# Introduction to RBM package

Dongmei Li

April 26, 2022

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] 50

> which(myresult$permutation_p<=0.05)

 [1]  12  30  57  71  96 111 124 126 168 201 225 259 286 380 402 418 439 448 452
[20] 487 499 504 507 549 557 607 622 632 648 652 653 655 675 703 741 746 763 768
[39] 782 808 828 859 861 889 898 913 949 953 968 998

> sum(myresult$bootstrap_p<=0.05)

[1] 5

> which(myresult$bootstrap_p<=0.05)

[1]  57  96 214 300 838

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

[1] 0

> 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] 27

> which(myresult2$bootstrap_p<=0.05)

 [1]  34  47 110 149 151 158 211 229 250 299 327 483 508 568 605 615 619 636 712
[20] 715 727 729 763 802 811 897 928

> 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] 55

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

[1] 55

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

[1] 46

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

 [1]   7  20  55  64  79  84  92 108 114 115 121 145 156 175 186 196 200 210 249
[20] 272 280 297 311 329 358 404 405 460 490 491 509 526 528 539 550 611 619 622
[39] 642 653 662 697 734 792 798 810 836 898 937 938 946 962 970 992 995

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

 [1]   7  10  20  55  79  92 108 114 115 121 137 145 156 170 175 186 192 196 200
[20] 210 256 272 280 297 311 329 358 404 405 485 490 491 509 526 528 539 550 619
[39] 642 653 662 697 734 798 809 810 820 863 898 900 938 962 970 992 993

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

 [1]   7  10  55  79  92 108 114 115 121 175 186 192 196 200 210 272 280 297 329
[20] 358 403 404 405 490 491 509 526 528 539 550 619 642 653 662 697 710 734 798
[39] 810 820 898 938 946 970 992 993
```

```
> 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] 1

> which(con2_adjp<=0.05/3)

[1]  55 114 297 619 938 992

> which(con3_adjp<=0.05/3)

[1] 114

> 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] 55

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

[1] 56

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

[1] 53
```

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

 [1]   8   9  22  24  37  41  45  55  76 117 120 165 185 197 207 224 313 319 321
[20] 369 406 442 481 495 507 509 520 534 568 572 580 586 599 603 605 619 631 638
[39] 645 687 688 705 746 758 793 798 811 846 895 909 946 954 957 974 992

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

 [1]   8   9  11  16  22  24  37  76  85  94 117 120 165 185 197 207 273 313 319
[20] 321 369 373 442 453 481 509 520 528 534 568 572 586 593 599 605 619 631 645
[39] 675 687 688 702 705 724 746 793 798 801 846 881 895 909 954 974 980 992

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

 [1]   8   9  11  22  24  37  45  76 117 120 165 185 197 207 273 313 319 321 369
[20] 373 442 505 509 520 556 568 572 580 586 593 599 605 619 631 638 645 675 687
[39] 688 705 724 746 793 798 846 881 895 909 919 954 962 974 992

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

[1] 8

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

[1] 3

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

[1] 6
```

# 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/RtmpcfUyr9/Rinstc92f42a567674/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] 48

> sum(diff_results$bootstrap_p<=0.05)

[1] 53

> 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] 5

> 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]
19  cg00016968 0.80628480            NA     0.8144082    0.83623180
764 cg00730260 0.90471270     0.9054229     0.9100268    0.91258610
851 cg00830029 0.58362500     0.5939787     0.6473961    0.67269640
887 cg00862290 0.43640520     0.5404716     0.6078680    0.56325950
911 cg00888479 0.07388961     0.0736108     0.1014980    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340     0.8491780
764    0.90575890    0.88760470    0.90756300     0.9094679
851    0.50820240    0.34657470    0.66276570     0.6463451
887    0.50259740    0.40111730    0.56646700     0.5455298
911    0.08633986    0.06765189    0.09070268     0.1241773
    diff_results$ordfit_t[diff_list_perm]
19                              -2.446404
764                             -1.808081
851                             -2.841244
887                             -3.217939
911                             -3.621731
    diff_results$permutation_p[diff_list_perm]
```

```
19                                          0
764                                         0
851                                         0
887                                         0
911                                         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]
280 cg00260778 0.6431989     0.6048896     0.5673506     0.5315091
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
280     0.6192053      0.619252     0.4675325     0.5563241
    diff_results$ordfit_t[diff_list_boot]
280                              4.170347
    diff_results$bootstrap_p[diff_list_boot]
280                                     0
```