

# BCH 339N Project

Alice Gee

4/15/2022

```
library(BiocManager)
library(DESeq2)
library(BiocParallel)
library(ggplot2)
library(dplyr)
library(tidyverse)
library(tidyr)
library(sys)
library(knitr)
library(pheatmap)
```

```

## rnaCounts: data.frame containing RNA-seq counts for each gene in each sample
## (genes are in rows of data.frame, samples in columns):
rnaCounts = read.table("rna_counts.tsv.gz",
                       sep="\t", header=TRUE, row.names=1, check.names=FALSE)

## riboCounts: data.frame containing ribo-seq counts for each gene in each sample
## (genes are in rows of data.frame, samples in columns):
riboCounts = read.table("ribo_counts.tsv.gz",
                        sep="\t", header=TRUE, row.names=1, check.names=FALSE)

## sampleAnnotation: data.frame with one row per sample; columns say what
## group (=combination of genotype+time), genotype, and time
## describe each sample:
sampleAnnotation = read.table("rna_sample_annotation.tsv",
                               sep="\t", header=TRUE, row.names=1, check.names=FALSE)

## sampleAnnotation2: data.frame with one row per sample; columns say what
## group (=combination of genotype+time), genotype, and time
## describe each sample:
sampleAnnotation2 = read.table("ribo_sample_annotation.tsv",
                               sep="\t", header=TRUE, row.names=1, check.names=FALSE)

## geneNamesAndDescriptions: data.frame with rownames corresponding to gene
## ids and three columns:
## (1) gene :: gene id (same as rownames,
## (2) symbol :: gene name/symbol
## (3) description :: gene description
geneNamesAndDescriptions = read.table("arabidopsis_thaliana_gene_names.tsv.gz",
                                      sep="\t", row.names=1, header=TRUE,
                                      quote="", comment.char="")
geneNamesAndDescriptions$gene = rownames(geneNamesAndDescriptions)
geneNamesAndDescriptions =
  geneNamesAndDescriptions[ , c("gene", "symbol", "description")]

## goAssociations: data.frame indicating what genes are associated with what
## gene sets, with four columns:
## (1) gene_ontology_primary_id :: gene set identifier
## (2) gene_ontology_name :: gene set name
## (3) gene_ontology_all_ids :: indicates what other gene ontology groups
##                               have been merged into this gene set; you
##                               don't need to worry about this column!
## (4) gene :: the gene ids associated with the gene set identified by
##             gene_ontology_primary_id column
goAssociations = read.table("gene_sets.tsv.gz",
                            sep="\t", row.names=NULL, header=TRUE,
                            quote="", comment.char="")

# ontogenic groups with the manually added genes
goAssociations2 = read.table("gene_sets2.csv",
                             sep= ",", row.names=NULL, header=TRUE,
                             quote="", comment.char="")

groupColors = c(
  "14BENDDAY" = "orchid3",
  "14BEXDARK" = "darkorchid4",
  "4GENDDAY" = "royalblue2",
  "4GEXDARK" = "navy",
  "COLENDAY" = "seagreen1",

```

```

    "COLEXDARK" = "seagreen"
)

heatPalette = colorRampPalette(c("dodgerblue", "lightskyblue", "white",
                               "lightgoldenrod", "orangered"))(100)

```

# Preliminary RNA-Seq and Ribo-Seq

```

# remove the outlier group from rna-counts
rnaCounts <- rnaCounts %>% select(-`4GEXDARK4`)
sampleAnnotation <- sampleAnnotation[colnames(rnaCounts),]

# remove the outlier group from ribo-counts
riboCounts <- riboCounts %>% select(-`4GEXDARK4`)
sampleAnnotation2 <- sampleAnnotation2[colnames(riboCounts),]

```

## DEseq with RNA-seq data

```

DESeqDataSet = DESeqDataSetFromMatrix(
  countData = rnaCounts,
  colData = sampleAnnotation,
  design = ~ time + genotype + time:genotype
)

```

```

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors

```

```

DESeqDataSet = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ time + genotype
)

```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```

DESeq_Results_RNA <- results(DESeqDataSet)
clean_DESeq_padj <- which(!is.na(DESeq_Results_RNA$padj))
RNA_Sig <- sum(DESeq_Results_RNA[clean_DESeq_padj, "padj"] <= 0.1)
RNA_Sig

## [1] 1225

sum(DESeq_Results_RNA[clean_DESeq_padj, "padj"] <= 0.1) * 0.1

## [1] 122.5

summary(DESeq_Results_RNA)

## 
## out of 18283 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 736, 4%
## LFC < 0 (down)    : 489, 2.7%
## outliers [1]       : 12, 0.066%
## low counts [2]     : 3182, 17%
## (mean count < 19)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

## Identify significant ontogenic groups

```

clean_DESeq_Results <- DESeq_Results_RNA[clean_DESeq_padj, ]
significant_genes <- rownames(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1), ])
clean_DESeq_Results <- data.frame(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1),
])
clean_DESeq_Results$gene <- significant_genes
RNA_Result_groupings <- clean_DESeq_Results %>% inner_join(goAssociations, by = "gene")
RNA_Result_groupings %>% group_by(gene_ontology_name) %>% summarize(count = n()) %>% arrange(desc(count)) %>%
  knitr::kable(caption = "Distribution of Significant Genes Groupings for RNA-Seq")

```

Distribution of Significant Genes Groupings for RNA-Seq

gene_ontology_name	count
pyruvate metabolic process	18
amino acid transport	15
cellular aldehyde metabolic process	13
tetrapyrrole metabolic process	13
cold acclimation	11
protein-chromophore linkage	10
cellular response to endogenous stimulus	8
response to disaccharide	7

gene_ontology_name	count
response to ozone	7
anion homeostasis	6

```

clean_DESeq_padj <- which(!is.na(DESeq_Results_RNA$padj))
clean_DESeq_Results <- DESeq_Results_RNA[clean_DESeq_padj, ]
significant_genes <- rownames(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1), ])
clean_DESeq_Results <- data.frame(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1),
])
clean_DESeq_Results$gene <- significant_genes
RNA_Result_groupings <- clean_DESeq_Results %>% inner_join(goAssociations2, by = "gene")
RNA_Result_groupings %>% group_by(gene_ontology_name) %>% summarize(count = n()) %>% arrange(desc(count)) %>%
filter(count >= 10) %>%
knitr::kable(caption = "Distribution of Significant Genes Groupings for RNA-Seq")

```

### Distribution of Significant Genes Groupings for RNA-Seq

gene_ontology_name	count
pyruvate metabolic process	18
amino acid transport	16
response to light stimulus	14
cellular aldehyde metabolic process	13
tetrapyrrole metabolic process	13
cold acclimation	11
protein-chromophore linkage	10

### log transformed the data

```
lgNorm = log2(counts(DESeqDataSet, normalized=TRUE) + 1)
```

### get for identified genes and their ontogenic function

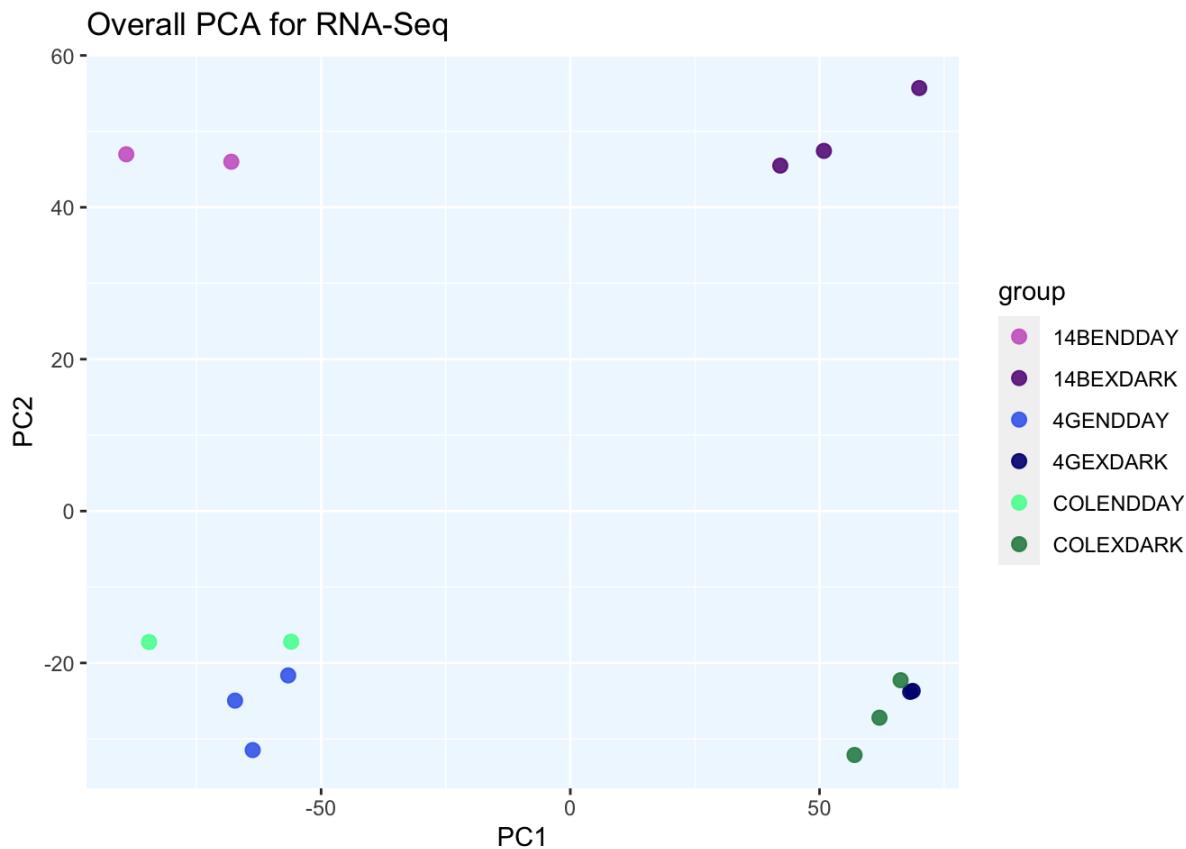
```

joined_set = inner_join(goAssociations, geneNamesAndDescriptions, by = "gene")
joined_set_ribo = inner_join(goAssociations, geneNamesAndDescriptions, by = "gene")
ontology_names <- joined_set %>% distinct(gene_ontology_name)
ontology_names <- ontology_names[["gene_ontology_name"]]
ontology_names_all <- goAssociations2 %>% distinct(gene_ontology_name)
ontology_names_all <- ontology_names_all[["gene_ontology_name"]]

```

### Overall PCA Plot

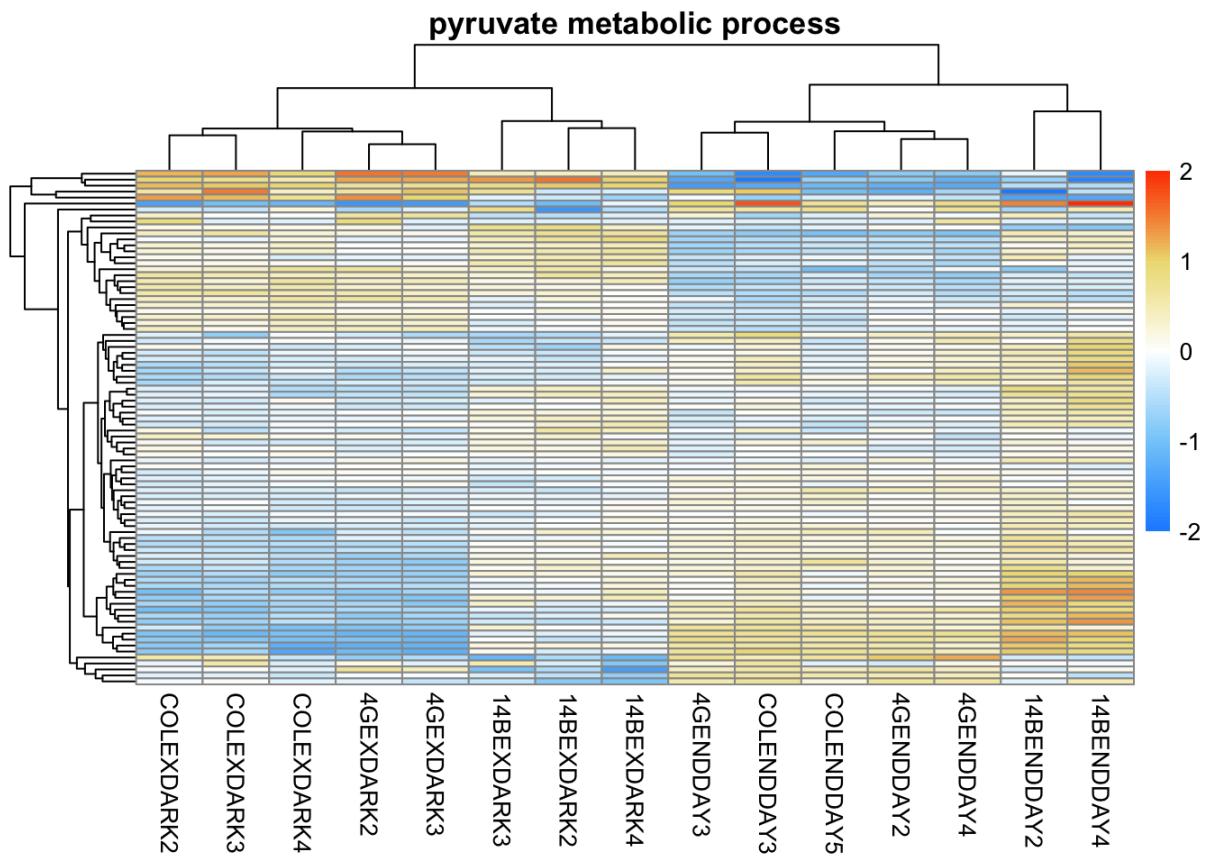
```
library(ggplot2)
pca = prcomp(t(lgNorm))
pcaData = data.frame(pca$x[, 1:2])
pcaData$group = sampleAnnotation[rownames(pcaData), "group"]
pcaData$sample = rownames(pcaData)
gg = ggplot(pcaData, aes(x=PC1, y=PC2, color=group, label=sample))
gg = gg + geom_point(size=2.5, alpha = 0.9)
gg = gg + scale_color_manual(values=groupColors)
gg = gg + theme(panel.background = element_rect(fill = 'aliceblue')) + ggtitle("Overall PCA for RNA-Seq")
print(gg)
```



## Specific pHeatmap per Gene Grouping

```
# "pyruvate metabolic process"
temp_set <- joined_set %>% filter(gene_ontology_name == "pyruvate metabolic process")
lgGo <- lgNorm[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  labels_row=geneNamesAndDescriptions[rownames(heatData), "symbol"],
  main = "pyruvate metabolic process",
  show_rownames = FALSE
)
```

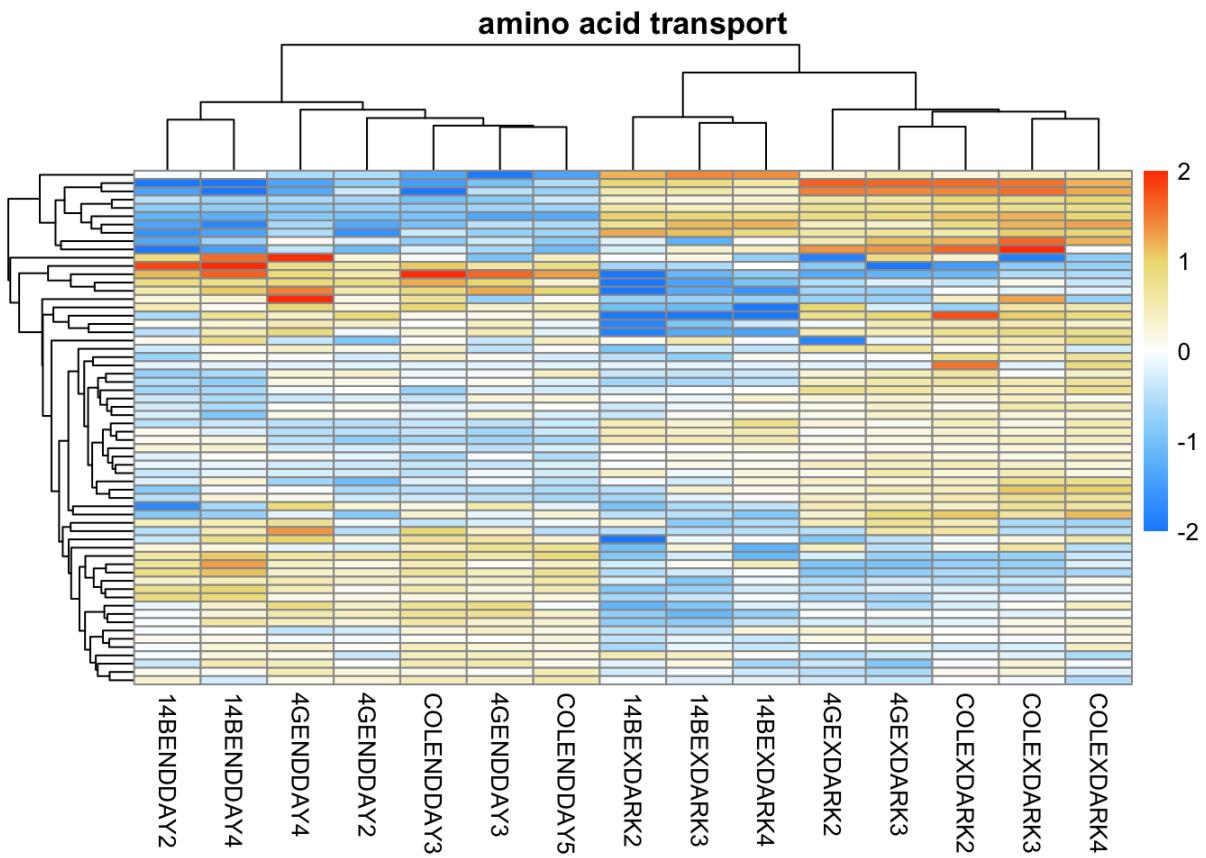


```

# "amino acid transport"
temp_set <- joined_set %>% filter(gene_ontology_name == "amino acid transport")
lgGo <- lgNorm[temp_set$gene, ]

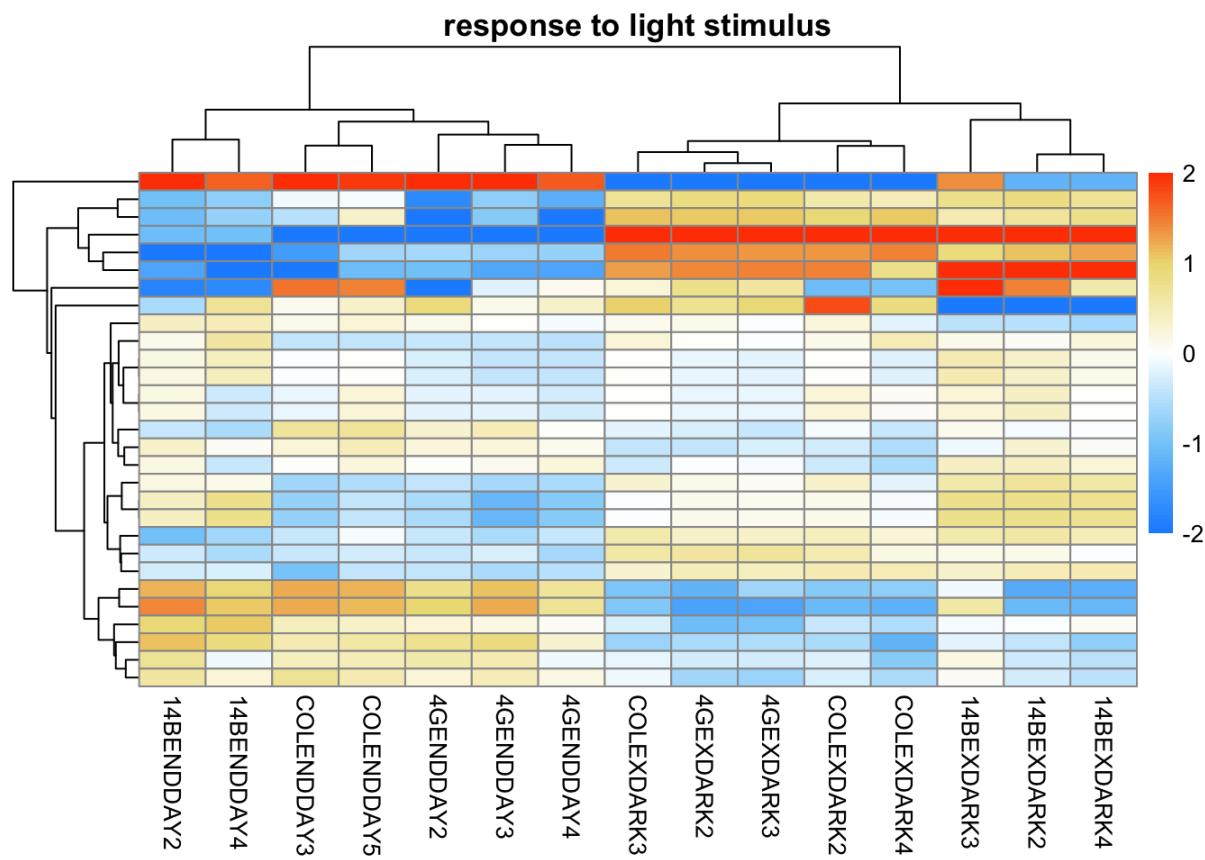
heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
    heatData,
    color = heatPalette,
    clustering_method = "average",
    labels_row= geneNamesAndDescriptions[rownames(heatData), "symbol"],
    main = "amino acid transport",
    show_rownames = FALSE
)

```



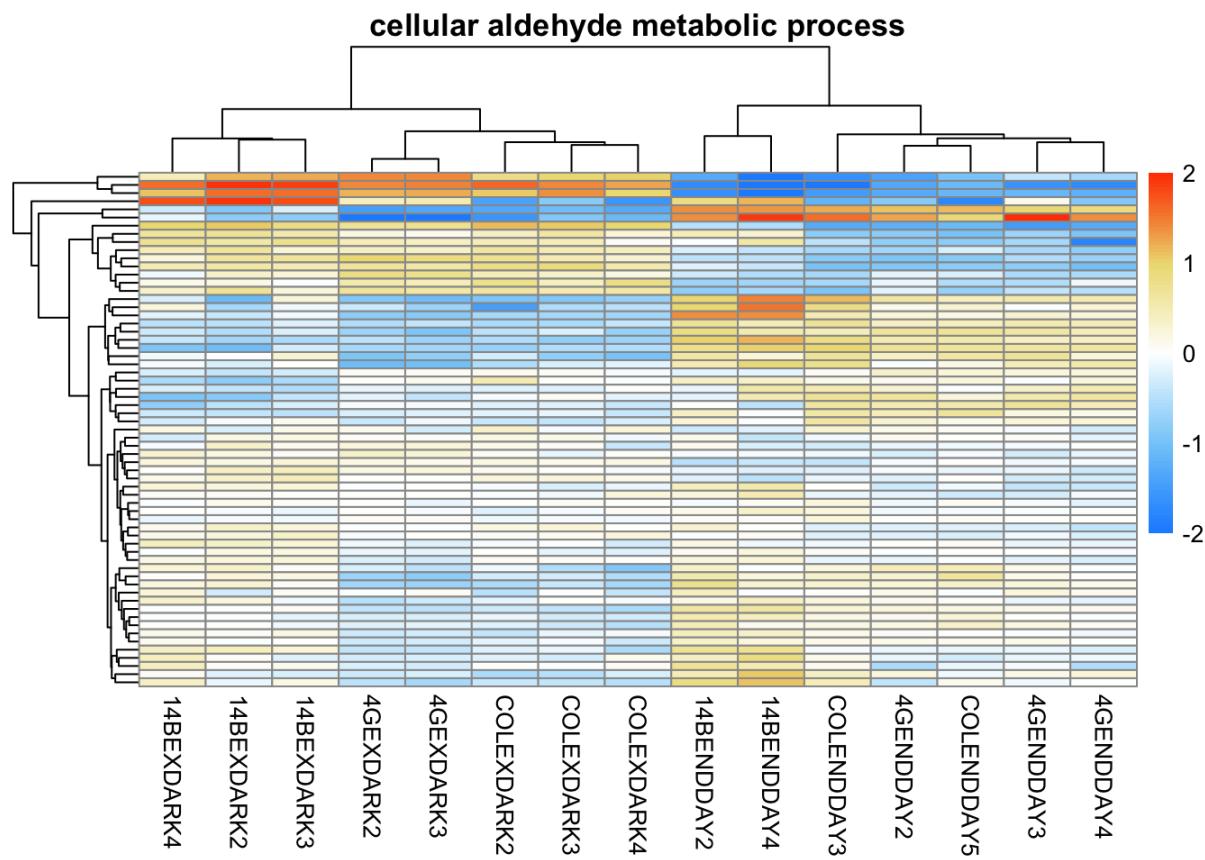
```
# "response to light stimulus"
temp_set <- goAssociations2 %>% filter(gene_ontology_name == "response to light stimulus")
lgGo <- lgNorm[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  main = "response to light stimulus",
  show_rownames = FALSE
)
```



```
# "cellular aldehyde metabolic process"
temp_set <- joined_set %>% filter(gene_ontology_name == "cellular aldehyde metabolic process")
lgGo <- lgNorm[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  labels_row= geneNamesAndDescriptions[rownames(heatData), "symbol"],
  main = "cellular aldehyde metabolic process",
  show_rownames = FALSE
)
```



## DEseq with ribo-seq data

```
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = riboCounts,
  colData = sampleAnnotation2,
  design = ~ time + genotype + time:genotype
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
DESeqDataSet = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ time + genotype
)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```

## final dispersion estimates

## fitting model and testing

DESeq_Results_ribo <- results(DESeqDataSet)

clean_DESeq_padj <- which(!is.na(DESeq_Results_ribo$padj))
Ribo_Sig <- sum(DESeq_Results_ribo[clean_DESeq_padj, "padj"] <= 0.1)
Ribo_Sig

## [1] 167

sum(DESeq_Results_ribo[clean_DESeq_padj, "padj"] <= 0.1) * 0.1

## [1] 16.7

summary(DESeq_Results_ribo)

## 
## out of 17868 with nonzero total read count
## adjusted p-value < 0.1
## LFC > 0 (up)      : 46, 0.26%
## LFC < 0 (down)    : 121, 0.68%
## outliers [1]       : 7, 0.039%
## low counts [2]     : 14489, 81%
## (mean count < 37)
## [1] see 'cooksCutoff' argument of ?results
## [2] see 'independentFiltering' argument of ?results

```

## normalize count data

```
lgNormRibo = log2(counts(DESeqDataSet, normalized=TRUE) + 1)
```

## Identify significant ontogenic groups

```

clean_DESeq_Results <- DESeq_Results_ribo[clean_DESeq_padj, ]
significant_genes <- rownames(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1), ])
clean_DESeq_Results <- data.frame(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1),
])
clean_DESeq_Results$gene <- significant_genes
RNA_Result_groupings <- clean_DESeq_Results %>% inner_join(goAssociations, by = "gene")
RNA_Result_groupings %>% group_by(gene_ontology_name) %>% summarize(count = n()) %>% arrange(desc(count)) %>% filter(count >= 2) %>%
knitr::kable(caption = "Distribution of Significant Genes Groupings For Ribo-Seq")

```

Distribution of Significant Genes Groupings For Ribo-Seq

gene_ontology_name	count
protein-chromophore linkage	8

gene_ontology_name	count
pyruvate metabolic process	6
cold acclimation	5
anion homeostasis	2
response to ozone	2

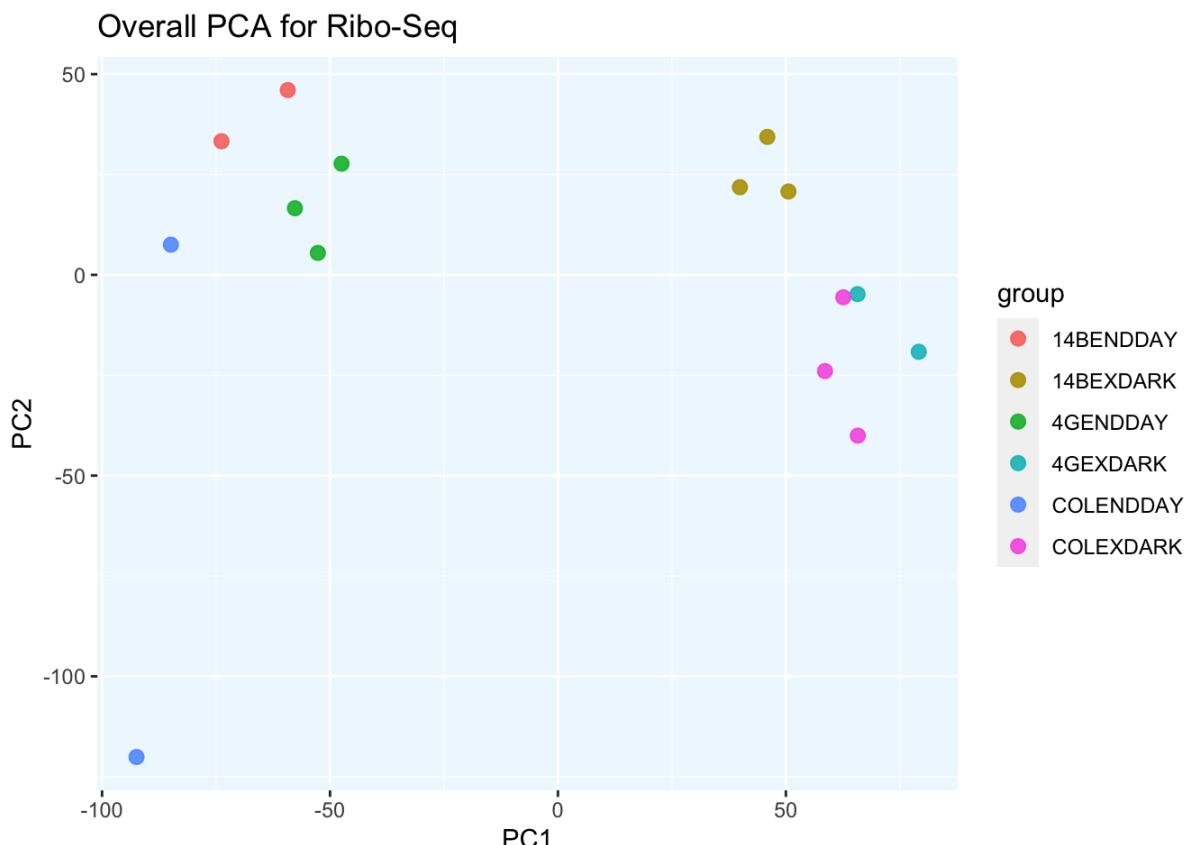
## get for identified genes and their ontogenic function

```
joined_set_ribo = inner_join(goAssociations, geneNamesAndDescriptions, by = "gene")
ontology_names <- joined_set_ribo %>% distinct(gene_ontology_name)
ontology_names <- ontology_names[["gene_ontology_name"]]
```

## Overall PCA for Ribo-Seq

```
pca = prcomp(t(lgNormRibo))
pcaData = data.frame(pca$x[, 1:2])
pcaData$group = sampleAnnotation[rownames(pcaData), "group"]
pcaData$sample = rownames(pcaData)

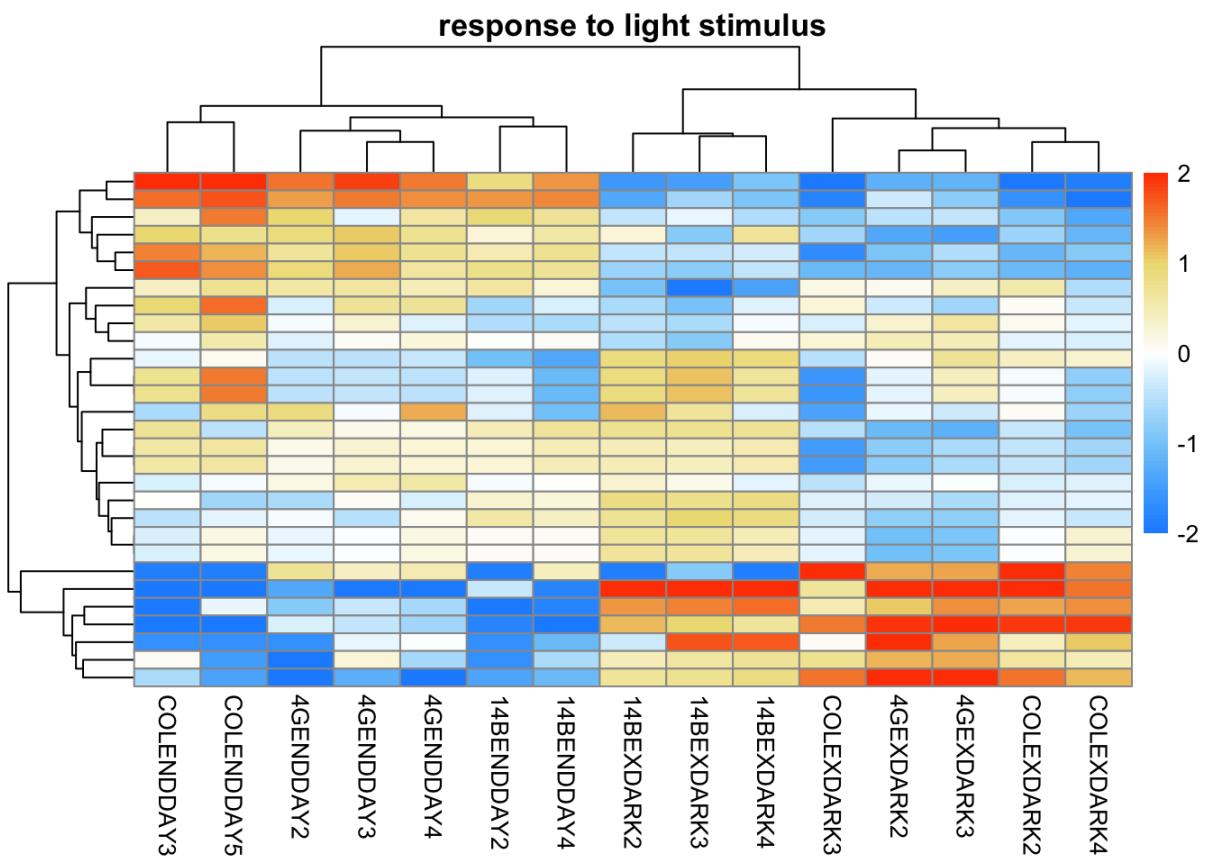
gg = ggplot(pcaData, aes(x=PC1, y=PC2, color=group, label=sample))
gg = gg + geom_point(size=2.5, alpha=0.9)
gg = gg + theme(panel.background = element_rect(fill = 'aliceblue')) + ggtitle("Overall PCA for Rib
o-Seq")
print(gg)
```



## pHeatmaps for specific groups

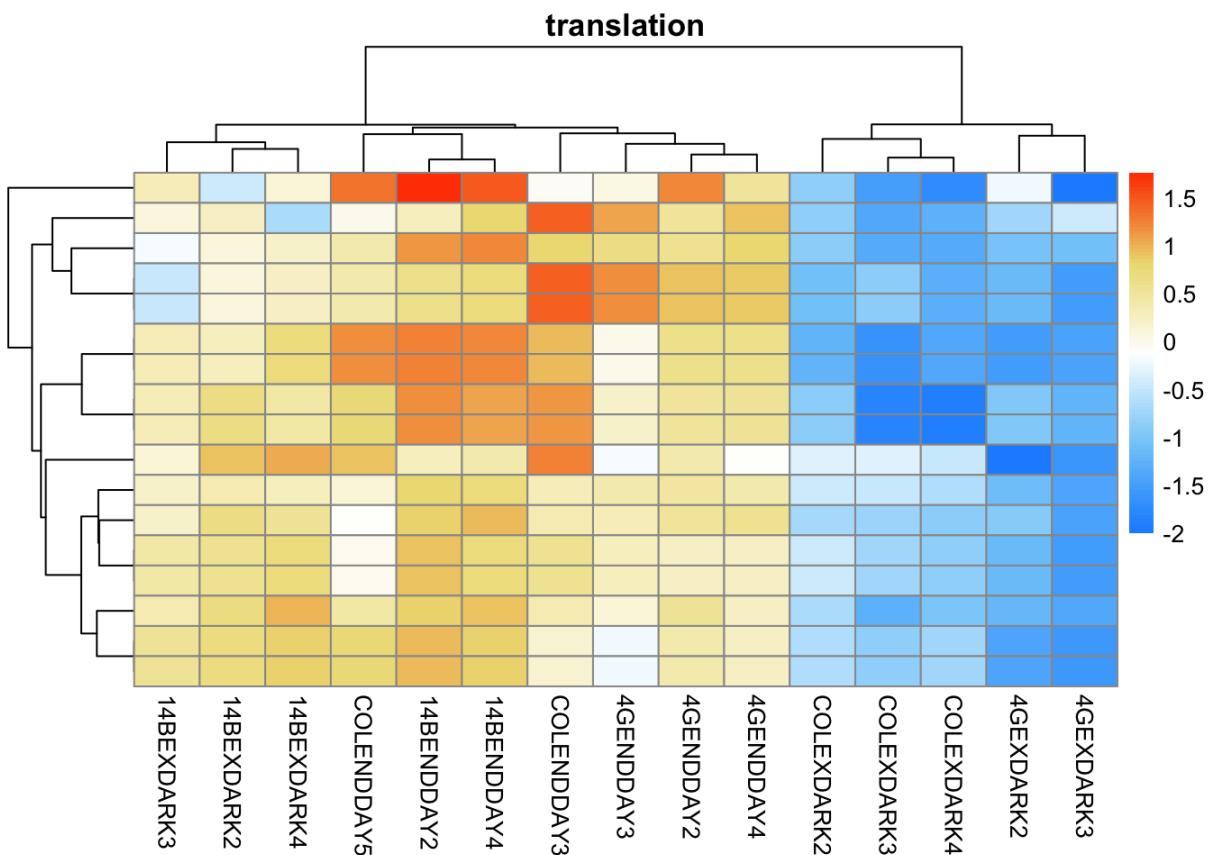
```
# "response to light stimulus"
temp_set <- goAssociations2 %>% filter(gene_ontology_name == "response to light stimulus")
lgGo <- lgNormRibo[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  main = "response to light stimulus",
  show_rownames = FALSE
)
```



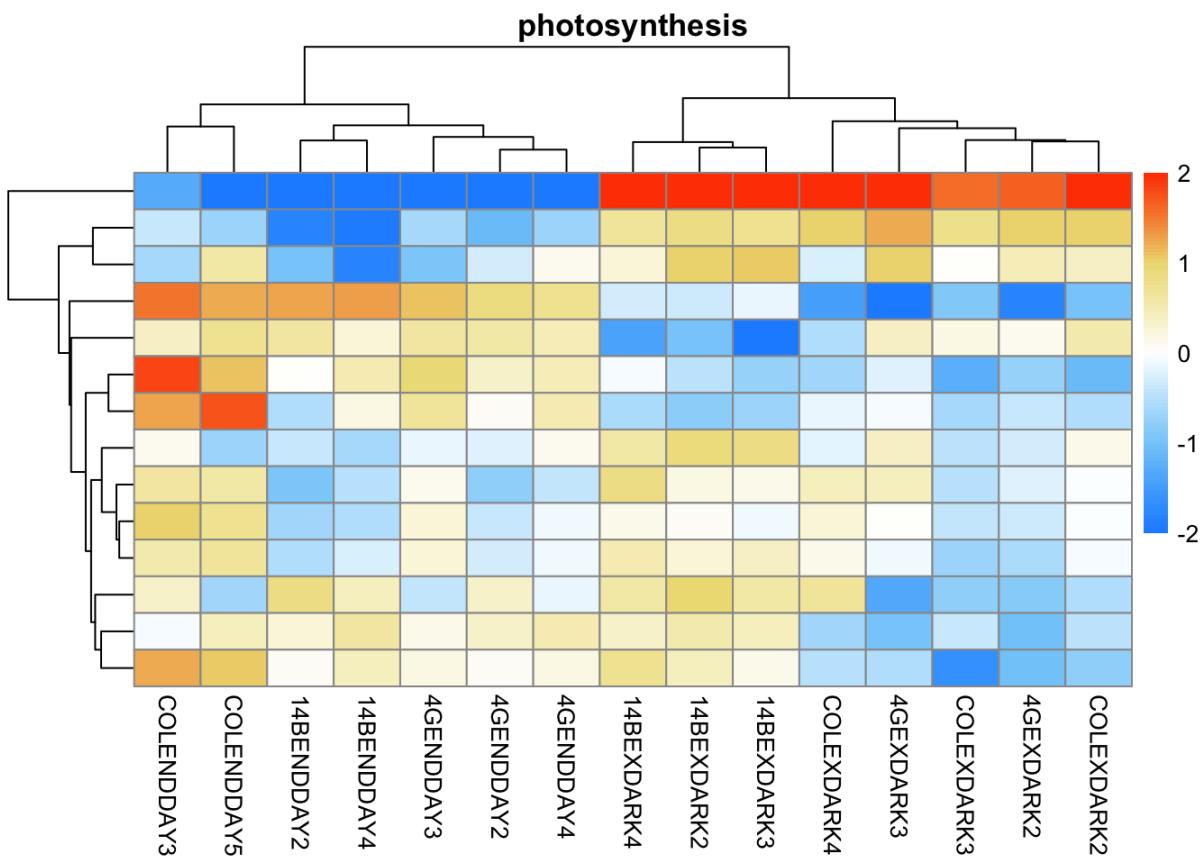
```
# "translation"
temp_set <- goAssociations2 %>% filter(gene_ontology_name == "translation")
lgGo <- lgNormRibo[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  main = "translation",
  show_rownames = FALSE
)
```



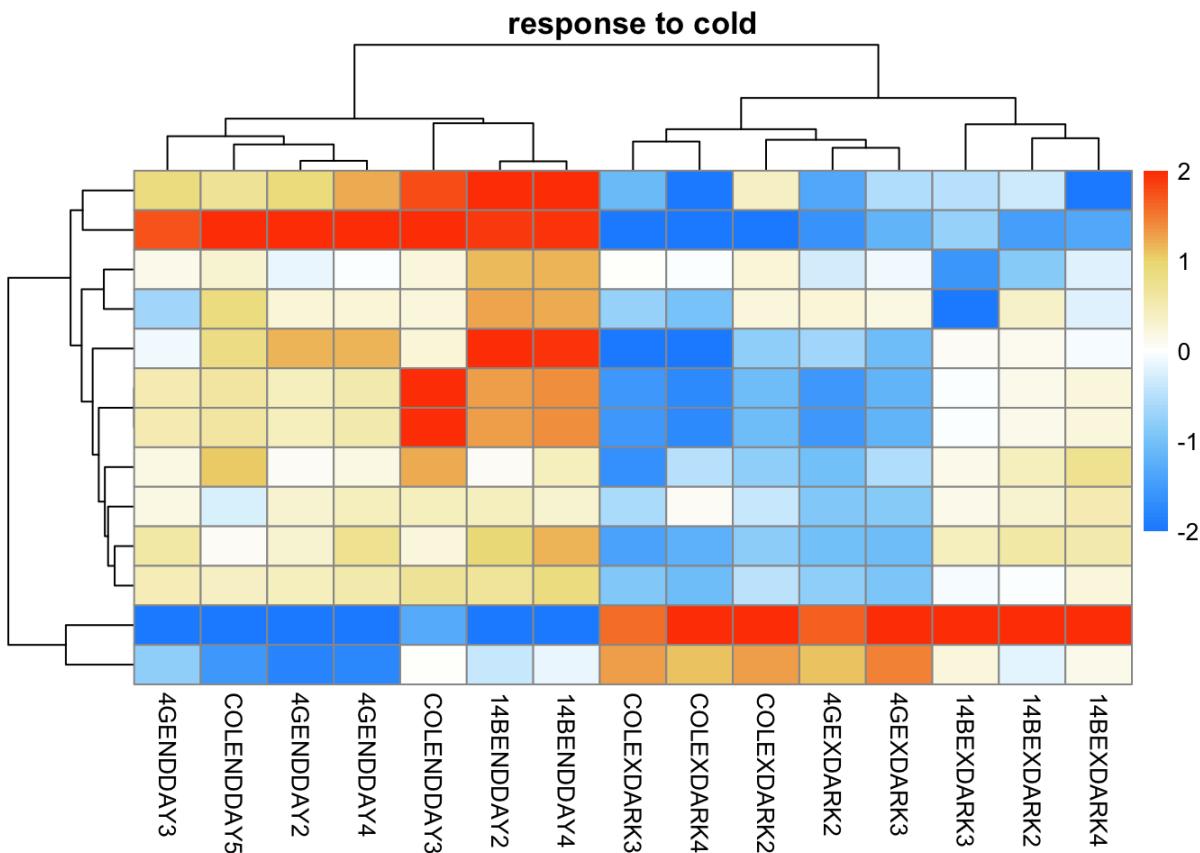
```
# "photosynthesis"
temp_set <- goAssociations2 %>% filter(gene_ontology_name == "photosynthesis")
lgGo <- lgNormRibo[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  main = "photosynthesis",
  show_rownames = FALSE
)
```



```
# "response to cold"
temp_set <- goAssociations2 %>% filter(gene_ontology_name == "response to cold")
lgGo <- lgNormRibo[temp_set$gene, ]

heatData = lgGo - rowMeans(lgGo)
heatData = as.data.frame(heatData)
heatData[heatData > 2] = 2; heatData[heatData < -2] = -2
fontsize_row = 10 - nrow(heatData) / 15
pheatmap(
  heatData,
  color = heatPalette,
  clustering_method = "average",
  main = "response to cold",
  show_rownames = FALSE
)
```



## Translational Efficiency

###Comparison genotypes 14B and 4G, Contrast A

```

riboCountsA <- riboCounts %>% select(-contains("COL"))
rnaCountsA <- rnaCounts %>% select(-contains("COL"))
sampleAnnotationA <- sampleAnnotation %>% filter(!str_detect(group, 'COL'))
sampleAnnotation2A <- sampleAnnotation2 %>% filter(!str_detect(group, 'COL'))

# rna and ribo
sampleAnnotationA$SeqType = "RNA"
sampleAnnotation2A$SeqType = "Ribo"
combinedCountsA = cbind(riboCountsA, rnaCountsA)
sampleAnnotation3A = rbind(sampleAnnotationA, sampleAnnotation2A)
colnames(combinedCountsA) = rownames(sampleAnnotation3A)

# time + genotype + time:genotype + SeqType + SeqType:time + SeqType:genotype + SeqType:time:genotype
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = combinedCountsA,
  colData = sampleAnnotation3A,
  design = ~ time * genotype * SeqType
)

```

```

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors

```

```

DESeqDataSet_both = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ (time + genotype + SeqType)^2
)

```

```

## estimating size factors

```

```

## estimating dispersions

```

```

## gene-wise dispersion estimates

```

```

## mean-dispersion relationship

```

```

## final dispersion estimates

```

```

## fitting model and testing

```

```

DESeq_Results_both <- results(DESeqDataSet_both)
clean_DESeq_padj <- which(!is.na(DESeq_Results_both$padj))
sum(DESeq_Results_both[clean_DESeq_padj, "padj"] <= 0.1)

```

```

## [1] 206

```

```

sum(DESeq_Results_both[clean_DESeq_padj, "padj"] <= 0.1) * 0.1

```

```
## [1] 20.6
```

```
# rna  
DESeqDataSet = DESeqDataSetFromMatrix(  
  countData = rnaCountsA,  
  colData = sampleAnnotationA,  
  design = ~ time + genotype + time:genotype  
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
## design formula are characters, converting to factors
```

```
DESeqDataSet = DESeq(  
  DESeqDataSet,  
  parallel=FALSE,  
  test = "LRT",  
  reduced = ~ time + genotype  
)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
DESeq_Results_RNA <- results(DESeqDataSet)
```

```
# ribo  
DESeqDataSet = DESeqDataSetFromMatrix(  
  countData = riboCountsA,  
  colData = sampleAnnotation2A,  
  design = ~ time + genotype + time:genotype  
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
## design formula are characters, converting to factors
```

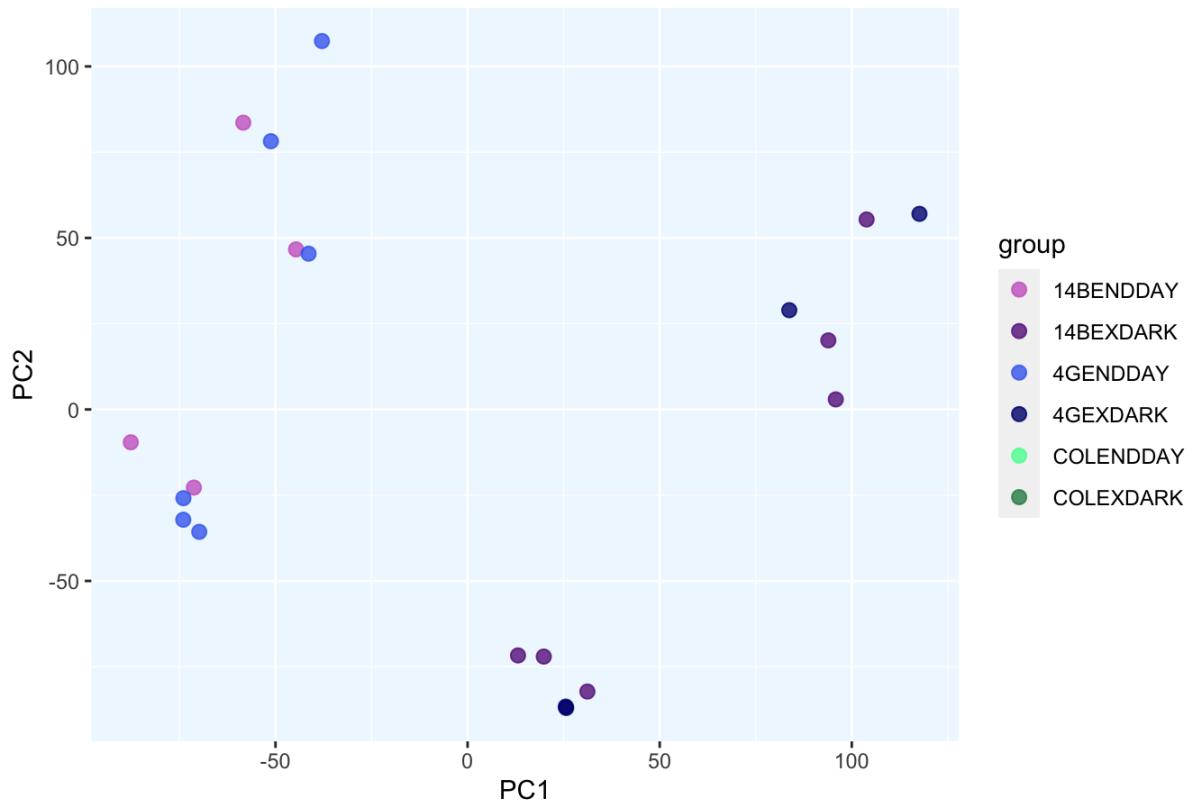
```
DESeqDataSet = DESeq(  
  DESeqDataSet,  
  parallel=FALSE,  
  test = "LRT",  
  reduced = ~ time + genotype  
)
```

```
## estimating size factors  
  
## estimating dispersions  
  
## gene-wise dispersion estimates  
  
## mean-dispersion relationship  
  
## final dispersion estimates  
  
## fitting model and testing  
  
DESeq_Results_ribo <- results(DESeqDataSet)  
  
lgNorm = log2(counts(DESeqDataSet_both, normalized=TRUE) + 1)
```

## Overall PCA Plot

```
pca = prcomp(t(lgNorm))  
pcaData = data.frame(pca$x[ , 1:2])  
pcaData$group = sampleAnnotation3A[rownames(pcaData), "group"]  
pcaData$sample = rownames(pcaData)  
gg = ggplot(pcaData, aes(x=PC1, y=PC2, color=group, label=sample))  
gg = gg + geom_point(size=2.5, alpha=0.8)  
gg = gg + scale_color_manual(values=groupColors)  
gg = gg + theme(panel.background = element_rect(fill = 'aliceblue')) + ggtitle("Overall PCA for Tra  
nslational Efficiency")  
print(gg)
```

### Overall PCA for Translational Efficiency



```
clean_DESeq_padj <- which(!is.na(DESeq_Results_both$padj))
clean_DESeq_Results <- DESeq_Results_both[clean_DESeq_padj, ]
significant_genes <- rownames(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1), ])
clean_DESeq_Results <- data.frame(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1),
])
clean_DESeq_Results$gene <- significant_genes
RNA_Result_groupings <- clean_DESeq_Results %>% inner_join(goAssociations2, by = "gene")
RNA_Result_groupings %>% group_by(gene_ontology_name) %>% summarize(count = n()) %>% arrange(desc(count)) %>%
  filter(count >= 5) %>%
  knitr::kable(caption = "Distribution of Significant Genes Groupings For TE")
```

### Distribution of Significant Genes Groupings For TE

gene_ontology_name	count
response to light stimulus	14
translation	14
pyruvate metabolic process	9
response to cold	6
protein-chromophore linkage	5
translational elongation	5

```

exclusive = rownames(DESeq_Results_both)[which(DESeq_Results_both$padj < 0.1 & DESeq_Results_ribo$padj < 0.1 & DESeq_Results_RNA$padj > 0.1)]

both = rownames(DESeq_Results_both)[which(DESeq_Results_both$padj < 0.1 & DESeq_Results_ribo$padj < 0.1 & DESeq_Results_RNA$padj < 0.1)]

intensified = both[which(DESeq_Results_both[both,2]*DESeq_Results_RNA[both,2] > 0)]

buffered = both[which(DESeq_Results_both[both,2]*DESeq_Results_RNA[both,2] < 0)]
```

```

goAssociations2 %>% filter(gene %in% exclusive) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for Exclusive TE Results")
```

#### Gene Groupings for Exclusive TE Results

gene_ontology_name	count
translation	8
response to light stimulus	7
NADH metabolic process	2
protein-chromophore linkage	2
proton motive force-driven ATP synthesis	2
response to cold	2
cold acclimation	1
glucose metabolic process	1
photosynthesis	1
pyruvate metabolic process	1
tetrapyrrole metabolic process	1

```

goAssociations2 %>% filter(gene %in% both) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for 'Both' TE Results")
```

#### Gene Groupings for 'Both' TE Results

gene_ontology_name	count
gluconeogenesis	1
pyruvate metabolic process	1
response to light stimulus	1

```

goAssociations2 %>% filter(gene %in% intensified) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for Intensified TE Results")
```

#### Gene Groupings for Intensified TE Results

gene_ontology_name	count
gluconeogenesis	1
pyruvate metabolic process	1

```
goAssociations2 %>% filter(gene %in% buffered) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for Buffered TE Results")
```

### Gene Groupings for Buffered TE Results

gene_ontology_name	count
response to light stimulus	1

```
goAssociations2 %>% filter(gene %in% exclusive) %>% knitr::kable(caption = "Gene Groupings for Exclusive TE Results")
```

### Gene Groupings for Exclusive TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0009631 (GO:0009631)	cold acclimation	AT1G77490
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT3G47470
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT5G01530
GO:0006090 (GO:0006090)	pyruvate metabolic process	AT1G12900
GO:0006412 (GO:0006412)	translation	AT3G27160
GO:0015986 (GO:0015986)	proton motive force-driven ATP synthesis	AT4G32260
GO:0006734 (GO:0006734)	NADH metabolic process	AT5G58330
GO:0006006 (GO:0006006)	glucose metabolic process	AT1G12900
GO:0006412 (GO:0006412)	translation	AT2G43030
GO:0006412 (GO:0006412)	translation	AT2G33800
GO:0009416 (GO:0009416)	response to light stimulus	AT5G01530
GO:0006412 (GO:0006412)	translation	AT3G15190
GO:0033013 (GO:0033013)	tetrapyrrole metabolic process	AT2G33430
GO:0009416 (GO:0009416)	response to light stimulus	AT4G25650
GO:0009416 (GO:0009416)	response to light stimulus	AT1G55480
GO:0009416 (GO:0009416)	response to light stimulus	AT3G50685
GO:0009416 (GO:0009416)	response to light stimulus	AT3G05350
GO:0015986 (GO:0015986)	proton motive force-driven ATP synthesis	AT4G32260
GO:0015979 (GO:0015979)	photosynthesis	AT1G55480
GO:0006412 (GO:0006412)	translation	AT2G43030

gene_ontology_primary_id	gene_ontology_name	gene
GO:0006734 (GO:0006734)	NADH metabolic process	AT5G58330
GO:0006412 (GO:0006412)	translation	AT2G33800
GO:0009409 (GO:0009409)	response to cold	AT1G32060
GO:0009416 (GO:0009416)	response to light stimulus	AT5G01530
GO:0006412 (GO:0006412)	translation	AT4G01310
GO:0009409 (GO:0009409)	response to cold	AT5G12250
GO:0009416 (GO:0009416)	response to light stimulus	AT4G25650
GO:0006412 (GO:0006412)	translation	AT3G15190

```
goAssociations2 %>% filter(gene %in% both) %>%
  knitr::kable(caption = "Gene Groupings for 'Both' TE Results")
```

#### Gene Groupings for 'Both' TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0006090 (GO:0006090)	pyruvate metabolic process	AT3G12780
GO:0006094 (GO:0006094)	gluconeogenesis	AT3G12780
GO:0009416 (GO:0009416)	response to light stimulus	AT1G51200

```
goAssociations2 %>% filter(gene %in% intensified) %>%
  knitr::kable(caption = "Gene Groupings for Intensified TE Results")
```

#### Gene Groupings for Intensified TE Results

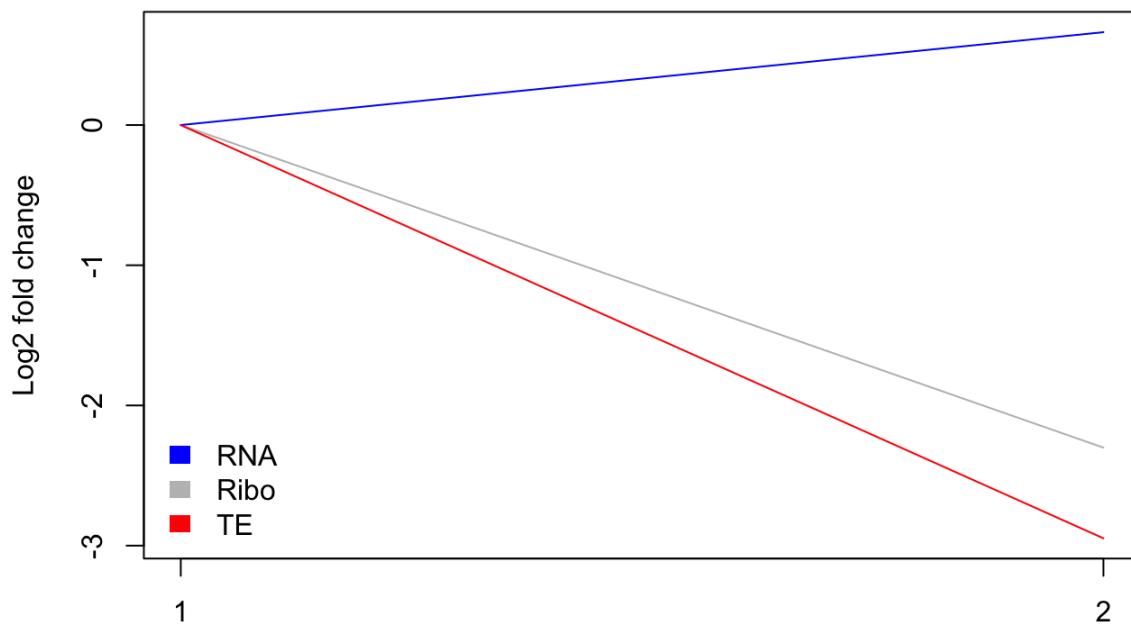
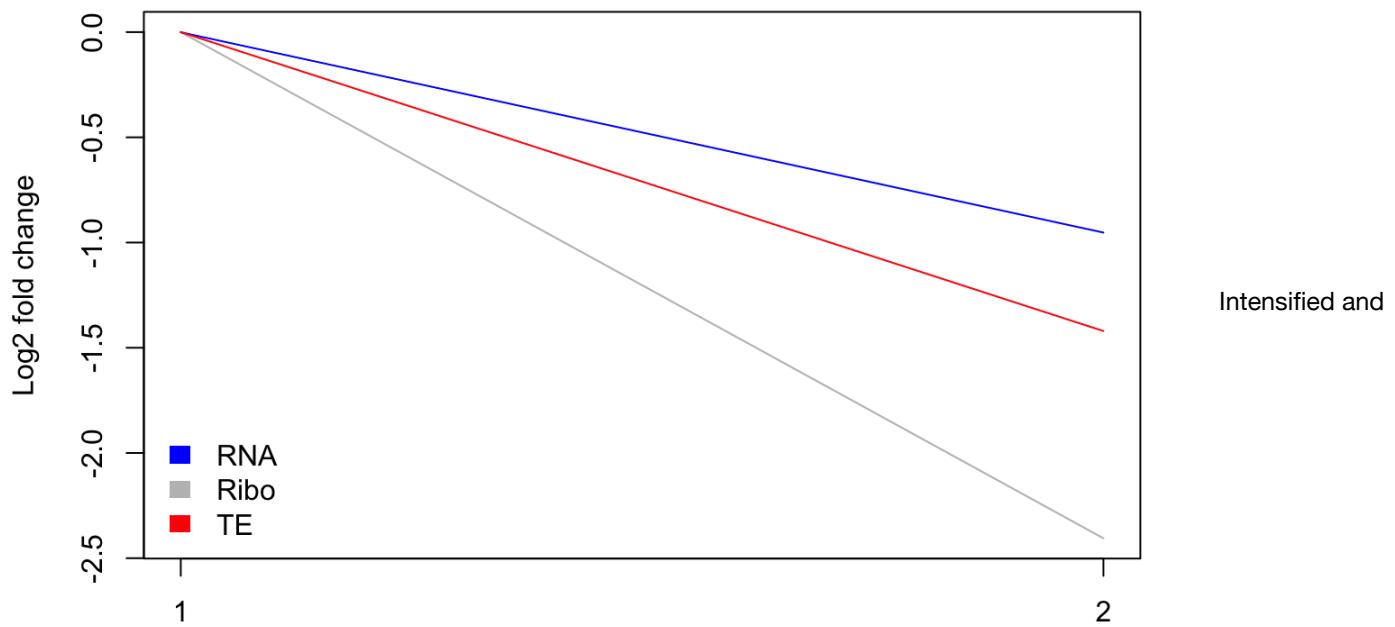
gene_ontology_primary_id	gene_ontology_name	gene
GO:0006090 (GO:0006090)	pyruvate metabolic process	AT3G12780
GO:0006094 (GO:0006094)	gluconeogenesis	AT3G12780

```
goAssociations2 %>% filter(gene %in% buffered) %>%
  knitr::kable(caption = "Gene Groupings for Buffered TE Results")
```

#### Gene Groupings for Buffered TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0009416 (GO:0009416)	response to light stimulus	AT1G51200

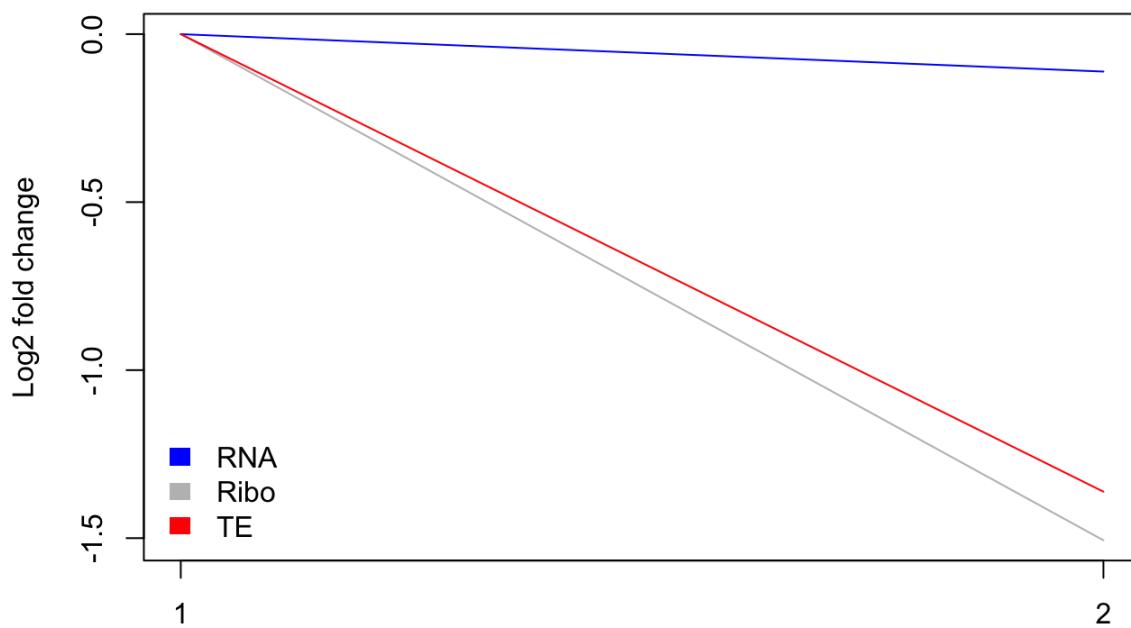
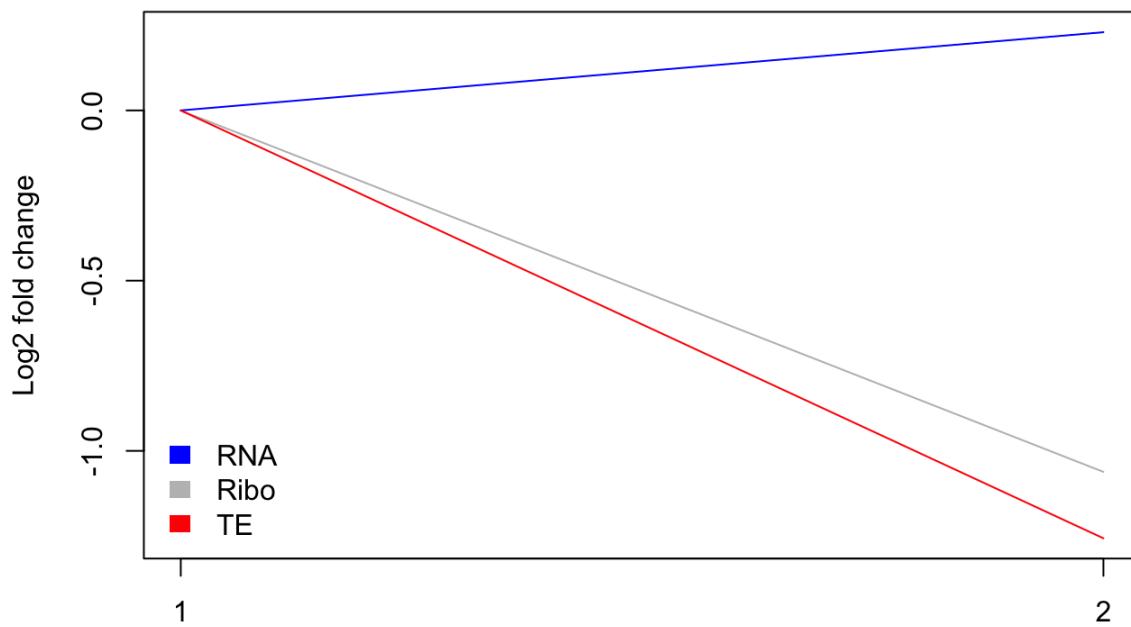
```
for (id in both){  
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)  
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)  
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),  
       ylab="Log2 fold change",xlab="",xaxt="n")  
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")  
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")  
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"  
)  
  axis(1,at=c(0,1),labels=c(1,2),las=1)  
  title(id)  
}
```

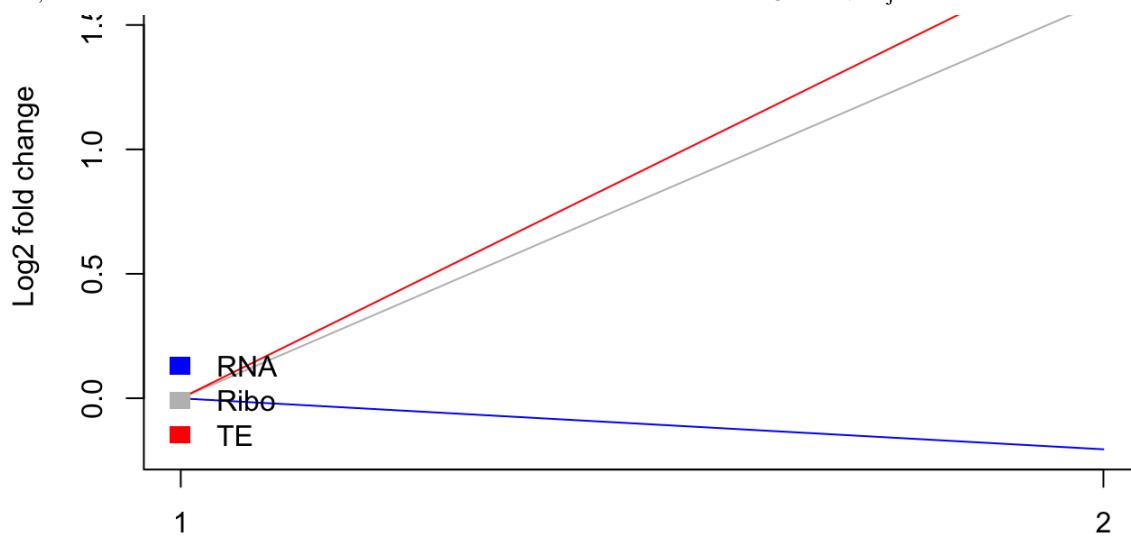
**AT1G51200****AT3G12780**

buffered: Genes regulated both by transcriptional and translational regulation (significant  $\Delta$ RNA,  $\Delta$ RPFs, and  $\Delta$ TE) include intensified and buffered genes. These genes are both DTGs and DTEGs.

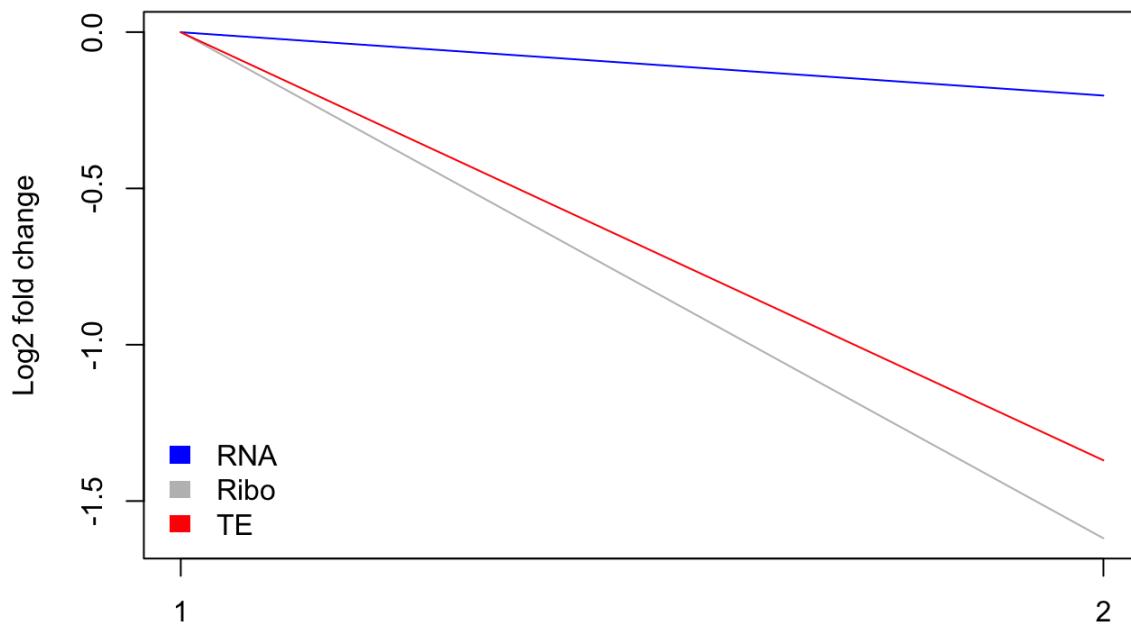
All lines going in the same direction  $\rightarrow$  change in translational efficiency is counteracting the change in RNA

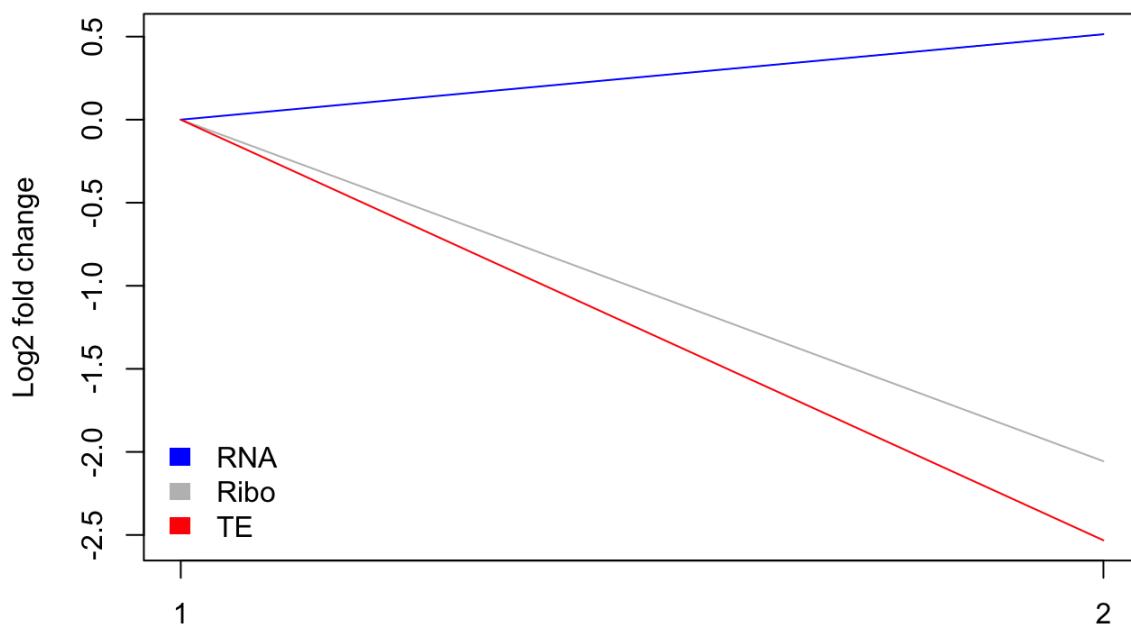
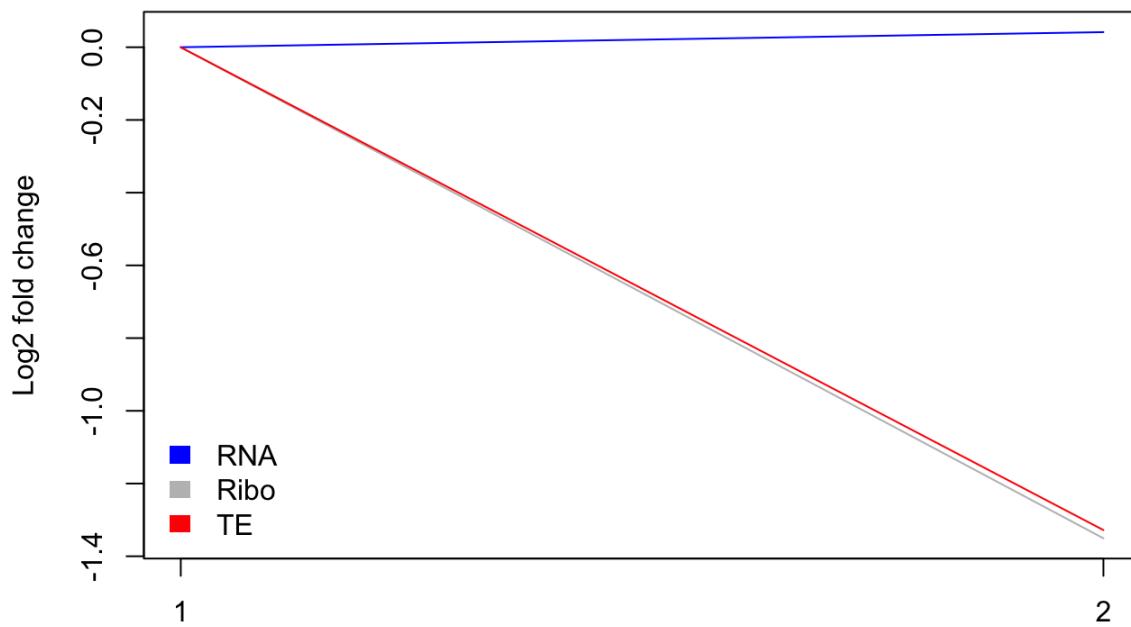
```
for (id in exclusive){  
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)  
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)  
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),  
       ylab="Log2 fold change",xlab="",xaxt="n")  
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")  
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")  
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"  
)  
  axis(1,at=c(0,1),labels=c(1,2),las=1)  
  title(id)  
}
```

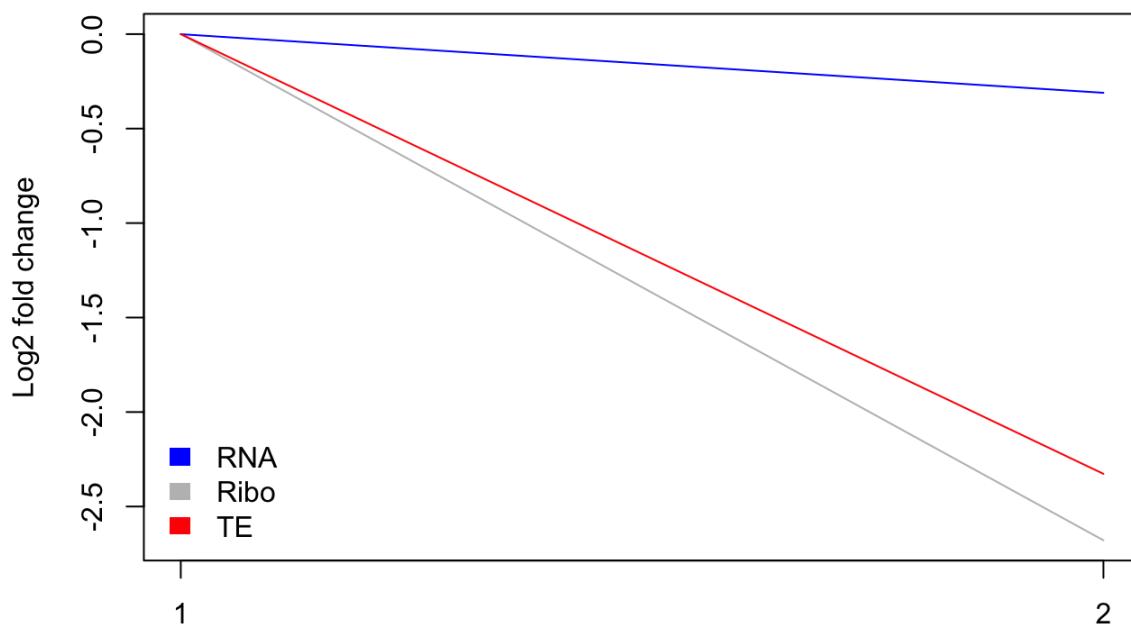
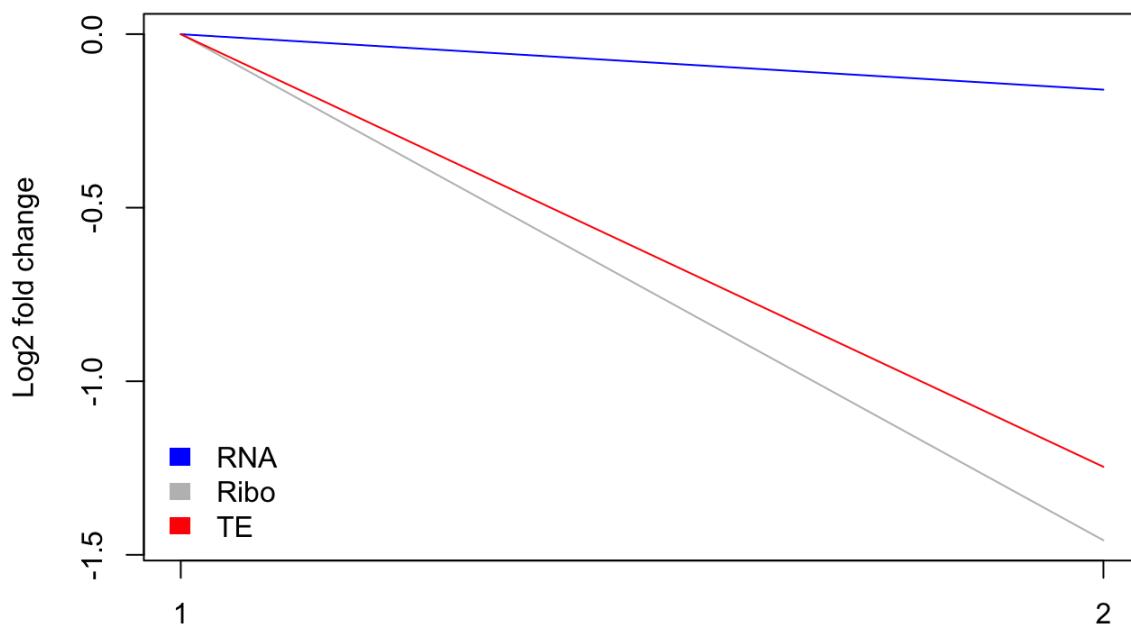
**AT1G12900****AT1G32060****AT1G55480**

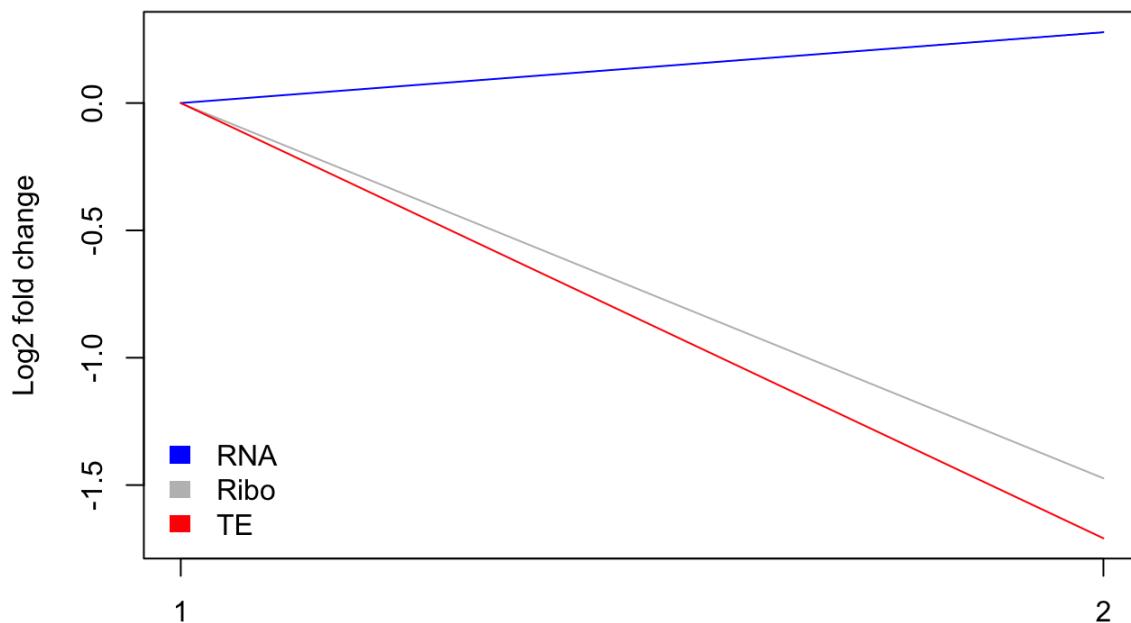
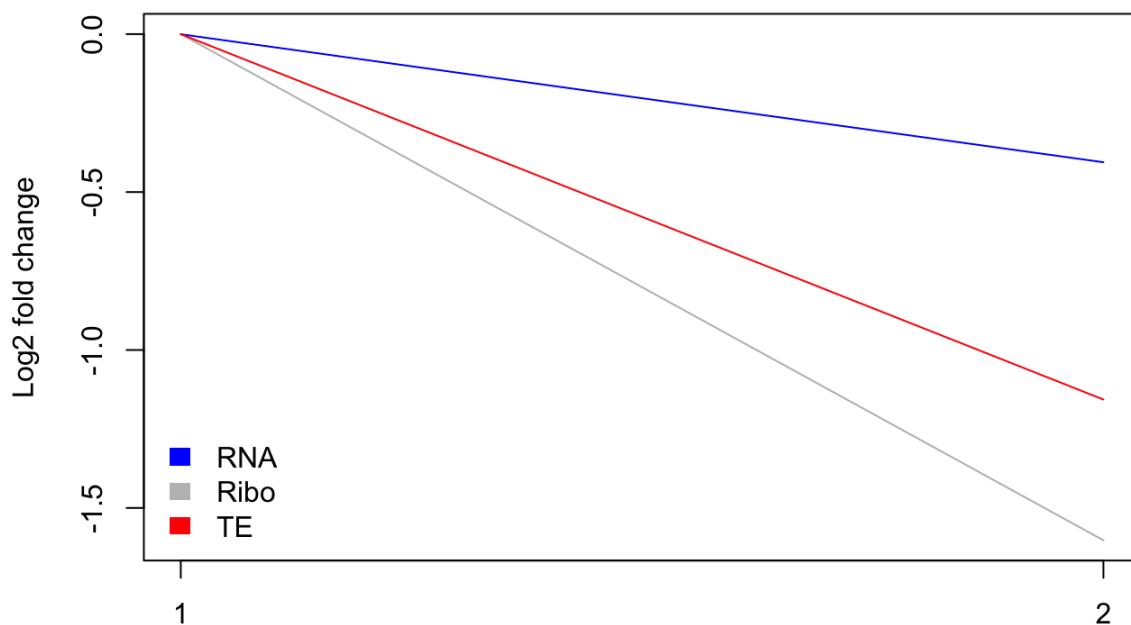


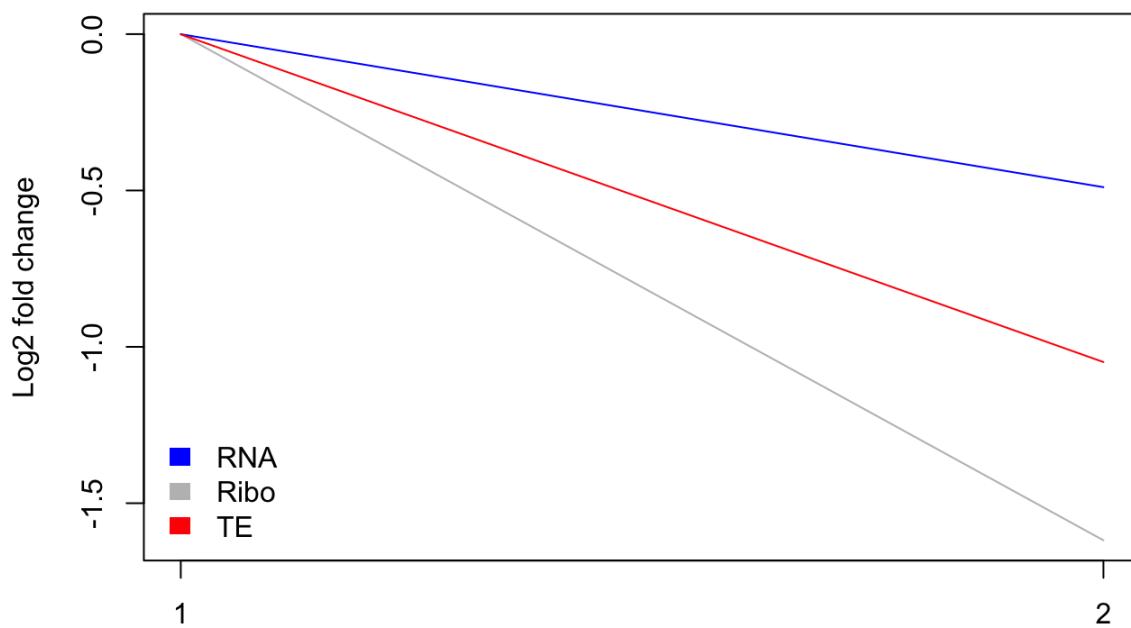
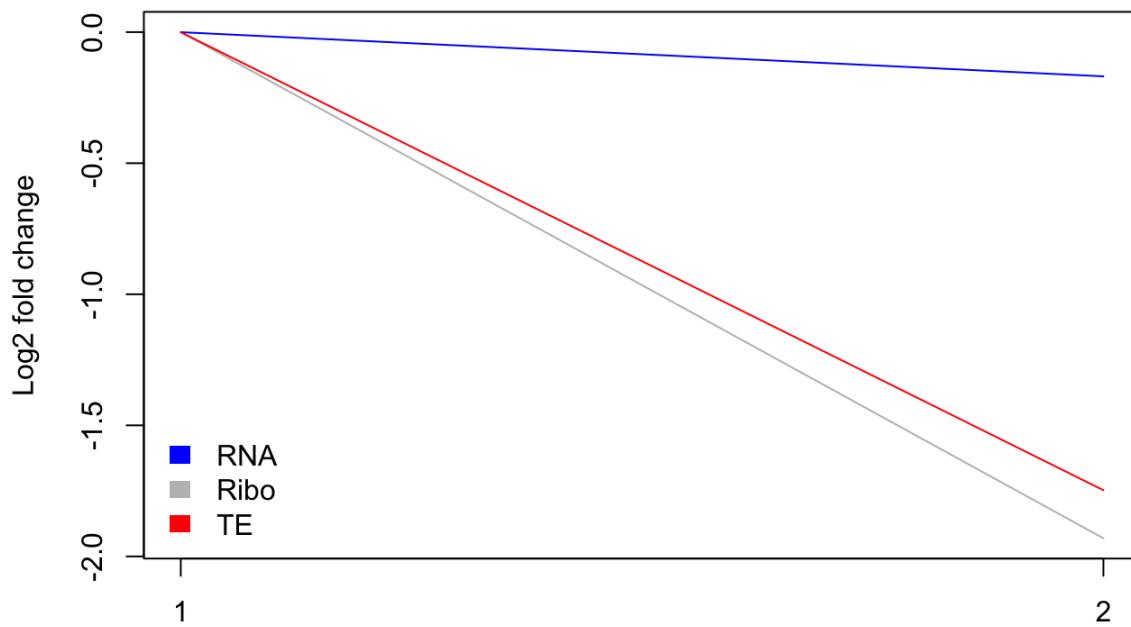
### AT1G77490

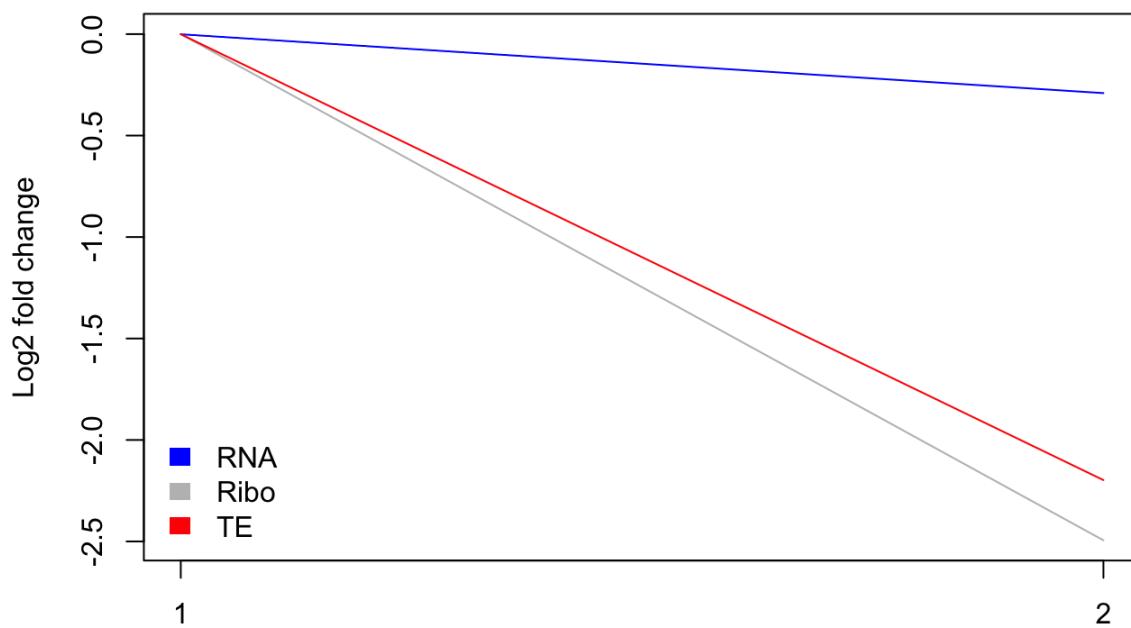
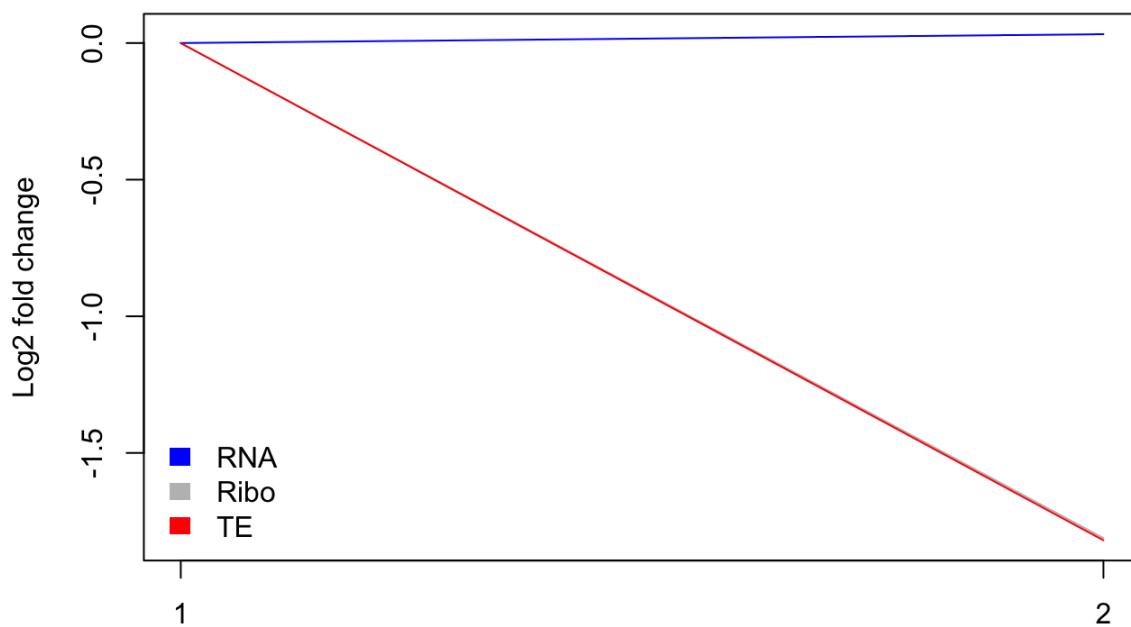


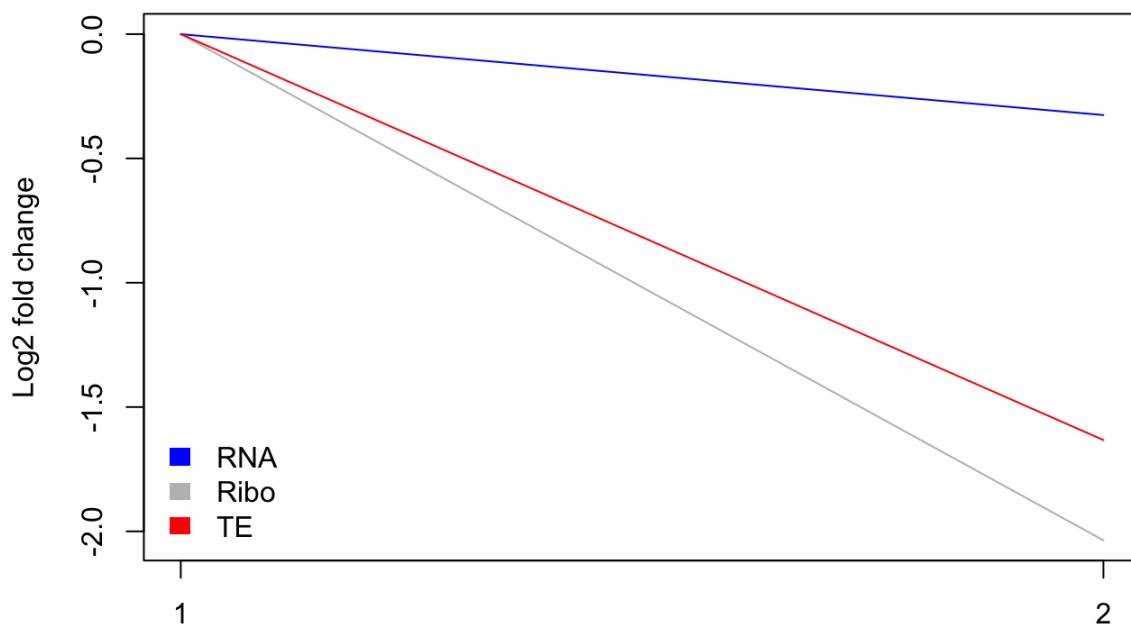
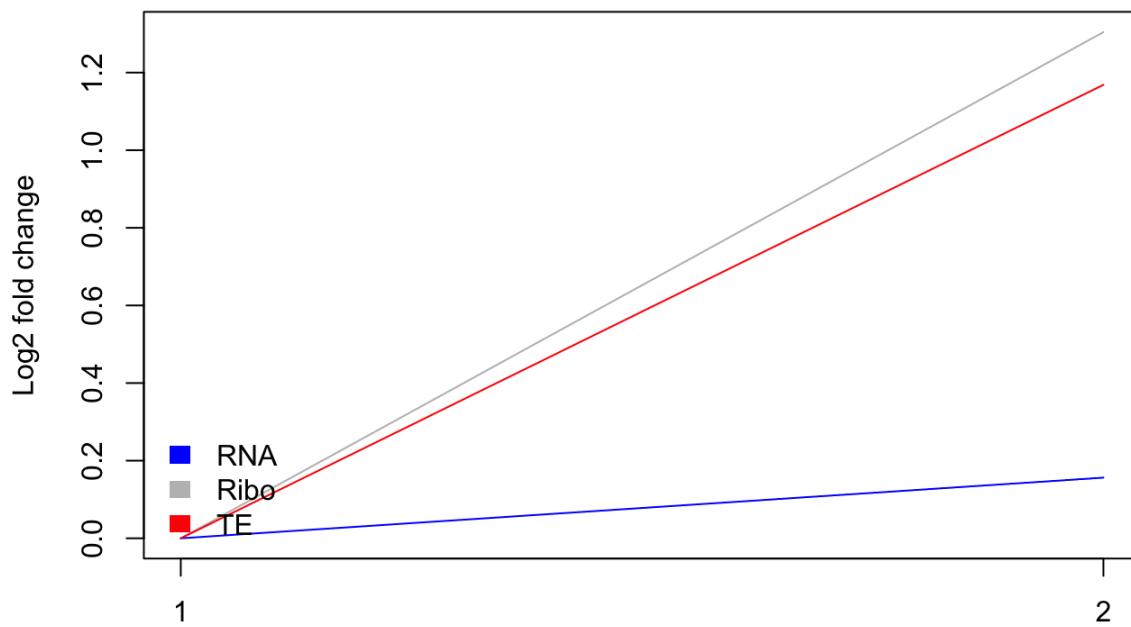
**AT2G33430****AT2G33800**

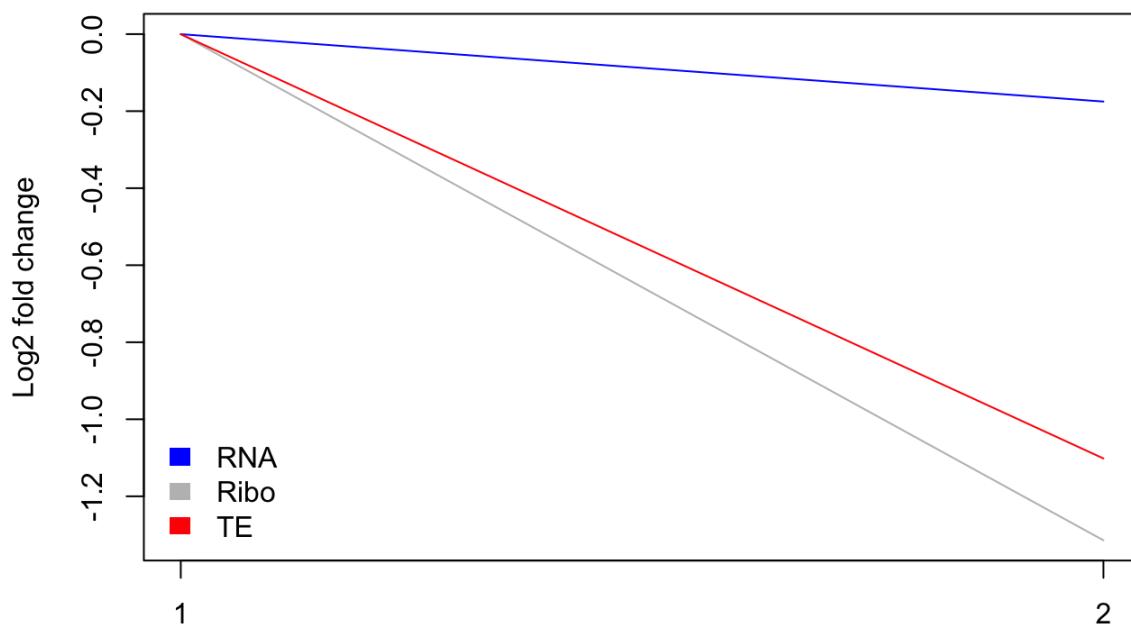
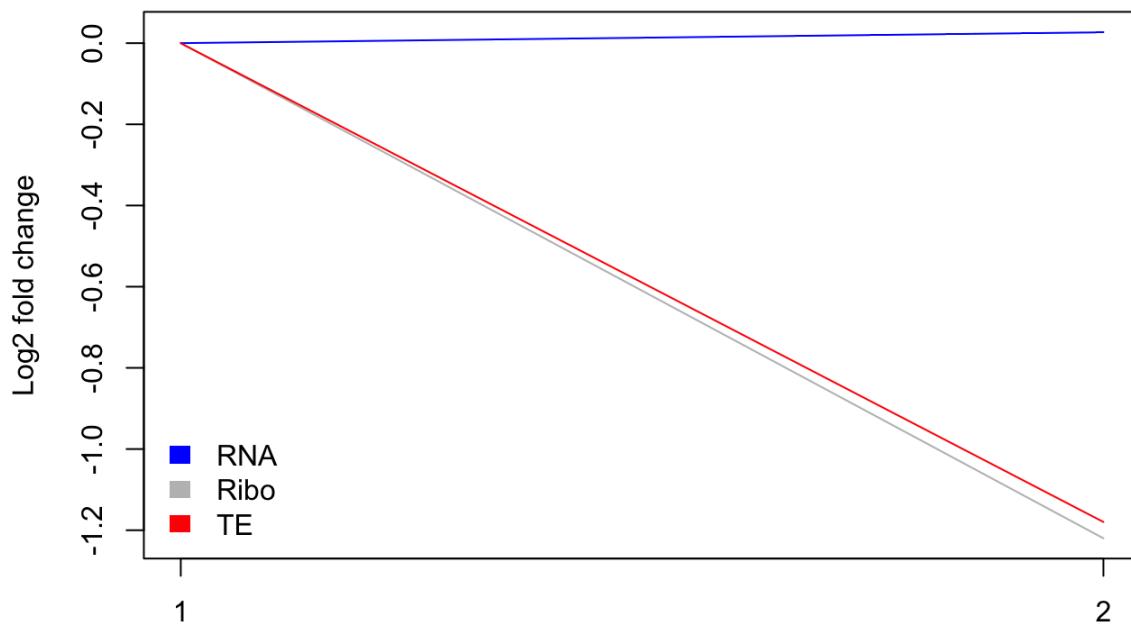
**AT2G36145****AT2G42690**

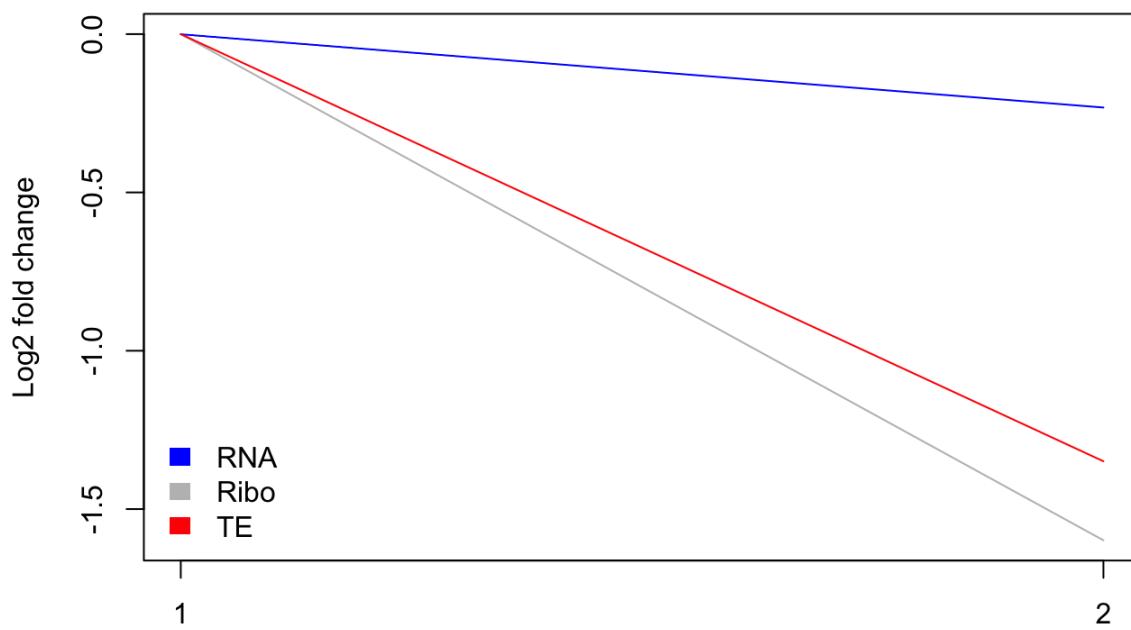
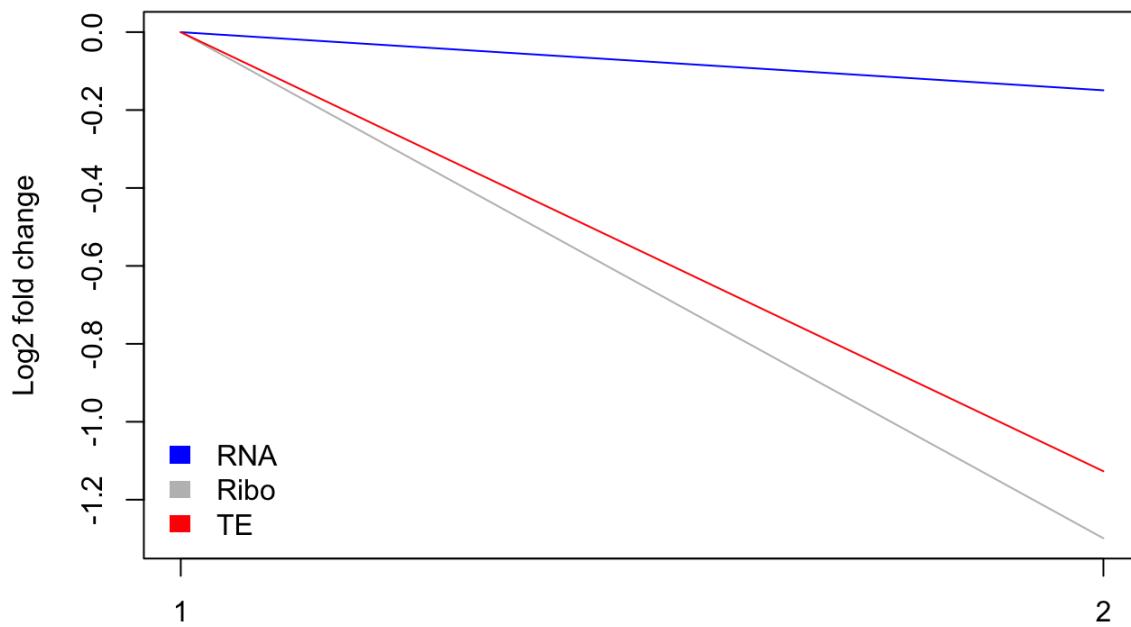
**AT2G43030****AT3G05350**

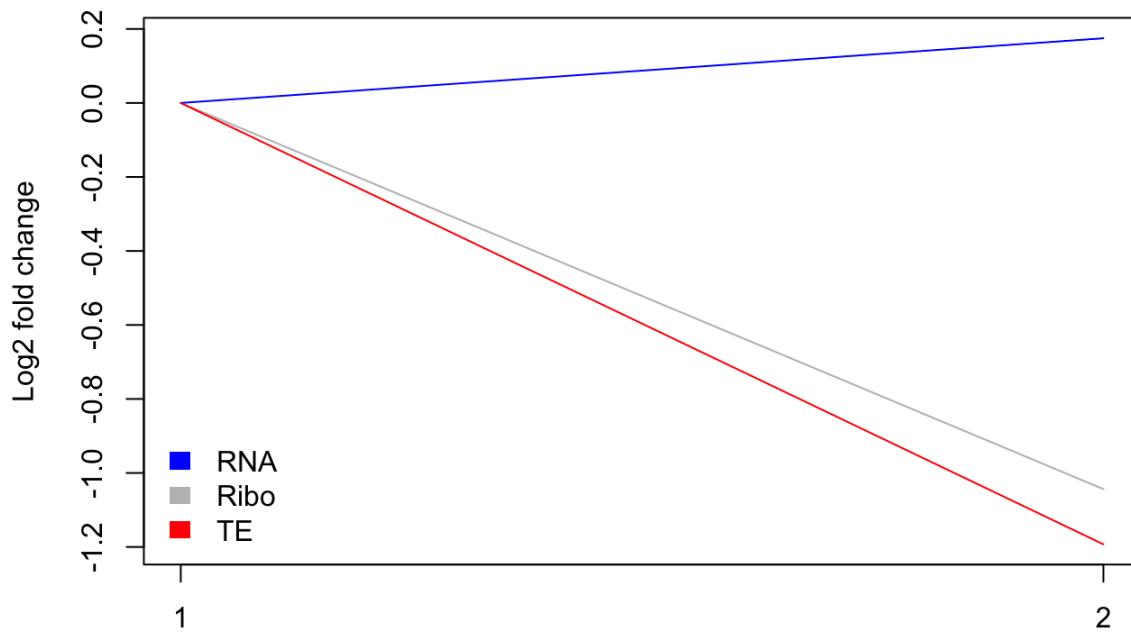
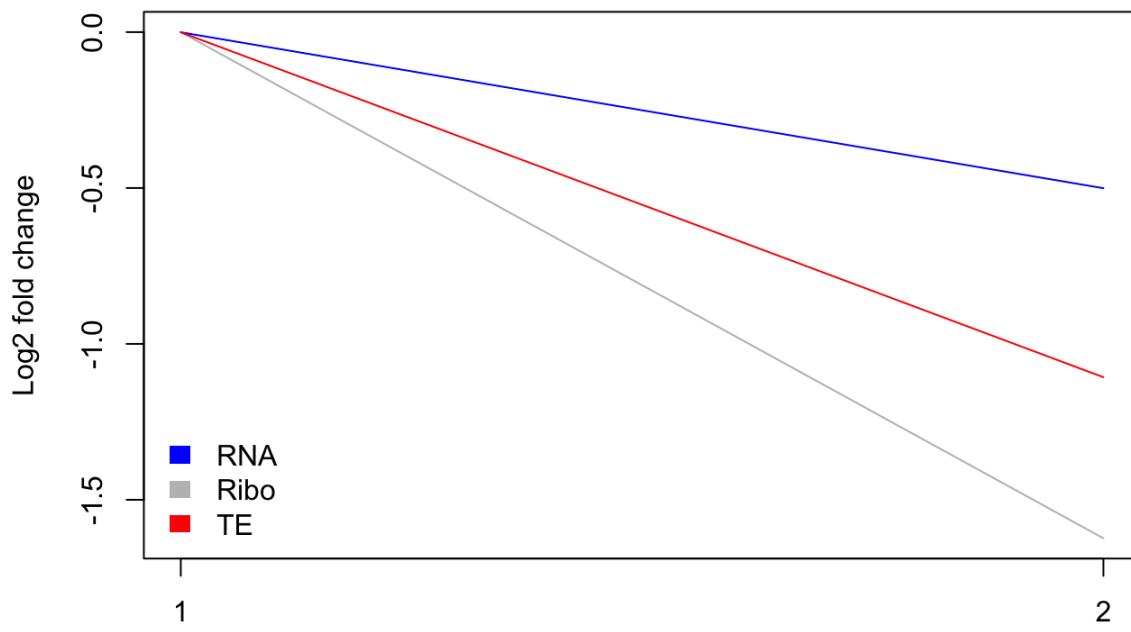
**AT3G15190****AT3G24430**

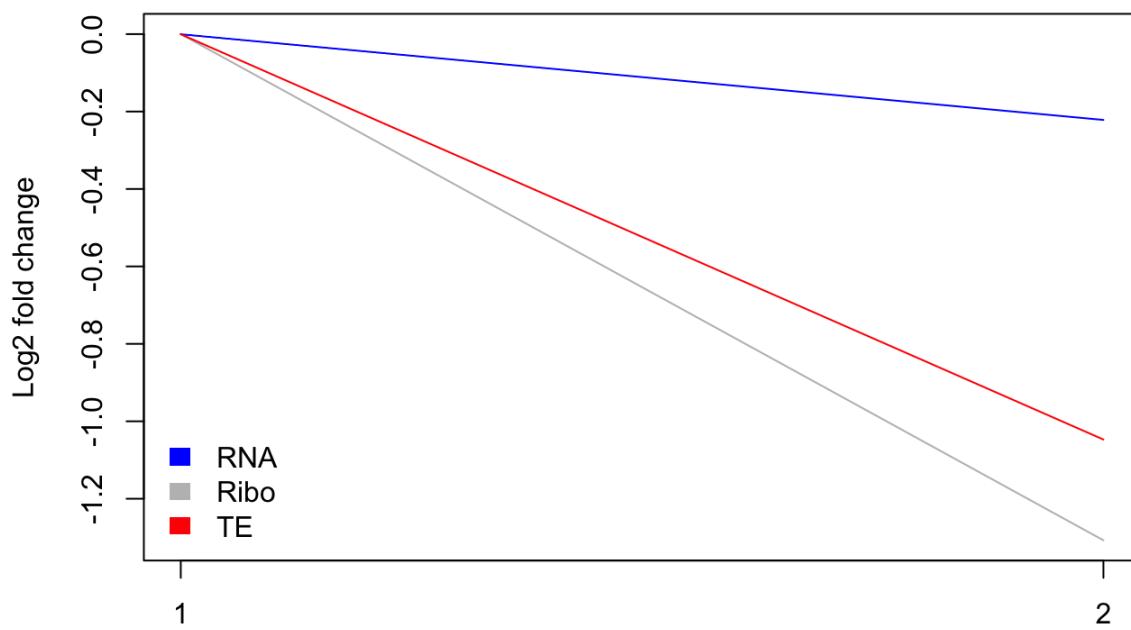
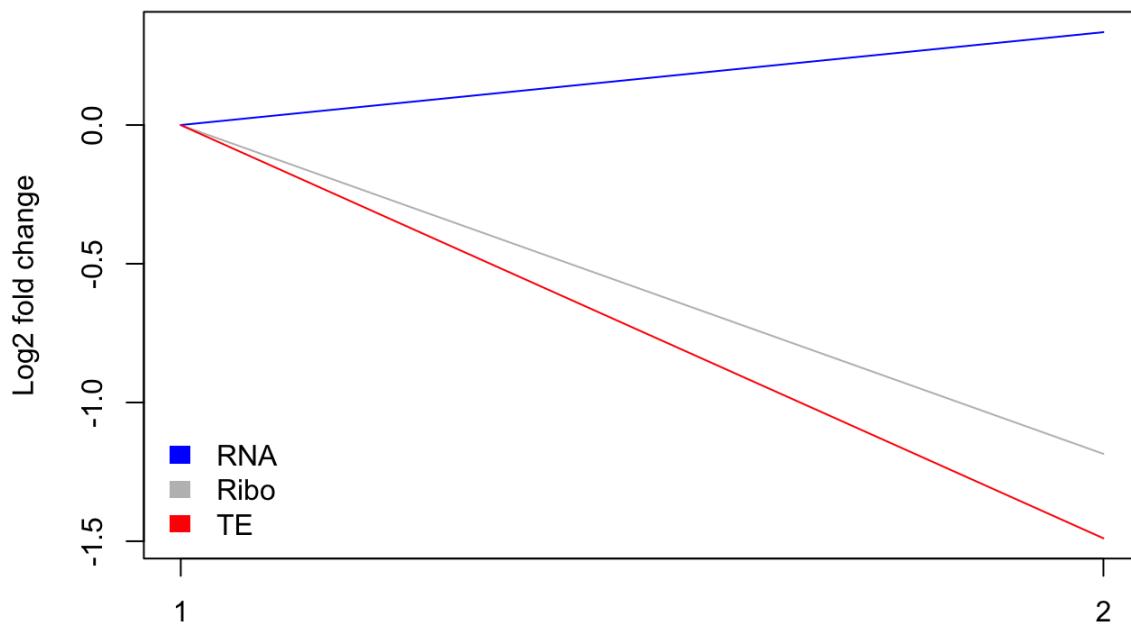
**AT3G27160****AT3G47470**

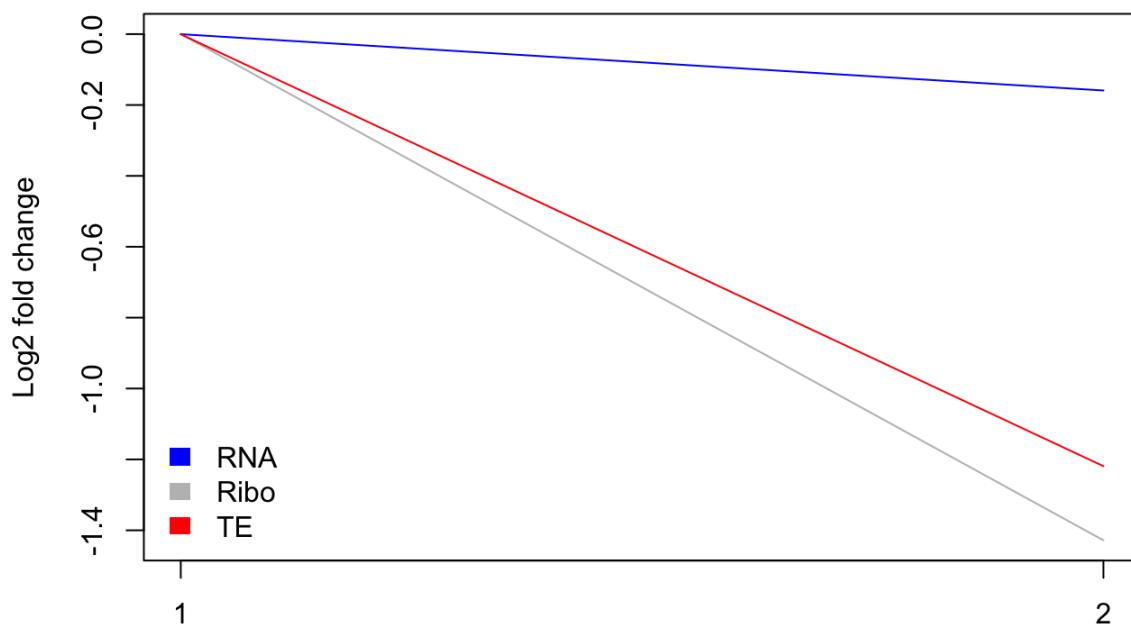
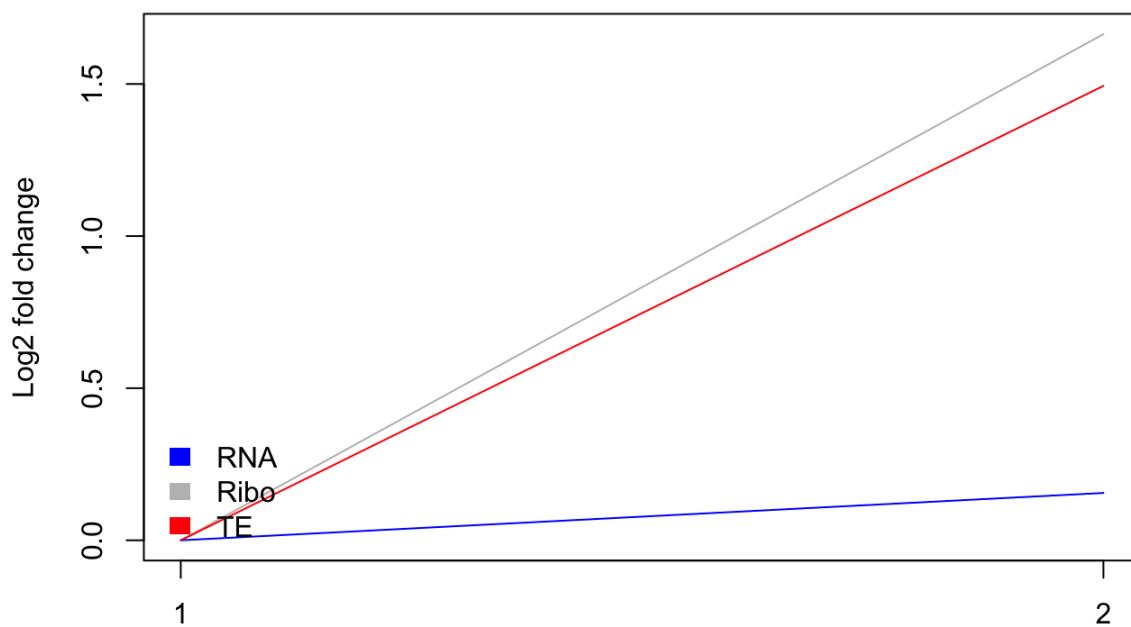
**AT3G50685****AT3G56240**

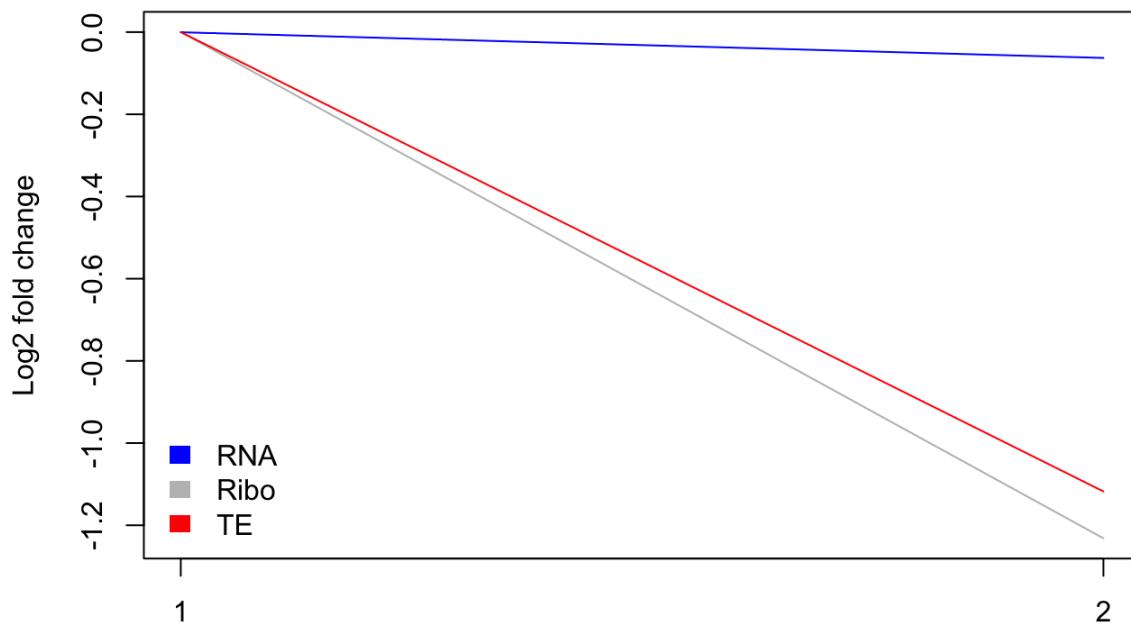
**AT3G60750****AT4G01310**

**AT4G25650****AT4G32260**

**AT5G01530****AT5G03880**

**AT5G05690****AT5G12250**

**AT5G23120****AT5G50960**

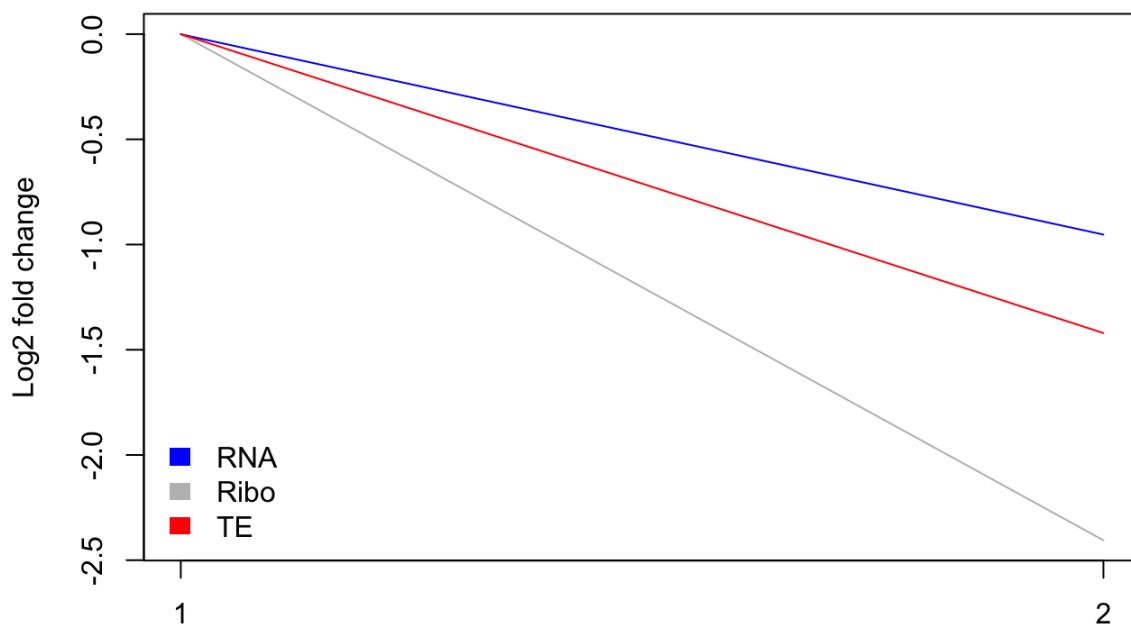
**AT5G58330**

exclusive focuses on findings that are translationally different only.

```

for (id in intensified){
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),
       ylab="Log2 fold change",xlab="",xaxt="n")
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"
  )
  axis(1,at=c(0,1),labels=c(1,2),las=1)
  title(id)
}

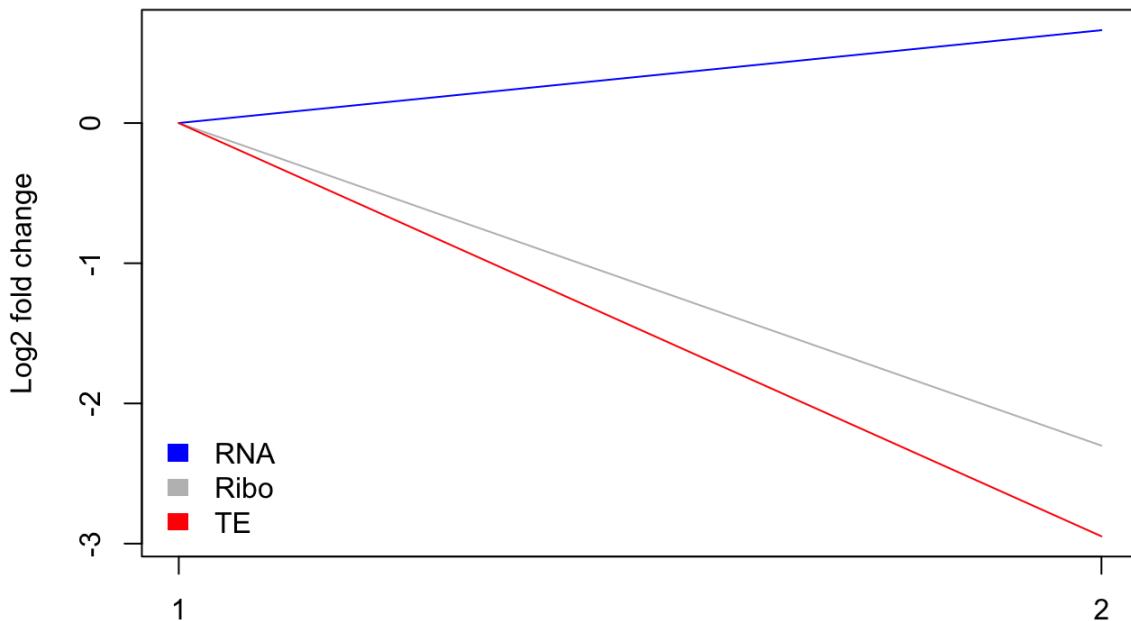
```

**AT3G12780**

```

for (id in buffered){
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),
       ylab="Log2 fold change",xlab="",xaxt="n")
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"
  )
  axis(1,at=c(0,1),labels=c(1,2),las=1)
  title(id)
}

```

**AT1G51200**

###Comparison genotypes Col and 14B, Contrast B

```

riboCountsB <- riboCounts %>% select(-contains("4G"))
rnaCountsB <- rnaCounts %>% select(-contains("4G"))
sampleAnnotationB <- sampleAnnotation %>% filter(!str_detect(group, '4G'))
sampleAnnotation2B <- sampleAnnotation2 %>% filter(!str_detect(group, '4G'))

# rna and ribo
sampleAnnotationB$SeqType = "RNA"
sampleAnnotation2B$SeqType = "Ribo"
combinedCountsB = cbind(riboCountsB, rnaCountsB)
sampleAnnotation3B = rbind(sampleAnnotationB, sampleAnnotation2B)
colnames(combinedCountsB) = rownames(sampleAnnotation3B)

# time + genotype + time:genotype + SeqType + SeqType:time + SeqType:genotype + SeqType:time:genotype
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = combinedCountsB,
  colData = sampleAnnotation3B,
  design = ~ time * genotype * SeqType
)

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors

```

```
DESeqDataSet_both = DESeq(  
  DESeqDataSet,  
  parallel=FALSE,  
  test = "LRT",  
  reduced = ~ time + genotype + SeqType)^2  
)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
DESeq_Results_both <- results(DESeqDataSet_both)  
clean_DESeq_padj <- which(!is.na(DESeq_Results_both$padj))  
sum(DESeq_Results_both[clean_DESeq_padj, "padj"] <= 0.1)
```

```
## [1] 145
```

```
sum(DESeq_Results_both[clean_DESeq_padj, "padj"] <= 0.1) * 0.1
```

```
## [1] 14.5
```

```
# rna  
DESeqDataSet = DESeqDataSetFromMatrix(  
  countData = rnaCountsB,  
  colData = sampleAnnotationB,  
  design = ~ time + genotype + time:genotype  
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in  
## design formula are characters, converting to factors
```

```
DESeqDataSet = DESeq(  
  DESeqDataSet,  
  parallel=FALSE,  
  test = "LRT",  
  reduced = ~ time + genotype  
)
```

```
## estimating size factors
```

```
## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing

DESeq_Results_RNA <- results(DESeqDataSet)

# ribo
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = riboCountsB,
  colData = sampleAnnotation2B,
  design = ~ time + genotype + time:genotype
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
DESeqDataSet = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ time + genotype
)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

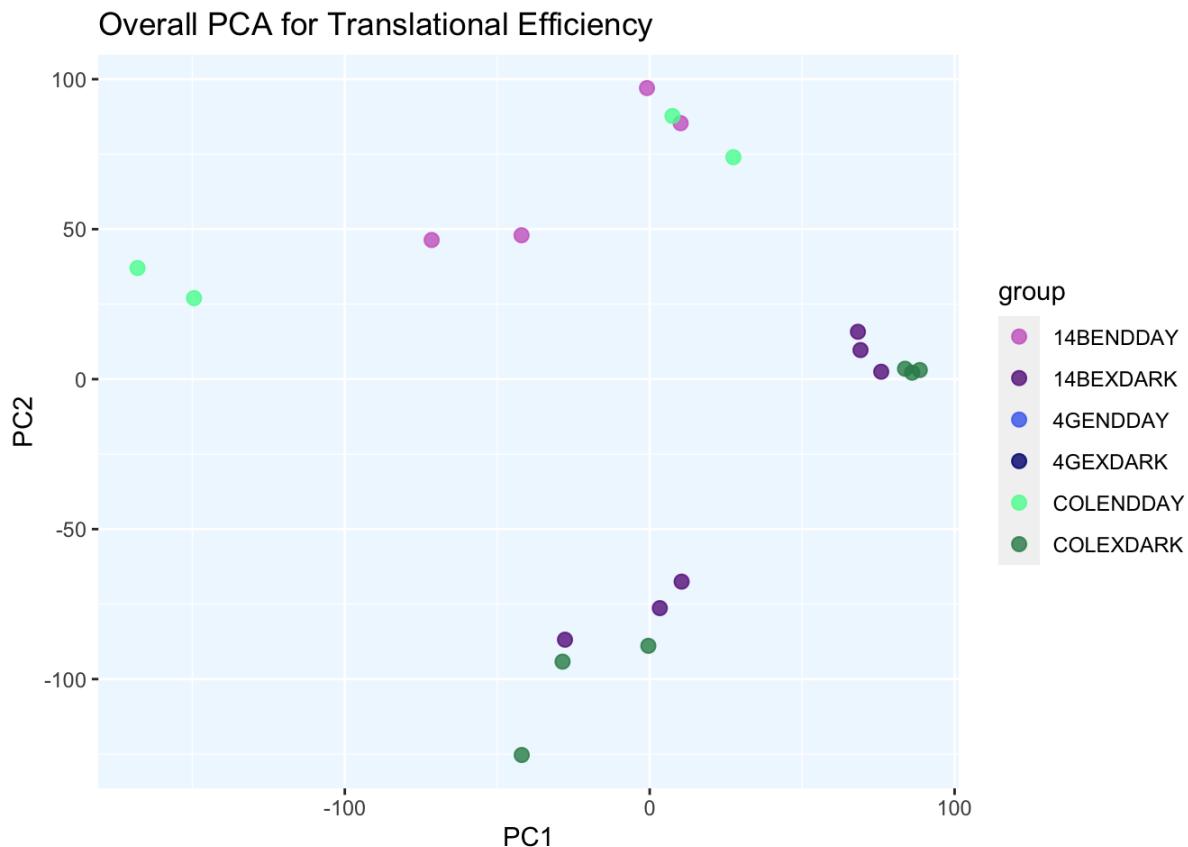
```
## fitting model and testing
```

```
DESeq_Results_ribo <- results(DESeqDataSet)
```

```
lgNorm = log2(counts(DESeqDataSet_both, normalized=TRUE) + 1)
```

## Overall PCA Plot

```
pca = prcomp(t(lgNorm))
pcaData = data.frame(pca$x[, 1:2])
pcaData$group = sampleAnnotation3B[rownames(pcaData), "group"]
pcaData$sample = rownames(pcaData)
gg = ggplot(pcaData, aes(x=PC1, y=PC2, color=group, label=sample))
gg = gg + geom_point(size=2.5, alpha=0.8)
gg = gg + scale_color_manual(values=groupColors)
gg = gg + theme(panel.background = element_rect(fill = 'aliceblue')) + ggtitle("Overall PCA for Translational Efficiency")
print(gg)
```



```
clean_DESeq_padj <- which(!is.na(DESeq_Results_both$padj))
clean_DESeq_Results <- DESeq_Results_both[clean_DESeq_padj, ]
significant_genes <- rownames(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1), ])
clean_DESeq_Results <- data.frame(clean_DESeq_Results[which(clean_DESeq_Results[, "padj"] <= 0.1),
  ])
clean_DESeq_Results$gene <- significant_genes
RNA_Result_groupings <- clean_DESeq_Results %>% inner_join(goAssociations2, by = "gene")
RNA_Result_groupings %>% group_by(gene_ontology_name) %>% summarize(count = n()) %>% arrange(desc(count)) %>%
  filter(count >= 5) %>%
  knitr::kable(caption = "Distribution of Significant Genes Groupings For TE")
```

Distribution of Significant Genes Groupings For TE

gene_ontology_name	count
pyruvate metabolic process	9
translation	9

gene_ontology_name	count
photosynthesis	8
response to light stimulus	8
protein-chromophore linkage	6
response to cold	6

```

exclusive = rownames(DESeq_Results_both)[which(DESeq_Results_both$padj < 0.1 & DESeq_Results_ribo$padj < 0.1 & DESeq_Results_RNA$padj > 0.1)]

both = rownames(DESeq_Results_both)[which(DESeq_Results_both$padj < 0.1 & DESeq_Results_ribo$padj < 0.1 & DESeq_Results_RNA$padj < 0.1)]

intensified = both[which(DESeq_Results_both[both,2]*DESeq_Results_RNA[both,2] > 0)]

buffered = both[which(DESeq_Results_both[both,2]*DESeq_Results_RNA[both,2] < 0)]

```

```

goAssociations2 %>% filter(gene %in% exclusive) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>%
  arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for Exclusive TE Results")

```

### Gene Groupings for Exclusive TE Results

gene_ontology_name	count
photosynthesis	7
protein-chromophore linkage	6
response to light stimulus	5
translation	4
response to cold	3
photosynthetic electron transport chain	2
proton motive force-driven ATP synthesis	2
transmembrane transport	2
cold acclimation	1
glucose metabolic process	1
negative regulation of catalytic activity	1
protein ubiquitination	1
pyruvate metabolic process	1
response to disaccharide	1
response to heat	1
response to ozone	1
tetrapyrrole metabolic process	1
thylakoid membrane organization	1

gene_ontology_name	count
valine biosynthetic process	1

```
goAssociations2 %>% filter(gene %in% both) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for 'Both' TE Results")
```

#### Gene Groupings for 'Both' TE Results

gene_ontology_name	count
translation	3
translational elongation	2
gluconeogenesis	1
pyruvate metabolic process	1

```
goAssociations2 %>% filter(gene %in% intensified) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for Intensified TE Results")
```

#### Gene Groupings for Intensified TE Results

gene_ontology_name	count
gluconeogenesis	1
pyruvate metabolic process	1
translation	1

```
goAssociations2 %>% filter(gene %in% buffered) %>% group_by(gene_ontology_name) %>% summarise(count = n()) %>% arrange(desc(count)) %>% knitr::kable(caption = "Gene Groupings for Buffered TE Results")
```

#### Gene Groupings for Buffered TE Results

gene_ontology_name	count
translation	2
translational elongation	2

```
goAssociations2 %>% filter(gene %in% exclusive) %>%
knitr::kable(caption = "Gene Groupings for Exclusive TE Results")
```

#### Gene Groupings for Exclusive TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0009631 (GO:0009631)	cold acclimation	AT4G24770
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT1G29930

gene_ontology_primary_id	gene_ontology_name	gene
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT1G61520
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT3G54890
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT3G61470
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT5G01530
GO:0018298 (GO:0018298)	protein-chromophore linkage	AT5G54270
GO:0034285 (GO:0034285)	response to disaccharide	AT2G47400
GO:0033013 (GO:0033013)	tetrapyrrole metabolic process	AT4G25080
GO:0006090 (GO:0006090)	pyruvate metabolic process	AT1G12900
GO:0010193 (GO:0010193)	response to ozone	AT4G25100
GO:0009416 (GO:0009416)	response to light stimulus	AT1G29930
GO:0009409 (GO:0009409)	response to cold	AT1G67090
GO:0015986 (GO:0015986)	proton motive force-driven ATP synthesis	AT4G32260
GO:0006412 (GO:0006412)	translation	AT1G07320
GO:0016567 (GO:0016567)	protein ubiquitination	AT5G22920
GO:0009767 (GO:0009767)	photosynthetic electron transport chain	AT4G03280
GO:0009408 (GO:0009408)	response to heat	AT5G02500
GO:0015979 (GO:0015979)	photosynthesis	AT1G06680
GO:0015979 (GO:0015979)	photosynthesis	AT5G38410
GO:0006006 (GO:0006006)	glucose metabolic process	AT1G12900
GO:0006412 (GO:0006412)	translation	AT2G43030
GO:0009416 (GO:0009416)	response to light stimulus	AT5G38420
GO:0009767 (GO:0009767)	photosynthetic electron transport chain	AT5G66190
GO:0015979 (GO:0015979)	photosynthesis	AT3G61470
GO:0055085 (GO:0055085)	transmembrane transport	AT2G39010
GO:0009416 (GO:0009416)	response to light stimulus	AT5G01530
GO:0043086 (GO:0043086)	negative regulation of catalytic activity	AT3G15360
GO:0010027 (GO:0010027)	thylakoid membrane organization	AT1G44575
GO:0015979 (GO:0015979)	photosynthesis	AT1G55670
GO:0009409 (GO:0009409)	response to cold	AT2G37220
GO:0015979 (GO:0015979)	photosynthesis	AT5G54270
GO:0055085 (GO:0055085)	transmembrane transport	AT2G39010
GO:0015986 (GO:0015986)	proton motive force-driven ATP synthesis	AT4G32260
GO:0009409 (GO:0009409)	response to cold	AT2G37220
GO:0006412 (GO:0006412)	translation	AT2G43030

gene_ontology_primary_id	gene_ontology_name	gene
GO:0009416 (GO:0009416)	response to light stimulus	AT5G01530
GO:0006412 (GO:0006412)	translation	AT1G07320
GO:0009099 (GO:0009099)	valine biosynthetic process	AT3G58610
GO:0015979 (GO:0015979)	photosynthesis	AT3G54890
GO:0015979 (GO:0015979)	photosynthesis	AT1G67090
GO:0009416 (GO:0009416)	response to light stimulus	AT1G29930

```
goAssociations2 %>% filter(gene %in% both) %>%
knitr::kable(caption = "Gene Groupings for 'Both' TE Results")
```

#### Gene Groupings for 'Both' TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0006090 (GO:0006090)	pyruvate metabolic process	AT3G12780
GO:0006094 (GO:0006094)	gluconeogenesis	AT3G12780
GO:0006412 (GO:0006412)	translation	AT3G27160
GO:0006412 (GO:0006412)	translation	AT2G41840
GO:0006414 (GO:0006414)	translational elongation	AT5G19510
GO:0006412 (GO:0006412)	translation	AT2G41840
GO:0006414 (GO:0006414)	translational elongation	AT5G19510

```
goAssociations2 %>% filter(gene %in% intensified) %>%
knitr::kable(caption = "Gene Groupings for Intensified TE Results")
```

#### Gene Groupings for Intensified TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0006090 (GO:0006090)	pyruvate metabolic process	AT3G12780
GO:0006094 (GO:0006094)	gluconeogenesis	AT3G12780
GO:0006412 (GO:0006412)	translation	AT3G27160

```
goAssociations2 %>% filter(gene %in% buffered) %>%
knitr::kable(caption = "Gene Groupings for Buffered TE Results")
```

#### Gene Groupings for Buffered TE Results

gene_ontology_primary_id	gene_ontology_name	gene
GO:0006412 (GO:0006412)	translation	AT2G41840
GO:0006414 (GO:0006414)	translational elongation	AT5G19510
GO:0006412 (GO:0006412)	translation	AT2G41840

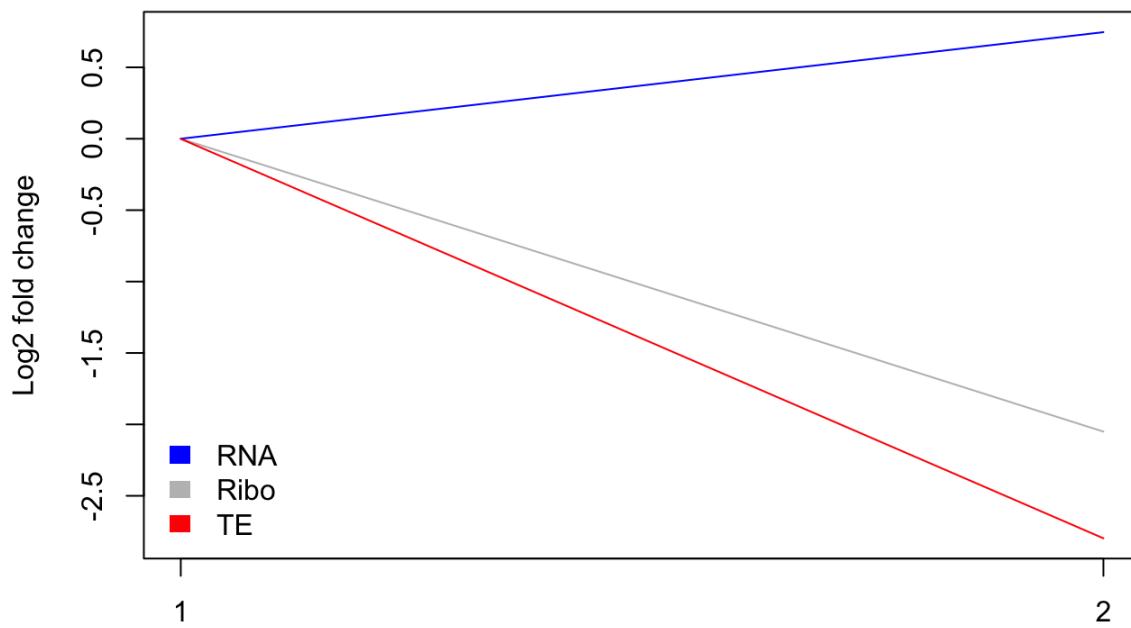
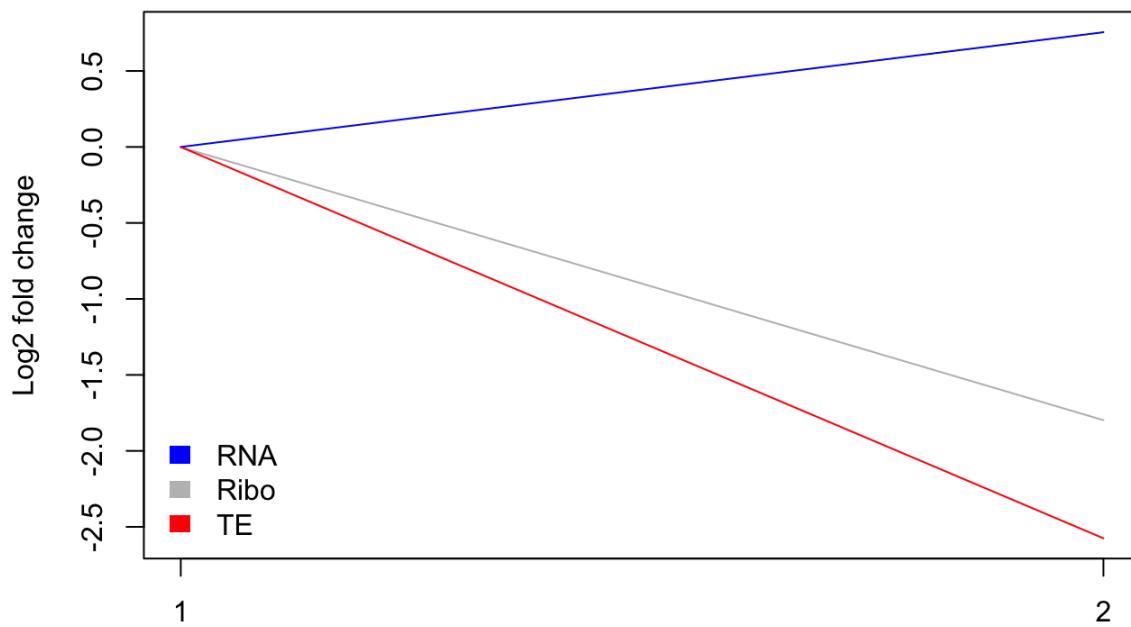
gene_ontology_primary_id	gene_ontology_name	gene
GO:0006414 (GO:0006414)	translational elongation	AT5G19510

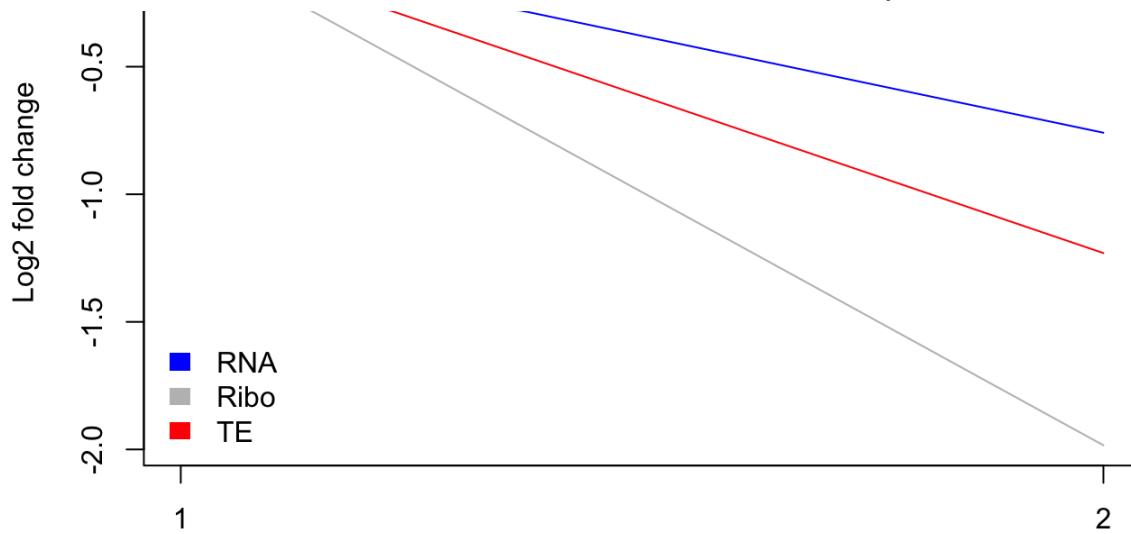
```

for (id in both){

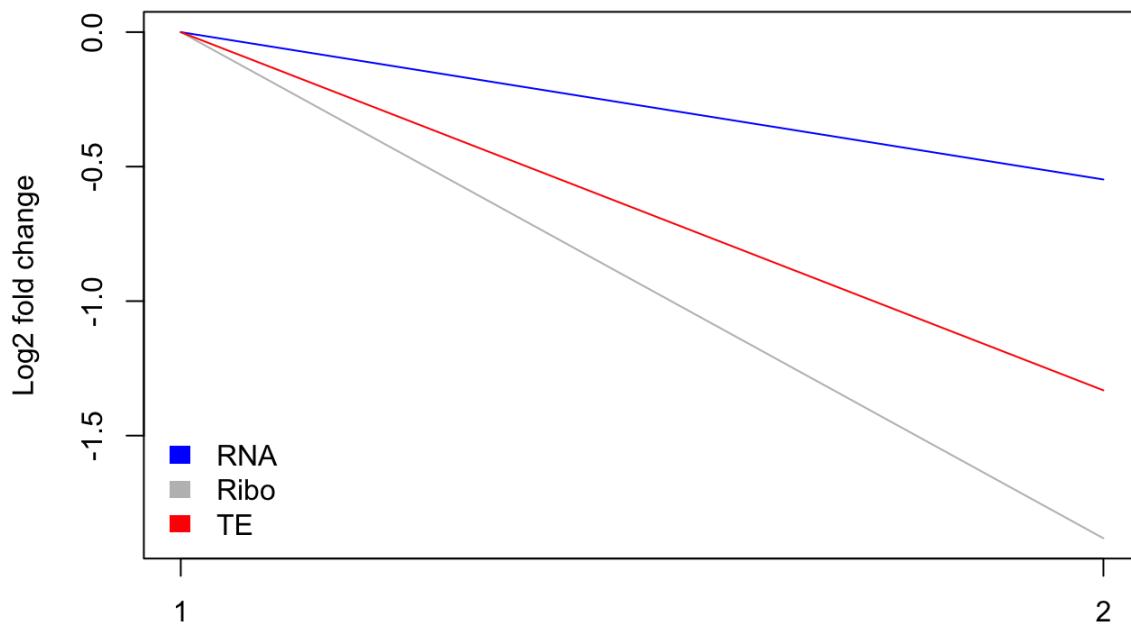
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),
       ylab="Log2 fold change",xlab="",xaxt="n")
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"
)
  axis(1,at=c(0,1),labels=c(1,2),las=1)
  title(id)
}

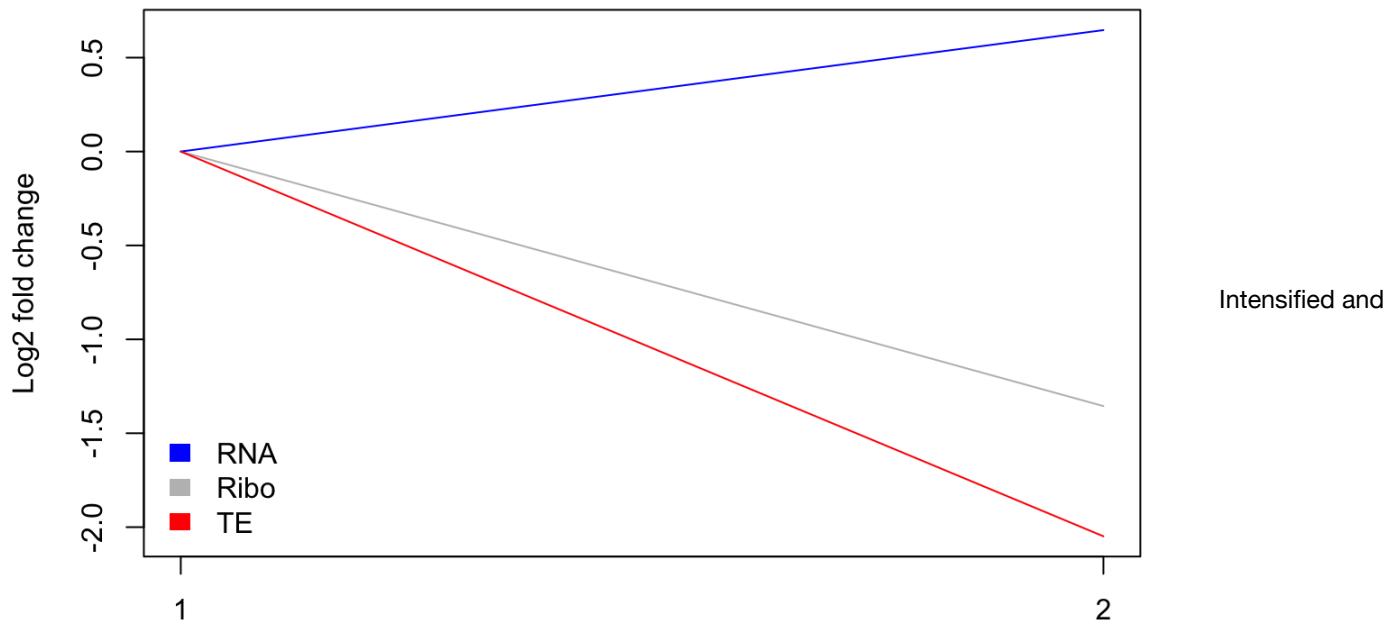
```

**AT1G29910****AT2G41840****AT3G12780**



### AT3G27160



**AT5G19510**

buffered: Genes regulated both by transcriptional and translational regulation (significant  $\Delta$ RNA,  $\Delta$ RPFs, and  $\Delta$ TE) include intensified and buffered genes. These genes are both DTGs and DTEGs.

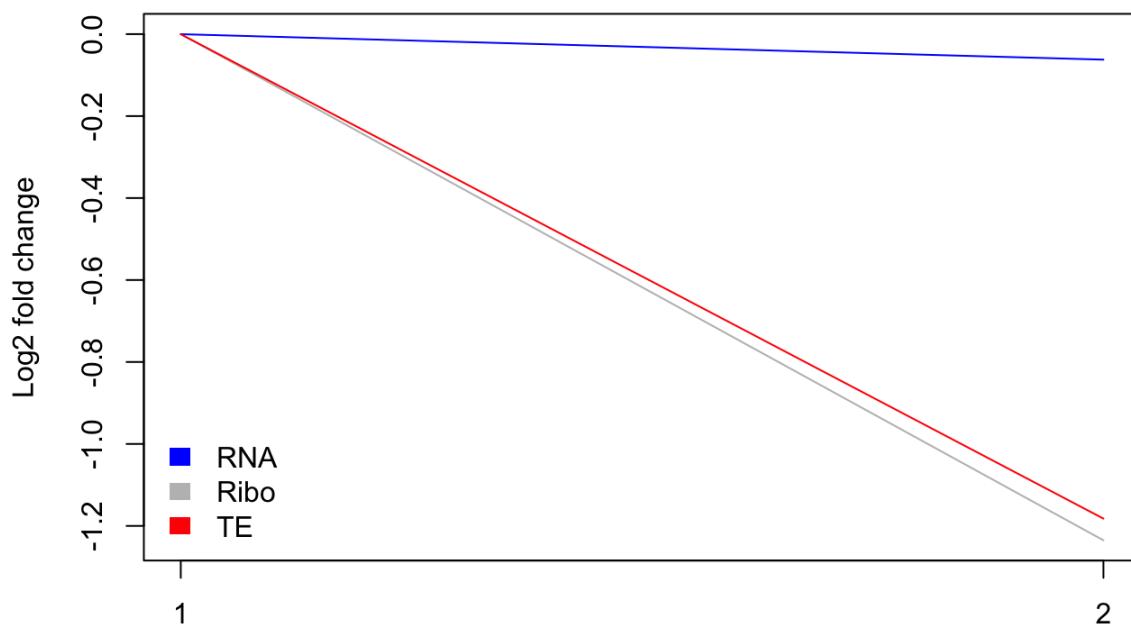
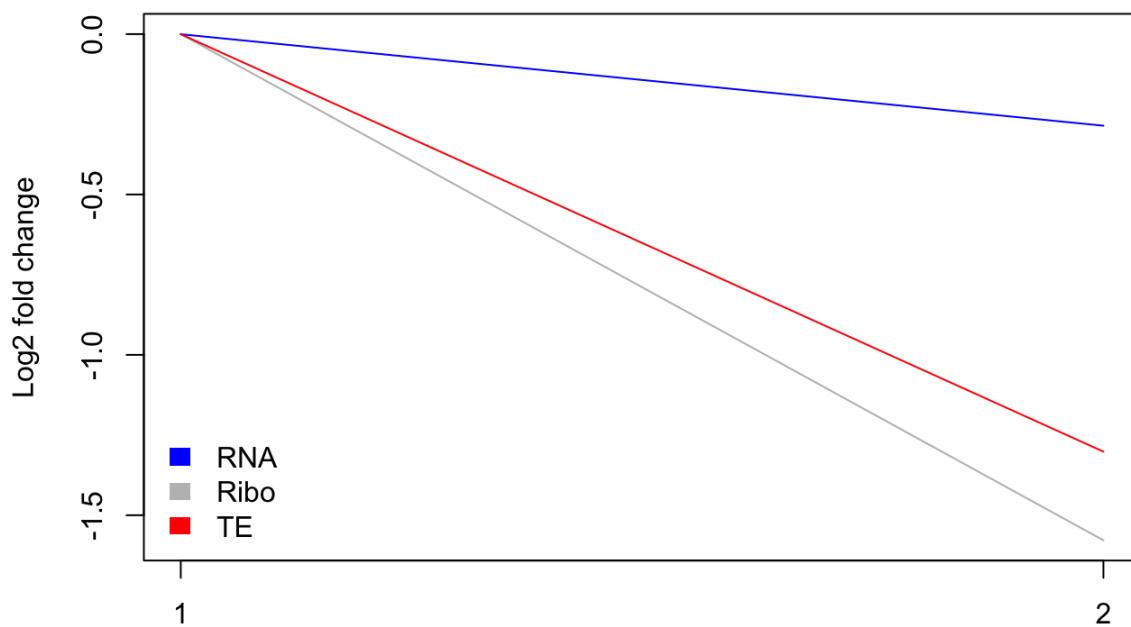
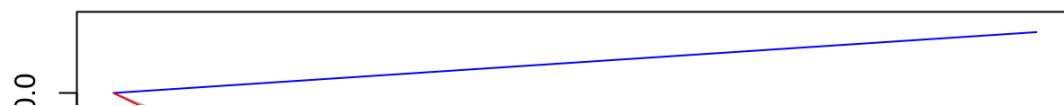
All lines going in the same direction  $\rightarrow$  change in translational efficiency is counteracting the change in RNA

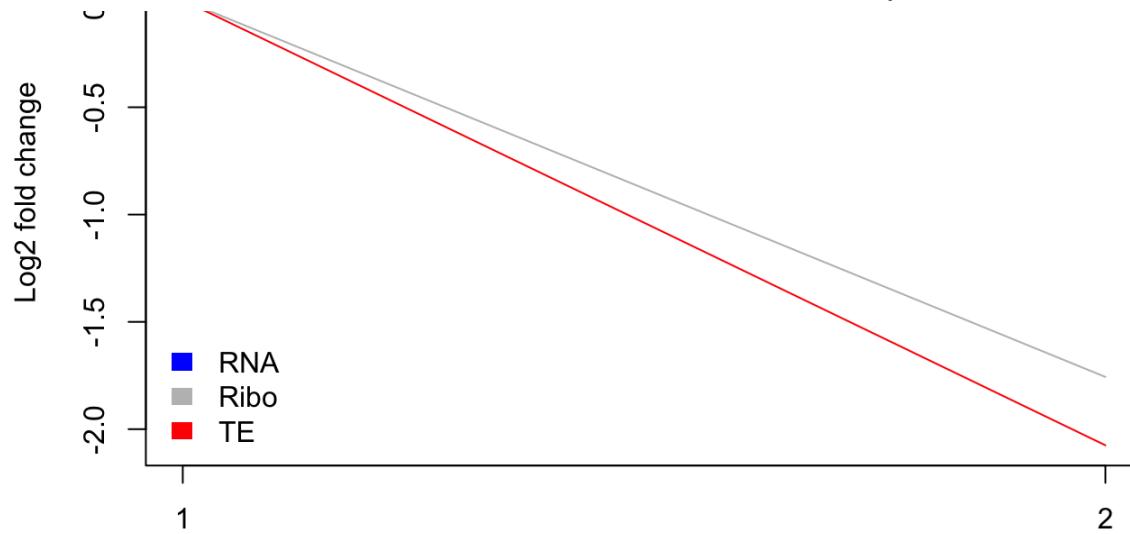
```

for (id in exclusive){
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),
       ylab="Log2 fold change",xlab="",xaxt="n")
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n")
}
axis(1,at=c(0,1),labels=c(1,2),las=1)
title(id)
}

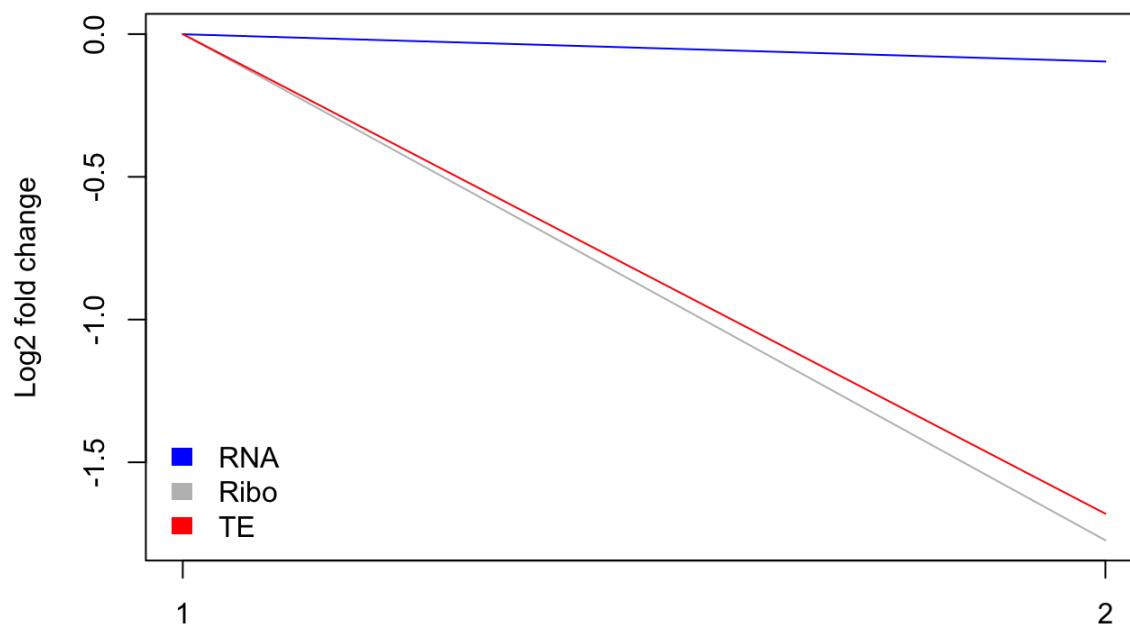
```

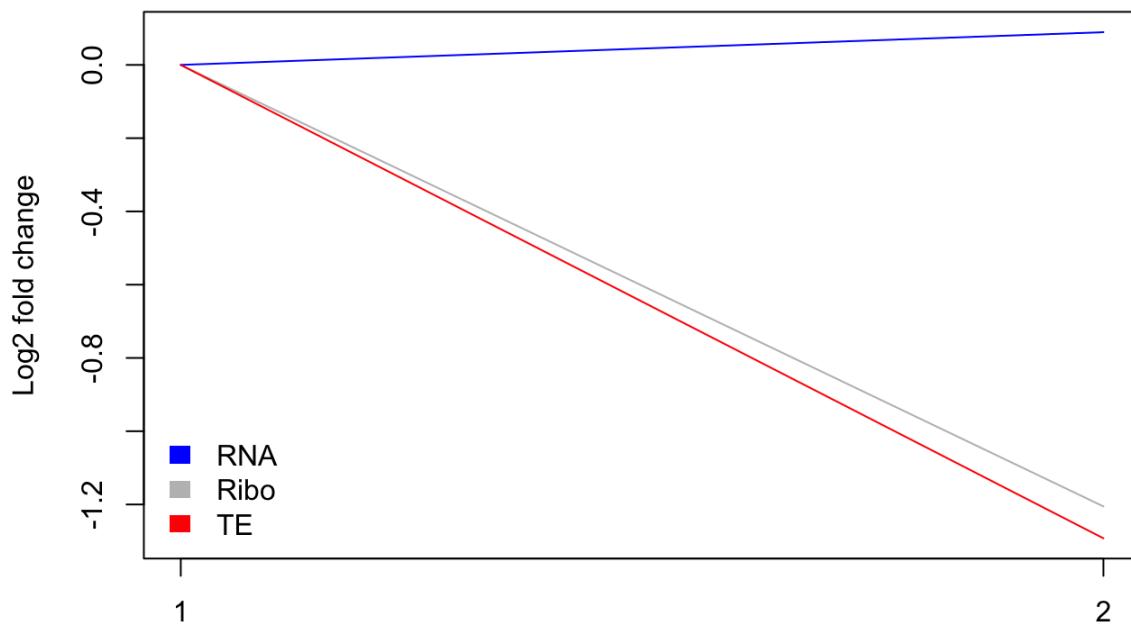
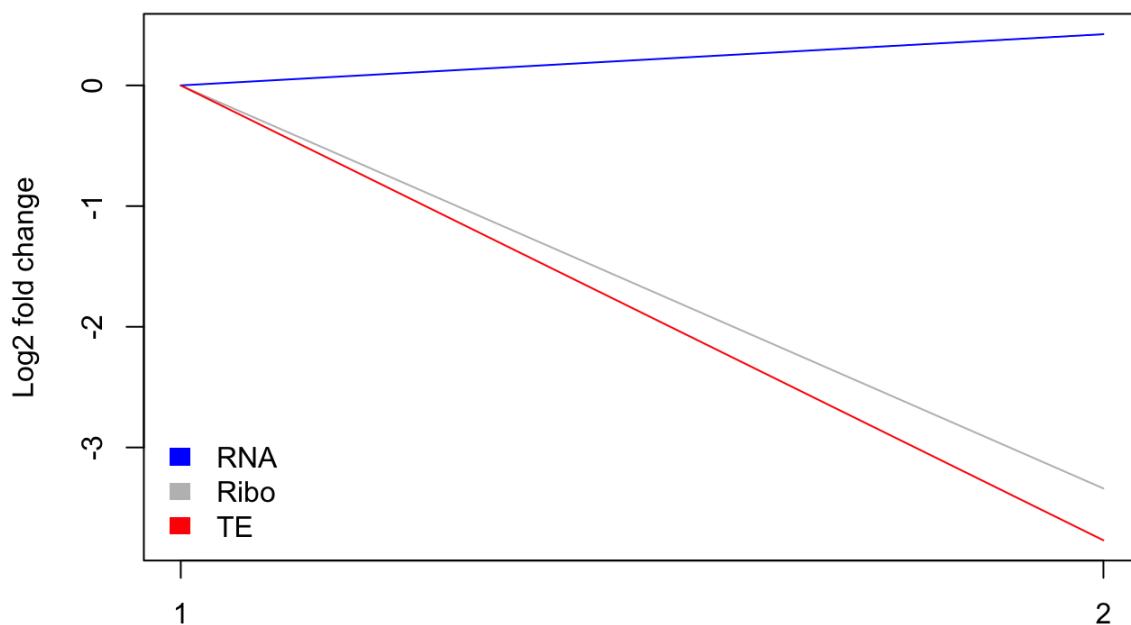


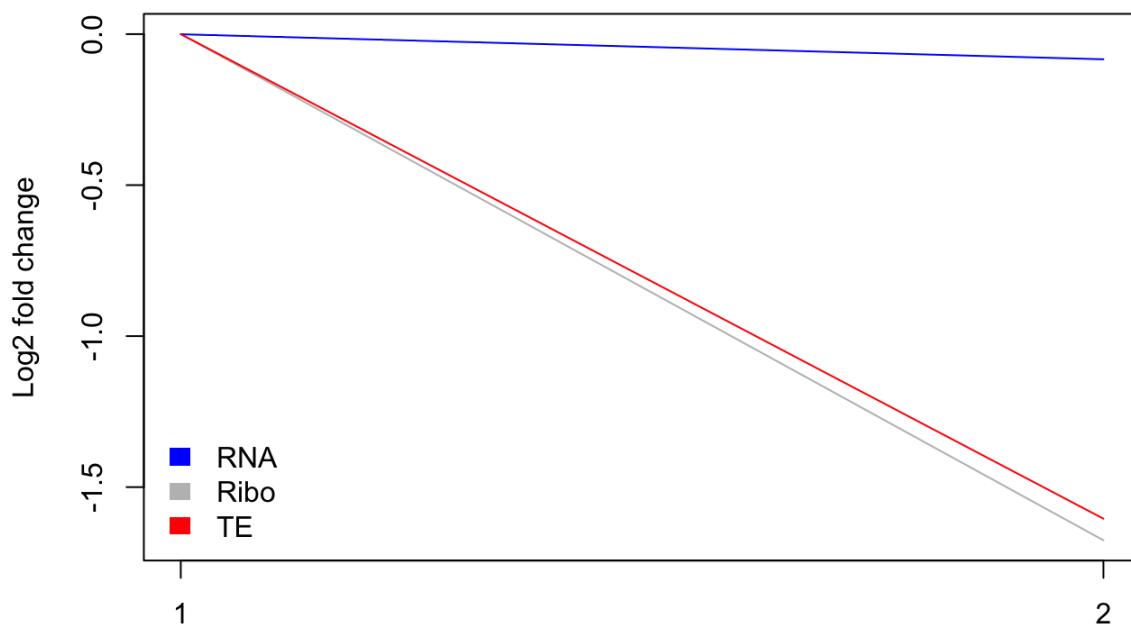
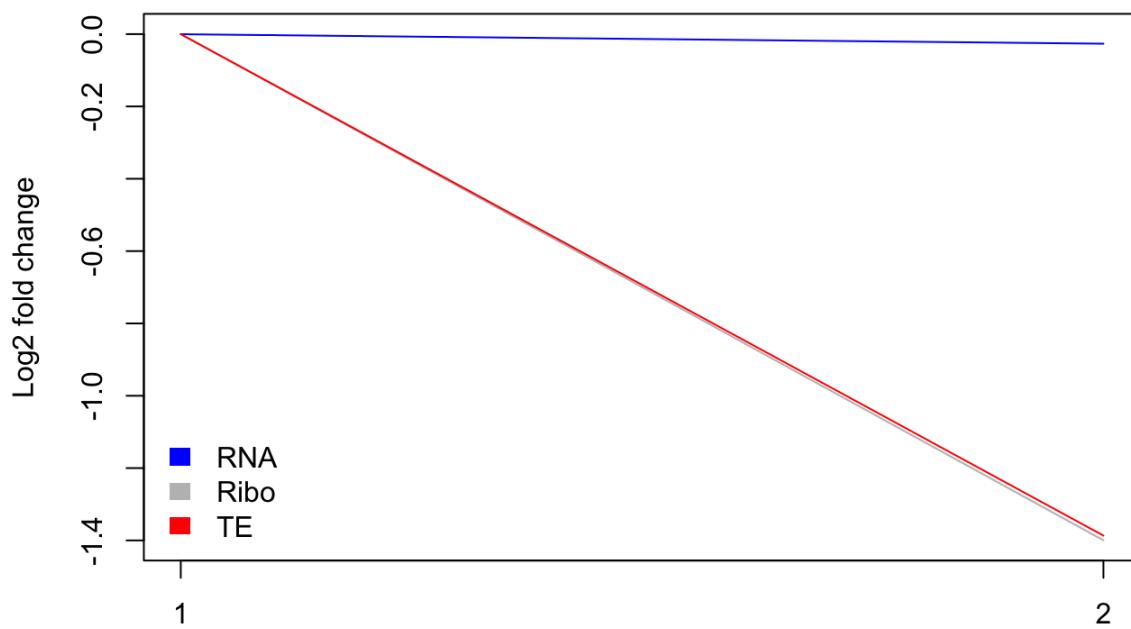
**AT1G03600****AT1G06680****AT1G07320**

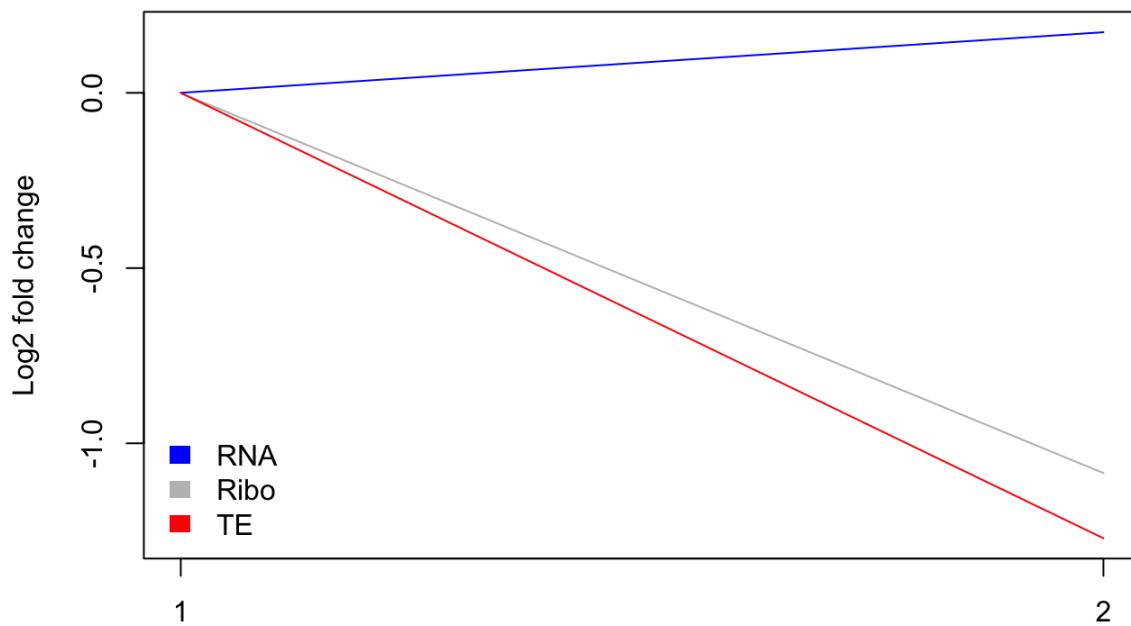
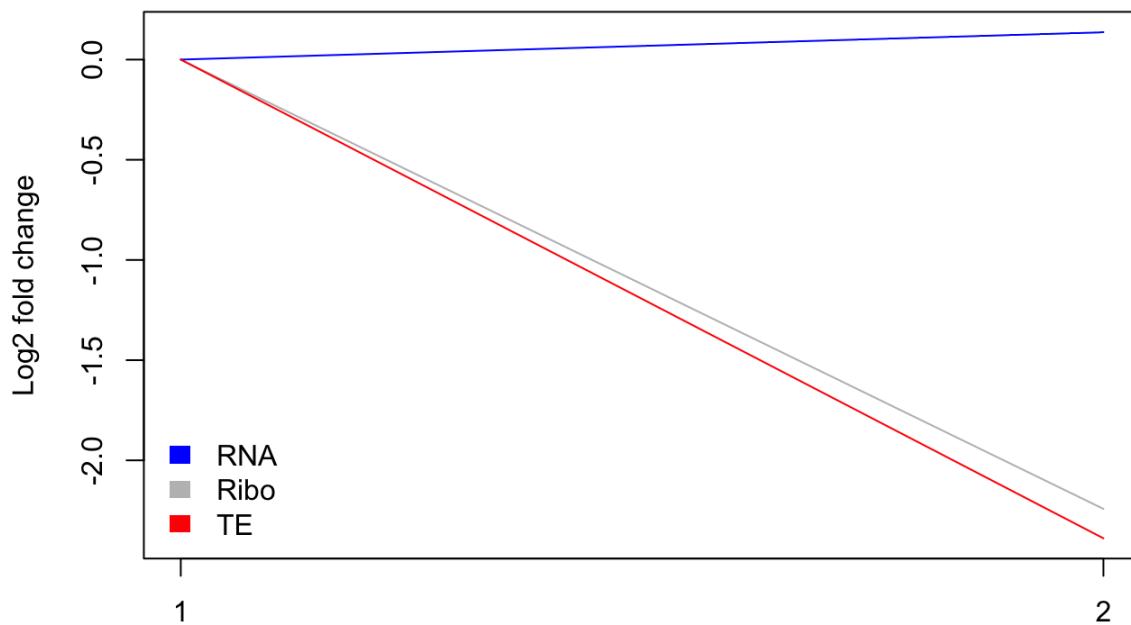


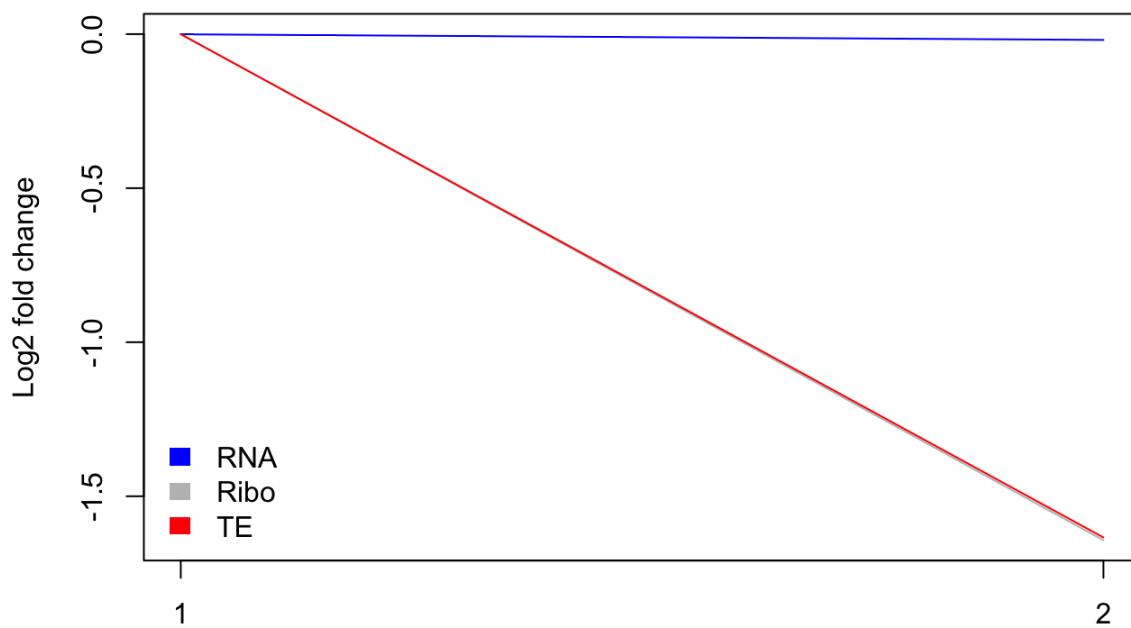
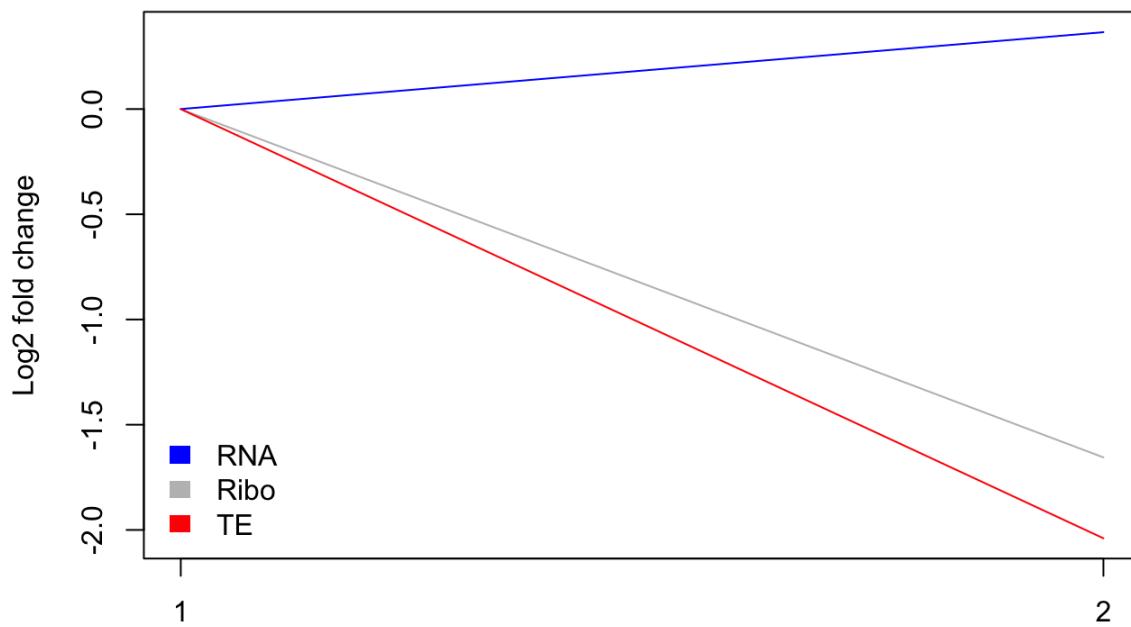
### AT1G12900

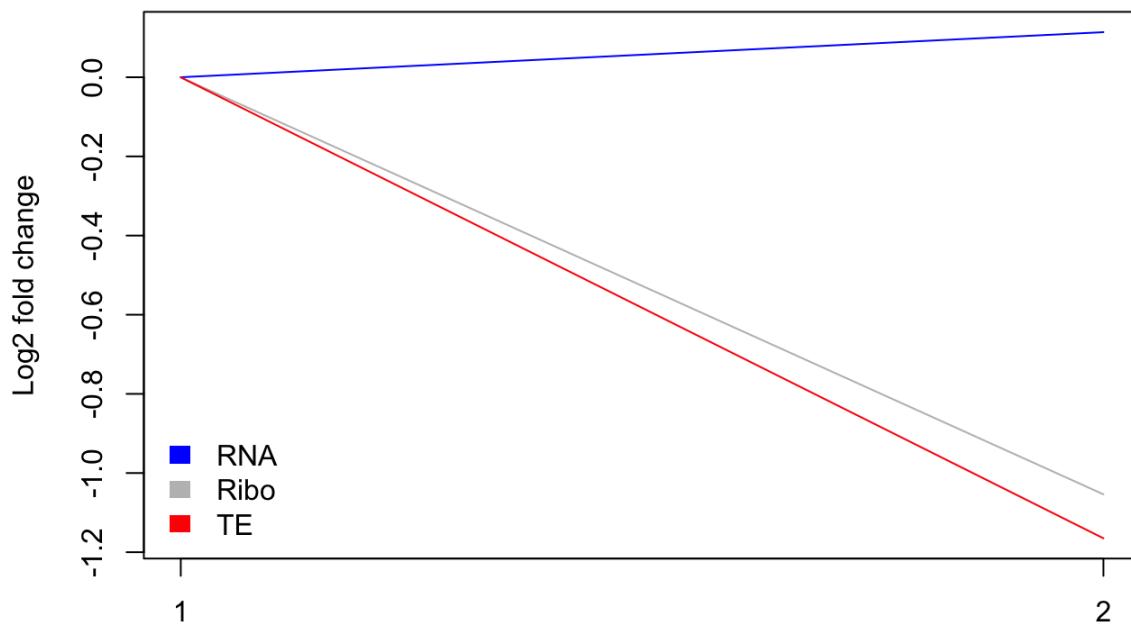
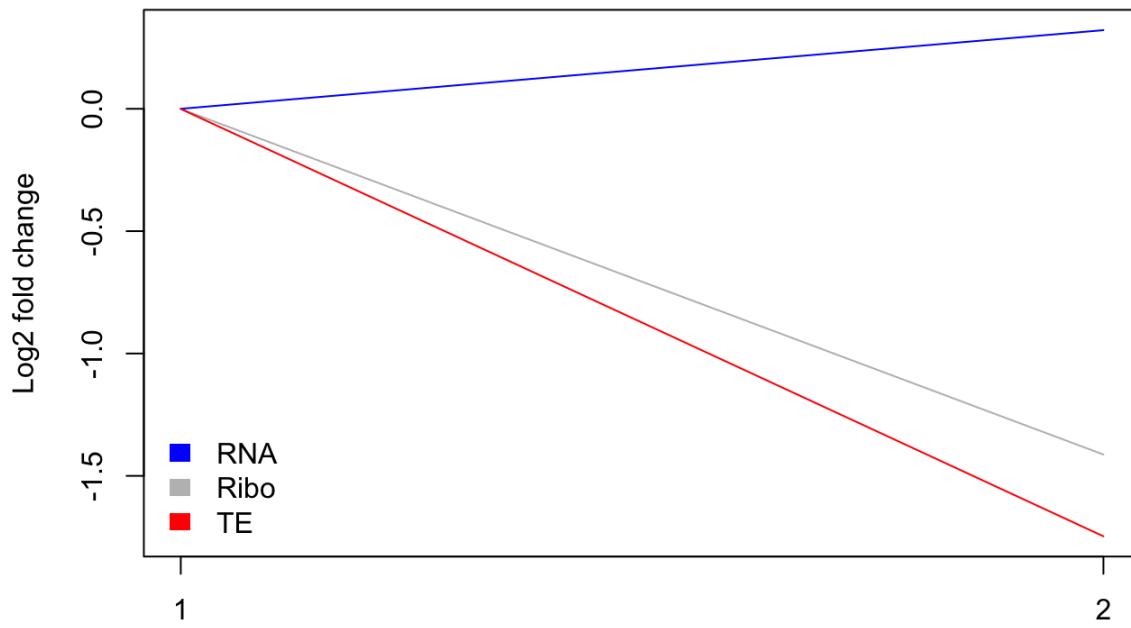


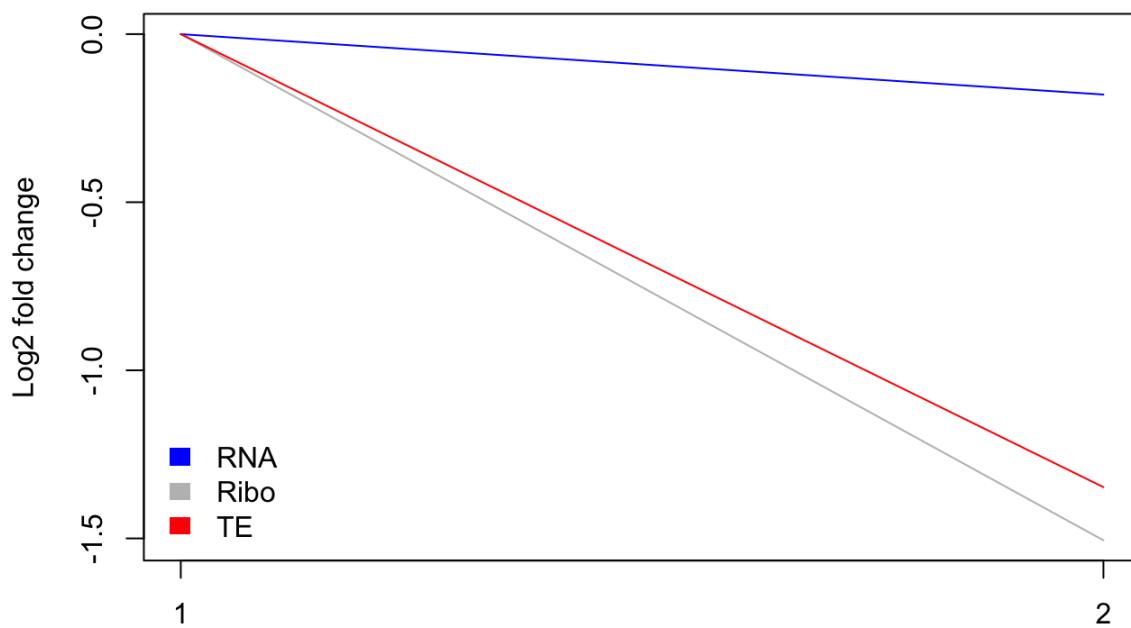
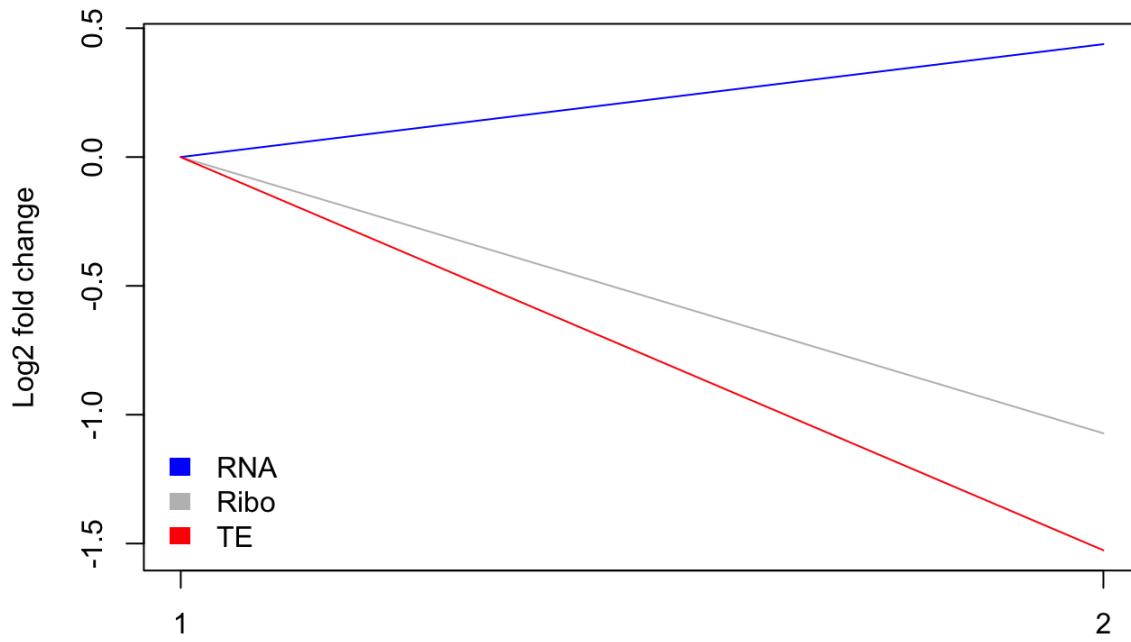
**AT1G23740****AT1G29930**

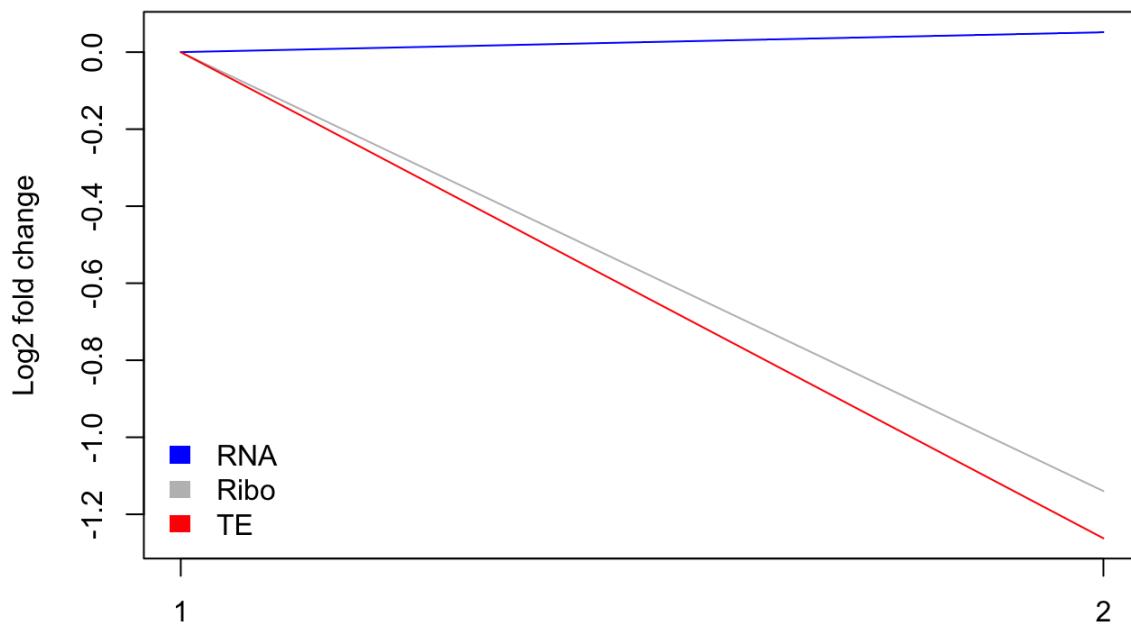
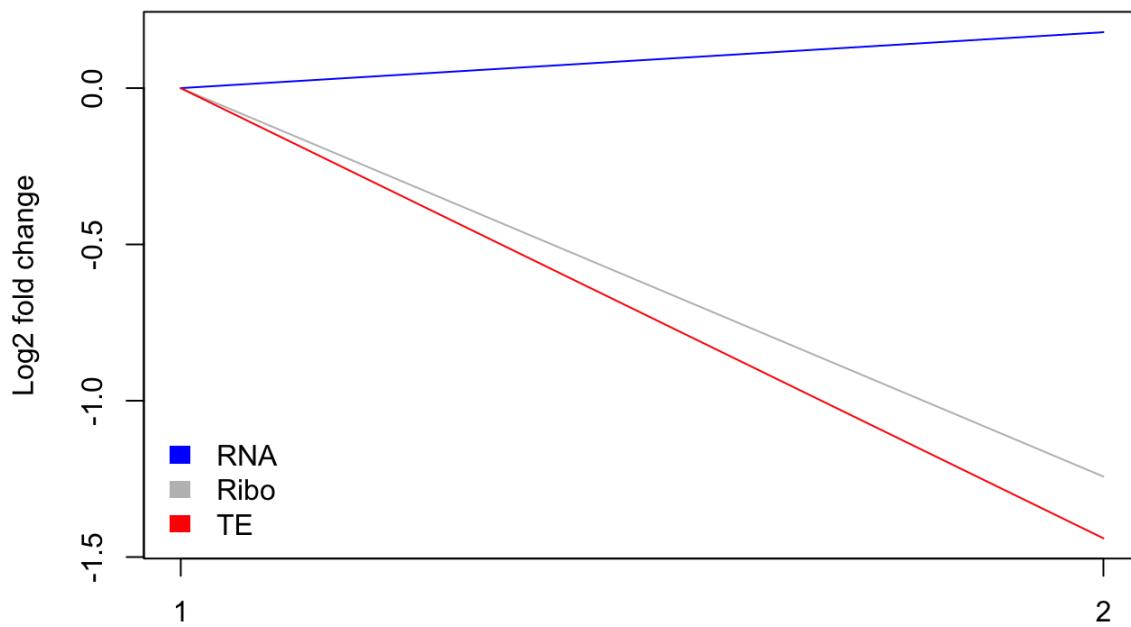
**AT1G44575****AT1G55670**

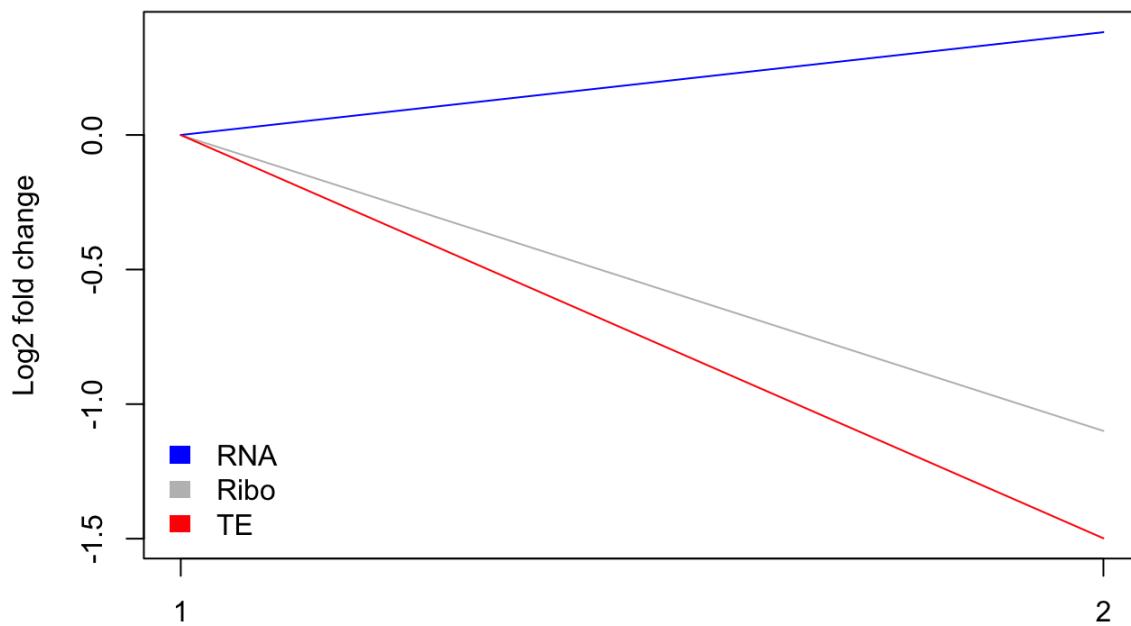
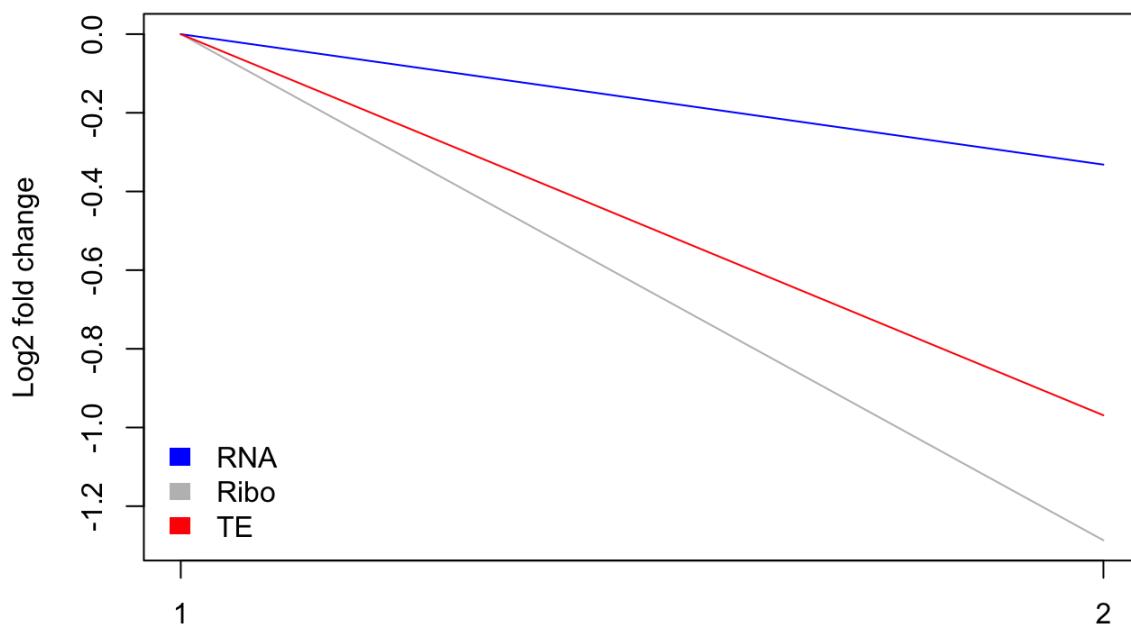
**AT1G61520****AT1G67090**

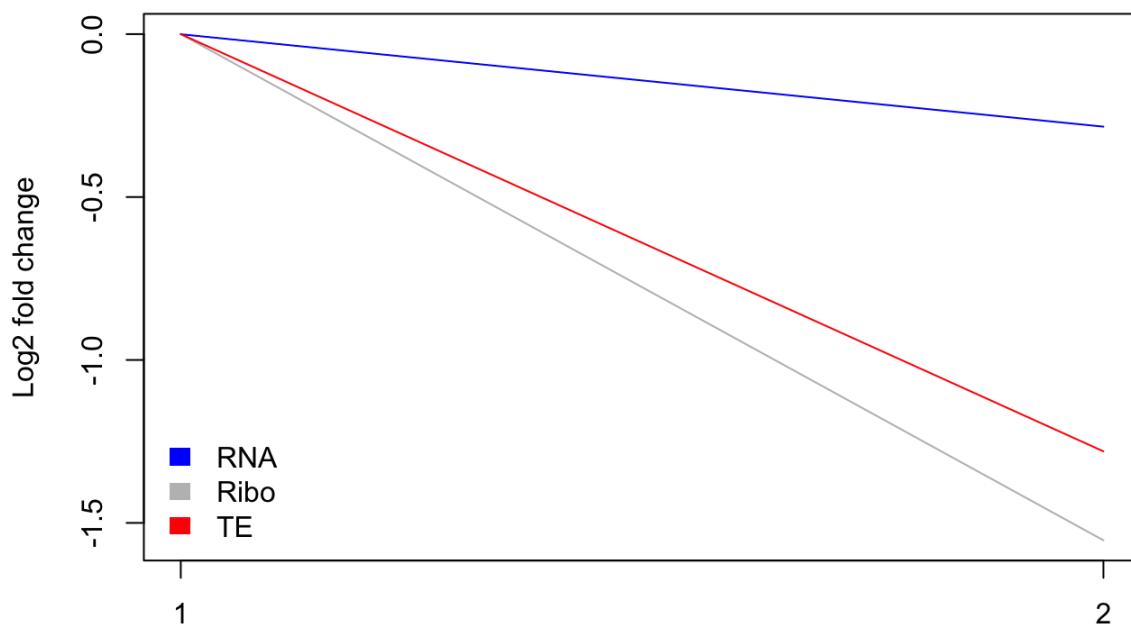
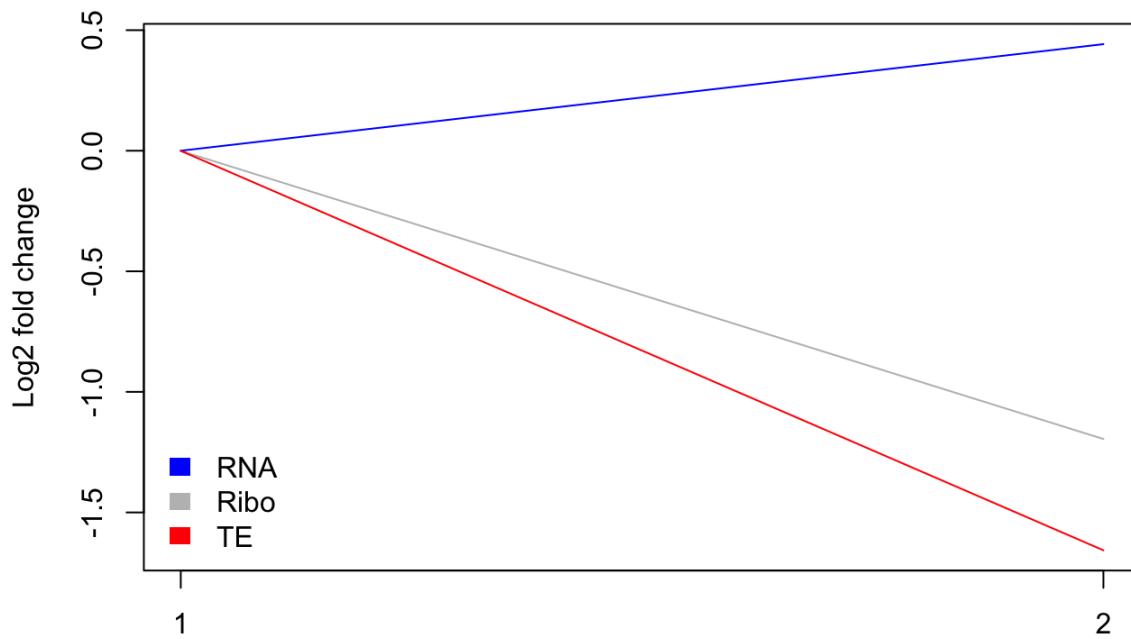
**AT2G37220****AT2G39010**

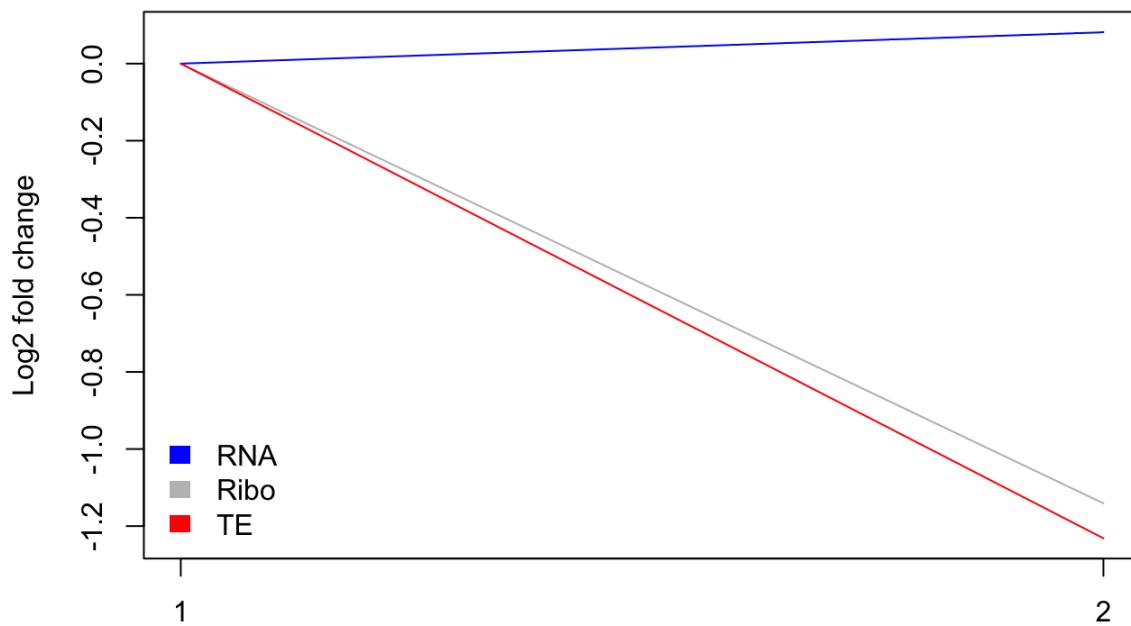
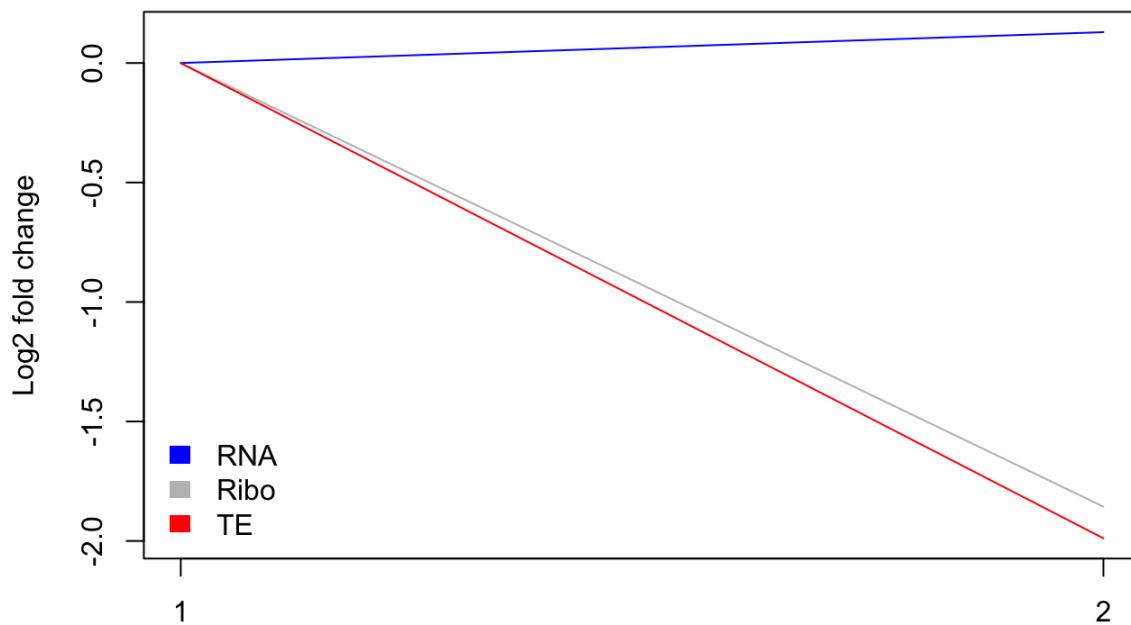
**AT2G43030****AT2G47400**

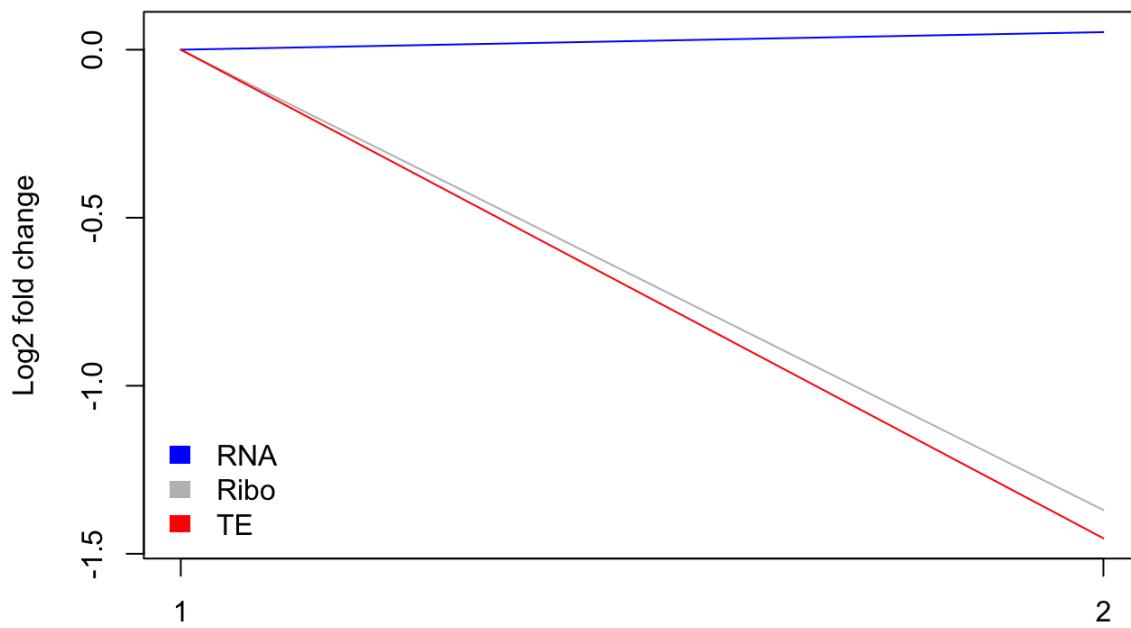
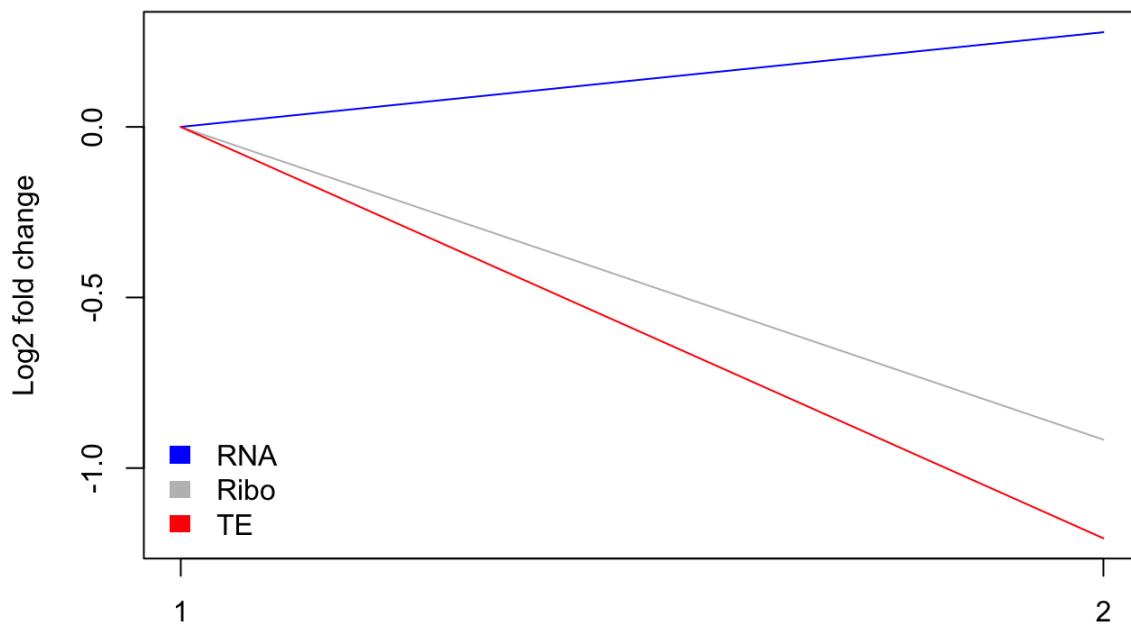
**AT3G15360****AT3G44310**

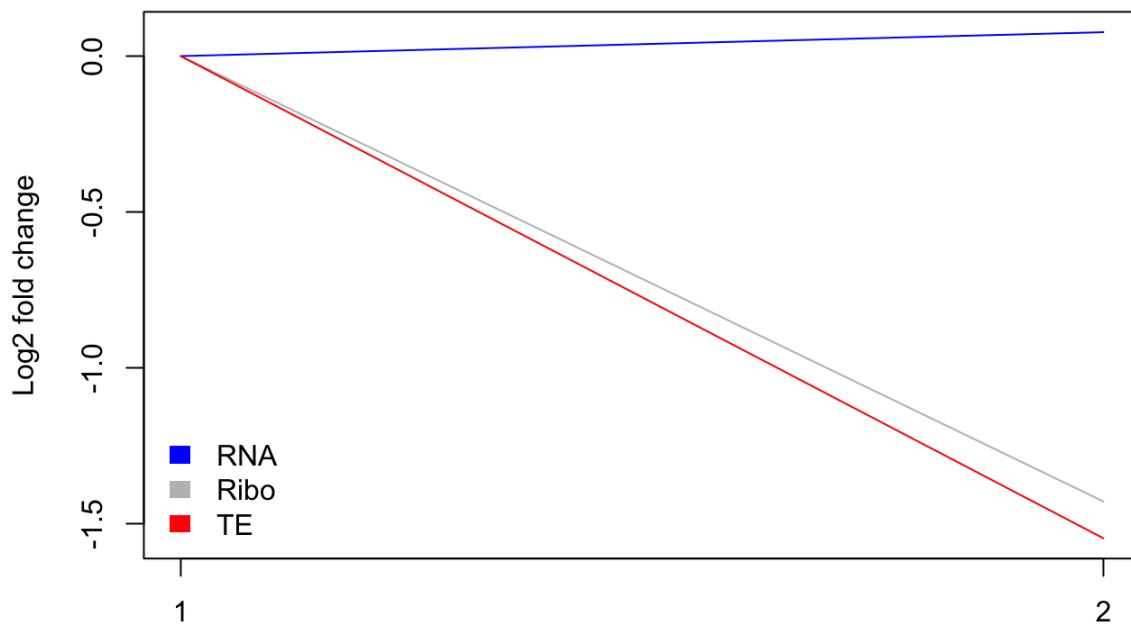
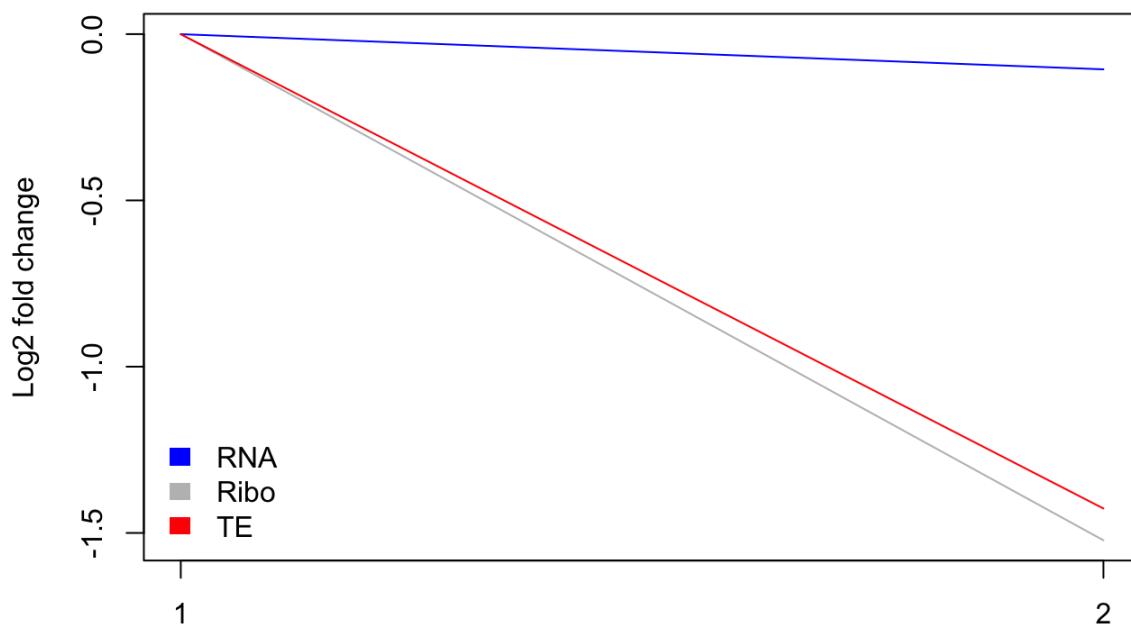
**AT3G54050****AT3G54890**

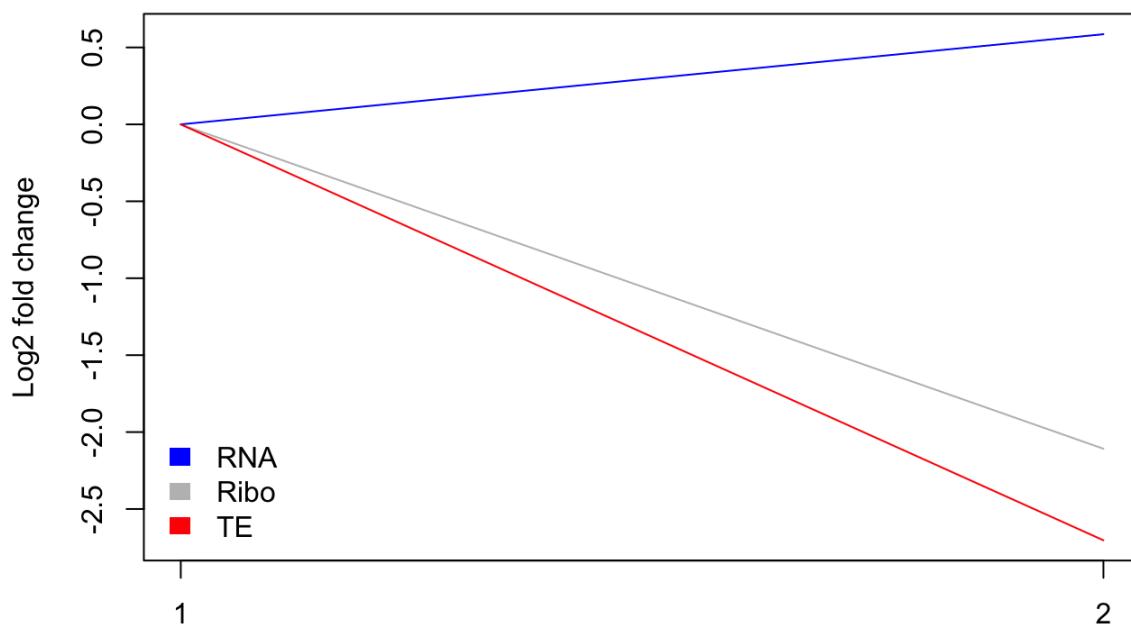
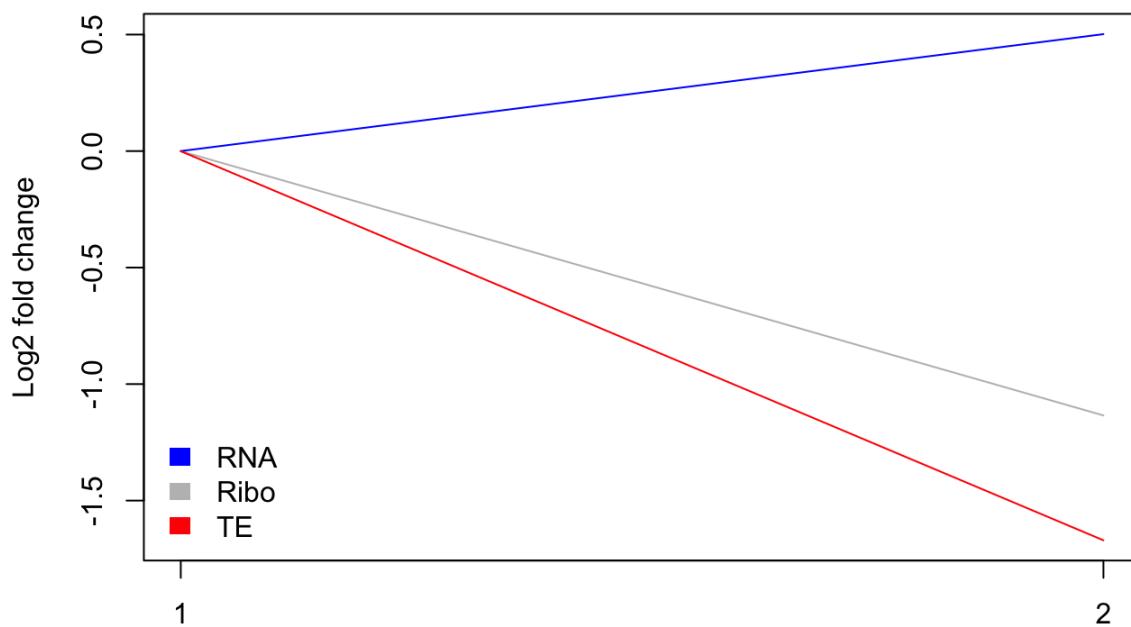
**AT3G58610****AT3G60750**

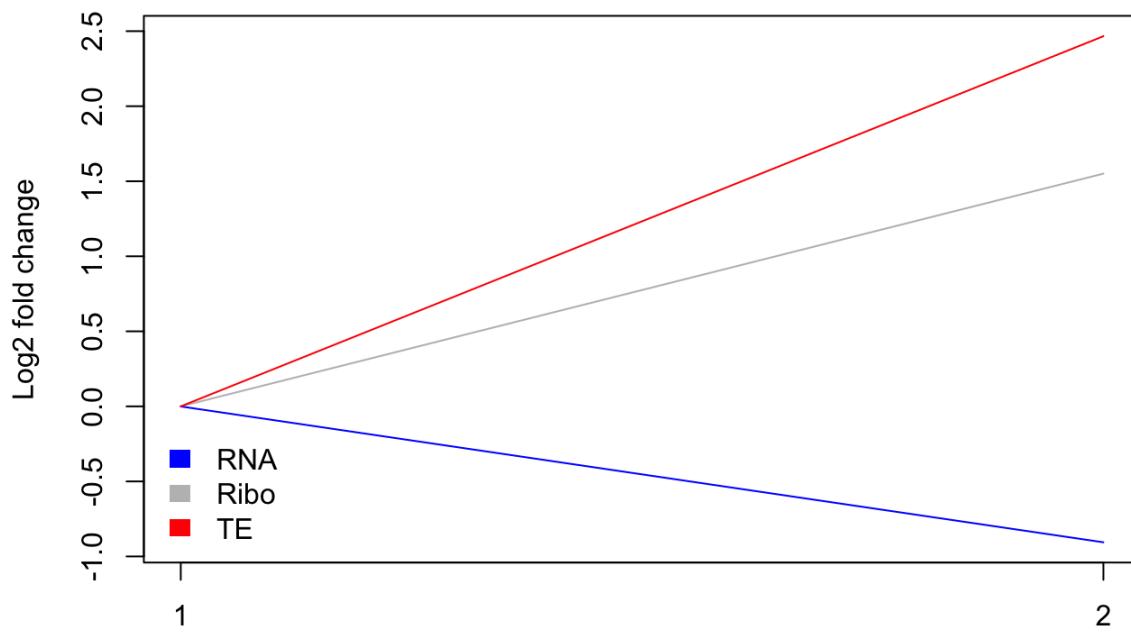
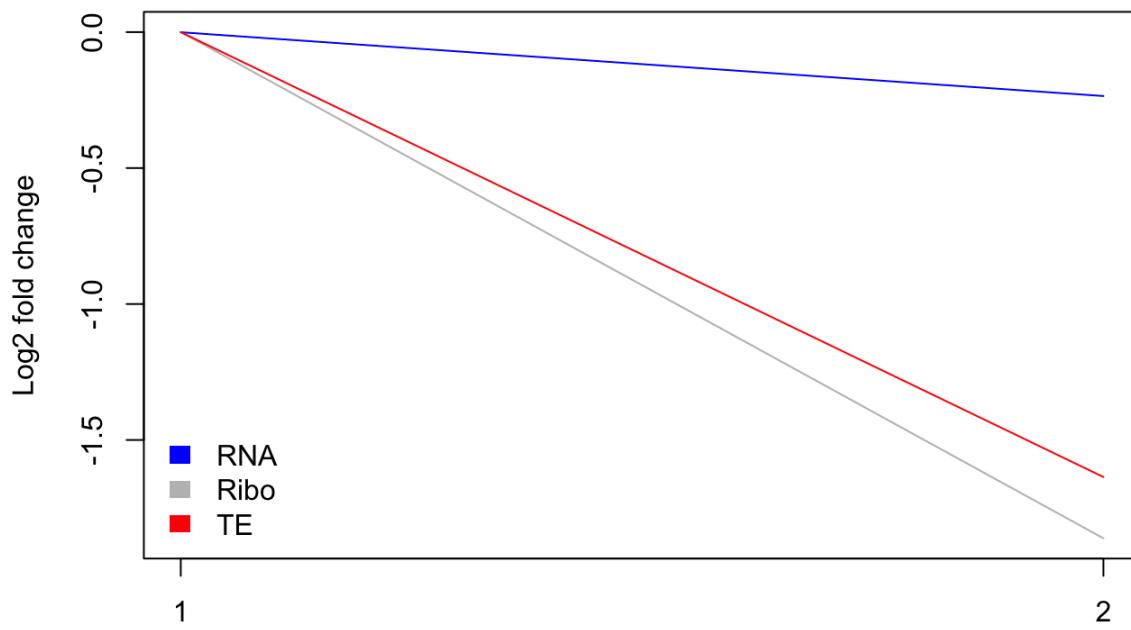
**AT3G61470****AT4G00100**

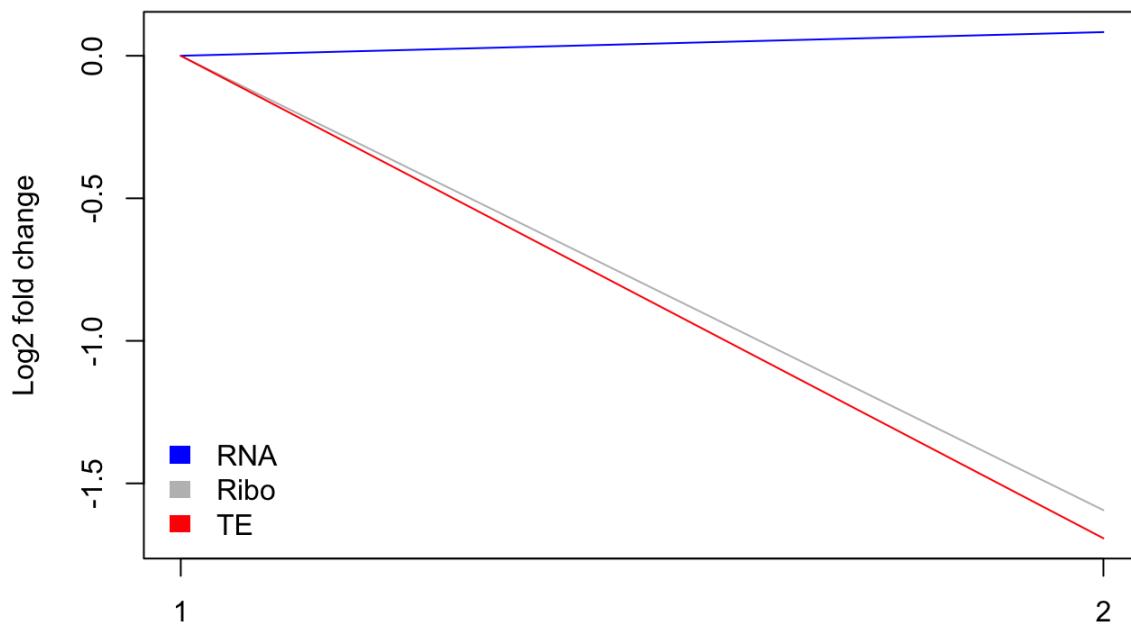
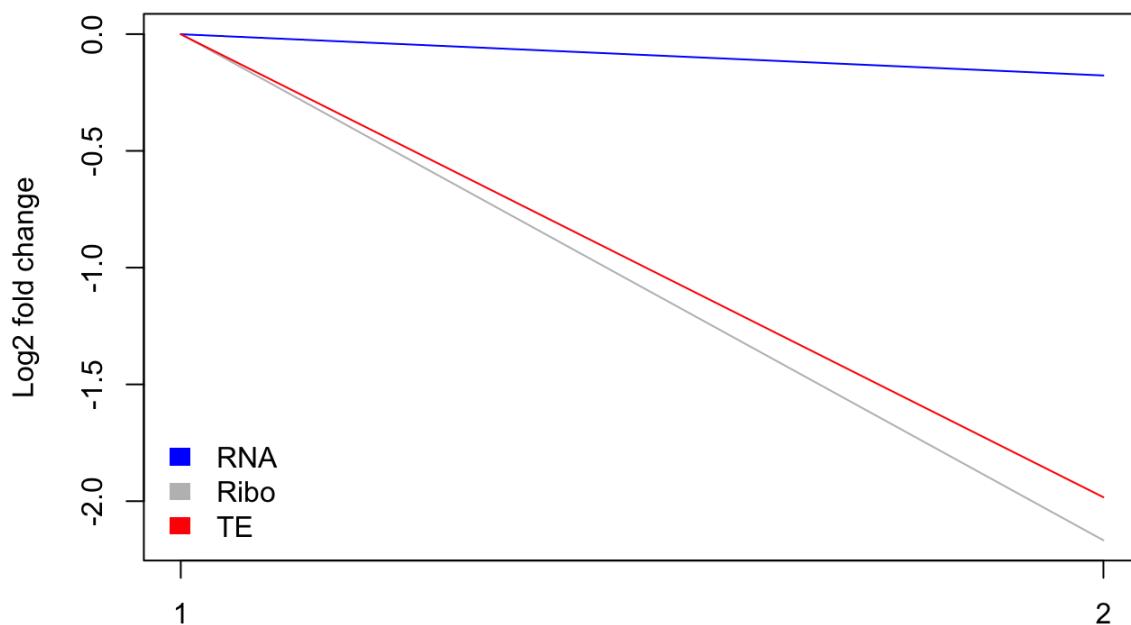
**AT4G03280****AT4G24770**

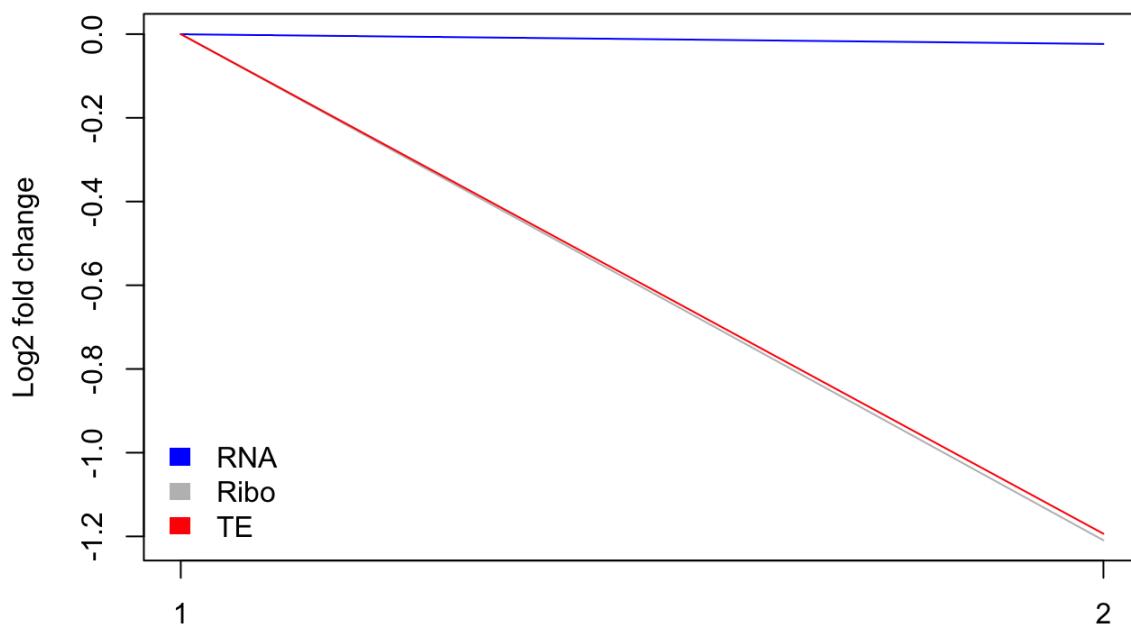
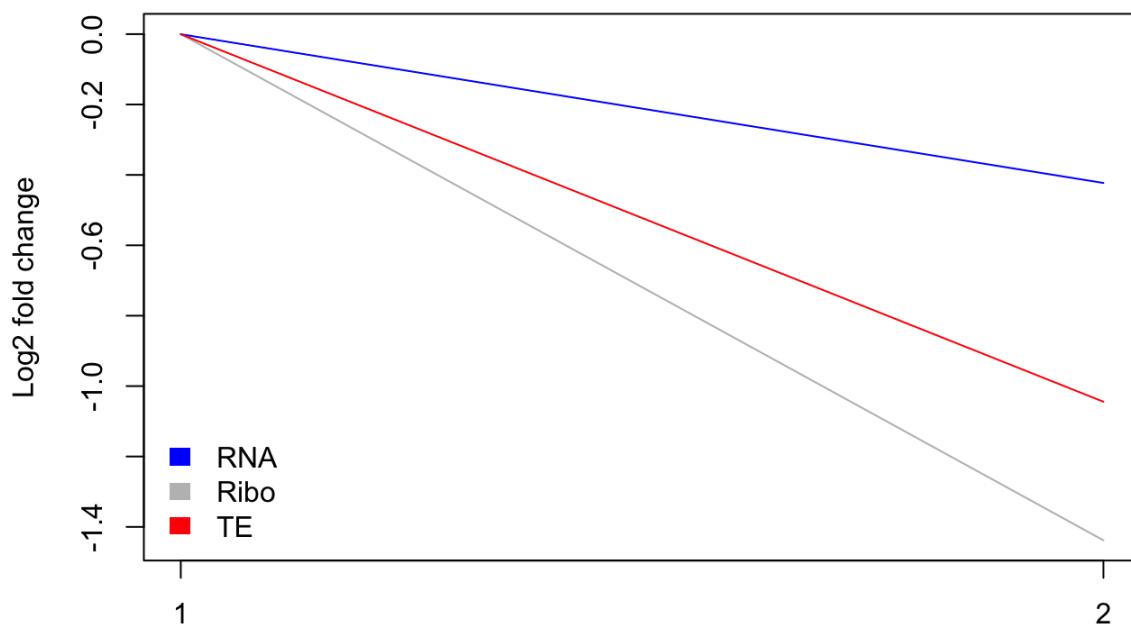
**AT4G25080****AT4G25100**

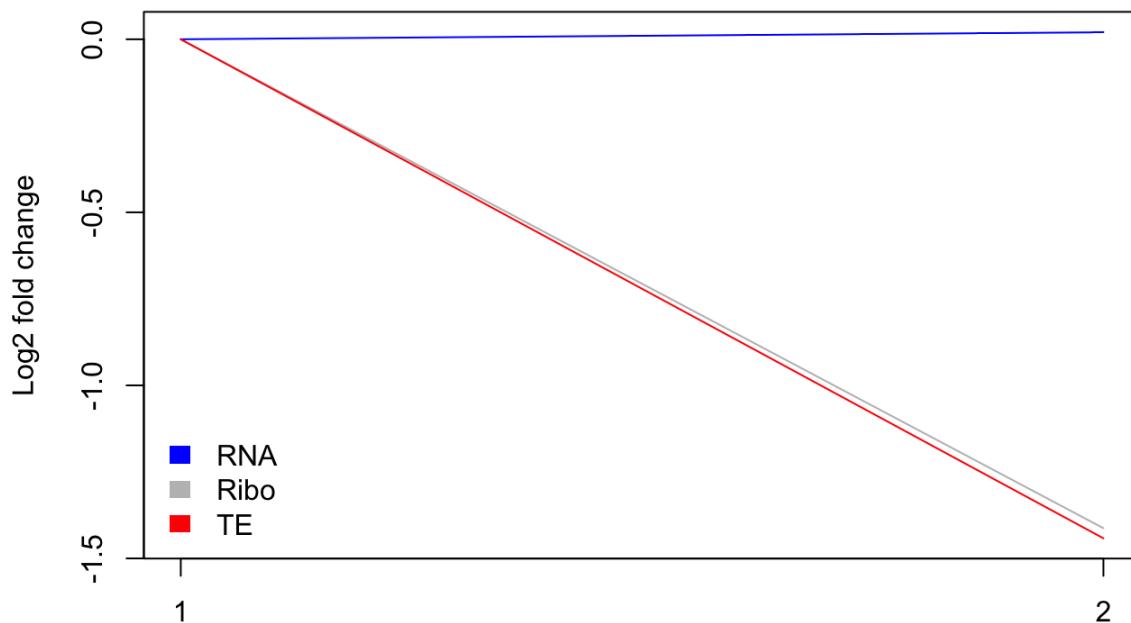
**AT4G32260****AT5G01530**

**AT5G02500****AT5G11770**

**AT5G22920****AT5G38410**

**AT5G38420****AT5G54270**

**AT5G54770****AT5G66190**

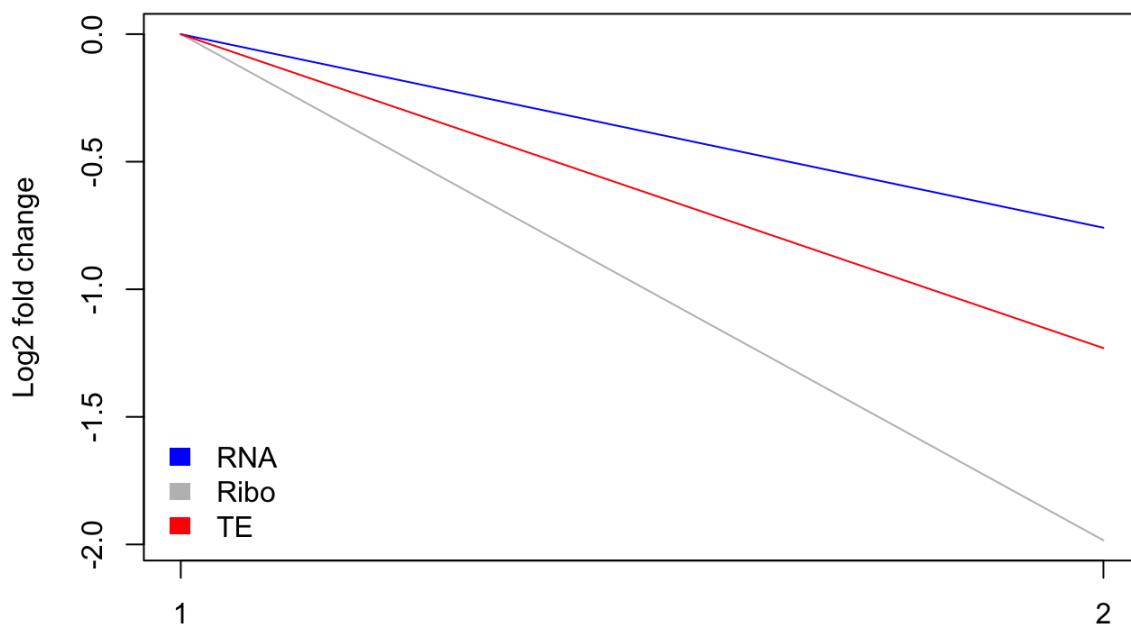
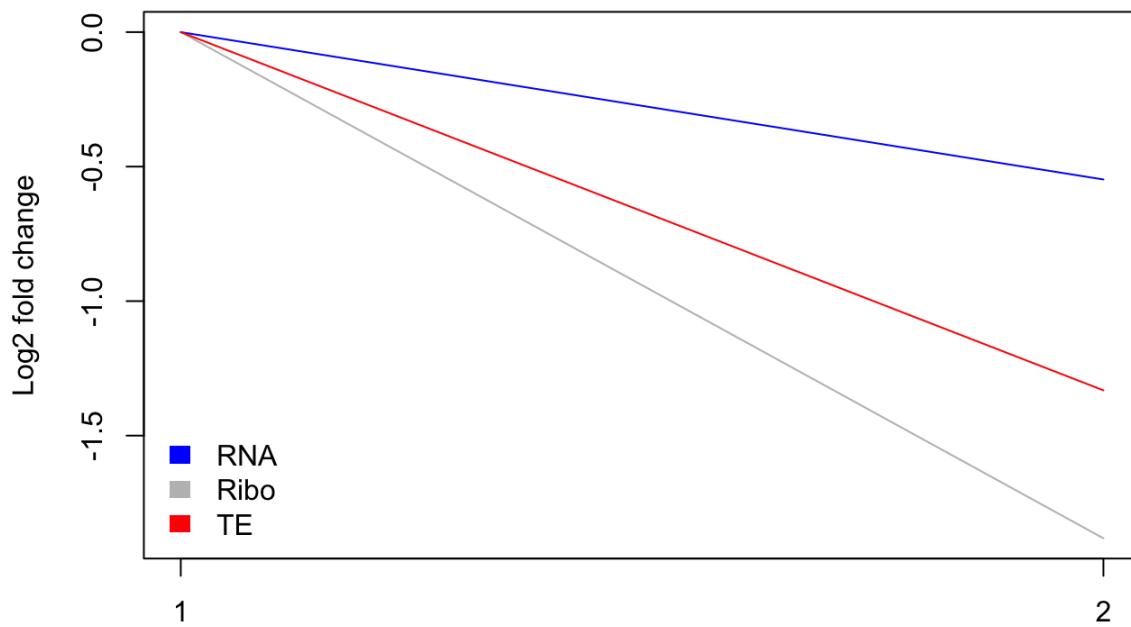
**AT5G66570**

exclusive focuses on findings that are translationally different only.

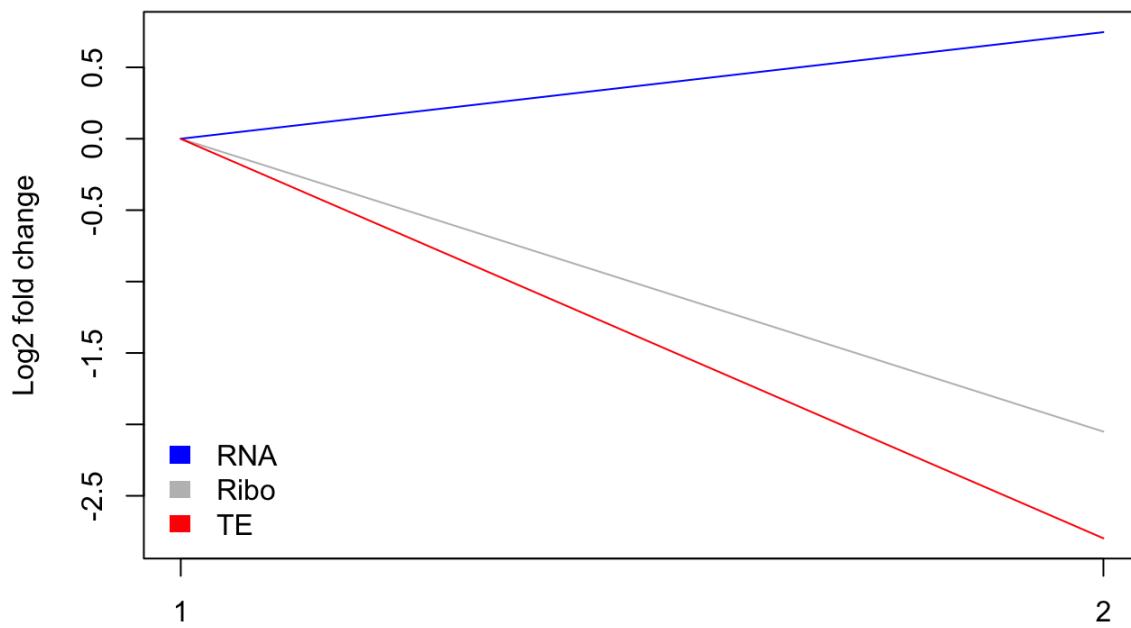
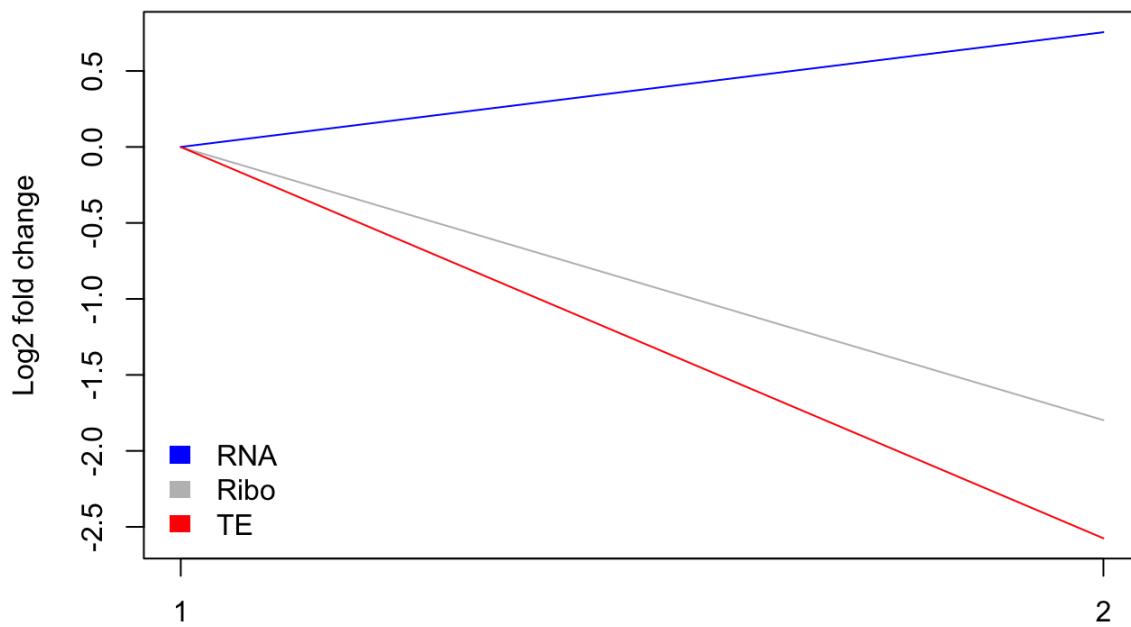
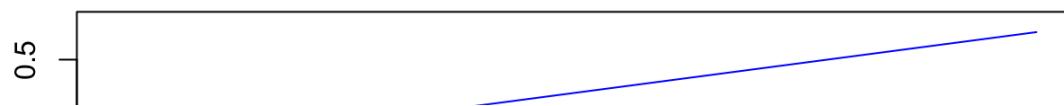
```

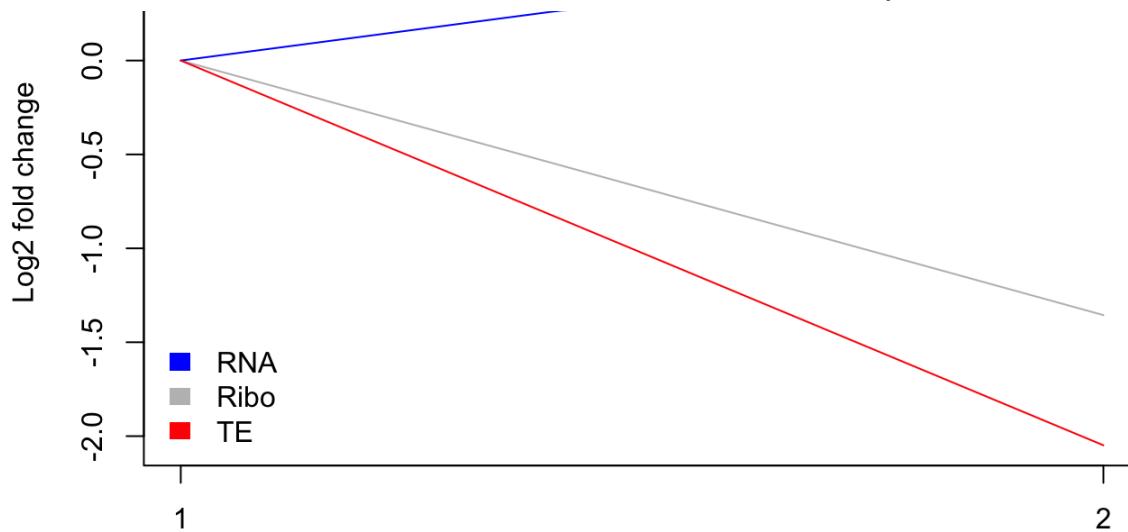
for (id in intensified){
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),
       ylab="Log2 fold change",xlab="",xaxt="n")
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"
  )
  axis(1,at=c(0,1),labels=c(1,2),las=1)
  title(id)
}

```

**AT3G12780****AT3G27160**

```
for (id in buffered){  
  ymax=max(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)  
  ymin=min(DESeq_Results_ribo[id,2],DESeq_Results_RNA[id,2],DESeq_Results_both[id,2],0)  
  plot(c(0,1), c(0,DESeq_Results_ribo[id,2]), type="l",col="gray", ylim=c(ymin,ymax),  
       ylab="Log2 fold change",xlab="",xaxt="n")  
  lines(c(0,1), c(0,DESeq_Results_RNA[id,2]),type="l",col="blue")  
  lines(c(0,1), c(0,DESeq_Results_both[id,2]), type="l",col="red")  
  legend("bottomleft",c("RNA","Ribo","TE"),fill=c("blue","gray","red"), cex=1, border = NA, bty="n"  
)  
  axis(1,at=c(0,1),labels=c(1,2),las=1)  
  title(id)  
}
```

**AT1G29910****AT2G41840****AT5G19510**



###Comparison genotypes Col and 4G, Contrast C

```

riboCountsC <- riboCounts %>% select(-contains("14B"))
rnaCountsC <- rnaCounts %>% select(-contains("14B"))
sampleAnnotationC <- sampleAnnotation %>% filter(!str_detect(group, '14B'))
sampleAnnotation2C <- sampleAnnotation2 %>% filter(!str_detect(group, '14B'))

# rna and ribo
sampleAnnotationC$SeqType = "RNA"
sampleAnnotation2C$SeqType = "Ribo"
combinedCountsC = cbind(riboCountsC, rnaCountsC)
sampleAnnotation3C = rbind(sampleAnnotationC, sampleAnnotation2C)
colnames(combinedCountsC) = rownames(sampleAnnotation3C)

# time + genotype + time:genotype + SeqType + SeqType:time + SeqType:genotype + SeqType:time:genotype
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = combinedCountsC,
  colData = sampleAnnotation3C,
  design = ~ time * genotype * SeqType
)
  
```

```

## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
  
```

```

DESeqDataSet_both = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ (time + genotype + SeqType)^2
)
  
```

```

## estimating size factors
  
```

```
## estimating dispersions

## gene-wise dispersion estimates

## mean-dispersion relationship

## final dispersion estimates

## fitting model and testing
```

```
DESeq_Results_both <- results(DESeqDataSet_both)
clean_DESeq_padj <- which(!is.na(DESeq_Results_both$padj))
sum(DESeq_Results_both[clean_DESeq_padj, "padj"] <= 0.1)
```

```
## [1] 0
```

```
sum(DESeq_Results_both[clean_DESeq_padj, "padj"] <= 0.1) * 0.1
```

```
## [1] 0
```

```
# rna
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = rnaCountsC,
  colData = sampleAnnotationC,
  design = ~ time + genotype + time:genotype
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
DESeqDataSet = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ time + genotype
)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
DESeq_Results_RNA <- results(DESeqDataSet)

# ribo
DESeqDataSet = DESeqDataSetFromMatrix(
  countData = riboCountsC,
  colData = sampleAnnotation2C,
  design = ~ time + genotype + time:genotype
)
```

```
## Warning in DESeqDataSet(se, design = design, ignoreRank): some variables in
## design formula are characters, converting to factors
```

```
DESeqDataSet = DESeq(
  DESeqDataSet,
  parallel=FALSE,
  test = "LRT",
  reduced = ~ time + genotype
)
```

```
## estimating size factors
```

```
## estimating dispersions
```

```
## gene-wise dispersion estimates
```

```
## mean-dispersion relationship
```

```
## final dispersion estimates
```

```
## fitting model and testing
```

```
DESeq_Results_ribo <- results(DESeqDataSet)
```

```
lgNorm = log2(counts(DESeqDataSet_both, normalized=TRUE) + 1)
```

## Overall PCA Plot

```
pca = prcomp(t(lgNorm))
pcaData = data.frame(pca$x[, 1:2])
pcaData$group = sampleAnnotation3C[rownames(pcaData), "group"]
pcaData$sample = rownames(pcaData)
gg = ggplot(pcaData, aes(x=PC1, y=PC2, color=group, label=sample))
gg = gg + geom_point(size=2.5, alpha=0.8)
gg = gg + scale_color_manual(values=groupColors)
gg = gg + theme(panel.background = element_rect(fill = 'aliceblue')) + ggtitle("Overall PCA for Tra
nslational Efficiency")
print(gg)
```

## Overall PCA for Translational Efficiency

