Introduction to RBM package

Dongmei Li

April 26, 2022

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] 19

> which(myresult$permutation_p<=0.05)

 [1]   1  40  88 121 206 225 235 254 322 350 364 440 550 565 690 739 744 794 933

> sum(myresult$bootstrap_p<=0.05)

[1] 19

> which(myresult$bootstrap_p<=0.05)

 [1]  34  52  72  74 101 146 206 209 254 275 365 492 556 747 810 852 876 947 963

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

[1] 8

> 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] 44

> which(myresult2$bootstrap_p<=0.05)

 [1]   1  44  62 100 155 158 173 233 247 278 351 364 366 377 384 393 394 396 406
[20] 485 496 533 557 561 604 608 617 653 658 668 677 686 694 699 704 730 751 780
[39] 796 825 835 875 901 904

> 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] 49

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

[1] 54

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

[1] 58

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

 [1]  31  57  59  60 118 126 145 151 152 164 171 181 207 212 239 247 281 286 302
[20] 359 365 367 404 409 423 427 432 442 473 475 481 496 627 687 711 743 749 762
[39] 795 805 847 873 889 901 923 931 933 935 968

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

 [1]  30  42  55  57  60 127 133 151 152 164 168 181 198 212 239 247 281 286 296
[20] 302 359 365 385 398 404 423 427 432 475 478 496 582 613 627 666 675 711 731
[39] 739 743 749 762 778 795 889 901 908 923 931 933 935 957 968 983

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

 [1]  31  55  57  59  60 118 126 145 151 152 164 171 181 207 223 239 247 281 286
[20] 321 359 365 367 398 404 420 423 427 432 442 458 474 475 478 481 496 613 675
[39] 711 731 749 762 795 847 873 889 901 908 914 923 931 935 953 957 968 970 983
[58] 996

> 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] 10

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

[1] 6

> which(con2_adjp<=0.05/3)

 [1]  57 151 152 181 239 432 795 889 901 923

> which(con3_adjp<=0.05/3)

[1] 239 281 889 923 931 935

> 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] 59

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

[1] 51

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

[1] 62

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

 [1]  11  34  38  81 107 137 157 162 169 177 180 250 253 261 266 278 280 285 293
[20] 309 324 329 354 371 419 457 482 501 513 514 515 521 524 545 547 566 588 593
[39] 641 665 686 691 697 748 750 772 813 819 824 830 843 852 886 904 913 941 950
[58] 960 975
```

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

 [1]  11  34  38 107 137 169 176 177 180 261 266 278 280 285 327 329 370 371 419
[20] 457 482 492 513 514 524 545 566 573 588 593 686 691 697 700 710 727 748 750
[39] 795 798 813 830 843 852 886 913 941 950 960 966 975

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

 [1]  11  34  38 107 137 157 169 177 180 250 252 261 266 278 280 285 317 324 327
[20] 329 354 371 419 457 482 492 499 513 514 515 521 524 545 573 588 593 670 685
[39] 686 687 691 697 700 727 741 748 750 783 795 798 824 830 843 846 858 873 886
[58] 904 913 915 960 975

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

[1] 6

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

[1] 7

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=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 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] "/tmp/Rtmp91e2Ln/Rinst2c85715cfd7785/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] 28

> 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] 7

> 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
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
245    0.04208405    0.05284988    0.03775905    0.03955271
764    0.90575890    0.88760470    0.90756300    0.90946790
    diff_results$ordfit_t[diff_list_perm]
245                              1.962457
764                             -1.808081
    diff_results$permutation_p[diff_list_perm]
245                                         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]
26  cg00024812 0.04783452    0.05005079    0.04229271    0.03882523
200 cg00183916 0.03525946    0.03984548    0.02765822    0.02789838
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
437 cg00424946 0.04122172    0.04325330    0.03339863    0.02876798
882 cg00858899 0.11427700    0.11919540    0.07690343    0.08321229
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
```

```
26     0.04305262    0.04834676    0.04380647    0.04187303
200    0.03034811    0.04302129    0.02753873    0.03067437
259    0.04030003    0.03996053    0.05086962    0.05445672
437    0.03353116    0.03719167    0.03096761    0.03234779
882    0.08961409    0.10730660    0.09203980    0.08726349
911    0.08633986    0.06765189    0.09070268    0.12417730
928    0.04490690    0.04204062    0.05050039    0.05268215
    diff_results$ordfit_t[diff_list_boot]
26                               1.731637
200                              2.272449
259                             -4.052697
437                              2.102892
882                              3.179415
911                             -3.621731
928                             -2.716443
    diff_results$bootstrap_p[diff_list_boot]
26                                       0
200                                      0
259                                      0
437                                      0
882                                      0
911                                      0
928                                      0
```