Differential expression analysis of count data using XBSeq package

Yuanhang Liu

2015-06-01

Introduction

XBSeq is a novel algorithm for testing RNA-seq differential expression (DE), where a statistical model was established based on the assumption that observed signals are the convolution of true expression signals and sequencing noises. The mapped reads in non-exonic regions are considered as sequencing noises, which follows a Poisson distribution. Given measurable observed signal and background noise from RNA-seq data, true expression signals, assuming governed by the negative binomial distribution, can be delineated and thus the accurate detection of differential expressed genes

Installation

XBSeq can be installed from Bioconductor by

```
source('http://www.bioconductor.org/biocLite.R')
biocLite("XBSeq")
```

```
library("XBSeq")
```

If you would like to install the development version of XBSeq, it is recommended that you refer to the github page of XBSeq.

Use XBSeq for testing differential expression

HTSeq procedure

In order to use XBSeq for testing DE, after sequence alignment, we need to run HTSeq twice to measure the reads mapped to exonic regions (observed signal) and non-exonic regions (background noise). Generally speaking, you will need to run the following code to generate observed signal and background noise.

```
htseq-count [options] <alignment_file> <gtf_file> > Observed_count.txt
htseq-count [options] <alignment_file> <gtf_file_bg> > background_count.txt
```

Details regarding how HTSeq works can be found here: http://www-huber.embl.de/HTSeq/doc/count.html

The gtf file used to measure observed signal can be downloaded from UCSC database: http://genome.ucsc.edu. The gtf file used to measure background noise can be downloaded in the gtf folder from github: https://github.com/Liuy12/XBSeq. If you would like to construct the gtf file by yourself, we also have deposited the perl script we used to construct the gtf file in github. Details regarding the procedure we used to construct the background gtf file can be found in the Details section in the vignette.

XBSeq testing for DE

After HTSeq procedure, then we will have two measurements for each gene, the observed signal and background noise. Here we will use a mouse RNA-seq dataset, which contains 3 replicates of wild type mouse liver tissues (WT) and 3 replicates of Myc transgenic mouse liver tissues (MYC). The dataset is obtained from Gene Expression Omnibus (GSE61875).

As a preliminary step, we have already carried out HTSeq procedure mentioned above to generate observed signal and background noise for each gene. The two datasets can be loaded into user's working space by

```
data(ExampleData)
```

We can first take a look at the two datasets:

```
head(Observed)
```

```
##               Sample_54_WT Sample_72_WT Sample_93_WT Sample_31_MYC
## 0610005C13Rik         3683         2956         3237          2985
## 0610007C21Rik         2136         1675         1519          1782
## 0610007L01Rik         2826         3584         1712          2694
## 0610007P08Rik          717          616          529           737
## 0610007P14Rik         3138         2145         2145          1995
## 0610007P22Rik          298          254          194           211
##               Sample_75_MYC Sample_87_MYC
## 0610005C13Rik          4043          4437
## 0610007C21Rik          2265          2214
## 0610007L01Rik          3133          3552
## 0610007P08Rik           864           923
## 0610007P14Rik          2871          2945
## 0610007P22Rik           272           333
```

```
head(Background)
```

```
##               Sample_54_WT Sample_72_WT Sample_93_WT Sample_31_MYC
## 0610005C13Rik          512          374          466           496
## 0610007C21Rik           50           44           40            42
## 0610007L01Rik           33           22           35            39
## 0610007P08Rik           14           14           28            20
## 0610007P14Rik           21           17           22            10
## 0610007P22Rik           16           19           26            14
##               Sample_75_MYC Sample_87_MYC
## 0610005C13Rik           504           648
## 0610007C21Rik            42            60
## 0610007L01Rik            40            22
## 0610007P08Rik            14            18
## 0610007P14Rik            24            23
## 0610007P22Rik            28            28
```

Rows represent reads mapped to each gene or corresponding background region. Column represent samples. And differential expression analysis will be carried out as follows:

Firstly, we need to construct a XBSeqDataSet object. Conditions are the design matrix for the experiment. Observe and background are the output matrix from HTSeq (Remeber to remove the bottom few lines of summary statistics of the output matrix).

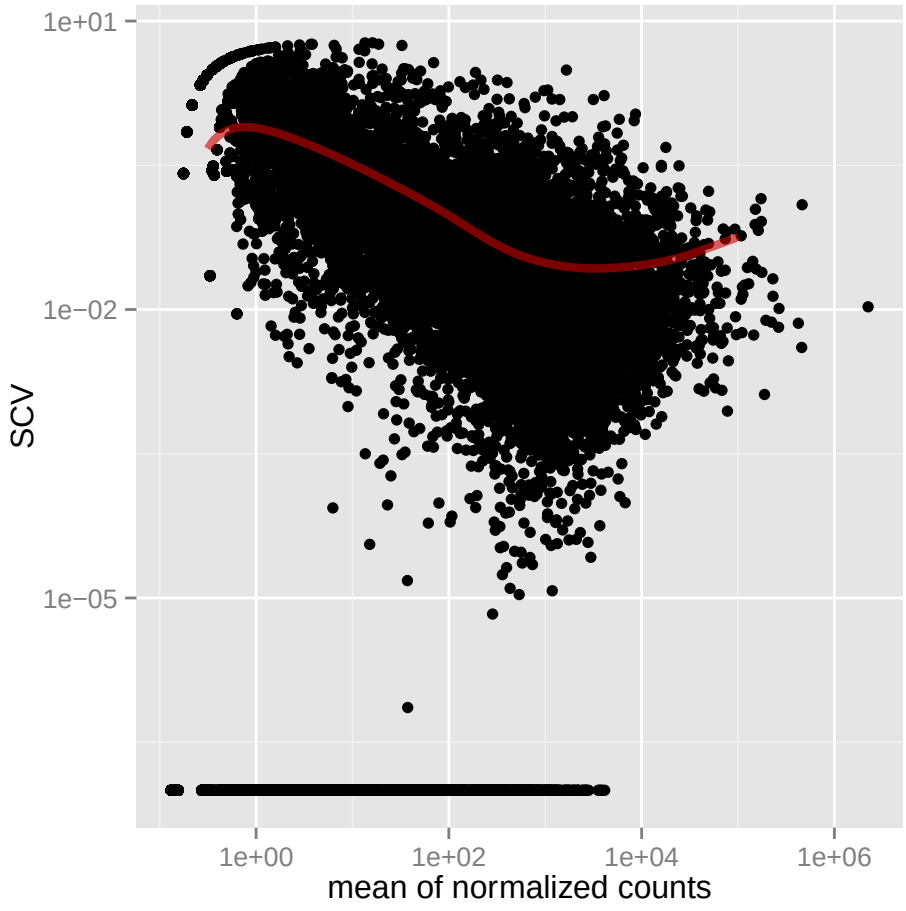
```
conditions <- factor(c(rep("C", 3), rep("T", 3)))
XB <- XBSeqDataSet(Observed, Background, conditions)
```

Then estimate the preliminary underlying signal followed by normalizing factor and dispersion estimates

```
XB <- estimateRealCount(XB)
XB <- estimateSizeFactors(XB)
XB <- estimateSCV(XB, method = "pooled", sharingMode = "maximum", fitType = "local")
```

Take a look at the scv fitting information

```
plotSCVEsts(XB)
```

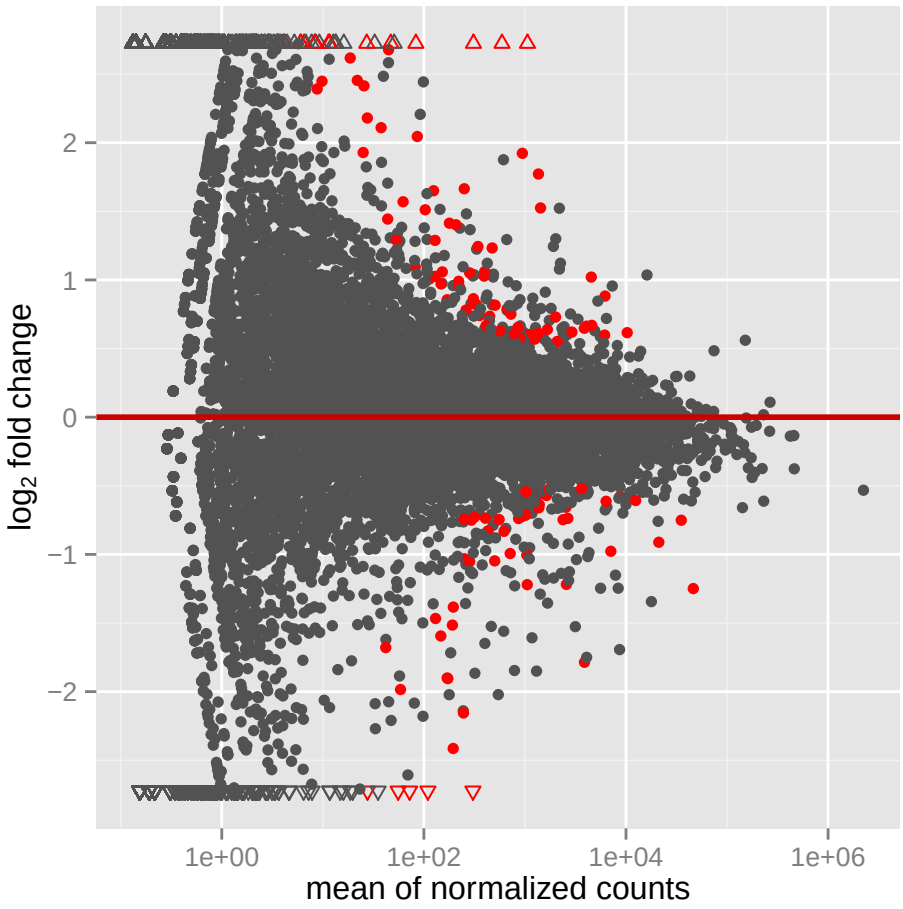


Carry out the DE test by using function XBSeqTest

```
Teststas <- XBSeqTest( XB, levels(conditions)[1L], levels(conditions)[2L] )
```

Plot Maplot based on test statistics

```
MAplot(Teststas, padj = FALSE, pcuff = 0.01, lfccuff = 1)
```




```
# Alternatively, all the codes above can be done with a wrapper function
# XBSeq
Teststats <- XBSeq(Observed, Background, conditions, method = "pooled", sharingMode = "maximum",
    fitType = "local", pvals_only = FALSE)
```

Compare the results with DESeq

Now we will carry out DE analysis on the same dataset by using DESeq and then compare the results obtained by these two methods

If you have not installed DESeq before, DESeq is also available from Bioconductor

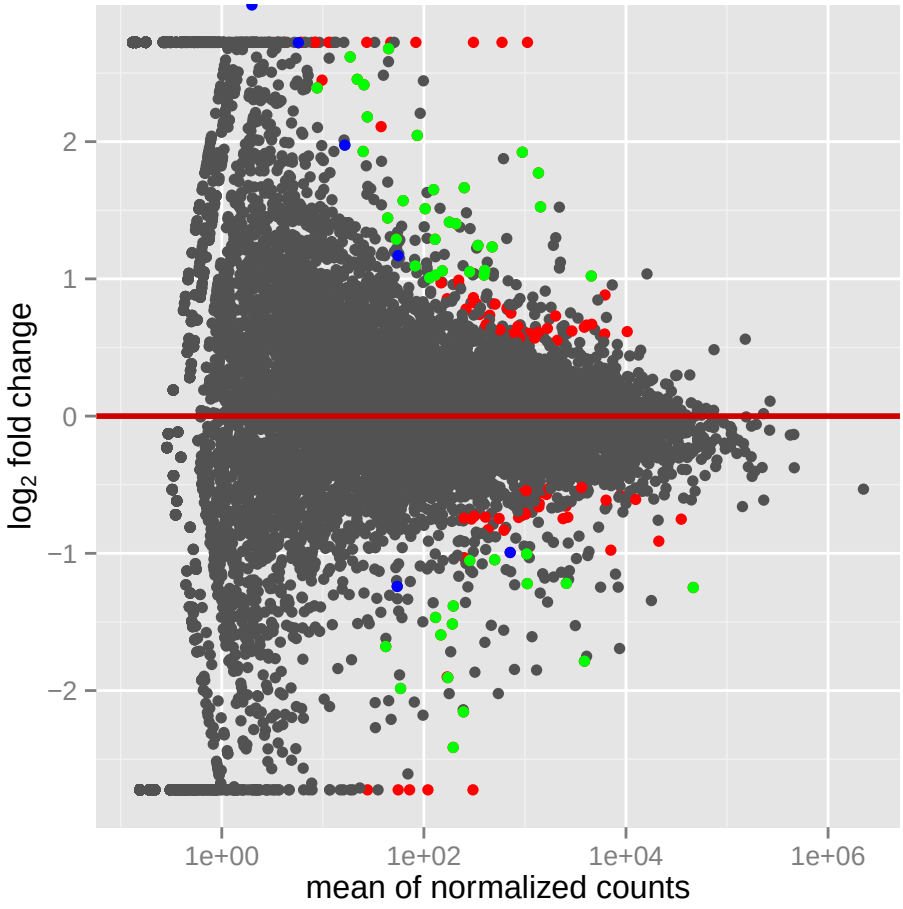
```
biocLite("DESeq")
```

Then DE analysis for DESeq can be carried out by:

```
library('DESeq')
library('ggplot2')
de <- newCountDataSet(Observed, conditions)
de <- estimateSizeFactors(de)
de <- estimateDispersions(de, method = "pooled", fitType="local")
res <- nbinomTest(de, levels(conditions)[1], levels(conditions)[2])
```

Then we can compare the results from XBSeq and DESeq

```
DE_index_DESeq <- with(res, which(pval < 0.01 & abs(log2FoldChange) > 1))
DE_index_XBSeq <- with(Teststas, which(pval < 0.01 & abs(log2FoldChange) > 1))
DE_index_inters <- intersect(DE_index_DESeq, DE_index_XBSeq)
DE_index_DESeq_uniq <- setdiff(DE_index_DESeq, DE_index_XBSeq)
DE_plot <- MAplot(Teststas, padj = FALSE, pcuff = 0.01, lfccuff = 1, shape = 16)
DE_plot + geom_point(data = Teststas[DE_index_inters, ], aes(x = baseMean, y = log2FoldChange),
    color = "green", shape = 16) + geom_point(data = Teststas[DE_index_DESeq_uniq,
    ], aes(x = baseMean, y = log2FoldChange), color = "blue", shape = 16)
```



The red dots indicate DE genes identified only by XBSeq. Then green dots are the shared results of XBSeq and DESeq. The blue dots are DE genes identified only by DESeq.

Details

Construction of gtf file for background region

- Exonic region annotation is obtained from UCSC database.
- Non-exonic regions are constructed by following several criteria:

  1. Download refFlat table from UCSC database
  2. Several functional elements (mRNA, peudo genes, etc.) of the genome are excluded from the whole genome annotation.

3. Construct the background region for each gene by making the region to have the same sturcture or length as the exonic region of the gene.

More details regarding how do we construct the background region annotation file of an real example can be found in manual page of ExampleData and also our publication of XBSeq.

Bug reports

Report bugs as issues on our GitHub repository or you can report directly to my email: liuy12@uthscsa.edu.

Session information

```
sessionInfo()
```

```
## R version 3.1.3 (2015-03-09)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.10.3 (Yosemite)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] ggplot2_1.0.1             DESeq_1.18.0
##  [3] lattice_0.20-31           locfit_1.5-9.1
##  [5] Biobase_2.26.0            XBSeq_0.99.5
##  [7] DESeq2_1.6.3              RcppArmadillo_0.5.100.1.0
##  [9] Rcpp_0.11.6               GenomicRanges_1.18.4
## [11] GenomeInfoDb_1.2.5        IRanges_2.0.1
## [13] S4Vectors_0.4.0           BiocGenerics_0.12.1
##
## loaded via a namespace (and not attached):
##  [1] acepack_1.3-3.3      annotate_1.44.0      AnnotationDbi_1.28.2
##  [4] base64enc_0.1-2      BatchJobs_1.6        BBmisc_1.9
##  [7] BiocParallel_1.0.3   brew_1.0-6           checkmate_1.5.2
## [10] cluster_2.0.1        codetools_0.2-11     colorspace_1.2-6
## [13] DBI_0.3.1            digest_0.6.8         evaluate_0.7
## [16] fail_1.2             foreach_1.4.2        foreign_0.8-63
## [19] formatR_1.2          Formula_1.2-1        genefilter_1.48.1
## [22] geneplotter_1.44.0   grid_3.1.3           gridExtra_0.9.1
## [25] gtable_0.1.2         Hmisc_3.16-0         htmltools_0.2.6
## [28] iterators_1.0.7      knitr_1.10           labeling_0.3
## [31] latticeExtra_0.6-26  magrittr_1.5         MASS_7.3-40
## [34] matrixStats_0.14.0   munsell_0.4.2        nnet_7.3-9
## [37] plyr_1.8.2           pracma_1.8.3         proto_0.3-10
## [40] RColorBrewer_1.1-2   reshape2_1.4.1       rmarkdown_0.5.1
```

```
## [43] rpart_4.1-9          RSQLite_1.0.0        scales_0.2.4
## [46] sendmailR_1.2-1      splines_3.1.3        stringi_0.4-1
## [49] stringr_1.0.0        survival_2.38-1      tools_3.1.3
## [52] XML_3.98-1.1         xtable_1.7-4         XVector_0.6.0
## [55] yaml_2.1.13
```

Acknowledgements

XBSeq is implemented in R based on the source code from DESeq and DESeq2.

References

Hung-I Harry Chen, Yuanhang Liu, Yi Zou, Zhao Lai, Devanand Sarkar, Yufei Huang, Yidong Chen, doi: http://dx.doi.org/10.1101/016196