Introduction to RBM package

Dongmei Li

April 15, 2025

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] 45

> which(myresult$permutation_p<=0.05)

 [1]  40  41  44  47  60  72  79 142 206 221 233 246 258 261 278 285 308 449 467
[20] 472 489 519 525 589 614 624 683 685 695 714 725 732 744 778 785 813 817 834
[39] 857 859 884 942 955 963 997

> sum(myresult$bootstrap_p<=0.05)

[1] 17

> which(myresult$bootstrap_p<=0.05)

 [1]  28  67  92 185 303 360 395 406 467 514 640 732 746 778 813 974 986

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

[1] 4

> 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] 17

> which(myresult2$bootstrap_p<=0.05)

 [1]  24  63  96 149 253 296 389 459 619 659 842 853 873 886 927 928 968

> 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] 61

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

[1] 45

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

[1] 45

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

 [1]  17  29  60  97 108 128 138 141 143 147 150 171 185 187 195 203 204 226 245
[20] 256 295 316 342 349 364 391 419 445 454 464 481 525 541 546 551 559 586 619
[39] 634 651 661 668 702 703 704 744 787 797 802 811 826 856 894 900 927 948 952
[58] 953 956 962 993

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

 [1]  97 108 128 138 141 142 143 147 171 185 203 204 226 256 295 349 391 419 445
[20] 464 481 525 541 546 551 562 586 619 634 651 661 668 702 703 704 744 787 797
[39] 802 826 856 865 900 952 956

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

 [1]  60  72  97 108 128 138 143 147 171 185 203 204 226 256 295 316 349 391 419
[20] 445 481 525 541 546 551 562 563 586 619 634 651 661 668 703 704 787 797 802
[39] 826 856 900 924 948 956 962

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

[1] 12
```

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

[1] 9

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

[1] 11

> which(con2_adjp<=0.05/3)

[1] 128 171 203 204 295 634 703 704 856

> which(con3_adjp<=0.05/3)

 [1] 203 204 295 349 391 551 703 704 797 826 856

> 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] 54

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

[1] 65

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

[1] 58

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

 [1]  27  32  69 101 105 107 119 128 133 150 166 174 197 222 233 274 281 310 314
[20] 316 318 323 329 358 359 367 373 378 379 413 416 439 554 560 563 583 604 686
[39] 696 697 739 745 773 784 789 809 848 878 888 903 929 944 972 986
```

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

 [1]  27  32  69 101 107 119 128 133 150 161 166 174 226 232 233 236 274 281 304
[20] 310 314 316 318 323 325 329 338 358 359 366 367 373 378 379 413 416 495 554
[39] 560 563 604 607 623 629 631 643 646 686 697 721 739 745 773 789 809 848 878
[58] 881 888 929 944 947 972 981 986

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

 [1]  27  69 101 107 119 133 150 161 166 174 191 202 226 233 236 274 281 310 314
[20] 316 318 323 358 359 367 378 379 413 416 495 554 560 563 601 604 623 629 631
[39] 643 686 697 739 745 773 801 809 848 863 878 881 888 929 934 939 966 972 981
[58] 986

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

[1] 5

> 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] 5
```

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/Rtmpu5pzXY/Rinstc5f0f70eb58/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.:0.59031   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] 47

> sum(diff_results$permutation_p<=0.05)

[1] 56
```

```
> sum(diff_results$bootstrap_p<=0.05)

[1] 61

> 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] 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]
19  cg00016968 0.80628480            NA    0.81440820    0.83623180
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
627    0.04740551    0.05332965    0.05775211    0.05579710
928    0.04490690    0.04204062    0.05050039    0.05268215
    diff_results$ordfit_t[diff_list_perm]
19                              -2.547097
627                             -1.797392
928                             -1.982308
    diff_results$permutation_p[diff_list_perm]
19                                       0
627                                      0
928                                      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]
146 cg00134539 0.6110132     0.5332178     0.4599934     0.4678742
397 cg00394658 0.2794090     0.4041033     0.4026232     0.4433929
```

```
979 cg00945507 0.1343225     0.2385460     0.3474976     0.2890334
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
146     0.6719151     0.6313738     0.4792961     0.4542830
397     0.3562606     0.2338838     0.4197463     0.4580688
979     0.1184851     0.1665385     0.3071842     0.2662474
    diff_results$ordfit_t[diff_list_boot]
146                            5.636263
397                           -3.219874
979                           -4.968792
    diff_results$bootstrap_p[diff_list_boot]
146                                    0
397                                    0
979                                    0
```