

Introduction to RBM package

Dongmei Li

November 6, 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	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:

```
> 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
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] 16

> which(myresult$permutation_p<=0.05)

[1] 12 51 76 112 154 168 350 351 395 396 595 630 689 781 962 997

> sum(myresult$bootstrap_p<=0.05)

[1] 20

> which(myresult$bootstrap_p<=0.05)

[1] 27 44 71 79 157 202 371 414 452 490 491 503 553 711 734 783 839 866 895
[20] 962

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

[1] 3

> 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] 15

> which(myresult2$bootstrap_p<=0.05)

[1] 113 222 233 285 369 381 471 517 596 671 685 787 874 897 938

> 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] 75

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

[1] 88

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

[1] 81

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

[1] 34 41 48 78 91 105 112 126 129 136 158 161 170 173 198 219 253 263 313
[20] 344 347 366 372 378 382 405 413 436 449 471 475 488 497 510 523 527 537 555
[39] 556 573 597 613 614 616 620 636 648 654 661 699 730 756 764 768 777 779 789
[58] 804 813 844 854 857 869 871 875 885 886 908 928 936 947 952 966 967 984

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

[1] 7 18 41 48 51 91 95 105 112 126 129 136 158 161 165 170 171 173 181
[20] 198 215 219 251 253 263 290 292 301 313 337 344 347 366 372 378 382 386 405
[39] 413 436 449 475 488 497 501 510 523 527 537 539 555 556 573 595 597 613 614
[58] 616 636 648 654 661 699 730 753 756 764 768 779 789 813 854 857 869 871 875
[77] 885 886 908 928 947 952 965 966 967 981 984 996

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

[1] 41 48 78 91 105 112 126 129 136 158 161 170 171 173 177 181 198 219 251
[20] 253 256 263 313 344 366 372 378 382 405 413 436 449 471 475 488 497 501 510
[39] 523 527 537 555 556 573 595 597 613 614 616 627 636 648 654 661 699 715 730
[58] 753 764 768 777 779 789 804 813 828 844 854 857 869 871 885 886 928 936 947
[77] 952 965 966 967 981

```

```

> con1_adj_p <- p.adjust(myresult_F$permutation_p[, 1], "BH")
> sum(con1_adj_p<=0.05/3)

[1] 22

> con2_adj_p <- p.adjust(myresult_F$permutation_p[, 2], "BH")
> sum(con2_adj_p<=0.05/3)

[1] 41

> con3_adj_p <- p.adjust(myresult_F$permutation_p[, 3], "BH")
> sum(con3_adj_p<=0.05/3)

[1] 20

> which(con2_adj_p<=0.05/3)

[1] 41 48 91 105 158 161 170 173 198 219 263 313 366 413 475 497 510 523 527
[20] 537 555 556 573 597 614 616 636 648 654 661 699 756 779 789 813 869 871 928
[39] 952 966 984

> which(con3_adj_p<=0.05/3)

[1] 41 161 170 173 219 263 313 366 378 413 597 614 616 648 779 813 857 869 871
[20] 952

> 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] 68

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

[1] 62

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

```

```

[1] 48

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

[1] 1 24 29 40 51 66 69 70 77 83 86 96 115 132 148 210 219 221 222
[20] 226 229 233 253 282 283 288 293 306 341 364 370 377 392 396 456 461 514 530
[39] 532 536 538 546 553 611 642 646 656 679 690 691 692 693 696 710 711 726 737
[58] 767 772 776 779 799 867 919 938 963 988 999

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

[1] 1 24 40 51 66 68 69 70 77 83 86 96 115 125 132 195 210 219 221
[20] 222 226 229 236 253 283 288 293 306 341 364 377 392 456 514 530 532 536 538
[39] 546 553 611 642 646 690 691 692 693 696 701 710 726 776 779 799 840 867 869
[58] 919 938 963 988 999

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

[1] 24 29 66 69 70 77 86 96 115 116 210 219 221 229 282 283 288 293 306
[20] 364 377 392 396 514 530 532 538 546 553 611 642 646 690 691 692 693 701 710
[39] 726 767 779 867 876 919 938 963 988 999

> con21_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 1], "BH")
> sum(con21_adj_p<=0.05/3)

[1] 10

> con22_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 2], "BH")
> sum(con22_adj_p<=0.05/3)

[1] 7

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=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 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] "C:/Users/biocbuild/bbs-3.11-bioc/tmpdir/RtmpCuW8d9/Rinstf2442186803/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] 67

> sum(diff_results$bootstrap_p<=0.05)

[1] 51

> 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] 10

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

[1] 4

> 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
245	cg00224508	0.04479948	0.04972043	0.04152814	0.04189373
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
911	cg00888479	0.07388961	0.07361080	0.10149800	0.09985076
	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	

245	0.04208405	0.05284988	0.03775905	0.03955271
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
911	0.08633986	0.06765189	0.09070268	0.12417730

diff_results\$ordfit_t[diff_list_perm]

19	-2.446404
83	2.514109
95	-3.252063
106	3.100324
245	1.962457
259	-4.052697
764	-1.808081
851	-2.841244
887	-3.217939
911	-3.621731

diff_results\$permutation_p[diff_list_perm]

19	0
83	0
95	0
106	0
245	0
259	0
764	0
851	0
887	0
911	0

```
> sig_results_boot <- cbind(ovarian_cancer_methylation[diff_list_boot, ], diff_results$ordfit_t[diff_list_boot, ])
> print(sig_results_boot)
```

	IlmnID	Beta	exmdata2[, 2]	exmdata3[, 2]	exmdata4[, 2]
200	cg00183916	0.03525946	0.03984548	0.02765822	0.02789838
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
882	cg00858899	0.11427700	0.11919540	0.07690343	0.08321229
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
200	0.03034811	0.04302129	0.02753873	0.03067437	
259	0.04030003	0.03996053	0.05086962	0.05445672	
280	0.61920530	0.61925200	0.46753250	0.55632410	
882	0.08961409	0.10730660	0.09203980	0.08726349	
	diff_results\$ordfit_t[diff_list_boot]				
200	2.272449				
259	-4.052697				
280	4.170347				

882	3.179415
	diff_results\$bootstrap_p[diff_list_boot]
200	0
259	0
280	0
882	0