

# Introduction to RBM package

Dongmei Li

June 4, 2025

Clinical and Translational Science Institute, University of Rochester School of Medicine and  
Dentistry, Rochester, NY 14642-0708

## Contents

<b>1</b>	<b>Overview</b>	<b>1</b>
<b>2</b>	<b>Getting started</b>	<b>2</b>
<b>3</b>	<b>RBM_T and RBM_F functions</b>	<b>2</b>
<b>4</b>	<b>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
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] 30

> which(myresult$permutation_p<=0.05)

[1] 12 17 141 161 210 288 297 303 304 309 312 355 373 386 398 418 446 508 689
[20] 740 761 765 796 802 810 816 818 926 963 989

> sum(myresult$bootstrap_p<=0.05)

[1] 7

> which(myresult$bootstrap_p<=0.05)

[1] 116 309 509 746 810 893 990

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

[1] 1

> 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] 20

> which(myresult2$bootstrap_p<=0.05)

[1] 61 92 112 132 179 237 246 247 485 560 618 651 695 731 746 831 839 899 977
[20] 981

> 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] 64

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

[1] 73

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

[1] 64

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

[1] 14 16 19 49 62 75 80 97 105 107 112 116 118 146 177 186 190 202 221
[20] 261 262 269 303 313 320 323 336 387 392 394 406 426 431 432 433 480 522 528
[39] 552 560 621 646 658 670 691 693 707 714 735 762 779 790 819 834 856 879 907
[58] 918 920 928 932 940 973 982

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

[1] 14 16 19 34 49 80 105 107 112 116 118 129 134 146 169 174 186 190 202
[20] 214 221 261 262 269 282 303 313 320 323 336 383 387 392 394 406 426 428 431
[39] 432 433 437 480 522 528 552 560 569 613 621 646 667 670 691 693 707 714 735
[58] 763 776 779 790 796 832 834 879 907 918 920 928 932 940 973 982

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

[1] 14 16 34 49 75 80 105 107 112 116 118 134 146 169 186 190 202 221 261
[20] 282 303 313 319 320 323 336 383 387 394 406 426 428 431 432 433 438 480 522
[39] 528 552 560 621 628 646 667 670 691 707 714 735 762 774 779 790 832 834 879
[58] 899 907 920 928 940 973 982

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

```

```

[1] 14

> 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] 13

> which(con2_adjp<=0.05/3)

[1] 19 80 202 261 426 528 621 691 735 940

> which(con3_adjp<=0.05/3)

[1] 105 116 118 202 261 426 528 552 621 691 735 879 940

> 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] 57

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

[1] 46

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

```

```

[1] 18 32 38 69 72 91 119 162 169 176 189 191 226 251 253
[16] 264 290 303 320 336 375 383 393 431 441 491 492 517 518 533
[31] 552 577 589 600 603 624 635 668 682 684 694 699 711 735 741
[46] 744 756 768 786 800 850 857 860 911 924 936 992 1000

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

[1] 18 32 38 69 72 91 119 156 162 169 184 191 226 251 264
[16] 290 297 300 303 336 370 375 393 431 441 491 495 517 518 533
[31] 552 577 603 619 624 635 668 682 684 694 699 711 735 741 744
[46] 768 786 800 850 878 882 911 936 992 995 999 1000

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

[1] 18 32 38 69 71 72 119 169 179 184 191 226 251 303 336
[16] 370 393 431 441 463 491 517 518 533 577 603 619 624 635 668
[31] 682 684 694 743 744 768 786 800 849 850 878 911 936 992 995
[46] 1000

> 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] "F:/biocbuild/bbs-3.22-bioc/tmpdir/RtmpQnLXSD/Rinst1a97c618d497b/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] 47

> sum(diff_results$bootstrap_p<=0.05)

[1] NA

> 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] 1

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

[1] NA

> 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] exmdata5[, 2]
851 cg00830029 0.583625      0.5939787      0.6473961      0.6726964      0.5082024
      exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
851      0.3465747      0.6627657      0.6463451
      diff_results$ordfit_t[diff_list_perm]
851                                     -2.986319
      diff_results$permutation_p[diff_list_perm]
851                                     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]
146 cg00134539 0.6110132      0.5332178      0.4599934      0.4678742
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
146      0.6719151      0.6313738      0.4792961      0.454283
      diff_results$ordfit_t[diff_list_boot]
146                                     5.636263
      diff_results$bootstrap_p[diff_list_boot]
146                                     0

```