Introduction to RBM package

Dongmei Li

October 29, 2019

Clinical and Translational Science Institute, University of Rochester School of Medicine and Dentistry, Rochester, NY 14642-0708

Contents

1 Overview 1

2 Getting started 2

3 `RBM_T` and `RBM_F` functions 2

4 Ovarian cancer methylation example using the `RBM_T` function 6

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] 36

> which(myresult$permutation_p<=0.05)

 [1]   3  87 185 219 224 229 243 255 259 290 307 349 408 421 475 497 511 554 556
[20] 608 618 684 717 729 748 788 814 819 825 886 894 896 935 951 953 986

> sum(myresult$bootstrap_p<=0.05)

[1] 19

> which(myresult$bootstrap_p<=0.05)

 [1]  37  43 288 290 478 511 554 563 571 608 618 693 694 729 738 791 798 828 865

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

[1] 3

> 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]  22  73 162 167 194 203 208 276 317 338 346 363 375 400 420 461 472 526 581
[20] 592 606 614 633 651 654 668 713 725 764 776 836 860 873 890 899 928 947 955
[39] 961 980

> 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] 83

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

[1] 78

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

[1] 52

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

 [1]   15   39   58   87   96   97  103  107  128  134  137  160  161  210  230
[16]  246  247  251  256  282  284  286  290  304  305  344  358  369  392  399
[31]  413  414  415  419  426  433  454  460  462  464  479  480  516  589  599
[46]  601  631  638  651  683  685  692  693  697  724  732  741  757  762  769
[61]  773  794  802  807  845  855  866  874  888  907  909  914  917  955  956
[76]  960  967  977  979  984  992  993 1000

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

 [1]    6   15   39   58   70   78   96   97  120  134  137  210  213  230  246
[16]  247  251  256  286  304  305  344  358  392  399  413  426  433  454  464
[31]  479  480  486  510  516  569  589  599  601  631  638  651  675  679  683
[46]  685  692  693  697  713  724  757  769  773  790  794  808  837  845  855
[61]  866  874  888  907  909  914  917  955  956  960  962  967  977  979  984
[76]  992  993 1000

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

```
 [1]   15   39   58   78   96   97  128  134  137  210  230  256  286  304  305
[16]  344  392  399  403  426  451  454  480  497  510  516  599  631  638  685
[31]  692  693  697  724  757  769  773  794  855  907  909  917  955  956  960
[46]  967  977  979  984  992  993 1000
```

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

```
[1] 19
```

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

```
[1] 14
```

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

```
[1] 3
```

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

```
 [1]   96  137  304  344  358  516  599  638  724  794  956  977  992 1000
```

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

```
[1] 210 256 794
```

```
> 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] 53
```

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

```
[1] 53
```

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

[1] 63

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

 [1]  11  27  41  46  52  79  95  99 146 198 202 215 313 333 342 353 354 355 367
[20] 377 404 421 459 469 472 493 494 503 509 515 585 595 604 606 619 667 699 705
[39] 722 796 807 835 836 845 855 869 880 903 928 932 987 991 998

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

 [1]  11  16  27  41  46  52  76  79  95 130 146 197 198 202 215 264 313 333 342
[20] 354 355 377 383 421 459 469 503 509 515 585 595 597 599 604 606 619 699 705
[39] 722 754 774 796 807 836 845 855 880 903 904 928 932 987 998

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

 [1]  11  27  38  41  46  79  95  99 130 146 167 168 197 198 215 264 313 323 342
[20] 353 354 355 377 383 404 421 446 459 469 472 489 494 503 509 558 561 595 604
[39] 606 619 641 667 699 705 722 754 796 807 835 836 845 852 855 868 880 888 903
[58] 928 932 945 981 987 998

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

[1] 3

> 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] 4
```

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/RtmpmlIcax/Rinstc721284110dc/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] 38
```

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

```
[1] 86
```

```
> 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] 1
```

```
> 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
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
245    0.04208405    0.05284988    0.03775905    0.03955271
    diff_results$ordfit_t[diff_list_perm]
245                              1.962457
    diff_results$permutation_p[diff_list_perm]
245                                          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]
200 cg00183916 0.03525946    0.03984548    0.02765822    0.02789838
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
```

```
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
882 cg00858899 0.11427700    0.11919540    0.07690343    0.08321229
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
200    0.03034811    0.04302129    0.02753873    0.03067437
259    0.04030003    0.03996053    0.05086962    0.05445672
285    0.11693600    0.10650430    0.12281160    0.12310430
627    0.04740551    0.05332965    0.05775211    0.05579710
848    0.05230160    0.06119713    0.06542751    0.06240686
882    0.08961409    0.10730660    0.09203980    0.08726349
911    0.08633986    0.06765189    0.09070268    0.12417730
    diff_results$ordfit_t[diff_list_boot]
200                              2.272449
259                             -4.052697
285                             -3.093997
627                             -2.239498
848                             -2.314412
882                              3.179415
911                             -3.621731
    diff_results$bootstrap_p[diff_list_boot]
200                                        0
259                                        0
285                                        0
627                                        0
848                                        0
882                                        0
911                                        0
```