Identifying over-represented GO terms

In this notebook we will identify the over-represented GO terms in the proteins previously identified as changing RNA binding. The steps in this analysis are:

1. Model the relationship between differential RNA binding and an expected bias factor
2. Identify all the over-represented GO terms
3. Filter by FDR and degree of over-representation
4. Plot
5. Compare GO terms over-represented when using `limma` or `lm`

```r
library(tidyverse)
library(goseq)
library(Hmisc)
library(limma)
library(biobroom)

# set up standardised plotting scheme
theme_set(theme_bw(base_size = 20) +
            theme(panel.grid.major=element_blank(),
                  panel.grid.minor=element_blank(),
                  aspect.ratio=1))

cbPalette <- c("#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7", "#999999")
```

We need to source some functions which are stored in a script called "GO.R".

```r
source("./GO.R")
```

Below, we load the result for differential RNA binding testing from the previous notebooks

```r
limma_rna_binding_fit <- readRDS("../results/limma_rna_binding_fit.rds")

compare_methods <- readRDS("../results/compare_methods_rna_binding_results.rds")
```

To perform the differential RNA binding testing, we will use the R package `goseq`. This was orignally designed to account for length bias in RNA-Seq count-based differential expression testing but can be applied to any GO over-representation scenario where one expects there is a bias due to increased power to detect changes for some features. The basic idea is that the bias factor must have a monotonic relationship with the probability of a feature presenting as significantly changed. `goseq` estimates this relationship with a spline fit (we will see this later). `goseq` then takes this relationship into account when performing the GO over-representation testing. For more details about `goseq`, see here: https://genomebiology.biomedcentral.com/articles/10.1186/gb-2010-11-2-r14

In our case, we have reason to suspect that proteins with higher intensity will be more likely to be identified as having significantly altered RNA binding since the variance will be smaller so we will have more power to detect differences.

The first step is to make a probability weight function (PWF) linking the bias factor with the probabilty of significant difference using the `nullp` function.

For this, we need a boolean to indicate which proteins we identify as having a significant change and a vector of the same length which details our bias factor.

The `limma` output contains a "AveExpr" value for each protein which represents the mean intensity. This is precisely the bias factor we are after so we can use this directly.

```
bias <- compare_methods$AveExpr
print(length(bias))
```

```
## [1] 1916
```

For the booleans, we can extract this from our results tables. We'll set our thresholds at 1% FDR and >= 1.4 fold change. Note that these are completely arbitrary thresholds of course!

```
limma_rna_binding_fit_treat <- treat(limma_rna_binding_fit, lfc=log2(1.4))

limma_rna_binding_changes <- topTreat(limma_rna_binding_fit_treat, coef = "conditionG1:typeOOPS", n = In
                                      p.value=0.01, adjust.method="BH", confint=0.95)

limma_rna_binding_changes <- limma_rna_binding_changes[limma_rna_binding_changes$logFC>log2(1.5),]

limma_sig_bool <- compare_methods$Row.names %in% rownames(limma_rna_binding_changes)
names(limma_sig_bool) <- compare_methods$Row.names
print(table(limma_sig_bool))
```

```
## limma_sig_bool
## FALSE  TRUE
##  1806   110
```

```
lm_sig_bool <- (compare_methods$lm_BH < 0.01 & compare_methods$logFC>log2(1.4))
names(lm_sig_bool) <- compare_methods$Row.names
print(table(lm_sig_bool))
```

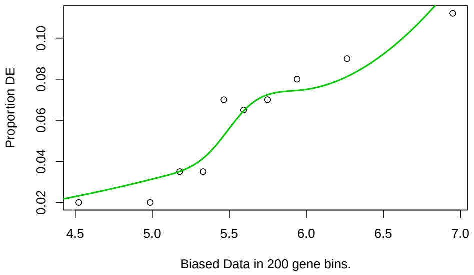
```
## lm_sig_bool
## FALSE  TRUE
##  1597   319
```

Questions: Why do we have fewer proteins with an increase in RNA binding according to limma?

By default, the function expects a genome name and it will go and fetch the bias data. However, we can also just provide this data ourselves. Below, we obtain the PWF and `goseq` outputs plots to show how closely these fit the observed relationship.

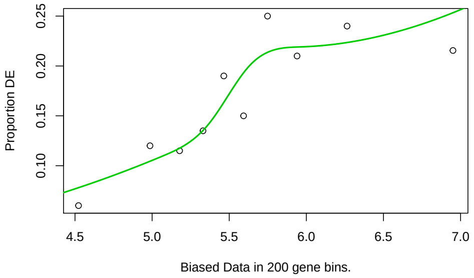
```
limma_pwf <- nullp(limma_sig_bool, bias.data=bias, plot.fit=TRUE)
```

```
## Warning in pcls(G): initial point very close to some inequality constraints
```



```
lm_pwf <- nullp(lm_sig_bool, bias.data=bias, plot.fit=TRUE)
```

```
## Warning in pcls(G): initial point very close to some inequality constraints
```



Now, we need those GO terms from the previous notebook

```
sapiens.go.full <- readRDS("../results/h_sapiens_go_full.rds")
head(sapiens.go.full)
```

```
##   UNIPROTKB      GO.ID                           TERM ONTOLOGY
## 1    P09874 GO:0000002 mitochondrial genome maintenance       BP
## 2    Q9UJZ1 GO:0000002 mitochondrial genome maintenance       BP
## 3    Q9Y243 GO:0000002 mitochondrial genome maintenance       BP
## 4    Q02078 GO:0000002 mitochondrial genome maintenance       BP
## 5    Q9BUK6 GO:0000002 mitochondrial genome maintenance       BP
## 6    Q96RR1 GO:0000002 mitochondrial genome maintenance       BP
```

And we're ready to run `goseq`. By default, `goseq` expects a `genome` and will obtain the GO terms automatically. However, if we provide our own to the `gene2cat` argument, it will use this instead. This also means we could use `goseq` to interrogate over-represented KEGG terms, interpro domains, etc by simply providing them at this point.

```
## Using manually entered categories.
```

```
## For 14 genes, we could not find any categories. These genes will be excluded.
```

```
## To force their use, please run with use_genes_without_cat=TRUE (see documentation).
## This was the default behavior for version 1.15.1 and earlier.

## Calculating the p-values...

## 'select()' returned 1:1 mapping between keys and columns

## Using manually entered categories.

## For 14 genes, we could not find any categories. These genes will be excluded.

## To force their use, please run with use_genes_without_cat=TRUE (see documentation).

## This was the default behavior for version 1.15.1 and earlier.

## Calculating the p-values...

## 'select()' returned 1:1 mapping between keys and columns
```

We need to adjust the p-values to account for multiple testing. Again, we'll use the Benjamini-Hochberg method.

```
limma_over_rep_go$BH <- p.adjust(limma_over_rep_go$over_represented_pvalue, method="BH")
lm_over_rep_go$BH <- p.adjust(lm_over_rep_go$over_represented_pvalue, method="BH")
```

Questions:

- How many GO terms did we test for over-representation in the 'limma' results?

- How many would we expect to have a p-value < 0.05 by chance?

- How many did have a p-value < 0.05?

- What can you infer from this?

We can filter the output to only include the GO terms which are over-represented at 1% FDR (there are 112 for `limma`)

```
sig_terms <- limma_over_rep_go %>% filter(BH<0.01)
print(dim(sig_terms))
```

```
## [1] 112   8
```

```
print(head(sig_terms))
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0044281            4.028782e-24                        1         63
## 2 GO:0055114            1.958873e-19                        1         32
## 3 GO:0044429            3.100699e-19                        1         41
## 4 GO:0019752            4.011556e-18                        1         46
## 5 GO:0043436            4.942581e-18                        1         46
## 6 GO:0006082            6.292715e-18                          1         46
##   numInCat                              term ontology           BH
## 1      310  small molecule metabolic process       BP 4.769272e-20
## 2       83       oxidation-reduction process       BP 1.159457e-15
## 3      159                mitochondrial part       CC 1.223536e-15
## 4      203 carboxylic acid metabolic process       BP 1.170206e-14
## 5      204          oxoacid metabolic process       BP 1.170206e-14
## 6      205    organic acid metabolic process       BP 1.241553e-14
```

We would also like to know the effect size, e.g how over-represented are the terms. We could use the `numDEInCat` and `numInCat` columns and calculate the over-representation as:

(`numInCat`/ Number of proteins observed) / (`numDEInCat`/ Number of proteins with change in RNA binding).

Questions: Why would it be sub-optimal to calculate the magnitude of the over-representation as indicated above?

The function below estimates the effect size, taking into account the bias (using the PWF)

```
# -----------------------------------------------------------------------------------------------------
# Function  : 'addAdjustedOverRep' A crude function to add an adjusted estimate of the over-representat
# Input
#           : obj = A data frame with the results from goseq. As generated by GetEnrichedGO
#           : pwf = a PWF from goseq
#           : gene2cat = A dataframe mapping features to categories
# Output   : The input obj + a column with estimated adjusted over-representation for each term ($adj_ov
# -----------------------------------------------------------------------------------------------------

addAdjustedOverRep <- function(obj, pwf, gene2cat){
  len_fore <- sum(pwf$DEgenes)
  len_back <- length(pwf$DEgenes)

  obj$adj_over_rep <- apply(obj[,c("numDEInCat", "numInCat", "category")], MARGIN=1, function(x){
    term_features <- gene2cat[gene2cat[["GO.ID"]]==x[["category"]], "UNIPROTKB"]
    term_weight <- mean(pwf[rownames(pwf) %in% term_features, "pwf"])
    non_term_weight <- mean(pwf[!rownames(pwf) %in% term_features, "pwf"])
    as.numeric(x[['numDEInCat']])/as.numeric(x[['numInCat']]) / (term_weight/non_term_weight) / (len_fo

  return(obj)
}
```

Now we can filter by the estimated over-representation too.

```
limma_over_rep_go %>% filter(BH<0.01)  %>%
  addAdjustedOverRep(lm_pwf, sapiens.go.full) %>%
  filter(adj_over_rep>2) %>%
  head()
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0055114            1.958873e-19                        1         32
## 2 GO:0045333            4.456564e-11                        1         15
## 3 GO:0015980            5.051024e-10                        1         16
## 4 GO:0009060            7.197845e-09                          1         11
## 5 GO:0004812            5.568684e-08                          1         11
## 6 GO:0016875            5.568684e-08                          1         11
##   numInCat                                          term ontology
## 1       83                      oxidation-reduction process       BP
## 2       34                             cellular respiration       BP
## 3       46 energy derivation by oxidation of organic compounds       BP
## 4       22                              aerobic respiration       BP
## 5       24                    aminoacyl-tRNA ligase activity       MF
## 6       24        ligase activity, forming carbon-oxygen bonds       MF
##             BH adj_over_rep
## 1 1.159457e-15     2.195040
## 2 4.058216e-08     2.608945
## 3 3.737126e-07     2.102115
## 4 3.873095e-06     2.831174
## 5 2.354360e-05     2.400932
## 6 2.354360e-05     2.400932
```

```
lm_over_rep_go %>% filter(BH<0.01)  %>%
  addAdjustedOverRep(lm_pwf, sapiens.go.full) %>%
  filter(adj_over_rep>2) %>%
  head()
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0044281            4.514926e-33                        1        133
## 2 GO:0043436            9.437890e-27                        1         97
## 3 GO:0006082            1.644155e-26                        1         97
## 4 GO:0019752            3.248454e-26                        1         96
## 5 GO:0055114            4.699760e-21                        1         52
## 6 GO:0044429            2.148131e-19                        1         73
##   numInCat                              term ontology           BH
## 1      310  small molecule metabolic process       BP 5.344769e-29
## 2      204         oxoacid metabolic process       BP 5.586287e-23
## 3      205    organic acid metabolic process       BP 6.487837e-23
## 4      203 carboxylic acid metabolic process       BP 9.613801e-23
## 5       83       oxidation-reduction process       BP 9.272627e-18
## 6      159                mitochondrial part       CC 3.632796e-16
##   adj_over_rep
## 1     2.493771
## 2     2.715729
## 3     2.699929
## 4     2.701236
## 5     3.566940
## 6     2.715217
```

Note that many of the above GO terms are essentially redundant and are simply the same set of proteins with each of the GO terms up the heirachy being over-represented. We can perform a simplification by taking each over-represented GO term, identifying all its offspring and ancesters and removing it if any of the terms above or below are more significantly over-represented.

Below, I've wrapped this all up into a function so that we can pass a data.frame of significantly over-represented GO terms and it will remove the redundant terms.

```r
remove_redundant_GO_terms <- function(go_df){

  all_observed_go <- unique(go_df$category) # identify all the GO terms
  all_observed_go <- all_observed_go[!is.na(all_observed_go)] # Remove any NAs

  # Get the ontologies for the GO terms
  ontologies <- AnnotationDbi::select(GO.db, all_observed_go, columns = c('ONTOLOGY'), keytype='GOID')
  ontologies <- setNames(ontologies$ONTOLOGY, ontologies$GOID)

  # Get the mappings from GO term to parent GO terms using functions in GO.R
  go2Offspring <- getAllMappings(all_observed_go, ontologies, verbose=FALSE, direction="offspring")
  go2Ancesters <- getAllMappings(all_observed_go, ontologies, verbose=FALSE, direction="ancester")

  # start with all observed GO terms being retained
  retained <- all_observed_go

  # We want to keep track of the GO IDs we have processed
  processed <- NULL

  # If any GO term has no detected offspring or ancesters, mark them as already processed
  # This will also mean they are always retained
  no_anc_off <- setdiff(all_observed_go, union(names(go2Ancesters), names(go2Offspring)))
  if(length(no_anc_off)>0){
    cat(sprintf("No offspring or ancesters could be found for these terms: %s", no_anc_off))
    processed <- no_anc_off
  }

  # When all observed go terms are in processed, stop while loop
  while(length(setdiff(all_observed_go, processed))!=0){

    # take one of the GO terms
    go_id <- setdiff(all_observed_go, processed)[1]

    # Find all the ancesters and offspring = go_tree
    # (Only include those also observed as over-rep GO)
    go_tree <- union(go2Ancesters[[go_id]], go2Offspring[[go_id]]) %>%
      intersect(all_observed_go) %>% c(go_id)

    top_go <- go_df %>%
      filter(category %in% go_tree) %>% # subset to the terms in go_tree
      arrange(over_represented_pvalue) %>% # order by p-value (ascending by default)
      pull(category) %>%# pull out the category
      head(1) # keep the top GO term

    # We want to remove all ancester and offspring terms within the go_tree for the top GO term
    terms_to_remove <- union(go2Ancesters[[top_go]], go2Offspring[[top_go]]) %>%
      intersect(go_tree)

    processed <- union(processed, go_tree) # all terms in the tree are now considered "processed"

    retained <- setdiff(retained, terms_to_remove) # remove the unwanted terms from retained
  }
  go_df <- go_df %>% filter(category %in% retained) # subset to the retained terms

  return(go_df)
}
```

Task: Modify the `remove_redundant_GO_terms` function to keep an arbitrary number of top GO terms within each sub tree of GO terms

```
limma_over_rep_go %>% filter(BH<0.01)  %>%
  addAdjustedOverRep(limma_pwf, sapiens.go.full) %>%
  filter(adj_over_rep>3) %>%
  remove_redundant_GO_terms() %>%
  head()
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0044281            4.028782e-24                        1         63
## 2 GO:0055114            1.958873e-19                        1         32
## 3 GO:0044429            3.100699e-19                        1         41
## 4 GO:0006091            1.414456e-13                        1         27
## 5 GO:0017144            5.526715e-12                        1         30
## 6 GO:0016491            6.280439e-11                        1         25
##   numInCat                                     term ontology
## 1      310             small molecule metabolic process       BP
## 2       83                  oxidation-reduction process       BP
## 3      159                           mitochondrial part       CC
## 4       92 generation of precursor metabolites and energy       BP
## 5      128                      drug metabolic process       BP
## 6       95                      oxidoreductase activity       MF
##             BH adj_over_rep
## 1 4.769272e-20     3.494409
## 2 1.159457e-15     6.204881
## 3 1.223536e-15     4.640499
## 4 1.860482e-10     5.018165
## 5 5.947750e-09     3.918863
## 6 5.310560e-08     4.245300
```

```
lm_over_rep_go %>% filter(BH<0.01)  %>%
  addAdjustedOverRep(lm_pwf, sapiens.go.full) %>%
  filter(adj_over_rep>3) %>%
  remove_redundant_GO_terms() %>%
  head()
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0055114            4.699760e-21                        1         52
## 2 GO:0044282            2.495342e-14                        1         37
## 3 GO:0004812            1.039095e-11                        1         20
## 4 GO:0006418            2.455402e-10                          1         20
## 5 GO:0005743            2.561231e-09                          1         26
## 6 GO:0006734            6.737147e-08                          1         12
##   numInCat                                        term ontology
## 1       83                 oxidation-reduction process       BP
## 2       62             small molecule catabolic process       BP
## 3       24               aminoacyl-tRNA ligase activity       MF
## 4       26 tRNA aminoacylation for protein translation       BP
## 5       48                mitochondrial inner membrane       CC
## 6       13                      NADH metabolic process       BP
##             BH adj_over_rep
## 1 9.272627e-18     3.566940
## 2 1.969324e-11     3.440306
## 3 6.150401e-09     4.365332
## 4 7.855960e-08     3.928257
## 5 6.890877e-07     3.221653
## 6 1.035771e-05     4.298834
```

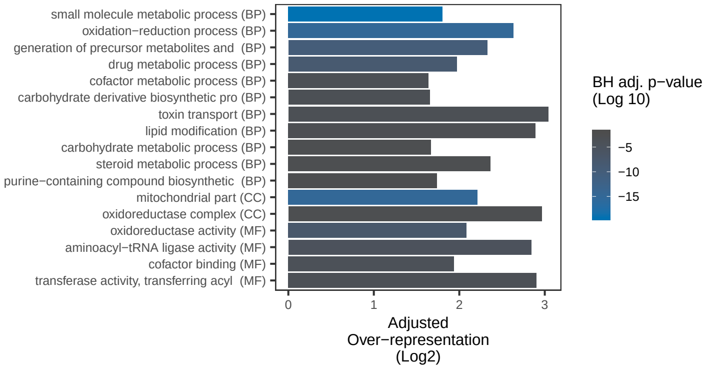
OK so now we're getting a more useful set of terms. The next thing we might want to do is some basic plot to show the result. Again, I've wrapped this up into a a function below so we can pass the GO dataframe straight through the above pipe to filter the results and then plot the filtered results.

```
plotTerms <- function(go_df,
                      horizontal=FALSE, # make plot hoizontal
                      plot_top=10,# plot the top n most significant GO terms
                      shorten_term=FALSE){ # shorten the term to max 30 char

  # re-order data frame by p-value
  if(horizontal){
    go_df <- go_df %>% arrange(ontology, over_represented_pvalue)
    go_df <- go_df %>% head(plot_top) # subset to top n most significant terms
  }
  else{
    go_df <- go_df %>% arrange(desc(ontology), desc(over_represented_pvalue))
    go_df <- go_df %>% tail(plot_top) # subset to top n most significant terms
  }

  if(shorten_term){
    go_df$term_for_plot <- substr(go_df$term, 1, 40) # cut at character 40
  }
  else{
    go_df$term_for_plot <- go_df$term
  }

  # add the ontology (BP, MF, CC) to the end of the term
  go_df$term_for_plot <- paste0(go_df$term_for_plot, " (", go_df$ontology, ")")

  # re-level factor make keep plotting order in order of dataframe (ontology, p-value)
  go_df$term_for_plot <- factor(go_df$term_for_plot, levels=rev(go_df$term_for_plot))

  p <- go_df %>%
    ggplot(aes(x=term_for_plot, y=log(adj_over_rep,2), fill=log(BH,10))) +
    geom_bar(stat="identity") + # When geom_bar is plotting a single data point, need to set stat="iden
    xlab("") +
    ylab("Adjusted\nOver-representation\n(Log2)") +
    scale_fill_continuous(name="BH adj. p-value\n(Log 10)\n", low=cbPalette[5], high="grey30", limits=c
    theme(text=element_text(size=15),
          plot.title=element_text(hjust=0.5))

  if(horizontal){
    p <- p + coord_flip() # Flip the coordinates
  }
  else{
    # If vertical bards, set the x-axis text at an angle so it fits better
    p <- p + theme(axis.text.x=element_text(size=12, angle=30, vjust=1, hjust=1))
  }

  return(list("p"=p, "data"=go_df))
}
```

```
limma_over_rep_go %>% filter(BH<0.01)  %>%
  addAdjustedOverRep(limma_pwf, sapiens.go.full) %>%
  filter(adj_over_rep>3) %>%
  remove_redundant_GO_terms() %>%
  plotTerms(horizontal=TRUE, shorten_term=TRUE, plot_top=20)
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## $p
```

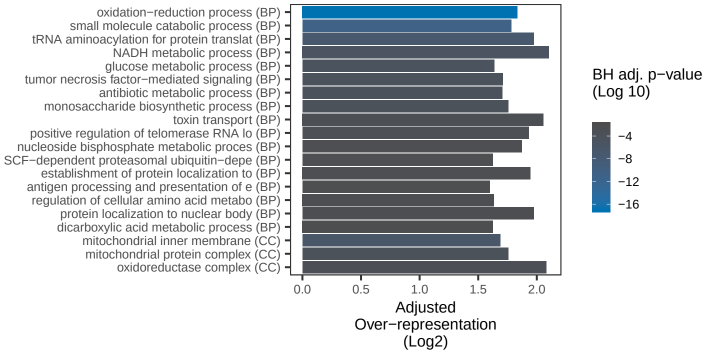


```
##
## $data
##      category over_represented_pvalue under_represented_pvalue numDEInCat
## 1  GO:0044281            4.028782e-24                1.0000000         63
## 2  GO:0055114            1.958873e-19                1.0000000         32
## 3  GO:0006091            1.414456e-13                1.0000000         27
## 4  GO:0017144            5.526715e-12                1.0000000         30
## 5  GO:0051186            1.245389e-06                0.9999997         21
## 6  GO:1901137            1.910455e-06                0.9999996         20
## 7  GO:1901998            2.425164e-06                0.9999999          7
## 8  GO:0030258            4.666279e-06                0.9999997          8
## 9  GO:0005975            1.073440e-05                0.9999976         17
## 10 GO:0008202            1.109125e-05                0.9999987         10
## 11 GO:0072522            1.876277e-05                0.9999961         15
## 12 GO:0044429            3.100699e-19                1.0000000         41
## 13 GO:1990204            1.136036e-05                0.9999993          7
## 14 GO:0016491            6.280439e-11                1.0000000         25
## 15 GO:0004812            5.568684e-08                1.0000000         11
## 16 GO:0048037            3.662326e-07                0.9999999         18
## 17 GO:0016747            1.009188e-06                0.9999999          9
##    numInCat
## 1       310
## 2        83
## 3        92
## 4       128
## 5       115
## 6       107
## 7        10
## 8        21
## 9        89
## 10       35
## 11       71
## 12      159
## 13       16
## 14       95
## 15       24
## 16       85
## 17       23
##                                                                            term
## 1                                              small molecule metabolic process
## 2                                                   oxidation-reduction process
## 3                                generation of precursor metabolites and energy
## 4                                                        drug metabolic process
## 5                                                    cofactor metabolic process
## 6                                  carbohydrate derivative biosynthetic process
## 7                                                               toxin transport
## 8                                                            lipid modification
## 9                                                carbohydrate metabolic process
## 10                                                    steroid metabolic process
## 11                              purine-containing compound biosynthetic process
## 12                                                           mitochondrial part
## 13                                                       oxidoreductase complex
## 14                                                      oxidoreductase activity
## 15                                               aminoacyl-tRNA ligase activity
## 16                                                             cofactor binding
## 17 transferase activity, transferring acyl groups other than amino-acyl groups
##    ontology           BH adj_over_rep
## 1        BP 4.769272e-20     3.494409
## 2        BP 1.159457e-15     6.204881
## 3        BP 1.860482e-10     5.018165
## 4        BP 5.947750e-09     3.918863
## 5        BP 2.541882e-04     3.113399
## 6        BP 3.627390e-04     3.145541
## 7        BP 4.284939e-04     8.233090
## 8        BP 7.672140e-04     7.426975
## 9        BP 1.588422e-03     3.177366
## 10       BP 1.606653e-03     5.153704
## 11       BP 2.524019e-03     3.335070
## 12       CC 1.223536e-15     4.640499
## 13       CC 1.620289e-03     7.800007
## 14       MF 5.310560e-08     4.245300
## 15       MF 2.354360e-05     7.172311
## 16       MF 9.224386e-05     3.832178
## 17       MF 2.203678e-04     7.483026
##                                    term_for_plot
## 1          small molecule metabolic process (BP)
## 2               oxidation-reduction process (BP)
## 3  generation of precursor metabolites and  (BP)
## 4                    drug metabolic process (BP)
## 5                cofactor metabolic process (BP)
## 6  carbohydrate derivative biosynthetic pro (BP)
## 7                           toxin transport (BP)
## 8                        lipid modification (BP)
## 9            carbohydrate metabolic process (BP)
## 10                steroid metabolic process (BP)
## 11 purine-containing compound biosynthetic  (BP)
## 12                       mitochondrial part (CC)
## 13                   oxidoreductase complex (CC)
## 14                  oxidoreductase activity (MF)
## 15           aminoacyl-tRNA ligase activity (MF)
## 16                         cofactor binding (MF)
## 17 transferase activity, transferring acyl  (MF)
```

```
lm_over_rep_go %>% filter(BH<0.01)  %>%
  addAdjustedOverRep(lm_pwf, sapiens.go.full) %>%
  filter(adj_over_rep>3) %>%
  remove_redundant_GO_terms() %>%
  plotTerms(horizontal=TRUE, shorten_term=TRUE, plot_top=20)
```

```
## 'select()' returned 1:1 mapping between keys and columns
```

```
## $p
```



```
##
## $data
##      category over_represented_pvalue under_represented_pvalue numDEInCat
## 1  GO:0055114            4.699760e-21                1.0000000         52
## 2  GO:0044282            2.495342e-14                1.0000000         37
## 3  GO:0006418            2.455402e-10                1.0000000         20
## 4  GO:0006734            6.737147e-08                1.0000000         12
## 5  GO:0006006            1.455207e-07                1.0000000         21
## 6  GO:0033209            4.392061e-07                0.9999999         18
## 7  GO:0016999            5.618775e-07                0.9999999         18
## 8  GO:0046364            6.939182e-07                0.9999999         16
## 9  GO:1901998            6.388143e-06                0.9999998          9
## 10 GO:1904874            7.699857e-06                0.9999996         10
## 11 GO:0033865            1.516983e-05                0.9999981         12
## 12 GO:0031146            2.457018e-05                0.9999961         14
## 13 GO:0070199            2.482724e-05                0.9999988          9
## 14 GO:0002479            2.993548e-05                0.9999951         14
## 15 GO:0006521            3.676311e-05                0.9999945         13
## 16 GO:1903405            3.785542e-05                0.9999989          8
## 17 GO:0043648            5.548052e-05                0.9999909         13
## 18 GO:0005743            2.561231e-09                1.0000000         26
## 19 GO:0098798            1.044075e-06                0.9999998         17
## 20 GO:1990204            3.925597e-06                0.9999997         11
##    numInCat
## 1        83
## 2        62
## 3        26
## 4        13
## 5        38
## 6        31
## 7        32
## 8        25
## 9        10
## 10       12
## 11       21
## 12       26
## 13       11
## 14       26
## 15       23
## 16        9
## 17       25
## 18       48
## 19       32
## 20       16
##                                                                                               term
## 1                                                                      oxidation-reduction process
## 2                                                                 small molecule catabolic process
## 3                                                      tRNA aminoacylation for protein translation
## 4                                                                           NADH metabolic process
## 5                                                                        glucose metabolic process
## 6                                                 tumor necrosis factor-mediated signaling pathway
## 7                                                                     antibiotic metabolic process
## 8                                                              monosaccharide biosynthetic process
## 9                                                                                  toxin transport
## 10                               positive regulation of telomerase RNA localization to Cajal body
## 11                                                       nucleoside bisphosphate metabolic process
## 12                        SCF-dependent proteasomal ubiquitin-dependent protein catabolic process
## 13                                            establishment of protein localization to chromosome
## 14 antigen processing and presentation of exogenous peptide antigen via MHC class I, TAP-dependent
## 15                                            regulation of cellular amino acid metabolic process
## 16                                                            protein localization to nuclear body
## 17                                                             dicarboxylic acid metabolic process
## 18                                                                    mitochondrial inner membrane
## 19                                                                   mitochondrial protein complex
## 20                                                                          oxidoreductase complex
##    ontology           BH adj_over_rep
## 1        BP 9.272627e-18     3.566940
## 2        BP 1.969324e-11     3.440306
## 3        BP 7.855960e-08     3.928257
## 4        BP 1.035771e-05     4.298834
## 5        BP 1.980085e-05     3.111141
## 6        BP 5.147843e-05     3.268283
## 7        BP 6.521084e-05     3.260468
## 8        BP 7.975343e-05     3.377015
## 9        BP 5.817141e-04     4.160712
## 10       BP 6.858492e-04     3.813483
## 11       BP 1.221635e-03     3.662603
## 12       BP 1.825496e-03     3.083310
## 13       BP 1.825496e-03     3.853160
## 14       BP 2.134797e-03     3.026172
## 15       BP 2.441537e-03     3.105898
## 16       BP 2.441537e-03     3.927636
## 17       BP 3.333900e-03     3.086860
## 18       CC 6.890877e-07     3.221653
## 19       CC 1.166015e-04     3.376489
## 20       CC 3.809116e-04     4.232450
##                                       term_for_plot
## 1              oxidation-reduction process (BP)
## 2         small molecule catabolic process (BP)
## 3  tRNA aminoacylation for protein translat (BP)
## 4                    NADH metabolic process (BP)
## 5                 glucose metabolic process (BP)
## 6  tumor necrosis factor-mediated signaling (BP)
## 7              antibiotic metabolic process (BP)
## 8       monosaccharide biosynthetic process (BP)
## 9                           toxin transport (BP)
## 10 positive regulation of telomerase RNA lo (BP)
## 11 nucleoside bisphosphate metabolic proces (BP)
## 12 SCF-dependent proteasomal ubiquitin-depe (BP)
## 13 establishment of protein localization to (BP)
## 14 antigen processing and presentation of e (BP)
## 15 regulation of cellular amino acid metabo (BP)
## 16     protein localization to nuclear body (BP)
## 17      dicarboxylic acid metabolic process (BP)
## 18             mitochondrial inner membrane (CC)
## 19            mitochondrial protein complex (CC)
## 20                   oxidoreductase complex (CC)
```

Ok, so now we have a reasonable list of over-represented GO terms and some pretty plots to show off our results…

Task: Modify the plotTerms function so that it makes a separate plot for each ontology

One thing you may have noticed in the above is that the over-represented terms are slightly different for `lm` and `limma` above. For example, "tRNA aminoacylation for protein translation" is one of the most significantly overrepresented MF GO term in the `lm` proteins but not present in the `limma` over-represented GO terms (although "aminoacyl-tRNA ligase activity" is there in the MF terms)

Below, we take the GO term "tRNA aminoacylation for protein translation". Note that very few of the proteins annotated with "tRNA aminoacylation for protein translation" are detected as having a significant change in RNA binding according to `limma` - 11/26 vs 20/26 for `lm`!!

```
lm_over_rep_go %>% filter(term=="tRNA aminoacylation for protein translation") %>%
  addAdjustedOverRep(lm_pwf, sapiens.go.full)  %>%
  head(10)
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0006418            2.455402e-10                        1         20
##   numInCat                                     term ontology
## 1       26 tRNA aminoacylation for protein translation       BP
##            BH adj_over_rep
## 1 7.85596e-08     3.928257
```

```
limma_over_rep_go %>% filter(term=="tRNA aminoacylation for protein translation") %>%
  addAdjustedOverRep(lm_pwf, sapiens.go.full)  %>%
  head(10)
```

```
##     category over_represented_pvalue under_represented_pvalue numDEInCat
## 1 GO:0006418            2.046793e-07                        1         11
##   numInCat                                     term ontology
## 1       26 tRNA aminoacylation for protein translation       BP
##             BH adj_over_rep
## 1 6.376299e-05     2.160541
```

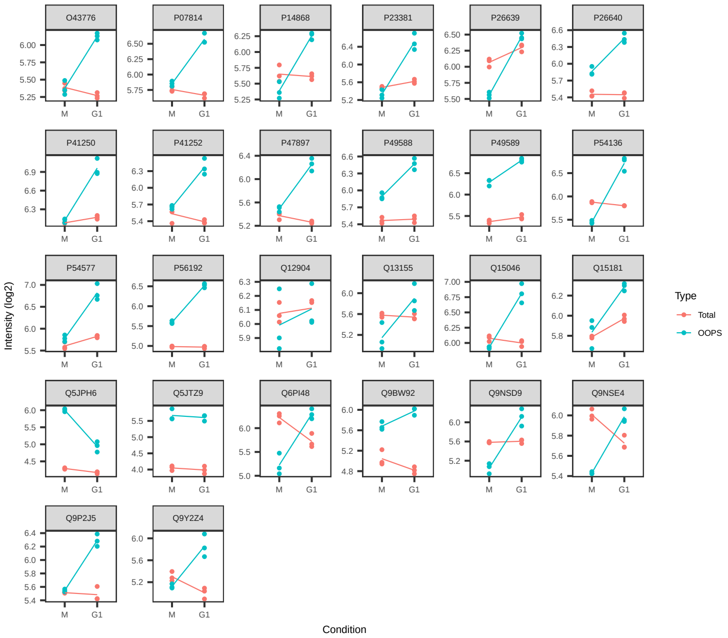
We want to take a look at the intensity values, so let's make that `plotIntensities()` function again

```
combined_intensities <- readRDS("../results/combined_intensities.rds")

plotIntensities <- function(obj){

  p <- tidy(obj, addPheno=TRUE) %>%
    ggplot(aes(Condition, value, colour=Type, group=Type)) +
    geom_point() +
    stat_summary(geom="line", fun.y=mean) +
    facet_wrap(~gene, scales='free') +
    ylab("Intensity (log2)")

  invisible(p)
}
```

If we take a look at the intensity values for these 26 proteins annotated with "tRNA aminoacylation for protein translation", most do seem to have a clear increase in RNA binding. Note that not all of these are cytosolic tRNA ligases. Q5JPH6 + Q5JTZ9 are mitchondrial tRNA ligase. Q15181=Inorganic pyrophosphatase and Q12904 & Q13155 interact with the Aminoacyl tRNA synthase complex.

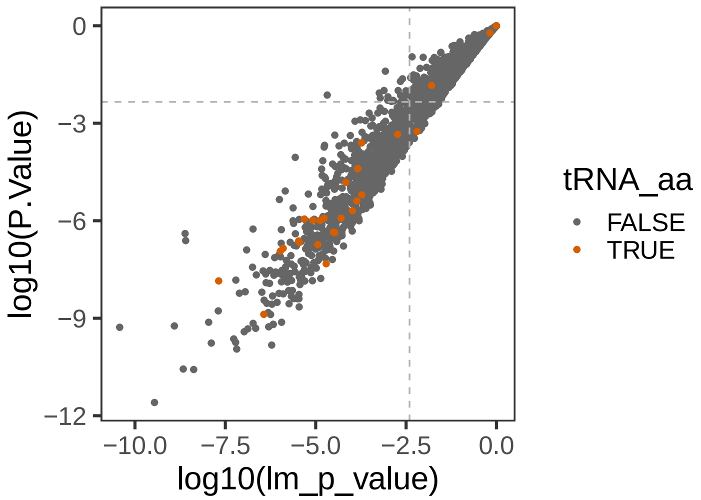
So, why does `limma` only detect 11/26 of these as increasing RNA binding given the apparent signficiant changes…

```
tRNA_aa <- sapiens.go.full %>%
  filter(TERM=="tRNA aminoacylation for protein translation") %>%
  pull(UNIPROTKB)

tRNA_aa_intensities <- combined_intensities[intersect(rownames(combined_intensities), tRNA_aa),] %>%
  plotIntensities()

print(tRNA_aa_intensities + theme(text=element_text(size=10)))
```



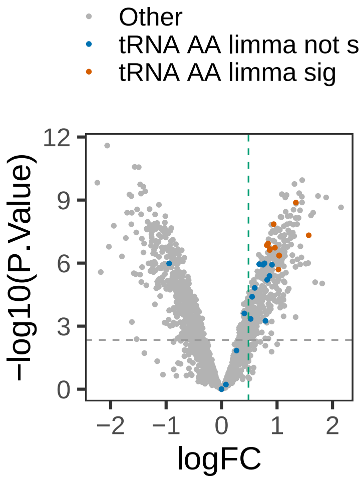
Next, we compare the p-values again. Note that the four proteins not in the bottom left quadrant (e.g <1% FDR with both `lm` and `limma`) are the two Mt tRNA-ligases and the two aminoacyl tRNA synthase complex interactors.

```
max_p_sig_lm <- compare_methods %>% filter(lm_BH<0.01) %>% pull(lm_p_value) %>% max()
max_p_sig_limma <- compare_methods %>% filter(adj.P.Val<0.01) %>% pull(P.Value) %>% max()

compare_methods %>%
  mutate(tRNA_aa=Row.names %in% tRNA_aa) %>%
  arrange(tRNA_aa) %>%
  ggplot(aes(x=log10(lm_p_value), y=log10(P.Value), colour=tRNA_aa)) +
  geom_point() +
  scale_colour_manual(values=c("grey40", cbPalette[6])) +
  geom_vline(xintercept=log10(max_p_sig_lm), linetype=2, colour="grey70") +
  geom_hline(yintercept=log10(max_p_sig_limma), linetype=2,colour="grey70")
```



Finally, we look at the point estimates for the fold change. The green dashed lines represent our 1.4-fold threshold. The dashed grey-line is the threshold for the BH adjusted p-value being < 0.01.

```
max_p_sig_limma <- compare_methods %>% filter(adj.P.Val<0.01) %>% pull(P.Value) %>% max()

compare_methods %>%
  mutate(tRNA_aa=Row.names %in% tRNA_aa) %>%
  mutate(limma_sig=Row.names %in% rownames(limma_rna_binding_changes)) %>%
  mutate(tRNA_aa_limma=recode(interaction(tRNA_aa, limma_sig),
                              "TRUE.TRUE"="tRNA AA limma sig",
                              "TRUE.FALSE"="tRNA AA limma not sig",
                              "FALSE.TRUE"="Other",
                              "FALSE.FALSE"="Other")) %>%
  arrange(tRNA_aa) %>%
  ggplot(aes(logFC, -log10(P.Value), colour=tRNA_aa_limma)) +
  geom_point(size=1) +
  scale_colour_manual(values=c("grey70", cbPalette[5], cbPalette[6]), name="") +
  geom_vline(xintercept=log2(1.4), linetype=2, colour=cbPalette[3]) +
  geom_hline(yintercept=-log10(max_p_sig_limma), linetype=2, colour="grey60") +
  theme(legend.position="top", legend.direction=2)
```



Question: Why are there proteins with p-values less than the FDR threshold (above the grey line) and >1.4 fold change but not identified as significant with limma?

This demonstrates the downsides of applying a thresholds to the confidence interval for the estimated fold change when we don't know what fold change is biological relevant. These proteins were subsequently shown to have a consistent (if still slight) change in RNA binding in a Thymidine + Nocadazole experiment suggesting this is probably a real change in RNA binding. So, while a threshold on the log fold change is a very sensible approach, be careful about what threshold you use!

And that's the end of the workshop!

Task: Repeat the above analysis but for proteins with a decrease in RNA binding with either 'lm' or 'limma'. How would you interpret this set of GO terms?