Introduction to RBM package

Dongmei Li

October 29, 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]  42  59  72  81  82 119 158 189 271 472 507 522 524 553 556 597 616 632 646
[20] 747 750 763 914 978

> sum(myresult$bootstrap_p<=0.05)

[1] 1

> which(myresult$bootstrap_p<=0.05)

[1] 189

> 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] 17

> which(myresult2$bootstrap_p<=0.05)

 [1] 117 136 270 272 331 424 516 590 591 651 686 729 736 779 862 888 941

> 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] 55
```

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

```
[1] 66
```

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

```
[1] 62
```

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

```
 [1]   8  13  71  94  96  99 100 116 117 119 137 150 164 173 182 184 223 265 308
[20] 320 358 413 422 441 451 478 536 538 542 553 571 585 619 635 636 654 684 705
[39] 715 718 742 755 757 759 772 781 786 802 832 859 866 869 894 913 988
```

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

```
 [1]   8  13  78  81  94  96  99 100 116 117 119 137 150 173 182 184 212 223 265
[20] 276 281 308 320 325 358 413 422 441 451 478 492 536 538 542 553 559 561 619
[39] 635 636 654 664 684 699 705 715 718 742 755 757 759 772 778 781 786 789 802
[58] 822 826 859 869 870 894 913 939 988
```

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

```
 [1]   8  13  71  94  96  99 100 116 117 119 121 137 150 164 173 182 184 212 223
[20] 265 276 281 300 308 320 358 413 422 441 451 478 492 536 538 553 554 619 635
[39] 636 643 654 664 684 705 715 718 742 755 757 759 772 778 781 786 802 821 859
[58] 869 894 913 939 949
```

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

```
[1] 4

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

[1] 14

> 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]   8 116 184 308 320 451 553 636 718 755 757 781 802 894

> which(con3_adjp<=0.05/3)

 [1] 100 116 184 281 320 451 755 757 781 786 913

> 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] 57

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

[1] 57

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

[1] 51

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

```
 [1]  24  47  71  78 102 115 155 158 179 180 188 270 294 300 302 307 315 317 333
[20] 350 365 388 405 419 457 466 506 526 532 536 537 553 573 581 582 632 667 686
[39] 692 711 715 731 745 784 792 805 813 815 851 861 870 895 947 957 974 976 979
```

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

```
 [1]   4  24  40  71  78  95 102 115 158 179 188 194 218 264 292 300 302 315 333
[20] 350 365 387 388 401 434 457 466 526 532 537 553 563 569 581 582 632 656 667
[39] 686 711 715 731 745 780 784 792 805 813 815 861 870 895 919 947 957 974 976
```

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

```
 [1]  24  40  71  78 102 158 179 188 218 228 270 300 302 315 333 350 365 387 388
[20] 400 401 419 457 466 526 532 537 563 569 581 582 632 639 686 711 715 731 745
[39] 780 784 792 805 813 815 861 870 895 954 957 974 976
```

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

```
[1] 5
```

```
> 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 gemonewide 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] "/home/biocbuild/bbs-3.20-bioc/tmpdir/Rtmpm93DdM/Rinsta25a3f10d754/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] 47
```

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

```
[1] 69

> sum(diff_results$bootstrap_p<=0.05)

[1] 46

> 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] 9

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

[1] 7

> 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
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
103    0.67349260    0.73510200    0.75715920    0.78981220
131    0.17352980    0.12564280    0.18193170    0.20847670
259    0.04030003    0.03996053    0.05086962    0.05445672
627    0.04740551    0.05332965    0.05775211    0.05579710
764    0.90575890    0.88760470    0.90756300    0.90946790
848    0.05230160    0.06119713    0.06542751    0.06240686
851    0.50820240    0.34657470    0.66276570    0.64634510
928    0.04490690    0.04204062    0.05050039    0.05268215
    diff_results$ordfit_t[diff_list_perm]
```

```
19                              -2.547097
103                             -2.343784
131                             -3.562745
259                             -2.833203
627                             -1.797392
764                             -1.560713
848                             -1.687144
851                             -2.986319
928                             -1.982308
    diff_results$permutation_p[diff_list_perm]
19                                           0
103                                          0
131                                          0
259                                          0
627                                          0
764                                          0
848                                          0
851                                          0
928                                          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]
95  cg00081975 0.03633894    0.04975194    0.06024723    0.05598723
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
189 cg00176210 0.28756520    0.39161870    0.44272520    0.44725330
772 cg00743372 0.03922780    0.02919634    0.02187972    0.02568053
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
95     0.04561792    0.05115624    0.06068253    0.06168212
106    0.04693030    0.06837343    0.04534005    0.03709488
146    0.67191510    0.63137380    0.47929610    0.45428300
189    0.34106080    0.33765930    0.41252110    0.37024890
772    0.02796053    0.03512214    0.02575992    0.02093909
887    0.50259740    0.40111730    0.56646700    0.54552980
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
95                              -2.654324
106                              2.887876
146                              5.636263
189                             -3.232921
772                              1.885560
887                             -3.368752
```

```
979                             -4.968792
    diff_results$bootstrap_p[diff_list_boot]
95                                         0
106                                        0
146                                        0
189                                        0
772                                        0
887                                        0
979                                        0
```