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] 86

> which(myresult$permutation_p<=0.05)

 [1]  10  20  39  45  49  52  57  67  81  82  90  96  97 100 105 132 164 166 173
[20] 181 227 247 250 269 275 280 283 289 297 321 345 350 355 369 372 374 393 414
[39] 415 442 445 450 452 462 497 499 504 508 510 529 535 539 550 601 603 611 620
[58] 637 661 674 691 721 726 733 742 758 763 796 803 808 816 824 837 861 881 882
[77] 884 887 932 939 942 943 973 977 983 990

> sum(myresult$bootstrap_p<=0.05)

[1] 20

> which(myresult$bootstrap_p<=0.05)

 [1]   2  12 139 143 301 321 345 386 445 452 526 550 601 603 609 652 691 710 932
[20] 983

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

[1] 6

> 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] 17

> which(myresult2$bootstrap_p<=0.05)

 [1]   2  28 223 241 249 270 315 370 432 526 625 743 765 793 831 964 987

> 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] 65

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

[1] 60

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

[1] 62

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

 [1]   6   7  13  15  19  41  46  65  69  72  82 114 120 131 160 192 208 211 216
[20] 241 276 280 305 322 336 342 347 353 384 385 407 438 443 466 471 481 547 552
[39] 579 580 586 591 594 629 658 676 677 678 706 712 718 748 768 783 831 841 846
[58] 856 860 865 872 889 965 979 993

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

 [1]   6   7  13  15  19  27  41  46  65  69  72  82 110 114 131 160 192 198 208
[20] 211 216 276 280 305 322 336 342 347 353 384 385 438 463 471 481 547 579 580
[39] 586 594 629 658 676 677 678 706 712 718 748 768 820 828 831 841 846 856 889
[58] 951 965 979

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

 [1]   6   7  13  15  19  41  46  57  65  69  72  82 114 120 131 192 198 208 216
[20] 219 241 276 280 305 336 342 353 384 438 466 471 473 481 497 547 579 586 591
[39] 594 610 629 676 678 695 706 712 727 748 768 783 785 831 841 846 856 857 872
[58] 874 889 960 965 979
```

```
> 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] 7

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

[1] 11

> which(con2_adjp<=0.05/3)

[1]  19 131 192 336 481 586 846

> which(con3_adjp<=0.05/3)

 [1]   6  19  46  65  72  82 131 192 466 678 768

> 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] 58

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

[1] 48

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

[1] 44
```

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

 [1]   2   6  10  15  30  73  76  91 110 136 153 155 159 163 164 193 209 218 226
[20] 234 239 241 250 257 274 280 281 330 358 387 397 416 441 444 468 497 523 543
[39] 552 579 651 663 675 679 705 708 722 776 788 793 831 835 871 895 919 931 960
[58] 961

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

 [1]   6  10  30  73  88  91 110 153 159 163 193 209 218 226 234 241 250 257 274
[20] 281 292 358 387 416 441 444 468 497 523 543 579 651 675 678 679 696 705 726
[39] 776 788 793 809 831 919 931 946 960 961

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

 [1]   6  10  15  30  50  64  73  91 110 157 159 193 218 226 239 250 257 274 330
[20] 358 387 403 416 441 468 480 497 543 579 651 663 675 678 679 705 776 788 793
[39] 831 835 851 919 960 961

> 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] 4

> 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/tmp/Rtmpxiadc3/Rinstadf6063223f/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] 73

> sum(diff_results$bootstrap_p<=0.05)

[1] 51

> 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] 13

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

[1] 5

> 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]
19  cg00016968 0.80628480            NA    0.81440820    0.83623180
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
103    0.67349260    0.73510200    0.75715920    0.78981220
131    0.17352980    0.12564280    0.18193170    0.20847670
245    0.04208405    0.05284988    0.03775905    0.03955271
```

```
259    0.04030003    0.03996053    0.05086962    0.05445672
280    0.61920530    0.61925200    0.46753250    0.55632410
285    0.11693600    0.10650430    0.12281160    0.12310430
627    0.04740551    0.05332965    0.05775211    0.05579710
764    0.90575890    0.88760470    0.90756300    0.90946790
848    0.05230160    0.06119713    0.06542751    0.06240686
851    0.50820240    0.34657470    0.66276570    0.64634510
887    0.50259740    0.40111730    0.56646700    0.54552980
928    0.04490690    0.04204062    0.05050039    0.05268215
    diff_results$ordfit_t[diff_list_perm]
19                              -2.446404
103                             -2.268711
131                             -3.451679
245                              1.962457
259                             -4.052697
280                              4.170347
285                             -3.093997
627                             -2.239498
764                             -1.808081
848                             -2.314412
851                             -2.841244
887                             -3.217939
928                             -2.716443
    diff_results$permutation_p[diff_list_perm]
19                                          0
103                                         0
131                                         0
245                                         0
259                                         0
280                                         0
285                                         0
627                                         0
764                                         0
848                                         0
851                                         0
887                                         0
928                                         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 0.61101320    0.53321780     0.4599934    0.46787420
259 cg00234961 0.04192170    0.04321576     0.0570714    0.05327565
482 cg00468146 0.11144740    0.15416650     0.1982799    0.18517240
632 cg00615377 0.11265030    0.16140570     0.1940445    0.17468600
```

```
677 cg00651216 0.06825629    0.12529090     0.1440919    0.13907250
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
146    0.67191510    0.63137380    0.47929610    0.45428300
259    0.04030003    0.03996053    0.05086962    0.05445672
482    0.12285820    0.13271110    0.14196260    0.22159420
632    0.12573100    0.14483660    0.16338240    0.20130510
677    0.07669587    0.09597587    0.11690440    0.15194540
    diff_results$ordfit_t[diff_list_boot]
146                             5.394750
259                            -4.052697
482                            -3.212481
632                            -3.661161
677                            -3.387628
    diff_results$bootstrap_p[diff_list_boot]
146                                       0
259                                       0
482                                       0
632                                       0
677                                       0
```