03: Changes in RNA Binding

Eneko Villanueva,Rayner Queiroz, Manasa Ramakrishna, Tom Smith

November 12, 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 and then setting up for a Limma analysis

```
total_as_protein_quant <- readRDS("../results/total_as_res_pro_agg_norm")
oops_as_protein_quant <- readRDS("../results/rbp_as_res_pro_agg_norm")
```

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. Combining total and RBP data

It is relatively easy to perform a pairwise comparison between treated and untreated samples either in the RBP or Total proteome. 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 Arsenite experiments so we can compare just those proteins that are captures across both total and RBP datasets.

```
intersecting_as_proteins <- intersect(rownames(total_as_protein_quant),
    rownames(oops_as_protein_quant))
print(paste("Number of RBPs also captured in the Total Proteome for Control vs Arsenite treated samples
    length(intersecting_as_proteins)), sep = "")
```

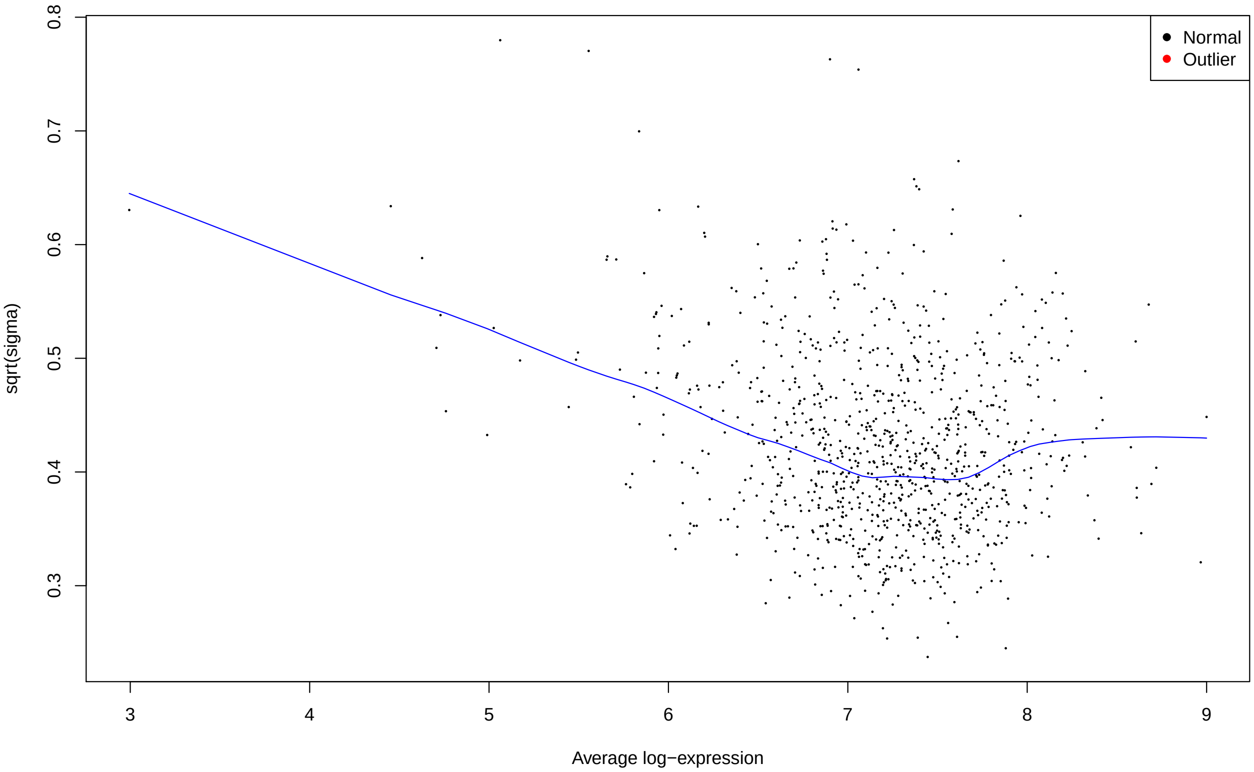
```
## [1] "Number of RBPs also captured in the Total Proteome for Control vs Arsenite treated samples is 98
```

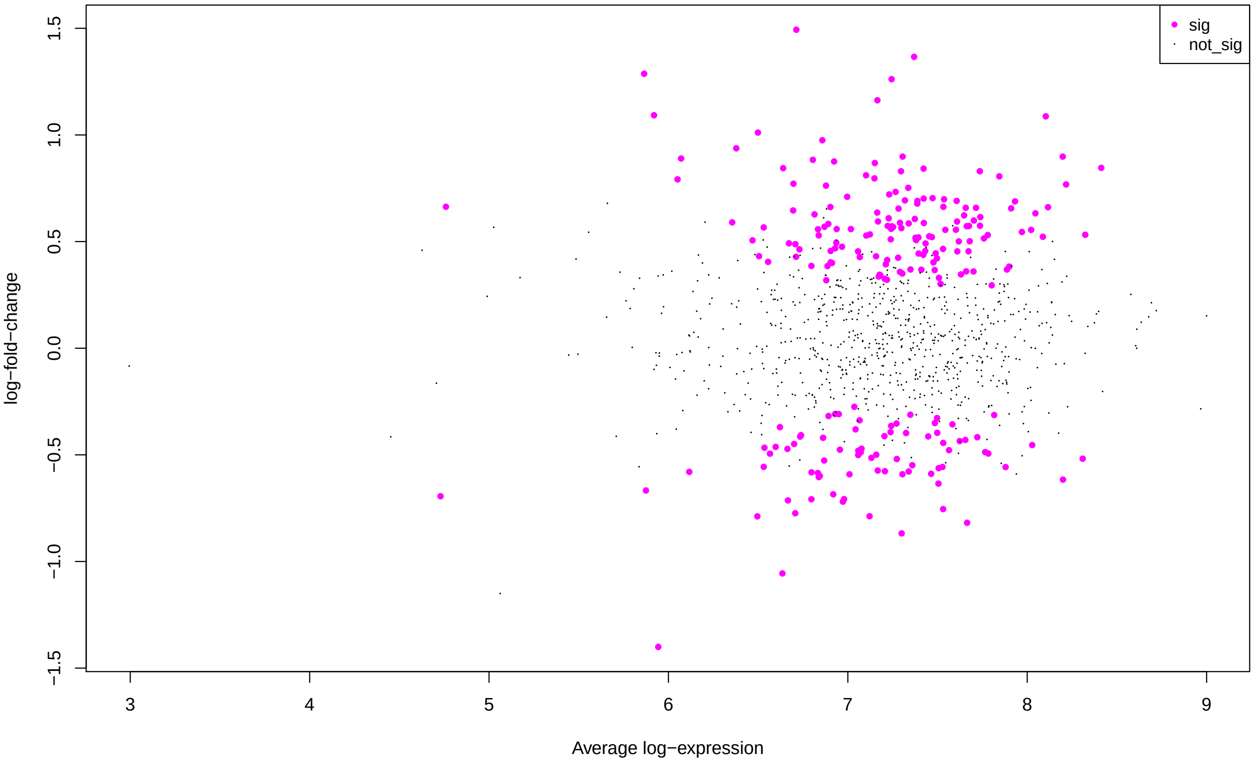
```
# Subset of intersecting AS proteins only
total_as_for_combination <- total_as_protein_quant[intersecting_as_proteins,
    ]
rbp_as_for_combination <- oops_as_protein_quant[intersecting_as_proteins,
    ]
```

```
combined_as_intensities = combine_esets(total_as_for_combination,
    rbp_as_for_combination)
pData(combined_as_intensities)$Condition = factor(pData(combined_as_intensities)$Condition,
    levels = c("Ctrl", "100uM-Arsenite", "400uM-Arsenite"))
```

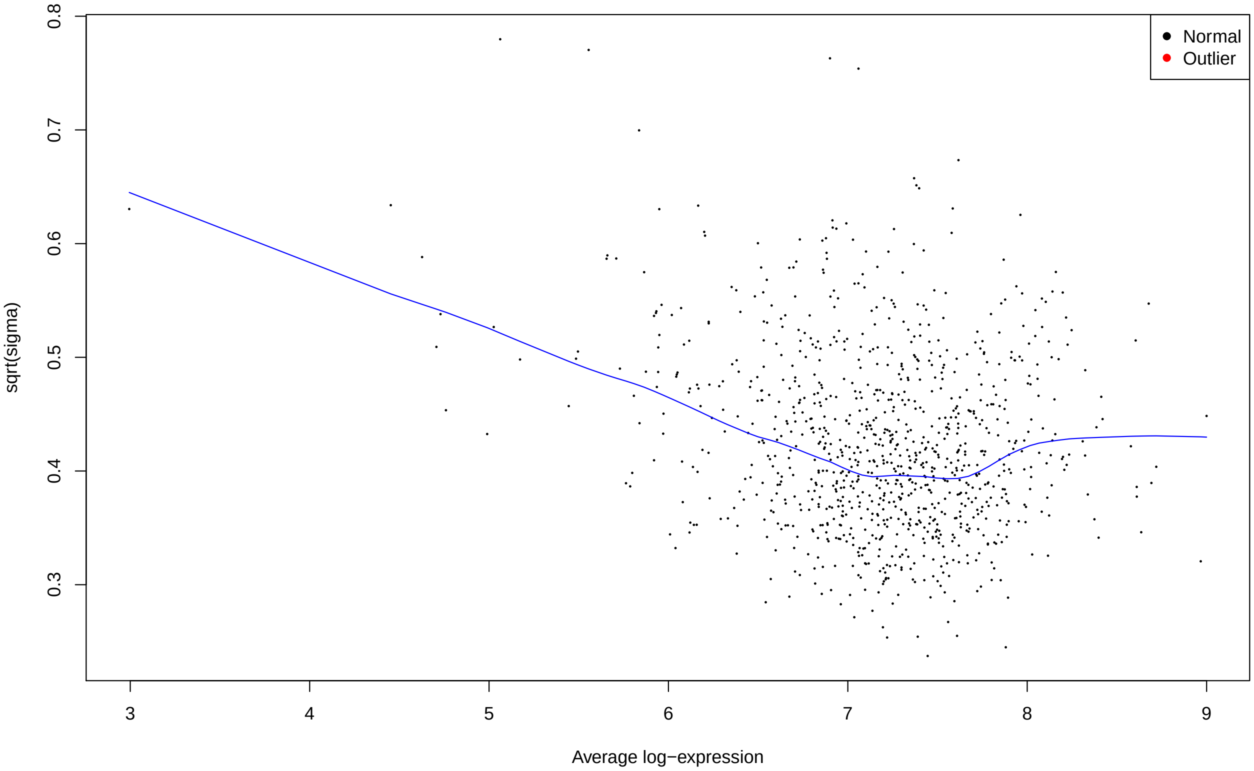
The we run `limma` on the combined intensities and this time test for a signficant interaction coefficient. There are 224 proteins differentially expressed in cells treated with 100uM NaAs2 relative to Control and similarly, there are 184 proteins differentially expressed in cells treated with 400uM NaAs2 relative to Control.

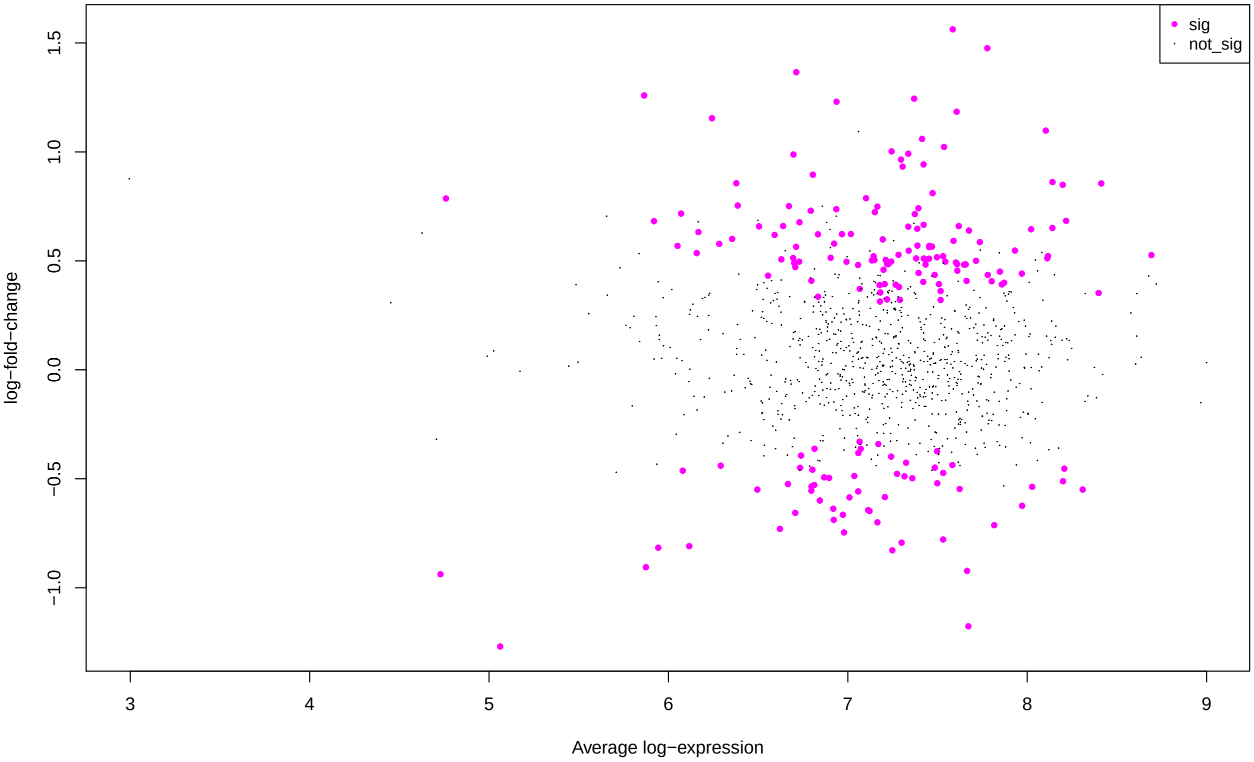
```
# Ctrl-vs-100umArsenite
as_rbps_de_100uM = run_limma(combined_as_intensities,
    "condition100uM-Arsenite:typeOOPS")
```





```
as_rbps_de_mod_100uM = modify_output(as_rbps_de_100uM)
as_rbps_p_value_100uM <- as_rbps_de_mod_100uM %>% filter(P.Value <=
    0.05)
write_csv(as_rbps_p_value_100uM, path = "../results/Ctrl-vs-100uM-Arsenite-Treated-rawp-le-0.05.csv")

# Ctrl-vs-400umArsenite
as_rbps_de_400uM = run_limma(combined_as_intensities,
    "condition400uM-Arsenite:typeOOPS")
```





```
as_rbps_de_mod_400uM = modify_output(as_rbps_de_400uM)
as_rbps_p_value_400uM <- as_rbps_de_mod_400uM %>% filter(P.Value <=
    0.05)
write_csv(as_rbps_p_value_400uM, path = "../results/Ctrl-vs-400uM-Arsenite-Treated-rawp-le-0.05.csv")
```

We are also interested in working out if there is any difference between the 100uM and 400uM treated cells.

I have set the design up slightly differently here incorporating the replicate number as a blocking factor. Blocking can be applied in any situation where there are batch effects or where the experiment has been conducted in blocks. The treatments can be adjusted for differences between the blocks by using a model formula of the form

design <- model.matrix(~Block+Treatment)

```
# Create a design matrix
pData(combined_as_intensities)$Rep = sapply(strsplit(pData(combined_as_intensities)$Sample_name,
    "\\_"), "[[", 2)
treat <- paste(combined_as_intensities$Condition, combined_as_intensities$Type,
    sep = ".")
treat = gsub("\\-", ".", treat)
design2 = model.matrix(~0 + treat)

# Calculate correlation between replicates
corfit <- duplicateCorrelation(combined_as_intensities,
    design2, block = combined_as_intensities$Rep)

# Fit a linear model to include a block and
# correlation
fit <- lmFit(combined_as_intensities, design2, block = pData(combined_as_intensities)$Rep,
    correlation = corfit$consensus)

# Create a vector to extract contrasts
cm <- makeContrasts(Ctrlvs100uM = (treatCtrl.Total -
    treatCtrl.OOPS) - (treat100uM.Arsenite.Total -
    treat100uM.Arsenite.OOPS), Ctrlvs400uM = (treatCtrl.Total -
    treatCtrl.OOPS) - (treat400uM.Arsenite.Total -
    treat400uM.Arsenite.OOPS), As400vs100uM = (treat400uM.Arsenite.Total -
    treat400uM.Arsenite.OOPS) - (treat100uM.Arsenite.Total -
    treat100uM.Arsenite.OOPS), levels = design2)

# Extract contrasts from the fitted linear model
fit2 <- contrasts.fit(fit, cm)

# Calculating the F-statistic
fit2 <- eBayes(fit2)

# Extracting the top hits for each contrast
Ctrl.100uM = topTable(fit2, adjust = "BH", coef = "Ctrlvs100uM",
    number = Inf, confint = T)
de_mod_100uM = modify_output(Ctrl.100uM)
p_value_100uM <- de_mod_100uM %>% filter(P.Value <=
    0.05)
write.table(p_value_100uM, "../results/Ctrl-vs-100uM-Arsenite-Treated-BLOCK-rawp-le-0.05.tsv",
    sep = "\t", row.names = F, quote = F)

Ctrl.400uM = topTable(fit2, adjust = "BH", coef = "Ctrlvs400uM",
    number = Inf, confint = T)
de_mod_400uM = modify_output(Ctrl.400uM)
p_value_400uM <- de_mod_400uM %>% filter(P.Value <=
    0.05)
write.table(p_value_400uM, "../results/Ctrl-vs-400uM-Arsenite-Treated-BLOCK-rawp-le-0.05.tsv",
    sep = "\t", row.names = F, quote = F)
```

```
As400.100uM = topTable(fit2, adjust = "BH", coef = "As400vs100uM",
    number = Inf, confint = T)
de_mod_100.400uM = modify_output(As400.100uM)
p_value_100.400uM <- de_mod_100.400uM %>% filter(P.Value <=
    0.05)
write.table(p_value_100.400uM, "../results/100-vs-400uM-Arsenite-Treated-BLOCK-rawp-le-0.05.tsv",
    sep = "\t", row.names = F, quote = F)

# Significantly DE proteins - only a few
de_mod_100uM %>% filter(adj.P.Val <= 0.05)
```

```
##    uniprot_id gene_name
## 1      Q12830      BPTF
## 2      Q12904     AIMP1
## 3      P62241      RPS8
## 4      Q15366     PCBP2
## 5      P55209    NAP1L1
## 6      P62277     RPS13
## 7      Q15029    EFTUD2
## 8      P11940    PABPC1
## 9      P15151       PVR
## 10     Q69YN4     VIRMA
## 11     P26038       MSN
## 12     Q9GZR7     DDX24
## 13     P42696     RBM34
## 14     O75821     EIF3G
## 15     P08648     ITGA5
##                                                              protein_desc
## 1                             Nucleosome-remodeling factor subunit BPTF
## 2  Aminoacyl tRNA synthase complex-interacting multifunctional protein 1
## 3                                                  40S ribosomal protein S8
## 4                                                Poly(rC)-binding protein 2
## 5                                  Nucleosome assembly protein 1-like 1
## 6                                                 40S ribosomal protein S13
## 7                      116 kDa U5 small nuclear ribonucleoprotein component
## 8                                           Polyadenylate-binding protein 1
## 9                                                      Poliovirus receptor
## 10                                               Protein virilizer homolog
## 11                                                                  Moesin
## 12                                         ATP-dependent RNA helicase DDX24
## 13                                                    RNA-binding protein 34
## 14                     Eukaryotic translation initiation factor 3 subunit G
## 15                                                       Integrin alpha-5
##    protein_length      logFC       CI.L       CI.R  AveExpr         t
## 1            3046 -1.4005286 -1.9320173 -0.8690400 5.943282 -5.495572
## 2             312  0.8297310  0.4899724  1.1694895 7.296327  5.093096
## 3             208  0.6613486  0.3683275  0.9543696 8.115893  4.707026
## 4             365 -0.7135067 -1.0299387 -0.3970747 6.666104 -4.702541
## 5             391  1.4932698  0.8059538  2.1805858 6.712858  4.531028
## 6             151  0.7328902  0.3945351  1.0712453 7.266314  4.517322
## 7             972 -0.7194406 -1.0622434 -0.3766379 6.972462 -4.376889
## 8             636 -0.7878690 -1.1679747 -0.4077633 7.121074 -4.322794
## 9             417  1.2614481  0.6523033  1.8705929 7.244096  4.318807
## 10           1812  0.8834219  0.4555717  1.3112721 6.805117  4.306172
## 11            577  0.8421574  0.4342916  1.2500232 7.422235  4.306167
## 12            859  0.6460012  0.3310497  0.9609526 6.695161  4.277644
## 13            430  1.2868395  0.6273371  1.9463420 5.864630  4.069330
## 14            320  0.8298233  0.4021360  1.2575107 7.735739  4.046450
## 15           1049 -0.6851070 -1.0402630 -0.3299510 6.918815 -4.023039
##         P.Value  adj.P.Val          B
## 1  2.199005e-05 0.02161622  2.7852349
## 2  5.497359e-05 0.02701952  1.9678902
## 3  1.341099e-04 0.02991727  1.1678212
## 4  1.355144e-04 0.02991727  1.1584532
## 5  2.020197e-04 0.02991727  0.7991578
## 6  2.085824e-04 0.02991727  0.7703691
## 7  2.895491e-04 0.02991727  0.4748579
## 8  3.285992e-04 0.02991727  0.3608113
## 9  3.316789e-04 0.02991727  0.3524008
## 10 3.416305e-04 0.02991727  0.3257473
## 11 3.416341e-04 0.02991727  0.3257377
## 12 3.652159e-04 0.02991727  0.2655526
## 13 5.948448e-04 0.04344608 -0.1744188
## 14 6.275905e-04 0.04344608 -0.2227554
## 15 6.629615e-04 0.04344608 -0.2722113
```

```
de_mod_400uM %>% filter(adj.P.Val <= 0.05)
```

```
##    uniprot_id gene_name
## 1      Q9Y314    NOSIP
## 2      P23396     RPS3
## 3      Q12904    AIMP1
## 4      Q16637     SMN1
## 5      P38432     COIL
## 6      O15479   MAGEB2
## 7      P26038      MSN
## 8      P10599      TXN
## 9      Q53EP0   FNDC3B
## 10     Q7KZF4     SND1
## 11     P13489     RNH1
## 12     Q69YN4    VIRMA
## 13     Q15269     PWP2
## 14     O95292     VAPB
## 15     O15355    PPM1G
##                                                               protein_desc
## 1                               Nitric oxide synthase-interacting protein
## 2                                                40S ribosomal protein S3
## 3  Aminoacyl tRNA synthase complex-interacting multifunctional protein 1
## 4                                           Survival motor neuron protein
## 5                                                                  Coilin
## 6                                          Melanoma-associated antigen B2
## 7                                                                  Moesin
## 8                                                             Thioredoxin
## 9                       Fibronectin type III domain-containing protein 3B
## 10                    Staphylococcal nuclease domain-containing protein 1
## 11                                                 Ribonuclease inhibitor
## 12                                              Protein virilizer homolog
## 13                                  Periodic tryptophan protein 2 homolog
## 14            Vesicle-associated membrane protein-associated protein B/C
## 15                                                Protein phosphatase 1G
##    protein_length      logFC       CI.L       CI.R  AveExpr         t
## 1             301  1.4757064  0.9938954  1.9575173 7.777524  6.387608
## 2             243 -1.1763949 -1.5605032 -0.7922866 7.672099 -6.387257
## 3             312  0.9649250  0.6017078  1.3281422 7.296327  5.540413
## 4             294  1.1542774  0.6872288  1.6213259 6.242580  5.154222
## 5             576  1.0595363  0.6196925  1.4993801 7.413852  5.023799
## 6             319 -0.7125269 -1.0166300 -0.4084238 7.815854 -4.886471
## 7             577  0.9425038  0.5064769  1.3785307 7.422235  4.508009
## 8             105  1.1844434  0.6152029  1.7536839 7.606756  4.339438
## 9            1204  1.2302102  0.6255229  1.8348974 6.937107  4.242906
## 10            910 -0.6995923 -1.0501239 -0.3490607 7.164802 -4.162294
## 11            461  0.7371492  0.3653631  1.1089353 6.935709  4.135016
## 12           1812  0.8951127  0.4377216  1.3525038 6.805117  4.081361
## 13            919 -0.7294358 -1.1097156 -0.3491561 6.621502 -4.000357
## 14            243  0.8107707  0.3872097  1.2343318 7.472438  3.992059
## 15            546  0.9918062  0.4704382  1.5131741 7.336967  3.967322
##         P.Value   adj.P.Val            B
## 1  3.074369e-06 0.001512195  4.652482687
## 2  3.076693e-06 0.001512195  4.651795299
## 3  1.987582e-05 0.006512644  2.942967562
## 4  4.778556e-05 0.011743302  2.133086586
## 5  6.446332e-05 0.012673489  1.856039771
## 6  8.848250e-05 0.014496383  1.562672671
## 7  2.131642e-04 0.029934339  0.747228387
## 8  3.160519e-04 0.038834872  0.381780188
## 9  3.961535e-04 0.043268766  0.172200903
## 10 4.784554e-04 0.045577245 -0.002882464
## 11 5.100200e-04 0.045577245 -0.062127552
## 12 5.783174e-04 0.047373837 -0.178646455
## 13 6.991257e-04 0.049500559 -0.354467074
## 14 7.128429e-04 0.049500559 -0.372469884
## 15 7.553493e-04 0.049500559 -0.426126833
```

```r
de_mod_100.400uM %>% filter(adj.P.Val <= 0.05)
```

```
##   uniprot_id gene_name                               protein_desc
## 1     Q9Y314    NOSIP  Nitric oxide synthase-interacting protein
## 2     P38432     COIL                                     Coilin
## 3     P23396     RPS3                   40S ribosomal protein S3
##   protein_length      logFC       CI.L       CI.R  AveExpr         t
## 1            301 -1.5146639 -1.9653567 -1.0639710 7.777524 -7.008911
## 2            576 -1.0195699 -1.4310061 -0.6081337 7.413852 -5.168082
## 3            243  0.8377466  0.4784462  1.1970470 7.672099  4.862615
##        P.Value    adj.P.Val        B
## 1 8.285394e-07 0.0008144543 5.607320
## 2 4.629316e-05 0.0227530881 2.107937
## 3 9.350028e-05 0.0306369260 1.482636
```

While there are several proteins whose p-value is significant, only a few survive multiple testing correction (shown above). Of these, there doesn't seem to be much in the way of a functional theme. Hence, there isn't much to go on for GO enrichment analysis. We could use the lists based just on raw p-value and see if we have any luck with them.

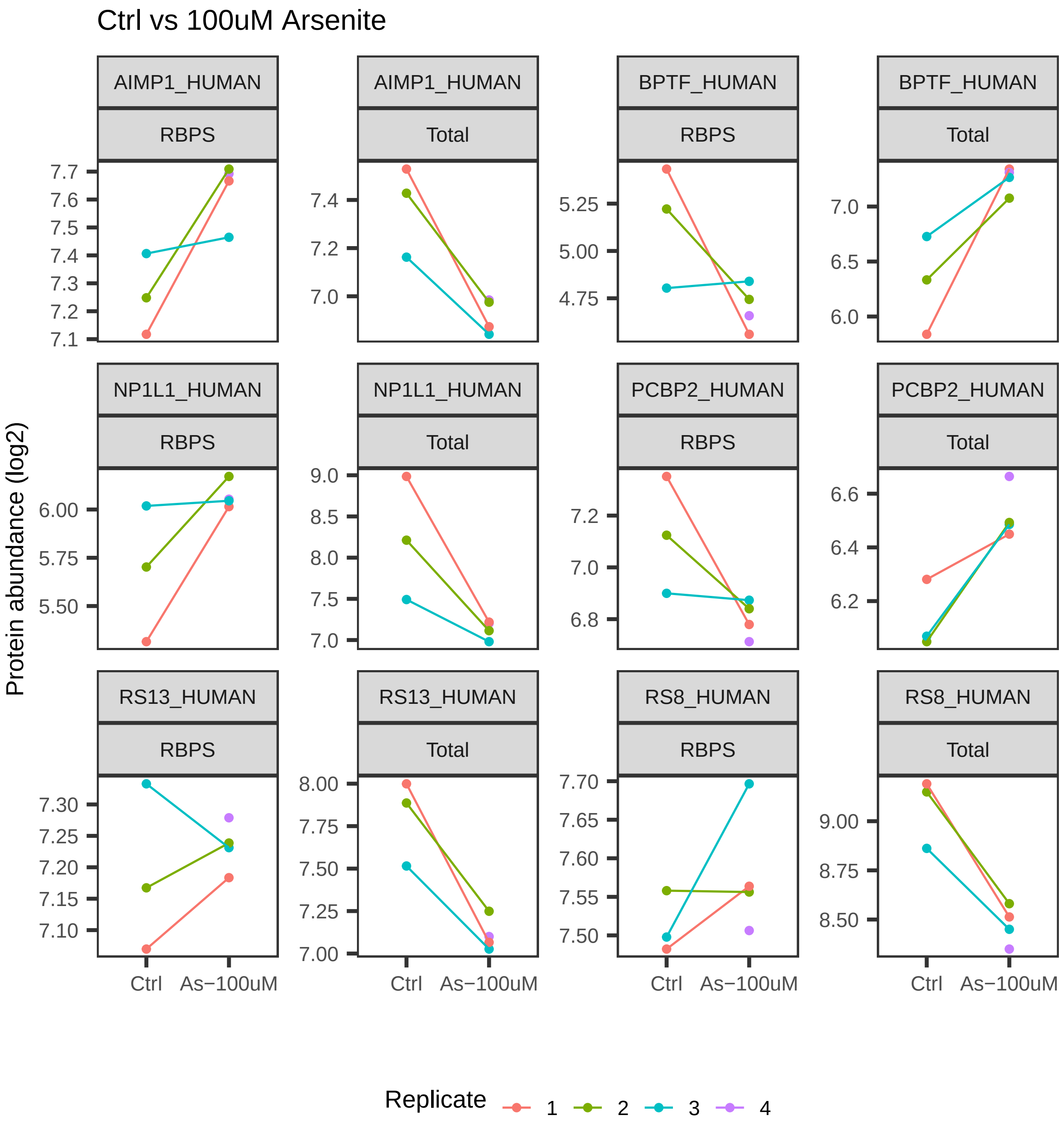
4. Plotting expression of Top 10 proteins across studies

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.

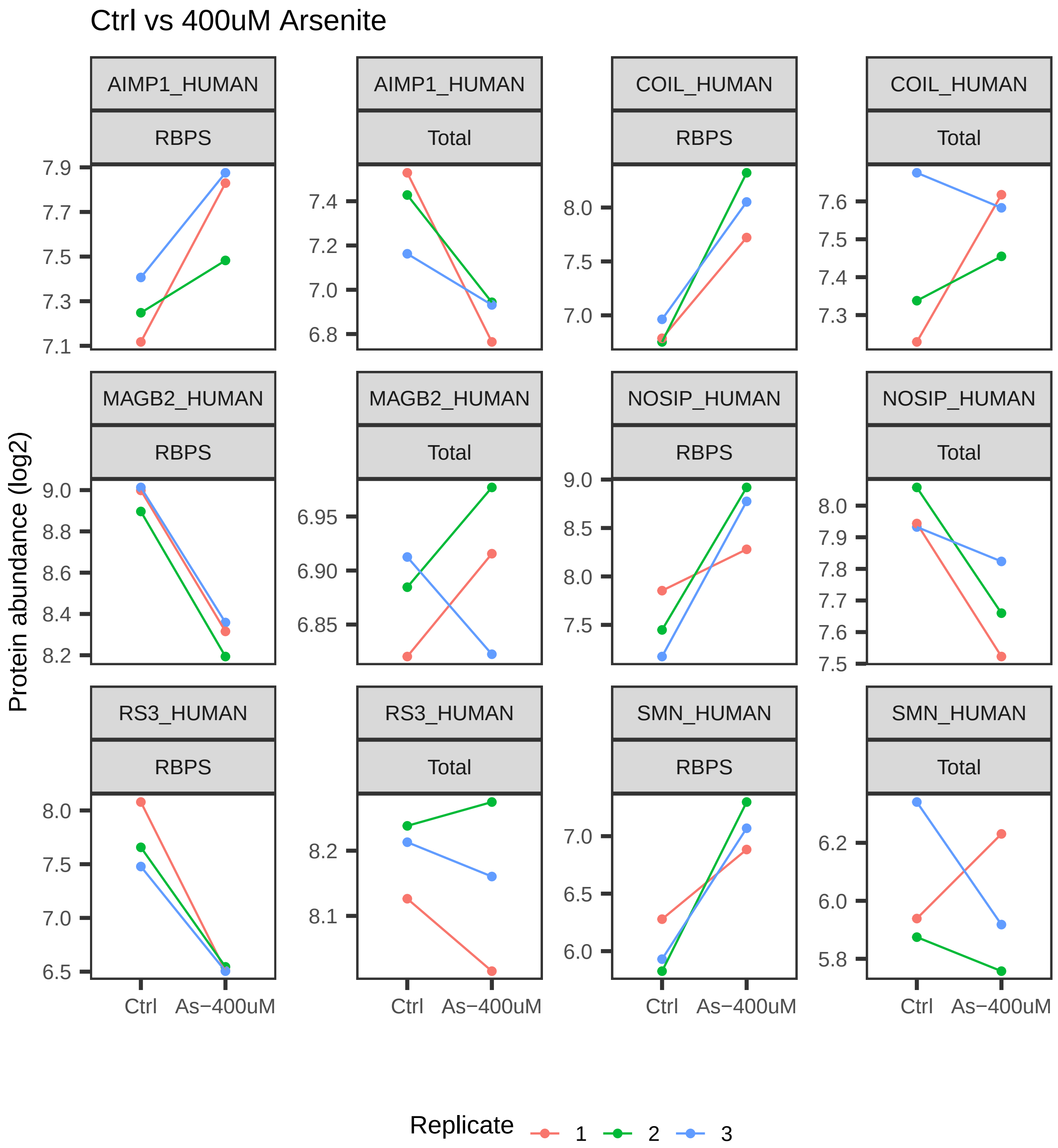
```r
# Combining Total and RBP data : AS
total_as_exprs <- makeLongExprs(total_as_protein_quant,
    intersecting_as_proteins)
oops_as_exprs <- makeLongExprs(oops_as_protein_quant,
    intersecting_as_proteins)
combined_as_exprs <- rbind(cbind(total_as_exprs, Type = "Total"),
    cbind(oops_as_exprs, Type = "RBPS"))

# Adding protein information
protein_info <- read.delim("../shared_files/human_protein_ids_plus_gene_names.tsv")
combined_as_exprs <- combined_as_exprs %>% merge(protein_info,
    by.x = "uniprotID", by.y = "Entry")

# Renaming the levels for better plotting
library(plyr)
combined_as_exprs$Condition = revalue(combined_as_exprs$Condition,
    c(X400uM.Arsenite = "As-400uM", X100uM.Arsenite = "As-100uM"))
str(combined_as_exprs)
```

```
## 'data.frame':    19660 obs. of  8 variables:
##  $ uniprotID    : chr  "A4D1E9" "A4D1E9" "A4D1E9" "A4D1E9" ...
##  $ Condition    : Factor w/ 3 levels "Ctrl","As-100uM",..: 1 1 2 1 2 2 2 2 3 2 ...
##  $ Replicate    : chr  "1" "3" "4" "2" ...
##  $ Intensity    : num  6.07 5.7 5.82 5.58 5.69 ...
##  $ Type         : Factor w/ 2 levels "Total","RBPS": 1 2 1 2 1 2 1 2 1 1 ...
##  $ Entry.name   : Factor w/ 20258 levels "1433B_HUMAN",..: 7183 7183 7183 7183 7183 7183 7183 7183 7
##  $ Protein.names: Factor w/ 20257 levels "(E3-independent) E2 ubiquitin-conjugating enzyme (EC 2.3.2
##  $ Gene.names   : Factor w/ 19968 levels "","A1BG","A1CF ACF ASP",..: 7044 7044 7044 7044 7044 7044 7
```

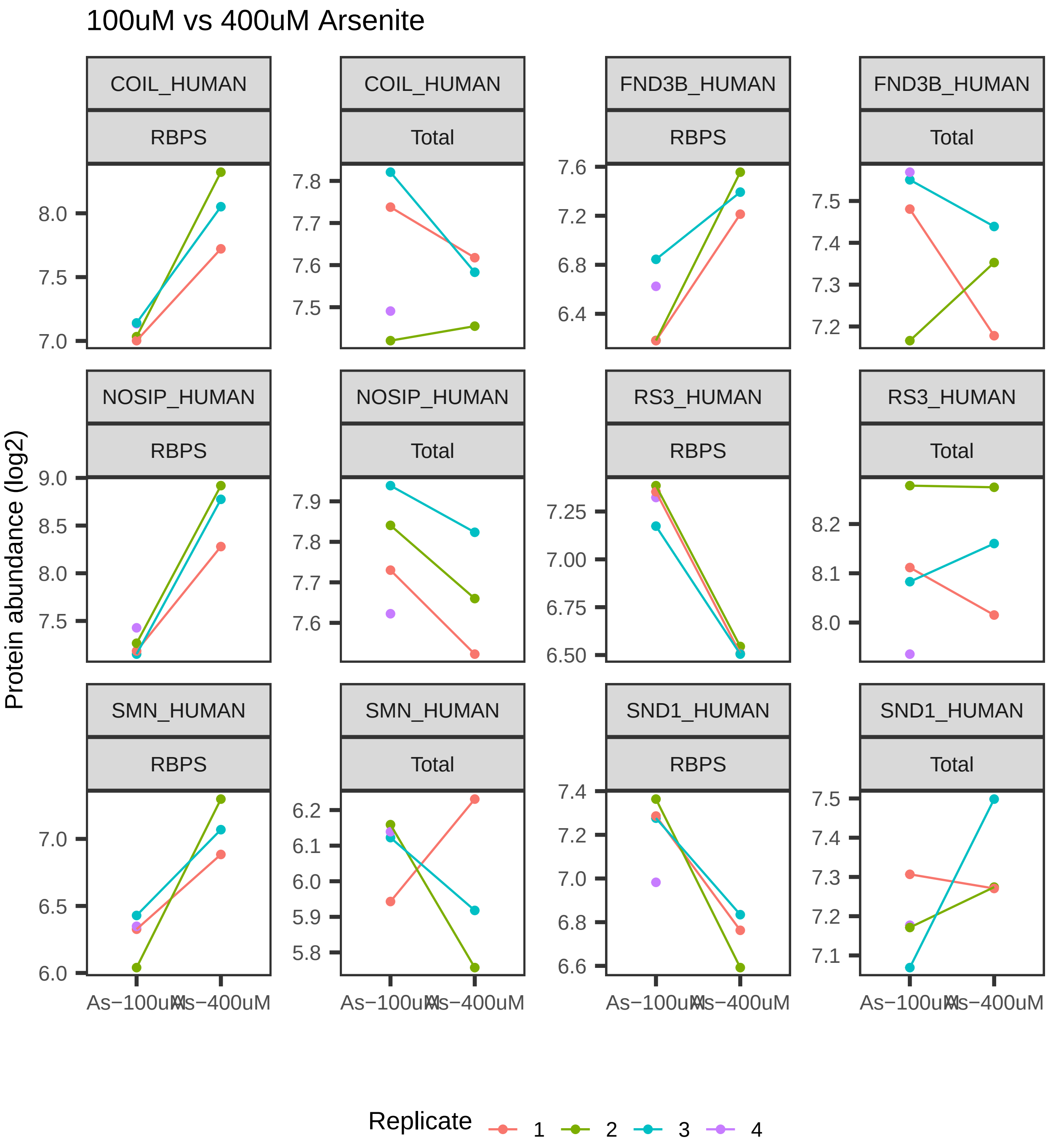
```r
# Top 10 proteins from both comparisons
lowest_p_ctrl_100uM_proteins <- p_value_100uM %>% arrange(P.Value) %>%
    pull(uniprot_id) %>% head(6)
lowest_p_ctrl_400uM_proteins <- p_value_400uM %>% arrange(P.Value) %>%
    pull(uniprot_id) %>% head(6)
lowest_p_100_400uM_proteins <- p_value_100.400uM %>%
    arrange(P.Value) %>% pull(uniprot_id) %>% head(6)

# Plots
plotTop10(combined_as_exprs[-which(combined_as_exprs$Condition ==
    "As-400uM"), ], lowest_p_ctrl_100uM_proteins, "Ctrl vs 100uM Arsenite")
```



```
plotTop10(combined_as_exprs[-which(combined_as_exprs$Condition ==
    "As-100uM"), ], lowest_p_ctrl_400uM_proteins, "Ctrl vs 400uM Arsenite")
```



```r
plotTop10(combined_as_exprs[-which(combined_as_exprs$Condition ==
    "Ctrl"), ], lowest_p_100_400uM_proteins, "100uM vs 400uM Arsenite")
```


100uM vs 400uM Arsenite
Protein abundance (log2)
COIL_HUMAN
RBPS
8.0
7.5
7.0
COIL_HUMAN
Total
7.8
7.7
7.6
7.5
FND3B_HUMAN
RBPS
7.6
7.2
6.8
6.4
FND3B_HUMAN
Total
7.5
7.4
7.3
7.2
NOSIP_HUMAN
RBPS
9.0
8.5
8.0
7.5
NOSIP_HUMAN
Total
7.9
7.8
7.7
7.6
RS3_HUMAN
RBPS
7.25
7.00
6.75
6.50
RS3_HUMAN
Total
8.2
8.1
8.0
SMN_HUMAN
RBPS
7.0
6.5
6.0
SMN_HUMAN
Total
6.2
6.1
6.0
5.9
5.8
SND1_HUMAN
RBPS
7.4
7.2
7.0
6.8
6.6
SND1_HUMAN
Total
7.5
7.4
7.3
7.2
7.1
As−100uM
As−400uM
Replicate
1
2
3
4