

Introduction to RBM package

Dongmei Li

November 11, 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] 32

> which(myresult$permutation_p<=0.05)

[1] 6 118 119 135 148 208 274 285 292 323 339 468 512 530 534 565 586 602 608
[20] 629 655 685 688 720 724 757 830 886 891 926 934 958

> sum(myresult$bootstrap_p<=0.05)

[1] 3

> which(myresult$bootstrap_p<=0.05)

[1] 98 367 629

> 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] 22

> which(myresult2$bootstrap_p<=0.05)

[1] 30 41 49 50 179 192 261 393 594 603 652 663 711 756 768 887 890 903 904
[20] 930 951 972

> 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] 59

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

[1] 56

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

[1] 35 91 100 110 135 162 191 197 201 215 236 238 271 284 332 337 348 349 356
[20] 366 370 387 400 430 439 454 505 553 567 580 595 607 623 632 633 642 666 667
[39] 668 692 697 760 846 854 857 858 871 880 881 887 915 926 931 947 994

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

[1] 35 54 77 86 91 100 110 135 162 179 191 197 201 236 238 284 311 332 337
[20] 348 349 356 366 370 387 430 439 454 505 543 547 553 567 580 595 607 623 632
[39] 633 642 666 667 668 692 697 760 825 854 857 858 871 880 881 887 915 926 931
[58] 947 994

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

[1] 35 86 91 97 100 110 162 191 197 201 215 236 238 251 252 332 337 348 349
[20] 356 370 387 430 437 439 454 567 573 580 591 607 608 623 632 633 635 642 652
[39] 655 667 668 692 697 717 745 760 825 846 854 871 880 887 915 920 961 994

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

[1] 7

```

```

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

[1] 10

> 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] 100 110 454 580 595 607 632 668 692 697

> which(con3_adjp<=0.05/3)

[1] 100 110 430 580 607 668

> 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] 58

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

[1] 62

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

[1] 10 27 44 53 61 62 79 94 103 125 135 151 165 179 196
[16] 197 216 224 245 249 272 274 283 304 314 321 322 324 391 418
[31] 421 449 461 469 489 497 508 534 548 554 562 591 625 632 641
[46] 659 673 696 706 712 713 731 780 812 823 835 839 861 898 900
[61] 912 916 931 966 999 1000

```

```

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

[1] 8 27 53 61 62 79 94 103 125 135 151 165 179 196 216 272 274 283 304
[20] 314 322 391 418 421 441 449 461 489 497 508 534 547 548 562 591 610 625 632
[39] 653 673 696 706 712 713 731 780 812 823 839 854 858 861 898 900 912 916 966
[58] 999

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

[1] 8 10 27 44 53 61 62 79 94 103 125 151 165 179 193
[16] 196 216 272 274 283 304 314 321 322 365 391 418 421 449 461
[31] 489 508 534 548 554 562 577 591 610 625 632 652 696 706 713
[46] 731 780 812 823 835 839 846 854 861 898 900 912 916 931 966
[61] 999 1000

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

[1] 11

> con22_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adj_p<=0.05/3)

[1] 7

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

[1] 10

```

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 gemone-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/Rtmp7JZVje/Rinst3be173e5ac/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] 85

```

```

> sum(diff_results$bootstrap_p<=0.05)

[1] 39

> 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] 12

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

[1] 1

> 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]
19	cg00016968	0.80628480	NA	0.81440820	0.83623180
103	cg00094319	0.73784280	0.73532960	0.75574900	0.73830220
106	cg00095674	0.07076291	0.05045181	0.03861991	0.03337576
245	cg00224508	0.04479948	0.04972043	0.04152814	0.04189373
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565
627	cg00612467	0.04777553	0.03783457	0.05380982	0.05582291
764	cg00730260	0.90471270	0.90542290	0.91002680	0.91258610
848	cg00826384	0.05721674	0.05612171	0.06644259	0.06358381
851	cg00830029	0.58362500	0.59397870	0.64739610	0.67269640
887	cg00862290	0.43640520	0.54047160	0.60786800	0.56325950
911	cg00888479	0.07388961	0.07361080	0.10149800	0.09985076
928	cg00901493	0.03737166	0.03903724	0.04684618	0.04981432
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
19	0.80831380	0.73306440	0.82968340	0.84917800	
103	0.67349260	0.73510200	0.75715920	0.78981220	
106	0.04693030	0.06837343	0.04534005	0.03709488	
245	0.04208405	0.05284988	0.03775905	0.03955271	
259	0.04030003	0.03996053	0.05086962	0.05445672	
627	0.04740551	0.05332965	0.05775211	0.05579710	
764	0.90575890	0.88760470	0.90756300	0.90946790	
848	0.05230160	0.06119713	0.06542751	0.06240686	
851	0.50820240	0.34657470	0.66276570	0.64634510	


```

887      0.50259740      0.40111730      0.56646700      0.54552980
911      0.08633986      0.06765189      0.09070268      0.12417730
928      0.04490690      0.04204062      0.05050039      0.05268215

```

```
diff_results$ordfit_t[diff_list_perm]
```

```

19      -2.547097
103     -2.343784
106      2.887876
245      1.494678
259     -2.833203
627     -1.797392
764     -1.560713
848     -1.687144
851     -2.986319
887     -3.368752
911     -3.490240
928     -1.982308

```

```
diff_results$permutation_p[diff_list_perm]
```

```

19      0
103     0
106     0
245     0
259     0
627     0
764     0
848     0
851     0
887     0
911     0
928     0

```

```

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

```

```

      IlmnID      Beta exmdata2[, 2] exmdata3[, 2] exmdata4[, 2]
979 cg00945507 0.1343225      0.238546      0.3474976      0.2890334
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
979      0.1184851      0.1665385      0.3071842      0.2662474
diff_results$ordfit_t[diff_list_boot]
979      -4.968792
diff_results$bootstrap_p[diff_list_boot]
979      0

```