RNA-SEQ DE WITH EDGER

V. STORME

READING THE COUNTS FROM A FILE

- edgeR requires a table of **integer** read counts
  - Rows corresponding to genes
  - Columns corresponding to independent libraries (samples)

READING THE COUNTS FROM A FILE

- Count data contained in a single tab-delimited or comma-separated text

geneID	A1	A2	A3	B1	B2	B3
ID0001	20	25	23	100	102	105
ID0002	30	31	27	12	10	9
…	…	…	…	…	…	…

```
> x <- read.delim("fileofcounts.txt", row.names="geneID")
> x <- read.csv("fileofcounts.csv", row.names="geneID")
> group <- factor(c(1,1,1,2,2,2))
> y <- DGEList(counts=x, group=group)
```

READING THE COUNTS FROM A FILE

- Counts for different samples stored in separate files:

A1.txt

geneID	counts
ID0001	20
ID0002	30
…	…

targets.txt

files	group	description
A1.txt	A	Treatment A rep 1
A2.txt	A	Treatment A rep 2
A3.txt	A	Treatment A rep 3
B1.txt	B	Treatment B rep 1
B2.txt	B	Treatment B rep 2
B3.txt	B	Treatment B rep 3

> targets <- read.delim("targets.txt")

> d <- readDGE(targets)

THE DGELIST DATA CLASS

- edgeR stores data in a simple list-based data object called a ***DGEList***
- Function ***readDGE*** makes a DGEList object directly
- Table of counts available as a matrix or a data.frame:
  - ***y <- DGEList(counts=x, group=group)***
- Components:
  - A matrix ***counts*** containing the integer counts
  - A data.frame ***samples*** containing info about the samples or libraries
    - Contains a column lib.size for the library size computed from the column sum of the counts
  - Optional: a data.frame genes containing annotation

MODELLING COUNTS
THE POISSON DISTRIBUTION

- Famous example by von Bortkiewicz (1898): observe the number of soldiers in the Prussian army who got kicked by horses over a number of years and corps

# kicks (=k)	# soldiers	fraction	Expected fraction
0	109	0.545	0.543
1	65	0.325	0.331
2	22	0.110	0.101
3	3	0.015	0.021
4	1	0.005	0.003

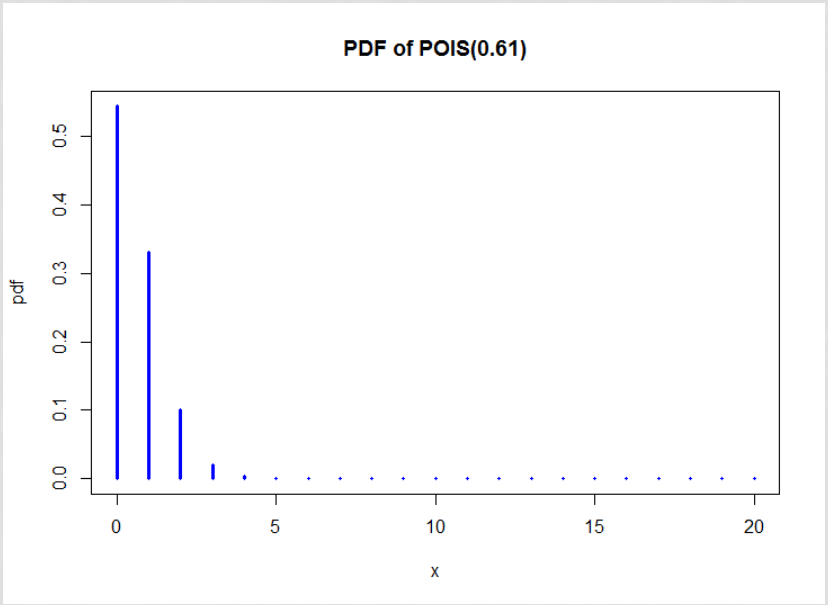
- Average nr of horsekicks per soldier:

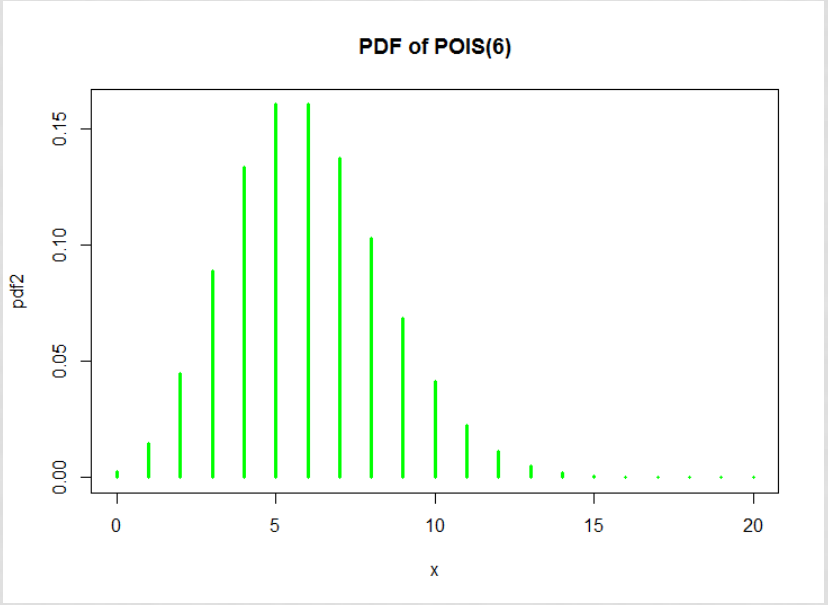
$$\bar{X} = \frac{0*109+1*65+2*22+3*3+4*1}{200} = 0.61$$

- The probability that the nr of kicks=k

$$P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!} \qquad \hat{\lambda} = \bar{X}$$

THE POISSON DISTRIBUTION





MODELLING RNA-SEQ COUNTS

- Let y_{gi} be the number of reads that map to gene g in sample i

$$f(y_{gi}|\mu_{gi}) = P(Y_{gi} = y_{gi}|\mu_{gi}) = \frac{\mu_{gi}^{y_{gi}} e^{-\mu_{gi}}}{y_{gi}!}$$

$$E(y_{gi}) = var(y_{gi}) = \mu_{gi}$$

- Overdispersion:
  - the observed variance is larger than expected.
    - SE is underestimated
    - test statistic is overestimated
    - the type I error is increased and thus also the false discovery rate

NEGATIVE BINOMIAL MODEL

- Is a generalization of the Poisson distribution
  - It allows the mRNA proportions to vary across samples, capturing better the variability across biological replicates

$$var(y_{gi}) = \mu_{gi} + \phi\mu_{gi}^2$$

  - ϕ is the dispersion and $\sqrt{\phi}$ is the biological coefficient of variation (BCV)

GENERALIZED LINEAR MODELS (GLM)

- A glm consists of 3 parts
  - A ***distribution***, specifying the conditional distribution of the the response Y given the predictor variables
  - A ***linear predictor***

$$\eta = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$

  - A ***link function*** g, linking the conditional expected value of Y to η

$$g(E[Y|X]) = \eta$$

- Poisson model:

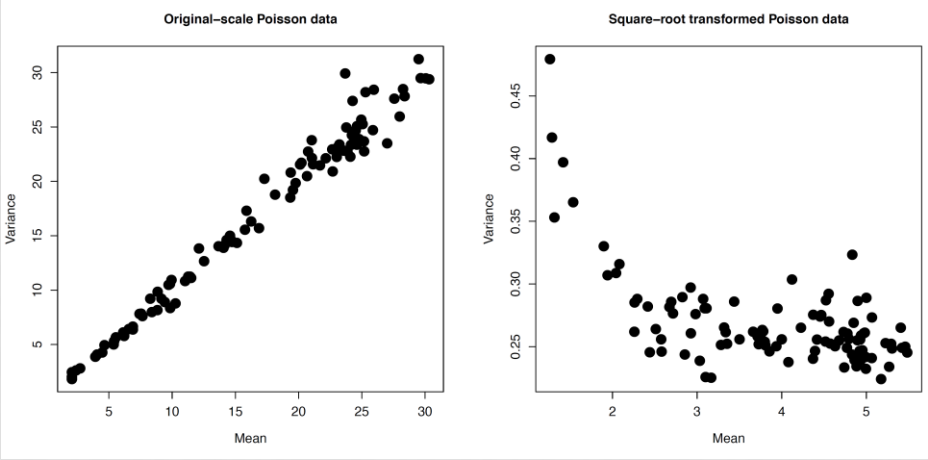
$$log(E[Y|X) = \beta_0 + \beta_1 x_1 + ... + \beta_p x_p$$
$$E(Y|X) = exp(\beta_0 + \beta_1 x_1 + ... + \beta_p x_p)$$

GLM FOR RNA-SEQ

- Distribution: negative binomial
- Link function: log
  - The link function transforms the mean, not the observed values

$$log(E[Y|X]) \neq E[log(Y|X)]$$

  - Transforming the observed values changes the association between mean and variance



NORMALIZATION

- Observed read counts depend on:
  - Abundance
  - Sequencing depth
  - Gene length
  - GC content
- edgeR is concerned with DE and not with the quantification of expression levels,
  - therefore no correction needed for gene length and GC content

NORMALIZATION

- There is correction for:
  - Sequencing depth represented by the library size
  - RNA composition: highly expressed genes can consume a substantial proportion of the total library size, causing the remaining genes to be under-sampled
    - Corrected for by **Trimmed Means of M values (TMM)**
- Normalization takes the form of correction factors that enter into the statistical model as ***offsets***

POISSON REGRESSION FOR RATES

- Rates provide the necessary standardization to make the outcomes comparable
- Rates are counts divided by some measure of exposure, in this case the library size
- Poisson model for rates:

$$log(E[Y/L|X]) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$
$$log(E[Y|X]) - log(L) = \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$
$$log(E[Y|X]) = \underbrace{log(L)}_{\text{offset}} + \beta_0 + \beta_1 X_1 + ... + \beta_p X_p$$

OFFSET

- Assume that we have RNA-seq reads for one gene, Is the gene differentially expressed?

```
count.data <- data.frame(counts = c(369, 287, 348, 433, 555, 294, 419),
                         cond = c("1", "1", "1", "1", "2", "2", "2"))
glm.pois <- glm(counts ~ cond, family = poisson, data = count.data)
coefficients(summary(glm.pois))

##             Estimate Std. Error z value  Pr(>|z|)
## (Intercept)   5.8840    0.02638 223.050 0.000e+00
## cond2         0.1626    0.03853   4.219 2.451e-05
```

OFFSET

- Incorporate library size as offset

```
count.data$lib.size <- c(3040296, 2717092, 3016179, 3707895,
                         4422272, 3467730, 3879114)
glm.pois <- glm(counts ~ cond + offset(log(lib.size)), family = poisson,
                data = count.data)
coefficients(summary(glm.pois))

##             Estimate Std. Error  z value Pr(>|z|)
## (Intercept) -9.06944    0.02638 -343.802  0.00000
## cond2       -0.06635    0.03853   -1.722  0.08506
```

$$log(E[Y|X]) = \beta_0 + \beta_1 x_1 + log(libsize)$$

$$log(E[\frac{Y}{libsize}|X]) = \beta_0 + \beta_1 x_1$$

- The counts are not explicitly scaled

TMM NORMALIZATION

- Set of trimmed genes
  - Remove the genes with 0 counts
  - Calculate for each remaining gene g and sample i the M and A values compared to a reference sample r
  - Calculate for each sample i the percentiles of the M and A values
  - Trim the M values by 30% and the A values by 5 %
  - Now G* genes are retained

gene	M	A
1	M^r_{1i}	A^r_{1i}
…	…	…
g	M^r_{gi}	A^r_{gi}
…	…	…
G	M^r_{Gi}	A^r_{Gi}

$$M^r_{gi} = log_2 \frac{y_{gi}/N_i}{y_{gr}/N_r}$$

$$A^r_{gi} = \frac{1}{2} log_2 \left(\frac{y_{gi}}{N_i} * \frac{y_{gr}}{N_r} \right)$$

!Assumption!
majority of the genes are **not DE**

TMM NORMALIZATION

(MAZA 2016, FRONTIERS IN GENETICS)

$$Y_{gkr} = \frac{X_{gkr}}{N_{kr}}$$

$$Y_g^{\mathrm{TMM}} = Yg11$$

$$\tau_{kr}^{\mathrm{TMM}} = \frac{1}{\#\mathcal{G}_{kr}^*} \sum_{g \in \mathcal{G}_{kr}^*} \frac{Y_{gkr}}{Y_g^{\mathrm{TMM}}}$$

where $\mathcal{G}_{kr}^*$ represents the set of not trimmed genes

$$\tilde{\tau}_{kr}^{\mathrm{TMM}} = \frac{\tau_{kr}^{\mathrm{TMM}}}{\tilde{\tau}^{\mathrm{TMM}}} \text{ where}$$

$$\tilde{\tau}^{\mathrm{TMM}} = \sqrt[KR]{\prod_{k=1}^{K} \prod_{r=1}^{R} \tau_{kr}^{\mathrm{TMM}}}$$

$$e_{kr}^{\mathrm{TMM}} = \tilde{\tau}_{kr}^{\mathrm{TMM}} N_{kr}$$

$$f_{kr}^{\mathrm{TMM}} = \tilde{\tau}_{kr}^{\mathrm{TMM}}$$

1. Normalise by library size
2. Choose a ref sample
3. Take the sum of the ratio of counts between a sample and the reference sample for each gene in the set of G genes
4. Adjust to multiply to 1
   - K conditions
   - R replicates
5. *Effective library size*
6. TMM normalization factor

NORMALIZATION AS OFFSET

- > ***calcNormFactors()***

$$log(E[Y|X]) = \beta_0 + \beta_1 X_1 + log(eff.libsize)$$

QUASI NEGATIVE BINOMIAL

- The NB model can be extended with quasi-likelihood methods to account for gene-specific variability for both biological and technical resources

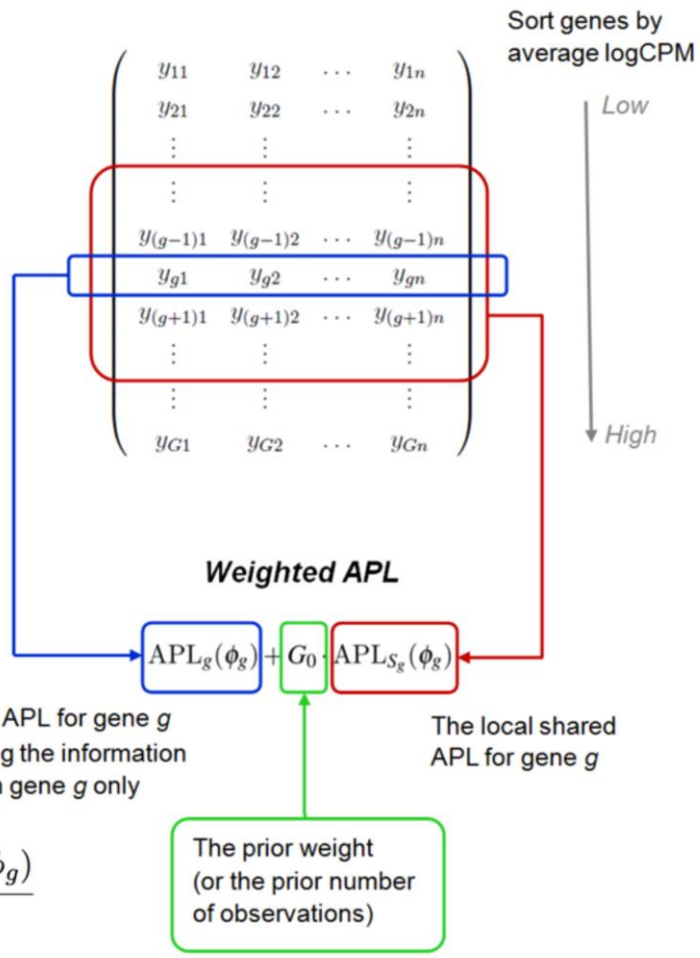
$$var(y_{gi}) = \sigma_g^2(\mu_{gi} + \phi\mu_{gi}^2)$$

- Where ϕ is the NB trended dispersion and σ^2_g is the gene-specific QL dispersion
- Estimation of the QL dispersion is difficult (empirical Bayes approach)
- **Minimum 3 replicates required**
- **Better FDR control**
- The estimation of QL dispersions is performed using the ***glmQLFit*** function

NB DISPERSIONS

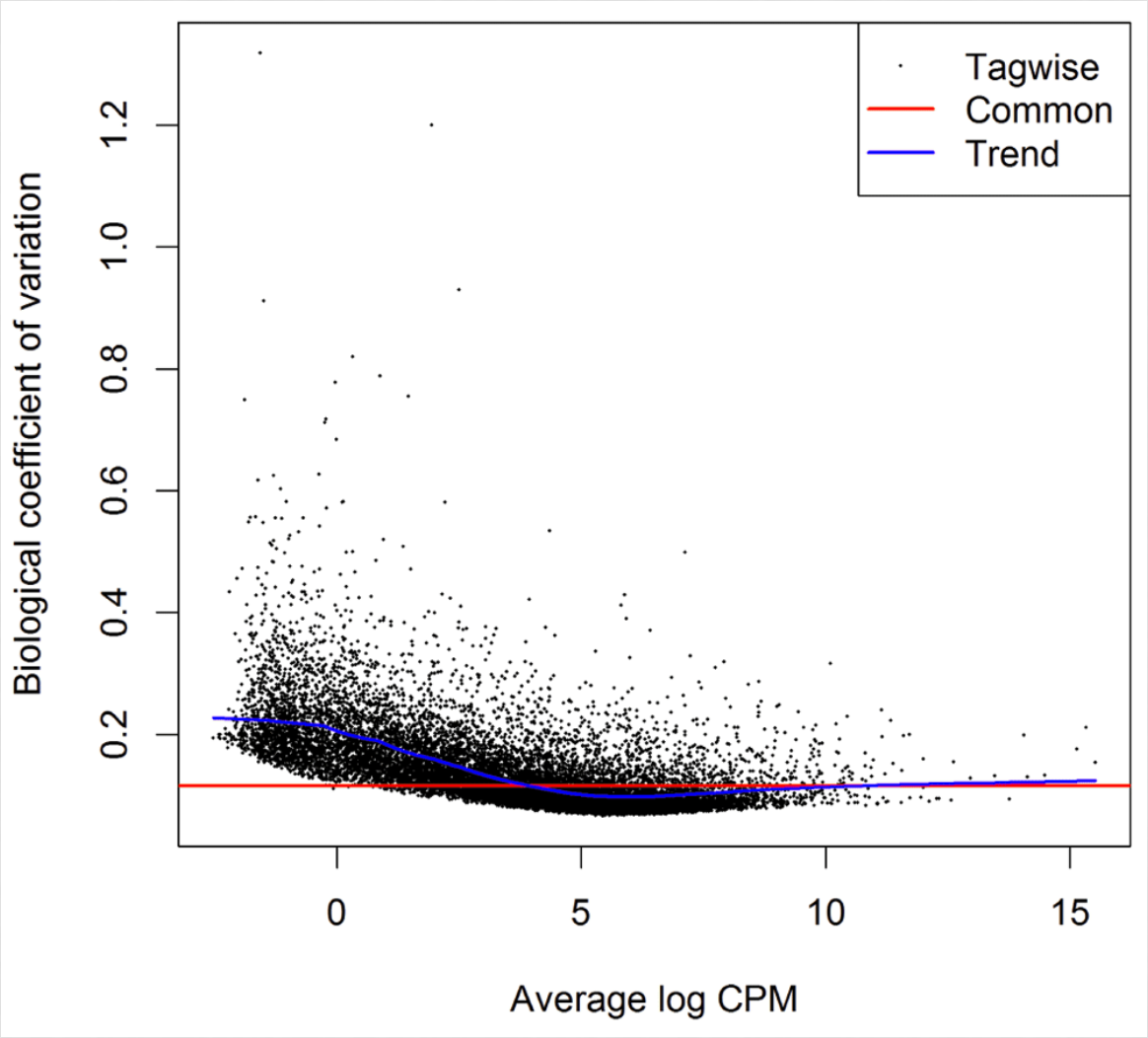
```
> estimateDisp()
```

edgeR: dispersion estimation



$$\mathrm{APL}_{S_g}(\phi_g) = \frac{\sum_{a \in C_g} w_a \cdot \mathrm{APL}_a(\phi_g)}{\sum_{a \in C_g} w_a}$$

Figure 4. Scatterplot of the biological coefficient of variation (BCV) against the average abundance...



Chen Y, Lun ATL and Smyth GK. From reads to genes to pathways: differential expression analysis of RNA-Seq experiments using Rsubread and the edgeR quasi-likelihood pipeline [version 2]. F1000Research 2016, 5:1438 (doi: 10.12688/f1000research.8987.2)

QL DISPERSIONS

- > ***plotQLDisp***
  - The raw QL dispersion estimates are squeezed towards a global trend
    - reduces the uncertainty of the estimates
    - improves testing power.
  - The extent of the squeezing is governed by the value of the *prior.df* estimated from the data.
    - Large prior.df:
      - QL dispersions are less variable between genes
      - strong EB moderation should be performed.
    - Smaller prior.df:
      - true unknown dispersions are highly variable
      - weaker moderation towards the trend is appropriate
- > ***glmQLFit(…robust=TRUE)***
  - allows gene-specific prior df estimates
  - lower values for outlier genes

MODULE FORMULAS AND DESIGN MATRICES

- Design matrices can be defined in many equivalent ways (different parameterization)
  - ***> model.matrix()***
- The contrasts need to be defined accordingly

DUMMY VARIABLES

- a dummy variable is one that takes only the value 0 or 1 to indicate the absence or presence of some categorical effect
  - Eg treatment-control experiment
    - Intercept model:
      - X=0 for a sample belonging to the control group
      - X=1 for a sample belonging to the treatment group
    - No intercept model:
      - X_1=1 for a sample belonging to the control group and 0 otw
      - X_2=1 for a sample belonging to the treatment group and 0 otw
  - Eg control – treatment A – treatment B
    - X_1=1 for a sample belonging to the A group and 0 otw
    - X_2=1 for a sample belonging to the B group and 0 otw

ANOVA AND REGRESSION

- A regression model in which the dependent variable is quantitative in nature but all the explanatory variables are dummies
- In an intercept model: number of dummies=k-1 where k represents the number of factor levels
- In a no intercept model: number of dummies=k

DESIGN MATRICES- EXAMPLE 1

- Assume treatment: control and treated
- Formula: y ~ 0 + treatment
  - Indicates no intercept

obs	sample	treatment
1	C1	control
2	C2	control
3	C3	control
4	T1	treated
5	T2	treated
6	T3	treated

		X_1	X_2
obs		treatmentcontrol	treatmenttreated
1		1	0
2		1	0
3		1	0
4		0	1
5		0	1
6		0	1

DESIGN MATRICES- EXAMPLE 1

- Assume treatment: control and treated
- Formula:

$$log(E[Y|X]) = \beta_1 X_1 + \beta_2 X_2 + log(eff.libsize)$$

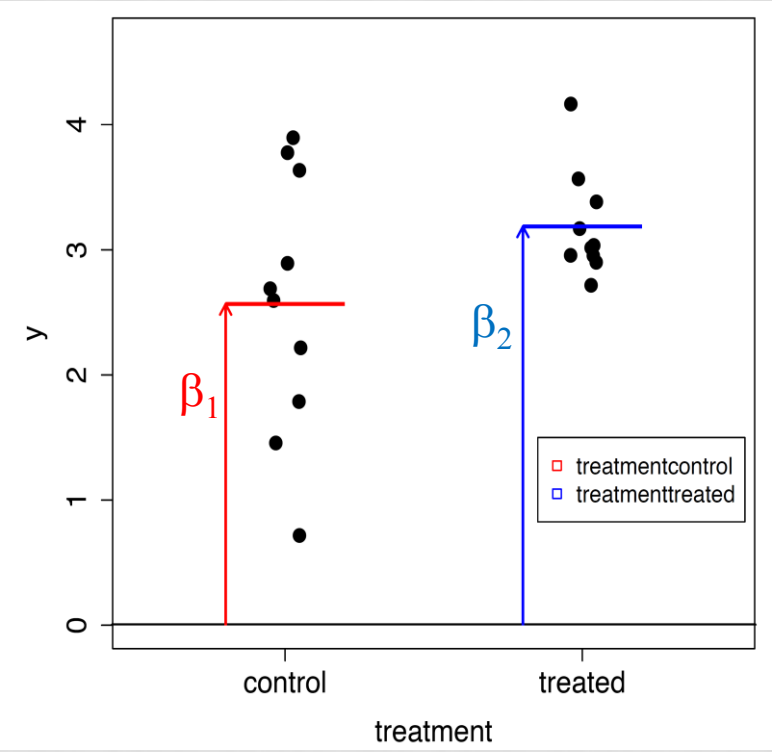
	X_1	X_2
obs	treatmentcontrol	treatmenttreated
1	1	0
2	1	0
3	1	0
4	0	1
5	0	1
6	0	1

$\times$

β
β_1
β_2

DESIGN MATRICES- EXAMPLE 1

$$log(E[Y|X = control]) = \beta_1 + log(eff.libsize)$$
$$log(E[Y|X = treated]) = \beta_2 + log(eff.libsize)$$
$$H_0 : log(E[Y|X = treated]) - log(E[Y|X = control]) = \beta_2 - \beta_1 = 0$$



- ***> TvsC <- makeContrasts(treatmenttreated – treatmentcontrol, levels=design)***
- ***> glmQLFTest(fit,contrast = TvsC)***
- ***> glmQLFTest(fit,contrast =c(-1,1))***

DESIGN MATRICES- EXAMPLE 2

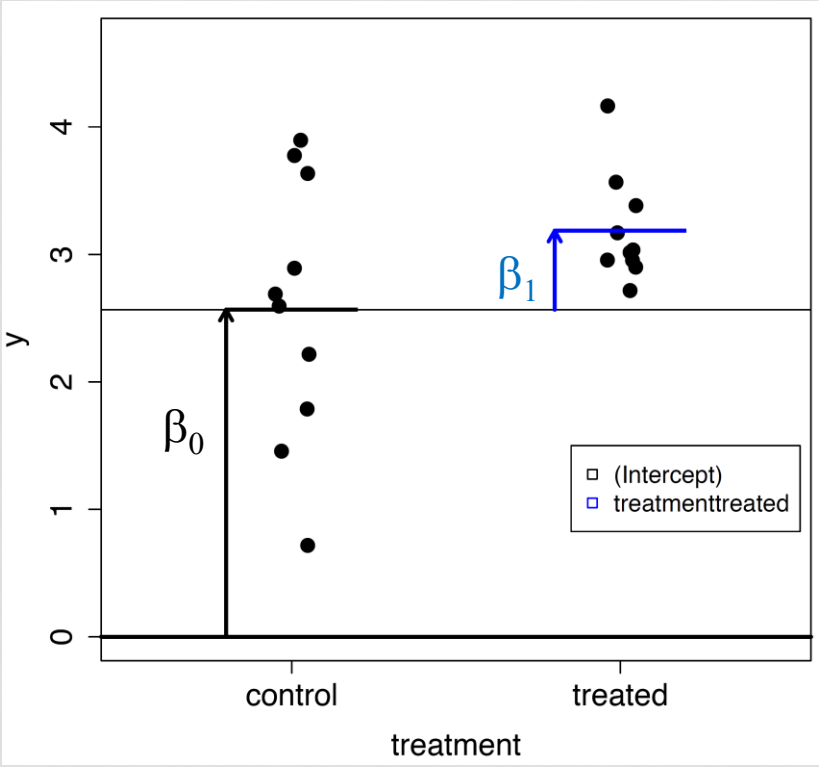
- Assume treatment: control and treated
- Formula: y ~ treatment
  - With intercept

obs	sample	treatment
1	C1	control
2	C2	control
3	C3	control
4	T1	treated
5	T2	treated
6	T3	treated

obs	Intercept	treatmenttreated
1	1	0
2	1	0
3	1	0
4	1	1
5	1	1
6	1	1

DESIGN MATRICES- EXAMPLE 2

$$log(E[Y|X = control]) = \beta_0 + log(eff.libsize)$$
$$log(E[Y|X = treated]) = \beta_0 + \beta_1 + log(eff.libsize)$$
$$H_0 : log(E[Y|X = treated]) - log(E[Y|X = control]) = \beta_1 = 0$$



- *> glmQLFTest(fit,coef=2)*

MULTIPLE HYPOTHESIS TESTING

- p-value
  - the probability of obtaining a test statistic at least as extreme as the one observed if the null hypothesis is true
- p=0.05
  - there is a 5% chance of getting that extreme result even in the absence of a real effect, a 5% chance of rejecting the null hypothesis while in fact it is true (= ***type 1 error***).
- Performing 10000 tests (one for each gene) and assuming that there is no true signal in the data might lead to 500 p-values below 0.05

MULTIPLE HYPOTHESIS TESTING

	accepted	rejected	total
True nulls	U	V (type I error)	m0
False nulls	T (type II error)	S	m1
	m - R	R	**m tests**

- Familywise error rate (FWER)
  - The probability of making at least one type I error

$$FWER = P[V \geq 1]$$

- False discovery rate (FDR)
  - Expected proportion of type I errors among the rejected hypotheses (if R=0 then FDR=0)

$$FDR = E\left[\frac{V}{R}\right]$$

- ***> topTags()***

CLUSTERING, HEATMAPS,…

- **> plotMDS()** draws a multi-dimensional scaling plot of the RNA samples
  - Default: distances correspond to **leading log-fold changes between each pair of samples** (by default top=500)

    Leading log-fold changes ie root-mean-square value RMS

    $d_{12}=\sqrt{(logFC_1^2 + logFC_2^2 + \cdots + logFC_{500}^2)/500}$

    Separate set of genes for each pairwise comparison

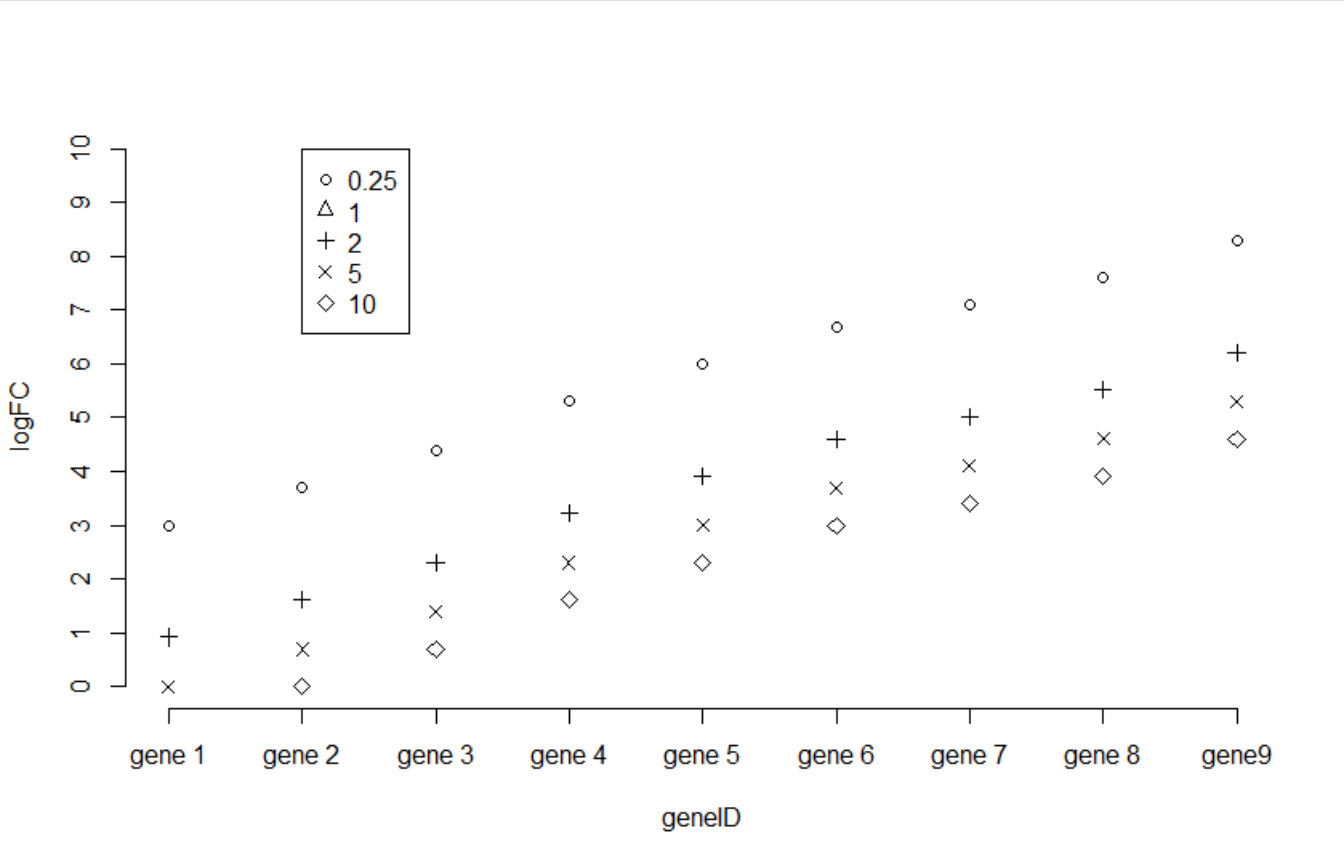
    selection.genes="pairwise" (default)
  - Option: distances in terms of BCV
    - **selection.genes="common"** selects the top genes with the largest standard deviation between samples

INPUT FOR POST-PROCESSING

- Which counts should be used as input for clustering or heatmap routines?
  - Still a matter of research
  - edgeR manual suggests using moderated log-counts-per-million
    - By default normalized library sizes are used
    - **> y <- cpm(d, prior.count=2, log=TRUE)**
  - **My suggestion:**
    - Use the fitted values normalised to a libsize of 1000000 counts
    - **$Log(E(y_{gi})/ N_i) = x^T_i \beta_g$**

```
> N <- dim(y$counts)[[1]]
> gene.fitted <- matrix(rep(NA,N*12),nrow=N)
> for (i in 1:N)
>  {
>  beta <- as.matrix(fit$coefficients[i,])
>  gene.fitted[i,] = exp(t(design %*% beta))*1000000
> }
```

NOTE ON PRIOR.COUNT



DEMO

- Data1 (data1.R)
  - control-treatment case
  - 3 independent biological samples for each treatment group
  - Analysis with a glm model
- Data2 (data2.R)
  - control-treatment case and a batch effect
  - 3 independent biological samples for each treatment group
  - Analysis with a glm model
- Data4 (data4.R)
  - 3 mutant lines and 1 ref line
  - 3 independent biological samples for each line
  - Analysis with a glm model

EXERCISES

- Analyse data 3:
  - control-treatment case
  - 3 independent biological samples for each treatment group
  - Count files and target file are in the *EXERCISES/data3* folder
  - Use an intercept model
- Analyse data 5:
  - 1 factor with 2 factor levels and a batch effect
  - 3 independent biological samples
  - Use ***data5_input.R*** in *EXERCISES/data5* folder to read the data
- Extra questions data 4:
  - set up a contrast for C vs A
  - Re-analyse the data using a no intercept model, and compare C vs A

REFERENCES

- McCarthy, D. J., Chen, Y., & Smyth, G. K. (2012). **Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation**. Nucleic acids research, 40(10), 4288–97. doi:10.1093/nar/gks042
- Robinson, M. D., & Smyth, G. K. (2007). **Moderated statistical tests for assessing differences in tag abundance**. Bioinformatics (Oxford, England), 23(21), 2881–7. doi:10.1093/bioinformatics/btm453
- Robinson, M. D., & Smyth, G. K. (2008). **Small-sample estimation of negative binomial dispersion, with applications to SAGE data**. Biostatistics (Oxford, England), 9(2), 321–32. doi:10.1093/biostatistics/kxm030
- Robinson, M. D., & Oshlack, A. (2010). **A scaling normalization method for differential expression analysis of RNA-seq data**. Genome biology, 11(3), R25. doi:10.1186/gb-2010-11-3-r25
- Lun, ATL., Chen, Y., & Smyth, G. K. (2016). **It's DE-licious: a recipe for differential expression analyses of RNA-seq experiments using quasi-likelihood methods in edgeR**. Methods in Molecular Biology, in press