**Using eDNA to reveal detrimental predator prey interactions caused by invasive species**

**Maria Riaz1,2,3\*, Dan Warren4,5, Claudia Wittwer1,2,3, Berardino Cocchiararo1,2, Inga Hundertmark6, Tobias Erik Reiners1,6, Sven Klimpel2,3,5, Markus Pfenninger2,5,7, Imran Khaliq8,9,10, Carsten Nowak1,2\***

**Appendix 5: R script**

---

title: "Fish and Amphibian eDNA"

author: "Dan Warren"

date: "3/27/2020"

output: html\_document

---

```{r setup, message = FALSE}

library(ggplot2)

library(GGally)

library(DT)

library(tools)

library(dplyr)

library(corrplot)

```

Loading in long format data, which is better for some plots.

```{r long-data, fig.width = 12, fig.height=10}

edna.df.long <- read.csv("~/Dropbox/BiK-F/Maria Riaz/All spp\_17032020\_long.csv")

edna.df.long <- edna.df.long %>%

group\_by(species, site, month) %>%

summarise(dna.conc = mean(dna.conc),

ct.value = mean(ct.value))

datatable(edna.df.long, extensions = 'Buttons',

options = list(dom = 'Blfrtip',

buttons = c('copy', 'csv',

'excel', 'pdf', 'print'),

lengthMenu = list(c(10,25,50,-1),

c(10,25,50,"All")))) %>%

formatRound(columns=colnames(edna.df.long), digits=3)

qplot(month, log(dna.conc), color = species, data = edna.df.long, facets = ~ site, geom = c("point", "line"))

```

Loading in wide-format data, which is probably going to be more useful for analyses.

```{r wide-data, warning=FALSE}

edna.df <- read.csv("~/Dropbox/BiK-F/Maria Riaz/All spp\_17032020\_wide.csv")

edna.df <- edna.df %>%

group\_by(site, month) %>%

summarise(Pelobates.fuscus.dna.conc = mean(Pelobates.fuscus.dna.conc),

Pelobates.fuscus.ct.value = mean(Pelobates.fuscus.ct.value),

Triturus.cristatus.dna.conc = mean(Triturus.cristatus.dna.conc),

Triturus.cristatus.ct.value = mean(Triturus.cristatus.ct.value),

Pseudorasbora.parva.dna.conc = mean(Pseudorasbora.parva.dna.conc),

Pseudorasbora.parva.ct.value = mean(Pseudorasbora.parva.ct.value),

Lepomis.gibbosus.dna.conc = mean(Lepomis.gibbosus.dna.conc),

Lepomis.gibbosus.ct.value = mean(Lepomis.gibbosus.ct.value))

ggpairs(sqrt(edna.df[,grep("dna.conc", colnames(edna.df))]))

```

This is just a quick visualization and shouldn't be over-interpreted. This is square root transformed dna concentrations across all months and sites plotted against each other. Obviously we're not going to analyze the data that way, because it's going to be super problematic due to site effects temporal autocorrelation, etc. What we're going to try for starters is to count transitions in state between pairs of months, and compare those to a null distribution that we get from a Monte Carlo test.

## Corrplots by month

### May

```{r}

edna.df[complete.cases(edna.df),] %>%

ungroup() %>%

filter(month == 5) %>%

select(ends\_with("dna.conc")) %>%

cor(method = "spearman") %>%

corrplot()

```

### June

```{r}

edna.df[complete.cases(edna.df),] %>%

ungroup() %>%

filter(month == 6) %>%

select(ends\_with("dna.conc")) %>%

cor(method = "spearman") %>%

corrplot()

```

### July

```{r}

edna.df[complete.cases(edna.df),] %>%

ungroup() %>%

filter(month == 7) %>%

select(ends\_with("dna.conc")) %>%

cor(method = "spearman") %>%

corrplot()

```

## May-June

Here we're going to build a function that counts state changes. The states we're interested in are:

\* For each species count transitions between subsequent months:

+ 0 -> 0 (stay.zero)

+ !0 -> !0 (persist)

+ 0 -> !0 (invade)

+ !0 -> 0 (extirpation)

```{r}

# edna.df.long <- edna.df.long[complete.cases(edna.df.long),]

species <- c("Lepomis.gibbosus", "Pseudorasbora.parva","Pelobates.fuscus", "Triturus.cristatus")

transitions <- c("persist", "invade",

"stay.zero", "extirpation")

output.colnames <- apply(expand.grid(species, transitions),

1, paste, collapse=".")

may.june.df <- setNames(data.frame(matrix(ncol = 16, nrow = 0)),

output.colnames)

for(i in levels(edna.df$site)){

may.june.df[i,] <- rep(0, 16)

this.site <- edna.df %>%

filter(site == i) %>%

filter(month == 5 | month == 6)

# Making sure we have both May and June

if(all(complete.cases(this.site))){

# And now we've just got a massive switch for all transitions

if(this.site[this.site$month == 5,"Pelobates.fuscus.dna.conc"] == 0 &

this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] == 0){

may.june.df[i, "Pelobates.fuscus.stay.zero"] <- 1

}

if(this.site[this.site$month == 5,"Pelobates.fuscus.dna.conc"] == 0 &

this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] > 0){

may.june.df[i, "Pelobates.fuscus.invade"] <- 1

}

if(this.site[this.site$month == 5,"Pelobates.fuscus.dna.conc"] > 0 &

this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] > 0){

may.june.df[i, "Pelobates.fuscus.persist"] <- 1

}

if(this.site[this.site$month == 5,"Pelobates.fuscus.dna.conc"] > 0 &

this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] == 0){

may.june.df[i, "Pelobates.fuscus.extirpation"] <- 1

}

if(this.site[this.site$month == 5,"Lepomis.gibbosus.dna.conc"] == 0 &

this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] == 0){

may.june.df[i, "Lepomis.gibbosus.stay.zero"] <- 1

}

if(this.site[this.site$month == 5,"Lepomis.gibbosus.dna.conc"] == 0 &

this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] > 0){

may.june.df[i, "Lepomis.gibbosus.invade"] <- 1

}

if(this.site[this.site$month == 5,"Lepomis.gibbosus.dna.conc"] > 0 &

this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] > 0){

may.june.df[i, "Lepomis.gibbosus.persist"] <- 1

}

if(this.site[this.site$month == 5,"Lepomis.gibbosus.dna.conc"] > 0 &

this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] == 0){

may.june.df[i, "Lepomis.gibbosus.extirpation"] <- 1

}

if(this.site[this.site$month == 5,"Pseudorasbora.parva.dna.conc"] == 0 &

this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] == 0){

may.june.df[i, "Pseudorasbora.parva.stay.zero"] <- 1

}

if(this.site[this.site$month == 5,"Pseudorasbora.parva.dna.conc"] == 0 &

this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] > 0){

may.june.df[i, "Pseudorasbora.parva.invade"] <- 1

}

if(this.site[this.site$month == 5,"Pseudorasbora.parva.dna.conc"] > 0 &

this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] > 0){

may.june.df[i, "Pseudorasbora.parva.persist"] <- 1

}

if(this.site[this.site$month == 5,"Pseudorasbora.parva.dna.conc"] > 0 &

this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] == 0){

may.june.df[i, "Pseudorasbora.parva.extirpation"] <- 1

}

if(this.site[this.site$month == 5,"Triturus.cristatus.dna.conc"] == 0 &

this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] == 0){

may.june.df[i, "Triturus.cristatus.stay.zero"] <- 1

}

if(this.site[this.site$month == 5,"Triturus.cristatus.dna.conc"] == 0 &

this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] > 0){

may.june.df[i, "Triturus.cristatus.invade"] <- 1

}

if(this.site[this.site$month == 5,"Triturus.cristatus.dna.conc"] > 0 &

this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] > 0){

may.june.df[i, "Triturus.cristatus.persist"] <- 1

}

if(this.site[this.site$month == 5,"Triturus.cristatus.dna.conc"] > 0 &

this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] == 0){

may.june.df[i, "Triturus.cristatus.extirpation"] <- 1

}

}

}

getfreqs <- function(transition.df){

this.freqmat <- setNames(data.frame(matrix(ncol = 16, nrow = 0)),

colnames(transition.df))

for(i in colnames(transition.df)){

this.colsum <- sum(transition.df[,i])

if(this.colsum == 0){

this.freqmat[i,] <- rep(0, 16)

} else {

for(j in colnames(transition.df)){

# Here we can exploit the fact that our data is 1/0 to get coocurrence of transitions

# by multiplying the two columns

this.freqmat[i,j] <- sum(transition.df[,i] \* transition.df[,j])/this.colsum

}

}

}

return(this.freqmat)

}

this.freqmat <- as.matrix(getfreqs(may.june.df))

corrplot(this.freqmat)

```

It should be noted that this plot is NOT a correlation matrix, which is why it is not symmetric around the diagonal and all values are positive. This matrix represents the number of times where the event on the horizontal was accompanied by the events on the vertical, divided by the number of times the event on the horizontal happened. In other words if half of the times when species A persisted in a population were accompanied by species B going extinct, the value there would be 0.5. If species B \*\*only\*\* went extinct when species A persisted, however, the accompanying proportion above the diagonal would be 1.

Next we're going to implement a Monte Carlo test where each column of the transition data frame is randomized and the data frame is then run through the getfreqs function. We're going to keep track of the replicates and then compare the empirical frequency of coocurrence of transitions to the distribution we obtain from the randomlization test. Randomizing columns independently won't work, because it results in combinations of events in a site that are impossible. Instead we need to group the outcomes by species and then randomize rows for each species - keeping row contents intact for the species but shuffling each species separately.

```{r may.june.mc}

nreps <- 1000

reps <- array(dim = c(nrow(this.freqmat), ncol(this.freqmat), nreps))

for(i in 1:nreps){

this.transition.df <- cbind(may.june.df %>%

select(contains("Lepomis")) %>%

sample\_frac(),

may.june.df %>%

select(contains("Pseudorasbora")) %>%

sample\_frac(),

may.june.df %>%

select(contains("Pelobates")) %>%

sample\_frac(),

may.june.df %>%

select(contains("Triturus")) %>%

sample\_frac())

# Reordering so columns are in the same order.

# Shouldn't be important, but just in case...

this.transition.df <- this.transition.df[,colnames(may.june.df)]

reps[,,i] <- as.matrix(getfreqs(this.transition.df))

}

# Just using the empirical frequency matrix to populate the one where we're going to store

# percentles for significance testing

mc.freqmat <- this.freqmat

mc.freqmat[which(!is.na(mc.freqmat))] <- NA

for(i in 1:nrow(mc.freqmat)){

for(j in 1:ncol(mc.freqmat)){

mc.freqmat[i,j] <- rank(c(this.freqmat[i,j], reps[i,j,]))[1]

}

}

mc.freqmat <- mc.freqmat/(nreps + 1)

p.mat <- mc.freqmat

p.mat[which(p.mat > 0.5)] <- 1 - p.mat[which(p.mat > 0.5)]

corrplot((mc.freqmat \* 2) - 1, p.mat = p.mat, insig = "blank", cl.pos = "n")

```

## June-July

```{r}

june.july.df <- setNames(data.frame(matrix(ncol = 16, nrow = 0)),

output.colnames)

for(i in levels(edna.df$site)){

print(i)

june.july.df[i,] <- rep(0, 16)

this.site <- edna.df %>%

filter(site == i) %>%

filter(month == 6 | month == 7)

# Making sure we have both June and July

if(all(complete.cases(this.site))){

# And now we've just got a massive switch for all transitions

if(this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] == 0 &

this.site[this.site$month == 7,"Pelobates.fuscus.dna.conc"] == 0){

june.july.df[i, "Pelobates.fuscus.stay.zero"] <- 1

}

if(this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] == 0 &

this.site[this.site$month == 7,"Pelobates.fuscus.dna.conc"] > 0){

june.july.df[i, "Pelobates.fuscus.invade"] <- 1

}

if(this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] > 0 &

this.site[this.site$month == 7,"Pelobates.fuscus.dna.conc"] > 0){

june.july.df[i, "Pelobates.fuscus.persist"] <- 1

}

if(this.site[this.site$month == 6,"Pelobates.fuscus.dna.conc"] > 0 &

this.site[this.site$month == 7,"Pelobates.fuscus.dna.conc"] == 0){

june.july.df[i, "Pelobates.fuscus.extirpation"] <- 1

}

if(this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] == 0 &

this.site[this.site$month == 7,"Lepomis.gibbosus.dna.conc"] == 0){

june.july.df[i, "Lepomis.gibbosus.stay.zero"] <- 1

}

if(this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] == 0 &

this.site[this.site$month == 7,"Lepomis.gibbosus.dna.conc"] > 0){

june.july.df[i, "Lepomis.gibbosus.invade"] <- 1

}

if(this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] > 0 &

this.site[this.site$month == 7,"Lepomis.gibbosus.dna.conc"] > 0){

june.july.df[i, "Lepomis.gibbosus.persist"] <- 1

}

if(this.site[this.site$month == 6,"Lepomis.gibbosus.dna.conc"] > 0 &

this.site[this.site$month == 7,"Lepomis.gibbosus.dna.conc"] == 0){

june.july.df[i, "Lepomis.gibbosus.extirpation"] <- 1

}

if(this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] == 0 &

this.site[this.site$month == 7,"Pseudorasbora.parva.dna.conc"] == 0){

june.july.df[i, "Pseudorasbora.parva.stay.zero"] <- 1

}

if(this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] == 0 &

this.site[this.site$month == 7,"Pseudorasbora.parva.dna.conc"] > 0){

june.july.df[i, "Pseudorasbora.parva.invade"] <- 1

}

if(this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] > 0 &

this.site[this.site$month == 7,"Pseudorasbora.parva.dna.conc"] > 0){

june.july.df[i, "Pseudorasbora.parva.persist"] <- 1

}

if(this.site[this.site$month == 6,"Pseudorasbora.parva.dna.conc"] > 0 &

this.site[this.site$month == 7,"Pseudorasbora.parva.dna.conc"] == 0){

june.july.df[i, "Pseudorasbora.parva.extirpation"] <- 1

}

if(this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] == 0 &

this.site[this.site$month == 7,"Triturus.cristatus.dna.conc"] == 0){

june.july.df[i, "Triturus.cristatus.stay.zero"] <- 1

}

if(this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] == 0 &

this.site[this.site$month == 7,"Triturus.cristatus.dna.conc"] > 0){

june.july.df[i, "Triturus.cristatus.invade"] <- 1

}

if(this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] > 0 &

this.site[this.site$month == 7,"Triturus.cristatus.dna.conc"] > 0){

june.july.df[i, "Triturus.cristatus.persist"] <- 1

}

if(this.site[this.site$month == 6,"Triturus.cristatus.dna.conc"] > 0 &

this.site[this.site$month == 7,"Triturus.cristatus.dna.conc"] == 0){

june.july.df[i, "Triturus.cristatus.extirpation"] <- 1

}

}

}

this.freqmat <- as.matrix(getfreqs(june.july.df))

corrplot(this.freqmat)

```

As above, it should be noted that this plot is NOT a correlation matrix.

And now the Monte Carlo tests.

```{r june.july.mc}

nreps <- 1000

reps <- array(dim = c(nrow(this.freqmat), ncol(this.freqmat), nreps))

for(i in 1:nreps){

this.transition.df <- cbind(june.july.df %>%

select(contains("Lepomis")) %>%

sample\_frac(),

june.july.df %>%

select(contains("Pseudorasbora")) %>%

sample\_frac(),

june.july.df %>%

select(contains("Pelobates")) %>%

sample\_frac(),

june.july.df %>%

select(contains("Triturus")) %>%

sample\_frac())

# Reordering so columns are in the same order.

# Shouldn't be important, but just in case...

this.transition.df <- this.transition.df[,colnames(june.july.df)]

reps[,,i] <- as.matrix(getfreqs(this.transition.df))

}

# Just using the empirical frequency matrix to populate the one where we're going to store

# percentles for significance testing

mc.freqmat <- this.freqmat

mc.freqmat[which(!is.na(mc.freqmat))] <- NA

for(i in 1:nrow(mc.freqmat)){

for(j in 1:ncol(mc.freqmat)){

mc.freqmat[i,j] <- rank(c(this.freqmat[i,j], reps[i,j,]))[1]

}

}

mc.freqmat <- mc.freqmat/(nreps + 1)

p.mat <- mc.freqmat

p.mat[which(p.mat > 0.5)] <- 1 - p.mat[which(p.mat > 0.5)]

corrplot((mc.freqmat \* 2) - 1, p.mat = p.mat, insig = "blank", cl.pos = "n")

```

# Parametric approach

The idea here is to do a binomial model with presence/absence of the amphibian species as the response and an autoregressive term to account for whether it was there in the previous month or not, and then use predator presence/absence in the current and previous time steps as predictors. This way we're not focusing on whether the two amphibians are correlated, and we're using something like 60 data points to estimate 4-6 parameters. Could give us a lot more power than the nonparametric tests above. It required significant reformating of the data, though, which is why it's loading from a different csv file.

```{r}

autoreg.df <- read.csv("~/Dropbox/BiK-F/Maria Riaz/All spp\_17032020\_wide\_for\_autoreg.csv")

pelobates.glm <- glm(Pelobates.fuscus ~ Triturus.cristatus + Pseudorasbora.parva + Lepomis.gibbosus + Pelobates.fuscus.prev.month + Triturus.cristatus.prev.month + Pseudorasbora.parva.prev.month + Lepomis.gibbosus.prev.month, data = autoreg.df, family = "binomial")

summary(pelobates.glm)

triturus.glm <- glm(Triturus.cristatus ~ Pelobates.fuscus + Pseudorasbora.parva + Lepomis.gibbosus + Pelobates.fuscus.prev.month + Triturus.cristatus.prev.month + Pseudorasbora.parva.prev.month + Lepomis.gibbosus.prev.month, data = autoreg.df, family = "binomial")

summary(triturus.glm)

pseudorasbora.glm <- glm(Pseudorasbora.parva ~ Pelobates.fuscus + Triturus.cristatus + Lepomis.gibbosus + Pelobates.fuscus.prev.month + Triturus.cristatus.prev.month + Pseudorasbora.parva.prev.month + Lepomis.gibbosus.prev.month, data = autoreg.df, family = "binomial")

summary(pseudorasbora.glm)

lepomis.glm <- glm(Lepomis.gibbosus ~ Pelobates.fuscus + Triturus.cristatus + Pseudorasbora.parva + Pelobates.fuscus.prev.month + Triturus.cristatus.prev.month + Pseudorasbora.parva.prev.month + Lepomis.gibbosus.prev.month, data = autoreg.df, family = "binomial")

summary(lepomis.glm)

```

Unfortunately this isn't giving us much signal of predator effects on prey. Mostly what we see is that the two prey species tend to co-occur, and that the primary determinant of whether a species is in a given locality is whether it was there the previous month. Not super exciting.