02: Sample Clustering and Batch Effects

Eneko Villanueva,Rayner Queiroz, Manasa Ramakrishna, Tom Smith

November 12, 2019

Contents

1. Introduction

Here we want to check for batch effects. The expectation is that the replicate number may influence protein abundance. All replicates for a given condition were run within a single TMT-plex experiment hopefully removing variation from multi-plexing and analysis by the spectrometer. BUT. . . ..

```
suppressMessages(library(gridExtra))
source("../../CamProt_R/Utility.R")
```

2. Reading normalised data

We start by reading in the normalised RDS files generated using the first analysis notebook '1_aggregate_to_protein_an_qc.Rmd'. Remember that the data was aggregated into peptides, then into proteins, missing values were imputed using "min" and finally, samples were log centre normalised. Now we want to see if there is anything odd about the data despite us having corrected for any technical artefacts from the experimental process.

```
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. Plotting variability

We would like to see the separation of samples into their treatment groups across all the datasets. RBP = OOPS. However, no experiment is perfect so we expect some degree of variability within each treatment group across the 3-4 replicates. What we want to keep an eye out for is outlier samples within each group.

3a. Initial PCAs

We start the process by plotting PCAs - we draw them in pairs PC1&2, PC3&4, PC5&6. We hope that there is no pattern or separation by replicate as this would indicate batch effects. PCAs will also flag any samples that are overall quite different to their replicate samples. All the plots can be found in the folder "../plots" and have the suffix "*PCA-on-normalised-data.pdf".

```
names = c("TotalProt-Control-vs-NaAs2-Treated", "RBPs-Control-vs-NaAs2-Treated")
c = 1

for (i in list(total_as_protein_quant, oops_as_protein_quant)) {
    p = plotSamplePCA(i)
    pdf(paste("../plots/", names[c], "_PCA-on-normalised-data.pdf",
        sep = ""), paper = "a4r", width = 14, height = 8)
    plot(p$l/sum(p$l), col = "red", ylab = "Variation explained")
    plot(p$pca)
    dev.off()
    c = c + 1
}
```

1. Total Protein : Control vs Arsenite treated
In this case, Replicate 1 of the Control seems to be very different to the rest of the samples and that's what PC1 is captuting (variation 62%). PC2 seems to approximately separate the treatment vs control samples (variation 10%). We have no real reason to remove Ctrl 1 so we will keep it in for now.

2. RBPs : Control vs Arsenite treated
In this case, Replicate 1 of the Control seems to be very different to the rest of the samples and that's what PC1 is captuting (variation 62%). PC2 seems to approximately separate the treatment vs control samples (variation 10%). We have no real reason to remove Ctrl 1 so we will keep it in for now.

Overall, the condition does not appear to have a large impact on relative protein abundance. Note that this does not mean there isn't a change in absolute protein abundance. Also, if the quantification was very noisy, we would also expect this to mask the biological signal. We'll check % CVs below to assess quantification noise.

Notes:
Check with Rayner about amount of protein that went into each study.

3b. Dodgey sample ? Go MAD !

We are checking median absolute deviation for samples that are problematic.To calculate MAD,
1. take the median across all proteins/peptides for a sample
2. subtract each value from the median to get deviations
3. take the absolute value of the deviations (remove +/- signs so all of them are positive)
4. take the median of the deviations

A low MAD says that the values of expression in the sample are very similar or don't deviate very much from the median i.e the dynamic range of protein expression values is low. Since a median is less affected by outliers, this gives us a more robust estimation of variation within a sample.

```
total_as_res_pro_agg = readRDS("../results/total_as_res_pro_agg.rds")
rbp_as_res_pro_agg = readRDS("../results/rbp_as_res_pro_agg.rds")

# Arsenite
apply(exprs(total_as_res_pro_agg), 2, function(x) mad(x,
    constant = 1))
```

```
##    126   127N   127C   128N   128C   129N   129C   130N   130C    131
## 152.40  92.60  52.80  35.40  45.45  34.25  28.40  31.95  42.90  57.55
```

```
apply(exprs(rbp_as_res_pro_agg), 2, function(x) mad(x,
    constant = 1))
```

```
##    126   127N   127C   128N   128C   129N   129C   130N   130C    131
## 134.25  67.70  49.75  33.25  33.30  30.35  36.00  38.55  61.95  37.75
```

3c. PCAs after dodgey samples being removed

Looking at the data values, the samples we think are problematic have very high Median Absolute Deviation values relative to their replicate friends. Interestingly, tag 126 has very high MAD for both Total and RBP experiments. For now, we will remove those samples with the largest mad in each of the experiments. Again, these PCAs can be found in the folder "../plots/" and have the suffix "PCA-on-normalised-data-dodgey-samples-removed.pdf".

```r
# Removing dodgey samples
total_as_protein_quant_nd = total_as_protein_quant[,
    grep("126", sampleNames(total_as_protein_quant),
        invert = T)]
oops_as_protein_quant_nd = oops_as_protein_quant[,
    grep("126", sampleNames(oops_as_protein_quant),
        invert = T)]

names = c("TotalProt-Control-vs-NaAs2-Treated", "RBPs-Control-vs-NaAs2-Treated")
c = 1

for (i in list(total_as_protein_quant_nd, oops_as_protein_quant_nd)) {
    p = plotSamplePCA(i)
    pdf(paste("../plots/", names[c], "_PCA-on-normalised-data-dodgey-samples-removed.pdf",
        sep = ""), paper = "a4r", width = 14, height = 8)
    plot(p$l/sum(p$l), col = "red", ylab = "Variation explained")
    plot(p$pca)
    dev.off()
    c = c + 1
}

saveRDS(total_as_protein_quant_nd, file = "../results/total_as_res_pro_agg_norm_nododgey.rds")
saveRDS(oops_as_protein_quant_nd, file = "../results/rbp_as_res_pro_agg_norm_nododgey.rds")
```

4. Combining all datasets

This won't be used downstream but is done to check whether there is clear separation between the experiments conducted in the study. To do so, we focus only on those proteins that are present in all treatments/conditions and are present in total and rbp proteomes.

```r
# Define intersecting proteins
intersecting_as_proteins <- intersect(rownames(total_as_protein_quant),
    rownames(oops_as_protein_quant))
print(length(intersecting_as_proteins))
```

```
## [1] 983
```

```r
# Combining data across all experiments
total_as_exprs <- exprs(total_as_protein_quant[intersecting_as_proteins,
    ])
colnames(total_as_exprs) <- paste0("Total-AS_", pData(total_as_protein_quant)$Sample_name)

oops_as_exprs <- exprs(oops_as_protein_quant[intersecting_as_proteins,
    ])
```

```r
colnames(oops_as_exprs) <- paste0("RBP-AS_", pData(oops_as_protein_quant)$Sample_name)

# Mega expression matrix
combined_exprs <- cbind(total_as_exprs, oops_as_exprs)
```

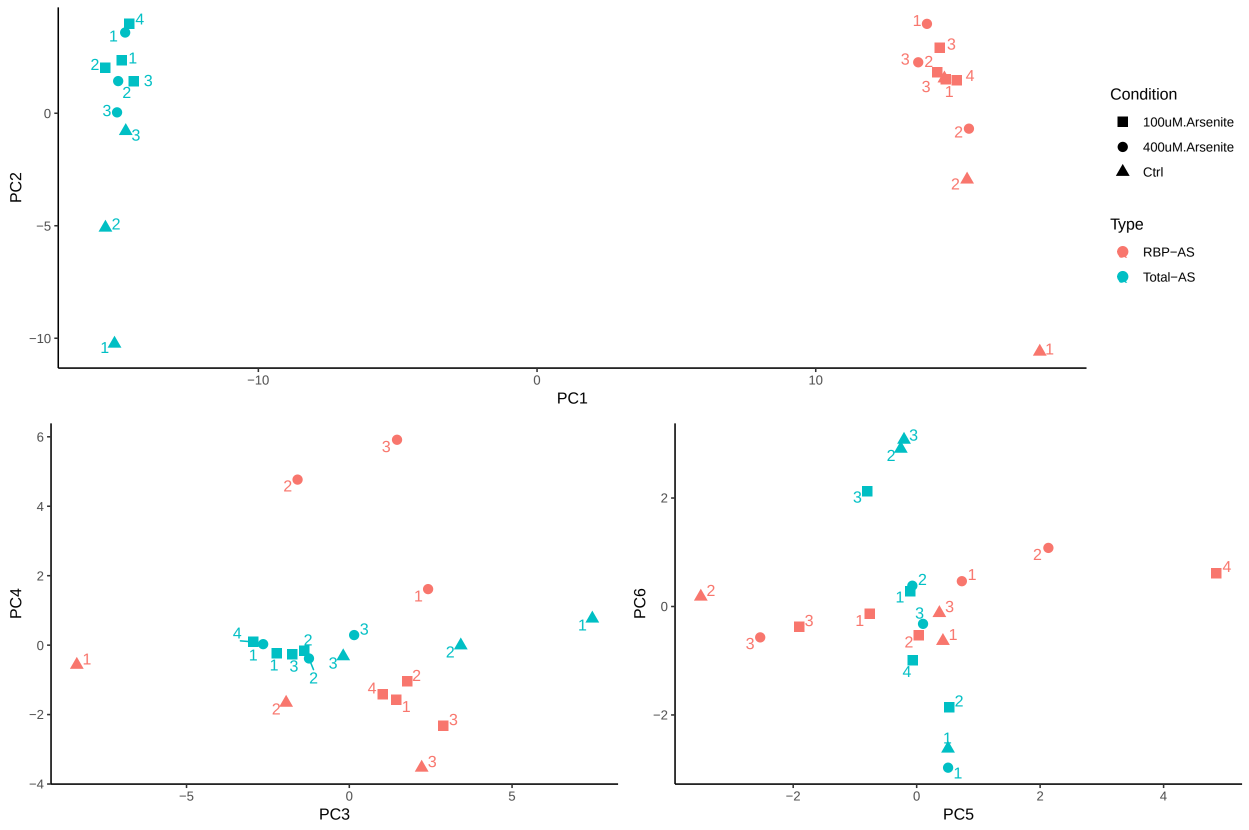
4a. Mega PCA across all samples

We plot PC1 vs PC2, PC3 vs PC4 and PC5 vs PC6 for these 20 samples below without the dodgey ones from above.Then we remake the plot having removed what we call the "dodgey" samples as defined above.

```r
# Mega PCA of all samples
pca <- prcomp(t(combined_exprs))
loadings <- pca$sdev^2

# Annotate samples to enable PCA labelling
projections <- data.frame(pca$x) %>% tibble::rownames_to_column("sample") %>%
    separate(sample, into = c("Type", "Condition",
        "Replicate"), sep = "_")

# Plotting the PCA with labels so we can tell which
# samples are most variable.
library(ggrepel)
p <- projections %>% ggplot(aes(PC1, PC2, shape = Condition,
    colour = Type)) + geom_point(size = 3) + geom_text_repel(aes(label = Replicate)) +
    scale_shape_manual(values = c(15, 16, 17, 18)) +
    theme_classic()

p2 <- p + aes(PC3, PC4) + theme(legend.position = "none")
p3 <- p + aes(PC5, PC6) + theme(legend.position = "none")

pdf("../plots/Mega-PCA-of-all-20-samples-3-conditions-3-replicates-samples.pdf",
    paper = "a4r", width = 14, height = 8)
plot(loadings/sum(loadings))
pt = arrangeGrob(p, p2, p3, nrow = 2, layout_matrix = rbind(c(1,
    1), c(2, 3)))
plot(pt)
dev.off()
```

```
## pdf
##   2
```

```r
plot(pt)
```



Looking at the PCA plots, albeit using a subset of the total data, overall, the experiment seems to have worked pretty well. PC1 separates the Total from the OOPS/RBP samples very very clearly. PC2 seems to separate the slightly dodgey controls (Tag126). Within the first three components, we capture 82% (PC1) + 8% (PC2) + 7% (PC3) = 95% of the variation in the data arising mainly from the fact that they are Total vs RBP samples and also protein extractions. The intra-condition variability is very very low i.e the replicates, for the most part, cluster tightly together with the exception of the RBP-US samples.

4b. Coefficient of variation

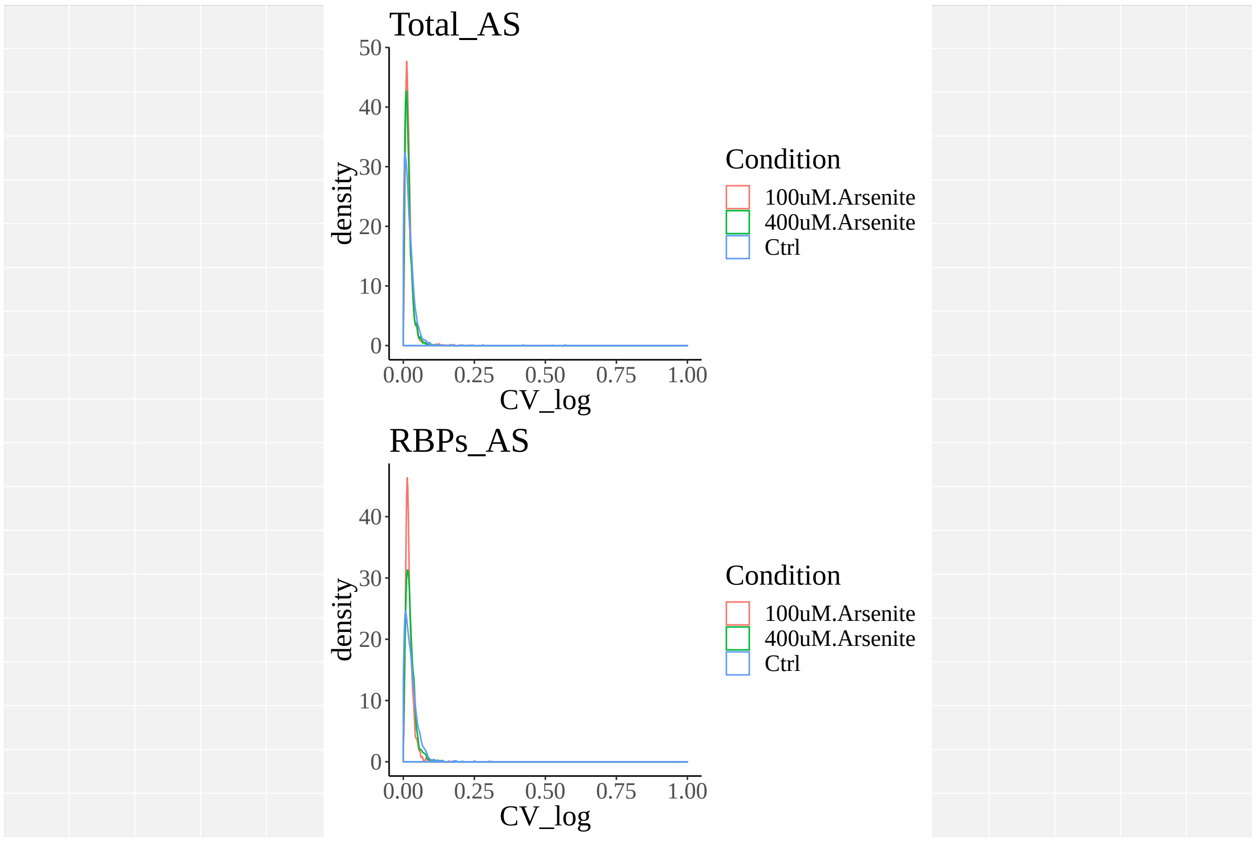
Below, we plot the distributions of coefficient of variance for each of the datasets. This is just an additional QC step that the data all looks OK and that there is no big difference in the variance within a condition since this could invalidate the assumptions of heteroscedasticity (same variance per group) that we will make in the modeling of protien abundance later.

```
p_as_total <- plotCVs(total_as_protein_quant_nd) +
    ggtitle("Total_AS") + xlim(0, 1)
# print(p_as_total) print(p_as_total + aes(CV_log))

p_as_oops <- plotCVs(oops_as_protein_quant_nd) + ggtitle("RBPs_AS") +
    xlim(0, 1)
# print(p_as_oops) print(p_as_oops + aes(CV_log))

g = arrangeGrob(p_as_total + aes(CV_log), p_as_oops +
    aes(CV_log), nrow = 2)

pdf("../plots/CV-plots-all-4-experiments-no-dodgey-samples.pdf",
    paper = "a4r", width = 14, height = 8)
```

```r
plot(g)
dev.off()
```

```
## pdf
##   2
```

```r
plot(g)
```



The CVs look as good as we would expect for TMT. 50% of the CVs should be < 0.2 after normalisation - see here. Also the OOPS data is not notably more variable which is good to see.