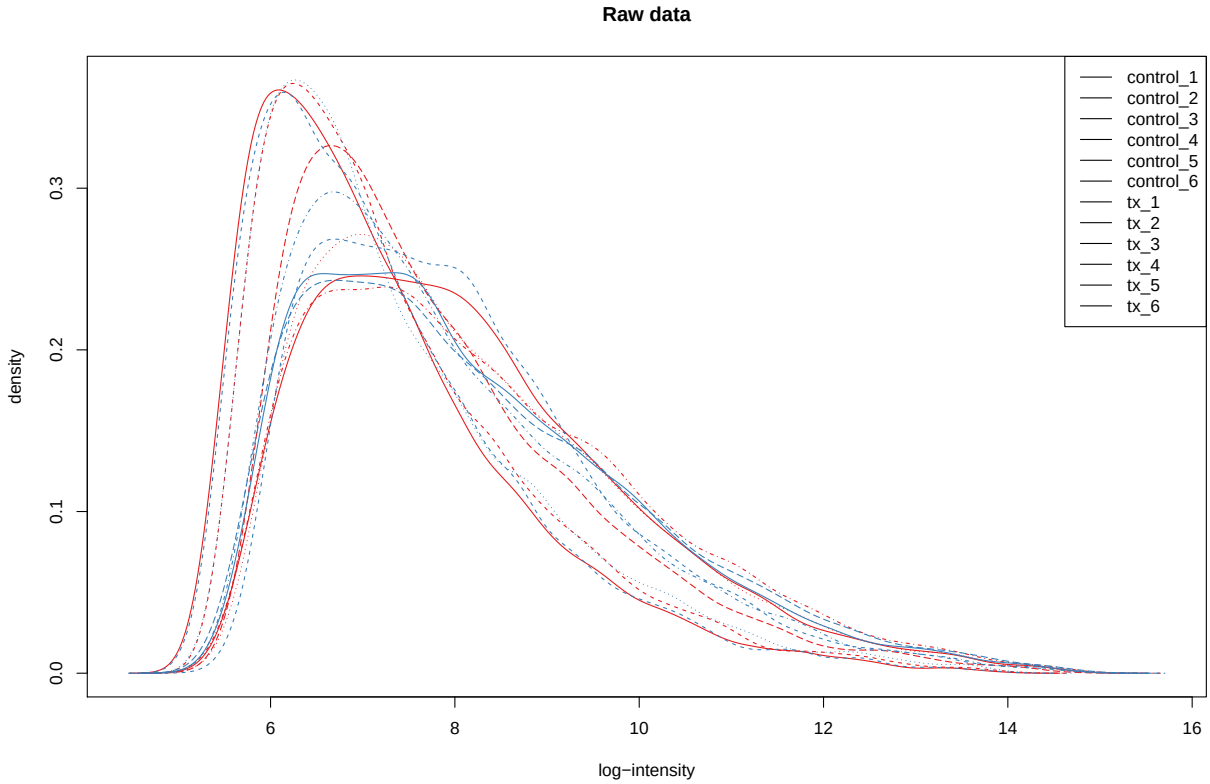




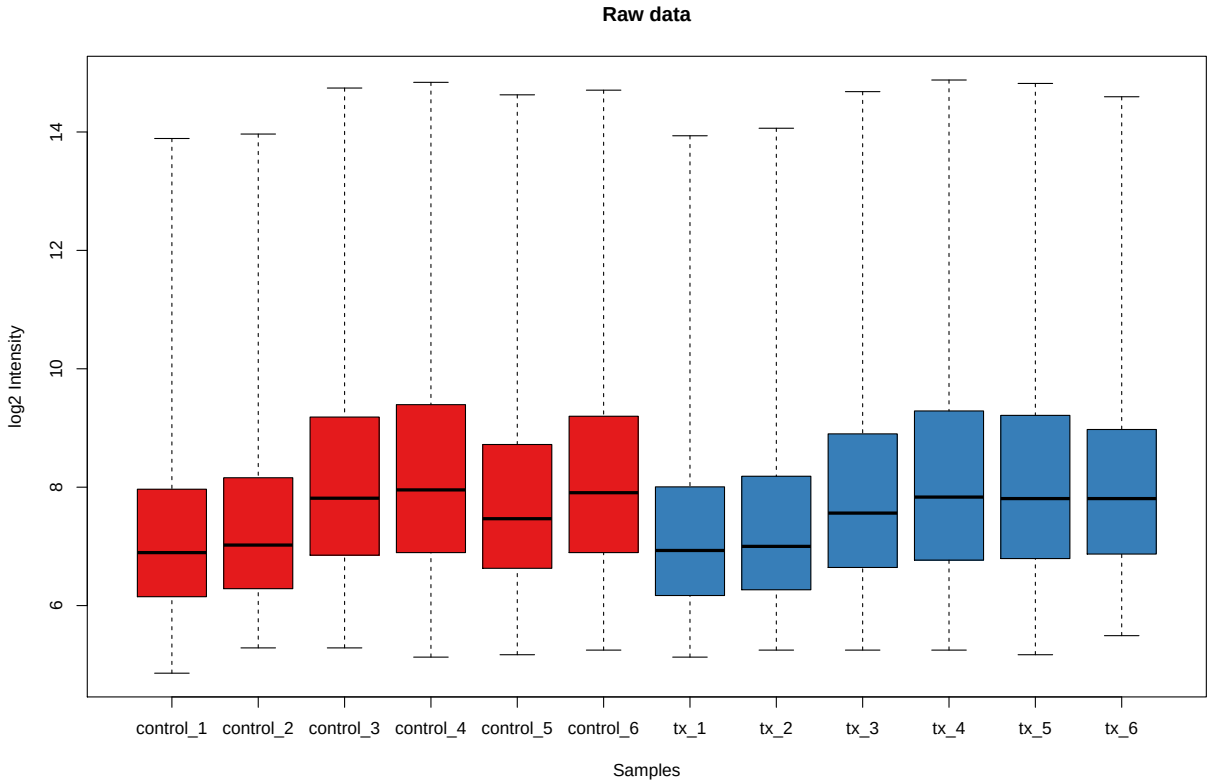

```
#par(mar = c(4, 4, .1, .1))
#image(rawData[,1])
#image(rawData[,2])
ph@data[,1]
```

```
##  [1] "control_1" "control_2" "control_3" "control_4" "control_5" "control_6"
##  [7] "tx_1"      "tx_2"      "tx_3"      "tx_4"      "tx_5"      "tx_6"
```

```
usr.col<-brewer.pal(9,"Set1")
mycols<-rep(usr.col, each=6)
hist(rawData, col=mycols, main="Raw data", target="core")
legend("topright", ph@data[,1], lty=rep(1,length(celfiles)),col=mycols, cex=0.2))
```



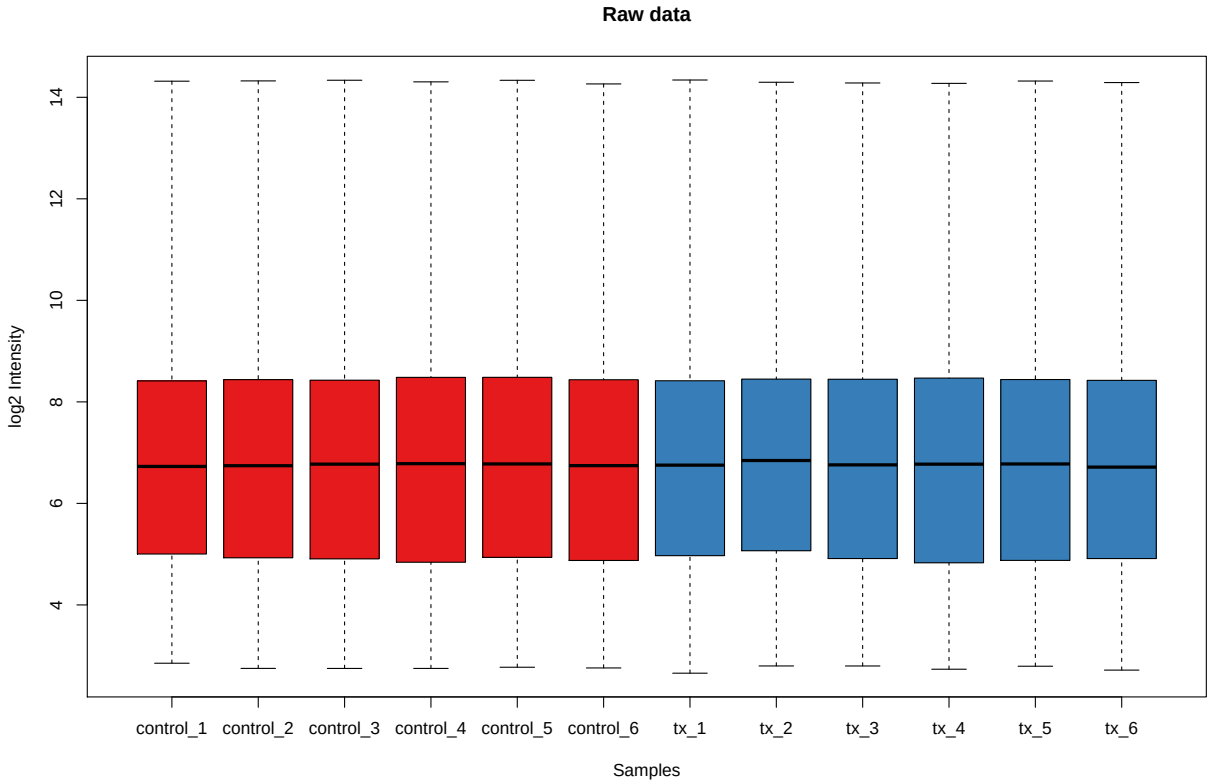

```
boxplot(rawData, xex=0.2, ylab="log2 Intensity", xlab="Samples",names=ph@data[,1],
        col=mycols, main="Raw data")
```




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

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

```
boxplot(eset, xex=0.2, ylab="log2 Intensity", xlab="Samples",names=ph@data[,1],
        col=mycols, main="Raw data")
```




```
eDat<-exprs(eset)
#look dimensions
dim(eDat)
```

```
## [1] 22277    12
```

Filter out genes with low expression with genefilter package

```
f1<-pOverA(0.5, log2(100))

#Intensity difference over log2(1.5)
f2<-function(x) (diff(range(x, na.rm=T))>log2(1.5))
ff<-filterfun(f1,f2)

#file with filtered data
index<-genefilter(eDat, ff)
sum(index)
```

```
## [1] 8238
```

```
eDatSet <- eDat[index,]
dim(eDatSet)
```

```
## [1] 8238   12
```

```
head(eDatSet)
```

```
##           control_1.CEL control_2.CEL control_3.CEL control_4.CEL control_5.CEL
## 121_at         8.294466      7.776913      7.760742      7.794467      7.573156
## 1294_at        7.027754      7.287919      6.872031      7.282931      7.397765
## 1320_at        7.261481      7.033216      7.347592      6.530578      6.651550
```

```
## 1438_at        7.820403      8.037049      7.821089      7.958800      8.733903
## 1598_g_at      9.939454     10.082396      9.973424     10.176153     10.667309
## 160020_at      8.563741      8.729115      7.816575      8.749889      8.631371
##           control_6.CEL trat_1.CEL trat_2.CEL trat_3.CEL trat_4.CEL trat_5.CEL
## 121_at         7.913777   8.283952   7.695757   7.735957   7.794047   7.672510
## 1294_at        6.933582   7.118586   7.312887   7.477681   7.540600   7.301958
## 1320_at        7.146359   7.569155   7.401021   6.563843   6.478267   6.909067
## 1438_at        7.822974   8.108310   8.054874   7.913167   8.314888   8.472371
## 1598_g_at      9.975629  10.144710   9.781344  10.528197  10.454364  10.499982
## 160020_at      7.835796   8.603519   8.152809   8.361701   8.455506   8.425206
##           trat_6.CEL
## 121_at      7.887497
## 1294_at     6.783170
## 1320_at     7.005180
## 1438_at     7.913699
## 1598_g_at  10.003061
## 160020_at   7.843085
```

Then, we calculate genes differentially expressed between conditions, using tools from limma package

```
TS<-gl(2,6, labels=c("control","treatment"))
#linear model
design<-model.matrix(~TS)
colnames(design)<-c("control", "contrast")
design
```

```
##    control contrast
## 1        1        0
## 2        1        0
## 3        1        0
## 4        1        0
## 5        1        0
## 6        1        0
## 7        1        1
## 8        1        1
## 9        1        1
## 10       1        1
## 11       1        1
## 12       1        1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$TS
## [1] "contr.treatment"
```

After creating our lineal model, we train the eDatset with it. Then, apply Bayes algorithm to obtain statistical values of the predictions. Finally, we can filter only the genes that are statistically different p>0.05 and with a limit fold change (lfc) of log2(2) to ensure high confidence in the selected genes.

```
#Fitting model
fit <- lmFit(eDatSet, design) #estimaciÃ³n de los parametros de ajuste lineal
# Bayesin statistics
fit <- eBayes(fit)
# keep genes with p>0.05 and lfc>2
top_genes <-topTable(fit, coef="contrast", number=nrow(eDatSet), adjust="fdr", p.value=0.05, lfc=log2(2)
dim(top_genes)
```

```
## [1] 2 6
```

```
top_genes
```

```
##              logFC   AveExpr        t      P.Value   adj.P.Val         B
## 218960_at 1.180523  7.110431 7.475536 1.983317e-06 0.008207179 0.4736100
## 212531_at 2.699712 10.414618 7.472652 1.992517e-06 0.008207179 0.4722941
```

Finally we retrieve the names of the genes with the annotation package "hgu133a.db"

```
rma <- top_genes
probes = row.names(rma)
symbols <- unlist(mget(probes, hgu133aSYMBOL, ifnotfound = NA))
Entrez_ID <-  unlist(mget(probes, hgu133aENTREZID, ifnotfound = NA))
rma <- cbind(symbols, Entrez_ID, rma)
```

```
rma
```

```
##           symbols Entrez_ID    logFC   AveExpr        t      P.Value
## 218960_at TMPRSS4     56649 1.180523  7.110431 7.475536 1.983317e-06
## 212531_at    LCN2      3934 2.699712 10.414618 7.472652 1.992517e-06
##             adj.P.Val         B
## 218960_at 0.008207179 0.4736100
## 212531_at 0.008207179 0.4722941
```

```
rn<-rownames(top_genes)
dat.s<-eDatSet[rn,]
rownames(dat.s) <- rma$symbols
```

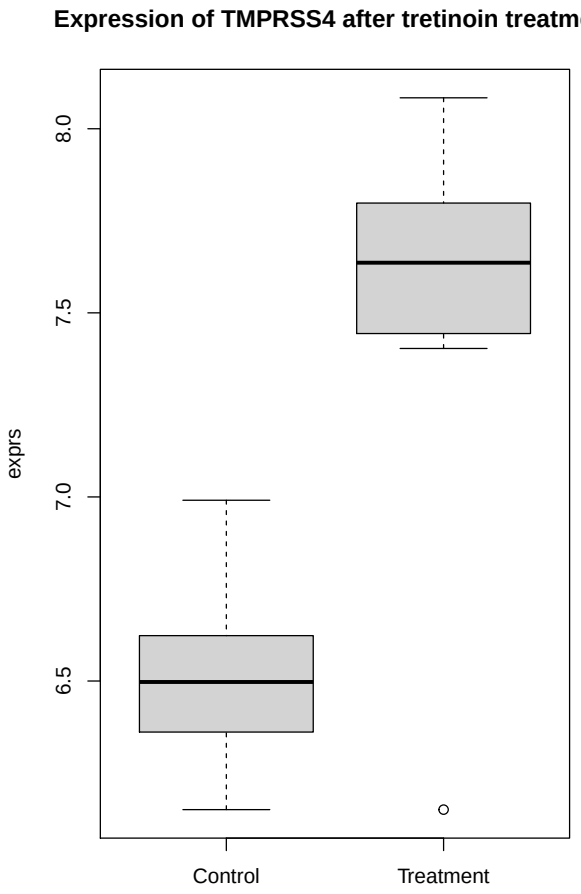
```
dat.s
```

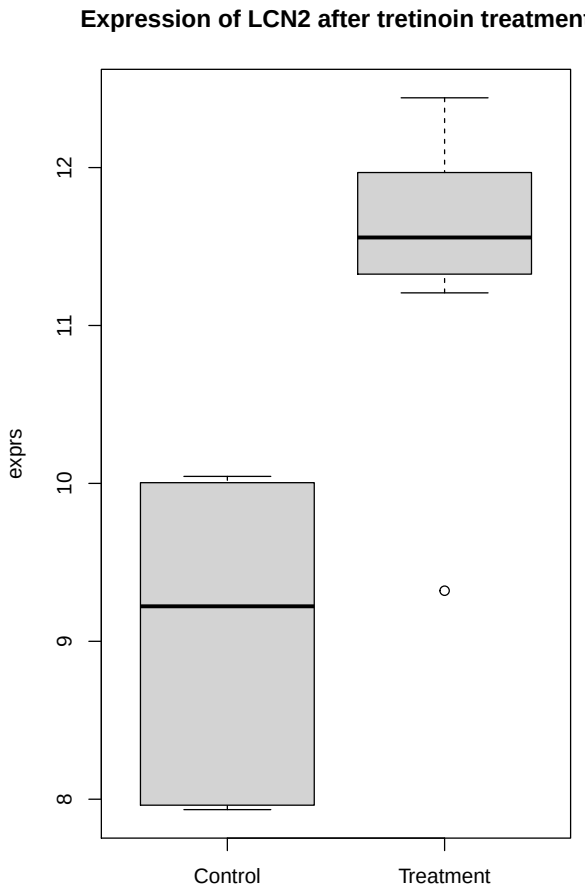
```
##         control_1.CEL control_2.CEL control_3.CEL control_4.CEL control_5.CEL
## TMPRSS4      6.361347      6.407723      6.623239      6.991007      6.587037
## LCN2         7.934040     10.044377      9.122937     10.004577      7.962181
##         control_6.CEL trat_1.CEL trat_2.CEL trat_3.CEL trat_4.CEL trat_5.CEL
## TMPRSS4      6.150662    7.48428   8.084061   7.786337   7.636178   7.403233
## LCN2         9.320458   11.55739  12.441891  11.801769  12.135501  11.206623
##         trat_6.CEL
## TMPRSS4   7.810064
## LCN2     11.443673
```

We can creato some simple plots with R base boxplot, like this:

```
par(mfrow = c(1,2))
boxplot(dat.s[1,1:6], dat.s[1,6:12], names = c("Control", "Treatment"), ylab="exprs")
title("Expression of TMPRSS4 after tretinoin treatment")

boxplot(dat.s[2,1:6], dat.s[2,6:12], names = c("Control", "Treatment"), ylab="exprs")
title("Expression of LCN2 after tretinoin treatment")
```


Expression of TMPRSS4 after tretinoin treatment
exprs
8.0
7.5
7.0
6.5
Control
Treatment


Expression of LCN2 after tretinoin treatment
exprs
12
11
10
9
8
Control
Treatment