

# Introduction to RBM package

Dongmei Li

April 30, 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] 46

> which(myresult$permutation_p<=0.05)

[1] 10 26 28 41 50 80 89 109 111 132 140 162 176 195 196 208 212 267 273
[20] 294 331 399 423 461 468 477 478 493 557 559 593 642 660 683 701 727 841 842
[39] 873 876 888 904 944 949 963 997

> sum(myresult$bootstrap_p<=0.05)

[1] 0

> which(myresult$bootstrap_p<=0.05)

integer(0)

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

[1] 1

> 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] 29

> which(myresult2$bootstrap_p<=0.05)

[1] 17 61 72 121 141 154 189 367 402 446 474 501 529 541 587 631 635 656 704
[20] 712 758 778 790 863 957 968 970 976 998

> 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] 51

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

[1] 54

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

[1] 5 10 77 87 89 96 111 117 118 200 203 227 278 285 292 294 312 342 360
[20] 370 374 398 407 414 462 551 567 587 597 612 617 654 663 683 687 731 746 761
[39] 776 802 820 822 841 845 848 856 860 869 877 884 901 903 910 923 935 957 965
[58] 969 986 999

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

[1] 5 10 87 89 96 117 118 136 200 278 285 292 294 312 318 342 398 407 414
[20] 422 459 460 551 571 579 587 597 612 654 683 687 731 746 761 802 819 820 822
[39] 841 845 848 856 869 884 901 903 957 965 969 986 999

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

[1] 5 10 77 87 89 96 111 117 200 227 285 290 294 312 318 342 370 374 392
[20] 398 407 414 460 551 571 579 587 597 612 617 654 683 687 731 761 795 802 819
[39] 820 822 841 845 848 856 869 884 901 903 910 957 965 969 986 999
```

```

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

[1] 15

> 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] 9

> which(con2_adjp<=0.05/3)

[1] 117 285 398 597 731 761 820 841 848 986

> which(con3_adjp<=0.05/3)

[1] 89 285 597 802 841 848 884 986 999

> 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] 51

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

[1] 43

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

[1] 39

```

```

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

[1] 19 34 53 55 66 80 134 149 153 167 182 193 196 212 270 274 284 292 337
[20] 360 368 411 430 435 437 446 464 473 544 629 652 665 666 673 705 731 773 781
[39] 788 805 826 845 856 888 908 926 938 958 960 977 981

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

[1] 34 53 55 80 104 109 153 156 167 180 196 270 271 284 292 295 360 368 411
[20] 430 435 437 446 464 473 600 629 652 665 666 671 673 731 773 781 805 845 888
[39] 908 926 960 977 981

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

[1] 19 34 53 55 66 109 149 153 156 167 180 196 212 270 284 292 368 403 411
[20] 430 435 437 446 464 473 629 652 665 666 671 731 805 845 888 908 938 960 977
[39] 981

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

[1] 5

> 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] 3

```

## 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/RtmpUWziIT/Rinst2ae67c4e9a7/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] 47

> sum(diff_results$bootstrap_p<=0.05)

[1] 57

> 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] 4

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

[1] 2

> 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)
> print(sig_results_perm)

      IlmnID      Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
280 cg00260778 0.64319890      0.60488960      0.56735060      0.53150910
764 cg00730260 0.90471270      0.90542290      0.91002680      0.91258610
848 cg00826384 0.05721674      0.05612171      0.06644259      0.06358381
851 cg00830029 0.58362500      0.59397870      0.64739610      0.67269640
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
280      0.6192053      0.61925200      0.46753250      0.55632410
764      0.9057589      0.88760470      0.90756300      0.90946790
848      0.0523016      0.06119713      0.06542751      0.06240686
851      0.5082024      0.34657470      0.66276570      0.64634510
      diff_results$ordfit_t[diff_list_perm]
280                                4.337628
764                               -1.560713
848                               -1.687144
851                               -2.986319
      diff_results$permutation_p[diff_list_perm]
280                                0
764                                0
848                                0
851                                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]
280	cg00260778	0.6431989	0.6048896	0.5673506	0.5315091
979	cg00945507	0.1343225	0.2385460	0.3474976	0.2890334
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
280	0.6192053	0.6192520	0.4675325	0.5563241	
979	0.1184851	0.1665385	0.3071842	0.2662474	
	diff_results\$ordfit_t[diff_list_boot]				
280	4.337628				
979	-4.968792				
	diff_results\$bootstrap_p[diff_list_boot]				
280	0				
979	0				