Introduction to RBM package

Dongmei Li

May 27, 2021

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] 63

> which(myresult$permutation_p<=0.05)

 [1]   5   7  60  80  90 116 137 140 147 163 165 168 187 190 193 214 227 234 278
[20] 300 303 308 311 319 344 352 359 398 404 424 432 445 452 454 457 469 516 521
[39] 528 538 558 580 583 603 613 623 627 683 685 718 720 771 780 800 805 824 883
[58] 887 901 928 936 969 996

> sum(myresult$bootstrap_p<=0.05)

[1] 6

> which(myresult$bootstrap_p<=0.05)

[1]  73 344 352 817 824 887

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

[1] 5

> 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] 13

> which(myresult2$bootstrap_p<=0.05)

 [1]  50 176 192 278 290 336 413 416 447 474 525 655 999

> 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] 59

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

[1] 53

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

[1] 56

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

 [1]  14  19  33  42  54  91 130 164 169 176 187 208 214 229 233 234 242 254 269
[20] 275 335 342 350 377 384 422 434 446 491 529 542 549 570 574 586 600 605 624
[39] 632 640 642 666 705 734 738 761 778 783 810 826 843 847 851 859 860 863 886
[58] 897 949

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

 [1]  14  19  33  41  42 130 164 169 187 204 208 229 234 242 254 275 335 338 341
[20] 342 350 377 384 422 446 491 542 549 570 574 586 594 600 605 624 632 642 666
[39] 705 734 738 754 783 786 826 847 851 859 860 880 897 911 949

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

 [1]  19  31  33  42  86 130 164 169 187 204 208 220 229 234 242 254 269 335 341
[20] 342 350 384 422 434 444 491 527 542 549 570 574 585 586 594 624 632 642 666
[39] 696 705 738 754 761 778 783 786 810 826 843 847 851 859 860 868 949 976
```

```
> 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] 10

> 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] 130 187 242 350 384 491 624 642 826 949

> which(con3_adjp<=0.05/3)

[1] 187 242 350 384 624 949

> 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] 47

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

[1] 48

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

[1] 59
```

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

 [1]  31  34  49  69 114 125 131 144 155 160 232 238 266 276 316 337 339 385 397
[20] 420 437 456 462 490 499 552 554 582 597 621 672 682 717 762 767 783 796 814
[39] 820 851 897 911 936 945 969 982 994

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

 [1]   5  34  49  67  69 114 125 131 144 155 160 232 238 266 316 323 337 339 385
[20] 397 437 456 462 499 502 529 552 554 572 582 585 637 672 690 722 767 796 820
[39] 851 870 874 897 911 936 945 969 982 994

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

 [1]  34  49  69 114 125 131 144 150 155 160 204 232 238 266 276 316 321 323 337
[20] 339 352 366 385 397 420 449 456 462 552 554 567 569 572 582 585 621 672 717
[39] 722 749 762 767 783 796 814 817 820 851 854 870 874 897 911 929 936 945 956
[58] 969 982

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

[1] 3

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

[1] 8

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

[1] 4
```

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] "D:/biocbuild/bbs-3.14-bioc/tmpdir/RtmpQh374u/Rinst42e06de79ba/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] 76

> sum(diff_results$bootstrap_p<=0.05)

[1] 73

> 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
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
542 cg00520135 0.77510370    0.79688730    0.81833620    0.83043920
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
19     0.80831380    0.73306440    0.82968340    0.84917800
106    0.04693030    0.06837343    0.04534005    0.03709488
245    0.04208405    0.05284988    0.03775905    0.03955271
280    0.61920530    0.61925200    0.46753250    0.55632410
542    0.83062760    0.55544810    0.83402240    0.89514710
764    0.90575890    0.88760470    0.90756300    0.90946790
851    0.50820240    0.34657470    0.66276570    0.64634510
887    0.50259740    0.40111730    0.56646700    0.54552980
```

```
911    0.08633986    0.06765189    0.09070268    0.12417730
    diff_results$ordfit_t[diff_list_perm]
19                              -2.446404
106                              3.100324
245                              1.962457
280                              4.170347
542                             -1.775375
764                             -1.808081
851                             -2.841244
887                             -3.217939
911                             -3.621731
    diff_results$permutation_p[diff_list_perm]
19                                        0
106                                       0
245                                       0
280                                       0
542                                       0
764                                       0
851                                       0
887                                       0
911                                       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
743 cg00717862 0.07999436    0.07873347    0.06089359    0.06171374
804 cg00777121 0.04540701    0.05430304    0.04154242    0.04221162
833 cg00814580 0.09348613    0.09619816    0.12010440    0.11534240
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
743    0.07594936    0.09062161    0.06475791    0.07271878
804    0.04911277    0.04872797    0.04261405    0.04474881
833    0.09577040    0.11598850    0.12860890    0.14111200
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
743                              3.444684
```

```
804                                1.995220
833                               -3.428319
979                               -4.750997
    diff_results$bootstrap_p[diff_list_boot]
146                                       0
259                                       0
280                                       0
743                                       0
804                                       0
833                                       0
979                                       0
```