# Introduction to RBM package

Dongmei Li

November 6, 2019

Clinical and Translational Science Institute, University of Rochester School of Medicine and Dentistry, Rochester, NY 14642-0708

## Contents

**1 Overview** **1**

**2 Getting started** **2**

**3 `RBM_T` and `RBM_F` functions** **2**

**4 Ovarian cancer methylation example using the `RBM_T` function** **6**

## 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] 32

> which(myresult$permutation_p<=0.05)

 [1]  42 105 108 175 220 229 250 368 372 375 388 420 442 478 525 530 573 600 703
[20] 742 745 790 791 831 839 865 895 931 946 947 952 975

> sum(myresult$bootstrap_p<=0.05)

[1] 7

> which(myresult$bootstrap_p<=0.05)

[1] 222 234 318 742 791 865 931

> 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] 5

> which(myresult2$bootstrap_p<=0.05)

[1] 136 270 870 962 985

> 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] 77

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

[1] 67

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

[1] 66

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

 [1]  17  20  21  23  35  56  72  95 105 130 164 175 189 193 201 208 245 251 259
[20] 310 328 331 360 366 370 379 382 384 417 458 471 475 486 504 508 516 551 573
[39] 589 603 618 626 627 630 660 684 693 700 707 708 724 756 763 770 777 781 807
[58] 837 847 857 861 877 881 894 900 911 920 923 926 928 944 948 954 965 967 992
[77] 993

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

 [1]  12  17  20  21  23  49  72  95 105 129 157 175 189 193 251 259 328 331 360
[20] 366 379 384 399 409 458 471 475 480 486 504 508 516 551 573 589 603 626 627
[39] 630 660 684 693 700 708 724 756 763 770 777 813 837 850 856 877 881 894 911
[58] 920 926 944 948 954 965 967 978 992 993

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

 [1]  17  20  21  49  60  72  95 105 175 181 189 193 201 208 251 259 328 331 359
[20] 360 365 366 379 384 409 423 458 471 475 486 504 516 551 573 603 615 618 626
[39] 627 630 660 684 700 708 724 756 763 770 777 807 813 837 847 877 881 894 920
[58] 923 926 944 948 962 965 978 992 993
```

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

[1] 5

> 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] 16

> which(con2_adjp<=0.05/3)

 [1]  21  95 259 379 573 626 627 630 770 777 881 894 992

> which(con3_adjp<=0.05/3)

 [1]  21  72  95 105 193 259 366 379 573 627 630 708 777 881 920 992

> 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] 58

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

[1] 52

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

[1] 48
```

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

 [1]   2   8  20  47  79  85 167 169 173 188 190 199 268 286 321 322 350 382 409
[20] 418 446 450 457 468 469 474 494 497 542 552 591 596 598 627 638 656 688 701
[39] 712 734 760 763 773 777 801 806 809 816 861 872 897 898 907 914 922 954 959
[58] 973

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

 [1]   2  29  47  64  85 161 167 169 171 173 180 190 268 286 321 346 350 382 409
[20] 418 446 450 457 468 469 483 494 542 591 596 598 627 637 638 656 688 701 712
[39] 750 760 763 773 777 801 806 816 872 897 907 952 959 973

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

 [1]   2  20  29  47  85 102 161 171 177 190 199 268 286 350 409 418 446 450 457
[20] 468 469 474 483 494 542 552 553 591 596 627 638 656 688 701 712 734 760 763
[39] 773 777 780 801 806 816 872 897 907 959

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

[1] 4

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

[1] 5

> 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/tmp/RtmpSJQrwZ/Rinst157645a0fa8a0/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] 70

> sum(diff_results$bootstrap_p<=0.05)

[1] 72

> 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] 5

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

[1] 2

> 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]
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
627 cg00612467 0.04777553    0.03783457    0.05380982    0.05582291
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
83     0.04965089    0.04833366    0.03466159    0.04390894
103    0.67349260    0.73510200    0.75715920    0.78981220
245    0.04208405    0.05284988    0.03775905    0.03955271
627    0.04740551    0.05332965    0.05775211    0.05579710
764    0.90575890    0.88760470    0.90756300    0.90946790
    diff_results$ordfit_t[diff_list_perm]
83                               2.514109
103                             -2.268711
245                              1.962457
627                             -2.239498
764                             -1.808081
    diff_results$permutation_p[diff_list_perm]
```

```
83                                          0
103                                         0
245                                         0
627                                         0
764                                         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]
259 cg00234961 0.0419217    0.04321576     0.0570714    0.05327565
280 cg00260778 0.6431989    0.60488960     0.5673506    0.53150910
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
259    0.04030003    0.03996053    0.05086962    0.05445672
280    0.61920530    0.61925200    0.46753250    0.55632410
    diff_results$ordfit_t[diff_list_boot]
259                            -4.052697
280                             4.170347
    diff_results$bootstrap_p[diff_list_boot]
259                                      0
280                                      0
```