Introduction to RBM package

Dongmei Li

April 15, 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] 29

> which(myresult$permutation_p<=0.05)

 [1]   3  16  17 121 145 171 241 255 291 304 317 367 378 403 424 454 502 525 552
[20] 559 563 585 690 734 787 812 835 898 945

> sum(myresult$bootstrap_p<=0.05)

[1] 5

> which(myresult$bootstrap_p<=0.05)

[1] 147 213 588 679 737

> 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] 10

> which(myresult2$bootstrap_p<=0.05)

 [1]  32 168 173 261 434 721 757 770 798 925

> 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] 64

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

[1] 58

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

[1] 54

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

 [1]   9  12  26  27  71  77  87 145 206 246 266 298 305 327 342 351 362 369 375
[20] 382 390 443 458 468 471 505 523 525 542 557 561 569 592 625 633 694 713 722
[39] 736 738 747 759 784 793 802 807 815 849 850 852 900 908 919 927 932 934 938
[58] 943 950 981 983 987 992 998

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

 [1]   9  12  23  26  27  44  71  78  87 111 121 145 206 234 246 298 299 305 327
[20] 342 351 362 369 390 428 468 505 523 525 542 557 569 592 625 633 694 722 736
[39] 738 747 759 784 793 802 813 815 849 852 880 898 908 919 934 943 950 981 983
[58] 987

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

 [1]   9  12  23  26  44  71  77  87 111 121 145 206 246 299 305 327 342 351 362
[20] 369 382 399 468 523 525 542 557 569 592 625 633 694 722 736 747 759 784 793
[39] 802 815 849 850 852 880 919 927 934 938 941 943 950 981 983 987

> 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] 10

> 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]   9  26  87 342 369 525 542 722 784 852

> which(con3_adjp<=0.05/3)

[1]   9  26 342 525 784

> 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] 73

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

[1] 58

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

[1] 72

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

```
 [1]  16  50  74  89  93 109 122 125 136 139 141 164 181 209 219 224 236 245 253
[20] 272 273 278 284 286 301 308 337 354 357 363 370 392 397 409 413 420 426 427
[39] 442 448 484 509 523 530 562 566 589 600 605 617 619 624 641 652 663 691 720
[58] 729 759 765 770 774 782 798 814 852 885 895 911 926 952 993 994
```

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

```
 [1]  41  48  50  67  74 109 122 139 141 181 209 219 224 236 245 253 278 284 286
[20] 337 354 357 370 392 397 413 420 427 448 479 484 509 523 589 600 602 617 619
[39] 624 641 663 675 686 720 759 770 774 782 808 814 852 885 895 926 952 979 993
[58] 994
```

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

```
 [1]   8  38  48  74  92 109 122 125 136 139 141 181 209 219 224 236 245 253 272
[20] 278 284 286 308 315 337 354 357 360 370 392 397 413 420 426 427 448 457 477
[39] 484 509 523 530 566 589 600 602 617 619 624 641 654 663 675 690 759 765 770
[58] 774 777 781 782 814 841 852 874 885 891 911 926 952 993 994
```

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

```
[1] 12
```

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

```
[1] 6
```

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

```
[1] 10
```

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] "/private/var/folders/r0/l4fjk6cj5xj0j3brt4bplpl40000gt/T/RtmpUrKhJA/Rinst971d158f98ba/RBM/d

> 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] 67
```

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

```
[1] 75
```

```
> 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] 3
```

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

```
[1] 14
```

```
> 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
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19      0.8083138    0.73306440    0.82968340    0.84917800
764     0.9057589    0.88760470    0.90756300    0.90946790
848     0.0523016    0.06119713    0.06542751    0.06240686
    diff_results$ordfit_t[diff_list_perm]
19                              -2.547097
764                             -1.560713
848                             -1.687144
    diff_results$permutation_p[diff_list_perm]
19                                           0
764                                          0
848                                          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
95  cg00081975 0.03633894    0.04975194    0.06024723    0.05598723
97  cg00083937 0.53046980    0.60529020    0.62733150    0.65623920
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
189 cg00176210 0.28756520    0.39161870    0.44272520    0.44725330
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
632 cg00615377 0.11265030    0.16140570    0.19404450    0.17468600
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
931 cg00901704 0.05734342    0.04812868    0.04478214    0.03878488
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
16     0.04222944    0.05324246    0.03728026    0.04062589
95     0.04561792    0.05115624    0.06068253    0.06168212
97     0.55974270    0.43157020    0.64046990    0.57876990
106    0.04693030    0.06837343    0.04534005    0.03709488
146    0.67191510    0.63137380    0.47929610    0.45428300
189    0.34106080    0.33765930    0.41252110    0.37024890
259    0.04030003    0.03996053    0.05086962    0.05445672
280    0.61920530    0.61925200    0.46753250    0.55632410
632    0.12573100    0.14483660    0.16338240    0.20130510
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
931    0.04497277    0.05751033    0.03089829    0.04423603
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
16                                 1.954876
95                                -2.654324
97                                -2.665377
106                                2.887876
146                                5.636263
189                               -3.232921
259                               -2.833203
280                                4.337628
632                               -3.722206
887                               -3.368752
911                               -3.490240
928                               -1.982308
931                                2.127264
979                               -4.968792
```

```
    diff_results$bootstrap_p[diff_list_boot]
16                                         0
95                                         0
97                                         0
106                                        0
146                                        0
189                                        0
259                                        0
280                                        0
632                                        0
887                                        0
911                                        0
928                                        0
931                                        0
979                                        0
```