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

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] 33

> which(myresult$permutation_p<=0.05)

 [1]  16  45  54  63 157 169 176 185 272 301 304 328 331 368 434 461 468 473 597
[20] 621 651 659 722 742 744 773 811 831 868 902 948 958 974

> sum(myresult$bootstrap_p<=0.05)

[1] 9

> which(myresult$bootstrap_p<=0.05)

[1]  61 153 185 230 288 368 409 710 742

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

[1] 5

> 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] 42

> which(myresult2$bootstrap_p<=0.05)

 [1]  17  43  44  47  89  91  98 105 119 156 189 221 226 230 263 331 414 419 436
[20] 446 477 513 520 543 580 584 607 609 639 673 674 679 731 763 778 786 802 849
[39] 864 872 890 921

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=0.05)

[1] 2
```

- 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] 68

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

[1] 52

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

[1] 60

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

 [1]   69   82   87  103  120  135  139  147  149  161  174  187  196  199  226
[16]  229  233  242  247  261  287  289  296  297  332  337  346  349  356  376
[31]  381  417  426  427  442  461  476  511  514  547  561  569  574  586  587
[46]  590  627  634  651  703  713  729  744  770  785  788  809  829  831  867
[61]  868  902  912  927  976  987  991 1000

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

 [1]  69  82  87 120 135 139 147 149 161 174 196 226 229 233 242 247 261 287 296
[20] 346 349 354 356 376 426 432 442 465 476 514 547 561 569 574 586 587 627 651
[39] 703 729 744 785 809 829 831 839 868 927 969 976 987 991

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

 [1]  72  82 120 135 147 149 161 174 196 199 226 233 242 247 261 287 296 324 332
[20] 337 346 349 356 376 399 426 427 442 461 476 511 514 547 561 569 574 586 587
[39] 627 648 703 713 744 770 785 788 809 813 829 831 839 868 892 907 908 966 969
[58] 976 987 991
```

```
> 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] 6

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

[1] 8

> which(con2_adjp<=0.05/3)

[1]  82 174 356 376 785 987

> which(con3_adjp<=0.05/3)

[1]  82 120 174 247 332 356 785 809

> 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] 50

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

[1] 69

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

[1] 58
```

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

 [1]  20  26  29 101 134 148 169 194 250 263 268 288 309 317 321 333 335 383 391
[20] 484 493 529 532 539 547 551 557 582 637 649 650 668 696 730 736 763 785 808
[39] 814 817 818 831 856 876 879 906 928 929 930 933

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

 [1]   7  20  26  29  48  94 101 107 148 169 194 202 219 225 250 263 288 309 317
[20] 321 335 378 383 391 404 426 484 486 493 529 532 533 539 542 547 551 557 577
[39] 585 637 649 650 668 671 685 693 696 721 736 746 763 766 798 814 817 818 824
[58] 831 856 876 879 886 906 928 929 930 950 985 998

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

 [1]   7  20  29  94 101 148 169 194 197 202 250 263 288 309 317 321 333 335 373
[20] 378 383 391 479 484 486 493 526 529 539 551 557 571 649 650 668 685 696 721
[39] 730 736 763 766 814 817 818 824 831 876 879 906 928 929 930 933 950 966 976
[58] 978

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

[1] 6

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

[1] 5

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

[1] 9
```

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] "C:/Users/biocbuild/bbs-3.10-bioc/tmpdir/RtmpOUcNbr/Rinst294c327d601/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] 39

> sum(diff_results$bootstrap_p<=0.05)

[1] 63

> 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] 3

> 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]
146 cg00134539 0.6110132     0.5332178     0.4599934     0.4678742
851 cg00830029 0.5836250     0.5939787     0.6473961     0.6726964
979 cg00945507 0.1343225     0.2385460     0.3474976     0.2890334
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
146     0.6719151     0.6313738     0.4792961     0.4542830
851     0.5082024     0.3465747     0.6627657     0.6463451
979     0.1184851     0.1665385     0.3071842     0.2662474
```

```
    diff_results$ordfit_t[diff_list_boot]
146                              5.394750
851                             -2.841244
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
146                                       0
851                                       0
979                                       0
```