Introduction to RBM package

Dongmei Li

November 25, 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] 28

> which(myresult$permutation_p<=0.05)

 [1]  80  86 122 125 130 206 212 214 222 238 253 324 335 351 392 396 417 487 541
[20] 542 575 587 669 705 731 839 968 990

> sum(myresult$bootstrap_p<=0.05)

[1] 0

> which(myresult$bootstrap_p<=0.05)

integer(0)

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

[1] 2

> 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] 9

> which(myresult2$bootstrap_p<=0.05)

[1] 286 293 359 363 400 605 703 863 981

> 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] 67

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

[1] 71

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

[1] 78

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

 [1]  10  14  41  56  71  89 105 111 133 150 166 179 205 206 216 225 228 256 264
[20] 279 286 291 298 338 361 399 419 435 442 450 470 484 508 515 553 557 563 582
[39] 605 635 640 644 679 683 697 701 702 715 739 742 762 767 773 779 781 799 808
[58] 846 876 907 916 924 958 960 968 988 989

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

 [1]   5  10  14  35  41  56  71 105 110 111 133 150 179 196 205 206 216 222 225
[20] 228 248 256 264 286 291 298 338 349 361 399 419 434 435 442 470 479 508 515
[39] 553 557 582 605 617 635 640 644 679 683 697 701 702 715 739 744 762 767 773
[58] 779 781 799 808 809 846 876 907 916 924 959 960 988 989

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

 [1]   5   9  10  27  32  35  40  41  56  71 105 110 111 133 150 166 205 206 214
[20] 216 222 225 228 248 256 264 279 291 298 338 361 399 419 434 435 470 484 508
[39] 515 529 533 553 557 563 572 582 605 617 623 629 635 640 644 679 683 697 701
[58] 702 715 730 739 762 767 773 779 795 808 809 846 907 916 924 958 959 960 968
[77] 988 989
```

```
> 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] 13

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

[1] 17

> which(con2_adjp<=0.05/3)

 [1] 105 133 256 291 361 419 644 701 702 715 739 767 773

> which(con3_adjp<=0.05/3)

 [1]  10 105 228 256 361 419 605 635 644 679 701 702 715 739 767 773 779

> 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] 25

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

[1] 44

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

[1] 44
```

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

 [1]  44  53 149 225 234 327 337 367 373 463 502 561 565 693 729 736 756 787 798
[20] 821 883 931 965 985 990

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

 [1]  19  44  53  97 140 149 225 234 264 304 327 337 367 373 431 435 457 459 463
[20] 495 502 508 561 565 597 642 659 693 729 736 743 756 787 798 821 833 855 872
[39] 896 931 965 978 990 995

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

 [1]  12  19  44  53  97 103 140 149 197 225 234 264 327 367 373 431 439 459 463
[20] 468 495 502 508 529 561 565 614 659 693 729 736 756 780 787 798 833 872 883
[39] 913 965 978 990 991 995

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

[1] 2

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

[1] 6

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

[1] 5
```

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/RtmpKhidEm/Rinst7ce0279e87bf/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] 47
```

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

```
[1] 42

> 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] 1

> 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]
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
103     0.6734926    0.73510200    0.75715920    0.78981220
848     0.0523016    0.06119713    0.06542751    0.06240686
851     0.5082024    0.34657470    0.66276570    0.64634510
    diff_results$ordfit_t[diff_list_perm]
103                             -2.343784
848                             -1.687144
851                             -2.986319
    diff_results$permutation_p[diff_list_perm]
103                                          0
848                                          0
851                                          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]
833 cg00814580 0.09348613    0.09619816     0.1201044     0.1153424
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
833     0.0957704     0.1159885     0.1286089      0.141112
    diff_results$ordfit_t[diff_list_boot]
833                            -3.278186
    diff_results$bootstrap_p[diff_list_boot]
833                                     0
```