Introduction to RBM package

Dongmei Li

October 26, 2021

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] 61

> which(myresult$permutation_p<=0.05)

 [1]   1  51  55  69  92 141 152 154 172 176 177 183 193 222 226 227 242 253 286
[20] 294 302 309 320 337 360 380 396 427 452 474 479 487 490 495 503 518 539 544
[39] 547 551 565 577 584 598 622 623 627 683 711 728 740 747 748 757 758 778 808
[58] 835 853 879 911

> sum(myresult$bootstrap_p<=0.05)

[1] 4

> which(myresult$bootstrap_p<=0.05)

[1]  51  79 717 778

> 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] 40

> which(myresult2$bootstrap_p<=0.05)

 [1]    4   42   55   59   65   71   92  142  225  233  238  251  262  288  299
[16]  319  323  328  378  411  421  495  496  556  568  639  645  662  701  702
[31]  706  734  735  806  822  865  887  903  984 1000

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=0.05)

[1] 3
```

- 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] 63

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

[1] 61

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

[1] 69

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

 [1]   2  34  44  70  77 119 140 145 170 190 200 207 208 219 222 231 245 264 265
[20] 281 292 296 313 314 376 396 403 406 415 437 452 477 480 492 507 513 517 527
[39] 542 559 574 591 595 597 612 616 631 633 644 645 720 723 730 731 757 797 816
[58] 834 907 908 912 929 939

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

 [1]   2  34  44  77 119 127 140 170 190 200 208 219 231 245 264 265 281 292 295
[20] 296 313 376 403 406 415 424 437 456 474 477 492 507 527 559 574 587 591 595
[39] 597 612 616 644 645 662 701 720 723 730 731 746 757 816 834 843 854 872 907
[58] 908 912 929 930

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

 [1]   2  30  34  44  47  48  70  77 119 140 170 190 200 207 208 219 231 264 265
[20] 281 292 296 313 336 376 396 403 406 415 437 451 474 477 492 507 513 517 527
[39] 559 574 591 595 597 612 616 631 644 645 646 662 701 720 723 729 730 746 757
[58] 797 816 834 843 861 907 908 912 929 930 942 949
```

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

[1] 9

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

[1] 1

> 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] 591

> which(con3_adjp<=0.05/3)

 [1]  34  44 200 208 219 292 296 415 492 591 597 834

> 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] 89

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

[1] 64

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

[1] 84
```

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

 [1]   4  27  41  45  52  65  72  99 106 114 173 183 234 243 250 268 289 306 313
[20] 320 331 359 365 380 388 390 399 402 417 432 441 444 449 452 466 496 508 510
[39] 517 529 549 553 558 575 577 596 599 600 604 626 630 636 641 649 667 677 709
[58] 721 723 724 726 739 741 757 761 770 773 779 781 787 817 820 826 828 837 838
[77] 851 865 871 884 899 900 903 935 948 961 964 986 997

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

 [1]   4  27  34  45  52  65  72  99 106 114 121 234 253 268 306 313 320 359 365
[20] 380 388 432 441 444 449 466 508 510 517 529 549 553 558 575 577 596 599 600
[39] 604 630 635 641 649 667 721 723 724 726 739 761 770 773 781 826 837 838 865
[58] 871 899 900 903 948 961 997

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

 [1]  27  34  45  48  65  72  86  99 106 114 148 173 183 234 239 250 253 267 268
[20] 270 306 308 320 331 336 359 365 372 380 388 402 431 432 441 444 449 466 508
[39] 517 549 553 558 575 577 596 600 626 630 635 641 649 683 689 709 721 723 724
[58] 726 739 741 757 761 770 773 817 826 837 838 851 865 867 871 884 891 896 899
[77] 900 901 903 913 948 961 989 997

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

[1] 9

> 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] 13
```

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] "D:/biocbuild/bbs-3.14-bioc/tmpdir/RtmpQFoqi4/Rinst156453e62d0c/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] 58

> sum(diff_results$bootstrap_p<=0.05)

[1] 60

> 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] 4

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

[1] 3

> 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]
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
83     0.04965089    0.04833366    0.03466159    0.04390894
245    0.04208405    0.05284988    0.03775905    0.03955271
280    0.61920530    0.61925200    0.46753250    0.55632410
764    0.90575890    0.88760470    0.90756300    0.90946790
    diff_results$ordfit_t[diff_list_perm]
83                               2.514109
245                              1.962457
280                              4.170347
764                             -1.808081
    diff_results$permutation_p[diff_list_perm]
```

```
83                                          0
245                                         0
280                                         0
764                                         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]
81  cg00071250 0.75466670    0.63467700    0.66801320    0.52600930
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
81      0.7038129    0.68161260    0.62075130    0.52060100
106     0.0469303    0.06837343    0.04534005    0.03709488
146     0.6719151    0.63137380    0.47929610    0.45428300
    diff_results$ordfit_t[diff_list_boot]
81                               2.729694
106                              3.100324
146                              5.394750
    diff_results$bootstrap_p[diff_list_boot]
81                                       0
106                                      0
146                                      0
```