Introduction to RBM package

Dongmei Li

July 1, 2024

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] 26

> which(myresult$permutation_p<=0.05)

 [1]   18   37   81  127  154  237  318  338  365  404  412  422  436  476  502
[16]  529  628  746  756  829  868  917  937  957  996 1000

> sum(myresult$bootstrap_p<=0.05)

[1] 5

> which(myresult$bootstrap_p<=0.05)

[1]    5  103  122  957 1000

> 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] 7

> which(myresult2$bootstrap_p<=0.05)

[1]   3 178 520 633 713 754 805

> 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] 43

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

[1] 61

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

[1] 59

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

 [1]  51  69  85  86  95 137 154 171 200 232 290 298 307 326 421 442 444 466 515
[20] 592 636 637 639 654 655 665 674 676 685 719 745 769 777 822 835 855 860 867
[39] 870 877 927 933 985

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

 [1]  69  83  85  86  95 137 154 171 197 200 290 294 298 307 309 326 354 359 421
[20] 435 442 444 466 514 515 516 557 571 588 592 620 624 633 636 637 639 646 654
[39] 655 658 664 665 674 685 689 713 719 764 769 770 803 822 825 835 855 870 876
[58] 877 927 933 985

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

 [1]  18  24  69  83  85  95 137 154 171 197 200 232 254 290 294 298 307 326 421
[20] 425 442 444 466 514 515 516 528 536 571 620 624 636 639 646 654 655 658 664
[39] 665 674 676 685 689 713 719 745 764 769 777 803 822 825 855 867 870 877 927
[58] 933 985

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

```
[1] 2

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

[1] 12

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

[1] 9

> which(con2_adjp<=0.05/3)

 [1]  95 171 200 466 516 571 639 655 685 689 713 769

> which(con3_adjp<=0.05/3)

[1]  95 171 466 571 674 685 769 870 933

> 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] 55

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

[1] 50

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

```
 [1]  47  51  70 123 139 146 162 183 201 202 207 208 223 239 248 259 261 289 354
[20] 363 379 404 406 408 411 435 480 489 498 505 519 536 543 559 570 598 620 640
[39] 675 685 688 692 706 723 727 740 741 744 807 817 818 867 895 897 906 926 937
[58] 944 973
```

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

```
 [1]   1  51  70 101 139 162 183 201 208 223 230 239 248 261 289 354 363 379 404
[20] 406 408 411 435 454 480 489 498 505 519 543 598 612 620 640 675 684 685 688
[39] 706 723 727 741 744 753 799 818 867 895 897 906 921 926 937 944 973
```

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

```
 [1]   1  45  51 139 146 162 183 186 201 208 223 239 248 261 289 363 379 406 411
[20] 479 480 489 498 505 519 536 543 559 620 640 675 688 692 706 723 727 741 753
[39] 799 817 818 867 895 897 921 926 937 944 968 973
```

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

```
[1] 10
```

```
> 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] 11
```

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/RtmpdKDCbQ/Rinst358294681d2/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] 60

> sum(diff_results$bootstrap_p<=0.05)

[1] 50

> 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] 2

> 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
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
83     0.04965089    0.04833366    0.03466159    0.04390894
851    0.50820240    0.34657470    0.66276570    0.64634510
    diff_results$ordfit_t[diff_list_perm]
83                               2.514109
851                             -2.841244
    diff_results$permutation_p[diff_list_perm]
83                                           0
851                                          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]
95  cg00081975 0.03633894    0.04975194    0.06024723    0.05598723
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
95     0.04561792    0.05115624    0.06068253    0.06168212
848    0.05230160    0.06119713    0.06542751    0.06240686
```

```
    diff_results$ordfit_t[diff_list_boot]
95                              -3.252063
848                             -2.314412
    diff_results$bootstrap_p[diff_list_boot]
95                                         0
848                                        0
```