Introduction to RBM package

Dongmei Li

October 30, 2018

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] 24

> which(myresult$permutation_p<=0.05)

 [1]  28  75 111 139 140 150 177 180 184 207 221 310 320 322 328 523 539 582 618
[20] 740 781 793 836 841

> sum(myresult$bootstrap_p<=0.05)

[1] 10

> which(myresult$bootstrap_p<=0.05)

 [1] 139 168 289 435 563 570 601 827 849 939

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

[1] 0

> 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] 15

> which(myresult2$bootstrap_p<=0.05)

 [1] 181 191 333 502 537 539 564 581 664 729 759 809 868 886 916

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=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] 67
```

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

```
[1] 55
```

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

```
[1] 64
```

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

```
 [1]   4   8  16  64  65  79  81  89 155 169 184 216 228 232 248 263 276 307 308
[20] 317 319 344 355 379 387 394 396 411 427 437 441 450 452 459 462 476 480 488
[39] 493 503 504 532 533 561 570 620 622 633 685 717 721 772 781 849 859 863 864
[58] 868 884 888 914 918 928 938 954 961 983
```

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

```
 [1]   8  64  65  79  81 169 184 228 232 248 276 307 319 344 355 394 396 411 427
[20] 439 441 450 452 462 468 476 488 493 503 504 532 533 570 620 622 633 685 689
[39] 694 717 721 772 781 833 859 864 868 888 914 918 919 954 955 961 983
```

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

```
 [1]   8  39  64  65  79  81  83  89 155 169 184 188 228 232 248 266 276 307 317
[20] 319 341 355 387 394 396 411 429 437 441 450 452 462 466 476 488 493 504 521
[39] 532 533 570 616 620 622 633 685 689 693 694 717 721 748 772 781 833 859 864
[58] 868 888 914 954 961 967 983
```

```
> 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] 3

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

[1] 16

> which(con2_adjp<=0.05/3)

[1] 248 493 622

> which(con3_adjp<=0.05/3)

 [1]  81 169 228 248 307 452 488 504 622 633 772 859 914 954 961 983

> 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] 61

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

[1] 57

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

[1] 51

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

```
 [1]  13  16  26  31  33  39  43  53  55  61 114 115 137 165 168 190 209 215 232
[20] 256 296 308 332 334 347 366 383 395 404 409 466 471 489 493 501 554 562 572
[39] 593 602 605 613 637 655 685 726 731 747 751 779 789 804 809 831 834 849 873
[58] 877 948 959 962
```

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

```
 [1]  16  26  31  33  39  53  55  61  75 114 115 137 140 168 181 215 237 279 296
[20] 308 347 366 383 395 404 409 425 441 466 471 489 493 531 554 572 587 593 602
[39] 605 608 613 637 655 672 717 731 747 751 779 789 804 809 849 873 877 948 972
```

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

```
 [1]  13  26  31  33  37  39  48  53  55  61 137 165 168 181 190 192 215 296 298
[20] 308 347 366 383 395 404 409 466 471 501 531 554 572 587 593 605 613 655 698
[39] 726 731 747 751 779 789 809 824 834 849 873 877 948
```

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

```
[1] 11
```

```
> con22_adjp <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adjp<=0.05/3)
```

```
[1] 4
```

```
> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)
```

```
[1] 6
```

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 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] "/private/tmp/RtmpzoN9Vr/Rinstff552cb475d0/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] 48

> sum(diff_results$bootstrap_p<=0.05)

[1] 34

> 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] 3

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

[1] 0

> 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]
103 cg00094319 0.7378428     0.7353296     0.7557490     0.7383022
280 cg00260778 0.6431989     0.6048896     0.5673506     0.5315091
851 cg00830029 0.5836250     0.5939787     0.6473961     0.6726964
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
103     0.6734926     0.7351020     0.7571592     0.7898122
280     0.6192053     0.6192520     0.4675325     0.5563241
851     0.5082024     0.3465747     0.6627657     0.6463451
    diff_results$ordfit_t[diff_list_perm]
103                             -2.268711
280                              4.170347
851                             -2.841244
    diff_results$permutation_p[diff_list_perm]
103                                          0
280                                          0
851                                          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)
```