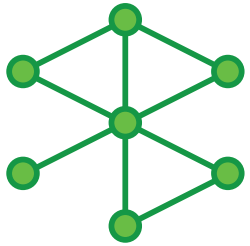
UNSW
AUSTRALIA

Never Stand Still

Differential Gene Expression Analysis using edgeR

Susan Corley
Systems Biology Initiative
11 November 2014



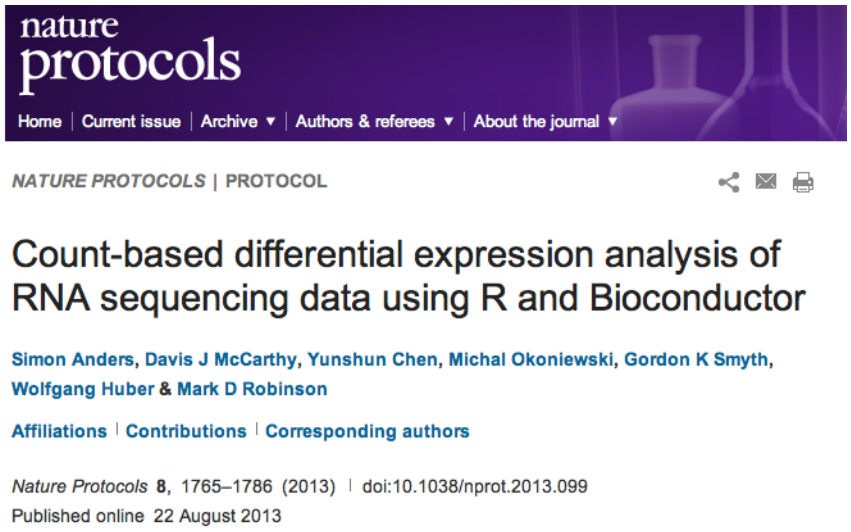
RNA-Seq analysis using edgeR

RNA-Seq steps

1. Quantify reads mapped to genes
    - Prepare table of counts
2. Normalize for different library size (total number of reads differs from sample to sample)
3. Calculate dispersion for each gene – e.g. what variation do you see in Gene A from sample to sample
4. Differential expression testing e.g. is expression of Gene A different in Treatment vs Controls, given normalised counts and the variation you see in each condition?

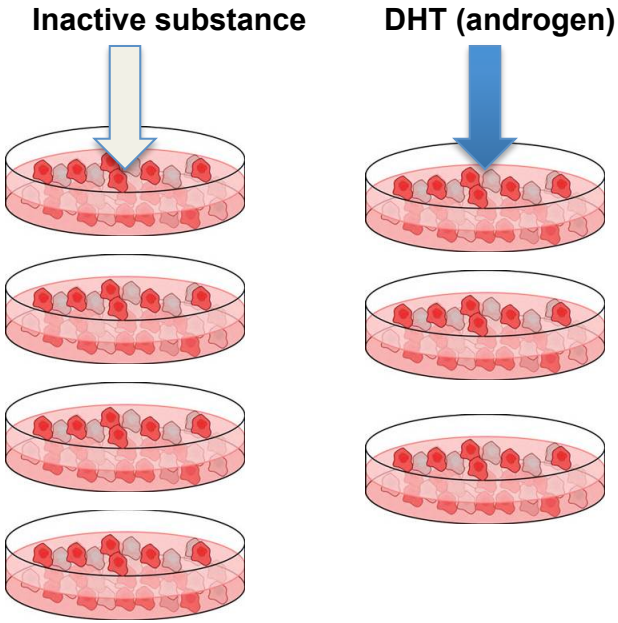
edgeR

- Prof Gordon Smyth and team at WEHI - edgeR and limma and VOOM
- edgeR a count based method (as is DESeq)
- Several papers explaining the method



Today's case study

- Case study in edgeR Users Guide (4.3)
- Experiment by Li et al. 2008
- LNCaP cells – androgen sensitive prostate cancer cell line



Today's case study

pnas_expression.txt

ensembl_ID	lane1	lane2	lane3	lane4	lane5	lane6	lane8
ENSG00000157214	5270	6208	7359	7521	9806	9786	4418
ENSG00000212875	4652	5409	6445	6668	5623	5727	2875
ENSG00000115053	4352	4612	5880	5892	4744	4948	1892
ENSG00000162669	4212	5000	5516	5812	4328	4172	1632
ENSG00000131051	4141	4926	5615	5548	8762	8479	3496
ENSG00000132570	4070	4704	5471	5548	6027	5665	2613
ENSG00000106070	4043	4513	6174	6385	1495	1507	420
ENSG00000122566	3775	4131	5427	5831	6959	7069	2505
ENSG00000008128	3467	4300	4514	5018	5800	5814	2079
ENSG00000096384	3354	3402	4147	4257	3240	3228	1108
ENSG00000142875	3166	3735	3920	4375	8123	7758	3880
ENSG00000087460	3055	3428	4292	4448	5301	5047	1913
ENSG00000169045	2903	3407	3901	4070	4114	4233	1635
ENSG00000151150	2845	3424	4021	4056	3263	3160	1270
ENSG00000081026	2808	3300	3708	3644	4252	4144	1896
ENSG00000212679	2727	3045	3369	3366	3497	3390	1709
ENSG00000099250	2700	2974	4605	4043	2640	2894	863
ENSG00000114867	2628	2862	4311	4500	5814	5715	1584
ENSG00000139220	2522	2789	3372	3484	3095	3242	1404
ENSG00000104067	2497	3117	3627	3900	4345	4519	1646
ENSG00000100201	2482	2996	3523	4118	4174	4045	1622
ENSG00000122786	2427	2763	2988	3063	1664	1626	585
ENSG00000159023	2273	2296	2667	2853	3202	3065	1194
ENSG00000101333	2256	2793	3456	3362	2702	2976	1320
ENSG00000140264	2248	2872	3594	3674	4355	5049	1472
ENSG00000154305	2246	2443	3129	3294	3701	4114	1572
ENSG00000217866	2168	2463	2883	2957	3291	3209	1682
ENSG00000170004	2162	2522	3240	3352	2948	3036	1100
ENSG00000167978	1935	2301	3155	3260	2698	2651	853
ENSG00000138326	1905	2007	2244	2286	2208	1986	975
ENSG00000196586	1902	2301	2726	2663	2753	2863	1086
ENSG00000184304	1884	2060	2504	2580	2072	2040	796
ENSG00000117523	1860	2164	2646	2754	2967	3154	1224
ENSG00000086205	1845	2077	2790	2682	1107	1072	375

Targets.txt

	Lane	Treatment	Label
Con1	1	Control	Con1
Con2	2	Control	Con2
Con3	3	Control	Con3
Con4	4	Control	Con4
DHT1	5	DHT	DHT1
DHT2	6	DHT	DHT2
DHT3	7	DHT	DHT3

edgeR analysis

We start by loading the Bioconductor libraries we will be using.

```
library(edgeR)
library(biomaRt)
library(gplots)
```

Read in count table and experimental design.

```
data <- read.delim("pnas_expression.txt", row.names=1, header=T)
targets <- read.delim("Targets.txt", header=T)
```

Create DGEList Object

Create a DGEList object.

```
y <- DGEList(counts=data[,1:7], group=targets$Treatment)
```

Change the column names of the object to align with treatment.

```
colnames(y) <- targets$Label
```

Check the dimensions of the DGEList object.

```
dim(y)
```

We see we have 37435 rows (i.e. genes) and 7 columns (samples)

Filtering out low count genes

Now we will filter out genes with low counts by only keeping those rows where the count per million (cpm) is at least 1 in at least three samples

```
keep <-rowSums( cpm(y)>1) >=3
```

```
y <- y[keep, ]
```

Check how many rows (genes) are retained now.

```
dim(y)
```

How many genes have been filtered out?

We see that we now have 16494 rows so (37435-16494) 20941 genes have been filtered out.

As we have removed the lowly expressed genes the total number of counts per sample has not changed greatly. Let us check the total number of reads per sample in the original data (data) and now after filtering.

```
colSums(data)
```

978576 1156844 1442169 1485604 1823460 1834335 681743

```
colSums(y$counts)
```

976847 1154746 1439393 1482652 1820628 1831553 680798

Normalization

We will now perform normalization to take account of different library size among samples. edgeR uses the TMM method of normalization (Robinson and Oshlack, 2010).

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

We will check the calculated normalization factors.

```
y$samples
```

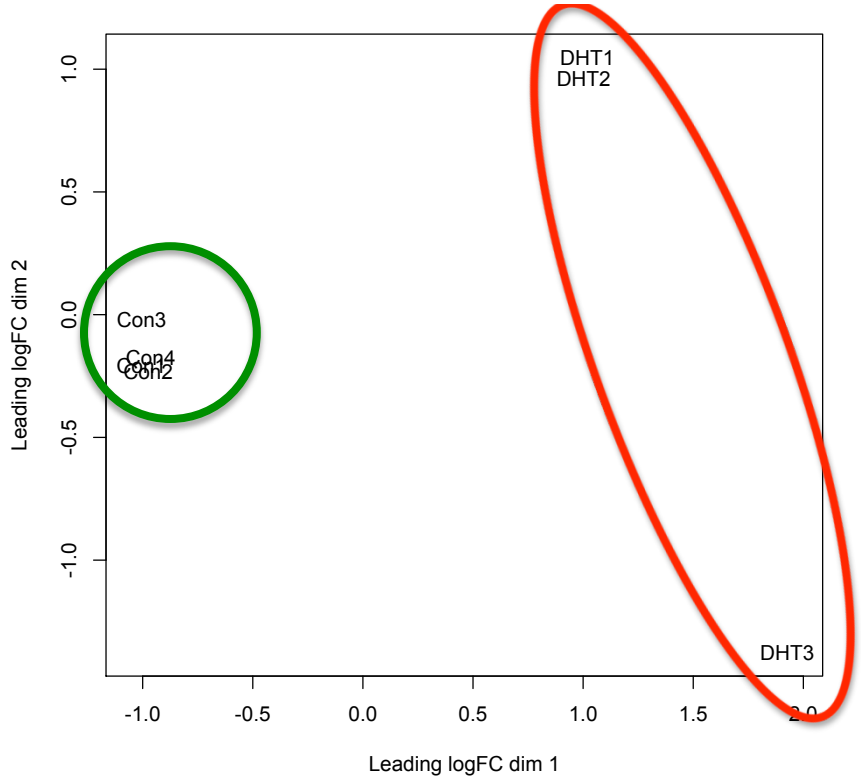
```
> y$samples
       group lib.size norm.factors
Con1 Control   976847    1.0296636
Con2 Control  1154746    1.0372521
Con3 Control  1439393    1.0362662
Con4 Control  1482652    1.0378383
DHT1     DHT  1820628    0.9537095
DHT2     DHT  1831553    0.9525624
DHT3     DHT   680798    0.9583181
```

Do the samples cluster by condition?

Let's have a look at whether the samples cluster by condition

plotMDS(y)

You should have produced this plot.



Dispersion

We will now estimate the dispersion

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

By using verbose we get the Disp and BCV values printed on the screen.

What value to you see for BCV?

The common dispersion estimates the overall Biological Coefficient of Variation (BCV) of the dataset averaged over all genes. The common dispersion is 0.02 and the BCV is the square root of the common dispersion ($\sqrt{0.02} = 0.14$). A BCV of 14% is typical for cell line experiment.

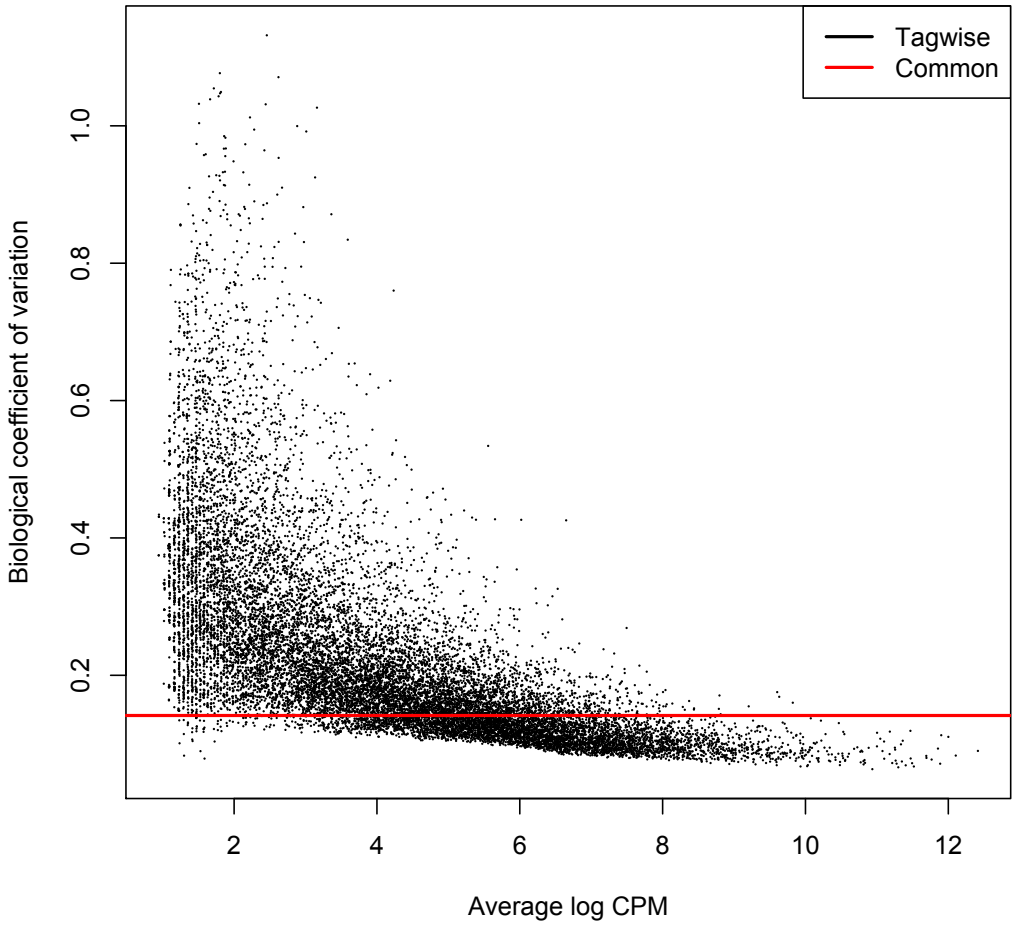
We now estimate gene-specific dispersion

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

We will plot the tagwise dispersion and the common dispersion

```
plotBCV(y)
```

Dispersion


Tagwise
Common
Biological coefficient of variation
0.2
0.4
0.6
0.8
1.0
2
4
6
8
10
12
Average log CPM

Differential expression testing

We now test for differentially expressed genes

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

Now we will use the topTags function to adjust for multiple testing. We will use the Benjimini Hochberg ("BH") method and we will produce a table of results

```
res <- topTags(et, n=nrow(y$counts), adjust.method="BH")$table
```

Let's have a look at the first rows of the table

```
head(res)
```

Add in gene symbols

The ensemble gene identifier is not as helpful as the gene symbol so let's add in a column with the gene symbol. We will use the BiomaRt package to do this.

We start by using the useMart function of BiomaRt to access the human data base of ensemble gene ids. Then we create a vector of our ensemble gene ids.

```
ensembl=useMart("ensembl", dataset="hsapiens_gene_ensembl")

ensembl_names<-rownames(res)
```

We then use the function getBM to get the gene symbol data we want

```
genemap <-getBM( attributes= c("ensembl_gene_id", "entrezgene", "hgnc_symbol"),
filters="ensembl_gene_id", values=ensembl_names, mart=ensembl)
```

This takes about a minute.

Have a look at the start of the genemap dataframe.

```
head(genemap)
```

Add in gene symbols

We use the match function to match up our data with the data we have just retrieved from the database.

```
idx <- match(ensembl_names, genemap$ensembl_gene_id )
```

```
res$entrez <- genemap$entrezgene [ idx ]
```

```
res$hgnc_symbol <- genemap$hgnc_symbol [ idx ]
```

Now lets have a look at the start of the res data frame which lists the 10 most significant differentially expressed genes

```
head(res)
```

As you see we have now added the hgnc symbol and the entrez id to our results.

We might also look at the cpm for these genes

Subsets of DE genes

We will now make subsets of the most significant upregulated and downregulated genes.

```
de <- res[res$FDR<0.05, ]

de_upreg <- res[res$FDR<0.05 & res$logFC >0,]

de_downreg <- res[res$FDR<0.05 & res$logFC <0,]
```

We can write out these results to our current directory.

```
write.csv( as.data.frame(de_all), file="DE_results.csv")

write.csv( as.data.frame(de_upreg), file="DE_upreg_results.csv")

write.csv( as.data.frame(de_downreg), file="DE_downreg_results.csv")
```

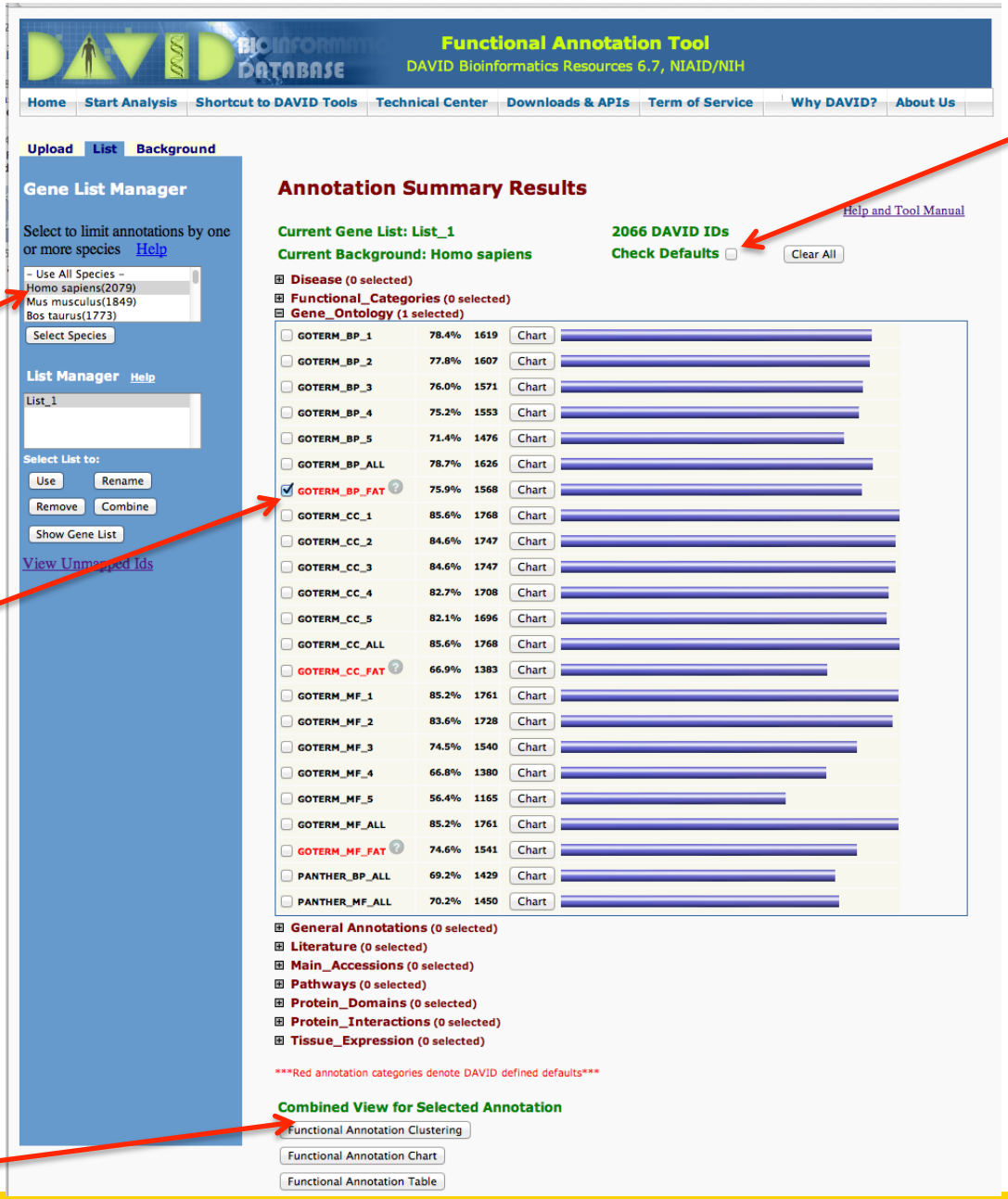
Create a list of DE gene symbols

```
de_top_3000 <-de[1:3000,]

de_top_gene_symbols <-de_top_3000$hgnc_symbol

write(de_top_gene_symbols, 'DE_gene_symbols.txt', sep="\t")
```

This txt file of gene symbols can then be uploaded into DAVID

SYSTEMS BIOLOGY INITIATIVE

UNSW AUSTRALIA



SYSTEMS BIOLOGY INITIATIVE

UNSW AUSTRALIA