



Supplementary material

Supplementary Data I. The R code that specify the dataset and statistical analyses.

Here, we present the lines of R code that specify our statistical analyses. The analyzed data are specified within the code lines themselves.

First, we assess whether the types of floral visitors were different between the studied environments (rupestrian field and typical cerrado) and between the periods of observation (predawn and midday).

For this purpose, we conducted a PERMANOVA considering the types of floral visitors as response variables and the environments and periods of observation as predictor variables.

Install the 'vegan' package if you have not already installed it in your computer.

```
install.packages("vegan")
```

Then, activate the 'vegan' package.

```
library(vegan)
```

Enter the predictor variables (period and environment) and put them in a single predictor matrix (predictor_variables).

```
period <- c("predawn", "midday", "predawn", "midday")
```

```
environment <- c("rupestrian_field", "rupestrian_field", "typical_cerrado", "typical_cerrado")
```

```
predictor_variables <- data.frame(period, environment); predictor_variables
```

Enter the response variables (the number of each type of floral visitor observed in each period and environment) and put them in a single response matrix (response_variables)

```
wasp <- c(37, 35, 0, 10)
```

```
ant <- c(11, 9, 25, 22)
```

```
small_bee <- c(3, 0, 33, 6)
```

```
large_bee <- c(3, 0, 3, 1)
```

```
butterfly <- c(1, 1, 6, 1)
```

```
moth <- c(0, 3, 0, 0)
```

```
dipteran <- c(3, 1, 6, 7)
```

```
beetle <- c(2, 0, 1, 0)
```

```
hummingbird <- c(0, 0, 9, 4)
```

```
response_variables <- data.frame(wasp, ant, small_bee, large_bee, butterfly, moth, dipteran, beetle, hummingbird);  
response_variables
```

Assess how response variables differ according the period of observation, using a 10000 permutations PERMANOVA with a Bray-Curtis metric of dissimilarity.

```
PERMANOVA1 <- adonis2(response_variables ~ period, method = "bray",  
  permutations = 10000, data = predictor_variables); PERMANOVA1
```

Assess how response variables differ according the environment of observation, using a 10000 permutations PERMANOVA with a Bray-Curtis metric of dissimilarity.

```
PERMANOVA2 <- adonis2(response_variables ~ environment, method = "bray",
  permutations = 10000, data = predictor_variables); PERMANOVA2
```

To proceed with the statistical models, we need to enter all the variables involved in this research and organize them into a single spreadsheet (data).

The individual plants. Each individual receives an identity letter (from "a" to "l"). Each plant individual was observed twice (one time at predawn and one time at the midday).

```
individual <- c("a", "b", "c", "d", "e", "f",
  "a", "b", "c", "d", "e", "f",
  "g", "h", "i", "j", "k", "l",
  "g", "h", "i", "j", "k", "l")
```

The periods of observation (predawn and midday) of floral visits on the individual plants.

```
period <- c("a_predawn", "a_predawn", "a_predawn", "a_predawn", "a_predawn", "a_predawn",
  "b_midday", "b_midday", "b_midday", "b_midday", "b_midday", "b_midday",
  "a_predawn", "a_predawn", "a_predawn", "a_predawn", "a_predawn", "a_predawn",
  "b_midday", "b_midday", "b_midday", "b_midday", "b_midday", "b_midday")
```

The types of environment (rupestrian field and typical cerrado) where the plant individuals were located.

```
environment <- c("rupestrian_field", "rupestrian_field", "rupestrian_field", "rupestrian_field", "rupestrian_field",
  "rupestrian_field", "rupestrian_field", "rupestrian_field", "rupestrian_field", "rupestrian_field", "rupestrian_field",
  "typical_cerrado", "typical_cerrado", "typical_cerrado", "typical_cerrado", "typical_cerrado", "typical_cerrado",
  "typical_cerrado", "typical_cerrado", "typical_cerrado", "typical_cerrado", "typical_cerrado", "typical_cerrado")
```

The number of floral visits counted for each plant individual, separated by period.

```
visits <- c(9, 4, 4, 17, 10, 13,
  9, 4, 3, 13, 11, 12,
  10, 14, 19, 17, 16, 9,
  6, 4, 6, 17, 11, 5)
```

The water potential (in MPa) in each individual plant, separated by period.

```
water_potential <- c(-1.20, -1.20, -1.20, -0.80, -1.00, -1.10,
  -1.20, -1.00, -1.30, -1.10, -1.40, -1.20,
  -0.50, -0.80, -0.10, -0.40, -0.40, -0.10,
  -1.13, -0.70, -1.12, -1.14, -0.60, -1.19)
```

The number of flowers counted for each plant individual, separated by period.

```
flowers <- c(15, 6, 35, 16, 9, 1,
  8, 7, 25, 13, 6, 12,
  2, 13, 4, 6, 10, 4,
  19, 9, 2, 5, 4, 6)
```

The number of flowers buds counted for each plant individual, separated by period.

```
buds <- c(111, 165, 133, 78, 88, 86,
  122, 81, 100, 113, 64, 61,
  179, 20, 153, 148, 80, 39,
  205, 29, 139, 104, 90, 46)
```

Merely to facilitate the editing of the graphics, we inserted a group categorization.

a - plant individuals at rupestrian field during the predawn

```

# b - plant individuals at rupestrian field during the midday
# c - plant individuals at typical cerrado during the predawn
# d - plant individuals at typical cerrado during the midday
group <- c("a", "a", "a", "a", "a", "a",
          "b", "b", "b", "b", "b", "b",
          "c", "c", "c", "c", "c", "c",
          "d", "d", "d", "d", "d", "d")

# Organize all these variables in single data set (data) and, then, specify which of them are factors.
data <- data.frame(individual, period, environment, visits, water_potential, flowers, buds, group) ;data

data$individual <- as.factor(data$individual)
data$period <- as.factor(data$period)
data$environment <- as.factor(data$environment)
data$group <- as.factor(data$group)

# In the following Generalized Estimating Equations (GEE), the 'geepack' package cannot correctly compute the model
clusters if the provided variable is a factor.
# For this package limitation, we inserted also another 'individual identity' in the data set using numbers (from 1 to 12).
# In practice, the 'geepack' package will correctly recognize them as 12 clusters (despite they are numbers) with size 2
each one.
data$individual_cluster <- c(rep(1:6, each = 2), rep(7:12, each = 2))

# Check the data set structure.
summary(data)
data

### MODEL 1 ###

# This model is a Generalized Estimating Equation (GEE).
# We assessed the influences of the environment types, periods of observation, and their interaction (predictors) on the
water potential of the plants (response).
# Since plant individuals were observed twice, their identities were considered as clusters in this model where observations
of the same individual were dependent.
# We considered a Gaussian residual distribution in this model.

# Install the 'geepack' package if you have not already installed it in your computer.

install.packages("geepack")

# Then, activate the 'geepack' package.

library(geepack)

# Set the Model 1.

model1 <- geeglm(water_potential ~ environment + period + flowers + buds + environment:period,
                 family = gaussian(link="identity"),
                 id = individual_cluster,
                 corstr = "exchangeable",
                 data = data) ;model1

# Inspect visually the residual distribution using a quantil-quantil plot and histogram.

```



```

model1_residuals <- residuals(model1, type = "pearson")
qqnorm(model1_residuals)
qqline(model1_residuals, col = "red")

hist(residuals(model1))

# Since we considered this residual distribution acceptable for a Gaussian family, we required the model results.

summary(model1)

# Given the model significance, conduct a post-hoc test for the Model 1.

# Install the 'emmeans' and 'multcomp' packages if you have not already installed them in your computer.

install.packages(c("emmeans", "multcomp"))

# Then, activate the 'emmeans' and 'multcomp' packages.

library(emmeans)
library(multcomp)

# Specify the post-hoc test.

em_model1 <- emmeans(model1,
                      specs = pairwise ~ environment:period,
                      adjust = "tukey")

# Require the results of the post-hoc test.

summary(em_model1$contrasts, infer = TRUE)

# Edit a figure to represent the comparisons made in the Model 1. This is the Figure 4a in the article.

# Install the 'ggplot2' package if you have not already installed it in your computer.

install.packages("ggplot2")

# Then, activate the 'ggplot2' package.

library(ggplot2)

Figure4a <- ggplot(data=data, aes(x=group ,y=water_potential, fill=environment)) +
  geom_boxplot(width=0.6, alpha=0.7, size=1, show.legend=TRUE) +
  theme_classic() +
  labs(x="Period", y="Water potential (MPa)", fill="Environment") +
  theme(text=element_text(size=16)) +
  scale_fill_manual(labels=c("rupestrian field", "typical cerrado"), values=c("#FF4500", "#FFD700")) +
  scale_x_discrete(labels=c("predawn", "midday", "predawn", "midday")) +
  geom_hline(yintercept=0, linewidth=1, linetype="dashed") +
  annotate("text", label=c("A", "A", "B", "A"), x=c("a", "b", "c", "d"), y=-0.05);Figure4a

```

MODEL 2

This model is a Generalized Estimating Equation (GEE).
 # We assessed the influences of the environment types and periods of observation (predictors) on the number of floral visits on the plants (response).
 # The interaction between the environments and periods (environment:period) was not included in this model given the lack significance.
 # Since plant individuals were observed twice, their identities were considered as clusters in this model where observations of the same individual were dependent.
 # We firstly considered a poisson residual distribution in this model.
 # However, given its overdispersion of residuals, we considered a model with negative binomial residual distribution instead.

The poisson GEE model (Model 2):

```
model2 <- geeglm(visits ~ environment + period + flowers + buds,
  family = poisson(link="log"),
  id = individual_cluster,
  corstr = "exchangeable",
  data = data) ;model2
```

Assessing the residual distribution of the Model 2:

```
res_pearson <- residuals(model2, type = "pearson")
dispersion <- sum(res_pearson^2) / model2$df.residual

plot(fitted(model2), res_pearson,
  xlab = "Adjusted", ylab = "Pearson Residuals",
  main = paste("poisson GEE - Dispersion =", round(dispersion, 3)),
  pch = 19, col = rgb(0, 0, 1, 0.5))
abline(h = 0, col = "red", lwd = 2)
abline(h = c(-2, 2), col = "gray", lty = 2)
```

Given the overdispersion of residuals of the Model 2, we conducted a similar model (Model 2 nb) with a negative binomial residual distribution:

Since the negative binomial residual distribution is not implemented in the 'geepack' package, we specified the cluster using the 'sandwich' and 'lmtest'.

Then, we implemented the cluster in a negative binomial GLM built with the 'MASS' package.

Install the 'MASS', 'sandwich', and 'lmtest' packages if you have not already installed them in your computer.

```
install.packages(c("MASS", "sandwich", "lmtest"))
```

Then, activate the 'MASS', 'sandwich', and 'lmtest' packages.

```
library(MASS)
library(sandwich)
library(lmtest)
```

Set the Model 2 nb.

```
model2_nb <- glm.nb(visits ~ environment + period + flowers + buds,
  data = data) ;model2_nb
```



```

# Then, create cluster matrixes for the plant individuals.

cluster_var <- data$individual_cluster

# Require the model results implementing the clusters.

model2_nb_cluster <- coeftest(model2_nb, vcov = vcovCL(model2_nb, cluster = cluster_var)) ; model2_nb_cluster

# Specify a post-hoc test to compare the average number of visits between the periods of survey in the previous model.

emm_model <- emmeans(model2_nb,
  specs = ~ period,
  vcov = vcovCL(model2_nb, cluster = cluster_var),
  type = "response")

# Require the post-hoc test results

pairs(emm_model)

# Edit a figure to represent the comparisons made in the previous model. This is the Figure 4b in the article.

Figure4b <- ggplot(data=data, aes(x=group ,y=visits, fill=environment)) +
  geom_boxplot(width=0.6, alpha=0.7, size=1) +
  theme_classic() +
  theme(legend.key.size=unit(1, 'cm')) +
  labs(x="Period", y="Number of flower visits", fill="Environment") +
  theme(text=element_text(size=16)) +
  scale_fill_manual(labels=c("rupestrian field", "typical cerrado"), values=c("#FF4500", "#FFD700")) +
  scale_x_discrete(labels=c("predawn", "midday", "predawn", "midday")) +
  annotate("text", label=c("A", "B", "A", "B"), x=c("a", "b", "c", "d"), y= 19.5); Figure4b

### MODEL 3 ###

# This model is a Generalized Estimating Equation (GEE).
# We assessed the influences of the environment types, number of flowers and flower buds in the inflorescences (predictors)
on the number of floral visits on the plants (response).
# Since plant individuals were observed twice, their identities were considered as clusters in this model where observations
of the same individual were dependent.
# We firstly considered a poisson residual distribution in this model.
# However, given its overdispersion of residuals, we considered a model with negative binomial residual distribution
instead.

model3 <- geeglm(visits ~ water_potential + environment + flowers + buds,
  family = poisson(link="log"),
  id = individual_cluster,
  corstr = "exchangeable",
  data = data)

# Assessing the residual distribution of the Model 3:

res_pearson <- residuals(model3, type = "pearson")
dispersion <- sum(res_pearson^2) / model3$df.residual

```

```

plot(fitted(model3), res_pearson,
     xlab = "Valores Ajustados", ylab = "Resíduos de Pearson",
     main = paste("GEE Poisson - Dispersão =", round(dispersion, 3)),
     pch = 19, col = rgb(0, 0, 1, 0.5))
abline(h = 0, col = "red", lwd = 2)
abline(h = c(-2, 2), col = "gray", lty = 2)

# Given the overdispersion of residuals of the Model 3, we conducted a similar model (Model 3 nb) with a negative
# binomial residual distribution:
# Since the negative binomial residual distribution is not implemented in the 'geepack' package, we specified the cluster
# using the 'sandwich' and 'lmtest'.
# Then, we implemented the cluster in a negative binomial GLM built with the 'MASS' package.

# Set the Model 3 nb.

model3_nb <- glm.nb(visits ~ water_potential + environment + flowers + buds,
                    data = data)

# Then, create cluster matrixes for the plant individuals.

cluster_var <- data$individual_cluster

# Require the model results implementing the clusters.

coeftest(model3_nb, vcov = vcovCL(model3_nb, cluster = cluster_var))

# Estimate the coefficient determination (pseudo-R2) of this model.

# Install the 'rsq' and 'miceadds' packages if you have not already installed them in your computer.

install.packages("rsq")
install.packages("miceadds")

# Activate the 'rsq' and 'miceadds' packages.

library(rsq)
library(miceadds)

# Respecify the previous model, but using the 'glm.cluster' function from the 'miceadds' package.
# Thus, you can obtain the necessary values to calculate the pseudo-R2 of this model.
model3_cluster <- glm.cluster(formula = visits ~ water_potential + environment + flowers + buds,
                             data = data, cluster = data$individual_cluster, family = MASS::negative.binomial(theta = model3_nb$theta))

summary(model3_cluster)

# Calculate the variance-function-based pseudo-R2 of this model.
rsq(model3_cluster$glm_res, type = "v")

# Estimate the predicted values of flower visits according to the water potential (since it was the significant predictor in
# the model) and 95% confidence intervals for these predictions.

# Install the 'ggeffects' package if you have not already installed it in your computer.

```



```

install.packages("ggeffects")

# Activate the 'ggeffects' package.

library(ggeffects)

predictions <- ggpredict(model3_nb, terms=c("water_potential")) ; predictions

confint(model3_nb, method = "Wald", level = 0.95)

# Edit a figure to represent the predicted flower visits according to the water potential. This is the Figure 5 in the article.

Figure5 <- ggplot() +
  geom_point(size=3, alpha=0.6, data=data, aes(x=water_potential, y=visits)) +
  geom_line(data=predictions, aes(x=x, y=predicted), size=1, show.legend=FALSE, colour="#FFD700") +
  geom_ribbon(data=predictions, aes(x=x, ymin=conf.low, ymax=conf.high), alpha=0.2, show.legend=FALSE, fill="#FFD700",
  colour=NA) +
  labs(x="Water potential (MPa)", y="Number of flower visits") +
  theme_classic() +
  geom_vline(xintercept=0, linewidth=1, linetype="dashed") +
  theme(text=element_text(size=16)) +
  scale_x_continuous(breaks = seq(-1.5, 0, by=0.2)) ; Figure5

```

