

Frog Breeding Responses to Climate Change

Shannon Carter

February 19, 2019

Climate change shifts plants' and animals' seasonal schedules, because events like breeding, migration, and metamorphosis are timed by climate cues. Here, I examine this response in populations of frogs in East Texas. 20 years ago, a collaborator in the USDA installed audio recorders in 8 ponds in East Texas. These recorders are programmed to record 6 times daily, capturing the breeding calls of frogs at the ponds. I've analyzed this data in R to measure if/how frog breeding timing is shifting and how this will affect frogs' interactions with each other. I was the lead author on this project, meaning I conceptualized the questions, performed all data management and analysis, and interpreted and wrote up results for publication. This RMarkdown document provides a summary of the analysis for one of the two major questions for this project. For more information, please feel free to contact me at skarter25@gmail.com or see the full publication in Ecology Letters.

Process and tidy raw data

We have 8 raw data files, each representing 1 pond. Each row represents one time point (6 per day from May 2000 through Dec. 2018) and includes the number of calls made by 12 species of frogs (in columns).

```
## POND YEAR FROGTIME FROGDATE HV HC BV BW RCA RCL RS GC PC PT AC RP
## 1 1 2000 18:00 5/9/2000 2 0 0 0 0 0 0 0 0 0 0 0 0
## 2 1 2000 19:00 5/9/2000 3 0 0 0 0 0 0 0 0 0 0 0
## 3 1 2000 20:00 5/9/2000 1 0 0 0 0 0 0 0 0 0 0 0
## 4 1 2000 21:00 5/9/2000 0 0 0 0 0 0 0 0 0 0 0 0
## 5 1 2000 22:00 5/9/2000 0 0 0 0 0 0 0 0 0 0 0 0
## 6 1 2000 23:00 5/9/2000 0 0 0 0 0 0 0 0 0 0 0 0
```

Here, I merge the data frames for the eight ponds, melt to long format, sum the six daily observations, and adjust time & date format.

```
## Stack raw data into one df
p18 <- rbind(p1, p2, p3, p4, p5, p6, p7, p8)
p18 <- p18[p18$YEAR != 2000,] # 2000 is an incomplete year, so we drop it

## Melt to long format and rename columns
mp18 <- melt(p18, id.vars = c("POND", "YEAR", "FROGTIME", "FROGDATE"),
             measure.vars = c("HV", "HC", "BV", "BW", "RCA", "RCL",
                              "RS", "GC", "PC", "PT", "AC", "RP")) # codes for 12 species
names(mp18) <- c("pond", "year", "time", "date", "sp", "calls")

## Adjust time variable
time <- paste(mp18$date, mp18$time)
time1 <- strptime(time, "%m/%d/%Y %H:%M")
time2 <- as.POSIXlt(time1)
time3 <- as.Date(time2, "%m/%d/%Y %H:%M")
mp18$time <- time3
mp18$doy <- as.numeric(strftime(mp18$time, format = "%j")) # adding a 'day of year'
mp18 <- subset(mp18, select = c("pond", "year", "time", "doy", "sp", "calls"))

## Sum all 6 daily observations and convert to df
daily <- mp18 %>%
  group_by(pond, year, time, doy, sp) %>%
```

```

summarize(dailysum = sum(calls))
daily <- as.data.frame(daily)

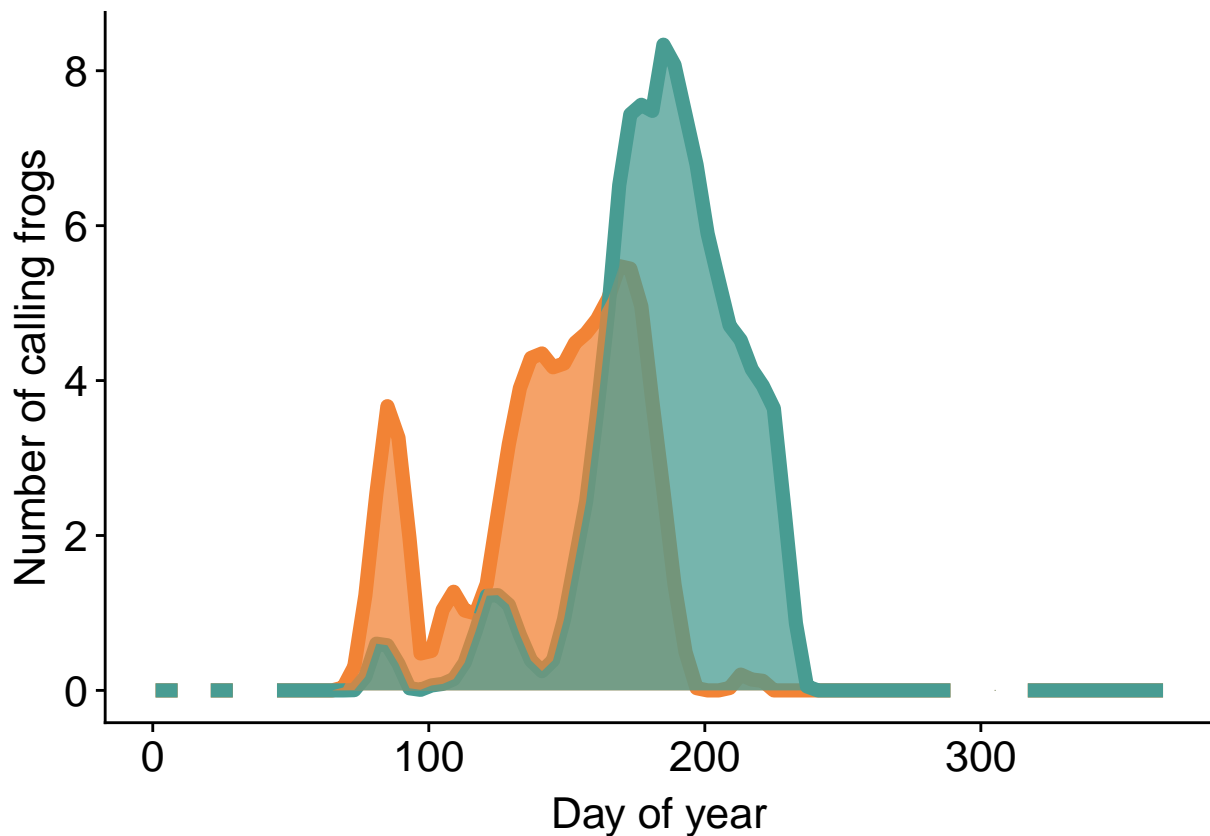
```

The result is a tidy long format data set with one observation per row

##	pond	year	time	doy	sp	dailysum
## 15048	1	2004	2004-06-03	155	RP	0
## 15049	1	2004	2004-06-04	156	HV	7
## 15050	1	2004	2004-06-04	156	HC	4
## 15051	1	2004	2004-06-04	156	BV	0
## 15052	1	2004	2004-06-04	156	BW	0
## 15053	1	2004	2004-06-04	156	RCA	0
## 15054	1	2004	2004-06-04	156	RCL	0
## 15055	1	2004	2004-06-04	156	RS	0

Visualizing the raw data

The above dataset gives time series for the calling behavior of each frog species. Using lowess functions, I can transform the time series into smooth distributions. This makes it easier to visualize when the frogs are active through the year and also enables quantification. Here's an example of those distributions for the gray tree frog (in orange) and the green frog (in teal) in 2007 at pond 4. The major objective of this project was to calculate how the area of overlap (visible in the plot at the area where the orange and teal distributions intersect) changed across years for pairwise combinations of species. This would tell us if climate change-driven shifts in breeding timing influence the competition between species.



Function to calculate overlap for every species pair

Now, I want to calculate the area of intersection between species calling distributions (depicted above), but for all the species and all the years and all the ponds. To do this, I wrote a custom function that 1) subsets the data by species, year, and pond 2) makes lowess smoothed distributions of calling activity for each species 3) calculates the integrated area of intersection between the two species' lowess distributions. I then apply this function to all 132 pairings of species at each of the 8 sites and for each of the 17 years.

User-written functions like this are a really powerful element of R! In analyzing this dataset, I built a number of custom functions as pipelines to perform calculations, plotting, and analysis on iterative slices of data. Without functions, managing analysis was totally unwieldy. Any upstream change to the data or my analysis would render all subsequent parts broken and I'd have to tediously retweak everything. Discovering and learning to build functions was a lifesaver. What initially took place in several separate steps—subsetting data, processing data, calculating key metrics, statistical models, and plotting—was now integrated into the same reproducible pipeline that could be easily be adjusted. By mastering functions, I was able to do rigorous model selection quickly and efficiently because changing model formulation was as simple as changing 1 line in the function and re-applying the function to the data.

```
## The function take 4 inputs: year, pond, species 1 and species 2
## I applied the function to all combinations of these parameters
overlapmatrix <- function(y, p, s1, s2) {

  ## Subset each species, year and pond from daily calls data
  of1 <- subset(daily, subset = (sp == s1 & year == y & pond == p))
  of2 <- subset(daily, subset = (sp == s2 & year == y & pond == p))

  ## Run lowess functions-- this smoothes the time series data into a distribution
  ## f, iter, and delta are parameters that control the degree and method of smoothing
  lf1 <- lowess(of1$dailysum, f = 1/50, iter = 3, delta = 4)
  lf2 <- lowess(of2$dailysum, f = 1/50, iter = 3, delta = 4)

  ## Calculate overlap
  # make a dataframe with time as x and each species' lowess curve as a separate y
  d <- data.frame(day = lf1$x,
                 frog1 = lf1$y,
                 frog2 = lf2$y)
  # designate the lower lowess because this will be the ceiling of the integrated area
  d$min <- pmin(d$frog1, d$frog2)
  # integrated area of curves is time x the lower lowess curve
  inter <- integrate.xy(d$day, d$min)
  # standardize overlap by making it a proportion of each species' full distribution
  prop1 <- inter/integrate.xy(d$day, d$frog1)
  prop2 <- inter/integrate.xy(d$day, d$frog2)

  ## Designate and return output
  out <- cbind(prop1, prop2)
  return(out)
}
```

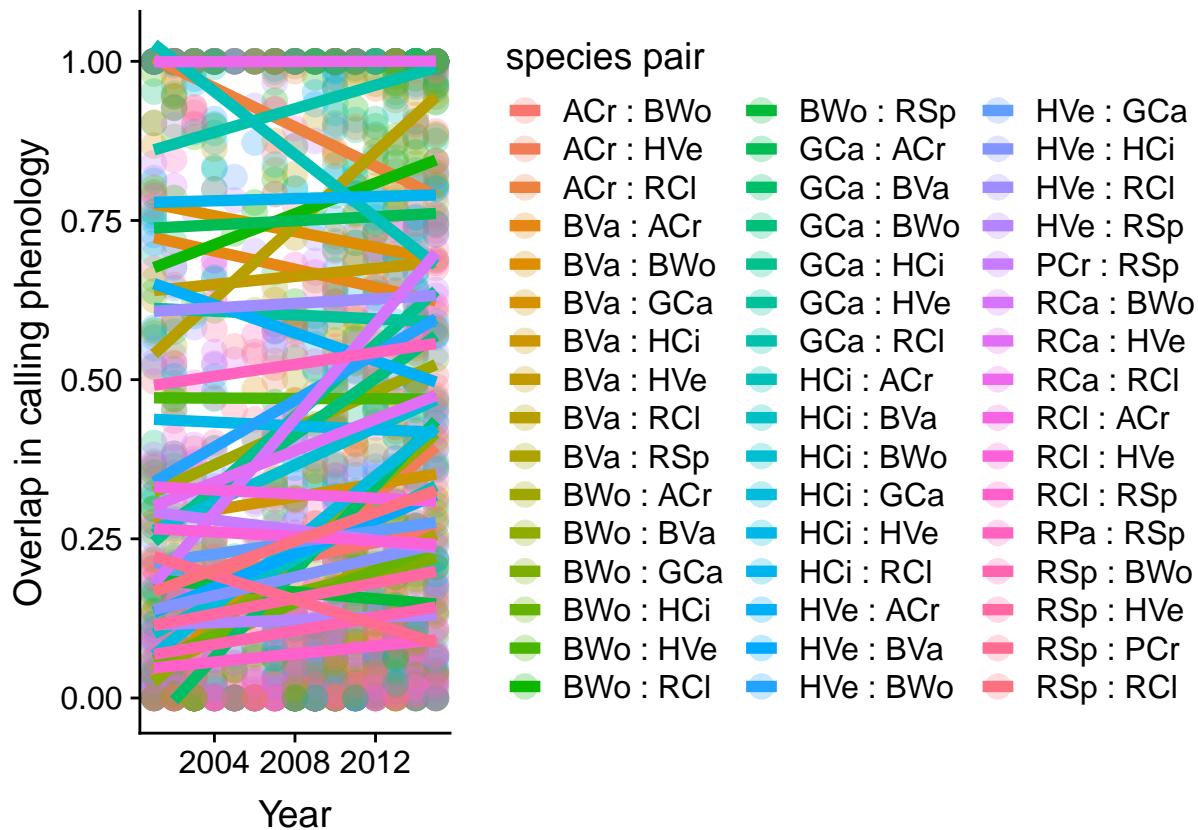
After running the function through all species and years, the result is a dataframe with the proportional overlap for each species pair in each year and pond in which both species were present. There are two overlap values because this was standardized relative to the absolute size of each species' distribution.

```
##   X year pond sp1 sp2   overlap overlapf2      spsp
## 1 1 2002    1  HC  HV 0.2876729 0.14013007 HCi : HVe
## 2 2 2004    1  HC  HV 0.8335911 0.19707868 HCi : HVe
```

```
## 3 3 2005    1  HC  HV 0.0000000 0.00000000 HCi : HVe
## 4 4 2006    1  HC  HV 0.1953258 0.04375981 HCi : HVe
## 5 5 2008    1  HC  HV 0.2490100 0.14031609 HCi : HVe
## 6 6 2014    1  HC  HV 1.0000000 0.02264580 HCi : HVe
```

Plotting all results

Now, we want to see how overlap between species temporal distributions changes across years for each species pair. This plot shows the regressions for each year, but it's super hard to read and ugly. In this figure, each color represents a species pair. The slope of the line indicates how the overlap in calling between that species pair changed across the 15 year period of this data. Positive slopes indicate that overlap increased, negative slopes indicate that overlap decreased.



The info we really want from this figure is the slope of each line and the amount of error around that slope (from replicate ponds). The following figure gives exactly that and is much easier to interpret and better looking. ggplot is great because it allows you to build a plot from multiple data sources and offers endless customization ability. Here, each row represents a species pair. Black points represent the average regression slope with error from pond replicates. Colored points represent the regression slope for each pond. Points that are positive indicate overlap increased across years. Points that are negative indicate that overlap decreased across years.

```
## Overlap trend over years for each species pair
## Reorder species based on the magnitude of the coef so it's easier to read
oy <- ggplot(over_year, aes(x = coef, y = reorder(spsp, coef))) + mytheme +

## Ponds shown independently
```

```

geom_point(size = 3, alpha = 0.75,
           aes(color = as.factor(pond))) +

## Average of all 8 ponds
geom_point(size = 4, colour = "black", shape = 18,
           data = over_year_err, aes(x = coef, y = spsp)) +

## SE bars on averages
geom_errorbarh(size = 1, height = 0,
              data = over_year_err, aes(xmin = coef - se, xmax = coef + se)) +

## Lines to visually separate species pairs
geom_hline(yintercept = seq(1.5, length(over_year$spsp), 1),
          color = "lightgray", linetype = 2) +

## Add a marker at 0 to show show the position where overlap stays constant across years
## Positive values indicate increasing overlap; negative indicates decreasing overlap
geom_vline(xintercept = 0, size = 1, linetype = 2) +

## Style elements
theme(panel.border = element_rect(colour = "black", fill = NA, size = 2),
      axis.text.y = element_text("mono", size = 12),
      axis.text.x = element_text(size = 15),
      axis.title.x = element_text(size = 13),
      axis.title.y = element_text(size = 15)) +

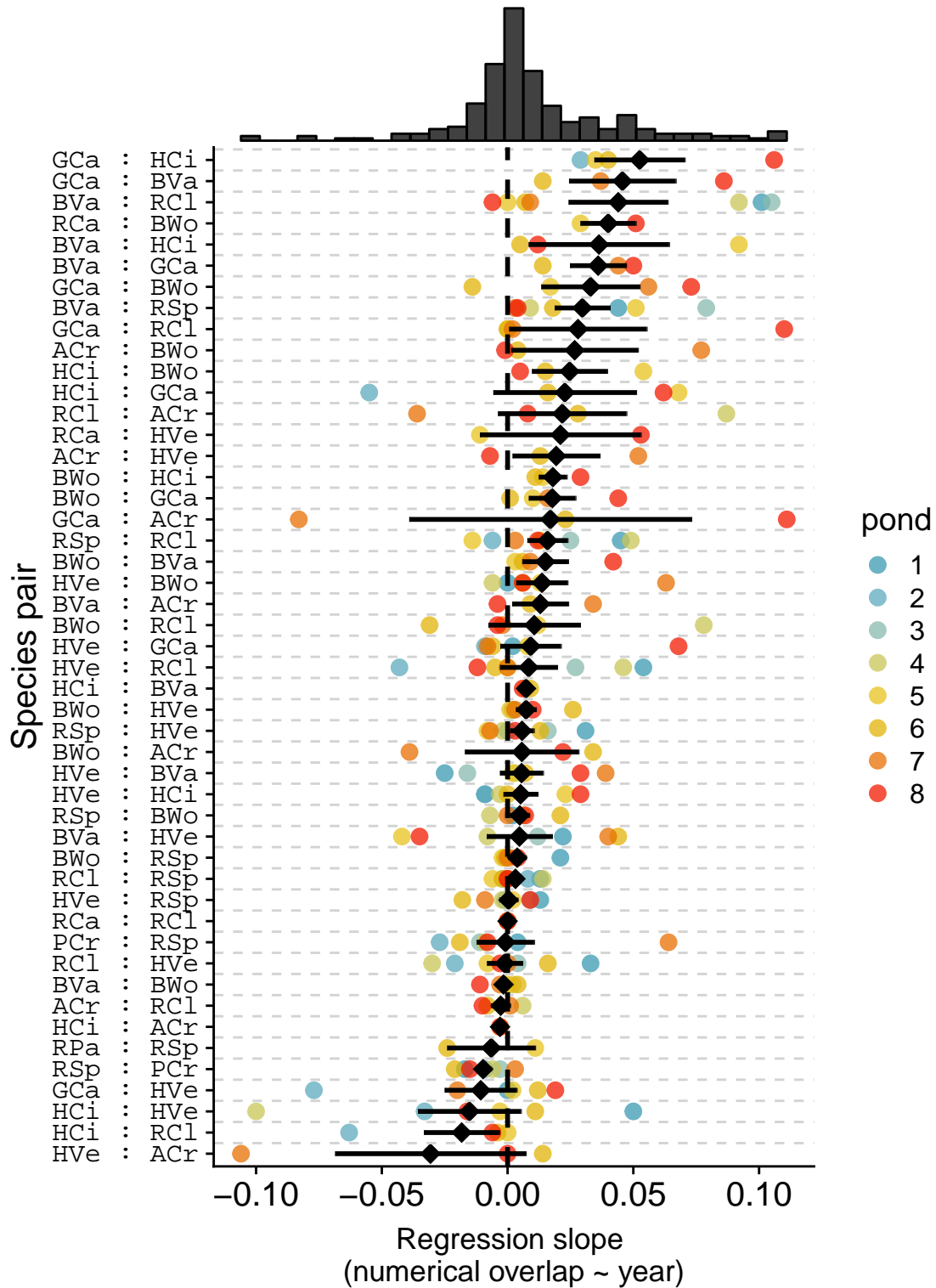
## Labels for axes and legend
labs(x = "Regression slope \n(numerical overlap ~ year)",
     y = "Species pair",
     color = 'pond') +

## Add a Wes Anderson color palette for extra fun
scale_color_manual(values = wes_palette(n = 8, name = "Zissou1", type = 'continuous'))

## Add a marginal histogram so it's easier to see the overall pattern
oy <- ggMarginal(oy, size = 7, margins = "x", type = "histogram",
                col = "black", fill = "black", alpha = 0.75)

```

A lot of data and a lot of coding went into this figure, and it was therefore a challenge to keep the design clean and elegant. I put a lot of thought into what the key messages were and simplified wherever possible. I'm very proud of the end result! The final paper had four figures that looked just like this, but measuring different trends and models. The consistent figure design helped readers process the information more quickly and make comparisons across different components of the paper. This paper got accepted to the top journal in my field, a distinction I largely attribute to the effort and thought that went into these figures.



Here, we can see that most points are positive and the marginal histogram is right skewed. This means that it was common for frogs to increase their overlap in breeding through the 17 year period we measured. We suspect this is a result of climate change. Typically, frogs try to partition their breeding in time. But most species can only breed (and therefore they only call) when it's raining. In this region, climate change has resulted in long periods of dry with rare but large rain events. This forces frogs to cluster their calling around

these few rain opportunities. The outcome is greater competition for the resulting offspring, which should reduce survival over times.