

Version dated: May 20, 2016

RH: PHYLOGENETIC FRAMEWORK FOR TRAIT COEVOLUTION

A unifying comparative phylogenetic framework including traits coevolving across interacting lineages

MARC MANCEAU^{1,3,4}, AMAURY LAMBERT^{2,3}, HÉLÈNE MORLON⁴

¹*Muséum National d'Histoire Naturelle, 75005 Paris, France;*

²*Laboratoire Probabilités et Modèles Aléatoires, UPMC Univ Paris 06, 75005 Paris, France;*

³*Center for Interdisciplinary Research in Biology, Collège de France, CNRS UMR 7241, 75005 Paris, France;*

⁴*Institut de Biologie, École Normale Supérieure, CNRS UMR 8197, 75005 Paris, France*

Corresponding author: Marc Manceau, Institut de Biologie de l'École Normale Supérieure, 46 rue d'Ulm, 75005 Paris, France; E-mail: marc.manceau@ens.fr

Abstract.— Models of phenotypic evolution fit to phylogenetic comparative data are widely used to make inferences regarding the tempo and mode of trait evolution. A wide range of models is already available for this type of analysis, and the field is still under active development. One of the most needed developments concerns models that better account for the effect of within- and between-clade interspecific interactions on trait evolution, that can result from processes as diverse as competition, predation, parasitism, or mutualism. Here, we begin by developing a very general comparative phylogenetic framework for (multi)-trait evolution that can be applied to both ultrametric and non-ultrametric trees. This framework not only encapsulates all previous classical models of univariate and multivariate phenotypic evolution, but also paves the way for the consideration of a much broader series of models in which lineages co-evolve, meaning that trait changes in one lineage are influenced by the value of traits in other, interacting lineages. Next, we provide a standard way for deriving the probabilistic distribution of traits at tip

branches under our framework. We show that a multivariate normal distribution remains the expected distribution for a broad class of models accounting for interspecific interactions. Our derivations allow us to fit various models efficiently, and in particular greatly reduce the computation time needed to fit the recently proposed phenotype matching model. Finally, we illustrate the utility of our framework by developing a toy model for mutualistic coevolution. Our framework should foster a new era in the study of coevolution from comparative data.

(Keywords: comparative phylogenetics, trait evolution, coevolution, interspecific interaction, character displacement, linear stochastic differential equations)

Evolutionary biologists have long been interested in the long-term evolution of phenotypic traits (Simpson 1944). In 1973, Felsenstein introduced one of the first models of phenotypic evolution, with the initial goal to account for shared ancestry when testing for statistical correlation between pairs of traits in extant species. In this founding paper, Felsenstein proposed that a one-dimensional quantitative trait evolving on a tree could be modeled as a Brownian process that splits into two independent Brownian processes at branching times. This model mimics a trait that would evolve as a mere effect of stochastic drift; it is now often used as a null model, but also to estimate the relative lability (or rate of evolution) of various traits in a given group of organisms or of a given trait across different groups of organisms (Thomas et al. 2006; Harmon et al. 2010).

Since these early developments, evolutionary biologists have designed a series of models to better understand the evolutionary processes that shape phenotypic evolution (see Pennell and Harmon 2013 for a review). The Ornstein-Uhlenbeck (OU) process has been proposed to model evolution under stabilizing selection, i.e. with a selective pressure pushing trait values toward a given optimum (Hansen 1997; Hansen and Martins 1996; Butler and King 2004). The ACDC model has been proposed to account for accelerating (AC) or decelerating (DC) rates of

phenotypic evolution through time (Blomberg et al. 2003). The latter scenario, where the evolutionary rate is high early in the history of a clade and subsequently declines toward the present, well known as the early burst (EB) model, has often been used to test support for adaptive radiation theory (Harmon et al. 2010; Moen and Morlon 2014). These univariate models representing the evolution of a single trait have been extended to multivariate models representing the simultaneous evolution of multiple traits, which permit investigators to directly test hypotheses about the coevolution between several phenotypic traits (Hansen et al. 2008; Bartoszek et al. 2012; Jhvueng and Maroulas 2014). Other extensions have been developed to account for variations in model parameters across clades (O’Meara et al. 2006; Revell and Collar 2009; Eastman et al. 2011; Butler and King 2004; Beaulieu et al. 2012). Finally, some of these models have been developed in the context of phylogenies including fossil data (i.e. non ultrametric trees, see Ruta et al. 2006; Slater 2015) in addition to phylogenies with only extant taxa (i.e. ultrametric trees). Most of these models have been implemented in open-access packages (Martins 2004; Harmon et al. 2008; Butler and King 2004; Thomas and Freckleton 2012; Clavel et al. 2015; Morlon et al. 2015), allowing their application to a broad variety of questions and datasets (see, e.g. Labra et al. 2009; Mahler et al. 2010; Dale et al. 2015; Quintero et al. 2015; Slater 2015).

Despite these developments, most currently available models ignore the effect of interspecific interactions on trait evolution. Given the importance of species interactions in classical evolutionary theories, such as Simpson’s adaptive radiation (Simpson 1944), Ehrlich & Raven’s escape and radiate (Ehrlich and Raven 1964) and Van Valen’s Red Queen (Van Valen 1973) theories, building models that better account for such interactions is fundamental. In a first attempt to take into account the role of competition for niche space on character evolution, a diversity-dependent (DD) model has been introduced, where the rate of phenotypic evolution declines as the number of lineages in the clade increases (Mahler et al. 2010; Weir and Mursleen 2013). While this model represents an important first step, it still assumes that trait changes in one lineage are independent from the value of traits in other, interacting lineages, therefore ignoring the widespread idea of trait- (or ecologically-) driven interspecific interactions. More

recently, the phenotype matching (PM) model relaxed these hypotheses and more explicitly accounted for interspecific interactions by modeling either the attraction or the repulsion of traits from a clade-wise average trait value (Nuismer and Harmon 2014; Drury et al. 2016). In the first case, referred to as matching mutualism, species traits tend to converge to similar values, whereas in the second case, referred to as matching competition, species traits tend to diverge.

The comparative phylogenetic approach developed by Drury et al. (2016) is one of the first that allows fitting a model where the evolution of trait values in one lineage is influenced by the trait values of other lineages. This approach focused on the evolution of traits within one clade. While within-clade interactions can be particularly relevant for some types of interactions (e.g. in the case of competitively driven character displacement, Brown and Wilson 1956), the effect of other types of antagonistic or mutualistic interactions on trait evolution is often most relevant between distantly related species. For example, host-parasite interactions are thought to drive a coevolutionary race between traits involved in host defence and parasite ability to infect. Similarly, prey-predator interactions may lead to the coevolution of prey traits involved in camouflage, repulsion, or escape strategies, together with predator traits involved in the ability to detect and capture its prey (Ehrlich and Raven 1964; Dawkins and Krebs 1979). Mutualistic plant-pollinator interactions also are thought to drive the coevolution between plant traits involved in pollen accessibility or flower attractiveness to their pollinator (secondary metabolites, floral traits), and pollinator traits involved in the ability to detect suitable plants and to exploit plant rewards (Fenster et al. 2004; Weiblen 2004; Sletvold et al. 2016). While these types of biotic interactions likely play a key role in trait evolution and have been crucial in the development of coevolutionary theories (Ehrlich and Raven 1964; Van Valen 1973), there currently exists no framework for fitting models of phenotypic evolution incorporating the effect of clade-clade interactions.

The current paper expands the work of Bartoszek et al. (2012) who presented a unified framework for studying coevolving traits in independently evolving lineages by providing a unified framework for coevolving traits in coevolving lineages. Our framework, based on linear stochastic differential equations (SDE), encompasses all models of continuous (multi-)trait

evolution mentioned above, and allows the treatment of a much broader set of models. The goal of the paper is two-fold: first, by providing general solutions to the distribution of traits at tip branches under our unified framework, we hope to help users to find their way in a dense and potentially overwhelming literature; second, by showing how the framework can be used to treat a broad class of within-clade and clade-clade coevolutionary scenarios, we hope to foster the development of models to test long-standing hypotheses on the role of competition, predation, parasitism and mutualism in evolution.

We begin by presenting our framework and showing how previous models as well as novel clade-clade coevolutionary models fit within this framework; next, we provide general solutions for the distribution of tip trait values under this framework; then, we illustrate how the framework can be used to study a toy model of clade-clade coevolution.

A GENERAL FRAMEWORK FOR PHENOTYPIC EVOLUTION

Notation for trees and traits

We introduce a general formalism to study (multi-)trait (co)evolution when the interaction between lineages within a clade or among several clades potentially affects how traits evolve. We consider a single or several clades represented by a single or several fixed, binary, time-calibrated phylogenetic trees (non-necessarily ultrametric, i.e. the trees can include fossils). Time t runs from the root of the oldest tree ($t = \tau_0 = 0$) to the most recent tip of all trees ($t = T$ is the present if at least one of the phylogenies includes extant species). The K successive branching and extinction times when considering the various trees altogether are denoted by $(\tau_i)_{i=1}^K$ and the time-intervals between two such events are called epochs, following Butler and King (2004). We denote by n_t the total number of lineages that arose before (and at) time t .

In the case of trait evolution within a single clade (Fig. 1), we assign numbers (from 1 to n_t) to lineages by order of arrival. At each branching event τ , one daughter lineage inherits the number assigned to the ancestral lineage while the other one is assigned n_τ .

We model the evolution of d one-dimensional quantitative traits. We denote by $X_t^{(i,j)}$ the

value of trait j ($1 \leq j \leq d$) on branch i at time t and X_t the column vector containing the values of all traits on all lineages at time t , ordered as follows :

$$X_t = {}^{tr}(X_t^{(1,1)}, X_t^{(1,2)}, \dots, X_t^{(1,d)}, X_t^{(2,1)}, \dots, X_t^{(n_t,d)}), \text{ where } {}^{tr} \text{ stands for the transposition.}$$

In the case of trait evolution in c distinct (co)evolving clades, we begin by arbitrarily ordering the clades from 1 to c ; then, we assign numbers to lineages following the formalism introduced above, first numbering lineages from clade 1, then clade 2, and so on. As above, we denote by X_t the column vector containing the values of all traits on all lineages at time t , which now is a concatenation of the c column vectors corresponding to each clade.

Trait evolution through time

Given one (or several) phylogenetic tree(s), a model of phenotypic evolution is entirely defined by initial conditions X_0 on the trait values at the root(s) and a set of rules dictating how the vector of traits X_t is updated (i) at branching times, (ii) on each epoch (i.e. between two branching or extinction times), and (iii) after a death time. These rules are illustrated in Figure 1 for a single trait evolving on a single small tree.

In line with most models of phenotypic evolution, we consider anagenetic character evolution, meaning that traits do not change at cladogenesis. Hence, at a given branching time τ , each of the daughter lineages inherits the trait value of their mother lineage. Our framework can easily be modified to treat non binary trees including polytomies. In practice, in the case of evolution within a single clade, the new vector X_τ is obtained by concatenating the d trait values of the branching lineage at time τ at the end of $X_{\tau-}$ (where $\tau-$ is the time just preceding the branching event). In the case of evolution in several clades, the new vector X_τ is obtained by inserting the d trait values of the branching lineage at time τ at the appropriate location in $X_{\tau-}$ (i.e., at the end of the part of $X_{\tau-}$ corresponding to the clade in which the branching event is occurring).

On each given epoch (τ_i, τ_{i+1}) ($i \in \{0, 1, \dots, K-1\}$), we assume that the evolution of the d traits on the n lineages is driven by a linear stochastic differential equation of the form :

$$\begin{cases} dX_t = (a_i(t) - A_i X_t)dt + \Gamma_i(t)dW_t \\ X(\tau_i) = X_{\tau_i} \end{cases} \quad (1)$$

where a_i is a vector of \mathbb{R}^{nd} whose coefficients can vary with time, A_i is a constant square matrix of size $\mathbb{R}^{nd} \times \mathbb{R}^{nd}$, Γ_i is a square matrix of the same size whose coefficients can vary with time, and W_t is a nd -Brownian motion (i.e. a vector composed of nd independent standard Brownian motions). Intuitively, the deterministic part $(a_i(t) - A_i X_t)dt$ reflects the direct effects of trait values for species in the clade(s) at time t on the evolution of these traits, including the effect of a trait value in one lineage on both its own evolution (as in the OU process) and the evolution of traits in the other lineages (as in the PM process); the stochastic part $\Gamma_i(t)dW_t$ reflects drift and the environmental noise influencing trait evolution. It has been proposed that correlations within the covariance matrix Γ_i represent non-causal correlations, for example linked to joint evolutionary responses to shared environmental conditions, while correlations within the interaction matrix A_i represent causal effects (Reitan et al. 2012; Liow et al. 2015). More work is needed to assess the relevance of the ‘causal/non-causal’ dichotomy, and the difference of patterns it can yield in the context of trait evolution on phylogenies. For simplicity, in the present paper, we will stick to the term ‘correlations’, and consider only models making the simplifying assumption that Γ_i is diagonal, but the framework could be equally adapted to incorporate correlations through these Γ_i matrices.

Finally, when a lineage goes extinct at a given time τ , its d trait values no longer evolve (i.e. they are frozen at the extinction time), and they no longer have any influence on the evolution of the traits of other lineages until reaching the end of the process at time $t = T$. In practice, this means that the vector X_τ is simply equal to $X_{\tau-}$, and that the d lines and columns in a_i , A_i and Γ_i corresponding to the now extinct lineage are all set to zero.

We will show later that this general formulation encapsulates all classical models of phenotypic evolution, ensures analytical tractability, and further allows the incorporation of a broad set of interspecific coevolutionary scenarios.

Given the above, initial conditions on X_0 , and the collection of (a_i) , (A_i) and (Γ_i) on each epoch fully define a process of trait evolution on one or several trees.

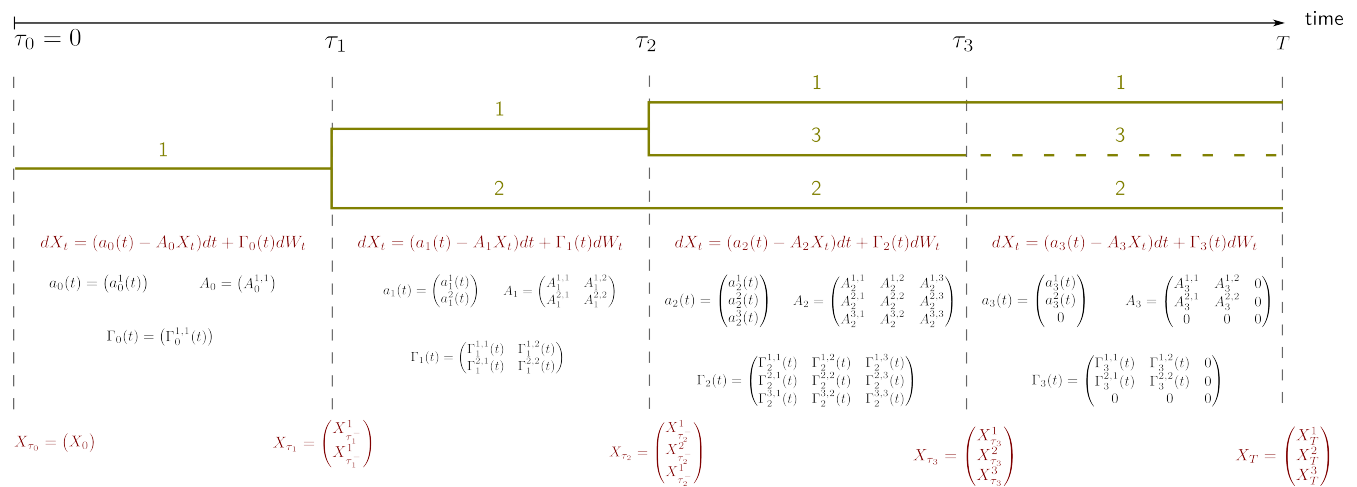


FIGURE 1: Formalism used throughout the paper, to model the evolution of one trait on a non-ultrametric tree. Epochs are separated with vertical dashed lines.

All models written under the formalism that we propose can easily be simulated numerically. First, the whole trajectory of the process can be simulated using a numerical scheme for SDE such as the Euler-Maruyama scheme (Gardiner et al. 1985) on each epoch, and augmenting the vector of traits at branching times with traits corresponding to the branching lineage (see Appendix D.1 and Fig. 5). Second, we show in the next section how to compute numerically the tip distribution. Tip values can then directly be drawn in a fast way from the tip distribution.

Application: existing and novel models of trait evolution

We first show that the general formulation above encapsulates all classical models of phenotypic evolution, before showing how it further allows considering a much broader set of models, including models of within and between clades coevolution.

Models of phenotypic evolution have traditionally been characterized by a stochastic differential equation specifying how a given trait evolves along a single lineage. Applying Equation (1) to trait k , on epoch i yields:

$$dX_t^{(k)} = \left(a_i^{(k)}(t) - \sum_{l=1}^{n_t d} A_i^{(k,l)} X_t^{(l)} \right) dt + \sum_{l=1}^{n_t d} \Gamma_i^{(k,l)}(t) dW_t^{(l)} \quad (2)$$

where the two sums are taken over all traits and all lineages. The term $\sum_{l=1}^{n_t d} A_i^{(k,l)} X_t^{(l)}$ is the term that specifies how the value of trait k and all other traits in all other lineages influence the evolution of trait k . Given a well-known differential equation specifying how a given trait evolves along a single lineage for a previously proposed model of phenotypic evolution (second column in Table 1), deriving the corresponding expressions for a , A and Γ using Equation (2) is straightforward. Table 1 summarizes these expressions for existing univariate models running on ultrametric trees.

The first three models (BM, ACDC and DD) are models in which trait evolution along a lineage is influenced neither by the trait value of this lineage nor the trait value of any other lineage. The corresponding A matrices are null matrices, as would be the case for any model with the latter property. The fourth model (OU) is a model in which trait evolution along a lineage is influenced by its own trait value, but not the trait values of other lineages. The corresponding A matrix is diagonal, as would be the case for any model with this property. Finally, the last model (PM) is a model in which trait evolution along a lineage is influenced by its own trait value and the trait values of other lineages, such that A has non-negative off-diagonal values. A remarkable property of A under this model is that all its off-diagonal values are identical. This is explained by the fact that the PM model is a neutral model, in the sense that the effect $A^{(k,l)}$ of lineage l on lineage k is the same for all lineages $k \neq l$. All other models in which the off-diagonal elements of A are identical would have this same property, known in probability theory as exchangeability.

Several variations around these models can still be embedded in our general framework: i) Models in which the rate of phenotypic evolution depends on a variable $Y(t)$ that itself varies through time (see, e.g. global temperature $T(t)$ in Clavel and Morlon 2016) can be formalised similarly to ACDC, with time t replaced by $Y(t)$. ii) Models accounting for the biogeographic background in which species coevolved (e.g. all the “+GEO” models in Drury et al. 2016) can be incorporated in our framework through the design of the A matrix (see details in Appendix C.3). iii) Considering non-ultrametric trees including fossils amounts to replacing vector V and

Key	Model name			
	Evolution along lineage k	a	A	Γ
BM	Brownian motion, random genetic drift $dX_t^{(k)} = \sigma dW_t^{(k)}$	0	0	σI
ACDC	Accelerating or decelerating rate, early burst $dX_t^{(k)} = \sigma_0 e^{rt} dW_t^{(k)}$	0	0	$\sigma_0 e^{rt} I$
DD	Diversity-Dependent $dX_t^{(k)} = \sigma_0 e^{rn_t} dW_t^{(k)}$	0	0	$\sigma_0 e^{rn_t} I$
OU	Ornstein-Uhlenbeck, stabilizing selection $dX_t^{(k)} = \psi(\theta - X_t^{(k)})dt + \sigma dW_t^{(k)}$	$\psi\theta V$	ψI	σI
PM	Phenotype Matching $dX_t^{(k)} = \psi(\theta - X_t^{(k)})dt$ $+ S \left(\frac{1}{n_t} \sum_{l=1}^{n_t} X_t^{(l)} - X_t^{(k)} \right) dt + \sigma dW_t^{(k)}$	$\psi\theta V$	$(\psi + S)I - \frac{S}{n_t}U$	σI

TABLE 1: Expression of a , A and Γ for models of trait evolution that have been proposed in the literature. The unity vector (vector full of 1) is denoted by V , I refers to the identity matrix (diagonal matrix with diagonal values equal to 1), and U refers to the unity matrix (matrix full of 1). Their size is the same as the size of the vector of traits X_t considered. Parameters are σ : rate of neutral phenotypic evolution; ψ : strength of stabilizing selection; θ : optimal phenotype; S : strength of between-lineage competition driving individual phenotypes away from clade-wise average phenotype; σ_0 : rate of phenotypic evolution at the root of the tree; r : parameter controlling the exponential rise or decay of the rate of phenotypic evolution with time (ACDC) or with the number of lineages (DD).

matrices I and U by their homologs V_{alive} , I_{alive} and U_{alive} , where the subscript specifies that the vector and matrices have 0 on lines and columns corresponding to lineages that are extinct in the given epoch. iv) Considering subclades in which trait evolution follows distinct modes or similar modes with distinct parameter values (as in Butler and King 2004) is also straightforward. One just needs to specify distinct parameters in a , A and Γ on the lines and columns corresponding to

lineages in the distinctive subclade. v) Multivariate trait evolution models, in which several distinct traits evolve in a correlated manner (Hansen et al. 2008; Bartoszek et al. 2012) are easily written in our framework, as shown with some examples in Appendix B.2. In multivariate models with lineages evolving independently from one another (e.g. multivariate combinations of BM, ACDC, DD and OU models), A and Γ are block diagonal matrices, with blocks of size the number of traits, each of them describing correlated multivariate evolution along a particular lineage. In this case, trait-trait correlations introduced through the A matrix correspond, as in Bartoszek et al. (2012), to the case when a given trait on a lineage is attracted to (or repulsed from) a linear combination of other traits in this lineage.

By considering previous models under this light, it becomes very clear that the set of models that have been considered so far represents a very small fraction of all the models that could potentially be considered. In particular the A matrix, which dictates how the value of a given trait influences the evolution of other traits – either different traits in the same lineage, or the same trait in other lineages, or yet different traits in other lineages – has so far been very constrained. It has been considered to be zero (BM, ACDC, DD), diagonal (OU), block diagonal (multivariate), and only recently with non-zero off-diagonal values (PM). Relaxing these constraints means that a much broader array of models incorporating the effect of interspecific interactions on phenotypic evolution can be considered. In particular, lineages do not need to be interchangeable. Evolution in complex networks of interactions can be considered by designing a priori the A matrix according to the known network. The effect of clade-clade interactions can be modeled by filling the A matrix with non-zero entries $A^{(k,l)}$ with k and l corresponding to lineages from different clades. For example, under a scenario of two clades coevolving with no effect of within-clade interactions, this leads to a A matrix with two off-diagonal blocks.

We can thus imagine a variety of coevolutionary scenarios, the only major constraint being that the effect of a trait value on the evolution of other traits is assumed to be linear (Equations (1) & (2)). Given a scenario, we can write the corresponding evolution of each trait on a given lineage on each epoch (Equation (2)), and deduce the collection of (a_i) , (A_i) and (Γ_i) defining the evolutionary process (Equation (1)). Below, we first show how to derive the

probabilistic distribution of traits at tip branches for any model that can be written under this framework before illustrating the approach with a particular model of clade-clade interaction.

DISTRIBUTION OF TIP TRAIT VALUES

The distribution of traits is Gaussian

Deriving the probabilistic distribution of traits at tip branches is key to our ability to fit phenotypic models to comparative data using maximum likelihood or Bayesian approaches. It also provides a very efficient way to simulate tip values for specific models, by drawing from the expected tip distribution.

We show (Appendix A.1) that if X_0 has a Gaussian distribution (including the particular case when X_0 is constant) and X_t evolves according to our general framework, then X_t remains a Gaussian vector at each time t . The trait vector X_t , of size $n_t d$, is thus uniquely defined by its expectation vector m_t and covariance matrix Σ_t , and has the following density:

$$\forall x \in \mathbb{R}^{n_t d}, \quad f(x) = \frac{1}{\sqrt{(2\pi)^{n_t d} \det(\Sigma_t)}} e^{-\frac{1}{2} \text{tr}(x - m_t) \Sigma_t^{-1} (x - m_t)}$$

In particular, the distribution of tip trait values at present time T is Gaussian with expectation vector m_T and covariance matrix Σ_T . We can compute m_T and Σ_T iteratively: starting with initial conditions m_0 and Σ_0 for $X_{\tau_0} = X_0$, we compute, until reaching the present:

1. $m_{\tau_{i+1}^-}$ and $\Sigma_{\tau_{i+1}^-}$ at the end of each epoch i
2. $m_{\tau_{i+1}}$ and $\Sigma_{\tau_{i+1}}$ at the branching time τ_{i+1}

Evolution of the distribution on each epoch

Knowing the expectation vector and covariance matrix $(m_{\tau_i}, \Sigma_{\tau_i})$ at the beginning of epoch i , we show (Appendix A.2) that $m_{\tau_{i+1}^-}$ and $\Sigma_{\tau_{i+1}^-}$ at the end of epoch i are given by the following

analytical expressions:

$$m_{\tau_{i+1}^-} = e^{(\tau_i - \tau_{i+1})A_i} m_{\tau_i} + \int_{\tau_i}^{\tau_{i+1}} e^{(s - \tau_{i+1})A_i} a_i(s) ds \quad (3a)$$

$$\Sigma_{\tau_{i+1}^-} = (e^{(\tau_i - \tau_{i+1})A_i}) \Sigma_{\tau_i}^{tr} (e^{(\tau_i - \tau_{i+1})A_i}) + \int_{\tau_i}^{\tau_{i+1}} (e^{(s - \tau_{i+1})A_i} \Gamma_i(s))^{tr} (e^{(s - \tau_{i+1})A_i} \Gamma_i(s)) ds \quad (3b)$$

Alternatively, we can write the evolution of m and Σ on epoch i as a set of ordinary differential equations (ODE), and integrate these ODEs numerically, with initial conditions given by $(m_{\tau_i}$ and $\Sigma_{\tau_i})$. Each component k (resp. (k, l)) of the expectation vector (resp. covariance matrix) evolves on epoch i according to (Appendix A.3) :

$$\frac{d}{dt} m_t^{(k)} = a_i^{(k)}(t) - \sum_{m=1}^{n_t d} A_i^{(k,m)} m_t^{(m)} \quad (4a)$$

$$\frac{d}{dt} \Sigma_t^{(k,l)} = - \sum_{m=1}^{n_t d} A_i^{(k,m)} \Sigma_t^{(m,l)} + A_i^{(l,m)} \Sigma_t^{(m,k)} - \Gamma_i^{(l,m)}(t) \Gamma_i^{(k,m)}(t) \quad (4b)$$

Equations (3a, 3b) and the ODE system described by Equations (4a, 4b) are mathematically equivalent. The first formulation is more computationally efficient when the integrals can be simplified analytically. For example when A is symmetric Equations (3a, 3b) can be simplified (Appendix C.1) and computed very efficiently. The second one provides a more intuitive interpretation of the components that influence the evolution of trait distribution, and is easily implementable for any model.

Evolution of the distribution at branching times

Knowing the expectation vector and covariance matrix $(m_{\tau_{i+1}^-}, \Sigma_{\tau_{i+1}^-})$ at the end of epoch i , which precedes the branching of a given lineage j , we build $m_{\tau_{i+1}}$ and $\Sigma_{\tau_{i+1}}$ at the branching event, as illustrated in Figure 2.

Recall that in the case of evolution within a single clade, $X_{\tau_{i+1}}$ is obtained by concatenating the d trait values of lineage j at time τ_{i+1}^- at the end of $X_{\tau_{i+1}^-}$. The d new components in $X_{\tau_{i+1}}$ are thus the exact copies of the trait values of lineage j , and have the same expectation and covariance matrix. Hence, the expectation vector $m_{\tau_{i+1}}$ is simply obtained by

concatenating $m_{\tau_{i+1}}^-$ with the d components of $m_{\tau_{i+1}}^-$ corresponding to lineage j . The covariance matrix $\Sigma_{\tau_{i+1}}$ is obtained as follows: the covariance matrix corresponding to the previously existing lineages is unchanged, given by $\Sigma_{\tau_{i+1}}^-$; to this main block, we add below (and to the right) a copy of the d lines (respectively the d columns) corresponding to the covariances between the d traits in lineage j and all the other traits; finally, we fill the last missing block in the bottom right corner of $\Sigma_{\tau_{i+1}}$ with the block corresponding to the covariance matrix among the d traits in lineage j (i.e. the $d \times d$ diagonal block of $\Sigma_{\tau_{i+1}}^-$ starting from line $(j-1)d+1$).

In the case of evolution in multiple clades, $m_{\tau_{i+1}}$ and $\Sigma_{\tau_{i+1}}$ are constructed following a similar procedure, by augmenting $m_{\tau_{i+1}}^-$ and $\Sigma_{\tau_{i+1}}^-$ with copies of blocks corresponding to lineage j , inserted at the appropriate location. We illustrate this update step in Figure 2.

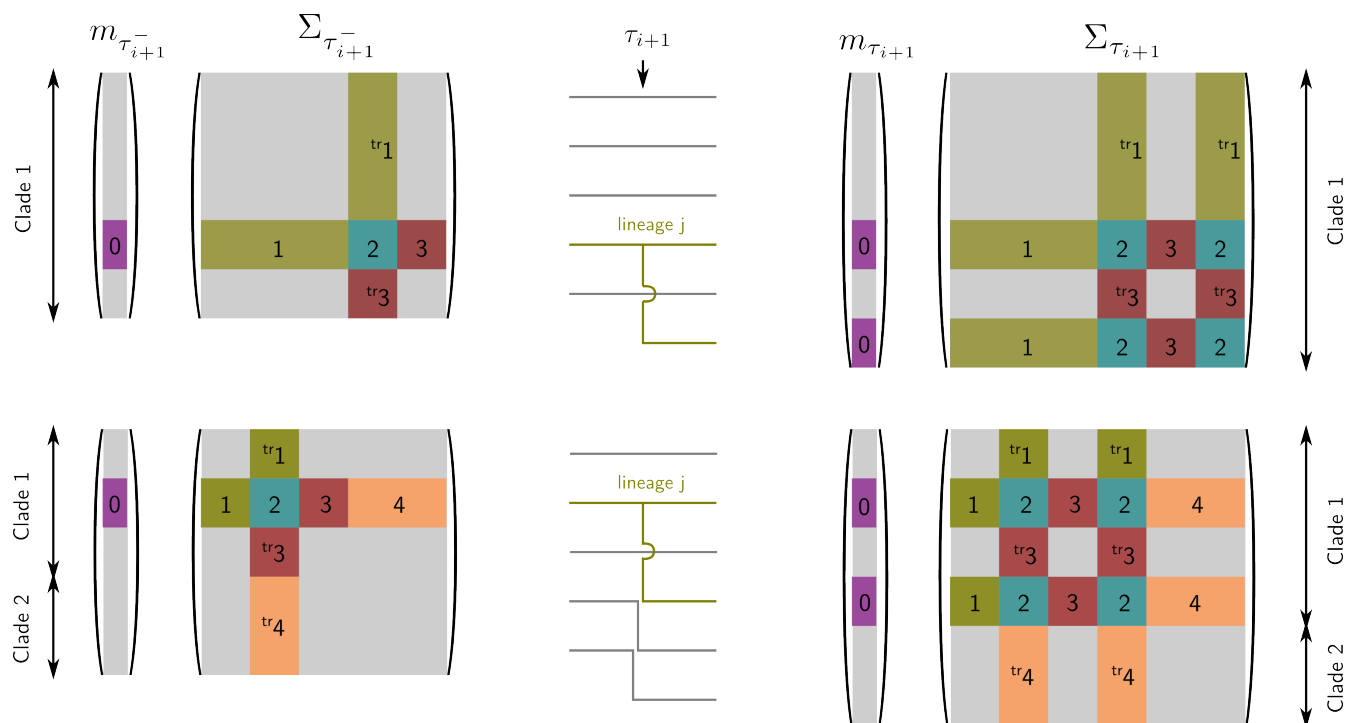


FIGURE 2: Toy example illustrating how to build the expectation vector and covariance matrix at branching times when there is one (top row) or two (bottom row) clades. Lineage j branches at time τ_{i+1} (middle). The vector $m_{\tau_{i+1}}$ and matrix $\Sigma_{\tau_{i+1}}$ (on the right) are constructed by augmenting $m_{\tau_{i+1}}^-$ and $\Sigma_{\tau_{i+1}}^-$ (on the left) with copies of blocks corresponding to lineage j (materialized by colors and numbers).

Tip trait distribution for particular models

Applying this general iterative procedure along a phylogenetic tree provides closed analytic tip distribution formulae for a wide variety of models. In Appendix B, we re-derive known tip distributions for models without lineage-lineage interaction, thus providing a unified review of mathematical results associated to these models. Tip distributions for classical univariate models (BM, ACDC, DD, OU) on ultrametric and non-ultrametric trees are summarized in Table S1. We confirm, as has been shown before (Uyeda et al. 2015), that the OU and AC models have identical tip distributions on ultrametric trees. We also re-derive results that can be found in Bartoszek et al. (2012) providing tip distributions for multivariate models.

Analytical formulae of tip distributions for models with lineage-lineage interactions have not yet been proposed. Drury et al. (2016) developed the inference tools that allow fitting the PM model, using the ODE system given in Equations (4a, 4b) (thereafter referred to as ‘ode’ method). Here, we develop the inference tools based on analytical reduction of Equations (3a, 3b), (thereafter referred to as ‘analytical’ method, see Appendix C.2), and compare the efficiencies of the two methods. Specifically, we simulated 10 pure-birth Yule trees with a per-lineage speciation rate of 1 per time unit, conditioned to having a given number of tips at present, using the ‘phytools’ R package (Revell 2012). We then computed the tip distribution corresponding to the PM model with parameters fixed at $(m_0, v_0, \theta, \psi, S, \sigma) = (0, 0, 1, 0.1, 1, 2)$ using both the analytical and the ode methods. The new analytical method is much more efficient than the previous ode method (Fig. 3). While we were previously limited to fitting the PM model to trees of less than 150 tips due to memory issues, the analytical methods allows fitting trees with up to 600 tips on a desktop computer.

Drury et al. (2016) also proposed an extension of the PM model accounting for the biogeographic history of lineages. In the case when each lineage is present in at most one location, the ‘analytical’ method can be extended, providing fast likelihood computation (see Appendix C.3). When there are lineages occurring in more than one location at the same time, we need to resolve numerically the ODE system in order to compute the likelihood of tip traits. While this is more time-consuming than finding a good ‘analytical’ reduction, the new

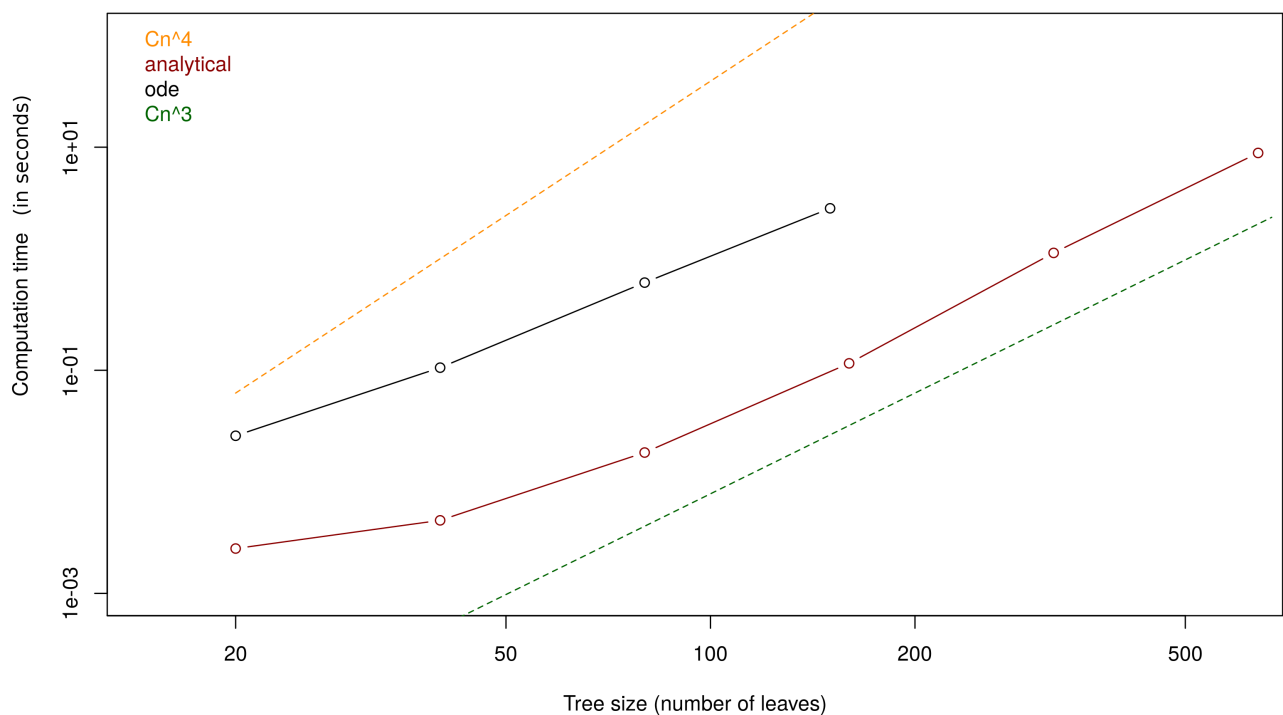


FIGURE 3: Time needed to compute the distribution of tip data following the PM model with parameters $(m_0, v_0, \theta, \psi, S, \sigma) = (0, 0, 1, 0.1, 1, 2)$. Trees are simulated under a pure-birth model conditioned on having a given number of leaves. Red curve : ‘analytical’ implementation; black curve: ‘ode’ implementation. Dashed yellow (resp. green) curve represent the slope of time increase as a power 4 (resp. 3) of the number of leaves.

333 implementation is more efficient than the one we previously proposed (Drury et al. 2016).

334 MODELING TRAIT EVOLUTION ON COEVOLVING CLADES

335 We illustrate how our framework can be used to study trait coevolution in scenarios of
 336 clade-clade interactions. We consider a simple model with two interacting clades (numbered 1
 337 and 2), in which a given trait in clade 1 coevolves with another given trait in clade 2. Following
 338 the approach introduced above, we define X_t the vector of trait values containing first the trait
 339 values for clade 1, and then the trait values for clade 2, and we write a stochastic differential

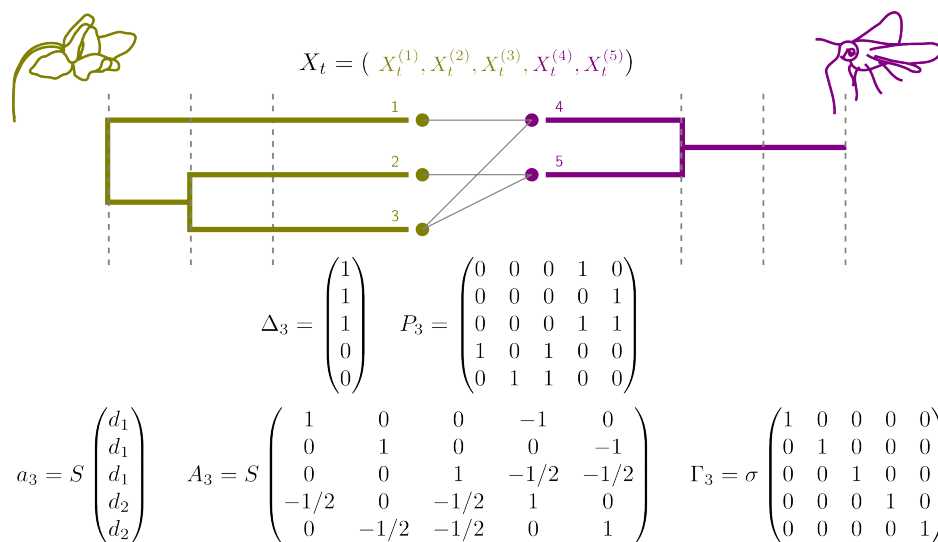


FIGURE 4: Hypothetical clade-clade coevolutionary scenario. Vertical dashed lines delimitate the successive epochs. The vector X_t contains the trait values on the third (last) epoch, P_3 is the matrix of network interactions, and a_3 , A_3 and Γ_3 together define trait evolution according to the clade-clade matching model defined in Equations (5) and (6).

equation specifying how trait value evolves along a single lineage k . In the spirit of the phenotype matching model, we propose here a formulation in which the trait of lineage k is attracted to (or repelled from, depending on the sign of S) the average trait value of the lineages it interacts with, plus (or minus) a shift:

$$dX_t^{(k)} = S \left(\delta_k d_1 + (1 - \delta_k) d_2 + \frac{1}{n_k} \sum_{l=1}^n p_{k,l} X_t^{(l)} - X_t^{(k)} \right) dt + \sigma dW_t^{(k)} \quad (5)$$

where S represents the attracting or repelling strength of species interactions on trait evolution, d_1 (resp. d_2) represents the shift for lineages from clade 1 (resp. clade 2), σ is the drift parameter, δ_k equals one if lineage k belongs to clade 1 and zero if it belongs to clade 2, $p_{k,l}$ equals one if lineages k and l interact and zero otherwise, $n_k = \sum_l p_{k,l}$ is the number of lineages interacting with lineage k , and n is the total number of lineages.

When S is positive, the trait value of lineage k is attracted to an optimal trait value given by the average trait value of the interacting species (plus a shift d_1 or d_2). An example of such a scenario of clade-clade matching mutualism is the coevolution between the length of floral tubes

and the length of butterfly proboscis in a plant-pollinator mutualistic network (illustrated in Fig. 4). In this example, we assume that the optimal length of a butterfly proboscis is the average length of the plant floral tubes it pollinates plus a shift d_2 , while the optimal length of a plant floral tube is the average proboscis length of its butterfly pollinators plus a shift d_1 . With $d_1 + d_2 = 0$, both traits can reach their optimal state, leading to a stable situation with butterfly proboscis a bit longer (if $d_1 > 0$) or shorter (if $d_1 < 0$) than plant floral tubes. With $d_1 + d_2 \neq 0$, traits cannot reach their optimal state, resulting in a runaway process where both traits tend to evolve toward an ever-moving optimum. For example, with positive d_1 and d_2 , the butterflies proboscis tends to get longer to better access the nectar, while the floral tube also tends to get longer to force the butterfly's body to touch the stamen. The parameters S and σ control respectively the strength of the interaction effect and the rate of stochastic phenotypic change. The bigger S , the closer the traits will track the optimum; the bigger σ , the bigger the fluctuations around this optimum.

When S is negative, the traits are repelled from the average trait value of the interacting species (plus a shift d_1 or d_2). This may capture natural situations of clade-clade competition driving trait displacement. Finally, some antagonistic interactions between traits could require to introduce two parameters $S_1 > 0$ and $S_2 < 0$ to capture match-vs-escape scenarios. For example, parasites might tend to develop cues matching those of their hosts while hosts develop cues to escape their parasites in a co-evolutionary arms race.

From Equation (5) we deduce the corresponding a , A and Γ on each epoch:

$$\begin{aligned} a &= S(\Delta d_1 + (V - \Delta)d_2) \\ A_{k,l} &= S(\mathbb{1}_{k=l} - \frac{p_{k,l}}{n_k}) \\ \Gamma &= \sigma I \end{aligned} \tag{6}$$

where Δ is the vector of elements δ_k (see Fig. 4 for an illustration). Matrix A is in general not symmetric anymore, as all species k do not have the same number n_k of species that they interact with.

As shown by Equation (6), entirely defining a model of clade-clade coevolution requires introducing a constant network of interaction on each epoch (the P matrix with elements $p_{k,l}$).

We can potentially re-define epochs to account for events of change in the interaction network in addition to speciation and extinction events, thus allowing interaction networks to evolve along branches. In practice, we typically (at best) have access to the current interaction network (Fig. 4), but not the ancestral networks. A solution to this would consist in treating the ancestral P matrices as parameters of the model, and searching the ancestral network(s) that maximize the fit to the data. Another approach would consist in reconstructing ancestral networks according to rules regarding the inheritance of interactions at speciation times. Developing these approaches is outside the scope of the current study, but we have shown how to compute tip trait distributions once they are developed.

We illustrate the computation of tip trait distributions for a model in which the ancestral networks are known: a generalist model where all species from clade 1 interact with all species from clade 2. We consider a ‘Generalist Matching Mutualism’ model of trait evolution (thereafter referred to as GMM, and illustrated in Fig. 5a), which is captured by Equation (5) with S positive and $p_{k,l} = 1$ for any two lineages k and l from different clades and $p_{k,l} = 0$ for any two lineages k and l from the same clade. Given that the model fits within our framework, we know that the trait distribution at the tips is Gaussian, and we can compute the expectation vector and covariance matrix corresponding to the model using Equations (3a, 3b), which we can reduce for this specific model in order to speed up the computation (Appendix C.4).

The tip distribution is relatively fast to compute (e.g. in the order of 0.8 seconds with two 100-tip trees on a desktop computer), such that fitting the model by maximum likelihood or in a Bayesian framework should not be problematic for trees with a few hundred tips. However, we do not aim here to carry an in-depth study of this particular model, nor to fit it to empirical data. Rather, we use our ability to rapidly compute tip trait distribution to get a first glimpse of the model behaviour under distinct sets of parameter values.

In Figure 5 (c,d,e,f), we plotted the distribution of the average \bar{X}^1 of trait values in clade 1 and the average \bar{X}^2 of trait values in clade 2 for traits evolving under the GMM model with four parameter sets chosen to lead to four distinct qualitative behaviours. From Equation (5), we can easily show that under GMM $\bar{X}^1 + \bar{X}^2$ is a drifted Brownian motion with drift term

404 $S(d_1 + d_2)$ and $\bar{X}^1 - \bar{X}^2$ is an OU process with optimum $(d_1 - d_2)/2$ and selection strength S .
 405 The shift parameters d_1 and d_2 thus directly determine the position of the optimum of the
 406 distribution. When $d_1 = -d_2$, the more likely values for the two average traits \bar{X}^1 and \bar{X}^2 are
 407 such that $\bar{X}^1 = \bar{X}^2 + d_1$ (see Fig. 5c). In contrast, when $d_1 \neq d_2$, the two communities have
 408 optimal trait values that are non-compatible, and the traits will tend to increase (resp.
 409 decrease) if $d_1 + d_2 > 0$ (resp. $d_1 + d_2 < 0$) (see Fig. 5d). The position of the peak in the tip
 410 distribution will thus depend also on the depth of the root and on the value of the parameter S .
 411 Moreover, the parameter S plays an important role in the hump thickness: the bigger S , the
 412 more constrained $\bar{X}^1 - \bar{X}^2$ around $(d_1 - d_2)/2$ (see Fig. 5f). The parameter σ also plays a role in
 413 the thickness of the hump, but in the orthogonal direction : increasing σ flattens the distribution
 414 by allowing different \bar{X}^1 and \bar{X}^2 values while retaining the constraint on $\bar{X}^1 - \bar{X}^2$ (see Fig. 5e).

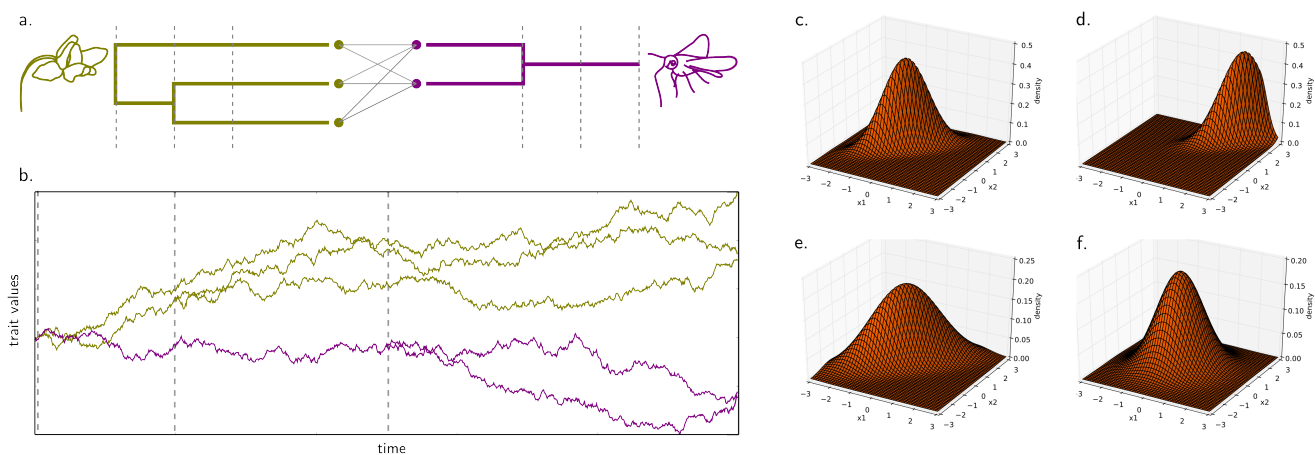


FIGURE 5: Trait evolution under the Generalist Matching Mutualism (GMM) model a) an illustrative generalist network of interactions between two clades. Vertical dashed lines delimitate the successive epochs. b) One realisation of trait evolution through time on the two phylogenies, with $(m_0, v_0) = (0, 0)$ and $(S, d_1, d_2, \sigma) = (1, -1, 1, 1)$, simulated using the Euler-Maruyama scheme (see Appendix D.1). cdef) Expected tip distribution for the average trait value in each clade, with parameter values $(S, d_1, d_2, \sigma) =$ c) $(2, -1, 1, 1)$, d) $(2, 0, 2, 1)$. e) $(2, -1, 1, 1.5)$, f) $(0.2, -1, 1, 1)$.

Implementation

Our framework is implemented in the R package ‘RPANDA’ (Morlon et al. 2015), including functions: 1) to compute tip distributions, 2) to simulate trait evolutionary trajectories using the Euler-Maruyama scheme (see Appendix D.1) and tip data by drawing from the expected tip distribution, and 3) to infer model parameters by maximum likelihood. In the most general user-defined use of our framework, the input is one or several phylogenetic tree(s) and the collection of (a_i, A_i, Γ_i) matrices on each epoch that define a specific model. In this case, the tip distribution is computed using the most general ‘ode’ method that solves numerically the ODEs. In addition, we implemented all models mentioned in Table 1 as well as GMM with the fastest described algorithm to compute their tip distribution. In Appendix E, we provide a tutorial explaining the structure of our code and illustrating how to use it.

DISCUSSION

We developed a modeling framework for traits coevolving in coevolving lineages and clades. We showed that under a wide variety of models where the evolution of a given trait on a given lineage is linearly related to its own value and the value of other traits on the same lineage, of the same trait on other lineages, and/or of other traits on other lineages, the expected tip trait distribution is Gaussian. We showed how to compute this tip distribution in general, as well as for specific models, including classical models of phenotypic evolution and new models of clade-clade coevolution.

All classical models of phenotypic evolution, such as univariate and multivariate BM, OU, ACDC and DD fit within our framework. They correspond to the situation where the evolution of traits on a given lineage is independent of trait values on other lineages. For these models, we already know that the tip trait distribution is Gaussian. However, finding the relevant computation of the expectation vectors and covariance matrices associated with each model in the dense literature of comparative phylogenetics can be overwhelming for neophytes. Our Appendix B unifies these computations under a common formalism, providing both the

expressions for the various existing models and their mathematical underpinning. This is done in the context of trees that are not necessarily ultrametric, meaning that all models can be applied to phylogenies including fossils. We hope that this Appendix can serve as a useful review for navigating phylogenetic approaches for understanding trait evolution.

The fact that the distribution of traits remains Gaussian when traits from different lineages coevolve is a new result. It is also a convenient result, because it means that computing the tip distribution only requires computing the expectation vector and covariance matrix associated with the different models. For example, we used this result in Drury et al. (2016) to compute tip trait distributions for the phenotype matching model (Nuismer and Harmon 2014) and fit it to comparative data by maximum likelihood. Here we vastly extend the set of potential coevolutionary models for which tip trait distributions can be computed and provide two general approaches for computing the expectation vector and covariance matrix. One of these two approaches (the ‘ode’ approach) consists in numerically integrating a set of ODEs. This is the approach that was used in Drury et al. (2016). The other approach (the ‘analytical’ approach) involves computing integrals and is more efficient when these integrals can be analytically reduced, which depends on the form of the model. Applying the ‘analytical’ approach to the PM model, we greatly improved its computational efficiency.

We provide a framework for computing tip trait distributions for a wide class of models accounting for within-clade and clade-clade interactions. We hope that this flexibility will foster the development and study of various models adapted to the specificities of particular scientific questions and biological systems. We did not study at length a particular coevolutionary model in this paper, but the PM model was thoroughly studied elsewhere (Drury et al. 2016). The Generalist Matching Mutualism model that we introduce here can be seen as a clade-clade analogue to the PM model. Both models are ‘generalist’ in the sense that all lineages are assumed to interact (within-clade in the case of PM and between clades in the case of GMM). This assumption can be relaxed by incorporating additional information. In our biogeographic models for example, lineages can only interact if they are sympatric (Drury et al. 2016). More generally, any information or hypothesis concerning the network of interactions between lineages

can be accounted for into the A matrices.

There are two main limitations to the modeling framework presented here. The first one is that trait evolution is always assumed to respond linearly to trait values in other lineages. Thus, non-linear effects such as a stronger selection for divergence when phenotypes are similar cannot be accounted for. The second one is the issue of model and parameter identifiability, in particular in the absence of fossils. A Gaussian distribution in \mathbb{R}^{nd} can potentially allow identifying several models and parameters, but there are distinct combinations for which a similar (or even identical) distribution is expected. For example, we already know that parameters of the OU model are non identifiable on phylogenies with only extant species (Ho and Ané 2014) and that OU and AC have identical tip distributions on ultrametric trees (Uyeda et al. 2015). Thus, while we wrote our framework in all generality, with a , A and Γ encompassing as many parameters as desired, and parameters that potentially vary between epochs, it is clear that simplifying assumptions need to be made in order to reduce this parameter space. Identifiability cannot always be checked analytically, as in the case of the OU and AC models. In addition, there can be differences between theoretical and *de facto* identifiability, with models that are identifiable in theory but are difficult to identify in practice. For example, we can show analytically that ψ and S from the PM model are theoretically identifiable, but in practice in most cases only $\psi + S$ can be estimated with precision. Also, *de facto* identifiability depends on the data available, such as the size and shape of a particular phylogeny, and whether it includes fossils or not (Slater et al. 2012). Furthermore, models taking into account interactions among lineages will have to assess the influence of extinct lineages in the past. This has been studied in Drury et al. (2016) for the PM model, by simulating trait evolution on trees including dead branches, before fitting the model on the reconstructed tree only. Our recommendation is to check identifiability on a case-by-case basis, by fitting the set of models under consideration to trait datasets simulated directly on the specific empirical phylogenies in hands. We provide the tools for rapidly simulating tip values under various models by sampling expected distributions.

One of the most challenging and exciting developments that we see ahead is to move from generalist models to models that account for specific interaction networks. We show in this paper

how to compute tip trait distributions for such models, assuming that the ancestral networks are known. While some fossil species interaction networks have been compiled (Dunne et al. 2008), such data is typically not available. Thus, if we are to really understand if and how species interactions affected long-term phenotypic evolution, we need to start developing models for reconstructing ancestral networks, analogous to the use of ancestral biogeographic models (see Ronquist and Sanmartín 2011 for a review) to incorporate biogeography into models of phenotypic evolution (see e.g. DD+GEO or MC+GEO, in Drury et al. 2016). Interestingly, our modeling framework could provide an approach to do so, informed by species phylogenies, the interaction network of present day species, and current species phenotypes. Indeed, rather than assuming that the ancestral networks are known, we could treat them as additional parameters to optimize upon, and find the ancestral networks that maximize the likelihood of the current data. Whether there will be enough information in the data to distinguish the probability of alternative ancestral networks remains to be tested, but the observed phylogenetic signal in empirical networks of interactions is encouraging (Ives and Godfray 2006; Rafferty and Ives 2013; Hadfield et al. 2014; Hayward and Horton 2014; Martín González et al. 2015). Our ability to distinguish the probability of alternative ancestral networks will be increased by proposing various scenarios regarding the inheritance of interactions at speciation times, such as scenarios in which daughter species interact with many or few of the species that interacted with their mother lineage. These upcoming developments can draw upon the existing literature on the cophylogeny problem (Conow et al. 2010), and will certainly have an important role to play in the on-going effort of understanding the evolution of species interaction networks (Loeuille and Loreau 2005; Martinez 2006; Nuismer et al. 2013).

Our framework for modeling trait evolution on phylogenetic trees includes most previously proposed models and can be used to develop a series of new models of within-clade and clade-clade coevolution. We hope that this will motivate new theoretical and empirical applications aimed at unravelling how species interactions evolve and influence phenotypic evolution over macro-evolutionary time-scales.

ACKNOWLEDGEMENTS

We are very grateful to Jonathan Drury, Florence Débarre and Luke Harmon, as well as members of our research teams for helpful discussions and comments on previous versions of the manuscript. MM acknowledges his PhD funding from the École Normale Supérieure; AL acknowledges funding from the Center for Interdisciplinary Research in Biology (Collège de France); HM acknowledges funding from ERC, grant ERC-CoG PANDA.

REFERENCES

- 530
- 531 Bartoszek, K., J. Pienaar, P. Mostad, S. Andersson, and T. F. Hansen. 2012. A phylogenetic
- 532 comparative method for studying multivariate adaptation. *J. Theor. Biol.* 314:204–215.
- 533 Beaulieu, J. M., D.-C. Jhvueng, C. Boettiger, and B. C. O’Meara. 2012. Modeling stabilizing
- 534 selection: expanding the ornstein–uhlenbeck model of adaptive evolution. *Evolution*
- 535 66:2369–2383.
- 536 Blomberg, S. P., T. Garland, and A. R. Ives. 2003. Testing for phylogenetic signal in comparative
- 537 data: behavioral traits are more labile. *Evolution* 57:717–745.
- 538 Brown, W. L. and E. O. Wilson. 1956. Character displacement. *Syst. Zool.* 5:49–64.
- 539 Butler, M. A. and A. A. King. 2004. Phylogenetic comparative analysis: a modeling approach for
- 540 adaptive evolution. *Am. Nat.* 164:683–695.
- 541 Clavel, J., G. Escarguel, and G. Merceron. 2015. mvmorph: an r package for fitting multivariate
- 542 evolutionary models to morphometric data. *Methods Ecol. Evol.* 6:1311–1319.
- 543 Clavel, J. and H. Morlon. 2016. Accelerated phenotypic evolution during cenozoic cooling across
- 544 birds and mammals. in prep .
- 545 Conow, C., D. Fielder, Y. Ovadia, and R. Libeskind-Hadas. 2010. Jane: a new tool for the
- 546 cophylogeny reconstruction problem. *Algorithms Mol. Biol.* 5:1–10.
- 547 Dale, J., C. J. Dey, K. Delhey, B. Kempnaers, and M. Valcu. 2015. The effects of life history
- 548 and sexual selection on male and female plumage colouration. *Nature* 527:367–370.
- 549 Dawkins, R. and J. R. Krebs. 1979. Arms races between and within species. *Proc. R. Soc. B*
- 550 205:489–511.
- 551 Drury, J., J. Clavel, M. Manceau, and H. Morlon. 2016. Estimating the effect of competition on
- 552 trait evolution using maximum likelihood inference. *Syst. Biol.* .

553 Dunne, J. A., R. J. Williams, N. D. Martinez, R. A. Wood, and D. H. Erwin. 2008. Compilation
554 and network analyses of cambrian food webs. *PLoS Biol.* 6:1–16.

555 Eastman, J. M., M. E. Alfaro, P. Joyce, A. L. Hipp, and L. J. Harmon. 2011. A novel
556 comparative method for identifying shifts in the rate of character evolution on trees. *Evolution*
557 65:3578–3589.

558 Ehrlich, P. R. and P. H. Raven. 1964. Butterflies and plants: a study in coevolution. *Evolution*
559 18:586–608.

560 Felsenstein, J. 1973. Maximum-likelihood estimation of evolutionary trees from continuous
561 characters. *Am. J. Hum. Genet.* 25:471–492.

562 Fenster, C. B., W. S. Armbruster, P. Wilson, M. R. Dudash, and J. D. Thomson. 2004.
563 Pollination syndromes and floral specialization. *Annu. Rev. Ecol. Evol. Syst.* 35:375–403.

564 Gardiner, C. W. et al. 1985. *Handbook of stochastic methods* vol. 4. Springer Berlin.

565 Hadfield, J. D., B. R. Krasnov, R. Poulin, and S. Nakagawa. 2014. A tale of two phylogenies:
566 comparative analyses of ecological interactions. *Am. Nat.* 183:174–187.

567 Hansen, T. F. 1997. Stabilizing selection and the comparative analysis of adaptation. *Evolution*
568 51:1341–1351.

569 Hansen, T. F. and E. P. Martins. 1996. Translating between microevolutionary process and
570 macroevolutionary patterns: the correlation structure of interspecific data. *Evolution*
571 50:1404–1417.

572 Hansen, T. F., J. Pienaar, and S. H. Orzack. 2008. A comparative method for studying
573 adaptation to a randomly evolving environment. *Evolution* 62:1965–1977.

574 Harmon, L. J., J. B. Losos, T. Jonathan Davies, R. G. Gillespie, J. L. Gittleman,
575 W. Bryan Jennings, K. H. Kozak, M. A. McPeck, F. Moreno-Roark, T. J. Near, et al. 2010.
576 Early bursts of body size and shape evolution are rare in comparative data. *Evolution*
577 64:2385–2396.

578 Harmon, L. J., J. T. Weir, C. D. Brock, R. E. Glor, and W. Challenger. 2008. Geiger:
579 investigating evolutionary radiations. *Bioinformatics* 24:129–131.

580 Hayward, J. and T. R. Horton. 2014. Phylogenetic trait conservation in the partner choice of a
581 group of ectomycorrhizal trees. *Mol. Ecol.* 23:4886–4898.

582 Ho, L. S. T. and C. Ané. 2014. Intrinsic inference difficulties for trait evolution with
583 ornstein-uhlenbeck models. *Methods Ecol. Evol.* 5:1133–1146.

584 Ives, A. R. and H. C. J. Godfray. 2006. Phylogenetic analysis of trophic associations. *Am. Nat.*
585 168:1–14.

586 Jhwueng, D.-C. and V. Maroulas. 2014. Phylogenetic ornstein–uhlenbeck regression curves. *Stat.*
587 *Probab. Lett.* 89:110–117.

588 Labra, A., J. Pienaar, and T. F. Hansen. 2009. Evolution of thermal physiology in liolaemus
589 lizards: adaptation, phylogenetic inertia, and niche tracking. *Am. Nat.* 174:204–220.

590 Liow, L. H., T. Reitan, and P. G. Harnik. 2015. Ecological interactions on macroevolutionary
591 time scales: clams and brachiopods are more than ships that pass in the night. *Ecol. Lett.*
592 18:1030–1039.

593 Loewille, N. and M. Loreau. 2005. Evolutionary emergence of size-structured food webs. *Proc.*
594 *Natl. Acad. Sci. U. S. A.* 102:5761–5766.

595 Mahler, D. L., L. J. Revell, R. E. Glor, and J. B. Losos. 2010. Ecological opportunity and the
596 rate of morphological evolution in the diversification of greater antillean anoles. *Evolution*
597 64:2731–2745.

598 Martín González, A. M., B. Dalsgaard, D. Nogués-Bravo, C. H. Graham, M. Schleuning, P. K.
599 Maruyama, S. Abrahamczyk, R. Alarcón, A. C. Araujo, F. P. Araújo, et al. 2015. The
600 macroecology of phylogenetically structured hummingbird–plant networks. *Glob. Ecol.*
601 *Biogeogr.* 24:1212–1224.

602 Martinez, N. D. 2006. Network evolution: exploring the change and adaptation of complex
603 ecological systems over deep time. Pages 287–302 in Ecological Networks: Linking Structure to
604 Dynamics in Food Webs (M. Pascual and J. Dunne, eds.). Oxford University Press.

605 Martins, E. P. 2004. Compare, version 4.6 b. computer programs for the statistical analysis of
606 comparative data.

607 Moen, D. and H. Morlon. 2014. From dinosaurs to modern bird diversity: extending the time
608 scale of adaptive radiation. *PLoS Biol.* 12:1–4.

609 Morlon, H., E. Lewitus, F. L. Condamine, M. Manceau, J. Clavel, and J. Drury. 2015. Rpanda:
610 an r package for macroevolutionary analyses on phylogenetic trees. *Methods Ecol. Evol.* .

611 Nuismer, S. L. and L. J. Harmon. 2014. Predicting rates of interspecific interaction from
612 phylogenetic trees. *Ecol. Lett.* 18:17–27.

613 Nuismer, S. L., P. Jordano, and J. Bascompte. 2013. Coevolution and the architecture of
614 mutualistic networks. *Evolution* 67:338–354.

615 O’Meara, B. C., C. Ané, M. J. Sanderson, and P. C. Wainwright. 2006. Testing for different rates
616 of continuous trait evolution using likelihood. *Evolution* 60:922–933.

617 Pennell, M. W. and L. J. Harmon. 2013. An integrative view of phylogenetic comparative
618 methods: Connections to population genetics, community ecology, and paleobiology. *Ann. NY*
619 *Acad. Sci.* 1289:90–105.

620 Quintero, I., P. Keil, W. Jetz, and F. W. Crawford. 2015. Historical biogeography using species
621 geographical ranges. *Syst. Biol.* 64:1059–1073.

622 Rafferty, N. E. and A. R. Ives. 2013. Phylogenetic trait-based analyses of ecological networks.
623 *Ecology* 94:2321–2333.

624 Reitan, T., T. Schweder, J. Henderiks, et al. 2012. Phenotypic evolution studied by layered
625 stochastic differential equations. *Ann. Appl. Stat.* 6:1531–1551.

626 Revell, L. J. 2012. phytools: an r package for phylogenetic comparative biology (and other
627 things). *Methods Ecol. Evol.* 3:217–223.

628 Revell, L. J. and D. C. Collar. 2009. Phylogenetic analysis of the evolutionary correlation using
629 likelihood. *Evolution* 63:1090–1100.

630 Ronquist, F. and I. Sanmartín. 2011. Phylogenetic methods in biogeography. *Annu. Rev. Ecol.*
631 *Evol. Syst.* 42:441–464.

632 Ruta, M., P. J. Wagner, and M. I. Coates. 2006. Evolutionary patterns in early tetrapods. i.
633 rapid initial diversification followed by decrease in rates of character change. *Proc. R. Soc. B*
634 273:2107–2111.

635 Simpson, G. G. 1944. *Tempo and mode in evolution*. Columbia University Press.

636 Slater, G. J. 2015. Iterative adaptive radiations of fossil canids show no evidence for
637 diversity-dependent trait evolution. *Proc. Natl. Acad. Sci. U. S. A.* 112:4897–4902.

638 Slater, G. J., L. J. Harmon, and M. E. Alfaro. 2012. Integrating fossils with molecular
639 phylogenies improves inference of trait evolution. *Evolution* 66:3931–3944.

640 Sletvold, N., J. Trunschke, M. Smit, J. Verbeek, and J. Ågren. 2016. Strong pollinator-mediated
641 selection for increased flower brightness and contrast in a deceptive orchid. *Evolution*
642 70:716–724.

643 Thomas, G. H. and R. P. Freckleton. 2012. Motmot: models of trait macroevolution on trees.
644 *Methods Ecol. Evol.* 3:145–151.

645 Thomas, G. H., R. P. Freckleton, and T. Székely. 2006. Comparative analyses of the influence of
646 developmental mode on phenotypic diversification rates in shorebirds. *Proc. R. Soc. B*
647 273:1619–1624.

648 Uyeda, J. C., D. S. Caetano, and M. W. Pennell. 2015. Comparative analysis of principal
649 components can be misleading. *Syst. Biol.* 64:677–689.

- 650 Van Valen, L. 1973. A new evolutionary law. *Evol. Theor.* 1:1–30.
- 651 Weiblen, G. D. 2004. Correlated evolution in fig pollination. *Syst. Biol.* 53:128–139.
- 652 Weir, J. T. and S. Mursleen. 2013. Diversity-dependent cladogenesis and trait evolution in the
653 adaptive radiation of the auks (aves: Alcidae). *Evolution* 67:403–416.

LIST OF FIGURES

- 1 Formalism used throughout the paper, to model the evolution of one trait on a non-ultrametric tree. Epochs are separated with vertical dashed lines. 8
- 2 Toy example illustrating how to build the expectation vector and covariance matrix at branching times when there is one (top row) or two (bottom row) clades. Lineage j branches at time τ_{i+1} (middle). The vector $m_{\tau_{i+1}}$ and matrix $\Sigma_{\tau_{i+1}}$ (on the right) are constructed by augmenting m_{τ_i} and Σ_{τ_i} (on the left) with copies of blocks corresponding to lineage j (materialized by colors and numbers). 14
- 3 Time needed to compute the distribution of tip data following the PM model with parameters $(m_0, v_0, \theta, \psi, S, \sigma) = (0, 0, 1, 0.1, 1, 2)$. Trees are simulated under a pure-birth model conditioned on having a given number of leaves. Red curve : ‘analytical’ implementation; black curve: ‘ode’ implementation. Dashed yellow (resp. green) curve represent the slope of time increase as a power 4 (resp. 3) of the number of leaves. 16
- 4 Hypothetical clade-clade coevolutionary scenario. Vertical dashed lines delimitate the successive epochs. The vector X_t contains the trait values on the third (last) epoch, P_3 is the matrix of network interactions, and a_3 , A_3 and Γ_3 together define trait evolution according to the clade-clade matching model defined in Equations (5) and (6). 17
- 5 Trait evolution under the Generalist Matching Mutualism (GMM) model a) an illustrative generalist network of interactions between two clades. Vertical dashed lines delimitate the successive epochs. b) One realisation of trait evolution through time on the two phylogenies, with $(m_0, v_0) = (0, 0)$ and $(S, d_1, d_2, \sigma) = (1, -1, 1, 1)$, simulated using the Euler-Maruyama scheme (see Appendix D.1). cdef) Expected tip distribution for the average trait value in each clade, with parameter values $(S, d_1, d_2, \sigma) =$ c) $(2, -1, 1, 1)$, d) $(2, 0, 2, 1)$. e) $(2, -1, 1, 1.5)$, f) $(0.2, -1, 1, 1)$ 20

Appendix

A unifying comparative phylogenetic framework including traits coevolving across interacting lineages

MARC MANCEAU^{1,3,4}, AMAURY LAMBERT^{2,3}, HÉLÈNE MORLON⁴

¹*Muséum National d'Histoire Naturelle, 75005 Paris, France;*

²*Laboratoire Probabilités et Modèles Aléatoires, UPMC Univ Paris 06, 75005 Paris, France;*

³*Center for Interdisciplinary Research in Biology, Collège de France, CNRS UMR 7241, 75005 Paris, France;*

⁴*Institut de Biologie, École Normale Supérieure, CNRS UMR 8197, 75005 Paris, France*

CONTENTS

A	Derivation of the distribution in a general setting	3
A.1	The distribution of trait values is Gaussian	3
A.2	Integrating the evolution of the distribution on each epoch	3
A.3	Evolution of the distribution through ODE resolution	4
B	Distribution for some models without interactions between lineages	7
B.1	Distribution of classic univariate models	7
B.2	Distribution of classic multivariate models	14
C	Distribution for some models with interactions between lineages	19
C.1	Distribution with a constant, A symmetric, and $\Gamma = \sigma I$	19
C.2	The phenotype matching (PM) model	19
C.3	The phenotype matching (PM) model with biogeography	22
C.4	The generalist matching mutualism (GMM) model	25
D	Simulation and Inference	31
D.1	Numerical methods for simulating data	31
D.2	Parameter inference	32

26	E Tutorial : using the RPANDA codes to study trait coevolution	35
27	E.1 The 'PhenotypicModel' class	36
28	E.2 Methods associated to the 'PhenotypicModel' class	41
29	E.3 Toward an in-depth understanding of the code structure	50

A DERIVATION OF THE DISTRIBUTION IN A GENERAL SETTING

A.1 *The distribution of trait values is Gaussian*

Recall that a vector is Gaussian if all linear combination of its components follows a normal distribution. We will thus show by induction that all linear combinations of the traits follow a normal distribution.

The process of trait evolution starts either at the stem root with a vector of size d defined by the initial conditions $X_{\tau_0} = {}^{tr}(X_0^1, \dots, X_0^d)$, or at the crown root with a vector of size $2d$ defined by the initial conditions : $X_{\tau_0} = {}^{tr}(X_0^1, \dots, X_0^d, X_0^1, \dots, X_0^d)$, or at any other step, provided the initial conditions are Gaussian by assumption.

Now, assume that X_{τ_i} is a Gaussian vector.

Then, $\forall t \in (\tau_i, \tau_{i+1})$, after integration we have the following closed expression for the value of the process X_t .

$$X_t = e^{-tA_i} \left(e^{\tau_i A_i} X_{\tau_i} + \int_{\tau_i}^t e^{sA_i} a_i(s) ds + \int_{\tau_i}^t e^{sA_i} \Gamma_i(s) dW_s \right) \quad (\text{S1})$$

Moreover, we have, for any deterministic function Φ (Gardiner et al. 1985),

$$\int_{t_n}^t \Phi_s dW_s \sim \mathcal{N} \left(0, \int_{t_n}^t \Phi_s {}^{tr} \Phi_s ds \right)$$

Hence, X_t is a linear combination of Gaussian vectors, which makes it a Gaussian vector.

Last, suppose that at time τ_{i+1} , the j th branch splits, in which case the vector grows. All linear combinations of the components of X_t at time τ_{i+1}^- have a normal distribution. And the d additional components added at time τ_{i+1} belong to the components at time τ_{i+1}^- . It follows that all linear combinations of the new vector still have a normal distribution.

A.2 *Integrating the evolution of the distribution on each epoch*

Still assuming that we know the (Gaussian) distribution of X_{τ_i} at the beginning of an epoch

(τ_i, τ_{i+1}) , a few more lines allow us to provide a closed formula for the distribution of X_t at all

time $t \in (\tau_i, \tau_{i+1})$. Indeed, using Equation (S1), and the fact that, if X and Y are two independent Gaussian vectors with expectation vectors respectively m_X and m_Y and covariance matrices respectively Σ_X and Σ_Y , then :

$$DX + d \sim \mathcal{N}(Dm_X + d, D\Sigma_X^{tr}D)$$

$$X + Y \sim \mathcal{N}(m_X + m_Y, \Sigma_X + \Sigma_Y)$$

It thus follows that, $\forall t \in [\tau_i, \tau_{i+1}]$,

$$m_t = e^{(\tau_i - t)A_i}m_{\tau_i} + \int_{\tau_i}^t e^{(s-t)A_i}a_i(s)ds \quad (3a)$$

$$\Sigma_t = (e^{(\tau_i - t)A_i})\Sigma_{\tau_i}^{tr}(e^{(\tau_i - t)A_i}) + \int_{\tau_i}^t (e^{(s-t)A_i}\Gamma_i(s))^{tr}(e^{(s-t)A_i}\Gamma_i(s))ds \quad (3b)$$

Applying these equations for $t = \tau_{i+1}$ thus gives the distribution of the trait vector at time τ_{i+1} , which is the result stated in Equations (3a, 3b) in the main text.

Remark that, unless one of the very first branches immediately dies at the beginning of the process at a fixed initial condition, the density of the tip distribution has support in \mathbb{R}^{nd} . One can check that Σ_t stays positive definite (implying that $\det \Sigma_t \neq 0$), even when some Γ_i are not positive definite (except the first one).

A.3 Evolution of the distribution through ODE resolution

The expectation and covariance formulae provided in Equations (3a, 3b) require to deal with an integral which is not always straightforward to compute. Alternatively, one can prefer to take the derivative of this expression, get a set of ODEs verified by the expectation and covariance elements through each epoch, and subsequently integrate the ODE system. We show now another way to derive this set of ODEs.

First, we write the stochastic differential equation on any epoch (τ_i, τ_{i+1}) and for each trait k , which is given in the most general setting by :

$$dX_t^{(k)} = \left(a_i^{(k)}(t) - \sum_{m=1}^{n_t d} A_i^{(k,m)} X_t^{(m)} \right) dt + \sum_{m=1}^{n_t d} \Gamma_i^{(k,m)}(t) dW_t^{(m)}$$

Itô's formula (Gardiner et al. 1985) then gives us :

$$\begin{aligned} d \left(X_t^{(k)} X_t^{(l)} \right) &= X_t^{(k)} dX_t^{(l)} + X_t^{(l)} dX_t^{(k)} + d \langle X_t^{(k)}, X_t^{(l)} \rangle \\ &= \left(a_i^{(l)}(t) X_t^{(k)} - \sum_{m=1}^{n_t d} A_i^{(l,m)} X_t^{(m)} X_t^{(k)} \right) dt + \sum_{m=1}^{n_t d} \Gamma_i^{(l,m)}(t) X_t^{(k)} dW_t^{(m)} \\ &+ \left(a_i^{(k)}(t) X_t^{(l)} - \sum_{m=1}^{n_t d} A_i^{(k,m)} X_t^{(m)} X_t^{(l)} \right) dt + \sum_{m=1}^{n_t d} \Gamma_i^{(k,m)}(t) X_t^{(l)} dW_t^{(m)} \\ &+ \sum_{m=1}^{n_t d} \Gamma_i^{(l,m)}(t) \Gamma_i^{(k,m)}(t) dt \end{aligned}$$

Taking the expectation, it follows that

$$\begin{aligned} \frac{d}{dt} \mathbb{E} \left(X_t^{(k)} X_t^{(l)} \right) &= a_i^{(l)}(t) \mathbb{E} \left(X_t^{(k)} \right) + a_i^{(k)}(t) \mathbb{E} \left(X_t^{(l)} \right) \\ &- \sum_{m=1}^{n_t d} A_i^{(l,m)} \mathbb{E} \left(X_t^{(m)} X_t^{(k)} \right) - \sum_{m=1}^{n_t d} A_i^{(k,m)} \mathbb{E} \left(X_t^{(m)} X_t^{(l)} \right) \\ &+ \sum_{m=1}^{n_t d} \Gamma_i^{(l,m)}(t) \Gamma_i^{(k,m)}(t) \end{aligned}$$

In the same fashion, we get

$$\frac{d}{dt} \mathbb{E} \left(X_t^{(k)} \right) = a_i^{(k)}(t) - \sum_{m=1}^{n_t d} A_i^{(k,m)} \mathbb{E} \left(X_t^{(m)} \right) \quad (4a)$$

This leads to

$$\begin{aligned} \frac{d}{dt} \left(\mathbb{E} \left(X_t^{(k)} \right) \mathbb{E} \left(X_t^{(l)} \right) \right) &= \mathbb{E} \left(X_t^{(l)} \right) \frac{d}{dt} \mathbb{E} \left(X_t^{(k)} \right) + \mathbb{E} \left(X_t^{(k)} \right) \frac{d}{dt} \mathbb{E} \left(X_t^{(l)} \right) \\ &= a_i^{(k)}(t) \mathbb{E} \left(X_t^{(l)} \right) - \sum_{m=1}^{n_t d} A_i^{(k,m)} \mathbb{E} \left(X_t^{(m)} \right) \mathbb{E} \left(X_t^{(l)} \right) \\ &+ a_i^{(l)}(t) \mathbb{E} \left(X_t^{(k)} \right) - \sum_{m=1}^{n_t d} A_i^{(l,m)} \mathbb{E} \left(X_t^{(m)} \right) \mathbb{E} \left(X_t^{(k)} \right) \end{aligned}$$

Putting together these different parts gives us the ODE satisfied by all covariances :

$$\begin{aligned} \frac{d}{dt} \text{Cov} \left(X_t^{(k)}, X_t^{(l)} \right) &= \frac{d}{dt} \left(\mathbb{E} \left(X_t^{(k)} X_t^{(l)} \right) - \mathbb{E}(X_t^{(k)}) \mathbb{E}(X_t^{(l)}) \right) \\ &= - \sum_{m=1}^{n_t d} \left[A_i^{(k,m)} \text{Cov} \left(X_t^{(m)}, X_t^{(l)} \right) + A_i^{(l,m)} \text{Cov} \left(X_t^{(m)}, X_t^{(k)} \right) - \Gamma_i^{(l,m)}(t) \Gamma_i^{(k,m)}(t) \right] \end{aligned} \quad (4b)$$

74 Note that in a vectorial formalism with the expectation vector m and covariance matrix
75 Σ , these sets of ODEs can be written equivalently as follows

$$\frac{dm_t}{dt} = a_i(t) - A_i m_t \quad (S2)$$

$$\frac{d\Sigma_t}{dt} = -A_i \Sigma_t - {}^{tr}\Sigma_t {}^{tr}A_i + \Gamma_i {}^{tr}\Gamma_i \quad (S3)$$

B DISTRIBUTION FOR SOME MODELS WITHOUT INTERACTIONS BETWEEN LINEAGES

B.1 Distribution of classic univariate models

We present in this section how previously known results of analytic tip distribution of univariate models fit in, and can be rediscovered with, our framework. Results are summarized in Table S1.

The scheme is identical for each model :

1. Reduce Equations (3a, 3b) or (4a, 4b) according to the model.
2. Look for an analytical solution at any time τ_i , by calculating manually the expectations and covariances at $\tau_1, \tau_2, \tau_3, \dots$
3. Prove by induction that the analytical solution holds at any time τ_i .

We call $t_{k,l}$ the time of the most recent common ancestor to lineages k and l , and $t_{k,k}$ the death time of lineage k , equal to T if it survives until present (see Fig. S1). We further note $1_{k \text{ alive}}(t)$ the quantity that equals one if lineage k is alive at time t and zero otherwise, and $1_{k=l}$ that equals one if $k = l$ and zero otherwise. Last, $t_1 \wedge t_2$ stands for the minimum of the two values t_1 and t_2 .

The unity vector (vector full of 1) is denoted by V , I refers to the identity matrix (diagonal matrix with diagonal values equal to 1), and U refers to the unity matrix (matrix full of 1). Their size is the same as the size of the vector of traits X_t considered. Considering non-ultrametric trees including fossils amounts to replacing vector V and matrices I and U by their homologs V_{alive} , I_{alive} and U_{alive} , where the subscript specifies that the vector and matrices have 0 on lines and columns corresponding to lineages that are extinct in the given epoch.

Brownian Motion (BM).—

We show how to get the well-known expression of the distribution of a trait evolving under BM, on non-necessarily ultrametric trees. We take $a = bV_{\text{alive}}$, $A = 0$ and $\Gamma = \sigma I_{\text{alive}}$, i.e. the process follows the equation :

Code	m_0	Σ_0	$(m_T)^{(k)}$	$(\Sigma_T)^{(k,l)}$
BM	m_0	v_0	$m_0 + bt_{k,k}$	$v_0 + \sigma^2 t_{k,l}$
OU	θ	0	θ	$\frac{\sigma^2}{2\psi} e^{-\psi(t_{k,k}+t_{l,l}-2t_{k,l})} (1 - e^{-2\psi t_{k,l}})$
OU	θ	$\frac{\sigma^2}{2\psi}$	θ	$\frac{\sigma^2}{2\psi} e^{-\psi(t_{k,k}+t_{l,l}-2t_{k,l})}$
ACDC	m_0	v_0	m_0	$v_0 + \frac{\sigma_0^2}{2r} (e^{2rn_{\tau_j}} - 1)$
DD	m_0	v_0	m_0	$v_0 + \sigma_0^2 \sum_{j=0}^{N-1} e^{2rn_{\tau_j}} (\tau_{j+1} - \tau_j) 1_{t_{k,l} > \tau_j}$

TABLE S1: Analytic tip distribution for models without interactions between traits or lineages. We recall that $t_{k,l}$ is the absolute time of the most recent common ancestor to lineages k and l , and $t_{k,k}$ is the death time of lineage k , equal to T if it survives until present.

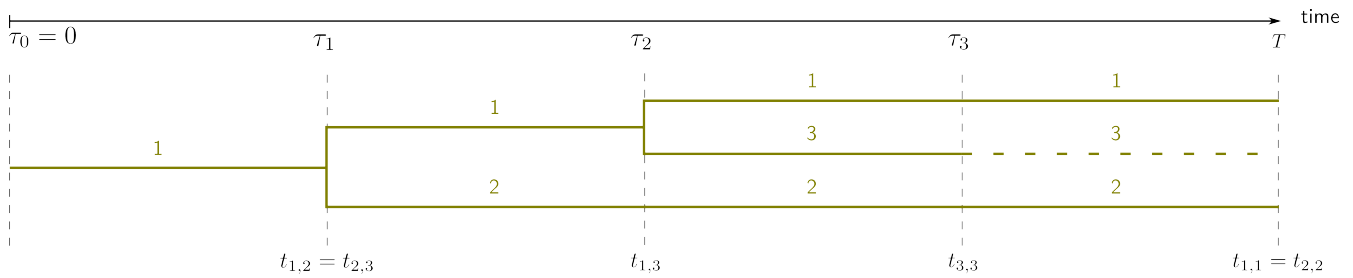


FIGURE S1: Formalism used in analytic formulae presented in Table S1.

$$dX_t = bV_{\text{alive}}dt + \sigma I_{\text{alive}}dW_t$$

Equations (3a) and (3b) lead to the following recurrence formulae driving the law of X_t on each epoch $[\tau_i, \tau_{i+1})$:

$$\begin{aligned}\mathbb{E}(X_t) &= \mathbb{E}(X_{\tau_i}) + b(t - \tau_i)V_{\text{alive}} \\ \text{Var}(X_t) &= \text{Var}(X_{\tau_i}) + \sigma^2(t - \tau_i)I_{\text{alive}}\end{aligned}$$

Alternatively, Equations (4a) and (4b) lead to the following recurrence formulae driving the law of X_t on each epoch $[\tau_i, \tau_{i+1})$:

$$\begin{aligned}\frac{d}{dt}\mathbb{E}(X_t^{(k)}) &= b1_{k \text{ alive}}(t) \\ \frac{d}{dt}\text{Cov}(X_t^{(k)}, X_t^{(l)}) &= \sigma^2 1_{k=l} 1_{k \text{ alive}}(t)\end{aligned}$$

We can show by induction on i that for any i the expectation and covariance matrix at time τ_i are such that, for any (k, l) :

$$\mathbb{E}(X_{\tau_i}^{(k)}) = \mathbb{E}(X_0) + b(t_{k,k} \wedge \tau_i) \quad (\text{S4})$$

$$\text{Cov}(X_{\tau_i}^{(k)}, X_{\tau_i}^{(l)}) = \text{Var}(X_0) + \sigma^2(t_{k,l} \wedge \tau_i) \quad (\text{S5})$$

Indeed, we verify Equations (S4, S5) at step $i = 1$.

Now, suppose Equations (S4, S5) hold at step n . Using either Equations (3a, 3b) or (4a, 4b), we get :

$$\begin{aligned}\mathbb{E}(X_{\tau_{n+1}}^{(k)}) &= \mathbb{E}(X_0) + b(t_{k,k} \wedge \tau_{n+1}) \\ \text{Cov}(X_{\tau_{n+1}}^{(k)}, X_{\tau_{n+1}}^{(l)}) &= \text{Var}(X_0) + \sigma^2(t_{k,l} \wedge \tau_{n+1})\end{aligned}$$

If τ_{n+1} is a death time of a lineage, Equations (S4, S5) are verified at step $n + 1$.

If τ_{n+1} is a branching time, we verify that the new lineage inherits the expectation and covariances of its mother, as well as the same coalescence times with other lineages. It also follows that Equations (S4, S5) are verified at step $n + 1$.

Finally, by induction, we get the tip distribution :

$$\begin{aligned}\mathbb{E}(X_T^{(k)}) &= \mathbb{E}(X_0) + bt_{k,k} \\ \text{Cov}(X_T^{(k)}, X_T^{(l)}) &= \text{Var}(X_0) + \sigma^2 t_{k,l}\end{aligned}$$

Ornstein-Uhlenbeck (OU).—

We can get another well-known distribution for a trait evolving under an

Ornstein-Uhlenbeck process on a tree. We take $a = \psi\theta V_{\text{alive}}$, $A = \psi I_{\text{alive}}$ and $\Gamma = \sigma I_{\text{alive}}$, i.e. the process follows the equation :

$$dX_t = (\psi\theta V_{\text{alive}} - \psi I_{\text{alive}} X_t)dt + \sigma I_{\text{alive}} dW_t$$

Expressions (3a) and (3b) simplify into the following recurrence formulae :

$$\begin{aligned}\mathbb{E}(X_t) &= e^{-\psi(t-\tau_i)I_{\text{alive}}} (\mathbb{E}(X_{\tau_i}) - \theta V_{\text{alive}}) + \theta V_{\text{alive}} \\ \text{Var}(X_t) &= e^{-2\psi(t-\tau_i)I_{\text{alive}}} \left(\text{Var}(X_{\tau_i}) - \frac{\sigma^2}{2\psi} I_{\text{alive}} \right) + \frac{\sigma^2}{2\psi} I_{\text{alive}}\end{aligned}$$

Alternatively, here again, one can prefer to apply Equations (4a) and (4b) :

$$\begin{aligned}\frac{d}{dt}\mathbb{E}(X_t^{(k)}) &= \psi 1_{k\text{ alive}}(t) \left(\theta - \mathbb{E}(X_t^{(k)}) \right) \\ \frac{d}{dt}\text{Cov}(X_t^{(k)}, X_t^{(l)}) &= -\psi(1_{k\text{ alive}}(t) + 1_{l\text{ alive}}(t))\text{Cov}(X_t^{(k)}, X_t^{(l)}) + \sigma^2 1_{k=l}\end{aligned}$$

We can show by induction that for any epoch i , the expectation and covariance matrix at time τ_i are such that, for all (k, l) :

$$\mathbb{E}(X_{\tau_i}^{(k)}) = \theta + e^{-\psi(t_{k,k}\wedge\tau_i)} (\mathbb{E}(X_0) - \theta) \quad (\text{S6})$$

$$\text{Cov}(X_{\tau_i}^{(k)}, X_{\tau_i}^{(l)}) = e^{-\psi(t_{k,k}\wedge\tau_i + t_{l,l}\wedge\tau_i - 2(t_{k,l}\wedge\tau_i))} \left[\frac{\sigma^2}{2\psi} + e^{-2\psi(t_{k,l}\wedge\tau_i)} \left(\text{Var}(X_0) - \frac{\sigma^2}{2\psi} \right) \right] \quad (\text{S7})$$

Indeed, we verify Equations (S6, S7) at step $i = 0$.

Now, suppose Equations (S6, S7) hold at step n . Using either Equations (3a, 3b) or (4a, 4b), we get :

$$\begin{aligned}\mathbb{E}(X_{\tau_{n+1}}^{(k)}) &= \theta + e^{-\psi(t_{k,k}\wedge\tau_{n+1})} (\mathbb{E}(X_0) - \theta) \\ \text{Cov}(X_{\tau_{n+1}}^{(k)}, X_{\tau_{n+1}}^{(l)}) &= e^{-\psi(t_{k,k}\wedge\tau_{n+1} + t_{l,l}\wedge\tau_{n+1} - 2(t_{k,l}\wedge\tau_{n+1}))} \left[\frac{\sigma^2}{2\psi} + e^{-2\psi(t_{k,l}\wedge\tau_{n+1})} \left(\text{Var}(X_0) - \frac{\sigma^2}{2\psi} \right) \right]\end{aligned}$$

If τ_{n+1} is a death time of a lineage, Equations (S6, S7) are verified at step $n + 1$.

If τ_{n+1} is a branching time, we verify that the new lineage inherits the expectation and

covariances of its mother, as well as the same coalescence times with other lineages. It also

follows that Equations (S6, S7) are verified at step $n + 1$.

Finally, by induction, we get the tip distribution :

$$\begin{aligned}\mathbb{E}(X_T^{(k)}) &= \theta + e^{-\psi t_{k,k}} (\mathbb{E}(X_0) - \theta) \\ \text{Cov}(X_T^{(k)}, X_T^{(l)}) &= e^{-\psi(t_{k,k}+t_{l,l}-2t_{k,l})} \left[\frac{\sigma^2}{2\psi} + e^{-2\psi t_{k,l}} \left(\text{Var}(X_0) - \frac{\sigma^2}{2\psi} \right) \right]\end{aligned}$$

Two classes of initial distributions are typically considered in the literature :

1. If we consider a process starting at $X_0 = \theta$ (i.e. with $\mathbb{E}(X_0) = \theta$ and $\text{Var}(X_0) = 0$), we get the following expectation vector m_T and covariance matrix Σ_T at the tips :

$$\begin{aligned}m_T &= {}^{tr}(\theta, \theta, \dots, \theta) \quad \text{and} \quad \Sigma_T = \frac{\sigma^2}{2\psi} \Upsilon_1 \\ \text{where } \Upsilon_1 &= [e^{-\psi(t_{k,k}+t_{l,l}-2t_{k,l})} (1 - e^{-2\psi t_{k,l}})]_{1 \leq k, l \leq K}\end{aligned}$$

2. When $\psi > 0$, if we consider a process starting under its stationary distribution (i.e. $\mathbb{E}(X_0) = \theta$ and $\text{Var}(X_0) = \frac{\sigma^2}{2\psi}$), it simplifies into the following expectation vector and covariance matrix :

$$\begin{aligned}m_T &= {}^{tr}(\theta, \theta, \dots, \theta) \quad \text{and} \quad \Sigma_T = \frac{\sigma^2}{2\psi} \Upsilon_2 \\ \text{where } \Upsilon_2 &= [e^{-\psi(t_{k,k}+t_{l,l}-2t_{k,l})}]_{1 \leq k, l \leq K}\end{aligned}$$

ACDC (accelerating or decelerating rate).—

In the ACDC process, the rate of phenotypic evolution varies exponentially through time, with $a = 0$, $A = 0$ and $\Gamma = \sigma_0 e^{rt} I_{\text{alive}}$ (here, $r > 0$). The process follows the equation :

$$dX_t = \sigma_0 e^{rt} I_{\text{alive}} dW_t$$

Here again, we can simplify Equations (3a, 3b) or (4a, 4b). With Equations (3a, 3b), we get the following recurrence formulae driving the law of X_t on each epoch (τ_i, τ_{i+1}) :

$$\begin{aligned}\mathbb{E}(X_t) &= \mathbb{E}(X_{\tau_i}) \\ \text{Var}(X_t) &= \text{Var}(X_{\tau_i}) + \frac{\sigma_0^2}{2r} (e^{2rt} - e^{2r\tau_i}) I_{\text{alive}} dt\end{aligned}$$

We can show by induction that for any i , the expectation and covariance matrix at time τ_i are such that, for any (k, l) :

$$\mathbb{E}(X_{\tau_i}^{(k)}) = \mathbb{E}(X_0) \quad (\text{S8})$$

$$\text{Cov}(X_{\tau_i}^{(k)}, X_{\tau_i}^{(l)}) = \text{Var}(X_0) + \frac{\sigma_0^2}{2r} (e^{2r(t_{k,l} \wedge \tau_i)} - 1) \quad (\text{S9})$$

Indeed, we verify Equations (S8, S9) at step $i = 0$.

Now, suppose Equations (S8, S9) hold at step n . Using either Equations (3a, 3b) or (4a, 4b), we get :

$$\begin{aligned} \mathbb{E}(X_{\tau_{n+1}}^{(k)}) &= \mathbb{E}(X_0) \\ \text{Cov}(X_{\tau_{n+1}}^{(k)}, X_{\tau_{n+1}}^{(l)}) &= \text{Var}(X_0) + \frac{\sigma_0^2}{2r} (e^{2r(t_{k,l} \wedge \tau_{n+1})} - 1) \end{aligned}$$

If τ_{n+1} is a death time of a lineage, Equations (S8, S9) are verified at step $n + 1$.

If τ_{n+1} is a branching time, we verify that the new lineage inherits the expectation and covariances of its mother, as well as the same coalescence times with other lineages. It also follows that Equations (S8, S9) are verified at step $n + 1$.

Finally, by induction, we get the tip distribution :

$$\begin{aligned} \mathbb{E}(X_T^{(k)}) &= \mathbb{E}(X_0) \\ \text{Cov}(X_T^{(k)}, X_T^{(l)}) &= \text{Var}(X_0) + \frac{\sigma_0^2}{2r} (e^{2rt_{k,l}} - 1) \end{aligned}$$

ACDC and OU processes lead to the same present-time distributions on ultrametric trees.—

This has been shown previously in Uyeda et al. 2015. More precisely, OU is equivalent to a model with accelerating rates at present, and only on ultrametric phylogenies.

Looking at expressions of expectations and covariance matrices under ACDC and OU with initial conditions $X_0 = \theta$, we see that we can choose parameters such that we get the exact same distribution. First take $\mathbb{E}(X_0) = \theta$: the two expectation vectors are identical. Moreover, we can choose parameters such that the covariance matrices are equal :

$$\begin{aligned} \frac{\sigma^2}{2\psi} e^{-2\psi(T-t_{k,l})} (1 - e^{-2\psi t_{k,l}}) &= \frac{\sigma_0^2}{2r} (e^{2rt_{k,l}} - 1) \\ \iff \frac{\sigma^2}{2\psi} e^{-2\psi T} (e^{2\psi t_{k,l}} - 1) &= \frac{\sigma_0^2}{2r} (e^{2rt_{k,l}} - 1) \\ \iff r = \psi \quad \text{and} \quad \sigma_0^2 &= \sigma^2 e^{-2\psi T} \end{aligned}$$

Note that this no longer holds on non-ultrametric trees, neither with different initial conditions on the OU.

Diversity-Dependent (DD).—

In the DD process, the rate of phenotypic evolution is fixed at the base of the tree and varies exponentially with the number of lineages in the reconstructed phylogeny, with $a = 0$, $A = 0$ and $B(t) = \sigma_0 e^{rn_t} I_{\text{alive}}$. The process follows the equation :

$$dX_t = \sigma_0 e^{rn_t} I_{\text{alive}} dW_t$$

Equations (3a, 3b) lead to the following recurrence formulae driving the law of X_t on each epoch (τ_i, τ_{i+1}) :

$$\mathbb{E}(X_t) = \mathbb{E}(X_{\tau_i})$$

$$\text{Var}(X_t) = \text{Var}(X_{\tau_i}) + \sigma_0^2 e^{2rn_{\tau_i}} (t - \tau_i) I_{\text{alive}}$$

Note that, alternatively, one can again prefer to apply Equations (4a, 4b). We can then show by induction that for any i , the expectation and covariance matrix at time τ_i are such that, for any (k, l) :

$$\mathbb{E}(X_{\tau_i}^{(k)}) = \mathbb{E}(X_0) \tag{S10}$$

$$\text{Cov}(X_{\tau_i}^{(k)}, X_{\tau_i}^{(l)}) = \text{Var}(X_0) + \sigma_0^2 \sum_{j=0}^{i-1} e^{2rn_{\tau_j}} (\tau_{j+1} - \tau_j) 1_{t_{k,l} > \tau_j} \tag{S11}$$

Indeed, we verify Equations (S10, S11) at step $i = 0$.

Now, suppose Equations (S10, S11) hold at step n . Using either Equations (3a, 3b) or (4a, 4b), we get :

$$\mathbb{E}(X_{\tau_{n+1}}^{(k)}) = \mathbb{E}(X_0)$$

$$\text{Cov}(X_{\tau_{n+1}}^{(k)}, X_{\tau_{n+1}}^{(l)}) = \text{Var}(X_0) + \sigma_0^2 \sum_{j=0}^n e^{2rn\tau_j} (\tau_{j+1} - \tau_j) 1_{t_{k,l} > \tau_j}$$

If τ_{n+1} is a death time of a lineage, Equations (S10, S11) are verified at step $n + 1$.

If τ_{n+1} is a branching time, we verify that the new lineage inherits the expectation and

covariances of its mother, as well as the same coalescence times with other lineages. It also

follows that Equations (S10, S11) are verified at step $n + 1$.

Finally, by induction, we get the tip distribution at present time $\tau_N = T$:

$$\mathbb{E}(X_T^{(k)}) = \mathbb{E}(X_0)$$

$$\text{Cov}(X_T^{(k)}, X_T^{(l)}) = \text{Var}(X_0) + \sigma_0^2 \sum_{j=0}^{N-1} e^{2rn\tau_j} (\tau_{j+1} - \tau_j) 1_{t_{k,l} > \tau_j}$$

B.2 Distribution of classic multivariate models

The same methodology applies to classic multivariate models that incorporate interactions

between traits within lineages but not between lineages. In our formalism, for all i , A_i and Γ_i are

block diagonal, with $d \times d$ blocks on the diagonal corresponding to the traits within each lineage.

We call these blocks respectively A^* and Γ^* . Moreover, the vector a_i is the repetition of identical

sequences a^* of d elements.

Writing the matrix products in Equations (3a, 3b) provides us with $d \times d$ blocks that

behave identically on each epoch. Indeed, we can use :

$$m_{\tau_i}^{*(k)} = \begin{pmatrix} \mathbb{E}(X_{\tau_i}^{(k,1)}) \\ \mathbb{E}(X_{\tau_i}^{(k,2)}) \\ \vdots \\ \mathbb{E}(X_{\tau_i}^{(k,d)}) \end{pmatrix} \text{ and } \Sigma_{\tau_i}^{*(k,l)} = \begin{pmatrix} \text{Cov}(X_{\tau_i}^{(k,1)}, X_{\tau_i}^{(l,1)}) & \text{Cov}(X_{\tau_i}^{(k,1)}, X_{\tau_i}^{(l,2)}) & \dots & \text{Cov}(X_{\tau_i}^{(k,1)}, X_{\tau_i}^{(l,d)}) \\ \text{Cov}(X_{\tau_i}^{(k,2)}, X_{\tau_i}^{(l,1)}) & \text{Cov}(X_{\tau_i}^{(k,2)}, X_{\tau_i}^{(l,2)}) & \dots & \text{Cov}(X_{\tau_i}^{(k,2)}, X_{\tau_i}^{(l,d)}) \\ \vdots & \vdots & \ddots & \vdots \\ \text{Cov}(X_{\tau_i}^{(k,d)}, X_{\tau_i}^{(l,1)}) & \text{Cov}(X_{\tau_i}^{(k,d)}, X_{\tau_i}^{(l,2)}) & \dots & \text{Cov}(X_{\tau_i}^{(k,d)}, X_{\tau_i}^{(l,d)}) \end{pmatrix}$$

In which case Equations (3a, 3b) lead to the recurrence formulae :

$$\begin{aligned}
 m_{\tau_{i+1}}^{*(k)} &= e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_{i+1})A^*}} m_{\tau_i}^{*(k)} + 1_{k \text{ alive}(\tau_{i+1})} \int_{\tau_i}^{\tau_{i+1}} e^{(s - \tau_{i+1})A^*} a^*(s) ds \\
 \Sigma_{\tau_{i+1}}^{*(k,l)} &= e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_{i+1})A^*}} \Sigma_{\tau_i}^{*(k,l)tr} \left(e^{(\tau_i - \tau_{i+1})1_{l \text{ alive}(\tau_{i+1})A^*}} \right) \\
 &\quad + 1_{k=l} \int_{\tau_i}^{\tau_{i+1}} \left(e^{(s - \tau_{i+1})A^*} \Gamma^* \right) tr \left(e^{(s - \tau_{i+1})A^*} \Gamma^* \right) ds
 \end{aligned}$$

We can then prove by induction that for any epoch i and any pair of lineages (k, l)

$$m_{\tau_i}^{*(k)} = e^{-\tau_i \wedge t_{k,k} A^*} m_0^* + \int_0^{\tau_i \wedge t_{k,k}} e^{(s - \tau_i \wedge t_{k,k}) A^*} a^*(s) ds \quad (\text{S12})$$

$$\Sigma_{\tau_i}^{*(k,l)} = e^{-\tau_i \wedge t_{k,k} A^*} \Sigma_0^{*tr} \left(e^{-\tau_i \wedge t_{l,l} A^*} \right) + \int_0^{\tau_i \wedge t_{k,k}} \left(e^{-\tau_i \wedge t_{k,k} A^*} \Gamma^* \right) tr \left(e^{-\tau_i \wedge t_{l,l} A^*} \Gamma^* \right) ds \quad (\text{S13})$$

Indeed, we verify Equations (S12, S13) at step $i = 0$.

Now, suppose Equations (S12, S13) hold at step i . Using Equations (3a, 3b), we get :

$$\begin{aligned}
 m_{\tau_{i+1}}^{*(k)} &= e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_i)A^*}} m_{\tau_i}^{*(k)} + 1_{k \text{ alive}(\tau_i)} \int_{\tau_i}^{\tau_{i+1}} e^{(s - \tau_{i+1})A^*} a^*(s) ds \\
 &= e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_i)A^*}} e^{-\tau_i \wedge t_{k,k} A^*} m_0^* + \int_0^{\tau_i \wedge t_{k,k}} e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_i)A^*}} e^{(s - \tau_i \wedge t_{k,k}) A^*} a^*(s) ds \\
 &\quad + 1_{k \text{ alive}(\tau_i)} \int_{\tau_i}^{\tau_{i+1}} e^{(s - \tau_{i+1})A^*} a^*(s) ds \\
 &= e^{-\tau_{i+1} \wedge t_{k,k} A^*} m_0^* + \int_0^{\tau_{i+1} \wedge t_{k,k}} e^{(s - \tau_{i+1} \wedge t_{k,k}) A^*} a^*(s) ds
 \end{aligned}$$

as well as :

$$\begin{aligned}
 \Sigma_{\tau_{i+1}}^{*(k,l)} &= e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_{i+1})A^*}} \Sigma_{\tau_i}^{*(k,l)tr} \left(e^{(\tau_i - \tau_{i+1})1_{l \text{ alive}(\tau_{i+1})A^*}} \right) \\
 &\quad + 1_{k=l} \int_{\tau_i}^{\tau_{i+1}} \left(e^{(s - \tau_{i+1})A^*} \Gamma^* \right) tr \left(e^{(s - \tau_{i+1})A^*} \Gamma^* \right) ds \\
 &= e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_{i+1})A^*}} e^{-\tau_i \wedge t_{k,k} A^*} \Sigma_0^{*tr} \left(e^{-\tau_i \wedge t_{l,l} A^*} \right) tr \left(e^{(\tau_i - \tau_{i+1})1_{l \text{ alive}(\tau_{i+1})A^*}} \right) \\
 &\quad + \int_0^{\tau_i \wedge t_{k,k}} e^{(\tau_i - \tau_{i+1})1_{k \text{ alive}(\tau_{i+1})A^*}} \left(e^{-\tau_i \wedge t_{k,k} A^*} \Gamma^* \right) tr \left(e^{-\tau_i \wedge t_{l,l} A^*} \Gamma^* \right) tr \left(e^{(\tau_i - \tau_{i+1})1_{l \text{ alive}(\tau_{i+1})A^*}} \right) ds \\
 &\quad + 1_{k=l} \int_{\tau_i}^{\tau_{i+1}} \left(e^{(s - \tau_{i+1})A^*} \Gamma^* \right) tr \left(e^{(s - \tau_{i+1})A^*} \Gamma^* \right) ds \\
 &= e^{-\tau_{i+1} \wedge t_{k,k} A^*} \Sigma_0^{*tr} \left(e^{-\tau_{i+1} \wedge t_{l,l} A^*} \right) + \int_0^{\tau_{i+1} \wedge t_{k,k}} \left(e^{-\tau_{i+1} \wedge t_{k,k} A^*} \Gamma^* \right) tr \left(e^{-\tau_{i+1} \wedge t_{l,l} A^*} \Gamma^* \right) ds
 \end{aligned}$$

If τ_{i+1} is a death time of a lineage, Equations (S12, S13) are verified at step $i + 1$.

If τ_{i+1} is a branching time, we verify that the new lineage inherits the expectation and covariances of its mother, as well as the same coalescence times with other lineages. It also follows that Equations (S12, S13) are verified at step $i + 1$.

Finally, by induction, we get the tip distribution :

$$m_T^{*(k)} = e^{-t_{k,k}A^*} m_0^* + \int_0^{t_{k,k}} e^{(s-t_{k,k})A^*} a^*(s) ds$$

$$\Sigma_T^{*(k,l)} = e^{-t_{k,k}A^*} \Sigma_0^{*tr} (e^{-t_{l,l}A^*}) + \int_0^{t_{k,l}} (e^{-t_{k,k}A^*} \Gamma^*)^{tr} (e^{-t_{l,l}A^*} \Gamma^*) ds$$

OU-BM model.—

As a first illustration, consider a model with $d = 3$ traits with equation on each epoch and on each lineage k as follows :

$$dX_t^{(k,1)} = \psi \left(b_1 + b_2 X_t^{(k,2)} + b_3 X_t^{(k,3)} - X_t^{(k,1)} \right) dt + \sigma_1 dW_t^{(k,1)}$$

$$dX_t^{(k,2)} = \sigma_2 dW_t^{(k,2)}$$

$$dX_t^{(k,3)} = \sigma_3 dW_t^{(k,3)}$$

These equations describe the evolution of two independent traits evolving following a BM (traits 2 and 3), and one trait following an OU with optimal trait value given by a linear combination of traits 2 and 3. Its main interest is to infer the dependence of one trait to two other independent traits on a phylogeny. Knowing the distribution at the beginning of a given epoch, we use Equations (3a, 3b) to compute the distribution at the end of the epoch.

A is block-diagonal with the following blocks A^* :

$$A^* = \begin{pmatrix} 1 & -b_2 & -b_3 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Writing $\Delta = s - \tau_{i+1}$, it follows that $e^{\Delta A^*}$ is block diagonal with 3×3 elements given by :

$$e^{\Delta A^*} = \begin{pmatrix} e^{\Delta} & -b_2(e^{\Delta} - 1) & -b_3(e^{\Delta} - 1) \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Moreover, Γ_i is block-diagonal with diagonal blocks :

$$\Gamma^* = \begin{pmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{pmatrix}$$

The matrix product $(e^{\Delta A_i} \Gamma_i)^{tr}(e^{\Delta A_i} \Gamma_i)$ is thus block-diagonal with 3×3 blocks :

$$\begin{pmatrix} (\sigma_1^2 + b_2^2 \sigma_2^2 + b_3^2 \sigma_3^2) e^{2\Delta} - 2(b_2^2 \sigma_2^2 + b_3^2 \sigma_3^2) e^{\Delta} + (b_2^2 \sigma_2^2 + b_3^2 \sigma_3^2) & -b_2 \sigma_2^2 (e^{\Delta} - 1) & -b_3 \sigma_3^2 (e^{\Delta} - 1) \\ -b_2 \sigma_2^2 (e^{\Delta} - 1) & \sigma_2^2 & 0 \\ -b_3 \sigma_3^2 (e^{\Delta} - 1) & 0 & \sigma_3^2 \end{pmatrix}$$

These matrices can be used to compute $m_T^{*(k)}$ and $\Sigma_T^{*(k,l)}$, with the help of Equations (S12, S13).

OU-OU model.—

Consider now a model with $d = 2$ traits with equation on each epoch and on each lineage k given by :

$$\begin{aligned} dX_t^{(k,1)} &= \psi \left(b_1 + b_2 X_t^{(k,2)} - X_t^{(k,1)} \right) dt + \sigma_1 dW_t^{(k,1)} \\ dX_t^{(k,2)} &= \psi \left(b_3 - X_t^{(k,2)} \right) dt + \sigma_2 dW_t^{(k,2)} \end{aligned}$$

These equations describe the evolution of one trait evolving following an OU (trait 2), and one trait following an OU with optimal trait value given by an affine transformation of trait 2. Its main interest is to infer the dependence of one trait to another trait on a phylogeny. Knowing the distribution at the beginning of a given epoch, we use Equations (3a, 3b) to compute the distribution at the end of the epoch.

A_i is block diagonal, with the following 2×2 blocks A^* :

$$A^* = \begin{pmatrix} 1 & -b_2 \\ 0 & 1 \end{pmatrix}$$

Again, writing $\Delta = s - \tau_{i+1}$, it follows that $e^{\Delta A_i}$ is block diagonal with 2×2 elements given by :

$$e^{\Delta A^*} = \begin{pmatrix} e^{\Delta} & -b_2 \Delta e^{\Delta} \\ 0 & e^{\Delta} \end{pmatrix}$$

Moreover, Γ_i is diagonal with repeated values :

$$\Gamma^* = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

The matrix product $(e^{\Delta A_i} \Gamma_i)^{tr} (e^{\Delta A_i} \Gamma_i)$ is thus block-diagonal with 2×2 blocks :

$$\begin{pmatrix} \sigma_1^2 e^{2\Delta} + b_2^2 \Delta^2 \sigma_2^2 e^{2\Delta} & -b_2 \sigma_2^2 \Delta e^{2\Delta} \\ -b_2 \sigma_2^2 \Delta e^{2\Delta} & \sigma_2^2 e^{2\Delta} \end{pmatrix}$$

These matrices can be used to compute $m_T^{*(k)}$ and $\Sigma_T^{*(k,l)}$, with the help of Equations (S12, S13).

C DISTRIBUTION FOR SOME MODELS WITH INTERACTIONS BETWEEN LINEAGES

C.1 Distribution with a constant, A symmetric, and $\Gamma = \sigma I$

When $\Gamma = \sigma I$ and A is symmetric, Equations (3a, 3b) become :

$$\begin{aligned}\mathbb{E}(X_t) &= e^{(\tau_i-t)A_i} \mathbb{E}(X_{\tau_i}) + \int_{\tau_i}^t e^{(s-t)A_i} a_i(s) ds \\ \text{Var}(X_t) &= (e^{(\tau_i-t)A_i}) \text{Var}(X_{\tau_i})^{tr} (e^{(\tau_i-t)A_i}) + \sigma^2 \int_{\tau_i}^t e^{2(s-t)A_i} ds\end{aligned}$$

If A_i is symmetric with coefficients in \mathbb{R} , it can be diagonalized by orthogonal passage matrices : we can exhibit a matrix Q verifying ${}^{tr}QA_iQ = \Lambda_i$ is diagonal and $Q^{-1} = {}^{tr}Q$.

$$\begin{aligned}\mathbb{E}(X_t) &= Qe^{(\tau_i-t)\Lambda_i} {}^{tr}Q \mathbb{E}(X_{\tau_i}) + Q \left(\int_{\tau_i}^t e^{(s-t)\Lambda_i} ds \right) {}^{tr}Q a_i \\ \text{Var}(X_t) &= Qe^{\Lambda_i(\tau_i-t)} {}^{tr}Q \text{Var}(X_{\tau_i}) Qe^{(\tau_i-t)\Lambda_i} {}^{tr}Q + \sigma^2 Q \left(\int_{\tau_i}^t e^{2(s-t)\Lambda_i} ds \right) {}^{tr}Q\end{aligned}$$

This is the expression that we need for the numerical integration, in particular, of the phenotype matching model.

Note that with A diagonalizable but not symmetric, Equations (3a, 3b) can also be reduced, but the transposition of A is no longer A , and it does not lead exactly to the same expression.

C.2 The phenotype matching (PM) model

We consider here the phenotype matching model introduced in Nuismer and Harmon (2014), with the following equation describing the evolution of any trait k on each epoch :

$$dX_t^{(k)} = \psi \left(\theta - X_t^{(k)} \right) dt + S \left(\left(\frac{1}{n_t} \sum_{l=1}^{n_t} X_t^{(l)} \right) - X_t^{(k)} \right) dt + \sigma dW_t^{(k)}$$

We introduce the line vector u , with value u_j that equals 1 if lineage j is alive, and 0 otherwise. In order to use our framework, we further want to express the model in the form given

by Equation (1). This is achieved by taking :

$$\begin{aligned} a_i &= \psi \theta^{tr} u \\ A_i &= (\psi + S) \text{diag}(u) - \frac{S}{u^{tr} u} {}^{tr} u u \\ \Gamma_i &= \sigma \text{diag}(u) \end{aligned}$$

239 where $\text{diag}(u)$ is the diagonal matrix with diagonal elements the elements of the vector u .

240 First, the tip distribution can be computed using the general algorithm that numerically
241 resolves the set of ODEs given in Equations (4a, 4b). Second, the PM model falls within the
242 class of models studied in the previous section, that is, with a symmetric A matrix. The tip
243 distribution can thus be numerically computed faster using this reduction.

244 We describe here a third (and faster) way to derive the tip distribution. It is based on an
245 analytical reduction of Equations (3a, 3b) that is specific to the PM model.

Remark that $\text{diag}(u)$ and ${}^{tr} u u$ commute, leading to the following calculus,

$$\begin{aligned} e^{(\tau_i - \tau_{i+1}) A_i} &= e^{(\tau_i - \tau_{i+1}) ((\psi + S) \text{diag}(u) - \frac{S}{u^{tr} u} {}^{tr} u u)} \\ &= e^{(\tau_i - \tau_{i+1}) (\psi + S) \text{diag}(u)} e^{-(\tau_i - \tau_{i+1}) \frac{S}{u^{tr} u} {}^{tr} u u} \\ &= \text{diag} \left(e^{(\tau_i - \tau_{i+1}) (\psi + S) u} \right) \left(\sum_{k \geq 0} \frac{\left(\frac{-(\tau_i - \tau_{i+1}) S}{u^{tr} u} \right)^k ({}^{tr} u u)^k}{k!} \right) \end{aligned}$$

Where e^w is the line vector with elements e^{w_j} . Further, remark that for any $k \geq 1$,

$$\begin{aligned} ({}^{tr} u u)^k &= ({}^{tr} u u) ({}^{tr} u u) ({}^{tr} u u) \dots ({}^{tr} u u) \\ &= {}^{tr} u (u {}^{tr} u) (u {}^{tr} u) \dots (u {}^{tr} u) u \\ &= (u {}^{tr} u)^{k-1} ({}^{tr} u u) \end{aligned}$$

246 For simplicity, we will write in the following $\Delta = \tau_i - \tau_{i+1}$, leading us to

$$\begin{aligned}
e^{\Delta A_i} &= \text{diag} \left(e^{(\psi+S)\Delta u} \right) \left(I + \sum_{k \geq 1} \frac{\left(\frac{-S\Delta}{u^{tr_u}} \right)^k (u^{tr_u})^{k-1} (tr_{uu})}{k!} \right) \\
&= \text{diag} \left(e^{(\psi+S)\Delta u} \right) \left(I + \frac{1}{u^{tr_u}} \left(\sum_{k \geq 1} \frac{(-(\tau_i - \tau_{i+1})S)^k}{k!} \right) tr_{uu} \right) \\
&= \text{diag} \left(e^{(\psi+S)\Delta u} \right) \left(I + \frac{1}{u^{tr_u}} (e^{-S\Delta} - 1) tr_{uu} \right) \\
&= \text{diag} \left(e^{(\psi+S)\Delta u} \right) + \frac{1}{u^{tr_u}} \text{diag} \left(e^{-S\Delta} e^{(\psi+S)\Delta u} \right) tr_{uu} - \frac{1}{u^{tr_u}} \text{diag} \left(e^{(\psi+S)\Delta u} \right) tr_{uu} \\
&= \text{diag} \left(e^{(\psi+S)\Delta u} \right) + \frac{1}{u^{tr_u}} (e^{\psi\Delta} - e^{(\psi+S)\Delta}) tr_{uu} \tag{S14}
\end{aligned}$$

Where the last equality is due to the product by tr_u , allowing to forget the cases where

$u_j = 0$ in the exponential.

We further need to compute

$$\begin{aligned}
\int_{\tau_i}^{\tau_{i+1}} e^{(s-\tau_{i+1})A_i} a_i ds &= \psi \theta \int_{\tau_i}^{\tau_{i+1}} e^{\psi(s-\tau_{i+1})} ds tr_u \\
&= \theta (1 - e^{\psi\Delta}) tr_u \tag{S15}
\end{aligned}$$

We thus get $m_{\tau_{i+1}}^-$ with the help of Equations (S14) and (S15).

Now, in order to simplify Equation (3b), remark that A_i and Γ_i are symmetric, and so are $e^{\Delta A_i}$ and $e^{\Delta A_i} \Gamma_i$. Moreover, Γ_i is diagonal, and commutes with any other matrix, leading to,

$$\Sigma_{\tau_{i+1}}^- = e^{\Delta A_i} \Sigma_{\tau_i} e^{\Delta A_i} + \int_{\tau_i}^{\tau_{i+1}} e^{2(s-\tau_{i+1})A_i} \Gamma_i \Gamma_i ds$$

The first term can be computed thanks to Equation (S14). For the second one, remark that $tr_{uu} \text{diag}(u) = tr_{uu}$, thus leading to

$$\begin{aligned}
\int_{\tau_i}^{\tau_{i+1}} e^{2(s-\tau_{i+1})A_i} \Gamma_i \Gamma_i ds &= \sigma^2 \int_{\tau_i}^{\tau_{i+1}} e^{2(\psi+S)(s-\tau_{i+1})} ds \text{diag}(u) \\
&\quad + \frac{\sigma^2}{u^{tr_u}} \int_{\tau_i}^{\tau_{i+1}} (e^{2\psi(s-\tau_{i+1})} - e^{2(\psi+S)(s-\tau_{i+1})}) ds tr_{uu} \text{diag}(u) \\
&= \sigma^2 \frac{(1 - e^{2(\psi+S)\Delta})}{2(\psi+S)} \text{diag}(u) + \frac{\sigma^2}{u^{tr_u}} \left(\frac{1 - e^{2\psi\Delta}}{2\psi} - \frac{1 - e^{2(\psi+S)\Delta}}{2(\psi+S)} \right) tr_{uu} \tag{S16}
\end{aligned}$$

We thus get $\Sigma_{\tau_{i+1}}^-$ with the help of Equations (S14) and (S16).

C.3 The phenotype matching (PM) model with biogeography

In this section we describe ways to compute the tip distribution under the PM model, taking into account the biogeography (that is, species interact only when they co-occur in the same localities). We consider a fixed number of islands N_I . Matrix U gives us the presence/absence of lineages in the distinct islands, with element u_{ij} that equals 1 if lineage j is present on island i and zero otherwise. Vector S gives the strength of interaction on each island. The model states that the trait of lineage j evolves through phenotype matching with all species that are sympatric :

$$dX_t^{(j)} = \psi \left(\theta - X_t^{(j)} \right) dt + \sum_{i=1}^{N_I} S_i u_{ij} \left(\frac{\sum_{l=1}^n u_{il} X_t^{(l)}}{\sum_{l=1}^n u_{il}} - X_t^{(j)} \right) dt + \sigma dW_t^{(j)}$$

Take for example 5 lineages evolving on 3 distinct islands with the following U matrix on a given epoch :

$$U = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This means that species number 1 is present on island 2 and 3, species number 2 is only present on island 1, and so on... Said differently, we see that species number 3 interacts on island 1 with species 2, and on island 2 with species 1 and 4. Our species traits are driven by the following equations :

$$\begin{aligned}
 dX_t^{(1)} &= \left(\psi \left(\theta - X_t^{(1)} \right) + S_2 \left(\frac{X_t^{(1)} + X_t^{(3)} + X_t^{(4)}}{3} - X_t^{(1)} \right) + S_3 \left(\frac{X_t^{(1)} + X_t^{(5)}}{2} - X_t^{(1)} \right) \right) dt + \sigma dW_t^{(1)} \\
 dX_t^{(2)} &= \left(\psi \left(\theta - X_t^{(2)} \right) + S_1 \left(\frac{X_t^{(2)} + X_t^{(3)}}{2} - X_t^{(2)} \right) \right) dt + \sigma dW_t^{(2)} \\
 dX_t^{(3)} &= \left(\psi \left(\theta - X_t^{(3)} \right) + S_1 \left(\frac{X_t^{(2)} + X_t^{(3)}}{2} - X_t^{(3)} \right) + S_2 \left(\frac{X_t^{(1)} + X_t^{(3)} + X_t^{(4)}}{3} - X_t^{(3)} \right) \right) dt + \sigma dW_t^{(3)} \\
 dX_t^{(4)} &= \left(\psi \left(\theta - X_t^{(4)} \right) + S_2 \left(\frac{X_t^{(1)} + X_t^{(3)} + X_t^{(4)}}{3} - X_t^{(4)} \right) \right) dt + \sigma dW_t^{(4)} \\
 dX_t^{(5)} &= \left(\psi \left(\theta - X_t^{(5)} \right) + S_3 \left(\frac{X_t^{(1)} + X_t^{(5)}}{2} - X_t^{(5)} \right) \right) dt + \sigma dW_t^{(5)}
 \end{aligned}$$

It thus follows that the vectorial equation can be written :

$$dX_t = \begin{pmatrix} \psi\theta \\ \psi\theta \\ \psi\theta \\ \psi\theta \\ \psi\theta \end{pmatrix} - \begin{pmatrix} \psi + \frac{2}{3}S_2 + \frac{1}{2}S_3 & 0 & -\frac{S_2}{3} & -\frac{S_2}{3} & -\frac{S_3}{2} \\ 0 & \psi + \frac{1}{2}S_1 & -\frac{S_1}{2} & 0 & 0 \\ -\frac{S_2}{3} & -\frac{S_1}{2} & \psi + \frac{1}{2}S_1 + \frac{2}{3}S_2 & -\frac{S_2}{3} & 0 \\ -\frac{S_2}{3} & 0 & -\frac{S_2}{3} & \psi + \frac{2}{3}S_2 & 0 \\ -\frac{S_1}{2} & 0 & 0 & 0 & \psi + \frac{1}{2}S_1 \end{pmatrix} X_t dt + \sigma dW_t$$

Provided no island is empty, the model can be written in our framework with $a = \psi\theta V$,

$\Gamma = \sigma I$, and, finally, A which is the matrix with elements :

$$\begin{aligned}
 (A)_{jj} &= \psi + \sum_{i=1}^{N_I} S_i u_{ij} \left(1 - \frac{1}{\sum_{l=1}^n u_{il}} \right) \\
 (A)_{jk} &= - \sum_{i=1}^{N_I} S_i u_{ij} u_{ik} \frac{1}{\sum_{l=1}^n u_{il}}
 \end{aligned}$$

Matrix A is symmetric, and we can thus use the developments presented in Appendix C.1 to speed up the computation time.

Nonetheless, a better analytical reduction can be derived when islands are exclusive, meaning that species are allowed to occur on one island only. Under this assumption, matrix $U^T U$ is diagonal with element $(U^T U)_{ii}$ being the number of lineages belonging to island i . We

now introduce the line vector r , of size N_I , full of ones. For simplicity, we also write in the following $\Delta = \tau_i - \tau_{i+1}$. With these notations, and provided no island is empty, the model can be written under our framework with :

$$\begin{aligned} a_i &= \psi \theta^T (rU) \\ A_i &= \text{diag}((\psi r + S)U) - {}^T U \text{diag}(S)(U^T U)^{-1} U \\ \Gamma_i &= \sigma \text{diag}(rU) \end{aligned}$$

As for the one island case, we can speed up the computation of the exponential by remarking that :

$$\begin{aligned} e^{\Delta A_i} &= e^{\Delta \text{diag}((\psi r + S)U)} e^{-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U} \\ &= e^{\Delta \text{diag}((\psi r + S)U)} \sum_{k \geq 0} \frac{(-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U)^k}{k!} \end{aligned}$$

We then observe that :

$$\begin{aligned} (-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U)^k &= (-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U)(-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U) \dots (-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U) \\ &= {}^T U (-\Delta \text{diag}(S))(U^T U)^{-1} (U^T U) (-\Delta \text{diag}(S))(U^T U)^{-1} (U^T U) \dots (U^T U) (-\Delta \text{diag}(S)) \\ &= {}^T U (-\Delta \text{diag}(S))^k (U^T U)^{-1} U \end{aligned}$$

Thus leading to the following expression :

$$\begin{aligned} e^{\Delta A_i} &= e^{\Delta \text{diag}((\psi r + S)U)} \left(I + \sum_{k \geq 1} \frac{(-\Delta {}^T U \text{diag}(S)(U^T U)^{-1} U)^k}{k!} \right) \\ &= \text{diag}(e^{\Delta(\psi r + S)U}) \left(I + {}^T U \left(\sum_{k \geq 1} \frac{(-\Delta \text{diag}(S))^k}{k!} \right) (U^T U)^{-1} U \right) \\ &= \text{diag}(e^{\Delta(\psi r + S)U}) (I + {}^T U (\text{diag}(e^{-\Delta S}) - I) (U^T U)^{-1} U) \\ &= \text{diag}(e^{\Delta(\psi r + S)U}) (I - {}^T U (U^T U)^{-1} U) + \text{diag}(e^{\Delta(\psi r + S)U}) {}^T U \text{diag}(e^{-\Delta S})(U^T U)^{-1} U \\ &= \text{diag}(e^{\Delta(\psi r + S)U}) (I - {}^T U (U^T U)^{-1} U) + \text{diag}(e^{\Delta(\psi r + S)U}) \text{diag}(e^{-\Delta S}) {}^T U (U^T U)^{-1} U \\ &= \text{diag}(e^{\Delta(\psi r + S)U}) (I - {}^T U (U^T U)^{-1} U) + \text{diag}(e^{\Delta \psi r U}) {}^T U (U^T U)^{-1} U \end{aligned} \tag{S17}$$

Where the second to last line holds under the assumption that each species belong to at most one island.

We further need to compute

$$\begin{aligned} \int_{\tau_i}^{\tau_{i+1}} e^{(s-\tau_{i+1})A_i} a_i ds &= \psi \theta \int_{\tau_i}^{\tau_{i+1}} \text{diag}(e^{(s-\tau_{i+1})\psi r U}) ds {}^T U^T r \\ &= \psi \theta \int_{\tau_i}^{\tau_{i+1}} e^{(s-\tau_{i+1})\psi} ds {}^T U^T r \\ &= \theta (1 - e^{\psi \Delta}) {}^T U^T r \end{aligned} \quad (\text{S18})$$

We thus get $m_{\tau_{i+1}}^-$ with the help of Equations (S17) and (S18).

We now turn to the reduction of the variance expression. Remark first that A_i and Γ_i are symmetric, and so are $e^{\Delta A_i}$ and $e^{\Delta A_i} \Gamma_i$. Moreover, Γ_i is diagonal, and commutes with $e^{\Delta A_i}$, leading to :

$$\Sigma_{\tau_{i+1}}^- = e^{\Delta A_i} \Sigma_{\tau_i} e^{\Delta A_i} + \int_{\tau_i}^{\tau_{i+1}} e^{2(s-\tau_{i+1})A_i} \Gamma_i \Gamma_i ds$$

The first term can be computed thanks to equation (S17). For the second one we get

$$\begin{aligned} \int_{\tau_i}^{\tau_{i+1}} e^{2(s-\tau_{i+1})A_i} \Gamma_i \Gamma_i ds &= \sigma^2 \int_{\tau_i}^{\tau_{i+1}} e^{2(s-\tau_{i+1})\text{diag}(r(\psi I + S)U)} ds (I - {}^T U (U^T U)^{-1} U) \text{diag}(rU) \\ &\quad + \sigma^2 \int_{\tau_i}^{\tau_{i+1}} e^{2(s-\tau_{i+1})\psi} \text{diag}(rU) ds {}^T U (U^T U)^{-1} U \text{diag}(rU) \\ &= \sigma^2 \int_{\tau_i}^{\tau_{i+1}} \text{diag}(e^{2(s-\tau_{i+1})(\psi r + S)U}) ds (\text{diag}(rU) - {}^T U (U^T U)^{-1} U) \\ &\quad + \sigma^2 \int_{\tau_i}^{\tau_{i+1}} \text{diag}(e^{2(s-\tau_{i+1})\psi r U}) ds {}^T U (U^T U)^{-1} U \end{aligned} \quad (\text{S19})$$

At the end, we get $\Sigma_{\tau_{i+1}}^-$ with the help of Equations (S17) and (S19).

C.4 The generalist matching mutualism (GMM) model

We recall the model formulation here. Assume that we rank first the n_1 plant traits, before the n_2 butterfly traits in the X vector. Traits evolve following the equation :

$$\forall k \in \{1, \dots, n_1\}, \quad dX_t^{(k)} = S \left(d_1 + \frac{1}{n_2} \sum_{l=n_1+1}^{n_1+n_2} X_t^{(l)} - X_t^{(k)} \right) dt + \sigma dW_t^{(k)}$$

$$\forall l \in \{n_1 + 1, \dots, n_1 + n_2\}, \quad dX_t^{(l)} = S \left(d_2 + \frac{1}{n_1} \sum_{k=1}^{n_1} X_t^{(k)} - X_t^{(l)} \right) dt + \sigma dW_t^{(l)}$$

In the general framework formulation, this leads to :

$$a(t) = \text{tr}(Sd_1, \dots, Sd_1, Sd_2, \dots, Sd_2)$$

$$A = \begin{pmatrix} S & 0 & \dots & 0 & \frac{-S}{n_2} & \dots & \dots & \frac{-S}{n_2} \\ 0 & \ddots & \ddots & \vdots & \vdots & & & \vdots \\ \vdots & \ddots & \ddots & 0 & \vdots & & & \vdots \\ 0 & \dots & 0 & \ddots & \frac{-S}{n_2} & \dots & \dots & \frac{-S}{n_2} \\ \frac{-S}{n_1} & \dots & \dots & \frac{-S}{n_1} & \ddots & 0 & \dots & 0 \\ \vdots & & & \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & & & \vdots & \vdots & \ddots & \ddots & 0 \\ \frac{-S}{n_1} & \dots & \dots & \frac{-S}{n_1} & 0 & \dots & 0 & S \end{pmatrix}$$

$$\Gamma = \sigma I$$

We would like to be able to compute the expectation and variance easily on each epoch.

We thus want to reduce Equations (3a, 3b). For simplicity, we will write in the following

$\Delta = \tau_i - \tau_{i+1}$. With some work, we can find the generic element of the matrix $e^{\Delta A}$.

First, we decompose $A = S(I + Z)$, where I is the identity matrix, and Z is made of two

blocks with elements $\frac{-1}{n_2}$ and $\frac{-1}{n_1}$. I and Z commute, meaning that :

$$e^{\Delta A} = e^{\Delta S(I+Z)} = e^{\Delta SI} e^{\Delta SZ} = e^{\Delta S} e^{\Delta SZ}$$

Moreover, we can find by induction the generic element of the matrix Z^k , as presented in

Figure (S2).

We then use this to find the generic element of the matrix

$e^{\Delta SZ} = \sum_{k \geq 0} \frac{S^k \Delta^k Z^k}{k!} = I + \sum_{k \geq 1} \frac{S^k \Delta^k Z^k}{k!}$. We recall that the odd and even parts of the

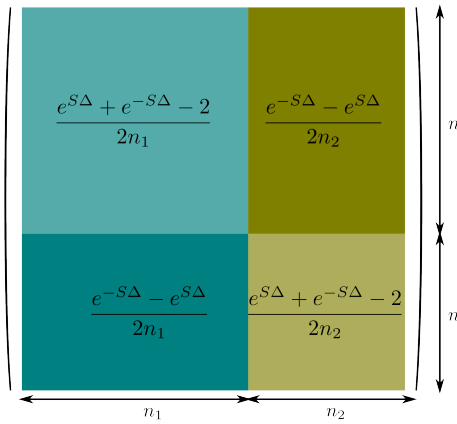
exponential are :

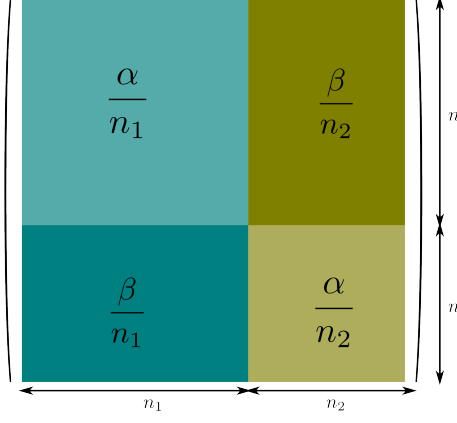
$$\begin{aligned}
 Z &= \begin{pmatrix} & \xrightarrow{n_2} \\ \uparrow n_1 & \begin{matrix} -\frac{1}{n_2} \end{matrix} \\ \begin{matrix} -\frac{1}{n_1} \end{matrix} & \downarrow n_2 \\ & \xleftarrow{n_1} \end{pmatrix} & Z^2 &= \begin{pmatrix} \begin{matrix} \frac{1}{n_1} \end{matrix} & \\ & \begin{matrix} \frac{1}{n_2} \end{matrix} \end{pmatrix} \\
 &\vdots & & \vdots \\
 Z^{2k+1} &= \begin{pmatrix} & \begin{matrix} -\frac{1}{n_2} \end{matrix} \\ \begin{matrix} -\frac{1}{n_1} \end{matrix} & \end{pmatrix} & Z^{2k} &= \begin{pmatrix} \begin{matrix} \frac{1}{n_1} \end{matrix} & \\ & \begin{matrix} \frac{1}{n_2} \end{matrix} \end{pmatrix}
 \end{aligned}$$

FIGURE S2: Generic element of the matrix Z^k , $\forall k \in \mathbb{N}^*$.

$$\begin{aligned}
 e^\lambda - e^{-\lambda} &= \sum_{k \geq 0} \frac{\lambda^k}{k!} - \sum_{k \geq 0} \frac{(-1)^k \lambda^k}{k!} = 2 \sum_{k \geq 0} \frac{\lambda^{2k+1}}{(2k+1)!} \\
 \text{and } e^\lambda + e^{-\lambda} &= 2 \sum_{k \geq 0} \frac{\lambda^{2k}}{(2k)!}
 \end{aligned}$$

305 Then, matrices $e^{\Delta SZ}$ and $e^{\Delta A}$ are composed of four distinct blocks, which expressions are
 306 shown in Figure S3.

$$e^{\Delta SZ} = I + \sum_{k \geq 1} \frac{S^k \Delta^k Z^k}{k!} = I +$$


$$e^{\Delta A} = e^{\Delta S} e^{\Delta SZ} = e^{\Delta S} I +$$


where $\alpha := \frac{e^{2S\Delta} - 2e^{S\Delta} + 1}{2}$
 $\beta := \frac{1 - e^{2S\Delta}}{2}$

FIGURE S3: Generic elements of matrices $e^{\Delta SZ}$ and $e^{\Delta A}$.

We thus got the main element from which we can derive the expectation vector $m_{\tau_{i+1}^-}$:

$$m_{\tau_{i+1}^-} = e^{\Delta A_i} m_{\tau_i} + \int_{\tau_i}^{\tau_{i+1}} e^{(s-\tau_{i+1})A_i} a_i(s) ds$$

$$= e^{\Delta A_i} m_{\tau_i} + \int_{\tau_i}^{\tau_{i+1}} \begin{pmatrix} Sd_1 e^{S(s-\tau_{i+1})} + Sd_1 \frac{e^{2S(s-\tau_{i+1})} - 2e^{S(s-\tau_{i+1})} + 1}{2} + Sd_2 \frac{1 - e^{2S(s-\tau_{i+1})}}{2} \\ \vdots \\ Sd_1 e^{S(s-\tau_{i+1})} + Sd_1 \frac{e^{2S(s-\tau_{i+1})} - 2e^{S(s-\tau_{i+1})} + 1}{2} + Sd_2 \frac{1 - e^{2S(s-\tau_{i+1})}}{2} \\ Sd_2 e^{S(s-\tau_{i+1})} + Sd_1 \frac{1 - e^{2S(s-\tau_{i+1})}}{2} + Sd_2 \frac{e^{2S(s-\tau_{i+1})} - 2e^{S(s-\tau_{i+1})} + 1}{2} \\ \vdots \\ Sd_2 e^{S(s-\tau_{i+1})} + Sd_1 \frac{1 - e^{2S(s-\tau_{i+1})}}{2} + Sd_2 \frac{e^{2S(s-\tau_{i+1})} - 2e^{S(s-\tau_{i+1})} + 1}{2} \end{pmatrix} ds$$

$$= e^{\Delta A_i} m_{\tau_i} + \int_{\tau_i}^{\tau_{i+1}} \begin{pmatrix} S \frac{d_1 + d_2}{2} + S \frac{d_1}{2} e^{2S(s-\tau_{i+1})} - S \frac{d_2}{2} e^{2S(s-\tau_{i+1})} \\ \vdots \\ S \frac{d_1 + d_2}{2} + S \frac{d_1}{2} e^{2S(s-\tau_{i+1})} - S \frac{d_2}{2} e^{2S(s-\tau_{i+1})} \\ S \frac{d_1 + d_2}{2} - S \frac{d_1}{2} e^{2S(s-\tau_{i+1})} + S \frac{d_2}{2} e^{2S(s-\tau_{i+1})} \end{pmatrix} ds$$

We now turn to the derivation of the covariance matrix, which requires simplifying :

$$\int_{\tau_i}^{\tau_{i+1}} \left(e^{(s-\tau_{i+1})A_i} \Gamma_i(s) \right) \text{tr} \left(e^{(s-\tau_{i+1})A_i} \Gamma_i(s) \right) ds = \sigma^2 \int_{\tau_i}^{\tau_{i+1}} \left(e^{(s-\tau_{i+1})A_i} \right) \text{tr} \left(e^{(s-\tau_{i+1})A_i} \right) ds$$

307

The expression of this last matrix is given in Figure S4.

$$\begin{aligned}
 & \sigma^2 \int_{\tau_i}^{\tau_{i+1}} \left(e^{(s-\tau_{i+1})A_i} \right) \text{tr} \left(e^{(s-\tau_{i+1})A_i} \right) ds \\
 &= \sigma^2 \int_{\tau_i}^{\tau_{i+1}} e^{S(s-\tau_{i+1})} I + \begin{pmatrix} \frac{\alpha}{n_1} & \frac{\beta}{n_2} \\ \frac{\beta}{n_1} & \frac{\alpha}{n_2} \end{pmatrix} e^{S(s-\tau_{i+1})} I + \begin{pmatrix} \frac{\alpha}{n_1} & \frac{\beta}{n_1} \\ \frac{\beta}{n_2} & \frac{\alpha}{n_2} \end{pmatrix} ds \\
 &= \sigma^2 \int_{\tau_i}^{\tau_{i+1}} e^{2S(s-\tau_{i+1})} I + \left(\frac{1}{n_1} + \frac{1}{n_2} \right) \begin{pmatrix} \frac{e^{4S(s-\tau_{i+1})} - 2e^{2S(s-\tau_{i+1})} + 1}{4} & \frac{1 - e^{4S(s-\tau_{i+1})}}{4} \\ \frac{1 - e^{4S(s-\tau_{i+1})}}{4} & \frac{e^{4S(s-\tau_{i+1})} - 2e^{2S(s-\tau_{i+1})} + 1}{4} \end{pmatrix} ds \\
 &= \sigma^2 \left[\frac{1 - e^{2S\Delta}}{2S} I + \left(\frac{1}{n_1} + \frac{1}{n_2} \right) \begin{pmatrix} \frac{1 - e^{4S\Delta}}{16S} - \frac{1 - e^{2S\Delta}}{4S} - \frac{\Delta}{4} & \frac{e^{4S\Delta} - 1}{16S} - \frac{\Delta}{4} \\ \frac{e^{4S\Delta} - 1}{16S} - \frac{\Delta}{4} & \frac{1 - e^{4S\Delta}}{16S} - \frac{1 - e^{2S\Delta}}{4S} - \frac{\Delta}{4} \end{pmatrix} \right]
 \end{aligned}$$

FIGURE S4: Generic elements of matrices that help us compute the covariance matrix of the distribution.

D SIMULATION AND INFERENCE

We do not give any new result in this Appendix section. Instead, we present the ways we implemented numerically simulations and inferences for all models described in the paper. These have been previously described in a number of papers.

D.1 Numerical methods for simulating data

Simulating the whole trajectory of the process.—

We use the Euler-Maruyama scheme, which works like the Euler scheme for ODEs, but with the addition of a small Gaussian random variable at each time step (Gardiner et al. 1985). We discretize each epoch (τ_i, τ_{i+1}) with a mesh Δ_t . We consider m standard Gaussian vectors of dimension $nd : (U_j)_{j=1}^m$. We approximate our SDE on this interval in the following way :

$$Y_0 = X_0$$

$$Y_{\tau_i+m\Delta_t} = Y_{\tau_i+(m-1)\Delta_t} + (a_i(\tau_i + (m-1)\Delta_t) - A_i Y_{\tau_i+(m-1)\Delta_t})\Delta_t + \Gamma(\tau_i + (m-1)\Delta_t)\sqrt{\Delta_t}U_m$$

When a branching occurs, the values of the process on the splitting branch are duplicated at the end of the vector Y . We then iterate this operation from the root up to present time.

This simulation allows us to get the whole trajectory of the process on the tree, which can mainly be used to produce pictures as in Figure S5, and eventually get a useful intuition on the process. However, we rarely use the whole trajectories, because observed data are only composed of tip trait values.

Simulating values of the process at the tips only.—

This second simulation protocol allows us to simulate the process values at the tips only. Suppose that we know the vector m of expectations and the covariance matrix Σ at the tips of the tree.

We then simply simulate numerically a Gaussian vector with law :

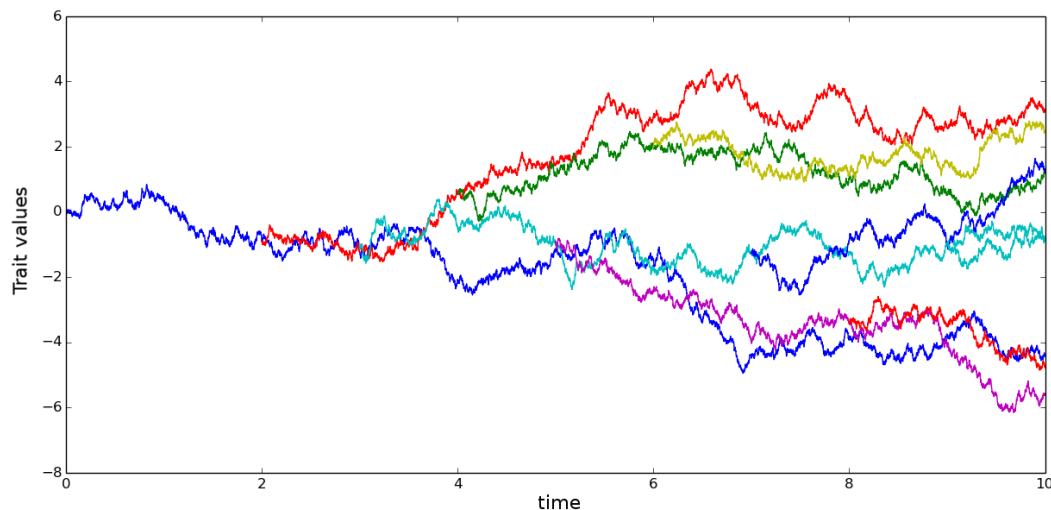


FIGURE S5: Evolution of a Brownian phenotypic trait along a tree, following the SDE : $dX_t = \sigma IdW_t$.

$$X_{t_f} \sim \mathcal{N}(m, \Sigma)$$

This is by far the quickest way to get the tip values. However, as the inference protocol relies on the use of the same vector of expectations and covariance matrix, one may prefer to use the other simulation protocols to test the consistency between simulation and inference. In case there is an issue with the derivation of the tip distribution, there would be a discrepancy between simulations and inferences.

D.2 Parameter inference

Parameter inference principle.—

We consider here that we know the topology of the true phylogeny with K tips, its branch lengths, and the state of d phenotypic traits at the tip, denoted by \mathcal{X} .

We assume any model of phenotypic evolution relying on linear SDEs, with vector of parameters p . We can compute the expectation m_p and the covariance Σ_p of the process X at tree tips. Its law is then : $X \sim \mathcal{N}(m_p, \Sigma_p)$, and, assuming that the variance matrix is invertible,

the density of the vector X is :

$$\forall x \in \mathbb{R}^{Kd}, \quad f(x) = \frac{1}{\sqrt{(2\pi)^{Kd} \det(\Sigma_p)}} e^{-\frac{1}{2} \text{tr}(x-m_p) \Sigma_p^{-1} (x-m_p)}$$

We can thus write the likelihood of the observed phenotypic traits as,

$$\begin{aligned} \mathcal{L}(p) &= f(\mathcal{X}|p) \\ &= \frac{1}{\sqrt{(2\pi)^{Kd} \det(\Sigma_p)}} e^{-\frac{1}{2} \text{tr}(\mathcal{X}-m_p) \Sigma_p^{-1} (\mathcal{X}-m_p)} \end{aligned}$$

The maximum likelihood estimators (MLE) are the parameter values that maximize the likelihood function, that is,

$$\hat{p} = \underset{p}{\operatorname{argmax}} \mathcal{L}(p)$$

Equivalently, we can minimize the following function,

$$-\ln(\mathcal{L}(p)) = \frac{1}{2} Kd \ln(2\pi) + \frac{1}{2} \ln(\det(\Sigma_p)) + \frac{1}{2} \text{tr}(\mathcal{X} - m_p) \Sigma_p^{-1} (\mathcal{X} - m_p)$$

or, removing the constants,

$$U(p) = \ln(\det(\Sigma_p)) + \text{tr}(\mathcal{X} - m_p) \Sigma_p^{-1} (\mathcal{X} - m_p)$$

338 *Analytical derivation of the MLE.—*

339 Among all models described in the paper, only the BM model allows the analytic
340 derivation of the MLE estimators. Take for illustration a BM model without drift starting with
341 $(m_0, v_0) = (0, 0)$. According to Table 1, the expectation m and covariance matrix Σ at the tips
342 are $m = 0$ and $\Sigma = \sigma^2 T$, where matrix T has element $T^{(k,l)} = t_{k,l}$.

We get the MLE $\hat{\sigma}$ by looking analytically for the minimum of U ,

$$\begin{aligned} U(\sigma) &= \ln(\det(\sigma^2 T)) + \text{tr} \mathcal{X} \frac{T^{-1}}{\sigma^2} \mathcal{X} \\ &= \ln \det T + 2n \ln \sigma + \frac{1}{\sigma^2} \text{tr} \mathcal{X} T^{-1} \mathcal{X} \\ \frac{dU}{d\sigma} &= \frac{2n}{\sigma} - \frac{2}{\sigma^3} \text{tr} \mathcal{X} T^{-1} \mathcal{X} \end{aligned}$$

Thus leading to,

$$\hat{\sigma}^2 = \frac{1}{n} \text{tr} \mathcal{X} T^{-1} \mathcal{X}$$

343 *Speeding up the MLE estimation by reducing the dimension of the parameter space.—*

Maximizing the likelihood can take a long time, especially when the dimension of the parameter space is large. It can thus be interesting to make assumptions that lower the number of parameters, when this is biologically tolerable. Examples include,

- starting an OU process with $m_0 = \theta$,
- considering no root variance, $v_0 = 0$,
- starting a PM model with $m_0 = \theta$ (in which case we easily show that the expectation remains θ in all lineages),
- putting $\psi = 0$ in the PM model.

In many models (e.g. BM, OU, ACDC, PM with $m_0 = \theta...$), distinct sets of parameters p_1 and p_2 are involved in the computation of m and Σ , and the expectation vector m can be expressed as $m = Cp_1$. In this case, at a given p_2 , we can analytically get the parameters p_1 maximizing $\ln(\mathcal{L}(p_1, p_2))$,

$$\frac{\partial}{\partial p_1} U(p_1, p_2) = 0 \iff \frac{d}{dp_1} \text{tr}(\mathcal{X} - Cp_1)\Sigma_{p_2}^{-1}(\mathcal{X} - Cp_1) = 0$$

Doing so, we get the same formula as in (Hansen 1997; Butler and King 2004), i.e. :

$$\hat{p}_1 = (\text{tr}C_1\Sigma_{p_2}^{-1}C_1)^{-1} \text{tr}C_1\Sigma_{p_2}^{-1}X$$

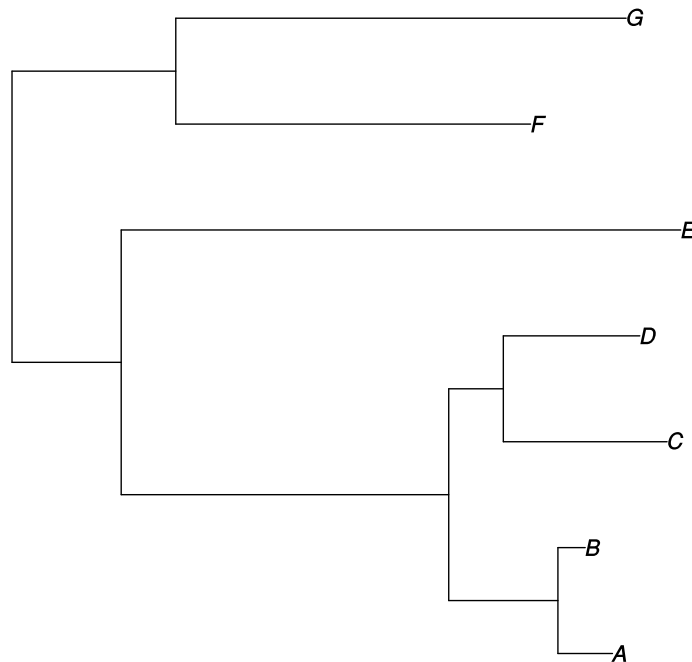
E TUTORIAL : USING THE RPANDA CODES TO STUDY TRAIT COEVOLUTION

The aim of this section is to describe the R codes associated to our framework. We describe the class `PhenotypicModel`, we show how to manipulate the different methods included in the class, we illustrate their use around a simple (non-ultrametric) tree, and we finally explain how to use our codes to write new models fitting the framework.

We first need to load usefull R packages, along with our codes, and a small, non-ultrametric, tree.

```
In [219]: source("Loading.R")

newick <- "(((A:1,B:0.5):2,(C:3,D:2.5):1):6,E:10.25):2,(F:6.5,G:8.25):3):1;"
tree <- read.tree(text=newick)
plot(tree)
```



E.1 The 'PhenotypicModel' class

Our code is structured around one main R class that we called 'PhenotypicModel', which is intended to mimic the framework that we proposed in the main text. Each object of the 'PhenotypicModel' encompasses informations on the tree, on the parameters of the model, on the starting values, and, finally, on the collection of (a_i, A_i, Γ_i) for all epochs.

Loading a pre-defined model.—

Because we wanted this code both to be user-friendly and to serve as an illustration of what can be written within this framework, we implemented all models in main Table 1 in a generic constructor `createModel`, in the file 'ModelBank.R', that takes for arguments the tree and the name of the required model.

Available models include :

BM Brownian Motion model with linear drift.

Starts with two lineages having the same value $X_0 \sim \mathcal{N}(m_0, v_0)$.

One trait in each lineage, all lineages evolving independently after branching following the equation.

$$dX_t^{(i)} = ddt + \sigma dW_t^{(i)}$$

BM_from0 Same as above, but starting with two lineages having the same value $X_0 \sim \mathcal{N}(0, 0)$.

BM_from0_driftless Same as above, but with $d = 0$.

OU Ornstein-Uhlenbeck model.

Starts with two lineages having the same value $X_0 \sim \mathcal{N}(m_0, v_0)$.

One trait in each lineage, all lineages evolving independently after branching, following the equation :

$$dX_t^{(i)} = \psi(\theta - X_t)dt + \sigma dW_t^{(i)}$$

OU_from0 Same as above, but starting with two lineages having the same value $X_0 \sim \mathcal{N}(0, 0)$.

383 **ACDC** ACcelerating or DeCelerating model.

384 Starts with two lineages having the same value $X_0 \sim \mathcal{N}(m_0, v_0)$.

One trait in each lineage, all lineages evolving independently after branching, following the equation :

$$dX_t^{(i)} = \sigma_0 e^{rt} dW_t^{(i)}$$

385 **DD** Diversity-Dependent model.

386 Starts with two lineages having the same value $X_0 \sim \mathcal{N}(m_0, v_0)$.

One trait in each lineage, all lineages evolving independently after branching, following the equation :

$$dX_t^{(i)} = \sigma_0 e^{rn_t} dW_t^{(i)}$$

387 **PM** Phenotype Matching model.

388 Starts with two lineages having the same value $X_0 \sim \mathcal{N}(m_0, v_0)$.

One trait in each lineage, all lineages evolving then non-independently following the expression :

$$dX_t^{(i)} = \psi \left(\theta - X_t^{(i)} \right) + S \left(\frac{1}{n} \sum_{k=1}^n X_t^{(k)} - X_t^{(i)} \right) + \sigma dW_t^{(i)}$$

389 **PM_OUless** Simplified Phenotype Matching model.

390 Starts with two lineages having the same value $X_0 \sim \mathcal{N}(m_0, v_0)$.

One trait in each lineage, all lineages evolving then non-independently following the expression :

$$dX_t^{(i)} = S \left(\frac{1}{n} \sum_{k=1}^n X_t^{(k)} - X_t^{(i)} \right) + \sigma dW_t^{(i)}$$

391 To get a first glimpse at ‘PhenotypicModel’ objects, we first create two such objects. The
392 first one is a Brownian Motion (BM), the second one is an Ornstein-Uhlenbeck process (OU).
393 Note that both models include m_0 and v_0 as parameters.

```
In [220]: modelBM <- createModel(tree, 'BM')
          modelOU <- createModel(tree, 'OU')
```

394 *Access to the content of the model.—*

395 The function `show` (resp. `print`) is intended to give basic (resp. full) information on a
396 specific 'PhenotypicModel' object.

In [221]: `show(modelBM)`

```
*****
*** Object of Class PhenotypicModel ***
*** Name of the model : [1] "BM"
*** Parameters of the model : [1] "m0"      "v0"      "d"      "sigma"
*** Description : Brownian Motion model with linear drift.
Starts with two lineages having the same value  $X_0 \sim \text{Normal}(m0, v0)$ .
One trait in each lineage, all lineages evolving independently after branching.
 $dX_t = d \, dt + \text{sigma} \, dW_t$ 
*** Periods : the model is cut into 13 parts.
For more details on the model, call : print(PhenotypicModel)
*****
```

In [222]: `print(modelOU)`

```
*****
*** Object of Class PhenotypicModel ***
*** Name of the model : [1] "OU"
*** Parameters of the model : [1] "m0"      "v0"      "psi"      "theta" "sigma"
*** Description : Ornstein-Uhlenbeck model.
Starts with two lineages having the same value  $X_0 \sim \text{Normal}(m0, v0)$ .
One trait in each lineage, all lineages evolving independently after branching.
 $dX_t = \text{psi}(\text{theta} - X_t) \, dt + \text{sigma} \, dW_t$ 
*** Epochs : the model is cut into 13 parts.
[1] 0.00 2.00 3.00 8.00 9.00 9.50 10.00 10.50 11.00 11.25 11.50 12.00
[13] 12.25
*** Lineages branching (to be copied at the end of the corresponding period) :
```

```
[1] 1 1 2 1 5 2 1 7 1 4 6 5 3

*** Positions of the new trait at the end of each period :

[1] 2 3 4 5 6 0 7 0 0 0 0 0 0

*** Initial condition :

function (params)
return(list(mean = c(params[1]), var = matrix(c(params[2]))))
<environment: 0x9617460>

*** Vectors a_i, A_i, Gamma_i on each period i :

function (i, params)
{
  vectorU <- getLivingLineages(i, eventEndOfPeriods)
  vectorA <- function(t) return(params[3] * params[4] * vectorU)
  matrixGamma <- function(t) return(params[5] * diag(vectorU))
  matrixA <- params[3] * diag(vectorU)
  return(list(a = vectorA, A = matrixA, Gamma = matrixGamma))
}
<environment: 0x9617460>

*** Constraints on the parameters :

function (params)
return(params[2] >= 0 && params[5] >= 0 && params[3] != 0)
<environment: 0x9617460>

*** Default parameter values : [1] 0 0 1 0 1

*** Tip labels :

[1] "A" "B" "C" "D" "E" "F" "G"

*** Tip labels for simulations :

[1] "A" "F" "E" "G" "C" "D" "B"

*****
```

397 *List of class attributes.—*

398 The latter command gave us some insight into how a `PhenotypicModel` is defined. It has

399 the following list of attributes :

400 **name** a name,

401 **paramsNames** the names of all parameters,

402 **comment** a short description,

403 **period** the vector of times at which successive branching and death of lineages occur,

404 **numbersCopy** vector containing the lineage number which branches or dies at the end of each
405 period,

406 **numbersPaste** vector containing the lineage number in which a daughter lineage is placed at
407 the end of each period (zero if the end of the period corresponds to a death),

408 **initialCondition** a function of the parameters giving the initial mean and variance of the
409 gaussian process at the root of the tree,

410 **aAGamma** the functions corresponding to $a_i(t)$, A_i , and $\Gamma_i(t)$ that define the evolution of the
411 process on each period, depending on parameters,

412 **constraints** a function of the parameters giving the definition range,

413 **params0** a vector of default parameter values.

414 Each of these attributes can be accessed and changed through the use of the following
415 syntax.

```
In [223]: modelBM['name']
```

416 Out [223]: 'BM'

```
In [224]: modelBM['paramsNames']
```

417 Out [224]: 'm0' 'v0' 'd' 'sigma'


```
In [225]: modelOU['paramsNames'] <- c("mean0", "var0", "selectionStrength", "equilibrium",
    "noise")

    show(modelOU)

*****

*** Object of Class PhenotypicModel ***

*** Name of the model : [1] "OU"

*** Parameters of the model : [1] "mean0"          "var0"          "selectionStrength"
[4] "equilibrium"      "noise"

*** Description : Ornstein-Uhlenbeck model.

Starts with two lineages having the same value  $X_0 \sim \text{Normal}(m_0, v_0)$ .

One trait in each lineage, all lineages evolving independently after branching.

 $dX_t = \psi(\theta - X_t) dt + \sigma dW_t$ 

*** Periods : the model is cut into 13 parts.

For more details on the model, call : print(PhenotypicModel)

*****
```

418 However, changes must be made cautiously, in order to keep a coherent model. For
 419 example, changing 'paramsNames' for a shorter vector would not be authorized, but other
 420 deleterious actions could work and lead to issues with methods associated to PhenotypicModel
 421 objects.

```
In [226]: modelOU['paramsNames'] <- c("mean0", "var0")

Error in validityMethod(as(object, superClass)): [PhenotypicModel : validation]
There should be the same number of default parameters and parameter names.
```

422 *E.2 Methods associated to the 'PhenotypicModel' class*

423 All 'PhenotypicModel' objects are associated to methods intended to do the basic operations
 424 that we need to do with models of trait evolution, i.e.,

425 1. simulate tip trait data,

2. compute the likelihood of tip trait data,

3. fit the model to tip trait data.

Simulating tip trait data.—

The method `simulateTipData` works for any `PhenotypicModel` object. We simply give it the model and the set of parameters and it returns a realisation of the process (tip data).

```
In [227]: dataBM <- simulateTipData(modelBM, c(0,0,0,1))
```

```
dataBM
```

```
*** Simulation of tip trait values ***
```

```
Simulates step-by-step the whole trajectory, but returns only the tip data.
```

```
Computation time : 0.3909395 secs
```

```
Out [227]:
```

A	-2.71863
F	1.043329
E	0.665404
G	-3.440327
C	0.272335
D	-0.7023421
B	-2.010951

A third, optional, argument, changes the behaviour of the method.

- "method=1" : first computes the tip distribution at present, before drawing a realization of this distribution,
- "method=2" : simulates step-by-step the whole trajectory of the process, plots the trajectories through time, and returns the tip data.
- "method=3" : (default) simulates step-by-step the whole trajectory of the process, before returning only the tip data.

```
In [228]: dataOU <- simulateTipData(modelOU, c(0,0,1,5,1), method=1)

dataOU
```

*** Simulation of tip trait values ***

Computes the tip distribution, and returns a simulated dataset drawn in this distribution.

Computation time : 0.0009741783 secs

441 Out [228]:

A	4.179412
B	5.776153
C	4.984526
D	4.480901
E	5.693471
F	4.636019
G	5.815942

```
In [229]: simulateTipData(modelBM, c(0,0,0,1), method=2)
```

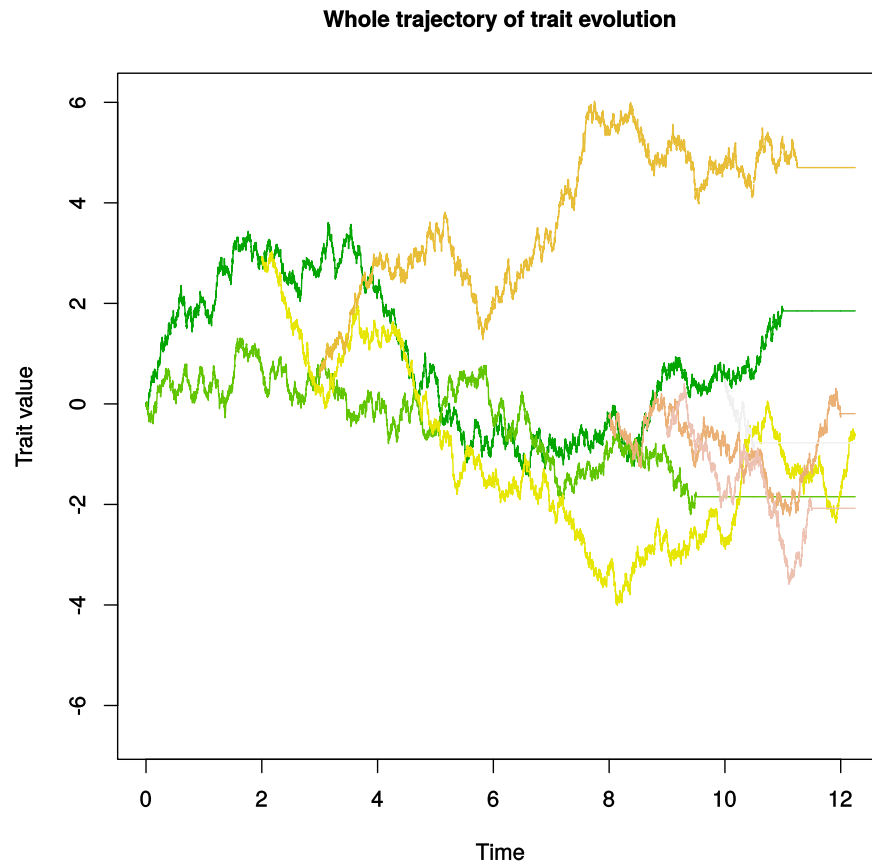
*** Simulation of tip trait values ***

Simulates step-by-step the whole trajectory, plots it, and returns tip data.

Computation time : 0.479032 secs

443 Out [229]:

A	1.850113
F	-1.846854
E	-0.6321431
G	4.701758
C	-0.1940776
D	-2.077116
B	-0.7752916



445

446 *Getting the distribution of the model under a given set of parameters.—*

447 The method `getTipDistribution` computes the mean vector m and variance matrix Σ
 448 such that, under the model, the tip trait data X follows $\mathcal{N}(m, \Sigma)$.

449 The related method `getDataLikelihood` returns the $-\ln(\text{likelihood})$ of a given data set
 450 under the model, with a given set of parameters.

In [230]: `getTipDistribution(modelBM, c(0,0,1,1))`

451 Out [230]:

452	\$mean	A	11						
		B	10.5						
		C	12						
		D	11.5						
		E	12.25						
		F	9.5						
		G	11.25						
453	\$Sigma		A	B	C	D	E	F	G
		A	11	10	8	8	2	0	0
		B	10.0	10.5	8.0	8.0	2.0	0.0	0.0
		C	8	8	12	9	2	0	0
		D	8.0	8.0	9.0	11.5	2.0	0.0	0.0
		E	2.00	2.00	2.00	2.00	12.25	0.00	0.00
		F	0.0	0.0	0.0	0.0	0.0	9.5	3.0
		G	0.00	0.00	0.00	0.00	0.00	3.00	11.25

In [231]: `getDataLikelihood(modelBM, dataBM, c(0,0,1,1))`

454 Out[231]: 36.0510113479088

455 *Maximum likelihood estimation of parameters.—*

456 The method `fitTipData` uses the latter two methods to find the set of parameters that
 457 minimizes $-\ln(\text{likelihood})$ for a given model, on a given data set. We can apply this method to
 458 simulated datasets, and compare the maximum likelihood estimators with the parameters used in
 459 the simulation.

460 Note that this function accepts a third, optional, parameter, that is the starting vector
 461 ‘params0’ given to optimize the likelihood. If no value is specified, the function takes the
 462 attribute ‘params0’ in the `PhenotypicModel` object.

In [232]: `fitTipData(modelBM, dataBM)`

*** Fit of tip trait data ***

Finds the maximum likelihood estimators of the parameters,

returns the likelihood and the inferred parameters.

Computation time : 0.02105212 secs

463 **Out [232] :**

464 **\$value** 13.3539168672421

465 **\$inferredParams m0** 0.112360024529455

466 **v0** 4.3703974585017e-08

467 **d** -0.0733871266399529

468 **sigma** 0.64761762031608

In [233]: fitTipData(modelOU, dataOU)

*** Fit of tip trait data ***

Finds the maximum likelihood estimators of the parameters,

returns the likelihood and the inferred parameters.

Computation time : 0.2915776 secs

469 **Out [233] :**

470 **\$value** 7.5162883379935

471 **\$inferredParams mean0** 13.5665180225751

472 **var0** 1.6815664554916e-05

473 **selectionStrength** 0.648513938633288

474 **equilibrium** 5.05532921748184

475 **noise** 0.766630199120977

476 It doesn't seem quite good, but it also seems like the choice in the starting parameters

477 m_0, v_0 has a bad influence. As presented in Appendix D.2, in many models (e.g. BM, OU,

478 ACDC, PM with $m_0 = \theta \dots$), distinct sets of parameters p_1 and p_2 are involved in the

479 computation of m and Σ , and the expectation vector m can be expressed as $m = Cp_1$. In

480 particular, many models verify $m = {}^{tr}(m_0, m_0, \dots m_0)$. When this is the case, the fit of tip data

481 can be improved and speeded up by using the third parameter of the function `GLSstyle=TRUE`.

```
In [234]: fitTipData(modelBM, dataBM, GLSstyle=TRUE)
          fitTipData(modelOU, dataOU, GLSstyle=TRUE)
```

```
*** Fit of tip trait data ***
```

```
Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.
Computation time : 0.03260899 secs
```

```
482 Out[234]:
```

```
483 $value 13.5302740469078
```

```
484 $inferredParams m0 -0.00550320295296933
```

```
485     v0 2.28469756397133e-07
```

```
486     d -0.313019528308928
```

```
487     sigma 0.663621107698308
```

```
*** Fit of tip trait data ***
```

```
Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.
Computation time : 0.1760004 secs
```

```
488 Out[234]:
```

```
489 $value 7.82305350777471
```

```
490 $inferredParams mean0 5.10957361631891
```

```
491     var0 3.36222531349288e-05
```

```
492     selectionStrength 1.87722870245168
```

```
493     equilibrium -1.98889519193151
```

```
494     noise 1.91905948952067
```

495 With so few data in hand, we could also prefer to consider directly models starting with
 496 $(m_0, v_0) = (0, 0)$. We create two new models ‘BM_from0’ and ‘OU_from0’ with the subtle

497 difference that $(m_0, v_0) = (0, 0)$ and the models thus retain respectively only two and three
498 parameters.

499 These two models are included in the 'ModelBank' file.

```
In [235]: modelBMfromZero <- createModel(tree, 'BM_from0')
          modelBMfromZero['paramsNames']
```

500 Out[235]: 'd' 'sigma'

```
In [236]: modelOUfromZero <- createModel(tree, 'OU_from0')
          modelOUfromZero['paramsNames']
```

501 Out[236]: 'psi' 'theta' 'sigma'

```
In [237]: fitTipData(modelBMfromZero, dataBM)
          fitTipData(modelOUfromZero, dataOU)
```

*** Fit of tip trait data ***

Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.

Computation time : 0.01061678 secs

502 Out[237]:

503 \$value 13.3540474589618

504 \$inferredParams d -0.0633929373190768

505 sigma 0.647501517840828

*** Fit of tip trait data ***

Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.

Computation time : 0.3246026 secs

506 Out[237]:

507 **\$value** 12.8406380062571

508 **\$inferredParams psi** -0.00173618484753718

509 **theta** -257.683877940727

510 **sigma** 0.598311115102137

511 While the first inference seems quite consistent, the second one is obviously wrong.

512 We would like here to warn users about the use of the `fitTipData` method. We did not
513 code an appropriate optimizer, and we use instead `optim`, the optimizer available in R, which
514 sometimes seems to be attracted to a wrong region in the parameter space. Starting the
515 optimization with different parameter sets might be the best practice to comfort the results.

516 For example, here, starting with another initial parameter set leads to a better likelihood
517 optimization.

```
In [238]: getDataLikelihood(modelOUfromZero, dataOU, c(1,5,1))
```

518 **Out[238]:** 7.17243577660605

```
In [239]: fitTipData(modelOUfromZero, dataOU, c(1,5,1))
```

```
*** Fit of tip trait data ***
```

```
Finds the maximum likelihood estimators of the parameters,  
returns the likelihood and the inferred parameters.
```

```
Computation time : 0.188132 secs
```

519 **Out[239]:**

520 **\$value** 6.69894523491705

521 **\$inferredParams psi** 9.38482325085794

522 **theta** 5.08087953439187

523 **sigma** 2.72996276707977

524 Finally, the functions `getTipDistribution`, `simulateTipData` and `fitTipData` all have a
525 last optional argument, called `v` for “verbose mode”. With `v=TRUE`, the functions gives
526 informations in the console, whereas with `v=FALSE` the function remains silent.

E.3 Toward an in-depth understanding of the code structure

This section can be skipped if you are not interested in using this framework to build your own model. Otherwise, it is worth understanding how the different models relate to each others.

Relationships between the different classes of models.—

The mother and most general class, for which all the above-mentioned functions are defined, is the `PhenotypicModel` class. When a model is only known as a `PhenotypicModel`, the method that computes the tip distribution, namely `getTipDistribution` is the most general one. It thus computes the distribution by resolving numerically the ODE system presented in main text Equations (4a, 4b), which can take a lot of time.

However, faster algorithms are available to compute the tip distribution under specific models (see e.g. analytical tip distribution formulas in Table S1). This is the rationale to define daughter-classes :

PhenotypicBM For the Brownian model.

PhenotypicOU For the Ornstein-Uhlenbeck model.

PhenotypicACDC For the Accelerating/Decelerating model.

PhenotypicDD For the Diversity-Dependent model.

PhenotypicPM For the Phenotype-Matching model.

PhenotypicGMM For the Generalist Matching Mutualism model.

PhenotypicADiag Models for which, $\forall i, A_i$ is symmetric and $\Gamma_i = \sigma I$.

For each of these daughter-classes, an other, more appropriated, function `getTipDistribution` has been written. `PhenotypicModels` which are also `PhenotypicOU`, will preferentially use methods defined for `PhenotypicOU` when they exist.

Application : three different ways to define an OU.—

In the `createModel` function, the keyword 'OU' constructs a model in the class

`PhenotypicOU`. In this class, the function `getTipDistribution` uses the analytical formula show in Appendix B.1 to speed up the computation of m and Σ .

Alternatively, the keyword 'OUbis' defines the exact same model, but as an instance of the class `PhenotypicADiag`. Thus, the function `getTipDistribution` uses the reduction show in Appendix C.1 to compute m and Σ .

Last, the keyword 'OUter' still defines the exact same model, but as an instance of the class `PhenotypicModel`. Thus, the function `getTipDistribution` uses the resolution of the ODE system to compute m and Σ .

The following lines of code show that the function returns the same value with the three different methods, but do not take the same amount of time.

```
In [240]: modelOU <- createModel(tree, 'OU')
          modelOUbis <- createModel(tree, 'OUbis')
          modelOUter <- createModel(tree, 'OUter')
          params <- c(0,0,0.2,1,2)
```

```
In [241]: getTipDistribution(modelOU, params, v=TRUE)
```

```
*** Computation of tip traits distribution through the analytical formula for an OU process ***
Computation time : 0.000497818 secs
```

```
Out [241]:
```

	A	0.8891968
	B	0.8775436
	C	0.909282
\$mean	D	0.8997412
	E	0.9137064
	F	0.8504314
	G	0.8946008

563 **\$Sigma**

	A	B	C	D	E	F	G
A	9.8772266	7.2724966	2.3654513	2.6142280	0.1171813	0.0000000	0.0000000
B	7.2724966	9.8500442	2.6142280	2.8891687	0.1295054	0.0000000	0.0000000
C	2.36545128	2.61422796	9.91770253	3.23775807	0.09593997	0.00000000	0.00000000
D	2.6142280	2.8891687	3.2377581	9.8994816	0.1060301	0.0000000	0.0000000
E	0.11718135	0.12950541	0.09593997	0.10603007	9.92553417	0.00000000	0.00000000
F	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	9.7762923	0.3657529
G	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.3657529	9.8889100

564

In [242]: `getTipDistribution(model0Ubis, params, v=TRUE)`

*** Computation of tip traits distribution through integrated formula ***

(Method working for models with a constant, A diagonalizable, and Gamma constant)

Computation time : 0.002770185 secs

565 **Out [242]:**

A	0.8891968
F	0.8504314
E	0.9137064
G	0.8946008
C	0.909282
D	0.8997412
B	0.8775436

566 **\$mean**

567 **\$Sigma**

	A	F	E	G	C	D	B
A	9.8772266	0.0000000	0.1171813	0.0000000	2.3654513	2.6142280	7.2724966
F	0.0000000	9.7762923	0.0000000	0.3657529	0.0000000	0.0000000	0.0000000
E	0.11718135	0.00000000	9.92553417	0.00000000	0.09593997	0.10603007	0.12950541
G	0.0000000	0.3657529	0.0000000	9.8889100	0.0000000	0.0000000	0.0000000
C	2.36545128	0.00000000	0.09593997	0.00000000	9.91770253	3.23775807	2.61422796
D	2.6142280	0.0000000	0.1060301	0.0000000	3.2377581	9.8994816	2.8891687
B	7.2724966	0.0000000	0.1295054	0.0000000	2.6142280	2.8891687	9.8500442

568

```
In [243]: getTipDistribution(modelOUter, params, v=TRUE)
```

```
*** Computation of tip traits distribution through ODE resolution ***
```

```
(Method working for any model)
```

```
Computation time : 0.01829243 secs
```

```
569 Out [243]:
```

A	0.8891984
F	0.8504309
E	0.9137081
G	0.8946024
C	0.9092837
D	0.8997429
B	0.8775447

```
571 $Sigma
```

	A	F	E	G	C	D	B
A	9.8772243	0.0000000	0.1171837	0.0000000	2.3654834	2.6142593	7.2725143
F	0.0000000	9.7762896	0.0000000	0.3657561	0.0000000	0.0000000	0.0000000
E	0.11718371	0.00000000	9.92553239	0.00000000	0.09594306	0.10603262	0.12950776
G	0.0000000	0.3657561	0.0000000	9.8889077	0.0000000	0.0000000	0.0000000
C	2.36548343	0.00000000	0.09594306	0.00000000	9.91769978	3.23780799	2.61425810
D	2.6142593	0.0000000	0.1060326	0.0000000	3.2378080	9.8994793	2.8891973
B	7.2725143	0.0000000	0.1295078	0.0000000	2.6142581	2.8891973	9.8500418

```
In [244]: dataOU <- simulateTipData(modelOU, c(0,0,0.2,1,2))
```

```
fitTipData(modelOU, dataOU)
```

```
fitTipData(modelOUBis, dataOU)
```

```
fitTipData(modelOUter, dataOU)
```

```
*** Simulation of tip trait values ***
```

```
Simulates step-by-step the whole trajectory, but returns only the tip data.
```

```
Computation time : 0.2363398 secs
```

```
*** Fit of tip trait data ***
```

Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.
Computation time : 0.1814284 secs

573 **Out [244] :**

574 **\$value** 15.0174906724384
575 **\$inferredParams m0** -26.3722559360675
576 **v0** 0.111663973605588
577 **psi** 0.0973609295443122
578 **theta** 14.9673044542728
579 **sigma** 1.12338425846849

*** Fit of tip trait data ***

Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.
Computation time : 0.7557919 secs

580 **Out [244] :**

581 **\$value** 15.0174906724384
582 **\$inferredParams m0** -26.3722559360675
583 **v0** 0.111663973605588
584 **psi** 0.0973609295443122
585 **theta** 14.9673044542728
586 **sigma** 1.12338425846849

*** Fit of tip trait data ***

Finds the maximum likelihood estimators of the parameters,
returns the likelihood and the inferred parameters.
Computation time : 6.088683 secs

587 **Out [244] :**

588 **\$value** 15.0174914969285

589 **\$inferredParams m0** -26.3722559360675

590 **v0** 0.111663973605588

591 **psi** 0.0973609295443122

592 **theta** 14.9673044542728

593 **sigma** 1.12338425846849

594 Focusing on the computation time, it is quite easily seen how interesting it can be to do
595 some more analytical work and write more appropriated `getTipDistribution` functions. Still,
596 the default function written for the mother class `PhenotypicModel` should always work.

597 *Using the framework to define a new model.—*

598 We illustrate here how the current code can be used to numerically study a specific model
599 that has not been implemented elsewhere. We focus here on the implementation of the ‘GMM’
600 model described in the main text, explaining step by step the following procedure, that is
601 generalizable to any model :

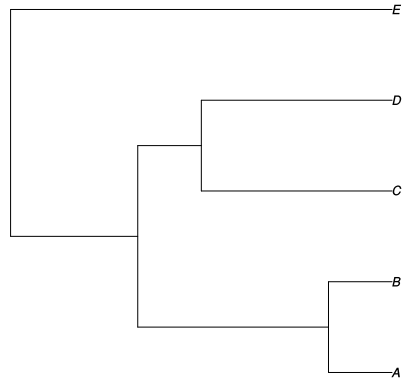
- 602 1. we identify what the periods are,
- 603 2. we write the model in a vectorial form on each period,
- 604 3. we implement it naively first,
- 605 4. we make analytical developments to speed up the computation time, and subsequently
606 introduce a new class more appropriated to this model.

607 For simplicity, we implement GMM for two ultrametric trees here. In our example, the
608 two trees will be :

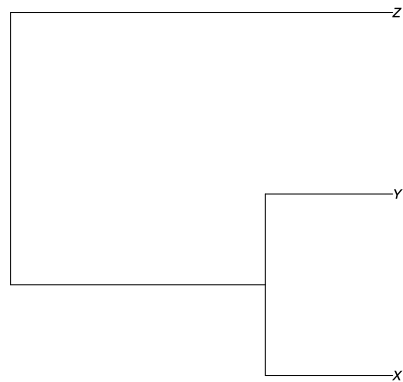
```
In [245]: newick1 <- "(((A:1,B:1):3,(C:3,D:3):1):2,E:6);"
          tree1 <- read.tree(text=newick1)
```

```
plot(tree1)

newick2 <- "(X:1.5,Y:1.5):3,Z:4.5);"
tree2 <- read.tree(text=newick2)
plot(tree2)
```



609



610

611 The first step consists in implementing a function `endOfPeriodsGMM(tree1, tree2)`,
612 which takes as input two trees (the trees corresponding to our two interacting clades), and
613 returns :

- 614 • the list of successive branching times (τ_i) (vector `periods`),
- 615 • information on which branch gives birth at that time (vector `copy`),

- the number assigned to the newly created branch at that time (vector `paste`),
- the number of lineages in clade 1 and 2 at each time (vectors `nLineages1` and `nLineages2`),
- the label of tips at the end (vector `labeling`).

For example, our function, called on the two preceding trees, returns :

```
In [246]: endOfPeriodsGMM(tree1, tree2)
```

```
Out [246]:
```

```
$periods 0  1.5  2  3  4.5  5  6
```

```
$copy 1  3  1  3  5  1  0
```

```
$paste 2  4  3  4  7  5  0
```

```
$nLineages1 2  2  3  4  4  5  0
```

```
$nLineages2 1  2  2  2  3  3  0
```

```
$labeling 'A' 'E' 'C' 'D' 'B' 'X' 'Z' 'Y'
```

The second step now consists in writing the model in the vectorial form required in the framework, on each epoch i . The form of the a , A and Γ matrices is shown in Appendix C.4, and depends on the number of lineages in the two clades on each epoch.

We introduce the constructor `createModelCoevolution(tree1, tree2)`, which is a function that takes as input two ultrametric trees corresponding to the two clades, and returns an object of class `PhenotypicModel`. It relies on the central function `aAGamma` that defines the collection of (a_i, A_i, Γ_i) on each epoch.

This first version of the GMM implementation allows us to simulate tip data, to get the tip distribution under any parameter set, and to fit tip data.

```
In [248]: modelGMMbis <- createModelCoevolution(tree1, tree2, keyword="GMMbis")
          modelGMMbis
```

Out [248]:

```
*****

*** Object of Class PhenotypicModel ***

*** Name of the model : [1] "GMMbis"

*** Parameters of the model : [1] "m0"      "v0"      "d1"      "d2"      "S"      "sigma"

*** Description : Generalist Matching Mutualism model.

Starts with 3 or 4 lineages having the same value  $X_0 \sim \text{Normal}(m0, v0)$ .

One trait in each lineage, all lineages evolving then non-independently
according to the GMM expression.

*** Periods : the model is cut into 7 parts.

For more details on the model, call : print(PhenotypicModel)
```

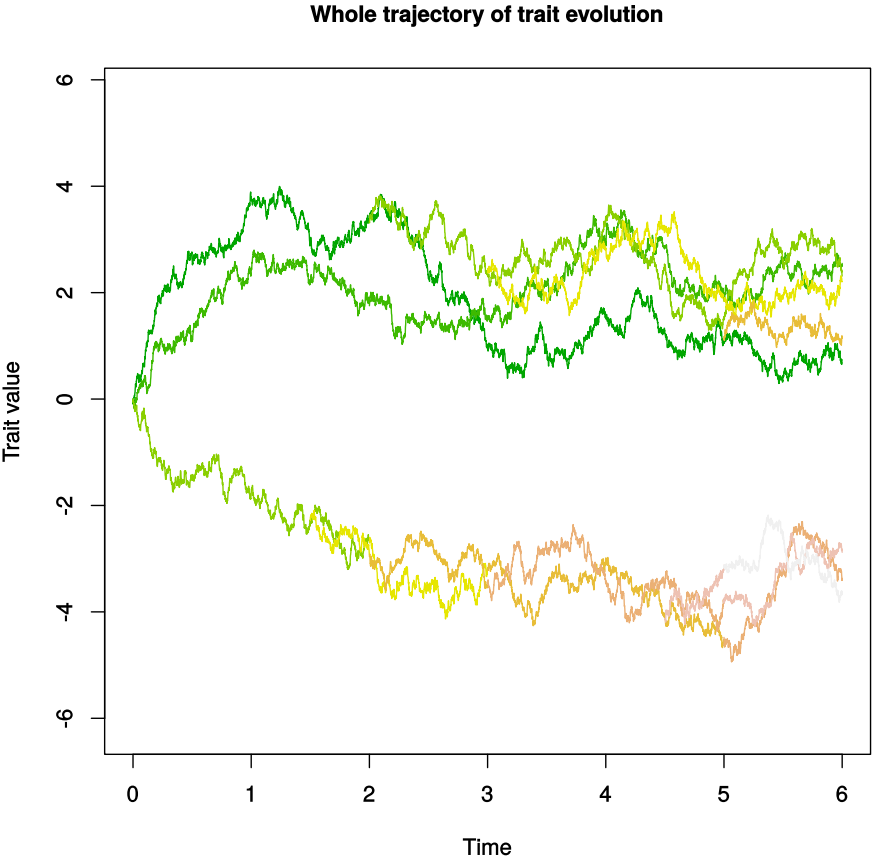
```
*****
```

In [249]: dataGMM <- simulateTipData(modelGMMbis, c(0,0,5,-5, 1, 1), method=2)

```
*** Simulation of tip trait values ***

Simulates step-by-step the whole trajectory, plots it, and returns tip data.

Computation time : 0.319762 secs
```



636

```
In [250]: getTipDistribution(modelGMMbis, c(0,0,5,-5,0.5,1))
```

637 Out [250]:

638 \$mean	A	2.493801
	E	2.493801
	C	2.493801
	D	2.493801
	B	2.493801
	X	-2.493801
	Z	-2.493801
	Y	-2.493801

		A	E	C	D	B	X	Z	Y
	A	2.196011	1.171214	1.215844	1.215844	1.563892	1.399735	1.341619	1.399735
	E	1.171214	2.141458	1.172730	1.172730	1.171214	1.337713	1.279597	1.337713
	C	1.215844	1.172730	2.199045	1.248832	1.215844	1.379237	1.321122	1.379237
639	\$Sigma D	1.215844	1.172730	1.248832	2.199045	1.215844	1.379237	1.321122	1.379237
	B	1.563892	1.171214	1.215844	1.215844	2.196011	1.399735	1.341619	1.399735
	X	1.399735	1.337713	1.379237	1.379237	1.399735	2.200083	1.190366	1.423215
	Z	1.341619	1.279597	1.321122	1.321122	1.341619	1.190366	2.158430	1.190366
	Y	1.399735	1.337713	1.379237	1.379237	1.399735	1.423215	1.190366	2.200083

```
In [251]: fitTipData(modelGMMbis, dataGMM, c(0,0,5,-5,1,1))
```

```
*** Fit of tip trait data ***
```

```
Finds the maximum likelihood estimators of the parameters,
```

```
returns the likelihood and the inferred parameters.
```

```
Computation time : 3.728739 secs
```

```
640 Out[251]:
```

```
641 $value 6.61385667009296
```

```
642 $inferredParams m0 0.00512480151380221
```

```
643     v0 2.69680996514239e-05
```

```
644     d1 5.03536962882004
```

```
645     d2 -5.83517142115953
```

```
646     S 0.231631941480316
```

```
647     sigma 0.361942471141108
```

648 However, this first implementation relies on the **PhenotypicModel** class, which uses the
649 method **getTipDistribution** that solves the ODE system on each epoch, and thus takes time.

650 The analytical reduction presented in Appendix C.4 can also be implemented. To this
651 end, we create a new class named **PhenotypicGMM**, associated with an other function

652 `getTipDistribution`. Using these developments allows us to compute more rapidly the tip
653 distribution under the model.

```
In [252]: modelGMM <- createModelCoevolution(tree1, tree2, keyword="GMM")
        modelGMM
```

Out [252]:

```
*****
*** Object of Class PhenotypicModel ***
*** Name of the model : [1] "GMM"
*** Parameters of the model : [1] "m0"      "v0"      "d1"      "d2"      "S"      "sigma"
*** Description : Generalist Matching Mutualism model.
Starts with 3 or 4 lineages having the same value  $X_0 \sim \text{Normal}(m_0, v_0)$ .
One trait in each lineage, all lineages evolving then non-independently
according to the GMM expression.
*** Periods : the model is cut into 7 parts.
For more details on the model, call : print(PhenotypicModel)
*****
```

```
In [253]: getTipDistribution(modelGMM, c(0,0,5,-5,0.5,1), v=TRUE)
        getTipDistribution(modelGMMbis, c(0,0,5,-5,0.5,1), v=TRUE)
```

```
*** Analytical computation of tip traits distribution ***
(Method working for the GMM model only)
Computation time : 0.0008528233 secs
```

654 Out [253]:

655	\$mean	A	2.493803							
		E	2.493803							
		C	2.493803							
		D	2.493803							
		B	2.493803							
		X	-2.493803							
		Z	-2.493803							
		Y	-2.493803							
656	\$Sigma		A	E	C	D	B	X	Z	Y
		A	2.196010	1.171213	1.215843	1.215843	1.563890	1.399736	1.341620	1.399736
		E	1.171213	2.141459	1.172730	1.172730	1.171213	1.337713	1.279597	1.337713
		C	1.215843	1.172730	2.199045	1.248832	1.215843	1.379238	1.321122	1.379238
		D	1.215843	1.172730	1.248832	2.199045	1.215843	1.379238	1.321122	1.379238
		B	1.563890	1.171213	1.215843	1.215843	2.196010	1.399736	1.341620	1.399736
		X	1.399736	1.337713	1.379238	1.379238	1.399736	2.200083	1.190366	1.423213
		Z	1.341620	1.279597	1.321122	1.321122	1.341620	1.190366	2.158430	1.190366
		Y	1.399736	1.337713	1.379238	1.379238	1.399736	1.423213	1.190366	2.200083

*** Computation of tip traits distribution through ODE resolution ***

(Method working for any model)

Computation time : 0.01734638 secs

657 **Out [253] :**

658	\$mean	A	2.493801
		E	2.493801
		C	2.493801
		D	2.493801
		B	2.493801
		X	-2.493801
		Z	-2.493801
		Y	-2.493801

	A	E	C	D	B	X	Z	Y
A	2.196011	1.171214	1.215844	1.215844	1.563892	1.399735	1.341619	1.399735
E	1.171214	2.141458	1.172730	1.172730	1.171214	1.337713	1.279597	1.337713
C	1.215844	1.172730	2.199045	1.248832	1.215844	1.379237	1.321122	1.379237
659 \$Sigma D	1.215844	1.172730	1.248832	2.199045	1.215844	1.379237	1.321122	1.379237
B	1.563892	1.171214	1.215844	1.215844	2.196011	1.399735	1.341619	1.399735
X	1.399735	1.337713	1.379237	1.379237	1.399735	2.200083	1.190366	1.423215
Z	1.341619	1.279597	1.321122	1.321122	1.341619	1.190366	2.158430	1.190366
Y	1.399735	1.337713	1.379237	1.379237	1.399735	1.423215	1.190366	2.200083