

Riborex Manual

Wenzheng Li, Weili Wang, Philip J. Uren, Luiz OF Penalva, Andrew D. Smith

Introduction

Riborex is a R package for identifying differentially translated genes from Ribo-seq data. Riborex integrates both RNA- and Ribo-seq read count data into a single generalized linear model (GLM) and generates a modified design matrix reflecting the integration. At its core, Riborex applies existing RNA-seq analysis tools such as edgeR, DESeq2 and Voom to this modified design matrix and identifies differential translation across conditions.

Detailed example

First, we need to load Riborex library.

```
library(riborex)
```

The input for Riborex are two read count tables summarized from RNA-seq and Ribo-seq data respectively. The read count table should be organized as a data frame with rows correspond to genes and columns correspond to samples as shown below.

```
data(riborexdata)
RNACntTable <- riborexdata$rna
RiboCntTable <- riborexdata$ribo
```

We can check the first five lines of the table:

```
head(RNACntTable,5)
```

##	BN_336	BN_337	BN_338	BN_339
## ENSRNOG000000000017	7	11	4	4
## ENSRNOG000000000024	2467	2478	3258	2316
## ENSRNOG000000000033	206	282	330	244
## ENSRNOG000000000034	758	672	1335	767
## ENSRNOG000000000036	237	163	211	189

```
head(RiboCntTable,5)
```

##	BN_341	BN_342	BN_343	BN_344
## ENSRNOG000000000017	15	5	10	2
## ENSRNOG000000000024	5206	5921	2864	1985
## ENSRNOG000000000033	30	30	23	13
## ENSRNOG000000000034	943	775	842	311
## ENSRNOG000000000036	80	49	30	7

Then we need to prepare two vectors to indicate the treatments of samples in RNA- and Ribo-seq data. Both RNA-seq and Ribo-seq can have different number of samples in control and treated conditions, and RNA-seq and Ribo-seq data can have different number of samples.

```

rnaCond <- c("control", "control", "treated", "treated")
riboCond <- c("control", "control", "treated", "treated")

```

After the two read count table and two condition vectors are ready, we can use `riborex()`, and we can choose which engine to use. By default, DESeq2 is used as the engine if you don't specify the engine option. Use `help(riborex)` in R to see more details about this function.

```

res.deseq2 <- riborex(RNACntTable, RiboCntTable, rnaCond, riboCond)

```

The format of the result is the same when DESeq2 is used in RNA-seq analysis.

```

res.deseq2

```

```

## log2 fold change (MAP): EXTRA2 treated vs control
## Wald test p-value: EXTRA2 treated vs control
## DataFrame with 13916 rows and 6 columns
##
##           baseMean log2FoldChange      lfcSE      stat
##           <numeric>      <numeric> <numeric> <numeric>
## ENSRNOG000000000017      7.694934      0.60242172 0.6633528 0.9081467
## ENSRNOG000000000024 3544.238071     -0.09262781 0.1967739 -0.4707321
## ENSRNOG000000000033 111.439451      0.12090513 0.4390872 0.2753556
## ENSRNOG000000000034 783.974916      0.07400898 0.3243979 0.2281426
## ENSRNOG000000000036 99.151141      -0.65373460 0.4894467 -1.3356604
## ...
## ENSRNOG000000061895 105.24110      0.42802240 0.4224309 1.0132366
## ENSRNOG000000061899 40.04057      -0.55910038 0.5828614 -0.9592339
## ENSRNOG000000061910 4237.53481      0.29957699 0.2366771 1.2657623
## ENSRNOG000000061928 1651.61680     -0.07583546 0.2370871 -0.3198633
## ENSRNOG000000061989 107.36281     -0.06620802 0.4009571 -0.1651250
##
##           pvalue      padj
##           <numeric> <numeric>
## ENSRNOG000000000017 0.3638007 0.7494602
## ENSRNOG000000000024 0.6378320 0.9022741
## ENSRNOG000000000033 0.7830430 0.9509349
## ENSRNOG000000000034 0.8195354 0.9607316
## ENSRNOG000000000036 0.1816603 0.5432690
## ...
## ENSRNOG000000061895 0.3109472 0.7034611
## ENSRNOG000000061899 0.3374409 0.7261818
## ENSRNOG000000061910 0.2055982 0.5775201
## ENSRNOG000000061928 0.7490719 0.9411896
## ENSRNOG000000061989 0.8688456 0.9744379

```

Also, you can use `summary()` for your results.

```

summary(res.deseq2)

```

```

##
## out of 13916 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 1139, 8.2%

```

```
## LFC < 0 (down)      : 1274, 9.2%
## outliers [1]        : 0, 0%
## low counts [2]      : 270, 1.9%
## (mean count < 3)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

And results can be saved by:

```
write.table(res.deseq2, "riborex_res_deseq2.txt", quote=FALSE)
```

If you want to use edgeR as your engine, you can use riborex () as:

```
res.edgeR <- riborex(RNAcntTable, RiboCntTable, rnaCond, riboCond, "edgeR")
```

The format of the result is the same when edgeR is used in RNA-seq analysis.

```
head(res.edgeR$table)
```

```
##              logFC    logCPM      LR    PValue      FDR
## ENSRNOG000000000017  1.3629124 -2.456968 1.80670421 0.1789041 0.4626704
## ENSRNOG000000000024 -0.3012715  6.212404 2.13615823 0.1438618 0.4037879
## ENSRNOG000000000033  0.0717875  1.313235 0.02631400 0.8711358 0.9636269
## ENSRNOG000000000034 -0.1343023  4.029136 0.12544284 0.7232047 0.9111520
## ENSRNOG000000000036 -0.8253990  1.132478 2.15594580 0.1420190 0.4007169
## ENSRNOG000000000040 -0.1905669 -1.555003 0.07536038 0.7836862 0.9338705
```

For edgeR engine, you can also choose to estimate dispersion of RNA-seq and Ribo-seq data separately by specifying engine as “edgeRD”.

```
res.edgeRD <- riborex(RNAcntTable, RiboCntTable, rnaCond, riboCond, "edgeRD")
```

If you want to use Voom as the engine, you can run riborex () as:

```
res.voom <- riborex(RNAcntTable, RiboCntTable, rnaCond, riboCond, "Voom")
```

The format of the result is the same when Voom is used in RNA-seq analysis.

```
head(res.voom)
```

```
##              logFC    AveExpr      t    P.Value adj.P.Val
## ENSRNOG000000000017  1.39674189 -2.8437969  1.2847048 0.2268559 0.5243467
## ENSRNOG000000000024 -0.30047073  6.0075571 -1.8737990 0.0893910 0.2991861
## ENSRNOG000000000033  0.07661457  0.7070877  0.1687028 0.8692764 0.9540355
## ENSRNOG000000000034 -0.13777833  3.9701893 -0.3933633 0.7020190 0.8871606
## ENSRNOG000000000036 -0.89165405  0.7106061 -1.3890222 0.1939374 0.4826701
## ENSRNOG000000000040 -0.20967655 -1.7042630 -0.2991046 0.7707701 0.9157225
##              B
## ENSRNOG000000000017 -4.928058
## ENSRNOG000000000024 -5.682138
## ENSRNOG000000000033 -6.320171
## ENSRNOG000000000034 -7.130133
## ENSRNOG000000000036 -5.425369
## ENSRNOG000000000040 -5.854482
```

Multi-factor experiment

Since we don't find any available ribosome profiling data generated in a multi-factor experiment, here we generate a pseudo dataset to demonstrate the usage of riborex in a multi-factor experiment. The pseudo dataset have 8 samples in RNA-seq and Ribo-seq, and two factors are included.

```
rna <- RNACntTable[,c(1,2,1,2,3,4,3,4)]
ribo <- RiboCntTable[,c(1,2,1,2,3,4,3,4)]
```

For multi-factor experiment, we prepare two data frames to indicate the treatment under each factor. Here for the 8 samples in both RNA- and Ribo-seq experiment, the 3rd and 4th samples are treated with drug1 and the 7th and 8th samples are treated with drug2.

```
rnaCond <- data.frame(factor1=c("control1", "control1", "treated1", "treated1",
                                "control1", "control1", "control1", "control1")),
                    factor2=c("control2", "control2", "control2", "control2",
                                "control2", "control2", "treated2", "treated2"))

riboCond <- data.frame(factor1=c("control1", "control1", "treated1", "treated1",
                                "control1", "control1", "control1", "control1")),
                    factor2=c("control2", "control2", "control2", "control2",
                                "control2", "control2", "treated2", "treated2"))
```

Also we need to prepare a contrast to specify the comparison we want to perform, for example, if we want to compare the influence of the usage of drug2. The contrast can be constructed as:

```
contrast = c("factor2", "control2", "treated2")
```

Then riborex () is used with contrast specified.

```
res.deseq2 <- riborex(rna, ribo, rnaCond, riboCond, "DESeq2", contrast = contrast)
```

We can see the summary of the result:

```
summary(res.deseq2)

##
## out of 13916 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 0, 0%
## LFC < 0 (down)    : 0, 0%
## outliers [1]      : 5, 0.036%
## low counts [2]    : 0, 0%
## (mean count < 1)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

Setup

This analysis was conducted on

sessionInfo()

```
## R version 3.3.1 (2016-06-21)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 14.04.4 LTS
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8      LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8      LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8  LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8     LC_NAME=C
##  [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel stats4      stats      graphics  grDevices  utils      datasets
## [8] methods    base
##
## other attached packages:
##  [1] riborex_1.0.0           edgeR_3.14.0
##  [3] limma_3.28.11          DESeq2_1.12.3
##  [5] SummarizedExperiment_1.2.3 Biobase_2.32.0
##  [7] GenomicRanges_1.24.2    GenomeInfoDb_1.8.1
##  [9] IRanges_2.6.1           S4Vectors_0.10.1
## [11] BiocGenerics_0.18.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.5             formatR_1.4             RColorBrewer_1.1-2
##  [4] plyr_1.8.4              XVector_0.12.0          tools_3.3.1
##  [7] zlibbioc_1.18.0         rpart_4.1-10            digest_0.6.9
## [10] RSQLite_1.0.0           annotate_1.50.0          evaluate_0.9
## [13] gtable_0.2.0            lattice_0.20-33         Matrix_1.2-6
## [16] DBI_0.4-1               yaml_2.1.13             gridExtra_2.2.1
## [19] genefilter_1.54.2       stringr_1.0.0           knitr_1.13
## [22] cluster_2.0.4           locfit_1.5-9.1          grid_3.3.1
## [25] nnet_7.3-12             data.table_1.9.6        AnnotationDbi_1.34.3
## [28] XML_3.98-1.4            survival_2.39-4         BiocParallel_1.6.2
## [31] foreign_0.8-66          rmarkdown_1.0           latticeExtra_0.6-28
## [34] Formula_1.2-1           geneplotter_1.50.0      ggplot2_2.1.0
## [37] magrittr_1.5            Hmisc_3.17-4            scales_0.4.0
## [40] htmltools_0.3.5         splines_3.3.1           xtable_1.8-2
## [43] colorspace_1.2-6        stringi_1.1.1           acepack_1.3-3.3
## [46] munsell_0.4.3           chron_2.3-47
```