

Vignette

Phenological characterization and modeling of diverse lentil (*Lens culinaris* Medik.) germplasm grown in multiple environments

Derek Wright

2019-09-20

Contents

Preliminary Steps	2
Additional Figure 1 LDP Origin Map	5
Figure 1 Field Trial Info	6
Supplemental Figure 1 Scaling	7
Figure 2 Data Overview	8
Additional Figure 2 DTF DTS DTM	14
Additional Figure 3 MacroEnv Phenology	16
Additional Figures - Phenology pdf	17
Supplemental Figure 2 Missing Data	18
Supplemental Figure 3 Correlation Plots	19
Additional Figures - Correlations	21
PhotoThermal Plane	24
Figure 3 Regressions	26
Figure 4 Modeling DTF	29
Figure 5 Test Model	33
Supplemental Figure 4 Compare Constants Entry	38
Supplemental Figure 5 Compare Constants All	40
Base Temperature & Critical Photoperiod	41

Figure 6 Tb and Pc	42
Figure 7 Tf	43
Additional Figure 7 Pc	44
Additional Figure 8 Tf Pc PTT	45
Supplemental Figure 6 Thermal Sums	47
Supplemental Figure 7 Photoperiodic Sums	49
Figure 8 Temperature Increase By MacroEnv	51
Additional Figure 9 Temperature Increase By Cluster	52
Figure 9 Origin Constants	53
Supplemental Figure 8 Adapted	56
Figure 10 AMMI	60
Figure 11 PCA	61
Additional Figure 10 PCA	65
Supplemental Figure 9 Cluster Origins	65
Additional Figure 11 LDP Origins By Cluster	67
Additional Figure 12 DTF By Cluster	68

Preliminary Steps

Load the nessessary R packages, Prepare the data for analysis.

```
# Load libraries
library(tidyverse) # data wrangling
library(scales) # rescale()
library(rworldmap) # mapBubbles()
library(ggrepel) # geom_text_repel() + geom_label_repel()
library(magick) # image editing
library(GGally) # ggpairs() + ggmatrix()
library(ggpubr) # ggarrange()
library(ggbeeswarm) # geom_quasirandom()
```

```

library(agricolae) # AMMI()
library(FactoMineR) # PCA() & HCPC()
library(plot3D) # 3D plots
# General color palettes
colors <- c("darkred", "darkorange3", "darkgoldenrod2", "deeppink3",
           "steelblue", "darkorchid4", "cornsilk4", "darkgreen")
# Expts color palette
colors_Expt <- c("lightgreen", "palegreen4", "darkgreen", "darkolivegreen3",
                  "darkolivegreen4", "springgreen4", "orangered2", "orangered4",
                  "palevioletred", "mediumvioletred", "orange2", "orange4",
                  "slateblue1", "slateblue4", "aquamarine3", "aquamarine4",
                  "deepskyblue3", "deepskyblue4" )
# Locations
names_Location <- c("Rosthern, Canada", "Sutherland, Canada", "Central Ferry, USA",
                      "Bhopal, India", "Jessore, Bangladesh", "Bardiya, Nepal",
                      "Cordoba, Spain", "Marchouch, Morocco", "Metaponto, Italy" )
# Experiments
names_Expt <- c("Rosthern, Canada 2016", "Rosthern, Canada 2017",
                 "Sutherland, Canada 2016", "Sutherland, Canada 2017",
                 "Sutherland, Canada 2018", "Central Ferry, USA 2018",
                 "Bhopal, India 2016", "Bhopal, India 2017",
                 "Jessore, Bangladesh 2016", "Jessore, Bangladesh 2017",
                 "Bardiya, Nepal 2016", "Bardiya, Nepal 2017",
                 "Cordoba, Spain 2016", "Cordoba, Spain 2017",
                 "Marchouch, Morocco 2016", "Marchouch, Morocco 2017",
                 "Metaponto, Italy 2016", "Metaponto, Italy 2017" )
# Experiment short names
names_ExptShort <- c("Ro16", "Ro17", "Su16", "Su17", "Su18", "Us18",
                      "In16", "In17", "Ba16", "Ba17", "Ne16", "Ne17",
                      "Sp16", "Sp17", "Mo16", "Mo17", "It16", "It17" )
# Macro-Environments
macroEnvs <- c("Temperate", "South Asia", "Mediterranean")
# Scaling function
traitScale <- function(x, trait) {
  xout <- rep(NA, nrow(x))
  for(i in unique(x$Expt)) {
    mn <- x %>% filter(Expt == i) %>% pull(trait) %>% min(na.rm = T)
    mx <- x %>% filter(Expt == i) %>% pull(trait) %>% max(na.rm = T)
    rg <- mx - mn
    xout <- ifelse(x$Expt == i, rescale(x %>% pull(trait), c(1,5), c(mn,mx)), xout)
  }
  xout
}
# ggplot theme
theme_AGL <- theme_bw() + theme(strip.background = element_rect(fill = "White"))
#
# Prep data
# Note: DTF2 = non-flowering genotypes <- group_by(Expt) %>% max(DTF)
rr <- read.csv("data/data_Raw.csv") %>%
  mutate(Rep = factor(Rep), Year = factor(Year), PlantingDate = as.Date(PlantingDate),
         Location = factor(Location, levels = names_Location),
         Expt = factor(Expt, levels = names_Expt),
         ExptShort = plyr::mapvalues(Expt, names_Expt, names_ExptShort),

```

```

ExptShort    = factor(ExptShort, levels = names_ExptShort),
DTF2_scaled = traitScale(., "DTF2"),
RDTF = round(1 / DTF2, 6),
VEG   = DTF - DTE,
REP   = DTM - DTF)
# Average raw data
dd <- rr %>% group_by(Entry, Name, Expt, ExptShort, Location, Year) %>%
  summarise_at(vars(DTE, DTF, DTS, DTM, VEG, REP, RDTF, DTF2, DTF2_scaled),
               funs(mean, na.rm = T)) %>% ungroup()
# Prep environmental data
ee <- read.csv("data/data_Env.csv") %>%
  mutate(Date      = as.Date(Date),
         ExptShort = plyr::mapvalues(Expt, names_Expt, names_ExptShort),
         ExptShort = factor(ExptShort, levels = names_ExptShort),
         Expt      = factor(Expt,      levels = names_Expt),
         Location  = factor(Location, levels = names_Location),
         DayLength_rescaled = rescale(DayLength, to = c(0, 40)) )
# Prep field trial info
xx <- dd %>% group_by(Expt) %>%
  summarise_at(vars(DTE, DTF, DTS, DTM), funs(min, mean, max), na.rm = T) %>%
  ungroup()
ff <- read.csv("data/data_Info.csv") %>%
  mutate(Start = as.Date(Start) ) %>%
  left_join(xx, by = "Expt")
for(i in unique(ee$Expt)) {
  ee <- ee %>%
    filter(Expt != i | (Expt == i & DaysAfterPlanting <= ff$DTM_max[ff$Expt == i]))
}
xx <- ee
for(i in unique(ee$Expt)) {
  xx <- xx %>%
    filter(Expt != i | (Expt == i & DaysAfterPlanting <= ff$DTF_max[ff$Expt == i]))
}
xx <- xx %>% group_by(Location, Year) %>%
  summarise(T_mean = mean(Temp_mean, na.rm = T), T_sd = sd(Temp_mean, na.rm = T),
            P_mean = mean(DayLength, na.rm = T), P_sd = sd(DayLength, na.rm = T)) %>%
  ungroup() %>%
  mutate(Expt = paste(Location, Year)) %>%
  select(-Location, -Year)
ff <- ff %>% left_join(xx, by = "Expt") %>%
  mutate(Expt      = factor(Expt,      levels = names_Expt),
         ExptShort = plyr::mapvalues(Expt, names_Expt, names_ExptShort),
         ExptShort = factor(ExptShort, levels = names_ExptShort),
         MacroEnv  = factor(MacroEnv, levels = macroEnvs),
         T_mean = round(T_mean, 1),
         P_mean = round(P_mean, 1))
# Lentil Diversity Panel metadata
ldp <- read.csv("data/data_LDP.csv")
# Country info
ct <- read.csv("data/data_Countries.csv") %>% filter(Country %in% ldp$Origin)
#
# Supplemental tables
#

```

```

write.csv(select(ldp, Entry, Name, Origin, Source),
          "SupplementalTable1.csv", row.names = F)
write.csv(select(ff, Location, Year, `Short Name` = ExptShort, `Planting Date` = Start,
               `Temperature (mean)` = T_mean, `Photoperiod (mean)` = P_mean,
               Latitude = Lat, Longitude = Lon,
               `Number of Seeds Sown` = Number_of_Seeds_Sown, `Plot Type` = Plot_Type),
          "SupplementalTable2.csv", row.names = F)

```

- ldp = Lentil Diversity Panel Metadata
 - rr = Raw Data
 - dd = Averaged Data
 - ee = Environmental Data
 - ff = Field Trial Info
 - ct = Country Info
-

Additional Figure 1 LDP Origin Map

```

# Prep data
x1 <- ldp %>% filter(Origin != "ICARDA") %>%
  group_by(Origin) %>% summarise(Count = n()) %>%
  left_join(select(ct, Origin = Country, Lat, Lon), by = "Origin") %>%
  ungroup() %>% as.data.frame()
x1[is.na(x1)] <- 0
# Plot map
invisible(png("Additional/AdditionalFigure01_LDPOriginMap.png",
              width = 1200, height = 685, res = 150))
par(mai = c(0,0,0,0), xaxs = "i", yaxs = "i")
mapBubbles(dF = x1, nameX = "Lon", nameY = "Lat",
            nameZSize = "Count", nameZColour = "darkred",
            xlim = c(-140,110), ylim = c(5,20),
            oceanCol = "grey90", landCol = "white", borderCol = "black")
invisible(dev.off())

```

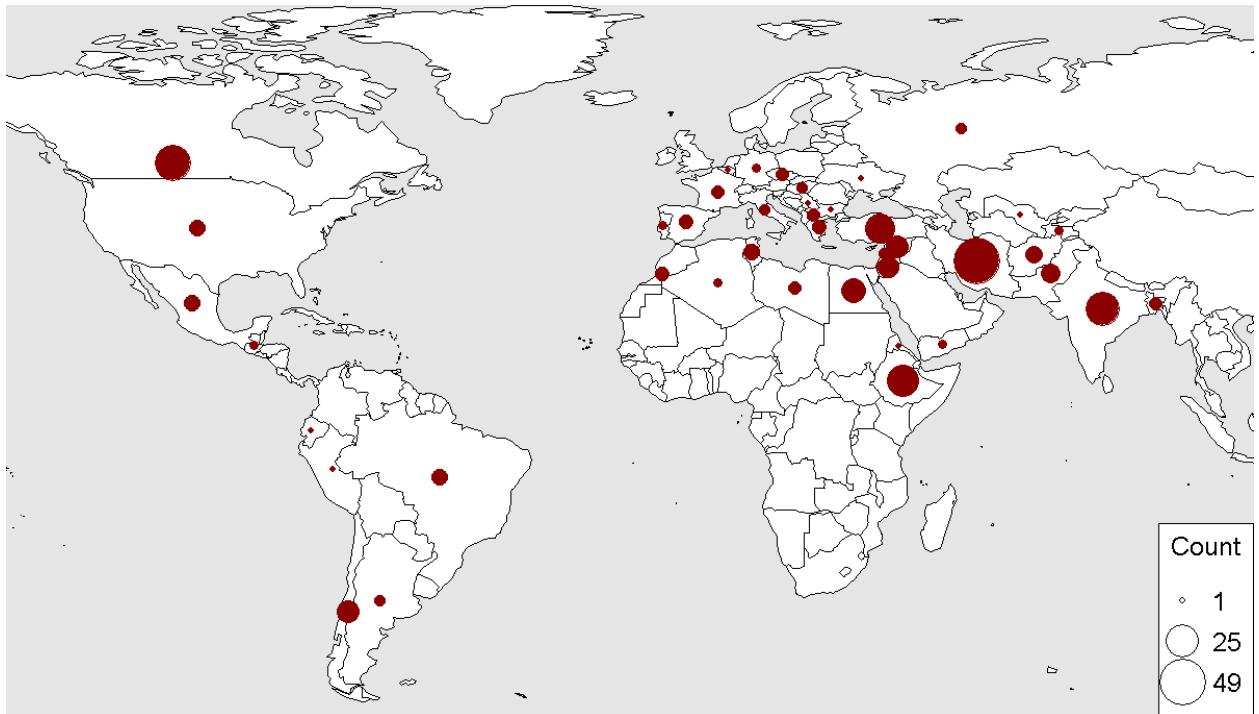


Figure 1 Field Trial Info

```

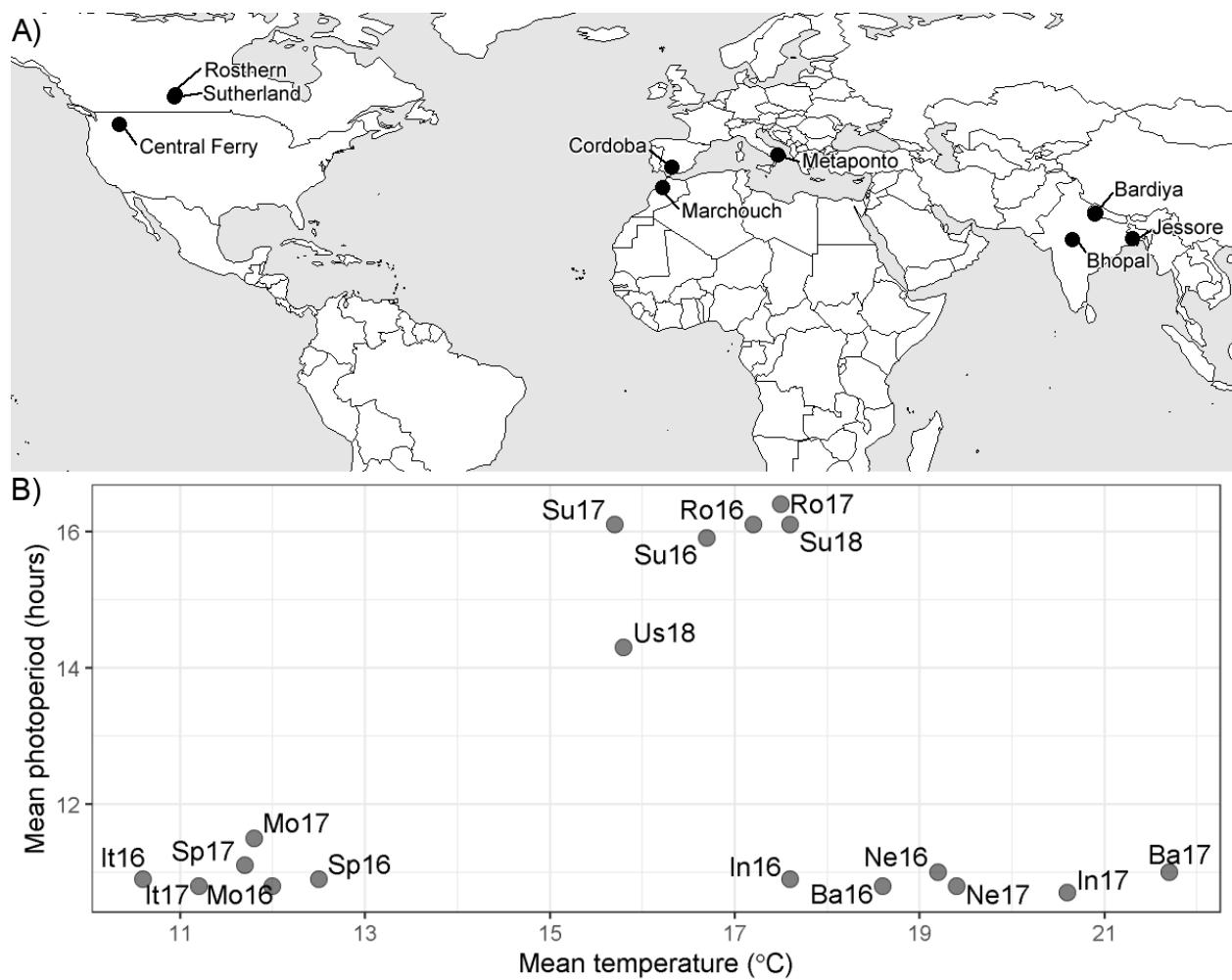
# Prep data
xx <- ff %>% mutate(size = 1)
# Plot A) Map
invisible(png("Temp/Temp_F1_1.png", width = 1200, height = 450, res = 150))
par(mai = c(0,0,0,0), xaxs = "i", yaxs = "i")
mapBubbles(dF = xx, nameX = "Lon", nameY = "Lat", nameZColour = "MacroEnv",
           nameZSize = "size", symbolSize = 0.5, pch = 20, fill = F,
           colourPalette = rep("black", 3), addColourLegend = F, addLegend = F,
           xlim = c(-140,110), ylim = c(10,35),
           oceanCol = "grey90", landCol = "white", borderCol = "black")
invisible(dev.off())
# Plot B) mean T and P
mp <- ggplot(ff, aes(x = T_mean, y = P_mean)) +
  geom_point(size = 3, color = "black", alpha = 0.5) +
  geom_text_repel(aes(label = ExptShort)) +
  scale_x_continuous(breaks = c(11,13,15,17,19,21)) +
  theme_AGL + theme(legend.position = "none") +
  labs(x = expression(paste("Mean temperature (", degree, "C)", sep = "")),
       y = "Mean photoperiod (hours)")
ggsave("Temp/Temp_F1_2.png", mp, width = 7, height = 3)
#
# Labels were added to "Temp/Temp_F1_1.png" in image editing software
# Append A) and B)
im1 <- image_read("Temp/Temp_F1_1_1.png") %>%

```

```

image_annotate("A)", size = 30, boxcolor = "white")
im2 <- image_read("Temp/Temp_F1_2.png") %>%
  image_scale("1200x") %>%
  image_annotate("B)", size = 30, boxcolor = "white")
im <- image_append(c(im1,im2), stack = T)
image_write(im, "Figure1_FieldTrialInfo.png")

```



Supplemental Figure 1 Scaling

```

# Prep data
levs <- c("Days from sowing to flower (days)", "Scaled (1-5)")
xx <- dd %>% select(Entry, Expt, ExptShort, DTF, DTF2_scaled) %>%
  left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  gather(Trait, Value, DTF, DTF2_scaled) %>%
  mutate(Trait = plyr::mapvalues(Trait, c("DTF", "DTF2_scaled"), levs),
        Trait = factor(Trait, levels = levs) )

```

```

# Plot DTF
mp <- ggplot(xx, aes(x = ExptShort, y = Value)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +
  geom_quasirandom(size = 0.1, alpha = 0.5) +
  facet_grid(Trait~MacroEnv, scales = "free") +
  theme_AGL + theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  labs(x = NULL, y = NULL)
ggsave("SupplementalFigure1_Scaling.png", mp, width = 8, height = 5)

```

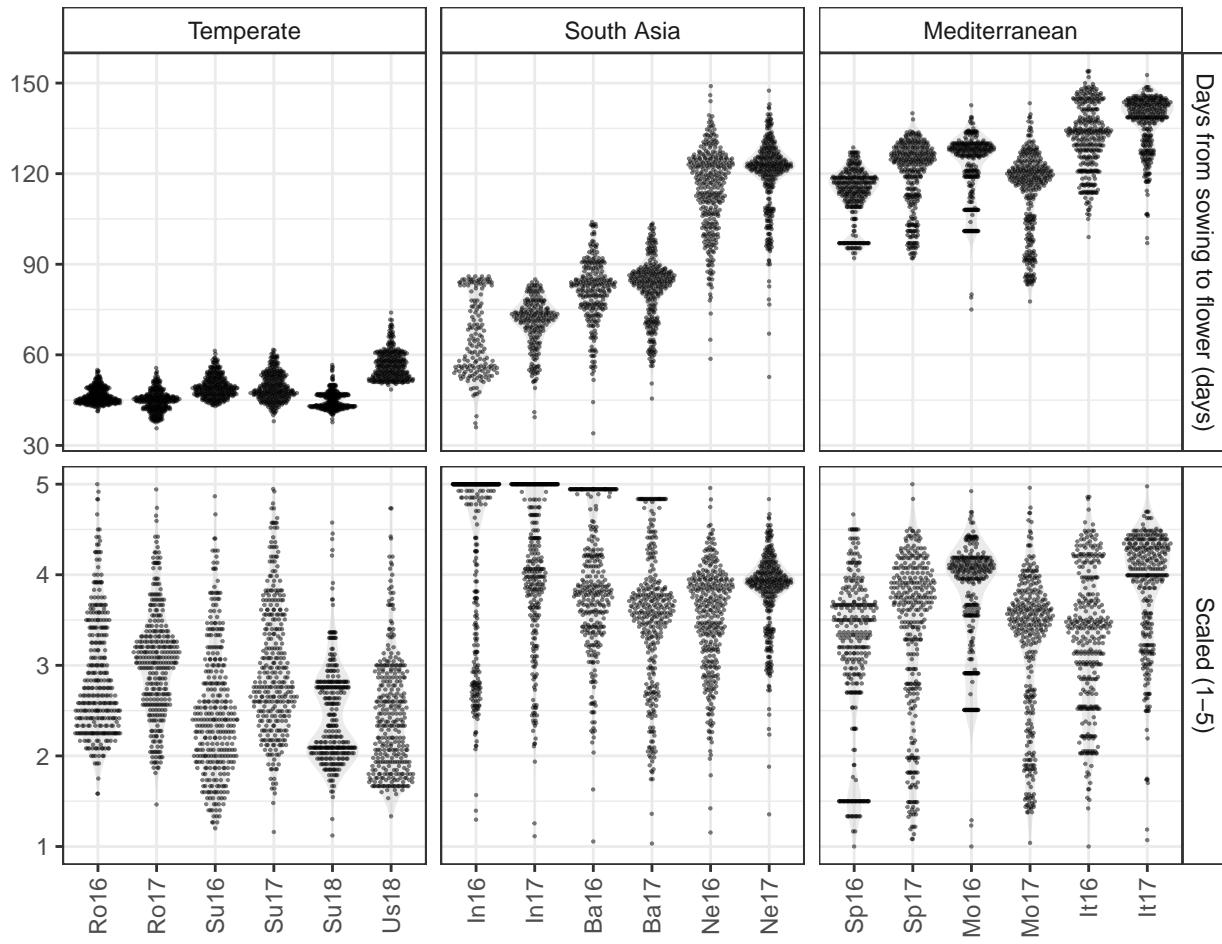


Figure 2 Data Overview

```

# Prep data
xx <- dd %>% select(Entry, Year, Expt, ExptShort, Location, DTF, DTS, DTM) %>%
  left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  gather(Trait, Value, DTF, DTS, DTM) %>%
  mutate(Trait = factor(Trait, levels = c("DTF", "DTS", "DTM")))
# Create plot function

```

```

ggDistroDTF <- function(x) {
  mp <- ggplot(x, aes(x = Trait, y = Value) ) +
    geom_violin(color = NA, fill = "grey", alpha = 0.3) +
    geom_quasirandom(size = 0.02) +
    facet_wrap(ExptShort~., scales = "free_x", dir = "v", ncol = 3, nrow = 2) +
    scale_y_continuous(sec.axis = dup_axis(name = "\n"),
                       limits = c(30,190), breaks = c(50,75,100,125,150,175)) +
    theme_AGL + theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
    labs(x = NULL, y = NULL)
  mp
}

# Plot A) DTF, DTS and DTM
mp1.1 <- ggDistroDTF(xx %>% filter(MacroEnv == "Temperate"))
mp1.2 <- ggDistroDTF(xx %>% filter(MacroEnv == "South Asia"))
mp1.3 <- ggDistroDTF(xx %>% filter(MacroEnv == "Mediterranean"))
mp1 <- ggmatrix(list(mp1.1, mp1.2, mp1.3), nrow = 1, ncol = 3,
                title = "A)", ylab = "Days After Planting",
                xAxisLabels = macroEnvs) +
  theme_AGL +
  theme(plot.margin = unit(c(0,1,0,0), "cm"),
        plot.title = element_text(hjust = -0.04))
ggsave("Temp/Temp_F2_1.png", mp1, width = 12, height = 4)

# Prep data
xx <- dd %>% select(Entry, Name, Expt, ExptShort, Location, Year, VEG, REP) %>%
  left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  gather(Trait, Value, VEG, REP) %>%
  mutate(Trait = factor(Trait, levels = c("VEG", "REP")))
# Create plot function
ggDistroREP <- function(x) {
  mp <- ggplot(x, aes(x = Trait, y = Value)) +
    geom_violin(color = NA, fill = "grey", alpha = 0.3) +
    geom_quasirandom(size = 0.02) +
    facet_wrap(ExptShort~., scales = "free_x", dir = "v", ncol = 3, nrow = 2) +
    scale_y_continuous(sec.axis = dup_axis(name = "\n"),
                       limits = c(25,135), breaks = c(25,50,75,100,125)) +
    theme_AGL + labs(x = NULL, y = "Days")
  mp
}

# Plot B) REP and VEG
mp2.1 <- ggDistroREP(xx %>% filter(MacroEnv == "Temperate"))
mp2.2 <- ggDistroREP(xx %>% filter(MacroEnv == "South Asia"))
mp2.3 <- ggDistroREP(xx %>% filter(MacroEnv == "Mediterranean"))
mp2 <- ggmatrix(list(mp2.1, mp2.2, mp2.3),
                nrow = 1, ncol = 3, title = "B)", ylab = "Days") +
  theme_AGL +
  theme(plot.margin = unit(c(0,1,0,0), "cm"),
        plot.title = element_text(hjust = -0.04))
ggsave("Temp/Temp_F2_2.png", mp2, width = 12, height = 4)

# Create plot function
ggEnvPlot <- function(x, nr = 2, nc = 3, mybreaks) {
  yy <- ff %>% filter(Expt %in% unique(x$Expt)) %>%
    select(ExptShort, Location, Year, min=DTF_min, max=DTF_max)
  mp <- ggplot(x) +

```

```

geom_rect(data = yy, aes(xmin = min, xmax = max),
          ymin = 0, ymax = 40, alpha = 0.4) +
  geom_line(aes(x = DaysAfterPlanting, y = DayLength_rescaled, lty = "Day Length")) +
  geom_line(aes(x = DaysAfterPlanting, y = Temp_mean, lty = "Temperature")) +
  geom_ribbon(aes(x = DaysAfterPlanting, ymin = Temp_min, ymax = Temp_max),
              alpha = 0.3) +
  facet_wrap(ExptShort~., scales = "free_x", dir = "v", nrow = 2, ncol = 3) +
  scale_x_continuous(breaks = mybreaks) +
  scale_linetype_manual(name = NULL, values = c("dashed", "solid"),
                        breaks = c("Temperature", "Day Length")) +
  coord_cartesian(ylim=c(0, 40)) +
  theme_AGL + theme(legend.position = "bottom") +
  labs(y = NULL, x = NULL) +
  guides(colour = guide_legend(order = 1), fill = guide_legend(order = 2))
mp
}
# Plot C) T and P
mp3.1 <- ggEnvPlot(ee %>% filter(MacroEnv == "Temperate"), mybreaks = c(25,50,75)) +
  labs(title = "C)", y = expression(paste(degree, "Celcius"))) +
  theme(plot.margin = unit(c(0,0,0,0), "cm"),
        plot.title = element_text(hjust = -0.11))
mp3.2 <- ggEnvPlot(ee %>% filter(MacroEnv == "South Asia"), mybreaks = c(25,75,125)) +
  theme(plot.margin = unit(c(0,0,0,0), "cm"),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank())
mp3.3 <- ggEnvPlot(ee %>% filter(MacroEnv == "Mediterranean"), mybreaks = c(50,100,150)) +
  scale_y_continuous(sec.axis = sec_axis(~ (16.62 - 9.11) * . / (40 - 0) + 9.11,
                                         name = "Hours", breaks = c(10, 12, 14, 16))) +
  theme(plot.margin = unit(c(0,0,0,0), "cm"),
        axis.text.y.left = element_blank(),
        axis.ticks.y.left = element_blank())
mp3 <- ggarrange(mp3.1, mp3.2, mp3.3, nrow = 1, ncol = 3, align = "h",
                  legend = "bottom", common.legend = T, widths = c(1.1,1,1.1))
ggsave("Temp/Temp_F2_3.png", mp3, width = 12, height = 4)
# Append A), B) and C)
mp1 <- image_read("Temp/Temp_F2_1.png")
mp2 <- image_read("Temp/Temp_F2_2.png")
mp3 <- image_read("Temp/Temp_F2_3.png")
mp <- image_append(c(mp1, mp2, mp3), stack = T)
image_write(mp, "Figure2_DataOverview.png")

```

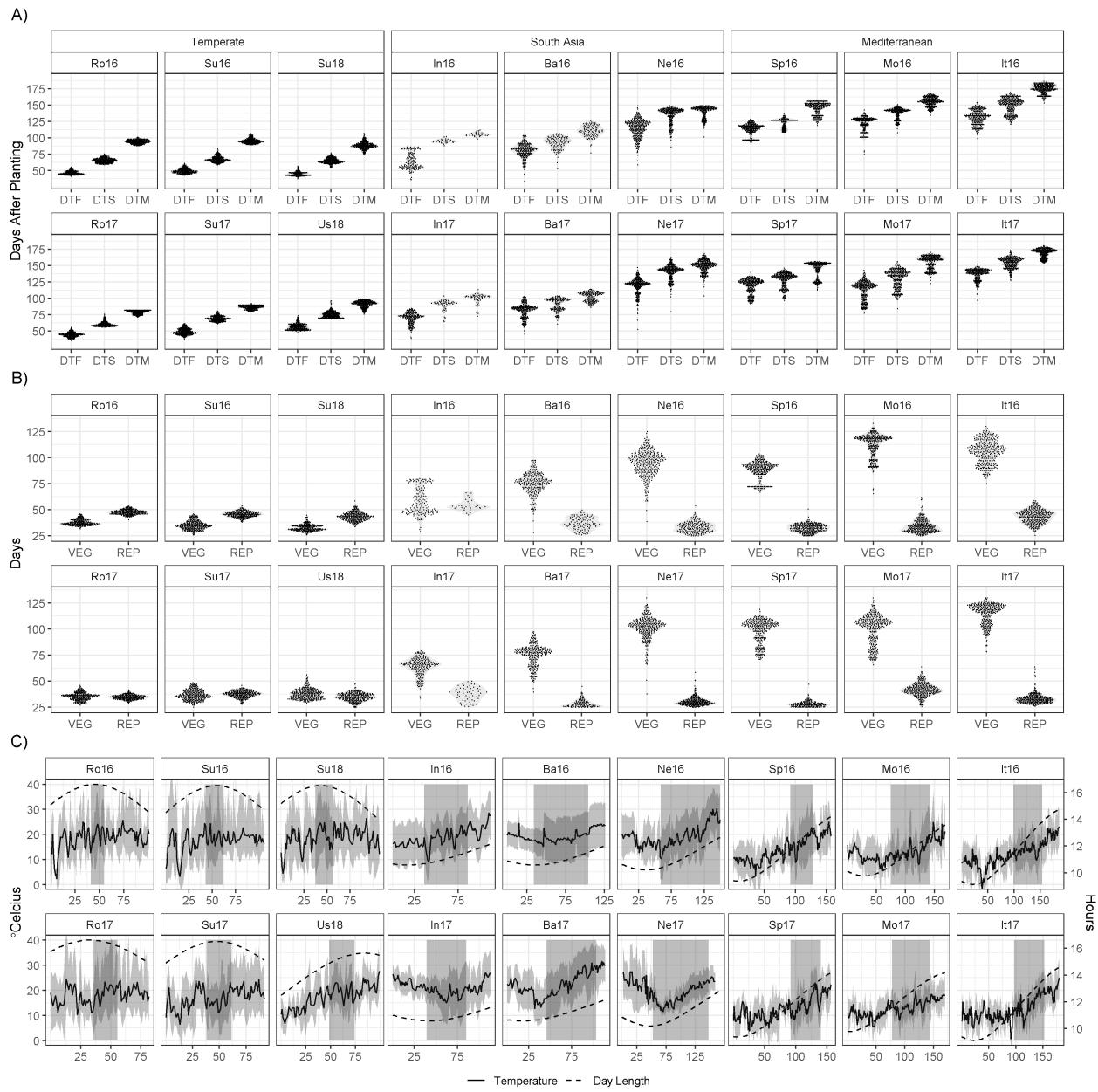


Figure 2 in Color

```
# Prep data
xx <- dd %>% select(Entry, Year, Expt, ExptShort, Location, DTF, DTS, DTM) %>%
  left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  gather(Trait, Value, DTF, DTS, DTM) %>%
  mutate(Trait = factor(Trait, levels = c("DTF", "DTS", "DTM")))
# Create plot function
ggDistroDTF <- function(x) {
  mp <- ggplot(x, aes(x = Trait, y = Value)) +
    geom_violin(color = NA, fill = "grey", alpha = 0.3) +
    geom_quasirandom(size = 0.02, aes(color = Trait)) +
    facet_wrap(ExptShort ~ ., scales = "free_x", dir = "v", ncol = 3, nrow = 2) +
    scale_color_manual(values = c("darkgreen", "darkred", "darkgoldenrod2")) +
    theme_minimal() +
    theme(panel.grid.major.x = element_line())
  return(mp)
}
```

```

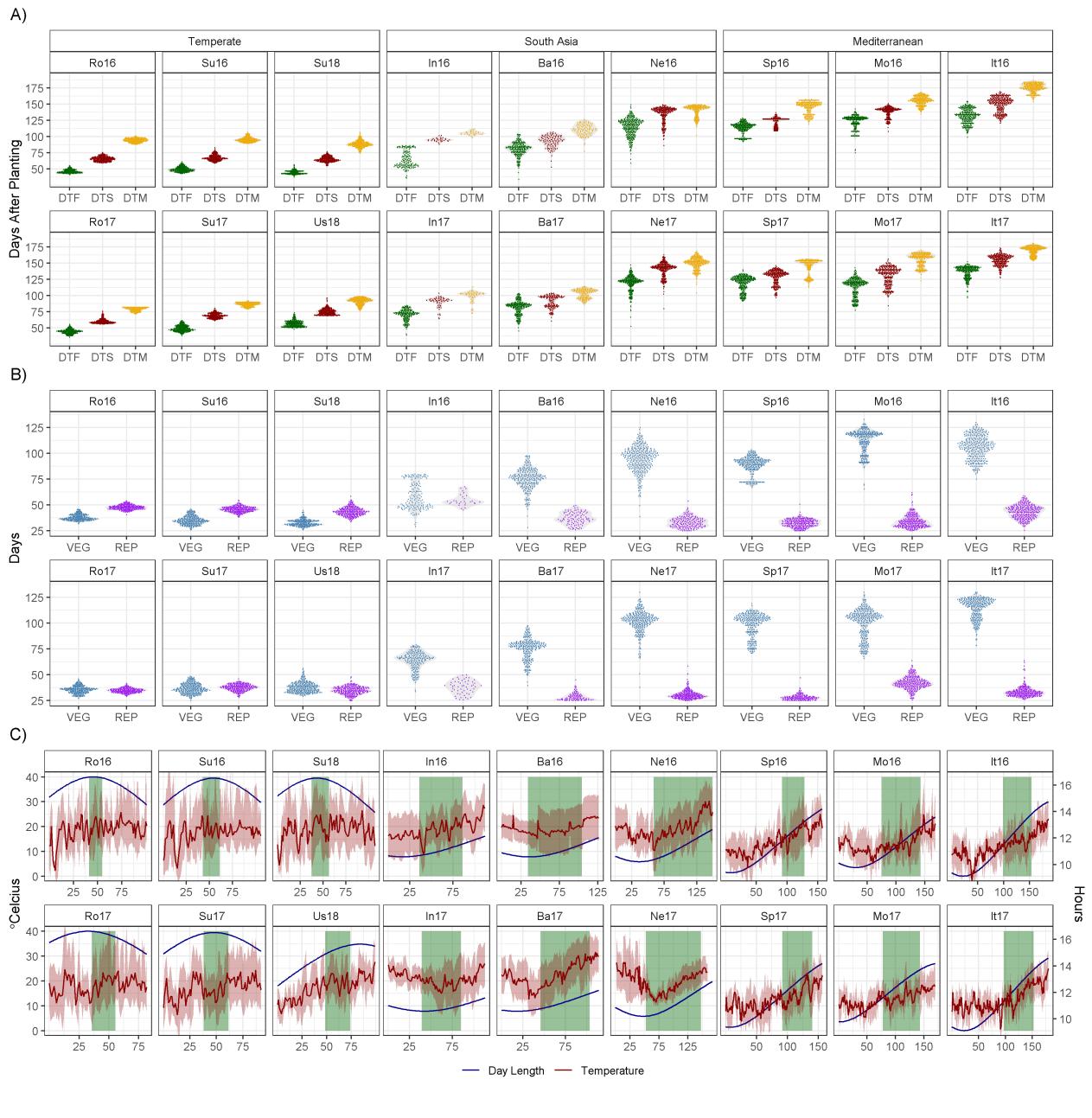
    scale_y_continuous(sec.axis = dup_axis(name = "\n"),
                       limits = c(30,190), breaks = c(50,75,100,125,150,175)) +
  theme_AGL + theme(axis.text.x = element_blank(), axis.ticks.x = element_blank()) +
  labs(x = NULL, y = NULL)
  mp
}
# Plot A) DTF, DTS and DTM
mp1.1 <- ggDistroDTF(xx %>% filter(MacroEnv == "Temperate"))
mp1.2 <- ggDistroDTF(xx %>% filter(MacroEnv == "South Asia"))
mp1.3 <- ggDistroDTF(xx %>% filter(MacroEnv == "Mediterranean"))
mp1 <- ggmatrix(list(mp1.1, mp1.2, mp1.3), nrow = 1, ncol = 3,
                title = "A)", ylab = "Days After Planting",
                xAxisLabels = macroEnvs) +
  theme_AGL +
  theme(plot.margin = unit(c(0,1,0,0), "cm"),
        plot.title = element_text(hjust = -0.04))
ggsave("Temp/Temp_F2_4.png", mp1, width = 12, height = 4)
# Prep data
xx <- dd %>% select(Entry, Name, Expt, ExptShort, Location, Year, VEG, REP) %>%
  left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  gather(Trait, Value, VEG, REP) %>%
  mutate(Trait = factor(Trait, levels = c("VEG", "REP")))
# Create plot function
ggDistroREP <- function(x) {
  mp <- ggplot(x, aes(x = Trait, y = Value)) +
    geom_violin(color = NA, fill = "grey", alpha = 0.3) +
    geom_quasirandom(size = 0.02, aes(color = Trait)) +
    facet_wrap(ExptShort~., scales = "free_x", dir = "v", ncol = 3, nrow = 2) +
    scale_color_manual(values = c("steelblue", "purple")) +
    scale_y_continuous(sec.axis = dup_axis(name = "\n"),
                      limits = c(25,135), breaks = c(25,50,75,100,125)) +
    theme_AGL + labs(x = NULL, y = "Days")
  mp
}
# Plot B) REP and VEG
mp2.1 <- ggDistroREP(xx %>% filter(MacroEnv == "Temperate"))
mp2.2 <- ggDistroREP(xx %>% filter(MacroEnv == "South Asia"))
mp2.3 <- ggDistroREP(xx %>% filter(MacroEnv == "Mediterranean"))
mp2 <- ggmatrix(list(mp2.1, mp2.2, mp2.3),
                nrow = 1, ncol = 3, title = "B)", ylab = "Days") +
  theme_AGL +
  theme(plot.margin = unit(c(0,1,0,0), "cm"),
        plot.title = element_text(hjust = -0.04))
ggsave("Temp/Temp_F2_5.png", mp2, width = 12, height = 4)
# Create plot function
ggEnvPlot <- function(x, nr = 2, nc = 3, mybreaks) {
  yy <- ff %>% filter(Expt %in% unique(x$Expt)) %>%
    select(ExptShort, Location, Year, min=DTF_min, max=DTF_max)
  mp <- ggplot(x) +
    geom_rect(data = yy, aes(xmin = min, xmax = max), fill = "darkgreen",
              ymin = 0, ymax = 40, alpha = 0.4) +
    geom_line(aes(x = DaysAfterPlanting, y = DayLength_rescaled, color = "Day Length")) +
    geom_line(aes(x = DaysAfterPlanting, y = Temp_mean, color = "Temperature")) +

```

```

geom_ribbon(aes(x = DaysAfterPlanting, ymin = Temp_min, ymax = Temp_max),
            fill = "darkred", alpha = 0.3) +
facet_wrap(ExptShort~, scales = "free_x", dir = "v", nrow = 2, ncol = 3) +
scale_x_continuous(breaks = mybreaks) +
scale_color_manual(name = NULL, values = c("darkblue", "darkred")) +
coord_cartesian(ylim=c(0, 40)) +
theme_AGL + theme(legend.position = "bottom") +
labs(y = NULL, x = NULL) +
guides(colour = guide_legend(order = 1), fill = guide_legend(order = 2))
mp
}
# Plot C) T and P
mp3.1 <- ggEnvPlot(ee %>% filter(MacroEnv == "Temperate"), mybreaks = c(25,50,75)) +
  labs(title = "C)", y = expression(paste(degree, "Celcius))) +
  theme(plot.margin = unit(c(0,0,0,0), "cm"),
        plot.title = element_text(hjust = -0.11))
mp3.2 <- ggEnvPlot(ee %>% filter(MacroEnv == "South Asia"), mybreaks = c(25,75,125)) +
  theme(plot.margin = unit(c(0,0,0,0), "cm"),
        axis.text.y = element_blank(),
        axis.ticks.y = element_blank())
mp3.3 <- ggEnvPlot(ee %>% filter(MacroEnv == "Mediterranean"), mybreaks = c(50,100,150)) +
  scale_y_continuous(sec.axis = sec_axis(~ (16.62 - 9.11) * . / (40 - 0) + 9.11,
                                         name = "Hours", breaks = c(10, 12, 14, 16))) +
  theme(plot.margin = unit(c(0,0,0,0), "cm"),
        axis.text.y.left = element_blank(),
        axis.ticks.y.left = element_blank())
mp3 <- ggarrange(mp3.1, mp3.2, mp3.3, nrow = 1, ncol = 3, align = "h",
                  legend = "bottom", common.legend = T, widths = c(1.1,1,1.1))
ggsave("Temp/Temp_F2_6.png", mp3, width = 12, height = 4)
# Append A), B) and C)
mp1 <- image_read("Temp/Temp_F2_4.png")
mp2 <- image_read("Temp/Temp_F2_5.png")
mp3 <- image_read("Temp/Temp_F2_6.png")
mp <- image_append(c(mp1, mp2, mp3), stack = T)
image_write(mp, "Figure2_DataOverview_Colored.png")

```



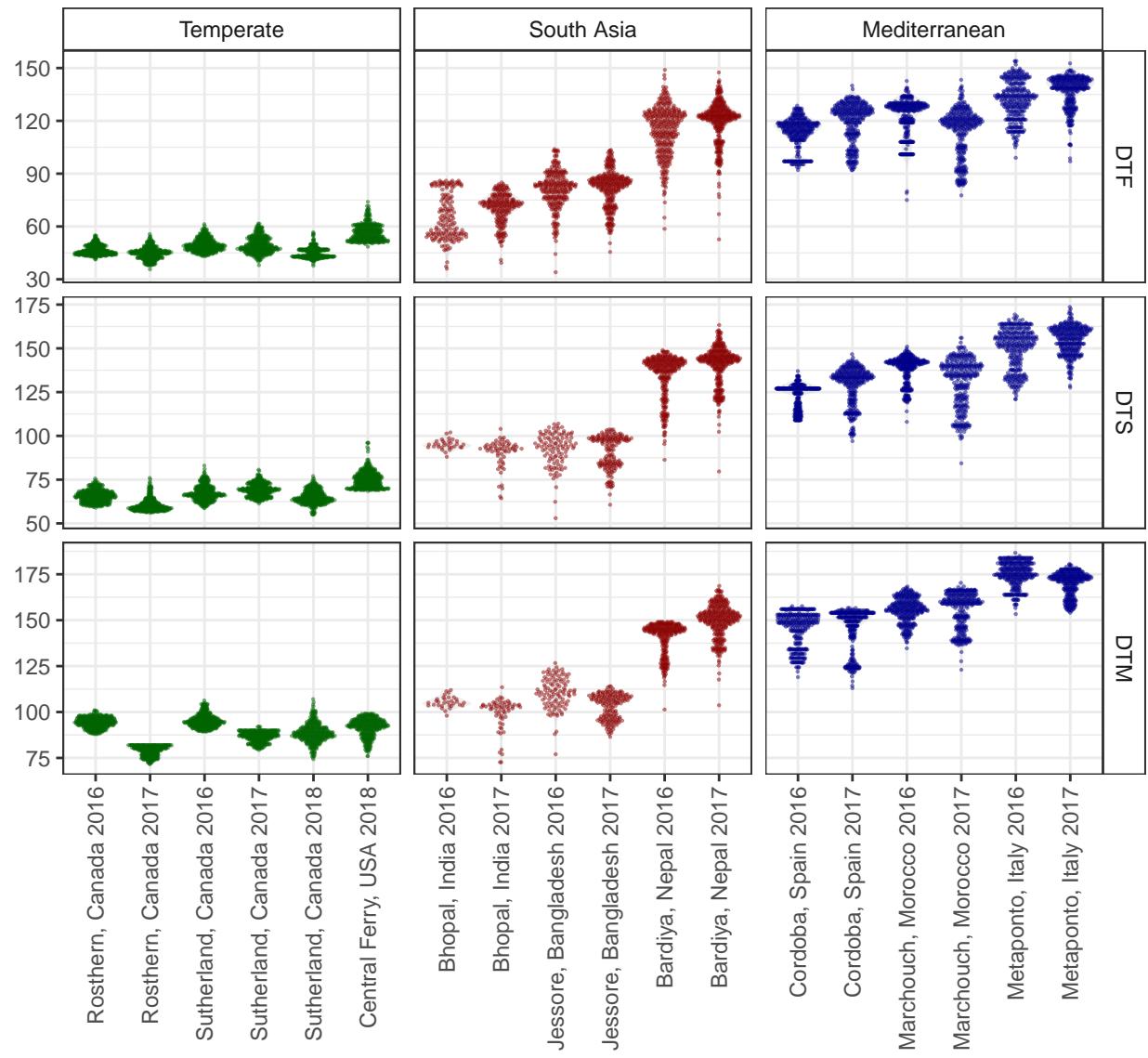
Additional Figure 2 DTF DTS DTM

```
# Prep data
xx <- dd %>% select(Entry, Expt, ExptShort, DTF, DTS, DTM) %>%
  left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  gather(Trait, Value, DTF, DTS, DTM) %>%
  mutate(Trait = factor(Trait, levels = c("DTF", "DTS", "DTM")) )
# Plot DTF, DTS, DTM
mp <- ggplot(xx, aes(x = Expt, y = Value)) +
```

```

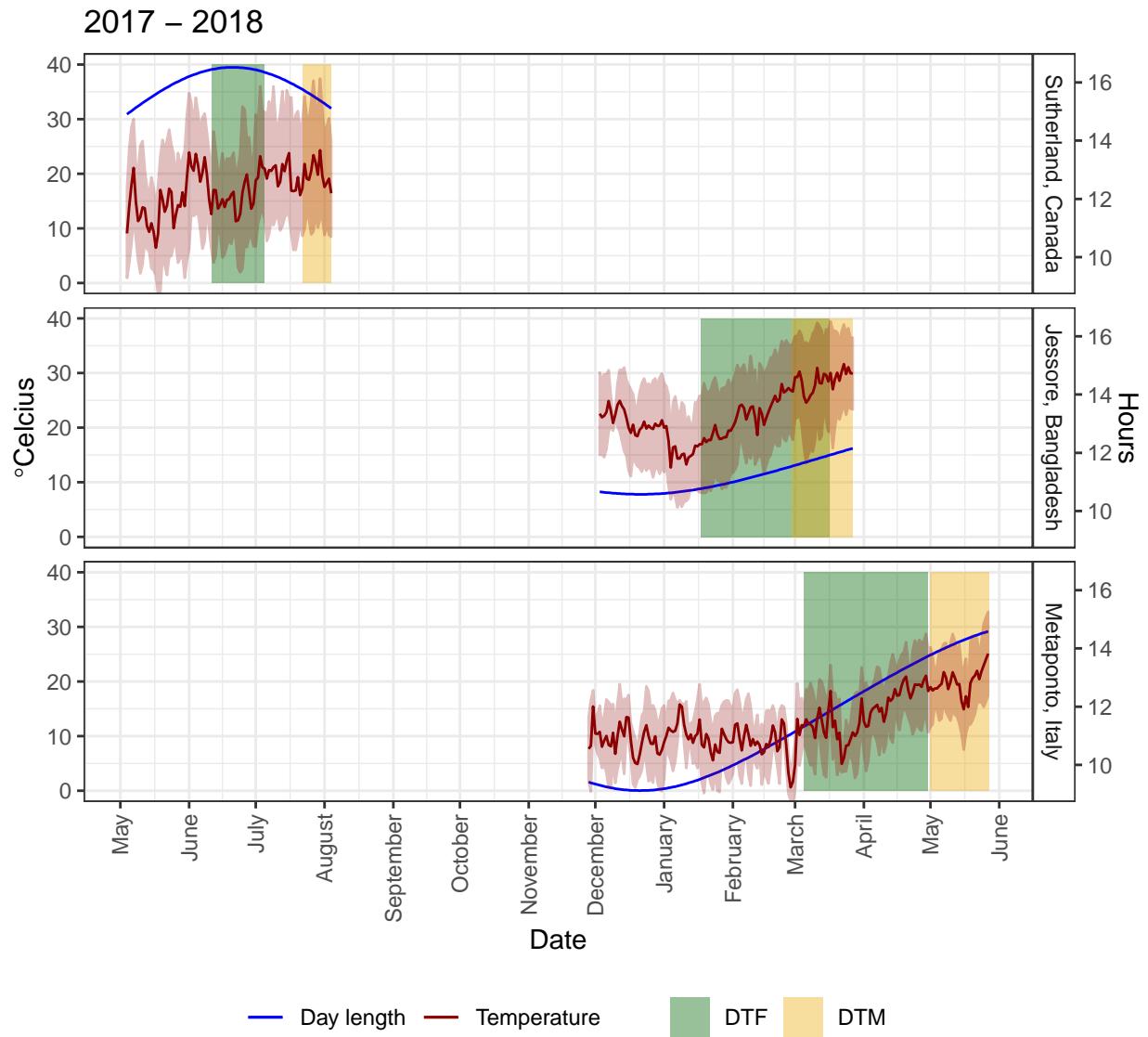
geom_violin(fill = "grey", alpha = 0.25, color = NA) +
geom_quasirandom(size = 0.1, alpha = 0.5, aes(color = MacroEnv)) +
facet_grid(Trait~MacroEnv, scales = "free") +
scale_color_manual(values = c("darkgreen", "darkred", "darkblue")) +
theme_AGL +
theme(legend.position = "none",
      axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
labs(x = NULL, y = NULL)
ggsave("Additional/AdditionalFigure02_DTFDTSDTM.png", mp, width = 8, height = 6)

```



Additional Figure 3 MacroEnv Phenology

```
# Prep data
xx <- ee %>% filter(ExptShort %in% c("Su17", "Ba17", "It17"))
yy <- ff %>% filter(Expt %in% unique(xx$Expt)) %>%
  mutate(DTF_min = Start + DTF_min, DTF_max = Start + DTF_max,
         DTM_min = Start + DTM_min, DTM_max = Start + DTM_max)
y1 <- select(yy, Expt, Location, Year, min = DTF_min, max = DTF_max) %>%
  mutate(Trait = "DTF")
y2 <- select(yy, Expt, Location, Year, min = DTM_min, max = DTM_max) %>%
  mutate(Trait = "DTM")
yy <- bind_rows(y1, y2)
# Plot DTF, DTM, T and P
mp <- ggplot(xx) +
  geom_rect(data = yy, aes(xmin = min, xmax = max, fill = Trait),
            ymin = 0, ymax = 40, alpha = 0.4) +
  geom_line(aes(x = Date, y = DayLength_rescaled, color = "Blue")) +
  geom_line(aes(x = Date, y = Temp_mean, color = "darkred") ) +
  geom_ribbon(aes(x = Date, ymin = Temp_min, ymax = Temp_max),
              fill = alpha("darkred", 0.25), color = alpha("darkred", 0.25)) +
  facet_grid(Location~, scales = "free_x", space = "free_x") +
  scale_color_manual(name = NULL, values = c("Blue", "darkred"),
                     labels = c("Day length", "Temperature") ) +
  scale_fill_manual(name = NULL, values = c("darkgreen", "darkgoldenrod2")) +
  coord_cartesian(ylim=c(0,40)) +
  theme_AGL +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  scale_x_date(breaks = "1 month", labels = date_format("%B")) +
  scale_y_continuous(sec.axis = sec_axis(~ (16.62 - 9.11) * . / (40 - 0) + 9.11,
                                         breaks = c(10, 12, 14, 16), name = "Hours")) +
  labs(title = "2017 - 2018", y = expression(paste(degree, "Celcius"), x = NULL)) +
  guides(colour = guide_legend(order = 1), fill = guide_legend(order = 2))
ggsave("Additional/AdditionalFigure03_MacroEnvPhenology.png", mp, width = 8, height = 6)
```



Additional Figures - Phenology pdf

```
# Create plotting function
gg_phenol <- function(x, xE, colnum) {
  ggplot(xE, aes(x = Trait, y = Value, group = Entry, color = ExptShort)) +
    geom_line(data = x, color = "grey", alpha = 0.5) +
    geom_line() + geom_point() +
    facet_grid(MacroEnv~ExptShort) +
    scale_color_manual(values = colors_Expt[colnum]) +
    theme_AGL + theme(legend.position = "none") +
    ylim(c(min(x$Value, na.rm = T), max(x$Value, na.rm = T))) +
    labs(x = NULL, y = NULL)
```

```

}

# Prep data
xx <- dd %>% select(Entry, Name, ExptShort, DTF, DTS, DTM) %>%
  left_join(select(ff, ExptShort, MacroEnv), by = "ExptShort") %>%
  gather(Trait, Value, DTF, DTS, DTM) %>%
  mutate(Trait = factor(Trait, levels = c("DTF", "DTS", "DTM")))
x1 <- xx %>% filter(MacroEnv == "Temperate")
x2 <- xx %>% filter(MacroEnv == "South Asia")
x3 <- xx %>% filter(MacroEnv == "Mediterranean")
# Create PDF
pdf("Additional/Plots_Phenology.pdf") #, width = 600, height = 600
par(mar=c(1.5, 2.5, 1.5, 0.5))
for(i in 1:324) {
  xE1 <- xx %>% filter(Entry == i, MacroEnv == "Temperate")
  xE2 <- xx %>% filter(Entry == i, MacroEnv == "South Asia")
  xE3 <- xx %>% filter(Entry == i, MacroEnv == "Mediterranean")
  mp1 <- gg_phenol(x1, xE1, 1:6)
  mp2 <- gg_phenol(x2, xE2, 7:12)
  mp3 <- gg_phenol(x3, xE3, 13:18)
  mp <- ggarrange(mp1, mp2, mp3, nrow = 3, ncol = 1) %>%
    annotate_figure(top = text_grob(paste("Entry", i, "|", unique(xE1$Name)), hjust=1.73))
  print(mp)
}
dev.off()

```

Supplemental Figure 2 Missing Data

```

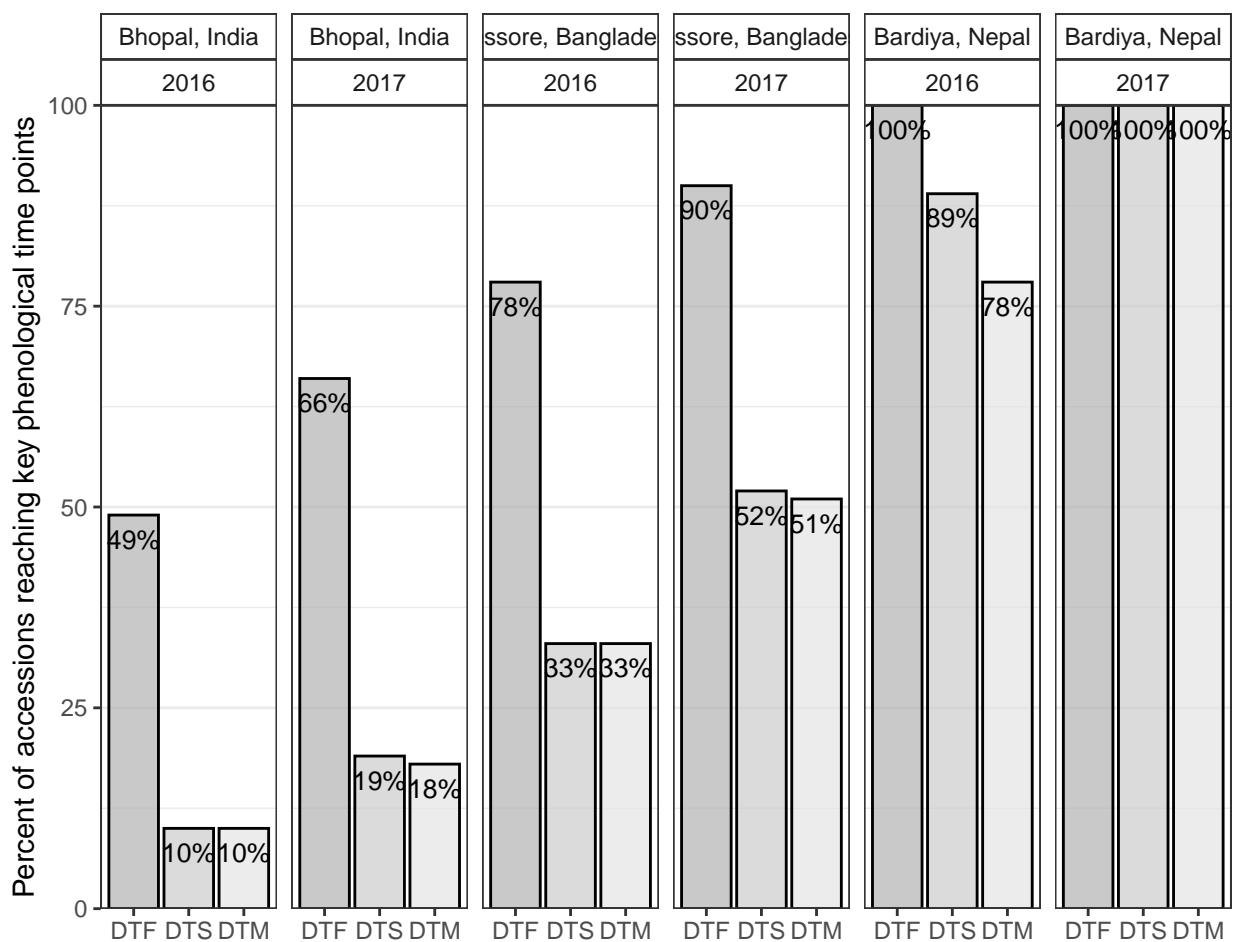
# Prep data
xx <- dd %>%
  filter(Location %in% c("Bhopal, India", "Jessore, Bangladesh", "Bardiya, Nepal")) %>%
  mutate(DTF = ifelse(is.na(DTF), 0, 1),
         DTS = ifelse(is.na(DTS), 0, 1),
         DTM = ifelse(is.na(DTM), 0, 1)) %>%
  group_by(Expt, Location, Year) %>%
  summarise_at(vars(DTF, DTS, DTM), funs(sum), na.rm = T) %>%
  ungroup() %>%
  gather(Trait, Flowered, DTF, DTS, DTM) %>%
  mutate(Total = ifelse(Expt == "Bardiya, Nepal 2016", 323, 324),
        # One accession was not planted in Bardiya, Nepal 2016
        DidNotFlower = Total - Flowered,
        Percent = round(100 * Flowered / Total),
        Label = paste0(Percent, "%"),
        Trait = factor(Trait, levels = c("DTF", "DTS", "DTM")))
# Plot Percent Flowered
mp <- ggplot(xx, aes(x = Trait, y = Percent, fill = Trait)) +
  geom_bar(stat = "identity", color = "black", alpha = 0.7) +
  geom_text(aes(label = Label), nudge_y = -3, size = 3.5) +
  facet_grid(.~Location + Year) +
  scale_fill_manual(values = c("grey70", "grey80", "grey90"))

```

```

scale_y_continuous(limits = c(0,100), expand = c(0,0)) +
theme_AGL +
theme(legend.position = "none",
      panel.grid.major.x = element_blank() ) +
labs(x = NULL, y = "Percent of accessions reaching key phenological time points")
ggsave("SupplementalFigure2_PercentFlowered.png", width = 10, height = 5)

```



Supplemental Figure 3 Correlation Plots

```

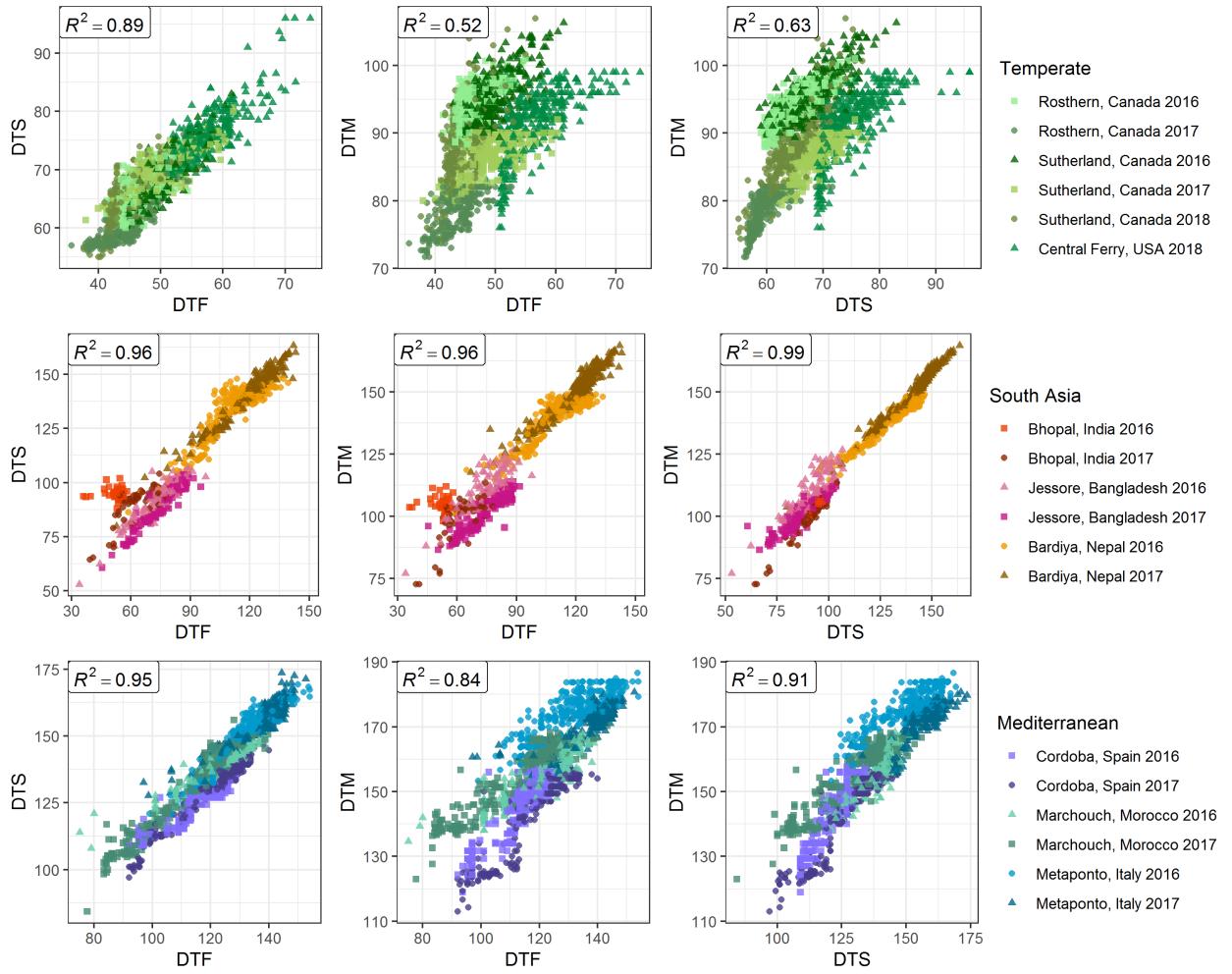
# Prep data
xx <- dd %>% left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  select(Entry, Expt, MacroEnv, DTF, DTS, DTM)
# Create plotting function
ggCorPlot <- function(x, legend.title, colNums) {
  # Plot A)
  r2 <- round(cor(x$DTF, x$DTS, use ="complete", method = "pearson"), 2)
  tp1 <- ggplot(x) + theme_AGL +

```

```

geom_point(aes(x = DTF, y = DTS, color = Expt, shape = Expt), alpha = 0.8) +
  geom_label(x = -Inf, y = Inf, hjust = 0, vjust = 1, parse = T,
             label = paste("italic(R)^2 == ", r2)) +
  scale_color_manual(name = legend.title, values = colors_Expt[colNums]) +
  scale_shape_manual(name = legend.title, values = c(15,16,17,15,16,17))
# Plot B)
r2 <- round(cor(x$DTF, x$DTM, use ="complete.obs", method = "pearson"), 2)
tp2 <- ggplot(x) + theme_AGL +
  geom_point(aes(x = DTF, y = DTM, color = Expt, shape = Expt), alpha = 0.8) +
  geom_label(x = -Inf, y = Inf, hjust = 0, vjust = 1, parse = T,
             label = paste("italic(R)^2 == ", r2)) +
  scale_color_manual(name = legend.title, values = colors_Expt[colNums]) +
  scale_shape_manual(name = legend.title, values = c(15,16,17,15,16,17))
# Plot C)
r2 <- round(cor(x$DTS, x$DTM, use = "complete", method = "pearson"), 2)
tp3 <- ggplot(x) + theme_AGL +
  geom_point(aes(x = DTS, y = DTM, color = Expt, shape = Expt), alpha = 0.8) +
  geom_label(x = -Inf, y = Inf, hjust = 0, vjust = 1, parse = T,
             label = paste("italic(R)^2 == ", r2)) +
  scale_color_manual(name = legend.title, values = colors_Expt[colNums]) +
  scale_shape_manual(name = legend.title, values = c(15,16,17,15,16,17))
# Append A), B) and C)
mp <- ggarrange(tp1, tp2, tp3, nrow = 1, ncol = 3,
                 common.legend = T, legend = "right")
mp
}
# Plot Correlations
mp1 <- ggCorPlot(xx%>%filter(MacroEnv == "Temperate"),      "Temperate",      1:6 )
mp2 <- ggCorPlot(xx%>%filter(MacroEnv == "South Asia"),      "South Asia",      7:12)
mp3 <- ggCorPlot(xx%>%filter(MacroEnv == "Mediterranean"), "Mediterranean", 13:18)
mp <- ggarrange(mp1, mp2, mp3, nrow = 3, ncol = 1, common.legend = T, legend = "right")
ggsave("SupplementalFigure3_Correlations.png", mp, width = 10, height = 8)

```



Additional Figures - Correlations

```
# Prep data
xx <- dd %>% left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  mutate(DTE = ifelse(Location == "Cordoba, Spain", NA, DTE))
x1 <- xx %>% filter(MacroEnv == "Temperate")
x2 <- xx %>% filter(MacroEnv == "South Asia")
x3 <- xx %>% filter(MacroEnv == "Mediterranean")
# Create plotting functions
my_lower <- function(data, mapping, cols = colors_Expt, ...) {
  ggplot(data = data, mapping = mapping) +
    geom_point(alpha=0.5, size=0.3, aes(color = Expt)) +
    theme_bw() + scale_color_manual(values = cols)
}
my_middle <- function(data, mapping, cols = colors_Expt, ...) {
  ggplot(data = data, mapping = mapping) +
    geom_density(alpha = 0.5) + theme_bw() +
    scale_color_manual(values = cols)
```

```

    scale_color_manual(name = NULL, values = cols) +
    scale_fill_manual(name = NULL, values = cols) +
    guides(color = F, fill = guide_legend(nrow = 3, byrow = T))
}

# See: https://github.com/ggobi/ggally/issues/139
my_upper <- function(data, mapping, color = I("black"), sizeRange = c(1,5), ...) {
  # Prep data
  x <- eval_data_col(data, mapping$x)
  y <- eval_data_col(data, mapping$y)
  #
  ct <- cor.test(x, y, method = "pearson")
  r <- unname(ct$estimate^2)
  rt <- format(r, digits=2)[1]
  cex <- max(sizeRange)
  tt <- as.character(rt)
  # plot the cor value
  p <- ggally_text(label = tt, mapping = aes(), color = color,
                    xP = 0.5, yP = 0.5, size = 6, ... ) + theme_bw()
  # Create color palette
  corColors <- RColorBrewer::brewer.pal(n = 10, name = "RdBu")[2:9]
  if (r <= -0.9) { corCol <- alpha(corColors[1], 0.5)
  } else if (r >= -0.9 & r <= -0.6) { corCol <- alpha(corColors[2], 0.5)
  } else if (r >= -0.6 & r <= -0.3) { corCol <- alpha(corColors[3], 0.5)
  } else if (r >= -0.3 & r <= 0) { corCol <- alpha(corColors[4], 0.5)
  } else if (r >= 0 & r <= 0.3) { corCol <- alpha(corColors[5], 0.5)
  } else if (r >= 0.3 & r <= 0.6) { corCol <- alpha(corColors[6], 0.5)
  } else if (r >= 0.6 & r <= 0.9) { corCol <- alpha(corColors[7], 0.5)
  } else { corCol <- alpha(corColors[8], 0.5) }
  # Plot
  p <- p +
    theme(panel.background = element_rect(fill = corCol),
          panel.grid.major = element_blank(),
          panel.grid.minor = element_blank(),
          axis.text = element_text(size = 5))
  p
}

# Plot Correlations for each Expt
for(i in names_Expt) {
  mp <- ggpairs(xx %>% filter(Expt == i),
                columns = c("DTE", "DTF", "DTS", "DTM", "REP"),
                upper = list(continuous = my_upper),
                diag = list(continuous = my_middle),
                lower = list(continuous = wrap(my_lower, cols = "black")),
                title = i) +
    theme(strip.background = element_rect(fill = "White"))
  ggsave(paste0("Additional/Corr/Corr_Expt", i, ".png"), mp, width = 6, height = 6)
}

# Plot A) Temperate
mp1 <- ggpairs(x1, columns = c("DTE", "DTF", "DTS", "DTM", "REP"),
                aes(color = Expt, fill = Expt),
                upper=list(continuous = my_upper),
                diag =list(continuous = wrap(my_middle, cols = colors_Expt[1:6])),
                lower=list(continuous = wrap(my_lower, cols = colors_Expt[1:6])),

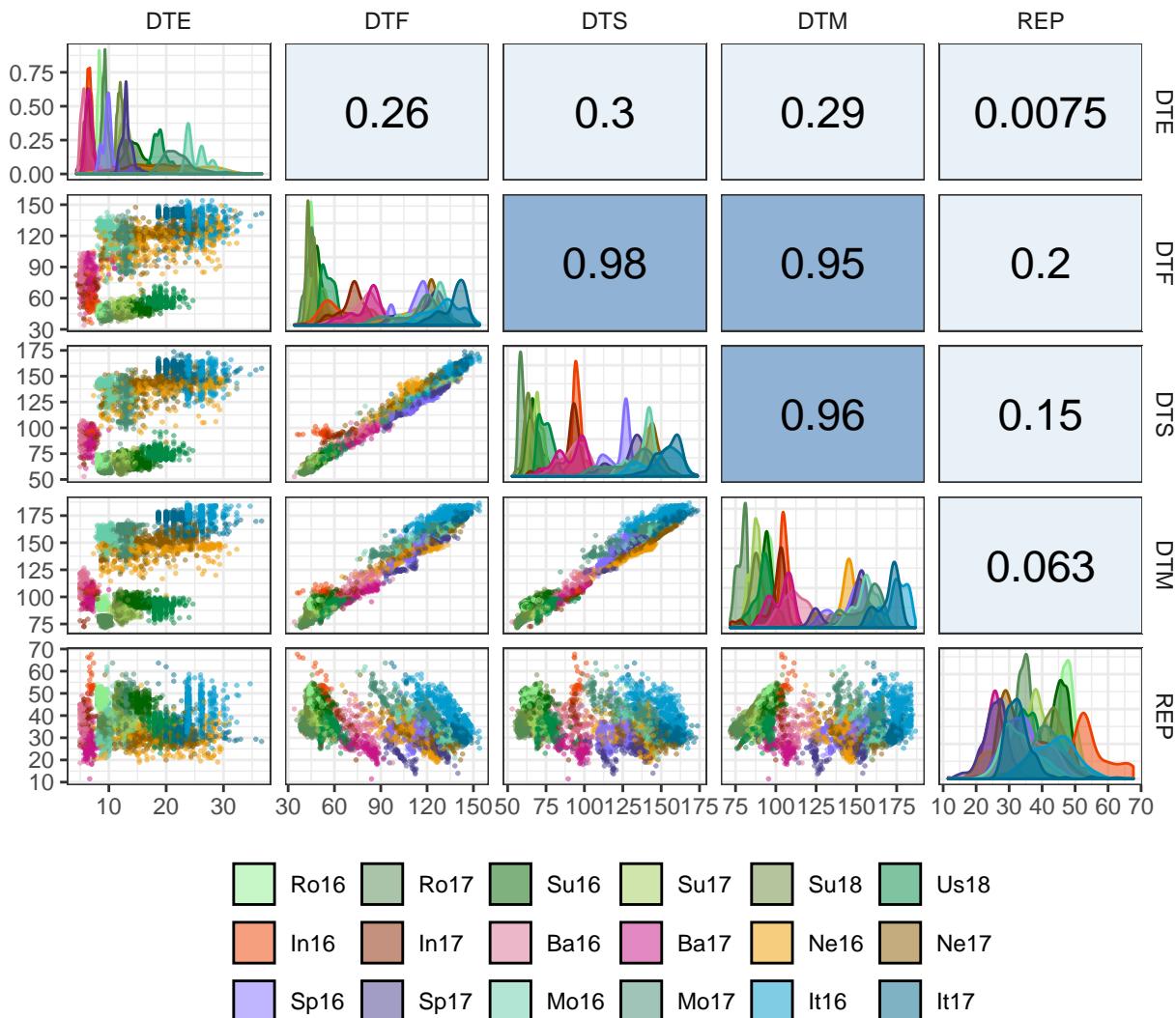
```

```

    title = "A) Temperate",
    legend = c(2,2)) +
  theme(strip.background = element_rect(fill = "White"),
        legend.position = "bottom")
ggsave("Additional/Corr/Corr_Temperate.png", mp1, width = 6, height = 6)
# Plot B) South Asia
mp2 <- ggpairs(x2, columns = c("DTE", "DTF", "DTS", "DTM", "REP"),
                aes(color = Expt, fill = Expt),
                upper = list(continuous = my_upper),
                diag = list(continuous = wrap(my_middle, cols = colors_Expt[7:12])),
                lower = list(continuous = wrap(my_lower, cols = colors_Expt[7:12])),
                title = "B) South Asia",
                legend = c(2,2)) +
  theme(strip.background = element_rect(fill = "White"),
        legend.position = "bottom")
ggsave("Additional/Corr/Corr_SouthAsia.png", mp2, width = 6, height = 6)
# Plot C) Mediterranean
mp3 <- ggpairs(x3, columns = c("DTE", "DTF", "DTS", "DTM", "REP"),
                aes(color = Expt, fill = Expt),
                upper = list(continuous = my_upper),
                diag = list(continuous = wrap(my_middle, cols = colors_Expt[13:18])),
                lower = list(continuous = wrap(my_lower, cols = colors_Expt[13:18])),
                title = "C) Mediterranean",
                legend = c(2,2)) +
  theme(strip.background = element_rect(fill = "White"),
        legend.position = "bottom")
ggsave("Additional/Corr/Corr_Mediterranean.png", mp3, width = 6, height = 6)
# Plot All
mp4 <- ggpairs(xx, columns = c("DTE", "DTF", "DTS", "DTM", "REP"),
                aes(color = ExptShort, fill = ExptShort),
                upper = list(continuous = my_upper),
                diag = list(continuous = my_middle),
                lower = list(continuous = my_lower),
                title = "D) ALL",
                legend = c(2,2)) +
  theme(strip.background = element_rect(fill = "White"),
        legend.position = "bottom")
ggsave("Additional/Corr/Corr_All.png", mp4, width = 6, height = 6)

```

D) ALL



PhotoThermal Plane

```
# Prep data
xx <- rr %>% filter(!is.na(RDTF)) %>%
  left_join(select(ff, Expt, T_mean, P_mean), by = "Expt")
# Create plotting function
gg_PTplane <- function(x, i, colored = T) {
  x1 <- x %>% filter(Entry == i) %>%
    left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
    arrange(MacroEnv) %>%
    mutate(myPal = as.character(plyr::mapvalues(MacroEnv,
      c("Temperate", "South Asia", "Mediterranean"), c("darkgreen", "darkred", "darkblue"))))
```

```

if(colored == F) {x1 <- x1 %>% mutate(myPal = "Black")}
x <- x1$T_mean
y <- x1$P_mean
z <- x1$RDTF
fit <- lm(z ~ x + y)
# Create PhotoThermal plane
fitpoints <- predict(fit)
grid.lines <- 12
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid(x = x.pred, y = y.pred)
z.pred <- matrix(predict(fit, newdata = xy),
                  nrow = grid.lines, ncol = grid.lines)
# Plot with regression plane
par(mar=c(1.5, 2.5, 1.5, 0.5))
scatter3D(x, y, z, pch = 18, cex = 2, zlim = c(0.005,0.03),
           col = x1$myPal, colvar = as.numeric(x1$MacroEnv), colkey = F,
           theta = 40, phi = 25, ticktype = "detailed", cex.lab = 1, cex.axis = 0.5,
           xlab = "Temperature", ylab = "Photoperiod", zlab = "1 / DTF", main = unique(x1$name),
           surf = list(x = x.pred, y = y.pred, z = z.pred, col = "black",
                       facets = NA, fit = fitpoints) )
}

# Plot each Entry
for (i in 1:324) {
  png(paste0("Additional/3D/3D_Entry_", stringr::str_pad(i, 3, pad = "0"), ".png"),
      width = 600, height = 600, res = 150)
  gg_PTplane(xx, i)
  dev.off()
}
# Create PDF
mp <- list()
pdf("Additional/Plots_3D.pdf") #, width = 600, height = 600
par(mar=c(1.5, 2.5, 1.5, 0.5))
for (i in 1:324) {
  gg_PTplane(xx, i)
}
dev.off()
# Create animation
lf <- list.files("Additional/3D")
mp <- image_read(paste0("Additional/3D/", lf))
animation <- image_animate(mp, fps = 2)
image_write(animation, "Additional/Animation_3D.gif")
# Plot ILL 5888, ILL 4400, Laird
xx <- xx %>% mutate(Name = gsub(" AGL", "", Name))
for (i in c(76,94,128)) {
  png(paste0("Temp/3D_Entry_", stringr::str_pad(i, 3, pad = "0"), ".png"),
      width = 600, height = 600, res = 150)
  gg_PTplane(xx, i)
  dev.off()
  #
  png(paste0("Temp/3D_Entry_", stringr::str_pad(i, 3, pad = "0"), "_BW.png"),
      width = 600, height = 600, res = 150)
  gg_PTplane(xx, i, colored = F)
}

```

```

    dev.off()
}

```

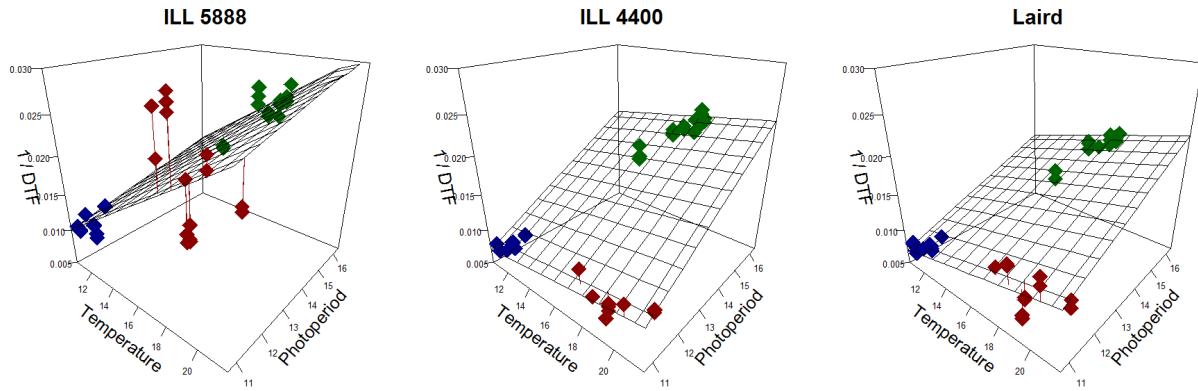


Figure 3 Regressions

```

gg_Reg <- function(filename, colored = T) {
  # Prep data
  myfills <- alpha(c("darkgreen", "darkred", "darkblue"), 0.5)
  myalpha = 0
  yy <- c("ILL 5888 AGL", "ILL 4400 AGL", "Laird AGL")
  xx <- rr %>% filter(Name %in% yy, !is.na(DTF)) %>%
    left_join(select(ff, Expt, MacroEnv, T_mean, P_mean), by = "Expt") %>%
    mutate(Name = gsub(" AGL", "", Name),
          Name = factor(Name, levels = gsub(" AGL", "", yy)),
          myfill = MacroEnv)
  if(colored == F) {
    myfills <- rep(alpha("White", alpha = 0), 3)
    myalpha <- 1
  }
  x1 <- xx %>% filter(MacroEnv != "South Asia")
  x2 <- xx %>% filter(MacroEnv != "Temperate")
  x3 <- xx %>% filter(MacroEnv != "Mediterranean")
  # Plot A) 1/f = a + bT
  mp1 <- ggplot(xx, aes(x = T_mean, y = RDTF)) +
    geom_point(aes(shape = MacroEnv, fill = MacroEnv, color = "Lens")) +
    geom_smooth(data = x1, method = "lm", se = F, color = "black", lty = 3) +
    geom_smooth(data = x2, method = "lm", se = F, color = "black") +
    scale_y_continuous(trans = "reverse", breaks = c(0.01, 0.02, 0.03),
                       sec.axis = dup_axis(~ 1./, name = "DTF",
                                          breaks = c(35, 50, 100, 150))) +
    scale_x_continuous(breaks = c(11, 13, 15, 17, 19, 21)) +
    scale_shape_manual(values = c(24, 22, 21)) +
    scale_fill_manual(values = myfills) +
    scale_color_manual(values = alpha("Black", myalpha))
}

```

```

theme_AGL +
theme(plot.margin = unit(c(-0.5,0,0,0.15), "cm"),
      plot.title = element_text(hjust = -0.115, vjust = -6) ) +
facet_grid(.~Name) + guides(color = F) +
labs(x = expression(paste("Temperature (", degree, "C)", sep = "")),
     y = "1 / DTF", title = "A"))
# Plot B) 1/f = a + cP
mp2 <- ggplot(xx, aes(x = P_mean, y = RDTF)) +
  geom_point(aes(shape = MacroEnv, fill = MacroEnv, color = "Lens")) +
  geom_smooth(data = x1, method = "lm", se = F, color = "black", lty = 3) +
  geom_smooth(data = x3, method = "lm", se = F, color = "black") +
  scale_y_continuous(trans = "reverse", breaks = c(0.01,0.02,0.03),
                     sec.axis = dup_axis(~ 1/, name="DTF", breaks = c(35,50,100,150))) +
  scale_x_continuous(breaks = c(11,12,13,14,15,16)) +
  scale_shape_manual(values = c(24,22,21)) +
  scale_fill_manual(values = myfills) +
  scale_color_manual(values = alpha("Black", myalpha)) +
  theme_AGL + guides(color = F) +
  theme(plot.margin = unit(c(-0.5,0,0,0.15), "cm"),
        plot.title = element_text(hjust = -0.115, vjust = -6) ) +
  facet_grid(.~Name) +
  labs(x = "Photoperiod (hours)", y = "1 / DTF", title = "B"))
# Append A) and B)
mp <- ggarrange(mp1, mp2, ncol = 1, nrow = 2, label.y = "Test",
                common.legend = T, legend = "bottom")
ggsave(filename, mp, width = 6, height = 4)
}

# Create A) and B)
gg_Reg("Temp/Temp_F3_1.png", F)
# Append C)s
im1 <- image_read("Temp/3D_Entry_094_BW.png")
im2 <- image_read("Temp/3D_Entry_076_BW.png")
im3 <- image_read("Temp/3D_Entry_128_BW.png")
im <- image_append(c(im1, im2, im3))
image_write(im, "Temp/Temp_F3_2.png")
# Append A), B) and C)
im1 <- image_read("Temp/Temp_F3_1.png")
im2 <- image_read("Temp/Temp_F3_2.png") %>%
  image_annotate("C", size = 50)
im <- image_append(c(im1, im2), stack = T)
image_write(im, "Figure3_2D3D.png")
#

```

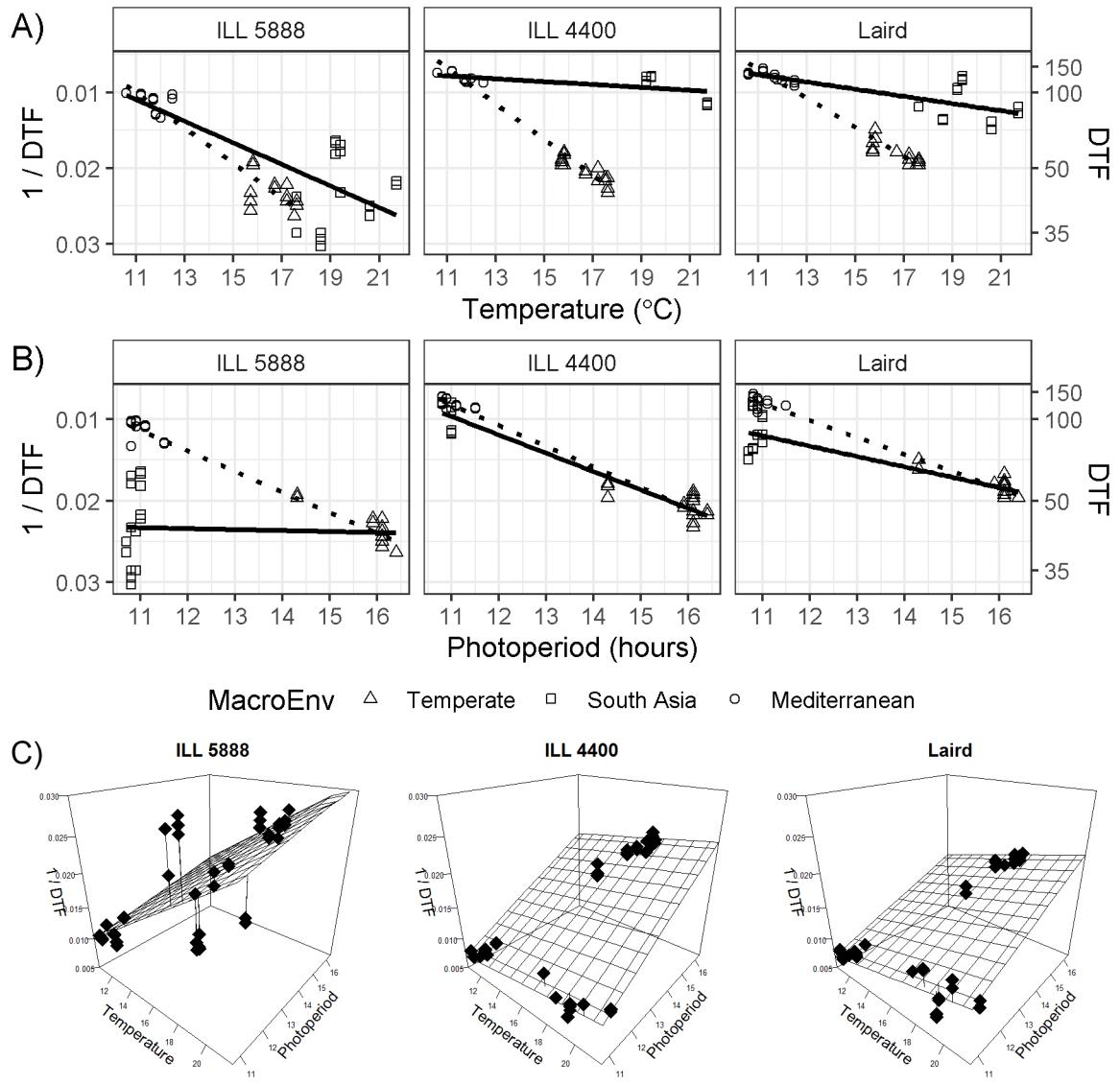


Figure 3 Colored

```
# Create A) and B)
gg_Reg("Temp/Temp_F3_3.png", T)
# Append C)s
im1 <- image_read("Temp/3D_Entry_094.png")
im2 <- image_read("Temp/3D_Entry_076.png")
im3 <- image_read("Temp/3D_Entry_128.png")
im <- image_append(c(im1, im2, im3))
image_write(im, "Temp/Temp_F3_4.png")
# Append A), B) and C)
im1 <- image_read("Temp/Temp_F3_3.png")
im2 <- image_read("Temp/Temp_F3_4.png") %>%
  image_annotate("C)", size = 50)
im <- image_append(c(im1, im2), stack = T)
image_write(im, "Figure3_2D3D_colored.png")
```

mp

Metaponto, Italy 2017

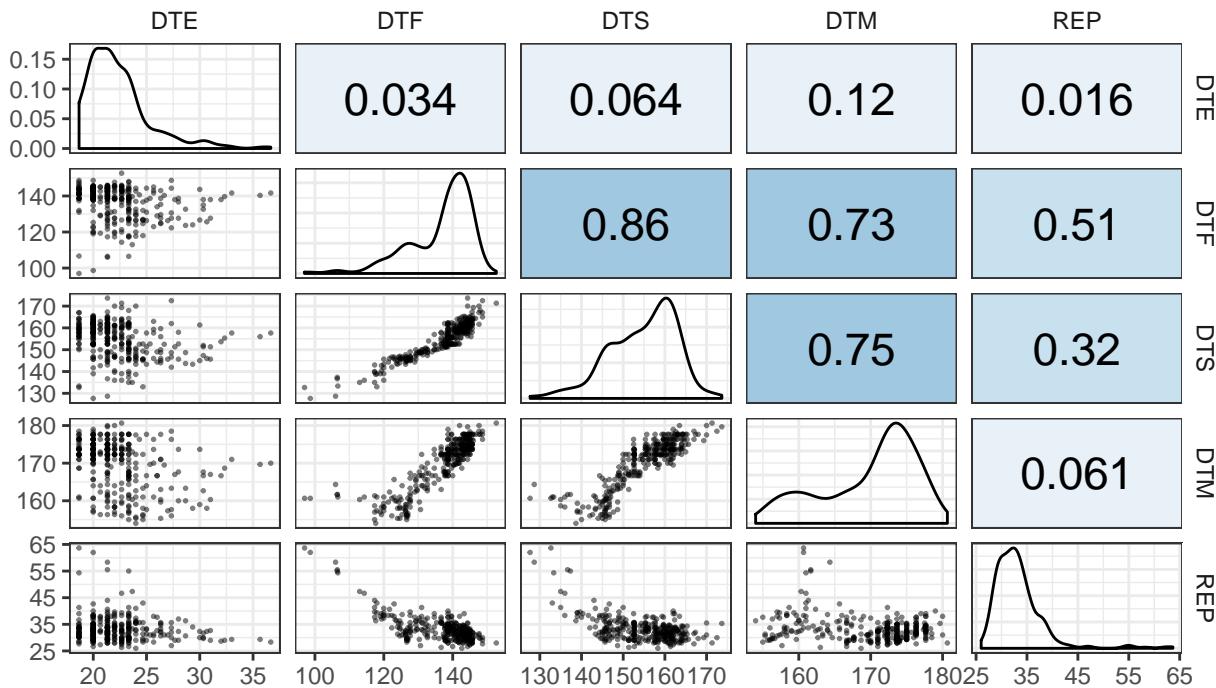


Figure 4 Modeling DTF

```
# Create functions
# Plot Observed vs Predicted
gg_model_1 <- function(x, title = NULL, xmin = min(x$DTF)-2, xmax = max(x$DTF)+2,
                       ymin = min(x$Predicted_DTF)-2, ymax = max(x$Predicted_DTF)+2, type = 1) {
  # Prep data
  if(type == 1) {
    myx <- "DTF"
    myy <- "Predicted_DTF"
    x <- x %>% filter(!is.na(DTF))
  }
  if(type == 2) {
    myx <- "RDTF"
    myy <- "Predicted_RDTF"
    x <- x %>% filter(!is.na(RDTF))
  }
  myPal <- colors_Expt[names_Expt %in% unique(x$Expt)]
  r2 <- round(modelR2(x = x[,myx], y = x[,myy]), 3)
  # Plot
  mp <- ggplot(x) +
```

```

geom_point(aes(x = get(myX), y = get(myY), color = Expt)) +
  geom_abline() +
  geom_label(x = xmin, y = ymax, hjust = 0, vjust = 1, parse = T,
             label = paste("italic(R)^2 == ", r2)) +
  scale_x_continuous(limits = c(xmin, ymax), expand = c(0, 0)) +
  scale_y_continuous(limits = c(ymin, ymax), expand = c(0, 0)) +
  scale_color_manual(name = NULL, values = myPal) +
  theme_AGL + labs(y = "Predicted", x = "Observed")
  if(!is.null(title)) { mp <- mp + labs(title = title) }
  mp
}

# Facets by Expt
gg_model_2 <- function(x, myX = "DTF", myY = "Predicted_DTF", title = NULL,
                       x1 = 30, x2 = 30, y1 = 145, y2 = 120) {
  # Prep data
  x <- x %>% filter(!is.na(get(myX)))
  xf <- x %>% group_by(Expt) %>%
    summarise(Mean = mean(DTF)) %>% ungroup() %>%
    mutate(r2 = NA, RMSE = NA)
  for(i in 1:nrow(xf)) {
    xi <- x %>% filter(Expt == xf$Expt[i])
    xf[i, "r2"] <- round(modelR2(x = xi[, myX], y = xi[, myY]), 2)
    xf[i, "RMSE"] <- round(modelRMSE(x = xi[, myX], y = xi[, myY]), 1)
  }
  # Plot
  mp <- ggplot(x, aes(x = get(myX), y = get(myY))) +
    geom_point(size = 0.75, pch = 21, fill = "grey") + geom_abline() +
    geom_text(x = x1, y = y1, color = "black", hjust = 0, vjust = 0,
              aes(label = paste("RMSE = ", RMSE, sep = "")), size = 3, data = xf) +
    geom_text(x = x2, y = y2, color = "black", hjust = 0, vjust = 0,
              aes(label = paste("italic(R)^2 == ", r2)), size = 3, data = xf, parse = T) +
    facet_wrap(Expt ~ ., ncol = 6, labeller = label_wrap_gen(width = 17)) +
    scale_x_continuous(limits = c(min(x[, myX]), max(x[, myX]))) +
    scale_y_continuous(limits = c(min(x[, myX]), max(x[, myX]))) +
    theme_AGL + labs(y = "Predicted", x = "Observed")
  if(!is.null(title)) { mp <- mp + labs(title = title) }
  mp
}

# R^2 function
modelR2 <- function(x, y) {
  1 - (sum((x - y)^2, na.rm = T) / sum((x - mean(x, na.rm = T))^2, na.rm = T))
}
modelRMSE <- function(x, y) {
  sqrt(sum((x - y)^2) / length(x))
}

```

$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}} = 1 - \frac{\sum(x-y)^2}{\sum(x-\bar{x})}$$

$$RMSE = \sqrt{\frac{\sum(y-x)^2}{n}}$$

Train the model using all siteyears

```

#####
# 1/f = a + bT + cP (All) #
#####

# Prep data
xx <- rr %>% filter(!is.na(RDTF)) %>%
  left_join(select(ff, Expt, T_mean, P_mean), by = "Expt") %>%
  select(Plot, Entry, Name, Rep, Expt, ExptShort, T_mean, P_mean, RDTF, DTF)
mr <- NULL; md <- NULL
mc <- select(ldp, Entry, Name) %>%
  mutate(a = NA, b = NA, c = NA, RR = NA, Environments = NA)
# Model
for(i in 1:324) {
  # Prep data
  xri <- xx %>% filter(Entry == i)
  xdi <- xri %>% group_by(Entry, Name, Expt) %>%
    summarise_at(vars(DTF, RDTF, T_mean, P_mean), funs(mean), na.rm = T)
  # Train Model
  mi <- lm(RDTF ~ T_mean + P_mean, data = xri)
  # Predict DTF
  xri <- xri %>% mutate(Predicted_RDTF = predict(mi),
                           Predicted_DTF = 1 / predict(mi))
  xdi <- xdi %>% mutate(Predicted_RDTF = predict(mi, newdata = xdi),
                           Predicted_DTF = 1 / predict(mi, newdata = xdi))
  # Save to table
  mr <- bind_rows(mr, xri)
  md <- bind_rows(md, xdi)
  # Save coefficients
  mc[i,c(3:5)] <- mi$coefficients
  # Calculate rr and # of environments used
  mc[i,6] <- 1 - sum((xri$DTF - xri$Predicted_DTF)^2, na.rm = T) /
    sum((xri$Predicted_DTF - mean(xri$DTF, na.rm = T))^2, na.rm = T)
  mc[i,7] <- length(unique(xri$Expt))
}
mr <- mr %>% mutate(Expt = factor(Expt, levels = names_Expt))
md <- md %>% mutate(Expt = factor(Expt, levels = names_Expt))
# Save Results
write.csv(mr, "data/model_18.csv", row.names = F)
write.csv(md, "data/model_18_d.csv", row.names = F)
write.csv(mc, "data/model_18_coefs.csv", row.names = F)
#
# Plot Each Entry
for(i in 1:324) {
  mp1 <- gg_model_1(mr %>% filter(Entry == i), paste("Entry", i, "| DTF"),
                     xmin = min(x$DTF), xmax = max(x$DTF),
                     ymin = min(x$Predicted_DTF), ymax = max(x$Predicted_DTF))
  mp2 <- gg_model_1(mr %>% filter(Entry == i), paste("Entry", i, "| RDTF"),
                     xmin = min(x$RDTF), xmax = max(x$RDTF),
                     ymin = min(x$Predicted_RDTF), ymax = max(x$Predicted_RDTF), type = 2)
  mp <- ggarrange(mp1, mp2, ncol = 2, common.legend = T, legend = "right")
  fname <- paste0("Additional/Model/Model_Entry_", stringr::str_pad(i, 3, pad = "0"), ".png")
  ggsave(fname, mp, width = 12, height = 5)
}
# Create pdf

```

```

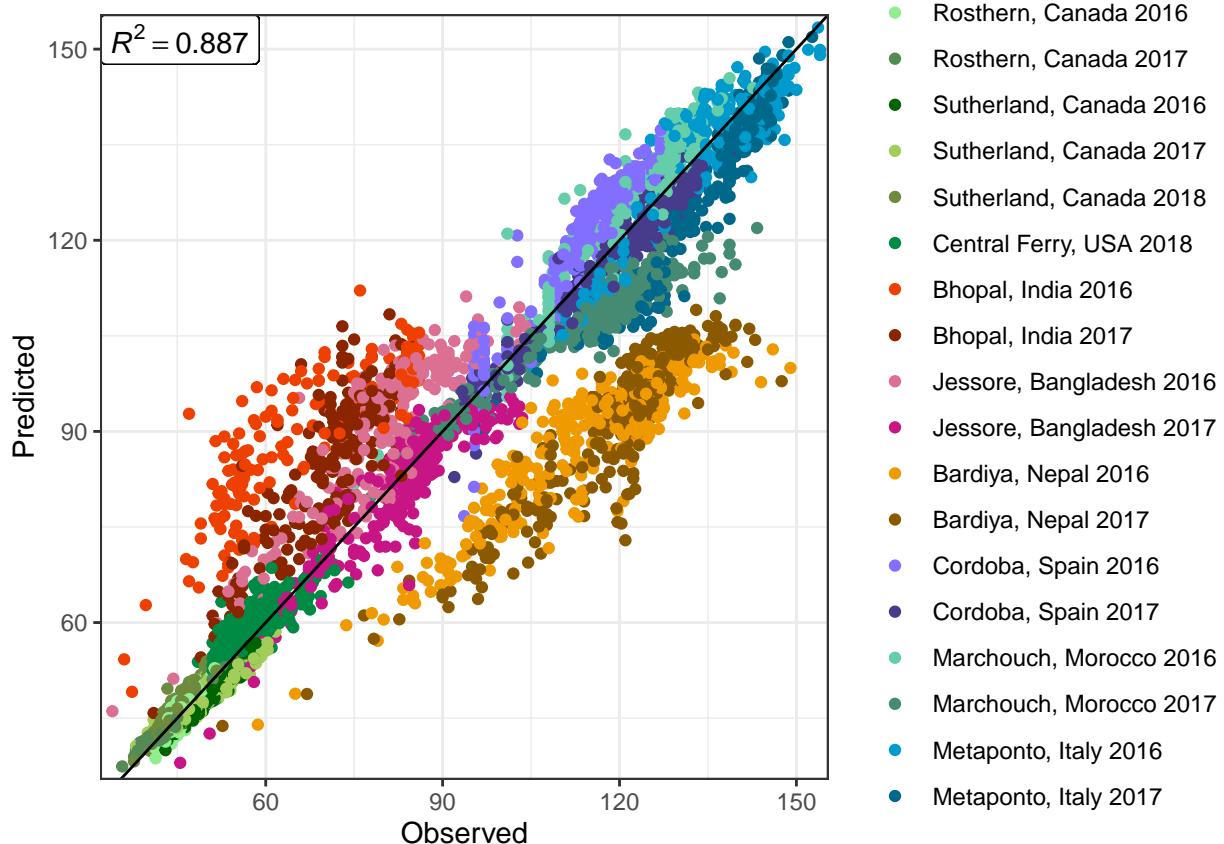
pdf("Additional/Plots_Model.pdf", width = 12, height = 5)
for(i in 1:324) {
  mp1 <- gg_model_1(mr %>% filter(Entry == i), paste("Entry", i, "| DTF"),
    xmin = min(x$DTF)-2, xmax = max(x$DTF)+2,
    ymin = min(x$Predicted_DTF)-2, ymax = max(x$Predicted_DTF)+2)
  mp2 <- gg_model_1(mr %>% filter(Entry == i), paste("Entry", i, "| RDTF"),
    xmin = min(x$RDTF)-0.001, xmax = max(x$RDTF)+0.001,
    ymin = min(x$Predicted_RDTF)-0.001, ymax = max(x$Predicted_RDTF)+0.001, type = 2)
  print(ggarrange(mp1, mp2, ncol = 2, common.legend = T, legend = "right"))
}
dev.off()

```

```

# Prep data
xx <- read.csv("data/model_18_d.csv") %>%
  mutate(Expt = factor(Expt, levels = names_Expt))
# Plot Observed vs Predicted
mp <- gg_model_1(xx)
ggsave("Figure4_Model.png", mp, width = 7, height = 5)

```



```

# Plot Observed vs Predicted
mp <- gg_model_2(xx, title = "All Locations")
ggsave("Additional/AdditionalFigure04_Model.png", mp, width = 8, height = 5)

```

All Locations

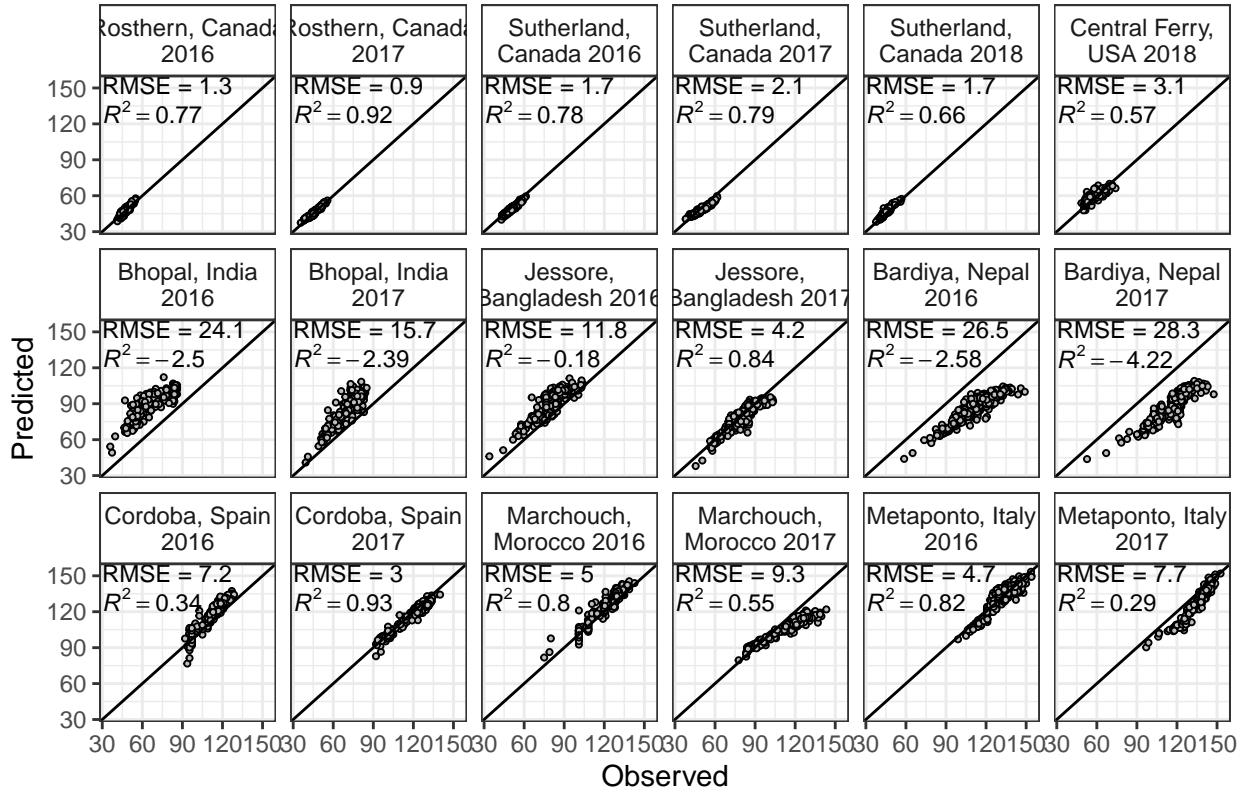


Figure 5 Test Model

Train the model without the location used for prediction

```
#####
# 1/f = a + bT + cP (Location Out) #
#####
# Prep data
xx <- rr %>% filter(!is.na(RDTF)) %>%
  left_join(select(ff, Expt, T_mean, P_mean), by = "Expt")
mr <- NULL; md <- NULL
# Model - For each Location, the model is re-trained without that locations data
for(i in 1:324) {
  for(k in unique(xx$Location)) {
    # Prep data
    xi1 <- xx %>% filter(Entry == i, Location != k)
    xi2 <- xx %>% filter(Entry == i, Location == k)
    xd2 <- xi2 %>% group_by(Expt) %>%
      summarise_at(vars(DTF, RDTF, T_mean, P_mean), funs(mean), na.rm = T)
    # Train model
    mi <- lm(RDTF ~ T_mean + P_mean, data = xi1)
    # Predict DTF
```

```

    xi2 <- xi2 %>% mutate(Predicted_DTF = 1 / predict(mi, newdata = xi2))
    xd2 <- xd2 %>% mutate(Predicted_DTF = 1 / predict(mi, newdata = xd2))
    # Save to table
    mr <- bind_rows(mr, xi2)
    md <- bind_rows(md, xd2)
  }
}

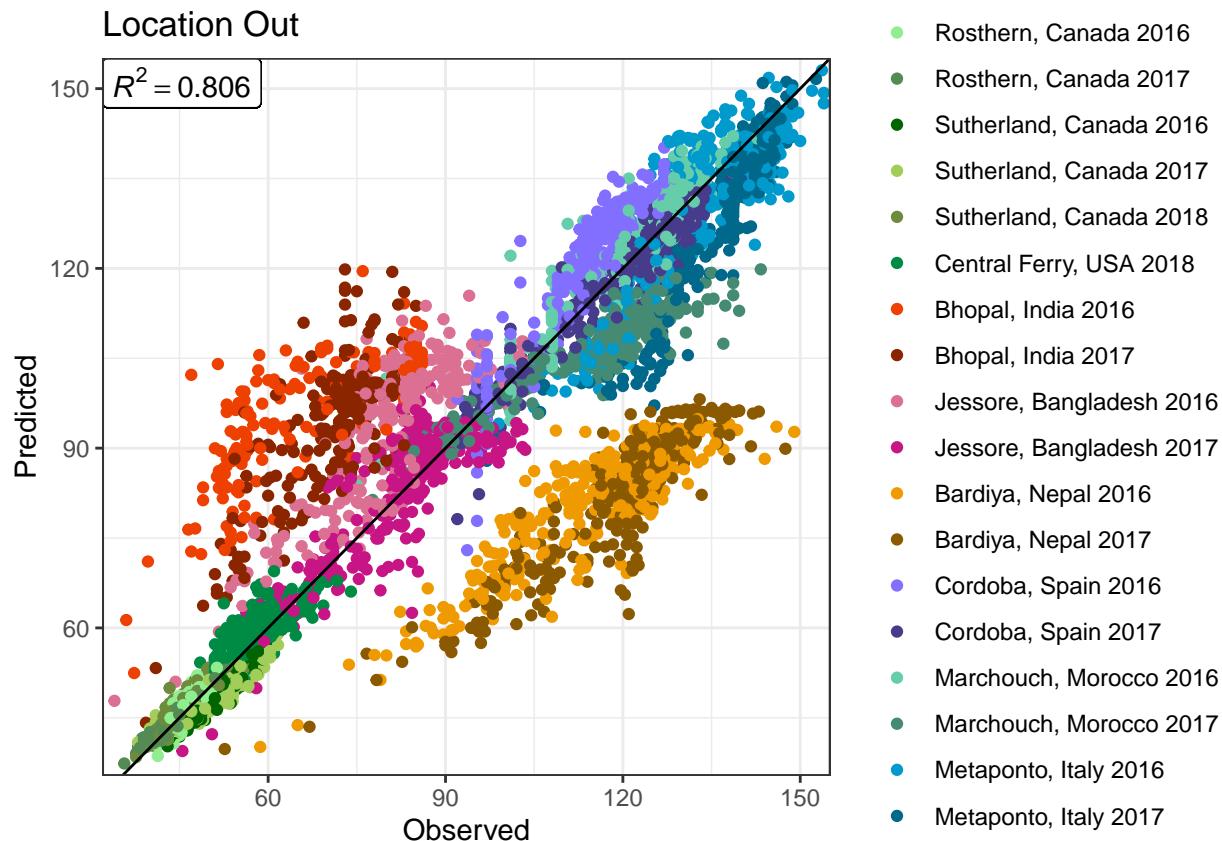
# Save Results
write.csv(mr,      "data/model_16.csv", row.names = F)
write.csv(md,      "data/model_16_d.csv", row.names = F)

```

```

# Prep data
xx <- read.csv("data/model_16_d.csv") %>%
  mutate(Expt = factor(Expt, levels = names_Expt))
# Plot Observed vs Predicted
mp <- gg_model_1(xx, "Location Out")
ggsave("Additional/AdditionalFigure05_ModelLocationOut.png", mp, width = 7, height = 5)

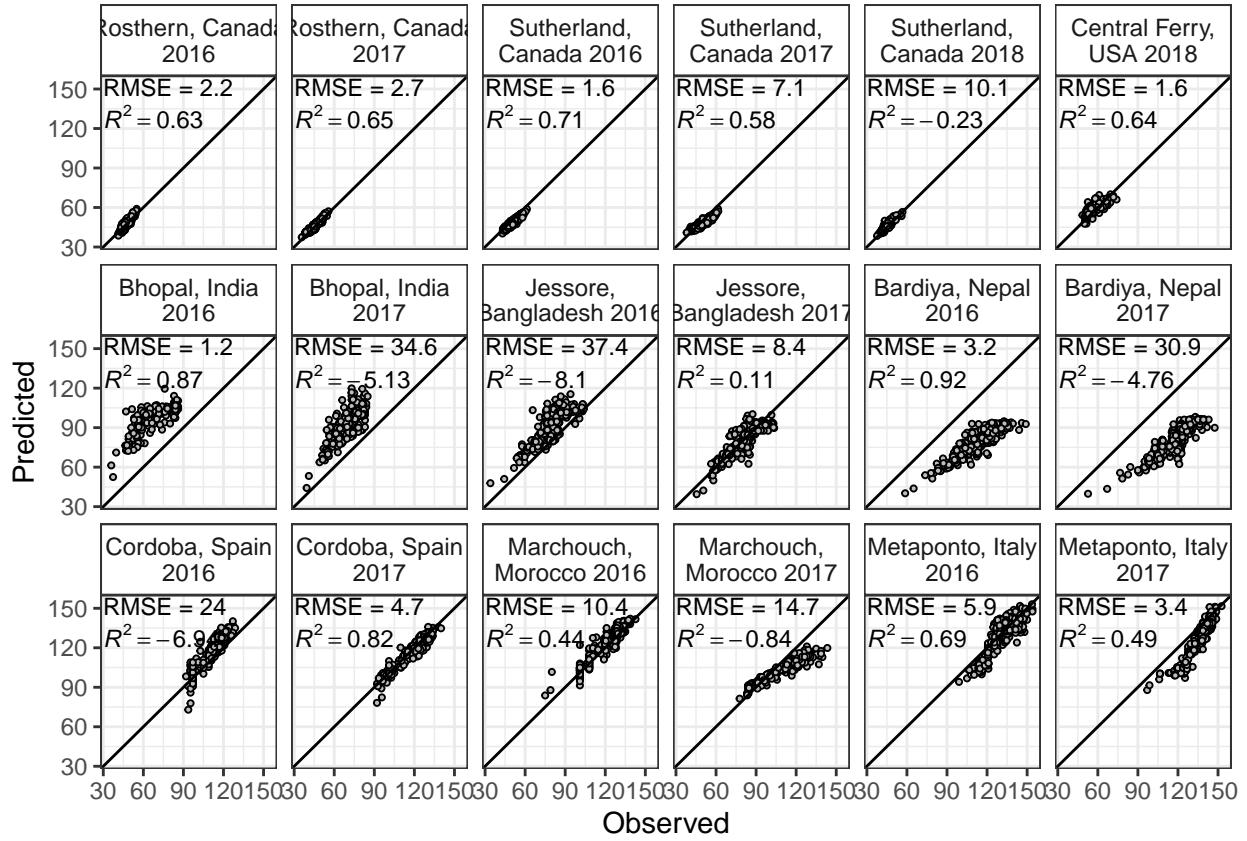
```



```

# Plot A)
mp <- gg_model_2(xx)
ggsave("Temp/Temp_F5_1.png", mp, width = 8, height = 5)

```



Train the model using only one siteyear from each macro-environment

```
#####
# 1/f = a + bT + cP (3 Locations) #
#####

# Prep data
xx <- rr %>% filter(!is.na(RDTF)) %>%
  left_join(select(ff, Expt, T_mean, P_mean), by = "Expt")
mr <- NULL; md <- NULL
mc <- select(ldp, Entry, Name) %>%
  mutate(a = NA, b = NA, c = NA, RR = NA, Environments = NA)
k <- c("Rosthern, Canada 2017", "Jessore, Bangladesh 2017", "Cordoba, Spain 2017")
# Model - only the ^above^ three site-years are used to train the model
for(i in 1:324) {
  # Prep data
  xi1 <- xx %>% filter(Entry == i, Expt %in% k)
  xi2 <- xx %>% filter(Entry == i)
  xd2 <- xi2 %>% group_by(Expt) %>%
    summarise_at(vars(DTF, RDTF, T_mean, P_mean), funs(mean), na.rm = T)
  # Train model
  mi <- lm(RDTF ~ T_mean + P_mean, data = xi1)
  # Predict DTF
  xi2 <- xi2 %>% mutate(Predicted_DTF = 1 / predict(mi, newdata = xi2))
  xd2 <- xd2 %>% mutate(Predicted_DTF = 1 / predict(mi, newdata = xd2))
  # Save to table
  mr <- bind_rows(mr, xi2)
```

```

md <- bind_rows(md, xd2)
# Save coefficients
mc[i,c(3:5)] <- mi$coefficients
# Calculate rr and # of environments used
mc[i,6] <- 1 - sum((xi2$DTF - xi2$Predicted_DTF)^2) /
  sum((xi2$Predicted_DTF - mean(xi2$DTF))^2)
mc[i,7] <- length(unique(xi2$Expt))
}

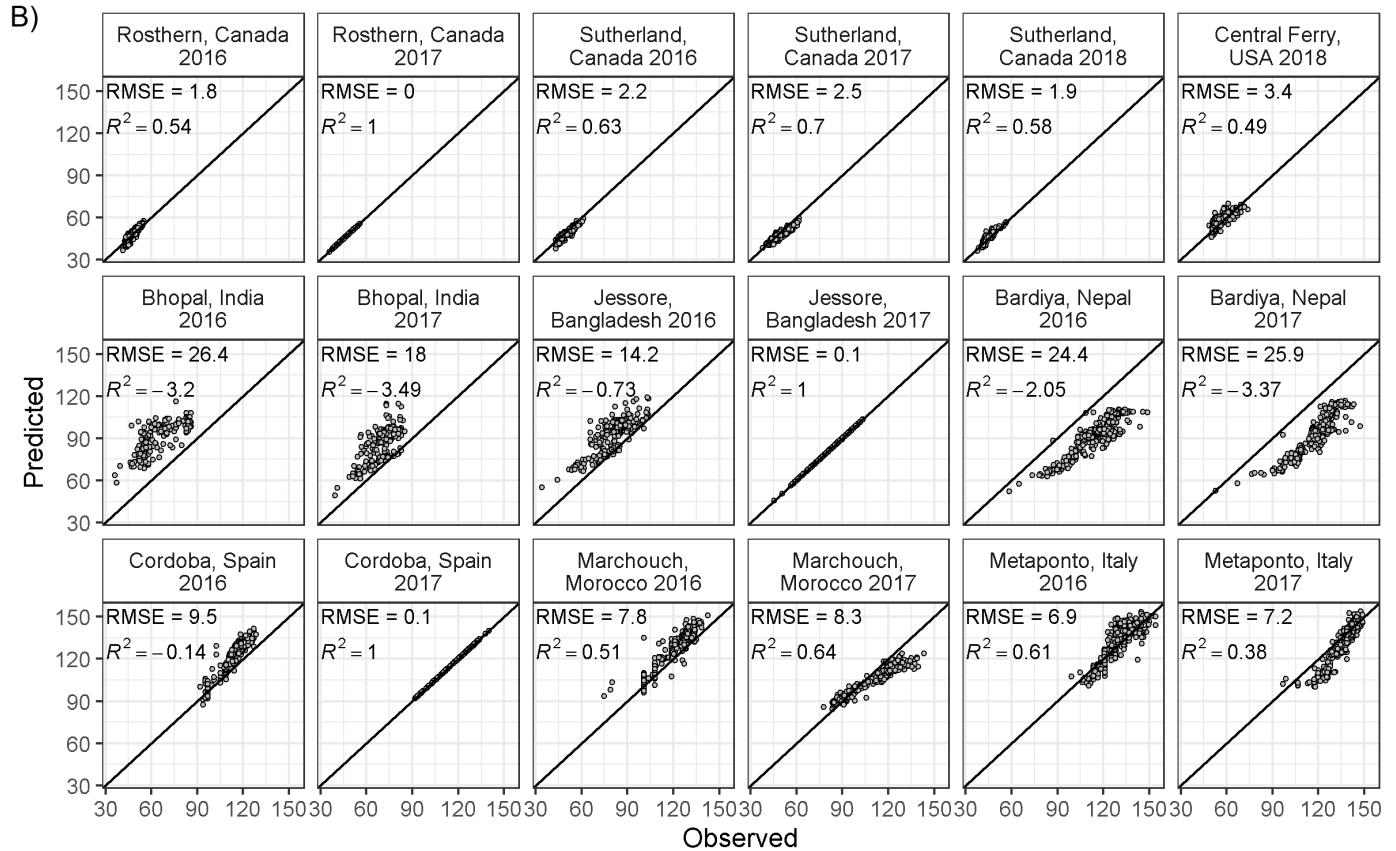
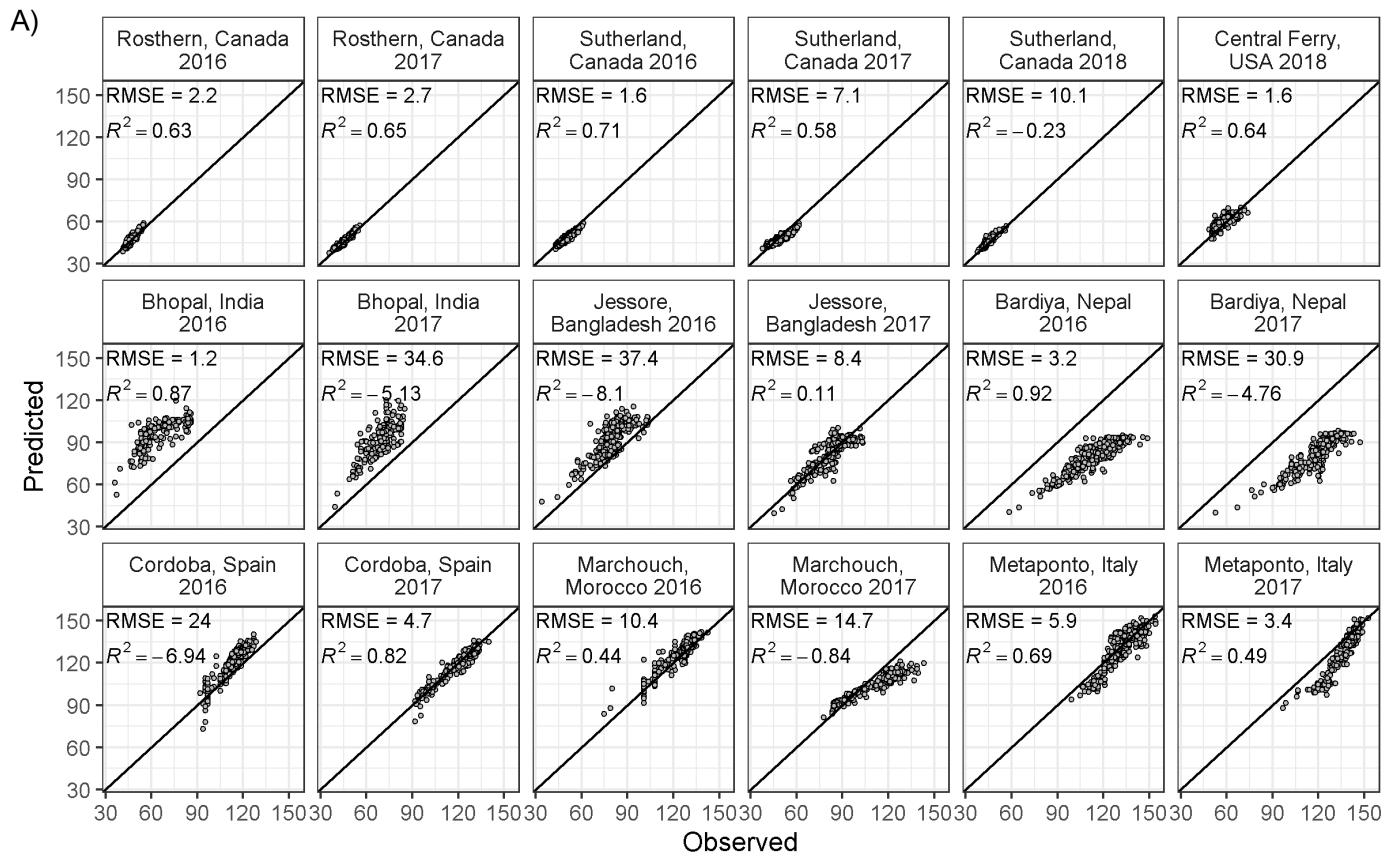
# Save Results
write.csv(mr,      "data/model_3.csv",           row.names = F)
write.csv(md,      "data/model_3_d.csv",          row.names = F)
write.csv(mc, "data/model_3_coefs.csv",   row.names = F)

# Prep data
xx <- read.csv("data/model_3_d.csv") %>%
  mutate(Expt = factor(Expt, levels = names_Expt))
# Plot Observed vs Predicted
mp <- gg_model_1(xx, "3 Locations")
ggsave("Additional/AdditionalFigure06_Model3.png", mp, width = 7, height = 5)

# Plot B)
mp <- gg_model_2(xx)
ggsave("Temp/Temp_F5_2.png", mp, width = 8, height = 5)

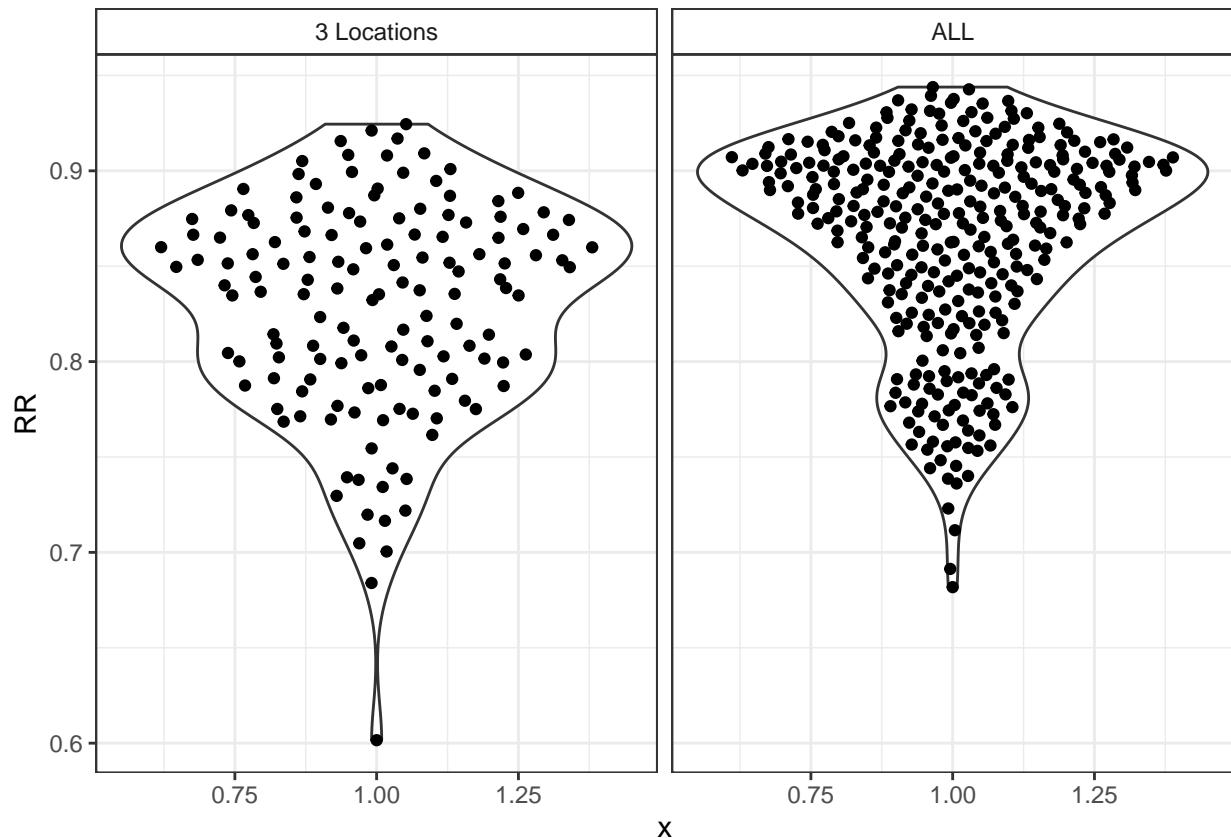
# Append A) and B)
im1 <- image_read("Temp/Temp_F5_1.png") %>%
  image_annotate("A", size = 50)
im2 <- image_read("Temp/Temp_F5_2.png") %>%
  image_annotate("B", size = 50)
im <- image_append(c(im1, im2), stack = T)
image_write(im, "Figure5_TestModel.png")

```



Check correlation coefficients

```
# Prep data
x1 <- read.csv("data/model_18_coefs.csv") %>% mutate(Type = "ALL")
x2 <- read.csv("data/model_3_coefs.csv") %>% mutate(Type = "3 Locations")
xx <- bind_rows(x1, x2)
# Plot RR
ggplot(xx, aes(x = 1, y = RR)) +
  geom_violin() + geom_quasirandom() +
  facet_grid(. ~ Type) +
  theme_AGL
```



Supplemental Figure 4 Compare Constants Entry

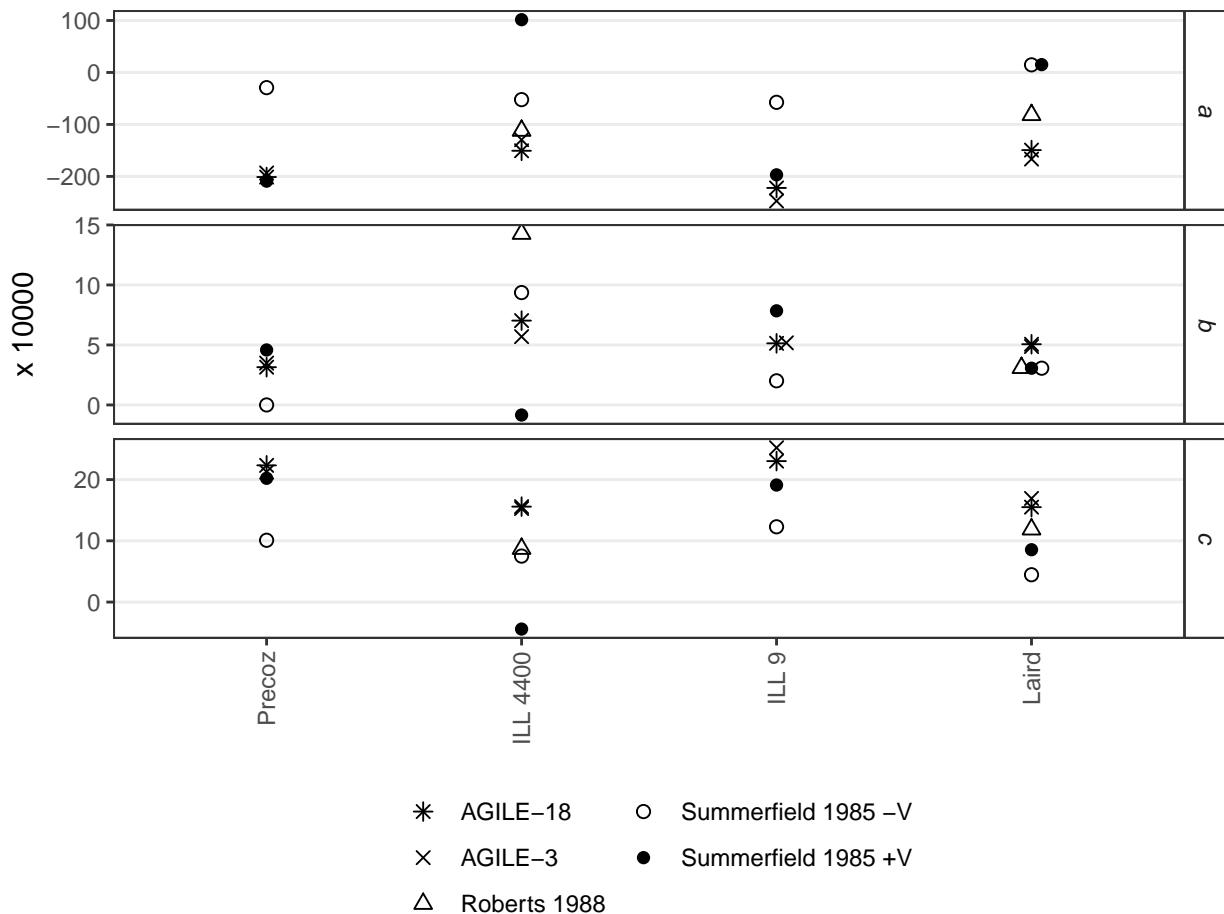
- Entry 76 = ILL 4400 (Syrian Local Large)
- Entry 77 = ILL 4605 (Precoz)
- Entry 118 = ILL 9
- Entry 128 = Laird

```
# Prep data
x1 <- read.csv("data/model_18_coefs.csv") %>%
```

```

filter(Entry %in% c(76, 77, 118, 128)) %>%
  mutate(Expt = "AGILE-18")
x2 <- read.csv("data/model_3_coefs.csv") %>%
  filter(Entry %in% c(76, 77, 118, 128)) %>%
  mutate(Expt = "AGILE-3")
# Summerfield et al., 1985
x3 <- x1 %>% mutate(Expt = "Summerfield 1985 -V")
x3[x3$Entry == 76, c("a","b","c")] <- c(-0.002918, 0, 0.0010093)
x3[x3$Entry == 77, c("a","b","c")] <- c(-0.0052226, 0.00093643, 0.00075104)
x3[x3$Entry == 118, c("a","b","c")] <- c(-0.0057408, 0.00020113, 0.0012292)
x3[x3$Entry == 128, c("a","b","c")] <- c( 0.0014689, 0.00030622, 0.00044640)
x4 <- x1 %>% mutate(Expt = "Summerfield 1985 +V")
x4[x4$Entry == 76, c("a","b","c")] <- c(-0.020910, 0.00045813, 0.0020210)
x4[x4$Entry == 77, c("a","b","c")] <- c( 0.0101590, -0.00008401, -0.00044067)
x4[x4$Entry == 118, c("a","b","c")] <- c(-0.0196948, 0.00078441, 0.0019110)
x4[x4$Entry == 128, c("a","b","c")] <- c( 0.0015094, 0.00030622, 0.00085502)
# Roberts et al., 1988
x5 <- x1 %>% filter(Entry %in% c(77, 128)) %>% mutate(Expt = "Roberts 1988")
x5[x5$Entry == 77, c("a","b","c")] <- c(-0.0112, 0.001427, 0.000871)
x5[x5$Entry == 128, c("a","b","c")] <- c(-0.008172, 0.000309, 0.001187)
#
xx <- bind_rows(x1, x2, x3, x4, x5) %>%
  gather(Constant, Value, a, b, c) %>%
  mutate(Entry = factor(Entry),
         Name = plyr:::mapvalues(Entry, c(76, 77, 118, 128),
                                   c("Precoz", "ILL 4400", "ILL 9", "Laird")),
         Expt = factor(Expt, levels = c("AGILE-18", "AGILE-3", "Roberts 1988",
                                         "Summerfield 1985 -V", "Summerfield 1985 +V")))
# Plot a, b and c
mp <- ggplot(xx, aes(x = Name, y = Value * 10000, shape = Expt)) +
  geom_beeswarm(size = 2) +
  facet_grid(Constant ~ ., scales = "free_y") +
  scale_shape_manual(name = NULL, values = c(8,4,2,1,16)) +
  guides(shape=guide_legend(nrow = 3, byrow = F)) +
  theme_AGL +
  theme(legend.position = "bottom",
        strip.text.y = element_text(face = "italic"),
        panel.grid.major.x = element_blank(),
        panel.grid.minor = element_blank(),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust = 1)) +
  labs(x = NULL, y = "x 10000")
ggsave("SupplementalFigure4_CompareConstants.png", mp, width = 3.5, height = 5)

```



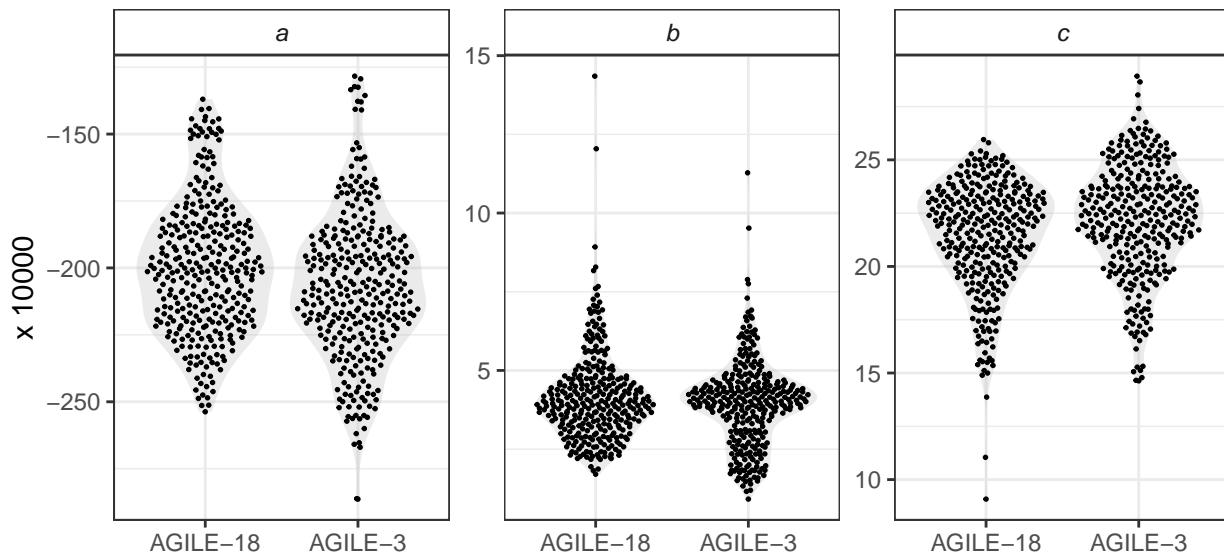
Supplemental Figure 5 Compare Constants All

```
# Prep data
x1 <- read.csv("data/model_18_coefs.csv") %>%
  mutate(Expt = "AGILE-18") %>% select(-RR)
x2 <- read.csv("data/model_3_coefs.csv") %>%
  mutate(Expt = "AGILE-3")
x3 <- read.csv("data/data_PCA_Results.csv")
xx <- bind_rows(x1, x2) %>%
  left_join(select(x3, Entry, Cluster), by = "Entry") %>%
  gather(Trait, Value, a, b, c)
# Plot a, b, and c
mp <- ggplot(xx, aes(x = Expt, y = Value * 10000 )) +
  geom_violin(alpha = 0.3, color = NA, fill = "grey") +
  geom_quasirandom(size = 0.3) +
  facet_wrap(Trait ~ ., scales = "free") +
  theme_AGL +
  theme(legend.position = "none",
```

```

    strip.text = element_text(face = "italic")) +
  labs(x = NULL, y = "x 10000")
ggsave("SupplementalFigure5_ConstantsCompare.png", mp, width = 6, height = 3)

```



Base Temperature & Critical Photoperiod

```

# Calculate Tf and Pf
xx <- read.csv("data/model_18_coefs.csv") %>% select(-Name) %>%
  mutate(predicted_Tf = 1/b, predicted_Pf = 1/c )
xx <- rr %>% left_join(xx, by = "Entry") %>%
  left_join(select(ff, Expt, T_mean, P_mean), by = "Expt") %>%
  mutate(Tb = -(a + c * P_mean) / b,
         Pc = -(a + b * T_mean) / c,
         Tf_0 = NA, Tf_5 = NA, Tf = NA, Pf = NA, PTT = NA)
for(i in 1:nrow(xx)) {
  e1 <- ee %>% filter(Expt == xx$Expt[i])
  for(k in 1:nrow(e1)) {
    e1$Tfsum[k] <- sum(e1$Temp_mean[1:k] - xx$Tb[i])
    e1$Pfsum[k] <- sum(e1$DayLength[1:k] - xx$Pc[i])
  }
  ei <- e1 %>%
    filter(Date <= xx$PlantingDate[i] + xx$DTF2[i], !is.na(Temp_mean))
  if(nrow(ei) > 0) {
    xx$Tf_0[i] <- round(sum(ei$Temp_mean), 1)
    xx$Tf_5[i] <- round(sum(ei$Temp_mean - 5), 1)
    xx$Tf[i] <- round(sum(ei$Temp_mean - xx$Tb[i]), 1)
    xx$Pf_0[i] <- round(sum(ei$DayLength), 1)
    xx$Pf_7[i] <- round(sum(ei$DayLength - 7), 1)
    xx$Pf[i] <- round(sum(ei$DayLength - xx$Pc[i]), 1)
  }
}

```

```

xx$PTT[i] <- round(sum((ei$Temp_mean - xx$Tb[i]) * (ei$DayLength - xx$Pc[i])), 1)
eTf <- e1 %>% filter(Tfsum > xx$predicted_Tf[i])
ePf <- e1 %>% filter(Pfsum > xx$predicted_Pf[i])
xx$predicted_DTF_Tf[i] <- eTf$DaysAfterPlanting[1]
xx$predicted_DTF_Pf[i] <- ePf$DaysAfterPlanting[1]
}
}

xx <- xx %>% left_join(select(ff, Expt, MacroEnv), by = "Expt") %>%
  group_by(Entry, Name, Expt, ExptShort, MacroEnv) %>%
  summarise_at(vars(DTF, Tb, Pc, Tf_0, Tf_5, Tf, Pf_0, Pf_7, Pf, PTT,
                     predicted_DTF_Tf, predicted_DTF_Pf,
                     predicted_Pf, predicted_Tf), funs(mean), na.rm = T) %>%
  ungroup()
# Save
write.csv(xx, "data/data_Tf_Pf.csv", row.names = F)

```

Figure 6 Tb and Pc

```

# Prep data
xx <- read.csv("data/model_18_coefs.csv") %>% select(-Name) %>%
  mutate(predicted_Tf = 1 / b,
        predicted_Pf = 1 / c)
xx <- rr %>% left_join(xx, by = "Entry") %>%
  left_join(select(ff, Expt, T_mean, P_mean), by = "Expt") %>%
  mutate(Tb = -(a + c * P_mean) / b,
        Pc = -(a + b * T_mean) / c)
x1 <- xx %>%
  filter(ExptShort %in% c("Su17", "Ba17", "It17")) %>%
  group_by(Entry, Name, Expt, ExptShort) %>%
  summarise_at(vars(Tb, Pc), funs(mean), na.rm = T)
# Plot A) Tb
mp1 <- ggplot(x1, aes(x = 1, y = Tb)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +
  geom_quasirandom(size = 0.3) +
  facet_grid(.~Expt, labeller = label_wrap_gen(width = 17)) +
  scale_y_continuous(breaks = seq(-80, 0, 20), minor_breaks = seq(-110, 0, 10)) +
  theme_AGL +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        panel.grid.minor.x = element_blank()) +
  labs(title = "A)", x = NULL,
       y = expression(paste(italic("T") [italic("b")], " = -(", italic("a"), " + ",
                            italic("cP"), ") / ", italic("b"))))

# Plot B) Pc
x2 <- x1 %>% filter(Entry %in% c(94, 105)) %>%
  ungroup() %>% mutate(Name = gsub(" AGL", "", Name))
mp2 <- ggplot(x1, aes(x = 1, y = Pc)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +

```

```

geom_quasirandom(size = 0.3) +
facet_grid(.~Expt, labeller = label_wrap_gen(width = 17)) +
geom_text_repel(data = x2,
               aes(label = Name), size = 3, nudge_y = 0.5) +
scale_y_continuous(breaks = c(-20, -15, -10, -5, 0, 5)) +
theme_AGL +
theme(axis.text.x = element_blank(),
      axis.ticks.x = element_blank(),
      panel.grid.minor.x = element_blank()) +
labs(title = "B)", x = NULL,
     y = expression(paste(italic("P") [italic("c")], " = -(", italic("a"), " + ",
                           italic("bT")), " / ", italic("c"))))

# Append A) and B)
mp <- ggarrange(mp1, mp2, nrow = 1, ncol = 2)
ggsave("Figure6_TbPc.png", mp, width = 8, height = 3)
ggsave("Temp/Temp_F6_1.png", mp1, width = 4, height = 3)
ggsave("Temp/Temp_F6_2.png", mp2, width = 4, height = 3)

```

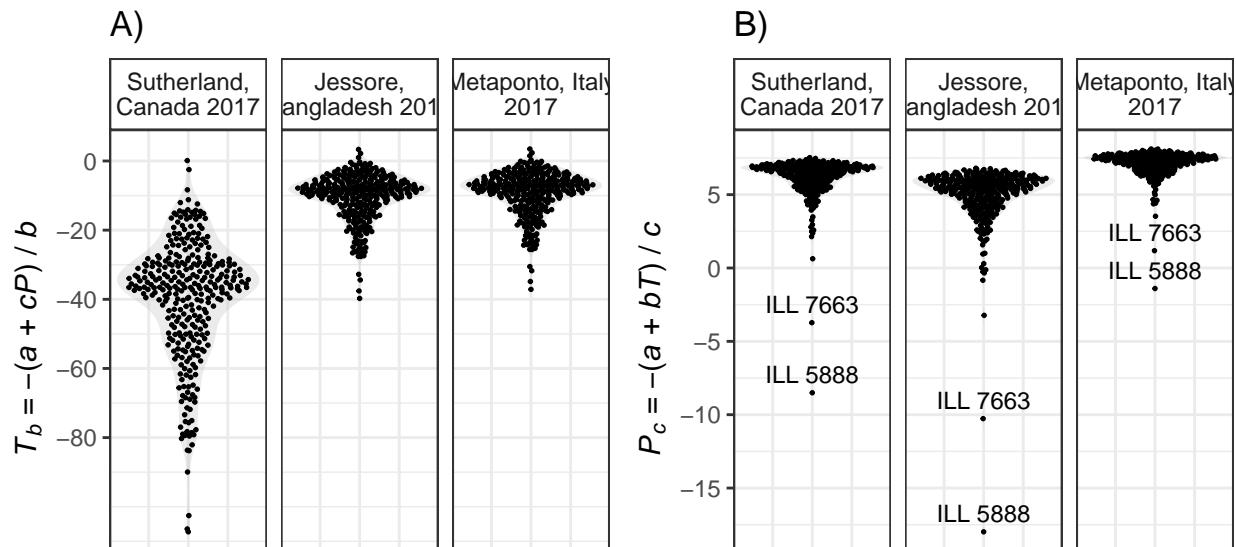


Figure 7 Tf

```

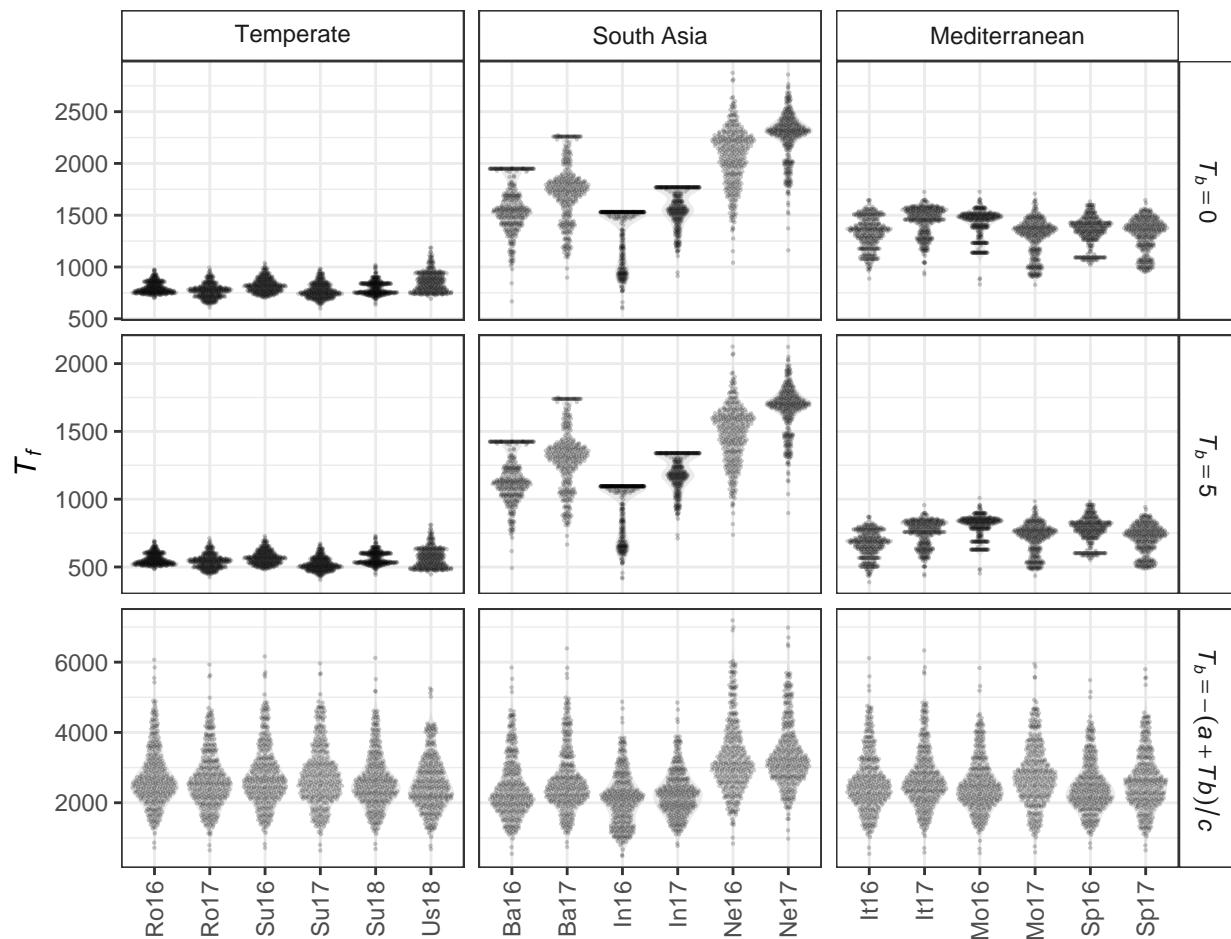
# Prep data
xx <- read.csv("data/data_Tf_Pf.csv") %>%
  mutate(MacroEnv = factor(MacroEnv, levels = macroEnvs))
x1 <- xx %>% select(Entry, Name, Expt, ExptShort, MacroEnv, Tf_0, Tf_5, Tf) %>%
  gather(Trait, Value, Tf_0, Tf_5, Tf) %>%
  mutate(Trait = factor(Trait, levels = c("Tf_0", "Tf_5", "Tf")))
new.lab <- as_labeller(c(
  Tf_0="italic(T)[italic(b)]==0", Tf_5="italic(T)[italic(b)]==5",
  Tf="italic(T)[italic(b)]==-(italic(a)+italic(Tb))/italic(c)", Mediterranean="Mediterranean",
  )

```

```

Temperate="Temperate", `South Asia`="South~Asia"), label_parsed)
# Plot Tf
mp <- ggplot(x1, aes(x = ExptShort, y = Value)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +
  geom_quasirandom(size = 0.1, alpha = 0.2) +
  facet_grid(Trait~MacroEnv, scales = "free", labeller = new.lab) +
  theme_AGL +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  labs(y = expression(italic("T")["f"])[italic("f")]), x = NULL)
ggsave("Figure7_Tf.png", mp, width = 7, height = 5)

```



Additional Figure 7 Pc

```

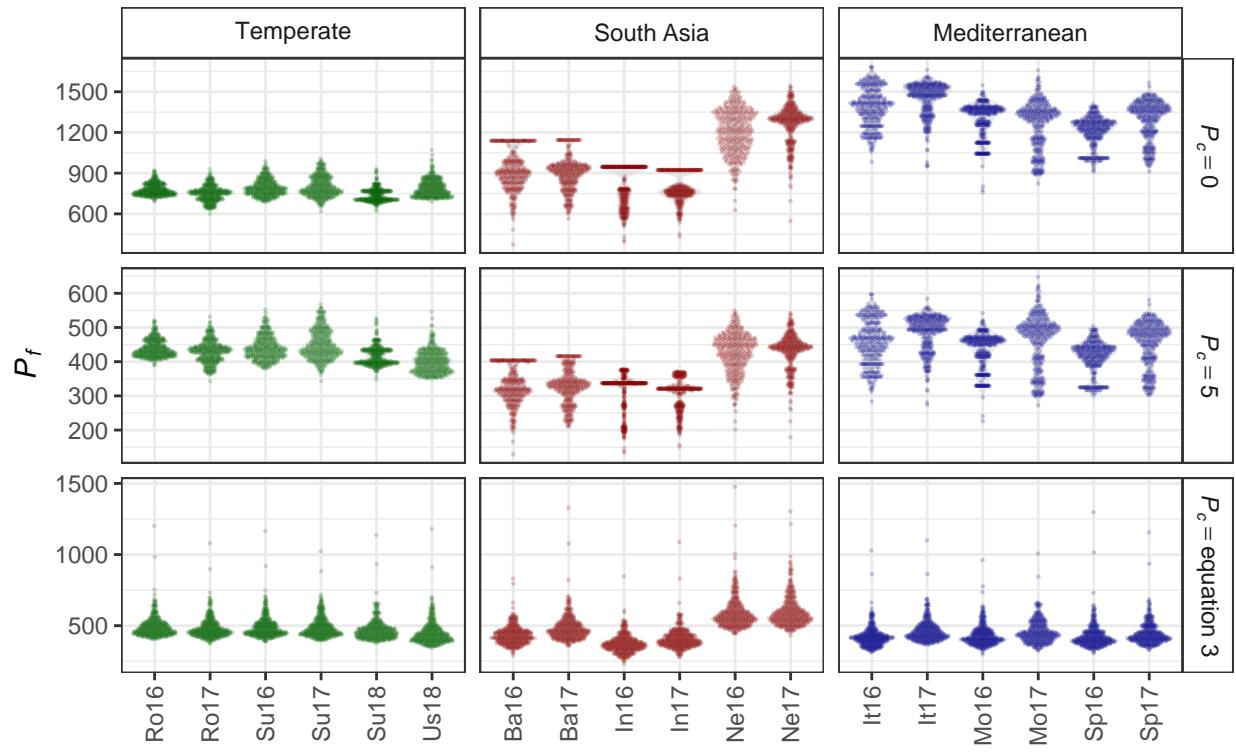
# Prep data
xx <- read.csv("data/data_Tf_Pf.csv") %>%
  mutate(MacroEnv = factor(MacroEnv, levels = macroEnvs))

```

```

x1 <- xx %>% select(Entry, Expt, ExptShort, MacroEnv, Pf_0, Pf_7, Pf) %>%
  gather(Trait, Value, Pf_0, Pf_7, Pf) %>%
  mutate(Trait = factor(Trait, levels = c("Pf_0", "Pf_7", "Pf")))
new.lab <- as_labeller(c(
  Pf_0="italic(P)[italic(c)]==0", Pf_7="italic(P)[italic(c)]==5",
  Pf="italic(P)[italic(c)]==equation~3", Mediterranean="Mediterranean",
  Temperate="Temperate", `South Asia`="South~Asia"), label_parsed)
# Plot Tf
mp <- ggplot(x1, aes(x = ExptShort, y = Value)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +
  geom_quasirandom(size = 0.1, alpha = 0.2, aes(color = MacroEnv)) +
  facet_grid(Trait~MacroEnv, scales = "free", labeller = new.lab) +
  scale_color_manual(values = c("darkgreen", "darkred", "darkblue")) +
  theme_AGL +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  labs(y = expression(italic("P")~[italic("f")]), x = NULL)
ggsave("Additional/AdditionalFigure07_Pc.png", mp, width = 6, height = 4)

```



Additional Figure 8 Tf Pc PTT

```

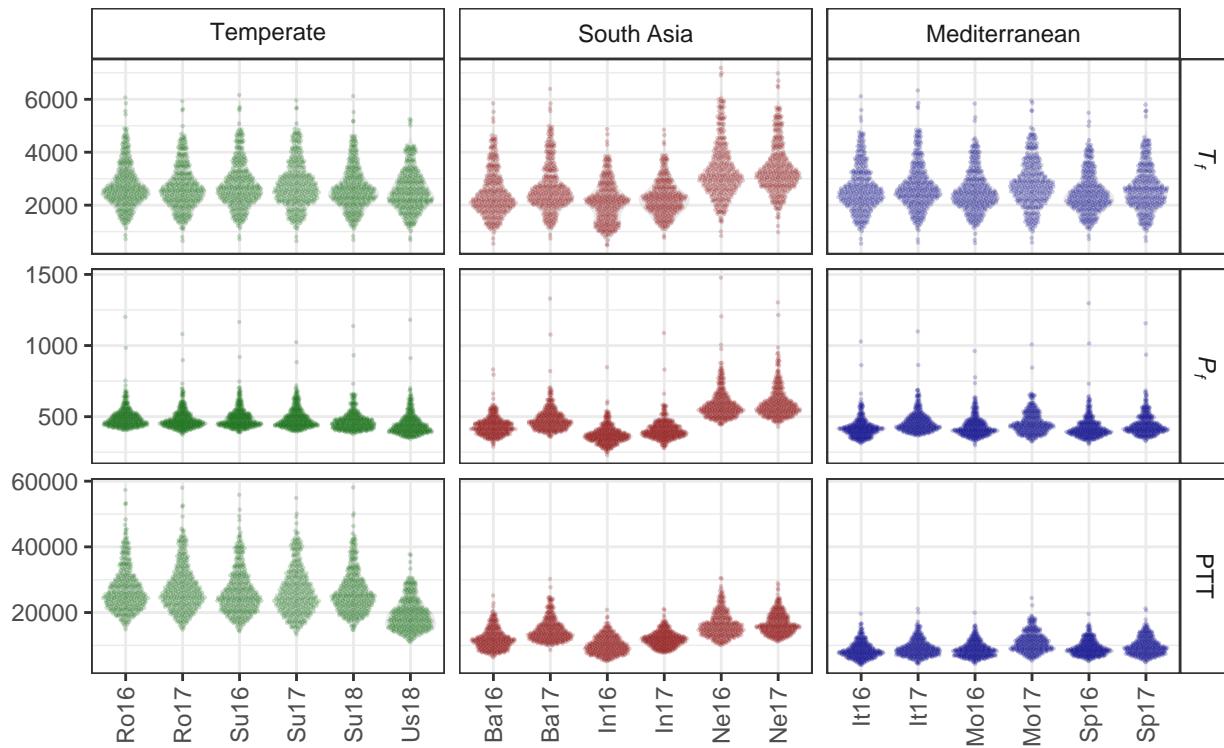
# Prep data
xx <- read.csv("data/data_Tf_Pf.csv") %>%

```

```

    mutate(MacroEnv = factor(MacroEnv, levels = macroEnvs))
x1 <- xx %>% select(Entry, Expt, ExptShort, MacroEnv, PTT, Tf, Pf) %>%
  gather(Trait, Value, PTT, Tf, Pf) %>%
  mutate(Trait = factor(Trait, levels = c("Tf", "Pf", "PTT")))
new.lab <- as_labeller(c(
  Tf="italic(T)[italic(f)]", Pf="italic(P)[italic(f)]", PTT="PTT",
  #Tf="italic(T)[b]==equation~2", Pf="italic(P)[c]==equation~3",
  Mediterranean="Mediterranean", Temperate="Temperate", `South Asia`="South~Asia"),
  label_parsed)
# Plot
mp <- ggplot(x1, aes(x = ExptShort, y = Value)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +
  geom_quasirandom(size = 0.1, alpha = 0.2, aes(color = MacroEnv)) +
  facet_grid(Trait~MacroEnv, scales = "free", labeller = new.lab) +
  scale_color_manual(values = c("darkgreen", "darkred", "darkblue")) +
  theme_AGL +
  theme(legend.position = "none",
        axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  labs(y = NULL, x = NULL)
ggsave("Additional/AdditionalFigure08_TfPcPTT.png", mp, width = 6, height = 4)

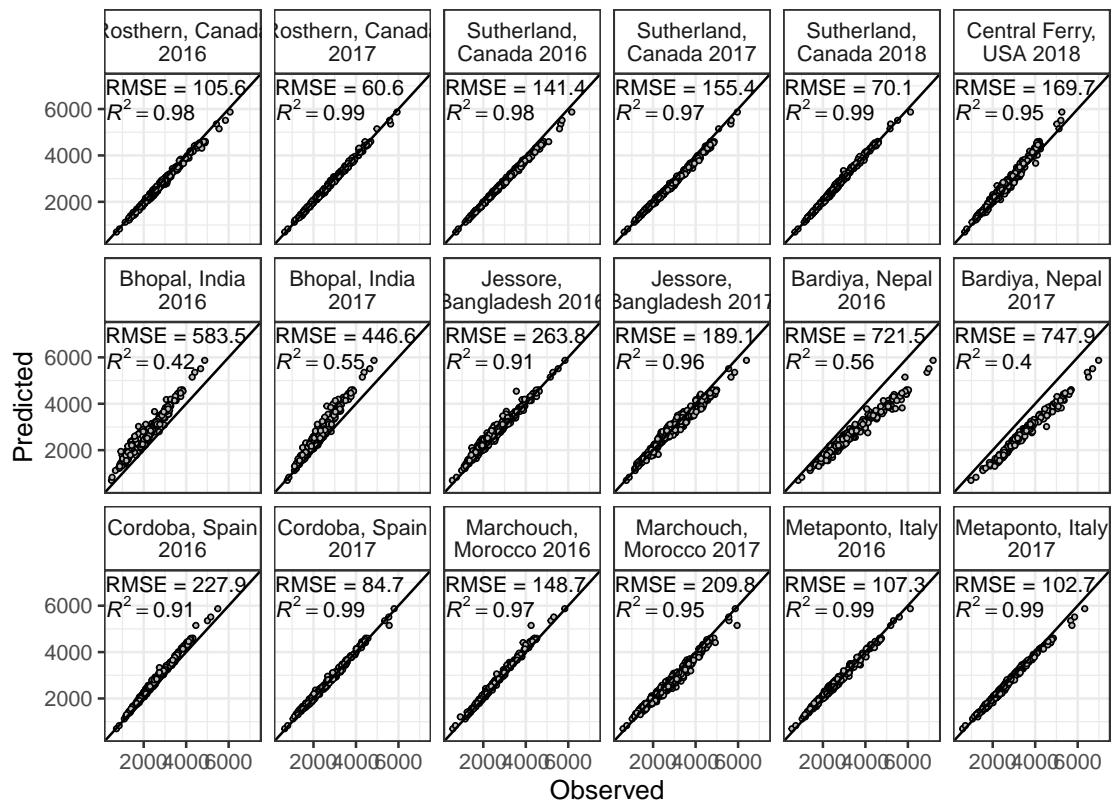
```



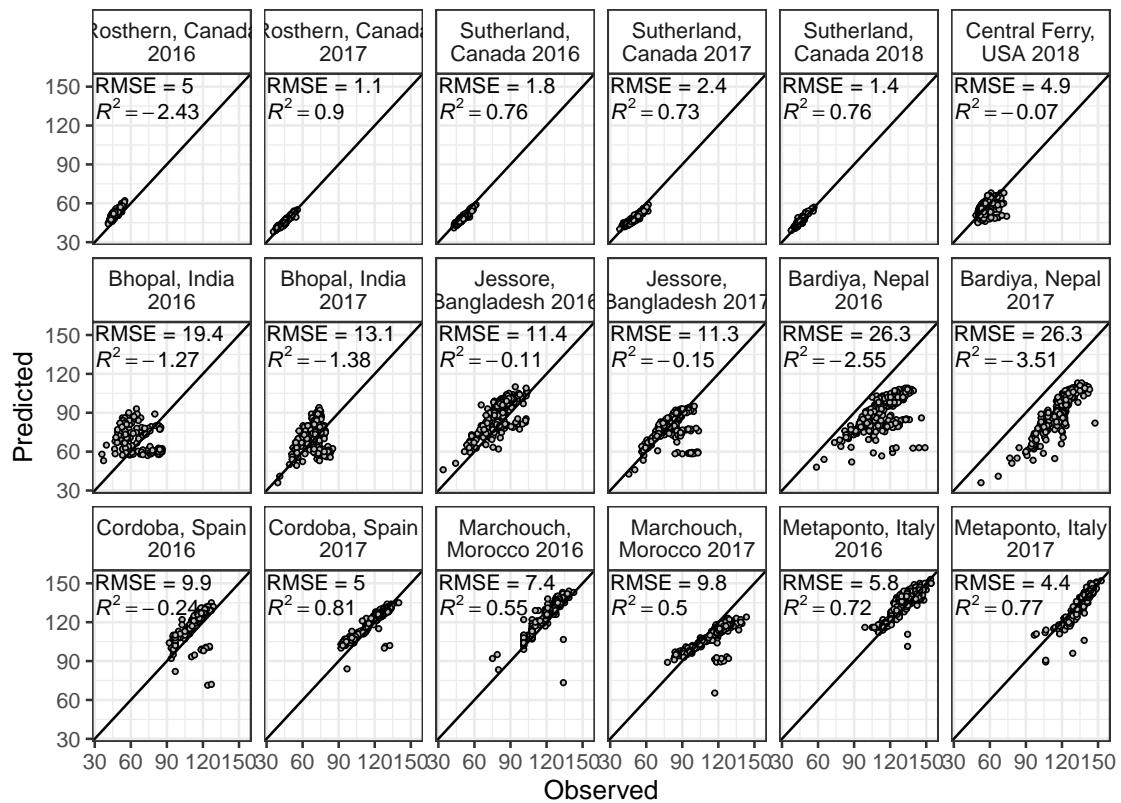
Supplemental Figure 6 Thermal Sums

```
# Prep data
xx <- read.csv("data/data_Tf_Pf.csv") %>%
  mutate(Expt = factor(Expt, levels = names_Expt),
        MacroEnv = factor(MacroEnv, levels = macroEnvs))
# Plot A)
mp1 <- gg_model_2(xx, "Tf", "predicted_Tf", "A) Thermal sum required for flowering",
                   200, 200, 6600, 5500)
# Plot B)
mp2 <- gg_model_2(xx, "DTF", "predicted_DTF_Tf", "B) Days from sowing to flower",
                   30, 30, 145, 125)
# Append A) and B)
mp <- ggarrange(mp1, mp2, nrow = 2, ncol = 1)
ggsave("SupplementalFigure6_Tf.png", mp, width = 8, height = 10)
```

A) Thermal sum required for flowering



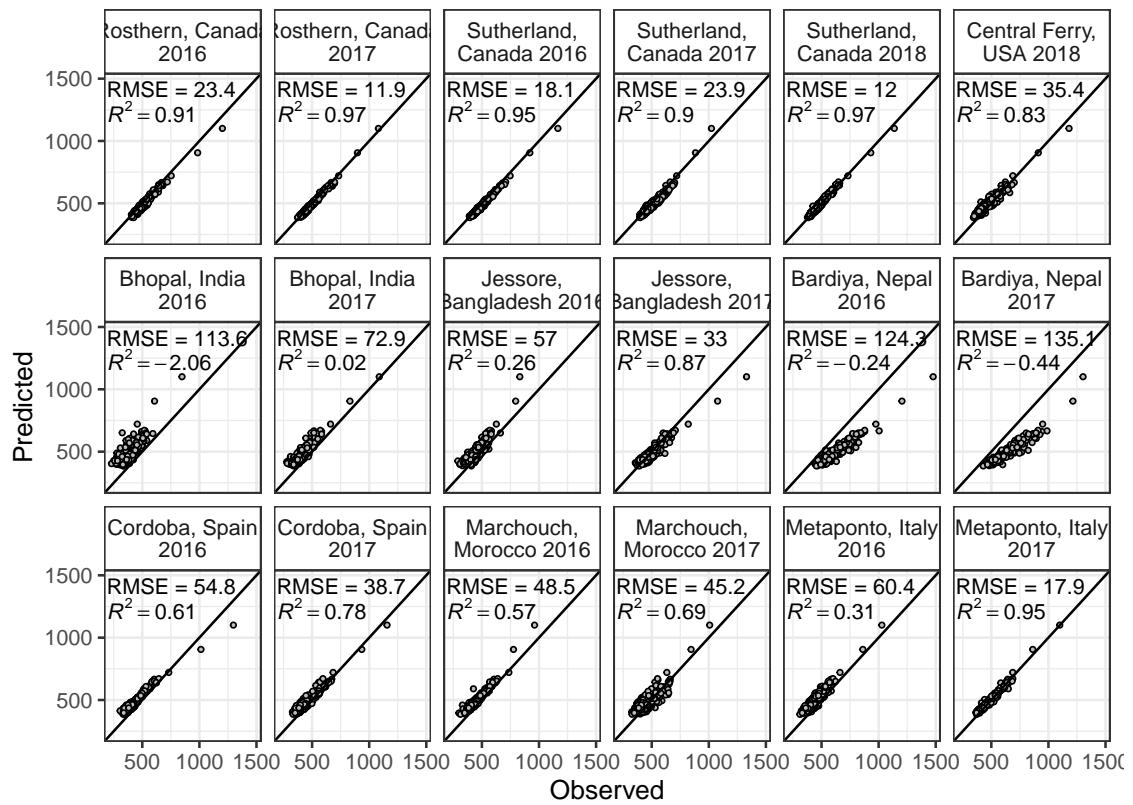
B) Days from sowing to flower



Supplemental Figure 7 Photoperiodic Sums

```
# Prep data
xx <- read.csv("data/data_Tf_Pf.csv") %>%
  mutate(Expt = factor(Expt, levels = names_Expt),
        MacroEnv = factor(MacroEnv, levels = macroEnvs))
# Plot A)
mp1 <- gg_model_2(xx, "Pf", "predicted_Pf", "A) Thermal sum required for flowering",
                    190, 190, 1350, 1150)
# Plot B)
mp2 <- gg_model_2(xx, "DTF", "predicted_DTF_Pf", "B) Days from sowing to flower",
                    30, 30, 145, 125)
# Append A) and B)
mp <- ggarrange(mp1, mp2, nrow = 2, ncol = 1)
ggsave("SupplementalFigure7_Pf.png", mp, width = 8, height = 10)
```

A) Thermal sum required for flowering



B) Days from sowing to flower

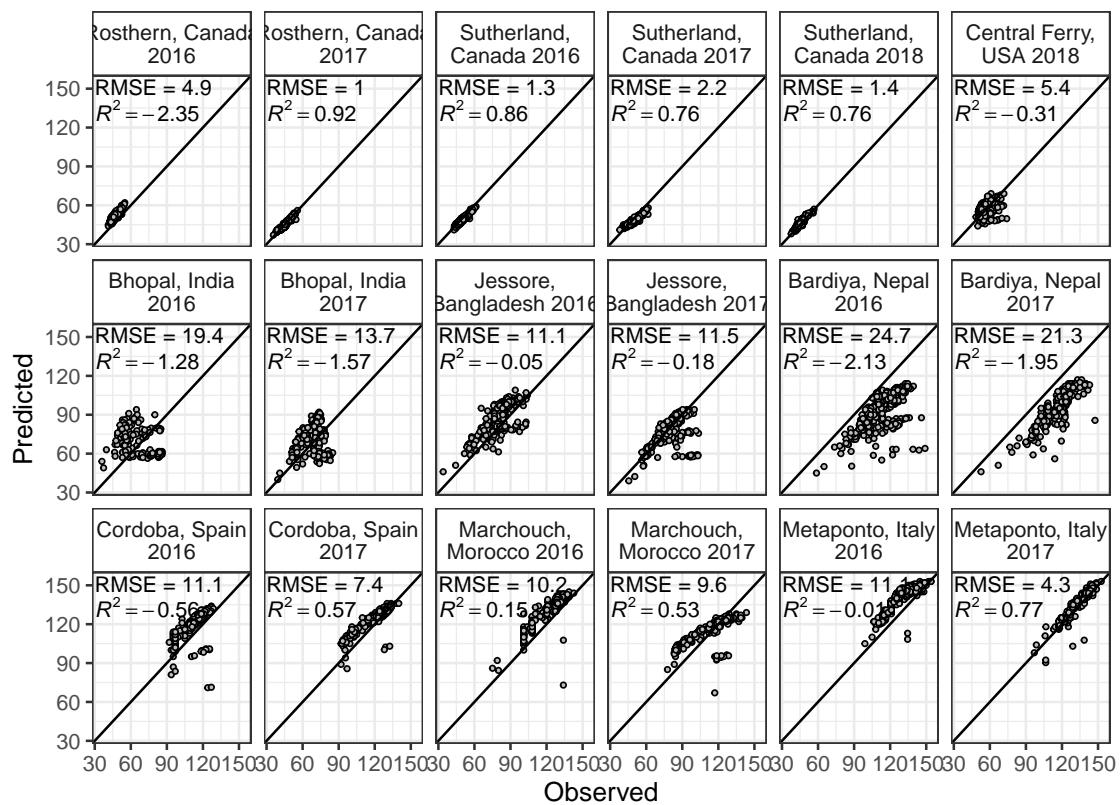


Figure 8 Temperature Increase By MacroEnv

```
# Prep data
yy <- c("Ro17", "Su17", "Us18", "In17", "Ba17", "Ne17", "Sp17", "Mo17", "It17")
xx <- read.csv("data/model_18_coefs.csv")
x1 <- dd %>%
  select(Entry, Expt, ExptShort, DTF) %>%
  left_join(xx, by = "Entry") %>%
  left_join(select(ff, Expt, MacroEnv, T_mean, P_mean), by = "Expt") %>%
  mutate(T_mean2 = T_mean + 1,
    DTF_GW = 1 / (a + b * T_mean2 + c * P_mean),
    DTF_P = 1 / (a + b * T_mean + c * P_mean),
    Diff = DTF_P - DTF_GW) %>%
  filter(ExptShort %in% yy)
x2 <- dd %>%
  select(Entry, Expt, ExptShort, DTF) %>%
  left_join(xx, by = "Entry") %>%
  left_join(select(ff, Expt, MacroEnv, T_mean, P_mean), by = "Expt") %>%
  mutate(T_mean2 = T_mean + 2,
    DTF_GW = 1 / (a + b * T_mean2 + c * P_mean),
    DTF_P = 1 / (a + b * T_mean + c * P_mean),
    Diff = DTF_P - DTF_GW) %>%
  filter(ExptShort %in% yy)
x1 <- x1 %>% mutate(GW = "A")
x2 <- x2 %>% mutate(GW = "B")
knitr::kable(x2 %>% group_by(MacroEnv) %>%
  summarise(Min = round(min(Diff), 2), Max = round(max(Diff), 2)) )
```

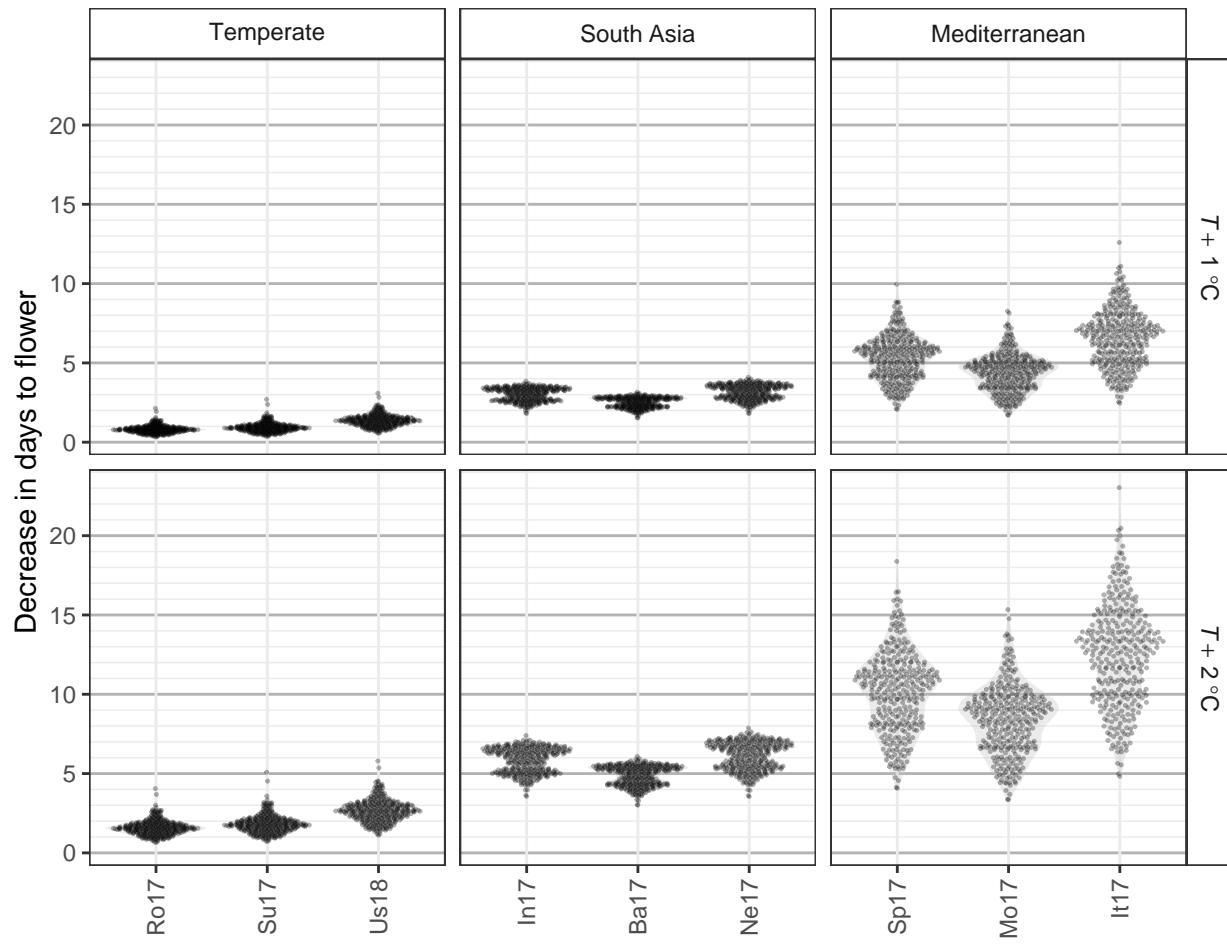
MacroEnv	Min	Max
Temperate	0.65	5.80
South Asia	3.01	7.87
Mediterranean	3.36	23.03

```
xx <- bind_rows(x1, x2)
write.csv(xx, "data/data_Temp_Increase.csv", row.names = F)
new.lab <- as_labeller(c(A=italic(T)~~~degree*C", B="italic(T)~~~degree*C",
  Mediterranean="Mediterranean", Temperate="Temperate", `South Asia`="South~Asia"),
  label_parsed)
# Plot
mp <- ggplot(xx, aes(x = ExptShort, y = Diff)) +
  geom_violin(fill = "grey", alpha = 0.3, color = NA) +
  geom_quasirandom(size = 0.2, alpha = 0.3) +
  facet_grid(GW~MacroEnv, scales = "free_x", labeller = new.lab) +
  theme_AGL +
  theme(legend.position = "none",
    axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
    panel.grid.major.y = element_line(colour = "grey70", size = 0.5)) +
```

```

scale_y_continuous(minor_breaks = seq(1,30,1),
                   breaks = seq(0,30,5)) +
  labs(y = "Decrease in days to flower", x = NULL)
ggsave("Figure8_TempIncrease.png", mp, width = 7, height = 5)

```



Additional Figure 9 Temperature Increase By Cluster

```

x1 <- read.csv("data/data_Temp_Increase.csv") %>% filter(GW == "A")
x2 <- select(read.csv("data/data_PCA_Results.csv"), Entry, Cluster)
xx <- left_join(x1, x2, by = "Entry") %>%
  mutate(Cluster = factor(Cluster)) %>%
  filter(ExptShort %in% c("Su17", "It17", "Ba17"))
mp <- ggplot(xx, aes(x = Cluster, y = Diff, fill = Cluster)) +
  geom_violin(color = NA, alpha = 0.6) +
  geom_quasirandom(color = "Black", alpha = 0.6) +
  facet_grid(ExptShort ~ ., scales = "free") +
  theme_AGL + theme(legend.position = "none")

```

```

scale_fill_manual(name = NULL, values = colors) +
  labs(title = ,
       y = "Decrease in days to flower", x = NULL)
ggsave("Additional/AdditionalFigure09_TempIncreaseByCluster.png", mp, width = 7, height = 5)

```

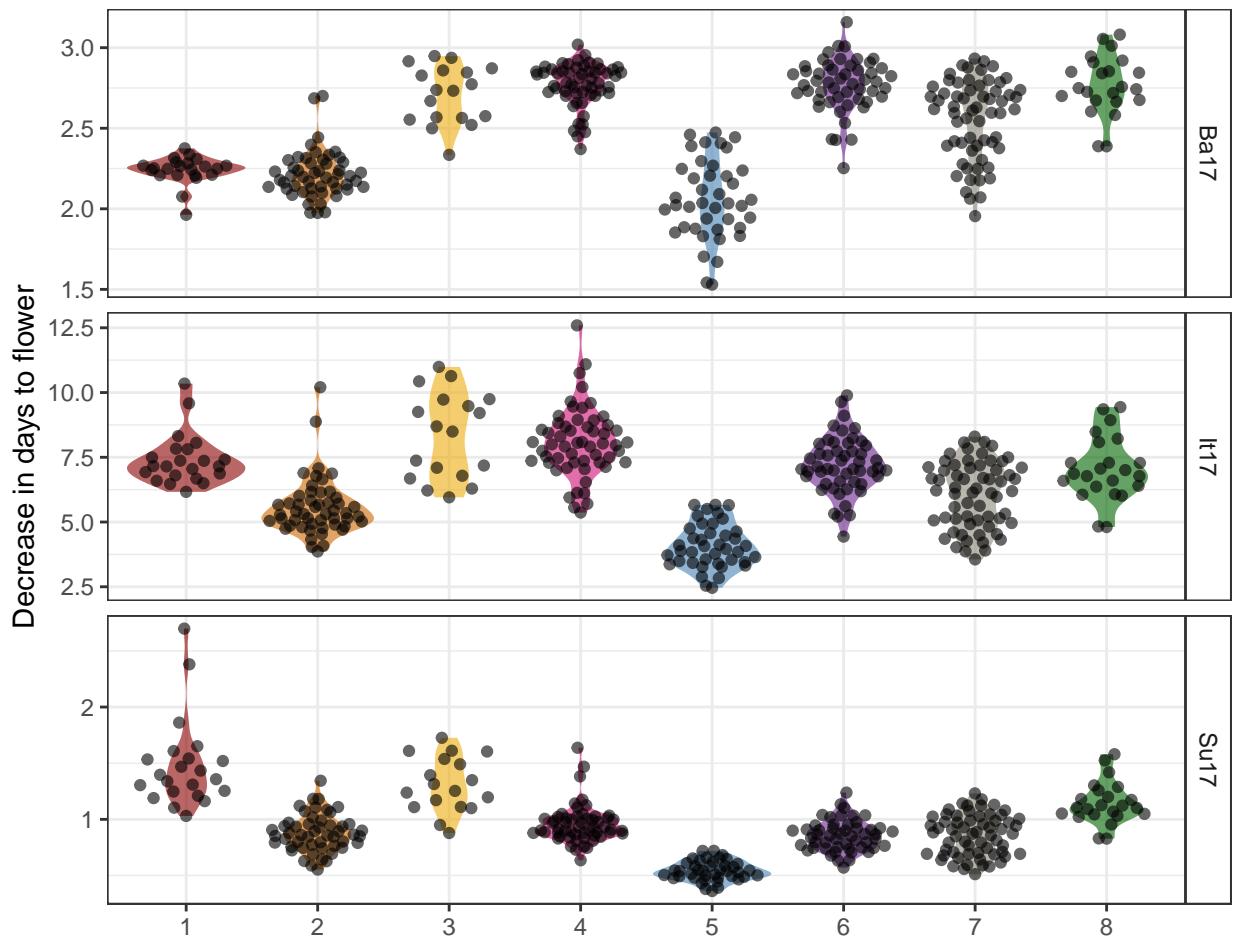


Figure 9 Origin Constants

```

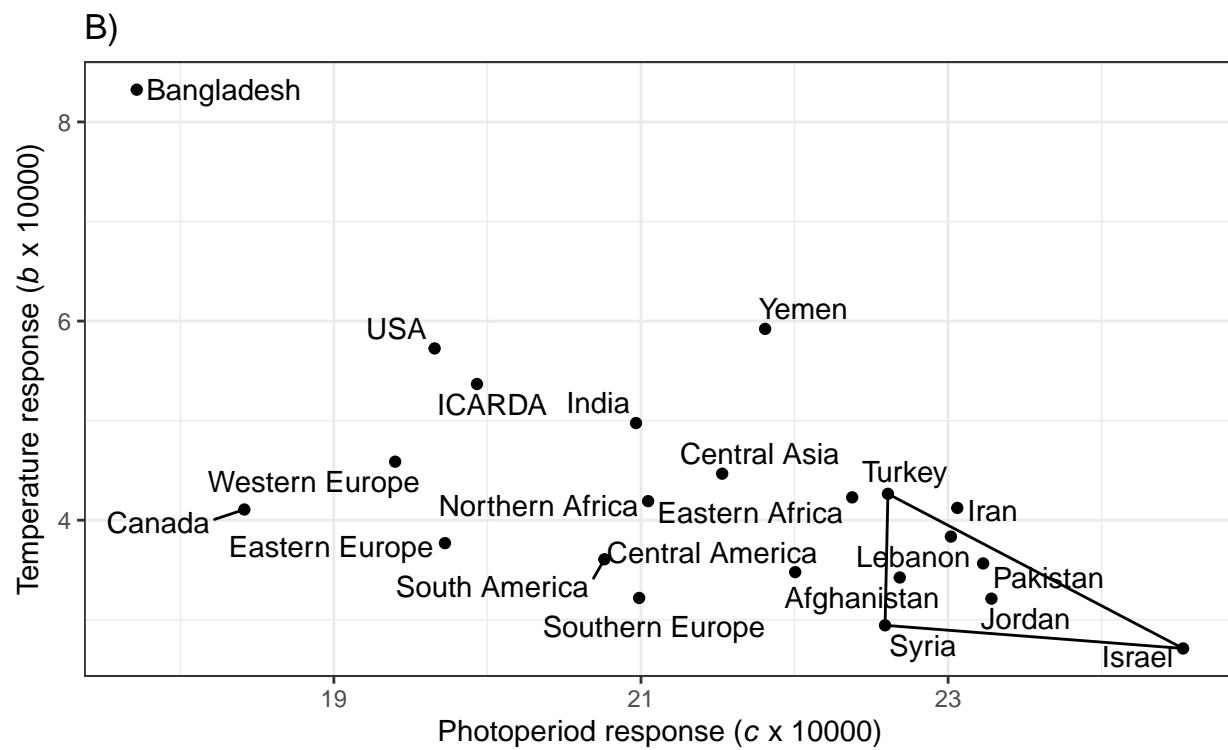
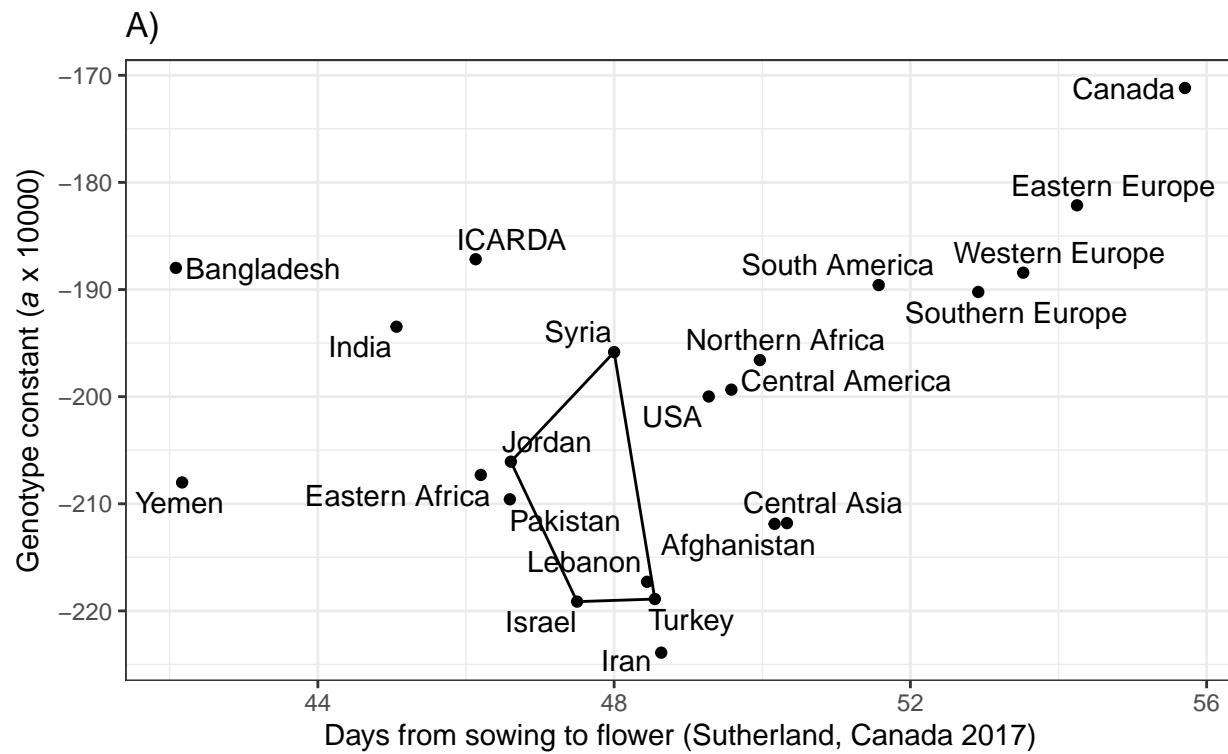
# Prep data
mycts <- c("Canada", "USA", "Iran", "Yemen",
          "India", "Pakistan", "Bangladesh", "Afghanistan",
          "Syria", "Jordan", "Turkey", "Lebanon", "Israel")
xx <- read.csv("data/model_18_coefs.csv") %>%
  left_join(select(ldp, Entry, Origin), by = "Entry") %>%
  left_join(select(ct, Origin=Country, SubRegion), by = "Origin") %>%
  mutate(SubRegion = ifelse(Origin == "ICARDA", "ICARDA", as.character(SubRegion)),
         Origin = ifelse(Origin %in% mycts, Origin, as.character(SubRegion)))
x1 <- xx %>%

```

```

left_join(dd %>% filter(ExptShort == "Su17") %>%
            select(Entry, DTF), by = "Entry") %>%
group_by(Origin) %>%
summarise_at(vars(DTF, a, b, c), funs(mean, sd)) %>%
filter(Origin != "Unknown")
x2 <- x1 %>% mutate(CO = 1) %>%
filter(Origin %in% c("Syria", "Jordan", "Turkey", "Lebanon", "Israel"))
# Plot A) a vs DTF
find_hull <- function(df) df[chull(df[, "DTF_mean"], df[, "a_mean"]), ]
polys <- plyr::ddply(x2, "CO", find_hull)
mp1 <- ggplot(x1, aes(x = DTF_mean, y = a_mean * 10000)) +
  geom_polygon(data = polys, fill = NA, color = "black") +
  geom_point() + geom_text_repel(aes(label = Origin)) +
  theme_AGL +
  labs(title = "A",
       y = expression(paste("Genotype constant (", italic(a), " x 10000)")),
       x = "Days from sowing to flower (Sutherland, Canada 2017)")
# Plot B) b vs c
find_hull <- function(df) df[chull(df[, "c_mean"], df[, "b_mean"]), ]
polys <- plyr::ddply(x2, "CO", find_hull)
mp2 <- ggplot(x1, aes(x = c_mean * 10000, y = b_mean * 10000)) +
  geom_polygon(data = polys, fill = NA, color = "black") +
  geom_point() + geom_text_repel(aes(label = Origin)) +
  theme_AGL +
  labs(title = "B",
       y = expression(paste("Temperature response (", italic(b), " x 10000))),
       x = expression(paste("Photoperiod response (", italic(c), " x 10000)")))
# Append A) and B)
mp <- ggarrange(mp1, mp2, ncol = 1, nrow = 2)
ggsave("Figure9_OriginCoefficients.png", mp, width = 8, height = 8)
ggsave("Temp/Temp_F9_1.png", mp1, width = 8, height = 4)
ggsave("Temp/Temp_F9_2.png", mp2, width = 8, height = 4)

```



Supplemental Figure 8 Adapted

```

# Create plotting function
gg_adapted <- function(myTitle, myExpt, sdMult = 2, myOrigin, myOrigins = myOrigin,
                        predictDTF = F, meanT, meanP) {
  # Prep data
  xx <- read.csv("data/model_18_coefs.csv") %>%
    left_join(select(ldp, Entry, Origin), by = "Entry")
  if(predictDTF == F) {
    xx <- xx %>%
      left_join(dd %>% filter(Expt == myExpt) %>%
                  select(Entry, DTF), by = "Entry") %>%
      mutate(Origin = as.character(Origin),
             Origin = ifelse(Origin %in% myOrigins, Origin, "Other"),
             DTF = ifelse(is.nan(DTF), max(.\$DTF, na.rm = T), DTF))
  }
  if(predictDTF == T) {
    xx <- xx %>%
      mutate(DTF = 1 / (a + b*meanT + c*meanP),
             #Origin = as.character(Origin),
             Origin = ifelse(Origin %in% myOrigins, as.character(Origin), "Other"))
  }
  xx <- xx %>% mutate(a = a*10000, b=b*10000, c=c*10000)
  yy <- xx %>%
    group_by(Origin) %>%
    summarise_at(vars(DTF, a, b, c), funs(mean, sd), na.rm = T) %>%
    rename(c=c_mean, b=b_mean, DTF=DTF_mean) %>%
    filter(Origin == myOrigin)
  myLT <- paste("Mean +/-", sdMult, "* sd")
  #
  OriginLevels <- unique(c("Other", "\\"Adapted\\\"", myOrigin, myOrigins))
  myYmin1 <- yy$c - sdMult * yy$c_sd
  myYmax1 <- yy$c + sdMult * yy$c_sd
  myXmin1 <- yy$DTF - sdMult * yy$DTF_sd
  myXmax1 <- yy$DTF + sdMult * yy$DTF_sd
  x1 <- xx %>%
    mutate(Origin = ifelse(DTF > myXmin1 & DTF < myXmax1 &
                           !Origin %in% myOrigins, "\\"Adapted\\\"", Origin),
          Origin = factor(Origin, levels = OriginLevels))
  # Plot
  mp1 <- ggplot(yy, aes(x = DTF, y = c)) +
    geom_point(data = x1, aes(color = Origin), alpha = 0.5) +
    geom_point(aes(pch = Origin), size = 3) +
    geom_errorbar(aes(ymin = myYmin1, ymax = myYmax1)) +
    geom_errorbarh(aes(xmin = myXmin1, xmax = myXmax1)) +
    theme_AGL +
    scale_shape_manual(name = myLT, values = 18) +
    scale_color_manual(values = colors[c(7,8,1,3,5,6,2)]) +
    labs(title = myTitle,
         y = expression(paste("Photoperiod response (", italic(c), " x 10000)")),
         x = "Days from sowing to flower (days)") +
    guides(shape = guide_legend(order = 1), colour = guide_legend(order = 2))
  # Prep data

```

```

myYmin2 <- yy$b - sdMult * yy$b_sd
myYmax2 <- yy$b + sdMult * yy$b_sd
myXmin2 <- yy$DTF - sdMult * yy$DTF_sd
myXmax2 <- yy$DTF + sdMult * yy$DTF_sd
x2 <- xx %>%
  mutate(Origin = ifelse(DTF > myXmin2 & DTF < myXmax2 &
                        !Origin %in% myOrigins, "\\"Adapted\\\"", Origin),
        Origin = factor(Origin, levels = OriginLevels))
# Plot
mp2 <- ggplot(yy, aes(x = DTF, y = b)) +
  geom_point(data = x2, aes(color = Origin), alpha = 0.5) +
  geom_point(aes(pch = Origin), size = 3) +
  geom_errorbar(aes(ymin = myYmin2, ymax = myYmax2)) +
  geom_errorbarh(aes(xmin = myXmin2, xmax = myXmax2)) +
  theme_AGL +
  scale_shape_manual(name = myLT, values = 18) +
  scale_color_manual(values = colors[c(7,8,1,3,5,6,2)]) +
  labs(y = expression(paste("Temperature response (", italic(b), " x 10000)")),
       x = "Days from sowing to flower (days)") +
  guides(shape = guide_legend(order = 1), colour = guide_legend(order = 2))
# Prep data
myDTFmin <- yy$DTF - sdMult * yy$DTF_sd
myDTFmax <- yy$DTF + sdMult * yy$DTF_sd
myYmin3 <- yy$c - sdMult * yy$c_sd
myYmax3 <- yy$c + sdMult * yy$c_sd
myXmin3 <- yy$b - sdMult * yy$b_sd
myXmax3 <- yy$b + sdMult * yy$b_sd
x3 <- xx %>%
  mutate(Origin = ifelse(DTF > myDTFmin & DTF < myDTFmax &
                        !Origin %in% myOrigins, "\\"Adapted\\\"", Origin),
        Origin = factor(Origin, levels = OriginLevels))
# Plot
mp3 <- ggplot(yy, aes(x = b, y = c)) +
  geom_point(data = x3, aes(color = Origin), alpha = 0.5) +
  geom_point(aes(pch = Origin), size = 3) +
  geom_errorbar(aes(ymin = myYmin3, ymax = myYmax3)) +
  geom_errorbarh(aes(xmin = myXmin3, xmax = myXmax3)) +
  theme_AGL +
  scale_shape_manual(name = myLT, values = 18) +
  scale_color_manual(values = colors[c(7,8,1,3,5,6,2)]) +
  labs(y = expression(paste("Photoperiod response (", italic(c), " x 10000)")),
       x = expression(paste("Temperature response (", italic(b), " x 10000)"))) +
  guides(shape = guide_legend(order = 1), colour = guide_legend(order = 2))
# Append Plots
mp <- ggarrange(mp1, mp2, mp3, ncol = 3, nrow = 1, align = "hv",
                 common.legend = T, legend = "right")
mp
}
# Plot
mp1 <- gg_adapted("A) Sutherland, Canada 2017", "Sutherland, Canada 2017", 2, "Canada")
mp2 <- gg_adapted("B) Jessore, Bangladesh 2017", "Jessore, Bangladesh 2017", 1,
                  "Bangladesh", c("India", "Bangladesh", "ICARDA"))
mp3 <- gg_adapted("C) Metaponto, Italy 2017", "Metaponto, Italy 2017", 2,

```

```

    "Italy", c("Italy", "Spain", "Greece", "Macedonia", "Serbia"))
mp4 <- gg_adapted(expression(paste("D", italic("T"), " = 13 ", degree, "C, ",
                                   italic("P"), " = 11.5 h"))), sdMult = 1,
                     myOrigin = "Pakistan", predictDTF = T, meanT = 13, meanP = 11.5)
mp <- ggarrange(mp1, mp2, mp3, mp4, nrow=4, ncol=1, align = "hv")
ggsave("SupplementalFigure8_Adapted.png", mp, width = 11, height = 14)
ggsave("Temp/Temp_SF8_1.png", mp1, width = 11, height = 3.75)
ggsave("Temp/Temp_SF8_2.png", mp2, width = 11, height = 3.75)
ggsave("Temp/Temp_SF8_3.png", mp3, width = 11, height = 3.75)
ggsave("Temp/Temp_SF8_4.png", mp4, width = 11, height = 3.75)

```

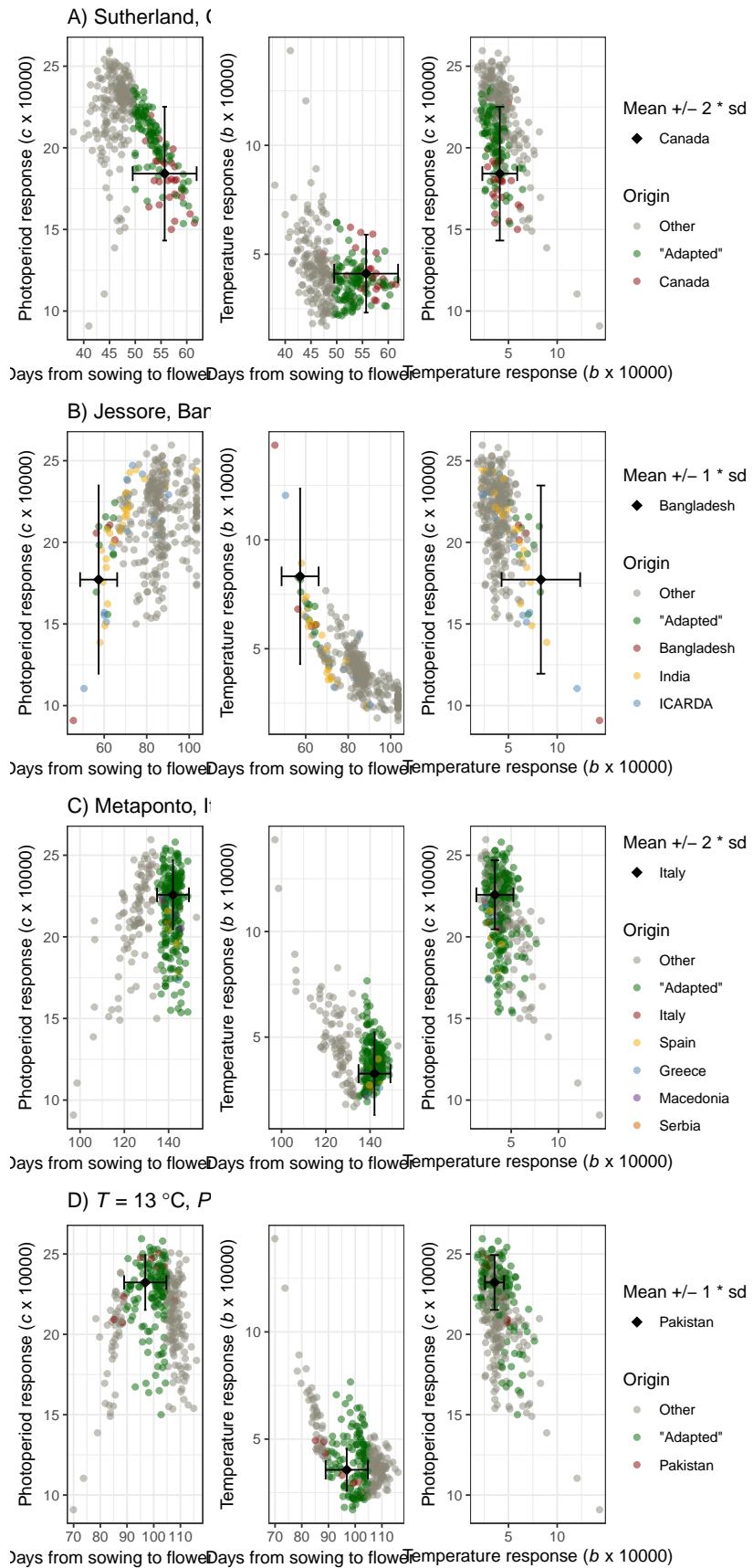


Figure 10 AMMI

```
# Run AMMI
model.dtf <- with(rr,  AMMI(ExptShort, Entry, Rep, DTF2_scaled, console = F))
# Save AMMI results
saveRDS(model.dtf, file = "data/AMMI.rds")

# Prep data
xx <- readRDS("data/AMMI.rds")
perc <- xx[[3]]
xx <- xx$biplot %>% rownames_to_column("Id")
ets <- select(ff, ExptShort, MacroEnv) %>%
  filter(ExptShort %in% xx$Id[xx$type == "ENV"]) %>%
  mutate(ExptShort = as.character(ExptShort)) %>%
  rename(Id = ExptShort)
gen <- xx %>% filter(type == "GEN")
env <- xx %>% filter(type == "ENV") %>%
  left_join(ets, by = "Id") %>%
  mutate(Id = factor(Id, levels = names_ExptShort))
xmed<-mean(env[, "PC1"])
ymed<-mean(env[, "PC2"])
yy <- c("darkgreen", "darkred", "steelblue")
# Plot AMMI
mp <- ggplot(gen) +
  theme_AGL +
  theme(legend.position = "none",
        strip.background = element_rect(fill = "White"),
        panel.grid.major = element_line(colour = "black", size = 0.75)) +
  scale_x_continuous(breaks = 0) +
  scale_y_continuous(breaks = 0) +
  geom_point(aes(x = PC1, y = PC2),
             color = "Black", fill = "grey80", pch = 21) +
  geom_segment(data = env, lwd = 1, arrow = arrow(length = unit(0.30, "cm")),
               aes(x = xmed, y = ymed, xend = PC1, yend = PC2, group = Id)) +
  ggrepel::geom_label_repel(data = env, fill = "grey90", aes(x = PC1, y = PC2, label = Id),
                            #          Ba   In   It   Mo   Ne   Ro   Sp   Su   Us
                            nudge_y = c(0,.5, 1,.5, .3,.3, -.2,0, .5,0, .5, .5, .2, 0, -.5,-.5,1, -.5),
                            nudge_x = c(.3,0, 0, 0, 0,0, .4,.4, .4,.3, 0,-.1, -.2,.3, 0, 0,0, 0),
                            segment.colour = alpha("Black", 0.5)) +
  labs(x = paste0("IPCA1 (", perc[1,1], "%)"),
        y = paste0("IPCA2 (", perc[2,1], "%)"))
ggsave("Figure10_AMMI.png", mp, width = 6, height = 4)
```

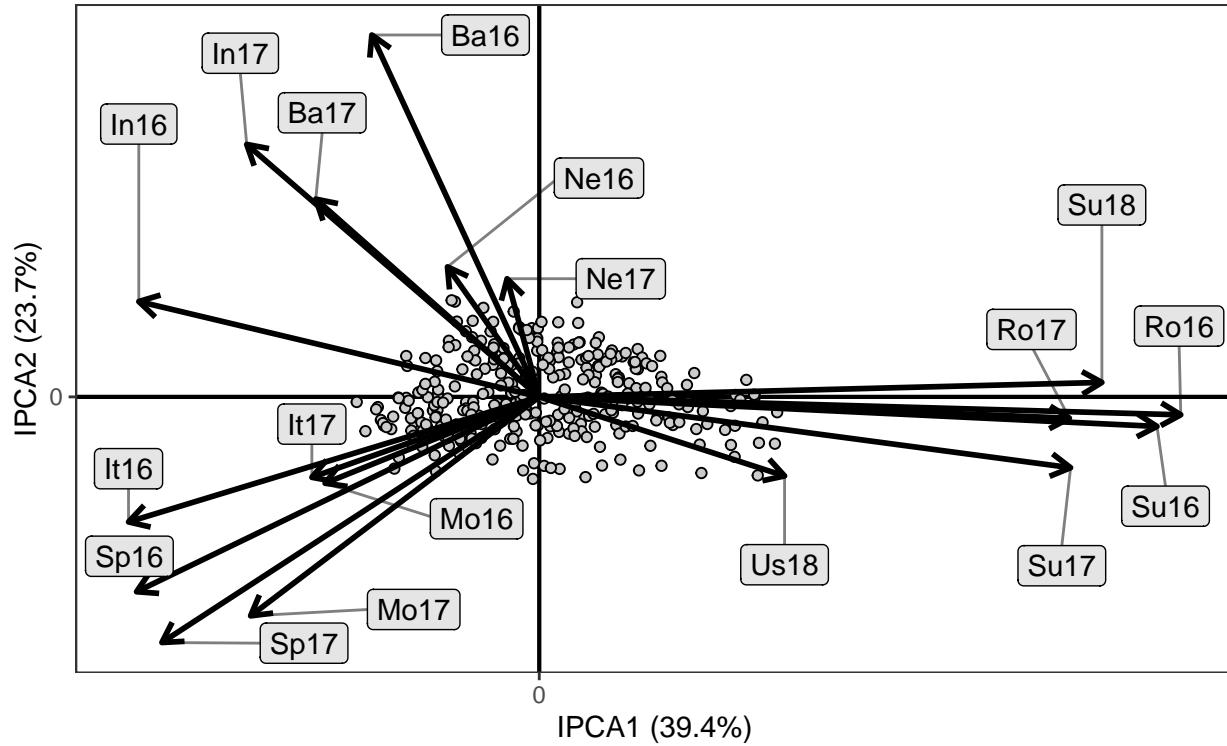


Figure 11 PCA

```

# Prep data
xx <- dd %>% select(Entry, Expt, DTF2_scaled) %>%
  spread(Expt, DTF2_scaled)
xx <- xx %>% column_to_rownames("Entry") %>% as.matrix()
# PCA
mypca <- PCA(xx, ncp = 10, graph = F)
# Heirarcical clustering
mypcaH <- HCPC(mypca, nb.clust = 8, graph = F)
perc <- round(mypca[[1]][,2], 1)
x1 <- mypcaH[[4]]$X %>%
  rownames_to_column("Entry") %>%
  mutate(Entry = as.numeric(Entry)) %>%
  rename(PC1=Dim.1, PC2=Dim.2, PC3=Dim.3, PC4=Dim.4, PC5=Dim.5,
         PC6=Dim.6, PC7=Dim.7, PC8=Dim.8, PC9=Dim.9, PC10=Dim.10,
         Cluster=clust) %>%
  left_join(select(ldp, Entry, Name, Origin), by = "Entry") %>%
  left_join(select(ct, Origin=Country, Region), by = "Origin") %>%
  select(Entry, Name, Origin, Region, everything())
write.csv(x1, "data/data_PCA_Results.csv", row.names = F)
#
x2 <- dd %>% left_join(select(x1, Entry, Cluster), by = "Entry") %>%
  group_by(Expt, ExptShort, Cluster) %>%

```

```

summarise(mean = mean(DTF2_scaled, na.rm = T), sd = sd(DTF2_scaled, na.rm = T)) %>%
ungroup() %>%
mutate(ClusterNum = plyr::mapvalues(Cluster, as.character(1:8), summary(x1$Cluster)))
x3 <- x1 %>% count(Cluster) %>%
mutate(Cluster = factor(Cluster, levels = rev(levels(Cluster))), y = n/2)
for(i in 2:nrow(x3)) { x3$y[i] <- sum(x3$n[1:(i-1)]) + (x3$n[i]/2) }
# Plot A) PCA 1v2
find_hull <- function(df) df[chull(df[, "PC1"], df[, "PC2"]), ]
polys <- plyr::ddply(x1, "Cluster", find_hull) %>% mutate(Cluster = factor(Cluster))
mp1.1 <- ggplot(x1) +
geom_polygon(data = polys, alpha = 0.15, aes(x = PC1, y = PC2, fill = Cluster)) +
geom_point(aes(x = PC1, y = PC2, colour = Cluster)) +
scale_fill_manual(values = colors) +
scale_color_manual(values = colors) +
theme_classic() + theme(legend.position = "none") +
labs(x = paste0("PC1 (", perc[1], "%)"),
y = paste0("PC2 (", perc[2], "%)"))
# Plot A) PCA 1v3
find_hull <- function(df) df[chull(df[, "PC1"], df[, "PC3"]), ]
polys <- plyr::ddply(x1, "Cluster", find_hull) %>% mutate(Cluster = factor(Cluster))
mp1.2 <- ggplot(x1) +
geom_polygon(data = polys, alpha = 0.15, aes(x = PC1, y = PC3, fill = Cluster)) +
geom_point(aes(x = PC1, y = PC3, colour = Cluster)) +
scale_fill_manual(values = colors) +
scale_color_manual(values = colors) +
theme_classic() + theme(legend.position = "none") +
labs(x = paste0("PC1 (", perc[1], "%)"),
y = paste0("PC3 (", perc[3], "%)"))
# Plot A) PCA 2v3
find_hull <- function(df) df[chull(df[, "PC2"], df[, "PC3"]), ]
polys <- plyr::ddply(x1, "Cluster", find_hull) %>% mutate(Cluster = factor(Cluster))
mp1.3 <- ggplot(x1) +
geom_polygon(data = polys, alpha = 0.15, aes(x = PC2, y = PC3, fill = Cluster)) +
geom_point(aes(x = PC2, y = PC3, colour = Cluster)) +
scale_fill_manual(values = colors) +
scale_color_manual(values = colors) +
theme_classic() + theme(legend.position = "none") +
labs(x = paste0("PC2 (", perc[2], "%)"),
y = paste0("PC3 (", perc[3], "%)"))
# Plot B) DTF
mp2 <- ggplot(x2, aes(x = ExptShort, y = mean, group = Cluster)) +
geom_ribbon(aes(ymin = mean - sd, ymax = mean + sd, fill = Cluster),
alpha = 0.1, color = NA) +
geom_point(aes(color = Cluster)) +
geom_line(aes(color = Cluster), size = 1) +
scale_color_manual(values = colors) +
scale_fill_manual(values = colors) +
coord_cartesian(ylim = c(0.95, 5.05), expand = F) +
theme_minimal() +
theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
legend.position = "none",
strip.placement = "outside",
axis.line = element_line(), axis.ticks = element_line())

```

```

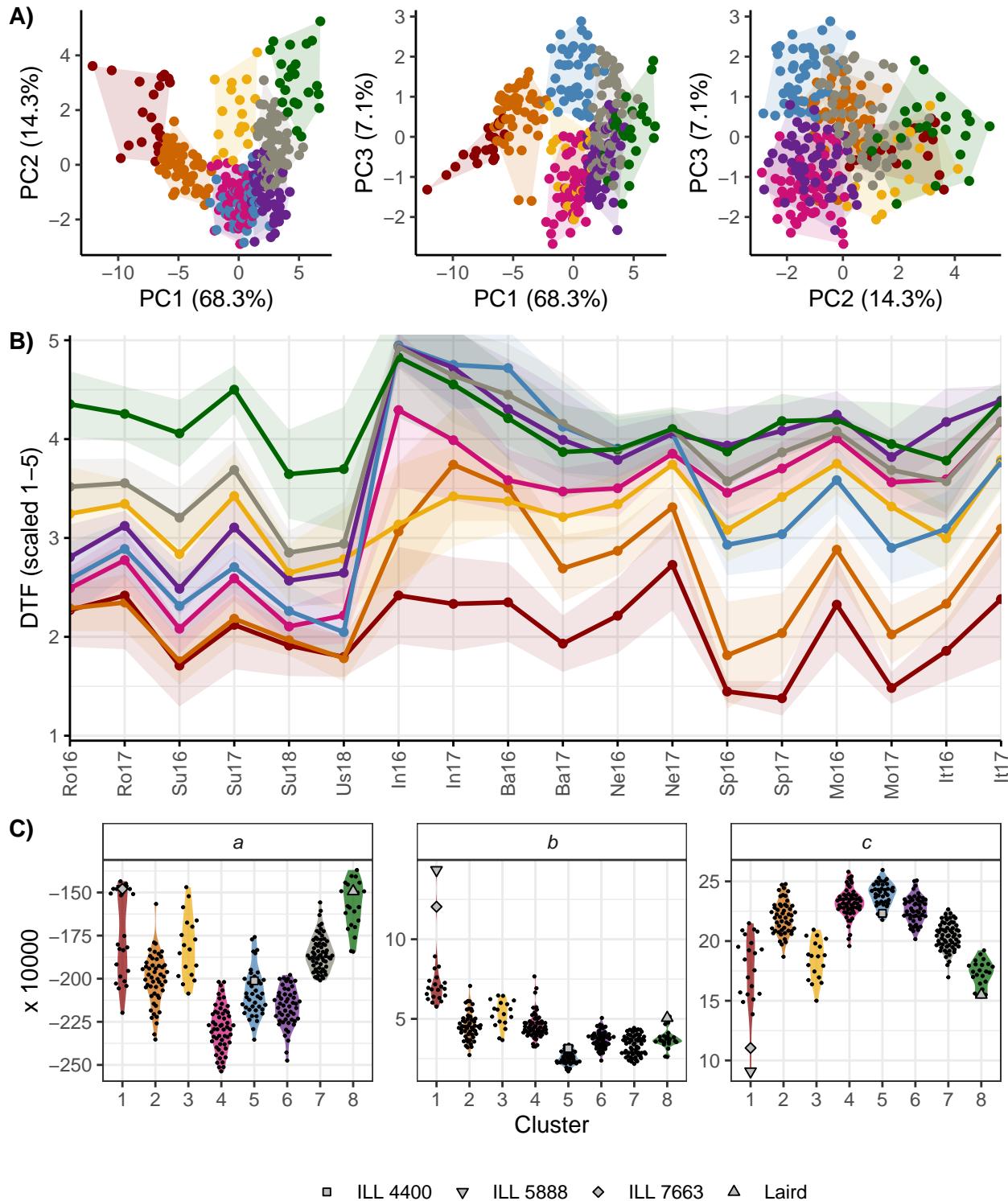
  labs(y = "DTF (scaled 1-5)", x = NULL)
# Prep data
xx <- read.csv("data/model_18_coefs.csv") %>%
  left_join(select(x1,Entry,Cluster), by = "Entry") %>%
  select(Entry, Name, Cluster, a, b, c) %>%
  gather(Trait, Value, 4:ncol(.))
genos <- c("ILL 7663 AGL", "ILL 5888 AGL", "ILL 4400 AGL", "Laird AGL")
x4 <- xx %>% filter(Name %in% genos) %>%
  mutate(Name = factor(Name, levels = genos),
        Name = gsub(" AGL", "", Name))
# Plot a, b and c
mp3 <- ggplot(xx, aes(x = Cluster, y = Value * 10000) ) +
  geom_violin(aes(fill = Cluster), color = NA, alpha = 0.7) +
  geom_quasirandom(size = 0.25) +
  geom_point(data = x4, fill = "grey", aes(shape = Name)) +
  facet_wrap(Trait~., nrow = 1, scales = "free") +
  theme_AGL +
  theme(legend.position = "bottom",
        strip.text = element_text(face = "italic")) +
  scale_shape_manual(name = NULL, values = c(22,25,23,24)) +
  scale_fill_manual(name = NULL, values = colors) +
  guides(fill = F) +
  labs(y = "x 10000", x = "Cluster")
# Append
mp1 <- ggarrange(mp1.1, mp1.2, mp1.3,
                  nrow = 1, ncol = 3, hjust = 0)
mp <- ggarrange(mp1, mp2, mp3, ncol = 1, nrow = 3,
                 heights = c(1, 1.5, 1.3), hjust = 0,
                 labels = c("A)","B)","C)"),
                 font.label = list(size = 11))
ggsave("Figure11_PCA.png", mp, width = 8, height = 8)
ggsave("Temp/Temp_F11_1.png", mp1, width = 8, height = 1*8/3.8)
ggsave("Temp/Temp_F11_2.png", mp2, width = 8, height = 1.5*8/3.8)
ggsave("Temp/Temp_F11_3.png", mp3, width = 8, height = 1.3*8/3.8)
#
summary(x1$Cluster)

```

```

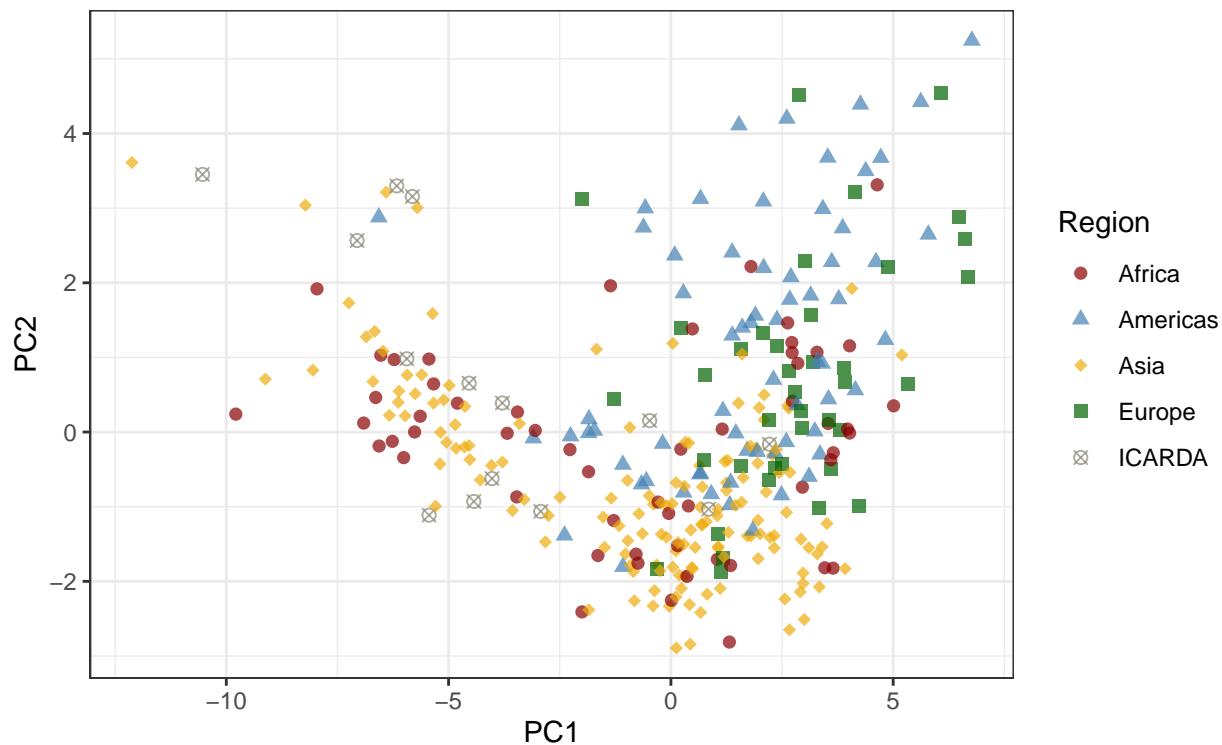
  1   2   3   4   5   6   7   8
22  51  18  56  41  51  62  23

```



Additional Figure 10 PCA

```
# Prep data
xx <- read.csv("data/data_PCA_Results.csv") %>%
  mutate(Region = ifelse(Origin == "ICARDA", "ICARDA", as.character(Region)))
# Plot PCA
mp <- ggplot(xx, aes(x = PC1, y = PC2, color = Region, shape = Region)) +
  geom_point(alpha = 0.7, size = 2) +
  scale_color_manual(values = colors[c(1,5,3,8,7)]) +
  scale_shape_manual(values = c(16,17,18,15,13)) +
  theme_AGL
ggsave("Additional/AdditionalFigure10_PCA.png", mp, width = 6, height = 4)
```



Supplemental Figure 9 Cluster Origins

```
# Prep data
x1 <- read.csv("data/data_PCA_Results.csv") %>% mutate(Cluster = factor(Cluster))
xx <- ldp %>% left_join(select(x1, Entry, Cluster), by = "Entry") %>%
  mutate(test1 = factor(paste(Origin, Cluster)))
x1 <- xx %>% filter(Origin != "ICARDA") %>%
  group_by(Origin, Cluster) %>% summarise(Count = n()) %>%
  spread(Cluster, Count) %>%
```

```

left_join(select(ct, Origin=Country, Lat, Lon), by = "Origin") %>%
ungroup() %>% as.data.frame()
x1[is.na(x1)] <- 0
# Plot A) Bars
mp <- ggplot(xx, aes(x = Origin, fill = Cluster)) +
  geom_bar(stat = "count") +
  facet_grid(.~Cluster, scales = "free", space = "free") +
  scale_fill_manual(values = colors) +
  theme_AGL +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        legend.position = "none") +
  labs(x = NULL)
ggsave("Temp/Temp_S9_1.png", width = 16, height = 4)
# Plot B) Map Pies
invisible(png("Temp/Temp_S9_2.png", width = 1200, height = 550, res = 150))
par(mai = c(0,0,0,0), xaxs = "i", yaxs = "i")
mapPies(dF = x1, nameX = "Lon", nameY = "Lat",
         nameZs = c("1","2","3","4","5","6","7","8"), symbolSize = 1,
         zColours = colors,
         xlim = c(-140,110), ylim = c(0,20),
         oceanCol = "grey90", landCol = "white", borderCol = "black")

```

```

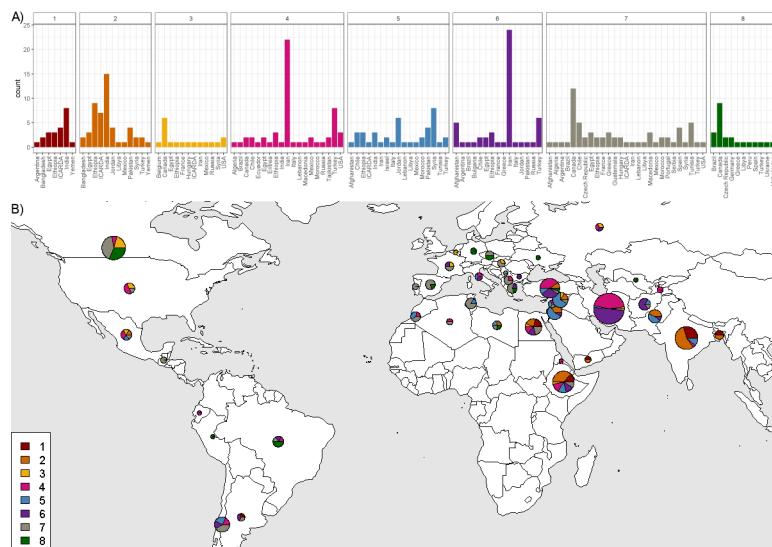
symbolMaxSize= 5  maxSumValues= 49  symbolScale= 0.7142857
List of 2
$ x: num [1:100] -125 -125 -125 -125 -125 ...
$ y: num [1:100] 57.3 57.6 57.9 58.2 58.5 ...

```

```

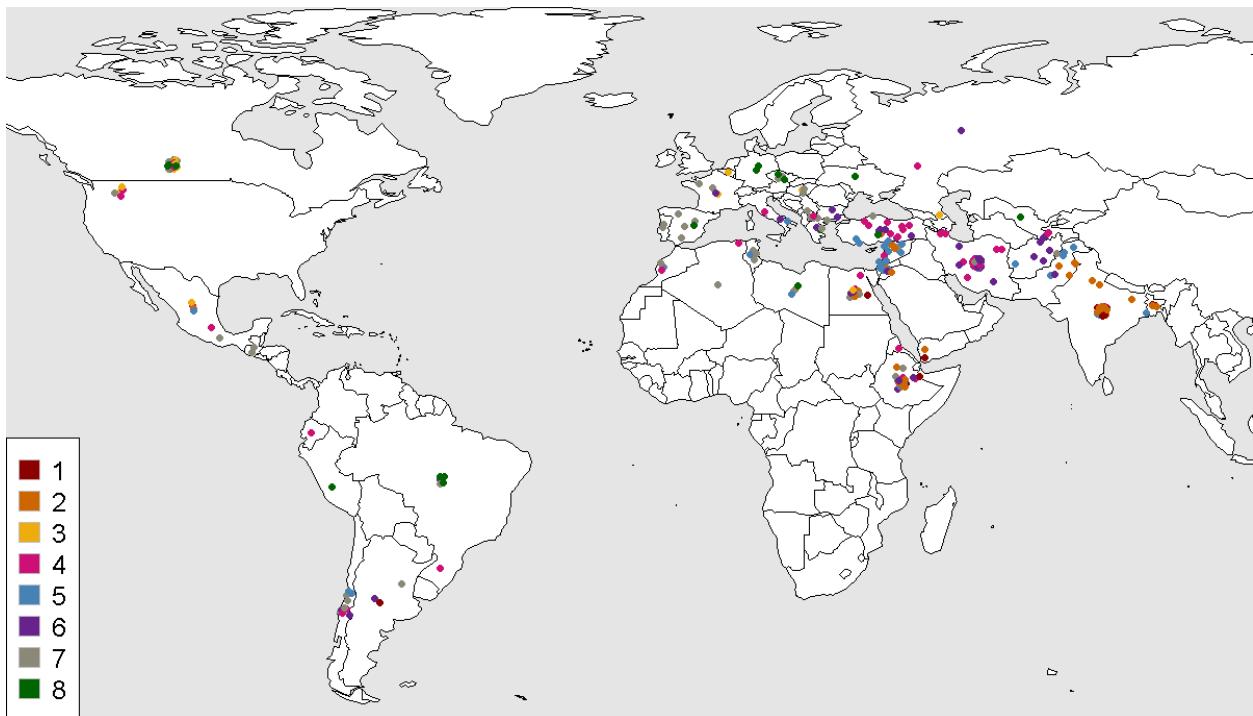
invisible(dev.off())
# Append A) and B)
im1 <- image_read("Temp/Temp_S9_1.png") %>% image_scale("1200x") %>%
  image_annotate("A)", size = 20)
im2 <- image_read("Temp/Temp_S9_2.png") %>%
  image_annotate("B)", size = 20)
im <- image_append(c(im1,im2), stack = T)
image_write(im, "SupplementalFigure9_ClusterOrigins1.png")

```



Additional Figure 11 LDP Origins By Cluster

```
# Prep data
x1 <- read.csv("data/data_PCA_Results.csv") %>% mutate(Cluster = factor(Cluster))
xx <- ldp %>% select(Entry, Name, Lat, Lon) %>% left_join(x1, by = "Entry") %>%
  left_join(select(ct, Origin=Country, cLat=Lat, cLon=Lon), by = "Origin") %>%
  mutate(Lat = ifelse(is.na(Lat), cLat, Lat),
         Lon = ifelse(is.na(Lon), cLon, Lon),
         Lat = ifelse(duplicated(Lat), jitter(Lat, 1, 1), Lat),
         Lon = ifelse(duplicated(Lon), jitter(Lon, 1, 1), Lon), Size = 1)
# Plot Map
invisible(png("Additional/AdditionalFigure11_LDPOriginsByCluster.png", width = 1200, height = 685, res ...
par(mai = c(0,0,0,0), xaxs = "i", yaxs = "i")
mapBubbles(dF = xx, nameX = "Lon", nameY = "Lat", nameZColour = "Cluster",
           nameZSize = "Size", symbolSize = 0.25, pch = 20,
           colourPalette = colors[1:8],
           xlim = c(-140,110), ylim = c(5,20), addLegend = F, colourLegendTitle = NULL,
           oceanCol = "grey90", landCol = "white", borderCol = "black")
invisible(dev.off())
```



Additional Figure 12 DTF By Cluster

```
# Prep data
x1 <- read.csv("data/data_PCA_Results.csv") %>% mutate(Cluster = factor(Cluster))
yy <- c("India", "Bangladesh", "Ethiopia", "Pakistan", "Jordan",
       "Iran", "Turkey", "Syria", "Chile", "Spain", "Czech Republic", "Canada" )
xx <- dd %>% left_join(ldp, by = "Entry") %>%
  filter(ExptShort %in% c("Ro17", "It17", "Ba17"), Origin != "Unknown") %>%
  left_join(select(x1, Entry, Cluster), by = "Entry") %>%
  mutate(Origin = factor(Origin, levels = unique(Origin)[rev(order(unique(Origin)))])) %>%
  filter(Origin %in% yy) %>%
  mutate(Origin = factor(Origin, levels = yy))
# Plot
mp <- ggplot(xx, aes(y = DTF2, x = Origin)) +
  geom_quasirandom(aes(color = Cluster)) +
  facet_grid(Expt~., scales = "free_y") +
  scale_color_manual(values = colors) +
  theme_AGL + theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5)) +
  labs(y = "Days from sowing to flower", x = NULL)
ggsave("Additional/AdditionalFigure12_DTFByCluster.png", mp, width = 10, height = 8)
```

