# 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

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] 42

> which(myresult$permutation_p<=0.05)

 [1]  12  47  83  98 112 113 179 206 307 322 354 359 372 386 398 401 491 499 503
[20] 531 540 566 595 603 618 633 644 665 676 703 708 713 751 768 848 853 860 885
[39] 887 912 923 979

> sum(myresult$bootstrap_p<=0.05)

[1] 8

> which(myresult$bootstrap_p<=0.05)

[1] 372 531 593 599 633 721 840 931

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

[1] 3

> 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] 9

> which(myresult2$bootstrap_p<=0.05)

[1] 139 149 204 459 514 560 578 612 853

> 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] 53

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

[1] 59

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

[1] 63

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

 [1]   1  54  56  72  96 105 146 155 196 207 255 258 270 289 292 294 352 354 358
[20] 365 373 387 434 447 457 468 482 484 497 512 555 569 636 688 717 720 747 759
[39] 771 774 781 820 821 830 831 836 883 909 925 926 928 951 984

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

 [1]  54  56 105 134 148 176 195 196 207 230 255 258 270 289 292 294 325 334 354
[20] 358 365 373 387 441 442 457 468 484 497 498 512 522 569 572 636 688 717 720
[39] 722 732 747 759 771 774 781 810 820 821 831 836 870 883 909 925 926 928 951
[58] 968 984

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

 [1]  17  56  84 105 134 146 172 176 195 196 207 230 255 270 289 292 294 325 352
[20] 354 359 373 416 434 441 457 465 468 482 484 497 512 569 572 614 636 657 671
[39] 688 717 720 722 747 759 760 771 774 781 810 820 821 831 836 870 873 883 909
[58] 926 928 948 951 968 993

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

```
[1] 7

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

[1] 6

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

[1] 9

> which(con2_adjp<=0.05/3)

[1] 289 457 512 688 781 928

> which(con3_adjp<=0.05/3)

[1] 105 207 457 497 688 781 821 883 968

> 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] 64

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

[1] 58

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

[1] 59

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

```
 [1]   2  62  80  84 121 130 140 142 188 193 200 205 225 230 244 284 291 294 330
[20] 338 368 418 419 424 432 439 467 489 507 515 521 523 524 554 575 578 586 588
[39] 603 623 632 658 661 676 678 694 697 713 744 756 762 767 848 854 881 884 889
[58] 905 906 911 912 942 966 981

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

 [1]  55  62  84 112 126 130 142 188 193 205 230 244 273 284 291 308 338 368 398
[20] 399 419 432 439 444 467 489 507 521 524 554 575 576 586 588 603 611 658 676
[39] 694 697 713 756 762 774 821 848 854 858 881 884 894 905 906 911 942 947 966
[58] 981

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

 [1]  55  62  80  84 112 121 130 142 166 175 193 225 230 244 273 284 294 330 338
[20] 368 399 424 432 439 467 489 498 507 521 523 524 554 575 586 588 603 611 632
[39] 676 678 697 713 716 743 756 762 774 848 854 858 881 884 905 906 911 942 947
[58] 966 981

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

[1] 7

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

[1] 2

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=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 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/RtmpG8L7xh/Rinst10d82e61425bfe/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] 82

> sum(diff_results$bootstrap_p<=0.05)

[1] 76

> 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] 4

> 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]
19  cg00016968 0.80628480            NA    0.81440820    0.83623180
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
173 cg00158308 0.06200759    0.08149979    0.08506825    0.14780450
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
346 cg00331237 0.05972383            NA    0.08204769    0.08345662
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
677 cg00651216 0.06825629    0.12529090    0.14409190    0.13907250
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
83     0.04965089    0.04833366    0.03466159    0.04390894
103    0.67349260    0.73510200    0.75715920    0.78981220
```

```
173    0.05964618    0.07405284    0.07625141    0.25718760
285    0.11693600    0.10650430    0.12281160    0.12310430
346    0.05372019    0.06241126    0.06955040    0.09140985
627    0.04740551    0.05332965    0.05775211    0.05579710
677    0.07669587    0.09597587    0.11690440    0.15194540
764    0.90575890    0.88760470    0.90756300    0.90946790
848    0.05230160    0.06119713    0.06542751    0.06240686
851    0.50820240    0.34657470    0.66276570    0.64634510
887    0.50259740    0.40111730    0.56646700    0.54552980
911    0.08633986    0.06765189    0.09070268    0.12417730
928    0.04490690    0.04204062    0.05050039    0.05268215
    diff_results$ordfit_t[diff_list_perm]
19                              -2.446404
83                               2.514109
103                             -2.268711
173                             -1.880788
285                             -3.093997
346                             -3.767916
627                             -2.239498
677                             -3.387628
764                             -1.808081
848                             -2.314412
851                             -2.841244
887                             -3.217939
911                             -3.621731
928                             -2.716443
    diff_results$permutation_p[diff_list_perm]
19                                          0
83                                          0
103                                         0
173                                         0
285                                         0
346                                         0
627                                         0
677                                         0
764                                         0
848                                         0
851                                         0
887                                         0
911                                         0
928                                         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]
200 cg00183916 0.03525946    0.03984548    0.02765822    0.02789838
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
743 cg00717862 0.07999436    0.07873347    0.06089359    0.06171374
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
200    0.03034811    0.04302129    0.02753873    0.03067437
259    0.04030003    0.03996053    0.05086962    0.05445672
280    0.61920530    0.61925200    0.46753250    0.55632410
743    0.07594936    0.09062161    0.06475791    0.07271878
    diff_results$ordfit_t[diff_list_boot]
200                              2.272449
259                             -4.052697
280                              4.170347
743                              3.444684
    diff_results$bootstrap_p[diff_list_boot]
200                                       0
259                                       0
280                                       0
743                                       0
```