

Introduction to RBM package

Dongmei Li

November 3, 2024

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] 40

> which(myresult$permutation_p<=0.05)

[1] 2 61 74 84 140 186 212 235 273 307 316 337 357 389 424 440 449 455 478
[20] 529 540 548 551 589 613 615 655 671 703 747 751 815 820 840 873 877 895 916
[39] 936 978

> sum(myresult$bootstrap_p<=0.05)

[1] 7

> which(myresult$bootstrap_p<=0.05)

[1] 84 180 205 548 678 704 830

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

[1] 6

> 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] 29

> which(myresult2$bootstrap_p<=0.05)

[1] 47 99 108 116 139 170 183 191 193 244 308 317 371 380 485 492 609 612 646
[20] 662 784 786 823 824 879 900 930 976 986

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

[1] 3

```

- 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] 80

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

[1] 72

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

[1] 67

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

[1] 16 61 65 77 79 88 107 142 155 173 182 208 227 255 256 292 301 303 310
[20] 327 330 336 373 388 389 393 394 395 412 418 420 423 434 442 443 451 487 491
[39] 495 504 510 541 565 567 573 585 593 595 637 667 690 705 713 737 740 745 759
[58] 776 787 794 805 810 824 825 827 828 836 839 849 883 884 892 895 901 902 919
[77] 922 937 971 985

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

[1] 16 61 65 68 77 79 83 88 98 107 142 173 207 208 238 255 292 301 303
[20] 310 321 327 330 336 388 389 393 394 412 418 420 434 442 443 487 491 495 504
[39] 510 565 567 595 690 724 737 740 745 750 759 767 787 794 805 810 824 825 828
[58] 836 839 849 882 884 891 892 895 901 902 919 922 937 983 985

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

```

[1] 16 61 65 77 79 88 98 107 142 173 208 213 255 256 292 301 303 321 327
[20] 330 373 388 389 393 394 395 412 418 420 434 442 443 487 491 495 504 510 565
[39] 567 593 637 663 689 690 724 737 740 745 759 787 805 825 827 828 836 839 849
[58] 883 884 891 892 895 901 919 922 937 985

```

```

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

```

```

[1] 14

```

```

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

```

```

[1] 4

```

```

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

```

```

[1] 8

```

```

> which(con2_adj_p<=0.05/3)

```

```

[1] 487 565 825 891

```

```

> which(con3_adj_p<=0.05/3)

```

```

[1] 16 77 412 442 510 565 759 901

```

```

> 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] 80

```

```

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

```

```

[1] 57

```

```

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

[1] 67

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

[1] 23 32 36 42 52 62 81 89 115 119 124 129 162 174 187 191 195 197 225
[20] 229 236 257 267 283 290 296 304 336 340 342 343 348 355 357 371 373 375 389
[39] 405 414 443 444 455 473 481 492 528 532 534 538 569 579 585 589 654 657 665
[58] 699 705 726 731 732 738 771 789 802 807 824 833 847 857 858 859 868 876 938
[77] 976 982 987 997

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

[1] 23 32 42 52 81 89 115 124 129 162 174 187 191 229 257 267 283 340 342
[20] 343 348 352 357 373 375 389 414 473 528 532 538 569 579 585 589 639 654 665
[39] 699 731 738 739 771 789 802 807 823 824 847 858 868 873 938 976 982 987 997

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

[1] 23 32 42 52 62 81 89 115 119 124 129 162 174 187 225 230 236 257 263
[20] 267 296 304 340 343 348 352 357 371 373 375 389 405 414 444 455 473 475 492
[39] 528 532 534 538 569 579 585 589 654 665 705 726 738 739 771 789 802 807 824
[58] 833 847 858 868 876 898 916 976 982 997

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

[1] 18

> 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] 7

```

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/Rtmpu6i0GV/Rinsta2e046d93e2a/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] 47

> sum(diff_results$permutation_p<=0.05)

[1] 58

> sum(diff_results$bootstrap_p<=0.05)

[1] 40

> 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] 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)
> print(sig_results_perm)

      IlmnID      Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
103 cg00094319 0.7378428      0.7353296      0.7557490      0.7383022
851 cg00830029 0.5836250      0.5939787      0.6473961      0.6726964
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
103      0.6734926      0.7351020      0.7571592      0.7898122
851      0.5082024      0.3465747      0.6627657      0.6463451
      diff_results$ordfit_t[diff_list_perm]
103                                -2.343784
851                                -2.986319
      diff_results$permutation_p[diff_list_perm]
103                                           0
851                                           0

```



```

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

```