

Introduction to RBM package

Dongmei Li

April 15, 2025

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] 51

> which(myresult$permutation_p<=0.05)

[1] 33 34 46 69 74 99 103 132 135 144 186 192 211 230 319 328 350 388 394
[20] 399 412 446 447 458 515 536 545 559 569 579 605 608 628 636 652 732 736 763
[39] 785 807 838 869 870 877 890 907 938 971 978 996 998

> sum(myresult$bootstrap_p<=0.05)

[1] 15

> which(myresult$bootstrap_p<=0.05)

[1] 10 62 230 328 359 388 473 481 517 519 603 636 655 732 826

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

[1] 10

> 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] 28

> which(myresult2$bootstrap_p<=0.05)

[1] 10 43 50 121 128 146 195 197 222 248 267 274 276 381 386
[16] 404 468 520 612 621 759 845 850 888 911 978 997 1000

> bootstrap2_adj_p <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adj_p<=0.05)

[1] 1

```

- 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] 37

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

[1] 38

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

[1] 49

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

[1] 30 36 39 40 42 89 112 146 178 230 232 289 328 334 354 375 378 477 575
[20] 629 661 676 720 727 762 764 766 819 866 868 889 891 907 918 926 939 990

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

[1] 22 26 30 36 39 40 77 89 112 115 146 178 230 232 289 320 328 354 375
[20] 378 465 477 575 676 720 727 762 764 766 772 866 867 889 891 918 926 939 984

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

[1] 26 30 36 39 40 42 46 77 85 89 95 112 146 160 178 230 232 264 289
[20] 328 330 354 355 375 378 408 465 477 487 552 575 629 661 676 720 727 764 766
[39] 772 854 866 867 889 891 918 926 939 980 990

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

```

[1] 5

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

[1] 5

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

[1] 6

> which(con2_adjp<=0.05/3)

[1] 39 112 230 720 764

> which(con3_adjp<=0.05/3)

[1] 39 40 112 230 354 764

> 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] 48

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

[1] 49

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

[1] 56

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

```

```

[1] 2 10 12 30 43 48 59 70 142 152 163 264 279 300 357 440 450 453 455
[20] 461 472 474 479 543 594 616 629 637 658 664 671 677 723 726 732 735 762 795
[39] 856 859 863 901 902 908 950 958 987 998

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

[1] 10 12 48 59 70 74 89 142 162 163 264 300 324 349 353 357 372 404 440
[20] 450 453 455 456 461 472 474 483 548 594 605 629 637 671 706 723 726 732 735
[39] 762 795 810 859 875 896 901 902 908 950 987

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

[1] 10 43 59 70 81 89 142 146 162 163 256 264 300 353 357 404 440 450 453
[20] 455 461 472 474 479 483 548 558 594 595 605 629 637 658 664 666 667 671 677
[39] 723 732 735 762 795 800 844 856 859 863 873 880 901 902 908 930 958 987

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

[1] 4

> 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] 4

```

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] "E:/biocbuild/bbs-3.21-bioc/tmpdir/RtmpWso8UE/Rinst1ce84c02638/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] 58
```

```
> sum(diff_results$bootstrap_p<=0.05)
```

```
[1] 79
```

```
> 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] 3
```

```
> boot_adj_p <- p.adjust(diff_results$bootstrap_p, "BH")
```

```
> sum(boot_adj_p<=0.05)
```

```
[1] 14
```

```
> 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
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]	
280	0.61920530	0.61925200	0.46753250	0.55632410	
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]				
280	4.337628				
627	-1.797392				
928	-1.982308				
	diff_results\$permutation_p[diff_list_perm]				
280	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
146	cg00134539	0.61101320	0.53321780	0.45999340	0.46787420
200	cg00183916	0.03525946	0.03984548	0.02765822	0.02789838
245	cg00224508	0.04479948	0.04972043	0.04152814	0.04189373
252	cg00230502	0.10061390	0.13517870	0.12538510	0.16304920
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
397	cg00394658	0.27940900	0.40410330	0.40262320	0.44339290
437	cg00424946	0.04122172	0.04325330	0.03339863	0.02876798
754	cg00725777	0.84394460	0.81308960	0.81177570	0.72432230
833	cg00814580	0.09348613	0.09619816	0.12010440	0.11534240
837	cg00816620	0.49801980	0.49442680	0.48261920	0.43686010
928	cg00901493	0.03737166	0.03903724	0.04684618	0.04981432
931	cg00901704	0.05734342	0.04812868	0.04478214	0.03878488
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	
146	0.67191510	0.63137380	0.47929610	0.45428300	
200	0.03034811	0.04302129	0.02753873	0.03067437	
245	0.04208405	0.05284988	0.03775905	0.03955271	
252	0.11970870	0.12036160	0.17423730	0.18155480	
280	0.61920530	0.61925200	0.46753250	0.55632410	
397	0.35626060	0.23388380	0.41974630	0.45806880	
437	0.03353116	0.03719167	0.03096761	0.03234779	
754	0.80431430	0.79096990	0.69559410	0.74409820	
833	0.09577040	0.11598850	0.12860890	0.14111200	
837	0.48249620	0.47616240	0.37076920	0.45801530	
928	0.04490690	0.04204062	0.05050039	0.05268215	
931	0.04497277	0.05751033	0.03089829	0.04423603	
979	0.11848510	0.16653850	0.30718420	0.26624740	
	diff_results\$ordfit_t[diff_list_boot]				
95	-2.654324				
146	5.636263				
200	1.765536				
245	1.494678				
252	-3.204466				
280	4.337628				
397	-3.219874				
437	1.574598				
754	2.894050				
833	-3.278186				
837	2.340811				
928	-1.982308				
931	2.127264				
979	-4.968792				

```
diff_results$bootstrap_p[diff_list_boot]
95 0
146 0
200 0
245 0
252 0
280 0
397 0
437 0
754 0
833 0
837 0
928 0
931 0
979 0
```