Introduction to RBM package

Dongmei Li

October 30, 2025

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] 20

> which(myresult$permutation_p<=0.05)

 [1]  61  69  87  96 219 249 365 486 495 496 505 512 565 627 678 720 743 756 832
[20] 841

> sum(myresult$bootstrap_p<=0.05)

[1] 8

> which(myresult$bootstrap_p<=0.05)

[1]   4 273 360 363 505 527 681 887

> 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] 7

> which(myresult2$bootstrap_p<=0.05)

[1]  20  97 435 551 590 698 827

> 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] 70

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

[1] 61

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

[1] 62

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

 [1]   7  12  35  45  46  67 105 112 133 156 171 205 209 230 252 262 263 277 285
[20] 289 313 319 327 331 338 399 411 426 429 433 446 450 463 472 475 488 495 499
[39] 524 539 540 566 579 623 631 636 648 674 675 705 728 761 769 770 785 826 834
[58] 838 840 862 887 890 909 917 946 947 963 984 985 989

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

 [1]   7  12  35  40  45  46  67 105 112 133 171 209 230 263 277 285 289 313 319
[20] 325 331 338 339 399 401 411 426 429 433 463 472 475 488 499 524 566 579 623
[39] 631 636 648 652 705 728 758 761 769 793 826 834 838 840 862 887 909 917 946
[58] 947 963 985 989

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

 [1]   7  12  45  46  67 105 112 156 171 209 230 262 263 277 285 289 313 319 331
[20] 338 399 420 426 433 450 472 475 488 499 524 539 540 566 579 624 631 636 648
[39] 674 684 695 705 728 761 769 770 793 825 826 834 838 840 862 887 909 917 946
[58] 947 963 974 985 989

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

```
[1] 14

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

[1] 12

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

[1] 16

> which(con2_adjp<=0.05/3)

 [1]  46 105 263 289 426 631 648 728 826 840 887 947

> which(con3_adjp<=0.05/3)

 [1]  45  46 105 263 426 566 631 636 648 705 728 826 840 887 947 963

> 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] 67

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

[1] 58

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

[1] 63

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

```
 [1]    5   41   42   45   50   51   56   57   63   65   72  103  109  114  135
[16]  153  171  212  213  216  227  273  276  287  289  294  354  384  393  399
[31]  417  425  426  431  435  439  454  473  491  541  550  556  559  586  607
[46]  636  653  694  705  725  737  760  764  792  820  822  826  830  849  853
[61]  866  899  939  957  981  996 1000
```

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

```
 [1]    5   19   41   50   56   57   63   65   72  103  109  114  135  147  153
[16]  171  216  227  273  287  289  338  384  393  399  417  426  431  436  439
[31]  454  473  491  529  541  556  559  586  636  653  669  671  694  705  731
[46]  737  764  822  826  830  849  861  866  899  939  960  981 1000
```

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

```
 [1]    5   19   41   42   45   50   51   56   57   63   65   72  103  114  116
[16]  135  153  171  212  213  216  227  273  287  289  325  393  399  426  431
[31]  439  473  491  504  529  541  556  559  586  636  668  669  671  694  705
[46]  737  760  764  792  820  822  826  830  849  861  866  899  939  943  957
[61]  981  996 1000
```

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

```
[1] 13
```

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

```
[1] 5
```

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

```
[1] 8
```

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/RtmpH9xrEa/Rinst277e277b870136/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.:0.59031   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] 47

> sum(diff_results$permutation_p<=0.05)

[1] 53

> sum(diff_results$bootstrap_p<=0.05)

[1] 62

> 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] 2

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

[1] 9

> 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 0.8062848            NA     0.8144082     0.8362318
764 cg00730260 0.9047127     0.9054229     0.9100268     0.9125861
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19      0.8083138     0.7330644     0.8296834     0.8491780
764     0.9057589     0.8876047     0.9075630     0.9094679
    diff_results$ordfit_t[diff_list_perm]
19                              -2.547097
764                             -1.560713
    diff_results$permutation_p[diff_list_perm]
19                                       0
764                                      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]
16  cg00014085 0.05906804    0.04518973    0.04211710    0.03665208
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
346 cg00331237 0.05972383            NA    0.08204769    0.08345662
522 cg00503840 0.16919790    0.21350810    0.23958290    0.24448410
772 cg00743372 0.03922780    0.02919634    0.02187972    0.02568053
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
16     0.04222944    0.05324246    0.03728026    0.04062589
106    0.04693030    0.06837343    0.04534005    0.03709488
131    0.17352980    0.12564280    0.18193170    0.20847670
146    0.67191510    0.63137380    0.47929610    0.45428300
346    0.05372019    0.06241126    0.06955040    0.09140985
522    0.25154270    0.16469600    0.23368520    0.27977040
772    0.02796053    0.03512214    0.02575992    0.02093909
887    0.50259740    0.40111730    0.56646700    0.54552980
911    0.08633986    0.06765189    0.09070268    0.12417730
    diff_results$ordfit_t[diff_list_boot]
16                                1.954876
106                               2.887876
131                              -3.562745
146                               5.636263
346                              -3.328798
522                              -2.442169
772                               1.885560
887                              -3.368752
911                              -3.490240
    diff_results$bootstrap_p[diff_list_boot]
16                                        0
106                                       0
131                                       0
146                                       0
346                                       0
522                                       0
772                                       0
887                                       0
911                                       0
```