

Introduction to RBM package

Dongmei Li

October 29, 2019

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	7

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:

```
> if (!requireNamespace("BiocManager", quietly=TRUE))
+   install.packages("BiocManager")
> BiocManager::install("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 Benjamini-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
<code>ordfit_t</code>	1000	-none-	numeric
<code>ordfit_pvalue</code>	1000	-none-	numeric
<code>ordfit_beta0</code>	1000	-none-	numeric
<code>ordfit_beta1</code>	1000	-none-	numeric
<code>permutation_p</code>	1000	-none-	numeric
<code>bootstrap_p</code>	1000	-none-	numeric

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

```

[1] 74

> which(myresult$permutation_p<=0.05)

[1] 1 17 37 40 43 53 79 91 123 132 142 144 170 184 206 221 235 254 258
[20] 279 280 298 314 321 337 342 346 359 414 422 483 496 503 511 513 514 523 531
[39] 539 540 578 599 616 620 621 627 638 647 652 671 682 694 704 710 721 736 784
[58] 808 834 842 847 849 860 867 869 870 889 939 944 952 953 974 978 992

> sum(myresult$bootstrap_p<=0.05)

[1] 6

> which(myresult$bootstrap_p<=0.05)

[1] 320 483 523 869 910 953

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

[1] 4

> bootstrap_adj_p <- p.adjust(myresult$bootstrap_p, "BH")
> sum(bootstrap_adj_p<=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] 8

> which(myresult2$bootstrap_p<=0.05)

[1] 88 139 234 523 527 595 765 806

> bootstrap2_adj_p <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adj_p<=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] 85

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

[1] 80

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

[1] 84

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

[1]  16  34  48  51  53  69  70  79  88  97 100 101 113 139 153 154 156 176 177
[20] 182 192 200 219 230 241 257 276 277 299 320 340 345 358 361 382 408 415 422
[39] 433 436 440 444 452 457 465 473 482 495 497 523 524 534 537 558 572 579 589
[58] 623 626 646 651 667 672 686 706 754 760 775 784 804 806 809 843 874 878 897
[77] 908 917 926 931 934 945 958 964 999

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

[1]  16  34  48  52  53  67  70  79  88  97 100 111 153 154 156 177 179 182 192
[20] 200 219 257 276 277 299 320 340 345 358 361 382 408 415 433 436 440 444 465
[39] 473 482 495 497 517 523 534 537 558 559 572 579 589 626 646 651 667 672 677
[58] 681 686 706 719 754 760 775 784 804 806 809 843 878 897 907 916 926 931 945
[77] 958 964 992 999

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

```

```

[1] 16 34 48 51 52 53 70 79 88 97 100 101 153 154 156 162 177 179 182
[20] 192 257 276 277 288 299 320 340 345 358 361 382 408 415 433 436 440 444 452
[39] 465 473 482 495 497 517 523 524 525 534 558 559 572 579 589 626 646 651 672
[58] 686 706 716 719 754 760 775 784 795 804 806 809 820 843 878 897 907 908 926
[77] 931 934 945 958 964 985 993 999

```

```

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

```

```
[1] 20
```

```

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

```

```
[1] 21
```

```

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

```

```
[1] 24
```

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

```

[1] 48 79 154 277 320 340 358 382 408 465 495 572 579 651 754 775 784 843 878
[20] 926 958

```

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

```

[1] 48 154 156 192 276 277 320 340 358 361 444 473 523 572 579 651 672 686 706
[20] 775 784 809 897 926

```

```

> 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] 64
```

```

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

[1] 38

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

[1] 52

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

[1] 7 21 31 34 42 50 56 79 123 124 125 131 142 154 172 177 195 205 229
[20] 266 308 310 323 330 331 332 340 347 367 394 397 403 408 426 443 482 527 591
[39] 638 643 650 673 683 703 714 797 800 802 824 827 835 844 850 870 913 918 945
[58] 947 948 950 959 975 988 998

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

[1] 7 21 28 42 50 51 154 172 177 195 205 229 266 308 310 330 331 347 367
[20] 394 395 397 408 426 643 673 703 714 800 802 835 844 850 870 913 948 950 998

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

[1] 7 21 28 50 123 142 154 172 177 193 195 205 229 266 308 310 330 331 332
[20] 347 367 394 397 403 408 426 432 443 591 643 650 673 703 714 797 800 802 833
[39] 835 844 847 850 870 913 918 945 947 948 950 959 975 998

> 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] 2

> 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 genome-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 illustration purpose, we chose the first 1000 loci in 8 randomly selected women with 4 ovarian 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] "/tmp/RtmpPdoOnE/Rinst24e0581cf2e8/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] 69

> sum(diff_results$bootstrap_p<=0.05)

[1] NA

> 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] 18

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

[1] NA

> 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
95	cg00081975	0.03633894	0.04975194	0.06024723	0.05598723
106	cg00095674	0.07076291	0.05045181	0.03861991	0.03337576

131	cg00121904	0.15449580	0.17949750	0.23608110	0.24354150
146	cg00134539	0.61101320	0.53321780	0.45999340	0.46787420
237	cg00215066	0.94926640	0.95311870	0.94634910	0.94561120
245	cg00224508	0.04479948	0.04972043	0.04152814	0.04189373
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
437	cg00424946	0.04122172	0.04325330	0.03339863	0.02876798
764	cg00730260	0.90471270	0.90542290	0.91002680	0.91258610
772	cg00743372	0.03922780	0.02919634	0.02187972	0.02568053
804	cg00777121	0.04540701	0.05430304	0.04154242	0.04221162
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
931	cg00901704	0.05734342	0.04812868	0.04478214	0.03878488
	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	
95	0.04561792	0.05115624	0.06068253	0.06168212	
106	0.04693030	0.06837343	0.04534005	0.03709488	
131	0.17352980	0.12564280	0.18193170	0.20847670	
146	0.67191510	0.63137380	0.47929610	0.45428300	
237	0.94837410	0.94665570	0.94089070	0.94600090	
245	0.04208405	0.05284988	0.03775905	0.03955271	
259	0.04030003	0.03996053	0.05086962	0.05445672	
280	0.61920530	0.61925200	0.46753250	0.55632410	
437	0.03353116	0.03719167	0.03096761	0.03234779	
764	0.90575890	0.88760470	0.90756300	0.90946790	
772	0.02796053	0.03512214	0.02575992	0.02093909	
804	0.04911277	0.04872797	0.04261405	0.04474881	
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	
931	0.04497277	0.05751033	0.03089829	0.04423603	
	diff_results\$ordfit_t[diff_list_perm]				
19		-2.446404			
83		2.514109			
95		-3.252063			
106		3.100324			
131		-3.451679			
146		5.394750			
237		1.419654			
245		1.962457			
259		-4.052697			
280		4.170347			
437		2.102892			

```

764          -1.808081
772          2.416991
804          1.995220
851         -2.841244
887         -3.217939
911         -3.621731
931          2.464709

```

```

diff_results$permutation_p[diff_list_perm]
19          0
83          0
95          0
106         0
131         0
146         0
237         0
245         0
259         0
280         0
437         0
764         0
772         0
804         0
851         0
887         0
911         0
931         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]
106	cg00095674	0.07076291	0.05045181	0.03861991	0.03337576
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
285	cg00263760	0.09050395	0.10197760	0.14801710	0.12242400
743	cg00717862	0.07999436	0.07873347	0.06089359	0.06171374
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
979	cg00945507	0.13432250	0.23854600	0.34749760	0.28903340
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
106	0.04693030	0.06837343	0.04534005	0.03709488	
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	

285	0.11693600	0.10650430	0.12281160	0.12310430
743	0.07594936	0.09062161	0.06475791	0.07271878
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
979	0.11848510	0.16653850	0.30718420	0.26624740

diff_results\$ordfit_t[diff_list_boot]

106	3.100324
146	5.394750
259	-4.052697
280	4.170347
285	-3.093997
743	3.444684
887	-3.217939
911	-3.621731
928	-2.716443
979	-4.750997

diff_results\$bootstrap_p[diff_list_boot]

106	0
146	0
259	0
280	0
285	0
743	0
887	0
911	0
928	0
979	0