

ABSSeq: a new RNA-Seq analysis method based on absolute expression differences and generalized Poisson model

Wentao Yang

October 13, 2014

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 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 by the absolute expression differences between conditions. It assumes that the absolute expression difference of each gene between conditions is contributed by two parts, the expression variation across samples and the differential expression. If one gene belongs to differential expression gene, its absolute expression difference should be larger than its expression variation and also relative large among the changes of all gene. Based on this hypothesis, **ABSSeq** employs two generalized Poisson model to account for the variation across samples and overall changes. It calculates a pvalue according to built model.

ABSSeq tests null hypothesis which takes into account the magnitude of expression difference through two directions: samples and genes, and therefore detects differential expression genes which are closer to the biological concept of differential expression.

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 information.

```
> 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.

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

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, `DESeq` 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),"User",runif(10,1,2))
> normMethod(obj1)
```

```
[1] "User"
```

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

```
[1] "DESeq"
```

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 `sizeFactors`.

```
> obj=normalFactors(obj)
> sizeFactors(obj)
```

```
[1] 1.2795846 1.1467525 0.7173574 1.1442041 1.1289141 0.9403370 0.8919186
[8] 1.1250916 0.8001784 0.8256618
```

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

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

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]
1	57.83127	13.952444	47.39618	0.8739699	1.771614	52.10898
2	1441.09266	944.406083	1216.96665	2113.2592984	2974.539898	6442.37107
3	2643.04521	2190.533778	2397.68917	1535.5651726	1970.034763	3532.77595
4	25.00812	9.592306	12.54605	18.3533686	18.601947	18.07862
5	1480.16784	3584.906194	3517.07544	2336.9956013	5210.316761	13064.46494
6	840.89788	811.857861	529.72202	133.7173998	1364.142777	1464.36860

	[,7]	[,8]	[,9]	[,10]
1	6.727071	0.000000	7.498328	33.91219
2	59607.458750	4748.946744	12452.223408	11702.12832
3	3263.750821	3134.855917	30215.762500	40999.83935
4	196.206250	4.444083	24.994427	58.13519
5	8571.410179	18460.719422	15957.691760	16652.09711
6	979.910071	1289.672792	1453.425916	1512.72596

With the size factors, we can calculate the absolute difference between conditions, variances, log2 of fold-change for each gene. It can be done by function `calPara` as

```
> obj=calPara(obj)
```

If we want to see correlation between the absolute difference and expression level, we can use function `plotDiffToBase`.

```
> plotDiffToBase(obj)
```

In the end, we model the data with generalized Poisson distribution and calculate the pvalue for each gene based on the absolute difference. It can be done by the function `GPTest`, which reports pvalues as well as adjusted pvalue, which can be accessed by `results` with names of `pvalue` and `adj.pvalue`.

```
> obj <- GPTest(obj)
> head(results(obj,c("pvalue","adj.pvalue")))
```

	pvalue	adj.pvalue
1	7.744078e-01	9.636577e-01
2	1.258483e-03	4.249171e-02
3	4.078875e-01	9.636577e-01
4	3.973940e-01	9.636577e-01
5	1.156328e-25	3.430164e-23
6	6.773337e-04	2.496716e-02

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

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

	baseMean	Amean	Bmean	absD	foldChange	Variance
1	18.93221	21.3625	16.50193	4.860566	-0.3724455	4.848230e+02
2	5626.96851	1607.7339	9646.20313	8038.469224	2.5849325	1.456647e+07
3	6676.06005	2182.0836	11170.03651	8987.952925	2.3558556	1.066942e+08
4	24.55270	16.7090	32.39640	15.687401	0.9552081	3.325769e+02
5	9177.72201	3178.6908	15176.75319	11998.062359	2.2553586	5.171148e+06

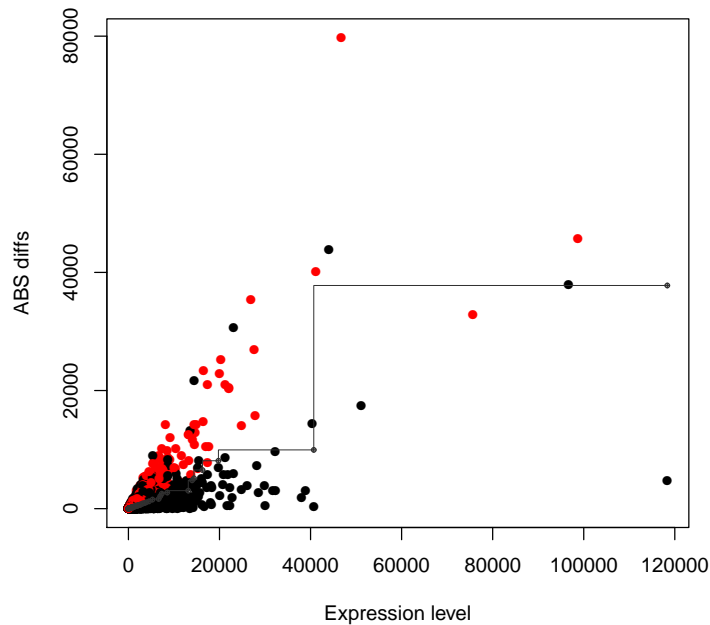


Figure 1: 'Absolute difference against expression level'-plot for count data. We show the correlation by isoreg and marked genes with different color according to a given fold-change.

```

6 1061.36066 731.1881 1391.53320 660.345078 0.9283608 7.279936e+04
      pvalue  adj.pvalue
1 7.744078e-01 9.636577e-01
2 1.258483e-03 4.249171e-02
3 4.078875e-01 9.636577e-01
4 3.973940e-01 9.636577e-01
5 1.156328e-25 3.430164e-23
6 6.773337e-04 2.496716e-02

```

Besides, we can also get this result by the function `ABSSeq`, which performs a default analysis by calling above functions in order and returns a table with mean expression of each group, log2 fold-change, pvalue and adjusted pvalue.

```

> data(simuN5)
> obj <- ABSDataSet(simuN5$counts, factor(simuN5$groups))
> res <- ABSSeq(obj)
> head(res)

      Amean      Bmean foldChange      pvalue  adj.pvalue
1   21.3625   16.50193 -0.3724455 7.757480e-01 9.641876e-01
2  1607.7339  9646.20313  2.5849325 1.257010e-03 4.242115e-02

```

3	2182.0836	11170.03651	2.3558556	4.077720e-01	9.641876e-01
4	16.7090	32.39640	0.9552081	3.978864e-01	9.641876e-01
5	3178.6908	15176.75319	2.2553586	1.157152e-25	3.432608e-23
6	731.1881	1391.53320	0.9283608	6.742083e-04	2.485196e-02

References

- [1] Wentao Yang, Philip Rosenstielb and Hinrich Schulenburg. *ABSSeq: a new RNA-Seq analysis method based on absolute expression differences and generalized Poisson model*. (2014).
- [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.