Visualising RNA binding dynamics

Here we will visualise total and RNA-bound protein abundances across conditions.

Below we load the required packages and set a plotting theme.

```
# load packages
library(tidyverse)
library(UniProt.ws)

# set up standardised plotting scheme
theme_set(theme_bw(base_size = 15) +
            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 start by reading in the data. Our input here is the protein-level quantification for the Nocodazole arrest/release experiment conducted for the OOPS NBT paper (https://www.nature.com/articles/s41587-018-0001-2). In this experiment, we wanted to assess changes in RNA binding in arrested/released cells. To do this, we quantified "total" protein abundance and RNA-bound (extracted by OOPS) protein abundance. The peptide-level abundances have been aggregated to protein level abundance and center-median normalised. Proteins with missing values have been removed. Only proteins quantified in both "total" and "OOPS" samples are included.

The input data here is identical to supplementary table 5 from the above paper.

```
protein_quant_raw <- readxl::read_excel('../raw/ncbi_30607034_OOPS_NBT_table_S5.xlsx',
                                        sheet=3, skip=1, n_max=1917)
```

In order to plot a functional subset of proteins, we will use the UniProt pathway annotations.

Warning: This cell will take a few minutes to run the query on the Uniprot database…

```
humanUP <- UniProt.ws(taxId=9606) # H.sapiens
protein_ids <- protein_quant_raw$master_protein

hsapiens.annot <- AnnotationDbi::select(
  humanUP,
  keys = protein_ids,
  columns = c("PATHWAY",  "PROTEIN-NAMES"),
  keystyle = "UNIPROTKB")
```

```
## Uniprot limits queries with a large amount of keys. It's recommended that the select method be invoke

## Getting extra data for A0AVT1, A1L0T0, A1L390... (400 total)

## Getting extra data for P22087, P22102, P22234... (400 total)

## Getting extra data for P78371, P78406, P78417... (400 total)

## Getting extra data for Q6ZSZ5, Q6ZUT6, Q712K3... (400 total)

## Getting extra data for Q9H307, Q9H3N1, Q9H3P7... (316 total)

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

```
hsapiens.pathway <-  hsapiens.annot %>% data.frame() %>%
  separate_rows(PATHWAY, sep="; ") %>% dplyr::select(UNIPROTKB, PROTEIN.NAMES, PATHWAY)
```

Identify the glycolysis proteins

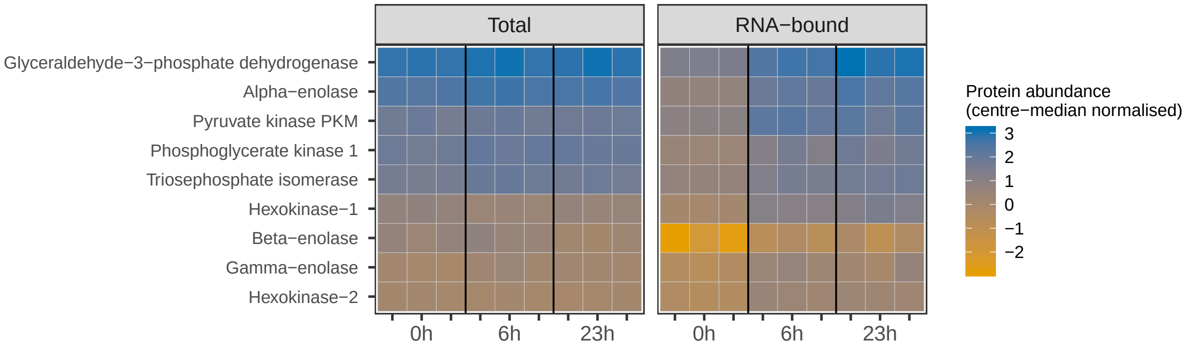
```
glycolysis_proteins <- hsapiens.pathway %>% filter(PATHWAY=='glycolysis')
glycolysis_proteins$cleaned_protein_name <- sapply(strsplit(glycolysis_proteins$PROTEIN.NAMES, split='\
```

Restructure the data and subset to the glycolysis proteins

```
glycolysis_intensities <- protein_quant_raw %>%
  gather(key='sample', value='intensity', -master_protein) %>%
  merge(glycolysis_proteins, by.x='master_protein', by.y='UNIPROTKB') %>%
  separate(sample, into=c('timepoint', 'replicate', 'type'), remove=FALSE) %>%
  mutate(type=factor(type, levels=c('total', 'OOPS'))) %>%
  mutate(timepoint=factor(timepoint, levels=c('0h', '6h', '23h')))

glycolysis_intensities$type <- recode(glycolysis_intensities$type,  'OOPS'='RNA-bound', 'total'='Total'
```

Plot the glycolysis proteins

```
protein_order <- glycolysis_intensities %>%
  group_by(cleaned_protein_name) %>% summarise(max_intensity=max(intensity)) %>%
  arrange(max_intensity) %>% pull(cleaned_protein_name)

p <- glycolysis_intensities %>%
  mutate(cleaned_protein_name=factor(cleaned_protein_name, levels=protein_order)) %>%
  ggplot(aes(interaction(replicate, timepoint), cleaned_protein_name, fill=intensity)) +
  geom_tile(colour='grey80', lwd=0.1) +
  facet_grid(.~type) +
  ylab('') + xlab('') +
  scale_x_discrete(labels=c('', '0h', '', '', '6h', '', '', '23h', '')) +
  geom_vline(xintercept=3.5) +
  geom_vline(xintercept=6.5) +
  scale_fill_gradient(low=cbPalette[1], high=cbPalette[5], name='Protein abundance\n(centre-median norma
  theme(axis.text.y=element_text(size=10), legend.title=element_text(size=10), legend.text=element_text

print(p)
```



```r
ggsave('../results/plots/rna_binding_changes_heatmap.png', width=10, height=5)
ggsave('../results/plots/rna_binding_changes_heatmap.pdf', width=10, height=5)
```