Introduction to RBM package

Dongmei Li

May 30, 2026

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] 13

> which(myresult$permutation_p<=0.05)

 [1] 108 224 429 460 520 693 710 752 755 790 898 933 978

> sum(myresult$bootstrap_p<=0.05)

[1] 1

> which(myresult$bootstrap_p<=0.05)

[1] 610

> 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] 47

> which(myresult2$bootstrap_p<=0.05)

 [1]  14  29  52  67  88 140 157 183 184 188 214 225 237 298 327 384 447 462 481
[20] 516 542 549 559 573 582 584 590 607 631 638 641 655 697 713 722 748 755 763
[39] 794 830 835 852 863 879 965 970 992

> 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] 34

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

[1] 45

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

[1] 52

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

 [1]  14  16  29  68 155 208 211 215 255 281 305 337 365 397 440 442 457 489 512
[20] 556 571 577 683 693 716 759 805 828 879 902 907 925 955 986

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

 [1]  14  29  63  68 106 116 155 208 211 215 255 281 302 305 337 365 397 440 442
[20] 457 489 512 519 556 571 577 591 601 621 641 675 683 693 716 731 759 770 805
[39] 825 828 879 902 907 955 998

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

 [1]  14  16  29  54  68 155 208 211 215 258 281 302 335 337 365 388 397 440 442
[20] 457 482 489 490 512 549 556 571 577 591 601 621 641 659 675 683 693 716 741
[39] 759 770 789 805 825 828 879 902 907 925 939 955 986 998

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

[1] 1
```

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

[1] 7

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

[1] 6

> which(con2_adjp<=0.05/3)

[1]  14  29  68 155 337 577 955

> which(con3_adjp<=0.05/3)

[1]  14  29 577 683 879 955

> 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] 62

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

[1] 50

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

 [1]   5  57  67  78  97 107 116 117 136 140 147 164 171 172 177 215 218 222 259
[20] 279 291 307 317 326 340 341 343 344 422 426 428 435 444 450 471 482 516 519
[39] 532 544 602 630 666 674 704 718 737 746 770 800 815 846 848 851 855 862 902
[58] 910 924 943 944 951 964 967
```

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

 [1]   5  57  67  78  97  98 107 116 117 120 140 147 164 172 177 215 218 225 259
[20] 291 307 317 326 340 341 343 344 361 422 439 444 450 471 482 493 522 532 543
[39] 544 602 630 652 674 675 691 694 704 718 746 800 815 846 848 851 862 883 924
[58] 944 951 967 978 999

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

 [1]   5  57  67  78 107 116 117 136 140 164 171 218 255 259 291 307 317 326 341
[20] 343 344 422 443 444 450 471 482 517 532 543 544 652 674 691 704 718 746 770
[39] 800 815 829 846 862 910 924 944 951 967 973 999

> 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] 11

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

[1] 2
```

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/RtmpbP3Jgp/Rinst2ae101e84ba/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                     NAs    :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
                   NAs    :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] 60

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

```
[1] 47

> 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] 7

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

[1] 2

> 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
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
450 cg00432979 0.03681359    0.04515700    0.04374394    0.03683598
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
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
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
245    0.04208405    0.05284988    0.03775905    0.03955271
450    0.04419125    0.04409653    0.02839263    0.03410020
764    0.90575890    0.88760470    0.90756300    0.90946790
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
    diff_results$ordfit_t[diff_list_perm]
19                              -2.547097
245                              1.494678
450                              1.274731
764                             -1.560713
851                             -2.986319
887                             -3.368752
911                             -3.490240
    diff_results$permutation_p[diff_list_perm]
```

```
19                                      0
245                                     0
450                                     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]
131 cg00121904 0.1544958     0.1794975     0.2360811     0.2435415
146 cg00134539 0.6110132     0.5332178     0.4599934     0.4678742
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
131     0.1735298     0.1256428     0.1819317     0.2084767
146     0.6719151     0.6313738     0.4792961     0.4542830
    diff_results$ordfit_t[diff_list_boot]
131                             -3.562745
146                              5.636263
    diff_results$bootstrap_p[diff_list_boot]
131                                     0
146                                     0
```