USC Department Of Translational Genomics
Keck Medicine of USC

TRGN 527: Applied Data Science and Bioinformatics

UNIT III. Supervised Statistical Tests

Week 6 - Lecture 1 – Case Study Part 2

Enrique I. Velazquez Villarreal, M.D., Ph.D., M.P.H., M.S. | **Assistant Professor**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Leader of the USC Bioinformatics Core – *USC CaRE2 Health Equity Center*

David W. Craig, Ph.D. | **Professor and Vice Chair**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Co-Director, Institute of Translational Genomics

Accessing data from CEL files

- ReadAffy()
  - This commands reads all CEL or CEL.gz files into the working directory.
  - The function Read.Affy() has some interesting parameters.
  - If only some of the files of the working directory are required, the files can be specified as a character vector.
  - The samples can also be renamed by using “sampleNames” parameter.
  - The experimental design information can be provided with “phenoData” or full description in MIAME format with the description parameter.

```
TRGN599_data<-ReadAffy()

#Installing required package for Affy
TRGN599_data
```

Accessing data from CEL files

- Looking at the actual object saved in the TRGN599_data variable:

Annotating Data

```
annotation(TRGN599_data)
```

```
## [1] "htmg430pm"
```

```
cdfName(TRGN599_data)
```

```
## [1] "HT_MG-430_PM"
```

Accessing data from CEL files

- Calling the TRGN599_data variable at the first time, R tries to install the htmg430pmcdf package.
- CDF files are chip definition format files defining which probes belong to which probesets on Affymerix arrays, and they are necessary to use in any of the standard summarization methods.
- These CDF environments are available from many affy arrays, but they can also be made with the makecdfenv package if needed.
- The TRGN599_data variable is an AffyBatch object holding here 12 samples and 45,141 probes.
  - Dubbed as genes, but there is clearly some redundancy involved here.
- Since there was no information provided about the experimental design while reading the data, it has to be done now
- At the moment, the phenotype slot is rather rudimentary here; let's add the information about the genotype and the treatment of the samples:

```
TRGN599_exp_des <- pData(TRGN599_data)
TRGN599_exp_des$Genotype <- factor(rep(c("WT","dTsc1"),each=6))
TRGN599_exp_des$Stimulation <- factor(rep(c("0h","4h"),6))
TRGN599_exp_des
```

```
##                                 sample Genotype Stimulation
## GSM738363_ykr264-htmg430pm.CEL.gz      1       WT          0h
## GSM738365_ykr265-htmg430pm.CEL.gz      2       WT          4h
## GSM738367_ykr267-htmg430pm.CEL.gz      3       WT          0h
## GSM738368_ykr268-htmg430pm.CEL.gz      4       WT          4h
## GSM738370_ykr270-htmg430pm.CEL.gz      5       WT          0h
## GSM738372_ykr271-htmg430pm.CEL.gz      6       WT          4h
## GSM738373_ykr273-htmg430pm.CEL.gz      7    dTsc1          0h
## GSM738375_ykr274-htmg430pm.CEL.gz      8    dTsc1          4h
## GSM738377_ykr276-htmg430pm.CEL.gz      9    dTsc1          0h
## GSM738378_ykr277-htmg430pm.CEL.gz     10    dTsc1          4h
## GSM738379_ykr279-htmg430pm.CEL.gz     11    dTsc1          0h
## GSM738380_ykr280-htmg430pm.CEL.gz     12    dTsc1          4h
```

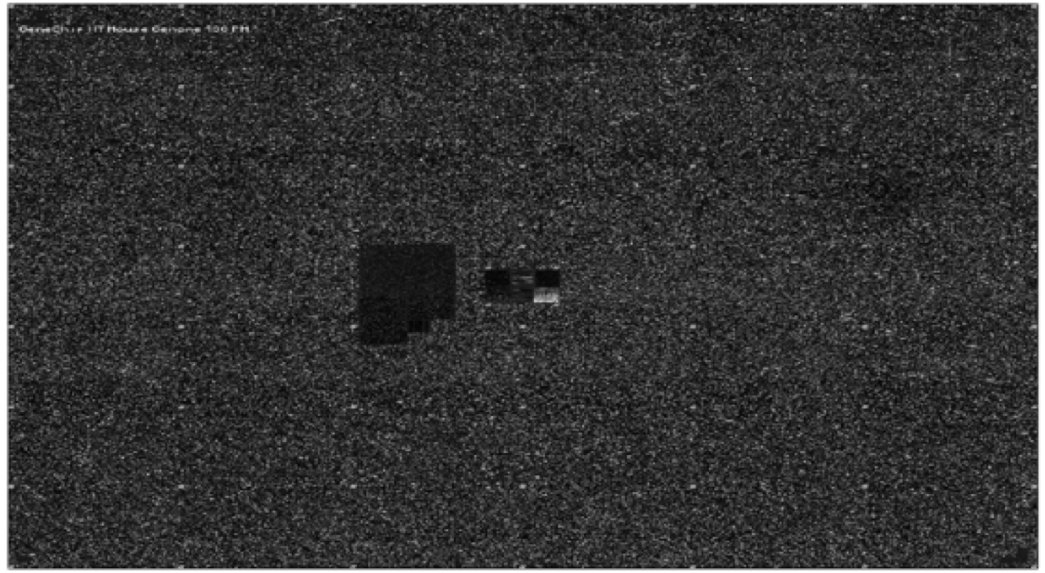
Accessing data from CEL files

- Now, let's check the data itself.
- CEL files also hold the raw images, and they can be accessed easily.
- A quick visual inspection highlights any serious technical error in the imaging process.
  - For example, if there is a gradual background gradient: it must be compensated during the preprocessing steps.

Generating Image

```
pData(TRGN599_data) <- TRGN599_exp_des
image(TRGN599_data[,1])
```

GSM738363_ykr264-htmg430pm.CEL.gz



Quality Control

- In high-throughput experiments, measuring and reporting the general quality is an important task for excluding measurements with disturbing problems.
- There are many Bioconductor packages assisting these duties.
  - Package simpleaffy provides some basic tools.
  - Package affyQCReport package generates a comprnhensive report on a set of CEL files automatically.
  - These packages are convenient wrappers around low-level functions in the affy package.

- One handy tool for checking if all the samples are similar is the RNA degradation plot
- Since there are multiple probes for individual genes on Affy platforms it can be observed how much RNA was degraded in the biological samples.
- The degradation plot lines are supposed to run in parallel.
- Samples severely deviating from the majority of the cohort should be discarded.

basic QC

```
TRGN599_deg <- AffyRNAdeg(TRGN599_data)
```

Quality Control

Summarizing Data

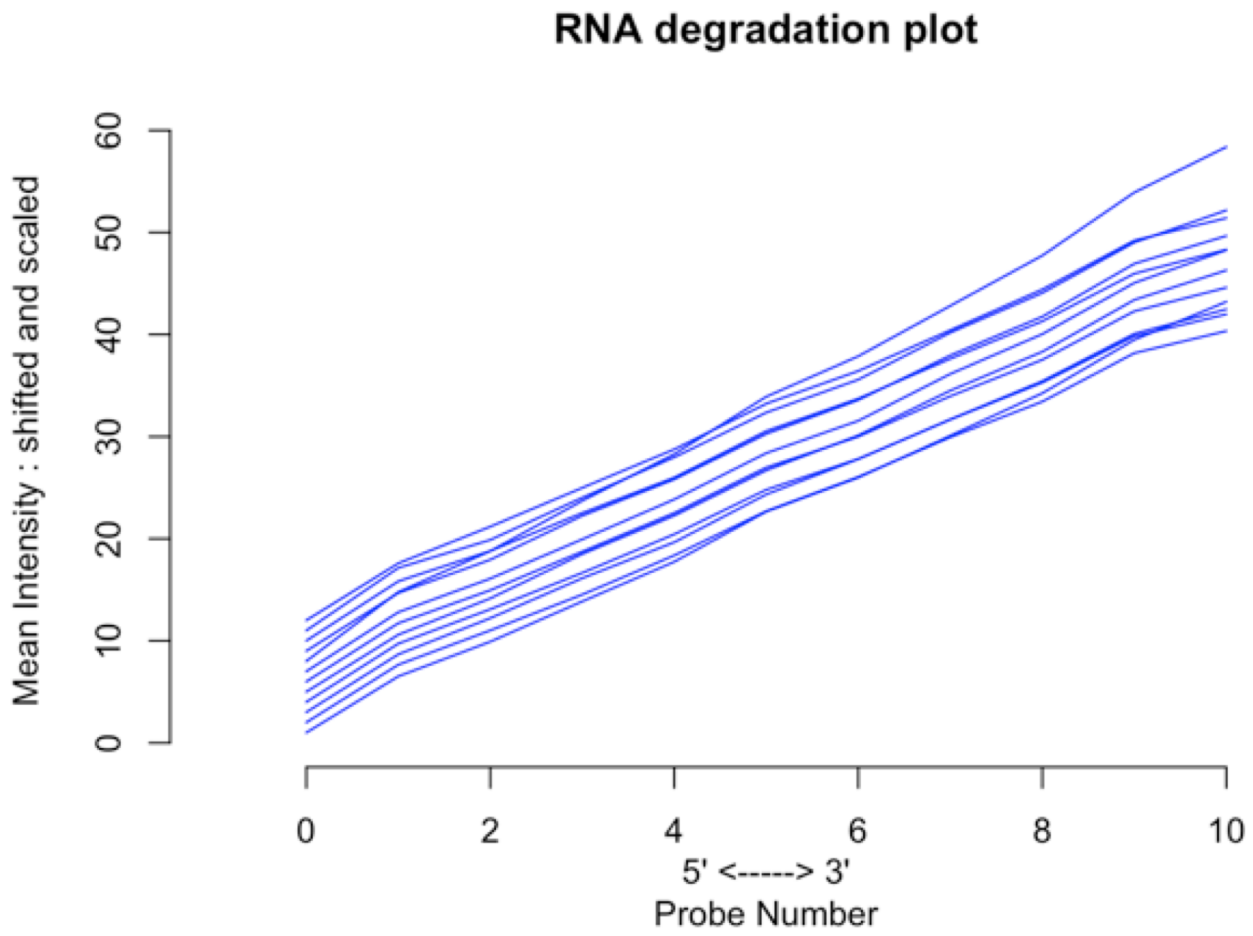
```
summaryAffyRNAdeg(TRGN599_deg)
```

```
##        GSM738363_ykr264-htmg430pm.CEL.gz GSM738365_ykr265-htmg430pm.CEL.gz
## slope                           4.15e+00                          3.81e+00
## pvalue                          2.07e-14                          2.89e-13
##        GSM738367_ykr267-htmg430pm.CEL.gz GSM738368_ykr268-htmg430pm.CEL.gz
## slope                           3.93e+00                          3.77e+00
## pvalue                          1.76e-13                          5.49e-13
##        GSM738370_ykr270-htmg430pm.CEL.gz GSM738372_ykr271-htmg430pm.CEL.gz
## slope                           4.09e+00                          3.83e+00
## pvalue                          2.29e-14                          3.10e-13
##        GSM738373_ykr273-htmg430pm.CEL.gz GSM738375_ykr274-htmg430pm.CEL.gz
## slope                           4.07e+00                          4.94e+00
## pvalue                          4.21e-14                          2.40e-14
##        GSM738377_ykr276-htmg430pm.CEL.gz GSM738378_ykr277-htmg430pm.CEL.gz
## slope                           4.03e+00                          3.80e+00
## pvalue                          6.09e-14                          2.77e-13
##        GSM738379_ykr279-htmg430pm.CEL.gz GSM738380_ykr280-htmg430pm.CEL.gz
## slope                           4.05e+00                          3.93e+00
## pvalue                          1.10e-13                          1.92e-13
```

Quality Control

Ploting Data

```
plotAffyRNAdeg(TRGN599_deg)
```



Quality Control

- Outliers can be detected by checking the overall array-to-array intensity correlations.
- The affyQCReport has an easy-to-use function to calculate and visualize this.
- Good quality arrays show high correlations.
- Naturally, even higher correlations appear in case of replicate samples.
- All array platforms contain a series of control probes.
- One set of these is the positive and negative control probes that measure the highest and lowest border signal levels.
- These control values are worth seeing if the intensity levels for these controls are in separate ranges plot on all arrays - affyQCReport has a dedicated function to achieve this.

Installing affyQCReport

```
# if (!requireNamespace("BiocManager", quietly = TRUE))
#     install.packages("BiocManager")
# BiocManager::install("affyQCReport", version = "3.8")
```

Quality Control

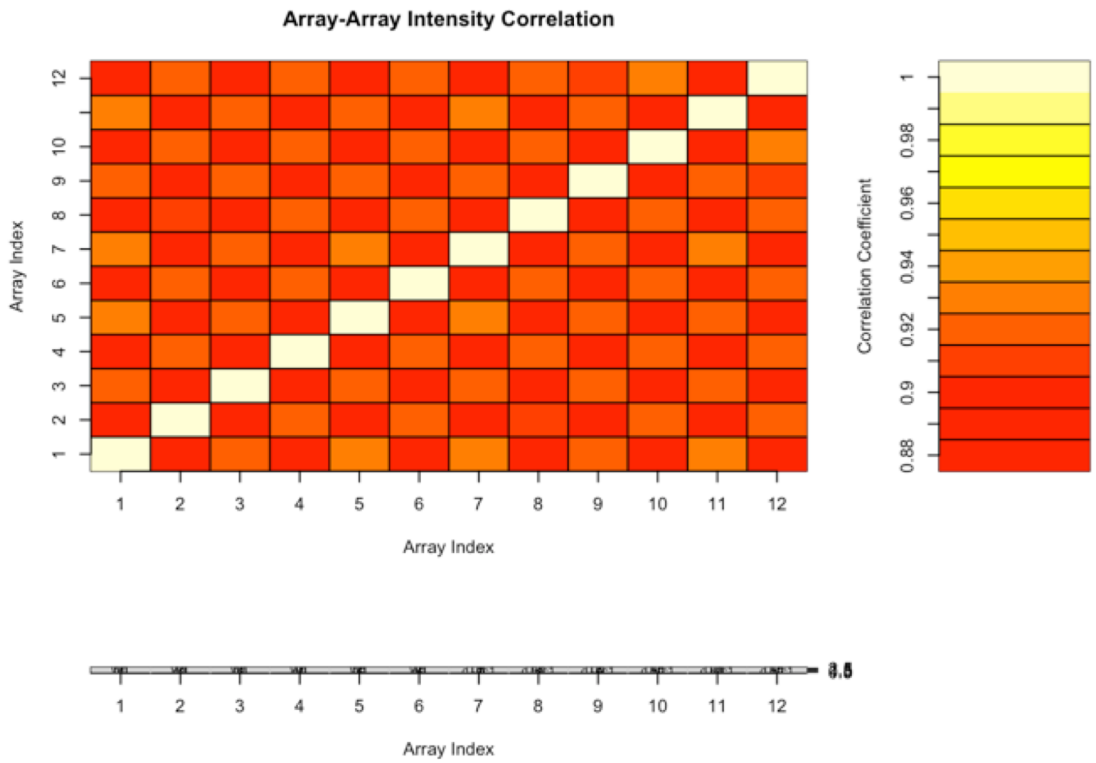
Using affyQCReport to generate a Correlation Plot

```
library(affyQCReport)
```

```
## Loading required package: lattice
```

```
correlationPlot(TRGN599_data)
```

```
## [1] TRUE
```

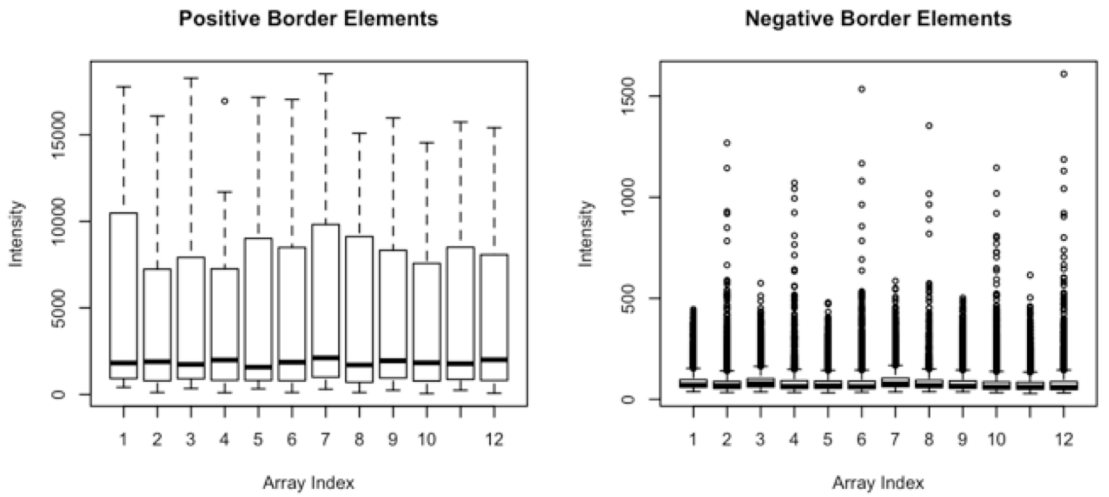


Quality Control

- Since all datasets in the public databases are supposed to pass quality control, usually the data analysts neglects to redo this for datasets obtained from GEO or ArrayExpress.

Generating BorderQCs

```
borderQC1(TRGN599_data)
```




```
## [1] TRUE
```

Normalization

- The appropriate normalization methodology of microarray measurements was in the focus of intensive research for many years.
- Several methods have been developed for both within-array and between-array normalization.
- The purpose of normalization is to compensate for intensity differences coming from the handling of the samples.
- Also to highlight those arising from the biological differences between the samples.
- Technical variances can come from very different sources, such as different amounts of RNA are used, one dye is more readily incorporated than the other in two-color systems, while, in one-color systems, labeling with different amount of dye may occur, or the scanner reading can be faulty.

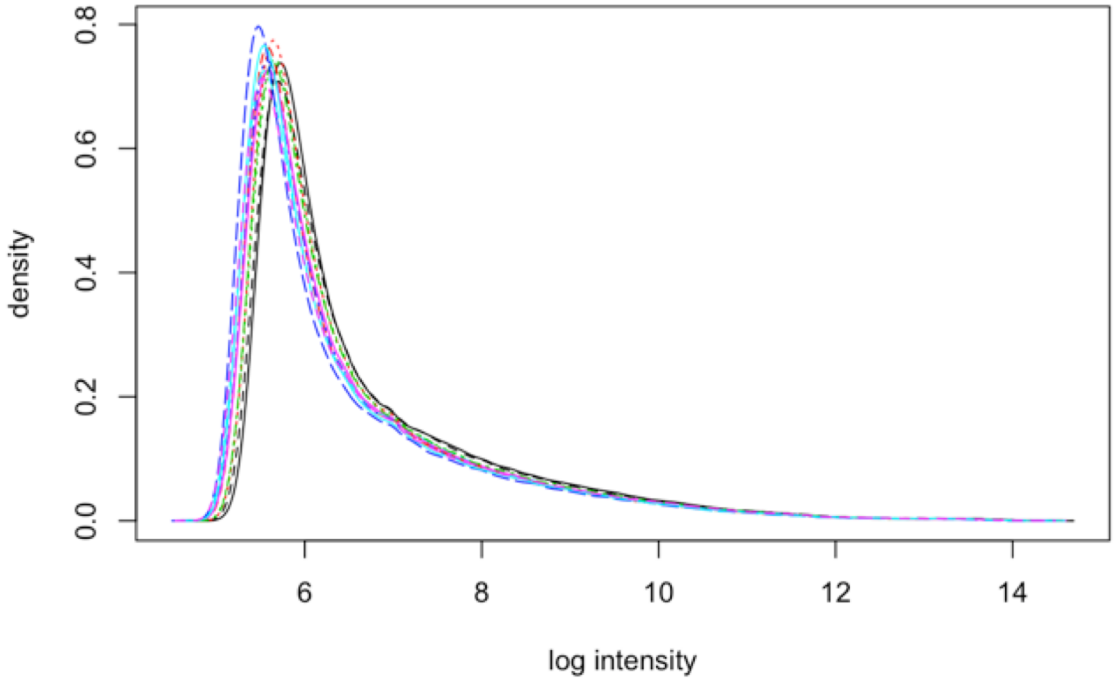
Normalization

- Within-array normalization adjusts the log ratios of a single array so that they average to zero within the array.
- Between-array normalization adjusts expression intensities or log ratios so that they have similar distribution among multiple arrays.
- Obviously, only two-color experiments have log ratios on a single array, so within-array normalization is used only there.
- Normalization approaches often used the following:
  - MAS 5.0- A normalization method provided by Affymetrix.
    - It takes into account MM probes, unlike other normalization methods.
  - Robust multi-array average (RMA).
    - It summarizes the PM using the median polish algorithm.
    - Quantile normalization is also part of RMA.
  - Variance stabilizing normalization (VSN)
    - Tries to keep the variance constant across all expression levels.
  - Locally weighted scatterplot smoothing (LOWESS normalization)
    - It is used for two-color data and applied to the log (Red/Green) versus log(sqrt(Red*Green)) plot (data transformed to M (log ratios))

Normalization

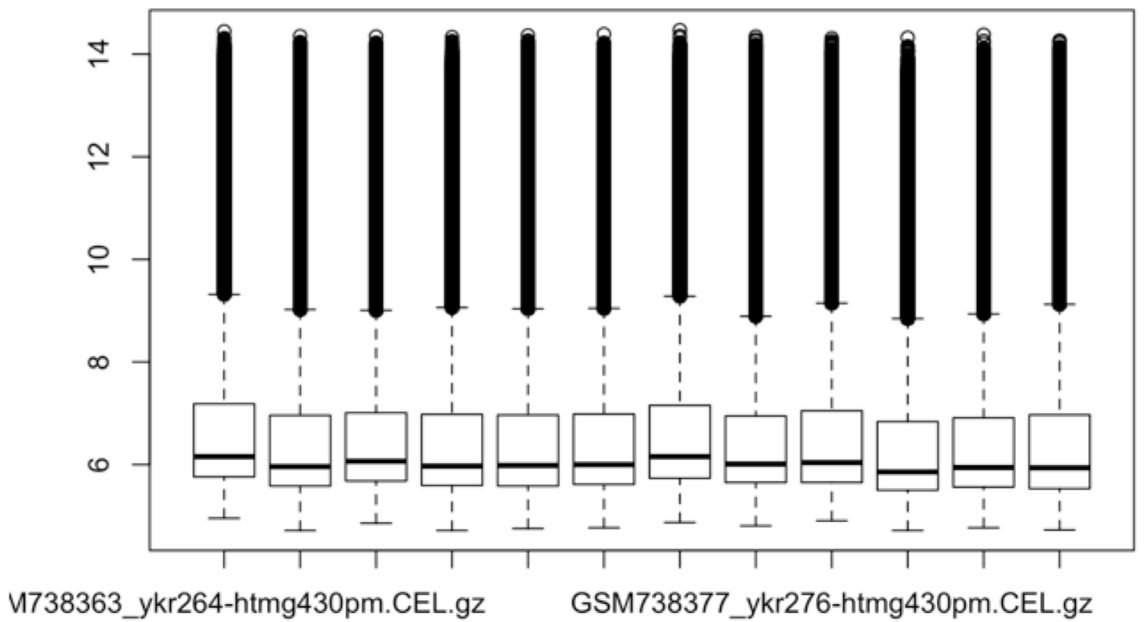
Normalization Process

```
hist(TRGN599_data)
```



Normalization

```
boxplot(log2(exprs(TRGN599_data)))
```



Normalization

Generating and expression set

```
TRGN599_datrma<-rma(TRGN599_data)
```

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

```
TRGN599_datrma # this is an expression set
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 45141 features, 12 samples
##   element names: exprs
## protocolData
##   sampleNames: GSM738363_ykr264-htmg430pm.CEL.gz
##     GSM738365_ykr265-htmg430pm.CEL.gz ...
##     GSM738380_ykr280-htmg430pm.CEL.gz (12 total)
##   varLabels: ScanDate
##   varMetadata: labelDescription
## phenoData
##   sampleNames: GSM738363_ykr264-htmg430pm.CEL.gz
##     GSM738365_ykr265-htmg430pm.CEL.gz ...
##     GSM738380_ykr280-htmg430pm.CEL.gz (12 total)
##   varLabels: sample Genotype Stimulation
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: htmg430pm
```

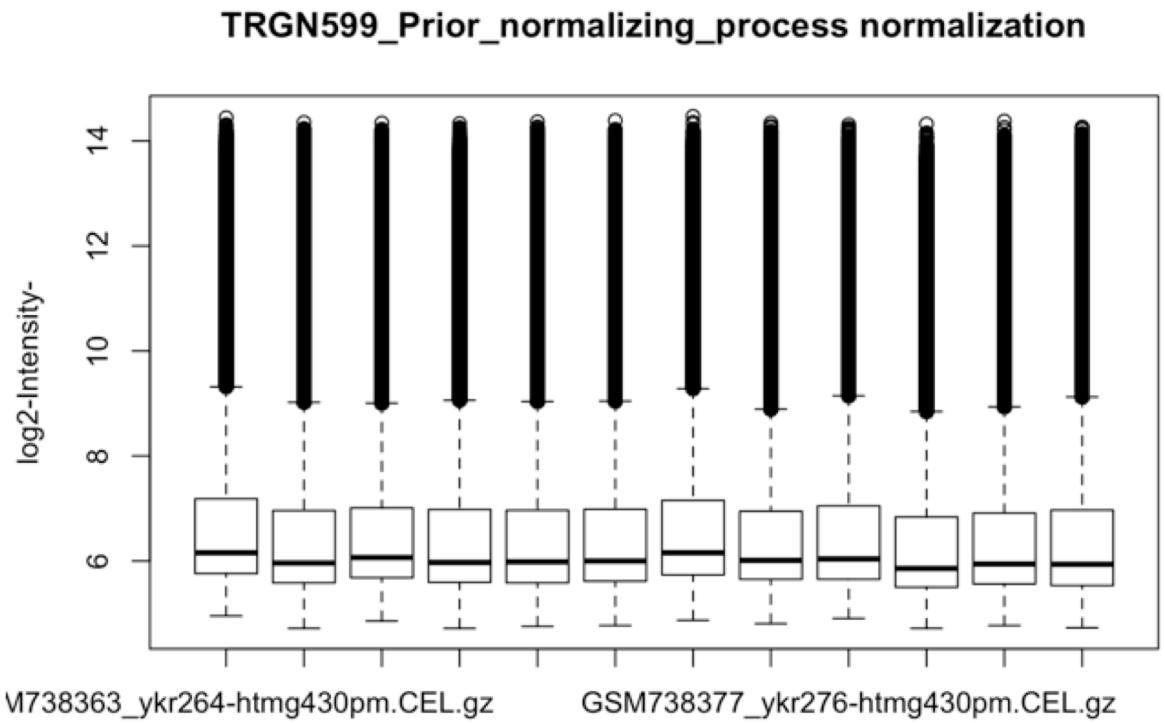
```
TRGN599_matexp<-exprs(TRGN599_datrma)
par(mfrow=c(1,2))
```

Normalization

- The produced TRGN599_datrma object is an ExpressionSet used in many downstream calculations.
- Very often, the central effort for any microarray data preprocessing is to obtain a normalized ExpressionSet from the original data, as all other calculations can be standardized.
- While the rma() function was executed, both background correction and normalization steps were performed.
- Let's check how the distribution of the intensities on each array was adjusted to have a common pattern.

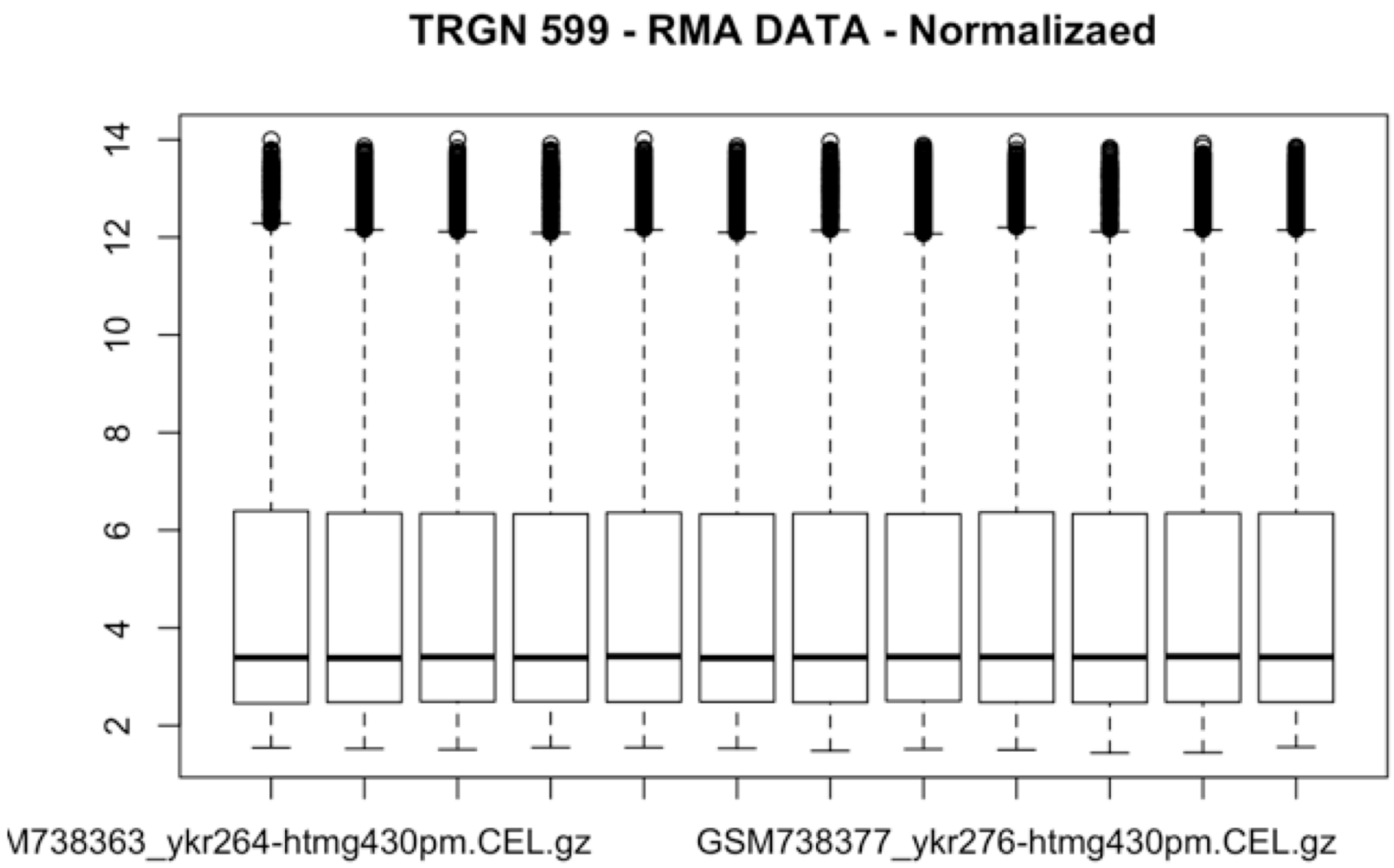
Generating Boxplots

```
boxplot(log2(exprs(TRGN599_data)),main="TRGN599_Prior_normalizing_process normalization",ylab="log2-Intensity-")
```



Normalization

```
boxplot(TRGN599_matexp,main="TRGN 599 - RMA DATA - Normalizaed")
```



Normalization

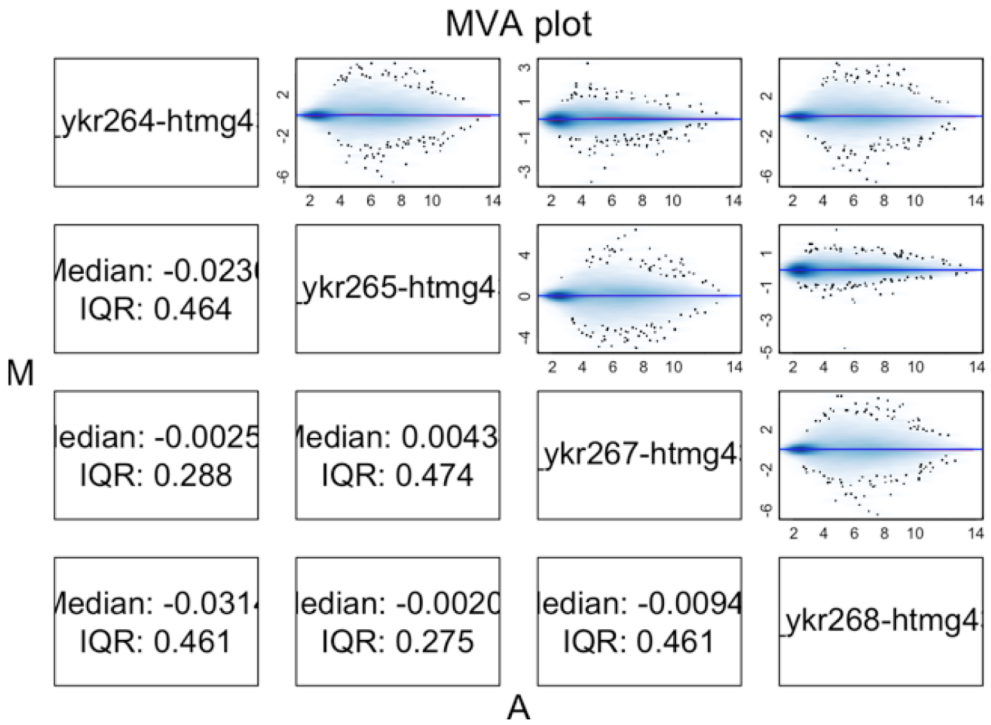
- These plots are supposed to be roughly horizontal if the normalization is efficient and successful.
- If so, the data are ready for further statistical analysis
  - For example
    - For differential gene expression identification

Generating MA plots

```
par(mfrow=c(1,1))
```

Comments: it could be use the

```
MAplot(TRGN599_datrma[,1:4],pairs=TRUE,plot.method="smoothScatter")
```



Normalization

```
TRGN599_datmas <- mas5(TRGN599_data)
```

```
## background correction: mas
## PM/MM correction : mas
## expression values: mas
## background correcting...done.
## 45141 ids to be processed
## |                    |
## |####################|
```

```
TRGN599_datmas
```

```
## ExpressionSet (storageMode: lockedEnvironment)
## assayData: 45141 features, 12 samples
##   element names: exprs, se.exprs
## protocolData
##   sampleNames: GSM738363_ykr264-htmg430pm.CEL.gz
##     GSM738365_ykr265-htmg430pm.CEL.gz ...
##     GSM738380_ykr280-htmg430pm.CEL.gz (12 total)
##   varLabels: ScanDate
##   varMetadata: labelDescription
## phenoData
##   sampleNames: GSM738363_ykr264-htmg430pm.CEL.gz
##     GSM738365_ykr265-htmg430pm.CEL.gz ...
##     GSM738380_ykr280-htmg430pm.CEL.gz (12 total)
##   varLabels: sample Genotype Stimulation
##   varMetadata: labelDescription
## featureData: none
## experimentData: use 'experimentData(object)'
## Annotation: htmg430pm
```