

Introduction to RBM package

Dongmei Li

April 10, 2025

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] 26

> which(myresult$permutation_p<=0.05)

[1] 40 97 132 144 214 216 247 256 273 346 455 518 519 544 576 611 645 658 717
[20] 745 746 767 838 845 882 985

> sum(myresult$bootstrap_p<=0.05)

[1] 7

> which(myresult$bootstrap_p<=0.05)

[1] 158 285 295 296 452 658 673

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

[1] 7

> 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] 33

> which(myresult2$bootstrap_p<=0.05)

[1] 50 118 212 226 242 253 259 282 284 303 309 380 412 415 424 434 474 503 504
[20] 513 554 595 601 719 749 793 805 822 905 958 968 976 983

> 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] 64

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

[1] 68

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

[1] 45

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

[1] 8 10 13 23 33 54 63 80 104 112 146 149 150 153 157 178 199 205 239
[20] 241 243 250 255 263 275 297 357 360 392 394 417 435 475 486 532 574 585 593
[39] 595 606 677 705 744 746 760 786 791 801 802 817 844 855 867 870 871 874 887
[58] 896 904 906 907 925 926 950

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

[1] 8 10 11 12 13 23 33 47 54 63 104 112 127 146 147 149 150 157 178
[20] 187 199 205 239 241 250 255 276 315 334 355 357 360 361 392 394 430 472 475
[39] 486 498 545 552 578 585 593 606 662 677 725 738 746 754 760 802 814 817 836
[58] 854 855 867 871 887 906 925 926 942 950 952

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

[1] 8 10 13 23 33 54 63 104 106 112 146 149 150 157 178 187 199 205 241
[20] 255 357 360 392 418 472 475 486 552 574 578 585 593 606 746 760 786 791 817
[39] 844 867 870 871 925 926 942

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

```

```

[1] 13

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

[1] 2

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

[1] 0

> which(con2_adjp<=0.05/3)

[1] 10 925

> which(con3_adjp<=0.05/3)

integer(0)

> 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] 46

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

[1] 37

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

```

```

[1] 76 80 81 113 114 120 131 142 148 152 241 262 285 299 314 333 344 353 354
[20] 372 410 418 446 447 480 513 515 537 542 584 602 673 702 714 726 748 750 759
[39] 761 772 787 848 861 874 879 884 886 903 906 948 950 964 979 997

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

[1] 80 81 113 120 131 142 152 243 262 274 275 285 299 333 354 372 418 436 446
[20] 447 468 480 542 584 602 630 673 702 725 748 759 761 787 833 848 861 879 884
[39] 886 903 906 948 950 979 989 997

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

[1] 4 76 80 113 131 142 152 241 262 299 314 354 372 418 446 447 468 542 584
[20] 602 673 702 748 759 761 772 787 848 861 884 886 906 948 950 964 980 997

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

[1] 9

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

[1] 3

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=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 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/RtmpaSs40G/Rinst229142535ac05f/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.:0.59031   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] 47

> sum(diff_results$permutation_p<=0.05)

[1] 76

```

```

> sum(diff_results$bootstrap_p<=0.05)

[1] 59

> 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] 11

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

[1] 8

> 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[diff_list_boot, ])
> print(sig_results_perm)

```

	IlmnID	Beta	exmdata2[, 2]	exmdata3[, 2]	exmdata4[, 2]	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]
19	cg00016968	0.80628480	NA	0.81440820	0.83623180	0.80831380	0.73306440	0.82968340	0.84917800
83	cg00072216	0.04505377	0.04598964	0.04000674	0.03231534	0.04965089	0.04833366	0.03466159	0.04390894
237	cg00215066	0.94926640	0.95311870	0.94634910	0.94561120	0.94837410	0.94665570	0.94089070	0.94600090
259	cg00234961	0.04192170	0.04321576	0.05707140	0.05327565	0.04030003	0.03996053	0.05086962	0.05445672
283	cg00262415	0.03850601	0.04621248	0.03579758	0.03765227	0.03746915	0.04200230	0.03014699	0.02903290
285	cg00263760	0.09050395	0.10197760	0.14801710	0.12242400	0.11693600	0.10650430	0.12281160	0.12310430
520	cg00502442	0.03163993	0.03581662	0.02785063	0.02549502	0.03111720	0.03189393	0.02415307	0.02941176
627	cg00612467	0.04777553	0.03783457	0.05380982	0.05582291	0.04740551	0.05332965	0.05775211	0.05579710
804	cg00777121	0.04540701	0.05430304	0.04154242	0.04221162	0.04911277	0.04872797	0.04261405	0.04474881
928	cg00901493	0.03737166	0.03903724	0.04684618	0.04981432	0.04490690	0.04204062	0.05050039	0.05268215
979	cg00945507	0.13432250	0.23854600	0.34749760	0.28903340				


```
979 0.11848510 0.16653850 0.30718420 0.26624740
```

```
diff_results$ordfit_t[diff_list_perm]
```

```
19 -2.547097
83 1.947226
237 1.021426
259 -2.833203
283 1.601804
285 -2.993292
520 1.319602
627 -1.797392
804 1.445572
928 -1.982308
979 -4.968792
```

```
diff_results$permutation_p[diff_list_perm]
```

```
19 0
83 0
237 0
259 0
283 0
285 0
520 0
627 0
804 0
928 0
979 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]
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
743	cg00717862	0.07999436	0.07873347	0.06089359	0.06171374
833	cg00814580	0.09348613	0.09619816	0.12010440	0.11534240
882	cg00858899	0.11427700	0.11919540	0.07690343	0.08321229
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]	
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	
743	0.07594936	0.09062161	0.06475791	0.07271878	
833	0.09577040	0.11598850	0.12860890	0.14111200	
882	0.08961409	0.10730660	0.09203980	0.08726349	
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]
```

146	5.636263
259	-2.833203
280	4.337628
743	2.918806
833	-3.278186
882	3.009085
911	-3.490240
979	-4.968792

```
diff_results$bootstrap_p[diff_list_boot]
```

146	0
259	0
280	0
743	0
833	0
882	0
911	0
979	0