Introduction to RBM package

Dongmei Li

October 14, 2015

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:

```
> 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] 47
```

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

 [1]  46  48  56  68  69  79  99 107 110 129 140 142 184 205 252 255 257 270 277
[20] 286 341 356 371 443 449 465 483 496 502 506 542 607 676 681 684 738 756 759
[39] 792 807 844 894 912 916 925 955 971

> sum(myresult$bootstrap_p<=0.05)

[1] 6

> which(myresult$bootstrap_p<=0.05)

[1] 465 467 506 705 894 925

> 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] 16

> which(myresult2$bootstrap_p<=0.05)

 [1]  50  63 135 203 560 608 647 714 721 781 813 845 887 924 985 995

> bootstrap2_adjp <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adjp<=0.05)

[1] 1
```

- 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] 84

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

[1] 65

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

[1] 86

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

 [1]   2  47  70  73  83  91 105 108 109 121 125 139 155 164 167 175 180 198 239
[20] 249 253 265 289 298 313 323 335 337 359 368 375 386 394 418 436 448 456 471
[39] 473 510 526 581 591 599 604 617 623 625 632 638 653 658 663 675 690 707 720
[58] 725 727 728 730 803 820 823 825 844 852 865 867 893 901 912 931 940 953 959
[77] 961 966 977 978 979 982 985 994

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

 [1]   2  47  70  73  83  91 105 108 110 121 125 139 155 167 180 198 239 249 253
[20] 265 289 298 313 323 335 337 368 375 386 418 436 456 471 581 591 617 623 625
[39] 632 653 658 675 685 690 707 725 727 728 796 803 820 823 825 852 865 893 901
[58] 931 940 953 959 979 982 985 994

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

 [1]   2  47  56  70  73  83 105 108 110 125 139 164 167 175 180 198 234 239 249
[20] 253 265 273 289 298 313 323 335 337 359 368 375 383 386 394 418 436 456 469
[39] 471 493 581 591 598 599 604 617 623 625 632 638 653 658 663 675 690 707 720
[58] 725 727 728 730 795 796 803 823 825 844 852 865 893 901 908 912 918 931 940
[77] 953 959 961 963 977 978 979 982 985 994
```

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

[1] 24

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

[1] 4

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

[1] 22

> which(con2_adjp<=0.05/3)

[1] 337 375 675 979

> which(con3_adjp<=0.05/3)

 [1] 239 313 337 375 418 456 471 581 623 625 632 653 658 675 707 823 865 940 959
[20] 979 985 994

> 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] 54

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

[1] 40

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

[1] 45
```

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

 [1]  13  22  38  56  57  91 104 111 115 138 139 187 215 224 248 264 268 309 335
[20] 345 352 405 416 419 446 460 461 468 483 499 521 542 545 549 607 640 710 716
[39] 719 720 749 772 776 789 801 831 832 835 837 867 893 899 910 942

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

 [1]  22  56  57 104 115 138 187 248 249 268 302 309 335 345 352 405 416 419 443
[20] 448 460 461 468 542 545 549 603 607 640 710 716 720 749 832 835 837 893 899
[39] 910 942

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

 [1]  13  22  56  57  91 115 138 187 210 224 248 249 309 335 352 384 405 416 419
[20] 460 521 542 545 549 579 607 640 710 716 720 749 772 789 801 806 823 831 832
[39] 835 837 867 889 893 910 942

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

[1] 7

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

[1] 6

> 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/RtmpVWrnEk/Rinst3ce03af2e08b/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] 54

> sum(diff_results$bootstrap_p<=0.05)

[1] 54

> 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] 7

> 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]
81  cg00071250 0.75466670    0.63467700    0.66801320    0.52600930
103 cg00094319 0.73784280    0.73532960    0.75574900    0.73830220
106 cg00095674 0.07076291    0.05045181    0.03861991    0.03337576
280 cg00260778 0.64319890    0.60488960    0.56735060    0.53150910
848 cg00826384 0.05721674    0.05612171    0.06644259    0.06358381
851 cg00830029 0.58362500    0.59397870    0.64739610    0.67269640
979 cg00945507 0.13432250    0.23854600    0.34749760    0.28903340
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
81      0.7038129    0.68161260    0.62075130    0.52060100
103     0.6734926    0.73510200    0.75715920    0.78981220
106     0.0469303    0.06837343    0.04534005    0.03709488
280     0.6192053    0.61925200    0.46753250    0.55632410
848     0.0523016    0.06119713    0.06542751    0.06240686
851     0.5082024    0.34657470    0.66276570    0.64634510
979     0.1184851    0.16653850    0.30718420    0.26624740
    diff_results$ordfit_t[diff_list_perm]
81                               2.729694
103                             -2.268711
```

```
106                               3.100324
280                               4.170347
848                              -2.314412
851                              -2.841244
979                              -4.750997
    diff_results$permutation_p[diff_list_perm]
81                                        0
103                                       0
106                                       0
280                                       0
848                                       0
851                                       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]
911 cg00888479 0.07388961     0.0736108      0.101498    0.09985076
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
911    0.08633986    0.06765189    0.09070268     0.1241773
    diff_results$ordfit_t[diff_list_boot]
911                              -3.621731
    diff_results$bootstrap_p[diff_list_boot]
911                                       0
```