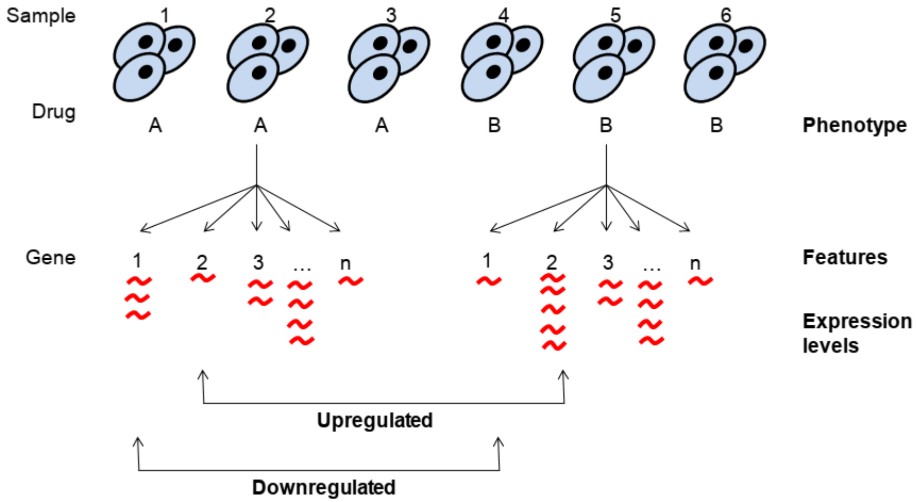
DIFFERENTIAL EXPRESSION ANALYSIS IN R WITH LIMMA

Differential expression analysis

John Blischak
Instructor


Sample
1
2
3
4
5
6
Drug
A
A
A
B
B
B
Phenotype
Gene
1
2
3
...
n
1
2
3
...
n
Features
Expression levels
Upregulated
Downregulated

What is the goal of a differential expression analysis?

- Identify the genes that are associated with a phenotype of interest
- Examples:
  - The response to a stimulus like a drug
  - Changes during development
  - The effect of a genetic mutation

Why differential expression?

- Novelty
    - Are there additional genes of interest?
- Context
    - Is the measurement for a given gene unique or common?
- Systems
    - Which biological pathways are important?

Many steps to complete an experiment

- Design study
- Perform experiment
- Collect data
- Pre-process data
- **Explore data**
- **Test data**
- **Interpret results**
- Share results

Caveats

- Measurements are relative, not absolute
- Statistical methods cannot rescue a poorly designed study

Differential Expression Methods

- Statistical Tests
  - t-test, Wilcoxon, Kolmogorov-Smirnov, F-test, etc.
  - GWAS: chi-squared
- Permutation-based
  - SAM
- Regression analysis
  - **LIMMA**
  - **DeSeq**

DeSeq is for RNA-seq, while LIMMA can be used for both microarray and RNA-seq. However, LIMMA only works using non-count data (RPKM, FPKM, TPM).

The experimental data

1. Study of breast cancer

    - Bioconductor package "breastCancerVDX"
    - Published in Wang et al., 2005 and Minn et al., 2007
    - 344 patients: 209 ER+, 135 ER-

2. Study of chronic lymphocytic leukemia (CLL)

    - Bioconductor package "CLL"
    - Drs. Sabina Chiaretti and Jerome Ritz
    - 22 patients: 8 stable, 14 progressive

Data in R

- Expression matrix (`x`)
- Feature data (`f`) - feature attributes
- Phenotype data (`p`) - sample attributes

Expression matrix

rows = features, columns = samples

```
class(x)
```

```
[1] "matrix"
```

```
x[1:5, 1:5]
```

```
                VDX_3     VDX_5     VDX_6
1007_s_at 11.965135 11.798593 11.777625
1053_at    7.895424  7.885696  7.949535
117_at     8.259272  7.052025  8.225930
```

```
dim(x)
```

```
[1] 22283   344
```

Feature data

rows = features, columns = any number of attributes

```
class(f)
```

```
[1] "data frame"
```

```
dim(f)
```

```
[1] 22283     3
```

```
f[1:3, ]
```

```
           symbol entrez   chrom
1007_s_at    DDR1    780  6p21.3
1053_at      RFC2   5982 7q11.23
117_at      HSPA6   3310    1q23
```

Phenotype data

rows = samples, columns = any number of attributes

```
class(p)
```

```
[1] "data frame"
```

```
dim(p)
```

```
[1] 344   3
```

```
# er = +/- for Estrogen Receptor
p[1:3, ]
```

```
      id age       er
VDX_3  3  36 negative
VDX_5  5  47 positive
VDX_6  6  44 negative
```

Object-oriented programming with Bioconductor classes

- **class** - defines a structure to hold complex data
- **object** - a specific instance of a class
- **methods** - functions that work on a specific class
    - **getters/accessors** - Get data stored in an object
    - **setters/** - Modify data stored in an object

```
source("https://bioconductor.org/biocLite.R")
biocLite("Biobase")
```

Create an ExpressionSet object

```
# Load package
library(Biobase)

# Create ExpressionSet object
eset <- ExpressionSet(assayData = x,
                      phenoData = AnnotatedDataFrame(p),
                      featureData = AnnotatedDataFrame(f))
```

```
# View the number of features (rows) and samples (columns)
dim(eset)
```

```
Features  Samples
   22283      344
```

```
?ExpressionSet
```

Access data from an ExpressionSet object

Expression matrix

```
x <- exprs(eset)
```

Feature data

```
f <- fData(eset)
```

Phenotype data

```
p <- pData(eset)
```

DIFFERENTIAL EXPRESSION ANALYSIS IN R WITH LIMMA

The limma package

John Blischak
Instructor

Advantages of the limma package

- Testing thousands of genes would require lots of boiler plate code

```
pval <- numeric(length = nrow(x))
r2 <- numeric(length = nrow(x))
for (i in 1:nrow(x)) {
  mod <- lm(x[i, ] ~ p[, "er"])
  result <- summary(mod)
  pval[i] <- result$coefficients[2, 4]
  r2[i] <- result$r.squared
}
```

- Improved inference by sharing information across genes

- Lots of functions for pre- and post-processing (see Ritchie et al., 2015 for an overview)

```
source("https://bioconductor.org/biocLite.R")
biocLite("limma")
```

Specifying a linear model

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

- Y - Expression level of gene
- B_0 - Mean expression level in ER-negative
- B_1 - Mean difference in expression level in ER-positive
- X_1 - ER status: 0 = negative, 1 = positive
- ϵ - Random noise

Specifying a linear model in R

```
model.matrix(~<explanatory>, data = <data frame>)
```

```
design <- model.matrix(~er, data = pData(eset))
```

```
head(design, 2)
```

```
      (Intercept) erpositive
VDX_3           1          0
VDX_5           1          1
```

```
colSums(design)
```

```
(Intercept)  erpositive
        344         209
```

```
table(pData(eset)[, "er"])
```

```
negative positive
     135      209
```

Testing with limma

```
library(limma)
```

```
# Fit the model
fit <- lmFit(eset, design)
```

```
# Calculate the t-statistics
fit <- eBayes(fit)
```

```
# Summarize results
results <- decideTests(fit[, "er"])
summary(results)
```

```
   erpositive
-1       6276
0       11003
1        5004
```

Group-means parametrization

$$Y = \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

- β_1 - Mean in ER-neg
- β_2 - Mean in ER-pos
- Test: $\beta_2 - \beta_1 = 0$

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- β_1 - Mean in group 1
- β_2 - Mean in group 2
- β_3 - Mean in group 3
- Tests:
  - $\beta_2 - \beta_1 = 0$
  - $\beta_3 - \beta_1 = 0$
  - $\beta_3 - \beta_2 = 0$

Design matrix for group-means

```
design <- model.matrix(~0 + er, data = pData(eset))
```

```
head(design)
```

```
      ernegative erpositive
VDX_3          1          0
VDX_5          0          1
VDX_6          1          0
VDX_7          1          0
VDX_8          1          0
VDX_9          0          1
```

```
colSums(design)
```

```
ernegative erpositive
       135        209
```

Contrasts matrix

```
library(limma)
cm <- makeContrasts(status = erpositive - ernegative,
                    levels = design)
```

```
cm
```

```
            Contrasts
Levels        status
  ernegative      -1
  erpositive       1
```

Testing the group-means parametrization

```
fit <- lmFit(eset, design)
```

```
head(fit$coefficients, 3)
```

```
          ernegative erpositive
1007_s_at  11.725148  11.823936
1053_at     8.126934   7.580204
117_at      7.972049   7.798623
```

```
fit2 <- contrasts.fit(fit, contrasts = cm)
```

```
head(fit2$coefficients, 3)
```

```
           Contrasts
                 status
  1007_s_at  0.09878782
  1053_at   -0.54673000
  117_at    -0.17342654
```

The parametrization does not change the results

```
# Calculate the t-statistics
fit2 <- eBayes(fit2)
```

```
# Count the number of differentially expressed genes
results <- decideTests(fit2)
summary(results)
```

```
   status
-1   6276
0   11003
1    5004
```

A study with 3 groups

- 3 different types of leukemias: ALL, AML, CML
    - Bioconductor package: leukemiasEset
    - Kohlmann et al. 2008, Haferlach et al. 2010

```
dim(eset)
```

```
Features  Samples
   20172       36
```

```
table(pData(eset)[, "type"])
```

```
ALL AML CML
 12  12  12
```

Group-means model for 3 groups

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

- β_1 - Mean expression level in group ALL
- β_2 - Mean expression level in group AML
- β_3 - Mean expression level in group CML
- Tests:
  - AML v. ALL: $\beta_2 - \beta_1 = 0$
  - CML v. ALL: $\beta_3 - \beta_1 = 0$
  - CML v. AML: $\beta_3 - \beta_2 = 0$

Group-means design matrix for 3 groups

```
design <- model.matrix(~0 + type, data = pData(eset))
```

```
head(design, 3)
```

```
          typeALL typeAML typeCML
sample_01       1       0       0
sample_02       1       0       0
sample_03       1       0       0
```

```
colSums(design)
```

```
typeALL typeAML typeCML
     12      12      12
```

Contrasts matrix for 3 groups

- AML v. ALL: $\beta_2 - \beta_1 = 0$
- CML v. ALL: $\beta_3 - \beta_1 = 0$
- CML v. AML: $\beta_3 - \beta_2 = 0$

```
library(limma)
cm <- makeContrasts(AMLvALL = typeAML - typeALL,
                    CMLvALL = typeCML - typeALL,
                    CMLvAML = typeCML - typeAML,
                    levels = design)
```

```
cm
```

```
         Contrasts
Levels    AMLvALL CMLvALL CMLvAML
  typeALL      -1      -1       0
  typeAML       1       0      -1
  typeCML       0       1       1
```

Testing 3 groups

```
library(limma)

# Fit coefficients
fit <- lmFit(eset, design)

# Fit contrasts
fit2 <- contrasts.fit(fit, contrasts = cm)

# Calculate t-statistics
fit2 <- eBayes(fit2)

# Summarize results
results <- decideTests(fit2)
summary(results)
```

```
   AMLvALL CMLvALL CMLvAML
-1     898    3401    1890
0    18323   13194   16408
1      951    3577    1874
```

The effect of hypoxia on stem cell function

- 3 different levels of oxygen: 1%, 5%, 21%
    - Bioconductor package: stemHypoxia
    - Prado-Lopez et al. 2010

```
dim(eset)
```

```
Features  Samples
   15325        6
```

```
table(pData(eset)[, "oxygen"])
```

```
ox01 ox05 ox21
   2    2    2
```

Factorial designs

- 2x2 design to study effect of low temperature in plants:
  - 2 types of *Arabidopsis thaliana*: col, vte2
  - 2 temperatures: normal, low
  - Maeda et al. 2010

```
dim(eset)
```

```
Features  Samples
   11871       12
```

```
table(pData(eset)[, c("type", "temp")])
```

```
      temp
type    low normal
  col     3      3
  vte2    3      3
```

Group-means model for 2x2 factorial

$$Y = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \beta_4 X_4 + \epsilon$$

- β_1 - Mean expression level in `col` plants at `low` temperature
- β_2 - Mean expression level in `col` plants at `normal` temperature
- β_3 - Mean expression level in `vte2` plants at `low` temperature
- β_4 - Mean expression level in `vte2` plants at `normal` temperature

Group-means design matrix for 2x2 factorial

```
group <- with(pData(eset), paste(type, temp, sep = "."))
group <- factor(group)
```

```
design <- model.matrix(~0 + group)
colnames(design) <- levels(group)
```

```
head(design, 3)
```

```
  col.low col.normal vte2.low vte2.normal
1       0          1        0           0
2       0          1        0           0
3       0          1        0           0
```

```
colSums(design)
```

```
    col.low  col.normal    vte2.low vte2.normal
          3           3           3           3
```

Contrasts for a 2x2 factorial

	β_1	β_2	β_3	β_4
`type`	col	col	vte2	vte2
`temp`	low	normal	low	normal

- Differences of `type` in normal `temp`: $\beta_4 - \beta_2 = 0$
- Differences of `type` in low `temp`: $\beta_3 - \beta_1 = 0$
- Differences of `temp` in vte2 `type`: $\beta_3 - \beta_4 = 0$
- Effect of `temp` in col `type`: $\beta_1 - \beta_2 = 0$
- Differences of `temp` between col and vte2 `type`: $(\beta_3 - \beta_4) - (\beta_1 - \beta_2) = 0$

Contrasts matrix for 2x2 factorial

```
library(limma)
cm <- makeContrasts(type_normal = vte2.normal - col.normal,
                    type_low = vte2.low - col.low,
                    temp_vte2 = vte2.low - vte2.normal,
                    temp_col = col.low - col.normal,
                    interaction = (vte2.low - vte2.normal) -
                                  (col.low - col.normal),
                    levels = design)
```

```
cm
```

```
             Contrasts
Levels         type_normal type_low temp_vte2 temp_col interaction
  col.low                0       -1         0        1          -1
  col.normal            -1        0         0       -1           1
  vte2.low               0        1         1        0           1
  vte2.normal            1        0        -1        0          -1
```

Testing 2x2 factorial

```
library(limma)

# Fit coefficients
fit <- lmFit(eset, design)

# Fit contrasts
fit2 <- contrasts.fit(fit, contrasts = cm)

# Calculate t-statistics
fit2 <- eBayes(fit2)

# Summarize results
results <- decideTests(fit2)
summary(results)
```

```
   type_normal type_low temp_vte2 temp_col interaction
-1           0      466      1635     1885         128
0        11871    10915      7635     6989       11640
1            0      490      2601     2997         103
```

Contrasts for doxorubicin study

	β_1	β_2	β_3	β_4
`genotype`	top2b	top2b	wt	wt
`treatment`	dox	pbs	dox	pbs

- Response of wild type mice to dox treatment: $\beta_3 - \beta_4 = 0$

- Response of Top2b null mice to dox treatment: $\beta_1 - \beta_2 = 0$

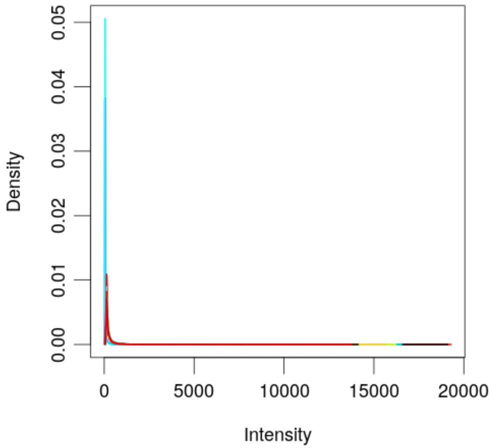
- Differences between Top2b null and wild type mice in response to dox treatment: $(\beta_1 - \beta_2) - (\beta_3 - \beta_4) = 0$

Pre-processing steps

- Log transform
- Quantile normalize
- Filter

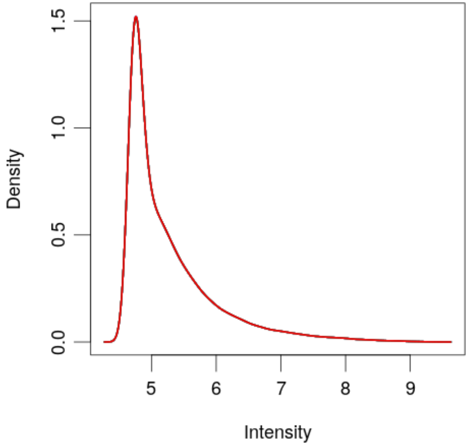
Visualization

```r
library(limma)

# Plot distribution of each sample
plotDensities(eset, legend = FALSE)
```



Quantile normalize

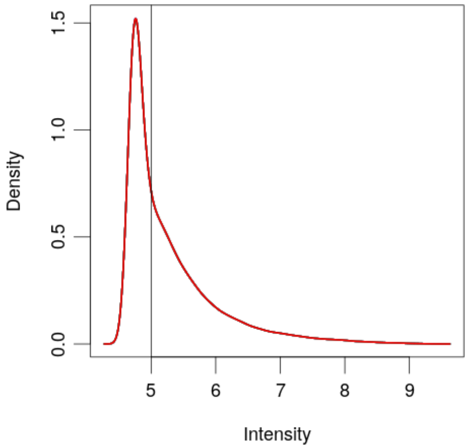
```r
# Quantile normalize
exprs(eset) <- normalizeBetweenArrays(exprs(eset))

plotDensities(eset, legend = FALSE)
```

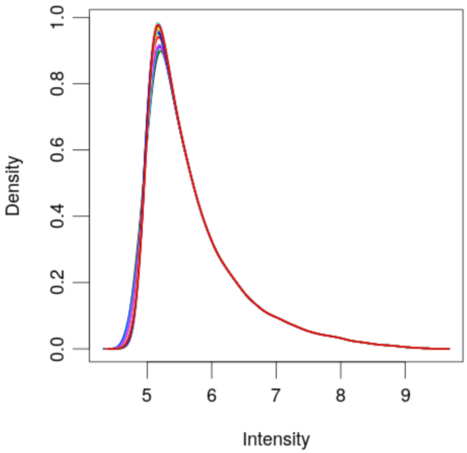


Filter genes

```
# View the normalized data
plotDensities(eset, legend = FALSE)
abline(v = 5)
```

```
# Create logical vector
keep <- rowMeans(exprs(eset)) > 5
# Filter the genes
eset <- eset[keep, ]
plotDensities(eset, legend = FALSE)
```





What are technical batch effects?

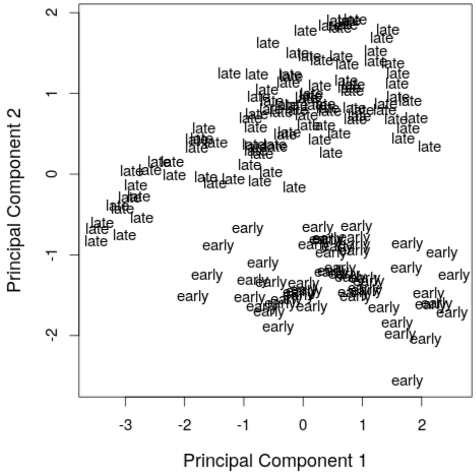
- Every batch of an experiment is slightly different
- Need to balance variables of interest across batches
- If properly balanced, batch effects can be removed

Diagnosing technical batch effects

- Dimension reduction techniques:
    - Principal Components Analysis (PCA)
    - MultiDimensional Scaling (MDS)
- Identify the largest sources of variation in a data set
- Are the largest sources of variation correlated with the variables of interest or technical batch effects?

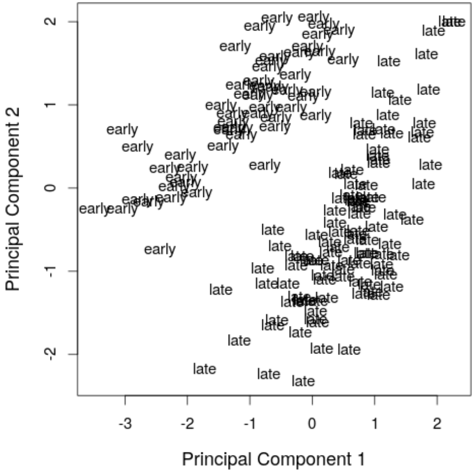
plotMDS

```r
library(limma)
plotMDS(eset, labels = pData(eset)[, "time"], gene.selection = "common")
```



removeBatchEffect

```
exprs(eset) <- removeBatchEffect(eset, batch = pData(eset)[, "batch"],
                                 covariates = pData(eset)[, "rin"])

plotMDS(eset, labels = pData(eset)[, "time"], gene.selection = "common")
```



Inspecting the results

```
results <- decideTests(fit2)
summary(results)
```

```
   status
-1   6276
0   11003
1    5004
```

```
topTable(fit2, number = 3)
```

```
            symbol entrez  chrom    logFC  AveExpr        t
205225_at     ESR1   2099 6q25.1 3.762901 11.37774 22.68392
209603_at    GATA3   2625  10p15 3.052348  9.94199 18.98154
209604_s_at  GATA3   2625  10p15 2.431309 13.18533 17.59968
                 P.Value    adj.P.Val        B
205225_at   2.001001e-70 4.458832e-66 149.1987
209603_at   1.486522e-55 1.656209e-51 115.4641
209604_s_at 5.839050e-50 4.337052e-46 102.7571
```

Obtain results for all genes

```
stats <- topTable(fit2, number = nrow(fit2), sort.by = "none")
```

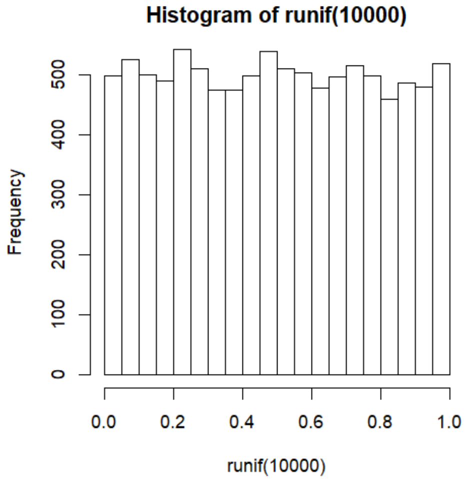
```
dim(stats)
```

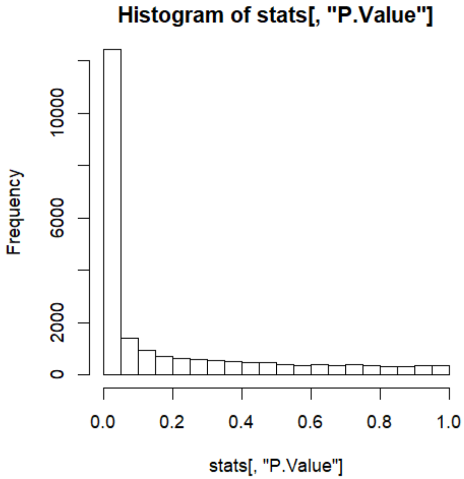
```
[1] 22283     9
```

Histogram of p-values

```
hist(runif(10000))
```

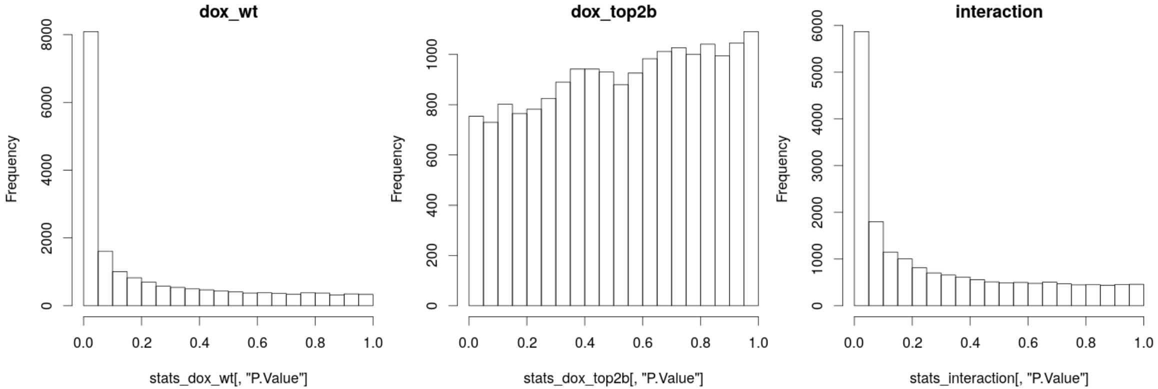
```
hist(stats[, "P.Value"])
```





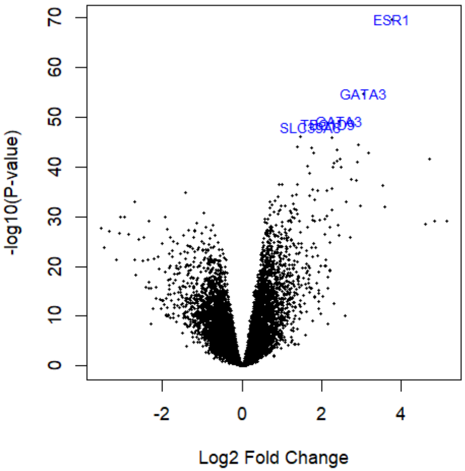
Histograms of p-values

- `topTable` and `hist`



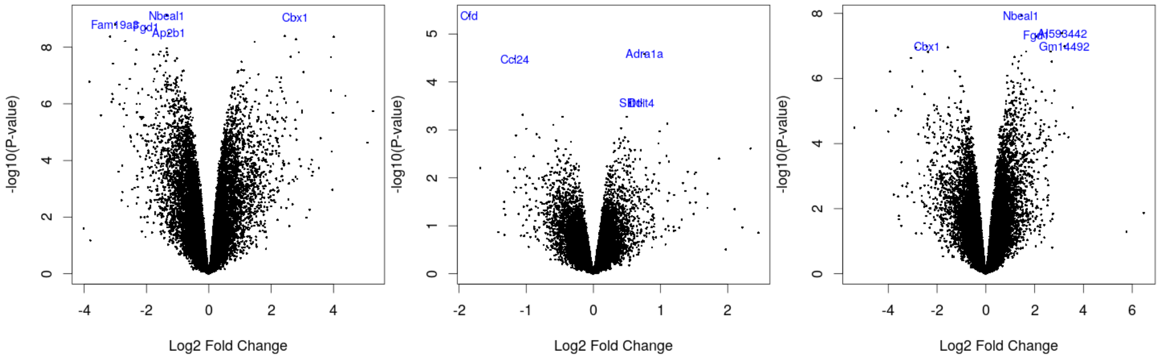
Volcano plot

```
volcanoplot(fit2, highlight = 5, names = fit2$genes[, "symbol"])
```



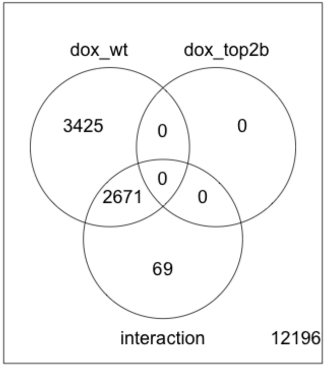
Volcano plots

```
# Extract the gene symbols
gene_symbols <- fit2$genes[, "symbol"]

# Create a volcano plot for the contrast dox_wt
volcanoplot(fit2, coef = "dox_wt", highlight = 5, names = gene_symbols)
```



Inspect the results

```
# Create a Venn diagram
vennDiagram(results)
```



Interpreting the results

```
results <- decideTests(fit2)
summary(results)
```

```
   status
-1   6276
0   11003
1    5004
```

```
topTable(fit2, number = 3)
```

```
            symbol entrez  chrom    logFC  AveExpr        t
205225_at     ESR1   2099 6q25.1 3.762901 11.37774 22.68392
209603_at    GATA3   2625  10p15 3.052348  9.94199 18.98154
209604_s_at  GATA3   2625  10p15 2.431309 13.18533 17.59968
                 P.Value    adj.P.Val        B
205225_at   2.001001e-70 4.458832e-66 149.1987
209603_at   1.486522e-55 1.656209e-51 115.4641
209604_s_at 5.839050e-50 4.337052e-46 102.7571
```

Biological databases

- KEGG: Kyoto Encyclopedia of Genes and Genomes
  - https://www.genome.jp/kegg/
  - Ex: Photosynthesis, Protein transport
- Gene Ontology Consortium (GO)
  - http://geneontology.org/
  - Ex: response to stress, developmental process

Enrichment testing

	In gene set	Not in gene set
DE	30	70
all	100	900

```
fisher.test(matrix(c(30, 100, 70, 900), nrow = 2))
```

```


Fisher's Exact Test for Count Data

data:  matrix(c(30, 100, 70, 900), nrow = 2)
p-value = 1.88e-07
alternative hypothesis: true odds ratio is not equal to 1
95 percent confidence interval:
 2.306911 6.320992
sample estimates:
odds ratio
  3.850476
```

Testing for KEGG enrichment

```
head(fit2$genes, 3)
```

```
          symbol entrez   chrom
1007_s_at   DDR1    780  6p21.3
1053_at     RFC2   5982 7q11.23
117_at     HSPA6   3310    1q23
```

```
entrez <- fit2$genes[, "entrez"]
```

```
enrich_kegg <- kegga(fit2, geneid = entrez, species = "Hs")
```

```
topKEGG(enrich_kegg, number = 3)
```

```
                         Pathway    N  Up Down         P.Up       P.Down
path:hsa04110         Cell cycle  115  30   82 6.192773e-01 5.081518e-12
path:hsa05166   HTLV-I infection  233  55  135 8.959082e-01 9.285167e-09
path:hsa01100 Metabolic pathways 1033 350  373 3.175782e-08 9.969693e-01
```

Testing for GO enrichment

```
enrich_go <- goana(fit2, geneid = entrez, species = "Hs")
```

```
topGO(enrich_go, ontology = "BP", number = 3)
```

```
                             Term Ont    N  Up Down P.Up       P.Down
GO:0002376  immune system process  BP 1935 426  914    1 7.925179e-32
GO:0006955        immune response  BP 1236 230  619    1 3.625368e-29
GO:0045087 innate immune response  BP  645 113  346    1 1.635833e-22
```

Test for enrichment of gene sets

```
# Extract the entrez gene IDs
entrez <- fit2$genes[, "entrez"]

# Test for enriched KEGG Pathways for contrast dox_wt
enrich_dox_wt <- kegga(fit2, coef = "dox_wt", geneid = entrez,
                       species = "Mm")

# View the top 5 enriched KEGG pathways
topKEGG(enrich_dox_wt, number = 5)
```

```
                                                             Pathway
path:mmu05322                           Systemic lupus erythematosus
path:mmu03008                      Ribosome biogenesis in eukaryotes
path:mmu05034                                             Alcoholism
path:mmu05412 Arrhythmogenic right ventricular cardiomyopathy (ARVC)
path:mmu05330                                    Allograft rejection
                N Up Down         P.Up       P.Down
path:mmu05322  76 37    1 3.657708e-10 9.999999e-01
path:mmu03008  71 34    4 3.320811e-09 9.997410e-01
path:mmu05034 130 47   17 2.358456e-07 9.733029e-01
path:mmu05412  52  2   26 9.995025e-01 4.140466e-07
path:mmu05330  26 16    0 6.720834e-07 1.000000e+00
```

Caveats

- Don't overinterpret
- Be skeptical of up- vs. down-regulated
- The background set of genes should only include tested genes
- More advanced methods available, including limma functions `camera` and `roast`