

# Introduction to RBM package

Dongmei Li

April 28, 2026

Clinical and Translational Science Institute, University of Rochester School of Medicine and  
Dentistry, Rochester, NY 14642-0708

## Contents

<b>1 Overview</b>	<b>1</b>
<b>2 Getting started</b>	<b>2</b>
<b>3 RBM_T and RBM_F functions</b>	<b>2</b>
<b>4 Ovarian cancer methylation example using the RBM_T function</b>	<b>6</b>

## 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] 66

> which(myresult$permutation_p<=0.05)

[1] 19 31 32 40 48 60 81 84 93 113 117 134 141 145 151 159 169 183 234
[20] 236 250 269 282 299 303 314 332 341 359 361 367 386 394 401 408 415 424 434
[39] 462 475 489 506 528 533 588 621 624 647 662 677 702 718 738 766 769 783 810
[58] 812 840 901 907 914 956 980 995 998

> sum(myresult$bootstrap_p<=0.05)

[1] 10

> which(myresult$bootstrap_p<=0.05)

[1] 19 145 197 249 269 303 367 394 506 823

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

[1] 0

> bootstrap_adj_p <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adj_p<=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] 23

> which(myresult2$bootstrap_p<=0.05)

[1] 26 102 106 108 213 220 242 296 318 333 351 405 449 496 541 610 611 750 828
[20] 855 893 940 984

> bootstrap2_adj_p <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adj_p<=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] 60

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

[1] 63

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

[1] 63

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

[1] 19 25 27 81 98 108 120 127 131 134 184 199 204 206 216 238 239 241 253
[20] 254 263 278 284 311 312 336 344 378 380 418 422 440 456 462 463 470 492 525
[39] 527 565 573 591 597 636 641 695 718 724 728 743 767 856 862 874 878 898 908
[58] 948 957 973

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

[1] 13 19 81 83 96 98 120 127 131 134 184 204 206 213 216 238 240 241 250
[20] 254 263 278 311 312 336 344 378 380 418 422 456 462 463 470 492 525 565 573
[39] 591 597 608 610 641 694 695 718 722 724 728 743 767 856 862 874 878 905 908
[58] 948 957 973 975 988 993

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

[1] 13 19 42 53 81 98 108 120 131 134 176 184 198 204 206 213 216 239 241
[20] 253 254 263 303 311 312 336 344 359 378 380 418 440 456 462 463 470 492 525
[39] 527 565 568 573 591 597 641 695 718 724 728 743 856 862 874 878 898 908 948
[58] 957 973 974 975 987 993
```

```

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

[1] 2

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

[1] 11

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

[1] 1

> which(con2_adj_p<=0.05/3)

[1] 98 131 204 254 418 525 591 597 641 718 908

> which(con3_adj_p<=0.05/3)

[1] 525

> 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] 63

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

[1] 53

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

[1] 58

```

```

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

[1] 36 50 90 115 125 136 159 176 180 183 190 224 238 244 247 254 279 309 316
[20] 328 350 400 402 410 428 432 440 442 455 457 476 486 497 502 524 525 568 597
[39] 617 660 689 691 698 703 722 727 741 761 787 802 806 808 818 821 830 847 848
[58] 859 876 948 963 973 988

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

[1] 13 36 50 101 115 136 155 159 176 180 190 244 247 254 309 378 388 400 402
[20] 410 428 432 440 442 447 457 476 486 522 524 568 617 657 660 689 691 698 703
[39] 741 802 806 808 818 821 830 848 865 876 948 963 973 979 988

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

[1] 36 50 101 115 155 159 176 180 190 238 244 247 254 309 328 350 363 378 388
[20] 400 402 410 428 432 442 447 457 486 497 502 522 524 525 568 617 657 689 691
[39] 703 711 727 741 761 802 806 808 818 821 847 848 865 929 948 951 963 973 979
[58] 988

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

[1] 16

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

[1] 9

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=0.05/3)

[1] 11

```

## 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 genome-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/r0/l4fjk6cj5xj0j3brt4bplpl40000gt/T/Rtmp98H8gM/Rinst14f1c125630ae/RBM/"

> 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] 55

> sum(diff_results$bootstrap_p<=0.05)

[1] NA

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

[1] 0

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

[1] 2

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

[1] NA

> diff_list_perm <- which(perm_adj_p<=0.05)
> diff_list_boot <- which(boot_adj_p<=0.05)
> sig_results_perm <- cbind(ovarian_cancer_methylation[diff_list_perm, ], diff_results$ordfit_t[diff_list_perm, ])
> print(sig_results_perm)

      IlmnID      Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
103 cg00094319 0.7378428      0.7353296      0.7557490      0.7383022
851 cg00830029 0.5836250      0.5939787      0.6473961      0.6726964
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
103      0.6734926      0.7351020      0.7571592      0.7898122
851      0.5082024      0.3465747      0.6627657      0.6463451
      diff_results$ordfit_t[diff_list_perm]
103                                -2.343784
851                                -2.986319
      diff_results$permutation_p[diff_list_perm]
103                                0
851                                0

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

      IlmnID      Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
146 cg00134539 0.6110132      0.5332178      0.4599934      0.4678742
632 cg00615377 0.1126503      0.1614057      0.1940445      0.1746860
979 cg00945507 0.1343225      0.2385460      0.3474976      0.2890334

```



	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]
146	0.6719151	0.6313738	0.4792961	0.4542830
632	0.1257310	0.1448366	0.1633824	0.2013051
979	0.1184851	0.1665385	0.3071842	0.2662474

  

	diff_results\$ordfit_t[diff_list_boot]
146	5.636263
632	-3.722206
979	-4.968792

  

	diff_results\$bootstrap_p[diff_list_boot]
146	0
632	0
979	0