

Introduction to RBM package

Dongmei Li

November 12, 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] 43

> which(myresult$permutation_p<=0.05)

[1] 9 30 63 97 101 113 153 180 224 264 279 283 284 298 304 375 487 551 574
[20] 576 586 595 600 639 658 660 664 690 721 751 754 773 777 781 826 829 844 849
[39] 855 885 951 968 999

> sum(myresult$bootstrap_p<=0.05)

[1] 21

> which(myresult$bootstrap_p<=0.05)

[1] 60 224 375 457 478 518 600 660 664 687 690 699 781 804 826 829 849 862 888
[20] 998 999

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

[1] 0

> 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] 7

> which(myresult2$bootstrap_p<=0.05)

[1] 40 76 473 574 575 990 998

> 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] 55

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

[1] 53

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

[1] 45

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

[1] 60 103 130 135 155 159 165 193 205 232 277 288 295 327 362 377 396 407 410
[20] 422 449 471 481 493 497 505 515 541 548 575 584 593 604 632 685 704 706 710
[39] 738 740 747 750 751 757 763 778 797 823 825 845 873 886 889 892 943

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

[1] 58 60 103 123 130 159 165 205 232 275 327 362 377 396 407 410 414 422 443
[20] 449 471 481 493 497 505 541 575 584 604 607 632 685 704 706 710 738 740 747
[39] 750 751 757 763 778 797 823 825 845 873 886 889 892 901 907

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

[1] 60 103 123 130 159 165 205 232 275 277 295 327 362 396 414 422 443 449 465
[20] 467 471 481 497 505 541 575 584 607 632 685 698 706 710 738 747 751 754 757
[39] 763 825 845 873 886 889 943
```

```

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

[1] 11

> 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] 9

> which(con2_adjp<=0.05/3)

integer(0)

> which(con3_adjp<=0.05/3)

[1] 60 327 362 396 449 541 575 584 738

> 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] 58

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

[1] 63

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

[1] 65

```

```

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

[1] 7 153 169 178 185 211 225 229 261 266 315 322 351 354 366 371 379 400 405
[20] 441 460 466 476 506 515 555 562 569 580 588 596 607 614 635 649 685 692 712
[39] 716 736 739 743 754 770 772 788 797 798 803 811 816 861 875 901 949 970 978
[58] 987

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

[1] 7 19 23 142 178 185 211 225 229 240 261 315 351 354 366 371 396 400 405
[20] 460 466 474 476 478 485 506 511 515 555 562 569 580 588 596 607 614 635 649
[39] 659 685 692 707 712 716 724 736 739 743 754 772 773 788 789 798 803 811 861
[58] 875 901 949 970 978 987

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

[1] 7 19 23 99 142 153 178 185 211 225 229 261 266 315 322 351 354 365 366
[20] 371 384 396 400 441 460 466 476 506 515 555 562 569 580 583 588 596 607 614
[39] 635 649 659 685 692 696 712 716 736 739 743 754 772 788 789 797 798 803 816
[58] 861 863 875 901 949 970 978 987

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

[1] 12

> 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] 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] "/private/var/folders/r0/l4fjk6cj5xj0j3brt4bplpl40000gt/T/RtmpdxnLgS/Rinstf71329f06ac8/RBM/d
```

```
> 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] 51

> 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] 3

> 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]	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	diff_results\$ordfit_t[diff_list_perm]	diff_results\$permutation_p[diff_list_perm]
103	cg00094319	0.73784280	0.73532960	0.75574900	0.73830220	0.67349260	0.73510200	0.75715920	0.7898122	-2.343784	0
627	cg00612467	0.04777553	0.03783457	0.05380982	0.05582291	0.04740551	0.05332965	0.05775211	0.0557971	-1.797392	0
851	cg00830029	0.58362500	0.59397870	0.64739610	0.67269640	0.50820240	0.34657470	0.66276570	0.6463451	-2.986319	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]
259	cg00234961	0.04192170	0.04321576	0.0570714	0.05327565
280	cg00260778	0.64319890	0.60488960	0.5673506	0.53150910
911	cg00888479	0.07388961	0.07361080	0.1014980	0.09985076
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
259	0.04030003	0.03996053	0.05086962	0.05445672	
280	0.61920530	0.61925200	0.46753250	0.55632410	
911	0.08633986	0.06765189	0.09070268	0.12417730	
	diff_results\$ordfit_t[diff_list_boot]				
259	-2.833203				
280	4.337628				
911	-3.490240				
	diff_results\$bootstrap_p[diff_list_boot]				
259	0				
280	0				
911	0				