

# 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] 35

> which(myresult$permutation_p<=0.05)

[1] 35 36 38 97 147 148 157 219 313 338 360 366 442 443 460 462 479 525 555
[20] 582 590 619 636 723 735 804 807 835 849 908 922 949 953 971 995

> sum(myresult$bootstrap_p<=0.05)

[1] 9

> which(myresult$bootstrap_p<=0.05)

[1] 147 157 307 366 479 519 586 588 626

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

[1] 9

> 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] 25

> which(myresult2$bootstrap_p<=0.05)

[1] 17 26 44 118 188 224 258 283 299 427 470 525 593 614 673 691 711 749 770
[20] 784 793 834 849 869 909

> 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] 53

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

[1] 53

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

[1] 22 26 33 40 62 70 78 96 115 146 148 166 177 179 200 212 222 237 326
[20] 354 360 364 365 379 414 421 430 450 483 511 512 528 579 607 627 677 688 699
[39] 709 734 737 763 792 816 817 822 835 846 855 866 879 880 881 882 897 941 978
[58] 986

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

[1] 22 26 33 40 62 78 96 115 136 146 148 177 179 190 200 212 222 237 326
[20] 354 360 364 365 379 414 421 430 483 511 512 528 579 580 607 627 688 734 737
[39] 763 772 816 822 832 835 879 880 881 882 897 915 953 978 986

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

[1] 22 26 33 40 62 64 70 78 115 146 148 166 177 179 200 212 222 237 326
[20] 332 354 360 365 379 414 421 430 483 511 512 528 607 627 677 688 699 734 737
[39] 763 792 816 822 835 855 866 879 880 881 915 953 964 978 986

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

[1] 10

```

```

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

[1] 0

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

[1] 8

> which(con2_adjp<=0.05/3)

integer(0)

> which(con3_adjp<=0.05/3)

[1] 40 62 483 763 822 835 978 986

> 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] 66

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

[1] 50

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

[1] 58

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

[1] 19 33 60 62 67 83 172 190 195 233 242 266 274 283 303 311 343 352 388
[20] 457 478 489 518 530 535 558 584 588 622 626 638 645 649 653 665 718 719 736
[39] 738 748 751 760 768 771 779 785 793 809 827 837 840 846 849 850 874 886 892
[58] 906 934 950 952 969 971 983 995 998

```

```

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

[1] 19 33 60 61 67 83 172 233 242 266 274 283 303 343 345 388 442 457 489
[20] 518 530 535 588 622 626 649 653 705 718 719 723 738 748 751 760 768 785 788
[39] 827 840 849 850 906 922 934 950 952 983 995 998

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

[1] 5 19 33 60 67 83 172 190 195 233 242 266 274 283 303 343 352 388 437
[20] 442 457 489 518 520 530 535 584 588 622 649 653 665 705 713 718 736 738 751
[39] 760 768 771 785 788 809 827 837 840 849 850 870 874 906 934 950 952 971 995
[58] 998

> 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] 3

> 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 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/RtmpsnMeYX/Rinst304134287146d2/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] 64

> sum(diff_results$bootstrap_p<=0.05)

```

```
[1] 41
```

```
> 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] 3
```

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

```
[1] 0
```

```
> 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[diff_list_perm, ])
> print(sig_results_perm)
```

	IlmnID	Beta	exmdata2[, 2]	exmdata3[, 2]	exmdata4[, 2]
97	cg00083937	0.53046980	0.60529020	0.62733150	0.65623920
627	cg00612467	0.04777553	0.03783457	0.05380982	0.05582291
764	cg00730260	0.90471270	0.90542290	0.91002680	0.91258610
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
97	0.55974270	0.43157020	0.64046990	0.5787699	
627	0.04740551	0.05332965	0.05775211	0.0557971	
764	0.90575890	0.88760470	0.90756300	0.9094679	
	diff_results\$ordfit_t[diff_list_perm]				
97	-2.665377				
627	-1.797392				
764	-1.560713				
	diff_results\$permutation_p[diff_list_perm]				
97	0				
627	0				
764	0				

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