

Introduction to RBM package

Dongmei Li

November 16, 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] 15

> which(myresult$permutation_p<=0.05)

[1] 26 40 102 161 350 377 488 495 501 517 541 547 927 940 972

> sum(myresult$bootstrap_p<=0.05)

[1] 1

> which(myresult$bootstrap_p<=0.05)

[1] 282

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

[1] 1

> 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] 17

> which(myresult2$bootstrap_p<=0.05)

[1] 88 114 153 182 272 296 331 377 400 488 584 593 623 654 658 744 933

> bootstrap2_adj_p <- p.adjust(myresult2$bootstrap_p, "BH")
> sum(bootstrap2_adj_p<=0.05)

[1] 2

```

- 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] 64

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

[1] 62

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

[1]  22  37  62  82  92 106 119 120 121 140 162 170 174 176 181 191 199 204 205
[20] 220 230 243 248 264 270 281 297 305 353 355 372 382 386 406 415 424 445 478
[39] 479 544 561 586 592 610 629 699 715 726 743 748 771 780 836 866 868 870 890
[58] 892 908 914 924 930 960 976

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

[1]   5  37  62  82  92 106 119 121 134 140 162 170 184 191 204 205 220 230 243
[20] 248 256 274 305 353 355 372 382 383 406 415 424 479 519 544 574 586 592 629
[39] 679 699 702 715 726 743 748 756 762 771 780 866 870 890 908 914 922 924 930
[58] 949 960 964 967 976 981 985

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

[1]  37  62  82  92 106 112 119 120 140 162 170 174 181 184 191 199 205 220 230
[20] 248 281 297 305 355 372 382 383 386 406 415 424 445 478 544 558 561 584 586
[39] 592 610 629 679 699 702 703 715 743 748 771 780 836 866 868 870 890 892 908
[58] 914 921 949 960 985

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

```

```

[1] 6

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

[1] 5

> 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] 106 140 629 748 780

> which(con3_adjp<=0.05/3)

[1] 424

> 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] 45

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

[1] 60

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

[1] 57

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

```

```
[1] 20 34 45 47 63 76 147 179 197 203 231 260 269 275 303 354 374 404 429
[20] 478 516 544 591 603 605 613 618 636 663 670 716 726 749 781 791 801 843 856
[39] 858 859 864 901 908 928 960
```

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

```
[1] 20 34 45 47 63 76 139 147 163 168 187 197 203 231 260
[16] 265 269 273 275 280 303 306 353 357 374 404 428 429 478 486
[31] 544 567 591 603 605 613 618 636 643 663 670 716 726 749 781
[46] 789 791 801 831 833 843 856 859 864 901 908 928 960 979 1000
```

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

```
[1] 20 34 41 45 47 63 76 119 137 147 163 187 197 203 223
[16] 231 260 269 275 280 303 306 353 354 374 404 429 478 544 567
[31] 591 603 605 613 618 636 663 670 716 749 754 781 791 801 831
[46] 839 843 856 858 859 864 896 901 902 908 928 1000
```

```
> 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] 3
```

```
> con23_adjp <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
```

```
> sum(con23_adjp<=0.05/3)
```

```
[1] 5
```

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] "/private/var/folders/db/4tvvx8jx4z3fm1gzlnlzw9rc0000gq/T/Rtmp6AgzdA/Rinstac157b70b5e8/RBM/d

> 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] 64

> sum(diff_results$bootstrap_p<=0.05)

[1] 43

> 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] 2

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

[1] 7

> 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]
237 cg00215066 0.94926640    0.95311870    0.94634910    0.94561120
245 cg00224508 0.04479948    0.04972043    0.04152814    0.04189373
      exmdata5[, 2] exmdata6[, 2] exmdata7[, 2] exmdata8[, 2]
237    0.94837410    0.94665570    0.94089070    0.94600090
245    0.04208405    0.05284988    0.03775905    0.03955271
      diff_results$ordfit_t[diff_list_perm]
237                                1.021426
245                                1.494678
      diff_results$permutation_p[diff_list_perm]
237                                0
245                                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]
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
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
285		0.11693600	0.10650430	0.12281160	0.12310430
743		0.07594936	0.09062161	0.06475791	0.07271878
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		
285			-2.993292		
743			2.918806		
911			-3.490240		
979			-4.968792		
	diff_results\$bootstrap_p[diff_list_boot]				
146			0		
259			0		
280			0		
285			0		
743			0		
911			0		
979			0		