

Introduction to RBM package

Dongmei Li

October 24, 2023

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
<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] 29

> which(myresult$permutation_p<=0.05)

[1] 158 188 192 196 306 338 340 344 349 400 442 455 468 548 573 638 659 689 703
[20] 718 765 766 781 800 873 895 903 905 928

> sum(myresult$bootstrap_p<=0.05)

[1] 0

> which(myresult$bootstrap_p<=0.05)

integer(0)

> 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] 24

> which(myresult2$bootstrap_p<=0.05)

[1] 73 89 118 135 214 225 261 292 317 346 363 468 473 506 529 580 595 644 690
[20] 715 840 897 899 908

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

[1] 1

```

- 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] 75

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

[1] 72

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

[1] 65

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

 [1]  4  13  17  25  27  56  60  67  91 112 114 142 171 172 176
[16] 185 186 199 208 226 234 296 318 328 339 348 362 381 393 394
[31] 405 426 451 454 458 462 479 506 522 534 538 619 628 643 647
[46] 649 662 679 687 705 715 716 727 755 757 763 786 788 801 808
[61] 855 863 876 880 882 888 903 907 925 928 929 945 949 968 1000

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

 [1]  4  13  17  25  27  39  60  67  79  91 112 142 164 171 172
[16] 176 184 185 186 208 226 296 318 339 362 381 393 394 405 426
[31] 429 451 458 469 488 506 522 538 539 619 628 638 643 647 649
[46] 662 679 705 715 727 755 757 763 771 786 801 808 845 849 855
[61] 863 880 903 907 925 928 929 930 932 949 968 1000

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

 [1]  12  13  17  25  27  60  67  79  91 112 142 171 172 176 186
[16] 208 226 296 306 339 362 367 381 393 394 405 426 451 454 458
[31] 488 506 522 538 539 619 628 643 647 662 687 705 715 716 727
[46] 755 757 763 786 788 801 855 863 874 880 903 907 925 928 929
[61] 932 949 956 968 1000

```

```

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

[1] 15

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

[1] 16

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

[1] 12

> which(con2_adjp<=0.05/3)

[1] 60 112 142 172 339 362 381 405 727 757 801 863 903 925 949
[16] 1000

> which(con3_adjp<=0.05/3)

[1] 17 27 112 171 339 405 458 727 880 903 925 1000

> 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] 54

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

[1] 53

```

```

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

[1] 1 12 14 17 23 25 26 28 40 61 92 98 123 151 180 182 243 254 261
[20] 274 291 298 333 410 422 432 433 497 504 514 524 566 591 600 605 609 638 664
[39] 669 670 680 742 747 772 785 798 826 833 876 900 901 926 942 948 962 980

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

[1] 12 17 23 25 26 28 61 92 98 123 151 182 217 243 254 261 274 291 402
[20] 422 433 497 504 514 525 591 600 605 609 632 638 646 663 664 669 670 680 681
[39] 696 742 747 772 778 785 798 826 833 876 900 942 944 948 962 980

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

[1] 1 12 17 25 26 28 61 92 98 123 151 180 182 201 212 243 254 261 274
[20] 291 298 333 337 402 422 432 433 497 504 514 520 591 600 605 606 632 638 669
[39] 680 681 696 742 747 826 833 876 900 901 915 932 942 948 962

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

[1] 2

> 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] 2

```

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/Rtmp5vglsl/Rinst1274796d439b22/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] 55

> sum(diff_results$bootstrap_p<=0.05)

[1] 36

> 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] 3

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

[1] 0

> 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]
245 cg00224508 0.04479948 0.04972043 0.04152814 0.04189373
280 cg00260778 0.64319890 0.60488960 0.56735060 0.53150910
627 cg00612467 0.04777553 0.03783457 0.05380982 0.05582291
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
245 0.04208405 0.05284988 0.03775905 0.03955271
280 0.61920530 0.61925200 0.46753250 0.55632410
627 0.04740551 0.05332965 0.05775211 0.05579710
      diff_results$ordfit_t[diff_list_perm]
245 1.962457
280 4.170347
627 -2.239498
      diff_results$permutation_p[diff_list_perm]
245 0
280 0
627 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)
```