03: Changes in RNA Binding

Tom Smith, Veronica Dezi, Manasa Ramakrishna

September 06, 2019

Contents

1. Introduction

In this section of the code, we are finally doing the interesting analysis which is finding out if there are any RBPs that are differentially expressed between conditions. Having looked at the data thus far, the extreme variability of the RBP Unstarved vs Starved samples might mean that we cannot really do a differential analysis with that set. However, we'll give it a go and see what happen.

2. Reading in normalised, outlier-free data

We start by reading in the normalised data minus samples with really small MADs. From a starting point of 20 samples, we are now down to 16.

```
total_ni_protein_quant <- readRDS("../results/total_ni_res_pro_agg_norm_nododgey.rds")
oops_ni_protein_quant <- readRDS("../results/rbp_ni_res_pro_agg_norm_nododgey.rds")

total_us_protein_quant <- readRDS("../results/total_us_res_pro_agg_norm_nododgey.rds")
oops_us_protein_quant <- readRDS("../results/rbp_us_res_pro_agg_norm_nododgey.rds")
```

3. LIMMA for differential protein expression analysis

LIMMA stands for Linear Models for Microarray and RNA-Seq Data and is a package used for the analysis of gene expression data from microarrays or RNAseq experiments. It's major selling point is that it is able to use linear models to assess differential expression in the context of multifactor designed experiments. Rather usefully, limma does distinguish data to be "from proteins" or "from RNA" which makes it quite handy to apply to Proteomics data.There are a few steps to DE analysis by limma.
1. Create a data matrix with samples in columns and proteins in rows. We can use the "exprs" slot in an MSnSet for this. 2. Create a design matrix that tells limma about samples, conditions and replicates. We can use the `pData` from MSnSet for this.
3. Fit a linear model to the data(1) using the design(2).
4. Define contrasts of interest i.e which gruops of samples you want to test for differential protein expression.
5. Extract results for the contrast of interest.
6. Look at the top proteins.

Initially, we perform this analysis for each of the 4 datasets separately.

3a. The Design matrix

Given we've removed various samples from each of the experiments, we need different design matrices for each of them.

```
# Extract information from the pData slot and make
# it suitable for design matrix formation

# NI_Total
setup_ni_tot = data.frame(cbind(Condition = sapply(strsplit(pData(total_ni_protein_quant)$Sample_name,
    "_"), "[[", 1), Replicate = sapply(strsplit(pData(total_ni_protein_quant)$Sample_name,
    "_"), "[[", 2)))
rownames(setup_ni_tot) = rownames(pData(total_ni_protein_quant))
setup_ni_tot$Condition = factor(setup_ni_tot$Condition,
    levels = c("4hr-Starved", "30min-Insulin"))

# NI_RBP
setup_ni_rbp = data.frame(cbind(Condition = sapply(strsplit(pData(oops_ni_protein_quant)$Sample_name,
    "_"), "[[", 1), Replicate = sapply(strsplit(pData(oops_ni_protein_quant)$Sample_name,
    "_"), "[[", 2)))
rownames(setup_ni_rbp) = rownames(pData(oops_ni_protein_quant))
setup_ni_rbp$Condition = factor(setup_ni_rbp$Condition,
    levels = c("4hr-Starved", "30min-Insulin"))

# US_Total
setup_us_tot = data.frame(cbind(Condition = sapply(strsplit(pData(total_us_protein_quant)$Sample_name,
    "_"), "[[", 1), Replicate = sapply(strsplit(pData(total_us_protein_quant)$Sample_name,
    "_"), "[[", 2)))
rownames(setup_us_tot) = rownames(pData(total_us_protein_quant))
setup_us_tot$Condition = factor(setup_us_tot$Condition,
    levels = c("Unstarved", "4hr-Starved"))

# US_RBP
setup_us_rbp = data.frame(cbind(Condition = sapply(strsplit(pData(oops_us_protein_quant)$Sample_name,
    "_"), "[[", 1), Replicate = sapply(strsplit(pData(oops_us_protein_quant)$Sample_name,
    "_"), "[[", 2)))
rownames(setup_us_rbp) = rownames(pData(oops_us_protein_quant))
setup_us_rbp$Condition = factor(setup_us_rbp$Condition,
    levels = c("Unstarved", "4hr-Starved"))

# Create-design-matrix

des_ni_tot <- make_design_matrix(setup_ni_tot)
```

```
##      (Intercept) 30min-Insulin
## 127N           1             0
## 127C           1             0
## 128N           1             0
## 128C           1             0
## 129N           1             1
## 129C           1             1
## 130N           1             1
## 130C           1             1
## 131            1             1
## attr(,"assign")
```

```
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$`setup$Condition`
## [1] "contr.treatment"
```

```
des_ni_rbp <- make_design_matrix(setup_ni_rbp)
```

```
##      (Intercept) 30min-Insulin
## 126            1             0
## 127N           1             0
## 127C           1             0
## 128N           1             0
## 128C           1             0
## 129N           1             1
## 129C           1             1
## 130N           1             1
## 131            1             1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$`setup$Condition`
## [1] "contr.treatment"
```

```
des_us_tot <- make_design_matrix(setup_us_tot)
```

```
##      (Intercept) 4hr-Starved
## 126            1           0
## 127N           1           0
## 127C           1           0
## 128N           1           0
## 128C           1           0
## 129N           1           1
## 129C           1           1
## 130N           1           1
## 131            1           1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
## attr(,"contrasts")$`setup$Condition`
## [1] "contr.treatment"
```

```
des_us_rbp <- make_design_matrix(setup_us_rbp)
```

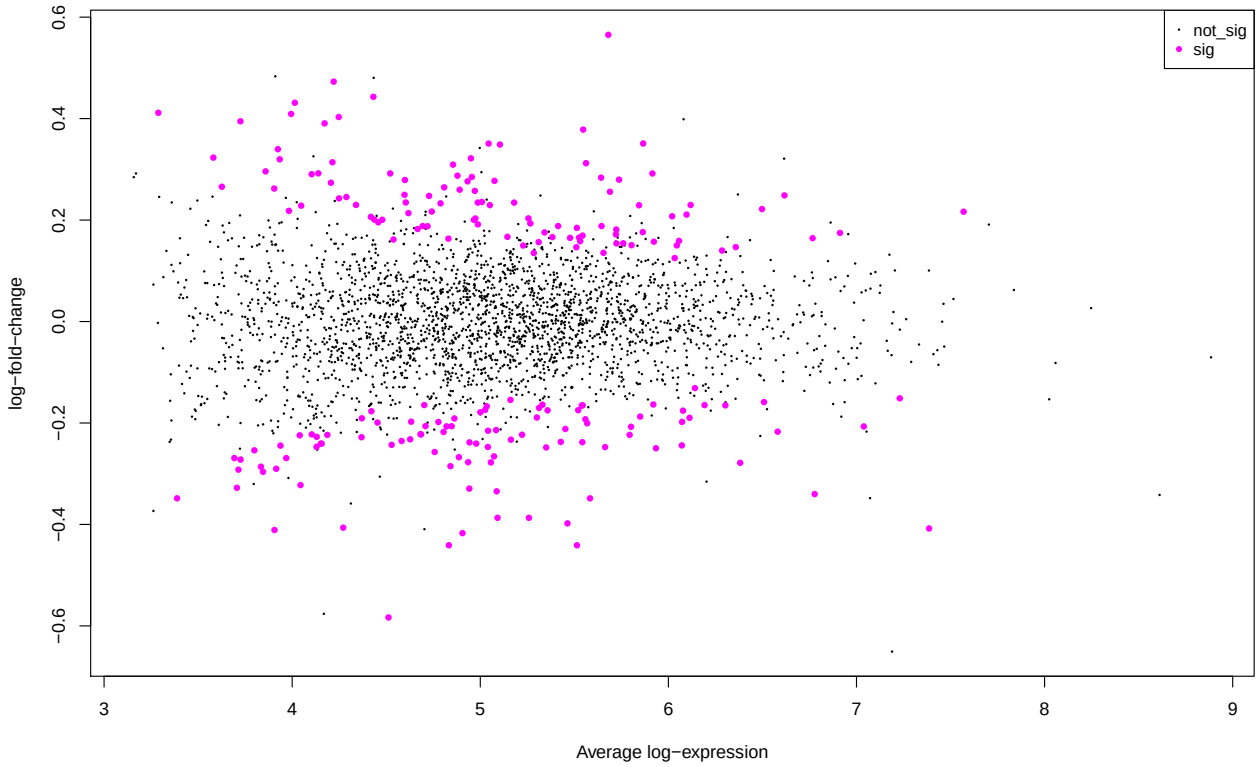
```
##      (Intercept) 4hr-Starved
## 126            1           0
## 127N           1           0
## 127C           1           0
## 128N           1           0
## 128C           1           0
## 129N           1           1
## 129C           1           1
## 130N           1           1
## 131            1           1
## attr(,"assign")
## [1] 0 1
## attr(,"contrasts")
```

```
## attr(,"contrasts")$`setup$Condition`
## [1] "contr.treatment"
```

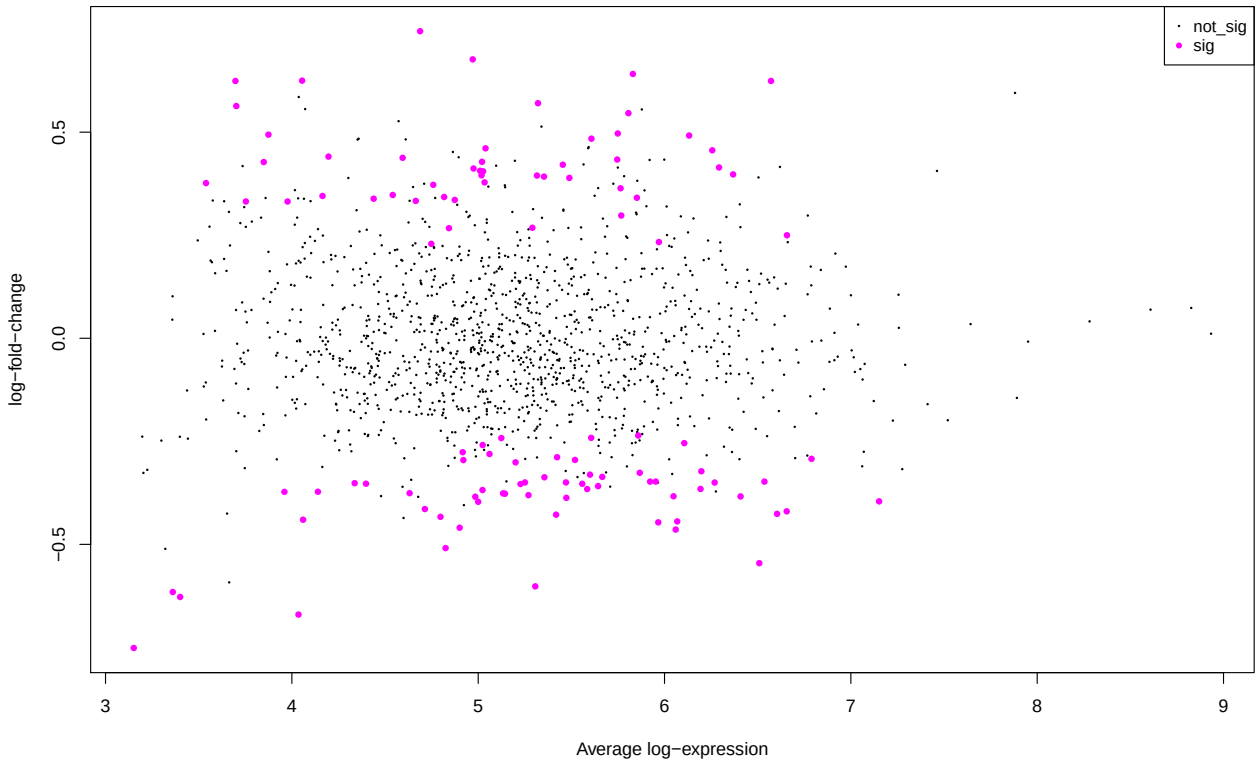
3b. Linear model and contrasts

Below we run limma to idenify the proteins with a significant change in abundance between conditions, one experiment at a time using the function 'simple_limma'. Only 6 RNA binding-proteins were significantly DE and in the 4hr-Starved samples relative to the Unstarved samples. Since the US RBP experiment had extremely variable data, I wouldn't be confident in pursuing any of these any further.

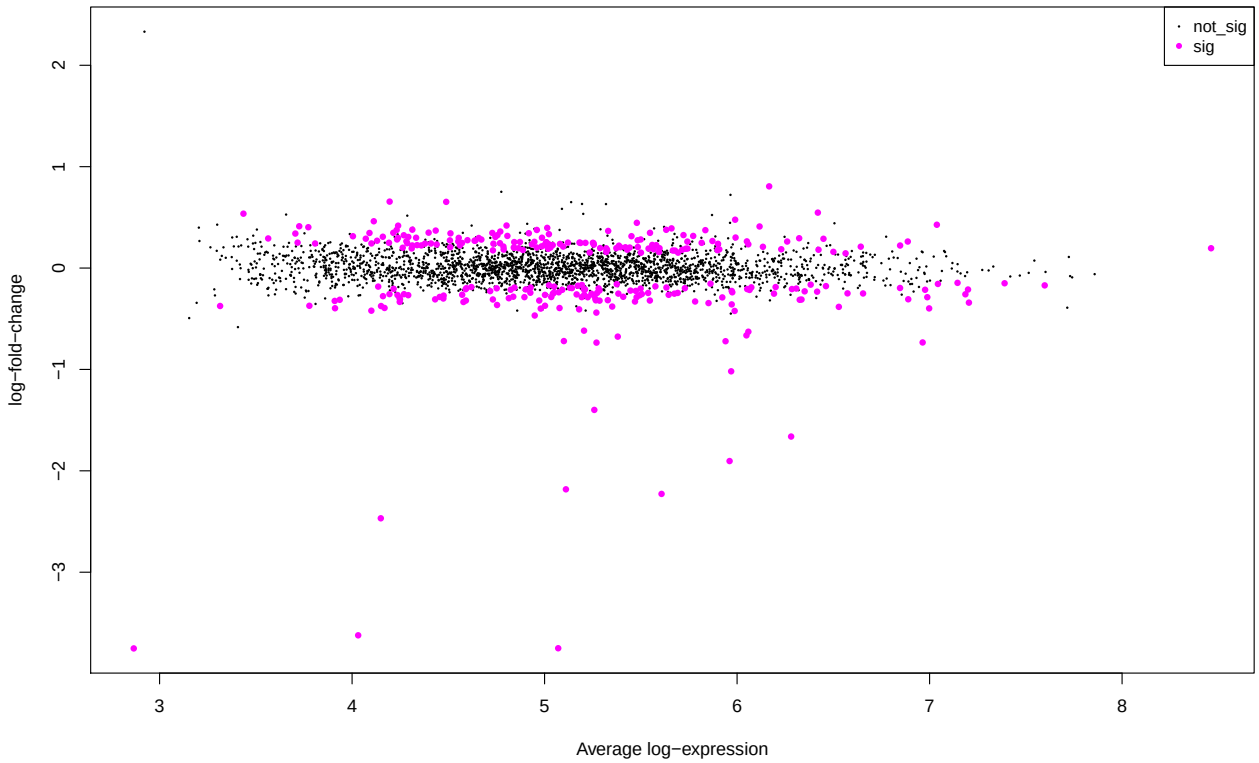
```
lm_tot_ni <- simple_limma(total_ni_protein_quant, des_ni_tot,
    "30min-Insulin")
```



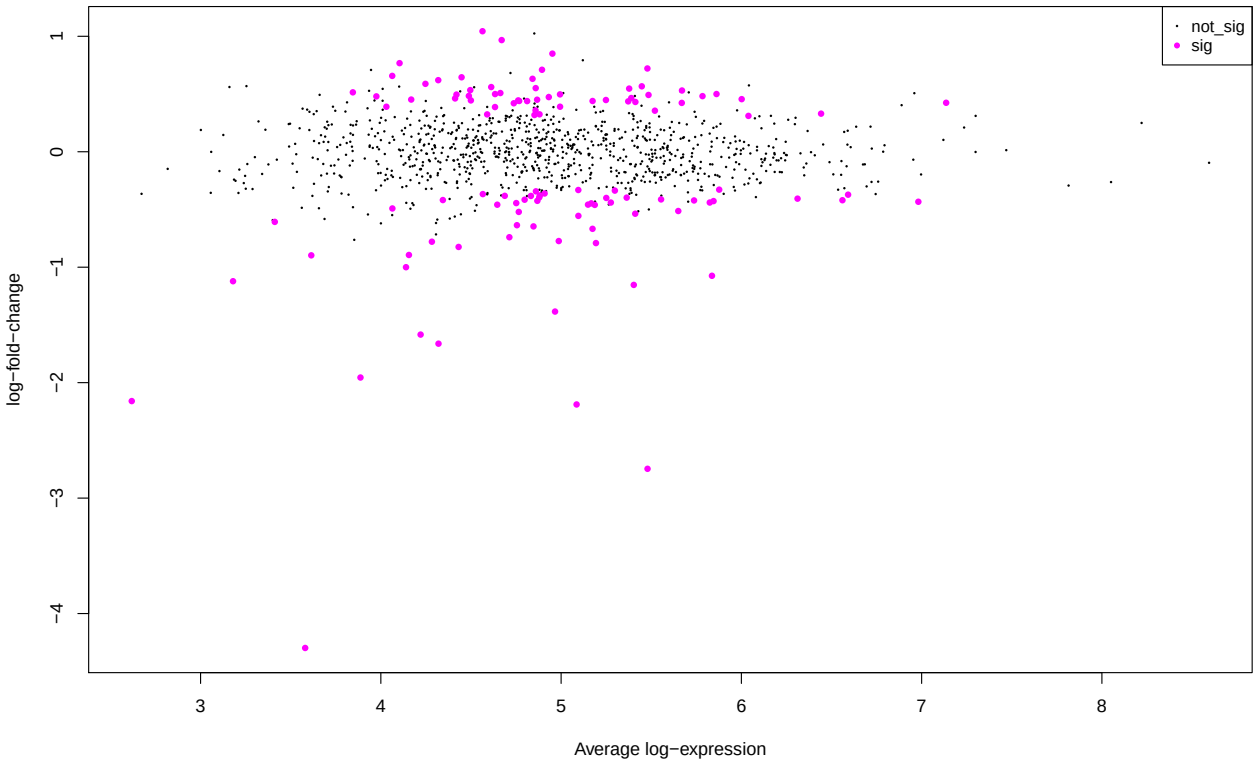

```
lm_rbp_ni <- simple_limma(oops_ni_protein_quant, des_ni_rbp,
    "30min-Insulin")
```




```
lm_tot_us <- simple_limma(total_us_protein_quant, des_us_tot,
    "4hr-Starved")
```




```
lm_rbp_us <- simple_limma(oops_us_protein_quant, des_us_rbp,
    "4hr-Starved")
```




```
lm_rbp_us_mod <- cbind(fData(oops_us_protein_quant)[rownames(head(lm_rbp_us)),
    ], head(lm_rbp_us))
lm_rbp_us_mod <- lm_rbp_us_mod[, c(14:15, 67, 82:87)]
lm_rbp_us_mod$protein_desc = sapply(strsplit(lm_rbp_us_mod$Master.Protein.Descriptions,
    "OS="), "[[", 1)
lm_rbp_us_mod$gene_name = sapply(strsplit(lm_rbp_us_mod$Master.Protein.Descriptions,
    "GN=|PE="), "[[", 2)
colnames(lm_rbp_us_mod)[1] = "uniprot_id"
lm_rbp_us_mod = lm_rbp_us_mod[, c(1, 11, 10, 4:9)]

# Write results to table
write.table(lm_tot_ni, row.names = T, quote = F, file = "../results/Untreated_vs_Insulin_ALL_Total-prote
write.table(lm_rbp_ni, row.names = T, quote = F, file = "../results/Untreated_vs_Insulin_ALL_RNA_binding

write.table(lm_tot_us, row.names = T, quote = F, file = "../results/Unstarved_vs_Starved_ALL_Total-prote
write.table(lm_rbp_us, row.names = T, quote = F, file = "../results/Unstarved_vs_Starved_ALL_RNA_binding
write.table(lm_rbp_us_mod, row.names = F, quote = F,
    file = "../results/Unstarved_vs_Starved_SIGNIF_RNA_binding_changes.tsv")
```

4. An alternate approach - combining total and RBP data

OK, so it's easy to perform the pairwise comparison. What about changes in RNA binding? For this, we need combine the two MSnSets into a single ExpressionSet. We start by intersecting proteins within the NI or US experiments so we can compare just those proteins that are captures across both total and RBP datasets.

```
intersecting_ni_proteins <- intersect(rownames(total_ni_protein_quant),
    rownames(oops_ni_protein_quant))
print(paste("Number of RBPs also captured in the Total Proteome for Non-Insulin vs Insulin-treated sampl
    length(intersecting_ni_proteins)), sep = "")
```

```
## [1] "Number of RBPs also captured in the Total Proteome for Non-Insulin vs Insulin-treated samples is
```

```
intersecting_us_proteins <- intersect(rownames(total_us_protein_quant),
    rownames(oops_us_protein_quant))
print(paste("Number of RBPs also captured in the Total Proteome for Unstarved vs Starved samples is",
    length(intersecting_us_proteins)), sep = "")
```

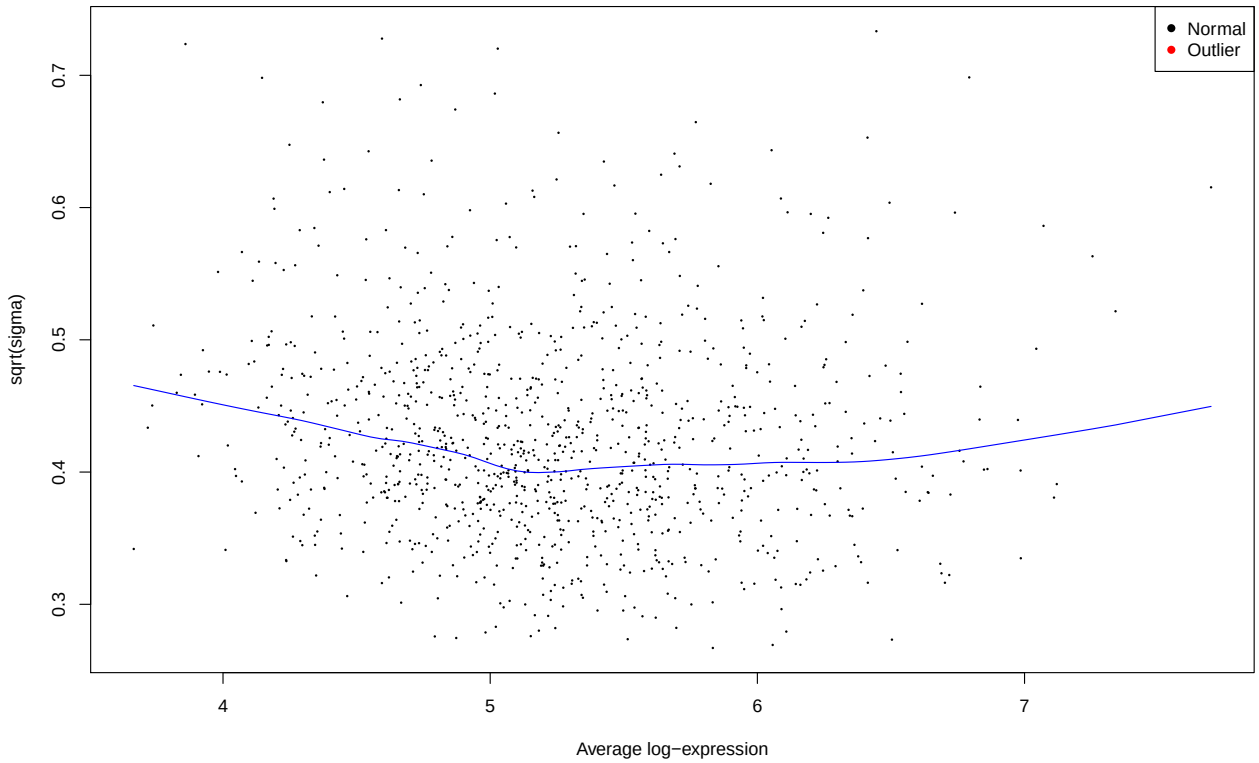
```
## [1] "Number of RBPs also captured in the Total Proteome for Unstarved vs Starved samples is 730"
```

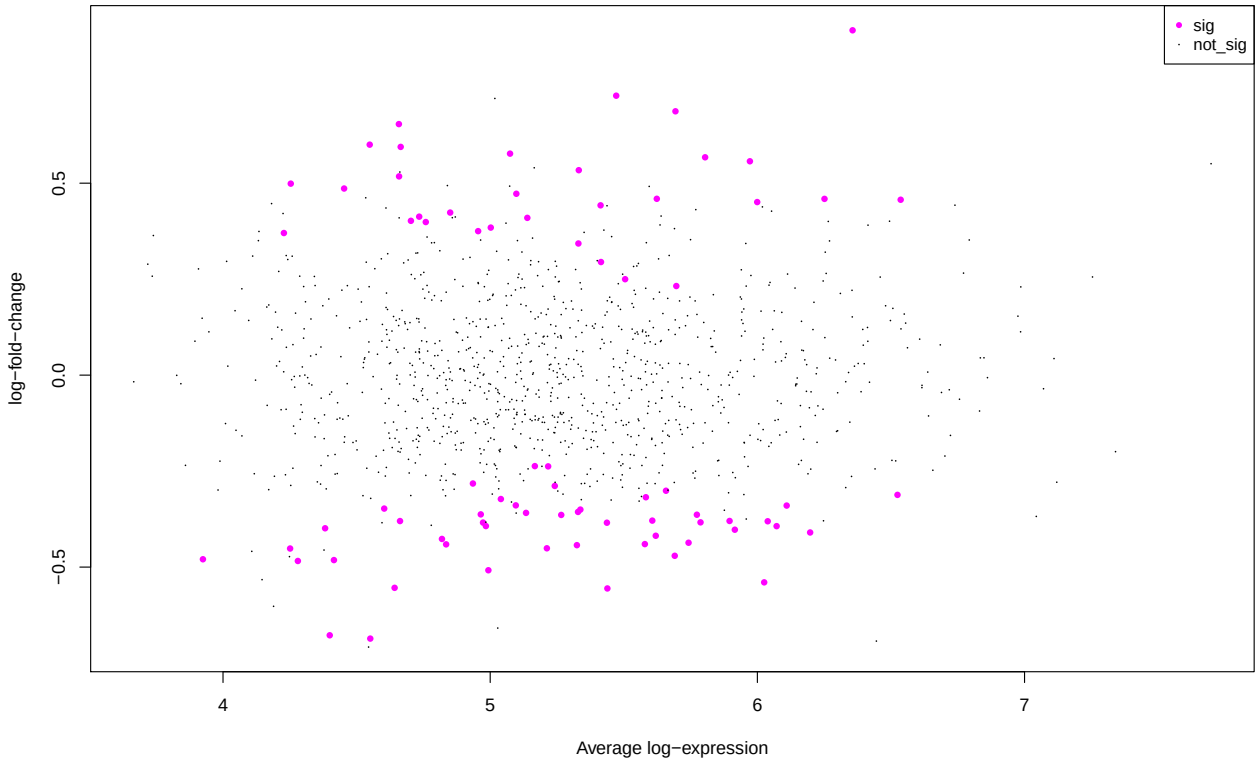
```
# Subset of intersecting NI proteins only
total_ni_for_combination <- total_ni_protein_quant[intersecting_ni_proteins,
    ]
rbp_ni_for_combination <- oops_ni_protein_quant[intersecting_ni_proteins,
    ]

# Subset of intersecting US proteins only
total_us_for_combination <- total_us_protein_quant[intersecting_us_proteins,
    ]
rbp_us_for_combination <- oops_us_protein_quant[intersecting_us_proteins,
    ]
```

```
combined_ni_intensities = combine_esets(total_ni_for_combination,
    rbp_ni_for_combination)
pData(combined_ni_intensities)$Condition = factor(pData(combined_ni_intensities)$Condition,
    levels = c("4hr-Starved", "30min-Insulin"))

combined_us_intensities = combine_esets(total_us_for_combination,
    rbp_us_for_combination)
pData(combined_us_intensities)$Condition = factor(pData(combined_us_intensities)$Condition,
    levels = c("Unstarved", "4hr-Starved"))
```

The we run `limma` on the combined intensities and this time test for a signficant interaction coefficient

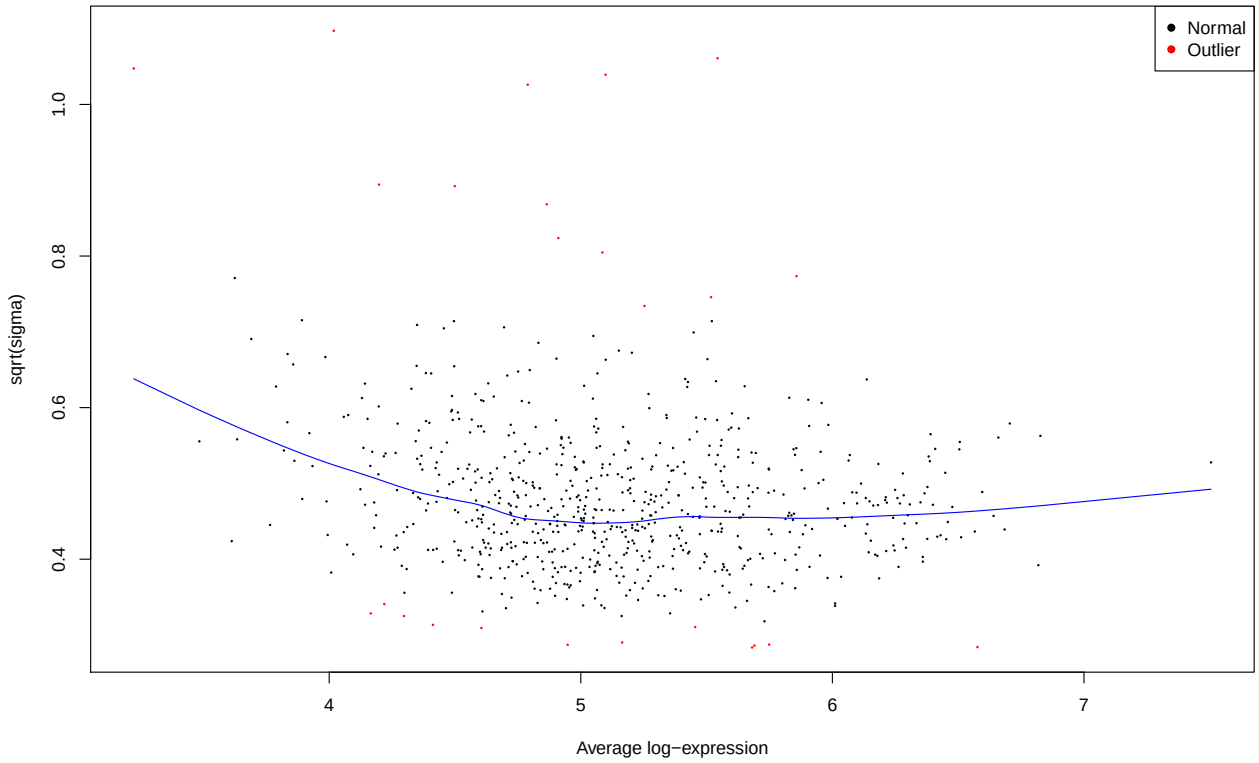
```
# N_I
ni_rbps_de = run_limma(combined_ni_intensities, "condition30min-Insulin:typeOOPS")
```

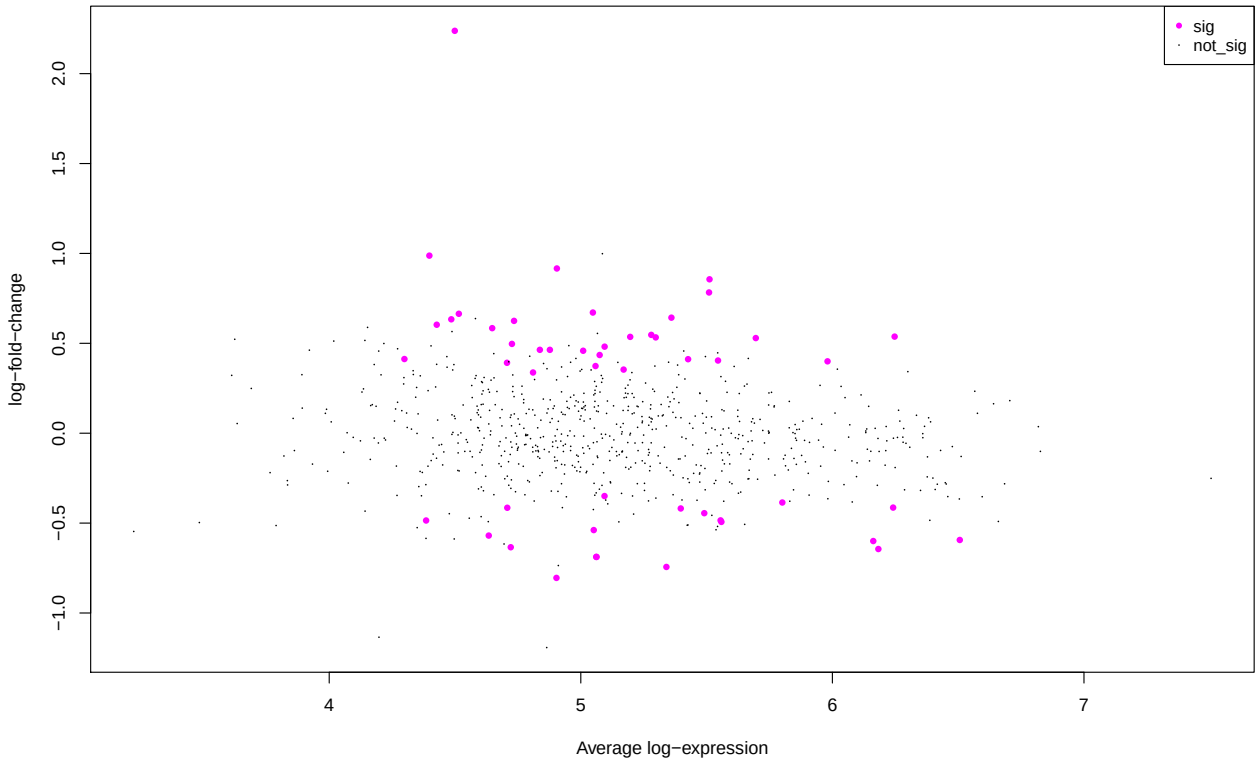





```
ni_rbps_de_mod = modify_output(ni_rbps_de)
ni_rbps_p_value <- ni_rbps_de_mod %>% filter(P.Value <=
    0.05)
write_csv(ni_rbps_p_value, path = "../results/Non-Insulin-vs-Insulin-Treated-rawp-le-0.05.csv")

# U_S
```

```
us_rbps_de = run_limma(combined_us_intensities, "condition4hr-Starved:typeOOPS")
```






```
us_rbps_de_mod = modify_output(us_rbps_de)
us_rbps_p_value <- us_rbps_de_mod %>% filter(P.Value <=
    0.05)
```

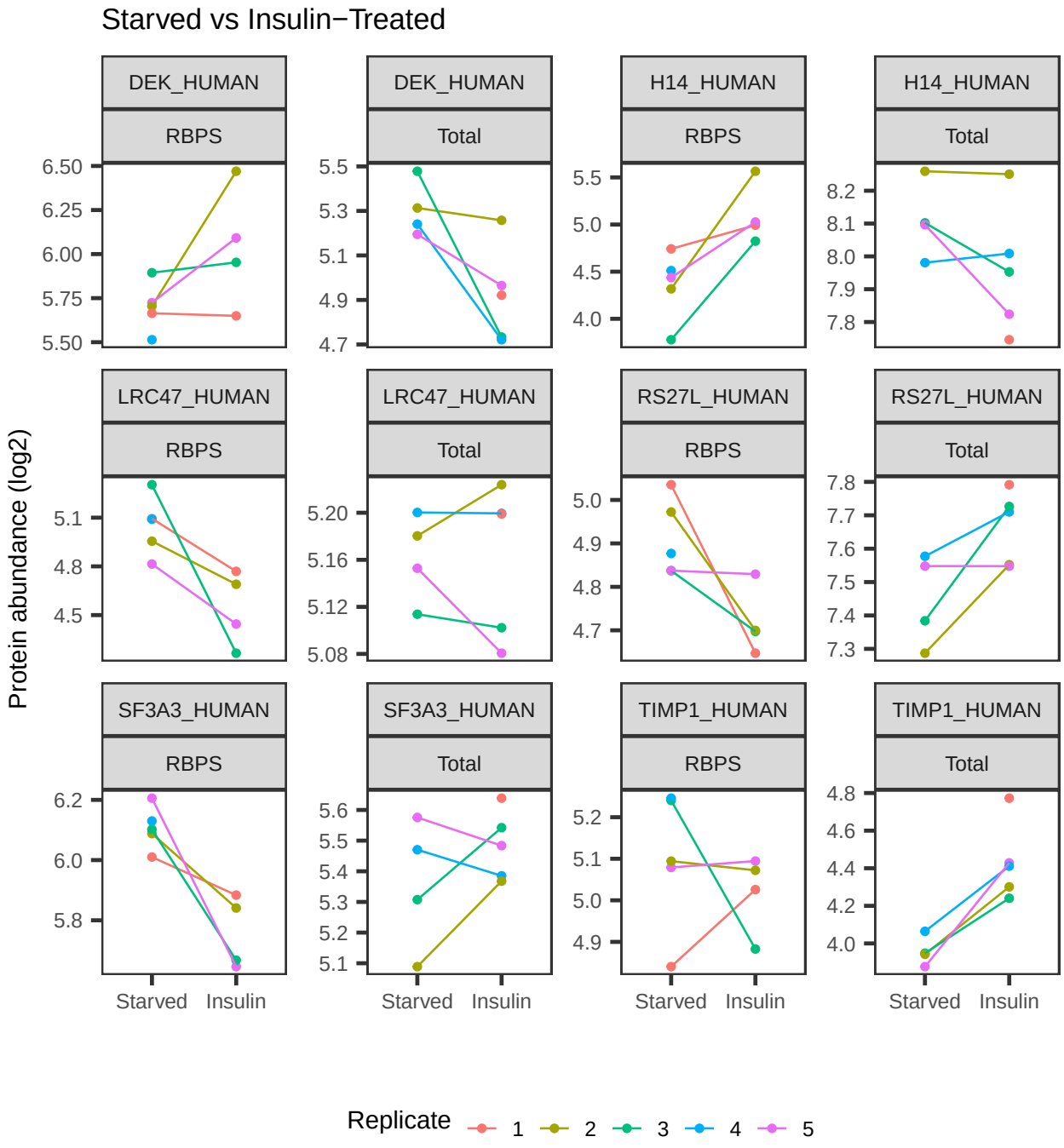
```
write_csv(us_rbps_p_value, path = "../results/Unstarved-vs-Starved-rawp-le-0.05.csv")
```

5. Plotting expression of Top 10 proteins across studies

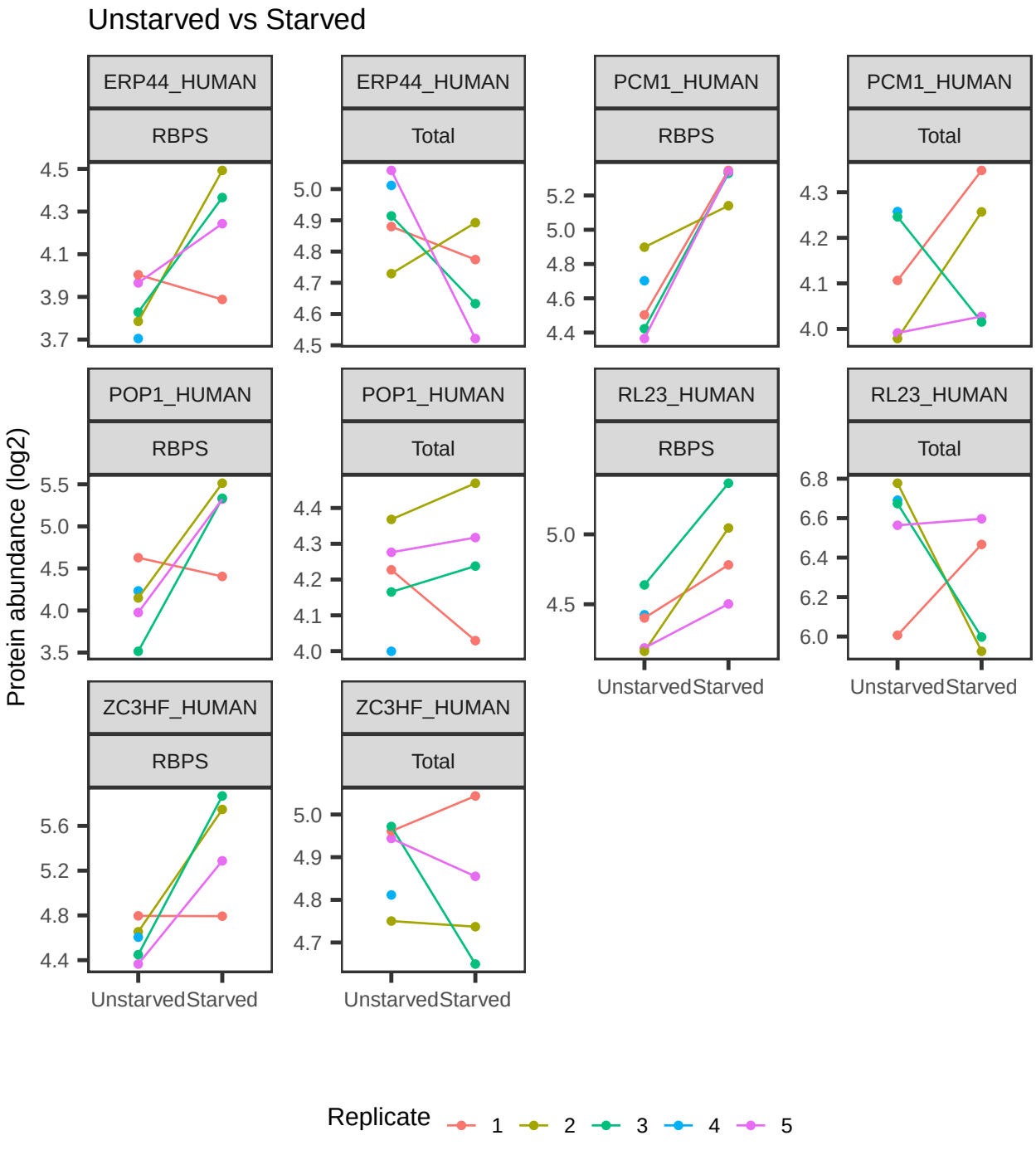
This is to see whether we have a reason for not spotting any DE proteins. Looking at the plots, it is pretty clear that the differences between treatments is pretty minimal i.e the trend lines are not very extreme and only in a few cases are the trend lines really varied between Total proteome and RBP-ome. Given this, I'm not surprised that we don't find many DE genes. To me, it would seem that either (1) the effect of the treatment isn't as strong as we'd hoped for or (2) Something BAAAD happened on the mass spectrometer.

```
# Combining Total and RBP data : NI
total_ni_exprs <- makeLongExprs(total_ni_protein_quant,
    intersecting_ni_proteins)
oops_ni_exprs <- makeLongExprs(oops_ni_protein_quant,
    intersecting_ni_proteins)
combined_ni_exprs <- rbind(cbind(total_ni_exprs, Type = "Total"),
    cbind(oops_ni_exprs, Type = "RBPS"))
combined_ni_exprs$Condition <- factor(combined_ni_exprs$Condition,
    levels = c("X4hr.Starved", "X30min.Insulin"))


# Combining Total and RBP data : US
total_us_exprs <- makeLongExprs(total_us_protein_quant,
    intersecting_us_proteins)
oops_us_exprs <- makeLongExprs(oops_us_protein_quant,
    intersecting_us_proteins)
combined_us_exprs <- rbind(cbind(total_us_exprs, Type = "Total"),
    cbind(oops_us_exprs, Type = "RBPS"))
combined_us_exprs$Condition <- factor(combined_us_exprs$Condition,
    levels = c("Unstarved", "X4hr.Starved"))

# Adding protein information
protein_info <- read.delim("../shared_files/human_protein_ids_plus_gene_names.tsv")
combined_ni_exprs <- combined_ni_exprs %>% merge(protein_info,
    by.x = "UniprotID", by.y = "Entry")
combined_ni_exprs$Condition = gsub("X4hr.|X30min.",
    "", combined_ni_exprs$Condition)
combined_ni_exprs$Condition = factor(combined_ni_exprs$Condition,
    levels = c("Starved", "Insulin"))
combined_us_exprs <- combined_us_exprs %>% merge(protein_info,
    by.x = "UniprotID", by.y = "Entry")
combined_us_exprs$Condition = gsub("X4hr.|X30min.",
    "", combined_us_exprs$Condition)
combined_us_exprs$Condition = factor(combined_us_exprs$Condition,
    levels = c("Unstarved", "Starved"))

# print(head(combined_ni_exprs))
# print(head(combined_us_exprs))

# Data
lowest_p_ni_proteins <- ni_rbps_de_mod %>% arrange(P.Value) %>%
    pull(uniprot_id) %>% head(6)
lowest_p_us_proteins <- us_rbps_de_mod %>% arrange(P.Value) %>%
```

```
    pull(uniprot_id) %>% head(6)

# Plots
plotTop10(combined_ni_exprs, lowest_p_ni_proteins,
    "Starved vs Insulin-Treated")
```




```
plotTop10(combined_us_exprs, lowest_p_us_proteins,
    "Unstarved vs Starved")
```


Unstarved vs Starved
ERP44_HUMAN
RBPS
ERP44_HUMAN
Total
PCM1_HUMAN
RBPS
PCM1_HUMAN
Total
POP1_HUMAN
RBPS
POP1_HUMAN
Total
RL23_HUMAN
RBPS
RL23_HUMAN
Total
ZC3HF_HUMAN
RBPS
ZC3HF_HUMAN
Total
Protein abundance (log2)
Unstarved
Starved
Replicate
1
2
3
4
5