20210603_pooled.R

kelse

2022-06-01

```
# Plotting Prometheus nanoDSF data which measures intrinsic Trp fluorescence
# as a function of increasing temperature (20-95C)
# Data collected on 20210106 and 20210419
# 3 replicates each with n=3
# Each replicate corresponds to new protein purification sample

#Load libraries
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'stringr' was built under R version 4.0.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
library(broom)
library(readxl)
library(minpack.lm)
library(ggpmisc)
```

```
## Warning: package 'ggpmisc' was built under R version 4.0.5
```

```
##
## Attaching package: 'ggpmisc'
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```r
library(RColorBrewer)

theme_set(theme_bw() +
            theme(axis.text = element_text(size = 12, color = "black"),
                  panel.grid.major = element_blank(),
                  panel.grid.minor = element_blank())
)

# Import and tidy data  --------------------------------------------------------
### update excel sheet with proper heading info separated by "_"
raw106 <- read_excel("20210603_pooled_Fis1_Fis1dN.xlsx", sheet = 1) %>%
  gather(., "tmp1", "Fluorescence", 2:10) %>% # update ##:## w/ column numbers
  separate(., col = tmp1, into = c("prot", "n", "tr"),
           sep = "_") %>%
  mutate(., Fluorescence = as.double(Fluorescence))

raw419 <- read_excel("20210603_pooled_Fis1_Fis1dN.xlsx", sheet = 2) %>%
gather(., "tmp1", "Fluorescence", 2:10) %>% # update ##:## w/ column numbers
  separate(., col = tmp1, into = c("prot", "n", "tr"),
           sep = "_") %>%
  mutate(., Fluorescence = as.double(Fluorescence))

# merge the two data sets
merged <- union(raw106, raw419)

#calculate Tm values for each technical replicate
Tm_all <- merged %>%
  filter(., temp < 85) %>%
  group_by(., prot, n, tr) %>%
  summarise(max = max(Fluorescence))
```

```
## `summarise()` has grouped output by 'prot', 'n'. You can override using the `.groups` argument.
```

```r
Tm_all
```

```
## # A tibble: 18 x 4
## # Groups:   prot, n [6]
##    prot   n     tr        max
##    <chr>  <chr> <chr>   <dbl>
##  1 Fis1   1     1     0.00491
##  2 Fis1   1     2     0.00501
##  3 Fis1   1     3     0.00504
##  4 Fis1   2     1     0.00436
##  5 Fis1   2     2     0.00420
##  6 Fis1   2     3     0.00433
##  7 Fis1   3     1     0.00323
##  8 Fis1   3     2     0.00332
##  9 Fis1   3     3     0.00327
## 10 Fis1dN 1     1     0.00509
## 11 Fis1dN 1     2     0.00546
## 12 Fis1dN 1     3     0.00534
## 13 Fis1dN 2     1     0.00518
## 14 Fis1dN 2     2     0.00482
## 15 Fis1dN 2     3     0.00481
## 16 Fis1dN 3     1     0.00501
## 17 Fis1dN 3     2     0.00837
## 18 Fis1dN 3     3     0.00495
```

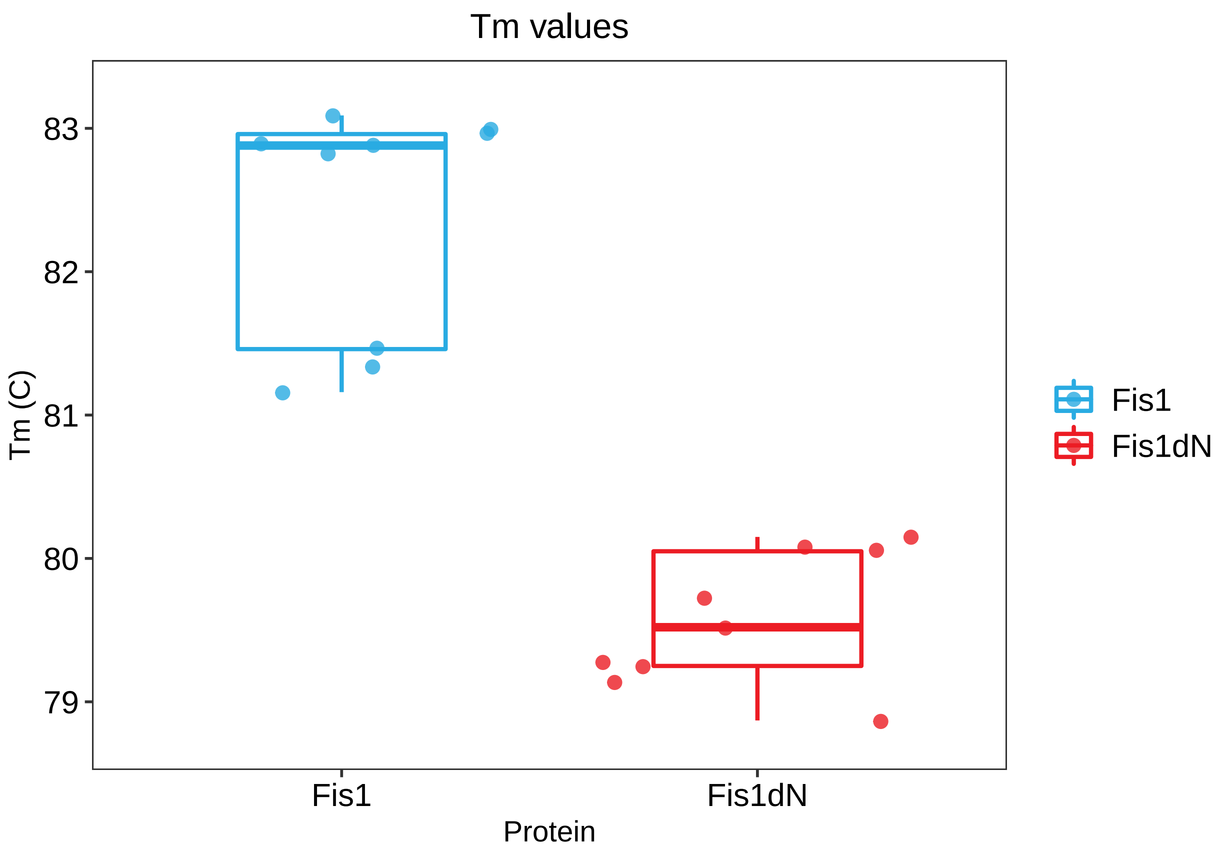
```r
# Searched merged for max fluorescence values reported in Tm_all
# add in corresponding temperature to max fluorescence value

Tm_all$Tm <- c(81.16, 81.33, 81.46, 83.00, 82.96, 83.09, 82.88, 82.90,
                 82.83, 79.25, 79.52, 79.27, 80.15, 79.72, 80.05, 79.13, 80.08, 78.87)


# calculate the Tm value for each protein using each TR from each N (9 total data points)
 Tm_all_avg <- Tm_all %>%
  group_by(., prot) %>%
  summarise(mean = mean(Tm),
            sd = sd(Tm))

 # T test using all TR (9)
  t_test_Tm <- Tm_all %>%
   t.test(Tm ~ prot, data = .)

#visualize Tm values as box plot with jitter to show all data points
# n=3 with 3 tr each
Tm_all %>%
  ggplot(aes(x = prot, y = Tm, color = prot)) +
  geom_boxplot(width = 0.5, lwd = 0.75) +
   geom_jitter(size = 2, alpha = 0.8) +
 scale_color_manual(values = c("Fis1" = "#29ABE2", "Fis1dN" = "#EC1C24")) +
 scale_y_continuous(limits = c(78.75, 83.25), breaks = c(79, 80, 81, 82, 83)) +
  labs(title = "Tm values", x = "Protein",
       y = "Tm (C)") +
  theme(legend.text = element_text(size = 12),
        legend.title = element_blank(),
        plot.title = element_text(hjust = 0.5),
        axis.text.x = element_text(size = 12),
        axis.text.y = element_text(size = 12))
```

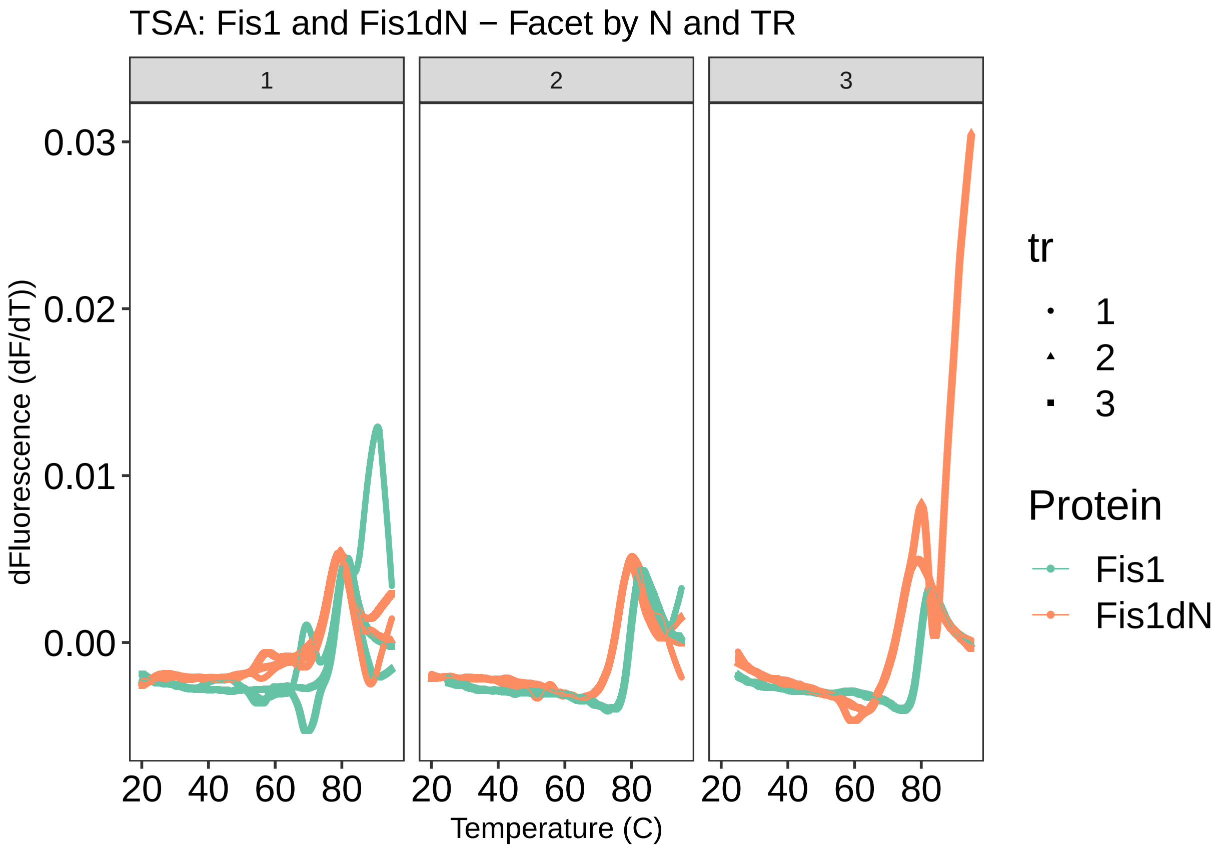


```
ggsave("Fis1_Fis1dN_Tm_boxplot.pdf",
       width = 6, height = 5, dpi = 300, useDingbats = FALSE)


# Average tr
# calculate average Fluorescence for each N (average of the 3 TR)
results_avg <- merged %>%
  group_by(., temp, prot, n) %>%
  summarise(., mean_fluor = mean(Fluorescence),
            sd = sd(Fluorescence)) %>%
  ungroup()
```

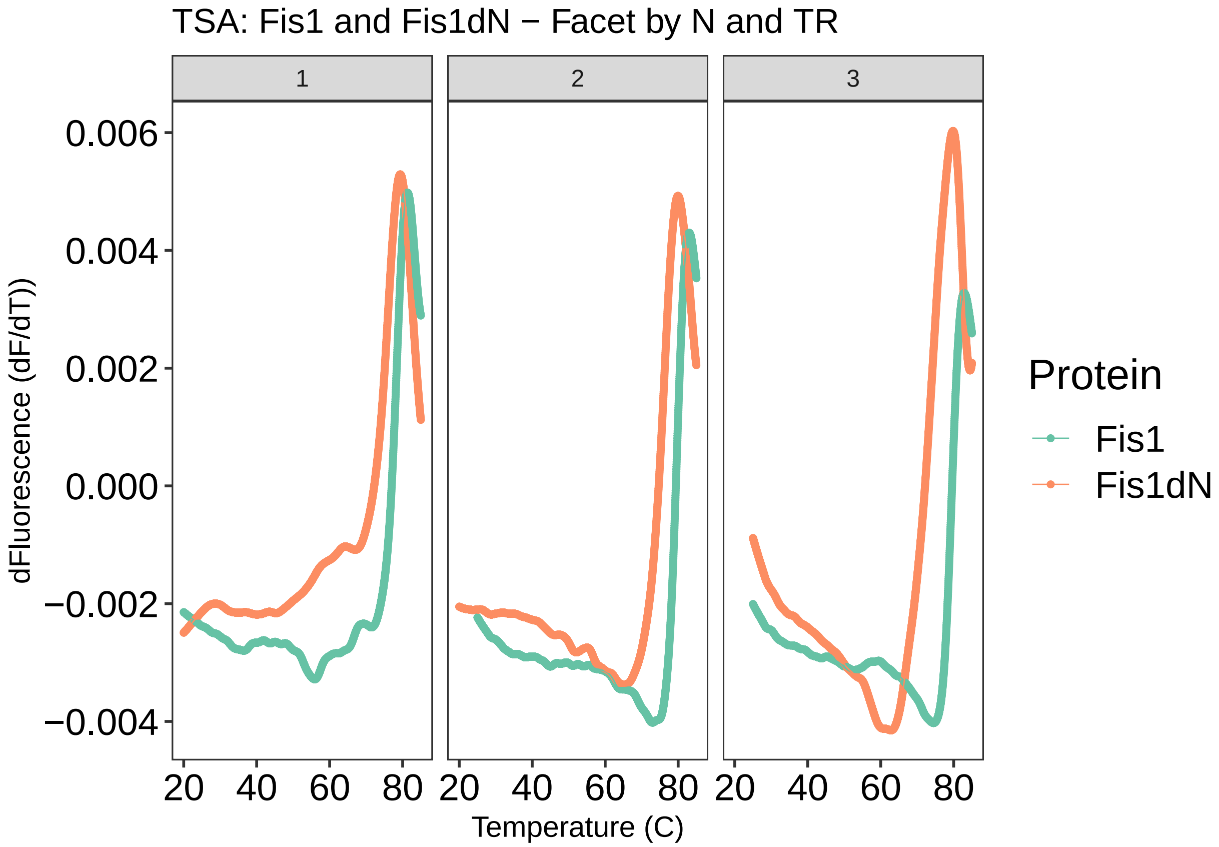
```
## `summarise()` has grouped output by 'temp', 'prot'. You can override using the `.groups` argument.
```

```
# Visualize data and facet by biological replicate ----------------------
merged %>%
  ggplot(., aes(x = temp, y = Fluorescence, color = prot, shape = tr)) +
  geom_point(size = 0.75) +
  geom_line(size = 0.25) +
  scale_color_brewer(palette = "Set2") +
  facet_wrap(n ~ .) +
  labs(title = "TSA: Fis1 and Fis1dN - Facet by N and TR",
       color = "Protein",
       x = "Temperature (C)",
       y = "dFluorescence (dF/dT))") +
  theme_bw() +
  theme(axis.text = element_text(size = 14, color = "black"),
        #axis.title.x = element_text(size = 18),
        #axis.title.y = element_text(size = 18),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        #panel.border = element_blank(),
        #panel.background = element_blank(),
        #axis.line.x = element_line(colour = 'black', size=1, linetype='solid'),
        #axis.line.y = element_line(colour = 'black', size=1, linetype='solid'),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 14, color = "black")
  )
```



```
ggsave("1st_deriv_melt_curve_facet_n_tr.pdf",
       width = 35, height = 16, units = "cm")

# Plotting averaged results (avg of Tm) - still split by n
results_avg %>%
  filter(., temp < 85) %>%
  ggplot(., aes(x = temp, y = mean_fluor, color = prot)) +
  geom_point(size = 0.75) +
  geom_line(size = 0.25) +
  scale_color_brewer(palette = "Set2") +
  facet_wrap(n ~ .) +
  labs(title = "TSA: Fis1 and Fis1dN - Facet by N and TR",
       color = "Protein",
       x = "Temperature (C)",
       y = "dFluorescence (dF/dT))") +
  theme_bw() +
  theme(axis.text = element_text(size = 14, color = "black"),
        panel.grid.major = element_blank(),
        panel.grid.minor = element_blank(),
        legend.title = element_text(size = 16),
        legend.text = element_text(size = 14, color = "black")
  )
```



```
ggsave("1st_deriv_melt_curve_facet_n.pdf",
       width = 35, height = 16, units = "cm")

# Compute average of Tm values of the technical replicates
# Filter out >80 C temps due to large fluorescence increase in one of the dN samples
# likely due to aggregation post unfolding

Tm_stats <- results_avg %>%
  group_by(., prot, n) %>%
  summarise(max = max(mean_fluor))
```

```
## `summarise()` has grouped output by 'prot'. You can override using the `.groups` argument.
```

```
Tm_stats
```

```
## # A tibble: 6 x 3
## # Groups:   prot [2]
##   prot   n          max
##   <chr>  <chr>    <dbl>
## 1 Fis1   1      0.00498
## 2 Fis1   2      0.00430
## 3 Fis1   3      0.00328
## 4 Fis1dN 1      0.00529
## 5 Fis1dN 2      0.00493
## 6 Fis1dN 3      0.0101
```

```
# Parsed results_avg to find temperature corresponding to max fluor
# Note: Fis1 dN n=3 had erroneous high fluor at end of run due to aggregation

Tm_stats$Tm <- c(81.33, 83.03, 82.88, 79.35, 79.95, 79.82)

# Determine avg Tm and SD from 3 biological replicates
Tm_stats_avg <- Tm_stats %>%
  summarise(mean = mean(Tm),
            sd = sd(Tm))

# T test using all TR (9)
Tm_stats_Ttest <- Tm_stats %>%
  t.test(Tm ~ prot, data = .)
```