

Supplementary Methods 1 for ‘Uncertain competition coefficients undermine inferences about coexistence’

J. Christopher D. Terry

Contents

1	Summary	1
2	Loading and looking at data	1
2.1	How much data?	2
2.2	How much uncertainty in types of parameters	2
3	Plotting raw data with model best-fit	4
4	Fitting set of models with different levels of parameterisation	5
4.1	Model Comparison Tables	7
5	Full Posteriors	8
5.1	Plotting	9
5.2	Determining fraction of pairs that follow identified pattern	11
5.3	Repeating across the whole bootstrap sample	12
6	Session Info	16

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 propagates 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

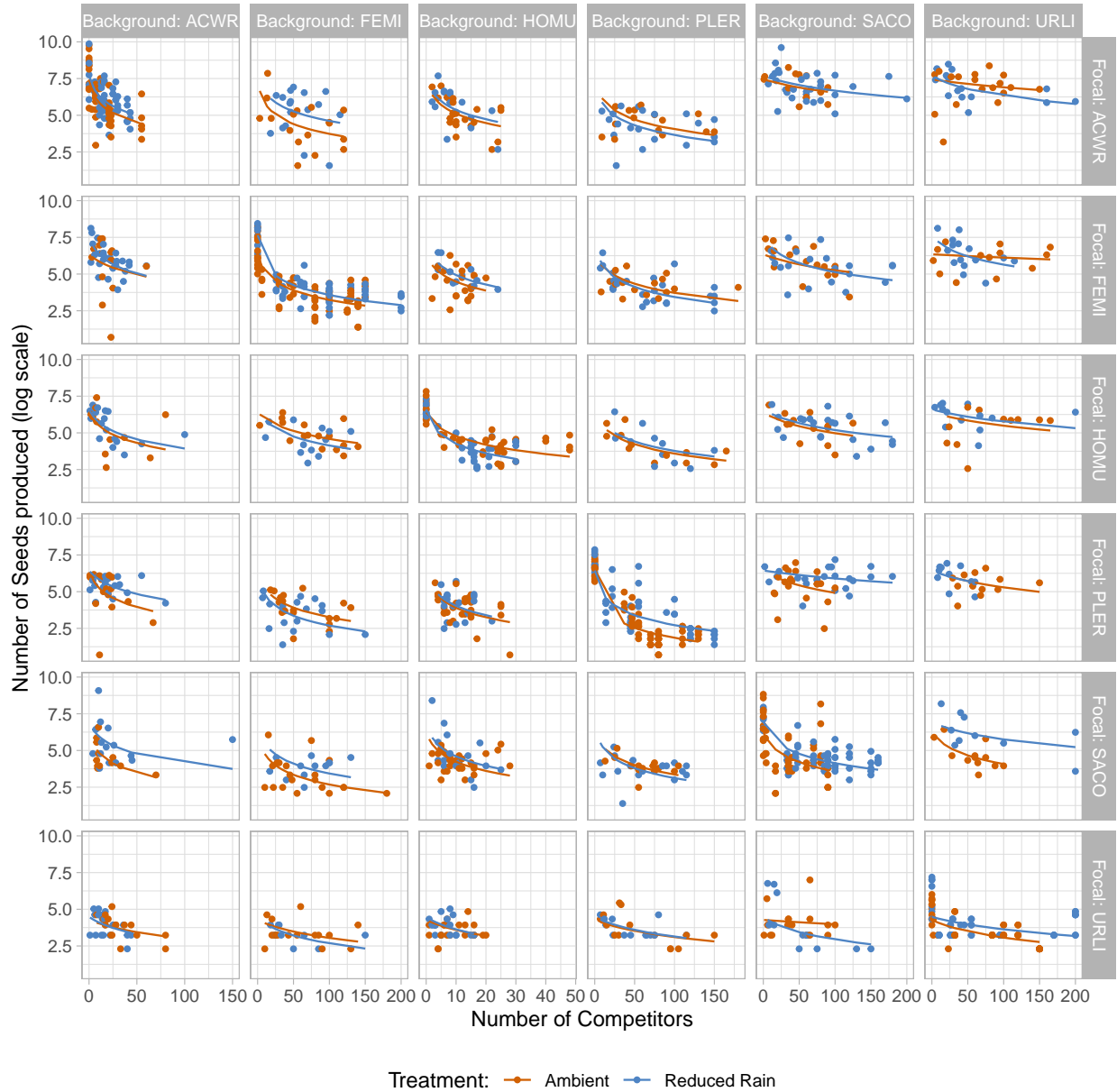
```
spp_list <- sort(na.omit( unique(seed_data$focal)))
```

```
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[Tr]/(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(100,100),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( 1,1, rep(.001, 6) ),
upper = c(10000,10000, rep(2, 6)),
control = list(maxiter = 100000),
algorithm = 'port')

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

```

```

mutate(k_NoTre =7,
       k_JustL =8,
       k_JustA =13,
       k_Orig = 14)%>%
mutate(aic_1NoTre = 2*k_NoTre - (2*LogL_NoTre) ,
       aic_2JustL = 2*k_JustL - (2*LogL_JustL) ,
       aic_3JustA = 2*k_JustA - (2*LogL_JustA) ,
       aic_4Orig = 2*k_Orig - (2*LogL_Orig))

return(XX)
}

```

4.1 Model Comparison Tables

```

AICTable<- map_df( spp_list, FindFourAICs) %>%
  left_join(count(seed_data, focal), c('FocalSp'='focal') )

AICTable %>%
  mutate(BIC_1NoTre = log(n)*k_NoTre - (2*LogL_NoTre),
         BIC_2JustL = log(n)*k_JustL - (2*LogL_JustL),
         BIC_3JustA = log(n)*k_JustA - (2*LogL_JustA),
         BIC_4Orig = log(n)*k_Orig - (2*LogL_Orig)) -> AICTable

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_JustL	-436.6224	-413.6492	-268.5777	-432.7851	-397.2313	-331.9808
LogL_JustA	-432.2594	-413.8134	-263.8901	-421.0175	-392.0230	-326.2338
LogL_Orig	-432.0466	-407.1755	-263.6904	-420.9895	-391.1440	-325.8967
k_NoTre	7	7	7	7	7	7
k_JustL	8	8	8	8	8	8
k_JustA	13	13	13	13	13	13
k_Orig	14	14	14	14	14	14
aic_1NoTre	890.5807	848.8241	551.5213	894.8662	829.0719	679.2124
aic_2JustL	889.2448	843.2984	553.1553	881.5702	810.4626	679.9616
aic_3JustA	890.5187	853.6269	553.7802	868.0349	810.0460	678.4675
aic_4Orig	892.0933	842.3510	555.3809	869.9789	810.2881	679.7933
n	293	305	214	325	274	266
BIC_1NoTre	916.3419	874.8663	575.0831	921.3529	854.3637	704.2969
BIC_2JustL	918.6861	873.0609	580.0831	911.8408	839.3676	708.6295
BIC_3JustA	938.3610	901.9909	597.5379	917.2246	857.0167	725.0530
BIC_4Orig	943.6157	894.4353	602.5045	922.9525	860.8719	729.9623

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

FocalSp	aic_1NoTre	aic_2JustL	aic_3JustA	aic_4Orig
ACWR	890.6	889.2	890.5	892.1
FEMI	848.8	843.3	853.6	842.4
HOMU	551.5	553.2	553.8	555.4
PLER	894.9	881.6	868.0	870.0
SACO	829.1	810.5	810.0	810.3
URLI	679.2	680.0	678.5	679.8

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

FocalSp	BIC_1NoTre	BIC_2JustL	BIC_3JustA	BIC_4Orig
ACWR	916.3	918.7	938.4	943.6
FEMI	874.9	873.1	902.0	894.4
HOMU	575.1	580.1	597.5	602.5
PLER	921.4	911.8	917.2	923.0
SACO	854.4	839.4	857.0	860.9
URLI	704.3	708.6	725.1	730.0

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

select(AICTable, starts_with('aic')) %>% colSums
```

```
## aic_1NoTre aic_2JustL aic_3JustA aic_4Orig
## 4694.076 4657.693 4654.474 4649.885
```

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)
```

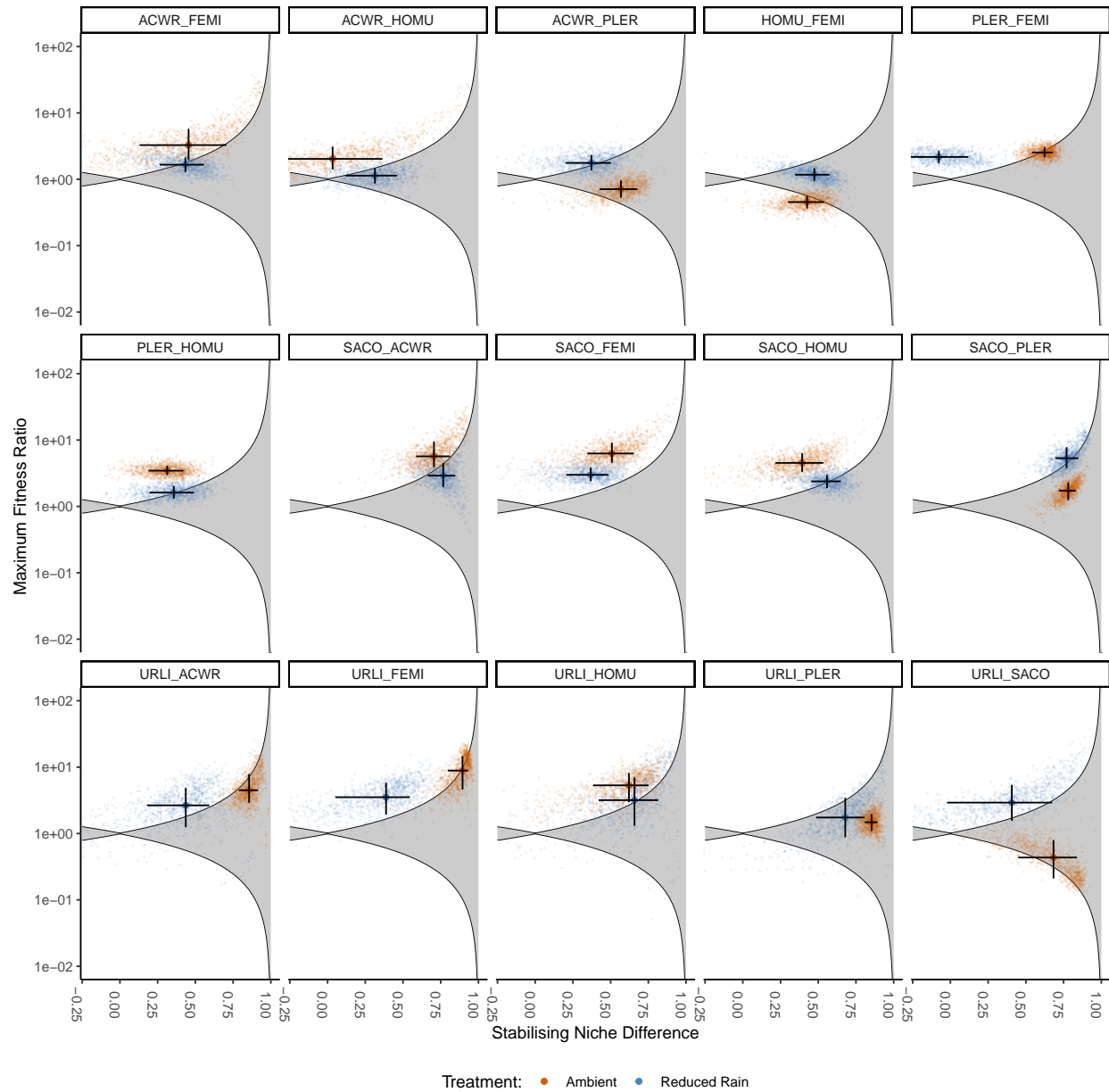
5.1 Plotting

```
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(LK_ObserveChange = ifelse(outcome_1 == 'coexist',
                                   WET *(1-DRY),
                                   (1-WET)*DRY),
         LK_CoexistBoth = WET*DRY,
         LK_NoCoexistBoth = (1-WET)*(1-DRY),
         LK_ObserveInverseChange = ifelse(outcome_1 == 'coexist',
                                           (1-WET)*DRY,
                                           WET *(1-DRY)),
         CHECK= LK_ObserveChange+LK_CoexistBoth+LK_NoCoexistBoth+LK_ObserveInverseChange) %>%
  mutate( MostLikelyNonChange = ifelse( LK_CoexistBoth>LK_NoCoexistBoth,
                                       LK_CoexistBoth,LK_NoCoexistBoth ),
         MostLikelyChange = ifelse( LK_ObserveChange >LK_ObserveInverseChange,
                                    LK_ObserveChange,LK_ObserveInverseChange ))%>%
  arrange(desc(MostLikelyChange))%>%
  select(SpeciesPair =species, WET,DRY,
         # starts_with('LK'),
         MostLikelyChange, MostLikelyNonChange ,
         outcome_Wet=outcome_1, outcome_Dry=outcome_2 )-> JoinResults2

knitr::kable(JoinResults2,digits =3)

```

SpeciesPair	WET	DRY	MostLikelyChange	MostLikelyNonChange	outcome_Wet	outcome_Dry
UR-SA	0.092	0.800	0.726	0.182	SA wins	coexist
SA-PL	0.285	0.998	0.714	0.284	PL wins	coexist
AC-HO	0.715	0.004	0.712	0.284	coexist	HO wins
HO-FE	0.901	0.225	0.698	0.203	coexist	HO wins
UR-AC	0.191	0.830	0.671	0.159	AC wins	coexist
SA-AC	0.729	0.095	0.660	0.245	coexist	AC wins
PL-FE	0.005	0.551	0.548	0.447	FE wins	coexist
AC-PL	0.361	0.855	0.546	0.309	PL wins	coexist
AC-FE	0.554	0.069	0.516	0.415	coexist	FE wins

SpeciesPair	WET	DRY	MostLikelyChange	MostLikelyNonChange	outcome_Wet	outcome_Dry
UR-FE	0.032	0.521	0.504	0.464	FE wins	coexist
PL-HO	0.440	0.000	0.440	0.560	HO wins	HO wins
SA-HO	0.428	0.004	0.426	0.570	HO wins	HO wins
UR-HO	0.378	0.059	0.356	0.585	HO wins	HO wins
UR-PL	0.841	1.000	0.159	0.841	coexist	coexist
SA-FE	0.014	0.000	0.014	0.986	FE wins	FE wins

NB the very small number of priority effects are being effectively ignored here (assigned to a win for

```
PosteriorResults %>%
  mutate( Treatment = ifelse(treatment ==1, 'WET', 'DRY'))%>%
  ungroup()%>%
  select( -treatment) %>%
  group_by(sp_pair, Treatment) %>%
  pivot_longer(names_to = 'Outcome', cols = `ACWR wins` : `FEMI wins` ) %>%
  ungroup()->Draws

Dry_Draws<- filter(Draws, Treatment == 'DRY', value>0) %>% rename( Dry_Outcome = Outcome, Dry_draws = value)
Wet_Draws<- filter(Draws, Treatment == 'WET', value>0) %>% rename( Wet_Outcome = Outcome, Wet_draws = value)

full_join( Wet_Draws,Dry_Draws,
           by = 'sp_pair', relationship = "many-to-many") %>%
  select(-Treatment.x, -Treatment.y) %>%
  mutate( comb_prob = (Dry_draws/1000) * (Wet_draws/1000) )%>%
  arrange( sp_pair, desc(comb_prob)) -> All_OutcomePaths

All_OutcomePaths
```

```
## # A tibble: 80 x 6
##   sp_pair   Wet_Outcome Wet_draws Dry_Outcome Dry_draws comb_prob
##   <chr>      <chr>      <int> <chr>      <int>      <dbl>
## 1 ACWR_FEMI coexist      554 ACWR wins      930  0.515
## 2 ACWR_FEMI ACWR wins      446 ACWR wins      930  0.415
## 3 ACWR_FEMI coexist      554 coexist        69  0.0382
## 4 ACWR_FEMI ACWR wins      446 coexist        69  0.0308
## 5 ACWR_FEMI coexist      554 FEMI wins         1  0.000554
## 6 ACWR_FEMI ACWR wins      446 FEMI wins         1  0.000446
## 7 ACWR_HOMU coexist      715 ACWR wins      976  0.698
## 8 ACWR_HOMU ACWR wins      222 ACWR wins      976  0.217
## 9 ACWR_HOMU HOMU wins        63 ACWR wins      976  0.0615
## 10 ACWR_HOMU coexist      715 HOMU wins        20  0.0143
## # i 70 more rows
```

```
All_OutcomePaths %>%
  group_by(sp_pair) %>%
  mutate(rank = 1:n()) %>%
  filter(rank <3) %>%
  select( -Wet_draws, - Dry_draws) %>%
  pivot_wider( id_cols = sp_pair,
               values_from = c(Wet_Outcome,Dry_Outcome,comb_prob ),
               names_from = rank) %>%
```

sp_pair	Wet_Outcome_1	Dry_Outcome_1	comb_prob_1	Wet_Outcome_2	Dry_Outcome_2	comb_prob_2
SACO_PLER	SACO wins	coexist	0.714	coexist	coexist	0.284
URLI_SACO	URLI wins	coexist	0.702	URLI wins	SACO wins	0.158
HOMU_FEMI	coexist	FEMI wins	0.698	coexist	coexist	0.203
ACWR_HOMU	coexist	ACWR wins	0.698	ACWR wins	ACWR wins	0.217
SACO_ACWR	coexist	SACO wins	0.659	SACO wins	SACO wins	0.245
URLI_ACWR	URLI wins	coexist	0.621	coexist	coexist	0.159
PLER_FEMI	PLER wins	coexist	0.548	PLER wins	PLER wins	0.447
ACWR_PLER	ACWR wins	coexist	0.546	coexist	coexist	0.309
ACWR_FEMI	coexist	ACWR wins	0.515	ACWR wins	ACWR wins	0.415
URLI_FEMI	URLI wins	coexist	0.493	URLI wins	URLI wins	0.453
SACO_FEMI	SACO wins	SACO wins	0.984	coexist	SACO wins	0.014
URLI_PLER	coexist	coexist	0.841	URLI wins	coexist	0.091
SACO_HOMU	SACO wins	SACO wins	0.568	coexist	SACO wins	0.425
PLER_HOMU	PLER wins	PLER wins	0.557	coexist	PLER wins	0.440
URLI_HOMU	URLI wins	URLI wins	0.550	coexist	URLI wins	0.356

```
mutate(ChangeIdentified = Wet_Outcome_1!=Dry_Outcome_1 ) %>%
select( sp_pair , ChangeIdentified, ### ReOrdering Columns
        Wet_Outcome_1 , Dry_Outcome_1, comb_prob_1,
        Wet_Outcome_2 , Dry_Outcome_2,  comb_prob_2)%>%
arrange( desc(ChangeIdentified) , desc(comb_prob_1 )) -> TopTwoMostLikelyOutcomes
```

```
knitr::kable(select(TopTwoMostLikelyOutcomes,
                    -ChangeIdentified),
              digits =3, format = 'latex', booktabs = TRUE) %>%
kableExtra::kable_styling(latex_options = "scale_down")
```

```
TopTwoMostLikelyOutcomes %>%
  filter( ChangeIdentified) %>%
  mutate(RelProb = comb_prob_1 /comb_prob_2 ) %>%
  ungroup %>%summarise( mean(RelProb), mean(comb_prob_1))
```

```
## # A tibble: 1 x 2
##   'mean(RelProb)' 'mean(comb_prob_1)'
##   <dbl>          <dbl>
## 1         2.56         0.619
```

```
## Checker:
All_OutcomePaths %>%group_by(sp_pair) %>%summarise(sum(comb_prob))
```

```
## # A tibble: 15 x 2
##   sp_pair   'sum(comb_prob)'
##   <chr>         <dbl>
## 1 ACWR_FEMI         1
## 2 ACWR_HOMU         1
## 3 ACWR_PLER         1
## 4 HOMU_FEMI         1
## 5 PLER_FEMI         1
## 6 PLER_HOMU         1
## 7 SACO_ACWR         1
## 8 SACO_FEMI         1
```

```
## 9 SACO_HOMU          1
## 10 SACO_PLER         1
## 11 URLI_ACWR         1
## 12 URLI_FEMI         1
## 13 URLI_HOMU         1
## 14 URLI_PLER         1
## 15 URLI_SACO         1
```

6 Session Info

```
sessionInfo()
```

```
## R version 4.3.0 (2023-04-21 ucrt)
## 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.utf8
## [2] LC_CTYPE=English_United Kingdom.utf8
## [3] LC_MONETARY=English_United Kingdom.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.utf8
##
## time zone: Europe/London
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] lubridate_1.9.2 forcats_1.0.0  stringr_1.5.0  dplyr_1.1.2
## [5] purrr_1.0.1     readr_2.1.4    tidyr_1.3.0    tibble_3.2.1
## [9] ggplot2_3.4.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] utf8_1.2.3      generics_0.1.3  xml2_1.3.4      stringi_1.7.12
## [5] hms_1.1.3       digest_0.6.31   magrittr_2.0.3  evaluate_0.21
## [9] grid_4.3.0      timechange_0.2.0 fastmap_1.1.1    httr_1.4.6
## [13] rvest_1.0.3     fansi_1.0.4     viridisLite_0.4.2 scales_1.2.1
## [17] textshaping_0.3.6 cli_3.6.1       rlang_1.1.1     munsell_0.5.0
## [21] withr_2.5.0     yaml_2.3.7      tools_4.3.0     tzdb_0.4.0
## [25] colorspace_2.1-0 webshot_0.5.5   kableExtra_1.3.4 vctr_0.6.2
## [29] R6_2.5.1        lifecycle_1.0.3 ragg_1.2.5       pkgconfig_2.0.3
## [33] pillar_1.9.0    gtable_0.3.3    glue_1.6.2       systemfonts_1.0.4
## [37] xfun_0.39       tidyselect_1.2.0 highr_0.10        rstudioapi_0.14
## [41] knitr_1.43      farver_2.1.1    htmltools_0.5.5  svglite_2.1.1
## [45] rmarkdown_2.22  labeling_0.4.2  compiler_4.3.0
```