Characterization of and selection on compound within-individual floral variation in *Vicia americana* (Fabaceae)

Mason W. Kulbaba[*]

June 17, 2025

Contents

Abstract

This document provides code to reproduce all results from the manuscript `Characterization of and selection on compound within-individual floral variation in *Vicia americana* (Fabaceae)`. The data file `vicia_final_data.csv` contains all data required to reproduce all results in the manuscript, and is located in the associated Zenodo repository. This study sought to describe the floral traits of *Vicia americana* as compound function-valued traits, and compare standardized linear selection estimates (e.g., β) as per Lande and Arnold (1983), with the functional regression approached used by Kulbaba, Clocher, and Harder (2017) and Harder et al. (2019).

[*] St. Mary's University, mason.kulbaba@stmu.ca, https://orcid.org/0000-0003-0619-7089

1 R

- The version of R used to make this document is 4.5.0.
- The version of the `rmarkdown` package used to make this document is 2.29.
- The version of the `bookdown` package used to make this document is 0.43.
- The version of the `dplyr` package used to make this document is 1.1.4.
- The version of the `glmmTMB` package used to make this document is 1.1.11.
- The version of the `DHARMa` package used to make this document is 0.4.7.
- The version of the `car` package used to make this document is 3.1.3.
- The version of the `caret` package used to make this document is 7.0.1.
- The version of the `Hmisc` package used to make this document is 5.2.3.
- The version of the `tidyr` package used to make this document is 1.3.1.
- The version of the `viridis` package used to make this document is 0.6.5.
- The version of the `refund` package used to make this document is 0.1.37.

Attach packages.

```r
suppressMessages(library("dplyr"))
suppressMessages(library("glmmTMB"))
suppressMessages(library("ggplot2"))
suppressMessages(library("DHARMa"))
suppressMessages(library("car"))
suppressMessages(library("caret"))
suppressMessages(library("Hmisc"))
suppressMessages(library("tidyr"))
suppressMessages(library("viridis"))
suppressMessages(library("refund"))
```

2 Data

Load data file

```r
data<- read.csv("vicia_final_data.csv")
```

where the variables are

- `PlantID` is a unique numerical identifier for each individual in the study (1-40).
- `Branch` is a unique numerical identifier for each sequentially produced raceme (1-10). The first raceme to flower was designated as 1, and was the most basal.
- `PosSeq` is the sequential flower position (1-49) across all sequentially flowering racemes.
- `BPos` is a composite of `Branch` and `Pos` (see below), indicating the raceme-specific flower position.

- `Pos` is the individual flower position within each raceme.
- `FL` is the length of flower.
- `FD` is the diameter of the flower where the banner petal attaches.
- `B` is the length (height) of the banner petal.
- `Date` is the date of flower opening, and when the three floral measurements were made.
- `flw_date` is the numerical day of the flowering season (1-17) the flower opened.
- `FlwFate` is whether or not a flower produced fruit (0 = no, 1 = yes).
- `seeds` is the number of seed produced in a given fruit.
- `aborted` is the number of aborted embryos.
- `unfert` is the number of unfertilized ovules.
- `Notes` records any specific notes for a given flower.
- `flw_vol` is flower volume as approximated as a cone ($V = \frac{1}{3}\pi\frac{FD^2}{2}FL$)

3 Standardized Linear Selection (e.g., Lande and Arnold (1983))

3.1 Relative fitness (seeds)

```
#make sure PlantID is a factor
data$PlantID<- as.factor(data$PlantID)

#calculate total seed set (fitness) at plant level
plant.seeds<- aggregate(data$seeds, by=list(data$PlantID), sum)

#reset column names
colnames(plant.seeds)<- c("PlantID", "tot_seeds")

#calculate relative fitness
plant.seeds$rel_seeds<- plant.seeds$tot_seeds/(mean(plant.seeds$tot_seeds, na.rm=T))

#Check
head(plant.seeds)
```

```
##   PlantID tot_seeds rel_seeds
## 1       1        16  1.412804
## 2       2        21  1.854305
## 3       3        21  1.854305
## 4       4         0  0.000000
## 5       5         0  0.000000
## 6       6         0  0.000000
```

3.2 Standardized traits

First need to calculate mean values for each floral trait, and then subtract the mean and divide by the trait standard deviation to standardize each traits for each individual plant.

```
#First calculate mean trait value for each trait (yes, not efficient, but I like to see the steps)
mean.B<- aggregate(data$B, by=list(data$PlantID), mean, na.rm=T)
mean.B$Group.1 <- NULL
colnames(mean.B)<- "mean.B"

mean.FL<- aggregate(data$FL, by=list(data$PlantID), mean, na.rm=T)
mean.FL$Group.1 <- NULL
colnames(mean.FL)<- "mean.FL"

mean.FD<- aggregate(data$FD, by=list(data$PlantID), mean, na.rm=T)
colnames(mean.FD)<- c("PlantID", "mean.FD")
```

Merge into a single dataframe (I know this is not efficient, I like to see the steps) with relative seed set

```
traits<- cbind(mean.B, mean.FL, mean.FD)

# add relative seed set
sel.data<- merge(traits, plant.seeds)

#check
sel.data
```

```
##    PlantID   mean.B   mean.FL  mean.FD tot_seeds rel_seeds
## 1        1 4.384872  8.513333 2.654615        16 1.4128035
## 2       10 6.617778  8.874444 3.336667         5 0.4415011
## 3       11 6.621667  8.786667 3.090000         0 0.0000000
## 4       12 7.206667 10.253333 3.389333         0 0.0000000
## 5       13 6.604706  9.202353 3.228235         0 0.0000000
## 6       14 5.826667  7.611667 3.036667         0 0.0000000
## 7       15 5.108750  8.624750 2.938750        45 3.9735099
## 8       16 6.948000  9.286000 3.758000         0 0.0000000
## 9       17 5.388571  9.061905 2.906190         0 0.0000000
## 10      18 4.353095  8.365714 2.548571        14 1.2362031
## 11      19 4.856389  8.720278 2.658056        19 1.6777042
## 12       2 5.508214  9.956071 3.398929        21 1.8543046
## 13      20 6.661429  9.056250 2.978750         5 0.4415011
## 14      21 6.302857  9.892857 3.171429        19 1.6777042
## 15      22 5.687857  9.230000 2.846429        10 0.8830022
## 16      23 6.930909 10.366364 3.241818         6 0.5298013
## 17      24 5.751667  9.445833 3.019167         0 0.0000000
## 18      25 6.131818  8.886364 3.013636        36 3.1788079
## 19      26 6.278095  9.806667 3.033333         0 0.0000000
## 20      27 6.824000 10.014000 2.988000        14 1.2362031
## 21      28 5.033333  8.496667 2.744000         7 0.6181015
## 22      29 4.633750  8.248750 2.838750         0 0.0000000
## 23       3 5.716111  9.909167 3.252500        21 1.8543046
## 24      30 5.400000  8.715556 2.953333        15 1.3245033
## 25      31 5.473333  8.485000 2.666667         8 0.7064018
## 26      32 4.132222  8.714815 2.558148        15 1.3245033
## 27      33 4.903200  8.210400 2.783600        10 0.8830022
## 28      34 4.875789  8.818421 2.684211        15 1.3245033
## 29      35 4.104706  7.813529 2.659412        43 3.7969095
## 30      36 4.860000  8.016429 2.556429         0 0.0000000
## 31      37 4.471667  8.850000 2.684444        17 1.5011038
## 32      38 3.898750  8.801250 2.686875        37 3.2671082
## 33      39 3.831429  7.962449 2.443673        32 2.8256071
## 34       4 5.460000 10.010476 3.026190         0 0.0000000
## 35      40 3.098571  7.821429 2.265714         4 0.3532009
## 36       5 6.362000  9.179000 3.379000         0 0.0000000
## 37       6 8.326667 10.480000 4.475000         0 0.0000000
## 38       7 6.560556  9.313333 3.035000        15 1.3245033
## 39       8 6.854706  8.866471 3.434706         0 0.0000000
## 40       9 7.553333  9.436667 3.492222         4 0.3532009
```

Now need to standardize individual plant mean (from above).

```
#Calculate total (population) mean for each trait
sel.data$B_z<- (sel.data$mean.B - mean(sel.data$mean.B, na.rm = T))/sd(sel.data$mean.B, na.rm = T)
sel.data$FL_z<- (sel.data$mean.FL - mean(sel.data$mean.FL, na.rm = T))/sd(sel.data$mean.FL, na.rm = T)
sel.data$FD_z<- (sel.data$mean.FD - mean(sel.data$mean.FD, na.rm = T))/sd(sel.data$mean.FD, na.rm = T)
```

3.3 Covariates

```
# total flowers
tot.flw<- aggregate(data$PosSeq, by=list(data$PlantID), max)
tot.flw$Group.1<- NULL

#total branches (racemes)
tot.branch<- aggregate(data$Branch, by=list(data$PlantID), max)
tot.branch$Group.1<- NULL
```

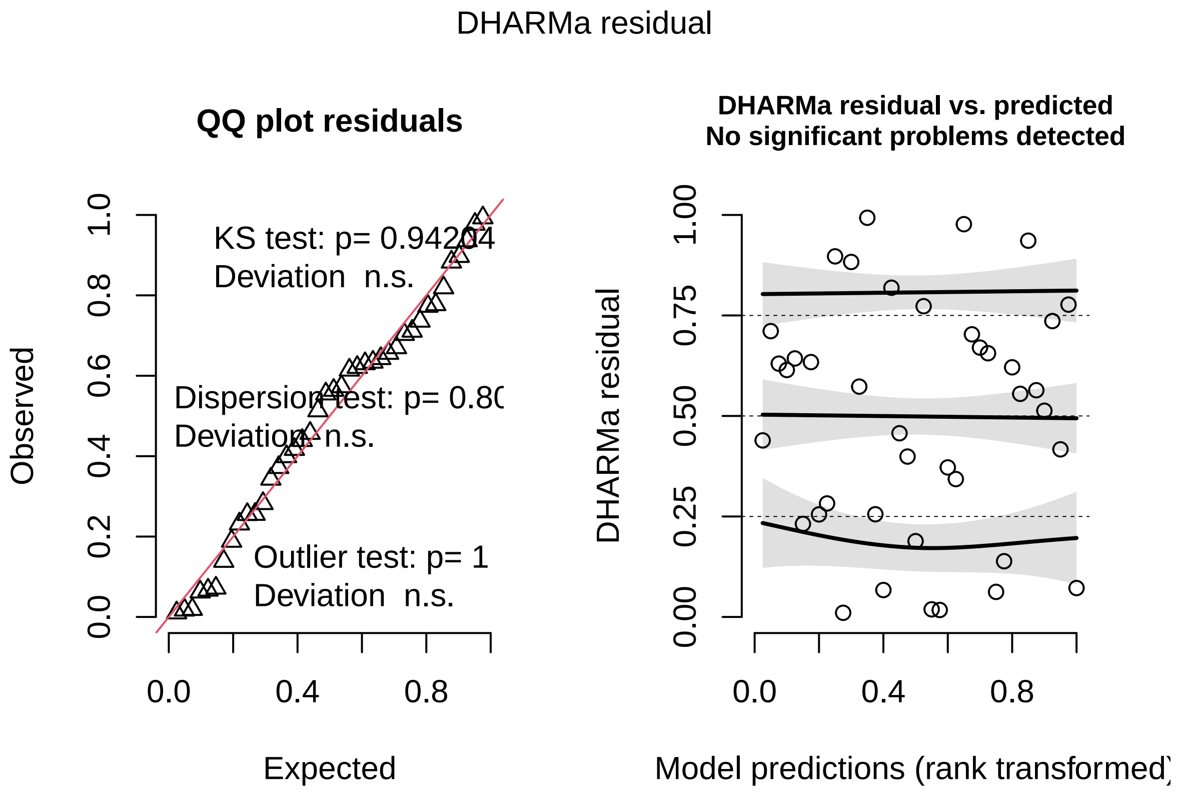
3.4 Estimate (β)

Start with a poisson distribution.

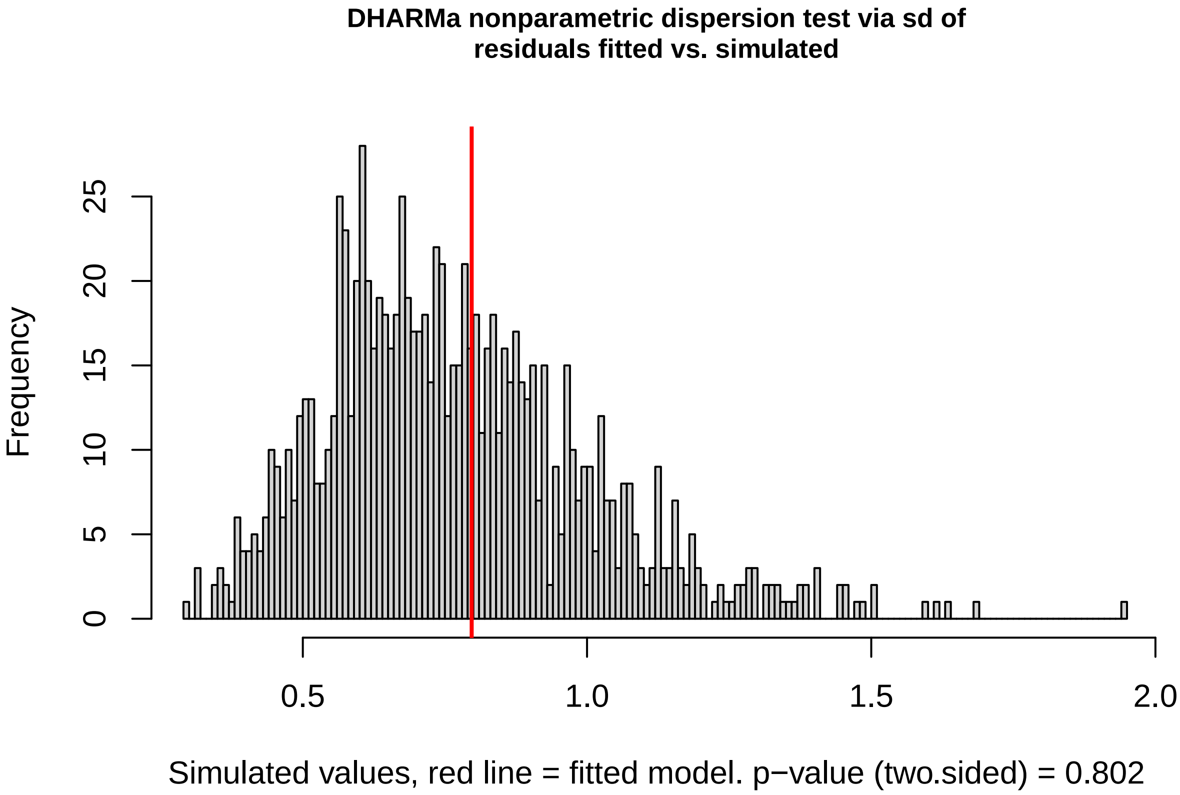
```
# model with standardized traits as fixed effects, and palntID as random
# Fit Poisson model
fit_pois <- glmmTMB(rel_seeds ~ B_z + FL_z + FD_z,
                    data = sel.data, family = poisson)
```

```
## Warning in glmmTMB(rel_seeds ~ B_z + FL_z + FD_z, data = sel.data, family =
## poisson): non-integer counts in a poisson model
```

```
# Model diagnostics using DHARMa
sim_resid <- simulateResiduals(fit_pois, n = 1000)
plot(sim_resid)
```



```
# Test for overdispersion
testDispersion(sim_resid)
```

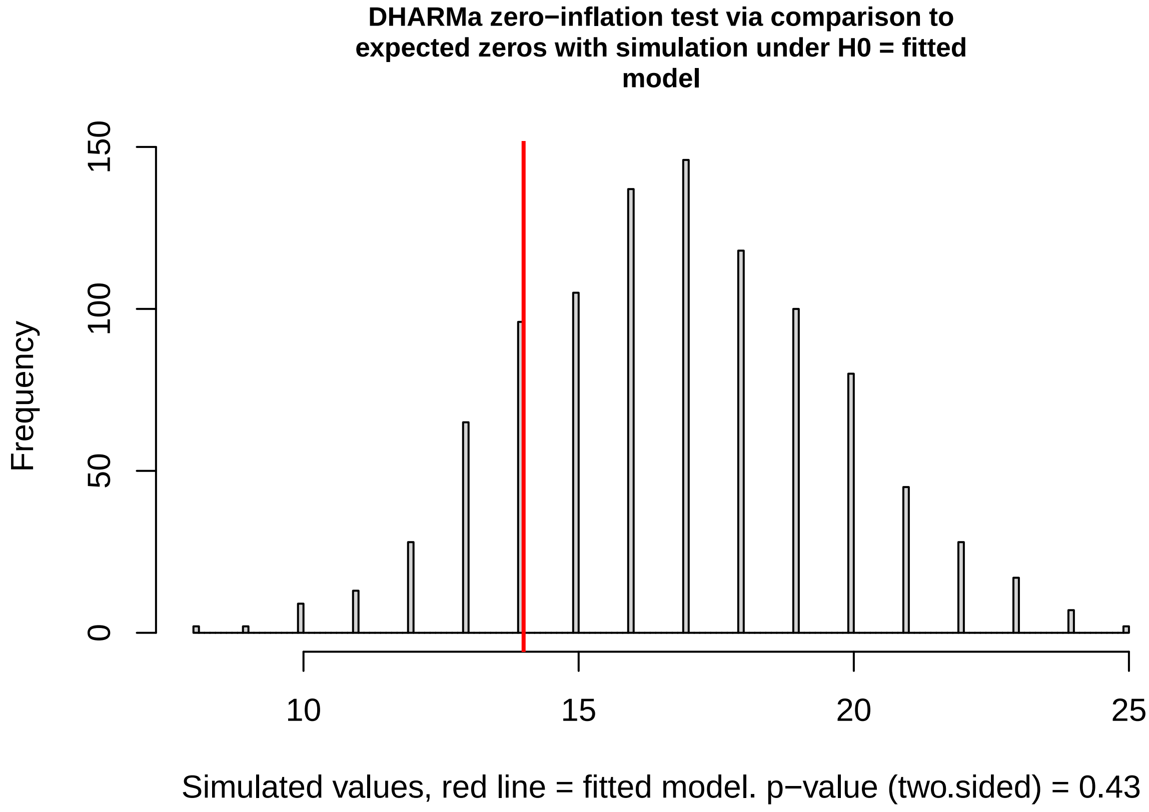


```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1.0352, p-value = 0.802
## alternative hypothesis: two.sided
```

```r
summary(fit_pois)
```

```
##  Family: poisson  ( log )
## Formula:          rel_seeds ~ B_z + FL_z + FD_z
## Data: sel.data
##
##      AIC       BIC    logLik -2*log(L)  df.resid
##    105.4     112.2     -48.7      97.4        36
##
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.14262    0.18035  -0.791   0.4290
## B_z         -0.73068    0.36197  -2.019   0.0435 *
## FL_z         0.20030    0.25371   0.789   0.4298
## FD_z         0.08304    0.41486   0.200   0.8414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Formal test for zero inflation
testZeroInflation(sim_resid) # not significant
```



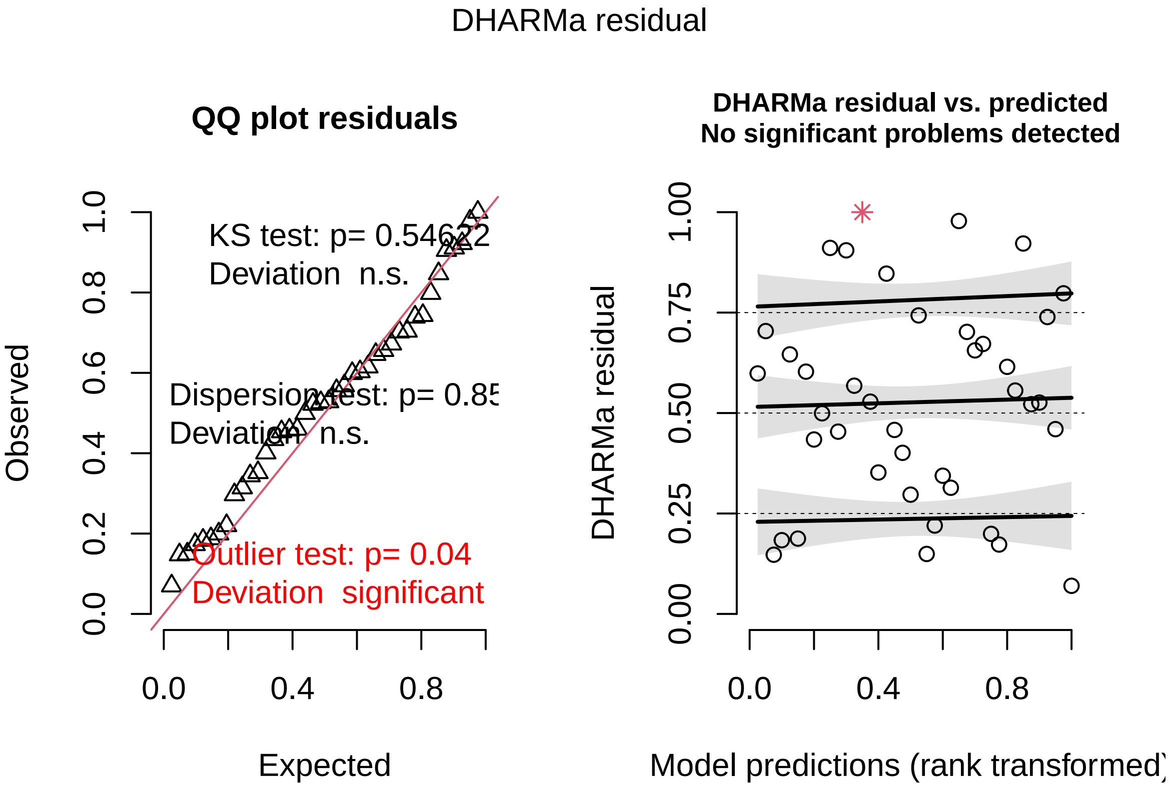
```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 0.8316, p-value = 0.43
## alternative hypothesis: two.sided
```

The above model looks like a good fit (according to diagnostics), and not over dispersed. However, try fitting with a negative binomial distribution and compare AIC across two models.

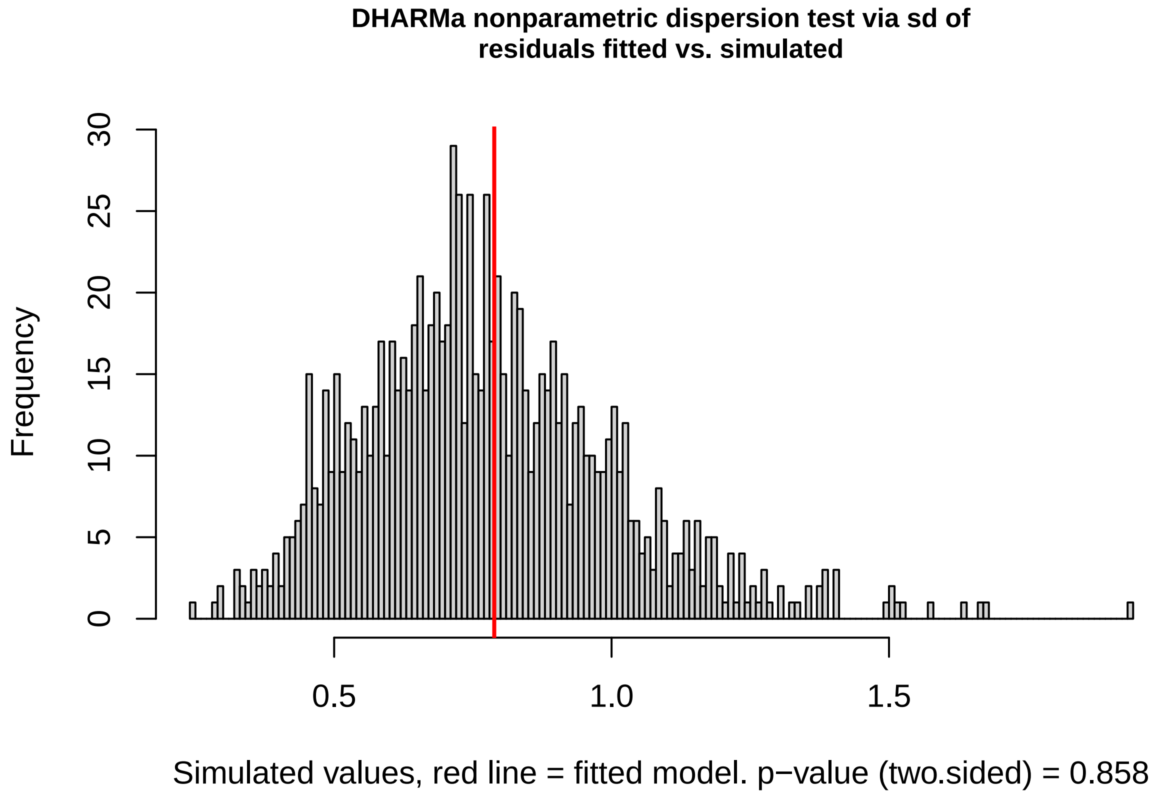
```
# Fit negative binomial model
fit_nb <- glmmTMB(rel_seeds ~ B_z + FL_z + FD_z,
                    data = sel.data, family = nbinom2)
```

```
## Warning in glmmTMB(rel_seeds ~ B_z + FL_z + FD_z, data = sel.data, family =
## nbinom2): non-integer counts in a nbinom2 model
```

```
# Model diagnostics using DHARMa
sim_resid <- simulateResiduals(fit_nb, n = 1000)
plot(sim_resid)
```



```
# Test for overdispersion
testDispersion(sim_resid)
```

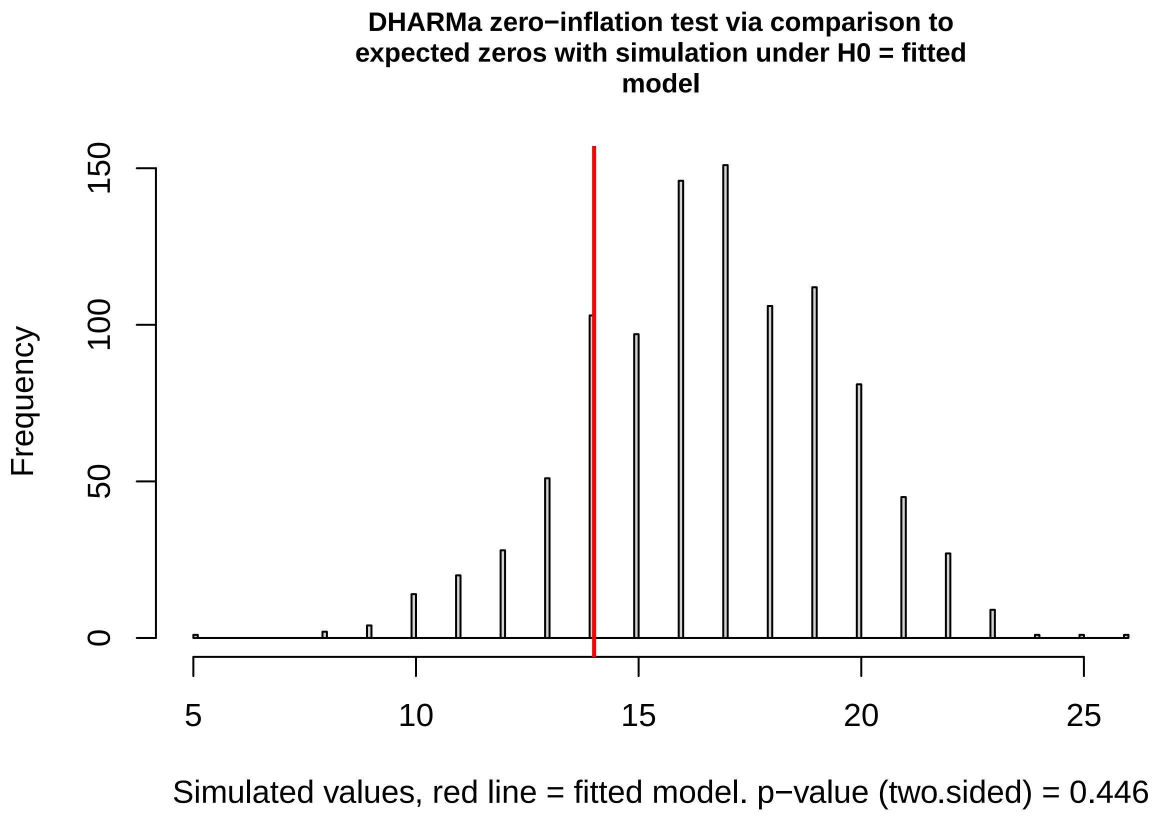


```
##
##  DHARMa nonparametric dispersion test via sd of residuals fitted vs.
##  simulated
##
## data:  simulationOutput
## dispersion = 1.0168, p-value = 0.858
## alternative hypothesis: two.sided
```

```
summary(fit_nb)
```

```
##  Family: nbinom2  ( log )
## Formula:          rel_seeds ~ B_z + FL_z + FD_z
## Data: sel.data
##
##      AIC       BIC    logLik -2*log(L)  df.resid
##    107.4     115.8     -48.7      97.4        35
##
##
## Dispersion parameter for nbinom2 family (): 1.32e+08
##
## Conditional model:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.14262    0.18035  -0.791   0.4291
## B_z         -0.73068    0.36197  -2.019   0.0435 *
## FL_z         0.20030    0.25371   0.789   0.4298
## FD_z         0.08304    0.41486   0.200   0.8414
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Formal test for zero inflation
testZeroInflation(sim_resid) # not significant
```



```
##
##  DHARMa zero-inflation test via comparison to expected zeros with
##  simulation under H0 = fitted model
##
## data:  simulationOutput
## ratioObsSim = 0.83867, p-value = 0.446
## alternative hypothesis: two.sided
```

```
AIC(fit_pois, fit_nb)
```

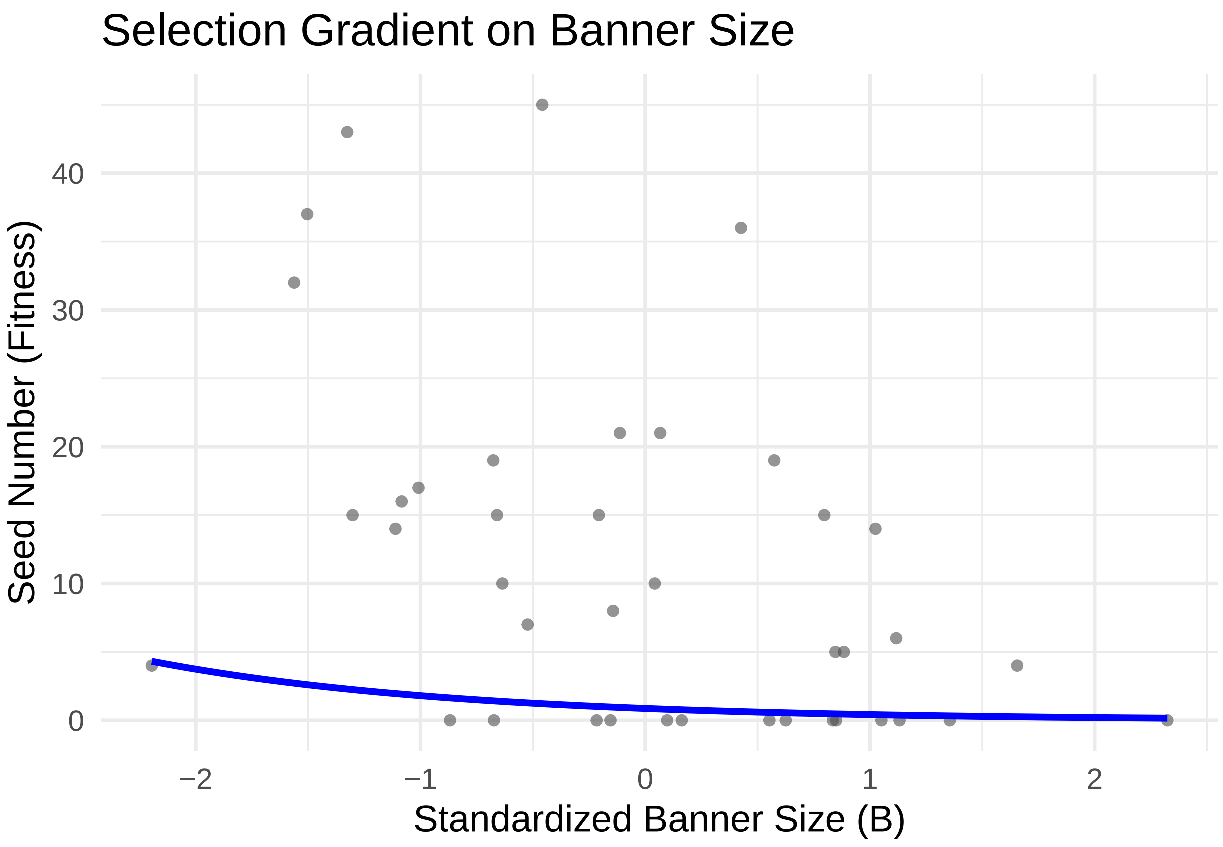
```
##          df      AIC
## fit_pois  4 105.4014
## fit_nb    5 107.4014
```

Both models fit well, and show the same pattern (significant effect of Banner height). As the AIC is slightly smaller with Poisson distribution, use this model.

Now produce a quick plot of the significant effect of banner height. A rather underwhelming figure.

```
# Create prediction data over the range of standardized B
newdata <- data.frame(
  B_z = seq(min(sel.data$B_z), max(sel.data$B_z), length.out = 100),
  FL_z = 0,  # Hold other traits at their means (0 after standardization)
  FD_z = 0
)

# Predict expected seed number from the Poisson model
newdata$predicted_seeds <- predict(fit_pois, newdata, type = "response")

# Plot observed data and predicted curve
ggplot(sel.data, aes(x = B_z, y = tot_seeds)) +
  geom_point(alpha = 0.6, color = "gray30") +
  geom_line(data = newdata, aes(x = B_z, y = predicted_seeds), color = "blue", size = 1.2) +
  labs(
    x = "Standardized Banner Size (B)",
    y = "Seed Number (Fitness)",
    title = "Selection Gradient on Banner Size"
  ) +
  theme_minimal(base_size = 14)
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```



4 Floral Integration

Floral integration was described with correlation coefficients among floral traits. To explore if correlations among traits change across racemes, we compared correlation coefficients on racemes 1-5. To facilitate comparison among racemes, the first five flowers were used to calculate these correlations. We calculated both within racemes (first five flowers), and among racemes (same position across first five racemes).

```
# define function to extract, r, se, and P-value
get_cor_stats <- function(x, y) {
  ct <- cor.test(x, y, method = "pearson")
  r <- ct$estimate
  n <- sum(complete.cases(x, y))
  se <- sqrt((1 - r^2) / (n - 2))
  data.frame(correlation = r, se = se, p_value = ct$p.value)
}
```

4.1 Withn raceme integration

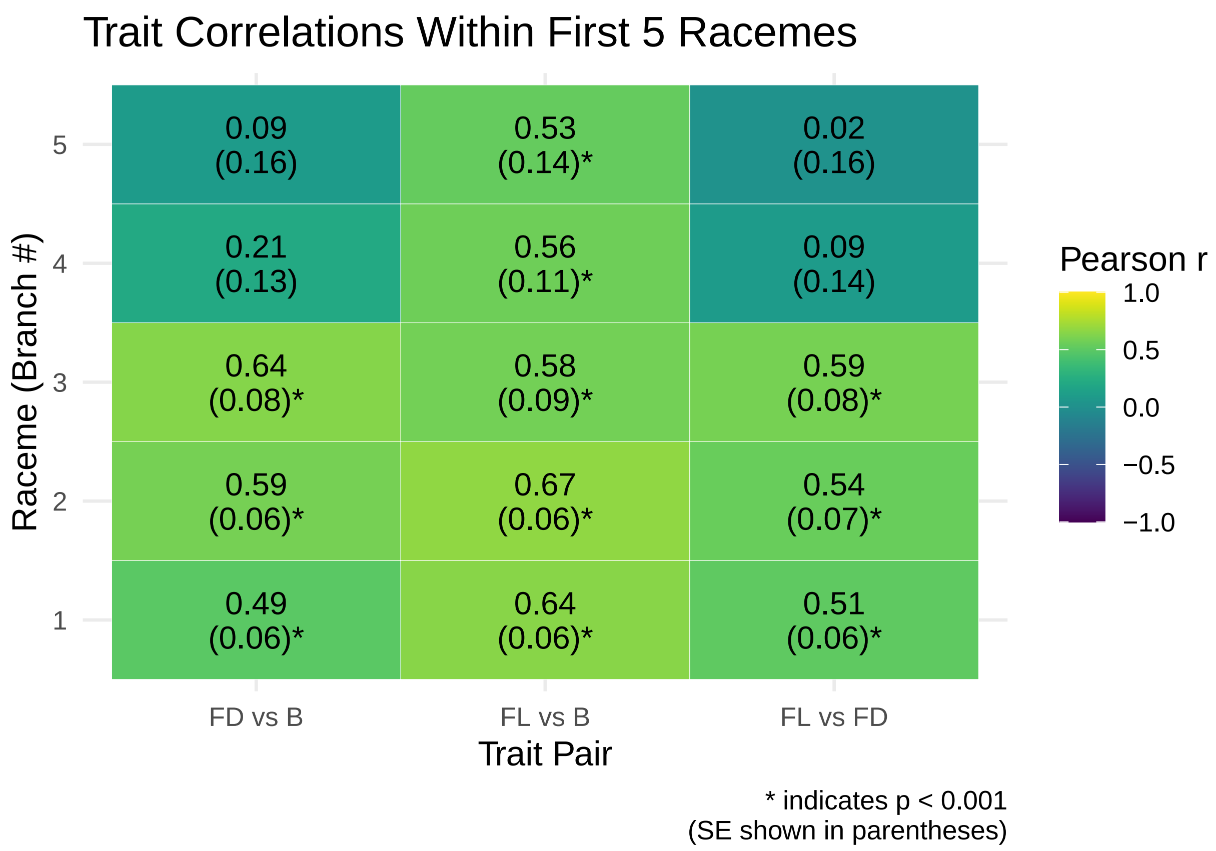
Calculate within-inflorescence (raceme) floral integration.

```
#within raceme integration
within_raceme <- data %>%
  filter(Branch %in% 1:5, Pos %in% 1:5) %>%
  group_by(Branch) %>%
  group_modify(~{
    df <- .
    bind_rows(
      get_cor_stats(df$FL, df$FD) %>% mutate(pair = "FL vs FD"),
      get_cor_stats(df$FL, df$B)  %>% mutate(pair = "FL vs B"),
      get_cor_stats(df$FD, df$B)  %>% mutate(pair = "FD vs B")
    )
  }) %>%
  ungroup() %>%
  select(Branch, pair, correlation, se, p_value)

within_raceme
```

```
## # A tibble: 15 x 5
##    Branch pair     correlation     se  p_value
##     <int> <chr>          <dbl>  <dbl>    <dbl>
##  1      1 FL vs FD      0.508  0.0625 5.48e-14
##  2      1 FL vs B       0.643  0.0555 8.08e-24
##  3      1 FD vs B       0.491  0.0632 4.86e-13
##  4      2 FL vs FD      0.542  0.0671 1.57e-13
##  5      2 FL vs B       0.671  0.0591 3.44e-22
##  6      2 FD vs B       0.586  0.0646 4.60e-16
##  7      3 FL vs FD      0.588  0.0843 4.51e-10
##  8      3 FL vs B       0.578  0.0851 1.04e- 9
##  9      3 FD vs B       0.636  0.0804 5.53e-12
## 10      4 FL vs FD      0.0902 0.136  5.09e- 1
## 11      4 FL vs B       0.562  0.113  6.43e- 6
## 12      4 FD vs B       0.209  0.133  1.22e- 1
## 13      5 FL vs FD      0.0176 0.164  9.15e- 1
## 14      5 FL vs B       0.531  0.139  5.02e- 4
## 15      5 FD vs B       0.0908 0.164  5.82e- 1
```

Now make a nice little table with heatmap features to show pattern of floral integration with racemes, across the first five racemes.

```
# prepare standard errors for inclusoin in table/heatmap
within_raceme <- within_raceme %>%
  mutate(sig = ifelse(p_value < 0.05, "*", ""),
         label = sprintf("%.2f\n(%.2f)%s", correlation, se, sig))

ggplot(within_raceme, aes(x = pair, y = factor(Branch), fill = correlation)) +
  geom_tile(color = "white") +
  geom_text(aes(label = label), color = "black", size = 4.2, lineheight = 0.9) +
  scale_fill_viridis(name = "Pearson r", limits = c(-1, 1)) +
  labs(
    title = "Trait Correlations Within First 5 Racemes",
    x = "Trait Pair", y = "Raceme (Branch #)",
    caption = "* indicates p < 0.001\n(SE shown in parentheses)"
  ) +
  theme_minimal(base_size = 13)
```



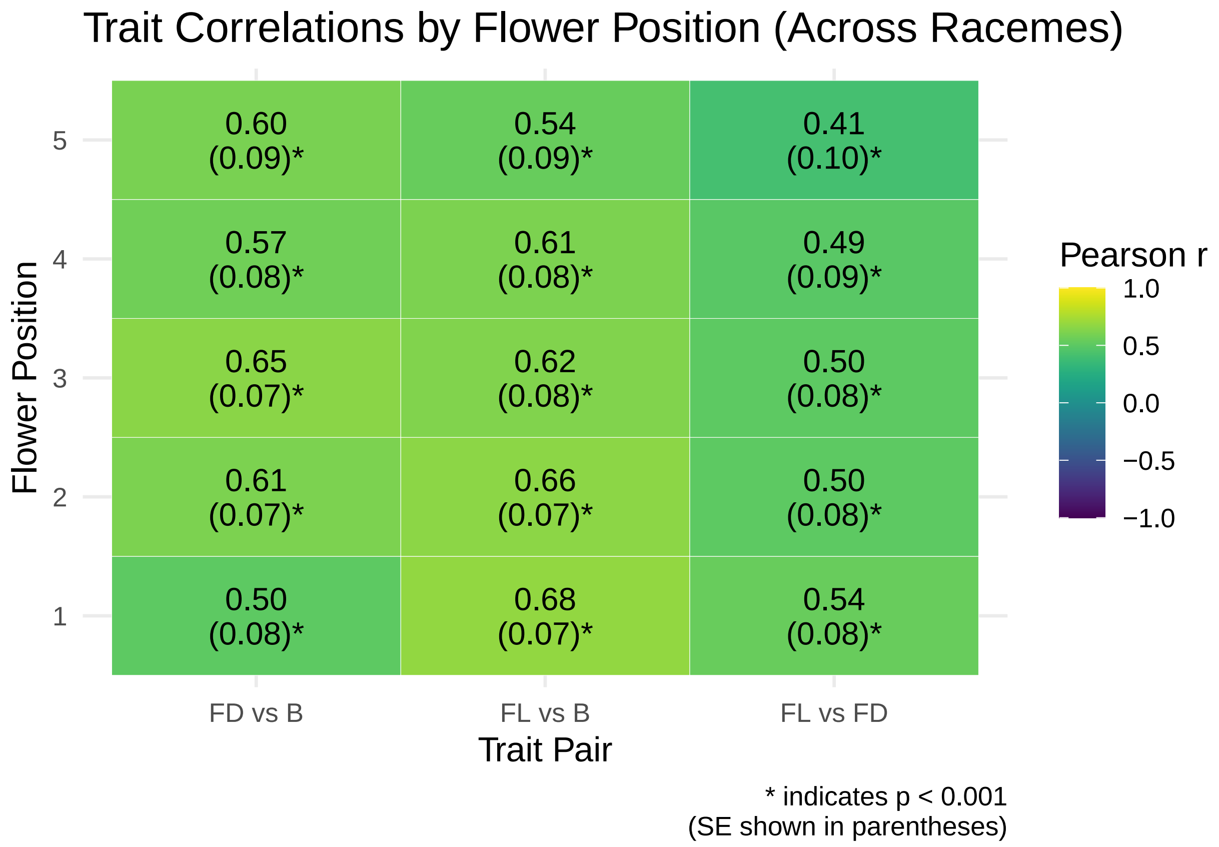
4.2 Among raceme integration

```
# among racemes
across_pos <- data %>%
  filter(Branch %in% 1:5, Pos %in% 1:5) %>%
  group_by(Pos) %>%
  group_modify(~{
    df <- .
    bind_rows(
      get_cor_stats(df$FL, df$FD) %>% mutate(pair = "FL vs FD"),
      get_cor_stats(df$FL, df$B)  %>% mutate(pair = "FL vs B"),
      get_cor_stats(df$FD, df$B)  %>% mutate(pair = "FD vs B")
    )
  }) %>%
  ungroup() %>%
  select(Pos, pair, correlation, se, p_value)

across_pos
```

```
## # A tibble: 15 x 5
##       Pos pair     correlation     se  p_value
##     <int> <chr>          <dbl>  <dbl>    <dbl>
##  1      1 FL vs FD       0.539 0.0785 3.54e-10
##  2      1 FL vs B        0.677 0.0686 5.25e-17
##  3     1 FD vs B        0.502 0.0806 7.90e- 9
##  4     2 FL vs FD       0.502 0.0803 6.83e- 9
##  5     2 FL vs B        0.657 0.0700 6.79e-16
##  6     2 FD vs B        0.607 0.0738 3.32e-13
##  7     3 FL vs FD       0.502 0.0836 2.67e- 8
##  8     3 FL vs B        0.623 0.0756 4.62e-13
##  9     3 FD vs B        0.651 0.0734 1.84e-14
## 10     4 FL vs FD       0.486 0.0853 1.11e- 7
## 11     4 FL vs B        0.607 0.0776 4.27e-12
## 12     4 FD vs B        0.570 0.0802 1.52e-10
## 13     5 FL vs FD       0.407 0.0979 7.58e- 5
## 14     5 FL vs B        0.536 0.0905 6.20e- 8
## 15     5 FD vs B        0.596 0.0861 7.09e-10
```

Similar table as before, but now for same flower position (1-5) position across subsequently produced racemes.

```
across_pos <- across_pos %>%
  mutate(sig = ifelse(p_value < 0.05, "*", ""),
         label = sprintf("%.2f\n(%.2f)%s", correlation, se, sig))

ggplot(across_pos, aes(x = pair, y = factor(Pos), fill = correlation)) +
  geom_tile(color = "white") +
  geom_text(aes(label = label), color = "black", size = 4.2, lineheight = 0.9) +
  scale_fill_viridis(name = "Pearson r", limits = c(-1, 1)) +
  labs(
    title = "Trait Correlations by Flower Position (Across Racemes)",
    x = "Trait Pair", y = "Flower Position",
    caption = "* indicates p < 0.001\n(SE shown in parentheses)"
  ) +
  theme_minimal(base_size = 13)
```



References

Harder, Lawrence D., Marina M. Strelin, Ilona C. Clocher, Mason W. Kulbaba, and Marcelo A. Aizen. 2019. "The Dynamic Mosaic Phenotypes of Flowering Plants." *New Phytologist* 224 (3): 1021–34. https://doi.org/10.1111/nph.15916.

Kulbaba, Mason W., Ilona C. Clocher, and Lawrence D. Harder. 2017. "Inflorescence Characteristics as Function-Valued Traits: Analysis of Heritability and Selection on Architectural Effects." *Journal of Systematics and Evolution* 55 (6): 559–65. https://doi.org/10.1111/jse.12252.

Lande, Russell, and Stevan J. Arnold. 1983. "The Measurement of Selection on Correlated Characters." *Evolution* 37 (6): 1210–26. https://doi.org/10.1111/j.1558-5646.1983.tb00236.x.