metaSeq: Meta-analysis of RNA-seq count data

Koki Tsuyuzaki¹, and Itoshi Nikaido².

April 24, 2017

¹Department of Medical and Life Science, Tokyo University of Science.
²Bioinformatics Research Unit, Advanced Center for Computing and Communication, RIKEN.

k.t.the-answer@hotmail.co.jp

Contents

1 Introduction 2
2 RSE: Read-Size Effect 2
3 Robustness against RSE 3
4 Getting started 4
5 Meta-analysis by non-NOISeq method 7
6 Setup 9
1 Introduction

This document provides the way to perform meta-analysis of RNA-seq data using \texttt{metaSeq} package. Meta-analysis is an attempt to integrate multiple data in different studies and retrieve much reliable and reproducible result. In transcriptome study, the goal of analysis may be differentially expressed genes (DEGs). In our package, the probability of one-sided \textit{NOISeq} \cite{1} is applied in each study. This is because the numbers of reads are often different depending on its study and \textit{NOISeq} is robust method against its difference (see the next section). By meta-analysis, genes which differentially expressed in many studies are detected as DEGs.

2 RSE: Read-Size Effect

In many cases, the number of reads are depend on study. For example, here we prepared multiple RNA-Seq count data designed as Breast Cancer cell lines vs Normal cells measured in 4 different studies (this data is also accessible by \texttt{data(BreastCancer)}).

<table>
<thead>
<tr>
<th>ID in this vignette</th>
<th>Accession (SRA / ERA Accession)</th>
<th>Experimental Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>StudyA</td>
<td>SRP008746</td>
<td>Breast Cancer (n=3) vs Normal (n=2)</td>
</tr>
<tr>
<td>StudyB</td>
<td>SRP006726</td>
<td>Breast Cancer (n=1) vs Normal (n=1)</td>
</tr>
<tr>
<td>StudyC</td>
<td>SRP005601</td>
<td>Breast Cancer (n=7) vs Normal (n=1)</td>
</tr>
<tr>
<td>StudyD</td>
<td>ERP000992</td>
<td>Breast Cancer (n=2) vs Normal (n=1)</td>
</tr>
</tbody>
</table>

Figure 1: Difference of the number of reads
As shown in the figure 1, the number of reads in StudyA, B, C, and D are relatively different. Generally, statistical test is influenced by the number of reads; the more the number of reads is large, the more the statistical tests are tend to be significant (see the next section). Therefore, in meta-analysis of RNA-seq data, data may be suffered from this bias. Here we call this bias as RSE (Read Size Effect).

3 Robustness against RSE

In the point of view of robustness against RSE, we evaluated five widely used method in RNA-seq; DESeq [2], edgeR [3], baySeq [4], and NOISeq [1]. Here we used only StudyA data. All counts in the matrix are repeatedly down-sampled in accordance with distributions of binomial (the probability equals 0.5). 1 (original), 1/2, 1/4, 1/8, 1/16, and 1/32-fold data are prepared as low read size situation. In each read size, four methods are conducted (figure 2.A, this data is also accessible by data(StudyA) and data(pvals)), then we focussed on how top500 genes of original data in order of significance will change its members, influenced by low read size (figure 2.B).

![Figure 2: A(left): RSE in each RNA-Seq method, B(right): Top 500 genes in order of significance](image)

Ideal method will returns same result regardless of read size, because same data was used. As shown in figure 2, NOISeq is not almost affected by the number of reads and robustly detects same genes as DEGs. Therefore, we concluded that NOISeq is suitable method at least in the point of view of meta-analysis. Note that probability of NOISeq is not equal to p-value; it is the probability that a gene is differentially expressed [1]. Our package integrates its probability by Fisher’s method [5] or Stouffer’s method (inverse normal method) [6]. In regard to Stouffer’s method, weighting by the number of replicates (sample size) is used.
4 Getting started

At first, install and load the `metaSeq` and `snow`.

```r
> library("metaSeq")
> library("snow")
```

The RNA-seq expression data in breast cancer cell lines and normal cells is prepared.
The data is measured from 4 different studies. The data is stored as a matrix (23368 rows × 18 columns).

```r
> data(BreastCancer)
```

We need to prepare two vectors. First vector is for indicating the experimental condition (e.g., 1: Cancer, 2: Normal) and second one is for indicating the source of data (e.g., A: StudyA, B: StudyB, C: StudyC, D: StudyD).

```r
> flag1 <- c(1,1,1,0, 0, 1,0, 1,1,1,1,1,1,0, 1,1,0)
```

Then, we use `meta.readData` to create R object for `meta.oneside.noiseq`.

```r
> cds <- meta.readData(data = BreastCancer, factor = flag1, studies = flag2)
```

Onesided-NOISeq is performed in each studies and each probabilities are summarized as a member of list object.

```r
> ## This is very time consuming step.
> # cl <- makeCluster(4, "SOCK")
> # result <- meta.oneside.noiseq(cds, k = 0.5, norm = "tmm", replicates = "biological",
> # factor = flag1, conditions = c(1, 0), studies = flag2, cl = cl)
> # stopCluster(cl)
> #
> ## Please load pre-calculated result (Result.Meta)
> ## by data function instead of scripts above.
> data(Result.Meta)
> result <- Result.Meta
```

Fisher’s method and Stouffer’s method can be applied to the result of `meta.oneside.noiseq`.

```r
> F <- Fisher.test(result)
> S <- Stouffer.test(result)
```

These outputs are summarized as list whose length is 3. First member is the probability which means a gene is upper-regulated genes, and Second member is lower-regulated genes. Weight in each study is also saved as its third member (weight is used only by Stouffer’s method).
> head(F$Upper)

<table>
<thead>
<tr>
<th></th>
<th>A1BG</th>
<th>A1BG-AS1</th>
<th>A1CF</th>
<th>A2LD1</th>
<th>A2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-SBSRNA4</td>
<td>0.3842542</td>
<td>0.5316118</td>
<td>0.5325544</td>
<td>NA</td>
<td>0.1358559</td>
</tr>
<tr>
<td></td>
<td>0.2252807</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> head(F$Lower)

<table>
<thead>
<tr>
<th></th>
<th>A1BG</th>
<th>A1BG-AS1</th>
<th>A1CF</th>
<th>A2LD1</th>
<th>A2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-SBSRNA4</td>
<td>0.8420357</td>
<td>0.6078896</td>
<td>0.4047202</td>
<td>NA</td>
<td>0.3661371</td>
</tr>
<tr>
<td></td>
<td>0.6197968</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> F$Weight

Study 1  Study 2  Study 3  Study 4
5     2     8     3

> head(S$Upper)

<table>
<thead>
<tr>
<th></th>
<th>A1BG</th>
<th>A1BG-AS1</th>
<th>A1CF</th>
<th>A2LD1</th>
<th>A2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-SBSRNA4</td>
<td>0.3709297</td>
<td>0.2663748</td>
<td>0.2711745</td>
<td>NA</td>
<td>0.2957139</td>
</tr>
<tr>
<td></td>
<td>0.2996707</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> head(S$Lower)

<table>
<thead>
<tr>
<th></th>
<th>A1BG</th>
<th>A1BG-AS1</th>
<th>A1CF</th>
<th>A2LD1</th>
<th>A2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-SBSRNA4</td>
<td>0.6290703</td>
<td>0.7336252</td>
<td>0.7288255</td>
<td>NA</td>
<td>0.7042861</td>
</tr>
<tr>
<td></td>
<td>0.7003293</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

> S$Weight

Study 1  Study 2  Study 3  Study 4
5     2     8     3
Generally, by meta-analysis, detection power will improved and much genes are detected as DEGs.

<table>
<thead>
<tr>
<th>Method</th>
<th>Study</th>
<th>Number of DEGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOISeq</td>
<td>A</td>
<td>86</td>
</tr>
<tr>
<td>NOISeq</td>
<td>B</td>
<td>563</td>
</tr>
<tr>
<td>NOISeq</td>
<td>C</td>
<td>99</td>
</tr>
<tr>
<td>NOISeq</td>
<td>D</td>
<td>210</td>
</tr>
<tr>
<td>NOISeq</td>
<td>A, B, C, D (not meta-analysis)</td>
<td>21</td>
</tr>
<tr>
<td>metaSeq (Fisher, Upper)</td>
<td>A, B, C, D</td>
<td>407</td>
</tr>
<tr>
<td>metaSeq (Fisher, Lower)</td>
<td>A, B, C, D</td>
<td>1483</td>
</tr>
<tr>
<td>metaSeq (Stouffer, Upper)</td>
<td>A, B, C, D</td>
<td>116</td>
</tr>
<tr>
<td>metaSeq (Stouffer, Lower)</td>
<td>A, B, C, D</td>
<td>2271</td>
</tr>
</tbody>
</table>
5 Meta-analysis by non-NOISeq method

For some reason, we may want to use non-NOISeq method like DESeq, edgeR, or even cuffdiff [7]. We prepared other.oneside.noiseq as optional function for such methods. Returned object can be directly applied to Fisher.test and Stouffer.test.

We have to prepare at least 2 matrix filled with p-value or probability. First matrix is for upper-regulated genes between control group and treatment group. On the other hand, second matrix is for lower-regulated genes. As optional parameter, weight in each study is also available. Weight is need for Stouffer’s method but not necessary for Fisher’s method.

```r
## Assume this matrix as one-sided p-values
## generated by non-NOISeq method (e.g., cuffdiff)
upper <- matrix(runif(300), ncol=3, nrow=100)
lower <- 1 - upper
rownames(upper) <- paste0("Gene", 1:100)
rownames(lower) <- paste0("Gene", 1:100)
weight <- c(3,6,8)
```

Next, other.oneside.pvalues will return a list object for Fisher.test or Stouffer.test by upper, lower, and weight.

```r
## other.oneside.pvalues function return a matrix
## which can input Fisher.test or Stouffer.test
result <- other.oneside.pvalues(upper, lower, weight)
```

result above can be applied to Fisher.test and Stouffer.test.

```r
F <- Fisher.test(result)
str(F)
```

```
List of 3
$ Upper : Named num [1:100] 0.851 0.459 0.604 0.103 0.506 ... 
  ..- attr(*, "names")= chr [1:100] "Gene1" "Gene2" "Gene3" "Gene4" ... 
$ Lower : Named num [1:100] 0.389 0.44 0.61 0.959 0.743 ... 
  ..- attr(*, "names")= chr [1:100] "Gene1" "Gene2" "Gene3" "Gene4" ... 
$ Weight: Named num [1:3] 3 6 8 
  ..- attr(*, "names")= chr [1:3] "Exp 1" "Exp 2" "Exp 3"
```

```r
head(F$Upper)
```

```
Gene1  Gene2  Gene3  Gene4  Gene5  Gene6
0.85131166 0.45938728 0.60374919 0.10278915 0.50570896 0.04582429
```

```r
head(F$Lower)
```

```
Gene1  Gene2  Gene3  Gene4  Gene5  Gene6
0.3887235 0.4402691 0.6101499 0.9590429 0.7432250 0.9887810
```
> F$Weight

Exp 1 Exp 2 Exp 3
    3   6   8

> S <- Stouffer.test(result)
> str(S)

List of 3
$ Upper : Named num [1:100] 0.7623 0.2976 0.5302 0.0769 0.3511 ...  
  ..- attr(*, "names")= chr [1:100] "Gene1" "Gene2" "Gene3" "Gene4" ...  
$ Lower : Named num [1:100] 0.2380 0.7024 0.4700 0.9231 0.6490 ...  
  ..- attr(*, "names")= chr [1:100] "Gene1" "Gene2" "Gene3" "Gene4" ...  
$ Weight: Named num [1:3] 3 6 8  
  ..- attr(*, "names")= chr [1:3] "Exp 1" "Exp 2" "Exp 3"

> head(S$Upper)

     Gene1     Gene2     Gene3     Gene4     Gene5     Gene6
0.76227804 0.29755779 0.53015586 0.07687506 0.35110004 0.03591811

> head(S$Lower)

     Gene1     Gene2     Gene3     Gene4     Gene5     Gene6
0.2377220 0.7024422 0.4698441 0.9231249 0.6489000 0.9640819

> S$Weight

Exp 1 Exp 2 Exp 3
    3   6   8
6 Setup

This vignette was built on:

> sessionInfo()

R version 3.4.0 (2017-04-21)
Platform: x86_64-pc-linux-gnu (64-bit)
Running under: Ubuntu 16.04.2 LTS

Matrix products: default
BLAS: /home/biocbuild/bbs-3.5-bioc/R/lib/libRblas.so
LAPACK: /home/biocbuild/bbs-3.5-bioc/R/lib/libRlapack.so

locale:
[1] LC_CTYPE=en_US.UTF-8   LC_NUMERIC=C
[3] LC_TIME=en_US.UTF-8    LC_COLLATE=C
[5] LC_MONETARY=en_US.UTF-8 LC_MESSAGES=en_US.UTF-8
[7] LC_PAPER=en_US.UTF-8   LC_NAME=C
[9] LC_ADDRESS=C           LC_TELEPHONE=C

attached base packages:
[1] splines parallel stats graphics grDevices utils
[7] datasets methods base

other attached packages:
[1] metaSeq_1.16.0 Rcpp_0.12.10 snow_0.4-2
[4] NOISeq_2.20.0 Matrix_1.2-9 Biobase_2.36.0
[7] BiocGenerics_0.22.0

loaded via a namespace (and not attached):
[1] compiler_3.4.0 tools_3.4.0 grid_3.4.0 lattice_0.20-35
References


