ABSSeq: a new RNA-Seq analysis method based on modelling absolute expression differences

Wentao Yang

February 17, 2017

1 Introduction

This vignette is intended to give a brief introduction of the `ABSSeq` `R` package by analyzing the simulated data from Soneson et al. [2] . For details about the approach, consult Yang [1]. Currently, `ABSSeq` can just be applied on pairwise study.

We assume that we have counts data from an experiment, which consists of two conditions and several replicates for each condition in a matrix. The expected expression of each gene is estimated from number of read count, proportional to the expectation value of the true concentration of count. As a result, a normalization method need to be apply on the original counts. The normalized counts usually have enormous variation across genes and compared conditions. The reliable identification of differential expression (DE) genes from such data requires a probabilistic model to account for ambiguity caused by sample size, biological and technical variations, levels of expression and outliers.

`ABSSeq` infers differential expression directly by the counts difference between conditions. It assumes that the sum counts difference between conditions follow a Negative binomial distribution with mean `mu` proportional to expression level and dispersion factor `r` (size). The `mu` and `r` is determined by variation in the experiment, i.e., biological variation, sequencing and mapping biases. Typically, the number of replicates in a study is small and not enough to reveal all variation. To overcome this problem, a common approach is to borrow information across genes. Here, we use local regression to smooth dispersion across genes. The smoothed dispersions are then used to produce pseudocounts in the `mu` estimation to accounts for dynamic dispersions at expression level, which in turn moderates the fold-change across expression level. However, the information borrowed across genes based on dispersions is usually incomplete since it often utilizes part of genes with a positive dispersion, which might load to underestimation and influence the DE inference. To overcome it, ABSSeq introduces a penalty for dispersion estimation, that helps avoid extremely siginicant DEs with small change at high expression level.

`ABSSeq` tests counts difference directly against a baseline estimated from the data set (`mu`), and therefore reports p-values related to magnitude of difference (fold-change). In addition, `ABSSeq` moderates the fold-changes by two steps: the expression level and gene-specific dispersion, that might facilitate the gene ranking by fold-change and visualization (Heatmap). New alternative approach

(named aFold) was introduced, which calls DE genes via log fold-change (see last Section for example).

2 Pairwise study

We firstly import the ABSSeq package.

```
> library(ABSSeq)
```

Then, we load a simulated data set. It is a list and contains three elements: the counts matrix, denoted by 'counts', the groups, denoted by 'groups' and differential expression genes, denoted by 'DEs'.

```
> data(simuN5)
> names(simuN5)
```

```
[1] "counts" "groups" "DEs"
```

The data is simulated from Negative binomial distribution with means and variances from Pickrell's data [3] and added outliers randomly [2]. This data includes group informtion.

```
> simuN5$groups
```

```
 [1] 0 0 0 0 0 1 1 1 1 1
```

But we also can define groups as

```
> conditions <- factor(c(rep(1,5),rep(2,5)))
```

We construct an ABSDataSet object by combining the counts matrix and defined groups with the ABSDataSet function. Here, we can also initiate a paired comparison for specific samples, such as data for cancer and normal tissue from same individuals, by seting the paired parameter in ABSDataSet object.

```
> obj <- ABSDataSet(simuN5$counts, factor(simuN5$groups))
> obj1 <- ABSDataSet(simuN5$counts, conditions)
> pairedobj <- ABSDataSet(simuN5$counts, conditions,paired=TRUE)
```

The default normalization method is quartile, used the up quantile of data. However, there are also other choices for users, that is, total by total reads count, geometric from DESeq [4] and user through size factors provided by users. The normalization method can be checked and revised by normMethod.

```
> obj1 <- ABSDataSet(simuN5$counts, factor(simuN5$groups),normMethod="user",sizeFactor=run
> normMethod(obj1)
```

```
[1] "user"
```

```
> normMethod(obj1) <- "geometric"
> normMethod(obj1)
```

```
[1] "geometric"
```

Once we get the `ABSDataSet` object, We can estimate the size factor for each sample by selected method as mentioned above used the function `normalFactors`. And we can see the size factors by `sFactors`.

```
>   obj=normalFactors(obj)
>   sFactors(obj)
```

```
 [1] 1.2876030 1.1171328 0.7203705 1.1641544 1.1394777 0.9496168 0.8926659
 [8] 1.1102453 0.7994882 0.8192454
```

Then, we can get the normalized counts by `counts`.

```
>   head(counts(obj,norm=TRUE))
```

```
        [,1]        [,2]       [,3]         [,4]       [,5]        [,6]
1   57.47113   14.322379   47.19794    0.8589926    1.75519    51.59976
2 1432.11839  969.446046 1211.87641 2077.0441527 2946.96413  6379.41563
3 2626.58590 2248.613544 2387.66028 1509.2500315 1951.77136  3498.25334
4   24.85238    9.846636   12.49357   18.0388450   18.42950    17.90196
5 1470.95023 3679.956322 3502.36448 2296.9462631 5162.01400 12936.79779
6  835.66127  833.383443  527.50634  131.4258707 1351.49636  1450.05865
         [,7]         [,8]         [,9]       [,10]
1     6.72144     0.000000     7.504801    34.17779
2 59557.55531  4812.449949 12462.973271 11793.78046
3  3261.01840  3176.775402 30241.847425 41320.95384
4   196.04199     4.503509    25.016004    58.59051
5  8564.23418 18707.577287 15971.467854 16782.51785
6   979.08969  1306.918375  1454.680642  1524.57377
```

With the size factors, we can calculate the absolate counts difference between conditions, mean (`mu`), size factor (`r`) and moderate log2 of fold-change for each gene. It can be done by function `callParameter` as

```
>   obj=callParameter(obj)
```

If we want to see correlation between the absolute log2 fold-change (with or without moderation) and expression level in same conditions, we can use function `plotDifftoBase`.

```
> obj <- callDEs(obj)
> plotDifftoBase(obj)
```

In the end, we model the counts differences with Negative binomial distribution and calculate the pvalue for each gene. It can be done by the function `callDEs`, which reports pvalues as well as adjusted pvalue, that can be accessed by `results` with names of `pvalue` and `adj.pvalue`. Noticely, this function also provides fold-change moderation according to gene-specific dispersion by utilizing `qnbinom`, which will report fold-changes closer to gene's dipersion. In the end, ABSSeq produces three kinds fold-changes: the original (denoted by 'rawFC'), corrected by expression level (denoted by 'lowFC') and moderated by expression level and gene-specific dispersion (denoted by 'foldChange'), which are stored in the `ABSDataSet` object and could be also retrieved by `results`.

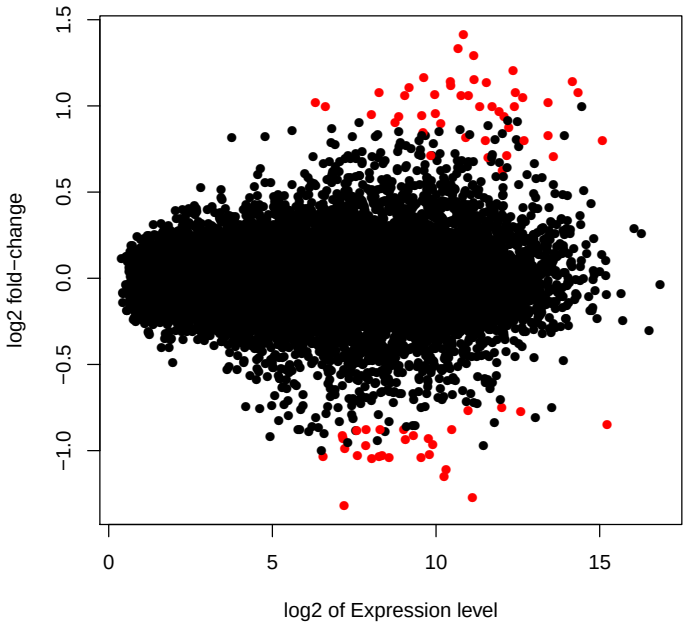

Figure 1: 'Absolute log2 fold-change against expression level'-plot for count data. We show the fitted and raw data with different colors.

```
> obj <- callDEs(obj)
> head(results(obj,c("rawFC","lowFC","foldChange","pvalue","adj.pvalue")))
```

```
       rawFC      lowFC  foldChange       pvalue adj.pvalue
1 -0.1728137 -0.0982513 -0.04054861 7.769550e-01 1.00000000
2  2.9334944  2.6952604  0.35465479 4.372581e-02 1.00000000
3  2.0176521  0.9252355  0.52541739 2.038786e-02 0.76740895
4  0.9009722  0.6176966  0.22072718 1.462254e-01 1.00000000
5  2.2538040  2.1368378  1.04714982 3.376019e-05 0.01168384
6  1.1932075  1.0136640  0.41574774 6.404134e-02 1.00000000
```

The `results` function can be used to access all information in an `ABSDataSet`.

```
> head(results(obj))
```

```
      Amean     Bmean    baseMean  absD     Variance      rawFC      lowFC
1  3.550959  3.378145    61.27647    22 1.178486e+03 -0.1728137 -0.0982513
2 10.639635 13.573130 23977.98994 86369 5.256701e+08  2.9334944  2.6952604
3 11.041770 13.059422  5567.17190 10840 2.077424e+06  2.0176521  0.9252355
4  4.083330  4.984303   116.53577   218 6.180113e+03  0.9009722  0.6176966
5 11.528806 13.782610 18467.70434 56850 1.767170e+07  2.2538040  2.1368378
6  9.181859 10.375066  1795.74630  3036 2.494667e+05  1.1932075  1.0136640
```

```
   foldChange       pvalue adj.pvalue trimmed
1 -0.04054861 7.769550e-01 1.00000000       0
2  0.35465479 4.372581e-02 1.00000000       0
3  0.52541739 2.038786e-02 0.76740895       2
4  0.22072718 1.462254e-01 1.00000000       0
5  1.04714982 3.376019e-05 0.01168384       0
6  0.41574774 6.404134e-02 1.00000000       0
```

Besides, we can also get this result by the function `ABSSeq`, which perfoms a default analysis by calling above functions in order and returns a `ABSDataSet` object with all information.

```
> data(simuN5)
> obj <- ABSDataSet(simuN5$counts, factor(simuN5$groups))
> obj <- ABSSeq(obj)
> res=results(obj,c("Amean","Bmean","foldChange","pvalue","adj.pvalue"))
> head(res)

      Amean     Bmean  foldChange       pvalue adj.pvalue
1  3.550959  3.378145 -0.04054861 7.769550e-01 1.00000000
2 10.639635 13.573130  0.35465479 4.372581e-02 1.00000000
3 11.041770 13.059422  0.52541739 2.038786e-02 0.76740895
4  4.083330  4.984303  0.22072718 1.462254e-01 1.00000000
5 11.528806 13.782610  1.04714982 3.376019e-05 0.01168384
6  9.181859 10.375066  0.41574774 6.404134e-02 1.00000000
```

Morever, ABSSeq also allow testing on user-defined baseline for counts difference by giving a same value to `minRates` and `maxRates` as

```
> data(simuN5)
> obj <- ABSDataSet(simuN5$counts, factor(simuN5$groups),minRates=0.2, maxRates=0.2)
> #or by slot functions
> #minRates(obj) <- 0.2
> #maxRates(obj) <- 0.2
> obj <- ABSSeq(obj)
> res=results(obj,c("Amean","Bmean","foldChange","pvalue","adj.pvalue"))
> head(res)

      Amean     Bmean  foldChange       pvalue   adj.pvalue
1  3.550959  3.378145 -0.04054861 7.143875e-01 1.0000000000
2 10.639635 13.573130  0.35465479 2.312239e-02 0.4145062908
3 11.041770 13.059422  0.52541739 2.905610e-03 0.1198708306
4  4.083330  4.984303  0.22072718 8.973495e-02 0.6259841791
5 11.528806 13.782610  1.04714982 4.415481e-07 0.0001410576
6  9.181859 10.375066  0.41574774 1.549891e-02 0.3396090201
```

ABSSeq penalizes the dispersion estimation by adding a value to the observed dispersion for each gene, which is obained by quantile estimation on the all observed dispersions. It also allow penalty of value provided by user as

```
> data(simuN5)
> obj <- ABSDataSet(simuN5$counts, factor(simuN5$groups),minDispersion=0.1)
```

```
> #or by slot functions
> #minimalDispersion(obj) <- 0.2
> obj <- ABSSeq(obj)
> res=results(obj,c("Amean","Bmean","foldChange","pvalue","adj.pvalue"))
> head(res)

      Amean     Bmean  foldChange       pvalue adj.pvalue
1  3.550959  3.378145 -0.04054861 0.7615146075  1.0000000
2 10.639635 13.573130  0.35465479 0.0453497761  1.0000000
3 11.041770 13.059422  0.52541739 0.0361365294  1.0000000
4  4.083330  4.984303  0.22072718 0.1492710375  1.0000000
5 11.528806 13.782610  1.04714982 0.0003602342  0.1662281
6  9.181859 10.375066  0.41574774 0.0860743591  1.0000000
```

In addition, ABSSeq provides special parameter estimation for data set without replicates. It firstly treat the two groups as replicates and separate genes into two sets by expression level depended fold-change cutoffs. Then the set with fold-change under cutoffs is used to estimate the dispersion for each gene by local regression as well as fold-change moderation. Here is the example, which replaces the `callParameter` by `callParameterwithoutReplicates`.

```
> data(simuN5)
> obj <- ABSDataSet(simuN5$counts[,c(1,2)], factor(c(1,2)))
> obj <- ABSSeq(obj)
> res=results(obj,c("Amean","Bmean","foldChange","pvalue","adj.pvalue"))
> head(res)

      Amean     Bmean   foldChange       pvalue  adj.pvalue
1  6.131372  4.187512 -1.286383183 0.0036671446 0.019625839
2 10.750651 10.188132 -0.541720688 0.1780842534 0.392352204
3 11.625308 11.401232 -0.218566382 0.8885395599 1.000000000
4  4.948680  3.682492 -0.726325630 0.0882123425 0.234446277
5 10.789227 12.111677  1.280819536 0.0009870077 0.006777839
6  9.974088  9.970154 -0.003775191 0.9999999998 1.000000000
```

3 Detecting DE via aFold

Recently, ABSSeq integrates a new method for DE detection: aFold. aFold utilizes a ploynormial function to model the uncertainty of observed reads count and moderate the fold-change calculation. aFold takes into account variations among samples and genes and reports DE and fold-change in a reliable way. The fold-change produced by aFold may help the experimentalist to avoid arbitrary choice of cut-off thresholds and may enhance subsequent downstream functional analyses. Here is the example for how to use aFold in ABSSeq.

```
> data(simuN5)
> obj <- ABSDataSet(counts=simuN5$counts, groups=factor(simuN5$groups))
> obj <- ABSSeq(obj, useaFold=TRUE)
> res=results(obj,c("Amean","Bmean","foldChange","pvalue","adj.pvalue"))
> head(res)
```

```
      Amean     Bmean  foldChange       pvalue   adj.pvalue
1  3.550959  3.378145 -0.04054861 8.215065e-01 9.915090e-01
2 10.639635 13.573130  0.35465479 4.846615e-02 5.966797e-01
3 11.041770 13.059422  0.52541739 3.462742e-03 1.133895e-01
4  4.083330  4.984303  0.22072718 2.194093e-01 9.699548e-01
5 11.528806 13.782610  1.04714982 5.669672e-09 3.071237e-06
6  9.181859 10.375066  0.41574774 2.071317e-02 3.823190e-01
```

References

[1] Wentao Yang, Philip Rosenstielb and Hinrich Schulenburg. *ABSSeq: a new RNA-Seq analysis method based on modelling absolute expression differences.* (2016).

[2] Soneson C, Delorenzi M *A comparison of methods for differential expression analysis of RNA-seq data.* BMC Bioinformatics 2013, 14(1):91.

[3] Pickrell JK, Marioni JC, Pai AA, Degner JF, Engelhardt BE, Nkadori E, Veyrieras J-B, Stephens M, Gilad Y, Pritchard JK *Understanding mechanisms underlying human gene expression variation with RNA sequencing* Nature 2010, 464(7289):768-772.

[4] Anders S, Huber W *Differential expression analysis for sequence count data.* Genome Biol 2010, 11(10):R106.