Data analyses for Experiment 3

Scott Burgess

2024-09-15

This code analyzes the data associated with Experiment 3 and produces Figure 4, 5, and 6 in the manuscript:

Barnes DK, Burgess SC. Fitness consequences of marine larval dispersal: the role of neighborhood density, spatial arrangement, and genetic relatedness on survival, growth, reproduction, and paternity. *Journal of Evolutionary Biology*

Code finalized September 2024.

Any comments or error reporting, please contact Scott Burgess: sburgess@bio.fsu.edu

Load required libraries

```
library('tidyverse')
library('glmmTMB')
library('gridExtra')
library('emmeans')
library('DHARMa')
library('vegan')
```

```
sessionInfo()
```

```
## R version 4.4.0 (2024-04-24)
## Platform: aarch64-apple-darwin20
## Running under: macOS Sonoma 14.6.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.4-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] vegan_2.6-6.1   lattice_0.22-6  permute_0.9-7   DHARMa_0.4.6
##  [5] emmeans_1.10.2  gridExtra_2.3   glmmTMB_1.1.9   lubridate_1.9.3
##  [9] forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2
## [13] readr_2.1.5     tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.1
## [17] tidyverse_2.0.0
```

```
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4          generics_0.1.3      stringi_1.8.4
##  [4] lme4_1.1-35.3       hms_1.1.3           digest_0.6.35
##  [7] magrittr_2.0.3      estimability_1.5.1  evaluate_0.23
## [10] grid_4.4.0          timechange_0.3.0    mvtnorm_1.2-5
## [13] fastmap_1.2.0       Matrix_1.7-0        mgcv_1.9-1
## [16] fansi_1.0.6         scales_1.3.0        numDeriv_2016.8-1.1
## [19] cli_3.6.2           rlang_1.1.3         munsell_0.5.1
## [22] splines_4.4.0       withr_3.0.0         yaml_2.3.8
## [25] parallel_4.4.0      tools_4.4.0         tzdb_0.4.0
## [28] coda_0.19-4.1       nloptr_2.0.3        minqa_1.2.7
## [31] colorspace_2.1-0    boot_1.3-30         vctrs_0.6.5
## [34] R6_2.5.1            lifecycle_1.0.4     MASS_7.3-60.2
## [37] cluster_2.1.6       pkgconfig_2.0.3     pillar_1.9.0
## [40] gtable_0.3.5        Rcpp_1.0.12         glue_1.7.0
## [43] xfun_0.44           tidyselect_1.2.1    rstudioapi_0.16.0
## [46] knitr_1.47          xtable_1.8-4        htmltools_0.5.8.1
## [49] nlme_3.1-164        rmarkdown_2.28      TMB_1.9.11
## [52] compiler_4.4.0
```

Import data from Experiment 3

Phenotypic data

```
dat <- read.csv("Data/Experiment_3.csv")
```

Paternity data

```
Paternity_dat_0_9 <- read.table("Data/D0_9_final_Paternity.txt",header=T,sep=",")
```

Inferred fathers data

```
BestCluster_0_9 <- read.csv("Data/D0_9_final_BestCluster.csv",header=T)
```

Full sib family data

```
FSFamily_0_9 <- read.csv("Data/D0_9_final_BestFSFamily.csv")
```

Process phenotypic data

Only use the summed data over all time periods

```
dat <- dat %>% filter(Time_days=="all") %>% select(-Time_days, -Bifurcations, -Zooids)
```

Add ID for focal colonies

```
focals <- c("A","B","D","G")
dat$focal <- ifelse(dat$Position %in% focals, 1, 0)
```

Create a vector of treatment levels and set the order (for plotting)

```
treatment.vec <- c("alone","far","near","both")
dat$Treatment <- factor(dat$Treatment, levels=treatment.vec)
```

Process paternity data

```
# Add sample data to each data frame
brks <- seq(0,160,0.1)

add_sample_data <- function(d){
  d$MotherID <- rapply(strsplit(d$OffspringID,"_"),function(x) head(x,1))
  d$FatherID <- rapply(strsplit(d$InferredDad1,"_"),function(x) head(x,1))
  tmp <- dat[match(d$MotherID,dat$Colony),-1]
  names(tmp) <- paste0("Mother.",names(tmp))
  d <- cbind.data.frame(d,tmp)
  tmp <- dat[match(d$FatherID,dat$Colony),-1]
  names(tmp) <- paste0("Father.",names(tmp))
  d <- cbind.data.frame(d,tmp)

# Calculate euclidean distance between observed parents
  d$Distance <- NA
  for(i in 1:nrow(d)){
    d$Distance[i] <- dist(
      matrix(d[i,which(names(d) %in% c("Mother.X","Father.X","Mother.Y","Father.Y"))],2,2,byrow=T),meth
  }

  return(d)
}

Paternity_dat_0_9 <- add_sample_data(Paternity_dat_0_9)
```

Calculate the number of offspring sampled per mother

```
OffspringIDs <- BestCluster_0_9$OffspringID
tmp <- rapply(strsplit(OffspringIDs,"_"),function(x) head(x,1))

MotherID <- paste0(tmp,"_parent")

offspring_per_mother <- as.data.frame(table(MotherID))
offspring_per_mother
```

```
##      MotherID Freq
## 1  01b_parent   25
## 2  02a_parent   25
## 3  04a_parent   24
## 4  05b_parent   26
## 5  06a_parent   25
## 6  08a_parent   26
## 7  09a_parent   24
## 8  12b_parent   27
## 9  13b_parent   25
## 10 15b_parent   23
## 11 17a_parent   25
## 12 18a_parent   25
## 13 19b_parent   23
## 14 20a_parent   23
## 15 21a_parent   24
```

```
## 16 22a_parent   24
## 17 23a_parent   25
## 18 25b_parent   24
## 19 26a_parent   26
## 20 28a_parent   24
## 21 29b_parent   24
## 22 30a_parent   24
## 23 31a_parent   24
## 24 32a_parent   25
## 25 34b_parent   25
```

How many settlers were genotyped?

```
with(offspring_per_mother,table(Freq))
```

```
## Freq
## 23 24 25 26 27
##  3  9  9  3  1
```

```
nrow(offspring_per_mother) # mothers
```

```
## [1] 25
```

```
sum(offspring_per_mother$Freq) # offspring
```

```
## [1] 615
```

```
nrow(dat) # potential fathers
```

```
## [1] 32
```

Analyze Reproductive Output

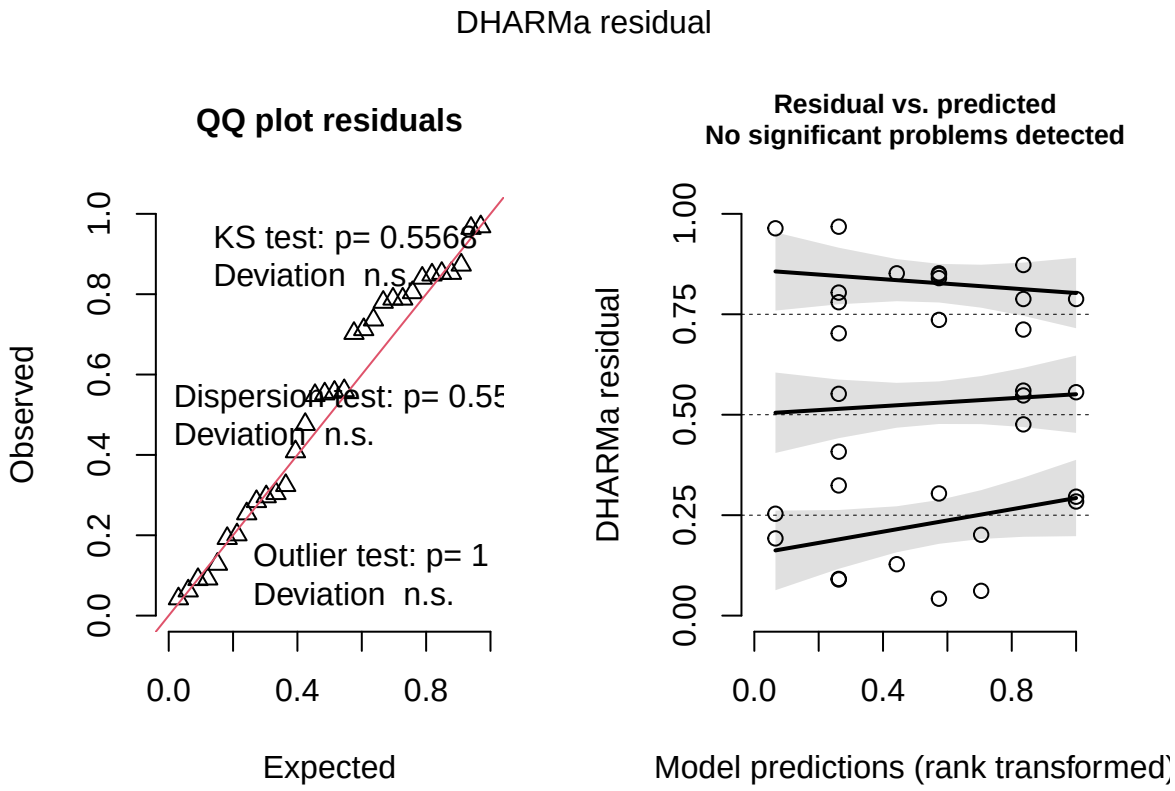
```
m1 <- glmmTMB(Offspring ~ Treatment + (1|Block),
              ziformula = ~.,
              data = dat,
              family = truncated_nbinom1)

m2 <- glmmTMB(Offspring ~ 1 + (1|Block),
              ziformula = ~.,
              data = dat,
              family = truncated_nbinom1)

result_reprod <- round(anova(m1, m2, test = "Chisq"), 3)
result_reprod
```

```
## Data: dat
## Models:
## m2: Offspring ~ 1 + (1 | Block), zi=~., disp=~1
```

```
## m1: Offspring ~ Treatment + (1 | Block), zi=~., disp=~1
##    Df    AIC    BIC  logLik deviance Chisq Chi Df Pr(>Chisq)
## m2  5 450.70 458.03 -220.35   440.70
## m1 11 454.55 470.68 -216.28   432.55 8.147      6      0.228
```

Check model fit

```
simulateResiduals(fittedModel = m1, plot = T)
```




```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DH
##
## Scaled residual values: 0.968 0.2535617 0.1921669 0.408 0.736 0.87259 0.09136761 0.852 0.324 0.20126
```

Get predicted values for plotting

```
predictions_reprod = data.frame(Treatment = treatment.vec)

p <- emmeans(m1, ~ Treatment,
             component = "cond",
             type = "response")

predictions_reprod$fit <- summary(p)$response
predictions_reprod$lwr <- summary(p)$asymp.LCL
predictions_reprod$upr <- summary(p)$asymp.UCL
```

```
nd <- expand.grid(Treatment = treatment.vec,
                  Block = rownames(coef(m1)$cond$Block))

block_effects_reprod <- data.frame(nd,
                                   fit = predict(m1,
                                                 newdata = nd,
```

```
                                                type="response",
                                                se=F))
```

Reproductive output was similar for individuals in all treatments ($\chi^2 = 8.147$, df = 6, p = 0.228).

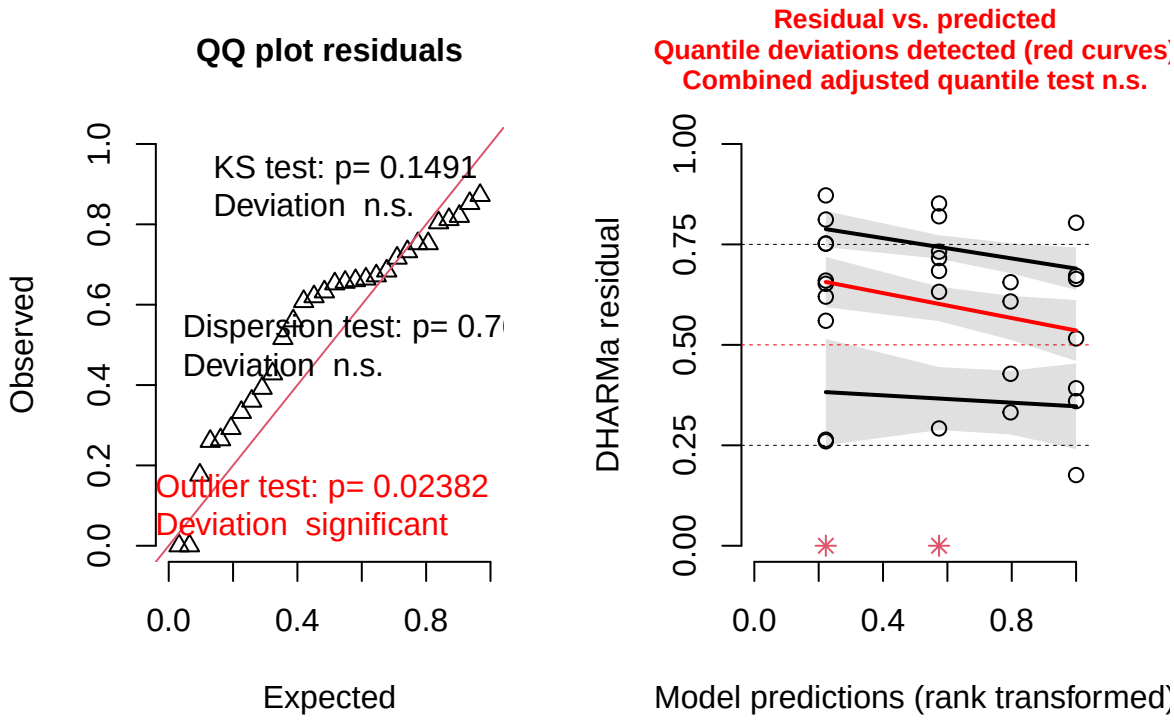
Analyze Relative Growth Rate

```
m1 <- glmmTMB(rgr40 ~ Treatment + (1|Block), data = dat)

m2 <- glmmTMB(rgr40 ~ 1 + (1|Block), data = dat)

result_rgr <- round(anova(m1, m2, test = "Chisq"), 3)
result_rgr
```

```
## Data: dat
## Models:
## m2: rgr40 ~ 1 + (1 | Block), zi=~0, disp=~1
## m1: rgr40 ~ Treatment + (1 | Block), zi=~0, disp=~1
##    Df     AIC     BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m2  3 -141.96 -137.76 73.980  -147.96
## m1  6 -136.90 -128.49 74.448  -148.90 0.936      3      0.817
```

Check model fit

```
simulateResiduals(fittedModel = m1, plot = T)
```




```
## Object of Class DHARMa with simulated residuals based on 250 simulations with refit = FALSE . See ?DH
##
## Scaled residual values: 0.62 0.652 0.752 0.264 0.672 0.632 0 0.392 0.26 0.684 0 0.872 0.332 0.516 0.7
```

Check influence of outliers

```
outliers <- dat[which(resid(m1) < -0.06), 'Colony']
tmp <- dat %>% filter(!Colony %in% outliers)
m1 <- glmmTMB(rgr40 ~ Treatment + (1|Block), data = tmp)

m2 <- glmmTMB(rgr40 ~ 1 + (1|Block), data = tmp)

round(anova(m1, m2, test = "Chisq"), 3)
```

```
## Data: tmp
## Models:
## m2: rgr40 ~ 1 + (1 | Block), zi=~0, disp=~1
## m1: rgr40 ~ Treatment + (1 | Block), zi=~0, disp=~1
##    Df     AIC    BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m2  3 -168.80 -164.8 87.399  -174.80
## m1  6 -164.79 -156.8 88.397  -176.79 1.995      3      0.573
```

No qualitative change.

Relative growth rate was similar for individuals in all treatments ($\chi^2 = 0.936$, df = 3, p = 0.817).

Remove offspring from full sib families with Inclusion probabilities < 0.7

```
tmp <- FSFamily_0_9 %>% filter(Prob.Inc.. < 0.7)

offspring_to_exclude <- c(tmp$Member1,tmp$Member2)

BestCluster_0_9 <- BestCluster_0_9 %>% filter(!(OffspringID %in% offspring_to_exclude))
```

How many kept and excluded?

```
nrow(FSFamily_0_9) # number of full sib families
```

```
## [1] 528
```

```
nrow(FSFamily_0_9 %>% filter(Prob.Inc.. > 0.7)) # number of full sib families with high confidence
```

```
## [1] 511
```

```
(nrow(FSFamily_0_9 %>% filter(Prob.Inc.. > 0.7)) / nrow(FSFamily_0_9)) * 100 # %
```

```
## [1] 96.7803
```

```
# number of full sib families with very high confidence (1)
nrow(FSFamily_0_9 %>% filter(Prob.Inc.. == 1))
```

```
## [1] 466
```

```
(nrow(FSFamily_0_9 %>% filter(Prob.Inc.. == 1)) / nrow(FSFamily_0_9)) * 100 # %
```

```
## [1] 88.25758
```

```
nrow(tmp) # number full sib families excluded
```

```
## [1] 17
```

```
length(offspring_to_exclude)  # number offspring excluded
```

```
## [1] 34
```

```
length(unique(BestCluster_0_9$MotherID)) # From this many mothers
```

```
## [1] 25
```

```
nrow(BestCluster_0_9) # this many offspring
```

```
## [1] 581
```

```
length(unique(BestCluster_0_9$FatherID)) # were sired by this many fathers
```

```
## [1] 291
```

```
# this many of which were from outside the array
BestCluster_0_9 %>% filter(grepl("#",FatherID)) %>% summarize(n=n_distinct(FatherID))
```

```
##     n
## 1 286
```

```
# this many  offspring sired by colonies in the array
BestCluster_0_9 %>% filter(!(grepl("#",FatherID))) %>% summarize(n=n())
```

```
##    n
## 1 13
```

```
BestCluster_0_9 %>% group_by(FatherID) %>%
  summarize(n=n_distinct(MotherID)) %>% count(n)
```

```
## Storing counts in `nn`, as `n` already present in input
## i Use `name = "new_name"` to pick a new name.
```

```
## # A tibble: 4 x 2
##       n    nn
##   <int> <int>
## 1     1   126
## 2     2   114
## 3     3    46
## 4     4     5
```

Calculate the number of unique sires per focal colony

```
n_fathers <- function(d){
  d$Mother_ID <- rapply(strsplit(d$MotherID,"_"),function(x) head(x,1))

  foo1 <- d %>%
    group_by(Mother_ID,FatherID) %>%
    summarise(n=n()) %>%
    mutate(freq=n/sum(n)) %>%
    ungroup()

  foo2 <- foo1 %>% group_by(Mother_ID) %>%
    summarise(nGenotyped=sum(n),
              UniqueNumberFather=n_distinct(FatherID))

  foo2$standardized.n.father <- with(foo1, tapply(n, Mother_ID, vegan::rarefy, sample=19))
  foo2$standardized.n.father <- round(foo2$standardized.n.father,3)
    # add sample info
  foo2 <- cbind.data.frame(foo2,dat[match(foo2$Mother_ID,dat$Colony),c(1:7,12)])

  foo2
}

n.fathers_0_9 <- n_fathers(d=BestCluster_0_9)
sort(n.fathers_0_9$standardized.n.father)
```

```
##  [1]  5.130 15.567 15.761 15.928 16.522 16.779 16.844 17.141 17.286 17.290
## [11] 17.371 17.519 17.519 17.648 17.761 17.761 17.761 18.000 18.380 18.380
## [21] 18.430 18.430 18.430 18.430 19.000
```

```
sort(n.fathers_0_9$UniqueNumberFather)
```

```
##  [1]  6 17 18 18 19 19 19 19 20 20 20 21 21 21 22 22 22 22 23 23 24 24 24 24 24
```

Calculate proportion for a beta glmm

```
n.fathers_0_9$standardized.prop.father <- n.fathers_0_9$standardized.n.father / 20

# make the 1's 0.9999 so can fit beta glmm
n.fathers_0_9$standardized.prop.father <- ifelse(n.fathers_0_9$standardized.prop.father == 1,
```

```
                                                0.99999,
                                                n.fathers_0_9$standardized.prop.father)
```

Analyze Number of Unique Sires per Mother

```
m1 <- glmmTMB(standardized.prop.father  ~ Treatment +
                (1|Block),
              family = 'beta_family',
              data = n.fathers_0_9)

m2 <- glmmTMB(standardized.prop.father  ~ 1 +
                (1|Block),
              family = 'beta_family',
              data = n.fathers_0_9)

result_sires <- round(anova(m1, m2, test = "Chisq"), 3)
result_sires
```

```
## Data: n.fathers_0_9
## Models:
## m2: standardized.prop.father ~ 1 + (1 | Block), zi=~0, disp=~1
## m1: standardized.prop.father ~ Treatment + (1 | Block), zi=~0, disp=~1
##    Df     AIC     BIC logLik deviance Chisq Chi Df Pr(>Chisq)
## m2  3 -42.865 -39.209 24.433  -48.865
## m1  6 -39.368 -32.054 25.684  -51.368 2.502      3      0.475
```

Get predicted values for plotting

```
predictions_sires = data.frame(Treatment = treatment.vec)

p <- emmeans(m1, ~ Treatment,
             component = "cond",
             type = "response")

predictions_sires$fit <- summary(p)$response * 20
predictions_sires$lwr <- summary(p)$asymp.LCL * 20
predictions_sires$upr <- summary(p)$asymp.UCL * 20
```

```
nd <- expand.grid(Treatment = treatment.vec,
                  Block = rownames(coef(m1)$cond$Block))

block_effects_sires <- data.frame(nd,
                                  fit = predict(m1,
                                                newdata = nd,
                                                type="response",
                                                se=F))

block_effects_sires$fit <- block_effects_sires$fit * 20
```

The number of standardized unique sires per colony did not differ among treatments ($\chi^2 = 2.502$, df = 3, p = 0.475).

Calculate the proportion of offspring from an assigned father

Get the mother ID's with an assigned father

```
offspring.proportion <- function(a,b){
# Get the maternal ID for offspring with a father in the array
    d1 <- a %>%
    distinct(MotherID,.keep_all=T)
# Add mother and father ID's to BestCluster data frame
  b$Mother_ID <- rapply(strsplit(b$MotherID,"_"),function(x) head(x,1))
  b$Father_ID <- rapply(strsplit(b$FatherID,"_"),function(x) head(x,1))
# Get the data from BestCluster for offspring with a father in the array
# and calculate the number and frequency of offspring
# per mother-father combination
  d2 <- b %>%
    filter(Mother_ID %in% d1$MotherID) %>%
    group_by(Mother_ID,Father_ID) %>%
    summarise(n=n()) %>%
    mutate(freq=n/sum(n)) %>%
    ungroup()
# Calculate the max frequency of fathers per mother
  d3 <- d2 %>%
    group_by(Mother_ID) %>%
    mutate(max.freq=max(freq)) %>%
    ungroup()
# Collect and arrange data
  d4 <- d3[grep("#",d3$Father_ID,invert=T),]
  d5 <- cbind.data.frame(d4, d1[match(d4$Mother_ID,d1$MotherID),])
  d6 <- d5 %>% select(Mother_ID, Father_ID,
                      n, freq, max.freq,
                      ProbDad1,
                      Mother.Block, Mother.Treatment, Mother.Position, Mother.Direction,
                      Father.Block, Father.Treatment, Father.Position, Father.Direction)

  d6
}

offspring.proportion_0_9 <- offspring.proportion(Paternity_dat_0_9, BestCluster_0_9)
offspring.proportion_0_9
```

```
##     Mother_ID Father_ID n       freq   max.freq ProbDad1 Mother.Block
## 2         02a       03a 2 0.08695652 0.08695652        1            3
## 2.1       02a       12b 1 0.04347826 0.08695652        1            3
## 6         06a       20a 2 0.08000000 0.08000000        1            4
## 9         09a       12b 1 0.04545455 0.09090909        1            2
## 19        20a       06a 4 0.19047619 0.19047619        1            4
## 21        22a       12b 2 0.09090909 0.13636364        1            2
## 25        26a       05b 1 0.04166667 0.04166667        1            4
##     Mother.Treatment Mother.Position Mother.Direction Father.Block
## 2               both               D                C            3
## 2.1             both               D                C            3
## 6               near               H                S            4
## 9               both               F                S            2
## 19              near               G                C            4
## 21              both               D                C            2
```

```
## 25               far               B               C            4
##     Father.Treatment Father.Position Father.Direction
## 2               both               E               N
## 2.1             both               E               N
## 6               near               G               C
## 9               both               E               S
## 19              near               H               S
## 21              both               E               S
## 25               far               C               N
```

```
# Out of 511 offspring with full sib inclusion probabilities > 0.7,
# 13 (2.5%) offspring were assigned paternity from one of
# five candidate fathers in the experimental array
sum(offspring.proportion_0_9$n)
```

```
## [1] 13
```

```
length(unique(offspring.proportion_0_9$Father_ID))
```

```
## [1] 5
```

```
offspring.proportion_0_9 %>% count(Father_ID)
```

```
##   Father_ID n
## 1       03a 1
## 2       05b 1
## 3       06a 1
## 4       12b 3
## 5       20a 1
```

```
Paternity_dat_0_9 %>% count(Distance)
```

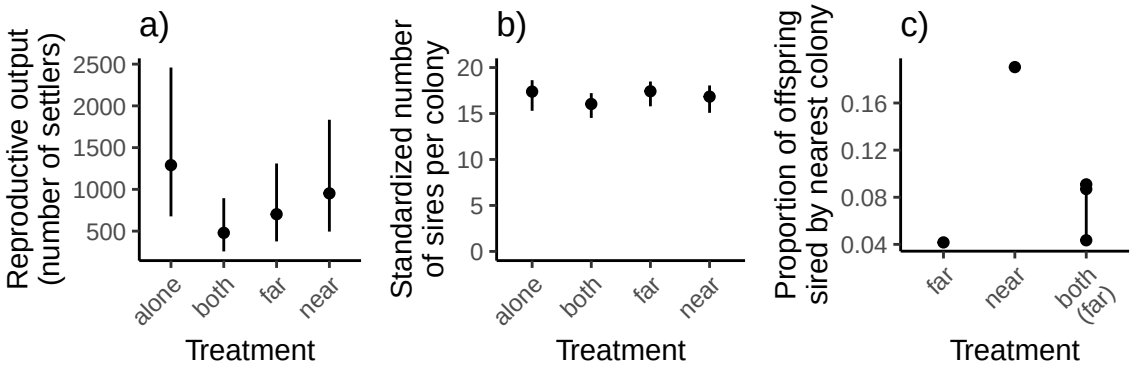
```
##   Distance  n
## 1  0.15000 10
## 2  0.85000  1
## 3  1.00000  1
## 4 70.00016  1
```

Make Figure 4

```
# Panel A
panel_A <- ggplot(data = predictions_reprod,
                  aes(x = Treatment,
                      y = fit)) +
  # geom_jitter(data = block_effects_reprod,
  #             aes(x = Treatment,
  #                 y = fit),
  #             width = 0.05,
```

```
  #             alpha = 0.4,
  #             color = "grey") +
  geom_linerange(data = predictions_reprod,
                 aes(ymin = lwr,
                     ymax = upr)) +
  geom_point() +
  labs(y = "Reproductive output\n(number of settlers)",
       title = "a)") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Panel B
panel_B <- ggplot(data = predictions_sires,
                  aes(x = Treatment,
                      y = fit)) +
  # geom_jitter(data = block_effects_sires,
  #             aes(x = Treatment,
  #                 y = fit),
  #             width = 0.05,
  #             alpha = 0.4,
  #             color = "grey") +
  geom_linerange(aes(ymin = lwr,
                     ymax = upr)) +
  geom_point() +
  ylim(0, 20) +
  labs(y = "Standardized number\nof sires per colony",
       title = "b)") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

# Panel C
tmp <- offspring.proportion_0_9 %>% filter(Mother.Block==Father.Block)

# far
y1 <- tmp %>% filter(Mother.Position=="B" & Father.Position == "C" |
                       Mother.Position=="C" & Father.Position == "B")
# near
y2 <- tmp %>% filter(Mother.Position=="G" & Father.Position %in% c("H"))

# both
y3 <- tmp %>% filter(Mother.Position=="D" & Father.Position %in% c("E","F"))

tmp <- rbind.data.frame(y1, y2, y3)

panel_C <- ggplot(data = tmp,
                  aes(x = Father.Treatment,
                      y = freq)) +
  geom_point() +
# for Mother_ID 02a, there were two fathers from the same block
    geom_linerange(data = tmp %>% filter(Mother_ID == "02a"),
                 aes(ymin = freq,
                     ymax = max.freq)) +
  labs(x = "Treatment",
```

```
       y = "Proportion of offspring\nsired by nearest colony",
       title = "c)") +
  scale_x_discrete(label = c("far","near","both\n(far)")) +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))

grid.arrange(panel_A, panel_B, panel_C,
             nrow = 1,
             ncol = 3)
```

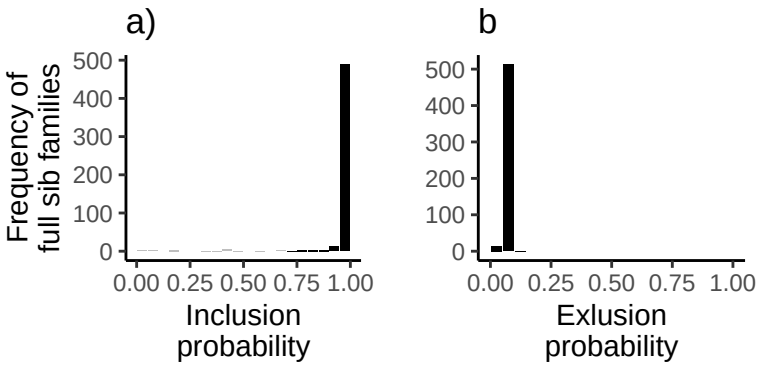



```
## pdf
##   2
```

Make Figure 5 (Inclusion and exclusion probabilities)

```
# Set breaks for plotting
brks <- seq(0,1,0.05)

# Get frequency of Inclusion probabilities
Inc <- with(FSFamily_0_9, hist(Prob.Inc..,
                               breaks = brks,
                               plot = F))

# Get frequency of Exclusion probabilities
Exc <- with(FSFamily_0_9,hist(Prob.Exc..,
                              breaks = brks,
                              plot = F))

# Panel A
d <- cbind.data.frame(mids = Inc$mids, counts = Inc$counts)
y <- d %>% filter(counts>0)

panel_A <- ggplot() +
  geom_col(data = y %>% filter(mids < 0.7),
           aes(x = mids,
               y = counts),
           fill = "grey",
           show.legend = F) +
  xlim(0,1) + ylim(0,max(y$counts)) +
  geom_col(data = y %>% filter(mids >= 0.7),
           aes(x = mids,
```

```
                y = counts),
            fill = "black",
            show.legend = F) +
  labs(x = "Inclusion\nprobability",
       y = "Frequency of\nfull sib families",
       title = "a)") +
  theme_classic()

# Panel B
d <- cbind.data.frame(mids = Exc$mids, counts = Exc$counts)
y <- d %>% filter(counts>0)

panel_B <- ggplot() +
  geom_col(data = y,
           aes(x = mids,
               y = counts),
           fill = "black",
           show.legend = F) +
  xlim(0,1) +
  labs(x = "Exlusion\nprobability",
       y = "",
       title = "b") +
  theme_classic()

grid.arrange(panel_A, panel_B, nrow = 1, ncol = 2)
```




```
## pdf
##   2
```

Calculate the frequency of potential and observed distances between parents

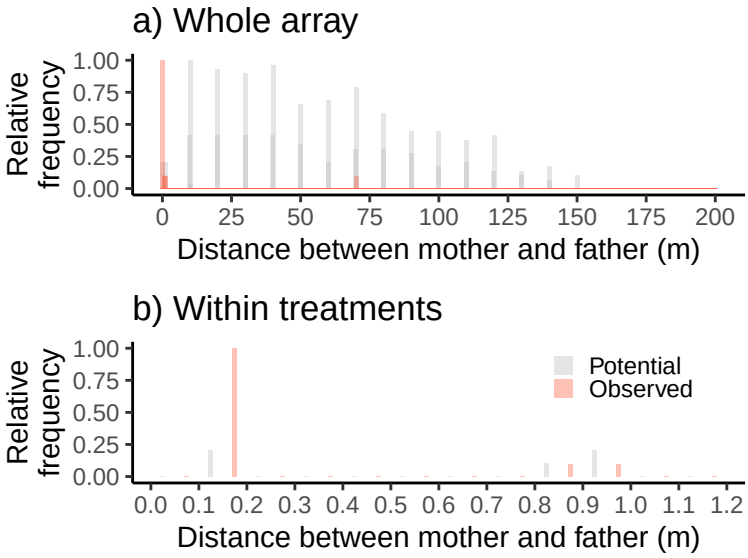
```
brks <- seq(0,200,0.1)
# # Potential distances
tmp <- dat %>% select("X", "Y")
```

```r
mat <- dist(tmp, diag = T)
possible_distances <- mat[lower.tri(mat, diag = T)]
a <- hist(possible_distances, breaks = brks, plot = F)
possible_density <- data.frame(density = a$density / max(a$density),
                               mids = a$mids)
# Observed
density_freq <- function(d){
  b <- hist(d$Distance,breaks=brks,plot=F)
  data.frame(n = b$counts,
             density = b$density / max(b$density),
             mids = b$mids)
}

observed_density_0_9 <- density_freq(Paternity_dat_0_9)

possible_density$Metric <- "Potential"
observed_density_0_9$Metric <- "Observed"
density_data <- rbind.data.frame(possible_density, observed_density_0_9[,-1])
density_data$Metric <- factor(density_data$Metric,
                              levels = c("Potential","Observed"))
```

Make Figure 6

```r
panel_A <- ggplot() +
  geom_col(data = density_data,
                  aes(x = mids,
                      y = density,
                      fill = Metric),
           width = 3,
           alpha = 0.4,
           position_dodge(0.01)) +
  labs(x = "Distance between mother and father (m)",
       y = "Relative\nfrequency",
       title = "a) Whole array") +
  scale_fill_manual(name = 'd',
                    breaks = c('Potential','Observed'),
                    values = c('Potential' = 'grey',
                               'Observed' = 'tomato')) +
  scale_x_continuous(breaks = seq(0,300,25)) +
  theme_classic() +
  theme(legend.position = 'none')

panel_B <- ggplot() +
  geom_col(data = density_data %>% filter(mids < 1.2),
                  aes(x = mids,
                      y = density,
                      fill = Metric),
           width = 0.02,
           alpha = 0.4,
           position_dodge(0.1)) +
  labs(x = "Distance between mother and father (m)",
       y = "Relative\nfrequency",
```

```
       title = "b) Within treatments") +
  scale_fill_manual(name = '',
                    breaks = c('Potential','Observed'),
                    values = c('Potential' = 'grey',
                               'Observed' = 'tomato')) +
  scale_x_continuous(breaks = seq(0,3,0.1)) +
  theme_classic() +
  theme(legend.position = c(0.8, 0.9),
        legend.key.size = unit(0.3, 'cm'))

grid.arrange(panel_A, panel_B,
             ncol = 1,
             nrow = 2)
```




```
## pdf
##   2
```