Autocorrelation experiment - 5 days - Mortality

Debora

2025-06-16

Load and Preprocess Datasets

We load two datasets: one with a preparation technique (repeated frond selection) and one without.

```r
options(scipen=999)
# This dataset contains replicates for which a preparation technique was performed (repeated first born
original_dataset_2 <- read.csv("https://raw.githubusercontent.com/Cuddington-Lab/thermal-experiments/ma

# This dataset contains replicates of experiments performed without a preparation technique
original_dataset_1 <- read.csv("https://raw.githubusercontent.com/Cuddington-Lab/thermal-experiments/ma
```

Combine and clean up datasets

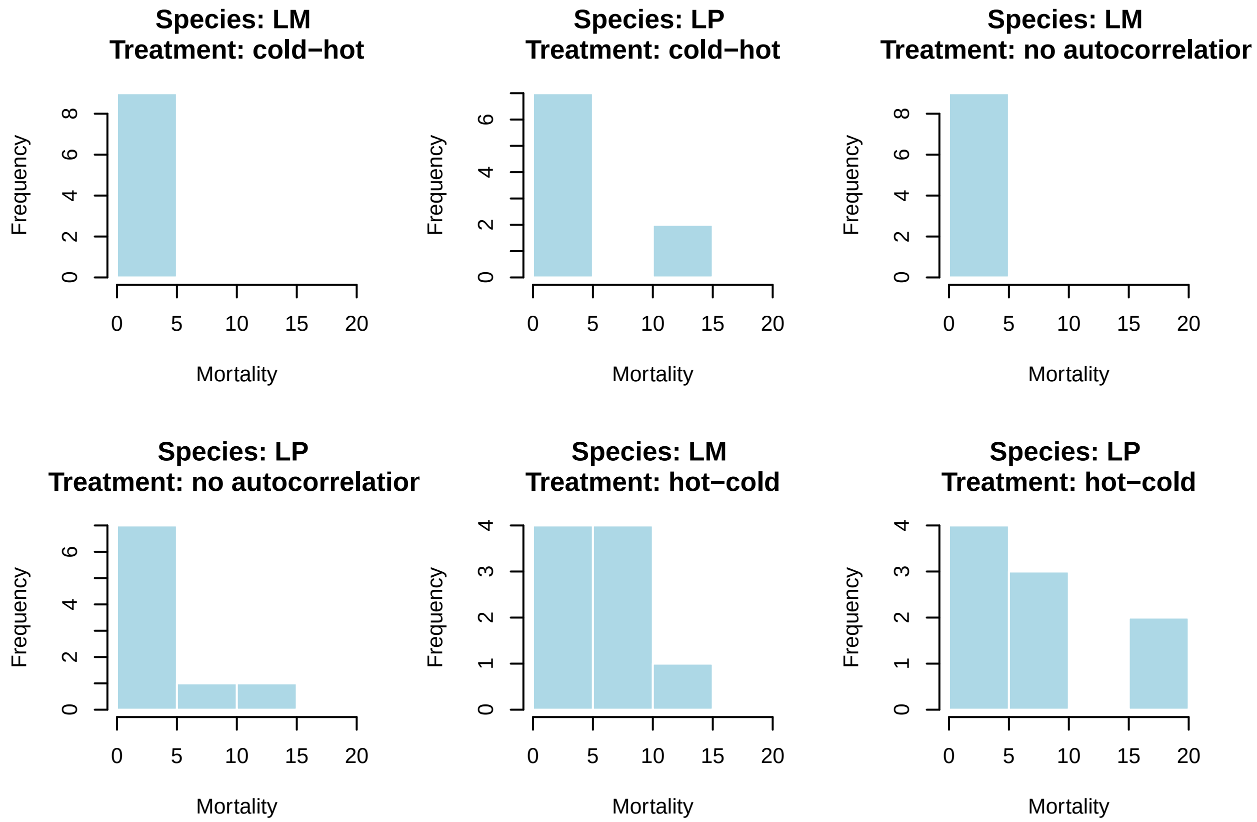
We combine both datasets and filter out rows based on specific conditions for standard deviation (`Obs_sd`) and autocorrelation (`Obs_ac`) to clean the data.

```r
# Blending both datasets, as there are no significant differences between preparation methods (probably
datin <- rbind(original_dataset_1,original_dataset_2)

datin <- datin[!(datin$Treatment == 0 & (datin$Obs_sd <= 2.1 | datin$Obs_sd >= 2.9))
               &!(datin$Treatment == 0.95 & (datin$Obs_sd <= 2.1 | datin$Obs_sd >= 2.9)),]

datin <- datin[!(datin$Treatment == 0 & (datin$Obs_ac <= -0.2 | datin$Obs_ac >= 0.2))
               &!(datin$Treatment == 0.95 & (datin$Obs_ac <= 0.92 | datin$Obs_ac >= 0.98)),]

datin <- subset(datin, !Errors == "y"|is.na(Errors))
datin <- subset(datin, !Treatment == "constant")
```

Dataset: each row of the dataset corresponds to a replicate; here we are showing 3 experiment runs, each including 3 treatments

	Exp_run	Treatment	Species	Frond_count	Mortality	Total_frond
190	9	cold-hot	LP	22	2	24
191	9	cold-hot	LM	35	0	35
192	9	no autocorrelation	LP	18	4	22
193	9	no autocorrelation	LM	21	2	23
194	9	hot-cold	LP	17	7	24
195	9	hot-cold	LM	17	5	22

Overview of data distribution



Summary table: total sum of mortality across treatments

```
##
##
## Treatment              Common duckweed    Dotted duckweed
## -------------------  -----------------  -----------------
## no autocorrelation   14                        31
## cold-hot             1                         33
## hot-cold             54                        60
```

Model Fitting

```
results_list <- list()
compare_results <- list()

for (species in unique(datins$Species)) {

  species_data <- subset(datins, Species == species)

  simple <- glm(Mortality ~ Treatment, data = species_data, family = poisson)
  library(lme4)
exp_number <- glmer(Mortality ~ Treatment + (1|Exp_run), data=species_data, family=poisson, control = g

# Store models
```

```r
results_list[[species]] <- list(simple = simple, exp_number = exp_number)

# Compare models by AIC
Cand.modsF <- list("no random effects" = results_list[[species]]$simple,
                   "experiment number" = results_list[[species]]$exp_number)

# Get AIC for each model and store it
AIC_values <- sapply(Cand.modsF, function(model) AIC(model))

# Function to compute AICc
compute_AICc <- function(model, n) {
  aic <- AIC(model)
  k <- length(coef(model))  # Number of parameters
  aicc <- aic + (2 * k * (k + 1)) / (n - k - 1)  # AICc formula
  return(aicc)
}

n <- nrow(species_data)

# Compute AICc for each model
AICc_values <- sapply(Cand.modsF, compute_AICc, n = n)

# Create a summary table using AICc
compare_results[[species]] <- knitr::kable(data.frame(Model = names(AICc_values), AICc = AICc_values),

cat("Model assessment:", species, "\n")
print(compare_results[[species]])

# Find the best model based on AICc
best_model_name <- names(AICc_values)[which.min(AICc_values)]
results_list[[species]] <- Cand.modsF[[best_model_name]]

  best_model <- results_list[[species]]

  model_dispersion <- sum(residuals(best_model, type = "pearson")^2) / df.residual(best_model)
  print(paste("Model dispersion ratio for species", species, ":", model_dispersion))

  cat("Wald test type 2 for significance of predictor:", species, "\n")
  library(car)
phi <- sum(residuals(best_model, type="pearson")^2) / df.residual(best_model)
anova_table <- Anova(best_model, type = "II")

# Adjusting chi-square values (quasipoisson)
anova_table$`Chisq` <- anova_table$`Chisq` / phi

# Recalculating p-values using chi-square distribution
anova_table$`Pr(>Chisq)` <- pchisq(anova_table$`Chisq`, anova_table$Df, lower.tail = FALSE)

  print(paste("Adjusted predictor significance", species, ":"))
print(anova_table)
}
```

```
## Model assessment: LP
##
```

```
##
##                      Model                  AICc
## ------------------  ------------------  -------
## no random effects   no random effects    221.96
## experiment number   experiment number    168.46
## [1] "Model dispersion ratio for species LP : 2.27711234930369"
## Wald test type 2 for significance of predictor: LP
## [1] "Adjusted predictor significance LP :"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Mortality
##            Chisq Df Pr(>Chisq)
## Treatment 5.5482  2    0.06241 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Model assessment: LM
##
##
##                      Model                  AICc
## ------------------  ------------------  -------
## no random effects   no random effects    108.35
## experiment number   experiment number    101.23
## [1] "Model dispersion ratio for species LM : 1.09452335724362"
## Wald test type 2 for significance of predictor: LM
## [1] "Adjusted predictor significance LM :"
## Analysis of Deviance Table (Type II Wald chisquare tests)
##
## Response: Mortality
##            Chisq Df   Pr(>Chisq)
## Treatment 32.008  2 0.0000001121 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
## [1] "Original model summary - species : LM"
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: poisson  ( log )
## Formula: Mortality ~ Treatment + (1 | Exp_run)
##    Data: species_data
## Control: glmerControl(optimizer = "nloptwrap")
##
##      AIC      BIC   logLik deviance df.resid
##    101.1    106.3    -46.5     93.1       23
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.4999 -0.4380 -0.2976  0.6142  2.3798
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  Exp_run (Intercept) 0.4273   0.6537
## Number of obs: 27, groups:  Exp_run, 9
##
## Fixed effects:
```

```
##                   Estimate Std. Error z value   Pr(>|z|)
## (Intercept)         0.2621     0.3563   0.736    0.46199
## Treatmentcold-hot  -2.6391     1.0172  -2.594    0.00947 **
## Treatmenthot-cold   1.3499     0.2947   4.580 0.00000464 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) Trtmntc-
## Trtmntcld-h -0.190
## Trtmntht-cl -0.657  0.230
## [1] "Adjusted model summary - species : LM"
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)           0.26       0.37     0.7     0.48
## Treatmentcold-hot    -2.64       1.06    -2.5     0.01 *
## Treatmenthot-cold     1.35       0.31     4.4  0.00001 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Post-hoc test for common duckweed

```
## Non-adjusted post-hoc for species: LM
## $emmeans
##  Treatment            rate     SE  df asymp.LCL asymp.UCL
##  no autocorrelation 1.2996 0.4630 Inf    0.6465     2.613
##  cold-hot           0.0928 0.0939 Inf    0.0128     0.674
##  hot-cold           5.0128 1.3805 Inf    2.9219     8.600
##
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
##
## $contrasts
##  contrast                          ratio      SE  df null z.ratio p.value
##  no autocorrelation / (cold-hot) 14.0001 14.2409 Inf    1   2.594  0.0257
##  no autocorrelation / (hot-cold)  0.2593  0.0764 Inf    1  -4.580  <.0001
##  (cold-hot) / (hot-cold)          0.0185  0.0184 Inf    1  -4.022  0.0002
##
## P value adjustment: tukey method for comparing a family of 3 estimates
## Tests are performed on the log scale
##
##
## Adjusted post-hoc for species: LM
## $emmeans
##  Treatment            rate     SE  df asymp.LCL asymp.UCL
##  no autocorrelation 1.2996 0.4844 Inf    0.6259     2.698
##  cold-hot           0.0928 0.0983 Inf    0.0117     0.739
##  hot-cold           5.0128 1.4443 Inf    2.8499     8.817
##
## Confidence level used: 0.95
## Intervals are back-transformed from the log scale
##
## $contrasts
##  contrast                          ratio      SE  df null z.ratio p.value
##  no autocorrelation / (cold-hot) 14.0001 14.8987 Inf    1   2.480  0.0351
##  no autocorrelation / (hot-cold)  0.2593  0.0799 Inf    1  -4.378  <.0001
##  (cold-hot) / (hot-cold)          0.0185  0.0192 Inf    1  -3.845  0.0004
##
## P value adjustment: tukey method for comparing a family of 3 estimates
## Tests are performed on the log scale
```