

Microarray Gene Expression Analysis with R

Example: Interferon Regulatory Factor 6 (*IRF6*)

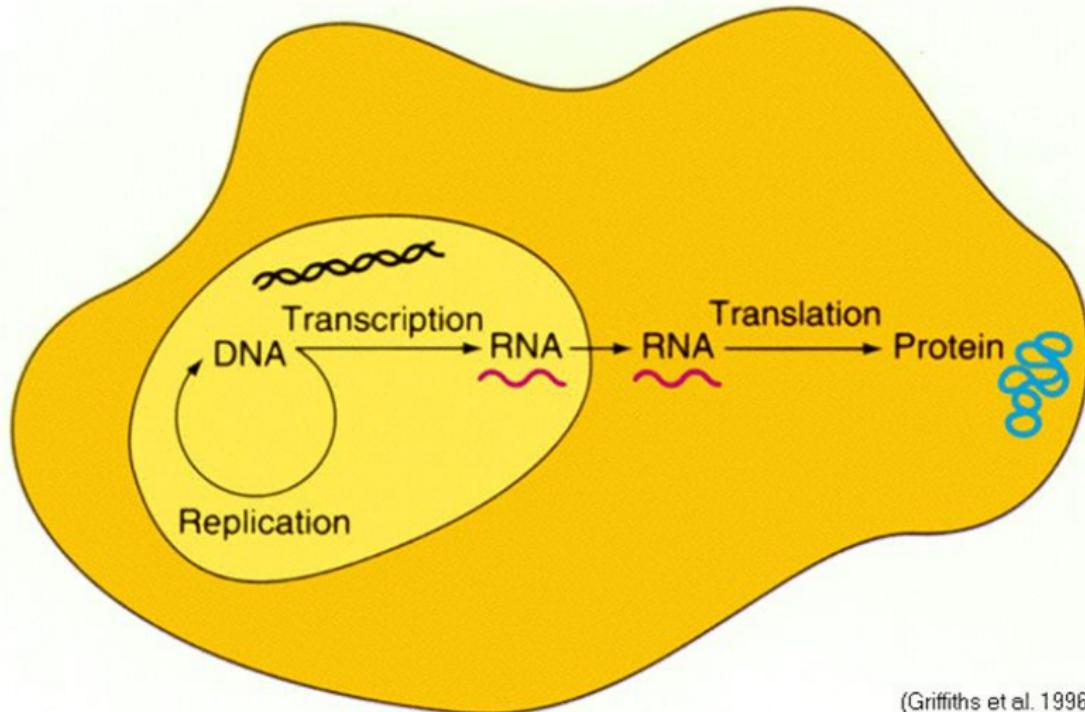
Ahmed Moustafa (amoustafa@aucegypt.edu)

Objectives

- ▶ Load microarray dataset into R
- ▶ Explore the dataset with basic visualizations
- ▶ Identify differentially expressed genes (DEGs)
- ▶ Generate annotation of the DEGs (*Tentative*)



The Central Dogma of Biology



(Griffiths et al. 1996)

Figure 1: DNA makes RNA and RNA makes protein

Cleft Lip and Palate 1/3

Cleft lip and cleft palate (**CLP**) are splits in the upper lip, the roof of the mouth (palate) or both. They result when facial structures that are developing in an unborn baby do not close completely. CLP is one of the most common birth defects with a frequency of 1/700 live births.



Cleft palate



Cleft lip and cleft palate



Cleft Lip and Palate 2/3

Children with cleft lip with or without cleft palate face a variety of challenges, depending on the type and severity of the cleft.

- ▶ **Difficulty feeding.** One of the most immediate concerns after birth is feeding.
- ▶ **Ear infections and hearing loss.** Babies with cleft palate are especially at risk of developing middle ear fluid and hearing loss.
- ▶ **Dental problems.** If the cleft extends through the upper gum, tooth development may be affected.
- ▶ **Speech difficulties.** Because the palate is used in forming sounds, the development of normal speech can be affected by a cleft palate. Speech may sound too nasal.

Reference: Mayo Foundation for Medical Education and Research

Cleft Lip and Palate 3/3

- ▶ DNA variation in Interferon Regulatory Factor 6 (**IRF6**) causes Van der Woude syndrome (**VWS**)
- ▶ VWS is the most common syndromic form of cleft lip and palate.
- ▶ However, the causing variant in IRF6 has been found in *only* 70% of VWS families!
- ▶ IRF6 is a **transcription factor** with a conserved helix-loop-helix DNA binding domain and a less well-conserved protein binding domain.

Reference: Hum Mol Genet. 2014 May 15; 23(10): 2711–2720

Question

Given:

1. The pathogenic variant in IRF6 exists in only 70% of the VWS families
2. IRF6 is a transcription factor

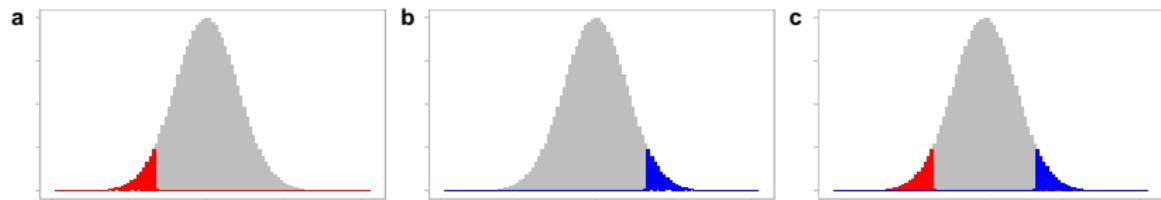
How can we identify other genes that might be involved in the remaining 30% of the VWS families?

Hint

- ▶ Usually, genes that are regulated by a transcription factor belong to the same biological process or pathway.
- ▶ Therefore, by comparing the gene expression patterns between wild-type (functional) *Irf6* and knockout (non-functional) *Irf6*, it could be possible to identify genes that are regulated (targeted) by *Irf6*.

Hypothesis

- ▶ $H_0 : \mu_{WT} = \mu_{KO}$
- ▶ $H_A : \mu_{WT} \neq \mu_{KO}$
- ▶ Where μ is the *mean* of the gene expression values of a gene.
- ▶ **One**-sided or **Two**-sided testing?



Why Microarray?



Figure 3

Why Microarray?

- ▶ No need for candidate genes (or genes of interest)
- ▶ One experiment assesses the entire transcriptome
- ▶ One experiment generates many hypotheses
- ▶ Only small amount of RNA is required (~15–200 ng)



Original Paper

Nat Genet. 2006 Nov;38(11):1335-40. Epub 2006 Oct 15.

Abnormal skin, limb and craniofacial morphogenesis in mice deficient for interferon regulatory factor 6 (Irf6).

Ingraham CR¹, Kinoshita A, Kondo S, Yang B, Sajan S, Trout KJ, Malik MI, Dunnwald M, Goudy SL, Lovett M, Murray JC, Schutte BC.

Author information

Abstract

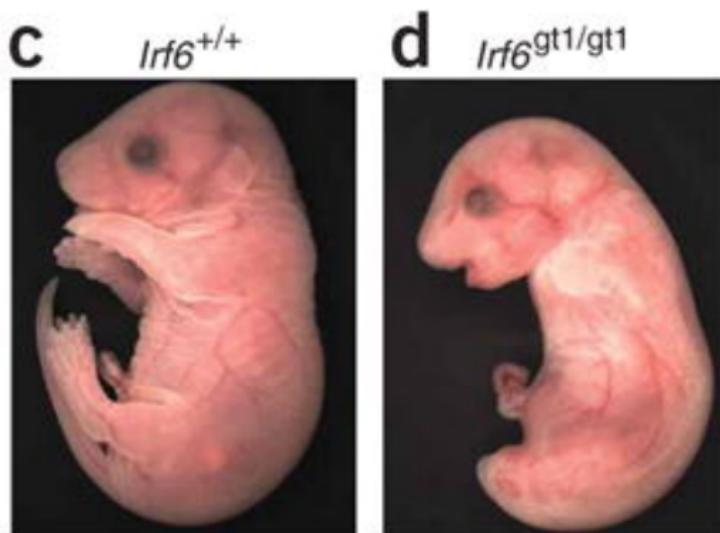
Transcription factor paralogs may share a common role in staged or overlapping expression in specific tissues, as in the Hox family. In other cases, family members have distinct roles in a range of embryologic, differentiation or response pathways (as in the Tbx and Pax families). For the interferon regulatory factor (IRF) family of transcription factors, mice deficient in Irf1, Irf2, Irf3, Irf4, Irf5, Irf7, Irf8 or Irf9 have defects in the immune response but show no embryologic abnormalities. Mice deficient for Irf6 have not been reported, but in humans, mutations in IRF6 cause two mendelian orofacial clefting syndromes, and genetic variation in IRF6 confers risk for isolated cleft lip and palate. Here we report that mice deficient for Irf6 have abnormal skin, limb and craniofacial development. Histological and gene expression analyses indicate that the primary defect is in keratinocyte differentiation and proliferation. This study describes a new role for an IRF family member in epidermal development.

PMID: 17041601 PMCID: PMC2082114 DOI: 10.1038/ng1903

Figure 5: PMID: 17041601

Experimental Design

- ▶ 3 IRF6 wild-type (+/+) and 3 knockout (-/-) mouse embryos.
- ▶ E17.5 embryos were removed from euthanized mothers.
- ▶ Skin was removed from embryos.
- ▶ Total RNA was isolated from the skin.
- ▶ Resultant RNA was hybridized to Affymetrix GeneChip Mouse Genome 430 2.0 arrays.



Dataset

- ▶ The original dataset can be obtained from NCBI GEO with accession GSE5800

| ID | KO1 | KO2 | KO3 | WT1 | WT2 | WT3 |
|------------|---------|---------|---------|---------|---------|---------|
| 1415670_at | 6531.0 | 5562.8 | 6822.4 | 7732.1 | 7191.2 | 7551.1 |
| 1415671_at | 11486.3 | 10542.7 | 10641.4 | 10408.2 | 9484.5 | 7650.1 |
| 1415672_at | 14339.2 | 13526.1 | 14444.7 | 12936.6 | 13841.7 | 13285.1 |
| 1415673_at | 3156.8 | 2219.5 | 3264.4 | 2374.2 | 2201.8 | 2525.1 |

- ▶ Irf6.xlsx (Excel): <https://goo.gl/hrmnct>
- ▶ Irf6.tsv (Tab-separated values): [\(Please download\)](https://goo.gl/tYAz6v)

Loading

First, we are going to load the dataset from the .tsv file into R as a variable called data using the `read.table` function. data is just an arbitrary **variable** name to hold the result of `read.table` and it can be called/named *almost* anything.

```
# Load the data from a file into a variable  
data = read.table("Irf6.tsv", header = TRUE, row.names = 1)  
  
# Convert the data.frame (table) in a matrix (numeric)  
data = as.matrix(data)
```

Note: the hash sign (#) indicates that what comes after is a *comment*. Comments are for documentation and readability of the R code and they are not evaluated (or executed).

Checking

```
dim(data) # Dimension of the dataset
```

```
## [1] 45101      6
```

```
head(data) # First few rows
```

| | KO1 | KO2 | KO3 | WT1 | WT2 | V |
|--------------|---------|---------|---------|---------|---------|-----|
| 1415670_at | 6531.0 | 5562.8 | 6822.4 | 7732.1 | 7191.2 | 75 |
| 1415671_at | 11486.3 | 10542.7 | 10641.4 | 10408.2 | 9484.5 | 76 |
| 1415672_at | 14339.2 | 13526.1 | 14444.7 | 12936.6 | 13841.7 | 132 |
| 1415673_at | 3156.8 | 2219.5 | 3264.4 | 2374.2 | 2201.8 | 25 |
| 1415674_a_at | 4002.0 | 3306.9 | 3777.0 | 3760.6 | 3137.0 | 29 |
| 1415675_at | 3468.4 | 3347.4 | 3332.9 | 3073.5 | 3046.0 | 29 |

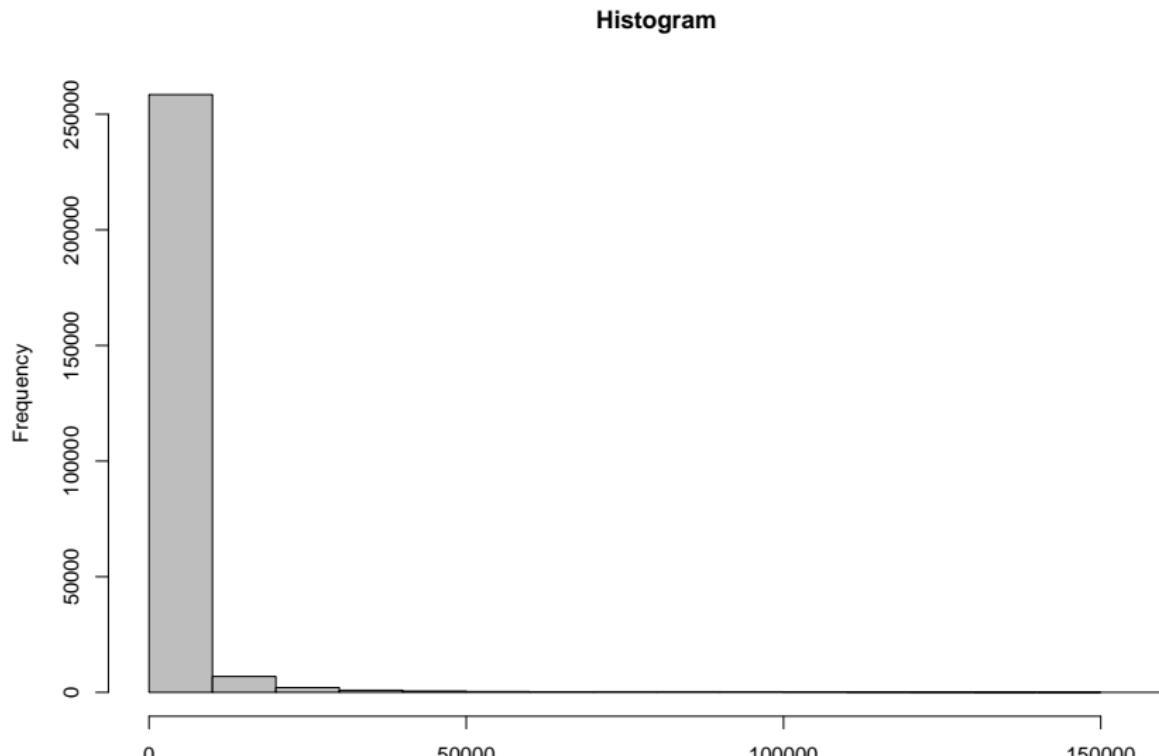
Number of Genes and IDs

```
number_of_genes = nrow(data) # number of genes = number of  
number_of_genes  
  
## [1] 45101  
  
ids = row.names(data) # The ids of the genes are the names  
head(ids)  
  
## [1] "1415670_at"    "1415671_at"    "1415672_at"    "1415673_at"  
## [5] "1415674_a_at"  "1415675_at"
```

Exploring

Check the behavior of the data (e.g., normal?, skewed?)

```
hist(data, col = "gray", main="Histogram")
```

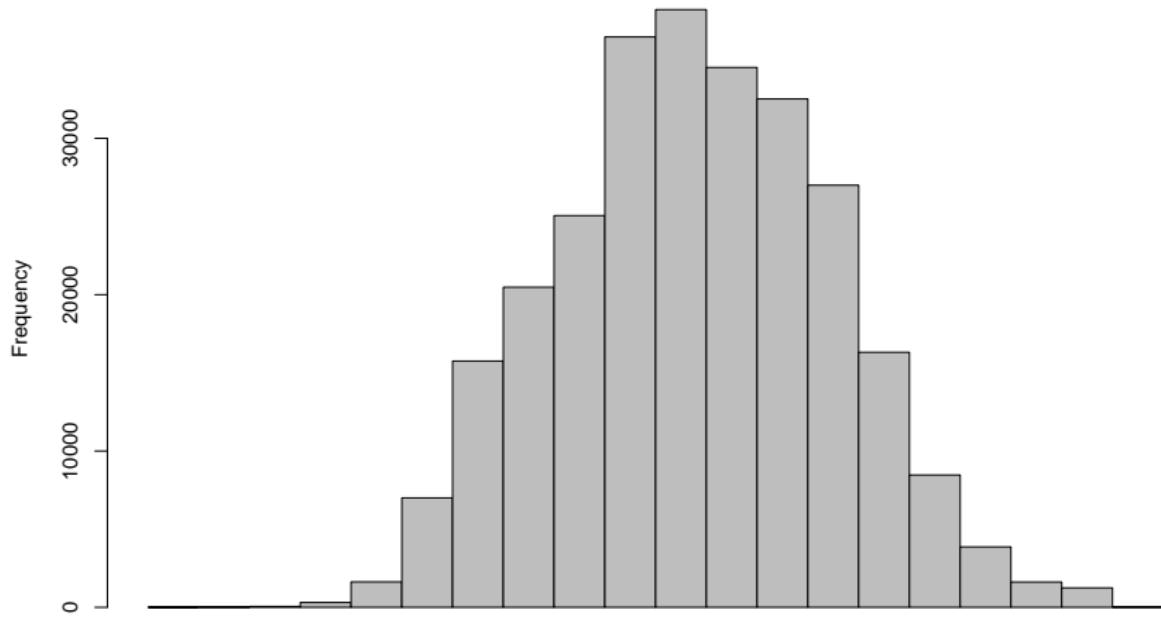


Transforming

\log_2 transformation (why?)

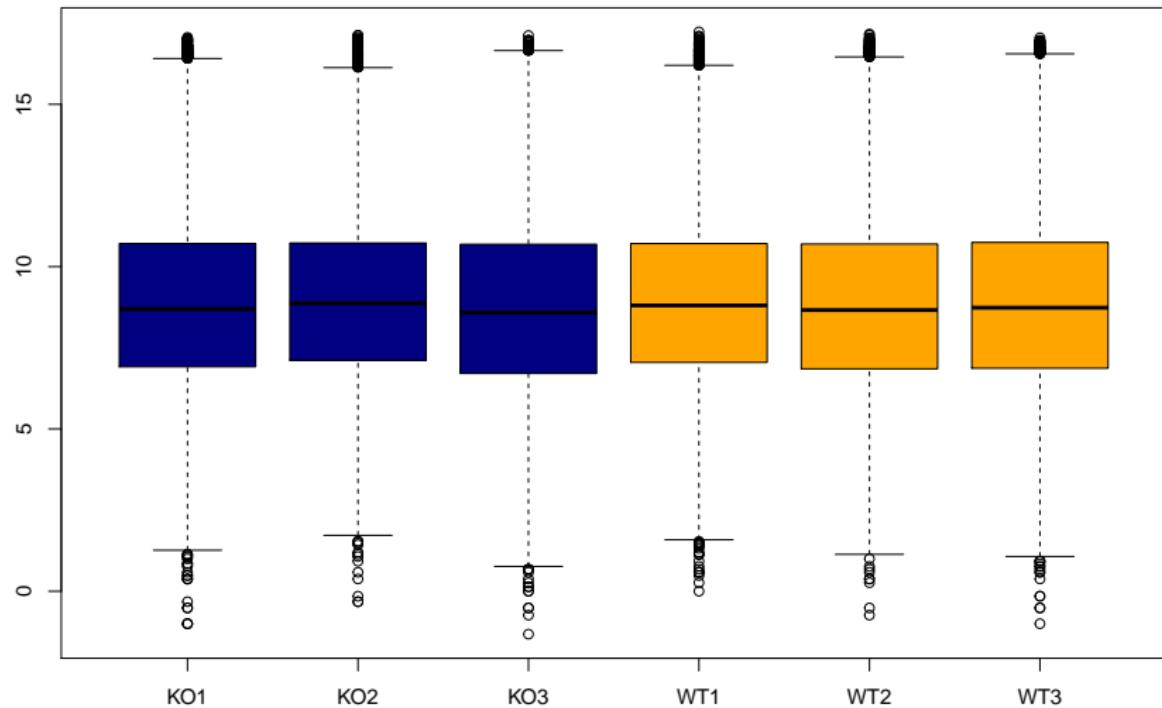
```
data2 = log2(data)  
hist(data2, col = "gray")
```

Histogram of data2



Boxplot

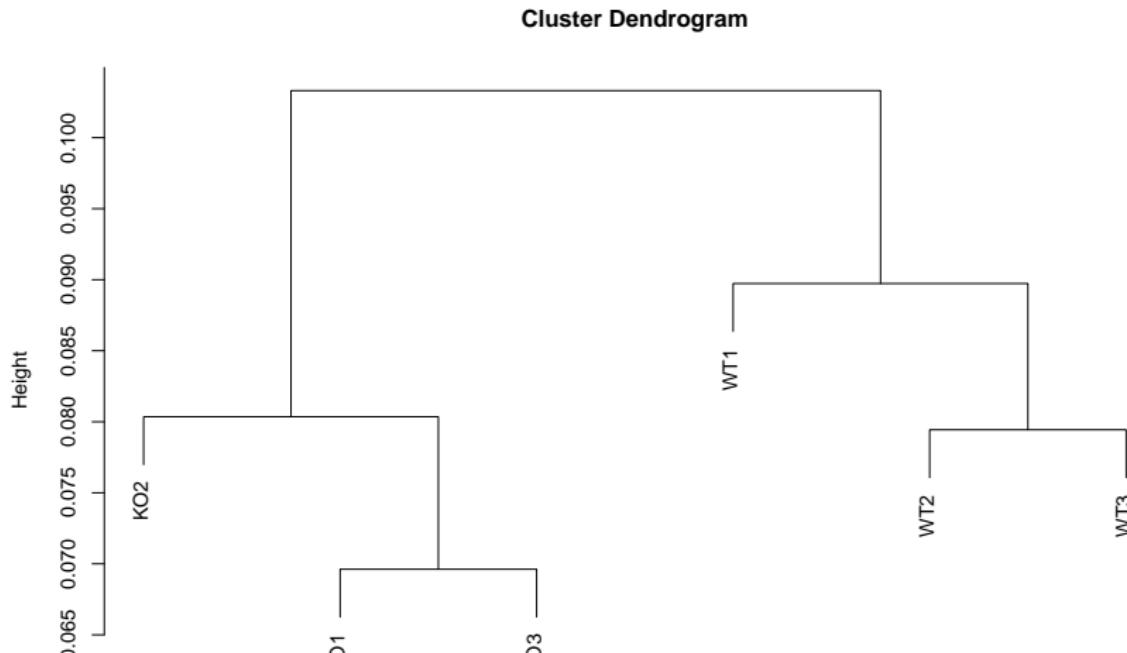
```
colors = c(rep("navy", 3), rep("orange", 3))
boxplot(data2, col = colors)
```



Clustering 1/2

Hierarchical clustering of the **samples** (i.e., columns) based on the correlation coefficients of the expression values

```
hc = hclust(as.dist(1 - cor(data2)))
plot(hc)
```



Clustering 2/2

To learn more about a function (e.g., `hclust`), you may type
`?function` (e.g., `?hclust`) in the console to launch R documentation on that function:

Splitting Data Matrix into Two 1/2

```
ko = data2[, 1:3] # KO matrix  
head(ko)
```

| | KO1 | KO2 | KO3 |
|--------------|----------|----------|----------|
| 1415670_at | 12.67309 | 12.44160 | 12.73606 |
| 1415671_at | 13.48763 | 13.36396 | 13.37740 |
| 1415672_at | 13.80768 | 13.72346 | 13.81825 |
| 1415673_at | 11.62425 | 11.11602 | 11.67260 |
| 1415674_a_at | 11.96651 | 11.69126 | 11.88303 |
| 1415675_at | 11.76005 | 11.70883 | 11.70256 |

Splitting Data Matrix into Two 2/2

```
wt = data2[, 4:6] # WT matrix  
head(wt)
```

| | WT1 | WT2 | WT3 |
|--------------|----------|----------|----------|
| 1415670_at | 12.91664 | 12.81202 | 12.88262 |
| 1415671_at | 13.34543 | 13.21136 | 12.90128 |
| 1415672_at | 13.65917 | 13.75673 | 13.69759 |
| 1415673_at | 11.21323 | 11.10447 | 11.30224 |
| 1415674_a_at | 11.87675 | 11.61517 | 11.50755 |
| 1415675_at | 11.58567 | 11.57270 | 11.50898 |

Gene (Row) Mean Expression

```
# Compute the means of the KO samples  
ko.means = rowMeans(ko)  
head(ko.means)
```

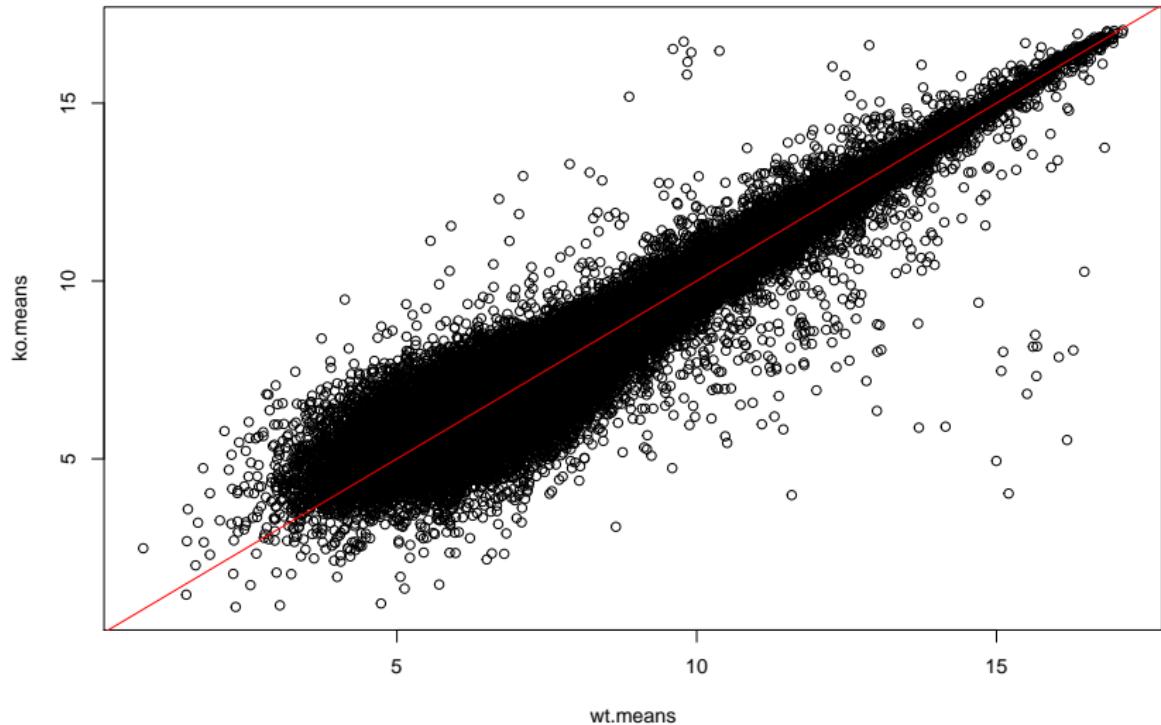
```
##    1415670_at    1415671_at    1415672_at    1415673_at 1415  
##    12.61692     13.40966     13.78313     11.47096  
##    1415675_at  
##    11.72381
```

```
# Compute the means of the WT samples  
wt.means = rowMeans(wt)  
head(wt.means)
```

```
##    1415670_at    1415671_at    1415672_at    1415673_at 1415  
##    12.87043     13.15269     13.70450     11.20664  
##    1415675_at  
##    11.55578
```

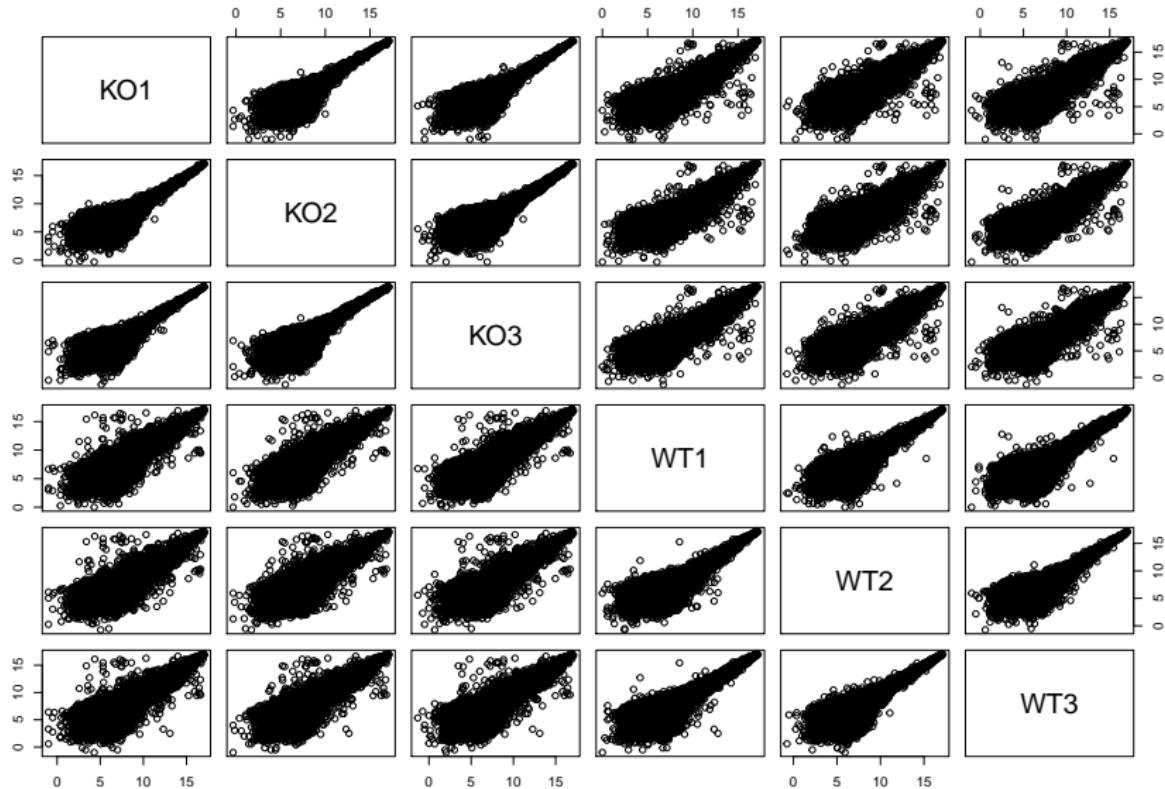
Scatter 1/2

```
plot(ko.means ~ wt.means) # The actual scatter plot  
abline(0, 1, col = "red") # Only a diagonal line
```



Scatter 2/2

```
pairs(data2) # All pairwise comparisons
```



Differentially Expressed Genes (DEGs)

To identify DEGs, we will identify:

- ▶ **Biologically** significantly differentially expressed
- ▶ **Statistically** significantly differentially expressed

Then, we will take the **overlap (intersection)** of the two sets

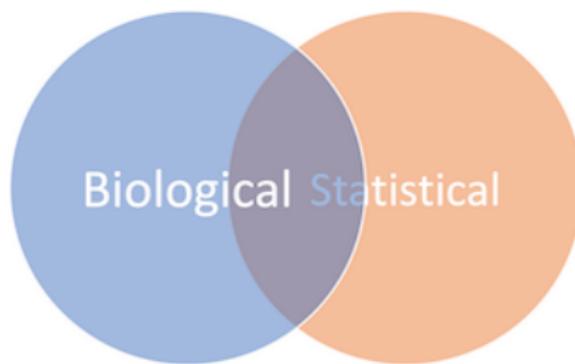


Figure 7

Biological Significance (fold-change) 1/2

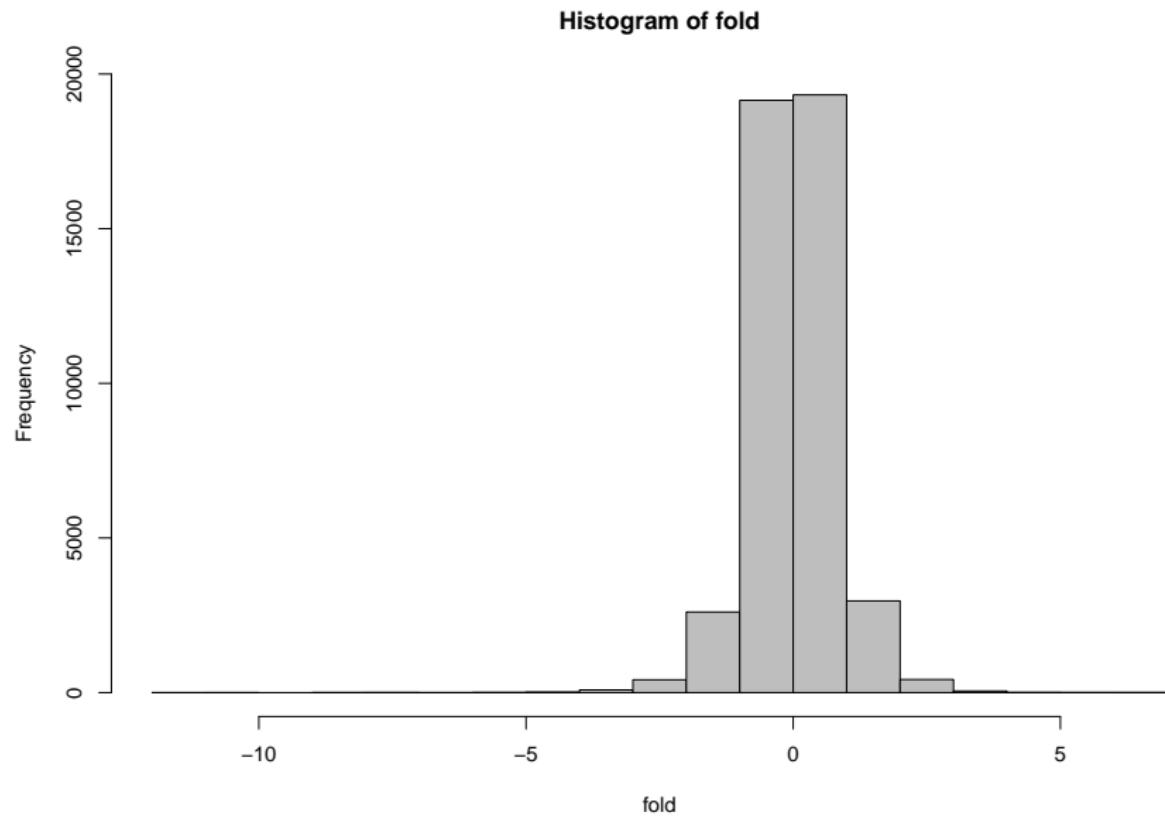
```
fold = ko.means - wt.means # Difference between means  
head(fold)
```

```
##    1415670_at    1415671_at    1415672_at    1415673_at 1415  
## -0.25351267    0.25697097    0.07863227    0.26431191 0  
##    1415675_at  
##    0.16803065
```

- ▶ What do the positive and negative values of the fold-change indicate? Considering the WT condition is the **reference** (or **control**)
 - ▶ **+ve** fold-change → **Up**-regulation ↑
 - ▶ **-ve** fold-change → **Down**-regulation ↓

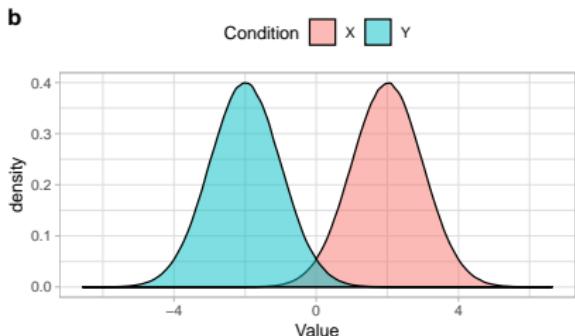
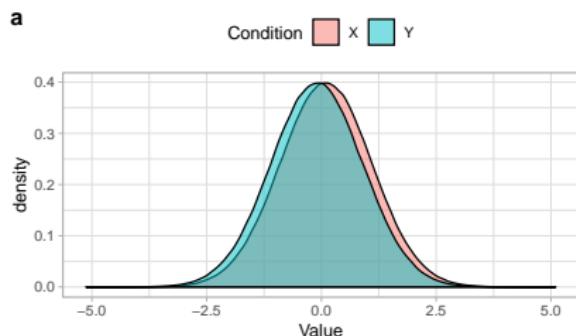
Biological Significance (fold-change) 2/2

```
hist(fold, col = "gray") # Histogram of the fold
```



Statistical Significance (p -value) 1/3

- To assess the statistical significance of the difference in the expression values for each gene between the two conditions (e.g., WT and KO), we are going to use t -test.



t-test

Let's say there are two samples x and y from the two populations, X and Y , respectively, to determine whether the means of two populations are significantly different, we can use `t.test`.

```
?t.test
```

```
## Student's t-Test
##
## Description:
##
##     Performs one and two sample t-tests on vectors of
##     observations.
##
## Usage:
##
##     t.test(x, ...)
##
##     ## Default S3 method:
##     t.test(x, y = NULL,
##           alternative = c("two.sided", "less", "greater"),
##           conf.level = 0.95,
##           var.equal = FALSE,
##           na.action = na.omit,
##           trim = 0,
##           plot = TRUE)
```

t-test : Example 1

```
x = c(4, 3, 10, 7, 9) ; y = c(7, 4, 3, 8, 10)
t.test(x, y)
```

```
##  
##  Welch Two Sample t-test  
##  
## data: x and y  
## t = 0.1066, df = 7.9743, p-value = 0.9177  
## alternative hypothesis: true difference in means is not  
## 95 percent confidence interval:  
## -4.12888 4.52888  
## sample estimates:  
## mean of x mean of y  
##       6.6       6.4
```

```
t.test(x, y)$p.value
```

```
## [1] 0.917739
```

t-test : Example 2

```
x = c(6, 8, 10, 7, 9) ; y = c(3, 2, 1, 4, 5)
t.test(x, y)
```

```
##  
##  Welch Two Sample t-test  
##  
## data: x and y  
## t = 5, df = 8, p-value = 0.001053  
## alternative hypothesis: true difference in means is not  
## 95 percent confidence interval:  
##  2.693996 7.306004  
## sample estimates:  
## mean of x mean of y  
##          8          3
```

```
t.test(x, y)$p.value
```

```
## [1] 0.001052826
```

Statistical Significance (p -value) 2/3

Let's compute the p -value for all genes using a for-loop of `t.test`, one gene at a time:

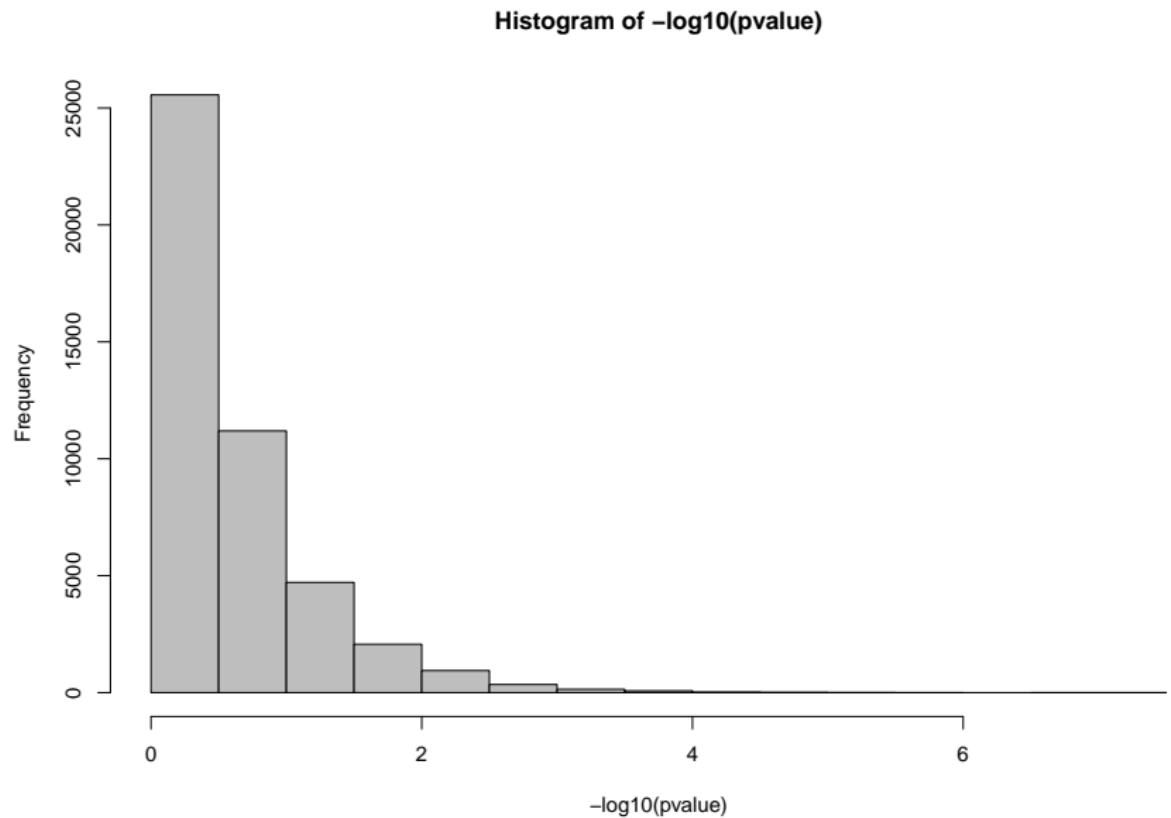
```
pvalue = NULL # Empty list for the p-values

for(i in 1 : number_of_genes) { # for each gene from to the
  x = wt[i, ] # wt values of gene number i
  y = ko[i, ] # ko values of gene number i
  t = t.test(x, y) # t-test between the two conditions
  pvalue[i] = t$p.value # Store p-value number i into the
}
head(pvalue)

## [1] 0.092706280 0.182663337 0.129779075 0.272899180 0.26
```

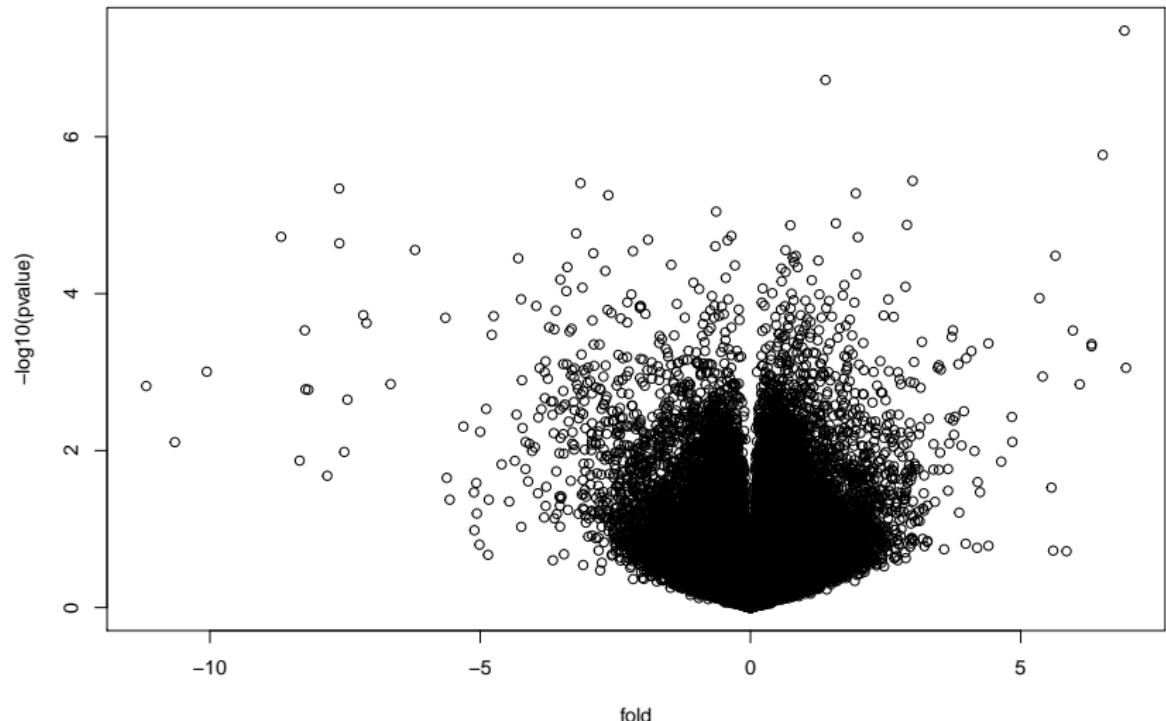
Statistical Significance (p -value) 3/3

```
hist(-log10(pvalue), col = "gray") # Histogram of p-values
```



Volcano : Statistical & Biological 1/3

```
plot(-log10(pvalue) ~ fold)
```



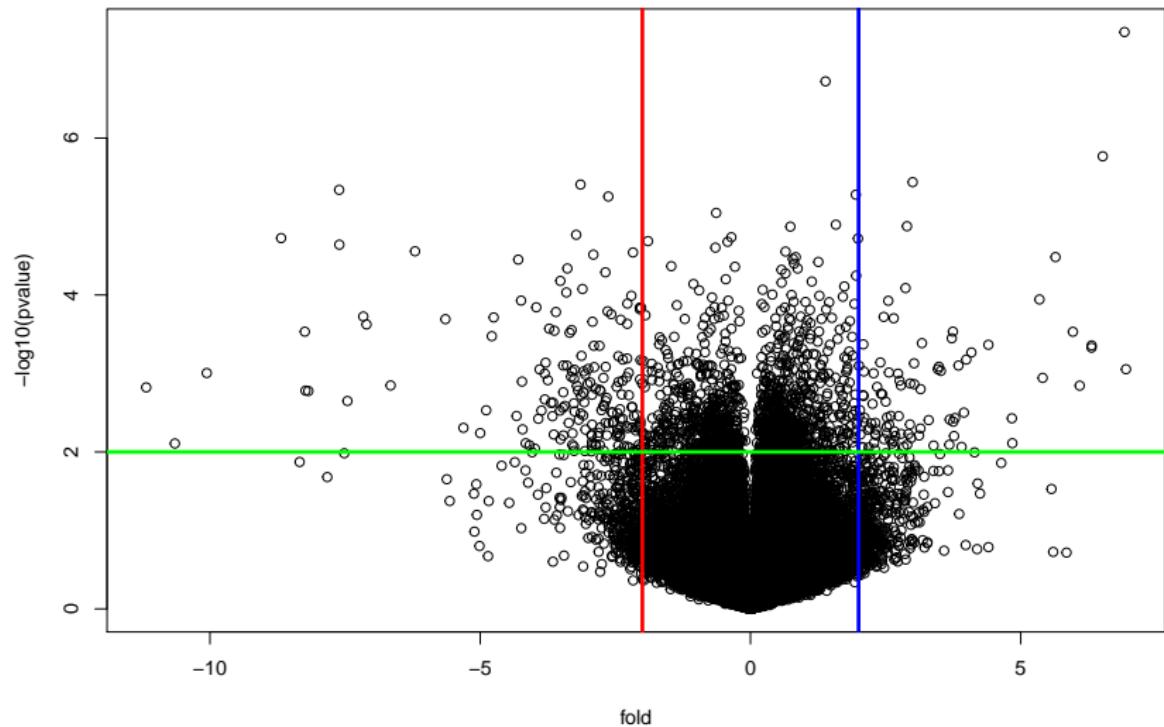
Volcano : Statistical & Biological 2/3

```
fold_cutoff = 2
pvalue_cutoff = 0.01

plot(-log10(pvalue) ~ fold)

abline(v = fold_cutoff, col = "blue", lwd = 3)
abline(v = -fold_cutoff, col = "red", lwd = 3)
abline(h = -log10(pvalue_cutoff), col = "green", lwd = 3)
```

Volcano : Statistical & Biological 3/3



Filtering for DEGs 1/3

```
filter_by_fold = abs(fold) >= fold_cutoff # Biological  
sum(filter_by_fold) # Number of genes satisfy the condition
```

```
## [1] 1051
```

```
filter_by_pvalue = pvalue <= pvalue_cutoff # Statistical  
sum(filter_by_pvalue)
```

```
## [1] 1564
```

```
filter_combined = filter_by_fold & filter_by_pvalue # Combined  
sum(filter_combined)
```

```
## [1] 276
```

Filtering for DEGs 2/3

```
filtered = data2[filter_combined, ]  
dim(filtered)
```

```
## [1] 276    6
```

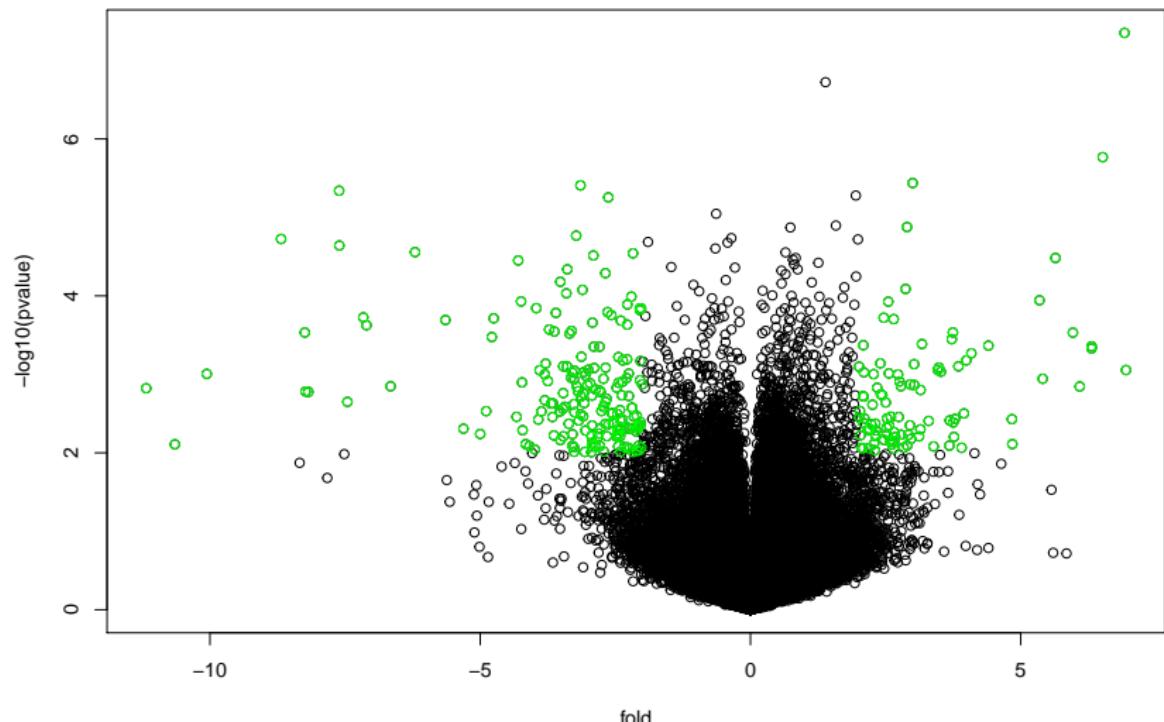
```
head(filtered)
```

| | KO1 | KO2 | KO3 | WT1 | V |
|--------------|-----------|-----------|-----------|----------|--------|
| 1416200_at | 13.312004 | 12.973357 | 12.868456 | 7.40429 | 8.558 |
| 1416236_a_at | 14.148397 | 14.039236 | 14.130007 | 12.23604 | 12.022 |
| 1417808_at | 5.321928 | 5.442944 | 4.053111 | 15.16978 | 15.070 |
| 1417932_at | 10.602884 | 10.257152 | 10.496055 | 13.98445 | 14.203 |
| 1418050_at | 10.622052 | 10.975490 | 10.795066 | 12.86513 | 13.012 |
| 1418100_at | 9.117903 | 8.634811 | 9.057721 | 12.90358 | 12.842 |

Filtering for DEGs 3/3

```
plot(-log10(pvalue) ~ fold)
```

```
points(-log10(pvalue[filter_combined]) ~ fold[filter_combined],  
       col = "green")
```



Exercise

On the volcano plot, highlight the up-regulated genes in red and the down-regulated genes in blue

Solution 1/2

- ▶ Up-regulated genes

```
# Screen for the up-regulated genes (+ve fold)
```

```
filter_up = filter_combined & fold > 0
```

```
head(filter_up)
```

```
##    1415670_at    1415671_at    1415672_at    1415673_at 1415674_at  
##          FALSE        FALSE        FALSE        FALSE        FALSE  
##    1415675_at  
##          FALSE
```

```
# Number of filtered genes
```

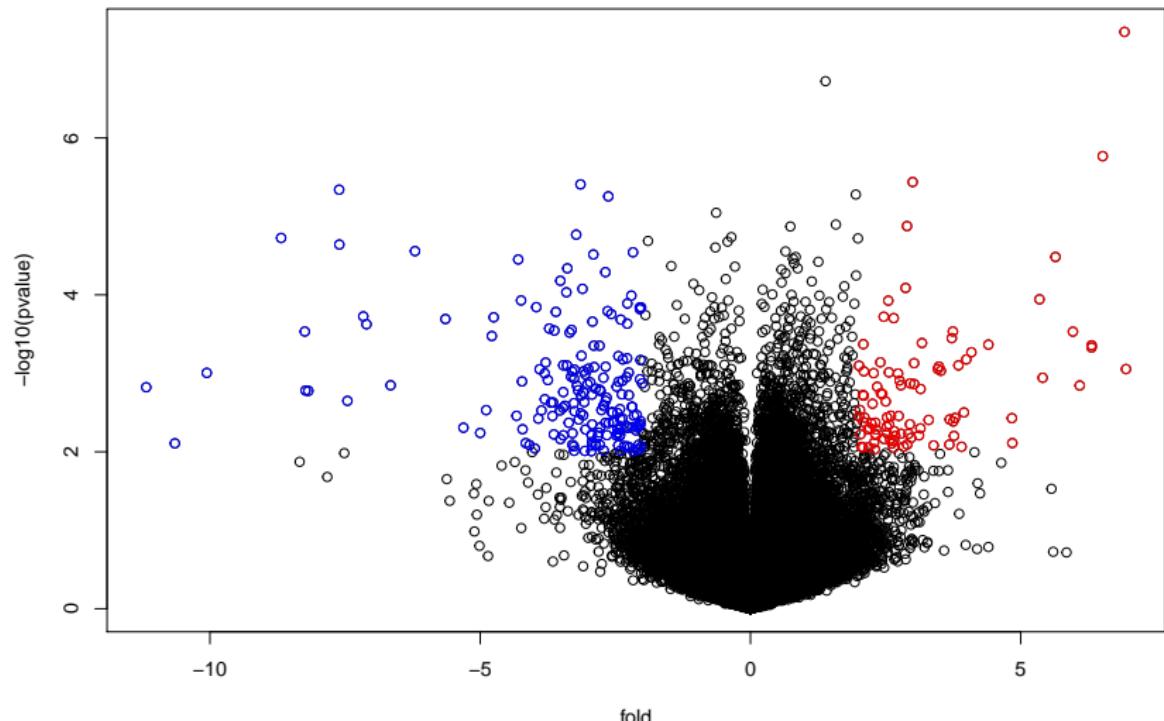
```
sum(filter_up)
```

```
## [1] 95
```

- ▶ Down-regulated genes

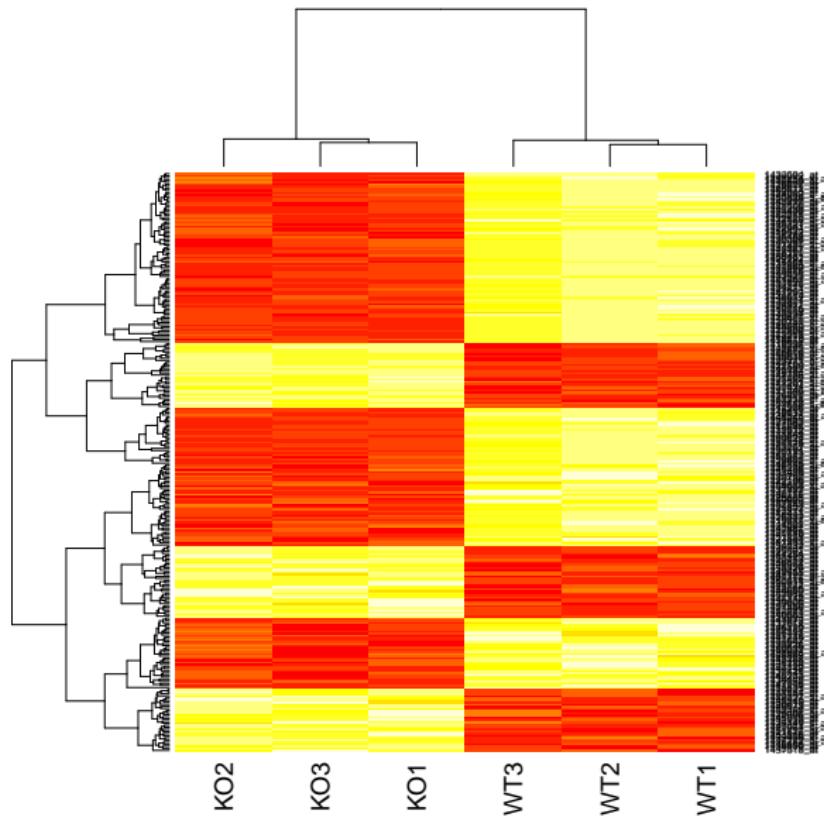
Solution 2/2

```
plot(-log10(pvalue) ~ fold)
points(-log10(pvalue[filter_up]) ~ fold[filter_up], col = "blue")
points(-log10(pvalue[filter_down]) ~ fold[filter_down], col = "red")
```



Heatmap 1/5

heatmap(filtered)



Heatmap 2/5

- ▶ By default, `heatmap` clusters genes (rows) and samples (columns) based on the Euclidean distance.
- ▶ In the context of gene expression, we need to cluster genes and samples based on the correlation to explore patterns of **co-regulation (co-expression)** - *Guilt by Association*.
- ▶ To let `heatmap` cluster the genes and/or samples, the genes and samples will be clustered (grouped) by correlation coefficients (using `cor`) among the genes and samples.

Heatmap 3/5

Clustering of the columns (samples)

```
col_dendrogram = as.dendrogram(hclust(as.dist(1-cor(filtered
```

Clustering of the rows (genes)

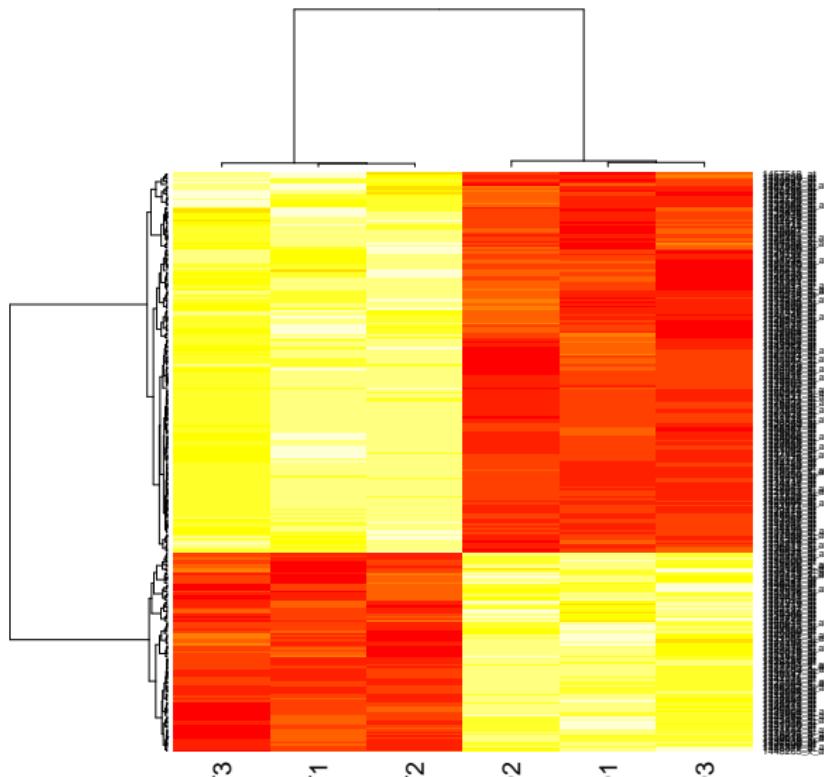
```
row_dendrogram = as.dendrogram(hclust(as.dist(1-cor(t(filtered
```

© Original Artist
Reproduction rights obtainable from
www.CartoonStock.com



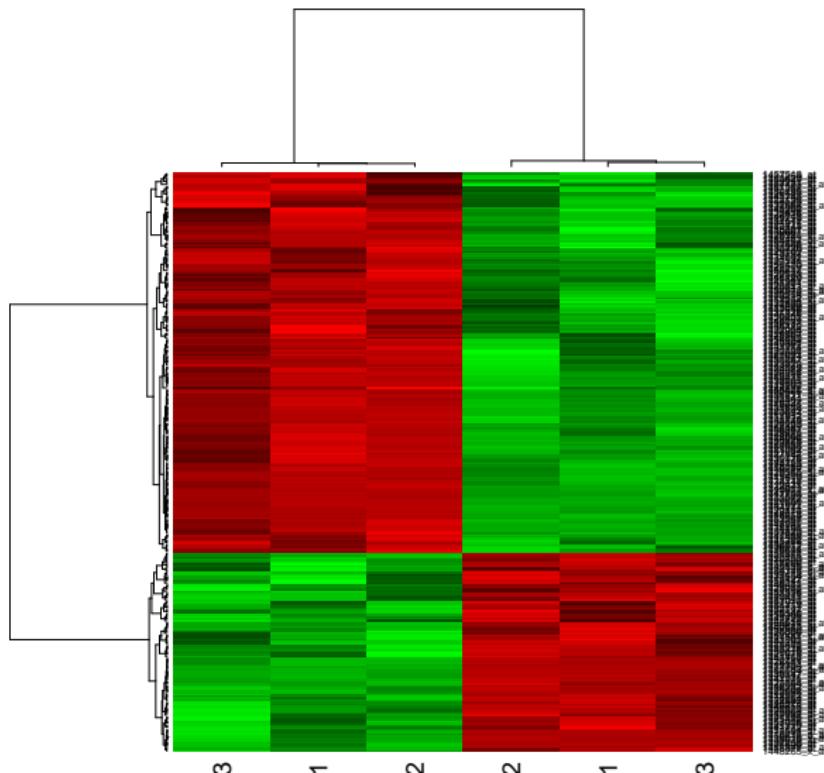
Heatmap 4/5

Heatmap with the rows and columns clustered by correlation
heatmap(filtered, Rowv=row_dendrogram, Colv=col_dendrogram)



Heatmap 5/5

```
library(gplots) # Load the gplots library  
heatmap(filtered, Rowv=row_dendrogram, Colv=col_dendrogram)
```



Annotation 1/3

To obtain the functional annotation of the differentially expressed genes, we are going first to extract their probe ids:

```
filterd_ids = row.names(filtered) # ids of the filtered DE  
length(filterd_ids)
```

```
## [1] 276
```

```
head(filterd_ids)
```

```
## [1] "1416200_at"      "1416236_a_at"  "1417808_at"    "141793  
## [5] "1418050_at"      "1418100_at"
```

Annotation 2/3

Then we will generate a comprehensive functional annotation via BioConductor packages annaffy and mouse4302.db.

To install BioConductor packages (if they are not already installed):

```
source("http://bioconductor.org/biocLite.R")
biocLite(c("annaffy", "mouse4302.db"))
```

Load the packages and extract annotation of the filtered ids:

```
library(annaffy)
annotation_table = aafTableAnn(filterd_ids, "mouse4302.db")
saveHTML(annotation_table, file="filtered.html")
browseURL("filtered.html")
```

```
## Loading required package: Biobase
```

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

Annotation 3/3

| Probe | Symbol | Description | Chromosome | Chromosome Location | GenBank | Gene | UniGene | PubMed | Gene Ontology | Pathway |
|------------|--------|----------------|------------|-----------------------|-----------|-------|-----------|--------|---|-------------------------------|
| 1416200_at | IL33 | interleukin 33 | 19 | 29945789, 29925113 | NM_133775 | 77125 | Mm.182359 | 267 | macrophage activation involved in immune response microglial cell activation involved in immune response negative regulation of leukocyte migration negative regulation of T-helper 1 type immune response positive regulation of type 2 immune response positive regulation of type 2 immune response cytokine activity cytokine activity cytokine activity extracellular region extracellular space nucleus nucleoplasm chromosome cytosol transcription, DNA-templated positive regulation of gene expression cytoplasmic vesicle positive regulation of proteasomal ubiquitin-dependent protein catabolic process positive regulation of proteasomal ubiquitin-dependent protein catabolic process negative regulation of interferon-gamma production positive regulation of interleukin-13 production positive regulation of interleukin-4 production positive regulation of interleukin-5 production positive regulation of interleukin-6 | Cytosolic DNA-sensing pathway |

Figure 9: Annotation

Exercise

Generate separate annotations for the up-regulated and down-regulated differentially expressed genes between WT and KO. Discuss each set in a molecular context as detailed as possible.

Solution 1/2

Generating two lists of IDs for the up- and down- regulated genes

```
up_ids = ids$filter_up] # IDs of up-regulated genes  
length(up_ids)
```

```
## [1] 95
```

```
head(up_ids)
```

```
## [1] "1416200_at"      "1416236_a_at" "1418203_at"      "141939  
## [5] "1419427_at"      "1419591_at"
```

```
down_ids = ids$filter_down] # IDs of down-regulated genes  
length(down_ids)
```

```
## [1] 181
```

```
head(down_ids)
```

Solution 2/2

Obtaining the annotation of the up- and down- regulated genes into tables

```
up_table = aafTableAnn(up_ids, "mouse4302.db")
head(up_table)
saveHTML(up_table, file="up_table.html")
browseURL("up_table.html")

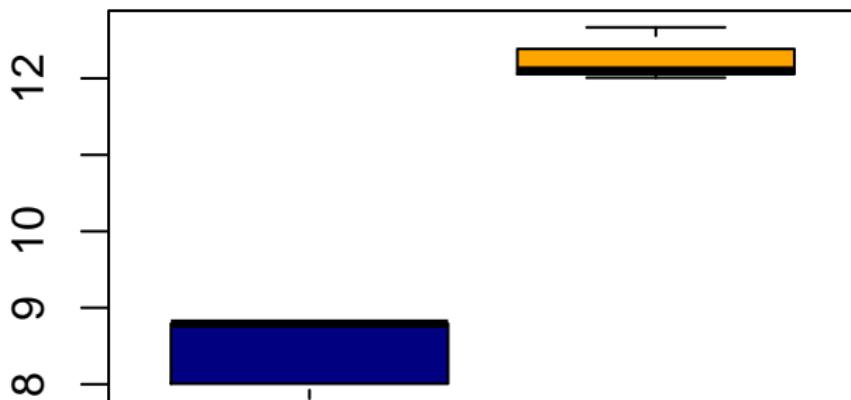
down_table = aafTableAnn(down_ids, "mouse4302.db")
head(down_table)
saveHTML(down_table, file="down_table.html")
browseURL("down_table.html")
```

```
## An object of class "aafTable"
## Slot "probeids":
## [1] "1416200_at"
##
## Slot "table":
```

Sanity Check (Irf6)

| | | | | | | | |
|--|--------------------------------|---|-----------|-----------------|-----------|----|---|
| | interferon regulatory factor 6 | I | 193153111 | NM_016851 54139 | Mm.273695 | 65 | regulatory region DNA binding DNA binding transcription factor activity, sequence-specific DNA binding nucleus cytoplasm N33861 transcription, DNA-templated regulation of transcription, DNA-templated cell cycle arrest negative regulation of cell proliferation negative regulation of cell proliferation cell differentiation keratinocyte differentiation skin development skin development keratinocyte proliferation keratinocyte proliferation positive regulation of transcription, DNA-templated cell development mammary gland epithelial cell differentiation extracellular exosome |
|--|--------------------------------|---|-----------|-----------------|-----------|----|---|

Figure 10: Down Regulation of Irf6



Multiple Testing Correction 1/3

We conducted 10^6 statistical tests. The computed p -values need to be corrected for *multiple testing*. The correction can be performed using `p.adjust`, which simply takes the original p -values a vector and returns the adjusted (corrected) p -values:

```
adjusted.pvalues = p.adjust(pvalue, method = "fdr")
```

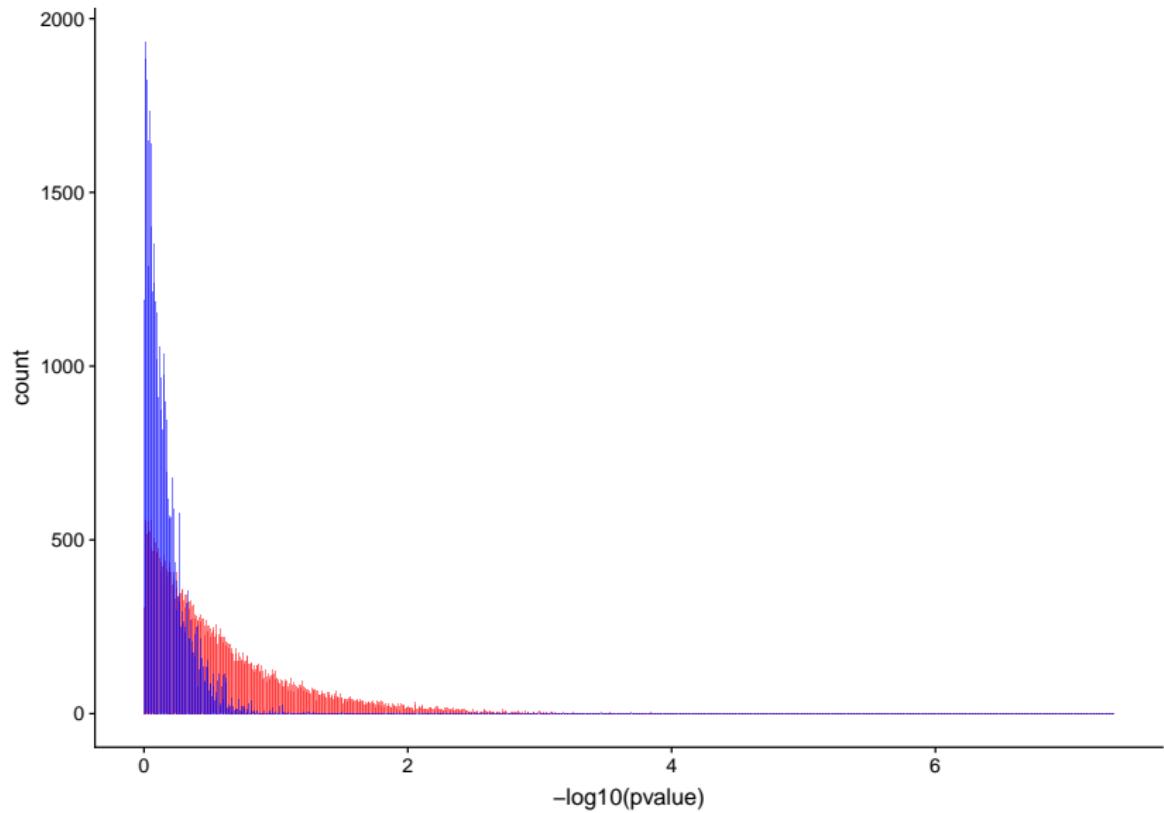
Number **original** p -values $\leq 0.05 = 5099$ while the number
adjusted (corrected) p -values $< 0.05 \geq 9$

Multiple Testing Correction 2/3

Here is an example of the original p -values and corresponding adjusted p -values:

| pvalue | adjusted.pvalue |
|-----------|-----------------|
| 0.0927063 | 0.5278755 |
| 0.1826633 | 0.6346918 |
| 0.1297791 | 0.5805456 |
| 0.2728992 | 0.7025472 |
| 0.2623772 | 0.6967834 |
| 0.0059478 | 0.2518079 |

Multiple Testing Correction 3/3



Homework

- ▶ Identify the top 10 *biologically* significant genes (i.e., by fold-change)
- ▶ Identify the top 10 *statistically* significant genes (i.e., by *p*-value)
- ▶ Identify the overlapping (if any) between the two sets of genes

Thank you!

