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] 34

> which(myresult$permutation_p<=0.05)

 [1]  68 160 162 163 185 188 215 226 229 272 274 290 349 376 474 516 539 566 568
[20] 634 639 664 675 689 726 783 795 836 856 871 928 964 966 998

> sum(myresult$bootstrap_p<=0.05)

[1] 6

> which(myresult$bootstrap_p<=0.05)

[1] 131 208 303 576 631 831

> 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] 32

> which(myresult2$bootstrap_p<=0.05)

 [1]  12  69  96  97  99 166 180 244 264 277 294 303 306 322 395 477 516 517 520
[20] 577 596 599 615 660 757 761 782 790 902 921 967 976

> 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] 60

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

[1] 73

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

[1] 67

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

 [1]  13  60  63  67 101 114 177 183 211 245 262 282 290 301 312 330 336 348 375
[20] 377 387 392 405 406 417 421 424 477 484 511 532 546 550 574 576 577 593 600
[39] 630 633 635 636 657 675 687 704 708 720 735 738 747 759 810 819 833 882 891
[58] 896 978 998

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

 [1]   2  13  60  67 101 114 157 165 177 183 191 211 245 262 264 270 282 288 290
[20] 301 312 330 336 348 375 377 387 392 405 406 417 421 424 467 477 484 511 532
[39] 546 550 560 569 574 576 593 600 619 630 633 635 636 675 687 704 708 720 728
[58] 735 738 747 759 789 810 819 833 882 891 896 917 927 978 984 998

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

 [1]  13  43  60  67  89 101 114 165 177 183 211 262 270 282 290 301 312 330 336
[20] 348 375 377 387 392 405 406 417 419 421 424 428 467 477 484 511 532 546 550
[39] 561 569 574 576 593 600 603 613 630 633 635 671 675 704 708 720 722 738 747
[58] 759 784 810 833 882 891 896 917 978 998

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

```
[1] 13

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

[1] 11

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

[1] 20

> which(con2_adjp<=0.05/3)

 [1]  60 177 211 375 405 421 477 532 630 738 747

> which(con3_adjp<=0.05/3)

 [1]  13  60 177 183 211 312 392 405 421 477 484 511 532 574 593 630 720 738 747
[20] 998

> 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] 85

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

[1] 69

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

```
 [1]  45  47  56  65  96  97 137 139 156 172 175 195 225 270 301 312 313 316 322
[20] 326 329 330 366 392 415 434 435 446 492 538 545 562 566 571 582 586 592 623
[39] 629 635 654 668 684 716 770 772 814 819 821 822 865 924 945 948 954 964 966
[58] 986 990

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

 [1]  19  34  45  47  56  59  65  97 137 139 156 172 173 175 195 209 224 264 266
[20] 270 286 301 312 313 316 322 324 326 329 330 337 360 366 415 434 435 446 455
[39] 467 485 492 507 511 527 531 538 545 562 566 571 575 582 586 592 621 623 629
[58] 634 635 654 668 684 710 716 728 731 770 796 801 814 819 822 855 859 865 901
[77] 923 924 945 948 954 959 966 986 990

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

 [1]  19  23  45  56  65 137 139 156 169 172 174 175 225 270 286 301 312 316 322
[20] 326 329 330 366 415 434 435 446 455 485 492 514 538 545 561 562 566 571 575
[39] 582 586 592 621 623 629 635 654 684 686 710 716 731 770 796 814 819 821 822
[58] 859 865 881 923 924 948 954 959 964 966 981 990

> 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] 12

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

[1] 8
```

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] "F:/biocbuild/bbs-3.20-bioc/tmpdir/RtmpA1veBq/Rinst26542f0c1871/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] 51

> sum(diff_results$bootstrap_p<=0.05)

[1] 60

> 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] 3

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

[1] 12

> 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]
237 cg00215066 0.94926640    0.95311870    0.94634910    0.94561120
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
237    0.94837410    0.94665570    0.94089070    0.94600090
245    0.04208405    0.05284988    0.03775905    0.03955271
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_perm]
237                              1.419654
245                              1.962457
979                             -4.750997
    diff_results$permutation_p[diff_list_perm]
237                                         0
245                                         0
979                                         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]
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
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
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
346 cg00331237 0.05972383            NA    0.08204769    0.08345662
482 cg00468146 0.11144740    0.15416650    0.19827990    0.18517240
632 cg00615377 0.11265030    0.16140570    0.19404450    0.17468600
677 cg00651216 0.06825629    0.12529090    0.14409190    0.13907250
882 cg00858899 0.11427700    0.11919540    0.07690343    0.08321229
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
106    0.04693030    0.06837343    0.04534005    0.03709488
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
285    0.11693600    0.10650430    0.12281160    0.12310430
346    0.05372019    0.06241126    0.06955040    0.09140985
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
882    0.08961409    0.10730660    0.09203980    0.08726349
911    0.08633986    0.06765189    0.09070268    0.12417730
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
106                              3.100324
146                              5.394750
259                             -4.052697
280                              4.170347
285                             -3.093997
346                             -3.767916
482                             -3.212481
632                             -3.661161
677                             -3.387628
882                              3.179415
911                             -3.621731
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
106                                       0
146                                       0
259                                       0
280                                       0
285                                       0
```

346	0
482	0
632	0
677	0
882	0
911	0
979	0