Introduction to RBM package

Dongmei Li

April 26, 2023

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

Contents

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 Bejamini-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] 33

> which(myresult$permutation_p<=0.05)

 [1]  33  49  64 146 162 165 214 321 376 400 404 409 462 486 549 617 622 706 715
[20] 726 762 787 796 797 844 849 858 865 902 933 951 957 998

> sum(myresult$bootstrap_p<=0.05)

[1] 0

> which(myresult$bootstrap_p<=0.05)

integer(0)

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

[1] 9

> bootstrap_adjp <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adjp<=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] 33

> which(myresult2$bootstrap_p<=0.05)

 [1]  26  69 158 161 234 236 282 303 320 324 348 416 490 506 546 590 603 614 618
[20] 626 662 680 693 702 721 737 752 833 849 879 894 895 943

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=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] 68

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

[1] 63

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

[1] 55

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

 [1]   5  17  22  23  29  95 102 116 118 127 135 159 165 170 194 201 204 205 207
[20] 212 217 262 263 265 277 319 356 357 358 363 389 426 430 432 445 470 476 505
[39] 608 661 664 682 704 726 730 731 741 773 779 783 794 796 800 801 827 832 842
[58] 866 893 924 933 935 945 947 959 973 982 994

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

 [1]   5  17  22  23  95 116 127 135 140 159 165 170 182 194 204 205 211 212 217
[20] 262 263 265 277 336 358 363 401 426 432 445 470 476 505 519 560 654 690 704
[39] 726 730 731 741 746 773 782 783 796 800 801 827 832 842 866 893 924 933 935
[58] 945 947 959 973 982 994

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

 [1]   5  22  23  34  95 102 116 127 140 159 165 170 182 201 204 205 212 217 262
[20] 263 265 277 358 363 426 432 445 470 476 505 690 704 721 726 730 739 741 773
[39] 779 783 796 800 801 827 842 866 893 924 933 935 945 947 959 982 994

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

```
[1] 13

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

[1] 12

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

[1] 2

> which(con2_adjp<=0.05/3)

 [1] 159 165 358 505 726 801 842 866 924 935 945 959

> which(con3_adjp<=0.05/3)

[1] 159 945

> 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] 60

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

[1] 55

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

[1] 65

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

```
 [1]  43  65  85 115 116 148 154 164 211 212 249 254 258 266 284 289 314 331 336
[20] 343 365 381 397 430 439 440 457 495 554 570 605 616 631 650 652 687 709 712
[39] 737 766 772 791 796 810 848 910 916 928 936 941 954 955 959 960 962 967 972
[58] 976 982 990

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

 [1]  11  43  85 115 116 148 197 212 249 254 266 284 288 289 314 331 336 343 352
[20] 365 381 397 430 440 457 479 495 554 570 605 616 631 635 670 687 693 709 737
[39] 766 791 796 848 875 916 928 936 941 954 959 962 967 972 976 982 990

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

 [1]  43  65  85 115 116 148 154 166 211 212 249 254 266 288 289 316 330 331 336
[20] 343 352 365 376 381 397 430 436 439 440 457 495 554 570 605 616 631 648 652
[39] 687 689 709 712 737 744 745 766 772 791 796 810 848 875 910 916 928 936 941
[58] 954 955 960 962 967 976 982 990

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

[1] 4

> 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] 14
```

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 gemonewide 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 illutration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovariance 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-3.17-bioc/tmpdir/RtmpEDcePd/Rinst28fc28e45bee/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] 52

> sum(diff_results$bootstrap_p<=0.05)

[1] 59

> 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] 8

> 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]
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
245    0.04208405    0.05284988    0.03775905    0.03955271
851    0.50820240    0.34657470    0.66276570    0.64634510
    diff_results$ordfit_t[diff_list_perm]
245                              1.962457
851                             -2.841244
    diff_results$permutation_p[diff_list_perm]
245                                          0
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]
95  cg00081975 0.03633894    0.04975194    0.06024723    0.05598723
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
```

```
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
397 cg00394658 0.27940900    0.40410330    0.40262320    0.44339290
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
95     0.04561792    0.05115624    0.06068253    0.06168212
146    0.67191510    0.63137380    0.47929610    0.45428300
259    0.04030003    0.03996053    0.05086962    0.05445672
285    0.11693600    0.10650430    0.12281160    0.12310430
397    0.35626060    0.23388380    0.41974630    0.45806880
911    0.08633986    0.06765189    0.09070268    0.12417730
928    0.04490690    0.04204062    0.05050039    0.05268215
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
95                               -3.252063
146                               5.394750
259                              -4.052697
285                              -3.093997
397                              -3.070559
911                              -3.621731
928                              -2.716443
979                              -4.750997
    diff_results$bootstrap_p[diff_list_boot]
95                                        0
146                                       0
259                                       0
285                                       0
397                                       0
911                                       0
928                                       0
979                                       0
```