

# Introduction to RBM package

Dongmei Li

April 25, 2023

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
<code>ordfit_t</code>	1000	-none-	numeric
<code>ordfit_pvalue</code>	1000	-none-	numeric
<code>ordfit_beta0</code>	1000	-none-	numeric
<code>ordfit_beta1</code>	1000	-none-	numeric
<code>permutation_p</code>	1000	-none-	numeric
<code>bootstrap_p</code>	1000	-none-	numeric

```
> sum(myresult$permutation_p<=0.05)
```

```

[1] 22

> which(myresult$permutation_p<=0.05)

[1] 21 39 47 96 107 130 195 201 270 318 452 478 525 577 629 635 684 769 775
[20] 827 835 877

> sum(myresult$bootstrap_p<=0.05)

[1] 10

> which(myresult$bootstrap_p<=0.05)

[1] 39 48 80 130 337 355 461 622 647 684

> 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] 24

> which(myresult2$bootstrap_p<=0.05)

[1] 85 99 104 108 152 197 226 312 324 352 362 367 415 499 592 659 680 692 722
[20] 872 879 914 950 982

> 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] 56

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

[1] 60

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

[1] 51

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

 [1]  38  81  85  97 113 179 189 228 229 230 232 258 264 295 296 306 309 342 343
[20] 348 364 376 380 386 389 394 398 400 403 426 462 464 466 483 492 518 535 587
[39] 594 598 632 662 707 708 712 721 724 734 865 871 873 908 927 932 940 973

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

 [1]  38  59  81  85  97 111 113 143 179 189 213 228 229 260 264 295 296 303 306
[20] 342 348 364 375 376 380 389 391 394 398 400 403 426 454 464 466 472 483 492
[39] 535 577 581 587 594 598 632 662 663 702 707 708 712 721 724 734 833 856 865
[58] 873 932 973

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

 [1]  38  49  63  72  85  97 113 135 143 179 189 213 229 260 264 296 303 306 342
[20] 343 348 364 375 376 380 389 398 400 426 462 466 472 483 492 535 594 598 632
[39] 662 707 708 712 721 724 734 811 865 873 932 940 973

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

[1] 8

```

```

> con2_adj_p <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adj_p<=0.05/3)

[1] 13

> con3_adj_p <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adj_p<=0.05/3)

[1] 7

> which(con2_adj_p<=0.05/3)

[1] 97 296 306 342 348 464 466 594 598 707 712 865 873

> which(con3_adj_p<=0.05/3)

[1] 296 306 389 598 707 712 973

> 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] 56

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

[1] 45

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

[1] 56

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

[1] 27 41 47 59 62 63 65 69 100 114 154 155 173 211 215 275 317 346 347
[20] 365 404 411 416 421 473 488 493 501 509 523 545 546 602 603 643 665 705 710
[39] 719 731 750 763 809 847 874 876 885 896 906 911 938 967 984 985 989 991

```

```

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

[1] 41 65 69 100 114 132 154 173 215 216 275 317 346 347 365 404 408 411 416
[20] 421 488 508 509 523 556 602 606 616 617 643 647 665 710 750 763 847 874 876
[39] 885 896 906 911 967 985 989

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

[1] 41 63 65 69 100 114 132 154 173 194 215 216 265 275 317 346 347 365 404
[20] 411 416 421 473 488 493 501 508 509 523 543 556 565 571 602 616 631 643 665
[39] 673 705 710 750 763 847 855 874 885 896 906 911 938 939 967 977 984 989

> 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] 1

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=0.05/3)

[1] 11

```

## 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/RtmpgTBntn/Rinst1c9e3b6e65c910/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.:0.59032  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] 45

```

```

> sum(diff_results$permutation_p<=0.05)

```

```

[1] 34

```

```

> sum(diff_results$bootstrap_p<=0.05)

```

```
[1] 50
```

```
> 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] 2
```

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

```
[1] 3
```

```
> 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]
245	cg00224508	0.04479948	0.04972043	0.04152814	0.04189373
764	cg00730260	0.90471270	0.90542290	0.91002680	0.91258610
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
245	0.04208405	0.05284988	0.03775905	0.03955271	
764	0.90575890	0.88760470	0.90756300	0.90946790	
	diff_results\$ordfit_t[diff_list_perm]				
245	1.962457				
764	-1.808081				
	diff_results\$permutation_p[diff_list_perm]				
245	0				
764	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
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565
911	cg00888479	0.07388961	0.07361080	0.10149800	0.09985076
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
95	0.04561792	0.05115624	0.06068253	0.06168212	
259	0.04030003	0.03996053	0.05086962	0.05445672	
911	0.08633986	0.06765189	0.09070268	0.12417730	
	diff_results\$ordfit_t[diff_list_boot]				

95	-3.252063
259	-4.052697
911	-3.621731
diff_results\$bootstrap_p[diff_list_boot]	
95	0
259	0
911	0