Introduction to RBM package

Dongmei Li

April 24, 2017

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] 95
```

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

 [1]   3  27  49  52  73  88  90  93  95 100 113 116 129 148 162 170 179 184 201
[20] 210 223 230 233 241 242 247 248 255 269 316 328 335 336 338 352 377 404 408
[39] 411 433 434 453 456 458 479 488 497 513 528 532 548 562 568 583 585 598 603
[58] 608 641 664 672 680 688 689 698 709 718 746 756 782 797 799 815 822 824 839
[77] 841 843 846 874 884 885 890 892 909 912 930 932 939 951 972 979 984 991 993

> sum(myresult$bootstrap_p<=0.05)

[1] 26

> which(myresult$bootstrap_p<=0.05)

 [1]  52  88  93 184 201 210 230 242 411 488 497 532 548 583 585 633 672 689 698
[20] 824 841 843 874 892 951 991

> 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] 25

> which(myresult2$bootstrap_p<=0.05)

 [1]  52  92 198 207 245 282 302 326 353 365 407 434 446 448 455 535 541 563 565
[20] 567 681 709 860 893 904

> 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] 66

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

[1] 66

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

[1] 59

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

 [1]    6   13   23   77   89   93  112  139  140  144  155  160  164  165  183
[16]  193  231  238  263  293  308  314  340  347  359  386  410  421  432  440
[31]  445  446  456  461  468  475  493  507  515  527  552  577  578  594  610
[46]  617  652  667  674  677  696  711  734  755  758  767  779  785  788  793
[61]  810  833  842  958  966 1000

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

 [1]   13   23   89   93  112  140  144  154  160  164  165  183  193  231  238
[16]  247  263  270  290  293  308  314  347  421  432  440  445  446  456  461
[31]  468  475  489  492  493  507  515  527  577  590  594  610  617  634  652
[46]  667  669  674  677  696  734  742  755  758  761  767  785  793  810  823
[61]  833  842  919  947  958 1000

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

```
 [1]    6   13   23   67   77   89   93  112  139  143  160  164  165  183  193
[16]  231  238  247  263  270  290  312  347  421  432  440  445  446  456  468
[31]  507  509  515  527  577  590  594  599  610  617  634  652  667  669  677
[46]  681  696  734  742  755  758  785  793  810  833  842  947  958 1000
```

```
> 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] 15
```

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

```
[1] 15
```

```
> which(con2_adjp<=0.05/3)
```

```
 [1]   13  112  165  183  193  421  446  527  577  610  667  734  755  810 1000
```

```
> which(con3_adjp<=0.05/3)
```

```
 [1]  193  238  247  421  507  527  577  610  617  667  696  734  810  842 1000
```

```
> 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] 62
```

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

```
[1] 57
```

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

[1] 53

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

 [1]  11  46  79 119 133 137 150 175 176 202 210 217 227 228 250 252 261 282 304
[20] 322 326 337 361 399 480 501 504 514 518 554 556 559 561 564 582 587 605 614
[39] 617 625 650 673 674 685 691 736 750 785 791 797 805 816 818 835 839 842 868
[58] 887 898 957 975 977

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

 [1]   5  11  38  46  79 119 133 150 175 176 202 210 217 227 241 252 255 258 261
[20] 282 304 337 364 399 470 476 480 499 504 514 554 559 561 564 582 587 614 617
[39] 650 674 685 691 731 736 781 785 791 805 816 818 835 842 887 898 932 975 977

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

 [1]  11  46  79  86 119 133 137 171 175 176 202 210 227 261 282 293 306 321 364
[20] 399 470 476 480 504 514 546 554 556 564 582 587 614 617 625 650 674 685 691
[39] 729 736 768 785 797 805 816 818 835 839 842 848 887 957 975

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

[1] 10

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

[1] 9

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

[1] 3
```

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/RtmpVkk5Zi/Rinst179f117fbc97b/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] 75

> sum(diff_results$bootstrap_p<=0.05)

[1] 39

> 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] 13

> 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]
16  cg00014085 0.05906804    0.04518973    0.04211710    0.03665208
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
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
437 cg00424946 0.04122172    0.04325330    0.03339863    0.02876798
772 cg00743372 0.03922780    0.02919634    0.02187972    0.02568053
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
931 cg00901704 0.05734342    0.04812868    0.04478214    0.03878488
939 cg00906183 0.03949030    0.04365079    0.03720015    0.03575748
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
16     0.04222944    0.05324246    0.03728026    0.04062589
```

```
83     0.04965089    0.04833366    0.03466159    0.04390894
103    0.67349260    0.73510200    0.75715920    0.78981220
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
437    0.03353116    0.03719167    0.03096761    0.03234779
772    0.02796053    0.03512214    0.02575992    0.02093909
848    0.05230160    0.06119713    0.06542751    0.06240686
851    0.50820240    0.34657470    0.66276570    0.64634510
931    0.04497277    0.05751033    0.03089829    0.04423603
939    0.03856975    0.06024309    0.03594439    0.03502819
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_perm]
16                                2.325659
83                                2.514109
103                              -2.268711
106                               3.100324
245                               1.962457
280                               4.170347
437                               2.102892
772                               2.416991
848                              -2.314412
851                              -2.841244
931                               2.464709
939                               1.762879
979                              -4.750997
    diff_results$permutation_p[diff_list_perm]
16                                          0
83                                          0
103                                         0
106                                         0
245                                         0
280                                         0
437                                         0
772                                         0
848                                         0
851                                         0
931                                         0
939                                         0
979                                         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
```

```
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
259    0.04030003    0.03996053    0.05086962    0.05445672
    diff_results$ordfit_t[diff_list_boot]
259                             -4.052697
    diff_results$bootstrap_p[diff_list_boot]
259                                     0
```