Introduction to RBM package

Dongmei Li

May 2, 2019

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] 13

> which(myresult$permutation_p<=0.05)

 [1]   2  31  88 331 372 377 485 535 788 854 923 930 943

> sum(myresult$bootstrap_p<=0.05)

[1] 0

> which(myresult$bootstrap_p<=0.05)

integer(0)

> 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] 21

> which(myresult2$bootstrap_p<=0.05)

 [1]  49 106 135 174 220 283 297 356 409 428 479 632 703 712 737 776 832 855 878
[20] 950 986

> 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] 64

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

[1] 47

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

[1] 56

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

 [1]  10  12  15  37  40  42  52  53  85 126 165 186 212 239 240 256 293 324 328
[20] 355 363 378 462 467 495 521 525 537 538 539 544 574 580 587 589 594 595 599
[39] 601 608 626 636 640 642 650 664 674 689 700 701 702 704 728 745 750 789 800
[58] 876 886 899 945 957 973 983

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

 [1]  10  15  42  58  85 111 165 186 193 239 240 293 300 324 328 340 363 462 467
[20] 486 495 521 544 587 589 594 650 664 674 700 701 702 704 705 718 728 745 750
[39] 789 829 886 891 899 945 957 973 983

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

 [1]  10  42  52  58  71  85 165 186 193 212 240 293 300 324 328 355 363 378 383
[20] 412 467 487 495 521 525 542 544 574 589 594 595 601 636 642 650 664 674 687
[39] 700 701 702 704 705 718 728 743 745 789 800 851 876 886 891 899 957 983

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

[1] 7
```

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

[1] 2

> 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] 328 789

> which(con3_adjp<=0.05/3)

[1] 521 664

> 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] 38

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

[1] 53

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

 [1]  27  37  42  43  58 102 108 110 121 153 212 214 217 228 241 244 247 260 266
[20] 289 298 313 321 343 361 400 412 464 530 536 537 570 593 613 625 639 641 655
[39] 656 670 677 691 714 745 758 759 788 812 828 839 869 893 918 923 932 936 942
[58] 958 975 983 992
```

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

 [1]  27  37  42  58 110 153 212 217 260 266 298 313 321 343 400 464 473 530 536
[20] 537 570 593 613 625 639 641 655 664 670 714 745 758 788 819 869 936 975 983

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

 [1]  37  40  42  43  58 102 110 121 153 212 214 217 228 237 260 266 283 298 313
[20] 343 361 400 412 464 473 526 530 536 537 570 593 610 613 625 641 655 656 670
[39] 691 714 745 749 758 777 788 869 918 924 936 942 958 975 992

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

[1] 2

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

[1] 4

> 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/RtmpF6J8yP/Rinst84a135c5b788/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] 69

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

```
[1] 70

> 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] 5

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

[1] 6

> 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
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
83     0.04965089    0.04833366    0.03466159    0.04390894
103    0.67349260    0.73510200    0.75715920    0.78981220
106    0.04693030    0.06837343    0.04534005    0.03709488
245    0.04208405    0.05284988    0.03775905    0.03955271
848    0.05230160    0.06119713    0.06542751    0.06240686
    diff_results$ordfit_t[diff_list_perm]
83                              2.514109
103                            -2.268711
106                             3.100324
245                             1.962457
848                            -2.314412
    diff_results$permutation_p[diff_list_perm]
83                                           0
103                                          0
106                                          0
245                                          0
848                                          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
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
743 cg00717862 0.07999436    0.07873347    0.06089359    0.06171374
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
259    0.04030003    0.03996053    0.05086962    0.05445672
743    0.07594936    0.09062161    0.06475791    0.07271878
833    0.09577040    0.11598850    0.12860890    0.14111200
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
95                              -3.252063
146                              5.394750
259                             -4.052697
743                              3.444684
833                             -3.428319
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
95                                       0
146                                      0
259                                      0
743                                      0
833                                      0
979                                      0
```