UK_Drp1_WT_variants_TSA_merged.R

kelse

2022-06-01

```
# Plotting combined Thermal Shift Assay (TSA) data
# Amplification data = Temperature vs Fluorescence
# Sypro-Orange thermofluor TSA on Drp1 WT and UK variants +/- 100 or 500 uM GDP or GTP
# Data collected on 20210514 and 20210602
# 1st derivative of fluorescence with respect to temperature used to determine
# melting temperature (Tm)


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(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)
library(forcats)

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 "_"
raw <- read_excel("UK_Drp1_WT_variants_TSA_merged.xlsx", sheet = 1)

# Import raw data from both biological replicates and tidy
# Add additional column to signify which biologic replicate it is
n_1 <- read_excel("20210514_Drp1_WT_variants_GTP_GDP_ed_noG363D1.xlsx", sheet = 1)%>%
  gather(., "tmp1", "Fluorescence", 2:90) %>%
  separate(., col = tmp1, into = c("prot", "ligand", "tr"),
           sep = "_") %>%
  mutate(., Fluorescence = as.double(Fluorescence))%>%
  arrange(., prot, ligand) %>%
  mutate(dF = Fluorescence - lag(Fluorescence, default = first(Fluorescence)))

n_1$biorep <- "1"

n_2 <- read_excel("20210602_Drp1_WT_UK_variants_GDP_GTP_dN_edit.xlsx", sheet = 1) %>%
  gather(., "tmp1", "Fluorescence", 2:70) %>%
  separate(., col = tmp1, into = c("prot", "ligand", "tr"),
           sep = "_") %>%
  mutate(., Fluorescence = as.double(Fluorescence))%>%
  arrange(., prot, ligand) %>%
  mutate(dF = Fluorescence - lag(Fluorescence, default = first(Fluorescence)))

n_2$biorep <- "2"

# Merge data into one data frame
merged_data <- n_1 %>%
  union(., n_2)

# calculate average Fluorescence
results_avg <- raw %>%
  group_by(., prot, ligand, trans) %>%
  summarise(., mean_Tm = mean(Tm_temp),
            sd = sd(Tm_temp)) %>%
  ungroup() %>%
  arrange(., prot, ligand)
```

```
## ‘summarise()‘ has grouped output by ’prot’, ’ligand’. You can override using the ‘.groups‘ argument.
```

```r
results_avg
```

```
## # A tibble: 24 x 5
##    prot  ligand trans mean_Tm    sd
##    <chr> <chr>  <dbl>   <dbl> <dbl>
##  1 G363D 0          1    47.7 0.516
##  2 G363D 0          2    66   0
##  3 G363D 500GDP     1    51   1.10
##  4 G363D 500GDP     2    66.7 1.03
##  5 G363D 500GTP     1    49.6 0.548
##  6 G363D 500GTP     2    66   0
##  7 G401S 0          1    47.5 0.548
##  8 G401S 0          2    70.8 1.60
##  9 G401S 500GDP     1    51   0
## 10 G401S 500GDP     2    69.5 0.548
## # ... with 14 more rows
```

```
# calculate average dF from all tr
merged_avg <- merged_data %>%
  group_by(., Temperature, prot, ligand) %>%
  summarise(., mean_dF = mean(dF),
            sd = sd(dF)) %>%
  ungroup() %>%
  arrange(., prot, ligand)
```

```
## ‘summarise()‘ has grouped output by ’Temperature’, ’prot’. You can override using the ‘.groups‘ argu
```

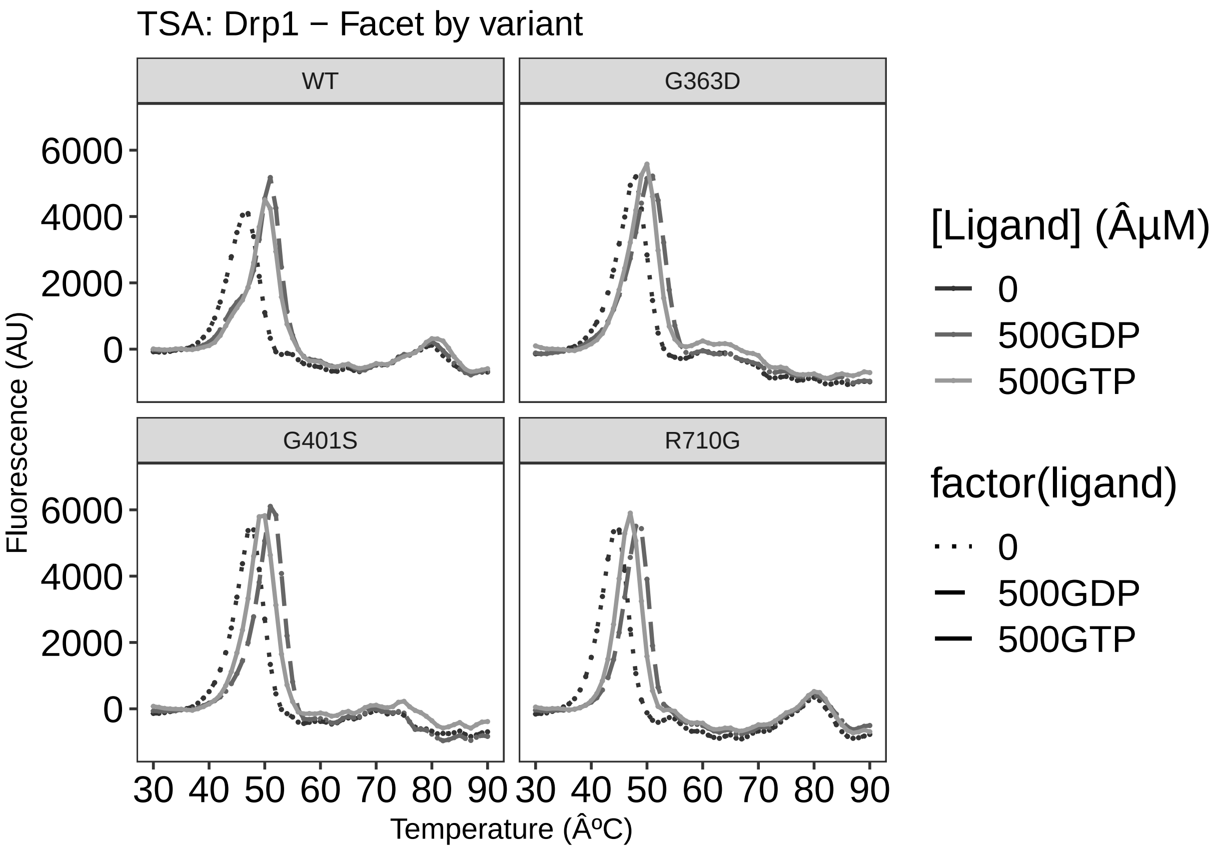
```
merged_avg
```

```
## # A tibble: 2,698 x 5
##    Temperature prot  ligand  mean_dF     sd
##          <dbl> <chr> <chr>     <dbl>  <dbl>
##  1          25 dN10  Drp1(1)   -958. 954.
##  2          26 dN10  Drp1(1)  -1009.  18.4
##  3          27 dN10  Drp1(1)   -475.  19.3
##  4          28 dN10  Drp1(1)   -279.  11.8
##  5          29 dN10  Drp1(1)   -250.  12.2
##  6          30 dN10  Drp1(1)   -252.  10.7
##  7          31 dN10  Drp1(1)   -238.   4.32
##  8          32 dN10  Drp1(1)   -217.   5.90
##  9          33 dN10  Drp1(1)   -213.   7.88
## 10          34 dN10  Drp1(1)   -222.   6.48
## # ... with 2,688 more rows
```

```
# set factor levels for protein to specify facet_wrap order

merged_avg$prot <- factor(merged_avg$prot, levels =c("WT", "G363D", "G401S", "R710G"))

raw$prot <- factor(raw$prot, levels =c("WT", "G363D", "G401S", "R710G"))

# Visualize thermal melt curves (1st deriv) faceted by protein
merged_avg %>%
  group_by(., prot) %>%
  filter(., ligand != "100GDP" & ligand != "100GTP") %>%
  filter(., prot != "L230dup" & prot != "L230dup2" & prot != "dN10"
         & prot != "dN5" & prot != "dN15") %>%
  ggplot(., aes(x = Temperature, y = mean_dF, color = factor(ligand), linetype = factor(ligand))) +
  geom_point(size = 0.25) +
  geom_line(size = 0.75) +
  scale_linetype_manual(values = c("0"="dotted", "500GDP"="longdash", "500GTP"="solid")) +
  scale_color_manual(values = c("0"="grey20", "500GDP"="grey40", "500GTP"="grey60")) +
  scale_x_continuous(limits = c(30, 90), breaks = c(30, 40, 50, 60, 70, 80, 90)) +
  scale_y_continuous(limits = c(-1200, 7000), breaks = c(0, 2000, 4000, 6000)) +
  facet_wrap(prot ~ .) +
  labs(title = "TSA: Drp1 - Facet by variant",
       color = "[Ligand] (ÂµM)",
       x = "Temperature (Âºc)",
       y = "Fluorescence (AU)") +
  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")
  )
```

```
## Warning: Removed 120 rows containing missing values (geom_point).
```

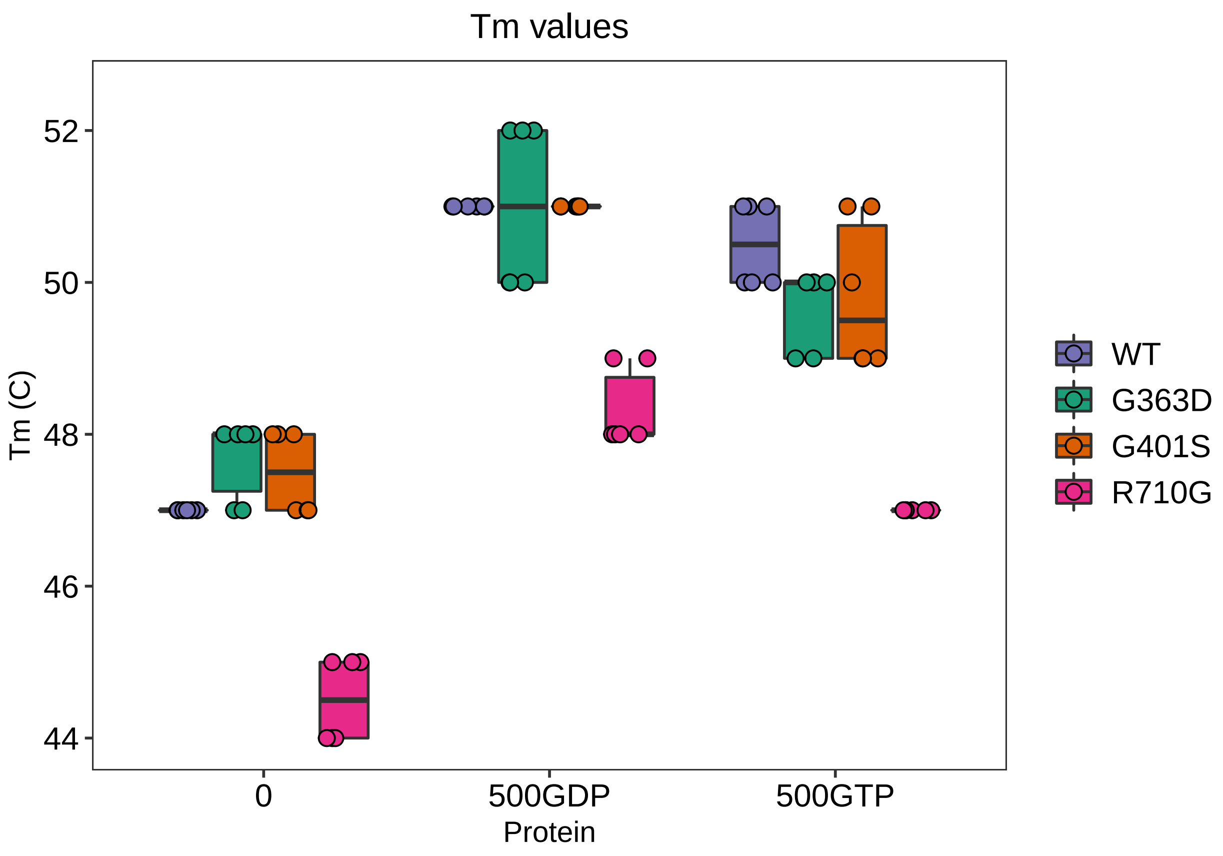
```
## Warning: Removed 30 row(s) containing missing values (geom_path).
```



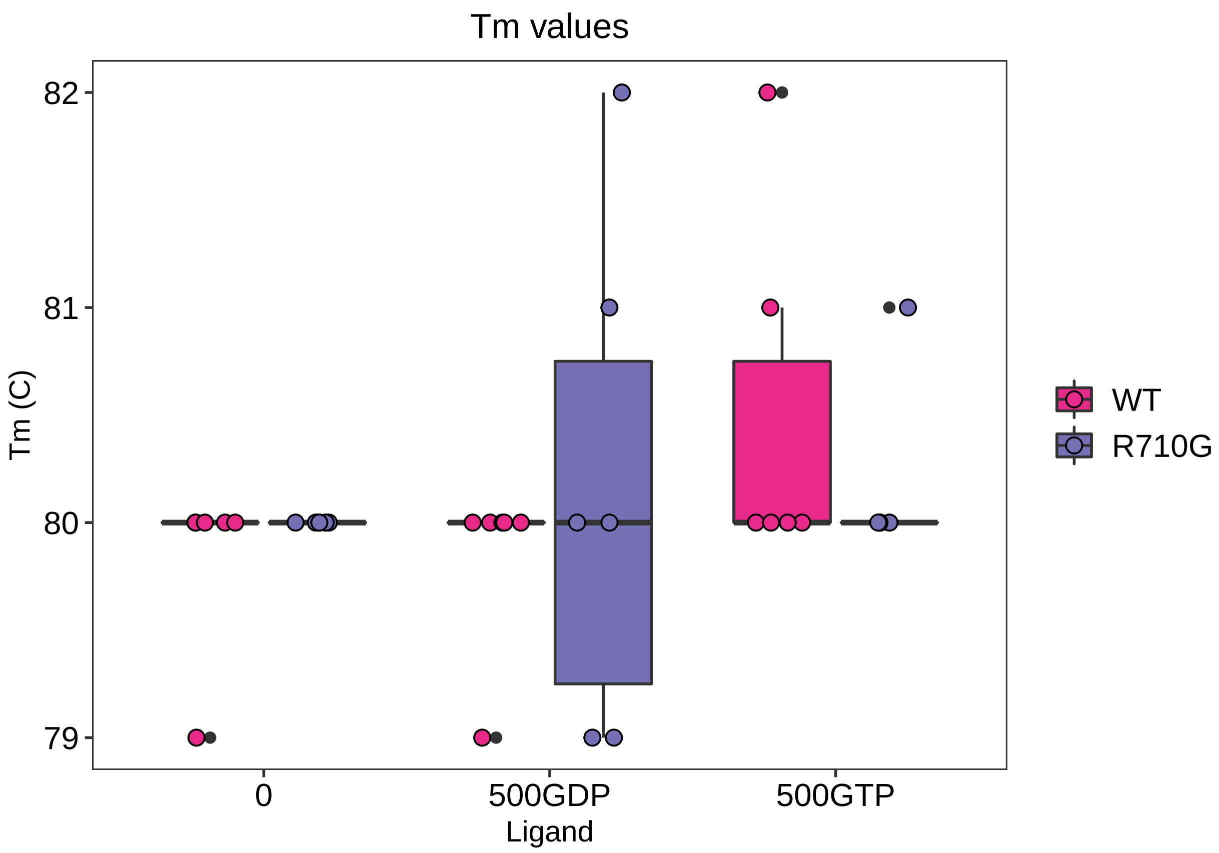
```
ggsave("melt_curve_facet_prot_1stderiv_linetype.pdf",
       width = 35, height = 16, units = "cm")
```

```
## Warning: Removed 120 rows containing missing values (geom_point).

## Warning: Removed 30 row(s) containing missing values (geom_path).
```

```
#visualize Tm values as bar plot
# Transition 1 - box plot with jitter
raw %>%
  filter(., trans == "1") %>%
  ggplot(aes(x = ligand, y = Tm_temp, fill = factor(prot))) +
  geom_boxplot() +
  scale_fill_manual(values = c("WT" = "#7570B3",
                                 "G363D" = "#1B9E77",
                                 "G401S" = "#D95F02",
                                 "R710G" = "#E7298A")) +
    geom_point(pch = 21, size = 2.5, position = position_jitterdodge()) +
  coord_cartesian(ylim = c(44,52.5)) +
  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("Drp1_WT_variants_tm_trans1_boxandjitter_plot.pdf",
       width = 9, height = 5, dpi = 300, useDingbats = FALSE)

# Transition 2 - box plot with jitter
raw %>%
  filter(., trans == "2" & prot != "G363D" & prot != "G401S") %>%
  ggplot(aes(x = ligand, y = Tm_temp, fill = factor(prot))) +
  geom_boxplot() +
  scale_fill_manual(values = c(WT = "#E7298A", R710G = "#7570B3")) +
  geom_point(pch = 21, size = 2.5, position = position_jitterdodge()) +
  labs(title = "Tm values", x = "Ligand",
       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("Drp1_WT_variants_tm_trans2_boxandjitter_plot.pdf",
       width = 9, height = 5, dpi = 300, useDingbats = FALSE)

### Perform ANOVA to determine if rates are significantly different
# Filter by ligand and transition (ie transition 1 or 2)
anova_trans1_0 <- raw %>%
  filter(., ligand == "0" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ prot, data = .)))

anova_trans1_GDP <- raw %>%
  filter(., ligand == "500GDP" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ prot, data = .)))

anova_trans1_GTP <- raw %>%
  filter(., ligand == "500GTP" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ prot, data = .)))

anova_trans1_WT <- raw %>%
  filter(., prot == "WT" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ ligand, data = .)))

anova_trans1_G363D <- raw %>%
  filter(., prot == "G363D" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ ligand, data = .)))

anova_trans1_G401S <- raw %>%
  filter(., prot == "G401S" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ ligand, data = .)))

anova_trans1_R710G <- raw %>%
  filter(., prot == "R710G" & trans == "1") %>%
  do(tidy(aov(Tm_temp ~ ligand, data = .)))

t_test_0_trans1 <- raw %>%
  filter(., ligand == "0" & trans == "1") %>%
  filter(., prot == "WT" | prot == "R710G") %>%
  t.test(Tm_temp ~ prot, data = .)

t_test_0_trans2 <- raw %>%
  filter(., ligand == "0" & trans == "2") %>%
  filter(., prot == "WT" | prot == "R710G") %>%
  t.test(Tm_temp ~ prot, data = .)

### For any p-value < 0.05 from ANOVA, run post-hoc Tukey test
anova_tukey_trans1_0 <- raw %>%
  filter(., ligand == "0" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ prot, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_0, file = "anova_tukey_trans1_0.csv")

anova_tukey_trans1_GDP <- raw %>%
  filter(., ligand == "500GDP" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ prot, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_GDP, file = "anova_tukey_trans1_GDP.csv")

anova_tukey_trans1_GTP <- raw %>%
  filter(., ligand == "500GTP" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ prot, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_GDP, file = "anova_tukey_trans1_GTP.csv")

anova_tukey_WT_trans1 <- raw %>%
  filter(., prot == "WT" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ ligand, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_GDP, file = "anova_tukey_WT_trans1.csv")

anova_tukey_G363D_trans1 <- raw %>%
  filter(., prot == "G363D" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ ligand, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_GDP, file = "anova_tukey_G363D_trans1.csv")
```

```r
anova_tukey_G401S_trans1 <- raw %>%
  filter(., prot == "G401S" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ ligand, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_GDP, file = "anova_tukey_G401S_trans1.csv")

anova_tukey_R710G_trans1 <- raw %>%
  filter(., prot == "R710G" & trans == "1") %>%
  do(tidy(TukeyHSD(aov(Tm_temp ~ ligand, data = .)))) %>%
  arrange(adj.p.value)

write.csv(anova_tukey_trans1_GDP, file = "anova_tukey_R710G_trans1.csv")
```