

Introduction to RBM package

Dongmei Li

November 2, 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] 28

> which(myresult$permutation_p<=0.05)

[1] 37 60 74 162 184 226 286 319 321 340 362 373 378 391 403 411 509 526 591
[20] 606 613 655 704 860 862 974 976 998

> sum(myresult$bootstrap_p<=0.05)

[1] 15

> which(myresult$bootstrap_p<=0.05)

[1] 60 74 99 162 222 377 411 610 626 658 674 710 805 831 982

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

[1] 0

> 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] 29

> which(myresult2$bootstrap_p<=0.05)

[1] 46 49 73 76 145 181 253 307 310 330 342 354 406 414 468 477 506 518 610
[20] 660 661 718 840 852 874 909 924 944 958

> 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] 61

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

[1] 59

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

[1] 57

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

[1] 14 26 62 78 115 163 187 216 219 237 238 252 268 286 293 298 307 313 318
[20] 323 334 337 385 399 401 447 449 453 454 470 486 488 500 537 565 572 579 587
[39] 594 603 616 640 672 686 705 719 722 744 745 748 778 780 857 879 894 917 918
[58] 919 929 958 977

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

[1] 14 26 62 78 111 115 128 163 187 216 219 237 238 252 268 270 293 307 311
[20] 313 318 323 334 337 385 399 447 448 449 452 453 454 469 470 473 481 488 500
[39] 579 594 603 616 640 672 674 705 719 745 748 778 867 879 894 906 919 923 929
[58] 930 958

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

[1] 14 55 62 78 115 128 163 187 216 219 237 238 252 268 293 298 307 313 317
[20] 318 334 337 385 399 448 449 453 470 486 488 500 537 539 571 579 587 594 616
[39] 640 672 674 719 722 744 745 748 778 780 873 879 894 917 918 919 929 958 977

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

```

```

[1] 12

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

[1] 6

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

[1] 10

> which(con2_adjp<=0.05/3)

[1] 216 313 318 453 488 719

> which(con3_adjp<=0.05/3)

[1] 62 216 313 318 453 640 719 748 879 958

> 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] 67

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

[1] 81

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

[1] 63

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

```

```

[1] 55 68 72 90 111 115 124 129 131 169 177 189 203 206 211 230 299 314 369
[20] 385 387 414 416 426 433 437 444 447 450 453 459 461 470 480 494 499 516 533
[39] 549 555 569 590 601 605 626 628 661 688 704 731 742 757 768 779 780 788 826
[58] 827 856 857 858 880 884 917 925 958 982

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

[1] 55 72 74 90 111 122 124 129 131 133 144 153 169 177 189 203 218 221 230
[20] 307 369 385 387 390 398 403 410 414 416 426 433 444 445 447 453 459 461 470
[39] 480 492 494 499 516 533 542 549 555 562 569 590 601 605 607 621 626 635 661
[58] 693 704 731 742 743 757 768 779 780 788 826 827 839 855 856 857 858 867 880
[77] 884 885 917 958 982

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

[1] 55 70 72 90 111 124 131 133 169 189 203 206 211 221 230 239 287 307 369
[20] 387 414 426 433 437 447 450 453 459 461 470 474 480 499 516 533 555 569 590
[39] 601 605 626 628 635 661 688 693 704 731 742 757 768 780 788 821 826 856 857
[58] 858 880 884 917 958 982

> 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] 10

> con23_adj_p <- p.adjust(myresult2_F$bootstrap_p[, 3], "BH")
> sum(con23_adj_p<=0.05/3)

[1] 9

```

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/RtmpZC4Ek2/Rinst2af63813527/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] 91

> sum(diff_results$bootstrap_p<=0.05)

[1] 21

> 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] 18

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

[1] 0

> 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]
5	cg00006414	0.07635468	0.07442468	0.15698040	0.08676092
16	cg00014085	0.05906804	0.04518973	0.04211710	0.03665208
83	cg00072216	0.04505377	0.04598964	0.04000674	0.03231534
103	cg00094319	0.73784280	0.73532960	0.75574900	0.73830220
106	cg00095674	0.07076291	0.05045181	0.03861991	0.03337576
131	cg00121904	0.15449580	0.17949750	0.23608110	0.24354150
189	cg00176210	0.28756520	0.39161870	0.44272520	0.44725330
219	cg00202702	0.04104248	0.04685628	0.03793627	0.03529329
237	cg00215066	0.94926640	0.95311870	0.94634910	0.94561120
245	cg00224508	0.04479948	0.04972043	0.04152814	0.04189373
251	cg00230368	0.05546448	0.04403809	0.04143668	0.03345086
280	cg00260778	0.64319890	0.60488960	0.56735060	0.53150910
283	cg00262415	0.03850601	0.04621248	0.03579758	0.03765227
349	cg00332745	0.04703361	0.04634372	0.03676908	0.04518837
437	cg00424946	0.04122172	0.04325330	0.03339863	0.02876798
848	cg00826384	0.05721674	0.05612171	0.06644259	0.06358381
851	cg00830029	0.58362500	0.59397870	0.64739610	0.67269640
979	cg00945507	0.13432250	0.23854600	0.34749760	0.28903340

	exmdata5[, 2]	exmdata6[, 2]	exmdata7[, 2]	exmdata8[, 2]
5	0.07982556	0.08111396	0.08271889	0.08045977
16	0.04222944	0.05324246	0.03728026	0.04062589
83	0.04965089	0.04833366	0.03466159	0.04390894
103	0.67349260	0.73510200	0.75715920	0.78981220
106	0.04693030	0.06837343	0.04534005	0.03709488
131	0.17352980	0.12564280	0.18193170	0.20847670
189	0.34106080	0.33765930	0.41252110	0.37024890
219	0.04074652	0.05125000	0.03908795	0.04075583
237	0.94837410	0.94665570	0.94089070	0.94600090
245	0.04208405	0.05284988	0.03775905	0.03955271
251	0.04921680	0.06053175	0.04160748	0.04809040
280	0.61920530	0.61925200	0.46753250	0.55632410
283	0.03746915	0.04200230	0.03014699	0.02903290
349	0.04975075	0.05253778	0.04444665	0.03717721
437	0.03353116	0.03719167	0.03096761	0.03234779
848	0.05230160	0.06119713	0.06542751	0.06240686
851	0.50820240	0.34657470	0.66276570	0.64634510
979	0.11848510	0.16653850	0.30718420	0.26624740

	diff_results\$ordfit_t[diff_list_perm]
5	-1.434079
16	1.954876
83	1.947226
103	-2.343784
106	2.887876
131	-3.562745
189	-3.232921
219	1.375603
237	1.021426
245	1.494678
251	1.902178
280	4.337628
283	1.601804
349	1.659117
437	1.574598
848	-1.687144
851	-2.986319
979	-4.968792

	diff_results\$permutation_p[diff_list_perm]
5	0
16	0
83	0
103	0
106	0
131	0

189	0
219	0
237	0
245	0
251	0
280	0
283	0
349	0
437	0
848	0
851	0
979	0

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

```
[1] IlmnID
[2] Beta
[3] exmdata2[, 2]
[4] exmdata3[, 2]
[5] exmdata4[, 2]
[6] exmdata5[, 2]
[7] exmdata6[, 2]
[8] exmdata7[, 2]
[9] exmdata8[, 2]
[10] diff_results$ordfit_t[diff_list_boot]
[11] diff_results$bootstrap_p[diff_list_boot]
<0 rows> (or 0-length row.names)
```