Introduction to RBM package

Dongmei Li

October 30, 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] 31
```

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

 [1]  85  93  97 121 125 149 169 195 211 293 325 345 375 440 459 462 475 530 628
[20] 640 653 656 720 775 812 826 848 879 903 987 990

> sum(myresult$bootstrap_p<=0.05)

[1] 5

> which(myresult$bootstrap_p<=0.05)

[1]  93 223 475 640 720

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

[1] 1

> 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]  16  42  58  75 228 276 287 305 318 415 446 510 637 659 696 699 784

> 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] 70

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

[1] 70

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

 [1]  25  36  37  70  81 114 124 132 138 142 145 198 211 212 217 228 242 246 255
[20] 263 273 318 321 332 334 343 353 380 384 420 438 485 500 502 582 612 619 620
[39] 628 649 701 715 720 723 739 746 750 760 766 793 809 849 874 908 909 911 941
[58] 992 996

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

 [1]  25  36  55  70  74  81 124 132 138 145 198 211 212 217 228 242 244 246 255
[20] 263 273 292 303 311 318 321 332 334 343 380 382 394 420 438 443 481 485 489
[39] 500 502 522 529 541 578 582 612 619 649 712 715 720 723 739 746 750 758 760
[58] 766 793 809 849 885 908 909 911 941 952 962 973 993

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

 [1]  25  36  70  81 124 132 138 145 211 212 217 228 242 244 255 273 282 303 311
[20] 321 329 332 334 343 353 380 394 397 404 420 438 443 473 481 485 500 502 525
[39] 541 559 578 582 599 612 628 649 703 712 715 720 723 739 746 750 760 766 790
[58] 793 809 849 874 908 909 911 941 962 992 993 995 996

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

```
[1] 10

> 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] 7

> which(con2_adjp<=0.05/3)

 [1]  36 124 138 145 212 228 311 334 438 649 766 793 909 911 962

> which(con3_adjp<=0.05/3)

[1] 228 485 649 723 793 909 992

> 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] 62

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

[1] 48

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

[1] 47

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

```
 [1]  13  14  23  33  34  41  53  69  73 119 126 150 157 163 164 186 191 205 215
[20] 218 277 280 293 301 328 381 408 409 417 434 459 482 509 531 532 549 551 565
[39] 581 588 593 596 649 652 661 709 737 739 743 754 756 762 794 799 839 840 845
[58] 850 868 914 994 998

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

 [1]  33  41  69  73  78 119 126 164 174 186 191 205 218 280 301 328 381 408 409
[20] 417 434 459 482 509 531 532 551 565 581 588 649 652 661 709 737 739 743 751
[39] 762 794 799 840 845 850 865 868 874 998

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

 [1]  23  33  69  73  78 119 150 157 164 186 191 194 218 280 301 381 408 409 417
[20] 434 459 482 509 531 532 589 593 649 652 661 709 719 737 739 743 751 756 762
[39] 794 799 840 845 850 868 872 892 998

> 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] 7

> 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/Rtmp7gmuYI/Rinst322525e421fa/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] 66

> sum(diff_results$bootstrap_p<=0.05)

[1] 65

> 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] 14

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

[1] 14

> 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
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
237 cg00215066 0.94926640    0.95311870    0.94634910    0.94561120
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
450 cg00432979 0.03681359    0.04515700    0.04374394    0.03683598
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
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
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
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
237    0.94837410    0.94665570    0.94089070    0.94600090
```

```
285    0.11693600    0.10650430    0.12281160    0.12310430
450    0.04419125    0.04409653    0.02839263    0.03410020
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
887    0.50259740    0.40111730    0.56646700    0.54552980
911    0.08633986    0.06765189    0.09070268    0.12417730
928    0.04490690    0.04204062    0.05050039    0.05268215
    diff_results$ordfit_t[diff_list_perm]
19                              -2.446404
83                               2.514109
103                             -2.268711
106                              3.100324
237                              1.419654
285                             -3.093997
450                              1.546114
627                             -2.239498
764                             -1.808081
848                             -2.314412
851                             -2.841244
887                             -3.217939
911                             -3.621731
928                             -2.716443
    diff_results$permutation_p[diff_list_perm]
19                                          0
83                                          0
103                                         0
106                                         0
237                                         0
285                                         0
450                                         0
627                                         0
764                                         0
848                                         0
851                                         0
887                                         0
911                                         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]
81  cg00071250 0.75466670    0.63467700    0.66801320    0.52600930
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
```

```
95  cg00081975 0.03633894    0.04975194    0.06024723    0.05598723
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
146 cg00134539 0.61101320    0.53321780    0.45999340    0.46787420
252 cg00230502 0.10061390    0.13517870    0.12538510    0.16304920
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
517 cg00499822 0.09723835    0.13925420    0.12969170    0.15998260
632 cg00615377 0.11265030    0.16140570    0.19404450    0.17468600
743 cg00717862 0.07999436    0.07873347    0.06089359    0.06171374
764 cg00730260 0.90471270    0.90542290    0.91002680    0.91258610
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
81     0.70381290    0.68161260    0.62075130    0.52060100
83     0.04965089    0.04833366    0.03466159    0.04390894
95     0.04561792    0.05115624    0.06068253    0.06168212
131    0.17352980    0.12564280    0.18193170    0.20847670
146    0.67191510    0.63137380    0.47929610    0.45428300
252    0.11970870    0.12036160    0.17423730    0.18155480
259    0.04030003    0.03996053    0.05086962    0.05445672
280    0.61920530    0.61925200    0.46753250    0.55632410
517    0.11009170    0.08752679    0.15305730    0.21607890
632    0.12573100    0.14483660    0.16338240    0.20130510
743    0.07594936    0.09062161    0.06475791    0.07271878
764    0.90575890    0.88760470    0.90756300    0.90946790
911    0.08633986    0.06765189    0.09070268    0.12417730
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
81                               2.729694
83                               2.514109
95                              -3.252063
131                             -3.451679
146                              5.394750
252                             -3.144056
259                             -4.052697
280                              4.170347
517                             -2.837883
632                             -3.661161
743                              3.444684
764                             -1.808081
911                             -3.621731
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
81                                        0
83                                        0
```

95	0
131	0
146	0
252	0
259	0
280	0
517	0
632	0
743	0
764	0
911	0
979	0