

Introduction to RBM package

Dongmei Li

January 22, 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] 37

> which(myresult$permutation_p<=0.05)

[1] 23 39 61 64 91 146 196 253 274 364 367 377 398 445 493 513 520 535 568
[20] 603 611 631 635 643 679 684 696 716 743 758 834 838 859 922 926 957 999

> sum(myresult$bootstrap_p<=0.05)

[1] 2

> which(myresult$bootstrap_p<=0.05)

[1] 225 817

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

[1] 4

> 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] 3

> which(myresult2$bootstrap_p<=0.05)

[1] 188 324 368

> 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] 72

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

[1] 80

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

[1] 75

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

[1]  30  35  36  39  66  83 138 141 197 245 250 251 273 276 277 282 290 301 334
[20] 345 353 359 370 415 453 460 471 482 493 494 503 508 535 537 572 573 577 579
[39] 583 595 617 653 659 677 682 685 733 738 739 757 764 768 796 799 803 811 816
[58] 829 840 845 856 874 895 896 907 914 928 942 943 964 965 976

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

[1]  30  36  37  63  83 105 138 141 197 245 250 251 273 277 282 290 301 334 345
[20] 353 359 370 385 411 413 415 416 453 460 482 493 494 503 508 535 537 554 556
[39] 572 573 577 579 583 595 597 617 648 653 659 677 682 685 717 738 739 757 764
[58] 780 785 796 799 803 811 812 816 829 840 845 847 874 895 896 907 928 942 943
[77] 964 965 976 994

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

[1]  30  36  39  45  55  66  83 100 105 138 141 167 197 245 251 273 277 282 290
[20] 301 334 345 353 359 370 411 415 416 453 460 482 483 494 503 508 535 537 560
[39] 572 577 579 583 595 617 653 659 664 677 682 685 693 725 738 739 757 764 796
[58] 799 803 811 812 816 829 840 847 874 895 896 907 914 928 943 964 965 994

```

```

> 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] 18

> 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] 83 141 251 273 277 290 334 353 535 537 577 659 682 739 799 840 943 965

> which(con3_adjp<=0.05/3)

[1] 30 141 251 290 370 415 577 583 659 757 799 811 816 895 965

> 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] 62

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

[1] 57

```

```

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

[1]      3      8     31     51     68     71     85    137    157    158    207    208    222    286    290
[16]   293   302   351   364   372   373   387   393   417   422   425   427   447   449   465
[31]   470   501   532   538   545   566   570   585   599   614   619   627   629   631   639
[46]   664   676   677   696   757   764   789   804   807   808   825   868   908   932   942
[61]   952   956   961   976   997  1000

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

[1]      3      8     31     51     68     71     85    121    132    137    157    158    207    208    216
[16]   222   267   286   290   293   302   334   387   422   425   447   449   465   489   501
[31]   538   545   556   566   570   585   592   599   626   627   629   631   639   664   676
[46]   677   696   757   764   789   804   825   868   908   932   942   952   956   973   976
[61]   997  1000

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

[1]      8     32     51    132    133    137    157    170    178    208    222    267    286    293    302
[16]   351    373    387    393    406    425    447    449    465    470    500    501    528    538    545
[31]   556    566    570    614    619    627    631    639    664    677    696    754    757    789    797
[46]   804    808    868    932    942    952    956    959    961    976    997  1000

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

[1] 8

> 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] 9

```

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/RtmpzMLuW0/Rinst2acfc045c/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] 54

> sum(diff_results$bootstrap_p<=0.05)

[1] 69

> 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] 12

> 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]
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
285    0.11693600    0.10650430    0.12281160    0.12310430
627    0.04740551    0.05332965    0.05775211    0.05579710
928    0.04490690    0.04204062    0.05050039    0.05268215
      diff_results$ordfit_t[diff_list_perm]
285                                -2.993292
627                                -1.797392
928                                -1.982308
      diff_results$permutation_p[diff_list_perm]
285                                0
627                                0
928                                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]
95	cg00081975	0.03633894	0.04975194	0.06024723	0.05598723
131	cg00121904	0.15449580	0.17949750	0.23608110	0.24354150
146	cg00134539	0.61101320	0.53321780	0.45999340	0.46787420
258	cg00234616	0.06639040	0.14705640	0.18254770	0.19942150
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565
285	cg00263760	0.09050395	0.10197760	0.14801710	0.12242400
482	cg00468146	0.11144740	0.15416650	0.19827990	0.18517240
632	cg00615377	0.11265030	0.16140570	0.19404450	0.17468600
677	cg00651216	0.06825629	0.12529090	0.14409190	0.13907250
743	cg00717862	0.07999436	0.07873347	0.06089359	0.06171374
848	cg00826384	0.05721674	0.05612171	0.06644259	0.06358381
979	cg00945507	0.13432250	0.23854600	0.34749760	0.28903340
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
95	0.04561792	0.05115624	0.06068253	0.06168212	
131	0.17352980	0.12564280	0.18193170	0.20847670	
146	0.67191510	0.63137380	0.47929610	0.45428300	
258	0.10620550	0.11668540	0.12630260	0.19163650	
259	0.04030003	0.03996053	0.05086962	0.05445672	
285	0.11693600	0.10650430	0.12281160	0.12310430	
482	0.12285820	0.13271110	0.14196260	0.22159420	
632	0.12573100	0.14483660	0.16338240	0.20130510	
677	0.07669587	0.09597587	0.11690440	0.15194540	
743	0.07594936	0.09062161	0.06475791	0.07271878	
848	0.05230160	0.06119713	0.06542751	0.06240686	
979	0.11848510	0.16653850	0.30718420	0.26624740	
	diff_results\$ordfit_t[diff_list_boot]				
95	-2.654324				
131	-3.562745				
146	5.636263				
258	-3.168405				
259	-2.833203				
285	-2.993292				
482	-3.318963				
632	-3.722206				
677	-3.457874				
743	2.918806				
848	-1.687144				
979	-4.968792				
	diff_results\$bootstrap_p[diff_list_boot]				
95	0				
131	0				
146	0				

258	0
259	0
285	0
482	0
632	0
677	0
743	0
848	0
979	0