# Introduction to RBM package

Dongmei Li

October 26, 2021

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 Bejamini-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] 19

> which(myresult$permutation_p<=0.05)

 [1]  28  30 106 140 141 182 235 264 286 289 298 301 350 451 531 791 837 928 971

> sum(myresult$bootstrap_p<=0.05)

[1] 18

> which(myresult$bootstrap_p<=0.05)

 [1]  45 100 106 184 237 238 298 376 426 444 485 564 571 656 878 927 963 993

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

[1] 1

> 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] 23

> which(myresult2$bootstrap_p<=0.05)

 [1]   7 110 120 152 189 214 216 549 550 553 589 592 616 628 635 661 670 688 716
[20] 828 882 907 985

> 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] 48

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

[1] 50

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

[1] 42

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

 [1]   4  22  28  29  33  40 101 123 130 190 196 204 233 250 269 273 280 285 323
[20] 348 390 393 410 421 456 508 514 529 534 612 619 621 630 678 707 729 761 764
[39] 782 794 826 830 837 856 866 875 930 992

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

 [1]   4  22  29  33  40  53 101 103 123 130 135 190 196 220 233 250 269 273 280
[20] 293 323 369 390 393 394 410 421 508 514 529 612 619 630 646 707 729 761 764
[39] 769 782 794 826 830 836 837 856 866 870 875 992

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

 [1]  29  33  40 101 103 123 130 190 196 204 220 227 233 250 269 280 293 323 390
[20] 393 410 507 508 514 563 594 612 619 630 707 729 754 761 764 767 769 782 830
[39] 856 875 945 992

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

[1] 9
```

```
> con2_adjp <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adjp<=0.05/3)

[1] 5

> con3_adjp <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adjp<=0.05/3)

[1] 5

> which(con2_adjp<=0.05/3)

[1]  33 123 269 729 992

> which(con3_adjp<=0.05/3)

[1] 123 250 269 729 992

> 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] 37

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

[1] 53

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

[1] 47

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

 [1]  23  64  75 118 142 148 204 225 235 244 266 303 344 346 389 394 420 422 462
[20] 488 509 581 592 648 659 674 748 791 806 841 851 881 892 913 961 976 978
```

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

 [1]  23  57  64  75 118 142 148 181 199 203 204 225 235 244 266 276 296 303 311
[20] 334 337 344 346 367 368 389 408 420 422 462 488 509 553 581 592 648 674 704
[39] 748 749 791 798 814 851 853 877 881 892 913 962 976 978 994

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

 [1]   6  52  57  75  81 118 148 155 181 199 203 204 225 235 240 244 266 303 311
[20] 344 346 364 389 408 411 420 462 535 554 648 659 674 744 791 841 851 853 871
[39] 881 891 892 913 961 962 976 978 994

> con21_adjp <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adjp<=0.05/3)

[1] 3

> con22_adjp <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adjp<=0.05/3)

[1] 4

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)

[1] 5
```

## 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 gemone-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 illutration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovariance 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] "/tmp/Rtmp2ZxMWV/Rinst1b9ee2b9b5409/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] 71

> sum(diff_results$bootstrap_p<=0.05)
```

```
[1] 39

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

[1] 0

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

[1] 14

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

[1] 1

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

        IlmnID       Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
5   cg00006414 0.07635468    0.07442468    0.15698040    0.08676092
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
237 cg00215066 0.94926640    0.95311870    0.94634910    0.94561120
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
437 cg00424946 0.04122172    0.04325330    0.03339863    0.02876798
520 cg00502442 0.03163993    0.03581662    0.02785063    0.02549502
772 cg00743372 0.03922780    0.02919634    0.02187972    0.02568053
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
5      0.07982556    0.08111396    0.08271889    0.08045977
83     0.04965089    0.04833366    0.03466159    0.04390894
103    0.67349260    0.73510200    0.75715920    0.78981220
106    0.04693030    0.06837343    0.04534005    0.03709488
131    0.17352980    0.12564280    0.18193170    0.20847670
237    0.94837410    0.94665570    0.94089070    0.94600090
245    0.04208405    0.05284988    0.03775905    0.03955271
280    0.61920530    0.61925200    0.46753250    0.55632410
437    0.03353116    0.03719167    0.03096761    0.03234779
```

```
520    0.03111720    0.03189393    0.02415307    0.02941176
772    0.02796053    0.03512214    0.02575992    0.02093909
848    0.05230160    0.06119713    0.06542751    0.06240686
851    0.50820240    0.34657470    0.66276570    0.64634510
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_perm]
5                              -1.389459
83                              2.514109
103                            -2.268711
106                             3.100324
131                            -3.451679
237                             1.419654
245                             1.962457
280                             4.170347
437                             2.102892
520                             1.873471
772                             2.416991
848                            -2.314412
851                            -2.841244
979                            -4.750997
    diff_results$permutation_p[diff_list_perm]
5                                          0
83                                         0
103                                        0
106                                        0
131                                        0
237                                        0
245                                        0
280                                        0
437                                        0
520                                        0
772                                        0
848                                        0
851                                        0
979                                        0

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

        IlmnID      Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
979 cg00945507 0.1343225      0.238546     0.3474976     0.2890334
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
979     0.1184851     0.1665385     0.3071842     0.2662474
    diff_results$ordfit_t[diff_list_boot]
979                            -4.750997
    diff_results$bootstrap_p[diff_list_boot]
```

979 0