BADER – Bayesian Analysis of Differential Expression in RNA Sequencing Data

Andreas Neudecker[1], Matthias Katzfuss[1]
[1]Institut für Angewandte Mathematik, Universität Heidelberg

April 15, 2025

1 Introduction

The BADER package is intended for the analysis of RNA sequencing data. The data come in form of count tables, $\{k_{ij}^T\}$, quantifying the expression levels for different entities j (e.g., genes), in samples i from a treatment group $T = A, B$. Due to low sample sizes there is considerable uncertainty in parameter estimation in this data. To not only estimate parameters but also quantify this uncertainity BADER uses a fully Bayesian approach.

The data is often assumed to be distributed according to an overdispersed poisson model. For example, Robinson and Smyth (2008) and Anders and Huber (2010) use a negative binomial model. In BADER we use the following two-stage model:

$$k_{ij}^T | \lambda_{ij}^T \overset{ind.}{\sim} Poi(s_i^T e^{\lambda_{ij}^T}); \quad \text{for all } i, j, T,$$
$$\lambda_{ij}^T | \mu_j^T, \alpha_j^T \overset{ind.}{\sim} N(\mu_j^T, e^{\alpha_j^T}); \quad \text{for all } i, j, T.$$

BADER estimates the posterior distributions of the log mean parameter μ_j^A, the log dispersion parameter α_j^T, the log fold change parameter $\gamma_j := \mu_j^B - \mu_j^A$, and the indicator variable showing whether a gene is differentially expressed.

2 Quickstart

In this section we will show how to use the BADER package on a dataset from Brooks et al. (2011). The dataset is a count table with expression measures for three treated and four untreated species of *drosohpila melanogaster*.

```
> library(BADER)
> datafile <- system.file( "extdata/pasilla_gene_counts.tsv", package="pasilla" )
> pasillaCountTable <- read.table( datafile, header=TRUE, row.names=1 )
> head(pasillaCountTable)
```

	untreated1	untreated2	untreated3	untreated4	treated1	treated2
FBgn0000003	0	0	0	0	0	0
FBgn0000008	92	161	76	70	140	88
FBgn0000014	5	1	0	0	4	0

```
FBgn0000015          0          2          1          2          1          0
FBgn0000017       4664       8714       3564       3150       6205       3072
FBgn0000018        583        761        245        310        722        299
           treated3
FBgn0000003        1
FBgn0000008       70
FBgn0000014        0
FBgn0000015        0
FBgn0000017     3334
FBgn0000018      308
```

The datasets consists of data created by both single-end and paired-end method. To keep things simple we only use the paired-end samples. We drop all genes with no counts for any sample. Also, as we don't want the code in this vignette to take a long time to run, we only use the first 500 genes. We describe the experiment design in the factor `design`.

```
> genetable <- pasillaCountTable[,c(3,4,6,7)]
> genetable <- genetable[rowSums(genetable) > 0,]
> genetable <- genetable[1:500,]
> design <- factor(c("A","A","B","B"))
```

Now we run the MCMC sampler on our data. For this vignette, we only use a low number of MCMC iterations. However, you will get better results if you let the MCMC sampler produce more samples from the posterior distribution.

```
> set.seed(21)
> results <- BADER(genetable,design,burn=1e3,reps=1e3,printEvery=1e3)
```

```
++++++++++++++++++++++++++++++++++++
++++++++++ Burn Step   1000 ++++++++++
++++++++++++++++++++++++++++++++++++
++++++++++ Repetition  1000 ++++++++++
++++++++++++++++++++++++++++++++++++
```

The `results` object is a list containing the posterior means of the log means, log dispersion parameters, and log fold change parameters, and the posterior probabilities of differential expression. For instance, we can print out a list of the 10 genes with the highest posterior probability of differential expression.

```
> names(results)
```

```
[1] "logMeanA"      "logMeanB"      "logFoldChange" "logDispA"
[5] "logDispB"      "diffProb"
```

```
> rownames(genetable)[order(results$diffProb,decreasing=TRUE)[1:10]]
```

```
 [1] "FBgn0000043" "FBgn0000071" "FBgn0000079" "FBgn0000406" "FBgn0000416"
 [6] "FBgn0000567" "FBgn0000579" "FBgn0001091" "FBgn0001137" "FBgn0001224"
```

3 The Posterior Distributions

We can control the output of the BADER function via the mode parameter. Setting mode="minimal" only returns the posterior means of the parameters listed above, mode="full" returns the full posterior distribution of the log fold change parameters, and mode="extended" also gives us the posterior distributions for the remaining parameters.

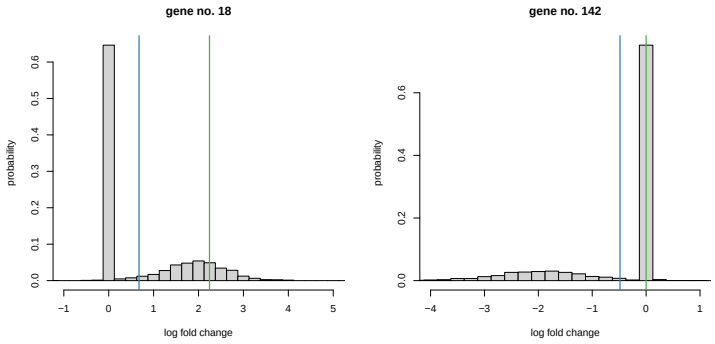
Having the full posterior distribution at hand lets us draw more sophisticated conclusions regarding genes with posterior probability of differential expression around 0.5. This is especially useful for datasets with strong overdispersion (i.e., low signal-to-noise ratio).

```
> set.seed(21)
> gam <- c(rnorm(50,0,2),rep(0,450))
> muA <- rnorm(500,3,1)
> muB <- muA + gam
> kA <- t(matrix(rnbinom(2*500,mu=exp(muA),size=exp(1)),nrow=2,byrow=TRUE))
> kB <- t(matrix(rnbinom(2*500,mu=exp(muB),size=exp(1)),nrow=2,byrow=TRUE))
> genetable <- cbind(kA,kB)
> design <- factor(c("A","A","B","B"))
> results <- BADER(genetable,design,mode="full",burn=1e3,reps=3e3,printEvery=1e3)

++++++++++++++++++++++++++++++++++++
++++++++++ Burn Step   1000 ++++++++++
++++++++++++++++++++++++++++++++++++
++++++++++ Repetition  1000 ++++++++++
++++++++++ Repetition  2000 ++++++++++
++++++++++ Repetition  3000 ++++++++++
++++++++++++++++++++++++++++++++++++
```

For example, for genes 18 and 142, we can compare the posterior distribution of the log fold change parameter with a point estimate (e.g., the posterior mean, in blue) and the true log fold change (in green). These distributions give us information not only about the probability of differential expression, but also about the uncertainty in the data regarding the log fold change parameter.

```
> par(mfrow=c(1,2))
> temp <- hist(results$logFoldChangeDist[,18],breaks=seq(-10.125,10.125,by=0.25),plot=FALS
> temp$density <- temp$density/sum(temp$density)
> plot(temp, freq=FALSE,ylab="probability",xlab="log fold change",main="gene no. 18",xlim=
> abline(v=gam[18],col="#4DAF4A", lwd=2)
> abline(v=results$logFoldChange[18],col="#377EB8",lwd=2)
> temp <- hist(results$logFoldChangeDist[,142],breaks=seq(-10.125,10.125,by=0.25),plot=FAL
> temp$density <- temp$density/sum(temp$density)
> plot(temp, freq=FALSE,ylab="probability",xlab="log fold change",main="gene no. 142",xlim
> abline(v=gam[142],col="#4DAF4A", lwd=2)
> abline(v=results$logFoldChange[142],col="#377EB8",lwd=2)
```



4 Gene Set Enrichment

In this section we show how to use the samples obtained from the posterior distribution can be used for further inference on sets or groups of genes ("gene set enrichment"). In our toy example we could assume that the genes $\mathcal{S} = \{1, 2, 3, 51, 52, 53\}$ and $\mathcal{T} = \{54, 55, 56\}$ belong to two groups and test whether these groups are enriched. We test the "competitive null hypothesis" (see Goeman and Bühlmann, 2007) that the genes in $\mathcal{S}, \mathcal{T}$ are at most as often differentially expressed as the remaining genes. We can test this hypothesis easily with the posterior distribution of the log fold change parameter.

```
> S <- c(1,2,3,51,52,53)
> T <- c(54,55,56)
> f <- function(sample,set)
+     mean(sample[set] != 0) > mean(sample[-set] != 0)
> mean(apply(results$logFoldChangeDist,1,f,set=S))
[1] 0.9983333
> mean(apply(results$logFoldChangeDist,1,f,set=T))
[1] 0.09233333
```

References

Anders, S. and Huber, W. (2010). Differential expression analysis for sequence count data. *Genome Biology*, 11(10):R106.

Brooks, A. N., Yang, L., Duff, M. O., Hansen, K. D., Park, J. W., Dudoit, S., Brenner, S. E., and Graveley, B. R. (2011). Conservation of an RNA regulatory map between Drosophila and mammals. *Genome Research*, 21(2):193–202.

Goeman, J. J. and Bühlmann, P. (2007). Analyzing gene expression data in terms of gene sets: methodological issues. *Bioinformatics (Oxford, England)*, 23(8):980–7.

Robinson, M. D. and Smyth, G. K. (2008). Small-sample estimation of negative binomial dispersion, with applications to SAGE data. *Biostatistics*, 9(2):321–32.