Differential Expression Analysis using *edgeR*

(examples taken from the *edgeR* User's Guide, First edition 17 September 2008)

Report Generated August 12, 2014

Contents

1 Introduction

This practical presents examples from the *edgeR* [1] user's guide. To view the latest version of the user's guide in full, you can use the following commands

```
library(edgeR)
edgeRUsersGuide()
```

2 deepSAGE of wild-type vs Dclk1 transgenic mice

2.1 Introduction

This section provides a detailed analysis of data from an experiment using deep-sequenced tag-based expression profiling [2]. The biological question addressed was the identification of transcripts differentially expressed in the hippocampus between wild-type mice and transgenic mice over-expressing a splice variant of the δC-doublecortin-like kinase-1 (Dclk1) gene. The splice variant, DCLK- short, makes the kinase constitutively active and causes subtle behavioural phenotypes.

The tag-based gene expression technology in this experiment could be thought of as a hybrid between SAGE and RNA-seq like SAGE it uses short sequence tags (around 7bp) to identify transcripts, but it uses the deep sequencing capabilities of the Solexa/Illumina 1G Genome Analyzer greatly to increase the number of tags that can be sequenced. The RNA samples came from wild-type male C57/BL6j mice and transgenic mice over- expressing DCLK-short with a C57/BL6j background. Tissue samples were collected from four individuals in each of the two groups by dissecting out both hippocampi from each mouse. Total RNA was isolated and extracted from the hippocampus cells and sequence tags were prepared using Illumina's Digital Gene Expression Tag Profiling Kit according to the manufacturer's protocol.

Sequencing was done using Solexa/Illumina's Whole Genome Sequencer. RNA from each biological sample was supplied to an individual lane in one Illumina 1G flowcell. The instrument conducted 18 cycles of base incorporation, then image analysis and basecalling were performed using the Illumina Pipeline. Sorting and counting the unique tags followed, and the raw data (tag sequences and counts) are what we will analyze here. [2] went on to annotate the tags by mapping them back to the genome. In general, the mapping of tags is an important and highly non-trivial part of a DGE experiment, but we shall not deal with this task in this case study.

2.2 Reading in the data

You will find the raw data for this example in the deepSAGE directory. Please change your working directory

The tag counts for the eight individual libraries are stored in eight separate plain text files:

```
dir()
```

In each file, the tag IDs and counts for each tag are provided in a table. It is best to create a tab-delimited, plain-text 'Targets' file, which, under the headings 'files', 'group' and 'description', gives the filename, the group and a brief description for each sample.

```
targets <- read.delim("Targets.txt", stringsAsFactors = FALSE)
targets
```

This object makes a convenient argument to the function `readDGE` which reads the tables of counts into our R session, calculates the sizes of the count libraries and produces a `DGEList` object for use by subsequent functions. The `skip` and `comment.char` arguments are used to skip over comment lines:

```
library(edgeR)
d <- readDGE(targets,skip=5,comment.char="!")
colnames(d) <- c("DCLK1","WT1","DCLK2","WT2","DCLK3","WT3","DCLK4","WT4")
d$samples
dim(d)
```

2.3 Filtering

For this dataset there were over 800,000 unique tags sequenced, most of which have a very small number of counts in total across all libraries. We want to keep tags that are expressed in at least one of wild-type or transgenic mice. In either case, the tag should be expressed in at least four libraries. We seek tags that achieve one count per million for at least four libraries:

```
keep <- rowSums(cpm(d) > 1) >= 4
d <- d[keep,]
dim(d)
```

Having filtered, reset the library sizes:

```
d$samples$lib.size <- colSums(d$counts)
```

2.4 Normalization

For this SAGE data, composition normalization is not so strongly required as for RNA- Seq data. Nevertheless, we align the upper-quartiles of the counts-per-million between the libraries:

```
d <- calcNormFactors(d,method="upperquartile")
d$samples
```

2.5 Data exploration

Before proceeding with the computations for differential expression, it is possible to produce a plot showing the sample relations based on multidimensional scaling:

```
plotMDS(d, method="bcv")
```

2.6 Estimating the dispersion

First we estimate the common dispersion to get an idea of the overall degree of inter-library variability in the data:

```
d <- estimateCommonDisp(d, verbose=TRUE)
```

The biological coefficient of variation is the square root of the common dispersion. Generally it is important to allow tag-specific dispersion estimates, so we go on to compute empirical Bayes moderated tagwise dispersion estimates. The trend is turned off as it is not usually required for SAGE data:

```
d <- estimateTagwiseDisp(d, trend="none")
```

The following plot displays the estimates:

```
plotBCV(d)
```

2.7 Differential Expression

Conduct exact conditional tests for differential expression between the mutant and the wild-type:

```
et <- exactTest(d, pair=c("WT","DCLK"))
```

Top ten differentially expressed tags:

```
topTags(et)
```

The following table shows the individual counts per million for the top ten tags. *edgeR* chooses tags that both have large fold changes and are consistent between replicates:

```
detags <- rownames(topTags(et)$table)
cpm(d)[detags, order(d$samples$group)]
```

The total number of differentiallly expressed genes at FDR < 0.05:

```
summary(de <- decideTestsDGE(et, p=0.05))
```

A smearplot displays the log-fold changes with the DE genes highlighted:

```
detags <- rownames(d)[as.logical(de)]
plotSmear(et, de.tags=detags)
abline(h = c(-2, 2), col = "blue")
```

Blue lines indicate 4-fold changes.

3 Androgen-treated prostate cancer cells (RNA-Seq, two groups)

You will find the files for this example in the ProstateCancer directory

3.1 Introduction

This case study considers RNA-Seq data from a treatment vs control experiment with relatively low biological variability.

3.2 RNA Samples

Genes stimulated by androgens (male hormones) are implicated in the survival of prostate cancer cells and are potential target of anti-cancer treatments. Three replicate RNA samples were collected from prostate cancer cells (LNCaP cell line) after treatment with an androgen hormone (100uM of DHT). Four replicate control samples were also collected from cells treated with an inactive compound [3].

3.3 Sequencing

35bp reads were sequenced on an Illumina 1G Genome Analyzer using seven lanes of one flow-cell. FASTA format files are available from `http://yeolab.ucsd.edu/yeolab/Papers.html`

3.4 Read mapping

Reads were mapped and summarized at the gene level as previously described by [4]. Reads were mapped to the NCBI36 build of the human genome using Bowtie, allowing up to two mismatches. Reads not mapping uniquely were discarded. The number of reads overlapping the genomic span of each Ensembl gene (version 53) was counted. Reads mapping to introns and non-coding regions were included. The tab-delimited file of read counts can be downloaded as pnas expression.txt from `http://sites.google.com/site/davismcc/useful-documents`, or found in the **ProstateCancer** directory.

3.5 Reading the data

Read the targets file associating treatments with samples:

```
targets <- readTargets("Targets.txt")
targets
```

Read the file of counts:

```
x <- read.delim("pnas_expression.txt", row.names=1, stringsAsFactors=FALSE)
head(x)
```

Put the counts and other information into a *DGEList* object:

```
y <- DGEList(counts=x[,1:7], group=targets$Treatment, genes=data.frame(Length=x[,8]))
colnames(y) <- targets$Label
dim(y)
```

3.6 Filtering

We filter out very lowly expressed tags, keeping genes that are expressed at a reasonable level in at least one treatment condition. Since the smallest group size is three, we keep genes that achieve at least one count per million (cpm) in at least three samples:

```
keep <- rowSums(cpm(y)>1) >= 3
y <- y[keep,]
dim(y)
```

Re-compute the library sizes:

```
y$samples$lib.size <- colSums(y$counts)
```

3.7 Normalizing

Compute effective library sizes using TMM normalization:

```
y <- calcNormFactors(y)
y$samples
```

3.8 Data exploration

An MDS plot shows distances, in terms of biological coefficient of variation (BCV), between samples:

```
plotMDS(y)
```

3.9 Estimating the dispersion

The common dispersion estimates the overall BCV of the dataset, averaged over all genes:

```
y <- estimateCommonDisp(y, verbose=TRUE)
```

The BCV (square root of the common dispersion) here is 14%, a typical size for a laboratory experiment with a cell line or a model organism.

Now estimate gene-specific dispersions:

```
y <- estimateTagwiseDisp(y)
```

Plot the estimated dispersions:

```
plotBCV(y)
```

3.10 Differential expression

Compute exact genewise tests for differential expression between androgen and control treatments:

```
et <- exactTest(y)
top <- topTags(et)
top
```

Check the individual cpm values for the top genes:

```
cpm(y)[rownames(top), ]
```

The total number of DE genes at 5% FDR is given by

```
summary(de <- decideTestsDGE(et))
```

Of the 4373 tags identified as DE, 2094 are up-regulated in DHT-treated cells and 2340 are down-regulated.

Plot the log-fold-changes, highlighting the DE genes:

```
detags <- rownames(y)[as.logical(de)]
plotSmear(et, de.tags=detags)
abline(h=c(-1, 1), col="blue")
```

The blue lines indicate 2-fold changes.

4 RNA-Seq of oral carcinomas vs matched normal tissue

The data for this example can be found in the OralCarcinoma directory

4.1 Introduction

This section provides a detailed analysis of data from a paired design RNA-seq experiment, featuring oral squamous cell carcinomas and matched normal tissue from three patients [5]. The aim of the analysis is to detect genes differentially expressed between tumor and normal tissue, adjusting for any differences between the patients. This provides an example of the GLM capabilities of *edgeR*.

RNA was sequenced on an Applied Biosystems SOLiD System 3.0 and reads mapped to the UCSC hg18 reference genome [5]. Read counts, summarised at the level of refSeq transcripts, are available in Table S1 of [5].

4.2 Reading in the data

The read counts for the six individual libraries are stored in one tab-delimited file. To make this file, we downloaded Table S1 from [5], deleted some unnecessary columns and edited the column headings slightly:

```
rawdata <- read.delim("TableS1.txt", check.names=FALSE, stringsAsFactors=FALSE)
head(rawdata)
```

For easy manipulation, we put the data into a *DGEList* object:

```
library(edgeR)
y <- DGEList(counts=rawdata[,4:9], genes=rawdata[,1:3])
```

4.3 Annotation

The study [5] was undertaken a few years ago, so not all of the RefSeq IDs provided by match RefSeq IDs currently in use. We retain only those transcripts with IDs in the current NCBI annotation, which is provided by the *org.Hs.eg.db* package:

```
library(org.Hs.eg.db)
idfound <- y$genes$RefSeqID %in% mappedRkeys(org.Hs.egREFSEQ)
y <- y[idfound,]
dim(y)
```

We add Entrez Gene IDs to the annotation:

```
egREFSEQ <- toTable(org.Hs.egREFSEQ)
head(egREFSEQ)
m <- match(y$genes$RefSeqID, egREFSEQ$accession)
y$genes$EntrezGene <- egREFSEQ$gene_id[m]
```

Now use the Entrez Gene IDs to update the gene symbols:

```
egSYMBOL <- toTable(org.Hs.egSYMBOL)
head(egSYMBOL)
m <- match(y$genes$EntrezGene, egSYMBOL$gene_id)
y$genes$Symbol <- egSYMBOL$symbol[m]
head(y$genes)
```

4.4 Filtering

Different RefSeq transcripts for the same gene symbol count predominantly the same reads. So we keep one transcript for each gene symbol. We choose the transcript with highest overall count:

```
o <- order(rowSums(y$counts))
y <- y[o,]
d <- duplicated(y$genes$Symbol)
y <- y[!d,]
nrow(y)
```

Normally we would also filter lowly expressed genes. For this data, all transcripts already have at least 50 reads for all samples of at least one of the tissues types.

Recompute the library sizes:

```
y$samples$lib.size <- colSums(y$counts)
```

Use Entrez Gene IDs as row names:

```
rownames(y$counts) <- rownames(y$genes) <- y$genes$EntrezGene
y$genes$EntrezGene <- NULL
```

4.5 Normalisation

TMM normalization is applied to this dataset to account for compositional difference between the libraries.

```
y <- calcNormFactors(y)
y$samples
```

4.6 Data exploration

The first step of an analysis should be to examine the samples for outliers and for other relationships. The function `plotMDS` produces a plot in which distances between samples correspond to leading biological coefficient of variation (BCV) between those samples:

```
plotMDS(y)
```

4.7 The design matrix

Before we fit negative binomial GLMs, we need to define our design matrix based on the experimental design. Here we want to test for differential expression between tumour and normal tissues within patients, i.e. adjusting for differences between patients. In statistical terms, this is an additive linear model with patient as the blocking factor:

```
Patient <- factor(c(8,8,33,33,51,51))
Tissue <- factor(c("N","T","N","T","N","T"))
data.frame(Sample=colnames(y),Patient,Tissue)
```

```
design <- model.matrix(~Patient+Tissue)
rownames(design) <- colnames(y)
```

4.8 Estimating the dispersion

First we estimate the overall dispersion for the dataset, to get an idea of the overall level of biological variability:

```
y <- estimateGLMCommonDisp(y, design, verbose=TRUE)
```

The square root of the common dispersion gives the coefficient of variation of biological variation. Here the common dispersion is found to be 0.162, so the coefficient of biological variation is around 0.402.

Then we estimate gene-wise dispersion estimates, allowing a possible trend with averge count size:

```
y <- estimateGLMTrendedDisp(y, design)
y <- estimateGLMTagwiseDisp(y, design)
```

4.9 Differential expression

Now proceed to determine differentially expressed genes. Fit genewise glms:

```
fit <- glmFit(y, design)
```

Conduct likelihood ratio tests for tumour vs normal tissue differences and show the top genes:

```
lrt <- glmLRT(fit)
topTags(lrt)
```

Note that `glmLFT` has conducted a test for the last coefficient in the linear model, which we can see is the tumor vs normal tissue effect:

```
colnames(design)
```

The genewise tests are for tumor vs normal differential expression, adjusting for baseline differences between the three patients. (The tests can be viewed as analogous to paired t-tests.) The top DE tags have tiny p-values and FDR values, as well as large fold changes.

Here's a closer look at the counts-per-million in individual samples for the top genes:

```
o <- order(lrt$table$PValue)
cpm(y)[o[1:10],]
```

We see that all the top genes have consistent tumour vs normal changes for the three patients. The total number of differentially expressed genes at 5% FDR is given by:

```
summary(de <- decideTestsDGE(lrt))
```

Plot log-fold change against log-counts per million, with DE genes highlighted:

```
detags <- rownames(y)[as.logical(de)]
plotSmear(lrt, de.tags=detags)
abline(h=c(-1, 1), col="blue")
```

5 RNA-Seq of pathogen inoculated Arabidopsis with batch effects

5.1 Introduction

This case study re-analyses Arabidopsis thaliana RNA-Seq data described by [6]. Summarized count data is available as a data object in the CRAN package *NBPSeq* comparing hrcC challenged and mock-inoculated samples [6]. Samples were collected in three batches, and adjustment for batch effects proves to be important. The aim of the analysis therefore is to detect genes differentially expressed in response to △hrcC challenge, while correcting for any differences between the batches.

5.2 RNA samples

Pseudomonas syringae is a bacterium often used to study plant reactions to pathogens. In this experiment, six-week old Arabidopsis plants were inoculated with the △hrcC mutant of *P. syringae*, after which total RNA was extracted from leaves. Control plants were inoculated with a mock pathogen. Three biological replicates of the experiment were conducted at separate times and using independently grown plants and bacteria.

5.3 Sequencing

The six RNA samples were sequenced one per lane on an Illumina Genome Analyzer. Reads were aligned and summarized per gene using GENE-counter. The reference genome was derived from the TAIR9 genome release (`www.arabidopsis.org`).

5.4 Filtering and normalization

Load the data from the *NBPSeq* package:

```
library(NBPSeq)
library(edgeR)
data(arab)
```

```
head(arab)
```

There are two experimental factors, treatment (hrcc vs mock) and the time that each replicate was conducted:

```
Treat <- factor(substring(colnames(arab),1,4))
Treat <- relevel(Treat, ref="mock")
Time <- factor(substring(colnames(arab),5,5))
```

There is no purpose in analysing genes that are not expressed in either experimental condition. We consider a gene to be expressed at a reasonable level in a sample if it has at least two counts for each million mapped reads in that sample. This cutoff is ad hoc, but serves to require at least 4 to 6 reads in this case. Since this experiment has three replicates for each condition, a gene should be expressed in at least three samples if it responds to at least one condition. Hence we keep genes with at least two counts per million (CPM) in at least three samples:

```
keep <- rowSums(cpm(arab)>2) >= 3
arab <- arab[keep, ]
table(keep)
```

Note that the filtering does not use knowledge of what treatment corresponds to each sample, so the filtering does not bias the subsequent differential expression analysis.

Create a DGEList and apply TMM normalization:

```
y <- DGEList(counts=arab,group=Treat)
y <- calcNormFactors(y)
y$samples
```

5.5 Data exploration

An MDS plot shows the relative similarities of the six samples.

```
plotMDS(y)
```

Distances on an MDS plot of a DGEList object correspond to *leading log-fold-change* between each pair of samples. Leading log-fold-change is the root-mean-square average of the largest log2-fold-changes between each pair of samples. Each pair of samples extracted at each time tend to cluster together, suggesting a batch effect. The hrcc treated samples tend to be above the mock samples for each time, suggesting a treatment effect within each time. The two samples at time 1 are less consistent than at times 2 and 3.

To examine further consistency of the three replicates, we compute predictive log2-fold- changes (logFC) for the treatment separately for the three times.

```
design <- model.matrix(~Time+Time:Treat)
logFC <- predFC(y,design,prior.count=1,dispersion=0.05)
```

The logFC at the three times are positively correlated with one another, as we would hope:

```
cor(logFC[,4:6])
```

The correlation is highest between times 2 and 3.

5.6 The design matrix

Before we fit GLMs, we need to define our design matrix based on the experimental design. We want to test for differential expressions between △hrcC challenged and mock-inoculated samples within batches, i.e. adjusting for differences between batches. In statistical terms, this is an additive linear model. So the design matrix is created as:

```
design <- model.matrix(~Time+Treat)
rownames(design) <- colnames(y)
design
```

5.7 Estimating the dispersion

Estimate the average dispersion over all genes:

```
y <- estimateGLMCommonDisp(y, design, verbose=TRUE)
```

The square root of dispersion is the coefficient of biological variation (BCV). The common BCV is on the high side, considering that this is a designed experiment using genetically identical plants.

Now estimate genewise dispersion estimates, allowing for a possible abundance trend:

```
y <- estimateGLMTrendedDisp(y, design)
y <- estimateGLMTagwiseDisp(y, design)
```

The genewise dispersions show a decreasing trend with expression level. At low logCPM, the dispersions are very large indeed:

```
plotBCV(y)
```

5.8 Differential Expression

Now proceed to determine differentially expressed genes. Fit genewise glms:

```
fit <- glmFit(y, design)
```

First we check whether there was a genuine need to adjust for the experimental times. We do this by testing for differential expression between the three times. There is considerable differential expression, justifying our decision to adjust for the batch effect:

```
lrt <- glmLRT(fit, coef=2:3)
topTags(lrt)
FDR <- p.adjust(lrt$table$PValue, method="BH")
sum(FDR < 0.05)
```

Now conduct likelihood ratio tests for the pathogen effect and show the top genes. By default, the test is for the last coefficient in the design matrix, which in this case is the treatment effect:

```
lrt <- glmLRT(fit)
topTags(lrt)
```

Here's a closer look at the individual counts-per-million for the top genes. The top genes are very consistent across the three replicates:

```
top <- rownames(topTags(lrt))
cpm(y)[top,]
```

The total number of genes significantly up-regulated or down-regulated at 5% FDR is summarized as follows:

```
summary(dt <- decideTestsDGE(lrt))
```

We can pick out which genes are DE:

```
isDE <- as.logical(dt)
DEnames <- rownames(y)[isDE]
```

Then we can plot all the logFCs against average count size, highlighting the DE genes:

```
plotSmear(lrt, de.tags=DEnames)
abline(h=c(-1,1), col="blue")
```

The blue lines indicate 2-fold up or down.

6 Profiles of Unrelated Nigerian Individuals

6.1 Background

RNA-Seq profiles were made from lymphoblastoid cell lines generated as part of the Interna- tional HapMap project from 69 unrelated Nigerian individuals [7]. RNA from each individual was sequenced on at least two lanes of the Illumina Genome Analyser 2 platform, and mapped reads to the human genome using MAQ v0.6.8.

The study group here is essentially an opportunity sample and the individuals are likely to be genetically diverse. In this analysis we look at genes that are differentially expressed between males and females.

6.2 Loading the data

Read counts summarized by Ensembl gene identifiers are available in the *tweeDEseqCountData* package:

```
library(tweeDEseqCountData)
data(pickrell1)
Counts <- exprs(pickrell1.eset)
dim(Counts)
Counts[1:5,1:5]
```

In this analysis we will compare female with male individuals.

```
Gender <- pickrell1.eset$gender
table(Gender)
rm(pickrell1.eset)
```

Annotation for each Ensemble gene is also available from the *tweeDEseqCountData* package:

```
data(annotEnsembl63)
annot <- annotEnsembl63[,c("Symbol","Chr")]
annot[1:5,]
rm(annotEnsembl63)
```

Form a DGEList object combining the counts and associated annotation:

```
library(edgeR)
y <- DGEList(counts=Counts, genes=annot[rownames(Counts),])
```

6.3 Filtering

Keep genes with least 1 count-per-million reads (cpm) in at least 20 samples:

```
isexpr <- rowSums(cpm(y)>1) >= 20
```

Keep only genes with defined annotation:

```
hasannot <- rowSums(is.na(y$genes))==0
y <- y[isexpr & hasannot,]
dim(y)
y$samples$lib.size <- colSums(y$counts)
```

6.4 Normalization

Apply scale normalization:

```
y <- calcNormFactors(y)
```

6.5 Data exploration

The library sizes vary from about 5 million to over 15 million:

```
barplot(y$samples$lib.size*1e-6, names=1:69, ylab="Library size (millions)")
```

6.6 Data exploration

First we estimate the biological coefficient of variation:

```
design <- model.matrix(~Gender)
y <- estimateDisp(y,design,robust=TRUE)
plotBCV(y)
```

6.7 Quasi-likelihood linear modeling

Then the glmQLFTest function estimates the quasi-likelihood (QL) or technical dispersion around the BCV trend. The large number of cases and the high variability means that the QL dispersions are not squeezed very heavily from the raw values:

```
fit <- glmQLFTest(y,design,robust=TRUE,plot=TRUE)
```

Now find genes differentially expression between male and females. Positive log-fold- changes mean higher in males. The highly ranked genes are mostly on the X or Y chromosomes. Top ranked is the famous *XIST* gene, which is known to be expressed only in females.

```
options(digits=3)
topTags(fit,n=15)
summary(decideTestsDGE(fit))
```

6.8 Gene set testing

The *tweeDEseqCountData* package includes a list of genes belonging to the male-specific region of chromosome Y, and a list of genes located in the X chromosome that have been reported to escape X-inactivation. We expect genes in the first list to be up-regulated in males, whereas genes in the second list should be up-regulated in females.

```
data(genderGenes)
Ymale <- rownames(y) %in% msYgenes
Xescape <- rownames(y) %in% XiEgenes
```

Roast gene set tests confirm that the male-specific genes are significantly up as a group in our comparison of males with females, whereas the X genes are significantly down as a group. The p-values are at their minimum possible values given the number of rotations:

```
index <- list(Y=Ymale,X=Xescape)
mroast(y,index,design,nrot=9999)
```

The results from competitive camera gene sets tests are even more convincing. The positive intergene correlations here show that the genes in each set tend to be biologically correlated:

```
camera(y,index,design)
```

See where the X and Y genes fall on the MA plot:

```
with(fit$table, plot(logCPM,logFC,pch=16,cex=0.2))
with(fit$table, points(logCPM[Xescape],logFC[Xescape],pch=16,col="red"))
with(fit$table, points(logCPM[Ymale],logFC[Ymale],pch=16,col="blue"))
legend("bottomleft",legend=c("Ymale genes","Xescape genes"),pch=16,,col=c("blue","red"))
```

References

[1] Davis J McCarthy, Yunshun Chen, and Gordon K Smyth. "Differential expression analysis of multifactor RNA-Seq experiments with respect to biological variation". In: *Nucleic Acids Research* 40 (Jan. 2012), pp. 4288–4297. ISSN: 1362-4962. DOI: 10.1093/nar/gks042. URL: http://www.ncbi.nlm.nih.gov/pubmed/22287627.

[2] P. A. C. 't Hoen et al. "Deep sequencing-based expression analysis shows major advances in robustness, resolution and inter-lab portability over five microarray platforms". In: *Nucleic Acids Research* 36 (2008), e141.

[3] H. R. Li et al. "Determination of tag density required for digital transcriptome analysis: Application to an androgen-sensitive prostate cancer model". In: *Proceedings of the National Academy of Sciences of the USA* 105 (2008), pp. 20179–20184.

[4] M. D. Young et al. "Gene ontology analysis for RNA-seq: accounting for selection bias". In: *Genome Biology* 11 (2010), R14.

[5] B. B. Tuch et al. "Tumor transcriptome sequencing reveals allelic expression imbalances associated with copy number alterations". In: *PLoS ONE* 5 (2010), e9317.

[6] J. S. Cumbie et al. "Gene-counter: A computational pipeline for the analysis of RNA-Seq data for gene expression differences". In: *PLoS ONE* 6 (2011), e25279.

[7] J. K. Pickrell et al. "Understanding mechanisms underlying human gene expression variation with RNA sequencing". In: *Nature* 464 (2010), pp. 768–772.