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] 7
```

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

[1]  81 200 341 460 472 744 811

> sum(myresult$bootstrap_p<=0.05)

[1] 4

> which(myresult$bootstrap_p<=0.05)

[1] 153 403 571 894

> 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] 21

> which(myresult2$bootstrap_p<=0.05)

 [1]  28  66 120 237 246 266 425 430 437 452 522 550 612 654 665 686 752 798 817
[20] 889 988

> 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] 76

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

[1] 76

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

 [1]   6   7  49  50  56  61  67  77 112 114 118 119 124 129 130 137 141 145 146
[20] 151 168 179 187 193 201 257 264 283 285 287 302 323 324 328 348 351 389 401
[39] 419 421 453 489 503 510 527 535 558 560 561 596 597 600 622 625 634 648 666
[58] 670 684 752 761 767 786 800 816 820 829 844 850 851 853 858 883 905 978 987
[77] 991

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

 [1]   6  49  50  56  61  67  77 114 119 124 130 137 141 145 146 151 179 187 193
[20] 201 257 264 278 279 283 285 287 314 323 324 328 351 381 389 419 421 453 489
[39] 496 503 504 510 527 535 560 561 596 597 600 622 625 634 641 645 648 666 670
[58] 752 761 767 786 803 816 820 829 832 850 851 853 854 858 883 905 978 987 991

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

 [1]   6   7  49  50  56  67  77 114 118 124 129 137 141 145 146 151 169 179 187
[20] 193 201 257 264 278 283 285 287 323 324 328 348 351 381 389 401 419 421 453
[39] 489 503 504 510 527 558 560 561 596 597 600 622 624 625 634 645 648 666 670
[58] 684 752 767 786 800 803 816 829 844 850 851 853 858 883 905 954 978 987 991
```

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

[1] 16

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

[1] 19

> 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]  77 124 145 179 187 283 285 287 324 328 419 421 504 561 596 625 648 666 752

> which(con3_adjp<=0.05/3)

 [1]   6  77 124 145 179 285 421 453 596 622 625 648 666 752 786 991

> 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] 66

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

[1] 53

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

[1] 46
```

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

 [1]   2  11  12  24  26  51  72  91 105 106 162 178 194 202 215 237 253 265 273
[20] 299 302 307 322 324 333 367 388 389 435 442 448 468 505 506 529 537 545 548
[39] 567 572 610 614 615 632 638 642 669 697 723 741 750 767 774 815 825 830 841
[58] 861 924 934 957 967 972 983 987 993

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

 [1]  11  12  14  24  28  30  51  64  91 105 189 194 215 237 253 273 288 299 302
[20] 320 324 350 388 389 398 435 448 468 505 506 537 567 572 614 615 632 637 638
[39] 642 669 695 697 723 741 750 774 825 861 934 953 957 983 987

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

 [1]  12  14  26  28  41 105 178 194 202 215 237 253 273 288 299 320 322 324 388
[20] 389 448 468 506 529 537 548 556 567 572 615 632 638 642 669 697 723 741 744
[39] 750 774 815 825 841 861 983 987

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

[1] 16

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

[1] 4

> 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] "C:/Users/biocbuild/bbs-3.6-bioc/tmpdir/RtmpqsYVfd/Rinst126c19be41a5/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] 60

> sum(diff_results$bootstrap_p<=0.05)

[1] 88

> 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] 6

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

[1] 16

> 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
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
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
    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
259    0.04030003    0.03996053    0.05086962    0.05445672
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
    diff_results$ordfit_t[diff_list_perm]
19                              -2.446404
83                               2.514109
259                             -4.052697
764                             -1.808081
```

```
851                              -2.841244
887                              -3.217939
    diff_results$permutation_p[diff_list_perm]
19                                          0
83                                          0
259                                         0
764                                         0
851                                         0
887                                         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]
83  cg00072216 0.04505377    0.04598964    0.04000674    0.03231534
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
131 cg00121904 0.15449580    0.17949750    0.23608110    0.24354150
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
283 cg00262415 0.03850601    0.04621248    0.03579758    0.03765227
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
520 cg00502442 0.03163993    0.03581662    0.02785063    0.02549502
632 cg00615377 0.11265030    0.16140570    0.19404450    0.17468600
743 cg00717862 0.07999436    0.07873347    0.06089359    0.06171374
772 cg00743372 0.03922780    0.02919634    0.02187972    0.02568053
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
911 cg00888479 0.07388961    0.07361080    0.10149800    0.09985076
928 cg00901493 0.03737166    0.03903724    0.04684618    0.04981432
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    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
131    0.17352980    0.12564280    0.18193170    0.20847670
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
283    0.03746915    0.04200230    0.03014699    0.02903290
285    0.11693600    0.10650430    0.12281160    0.12310430
520    0.03111720    0.03189393    0.02415307    0.02941176
632    0.12573100    0.14483660    0.16338240    0.20130510
743    0.07594936    0.09062161    0.06475791    0.07271878
772    0.02796053    0.03512214    0.02575992    0.02093909
851    0.50820240    0.34657470    0.66276570    0.64634510
911    0.08633986    0.06765189    0.09070268    0.12417730
928    0.04490690    0.04204062    0.05050039    0.05268215
```

```
979    0.11848510    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_boot]
83                               2.514109
103                             -2.268711
131                             -3.451679
146                              5.394750
259                             -4.052697
280                              4.170347
283                              2.057672
285                             -3.093997
520                              1.873471
632                             -3.661161
743                              3.444684
772                              2.416991
851                             -2.841244
911                             -3.621731
928                             -2.716443
979                             -4.750997
    diff_results$bootstrap_p[diff_list_boot]
83                                        0
103                                       0
131                                       0
146                                       0
259                                       0
280                                       0
283                                       0
285                                       0
520                                       0
632                                       0
743                                       0
772                                       0
851                                       0
911                                       0
928                                       0
979                                       0
```