

Community context mediates effects of pollinator loss on seed production

K. C. Arrowsmith, V. A. Reynolds, H. M Briggs, and B. J. Brosi

```
knitr::opts_chunk$set(warning = FALSE, message = FALSE)
```

Overview

This document contains all code necessary to replicate the analyses for the above article. Any questions about this project can be directed to Kaysee Arrowsmith at kcarrows@uw.edu.

Particulars

The data for this analysis come from an experimental bumble bee removal experiment that took place over three years (2011, 2013, and 2014; in 2012, there was a severe drought and we were not able to conduct the experiment) across 14 replicates. Each site had two experimental states (control and manipulation). At each site, we selected focal *Delphinium barbeyi* individuals that were identified with ID numbers (`delph.plant.num`). We further selected specific flowers on that focal individual (`delph.flower.num`) for which we collected and counted seeds. *D. barbeyi* flowers typically produce three carpels, so each selected *D. barbeyi* flower may appear in the dataset up to three times, with each row presenting a separate count of viable seeds produced by that carpel. For each site/state combination, we conducted a pollinator survey, which is summarized in this dataset with *Bombus* richness (`bombus.rich`), abundance (`bombus.abund`, scaled to `bomabund.scale`), and mean fidelity (`mean.fidelity`). The identities of these bees were also considered to determine the relative abundance of long-tongued bees (`prop.long`). For manipulation surveys only, we noted the species of *Bombus* removed (`species.removed`, always the most abundant species from that day's survey) and the number and relative abundance of that species (`prop.removed`). Once per site (during the control period), we performed a floral survey in which is summarized in these data with the relative abundance of *D. barbeyi* (`prop.delph`) and the similarity between the rest of the floral community and *D. barbeyi* on the morphological axes of color (`color.sim`) and corolla length (`corolla.sim`).

Load Packages, Functions, Data

```
library(tidyverse)
library(kableExtra)
library(broom.mixed)
library(glmmTMB)
library(MuMIn)
library(xtable)
library(DHARMA)
library(performance)
library(MASS)
library(RColorBrewer)
theme_set(theme_classic())
```

```

# Function to calculate confidence intervals
## https://stackoverflow.com/questions/48612153/how-to-calculate-confidence-intervals-for-a-vector
confidence_interval <- function(vector, interval) {
  # Standard deviation of sample
  vec_sd <- sd(vector)
  # Sample size
  n <- length(vector)
  # Mean of sample
  vec_mean <- mean(vector)
  # Error according to t distribution
  error <- qt((interval + 1)/2, df = n - 1) * vec_sd / sqrt(n)
  # Confidence interval as a vector
  result <- c("lower" = vec_mean - error, "upper" = vec_mean + error)
  return(result)
}

```

Clean data

```

dat <- read.csv("../Data/RMBLseeds-cleaned.csv", stringsAsFactors = F) %>%
  filter(!is.na(viable))

```

Check Model Assumptions

Overdispersion

We start with a global model using a Poisson distribution and check for overdispersion.

```

global.mod <- glmmTMB(viable ~
  prop.removed +
  bomabund.scale +
  prop.long +
  mean.fidelity +
  prop.delph +
  color.sim +
  corolla.sim +
  prop.removed:bomabund.scale +
  prop.removed:prop.long +
  prop.removed:mean.fidelity +
  prop.removed:prop.delph +
  prop.removed:color.sim +
  prop.removed:corolla.sim +
  (1|site/delph.plant.num/delph.flower.num) +
  (1|year),
  data = dat,
  family = poisson)

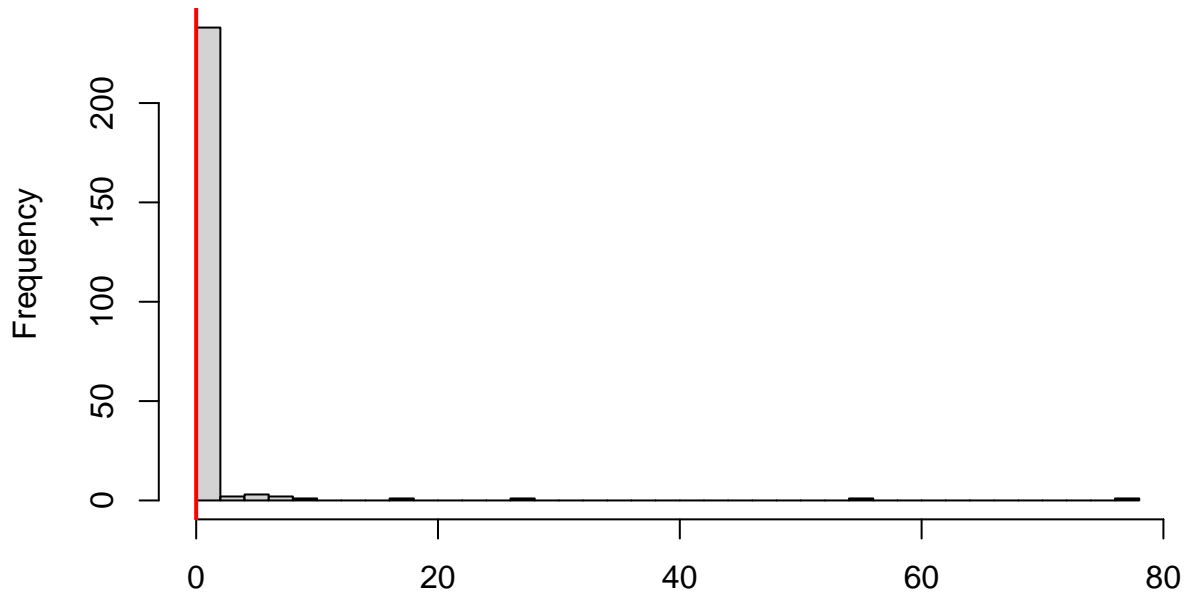
check_overdispersion(global.mod)

## # Overdispersion test
##
##      dispersion ratio =      1.907
##  Pearson's Chi-Squared = 3387.408
##      p-value = < 0.001

```

```
testDispersion(simulateResiduals(global.mod))
```

DHARMA nonparametric dispersion test via sd of residuals fitted vs. simulated



Simulated values, red line = fitted model. p-value (two.sided) = 0

```
##
## DHARMA nonparametric dispersion test via sd of residuals fitted vs.
## simulated
##
## data: simulationOutput
## dispersion = 0.00010848, p-value < 2.2e-16
## alternative hypothesis: two.sided

Significant overdispersion detected (p < 0.05).
```

Zero Inflation

To account for overdispersion, we switch from a Poisson distribution to a negative binomial.

```
nbinom.mod <- glmmTMB(viable ~
  prop.removed +
  bomabund.scale +
  prop.long +
  mean.fidelity +
  prop.delph +
  color.sim +
  corolla.sim +
  prop.removed:bomabund.scale +
  prop.removed:prop.long +
  prop.removed:mean.fidelity +
  prop.removed:prop.delph +
  prop.removed:color.sim +
```

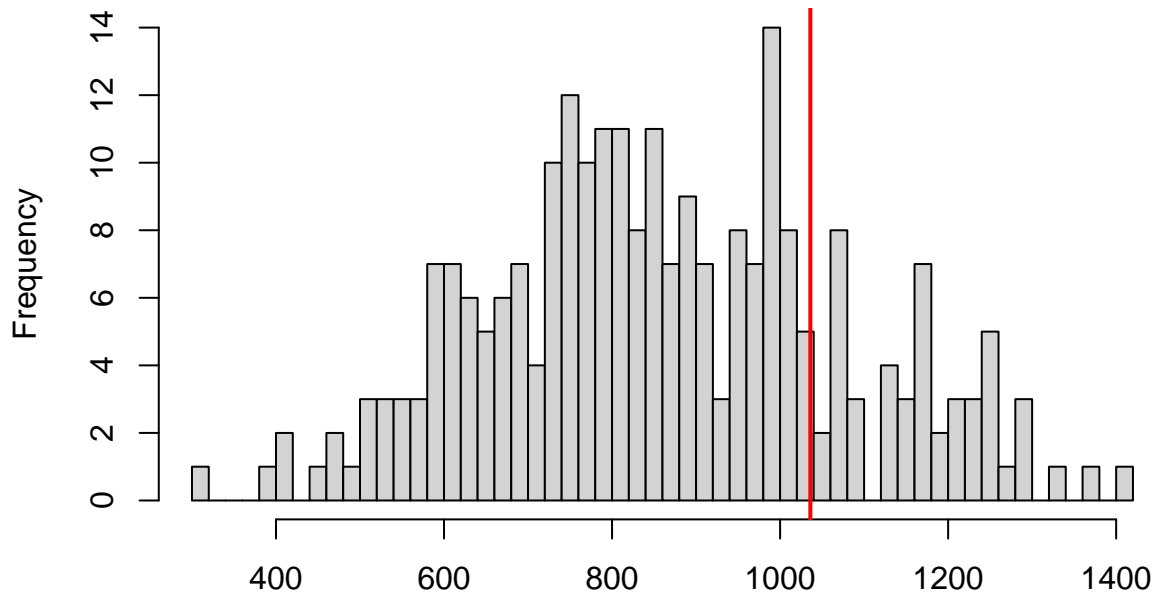
```

prop.removed:corolla.sim +
  (1|site/delph.plant.num/delph.flower.num) +
  (1|year),
data = dat,
family = nbinom2)

testZeroInflation(simulateResiduals(nbinom.mod))

```

**DHARMA zero-inflation test via comparison to
expected zeros with simulation under H0 = fitted
model**



Simulated values, red line = fitted model. p-value (two.sided) = 0.392

```

##
## DHARMA zero-inflation test via comparison to expected zeros with
## simulation under H0 = fitted model
##
## data: simulationOutput
## ratioObsSim = 1.2058, p-value = 0.392
## alternative hypothesis: two.sided

```

Ratio > 1 indicates possible zero-inflation, but the p-value is not significant ($p > 0.05$), so this test is not definitive. We will therefore use two more methods to try to gain more information. First, we perform a chi-squared test to see if there are more zeros than expected. Then, we use an ANOVA to compare models with and without controlling for zero-inflation.

```

# Chi-square test for zero-inflation
contingency <- dat %>%
  mutate(value.cut = cut(viable, breaks = c(-1, 0, 50), labels = c("0", ">0"))) %>%
  with(table(value.cut, state, useNA = "ifany"))
chisq.test(contingency, simulate.p.value = T)

```

```

##
## Pearson's Chi-squared test with simulated p-value (based on 2000

```

```
## replicates)
##
## data: contingency
## X-squared = 5.8232, df = NA, p-value = 0.01249
```

Chi-squared test is significant ($p < 0.05$), providing evidence of possible zero-inflation.

```
# Zero-inflated model
nbinom.zi <- glmmTMB(viable ~
  prop.removed +
  bomabund.scale +
  prop.long +
  mean.fidelity +
  prop.delph +
  color.sim +
  corolla.sim +
  prop.removed:bomabund.scale +
  prop.removed:prop.long +
  prop.removed:mean.fidelity +
  prop.removed:prop.delph +
  prop.removed:color.sim +
  prop.removed:corolla.sim +
  (1|site/delph.plant.num/delph.flower.num) +
  (1|year),
  data = dat,
  family = nbinom2,
  zi = ~.)

# Likelihood Ratio Test
anova(nbinom.mod, nbinom.zi)
```

```
## Data: dat
## Models:
## nbinom.mod: viable ~ prop.removed + bomabund.scale + prop.long + mean.fidelity + , zi=~0, disp=~1
## nbinom.mod:   prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + , zi=~., disp=~1
## nbinom.mod:   prop.removed:prop.long + prop.removed:mean.fidelity + prop.removed:prop.delph + , zi=~., disp=~1
## nbinom.mod:   prop.removed:color.sim + prop.removed:corolla.sim + (1 | , zi=~., disp=~1
## nbinom.mod:   site/delph.plant.num/delph.flower.num) + (1 | year), zi=~0, disp=~1
## nbinom.zi: viable ~ prop.removed + bomabund.scale + prop.long + mean.fidelity + , zi=~., disp=~1
## nbinom.zi:   prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + , zi=~0, disp=~1
## nbinom.zi:   prop.removed:prop.long + prop.removed:mean.fidelity + prop.removed:prop.delph + , zi=~., disp=~1
## nbinom.zi:   prop.removed:color.sim + prop.removed:corolla.sim + (1 | , zi=~0, disp=~1
## nbinom.zi:   site/delph.plant.num/delph.flower.num) + (1 | year), zi=~., disp=~1
##      Df    AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## nbinom.mod 19 7146.5 7250.9 -3554.3   7108.5
## nbinom.zi  37 6331.2 6534.4 -3128.6   6257.2 851.33    18 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

AIC is lower for zero-inflated model, and ANOVA is statistically significant ($p \ll 0.05$).

All of these tests tell us that our data are likely to be zero-inflated and we will therefore use a zero-inflated GLMM.

Collinearity

Next, we check for collinearity between these variables to ensure that they can all be used in model selection without problems.

```
check_collinearity(nbinom.zi)
```

```
## # Check for Multicollinearity
##
## * conditional component:
##
## Low Correlation
##
##           Term  VIF Increased SE Tolerance
##      bomabund.scale 2.72      1.65      0.37
##           prop.long 4.47      2.11      0.22
##      mean.fidelity 4.80      2.19      0.21
##           prop.delph 2.64      1.62      0.38
##           color.sim 3.17      1.78      0.32
##      corolla.sim 2.41      1.55      0.42
## prop.removed:color.sim 4.69      2.17      0.21
## prop.removed:corolla.sim 2.67      1.63      0.37
##
## Moderate Correlation
##
##           Term  VIF Increased SE Tolerance
## prop.removed:prop.delph 5.31      2.30      0.19
##
## High Correlation
##
##           Term    VIF Increased SE Tolerance
##      prop.removed 253.54      15.92      0.00
## prop.removed:bomabund.scale 15.30      3.91      0.07
##      prop.removed:prop.long 185.30      13.61      0.01
##      prop.removed:mean.fidelity 301.41      17.36      0.00
##
## * zero inflated component:
##
## Low Correlation
##
##           Term  VIF Increased SE Tolerance
##      bomabund.scale 2.35      1.53      0.42
##           prop.long 4.97      2.23      0.20
##      mean.fidelity 3.68      1.92      0.27
##           prop.delph 2.99      1.73      0.33
##           color.sim 2.62      1.62      0.38
##      corolla.sim 3.36      1.83      0.30
## prop.removed:color.sim 3.86      1.96      0.26
## prop.removed:corolla.sim 3.63      1.91      0.28
##
## Moderate Correlation
##
##           Term  VIF Increased SE Tolerance
## prop.removed:prop.delph 6.69      2.59      0.15
##
```

```
## High Correlation
##
##           Term      VIF Increased SE Tolerance
##      prop.removed 265.74      16.30      0.00
## prop.removed:bomabund.scale 12.20      3.49      0.08
##      prop.removed:prop.long 234.04      15.30      0.00
## prop.removed:mean.fidelity 348.47      18.67      0.00
```

We see moderate to high correlation between all of the variables involving the manipulation (`prop.removed` and all interaction terms), which is expected. Importantly, we see low correlation with all of the additive variables that do not involve the strength of the manipulation.

Main Analysis

Model Selection

We use a model selection approach to determine which fixed effects are most important in predicting the number of viable seeds produced by *D. barbeyi*.

```
# Creating all combinations of fixed effects and pasting them into a formula
vars <- c("viable",
  "prop.removed",
  "bomabund.scale",
  "prop.long",
  "mean.fidelity",
  "prop.delph",
  "color.sim",
  "corolla.sim",
  "prop.removed:bomabund.scale",
  "prop.removed:prop.long",
  "prop.removed:mean.fidelity",
  "prop.removed:color.sim",
  "prop.removed:corolla.sim",
  "prop.removed:prop.delph"
)
N <- as.list(seq(1:(length(vars)-1)))
COMB <- sapply(N, function(m) combn(x=vars[2:length(vars)], m))
COMB2 <- list()
k=0
for(i in seq(COMB)){
  tmp <- COMB[[i]]
  for(j in seq(ncol(tmp))){
    k <- k + 1
    COMB2[[k]] <- formula(paste(
      "viable",
      "~",
      paste(tmp[,j], collapse=" + "),
      "+ (1|site/delph.plant.num/delph.flower.num) + (1|year)"))
  }
}

# Running glmmTMB for each formula and using glance to isolate AIC values
res <- vector(mode = "list", length(COMB2))
suppressWarnings(for(i in seq(COMB2)){
  res[[i]] <- try(glance(try(glmmTMB(COMB2[[i]], data=dat, family = nbinom2, zi = ~.))))
```

```

})

# Add Model ID column
ID <- c(1:length(res))
res2 <- mapply(cbind, res, "Model" = ID, SIMPLIFY = F)

# Removing models that failed to converge
filt <- Filter(function(x) length(x) > 4, res2)

# Extracting AIC
filt.df <- data.frame(matrix(unlist(filt), nrow = length(filt), byrow = T))
filt.df[,c('X1', 'X2', 'X4', 'X5')] <- list(NULL)
names(filt.df)[names(filt.df) == 'X3'] <- 'AIC'
names(filt.df)[names(filt.df) == 'X6'] <- 'Model'

# Arranging AICs in increasing order
filt.df <- arrange(filt.df, AIC)
filt.df$delta <- filt.df$AIC - filt.df[1,1]

# Filter out only model outputs with delta < 2
filt.2 <- filt.df[filt.df$delta < 2,]

# Taking the Model IDs from the "best" models (delta < 2) and connecting them with the actual formulas
forms <- data.frame(matrix(unlist(COMB2), nrow = length(COMB2), byrow = T))
colnames(forms) <- "Formula"

forms$Model <- c(1:length(res))
forms$Model <- as.numeric(forms$Model)

mods <- merge(filt.2, forms, by = "Model")
mods <- arrange(mods, AIC)
mods$Formula <- as.character(mods$Formula)

mods %>%
  kable %>%
  kable_styling("striped", full_width = T)

```


Model	AIC	delta	Formula
5147	6317.690	0.0000000	viable ~ bomabund.scale + prop.long + prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + prop.removed:mean.fidelity + (1 site/delph.plant.num/delph.flower.num) + (1 year)
5812	6317.992	0.3012489	viable ~ prop.removed + bomabund.scale + prop.long + mean.fidelity + prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + (1 site/delph.plant.num/delph.flower.num) + (1 year)
6605	6318.005	0.3143809	viable ~ bomabund.scale + prop.long + mean.fidelity + prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + prop.removed:mean.fidelity + (1 site/delph.plant.num/delph.flower.num) + (1 year)
1406	6318.274	0.5840295	viable ~ prop.removed + mean.fidelity + color.sim + corolla.sim + prop.removed:bomabund.scale + (1 site/delph.plant.num/delph.flower.num) + (1 year)
7103	6318.555	0.8644367	viable ~ prop.removed + bomabund.scale + prop.long + mean.fidelity + prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + prop.removed:prop.delph + (1 site/delph.plant.num/delph.flower.num) + (1 year)
6604	6318.711	1.0210052	viable ~ bomabund.scale + prop.long + mean.fidelity + prop.delph + color.sim + corolla.sim + prop.removed:bomabund.scale + prop.removed:prop.long + (1 site/delph.plant.num/delph.flower.num)

Model Averaging

Because we do not have an obvious “best” model after model selection, we average all models with ($\Delta AIC < 2$).

```
form.list <- as.list(rep(NA, times = nrow(mods)))
for(i in 1:nrow(mods)) {
  form.list[[i]] = glmmTMB(as.formula(mods$Formula[i]), data = dat, family = nbinom2, zi = ~.)
}

# Model average
top.avg <- model.avg(form.list)
top.avg
```

```
##
## Call:
## model.avg(object = form.list)
##
## Component models:
## '1/2/3/5/6/8/9/10/12/13/15/16/19/20'
## '2/3/4/7/9/10/11/14/15/19'
## '1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/19'
## '1/2/3/4/5/6/8/9/10/11/12/13/15/16/19/20'
## '1/2/3/5/6/8/9/10/12/13/15/18/19/22'
## '1/2/3/4/5/6/7/8/9/10/11/12/13/14/15/17/19/21'
## '1/2/3/4/5/6/8/9/10/11/12/13/15/18/19/22'
## '1/2/3/5/6/8/9/10/12/13/15/16/17/19/20/21'
## '2/3/4/6/7/9/10/11/13/14/15/19'
## '2/3/4/9/10/11/15/16/19/20'
## '1/2/3/4/6/7/8/9/10/11/13/14/15/19'
## '1/2/3/4/7/8/9/10/11/14/15/19'
## '1/2/3/5/6/7/8/9/10/12/13/14/15/19'
## '1/2/3/4/6/8/9/10/11/13/15/16/19/20'
## '2/3/4/5/6/7/9/10/11/12/13/14/15/19'
## '1/2/3/4/5/6/8/9/10/11/12/13/15/16/17/19/20/21'
##
## Coefficients:
##      cond((Int)) cond(bomabund.scale) cond(prop.long) cond(prop.delph)
## full      2.277258      0.1503960      -0.6648009      0.4981928
## subset    2.277258      0.1981784      -0.8204219      0.7549813
##      cond(color.sim) cond(corolla.sim) cond(bomabund.scale:prop.removed)
## full      1.567684      0.7256633      0.1773814
## subset    1.567684      0.7256633      0.1773814
##      cond(mean.fidelity:prop.removed) zi((Int)) zi(bomabund.scale)
## full      0.1344819 -0.9821282      -0.1347679
## subset    0.3462848 -0.9821282      -0.1775850
##      zi(prop.long) zi(prop.delph) zi(color.sim) zi(corolla.sim)
## full      -1.339809      0.4468753      -0.609255      1.657248
## subset    -1.653440      0.6772126      -0.609255      1.657248
##      zi(bomabund.scale:prop.removed) zi(mean.fidelity:prop.removed)
## full      -3.856683      1.627521
## subset    -3.856683      4.190793
##      cond(prop.removed) cond(mean.fidelity) cond(bomabund.scale:prop.removed)
## full      0.06727994      -0.4975555      0.1773814
## subset    0.13917570      -0.6895550      0.1773814
##      zi(prop.removed) zi(mean.fidelity) zi(bomabund.scale:prop.removed)
```

```
## full          1.744890          2.121317          -3.856683
## subset        3.609489          2.939902          -3.856683
##      cond(mean.fidelity:prop.removed) zi(mean.fidelity:prop.removed)
## full          0.1344819          1.627521
## subset        0.3462848          4.190793
##      cond(prop.long:prop.removed) zi(prop.long:prop.removed)
## full          0.04502911          0.5281062
## subset        0.35116852          4.1185418
##      cond(prop.delph:prop.removed) zi(prop.delph:prop.removed)
## full          -0.09355757          -0.6553144
## subset        -0.59690857          -4.1809850
##      cond(prop.delph:prop.removed) zi(prop.delph:prop.removed)
## full          -0.09355757          -0.6553144
## subset        -0.59690857          -4.1809850
```

Supporting Analysis

Effect of Manipulation on Bombus Community

We test whether our manipulation has a significant impact on *Bombus* abundance.

```
bombus.summary <- dat %>%
  group_by(site, state, year) %>%
  summarise(prop.removed = unique(prop.removed),
            bombus.abund = unique(bombus.abund),
            prop.long = unique(prop.long))

rem.abund <- glmmTMB(bombus.abund ~ prop.removed + (1|site) + (1|year), data = bombus.summary, family =
summary(rem.abund)

## Family: poisson ( log )
## Formula:      bombus.abund ~ prop.removed + (1 | site) + (1 | year)
## Data: bombus.summary
##
##      AIC      BIC   logLik deviance df.resid
##    302.5    307.8   -147.2    294.5      24
##
## Random effects:
##
## Conditional model:
##   Groups Name      Variance Std.Dev.
##   site   (Intercept) 0.26066  0.5105
##   year   (Intercept) 0.01764  0.1328
## Number of obs: 28, groups:  site, 14; year, 3
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    4.04368    0.16095  25.124 < 2e-16 ***
## prop.removed  -0.46091    0.09732  -4.736 2.18e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The manipulation significantly reduces the abundance of bees in the community compared to the control.

We also run the same analysis to see whether our manipulation significantly impacted the proportion of

long-tongued bees in the community.

```
tongue.test <- glmmTMB(prop.long ~ state + (1|site) + (1|year), data = bombus.summary, family = binomial)
summary(tongue.test)
```

```
## Family: binomial ( logit )
## Formula: prop.long ~ state + (1 | site) + (1 | year)
## Data: bombus.summary
##
##      AIC      BIC   logLik deviance df.resid
##    43.4    48.7   -17.7    35.4      24
##
## Random effects:
##
## Conditional model:
## Groups Name      Variance Std.Dev.
## site   (Intercept) 7.627e-10 2.762e-05
## year   (Intercept) 1.580e-11 3.975e-06
## Number of obs: 28, groups: site, 14; year, 3
##
## Conditional model:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    0.5899    0.5579   1.057   0.29
## statemanipulation 0.2461    0.8062   0.305   0.76
```

There is no significant effect of the manipulation on the proportion of the *Bombus* community with long tongues.

Annual changes in focal species

D. barbeyi

```
db.abund <- dat %>%
  mutate(year = as.character(year)) %>%
  group_by(site, year) %>%
  summarise(num.delph = unique(num.delph),
            prop.delph = unique(prop.delph))

dbabund.test <- aov(num.delph ~ year, family = poisson, data = db.abund)
summary(dbabund.test)

##              Df    Sum Sq Mean Sq F value Pr(>F)
## year           2 30800614 15400307   3.582 0.0634 .
## Residuals     11 47292412  4299310
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

propdb.test <- aov(prop.delph ~ year, family = binomial, data = db.abund)
summary(propdb.test)

##              Df Sum Sq Mean Sq F value Pr(>F)
## year           2  0.1911  0.09553   2.985 0.0921 .
## Residuals     11  0.3520  0.03200
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Year has a marginally significant effect on *D. barbeyi* abundance and on the proportion of *D. barbeyi* at each site.

Bombus

```
bom.abund <- dat %>%
  mutate(year = as.character(year)) %>%
  group_by(site, state, year) %>%
  summarise(bombus.abund = unique(bombus.abund))

bomabund.test <- aov(bombus.abund ~ year, data = bom.abund)
summary(bomabund.test)
```

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## year           2  10009     5004   3.178 0.0589 .
## Residuals     25  39364     1575
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Year has a marginally significant effect on the overall abundance of *Bombus* in our study system.

Figures

Code to produce all figures included in the manuscript.

Model Predictions

Generate model predictions so that our plots will show the trends identified by our model selection and averaging.

```
pred.dat <- dat %>%
  mutate(viable = predict(top.avg,
                           dat,
                           type = "zlink", # zero-inflated results
                           full = T,
                           allow.new.levels = F))

# Data for geom_pointrange
pred.sum <- pred.dat %>%
  group_by(site, state, prop.removed, bombus.abund, prop.long, mean.fidelity, prop.delph, corolla.sim) %>%
  summarise(mean.viable = mean(exp(viable), na.rm = T),
            lower.ci = confidence_interval(na.omit(exp(viable)), 0.95)[[1]],
            higher.ci = confidence_interval(na.omit(exp(viable)), 0.95)[[2]])

# Manipulation bins - helpful for plotting a continuous interaction term
manip.bins <- data.frame(site = (pred.sum %>% filter(state == "manipulation"))$site,
                        prop.removed = (pred.sum %>% filter(state == "manipulation"))$prop.removed) %>%
  mutate(bin = ifelse(prop.removed < 0.5, "low", "high"))

manip.low <- manip.bins %>% filter(bin == "low")
manip.high <- manip.bins %>% filter(bin == "high")

pred.dat <- pred.dat %>%
  left_join(manip.bins, by = c("site", "prop.removed")) %>%
```


site	year	prop.delph	prop.removed	species.removed	tongue.length
Brush Creek Fork	2011	0.5666087	0.5200000	Bombus appositus	long
Deer Creek Gate 2	2011	0.1390160	0.3571429	Bombus nevadensis	long
Gothic Road Gate.2011	2011	0.1711975	0.4594595	Bombus appositus	long
Gothic Town	2011	0.4381538	0.3000000	Bombus appositus	long
Teocali View	2011	0.1911608	0.3750000	Bombus nevadensis	long
Back of Baldy Bend	2013	0.0405495	0.5000000	Bombus flavifrons	medium
Emerald Lake 2	2013	0.0726925	0.4883721	Bombus kirbiellus	long
Gothic Road 2	2013	0.0874804	0.6009615	Bombus appositus	long
Judd Falls TH 2	2013	0.1918133	0.4754098	Bombus flavifrons	medium
Past Rustlers	2013	0.1676446	0.4266667	Bombus flavifrons	medium
Brush Creek Sign	2014	0.4776090	0.6562500	Bombus appositus	long
Cold Springs Ranch	2014	0.7139811	0.6176471	Bombus appositus	long
Emerald Lake	2014	0.1638344	0.5263158	Bombus kirbiellus	long
Gothic Road Gate.2014	2014	0.2253277	0.5172414	Bombus appositus	long

```
# print(xtable(table2, type = "latex"), include.rownames = F, file = "../Figures/SiteSumTable.tex")
```

Table 3

```
# Run all of the models that the model selection identified
## Create a list with all of the model estimates extracted using tidy
## Filter only the zero-inflated estimates
top.mods <- vector(mode = "list", length(mods$Formula))
for(i in seq(mods$Formula)){
  top.mods[[i]] <-
    tidy(glmTMB(as.formula(mods$Formula[[i]]), data = dat, family = nbinom2, zi = ~.)) %>%
    filter(component == "zi") %>%
    mutate(var = ifelse(term == "sd_(Intercept)", group, term)) %>%
    dplyr::select(var, estimate) %>%
    pivot_wider(names_from = var, values_from = estimate)
}

# Turn the list of estimates for each model into a single dataframe
modselect.results <- data.frame()
temp.df <- data.frame()
for(i in seq(top.mods)){
  temp.df <- as.data.frame(unlist(top.mods[[i]]))
  colnames(temp.df) <- "estimate"
  temp.df$var <- rownames(temp.df)
  temp.wide <- pivot_wider(temp.df, names_from = var, values_from = estimate)
  modselect.results <- bind_rows(modselect.results, temp.wide)
}
```

```

# Add model number to this dataframe
modselect.results$Model <- mods$Model

# Merge the two formats of interaction columns
## Don't need to do this for prop.removed:prop.long since all the necessary info is already in one column
modselect.results <- modselect.results %>%
  mutate(`prop.removed:prop.delph` = ifelse(is.na(`prop.removed:prop.delph`), `prop.delph:prop.removed`)

# Reorganize this table and change variable naming convention
modselect.table <- modselect.results %>%
  dplyr::select(Model,
    `(Intercept)`,
    prop.removed,
    bomabund.scale,
    mean.fidelity,
    prop.long,
    prop.delph,
    color.sim,
    corolla.sim,
    `prop.removed:bomabund.scale`,
    `prop.long:prop.removed`,
    `prop.removed:mean.fidelity`,
    `prop.removed:prop.delph`)

modselect.tex <- modselect.table %>%
  left_join(mods, by = "Model") %>%
  dplyr::select(AIC,
    delta,
    prop.removed,
    bomabund.scale,
    prop.long,
    mean.fidelity,
    prop.delph,
    color.sim,
    corolla.sim,
    `prop.removed:bomabund.scale`,
    `prop.long:prop.removed`,
    `prop.removed:mean.fidelity`,
    `prop.removed:prop.delph`) %>%
  mutate(delta = as.character(format(delta, scientific = F, digits = 2))) %>%
  mutate_if(is.numeric, round, digits = 2) %>%
  replace(is.na(.), "")

modselect.tex %>%
  kable() %>%
  kable_styling("striped", full_width = F) %>%
  scroll_box(width = "500px", height = "200px")

# print(xtable(modselect.tex, type = "latex", file = "../Data/ModelSelectionTable.tex")

```


AIC	delta	prop.removed	bomabund.scale	prop.long	mean.fidelity	prop.delph	color.sim	corolla.sim	pro
6317.69	0.00		-0.29	-1.1		0.18	-0.45	1.46	
6317.99	0.30	3.39	-0.12	-1.52	3.44	0.51	-0.63	1.78	-3.
6318.00	0.31		-0.17	-1.64	2.58	0.58	-0.67	1.70	
6318.27	0.58	3.19			2.89		-0.49	1.58	-3.
6318.55	0.86	5.73	0.06	-3.01	3.65	1.96	-0.83	1.99	-4.
6318.71	1.02		-0.16	-1.8	2.77	0.57	-0.49	1.73	
6318.73	1.04		-0.3	-1.19		0.13	-0.27	1.47	
6318.88	1.19		-0.16	-2.45		1.46	-0.67	1.62	
6319.14	1.45	3.51		-1.49	3.38		-1.01	1.80	-3.
6319.28	1.59	3.35	-0.14	-1.38	3.2		-0.83	1.77	-3.
6319.44	1.75				2.1		-0.61	1.50	-3.
6319.50	1.81	3.07	-0.31	-0.69		-0.04	-0.37	1.46	-4.
6319.53	1.84		-0.19	-1.48	2.33		-0.90	1.68	
6319.54	1.85	2.97	-0.22		2.63		-0.25	1.55	-3.
6319.66	1.97	3.53		-1.62	3.59	0.53	-0.78	1.81	-3.
6319.68	1.99		-0.07	-2.74	2.32	1.64	-0.84	1.80	

Table 4

```

# Some formatting to make the table look nice
avg.table <- as.data.frame(summary(top.avg)[9]) %>%
  rownames_to_column(var = "variable") %>%
  filter(str_detect(variable, "zi"),
         str_detect(variable, "1", negate = T))

avg.rownames <- data.frame(variable = avg.table$variable,
                           formatted = c("Intercept",
                                           "BA",
                                           "PL",
                                           "PD",
                                           "CL",
                                           "CR",
                                           "PR:BA",
                                           "PR:MF",
                                           "PR",
                                           "MF",
                                           "PR:PL",
                                           "PR:PD"))

avg.table <- avg.table %>%
  left_join(avg.rownames, by = "variable") %>%
  dplyr::select(-variable) %>%
  column_to_rownames(var = "formatted") %>%
  rename("Estimate" = "coefmat.full.Estimate",
         "Std.Error" = "coefmat.full.Std..Error",
         "Adjusted.SE" = "coefmat.full.Adjusted.SE",
         "z-value" = "coefmat.full.z.value",
         "p-value" = "coefmat.full.Pr...z..")

avgtable.tex <- avg.table %>%
  mutate(arrange.key = c(1, 3, 4, 6, 7, 8, 9, 11, 2, 5, 10, 12)) %>%

```

```

  arrange(arrange.key) %>%
  dplyr::select(-arrange.key)

avgtable.tex %>%
  kable() %>%
  kable_styling("striped", full_width = T)

# print(xtable(avgtable.tex, type = "latex"), row.names = F, file = "../Data/AverageTable.tex")

```

	Estimate	Std.Error	Adjusted.SE	z-value	p-value
Intercept	-0.9821282	1.8617146	1.8623667	0.5273549	0.5979472
PR	1.7448895	2.0129621	2.0131175	0.8667599	0.3860736
BA	-0.1347679	0.2775273	0.2776843	0.4853276	0.6274440
PL	-1.3398090	1.3656800	1.3662575	0.9806416	0.3267695
MF	2.1213168	1.8793896	1.8799820	1.1283708	0.2591634
PD	0.4468753	1.0042241	1.0046728	0.4447968	0.6564666
CL	-0.6092550	1.1370887	1.1378416	0.5354480	0.5923401
CR	1.6572481	0.5288371	0.5291694	3.1317912	0.0017374
PR:BA	-3.8566826	0.9778549	0.9784187	3.9417506	0.0000809
PR:PL	0.5281062	1.4325124	1.4325868	0.3686382	0.7123974
PR:MF	1.6275205	2.2400225	2.2401893	0.7265103	0.4675260
PR:PD	-0.6553144	1.8626240	1.8630337	0.3517459	0.7250289

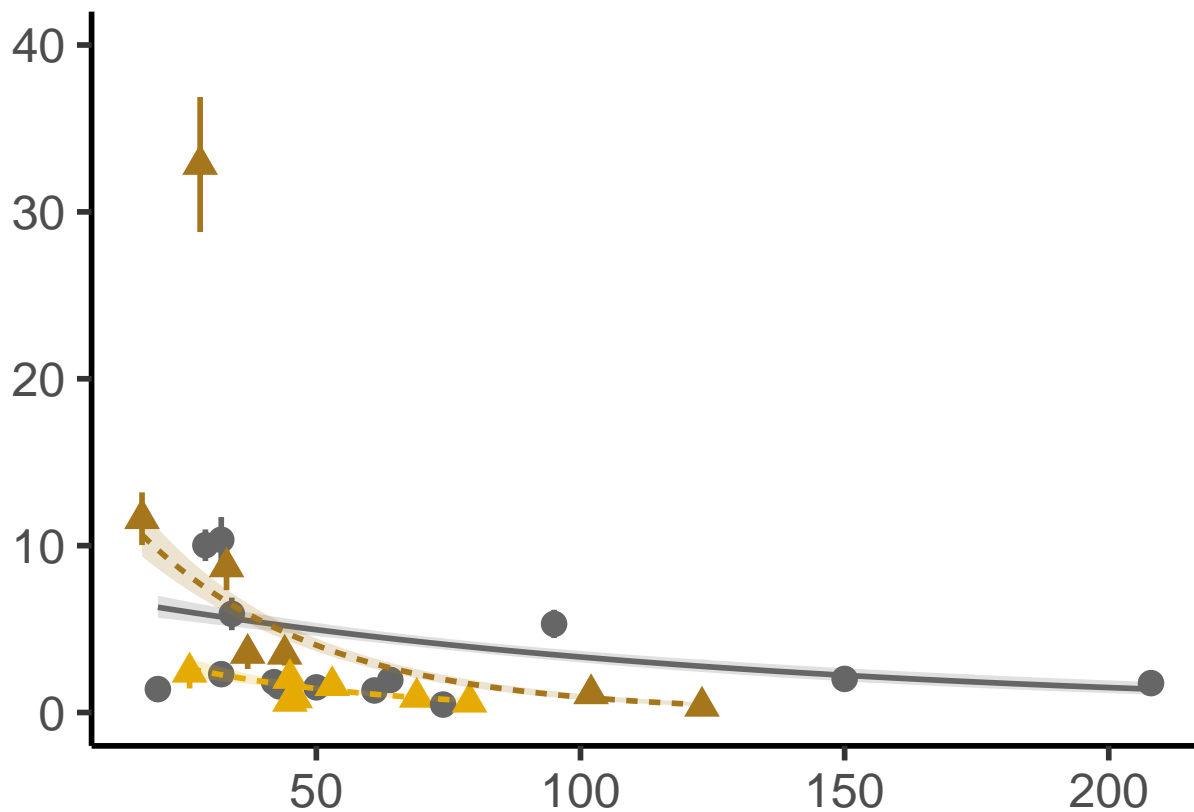
Figure 1

```

abund.all <- ggplot() +
  geom_pointrange(data = pred.sum,
    aes(x = bombus.abund,
      y = mean.viable,
      ymax = higher.ci,
      ymin = lower.ci,
      col = bin,
      shape = state),
    size = 1,
    linewidth = 1) +
  geom_smooth(data = pred.dat,
    aes(x = bombus.abund,
      y = exp(viable),
      col = bin,
      fill = bin,
      linetype = state),
    method = "glm.nb",
    se = T,
    alpha = 0.2) +
  scale_color_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  scale_fill_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  labs(x = ~italic(Bombus)~" Abundance",
    y = "Viable Seeds") +
  theme_classic(base_size = 22) +
  theme(axis.title = element_blank(),
    legend.position = "none") +
  lims(y = c(0, 40))

```

```
abund.all
```



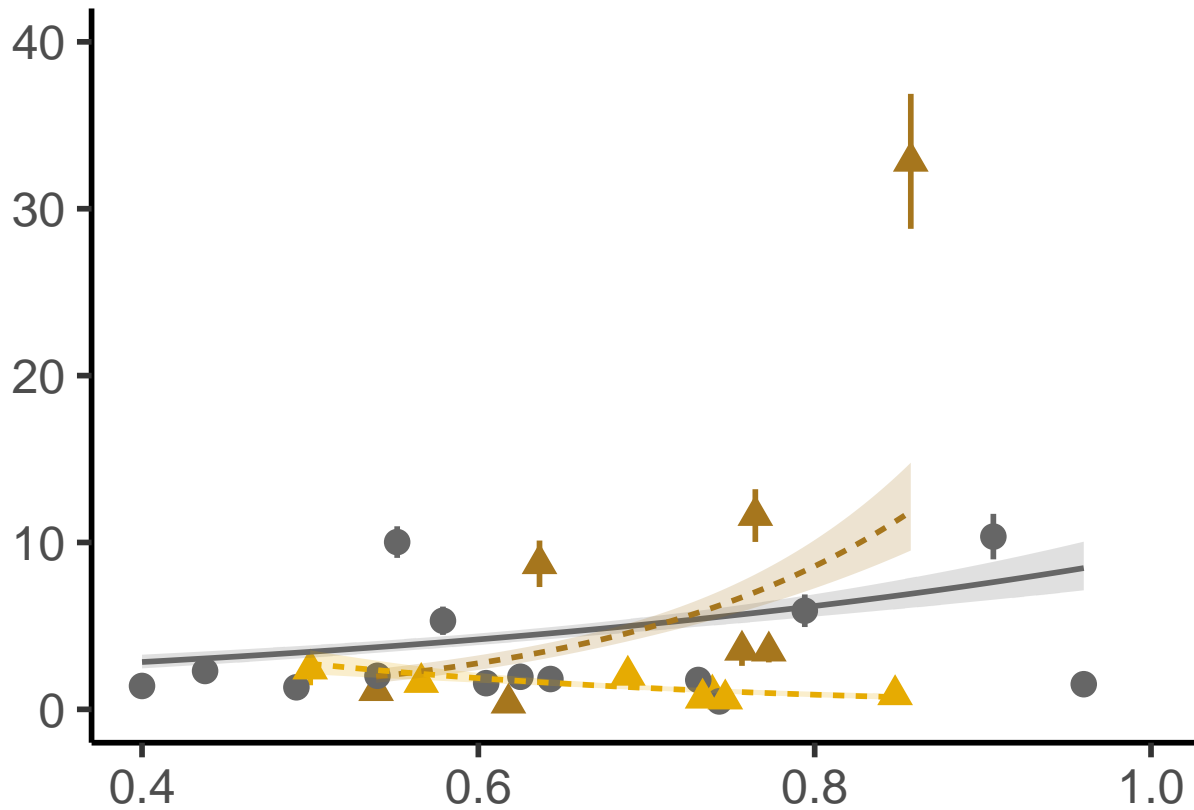
```
# ggsave("../Figures/bomabund.png", plot = abund.all, width = 7, height = 10, units = "in")
```

Figure 2

```
tongue.all <- ggplot() +
  geom_pointrange(data = pred.sum,
    aes(x = prop.long,
        y = mean.viable,
        ymax = higher.ci,
        ymin = lower.ci,
        col = bin,
        shape = state),
    size = 1,
    linewidth = 1) +
  geom_smooth(data = pred.dat,
    aes(x = prop.long,
        y = exp(viable),
        col = bin,
        fill = bin,
        linetype = state),
    method = "glm.nb",
    se = T,
    alpha = 0.2) +
  scale_color_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  scale_fill_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  theme_classic(base_size = 22) +
```

```
theme(axis.title = element_blank(),
      legend.position = "none") +
lims(x = c(0.4, 1),
     y = c(0, 40))
```

```
tongue.all
```



```
# ggsave("../Figures/tongue-length.png", plot = tongue.all, width = 7, height = 10, units = "in")
```

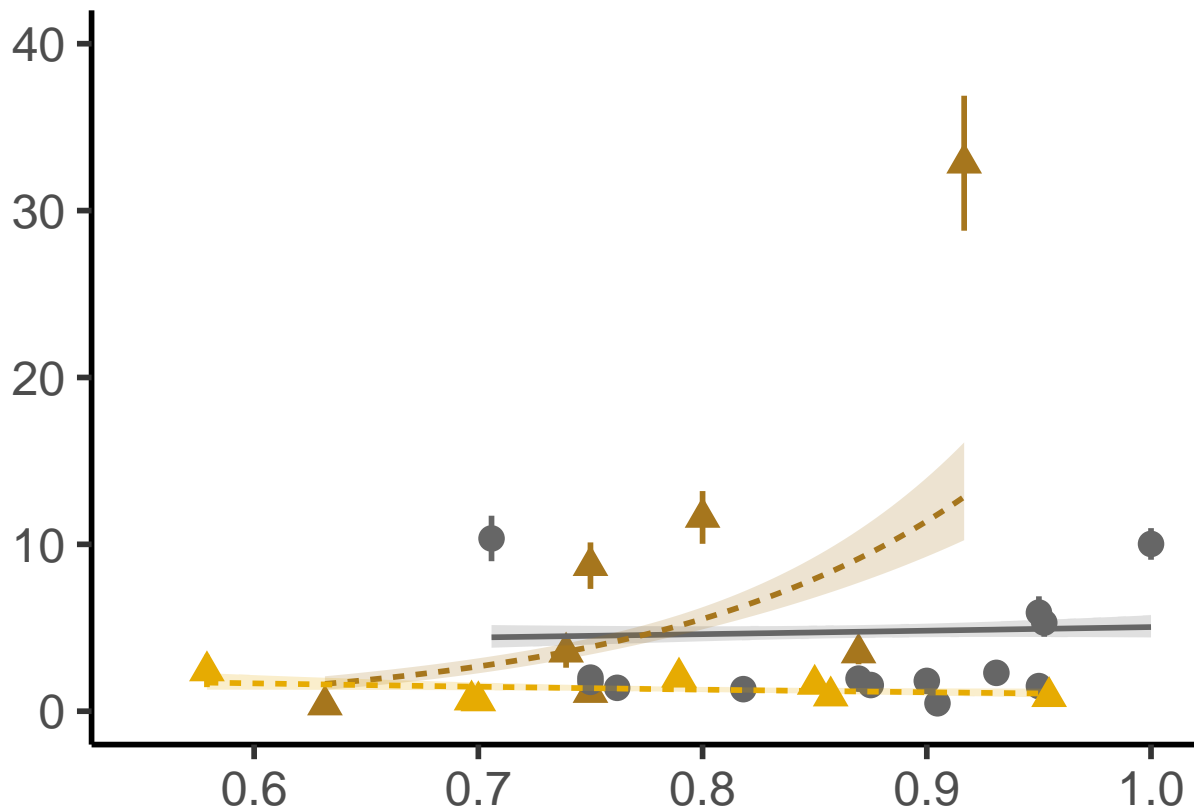
Figure 3

```
fidelity.all <- ggplot() +
  geom_pointrange(data = pred.sum,
                 aes(x = mean.fidelity,
                     y = mean.viable,
                     ymax = higher.ci,
                     ymin = lower.ci,
                     col = bin,
                     shape = state),
                 size = 1,
                 linewidth = 1) +
  geom_smooth(data = pred.dat,
             aes(x = mean.fidelity,
                 y = exp(viable),
                 col = bin,
                 fill = bin,
                 linetype = state),
```

```

    method = "glm.nb",
    se = T,
    alpha = 0.2) +
  scale_color_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  scale_fill_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  theme_classic(base_size = 22) +
  theme(axis.title = element_blank(),
        legend.position = "none") +
  lims(x = c(0.55, 1),
        y = c(0, 40))
fidelity.all

```



```

# ggsave("../Figures/fidelity-all.png", plot = fidelity.all, width = 7, height = 10, units = "in")

```

Supplemental Information

These tables and figures are not included in the manuscript but are made available here in case of interest. Many are in response to reviewer comments.

Bombus Tongue Lengths

```

tongue.dat %>%
  kable() %>%
  kable_styling("striped", full_width = F)

```

bombus	tongue.length
Bombus appositus	long
Bombus bifarius	medium
Bombus californicus	long
Bombus flavifrons	medium
Bombus frigidus	medium
Bombus kirbiellus	long
Bombus mixtus	medium
Bombus nevadensis	long
Bombus occidentalis	short
Bombus rufocinctus	short
Bombus sylvicola	medium

Site-level *Bombus* abundance

```
bom <- read.csv("../Data/bombus-summary.csv", stringsAsFactors = F)

bom.community <- bom %>%
  pivot_wider(names_from = "bee.sp",
              values_from = "n",
              values_fill = 0)

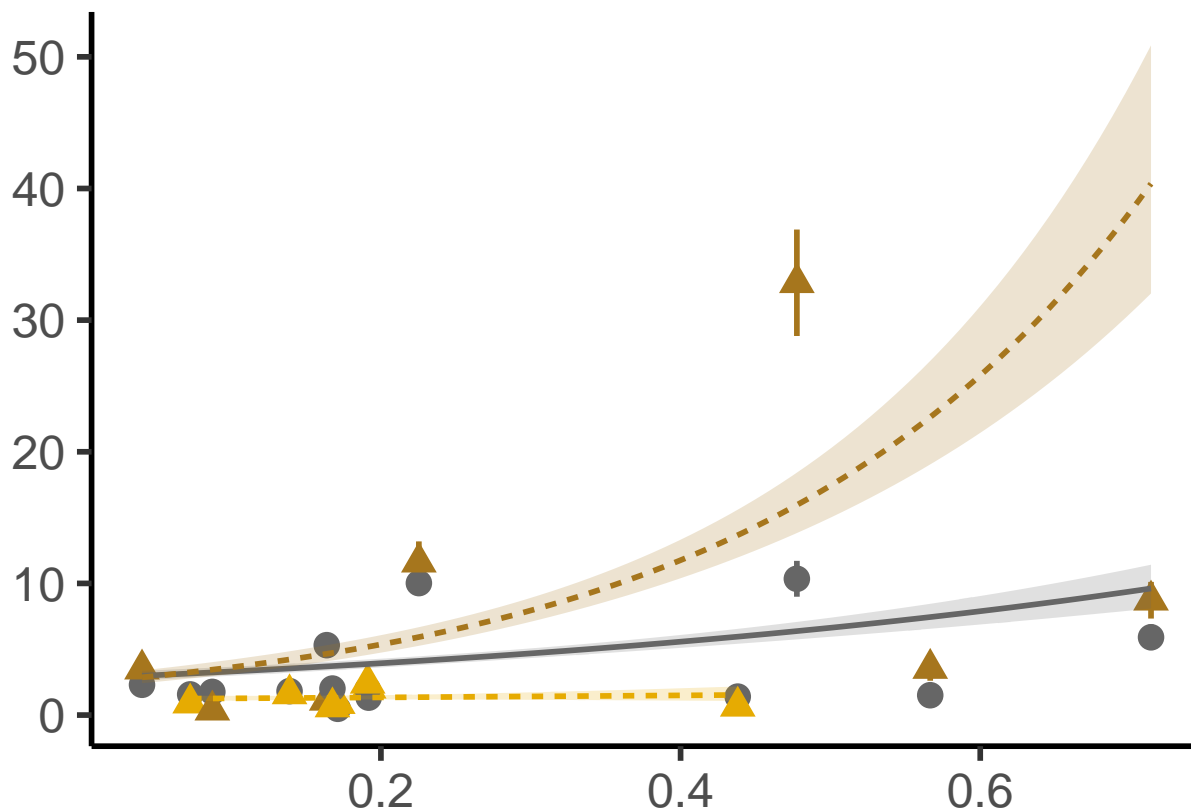
bom.community %>%
  kable() %>%
  kable_styling("striped", full_width = F) %>%
  scroll_box(width = "500px", height = "200px")
```

Relationship between *D. barbeyi* relative abundance and seed production

```
delph.plot <- ggplot() +
  geom_pointrange(data = pred.sum,
                 aes(x = prop.delph,
                     y = mean.viable,
                     ymax = higher.ci,
                     ymin = lower.ci,
                     col = bin,
                     shape = state),
                 size = 1,
                 linewidth = 1) +
  geom_smooth(data = pred.dat,
              aes(x = prop.delph,
                  y = exp(viable),
                  col = bin,
                  fill = bin,
                  linetype = state),
              method = "glm.nb",
              se = T,
              alpha = 0.2) +
  scale_color_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  scale_fill_manual(values = brewer.pal(8, "Dark2")[8:6]) +
  theme_classic(base_size = 22) +
  theme(axis.title = element_blank(),
```

site	state	Bombus appositus	Bombus bifarius	Bombus flavifrons	Bombus kirbiellus
Back of Baldy Bend	control	10	2	16	2
Back of Baldy Bend	manipulation	27	0	10	6
Brush Creek Fork	control	26	0	1	0
Brush Creek Fork	manipulation	8	2	6	0
Brush Creek Sign	control	21	3	0	0
Brush Creek Sign	manipulation	12	3	0	0
Cold Springs Ranch	control	21	7	0	0
Cold Springs Ranch	manipulation	13	11	1	0
Deer Creek Gate 2	control	11	9	5	0
Deer Creek Gate 2	manipulation	15	12	9	0
Emerald Lake	control	4	5	31	50
Emerald Lake	manipulation	19	9	32	36
Emerald Lake 2	control	5	5	12	21
Emerald Lake 2	manipulation	17	7	11	33
Gothic Road 2	control	125	16	28	3
Gothic Road 2	manipulation	35	8	29	7
Gothic Road Grate.2011	control	34	19	0	0
Gothic Road Grate.2011	manipulation	7	6	0	0
Gothic Road Grate.2014	control	15	8	0	0
Gothic Road Grate.2014	manipulation	5	3	0	0
Gothic Town	control	6	7	3	0
Gothic Town	manipulation	5	5	6	1
Judd Falls TH 2	control	28	0	29	1
Judd Falls TH 2	manipulation	28	1	13	0
Past Rustlers	control	63	4	64	4
Past Rustlers	manipulation	36	6	14	8
Teocali View	control	15	17	6	0
Teocali View	manipulation	9	6	7	0

```
legend.position = "none")
delph.plot
```



Session Info

```
sessionInfo()
```

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods   base
##
## other attached packages:
## [1] RColorBrewer_1.1-3 MASS_7.3-54      performance_0.8.0 DHARMa_0.4.4
## [5] xtable_1.8-4      MuMIn_1.43.17    glmmTMB_1.1.3     broom.mixed_0.2.7
## [9] kableExtra_1.3.4  forcats_1.0.0    stringr_1.5.0     dplyr_1.1.0
## [13] purrr_1.0.1       readr_2.1.2      tidyr_1.3.0       tibble_3.1.8
## [17] ggplot2_3.4.0     tidyverse_1.3.1
```



```

##
## loaded via a namespace (and not attached):
## [1] nlme_3.1-153      fs_1.6.1          lubridate_1.9.1
## [4] insight_0.14.5    webshot_0.5.2     httr_1.4.4
## [7] numDeriv_2016.8-1.1 tools_4.1.2       TMB_1.7.22
## [10] backports_1.4.1   utf8_1.2.3        R6_2.5.1
## [13] mgcv_1.8-38       DBI_1.1.1          colorspace_2.1-0
## [16] withr_2.5.0       tidysselect_1.2.0 emmeans_1.7.4-1
## [19] compiler_4.1.2    cli_3.6.0          rvest_1.0.2
## [22] xml2_1.3.3        sandwich_3.0-1     labeling_0.4.2
## [25] scales_1.2.1      mvtnorm_1.1-3      systemfonts_1.0.3
## [28] digest_0.6.31     minqa_1.2.5        rmarkdown_2.11
## [31] svglite_2.0.0     pkgconfig_2.0.3    htmltools_0.5.4
## [34] lme4_1.1-31       highr_0.10         dbplyr_2.1.1
## [37] fastmap_1.1.0     rlang_1.0.6        readxl_1.3.1
## [40] rstudioapi_0.13   farver_2.1.1       generics_0.1.3
## [43] zoo_1.8-9         jsonlite_1.8.4     magrittr_2.0.3
## [46] Matrix_1.3-4      Rcpp_1.0.10        munsell_0.5.0
## [49] fansi_1.0.4       lifecycle_1.0.3    stringi_1.7.12
## [52] multcomp_1.4-19   yaml_2.3.7         grid_4.1.2
## [55] crayon_1.5.2      lattice_0.20-45    haven_2.4.3
## [58] splines_4.1.2     hms_1.1.1          knitr_1.42
## [61] pillar_1.8.1      boot_1.3-28        estimability_1.3
## [64] codetools_0.2-18 stats4_4.1.2        reprex_2.0.1
## [67] glue_1.6.2        evaluate_0.20      modelr_0.1.8
## [70] vctrs_0.5.2       nloptr_2.0.3       tzdb_0.2.0
## [73] cellranger_1.1.0  gtable_0.3.1       assertthat_0.2.1
## [76] xfun_0.37         broom_1.0.3        coda_0.19-4
## [79] survival_3.2-13   viridisLite_0.4.1  timechange_0.2.0
## [82] TH.data_1.1-1     ellipsis_0.3.2

```