

Introduction to RBM package

Dongmei Li

May 18, 2026

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] 22

> which(myresult$permutation_p<=0.05)

[1] 50 72 118 144 184 286 290 375 476 567 568 572 689 738 771 793 808 830 836
[20] 841 890 974

> sum(myresult$bootstrap_p<=0.05)

[1] 1

> which(myresult$bootstrap_p<=0.05)

[1] 790

> 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] 23

> which(myresult2$bootstrap_p<=0.05)

[1] 78 87 155 171 217 241 272 315 348 379 489 496 536 563 565 694 777 806 826
[20] 918 940 947 992

> 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] 77

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

[1] 66

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

[1] 71

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

[1] 20 32 64 66 77 79 84 132 140 142 148 151 155 174 179 193 199 200 239
[20] 242 259 264 268 313 317 348 367 368 384 391 403 418 443 448 459 472 473 475
[39] 479 496 501 509 533 556 588 619 642 667 689 694 710 711 717 741 743 746 766
[58] 767 769 788 798 800 821 822 845 854 856 873 875 879 887 907 936 945 955 977
[77] 997

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

[1] 11 20 30 32 42 66 77 84 95 100 148 155 179 193 199 200 239 242 259
[20] 264 268 295 313 317 348 367 368 391 403 443 448 472 473 475 479 496 501 509
[39] 533 556 616 619 642 667 689 694 702 710 717 741 743 767 769 788 798 800 845
[58] 856 859 873 879 880 885 936 945 997

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

[1] 11 20 32 64 66 77 84 140 148 155 174 193 199 200 239 242 259 268 295
[20] 311 313 317 348 367 368 384 391 403 443 448 459 472 473 475 479 496 501 509
[39] 533 556 588 616 619 642 667 689 694 702 710 711 717 741 743 766 767 769 788
[58] 789 798 800 845 856 873 875 879 880 907 936 945 977 997

```

```

> 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] 8

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

[1] 17

> which(con2_adjp<=0.05/3)

[1] 239 242 313 367 368 710 769 788

> which(con3_adjp<=0.05/3)

[1] 20 77 199 242 313 317 391 448 473 619 667 710 741 767 788 798 997

> 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] 47

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

[1] 55

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

[1] 53

```

```

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

[1] 22 70 79 88 93 110 113 117 124 145 150 162 169 179 206 229 259 278 285
[20] 304 340 351 391 397 421 426 528 539 554 562 580 640 657 729 737 772 854 856
[39] 871 876 915 920 925 937 941 960 986

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

[1] 79 88 110 113 124 145 150 152 162 176 179 206 229 272 278 285 304 340 351
[20] 390 391 397 401 421 425 426 427 450 460 528 539 554 562 563 607 637 640 657
[39] 679 714 723 729 737 788 854 856 876 915 920 921 925 937 941 960 986

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

[1] 70 79 88 93 110 113 117 124 145 150 162 176 179 206 229 237 239 272 278
[20] 304 319 333 340 351 372 391 401 421 426 481 528 539 554 562 563 580 657 679
[39] 714 729 737 763 825 854 856 871 876 915 920 925 941 960 986

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

[1] 5

> 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] 5

```

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/RtmpEkC7Ls/Rinst2f68243b4916/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                      NAs      :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
                      NAs      :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

```



```

851             -2.986319
887             -3.368752
      diff_results$permutation_p[diff_list_perm]
19             0
259            0
764            0
848            0
851            0
887            0

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

```