Introduction to RBM package

Dongmei Li

May 1, 2024

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] 36

> which(myresult$permutation_p<=0.05)

 [1]   9  29  47  51 141 144 146 160 165 217 256 278 405 438 446 461 515 554 608
[20] 627 628 642 652 680 702 726 746 758 785 813 818 827 905 924 949 959

> sum(myresult$bootstrap_p<=0.05)

[1] 38

> which(myresult$bootstrap_p<=0.05)

 [1]   1  21  65 146 160 224 225 226 263 323 378 380 405 424 446 461 479 552 554
[20] 603 607 628 632 634 642 651 677 680 731 746 758 818 866 878 909 941 944 959

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

[1] 3

> 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] 14

> which(myresult2$bootstrap_p<=0.05)

 [1]  18  40 127 169 305 320 347 374 407 455 494 633 813 988

> 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] 51

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

[1] 46

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

[1] 46

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

 [1]   5  68 114 137 160 169 199 215 233 236 239 245 253 292 301 304 322 328 337
[20] 351 393 405 412 435 439 467 483 489 524 527 538 568 591 624 674 707 718 721
[39] 740 770 837 851 890 897 946 954 969 971 976 986 995

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

 [1]   5  68  78  89 137 169 195 199 215 236 239 245 253 292 301 337 344 351 393
[20] 405 412 435 439 467 483 489 524 527 538 591 674 682 707 718 721 740 770 793
[39] 835 837 890 897 946 954 969 986

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

 [1]   5  68  78 114 137 148 169 199 227 233 236 239 245 253 292 301 304 306 322
[20] 351 405 411 412 439 467 483 489 524 527 538 568 674 679 707 718 721 740 837
[39] 890 897 946 954 969 971 985 986

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

[1] 11
```

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

[1] 4

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

[1] 6

> which(con2_adjp<=0.05/3)

[1]  68 351 538 946

> which(con3_adjp<=0.05/3)

[1]  68 351 538 946 954 969

> 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] 54

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

[1] 55

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

[1] 44

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

 [1]  12  32  46  75  93  94 108 109 113 137 144 152 161 178 186 198 201 210 249
[20] 251 262 263 278 279 283 367 387 409 427 430 465 495 571 588 657 665 666 682
[39] 693 735 793 805 808 852 879 890 908 927 929 955 959 965 973 978
```

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

 [1]  32  42  93  94 108 109 113 137 152 161 178 183 198 210 249 251 259 263 278
[20] 279 333 335 337 361 387 409 427 430 465 495 507 517 571 588 652 657 682 735
[39] 752 793 797 805 834 852 876 879 890 927 929 934 955 959 965 973 978

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

 [1]  12  42  46  93  94 102 108 109 113 137 152 183 198 210 246 249 251 263 278
[20] 279 387 427 430 465 495 507 571 588 635 652 657 682 735 793 805 852 876 879
[39] 890 927 929 955 973 978

> 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] 5

> 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] "/tmp/Rtmp6KkzWg/Rinst3f4d3d7bd07efe/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] 87
```

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

```
[1] 57
```

```
> 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] 11
```

```
> 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]
16  cg00014085 0.05906804    0.04518973    0.04211710    0.03665208
19  cg00016968 0.80628480            NA    0.81440820    0.83623180
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
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
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
931 cg00901704 0.05734342    0.04812868    0.04478214    0.03878488
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
16     0.04222944    0.05324246    0.03728026    0.04062589
19     0.80831380    0.73306440    0.82968340    0.84917800
83     0.04965089    0.04833366    0.03466159    0.04390894
131    0.17352980    0.12564280    0.18193170    0.20847670
146    0.67191510    0.63137380    0.47929610    0.45428300
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
627    0.04740551    0.05332965    0.05775211    0.05579710
764    0.90575890    0.88760470    0.90756300    0.90946790
931    0.04497277    0.05751033    0.03089829    0.04423603
    diff_results$ordfit_t[diff_list_perm]
```

```
16                               2.325659
19                              -2.446404
83                               2.514109
131                             -3.451679
146                              5.394750
237                              1.419654
245                              1.962457
280                              4.170347
627                             -2.239498
764                             -1.808081
931                              2.464709
    diff_results$permutation_p[diff_list_perm]
16                                           0
19                                           0
83                                           0
131                                          0
146                                          0
237                                          0
245                                          0
280                                          0
627                                          0
764                                          0
931                                          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
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    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
280    0.61920530    0.61925200    0.46753250    0.55632410
848    0.05230160    0.06119713    0.06542751    0.06240686
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
146                              5.394750
259                             -4.052697
280                              4.170347
848                             -2.314412
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
```

```
146                                         0
259                                         0
280                                         0
848                                         0
979                                         0
```