

Introduction to RBM package

Dongmei Li

October 30, 2018

Clinical and Translational Science Institute, University of Rochester School of Medicine and
Dentistry, Rochester, NY 14642-0708

Contents

1 Overview	1
2 Getting started	2
3 RBM_T and RBM_F functions	2
4 Ovarian cancer methylation example using the RBM_T function	6

1 Overview

This document provides an introduction to the RBM package. The RBM package executes the resampling-based empirical Bayes approach using either permutation or bootstrap tests based on moderated t-statistics through the following steps.

- Firstly, the RBM package computes the moderated t-statistics based on the observed data set for each feature using the `lmFit` and `eBayes` function.
- Secondly, the original data are permuted or bootstrapped in a way that matches the null hypothesis to generate permuted or bootstrapped resamples, and the reference distribution is constructed using the resampled moderated t-statistics calculated from permutation or bootstrap resamples.
- Finally, the p-values from permutation or bootstrap tests are calculated based on the proportion of the permuted or bootstrapped moderated t-statistics that are as extreme as, or more extreme than, the observed moderated t-statistics.

Additional detailed information regarding resampling-based empirical Bayes approach can be found elsewhere (Li et al., 2013).

2 Getting started

The RBM package can be installed and loaded through the following R code.
Install the RBM package with:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+   install.packages("BiocManager")
> BiocManager::install("RBM")
```

Load the RBM package with:

```
> library(RBM)
```

3 RBM_T and RBM_F functions

There are two functions in the RBM package: `RBM_T` and `RBM_F`. Both functions require input data in the matrix format with rows denoting features and columns denoting samples. `RBM_T` is used for two-group comparisons such as study designs with a treatment group and a control group. `RBM_F` can be used for more complex study designs such as more than two groups or time-course studies. Both functions need a vector for group notation, i.e., "1" denotes the treatment group and "0" denotes the control group. For the `RBM_F` function, a contrast vector need to be provided by users to perform pairwise comparisons between groups. For example, if the design has three groups (0, 1, 2), the `aContrast` parameter will be a vector such as ("X1-X0", "X2-X1", "X2-X0") to denote all pairwise comparisons. Users just need to add an extra "X" before the group labels to do the contrasts.

- Examples using the `RBM_T` function: `normdata` simulates a standardized gene expression data and `unifdata` simulates a methylation microarray data. The p -values from the `RBM_T` function could be further adjusted using the `p.adjust` function in the `stats` package through the Benjamini-Hochberg method.

```
> library(RBM)
> normdata <- matrix(rnorm(1000*6, 0, 1),1000,6)
> mydesign <- c(0,0,0,1,1,1)
> myresult <- RBM_T(normdata,mydesign,100,0.05)
> summary(myresult)
```

	Length	Class	Mode
ordfit_t	1000	-none-	numeric
ordfit_pvalue	1000	-none-	numeric
ordfit_beta0	1000	-none-	numeric
ordfit_beta1	1000	-none-	numeric
permutation_p	1000	-none-	numeric
bootstrap_p	1000	-none-	numeric

```
> sum(myresult$permutation_p<=0.05)
```

```

[1] 31

> which(myresult$permutation_p<=0.05)

[1] 26 32 38 44 80 127 152 165 204 236 280 290 415 453 483 514 550 597 620
[20] 633 637 642 646 677 689 705 787 798 839 893 922

> sum(myresult$bootstrap_p<=0.05)

[1] 11

> which(myresult$bootstrap_p<=0.05)

[1] 31 169 304 319 390 435 502 677 689 705 711

> permutation_adjp <- p.adjust(myresult$permutation_p, "BH")
> sum(permutation_adjp<=0.05)

[1] 2

> bootstrap_adjp <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adjp<=0.05)

[1] 0

> unifdata <- matrix(runif(1000*7,0.10, 0.95), 1000, 7)
> mydesign2 <- c(0,0,0, 1,1,1,1)
> myresult2 <- RBM_T(unifdata,mydesign2,100,0.05)
> sum(myresult2$permutatioin_p<=0.05)

[1] 0

> sum(myresult2$bootstrap_p<=0.05)

[1] 24

> which(myresult2$bootstrap_p<=0.05)

[1] 27 36 121 135 140 175 209 219 258 316 354 393 469 471 525 573 622 662 743
[20] 809 817 859 926 931

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=0.05)

[1] 0

```

- Examples using the RBM_F function: normdata_F simulates a standardized gene expression data and unifdata_F simulates a methylation microarray data. In both examples, we were interested in pairwise comparisons.

```

> normdata_F <- matrix(rnorm(1000*9,0,2), 1000, 9)
> mydesign_F <- c(0, 0, 0, 1, 1, 1, 2, 2, 2)
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult_F <- RBM_F(normdata_F, mydesign_F, aContrast, 100, 0.05)
> summary(myresult_F)

              Length Class  Mode
ordfit_t      3000   -none-  numeric
ordfit_pvalue 3000   -none-  numeric
ordfit_beta1   3000   -none-  numeric
permutation_p 3000   -none-  numeric
bootstrap_p    3000   -none-  numeric

> sum(myresult_F$permutation_p[, 1]<=0.05)

[1] 62

> sum(myresult_F$permutation_p[, 2]<=0.05)

[1] 60

> sum(myresult_F$permutation_p[, 3]<=0.05)

[1] 63

> which(myresult_F$permutation_p[, 1]<=0.05)

[1] 34 37 115 121 124 126 134 139 146 199 221 236 259 284 287 288 299 302 304
[20] 308 312 323 325 346 349 362 368 406 413 427 448 452 481 484 485 496 497 534
[39] 561 588 600 609 660 673 680 714 726 731 735 750 784 787 826 870 876 880 884
[58] 929 950 959 987 995

> which(myresult_F$permutation_p[, 2]<=0.05)

[1] 37 115 121 124 126 139 146 199 236 272 284 288 299 301 302 304 308 312 325
[20] 339 346 348 349 362 368 406 413 427 448 452 481 484 485 496 497 534 555 600
[39] 609 673 680 712 714 726 731 735 750 784 787 796 858 876 880 884 908 929 937
[58] 950 959 995

> which(myresult_F$permutation_p[, 3]<=0.05)

[1] 37 121 124 126 134 146 199 221 258 284 287 288 299 301 302 304 312 325 339
[20] 346 355 362 368 406 413 427 448 452 481 485 496 534 548 550 555 568 600 606
[39] 609 673 680 714 726 731 750 773 784 787 793 858 860 870 876 880 884 908 925
[58] 929 937 950 959 987 995

> con1_adjp <- p.adjust(myresult_F$permutation_p[, 1], "BH")
> sum(con1_adjp<=0.05/3)

```

```

[1] 7

> con2_adj_p <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adj_p<=0.05/3)

[1] 12

> con3_adj_p <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adj_p<=0.05/3)

[1] 13

> which(con2_adj_p<=0.05/3)

[1] 146 312 346 452 485 496 600 609 680 726 880 995

> which(con3_adj_p<=0.05/3)

[1] 121 199 312 406 427 481 485 496 609 680 714 880 995

> unifdata_F <- matrix(runif(1000*18, 0.15, 0.98), 1000, 18)
> mydesign2_F <- c(rep(0, 6), rep(1, 6), rep(2, 6))
> aContrast <- c("X1-X0", "X2-X1", "X2-X0")
> myresult2_F <- RBM_F(unifdata_F, mydesign2_F, aContrast, 100, 0.05)
> summary(myresult2_F)

      Length Class  Mode
ordfit_t      3000   -none- numeric
ordfit_pvalue 3000   -none- numeric
ordfit_beta1  3000   -none- numeric
permutation_p 3000   -none- numeric
bootstrap_p   3000   -none- numeric

> sum(myresult2_F$bootstrap_p[, 1]<=0.05)

[1] 61

> sum(myresult2_F$bootstrap_p[, 2]<=0.05)

[1] 64

> sum(myresult2_F$bootstrap_p[, 3]<=0.05)

[1] 64

> which(myresult2_F$bootstrap_p[, 1]<=0.05)

```

```

[1] 5 34 45 61 100 101 111 122 175 201 202 203 210 223 227 254 301 331 357
[20] 359 378 387 414 439 486 497 506 523 529 537 541 551 567 598 606 608 612 613
[39] 623 642 645 663 689 697 705 706 710 715 753 769 826 828 833 838 894 901 903
[58] 940 964 974 981

> which(myresult2_F$bootstrap_p[, 2]<=0.05)

[1] 5 7 34 45 61 100 101 111 122 175 186 201 202 203 210 223 227 254 278
[20] 301 360 378 387 407 414 425 439 452 486 487 488 506 529 538 541 567 595 598
[39] 606 623 642 645 663 676 677 689 697 705 706 715 769 813 826 828 833 838 894
[58] 901 903 940 964 966 974 981

> which(myresult2_F$bootstrap_p[, 3]<=0.05)

[1] 5 7 34 45 61 66 67 100 111 122 128 175 186 201 203 210 223 227 254
[20] 278 301 331 339 378 387 407 425 439 486 488 506 529 538 541 551 595 606 612
[39] 613 623 642 654 663 677 679 689 706 710 715 753 769 778 809 826 828 833 838
[58] 862 894 903 940 964 974 981

> con21_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adj_p<=0.05/3)

[1] 9

> con22_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adj_p<=0.05/3)

[1] 7

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=0.05/3)

[1] 7

```

4 Ovarian cancer methylation example using the RBM_T function

Two-group comparisons are the most common contrast in biological and biomedical field. The ovarian cancer methylation example is used to illustrate the application of RBM_T in identifying differentially methylated loci. The ovarian cancer methylation example is taken from the genome-wide DNA methylation profiling of United Kingdom Ovarian Cancer Population Study (UKOPS). This study used Illumina Infinium 27k Human DNA methylation Beadchip v1.2 to obtain DNA methylation profiles on over 27,000 CpGs in whole blood cells from 266 ovarian cancer women and 274 age-matched healthy controls. The data are downloaded from the NCBI GEO website with access number GSE19711. For illustration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovarian cancer cases (pre-treatment) and 4 healthy controls. The following codes show the process of generating significant differential DNA methylation loci using the RBM_T function and presenting the results for further validation and investigations.

```

> system.file("data", package = "RBM")

[1] "C:/Users/biocbuild/bbs-3.8-bioc/tmpdir/RtmpM17Ntn/Rinst26006ead39a2/RBM/data"

> data(ovarian_cancer_methylation)
> summary(ovarian_cancer_methylation)

      IlmnID      Beta      exmdata2[, 2]      exmdata3[, 2]
cg00000292: 1  Min.   :0.01058  Min.   :0.01187  Min.   :0.009103
cg00002426: 1  1st Qu.:0.04111  1st Qu.:0.04407  1st Qu.:0.041543
cg00003994: 1  Median :0.08284  Median :0.09531  Median :0.087042
cg00005847: 1  Mean    :0.27397  Mean    :0.28872  Mean    :0.283729
cg00006414: 1  3rd Qu.:0.52135  3rd Qu.:0.59032  3rd Qu.:0.558575
cg00007981: 1  Max.    :0.97069  Max.    :0.96937  Max.    :0.970155
(Other)    :994              NA's     :4
exmdata4[, 2]      exmdata5[, 2]      exmdata6[, 2]      exmdata7[, 2]
Min.   :0.01019  Min.   :0.01108  Min.   :0.01937  Min.   :0.01278
1st Qu.:0.04092  1st Qu.:0.04059  1st Qu.:0.05060  1st Qu.:0.04260
Median :0.09042  Median :0.08527  Median :0.09502  Median :0.09362
Mean    :0.28508  Mean    :0.28482  Mean    :0.27348  Mean    :0.27563
3rd Qu.:0.57502  3rd Qu.:0.57300  3rd Qu.:0.52099  3rd Qu.:0.52240
Max.    :0.96658  Max.    :0.97516  Max.    :0.96681  Max.    :0.95974
              NA's     :1
exmdata8[, 2]
Min.   :0.01357
1st Qu.:0.04387
Median :0.09282
Mean    :0.28679
3rd Qu.:0.57217
Max.    :0.96268

> ovarian_cancer_data <- ovarian_cancer_methylation[, -1]
> label <- c(1, 1, 0, 0, 1, 1, 0, 0)
> diff_results <- RBM_T(aData=ovarian_cancer_data, vec_trt=label, repetition=100, alpha=0.05)
> summary(diff_results)

      Length Class  Mode
ordfit_t      1000  -none- numeric
ordfit_pvalue 1000  -none- numeric
ordfit_beta0   1000  -none- numeric
ordfit_beta1   1000  -none- numeric
permutation_p 1000  -none- numeric
bootstrap_p    1000  -none- numeric

> sum(diff_results$ordfit_pvalue<=0.05)

[1] 45

```

```

> sum(diff_results$permutation_p<=0.05)

[1] 49

> sum(diff_results$bootstrap_p<=0.05)

[1] 44

> ordfit_adj_p <- p.adjust(diff_results$ordfit_pvalue, "BH")
> sum(ordfit_adj_p<=0.05)

[1] 0

> perm_adj_p <- p.adjust(diff_results$permutation_p, "BH")
> sum(perm_adj_p<=0.05)

[1] 0

> boot_adj_p <- p.adjust(diff_results$bootstrap_p, "BH")
> sum(boot_adj_p<=0.05)

[1] 0

> diff_list_perm <- which(perm_adj_p<=0.05)
> diff_list_boot <- which(boot_adj_p<=0.05)
> sig_results_perm <- cbind(ovarian_cancer_methylation[diff_list_perm, ], diff_results$ordfit_t)
> print(sig_results_perm)

[1] IlmnID
[2] Beta
[3] exmdata2[, 2]
[4] exmdata3[, 2]
[5] exmdata4[, 2]
[6] exmdata5[, 2]
[7] exmdata6[, 2]
[8] exmdata7[, 2]
[9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_perm]
[11] diff_results$permutation_p[diff_list_perm]
<0 rows> (or 0-length row.names)

> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t)
> print(sig_results_boot)

[1] IlmnID
[2] Beta
[3] exmdata2[, 2]
[4] exmdata3[, 2]

```

```
[5] exmdata4[, 2]
[6] exmdata5[, 2]
[7] exmdata6[, 2]
[8] exmdata7[, 2]
[9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_boot]
[11] diff_results$bootstrap_p[diff_list_boot]
<0 rows> (or 0-length row.names)
```