

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	1
2	Getting started	2
3	RBM_T and RBM_F functions	2
4	Ovarian cancer methylation example using the RBM_T function	6

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 Benjamini-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] 40

> which(myresult$permutation_p<=0.05)

[1] 10 14 27 118 124 150 188 202 275 300 304 306 408 410 411 428 473 498 518
[20] 540 542 621 631 655 688 721 728 732 734 737 789 800 816 860 867 881 917 940
[39] 973 986

> sum(myresult$bootstrap_p<=0.05)

[1] 12

> which(myresult$bootstrap_p<=0.05)

[1] 85 100 118 124 553 576 578 621 731 743 819 881

> 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] 41

> which(myresult2$bootstrap_p<=0.05)

[1] 12 44 48 86 141 156 168 181 214 264 267 317 352 353 361 382 404 425 462
[20] 489 540 554 573 621 633 642 662 670 692 757 770 789 819 820 850 866 876 886
[39] 887 947 957

> 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] 66

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

[1] 42

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

[1] 54

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

[1] 1 7 25 30 37 46 60 84 110 111 121 123 126 146 154 174 186 220 226
[20] 234 237 241 244 276 307 310 340 345 376 392 421 455 489 509 553 564 570 571
[39] 580 598 610 629 632 706 717 735 759 790 804 815 829 855 870 876 888 889 895
[58] 900 901 916 931 938 940 950 998 999

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

[1] 15 24 30 37 60 111 123 126 127 146 154 186 220 226 241 251 276 283 307
[20] 327 376 455 571 572 580 585 598 706 717 735 796 815 855 870 876 887 889 900
[39] 931 938 950 999

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

[1] 24 30 37 60 84 111 123 126 127 154 174 186 188 220 226 235 241 251 276
[20] 307 310 317 322 327 376 392 421 455 553 570 571 580 585 598 629 632 706 717
[39] 735 790 815 829 855 870 876 889 895 900 901 916 938 940 950 999
```

```

> con1_adj_p <- p.adjust(myresult_F$permutation_p[, 1], "BH")
> sum(con1_adj_p<=0.05/3)

[1] 10

> con2_adj_p <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adj_p<=0.05/3)

[1] 6

> con3_adj_p <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adj_p<=0.05/3)

[1] 7

> which(con2_adj_p<=0.05/3)

[1] 37 123 220 241 815 938

> which(con3_adj_p<=0.05/3)

[1] 37 307 421 580 706 829 938

> 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] 69

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

[1] 64

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

[1] 71

```

```

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

[1] 22 46 75 87 105 158 160 185 222 231 241 256 286 319 323
[16] 334 350 352 358 369 393 399 405 410 424 432 466 468 474 475
[31] 476 500 525 526 533 545 550 556 574 580 595 605 618 625 628
[46] 631 632 638 659 667 669 674 696 719 722 734 741 754 815 827
[61] 829 856 867 903 932 933 965 977 1000

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

[1] 22 38 70 75 87 105 166 171 185 194 219 222 231 241 256
[16] 293 319 323 334 341 350 352 358 369 393 394 399 410 445 466
[31] 475 476 500 525 526 533 550 561 574 580 595 618 625 628 631
[46] 632 659 667 669 674 696 722 734 815 827 829 867 932 933 937
[61] 962 965 977 1000

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

[1] 22 38 46 70 87 93 105 158 160 166 171 194 222 241 256
[16] 293 308 319 323 334 341 350 352 358 369 399 410 432 466 468
[31] 476 480 500 525 526 533 545 550 561 572 574 580 595 618 625
[46] 628 631 632 638 646 659 667 669 674 696 704 719 722 734 815
[61] 827 848 856 901 903 932 937 945 965 977 1000

> con21_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adj_p<=0.05/3)

[1] 9

> con22_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adj_p<=0.05/3)

[1] 5

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=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 illustration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovarian 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/db/4tvngx8jx4z3fm1gzlnlzw9rc0000gq/T/RtmpqHWYzi/Rinst9fe444eaedf9/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] 65

> sum(diff_results$bootstrap_p<=0.05)

[1] 21

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

[1] 0

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

[1] 5

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

[1] 0

> diff_list_perm <- which(perm_adj<=0.05)
> diff_list_boot <- which(boot_adj<=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
517 cg00499822 0.09723835    0.13925420    0.12969170    0.15998260
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19      0.8083138    0.73306440    0.82968340    0.84917800
517      0.1100917    0.08752679    0.15305730    0.21607890
764      0.9057589    0.88760470    0.90756300    0.90946790
851      0.5082024    0.34657470    0.66276570    0.64634510
928      0.0449069    0.04204062    0.05050039    0.05268215
      diff_results$ordfit_t[diff_list_perm]

```



```

19          -2.547097
517         -2.943836
764         -1.560713
851         -2.986319
928         -1.982308
  diff_results$permutation_p[diff_list_perm]
19          0
517         0
764         0
851         0
928         0

> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t[
> print(sig_results_boot)

[1] IlmnID
[2] Beta
[3] exmdata2[, 2]
[4] exmdata3[, 2]
[5] exmdata4[, 2]
[6] exmdata5[, 2]
[7] exmdata6[, 2]
[8] exmdata7[, 2]
[9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_boot]
[11] diff_results$bootstrap_p[diff_list_boot]
<0 rows> (or 0-length row.names)

```