edgeR: Methods for differential expression in digital gene expression datasets

Mark Robinson
mrobinson@wehi.edu.au

March 4, 2009

1 Introduction

This document gives a brief introduction and overview of the R Bioconductor package `edgeR`, which provides statistical routines for determining differential expression in digital gene expression data. The routines can be applied equally to SAGE, CAGE, Illumina/Solexa, 454 or ABI SOLiD experiments. In fact, the methods may be useful in other experiments where counts are observed.

R packages for the processing of raw data files for digital gene expression (DGE) datasets are still in development stages (e.g. `ShortRead`) at time of writing. The methods presented here require a simple `DGEList` object that contains three pieces of information:

1. `data`: a table of counts where each row represents a gene/exon (or whatever genomic feature is being tracked) and each column is a different sample.
2. `group`: a vector (with length equal to the number of columns of `data`) denoting the experimental group.
3. `lib.size` (same length as `group`): the total number of reads sequenced for each sample.

We now discuss a couple examples.

2 Moderated negative binomial dispersions

The basic model we use for DGE data is based on the negative binomial distribution. The model is very flexible. For example, if Y is distributed as $NB(\mu, \phi)$, then the expected value of Y is μ and the variance is $\mu + \mu^2 \cdot \phi$, thus giving sufficient flexibility for many scenarios in observing count data.

The observed data can be denoted as Y_{gij} where g is the gene (tag, exon, etc.), i is the experimental group and j is the index of the replicate. We can model the counts as

$$Y_{gij} \sim NB(M_j \cdot p_{gi}, \phi_g)$$

where p_{gi} represents the proportion of the sequenced sample for group i that is tag g and M_j represents the library size. It is of interest to find genes where, for example, p_{g1} is significantly different from p_{g2}. The parameter ϕ_g is the overdispersion (relative to the Poisson) and represents the biological, or sample-to-sample variability. The methods we developed moderate the dispersion estimates towards a common dispersion, much like how the `limma` package moderates the variances in the analysis of microarray data.

To illustrate the methods, we generate some count data. Here, I have sampled from a negative binomial distribution and created the list object that is necessary for the moderated dispersion functions:

```
> library(edgeR)
> set.seed(101)
> n <- 200
> lib.sizes <- c(40000, 50000, 38000, 40000)
> p <- runif(n, min = 1e-04, 0.001)
> mu <- outer(p, lib.sizes)
> mu[1:5, 3:4] <- mu[1:5, 3:4] * 8
> y <- matrix(rnbinom(4 * n, size = 4, mu = mu), nrow = n)
> rownames(y) <- paste("tag", 1:nrow(y), sep = ".")
> y[1:10, ]
```

	[,1]	[,2]	[,3]	[,4]
tag.1	15	13	117	77
tag.2	3	4	49	33
tag.3	25	56	302	332
tag.4	40	13	271	91
tag.5	13	3	51	56
tag.6	14	7	31	18
tag.7	16	39	19	9
tag.8	6	28	6	6
tag.9	10	42	80	14
tag.10	33	25	5	27

```
> d <- DGEList(data = y, group = rep(1:2, each = 2), lib.size = lib.sizes)
> d

DGEList: 200 rows, 4 libraries

> names(d)

[1] "data"     "lib.size" "group"
```

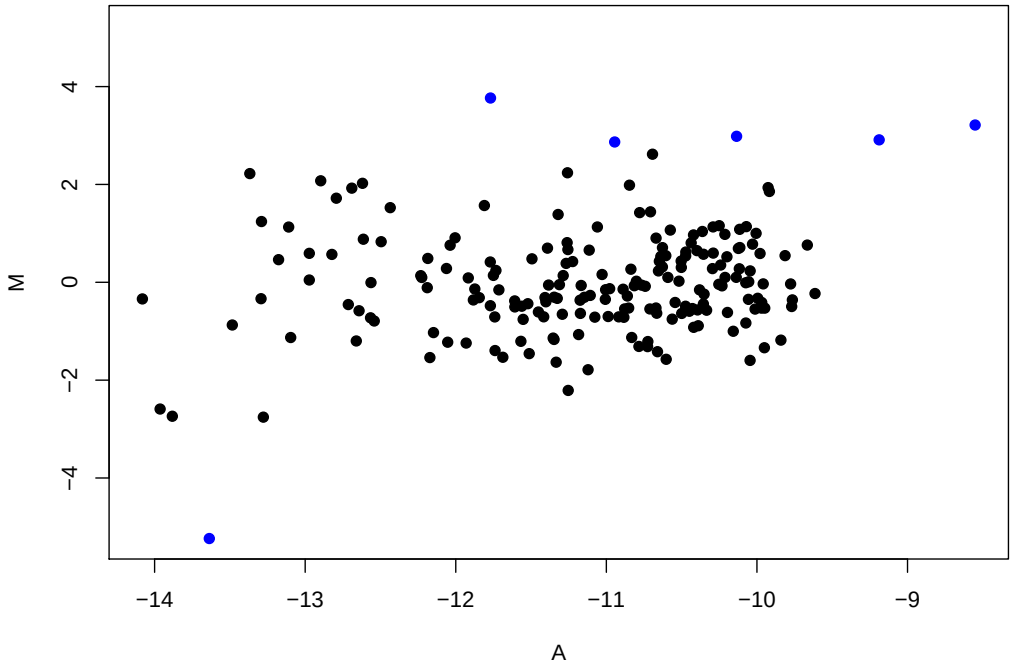
To run the moderated analysis, we first need to determine how much moderation is necessary. For this, we use an empirical Bayes rule and involves calculating a weight parameter α. Following this, the main function to do the statistical testing is called `deDGE`.

```
> alpha <- alpha.approxeb(d)

[quantileAdjust] Iteration (dot=1000) 1 :
[quantileAdjust] Iteration (dot=1000) 2 :

> alpha

EBList: alpha=4.015982

> ms <- deDGE(d, alpha = alpha$alpha)

Calculating shrinkage overdispersion parameters.
[quantileAdjust] Iteration (dot=1000) 1 :
[quantileAdjust] Iteration (dot=1000) 2 :
Calculating Fisher exact p-values (dot=1000):

> ms
```

```
deDGEList: 4 samples, adjusted to library size of 41755.95
```

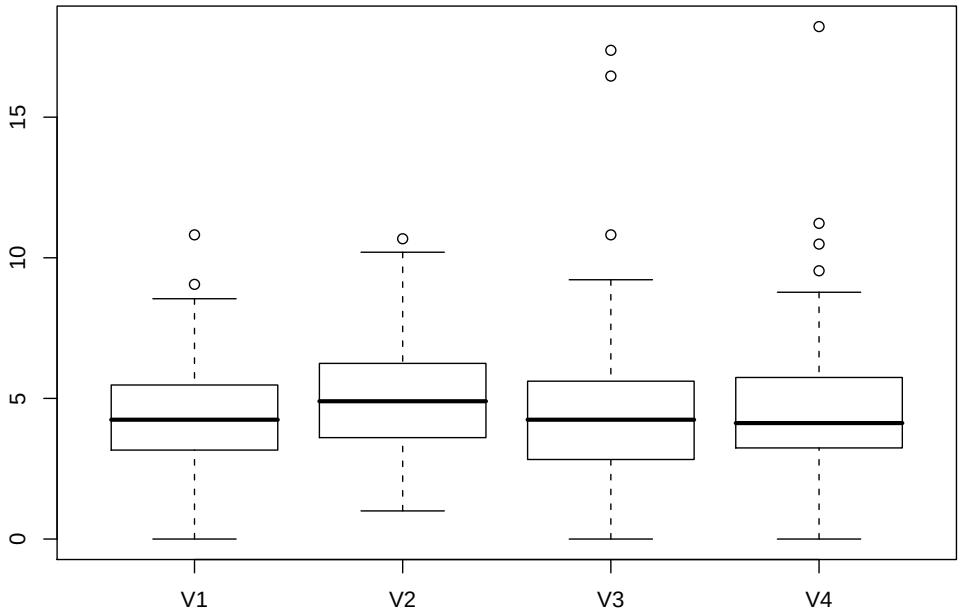
It may be informative to look at the data via MA plots, as is commonly done in the analysis of microarray experiments. In addition, we can highlight those genes that are determined to be differentially expressed (above, we made 5 tags have a much higher mean, and these are highlighted in blue). To do this, you can call the commands:

```
> adj.p <- p.adjust(ms$exact, "fdr")
> k <- (adj.p < 0.05)
> plotMA(ms, col = c("black", "blue")[k + 1])
```



Also, we may want to look at distributions of the counts to see the effect of different total numbers of reads. An easy way to do this is:

```
> boxplot(as.data.frame(sqrt(d$data)))
```



Or, you can have a look at the output, according to:

```
> topTags(ms)
```

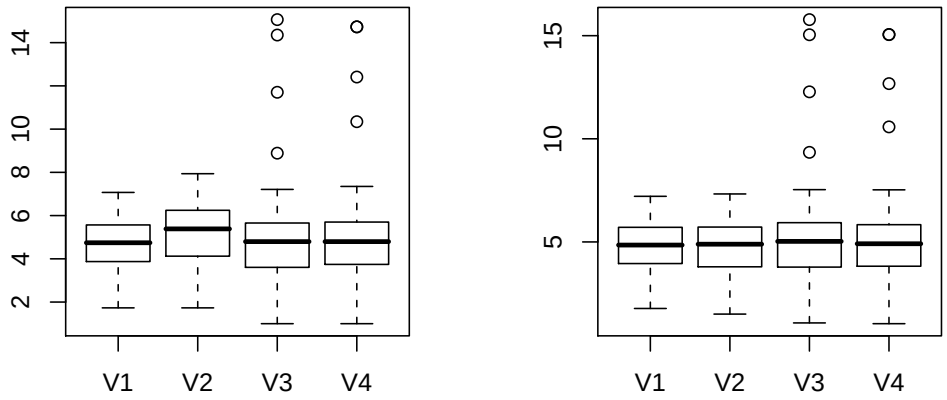
```
                A         M      P.Value   adj.P.Val
tag.184 -13.636741 -5.236964 6.442422e-05 0.008390416
tag.3    -8.551001  3.214721 8.390416e-05 0.008390416
tag.2   -11.769441  3.766485 1.369328e-04 0.009128855
tag.4    -9.188338  2.911640 3.478712e-04 0.014107045
tag.1   -10.135202  2.984303 3.526761e-04 0.014107045
tag.5   -10.944627  2.868362 1.090710e-03 0.036357002
tag.105 -10.693473  2.618183 2.448307e-03 0.069951632
tag.14  -11.258030  2.238719 1.125103e-02 0.238357234
tag.164 -11.253361 -2.209732 1.125103e-02 0.238357234
tag.123 -13.277809 -2.755947 1.191786e-02 0.238357234
```

3 Poisson example

It has been observed that in some high-throughput (or deep) sequencing approaches that not a great deal of overdispersion is observed. Specifically, the means and variances appear to be very close to each other, suggesting the Poisson distribution is a good fit. Methods within the `edgeR` package may still be useful, including the quantile adjustment (effectively a normalization) and the exact testing routines.

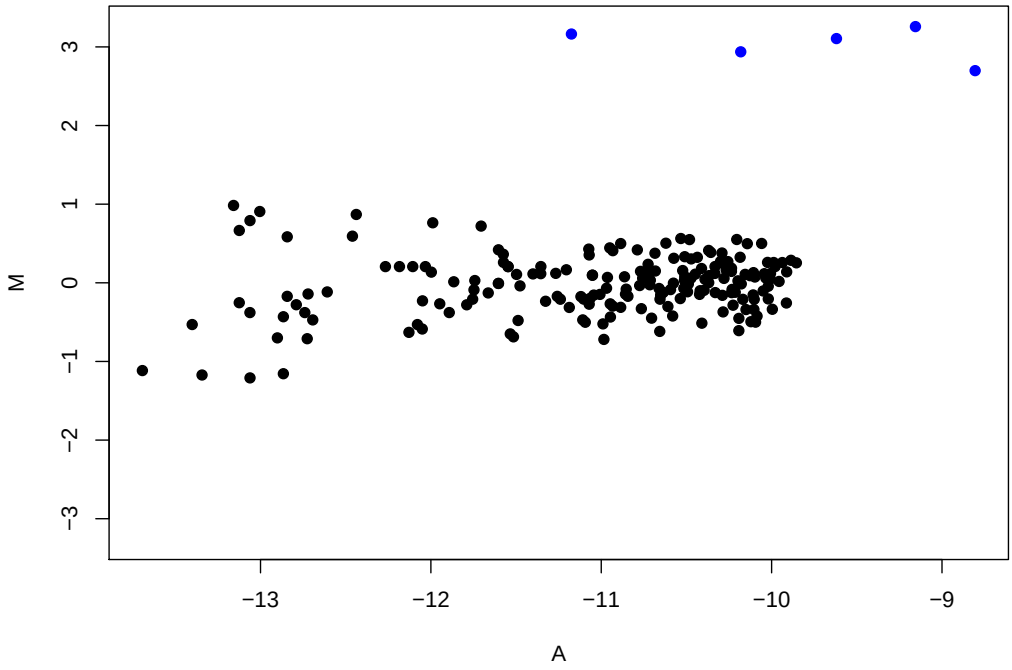
To illustrate this, we sample Poisson data and run `deDGE` with the `doPoisson` argument set to `TRUE`. The data is quantile-adjusted and the exact test is invoked after first setting the dispersion parameter to 0. To observe that it does a normalization of sorts, we can again look at boxplots.

```
> set.seed(101)
> y <- matrix(rpois(4 * n, lambda = mu), nrow = n)
> d <- DGEList(data = y, group = rep(1:2, each = 2), lib.size = lib.sizes)
> ms <- deDGE(d, doPoisson = TRUE)

Quantile adjusting as Poisson.
[quantileAdjust] Iteration (dot=1000) 1 :
Calculating Fisher exact p-values (dot=1000):

> par(mfrow = c(1, 2))
> boxplot(as.data.frame(sqrt(d$data)))
> boxplot(as.data.frame(sqrt(ms$pseudo)))
```



Again, the same functions can be used to access the results:

```
> adj.p <- p.adjust(ms$exact, "fdr")
> k <- (adj.p < 0.05)
> plotMA(ms, col = c("black", "blue")[k + 1])
> topTags(ms)

             A          M      P.Value    adj.P.Val
4    -9.155839  3.2583285 1.097201e-76 2.194402e-74
3    -8.805469  2.6980439 1.066188e-65 1.066188e-63
1    -9.619455  3.1059797 3.222678e-52 2.148452e-50
2   -11.175028  3.1639503 3.067636e-20 1.377498e-18
5   -10.181229  2.9376157 3.443745e-20 1.377498e-18
56  -10.192387 -0.6079180 1.807616e-02 6.025387e-01
147 -10.984463 -0.7196385 2.446208e-02 6.989164e-01
163 -10.204730  0.5509445 3.353081e-02 8.382702e-01
86  -10.058616  0.5005776 4.566609e-02 9.604483e-01
142 -11.515268 -0.6872191 5.365001e-02 9.604483e-01
```



4 Future improvements and extension

Here, we list some improvements that are planned for the `edgeR` package:

1. As the packages for the processing of raw high-throughput sequencing data become more mature, `edgeR` may need to adapt and operate on different objects. As shown above, `edgeR` operates on a simple object containing simple data summaries which will presumably be readily available from pre-processing steps.

2. At present, the package only does 2-sample comparisons. The methods are straightforward to extend to multiple samples, but other considerations will need to be made and further development is required.

3. Some speed improvements have been made but as the datasets become larger, some further optimizations may be necessary.

5 References

See the following manuscripts for further details:

1. Robinson MD, Smyth GK. *Moderated statistical tests for assessing differences in tag abundance.***Bioinformatics**. 2007 Nov 1;23(21):2881-7.

2. Robinson MD, Smyth GK. *Small-sample estimation of negative binomial dispersion, with applications to SAGE data.* **Biostatistics**. 2008 Apr;9(2):321-32.