Introduction to RBM package

Dongmei Li

April 16, 2015

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:

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("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] 27
```

```
> which(myresult$permutation_p<=0.05)

 [1] 129 137 142 154 206 223 304 310 373 401 432 434 456 548 556 656 658 680 681
[20] 701 721 735 839 851 976 984 991

> sum(myresult$bootstrap_p<=0.05)

[1] 23

> which(myresult$bootstrap_p<=0.05)

 [1]  43  63  95 126 163 173 216 356 390 432 434 456 458 520 686 687 693 735 766
[20] 851 890 924 973

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

[1] 0

> 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] 20

> which(myresult2$bootstrap_p<=0.05)

 [1]   1  16  59  61  64 145 262 318 323 360 422 658 754 763 828 850 851 887 894
[20] 989

> 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] 71

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

[1] 50

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

[1] 75

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

 [1]  55  63  69  99 100 110 122 139 140 143 147 150 176 179 203 226 231 242 243
[20] 261 272 306 312 317 323 345 361 370 376 395 397 410 423 454 457 461 462 464
[39] 508 530 566 573 580 590 619 622 627 639 655 663 679 702 736 739 746 759 763
[58] 779 788 803 825 838 845 862 870 874 887 893 938 958 995

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

 [1]  69  99 100 110 122 139 140 150 179 226 231 242 243 272 312 317 323 345 361
[20] 370 374 376 395 397 410 423 454 457 464 508 530 566 573 574 622 627 639 655
[39] 663 679 702 736 779 838 845 874 887 938 958 995

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

 [1]  55  63  81  93  99 100 110 116 122 139 140 147 150 176 179 189 190 202 226
[20] 231 242 243 299 307 312 317 323 345 361 370 376 378 395 397 423 454 461 462
[39] 464 501 508 509 510 530 558 566 573 590 619 622 627 639 655 663 679 702 736
[58] 739 759 763 779 789 802 825 838 845 857 862 870 874 887 896 938 966 995

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

```
[1] 14

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

[1] 6

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

[1] 8

> which(con2_adjp<=0.05/3)

[1] 179 345 361 573 679 874

> which(con3_adjp<=0.05/3)

[1]  99 231 312 345 423 464 639 655

> 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] 50

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

[1] 50

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

[1] 40

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

```
 [1]   4  21  51 116 125 191 230 235 245 257 259 268 277 289 300 311 322 337 390
[20] 396 411 414 440 447 448 451 460 469 528 531 578 630 670 678 682 688 696 751
[39] 752 790 815 843 845 853 870 874 876 877 984 994
```

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

```
 [1]   4  21  51  88 102 116 125 133 154 168 191 200 222 230 235 245 257 259 300
[20] 311 322 337 382 390 396 414 440 469 517 528 578 597 682 688 696 737 751 752
[39] 790 815 843 845 865 870 874 876 889 929 984 994
```

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

```
 [1]   4  21  51 116 125 129 168 191 227 230 245 257 259 300 311 322 337 390 396
[20] 414 440 451 469 528 531 638 670 672 682 696 751 752 790 843 844 845 870 874
[39] 876 984
```

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

```
[1] 8
```

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

```
[1] 7
```

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

```
[1] 6
```

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/RtmptCGx9J/Rinst104c53480f1a0/RBM/data"
```

```
> data(ovarian_cancer_methylation)
> summary(ovarian_cancer_methylation)
```

```
        IlmnID        case1             case2             control1
 cg00000292:  1   Min.   :0.01058   Min.   :0.01138   Min.   :0.009103
 cg00002426:  1   1st Qu.:0.04111   1st Qu.:0.04290   1st Qu.:0.041543
 cg00003994:  1   Median :0.08284   Median :0.10438   Median :0.087042
 cg00005847:  1   Mean   :0.27397   Mean   :0.29086   Mean   :0.283729
 cg00006414:  1   3rd Qu.:0.52135   3rd Qu.:0.54436   3rd Qu.:0.558575
 cg00007981:  1   Max.   :0.97069   Max.   :0.96901   Max.   :0.970155
 (Other)   :994
    control2           case3             case4              control3
 Min.   :0.01019   Min.   :0.01108   Min.   :0.009753   Min.   :0.01278
 1st Qu.:0.04092   1st Qu.:0.04059   1st Qu.:0.041818   1st Qu.:0.04260
 Median :0.09042   Median :0.08527   Median :0.092807   Median :0.09362
 Mean   :0.28508   Mean   :0.28482   Mean   :0.283113   Mean   :0.27563
 3rd Qu.:0.57502   3rd Qu.:0.57300   3rd Qu.:0.558211   3rd Qu.:0.52240
 Max.   :0.96658   Max.   :0.97516   Max.   :0.963620   Max.   :0.95974
                   NA's   :1         NA's   :1
    control4
 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] 31
```

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

```
[1] 51

> sum(diff_results$bootstrap_p<=0.05)

[1] 32

> 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] 11

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

[1] 0

> 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      case1      case2   control1   control2      case3
66  cg00059424 0.02742616 0.02554150 0.03049395 0.02910234 0.02547771
76  cg00065408 0.03952223 0.03967472 0.04799694 0.04929252 0.04064262
110 cg00098239 0.02698720 0.02142180 0.01856646 0.01934917 0.02510008
129 cg00121158 0.03045297 0.02728770 0.03573820 0.03316130 0.02853104
245 cg00224508 0.04479948 0.04477529 0.04152814 0.04189373 0.04208405
432 cg00419564 0.03638860 0.03661916 0.04101457 0.04065540 0.03283922
460 cg00445824 0.14782870 0.16655800 0.14393210 0.13479670 0.20038750
660 cg00634577 0.03182804 0.03432180 0.03525499 0.03612398 0.03384897
690 cg00661202 0.01639344 0.01586308 0.01876500 0.02097005 0.01490915
764 cg00730260 0.90471270 0.90207400 0.91002680 0.91258610 0.90575890
772 cg00743372 0.03922780 0.03499011 0.02187972 0.02568053 0.02796053
         case4   control3   control4 diff_results$ordfit_t[diff_list_perm]
66  0.02523713 0.04478458 0.03391813                             -2.203752
76  0.03622954 0.04778213 0.04327211                             -3.115264
110 0.02291811 0.01784160 0.02109290                              1.976098
129 0.02993313 0.03318921 0.03345607                             -2.118418
245 0.05731476 0.03775905 0.03955271                              1.811314
432 0.03934095 0.04413387 0.04462037                             -2.407553
460 0.16185300 0.11630830 0.13912630                              2.993440
660 0.03062112 0.03502489 0.03710039                             -1.402274
```

```
690 0.01764273 0.01847447 0.01803320                             -1.267423
764 0.90290550 0.90756300 0.90946790                             -2.477349
772 0.03001808 0.02575992 0.02093909                              2.824452
    diff_results$permutation_p[diff_list_perm]
66                                            0
76                                            0
110                                           0
129                                           0
245                                           0
432                                           0
460                                           0
660                                           0
690                                           0
764                                           0
772                                           0
```

```
> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t
> print(sig_results_boot)
```

```
 [1] IlmnID
 [2] case1
 [3] case2
 [4] control1
 [5] control2
 [6] case3
 [7] case4
 [8] control3
 [9] control4
[10] diff_results$ordfit_t[diff_list_boot]
[11] diff_results$bootstrap_p[diff_list_boot]
<0 rows> (or 0-length row.names)
```