Introduction to RBM package

Dongmei Li

May 11, 2023

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] 87

> which(myresult$permutation_p<=0.05)

 [1]   2  11  12  29  32  36  44  52  53  67  71  90 104 120 121 125 129 133 145
[20] 151 181 189 193 205 206 207 214 222 224 226 246 253 256 284 314 358 375 382
[39] 398 418 432 437 444 448 465 499 503 533 539 542 551 558 563 610 646 652 655
[58] 673 685 695 727 731 736 743 753 764 779 800 802 815 821 824 835 872 880 891
[77] 906 916 920 930 938 948 952 956 966 976 981

> sum(myresult$bootstrap_p<=0.05)

[1] 10

> which(myresult$bootstrap_p<=0.05)

 [1]  36 121 129 189 237 246 253 452 916 948

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

[1] 7

> 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] 15

> which(myresult2$bootstrap_p<=0.05)

 [1]  12  19  63  88 234 340 427 594 636 657 660 673 768 805 880

> 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] 80

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

[1] 81

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

[1] 72

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

 [1]  26  35  46  48  85  98 107 130 162 213 217 219 229 272 338 359 360 366 370
[20] 391 393 399 405 411 423 439 446 447 455 458 477 479 481 489 490 515 530 536
[39] 539 565 566 569 584 593 599 614 615 627 634 654 672 689 701 702 724 729 746
[58] 750 768 779 798 801 807 836 841 843 849 850 853 854 859 863 869 887 889 944
[77] 951 980 991 996

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

 [1]  26  46  48  85  98 102 104 130 162 213 217 219 224 229 251 270 280 359 360
[20] 366 370 391 393 405 423 439 446 447 455 458 477 479 481 490 536 538 539 565
[39] 566 569 584 587 593 599 614 615 627 634 654 672 684 689 701 702 729 746 750
[58] 768 779 798 801 807 836 841 843 849 850 853 854 859 863 869 876 887 889 944
[77] 951 980 989 991 996

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

```
 [1]  26  35  46  48  85  93  98 130 162 198 213 217 219 229 249 359 360 366 391
[20] 393 399 405 411 423 446 455 458 477 479 481 490 536 539 565 569 593 599 614
[39] 615 627 634 654 672 689 700 701 702 707 729 746 768 779 798 801 807 836 841
[58] 843 850 853 854 859 863 869 876 887 889 951 974 980 991 996

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

[1] 18

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

[1] 20

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

[1] 10

> which(con2_adjp<=0.05/3)

 [1] 130 359 360 393 458 479 490 539 615 634 701 768 779 798 801 807 863 951 980
[20] 991

> which(con3_adjp<=0.05/3)

 [1] 130 359 360 615 634 702 768 798 801 980

> 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] 65

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

```
[1] 63

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

[1] 69

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

 [1]  38 108 117 124 125 127 146 150 190 207 223 233 253 257 261 282 288 311 321
[20] 323 332 336 344 364 377 423 427 431 433 439 450 457 500 527 562 583 600 617
[39] 627 636 638 655 683 688 691 709 730 753 791 818 820 824 840 855 858 883 885
[58] 891 939 944 952 967 974 982 999

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

 [1]  57  65 108 117 125 127 128 133 146 150 190 204 207 233 253 257 261 282 287
[20] 288 321 323 329 332 336 344 364 377 423 427 431 439 450 457 507 517 527 550
[39] 583 600 617 627 638 655 683 688 709 753 791 811 818 820 835 840 855 870 883
[58] 885 891 967 974 982 988

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

 [1]  57  65 108 111 117 125 127 146 150 207 223 233 253 257 261 282 321 323 327
[20] 329 332 336 344 364 377 423 427 431 439 450 457 527 557 580 583 589 617 624
[39] 627 638 654 681 683 688 692 707 709 720 730 753 755 773 791 811 818 820 824
[58] 840 855 870 883 891 939 952 955 967 974 982 988

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

[1] 11

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

[1] 8

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

[1] 3
```

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/v1/y6dg5h4n163dzmrfl6t_r5480000gp/T/RtmpOTZlW5/Rinst655f5bd2af49/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] 45

> sum(diff_results$permutation_p<=0.05)

[1] 78

> 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] 7

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

[1] 2

> 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
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
237 cg00215066 0.94926640    0.95311870    0.94634910    0.94561120
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
```

```
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
83     0.04965089    0.04833366    0.03466159    0.04390894
237    0.94837410    0.94665570    0.94089070    0.94600090
245    0.04208405    0.05284988    0.03775905    0.03955271
280    0.61920530    0.61925200    0.46753250    0.55632410
764    0.90575890    0.88760470    0.90756300    0.90946790
911    0.08633986    0.06765189    0.09070268    0.12417730
    diff_results$ordfit_t[diff_list_perm]
19                               -2.446404
83                                2.514109
237                               1.419654
245                               1.962457
280                               4.170347
764                              -1.808081
911                              -3.621731
    diff_results$permutation_p[diff_list_perm]
19                                         0
83                                         0
237                                        0
245                                        0
280                                        0
764                                        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]
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
349 cg00332745 0.04703361    0.04634372    0.03676908    0.04518837
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
146    0.67191510    0.63137380    0.47929610    0.45428300
349    0.04975075    0.05253778    0.04444665    0.03717721
    diff_results$ordfit_t[diff_list_boot]
146                               5.394750
349                               2.165826
    diff_results$bootstrap_p[diff_list_boot]
146                                        0
349                                        0
```