Differential Expression Analysis using *edgeR*

Oscar Rueda & Bernard Pereira

July 25, 2015

Contents

1 Introduction

In this tutorial, we will be using *edgeR*[1] to analyse some RNA-seq data taken from.

You can find out more about *edgeR* from:

- EdgeR paper
- Bioconductor website

There are, of course, other Bioconductor tools available to analyse RNA-seq data, and these will differ in their details and in the way the carry out some tasks. However, the current 'best practice' workflow follows the principles outlined here.

1.1 The Data

We will be using data downloaded from GEO (GSE29968)[2]. Here is the authors' summary of the data from the GEO website (`http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE29968`):

> We performed the integrative transcriptome analysis of human esophageal squamous cell carcinoma (ESCC) using Illumina high-throughput sequencing. A total of 187 million 38bp sequencing reads were generated containing 7 billion bases for three pairs of matched patient-derived ESCC clinical specimens and their adjacent non-tumorous tissues. By investigating the digital gene expression profiling, we found 1425 genes significantly differentially expressed and detected more than 9000 SNPs across all six samples. We also identified protein tyrosine kinase 6 (PTK6) as a novel tumor suppressor gene, which is critical in ESCC development.

In this practical, we will use *edgeR* to find genes that are differentially expressed between the tumours and normal tissue.

2 DE Workflow

2.1 Reading in the Data

We first need to load the required library and data required for this practical. You may use the file previously generated, or the set of read counts in Day3/Counts.RData.

Note that the genes in this file are identified by their Entrez gene ids.

```
library(edgeR)
load("Day3/Counts.RData")
Counts <- tmp$counts
colnames(Counts) <- c("16N", "16T", "18N", "18T", "19N", "19T")

dim(Counts)
head(Counts)
```

2.2 Creating a DGEList object

We will now create a DGEList object to hold our read counts. This object is a container for the counts themsleves, and also for all the associated metadata - these include, for example, sample names, gene names and normalisation factors once these are computed. The DGEList is an example of the custom task-specific structures that are frequently used in Bioconductor to make analyses easier.

```
dgList <- DGEList(counts=Counts, genes=rownames(Counts))
```

```
dgList
dgList$samples
head(dgList$counts)     #Many rows!
head(dgList$genes)      #Likewise!
```

2.3 Filtering

There are approximately 26000 genes in this dataset. However, many of them will not be expressed, or will not be represented by enough reads to contribute to the analysis. Removing these genes means that we have ultimately have fewer tests to perform, thereby reducing the problems associated with multiple testing. Here, we retain only those genes that are represented at least 1cpm reads in at least two samples (cpm=counts per million).

```
countsPerMillion <- cpm(dgList)
summary(countsPerMillion)
#'summary' is a useful function for exploring numeric data; eg. summary(1:100)
```

```
countCheck <- countsPerMillion > 1
head(countCheck)

keep <- which(rowSums(countCheck) >= 2)
dgList <- dgList[keep,]
summary(cpm(dgList))    #compare this to the original summary
```

2.4 Normalisation

As we have discussed, it is important to normalise RNA-seq both within and between samples. *edgeR* implements the trimmed mean of M-values (TMM) method.

```
?calcNormFactors
dgList <- calcNormFactors(dgList, method="TMM")
```

2.5 Data Exploration

We can examine inter-sample relationships by producing a plot based on mutlidimensional scaling.

```
plotMDS(dgList)
```

2.6 Setting up the Model

We are now ready to set up the model! We first need to specify our design matrix, which describes the setup of the experiment.

```
sampleType<- rep("N", ncol(dgList))      #N=normal; T=tumour
sampleType[grep("T", colnames(dgList))] <- "T"
#'grep' is a string matching function.

sampleReplicate <- paste("S", rep(1:3, each=2), sep="")
```

```
designMat <- model.matrix(~sampleReplicate + sampleType)
designMat
```

2.7 Estimating Dispersions

As disucssed, we need to estimate the dispersion parameter for our negative binomial model. As there are only a few samples, it is difficult to estimate the dispersion accurately for each gene, and so we need a way of 'sharing' information between genes. Possible solutions include:

- Using a common estimate across all genes.
- Fitting an estimate based on the mean-variance trend across the dataset, such that genes similar abundances have similar variance estimates (trended dispersion).
- Computing a genewise dispersion (tagwise dispersion)

In *edgeR*, we use an empirical Bayes method to 'shrink' the genewise dispersion estimates towards the common dispersion (tagwise dispersion).

Note that either the common or trended dispersion needs to be estimated before we can estimate the tagwise dispersion.

```
dgList <- estimateGLMCommonDisp(dgList, design=designMat)
dgList <- estimateGLMTrendedDisp(dgList, design=designMat)
dgList <- estimateGLMTagwiseDisp(dgList, design=designMat)
```

We can plot the estimates and see how they differ. The biological coefficient of variation (BCV) is the square root of the dispersion parameter in the negative binomial model.

```
plotBCV(dgList)
```

2.8 Differential Expression

We can now find our differentially expressed genes. After fitting the model, we can use the topTags() function to explore the results, and set theresholds to identify subsets of differentially expressed genes.

```
fit <- glmFit(dgList, designMat)
lrt <- glmLRT(fit, coef=4)
```

```
edgeR_result <- topTags(lrt)
?topTags

save(topTags(lrt,n=15000)$table, file='Day3/edgeR_Result.RData')        #We will need this later
```

Finally, we can plot the log-fold changes of all the genes, and the highlight those that are differentially expressed.

```
?decideTests
deGenes <- decideTestsDGE(lrt, p=0.001)
deGenes <- rownames(lrt)[as.logical(deGenes)]
plotSmear(lrt, de.tags=deGenes)
abline(h=c(-1, 1), col=2)
```

References

[1] Robinson, MD.et al. (2010) *edgeR: a Bioconductor package for differential expression analysis of digital gene expression data*, Bioinformatics, 26 (1) 139-140.

[2] Ma, S. et al. (2012) *Identification of PTK6, via RNA Sequencing Analysis, as a Suppressor of Esophageal Squamous Cell Carcinoma*, Gastroenterology, 143 (3) 675-686.