

# Introduction to RBM package

Dongmei Li

April 28, 2020

Clinical and Translational Science Institute, University of Rochester School of Medicine and  
Dentistry, Rochester, NY 14642-0708

## Contents

<b>1 Overview</b>	<b>1</b>
<b>2 Getting started</b>	<b>2</b>
<b>3 RBM_T and RBM_F functions</b>	<b>2</b>
<b>4 Ovarian cancer methylation example using the RBM_T function</b>	<b>6</b>

## 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] 116

> which(myresult$permutation_p<=0.05)

[1] 1 8 17 19 41 51 58 60 65 67 79 98 108 116 119 126 131 134
[19] 166 172 174 177 187 192 193 196 199 205 230 240 245 263 268 273 280 285
[37] 327 334 335 355 364 366 380 381 393 394 404 427 435 441 452 455 457 463
[55] 473 476 480 481 483 491 499 510 512 527 532 559 572 605 609 611 628 653
[73] 657 667 674 686 706 714 716 738 741 750 751 754 756 774 792 794 798 818
[91] 826 832 840 848 851 863 878 883 889 897 898 904 907 908 910 918 923 924
[109] 945 951 952 963 966 968 993 997

> sum(myresult$bootstrap_p<=0.05)

[1] 6

> which(myresult$bootstrap_p<=0.05)

[1] 192 199 205 489 510 898

> 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] 41

> which(myresult2$bootstrap_p<=0.05)

[1] 55 57 84 90 98 116 121 183 188 223 277 297 300 322 378 401 437 455 457
[20] 462 469 474 481 502 519 526 565 568 571 572 655 679 693 723 756 770 856 858
[39] 885 904 910

```

```
> 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] 58
```

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

```
[1] 56
```

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

```
[1] 64
```

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

```
[1] 37 53 59 78 113 117 122 136 149 154 163 196 214 223 275 277 279 284 293
[20] 304 342 366 368 417 418 423 448 463 464 466 474 479 482 489 498 585 605 643
[39] 646 697 791 793 795 808 838 853 881 890 893 905 910 955 956 961 968 970 981
[58] 996
```

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

```
[1] 37 44 53 59 78 110 113 117 126 136 149 154 163 196 214 275 279 304 342
[20] 366 368 417 418 423 427 448 466 474 479 482 489 545 552 605 643 646 648 649
[39] 697 760 792 793 795 808 838 853 881 891 893 904 905 910 955 956 961 970
```

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

```

[1] 4 37 40 53 78 110 113 117 126 136 149 154 163 196 214 223 239 275 277
[20] 284 293 304 342 366 368 417 418 423 427 448 463 466 474 479 482 489 519 552
[39] 608 613 643 648 649 736 760 791 792 795 808 823 838 853 881 893 898 904 905
[58] 910 955 956 970 981 994 996

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

[1] 17

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

[1] 12

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

[1] 11

> which(con2_adj_p<=0.05/3)

[1] 37 53 78 154 196 275 417 423 448 482 808 893

> which(con3_adj_p<=0.05/3)

[1] 53 154 275 304 368 423 448 643 808 893 956

> 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] 58

```

```

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

[1] 52

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

[1] 43 63 72 101 142 160 161 180 182 203 268 282 308 315 318 364 382 401 408
[20] 418 435 443 463 466 510 512 516 533 543 555 616 620 623 637 694 696 709 714
[39] 727 742 761 766 801 805 831 836 846 854 855 862 866 873 880 905 966

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

[1] 43 49 59 63 72 93 97 101 160 161 180 182 201 203 268 280 282 308 318
[20] 401 408 443 459 463 510 516 533 543 546 573 581 591 616 620 623 637 694 709
[39] 742 761 766 801 804 805 829 831 836 846 854 855 862 865 866 880 905 956 966
[58] 990

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

[1] 43 63 72 160 161 180 182 201 203 268 280 282 308 318 361 382 401 426 435
[20] 443 463 466 512 516 533 543 546 555 573 620 623 637 694 709 727 742 761 801
[39] 804 805 829 831 846 854 855 862 865 866 880 956 966 994

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

[1] 6

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

[1] 7

> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adjp<=0.05/3)

[1] 2

```

## 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/RtmpgZgLgk/Rinst11f048036428/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] 62

> sum(diff_results$bootstrap_p<=0.05)

[1] 86

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

[1] 0

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

[1] 12

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

[1] 22

> diff_list_perm <- which(perm_adj_p<=0.05)
> diff_list_boot <- which(boot_adj_p<=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]
16	cg00014085	0.05906804	0.04518973	0.04211710	0.03665208
103	cg00094319	0.73784280	0.73532960	0.75574900	0.73830220
106	cg00095674	0.07076291	0.05045181	0.03861991	0.03337576
237	cg00215066	0.94926640	0.95311870	0.94634910	0.94561120
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
437	cg00424946	0.04122172	0.04325330	0.03339863	0.02876798
542	cg00520135	0.77510370	0.79688730	0.81833620	0.83043920
772	cg00743372	0.03922780	0.02919634	0.02187972	0.02568053
804	cg00777121	0.04540701	0.05430304	0.04154242	0.04221162
848	cg00826384	0.05721674	0.05612171	0.06644259	0.06358381
851	cg00830029	0.58362500	0.59397870	0.64739610	0.67269640
911	cg00888479	0.07388961	0.07361080	0.10149800	0.09985076
	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]	
16	0.04222944	0.05324246	0.03728026	0.04062589	
103	0.67349260	0.73510200	0.75715920	0.78981220	



106	0.04693030	0.06837343	0.04534005	0.03709488
237	0.94837410	0.94665570	0.94089070	0.94600090
280	0.61920530	0.61925200	0.46753250	0.55632410
437	0.03353116	0.03719167	0.03096761	0.03234779
542	0.83062760	0.55544810	0.83402240	0.89514710
772	0.02796053	0.03512214	0.02575992	0.02093909
804	0.04911277	0.04872797	0.04261405	0.04474881
848	0.05230160	0.06119713	0.06542751	0.06240686
851	0.50820240	0.34657470	0.66276570	0.64634510
911	0.08633986	0.06765189	0.09070268	0.12417730

diff\_results\$ordfit\_t[diff\_list\_perm]

16	2.325659
103	-2.268711
106	3.100324
237	1.419654
280	4.170347
437	2.102892
542	-1.775375
772	2.416991
804	1.995220
848	-2.314412
851	-2.841244
911	-3.621731

diff\_results\$permutation\_p[diff\_list\_perm]

16	0
103	0
106	0
237	0
280	0
437	0
542	0
772	0
804	0
848	0
851	0
911	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]
95	cg00081975	0.03633894	0.04975194	0.06024723	0.05598723
106	cg00095674	0.07076291	0.05045181	0.03861991	0.03337576
146	cg00134539	0.61101320	0.53321780	0.45999340	0.46787420
258	cg00234616	0.06639040	0.14705640	0.18254770	0.19942150
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565

285	cg00263760	0.09050395	0.10197760	0.14801710	0.12242400
346	cg00331237	0.05972383	NA	0.08204769	0.08345662
349	cg00332745	0.04703361	0.04634372	0.03676908	0.04518837
350	cg00333226	0.08320694	0.11587190	0.14999650	0.14307800
397	cg00394658	0.27940900	0.40410330	0.40262320	0.44339290
482	cg00468146	0.11144740	0.15416650	0.19827990	0.18517240
632	cg00615377	0.11265030	0.16140570	0.19404450	0.17468600
677	cg00651216	0.06825629	0.12529090	0.14409190	0.13907250
708	cg00679556	0.70602880	0.66481610	0.61718460	0.65394570
736	cg00706536	0.43475180	0.48246400	0.52171670	0.54656730
743	cg00717862	0.07999436	0.07873347	0.06089359	0.06171374
833	cg00814580	0.09348613	0.09619816	0.12010440	0.11534240
848	cg00826384	0.05721674	0.05612171	0.06644259	0.06358381
882	cg00858899	0.11427700	0.11919540	0.07690343	0.08321229
887	cg00862290	0.43640520	0.54047160	0.60786800	0.56325950
911	cg00888479	0.07388961	0.07361080	0.10149800	0.09985076
979	cg00945507	0.13432250	0.23854600	0.34749760	0.28903340

exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]

95	0.04561792	0.05115624	0.06068253	0.06168212
106	0.04693030	0.06837343	0.04534005	0.03709488
146	0.67191510	0.63137380	0.47929610	0.45428300
258	0.10620550	0.11668540	0.12630260	0.19163650
259	0.04030003	0.03996053	0.05086962	0.05445672
285	0.11693600	0.10650430	0.12281160	0.12310430
346	0.05372019	0.06241126	0.06955040	0.09140985
349	0.04975075	0.05253778	0.04444665	0.03717721
350	0.10704480	0.13751630	0.12588230	0.13863730
397	0.35626060	0.23388380	0.41974630	0.45806880
482	0.12285820	0.13271110	0.14196260	0.22159420
632	0.12573100	0.14483660	0.16338240	0.20130510
677	0.07669587	0.09597587	0.11690440	0.15194540
708	0.71234210	0.65323770	0.63793100	0.66387340
736	0.46315280	0.47922440	0.47347480	0.50734820
743	0.07594936	0.09062161	0.06475791	0.07271878
833	0.09577040	0.11598850	0.12860890	0.14111200
848	0.05230160	0.06119713	0.06542751	0.06240686
882	0.08961409	0.10730660	0.09203980	0.08726349
887	0.50259740	0.40111730	0.56646700	0.54552980
911	0.08633986	0.06765189	0.09070268	0.12417730
979	0.11848510	0.16653850	0.30718420	0.26624740

diff\_results\$ordfit\_t[diff\_list\_boot]

95	-3.252063
106	3.100324
146	5.394750
258	-3.046867

259	-4.052697
285	-3.093997
346	-3.767916
349	2.165826
350	-2.458696
397	-3.070559
482	-3.212481
632	-3.661161
677	-3.387628
708	2.465737
736	-2.733375
743	3.444684
833	-3.428319
848	-2.314412
882	3.179415
887	-3.217939
911	-3.621731
979	-4.750997

	diff_results\$bootstrap_p[diff_list_boot]
95	0
106	0
146	0
258	0
259	0
285	0
346	0
349	0
350	0
397	0
482	0
632	0
677	0
708	0
736	0
743	0
833	0
848	0
882	0
887	0
911	0
979	0