Supplementary methods

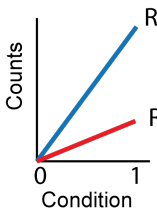
Detection of differential translation genes (DTGs) using DESeq2 interaction term

Here we present the protocol to carry out differential translation analysis using DESeq2 interaction term, and also pointers on how to interpret and visualize this data.

design =~ c + s + c : s where c is condition and s is sequencing type

$\log(\text{counts}_{c,s}) = \beta_0 + \beta_1.(c) + \beta_2.(s) + \beta_3.(c).(s)$

where c=0,1 and s=0,1



	Sequencing	Condition	Log fold changes from DESeq2	Measures changes in:
RNA (0)	0 -> 0	0 -> 1	$\log(\text{counts}_{c=1,s=0}) - \log(\text{counts}_{c=0,s=0})$ $= (\beta_0 + \beta_1) - \beta_0 = \beta_1$	Transcription
Ribo (1)	1 -> 1	0 -> 1	$\log(\text{counts}_{c=1,s=1}) - \log(\text{counts}_{c=0,s=1})$ $= (\beta_0 + \beta_1 + \beta_2 + \beta_3) - (\beta_0 + \beta_2) = \beta_1 + \beta_3$	Transcription + Translation

Therefore, β_3 = coefficient of the interaction term = Translation changes

$$= [\beta_1 + \beta_3] - \beta_1$$

$$= [\log(\text{counts}_{c=1,s=1}) - \log(\text{counts}_{c=0,s=1})] - [\log(\text{counts}_{c=1,s=0}) - \log(\text{counts}_{c=0,s=0})]$$

$$= [\log(\text{counts}_{c=1,s=1}) - \log(\text{counts}_{c=1,s=0})] - [\log(\text{counts}_{c=0,s=1}) - \log(\text{counts}_{c=0,s=0})]$$

$$= \log\left(\frac{\text{counts}_{c=1,s=1}}{\text{counts}_{c=1,s=0}}\right) - \log\left(\frac{\text{counts}_{c=0,s=1}}{\text{counts}_{c=0,s=0}}\right)$$

$$= \log\left(\frac{\text{Ribosome occupancy at 1}}{\text{mRNA abundance at 1}}\right) - \log\left(\frac{\text{Ribosome occupancy at 0}}{\text{mRNA abundance at 0}}\right)$$

= log(TE at condition 1) - log(TE at condition 0), where TE is the translation efficiency

Figure 1: Interaction term for differential translation regulation: Given an experimental setup with Ribo-seq (s = 1) and RNA-seq (s = 0) carried out at multiple conditions (c, here 0 and 1), the change in ribosome occupancy is confounded by the change in mRNA levels. Here we show that the interaction term coefficient is the actual change in translation correcting for change driven by transcription. The design with the interaction term, translates to a linear equation corresponding to the log counts at a given c (condition) and s (sequencing type). Substituting the c and s values in the equation given above, would then give us the transcription and translation (confounded) fold changes. So to obtain the translation only fold change we have to subtract from it the transcription effect which gives us the interaction term coefficient B3 as shown above. Furthermore, we show that this interaction term coefficient can give us the fold change in translation efficiency

```
library(DESeq2)
```

Input files

Calculating differential translation genes (DTGs) requires the count matrices from Ribo-seq and RNA-seq. These should be the raw counts obtained from feature counts or any other tool for counting reads, they should not be normalized or batch corrected.

1) Count matrix files (Ribo-seq and RNA-seq)

GeneID	Sample1	Sample2	Sample3	Sample4
ENSG1	1290	130	2	10
ENSG2	0	2	10	5

It also requires a sample information file which should be in the same order as samples in the count matrices. It should include information on sequencing type, treatment, batch or any other covariate you need to model.

2) Sample Information file

SampleID	Condition	SeqType	Batch
Sample1	Control	Ribo-seq	1
Sample2	Control	Ribo-seq	2
Sample3	Condition	Ribo-seq	3
Sample4	Condition	Ribo-seq	4
Sample5	Control	RNA-seq	1
Sample6	Control	RNA-seq	2
Sample7	Condition	RNA-seq	3
Sample8	Condition	RNA-seq	4

```
### Get count matrix files and sample information

# Input filenames
ribo_file <- "ribo_sample.txt"
rna_file <- "rna_sample.txt"
data_file <- "sample_info.txt"

# Read and merge count matrices
ribo <- read.delim(ribo_file)
rna <- read.delim(rna_file)
merge <- cbind(ribo,rna)

head(merge)
```

```
##   ribo_cond1_S1 ribo_cond1_S2 ribo_cond1_S3 ribo_cond1_S4 ribo_cond2_S1
## 1           189           202            85            35           541
## 2            86            45            69            45            89
## 3           117            86           171           157           101
## 4             3             7             2            11            19
## 5           817           425          1348           560           858
## 6           418           533           147           187           157
##   ribo_cond2_S2 ribo_cond2_S3 ribo_cond2_S4 rna_cond1_S1 rna_cond1_S2
## 1           236           208           213         9158        10144
```

```
## 2            33            38            51          407          202
## 3           301           195            80          319          308
## 4            16            13            15          188          226
## 5          1135          2471           686         1019          516
## 6           571           354           202         2278         2240
##   rna_cond1_S3 rna_cond1_S4 rna_cond2_S1 rna_cond2_S2 rna_cond2_S3
## 1         4154         1854        26708        11236         9879
## 2          362          118          106           27           52
## 3          234          448          518          254          767
## 4          217          285          651          800         1046
## 5          305          574         1039          604          443
## 6         3587         1843         3172         1074         4358
##   rna_cond2_S4
## 1         9849
## 2           46
## 3          543
## 4          261
## 5          535
## 6         2777
```

```
# Sample information file
coldata <- read.delim(data_file)
coldata <- as.data.frame(apply(coldata,2,as.factor))

head(coldata)
```

```
##        SampleID Condition  SeqType Batch
## 1 ribo_cond1_S1         1 Ribo-seq     1
## 2 ribo_cond1_S2         1 Ribo-seq     2
## 3 ribo_cond1_S3         1 Ribo-seq     3
## 4 ribo_cond1_S4         1 Ribo-seq     4
## 5 ribo_cond2_S1         2 Ribo-seq     1
## 6 ribo_cond2_S2         2 Ribo-seq     2
```

Detecting differential translation regulation

We implemented the interaction term model to identify DTGs as explained in Figure 1.

```
### DESeq2 object with batch and interaction term in the design
ddsMat <- DESeqDataSetFromMatrix(countData = merge,
            colData = coldata, design =~ Batch + Condition + SeqType + Condition:SeqType)

ddsMat$SeqType = relevel(ddsMat$SeqType,"RNA-seq")
ddsMat <- DESeq(ddsMat)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
resultsNames(ddsMat)
```

```
## [1] "Intercept"                    "Batch_2_vs_1"
## [3] "Batch_3_vs_1"                 "Batch_4_vs_1"
## [5] "Condition_2_vs_1"             "SeqType_Ribo.seq_vs_RNA.seq"
## [7] "Condition2.SeqTypeRibo.seq"
```

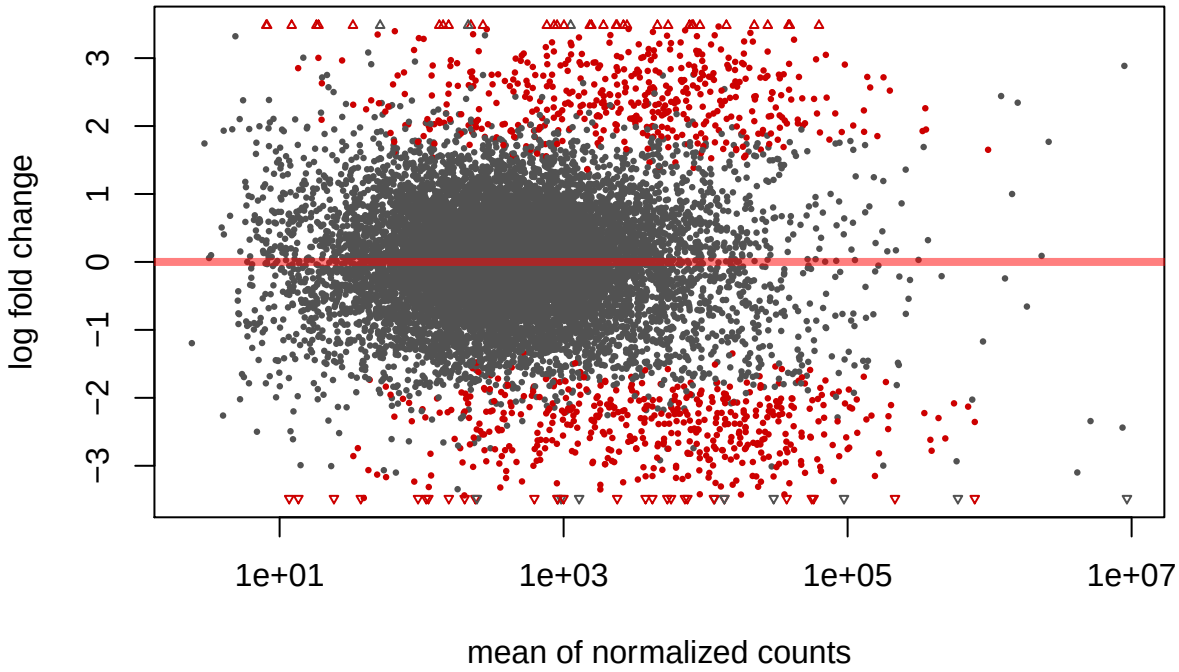
```
# Choose the term you want to look at from resultsNames(ddsMat)
# Condition2.SeqTypeRibo.seq means Changes in Ribo-seq levels in Condition2 vs
# Condition1 accounting for changes in RNA-seq levels in Condition2 vs Condition1
res <- results(ddsMat, contrast=list("Condition2.SeqTypeRibo.seq"))

summary(res)
```

```
##
## out of 13163 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 506, 3.8%
## LFC < 0 (down)    : 511, 3.9%
## outliers [1]      : 0, 0%
## low counts [2]    : 0, 0%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
length(which(res$padj < 0.05))
```

```
## [1] 821
```

```
DESeq2::plotMA(res)
```




```
### If you have multiple conditions or a time-series you can use the LRT as below

ddsMat_LRT <- DESeq(ddsMat, test="LRT", reduced =~ Batch + Condition + SeqType,
```

```
                    full =~ Batch + Condition + SeqType + Condition:SeqType)
```

```
## using pre-existing size factors
## estimating dispersions
## found already estimated dispersions, replacing these
## gene-wise dispersion estimates
## mean-dispersion relationship
## final dispersion estimates
## fitting model and testing
```

```
res_lrt <- results(ddsMat_LRT)

summary(res_lrt)
```

```
##
## out of 13163 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)     : 470, 3.6%
## LFC < 0 (down)   : 479, 3.6%
## outliers [1]     : 0, 0%
## low counts [2]   : 0, 0%
## (mean count < 2)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results
```

```
length(which(res_lrt$padj < 0.05))
```

```
## [1] 759
```

Visualisation and interpretation

DTGs can be of several categories:

1) Purely translation regulated
2) Regulated by both transcription and translation
    a) Buffered
    b) Amplified
    c) Weakened

Depending on differential expression for RNA-seq and Ribo-seq of the genes you can categorize them into these. So for example, for the gene below the mRNA abundance is increasing but the ribosome occupancy is remaining constant. It is identified as a DTG, for which the translation regulation counteracts the transcription.

```
### DESeq2 object with batch for Ribo-seq
ind = which(coldata$SeqType == "Ribo-seq")
coldata_ribo = coldata[ind,]
ddsMat_ribo <- DESeqDataSetFromMatrix(countData = ribo,
            colData = coldata_ribo, design =~ Batch + Condition)
ddsMat_ribo <- DESeq(ddsMat_ribo)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

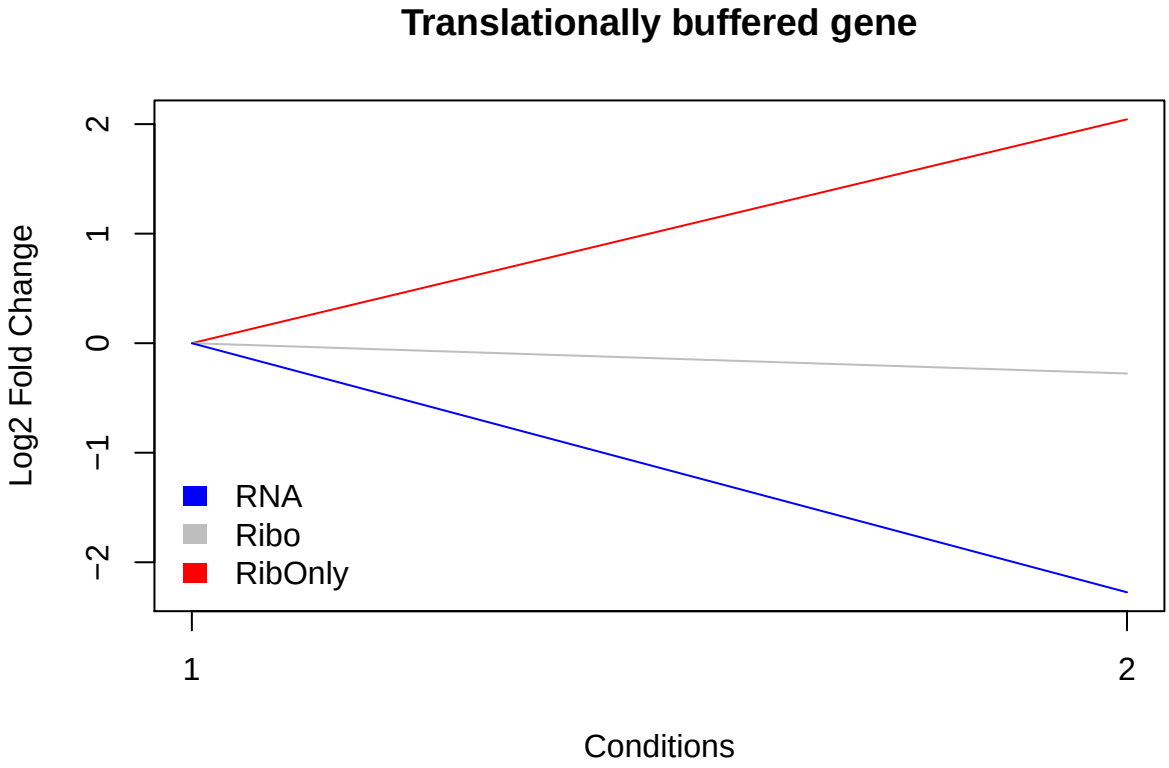
```
res_ribo <- results(ddsMat_ribo, contrast=list("Condition_2_vs_1"))
```

```
### DESeq2 object with batch for RNA-seq
ind = which(coldata$SeqType == "RNA-seq")
coldata_rna = coldata[ind,]
ddsMat_rna <- DESeqDataSetFromMatrix(countData = rna,
            colData = coldata_rna, design =~ Batch + Condition)
ddsMat_rna <- DESeq(ddsMat_rna)
```

```
## estimating size factors

## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
res_rna <- results(ddsMat_rna, contrast=list("Condition_2_vs_1"))
```

```
DTG = which(res$padj < 0.05)[1]

### Example of a differentially translated gene
y_u = max(res$log2FoldChange[DTG],res_ribo$log2FoldChange[DTG], res_rna$log2FoldChange[DTG])
y_l = min(res$log2FoldChange[DTG],res_ribo$log2FoldChange[DTG], res_rna$log2FoldChange[DTG])
plot(c(1,2),c(0,res$log2FoldChange[DTG]),type="l",col="red",xaxt="n",
     xlab="Conditions",ylim=c(y_l,y_u),ylab="Log2 Fold Change",
     main="Translationally buffered gene")
lines(c(1,2),c(0,res_ribo$log2FoldChange[DTG]),col="gray")
lines(c(1,2),c(0,res_rna$log2FoldChange[DTG]),col="blue")
axis(1,at=c(1,2),labels=c(1,2),las=1)
legend("bottomleft",c("RNA","Ribo","RibOnly"), fill=c("blue","gray","red"),
       cex=1, border = NA, bty="n")
```



We applied the above given script to obtain DTGs in primary human fibroblasts which were stimulated with TGFB and captured at 5 time points (This data also conatined a sample batch effect). In order to compare DTG detection on real data, we execute Xtail, RiboDiff and Riborex in default settings. Since there were 5 time points (4 after stimulation, 1 basal) we did pairwise comparisons of each time point vs basal to obtain all the possible DTGs.