Fitness of *Penstemon digitalis* increases with floral scent emission

BIOS15 Exam 2025-26

Oliver E. Todreas

2026-01-14

Introduction

Floral scent is often thought to be an important part of floral phenotypes, but is rarely studied (Parachnowitsch *et al.*, 2012). Our hypothesis is that higher floral scent emission results in higher reproductive success. This hypothesis is formed by the assumption that floral scent can help guide pollinators to flowers, thus increasing the likelihood that offspring are successfully produced. To test this hypothesis, we analyze data collected in a garden study on a variety of phenotypic variables of *Penstemon digitalis*. Specifically, we analyze the effect of floral scent emission on a fitness variable based on fruit mass and fruit number. In our analysis we include the population of origin from which a given individual came and account for the hypothesized random effect of experimental block.

Methods

The data used contain a variety of phenotypic data for *P. digitalis* individuals. Each data point represents one plant. The floral scent emission was measured per plant in $ng/L/h$, while fitness was defined as number of fruits times fruit mass in mg. Both floral scent emission and fitness were natural log-transformed to remove right skew in the data, shown in the model description below. As a result, one individual, whose log of floral scent emission was undefined, was dropped. Furthermore, each plant was sourced from one of three populations in the north-eastern USA, abbreviated NR, WF, and TH. Plants were arranged in a randomized complete block design, where 35 blocks each contained one plant from each of the three populations. We hypothesized that different populations might exhibit higher fitness than others, and therefore included population of origin as a fixed effect. Furthermore, we included the blocks as random effects. Thu, the following generalized linear mixed model with random intercepts was fitted to the data, shown here in R syntax.

```
log(fitness) ~ log(floral scent emission) + population + (1|block)
```

Results

The whole model (both fixed and random effects) fitted to the data explained 26.00% of the variance in fitness across the dataset (conditional pseudo-R-squared = 0.2600$). 18.58% of the variance in fitness was explained by the experimental block (Table 1). Fitness was predicted to increased by 0.16% for each 1% increase in total scent emission ($elasticity = 0.16 \pm 0.05 mg/ng/L/h$, Table 2, Figure 1). At any given total scent emission, population WF had the highest predicted fitness, followed by population NR and TH, in descending order (Table 2, Figure 1). On the data scale, this meant that at the first quartile of floral scent emission ($Q1 = 1.72 ng/L/h$), the predicted fitness was 68.5, 70.6, and 73.5 mg for populations TH, NR, and WF respectively. At the third quartile of floral scent emission (Q3 = 7.80 $ng/L/h$), the predicted fitness was 86.7, 89.2, and 92.9 mg for populations TH, NR, and WF respectively. Thus, an increase in fitness of 18.1, 18.7, and 19.4 mg was predicted by the model across the interquartile range of total floral scent.

Table 1: Variance partitioning in generalized linear mixed model. The variance within groups represented the majority of the variance in the model.

Component	Variance (mg^2)
Variance among blocks	0.05
Variance within groups	0.22
Percent variance explained by block	18.58

Table 2: Parameter estimates and standard errors for the fixed effects of the generalized linear mixed model given on the link scale. Conditional pseudo-R-squared = 0.26.

Parameter	Estimate	Standard error
Slope (ln(mg))/ln(ng/L/h))	0.16	0.05
NR intercept (ln(mg))	4.17	0.12
TH intercept (ln(mg))	4.14	0.13
WF intercept (ln(mg))	4.21	0.13

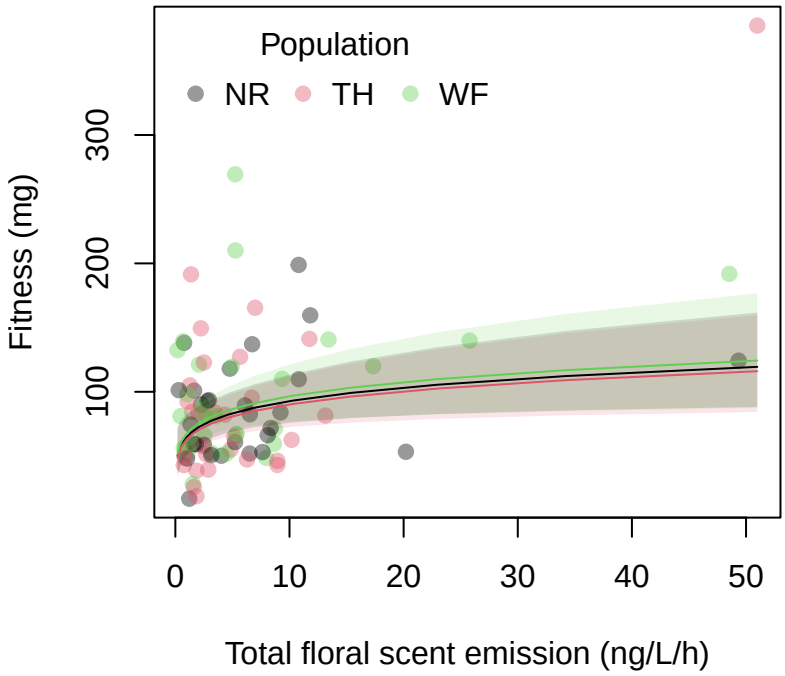

Figure 1: Effect of total floral scent emission and population on fitness (random effects not included in predictions) shown on the data scale. Ribbons represent 95% confidence intervals.

Conclusions

The analysis presented in this report supports the hypothesis that increased floral scent emission increases fitness of *P. digitalis*. The increase was small but noticable: there was a 0.16% increase in fitness per 1% increase in total scent emission. We found that the populations were relatively similar in how total scent emission affected fitness, but that fitness was generally the fighest for WF and the lowest for TH. We also found that a fair share of variance was represented by the experimental blocks, suggesting that the

design captured the variance within groups to a large extent. In conclusion, total scent emission had a small but noticable effect on the fitness of plants, and populations were generally quite similar in regards to this relationship. These findings point towards the effect of floral scent emission and how it can affect fruiting success, highlighting the importance of the variable.

References

A. Parachnowitsch, R. Raguso, A. Kessler. 2012. Phenotypic selection to increase floral scent emission, but not flower size or colour in bee-pollinated *Penstemon digitalis*. *New Phytologist* **195**: 667–675.

Appendix

The repository for this project can be found at github.com/otodreas/BIOS15_Exam/

The script, data, and the report itself can be found in the GitHub repo at the following locations

```
root/
└── Report/
    ├── Scripts/
    │   ├── Utils/
    │   │   ├── Plot.R
    │   │   ├── Predictor.R
    │   │   ├── Table1.R
    │   │   └── Table2.R
    │   └── Analysis.R
    ├── Data/
    │   └── penstemon_copy.txt
    └── Todreas_BIOS15_Exam.pdf
```

Note: the tree above is simplified for clarity.

Below is a copy of `Report/Scripts/Analysis.R` and `Report/Scripts/Utils/Plot.R`.

Feel free to take a look at `Report/Scripts/Utils/Predictor.R`, `Report/Scripts/Utils/Table1.R`, or `Report/Scripts/Utils/Table2.R` if you want to check how I made the tables or generated the example data I used in the report.

`Report/Scripts/Analysis.R`

```
# This script fits a GLM to the dataset provided. The goal of the analysis is
# to determine how total floral scent emission effects a plant's fitness.
# Scripts that generate plots and tables can be found in /Utils/


# =====================
# CONFIGURE ENVIRONMENT
# =====================

# Clear variables
rm(list = ls())

# Load packages
library(here)
library(dplyr)
library(glmmTMB)
library(MuMIn)

# Supress update warning from MuMIn because all the variables used in the
```

```r
# original model call have the same values as when the model was fitted
options(MuMIn.noUpdateWarning = TRUE)

# Set data filepath
data_path <- here("Report", "Data", "penstemon_copy.txt")


# =========
# LOAD DATA
# =========

# Load raw data
df <- as_tibble(read.table(data_path, header = TRUE)) |>
  select(Pop, Block, tscent, fitness) |>  # Select relevant columns
  mutate(across(c(Pop, Block), as.factor)) |>  # Make grouped variables factors
  filter(tscent != 0 & fitness != 0) |>  # Drop values whose log is undefined
  mutate(log_tscent = log(tscent)) |>  # Create log_tscent column
  mutate(log_fitness = log(fitness))  # Create log_fitness column

# NOTE: one row is dropped for 0 values


# ==================================
# FIT MODEL AND GENERATE PREDICTIONS
# ==================================

# Fit glmm on a log-log scale
m <- glmmTMB(log_fitness ~ log_tscent + Pop + (1|Block), data = df)

# Create data frames with sequences to generate new predictions on for each pop
new_data <- list()

for (i in seq_along(levels(df$Pop))) {
  new_data[[i]] <- tibble(
    log_tscent = seq(min(df$log_tscent), max(df$log_tscent), length.out = 15),
    Pop = factor(rep(levels(df$Pop)[i], 15))
  )
}

names(new_data) <- levels(df$Pop)

# Generate predictions and standard error estimates
preds <- list()

for (i in seq_along(new_data)) {
  preds[[i]] <- predict(
    m,
    newdata = new_data[[i]],
    type = "response",  # Calculate predictions on response scale
    se.fit = TRUE,  # Save standard error fit for each input
    re.form = NA  # Do not include random effects in predictions
  )
}

names(preds) <- levels(df$Pop)

# Get r^2
rsq <- r.squaredGLMM(m)
```

```r
# Perform variance partitioning (v_part called in Utils/Table1.R)
v_part <- c(attr(VarCorr(m)$cond$Block, "stddev")^2, attr(VarCorr(m)$cond, "sc")^2)
```

Report/Scripts/Utils/Plot.R

```r
# This script calls the script ../Analysis.R and creates a plot.


# =====================
# CONFIGURE ENVIRONMENT
# =====================

# Clear variables
rm(list = ls())

# Load libraries
library(here)

# Source analysis file
source(here("Report", "Scripts", "Analysis.R"), echo = FALSE)


# ==========
# MAKE PLOTS
# ==========

# Create plot on data scale
plot(
  df$tscent,
  df$fitness,
  xlab = "Total floral scent emission (ng/L/h)",
  ylab = "Fitness (mg)",
  col = adjustcolor(seq_along(levels(df$Pop)), alpha.f = 0.4),
  pch = 19,
)

# Draw legend
legend(
  "topleft",
  legend = levels(df$Pop),
  col = adjustcolor(seq_along(levels(df$Pop)), alpha.f = 0.4),
  pch = 19,
  bty = "n",
  horiz = TRUE,
  title = "Population",
  inset = 0.02
)

# Draw 95% CI ribbons on data scale for each prediction
for (i in seq_along(preds)) {
  polygon(
    # x-coordinates of ribbon
    c(exp(new_data[[i]]$log_tscent), rev(exp(new_data[[i]]$log_tscent))),

    # y-coordinates of ribbon
    c(
      exp(preds[[i]]$fit + 1.96 * preds[[i]]$se.fit),
```

```
      rev(exp(preds[[i]]$fit - 1.96 * preds[[i]]$se.fit))
    ),
    col = adjustcolor(i, alpha.f = 0.15),
    border = FALSE
  )
}

# Draw regression lines on data scale
for (i in seq_along(preds)) {
  lines(exp(new_data[[i]]$log_tscent), exp(preds[[i]]$fit), col = i)
}
```