Phylogenetic mixed model

Current model description

For each of n species, we assumed data were generated from the following sampling distribution:

$$y_j \sim \mathcal{N}(\mu_j, \sigma_e^2) \tag{1}$$

where:

$$\mu_j = \alpha_j + \beta_j X \tag{2}$$

Predictor X is continuous (in this case, year), and its effects on the phenology of species j are determined by parameter β_j representing response over time. We assumed each trait, including the specific-level intercept α_j, evolved independently following a Brownian motion model of evolution. Thus, species traits are elements of the following normal random vectors:

$$\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_n\}^T \text{ such that } \boldsymbol{\alpha} \sim \mathcal{N}(\mu_\alpha, \boldsymbol{\Sigma_\alpha}) \tag{3}$$

$$\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_n\}^T \text{ such that } \boldsymbol{\beta} \sim \mathcal{N}(\mu_\beta, \boldsymbol{\Sigma_\beta})$$

$$\tag{4}$$

where the means of the multivariate normal distributions are root trait values (i.e., trait values prior to evolving across a phylogenetic tree) and $\boldsymbol{\Sigma_i}$ are $n \times n$ phylogenetic variance-covariance matrices of the form:

$$\begin{bmatrix} \sigma_i^2 & \lambda_i \times \sigma_i^2 \times \rho_{12} & \ldots & \lambda_i \times \sigma_i^2 \times \rho_{1n} \\ \lambda_i \times \sigma_i^2 \times \rho_{21} & \sigma_i^2 & \ldots & \lambda_i \times \sigma_i^2 \times \rho_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ \lambda_i \times \sigma_i^2 \times \rho_{n1} & \lambda_i \times \sigma_i^2 \times \rho_{n2} & \ldots & \sigma_i^2 \end{bmatrix} \tag{5}$$

where σ_i^2 is the variance of the Brownian motion model of evolution and thus represents the rate of evolution across a tree for trait i (assumed to be constant along all branches). Parameter λ_i is Pagel's D, which scales the covariance and therefore is a measure of the "phylogenetic signal" within trait i (note: because estimates can be partitioned into phylogenetic and nonphylogenetic components in this model, $0 \leq \lambda \leq 1$; see Freckleton et al. 2002 Am Nat). Finally, ρ_{xy} is the phylogenetic correlation between species x and y, or the fraction of the tree shared by the two species.

The above specification is exactly equivalent to writing equation 2 in terms of root trait values and residuals, such that:

$$\mu_j = \mu_\alpha + \mu_{\beta_1} X_1 + \mu_{\beta_2} X_2 + \mu_{\beta_3} X_3 + e_{\alpha_j} + e_{\beta_{1,j}} + e_{\beta_{2,j}} + e_{\beta_{3,j}} \qquad (6)$$

where the residual error terms (e.g., e_{α_j}) are elements of normal random vectors from multivariate normal distributions centered on 0 with the same phylogenetic variance-covariance matrices as in equation 5.

Lizzie's basic model and code description

Let's start with the basic **PMM (phylogenetic mixed model)**

As I understand it:

$$y = \alpha + \beta x + a + e \tag{7}$$
$$a \sim normal(0, \sigma_P^2 \Sigma) \tag{8}$$
$$e \sim normal(0, \sigma_R^2 I) \tag{9}$$

... where α and β, respectively, are the intercept and the slope for the cofactor x, a is the phylogenetic random effect, and e is the residual error. Σ is a phylogenetic correlation matrix, I stands for the relevant identity matrix. (The above from Chapter 11: General Quantitative Genetic Methods for Comparative Biology, by Villemereuil & Nakagawa, though seems similar to Housworth.)

What we would like to do is have the phylogenetic effect structure the species-level slopes (while also allowing partial pooling on the species given uneven data, that is, for some species we might have 3 observations, each at a different x value, and for other species, we have 30 observations).

We think we have this running in `ubermini_2.R` (simulation code) and `ubermini_2.stan`.

In the Stan code, β is:

```
b_force~ multi_normal(rep_vector(b_z,n_sp),
lambda_vcv(Vphy, lam_interceptsb, sigma_interceptsb)
```

So the `b_z` represents the root trait value, and all the action happens in the second part of the function, which calls `lambda_vcv`, which is defined at the top of the Stan code. This function sets up two matrices (both the size of the VCV in your data): (1) a matrix of your VCV with λ on the off-diagonals only (`local_vcv`) and (2) a matrix with σ on the diagonal (`sigma_mat`) and then returns the product `sigma_mat * lambda_mat * sigma_mat`. So that, when you put the β code above with the `lambda_vcv` function you get:

$$\beta \sim MVN(\mu, \Phi\Sigma)$$
$$\Phi\Sigma = \begin{bmatrix} \sigma_\beta^2 & \lambda\sigma_\beta^2 & \lambda\sigma_\beta^2 \\ \lambda\sigma_\beta^2 & \sigma_\beta^2 & \lambda\sigma_\beta^2 \\ \lambda\sigma_\beta^2 & \lambda\sigma_\beta^2 & \sigma_\beta^2 \end{bmatrix}$$