Introduction to RBM package

Dongmei Li

November 8, 2024

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] 24

> which(myresult$permutation_p<=0.05)

 [1]  24  33  47 126 141 149 202 209 230 379 398 400 417 427 444 651 690 750 792
[20] 909 915 918 949 967

> sum(myresult$bootstrap_p<=0.05)

[1] 2

> which(myresult$bootstrap_p<=0.05)

[1] 202 445

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

[1] 1

> 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] 21

> which(myresult2$bootstrap_p<=0.05)

 [1]  27  70  77 123 240 306 370 406 464 521 665 679 702 704 727 802 829 836 873
[20] 922 942

> 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] 70

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

[1] 61

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

[1] 63

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

 [1]   12   25   30   34   44   54   88  103  104  125  128  171  178  197  207
[16]  231  253  255  282  294  314  317  323  325  326  340  349  360  364  370
[31]  372  395  414  418  422  435  448  460  462  468  479  498  510  532  557
[46]  593  612  621  630  632  667  669  696  766  793  847  854  855  856  922
[61]  929  939  949  953  966  976  982  987  992 1000

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

 [1]   12   25   30   34   44   54   93  103  104  125  128  171  207  209  231
[16]  253  255  269  282  294  314  317  323  326  349  360  364  370  372  395
[31]  414  418  422  439  448  460  462  468  479  498  518  532  557  612  630
[46]  667  766  793  847  854  856  858  939  949  953  976  982  983  987  992
[61] 1000

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

 [1]   12   25   30   34   44   54   61   79  103  104  125  128  171  175  207
[16]  231  253  255  282  294  314  317  323  325  326  349  360  364  370  372
[31]  395  414  422  435  447  448  460  462  468  479  498  510  532  557  612
[46]  667  766  793  814  819  847  854  855  856  929  939  949  953  976  982
[61]  987  992 1000
```

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

[1] 15

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

[1] 17

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

[1] 14

> which(con2_adjp<=0.05/3)

 [1]   25   54  103  104  372  395  422  479  557  793  847  854  856  953  982
[16]  992 1000

> which(con3_adjp<=0.05/3)

 [1]  12  54 103 104 231 253 282 370 372 422 462 793 953 982

> 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] 60

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

[1] 50

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

[1] 48
```

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

 [1]  35  37  41  80  87  94  97 127 162 176 194 199 200 201 245 260 267 312 315
[20] 316 317 348 385 417 433 436 449 486 538 543 548 583 584 585 606 619 623 642
[39] 687 717 741 773 774 798 800 802 808 809 833 842 856 861 891 914 919 926 930
[58] 951 957 993

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

 [1]   5  12  41  87  90  94 127 162 176 194 199 201 218 260 267 312 315 317 385
[20] 411 417 436 449 474 486 538 543 548 584 585 596 606 619 687 717 773 774 798
[39] 800 802 808 809 833 842 890 919 926 930 951 983

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

 [1]   4   5  41  87  94  97 127 162 176 194 200 277 312 315 316 317 385 417 433
[20] 449 474 486 538 543 548 583 584 585 623 640 687 734 773 774 798 800 802 808
[39] 809 833 842 861 890 919 926 930 951 957

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

[1] 4

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

[1] 3

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)

[1] 2
```

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/var/folders/r0/l4fjk6cj5xj0j3brt4bplpl40000gt/T/RtmphYntGX/Rinst2d7129819cf0/RBM/d

> 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] 47
```

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

[1] 63

> sum(diff_results$bootstrap_p<=0.05)

[1] 68

> 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] 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]
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
848    0.05230160    0.06119713    0.06542751    0.06240686
851    0.50820240    0.34657470    0.66276570    0.64634510
911    0.08633986    0.06765189    0.09070268    0.12417730
    diff_results$ordfit_t[diff_list_perm]
848                             -1.687144
851                             -2.986319
911                             -3.490240
    diff_results$permutation_p[diff_list_perm]
848                                         0
851                                         0
911                                         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
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
632 cg00615377 0.11265030    0.16140570    0.19404450    0.17468600
677 cg00651216 0.06825629    0.12529090    0.14409190    0.13907250
833 cg00814580 0.09348613    0.09619816    0.12010440    0.11534240
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
95     0.04561792    0.05115624    0.06068253    0.06168212
146    0.67191510    0.63137380    0.47929610    0.45428300
280    0.61920530    0.61925200    0.46753250    0.55632410
632    0.12573100    0.14483660    0.16338240    0.20130510
677    0.07669587    0.09597587    0.11690440    0.15194540
833    0.09577040    0.11598850    0.12860890    0.14111200
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
95                              -2.654324
146                              5.636263
280                              4.337628
632                             -3.722206
677                             -3.457874
833                             -3.278186
979                             -4.968792
    diff_results$bootstrap_p[diff_list_boot]
95                                        0
146                                       0
280                                       0
632                                       0
677                                       0
833                                       0
979                                       0
```