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] 29

> which(myresult$permutation_p<=0.05)

 [1]  24  32  48 108 112 133 140 152 202 234 289 312 318 393 398 407 408 474 475
[20] 496 538 577 579 663 692 812 814 872 986

> sum(myresult$bootstrap_p<=0.05)

[1] 3

> which(myresult$bootstrap_p<=0.05)

[1] 315 398 747

> 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] 54

> which(myresult2$bootstrap_p<=0.05)

 [1]   12   45   71   72  105  109  130  138  195  202  213  217  227  231  238
[16]  279  286  287  312  330  360  362  368  372  390  408  446  466  486  499
[31]  554  575  642  643  677  687  722  727  760  762  765  800  803  834  852
[46]  856  872  876  882  910  960  968  971 1000

> 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] 62

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

[1] 65

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

[1] 46

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

 [1]   2  17  28  78  91 146 155 156 163 182 199 214 218 219 225 231 234 247 255
[20] 274 280 293 314 318 362 375 393 433 440 444 482 486 500 501 504 509 531 548
[39] 553 584 599 647 665 667 672 680 704 707 749 759 800 805 826 829 849 852 897
[58] 910 928 969 974 975

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

 [1]   2  28  58  78  86  91 125 146 155 156 163 182 199 214 219 224 225 231 255
[20] 274 280 293 318 344 359 362 375 393 433 444 460 462 482 500 504 509 531 553
[39] 584 599 629 665 667 672 680 681 704 707 749 759 805 826 829 831 849 852 873
[58] 897 910 928 969 974 975 990 992

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

 [1]   2  28  86  91 146 156 182 199 214 219 224 225 231 274 286 293 318 362 375
[20] 433 482 504 509 531 553 584 599 667 680 689 704 707 749 759 805 826 843 849
[39] 852 897 910 928 945 969 974 975
```

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

```
[1] 0
```

```
> 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] 2
```

```
> which(con2_adjp<=0.05/3)
```

```
 [1]   2  28 274 293 375 553 584 667 852 928
```

```
> which(con3_adjp<=0.05/3)
```

```
[1] 293 584
```

```
> 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] 60
```

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

```
[1] 67
```

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

 [1]  16  56  57  68  88 103 119 162 231 236 242 253 263 268 288 290 291 330 343
[20] 394 420 425 440 441 499 506 542 543 546 557 560 567 596 599 624 640 656 658
[39] 662 688 689 690 707 718 751 756 790 807 811 828 863 892 900 909 921 930 933
[58] 942 951 963 993

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

 [1]  16  27  38  56  57  64  68  88  95 103 119 128 162 220 231 233 242 253 263
[20] 285 288 290 291 343 394 420 425 440 454 506 542 546 557 560 650 656 658 662
[39] 688 689 690 756 790 794 807 811 828 863 867 877 892 900 909 930 933 942 951
[58] 953 963 993

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

 [1]  16  27  56  57  64  68  88 103 110 119 128 162 220 231 233 236 242 253 263
[20] 288 290 291 330 337 342 371 382 394 420 425 440 468 499 506 542 543 546 557
[39] 560 567 596 656 658 662 670 688 689 690 718 751 756 790 794 811 828 847 877
[58] 900 921 930 933 942 949 951 953 963 993

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

[1] 7

> 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] 7
```

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-3.22-bioc/tmpdir/RtmpoHZtMy/Rinst88e026781758/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] 57

> sum(diff_results$bootstrap_p<=0.05)

[1] 57

> 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] 7

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

[1] 1

> 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
237 cg00215066 0.94926640    0.95311870    0.94634910    0.94561120
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
283 cg00262415 0.03850601    0.04621248    0.03579758    0.03765227
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
931 cg00901704 0.05734342    0.04812868    0.04478214    0.03878488
992 cg00954003 0.03562408    0.04616037    0.02711775    0.03471738
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
83     0.04965089    0.04833366    0.03466159    0.04390894
237    0.94837410    0.94665570    0.94089070    0.94600090
245    0.04208405    0.05284988    0.03775905    0.03955271
283    0.03746915    0.04200230    0.03014699    0.02903290
851    0.50820240    0.34657470    0.66276570    0.64634510
931    0.04497277    0.05751033    0.03089829    0.04423603
992    0.03473852    0.04174397    0.02698795    0.03493307
```

```
    diff_results$ordfit_t[diff_list_perm]
83                               1.947226
237                              1.021426
245                              1.494678
283                              1.601804
851                             -2.986319
931                              2.127264
992                              1.653496
    diff_results$permutation_p[diff_list_perm]
83                                           0
237                                          0
245                                          0
283                                          0
851                                          0
931                                          0
992                                          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]
833 cg00814580 0.09348613    0.09619816     0.1201044     0.1153424
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
833     0.0957704     0.1159885     0.1286089      0.141112
    diff_results$ordfit_t[diff_list_boot]
833                             -3.278186
    diff_results$bootstrap_p[diff_list_boot]
833                                        0
```