Genome-wide assessment of differential translations with ribosome profiling data – the xtail package

Zhengtao Xiao[1−3], Qin Zou[1,3], Yu Liu[1−3], and Xuerui Yang[1−3]

[1]MOE Key Laboratory of Bioinformatics,
[2]Tsinghua-Peking Joint Center for Life Sciences,
[3]School of Life Sciences, Tsinghua University, Beijing 100084, China.

October 19, 2015

1 Introduction

This package is for identifing genes undergoing differential translation with ribosome profiling data. Xtail stats with a simple assumption that if a gene undergoing translational dyresgulation under certain exprimental or physiological condition, the change of its RPF abundance should be discoordinated with that of mRNA expression. Therefore, Xtail first estimates fold changes of RPF and mRNA across two conditions, separately, to quantify the magnitude of differential translation. In a parallel process, Xtail also estimates ratios of RPF over mRNA, in two conditions, to assess the differential translation. This strategy avoids fase disvoeries of differential translation due to extremely larger expression change in mRNA or RPF between two conditions.

`Xtail` consists of three major steps: (1) modeling of ribosome profiling data using negative binomial distribution (NB), (2) estabilishment of probability distributions for fold changes of mRNA or RPF (or RPF-to mRNA ratios), and (3) evaluation of statistical significance and magnitude of differential translations. The differential translation of each gene is evaluated by two parallel: one is difference of log2 fold change of mRNA and RPF across two condition, another one is difference of log2 ratio of RPF over mRNA between two conditions. The more conserved one is selected as the final assement of differential translation.

As default, the package adapts the strategy of DESeq2 [1] to normalize read counts of mRNA and RPF in all samples, and fits NB distributions with dispersions α and μ.

This guide provides step-by-step instructions on how to load data, how to excute the package and how to interpret output.

2 Data Preparation

As input, the `Xtail` package uses read counts of RPF and mRNA, in the form of rectangular table of values. The rows and columns of the table correspond the genes and samples. Each cell in the *g-th* row and the *i-th* columns is the count number of reads mapped to gene *g* in sample *i*. The mRNA count data and RPF count data are stored in two text files.We can read them in using **R**'s standard function. For example,

```
mrna <- read.table("mRNA_counts.txt", header=TRUE, row.names=1)
rpf <- read.table("RPF_counts.txt", header=TRUE, row.names=1)
```

Here, header=TRUE indicates that the first line contains columns names and row.names=1 means that the first column should be used as row names (gene names or gene ID).

Xtail takes in raw read counts of RPF and mRNA, and performs median-of-ratios normalization by default. This normalization method is also recommend by Reddy R. [2]. Alternatively, users can provide normalized read counts and skip the built-in normalization in Xtail.

In this vignette, we select a published ribosome profiling dataset from human prostate cancer cell PC3 after mTOR signaling inhibition with PP242 [3]. This dataset consist of 11391 genes from two conditions ("treatment" vs. "control") with each condition having two replicates.

3 An Example

Here we perform an analysis on the ribosome profiling data described above. First we load the library and data.

```
library(xtail)
data(xtaildata)
```

Next we can view the first five lines of the mRNA (`mrna`) and RPF (`rpf`) elements of `xtaildata`.

```
mrna <- xtaildata$mrna
rpf <- xtaildata$rpf
head(mrna,5)

##                 control1 control2 treat1 treat2
## ENSG00000000003      825      955    866   1039
## ENSG00000000419     1054      967    992    888
## ENSG00000000457       71       75    139     95
## ENSG00000000460      191      162    199    201
## ENSG00000000971       81        2     88     11

head(rpf,5)

##                 control1 control2 treat1 treat2
## ENSG00000000003      143      302    197    195
## ENSG00000000419      234      481    383    306
## ENSG00000000457       12       17     17     15
## ENSG00000000460       45       88     63     37
## ENSG00000000971       31        7     36      2
```

We assign condition labels corresponding to the columns of the mRNA and RPF inputs.

```
condition <- c("control","control","treat","treat")
```

Next, we run the main function, `xtail()`. By default, the second condition (here is "treat") would be compared against the first condition (here is "control"). Those genes with the minimum average expression of mRNA counts and RPF counts acrossing all conditions larger than 1 are used (can be changed by set `minMeanCount`). All the available CPU cores are used for running program. The argument `"bins"` is the number of bins used for calculating the probability densities of log2FC and log2R. This paramater will determine how accurate the final pvalue. Here, in order to keep the run-time of this vignette small, we will set `bins` to "1000". Detailed description of the arguments of the `xtail` function can be read by typing `?xtail` or `help(xtail)` at the **R** prompt.

```
test.results <- xtail(mrna,rpf,condition,bins=1000)

## Calculating the library size factors
## 1.  Estimate the log2 fold change in mrna
## 2.  Estimate the log2 fold change in rpf
## 3.  Estimate the difference between two log2 fold changes
## 4.  Estimate the log2 ratio in first condition
## 5.  Estimate the log2 ratio in second condition
## 6.  Estimate the difference between two log2 ratios
```

Now we can examine the first five lines of the results produced by the 'xtail' run.

```
head(test.results,5)

##                  log2FC_TE_v1    OVL_v1 pvalue_v1 log2FC_TE_v2    OVL_v2 pvalue_v2
## ENSG00000000003    0.05867991 0.8742323 0.8162414    0.0623457 0.8693434 0.8079550
## ENSG00000000419    0.35599462 0.3141530 0.1536697    0.3592684 0.2949123 0.1360275
## ENSG00000000457   -0.29931892 0.7070745 0.5865027   -0.2897093 0.7480170 0.6409663
## ENSG00000000460   -0.31079662 0.5771754 0.4203029   -0.3109160 0.5780560 0.4249511
## ENSG00000000971   -0.62232332 0.8150186 0.7105031   -0.6457720 0.8284256 0.7443768
##                  OVL_final pvalue_final pvalue.adjust log2FC_TE_final      CI(95%)
## ENSG00000000003 0.8742323    0.8162414     0.9957191      0.05867991 [-0.47,0.59]
## ENSG00000000419 0.3141530    0.1536697     0.9957191      0.35599462 [-0.14,0.84]
## ENSG00000000457 0.7480170    0.6409663     0.9957191     -0.28970931 [-1.54,0.96]
## ENSG00000000460 0.5780560    0.4249511     0.9957191     -0.31091597 [-1.08,0.47]
## ENSG00000000971 0.8284256    0.7443768     0.9957191     -0.64577202     [-5.3,4]
```

The results of fist pipline are named with suffix "_v1", which are generated by comparing with mRNA and RPF log2 fold change: The element `log2FC_TE_v1` represents the log2 fold change of TE; `OVL_v1` is overlap coefficience, which quantify the statistical confidence of difference between two distributions, here is difference of log2 fold change of mRNA and RPF. The `pvalue_v1` represent statistical significance. The sencond pipline are named with suffix "_v2", which are derived from comparing log2 ratios between two conditions: `log2FC_TE_v2`, `OVL_v2`, and `pvalue_v2` are log2 ratio of TE, overlap coefficience, and pvalues. Finally, the more conserved results (with larger-Pvalue) was select as the final assessment of differential translation, which are named with suffix "_final". The `pvalue.adjust` is the estimated false discovery rate corresponding to the `pvalue_final`. The `CI` is the credible interval of `log2FC_TE_final` (95% by default), and which could be changed by setting `ci` in `xtail` function.

Finally, the plain-text file of the results can be exported using the base **R** functions *write.csv* or *write.table*.

```
write.table(test.results,"test_results.txt",quote=F,sep="\t")
```

Session Info

```
sessionInfo()

## R version 3.2.2 Patched (2015-08-23 r69167)
## Platform: x86_64-suse-linux-gnu (64-bit)
## Running under: openSUSE 13.2 (Harlequin) (x86_64)
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [4] LC_COLLATE=en_US.UTF-8     LC_MONETARY=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
## [10] LC_TELEPHONE=C             LC_MEASUREMENT=en_US.UTF-8
##                                 LC_TIME=en_US.UTF-8
##                                 LC_MESSAGES=en_US.UTF-8
##                                 LC_ADDRESS=C
##                                 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] parallel  stats4    stats     graphics  grDevices utils     datasets  methods
## [9] base
##
```

```
## other attached packages:
##  [1] xtail_1.1.2                DESeq2_1.10.0              RcppArmadillo_0.6.100.0.0
##  [4] Rcpp_0.12.1                SummarizedExperiment_1.0.0 Biobase_2.30.0
##  [7] GenomicRanges_1.22.0       GenomeInfoDb_1.6.0         IRanges_2.4.0
## [10] S4Vectors_0.8.0            BiocGenerics_0.16.0        knitr_1.11
##
## loaded via a namespace (and not attached):
##  [1] RColorBrewer_1.1-2   formatR_1.2.1        futile.logger_1.4.1  highr_0.5.1
##  [5] plyr_1.8.3           XVector_0.10.0       futile.options_1.0.0 tools_3.2.2
##  [9] zlibbioc_1.16.0      rpart_4.1-10         digest_0.6.8         RSQLite_1.0.0
## [13] annotate_1.48.0      evaluate_0.8         gtable_0.1.2         lattice_0.20-33
## [17] DBI_0.3.1            proto_0.3-10         gridExtra_2.0.0      genefilter_1.52.0
## [21] cluster_2.0.3        stringr_1.0.0        locfit_1.5-9.1       nnet_7.3-11
## [25] grid_3.2.2           AnnotationDbi_1.32.0 XML_3.98-1.3         survival_2.38-3
## [29] BiocParallel_1.4.0   foreign_0.8-66       latticeExtra_0.6-26  Formula_1.2-1
## [33] geneplotter_1.48.0   ggplot2_1.0.1        reshape2_1.4.1       lambda.r_1.1.7
## [37] magrittr_1.5         scales_0.3.0         Hmisc_3.17-0         MASS_7.3-44
## [41] splines_3.2.2        xtable_1.7-4         BiocStyle_1.8.0      colorspace_1.2-6
## [45] stringi_0.5-5        acepack_1.3-3.3      munsell_0.4.2
```

References

[1] Love MI, Huber W, Anders S: *Moderated Estimation of Fold Change and Dispersion for RNA-Seq Data with DESeq2*. Genome Biology 2014, 15:550. A Comparison of Methods: Normalizing High-Throughput RNA Sequencing Data.

[2] Reddy R: *A Comparison of Methods: Normalizing High-Throughput RNA Sequencing Data. Cold Spring Harbor Labs Journals*. bioRxiv 2015:1-9.

[3] Hsieh AC, Liu Y, Edlind MP, et al.: *The translational landscape of mTOR signaling steers cancer initiation and metastasis*. Nature 2012, 485:55-61.