edgeR: Methods for differential expression in digital gene expression datasets

Mark Robinson
mrobinson@wehi.edu.au

July 26, 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 use the Zhang et al. (1997) SAGE dataset. First, we read in the raw data:

```
> library(edgeR)
> WD <- getwd()
```

```
> dataPath <- paste(.find.package("edgeR"), "data", sep = "/")
> fn <- dir(dataPath, "txt")
> fn
```

```
[1] "NC1.txt"   "NC2.txt"   "Tu102.txt" "Tu98.txt"
```

```
> setwd(dataPath)
> head(read.table(fn[1], header = TRUE))
```

```
  Tag_Sequence Count
1   AAAAAAAAAA    17
2   AAAAAAAAGA     1
3   AAAAAAACCC     1
4   AAAAAAAGCA     1
5   AAAAAAATCA     4
6   AAAAAAATTT     1
```

```
> dataList <- readDGE(fn, header = TRUE)
> setwd(WD)
```

`dataList` is a `list` object with elements for the table of counts and the library sizes. We need to add to this a 'group' variable which specifies the experimental condition and then we create a `DGEList` object that is the container for digital gene expression data. Note that your raw data may come from a different source (e.g. another R object), but the `DGEList` needs the 3 pieces of information mentioned above.

```
> cls <- gsub("[0-9]", "", colnames(dataList$data))
> cls
```

```
[1] "NC" "NC" "Tu" "Tu"
```

```
> keep <- rowSums(dataList$data) > 8
> sum(keep)
```

```
[1] 2457
```

```
> d <- DGEList(data = dataList$data[keep, ], group = cls, lib.size = dataList$lib.size)
> d
```

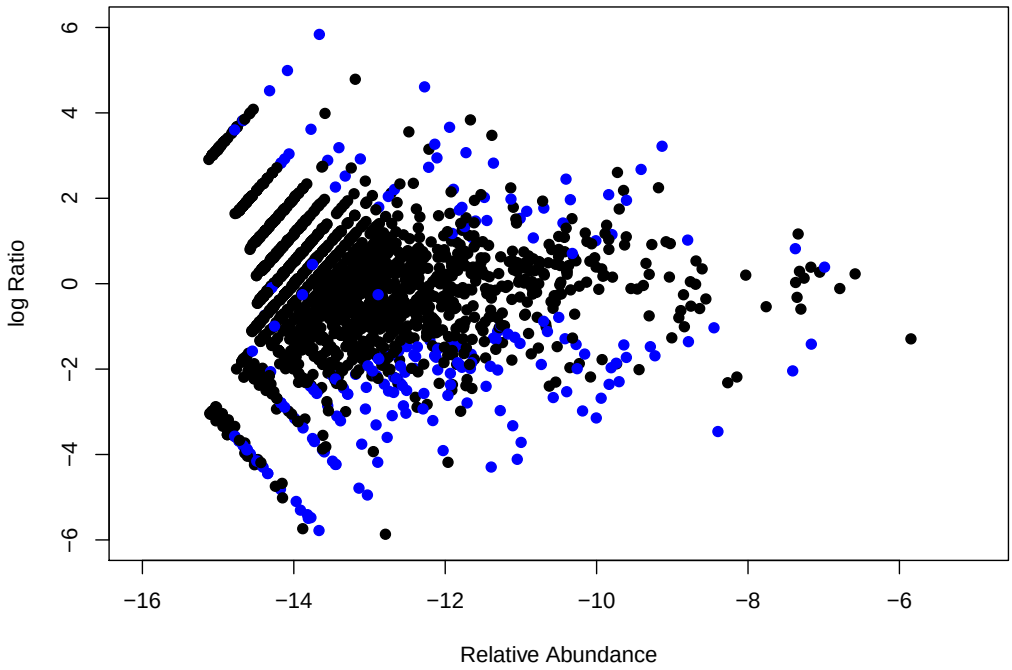
```
DGEList:
$data
           NC1 NC2 Tu102 Tu98
AAAAAAAAAA  17  39    34   40
AAAAATAAAG   4   4     5    1
AAAACATTCT 161 134   177   62
AAAACTGAGA   3   4     1    3
AAAAGAAACT   1   3    16    7
2452 more rows ...

$lib.size
  NC1   NC2 Tu102  Tu98
49610 48479 55700 41371

$group
[1] NC NC Tu Tu
Levels: NC Tu
```

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 :..

> alpha

EBList:
$alpha
[1] 1.235705e-07

$common.dispersion
[1] 0.2070189

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

Calculating shrinkage overdispersion parameters.
[quantileAdjust] Iteration (dot=1000) 1 :..
[quantileAdjust] Iteration (dot=1000) 2 :..
[quantileAdjust] Iteration (dot=1000) 3 :..
[quantileAdjust] Iteration (dot=1000) 4 :..
[quantileAdjust] Iteration (dot=1000) 5 :..

> ms

deDGEList:
$ps$p.common
  AAAAAAAAAA   AAAAATAAAG   AAAACATTCT   AAAACTGAGA   AAAAGAAACT
6.788158e-04 7.172804e-05 2.675299e-03 5.636977e-05 1.373587e-04
2452 more elements ...

$ps$p.group
               NC           Tu
[1,] 5.731440e-04 7.853135e-04
[2,] 8.155863e-05 6.179625e-05
[3,] 3.004834e-03 2.346097e-03
[4,] 7.136421e-05 4.121482e-05
[5,] 4.079830e-05 2.341273e-04
2452 more rows ...

$r
[1]   5.391110 999.000000   7.594998 999.000000  23.318710
2452 more elements ...

$group
[1] NC NC Tu Tu
Levels: NC Tu

$M
[1] 48519.75
```

It may be informative to look at the data via logRatio vs. Abundance (or M vs. A) plots, as is commonly done in the analysis of microarray experiments for 2-group comparisons. In addition, we can highlight those genes that are determined to be differentially expressed (here, using a 5% FDR multiple testing correction). To do this, you can call the commands:

```
> exactP <- exactTestNB(ms$pseudo, ms$group, pair = c("Tu", "NC"),
+     ms$M * ms$ps$p.common, ms$r, verbose = TRUE)

Calculating Fisher exact p-values (dot=1000):..

> exactPadj <- p.adjust(exactP, "fdr")
> k <- (exactPadj < 0.05)
> plotMA(ms, xlim = c(-16, -5), ylim = c(-6, 6), xlab = "Relative Abundance",
+     ylab = "log Ratio", col = c("black", "blue")[k + 1])
```



Or, you can have a look at the sorted list of differential expression calls, according to:

```
> topTags(ms)

No pair supplied, so comparing groups  NC  and  Tu
Calculating Fisher exact p-values (dot=1000):..
                    A         M      P.Value    adj.P.Val
CTAAGACTTC  -7.164019 -1.415383 3.804167e-54 9.346838e-51
TGCTCCTACC  -9.940205 -2.681416 9.237219e-29 1.134792e-25
GGGGTCAGGG  -9.605289  1.952856 3.394922e-25 2.780441e-22
GTGCTGAATG -10.395262 -2.529110 8.530643e-25 5.239947e-22
GATGACCCCC -12.027821 -3.907751 3.194294e-20 1.569676e-17
```

```
CGCTGTGGGG -11.273192 -2.971550 2.041462e-19 8.359786e-17
TTATGGGATC -11.725738  3.068306 2.003781e-15 7.033271e-13
TCACCGGTCA -11.048334 -4.111983 3.934644e-15 1.208428e-12
TAATTTTTGC -13.662883  5.837570 6.840848e-14 1.867551e-11
AGCAGATCAG  -9.640442 -1.433524 9.395166e-14 2.308392e-11
```

The above procedure will work for experiments with more than 2 groups as well. At present, the testing can only be done in a pairwise fashion. Therefore, with multiple groups you will need to specify the pair of groups to do the testing on.

3 Poisson example

It has been observed that in some 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 when the exact test is invoked, the dispersion parameter to 0. For example:

```
> set.seed(101)
> n <- 1000
> 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(rpois(4 * n, lambda = mu), nrow = n)
> dP <- DGEList(data = y, group = rep(1:2, each = 2), lib.size = lib.sizes)
> msP <- deDGE(dP, doPoisson = TRUE)

Quantile adjusting as Poisson.
[quantileAdjust] Iteration (dot=1000) 1 :.

> msP

deDGEList:
$ps$p.common
[1] 0.0017765587 0.0006020589 0.0032218999 0.0028955935 0.0014281982
995 more elements ...

$ps$p.group
                1           2
[1,] 0.0002778049 0.003487375
[2,] 0.0001777560 0.001089893
[3,] 0.0006445295 0.006141742
[4,] 0.0006782086 0.005410418
[5,] 0.0002556055 0.002769088
995 more rows ...

$r
[1] 1000 1000 1000 1000 1000
995 more elements ...
```

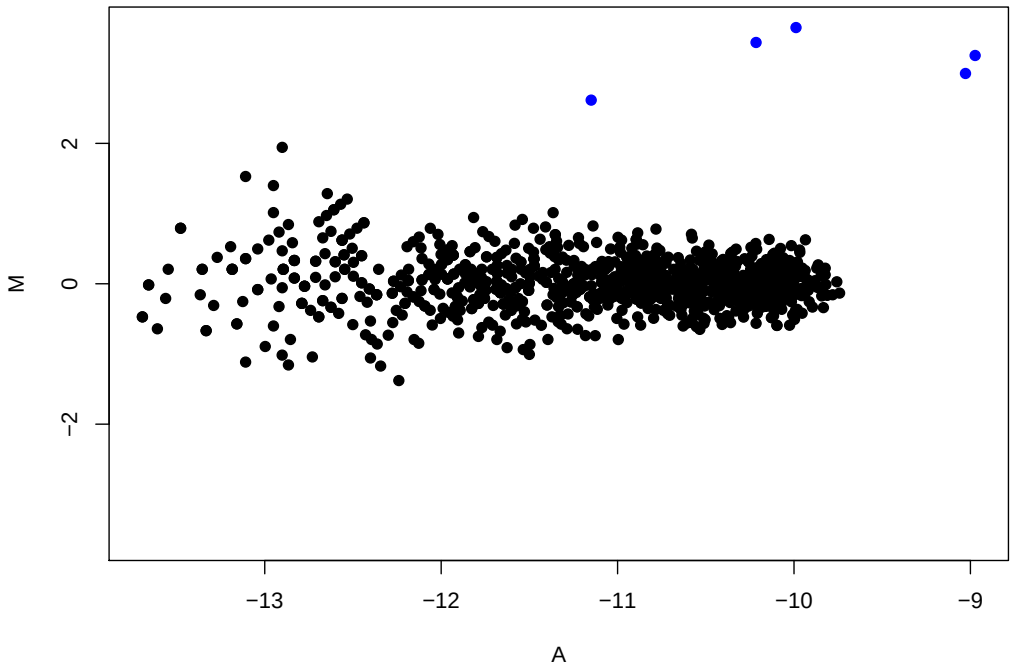
```
$group
[1] 1 1 2 2
Levels: 1 2

$M
[1] 41755.95
```

And you can proceed as before:

```
> topTags(msP)
```

```
No pair supplied, so comparing groups  1  and  2
Calculating Fisher exact p-values (dot=1000):.
             A          M      P.Value    adj.P.Val
3    -8.973301  3.2523296 4.634075e-85 4.634075e-82
4    -9.028014  2.9959392 4.933764e-70 2.466882e-67
1    -9.988642  3.6499978 6.859802e-57 2.286601e-54
5   -10.215084  3.4374201 6.548504e-43 1.637126e-40
2   -11.149706  2.6162169 2.261667e-14 4.523333e-12
344 -12.900979  1.9436057 3.835980e-03 6.393299e-01
733  -9.935133  0.6263777 7.534488e-03 7.887331e-01
802 -11.500153 -1.0062310 7.595947e-03 7.887331e-01
564 -10.025033 -0.5905975 9.262914e-03 7.887331e-01
124 -10.782468  0.7790100 9.470422e-03 7.887331e-01
```

```
> plotMA(msP, col = c(rep("blue", 5), rep("black", n - 5)))
```



4 Future improvements and extension

Here, we list some improvements/extensions 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. Some speed improvements have been made but as the datasets become larger, some further optimizations may be necessary. We are exploring various ways to do this.
3. The current quantile-based normalization assumes the library sizes as fixed. Depending on the circumstances of the samples in question, it may be necessary to explore something like an *effective* library size.
4. We are exploring more general testing procedures.

5 Setup

This vignette was built on:

```
> sessionInfo()

R version 2.9.1 (2009-06-26)
x86_64-unknown-linux-gnu

locale:
LC_CTYPE=en_US;LC_NUMERIC=C;LC_TIME=en_US;LC_COLLATE=en_US;LC_MONETARY=C;LC_MESSAGES=en_US;LC_PAPER=en_U

attached base packages:
[1] tools     stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] edgeR_1.2.4   Biobase_2.4.1
```

6 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.