

Introduction to RBM package

Dongmei Li

January 25, 2026

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] 113

> which(myresult$permutation_p<=0.05)

[1] 2 5 10 37 54 65 67 69 80 111 113 128 136 149 152 172 194 200
[19] 205 210 217 221 225 233 253 255 276 285 288 292 297 301 304 307 311 328
[37] 333 336 344 353 362 366 368 375 377 381 384 386 388 416 426 429 430 446
[55] 456 458 463 469 475 481 492 495 502 515 524 532 542 544 546 573 577 589
[73] 591 607 629 633 646 647 679 688 698 705 706 730 731 736 742 777 778 792
[91] 795 796 814 817 818 838 848 865 867 873 875 876 884 906 909 913 916 918
[109] 933 939 947 964 991

> sum(myresult$bootstrap_p<=0.05)

[1] 5

> which(myresult$bootstrap_p<=0.05)

[1] 69 152 377 384 426

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

[1] 5

> 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] 4

> which(myresult2$bootstrap_p<=0.05)

[1] 720 733 757 838

> 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] 58
```

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

```
[1] 69
```

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

```
[1] 52
```

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

```
[1] 1 81 108 126 135 146 154 162 180 183 188 190 191 194 217 223 276 281 297
[20] 325 332 358 360 389 416 451 471 477 479 524 525 534 570 589 637 641 651 664
[39] 698 703 726 740 760 764 778 781 797 801 836 838 846 930 936 937 948 953 956
[58] 985
```

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

```
[1] 1 44 53 81 83 108 109 135 146 154 161 180 188 190 191 194 276 281 297
[20] 325 332 351 358 360 389 392 416 436 451 471 477 479 524 525 534 560 570 589
[39] 602 637 641 651 661 664 698 703 716 726 760 764 778 781 797 801 802 820 836
[58] 838 846 860 900 926 932 936 937 948 953 956 985
```

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

```

[1] 1 53 81 90 126 135 146 162 180 188 191 194 223 235 297 311 325 332 360
[20] 371 389 416 471 479 502 508 509 524 525 570 637 641 651 664 702 703 716 726
[39] 740 764 778 781 797 801 802 836 860 932 936 953 956 992

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

[1] 6

> 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] 7

> which(con2_adj_p<=0.05/3)

[1] 297 360 389 471 641 651 703 760 797 801 836

> which(con3_adj_p<=0.05/3)

[1] 297 332 389 471 570 651 936

> 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] 60

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

[1] 68

```

```

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

[1] 62

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

[1] 3 19 30 61 67 72 74 79 82 83 186 189 194 249 265 313 332 339 374
[20] 388 393 471 493 504 514 522 524 534 543 545 556 557 560 569 575 579 599 618
[39] 638 679 685 718 740 745 764 796 797 815 830 837 842 857 863 896 900 915 932
[58] 941 954 986

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

[1] 3 5 10 11 19 30 48 61 72 74 79 82 83 117 178 186 194 249 263
[20] 265 313 332 370 374 388 393 459 471 493 504 511 514 522 524 534 545 556 557
[39] 560 575 579 599 611 618 628 638 659 679 685 710 728 739 740 795 796 797 815
[58] 830 842 849 863 896 900 915 932 941 954 986

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

[1] 3 10 11 30 61 67 72 74 79 82 178 185 186 189 194 249 263 313 332
[20] 339 344 369 383 384 388 429 459 481 493 504 511 514 522 524 534 543 557 575
[39] 599 611 618 659 679 685 710 728 739 740 795 796 797 815 837 842 849 863 896
[58] 900 915 932 973 986

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

[1] 8

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

[1] 6

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=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 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] "/tmp/Rtmpu0U0vn/Rinst2ae4f73d32d/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] 78

> sum(diff_results$bootstrap_p<=0.05)

[1] 66

> 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] 8

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

[1] 8

> 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
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
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
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19      0.80831380    0.73306440    0.82968340    0.84917800
131      0.17352980    0.12564280    0.18193170    0.20847670

```


245	0.04208405	0.05284988	0.03775905	0.03955271
280	0.61920530	0.61925200	0.46753250	0.55632410
627	0.04740551	0.05332965	0.05775211	0.05579710
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

diff_results\$ordfit_t[diff_list_perm]

19	-2.547097
131	-3.562745
245	1.494678
280	4.337628
627	-1.797392
764	-1.560713
851	-2.986319
887	-3.368752

diff_results\$permutation_p[diff_list_perm]

19	0
131	0
245	0
280	0
627	0
764	0
851	0
887	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]
83	cg00072216	0.04505377	0.04598964	0.04000674	0.03231534
146	cg00134539	0.61101320	0.53321780	0.45999340	0.46787420
252	cg00230502	0.10061390	0.13517870	0.12538510	0.16304920
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
346	cg00331237	0.05972383	NA	0.08204769	0.08345662
632	cg00615377	0.11265030	0.16140570	0.19404450	0.17468600
979	cg00945507	0.13432250	0.23854600	0.34749760	0.28903340
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
83	0.04965089	0.04833366	0.03466159	0.04390894	
146	0.67191510	0.63137380	0.47929610	0.45428300	
252	0.11970870	0.12036160	0.17423730	0.18155480	
259	0.04030003	0.03996053	0.05086962	0.05445672	
280	0.61920530	0.61925200	0.46753250	0.55632410	
346	0.05372019	0.06241126	0.06955040	0.09140985	
632	0.12573100	0.14483660	0.16338240	0.20130510	
979	0.11848510	0.16653850	0.30718420	0.26624740	

```

diff_results$ordfit_t[diff_list_boot]
83      1.947226
146     5.636263
252    -3.204466
259    -2.833203
280     4.337628
346    -3.328798
632    -3.722206
979    -4.968792
diff_results$bootstrap_p[diff_list_boot]
83      0
146     0
252     0
259     0
280     0
346     0
632     0
979     0

```