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] 24

> which(myresult$permutation_p<=0.05)

 [1]    1   59   88  100  104  113  170  197  251  256  260  322  368  396  492
[16]  505  593  686  793  839  850  908  990 1000

> sum(myresult$bootstrap_p<=0.05)

[1] 5

> which(myresult$bootstrap_p<=0.05)

[1] 104 152 321 425 700

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

[1] 1

> 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] 38

> which(myresult2$bootstrap_p<=0.05)

 [1]   3  11  12  52 106 141 159 189 202 230 273 276 299 303 321 334 349 358 385
[20] 411 429 432 473 512 534 563 587 590 610 639 685 719 756 822 884 951 953 983

> 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] 71

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

[1] 73

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

[1] 70

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

 [1]  10  16  51  63  64  81  88  92  98  99 101 102 132 136 144 147 156 187 213
[20] 214 223 242 246 250 255 259 294 295 304 314 324 326 335 349 350 423 426 448
[39] 468 476 488 520 539 566 574 580 582 607 619 636 653 656 657 670 674 693 725
[58] 802 825 831 837 868 881 889 897 898 911 932 953 979 992

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

 [1]  16  51  63  64  81  88  92  98  99 101 102 132 144 147 156 187 213 214 219
[20] 223 242 259 294 295 304 314 324 326 346 349 350 423 426 468 476 488 520 539
[39] 566 574 580 591 607 610 619 636 638 653 656 657 670 674 693 699 702 707 712
[58] 725 802 831 834 837 868 881 889 897 898 901 908 911 932 953 992

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

 [1]  10  16  51  64  81  88  90  92  98 101 102 132 136 144 147 187 194 214 219
[20] 223 242 250 259 287 294 295 304 324 335 349 423 426 448 460 468 476 488 504
[39] 520 539 566 574 580 607 619 636 653 656 657 668 670 674 693 707 725 802 825
[58] 826 831 837 868 881 897 898 901 911 932 970 979 992

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

```
[1] 15

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

[1] 15

> 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]  64  92  98 132 476 580 636 653 657 802 831 881 898 932 992

> which(con3_adjp<=0.05/3)

 [1]  64  88  98 304 324 468 476 580 868 898 992

> 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] 55

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

[1] 57

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

[1] 61

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

```
 [1]  15  18  27  34  36  73  90 122 127 166 174 186 199 222 232 244 258 260 299
[20] 312 354 375 382 384 400 455 459 477 511 515 523 551 552 561 572 608 616 621
[39] 632 647 649 660 666 700 709 751 761 818 867 889 891 904 962 964 980
```

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

```
 [1]  15  18  19  36  62  73  82  87  90 122 127 166 174 186 199 222 232 258 299
[20] 312 354 375 384 400 455 459 464 477 485 515 523 551 552 561 572 608 616 621
[39] 632 647 649 666 696 700 709 739 751 800 816 818 838 850 867 904 962 964 989
```

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

```
 [1]  15  18  27  33  34  36  62  73  82  87  90 122 127 166 174 186 199 222 244
[20] 258 260 262 299 312 354 375 400 455 459 464 477 485 506 511 515 523 548 549
[39] 551 552 561 572 608 616 621 622 632 647 649 666 700 708 709 751 756 816 818
[58] 867 904 962 980
```

```
> 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] 13
```

```
> 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] "/private/var/folders/r0/l4fjk6cj5xj0j3brt4bplpl40000gt/T/RtmpN1QDWY/Rinst204b4c9f21f7/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] 61

> sum(diff_results$bootstrap_p<=0.05)

[1] 71

> 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] 4

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

[1] 10

> 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]
5   cg00006414 0.07635468    0.07442468    0.15698040    0.08676092
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
346 cg00331237 0.05972383            NA    0.08204769    0.08345662
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
5      0.07982556    0.08111396    0.08271889    0.08045977
103    0.67349260    0.73510200    0.75715920    0.78981220
346    0.05372019    0.06241126    0.06955040    0.09140985
848    0.05230160    0.06119713    0.06542751    0.06240686
    diff_results$ordfit_t[diff_list_perm]
5                                -1.389459
103                              -2.268711
346                              -3.767916
848                              -2.314412
    diff_results$permutation_p[diff_list_perm]
5                                            0
103                                          0
346                                          0
848                                          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
189 cg00176210 0.28756520    0.39161870    0.44272520    0.44725330
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
632 cg00615377 0.11265030    0.16140570    0.19404450    0.17468600
743 cg00717862 0.07999436    0.07873347    0.06089359    0.06171374
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
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
189    0.34106080    0.33765930    0.41252110    0.37024890
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
632    0.12573100    0.14483660    0.16338240    0.20130510
743    0.07594936    0.09062161    0.06475791    0.07271878
911    0.08633986    0.06765189    0.09070268    0.12417730
928    0.04490690    0.04204062    0.05050039    0.05268215
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
146                              5.394750
189                             -3.097987
259                             -4.052697
280                              4.170347
285                             -3.093997
632                             -3.661161
743                              3.444684
911                             -3.621731
928                             -2.716443
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
146                                        0
189                                        0
259                                        0
280                                        0
285                                        0
632                                        0
743                                        0
911                                        0
928                                        0
```

979	0