Introduction to RBM package

Dongmei Li

April 27, 2020

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] 19

> which(myresult$permutation_p<=0.05)

 [1] 101 157 250 305 350 358 367 463 469 475 537 555 638 706 748 839 844 869 978

> sum(myresult$bootstrap_p<=0.05)

[1] 2

> which(myresult$bootstrap_p<=0.05)

[1] 236 280

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

[1] 2

> 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] 73

> which(myresult2$bootstrap_p<=0.05)

 [1]   8  38  67  78 108 117 136 145 150 166 173 179 187 202 215 216 226 239 254
[20] 268 277 293 295 316 339 340 350 371 389 393 402 408 417 457 458 460 473 479
[39] 503 505 508 535 548 567 572 581 595 602 653 661 671 675 741 742 777 784 791
[58] 795 807 809 812 818 834 848 850 889 893 914 915 937 959 984 986

> 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] 74

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

[1] 75

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

[1] 75

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

 [1]  31  44  58  66  73  90 101 116 124 125 126 136 143 193 202 206 258 260 274
[20] 275 292 335 336 369 370 391 397 401 408 423 438 463 465 478 482 505 528 529
[39] 538 545 548 564 572 573 578 581 582 583 626 658 686 716 733 735 789 795 807
[58] 821 831 852 874 888 890 901 907 916 917 925 929 943 955 977 985 987

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

 [1]  31  32  44  66  73  90 101 111 116 117 124 125 143 193 202 205 206 215 258
[20] 260 274 275 287 292 336 349 369 391 397 401 423 438 463 465 478 482 497 505
[39] 529 538 545 564 572 575 578 581 582 583 624 686 687 726 733 735 789 795 807
[58] 830 831 852 858 888 890 893 901 907 914 917 925 929 943 955 977 987 996

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

 [1]   7  31  44  66  73  90 101 111 116 124 125 126 142 143 193 202 205 206 242
[20] 245 258 260 274 275 292 336 349 369 370 391 397 401 423 438 463 465 482 497
[39] 505 529 538 545 564 572 573 574 578 581 582 583 624 626 658 686 714 716 726
[58] 735 762 795 807 831 852 858 890 901 907 914 917 925 929 943 955 977 987
```

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

[1] 12

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

[1] 16

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

[1] 13

> which(con2_adjp<=0.05/3)

 [1]  73  90 124 202 258 391 438 463 482 564 578 581 735 807 955 987

> which(con3_adjp<=0.05/3)

 [1] 124 258 391 401 438 465 482 578 581 686 807 977 987

> 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] 59

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

[1] 56
```

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

 [1]   61   65   79   89   90  110  129  130  189  190  211  235  262  266  271
[16]  289  332  407  412  420  424  445  479  495  558  582  617  621  623  629
[31]  635  657  676  677  695  702  715  737  744  752  778  780  791  802  818
[46]  825  829  850  864  870  881  884  903  980  981  983  989  994 1000

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

 [1]   61   65   68   79  110  129  130  145  150  189  190  211  228  235  262
[16]  266  315  318  332  407  412  424  445  458  495  558  581  582  604  617
[31]  621  623  629  657  676  677  702  715  737  744  780  802  804  825  829
[46]  850  864  870  881  884  890  903  925  929  980  987  989  994 1000

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

 [1]   61   65   89  129  150  189  190  211  235  243  262  263  266  271  289
[16]  332  334  407  412  424  445  458  495  533  558  581  582  612  617  623
[31]  629  657  677  695  702  737  752  780  802  804  825  829  864  870  881
[46]  884  890  903  925  980  981  983  987  989  994 1000

> 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] 6

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

[1] 7
```

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/RtmpFaNxZ9/Rinst616c782c1187/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] 63

> sum(diff_results$bootstrap_p<=0.05)

[1] 61

> 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] 7

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

[1] 8

> 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]
16  cg00014085 0.05906804    0.04518973    0.04211710    0.03665208
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
437 cg00424946 0.04122172    0.04325330    0.03339863    0.02876798
772 cg00743372 0.03922780    0.02919634    0.02187972    0.02568053
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
16     0.04222944    0.05324246    0.03728026    0.04062589
83     0.04965089    0.04833366    0.03466159    0.04390894
106    0.04693030    0.06837343    0.04534005    0.03709488
245    0.04208405    0.05284988    0.03775905    0.03955271
280    0.61920530    0.61925200    0.46753250    0.55632410
437    0.03353116    0.03719167    0.03096761    0.03234779
772    0.02796053    0.03512214    0.02575992    0.02093909
```

```
    diff_results$ordfit_t[diff_list_perm]
16                                2.325659
83                                2.514109
106                               3.100324
245                               1.962457
280                               4.170347
437                               2.102892
772                               2.416991
    diff_results$permutation_p[diff_list_perm]
16                                          0
83                                          0
106                                         0
245                                         0
280                                         0
437                                         0
772                                         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
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
397 cg00394658 0.27940900    0.40410330    0.40262320    0.44339290
804 cg00777121 0.04540701    0.05430304    0.04154242    0.04221162
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
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
280    0.61920530    0.61925200    0.46753250    0.55632410
397    0.35626060    0.23388380    0.41974630    0.45806880
804    0.04911277    0.04872797    0.04261405    0.04474881
887    0.50259740    0.40111730    0.56646700    0.54552980
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
280                               4.170347
397                              -3.070559
804                               1.995220
887                              -3.217939
```

```
979                                -4.750997
    diff_results$bootstrap_p[diff_list_boot]
95                                          0
146                                         0
259                                         0
280                                         0
397                                         0
804                                         0
887                                         0
979                                         0
```