Phylogenetic Generalized Multilevel Models (PGMMs)

General Principles

Building upon the hierarchical frameworks introduced in Varying intercepts and Varying slopes, we extend varying effects to incorporate *phylogenetic structuring*. Unlike standard multilevel models that assume exchangeability among group levels, comparative biological data inherently violate the assumption of independent and identically distributed observations due to shared evolutionary descent. This tendency for closely related taxa to exhibit phenotypic similarities is quantified as phylogenetic signal. To rigorously control for this statistical non-independence, we employ **Phylogenetic Generalized Multilevel Models (PGMMs)** (brms Development Team 2025).

In these architectures, species identity is specified as a structured varying effect—conceptually analogous to subject-level effects in longitudinal designs. However, rather than assuming independent intercepts, we parameterize the variance-covariance structure of this effect using a *phylogenetic variance-covariance* (VCV) matrix derived from the tree topology. This parameterization enables a precise partitioning of variance among the phylogenetic signal, fixed environmental or biological covariates, and residual (non-phylogenetic) variation.

In this chapter, we delineate the implementation of several foundational phylogenetic models within BF:

1. **Simple Phylogenetic Model**: A Gaussian response model for continuous traits, integrating a phylogenetic VCV matrix to model the correlated error structure among taxa.
2. **Phylogenetic Poisson Model**: Formulated for count data (e.g., substitution events or species richness), this model accommodates discrete distributions while simultaneously estimating phylogenetic covariance and addressing overdispersion.
3. **Models with Repeated Measurements**: Extensions accommodating multiple intra-specific observations, allowing for the explicit decoupling of intra-specific trait variation from inter-specific phylogenetic effects.

4. **Phylogenetic Meta-Analysis**: A framework for aggregating effect sizes across disparate studies, weighting estimates by their precision while controlling for the phylogenetic non-independence of the focal taxa.
5. **Phylogenetic Varying Slopes**: Advanced parameterizations that allow the coefficients of predictors to covary with the phylogeny, thereby modeling evolutionary shifts in trait allometries or ecological correlations.

Example 1: Simple Phylogenetic Model

The simple phylogenetic model is deployed to evaluate the effect of one or more predictors, x_i (`x`), on a continuous phenotypic trait, y_i (`y`), while explicitly accounting for the phylogenetic relatedness of the taxa (brms Development Team 2025). Let N denote the total number of observations. The phylogenetic structure is introduced via a varying intercept, u_{phylo}, where the prior covariance among species is proportional to the phylogenetic relatedness matrix, $\mathbf{A}$.

To optimize computational efficiency during sampling, we utilize the Cholesky decomposition of this matrix, such that $\mathbf{A} = \mathbf{L}\mathbf{L}^T$. Here, $\mathbf{L}$ (`L`) is a lower triangular matrix of dimensions $M \times M$ (where M represents the number of unique species). This Cholesky factor is mapped to the N observations via an indexing vector, `phylo_idx`, facilitating the efficient mathematical transformation of uncorrelated standard normal variables into the phylogenetically correlated intercepts u_{phylo}.

Implementation

Python

```python
from BayesForge import bf
import jax.numpy as jnp

m = bf(platform='cpu')

# Load and prepare data
df = m.load.phylo_simple()
L_df = m.load.phylo_L_simple()
L = L_df.values
species_to_idx = {sp: i for i, sp in enumerate(L_df.columns)}
df["phylo_idx"] = df["phylo"].map(species_to_idx)

m.data_on_model = {
    "y": jnp.array(df["y"].values),
    "x": jnp.array(df["x"].values),
```

```
    "phylo_idx": jnp.array(df["phylo_idx"].values, dtype=jnp.int32),
    "L": jnp.array(L)
}

def model(y, x, phylo_idx, L):
    # Priors
    intercept = m.dist.normal(0, 50, name="intercept")
    b_x = m.dist.normal(0, 10, name="b_x")

    # Standard deviation for phylogenetic effect
    sd_phylo = m.dist.half_normal(20, name="sd_phylo")
    sigma = m.dist.half_normal(20, name="sigma")

    # Phylogenetic random effect (non-centered parameterization)
    num_species = L.shape[0]
    z_phylo = m.dist.normal(jnp.zeros(num_species), 1.0, name="z_phylo")
    u_phylo = jnp.matmul(L, z_phylo) * sd_phylo

    # Linear predictor
    mu = intercept + b_x * x + u_phylo[phylo_idx]

    # Likelihood
    m.dist.normal(mu, sigma, name="obs", obs=y)

m.fit(model)
```

```
bf v 0.0.48 package loaded

E0527 08:57:44.830290 1346013 cuda_dnn.cc:523] Loaded runtime CuDNN library: 9.1.0 but source
E0527 08:57:44.832796 1346013 cuda_dnn.cc:523] Loaded runtime CuDNN library: 9.1.0 but source

jax.local_device_count 32

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]
```

R

```r
library(BayesForge)
m <- importBF(platform="cpu")

model <- function(y, x, phylo_idx, L) {
    # Priors
    intercept <- bf.dist.normal(0, 50, name = "intercept")
    b_x <- bf.dist.normal(0, 10, name = "b_x")

    sd_phylo <- bf.dist.half_normal(20, name = "sd_phylo")
    sigma <- bf.dist.half_normal(20, name = "sigma")

    # Phylogenetic effects
    num_species <- dim(L)[1]
    z_phylo <- bf.dist.normal(rep(0, num_species), 1.0, name = "z_phylo")
    u_phylo <- (L %*% z_phylo) * sd_phylo

    # Linear predictor
    mu <- intercept + b_x * x + u_phylo[phylo_idx]

    # Likelihood
    bf.dist.normal(mu, sigma, name = "obs", obs = y)
}

m$fit(model)
```

Julia

```julia
using BayesForge
m = importBF(platform="cpu")

@BF function model(y, x, phylo_idx, L):
    # Priors
    intercept = m.dist.normal(0, 50, name = "intercept")
    b_x = m.dist.normal(0, 10, name = "b_x")

    sd_phylo = m.dist.half_normal(20, name = "sd_phylo")
    sigma = m.dist.half_normal(20, name = "sigma")

    # Phylogenetic effects
```

```
    num_species = size(L, 1)
    z_phylo = m.dist.normal(zeros(num_species), 1.0, name = "z_phylo")
    u_phylo = (L * z_phylo) .* sd_phylo

    # Linear predictor
    mu = intercept .+ b_x .* x .+ u_phylo[phylo_idx]

    # Likelihood
    m.dist.normal(mu, sigma, name = "obs", obs = y)
end

m.fit(model)
```

Mathematical Details

The full hierarchical model is formally specified as follows:

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i + u_{\text{phylo}[i]}$$

$$\mathbf{u}_{\text{phylo}} \sim \text{MultivariateNormal}(\mathbf{0}, \sigma^2_{\text{phylo}}\mathbf{A})$$

$$\alpha \sim \text{Normal}(0, 50)$$

$$\beta \sim \text{Normal}(0, 10)$$

$$\sigma, \sigma_{\text{phylo}} \sim \text{HalfNormal}(0, 20)$$

Where:

- y_i denotes the observed continuous phenotypic trait for the i-th observation.
- x_i represents the value of the fixed environmental or biological covariate for the i-th observation.

- $\mathbf{u}_{\text{phylo}}$ is the vector of phylogenetically structured varying intercepts. It is modeled as a multivariate normal distribution with covariance proportional to the phylogenetic variance-covariance (VCV) matrix $\mathbf{A}$. The index mapping, $u_{\text{phylo}[i]}$, assigns the species-specific effect to the corresponding observation.
- σ_{phylo} is the phylogenetic standard deviation, which scales the magnitude of the variance attributable to shared evolutionary history.
- σ represents the residual (non-phylogenetic) standard deviation, capturing unmeasured environmental variation or intra-specific error.

Example 2: Phylogenetic Poisson Model

To model discrete count data, y_i (`y`), we employ a Poisson likelihood parameterized via a log link function. Because standard Poisson distributions strictly assume that the variance equals the mean, empirical biological data—such as mutation counts or species richness—frequently exhibit overdispersion. To mathematically accommodate this extra-Poisson variation, we introduce an **Observation-Level Random Effect (OLRE)**, denoted as $\epsilon_{\text{obs}[i]}$ (`u_o`). This unstructured varying effect operates concurrently with the phylogenetically structured varying intercept, $u_{\text{phylo}[i]}$ (`u_p`), allowing the model to decouple residual individual-level variance from the shared evolutionary signal.

The computational data structure necessitates a response vector of counts (`y`) and a fixed covariate vector x_i (`x`), both of length N. The shared evolutionary history is again incorporated via the Cholesky factor of the phylogenetic variance-covariance matrix, $\mathbf{L}$ (`L`), of dimensions $M \times M$, where M denotes the total number of unique taxa represented in the phylogeny. Finally, two indexing vectors are required to map these latent hierarchical variables to the observations: `phylo_idx` (length N) maps the M species-specific phylogenetic effects to the corresponding data points, while `obs_idx` (length N, typically an integer sequence defining each row) assigns a unique OLRE to each individual observation.

Implementation

Python

```python
# Load and prepare data
df = m.load.phylo_poisson()
L_df = m.load.phylo_L_poisson()
species_to_idx = {sp: i for i, sp in enumerate(L_df.columns)}

m.data_on_model = {
    "y": jnp.array(df["y"].values),
```

```
    "x": jnp.array(df["x"].values),
    "phylo_idx": jnp.array(df["phylo"].map(species_to_idx).values, dtype=jnp.int32),
    "obs_idx": jnp.arange(len(df), dtype=jnp.int32),
    "L": jnp.array(L_df.values)
}

def model(y, x, phylo_idx, obs_idx, L):
    # Priors
    intercept = m.dist.normal(0, 5, name="intercept")
    b_x = m.dist.normal(0, 2, name="b_x")

    sd_phylo = m.dist.half_normal(1, name="sd_phylo")
    sd_obs = m.dist.half_normal(1, name="sd_obs")

    # Effects (Non-centered)
    z_p = m.dist.normal(jnp.zeros(L.shape[0]), 1, name="z_p")
    u_p = jnp.matmul(L, z_p) * sd_phylo

    z_o = m.dist.normal(jnp.zeros(len(y)), 1, name="z_o")
    u_o = z_o * sd_obs

    # Predictor (Log link)
    mu = intercept + b_x * x + u_p[phylo_idx] + u_o[obs_idx]

    m.dist.poisson(jnp.exp(mu), name="obs", obs=y)

m.fit(model)
```

```
  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]
```

R

```
model <- function(y, x, phylo_idx, obs_idx, L) {
    intercept <- bf.dist.normal(0, 5, name = "intercept")
    b_x <- bf.dist.normal(0, 2, name = "b_x")

    sd_phylo <- bf.dist.half_normal(1, name = "sd_phylo")
    sd_obs <- bf.dist.half_normal(1, name = "sd_obs")

    num_species <- dim(L)[1]
    z_p <- bf.dist.normal(rep(0, num_species), 1, name = "z_p")
    u_p <- (L %*% z_p) * sd_phylo

    z_o <- bf.dist.normal(rep(0, length(y)), 1, name = "z_o")
    u_o <- z_o * sd_obs

    mu <- intercept + b_x * x + u_p[phylo_idx] + u_o[obs_idx]
    bf.dist.poisson(exp(mu), name = "obs", obs = y)
}
```

Julia

```
@BF function model(y, x, phylo_idx, obs_idx, L):
    intercept = m.dist.normal(0, 5, name = "intercept")
    b_x = m.dist.normal(0, 2, name = "b_x")

    sd_phylo = m.dist.half_normal(1, name = "sd_phylo")
    sd_obs = m.dist.half_normal(1, name = "sd_obs")

    num_species = size(L, 1)
    z_p = m.dist.normal(zeros(num_species), 1, name = "z_p")
    u_p = (L * z_p) .* sd_phylo

    z_o = m.dist.normal(zeros(length(y)), 1, name = "z_o")
    u_o = z_o .* sd_obs

    mu = intercept .+ b_x .* x .+ u_p[phylo_idx] .+ u_o[obs_idx]
    m.dist.poisson(exp.(mu), name = "obs", obs = y)
end
```

Mathematical Details

Here is the refined version of the mathematical specifications. I have aligned the notation with the rigorous standards established in the previous sections, formalized the matrix/vector notation, and elevated the variable descriptions to explicitly capture the statistical mechanics of the model.

Mathematical Details

The full hierarchical model for overdispersed phylogenetic count data is formally specified as follows:

$$y_i \sim \text{Poisson}(\lambda_i)$$

$$\log(\lambda_i) = \alpha + \beta x_i + u_{\text{phylo}[i]} + \epsilon_{\text{obs}[i]}$$

$$\mathbf{u}_{\text{phylo}} \sim \text{MultivariateNormal}(\mathbf{0}, \sigma^2_{\text{phylo}}\mathbf{A})$$

$$\epsilon_{\text{obs}[i]} \sim \text{Normal}(0, \sigma_{\text{obs}})$$

$$\alpha \sim \text{Normal}(0, 5), \quad \beta \sim \text{Normal}(0, 2)$$

$$\sigma_{\text{phylo}}, \sigma_{\text{obs}} \sim \text{HalfNormal}(0, 1)$$

Where:

- y_i denotes the observed discrete count for the i-th observation.
- λ_i represents the expected mean rate parameter of the Poisson likelihood for the i-th observation.
- x_i represents the value of the fixed environmental or biological covariate.
- $\mathbf{u}_{\text{phylo}}$ is the vector of phylogenetically structured varying intercepts. It is modeled as a multivariate normal distribution with covariance proportional to the phylogenetic variance-covariance (VCV) matrix $\mathbf{A}$. The index mapping, $u_{\text{phylo}[i]}$, assigns the species-specific evolutionary effect to the corresponding observation.

- $\epsilon_{\text{obs}[i]}$ is the observation-level random effect (OLRE), an unstructured, independent varying intercept drawn from a zero-mean normal distribution. This term is explicitly incorporated to absorb extra-Poisson variation (overdispersion) operating at the level of the individual datum.
- σ_{phylo} and σ_{obs} quantify the standard deviations of their respective varying effects, scaling the magnitude of variance attributable to shared evolutionary descent and residual individual-level overdispersion, respectively.

Example 3: Models with Repeated Measurements

In experimental or observational designs where multiple intra-specific individuals i are sampled per taxon, it is imperative to rigorously partition the phenotypic variance of the response y_i (`y`). We achieve this by estimating two distinct species-level components: a phylogenetically structured varying intercept, u_{phylo} (`u_p`), and an unstructured, phylogenetically independent species-specific effect, u_{spec} (`u_s`). This dual parameterization effectively isolates the variance attributable to shared evolutionary descent from idiosyncratic species-level adaptations or intra-class correlation.

The computational architecture requires a continuous response vector (`y`) and a fixed covariate vector x_i (`x`), both of length N (the total number of observations). The evolutionary relationships are parameterized using the Cholesky factor of the phylogenetic variance-covariance (VCV) matrix, $\mathbf{L}$ (`L`), of dimensions $M \times M$, where M represents the total number of unique taxa. To project these latent species-level effects onto the N individual observations, the model employs two indexing vectors: `phylo_idx` maps each data point to its corresponding phylogenetically structured effect, while `species_idx` maps the same observations to their independent, species-specific intercepts.

Implementation

Python

```
# Load and prepare data
df = m.load.phylo_repeated()
L_df = m.load.phylo_L_repeated()
species_to_idx = {sp: i for i, sp in enumerate(L_df.columns)}

m.data_on_model = {
    "y": jnp.array(df["y"].values),
    "x": jnp.array(df["x"].values),
    "phylo_idx": jnp.array(df["phylo"].map(species_to_idx).values, dtype=jnp.int32),
    "species_idx": jnp.array(df["species"].map(species_to_idx).values, dtype=jnp.int32),
```

```
    "L": jnp.array(L_df.values)
}

def model(y, x, phylo_idx, species_idx, L):
    # Priors
    intercept = m.dist.normal(0, 50, name="Intercept")
    b_x = m.dist.normal(0, 10, name="b_x")

    sd_p = m.dist.half_normal(20, name="sd_p")
    sd_s = m.dist.half_normal(20, name="sd_s")
    sigma = m.dist.half_normal(20, name="sigma")

    # Phylogenetic effect
    z_p = m.dist.normal(jnp.zeros(L.shape[0]), 1.0, name="z_p")
    u_p = jnp.matmul(L, z_p) * sd_p

    # Species effect (Independent)
    z_s = m.dist.normal(jnp.zeros(L.shape[0]), 1.0, name="z_s")
    u_s = z_s * sd_s

    # Predictor
    mu = intercept + b_x * x + u_p[phylo_idx] + u_s[species_idx]
    m.dist.normal(mu, sigma, name="obs", obs=y)

m.fit(model)
```

```
  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]
```

R

```
model <- function(y, x, phylo_idx, species_idx, L) {
    intercept <- bf.dist.normal(0, 50, name = "Intercept")
    b_x <- bf.dist.normal(0, 10, name = "b_x")
```

```
    sd_p <- bf.dist.half_normal(20, name = "sd_p")
    sd_s <- bf.dist.half_normal(20, name = "sd_s")
    sigma <- bf.dist.half_normal(20, name = "sigma")

    num_sp <- dim(L)[1]
    u_p <- (L %*% bf.dist.normal(rep(0, num_sp), 1, name="z_p")) * sd_p
    u_s <- bf.dist.normal(rep(0, num_sp), 1, name="z_s") * sd_s

    mu <- intercept + b_x * x + u_p[phylo_idx] + u_s[species_idx]
    bf.dist.normal(mu, sigma, obs = y)
}
```

Julia

```
@BF function model(y, x, phylo_idx, species_idx, L):
    intercept = m.dist.normal(0, 50, name = "Intercept")
    b_x = m.dist.normal(0, 10, name = "b_x")

    sd_p = m.dist.half_normal(20, name = "sd_p")
    sd_s = m.dist.half_normal(20, name = "sd_s")
    sigma = m.dist.half_normal(20, name = "sigma")

    num_sp = size(L, 1)
    u_p = (L * m.dist.normal(zeros(num_sp), 1, name="z_p")) .* sd_p
    u_s = m.dist.normal(zeros(num_sp), 1, name="z_s") .* sd_s

    mu = intercept + b_x * x + u_p[phylo_idx] + u_s[species_idx]
    m.dist.normal(mu, sigma, obs = y)
end
```

Mathematical Details

The full hierarchical model for continuous traits with intra-specific repeated measurements is formally specified as follows:

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + \beta x_i + u_{\text{phylo}[\text{species}[i]]} + u_{\text{spec}[\text{species}[i]]}$$

$$\mathbf{u}_{\text{phylo}} \sim \text{MultivariateNormal}(\mathbf{0}, \sigma^2_{\text{phylo}}\mathbf{A})$$

$$u_{\text{spec}[j]} \sim \text{Normal}(0, \sigma_{\text{spec}}) \quad \text{for } j \in \{1, \dots, M\}$$

Where:

- y_i denotes the observed continuous phenotypic trait for the i-th observation.
- μ_i represents the expected conditional mean for the i-th observation.
- x_i represents the value of the fixed environmental or biological covariate for the i-th observation.
- $\mathbf{u}_{\text{phylo}}$ is the vector of phylogenetically structured varying intercepts of length M. It is modeled as a multivariate normal distribution with covariance proportional to the phylogenetic variance-covariance (VCV) matrix $\mathbf{A}$. The index mapping, species$[i]$, assigns the shared evolutionary effect to all individuals belonging to the same taxon.
- $u_{\text{spec}[j]}$ represents the phylogenetically independent varying intercept for the j-th species. Modeled as an unstructured, zero-mean normal distribution, it captures species-specific adaptations and intra-class correlation (resemblance among individuals of the same species) that cannot be explained by shared ancestry alone.
- σ_{phylo} and σ_{spec} quantify the standard deviations of their respective varying effects, explicitly partitioning the inter-specific variance into phylogenetic and non-phylogenetic components.
- σ represents the residual standard deviation, capturing intra-specific (individual-level) phenotypic variation and measurement error.

Hierarchical Parameterization & Covariance

Technically, a Nested Varying effect structure can be readily implemented within BF. This hierarchical parameterization is particularly effective for modeling complex covariance structures across multiple tiers of evolutionary divergence (e.g., distinct populations or clades nested within focal species). By explicitly mapping these sub-specific strata, the model provides a more nuanced and biologically realistic partitioning of phenotypic variance compared to a standard additive formulation.

Example 4: Phylogenetic Meta-Analysis

Phylogenetic meta-analysis aggregates empirical effect sizes, y_i (`y`), extracted from multiple independent studies while rigorously controlling for the shared evolutionary history of the focal taxa. Unlike standard phylogenetic regressions, this framework explicitly accounts for varying measurement precision across studies by incorporating the known, observation-specific standard error, se_i (`se`), directly into the variance structure of the model.

The computational architecture necessitates two continuous data vectors of length N (the total number of observed effect sizes): the point estimates (`y`) and their associated standard errors (`se`). The evolutionary relationships are parameterized using the Cholesky factor of the phylogenetic variance-covariance (VCV) matrix, $\mathbf{L}$ (`L`), of dimensions $M \times M$, where M represents the total number of unique taxa. To properly partition the variance, the model employs two indexing vectors: `phylo_idx` maps the N observations to their corresponding phylogenetically structured effects, while `obs_idx` assigns a unique observation-level random effect (OLRE) to explicitly model residual, study-specific heterogeneity that is not explained by sampling variance or shared ancestry.

Implementation

Python

```python
# Load and prepare data
df = m.load.phylo_meta()
L_df = m.load.phylo_L_meta()
species_to_idx = {sp: i for i, sp in enumerate(L_df.columns)}
df["se"] = jnp.sqrt(1.0 / (jnp.array(df["N"].values) - 3.0))

m.data_on_model = {
    "y": jnp.array(df["y"].values),
    "se": jnp.array(df["se"].values),
    "phylo_idx": jnp.array(df["phylo"].map(species_to_idx).values, dtype=jnp.int32),
    "obs_idx": jnp.arange(len(df), dtype=jnp.int32),
    "L": jnp.array(L_df.values)
}

def model(y, se, phylo_idx, obs_idx, L):
    # Priors
    intercept = m.dist.normal(0.0, 10.0, name="Intercept")
    sd_obs = m.dist.half_normal(10, name="sd_obs")
    sd_phylo = m.dist.half_normal(10, name="sd_phylo")
```

```
    # Effects
    u_p = jnp.matmul(L, m.dist.normal(jnp.zeros(L.shape[0]), 1, name="z_p")) * sd_phylo
    u_o = m.dist.normal(jnp.zeros(len(y)), 1, name="z_o") * sd_obs

    # Mean
    mu = intercept + u_p[phylo_idx] + u_o[obs_idx]

    # Likelihood (Normal with fixed SE)
    m.dist.normal(mu, se, name="Y", obs=y)

m.fit(model)
```

```
  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]
```

R

```
model <- function(y, se, phylo_idx, obs_idx, L) {
    intercept <- bf.dist.normal(0, 10, name = "Intercept")
    sd_o <- bf.dist.half_normal(10, name = "sd_o")
    sd_p <- bf.dist.half_normal(10, name = "sd_p")

    u_p <- (L %*% bf.dist.normal(rep(0, dim(L)[1]), 1, name="z_p")) * sd_p
    u_o <- bf.dist.normal(rep(0, length(y)), 1, name="z_o") * sd_o

    mu <- intercept + u_p[phylo_idx] + u_o[obs_idx]
    bf.dist.normal(mu, se, obs = y)
}
```

Julia

```
@BF function model(y, se, phylo_idx, obs_idx, L):
    intercept = m.dist.normal(0, 10, name = "Intercept")
    sd_o = m.dist.half_normal(10, name = "sd_o")
    sd_p = m.dist.half_normal(10, name = "sd_p")

    u_p = (L * m.dist.normal(zeros(size(L, 1)), 1, name="z_p")) .* sd_p
    u_o = m.dist.normal(zeros(length(y)), 1, name="z_o") .* sd_o

    mu = intercept .+ u_p[phylo_idx] .+ u_o[obs_idx]
    m.dist.normal(mu, se, obs = y)
end
```

Mathematical Details

The full hierarchical framework for phylogenetic meta-analysis is formally specified as follows:

$$y_i \sim \text{Normal}(\mu_i, se_i)$$

$$\mu_i = \alpha + u_{\text{phylo}[i]} + \epsilon_{\text{obs}[i]}$$

$$\mathbf{u}_{\text{phylo}} \sim \text{MultivariateNormal}(\mathbf{0}, \sigma^2_{\text{phylo}}\mathbf{A})$$

$$\epsilon_{\text{obs}[i]} \sim \text{Normal}(0, \sigma_{\text{obs}})$$

Where:

- y_i denotes the i-th observed empirical effect size (e.g., Fisher's z-transformed correlation coefficient or Hedges' g).
- se_i represents the known, fixed standard error associated with the i-th effect size, which directly modulates the precision weighting of the observation within the likelihood function.
- μ_i represents the expected conditional mean (the latent true effect size) for the i-th observation.

- $\mathbf{u}_{\text{phylo}}$ is the vector of phylogenetically structured varying intercepts. Modeled as a multivariate normal distribution with covariance proportional to the phylogenetic variance-covariance (VCV) matrix $\mathbf{A}$, this term accounts for the shared evolutionary descent of the focal taxa. The index mapping, $u_{\text{phylo}[i]}$, assigns the species-specific evolutionary effect to the corresponding observation.
- $\epsilon_{\text{obs}[i]}$ is the observation-level random effect (OLRE). Parameterized as an unstructured, zero-mean normal distribution, it captures residual, study-specific heterogeneity (conceptually analogous to the between-study variance parameter, τ^2, in classical random-effects meta-analysis) that persists beyond known sampling variance (se_i) and shared phylogenetic history.
- σ_{phylo} and σ_{obs} quantify the standard deviations of their respective varying effects, explicitly scaling the magnitude of variance attributable to the phylogenetic signal versus residual study-level heterogeneity.

Example 5: Phylogenetic Varying Slopes

Phylogenetic varying slopes models extend the hierarchical framework by allowing the relationship between a fixed covariate, x_i (`x`), and the phenotypic response, y_i (`y`), to systematically covary with the phylogeny. This architecture is predicated on the biological assumption that ecological correlations or trait allometries are not static across clades, but themselves evolve over time.

The computational data structure necessitates a continuous response vector (`y`) and a covariate vector (`x`), both of length N (the total number of observations). The shared evolutionary history is parameterized using the Cholesky factor of the phylogenetic variance-covariance (VCV) matrix, $\mathbf{L}_A$ (`L_A`), of dimensions $M \times M$, where M denotes the total number of unique taxa. An indexing vector, `phylo_idx` (length N), maps the observations to their corresponding positions in the tree. Crucially, this advanced parameterization estimates both phylogenetically structured varying intercepts, $u_{\alpha,\text{phylo}}$, and phylogenetically structured varying slopes, $u_{\beta,\text{phylo}}$, thereby explicitly modeling the evolutionary divergence of predictor-response relationships.

Implementation

Python

```
# Load and prepare data
df = m.load.phylo_slopes()
L_df = m.load.phylo_L_slopes()
species_to_idx = {sp: i for i, sp in enumerate(L_df.columns)}
```

```
m.data_on_model = {
    "y": jnp.array(df["y"].values),
    "x": jnp.array(df["x"].values),
    "phylo_idx": jnp.array(df["phylo"].map(species_to_idx).values, dtype=jnp.int32),
    "L_A": jnp.array(L_df.values)
}

def model(y, x, phylo_idx, L_A):
    # Population-level effects
    intercept = m.dist.normal(0.0, 10.0, name="Intercept")
    b_x = m.dist.normal(0.0, 10.0, name="b_x")
    sigma = m.dist.half_normal(5, name="sigma")

    # Group-level sd and correlation
    sd_a = m.dist.half_normal(10, name="sd_a")
    sd_b = m.dist.half_normal(10, name="sd_b")
    L_corr = m.dist.lkj_cholesky(2, concentration=2.0, name="L_corr")

    # Combined L_sigma = diag(sds) @ L_corr
    L_sigma = jnp.diag(jnp.array([sd_a, sd_b])) @ L_corr

    # Standardized species effects (M x 2)
    z = m.dist.normal(jnp.zeros((L_A.shape[0], 2)), 1.0, name="z")
    U = jnp.matmul(jnp.matmul(L_A, z), L_sigma.T)

    u_a = U[:, 0]
    u_b = U[:, 1]

    mu = intercept + u_a[phylo_idx] + (b_x + u_b[phylo_idx]) * x
    m.dist.normal(mu, sigma, name="obs", obs=y)

m.fit(model)
```

```
  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]

  0%|          | 0/2000 [00:00<?, ?it/s]
```

R

```r
model <- function(y, x, phylo_idx, L_A) {
    intercept <- bf.dist.normal(0, 10, name = "Intercept")
    b_x <- bf.dist.normal(0, 10, name = "b_x")
    sigma <- bf.dist.half_normal(5, name = "sigma")

    sd_a <- bf.dist.half_normal(10, name = "sd_a")
    sd_b <- bf.dist.half_normal(10, name = "sd_b")
    L_corr <- bf.dist.lkj_cholesky(2, 2.0, name = "L_corr")

    L_sigma <- diag(c(sd_a, sd_b)) %*% L_corr

    num_sp <- dim(L_A)[1]
    z <- bf.dist.normal(matrix(0, num_sp, 2), 1, name = "z")
    U <- (L_A %*% z) %*% t(L_sigma)

    u_a <- U[, 1]
    u_b <- U[, 2]

    mu <- intercept + u_a[phylo_idx] + (b_x + u_b[phylo_idx]) * x
    bf.dist.normal(mu, sigma, name = "obs", obs = y)
}
```

Julia

```julia
@BF function model(y, x, phylo_idx, L_A):
    intercept = m.dist.normal(0, 10, name = "Intercept")
    b_x = m.dist.normal(0, 10, name = "b_x")
    sigma = m.dist.half_normal(5, name = "sigma")

    sd_a = m.dist.half_normal(10, name = "sd_a")
    sd_b = m.dist.half_normal(10, name = "sd_b")
    L_cor = m.dist.lkj_cholesky(2, 2.0, name = "L_cor")

    L_sigma = diagm([sd_a, sd_b]) * L_cor

    num_sp = size(L_A, 1)
    z = m.dist.normal(zeros(num_sp, 2), 1, name = "z")
    U = (L_A * z) * transpose(L_sigma)
```

```
    u_a = U[:, 1]
    u_b = U[:, 2]

    mu = intercept .+ u_a[phylo_idx] .+ (b_x .+ u_b[phylo_idx]) .* x
    m.dist.normal(mu, sigma, name = "obs", obs = y)
end
```

Mathematical Details

The hierarchical specification for the phylogenetic varying slopes model is:

$$y_i \sim \text{Normal}(\mu_i, \sigma)$$

$$\mu_i = \alpha + u_{\alpha,\text{phylo}[i]} + (\beta + u_{\beta,\text{phylo}[i]})x_i$$

$$\begin{pmatrix} \mathbf{u}_{\alpha,\text{phylo}} \\ \mathbf{u}_{\beta,\text{phylo}} \end{pmatrix} \sim \text{MultivariateNormal}(\mathbf{0}, \Sigma \otimes \mathbf{A})$$

Where:

- y_i is the observed phenotype for the i-th observation.
- x_i represents the value of the predictor variable for the i-th observation.
- $u_{\alpha,\text{phylo}}$ (`u_a`) and $u_{\beta,\text{phylo}}$ (`u_b`) are the phylogenetically correlated varying intercepts and slopes, respectively, which model the evolutionary divergence in both trait means and their mutual correlations.
- $\mathbf{A}$ denotes the phylogenetic kinship or variance-covariance matrix derived from the tree topology.
- Σ is a 2×2 covariance matrix representing the across-species variance of intercepts and slopes, and their evolutionary correlation. It is parameterized using a Cholesky factor `L_sigma` in the code.
- α and β represent the population-level intercept and slope, respectively.
- σ is the residual standard deviation capturing intra-specific variance or measurement error.

Here is the drafted notes section, strictly maintaining the rigorous academic tone, precise mathematical terminology, and formal markdown structure we have established for your documentation.

Notes

Methodological Frontiers: Gaussian Processes and Advanced Inferences

Gaussian Processes as an Alternative to Varying Effects While this chapter formalizes phylogenetic non-independence via structured varying intercepts and slopes, this architecture is mathematically homologous to a Gaussian Process (GP) defined over the discrete space of the tree's terminal nodes (McElreath 2018). In the standard Generalized Phylogenetic Multilevel Model (GPMM), the phylogenetic variance-covariance (VCV) matrix **A** implicitly assumes that continuous trait evolution proceeds via a neutral **Brownian Motion (BM)** model. By reframing the phylogenetic effect as a Gaussian Process, researchers can employ parameterized covariance kernel functions. For instance, incorporating an exponential decay parameter into the covariance kernel allows the model to capture an **Ornstein-Uhlenbeck (OU)** process Uyeda and Harmon (2014), thereby rigorously modeling stabilizing selection toward phenotypic optima.

A Crucial Prerequisite: The Two-Phase Analytical Pipeline It is critical to note that the statistical validity of the GPMMs and GP models discussed above fundamentally depends on the accuracy of the VCV matrix **A**. This highlights a foundational dichotomy in modern comparative biology: *the Two-Phase Analytical Pipeline*. The models in this chapter operate strictly in Phase 2 (*Phylogenetic Comparative Methods*), analyzing macroevolutionary trait dynamics along the branches of a pre-existing tree. However, the generation of that tree constitutes Phase 1 (*Phylogenetic Inference*), a mathematically distinct step that models the biochemical evolution of molecular sequences to estimate topology and absolute divergence times (the chronogram). If Phase 1 sequence models fail to capture biological reality, the resulting branch lengths will systematically bias the Phase 2 trait models.

Methodological Frontiers in Phylogenetic Inference (Tree Generation)

To generate the highly accurate VCV matrices required for robust downstream trait modeling, Phase 1 inference deploys an increasingly sophisticated suite of sequence evolution models:

- **Modeling Evolutionary Heterogeneity:** Molecular evolution is inherently heterogeneous. To prevent severe topological biases, modern inferences model spatial heterogeneity across the genome using discrete **Gamma distributions (+Γ)** for among-site rate variation (Yang 1994). Similarly, to address temporal heterogeneity, **Relaxed Molecular Clocks** treat branch-specific evolutionary rates as varying effects drawn from parametric distributions (e.g., uncorrelated log-normal), allowing branch lengths to stochastically deviate from a strict clock assumption (Drummond et al. 2006).
- **Site-Heterogeneous and Structural Models:** Contemporary Bayesian and

Maximum Likelihood inferences integrate complex profile *mixture models* (e.g., CAT, LG+C60) Wang et al. (2018) to accurately capture the biochemical constraints of amino acid replacements across different protein domains.

- **The Multispecies Coalescent (MSC):** Moving beyond traditional sequence concatenation, the MSC explicitly models *incomplete lineage sorting (ILS)*, mathematically reconciling incongruent gene trees into a statistically rigorous, species-level phylogeny Yang and Rannala (2021).
- **Deep Learning and Graph Embeddings:** At the forefront of computational phylogenetics, novel architectures utilizing self-attention mechanisms (*PhyloTransformers*) and Graph Convolutional Networks (GCNs) are being trained to infer complex topologies directly from raw sequences, capturing non-linear evolutionary dynamics that evade traditional substitution matrices Braichenko, Borges, and Kosiol (2025).

Reference(s)

Beaulieu, Jeremy M, Dwueng-Chwuan Jhwueng, Carl Boettiger, and Brian C O'Meara. 2012. "Modeling Stabilizing Selection: Expanding the Ornstein–Uhlenbeck Model of Adaptive Evolution." *Evolution* 66 (8): 2369–83.

Braichenko, Svitlana, Rui Borges, and Carolin Kosiol. 2025. "Phylogenetic Methods Meet Deep Learning." *Genome Biology and Evolution* 17 (10): evaf177.

brms Development Team. 2025. *Estimating Phylogenetic Multilevel Models with Brms*. https://cran.r-project.org/web/packages/brms/vignettes/brms_phylogenetics.html.

Butler, Marguerite A, and Aaron A King. 2004. "Phylogenetic Comparative Analysis: A Modeling Approach for Adaptive Evolution." *The American Naturalist* 164 (6): 683–95.

Drummond, Alexei J, Simon YW Ho, Matthew J Phillips, and Andrew Rambaut. 2006. "Relaxed Phylogenetics and Dating with Confidence." *PLoS Biology* 4 (5): e88.

Hansen, Thomas F. 1997. "Stabilizing Selection and the Comparative Analysis of Adaptation." *Evolution* 51 (5): 1341–51.

Heled, Joseph, and Alexei J Drummond. 2010. "BayesForge of Species Trees from Multilocus Data." *Molecular Biology and Evolution* 27 (3): 570–80.

Lartillot, Nicolas, and Hervé Philippe. 2004. "A Bayesian Mixture Model for Across-Site Heterogeneities in the Amino-Acid Replacement Process." *Molecular Biology and Evolution* 21 (6): 1095–1109.

Le, Si Quang, Cuong Cao Dang, and Olivier Gascuel. 2012. "Modeling Protein Evolution with Several Amino Acid Replacement Matrices Depending on Site Rates." *Molecular Biology and Evolution* 29 (10): 2921–36.

McElreath, Richard. 2018. *Statistical Rethinking: A Bayesian course with examples in R and Stan*. Chapman; Hall/CRC.

Rannala, Bruce, and Ziheng Yang. 2003. "Bayes Estimation of Species Divergence Times and Ancestral Population Sizes Using DNA Sequences from Multiple Loci." *Genetics* 164 (4): 1645–56.

Sun, Chaoyue, Yanjun Li, Simone Marini, Alberto Riva, Dapeng Oliver Wu, Ruogu Fang, Marco Salemi, and Brittany Rife Magalis. 2024. "Phylogenetic-Informed Graph Deep Learning to Classify Dynamic Transmission Clusters in Infectious Disease Epidemics." *Bioinformatics Advances* 4 (1): vbae158.

Uyeda, Josef C, and Luke J Harmon. 2014. "A Novel Bayesian Method for Inferring and Interpreting the Dynamics of Adaptive Landscapes from Phylogenetic Comparative Data." *Systematic Biology* 63 (6): 902–18.

Wang, Hua-Chen, Bui Quang Minh, Edward Susko, and Andrew J Roger. 2018. "Modeling Site Heterogeneity with Posterior Mean Site Frequency Profiles Accelerates Accurate Phylogenomic Estimation." *Systematic Biology* 67 (2): 216–35.

Yang, Ziheng. 1994. "Maximum Likelihood Phylogenetic Estimation from DNA Sequences with Unequal Rates over Sites." *Journal of Molecular Evolution* 39 (3): 306–14.

Yang, Ziheng, and Bruce Rannala. 2021. "Multispecies Coalescent and Its Applications to Infer Species Phylogenies and Cross-Species Gene Flow." *National Science Review* 8 (12): nwab127.

Zou, Zhengting, Huaien properly Zhang, Yuantong Li, and Zhen Su. 2020. "Deep Learning Inference of Phylogeny from Sequence Alignments." *Systematic Biology* 69 (6): 1098–1111.