An Introduction to *QNB*

Lian Liu <liulian19860905@163.com>

Modified: 6 Jul, 2017. Compiled: July 19, 2017

1 Introduction

As a newly emerged research area, RNA epigenetics has drawn increasing attention recently for the participation of RNA methylation and other modifications in a number of crucial biological processes. Thanks to high throughput sequencing techniques, such as m6A-Seq, transcriptome-wide RNA methylation profile is now available in the form of count-based data, with which it is often of interests to study the dynamics in epitranscriptomic layer. However, the sample size of RNA methylation experiment is usually very small due to its costs; and additionally, there usually exist a large number of genes whose methylation level cannot be accurately estimated due to their low expression level, making differential RNA methylation analysis a difficult task. We present *QNB* R-package, a statistical approach for differential RNA methylation analysis with count-based small-sample sequencing data. The method is based on 4 independent negative binomial dis-tributions with their variances and means linked by local regressions. QNB showed improved performance on simulated and real m6A-Seq datasets when compared with competing algorithms. And the QNB model is also applicable to other datasets related RNA modifications, including but not limited to RNA bisulfite sequencing, m1A-Seq, Par-CLIP, RIP-Seq, etc. Please don't hesitate to contact <liulian19860905@163.com> if you have any problem. The inputs of the main function `qnbtest` are four reads count data frame for IP samples of two conditions and Input samples of two conditions.The *QNB* package fullfills the following one key function:

- differential RNA methylation analysis for count-based small-sample sequencing data with a quad-negative binomial model

We will in the next see how the the main functions can be accomplished in a single command.

2 Input data

As input, the *QNB* package expects count data from two conditions (e. g., treated and control) as obtained, e. g., from MeRIP-Seq, in the form of two rectangular tables of integer values for each condition, one is Input control sample and

another is IP sample. The table cell in the i-th row and the j-th column of the table shows the reads count of the methylation site i in sample j.

The count values must be raw counts of sequencing reads. So, please do not supply other quantities, such as (rounded) normalized counts – this will lead to nonsensical results.

In this vignette, we will work with DAA dataset. The original DAA raw data in SRA format was obtained directly GEO (GSE48037), which consists of 3 IP and 3 Input MeRIP-Seq replicates obtained under wild type condition and after DAA treatment, respectively (a total of 12 libraries). The short sequencing reads are firstly aligned to human genome assembly hg19 with Tophat2 [1], and then get RNA N6-methyl-adenosine (m6A) sites using *exomePeak* R/Bioconductor package [2] with UCSC gene annotation database [3]. In the peak calling step, to obtain a consensus RNA methylation site set between two experimental conditions (wild type and DAA treatment), we merged 6 IP samples and 6 Input samples, respectively. Then we used Bioconductor packages on R platform to obtain the reads count matrix. In the matrix, it includes the reads counts of m6A methylation sites from IP and Input samples (each with 3 replicates) under two conditions.

3 Differential RNA Methylation Analysis

The main function of *QNB* R-package is to analyse differential RNA methylation. Meths are the reads count matrix of IP samples from two conditions, and unmeths are Input control samples from two condition. To get the differential RNA methylation, we estimate the dispersion for each site between treated (including IP and Input control sample) and untreated (including IP and Input control sample). In addtion, IP and Input control samples must be the same replicates, but it is may be the different replicates under two conditons.

To estimate the dispersion, there are four ways how the empirical dispersion can be computed:

- pooled - Use the samples from all conditions with replicates to estimate a single pooled empirical dispersion value, called `"pooled"`, and assign it to all samples.
- per-condition - For each condition with replicates, compute an empirical dispersion value by considering the data from samples for this condition.
- blind - Ignore the sample labels and compute an empirical dispersion value as if all samples were replicates of a single condition. This can be done even if there are no biological replicates.
- auto - Select mode according to the size of samples automatically. The default is auto. By default, QNB package implements the `"per-condition"` mode for a more sensitive estimation of the raw variance parameter when biological replicates are provided; while the `"blind"` mode is implemented when biological replicates are not available.

Other parameters:

- size.factor - A list of size factor. The size factor of the IP and input sample of the biological replicate and directly reflect their sequencing depth. If `size.factor=NA`, *QNB* will compute the size factor of IP and input sample of each replicate. If user could provide the size factor, the names of each term must be `control_ip, treated_ip, control_input, treated_input` in list.
- plot.dispersion - The default is TRUE. If `plot.dispersion=FALSE`, it will not save the dispersion figure.
- output.dir - The saved file path. The default is NA. If `output.dir=NA`, the path is the current path.

Let us firstly load the package and get the toy data (came with the package) ready.

```
> library(QNB)
> f1 = system.file("extdata", "control_ip.txt", package="QNB")
> f2 = system.file("extdata", "treated_ip.txt", package="QNB")
> f3 = system.file("extdata", "control_input.txt", package="QNB")
> f4 = system.file("extdata", "treated_input.txt", package="QNB")
> meth1 = read.table(f1, header=TRUE)
> meth2 = read.table(f2, header=TRUE)
> unmeth1 = read.table(f3, header=TRUE)
> unmeth2 = read.table(f4, header=TRUE)
> head(meth1)

  S1 S2 S3
1  7  9  5
2  1  6  3
3  2  0  0
4  3  6  5
5  7  1  4
6  0  0  0

> head(unmeth1)

  S1 S2 S3
1  8  2  1
2  0  5  0
3  0  0  1
4  5  2  5
5  1  2  1
6  0  1  0
```

3.1 Standard comparison between two experimental conditions

When there are replicates under two conditions, we could select `mode="per-condition"` or `mode="pooled"` to estimate the dispersion. The default is `auto`. By default, QNB package implements the "per-conditon" mode for a more sensitive estimation of the raw variance parameter when biological replicates are provided; while the "blind" mode is implemented when biological replicates are not available. If `mode="per-condition"`, we estimate one dispersion for each condition, respectively. If `mode="pooled"`, we combind all replicates to generate one dataset from control samples and IP samples, then estimate one dispersion for two conditions.

```
> result = qnbtest(meth1, meth2, unmeth1, unmeth2, mode="per-condition")

[1] "Estimating dispersion for each RNA methylation site, this will take a while ..."

> head(result)

  p.treated p.control    log2.RR    log2.OR     pvalue
1 0.5611534 0.5149039  0.1240921  0.2686460 0.78490363
2 0.7143968 0.4998503  0.5152295  1.3235773 0.36272519
3 0.7024927 0.5059688  0.4734347  1.2051118 0.79032881
4 0.6195112 0.3934275  0.6550325  1.3278586 0.06613825
5 0.4894613 0.6139670 -0.3269663 -0.7302625 0.57085286
6 0.3922649 0.0000000         Inf         Inf 0.40214557
           q      padj
1 12.2742967 0.9727062
2  8.0316299 0.9727062
3  3.1432272 0.9727062
4 19.9338001 0.8216957
5  8.7317667 0.9727062
6  0.5874658 0.9727062
```

The results will be saved in the specified output directory, including the dispersion figure(if `plot.dispersion=TRUE`) and the result table (including 7 columns (p.treated, p.control, log2.RR, log2.OR, pvalue, q, padj)).

- p.treated - The percentage of methylation under treated condition.
- p.control - The percentage of methylation under control condition.
- log2.RR - The normalized risk ratio.
- log2.OR - The normalized odds ratio.
- pvalue - Indicate the significance of the methylation site as an RNA differential methylation site

- q - The standardized feature abundance, which is proportional to the expression level of the RNA transcript.
- padj - The FDR of the methylation site, indicating the significance of the peak as an RNA differential methylation site after multiple hypothesis correction usting BH method.

In QNB, we compute size factor of IP and input samples using the "geometric" approach developed for RNA-Seq data. If `size.factor=NA`, *QNB* will compute size factor according to the samples which are provided, otherwise, user could provide the size factor according to their request. The format of size factor must be a list, and the name of each term must be `control_ip, treated_ip, control_input, treated_input` in list.

```
> total_number_reads_control_ip <- c(3015921,2563976,198530)
> total_number_reads_treated_ip <- c(1565101,152389,323569)
> total_number_reads_control_input <- c(108561,302534,108123)
> total_number_reads_treated_input <- c(301270,208549,308654)
> standard_library_size <- exp(mean(log( c(total_number_reads_control_ip,
+                                    total_number_reads_treated_ip,
+                                    total_number_reads_control_input,
+                                    total_number_reads_treated_input))))
> size.factor <- list(control_ip = total_number_reads_control_ip/standard_library_size,
+                     treated_ip = total_number_reads_treated_ip/standard_library_size,
+                     control_input =
+                       total_number_reads_control_input/standard_library_size,
+                     treated_input =
+                       total_number_reads_treated_input/standard_library_size)
> result <- qnbtest(meth1, meth2, unmeth1, unmeth2,
+                   size.factor = size.factor)

[1] "Estimating dispersion for each RNA methylation site, this will take a while ..."

>
```

3.2 Comparison without replicates

Proper replicates are essential to interpret a biological experiment. After all, any attempt to work without replicates will lead to conclusions of very limited reliability. But the *QNB* package can deal with them.

If you have replicates for one condition but not for the other, or without any replicates for two conditions, you can select `mode="blind"` to estimate the dispersion. We combind all samples under two conditions to generate replicates for two conditions. Then we estimate one dispersion for two conditions.

```
> f1 = system.file("extdata", "no_rep_controlip.txt", package="QNB")
> f2 = system.file("extdata", "no_rep_treatedip.txt", package="QNB")
```

```
> f3 = system.file("extdata", "no_rep_controlinput.txt", package="QNB")
> f4 = system.file("extdata", "no_rep_treatedinput.txt", package="QNB")
> no_rep_meth1 = read.table(f1, header=TRUE)
> no_rep_meth2 = read.table(f2, header=TRUE)
> no_rep_unmeth1 = read.table(f3, header=TRUE)
> no_rep_unmeth2 = read.table(f4, header=TRUE)
> head(no_rep_meth1)

  x
1 7
2 1
3 2
4 3
5 7
6 0

> head(no_rep_unmeth1)

  x
1 8
2 0
3 0
4 5
5 1
6 0

> result = qnbtest(no_rep_meth1,
+                  no_rep_meth2,
+                  no_rep_unmeth1,
+                  no_rep_unmeth2,
+                  mode="blind")

[1] "Estimating dispersion for each RNA methylation site, this will take a while ..."
```

3.3 Select mode automatically

If you could not decide which mode to estimate dispersion, `mode="auto"` will select suitable way to estimate dispersion according to the replicates. By default, QNB package implements the "per-conditon" mode for a more sensitive estimation of the raw variance parameter when biological replicates are provided; while the "blind" mode is implemented when biological replicates are not available.

```
> result = qnbtest(meth1, meth2, unmeth1, unmeth2)

[1] "Estimating dispersion for each RNA methylation site, this will take a while ..."
```

3.4 The complete processing flow

The following is the complete prossing, including peak calling using exomePeak R/Bioconductor package , get reads count and differential RNA methylation analysis.

```
> library(exomePeak)
> library(QNB)
> library(GenomicFeatures)
> library(Rsamtools)
> #peak calling using exomePeak
> GENE_ANNO_GTF = system.file("extdata", "example.gtf", package="exomePeak")
> f1 = system.file("extdata", "IP1.bam", package="exomePeak")
> f2 = system.file("extdata", "IP2.bam", package="exomePeak")
> f3 = system.file("extdata", "treated_IP1.bam", package="exomePeak")
> IP_BAM = c(f1,f2,f3)
> f4 = system.file("extdata", "Input1.bam", package="exomePeak")
> f5 = system.file("extdata", "Input2.bam", package="exomePeak")
> f6 = system.file("extdata", "treated_Input1.bam", package="exomePeak")
> INPUT_BAM = c(f4,f5,f6)
> res = exomepeak(GENE_ANNO_GTF=GENE_ANNO_GTF, IP_BAM=IP_BAM, INPUT_BAM=INPUT_BAM)

[1] "Divide transcriptome into chr-gene-batch sections ..."
[1] "Get Reads Count ..."
[1] "This step may take a few hours ..."
[1] "100 %"
[1] "Get all the peaks ..."
[1] "Get the consistent peaks ..."
[1] "--------------------------------"
[1] "The bam files used:"
[1] "3 IP replicate(s)"
[1] "3 Input replicate(s)"
[1] "--------------------------------"
[1] "Peak calling result: "
[1] "16 peaks detected on merged data."
[1] "Please check 'peak.bed/xls' under C:/Users/S41-70/AppData/Local/Temp/RtmpOmbUj4/Rbuilde
[1] "13 consistent peaks detected on every replicates. (Recommended list)"
[1] "Please check 'con_peak.bed/xls' under C:/Users/S41-70/AppData/Local/Temp/RtmpOmbUj4/Rbu

> #get reads count
> peak=res$all_peaks
> untreated_ip=matrix(0,nrow=length(peak),ncol=2)
> untreated_input=matrix(0,nrow=length(peak),ncol=2)
> treated_ip=matrix(0,nrow=length(peak),ncol=1)
> treated_input=matrix(0,nrow=length(peak),ncol=1)
> txdb <- makeTxDbFromUCSC(genome="hg19")
> exonRanges <- exonsBy(txdb, "tx")
```

```
> #get ip reads count
> #f1
> aligns <- readGAlignments(f1)
> para <- ScanBamParam(what="mapq")
> mapq <- scanBam(f1, param=para)[[1]][[1]]
> # filter reads with mapq smaller than 30.
> mapq[is.na(mapq)] <- 255  # Note: mapq "NA" means mapq = 255
> ID_keep <- (mapq >30)
> filtered <- aligns[ID_keep]
> id <- countOverlaps(filtered,exonRanges)
> transcriptome_filtered_aligns <- filtered[id>0]
> counts <- countOverlaps(peak, transcriptome_filtered_aligns)
> #counts <- countOverlaps(peak, filtered)
> untreated_ip[,1] <- counts
> #f2
> aligns <- readGAlignments(f2)
> para <- ScanBamParam(what="mapq")
> mapq <- scanBam(f2, param=para)[[1]][[1]]
> # filter reads with mapq smaller than 30.
> mapq[is.na(mapq)] <- 255  # Note: mapq "NA" means mapq = 255
> ID_keep <- (mapq >30)
> filtered <- aligns[ID_keep]
> id <- countOverlaps(filtered,exonRanges)
> transcriptome_filtered_aligns <- filtered[id>0]
> counts <- countOverlaps(peak, transcriptome_filtered_aligns)
> #counts <- countOverlaps(peak, filtered)
> untreated_ip[,2] <- counts
> #f3
> aligns <- readGAlignments(f3)
> para <- ScanBamParam(what="mapq")
> mapq <- scanBam(f3, param=para)[[1]][[1]]
> # filter reads with mapq smaller than 30.
> mapq[is.na(mapq)] <- 255  # Note: mapq "NA" means mapq = 255
> ID_keep <- (mapq >30)
> filtered <- aligns[ID_keep]
> id <- countOverlaps(filtered,exonRanges)
> transcriptome_filtered_aligns <- filtered[id>0]
> counts <- countOverlaps(peak, transcriptome_filtered_aligns)
> #counts <- countOverlaps(peak, filtered)
> treated_ip[,1] <- counts
> #f4
> aligns <- readGAlignments(f4)
> para <- ScanBamParam(what="mapq")
> mapq <- scanBam(f4, param=para)[[1]][[1]]
> # filter reads with mapq smaller than 30.
> mapq[is.na(mapq)] <- 255  # Note: mapq "NA" means mapq = 255
```

```
> ID_keep <- (mapq >30)
> filtered <- aligns[ID_keep]
> id <- countOverlaps(filtered,exonRanges)
> transcriptome_filtered_aligns <- filtered[id>0]
> counts <- countOverlaps(peak, transcriptome_filtered_aligns)
> #counts <- countOverlaps(peak, filtered)
> untreated_input[,1] <- counts
> #get input reads count
> #f5
> aligns <- readGAlignments(f5)
> para <- ScanBamParam(what="mapq")
> mapq <- scanBam(f5, param=para)[[1]][[1]]
> # filter reads with mapq smaller than 30.
> mapq[is.na(mapq)] <- 255  # Note: mapq "NA" means mapq = 255
> ID_keep <- (mapq >30)
> filtered <- aligns[ID_keep]
> id <- countOverlaps(filtered,exonRanges)
> transcriptome_filtered_aligns <- filtered[id>0]
> counts <- countOverlaps(peak, transcriptome_filtered_aligns)
> #counts <- countOverlaps(peak, filtered)
> untreated_input[,2] <- counts
> #f6
> aligns <- readGAlignments(f6)
> para <- ScanBamParam(what="mapq")
> mapq <- scanBam(f6, param=para)[[1]][[1]]
> # filter reads with mapq smaller than 30.
> mapq[is.na(mapq)] <- 255  # Note: mapq "NA" means mapq = 255
> ID_keep <- (mapq >30)
> filtered <- aligns[ID_keep]
> id <- countOverlaps(filtered,exonRanges)
> transcriptome_filtered_aligns <- filtered[id>0]
> counts <- countOverlaps(peak, transcriptome_filtered_aligns)
> #counts <- countOverlaps(peak, filtered)
> treated_input[,1] <- counts
> #differential RNA methylation analysis
> result = qnbtest(untreated_ip,treated_ip,untreated_input,treated_input)

[1] "Estimating dispersion for each RNA methylation site, this will take a while ..."
```

4 Session Information

```
> sessionInfo()

R version 3.4.0 (2017-04-21)
Platform: x86_64-w64-mingw32/x64 (64-bit)
Running under: Windows 8.1 x64 (build 9600)
```

```
Matrix products: default

locale:
[1] LC_COLLATE=C
[2] LC_CTYPE=Chinese (Simplified)_China.936
[3] LC_MONETARY=Chinese (Simplified)_China.936
[4] LC_NUMERIC=C
[5] LC_TIME=Chinese (Simplified)_China.936

attached base packages:
[1] stats4    parallel  stats     graphics  grDevices
[6] utils     datasets  methods   base

other attached packages:
 [1] QNB_1.1.9                   exomePeak_2.10.0
 [3] GenomicAlignments_1.12.1    SummarizedExperiment_1.6.3
 [5] DelayedArray_0.2.7          matrixStats_0.52.2
 [7] rtracklayer_1.36.3          Rsamtools_1.28.0
 [9] Biostrings_2.44.1           XVector_0.16.0
[11] GenomicFeatures_1.28.3      AnnotationDbi_1.38.1
[13] Biobase_2.36.2              GenomicRanges_1.28.3
[15] GenomeInfoDb_1.12.2         IRanges_2.10.2
[17] S4Vectors_0.14.3            BiocGenerics_0.22.0
[19] locfit_1.5-9.1

loaded via a namespace (and not attached):
 [1] Rcpp_0.12.11             compiler_3.4.0
 [3] bitops_1.0-6             tools_3.4.0
 [5] zlibbioc_1.22.0          biomaRt_2.32.1
 [7] digest_0.6.12            bit_1.1-12
 [9] RSQLite_2.0              memoise_1.1.0
[11] tibble_1.3.3             lattice_0.20-35
[13] pkgconfig_2.0.1          rlang_0.1.1
[15] Matrix_1.2-10            DBI_0.7
[17] GenomeInfoDbData_0.99.0 bit64_0.9-7
[19] grid_3.4.0               XML_3.98-1.9
[21] BiocParallel_1.10.1      blob_1.1.0
[23] RCurl_1.95-4.8
```

References

[1] D. Kim, G. Pertea, C. Trapnell, H. Pimentel, R. Kelley, and S. L. Salzberg. Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology*, 14(4):295–311, 2013.

[2] J. Meng, X. Cui, M. K. Rao, Y. Chen, and Y. Huang. Exome-based analysis for rna epigenome sequencing data. *Bioinformatics*, 29(12):1565–7, 2013.

[3] D. Karolchik, G. P. Barber, J. Casper, H. Clawson, M. S. Cline, M. Diekhans, T. R. Dreszer, P. A. Fujita, L. Guruvadoo, M. Haeussler, R. A. Harte, S. Heitner, A. S. Hinrichs, K. Learned, B. T. Lee, C. H. Li, B. J. Raney, B. Rhead, K. R. Rosenbloom, C. A. Sloan, M. L. Speir, A. S. Zweig, D. Haussler, R. M. Kuhn, and W. J. Kent. The ucsc genome browser database: 2014 update. *Nucleic Acids Research*, 42(D1):D764–D770, 2014.