

Affymetrix-Limma Workflow

Asad

4/20/2023

1. Calling Libraries

```
library(affy)

## 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, 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

## 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")'.

library(oligo)

## Loading required package: oligoClasses

## Welcome to oligoClasses version 1.56.0

## 
## Attaching package: 'oligoClasses'
```

```

## The following object is masked from 'package:affy':
##
##      list.celfiles

## Loading required package: Biostrings

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##      expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

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

## Loading required package: XVector

## Loading required package: GenomeInfoDb

##
## Attaching package: 'Biostrings'

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

## =====

## Welcome to oligo version 1.58.0

## =====

##
## Attaching package: 'oligo'

## The following objects are masked from 'package:affy':
##
##      intensity, MAplot, mm, mm<-, mmindex, pm, pm<-, pmindex,
##      probeNames, rma

```

```

library(GEOquery)

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

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

library(Biobase)
library(tidyr)

## 
## Attaching package: 'tidyverse'

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

library(splitstackshape)
library(arrayQualityMetrics)
library(dplyr)

## 
## Attaching package: 'dplyr'

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

## The following objects are masked from 'package:Biostrings':
##       collapse, intersect, setdiff, setequal, union

## The following object is masked from 'package:GenomeInfoDb':
##       intersect

## The following object is masked from 'package:XVector':
##       slice

## The following objects are masked from 'package:IRanges':
##       collapse, desc, intersect, setdiff, slice, union

## The following objects are masked from 'package:S4Vectors':
##       first, intersect, rename, setdiff, setequal, union

## The following object is masked from 'package:Biobase':
##       combine

```

```

## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching packages ----- tidyverse 1.3.1 --

## v ggplot2 3.4.2      v purrr    1.0.1
## v tibble   3.2.1      v stringr  1.5.0
## v readr    2.1.2      vforcats  0.5.1

## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::collapse()  masks Biostrings::collapse(), IRanges::collapse()
## x dplyr::combine()   masks Biobase::combine(), BiocGenerics::combine()
## x purrr::compact()   masks XVector::compact()
## x dplyr::desc()      masks IRanges::desc()
## x tidyr::expand()    masks S4Vectors::expand()
## x dplyr::filter()   masks stats::filter()
## x dplyr::first()    masks S4Vectors::first()
## x dplyr::lag()       masks stats::lag()
## x ggplot2::Position() masks BiocGenerics::Position(), base::Position()
## x purrr::reduce()    masks IRanges::reduce()
## x dplyr::rename()   masks S4Vectors::rename()
## x dplyr::slice()    masks XVector::slice(), IRanges::slice()
## x dplyr::summarize() masks oligo::summarize()

library(limma)

##
## Attaching package: 'limma'

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

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

library(annotate)

## Loading required package: AnnotationDbi

```

```

##  

## Attaching package: 'AnnotationDbi'  

##  

## The following object is masked from 'package:dplyr':  

##  

##     select  

##  

## Loading required package: XML  

##  

library(AnnotationDbi)  

library(org.Hs.eg.db)

##  

##  

library(hgu133plus2.db)

##  

##  

library(EnhancedVolcano)

## Loading required package: ggrepel

## Registered S3 methods overwritten by 'ggalt':  

##   method           from  

##   grid.draw.absoluteGrob  ggplot2  

##   grobHeight.absoluteGrob ggplot2  

##   grobWidth.absoluteGrob ggplot2  

##   grobX.absoluteGrob    ggplot2  

##   grobY.absoluteGrob    ggplot2

```

1. Calling Libraries

```
setwd("D:/CancerData/Data")
```

Read celFiles

```

celFiles <- list.celfiles()  

affyRaw <- read.celfiles(celFiles)

## Loading required package: pd.hg.u133.plus.2

## Loading required package: RSQLite

## Loading required package: DBI

## Platform design info loaded.

```

```

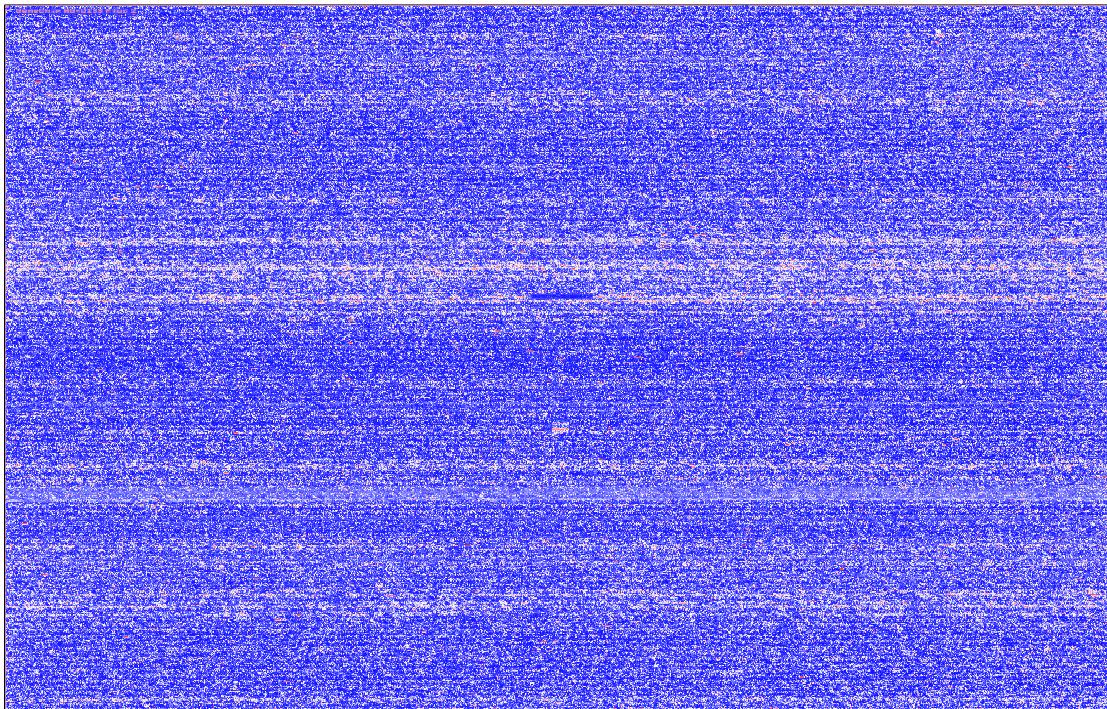
## Reading in : GSM869667.CEL
## Reading in : GSM869668.CEL
## Reading in : GSM869669.CEL
## Reading in : GSM869670.CEL
## Reading in : GSM869671.CEL
## Reading in : GSM869672.CEL
## Reading in : GSM869673.CEL
## Reading in : GSM869674.CEL
## Reading in : GSM869675.CEL
## Reading in : GSM869676.CEL
## Reading in : GSM869678.CEL
## Reading in : GSM869679.CEL

#Generate pseudo-image of chip intensity for individual sample (CEL file)

image(affyRaw[, 1])

```

GSM869667.CEL – exprs



```

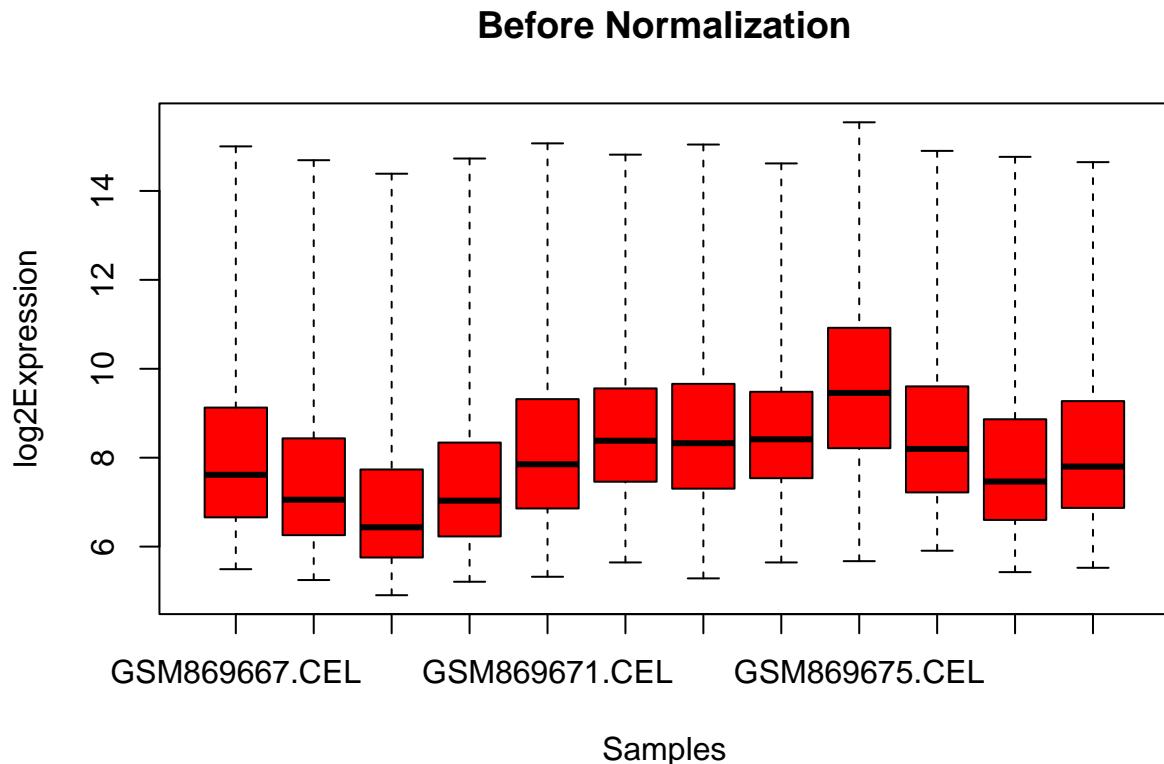
#Check log2 Transformation and Apply if Required # [log2 normalized expression value should range between
0 to 16] affyRaw<- log2(exprs(affyRaw)) head(affyRaw, 10)

#Check Quality of Data Before Normalization arrayQualityMetrics(affyRaw, outdir="quality_assesment",
force = T) browseURL(file.path("quality_assesment", "index.html"))

#Check probe intensity plot before normalization

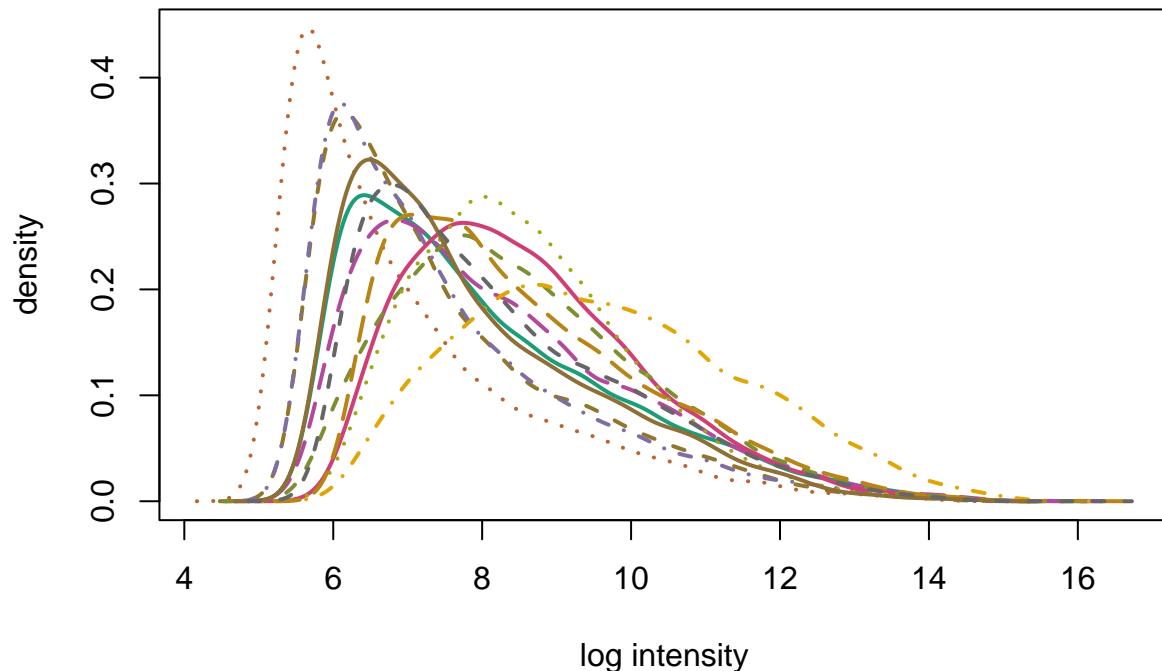
```

```
x<-boxplot(affyRaw, xlab='Samples', ylab='log2Expression', col='Red',
            main= 'Before Normalization')
```



```
hist(affyRaw, lwd=2,xlab='log intensity',
      main="CEL file densities before normalisation")
```

CEL file densities before normalisation



RMA Normalization Method

```
normalized.data <- rma(affyRaw)

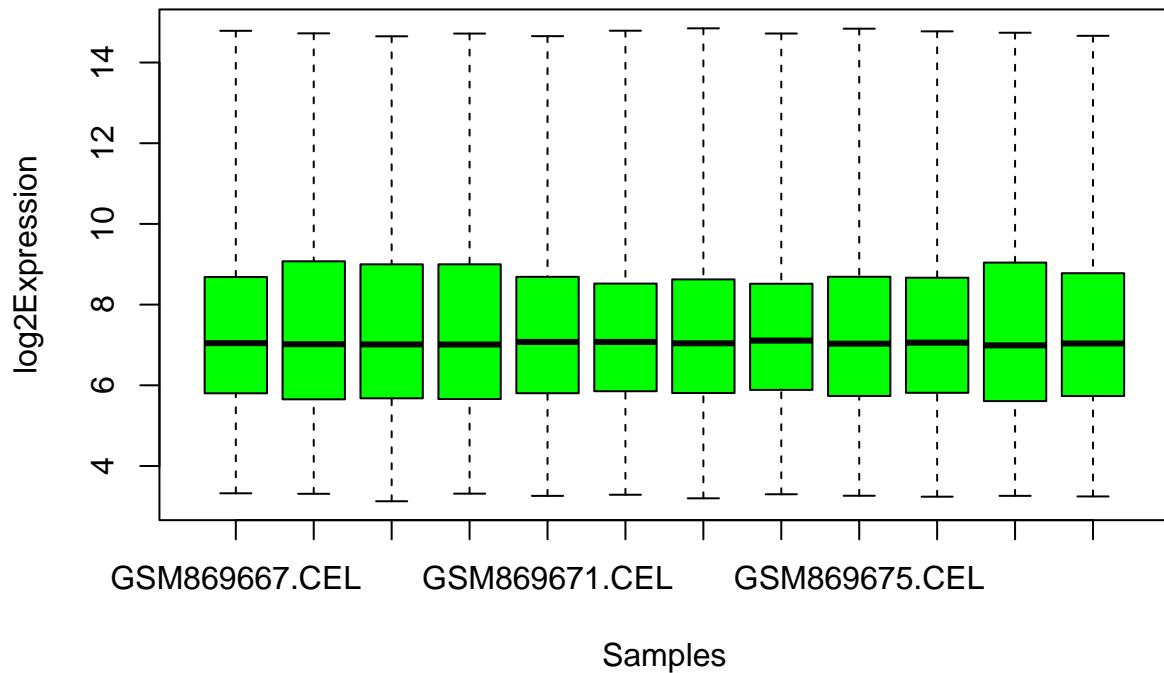
## Background correcting
## Normalizing
## Calculating Expression

#Check Quality of Data After RMA Normalization
arrayQualityMetrics(normalized.data, outdir="quality_assesment2", force = T)
browseURL(file.path("quality_assesment2", "index.html"))

#Check probe intensity plot after normalization

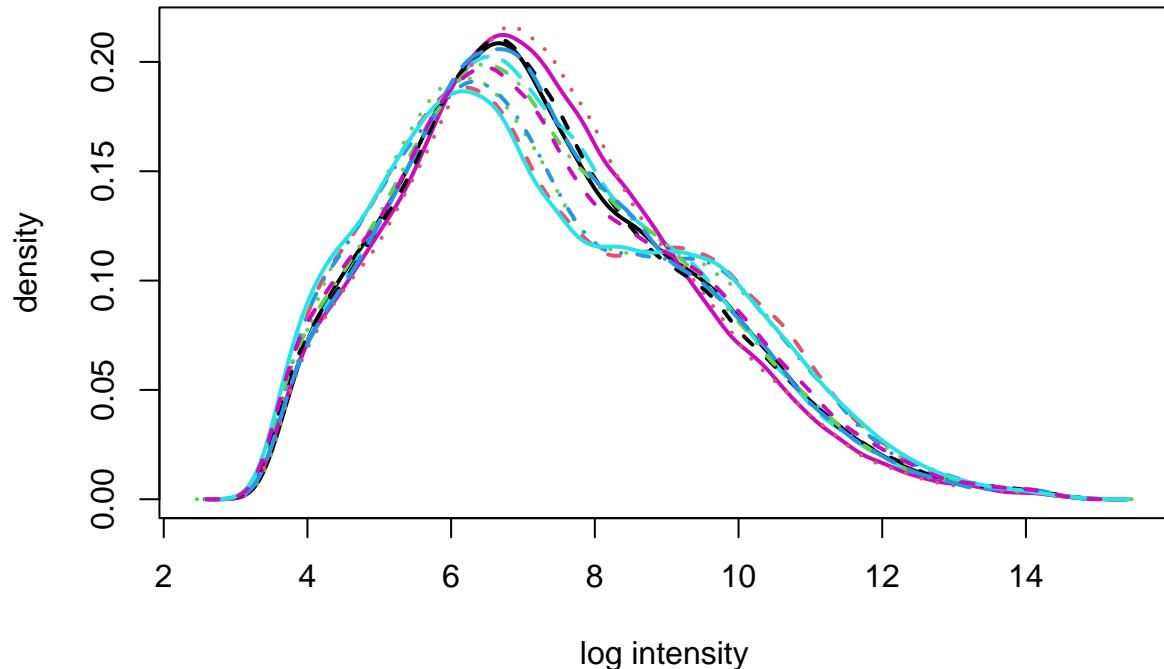
y<-boxplot(normalized.data, xlab='Samples', ylab='log2Expression', col='Green',
            main= 'After RMA Normalization')
```

After RMA Normalization



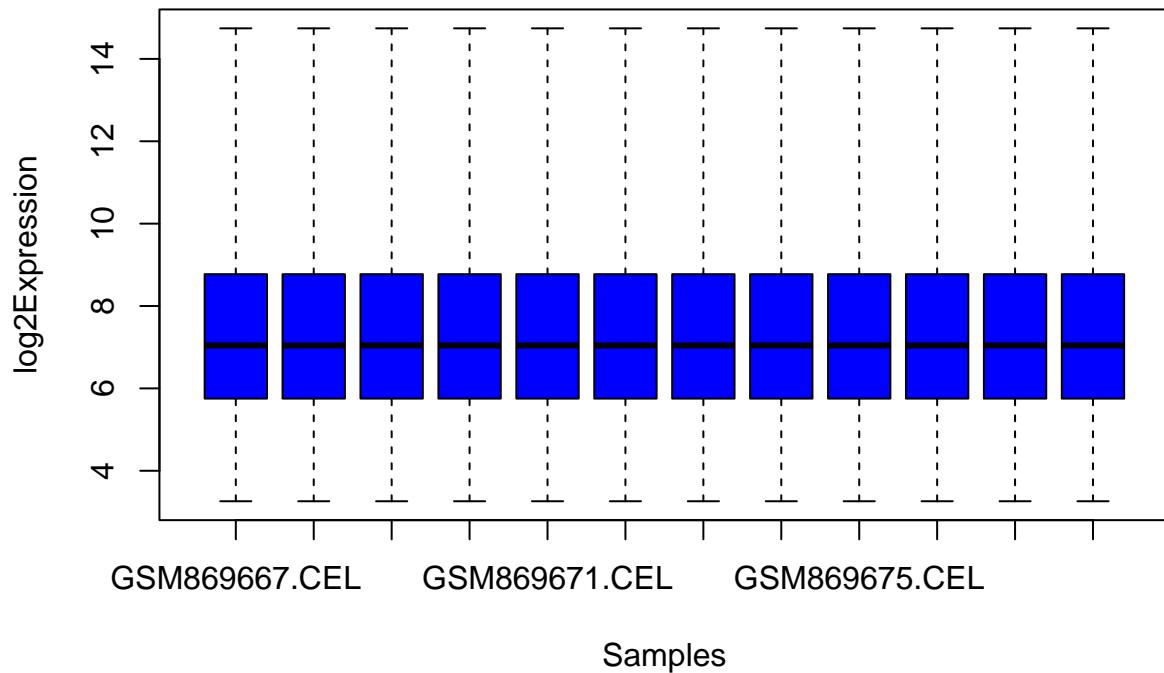
```
hist(normalized.data, lwd=2,xlab='log intensity',
  main="CEL file densities after RMA normalisation")
```

CEL file densities after RMA normalisation



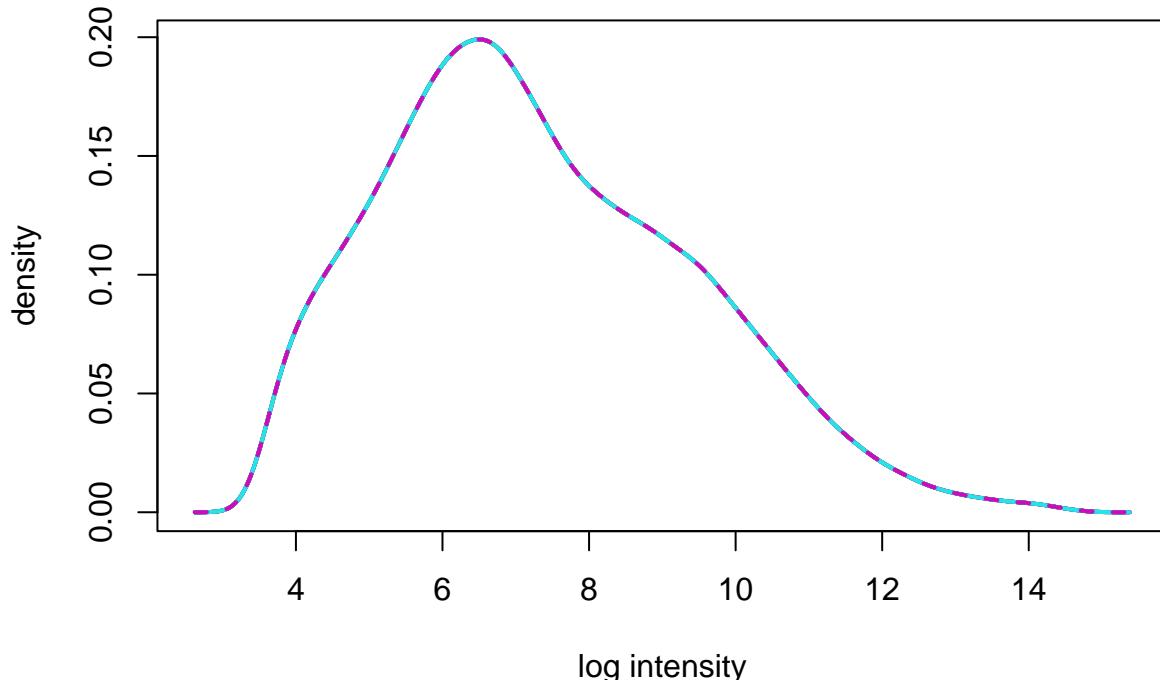
```
#Quantile normalization and assessment  
Q_normalised <- normalize(normalized.data,method='quantile')  
z<-boxplot(Q_normalised, xlab='Samples', ylab='log2Expression',  
           col='Blue',main= 'After Quantile Normalization')
```

After Quantile Normalization



```
hist(Q_normalised, lwd=2,xlab='log intensity',
      main="CEL file densities after quantile normalisation")
```

CEL file densities after quantile normalisation



```
#Naming the Probe IDs to Gene IDs
```

```
ID <- featureNames(normalized.data)
Symbol <- getSYMBOL(ID, 'hg133plus2.db')
fData(normalized.data) <- data.frame(Symbol=Symbol)
```

```
#Perform DEG Analysis
```

```
Groups<- factor(c("Normal", "Normal", "Normal", "Normal", "Normal",
                  "Normal", "Tumor", "Tumor", "Tumor", "Tumor", "Tumor", "Tumor" ))
design<- model.matrix(~Groups)
head(design)
```

```
##      (Intercept) GroupsTumor
## 1              1         0
## 2              1         0
## 3              1         0
## 4              1         0
## 5              1         0
## 6              1         0
```

```
colnames(design)<- c("Normal", "Tumor")
fit<- lmFit(normalized.data, design)
fit2<- eBayes(fit)
options(digits = 2)
topTable(fit2)
```

```

##                               Symbol Normal   Tumor AveExpr      F P.Value adj.P.Val
## 201492_s_at          RPL41     15 -0.0056     15 38135 1.2e-24  2.4e-20
## AFFX-hum_alu_at    <NA>     15  0.1403     15 33401 2.8e-24  2.4e-20
## 203107_x_at         RPS2     14  0.1608     14 32912 3.1e-24  2.4e-20
## 201429_s_at         RPL37A    14  0.1429     15 32788 3.2e-24  2.4e-20
## 213477_x_at         EEF1A1    14  0.1188     14 32410 3.4e-24  2.4e-20
## 212988_x_at         ACTG1    14 -0.0036     14 31303 4.3e-24  2.4e-20
## 211983_x_at         ACTG1    14 -0.0049     14 30881 4.7e-24  2.4e-20
## 211970_x_at         ACTG1    14 -0.0393     14 30793 4.8e-24  2.4e-20
## 224585_x_at         ACTG1    14  0.0081     14 30470 5.1e-24  2.4e-20
## 201550_x_at         ACTG1    14 -0.0480     14 30251 5.3e-24  2.4e-20

```

```

DEGs<- topTable(fit2, coef = 2, n=Inf, adjust="BH")
write.csv(DEGs, file="DEGs.csv")

```

```
#Stratify
```

```

Significant_DEGs<- filter(DEGs,logFC>1|logFC< -1, adj.P.Val<0.7)
Significant_UP<- filter(Significant_DEGs,logFC> 1)
Significant_Down<- filter(Significant_DEGs,logFC< -1)

```

```
#Create basic volcano plot
```

```

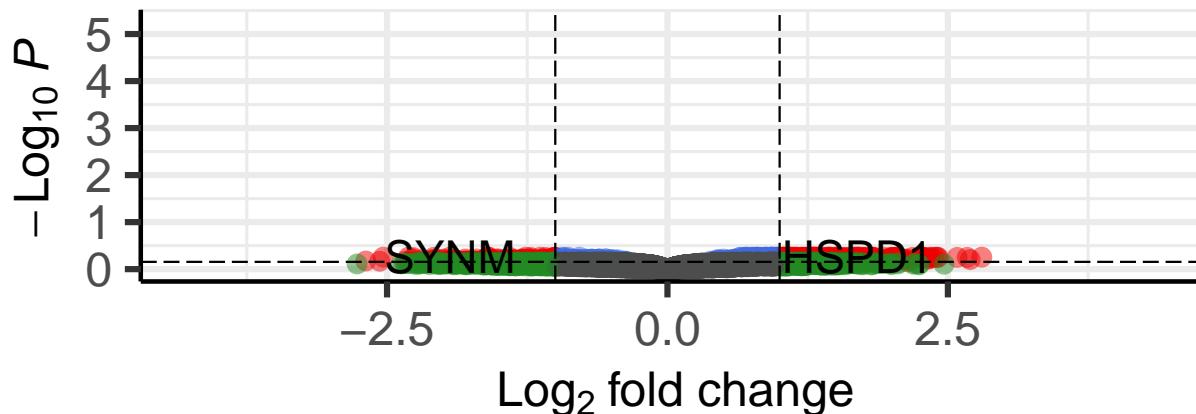
EnhancedVolcano(DEGs,
  lab = DEGs$Symbol,
  x = 'logFC',
  y = 'adj.P.Val',
  pCutoff = 0.7,
  FCcutoff = 1,
  pointSize = 3.0,
  labSize = 6.0)

```

Volcano plot

EnhancedVolcano

● NS ● Log₂ FC ● p-value ● p-value and log₂ FC



total = 54675 variables

```
#Visualizing DEG of Interest
```

```
DEG_of_Interest <- topTable(fit2, coef = 2, number=Inf, lfc=1, p.value = 1)
```

```
#Generating Heatmap
```

```
Genes_of_interest <- normalized.data[rownames(DEG_of_Interest),]  
heatmap(exprs(Genes_of_interest))
```

