Variables

year (*t*)
event day of year (*y*)
species (which we assume to be unique; and encoded as an integer) (*j*)
number of species (J)
number of data points (N), indexed by (*i*)

Model

$$y_i = a_{j[i]} + b_{j[i]} t_i + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma_y)$$

$$a_j \sim N(\mu_a, \sigma_a)$$
$$b_j \sim N(\mu_b, \sigma_b)$$

This model fits a line for each species (First equation). Each species has its own intercept *a* and slope *b*. Plus there will be error associated with the fit, as is typical in regressions. That's the ϵ, which is normally distributed (second equation). If we stopped with just the first two equations (changing $a_{j[i]}$ to a_i and $b_{j[i]}$ to b_i), we would have a completely unpooled model. In other words, we would be fitting independent lines to each species time-series. Note that the first two equations can be combined into a single equivalent one:

$$y_i \sim N(a_{j[i]} + b_{j[i]} t_i \, , \sigma_y)$$

Instead of fitting each line independently, though, we say that we expect that species are similar in some way – they come from a sample of all possible species, which we assume has normal distributions (third and fourth equations). In particular, we assume that the intercepts among all species come from a normal distribution with some mean μ_a and standard deviation σ_a. And we assume that slopes among all species come from a normal distribution with some mean μ_b and standard deviation σ_b.

Note that just as we're able to combine equations 1 and 2, we can likewise break apart equations 3 and 4:

$$a_j = \mu_a + \zeta_j$$
$$\zeta_j \sim N(0, \sigma_a)$$

$$b_j = \mu_b + \gamma_j$$
$$\gamma_j \sim N(0, \sigma_b)$$

This is useful, because now we can combine all the parts we really care about into the main equation, and just model all the variance as normals with mean zero:

$$y_i = (\mu_a + \zeta_{j[i]}) + (\mu_b + \gamma_{j[i]})t_i + \epsilon_i$$
$$\epsilon_i \sim N(0, \sigma_y)$$
$$\zeta_j \sim N(0, \sigma_a)$$
$$\gamma_j \sim N(0, \sigma_b)$$

Or, combining the first two lines, like we did before:

$$y_i \sim N\left((\mu_a + \zeta_{j[i]}) + (\mu_b + \gamma_{j[i]})t_i, \sigma_y\right)$$
$$\zeta_j \sim N(0, \sigma_a)$$
$$\gamma_j \sim N(0, \sigma_b)$$

These algebraic steps were the sort that were done for the model that Lizzie did with Andrew with the 'types' included. You can basically understand the $\zeta_{j[i]}$ and $\gamma_{j[i]}$ parts (sorry for all the weird Greek letters) as the variation around the mean intercept for each species and the mean slope for each species.

We probably don't care much about intercepts in general. Instead we care about the slopes. Perhaps we care a bit about what the mean semi-pooled slope is across all species: μ_b. We'll definitely be interested in all the individual slopes: $\mu_b + \gamma_j$ (the mean of all species' slopes plus the difference of each individual species' slope from the mean). (This is simply all the b_j values in the original version of the model.)

Note that, properly, we should be also modeling the covariance between the original a_j and b_j values – or equivalently, all the ζ_j and γ_j values – rather than assuming the intercepts and slopes are independent of one another (which is what happens when we don't model the covariance).

So the model becomes:

$$y_i \sim N(a_{j[i]} + b_{j[i]}t_i, \sigma_y)$$

$$\begin{pmatrix} a_j \\ b_j \end{pmatrix} \sim N\left(\begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}, \begin{pmatrix} \sigma_a^2 & \rho\sigma_a\sigma_b \\ \rho\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix}\right)$$

Or, equivalently:

$$y_i \sim N\left((\mu_a + \zeta_{j[i]}) + (\mu_b + \gamma_{j[i]})t_i, \sigma_y\right)$$
$$\begin{pmatrix} \zeta_j \\ \gamma_j \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_a^2 & \rho\sigma_a\sigma_b \\ \rho\sigma_a\sigma_b & \sigma_b^2 \end{pmatrix}\right)$$

I've coded up the first version of the model in STAN. It's called `synchrony1_notype_MK.stan` and you can run it on fake data generated by Lizzie's R code using the R script `fakedata.R` You'll see that you get a few warning messages. I believe that's due to the lack of the covariance matrix, which overly constrains the parameters and makes it hard for metropolis algorithm to do its thing. The model runs in a reasonable amount of time. But I think that additional steps to convert it to the last version of the model (using zero-centered distributions) will speed up computation time. I was able to get reasonable parameter estimates with rhats all near 1. So I think it's a good first step. Next step would be to add in the covariance matrix.

Then trying to figure out how to use the results and/or change the model to answer the question about whether pairs of species are becoming more out-of-sync over time. Dan and I talked about this a bit and we realized that people can mean different things when they talk about synchrony. There's the type that I think this model is trying to capture – an overall trend in, say, flowering time vs. arrival/hatching of the flower's key pollinator. But Dan also pointed out that year-to-year changes can matter, too. That is, you could have the same slope for a species pair – and that slope could even be zero – but the species might track each other for a while and then go out-of-phase (or vice versa). This model won't capture that sort of going out-of-sync dynamic.