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

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] 25
```

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

 [1] 107 158 181 316 330 419 434 457 517 559 573 591 601 610 633 658 672 746 765
[20] 805 861 899 918 935 938

> sum(myresult$bootstrap_p<=0.05)

[1] 4

> which(myresult$bootstrap_p<=0.05)

[1] 419 518 555 939

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

[1] 10

> 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] 9

> which(myresult2$bootstrap_p<=0.05)

[1]  37  63 218 234 441 453 525 752 792

> 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] 83

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

[1] 86

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

[1] 59

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

 [1]   2  21  26  40  55  76  79  90  94  97 101 121 126 132 152 168 169 177 184
[20] 185 198 218 219 242 308 316 333 347 357 385 406 411 412 428 443 447 452 477
[39] 492 520 531 567 580 581 590 599 614 628 635 639 643 648 681 690 692 698 703
[58] 716 720 747 768 772 781 788 798 800 803 807 818 828 857 877 878 885 896 902
[77] 910 912 928 939 959 972 992

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

 [1]   2  21  26  33  40  70  76  79  90  94  97 101 121 126 152 168 169 177 184
[20] 185 198 219 242 299 308 316 333 347 357 367 385 406 411 412 428 443 447 492
[39] 499 520 531 557 567 580 581 590 599 614 628 635 643 648 681 690 692 698 703
[58] 716 720 747 768 772 781 788 795 800 803 807 828 831 870 877 878 885 887 895
[77] 896 902 910 925 928 939 959 968 972 992

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

 [1]  21  26  33  40  79  90  94  97 101 121 126 152 168 177 184 214 219 242 308
[20] 333 347 357 385 406 412 428 443 520 531 567 580 581 599 628 633 635 643 681
[39] 690 692 698 703 716 720 768 772 781 788 798 803 807 828 870 896 902 910 928
[58] 939 959
```

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

[1] 8

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

[1] 16

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

[1] 1

> which(con2_adjp<=0.05/3)

 [1]  90 101 126 308 333 428 443 599 635 681 690 788 902 910 928 959

> which(con3_adjp<=0.05/3)

[1] 177

> 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] 55

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

[1] 46

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

[1] 43
```

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

 [1]   6   9  10  16  27  41 157 169 209 250 266 272 279 281 308 333 347 366 370
[20] 399 400 418 425 448 475 481 523 528 556 565 594 608 619 623 626 637 643 667
[39] 710 726 738 745 765 773 778 782 787 790 839 886 894 908 927 969 982

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

 [1]   6   9  23  28  41  49 136 169 209 250 266 272 281 329 333 347 366 370 400
[20] 418 425 448 475 480 481 528 545 556 558 565 594 637 643 667 680 726 738 778
[39] 782 787 790 839 894 908 927 969

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

 [1]   6  27  41 169 209 250 266 272 279 283 329 333 341 347 366 370 400 418 425
[20] 441 448 475 481 528 556 558 594 608 637 643 726 738 745 778 782 787 839 859
[39] 894 908 927 969 982

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

[1] 5

> 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] 4
```

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/RtmpDUN1co/Rinst12b335dee13f5/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] 50

> sum(diff_results$bootstrap_p<=0.05)

[1] 61

> 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] 3

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

[1] 5

> 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]
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
285 cg00263760 0.09050395    0.10197760    0.14801710    0.12242400
887 cg00862290 0.43640520    0.54047160    0.60786800    0.56325950
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
245    0.04208405    0.05284988    0.03775905    0.03955271
285    0.11693600    0.10650430    0.12281160    0.12310430
887    0.50259740    0.40111730    0.56646700    0.54552980
    diff_results$ordfit_t[diff_list_perm]
245                              1.962457
285                             -3.093997
887                             -3.217939
    diff_results$permutation_p[diff_list_perm]
245                                         0
285                                         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
95  cg00081975 0.03633894    0.04975194    0.06024723    0.05598723
259 cg00234961 0.04192170    0.04321576    0.05707140    0.05327565
520 cg00502442 0.03163993    0.03581662    0.02785063    0.02549502
804 cg00777121 0.04540701    0.05430304    0.04154242    0.04221162
    exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
83     0.04965089    0.04833366    0.03466159    0.04390894
95     0.04561792    0.05115624    0.06068253    0.06168212
259    0.04030003    0.03996053    0.05086962    0.05445672
520    0.03111720    0.03189393    0.02415307    0.02941176
804    0.04911277    0.04872797    0.04261405    0.04474881
    diff_results$ordfit_t[diff_list_boot]
83                               2.514109
95                              -3.252063
259                             -4.052697
520                              1.873471
804                              1.995220
    diff_results$bootstrap_p[diff_list_boot]
83                                       0
95                                       0
259                                      0
520                                      0
804                                      0
```