

Appendix for ‘Bees Without Flowers’

Contents

1	Set up the workspace	1
2	Core model formulas	1
3	Model fitting	2
4	Model comparison	2
5	Description of the full Honeydew model	4
6	Code for Figure 3	5
7	Post-hoc comparisons between specific treatment pairs	8
8	Comparing to a model with environmental predictors & continuous date	9

1 Set up the workspace

```
set.seed(1)
library(lme4)      # For model fitting
library(tidyverse)
library(mixtools)  # For bivariate Gaussian ellipses

d = read.csv("Meiners_BeeHoneydew_data.csv")
```

2 Core model formulas

We will be focusing on models that include fixed effect for the experimental manipulations and for site, as well as a continuous time-of-day variable to capture variation in bee activity associated with diurnal patterns. We modeled variation among days (e.g. due to differences in recent weather events) and among individual plants using random effects.

While additional variables were recorded during sampling, their primary purpose was to keep the sampling effort focused on a narrow range of environmental conditions (because environmental effects such as humidity were not related to our primary hypotheses). We thus did not expect these variables to vary enough in our samples to substantially affect the results, and did not include them in most of our analyses. In the final section of this Appendix, we show that the exclusion of these variables (and of a continuous measure of seasonality) do not affect the statistical significance or point estimates associated with any treatment effects, and that they do not significantly improve model fit in terms of χ^2 or AIC.

```
raw_formula = "Bee_Count ~ Mold * Insecticide +
               Sugar * Paint +
               scale(min_day) +
               Site +
               (1|Plant_Code) +
               (1|julDate)"
```

```
formula = as.formula(raw_formula)
```

We also fit a model that did not include sugar as a predictor variable, to assess whether its inclusion substantially improves our ability to predict bee density.

```
# Drop "Sugar" and the asterisk from the formula
```

```
no_sugar_formula = as.formula(  
  gsub("Sugar \\* ", "", raw_formula)  
)
```

```
print(no_sugar_formula)
```

```
## Bee_Count ~ Mold * Insecticide + Paint + scale(min_day) + Site +
```

```
##      (1 | Plant_Code) + (1 | julDate)
```

3 Model fitting

We modeled the bee counts with negative binomial and Poisson mixed models with the default log link.

```
# Some versions of the model only reach the maximum-likelihood
```

```
# estimate without warnings when we use this optimizer
```

```
control = glmerControl(optimizer = "bobyqa")
```

```
Honeydew <- glmer.nb(  
  formula,  
  data=d,  
  control = control  
)
```

```
Honeydew_poisson <- glmer(  
  formula,  
  data=d,  
  family = poisson,  
  control = control  
)
```

```
Honeydew_no_sugar <- glmer.nb(  
  no_sugar_formula,  
  data=d,  
  control = control  
)
```

4 Model comparison

Which elements of the models fit above are essential? See what happens to model performance (AIC) when various degrees of freedom are removed from the full **Honeydew** negative binomial model.

```
# Drop predictors from the model & reformat the output for
```

```
# subsequent work (e.g. removing headings, renaming columns).
```

```
# When `drop` says the number of degrees of freedom is NA, it actually
```

```
# means zero, so replace the NAs.
```

```

# If the model in `x` is already simplified, then report a larger
# reduction in degrees-of-freedom.
make_dropped_df = function(x, distribution, n_fewer_df = 0){
  drop1(x) %>%
    structure(heading = NULL) %>%
    rownames_to_column(var = "dropped") %>%
    cbind(distribution = distribution) %>%
    mutate(Df = ifelse(is.na(Df), 0, Df)) %>%
    mutate(Df = Df + n_fewer_df) %>%
    rename_(`df reduction` = "Df")
}

# Use the above function on both of the full models, then manually add
# a row for the no_sugar model.
# Finally, eliminate "<" and ">" to prevent formatting errors
initial_dropped_df = rbind(make_dropped_df(Honeydew, "Negative Binomial", 0),
                           make_dropped_df(Honeydew_poisson, "Poisson", 1)) %>%
  rbind(data_frame(dropped = "Sugar", `df reduction` = 2,
                   AIC = AIC(Honeydew_no_sugar),
                   distribution = "Negative Binomial")) %>%
  mutate(dropped = gsub("[\\<\\>]", "", dropped))

```

Omitting site or either of the interaction terms (lines 1, 2, 4 and 5) produces a relatively small change in AIC, compared with the full model (line 3). However, none of the models without overdispersion (i.e. the Poisson-distributed models) had any appreciable AIC weight, nor did the model that removed all sugar effects (line 10).

```

# Sort, calculate DeltaAIC & AIC weights, format for printing with
# reasonable precision
initial_dropped_df %>%
  arrange(AIC) %>%
  mutate(`$\\Delta$AIC` = AIC - AIC[1]) %>%
  select(-AIC) %>%
  mutate(`AIC weight (%)` = 100 * exp(-`$\\Delta$AIC` / 2) /
        sum(exp(-`$\\Delta$AIC` / 2))) %>%
  cbind(``=1:nrow(.), .) %>%
  knitr::kable(digits = c(rep(1, 4), 2, 1), align = c("llclrr"))

```

	dropped	df reduction	distribution	Δ AIC	AIC weight (%)
1	Site	2	Negative Binomial	0.00	59.4
2	Sugar:Paint	1	Negative Binomial	1.58	26.9
3	none	0	Negative Binomial	3.57	10.0
4	Mold:Insecticide	1	Negative Binomial	5.68	3.5
5	scale(min_day)	1	Negative Binomial	11.40	0.2
6	Site	3	Poisson	28.16	0.0
7	Sugar:Paint	2	Poisson	29.97	0.0
8	none	1	Poisson	31.94	0.0
9	Mold:Insecticide	2	Poisson	34.06	0.0
10	Sugar	2	Negative Binomial	38.69	0.0
11	scale(min_day)	2	Poisson	71.34	0.0

χ^2 tests show the same result: omitting sugar effects or overdispersion significantly reduces model performance ($P < .000001$).

```
anova(Honeydew, Honeydew_no_sugar)
```

```
## Data: d
## Models:
## Honeydew_no_sugar: Bee_Count ~ Mold * Insecticide + Paint + scale(min_day) + Site +
## Honeydew_no_sugar:      (1 | Plant_Code) + (1 | julDate)
## Honeydew: Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
## Honeydew:      Site + (1 | Plant_Code) + (1 | julDate)
##           Df      AIC      BIC logLik deviance Chisq Chi Df
## Honeydew_no_sugar 11 820.74 864.02 -399.37   798.74
## Honeydew          13 785.62 836.77 -379.81   759.62 39.12    2
##           Pr(>Chisq)
## Honeydew_no_sugar
## Honeydew          3.201e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(Honeydew, Honeydew_poisson)
```

```
## Data: d
## Models:
## Honeydew_poisson: Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
## Honeydew_poisson:      Site + (1 | Plant_Code) + (1 | julDate)
## Honeydew: Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
## Honeydew:      Site + (1 | Plant_Code) + (1 | julDate)
##           Df      AIC      BIC logLik deviance Chisq Chi Df
## Honeydew_poisson 12 813.99 861.21 -395.00   789.99
## Honeydew          13 785.62 836.77 -379.81   759.62 30.374    1
##           Pr(>Chisq)
## Honeydew_poisson
## Honeydew          3.562e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5 Description of the full Honeydew model

```
summary(Honeydew, correlation = FALSE)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
## Approximation) [glmerMod]
## Family: Negative Binomial(1.8463) ( log )
## Formula:
## Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
## Site + (1 | Plant_Code) + (1 | julDate)
## Data: d
## Control: control
##
##           AIC      BIC logLik deviance df.resid
##       785.6    836.8  -379.8   759.6      365
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.0060 -0.5083 -0.3610  0.1850  4.6619
```

```
##
## Random effects:
## Groups      Name      Variance Std.Dev.
## Plant_Code (Intercept) 0.3982  0.6311
## julDate     (Intercept) 0.1712  0.4138
## Number of obs: 378, groups: Plant_Code, 63; julDate, 9
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -1.94403    0.50128  -3.878 0.000105 ***
## Mold           1.15201    0.49711   2.317 0.020481 *
## Insecticide    0.45650    0.52688   0.866 0.386259
## Sugar          2.41458    0.47328   5.102 3.36e-07 ***
## Paint         -0.46287    0.60340  -0.767 0.443017
## scale(min_day) 0.27353    0.08792   3.111 0.001863 **
## SiteB          0.28098    0.45612   0.616 0.537881
## SiteC          0.04974    0.46442   0.107 0.914710
## Mold:Insecticide -1.48408    0.71785  -2.067 0.038697 *
## Sugar:Paint     0.08231    0.70730   0.116 0.907354
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(Honeydew)
```

```
## Analysis of Variance Table
##              Df Sum Sq Mean Sq F value
## Mold           1  5.002    5.002  5.0019
## Insecticide     1 12.643   12.643 12.6429
## Sugar           1 39.648   39.648 39.6479
## Paint           1  2.841    2.841  2.8408
## scale(min_day)  1 10.158   10.158 10.1582
## Site            2  0.414    0.207  0.2069
## Mold:Insecticide 1  4.880    4.880  4.8797
## Sugar:Paint     1  0.013    0.013  0.0133
```

6 Code for Figure 3

```
# Indicator for, "was experimental manipulation i applied to treatment j?"
Mold =      c(1, 1, 0, 0, 0, 0, 0)
Insecticide = c(0, 1, 0, 1, 0, 0, 0)
Sugar =      c(0, 0, 0, 0, 0, 1, 1)
Paint =      c(0, 0, 0, 0, 1, 0, 1)

treat_names = c("Natural Mold", "Natural Mold + Insecticide", "Control",
                "Insecticide", "Black Paint", "Sugar", "Sugar + Black Paint")

newdata = cbind(
  `(Intercept)` = 1,
  Mold = Mold,
  Insecticide = Insecticide,
  Sugar = Sugar,
  Paint = Paint,
  `scale(min_day)` = 0,
```

```

SiteB = 0,
SiteC = 0,
`Mold:Insecticide` = Mold * Insecticide,
`Sugar:Paint` = Sugar * Paint
)
row.names(newdata) = treat_names
newdata

##               (Intercept) Mold Insecticide Sugar Paint
## Natural Mold              1    1              0    0    0
## Natural Mold + Insecticide 1    1              1    0    0
## Control                   1    0              0    0    0
## Insecticide                1    0              1    0    0
## Black Paint                1    0              0    0    1
## Sugar                      1    0              0    1    0
## Sugar + Black Paint        1    0              0    1    1
##               scale(min_day) SiteB SiteC Mold:Insecticide
## Natural Mold              0    0    0              0
## Natural Mold + Insecticide 0    0    0              1
## Control                   0    0    0              0
## Insecticide                0    0    0              0
## Black Paint                0    0    0              0
## Sugar                      0    0    0              0
## Sugar + Black Paint        0    0    0              0
##               Sugar:Paint
## Natural Mold              0
## Natural Mold + Insecticide 0
## Control                   0
## Insecticide                0
## Black Paint                0
## Sugar                      0
## Sugar + Black Paint        1

# mean and variance from lme4's Laplace approximation
parameter_mu = fixef(Honeydew)
parameter_sigma = as.matrix(vcov(Honeydew))

# Generate Monte Carlo samples from lme4's approximate likelihood surface
posterior_samples = MASS::mvrnorm(1E6, parameter_mu, parameter_sigma) %*% t(newdata)

mu = colMeans(posterior_samples)

# Density of bivariate normal between Control and a named treatment
bivariate_normal_control_density = function(x, name){
  names = c("Control", name)
  mvtnorm::dmvnorm(x,
                    mu[names],
                    cov(posterior_samples)[names, names])
}

label_df = data.frame(
  x = c(rep(log(.05), 3), .9),
  y = c(mu[c("Sugar", "Natural Mold", "Natural Mold + Insecticide")], 1),

```

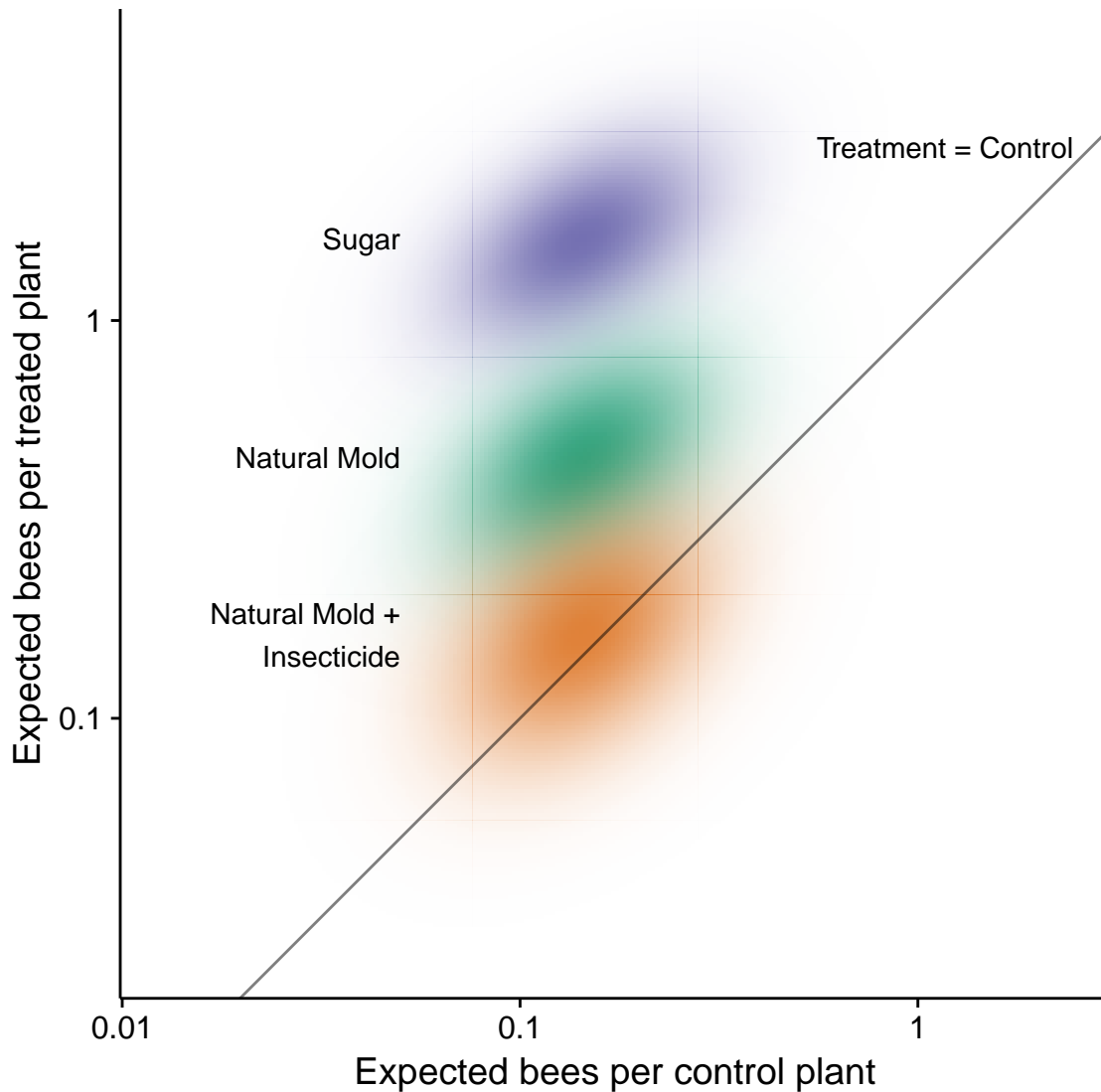
```

  label = c("Sugar", "Natural Mold", "Natural Mold +\nInsecticide", "Treatment = Control")
)

# for each set of x and y values, calculate bivariate densities,
# then tidy up the results for ggplot
plot_data = expand.grid(x = seq(log(.01), log(3), length = 250),
  y = seq(log(.02), log(6), length = 250)) %>%
  mutate(Sugar = bivariate_normal_control_density(., "Sugar"),
    `Natural Mold` = bivariate_normal_control_density(., "Natural Mold"),
    `Natural Mold + Insecticide` =
      bivariate_normal_control_density(., "Natural Mold + Insecticide")) %>%
  gather(key = treatment, value = likelihood, Sugar,
    `Natural Mold`, `Natural Mold + Insecticide`) %>%
  mutate(scaled_likelihood = likelihood / max(likelihood))

plot_data %>%
  ggplot(aes(x = x, y = y, alpha = scaled_likelihood,
    fill = treatment)) +
  geom_tile() +
  scale_alpha_continuous(range = c(0, 1), guide = FALSE) +
  geom_abline(intercept = 0, slope = 1, color = alpha("black", .5)) +
  cowplot::theme_cowplot() +
  scale_fill_brewer(palette = "Dark2", guide = FALSE) +
  xlab("Expected bees per control plant") +
  ylab("Expected bees per treated plant") +
  coord_equal() +
  scale_x_continuous(breaks = log(10^seq(-10, 10)), labels = 10^seq(-10, 10),
    expand = c(0, 0)) +
  scale_y_continuous(breaks = log(10^seq(-10, 10)), labels = 10^seq(-10, 10),
    expand = c(0, 0)) +
  geom_text(data = label_df, aes(x = x, y = y, label = label),
    inherit.aes = FALSE, hjust = "right")

```



7 Post-hoc comparisons between specific treatment pairs

```
# P-value for "Sugar visits > Mold visits"
mean(posterior_samples[, "Natural Mold"] > posterior_samples[, "Sugar"])

## [1] 0.00085

# P-value for "Mold visits > Control visits"
mean(posterior_samples[, "Control"] > posterior_samples[, "Natural Mold"])

## [1] 0.010143

# P-value for "Mold + Insecticide visits < Mold visits"
mean(posterior_samples[, "Natural Mold + Insecticide"] > posterior_samples[, "Natural Mold"])

## [1] 0.017251

# P-value for "Sugar+Paint visits < Sugar visits" (not significant)
mean(posterior_samples[, "Sugar + Black Paint"] > posterior_samples[, "Sugar"])
```



```
## [1] 0.151
```

8 Comparing to a model with environmental predictors & continuous date

We could have obtained essentially the same results with a much larger model that included a fixed effect for date and environmental conditions (i.e., there would still be a significant sugar effect with point estimate ≈ 2.4 , a significant mold effect with point estimate ≈ 1.2 , and a significant mold/insecticide interaction with point estimate ≈ -1.5).

However, the eight degrees of freedom associated with environmental conditions and the fixed effect for date do not improve the model fit enough to justify the additional complexity ($\chi^2 > 0.25$, higher AIC), so we disregard them outside of this section.

```
Honeydew_env = glmer.nb(Bee_Count ~ Mold * Insecticide +
  Sugar * Paint +
  scale(min_day) +
  Site +
  scale(min_day) +
  scale(Temp_F) +
  scale(Wind_mph) +
  Conditions +
  scale(Barometric) +
  scale(Humidity) +
  scale(julDate) +
  (1|Plant_Code) +
  (1|julDate),
  data = d,
  control = control)
print(summary(Honeydew_env), correlation = FALSE)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
## Family: Negative Binomial(2.1074) ( log )
## Formula:
## Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
##   Site + scale(min_day) + scale(Temp_F) + scale(Wind_mph) +
##   Conditions + scale(Barometric) + scale(Humidity) + scale(julDate) +
##   (1 | Plant_Code) + (1 | julDate)
## Data: d
## Control: control
##
##      AIC      BIC   logLik deviance df.resid
##    791.7    874.3   -374.9    749.7     357
##
## Scaled residuals:
##      Min       1Q   Median       3Q      Max
## -1.0095 -0.5238 -0.3413  0.1952  4.7630
##
## Random effects:
## Groups      Name             Variance Std.Dev.
## Plant_Code (Intercept) 0.4284    0.6545
## julDate     (Intercept) 0.1009    0.3176
```

```
## Number of obs: 378, groups: Plant_Code, 63; julDate, 9
##
## Fixed effects:
##
##              Estimate Std. Error z value
## (Intercept) -1.94570    0.69952  -2.781
## Mold        1.17431    0.50501   2.325
## Insecticide  0.42923    0.53528   0.802
## Sugar       2.39457    0.47917   4.997
## Paint      -0.61127    0.61236  -0.998
## scale(min_day) 0.01771    0.15687   0.113
## SiteB       0.73399    0.57844   1.269
## SiteC       0.50101    0.53278   0.940
## scale(Temp_F) 0.22330    0.18097   1.234
## scale(Wind_mph) 0.19258    0.12820   1.502
## ConditionsCompletly Cloudy (no Shadow) 0.39652    0.55278   0.717
## ConditionsFull Sun -0.65723    0.55214  -1.190
## ConditionsPartly Cloudy (>50% sun) 0.23767    0.44193   0.538
## scale(Barometric) -0.34483    0.17986  -1.917
## scale(Humidity) -0.03533    0.19502  -0.181
## scale(julDate) -0.13315    0.16300  -0.817
## Mold:Insecticide -1.47040    0.73221  -2.008
## Sugar:Paint  0.29472    0.71951   0.410
##
##              Pr(>|z|)
## (Intercept)  0.00541 **
## Mold        0.02006 *
## Insecticide  0.42262
## Sugar       5.81e-07 ***
## Paint       0.31817
## scale(min_day) 0.91013
## SiteB       0.20448
## SiteC       0.34703
## scale(Temp_F) 0.21724
## scale(Wind_mph) 0.13304
## ConditionsCompletly Cloudy (no Shadow) 0.47318
## ConditionsFull Sun 0.23391
## ConditionsPartly Cloudy (>50% sun) 0.59072
## scale(Barometric) 0.05521 .
## scale(Humidity) 0.85623
## scale(julDate) 0.41397
## Mold:Insecticide 0.04462 *
## Sugar:Paint  0.68209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(Honeydew, Honeydew_env)
```

```
## Data: d
## Models:
## Honeydew: Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
## Honeydew:      Site + (1 | Plant_Code) + (1 | julDate)
## Honeydew_env: Bee_Count ~ Mold * Insecticide + Sugar * Paint + scale(min_day) +
## Honeydew_env:      Site + scale(min_day) + scale(Temp_F) + scale(Wind_mph) +
## Honeydew_env:      Conditions + scale(Barometric) + scale(Humidity) + scale(julDate) +
## Honeydew_env:      (1 | Plant_Code) + (1 | julDate)
##              Df      AIC      BIC logLik deviance Chisq Chi Df Pr(>Chisq)
```

## Honeydew	13	785.62	836.77	-379.81	759.62			
## Honeydew_env	21	791.70	874.34	-374.85	749.70	9.915	8	0.271