Supplementary Methods 1

Chris Terry

Contents

1 Summary

This document is a knit `.rmd` file that works through the steps of the reanalysis of *Small rainfall changes drive substantial changes in plant coexistence* by Van Dyke et al 2022 (Nature), to support a 'Matters Arising' response. Original data from the authors is available at https://doi.org/10.5281/zenodo.7083314

Section 4 details the methods used to conduct a model comparison analysis to examine the level of statistical support for the treatment impacting the key model parameters and subsequent key coexistence quantities.

Section 5 propogates forward the authors original full bootstrap through to the final assessment of whether the predicted coexistence outcome species pair differs between treatments.

2 Loading and looking at data

```
library(tidyverse)
set.seed(1)

Path_to_Orig <- '../DroughtCompUncertainty/water_competition-2.0/Sedgwick_public/'

nls_boot_pairs<-read.csv(paste0(Path_to_Orig,"output/nls_boot_pairs_1000_full_model.csv"))
seed_data <- read.csv(paste0(Path_to_Orig,"data/drought_seed_production_data.csv"))

seed_data$Tr <- ifelse(seed_data$treat == "W", 1, 2)
seed_data <- seed_data %>%
  mutate( background = ifelse(is.na(background),
                              focal, background)) # where 0 background, naming after focal

seed_data$N_acwr <- ifelse(seed_data$background == "ACWR", seed_data$num_comp, 0)
seed_data$N_femi <- ifelse(seed_data$background == "FEMI", seed_data$num_comp, 0)
seed_data$N_homu <- ifelse(seed_data$background == "HOMU", seed_data$num_comp, 0)
seed_data$N_pler <- ifelse(seed_data$background == "PLER", seed_data$num_comp, 0)
seed_data$N_saco <- ifelse(seed_data$background == "SACO", seed_data$num_comp, 0)
seed_data$N_urli <- ifelse(seed_data$background == "URLI", seed_data$num_comp, 0)
```

2.1 How much data?

```
nrow(seed_data)
```

```
## [1] 1677
```

```
count(seed_data, background, focal, treat) %>% arrange(n) %>% head()
```

```
##   background focal treat  n
## 1       FEMI  URLI     W  9
## 2       URLI  SACO     W 10
## 3       PLER  HOMU     W 11
## 4       URLI  HOMU     D 11
## 5       URLI  PLER     W 11
## 6       ACWR  HOMU     D 12
```

2.2 How much uncertainty in types of parameters

```
### no lambda sd saved, so will need to reconstruct, following original code

spp_treat_boot_combos <-read.csv(paste0(Path_to_Orig, "output/final_output_nls_boot_1000.csv"))

#Data frame with medians and sds for parameters----
spp_list <- sort(na.omit( unique(seed_data$focal)))
treat_list <- sort( na.omit( unique(seed_data$Tr)))
spp_treat_combos <- expand.grid(species = spp_list, treatment = treat_list)
comp_labels <- sort( na.omit( unique(seed_data$background) ))
```

```
spp_treat_boot_combos$sp_pair <- paste(spp_treat_boot_combos$focal,
                                       spp_treat_boot_combos$competitor, sep = "_")

spp_treat_comp_combos <- expand.grid(focal = spp_list,
                                     competitor = comp_labels,
                                     treatment = treat_list)

boot_NEW <- spp_treat_comp_combos %>%  mutate(alpha = 0, alpha_sd = 0,
         lambda = 0, lambda_sd =0)
boot_NEW$sp_pair <- paste(boot_NEW$focal, boot_NEW$competitor, sep = "_")

for( i in 1:nrow(spp_treat_comp_combos)) {
  sp1 <- spp_treat_comp_combos[i, "focal"] %>% unlist
  sp2 <- spp_treat_comp_combos[i, "competitor"] %>% unlist
  treatt <- spp_treat_comp_combos[i , "treatment"] %>% unlist
  boot_NEW[i, "alpha"] <- median(with(spp_treat_boot_combos,
                                      alpha[treatment == treatt &
                                              focal == sp1 &
                                              competitor == sp2
                                      ]), na.rm=TRUE)
  boot_NEW[i, "alpha_sd"] <- sd(with(spp_treat_boot_combos,
                                     alpha[treatment == treatt &
                                             focal == sp1 &
                                             competitor == sp2
                                     ]), na.rm=TRUE)
  boot_NEW[i, "lambda"] <- median(with(spp_treat_boot_combos,
                                       lambda[focal == sp1 & competitor == sp2 &
                                                treatment == treatt]), na.rm=TRUE)
  boot_NEW[i, "lambda_sd"] <- sd(with(spp_treat_boot_combos,
                                      lambda[focal == sp1 & competitor == sp2 &
                                               treatment == treatt]), na.rm=TRUE)
}

boot_NEW %>%
  filter(focal==competitor ) %>%
  mutate(LamCV= lambda_sd/lambda) %>%
  summarise(mean(LamCV))
```

```
##   mean(LamCV)
## 1   0.3195691
```

```
boot_NEW %>%
  mutate(AlpCV= alpha_sd/alpha)%>%
  summarise(mean(AlpCV))
```

```
##   mean(AlpCV)
## 1   0.7400443
```

3 Plotting raw data with model best-fit

```
spp_list <- sort(na.omit( unique(seed_data$focal)))
```

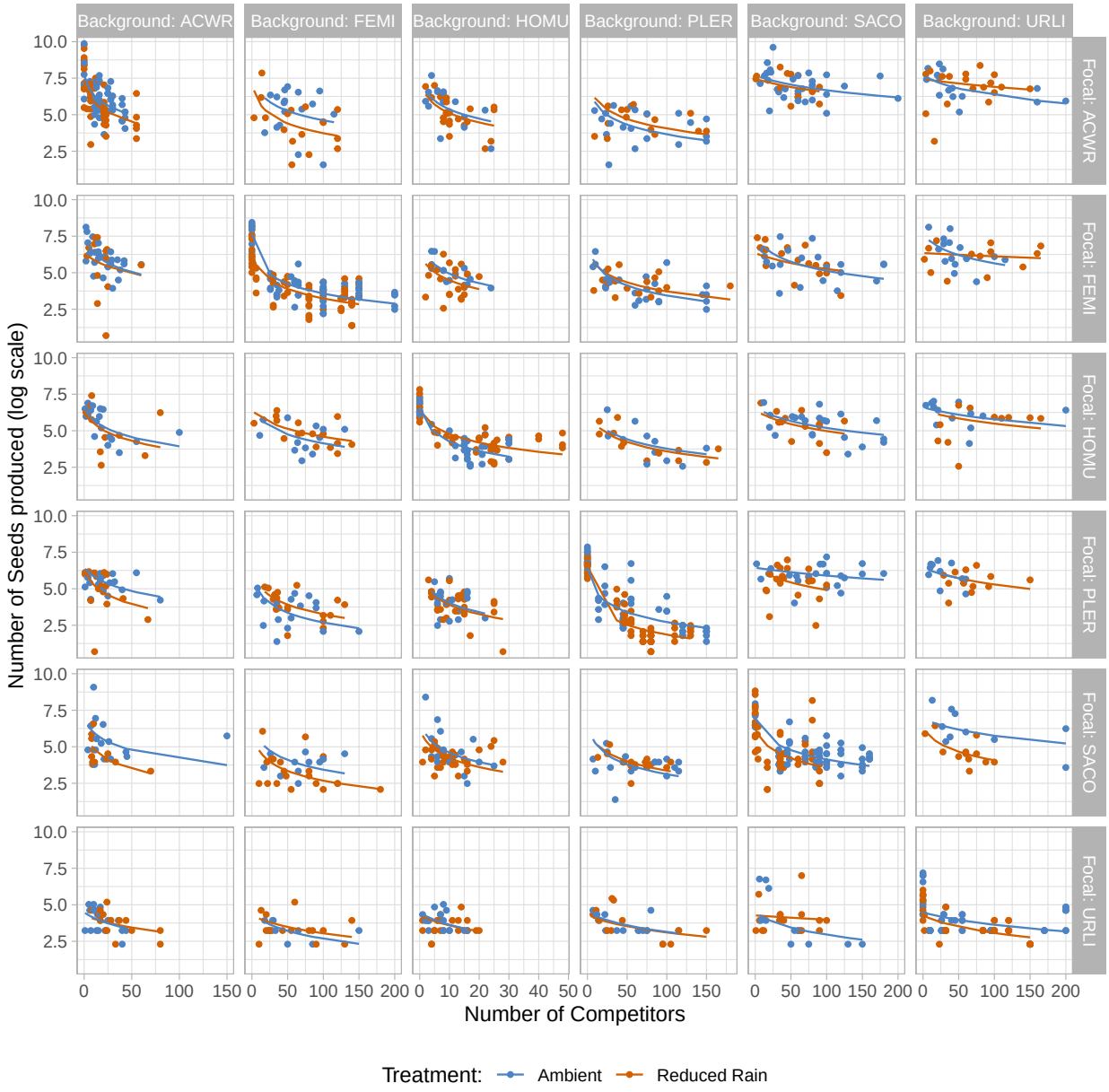
```
set.seed(3)

AddPreds<- function(focalsp){
  focal_data <- filter(seed_data, focal == focalsp)

  fit<- nls(log(num_seeds)~log(lambda[Tr]/(1+a_ACWR[Tr]*N_acwr+a_FEMI[Tr]*N_femi+
                                             a_HOMU[Tr]*N_homu+a_PLER[Tr]*N_pler+
                                             a_SACO[Tr]*N_saco+a_URLI[Tr]*N_urli)),
            data=focal_data,
            start=list('lambda'= c(100,100),
                       a_ACWR=c(0.1, 0.1), a_FEMI=c(0.1, 0.1),
                       a_HOMU=c(0.1, 0.1), a_PLER=c(0.1, 0.1),
                       a_SACO=c(0.1, 0.1),a_URLI=c(0.1, 0.1)),
            lower = c( 1, 1, rep(.001, 12) ),
            upper = c(10000, 10000, rep(2, 12)),
            control = list(maxiter = 100000),
            algorithm = 'port')
  focal_data$Prediction <- predict(fit)
  return(focal_data)
}

data_predict <- map_df( spp_list,AddPreds)
```

```
data_predict %>%
  rename(Focal = focal, Background = background) %>%
  arrange(num_comp)%>%
  ggplot(aes( x =num_comp, col = treat)) +
  geom_point(aes(y = log(num_seeds)), size = 1)+
  facet_grid(Focal~Background, scales = 'free_x',
             labeller= label_both)+
  geom_line(aes(y = Prediction))+
  theme_light()+
  xlab('Number of Competitors')+
  ylab('Number of Seeds produced (log scale)')+
  scale_color_manual(values=c('W' ="#4E84C4", 'D' = "#D16103"),
                     name = "Treatment:",
                     labels = c("Ambient", "Reduced Rain"))+
  theme(legend.position = 'bottom')
```




```
ggsave('ExFig1_RawPredict.png', height = 8, width = 8, dpi = 200)
```

4 Fitting set of models with different levels of parameterisation

```
## Mapping over all 6 focal species
FindFourAICs <- function(focalsp){

  focal_data <- filter(seed_data, focal == focalsp)

  fit_test<- nls(log(num_seeds)~log(lambda[Tr]/(1+a_ACWR[Tr]*N_acwr+a_FEMI[Tr]*N_femi+
                                                  a_HOMU[Tr]*N_homu+a_PLER[Tr]*N_pler+
```

```
                                                a_SACO[Tr]*N_saco+a_URLI[Tr]*N_urli)),
               data=focal_data,
               start=list('lambda'= c(100,100),
                          a_ACWR=c(0.1, 0.1), a_FEMI=c(0.1, 0.1),
                          a_HOMU=c(0.1, 0.1), a_PLER=c(0.1, 0.1),
                          a_SACO=c(0.1, 0.1),a_URLI=c(0.1, 0.1)),
               lower = c( 1, 1, rep(.001, 12) ),
               upper = c(10000, 10000, rep(2, 12)),
               control = list(maxiter = 100000),
               algorithm = 'port')

fit_test_nosplit<- nls(log(num_seeds)~log(lambda/(1+a_ACWR*N_acwr+a_FEMI*N_femi+
                                                    a_HOMU*N_homu+a_PLER*N_pler+
                                                    a_SACO*N_saco+a_URLI*N_urli)),
                       data=focal_data,
                       start=list('lambda'= c(100),
                                  a_ACWR=c(0.1), a_FEMI=c(0.1),
                                  a_HOMU=c(0.1), a_PLER=c(0.1),
                                  a_SACO=c(0.1),a_URLI=c(0.1)),
                       lower = c( 1, rep(.001, 6) ),
                       upper = c(10000, rep(2, 6)),
                       control = list(maxiter = 100000),
                       algorithm = 'port')

fit_test_justA<- nls(log(num_seeds)~log(lambda/(1+a_ACWR[Tr]*N_acwr+a_FEMI[Tr]*N_femi+
                                                  a_HOMU[Tr]*N_homu+a_PLER[Tr]*N_pler+
                                                  a_SACO[Tr]*N_saco+a_URLI[Tr]*N_urli)),
                     data=focal_data,
                     start=list('lambda'= c(100),a_ACWR=c(0.1, 0.1), a_FEMI=c(0.1, 0.1),
                                a_HOMU=c(0.1, 0.1), a_PLER=c(0.1, 0.1),
                                a_SACO=c(0.1, 0.1),a_URLI=c(0.1, 0.1)),
                     lower = c( 1, rep(.001, 12) ),
                     upper = c(10000, rep(2, 12)),
                     control = list(maxiter = 100000),
                     algorithm = 'port')

fit_test_justLam<- nls(log(num_seeds)~log(lambda/(1+a_ACWR[Tr]*N_acwr+a_FEMI[Tr]*N_femi+
                                                    a_HOMU[Tr]*N_homu+a_PLER[Tr]*N_pler+
                                                    a_SACO[Tr]*N_saco+a_URLI[Tr]*N_urli)),
                       data=focal_data,
                       start=list('lambda'= c(100,100),a_ACWR=c(0.1, 0.1), a_FEMI=c(0.1, 0.1),
                                  a_HOMU=c(0.1, 0.1), a_PLER=c(0.1, 0.1),
                                  a_SACO=c(0.1, 0.1),a_URLI=c(0.1, 0.1)),
                       lower = c( 1,1, rep(.001, 12) ),
                       upper = c(10000,10000, rep(2, 12)),
                       control = list(maxiter = 100000),
                       algorithm = 'port')

XX <- data.frame(FocalSp = focalsp,
                 LogL_NoTre = logLik(fit_test_nosplit),
                 LogL_JustA = logLik(fit_test_justA),
                 LogL_JustL = logLik(fit_test_justLam),
                 LogL_Orig  = logLik(fit_test)) %>%
```

```
    mutate(k_NoTre =7,
           k_JustA =13,
           k_JustL =8,
           k_Orig = 14)%>%
    mutate(aic_NoTre = 2*k_NoTre - (2*LogL_NoTre) ,
           aic_JustA = 2*k_JustA - (2*LogL_JustA) ,
           aic_JustL = 2*k_JustL - (2*LogL_JustL) ,
           aic_Orig  = 2*k_Orig  - (2*LogL_Orig))

return(XX)
}
```

4.1 AIC Tables

```
AICTable<- map_df( spp_list, FindFourAICs)



knitr::kable(t(AICTable), digits = 1)
```

FocalSp	ACWR	FEMI	HOMU	PLER	SACO	URLI
LogL_NoTre	-438.2903	-417.4120	-268.7606	-440.4331	-407.5359	-332.6062
LogL_JustA	-432.2594	-413.8134	-263.8901	-421.0175	-392.0230	-326.2338
LogL_JustL	-432.1867	-413.6031	-262.7776	-420.4936	-392.0226	-325.7340
LogL_Orig	-432.0466	-407.1755	-263.6904	-420.9895	-391.1440	-325.8967
k_NoTre	7	7	7	7	7	7
k_JustA	13	13	13	13	13	13
k_JustL	8	8	8	8	8	8
k_Orig	14	14	14	14	14	14
aic_NoTre	890.5807	848.8241	551.5213	894.8662	829.0719	679.2124
aic_JustA	890.5187	853.6269	553.7802	868.0349	810.0460	678.4675
aic_JustL	880.3734	843.2062	541.5552	856.9872	800.0453	667.4680
aic_Orig	892.0933	842.3510	555.3809	869.9789	810.2881	679.7933

```
knitr::kable(select(AICTable,FocalSp , starts_with('aic')), digits = 1)
```

FocalSp	aic_NoTre	aic_JustA	aic_JustL	aic_Orig
ACWR	890.6	890.5	880.4	892.1
FEMI	848.8	853.6	843.2	842.4
HOMU	551.5	553.8	541.6	555.4
PLER	894.9	868.0	857.0	870.0
SACO	829.1	810.0	800.0	810.3
URLI	679.2	678.5	667.5	679.8

```
dfLL <- data.frame(LogLik = select(AICTable, starts_with('LogL')) %>%
                     summarise_all(sum) %>% t)
```

```
dfLL$n_params = c(84, 42, 78, 48)

dfLL %>%
  mutate( AIC= (2*n_params) - (2*LogLik) )
```

```
##               LogLik n_params      AIC
## LogL_NoTre -2305.038       84 4778.076
## LogL_JustA -2249.237       42 4582.474
## LogL_JustL -2246.818       78 4649.635
## LogL_Orig  -2240.943       48 4577.885
```

5 Full Posteriors

5.0.1 Which pair-directions to plot?

```
#get all the pairs where the fitness difference is above 1 in the wet plots----
nls_boot_pairs$fd_superior <- ifelse(nls_boot_pairs$fd < 1, 1/nls_boot_pairs$fd, nls_boot_pairs$fd)
nls_boot_pairs$fd_sup_sp <- ifelse(nls_boot_pairs$fd <= 1, 1, 2)
nls_boot_pairs_sup <- nls_boot_pairs %>%filter(fd_sup_sp == 2 )
W_superior <- with(nls_boot_pairs_sup, sp_pair[treatment==1])

boots_pairs_w_sup <- nls_boot_pairs %>% filter(sp_pair %in% W_superior)
boots_pairs_w_sup$treat <- factor(boots_pairs_w_sup$treat, levels = c(1, 2))

boots_pairs_w_sup$treatment <- factor(boots_pairs_w_sup$treatment, levels = c(1, 2))
boots_pairs_w_sup$label <- paste0(substr(boots_pairs_w_sup$focal, 1, 2),
                                  "-", substr(boots_pairs_w_sup$competitor, 1, 2))

PairsToPlot <- unique(boots_pairs_w_sup$sp_pair )
```

5.0.2 Data Preparation

```
final_output_nls_boot <-read.csv(paste0(Path_to_Orig,
                                        "output/final_output_nls_boot_1000.csv"))
spp_treat_boot_combos <- final_output_nls_boot

#Data frame with medians and sds for parameters----
spp_list <- sort(na.omit( unique(seed_data$focal)))
treat_list <- sort( na.omit( unique(seed_data$Tr)))
spp_treat_combos <- expand.grid(species = spp_list, treatment = treat_list)
comp_labels <- sort( na.omit( unique(seed_data$background) ))

spp_treat_boot_combos$sp_pair <- paste(spp_treat_boot_combos$focal,
                                       spp_treat_boot_combos$competitor, sep = "_")

spp_treat_comp_combos <- expand.grid(focal = spp_list,
                                     competitor = comp_labels,
                                     treatment = treat_list)
```

```
nls_boot_pairs$sp_pair <- paste(nls_boot_pairs$focal, nls_boot_pairs$competitor, sep = "_")

nls_boot_pairs %>%
  filter(sp_pair %in% PairsToPlot) %>%
  mutate( Treatment = ifelse(treatment ==1, 'WET', 'DRY')) ->  OriginalErrorBarsForAdding
```

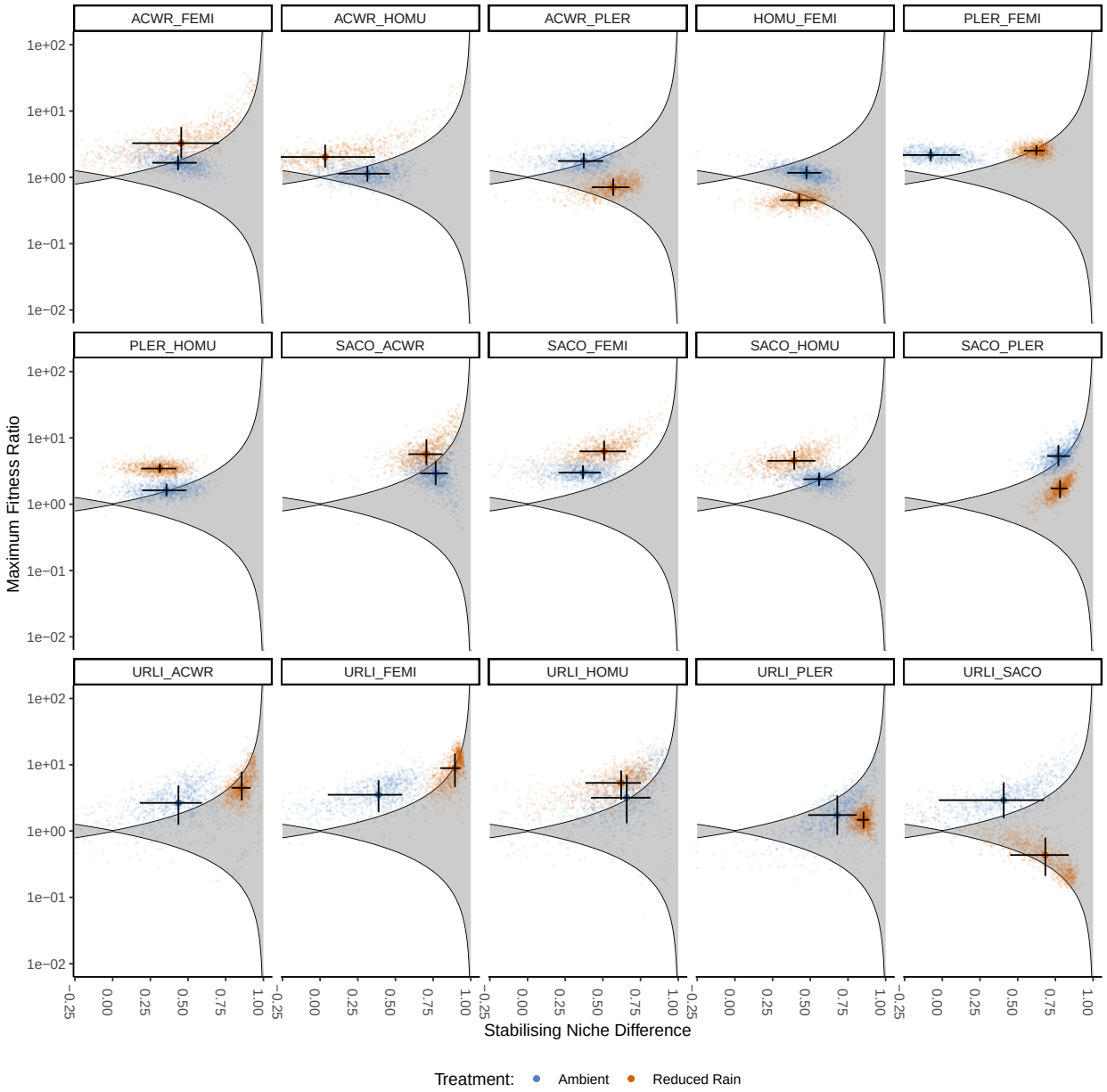
5.0.3 Create coexistence area for plot - min/max fitness difference that permits coexistence

```
niche_differentiation <- seq(from = -.25, to = 1, by = 0.001)
niche_overlap <- 1-niche_differentiation
fitness_ratio_min <- niche_overlap
fitness_ratio_max <- 1/niche_overlap

coexistarea_df <- data.frame(niche_diff = niche_differentiation,
                             min_fitness_ratio = fitness_ratio_min,
                             max_fitness_ratio = fitness_ratio_max)
```

5.1 Plotting

```
spp_treat_boot_combos %>%
  filter(sp_pair %in% PairsToPlot) %>%
  mutate( Treatment = ifelse(treatment ==1, 'WET', 'DRY')) %>%
  arrange(snd) %>%
  ggplot()+
  geom_line(data = coexistarea_df, aes(x = niche_diff, y = max_fitness_ratio)) +
  geom_line(data = coexistarea_df,  aes(x = niche_diff, y = min_fitness_ratio)) +
  geom_ribbon(data = coexistarea_df, aes(x = niche_diff, ymin = min_fitness_ratio,
                                         ymax = max_fitness_ratio), fill = 'grey80') +
  geom_point(aes( x = snd, y = fd,col = Treatment), alpha = 0.1, size = 0.1)+
  facet_wrap(~sp_pair,  nrow = 3)+
  scale_y_log10()+
  coord_cartesian(xlim = c(-0.2,1), ylim = c(0.01, 100) )+
  theme_classic()+
  geom_point(aes(x = snd, y = fd, color = Treatment),
             data= OriginalErrorBarsForAdding) +
  geom_errorbar(aes(x = snd, ymin = fd_low, ymax = fd_high),
                data= OriginalErrorBarsForAdding) +
  geom_errorbarh(aes(y = fd, xmin = snd_low, xmax =snd_high),
                 data= OriginalErrorBarsForAdding)+
  scale_color_manual(values=c('WET' ="#4E84C4", 'DRY' = "#D16103"),
                     name = "Treatment:",
                     labels = c("Ambient", "Reduced Rain"))+
  theme(legend.position = 'bottom', axis.text.x = element_text(angle = -90))+
  ylab('Maximum Fitness Ratio')+
  xlab('Stabilising Niche Difference')
```




```
ggsave('ExFig2_CoexistPlots.png', height = 10, width = 8, dpi = 200)
```

5.2 Determining fraction of pairs that follow identified pattern

```
# original results
nls_boot_pairs_unique <-  nls_boot_pairs %>%  filter(focal != competitor )
nls_boot_pairs_unique$treatment <- factor(nls_boot_pairs_unique$treatment, levels = c(1, 2))
nls_boot_pairs_unique$fd_superior <- ifelse(nls_boot_pairs_unique$fd < 1,
                                            1/nls_boot_pairs_unique$fd,
                                            nls_boot_pairs_unique$fd)
nls_boot_pairs_unique$coexist <- ifelse((nls_boot_pairs_unique$snd >
```

```
                                          (1-1/nls_boot_pairs_unique$fd_superior)),
                                        1, 0 )

##Stabilizing niche and fitness differences: (Table ED3)
pars_boot <- nls_boot_pairs_unique %>%
  select(focal,competitor,treatment,snd,
         fd,fd_superior,coexist, fd_sup_sp)

pars_boot$species <- paste0(substr(pars_boot$focal, 1, 2),
                            "-",
                            substr(pars_boot$competitor, 1, 2))

pars_boot$outcome <- ifelse(pars_boot$coexist == 1, "coexist",
                            ifelse(pars_boot$fd_sup_sp == 2,
                                   paste0(substr(pars_boot$species, 4, 5), " wins"),
                                   paste0(substr(pars_boot$species, 1, 2), " wins")))

pair_labels <- pars_boot %>%
  filter(treatment == 1 ) %>%
  filter(fd > 1)

pair_labels <- unique(pair_labels$species)
pars_boot <- pars_boot %>% filter(species %in% pair_labels)
pars_boot <- subset(pars_boot, select =  -c(focal, competitor, coexist,
                                            fd_sup_sp, fd_superior))

pars_boot_wide<-pivot_wider(data = pars_boot,
                            names_from = treatment,
                            values_from = c(snd, fd, outcome))%>%
  arrange(species)
```

5.3 Repeating across the whole bootstrap sample

```
spp_treat_boot_combos %>%
  filter(sp_pair %in% PairsToPlot) %>%
  mutate(fd_superior <- ifelse(fd < 1, 1/fd,fd) ,
         fd_sup_sp <- ifelse(fd <= 1, 1, 2),
         coexist = ifelse((snd > (1-1/fd_superior)), 1, 0 ),
         outcome = ifelse(coexist == 1, "coexist",
                          ifelse(fd_sup_sp == 2,
                                 paste0(focal , " wins"),
                                 paste0(competitor , " wins")))) %>%
  group_by(treatment, sp_pair) %>%
  count(outcome) %>%
  pivot_wider(id_cols = c(treatment, sp_pair), names_from = outcome,
              values_from = n, values_fill = 0) -> PosteriorResults

## Just looking at change in 'coexist' result

PosteriorResults %>%
  mutate( Treatment = ifelse(treatment ==1, 'WET', 'DRY'))%>%
```

```
  mutate(coexist_frac = coexist/1000) %>%
  ungroup() %>%
  select(Treatment, sp_pair, coexist_frac) %>%
  pivot_wider(id_cols = sp_pair, names_from = Treatment, values_from = coexist_frac) %>%
  mutate( AbsoluteChange = WET-DRY) %>%
  mutate(species  = paste0(substr(sp_pair     , 1, 2),## For Joining with pars_boot_wide
                           "-",
                           substr(sp_pair     , 6, 7))) %>%
  left_join(pars_boot_wide, by = "species") -> JoinResults


JoinResults$OriginalFindsChange <- JoinResults$outcome_1 != JoinResults$outcome_2


## probability of each draw showing switch from coexistence = prob(coexist1) *prob
## Assuming just two outcomes: coexist, or something else
# If change from victory -> coexist, probability observe change =  1-prob(coexist1) * prob(coexist)
# If change from coexist -> victory, probability observe change =  prob(coexist1) * 1-prob(coexist)

JoinResults %>%
  filter(OriginalFindsChange) %>%
  mutate(LikelihoodObserveChange = ifelse(outcome_1 == 'coexist',
                                          WET *(1-DRY),
                                          (1-WET)*DRY)) %>%
  select(SpeciesPair =species,
         LikelihoodCoexist_Wet= WET,
         LikelihoodCoexist_Dry = DRY,
         outcome_Wet=outcome_1, outcome_Dry=outcome_2,
         LikelihoodObserveChange )-> JoinResults2

knitr::kable(JoinResults2,digits =3)
```

SpeciesPair	LikelihoodCoexist_Wet	LikelihoodCoexist_Dry	outcome_Wet	outcome_Dry	LikelihoodObserveChange
AC-FE	0.554	0.069	coexist	FE wins	0.516
AC-HO	0.715	0.004	coexist	HO wins	0.712
AC-PL	0.361	0.855	PL wins	coexist	0.546
HO-FE	0.901	0.225	coexist	HO wins	0.698
PL-FE	0.005	0.551	FE wins	coexist	0.548
SA-AC	0.729	0.095	coexist	AC wins	0.660
SA-PL	0.285	0.998	PL wins	coexist	0.714
UR-AC	0.191	0.830	AC wins	coexist	0.671
UR-FE	0.032	0.521	FE wins	coexist	0.504
UR-SA	0.092	0.800	SA wins	coexist	0.726

```
JoinResults2$LikelihoodObserveChange %>% mean
```

```
## [1] 0.6296292
```

6 Session Info

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19045)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] forcats_0.5.2   stringr_1.4.1   dplyr_1.0.10    purrr_0.3.5
## [5] readr_2.1.3     tidyr_1.2.1     tibble_3.1.8    ggplot2_3.3.6
## [9] tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.2.0    xfun_0.34           haven_2.5.1
##  [4] gargle_1.2.1        colorspace_2.0-3    vctrs_0.5.0
##  [7] generics_0.1.3      htmltools_0.5.3     yaml_2.3.6
## [10] utf8_1.2.2          rlang_1.0.6         pillar_1.8.1
## [13] withr_2.5.0         glue_1.6.2          DBI_1.1.3
## [16] dbplyr_2.2.1        modelr_0.1.9        readxl_1.4.1
## [19] lifecycle_1.0.3     munsell_0.5.0       gtable_0.3.1
## [22] cellranger_1.1.0    ragg_1.2.4          rvest_1.0.3
## [25] evaluate_0.17       labeling_0.4.2      knitr_1.40
## [28] tzdb_0.3.0          fastmap_1.1.0       fansi_1.0.3
## [31] highr_0.9           broom_1.0.1         scales_1.2.1
## [34] backports_1.4.1     googlesheets4_1.0.1 jsonlite_1.8.3
## [37] systemfonts_1.0.4   farver_2.1.1        fs_1.5.2
## [40] textshaping_0.3.6   hms_1.1.2           digest_0.6.30
## [43] stringi_1.7.8       grid_4.1.1          cli_3.4.1
## [46] tools_4.1.1         magrittr_2.0.3      crayon_1.5.2
## [49] pkgconfig_2.0.3     ellipsis_0.3.2      xml2_1.3.3
## [52] reprex_2.0.2        googledrive_2.0.0   lubridate_1.8.0
## [55] assertthat_0.2.1    rmarkdown_2.17      httr_1.4.4
## [58] rstudioapi_0.14     R6_2.5.1            compiler_4.1.1
```