

# Introduction to RBM package

Dongmei Li

May 12, 2026

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

## Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Getting started</b>	<b>2</b>
<b>3</b>	<b>RBM_T and RBM_F functions</b>	<b>2</b>
<b>4</b>	<b>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] 18

> which(myresult$permutation_p<=0.05)

[1] 55 77 121 169 238 277 333 336 347 409 460 482 518 540 606 712 724 991

> sum(myresult$bootstrap_p<=0.05)

[1] 10

> which(myresult$bootstrap_p<=0.05)

[1] 175 280 408 569 577 587 747 802 878 936

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

[1] 3

> 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] 21

> which(myresult2$bootstrap_p<=0.05)

[1] 30 55 184 229 237 269 286 345 363 373 387 595 660 738 764 805 817 822 830
[20] 846 963

> 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] 54

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

[1] 71

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

[1] 76

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

[1] 23 25 68 70 77 98 140 179 186 225 246 251 267 274 277 295 303 304 308
[20] 372 394 406 419 421 428 441 445 463 505 507 528 531 538 562 609 626 663 674
[39] 684 698 713 726 731 776 803 813 840 876 881 899 901 920 946 970

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

[1] 7 11 23 24 25 68 77 98 107 171 179 186 221 225 244 246 267 274 277
[20] 295 303 304 308 309 312 320 322 372 394 406 419 421 425 437 441 445 505 507
[39] 528 531 538 562 585 626 663 664 674 684 700 726 731 755 803 813 818 829 835
[58] 840 853 875 876 881 890 895 899 901 915 920 940 946 970

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

[1] 7 11 23 24 25 45 68 70 77 98 107 138 171 179 186 221 225 244 246
[20] 251 267 274 277 291 295 303 304 308 309 312 320 322 372 394 405 419 421 428
[39] 437 441 445 505 507 513 528 531 538 562 582 605 626 663 674 684 713 726 731
[58] 755 803 813 818 840 853 865 866 872 875 876 881 895 899 920 940 946 970 976

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

```

```

[1] 4

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

[1] 11

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

[1] 15

> which(con2_adjp<=0.05/3)

[1] 23 77 186 308 320 441 505 528 626 731 970

> which(con3_adjp<=0.05/3)

[1] 7 23 179 186 267 308 372 394 419 421 507 538 562 674 755

> 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] 54

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

[1] 56

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

[1] 57

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

```

```

[1] 75 93 141 144 155 190 219 220 233 246 247 264 315 321 347 359 381 389 408
[20] 409 433 435 455 459 460 495 504 530 553 572 586 593 605 615 616 642 649 671
[39] 747 775 793 795 803 811 901 903 921 932 938 941 944 959 972 984

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

[1] 52 75 93 110 141 144 155 190 220 233 246 264 315 321 347 359 381 389 408
[20] 409 426 428 433 455 459 495 553 572 579 581 586 593 605 615 616 642 649 747
[39] 767 775 795 803 811 856 862 901 903 921 932 938 941 944 959 963 972 984

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

[1] 75 88 93 107 141 144 190 220 233 246 264 315 321 335 346 347 359 381 389
[20] 408 409 433 455 459 460 495 504 530 533 553 572 586 593 605 615 642 649 671
[39] 683 747 767 775 793 795 803 811 901 903 921 932 938 941 944 959 963 984 985

> con21_adjp <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adjp<=0.05/3)

[1] 6

> con22_adjp <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adjp<=0.05/3)

[1] 6

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=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 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/RtmpUe2VmN/Rinst2ae66a7884b/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] 57

> sum(diff_results$bootstrap_p<=0.05)

[1] 39

> 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] 5

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

[1] 1

> 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]
83	cg00072216	0.04505377	0.04598964	0.04000674	0.03231534
103	cg00094319	0.73784280	0.73532960	0.75574900	0.73830220
131	cg00121904	0.15449580	0.17949750	0.23608110	0.24354150
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
851	cg00830029	0.58362500	0.59397870	0.64739610	0.67269640
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
83	0.04965089	0.04833366	0.03466159	0.04390894	
103	0.67349260	0.73510200	0.75715920	0.78981220	
131	0.17352980	0.12564280	0.18193170	0.20847670	
280	0.61920530	0.61925200	0.46753250	0.55632410	
851	0.50820240	0.34657470	0.66276570	0.64634510	
	diff_results\$ordfit_t[diff_list_perm]				
83	1.947226				
103	-2.343784				
131	-3.562745				
280	4.337628				
851	-2.986319				
	diff_results\$permutation_p[diff_list_perm]				
83	0				
103	0				
131	0				
280	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]
911 cg00888479 0.07388961    0.0736108    0.101498    0.09985076
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
911    0.08633986    0.06765189    0.09070268    0.1241773
      diff_results$ordfit_t[diff_list_boot]
911                                -3.49024
      diff_results$bootstrap_p[diff_list_boot]
911                                0
```