



**HAL**  
open science

# Shifted stochastic processes evolving on trees : application to models of adaptive evolution on phylogenies.

Paul Bastide

► **To cite this version:**

Paul Bastide. Shifted stochastic processes evolving on trees : application to models of adaptive evolution on phylogenies.. Statistics [math.ST]. Université Paris Saclay (COMUE), 2017. English. NNT : 2017SACLS370 . tel-01629648

**HAL Id: tel-01629648**

**<https://theses.hal.science/tel-01629648v1>**

Submitted on 6 Nov 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

NNT : 2017SACLS370

THÈSE DE DOCTORAT  
DE L'UNIVERSITÉ PARIS-SACLAY  
PRÉPARÉE À L'UNIVERSITÉ PARIS-SUD

Ecole doctorale n°574  
Ecole doctorale de mathématiques Hadamard  
Spécialité de doctorat : Mathématiques appliquées  
par  
**PAUL BASTIDE**

Modèles de processus stochastiques avec sauts sur arbres :  
application à l'évolution adaptative sur des phylogénies

*Shifted stochastic processes evolving on trees: application to  
models of adaptive evolution on phylogenies*

Thèse présentée et soutenue à Paris, le 19 octobre 2017.

Composition du Jury :

Mme.	ELISABETH GASSIAT	Professeure Université Paris Sud	(Examinatrice)
M.	AMAURY LAMBERT	Professeur Université Pierre et Marie Curie	(Président du Jury)
M.	PHILIPPE LEMEY	Principal Investigator Katholieke Universiteit Leuven	(Rapporteur)
M.	MAHENDRA MARIADASSOU	Chargé de recherche INRA	(Directeur de thèse)
Mme.	HÉLÈNE MORLON	Directrice de recherche CNRS	(Examinatrice)
M.	PIERRE PUDLO	Professeur Aix-Marseille Université	(Rapporteur)
M.	STÉPHANE ROBIN	Directeur de recherche INRA	(Directeur de thèse)



Quelqu'un tisse de l'eau (avec des motifs d'arbres  
en filigrane). Mais j'ai beau regarder,  
je ne vois pas la tisserande,  
ni ses mains même, qu'on voudrait toucher

Quand toute la chambre, le métier, la toile  
se sont évaporés,  
on devrait discerner des pas dans la terre humide...

Philippe JACCOTTET  
On voit, in *Pensées sous les nuages*  
nrf, Éditions Gallimard (1976 - 1983 - 1994)







# Contents

<b>Introduction</b>	<b>9</b>
<b>1 Background</b>	<b>17</b>
1.1 Space of Convex Characters on Phylogenies . . . . .	18
1.2 Latent Variable Models of Character Evolution . . . . .	25
1.3 Discrete Models of Evolution and Tree Reconstruction . . . . .	30
1.4 Continuous Models of Evolution . . . . .	37
1.5 Model Selection . . . . .	58
Appendices . . . . .	71
1.A The Ornstein-Uhlenbeck Process . . . . .	71
1.B Multivariate Analysis Tools . . . . .	74
<b>2 Shift Detection for Univariate Processes</b>	<b>77</b>
2.1 Introduction . . . . .	78
2.2 Statistical Modeling . . . . .	81
2.3 Identifiability and Complexity of a Model . . . . .	85
2.4 Statistical Inference . . . . .	92
2.5 Simulations Studies . . . . .	96
2.6 Case Study: Chelonian Carapace Length Evolution . . . . .	100
Appendices . . . . .	104
2.A Enumeration of Equivalence Classes . . . . .	104
2.B A Vandermonde Like Identity . . . . .	105
2.C Technical Details of the EM . . . . .	108
2.D Optimal Shift Location with Fixed Root . . . . .	111
2.E Proof of Proposition 2.4.1 for Model Selection . . . . .	112
2.F Supplementary Figures . . . . .	113
2.G Practical Implementation . . . . .	117
<b>3 Shift Detection for Multivariate Processes</b>	<b>119</b>
3.1 Introduction . . . . .	120
3.2 Model . . . . .	122
3.3 pPCA and Shifts . . . . .	128
3.4 Simulations Studies . . . . .	129
3.5 Examples . . . . .	136
3.6 Discussion . . . . .	140
Appendices . . . . .	143
3.A PCA: Mathematical Derivations . . . . .	143
3.B PhylogeneticEM case study: New World Monkeys . . . . .	143



3.C	EM Inference . . . . .	146
3.D	Simulations Appendices . . . . .	153
<b>4</b>	<b>Trait Evolution on Phylogenetic Networks</b>	<b>159</b>
4.1	Introduction: Phylogenetic Networks . . . . .	160
4.2	Continuous Trait Evolution on a Network . . . . .	164
4.3	Tests of Phylogenetic Signal . . . . .	174
4.4	The <i>Julia</i> package <i>PhyloNetworks</i> . . . . .	180
4.5	Perspectives . . . . .	181
	Appendices . . . . .	183
4.A	Documentation: Continuous Trait Evolution . . . . .	183
4.B	Decomposition of the Covariance Matrix . . . . .	190
<b>5</b>	<b>Extensions and Perspectives</b>	<b>197</b>
5.1	Dealing with Tree and Trait Uncertainty . . . . .	197
5.2	Convergence and Sparsity . . . . .	203
5.3	Non-Ultrametric Trees . . . . .	205
5.4	Sampling Scheme and Missing Data . . . . .	212
<b>6</b>	<b>Résumé substantiel</b>	<b>215</b>

# Introduction

Evolutionary ecology is interested with the astonishing diversity of life forms on Earth. Not mentioning the immeasurable gap that might exist between, for instance, *Pyrolobus fumarii*, a thermophilic archeal species that lives around the very hot hydrothermal vents of the Atlantic mid-ocean ridge (Blöchl et al., 1997) and *Cypripedium calceolus*, a “dainty and charming, trembling and delicate” orchid species (Huysmans, 1930), diversity can be seen at every scale of the tree of life, from the domain to the family or genus level. *Coccinellidae* (lady bugs) are for example known to wear a great variety of body patterns, from the common Europeans *Coccinella septempunctata*, red with seven black dots, and *Psyllobora vigintiduopunctata*, yellow with twenty two black dots, to the American *Brachiacantha ursina*, black with ten yellow dots<sup>1</sup>.

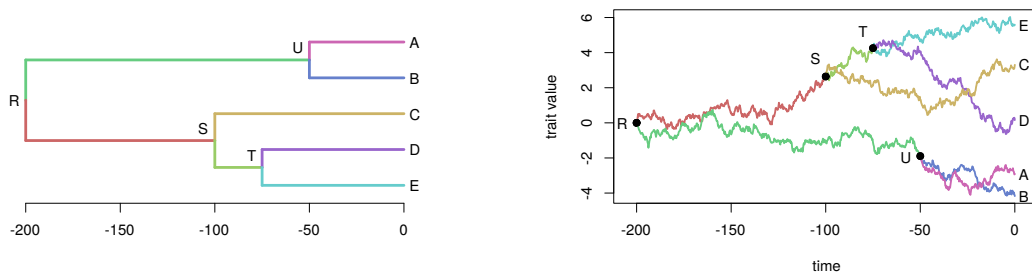


When observing a set of species, it is natural to wonder how a given trait pattern arose in the population. How much of it can be explained by external causes, such as climate or habitat? Is it possible that some of it is just the result of “chance”? The answers to these questions highly depend on the meaning one puts behind this word. Using a very narrow definition of chance, 18<sup>th</sup> centuries philosophers famously concluded in the necessary existence of a “watchmaker” that shaped every life form (see e.g. Rousseau, 1762b; Paley, 1802). If not convinced by this “Intelligent Design” theory, one might assume that all the traits of the species arose randomly, independently from one another. Although simple and quite intuitive, such an assumption is often wrong, and can lead to misleading conclusions. Indeed, it completely ignores the fact that living species are *not independent*. According to the theory of evolution (see e.g. Darwin, 1859, for an introduction), these species are linked by a *phylogenetic tree*, that represents the family relationships between them. When looking at species traits, it is then natural to expect that two closely related species, i.e. species that diverged only a short time ago, should look more alike than two distantly related species. Making explicit what a trait distribution produced by chance alone looks like is one of the goals of *Phylogenetic Comparative Methods*, the framework we use in this thesis to study trait evolution. It can be seen as a way to specify a correct *null model*.

## Phylogenetic Comparative Methods

Phylogenetic Comparative Methods (PCM) stem from the idea that, if we know, first, a dated phylogenetic tree between species, that tells us when speciation events took place, and, second, a dynamic model of quantitative trait evolution that describes how the traits change in time, then we should be able to find the expected current trait distribution in the species population we are studying. In this framework, assuming that the phylogenetic tree is known, “chance” is then entirely defined by the model of trait evolution, usually chosen in a family of *stochastic processes*. Such a model allows us to *quantify* the variations of the modeled traits. The simplest stochastic process one might use is the *Brownian Motion* (BM). Under this model of evolution, trait values have no trend, with independent and Gaussian increments.

The global model is then obtained by putting together the dated phylogenetic tree and the process of evolution in the following way. The trait of a given ancestral species evolves in time on a branch of the tree according to a BM. When a speciation event occurs, the two children species inherit their mother’s trait, and then carry on evolving independently as a two BM. Note that, even though the two children species are supposed to evolve independently, the simple fact that they inherited their trait from the same species introduces some correlations between them. To see this, assume for instance that, by chance, their mother species deviated to extremely large values of the trait. Then, the two children species will start their evolution time with very large values, and hence are likely to look alike for a while, being larger than most of the other species. If the process is a BM, then these correlations can be quantified easily: the covariance between the traits of two extant species is proportional to their time of shared evolution (i.e. the time elapsed between the root of the tree and their most recent common ancestor).



(a) A dated phylogenetic tree. The vertical positions of the tips are arbitrary.

(b) BM on the branches of the tree. The vertical axis gives the value of the trait.

Realization of a univariate BM process on a dated phylogenetic tree. The colors of the branches (left) match with the colors of the distinct processes (right). For instance, the ancestral red species has a trait that evolves from 0 (node *R*, time  $-200$ ) to around 2.6 (node *S*, time  $-100$ ). Only tip values are observed (at time  $t = 0$ ). Purple and blue species (*A* and *B*) inherited their trait values quite recently from the green species (*U*). They had little time to diverge away from this value, so they are likely to have similar trait values. Their trait values are marginally correlated (but not anymore when conditioning on *U*).

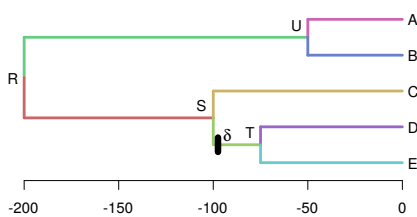
The art of PCM is then to define processes that correctly describe the dynamic evolution of the trait, and then to study the kind of trait distribution it produces at the tips of the tree, for living, observed species.

## Shift Detection

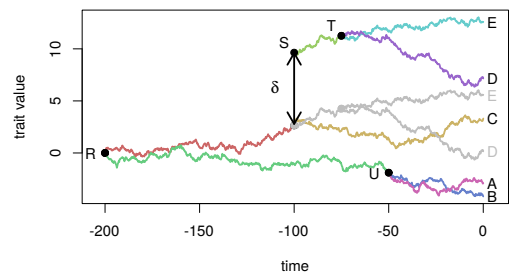
This resulting trait distribution at the tips can be seen as a null model, that describes the correlations between species, and the ranges of variations of the traits, that can be expected by “chance”. If the observed trait distribution significantly differs from the expected one, then it might be the clue that some special events shaped the history of the trait.

In this thesis, we are particularly interested in *shifts* that might happen on the trait at some points of species histories. Such a shift might happen for several biological reasons, such as a migration to a new environment, or a rapid climate change. When a species experiences such a shift, then it passes along its new value to its offspring, so that all its descending species inherit from this change. Such a situation is shown in the next figure.

Of course, several shifts can happen on the tree. The main goal of this thesis is to find their number and location on the tree. Both questions raise serious statistical issues, as shown in the next two paragraphs.



(a) A dated phylogenetic tree.



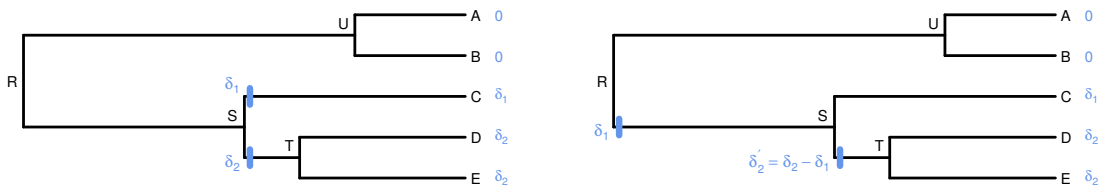
(b) BM on the branches of the tree.

Same process, but with a shift on the light green branch (between *S* and *T*). Tips *D* and *E* are affected by the ancestral shift: their trait value is much larger than the one expected without any shift (in grey).

## Identifiability

It is important to bear in mind that, even if we defined a dynamic model of trait evolution in time, we only have access to the state of the system today, for *extant* species. The stochastic process is hence only seen through its last values on the tree, at only one time point. Such a situation is bound to produce *identifiability issues*.

Adding shifts to the process only makes the problem worse. As shown in the next figure, it is easy to see that two different scenarios, with shifts of different values happening on different branches, can produce the exact same trait distribution at the tips of the tree. Such scenarios, although biologically distinct, cannot be distinguished from data collected only at the tips. They are not identifiable. Quantifying such identifiability problems is important both from a statistical point of view, as ignoring these problems would lead to ill-defined models, and from a biological point of view, as it allows us to know what the collected data can – and cannot – say about the past history of the trait.

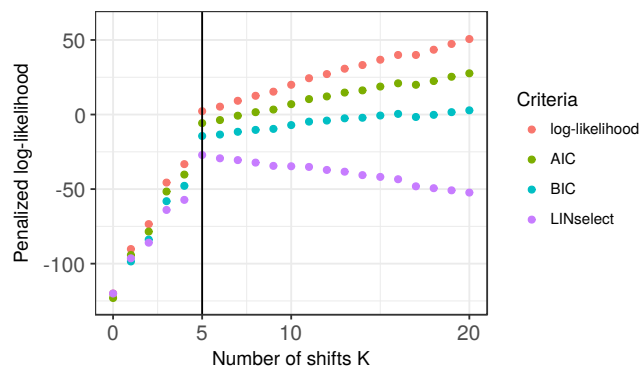


Two equivalent scenarios. The trait is assumed to evolve as a shifted BM, with ancestral value 0, and shifts marked on the branches. Expectations at the tips are indicated in blue. Both scenarios produce the same trait distribution, but they tell a different biological story. On the left one, the two children species of S went through independent traumatic events that led to shifted trait values. On the right one, ancestors of S first went through a shift, that is passed along to its first child C directly, while its second child experiences another compensatory shift.

## Model Selection

After addressing the identifiability problem, we will show how we can find the best scenario, for a given number  $K$  of shifts. Among all the models with  $K$  shifts, we are able to find the one with the highest likelihood. However, the best fitting model with  $K + 1$  shifts, as it has more parameters, is known to have a better likelihood score. Following the same logic, the scenario with the best likelihood will always be the one with one shift per species, defining a special regime for each of them. This is a problem, as we would like to keep only *significant* shifts, that are typically rather scarce.

One common way to solve this problem is to use a criterion based on *penalized* likelihood, with a penalty that increases with the number of shifts. The penalty needs to be carefully crafted, so that it only compensates the increase of the likelihood due to overfitting. The next figure shows the principle of this method. It also stresses out the importance of the choice of the penalty to use: on the simple example presented, both AIC and BIC, two very commonly used penalized criteria, clearly fail to correct for over-fitting. A theoretically sound penalty, that takes into account the true size of usable models for a given number of shifts, will be derived.



Typical profile for a maximum likelihood inference of parameters for a dataset simulated using a BM on a tree with 64 taxa, and  $K_{\text{true}} = 5$  shifts. In red, each dot is the maximum likelihood obtained for a model with a given number  $K$  of shifts allowed. This likelihood is always increasing, as expected. A penalized criterion that is correctly working should have a maximum in  $K = 5$  (vertical line), allowing us to infer the true number of shifts. Standard penalized criterion AIC and BIC (green and blue) fail to recover this true number of shifts (they are increasing). In contrast, the proposed model selection method (LINselect, purple) behaves correctly: it has a clear maximum for  $K = 5$ .

These models and broad statistical methods are at the core of this thesis. In the main text, we explain them in more details (Chapter 1), develop and extend them to other stochastic processes (Chapter 2 and Chapter 3), and to other kinds of species relationships (Chapter 4).

## Outline

### Chapter 1: Background

The first chapter aims at presenting in a structured way all the tools and concepts used in the rest of the manuscript, both from a biological and a statistical point of view. We start by reviewing some structural results that have to do with the combinatorial nature of the space of objects we consider: traits mapped on phylogenetic trees. We then introduce several explicit models of trait evolution, both discrete and continuous. Finally, we give a brief overview of some statistical methods for model selection based on penalized likelihood. So as to present the reader with a coherent overview of the problems at stake, some of the results presented in this chapter are also developed further in the next two chapters, which correspond to published or submitted journal articles.

### Chapter 2: Shift Detection for Univariate Processes

In this chapter, we set up the entire method for shift detection in the univariate case. We start by defining a rigorous statistical framework to model shifts on the tree, for both the Brownian Motion and the Ornstein-Uhlenbeck processes. Building on the combinatorial results exposed in the previous chapter, and thanks to a careful analysis of the models, we give an extensive solution to the identifiability problem. This implies two different questions. First, given a shift scenario, we are able to enumerate and count all the different scenarios that produce the same trait distribution at the tips, i.e. that are *equivalent*. Second, given a fixed number of shifts  $K$ , we can compute the number of equivalent classes of models with  $K$  shifts, that is the number of truly different models, that give different trait distribution structures at the tips of the tree. Except for binary trees (i.e. a tree where all internal nodes have exactly two children), this number depends on the topology of the tree at hand.

We then describe a statistical inference method, that relies on a two steps strategy. First, for any given number  $K$  of shifts, we implement an Expectation Maximization (EM) algorithm that allows us to find the maximum likelihood solution. Second, given all the solutions found over a range of values of  $K$ , we derive a model selection penalty to choose the right number of shifts. This penalty is based on the LINselect procedure, and inherits some of its properties and theoretical guaranties. In particular, the penalty can be used directly, without the need to calibrate a multiplying constant, and an oracle inequality can be derived in some particular cases.

The method is efficiently implemented in an R package `PhylogeneticEM`. Its accuracy is assessed through a set of extensive simulation scenarios. It is then used to study the evolutionary history of the *Chelonians*, a family containing all living turtles and tortoises.

### Chapter 3: Shift Detection for Multivariate Processes

In this chapter, we build upon the univariate case to address multivariate traits. We introduce a new set of assumptions on the Ornstein-Uhlenbeck process, that allows us

to explicitly deal with correlated traits, while still being able to conduct shift detection in an efficient way. Compared to previous state-of-the-art methods, that all assumed independent traits, this is a salient feature of our framework. It is all the more important as we show that a common method to de-correlate traits measured at the tips of a tree, called phylogenetic PCA (pPCA) actually fails to do so when there are some shifts in the traits evolution.

The EM likelihood maximization is made efficient thanks to a new “upward-downward” algorithm, that can compute all the quantities needed at the E step with only two traversals of the tree, and can cope with missing data. This algorithm is implemented in C++, and the method is integrated in the R package *PhylogeneticEM*.

The efficiency and accuracy of the method is assessed through an extensive set of simulations, designed to test the behavior of the framework when its assumptions are violated. The method is then used to study the evolutionary history of *new world monkeys* and *anolis lizards*.

## Chapter 4: Trait Evolution on Phylogenetic Networks

In this chapter, we explore a new paradigm for species evolution: phylogenetic networks. Compared to phylogenetic trees, phylogenetic networks can have some *hybridization* events. Instead of being always *vertical* (i.e. from a species to its offspring), the genetic transmission can sometimes be *horizontal* (i.e. between two contemporary species). Such events are known to happen from time to time in every families of species, and are even quite common in some, such as bacterial organisms or plants.

We start by giving a swift overview of state of the art methods to reconstruct such networks from DNA sequences. They are all quite recent, and promised to a bright future. We then show how these newly inferred structures can be used to describe trait evolution. A simple BM model is fully described, and an efficient algorithm to compute the trait distribution structure it induces at the tips is derived.

From this first null model of trait evolution on a network, we again study the impact that shifts might have on the trait distribution. Here, we limit ourselves to very particular shifts, that follow hybridization events. Such shifts can model *heterosis* (hybrid vigor or depression). This phenomenon is well known by agronomists, who for instance use it to improve cultivated lineages of crops. It describes the fact that, sometimes, a hybrid individual exhibits a trait with an outstanding value, that is outside of the range of its two parents trait values. The shifts used to model heterosis have a fixed and known position (on branches just below hybrid nodes), so that, contrary to the previous problem, we do not need to search the entire tree to find potential shifts. We show how a classical linear regression framework, along with a standard Fisher test for the coefficients, can inform us on the presence or absence of heterosis.

The methods developed in this chapter are integrated in the *Julia* package *PhyloNetworks*. *Julia* is a new programming language, that aims at combining the ease of use of *R*, and the speed of *C*. The *PhyloNetworks* package has for ambition to become the standard tool for inferring, analyzing, and manipulating phylogenetic networks on *Julia*.

## Chapter 5: Extensions and Perspectives

In this last chapter, we browse some of the extensions that could be interesting to study in future work. Models are only sketched in this chapter, that aims at providing the main ideas behind each extension possibility.



We first study the impact of uncertainties on our framework. Indeed, in all our developments, we always assumed that the data at the tips was measured without error, and that the underlying tree was perfectly known. Both these assumptions are erroneous, and we show that ignoring these sources of uncertainties can bias our analysis. A simple method to deal with measurement error is described. Interestingly, we show how this same framework could be used to conduct a factor analysis of the traits.

We then see how the framework could be adapted to include two desirable features: convergence, and shift trait sparsity. Structural penalties can allow us to deal with these new constraints. Two distinct sets of species are said to be *convergent* if they developed the same trait features independently. This is not allowed in our previous framework, and we propose to include it using a fused-ANOVA penalty. In the multivariate case, for simplicity reasons, we assumed that when a shift occurred, it affected all traits at once. This is a strong assumption, that makes the analysis highly dependent on the set of traits included in – or excluded from – the analysis. A sparse group sparse lasso penalty is proposed to tackle this issue, and impose a sparse number of traits to change at each shift.

Another strong assumption we made was that the trees we considered were *ultrametric*: we assumed that all the trait measurements we had were coming from contemporary, presently living species. This excludes for instance any kind of fossil measurement. Although rare, these fossils provide us with unique insights into the evolution process, and should not be ignored. Alleviating the ultrametry assumption is however not straightforward. First, it breaks our careful identifiability study. Biologically, this is a good thing, as previously un-distinguishable scenarios can become identifiable again. Mathematically, however, it makes the space of models more difficult to study, and highly dependent on the tree topology. Second, it prevents us from using a re-scaling trick, that allowed us to carry efficient computations on the Ornstein-Uhlenbeck. Some new heuristics are proposed to deal with this issue.

Finally, the last section presents a method to deal with missing data in a more satisfying way. Indeed, we assumed in our framework that data were missing completely at random, i.e. that the sampling was uniform over all traits and species, with missingness happening randomly from time to time. However, the actual sampling might sometimes have a specific structure, that depends on the very data being measured. A simple example of this would be for an experimenter to be more likely to “miss” a small trait value than a larger one. Such sampling scheme can be explicitly incorporated in the statistical analysis, and we sketch the changes that would be needed in our framework to include them.

## Chapitre 6: Résumé substantiel

Ce dernier court chapitre, en français, décrit le contexte et les résultats principaux présentés dans ce document. Il peut être lu de manière autonome, et est fortement redondant avec la présente introduction, qui s’achève sur ces mots.

---

<sup>1</sup>Photo Credits:

- Photographer: Dominik Stodulski, Graphic Processing: MathKnight - File:BIEDRONA.JPG in WikiCommons, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=35283266>  
 - CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=338902>  
 - Smidon33 — Personal Work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=17343450>





# Chapter 1

## Background

### Contents

---

<b>1.1</b>	<b>Space of Convex Characters on Phylogenies</b>	<b>18</b>
1.1.1	Phylogenetic Tree	18
1.1.2	Convex Characters	20
1.1.3	Parsimony	22
<b>1.2</b>	<b>Latent Variable Models of Character Evolution</b>	<b>25</b>
1.2.1	A Generic Model for Trait Evolution on a Tree	25
1.2.2	Directed Graphical Models and Latent Variables	27
1.2.3	Pruning Algorithm	29
1.2.4	Expectation Maximization	29
<b>1.3</b>	<b>Discrete Models of Evolution and Tree Reconstruction</b>	<b>30</b>
1.3.1	Continuous Time Markov Chains	30
1.3.2	Models of DNA Evolution	33
1.3.3	Molecular Clocks	35
1.3.4	Tree Reconstruction	37
<b>1.4</b>	<b>Continuous Models of Evolution</b>	<b>37</b>
1.4.1	Models of Evolution	38
1.4.2	Phylogenetic Comparative Methods	42
1.4.3	Detecting Shifts	50
1.4.4	Algorithms for Likelihood Computation	53
1.4.5	Extensions	55
<b>1.5</b>	<b>Model Selection</b>	<b>58</b>
1.5.1	Penalized Likelihood	58
1.5.2	Model selection <i>à la</i> <a href="#">Birgé &amp; Massart (2001)</a>	62
1.5.3	The Lasso Penalty	65
1.5.4	Model selection with LINselect	68
<b>Appendices</b>		<b>71</b>
<b>1.A</b>	<b>The Ornstein-Uhlenbeck Process</b>	<b>71</b>
1.A.1	Stochastic Differential Equation and General Solution	71
1.A.2	Induced Variance Structure	71
1.A.3	Incomplete Data Formulation	74
<b>1.B</b>	<b>Multivariate Analysis Tools</b>	<b>74</b>
1.B.1	Kronecker Product and Vectorization	74
1.B.2	Hadamard Product	76

---

In this chapter, we introduce some of the theoretical results that are at the root of the developments exposed in the following of this document. We try to provide the interested reader with a coherent overview of both the biological and statistical literature, and to stress the important questions faced by the field.

Three main points are covered in this introduction. This manuscript is concerned with trait evolution on phylogenetic tree. Before describing any model, we first give a rigorous definition of phylogenetic trees as a mathematical object, and describe how a character can be mapped on it (Section 1.1). This allows us to study some of the combinatorial properties of the space of problems we will be navigating through.

In the next three sections, we introduce some dynamic models of trait evolution on a tree. We first show how a generic model can be defined, and display some of its global properties (Section 1.2). We then describe in more depths two important instances of this model, for discrete (Section 1.3) and continuous (Section 1.4) traits. We gave a special care to that last section, that is at the core of our work.

Finally, we recall some important results about model selection (Section 1.5), that will be useful in the statistical analysis of the problem. This section can be read independently.

## 1.1 Space of Convex Characters on Phylogenies

In this section, we give a definition of phylogenetic trees, and show how to track and count the changes of a discrete character state evolving on the tree. The exposition and theorems are inspired from these two landmark books: [Felsenstein \(2004\)](#) and [Semple & Steel \(2003\)](#). The combinatorial results recalled in this section are at the root of our attempts to assess the identifiability of the models we considered in this thesis.

### 1.1.1 Phylogenetic Tree

We formally introduce here phylogenetic trees, that are one of the central objects of this thesis. Trees are often used in evolutionary biology to describe the relationships between extant species. They can be defined as follow (we assume in this definition that the reader is familiar with the basics of graph theory, and refer to [Giraud 2014](#), Chap. 7, for a brief introduction of the notions needed).

**Definition 1.1.1** (Tree). A *tree*  $T = (V, E)$  with a set of vertices (or nodes)  $V$  and edges  $E$  is a connected acyclic graph.

The *leaves* (or *tips*) of the tree are the vertices of degree one, and all the other nodes are said to be *interior* or *ancestral*.

A *binary* or *fully resolved* tree is a tree where every interior node has degree exactly 3, except for at most one with degree 2 (the root, if any). In non-binary trees, a node with degree more than 3 is called a *polytomy*.

Binary trees are preferred, when possible, as they represent a situation where, taking any extant species, we can sort out all the others, from the most closely related, to the most distant one. In many applications, we want the tree to give us information not only on the relationships between species, but also on the relative position of their ancestors. Specifically, we often need a *rooted* tree, where an interior node is marked to be the ancestor of all other nodes. An example of such a tree is given [Figure 1.1.1](#).

**Definition 1.1.2** (Rooted Tree). A tree can be *rooted* if one interior node is distinguished as the root. On a binary tree, the root is the only node with degree 2.

A rooted tree can be oriented from the root to the leaves. For any node  $a \in V$ , we then denote by  $\text{anc}(a)$  the set of all *ancestors* of  $a$ :  $b \in \text{anc}(a)$  if and only if  $b = a$  or

there is an oriented path  $b \rightarrow a$ . The direct parent of  $a$  is the unique node  $\text{pa}(a)$  such that  $\text{pa}(a) \in \text{anc}(a)$ , and  $(\text{pa}(a), a) \in E$ .

Similarly, we define the set  $\text{des}(a)$  of *descendants* of  $a$ :  $b \in \text{des}(a)$  if and only if  $a \in \text{anc}(b)$ . The set of direct descendants of  $a$  are the *children* of  $a$ :  $b \in \text{child}(a)$  if and only if  $a = \text{pa}(b)$ .

On rooted trees, several natural orders of the nodes can be defined. These orders are useful to compute some quantities iteratively and efficiently on the tree.

**Definition 1.1.3** (Pre and Post Orders). The nodes of the tree are said to be sorted in a *preorder* if any node comes after all its parents: for any two nodes numbered  $i$  and  $j$ , if there is an oriented path going from  $i$  to  $j$ , then  $i \leq j$ . This corresponds to a numbering of the tree from the root to the leaves. A preorder is a particular case of *topological sorting* when the graph is a tree, and can be obtained in linear time (Kahn, 1962).

A preorder can be turned into a *postorder* by reversing it. A postorder can also be called a *pruning* order, as it is used in the well known Felsenstein pruning algorithm (see Section 1.2.3).

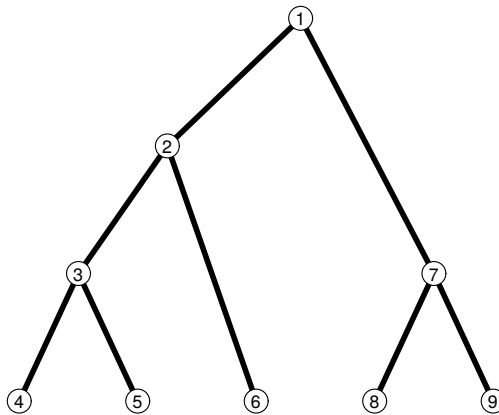


Figure 1.1.1 – A rooted binary tree with 5 tips (nodes  $\{4, 5, 6, 8, 9\}$ ) and 4 internal nodes (nodes  $\{1, 2, 3, 7\}$ ). The root is node 1. In addition, we have:  $\text{pa}(3) = \{2\}$ ,  $\text{anc}(3) = \{1, 2, 3\}$ ,  $\text{child}(3) = \{4, 5\}$  and  $\text{des}(3) = \{3, 4, 5\}$ . The nodes are numbered in a preorder. When visiting the nodes in this order, we know when we see a new node that we already visited all its ancestors. A postorder is obtained by reversing this order.

As stated above, trees are used in evolutionary biology to represent the relationships between a set of extant species, that are typically found at the tips. We hence need a map linking the observed species (each identified by a label), and the tips of a tree. Following Semple & Steel (2003, def. 2.1.2), we define a *phylogenetic tree* as a pair of a tree, and a set of associated labels.

**Definition 1.1.4** (Phylogenetic Tree). A *phylogenetic tree (on  $X$ )*  $\mathcal{T}$  is a pair  $(T, \phi)$ , where  $T = (V, E)$  is a tree, and  $\phi : X \rightarrow L$  is a bijection between a set of *labels*  $X$  and the set  $L$  of the leaves of  $T$ .

The set  $X$  can be viewed as a set of labels that uniquely define the leaves of the tree. In the context of evolutionary biology, these can be thought as the names of the extant species studied. For each species, we can then measure a set of discrete traits, such as the presence of absence of a given feature, or, more commonly, a molecular sequence. The distribution of these characters, that are the result of evolution for all extant species, are likely to be constrained by the tree. In the following section, we recall some classical results on the class of *convex characters*, that are characters that can be seen as the result of a homoplasy-free evolution along the tree (see definitions below).

### 1.1.2 Convex Characters

These notions and properties are analysed in Chapter 4 of [Semple & Steel \(2003\)](#). We recall here some of the main definitions and results, that will be useful in Section 2.3 of Chapter 2, to characterize the set of equivalent solutions.

**Definition 1.1.5** (Character). A discrete *full character*  $\chi : X \rightarrow C$  is an application mapping the leaves (identified by their labels) of a phylogenetic tree  $\mathcal{T} = (T, \phi)$  on  $X$  to a *character* or *state* set  $C$ . If  $|\chi(X)| = r$ , then  $\chi$  is an *r-state character*.

**Definition 1.1.6** (Extension). An *extension* of a character  $\chi$  to  $\mathcal{T}$  is an application  $\bar{\chi} : V \rightarrow C$  such that  $\bar{\chi} \circ \phi = \chi$ . We denote by  $\text{Ex}(\chi, \mathcal{T})$  the set of all extensions of a character  $\chi$  on  $\mathcal{T}$ .

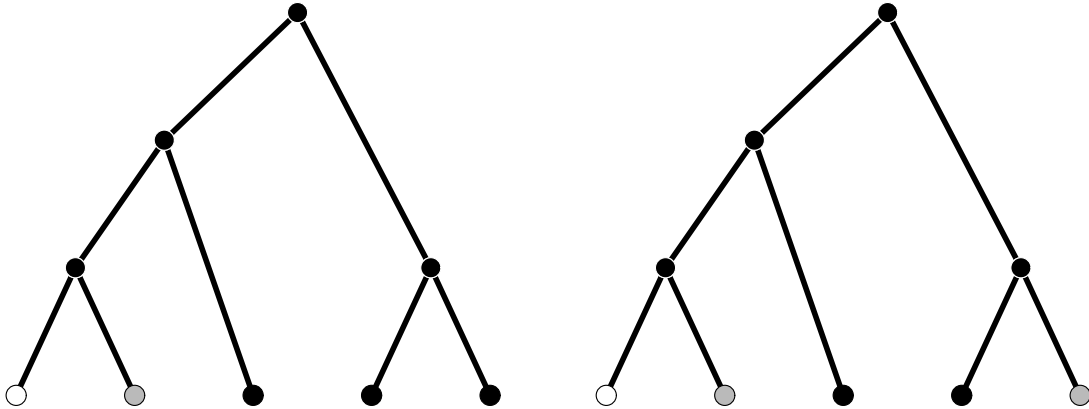
On a rooted phylogenetic tree, an extension can be seen as an *ancestral state reconstruction* of the character: it describes a possible state for un-observed characters at the ancestral nodes of the tree. See Figure 1.1.2 for an example of a convex character, along with an associated extension.

**Definition 1.1.7** (Convex Character). A character  $\chi : X \rightarrow C$  is said to be *convex* on a phylogenetic tree  $\mathcal{T} = (T, \phi)$  on  $X$  with  $T = (V, E)$  if there is an extension  $\bar{\chi} : V \rightarrow C$  of  $\chi$  such that, for any  $c \in C$ , the sub-graph of  $T$  induced by  $\{v \in V \mid \bar{\chi}(v) = c\}$  is connected.

(To be consistent with the literature, we use here the terminology defined in [Steel 1992](#). “Convex” is here to be understood as “connex”: a set of nodes is convex if any two of its elements can be linked by a path on the tree staying inside the said set.) If we consider a rooted phylogenetic tree, then we can study the evolution of a character from the root to the tips. Each node can then pass along its character value to its children, and, sometimes, the character might *shift*, going from one state to another. If those shifts can only lead to novel states, then we have a *homoplasy free* evolution process, as defined below.

**Definition 1.1.8** (Homoplasy-free evolution). Let  $B : V \rightarrow C$  be an application associating each node of a rooted phylogenetic tree  $\mathcal{T} = (T, \phi)$ , with  $T = (V, E)$ , to a state in  $C$ . We define the following properties of  $B$ :

- (i)  $B$  is said to exhibit a *reverse transition* if, on a lineage, a character goes back to a previous state, i.e. if there exists a path  $(v_1, \dots, v_k)$  in  $\mathcal{T}$  going away from the root, such that  $B(v_1) = B(v_k)$ , and that, for some  $i \in \llbracket 2, k-1 \rrbracket$ ,  $B(v_1) \neq B(v_i)$ .
- (ii)  $B$  is said to exhibit a *convergent transition* if two separate lineages converge independently to the same state, i.e. if there exists two paths  $(a, v_1, \dots, v_k)$  and  $(a, w_1, w_2, \dots, w_l)$  starting from node  $a$  and going away from the root, such that  $v_k \neq w_l$ ,  $B(v_k) = B(w_l)$ , and  $B(a) \neq B(v_k)$ .



(a) Convex character, homoplasmy free evolution. (b) Non-convex character, convergent transition.

Figure 1.1.2 – A phylogenetic tree with a convex (left) and non-convex (right) character. For the convex character, “white” and “grey” are unique innovations of the two tips on the left. For the non-convex one, the “grey” innovation appears twice: this is an example of *convergent* evolution. There are several possible extension of the character on the internal node (see Section 1.1.3), but none is homoplasmy free for the non-convex character.

- (iii) If  $B$  does not exhibit any reverse or convergent transition, then  $B$  is said to be *homoplasmy free*.

The two notions of homoplasmy-free evolution and convex characters are in fact two representations of the same concept, as we can see with the following proposition, extracted from [Semple & Steel \(2003\)](#) (Chap. 4).

**Proposition 1.1.1** (Link between convex and homoplasmy-free characters). *An homoplasmy free map on a rooted tree and a convex character on an un-rooted tree are linked by the following transformations:*

- (i) Let  $B : V \rightarrow C$  be a homoplasmy free application on a rooted phylogenetic tree  $\mathcal{T} = (T, \phi)$ , with  $T = (V, E)$ , and let  $\mathcal{T}'$  the un-rooted phylogenetic tree associated with  $\mathcal{T}$ . Then the character  $\chi = B \circ \phi : X \rightarrow C$  is convex on  $\mathcal{T}'$ , with associated extension  $\bar{\chi} = B$ .
- (ii) Reciprocally, let  $\chi : X \rightarrow C$  be a convex character on an un-rooted tree  $\mathcal{T}' = (T, \phi)$ , with  $T = (V, E)$ , and let  $\bar{\chi} : V \rightarrow C$  be its associated extension. Choose  $\rho$  and consider the tree  $\mathcal{T}$  associated to  $\mathcal{T}'$  and rooted at  $\rho$ . Then the map  $B : V \cup \{\rho\} \rightarrow C$  such that  $B(v) = \bar{\chi}(v)$  for any  $v \in V$  and  $B(\rho) = \bar{\chi}(w)$  for an arbitrary fixed  $w \in \text{child}(\rho)$ , is homoplasmy-free.

Associating a color to each state of a character, it is possible to count the number of convex colorings on a binary tree. This is given by [Steel \(1992\)](#) (Proposition 1, item 4) and proposition 4.1.4 in [Semple & Steel \(2003\)](#).

**Proposition 1.1.2** (Semple & Steel (2003), proposition 4.1.4). *Let  $\mathcal{T}$  be a binary phylogenetic tree on  $X$ , and let  $C$  be a set of  $c \geq r$  states. Then the number of full  $r$ -states characters  $\chi : X \rightarrow C$  that are convex on  $\mathcal{T}$  is*

$$\frac{c!}{(c-r)!} \binom{2n-r-1}{r-1},$$

where  $n = |X|$  and  $c = |C|$ .

**Corollary 1.1.1** (Steel (1992), proposition 1, item 4). *If the arranging order of the state does not matter, then the above formula simplifies to*

$$\binom{2n-r-1}{r-1}.$$

*Example 1.1.1.* On the tree with  $n = 5$  tips with  $r = 3$  characters presented Figure 1.1.2, there are  $\binom{10-3-1}{3-1} = 15$  coloring of the tips that induce convex characters, i.e. that can be obtained with an homoplasy free evolution.

In Section 2.3 of Chapter 2, we derive this formula using the homoplasy-free evolution formalism, and extend it to any tree (not necessarily binary).

### 1.1.3 Parsimony

Among all extensions of a character to ancestral nodes, some require less shifts between states than the others. In the following, we study the extensions that are *parsimonious*, in that they induce the less possible shifts.

**Definition 1.1.9** (Parsimony Score, Semple & Steel, 2003, Definition 5.1.1). Let  $\chi$  be a character on a phylogenetic tree  $\mathcal{T}$ , and  $\bar{\chi} \in \text{Ex}(\chi, \mathcal{T})$  an extension of  $\chi$ . The *changing number*  $\ell(\bar{\chi}, \mathcal{T})$  of  $\bar{\chi}$  is the number of characters shifts induced by the reconstruction  $\bar{\chi}$ :

$$\ell(\bar{\chi}, \mathcal{T}) = |\{(u, v) \in E \mid \bar{\chi}(u) \neq \bar{\chi}(v)\}|.$$

The *parsimony score*  $\ell(\chi, \mathcal{T})$  of  $\chi$  on  $\mathcal{T}$  is then the minimum number of shifts required to produce  $\chi$ :

$$\ell(\chi, \mathcal{T}) = \min_{\bar{\chi} \in \text{Ex}(\chi, \mathcal{T})} \ell(\bar{\chi}, \mathcal{T}).$$

An extension reaching the minimum above is a *minimum extension* of  $\chi$ .

For a character following a homoplasy free evolution, as each shift induces a new state, the number of changes needed to get  $r$  states at the tips is exactly  $r - 1$ . The link between convex characters and parsimony scores can be summarized by the following proposition:

**Proposition 1.1.3** (Semple & Steel, 2003, Proposition 5.1.3). *Let  $\chi$  be an  $r$ -state character on a phylogenetic tree  $\mathcal{T}$ . Then:*

$$\ell(\chi, \mathcal{T}) \geq r - 1,$$

*and the equality is reached if and only if  $\chi$  is convex on  $\mathcal{T}$ .*

In the following, we describe two classical dynamic programming algorithms to find a minimum extension of a given, not necessarily convex, character on a rooted phylogenetic tree. The goal of Section 2.3 in Chapter 2 is to enumerate or count the set of all minimum extensions of a given character.

**The Fitch Algorithm.** We only give a brief description of the algorithm here, but see [Felsenstein \(2004\)](#), chapter 2, or [Semple & Steel \(2003\)](#), section 5.2, for a more formal approach. For this algorithm, we assume that we have a character  $\chi$  on a rooted phylogenetic tree  $\mathcal{T}$ , and that the nodes of the tree are in a postorder (from the tips to the root). We propagate two quantities: at node numbered  $i$ ,  $\psi_i \subset C$  is a set of acceptable states, and  $\ell_i$  is the parsimony score associated with those states. See [Algorithm 1.1.1](#) for a formal description of the upward phase propagation, and [Figure 1.1.3](#) for an example. At the root  $\rho$  of the tree, we get  $\ell_\rho$  the parsimony score of the character, and  $\psi_\rho$  the set of characters that might enter in the composition of an associated minimal extension of the character.

From this set  $\psi_\rho$ , it is possible to start a downward phase, traversing the tree in a preorder, in order to define a minimal extension at all the nodes of the tree.

---

**Algorithm 1.1.1** Fitch Algorithm
 

---

```

for  $i \in \llbracket 1, |V| \rrbracket$  do
  if  $i$  is a tip then
     $\psi_i \leftarrow \{\chi \circ \phi(i)\}$ 
     $\ell_i \leftarrow 0$ 
  else  $\{i$  is an internal node $\}$ 
     $E \leftarrow \bigcap_{j \in \text{des}(i)} \psi_j$ 
    if  $E = \emptyset$  then
       $\psi_i \leftarrow \bigcup_{j \in \text{des}(i)} \psi_j$ 
       $\ell_i \leftarrow \left( \sum_{j \in \text{des}(i)} \ell_j \right) + 1$ 
    else
       $\psi_i \leftarrow E$ 
       $\ell_i \leftarrow \sum_{j \in \text{des}(i)} \ell_j$ 
    end if
  end if
end for

```

---

The upward phase of this algorithm has a complexity linear both in the number of tips and the number of characters (i.e. in  $O(|X| \times |C|)$ ). The Sankoff Algorithm described below is slightly less efficient, but can be generalized more easily.

**The Sankoff Algorithm.** This algorithm is more formally based on a *Dynamic Programming* approach. As previously, we assume that we have an  $r$ -state character  $\chi : X \rightarrow C$  on a rooted phylogenetic tree  $\mathcal{T} = (T, \phi)$ , with  $T = (V, E)$ . Here, we also assume that the cost of going from one state to the other is not always 1: for any two states  $a, b \in C$ , the cost of the transition  $a \rightarrow b$  is denoted by  $c_{ab}$ . Those costs are assumed to be known. We still assume that all the nodes  $V$  are numbered in a postorder (from the tips to the root). For convenience, we identify  $V$  with its numbering, so that, for any node  $v$  numbered  $i$ ,  $\phi(i) = \phi(v)$ . The goal of the algorithm is then to compute, for any node numbered  $i$  and any state  $a \in C$ , the minimum cost  $S_i(a)$  for node  $i$  to be in state  $a$ . This is obtained by taking the sum, over all its descending nodes  $j$ , of the minimum over all the states  $b$  of the minimum cost  $S_j(b)$  for node  $j$  to be in state  $b$ , plus the cost of going from state  $a$  to state  $b$ . See [Algorithm 1.1.2](#) for a formal description, and [Figure 1.1.4](#) for an example.

At the root node (numbered  $|V|$ ), we get the minimal cost of the character on the tree  $c(\chi, \mathcal{T}) = \min_{a \in C} S_{|V|}(a)$ . If all the transition costs are set equal to 1, then we get



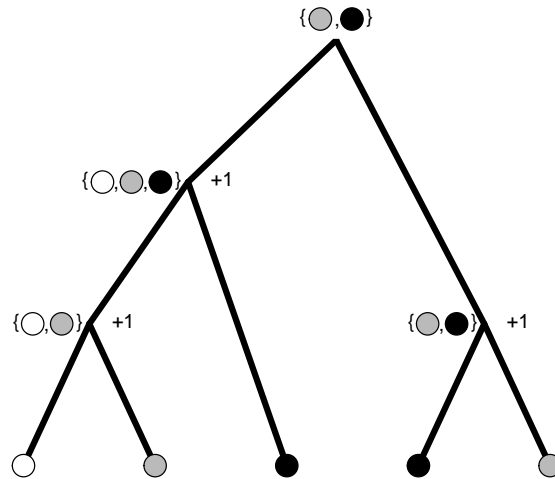


Figure 1.1.3 – Example of the Fitch algorithm on a tree with 5 tips and 3 states (showed in black, gray and white). The states of the tips are showed by a big colored circle. The set  $\psi_i$  is shown at each node  $i$ . When the union is taken, the increment of the score is shown by a +1 sign. Here, the parsimony score is equal to 3. Note that the character represented is not convex.

---

**Algorithm 1.1.2** Sankoff Algorithm
 

---

```

for  $i \in [1, |V|]$  do
  if  $i$  is a tip then
    for  $a \in C$  do
      if  $\chi \circ \phi(i) = a$  then
         $S_i(a) \leftarrow 0$ 
      else
         $S_i(a) \leftarrow 1$ 
      end if
    end for
  else  $\{i$  is an internal node $\}$ 
    for  $a \in C$  do
       $S_i(a) \leftarrow \sum_{j \in \text{des}(i)} \min_{b \in C} [c_{ab} + S_j(b)]$ 
    end for
  end if
end for

```

---

the parsimony cost:  $\ell(\chi, T) = c(\chi, T)$ .

As such, the algorithm only provides us with the parsimony cost, and not with a minimal extension of  $\chi$ . As previously, such a minimal extension can be obtained by a downward phase, selecting at each node a state that realizes the minimum cost, and choosing the state that is identical to its parent's state when possible.

The upward phase of this algorithm is in  $O(|X| \times |C|^2)$ : it is linear in the number of tips in the tree, and quadratic in the number of states.

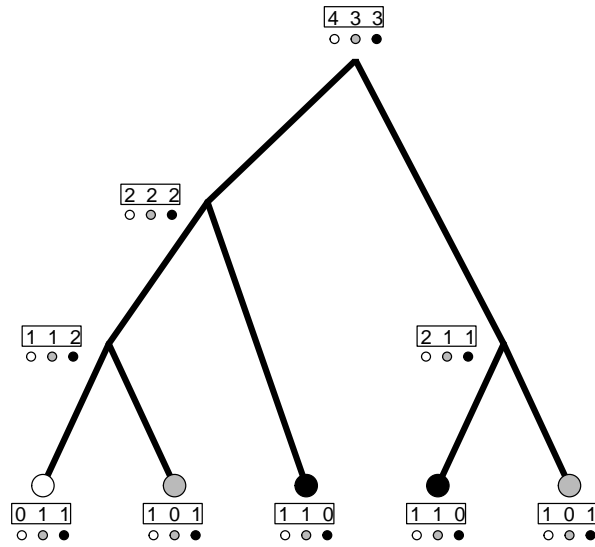


Figure 1.1.4 – Example of the Sankoff algorithm on a tree with 5 tips and 3 states (showed in black, gray and white). The states of the tips are showed by a big colored circle. The vector  $\mathbf{S}_i$  for the three states is shown at each node  $i$  as a box with associated states below. The transition costs are all set to 1. Again, the parsimony score is equal to 3 (minimum of  $\mathbf{S}_\rho$  at the root).

## 1.2 Latent Variable Models of Character Evolution

When studying a discrete trait, parsimony provides us with a first and simple criterion. It is based on the “Occam’s Razor” principle, that states that between two equivalent solutions, one should always choose the simplest one. However, evolution is a dynamic process, and it is not clear at all that it actually follows a parsimony principle. Instead of looking at the trait *a posteriori* and try to explain it using parsimony, we can try to model directly the dynamic evolution of the character through time. Defining such a model has two main advantages. First, it makes all the assumptions used in the model explicit, which makes its limitations more obvious, and can help us choose between various models, depending on the organism or biological process studied. Second, with a model, comes a natural way to evaluate it, in light of the data: likelihood. This continuous score can replace advantageously the discrete parsimony score, that has some identifiability issues, and the statistical properties of which are hard to study.

In this section, we describe a popular model for trait evolution on a tree. Under some broad assumptions, we show that this model can be casted into the framework of latent graphical models. We recall some well known properties of these models, that will be useful when studying particular instances of trait evolution models (in Sections 1.3 and 1.4).

### 1.2.1 A Generic Model for Trait Evolution on a Tree

We describe here a very generic model of trait evolution on a tree. We start by making the assumption that the tree is calibrated in time, i.e. that the branch lengths represent units of times elapsed between a node and its child. The idea is then to assume that one

or several traits, that can be discrete or continuous, evolve in time according to a given process, left unspecified here, but assumed to model the dynamics of trait evolution through time, on a given branch and for a given species. When there is a speciation event, at a node of the tree, this process splits up into two independent instances of this same process, starting at the same value. We describe in the following definition such a *branching process*, where branching times are fixed by an underlying known phylogenetic tree.

**Definition 1.2.1** (Generic Trait Evolution). Let  $T = (E, V)$  be a rooted tree, with root  $\rho$ , oriented from the root to the tips. Assume that each edge  $e \in E$  of the tree has an associated branch length  $\ell_e$ . Given a preorder numbering of the vertices, denote by  $(\mathbf{X}_i)_{1 \leq i \leq |V|}$  the sequence of random variables, taking its values in an arbitrary character space  $\mathcal{C}$  (that can be discrete or continuous, and possibly multidimensional), describing the trait of each node. The law of  $(\mathbf{X}_i)_{1 \leq i \leq |V|}$  is defined by:

- $\mathbf{X}_1 \sim \mathcal{D}(\boldsymbol{\theta}_1)$ : the root follows a given law  $\mathcal{D}$ , with parameters  $\boldsymbol{\theta}_1$ .
- Let  $e \in E$  be a branch, with child node  $i$ , and parent node  $\text{pa}(i)$ . On this branch, the trait evolve according to a stochastic process  $(\mathbf{W}_t^e, 0 \leq t \leq \ell_e)$  with law  $\mathcal{P}(\boldsymbol{\theta}_e)$ , independently from other species, conditionally on  $\mathbf{W}_0^e = \mathbf{X}_{\text{pa}(i)}$ .
- At node  $i$ , define  $\mathbf{X}_i = \mathbf{W}_{\ell_e}^e$ .
- Iterate down the tree.

One strong assumption that is made here is that, conditionally on their ancestors, species evolve *independently*. This is a key assumption, that we cannot alleviate easily. It will be essential in our approach of the problem, as shown below. It is however not very realistic, as it excludes any kind of interactions, such as competition, predation or mutualism, between species living at the same period of time. Some attempts have recently been made to introduce interactions for some classes of Gaussian processes (see e.g. Drury et al., 2016; Manceau et al., 2016; Bartoszek et al., 2016). In the rest of this thesis, we don't question this assumption anymore.

Some more assumptions need to be made on the evolution process  $\mathcal{P}$  itself. These assumptions depend on the trait studied, and on the species considered. In Sections 1.3 and 1.4, we describe in details some of them for discrete or continuous traits. One generic assumption that is almost always made however, for mathematical convenience, is that the process has the *Markov property*: if  $(\mathbf{W}_t, 0 \leq t)$  follows  $\mathcal{P}$ , then, for any  $0 \leq s < t$ , the law of  $\mathbf{W}_t$  given  $(\mathbf{W}_u, 0 \leq u \leq s)$  is the same as the law of  $\mathbf{W}_t$  given  $\mathbf{W}_s$ . In other words, the state of the traits depends on its previous states only through the last one known. This is another essential assumption, that we will not try to alleviate in the rest of this manuscript.

Under those two assumptions (conditional independence, and Markov property), we show in the next section how this generic model can be casted in a useful statistical framework.

Observe that in the definition, the process can have different parameters  $\boldsymbol{\theta}_e$  on each branch  $e$ . Going further, each branch could even have its own process  $\mathcal{P}_e$ . However, in most applications, the process, as well as the parameters, are taken constant for all the branches (i.e.  $\boldsymbol{\theta}_e = \boldsymbol{\theta}, \forall e \in E$ ). If it is the case, then from the definition above it follows that, for any node  $i$ ,  $\mathbf{X}_i = \mathbf{W}_{t_i}$ , where  $t_i$  is the time elapsed between the root and node  $i$ , and  $(\mathbf{W}_t, 0 \leq t \leq t_i)$  follows the law  $\mathcal{P}(\boldsymbol{\theta})$ , conditionally on  $\mathbf{W}_0 = \mathbf{X}_1$ .

One of the major goal of this thesis is, for some particular processes  $\mathcal{P}$ , to relax this assumptions of uniformity, and to assume that the parameters are only constant by part, *shifting* only a few times on the tree.

## 1.2.2 Directed Graphical Models and Latent Variables

We introduce here directed graphical models and latent variable models, and show how the generic model presented in the previous section can be casted in this statistical framework. We then recall some of their useful properties, as well as inference procedures, such as the Expectation Maximization algorithm, a popular algorithm for the inference of the parameters of these models through likelihood maximization.

**Definition 1.2.2** (Directed Graphical Model). A set  $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_{|V|})$  of random variables on an arbitrary state space  $\mathcal{C}$  follows a *Directed Graphical Model* if it lies at the vertices of a directed acyclic graph  $G = (V, E)$ , and is such that its joint distribution can be factorized as:

$$p_{\theta}(\mathbf{X}) = \prod_{i \in V} p_{\theta}(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)})$$

where  $\text{pa}(i)$  is the set of all direct parents of  $i$  in the graph  $G$ , and  $\theta$  is the vector of parameters of the distribution. Note that one may have  $\text{pa}(i) = \emptyset$  (e.g. at the root of a tree), in which case  $p_{\theta}(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)}) = p_{\theta}(\mathbf{X}_i)$  by convention. Equivalently,  $\mathbf{X}$  follows a Directed Graphical Model if, for any two nodes  $i$  and  $j$  such that  $j$  is not a descendant of  $i$  in the graph, then  $\mathbf{X}_i$  is independent of  $\mathbf{X}_j$  conditionally on  $\mathbf{X}_{\text{pa}(i)}$ .

When the underlying graph is a tree, each node has only one parent, and the formula above means that we can express the joint distribution as the product over all branches of the laws of each node knowing its parent. Hence, in such a model, we just need a *transmission rule* for the trait from a parent node to its children to know the entire distribution of the tree.

This formalism is quite fruitful, and can encompass the generic model of trait evolution described in the previous section:

**Proposition 1.2.1.** *Let  $(\mathbf{X}_i)_{1 \leq i \leq |V|}$  be some random variables on a rooted and directed tree  $T = (E, V)$ , generated according to a process of trait evolution described in Definition 1.2.1, with a process  $\mathcal{P}$  that has the Markovian property. Then  $(\mathbf{X}_i)_{1 \leq i \leq |V|}$  follows a graphical model on the tree  $T$ .*

*Proof.* The proof relies essentially on the two assumptions that we stressed above: conditional independence and Markov property. Using the second definition, take  $i$  and  $j$  two nodes of  $T$ , such that  $j$  is not a descendant of  $i$ , and show that:

$$p(\mathbf{X}_i, \mathbf{X}_j \mid \mathbf{X}_{\text{pa}(i)}) = p(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)})p(\mathbf{X}_j \mid \mathbf{X}_{\text{pa}(i)}).$$

Writing  $p(\mathbf{X}_i, \mathbf{X}_j \mid \mathbf{X}_{\text{pa}(i)}) = p(\mathbf{X}_i \mid \mathbf{X}_j, \mathbf{X}_{\text{pa}(i)})p(\mathbf{X}_j \mid \mathbf{X}_{\text{pa}(i)})$ , this amounts to prove that  $p(\mathbf{X}_i \mid \mathbf{X}_j, \mathbf{X}_{\text{pa}(i)}) = p(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)})$ . If  $j$  is an ancestor of  $i$ , then this is true thanks to the Markov property. Otherwise, let  $k$  be the most recent common ancestor of  $i$  and  $j$ . As  $j$  is not a descendant of  $i$ ,  $k$  is distinct from  $i$  and  $j$ . Then as  $k$  is an ancestor of  $i$ , by the Markov property, we get  $p(\mathbf{X}_i \mid \mathbf{X}_j, \mathbf{X}_{\text{pa}(i)}) = p(\mathbf{X}_i \mid \mathbf{X}_j, \mathbf{X}_{\text{pa}(i)}, \mathbf{X}_k)$ . From the conditional independence of  $\mathbf{X}_i$  and  $\mathbf{X}_j$  given  $\mathbf{X}_k$ , this is equal to  $p(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)}, \mathbf{X}_k)$ . We then conclude using the Markov property one more time.  $\square$

Once the model is defined, to do some statistical inference, we need to know which nodes are observed, and which ones are hidden. For a phylogenetic tree, representing the evolutionary history of species, we typically only have access to traits that can be observed today, at the tips of the tree. Apart from fossils, that can provide us with some insight on ancestral traits, internal nodes remain mostly unobserved.

**Definition 1.2.3** (Latent Variable Model on a Tree). A set of variables  $\mathbf{Z}$  is said to be *latent*, or *hidden* if it is un-observed, but has a direct effect on a set of observed variables  $\mathbf{Y}$ . If the set  $\mathbf{X} = (\mathbf{Y}, \mathbf{Z})$  is such that:

- $\mathbf{X}$  follows a directed graph model on a tree  $T$ ,
- $\mathbf{Y}$  are observed variables at the leaves of the tree,
- $\mathbf{Z}$  are latent variables at the internal nodes of the tree,

then  $\mathbf{Y}$  is said to follow a latent variable model on the tree  $T$ .

Intuitively, this class of models corresponds to a case where the law of a node given its parents is known, and defined by a model of trait evolution, that can be discrete or continuous, and where only extant species (at the tips of the tree) are observed. See Figure 1.2.1 for a simple example of such a setting.

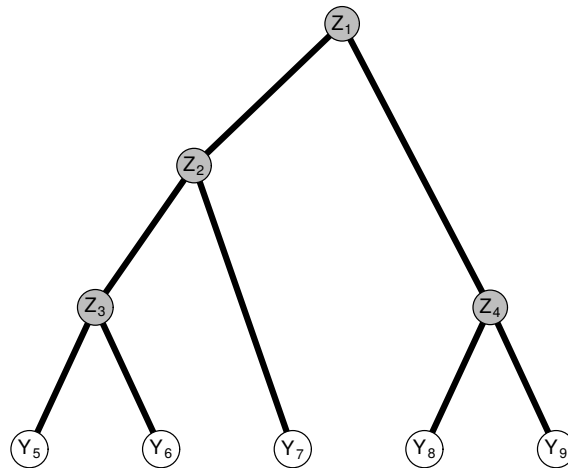


Figure 1.2.1 – Latent Variable Tree Model. The observed variables are shown in white, and the latent variables in gray. The variables are numbered in a preorder. The likelihood of the completed dataset can be factorized on the edges, with terms of the form  $p(X_i | X_{\text{pa}(i)})$ , such as e.g.  $p(Y_5 | Z_3)$ .

These models appear naturally in phylogeny, and, thanks to the underlying tree, have some nice hierarchical properties that can help us speed up many computations, including the likelihood one.

### 1.2.3 Pruning Algorithm

When we have a latent variable model on a tree, the likelihood of the data can be computed thanks to a *pruning algorithm*, that goes up the tree from the tips to the root, similarly to the Sankoff algorithm. The idea relies on the conditional independence of daughter nodes given a parent node in the tree graphical model. It has been exposed and studied in some particular examples of evolution model by Felsenstein (Felsenstein, 1973b,a, 1981, 2004).

**Proposition 1.2.2** (Pruning-like Propagation). *Denote by  $\mathbf{Y}^i$  the set of all the traits for tips that are below a given node  $i$  (i.e. all tips  $j \in \text{des}(i)$ ). Then the conditional likelihood of  $\mathbf{Y}^i$  given  $\mathbf{X}_i$  can be written as:*

$$p_{\theta}(\mathbf{Y}^i \mid \mathbf{X}_i = \mathbf{x}) = \prod_{j \in \text{child}(i)} \int_{\mathbf{u} \in \mathcal{X}} p_{\theta}(\mathbf{Y}^j \mid \mathbf{X}_j = \mathbf{u}) \cdot p_{\theta}(\mathbf{X}_j = \mathbf{u} \mid \mathbf{X}_i = \mathbf{x}) d\mathbf{u} \quad (1.1)$$

If the nodes of the tree are in a postorder, then this equation makes it possible to propagate the information from the tips of the tree to the root. When some assumptions are made on the space  $\mathcal{C}$  or on the law  $p_{\theta}$ , then some explicit formulas can be derived. For instance, if  $\mathcal{C}$  is discrete, then the integrals are just sums, and the formula can be handled more easily (see Section 1.3). Similarly, if the law studied are Gaussian, then all the integrals can be solved analytically, and we get actualization formulas similar to the ones in a Kalman filter (see Section 1.4, and Chapters 2 and 3).

### 1.2.4 Expectation Maximization

Instead of trying to integrate out the marginal distribution of the observed trait values  $\mathbf{Y}$  directly, it can be better to work with the likelihood of the completed dataset  $\mathbf{X} = (\mathbf{Z}, \mathbf{Y})$ , that is in many cases easier to compute. The well known Expectation Maximization (EM) algorithm is designed to exploit this feature. It was introduced by Dempster et al. (1977), and relies on the following decomposition of the likelihood of the observed data:

**Proposition 1.2.3** (Marginal Likelihood Decomposition).

$$\log p_{\theta}(\mathbf{Y}) = \mathbb{E}_{\theta}[\log p_{\theta}(\mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}] - \mathbb{E}_{\theta}[\log p_{\theta}(\mathbf{Z} \mid \mathbf{Y}) \mid \mathbf{Y}]$$

See e.g. Robin (2014) for a proof of this proposition and the following one. The EM algorithm is then an iterative algorithm that maximizes the marginal log-likelihood  $\log p_{\theta}(\mathbf{Y})$ , that is deemed intractable, through the conditional expectation of the completed log likelihood  $\mathbb{E}_{\theta}[\log p_{\theta}(\mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}]$ , that is supposed to be easier to handle. Informally, the algorithm can be stated as follow:

**Algorithm 1.2.1** (Expectation Maximization). *Repeat until convergence:*

**E step** *Given a current estimate  $\theta^h$  of the vector of parameters  $\theta$ , compute the moments of  $\log p_{\theta^h}(\mathbf{Z} \mid \mathbf{Y})$  needed to compute  $\mathbb{E}_{\theta^h}[\log p_{\theta}(\mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}]$  (as a function of  $\theta$ ).*

**M step** *Update the estimate of  $\theta$  as:*

$$\theta^{h+1} = \underset{\theta}{\operatorname{argmax}} \mathbb{E}_{\theta^h}[\log p_{\theta}(\mathbf{Y}, \mathbf{Z}) \mid \mathbf{Y}].$$

In general, there are no guarantees for this algorithm to converge to the global maximum of the marginal likelihood. However, we are guaranteed to converge to a local maximum, thanks to the following property, that states that the likelihood is increasing at each step of the algorithm:

**Proposition 1.2.4** (Increase of the Likelihood). *If  $\theta^h$  and  $\theta^{h+1}$  are estimates obtained by the EM algorithm described above, then:*

$$\log p_{\theta^{h+1}}(Y) \geq \log p_{\theta^h}(Y).$$

This algorithm is quite general, and can be used both for discrete and continuous models of evolution. It is at the core of our inference strategy in Chapters 2 and 3 of this thesis. For the situations we are looking at, one of the key points is to efficiently compute the E step. As our latent observations are linked by a directed tree, this can be done thanks to pruning-like algorithms, using the formula shown above (Proposition 1.2.2). These algorithms, that can also be called “forward-backward”, are designed to compute all the needed quantities in just two traversals of the tree, from the tips to the root, and back.

## 1.3 Discrete Models of Evolution and Tree Reconstruction

The first application of the above formalism are discrete models of trait evolution. As they encompass models of DNA or protein evolution, for which a great amount of data has become available over the last few decades, these models have received a lot of attention. They are often at the core of phylogenetic inference strategies to reconstruct the tree between a set of species. It is hence important to understand how they work, and on which modeling assumptions they are based. We refer the interested reader to O’Meara (2012) and Felsenstein (2004) for a more comprehensive review of these models and associated methods.

### 1.3.1 Continuous Time Markov Chains

If the studied trait is discrete, it is natural to model its evolution as a Continuous Time Markov Chain (CTMC). The Markov property of this process  $\mathcal{P}$  ensures that the resulting random variables follow a latent graphical model on the phylogenetic tree, as defined in 1.2.3 (see Proposition 1.2.1). We start by defining a general CTMC and explore a few of its properties. We then show how it can be used to model trait evolution on a tree. Finally, we give some classical models of trait evolution based on this formalism, along with their assumptions.

**A General Model.** We start by defining the general model, with its core assumptions.

**Definition 1.3.1** (Continuous Time Markov Chain (CTMC)). A *continuous time Markov chain*  $(X_t; t \geq 0)$  is a random process taking its values in a discrete finite set  $C$ , and that is such that, for any  $0 < s < t$ , the conditional law of  $X_t$  given  $(X_u; u \leq s)$  only depends on the last known value  $X_s$ , i.e. for all  $n \in \mathbb{N}$ ,  $0 \leq t_0 < t_1 < \dots < t_n < s$  and  $x_0, x_1, \dots, x_n, x, y \in C$ , we have:

$$\mathbb{P}\left[X_t = y \mid X_{t_0} = x_0, X_{t_1} = x_1, \dots, X_{t_n} = x_n, X_s = x\right] = \mathbb{P}\left[X_t = y \mid X_s = x\right].$$

The process is said to be *homogeneous* if  $\mathbb{P}[X_t = y \mid X_s = x]$  depends on  $t$  and  $s$  only through the difference  $t - s$ . In that case, for any time  $t$ , we define the *transition matrix*  $\mathbf{P}(t)$  on  $C^2$  that is such that:

$$\mathbb{P}[X_t = y \mid X_s = x] = P_{xy}(t - s).$$

This matrix expresses the probability, starting from a state  $x$  at time  $s$ , to be in state  $y$  at time  $s + (t - s)$ . For such a process, we can easily compute the probability  $\boldsymbol{\mu}_x(t)$  of the chain to be in state  $x \in C$  at time  $t$  from the initial state:

$$\boldsymbol{\mu}(t) = \boldsymbol{\mu}(0)\mathbf{P}(t),$$

where  $\boldsymbol{\mu}(t)$  is seen as a row-vector on  $C$ .

For any time  $t$ , the transition matrix  $\mathbf{P}(t)$  defines the macro state of the process. However, thanks to the Markovian nature of the process, it is possible to reconstruct the states from the infinitesimal rate of transition from one state to another on a very short time scale. That is the goal of the *instantaneous rate matrix*, as defined below.

**Proposition 1.3.1** (Instantaneous Rate Matrix). *Given a CTMC with a transition probability matrix function  $(\mathbf{P}(t); t \geq 0)$ , there is a rate  $r > 0$  and a matrix  $\mathbf{Q}$  on  $C^2$  such that:*

$$\begin{cases} Q_{xy} \geq 0 & \forall (x, y) \in C^2, x \neq y \\ Q_{xx} = - \sum_{y \in C, y \neq x} Q_{xy} \leq 0 \end{cases} \quad (1.2)$$

and, for  $h \rightarrow 0$ :

$$\begin{cases} P_{xy}(h) = hrQ_{xy} + o(h) \\ P_{xx}(h) = 1 + hrQ_{xx} + o(h). \end{cases} \quad (1.3)$$

And we have the following relationship:

$$\mathbf{P}(t) = e^{\mathbf{Q}rt}.$$

Equations (1.3) justify that we see  $\mathbf{Q}$  as an instantaneous rate matrix: on a short time scale  $h$ , the probability of going from a state  $x$  to a state  $y$  is  $h \times rQ_{xy}$ , i.e. the time elapsed times the rate of change.

Remark that the rate  $r$  that multiplies the time is not identifiable from  $\mathbf{Q}$ . It can be seen as a scaling parameter, that accelerates or decelerates time for the process evolution. In order to make it identifiable, we need some kind of normalization on  $\mathbf{Q}$ . This will be achieved by normalizing the expectation of the time needed between two events (i.e. the expected time before a transition, see Proposition 1.3.2).

**Stationarity.** In the following, we make the assumption that the process has a *stationary state*, as defined below.

**Definition 1.3.2** (Stationary State). A vector  $\boldsymbol{\pi}$  of states probability on  $C$  is said to describe a *stationary state* of a CTMC if it is not affected by the transition matrix of the process at any time  $t$ :

$$\boldsymbol{\pi} = \boldsymbol{\pi}\mathbf{P}(t).$$



Equivalently,  $\boldsymbol{\pi}$  is a stationary state iff

$$\boldsymbol{\pi}\mathbf{Q} = \mathbf{0}.$$

If it exists and is unique, the stationary state is also the asymptotic state of the chain:

$$\lim_{t \rightarrow +\infty} P_{xy}(t) = \pi_y \quad \forall (x, y) \in C.$$

In other words,  $\boldsymbol{\pi}$  represents the *equilibrium frequencies* of each state, that are not affected by the dynamical model of evolution. In the following, we only consider chains that have a stationary state, and often assume that the equilibrium has been reached, i.e. that the chain starts with initial distribution  $\boldsymbol{\pi}$ . As the equilibrium is also the asymptotic state of the chain, this amounts to assuming that, before we started looking at it, the chain has been running for “a large amount of time”, and has already reached equilibrium.

**Normalization of  $\mathbf{Q}$ .** For a stationary CTMC, we can normalize the instantaneous rate matrix  $\mathbf{Q}$ , so that the parameter  $r$  becomes identifiable. To do that, we normalize the expectation of  $T_1$  the random variable giving the time of the first event of the chain. From the standard properties of a CTMC recalled below, we know the conditional law of  $T_1$  given the initial state of the chain.

**Proposition 1.3.2** (Law of  $T_1$ ). *Conditionally to the initial state of the CTMC  $X_0 = x$ , the time of the first event  $T_1$  and the state  $Z_1 = X_{T_1}$  are two independent random variables, with  $T_1$  following an exponential law with parameter  $r q_x = -r Q_{xx}$ , and  $Z_1$  a multinomial on  $C$  with probabilities given by  $\left(\frac{Q_{xy}}{q_x}, y \neq x\right)$ .*

For a stationary CTMC, with stationary distribution  $\boldsymbol{\pi}$ , the expectation of  $T_1$  is then given by:

$$\mathbb{E}[T_1] = \sum_{x \in C} \mathbb{P}[X_0 = x] \mathbb{E}[T_1 | X_0 = x] = r \sum_{x \in C} \pi_x q_x$$

Imposing  $\mathbb{E}[T_1] = r$  imposes some constraints on  $\mathbf{Q}$ . As all subsequent times between events have the same law as  $T_1$ , this amounts to imposing a basal rate, given by  $\mathbf{Q}$ , of one event for every unit of time. The true rate is then controlled by the parameter  $r$ .

**Time Reversibility.** It is often convenient to assume that the CTMC is *time reversible*, meaning that, when played backward, the chain has the same distribution. For a stationary process, this amounts to the following definition.

**Definition 1.3.3** (Time Reversibility). A stationary CTMC is said to be *time reversible* if, for any two states  $(x, y) \in C^2$  and any time  $t$ :

$$\pi_x P_{xy}(t) = \pi_y P_{yx}(t)$$

Equivalently, the stationary CTMC is time reversible iff, for any two states  $(x, y) \in C^2$  and any time  $t$ :

$$\pi_x Q_{xy} = \pi_y Q_{yx}$$

Since it implies that  $\text{Diag}(\boldsymbol{\pi})\mathbf{Q}$  is symmetric, it makes  $\mathbf{Q}$  easier to diagonalise, which is useful when trying to compute  $\mathbf{P}(t) = e^{\mathbf{Q}t}$ . (Where  $\text{Diag}(\boldsymbol{\pi})$  is the diagonal matrix with the diagonal vector of values equal to  $\boldsymbol{\pi}$ .)

**On a Tree.** We described above the properties of one CTMC running in time. When applied to trait modeling, we need to show how this process can be applied on a tree. The idea is to use the process defined in Section 1.2.1, taking for the process  $\mathcal{P}$  a CTMC.

**Definition 1.3.4** (CTMC on a tree). Let  $T = (E, V)$  be a rooted tree, with root  $\rho$ , oriented from the root to the tips. Assume that each edge  $e \in E$  of the tree has an associated branch length  $\ell_e$ . Given a preorder numbering of the vertices, denote by  $(\mathbf{X}_i)_{1 \leq i \leq |V|}$  the sequence of random variables, taking its values in an arbitrary character space  $C$ , describing the character of each vertex. Denote by  $CTMC(\mathbf{Q}, r, \boldsymbol{\pi})$  the stationary CTMC with instantaneous rate matrix  $\mathbf{Q}$ , rate  $r$ , and stationary frequencies  $\boldsymbol{\pi}$ . Then a stationary CTMC model of evolution on the tree can be defined as:

- $X_1 \sim \boldsymbol{\pi}$ : the root is in the stationary state of the  $CTMC(\mathbf{Q}, r, \boldsymbol{\pi})$ .
- On a given branch  $e$ , the character evolves as a  $CTMC(\mathbf{Q}, r, \boldsymbol{\pi})$  for a time  $\ell_e$ .
- At a given node  $i$ , the process splits up into two independent CTMCs with the same initial state.

From this definition, each node  $X_i$  is the result of the  $CTMC(\mathbf{Q}, r, \boldsymbol{\pi})$ , running for a time  $t_i$ , the distance on the tree between node  $i$  and the root.

Thanks to Proposition 1.2.1, the model obtained follows the properties of a graphical tree model, as defined in 1.2.3. In particular, the probabilities needed for the factorization described in Definition 1.2.2 are exactly given by matrix  $\mathbf{P}(t)$ : for a node  $i$  with parent edge  $e$  and parent node  $\text{pa}(i)$ , and any two states  $(x, y) \in C$ ,

$$\mathbb{P}[X_i = y \mid X_{\text{pa}(i)} = x] = P_{xy}(\ell_e).$$

An example of such a model on a small tree is presented Figure 1.3.1.

This is the first example of a process evolving on a tree as defined in Section 1.2.1, for a discrete trait. It will be used again in the Section 1.4 for continuous traits. We stress here again that this construction makes the strong assumptions that all species evolve independently from one another, and that the process has the same parameters on all the branches, meaning that evolution is supposed to follow the same rules over the whole tree. This last assumption will be partly relaxed in Section 1.3.3.

### 1.3.2 Models of DNA Evolution

The general CTMC framework described above can be applied to any discrete trait, such as the color of a flower, or the absence or presence of a given morphological character. It is however important to check that the assumptions we made are compatible with the evolutionary mechanisms of the studied trait. One of the most fruitful example of traits these models can be applied on are DNA sequences. In the following, we study this particular example in more depth.

**Discussion of the Assumptions.** We recall here all the assumptions made in the construction above, and see how they can relate to DNA sequence evolution.

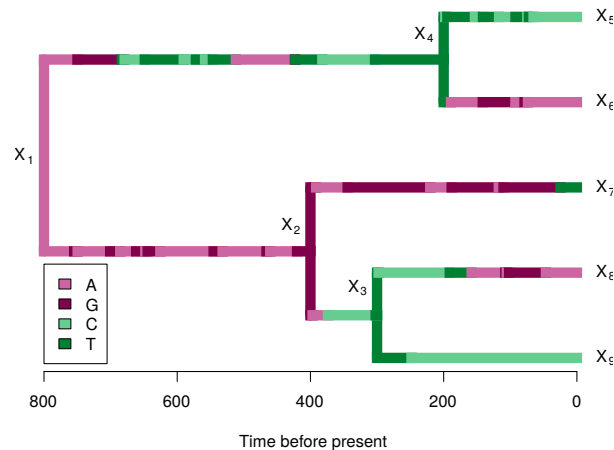


Figure 1.3.1 – Example of a CTMC process evolving on a phylogenetic tree with five tips to model DNA evolution of one site. The states are mapped on the tree using a color code for nucleotide bases: purines ( $A, G$ ) are in red tones, and pyrimidines ( $C, T$ ) are in green tones. We can see the states changing on the branches of the tree, following the branching process. The CTMC is a Kimura model (see Section 1.3.2), with rate  $r = 0.1$  and ratio of transition to transversions  $R = \frac{\alpha}{2\beta} = 10$  (transitions happen more easily than transversion). The figure was generated using function `sim.history` of R package `phytools` (Revell, 2012).

**Independence of Sites:** First, we need to define the state space  $C$ . Here, we are studying a sequence, that contains  $m$  sites, each exhibiting one of the four nucleotide base ( $A, G, C, T$ ). The state space is hence  $\{A, G, C, T\}^m$ . As the length of the sequence  $m$  can be large, one way of reducing the state space is to assume that all the sites are independent from one another. In that case, instead of studying the sequence as a whole, we can study each site independently, and we only need to deal with a state space with four elements. This assumption is very convenient, and makes the model tractable, but might not be very realistic. Indeed, at least for a coding sequence, one might think that the sites are somehow related. Following this assumption, from now on we will only consider the evolution of one site, with state space  $\{A, C, T, G\}$ .

**Independence of Species:** Similarly, at least for coding sequences, it is not clear that this assumption is justified. It is however essential in our modeling approach. As pointed out above, alleviating this assumption would require a completely new model, and is outside of the scope of this manuscript.

**Markovian Property:** As before, this assumption states that the evolutionary process has “no memory”, and is made for mathematical convenience.

**Stationarity and Time Reversibility:** As pointed out before, these are technical assumptions, that simplify the computations. They have no real biological ground, and some work has already been done to alleviate them (Galtier & Gouy, 1998; Boussau et al., 2006).

**The General Time-Reversible (GTR) Model.** Applying the assumptions above, we can derive a general formulation for a CTMC on a tree. Thanks to the independence of

sites, we can restrict to a model with 4 states, which has in general  $4^2 + 4$  parameters, for matrix  $\mathbf{Q}$  and stationary distribution  $\boldsymbol{\pi}$  (the evolution rate  $r$  will be treated separately). The following constraints apply to these parameters:

Constraint	Equation	Number
Instantaneous Rate Matrix (Def. (1.2))	$\forall x \in C, \sum_{y \in C} Q_{xy} = 0$	4
Probability Distribution	$\sum_{x \in C} \pi_x = 1$	1
Normalization (Prop. 1.3.2)	$-\sum_{x \in C} \pi_x Q_{xx} = 1$	1
Time Reversible (Def. 1.3.3)	$\text{Diag}(\boldsymbol{\pi})\mathbf{Q}$ symmetric	$4 \times 3/2 = 6$
Total		12

Hence, there are only  $20 - 12 = 8$  free parameters in the model, and the instantaneous rate matrix can be written as (Lanave et al., 1984; Felsenstein, 2004):

$$\mathbf{Q} = \begin{matrix} & \begin{matrix} A & G & C & T \end{matrix} \\ \begin{matrix} A \\ G \\ C \\ T \end{matrix} & \begin{pmatrix} - & \alpha\pi_G & \beta\pi_C & \gamma\pi_T \\ \alpha\pi_A & - & \delta\pi_C & \epsilon\pi_T \\ \beta\pi_A & \delta\pi_G & - & \eta\pi_T \\ \gamma\pi_A & \epsilon\pi_G & \eta\pi_C & - \end{pmatrix} \end{matrix}$$

where the diagonal elements are such that each row sums to zero. This model is a good compromise between complexity and tractability, and is routinely used (O’Meara, 2012). If the researcher has a better idea of the biological process, it is possible to make extra assumptions. We list here the most common ones, from the least to the most complex.

**Jukes & Cantor (1969):** This model assumes that the instantaneous rate for going from any base to any other is just  $1/3$ . It has only one parameter, the evolution rate  $r$ .

**Kimura (1980)** This models distinguishes “transitions” from “transversions”, based on the distinction between *purines* (A, G) and *pyrimidines* (C, T). Because of some structural constraints, a change within the same group, or *transition*, happens easily, with rate  $\alpha$ . A cross-group change, or *transversion*, is less frequent, and happens with a rate  $\beta$ . It has two free parameters: the rate  $r$ , and the ratio of transitions to transversions  $\alpha/2\beta$ . See Figure 1.3.1 for an example of this process.

**Tamura & Nei (1993):** This model does not assume that the equilibrium frequencies  $\boldsymbol{\pi}$  are all equal to  $1/4$ . In addition, it distinguishes transitions within purines or pyrimidines, with respective rates  $\alpha_1$  and  $\alpha_2$ , and transversions, with rate  $\beta$ . It has 6 free parameters (the rate  $r$ , three for the vector  $\boldsymbol{\pi}$ , and two for the transition/transversion rates). There are two common independent simplifications of this model. If  $\alpha_1 = \alpha_2$ , this model is also known as the *HKY* model (Hasegawa et al., 1985). If the extra constraint  $\alpha_1 - \alpha_2 = \alpha(\frac{1}{\pi_A + \pi_G} - \frac{1}{\pi_C + \pi_T})$  is imposed, this model is called the *F84* model (Kishino & Hasegawa, 1989; Felsenstein, 2004).

### 1.3.3 Molecular Clocks

It follows from the normalization defined in Proposition 1.3.2 that the time  $rt$  of the evolution process is expressed in expected number of mutation, rather than real time. However, for some downstream analysis, including the modelling of continuous characters on phylogenies, the main focus of this thesis, it is important that the tree should be

calibrated in real time  $t$ . In particular, this implies that the tree is *ultrametric*, i.e. that all the tips are synchronized in the present. To be able to do that, we need to make extra assumptions about the mutation rate  $r$ . The simplest one is to assume that it is constant across the whole tree, and homogeneous across all the sites studied. For models of DNA evolution, this is known as the *molecular clock* assumption (Zuckerlandl & Pauling, 1965), and amounts to assuming that all sites always evolve at the same pace (i.e. mutations are running like clockwork). These two assumptions (steadiness and homogeneity) are quite strong and rather un-realistic. They can be relaxed in several ways.

**Gamma Distribution of Rates.** One way to relax the homogeneity assumption is to assume that the rates are different at each site, but all drawn from a common probability distribution. This was first proposed by Yang (1993), who used a  $\Gamma(\alpha, 1/\alpha)$  distribution for the rates across the sites. This parametrization using the “shape parameter”  $\alpha$  ensures that the expectation of the distribution is 1, and the variance  $1/\alpha$ . A discretization of this distribution is needed to make the computations tractable (Yang, 1994).

**Spatial Autocorrelation.** Under the previous Gamma distribution, the sites are still supposed to be independent. To model the correlation of sites across the DNA molecule, we can use a Hidden Markov Model (HMM). This model assumes that the rate of evolution of one site only depends on its neighbors (Yang, 1995; Felsenstein & Churchill, 1996).

**“No Common Mechanism”.** One radical way to solve the problem of heterogeneity of rates is to assume that each site has its own rate on each branch of the tree. This amounts to adding a matrix  $(r_{e,i})_{e \in E, 1 \leq i \leq m}$  of  $|E| \times m$  extra parameters. This is called the “no common mechanism” model, and has been developed by Tuffley & Steel (1997). Even though this model might have too many parameters to be biologically relevant (Huelsenbeck et al., 2011), it is quite important from a theoretical point of view, as it makes the link between maximum parsimony and maximum likelihood methods of tree reconstruction (Tuffley & Steel, 1997, see next section).

**Relaxed Molecular Clocks.** The rates can also vary in time for one single site. HMMs can also be used to model a changing pace of evolution, with periods of times where the site evolves rapidly, and others where the evolution is slower. Such models are called *covarions* models (Fitch & Markowitz, 1970; Galtier, 2001), and are discrete (i.e. the rate can change between two branches, but is steady on one branch). Other models assume a Poissonian distribution of changes of rates on the tree, such as Huelsenbeck et al. (2000). It is however difficult to take both space and time autocorrelations into account in the same model.

We only browsed some of the most used models of DNA sequence evolution, but the field is still active, and many refinements can be made (O’Meara, 2012). One of the most commonly used model is the “GTR+ $\Gamma$  with clocks” model, that assumes a GTR CTMC, a (discrete)  $\Gamma$  distribution for spatial distribution of rates, and a relaxed molecular clock for the evolution of the rates in time.

### 1.3.4 Tree Reconstruction

All the models described above assumed that the tree was already known, and were simply describing how the character evolves on it. In this section, we briefly recall how these models can be used for tree reconstruction from trait data for a set of extant species that are to be related by the phylogeny. Here, the trait studied will mostly be a set of aligned DNA sequences. Getting these sequences, and preparing them to be analysed, is a difficult question in itself, that we won't cover here (see e.g. [Li & Homer, 2010](#), for a review). The basic strategy to infer a tree from a set of traits is two fold:

1. Explore the space of phylogenetic trees, to propose some candidates.
2. Rate these candidates with a common score, that can be likelihood, or parsimony.

Thanks to all the models and notions we introduced above, we have all the background needed to handle point 2. Point 1 is out of the scope of this introduction, and we will only sketch some strategies to tackle it.

**Computing a Score.** Depending on the model chosen, the score is either parsimony or likelihood. We reviewed in Section 1.1.3 two efficient algorithms to compute the parsimony score of a given tree topology. From the “no common mechanism” model mentioned in section 1.3.3, finding a minimum parsimony score can actually be re-framed into a maximum likelihood problem, although in practice this is not efficient. Depending on the model for character evolution chosen, the computation of the likelihood is more or less computationally intensive. In any case, it benefits from the tree-structured model defined in Section 1.2.3, and some pruning algorithms can be used. We refer to the papers describing the models for more information about each model inference strategy.

**Exploring the Tree Space.** The problem of efficiently searching the tree space is difficult, mostly because this space is large, growing as  $n^n$ , if  $n$  is the number of tips. Restricting ourselves to binary trees, the exact number  $T_n$  of different topologies linking a set of  $n$  species can be easily computed ([Cavalli-Sforza & Edwards, 1967](#); [Felsenstein, 2004](#)) as:

$$T_n = \prod_{i=1}^{n-2} (2i + 1).$$

Hence, an extensive search of the space is not possible, and we have to resort to heuristics to explore the space the best we can. Many strategies can be imagined, and we refer to [Felsenstein \(2004, Chap. 4\)](#) for a review of the main approaches used. Most of them rely on a “hill climbing” strategy, trying to improve the score by exploring the “neighborhood” of a candidate tree. Such strategies are only guaranteed to converge to local extrema, and choice of the starting point is often crucial. They each rely on their own metric on the tree space, that defines which trees are close from one another (see [St. John, 2016](#), for a review).

## 1.4 Continuous Models of Evolution

In the previous section, we applied the generic model of trait evolution described in Section 1.2.1 to discrete traits. In this section, we show how to model the evolution of

a set of continuous traits, using a continuous state space Markov process  $\mathcal{P}$ . We first describe the two processes we use in the rest of the manuscript, before showing some already existing methods for their inference. This will lead us to review the state of the art on the question of automatic shift detection, the main subject of this manuscript.

### 1.4.1 Models of Evolution

The two processes we are going to study in more depths are Gaussian processes, namely, the Brownian Motion and the Ornstein-Uhlenbeck. We review them both in this section, along with their assumptions. Thanks to the definition we gave of a stochastic process on a tree (Def. 1.2.1), we only need to describe the process used  $\mathcal{P}$ , and the law of the root trait.

#### 1.4.1.1 Brownian Motion

**Definition of the Process.** The Brownian Motion (BM) is the simplest Gaussian process that can be used, and it was the first one introduced to model trait evolution on a tree, with the seminal articles of Cavalli-Sforza & Edwards (1967); Felsenstein (1985). A multivariate BM  $(\mathbf{W}_t, 0 \leq t)$  of dimension  $p$ , with variance  $\mathbf{R} = \Sigma\Sigma^T$  and expectation  $\boldsymbol{\mu}$  is defined by the following Stochastic Differential Equation (SDE):

$$\begin{cases} \mathbf{W}_0 = \boldsymbol{\mu} \\ d\mathbf{W}_t = \Sigma d\mathbf{B}_t, \forall t \geq 0, \end{cases}$$

where  $\mathbf{B}_t$  is the multivariate BM with variance  $\mathbf{I}_p$ , uniquely defined as the process with independent and stationary increments, almost surely continuous, and such that  $\mathbf{B}_t \sim \mathcal{N}(\mathbf{0}, t\mathbf{I}_p)$  for any  $t \geq 0$ . We show Figure 1.4.1 one realization of this process for a simple tree.

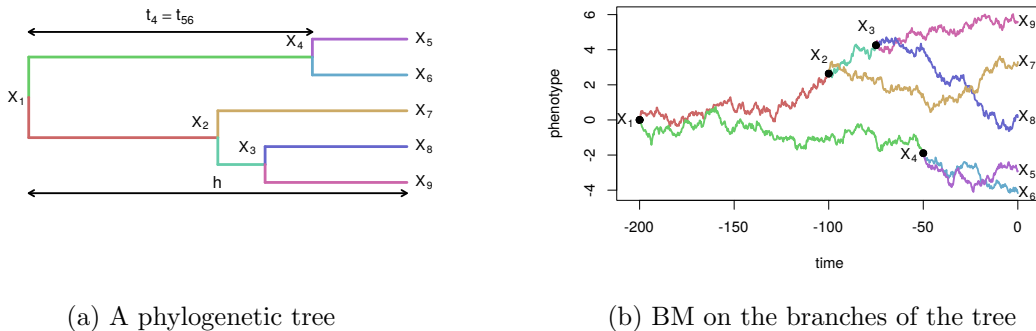


Figure 1.4.1 – Realization of a univariate BM process (with  $\boldsymbol{\mu} = \mathbf{0}$  and  $\sigma^2 = 0.04$ ) on a calibrated tree. The colors of the branches (left) match with the colors of the distinct processes (right). Only tip values are observed (at time  $t = 0$ ).

**Induced Data Structure.** Once the model is defined, we need to explore its consequences on the structure of the traits observed at the tips of the tree. Let  $\mathbf{Y}$  be the  $n \times p$  matrix of the observed  $p$  traits at the  $n$  tips of the tree. Then, if  $\mathbf{Y}$  is the result of a BM



evolution on the tree, with root node randomly distributed as a multivariate Gaussian  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ , it follows the distribution (Felsenstein, 2004; Clavel et al., 2015):

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}(\text{vec}(\mathbf{1}_n \boldsymbol{\mu}^T), \mathbf{R} \otimes \mathbf{C}_n + \boldsymbol{\Gamma} \otimes (\mathbf{1}_n \mathbf{1}_n^T)) \quad (1.4)$$

where  $\mathbf{C}_n = [t_{ij}]_{1 \leq i, j \leq n}$ , with  $t_{ij}$  the time of shared evolution of tips  $i$  and  $j$ , i.e. the time elapsed between the root and the most recent common ancestor (mrca) of  $i$  and  $j$ , and  $\mathbf{1}_n$  is the vector of ones. The operator  $\text{vec}$  is the vectorization operator, that constructs a vector by “stacking” all the columns of a matrix, and  $\otimes$  denotes the Kronecker product (see Appendix 1.B for a definition and some properties of these operators).

The formula above entirely defines the law of the matrix of observed traits at the tips, given the law of the root and the variance matrix  $\mathbf{R}$  of the BM, sometimes called the *rate matrix*. Another way to describe the covariance structure is to write the covariance  $\text{Cov}[Y_{ik}; Y_{jl}]$  between trait  $k$  at tip  $i$ , and trait  $l$  at tip  $j$  (with  $1 \leq i, j \leq n$  and  $1 \leq k, l \leq p$ ):

$$\text{Cov}[Y_{ik}; Y_{jl}] = t_{ij} R_{kl} + \Gamma_{kl}. \quad (1.5)$$

In other words, the covariance between  $Y_{ik}$  and  $Y_{jl}$  only depends on the product of the time of shared evolution  $t_{ij}$  between the tips, and the variance  $R_{kl}$  between the traits (plus the residual variance of the root traits). This factorization of the total variance in a product of the variance induced by the tree structure and the variance induced by the process is a direct consequence of the independence of increments of the BM, and will be very useful in the following. Note that often, the model is described conditionally to the root, so that the root variance  $\boldsymbol{\Gamma}$  is reduced to the null matrix.

#### 1.4.1.2 Ornstein-Uhlenbeck

**Definition of the Process.** The Ornstein-Uhlenbeck (OU) is a simple refinement over the BM. In addition to a Brownian, stochastic part, it has a deterministic call-back component, that pulls the modeled traits back to a central parameter,  $\boldsymbol{\beta}$ . Its SDE can be written as:

$$\begin{cases} \mathbf{W}_0 = \boldsymbol{\mu} \\ d\mathbf{W}_t = -\mathbf{A}(\mathbf{W}_t - \boldsymbol{\beta}) + \boldsymbol{\Sigma} d\mathbf{B}_t, \quad \forall t \geq 0, \end{cases}$$

where  $\mathbf{A}$  is the *selection strength* matrix, that describes how the modeled traits  $\mathbf{W}$  go back to their *optima*  $\boldsymbol{\beta}$ . This process was introduced in the field of phylogenetic comparative methods by Hansen & Martins (1996); Hansen (1997). See Section 1.4.1.3 (below) for a biological interpretation of this process to model stabilizing selection.

**Properties.** Contrary to the BM, and because of the call-back term of the equation, the increments of the OU are no longer independent and stationary. This makes its distribution more complex to write (see next paragraph below), and will make its inference more difficult (see Section 2.4.1 and Appendix 2.C.3).

However, contrary to the BM, the OU has a bounded variance, and admits a stationary state, provided that all the eigenvalues of the selection strength matrix  $\mathbf{A}$  are positive. When the process is univariate, the selection strength reduces to a positive scalar  $\alpha$ , and the distribution of the stationary state can be easily expressed as a Gaussian with mean  $\beta$  the optimal value, and variance  $\sigma^2/(2\alpha)$ . Often, it is easier to express the selection strength as a “phylogenetic half-life”  $t_{1/2} = \ln(2)/\alpha$  (Hansen, 1997). It is defined as the time needed by the process to cover half the distance from its current



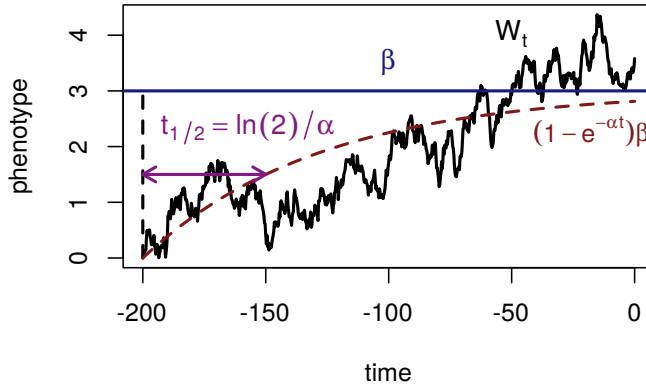


Figure 1.4.2 – One realization of a univariate OU process, with fixed root  $\mu = 0$ , variance  $\sigma^2 = 0.04$ , optimal value  $\beta = 3$ , and selection strength  $\alpha$  such that  $t_{1/2} = \ln(2)/(2\alpha) = 25\%$  of the total time allocated ( $h = 200$ ).

value to the optimum. Being homogeneous to a time, it can be handily compared with the total height  $h$  of the tree. If  $t_{1/2}$  is small compared to  $h$ , it means that the process has enough time to reach its equilibrium during its time course on the tree. On the contrary, if  $t_{1/2}$  is large compared to  $h$ , then the process will never reach its optimal state before present. See Figure 1.4.2 for a simple illustration of a univariate OU process (non branching).

**Induced Data Structure.** The distribution induced by such a multivariate OU at the tips of the tree is slightly more complex than in the Brownian case. It can be showed (Bartoszek et al., 2012; Clavel et al., 2015, and see Section 1.A) that the distribution of the matrix  $\mathbf{Y}$  of observed traits at the tips is Gaussian, and we can express its moments as follows. For  $1 \leq i, j \leq n$  two tips, let  $\mathbf{Y}^i = (Y_{i1}, \dots, Y_{ip})^T$  the (transpose) row-vector of the traits at tip  $i$ . Then:

$$\begin{aligned} \mathbb{E}[\mathbf{Y}^i] &= e^{-\mathbf{A}t_i} \boldsymbol{\mu} + (\mathbf{I} - e^{-\mathbf{A}t_i}) \boldsymbol{\beta} \\ \text{Cov}[\mathbf{Y}^i; \mathbf{Y}^j] &= \begin{cases} e^{-\mathbf{A}t_i} \boldsymbol{\Gamma} e^{-\mathbf{A}^T t_j} - e^{-\mathbf{A}t_i} \mathbf{S} e^{-\mathbf{A}^T t_j} + e^{-\mathbf{A}(t_i-t_{ij})} \mathbf{S} e^{-\mathbf{A}^T(t_j-t_{ij})} & \text{general case} \\ e^{-\mathbf{A}(t_i-t_{ij})} \boldsymbol{\Gamma} e^{-\mathbf{A}^T(t_j-t_{ij})} & \text{stationary root} \end{cases} \end{aligned} \quad (1.6)$$

where root node is randomly distributed as a multivariate Gaussian  $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma})$ , and  $\mathbf{S}$  is the stationary variance matrix of the OU (see Equations (1.30) and (1.32) in Section 1.A for a general expression of  $\mathbf{S}$ ). The second formula for the variance is obtained when the root is drawn with the stationary variance, i.e. if  $\boldsymbol{\Gamma} = \mathbf{S}$ .

Remark that this expression implies that the initial value  $\boldsymbol{\mu}$  and the optimum value  $\boldsymbol{\beta}$  are not identifiable if the tree is ultrametric, i.e. if  $t_i = h$  for any  $1 \leq i \leq n$ , where  $h$  is the height of the tree. Indeed, in this case, we only observe the combination  $\boldsymbol{\lambda} = e^{-\mathbf{A}h} \boldsymbol{\mu} + (\mathbf{I} - e^{-\mathbf{A}h}) \boldsymbol{\beta}$  at all the tips, and only this parameter is identifiable. This was pointed out in the univariate case by Ho & Ané (2014). To circumvent this problem, we often

make the assumption that the root mean value is the ancestral stationary state. In that case, we get  $\mu = \beta = \lambda$ . See Figure 1.4.3 for a simple illustration of this phenomenon.

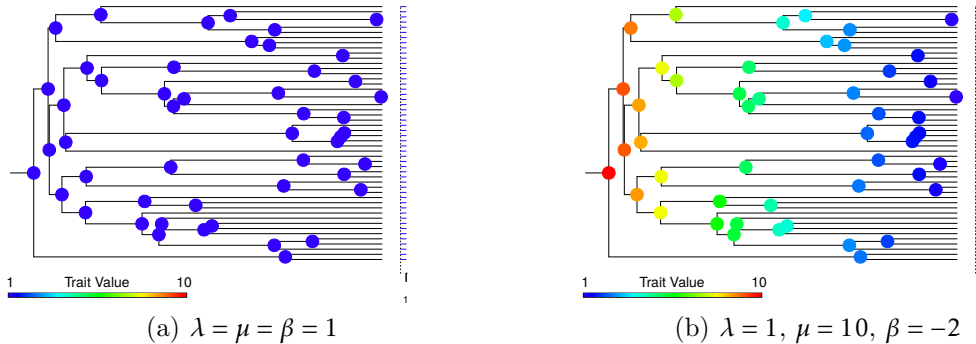


Figure 1.4.3 – Representation of the trait evolution of a univariate OU, with fixed root ( $\gamma^2 = 0$ ), and recall parameter  $\alpha$  chosen such that  $t_{1/2} = 50\%$  of the tree height. All the tips have the same expectation value, equal to  $\lambda = e^{-\alpha h}\mu + (1 - e^{-\alpha h})\beta = 1$ . The ancestral expectations at the internal nodes of the tree are indicated by a color. On the first scenario (left), the trait starts with a value  $\mu = 1$  equal to its optimum value  $\beta$ , and hence never moves away from it. In the second scenario (right), the trait starts with a value  $\mu = 10$ , far away from the optimum value  $\beta = -2$ . Hence, in the times allowed for its evolution, its expectation tries to move from 10 to  $-2$ , but when time stops, all the tips have reached the (out of equilibrium) value of  $\lambda = 1$ . The two scenarios give the exact same observed distribution at the tips, and hence are not identifiable from one another.

Note that these expressions simplify greatly when  $\mathbf{A}$  is scalar (i.e. when  $\mathbf{A} = \alpha\mathbf{I}$ ), and *a fortiori* when the trait is univariate. See Section 1.A and the Chapters 2 and 3 of this thesis for exact expressions.

### 1.4.1.3 Biological Interpretation

It is important to understand that the BM and OU used to model trait evolution at an evolutionary time-scale are distinct from models of phenotype evolution used in Quantitative Genetics, that describe evolution on a much shorter time-scale. Depending on the type of genetic variation and influence of the environment, many models of microevolution can be developed. Lande (1976) famously showed how constant additive genetic drift under a static adaptive landscape could give rise to trait evolution following either a BM (for a flat adaptive landscape) or an OU (for a landscape with a single peak). These processes are however valid only on a very short time-scale, ranging from  $10^3$  to  $10^5$  years (Hunt & Rabosky, 2014). A good example of an OU evolution from an optimum to another was observed on the particularly rich fossil record of the armor development in a lineage of stickleback fish (Hunt et al., 2008; Hunt & Rabosky, 2014). Even if the selection strength for this group is quite weak, only a few thousands years were needed for the adaptation to happen. If the selection is high, this phenomenon can even be observed empirically, in real-times studies. Evidences of adaptation could hence be found for an arboreal niche in native *Anolis* species *carolinensis* after the introduction of an invasive species *sagrei* from Cuba to small islands off Florida in only 15 years (1995–2010 Stuart et al., 2014).

Those time scales are to be compared with the height of a typical phylogenetic tree, ranging between  $10^7$  to  $10^8$  years (e.g. around  $2.5 \cdot 10^7$  years for New World Monkeys [Aristide et al., 2016](#),  $1.1 \cdot 10^8$  years for birds [Jetz et al., 2012](#), and  $2.1 \cdot 10^8$  years for Chelonians, [Jaffe et al., 2011](#)). On such long stints, the assumptions used by [Lande \(1976\)](#) do not hold anymore. It indeed has been observed that the rate of phenotypic change inferred from fossil records was much lower than expected from these models, which has been referred to as “the paradox of stasis” ([Hansen & Houle, 2004](#)). The interpretation of the use of BM and OU processes in this context hence need to be adapted.

The BM used does not represent the plain consequences of a genetic drift, but rather the random evolution of adaptive niches. The assumption is that the traits are going to their optimal values very quickly (compared to the total height of the tree), and hence that what we are tracking down are the successive optima, rather than the bare trait. The BM would hence represent the stochastic evolution of the ecological niche in time, or *secondary optimum*, to follow the nomenclature of [Hansen \(1997\)](#). That would explain why the rate of evolution measured on macro-evolutionary time scales is not the same as the rate measured for microevolution. It is also worth noting that, using this interpretation, the BM can be used to model adaptive traits on phylogenies, provided they adapt to a random environment ([Felsenstein, 2004](#), Chap. 24).

The interpretation of the OU in this context is then similar, except that the secondary optimum itself is converging toward another *primary optimum* ([Hansen, 1997](#)). It expresses the idea that the adaptive landscape usually goes through many local, stochastic changes, but sometimes shifts drastically, due to a major environmental event, such as migration, or climate change (see e.g. [Jaffe et al., 2011](#)). The OU has in addition the advantage over the BM that its variance is bounded, and that it has a stationary state, which can make it more suitable to study stabilizing selection ([Hansen & Orzack, 2005](#)).

Following the interpretation given above, we will always assume in the following that we are studying the secondary optimum, rather than the traits, of the species at hand. This secondary optimum is approached by the empirical mean of the traits within all the individuals measured for a given species. Doing that, we completely ignore the intraspecific variations, as well as measurement errors, that are often not very well known. There are however several methods to take them into account, and ignoring them can lead to severe biases in the analysis. See Section 1.4.5.1 of this introduction for a brief review of these methods, and Perspective Section 5.1 for a study of the impact of these errors on shift detection, and ways to explicitly model them in our framework.

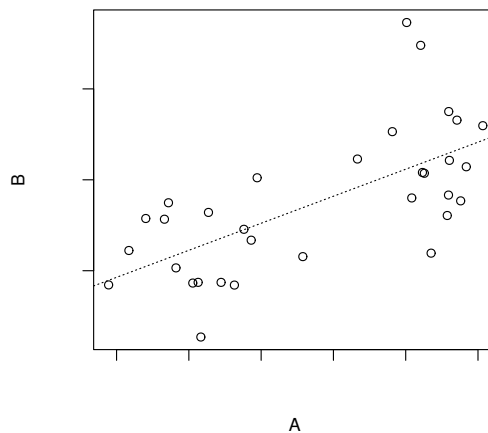
## 1.4.2 Phylogenetic Comparative Methods

Phylogenetic Comparative Methods are the tools used to study continuous traits of related species. It relies on models of evolution such as the ones presented above, and aims at finding patterns in the traits studied, while taking into account the relatedness of the species at hand. We will see how this problem can be recasted as a general linear model, i.e. a linear model where the residual errors are not independent. This formulation leads us to see the tree as a “nuisance” structure parameter, that can be taken into account in several ways. Using the regression term of the model, we also see how some fixed shifts can be introduced in the processes described above.

### 1.4.2.1 Felsenstein’s Phylogenetic Contrasts

Until now, we have been looking at the problem in a dynamic fashion, trying to model the evolution of one or several traits in time, on a phylogenetic tree. However, the data accessible to researchers are only traits measured on extant species, in the present. Hence, apart from some rare cases where good fossil records are available, inference on the dynamic component of the problem is bound to be quite limited, and interest in this topic quite recent. Historically, the questions yielded by this kind of data were more ecological, and the focus was on trying to infer some correlations between several observed traits.

Let’s follow [Felsenstein \(1985\)](#) original example, and assume that we measured two traits (say, traits A and B) for 32 species of a same phylogenetic group (see also [Felsenstein, 2004](#), Chap. 25). This kind of data is abundant in the literature (see e.g. [Pennell & Harmon, 2013](#)). Suppose that we make a scatter plot of the two traits, for all species, as shown [Figure 1.4.4](#). On this figure, there seems to be a clear trend, and it is tempting to conclude that the two traits A and B are indeed biologically correlated.

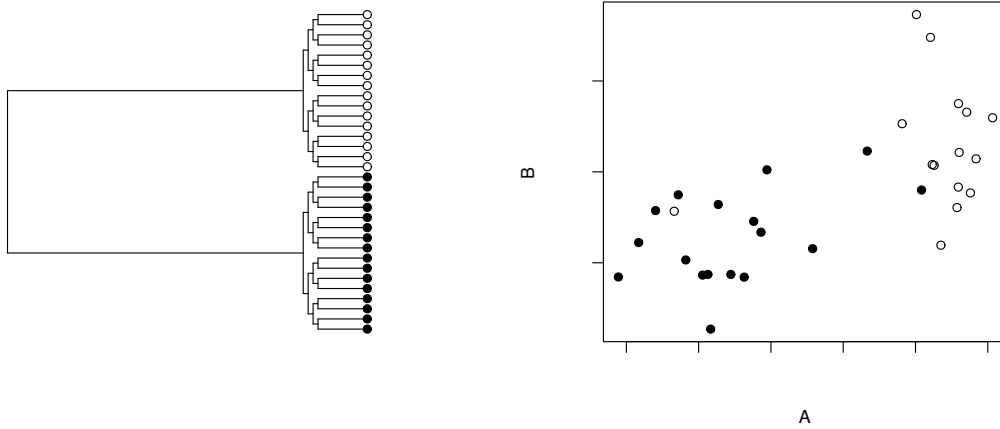


[Figure 1.4.4](#) – Scatter plot of traits A and B. The dotted line is the simple regression line (R function `lm`, [R Core Team, 2017](#)). A Pearson correlation test between the two traits (R function `cor.test`) gives a p-value of  $3.9e-05$ . This would lead us to conclude that the two traits are biologically correlated. (The arbitrary scales on the axes are omitted).

However, this analysis does not take the phylogenetic tree into account. In other words, it assumes that all the species are *independent*, where in fact they are all related. Indeed, two species that are close parents in the tree are bound to have a more similar trait than two distantly related ones, since they had less time to diverge away. We can see the influence of the tree by plotting the tree along with the scatter plot, as shown [Figure 1.4.5](#).

[Figure 1.4.5](#) shows us that ignoring the tree can be misleading on the actual biological process going on. Indeed, the traits plotted were actually simulated according to two *independent* BM on the tree, so, in contradiction with the conclusion of our first naive regression, there are no biological correlation between traits A and B.

One first way to deal with this structure induced by the phylogenetic tree is to use the so-called *phylogenetic contrasts* ([Felsenstein, 1985](#)). In this model, we assume that



(a) The phylogenetic tree linking the 32 species. There are two well formed clades, showed in white and black. (b) The same scatter plot as before, but with species colored according to their clade. (Arbitrary scales omitted.)

Figure 1.4.5 – When the species are colored according to their clades, we can see that they are two clearly formed groups. The correlation seems to be entirely driven by this distinction. Within each group, the correlation is not obvious.

the traits observed at the tips were generated according to a multivariate BM on the (known) phylogenetic tree, and we take advantage of the independent increments of the BM to construct new variables, or *contrasts* that are *independent* from each other. We describe the procedure in the pseudo-algorithm below for a univariate trait, the extension to multivariate being straightforward.

**Algorithm 1.4.1** (Phylogenetic Contrasts (Felsenstein, 1985)). Assume that we have a latent tree model as defined in Definition 1.2.3 on a rooted phylogenetic tree  $T = (E, V)$ , with the characters  $(X_1, \dots, X_{|V|})$  following a BM model of evolution on the tree, with variance  $\sigma^2$ . We assume that the tree is bifurcating, and that each branch  $e$  has length  $\ell_e$ . Iterate the following steps:

1. Find two adjacent tips on the tree, numbered  $i$  and  $j$ , with common ancestor node  $k$ .
2. Compute the contrast  $C_a = X_i - X_j$ , and update the ancestral trait value of node  $k$  to:

$$X'_k = \frac{\ell_j X_i + \ell_i X_j}{\ell_i + \ell_j}.$$

The two new quantities have the following properties:

$$\begin{aligned} \mathbb{E}[C_a] &= 0 & \text{Var}[C_a] &= \sigma^2(\ell_i + \ell_j) \\ \text{Cov}[C_a; X'_k] &= 0 & \text{Cov}[X'_k; X_l] &= \text{Cov}[X_i; X_l] = \text{Cov}[X_j; X_l], \forall l \notin \{i, j, k\} \\ \text{Var}[X'_k] &= \sigma^2 \frac{\ell_i \ell_j}{\ell_i + \ell_j} + \text{Var}[X_k] \end{aligned}$$

3. Drop the two tips  $i$  and  $j$  from the tree, and replace their ancestor  $X_k$  by  $X'_k$ . Add an extra length  $\delta_k = \frac{\ell_i \ell_j}{\ell_i + \ell_j}$  to the branch going to node  $k$  (so that  $\ell'_k = \ell_k + \delta_k$ ).

4. Go back to step 1 if the output tree has more than 3 nodes.

After this iteration, starting with  $n$  measures at the tips, we end up with  $n - 1$  contrasts (the tree is bifurcating), each independent, and with a known variance.

If we apply this method to our toy-dataset, we get the scatter plot presented Figure 1.4.6. The spurious correlation effect does not appear anymore, and the traits, that were indeed simulated to be independent, are not seen as correlated anymore.

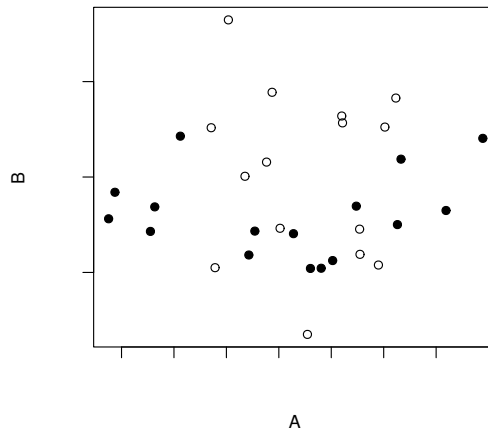


Figure 1.4.6 – Scatter plot of contrasts on traits A and B (computed with `ape` (Paradis et al., 2004) function `pic`). A Pearson correlation test between the two traits gives a p-value of 0.88: there is no significant correlation between the two traits. (Arbitrary scales omitted.)

Algorithm 1.4.1 is quite efficient, as it only needs one tree traversal. However, it is not very flexible, and models other than the BM cannot be used. The general framework of the phylogenetic regression will allow us to extend this approach to a more general class of models.

#### 1.4.2.2 The Phylogenetic Regression Framework

The idea of Phylogenetic regression is to recast the problem in the framework of a generalized linear model. If  $\mathbf{Y}$  is an  $n \times p$  matrix of  $p$  traits measured at the  $n$  tips of a phylogenetic tree, we write:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E} \quad (1.7)$$

where  $\mathbf{X}$  is an  $n \times q$  matrix of regressors,  $\boldsymbol{\theta}$  a  $q \times p$  matrix of coefficients, and  $\mathbf{E}$  an  $n \times p$  error vector, that is such that

$$\text{vec}(\mathbf{E}) \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\mathcal{S}})$$

with  $\boldsymbol{\mathcal{S}}$  an  $np \times np$  variance matrix given by the model of trait evolution used. Note that this can also be seen as a *mixed* model, as the errors are not independent identically distributed (i.i.d.). This formalism was introduced by Grafen (1989, 1992) for the Brownian Motion, and has been extended to fit a myriad of other phylogenetic models of trait evolution. The main idea is that the model of evolution will induce different kinds

of correlations between the species, and hence induce a different form for the variance matrix  $\mathbf{S}$ . We give here a couple of examples for  $\mathbf{S}$ .

**Brownian Motion.** Taking the notations of Section 1.4.1.1, if the root is assumed to be fixed (or if inference is done conditionally to the root's value), then a set of traits resulting from a BM can be written as:

$$\mathbf{Y} = \mathbf{1}_n \boldsymbol{\mu}^T + \mathbf{E} \quad \text{with} \quad \text{vec}(\mathbf{E}) \sim \mathcal{N}(\mathbf{0}, \mathbf{R} \otimes \mathbf{C}_n) \quad (1.8)$$

and this fits model (1.7), with  $\mathbf{X} = \mathbf{1}_n$ ,  $\boldsymbol{\theta} = \boldsymbol{\mu}^T$ , and  $\mathbf{S} = \mathbf{R} \otimes \mathbf{C}_n$ . In that case, this model is the same as the one used for the phylogenetic contrasts method, that can be seen as an efficient way deal with this regression problem. The strength of this new formulation is to recast the problem in a well known statistical framework, and hence to benefit from all the classical inference and analysis tools available in the literature (see, e.g. Mardia et al., 1979, for multivariate regression models). Note that in the Brownian case, some explicit maximum likelihood estimators can be exhibited, although their naive computation require the inversion of matrix  $\mathbf{C}_n$ , which can be computationally intensive for large trees. We refer to Section 1.4.4 for some computational tools to make these computations more light-weight.

This model can be easily enriched by adding some covariates in the regression matrix  $\mathbf{X}$ . This can for instance allow us to explore the direct links between a trait and an environmental predictor (Grafen, 1989).

**Ornstein-Uhlenbeck.** Similarly, using the developments of Section 1.4.1.2, we can cast the multivariate OU on an ultrametric tree in this framework. The expressions of matrices  $\mathbf{X}$  and  $\mathbf{S}$  directly follows from Equation (1.6). Note that the expression of  $\mathbf{S}$  is a bit tedious in the general case, and cannot be nicely factorized as in the BM case.

The univariate OU, with selection strength  $\alpha$ , variance  $\sigma^2$ , and initial variance at the root  $\gamma^2$  is easier to write:

$$\mathbf{Y} = \mathbf{1} \lambda + \mathbf{E} \quad \text{with} \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{S}(\alpha))$$

with  $\lambda = e^{-\alpha h} \boldsymbol{\mu} + (1 - e^{-\alpha h}) \boldsymbol{\beta}$ , and for  $1 \leq i, j \leq n$  two tips,

$$S(\alpha)_{ij} = e^{-2\alpha h} \gamma^2 + e^{-2\alpha h} (e^{2\alpha t_{ij}} - 1) \frac{\sigma^2}{2\alpha} \quad (1.9)$$

The two cases where the root variance is either null or equal to the stationary variance are often considered, in which case the expression simplifies to:

$$S(\alpha)_{ij} = \begin{cases} e^{-2\alpha h} (e^{2\alpha t_{ij}} - 1) \frac{\sigma^2}{2\alpha} & \text{fixed root} \\ e^{2\alpha d_{ij}} \frac{\sigma^2}{2\alpha} & \text{stationary root} \end{cases}$$

where  $d_{ij} = 2(h - t_{ij})$  is the *phylogenetic distance* between  $i$  and  $j$ . Note that, when  $\alpha$  goes to zero, we recover the variance structure of a simple BM, as expected (e.g. for a fixed root):

$$\mathbf{S}(\alpha) \xrightarrow{\alpha \rightarrow 0} \sigma^2 \mathbf{C}_n$$



### 1.4.2.3 Some Tree Transformations

Once recasted in this linear regression framework, we can see that the model of trait evolution on the tree only impacts the problem through the variance matrix  $\mathbf{S}$ . Forgetting for a moment the mechanistic model, the biological interpretation of which might be dubious, as seen in Section 1.4.1.3, it can be tempting to see the tree only as a measure of correlation between traits, whose strength needs to be adjusted. Restricting ourselves to the univariate case for the sake of clarity, recall that the Brownian model can be written as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\theta} + \mathbf{E} \quad \text{with} \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{C}_n) \quad (1.10)$$

and the tree structure of the data is entirely represented by matrix  $\mathbf{C}_n = [t_{ij}]_{1 \leq i, j \leq n}$ . The idea originally presented in Pagel (1999) is to take this model as a base scenario, and to alter it in several ways in order to weaken the Brownian-induced structure, that might be ill suited to some traits. It is a very pragmatic view of the problem, as it introduces an *ad hoc* parameter that is chosen in order to fit the data at best. We review here some of these classical tree transformations.

**Pagel's  $\lambda$  (Pagel, 1999).** This first and most popular transformation introduces a new parameter  $\lambda$ , that alters the variance matrix in the following way:

$$\begin{cases} C(\lambda)_{ij} = \lambda C_{ij} & \forall i \neq j \\ C(\lambda)_{ii} = C_{ii} & \forall 1 \leq i \leq n \end{cases} \quad (1.11)$$

The parameter  $\lambda$  only affects the covariances between tips, and leave their variances unchanged. This modified variance matrix can be seen as resulting from a BM evolving on a tree with modified branch lengths, in the following way:

$$\ell_i(\lambda) = \begin{cases} \lambda \ell_i & \text{if } i \text{ is an internal node} \\ \ell_i + (1 - \lambda)t_{\text{pa}(i)} = \lambda \ell_i + (1 - \lambda)t_i & \text{if } i \text{ is an external node (tip)} \end{cases} \quad (1.12)$$

This amounts to multiplying all the internal branch lengths by  $\lambda$ , and then lengthening the external branch lengths so that the resulting tree keeps its original height. When  $\lambda = 1$ , the model reduces to a BM on the original tree. When  $\lambda = 0$ , all the tips are independent, and the model is equivalent to a BM on a *star tree* (see Fig. 1.4.7). This parameter  $\lambda$  is hence sometimes seen as a measure of the *phylogenetic signal* exhibited by the data

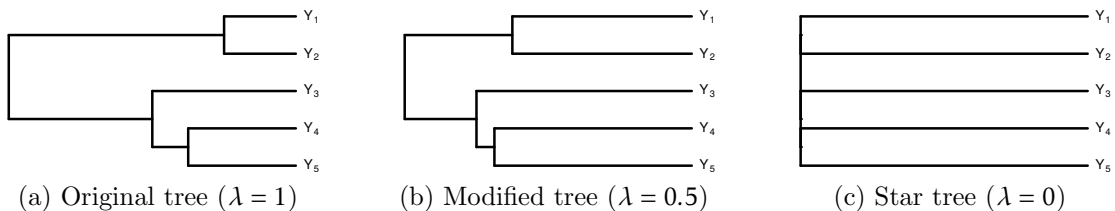


Figure 1.4.7 – Transformations of the tree induced by Pagel's  $\lambda$  parameter.

Note that this extra parameter  $\lambda$  needs to be fitted along with  $\boldsymbol{\theta}$  and  $\sigma^2$ . There are no explicit estimator for this parameter, and it needs to be optimized numerically. It is then common to design a likelihood ratio test to test  $\lambda = 0$  against  $\lambda > 0$ , in order to test



the *heritability* of the trait at hand, i.e. whether the trait is impacted by the phylogeny or not. (Note that as the null hypothesis lies at the boundary of the domain, one must be cautious in defining the asymptotic distribution of the statistic, see e.g. [Self & Liang 1987](#)).

**Pagel's  $\kappa$  ([Pagel, 1999](#)).** In this model, all the branch lengths are set up to the power  $\kappa$ :

$$\ell_i(\kappa) = (\ell_i)^\kappa \quad \forall 1 \leq i \leq n$$

When  $\kappa > 1.0$ , more change is expected on long branches, while short branches are even shortened. On the contrary, if  $\kappa < 1.0$ , then short branches are made longer, while long branch are shortened. Note that the resulting tree is not ultrametric anymore (see [Fig. 1.4.8](#)). The variance matrix  $\mathbf{C}$  is obtained by running a BM on the re-scaled tree.

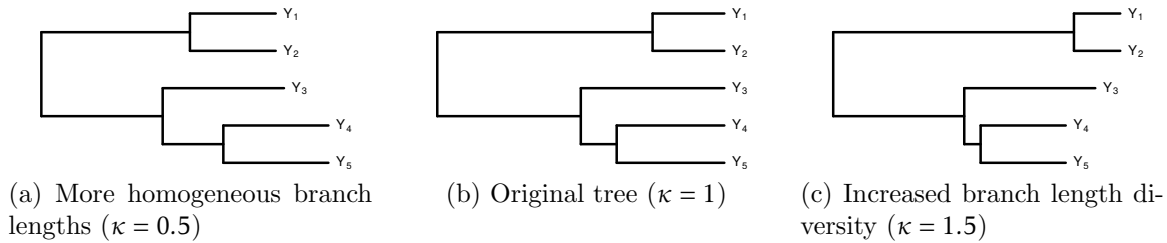


Figure 1.4.8 – Transformations of the tree induced by Pagel's  $\kappa$  parameter.

**Pagel's  $\delta$  ([Pagel, 1999](#)).** In this model, the node depths are all set up to the power  $\delta$ :

$$t_i(\delta) = (t_i)^\delta \cdot h^{1-\delta} \quad \forall 1 \leq i \leq n$$

where the factor  $h^{1-\delta}$  ensures that the resulting tree keeps the same total height  $h$ . When  $\delta > 1.0$ , more change is expected toward the end of the tree, i.e. the evolution of the character happened late in time. On the contrary, if  $\delta < 1.0$ , then most of the trait evolution is expected to happen near the root, and the trait stays stable later in time (see [Fig. 1.4.9](#)).

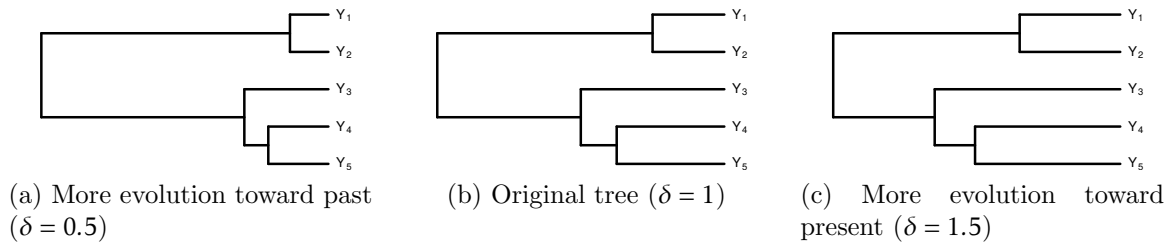


Figure 1.4.9 – Transformations of the tree induced by Pagel's  $\delta$  parameter.

#### 1.4.2.4 The OU as a Tree Transformation

In the previous section, we have given a mechanistic interpretation of the OU to model trait evolution. However, looking at Equation (1.9), we can see that the variance structure induced by the OU can be obtained by running a BM on a tree with branch lengths

modified as follow:

$$\ell_i(\alpha) = \frac{1}{2\alpha} e^{-2\alpha h} (e^{-2\alpha t_i} - e^{-2\alpha t_j}) \quad (1.13)$$

To yield exactly the same distribution, the BM used must be taken with a root variance of  $e^{-2\alpha h} \gamma^2$ , where  $\gamma^2$  is the root variance of the original OU.

Note that this only holds for ultrametric trees, and that this branch lengths transformation is similar but distinct from the classical time transformation used to get a Brownian solution to the OU SDE (see Section 1.A, Lemma 1.A.1 for a recall of this unused transformation). This branch lengths transformation has been described and used several times in the literature, under different forms (Blomberg et al., 2003; Ho & Ané, 2013a; Pennell et al., 2015). It is at the core of our inference strategy in Chapter 3.

The induced tree transformation is presented Figure 1.4.10. The effects on the tree are similar to Pagel's  $\delta$  transformation when  $\delta > 1.0$ : the higher  $\alpha$  is, the more evolution happens toward the present. In the limit  $\alpha \rightarrow +\infty$ , the tree tends to a star tree.

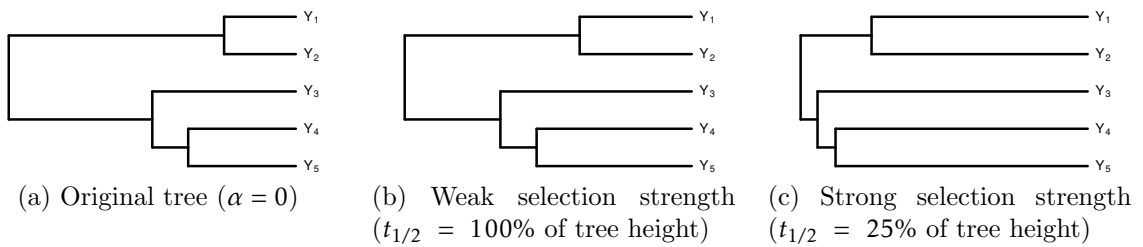


Figure 1.4.10 – Transformations of the tree induced the OU process.

### 1.4.2.5 Other Models of Evolution

We have been focusing until now on only two models of evolution, namely, the BM and the OU, but the flexibility of the phylogenetic regression framework allows for many more models. When crafting a model, the only limits are tractability and identifiability. Indeed, the model needs to be summarized in the variance matrix  $\mathcal{S}$ , so that, first, one needs to be able to carry the computations out to get the form of the matrix, and then ensure that no other common process could result in the same variance structure. To give the reader an idea of this mechanism, we give some details of two other popular models.

**ACDC.** The *Accelerating / Decelerating* (ACDC) model, sometimes also called the *Early Burst* (EB) model, was first introduced in Blomberg et al. (2003), and further developed in Harmon et al. (2010). Using the same notations as before, the SDE describing the trait evolution in time can be written as:

$$\begin{cases} W_0 = \mu \\ dW_t = \sigma^2(t) dB_t, \forall t \geq 0, \end{cases} \quad \text{with} \quad \sigma^2(t) = \sigma_0^2 e^{rt}$$

The variance matrix of the traits at the tips of the tree can then easily be derived as (Blomberg et al., 2003; Uyeda et al., 2015):

$$S(r)_{ij} = \sigma_0^2 \frac{e^{rt_{ij}} - 1}{r}$$

The trait is hence assumed to follow a BM on the tree, but with a variance that is changing in time, either increasing ( $r > 0$ , AC) or decreasing ( $r < 0$ , DC). The effect is hence similar to Pagel's  $\delta$  transform described above. It indeed can also be seen as a tree transformation, just like the OU. In fact, it is easy to see from Equation (1.9) showing the variance matrix of the OU that the AC model ( $r > 0$ ) is equivalent to the OU in some cases, as expressed by the following proposition, first proved in the Appendix of Uyeda et al. (2015):

**Proposition 1.4.1** (Equivalence of OU and ACDC (Uyeda et al., 2015)). *The OU with a fixed root ( $\gamma^2 = 0$ ), such that the initial value is equal to the optimal value ( $\mu = \beta$ ), with selection strength  $\alpha$  and variance  $\sigma^2$ , on an ultrametric tree, yields the same covariance matrix than an AC with parameters:*

$$\begin{cases} \sigma_0^2 = \sigma^2 e^{-2\alpha h} \\ r = 2\alpha \end{cases}$$

**OUBM and OUOU.** To explore situations where the optimum value  $\beta$  of an OU varies in time, Hansen et al. (2008); Bartoszek et al. (2012) introduced the ‘‘OUBM’’ and ‘‘OUOU’’ models of phenotypic evolution. In these models, the primary optimum  $\beta(x)$  is explicitly modeled as a BM or an OU process, while the secondary optimum  $y$  is selectively called back to it following an OU. In its simplest form, the OUBM model can be expressed by the set of equations (Hansen et al., 2008):

$$\begin{cases} dy = -\alpha(y - \beta(x))dt + \sigma_y dB_y & \text{with } \beta(x) = b_0 + b_1 x \\ dx = \sigma_x dB_x \end{cases}$$

where  $B_x$  and  $B_y$  are independent BMs with variances  $\sigma_x^2$  and  $\sigma_y^2$ ,  $\alpha$  is the selection strength,  $y$  is the response variable following an OU, and  $x$  a predictor variable determining the evolution of the primary optimum  $\beta(x)$ . It is possible to derive the variance matrix at the tips induced by such a model (Hansen et al., 2008):

$$S_{ij} = \frac{b_1^2 \sigma_x^2 + \sigma_y^2}{2\alpha} (1 - e^{-2\alpha t_{ij}}) e^{-\alpha d_{ij}} + b_1^2 \sigma_x^2 t_{ij} \left( \left( \frac{1 - e^{-\alpha h}}{\alpha h} \right)^2 - 2 \frac{1 - e^{-\alpha h}}{\alpha h} \frac{1 - e^{-\alpha t_{ij}}}{\alpha t_{ij}} e^{-\alpha t_{ij}/2} \right).$$

The OUOU model is similar, except that the second equation is replaced by an OU (see Bartoszek et al., 2012 for a study of this model in a multivariate setting). Similar expressions for the distribution of the traits at the tips of the tree can be obtained.

### 1.4.3 Detecting Shifts

In all the models described above, we assumed that there were only one set of parameters controlling the process on the entire tree. When the species linked by the tree are heterogeneous (for instance, if the tree is big and spans over a long period of time), then this assumption is likely to be false, and one might want to define regions in the tree, each with its own ‘‘regime’’, i.e. allowing some parameters to differ from one region to another. Looking at the process dynamically, this amounts in considering parameters that are piecewise constants, and experience some *shifts* in their history. When having an effect on the expectation, we will show how these shifts can easily be included in the linear regression framework, provided their position is known *a priori*. The main focus

of this thesis is to automatically detect the position of those shifts. We first review some methods in the literature trying to tackle this issue, before presenting models putting shifts in other parameters of the process (such as variance or selection strength).

### 1.4.3.1 Including Shifts in the Regressors

The problem of adding shifts in the optimal value of an OU was first tackled by the seminal article of [Butler & King \(2004\)](#) for the OU. We do not expose the specifics of their method here, but instead show how this problem can be recast in the linear model framework. We only sketch here the main results, but this problem is carefully exposed in Section 2.2.3 for the univariate process, and Section 3.2 for the multivariate one. Although quite simple, this new parametrization of the problem is very powerful, as it allows us to benefit from the proficient literature on linear models.

In this section, we assume that we have an ultrametric tree  $T = (E, V)$ , with all nodes numbered from 1 to  $|V| = m + n$ , with  $n$  tips and  $m$  internal nodes. We give to each branch the number of its daughter node (including a fictive branch associated with the root). We study a multivariate BM or OU on the tree, with dimension  $p$ . We assume that, for the BM, the mean parameter  $\mu$ , and, for the OU, the primary optimum  $\beta$  can shift on the phylogeny. Denote by  $\Delta$  the  $(m + n) \times p$  matrix of those shifts: line  $j$  takes the  $p$  values of the shifts on each traits if there is indeed a shift on the branch leading to node  $j$ , and is null otherwise. We further define the incidence matrix as follow.

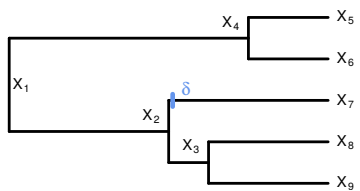
**Definition 1.4.1** (Incidence Matrix). The *incidence* matrix  $\mathbf{U}$  associated with the tree is the matrix of size  $(m + n) \times (m + n)$  defined by the formula:

$$U_{ij} = \begin{cases} 1 & \text{if } j \text{ is an ancestor of } i \\ 0 & \text{else.} \end{cases}$$

Denote further by  $\mathbf{T}$  the  $n \times (m + n)$  sub-matrix of  $\mathbf{U}$  with only lines corresponding to tips.

We refer to Example 1.4.1 for an example of such a matrix on the small tree presented Figure 1.4.1. (See also Example 2.2.1 in Section 2.2.3 of Chapter 2).

*Example 1.4.1.* The incidence matrix of the tree presented below is:



$$\mathbf{U} = \begin{matrix} & \begin{matrix} X_1 & X_2 & X_3 & X_4 & X_5 & X_6 & X_7 & X_8 & X_9 \end{matrix} \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_6 \\ X_7 \\ X_8 \\ X_9 \end{matrix} & \left( \begin{array}{ccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{matrix} \Bigg\} \mathbf{T}$$

If we model a shifted BM, with a shift on the branch before (tip) node 7, and ancestral

value  $\boldsymbol{\mu}$ , then the shift matrix and the matrix of trait expectations  $\mathbb{E}[\mathbf{Y}]$  are:

$$\Delta = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} \begin{pmatrix} \boldsymbol{\mu}^T \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \boldsymbol{\delta}^T \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbb{E}[\mathbf{Y}] = \mathbf{T}\Delta = \begin{matrix} 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} \begin{pmatrix} \boldsymbol{\mu}^T \\ \boldsymbol{\mu}^T \\ \boldsymbol{\mu}^T + \boldsymbol{\delta}^T \\ \boldsymbol{\mu}^T \\ \boldsymbol{\mu}^T \end{pmatrix}$$

Using this, we get the two following propositions (See Section 2.2.3 for the derivation in the univariate case):

**Proposition 1.4.2** (Shifts in the Expectation - BM). *Assume that there are some shifts in the mean parameter  $\boldsymbol{\mu}$  of the BM on the  $K$  branches leading to nodes  $j_1, \dots, j_K$ . Then the model reduces to:*

$$\mathbf{Y} = \mathbf{T}\Delta + \mathbf{E}$$

where  $\mathbf{E}$  is the error vector with a structure induced by the BM (see Equation (1.4)), and  $\Delta$  is the matrix of shifts with all lines except  $(j_1, \dots, j_K)$  equal to zero.

**Proposition 1.4.3** (Shifts in the Optimal Value - OU). *Assume that there are some shifts in the optimal value parameter  $\beta$  of the OU on the  $K$  branches leading to nodes  $j_1, \dots, j_K$ . Then the model reduces to:*

$$\text{vec}(\mathbf{Y}^T) = (\mathbf{T} \otimes \mathbf{I}_p) \mathbf{W}(\mathbf{A}) \text{vec}(\Delta^T) + \text{vec}(\mathbf{E}^T)$$

where  $\mathbf{E}$  is the error vector with a structure induced by the OU (see Equation (1.6)), and  $\mathbf{W}(\mathbf{A})$  is the bloc-diagonal matrix of size  $(m+n)p \times (m+n)p$ :

$$\mathbf{W}(\mathbf{A}) = \begin{pmatrix} \mathbf{I}_p & & & \\ & \mathbf{I}_p - e^{-\mathbf{A}(h-t_{\text{pa}(2)})} & & \\ & & \ddots & \\ & & & \mathbf{I}_p - e^{-\mathbf{A}(h-t_{\text{pa}(m+n)})} \end{pmatrix}$$

Thanks to these propositions, including shifts in parameters  $\boldsymbol{\mu}$  (BM) or  $\beta$  (OU) reduces to including the columns corresponding to the branches where the shifts occur in the regression matrix  $\mathbf{X}$  of the linear regression framework. The only thing to be careful about are identifiability problems, as the full matrix  $\mathbf{T}$  is not of full rank. These questions are carefully studied in Section 2.3 of this document.

### 1.4.3.2 Automatic Shift Detection

When the position of the shifts is unknown, using the linear model written above, the problem of finding the branches where to add shifts reduces to a problem of variable selection: we need to find the lines of the coefficient matrix  $\Delta$  that are non-zero. We will exploit this vision of the problem in our methods, developed in Chapters 2 and 3.

Very recent work by [Khabbazian et al. \(2016\)](#) also used this formalism, using a lasso-based penalty ([Tibshirani, 1996](#)) to select for the non-zero lines of  $\mathbf{\Delta}$ , for a multivariate OU with independent traits. We will discuss this method further in Chapter 3.

Before that, a first method was designed by [Mahler et al. \(2013\)](#) (called SURFACE) to find shifts in the optimal parameter of an OU with independent traits. It is similar to a step-wise model selection procedure. This method has two phases. In the first, forward phase, some shifts are added one by one on the tree, until the AIC score does not improve anymore. The model is however parametrized in term of regimes, rather than in shifts, so that each regime has its own optimum value. In the second, backward phase, we try to merge some of the regimes together, in order to improve the AIC score by reducing the number of parameters. This backward phase is very interesting, as it tries to find some *convergent* regimes (see Section 1.1.2), i.e. species in different parts of the tree that reached a similar ecological niche, although through different historical paths. This is a question of interest to biologists, that can not be treated easily with our formalism in terms of a linear model. That is why, despite its flaws (it is quite slow, and this stepwise procedure does not guaranty any kind of optimal solution to the initial problem) SURFACE is widely used. In a Bayesian framework, other methods have been developed to find shifts on a univariate BM ([Eastman et al., 2013](#)) or OU ([Uyeda & Harmon, 2014](#)). We refer to Chapter 2 for a brief description of these methods.

#### 1.4.3.3 Other Kind of Shifts

Other than the mean or primary optimum values, some authors have considered shifts in the variance or selection strength. For the OU, [Beaulieu et al. \(2012\)](#) and [Clavel et al. \(2015\)](#) considered fixed, user defined shifts in all the parameters, respectively for an univariate or multivariate process. For the BM, [Eastman et al. \(2011\)](#) considered a Bayesian method to automatically detects some shifts in the variance  $\sigma^2$ . [BaMM Rabosky et al. \(2013\)](#); [Rabosky \(2014\)](#); [Rabosky et al. \(2014\)](#); [Shi & Rabosky \(2015\)](#) is an other rather popular software to detect shifts in the variance parameter. It is also designed to detect shifts in the speciation parameter, using a birth-death model of evolution, to explain the shape of the tree. The model used is rather complex, and has been criticized as possibly flawed by [Moore et al. \(2016\)](#). Those critics have been recently addressed by [Rabosky et al. \(2017\)](#). We only point these models to the interested reader, but we did not study them in depth as they use a different formalism.

### 1.4.4 Algorithms for Likelihood Computation

As pointed out in the previous section, the generality of the linear model framework is nice from a theoretical point of view, but does not solve the inference problem. We describe here a few methods that can be used to speed up the computations.

#### 1.4.4.1 The Pruning Algorithm

We already described the principles of the pruning algorithm in Section 1.2.3. In the particular case where the process is Gaussian, all the integrals written in Proposition 1.2.2 can be solved analytically, and hence the likelihood of a model can be computed in a single traversal of the tree, with a complexity of the order of  $n$  inversions of matrices of size  $p$  (the number of traits). We refer to Section 3.C.2 (upward step) for a comprehensive description of such an algorithm in a multivariate framework, with missing data.

The pruning algorithm can be seen as an adaptation on a tree of the “forward-backward” algorithm (used for instance in segmentation, see e.g. [Rabiner, 1989](#)). From its original description in phylogeny by [Felsenstein \(1973b\)](#), variants of this algorithm have been flowering in the literature, under different names: [Hadfield & Nakagawa \(2010\)](#), [Fitzjohn \(2012\)](#) (Gaussian Elimination Method), [Freckleton \(2012\)](#), [Lartillot \(2014\)](#) (phylogenetic Kalman filter), [Pybus et al. \(2012\)](#); [Cybis et al. \(2015\)](#) (in a Bayesian setting). In a non-Gaussian setting, [Landis et al. \(2013\)](#) and [Duchen et al. \(2017\)](#) adapted the algorithm to Lévy processes, while [Hiscott et al. \(2016\)](#) proposed an extension of this algorithm for a broad class of processes, based on some efficient numeric integral approximations.

#### 1.4.4.2 The “3-Point Structure” Algorithm

Another approach to likelihood computation was taken by [Ho & Ané \(2013a\)](#). For a linear (univariate) regression model as described in Section 1.4.2.2 (see Eq. (1.7)), the likelihood of the observed vector  $\mathbf{Y}$  at the tips can be written as:

$$-2 \log p(\mathbf{Y}) = n \log(2\pi) + \log |\mathbf{S}| + (\mathbf{Y} - \mathbf{X}\boldsymbol{\theta})^T \mathbf{S}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\theta}),$$

and the maximum likelihood estimator of the coefficients is given by:

$$\hat{\boldsymbol{\theta}} = (\mathbf{X}^T \mathbf{S}^{-1} \mathbf{X})^{-1} \mathbf{X}^T \mathbf{S}^{-1} \mathbf{Y}.$$

Computing these quantities requires two potentially time-consuming tasks: inverting matrix  $\mathbf{S}$ , and computing its determinant. When the tree is large (i.e. the dimension  $n$  of  $\mathbf{S}$  is large), these tasks might become overwhelming. However, [Ho & Ané \(2013a\)](#) noticed that  $\mathbf{S}^{-1}$  only appears in the likelihood as products of the form  $\mathbf{A}^T \mathbf{S}^{-1} \mathbf{B}$ , with  $\mathbf{A}$  and  $\mathbf{B}$  vector or matrices of adequate size. They hence designed an algorithm to compute these quantities efficiently, recalling that  $\mathbf{S}$  is a tree variance matrix, and hence has a special structure, as defined below.

**Definition 1.4.2** (3-point structure). A matrix  $\mathbf{S}$  has a 3-point structure if it is symmetric, with non-negative entries, and satisfies the following 3-point condition: for any  $i, j, k$  (not necessarily distinct), the two smallest of  $V_{ij}$ ,  $V_{ik}$  and  $V_{jk}$  are distinct.

It stems from the following theorem that all the models presented above that can be seen as a BM on a re-scaled tree, including the OU on an ultrametric tree, have the 3-point structure.

**Theorem 1.4.1** (BM on a re-scaled tree).  $\mathbf{S}$  is 3-point structured if and only if it is the covariance matrix of a random variable at the tips of some rooted tree under a BM model.

The OU on a non-ultrametric tree can be shown to have a *generalized* 3-point structure, and the algorithm can be extended to this case. For the sake of simplicity, we only present it here for the canonical 3-point structure. The algorithm uses a preorder of the tree, computing the needed quantities at each node, going from the tips to the root.

**Algorithm 1.4.2.** We assume that we are in the setting presented above, with a variance matrix  $\mathbf{S}$  having the 3-point structure. Assuming that the root has  $k$  child branches, this ensures that  $\mathbf{S}$  can be decomposed in the following way:

$$\mathbf{S} = t \mathbf{1}^T \mathbf{1} + \mathbf{A} \quad \text{with} \quad \mathbf{A} = \begin{pmatrix} \mathbf{S}_1 & & 0 \\ & \ddots & \\ 0 & & \mathbf{S}_k \end{pmatrix}$$



where  $t \geq 0$ , and  $\mathbf{A}$  is a block-diagonal matrix with each diagonal block matrix  $\mathbf{S}_s$  being the variance matrix induced by the sub-tree starting from child  $s$ ,  $1 \leq s \leq k$ .

The goal of the algorithm is then to compute the following quantities :

$$\begin{cases} d = \log|\mathbf{S}| \\ \mathbf{Q} = \mathbf{X}^T \mathbf{S}^{-1} \mathbf{Y} \end{cases} \quad \text{and} \quad \begin{cases} p = \mathbf{1}^T \mathbf{S}^{-1} \mathbf{1} \\ \hat{\boldsymbol{\mu}}_{\mathbf{Y}} = \mathbf{1}^T \mathbf{S}^{-1} \mathbf{Y} / p \\ \hat{\boldsymbol{\mu}}_{\mathbf{X}}^T = \mathbf{X}^T \mathbf{S}^{-1} \mathbf{1} / p \end{cases}$$

**Initialization** For a tree with a single tip of length  $t$ , we get:

$$\begin{cases} d = t \\ \mathbf{Q} = \mathbf{x}^T \mathbf{y} / t \end{cases} \quad \text{and} \quad \begin{cases} p = 1/t \\ \hat{\boldsymbol{\mu}}_{\mathbf{Y}} = \mathbf{y} \\ \hat{\boldsymbol{\mu}}_{\mathbf{X}}^T = \mathbf{x}^T \end{cases}$$

where  $\mathbf{x}$  and  $\mathbf{y}$  are the rows of  $\mathbf{X}$  and  $\mathbf{Y}$  corresponding to the current tips.

**Propagation** Recall the Woodbury formula and Sylvester's determinant for matrices of the form  $\mathbf{M} = \mathbf{A} + \mathbf{UCV}$ :

$$\begin{aligned} \mathbf{M}^{-1} &= \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{U} (\mathbf{C}^{-1} + \mathbf{V} \mathbf{A}^{-1} \mathbf{U})^{-1} \mathbf{V} \mathbf{A}^{-1} \\ |\mathbf{M}| &= |\mathbf{A}| |\mathbf{C}| |\mathbf{C}^{-1} + \mathbf{V} \mathbf{A}^{-1} \mathbf{U}| \end{aligned}$$

These formulas applied to our case give:

$$\mathbf{S}^{-1} = \mathbf{A}^{-1} - \frac{t}{1 + tp_A} \mathbf{A}^{-1} \mathbf{1} \mathbf{1}^T \mathbf{A}^{-1} \quad \text{where} \quad p_A = \mathbf{1}^T \mathbf{A}^{-1} \mathbf{1} = \sum_{s=1}^k p_s$$

and:

$$\begin{cases} d = \sum_{s=1}^k \log|\mathbf{S}_s| + \log(1 + tp_A) \\ \mathbf{Q} = \sum_{s=1}^k \mathbf{Q}_s - \frac{tp_A^2}{1 + tp_A} \hat{\boldsymbol{\mu}}_{\mathbf{X}}^T \hat{\boldsymbol{\mu}}_{\mathbf{Y}} \end{cases} \quad \text{and} \quad \begin{cases} p = \frac{p_A}{1 + tp_A} \\ \hat{\boldsymbol{\mu}}_{\mathbf{Y}} = \sum_{s=1}^k \frac{p_s}{p_A} \hat{\boldsymbol{\mu}}_{\mathbf{Y},s} \\ \hat{\boldsymbol{\mu}}_{\mathbf{X}}^T = \sum_{s=1}^k \frac{p_s}{p_A} \hat{\boldsymbol{\mu}}_{\mathbf{X},s}^T \end{cases}$$

**Finalization** At the root of the full tree, return  $\log|\mathbf{V}|$  and  $\mathbf{Q}$ .

## 1.4.5 Extensions

We browse here a few models and methods that extend the setting studied here, in one or several ways.

### 1.4.5.1 Measurement Errors and Intraspecific Variations

In all the methods presented in this manuscript, we assume that the variance structure is entirely dictated by the tree structure, and that there is only one observation per trait and per species. Indeed, in the regression model (1.7), the variance structure of the error vector  $\mathbf{E}$  is completely defined by the evolution model on the tree. This amounts to make



the strong assumption that the evolution process is the only source of variation in the trait dataset. Using the interpretation given in Section 1.4.1.3, this also means that we identify the mean value of a trait observed on several individuals of a given species with the secondary optimum of this species in the present.

We make this assumption for the sake of simplicity. It ignores at least two important sources of variations: measurement error and intraspecific variations. Measurement error can be a real issue, especially for large datasets, where the trait measurements are obtained from diverse sources, that sometimes lack precision (for instance, in [Rose et al., 2016](#), most of the 1090 moss shapes used are obtained through botanical drawings found in a vast and heterogeneous literature). Intraspecific variation is a more biological source of variation, and cannot be avoided, even assuming a perfect sampling. Several recent empirical and simulation studies found that these two phenomena could have a strong deleterious impact on phylogenetic comparative methods ([Silvestro et al., 2015](#); [Cooper et al., 2016](#)).

Using a mixed model framework such as the one presented in Section 1.4.2.2, these errors can be easily taken into account ([Lynch, 1991](#); [Ives et al., 2007](#); [Felsenstein, 2008](#)). We present here the main features of the model, and refer to Section 5.1 in Chapter 5 for a brief presentation of how this feature could be added to our framework. The formalism we use here is derived from [Felsenstein \(2008\)](#). We first assume that several individuals can be sampled independently for a given species. To cast this situation in the phylogenetic comparative methods framework, we can artificially mark each of the individuals of a given species  $A$  as separate “species”, numbered  $A_1, \dots, A_{n_A}$ , but with a zero length phylogenetic distance from one another:  $d_{A_i, A_j} = 0$  for any two individuals of the same species ( $1 \leq i, j \leq n_A$ ), and  $d_{A_i, B_j} = d_{A, B}$  for any two individuals of two different species. Using this convention, we can restrict ourselves without loss of generality to the situation described in Section 1.4.2.2, where  $p$  traits are measured on  $n$  species (where each species is in fact a couple “individual-species”). The intraspecific or measurement error is then obtained by adding a  $p$ -dimensional error vector term  $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$  to the observed vector of traits for each species: for any two species  $1 \leq i, j \leq n$  and two traits  $1 \leq k, l \leq p$ , the covariance between trait  $k$  of species  $i$  and trait  $l$  of species  $j$  is given by:

$$\text{Cov}[Y_{ik}; Y_{jl}] = E_{ijkl} + \delta_{ij}\epsilon_{kl}$$

where  $\delta_{ij} = 0$  if  $i \neq j$ , and  $\delta_{ij} = 1$  if  $i = j$ ; and  $\mathbf{E}$  is defined as in Section 1.4.2.2 by the evolution model used on the tree:  $\text{vec}(\mathbf{E}) \sim \mathcal{N}(\mathbf{0}, \mathbf{S})$ . Writing  $\mathbf{I}_n$  the  $n \times n$  identity matrix, we get a generalization of Equation (1.7):

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}(\text{vec}(\mathbf{X}\boldsymbol{\theta}), \mathbf{S} + \mathbf{Q} \otimes \mathbf{I}).$$

Note that, in the univariate BM case, this is equivalent to adding a terminal tip branch length  $\tau = q^2/\sigma^2$  to all the species. This makes this model close to Pagel’s  $\lambda$  transform ([Pagel, 1999](#)) exposed in Section 1.4.2.3: if the intraspecific variations are much larger than the phylogenetic ones, then the phylogenetic signal is weak, and vice versa. See Section 5.1 of Chapter 5 for another interpretation in terms of phylogenetic signal, but with the  $\alpha$  parameter of an OU.

For a BM model of evolution, this model was treated by [Ives et al. \(2007\)](#) with a known measurement error, and [Felsenstein \(2008\)](#) proposed an extended phylogenetic contrasts method to directly infer this error from the data. [Goolsby et al. \(2017\)](#) further extended the method to deal with missing data, and to allow for an OU model of evolution on an

ultrametric tree. This implementation, available as the R package `Rphylopars`, is quite fast, and relies on an adapted version of the 3-point structure algorithm (Ho & Ané, 2013a) presented in Section 1.4.4.2. In both articles, the estimation of the covariances matrices is done thanks to an EM algorithm. This EM algorithm, that is written in a matrix form, and uses  $\mathbf{E}$  and  $\boldsymbol{\epsilon}$  as hidden variables, is different from the one we develop in Chapters 2 and 3 (even with no shift).

In practice, we saw in Section 1.4.2.2 that the predictor variables could themselves be traits, that can also be measured with errors. Hansen & Bartoszek (2012) proposed an extended (univariate) OU framework to deal with the uncertainty associated with all the measured traits, whether they are considered as observations or predictors.

### 1.4.5.2 xxSSE Methods

One of our core assumption is that the trait evolves on a *fixed* tree, that is supposed to be known *a priori*. Implicitly, this amounts to assuming that the trait and the tree are independent, so that the trait has no influence on speciation. Note that, using the interpretation described in Section 1.4.1.3, the stochastic process models the secondary optima, and not the trait directly, so that this assumption can be justified.

Some methods were developed to model the very interactions between the trait and speciation rate. Maddison et al. (2007) first described a popular method to deal with binary traits, called BiSSE (for Binary State Speciation Extinction). In this model, a single binary trait evolves on a rooted ultrametric phylogenetic tree with branch lengths, that is assumed to be known and complete, i.e. all extant species have been sampled. The trait can take two values, 0 or 1. It follows a CTMC on the tree, with transition rates given by  $q_{01}$  and  $q_{10}$ . In addition, a given lineage with trait  $i$  ( $i \in \{0, 1\}$ ) has a speciation rate  $\lambda_i$ , and an extinction rate  $\mu_i$ . Conditionally on these parameters, the speciations and trait transitions are assumed to be independent.

Within this model, Maddison et al. (2007) are able to compute the likelihood of a set of parameters, using a postorder traversal of the tree (from tips to root) to update  $D_{n0}(t)$  and  $D_{n1}(t)$  the probabilities that a lineage beginning at time  $t$  with state 0 or 1 evolves into the clade observed from node  $n$ . Their approach relies on the numerical integration of a system of differential equations on each branch of the tree.

Several extensions of this model have been proposed. Fitzjohn et al. (2009) completed the BiSSE framework, allowing the tree to be incompletely sampled. To deal with discrete traits with more than two states, Fitzjohn (2012) developed the MuSSE model. A quantitative trait, evolving according to any diffusion process with a known transition density, can also be used in the QuaSSE model developed again by Fitzjohn (2010). This last extension requires the numerical computation of non-explicit integrals along the tree, and is reported to be quite slow. Finally, an adaptation of this framework to biogeographic models, coined GeoSSE, has been proposed by Goldberg et al. (2011).

### 1.4.5.3 Bayesian Methods

In all the methods presented above, the tree is supposed to be known and fixed, and the trait is modeled conditionally to it. However, the phylogenetic tree is itself the result of a complex statistical inference procedure, as we saw in Section 1.3.4. This two steps framework is not really satisfying, as it implies that we completely ignore the information we have about the uncertainty of the reconstructed tree, which can lead to several bias

in downstream analysis (see Section 5.1 for a study of the impact of for example, branch lengths misspecifications on the shifts detection analysis).

One way to alleviate this problem is to integrate both the tree reconstruction and the trait analysis in a single unified statistical framework. Because they are quite flexible, Bayesian methods can be well suited to conduct such a task. The idea was first exposed in Huelsenbeck & Rannala (2003), and refined in Lemey et al. (2010); Pybus et al. (2012); Cybis et al. (2015); Gill et al. (2016); Tolkoﬀ et al. (2017), to allow for more complex models of trait evolution and introduce some efficient likelihood computation algorithms. We briefly recall the main idea behind these methods, and refer the interested reader to these articles for more details on each model.

Assume that we have access to two different sources of data for a set of  $n$  species: a set of  $n$  aligned sequences  $\mathbf{S}$  and a  $n \times p$  matrix  $\mathbf{Y}$  of  $p$  continuous traits. We take two different models of evolution to model both data sets.  $\mathbf{S}$  is the result of a given process (typically, a CTMC, see Section 1.3.4) on a tree  $\tau$ , with parameters  $\nu$ .  $\mathbf{Y}$  is modeled by another process (such as a BM) on the same tree  $\tau$ , with parameters  $\theta$ . So that all the computations can be carried out, we make the crucial assumption that, conditionally on the phylogenetic tree  $\tau$ , the sequences  $\mathbf{S}$  and the traits  $\mathbf{Y}$  are *independent*:

$$p(\mathbf{S}, \mathbf{Y} | \tau, \nu, \theta) = p(\mathbf{S} | \tau, \nu) p(\mathbf{Y} | \tau, \theta).$$

The joint posterior distribution of the parameters given the data can hence be decomposed as:

$$p(\tau, \nu, \theta | \mathbf{S}, \mathbf{Y}) \propto p(\mathbf{Y} | \tau, \theta) p(\theta) p(\mathbf{S}, \tau, \nu)$$

where  $p(\theta)$  is a prior distributions on the parameters.

The two problems of sequence and trait modelling can hence be handled separately. The sequence study can be done with classical Bayesian inference tools (see e.g. Drummond et al., 2012), while the trait study benefits from many of the likelihood computation algorithms developed above. The methods mentioned above are implemented in the BEAST software (Drummond et al., 2012).

## 1.5 Model Selection

We have seen in Section 1.4.3 that the problem of shift detection could be recast into a problem of variable selection in a linear model framework. We review here some important results for model selection in this setting, that will be useful in the next chapters of this thesis. The exposition of this section is highly inspired from Giraud (2014) and Massart (2007), and we refer the interested reader to these two books for a more complete presentation of the topic.

### 1.5.1 Penalized Likelihood

#### 1.5.1.1 Statistical Setting

In this section, unless otherwise stated, we consider the following standard statistical setting:

$$\mathbf{Y} = \boldsymbol{\mu} + \sigma^2 \mathbf{E} \quad \text{with} \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n) \quad (1.14)$$

where  $\mathbf{Y}$  is a vector of  $n$  iid Gaussian observations, with unknown mean  $\boldsymbol{\mu}$  and variance  $\sigma^2$ . We focus here on the univariate Gaussian case for the clarity of exposition. Not all

the methods presented here have a natural extension to the multivariate setting, we will study some of them in Chapter 3.

We further assume that we have a collection of *models*  $\mathcal{S} = \{S_\eta, \eta \in \mathcal{M}\}$  that are linear subspaces  $S_\eta$  of  $\mathbb{R}^n$  indexed by a finite or countable set  $\mathcal{M}$ . We assume that for each model  $\eta \in \mathcal{M}$ , we have a maximum likelihood estimator  $\hat{\boldsymbol{\mu}}_\eta$  of  $\boldsymbol{\mu}$ . The following straightforward proposition allows us to see this estimator as a *projection* of  $\mathbf{Y}$  on  $S_\eta$ .

**Proposition 1.5.1** (Likelihood, Least-squares, and Projection). *The log-likelihood of a Gaussian vector  $\mathbf{Y} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}_n)$  is given by:*

$$\log p(\mathbf{Y}; \boldsymbol{\mu}, \sigma^2) = -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \|\mathbf{Y} - \boldsymbol{\mu}\|^2.$$

When  $\sigma^2$  is known, the maximum likelihood estimator of  $\hat{\boldsymbol{\mu}}_\eta$  in model  $\eta \in \mathcal{M}$  is given by:

$$\hat{\boldsymbol{\mu}}_\eta = \operatorname{argmax}_{\mathbf{s} \in S_\eta} \log p(\mathbf{Y}; \mathbf{s}, \sigma^2) = \operatorname{argmin}_{\mathbf{s} \in S_\eta} \|\mathbf{Y} - \mathbf{s}\|^2 = \operatorname{Proj}_\eta \mathbf{Y}.$$

When  $\sigma^2$  is unknown, its maximum likelihood estimate in model  $\eta \in \mathcal{M}$  is given by:

$${}^{ML} \hat{\sigma}_\eta^2 = \frac{1}{n} \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2$$

hence, injecting this expression in the likelihood, we get:

$$\hat{\boldsymbol{\mu}}_\eta = \operatorname{argmax}_{\mathbf{s} \in S_\eta} -\frac{n}{2} \log\left(\frac{1}{n} \|\mathbf{Y} - \mathbf{s}\|^2\right) = \operatorname{argmin}_{\mathbf{s} \in S_\eta} \|\mathbf{Y} - \mathbf{s}\|^2 = \operatorname{Proj}_\eta \mathbf{Y}.$$

Hence, for each  $\eta \in \mathcal{M}$ , whether  $\sigma^2$  is known or not, the maximum likelihood estimator of  $\boldsymbol{\mu}$  in model  $\eta$  is given by the least square estimator, and is the projection of  $\mathbf{Y}$  on  $S_\eta$ . We denote further by  $\boldsymbol{\mu}_\eta = \operatorname{Proj}_\eta \boldsymbol{\mu}$  the projection of the unknown mean  $\boldsymbol{\mu}$  on model  $S_\eta$ .

Note that the setting presented above includes the linear regression setting:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \sigma^2 \mathbf{E} \quad \text{with} \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$$

where, instead of estimating  $\hat{\boldsymbol{\mu}}_\eta \in \mathbb{R}^n$  living in some linear space  $S_\eta$ , we try to estimate  $\hat{\boldsymbol{\beta}}_\eta \in \mathbb{R}^p$  as being *sparse*, i.e. having a small number of non-zero coefficients (here,  $p$  is the number of regressors, and  $\mathbf{X}$  is an  $n \times p$  matrix). A model  $\eta$  is then defined by the position of the non-zero coefficients, and the associated linear subspace for the expectation of  $\mathbf{Y}$  is then given by the associated columns of the regression matrix  $\mathbf{X}$ :  $S_\eta = \operatorname{Span}\{\mathbf{X}_j : j \in \eta\}$ . In the rest of this section, we will stick to setting (1.14) for simplicity reasons, but it is worth keeping in mind that all the results presented here can also be applied to this linear regression framework.

The goal of *model selection* is to select a model  $\hat{\eta}$  among the collection  $\mathcal{M}$ , according to some criterion, and to study the properties of such a choice.

### 1.5.1.2 Risk and Oracle

Denote by  $D_\eta$  the dimension of model  $S_\eta$ , for  $\eta \in \mathcal{M}$ . It is straightforward to see that the higher the dimension of a model is, the better its likelihood or least-square score.

Indeed, when projecting on a space with a higher dimension, one gets more degree of freedom to adjust to the data, hence obtaining a better fit. Just taking the likelihood as a model selection criterion would therefore be inefficient, as it would amount in always choosing the model with the highest dimension.

The criterion that we would like to use is the *risk* of an estimator, as defined below:

**Definition 1.5.1** (Risk of an estimator). The *risk* of an estimator  $\hat{\boldsymbol{\mu}}_\eta$  for  $\eta \in \mathcal{M}$  is given by:

$$R(\hat{\boldsymbol{\mu}}_\eta) = \mathbb{E} \left[ \left\| \boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_\eta \right\|^2 \right] = \left\| \boldsymbol{\mu} - \boldsymbol{\mu}_\eta \right\|^2 + D_\eta \sigma^2$$

where we recognize in the second inequality the famous *bias-variance* decomposition of the risk.

When given a collection  $\mathcal{S}$  of models, the “best” model in term of risk is called the *oracle* estimator, defined as:

**Definition 1.5.2** (Oracle). The *oracle* estimator  $\hat{\boldsymbol{\mu}}_0$  is defined as the estimator minimizing the risk:

$$\hat{\boldsymbol{\mu}}_0 = \underset{\eta \in \mathcal{M}}{\operatorname{argmin}} R(\hat{\boldsymbol{\mu}}_\eta).$$

The oracle risk is a benchmark, as it is the best we can ever achieve using our collection of model  $\mathcal{S}$ . Note that, if the collection is poorly chosen, this oracle risk is not necessarily small.

### 1.5.1.3 Akaike’s Final Prediction Error (FPE)

Unfortunately, as  $\boldsymbol{\mu}$  is unknown, this oracle cannot be computed directly from the data. One natural approach is then to use an estimator of the risk. This is the path taken to derive Akaike’s Final Prediction Error (FPE [Akaike, 1969](#)), that uses the following unbiased estimator of the risk:

**Proposition 1.5.2** (FPE). *Given a model  $\eta \in \mathcal{M}$ , the expected least squares is:*

$$\mathbb{E} \left[ \left\| \mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta \right\|^2 \right] = \mathbb{E} \left[ \left\| (\mathbf{I} - \operatorname{Proj}_\eta)(\boldsymbol{\mu} + \mathbf{E}) \right\|^2 \right] = R(\hat{\boldsymbol{\mu}}_\eta) - D_\eta \sigma^2 + (n - D_\eta) \sigma^2.$$

*As the term  $n\sigma^2$  is identical for all models, it won’t play a role in the model selection. We hence use the following unbiased estimator of  $R(\hat{\boldsymbol{\mu}}_\eta) - n\sigma^2$ :*

$$\left\| \mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta \right\|^2 + 2D_\eta \hat{\sigma}_\eta^2$$

*where  $\hat{\sigma}_\eta^2 = \left\| \mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta \right\|^2 / (n - D_\eta)$  is an unbiased estimator of the variance. The FPE estimator is then given by:*

$$\hat{\eta}_{FPE} = \underset{\eta \in \mathcal{M}}{\operatorname{argmin}} \left\{ \left\| \mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta \right\|^2 \left( 1 + \frac{2D_\eta}{n - D_\eta} \right) \right\}$$

When the variance  $\sigma^2$  is known, this criterion is equivalent to the famous Akaike Information Criterion, that we introduce in the next section. It is only guaranteed to work *in expectation*, or in the limit case where  $n$  goes to  $+\infty$ . When the number of models grows exponentially with the dimension  $d$  (i.e. the set of models  $S_\eta$  with dimension  $D_\eta = d$  grows very fast), then this criterion cannot handle the variance of the estimated risks: just by chance, a bad model with a high dimension is likely to have a small risk, and hence be favored against the oracle. This criterion has hence a tendency to select models with a high dimension ([Giraud, 2014](#), Chap. 2).

### 1.5.1.4 Penalized Likelihood and Least Squares

The criterion we derived above has the form of a *penalized* least-squares, with penalty  $2D_\eta/(n - D_\eta)$ : when the dimension of the model  $D_\eta$  raises, then the least squares will go down, but the penalty will raise. This criterion on the least squares can be re-written in term of a penalty on the log-likelihood, using Proposition 1.5.1. In the following, we will use the following equivalent criteria, on the least-squares or the likelihood:

$$\text{Crit}_{lsq}(\eta) = \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 \left(1 + \frac{\text{pen}(\eta)}{n - D_\eta}\right) \quad (1.15)$$

$$\text{Crit}_{ll}(\eta) = \frac{n}{2} \log \left( \frac{\|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2}{n} \right) + \frac{1}{2} \text{pen}'(\eta) \quad (1.16)$$

$$\text{pen}'(\eta) = n \log \left(1 + \frac{\text{pen}(\eta)}{n - D_\eta}\right) \quad (1.17)$$

This equivalency between criteria on the least-squares and the log-likelihood allows us to see the problem of model selection using two complementary approaches. However, this does not hold anymore in the multivariate setting, and we will see in Chapter 3 that only the formulation in term of least squares has a natural extension.

### 1.5.1.5 Akaike's Information Criterion (AIC)

The Akaike's Information Criterion (AIC) can be seen as a criterion on the likelihood, approximately equivalent to the FPE for  $n$  large, using Equation (1.17):

$$\begin{aligned} \text{pen}'_{\text{AIC}}(\eta) &= n \log \left(1 + \frac{\text{pen}_{\text{FPE}}(\eta)}{n - D_\eta}\right) = n \log \left(1 + \frac{2D_\eta}{n - D_\eta}\right) \\ &\approx n \frac{2D_\eta}{n - D_\eta} \approx 2D_\eta. \end{aligned}$$

This famous and widely used criterion inherits the flaws described for the FPE, and is only valid asymptotically. It can also be seen as an approximation for a Kullback-Leibler based Information criterion (hence its name). In the next sections, we describe some other criteria that try to alleviate these flaws. But first, let's recall another famous information criterion, based on a Bayesian paradigm.

### 1.5.1.6 Bayesian Information Criterion (BIC)

The Bayesian Information Criterion (BIC) can be expressed in our framework, taking in (1.16):

$$\text{pen}'_{\text{BIC}}(\eta) = D_\eta \log(n).$$

Its derivation relies on a Laplace approximation of the Bayes factors. We briefly recall here the construction, for a finite collection of models  $\mathcal{M}$  (see Lebarbier & Mary-Huard 2006 for a thorough introduction to the BIC construction and properties). In a Bayesian paradigm, each model  $\eta$  has a prior probability  $\pi_\eta$  to be chosen, and then the parameters  $\boldsymbol{\theta} = (\boldsymbol{\mu}, \sigma^2)$  of the distribution are drawn from an *a priori* distribution  $p(\boldsymbol{\theta} \mid \eta)$ . The



model selection is then based on the posterior distribution of a model  $p(\eta \mid \mathbf{Y})$  given the observations:

$$\hat{\eta}_{\text{BIC}} = \operatorname{argmax}_{\eta \in \mathcal{M}} p(\eta \mid \mathbf{Y}) = \operatorname{argmax}_{\eta \in \mathcal{M}} p(\mathbf{Y} \mid \eta) \pi_{\eta} = \operatorname{argmax}_{\eta \in \mathcal{M}} p(\mathbf{Y} \mid \eta),$$

where we used Bayes formula, and assumed a uniform prior  $\pi_{\eta} = 1/|\mathcal{M}|$  on all models  $\eta \in \mathcal{M}$ . To estimate the marginal likelihood of the data given the model, we write:

$$p(\mathbf{Y} \mid \eta) = \int_{\boldsymbol{\theta} \in \mathcal{S}_{\eta}} p(\mathbf{Y} \mid \boldsymbol{\theta}, \eta) p(\boldsymbol{\theta} \mid \eta) d\boldsymbol{\theta},$$

and assume that  $p(\mathbf{Y} \mid \boldsymbol{\theta}, \eta)$  is concentrated around its maximum  $p(\mathbf{Y} \mid \boldsymbol{\theta}_{\eta}^*, \eta)$ . The Laplace approximation, recalled below, then allows us to find the penalty stated above, dropping all the terms that are constant with  $n$ , and approximating  $\boldsymbol{\theta}_{\eta}^*$  by the maximum likelihood estimator  $\hat{\boldsymbol{\theta}}_{\eta}$ .

**Proposition 1.5.3** (Laplace Approximation). *Let  $L : \mathbb{R}^q \rightarrow \mathbb{R}$  be a twice differential function on  $\mathbb{R}^q$ , with a unique maximum  $u^*$ . Then:*

$$\int_{\mathbb{R}^q} e^{nL(u)} du \approx \left(\frac{2\pi}{n}\right)^{q/2} |-L''(u^*)|^{-1/2} e^{nL(u^*)} \quad \text{for } n \rightarrow \infty.$$

## 1.5.2 Model selection à la Birgé & Massart (2001)

In this section, we offer a first alternative to the well known but limited model selection criteria we recalled above. Assuming that the variance  $\sigma^2$  is known, we are able to exhibit a model selection criterion with non-asymptotic guaranties, in the form of an *oracle inequality*. We then present a famous heuristic to extend the penalty to a case where the variance is not known.

### 1.5.2.1 A non-asymptotic Oracle inequality

In this section, we restrict ourselves to the setting presented in Equation (1.14), with a *known* variance  $\sigma^2$ . In that case, we do not need to control for the extra variation induced by the estimation of the variance, and the criterion on the least squares can be written as:

$$\text{Crit}(\eta) = \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_{\eta}\|^2 + \sigma^2 \text{pen}(\eta). \quad (1.18)$$

The following theorem can be proven using some *concentration inequalities*, i.e. inequalities that control the dispersion of a random variable around its expectation (see Massart 2007 for an introduction to classical concentration inequalities).

**Theorem 1.5.1** (Birgé & Massart 2001; Massart 2007). *Assume that the model is defined as in (1.14), and that  $\hat{\eta}$  is a minimizer of criterion (1.18). Let  $\{L_{\eta}\}_{\eta \in \mathcal{M}}$  be some family of positive numbers such that*

$$\sum_{\eta \in \mathcal{M}} e^{-L_{\eta} D_{\eta}} = \Omega < \infty \quad (1.19)$$

Let  $A > 1$  and further assume that:

$$\sigma^2 \text{pen}(\eta) \geq A\sigma^2 \left( \sqrt{D_{\eta}} + \sqrt{2L_{\eta} D_{\eta}} \right)^2. \quad (1.20)$$

Then the selected least-square estimator  $\hat{\boldsymbol{\mu}}_{\hat{\eta}}$  satisfies the following risk bound:

$$\mathbb{E} \left[ \|\hat{\boldsymbol{\mu}}_{\hat{\eta}} - \boldsymbol{\mu}\|^2 \right] \leq C(A) \left[ \inf_{\eta \in \mathcal{M}} \left( \|\boldsymbol{\mu} - \boldsymbol{\mu}_{\eta}\|^2 + \sigma^2 \text{pen}(\eta) \right) + (1 + \Omega)\sigma^2 \right] \quad (1.21)$$

where  $C(A)$  depends only on  $A$ .

Inequality (1.21) is an oracle risk bound on the risk of the selected estimator. It means that the selected estimator *almost* performs as well as the best model in term of compromise between the bias ( $\|\boldsymbol{\mu} - \boldsymbol{\mu}_{\eta}\|^2$ ) and the model complexity dependent variance ( $\sigma^2 \text{pen}(\eta)$ ). Note that constant  $C(A)$  is not made explicit here, and can be quite large. Finding the constant that is closest to 1 (finding a so called “sharp” risk bound) is an active field of research in its own. Note that this inequality is non-asymptotic: it is true even for a small number of observations  $n$ , even though the infimum on the right hand side might not be small enough in this case.

What makes this approach working better than the FPE or AIC criterion, is that the penalty takes into account the possible growing complexity of the collection of models with a given dimension  $D$ , through the weights  $(L_{\eta})_{\eta \in \mathcal{M}}$ . When facing a problem with its associated collection of models  $\mathcal{M}$ , one needs, in order to apply this inequality, to craft the right set of weights satisfying inequality (1.19). We show in Section 1.5.2.2 an example of such a derivation in the setting of coordinate sparsity, that looks like the one we will face in the next chapters.

Finally, remark that the condition (1.20) on the penalty depends on the variance  $\sigma^2$ , that is, in practice unknown. It is then tempting to apply this criterion with a constant  $\kappa = A\sigma^2$ , getting rid of the unknown variance. Doing that, we loose the natural scale that was imposed on  $A$  ( $A > 1$ ), and we need to calibrate the constant  $\kappa$  correctly. We will briefly present some heuristics to do that in the last subsection on this method (Section 1.5.2.3).

### 1.5.2.2 Derivation in the Coordinate Sparse Setting

In the *coordinate sparse setting*, we assume that the true model has only a few non-zero coordinates. The associated collection of models can be defined as follow:

**Definition 1.5.3** (Coordinate-Sparse Setting). Let  $(\mathbf{e}_i)_{1 \leq i \leq n}$  be the canonical basis of  $\mathbb{R}^n$ . The collection of models  $\{S_{\eta}, \eta \in \mathcal{M}\}$  is indexed by  $\mathcal{P}(\llbracket 1, n \rrbracket)$  the set of all the subsets of  $\llbracket 1, n \rrbracket$ , and for any  $\eta \in \mathcal{M}$ , the linear subspace  $S_{\eta}$  is defined by the vectors  $\mathbf{s}$  in  $\mathbb{R}^n$  such that  $s_i = 0$  for any  $i \notin \eta$ , i.e.  $S_{\eta} = \text{Span}\{\mathbf{e}_j, j \in \eta\}$ . Note that the complexity of the models with dimension  $D$  is:  $|\{\eta \in \mathcal{M} : |\eta| = D\}| = \binom{n}{D}$ .

In this setting the sum featured in inequality (1.19) can be written as:

$$\Omega = \sum_{\eta \in \mathcal{M}} e^{-L_{\eta} D_{\eta}} = \sum_{D \geq 0} \binom{n}{D} e^{-L_D D}.$$

Assuming that  $L_{\eta}$  depends only on the dimension of  $S_{\eta}$  (i.e.  $L_{\eta} = L_D, \forall |\eta| = D$ ), a natural choice of weights could then be:

$$L_D = \frac{n}{D} \log \left( 1 + \frac{1}{n} \right) + \log(n),$$



leading to a sum controlled by:

$$\Omega = \sum_{D \geq 0} \binom{n}{D} \left(1 + \frac{1}{n}\right)^{-n} \left(\frac{1}{n}\right)^D = \left(1 + \frac{1}{n}\right)^{-n} \left(\frac{1}{n} + 1\right)^n = 1.$$

Using the inequality  $n \log\left(1 + \frac{1}{n}\right) \leq 1$ , this would lead to the penalty:

$$AD \left(1 + \sqrt{2L_D}\right)^2 \leq AD \left(1 + \sqrt{2} \sqrt{\frac{1}{D} + \log(n)}\right)^2 =: \text{pen}(D), \quad (1.22)$$

that complies with inequality (1.20).

Another natural choice of weights might be, for  $B > 0$ :

$$L_D = B + \frac{1}{D} \log\left(\frac{n}{D}\right)$$

allowing us to control the sum:

$$\Omega = \sum_{D \geq 0} \exp\left[-L_D D + \log\left(\frac{n}{D}\right)\right] \leq \frac{1}{1 - e^{-B}}.$$

To comply with inequality (1.20), we can then use:

$$AD \left(1 + \sqrt{2L_D}\right)^2 \leq AD \left(1 + \sqrt{2} \sqrt{B + 1 + \log\left(\frac{n}{D}\right)}\right)^2 =: \text{pen}(D), \quad (1.23)$$

where we used the classical inequality:  $\log\left(\frac{n}{D}\right) \leq D \left(1 + \log\left(\frac{n}{D}\right)\right)$ . This choice gives us another “degree of freedom”  $B$  that might be calibrated to improve the efficiency of the penalty.

Depending on the collection of models at hand, such a derivation needs to be adapted. We refer to [Lebarbier \(2005\)](#) for an example of application in the model selection problem associated with segmentation.

### 1.5.2.3 Unknown Variance: the Slope Heuristic

As pointed out above, when  $\sigma^2$  is unknown, the penalty can be written as  $\kappa \text{pen}_{\text{sh}}(\eta)$ , where  $\text{pen}_{\text{sh}}$  is the *penalty shape*, defined by inequality (1.20) (dropping the constants), and  $\kappa = A\sigma^2$  is the constant that needs to be calibrated. One popular approach is to use the *slope heuristic*, developed by [Birgé & Massart \(2007\)](#); [Arlot & Massart \(2009\)](#); [Baudry et al. \(2012\)](#). We roughly explain the idea behind this heuristic here, and refer the interested reader to the papers cited for a description of the theoretical foundation, and the practical computability of such a method.

Putting together Definition 1.5.2 and Equation (1.18), an *oracle penalty*, defined for  $\eta \in \mathcal{M}$  as  $R(\hat{\boldsymbol{\mu}}_\eta) - \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|$ , would always select for the oracle estimator. This can be decomposed as:

$$\begin{aligned} \text{pen}_{\text{opt}}(\eta) &= R(\hat{\boldsymbol{\mu}}_\eta) - \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 \\ &= \mathbb{E}\left[\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_\eta\|^2\right] - \|\boldsymbol{\mu} - \boldsymbol{\mu}_\eta\|^2 &&= v_\eta \\ &\quad + \|\boldsymbol{\mu} - \boldsymbol{\mu}_\eta\|^2 - \|\mathbf{Y} - \boldsymbol{\mu}_\eta\|^2 + \|\mathbf{Y} - \boldsymbol{\mu}\|^2 &&= \delta_\eta \\ &\quad + \|\mathbf{Y} - \boldsymbol{\mu}_\eta\|^2 - \|\mathbf{Y} - \boldsymbol{\mu}\|^2 - \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 &&= \hat{v}_\eta \end{aligned}$$

Note that  $\mathbb{E}[\delta_\eta] = 0$ , and  $\mathbb{E}[\hat{v}_\eta] = v_\eta = D_\eta$ , so that the FPE penalty is obtained by taking the expectation of this optimal penalty. The slope heuristic then stems from the following observations:

- $\text{pen}_{\min}(\eta) = \hat{v}_\eta$  can be seen as a “minimal” penalty, as the associated criterion is  $\text{Crit}_{\min}(\eta) = \|\mathbf{Y} - \boldsymbol{\mu}_\eta\|^2 - \|\mathbf{Y} - \boldsymbol{\mu}\|^2$ , and would select for a model simply minimizing the bias, hence always selecting the model with the highest dimension. The penalty is then minimal, as any larger penalty would result in selecting a model with possibly a smaller dimension.
- $\text{pen}_{\text{opt}}(\eta) \approx 2\hat{v}_\eta$ : because one can expect that  $v_\eta \approx \hat{v}_\eta$ , and that  $\delta_\eta \approx 0$  (provided these quantities concentrate around their expectations). Hence, the optimal penalty is about twice the minimal penalty.
- $\text{pen}_{\text{opt}}(\eta) = \kappa_{\text{opt}} \text{pen}_{\text{sh}}(\eta)$ : according to our assumption, the optimal penalty can be expressed as a constant times our penalty shape. Putting this together with the previous remarks, we get:

$$-\|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 \approx -\left(\|\mathbf{Y} - \boldsymbol{\mu}_\eta\|^2 - \|\mathbf{Y} - \boldsymbol{\mu}\|^2\right) + \frac{\kappa_{\text{opt}}}{2} \text{pen}_{\text{sh}}(\eta)$$

But, for large dimensions,  $\left(\|\mathbf{Y} - \boldsymbol{\mu}_\eta\|^2 - \|\mathbf{Y} - \boldsymbol{\mu}\|^2\right)$  should be approximately constant, as the bias is expected to be constant for most complex models. Hence, asymptotically, the least squares scores should grow linearly with  $\text{pen}_{\text{sh}}(\eta)$ , with a slope approximately equal to  $\frac{\kappa_{\text{opt}}}{2}$ . Hence, an estimator  $\hat{\kappa}$  can be obtained as half the asymptotic slope when plotting the least squares scores against the value of the penalty shape.

This heuristic can be applied to any relevant contrast (e.g. least squares or log-likelihood), and gives some good results in practice, when a theoretical penalty shape can be derived. It is implemented in the R-package `capushe` (Brault et al., 2012).

### 1.5.3 The Lasso Penalty

The very popular Lasso regularization procedure was introduced by Tibshirani (1996), and is based on an  $\ell_1$ -norm penalty of the least squares. It can be seen as a convex relaxation of the previous penalty, making it particularly efficient. We sketch here the main steps of its derivation in our setting, and show how it can be used to get several sparsity settings. The exposition adopted here is highly inspired from Giraud (2014), and we refer to this book for a deeper introduction of the method, including some geometrical and algebraic insights.

#### 1.5.3.1 A convex relaxation

For the coordinate-sparse setting defined in 1.5.3, we saw in Equation (1.22) that the penalized criterion (1.18) was of the order of:

$$\text{Crit}(\eta) = \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 + \lambda |\eta| \quad \text{with} \quad \lambda = A\sigma^2 \left(1 + \sqrt{2\log(n)}\right)^2.$$

Denote by  $\text{Supp}(\mathbf{x}) = \{j : \mu_j \neq 0\}$  the support of a vector  $\mathbf{x} \in \mathbb{R}^n$ . Then, for  $\eta \in \mathcal{M}$ , the linear subspace  $S_\eta$  of Definition 1.5.3 can be written as  $S_\eta = \{\mathbf{x} : \text{Supp}(\mathbf{x}) = \eta\}$ . Hence,

the least squares estimator for model  $\eta$  is equal to  $\hat{\boldsymbol{\mu}}_\eta = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n: \operatorname{Supp}(\mathbf{x})=\eta} \|\mathbf{Y} - \mathbf{x}\|^2$ . The selected model minimizing the criterion can hence be written as:

$$\hat{\eta} = \operatorname{argmin}_{\eta \in \mathcal{M}} \min_{\mathbf{x} \in \mathbb{R}^n: \operatorname{Supp}(\mathbf{x})=\eta} \left\{ \|\mathbf{Y} - \mathbf{x}\|^2 + \lambda \|\mathbf{x}\|_0 \right\},$$

where  $\|\mathbf{x}\|_0$  is the  $\ell_0$ -norm of  $\mathbf{x}$ , defined as the number of non-zero coordinate of  $\mathbf{x}$  (i.e.  $\|\mathbf{x}\|_0 = |\operatorname{Supp}(\mathbf{x})|$ ). This leads to the following definition of the selected estimator:

$$\hat{\boldsymbol{\mu}}_{\hat{\eta}} = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|\mathbf{Y} - \mathbf{x}\|^2 + \lambda \|\mathbf{x}\|_0 \right\}.$$

Because of the term in  $\|\mathbf{x}\|_0$ , the minimization of the previous criterion is hard, and one might need to test all the possible models, yielding a combinatorial complexity, that can swiftly become overwhelming for large dimensions. One way to circumvent this problem is to replace this non-convex  $\ell_0$ -norm by a  $\ell_1$ -norm, defined as  $\|\mathbf{x}\|_1 = \sum_{j=1}^n |x_j|$  for  $\mathbf{x} \in \mathbb{R}^n$ . For  $\lambda > 0$ , define the lasso estimator as:

$$\hat{\boldsymbol{\mu}}_\lambda = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^n} \left\{ \|\mathbf{Y} - \mathbf{x}\|^2 + \lambda \|\mathbf{x}\|_1 \right\}. \quad (1.24)$$

Thanks to this new penalty based on the convex  $\ell_1$ -norm, the criterion to optimize is itself convex, and hence can benefit from all the standard minimization techniques, that are known to be fast and efficient in this convex setting.

In addition to this nice computational feature, it can be shown that, choosing  $\lambda_{\text{opt}} = 3\sigma\sqrt{2\log(n) + 2L}$  (for any  $L > 0$ ), the estimator  $\hat{\boldsymbol{\mu}}_{\lambda_{\text{opt}}}$  defined by Equation (1.24) satisfies an oracle inequality (see Cor. 4.3 in Giraud, 2014). This property gives us some guaranties on the theoretical behavior of  $\hat{\boldsymbol{\mu}}_\lambda$ , even though, as the variance is unknown,  $\lambda_{\text{opt}}$  cannot be used in practice. The problem of selecting  $\lambda$  is not easy, and can be solved with cross validation, or with an estimator selection procedure (Baraud et al., 2010).

The  $\ell_1$ -norm penalty allows us to select a sparse model, but, as it changes the criterion to optimize, it introduces a bias in the estimation of the coefficients, that appears shrunked in the standard lasso solution. One way to circumvent this problem is to use the so-called *Gauss-lasso* estimator of the coefficient, defined as:

$$\hat{\boldsymbol{\mu}}_\lambda = \operatorname{Proj}_{\hat{S}_\lambda} \mathbf{Y} \quad \text{where} \quad \hat{S}_\lambda = S_{\hat{\eta}_\lambda}.$$

In other words, the lasso penalty is used to select a model, defined by the non-zero components of the estimator, and then we use the standard least-square estimator in this model.

### 1.5.3.2 Sparsity Patterns

The lasso penalty above was derived in the context of coordinate sparsity. To deal with other kinds of sparsity patterns, it needs to be adapted. We present here some classical penalties to achieve group, sparse-group, and variation sparsity.

**Group Sparsity.** Here, we assume that the coordinates of  $\boldsymbol{\mu}$  are separated into meaningful groups. The group-sparsity is designed to select for *groups* of coefficients that are simultaneously non-zeros, rather than single coefficients one by one. As an example,

we show below how re-framing a multivariate regression to a univariate one using the vectorization can lead to such a sparsity pattern.

Assume that the coordinates of  $\boldsymbol{\mu}$  are split into  $p$  groups  $\{G_k\}_{1 \leq k \leq p}$  that form a partition of  $\llbracket 1, n \rrbracket$ . For  $\boldsymbol{\lambda} = (\lambda_k)_{1 \leq k \leq p}$  a vector of positive coefficients, the group-lasso estimator  $\hat{\boldsymbol{\mu}}_{\boldsymbol{\lambda}}$  is defined by (Yuan & Lin, 2006):

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{\lambda}} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \mathbf{x}\|^2 + \sum_{k=1}^p \lambda_k \|\mathbf{x}_{G_k}\| \right\},$$

where  $\|\cdot\|$  represents the euclidean norm, and  $\mathbf{x}_{G_k}$  the vector with coefficients of  $\mathbf{x}$  that are in  $G_k$ . Some theoretical properties can also be shown on this estimator (see Giraud, 2014).

Let's now give an example of such a setting, considering the *multivariate* regression framework. Assume that  $\mathbf{Y}$  is now a  $n \times p$  matrix, with all its coefficients i.i.d. Taking the setting of Proposition 1.4.2 (on a star-tree, with  $\mathbf{R} = \sigma^2 \mathbf{I}_p$ ), this model can be written as:

$$\mathbf{Y} = \mathbf{T}\boldsymbol{\Delta} + \mathbf{E}$$

where  $\mathbf{E}$  is an error vector,  $\mathbf{T}$  is a regression matrix (size  $n \times (m+n)$ ) that represents the tree-structure, and  $\boldsymbol{\Delta}$  is the matrix of shifts (size  $(m+n) \times p$ ), each line  $\boldsymbol{\Delta}^j$  representing the vector of shifts values of the  $p$  traits on branch  $j$ . To select for a small number of shifts in this setting, we need to select for a small number of non-zero *lines* of  $\boldsymbol{\Delta}$ . Writing the vectorized form of the problem, we get:

$$\operatorname{vec}(\mathbf{Y}) = (\mathbf{I}_p \otimes \mathbf{T}) \operatorname{vec}(\boldsymbol{\Delta}) + \operatorname{vec}(\mathbf{E}) \quad \text{with} \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_{np}).$$

To select for the non-zero lines of matrix  $\boldsymbol{\Delta}$ , we need to group the coefficients of vector  $\operatorname{vec}(\boldsymbol{\Delta})$  according to their original line, so that they are all set to zero at the same time when needed. For  $1 \leq i \leq n$ , the group  $G_i = \{kn + i : 0 \leq k \leq p-1\}$  represents the coordinates of the elements of line  $i$  in the vectorized space (with dimension  $np$ ). See Figure 1.5.1 for a graphical representation of this group. The group-sparsity constraint would hence try to select for such non-zero groups, hence selecting for non-zeros lines.

Note that here, to frame the model into the correct framework (i.i.d. vector), we had to assume a star tree, and an independent BM model. We show in Section 3.C.3 how to deal with the correlations induced by a real tree and a multivariate BM.

**Sparse Group Sparsity.** In this setting, we further assume that only a few coefficients in each of the selected groups are non-zeros. The associated estimator is obtained with a similar penalty, just adding an overall sparsity constraint (Simon et al., 2013):

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{\lambda}, \delta} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \mathbf{x}\|^2 + \sum_{k=1}^p \lambda_k \|\mathbf{x}_{G_k}\| + \delta \|\mathbf{x}\|_1 \right\},$$

**Fused Lasso: Variation Sparsity.** In this setting, we assume that coefficients are in a meaningful order and do not change a lot, i.e. that the difference  $\mu_i - \mu_{i+1}$  is often equal to 0. This leads to the following criterion (Tibshirani et al., 2005):

$$\hat{\boldsymbol{\mu}}_{\boldsymbol{\lambda}} = \underset{\mathbf{x} \in \mathbb{R}^n}{\operatorname{argmin}} \left\{ \|\mathbf{Y} - \mathbf{x}\|^2 + \lambda \sum_{j=1}^{n-1} |x_{j+1} - x_j| \right\}.$$

Up to a re-parametrization, this problem is equivalent to the standard lasso.

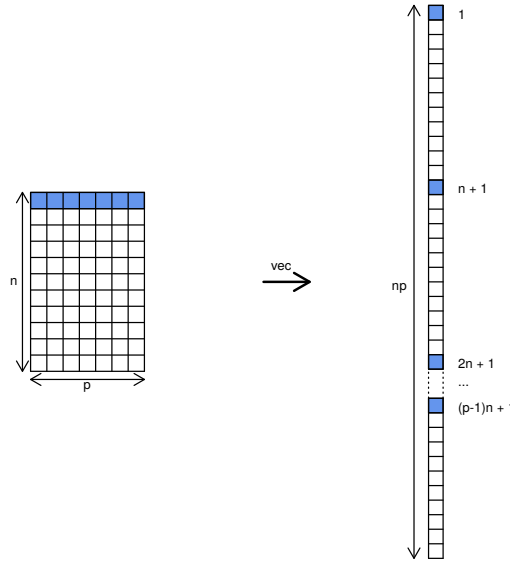


Figure 1.5.1 – A matrix (left) and its vectorized form (right), obtained by stacking all the columns. The first line of the matrix (in blue) is dispersed on the vector, forming a group  $G_1 = \{kn + 1 : 0 \leq k \leq p - 1\}$  of coordinates.

#### 1.5.4 Model selection with LINselect

We go back to the setting stated in (1.14), with unknown variance, and criterion as defined in (1.15) or (1.16). The goal here is to derive a penalty that does not depend on the unknown variance. Note that criterion (1.15) can be re-written as:

$$\text{Crit}_{lsq}(\eta) = \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 + \hat{\sigma}_\eta^2 \text{pen}(\eta)$$

so that it is similar to criterion (1.18) used in the previous section, only replacing the unknown variance  $\sigma^2$  by an unbiased estimator  $\hat{\sigma}_\eta^2 = \|\mathbf{Y} - \hat{\boldsymbol{\mu}}_\eta\|^2 / (n - D_\eta)$ .

##### 1.5.4.1 A non-asymptotic Oracle inequality

Baraud et al. (2009) derived an oracle inequality in this Gaussian setting with unknown variance. The penalty they used is designed to control exactly the extra variations induced by the plug-in of the variance estimator in the criterion. It relies on the following function:

**Definition 1.5.4** (Baraud et al., 2009, Def. 2 and 3). Let  $D, N$  be two positive numbers, and  $X_D, X_N$  be two independent  $\xi^2$  random variables with degrees of freedom  $D$  and  $N$  respectively. For  $x \leq 0$ , we define

$$\text{Dkhi}[D, N, x] = \frac{1}{\mathbb{E}[X_D]} \mathbb{E} \left[ \left( X_D - x \frac{X_N}{N} \right)_+ \right]$$

And, for  $0 < q \leq 1$  we define  $\text{EDkhi}[D, N, q]$  as the unique solution of the equation

$$\text{Dkhi}[D, N, \text{EDkhi}[D, N, q]] = q$$

**Theorem 1.5.2** (Baraud et al. 2009, Th. 2 and Cor. 1). *In the setting defined above, assume that  $n - D_\eta \geq 2$  for any  $\eta \in \mathcal{M}$ , and let  $\hat{\eta}$  be a minimizer of (1.15) or (1.16). Let  $\{L_\eta\}_{\eta \in \mathcal{M}}$  be some family of positive numbers such that:*

$$\sum_{\eta \in \mathcal{M}} (D_\eta + 1)e^{-L_\eta} = \Omega' < +\infty. \quad (1.25)$$

Let  $A > 1$  and further assume that

$$\text{pen}(\eta) = \text{pen}_{A, \mathcal{L}}(\eta) = A \frac{n - D_\eta}{n - D_\eta - 1} \text{EDkhi}[D_\eta + 1, n - D_\eta - 1, e^{-L_\eta}].$$

Then the selected estimator  $\hat{\boldsymbol{\mu}}_{\hat{\eta}}$  satisfies the following risk bound:

$$\mathbb{E} \left[ \frac{\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{\hat{\eta}}\|^2}{\sigma^2} \right] \leq \frac{A}{A-1} \inf_{\eta \in \mathcal{M}} \left\{ \frac{\|\boldsymbol{\mu} - \boldsymbol{\mu}_\eta\|^2}{\sigma^2} \left( 1 + \frac{\text{pen}(\eta)}{n - D_\eta} \right) + \text{pen}(\eta) - D_\eta \right\} + 2A^2 \frac{\Omega'}{A-1}.$$

In addition, if  $\kappa < 1$ ,  $n - D_\eta \geq 7$  and  $\max(L_\eta, D_\eta) \leq \kappa n$  for any  $\eta \in \mathcal{M}$ , then:

$$\mathbb{E} \left[ \frac{\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{\hat{\eta}}\|^2}{\sigma^2} \right] \leq C(A, \kappa) \left[ \inf_{\eta \in \mathcal{M}} \left\{ \frac{\|\boldsymbol{\mu} - \boldsymbol{\mu}_\eta\|^2}{\sigma^2} + \max(L_\eta, D_\eta) \right\} + \Omega' \right]. \quad (1.26)$$

This theorem looks like Theorem 1.5.1 that we recalled above. Note that the condition on the weights (1.25) is similar but not identical to previous condition (1.19). Inequality (1.26) is an oracle inequality as soon as the weights  $L_\eta$  can be taken of the order of  $D_\eta$  for all  $\eta \in \mathcal{M}$ . As previously, those weights need to be calibrated for each model collection one considers (see below for an example). Contrary to Theorem 1.5.1, the choice of the penalty is explicit, and does not depend on any unknown parameter beside the normalizing constant  $A > 1$ . This makes the choice of this constant much more robust, and Baraud et al. (2009) advise to take  $A \approx 1.1$ .

#### 1.5.4.2 Derivation in the Coordinate Sparse Setting

We go back to the coordinate-sparse setting of Definition 1.5.3, and derive the penalty in this framework, as we did in Section 1.5.2.2 for the Birgé-Massart criterion. Suppose that we look at models of dimensions no greater than  $p \leq n - 7$ . Then:

$$\Omega' = \sum_{\eta \in \mathcal{M}} (D_\eta + 1)e^{-L_\eta} = \sum_{D=0}^p \binom{n}{D} (D+1)e^{-L_D} = \sum_{D=0}^p \frac{1}{D+1} \leq 1 + \log(p) \leq 1 + \log(n)$$

where we took weights  $L_D = \log \binom{n}{D} + 2 \log(D+1)$ . We get the following control of the weights, using the classical bounds  $\log \binom{n}{D} \leq D \log(n)$  and  $\log(1+D) \leq D$ :

$$L_D \leq D \log(n) + 2D \leq p(2 + \log(n)).$$

Finally, taking

$$p \leq \min \left( \frac{\kappa n}{2 + \log(n)}, n - 7 \right),$$

we get the following oracle inequality, for some constant  $C'(A, \kappa)$ :

$$\mathbb{E} \left[ \frac{\|\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}_{\hat{\eta}}\|^2}{\sigma^2} \right] \leq C'(A, \kappa) \inf_{\eta \in \mathcal{M}} \left\{ \frac{\|\boldsymbol{\mu} - \boldsymbol{\mu}_\eta\|^2}{\sigma^2} + (D_\eta + 1) \log(n) \right\}$$

Remark that we miss the oracle estimator up to a  $\log(n)$  term, that is known to be un-avoidable in this setting ([Baraud et al., 2009](#); [Donoho & Johnstone, 1994](#)).

## Appendix

### 1.A The Ornstein-Uhlenbeck Process

#### 1.A.1 Stochastic Differential Equation and General Solution

Let  $\mathbf{W}(t)$  be the vector of the  $p$  traits values on one lineage,  $\mathbf{A}$  a squared  $p \times p$  matrix of strength of selections,  $\Sigma$  a diffusion matrix, with  $\mathbf{R} = \Sigma \Sigma^T$  the rate matrix,  $\boldsymbol{\beta}(t)$  a vector of optimal values, and  $\mathbf{B}_t$  the standard multi-variate Brownian motion. Then the OU process is defined by the following Stochastic Differential Equation (SDE):

$$d\mathbf{W}(t) = -\mathbf{A}(\mathbf{W}(t) - \boldsymbol{\beta}(t))dt + \Sigma d\mathbf{B}_t$$

The solution of which is given by:

$$\mathbf{W}(t) = e^{-\mathbf{A}t}\mathbf{W}(0) + \int_0^t e^{-\mathbf{A}(t-v)}\mathbf{A}\boldsymbol{\beta}(v)dv + \int_0^t e^{-\mathbf{A}(t-v)}\Sigma d\mathbf{B}_v \quad (1.27)$$

Writing this between one node  $j$  and its parent, we get:

$$\mathbf{W}(t_j) = e^{-\mathbf{A}t_j}\mathbf{W}(t_{\text{pa}(j)}) + \int_{t_{\text{pa}(j)}}^{t_j} e^{-\mathbf{A}(t_j-v)}\mathbf{A}\boldsymbol{\beta}(v)dv + \int_{t_{\text{pa}(j)}}^{t_j} e^{-\mathbf{A}(t_j-v)}\Sigma d\mathbf{B}_v \quad (1.28)$$

For a univariate OU with a constant central parameter  $\beta$ , we can get an explicit solution, that can be expressed as a Brownian Motion re-scaled by an exponential transformation of the time, as shown in the next lemma.

**Lemma 1.A.1** (Brownian Solution for the OU). *The stochastic process defined by:*

$$X_t = X_0 e^{-\alpha t} + \beta(1 - e^{-\alpha t}) + \frac{\sigma}{\sqrt{2\alpha}} e^{-\alpha t} B_{e^{2\alpha t} - 1}$$

is an OU, solution of the EDS  $dX_t = \alpha(\beta - X_t) + \sigma dB_t$ .

#### 1.A.2 Induced Variance Structure

We first need to compute the variance covariance matrix of the observations. Let  $i$  and  $j$  be two nodes. Then, from equation (1.27), we get:

$$\begin{aligned} \text{Cov}[\mathbf{X}_i; \mathbf{X}_j] &= \text{Cov}\left[e^{-\mathbf{A}t_i}\mathbf{W}(0); e^{-\mathbf{A}t_j}\mathbf{W}(0)\right] + \text{Cov}\left[\int_0^{t_i} e^{-\mathbf{A}(t_i-v)}\Sigma d\mathbf{B}_v; \int_0^{t_j} e^{-\mathbf{A}(t_j-v)}\Sigma d\mathbf{B}_v\right] \\ &= e^{-\mathbf{A}t_i}\text{Var}[\mathbf{W}(0)]e^{-\mathbf{A}^T t_j} + \mathbb{E}\left[\left(\int_0^{t_i} e^{-\mathbf{A}(t_i-v)}\Sigma d\mathbf{B}_v\right)\left(\int_0^{t_j} e^{-\mathbf{A}(t_j-v)}\Sigma d\mathbf{B}_v\right)^T\right] \\ &= e^{-\mathbf{A}t_i}\boldsymbol{\Gamma}e^{-\mathbf{A}^T t_j} + \int_0^{t_{ij}} e^{-\mathbf{A}(t_i-v)}\Sigma\Sigma^T e^{-\mathbf{A}^T(t_j-v)}dv \\ &= \underbrace{e^{-\mathbf{A}t_i}\boldsymbol{\Gamma}e^{-\mathbf{A}^T t_j}}_{\mathbf{V}^{OU_r}} + \underbrace{e^{-\mathbf{A}(t_i-t_{ij})}\left(\int_0^{t_{ij}} e^{-\mathbf{A}v}\Sigma\Sigma^T e^{-\mathbf{A}^T v}dv\right)e^{-\mathbf{A}^T(t_j-t_{ij})}}_{\mathbf{V}^{OU_p}} \end{aligned}$$

where  $\boldsymbol{\Gamma}$  is the variance-covariance matrix of the traits vector at the root.



### 1.A.2.1 Using an Eigendecomposition of $\mathbf{A}$

**General Expression.** We further assume that  $\mathbf{A}$  has an eigendecomposition with only real eigenvalues:  $\mathbf{A} = \mathbf{P}\mathbf{D}\mathbf{P}^{-1}$ , with  $\mathbf{P}$  invertible, and  $\mathbf{D} = \text{Diag}(\lambda_q, 1 \leq q \leq p)$ . Then, demoting by  $\odot$  the Hadamard product (coefficient-wise multiplication of matrices, see Section 1.B.2), the second term of the equation above becomes (Bartoszek et al., 2012; Clavel et al., 2015):

$$\mathbf{V}^{OUp} = \mathbf{P} \left( \left[ \frac{1}{\lambda_q + \lambda_r} e^{-\lambda_q t_i} e^{-\lambda_r t_j} (e^{(\lambda_q + \lambda_r) t_{ij}} - 1) \right]_{1 \leq q, r \leq p} \odot \mathbf{P}^{-1} \mathbf{R} \mathbf{P}^{-T} \right) \mathbf{P}^T \quad (1.29)$$

In order to get a stationary state, we also assume that  $\mathbf{A}$  has only positive eigenvalues. The stationary variance-covariance matrix is then given by (taking  $t_i$  and  $t_j$  to  $+\infty$ ):

$$\mathbf{S} = \mathbf{P} \left( \left[ \frac{1}{\lambda_q + \lambda_r} \right]_{1 \leq q, r \leq p} \odot \mathbf{P}^{-1} \mathbf{R} \mathbf{P}^{-T} \right) \mathbf{P}^T \quad (1.30)$$

and the variance matrix due to the process can be re-written as:

$$\mathbf{V}^{OUp} = e^{-\mathbf{A}(t_i - t_{ij})} \mathbf{S} e^{-\mathbf{A}^T(t_j - t_{ij})} - e^{-\mathbf{A}t_i} \mathbf{S} e^{-\mathbf{A}^T t_j} \quad (1.31)$$

and we get:

$$\text{Cov}[\mathbf{X}_i; \mathbf{X}_j] = e^{-\mathbf{A}t_i} \mathbf{\Gamma} e^{-\mathbf{A}^T t_j} - e^{-\mathbf{A}t_i} \mathbf{S} e^{-\mathbf{A}^T t_j} + e^{-\mathbf{A}(t_i - t_{ij})} \mathbf{S} e^{-\mathbf{A}^T(t_j - t_{ij})}$$

**Root in Stationary State.** If  $\mathbf{S} = \mathbf{\Gamma}$ , then the expression simplifies as:

$$\text{Cov}[\mathbf{X}_i; \mathbf{X}_j] = \mathbf{P} \left( \left[ \frac{1}{\lambda_q + \lambda_r} e^{-\lambda_q(t_i - t_{ij})} e^{-\lambda_r(t_j - t_{ij})} \right]_{1 \leq q, r \leq p} \odot \mathbf{P}^{-1} \mathbf{\Sigma} \mathbf{\Sigma}^T \mathbf{P}^{-T} \right) \mathbf{P}^T$$

or, in a matricial form:

$$\text{Cov}[\mathbf{X}_i; \mathbf{X}_j] = e^{-\mathbf{A}(t_i - t_{ij})} \mathbf{\Gamma} e^{-\mathbf{A}^T(t_j - t_{ij})}$$

**Scalar Case.** If  $\mathbf{A}$  is scalar, i.e.  $\mathbf{A} = \alpha \mathbf{I}$ , then the stationary variance is equal to:

$$\mathbf{S} = \frac{1}{2\alpha} \mathbf{R}$$

and the expressions above simplify to:

$$\begin{aligned} \text{Cov}[\mathbf{X}_i; \mathbf{X}_j] &= e^{-\alpha(t_i + t_j)} \mathbf{\Gamma} + e^{-\alpha(t_i + t_j)} (e^{2\alpha t_{ij}} - 1) \mathbf{S} && \text{general case} \\ &= e^{-\alpha(t_i + t_j - 2t_{ij})} \mathbf{\Gamma} && \text{stationary root} \end{aligned}$$

And we recover the known expressions of the univariate case (see e.g. Hansen, 1997; Ho & Ané, 2013b) by taking  $\mathbf{R} = \sigma^2$ . These expression will be used in the next two chapters.

### 1.A.2.2 Using the Kronecker Sum

Using the same method as in Meucci (2009), we can derive the formulas above without using the diagonalization of matrix  $\mathbf{A}$ . To do that, we need the following result:

**Lemma 1.A.2.** *For any two matrices  $\mathbf{M}$ ,  $\mathbf{N}$ , with  $\mathbf{M}$  invertible, we have:*

$$\begin{aligned} \text{vec}\left(\int_0^t e^{-\mathbf{M}v} \mathbf{N} e^{-\mathbf{M}^T v} dv\right) &= \int_0^t \text{vec}\left(e^{-\mathbf{M}v} \mathbf{N} e^{-\mathbf{M}^T v}\right) dv \\ &= \int_0^t e^{-\mathbf{M}v} \otimes e^{-\mathbf{M}^T v} \text{vec}(\mathbf{N}) dv \\ &= \int_0^t e^{-(\mathbf{M} \oplus \mathbf{M})v} \text{vec}(\mathbf{N}) dv \\ &= (\mathbf{M} \oplus \mathbf{M})^{-1} \left(\mathbf{I} - e^{-(\mathbf{M} \oplus \mathbf{M})t}\right) \text{vec}(\mathbf{N}) \end{aligned}$$

(using the first formula of Proposition 1.B.2, and the last of Proposition 1.B.1).

This lemma can be used to derive the following expression for  $\text{vec}(\mathbf{V}^{OU_p})$ :

$$\begin{aligned} \text{vec}(\mathbf{V}^{OU_p}) &= \text{vec}\left(e^{-\mathbf{A}(t_i - t_{ij})} \left(\int_0^{t_{ij}} e^{-\mathbf{A}v} \Sigma \Sigma^T e^{-\mathbf{A}^T v} dv\right) e^{-\mathbf{A}^T (t_j - t_{ij})}\right) \\ &= e^{-\mathbf{A}(t_i - t_{ij}) \oplus \mathbf{A}(t_j - t_{ij})} \text{vec}\left(\int_0^{t_{ij}} e^{-\mathbf{A}v} \Sigma \Sigma^T e^{-\mathbf{A}^T v} dv\right) \\ &= e^{-\mathbf{A}t_i \oplus \mathbf{A}t_j} e^{(\mathbf{A} \oplus \mathbf{A})t_{ij}} (\mathbf{A} \oplus \mathbf{A})^{-1} (\mathbf{I}_{p^2} - e^{-(\mathbf{A} \oplus \mathbf{A})t_{ij}}) \text{vec}(\mathbf{R}) \\ &= e^{-\mathbf{A}t_i \oplus \mathbf{A}t_j} e^{(\mathbf{A} \oplus \mathbf{A})t_{ij}} (\mathbf{I}_{p^2} - e^{-(\mathbf{A} \oplus \mathbf{A})t_{ij}}) (\mathbf{A} \oplus \mathbf{A})^{-1} \text{vec}(\mathbf{R}) \end{aligned}$$

(as  $(\mathbf{A} \oplus \mathbf{A})$  commutes with  $e^{-(\mathbf{A} \oplus \mathbf{A})t_{ij}}$ ). Sending as previously  $t_i$  and  $t_j$  to  $+\infty$ , we get the stationary variance:

$$\text{vec}(\mathbf{S}) = (\mathbf{A} \oplus \mathbf{A})^{-1} \text{vec}(\mathbf{R}) \quad (1.32)$$

and we get:

$$\begin{aligned} \text{vec}(\mathbf{V}^{OU_p}) &= e^{-\mathbf{A}t_i \oplus \mathbf{A}t_j} e^{(\mathbf{A} \oplus \mathbf{A})t_{ij}} \left[\text{vec}(\mathbf{S}) - e^{-(\mathbf{A} \oplus \mathbf{A})t_{ij}} \text{vec}(\mathbf{S})\right] \\ &= e^{-\mathbf{A}(t_i - t_{ij}) \oplus \mathbf{A}(t_j - t_{ij})} \text{vec}(\mathbf{S}) - e^{-\mathbf{A}t_i \oplus \mathbf{A}t_j} \text{vec}(\mathbf{S}) \end{aligned}$$

which simplifies to the same formula as above:

$$\text{vec}(\mathbf{V}^{OU_p}) = \text{vec}\left(e^{-\mathbf{A}(t_i - t_{ij})} \mathbf{S} e^{-\mathbf{A}^T (t_j - t_{ij})} - e^{-\mathbf{A}t_i} \mathbf{S} e^{-\mathbf{A}^T t_j}\right)$$

which lead to the same expressions for the variance.

### 1.A.3 Incomplete Data Formulation

Using equation (1.28), we get, for  $j \in \llbracket 2, m+n \rrbracket$ :

$$\mathbf{X}_j \mid \mathbf{X}_{\text{pa}(j)} \sim \mathcal{N}\left(e^{-\mathbf{A}\ell_j}\mathbf{X}_{\text{pa}(j)} + (\mathbf{I}_p - e^{-\mathbf{A}\ell_j})\boldsymbol{\beta}_j, \boldsymbol{\Upsilon}_i = \int_0^{\ell_i} e^{-\mathbf{A}u}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T e^{-\mathbf{A}^T u} du\right)$$

with, using lemma 1.A.2 and the commutation:

$$\text{vec}(\boldsymbol{\Upsilon}_j) = (\mathbf{A} \oplus \mathbf{A})^{-1} (\mathbf{I} - e^{-(\mathbf{A} \oplus \mathbf{A})t}) \text{vec}(\mathbf{R}) = (\mathbf{I} - e^{-(\mathbf{A} \oplus \mathbf{A})t}) \text{vec}(\mathbf{S})$$

Hence:

$$\boldsymbol{\Upsilon}_i = \mathbf{S} - e^{-\mathbf{A}\ell_j}\mathbf{S}e^{-\mathbf{A}^T\ell_j}$$

And, if  $\mathbf{A}$  is diagonalizable in  $\mathbb{R}$ :

$$\boldsymbol{\Upsilon}_i = \mathbf{P} \left( \left[ \frac{1}{\lambda_q + \lambda_r} (1 - e^{-(\lambda_q + \lambda_r)\ell_i}) \right]_{1 \leq q, r \leq p} \odot \mathbf{P}^{-1}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^T\mathbf{P}^{-T} \right) \mathbf{P}^T$$

*Remark 1.A.1.* If  $\mathbf{A}$  is scalar, the formula above simplifies to:

$$\mathbf{X}_j \mid \mathbf{X}_{\text{pa}(j)} \sim \mathcal{N}\left(e^{-\alpha\ell_j}\mathbf{X}_{\text{pa}(j)} + (\mathbf{I}_p - e^{-\alpha\ell_j})\boldsymbol{\beta}_j, (1 - e^{-2\alpha\ell_j})\frac{1}{2\alpha}\mathbf{R}\right)$$

## 1.B Multivariate Analysis Tools

In this section, we recall some classical mathematical tools used in multivariate analysis. Those will be useful mainly in Chapter 3. For an extensive view of these tools, we refer to [Mardia et al. \(1979\)](#).

### 1.B.1 Kronecker Product and Vectorization

The Kronecker product and the vectorization operation appear naturally in multivariate analysis, to describe the distribution of a matrix for instance.

#### 1.B.1.1 Kronecker Product

**Definition 1.B.1.** Let  $\mathbf{A}$  and  $\mathbf{B}$  be two matrices of size  $m \times n$  and  $p \times q$ . Their Kronecker product  $\mathbf{A} \otimes \mathbf{B}$  is the  $mp \times nq$  matrix defined by:

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} A_{11}\mathbf{B} & \dots & A_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ A_{m1}\mathbf{B} & \dots & A_{mn}\mathbf{B} \end{pmatrix}$$

**Proposition 1.B.1.** Let  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  be matrices of size  $m \times n$ ,  $p \times q$  and  $k \times l$  respectively. The following properties hold:

- $\otimes$  is distributive over  $+$ , associative, and is not commutative.
- $(\mathbf{A} \otimes \mathbf{B})(\mathbf{C} \otimes \mathbf{D}) = (\mathbf{AC}) \otimes (\mathbf{BD})$  (when it makes sense)
- $(\mathbf{A} \otimes \mathbf{B})^T = \mathbf{A}^T \otimes \mathbf{B}^T$

- $\text{Rank}(\mathbf{A} \otimes \mathbf{B}) = \text{Rank}(\mathbf{A})\text{Rank}(\mathbf{B})$
  - If  $\mathbf{A}$  and  $\mathbf{B}$  are rectangular with singular values  $(\lambda_1, \dots, \lambda_{r_A})$  and  $(\mu_1, \dots, \mu_{r_B})$ , then  $\mathbf{A} \otimes \mathbf{B}$  has singular values  $(\lambda_j \mu_l)_{1 \leq j \leq r_A; 1 \leq l \leq r_B}$ .
- In addition, for squared matrices  $\mathbf{A}$  and  $\mathbf{B}$  sizes  $m \times m$  and  $p \times p$ :
- $(\mathbf{A} \otimes \mathbf{B})^{-1} = \mathbf{A}^{-1} \otimes \mathbf{B}^{-1}$  (if  $\mathbf{A}$  and  $\mathbf{B}$  are non-singular)
  - $|\mathbf{A} \otimes \mathbf{B}| = |\mathbf{A}|^p |\mathbf{B}|^m$
  - $\text{tr}(\mathbf{A} \otimes \mathbf{B}) = \text{tr}(\mathbf{A})\text{tr}(\mathbf{B})$
  - If  $\mathbf{A}$  and  $\mathbf{B}$  are squared with eigenvalues  $(\lambda_1, \dots, \lambda_m)$  and  $(\mu_1, \dots, \mu_p)$ , then  $\mathbf{A} \otimes \mathbf{B}$  has eigenvalues  $(\lambda_j \mu_l)_{1 \leq j \leq m; 1 \leq l \leq p}$ .
  - $e^{\mathbf{A}} \otimes e^{\mathbf{B}} = e^{\mathbf{A} \oplus \mathbf{B}}$  where  $\mathbf{A} \oplus \mathbf{B} = \mathbf{A} \otimes \mathbf{I}_m + \mathbf{I}_p \otimes \mathbf{B}$

### 1.B.1.2 Vectorization

**Definition 1.B.2.** The vectorized vector of an  $m \times n$  matrix  $\mathbf{A}$  is obtained by staking its columns together:

$$\text{vec}(\mathbf{A}) = (A_{11}, \dots, A_{m1}, A_{12}, \dots, A_{m2}, \dots, A_{1n}, \dots, A_{mn})^T$$

The vectorization operation and the Kronecker product work particularly well together, as shown by the following proposition.

**Proposition 1.B.2** (Vectorization and Kronecker product). *Let  $\mathbf{A}$ ,  $\mathbf{B}$  and  $\mathbf{C}$  be matrices of size  $m \times n$ ,  $n \times p$  and  $p \times q$  respectively. Then the following identities hold:*

$$\begin{aligned} \text{vec}(\mathbf{ABC}) &= (\mathbf{C}^T \otimes \mathbf{A}) \text{vec}(\mathbf{B}) \\ &= (\mathbf{I}_q \otimes \mathbf{AB}) \text{vec}(\mathbf{C}) \\ &= (\mathbf{C}^T \mathbf{B}^T \otimes \mathbf{I}_m) \text{vec}(\mathbf{A}). \end{aligned}$$

In particular:

$$\begin{aligned} \text{vec}(\mathbf{AB}) &= (\mathbf{I}_p \otimes \mathbf{A}) \text{vec}(\mathbf{B}) \\ &= (\mathbf{B}^T \otimes \mathbf{I}_m) \text{vec}(\mathbf{A}). \end{aligned}$$

Vectorization can also be used in cooperation with Kronecker products to write Mahalanobis norms. These quantities naturally appear in the density of matrix normal distributions.

**Proposition 1.B.3** (Vectorization and Mahalanobis Norm). *Let  $\mathbf{A}$  and  $\mathbf{B}$  be two symmetric matrices of size  $n \times n$  and  $m \times m$  respectively. Let  $\mathbf{X}$  be a matrix of size  $m \times n$ . Then:*

$$\|\text{vec}(\mathbf{X})\|_{(\mathbf{A} \otimes \mathbf{B})^{-1}}^2 = \|\text{vec}(\mathbf{X}^T)\|_{(\mathbf{B} \otimes \mathbf{A})^{-1}}^2.$$

*Proof.*

$$\begin{aligned} \|\text{vec}(\mathbf{X})\|_{(\mathbf{A} \otimes \mathbf{B})^{-1}}^2 &= \text{vec}(\mathbf{X})^T (\mathbf{A}^{-1/2} \otimes \mathbf{B}^{-1/2})^T (\mathbf{A}^{-1/2} \otimes \mathbf{B}^{-1/2}) \text{vec}(\mathbf{X}) \\ &= \|\text{vec}((\mathbf{B}^{-1/2}) \mathbf{X} (\mathbf{A}^{-1/2})^T)\|^2 \\ &= \|\text{vec}((\mathbf{A}^{-1/2}) \mathbf{X}^T (\mathbf{B}^{-1/2})^T)\|^2 \\ &= \|\text{vec}(\mathbf{X}^T)\|_{(\mathbf{B} \otimes \mathbf{A})^{-1}}^2 \end{aligned}$$

□

## 1.B.2 Hadamard Product

The Hadamard product enjoys less properties than the Kronecker product, but is sometimes useful to write compact expressions.

**Definition 1.B.3.** Let  $\mathbf{A}$  and  $\mathbf{B}$  be matrices of size  $m \times n$ . Their Hadamard product  $\mathbf{A} \odot \mathbf{B}$  is the  $m \times n$  matrix defined by:

$$(\mathbf{A} \odot \mathbf{B})_{ij} = \mathbf{A}_{ij} \mathbf{B}_{ij}$$

**Proposition 1.B.4.** *The following properties hold:*

- *Distributive over +, associative, commutative.*
- *If  $\mathbf{X}$  and  $\mathbf{Y}$  are vectors, and  $\mathbf{D}_\mathbf{X} = \text{Diag}(\mathbf{X})$ ,  $\mathbf{D}_\mathbf{Y} = \text{Diag}(\mathbf{Y})$ , then:  $\mathbf{X}^T (\mathbf{A} \odot \mathbf{B}) \mathbf{Y} = \text{tr}(\mathbf{D}_\mathbf{X} \mathbf{A} \mathbf{D}_\mathbf{Y} \mathbf{B}^T)$*
- $\text{Rank}(\mathbf{A} \odot \mathbf{B}) \leq \text{Rank}(\mathbf{A}) \text{Rank}(\mathbf{B})$
- $|\mathbf{A} \odot \mathbf{B}| \geq |\mathbf{A}| |\mathbf{B}|$  (*Schur product theorem*)

**Proposition 1.B.5** (Vectorization and Hadamard product). *Let  $\mathbf{A}$ ,  $\mathbf{B}$  be matrices of size  $m \times n$ .*

$$\text{vec}(\mathbf{A} \odot \mathbf{B}) = \text{vec}(\mathbf{A}) \odot \text{vec}(\mathbf{B})$$

**Lemma 1.B.1** (Kronecker and Hadamard products [Kollo & Neudecker \(1993\)](#); [Mond & Pečarić \(2000\)](#)). *Let  $\mathbf{J}$  be a  $n^2 \times n$  matrix, such that  $\mathbf{J}^T = [\mathbf{E}_{11} \mathbf{E}_{22} \cdots \mathbf{E}_{nn}]$  is the matrix of concatenation of base matrices  $\mathbf{E}_{ij}$  that are matrices of zeros, with one one at entry  $(i, i)$ . We have  $\mathbf{J} \mathbf{J}^T = \mathbf{I}_{n^2}$ . Let  $\mathbf{A}$  and  $\mathbf{B}$  two  $n \times n$  matrices. Then:*

$$\mathbf{A} \odot \mathbf{B} = \mathbf{J}^T (\mathbf{A} \otimes \mathbf{B}) \mathbf{J}.$$

# Chapter 2

## Shift Detection for Univariate Processes

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>78</b>
2.1.1	Motivations: Environmental Shifts	78
2.1.2	Stochastic Process on a tree	78
2.1.3	Scope of this article	80
<b>2.2</b>	<b>Statistical Modeling</b>	<b>81</b>
2.2.1	Probabilistic Model	81
2.2.2	Incomplete Data Model Point of View	83
2.2.3	Linear Regression Model Point of View	83
<b>2.3</b>	<b>Identifiability and Complexity of a Model</b>	<b>85</b>
2.3.1	Identifiability Issues	85
2.3.2	Complexity of a Collection of Models	89
2.3.3	Another Characterization of Parsimony	92
<b>2.4</b>	<b>Statistical Inference</b>	<b>92</b>
2.4.1	Expectation Maximization	92
2.4.2	Model Selection	93
<b>2.5</b>	<b>Simulations Studies</b>	<b>96</b>
2.5.1	Simulations Scheme	96
2.5.2	Inference Procedures	96
2.5.3	Scores Used	97
2.5.4	Results	97
<b>2.6</b>	<b>Case Study: Chelonian Carapace Length Evolution</b>	<b>100</b>
2.6.1	Description of the Dataset	100
2.6.2	Method	101
2.6.3	Results	101
2.6.4	Comparison with other methods	102
<b>Appendices</b>		<b>104</b>
<b>2.A</b>	<b>Enumeration of Equivalence Classes</b>	<b>104</b>
<b>2.B</b>	<b>A Vandermonde Like Identity</b>	<b>105</b>
<b>2.C</b>	<b>Technical Details of the EM</b>	<b>108</b>
2.C.1	E Step	108
2.C.2	Complete Likelihood Computation	108
2.C.3	M step	109
2.C.4	Initialization	111
<b>2.D</b>	<b>Optimal Shift Location with Fixed Root</b>	<b>111</b>
<b>2.E</b>	<b>Proof of Proposition 2.4.1 for Model Selection</b>	<b>112</b>
<b>2.F</b>	<b>Supplementary Figures</b>	<b>113</b>
2.F.1	Simulation Study: Sensitivity and False Positive Rate	113
2.F.2	Simulation Study: Complementary Analysis	114
2.F.3	Chelonia Dataset: Comparison of Inferred Shift Locations	114
<b>2.G</b>	<b>Practical Implementation</b>	<b>117</b>

---

## Foreword

This chapter has been published under the title *Detection of adaptive shifts on phylogenies by using shifted stochastic processes on a tree* in the *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. [Bastide et al. \(2017b\)](#).

**Abstract.** Comparative and evolutive ecologists are interested in the distribution of quantitative traits among related species. The classical framework for these distributions consists of a random process running along the branches of a phylogenetic tree relating the species. We consider shifts in the process parameters, which reveal fast adaptation to changes of ecological niches. We show that models with shifts are not identifiable in general. Constraining the models to be parsimonious in the number of shifts partially alleviates the problem but several evolutionary scenarios can still provide the same joint distribution for the extant species. We provide a recursive algorithm to enumerate all the equivalent scenarios and to count the number of effectively different scenarios. We introduce an incomplete-data framework and develop a maximum likelihood estimation procedure based on the EM algorithm. Finally, we propose a model selection procedure, based on the cardinal of effective scenarios, to estimate the number of shifts and for which we prove an oracle inequality.

## 2.1 Introduction

### 2.1.1 Motivations: Environmental Shifts

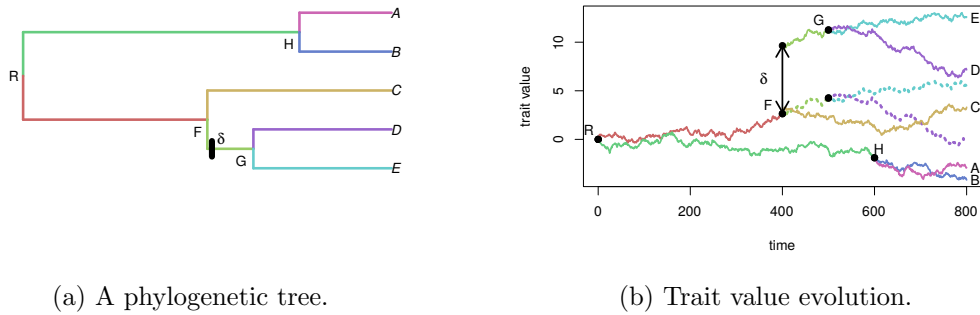
An important goal of comparative and evolutionary biology is to decipher the past evolutionary mechanisms that shaped present day diversity, and more specifically to detect the dramatic changes that occurred in the past (see for instance [Losos, 1990](#); [Mahler et al., 2013](#); [Davis et al., 2007](#); [Jaffe et al., 2011](#)). It is well established that related organisms do not evolve independently ([Felsenstein, 1985](#)): their shared evolutionary history is well represented by a phylogenetic tree. In order to explain the pattern of traits measured on a set of related species, one needs to take these correlations into account. Indeed, a given species will be more likely to have a similar trait value to her “sister” (a closely related species) than to her “cousin” (a distantly related species), just because of the structure of the tree. On top of that structure, when considering a *functional* trait (i.e. a trait directly linked to the fitness of its bearer), such as shell size for turtles ([Jaffe et al., 2011](#)), one needs to take into account the effect of the species environment on its traits. Indeed, a change in the environment for a subset of species, like a move to the Galàpagos Islands for turtles, will affect the observed trait distribution, here with a shift towards giant shell sizes compared to mainland turtles. The observed present-day trait distribution hence contains the footprint of adaptive events, and should allow us to detect unobserved past events, like the migration of one ancestral species to a new environment. Our goal here is to devise a statistical method based on a rigorous maximum likelihood framework to automatically detect the past environmental shifts that shaped the present day trait distribution.

### 2.1.2 Stochastic Process on a tree

We model the evolution of a quantitative adaptive trait using the framework of stochastic processes on a tree. Specifically, given a rooted phylogenetic tree, we assume that the

trait evolves according to a given stochastic process on each branch of the tree. At each speciation event, or equivalently node of the tree, one independent copy with the same initial conditions and parameters is created for each daughter species, or outgoing branches.

**Tree Structure.** This model is our null model: it accounts for the tree-induced distribution of trait values in the absence of shifts. Depending on the phenomenon studied, several stochastic processes can be used to capture the dynamic of the trait evolution. In the following, we will use the Brownian Motion (BM) and the Ornstein-Uhlenbeck (OU) processes.



(a) A phylogenetic tree.

(b) Trait value evolution.

Figure 2.1.1 – Trait evolution under a Brownian Motion. The ancestral value of the trait is 0, and the observed values (time 800) range from  $-4$  to 11 for extant species. One shift occurs on the parent branch of (D,E), changing the trajectory of their ancestral trait value from the grey one to the colored one. The shift increases the observed dispersion.

**Brownian Motion.** Since the seminal article of [Felsenstein \(1985\)](#), the BM has been used as a neutral model of trait evolution. If  $(B_t; t \geq 0)$  is the Brownian motion, a character  $(W_t; t \geq 0)$  evolves on a lineage according to the stochastic differential equation:  $dW_t = \sigma dB_t$ ,  $\sigma^2$  being a variance parameter. If  $\mu$  is the ancestral value at the root of the tree ( $t = 0$ ), then  $W_t \sim \mathcal{N}(\mu, \sigma^2 t)$ . The variance  $\sigma^2 t$  of the trait is proportional to the time of evolution and the covariance  $\sigma^2 t_{ij}$  between two species  $i$  and  $j$  is proportional to their time of shared evolution.

**Ornstein-Uhlenbeck Process.** An unbounded variance is quite unrealistic for adaptive traits ([Butler & King, 2004](#)). For that reason, the OU process, that models stabilizing selection around an adaptive optimum ([Hansen, 1997](#)) is usually preferred to the BM. It is defined by the stochastic differential equation  $dW_t = -\alpha(W_t - \beta)dt + \sigma dB_t$ , and has stationary distribution  $\mathcal{N}(\beta, \sigma^2/2\alpha)$ . In this equation,  $W_t$  is the *secondary optimum* of a species, a trade-off between all selective constraints – e.g. ecological – on the trait and can be approached by the population mean of that species. The term  $-\alpha(W_t - \beta)dt$  of the equation represents the effects of stabilizing selection towards a *primary optimum*  $\beta$ , that depends only on the ecological niche of the species. The selection strength is controlled by the call-back parameter  $\alpha$ . For interpretation purpose, we will use the *phylogenetic half-life*  $t_{1/2} = \ln(2)/\alpha$ , defined as the time needed for the expected trait value to move half the distance from the ancestral state to the primary optimum ([Hansen, 1997](#)). The



term  $\sigma dB_t$  represents the random effects of uncontrolled factors, ranging from genetic drift to environmental fluctuations. We refer to Hansen (1997); Hansen et al. (2008) for further discussion and deeper biological interpretations of the hypothesis underlying this model of evolution. The aim of our work is to detect environmental shifts.

**Environmental Shifts.** In addition to the previous mechanisms, we assume that abrupt environmental changes affected the ecological niche of some ancestral species. We model these changes as instantaneous shifts in the parameters of the stochastic process. Shifted parameters are inherited along time and thus naturally create clusters of extant species that share the same parameters trajectories. In the BM process, shifts affect the mean value of the trait and are thus instantaneously passed on to the trait itself (see Figure 2.1.1) whereas in the OU process, shifts affect the primary optimum  $\beta$ . In this case, the trait converges to its new stationary value with an exponential decay of half-life  $t_{1/2}$  inducing a lag that makes recent shifts harder to detect (Hansen & Bartoszek, 2012). In the remainder, we assume that all other parameters ( $\sigma^2$  for the BM and  $\sigma^2, \alpha$  for the OU) are fixed and constant (but see Beaulieu et al., 2012; Rabosky, 2014, for partial relaxations of this hypothesis).

### 2.1.3 Scope of this article

**State of the Art.** Phylogenetics Comparative Methods (PCM) are an active field that has seen many fruitful developments in the last few years (see Pennell & Harmon, 2013, for an extensive review). Several methods have been specifically developed to study adaptive evolution, starting with the work of Butler & King (2004). Butler & King (2004) only consider shifts in the optimal value  $\beta$  whereas Beaulieu et al. (2012) also allow for shifts in the selection strength  $\alpha$  and the variance  $\sigma^2$ . Both have in common that shift locations are assumed to be known. Several extensions of the model without or with known shifts have also been proposed: Hansen et al. (2008) extended the original work of Hansen (1997) on OU processes to a two-tiered model where  $\beta(t)$  is itself a stochastic process (either BM or OU). Bartoszek et al. (2012) extended it further to multivariate traits whereas Hansen & Bartoszek (2012) introduced errors in the observations. Expanding upon the BM, (Landis et al., 2013) replaced fixed shifts, known or unknown, by random jump processes using Levy processes. Non-Gaussian models of trait evolution were also recently considered by Hiscott et al. (2016), who adapted Felsenstein’s pruning algorithm for the likelihood computation of these models, using efficient integration techniques. Finally, Ho & Ané (2013b) derived consistency results for estimation of the parameters of an OU on a tree and Bartoszek & Sagitov (2015); Sagitov & Bartoszek (2012) computed confidence intervals of the same parameters by assuming an unknown random tree topology and averaging over it.

The first steps toward automatic detection of shifts, which is the problem of interest in this paper, have been done in a Bayesian framework, for both the BM (Eastman et al., 2013) and the OU (Uyeda & Harmon, 2014). Using RJ-MCMC, they provide the user with the posterior distribution of the number and location of shifts on the tree. Convergence is however severely hampered by the size of the search space. The growing use of PCM in fields where large trees are the norm makes maximum likelihood based point estimates of the shift locations more practical. A stepwise selection procedure for the shifts has been proposed in Ingram & Mahler (2013). The procedure adds shifts one at the time and is therefore rather efficient but the selection criterion is heuristic and

has no theoretical grounding for that problem, where observations are correlated through the tree structure. These limitations have been pointed out in [Ho & Ané \(2014\)](#). In this article, the authors describe several identifiability problems that arise when trying to infer the position of the shifts on a tree, and propose a different stepwise algorithm based on a more stringent selection criterion, heuristically inspired by segmentation algorithms.

To rigorously tackle this issue, we introduced a framework where a univariate trait evolves according to an OU process with stationary root state (S) on an Ultrametric tree (U). Furthermore, as the exact position of a shift on a branch is not identifiable for an ultrametric tree, we assume that shifts are concomitant to speciation events and only occur at Nodes (N) of the tree. We refer to this model as OUsun hereafter.

**Our Contribution.** In this work, we make several major contributions to the problem at hand. First, we derive a statistical method to find a maximum likelihood estimate of the parameters of the model. When the number of shifts is fixed, we work out an Expectation Maximization (EM) algorithm that takes advantage of the tree structure of the data to efficiently maximize the likelihood. Second, we show that, given the model used and the kind of data available, some evolutionary scenarios remain indistinguishable. Formally, we exhibit some identifiability problems in the location of the shifts, even when their number is fixed, and subsequently give a precise characterization of the space of models that can be inferred from the data on extant species. Third, we provide a rigorous model selection criterion to choose the number of shifts needed to best explain the data. Thanks to our knowledge of the structure of the spaces of models, acquired through our identifiability study, we are able to mathematically derive a penalization term, together with an oracle inequality on the estimator found. Fourth and finally, we implement the method on the statistical software R ([R Core Team, 2017](#)), and show that it correctly recovers the structure of the model on simulated datasets. When applied to a biological example, it gives results that are easily interpretable, and coherent with previously developed methods. All the code used in this article is publicly available on GitHub (<https://github.com/pbastide/PhylogeneticEM>).

**Outline.** In Section 2.2, we present the model, using two different mathematical point of views, that are both useful in different aspects of the inference. In Section 2.3, we tackle the identifiability problems associated with this model, and describe efficient algorithms to enumerate, first, all equivalent models within a class, and, second, the number of truly different models for a given number of shifts. These two sections form the foundation of Section 2.4, in which we describe our fully integrated maximum likelihood inference procedure. Finally, in Sections 2.5 and 2.6, we conduct some numerical experiments on simulated and biological datasets.

## 2.2 Statistical Modeling

### 2.2.1 Probabilistic Model

**Tree Parametrization.** As shown in Figure 2.2.1, we consider a rooted tree  $\mathcal{T}$  with  $n$  tips and  $m$  internal nodes ( $m = n - 1$  for binary trees). The internal nodes are numbered from 1 (the root) to  $m$ , and the tips from  $m+1$  to  $m+n$ . Let  $i$  be an integer,  $i \in \llbracket 2, m+n \rrbracket$ . Then  $\text{pa}(i)$  is the unique parent of node  $i$ . The branch leading to  $i$  from  $\text{pa}(i)$  is noted

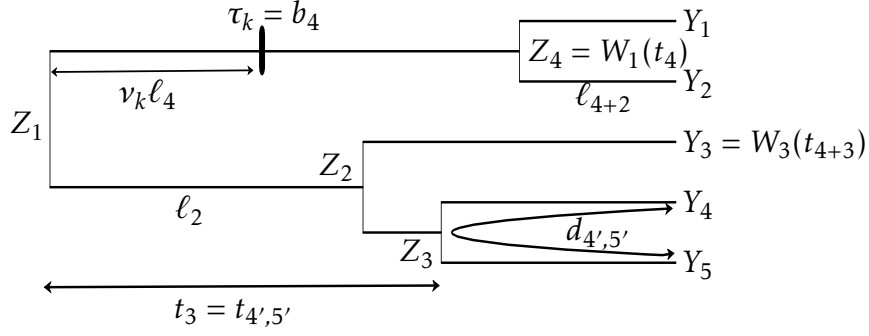


Figure 2.2.1 – A rooted and time calibrated phylogenetic tree with the notations used to parametrize the tree ( $l, t, d, b$ ) and the observed ( $Y$ ) and non-observed ( $Z$ ) variables.

$b_i$  and has length  $\ell_i = t_i - t_{\text{pa}(i)}$  where  $t_i$  is the time elapsed between the root and node  $i$ . By convention, we set  $t_1 = 0$  and  $t_{\text{pa}(1)} = -\infty$  for the root. The last convention ensures that the trait follows the stationary distribution (if any) of the process at the root. We denote  $\text{anc}(i) = \{\text{pa}^r(i) : r \geq 0\}$  the set composed of node  $i$  and of all its ancestors up to the root. For a couple of integers  $(i, j)$ ,  $(i, j) \in \llbracket 1, m+n \rrbracket^2$ , nodes  $i$  and  $j$  are at phylogenetic distance  $d_{ij}$  and the time of their most recent common ancestor (mrca) is  $t_{ij}$ . We consider ultrametric trees, for which  $t_{m+1} = \dots = t_{m+n} =: h$  and note  $h$  the tree height. In the following, the tree is fixed and assumed to be known.

**Trait Values.** We denote by  $\mathbf{X}$  the vector of size  $m+n$  of the trait values at the nodes of the tree. We split this vector between non-observed values  $\mathbf{Z}$  (size  $m$ ) at the internal nodes, and observed values  $\mathbf{Y}$  (size  $n$ ) at the tips, so that  $\mathbf{X}^T = (\mathbf{Z}^T, \mathbf{Y}^T)$ . According to our model of trait evolution, the random variable  $X_i$ ,  $i \in \llbracket 1, m+n \rrbracket$ , is the result of a stochastic process stopped at time  $t_i$ . In the following, we assume that the inference in the BM case is done conditionally to a fixed root value  $X_1 = \mu$ . In the OUsun case, we assume that the root trait value is randomly drawn from the stationary distribution:  $X_1 \sim \mathcal{N}(\mu = \beta_1, \gamma^2 = \frac{\sigma^2}{2\alpha})$ , where  $\beta_1$  is the ancestral optimal value.

**Shifts.** We assume that  $K$  shifts occur on the tree,  $K \in \mathbb{N}$ . The  $k^{\text{th}}$  shift,  $k \in \llbracket 1, K \rrbracket$ , occurs at the beginning of branch  $\tau_k$ ,  $\tau_k \in \{b_i, i \in \llbracket 2, m+n \rrbracket\}$ , and has intensity  $\delta_k$ ,  $\delta_k \in \mathbb{R}$ . The interpretation of this intensity depends on the process. In the following, we use the vector  $\Delta$  of shifts on the branches, of size  $m+n$ , with  $K+1$  non-zero entries, and defined as follows (see example 2.2.1):

$$\Delta_1 = \mu \quad (= \beta_1 \text{ for an OUsun}) \quad \text{and} \quad \forall i \in \llbracket 2, n+m \rrbracket, \Delta_i = \begin{cases} \delta_k & \text{if } \tau_k = b_i \\ 0 & \text{otherwise.} \end{cases}$$

Note that no proper shift occurs on the root branch, but that the root trait value or mean,  $\mu$ , is formalized as an initial fictive shift on this fictive branch.

**Parameters.** The parameters needed to describe an OUsun (respectively, a BM) are  $\theta = (\gamma, \alpha, \Delta)$  (resp.  $\theta = (\sigma, \Delta)$ ). Note that, as  $\sigma^2 = 2\alpha\gamma^2$ , only the two parameters  $\alpha$  and  $\gamma$  are needed to describe the OUsun. We denote by OUsun( $\theta$ ) (resp. BM( $\theta$ )) the OUsun (resp. BM) process running on the tree with parameters  $\theta$ .

### 2.2.2 Incomplete Data Model Point of View

If the trait values were observed at all nodes of the tree, including ancestral ones, shifts would be characterized by unexpectedly large differences between a node and its parent. A way to mimic this favorable case is to use an incomplete data model, as described below. This representation of the model will be useful for the parametric inference using an EM algorithm (Section 2.4.1).

**Brownian Motion.** As the shifts occur directly in the mean of the process, we get:

$$X_1 = \mu \quad \text{and} \quad \forall i \in \llbracket 2, m+n \rrbracket, X_i | X_{\text{pa}(i)} \sim \mathcal{N}(X_{\text{pa}(i)} + \Delta_i, \ell_i \sigma^2) \quad (2.1)$$

The trait value at node  $i$ ,  $i \in \llbracket 2, m+n \rrbracket$ , is centered on the value of its parent node  $X_{\text{pa}(i)}$ , with a variance proportional to the evolution time  $\ell_i$  between  $i$  and  $\text{pa}(i)$ . The effect of a non-zero shift  $\Delta_i$  on branch  $b_i$  is simply to translate the trait value by  $\Delta_i$ .

**Ornstein-Uhlenbeck.** The shifts occur on the primary optimum  $\beta$ , which is piecewise constant. As the shifts are assumed to occur at nodes, the primary optimum is entirely defined by its initial value  $\beta_1$  and its values  $\beta_2, \dots, \beta_{n+m}$  on branches of the tree, where  $\beta_i$  is the value on branch  $b_i$  leading to node  $i$ .

$$\beta_1 \in \mathbb{R} (= \mu \text{ for an OUsun}) \quad \text{and} \quad \forall i \in \llbracket 2, m+n \rrbracket, \beta_i = \beta_{\text{pa}(i)} + \Delta_i \quad (2.2)$$

Assuming that the root node is in the stationary state, we get:

$$\begin{cases} X_1 \sim \mathcal{N}(\mu = \beta_1, \gamma^2 = \frac{\sigma^2}{2\alpha}) \\ X_i | X_{\text{pa}(i)} \sim \mathcal{N}(X_{\text{pa}(i)} e^{-\alpha \ell_i} + \beta_i (1 - e^{-\alpha \ell_i}), \frac{\sigma^2}{2\alpha} (1 - e^{-2\alpha \ell_i})) \end{cases} \quad \forall i \in \llbracket 2, m+n \rrbracket \quad (2.3)$$

The trait value at node  $i$  depends on both the trait value at the father node  $X_{\text{pa}(i)}$  and the value  $\beta_i$  of the primary optimum on branch  $b_i$ . Contrary to the BM case, the shifts only appear indirectly in the distributions of  $X_i$ s, through the values of  $\beta$ , and with a shrinkage of  $1 - e^{-\alpha d}$  for shifts of age  $d$ , which makes recent shifts ( $d$  small compared to  $1/\alpha$ ) harder to detect.

### 2.2.3 Linear Regression Model Point of View

A more compact and direct representation of the model is to use the tree incidence matrix to link linearly the observed values (at the tips) with the shift values, as explained below. We will use this linear regression framework for the Lasso (Tibshirani, 1996) initialization of the EM (Section 2.4.1) and the model selection procedure (Section 2.4.2). It will also help us to explore identifiability issues raised in the next section.

**Matrix of a Tree.** It follows from the recursive definition of  $\mathbf{X}$  that it is a Gaussian vector. In order to express its mean vector given the shifts, we introduce the tree squared matrix  $\mathbf{U}$ , of size  $(m+n)$ , defined by its general term:  $U_{ij} = \mathbb{I}\{j \in \text{anc}(i)\}, \forall (i, j) \in \llbracket 1, m+n \rrbracket^2$ . In other words, the  $j^{\text{th}}$  column of this matrix,  $j \in \llbracket 1, m+n \rrbracket$ , is the indicator vector of the descendants of node  $j$ . To express the mean vector of the observed values  $\mathbf{Y}$ , we also need the sub-matrix  $\mathbf{T}$ , of size  $n \times (m+n)$ , composed of the bottom  $n$  rows of matrix  $\mathbf{U}$ , corresponding to the tips (see example 2.2.1 below). Likewise, the  $i^{\text{th}}$  row of  $\mathbf{T}$ ,  $i \in \llbracket 1, n \rrbracket$ , is the indicator vector of the ancestors of leaf  $m+i$ .

**Brownian Motion.** From the tree structure, we get:

$$\mathbf{X} = \mathbf{U}\boldsymbol{\Delta} + \mathbf{E}_X \quad \text{and} \quad \mathbf{Y} = \mathbf{T}\boldsymbol{\Delta} + \mathbf{E}_Y \quad (2.4)$$

Here,  $\mathbf{E}_X \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{XX})$  is a Gaussian error vector with co-variances  $[\boldsymbol{\Sigma}_{XX}]_{ij} = \sigma^2 t_{ij}$  for any  $1 \leq i, j \leq m+n$ , and  $\mathbf{E}_Y$  is the vector made of the last  $n$  coordinates of  $\mathbf{E}_X$ .

**Ornstein-Uhlenbeck.** For the OUsun, shifts occur on the primary optimum, and there is a lag term, so that:

$$\boldsymbol{\beta} = \mathbf{U}\boldsymbol{\Delta} \quad \text{and} \quad \mathbf{X} = (\mathbf{U} - \mathbf{A}\mathbf{U}\mathbf{B})\boldsymbol{\Delta} + \mathbf{E}_X \quad (2.5)$$

where  $\mathbf{A} = \text{Diag}(e^{-\alpha t_i}, 1 \leq i \leq m+n)$  and  $\mathbf{B} = \text{Diag}(0, e^{\alpha t_{\text{pa}(i)}}, 2 \leq i \leq m+n)$  are diagonal matrices of size  $m+n$ . As previously,  $\mathbf{E}_X \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{XX})$ , but  $\boldsymbol{\Sigma}_{XX} = \gamma^2 [e^{-\alpha d_{ij}}]_{1 \leq i, j \leq m+n}$ . As the tree is ultrametric, this expression simplifies to the following one when considering only observed values:

$$\mathbf{Y} = \mathbf{T}\mathbf{W}(\alpha)\boldsymbol{\Delta} + \mathbf{E}_Y \quad (2.6)$$

where  $\mathbf{E}_Y$  is the Gaussian vector made of the last  $n$  coordinates of  $\mathbf{E}_X$ , and  $\mathbf{W}(\alpha) = \text{Diag}(1, 1 - e^{-\alpha(h-t_{\text{pa}(i)})}, 2 \leq i \leq m+n)$  is a diagonal matrix of size  $m+n$ . Note that if  $\alpha$  is positive, then  $\alpha(h-t_{\text{pa}(i)}) > 0$  for any  $i \in \llbracket 1, m+n \rrbracket$ , and  $\mathbf{W}(\alpha)$  is invertible.

*Example 2.2.1.* The tree presented in Figure 2.2.1 has five tips and one shift on branch  $4 + 3 = 7$ , so:

$$\mathbf{U} = \begin{array}{c} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \\ Y_5 \end{array} \left( \begin{array}{cccccccccc} Z_1 & Z_2 & Z_3 & Z_4 & Y_1 & Y_2 & Y_3 & Y_4 & Y_5 \\ \left( \begin{array}{cccccccccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right) \end{array} \right) \quad \text{and} \quad \boldsymbol{\Delta} = \begin{pmatrix} \mu \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \delta_1 \\ 0 \\ 0 \end{pmatrix}$$

And, respectively, for a BM or an OUsun ( $\mu = \beta_1$ ):

$$\mathbb{E}[\mathbf{Y}] = \mathbf{T}\boldsymbol{\Delta} = (\mu, \mu, \mu + \delta_1, \mu, \mu)^T \quad (\text{BM})$$

$$\mathbb{E}[\mathbf{Y}] = \mathbf{T}\mathbf{W}(\alpha)\boldsymbol{\Delta} = (\beta_1, \beta_1, \beta_1 + \delta_1(1 - e^{-\alpha(h-t_2)}), \beta_1, \beta_1)^T \quad (\text{OU})$$

**Space of Expectations.** Expressions (2.4) and (2.6) allow us to link the parameter  $\boldsymbol{\theta}$  to the probability distribution of observations  $\mathbf{Y}$  and to explore identifiability issues. In this linear formulation, detecting shifts boils down to identifying the non-zero components of  $\boldsymbol{\Delta}$ . The following lemma highlights the parallels between solutions of the BM and OUsun processes:

**Lemma 2.2.1** (Similar Solutions). *Let  $\mathbf{m}_Y \in \mathbb{R}^n$  be a vector,  $\mathcal{T}$  an ultrametric tree,  $\alpha$  a positive real number, and  $\sigma, \gamma$  non-negative real numbers. Then there exists at least one vector  $\boldsymbol{\Delta}^{\text{BM}}, \boldsymbol{\Delta}^{\text{OU}} \in \mathbb{R}^{m+n}$  (respectively,  $\boldsymbol{\Delta}^{\text{OU}} \in \mathbb{R}^{m+n}$ ), such that the vector of expectations*

at the tips of a  $BM(\sigma, \Delta^{BM})$  (respectively, an  $OU_{sun}(\gamma, \alpha, \Delta^{OU})$ ) running on the tree  $\mathcal{T}$  is exactly  $\mathbf{m}_Y$ .

Furthermore,  $\Delta^{BM}$  is a solution to this problem for the BM if and only if  $\Delta^{OU} = \mathbf{W}(\alpha)^{-1} \Delta^{BM}$  is a solution for the  $OU_{sun}$ , and  $\Delta^{BM}$  and  $\mathbf{W}(\alpha)^{-1} \Delta^{BM}$  have the same support. These two vectors are said to be similar.

*Proof.* The first part of this lemma follows directly from formulas (2.4) (BM) and (2.6) (OU). Indeed, the maps  $\Delta \mapsto \mathbf{T}\Delta$  and  $\Delta \mapsto \mathbf{T}\mathbf{W}(\alpha)\Delta$  both span  $\mathbb{R}^n$ . The second part of the lemma is a consequence of  $\mathbf{W}(\alpha)$  being diagonal and invertible (for  $\alpha > 0$ ).  $\square$

*Remark 2.2.1.* Lemma 2.2.1 shows that the  $OU_{sun}$  and BM processes that induce a given  $\mathbf{m}_Y$  use shifts located on the same branches, although they may differ on other parameters.

## 2.3 Identifiability and Complexity of a Model

### 2.3.1 Identifiability Issues

As we only have access to  $\mathbf{Y}$ , and not  $\mathbf{X}$ , we only have partial information about the shifts occurrence on the tree. In fact, several different allocations of the shifts can produce the same trait distribution at the tips, and hence are not identifiable. In other words, there exists parameters  $\theta \neq \theta'$  with the same likelihood function:  $p_\theta(\cdot) = p_{\theta'}(\cdot)$ . Note that the notion of identifiability is intrinsic to the model and affects all estimation methods. Restricting ourselves to the parsimonious allocations of shifts only partially alleviates this issue, and, using a “random cluster model” representation of the problem, we are able to enumerate, first, all the equivalent solutions to a given problem, and, second, all the equivalence classes for a given number of shifts.

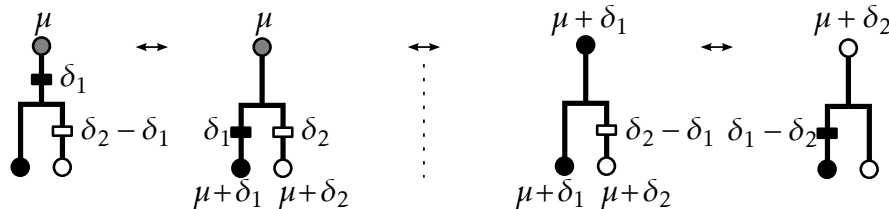


Figure 2.3.1 – Equivalent allocations in the BM case. Mean tip values are represented by colors and equal for all allocations. The two allocations on the right are parsimonious.

**No Homoplasy Assumption.** We assume in the following that there is no convergent evolution. This means that each shift creates a new (and unique) mean trait value for extant species that are below it. This assumption is reasonable considering that shifts are real valued and makes the model similar to “infinite alleles” models in population genetics. This assumption confines but does not eliminate the identifiability issue, as seen in Figure 2.3.1.

#### 2.3.1.1 Definition of the problem

Figure 2.3.1 shows a simple example where the model is not identifiable in the BM case. Here, four distinct allocations give the same mean values  $(\mu + \delta_1, \mu + \delta_2)$  at the tips. The lack of identifiability is due to the non-invertibility of the tree matrix  $\mathbf{T}$ .



**Proposition 2.3.1** (Kernel of the Tree Matrix  $\mathbf{T}$ ). *Let  $i$  be an internal node,  $i \in \llbracket 1, m \rrbracket$ , with  $L_i$  children nodes  $(d_1, \dots, d_{L_i}) \in \llbracket 2, m+n \rrbracket^{L_i}$ . Then the vector  $\mathbf{K}^i$  defined as follow:*

$$\forall j \in \llbracket 1, m+n \rrbracket, K_j^i = \begin{cases} 1 & \text{if } j = i \\ -1 & \text{if } j \in (d_1, \dots, d_{L_i}) \\ 0 & \text{otherwise} \end{cases}$$

*is in the kernel of  $\mathbf{T}$ . In addition, the  $m$  vectors constructed this way form a basis of the kernel space of  $\mathbf{T}$ .*

These kernel vectors effectively “cancel out” a shift on a branch by balancing it with the opposite shift on all immediate child branches. Note that the root mean value is treated as a shift.

*Proof of Proposition 2.3.1.* Let  $i \in \llbracket 1, m \rrbracket$  be an internal node with  $L_i$  children nodes  $(d_1, \dots, d_{L_i})$  and  $\mathbf{K}^i$  the corresponding vector, defined as in the proposition. Then, for any  $j \in \llbracket 1, m+n \rrbracket$ :

$$(\mathbf{UK}^i)_j = U_{ji}K_i^i + \sum_{l=1}^{L_i} U_{jd_l}K_{d_l}^i = \mathbb{I}\{i \in \text{anc}(j)\} - \sum_{l=1}^{L_i} \mathbb{I}\{d_l \in \text{anc}(j)\}$$

We can then distinguish three possibilities:

- If  $i \notin \text{anc}(j)$ , then  $d_l \notin \text{anc}(j)$  for all  $l \in \llbracket 1, L_i \rrbracket$  and  $(\mathbf{TK}^i)_j = 0$ .
- If  $j = i$ , then  $i \in \text{anc}(j)$ , and, by definition,  $d_l \notin \text{anc}(j)$  for any  $l \in \llbracket 1, L_i \rrbracket$ , so  $(\mathbf{UK}^i)_i = 1$ .
- Else, if  $i$  is an ancestor of  $j$ , with  $i \neq j$ , then, as  $i$  is internal, one (and only one) of its child  $d_l$  is also an ancestor of  $j$  (potentially  $j$  itself), so that the sum cancels out.

This proves that

$$\forall i \in \llbracket 1, m \rrbracket, \mathbf{UK}^i = (\delta_{ij})_{1 \leq j \leq m+n} \quad (2.7)$$

In particular, as  $\mathbf{TK}^i$  is the vector of the last  $n$  coordinates of  $\mathbf{UK}^i$ , this shows that the vectors  $(\mathbf{K}^1, \dots, \mathbf{K}^m)$  are in the kernel of  $\mathbf{T}$ .

Then, as we found  $m$  independent vectors in the kernel of  $\mathbf{T}$ , which is a space of dimension lower than  $m$  (as the  $n$  columns of  $\mathbf{T}$  representing tips are linearly independent, and by the rank theorem), this family of vectors is a basis of the kernel space.  $\square$

The following lemma describes the relationships that exist between these kernel vectors and the tree matrix  $\mathbf{U}$  defined in Section 2.2.3.

**Lemma 2.3.1.** *Let  $b$  be the canonical basis of  $\mathbb{R}^{m+n}$ , and  $S$  a supplementary space of  $\ker(\mathbf{T})$ . Then  $b' = (\mathbf{K}^1, \dots, \mathbf{K}^m, \mathbf{b}_{m+1}, \dots, \mathbf{b}_{m+n})$  is a basis adapted to the decomposition  $\ker(\mathbf{T}) \oplus S$ , and the matrix  $\mathbf{U}$  (as defined in Section 2.2.3) is the change of basis matrix between  $b$  and  $b'$ .*

*As a consequence,  $\mathbf{U}$  is invertible.*

*Proof of lemma 2.3.1.* First,  $(\mathbf{b}_{m+1}, \dots, \mathbf{b}_{m+n})$  is a family of  $n$  independent vectors of  $S$  of dimension  $n$ , so is a basis of  $S$ , and  $b'$  is a basis adapted to  $\ker(\mathbf{T}) \oplus S$ .

Let  $i \in \llbracket 1, m+n \rrbracket$ . Let's show that  $\mathbf{U}\mathbf{b}'_i = \mathbf{b}_i$ . If  $m+1 \leq i \leq m+n$ , then  $\mathbf{b}'_i = \mathbf{b}_i$ , and  $\mathbf{U}\mathbf{b}_i = \mathbf{b}_i$  is the  $i^{\text{th}}$  column of  $\mathbf{U}$ . Otherwise, if  $1 \leq i \leq m$ , then  $\mathbf{b}'_i = \mathbf{K}^i$ , and, from equation (2.7),  $\mathbf{U}\mathbf{b}'_i = \mathbf{b}_i$ . This shows that  $\mathbf{U}$  is the change of basis matrix between  $b$  and  $b'$ .  $\square$

**“Random Cluster Model” Representation.** When inferring the shifts, we have to keep in mind this problem of non-identifiability, and be able to choose, if necessary, one or several possible allocations among all the equivalent ones. In order to study the properties of the allocations, we use a *random cluster model*, as defined in Mossel & Steel (2004). The following definition states the problem as a node coloring problem.

**Definition 2.3.1** (Node Coloring). Let  $\mathcal{C}_K$  be a set of  $K$  arbitrary “colors”,  $K \in \mathbb{N}^*$ . For a given shift allocation, the color of each node is given by the application  $B: \llbracket 1, m+n \rrbracket \rightarrow \mathcal{C}_K$  recursively defined in the following way:

- Choose a color  $c \in \mathcal{C}_K$  for the root:  $B(1) = c$ .
- For a node  $i$ ,  $i \in \llbracket 2, m+n \rrbracket$ , set  $B(i)$  to  $B(\text{pa}(i))$  if there is no shift on branch  $i$ , otherwise choose another color  $c$ ,  $c \in \mathcal{C}_K \setminus \{B(\text{pa}(i))\}$ , and set  $B(i)$  to  $c$ .

Hereafter, we identify  $(\mathcal{C}_K)^{\llbracket 1, m+n \rrbracket}$  with  $(\mathcal{C}_K)^{m+n}$  and refer to a node coloring indifferently as an application or a vector.

As the shifts only affect  $\mathbb{E}[\mathbf{X}]$  and we only have access to  $\mathbb{E}[\mathbf{Y}]$ , we identify colors with the distinct values of  $\mathbb{E}[\mathbf{Y}]$ :

**Definition 2.3.2** (Adapted Node Coloring). A node coloring is said to be *adapted* to a shifted random process on a tree if two *tips* have the same color if and only if they have the same mean value under that process.

**Proposition 2.3.2** (Adapted Coloring for BM and OUsun). *Let  $\sigma$  and  $\gamma$  be two non-negative real numbers, and  $\alpha$  a positive real number. Then:*

- (i) *In the BM case, if  $\mathcal{C}$  is the set of possible mean values taken by the nodes of the tree, then the knowledge of the node colors is equivalent to the knowledge of  $\Delta$ . Furthermore, the associated node coloring is adapted to the original BM.*
- (ii) *In the OUsun case, from lemma 2.2.1, we can find a similar BM process, i.e. with shifts on the same branches. Then the knowledge of the node coloring associated to this similar BM process is equivalent to the knowledge of the vector of shifts of the OUsun, and the node coloring obtained is adapted to the original OUsun.*

*Proof of Proposition 2.3.2.* The proof of (i) relies on expression (2.4), that states that  $\mathbb{E}[\mathbf{X}] = \mathbf{U}\Delta$ . Defining  $\mathcal{C}$  as the set of all distinct values of  $\mathbb{E}[\mathbf{X}]$ , we can identify  $\mathbb{E}[\mathbf{X}]$  with the node coloring application that maps any node  $i$  with  $\mathbb{E}[X]_i$ . Since  $\mathbf{U}$  is invertible (see lemma 2.3.1 above), we can go from one formalism to the other.

For (ii), we use lemma 2.2.1 to find a similar BM, and then use (i).  $\square$

From now on, we will study the problem of shifts allocation as a discrete-state coloring problem.



### 2.3.1.2 Parsimony

As we saw on Figure 2.3.1 there are multiple colorings of the internal nodes that lead to a given tips coloring. Among all these solutions, we choose to study only the *parsimonious* ones. This property can be seen as an optimality condition, as defined below:

**Definition 2.3.3** (Parsimonious Allocation). Given a vector of mean values at the tips produced by a given shifted stochastic process running on the tree, an adapted node coloring is said to be *parsimonious* if it has a minimum number of color changes. We denote by  $\mathcal{S}_K^P$  the set of parsimonious allocations of  $K$  shifts on the  $(m+n-1)$  branches of the tree (not counting the root branch).

As  $K$  shifts cannot produce more than  $K+1$  colors, we can define an application  $\phi : \mathcal{S}_K^P \rightarrow (\mathcal{C}_{K+1})^n$  that maps a parsimonious allocation of shifts to its associated tip partition.

**Definition 2.3.4** (Equivalence). Two allocations are said to be *equivalent* (noted  $\sim$ ) if they produce the same partition of the tips and are both parsimonious. Mathematically:

$$\forall s_1, s_2 \in \mathcal{S}_K^P, s_1 \sim s_2 \iff \phi(s_1) = \phi(s_2)$$

In other words, two allocations are equivalent if they produce the same tip *coloring* up to a permutation of the colors. Given  $d \in (\mathcal{C}_{K+1})^n$  a coloring of the tips of  $\mathcal{T}$  with  $K+1$  colors,  $\phi^{-1}(d)$  is the set of equivalent parsimonious node coloring that coincide with  $d$  (up to a permutation of the colors) on the tree leaves.

Several dynamic programming algorithms already exist to compute the minimal number of shifts required to produce a given tips coloring, and to find one associated parsimonious solution (see Fitch, 1971; Sankoff, 1975; Felsenstein, 2004). Here, we need to be a little more precise, as we want to both count and enumerate all possible equivalent node colorings associated with a tip coloring. For the sake of brevity, we only present the algorithm that counts  $|\phi^{-1}(d)|$ , for  $d \in (\mathcal{C}_K)^n$ . This algorithm can be seen as a corollary of the enumeration algorithm (presented and proved in Appendix 2.A) and an extension of Fitch algorithm where we keep track of both the cost of an optimal coloring and the number of such colorings. It has  $O(K^2Ln)$  time complexity where  $L$  is the maximal number of children of the nodes of the tree.

**Proposition 2.3.3** (Size of an equivalence class). *Let  $d$  be a coloring of the tips,  $d \in (\mathcal{C}_K)^n$ , and let  $i$  be a node of tree  $\mathcal{T}$  with  $L_i$  daughter nodes  $(i_1, \dots, i_{L_i})$ ,  $L_i \geq 2$ . Denote by  $\mathcal{T}_i$  the sub-tree rooted at node  $i$ .*

*For  $k \in \mathcal{C}_K$ ,  $S_i(k)$  is the cost of starting from node  $i$  with color  $k$ , i.e. the minimal number of shifts needed to get the coloring of the tips of  $\mathcal{T}_i$  defined by  $d$ , when starting with node  $i$  in color  $k$ . Denote by  $T_i(k)$  the number of allocations on  $\mathcal{T}_i$  that achieve cost  $S_i(k)$ .*

*If  $i$  is a tip ( $m+1 \leq i \leq m+n$ ), then,*

$$S_i(k) = \begin{cases} 0 & \text{if } d(i) = k \\ +\infty & \text{otherwise} \end{cases} \quad T_i(k) = \begin{cases} 1 & \text{if } d(i) = k \\ 0 & \text{otherwise} \end{cases}$$

*Otherwise, if  $i$  is a node, for  $1 \leq l \leq L_i$ , define the set of admissible colors for daughter  $i_l$ :*

$$\mathcal{K}_k^l = \operatorname{argmin}_{p \in \mathcal{C}_K} \{S_{i_l}(p) + \mathbb{I}\{p \neq k\}\}$$

As these sets are not empty, let  $(p_1, \dots, p_L) \in \mathcal{K}_k^1 \times \dots \times \mathcal{K}_k^L$ . Then:

$$S_i(k) = \sum_{l=1}^L S_i(p_l) + \mathbb{I}\{p_l \neq k\} \quad \text{and} \quad T_i(k) = \prod_{l=1}^L \sum_{p_l \in \mathcal{K}_k^l} T_i(p_l)$$

At the root, if  $\mathcal{L} = \operatorname{argmin}_{k \in \mathcal{K}} S_1(k)$ , then  $|\phi^{-1}(d)| = \sum_{k \in \mathcal{L}} T_1(k)$ .

**OU Practical Case.** We can illustrate this notion on a simple example. We consider an OUsun on a random tree of unit height (total height  $h = 1$ ). We put three shifts on the tree, producing a given trait distribution. Then, using proposition 2.3.2 and our enumeration algorithm, we can reconstruct the 5 possible allocations of shifts that produce the exact same distribution at the tips. These solutions are shown in Figure 2.3.2. Note that the colors are not defined by the values of the optimal regime  $\beta$ , but by the mean values  $\mathbb{E}[\mathbf{Y}]$  of the process at the tips. As a result, the groups shown in blue and red in the first solution have the same optimal value in this configuration, but not in any other. The second solution shown illustrates the fact that all the shifts values are inter-dependent, as changing the position of only one of them can have repercussions on all the others. Finally, the third solution shows that the timing of shifts matters: to have the same impact as an old shift, a recent one must have a much higher intensity (under constant selection strength such as in the OUsun).

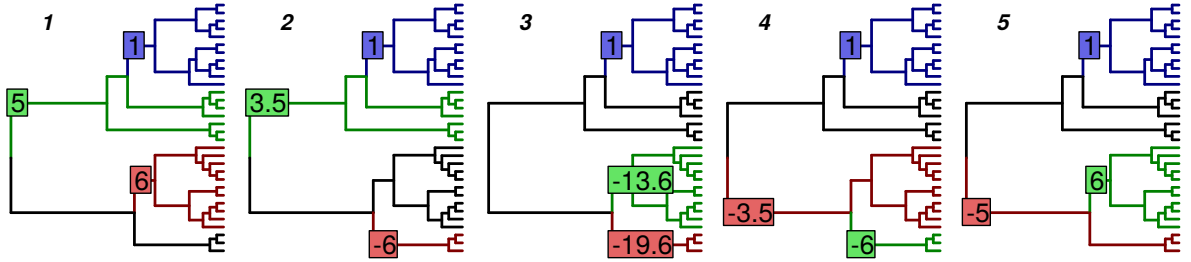


Figure 2.3.2 – Five equivalent shift allocations that produce colorings that are adapted to an OUsun, with  $\alpha = 3$  and  $\gamma^2 = 0.1$ . The box at the root represents the ancestral optimum  $\beta_1$ , and the boxes on the branches represent the positions and values of the shifts on the optimal value. While accounting for very different evolutionary scenarios, all allocations produce the same trait distribution at the tips.

**Possible Relaxation of the No-homoplasy Assumption.** Note that the algorithms used for counting and enumerating the configurations of an equivalence class are valid even without the no-homoplasy hypothesis. The no-homoplasy hypothesis is however crucial in the next Section to establish a link between the number of shifts and the number of distinct tips colors.

### 2.3.2 Complexity of a Collection of Models

**Number of Different Tips Colors.** As we make the inference on the parameters with a fixed number of shifts  $K$  (see Section 2.4.1), we need a model selection procedure to choose  $K$ . This procedure depends on the *complexity* of the collection of models that use  $K$  shifts, defined as the number of *distinct* models. To do that, we count the number of *tree-compatible* partitions of the tips into  $K + 1$  groups, as defined in the next proposition:

**Proposition 2.3.4.** *Under the no homoplasy assumption, an allocation of  $K$  shifts on a tree is parsimonious if and only if it creates exactly  $K + 1$  tip colors. The tip partition into  $K + 1$  groups associated with this coloring is said to be tree-compatible. The set  $\mathcal{D}_{K+1} \subset (\mathcal{C}_{K+1})^n$  of such partitions is the image of  $\mathcal{S}_K^P$  by the map  $\phi$  defined in the previous section.*

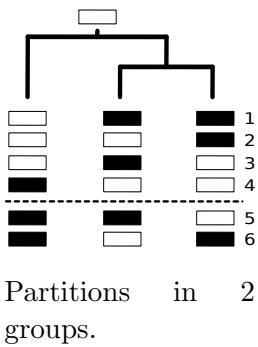
*Proof of Proposition 2.3.4.* First, note that  $K$  shifts create at most  $K + 1$  colors. If each shift produces a new tip mean value (no homoplasy), the only way to create  $K$  or less colors is to “forget” one of the shifts, i.e. to put shifts on every descendant of the branch where it happens. Such an allocation is not parsimonious, as we could just add the value of the forgotten shift to all its descendant to get the same coloring of the tips with one less shift. So a parsimonious allocation cannot create less than  $K + 1$  colors, and hence creates exactly  $K + 1$  colors.

Reciprocally, if an allocation with  $K$  shifts that produces  $p$  groups is not parsimonious, then we can find another parsimonious one that produces the same  $p$  groups with  $p - 1$  shifts, with  $p - 1 < K$ , i.e.  $p < K + 1$ . So, by contraposition, if the allocation produces  $K + 1$  groups, then it is parsimonious.  $\square$

Using the equivalence relation defined in Definition 2.3.4, we can formally take the quotient set of  $\mathcal{S}_K^P$  by the relation  $\sim$  to get the set of parsimonious allocations of  $K$  shifts on the  $m + n - 1$  branches of the tree that are identifiable:  $\mathcal{S}_K^{PI} = \mathcal{S}_K^P / \sim$ . In other words, the set  $\mathcal{S}_K^{PI}$  is constituted of one representative of each equivalence class. Under the no homoplasy assumption, there is thus a bijection between identifiable parsimonious allocations of  $K$  shifts and tree-compatible partitions of the tips in  $K + 1$  groups:  $\mathcal{S}_K^{PI} \xrightarrow{\sim} \mathcal{D}_{K+1}$ .

The number  $N_{K+1}^{(T)} = |\mathcal{D}_{K+1}|$  is the complexity of the class of models with  $K$  shifts defined as the number of distinct identifiable parsimonious possible configurations one can get with  $K$  shifts on the tree. To compute  $N_K^{(T)}$ , we will need  $M_K^{(T)}$  the number of *marked* tree-compatible partitions in  $K$  groups. These are composed of all the tree-compatible partitions where one group, among those that could be in the same state as the root, is distinguished with a mark (see example 2.3.1 below).

*Example 2.3.1* (Difference between  $N_2^{(T)}$  and  $M_2^{(T)}$ ).



- If we consider only unmarked partitions, then colorings 1, 2 and 3 induce the same partitions as, respectively, colorings 4, 5 and 6, and  $N_2^{(T)} = 3$ .
- For marked partitions, fix the root state to an arbitrary color, for instance white, and consider the white group as marked. Then colorings 5 and 6 are not tree-compatible (they require two shifts). And although they induce the same partition, colorings 1 and 4 correspond to different marked partitions: each marks a different group of leaves. Therefore  $M_2^{(T)} = 4$ .

**Proposition 2.3.5** (Computation of the Number of Equivalent Classes). *Let  $i$  be a node of tree  $\mathcal{T}$ , and  $K \in \mathbb{N}^*$ .*

*If  $i$  is a tip, then  $N_K^{(T_i)} = M_K^{(T_i)} = \mathbb{I}\{K = 1\}$ .*

Else, if  $i$  is a node with  $L_i$  daughter nodes  $(i_1, \dots, i_{L_i})$ ,  $L_i \geq 2$ , then:

$$\begin{cases} N_K^{(\mathcal{T}_i)} = \sum_{\substack{I \subset \llbracket 1, L_i \rrbracket \\ |I| \geq 2}} \sum_{\substack{k_1 + \dots + k_{L_i} = K + |I| - 1 \\ k_1, \dots, k_{L_i} \geq 1}} \prod_{l \in I} M_{k_l}^{(\mathcal{T}_{i_l})} \prod_{l \notin I} N_{k_l}^{(\mathcal{T}_{i_l})} + \sum_{\substack{k_1 + \dots + k_{L_i} = K \\ k_1, \dots, k_{L_i} \geq 1}} \prod_{l=1}^{L_i} N_{k_l}^{(\mathcal{T}_{i_l})} \\ M_K^{(\mathcal{T}_i)} = \sum_{\substack{I \subset \llbracket 1, L_i \rrbracket \\ |I| \geq 1}} \sum_{\substack{k_1 + \dots + k_{L_i} = K + |I| - 1 \\ k_1, \dots, k_{L_i} \geq 1}} \prod_{l \in I} M_{k_l}^{(\mathcal{T}_{i_l})} \prod_{l \notin I} N_{k_l}^{(\mathcal{T}_{i_l})} \end{cases} \quad (2.8)$$

In the binary case, this relation becomes, if  $i$  has two daughters  $i_\ell$  and  $i_r$ :

$$\begin{cases} N_K^{(\mathcal{T}_i)} = \sum_{\substack{k_1 + k_2 = K \\ k_1, k_2 \geq 1}} N_{k_1}^{(\mathcal{T}_{i_\ell})} N_{k_2}^{(\mathcal{T}_{i_r})} + \sum_{\substack{k_1 + k_2 = K + 1 \\ k_1, k_2 \geq 1}} M_{k_1}^{(\mathcal{T}_{i_\ell})} M_{k_2}^{(\mathcal{T}_{i_r})} \\ M_K^{(\mathcal{T}_i)} = \sum_{\substack{k_1 + k_2 = K \\ k_1, k_2 \geq 1}} M_{k_1}^{(\mathcal{T}_{i_\ell})} N_{k_2}^{(\mathcal{T}_{i_r})} + N_{k_1}^{(\mathcal{T}_{i_\ell})} M_{k_2}^{(\mathcal{T}_{i_r})} + \sum_{\substack{k_1 + k_2 = K + 1 \\ k_1, k_2 \geq 1}} M_{k_1}^{(\mathcal{T}_{i_\ell})} M_{k_2}^{(\mathcal{T}_{i_r})} \end{cases} \quad (2.9)$$

*Proof.* We will prove this proposition in the binary case, the general case being a natural extension of it. If  $\mathcal{T}$  is a binary tree with  $\mathcal{T}_\ell$  and  $\mathcal{T}_r$  as left and right sub-trees, one faces two situations when partitioning the tips in  $K$  groups:

- The left and right sub-trees have no group in common. Then, the number of groups in  $\mathcal{T}$  is equal to the number of groups in its two sub-trees, and there are  $\sum_{k_1 + k_2 = K} N_{k_1}^{(\mathcal{T}_\ell)} N_{k_2}^{(\mathcal{T}_r)}$  such partitions. This is the first term of the equation on  $N_K^{(\mathcal{T})}$  in (2.9).
- The left and right sub-trees have at least one group in common. Then, from the no homoplasy assumption, they have exactly one group in common: the ancestral state of the root. Suppose that this ancestral state is marked. Then it must be present in the two sub-trees, and there are  $\sum_{k_1 + k_2 = K + 1} M_{k_1}^{(\mathcal{T}_\ell)} M_{k_2}^{(\mathcal{T}_r)}$  such partitions. This ends the proof of the formula on  $N_K^{(\mathcal{T})}$ .

To get the formula on  $M_K^{(\mathcal{T})}$ , we use the same kind of arguments. The second part of the formula is the same as the one for  $N_K^{(\mathcal{T})}$ , and the first part corresponds to trees for which the marked partition is present in only one of the two sub-trees.  $\square$

The complexity of the algorithm described above is  $O(2^L(K+L)^L L n)$ . Note that  $N_K^{(\mathcal{T})}$  depends on the topology of the tree  $\mathcal{T}$  in general. However, if the tree is binary, a closed form solution of the recurrence relation (2.8), which does not depend on the topology, exists.

**Corollary 2.3.1** (Closed Formula Binary Trees). *For a rooted binary tree with  $n$  tips, we have:*

$$N_{K+1}^{(\mathcal{T})} = N_{K+1}^{(n)} = |\mathcal{S}_K^{PI}| = \binom{2n - 2 - K}{K} \text{ and } M_K^{(\mathcal{T})} = M_K^{(n)} = \binom{2n - K}{K - 1}$$

The demonstration of this formula is not straightforward, and is based on a Vandermonde-like equality, detailed in Appendix 2.B. The formula is then obtained using a strong induction on the number of tips of the tree.

*Remark 2.3.1.* Note that, when  $K$  is large compared to  $\sqrt{n}$ , the average number of configurations per equivalence class goes to infinity. This can be checked by comparing the total number of configurations  $\binom{2n-1}{K-1}$  with the total number of classes  $\binom{2n-K-1}{K-1}$ . As a consequence, we only consider models for which  $K < \sqrt{n}$  in the remainder.

*Remark 2.3.2.* This formula was already obtained in a different context in Steel (1992) (Proposition 1) and, with a slightly different formulation, in Semple & Steel (2003, Proposition 4.1.4). In these works, the authors are interested in counting the “ $r$ -states convex characters on a binary tree”. Under the no-homoplasmy assumption, this number can be shown to be equal to  $|\mathcal{S}_{r-1}^{PI}|$ .

### 2.3.3 Another Characterization of Parsimony

The following proposition gives an alternative definition of parsimony under the no-homoplasmy hypothesis using the linear formulation of the problem. It will be used for model selection in Section 2.4.2.

**Proposition 2.3.6** (Equivalence between parsimony and independence). *Let  $\mathbf{m}_Y$  be a given mean vector,  $\mathbf{m}_Y \in \mathbb{R}^n$ , and  $\Delta$  a vector of shifts such that  $\mathbf{T}\Delta = \mathbf{m}_Y$ , with  $\mathbf{T}$  the tree matrix defined in Section 2.2.3. Under the no homoplasmy assumption, the vector of shifts  $\Delta$  is parsimonious if and only if the corresponding column-vectors of the tree matrix  $(T_i)_{i \in \text{Supp}(\Delta)}$  are linearly independent.*

*Proof of Proposition 2.3.6.* By contraposition, let’s first assume that the vector-columns  $(T_i)_{i \in \text{Supp}(\Delta)}$  are linearly dependent, and prove that  $\Delta$  is not parsimonious. This means that we can find a vector  $\mathbf{E}$ ,  $\mathbf{E} \in \mathbb{R}^{m+n}$ , such that  $\text{Supp}(\mathbf{E}) \subset \text{Supp}(\Delta)$ , and  $\mathbf{T}\mathbf{E} = 0$ . We can hence find  $j \in \text{Supp}(\Delta)$ ,  $j > 1$ , such that  $E_j \neq 0$ . Then if  $\lambda = -\Delta_j/E_j$ , the vector  $\Delta' = \Delta + \lambda\mathbf{E}$  is a vector of shifts on the tree with one less non-zero coordinate than  $\Delta$  such that  $\mathbf{T}\Delta' = \mathbf{m}_Y$ . Hence,  $\Delta$  is not parsimonious.

Reciprocally, by contraposition, assume that  $\Delta$  is not parsimonious. Then by proposition 2.3.4, it produces  $p$  groups, with  $p \leq K$ . Hence the application associated with  $(T_i)_{i \in \text{Supp}(\Delta)}$  goes from a space of dimension  $K + 1$  to a space of dimension  $p \leq K$ , and hence is not injective, and the family  $(T_i)_{i \in \text{Supp}(\Delta)}$  is not independent.  $\square$

## 2.4 Statistical Inference

### 2.4.1 Expectation Maximization

**Principle.** As shown in Section 2.2.2, both BM and OUsun models can be seen as incomplete data models. The Expectation Maximization algorithm (EM, Dempster et al., 1977) is a widely used algorithm for likelihood maximization of these kinds of models. It is based on the decomposition:  $\log p_\theta(\mathbf{Y}) = \mathbb{E}_\theta[\log p_\theta(\mathbf{Z}, \mathbf{Y}) | \mathbf{Y}] - \mathbb{E}_\theta[\log p_\theta(\mathbf{Z} | \mathbf{Y}) | \mathbf{Y}]$ . Given an estimate  $\theta^{(h)}$  of the parameters, we need to compute some moments of  $p_{\theta^{(h)}}(\mathbf{Z} | \mathbf{Y})$  (E step), and then find a new estimate  $\theta^{(h+1)} = \arg\max_\theta \mathbb{E}_{\theta^{(h)}}[\log p_\theta(\mathbf{Z}, \mathbf{Y}) | \mathbf{Y}]$  (M step). The parameters are given for the BM and OUsun in subsection 2.2.1. We assume here that the number of shifts  $K$  is fixed.

We only provide the main steps of the EM. Additional details can be found in Appendix 2.C.

**E step.** As  $\mathbf{X}$  is Gaussian, the law of the hidden variables  $\mathbf{Z}$  knowing the observed variables  $\mathbf{Y}$  is entirely defined by its expectation and variance-covariance matrix, and can be computed using classical formulas for Gaussian conditioning. The needed moments of  $\mathbf{Z} | \mathbf{Y}$  can also be computed using a procedure that is linear in the number of tips (called “Upward-downward”) that takes advantage of the tree structure and bypasses inversion of the variance-covariance matrix (see [Lartillot, 2014](#), for a similar algorithm).

**Complete Likelihood Computation.** Using the model described in Section 2.2.2, we can use the following decomposition of the complete likelihood:

$$p_{\theta}(\mathbf{X}) = p_{\theta}(X_1) \prod_{j=2}^{m+n} p_{\theta}(X_j | X_{\text{pa}(j)})$$

Each term of this product is then known, and we easily get  $\mathbb{E}_{\theta^{(h)}}[\log p_{\theta}(\mathbf{Z}, \mathbf{Y}) | \mathbf{Y}]$ .

**M step.** The difficulty comes here from the discrete variables (location of shifts on the branches). The maximization is exact for the BM but we only raise the objective function for the OUsun, hence computing a Generalized EM (GEM, see [Dempster et al., 1977](#)). This stems from the independent increment nature of the BM: shifts only affect  $p_{\theta}(X_j | X_{\text{pa}(j)})$  on the branches where they occur and the maximization reduces to finding the  $K$  highest components of a vector, which has complexity  $O(n + K \log(n))$ . By contrast, OUsun has autocorrelated increments: shifts affect  $p_{\theta}(X_j | X_{\text{pa}(j)})$  on the branches where they occur and on all subsequent branches. Maximization is therefore akin to segmentation on a tree, which has complexity  $O(n^K)$ .

**Initialization.** Initialization is always a crucial step when using an EM algorithm. Here, we use the linear formulation (2.4) or (2.6), and initialize the vector of shifts using a Lasso regression. The selection strength  $\alpha$  is initialized using pairs of tips likely to be in the same group.

## 2.4.2 Model Selection

**Model Selection in the iid Case with Unknown Variance.** Model selection in a linear regression setting has received a lot of attention over the last few years. In [Baraud et al. \(2009\)](#), the authors developed a non-asymptotic method for model selection in the case where the errors are independent and identically distributed (iid), with an unknown variance. In the following, we first recall their main results, and then adapt it to our setting of non-independent errors.

We assume that we have the following model of *independent* observations:

$$\mathbf{Y}' = \mathbf{s}' + \gamma \mathbf{E}' \quad \text{with} \quad \mathbf{E}' \sim \mathcal{N}(0, \mathbf{I}_n)$$

and we define a collection  $\mathcal{S}' = \{S'_{\eta}, \eta \in \mathcal{M}\}$  of linear subspaces of  $\mathbb{R}^n$  that we call *models*, and that are indexed by a finite or countable set  $\mathcal{M}$ . For each  $\eta \in \mathcal{M}$ , we denote by  $\hat{\mathbf{s}}'_{\eta} = \text{Proj}_{S'_{\eta}} \mathbf{Y}'$  the orthogonal projection of  $\mathbf{Y}'$  on  $S'_{\eta}$ , that is a least-square estimator of  $\mathbf{s}'$ , and  $\mathbf{s}'_{\eta} = \text{Proj}_{S'_{\eta}} \mathbf{s}'$  the projection of  $\mathbf{s}'$ .

We extract from [Baraud et al. \(2009\)](#) the following theorem, that bounds the risk of the selected estimator, and provides us with a non-asymptotic guarantee. It relies on a penalty depending on the EDkhi function, as defined below:



**Definition 2.4.1** (Baraud et al., 2009, Section 4, definitions 2 and 3). Let  $D, N$  be two positive integers, and  $X_D, X_N$  be two independent  $\chi^2$  random variables with degrees of freedom  $D$  and  $N$  respectively. For  $x \leq 0$ , define

$$\text{Dkhi}[D, N, x] = \frac{1}{\mathbb{E}[X_D]} \mathbb{E} \left[ \left( X_D - x \frac{X_N}{N} \right)_+ \right]$$

And define  $\text{EDkhi}[D, N, q]$  as the unique solution of the equation  $\text{Dkhi}[D, N, \text{EDkhi}[D, N, q]] = q$  (for  $0 < q \leq 1$ ).

**Theorem 2.4.1** (Baraud et al. (2009), Section 4, theorem 2 and corollary 1). *In the setting defined above, let  $D_\eta$  be the dimension of  $S'_\eta$ , and assume that  $N_\eta = n - D_\eta \geq 2$  for all  $\eta \in \mathcal{M}$ . Let  $\mathcal{L} = \{L_\eta\}_{\eta \in \mathcal{M}}$  be some family of positive numbers such that  $\Omega' = \sum_{\eta \in \mathcal{M}} (D_\eta + 1)e^{-L_\eta} < +\infty$ , and assume that, for  $A > 1$ ,*

$$\text{pen}(\eta) = \text{pen}_{A, \mathcal{L}}(\eta) = A \frac{N_\eta}{N_\eta - 1} \text{EDkhi}[D_\eta + 1, N_\eta - 1, e^{-L_\eta}]$$

Take  $\hat{\eta}$  as the minimizer of the criterion:  $\hat{\eta} = \text{argmin}_{\eta \in \mathcal{M}} \left\| \mathbf{Y}' - \hat{\mathbf{s}}'_\eta \right\|^2 \left( 1 + \frac{\text{pen}(\eta)}{N_\eta} \right)$ .

Then, assuming that  $N_\eta \geq 7$  and  $\max(L_\eta, D_\eta) \leq \kappa n$  for any  $\eta \in \mathcal{M}$ , with  $\kappa < 1$ , the following non-asymptotic bound holds:

$$\mathbb{E} \left[ \frac{\left\| \mathbf{s}' - \hat{\mathbf{s}}'_{\hat{\eta}} \right\|^2}{\gamma^2} \right] \leq C(A, \kappa) \left[ \inf_{\eta \in \mathcal{M}} \left\{ \frac{\left\| \mathbf{s}' - \mathbf{s}'_\eta \right\|^2}{\gamma^2} + \max(L_\eta, D_\eta) \right\} + \Omega' \right]$$

The penalty used here ensures an oracle inequality: in expectation, the risk of the selected estimator is bounded by the risk of the best possible estimator of the collection of models, up to a multiplicative constant, and a residual term that depends on the dimension of the oracle model. Note that if the collection of models is poor, such an inequality has low value. We refer to Baraud et al. (2009) for a more detailed discussion of this result.

**Adaptation to the Tree-Structured Framework.** We use the linear formulation described in 2.2.3, and assume that we are in the OUsun model (this procedure would also work for a BM with a deterministic root). Then, if  $\mathbf{V}$  is a matrix of size  $n$ , with  $V_{ij} = e^{-\alpha d_{ij}}, \forall (i, j) \in \llbracket 1, n \rrbracket^2$ , we have:

$$\mathbf{Y} = \mathbf{TW}(\alpha)\mathbf{\Delta} + \gamma\mathbf{E} = \mathbf{s} + \gamma\mathbf{E} \quad \mathbf{E} \sim \mathcal{N}(0, \mathbf{V})$$

We assume that  $\alpha$  is fixed, so that the design matrix  $\mathbf{TW}(\alpha)$  and the structure matrix  $\mathbf{V}$  are known and fixed. A *model* is defined here by the position of the shifts on the branches of the tree, i.e. by the non-zero components of  $\mathbf{\Delta}$  (with the constraint that the first component, the root, is always included in the model). We denote by  $\mathcal{M} = \bigcup_{K=0}^{p-1} \mathcal{S}_K^{PI}$  the set of allowed (parsimonious) allocations of shifts on branches (see Section 2.3.2),  $p$  being the maximum allowed dimension of a model. From proposition 2.3.6, for  $\eta \in \mathcal{M}$ , the columns vectors  $\mathbf{T}_\eta$  are linearly independent, and the model  $S_\eta = \text{Span}(\mathbf{T}_i, i \in \eta)$  is a linear sub-space of  $\mathbb{R}^n$  of dimension  $D_\eta = |\eta| = K_\eta + 1$ ,  $K_\eta$  being the number of shifts

in model  $\eta$ . Note that as  $\mathbf{W}(\alpha)$  is diagonal invertible, it does not affect the definition of the linear subspaces. The set of models is then  $\mathcal{S} = \{S_\eta, \eta \in \mathcal{M}\}$ .

We define the Mahalanobis norm associated to  $\mathbf{V}^{-1}$  by:  $\|\mathbf{R}\|_{\mathbf{V}^{-1}} = \mathbf{R}^T \mathbf{V}^{-1} \mathbf{R}$ ,  $\forall \mathbf{R} \in \mathbb{R}^n$ .

The projection on  $S_\eta$  according to the metric defined by  $\mathbf{V}^{-1}$  is then:

$$\hat{\mathbf{s}}_\eta = \text{Proj}_{S_\eta}^{\mathbf{V}^{-1}}(\mathbf{Y}) = \underset{\mathbf{a} \in S_\eta}{\text{argmin}} \|\mathbf{Y} - \mathbf{a}\|_{\mathbf{V}^{-1}}^2 \quad \text{and} \quad \mathbf{s}_\eta = \text{Proj}_{S_\eta}^{\mathbf{V}^{-1}}(\mathbf{s})$$

For a given number of shifts  $K$ , we define the best model with  $K$  shifts as the one maximizing the likelihood, or, equivalently, minimizing the least-square criterion for models with  $K$  shifts:

$$\hat{\mathbf{s}}_K = \underset{\eta \in \mathcal{S}, |\eta|=K+1}{\text{argmin}} \|\mathbf{Y} - \hat{\mathbf{s}}_\eta\|_{\mathbf{V}^{-1}}^2$$

The idea is then to slice the collection of models by the number of shifts  $K$  they employ. Thanks to the EM algorithm above, we are able to select the best model in such a set. The problem is then to select a reasonable number of shifts. To compensate the increase in the likelihood due to over-fitting, using the model selection procedure described above, we select  $K$  using the following penalized criterion:

$$\text{Crit}_{LS}(K) = \|\mathbf{Y} - \hat{\mathbf{s}}_K\|_{\mathbf{V}^{-1}}^2 \left(1 + \frac{\text{pen}(K)}{n - K - 1}\right) \quad (2.10)$$

As noted in [Baraud et al. \(2009\)](#), the previous criterion can equivalently be re-written in term of likelihood, as:

$$\text{Crit}_{LL}(K) = \frac{n}{2} \log \left( \frac{\|\mathbf{Y} - \hat{\mathbf{s}}_K\|_{\mathbf{V}^{-1}}^2}{n} \right) + \frac{1}{2} \text{pen}'(K) \quad (2.11)$$

with  $\text{pen}'(K) = n \log \left(1 + \frac{\text{pen}(K)}{n - K - 1}\right)$ . As we use maximum-likelihood estimators, we chose this formulation for the implementation. The following proposition then holds:

**Proposition 2.4.1** (Form of the Penalty and guaranties ( $\alpha$  known)). *Let  $\mathcal{L} = \{L_K\}_{K \in \llbracket 0, p-1 \rrbracket}$ , with  $p \leq \min\left(\frac{\kappa n}{2 + \log(2) + \log(n)}, n - 7\right)$ , the maximum dimension of a model, with  $\kappa < 1$ , and:*

$$L_K = \log |S_K^{PI}| + 2 \log(K + 2), \forall K \in \llbracket 0, p - 1 \rrbracket \quad (2.12)$$

Let  $A > 1$  and assume that

$$\text{pen}_{A, \mathcal{L}}(K) = A \frac{n - K - 1}{n - K - 2} \text{EDkhi}[K, n - K - 2, e^{-L_K}]$$

Suppose that  $\hat{K}$  is a minimizer of (2.10) or (2.11) with this penalty. Then:

$$\mathbb{E} \left[ \frac{\|\mathbf{s} - \hat{\mathbf{s}}_{\hat{K}}\|_{\mathbf{V}^{-1}}^2}{\gamma^2} \right] \leq C(A, \kappa) \inf_{\eta \in \mathcal{M}} \left\{ \frac{\|\mathbf{s} - \mathbf{s}_\eta\|_{\mathbf{V}^{-1}}^2}{\gamma^2} + (K_\eta + 2)(3 + \log(n)) \right\}$$

with  $C(A, \kappa)$  a constant depending on  $A$  and  $\kappa$  only.

The proof of this proposition can be found in Appendix 2.E. It relies on theorem 2.4.1, adapting it to our tree-structured observations.



*Remark 2.4.1.* With this oracle inequality, we can see that we are missing the oracle by a  $\log(n)$  term. This term is known to be unavoidable, see [Baraud et al. \(2009\)](#) for further explanations.

*Remark 2.4.2.* Note that the chosen penalty may depend on the topology of the tree through the term  $|\mathcal{S}_K^{PI}|$  (see [Section 2.3.2](#)).

*Remark 2.4.3.* The penalty involves a constant  $A > 1$ , that needs to be chosen by the user. Following [Baraud et al. \(2009\)](#) who tested a series of values, we fixed this constant to  $A = 1.1$ .

## 2.5 Simulations Studies

### 2.5.1 Simulations Scheme

We tested our algorithm on data simulated according to an OUsun, with varying parameters. The simulation scheme is inspired by the work of [Uyeda & Harmon \(2014\)](#). We first generated three distinct trees with, respectively, 64, 128 and 256 tips, using a pure birth process with birth rate  $\lambda = 0.1$ . The tree heights were scaled to one, and their topology and branch lengths were fixed for the rest of the simulations. We then used a star-like simulation study scheme, fixing a base scenario, and exploring the space of parameters one direction at the time. The base scenario was taken to be relatively “easy”, with  $\beta_1 = 0$  (this parameter was fixed for the rest of the simulations),  $\alpha_b = 3$  (i.e.  $t_{1/2,b} = 23\%$ ),  $\gamma_b^2 = 0.5$  and  $K_b = 5$ . The parameters then varied in the following ranges: the phylogenetic half life  $t_{1/2} = \ln(2)/\alpha$  took 11 values in  $[0.01, 10]$ ; the root variance  $\gamma^2 = \frac{\sigma^2}{2\alpha}$  took 9 values in  $[0.05, 25]$ ; the number of shifts  $K$  took 9 values in  $[0, 16]$  (see [Figures 2.5.2-2.5.3](#) for the exact values taken). The problem was all the more *difficult* that  $\gamma^2$ ,  $t_{1/2}$  or  $K$  were large.

For each simulation, the  $K$  shifts were generated in the following way. First, their values were drawn according to a mixture of two Gaussian distributions,  $\mathcal{N}(4, 1)$  and  $\mathcal{N}(-4, 1)$ , in equal proportions. The mixture was chosen to avoid too many shifts of small amplitude. Then, their positions were chosen to be balanced: we first divided the tree in  $K$  segments of equal heights, and then randomly drew in each segment an edge where to place a shift. We only kept parsimonious allocations.

Each of these configurations was repeated 200 times, leading to 16200 simulated data sets. An instance of a tree with the generated data is plotted in [Figure 2.5.1](#).

### 2.5.2 Inference Procedures

For each generated dataset, we ran our EM procedure with fixed values of  $K \in \llbracket 0, \lfloor \sqrt{n} \rfloor \rrbracket$ ,  $n$  being the number of tips of the tree. Remark that for  $n = 64$ ,  $\lfloor \sqrt{n} \rfloor = 8$ , and we have no hope of detecting true values of  $K$  above 8 (see [remark 2.3.1](#) for an explanation of the bound in  $\sqrt{n}$ ). The number of shifts  $K_s$  was chosen thanks to our penalized criterion, and we kept inferences corresponding to both  $K_s$  and the true number  $K_t$ .

We ran two sets of estimations for  $\alpha$  either known or estimated. The computations took respectively 66 and 570 (cumulated) days of CPU time. This amounts to a mean computational time of around 6 minutes (367 seconds) for one estimation when  $\alpha$  is fixed, and 52 minutes (3137 seconds) when  $\alpha$  is estimated, with large differences between easy and difficult scenarios.

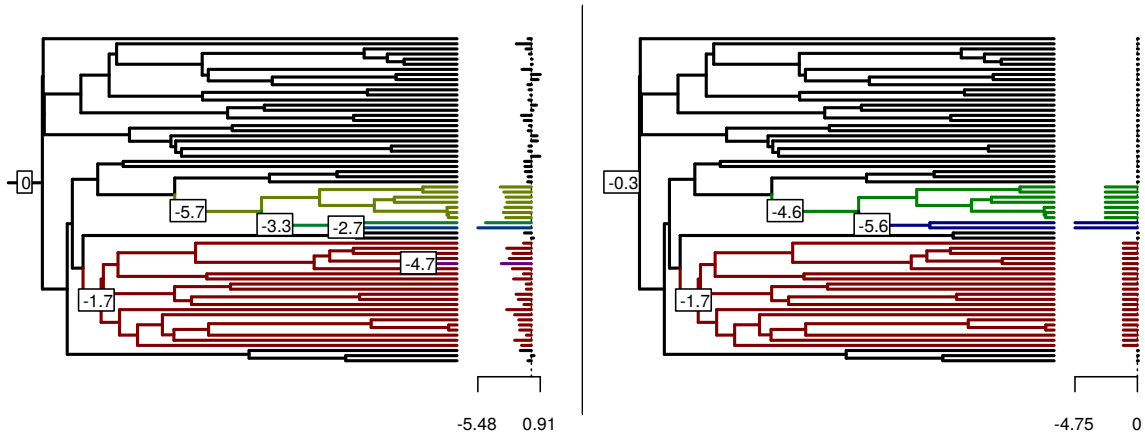


Figure 2.5.1 – *Left*: Simulated configuration (with  $t_{1/2} = 0.75$ ,  $\gamma^2 = 0.5$  and  $K = 5$ ). The shifts positions and values are marked on the tree. The value of the character generated (positive or negative) is represented on the right. The colors of the branches correspond to the true regimes, black being the ancestral state. *Right*: One of the three equivalent allocations of shifts for the model inferred from the data, with corresponding vector of mean tip values. Shifts not recovered are located on pendant edges, and have low influence on the data. The two other equivalent allocations can be easily deduced from this one.

### 2.5.3 Scores Used

The convergence of the EM algorithm was assessed through the comparison of the likelihood of the true and estimated parameters, and the comparison of mean number of EM steps needed when  $\alpha$  is fixed or estimated. The quality of the estimates of  $\beta_1$ ,  $t_{1/2}$  and  $\gamma^2$  was assessed using the coefficient of variation. The model selection procedure was evaluated by comparing the true number of shifts with the estimated one, which should be lower. We do not expect to find the exact number as some shifts, which are too small or too close of the tips, cannot be detected. To evaluate the quality of the clustering of the tips, the only quantity we can observe, we used the Adjusted Rand Index (ARI, [Hubert & Arabie, 1985](#)) between the true clustering of the tips, and the one induced by the estimated shifts. The ARI is proportional to the number of concordant pairs in two clusterings and has maximum value of 1 (for identical clusterings) and expected value of 0 (for random clusterings). Note that this score is conservative as shifts of small intensity, which are left aside by our model selection procedure, produce “artificial” groups that cannot be reconstructed.

### 2.5.4 Results

The selection strength is notoriously difficult to estimate, with large ranges of values giving similar behaviors (see [Thomas et al., 2014](#)). We hence first analyse the impact of estimating  $\alpha$  on our estimations, showing that the main behavior of the algorithm stays the same. Then, we study the shifts reconstruction procedure.

**Convergence and Likelihood.** For  $\alpha$  known, all estimations converged in less than 49 iterations, with a median number of 13 iterations. For  $\alpha$  estimated, the number of iterations increased greatly, with a median of 69, and a fraction of estimations (around

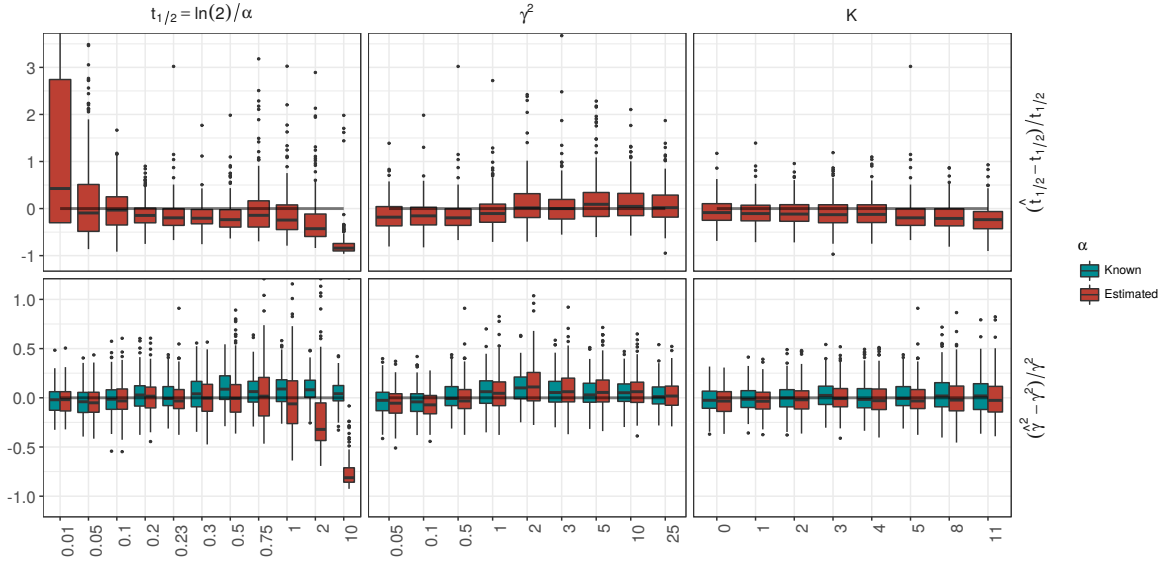


Figure 2.5.2 – Box plots over the 200 repetitions of each set of parameters, for the phylogenetic half-life (top) and root variance (bottom) with  $K$  estimated, and  $\alpha$  fixed to its true value (blue) or estimated (red), on a tree with 128 taxa. For better legibility, the  $y$ -axis of these two rows were re-scaled, omitting some outliers (respectively, for  $t_{1/2}$  and  $\gamma^2$ , 0.82% and 0.46% of points are omitted). The whisker of the first box for  $t_{1/2}$  goes up to 7.5.

3.2%) that reached the maximum allowed number (fixed at 1000 iterations) without converging. Unsurprisingly, the more difficult the problem, the more iterations were needed. The log-likelihoods of the estimated parameters are close to the true ones, even when  $\alpha$  is estimated (see supplementary Figure 2.F.3 in Appendix 2.F, first row).

**Estimation of Continuous Parameters.** Figure 2.5.2 (first row) shows that we tend to slightly over-estimate  $\alpha$  in general. The estimation is particularly bad for large values of  $\alpha$  (with a high variance on the result, see first box of the row), and low values of  $\alpha$ . In this regime, the model is “over-confident”, as it finds a higher selection strength than the real one and therefore a smaller variance (second row of Figure 2.5.2). For smaller and bigger trees, the estimators behave in the same way, but with degraded or improved values, as expected. We also note that taking the true number of shifts instead of the estimated one slightly degrades our estimation of these parameters (see supplementary Figure 2.F.3 in Appendix 2.F). The estimation of  $\beta_1$  is not affected by the knowledge of  $\alpha$  or  $K$  (see Figure 2.5.3, first row), and only has an increased variance for more difficult configurations. In the remainder, we only show results obtained for estimated  $\alpha$  as estimating  $\alpha$  does not impact ARI,  $\hat{K}$  and  $\hat{\beta}_0$  (see supplementary Figure 2.F.4 in Appendix 2.F).

**Estimation of the Number of Shifts.** The way shifts were drawn ensures that they are not too small in average, and that they are located all along the tree. Still, some shifts have a very small influence on the data, and are hence hard to detect (see Figure 2.5.1). The selection model procedure almost always under-estimates the number of shifts, ex-

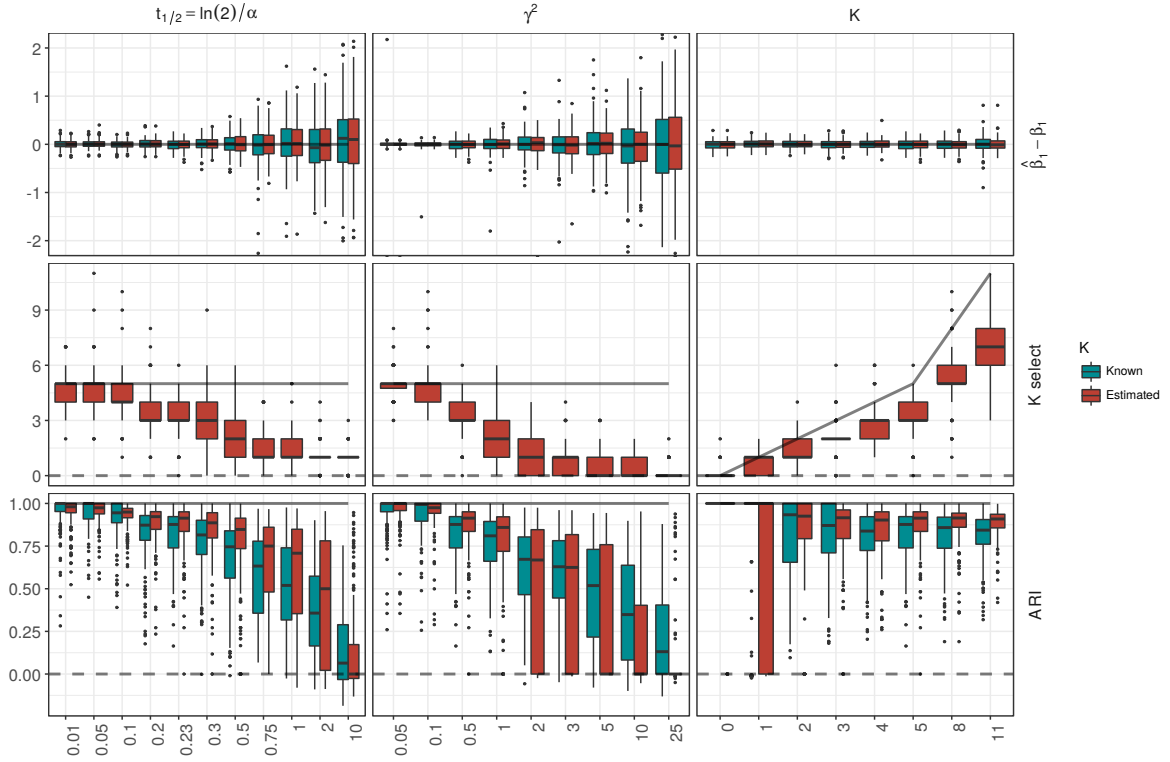


Figure 2.5.3 – Same for  $\beta_1$  (top), the number of shifts (middle) and ARI (bottom), with  $\alpha$  estimated, and  $K$  fixed to its true value (blue) or estimated (red). As previously, the  $y$ -axis of the first row ( $\beta_1$ ) was re-scaled, omitting some outliers (1.39% of points are omitted).

cept in very favorable cases (Figure 2.5.3, second row). This behavior is nonetheless expected, as allowing more shifts does not guarantee that the right shifts will be found (see supplementary Figures 2.F.1 and 2.F.2 in Appendix 2.F).

**Clustering of the Tips.** The ARI tends to be degraded for small values of  $\alpha$  or high variance, but remains positive (Figure 2.5.3, third row). When only one shift occurs, the ARI is very unstable, but for any other value of  $K$ , it stays quite high. Finally, knowing the number of shifts does not improve the ARI.

**Equivalent Solutions.** When  $\alpha$  and  $K$  are both estimated, only 5.1% of the configurations have 2 or more equivalent solutions. One inferred configuration with three equivalent solutions is presented Figure 2.5.1.

**Comparison with bayou.** As mentioned above, our simulation scheme, although not completely equivalent to the scheme used in Uyeda & Harmon (2014), is very similar, so that we can compare our results with theirs. The main differences lies in the facts that we took a grid on  $\gamma^2 = \sigma^2/2\alpha$  instead of  $\sigma^2$ , and that we took shifts with higher intensities, making the detection of shifts easier. We can see that we get the same qualitative behaviors for our estimators, with the selection strength  $\alpha$  over or under estimated, respectively, in small or large values regions. The main difference lies in

the estimation of the number of shifts. Maybe because of the priors they used ( $K \sim \text{Conditional Poisson}(\lambda = 9, K_{\max} = n/2)$ ), they tend to estimate similar numbers of shifts (centered on 9) for any set of parameters. In particular, while our method seems to be quite good at detecting situations where there are no shifts at all, theirs seems unable to catch these kind of configurations, despite the fact that their shifts have low intensity, leading to a possible over-fitting of the data.

Overall, the behavior of the algorithm is quite satisfying. Our model selection procedure avoids over-fitting, while recovering the correct clustering structure of the data. It furthermore allows for a reasonable estimation of the continuous parameters, except for  $\alpha$  which is notoriously difficult to estimate.

## 2.6 Case Study: Chelonian Carapace Length Evolution

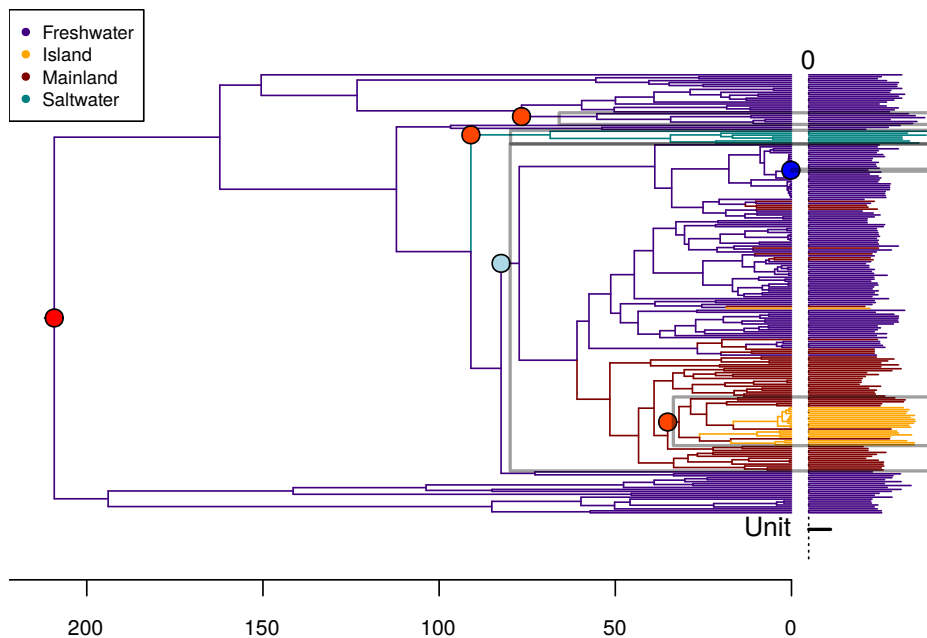


Figure 2.6.1 – Phylogenetic tree of the Chelonians. Log-transformed trait values are represented on the right. Branch colors represent the habitats. The shifts found by our EM algorithm are shown as circles, with a color indicating the value of the shift, from blue (negative) to red (positive). Boxes highlight the groups induced by the shifts. The  $x$ -scale is in million years.

### 2.6.1 Description of the Dataset

Extant species of the order Testudines, or Chelonii, are turtles and tortoises, living all across the globe, and exhibiting a wide variation in body size, from the small desert speckled tortoise (*Homopus signatus*, 10 cm), to the large marine leatherback sea turtle (*Dermochelys coriacea*, 244 cm). In order to test the hypothesis of island and marine gigantism, that could explain the extreme variations observed, [Jaffe et al. \(2011\)](#) compiled a dataset containing a measure of the carapace length for 226 species, along with a phylogenetic tree of these species, spanning 210 million years (my) (see Figure 2.6.1). They

assigned each species to one of four habitats: mainland-terrestrial, freshwater, marine and island-terrestrial. Then, testing several fixed regimes allocations on the branches of the tree using the method described in [Butler & King \(2004\)](#), they found the best support in favor of a “OU2” model that assigned one regime to each habitat. Following [Uyeda & Harmon \(2014\)](#), we will refer to this model as the “OU<sub>habitat</sub>” model. Note that this model is ambiguously defined, as it requires to assign a habitat to each ancestral species. Using proposition [2.3.3](#), we found that there were 48 equivalent parsimonious ways of doing so that respect the habitats observed at the tips of the tree. One of these habitat reconstruction is presented [Figure 2.6.1](#).

## 2.6.2 Method

We used the version of the dataset embedded in the package `geiger` ([Pennell et al., 2014](#)), that contains a phylogenetic tree and a vector of log-carapaces lengths. The corresponding habitats are reported in the Appendix of [Jaffe et al. \(2011\)](#).

We ran our algorithm with a number of shifts going from 0 to 20. Rather than estimating  $\alpha$  directly within the EM as we did for the simulations, we took  $\alpha$  varying on a grid, taking 6 values regularly spaced between 0.01 and 0.1, but fixed for each estimation. We found that this approach, although computationally more intensive, gave better results. These  $6 \times 20 = 120$  estimations took around 2 hours of CPU time. For each number of shifts, going again from 0 to 20, we kept the solution with the maximal likelihood, and we applied the model selection criterion to them. This method gave a solution with 5 shifts, and a selection strength of 0.06 (i.e. 5.5 % of the total height of the tree). Using a finer grid for  $\alpha$  gives highly similar results, allocating shifts to the same edges. These last estimations are given below.

## 2.6.3 Results

Our method selected a solution with 5 shifts, a rather strong selection strength ( $t_{1/2} = 5.4\%$  of the tree height), and a rather low root variance ( $\gamma^2 = 0.22$ , see [table 2.1](#), first column). Two of those shifts are closely related to the habitats defined in [Jaffe et al. \(2011\)](#) (see [Figure 2.6.1](#)). The ancestral optimal value, that applies here to two clades of freshwater turtles, is estimated to be around 38 cm. A small decrease in size for a large number of mainland and freshwater turtles is found (optimal value 24 cm). Marine turtles (super-family Cheloniodea) are found to have an increased carapace length (with an optimal value of 130 cm), as well as a clade containing soft-shell tortes (family Trionychidae, optimal size 110 cm), and a clade containing almost all island tortoises, including several sub-species of Galápagos tortoises (*Geochelone nigra*). Only the Ryukyu black-breasted leaf turtle (*Geoemyda japonica*), endemic to the Ryukyu Islands in Japan, and distant on the phylogenetic tree, is not included in this group. Note that the group also contains some mainland tortoises of the genus *Geochelone*, that are closely related to Galápagos tortoises. This is typical of our method: it constructs groups that are both phenotypically and phylogenetically coherent. Finally, one species is found to have its own group, the black-knobbed map turtle (*Graptemys nigrinoda*), with a very low optimal value of  $1.4 \times 10^{-20}$  cm, for a measured trait of 15 cm. The fact that the shift has a very high negative value ( $-49$  in log scale) is just an artifact due to the actualization factor on a very small branch (0.18 my, for an inferred phylogenetic half-life of 11 my). This is a rather unexpected choice of shift location. When considering the

linear model as transformed by the cholesky matrix of the variance to get independent errors (as in the proof of proposition 2.4.1), we find a leverage of 0.94, indicating that this species trait behaves in the transformed space as an outsider.

#### 2.6.4 Comparison with other methods

In order to compare our results to previously published ones, we reproduced some of the analysis already conducted on this dataset. We hence ran the methods described in Jaffe et al. (2011) (using the R package OUwie, with fixed positions for the shifts), Uyeda & Harmon (2014) (implemented in package bayou), Ingram & Mahler (2013) (package SURFACE), and Ho & Ané (2014) (function OUshifts in package phylolm). See Section 2.F.3 in Appendix 2.F for more details on these methods and the parameters we used.

The shifts allocated on the tree by methods bayou, SURFACE and OUshifts are presented on Supplementary Figure 2.F.5 (Appendix 2.F). We can see that 3 among the most strongly supported shifts in the posterior distribution given by bayou, as well as some among the oldest shifts found by SURFACE and OUshifts are similar to the ones found by our method. The bayou method finds equal support for many shifts, all over the tree, and the median of the posterior distribution is 17 shifts, which is pretty close to the mode of the prior put on the number of shifts (15). The SURFACE and OUshifts methods select respectively 33 and 8 shifts, including many on pendant edges, that are not easily interpretable. The backward step of SURFACE allowed to merge the regimes found for marine turtles and soft-shell tortoises that our method found to have very similar optimal values. The results of the five methods are summarized Table 2.1. Note that these models are not nested, due to the status assigned to the root, and to the possible convergences.

Compared to step-wise heuristics, our integrated maximum likelihood based approach allows us to have a more “global” view of the tree, and hence to select a solution that accounts better for the global structure of the trait distribution. Thanks to its rigorous model selection procedure, our model seems to report significant shifts only, that are more easily interpretable than the solutions found by other methods, and that do not rely on any chosen prior.

	Habitat	EM	bayou	SURFACE	OUshifts
Number of shifts	16	5	17	33	8
Number of regimes	4	6	18	13	9
lnL	-133.86	-97.59	-91.54	30.38	-79.79
Marginal lnL	NA	NA	-149.09	NA	NA
$\alpha$ ( $\times h$ , per my)	9.32	12.76	36.54	1.72	3.25
$\ln 2/\alpha$ (my)	15.56	11.36	3.97	84.28	44.64
$\sigma^2$ ( $\times h$ , per my)	6.21	5.57	11.91	0.72	2.29
$\gamma^2$	0.33	0.22	0.16	0.21	0.35
CPU time (min)	65.25	134.49	136.81	634.16	8.28

Table 2.1 – Summary of the results obtained with several methods for the Chelonian Dataset. For bayou, the median of the posterior distributions is given.



## Acknowledgments

We would like to thank Cécile Ané for helpful discussions on an early draft of this manuscript. We are grateful to the INRA MIGALE bioinformatics platform (<http://migale.jouy.inra.fr>) for providing the computational resources needed for the experiments. We also thank the two anonymous reviewers whose careful and critical reading greatly helped improve this manuscript.

## Toward the Multivariate Analysis

In this chapter, we presented a general framework for the analysis of univariate trait evolution with shifts on a phylogenetic tree. It sets the ground for a multivariate trait study, that is the object of the next chapter. To scale up to several traits, the main change we make in the inference framework is the way we deal with the OU process. Instead of developing heuristics to maximize the OU likelihood, as we did in this chapter (see Section 2.C.3), we make use of the re-scaling trick presented in the introductory chapter (see Section 1.4.2.4). This allows us to reduce the OU to a more manageable BM, for which we have exact and fast algorithms. To validate this new inference method, we applied it to univariate datasets, so that we could compare its results with the ones of the previous method. The three main results (not shown) were: (1) the new method gives very similar results to the old one, with likelihoods equal up to numerical accuracy, (2) the new method is substantially faster than the old one (with speed even improved thanks to the efficient implementation of the “upward-downward” algorithm) and (3) the new method is much more sensitive to the initialization, as the EM algorithm almost never moves away from the first shift configuration. This last feature is a bit surprising, and quite unsatisfactory. Special care was hence taken in the design of the initialization step, performed thanks to a lasso regression.



## Appendix

### 2.A Enumeration of Equivalence Classes

**Definition 2.A.1** (Coloring concatenation). Let  $i$  be a node of tree  $\mathcal{T}$  with  $L_i$  daughter nodes  $(i_1, \dots, i_{L_i})$ ,  $L_i \geq 2$ , and assume that the tips are colored according to the application  $d \in (\mathcal{C}_K)^n$ . We denote by  $\mathcal{T}_i$  the sub-tree rooted at node  $i$ , and by  $\mathcal{A}_{\mathcal{T}_i}(d)$  the set of parsimonious shifts allocations on  $\mathcal{T}_i$  that produce a coloring of the tips compatible with  $d$ . At the root,  $\mathcal{A}_{\mathcal{T}_1}(d) = \mathcal{A}_{\mathcal{T}}(d) = \phi^{-1}(d)$ .

- For  $k \in \mathcal{C}_K$ ,  $S_i(k)$  is the *cost* of starting from node  $i$  with color  $k$ , *i.e.* the minimal number of shifts needed to get the right coloring of the tips of  $\mathcal{T}_i$ , when starting with color  $k$ .
- $S_i^{tot} = \min_{k \in \mathcal{C}_K} S_i(k)$  is the minimal cost of subtree  $\mathcal{T}_i$ , *i.e.* the number of shifts of a parsimonious coloring.  $\mathcal{L}_i = \operatorname{argmin}_{k \in \mathcal{C}_K} S_i(k)$  is the set of colors root  $i$  can take in a parsimonious coloring of sub-tree  $\mathcal{T}_i$ .
- For  $k \in \mathcal{C}_K$ ,  $\mathcal{B}_{\mathcal{T}_i}^k$  is the set of colorings of  $\mathcal{T}_i$  that respect the colors at the tips, have  $S_i(k)$  shift, and start with color  $k$ .
- For  $\mathcal{K} \subset \mathcal{C}_K$ ,  $\mathcal{B}_{\mathcal{T}_i}^{\mathcal{K}} = \bigcup_{k \in \mathcal{K}} \mathcal{B}_{\mathcal{T}_i}^k$ . Hence,  $\mathcal{A}_{\mathcal{T}_i}(d) = \mathcal{B}_{\mathcal{T}_i}^{\mathcal{L}_i}$ , and the computation of  $\mathcal{A}_{\mathcal{T}_i}(d)$  only requires the computation of  $S_i(k)$  and  $\mathcal{B}_{\mathcal{T}_i}^k$  for any  $k \in \mathcal{C}_K$ .
- For  $(p_1, \dots, p_{L_i}) \in (\mathcal{C}_K)^{L_i}$ , and  $(B_1, \dots, B_{L_i}) \in \mathcal{B}_{\mathcal{T}_{i_1}}^{p_1} \times \dots \times \mathcal{B}_{\mathcal{T}_{i_{L_i}}}^{p_{L_i}}$  we define, for  $k \in \mathcal{C}_K$ , the concatenation  $B = \bigoplus_{l=1}^{L_i} B_l, B \in \mathcal{B}_{\mathcal{T}_i}^k$ , by: 
$$\begin{cases} B(i) = k \\ B(j) = B_l(j) & \text{if } i \in \mathcal{T}_{i_l} \end{cases}$$
 As the sub-trees  $\mathcal{T}_{i_l}, l \in \llbracket 1, L_i \rrbracket$  do not overlap, this application is correctly defined on the nodes of  $\mathcal{T}_i$ .

Using these definitions, we can state the following recursion formula:

**Proposition 2.A.1** (Enumeration Recursion Formula). *Let  $k \in \mathcal{C}_K$ , and  $i \in \llbracket 1, m+n \rrbracket$ . If  $i$  is a tip of the tree, then*

$$S_i(k) = \begin{cases} 0 & \text{if } d(i) = k \\ +\infty & \text{otherwise} \end{cases} \quad \mathcal{B}_{\mathcal{T}_i}^k = \begin{cases} \{i \mapsto k\} & \text{if } d(i) = k \\ \emptyset & \text{otherwise} \end{cases}$$

*If  $i$  is a node of tree  $\mathcal{T}$  with  $L_i$  daughter nodes  $(i_1, \dots, i_{L_i})$ ,  $L_i \geq 2$ , and assuming that  $S_{i_l}(k)$  and  $\mathcal{B}_{\mathcal{T}_{i_l}}^k$  are known for any  $l \in \llbracket 1, L_i \rrbracket$  and  $k \in \mathcal{C}_K$ , define, for  $l \in \llbracket 1, L_i \rrbracket$ :*

$$\mathcal{K}_k^l = \operatorname{argmin}_{1 \leq p \leq K} \{S_{i_l}(p) + \mathbb{I}\{p \neq k\}\}$$

*As these sets are not empty, let  $(p_1, \dots, p_{L_i}) \in \mathcal{K}_k^1 \times \dots \times \mathcal{K}_k^{L_i}$ . Then*

$$S_i(k) = \sum_{l=1}^{L_i} (S_{i_l}(p_l) + \mathbb{I}\{p_l \neq k\}) \quad \text{and} \quad \mathcal{B}_{\mathcal{T}_i}^k = \left\{ \bigoplus_{l=1}^{L_i} B_l : \forall l \in \llbracket 1, L_i \rrbracket, B_l \in \mathcal{B}_{\mathcal{T}_{i_l}}^{\mathcal{K}_k^l} \right\}$$

*Proof.* The actualization of  $S_i(k)$  is the same as in the Sankoff algorithm (Sankoff, 1975). The set  $\mathcal{B}_{T_i}^k$  is then obtained by enumerating all the possible ways of concatenating children sets  $\mathcal{B}_{T_{i_l}}^{k_l}$ , each of which is the ensemble of solutions for the sub-tree  $T_{i_l}$  that realize the minimal number of shifts when starting in state  $k$ .  $\square$

Remarking that  $T_i(k) = |\mathcal{B}_{T_i}^k|$ , proposition 2.3.3 of the main text follows immediately.

## 2.B A Vandermonde Like Identity

**Proposition 2.B.1.** *Let  $(n, n') \in \mathbb{N}$  and  $K \in \mathbb{N}$ . With the standard convention that  $\binom{n}{k} = 0$  if  $n < k$ ,*

$$\binom{n+n'-K}{K} = \sum_{k=0}^K \binom{n-k}{k} \binom{n'-K+k}{K-k} + \sum_{k=0}^{K-1} \binom{(n-1)-k}{k} \binom{(n'-1)-(K-1)+k}{(K-1)-k}$$

which can be rewritten in a more symmetric way as:

$$\binom{n+n'-K}{K} = \sum_{\substack{k, k' \geq 0 \\ k+k'=K}} \binom{n-k}{k} \binom{n'-k'}{k'} + \sum_{\substack{k, k' \geq 0 \\ k+k'=K-1}} \binom{(n-1)-k}{k} \binom{(n'-1)-k'}{k'} \quad (2.13)$$

Similarly,

$$\begin{aligned} \binom{n+n'+1-K}{K} &= \sum_{k=0}^K \binom{n-k}{k} \binom{n'-K+k}{K-k} \\ &\quad + \sum_{k=0}^{K-1} \binom{(n-1)-k}{k} \binom{n'-(K-1)+k}{(K-1)-k} + \binom{n-k}{k} \binom{(n'-1)-(K-1)+k}{(K-1)-k} \end{aligned}$$

which can be rewritten in a more symmetric way as:

$$\begin{aligned} \binom{n+n'+1-K}{K} &= \sum_{\substack{k, k' \geq 0 \\ k+k'=K}} \binom{n-k}{k} \binom{n'-k'}{k'} \\ &\quad + \sum_{\substack{k, k' \geq 0 \\ k+k'=K-1}} \binom{(n-1)-k}{k} \binom{n'-k'}{k'} + \binom{n-k}{k} \binom{(n'-1)-k'}{k'} \end{aligned} \quad (2.14)$$

Note that Eq (2.13) generalizes in some way the Vandermonde identity which states

$$\binom{n+n'}{K} = \sum_{k=0}^K \binom{n}{k} \binom{n'}{K-k} \quad (2.15)$$

Although several proofs of the Vandermonde identity are known (geometric, algebraic and combinatorial), we only provide a geometric proof of this Vandermonde-like identity.

Consider a grid of size  $(n+n') \times K$ . We are interested in grid-valued paths that can move either by  $(1, 0)$  or by  $(2, 1)$ . In other words, if the  $k^{\text{th}}$  position of a path is  $(x_k, y_k)$ ,

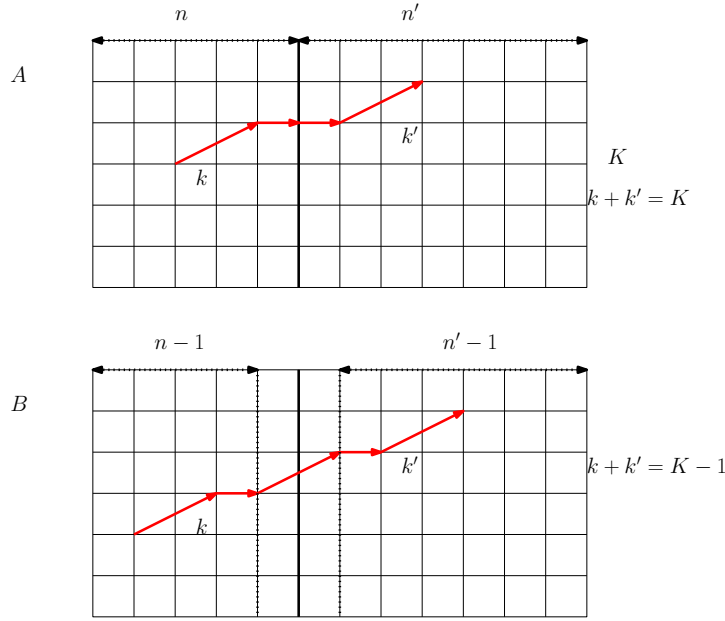


Figure 2.B.1 – Partition of paths according to whether they reach (A) or cross (B) the line  $x = n$

then its next position  $(x_{k+1}, y_{k+1})$  is either  $(x_k + 1, y_k)$  or  $(x_k + 2, y_k + 1)$ . We are interested in paths starting at  $(0, 0)$  and ending at  $(n + n', K)$ .

Such a path consists of  $K$  moves of type  $(2, 1)$  and  $n + n' - 2K$  moves of type  $(1, 0)$  and is uniquely determined by the positions of the moves of the former type. There are  $\binom{n+n'-2K+K}{K} = \binom{n+n'-K}{K}$  distinct positions and therefore as many such paths.

We now sort the paths according to the value  $i$  they take when either reaching the line  $x = n$  or reaching the line  $x = n + 1$  without reaching the line  $x = n$  first. We refer to the latter paths as crossing the line  $x = n$ . Note that this sorting induces a partition of all paths (see Figure 2.B.1)

A path reaching  $x = n$  at position  $i$  uniquely gives rise to two paths: one from  $(0, 0)$  to  $(n, i)$  and one from  $(n, i)$  to  $(n + n', K)$  or equivalently from  $0$  to  $(n', K - i)$ . There are  $\binom{n-i}{i}$  different paths of the first kind and  $\binom{n'-K-i}{K-i}$  of the second. There are therefore  $\binom{n-i}{i} \binom{n'-K-i}{K-i}$  paths that pass through  $(n, i)$ .

A path crossing the line  $x = n$  and reaching the line  $x = n + 1$  at  $i$  must do so with a last move of type  $(2, 1)$ . It therefore uniquely defines a path from  $(0, 0)$  to  $(n - 1, i - 1)$  and a path from  $(n + 1, i)$  to  $(n + n', K)$ , or equivalently from  $(0, 0)$  to  $(n' - 1, K - i)$ . There are therefore  $\binom{n-i}{i-1} \binom{n'-1-K+i}{K-i}$  paths that cross the line  $x = n$  and pass through  $(n + 1, i)$ .

Putting everything together, we get:

$$\begin{aligned} \binom{n+n'-K}{K} &= \sum_{i=0}^K \binom{n-i}{i} \binom{n'-K+i}{K-i} + \sum_{i=0}^K \binom{n-i}{i-1} \binom{n'-1-K+i}{K-i} \\ &= \sum_{i=0}^K \binom{n-i}{i} \binom{n'-K+i}{K-i} + \sum_{i=0}^{K-1} \binom{(n-1)-i}{i} \binom{(n'-1)-(K-1)+i}{(K-1)-i} \end{aligned}$$

which is exactly Eq. (2.13).

To prove Eq. (2.14), we start from a grid of size  $(n+n'+1) \times K$  and are again interested

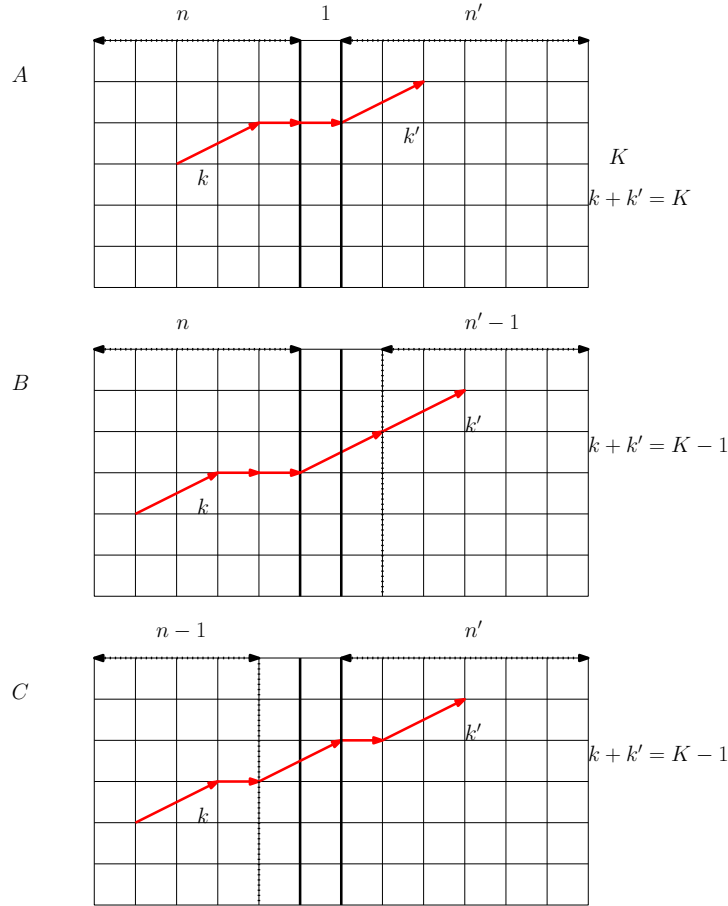


Figure 2.B.2 – Partition of paths according to whether to the move used between  $x = n$  and  $x = n + 1$ . Cases A, B and C correspond to the items listed in the main text.

in the paths starting from the bottom left corner and ending in the upper right corner using only  $(2, 1)$  and  $(1, 0)$  moves. These paths have exactly  $K$  moves of type  $(2, 1)$  and there are  $\binom{n+n'+1-K}{K}$  of them. This time, we partition paths upon the move observed between  $x = n$  and  $x = (n + 1)$ .

The move can be (see also Figure 2.B.2):

- $(1, 0)$ , in which case  $k$  (resp.  $k'$ ) moves of type  $(2, 1)$  are used in the interval  $[1, n]$  (resp.  $[n + 1, n + n' + 1]$ ) such that  $k + k' = K$ ;
- $(2, 1)$  starting from  $x = n$  and therefore ending at  $x = n + 2$ , in which case  $k$  (resp.  $k'$ ) moves of type  $(2, 1)$  are used in the interval  $[1, n]$  (resp.  $[n + 2, n + n' + 1]$ ) such that  $k + k' = K - 1$  (one move  $(2, 1)$  has already been consumed);
- $(2, 1)$  ending at  $x = n + 1$  and therefore starting from  $x = n - 1$  in which case  $k$  (resp.  $k'$ ) moves of type  $(2, 1)$  are used in the interval  $[1, n - 1]$  (resp.  $[n + 1, n + n' + 1]$ ) such that  $k + k' = K - 1$  (one move  $(2, 1)$  has already been consumed);

Wrapping everything together and using the same arguments as before, we get Eq. (2.14).

## 2.C Technical Details of the EM

### 2.C.1 E Step

Given a set of parameters  $\theta^{(h)}$ , we have:

$$\mathbf{X} = (\mathbf{Z}, \mathbf{Y}) \sim \mathcal{N}\left(\mathbf{m}^{(h)} = \begin{pmatrix} \mathbf{m}_{\mathbf{Z}}^{(h)} \\ \mathbf{m}_{\mathbf{Y}}^{(h)} \end{pmatrix}, \Sigma^{(h)} = \begin{pmatrix} \Sigma_{\mathbf{ZZ}}^{(h)} & \Sigma_{\mathbf{YZ}}^{(h)} \\ \Sigma_{\mathbf{ZY}}^{(h)} & \Sigma_{\mathbf{YY}}^{(h)} \end{pmatrix}\right)$$

hence:

$$\begin{aligned} \mathbf{Z} | \mathbf{Y} &\sim \mathcal{N}\left(\mathbf{m}_{\mathbf{Z}|\mathbf{Y}}^{(h)} = \mathbf{m}_{\mathbf{Z}}^{(h)} + \Sigma_{\mathbf{ZY}}^{(h)}(\Sigma_{\mathbf{YY}}^{(h)})^{-1}(\mathbf{Y} - \mathbf{m}_{\mathbf{Y}}^{(h)}), \right. \\ &\quad \left. \Sigma_{\mathbf{Z}|\mathbf{Y}} = \Sigma_{\mathbf{ZZ}}^{(h)} - \Sigma_{\mathbf{ZY}}^{(h)}(\Sigma_{\mathbf{YY}}^{(h)})^{-1}\Sigma_{\mathbf{YZ}}^{(h)}\right) \end{aligned}$$

*Remark 2.C.1.* We can see that this approach forces us to invert  $\Sigma_{\mathbf{YY}}^{(h)}$ , a  $n \times n$  matrix, which is a costly operation, of order  $O(n^3)$ . It also computes the complete matrix  $\Sigma_{\mathbf{Z}|\mathbf{Y}}$  whereas we only need a linear number of its coefficients: conditional variances and covariances of the form  $\text{Cov}[\mathbf{Z}_i; \mathbf{Z}_{\text{pa}(i)} | \mathbf{Y}]$ . Due to the tree structure and to the Gaussian nature of the processes studied, it is possible to compute all the quantities needed in a linear time, using a “forward-backward”-like algorithm (here, “upward-downward”, see [Lartillot \(2014\)](#) for a similar algorithm.). The upward step is similar to the pruning algorithm described in [Felsenstein \(2004, chap. 23\)](#). See also [Ho & Ané \(2013a\)](#) for an algorithm linear in the number of iterations.

### 2.C.2 Complete Likelihood Computation

Using the incomplete data model described in section 2.2.2, we can write:

$$p_{\theta}(\mathbf{X}) = p_{\theta}(X_1) \prod_{j=2}^{m+n} p_{\theta}(X_j | X_{\text{pa}(j)})$$

Taking the expectation, we get for the BM:

$$\begin{aligned} -2\mathbb{E}[\log p_{\theta}(\mathbf{X}) | \mathbf{Y}] &= A + \log \gamma^2 + \frac{1}{\gamma^2} \left( \text{Var}[X_1 | \mathbf{Y}] + (\mathbb{E}[X_1 | \mathbf{Y}] - \mu)^2 \right) \\ &\quad + (m+n-1) \log \sigma^2 + \frac{1}{\sigma^2} \sum_{j=2}^{m+n} \ell_j^{-1} \text{Var}[X_j - X_{\text{pa}(j)} | \mathbf{Y}] \\ &\quad + \frac{1}{\sigma^2} \sum_{j=2}^{m+n} C_j^{BM}(\Delta) \end{aligned} \tag{2.16}$$

and, for the OUsun:

$$\begin{aligned} -2\mathbb{E}[\log p_{\theta}(\mathbf{X}) | \mathbf{Y}] &= B + \sum_{j=2}^{m+n} \log c_j(\alpha) + \frac{1}{\gamma^2} \text{Var}[X_1 | \mathbf{Y}] + (m+n) \log \gamma^2 \\ &\quad + \frac{1}{\gamma^2} \sum_{j=2}^{m+n} c_j(\alpha)^{-1} \text{Var}[X_j - X_{\text{pa}(j)} e_j | \mathbf{Y}] \\ &\quad + \frac{1}{\gamma^2} (\mathbb{E}[X_1 | \mathbf{Y}] - \beta_0)^2 + \frac{1}{\gamma^2} \sum_{j=2}^{m+n} C_j^{OU}(\alpha, \Delta) \end{aligned} \tag{2.17}$$

where  $A$  and  $B$  are constants, and for each node  $j$ ,  $j \in \llbracket 2, m+n \rrbracket$ , we define an actualization factor  $c_j(\alpha) = 1 - e_j(\alpha)^2$ , with  $e_j(\alpha) = e^{-\alpha \ell_j}$ , and  $C_j^{BM}$  and  $C_j^{OU}$  are costs associated with branch  $b_j$ :

$$\begin{cases} C_j^{BM}(\Delta) = \frac{1}{\ell_j} \left( \mathbb{E}[X_j | \mathbf{Y}] - \mathbb{E}[X_{\text{pa}(j)} | \mathbf{Y}] - \Delta_j \right)^2 \\ C_j^{OU}(\alpha, \Delta) = \frac{1}{c_j(\alpha)} \left( \mathbb{E}[X_j | \mathbf{Y}] - e_j \mathbb{E}[X_{\text{pa}(j)} | \mathbf{Y}] - \beta_j (1 - e_j) \right)^2 \end{cases}$$

### 2.C.3 M step

Assuming that  $p_{\theta^{(h)}}(\mathbf{Z} | \mathbf{Y})$  is known, we need to compute  $\theta^{(h+1)}$  by maximizing  $\mathbb{E}_{\theta^{(h)}}[\log p_{\theta}(\mathbf{Z}, \mathbf{Y}) | \mathbf{Y}]$ . We have to deal with parameters of different nature, discrete or continuous. For a given vector  $\Delta^{(h+1)}$  of  $K$  non-zero shifts, we can exhibit closed formulas for  $\mu^{(h+1)}$ ,  $\sigma^{(h+1)}$  and  $\gamma^{(h+1)}$ , for the BM:

$$\begin{cases} \mu^{(h+1)} = \mathbb{E}^{(h)}[Z_1 | \mathbf{Y}] \\ \gamma^{2(h+1)} = \mathbb{V}\text{ar}^{(h)}[Z_1 | \mathbf{Y}] \\ \sigma^{2(h+1)} = \frac{1}{m+n-1} \left[ \sum_{j=2}^{m+n} \ell_j^{-1} \mathbb{V}\text{ar}^{(h)}[X_j - X_{\text{pa}(j)} | \mathbf{Y}] + C_j^{BM}(\Delta^{(h+1)}) \right] \end{cases}$$

and, for the OUsun:

$$\begin{aligned} (m+n)\gamma^{2(h+1)} &= \mathbb{V}\text{ar}^{(h)}[X_1 | \mathbf{Y}] + \sum_{j=2}^{m+n} c_j(\alpha)^{-1} \mathbb{V}\text{ar}^{(h)}[X_j - X_{\text{pa}(j)} e^{-\alpha \ell_j} | \mathbf{Y}] \\ &\quad + \left( \mathbb{E}^{(h)}[X_1 | \mathbf{Y}] - \beta_0^{(h+1)} \right)^2 + \sum_{j=2}^{m+n} C_j^{OU}(\alpha^{(h)}, \Delta^{(h+1)}) \end{aligned}$$

There is no such closed formula for  $\alpha$ . In the implementation we propose, this parameter is actualized after all the others, by doing a numerical maximization of the objective function.

Finally, the vector  $\Delta^{(h+1)}$  of  $K$  non-zero shifts can be chosen in an optimal way for the BM thanks to a simple algorithm explained below. In the OUsun case, we can only increase the objective function, and not maximize it. In that case, we hence use a Generalized EM algorithm (GEM, see [Dempster et al., 1977](#)).

**Optimal Shift Location for the BM.** We want to minimize the sum of costs:

$$C^{BM}(\Delta) = \sum_{j=2}^{m+n} C_j^{BM}(\Delta)$$

Each cost is associated to a branch  $b_j$ ,  $j \in \llbracket 2, m+n \rrbracket$ , and, when the sum is minimal,  $C_j^{BM}(\Delta)$  can only take two values:

$$\begin{cases} \tilde{C}_j^{BM} = \ell_j^{-1} \left( \mathbb{E}^{(h)}[X_j | \mathbf{Y}] - \mathbb{E}^{(h)}[X_{\text{pa}(j)} | \mathbf{Y}] \right)^2 & \text{if no shift on branch } b_j \\ 0 & \text{if one shift on branch } b_j \end{cases}$$

The sum can hence be minimized in the following way:

1. Compute  $\tilde{C}_j^{BM}$  for all  $j \in \llbracket 2, m+n \rrbracket$ .
2. Find the  $K$  highest costs  $(j_1, \dots, j_K) \in \llbracket 2, m+n \rrbracket^K$ .
3. Set  $\Delta_{j_k}^{(h+1)} = \mathbb{E}^{(h)}[X_{j_k} | \mathbf{Y}] - \mathbb{E}^{(h)}[X_{\text{pa}(j_k)} | \mathbf{Y}]$  for all  $k \in \llbracket 1, K \rrbracket$ , and  $\Delta_j^{(h+1)} = 0$  if  $j \notin \{j_1, \dots, j_K\}$ .

This exact and fast algorithm works for the BM because all the costs are independent. Note that it would work for any Levy Process without memory, such as those proposed in Landis et al. (2013) to model evolution of quantitative traits.

**GM Step for Shifts Locations for the OU.** With  $\alpha^{(h)}$  fixed, we want to minimize the sum of costs:

$$C^{OU}(\alpha^{(h)}, \Delta) = \left( \mathbb{E}^{(h)}[X_1 | \mathbf{Y}] - \beta_0 \right)^2 + \sum_{j=2}^{m+n} C_j^{OU}(\alpha^{(h)}, \Delta)$$

The previous algorithm does not work, because the costs are not independent. Solving the problem exactly would require to visit all the possible configurations, and the complexity would be too high, of order  $O\left(\binom{m+n}{K}\right) = O(n^K)$ . To reduce the execution time of the algorithm, we use heuristics to lower, if not minimize, the sum of costs. We use the following formulation:

$$C^{OU}(\alpha^{(h)}, \Delta) = \|\mathbf{F}^{(h)} - \mathbf{A}^{(h)}\mathbf{U}\Delta\|^2$$

where  $\mathbf{U}$  the complete tree matrix given in subsection 2.2.3,  $\mathbf{A}^{(h)}$  a diagonal matrix depending on  $\alpha^{(h)}$   $\mathbf{A}^{(h)} = \text{Diag}\left(1, \sqrt{\frac{1-e^{-\alpha^{(h)}\ell_j}}{1+e^{-\alpha^{(h)}\ell_j}}}; 2 \leq j \leq m+n\right)$ , and  $\mathbf{F}^{(h)}$  a vector of expectations, with  $F_1^{(h)} = \mathbb{E}^{(h)}[X_1 | \mathbf{Y}]$ , and, for  $2 \leq j \leq m+n$ ,

$$F_j^{(h)} = \left(1 - e^{-2\alpha^{(h)}\ell_j}\right)^{-1/2} \left( \mathbb{E}^{(h)}[X_j | \mathbf{Y}] - \mathbb{E}^{(h)}[X_{\text{pa}(j)} | \mathbf{Y}] e^{-\alpha^{(h)}\ell_j} \right)$$

We can then use a Lasso algorithm to impose sparsity constraints on  $\Delta$ . If  $\Delta_{-1}$  is the vector of shifts without the initial value (intercept), then a Lasso estimator is given by, for  $\lambda \geq 0$ :

$$\hat{\Delta}_\lambda = \underset{\Delta}{\text{argmin}} \left\{ \|\mathbf{F}^{(h)} - \mathbf{A}^{(h)}\mathbf{U}\Delta\|^2 + \lambda |\Delta_{-1}|_1 \right\}$$

The estimated vectors  $\hat{\Delta}_\lambda$  have a support that is sparser when  $\lambda$  becomes higher. One then only need to find the right penalty factor  $\lambda$  that ensure that the support has exactly  $K$  non zero coordinates, plus the initial value. We ensure that the  $K$  shifts are allocated in a parsimonious way by checking their linear independence, using proposition 2.3.6.

An other method is to take the previous solution  $\Delta^{(h)}$ , and test all the configurations where only one shift has moved, and take the best one. In both methods, one also has to ensure that the objective function is increased by the new choice of shifts, so that the GEM algorithm works correctly. This step is generally the longest one in one iteration of the EM.

### 2.C.4 Initialization

Initialization is always a crucial step when using an EM algorithm. The vector of shifts  $\Delta$  is initialized thanks to a Lasso procedure. To do that, we use the linear formulation (2.4) or (2.6) of the main text, and we calibrate the penalty so that the initialization vector has a non zero first coordinate (initial value), and  $K$  other non-zero coordinates. The variance-covariance matrix is initialized with defaults parameters, and is taken into account thanks to a Cholesky decomposition.

We also initialize the selection strength  $\alpha$ . We use the following property: if  $Y_i$  and  $Y_j$  are two tips in the same group, then, under an OUsun,  $\mathbb{E}[(Y_i - Y_j)^2] = 2\gamma^2(1 - e^{-\alpha d_{ij}})$ . Using regression techniques, we can get an initial estimation of  $\alpha$  and  $\gamma^2$  from all these couples. In practice, we first initialize the position of the shifts, and then use only pairs of tips from the same estimated group. Then, as the groups are only approximated, some of the selected pairs  $(Y_i, Y_j)$  might not share the same expectation, and we use a robust regression to get more accurate initial estimates.

## 2.D Optimal Shift Location with Fixed Root

The algorithm for optimal shift location we described above (see Section 2.C.3) actually only works for a BM with a random root. When the inference is done conditionally to the root value, then an extra step is needed. Indeed, the objective function is then:

$$\begin{aligned} -2\mathbb{E}[\log p_{\theta}(\mathbf{X}) \mid \mathbf{Y}, X_1] &= A + (m+n-1)\log \sigma^2 + \frac{1}{\sigma^2} \sum_{j=2}^{m+n} C_j^{BM}(\boldsymbol{\tau}, \boldsymbol{\delta}) \\ &+ \frac{1}{\sigma^2} \sum_{j=2}^{m+n} \ell_j^{-1} \text{Var} [X_j - X_{\text{pa}(j)} \mid \mathbf{Y}, X_1], \end{aligned}$$

with

$$C_j^{BM}(\Delta) = \frac{1}{\ell_j} \left( \mathbb{E}[X_j \mid \mathbf{Y}, X_1] - \mathbb{E}[X_{\text{pa}(j)} \mid \mathbf{Y}, X_1] - \Delta_j \right)^2$$

And, for the optimal shift location, we want to minimize the sum of costs:

$$C^{BM}(\Delta) = \sum_{j=2}^{m+n} C_j^{BM}(\Delta).$$

Each cost is associated to a branch  $b_j$ ,  $j \in \llbracket 2, m+n \rrbracket$ . Denote by  $m_j = \mathbb{E}[X_j \mid \mathbf{Y}, X_1]$  for any  $j \geq 2$ . As previously, when the sum is minimal, for a node  $j$  that is *not a direct descendant of the root*, a cost can only take two values:

$$\begin{cases} \tilde{C}_j^{BM}(\Delta) = \ell_j^{-1} (m_j - m_{\text{pa}(j)})^2 & \text{if no shift on branch } b_j \\ 0 & \text{if one shift on branch } b_j \end{cases}$$

The problem arise for branches that are descendants of the root, that we denote by  $(r_1, \dots, r_l)$ . Indeed, their costs are, for  $1 \leq a \leq l$ :

$$C_{r_a}^{BM}(\Delta) = \frac{1}{\ell_{r_a}} (m_{r_a} - \mu - \Delta_{r_a})^2$$



and the unknown parameter  $\mu$  appears in several costs, so that the joint maximization in  $\mu$  and  $\Delta$  is not straightforward.

First, let's assume that the root has only two descendants. Then, we can solve the problem by formally “un-rooting” the tree, replacing the two root branches by a single branch, with a new ad-hoc cost. Indeed, if there are no shifts on these two root branches, we get:

$$\hat{\mu} = \frac{\ell_{r_1}^{-1} m_{r_1} + \ell_{r_2}^{-1} m_{r_2}}{\ell_{r_1}^{-1} + \ell_{r_2}^{-1}} \quad \text{and} \quad \tilde{C}_{r_{1,2}}^{BM} = \tilde{C}_{r_1}^{BM} + \tilde{C}_{r_2}^{BM} = (\ell_{r_1} + \ell_{r_2})^{-1} (m_{r_2} - m_{r_1})^2.$$

Then, we can use the previously described algorithm to spot the shifts, treating this fictive branch with its new cost as all the others. If a shift has to be placed on it, then it will cancel the cost of the fictive branch. Its position is not identifiable, and we can choose to put it on branch  $r_1$  (then  $\hat{\mu} = m_{r_2}$  and  $\hat{\delta} = m_{r_1} - m_{r_2}$ ), or on  $r_2$  (then  $\hat{\mu} = m_{r_1}$  and  $\hat{\delta} = m_{r_2} - m_{r_1}$ ) indistinctly.

Now, let's assume that the root has three or more descendants. The trick used above cannot be adapted, and we need to resort to a heuristic to minimize the sum of costs. Instead of optimizing it jointly in  $\Delta$  and  $\mu$ , we do a two-steps minimization:

1. Fix  $\mu$  to its previous value  $\mu^{(h-1)}$ , and optimize the sum in the position and values  $\Delta^{(h)}$  of the shifts.
2. Fix  $\Delta^{(h)}$ , and optimize the costs in  $\mu$ , taking:

$$\mu^{(h)} = \frac{\sum_{1 \leq a \leq l} \ell_{r_a}^{-1} (m_{r_a}^{(h)} - \Delta_{r_a}^{(h)})}{\sum_{1 \leq a \leq l} \ell_{r_a}^{-1}}.$$

Because this optimization happens inside the EM loop, it is sufficient to just increase the objective function (so that we get a Generalized EM).

## 2.E Proof of Proposition 2.4.1 for Model Selection

We prove the proposition using the linear formulation  $\mathbf{s} + \gamma \mathbf{E}$ , with  $\mathbf{E} \sim \mathcal{N}(0, \mathbf{V})$ , as derived in the main text for the OUsun (with  $\mathbf{s} = \mathbf{TW}(\alpha)\Delta$ ,  $\gamma^2 = \sigma^2/(2\alpha)$ , and  $V_{ij} = e^{-\alpha d_{ij}}$ , see Formula (2.5)). Note that this framework also holds for the BM with a fixed root (with  $\mathbf{s} = \mathbf{T}\Delta$ ,  $\gamma = \sigma$ , and  $V_{ij} = t_{ij}$ , see Formula (2.4)).

We first handle the case where there are no correlations ( $\mathbf{V}$  diagonal), and then use a Cholesky decomposition to handle the general case. Note that the case  $\mathbf{V}$  diagonal can be seen as the limit of the OUsun when  $\alpha = +\infty$ , or as a BM on a star tree.

### Case $\mathbf{V}$ diagonal

In the iid case, we just need to check the conditions of theorem 2.4.1. This paragraph is highly inspired by the derivation of the bound for the detection of non-zero mean components exposed in Baraud et al. (2009) (sub-section 5.2). Assume that  $D_\eta = K_\eta + 1 \leq p \leq n - 7$  for all  $\eta \in \mathcal{M}$ . The estimator is defined by  $\hat{\mathbf{s}}_{\hat{K}}$ , with:

$$\hat{K} = \underset{0 \leq K \leq p-1}{\operatorname{argmin}} \|\mathbf{Y} - \hat{\mathbf{s}}_K\|_{\mathbf{V}^{-1}}^2 \left( 1 + \frac{\operatorname{pen}_{A, \mathcal{L}}(K)}{n - K - 1} \right)$$

From the definition of  $\hat{\mathbf{s}}_K$ , and as the penalty depends on the model only through its number of shifts, we get that  $\hat{\mathbf{s}}_{\hat{K}} = \hat{\mathbf{s}}_{\hat{\eta}}$  the minimizer of the criterion of theorem 2.4.1 (with  $N_\eta = n - D_\eta = n - K_\eta - 1$ ). We then have:

$$\Omega' = \sum_{\eta \in \mathcal{M}} (D_\eta + 1) e^{-L_\eta} = \sum_{K=0}^{p-1} |\mathcal{S}_K^{PI}| (K+2) e^{-L_K}$$

With the weights  $L_K$  defined in Equation (2.12) of the proposition, we get:

$$\Omega' = \sum_{K=0}^{p-1} \frac{1}{K+2} \leq \log(p) \leq \log(n)$$

As:

$$\begin{aligned} L_K &\leq \log \binom{n+m-1}{K} + 2 \log(K+2) \leq K \log(n+m-1) + 2 \log(K+2) \\ &\leq K \log(2n-2) + 2(K+1) \\ &\leq (K+1)(2 + \log(2) + \log(n)) \\ &\leq p(2 + \log(2) + \log(n)) \end{aligned}$$

if  $p \leq \min\left(\frac{\kappa n}{2 + \log(2) + \log(n)}, n-7\right)$ , then  $\max(L_\eta, D_\eta) \leq \kappa n$  for any  $\eta \in \mathcal{M}$ , and we get the announced bound from the second proposition of theorem 2.4.1.

*Case V not diagonal*

Using a Cholesky decomposition, we can find a lower triangular matrix  $\mathbf{L}$  such that  $\mathbf{V} = \mathbf{L}\mathbf{L}^T$ . Then, denoting  $\mathbf{Y}' = \mathbf{L}^{-1}\mathbf{Y}$ ,  $\mathbf{s}' = \mathbf{L}^{-1}\mathbf{s}$ , and  $\mathbf{E}' = \mathbf{L}^{-1}\mathbf{E}$ , we have  $\mathbf{Y}' = \mathbf{s}' + \gamma\mathbf{E}'$ , with  $\mathbf{E}' \sim \mathcal{N}(0, \mathbf{I}_n)$ , and we can apply theorem 2.4.1 as above. As we changed the metric, the estimators are projections on the linear spaces  $S'_\eta = \mathbf{L}^{-1}S_\eta$  for  $\eta \in \mathcal{M}$ , and we have:

$$\begin{aligned} \hat{\mathbf{s}}'_\eta &= \text{Proj}_{S'_\eta} \mathbf{Y}' = \underset{\mathbf{a}' \in S'_\eta}{\text{argmin}} \|\mathbf{Y}' - \mathbf{a}'\|^2 = \underset{\mathbf{a}' \in S'_\eta}{\text{argmin}} \|\mathbf{L}^{-1}\mathbf{Y} - \mathbf{L}^{-1}\mathbf{L}\mathbf{a}'\|^2 \\ &= \underset{\mathbf{a}' \in S'_\eta}{\text{argmin}} \|\mathbf{Y} - \mathbf{L}\mathbf{a}'\|_{\mathbf{V}^{-1}}^2 = \mathbf{L}^{-1} \underset{\mathbf{a} \in S_\eta}{\text{argmin}} \|\mathbf{Y} - \mathbf{a}\|_{\mathbf{V}^{-1}}^2 = \mathbf{L}^{-1} \hat{\mathbf{s}}_\eta \end{aligned}$$

So  $\|\mathbf{s} - \hat{\mathbf{s}}_{\hat{\eta}}\|_{\mathbf{V}^{-1}}^2 = \|\mathbf{s}' - \hat{\mathbf{s}}'_{\hat{\eta}}\|^2$  and  $\|\mathbf{Y} - \hat{\mathbf{s}}_{\hat{\eta}}\|_{\mathbf{V}^{-1}}^2 = \|\mathbf{Y}' - \hat{\mathbf{s}}'_{\hat{\eta}}\|^2$ , and, as the form of the penalty does not depend on  $V$ , by minimizing:

$$\text{Crit}_{LS}(K) = \|\mathbf{Y}' - \hat{\mathbf{s}}'_K\|^2 \left(1 + \frac{\text{pen}_{A,\mathcal{L}}(K)}{n-K-1}\right) = \|\mathbf{Y} - \hat{\mathbf{s}}_K\|_{\mathbf{V}^{-1}}^2 \left(1 + \frac{\text{pen}_{A,\mathcal{L}}(K)}{n-K-1}\right)$$

we get the announced bound on  $\mathbb{E} \left[ \frac{\|\mathbf{s} - \hat{\mathbf{s}}_{\hat{K}}\|_{\mathbf{V}^{-1}}^2}{\gamma^2} \right] = \mathbb{E} \left[ \frac{\|\mathbf{s}' - \hat{\mathbf{s}}'_{\hat{K}}\|^2}{\gamma^2} \right]$ .

## 2.F Supplementary Figures

### 2.F.1 Simulation Study: Sensitivity and False Positive Rate

**Definition of the Scores.** We denote by  $TP$  the number of True Positives, i.e. the predicted edges on which a shift actually occurred, and  $FP$  the number of False Positives. The sensitivity  $\frac{TP}{K_t}$  is the proportion of well predicted shifts among all shifts to be

predicted, and the False Positive Rate (FPR)  $\frac{FP}{n+m-K_t}$  is the proportion of false positive among all edges with no shifts.

Note that here, due to the possible lack of identifiability, the position of the shifts on the tree is not well defined, as a shift can be on a particular edge for one of the equivalent solutions, but not on the others (see Section 2.3.1). These two scores are hence ill defined for our problem. To avoid such problems, we restrict ourselves to the 92% of unambiguous configurations that occurred during the simulations.

**Interpretation of Results.** Figure 2.F.1 shows that the FPR are systematically worse when using the true number of shifts, indicating that the additional shifts found when compared to the selected number are misplaced. The FPR remains very low, as only a small number of shifts is to be found. Unsurprisingly, the Sensitivity is on the contrary improved when taking the real number of shifts, as shown Figure 2.F.2. In addition, the sensitivity is highly degraded when  $\alpha$  is small or  $\gamma^2$  is high, but does not exhibit a clear tendency in the real number of shifts, and the knowledge of the true value of  $\alpha$  does not seem to matter.

## 2.F.2 Simulation Study: Complementary Analysis

To complete the analysis conducted in the main text, Figure 2.F.3 presents the variations of the log-likelihood, phylogenetic half-life, and root variance when  $\alpha$  is estimated, and the number of shifts is known or estimated. We can see here that the likelihood is slightly higher when the number of shifts is fixed, which is coherent with the behavior of our model selection procedure, that tends to under-estimate the true number of shifts (see Figure 2.5.3 of the main text). We also note that knowing the true number of shifts has not a great influence on the estimation of  $\alpha$  and  $\gamma$ , making the later worse, if anything.

Figure 2.F.4 shows the variations of the estimations of  $\beta_1$ , the number of shifts and the ARI when the number of shifts is estimated, and  $\alpha$  is fixed or estimated. This confirms our earlier statement, that not knowing  $\alpha$  with precision does not have a great impact on the model selection procedure (see also Cressler et al., 2015).

## 2.F.3 Chelonia Dataset: Comparison of Inferred Shift Locations

On Figure 2.F.5, we present and compare the shift locations found by our method, and methods bayou and SURFACE. The differences found are explored deeper in the main text (Section 2.6.4).

**Details on the Methods.** In this paragraph, we give more details on the 4 already existing methods that we compared to ours in the dataset. We first fitted an  $OU_{habitat}$  model with fixed regimes as in Jaffe et al. (2011), using the R package OUwie (Beaulieu et al., 2012). We tested all of the 48 possible ways of allocating internal nodes, and took the solution with the highest likelihood. Using the package bayou (Uyeda & Harmon, 2014), we reproduced the Bayesian analysis of the data, using two independent chains of 500000 generations each, discarding the first 150000 generations as burning. We assigned the priors that were used in the original study on the parameters, namely:  $P(\alpha) \sim \text{LogNormal}(\ln \mu = -5, \ln \sigma = 2.5)$ ,  $P(\sigma^2) \sim \text{LogNormal}(\ln \mu = 0, \ln \sigma = 2)$ ,  $P(\beta_i) \sim \text{Normal}(\mu = 3.5, \sigma = 1.5)$ ,  $P(K) \sim \text{Conditional Poisson}(\lambda = 15, K_{max} = 113)$ . The computations took around 2.3 hours of CPU time. We also ran the stepwise-AIC method

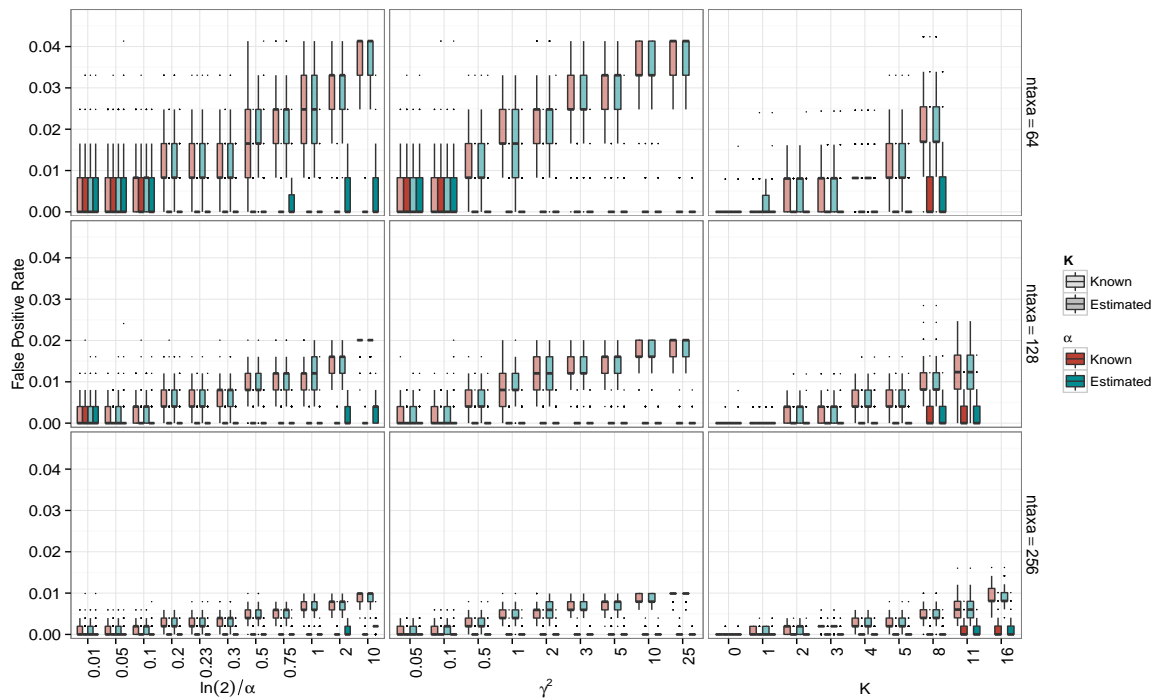


Figure 2.F.1 – False Positive Rate computed for the different configurations. Note the  $\gamma$  scale, that only goes to 0.05.

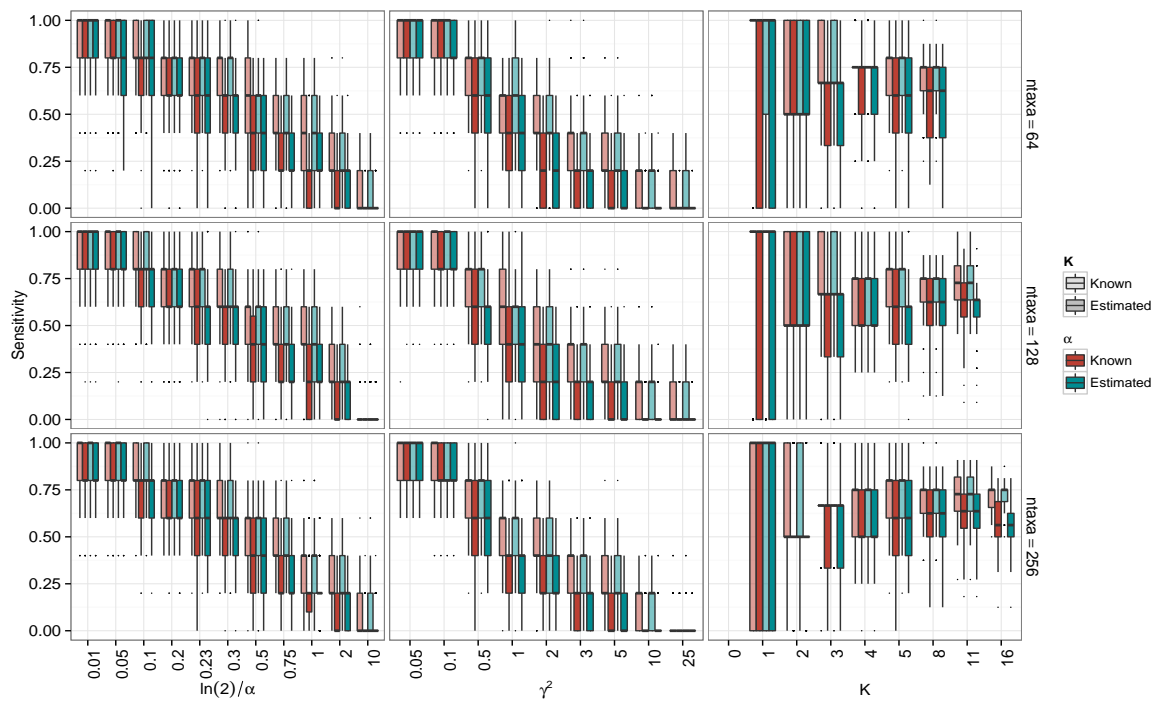


Figure 2.F.2 – Sensitivity computed for the different configurations, with box-plots over the repetitions.

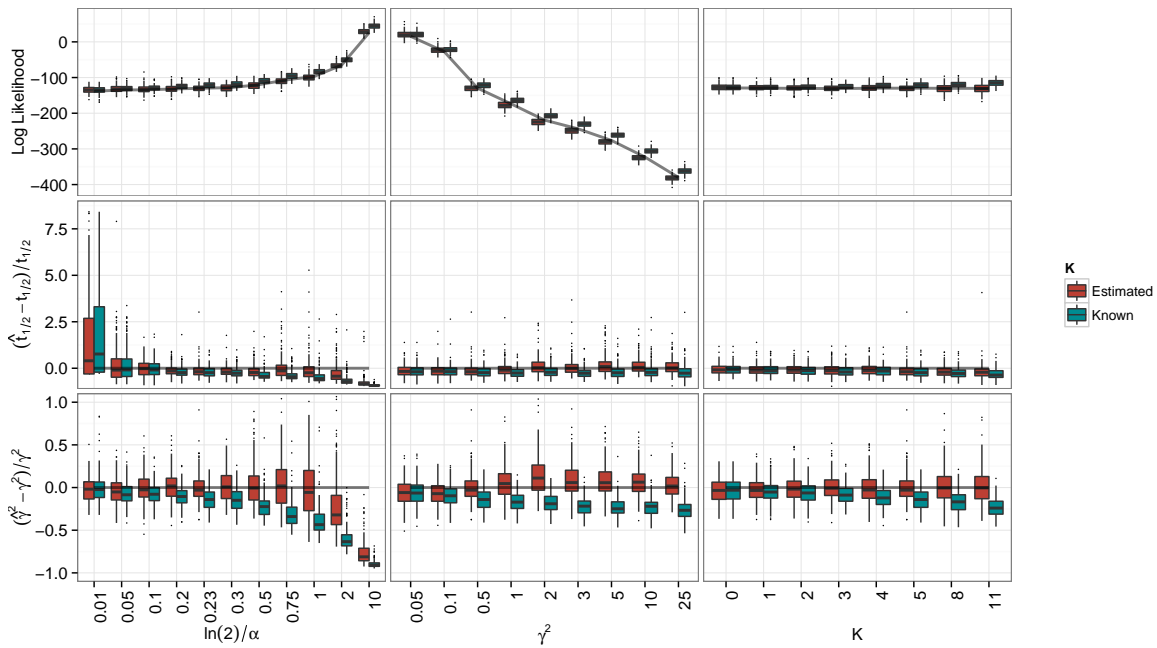


Figure 2.F.3 – Box plots over the 200 repetitions of each set of parameters, for the log-likelihood (top), phylogenetic half-life (middle) and root variance (bottom) with  $\alpha$  estimated, and  $K$  fixed or estimated, on a tree with 128 taxa.

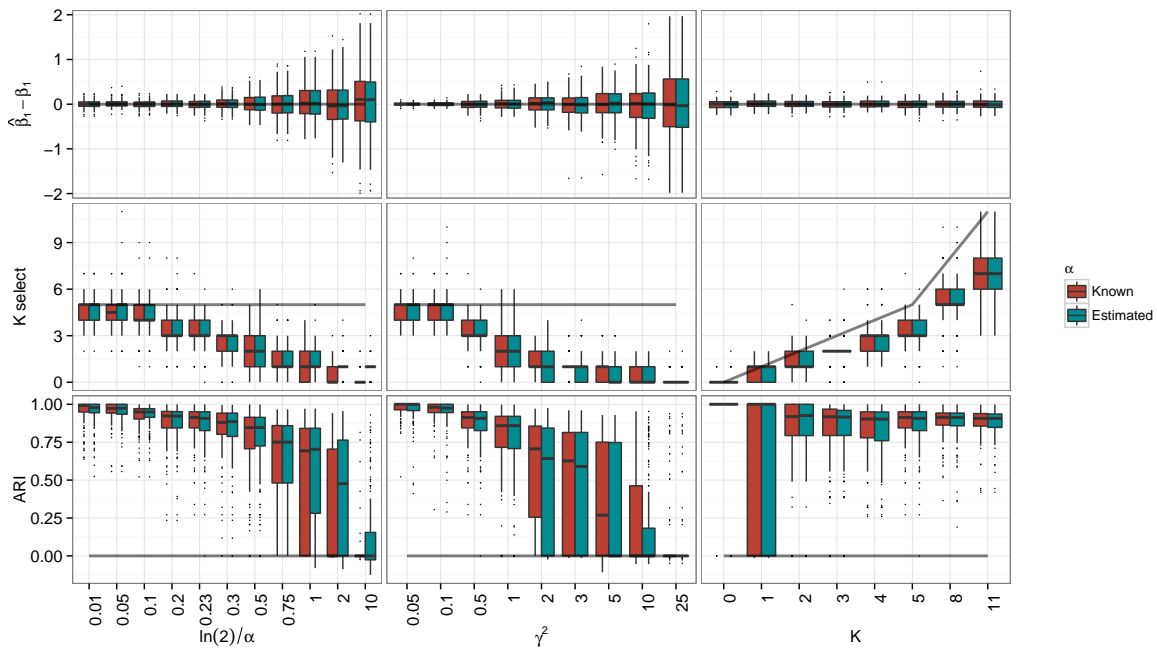


Figure 2.F.4 – Same for  $\beta_1$  (top), the number of shifts (middle) and ARI (bottom), with  $K$  estimated, and  $\alpha$  fixed or estimated.

For better legibility, strips with  $t_{1/2}$ ,  $\gamma^2$  and  $\beta_1$  on these two figures were re-scaled, omitting some outliers (respectively, 0.21%, 0.27% and 0.27% of points are omitted). The whisker of the first box for  $t_{1/2}$  goes up to 7.5.

SURFACE, that relies on a forward-backward procedure. This took around 11 hours of CPU time. Finally, we ran the function `OUshifts` from the package `phylolm`, that uses a modified BIC criterion and a heuristic stepwise procedure to detect shifts (Ho & Ané, 2014). Thanks to an efficient linear algorithm, detailed in Ho & Ané (2013a), this function is pretty fast, taking only about 8 minutes of CPU time. Note that the model used in these last three methods (`bayou`, SURFACE and `OUshifts`) are slightly different from ours, as they assume that the root is fixed to the ancestral optimum state, and not drawn from its stationary distribution.

**Note on Computation Times.** We found that the running time for our method was similar to the running time of previous algorithms (see Table 2.1 in the main text). However, our computations can be highly parallelized, as each run for a fixed number of shift is independent from the others. For instance, in the previous example, the computation time could be divided by 6, each estimation for a fixed  $\alpha$  running on a different core. On the contrary, the SURFACE method cannot be parallelized at all, and only independent chains can be parallelized for Bayesian methods, so that the computation time can only be divided by 2 in our example.

## 2.G Practical Implementation

The statistical method described here was implemented on the statistical software R (R Core Team, 2017), and the code is freely available on GitHub (<https://github.com/pbastide/Phylogenetic-EM>). Phylogenetic trees were handled thanks to the package `ape` (Paradis et al., 2004). Packages `TreeSim` (Stadler, 2011), `robustbase` (Rousseeuw et al., 2014) and `quadrupen` (Grandvalet et al., 2012) were used, respectively, for random tree generation, robust regression and Lasso regression. The penalty described in proposition 2.4.1 is implemented in package `LINselect` (Baraud et al., 2013).

Parallelization was achieved thanks to R packages `foreach` (Weston, 2014b) and `doParallel` (Weston, 2014a).

Package `mclust` (Fraley et al., 2012) was used for ARI computations. Plots were made thanks to packages `ggplot2` (Wickham, 2009) and `reshape2` (Wickham, 2007).

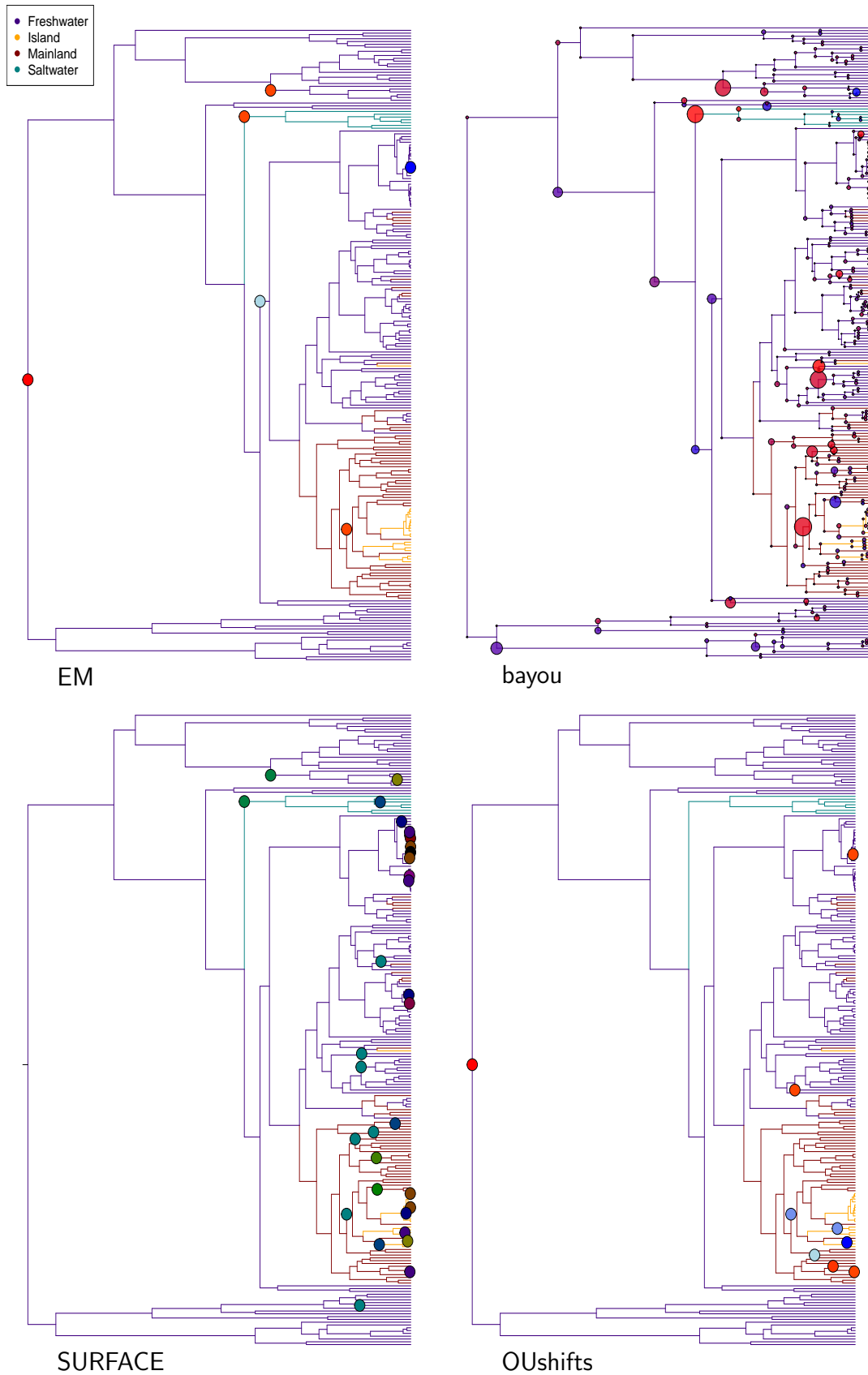


Figure 2.F.5 – Solutions found by the EM (top left), bayou (top right), SURFACE (bottom left) and OUshifts (bottom right). The branch coloring represents the habitats. For the EM, bayou and OUshifts, the shifts coloring represents their values, from blue (negative) to red (positive). For bayou, the size of the circles are proportional to their posterior probability. For SURFACE, the 13 colors of the shifts represent the regimes.



# Chapter 3

## Shift Detection for Multivariate Processes

### Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>120</b>
3.1.1	Motivation	120
3.1.2	State of the Art	121
3.1.3	Scope of the Article	122
<b>3.2</b>	<b>Model</b>	<b>122</b>
3.2.1	Trait Evolution on a Tree	122
3.2.2	Simplifying Assumptions	123
3.2.3	Identifiability Issues	125
3.2.4	Re-scaling of the Tree	125
3.2.5	Statistical Inference	126
3.2.6	Implementation	127
<b>3.3</b>	<b>pPCA and Shifts</b>	<b>128</b>
3.3.1	pPCA is biased in the presence of shifts	128
3.3.2	Illustration: a simple example	128
<b>3.4</b>	<b>Simulations Studies</b>	<b>129</b>
3.4.1	Experimental Design	129
3.4.2	Results	132
<b>3.5</b>	<b>Examples</b>	<b>136</b>
3.5.1	New World Monkeys	137
3.5.2	Lizards	137
3.5.3	Comments	139
<b>3.6</b>	<b>Discussion</b>	<b>140</b>
3.6.1	Interpretation Issues	141
3.6.2	Noncausal Correlations	141
3.6.3	Nature of the jumps	142
<b>Appendices</b>		<b>143</b>
<b>3.A</b>	<b>PCA: Mathematical Derivations</b>	<b>143</b>
<b>3.B</b>	<b>PhylogeneticEM case study: New World Monkeys</b>	<b>143</b>
3.B.1	Loading and Plotting the data	143
3.B.2	Analyzing the data	144
3.B.3	Plotting Equivalent Solutions	145
<b>3.C</b>	<b>EM Inference</b>	<b>146</b>
3.C.1	M step	147
3.C.2	E step	148
3.C.3	EM Initialization	151
3.C.4	Grid on $\alpha$	152
<b>3.D</b>	<b>Simulations Appendices</b>	<b>153</b>
3.D.1	Kullback-Leibler Divergences	153
3.D.2	Note on the ARI (Hubert & Arabie, 1985)	155
3.D.3	Supplementary Figures	156

---

## Foreword

This chapter has been submitted under the title *Inference of Adaptive Shifts for Multivariate Correlated Traits* to a Biology oriented journal. It is available as a preprint ([Bastide et al., 2017a](#)).

**Abstract.** To study the evolution of several quantitative traits, the classical phylogenetic comparative framework consists of a multivariate random process running along the branches of a phylogenetic tree. The Ornstein-Uhlenbeck (OU) process is sometimes preferred to the simple Brownian Motion (BM) as it models stabilizing selection toward an optimum. The optimum for each trait is likely to be changing over the long periods of time spanned by large modern phylogenies. Our goal is to automatically detect the position of these shifts on a phylogenetic tree, while accounting for correlations between traits, which might exist because of structural or evolutionary constraints. We show that, in the presence shifts, phylogenetic Principal Component Analysis (pPCA) fails to decorrelate traits efficiently, so that any method aiming at finding shift needs to deal with correlation simultaneously. We introduce here a simplification of the full multivariate OU model, named scalar OU (scOU), which allows for noncausal correlations and is still computationally tractable. We extend the equivalence between the OU and a BM on a re-scaled tree to our multivariate framework. We describe an Expectation Maximization algorithm that allows for a maximum likelihood estimation of the shift positions, associated with a new model selection criterion, accounting for the identifiability issues for the shift localization on the tree. The method, freely available as an R-package ([PhylogeneticEM](#)) is fast, and can deal with missing values. We demonstrate its efficiency and accuracy compared to another state-of-the-art method ( [\$\ell\$ 1ou](#)) on a wide range of simulated scenarios, and use this new framework to re-analyze recently gathered datasets on New World Monkeys and *Anolis* lizards.

(Keywords: Ornstein-Uhlenbeck, Change-point detection, Adaptive evolution, Phylogeny, Model selection, [PhylogeneticEM](#))

## 3.1 Introduction

### 3.1.1 Motivation

A major goal of comparative and evolutionary biology is to decipher the past evolutionary mechanisms that shaped the present day diversity. Taking advantage of the increasing amount of molecular data made available by powerful sequencing techniques, sophisticated mathematical models have made it possible to infer reliable phylogenetic trees for ever growing groups of taxa (see e.g. [Meredith et al., 2011](#); [Jetz et al., 2012](#)). Models of phenotypic evolution for such large families need to cope with the heterogeneity of observed traits across the species tree. One source of heterogeneity is the mechanism of “evolution by jumps” as hypothesized by [Simpson \(1944\)](#). It states that there exists an adaptive landscape shaping the evolution of functional traits, and that this landscape might shift, sometimes in a dramatic fashion, in response to environmental changes such as migration, or colonization of a new ecological niche. Such shifts, like the one observed in the brain shape and size of New World Monkeys in association with dietary and locomotion changes ([Aristide et al., 2015, 2016](#)), need to be explicitly accounted for in models of phenotypic evolution.

To detect such adaptive shifts, we must cope with two constraints: species do not evolve independently (Felsenstein, 1985) and adaptive evolution is an intrinsically multivariate phenomenon. The first constraint arises from the shared evolutionary history of species, usually represented as a phylogenetic tree. It means that traits observed on closely related taxa are on average more similar than traits observed on distantly related species. The second constraint results from natural selection acting on many traits at once. Functional traits are indeed often interdependent, either because they are regulated by the same portions of the genetic architecture or because they are functionally constrained (e.g. limb bones lengths in Greater Antillean *Anolis* lizards Mahler et al. (2010)).

This work aims to develop a likelihood-based method to detect rapid adaptive events, referred to as shifts, using a time calibrated phylogenetic tree and potentially incomplete observations of a multivariate functional trait at the tips of that tree. The shifts can be used to cluster together species sharing a common adaptive history.

### 3.1.2 State of the Art

Phylogenetic comparative methods (PCM) are the *de facto* tools for studying phenotypic evolution. Most of them can be summarized as stochastic processes on a tree. Specifically, given a rooted phylogeny, the traits evolve according to a stochastic process on each branch of the tree. At each speciation event, one independent copy with the same initial conditions is created for each daughter species. A common stochastic process in this setting is the Brownian Motion (BM, Felsenstein, 1985). It is well suited to model the random drift of a quantitative, neutral and polygenic trait (see e.g. Felsenstein, 2004, chap. 24). Unfortunately, the BM has no stationary distribution and cannot adequately model adaptation to a specific optimum (Hansen & Orzack, 2005). The Ornstein-Uhlenbeck (OU) process is therefore preferred to the BM in the context of adaptive evolution (Hansen, 1997; Hansen et al., 2008). Note that, as pointed out by Hansen et al. (2008) and Cooper et al. (2016), this model is distinct from the process theoretically derived by Lande (1976) for stabilizing selection toward an optimum on an adaptive landscape at a micro-evolutionary timescale, and is better seen as a heuristic for the macro-evolution of the “secondary optima” themselves in a Simpsonian interpretation of evolution (Hansen et al., 2008). Recently, Levy processes have also been used to capture Simpsonian evolution (Landis et al., 2013; Duchon et al., 2017).

Extensions to multivariate traits have been proposed for both BM (Felsenstein, 1985) and OU processes (Bartoszek et al., 2012). Cybis et al. (2015) considered even more complex models, with a mix of both quantitative and discrete characters modeled with an underlying multivariate BM and a threshold model (Felsenstein, 2005, 2012) for drawing discrete characters from the underlying continuous BM.

The work on adaptive shifts also enjoyed a growing interest in the last decade. In their seminal work, Butler & King (2004) considered a univariate trait with known shift locations on the tree and estimated shift amplitudes in the trait optimal value using a maximum-likelihood framework. Beaulieu et al. (2012) extended the work by estimating shift amplitudes not only in the optimal value but also in the evolutionary rate. The focus then moved to estimating the number and locations of shifts. Eastman et al. (2011, 2013) detected shifts, respectively, in the evolutionary rate or the trait expectations, for traits evolving as BM, in a Bayesian setting using reversible jump Markov Chain Monte Carlo (rjMCMC). Ingram & Mahler (2013); Uyeda & Harmon (2014); Bastide et al. (2017b)

detected shifts in the optimal value of a trait evolving as an OU. [Uyeda & Harmon \(2014\)](#) and [Bastide et al. \(2017b\)](#) detect all shifts for a given number of shifts and use either rjMCMC or penalized likelihood to select the number of shifts. By contrast, [Ingram & Mahler \(2013\)](#) uses a stepwise procedure, based on AIC, to detect shifts sequentially, stopping when adding a shift does not improve the criteria anymore.

Extensions from univariate to multivariate shifts are more recent. It should be noted that all methods assume that shifts affect all traits simultaneously. Given known shift locations and a multivariate OU process, [Bartoszek et al. \(2012\)](#) was the first to develop a likelihood-based method (package `mvSLOUCH`) to estimate both matrices of multivariate evolutionary rates and selection strengths. [Clavel et al. \(2015\)](#) soon followed with `mvmorph`, a comprehensive package covering a wide range of multivariate processes. Detection of shifts in multivariate traits is more involved and both [Ingram & Mahler \(2013\)](#) and [Khabbazian et al. \(2016\)](#) make the simplifying assumption that all traits are independent, conditional on their shared shifts. [Ingram & Mahler \(2013\)](#) then proceed with the same stepwise procedure as in the univariate case whereas [Khabbazian et al. \(2016\)](#) uses a lasso-regression to detect the shifts and a phylogenetic BIC (pBIC) criterion to select the number of shifts.

### 3.1.3 Scope of the Article

In this work, we present a new likelihood-based method to detect evolutionary shifts in multivariate OU models. We make the simplifying assumptions that all traits have the same selection strength but, unlike in [Khabbazian et al. \(2016\)](#) and [Ingram & Mahler \(2013\)](#), traits can be correlated. Our contribution is multifaceted. We show that the scalar assumption that we make (see Section 3.2) and the independence assumption share a similar feature in their structure that make the shift detection problem tractable. Building upon a formal analysis made in the univariate case ([Bastide et al., 2017b](#)), we show that the problem suffers from identifiability issues as two or more distinct shift configurations may be indistinguishable. We propose a latent variable model combined with an OU to BM reparametrization trick to estimate the unknown number of shifts and their locations. Our method is fast and can handle missing data. It also proved accurate in a large scale simulation study and was able to find back known shift locations in re-analysis of public datasets. Finally, we show that the standard practice of decorrelating traits using phylogenetic principal component analysis (pPCA) before using a method designed for independent traits can be misleading in the presence of shifts.

The article is organized as followed. We present the model and inference procedure in Section 3.2, the theoretical bias of pPCA in the presence of shifts in Section 3.3, the simulation study in Section 3.4, the re-analysis of the New World Monkeys and Greater Antillean *Anolis* lizards datasets in Section 3.5 and discuss the results and limitations of our method in Section 3.6.

## 3.2 Model

### 3.2.1 Trait Evolution on a Tree

**Tree.** We consider a fixed and time-calibrated phylogenetic tree linking the present-day species studied. The tree is assumed ultrametric with height  $h$ , but with possible polytomies. We denote by  $n$  the number of tips and by  $m$  the number of internal nodes,

such that  $N = n + m$  is the total number of nodes. For a fully bifurcating tree,  $m = n - 1$ , and  $N = 2n - 1$ .

**Traits.** We note  $\mathbf{Y}$  the matrix of size  $n \times p$  of measured traits at the tips of the tree. For each tip  $i$ , the row-vector  $\mathbf{Y}^i$  represents the  $p$  measured traits at tip  $i$ . Some of the data might be missing, as discussed later (see Section 3.2.5).

**Brownian Motion (BM).** The multivariate BM has  $p + p(p + 1)/2$  parameters:  $p$  for the ancestral mean value vector  $\boldsymbol{\mu}$ , and  $p(p + 1)/2$  for the drift rate (in the genetic sense) matrix  $\mathbf{R}$ . The variance of a given trait grows linearly in time, and the covariance between two traits  $k$  and  $l$  at nodes  $i$  and  $j$  is given by  $t_{ij}R_{kl}$ , where  $t_{ij}$  is the time elapsed between the root and the most recent common ancestor (MRCA) of  $i$  and  $j$  (see e.g. Felsenstein, 2004, chap. 24). Using the vectorized version of matrix  $\mathbf{Y}$  (where  $\text{vec}(\mathbf{Y})$  is the vector obtained by “stacking” all the columns of  $\mathbf{Y}$ ), we get:  $\text{Var}[\text{vec}(\mathbf{Y})] = \mathbf{R} \otimes \mathbf{C}$ , where  $\otimes$  is the Kronecker product, and  $\mathbf{C} = [t_{ij}]_{1 \leq i, j \leq n}$ .

**Ornstein-Uhlenbeck (OU).** The Ornstein-Uhlenbeck process has  $p^2$  extra parameters in the form of a selection strength matrix  $\mathbf{A}$ . The traits evolve according to the stochastic differential equation  $d\mathbf{X}_t = \mathbf{A}(\boldsymbol{\beta} - \mathbf{X}_t)dt + \mathbf{R}d\mathbf{W}_t$ , where  $\mathbf{W}_t$  stands for the standard  $p$ -variate Brownian motion. The first part represents the attraction to a “primary optimum”  $\boldsymbol{\beta}$ , with a dynamic controlled by  $\mathbf{A}$ . This matrix is not necessarily symmetric in general, but it must have positive eigenvalues for the traits to indeed be attracted to their optima. This assumption also ensures the existence of a stationary state, with mean  $\boldsymbol{\beta}$  and variance  $\boldsymbol{\Gamma}$  (see Bartoszek et al., 2012; Clavel et al., 2015, for further details and general expression of  $\boldsymbol{\Gamma}$ ).

**Shifts.** We assume that some environmental changes affected the traits evolution in the past. In the BM model, we take those changes into account by allowing the process to be discontinuous, with shifts occurring in its mean value vector (as e.g. Eastman et al., 2013). This is reasonable if the adaptive response to a change in the environment is fast enough compared to the evolutionary time scale. For the OU, we assume that environmental changes result in a shift in the primary optimum  $\boldsymbol{\beta}$  (as e.g. Butler & King, 2004). The process is hence continuous, and goes to a new optimum, with a dynamic controlled by  $\mathbf{A}$ . In both cases, we make the standard assumptions that all traits shift at the same time (but with different magnitudes), that each shift occurs at the beginning of its branch, and that all other parameters ( $\mathbf{A}, \mathbf{R}$ ) of the process remain unchanged. We further assume that each jump induces a specific optimum, which implies that there is no homoplasy for the optimum, that is, no convergent evolution.

### 3.2.2 Simplifying Assumptions

**Trait Independence Assumption.** The general OU as described above is computationally hard to fit (Clavel et al., 2015), even when the shifts are fixed *a priori*. For automatic detection to be tractable in practice, several assumptions can be made. The two methods that (to our knowledge) tackle this problem in the multivariate setting assume that all the traits are independent, i.e. that matrices  $\mathbf{A}$  and  $\mathbf{R}$  are *diagonal* (Ingram & Mahler, 2013; Khabbazian et al., 2016). This is often justified by assuming that *a priori* preprocessing with phylogenetic Principal Component Analysis (pPCA, Revell, 2009) leads to

independent traits. However, pPCA assumes a no-shift BM evolution of the traits, and it can introduce a bias in the downstream analysis conducted on the scores, as shown by Uyeda et al. (2015). The choice of the number of PC axes to keep is also crucial, and can qualitatively change the results obtained, leading to the detection of artificial shifts near the root when not enough PC axes are kept for the analysis, as observed by Khabbazian et al. (2016). Finally, we show theoretically (Section 3.3) and numerically (Section 3.4, last paragraph) that pPCA fails to decorrelate the data in the presence of shifts and may even hamper shift detection accuracy.

**Scalar OU (scOU).** We offer here an alternative to the independence assumption. Computations are greatly simplified when matrices  $\mathbf{A}$  and  $\mathbf{R}$  commute. This happens when both of these matrices are diagonal for example, or when  $\mathbf{R}$  is unconstrained and  $\mathbf{A}$  is *scalar*, i.e. of the form  $\mathbf{A} = \alpha \mathbf{I}_p$ , where  $\mathbf{I}_p$  is the identity matrix. We call a process satisfying the latter assumptions a *scalar OU* (scOU), as it behaves essentially as a univariate OU. In particular, its stationary variance is simply given by  $\mathbf{\Gamma} = \mathbf{R}/(2\alpha)$  (analogous to the formula  $\gamma^2 = \sigma^2/(2\alpha)$  in the univariate case, see e.g. Hansen, 1997).

We define the scOU model as follows: at the root  $\rho$ , the traits are either drawn from the stationary normal distribution with mean  $\boldsymbol{\mu}$  and variance  $\mathbf{\Gamma}$  ( $\mathbf{X}^\rho \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{\Gamma})$ ), or fixed and equal to  $\boldsymbol{\mu}$ . The initial optimum vector is  $\boldsymbol{\beta}_0$  and the conditional distribution of trait  $\mathbf{X}^i$  at node  $i$  given trait  $\mathbf{X}^{\text{pa}(i)}$  at its parent node  $\text{pa}(i)$  is

$$\mathbf{X}^i \mid \mathbf{X}^{\text{pa}(i)} \sim \mathcal{N}\left(e^{-\alpha\ell_i}\mathbf{X}^{\text{pa}(i)} + (1 - e^{-\alpha\ell_i})\boldsymbol{\beta}_i, \frac{1}{2\alpha}(1 - e^{-\alpha\ell_i})\mathbf{R}\right) \quad (3.1)$$

where  $\boldsymbol{\beta}_i = \boldsymbol{\beta}_{\text{pa}(i)} + \boldsymbol{\Delta}^i$  is the optimal value of the process on the branch with length  $\ell_i$  going from  $\text{pa}(i)$  to  $i$  and  $\boldsymbol{\Delta}$  is the  $N \times p$  matrix of shifts on the branches of the tree: for any node  $i$  and any trait  $l$ ,  $\Delta_{il}$  is 0 if there are no shift on the branch going from  $\text{pa}(i)$  to  $i$ , and the value of the shift on trait  $l$  otherwise. At the root, we define  $\boldsymbol{\beta}_\rho = \boldsymbol{\beta}_0$  and, for each trait  $l$ :  $\Delta_{\rho l} = e^{-\alpha h}\boldsymbol{\mu}_l + (1 - e^{-\alpha h})\boldsymbol{\beta}_{0l}$ , where  $h$  is the age of the root (or tree height).

The scOU model can also be expressed under a linear form. Let  $\mathbf{U}$  be the  $N \times N$  matrix where  $U_{ij}$  is 1 if node  $j$  is an ancestor of node  $i$  and 0 otherwise. Let  $\mathbf{T}$  be the  $n \times N$  matrix made of the  $n$  rows of  $\mathbf{U}$  corresponding to tip taxa. For a given  $\alpha$ , we further define the diagonal  $N$  matrix  $\mathbf{W}(\alpha)$  with diagonal term  $W_{ii}(\alpha) = 1 - e^{-\alpha a_{\text{pa}(i)}}$  for any non-root node  $i$ , where  $a_{\text{pa}(i)}$  is the age of node  $\text{pa}(i)$ , and  $W_{\rho\rho}(\alpha) = 1$  for the root node  $\rho$ . Then the joint distribution of the observed traits  $\mathbf{Y}$  is normal

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}(\text{vec}(\mathbf{T}\mathbf{W}(\alpha)\boldsymbol{\Delta}), \mathbf{R} \otimes \mathbf{F}(\alpha)) \quad (3.2)$$

where  $\mathbf{F}(\alpha)$  is the symmetric scaled correlation matrix between the  $n$  tips, with entries  $F_{ij} = \frac{1}{2\alpha}e^{-\alpha d_{ij}}$  if the root is drawn from the stationary distribution, and  $F_{ij} = \frac{1}{2\alpha}e^{-2\alpha d_{ij}}(1 - e^{-2\alpha t_{ij}})$  if the root is fixed, where  $d_{ij}$  is the tree distance between nodes  $i$  and  $j$ . In the next section, this will allow us to rewrite scOU as a BM on a tree with rescaled branch lengths. This observation is at the core of our statistical inference strategy.

The scOU process allows us to handle the correlations that might exist between traits, and spares us from doing a preliminary pPCA. This however comes at the cost of assuming that all the traits evolve at the same rate toward their respective optima, with the same selection strength  $\alpha$ . See the 3.6 for further analysis of these assumptions.



### 3.2.3 Identifiability Issues

**Root State.** It can be easily checked that the parameters  $\boldsymbol{\mu}$  and  $\boldsymbol{\beta}_0$  at the root are not jointly identifiable from observations at the tips of an ultrametric tree, only the combination  $\boldsymbol{\lambda} = e^{-\alpha h}\boldsymbol{\mu} + (1 - e^{-\alpha h})\boldsymbol{\beta}_0$  is. See [Ho & Ané \(2014\)](#) for a derivation in the univariate case. Note that  $\boldsymbol{\lambda}$  corresponds to the first row of the shift matrix  $\boldsymbol{\Delta}$ . As we cannot decide from the data, we assume by default  $\boldsymbol{\beta}_0 = \boldsymbol{\mu} = \boldsymbol{\lambda}$ .

**Shift Position.** The location of the shifts may not always be uniquely determined, as several sets of locations (and magnitudes) may yield the same joint marginal distribution of the traits at the tips. These identifiability issues have been carefully studied in [Bastide et al. \(2017b\)](#) for the univariate case. Because we assume that all traits shift at the same time, the sets of equivalent shift locations are the same in the multivariate case as in the univariate case; only the number of parameters involved is different. So, the problem of counting the total number of parsimonious, non-equivalent shift allocations remains the same, as well as the problem of listing the allocations that are equivalent to a given one. As a consequence, all the combinatorial results and algorithms used in [Bastide et al. \(2017b\)](#) are still valid here; only the model selection criterion needs be adapted (see Section 3.2.5).

### 3.2.4 Re-scaling of the Tree

**Equivalency scOU / rBM.** As recalled above, the inference of OU models raises specific issues, mostly because some maximum likelihood estimates do not have a closed form expression. Many of these issues can be circumvented using the equivalence between the univariate BM and OU models described in [Blomberg et al. \(2003\)](#); [Ho & Ané \(2013a\)](#); [Pennell et al. \(2015\)](#), for ultrametric trees, when  $\alpha$  is known. Thanks to the scalar assumption, this equivalence extends to the multivariate case. Indeed, the marginal distribution of the traits at the observed tips  $\mathbf{Y}$  given in (3.2) is the same as the one arising from a BM model on a re-scaled tree defined by:

$$\begin{aligned} \mathbf{X}^p &\sim \mathcal{N}(\boldsymbol{\beta}_0, \ell_\rho(\alpha)\mathbf{R}) \text{ or } \mathbf{X}^p = \boldsymbol{\beta}_0 \text{ (fixed)} \\ \mathbf{X}^i \mid \mathbf{X}^{\text{pa}(i)} &\sim \mathcal{N}(\mathbf{X}^{\text{pa}(i)} + \boldsymbol{\Delta}^i(\alpha), \ell_i(\alpha)\mathbf{R}), \quad \text{for non-root node } i. \end{aligned}$$

where  $\ell_\rho(\alpha) = \frac{1}{2\alpha}e^{-2\alpha h}$ ,  $\ell_i(\alpha) = \frac{1}{2\alpha}e^{-2\alpha h}(e^{2\alpha t_i} - e^{2\alpha t_{\text{pa}(i)}})$ , and  $\boldsymbol{\Delta}^i(\alpha) = (\mathbf{W}(\alpha)\boldsymbol{\Delta})^i = (1 - e^{-\alpha(h-t_{\text{pa}(i)})})\boldsymbol{\Delta}^i$ . Note that, when the root is taken random, everything happens as if we added a fictive branch above the root with length  $\ell_\rho(\alpha)$ . The length of this branch increases when  $\alpha$  goes to zero.

We emphasize that only the distribution of the observed traits  $\mathbf{Y}$  is preserved and not the distribution of the complete dataset  $\mathbf{X}$ . As a consequence, ancestral traits at internal nodes cannot be directly inferred using this representation. Still, the equivalence recasts inference of  $\mathbf{R}$  and  $\mathbf{W}(\alpha)\boldsymbol{\Delta}$  in the scOU model into inference of the same parameters in a much simpler BM model, albeit on a tree with rescaled branch lengths  $\ell_i(\alpha)$ . Note that the rescaling depends on  $\alpha$ , which needs to be inferred separately. See the discussion (Section 3.6.1) for further analysis of this re-scaling.

### 3.2.5 Statistical Inference

**Incomplete Data Model.** We now discuss how to infer the set of parameters  $\theta = (\Delta, \mathbf{R})$ . We adopt a maximum likelihood strategy, which consists in maximizing the log-likelihood of the observed tip data  $\log p_{\theta}(\mathbf{Y})$  with respect to  $\theta$  to get the estimate  $\widehat{\theta}$ . The maximum likelihood estimate  $\widehat{\theta}$  is difficult to derive directly as the computation of  $\log p_{\theta}(\mathbf{Y})$  requires to integrate over the unobserved values of the traits at the internal nodes. We denote by  $\mathbf{Z}$  the unobserved matrix of size  $m \times p$  of these ancestral traits at internal nodes of the tree: for each internal node  $j$ ,  $\mathbf{Z}^j$  is the row-vector of the  $p$  ancestral traits at node  $j$ . Following Bastide et al. (2017b), we use the expectation-maximization (EM) algorithm (Dempster et al., 1977) that relies on an incomplete data representation of the model and takes advantage of the decomposition of  $\log p_{\theta}(\mathbf{Y})$  as  $\mathbb{E}[\log p_{\theta}(\mathbf{Y}, \mathbf{Z}) | \mathbf{Y}] - \mathbb{E}[\log p_{\theta}(\mathbf{Z} | \mathbf{Y}) | \mathbf{Y}]$ .

**EM.** The M step of the EM algorithm consists in maximizing  $\mathbb{E}[\log p_{\theta}(\mathbf{Y}, \mathbf{Z}) | \mathbf{Y}]$  with respect to  $\theta$ . For a given value of  $\alpha$ , thanks to the rescaling described in Section 3.2.4, the formulas to update  $\Delta$  and  $\mathbf{R}$  are explicit (see Appendix 3.C). The optimization of  $\alpha$  is achieved over a grid of values, at each point of which a complete EM algorithm is run. At the M step, we need the mean and variance of the unobserved traits  $\mathbf{Z}^j$  at each internal node  $j$  conditional on the observed traits  $\mathbf{Y}$  at the tips. The E step is dedicated to the computation of these values, which can be achieved via an upward-downward recursion (Felsenstein, 2004). The upward path goes from the leaves to the root, computing the conditional means and variances at each internal node given the values of its offspring in a recursive way. The downward recursion then goes from the root to the leaves, updating the values at each internal node to condition on the full taxon set. Thanks to the joint normality of the tip and internal node data, all update formulas have closed form matrix expressions, even when there are some missing values (see Appendix 3.C).

**Initialization.** The EM algorithm is known to be very sensitive to the initialization. Following Bastide et al. (2017b), we take advantage of the linear formulation (3.2) to initialize the shifts position using a lasso penalization (Tibshirani, 1996). This initialization method is similar to the procedure used in  $\ell 1ou$  (Khabbazian et al., 2016). See Appendix 3.C for more details.

**Missing Data.** EM was originally designed to handle missing data. As a consequence, the algorithm described above also applies when some traits are unobserved for some taxa. Indeed, the conditional distribution of the missing traits given the observed ones can be derived in the same way as in the E step. However, missing data break down the factorized structure of the dataset and some computational tricks are needed to handle the missing data efficiently (see Appendix 3.C).

**Model Selection.** For each value of the number of shifts  $K$ , the EM algorithm described above provides us with the maximum likelihood estimate  $\widehat{\theta}_K$ .  $K$  needs to be estimated to complete the inference procedure. We do so using a penalized likelihood approach. The model selection criterion relies on a reformulation of the model in terms of multivariate linear regression, where we remove the phylogenetic correlation, like independent contrasts and PGLS do. We can re-write (3.2), for a given  $\alpha$ , as

$$\widetilde{\mathbf{Y}} = \widetilde{\mathbf{T}}\Delta + \mathbf{E} \quad \text{where } \widetilde{\mathbf{Y}} = \mathbf{F}(\alpha)^{-1/2}\mathbf{Y}, \quad \widetilde{\mathbf{T}} = \mathbf{F}(\alpha)^{-1/2}\mathbf{TW}(\alpha),$$



where  $\mathbf{E}$  is a  $n \times p$  matrix with independent and identically distributed rows, each row being a (transposed) centered Gaussian vector with variance  $\mathbf{R}$ . In the univariate case (Bastide et al., 2017b), this representation allowed us to cast the problem in the setting considered by Baraud et al. (2009), and hence to derive a penalty on the log-likelihood, or, equivalently, on the least squares. Taking advantage of the well known fact that the maximum likelihood estimators of the coefficients are also the least square ones, and do not depend on the variance matrix  $\mathbf{R}$  (see, e.g. Mardia et al., 1979, Section 6), we propose to estimate  $K$  using the penalized least squares:

$$\widehat{K} = \arg \min_K \left( 1 + \frac{\text{pen}(K)}{n-K} \right) \sum_{j=1}^p \|\widetilde{\mathbf{Y}}_j - \widehat{\mathbf{Y}}_j^K\|^2$$

where  $\widetilde{\mathbf{Y}}_j$  is the column of  $\widetilde{\mathbf{Y}}$  for the  $j$ -th trait, and  $\widehat{\mathbf{Y}}_j^K$  the predicted means for trait  $j$  from the best model with  $K$  shifts. Using the EM results, this can be written as:

$$\widehat{K} = \arg \min_K \left( 1 + \frac{\text{pen}(K)}{n-K} \right) \text{tr} [\widehat{\mathbf{R}}(K, \hat{\alpha})]$$

where  $\widehat{\mathbf{R}}(K, \hat{\alpha})$  is the ML estimate of the variance parameter obtained by the EM for a fixed number  $K$  of shifts. The penalty is the same as in the univariate case:

$$\text{pen}(K) = A \frac{n-K-1}{n-K-2} \text{EDkhi} \left[ K, n-K-2, (K+1)^2 / |\mathcal{S}_K^{\text{PI}}| \right]$$

where EDkhi is the function from Definition 3 from Baraud et al. (2009) and  $|\mathcal{S}_K^{\text{PI}}|$  is the number of parsimonious identifiable sets of locations for  $K$  shifts, as defined in Bastide et al. (2017b). It hence might depends on the topology of the tree, for a tree with polytomies. For a fully resolved tree,  $|\mathcal{S}_K^{\text{PI}}| = \binom{2n-2-K}{K}$ .  $A$  is a normalizing constant, that must be greater than 1. In Baraud et al. (2009), the authors showed that it had little influence in the univariate case, and advised for a value around  $A = 1.1$ . We took this value as a default.

The criterion is directly inspired from the univariate case and inherits its theoretical properties in the special case  $\mathbf{R} = \sigma^2 \mathbf{I}_p$ . In general however, the criterion should be seen as a heuristic, although with good empirical properties (see Section 3.4).

### 3.2.6 Implementation

We implemented the method presented above in the `PhylogeneticEM` R package (R Core Team, 2017), available on the Comprehensive R Archive Network (CRAN). A thorough documentation of its functions, along with a brief tutorial, is available from the GitHub repository of the project ([pbastide.github.io/PhylogeneticEM](https://github.com/pbastide/PhylogeneticEM)). Thanks to a comprehensive suite of unitary tests, that cover approximately 79% of the code ([codecov.io/gh/pbastide/PhylogeneticEM](https://github.com/pbastide/PhylogeneticEM)), and that are run automatically on an independent Ubuntu server using the continuous integration tool Travis CI ([travis-ci.org](https://travis-ci.org)), the package was made as robust as possible. The computationally intensive parts of the analysis, such that the upward-downward algorithm of the M step, have been coded in C++ to improve performance (see Section 3.4 for a study of the computation times needed to solve problems of typical size). Because the inference on each  $\alpha$  value on the grid used is independent, they can be easily be done in parallel, and a built in option allows the user to choose the number of cores to be allocated to the computations.

### 3.3 pPCA and Shifts

Shift detection in multivariate settings is usually done by first decorrelating traits with pPCA before feeding phylogenetic PCs to detection procedures that assume independent traits. We show hereafter that even in the simple BM setting, phylogenetic PC may still be correlated in the presence of shifts. The problem is only exacerbated in the OU setting.

#### 3.3.1 pPCA is biased in the presence of shifts

Assume that the traits evolve as a shifted BM process on the tree, so that  $\text{vec}(\mathbf{Y}) \sim \mathcal{N}(\text{vec}(\mathbf{a}), \mathbf{R} \otimes \mathbf{C})$ , with  $\mathbf{a}$  being the  $n \times p$  matrix of trait means at the tips. Decomposing  $\mathbf{R}$  as  $\mathbf{R} = \mathbf{V}\mathbf{D}^2\mathbf{V}^T$ , pPCA relies on the fact that the columns of the matrix  $\mathbf{Y}\mathbf{V}$  are independent. Therefore, its efficiency relies on an accurate estimation of  $\mathbf{R}$ .

The estimate of  $\mathbf{R}$  used in pPCA is  $\hat{\mathbf{R}} = (n-1)^{-1}(\mathbf{Y} - \mathbf{1}_n \bar{\mathbf{Y}}^T)^T \mathbf{C}^{-1} (\mathbf{Y} - \mathbf{1}_n \bar{\mathbf{Y}}^T)$ , where  $\bar{\mathbf{Y}}^T = (\mathbf{1}_n^T \mathbf{C}^{-1} \mathbf{1}_n)^{-1} \mathbf{1}_n^T \mathbf{C}^{-1} \mathbf{Y}$ , which is known as the estimated phylogenetic mean vector (Revell, 2009). Decomposing the estimate of  $\mathbf{R}$  as  $\hat{\mathbf{R}} = \widehat{\mathbf{V}}\widehat{\mathbf{D}}^2\widehat{\mathbf{V}}^T$ , pPCA then computes the scores as  $\mathbf{S} = (\mathbf{Y} - \mathbf{1}_n \bar{\mathbf{Y}}^T) \widehat{\mathbf{V}}$ .

In the absence of shift, all species have the same mean vector  $\boldsymbol{\mu}$  so  $\mathbf{a} = \mathbf{1}_n \boldsymbol{\mu}^T$  and  $\mathbb{E}[\bar{\mathbf{Y}}] = \boldsymbol{\mu}$ . In the presence of shifts, species do not all share the same mean vector so the uniform centering is not valid anymore. As a consequence, the estimate of  $\mathbf{R}$  is biased (see appendix 3.A):

$$\mathbb{E}[\hat{\mathbf{R}}] = \mathbf{R} + \mathbf{B} \quad \text{where} \quad \mathbf{B} = \frac{1}{n-1} \mathbf{G}^T \mathbf{C}^{-1} \mathbf{G}, \quad \mathbf{G} = \mathbf{a} - \mathbf{1}_n \bar{\mathbf{a}}^T \quad (3.3)$$

The extra term  $\mathbf{B}$  is analogous to the between-group variance in the context of linear discriminant analysis and cancels out in the absence of shifts (note that  $\mathbf{R}$  is analogous to the within-group variance, see Mardia et al., 1979). Because  $\hat{\mathbf{R}}$  is biased, the columns of the score matrix  $\mathbf{S}$  resulting from pPCA are still correlated. We illustrate this phenomenon below using toy examples.

#### 3.3.2 Illustration: a simple example

To illustrate the impact of shifts on the decorrelation performed by (p)PCA, we used the simple tree presented in Figure 3.3.1a and considered three scenarios. In all scenarios, we simulated two highly correlated traits under a BM starting from (0,0) at the root and with covariance matrix  $\mathbf{R} = \begin{pmatrix} 1 & -0.9 \\ -0.9 & 1 \end{pmatrix}$ . The tree has two clearly marked clades, designed to highlight the differences between pPCA and PCA.  $\mathbf{R}$  is identical in all scenarios; any preprocessing aiming at decorrelating the traits should retrieve the eigenvectors of  $\mathbf{R}$  as PCs. In the first scenario, there are no trait shifts on the tree, corresponding to the pPCA assumptions, and pPCA is indeed quite efficient in finding the PCs (see Fig. 3.3.1b, left panel). In the second scenario, we added a shift on a long branch. This shift induces a species structure in the trait space that misleads standard PCA. The same structure can however be achieved by a large increment of the BM on that branch and large increments are likely on long branches. pPCA therefore copes with the shift quite well and is able to recover accurate PCs. More quantitatively, the bias induced by the shift on  $\hat{\mathbf{R}}$  is quite small,  $\mathbf{B} = \begin{pmatrix} 0.16 & 0.08 \\ 0.08 & 0.04 \end{pmatrix}$ , around one tenth of the values of  $\mathbf{R}$ . In

the third scenario, we put a shift on a small branch. The structure induced by the shift “breaks down” the upper clade and is unlikely to arise from the increment of a BM on that branch. It is therefore antagonistic to pPCA and results in a large bias for  $\hat{\mathbf{R}}$ : the extra term  $\mathbf{B}$  is equal to  $\begin{pmatrix} 1.58 & 0.79 \\ 0.79 & 0.4 \end{pmatrix}$  and comparable to  $\mathbf{R}$ . In that scenario, both PCA and pPCA find axes that are far away from the eigenvectors of  $\mathbf{R}$  (Figure 3.3.1b, right panel). The first eigenvector of  $\mathbf{R}$  captures the evolutionary drift correlation between traits, whereas the PCs of both PCA and pPCA capture a mix of evolutionary drift correlation and correlation resulting from shifts along the tree.

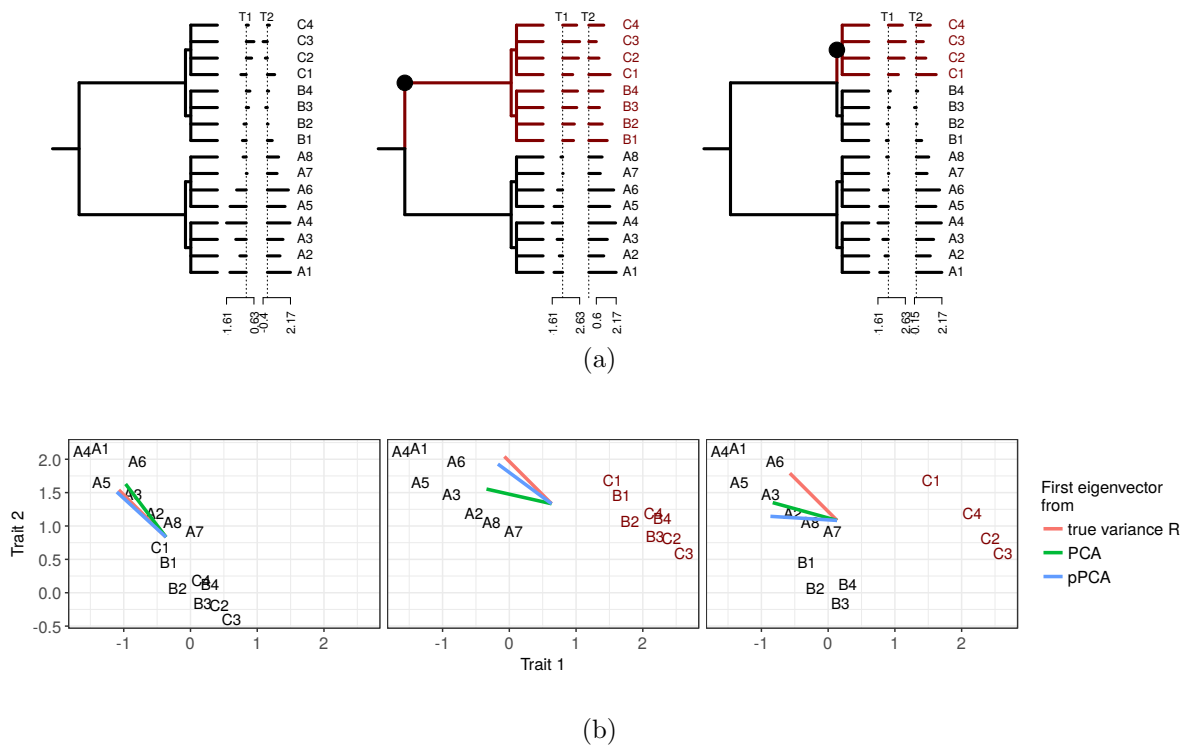


Figure 3.3.1 – Bivariate traits simulated as a BM under three scenarios: no shift (left), shift on a long branch (middle) and shift on a short branch (right). Species affected by the shift are in dark red. Top: Phylogenetic tree, shift position and simulated trait values. Bottom: Scatterplot of species in the trait space and corresponding first eigenvector computed from the true covariance  $\mathbf{R}$  (red) or found by PCA (green) and pPCA (blue).

## 3.4 Simulations Studies

### 3.4.1 Experimental Design

**General Setting.** We studied the performance of our method using a “star-like” experimental design, as opposed to a full-factorial design. We first considered a base scenario, corresponding to a base parameter set, and then varied each parameter in turn to assess its impact as in [Khabbazian et al. \(2016\)](#). The base scenario was chosen to be only moderately difficult, so that our method would find shifts most but not all of the time.

For the base scenario, we generated one 160-taxon tree according to a pure birth process, using the R package [TreeSim \(Stadler, 2011\)](#), with unit height and birth rate

$\lambda = 0.1$ . We then generated 4 traits on the phylogeny according to the scOU model, with a rather low selection strength  $\alpha_b = 1$  ( $t_{1/2} = 69\%$  of the tree height), and with a root taken with a stationary variance of  $\gamma_b^2 = \sigma_b^2 / (2\alpha_b) = 1$ . Diagonal entries of the rate matrix  $\mathbf{R}$  are  $\sigma_b^2$  and off-diagonal entries were set to  $\sigma_b^2 r_d$  with a base correlation of  $r_d = 0.4$  (correlated traits) when testing the effect of shift number and amplitude, or  $r_d = 0$  (independent traits) otherwise.

Finally, we added three shifts on this phylogeny, with fixed positions (see Figure 3.4.1). Shift amplitudes were calibrated so that the means at the tips differ by about 1 standard deviation, which constitute a reasonable shift signal (Khabbazian et al., 2016). Each configuration was replicated 100 times. We then used both our PhylogeneticEM and  $\ell 1ou$  package (Khabbazian et al., 2016) to study the simulated data. We excluded SURFACE (Ingram & Mahler, 2013) from the comparison as it is (i) quite slow, (ii) assumes the same evolutionary model as  $\ell 1ou$  and (iii) was found to achieve worse accuracy than  $\ell 1ou$  (Khabbazian et al., 2016). We used default setting for both methods. For PhylogeneticEM this implies an inference on an automatically chosen grid with 10  $\alpha$  values, on a log scale, and a maximum number of shifts of  $\sqrt{n} + 5$  (See Bastide et al. 2017b and Appendix 3.C for a justification of these default parameters).

**Number and Amplitude of Shifts.** We explored the effect of shifts by varying both their number and amplitude. We considered successively 0, 3, 7, 11, 15 shifts on the topology, with positions and values fixed as in Figure 3.4.1. Shifts values were chosen to form well separated tip groups; adjacent (in the tree) group means differ by about 1 standard deviation  $\gamma_b$ . To mimic adaptive events having different consequences on different traits, all shifts on a trait were then randomly multiplied by  $-1$  or  $+1$ . Finally and to assess the effect of shift amplitude, we rescaled all shifts by a common factor taking values in  $[0.5, 3]$ . Low scaling values correspond to smaller, harder to detect, shifts and high values to larger and easier to detect shifts.

**Selection Strength.** When exploring parameters not related to the shifts, we considered a base number of 3 shifts and a base scaling factor of 1.25, empirically found to correspond to a moderately difficult scenario. We also assumed independent traits with the same variance and selection strength (i.e. scalar  $\mathbf{A}$  and  $\mathbf{R}$ , see *model A* in appendix 3.D.1). We first varied  $\alpha$  from 1 to 3 (i.e.  $t_{1/2}$  varied between 35% and 23% of the tree height). The variance  $\sigma^2$  varied with  $\alpha$  to ensure that the stationary variance  $\gamma_b^2$  remained fixed at  $\gamma_b^2 = 1$ .

**Model Mis-specification.** The two current frameworks ( $\ell 1ou$  and scOU) for multivariate shift detection assume independent traits (diagonal  $\mathbf{A}$  and  $\mathbf{R}$ ) or correlated traits with equal selection strengths (scalar  $\mathbf{A}$  and arbitrary  $\mathbf{R}$ ). To assess robustness to model mis-specification, we simulated data under four classes of models, referred to as A, B, C, D. Model A is correctly specified for both scOU and  $\ell 1ou$  whereas B, C, D correspond respectively to mis-specifications for  $\ell 1ou$ , scOU and both. We used the Kullback-Leibler divergence between models A and B (resp. C, D) to choose parameters that attain comparable “levels” of mis-specification (see appendix 3.D.1 for details).

- *Model A* assumes scalar  $\mathbf{A}$  and  $\mathbf{R}$  (independent traits, same selection strength and variance) and meets the assumptions of both scOU and  $\ell 1ou$ .



datasets with groups that are also theoretically easier to discriminate compared to model A (see Figure 3.4.2). Indeed, we can quantify the difficulty of a dataset in terms of group separation by the Mahalanobis distance between the observed data and their expected mean, (phylogenetically) estimated in the absence of shifts:

$$D = \left\| \mathbf{Y}_{\text{vec}} - (\mathbf{1}^T \Sigma_d \mathbf{1})^{-1} \mathbf{1}^T \Sigma_d \mathbf{Y}_{\text{vec}} \right\|_{\Sigma_d^{-1}}^2 - (np - N_{\text{NA}}) \quad (3.4)$$

where  $\mathbf{Y}_{\text{vec}}$  is the vector of observed data at the tips (omitting missing values),  $\Sigma_d$  is the true variance of  $\mathbf{Y}_{\text{vec}}$  and  $N_{\text{NA}}$  is the number of missing values. In the absence of shifts  $\mathbb{E}[D] = 0$  and  $\mathbb{E}[D]$  increases when groups are well separated.

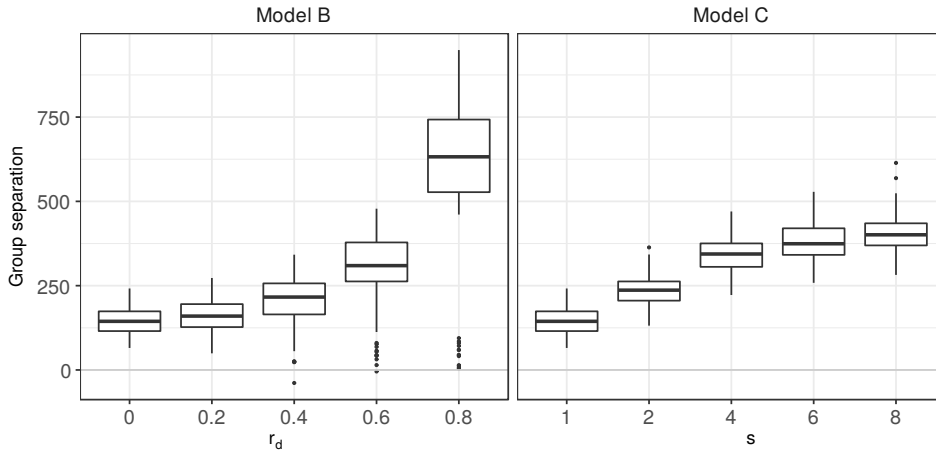


Figure 3.4.2 – Impact of trait correlation  $r_d$  (left) and unequal selection strengths  $s$  (right) on group separation, as defined in Eq. (3.4). Unequal selection strengths ( $s > 1$ ) and trait correlations ( $r_d > 0$ ) both increase group separation and make it easier to detect shifts.

**Number of Observations.** We varied the number of observations by (i) varying the number of taxa and (ii) adding missing values. To change the number of taxa, we generated 6 extra trees with the same parameters as before but with 32 to 256 taxa. The three shifts were fixed as in Figure 3.4.3. To test the ability of our method to handle missing data, we removed observations at random in our base scenario, taking care to keep at least one observed trait per species, so as not to change the number of taxa. The fraction of missing data varied from 5% to 50%.

### 3.4.2 Results

**Number and Amplitude of Shifts.** We assessed shifts detection accuracy with the Adjusted Rand Index (ARI, Hubert & Arabie, 1985) between the true clustering of the tips, and the clustering induced by the inferred shifts (Fig. 3.4.4, top). Before adjustment, the Rand index is proportional to the number of pairs of species correctly classified in the same group or correctly classified in different groups. The ARI has maximum value of 1 (for a perfectly inferred clustering) and has expected value of 0, conditional on the inferred number and size of clusters. We use this measure rather than the classical



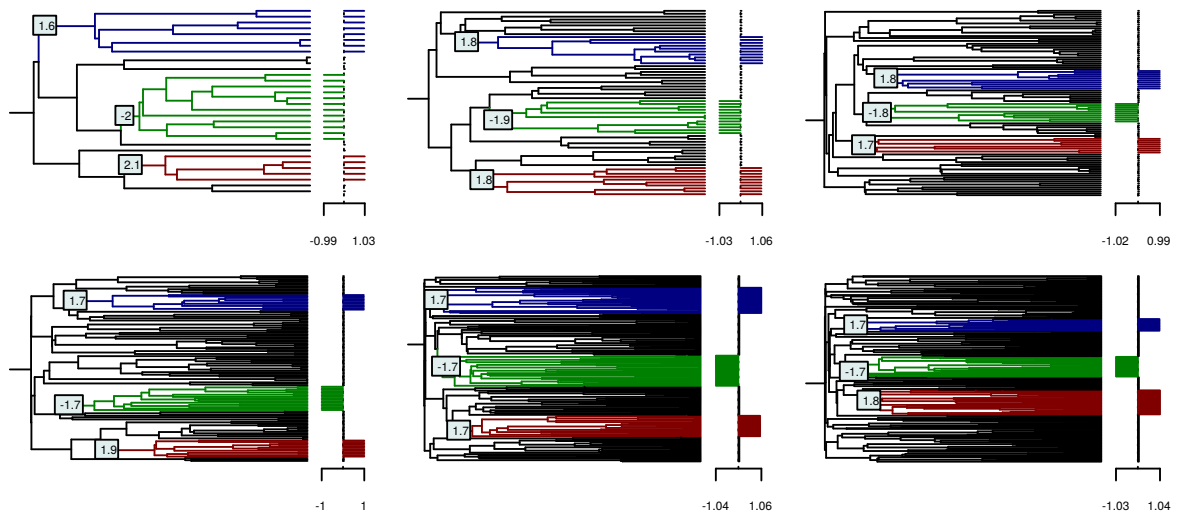


Figure 3.4.3 – Shifts locations and magnitudes used for the test trees with, respectively, 32, 64, 96, 128, 192, 256 taxa.

precision/sensitivity graphs as only the clustering can be recovered unambiguously (see Section 3.2.3). Note also that when there is no shift ( $K = 0$ ), there is only one true cluster, and the ARI is either 1 if no shift is found, or 0 otherwise (see appendix 3.D.2).

Figure 3.4.4 (top panel) shows that, unsurprisingly, both methods detect the number and positions of shifts more accurately when the shifts have higher amplitudes. *PhylogeneticEM* is also consistently better than *ℓ1ou* when there is a base correlation (here,  $r_b = 0.4$ , see section 3.4.1), which is expected as the independence assumption of *ℓ1ou* is then violated. The case  $K = 0$  (no shift) shows that *ℓ1ou* systematically finds shifts when there are none, leading to an ARI of 0. More generally, *ℓ1ou* is prone to over-estimating the number of shifts, even when they have a high magnitude (Fig. 3.4.4, bottom) whereas *PhylogeneticEM* is more conservative and underestimates the number of shifts when they are difficult to detect.

**Selection Strength and Model Mis-specifications.** Our method is relatively robust to model mis-specification (Fig. 3.4.5, top). The first panel confirms that, under model A, high values of  $\alpha$  reduce the stationary variance and lead to higher ARI values and lower RMSEs for continuous parameters (Fig. 3.4.5, bottom, leftmost panel). Similarly, *scOU* (resp. *ℓ1ou*) achieves high ARI values under well specified models A and B (resp. A and C). The mis-specification of model C (different selection strengths) does not affect *scOU* much: it has higher ARI dispersion than *ℓ1ou* but their median ARI are comparable. By contrast, *ℓ1ou* is severely affected by correlated evolution (model C) and higher levels of correlations lead to significantly lower accuracy, even though group separation is increased (Fig. 3.4.2, right). Finally, both methods are negatively affected by correlated selection strengths (Model D), although *ℓ1ou* seems more robust to this type of mis-specification.

Although shift detection is relatively unaffected by model mis-specification, parameter estimations suffers from it (Fig. 3.4.5, bottom, center and right panels). Both *ℓ1ou* and *scOU* behave better for model A than for model D and as expected, *scOU* is not affected by trait correlation (model B) whereas *ℓ1ou* is. Unequal selection strengths (model

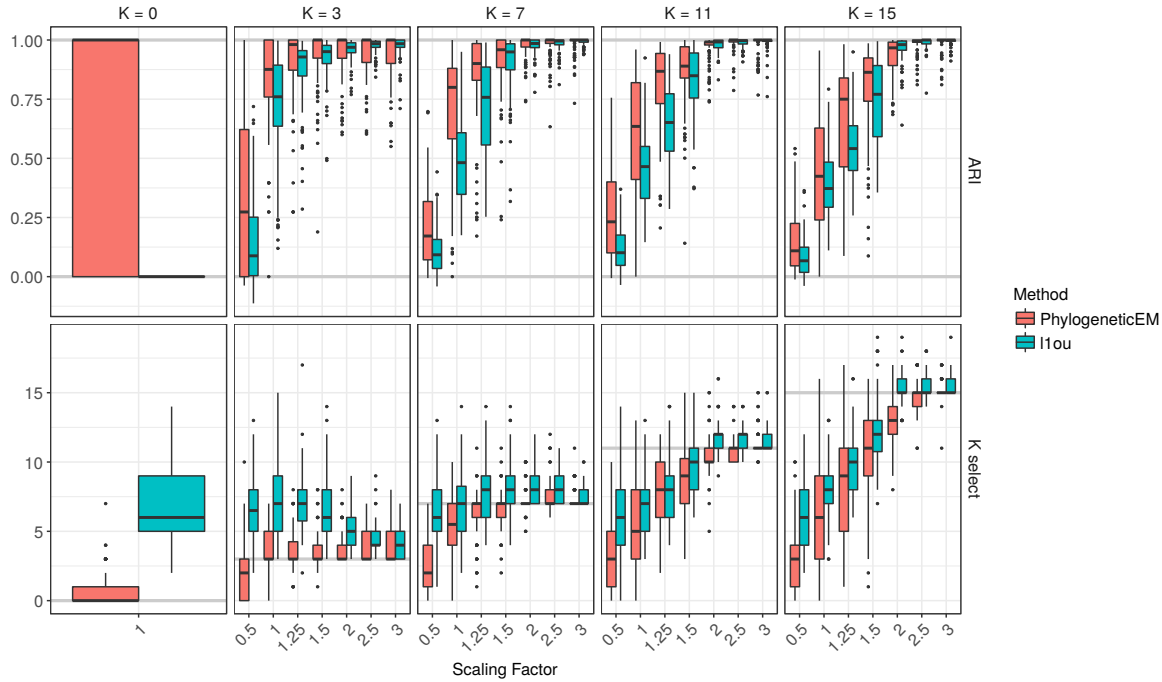


Figure 3.4.4 – ARI (top) and number of shifts selected (bottom) for the solutions found by PhylogeneticEM (red) and  $\ell_{1ou}$  (blue). Each box corresponds to one of the configuration shown in Figure 3.4.1, with a scaling factor varying between 0.5 and 3, and a true number of shift between 0 and 15 (solid lines, bottom). For the ARI, the two lines represent the maximum (1) and expected (0, for a random solution) ARI values.

C) degrades parameter estimation for both PhylogeneticEM and, surprisingly,  $\ell_{1ou}$ , that should in principle remain unaffected. Overall, features of trait evolution not properly accounted for by the inference methods (e.g. correlated selection strengths) are turned into overestimated variances. Note that the quality of the estimation of  $\Gamma$  is depends strongly on the estimation of  $\alpha$ , and could be improved by taking a finer grid for this parameter.

**Number of Observations and Computation Time.** For a given number of shifts, shift detection becomes easier as the number of taxa increases (Fig. 3.4.6, left). Furthermore, our method is robust against missing data with detection accuracy only slightly decreased when up to 50% of the observations are missing (Fig. 3.4.6, right). Finally, our implementation of the EM algorithm, using only two tree traversals (see appendix 3.C.2) and coded in C++, is reasonably fast. Inference takes roughly 15 minutes on a single core on the base 160 taxa tree and less than 45 minutes on the largest simulated trees (256 taxa).  $\ell_{1ou}$  scales less efficiently: it is faster for very small trees (32 taxa) but median running times go up to 20 hours for the large 256-taxon tree. Those long running times were unexpected and higher than the ones reported in Khabbazian et al. (2016). This discrepancy is partly due to the maximum number of shifts allowed, which strongly impacts the running time of  $\ell_{1ou}$ . Khabbazian et al. (2016) capped it at twice the true number of shifts (6 shifts in our base scenario), while we used the default setting, which is half the number of tips (i.e. from 16 to 128 shifts).



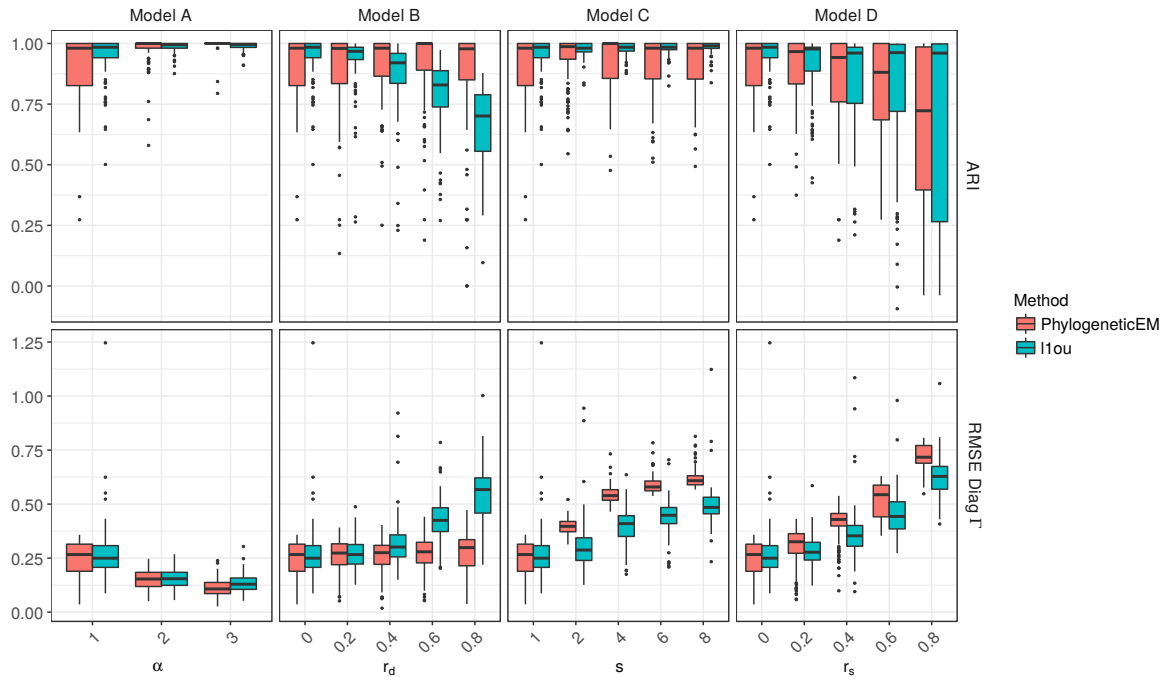


Figure 3.4.5 – ARI (top) and root mean squared error (RMSE) of the diagonal values of the estimated stationary variance  $\Gamma$  (bottom) for the solutions found by PhylogeneticEM (red) and  $\ell_{1ou}$  (blue). Each panel corresponds to a different type of mis-specification (except Model A) and the parameters  $r_d$ ,  $s$  and  $r_s$  control the level of mis-specification, with leftmost values corresponding to no mis-specification. For the ARI, the solid lines represent the maximum (1) and expected (0, for a random solution with the same number and size of clusters) ARI values.

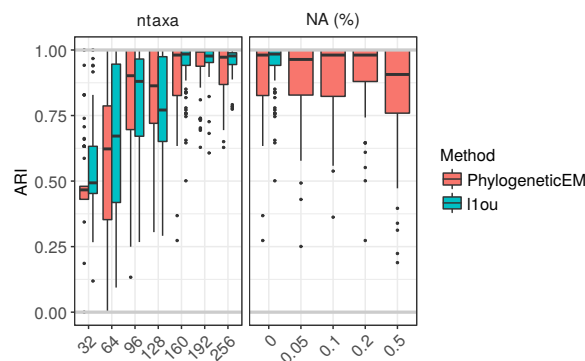


Figure 3.4.6 – ARI of the solutions found by PhylogeneticEM (red) and  $\ell_{1ou}$  (blue) when the number of taxa (left) or the number of missing values (right) increases. No ARI is available for  $\ell_{1ou}$  when there are missing values as it does not accept them in the version used here, v1.21.

**Impact of pPCA on Shift Detection Accuracy.** To illustrate how pPCA can both improve and hamper shift detection, we compared PhylogeneticEM on raw traits to  $\ell_{1ou}$  on both raw traits and phylogenetic PCs. Figure 3.4.8a shows that in our base scenario, with

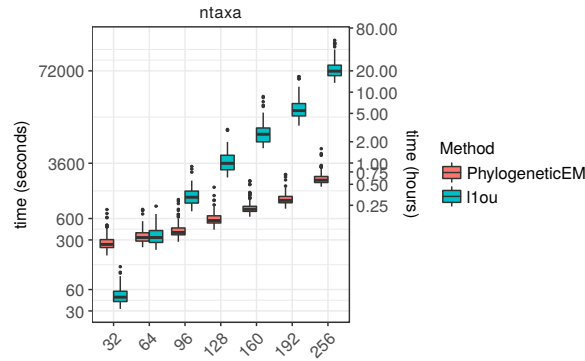


Figure 3.4.7 – Inference running times (in log-scale) of scOU and  $\ell 1_{ou}$ . All tests were run on a high-performance computing facility with CPU speeds ranging from 2.2 to 2.8Ghz.

three moderate shifts, pPCA preprocessing slightly decreases performance for low levels of correlations ( $r_d \leq 0.2$ ) but drastically improves them for moderate to high correlations levels ( $r_d \geq 0.6$ ). Although pre-processing is neutral at moderate correlation levels ( $r_d = 0.4$ ) with three “easy” shifts, it becomes harmful and degrades the performances of  $\ell 1_{ou}$  when the number or magnitude of the shifts increases (Fig. 3.4.8b). As expected, PhylogeneticEM is unaffected by the pPCA preprocessing, up to numerical issues.

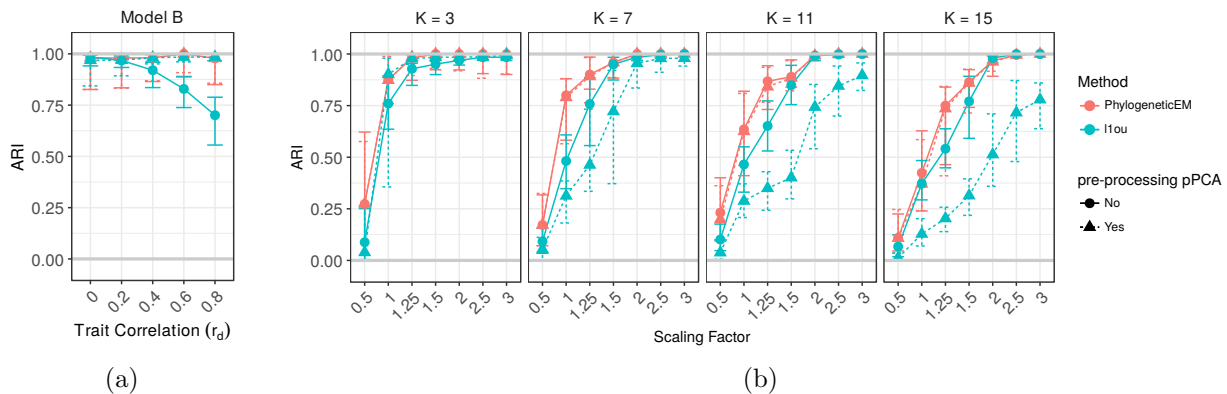


Figure 3.4.8 – ARI of the solutions found by PhylogeneticEM (red) and  $\ell 1_{ou}$  (blue), without (solid lines) or with (dotted lines) pPCA preprocessing. (a) Trait correlation ( $r_d$ ) increases from 0 to 0.8. (b) Each box corresponds to one of the configuration shown in Figure 3.4.1, and shifts are increasingly large with a scaling factor varying between 0.5 and 3.

### 3.5 Examples

We used PhylogeneticEM to re-analyse two publicly available datasets.

### 3.5.1 New World Monkeys

We first considered the evolution of brain shape in New World Monkeys studied by [Aristide et al. \(2016\)](#). The dataset consists of 49 species on a time-calibrated maximum-likelihood tree. The traits under study are the first two principal components (PC1, PC2) resulting from a PCA on 399 landmarks describing brain shape. We ran PhylogeneticEM on a grid of 30 values for the  $\alpha$  parameter. To make this parameter easily interpretable, we report the *phylogenetic half-life*  $t_{1/2} = \ln(2)/\alpha$  ([Hansen, 1997](#)), expressed in percentage of total tree height. Here,  $t_{1/2}$  took values between 0.46 % and 277.26 %. We allowed for a maximum of 20 shifts. The inference took 17.56 minutes, parallelized on 5 cores.

The model selection criterion suggests an optimal value of  $\widehat{K} = 4$  shifts (Fig. 3.5.1, inset graph). The criterion does not show a very sharp minimum, however, and a value of  $\widehat{K} = 5$  shifts also seems to be a good candidate. In order to compare our results with that presented in [Aristide et al. \(2016\)](#), we present the solution with 5 shifts (see Fig. 3.5.1, left). The solution with 4 shifts is very similar, except that the group with *Aotus* species is absent (in red, see Fig. 3.5.1, and supplementary Fig. 3.B.2 in Appendix 3.B). Note that, because of this added group, the solution with  $\widehat{K} = 5$  has 3 equivalent parsimonious allocations of the shifts (see supplementary Fig. 3.B.3 in Appendix 3.B). The groups found by PhylogeneticEM (Fig. 3.5.1) are in close agreement with the ecological niches defined in [Aristide et al. \(2016\)](#). There are three main differences. First, there is no jump associated with the *Pithecia* species who, although having their own ecological niche, seem to have quite similar brain shapes as closely related species. Second, *Callicebus* and *Aotus* are marked as convergent in [Aristide et al. \(2016\)](#) (in red, right), but form two distinct groups in our model (in pink and red, left). This is due to our assumption of no homoplasy. Finally, the group with *Chiropotes*, *Ateles* and *Cebus* species (in black) was found as having the “ancestral” trait optimum, while it is marked as “convergent” in [Aristide et al. \(2016\)](#). This is because we did not include any information from the fossil record (not available for brain shape), but instead used a parsimonious solution. Note that the coloring displayed in [Aristide et al. \(2016\)](#) is *not* parsimonious. The two models have the same number of distinct groups.

The selected  $\alpha$  value was found to be reasonably high, with  $t_{1/2} = 12.58\%$ . The estimated correlation between the two PCs was  $-0.13$ , confirming that PCA does not result in independent traits.

### 3.5.2 Lizards

We then considered the dataset from [Mahler et al. \(2013\)](#), which consists in 100 lizard species on a time-calibrated maximum likelihood tree and 11 morphological traits. We chose this example because of the large number of traits and the high correlation between traits, as all traits are highly correlated ( $0.82 < \rho < 0.97$ ) with snout-to-vent length (SVL).

To deal with the correlation between traits, [Mahler et al. \(2010, 2013\)](#) first performed a phylogenetic regression of all the traits against SVL, retrieved the residuals and then applied a phylogenetic PCA on SVL and the previous residuals, from which they used the first four components (pPC1 to pPC4) for their shift analysis. We first explored how the number of pPCs used can impact the shift detection. Hence we ran PhylogeneticEM 11 times, including 1 to 11 pPCs in the input dataset. Each run was done on a grid of 100 values of  $\alpha$ , with  $t_{1/2} = \ln(2)/\alpha \in [0.99, 693.15]$  % of tree height, and allowing for a maximum of 20 shifts. It appears that the result is quite sensitive to the number of pPCs included: the selected number of shifts varies from 20, the maximum allowed,

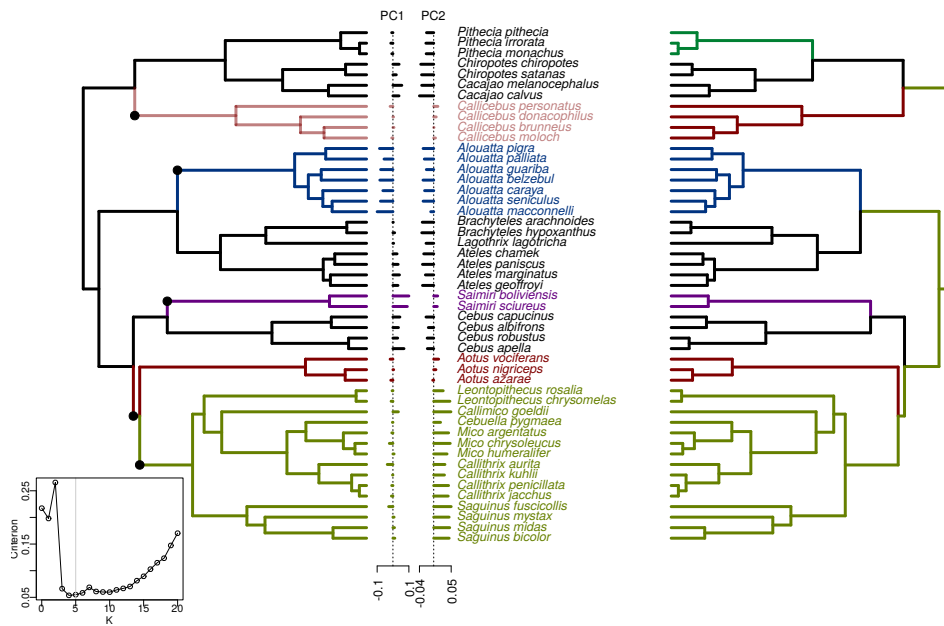


Figure 3.5.1 – Solution given by PhylogeneticEM for  $K = 5$  (left) against groups defined in [Aristide et al. \(2016, Fig. 3\)](#) (right), based on ecological criteria including *locomotion* (arboreal quadrupedal walk, clamber and suspensory locomotion or clawed locomotion), *diet* (leaves, fruits, seeds or insects) and *group size* (smaller or larger than 15 individuals). The inset graph shows the model selection criterion. The minimum is for  $K = 4$ , but  $K = 5$  is also a good candidate.

to 5 (Fig. 3.5.2). When 4 pPCs were used, as in the original study, the estimated covariance matrix  $\mathbf{R}$  contains many high correlations, showing that the pPCs are not phylogenetically independent (Fig. 3.5.2).

To avoid the difficult choice of the number of pPCs, we considered the direct analysis of the raw traits without any pre-processing, and found no shift when running PhylogeneticEM. Although the likelihood was found to increase with  $K$ , the model selection criterion profile was found erratic, suggesting numerical instability. A natural suspect for such instability is the extreme correlation between some traits (0.996 for tibia and metatarsal lengths), which results in bad conditioning of several matrices that must be inverted. To circumvent this problem, we used the two pseudo-orthogonalization strategies described above, running PhylogeneticEM on the SVL plus residuals dataset, and on the 11 pPCs, with the same parameters as above. Note that all these transformations use a rotation matrix, so that the likelihood and the least squares of the original or of any of the two transformed datasets are the same. Hence, the objective function, as well as the model selection criterion, should remain unchanged. Still, slight differences were found between the maximized likelihood for each pseudo-orthogonalized datasets. For each value of  $K$ , we therefore retained the solution with the highest likelihood.

Using the model selection criterion given in Section 3.2.5, we found  $\widehat{K} = 5$  shifts, which are displayed in Figure 3.5.3, along with the ecomorphs as described in [Mahler et al. \(2013\)](#).

Three of those shifts seem to single out grass-bush *Anolis*, that appear to have a rather small body size, with longer than expected lower limbs and tail, and shorter

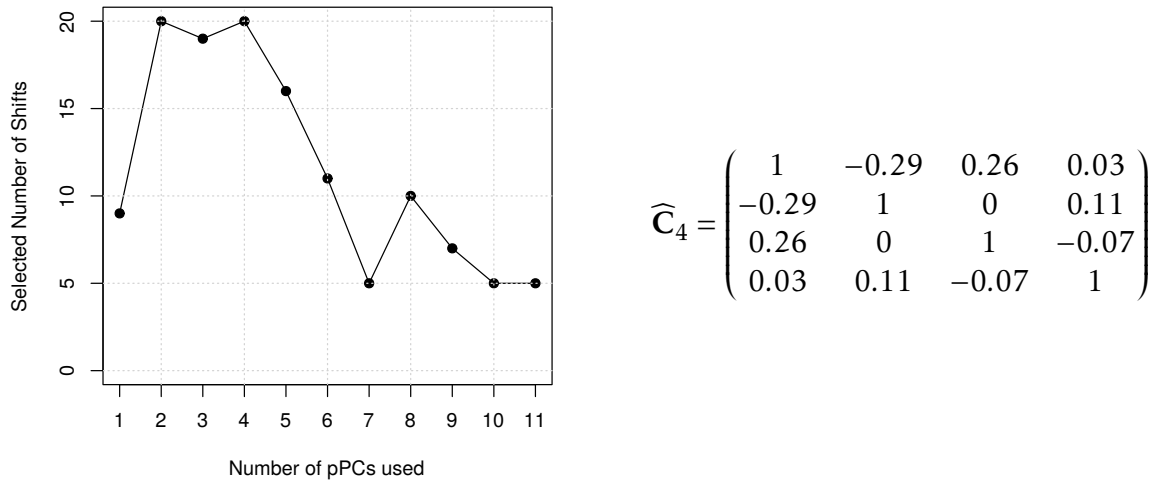


Figure 3.5.2 – Lizard dataset: selected number of shifts  $\hat{K}$  given the number of pPCs included in the analysis (left) and estimated correlation matrix between the first four pPCs (right).

upper limbs. The two others might be associated with twig *Anolis*, that have smaller than expected limbs and tails. Because of our no-homoplasy assumption, one of those shifts encompasses some species living in other ecomorphs (namely, trunk, trunk-crown and un-classified). The shift, designed to be coherent with the phylogeny, is located on the stem lineage of the smallest clade encompassing the bulk of twig lizards.

### 3.5.3 Comments

On both examples (p)PCA does not correct *a priori* for the correlation between the traits in the presence of shifts. In Section 3.3 we formally proved that it cannot correct for it, actually. As a consequence, any shift detection methods has to account for the correlation between traits.

Still, high correlations between traits may raise strong numerical issues, so PCA can be used as a *pseudo-orthogonalization* of traits, as well as any other linear distance-preserving transformation that would reduce the correlation between them. This does not dispense of considering the correlation between the transformed traits in the model.

The other interest of PCA is to reduce the dimension of the data, which may be desirable when dealing with a large number of traits, such as the original dataset from [Aristide et al. \(2016\)](#). Since PCA does not correct for the right correlation, we have no clue whether or not the dimension reduction performed by PCA is relevant for shift detection, or if it may remove precisely the direction along which the shifts occur. The relevant dimension reduction would consist in approximating the correlation matrix  $\mathbf{R}$  with a matrix of lower rank  $q < p$ . This can obviously not be done before the shifts are known, which suggests that shift detection and dimension reduction should be performed simultaneously.

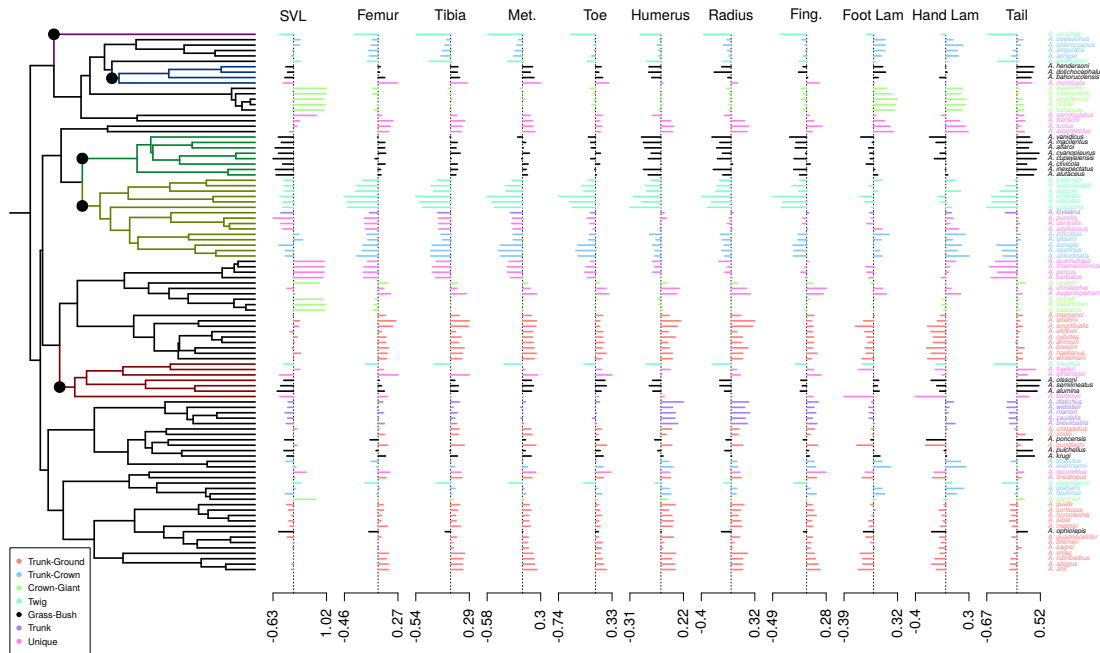


Figure 3.5.3 – Lizard dataset: solution found by PhylogeneticEM. Groups produced by the shifts are colored on the edges of the tree. The species are colored according to ecomorphs defined in Mahler et al. (2013). The traits are the snout-to-vent length (SVL), and the phylogenetic residuals of the regression against SVL of the following traits: femur length, tibia length, metatarsal IV length, toe IV length, humerus length, radius length, finger IV length, lamina number (toe and finger IV), and tail length. The same transformations were used as in Mahler et al. (2010, 2013)

### 3.6 Discussion

Many phenotypic traits appear to evolve relatively smoothly over time and across many taxa. However, changes in evolutionary pressures (dispersal to new geographic zones, diet change, etc) or key innovations (bipedal locomotion) may cause bursts of rapid trait evolution, coined evolutionary jumps by Simpson (1944). Phenotypic traits typically evolve in a coordinated way (Mahler et al., 2013; Aristide et al., 2015) and a multivariate framework is thus best suited to detect evolutionary jumps. We introduced here an Expectation Maximization algorithm embedded in a maximum-likelihood multivariate framework to infer shifts strength, location and number. Importantly, our method uses Gaussian elimination, just like Fitzjohn (2012), to avoid computing inverses of large variance-covariance matrices and can cope with missing data, an especially important problem in the multivariate setting where some traits are bound to be missing for some taxa. We demonstrated the applicability and accuracy of our method on simulated datasets and by identifying jumps for body size evolution in *Anolis* lizards and brain shapes of New World Monkeys. In both systems, the well-supported jumps occurred on stem lineages of clades that differ in terms of diet, locomotion, group size or foraging strategy (see Aristide et al. 2016 for a detailed discussion) supporting the Simpsonian

hypothesis.

### 3.6.1 Interpretation Issues

We emphasize that the interpretation of  $\alpha$  is a matter of discussion. We introduced the scOU in terms of adaptive evolution with a selection strength  $\alpha$  on the tree. However, the equivalency between OU and BM on a distorted tree suggests that  $\alpha$  can also be seen as a “phylogenetic signal” parameter, like Pagel’s  $\lambda$  (Pagel, 1999). When  $\alpha$  is small,  $\ell_i(\alpha) \simeq \ell_i$  so that branch lengths are unchanged and the phylogenetic variance is preserved. At the other end of the spectrum, when  $\alpha$  is large,  $\ell_i(\alpha) \simeq 0$  for inner branches and the rescaled tree behaves almost like a star tree. However and unlike Pagel’s  $\lambda$ ,  $\alpha$  also dictates how shifts in the optima in the original OU ( $\Delta^{OU}$ ) are transformed into shifts in the traits values in the rescaled BM ( $\Delta^{BM}(\alpha)$ ). For small  $\alpha$ , recall to the optima is weak and shifts on the optima affect the traits values minimally ( $\Delta^{BM}(\alpha) \simeq 0$ ). By contrast, for large  $\alpha$ , the recall is strong and shifts on the optima are instantaneously passed on to the traits values ( $\Delta^{BM}(\alpha) \simeq \Delta^{OU}$ ). Note however that in both cases, the topology is never lost: a shift, no matter how small its amplitude or how short the branch it occurs on, always affects the same species.

Note that if we observed traits values at some ancestral nodes (e.g. from the fossil record), the equivalency between BM and OU would break down:  $\alpha$  would recover its strict interpretation as selection strength. On non-ultrametric trees, our inference strategy does not benefit from the computational trick to speed up the M step. Similarly to the univariate case, we could write a *generalized* EM algorithm to handle this situation. In Bastide et al. (2017b), we used a lasso-based heuristic to raise, if not maximize, the objective function at the M step. It worked quite well, but was much slower. This approach could be extended to the multivariate setting, although with impaired computational burden. Note also that some shifts configuration that are not identifiable in the absence of fossil data become distinguishable with the addition of fossil data. This affects our model selection criterion, which relies on the number of distinct identifiable solutions. Computing this number on a non-ultrametric tree for an OU remains an open problem, and is probably highly dependent on the topology of the tree.

### 3.6.2 Noncausal Correlations

$\ell_{1ou}$ , SURFACE and PhylogeneticEM make many simplifying assumptions to achieve tractable models. Chief among them is the assumption that  $\mathbf{A}$  is diagonal. While  $\ell_{1ou}$  and SURFACE both assume independent traits, PhylogeneticEM can handle correlated traits through non-diagonal variance matrix  $\mathbf{R}$ . We warn the reader that correlations encoded by  $\mathbf{R}$  are not causal and only capture *coordinated* and non selective traits evolution: i.e. when arm length increases, so does leg length. In order to capture evolution of trait  $i$  *in response* to changes in trait  $j$  (i.e. when arm length strays away from its optimal value, does leg length move away or toward its own optimum) one should rather look at the value of  $A_{ij}$ , as was recently pointed out (Reitan et al., 2012; Liow et al., 2015; Manceau et al., 2016).

Our simplifying assumptions are justified by various considerations: our focus on inference of shifts rather than proper estimation of  $\mathbf{A}$  and  $\mathbf{R}$ , simulations showing that shift detection is robust to moderate values of off-diagonal terms in  $\mathbf{A}$ , difficulties to simultaneously estimate  $\alpha$  and shifts even in the univariate case (Butler & King, 2004), and



computational gain achieved by considering scalar or diagonal  $\mathbf{A}$ . They also suggest that if the focus is on causal correlation in the presence of shifts, a two-step strategy that first detects shifts using a crude but robust model, then includes those shifts in a more complex model, may achieve good performance.

The other simplifying assumption we made is that all traits shift at the same time. It makes formal analysis of identifiability issues and selection of the number of shifts similar to the univariate case, previously studied in Bastide et al. (2017b). The assumption is likely to be false in practice, however. Asynchronous shifts are an interesting extension of the model. An ambitious framework would be to build from the ground up a model that allows for different shifts on different traits. It would have to deal with the combinatorial complexity induced by asynchronous shifts, and to use a different selection criterion for the number of shifts. A less ambitious but more pragmatic approach would be a postprocessing of the shifts to select, for each shift, the traits that actually jumped. This would require derivation of confidence intervals for the shift values.

Finally, and unlike SURFACE and new version v1.40 of  $\ell 1ou$ , our model excludes convergent evolution. This limitation is shared with other shift detection methods such as bayou (Uyeda & Harmon, 2014) in the univariate case. This exclusion simplifies formal analysis and allows us to borrow from the framework of convex characters on a tree developed in Semple & Steel (2003) but is also likely to be false in practice. A straightforward extension of our method to detect convergence relies again on postprocessing of the shifts: the inferred optimal value of a trait after a shift can be tested to assess whether or not it is different from previously inferred optimal values and warrants a regime of its own.

### 3.6.3 Nature of the jumps

We model shifts as instantaneous and immediately following speciation events, like in the punctuated equilibrium theory of Eldredge & Gould (1972). We don't argue that this is necessary the case. Selection and drift can reasonably be seen as instantaneous over macroevolutionary timescales but by no means over microevolutionary timescales. There is very strong evidence, for example in peppered moths (Cook et al., 2012), that rapid adaptation can happen even in the absence of speciation. However our model does not allow us to distinguish between many small jumps distributed across a branch, one big jump anywhere on that branch and one big jump immediately following speciation, and therefore between punctuated or Simpsonian evolution.

## Acknowledgments

We are grateful to the INRA MIGALE bioinformatics platform (<http://migale.jouy.inra.fr>) for providing the computational resources needed for the experiments.

## Funding

The visit of PB to the University of Wisconsin-Madison during the fall of 2015 was funded by a grant from the Franco-American Fulbright Commission.



## Appendix

### 3.A PCA: Mathematical Derivations

**Expectation of the Estimated Variance-Covariance Matrix.** Taking  $\tilde{\mathbf{C}} = (\mathbf{1}_n^T \mathbf{C}^{-1} \mathbf{1}_n)^{-1} \mathbf{1}_n^T \mathbf{C}^{-1}$ , we have that  $\tilde{\mathbf{Y}}^T = \tilde{\mathbf{C}} \mathbf{Y}$ , and  $\tilde{\mathbf{a}}^T = \mathbb{E}[\tilde{\mathbf{Y}}^T] = \tilde{\mathbf{C}} \mathbf{a}$ . Denote by  $\mathbf{N}_{\mathbf{C}^{-1}} : \mathbb{R}^{n \times p} \rightarrow \mathbb{R}^{p^2}$  the function that to a  $n \times p$  matrix  $\mathbf{A}$  associates the  $p \times p$  matrix  $\mathbf{A}^T \mathbf{C}^{-1} \mathbf{A}$ . We get:

$$\begin{aligned} (n-1)\mathbb{E}[\hat{\mathbf{R}}] &= \mathbb{E}[\mathbf{N}_{\mathbf{C}^{-1}}(\mathbf{Y} - \mathbf{1}_n \tilde{\mathbf{Y}}^T)] = \mathbb{E}[\mathbf{N}_{\mathbf{C}^{-1}}((\mathbf{Y} - \mathbf{a}) + (\mathbf{a} - \mathbf{1}_n \tilde{\mathbf{a}}^T) + (\mathbf{1}_n \tilde{\mathbf{a}}^T - \mathbf{1}_n \tilde{\mathbf{Y}}^T))] \\ &= \mathbb{E}[\mathbf{N}_{\mathbf{C}^{-1}}((\mathbf{I} - \mathbf{1}_n \tilde{\mathbf{C}})(\mathbf{Y} - \mathbf{a}) + (\mathbf{a} - \mathbf{1}_n \tilde{\mathbf{a}}^T))] \\ &= \mathbb{E}[\mathbf{N}_{\mathbf{C}^{-1}}((\mathbf{I} - \mathbf{1}_n \tilde{\mathbf{C}})(\mathbf{Y} - \mathbf{a}))] + \mathbf{N}_{\mathbf{C}^{-1}}(\mathbf{a} - \mathbf{1}_n \tilde{\mathbf{a}}^T) \end{aligned}$$

where the two double products cancel out, as  $\mathbb{E}[\mathbf{Y}] = \mathbf{a}$ . But, for any non-singular symmetric matrix  $\mathbf{H}$ , we have:

$$\begin{aligned} \mathbb{E}[(\mathbf{Y} - \mathbf{a})^T \mathbf{H}^{-1} (\mathbf{Y} - \mathbf{a})] &= \sum_{1 \leq i, j \leq n} [\mathbf{H}^{-1}]_{ij} \mathbb{E}[(\mathbf{Y}^i - \mathbf{a}^i)(\mathbf{Y}^j - \mathbf{a}^j)^T] \\ &= \sum_{1 \leq i, j \leq n} [\mathbf{H}^{-1}]_{ij} C_{ij} \mathbf{R} = \text{tr}(\mathbf{H}^{-1} \mathbf{C}) \mathbf{R} \end{aligned}$$

Hence, applying this formula with  $\mathbf{H}^{-1} = (\mathbf{I} - \mathbf{1}_n \tilde{\mathbf{C}})^T \mathbf{C}^{-1} (\mathbf{I} - \mathbf{1}_n \tilde{\mathbf{C}}) = \mathbf{C}^{-1} - \mathbf{C}^{-1} \mathbf{1}_n \tilde{\mathbf{C}}$ , some straightforward matrix algebra manipulations give us:

$$(n-1)\mathbb{E}[\hat{\mathbf{R}}] = (n-1)\mathbf{R} + (\mathbf{a} - \mathbf{1}_n \tilde{\mathbf{a}}^T)^T \mathbf{C}^{-1} (\mathbf{a} - \mathbf{1}_n \tilde{\mathbf{a}}^T)$$

which is the result stated in the text, with  $\mathbf{G} = \mathbf{a} - \mathbf{1}_n \tilde{\mathbf{a}}^T = (\mathbf{I}_n - (\mathbf{1}_n^T \mathbf{C}^{-1} \mathbf{1}_n)^{-1} \mathbf{1}_n \mathbf{1}_n^T \mathbf{C}^{-1}) \mathbf{a}$ .

### 3.B PhylogeneticEM case study: New World Monkeys

In this section, we demonstrate the basic use of the R package `PhylogeneticEM` for the analysis of the New World Monkeys dataset ([Aristide et al., 2016](#)).

#### 3.B.1 Loading and Plotting the data

The data have been embedded in the R package `PhylogeneticEM`, to be loaded easily. The traits can be plotted on the tree thanks to the function `plot` applied to a void `params_process` object with dimension 2 (Fig. 3.B.1).

```
library(PhylogeneticEM)
data(monkeys)

plot(params_BM(p=2), data = monkeys$dat,
      phylo = monkeys$phy, show.tip.label = TRUE)
```

This `plot` function inherits from most of the optional arguments of the popular `ape` `plot` function (here for instance, the optional argument `show.tip.label` is used). Many other graphical parameters can be set by the user, so as to control the output of the function. All the results showed in the main text were produced by the package's plotting function. The two traits are represented on the right, each with its own scale. Plotting the data on the tree before analyzing it allows us to spot potential errors or outliers.

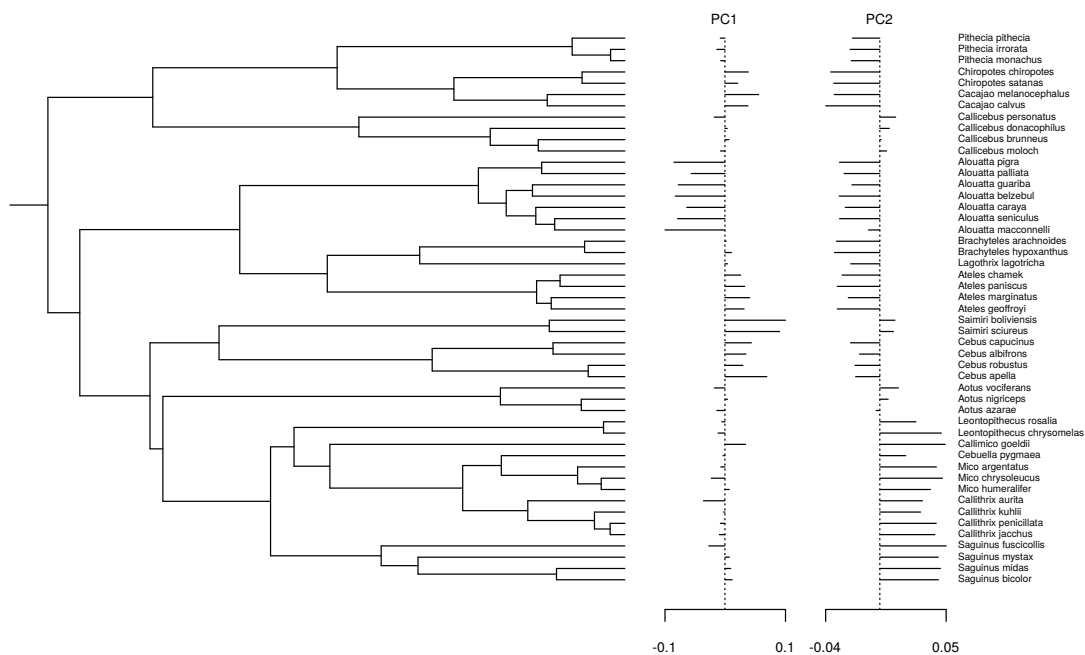


Figure 3.B.1 – New World Monkey dataset as plotted in PhylogeneticEM

### 3.B.2 Analyzing the data

The automatic shift detection is done using function `PhyloEM`. We show below how the function can be called, using an `scOU` process (with stationary root, the default), for a maximum number of shifts equal to 10, on an automatically chosen grid with 4 values for the selection strength  $\alpha$ , and parallelized on 2 cores. These parameters were chosen only to demonstrate the function, for this example analysis would run in about one minute. Different parameters were used to obtain the results below and in the main text. There are many more options available to guide the analysis, all described in the manual entry of the function.

```
res <- PhyloEM(Y_data = monkeys$dat,      ## data
               phylo = monkeys$phy,      ## phylogeny
               process = "scOU",         ## scalar OU
               K_max = 10,                ## maximal number of shifts
               nbr_alpha = 4,             ## number of alpha values
               parallel_alpha = TRUE,     ## parallelize on alpha values
               Ncores = 2)                ## number of computing cores
```

The result is stored in an object of class `PhyloEM`, which has several extractors available (see manual). By default, the plot function draws the maximum likelihood function selected by the method (Fig. 3.B.2). The same optional parameters can be used as before to control how the figure should look like.

```
plot(res, edge.width = 2, show.tip.label = TRUE)
```

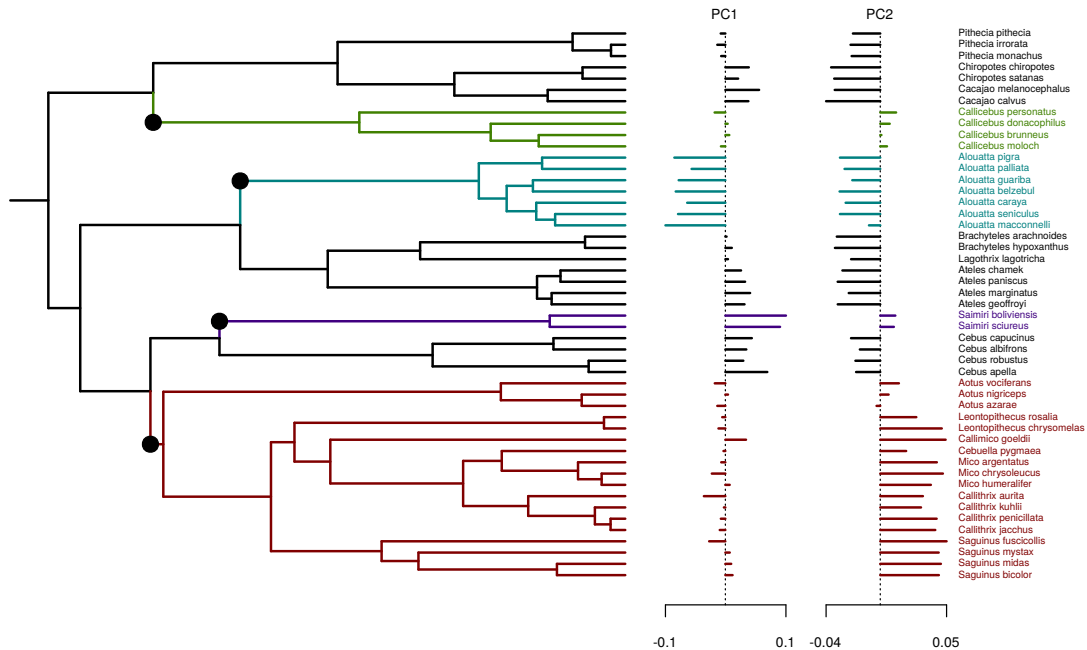


Figure 3.B.2 – Maximum likelihood solution with 4 shifts selected by the method.

The solution showed in the main text (Fig. 3.5.1) has 5 shifts, instead of 4. It can be plotted using the extractor `params_process`, which extracts some inferred parameters from an object of class `PhyloEM`.

```
params_5 <- params_process(res, K = 5)
plot(res, params = params_5)
```

### 3.B.3 Plotting Equivalent Solutions

The previous call actually results in a warning being issued:

“Warning in `params_process.PhyloEM(res, K = 5)`: There are several equivalent solutions for this shift position.”

Indeed, as mentioned in the main text, the solution with 5 shifts has three equivalent shift allocations on the branches. These solutions can be found and plotted thanks to the function `equivalent_shifts`, that returns an object that can be visualized (Fig. 3.B.3).

```
eq_shifts <- equivalent_shifts(monkeys$phy, params_5)
plot(eq_shifts, show_shifts_values = FALSE, shifts_cex = 0.5)
```

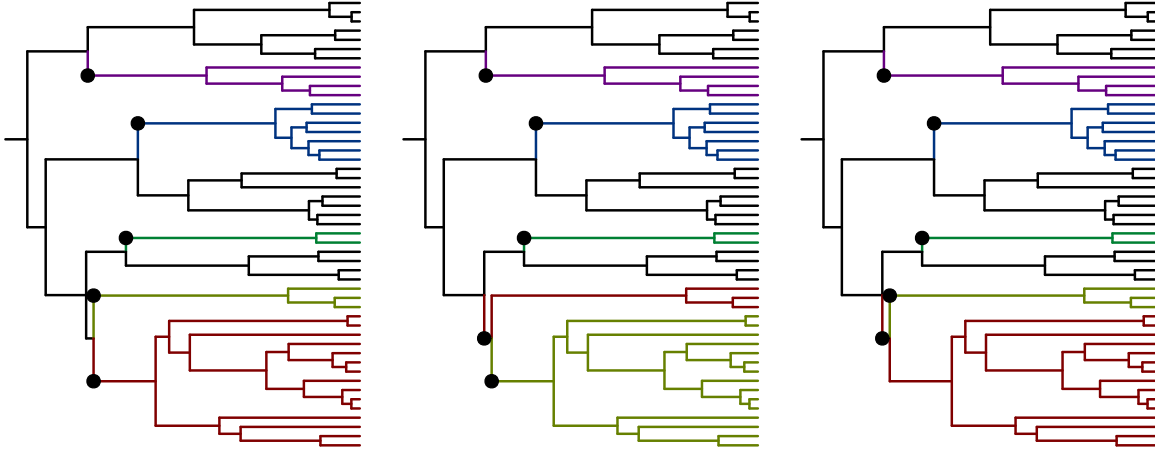


Figure 3.B.3 – The three equivalent maximum likelihood shift allocations for the solution with 5 shifts.

By default, the shifts values for the first trait is showed for all equivalent solutions. Black is always reserved to the “ancestral state”, and the value  $\lambda = \beta_0 = \mu$  of the ancestral optimal value is shown at the root. Here, the three equivalent solutions are quite straightforward, as one configuration has two shifts on sister edges. Note that the clustering of the species at the tips of the tree remains unchanged, while the historic scenario of the adaptive shifts is slightly altered. This ambiguity is inherent to the data. More information to resolve this ambiguity can only come from a prior distribution on shift values, or ideally from fossil data sampled in the right region of the tree.

### 3.C EM Inference

This section provides the update formulas for the EM algorithm in Section 3.2.5. Throughout this section, the superscript  $h$  refers to the current iteration index, e.g.  $\theta^{(h)}$  stands for the vector of parameters estimate at iteration  $h$ :  $\theta^{(h)} = (\mu^{(h)}, \Delta^{(h)}, \mathbf{R}^{(h)}, \mathbf{\Gamma}^{(h)})$ . We denote further by  $\mathbf{X}$  the  $N \times p$  matrix of the traits at all the nodes of the tree, that contains both  $\mathbf{Z}$  and  $\mathbf{Y}$ . In these derivations, nodes are numbered in a preorder, such that the root comes first:  $\rho = 1$ , the internal nodes are numbered from 1 to  $m$ , and the tips from  $m + 1$  to  $N = m + n$ .

**Conditional Expectation of the Complete Likelihood.** The EM algorithm mainly deals with  $\mathbb{E}[\log p_{\theta}(\mathbf{X}) \mid \mathbf{Y}^d]$ , where  $\mathbf{Y}^d$  is the vector of the observed tips data (that might be

missing some values). In our case we have that

$$\begin{aligned}
-2\mathbb{E}\left[\log p_{\theta}(\mathbf{X}) \mid \mathbf{Y}^d\right] &= p(m+n)\log 2\pi + p \sum_{j=2}^{m+n} \log \ell_j \\
&+ \log |\Gamma| + \text{tr}\left\{\Gamma^{-1}\text{Var}\left[\mathbf{X}^1 \mid \mathbf{Y}^d\right]\right\} + \left\|\mathbb{E}\left[\mathbf{X}^1 \mid \mathbf{Y}^d\right] - \boldsymbol{\mu}\right\|_{\Gamma^{-1}}^2 \\
&+ (m+n-1)\log |\mathbf{R}| + \sum_{j=2}^{m+n} \ell_j^{-1} \text{tr}\left\{\mathbf{R}^{-1}\text{Var}\left[\mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d\right]\right\} \\
&+ \sum_{j=2}^{m+n} \ell_j^{-1} \left\|\mathbb{E}\left[\mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d\right] - \boldsymbol{\Delta}^j\right\|_{\mathbf{R}^{-1}}^2. \tag{3.5}
\end{aligned}$$

### 3.C.1 M step

At the M step, the parameters are updated as the minimizers of (3.5) evaluated with the conditional moments of the hidden variables given  $\mathbf{Y}^d$ . We get the following updates.

**Root Parameters.**

$$\boldsymbol{\mu}^{(h+1)} = \mathbb{E}^{(h)}\left[\mathbf{X}^1 \mid \mathbf{Y}^d\right], \quad \Gamma^{(h+1)} = \text{Var}^{(h)}\left[\mathbf{X}^1 \mid \mathbf{Y}^d\right]. \tag{3.6}$$

where the conditional moments are obtained as part of the E step, see Equation (3.8). Notations  $\mathbb{E}^{(h)}$  and  $\text{Var}^{(h)}$  denote the moments taken with the law defined by current parameters  $\boldsymbol{\theta}^{(h)}$ .

**Rate Matrix.**

$$\begin{aligned}
(m+n-1)\mathbf{R}^{(h+1)} &= \sum_{j=2}^{m+n} \ell_j^{-1} \text{Var}^{(h)}\left[\mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d\right] \\
&+ \ell_j^{-1} \left(\mathbb{E}^{(h)}\left[\mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d\right] - \boldsymbol{\Delta}^{(h+1)j}\right) \\
&\cdot \left(\mathbb{E}^{(h)}\left[\mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d\right] - \boldsymbol{\Delta}^{(h+1)j}\right)^T. \tag{3.7}
\end{aligned}$$

**Optimal Shift Location.** Only the last term of (3.5) depends on the shifts so we have to minimize the sum of costs to find  $\boldsymbol{\Delta}^{(h+1)}$ :

$$\begin{aligned}
C^{(h)}(\boldsymbol{\Delta}) &= \sum_{j=2}^{m+n} C_j^{(h)}(\boldsymbol{\Delta}) \\
\text{with } C_j^{(h)}(\boldsymbol{\Delta}) &= \ell_j^{-1} \left\|\mathbb{E}^{(h)}\left[\mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d\right] - \boldsymbol{\Delta}^j\right\|_{(\mathbf{R}^{(h)})^{-1}}^2.
\end{aligned}$$

This minimization can be achieved using the same algorithm as in the univariate case (Bastide et al., 2017b) to get the optimal shifts allocations and values. Said algorithm essentially sorts the branches in decreasing order of  $C_j^{(h)}(\boldsymbol{\Delta})$  and assigns shifts to the first  $K$  branches.

### 3.C.2 E step

The aim of the E step is to compute the moments of the completed dataset given the observed traits at the tips, namely:

$$\mathbf{E}_j = \mathbb{E}[\mathbf{X}^j \mid \mathbf{Y}^d], \quad \mathbf{V}_j = \text{Var}[\mathbf{X}^j \mid \mathbf{Y}^d], \quad \mathbf{C}_{j,\text{pa}(j)} = \text{Cov}[\mathbf{X}^j; \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}^d] \quad (3.8)$$

where we dropped the dependency in  $\boldsymbol{\theta}^{(h)}$  for the sake of legibility, but all these moments are indeed taken with the laws given by the current parameters. We do so thanks to an upward-downward recursion on the tree, as described below. This algorithm can apply to a broad classes of Gaussian processes, provided that the moments of the traits at a child node are of the form:

$$\forall j \in \llbracket 2, m+n \rrbracket, \begin{cases} \mathbb{E}[\mathbf{X}^j \mid \mathbf{X}^{\text{pa}(j)}] = m_j(\mathbf{X}^{\text{pa}(j)}) = \mathbf{Q}_j \mathbf{X}^{\text{pa}(j)} + \mathbf{r}_j \\ \text{Var}[\mathbf{X}^j \mid \mathbf{X}^{\text{pa}(j)}] = \boldsymbol{\Sigma}_j \end{cases} \quad (3.9)$$

For a BM, we get

$$\mathbf{Q}_j = \mathbf{I}_p, \quad \mathbf{r}_j = \boldsymbol{\Delta}^j \quad \text{and} \quad \boldsymbol{\Sigma}_j = \ell_j \mathbf{R}.$$

A multivariate OU could also be handled, with:

$$\mathbf{Q}_j = e^{-\mathbf{A}\ell_j}, \quad \mathbf{r}_j = (\mathbf{I}_p - e^{-\mathbf{A}\ell_j})\boldsymbol{\beta}^j \quad \text{and} \quad \boldsymbol{\Sigma}_j = \boldsymbol{\Gamma} - e^{-\mathbf{A}\ell_j}\boldsymbol{\Gamma}e^{-\mathbf{A}^T\ell_j}.$$

Although we do not use these last formulas here (thanks to the equivalence between OU and BM in our setting), they are implemented in `PhylogeneticEM`, and could be readily used in an extension of the method to non-ultrametric trees with fossil taxa. To properly handle missing data in a unified framework, we first re-define *ad hoc* inversion and determinant operations that allow us to easily write the degenerated Gaussian likelihood that appears along the way.

**Missing data.** For a multivariate trait observed at node  $i$ , define the application  $f_{d_i} : \mathbb{R}^{p \times p} \rightarrow \mathbb{R}^{d_i \times d_i}$  that, given a matrix, returns the matrix with only rows and columns corresponding to observed traits. Define also the “pseudo-inverse”  $f_{d_i}^{-1} : \mathbb{R}^{d_i \times d_i} \rightarrow \mathbb{R}^{p \times p}$  that put the observed traits back into their places, and fills the un-defined lines and columns with zeros. This allows us to define a “low-dimensional inverse” as:

$$[\mathbf{S}]_{\text{ld}}^{-1} = f_{d_i}^{-1} \left( [f_{d_i}(\mathbf{S})]^{-1} \right), \quad \forall \mathbf{S} \in \mathbb{R}^{p \times p}$$

for all  $\mathbf{S}$  such that  $f_{d_i}(\mathbf{S})$  is invertible. We also define a “low dimensional determinant”, as:

$$|[\mathbf{S}]_{\text{ld}}^{-1}| = \left| [f_{d_i}(\mathbf{S})]^{-1} \right|, \quad \forall \mathbf{S} \in \mathbb{R}^{p \times p}.$$

These conventions amount to taking infinite values for the variance-covariance terms of non-observed traits. This allows us to write the following:

$$(2\pi)^{(p-d)/2} \Phi_{\mathbf{m},\mathbf{S}}(\mathbf{x}) = \Phi_{f_d(\mathbf{m}),f_d(\mathbf{S})}(f_d(\mathbf{x})).$$

where  $\Phi_{\mathbf{m},\mathbf{S}}$  denotes the density of a multivariate Gaussian, with expectation vector  $\mathbf{m}$  and variance matrix  $\mathbf{S}$ . That is, we write the density of a  $d$ -dimensional Gaussian as the density of a  $p$ -dimensional one, but with the exact same likelihood value, up to a normalizing constant  $(2\pi)^{(p-d)/2}$ . If  $d = 0$  (no data at one tip), then  $[\mathbf{S}]_{\text{ld}}^{-1}$  is a matrix of 0, and we take by convention  $|[\mathbf{S}]_{\text{ld}}^{-1}| = 1$ , so that  $\Phi_{f_d(\mathbf{m}),f_d(\mathbf{S})}(f_d(\mathbf{x})) = 1$ .

**Upward Recursion.** For a given node  $j$  in the tree, we denote by  ${}^j\mathbf{Y}^d$  the set of all traits observed at all the tips below node  $j$ . The aim of the upward recursion is to compute the Gaussian pdf  $f_{j\mathbf{Y}^d|\mathbf{X}^j}({}^j\mathbf{Y}^d; \mathbf{a})$  of  ${}^j\mathbf{Y}^d | \mathbf{X}^j$ , which we write as proportional to a Gaussian density in  $\mathbf{a}$ :

$$f_{j\mathbf{Y}^d|\mathbf{X}^j}({}^j\mathbf{Y}^d; \mathbf{a}) = A_j({}^j\mathbf{Y}^d) \Phi_{M_j({}^j\mathbf{Y}^d), S_j({}^j\mathbf{Y}^d)}(\mathbf{a}).$$

**Initialization:** For each tip  $i$ , the observed values  $(\mathbf{Y}^d)^i$  given the vector of values  $\mathbf{Y}^i$  follow a Dirac distribution:

$$\forall i \in \llbracket 1, n \rrbracket, f_{(\mathbf{Y}^d)^i|\mathbf{Y}^i}((\mathbf{Y}^d)^i; \mathbf{a}) = \delta_{(\mathbf{Y}^d)^i}(\mathbf{a}).$$

We can express this in the correct format:

$$\forall i \in \llbracket 1, n \rrbracket, f_{(\mathbf{Y}^d)^i|\mathbf{Y}^i}((\mathbf{Y}^d)^i; \mathbf{a}) = (2\pi)^{(p-d)/2} \Phi_{\mathbf{Y}^i, \mathbf{0}}(\mathbf{a})$$

but taking the “low dimensional” inverses and determinants defined above.

**Propagation:** The upward recursion formulas result from the standard properties of the conditional distribution of a multivariate Gaussian distribution plus the fact that  $L$  daughters of a given node  $\mathbf{X}^j$  are conditionally independent so

$$f_{j\mathbf{Y}^d|\mathbf{X}^j}({}^j\mathbf{Y}^d; \mathbf{a}) = \prod_{\ell=1}^L f_{j\ell\mathbf{Y}^d|\mathbf{X}^j}({}^{j\ell}\mathbf{Y}^d; \mathbf{a}).$$

We get

$$\left\{ \begin{array}{l} S_j({}^j\mathbf{Y}^d) = \left( \sum_{\ell=1}^L \mathbf{Q}_{j\ell}^T (S_{j\ell}({}^{j\ell}\mathbf{Y}^d) + \Sigma_{j\ell})^{-1} \mathbf{Q}_{j\ell} \right)^{-1} \\ M_j({}^j\mathbf{Y}^d) = S_j({}^j\mathbf{Y}^d) \sum_{\ell=1}^L \mathbf{Q}_{j\ell}^T (S_{j\ell}({}^{j\ell}\mathbf{Y}^d) + \Sigma_{j\ell})^{-1} (M_{j\ell}({}^{j\ell}\mathbf{Y}^d) - \mathbf{r}_{j\ell}) \\ \log A_j({}^j\mathbf{Y}^d) = -\frac{(L-1)p}{2} \log(2\pi) + \frac{1}{2} \log |S_j({}^j\mathbf{Y}^d)| \\ \quad + \sum_{\ell=1}^L \log A_{j\ell}({}^{j\ell}\mathbf{Y}^d) - \frac{1}{2} \log |S_{j\ell}({}^{j\ell}\mathbf{Y}^d) + \Sigma_{j\ell}| \\ \quad - \frac{1}{2} \sum_{\ell=1}^L (M_{j\ell}({}^{j\ell}\mathbf{Y}^d) - \mathbf{r}_{j\ell})^T (S_{j\ell}({}^{j\ell}\mathbf{Y}^d) + \Sigma_{j\ell})^{-1} (M_{j\ell}({}^{j\ell}\mathbf{Y}^d) - \mathbf{r}_{j\ell}) \\ \quad + \frac{1}{2} M_j({}^j\mathbf{Y}^d)^T S_j({}^j\mathbf{Y}^d)^{-1} M_j({}^j\mathbf{Y}^d) \end{array} \right.$$

where we keep track of the log of the constant  $A_j$ , for numerical accuracy. Remark that we only need to handle the infinite terms properly as described above, using the “low dimensional” inverses and determinants when needed. These terms will disappear as we go up to a node that has at least one tip with some observation for this particular trait. In the pathological case where a trait is never observed, the corresponding term remains infinite throughout the recursion, and hence does not bring any information as to the value of that trait, and does not change the likelihood. The variance of a root non-observed trait is then just the one put a priori in  $\mathbf{\Gamma}$  (see below).

**Root node and likelihood:** Once at the root, we have  $f_{\mathbf{Y}^d|\mathbf{X}^1}(\mathbf{Y}^d; \mathbf{a})$ , which is the likelihood of the observations given the root state  $\mathbf{X}^1 = \mathbf{a}$ , and we write:

$$f_{\mathbf{X}^1|\mathbf{Y}^d}(\mathbf{a}; \mathbf{Y}^d) \propto f_{\mathbf{Y}^d|\mathbf{X}^1}(\mathbf{Y}^d; \mathbf{a})f_{\mathbf{X}^1}(\mathbf{a})$$

which gives

$$\begin{cases} \text{Var}[\mathbf{X}^1 | \mathbf{Y}^d] = (\Gamma^{-1} + S_1(\mathbf{Y}^d)^{-1})^{-1} \\ \mathbb{E}[\mathbf{X}^1 | \mathbf{Y}^d] = \text{Var}[\mathbf{X}^1 | \mathbf{Y}^d](\Gamma^{-1}\boldsymbol{\mu} + S_1(\mathbf{Y}^d)^{-1}M_1(\mathbf{Y})). \end{cases}$$

**Downward Recursion.** We now derive a recursion that goes from the root back to the tips to compute the conditional moments required to evaluate (3.5). Going down the tree, we need to compute, for each node  $X_j$ ,  $2 \leq j \leq m$ ,  $\mathbf{E}_j$ ,  $\mathbf{V}_j$  and  $\mathbf{C}_{j,\text{pa}(j)}$  as in (3.8). (additionally conditioning on  $\mathbf{X}^1$  if the root is fixed).

**Initialization:** The initialization of the downward is given by the last step of the upward. If the root is random, we have

$$\begin{cases} \mathbf{V}_1 = \text{Var}[\mathbf{X}^1 | \mathbf{Y}^d] = (\Gamma^{-1} + S_1(\mathbf{Y}^d)^{-1})^{-1} \\ \mathbf{E}_1 = \mathbb{E}[\mathbf{X}^1 | \mathbf{Y}^d] = \text{Var}[\mathbf{X}^1 | \mathbf{Y}^d](\Gamma^{-1}\boldsymbol{\mu} + S_1(\mathbf{Y}^d)^{-1}M_1(\mathbf{Y})) \\ \mathbf{C}_{1,\text{pa}(1)} = \text{NA} \end{cases}$$

whereas, if we work conditionally to the root, we have  $\mathbf{V}_1 = \text{Var}[\mathbf{X}^1 | \mathbf{Y}^d, \mathbf{X}^1] = \mathbf{0}$ ,  $\mathbf{E}_1 = \mathbb{E}[\mathbf{X}^1 | \mathbf{Y}^d, \mathbf{X}^1] = \boldsymbol{\mu}$  and  $\mathbf{C}_{1,\text{pa}(1)} = \text{NA}$ .

**Propagation:** We have

$$f_{\mathbf{X}^{\text{pa}(j)}, \mathbf{X}^j | \mathbf{Y}^d}(\mathbf{a}, \mathbf{b}; \mathbf{Y}^d) = f_{\mathbf{X}^{\text{pa}(j)} | \mathbf{Y}^d}(\mathbf{a}; \mathbf{Y}^d) f_{\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)}, \mathbf{Y}^d}(\mathbf{b}; \mathbf{a}, \mathbf{Y}^d)$$

We know the first term from the recurrence, and we can compute the second term thanks to the upward step:

$$f_{\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)}, \mathbf{Y}^d}(\mathbf{b}; \mathbf{a}, \mathbf{Y}^d) = f_{\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)}, j\mathbf{Y}^d}(\mathbf{b}; \mathbf{a}, j\mathbf{Y}^d) \propto f_{\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)}}(\mathbf{b}; \mathbf{a}) f_{j\mathbf{Y}^d | \mathbf{X}^j}(j\mathbf{Y}^d; \mathbf{b})$$

As  $j\mathbf{Y}^d | \mathbf{X}^j \sim \mathcal{N}(M_j(j\mathbf{Y}^d), S_j(j\mathbf{Y}^d))$  and  $\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)} \sim \mathcal{N}(m_j(\mathbf{X}^{\text{pa}(j)}), \Sigma_j)$ , we get

$$\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)}, \mathbf{Y}^d \sim \mathcal{N}(\bar{m}_j(\mathbf{X}^{\text{pa}(j)}), \bar{\Sigma}_j)$$

with

$$\left\{ \begin{aligned} \bar{\Sigma}_j &= (S_j(j\mathbf{Y}^d)^{-1} + \Sigma_j^{-1})^{-1} \\ &= S_j(j\mathbf{Y}^d)(S_j(j\mathbf{Y}^d) + \Sigma_j)^{-1} \Sigma_j = \Sigma_j(S_j(j\mathbf{Y}^d) + \Sigma_j)^{-1} S_j(j\mathbf{Y}^d) \\ \bar{m}_j(\mathbf{X}^{\text{pa}(j)}) &= \bar{\Sigma}_j(S_j(j\mathbf{Y}^d)^{-1}M_j(j\mathbf{Y}^d) + \Sigma_j^{-1}m_j(\mathbf{X}^{\text{pa}(j)})) \\ &= \underbrace{S_j(j\mathbf{Y}^d)(S_j(j\mathbf{Y}^d) + \Sigma_j)^{-1} \mathbf{Q}_j \mathbf{X}^{\text{pa}(j)}}_{\bar{\mathbf{Q}}_j} \\ &\quad + \underbrace{S_j(j\mathbf{Y}^d)(S_j(j\mathbf{Y}^d) + \Sigma_j)^{-1} \mathbf{r}_j + \Sigma_j(S_j(j\mathbf{Y}^d) + \Sigma_j)^{-1} M_j(j\mathbf{Y}^d)}_{\bar{\mathbf{r}}_j} \end{aligned} \right.$$



Hence:

$$f_{\mathbf{X}^j | \mathbf{X}^{\text{pa}(j)}, \mathbf{Y}^d}(\mathbf{b}; \mathbf{a}, \mathbf{Y}^d) \propto \exp\left(-\frac{1}{2}(\mathbf{b} - \bar{\mathbf{m}}_j(\mathbf{a}))^T \bar{\boldsymbol{\Sigma}}_j^{-1}(\mathbf{b} - \bar{\mathbf{m}}_j(\mathbf{a}))\right)$$

And, as  $\begin{pmatrix} \mathbf{X}^j \\ \mathbf{X}^{\text{pa}(j)} \end{pmatrix} \Big| \mathbf{Y}^d \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{E}_j \\ \mathbf{E}_{\text{pa}(j)} \end{pmatrix}, \begin{pmatrix} \mathbf{V}_j & \mathbf{C}_{j,\text{pa}(j)} \\ \mathbf{C}_{j,\text{pa}(j)}^T & \mathbf{V}_{\text{pa}(j)} \end{pmatrix}\right)$ , by Gaussian conditioning, we get, for any  $\mathbf{a}$ :

$$\begin{cases} \bar{\mathbf{m}}_j(\mathbf{a}) = \mathbf{E}_j + \mathbf{C}_{j,\text{pa}(j)} \mathbf{V}_{\text{pa}(j)}^{-1}(\mathbf{a} - \mathbf{E}_{\text{pa}(j)}) \\ \bar{\boldsymbol{\Sigma}}_j = \mathbf{V}_j - \mathbf{C}_{j,\text{pa}(j)} \mathbf{V}_{\text{pa}(j)}^{-1} \mathbf{C}_{j,\text{pa}(j)}^T \end{cases}$$

From this we get:

$$\mathbf{C}_{j,\text{pa}(j)} = \bar{\mathbf{Q}}_j \mathbf{V}_{\text{pa}(j)}, \quad \mathbf{E}_j = \bar{\mathbf{r}}_j + \bar{\mathbf{Q}}_j \mathbf{E}_{\text{pa}(j)}, \quad \mathbf{V}_j = \bar{\boldsymbol{\Sigma}}_j + \bar{\mathbf{Q}}_j \mathbf{V}_{\text{pa}(j)} \bar{\mathbf{Q}}_j^T.$$

And, finally:

$$\begin{cases} \mathbf{C}_{j,\text{pa}(j)} = S_j(j\mathbf{Y}^d) (S_j(j\mathbf{Y}^d) + \boldsymbol{\Sigma}_j)^{-1} \mathbf{Q}_j \mathbf{V}_{\text{pa}(j)} \\ \mathbf{E}_j = S_j(j\mathbf{Y}^d) (S_j(j\mathbf{Y}^d) + \boldsymbol{\Sigma}_j)^{-1} (\mathbf{Q}_j \mathbf{E}_{\text{pa}(j)} + \mathbf{r}_j) + \boldsymbol{\Sigma}_j (S_j(j\mathbf{Y}^d) + \boldsymbol{\Sigma}_j)^{-1} M_j(j\mathbf{Y}^d) \\ \mathbf{V}_j = S_j(j\mathbf{Y}^d) (S_j(j\mathbf{Y}^d) + \boldsymbol{\Sigma}_j)^{-1} (\boldsymbol{\Sigma}_j + \mathbf{Q}_j \mathbf{V}_{\text{pa}(j)} \mathbf{Q}_j^T (S_j(j\mathbf{Y}^d) + \boldsymbol{\Sigma}_j)^{-1} S_j(j\mathbf{Y}^d)) \end{cases}$$

**Missing Data:** In presence of missing data, the downward formulas read

$$\begin{cases} \mathbf{C}_{j,\text{pa}(j)} = \bar{\boldsymbol{\Sigma}}_j \boldsymbol{\Sigma}_j^{-1} \mathbf{Q}_j \mathbf{V}_{\text{pa}(j)} \\ \mathbf{E}_j = \bar{\boldsymbol{\Sigma}}_j \boldsymbol{\Sigma}_j^{-1} (\mathbf{Q}_j \mathbf{E}_{\text{pa}(j)} + \mathbf{r}_j) + \bar{\boldsymbol{\Sigma}}_j S_j(j\mathbf{Y}^d)^{-1} M_j(j\mathbf{Y}^d) \\ \mathbf{V}_j = \bar{\boldsymbol{\Sigma}}_j (\mathbf{I}_p + \boldsymbol{\Sigma}_j^{-1} \mathbf{Q}_j \mathbf{V}_{\text{pa}(j)} \mathbf{Q}_j^T \boldsymbol{\Sigma}_j^{-1} \bar{\boldsymbol{\Sigma}}_j) \end{cases}$$

where  $\bar{\boldsymbol{\Sigma}}_j^{-1} = S_j(j\mathbf{Y}^d)^{-1} + \boldsymbol{\Sigma}_j^{-1}$  can be computed using the ‘‘low dimensional inverse’’ defined earlier for  $S_j(j\mathbf{Y}^d)$ , if needed.

Remark that these formulas involve the inversion of two matrices ( $\boldsymbol{\Sigma}_j$  and  $\bar{\boldsymbol{\Sigma}}_j^{-1}$ ), each of dimension  $p$  (typically small), which is not computationally intensive.

### 3.C.3 EM Initialization

Because it is only guaranteed to converge to a local optimum, the EM algorithm is highly sensitive to its starting point. As consequence, it needs to be provided with good initial guesses for the shifts positions and value, as well as the variance matrix  $\mathbf{R}$ . Initial values are determined as follows:

1. Do a lasso regression, assuming all traits are independent, choosing a penalty so that  $K$  shifts are found.
2. Find the groups of tips created by those shifts, and center each group by its empirical mean.
3. Use the centered data to estimate an empirical variance matrix. This is done using the Minimum Covariance Determinant (MCD) method, with function `covMcd` from package `robustbase` (Rousseeuw et al., 2014).

4. Use this estimated matrix to correct for correlations (see paragraph below), before running a lasso again.
5. For this second lasso, choose a penalty that selects for  $K + K_{\text{lag}}$  shifts, with  $K_{\text{lag}}$  a fixed value (default to 5). Then, using a Gauss-lasso procedure, select the best  $K$  shifts (in term of log-likelihood) among those.

This last step can be combinatorially intensive. To keep it fast, we bound the number of trials. It has proven to enhance the results of the algorithm substantially.

**Correcting for Correlations.** Using the same notations as in Section 3.2, the linear model (3.2) for the BM can be written as:

$$\text{vec}(\mathbf{Y}) \sim \mathcal{N}(\text{vec}(\mathbf{T}\Delta), \mathbf{R} \otimes \mathbf{C})$$

To apply the standard group-lasso (point 4), the coefficients of  $\mathbf{Y}$  must be independent, and identically distributed. Assume that we know  $\mathbf{R}$  the rate matrix. We can then de-correlate  $\mathbf{Y}$  by combining the equations of Section 3.2.5 (model selection) and Section 3.3.1. Indeed, we take the Cholesky transform of  $\mathbf{C} = \mathbf{C}_c \mathbf{C}_c^T$  and  $\mathbf{R} = \mathbf{R}_c \mathbf{R}_c^T$ . Then, as  $\mathbf{R} \otimes \mathbf{C} = (\mathbf{R}_c \otimes \mathbf{C}_c)(\mathbf{R}_c \otimes \mathbf{C}_c)^T$  and  $(\mathbf{R}_c \otimes \mathbf{C}_c)^{-1} = \mathbf{R}_c^{-1} \otimes \mathbf{C}_c^{-1}$  (see proposition 1.B.2), we get:

$$\text{vec}(\mathbf{C}_c^{-1} \mathbf{Y} (\mathbf{R}_c^{-1})^T) \sim \mathcal{N}(\text{vec}(\mathbf{C}_c^{-1} \mathbf{T} \Delta (\mathbf{R}_c^{-1})^T), \mathbf{I}_{np}).$$

Hence, using the standard group lasso on  $\mathbf{C}_c^{-1} \mathbf{Y} (\mathbf{R}_c^{-1})^T$ , we get an estimation of the non-zero lines of  $\Delta (\mathbf{R}_c^{-1})^T$ , which are the same as the non-zero lines of  $\Delta$ .

During point 4, we don't actually know  $\mathbf{R}$ , but we have an estimation of it, so that we can approximately decorrelate the observations using the transformation above.

### 3.C.4 Grid on $\alpha$

The inference presented above works for the rescaled BM, when the parameter  $\alpha$  is supposed to be known. In practice, this parameter needs to be estimated. One simple way to do that is to use a grid on  $\alpha$ . For each value on the grid, one can find an associated estimator, and then find the maximum likelihood estimator of the parameters by taking the best likelihood, for each number of shifts  $K$ . For instance, we plot below (Fig. 3.C.1) the likelihood profile in  $K$  for 30  $\alpha$  values on a grid, for the New World Monkey dataset (Aristide et al., 2016).

This grid of  $\alpha$  values can be provided by the user, depending on some *a priori* knowledge she might have of the problem at hand. If no grid is provided, one is automatically computed, with  $n_\alpha$  values, evenly spaced on a log scale ranging between  $\alpha_{\min}$  and  $\alpha_{\max}$ . Those extrema values are chosen in the following way.

$\alpha_{\min}$  The minimum value is chosen so that the maximum phylogenetic half-life ( $t_{1/2} = \ln(2)/\alpha$ ) is equal to  $A \ln(2)h$ , where  $h$  is the height of the tree, and  $A$  is a constant, by default equal to 3. This ensures that the lowest  $\alpha$  makes for a phylogenetic half-life approximately two times as high as the tree. Lower values of  $\alpha$  would make the process looking too much like a BM.

$\alpha_{\max}$  The maximum value of  $\alpha$  is chosen so that the correlations between tips is bounded by  $e^{-B/2}$ , with  $B$  a constant by default equal to 2. This is obtained by noting that

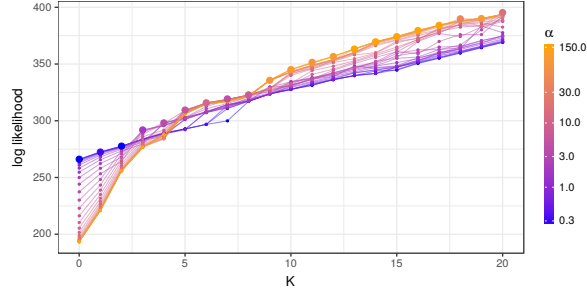


Figure 3.C.1 – Likelihood profile for all the  $\alpha$  values, on the New World Monkey dataset. Each colored line represents the likelihood of the solution for a given  $\alpha$ . The maximum value of the likelihood for each  $K$  is emphasized. The maximum is not reached by the same value of  $\alpha$  for each  $K$ . Colors in log scale.

the correlation between two tips  $i$  and  $j$  for a given trait  $k$  is given by (for a stationary root):

$$\text{Cov}[Y_{ik}; Y_{jk}] = \frac{\frac{R_{kk}}{2\alpha} e^{-2\alpha d_{ij}}}{\sqrt{\frac{R_{kk}}{2\alpha} \frac{R_{kk}}{2\alpha}}} = e^{-2\alpha d_{ij}} \leq e^{-2\alpha d_{\min}}$$

where  $d_{\min}$  is the minimum phylogenetic distance between two tips. Hence we choose  $\alpha_{\max} = B/(2d_{\min})$ .

## 3.D Simulations Appendices

### 3.D.1 Kullback-Leibler Divergences

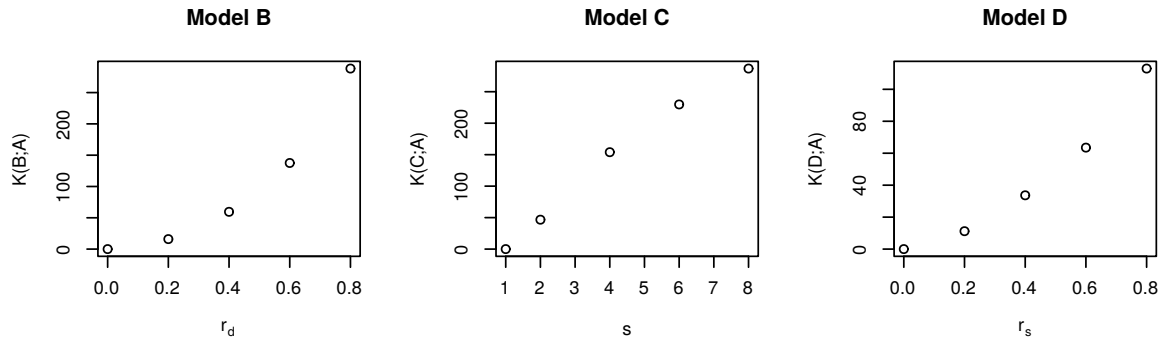


Figure 3.D.1 – KL divergences from the base model

Denote by  $\mathbf{I}_p$  the identity matrix of size  $p$ ,  $\mathbf{J}_p = \mathbf{1}^T \mathbf{1}$  the matrix filled with ones, and  $\mathbf{S}_p = \text{Diag}(s^{-(p+1)/2+q}; 1 \leq q \leq p)$  (so that  $|\mathbf{S}_p| = 1$ ). We consider the four following models:

**Model A:**  $\mathbf{A} = \alpha \mathbf{I}_p$  and  $\mathbf{R} = \sigma^2 \mathbf{I}_p$

**Model B:**  $\mathbf{A} = \alpha \mathbf{I}_p$  and  $\mathbf{R} = \mathbf{R}_{r_d} = \sigma^2(\mathbf{I}_p + r_d(\mathbf{J}_p - \mathbf{I}_p))$

**Model C:**  $\mathbf{A} = \alpha \mathbf{S}_p$  and  $\mathbf{R} = \sigma^2 \mathbf{S}_p$

**Model D:**  $\mathbf{A} = \alpha(\mathbf{I}_p + r_s(\mathbf{J}_p - \mathbf{I}_p))$  and  $\mathbf{R} = \frac{\sigma^2}{\lambda} \mathbf{I}_p$

The general formula for a Kullback divergence between two multivariate Gaussian distributions with means  $\boldsymbol{\mu}_i$  and variances  $\mathbf{V}_i$  ( $i \in \{1, 2\}$ ) is:

$$2\mathcal{K}[\mathcal{N}_1; \mathcal{N}_2] = \text{tr}(\mathbf{V}_2^{-1} \mathbf{V}_1) + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \mathbf{V}_2^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - np + \ln \frac{|\mathbf{V}_2|}{|\mathbf{V}_1|}$$

We assume that the root is in the stationary state. From the general formula for a multivariate OU, we derive the form of the variances for these four models (Bartoszek et al., 2012; Clavel et al., 2015):

**General Formula:**  $\mathbf{V}^{(i,j)} = \mathbf{P} \left( \left[ \frac{1}{\lambda_q + \lambda_r} e^{-\lambda_q(t_i - t_{ij})} e^{-\lambda_r(t_j - t_{ij})} \right]_{1 \leq q, r \leq p} \odot \mathbf{P}^{-1} \mathbf{R} \mathbf{P}^{-T} \right) \mathbf{P}^T$ , where  $\mathbf{P}$  is the orthogonal matrix of diagonalization of  $\mathbf{A}$ , associated with eigenvalues  $(\lambda_1, \dots, \lambda_p)$ .

**Model A:**  $\mathbf{V}_A = \frac{\sigma^2}{2\alpha} \mathbf{M}_\alpha \otimes \mathbf{I}_p$  with  $\mathbf{M}_\alpha = (e^{-\alpha d_{ij}})_{1 \leq i \leq j \leq n}$

**Model B:**  $\mathbf{V}_B = \frac{\sigma^2}{2\alpha} \mathbf{M}_\alpha \otimes \mathbf{R}_{r_d}$

**Model C:**  $\mathbf{V}_C^{(i,j)} = \frac{\sigma^2}{2\alpha} \text{Diag}(e^{-\alpha(\mathbf{S}_p)_{qq} d_{ij}}; 1 \leq q \leq p)$

**Model D:**  $\mathbf{V}_D^{(i,j)} = \frac{\sigma^2}{2\lambda\alpha} \mathbf{P} \text{Diag}\left(\frac{1}{1-r_s} e^{-\alpha(1-r_s)d_{ij}}, \frac{1}{1-r_s} e^{-\alpha(1-r_s)d_{ij}}, \frac{1}{1-r_s} e^{-\alpha(1-r_s)d_{ij}}, \frac{1}{1+3r_s} e^{-\alpha(1+3r_s)d_{ij}}\right) \mathbf{P}^T$

For model C, taking  $\mathbf{R} = \sigma^2 \mathbf{S}_p$  ensures that the variances at the tips for all the (independent) traits are equal to  $\gamma^2 = \frac{\sigma^2}{2\alpha}$ .

For model D, the characteristic polynomial of matrix  $\frac{1}{\alpha} \mathbf{A}$  is  $\chi(X) = (X + r_s - 1)^3 (X - 3r_s - 1)$ , so we wrote  $\mathbf{A} = \alpha \mathbf{P} \text{Diag}(1 - r_s, 1 - r_s, 1 - r_s, 1 + 3r_s) \mathbf{P}^T$ . This leads to a variance at the tips of  $\frac{\sigma^2}{2\alpha\lambda} \mathbf{P} \text{Diag}\left(\frac{1}{1-r_s}, \frac{1}{1-r_s}, \frac{1}{1-r_s}, \frac{1}{1+3r_s}\right) \mathbf{P}^T$ . By computing this matrix product (easy linear algebra formula), we find that  $\mathbf{P} \text{Diag}\left(\frac{1}{1-r_s}, \frac{1}{1-r_s}, \frac{1}{1-r_s}, \frac{1}{1+3r_s}\right) \mathbf{P}^T = (\lambda - \kappa) \mathbf{I}_p + \kappa \mathbf{J}_p$ , with  $\lambda = \frac{1+(p-2)r_s}{(1-r_s)(1+(p-1)r_s)}$  and  $\kappa = -\frac{r_s}{(1-r_s)(1+(p-1)r_s)}$ . Dividing the variance matrix by a factor  $\lambda$  hence ensures that the diagonal variances at the tips are still equal to  $\gamma^2 = \frac{\sigma^2}{2\alpha}$ .

We can then express the Kullback distance of models B, C and D to model A, using the general formula:

$$\begin{aligned} 2\mathcal{K}[i; A] &= \text{tr}(\mathbf{V}_A^{-1} \mathbf{V}_i) - np + \ln \frac{|\mathbf{V}_A|}{|\mathbf{V}_i|} + \left\| (\mathbf{T} \otimes \mathbf{I}_p) [\mathbf{W}(\mathbf{A}_A) - \mathbf{W}(\mathbf{A}_i)] \text{vec}(\Delta^T) \right\|_{\mathbf{V}_A^{-1}} \\ &= \frac{2\alpha}{\sigma^2} \text{tr}((\mathbf{M}_\alpha^{-1} \otimes \mathbf{I}_p) \mathbf{V}_i) - np + np \ln \frac{\sigma^2}{2\alpha} + p \ln |\mathbf{M}_\alpha| - \ln |\mathbf{V}_i| \\ &\quad + \left\| (\mathbf{T} \otimes \mathbf{I}_p) [\mathbf{W}(\mathbf{A}_A) - \mathbf{W}(\mathbf{A}_i)] \text{vec}(\Delta^T) \right\|_{\mathbf{V}_A^{-1}} \end{aligned}$$

For  $\mathcal{K}[B; A]$ , we can get a closed formula that does not depend on the topology (the expectations term cancels out):

$$2\mathcal{K}[B; A] = n \ln[(1-r)^3(1+3r)]$$

For the two other distances, there are no such nice simplified formula, and the result depends on the topology (even when there are no shifts). To get an idea of the distance when there are no shifts, we computed it on 100 randomly generated trees, and took the mean. With shifts, we computed the distances for the trees and shift positions chosen and shown above.

### 3.D.2 Note on the ARI (Hubert & Arabie, 1985)

**Partitions.** Let  $S$  be a set with  $n$  elements, and  $U, V$  two different partitions of  $S$ , with respectively  $R$  and  $C$  groups. Denote by  $n_{ij}$  the number of elements of  $S$  that are both in groups  $i \in \llbracket 1, R \rrbracket$  of  $U$  and  $j \in \llbracket 1, C \rrbracket$  of  $V$ , and by  $n_{i\cdot} = \sum_{j=1}^C n_{ij}$  (respectively,  $n_{\cdot j} = \sum_{i=1}^R n_{ij}$ ) the number of elements of  $S$  that are in group  $i \in \llbracket 1, R \rrbracket$  of  $U$  (resp.  $j \in \llbracket 1, C \rrbracket$  of  $V$ ).

**Rand Index.** We further define:

- $a$  the number of pairs of  $S$  that are in the same groups in both partitions  $U$  and  $V$ ,

$$a = \sum_{i=1}^R \sum_{j=1}^C \binom{n_{ij}}{2}$$

- $b$  the number of pairs of  $S$  that are in different groups in both partitions  $U$  and  $V$ ,

$$b = \binom{n}{2} - \left[ a + \left( \sum_{i=1}^R \binom{n_{i\cdot}}{2} - a \right) + \left( \sum_{j=1}^C \binom{n_{\cdot j}}{2} - a \right) \right] = \binom{n}{2} + a - \sum_{i=1}^R \binom{n_{i\cdot}}{2} - \sum_{j=1}^C \binom{n_{\cdot j}}{2}$$

Then the Rand index is defined as the number of agreeing pairs on the total number of pairs:

$$\text{Rand} = \frac{a + b}{\binom{n}{2}}$$

**Adjusted Rand Index.** Assume that the null model is a generalized hypergeometric models, where the partitions and the number of elements in each group are fixed (i.e. the  $n_{i\cdot}$  and  $n_{\cdot j}$  are fixed), and the element randomly distributed among them. Then:

$$\mathbb{E} \left[ \binom{n_{ij}}{2} \right] = \binom{n_{i\cdot}}{2} \binom{n_{\cdot j}}{2} / \binom{n}{2}$$

The ARI is then defined as (1 is the maximum value of the Rand index):

$$\text{ARI} = \frac{\text{Rand} - \mathbb{E}[\text{Rand}]}{1 - \mathbb{E}[\text{Rand}]}$$

which can be re-written as:

$$\text{ARI} = \frac{\sum_{i=1}^R \sum_{j=1}^C \binom{n_{ij}}{2} - \sum_{i=1}^R \binom{n_{i\cdot}}{2} \sum_{j=1}^C \binom{n_{\cdot j}}{2} / \binom{n}{2}}{\frac{1}{2} \left( \sum_{i=1}^R \binom{n_{i\cdot}}{2} + \sum_{j=1}^C \binom{n_{\cdot j}}{2} \right) - \sum_{i=1}^R \binom{n_{i\cdot}}{2} \sum_{j=1}^C \binom{n_{\cdot j}}{2} / \binom{n}{2}}$$

**One Class Partition.** Assume that  $R = 1$ , i.e. that one of the partition has only one class. Then:

$$\sum_{i=1}^R \sum_{j=1}^C \binom{n_{ij}}{2} = \sum_{j=1}^C \binom{n_{1j}}{2} = \sum_{j=1}^C \binom{n_{\cdot j}}{2}$$

and

$$\sum_{i=1}^R \binom{n_{i\cdot}}{2} \sum_{j=1}^C \binom{n_{\cdot j}}{2} = \binom{n_{1\cdot}}{2} \sum_{j=1}^C \binom{n_{\cdot j}}{2} = \binom{n}{2} \sum_{j=1}^C \binom{n_{\cdot j}}{2}$$

so that  $\text{ARI} = 0$ . Hence, if one of the true solution or the estimated solution has no shift, then the ARI is automatically equal to 0.

### 3.D.3 Supplementary Figures

**Sensitivity / Precision.** Because only the clustering of the tips induced by the shifts, and not their exact position on the branches of the tree, are identifiable, we used the ARI, rather than sensitivity and precision, to assess methods of shift detection. With this *caveat* in mind, we plot these quantities here for the interested reader. To do that, we removed the 6.53% of solutions that were not identifiable in the results of the methods.

These graphs confirm our conclusions drawn in the main text, with **PhylogeneticEM**, more conservative, having a better precision, along with a similar sensitivity than  $\ell 1\text{ou}$ . It is interesting to note that, even when the model is violated for **PhylogeneticEM**, the methods keeps a better or similar precision (see e.g. Model C in Fig. 3.D.3).

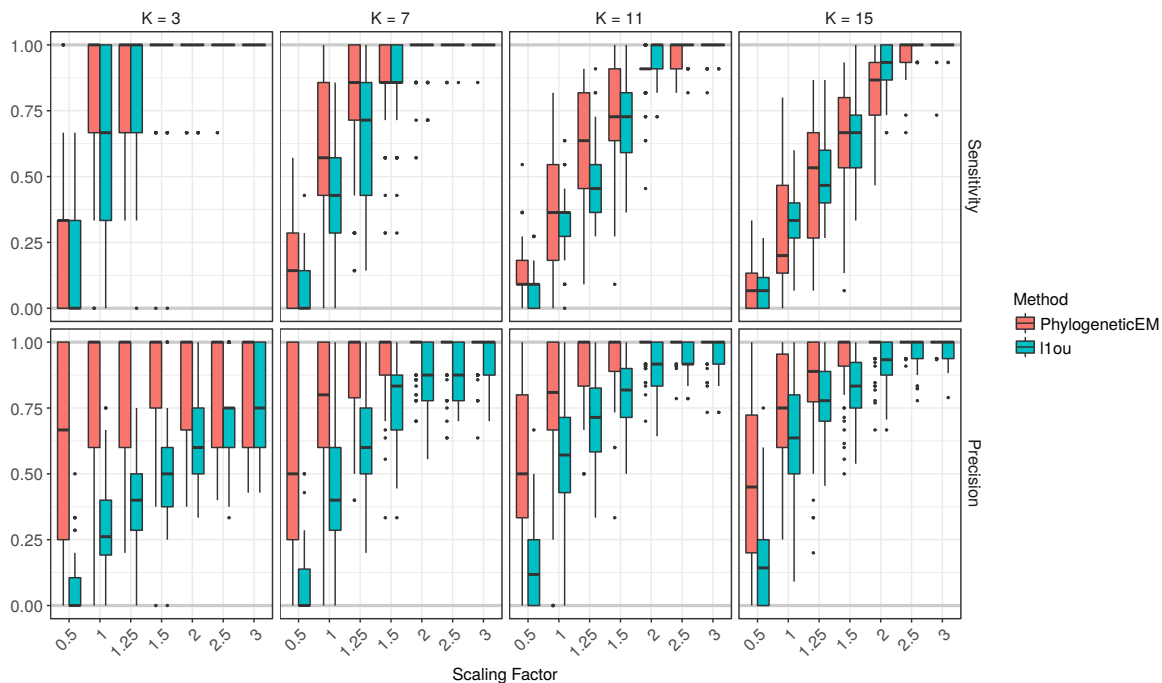


Figure 3.D.2 – Sensitivity (top) and precision (bottom) for the solutions found by **PhylogeneticEM** (red) and  $\ell 1\text{ou}$  (blue). Each box corresponds to one of the configuration shown in Figure 3.4.1, with a scaling factor varying between 0.5 and 3, and a true number of shift between 3 and 15 (solid lines, bottom).

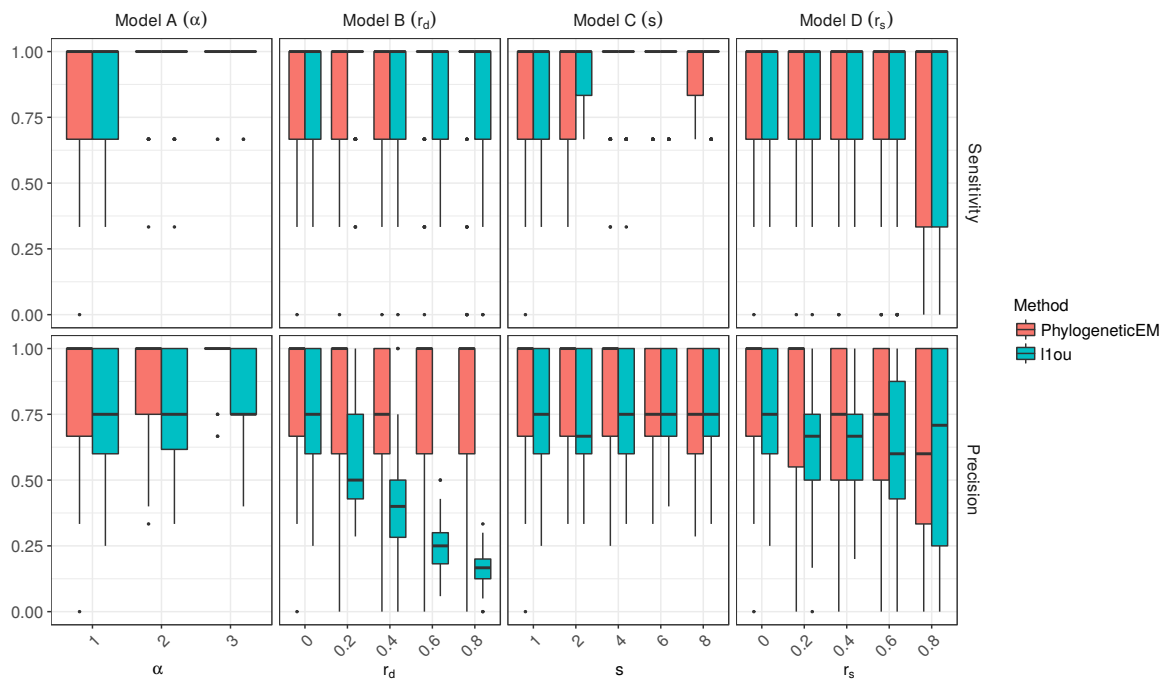


Figure 3.D.3 – Sensitivity (top) and precision (bottom) for the solutions found by PhylogeneticEM (red) and  $\ell_{1ou}$  (blue). Each panel corresponds to a different type of mis-specification (except Model A) and the parameters  $r_d$ ,  $s$  and  $r_s$  control the level of mis-specification, with leftmost values corresponding to no mis-specification. For the ARI, the solid lines represent the maximum (1) and expected (0, for a random solution) ARI values.





# Chapter 4

## Trait Evolution on Phylogenetic Networks

### Contents

---

<b>4.1</b>	<b>Introduction: Phylogenetic Networks</b>	<b>160</b>
4.1.1	Biological Insights	160
4.1.2	Formal Definition and Properties	160
4.1.3	Inference	162
<b>4.2</b>	<b>Continuous Trait Evolution on a Network</b>	<b>164</b>
4.2.1	Brownian Motion on a Network	164
4.2.2	Joint Law	167
4.2.3	Shifted BM and Heterosis	169
4.2.4	Phylogenetic Regression Model	171
<b>4.3</b>	<b>Tests of Phylogenetic Signal</b>	<b>174</b>
4.3.1	Pagel's $\lambda$ on a Network	174
4.3.2	Can we Detect Hybridization Events ?	175
4.3.3	Test of Heterosis	176
<b>4.4</b>	<b>The Julia package PhyloNetworks</b>	<b>180</b>
<b>4.5</b>	<b>Perspectives</b>	<b>181</b>
4.5.1	Shift Detection	181
4.5.2	Ornstein-Uhlenbeck	182
<b>Appendices</b>		<b>183</b>
<b>4.A</b>	<b>Documentation: Continuous Trait Evolution</b>	<b>183</b>
4.A.1	Trait simulation	183
4.A.2	Phylogenetic regression	184
4.A.3	Ancestral State Reconstruction	185
4.A.4	Phylogenetic ANOVA	189
4.A.5	Pagel's Lambda	190
<b>4.B</b>	<b>Decomposition of the Covariance Matrix</b>	<b>190</b>

---

### Foreword

This chapter describes unpublished work. It was started at the fall of 2015, during my stay at the University of Wisconsin-Madison, in the departments of Botany and Statistics, under the supervision of Cécile Ané. This is joint work with Cécile Ané, Claudia Solís-Lemus, and Mohammad Khabbazian. An article on this topic is in preparation. An application note on the Julia (Bezanson et al., 2017) package PhyloNetworks ([github.com/crsl4/PhyloNetworks.jl](https://github.com/crsl4/PhyloNetworks.jl)) is currently under review (minor revisions) for pub-

lication in *Molecular Biology and Evolution* (Solís-Lemus et al., 2017).

## 4.1 Introduction: Phylogenetic Networks

### 4.1.1 Biological Insights

In the previous chapters of this thesis, we assumed that the relationships between species was well represented by a phylogenetic tree. However, a tree can only represent the *vertical* transmission of the genetic material, from an ancestral species to its offspring. This is the most common mechanism, but *horizontal* transmission, between contemporary species, is also possible for some organisms. The main events inducing horizontal transmission of genes are *hybridization* and *Horizontal Gene Transfer*.

Hybridization happens when two distinct species produce a fertile new species, that inherits its genetic material from both parents, in varying proportions. It is sometimes associated with chromosomal doubling (allopolyploidy). Hybridization is known to be quite common for plant species, but also for animal species (25% of plant species, and 10% of animal species, are known to hybridize with at least one other species, Mallet, 2005). It is thought to be an important driver of genetic diversity and evolutionary innovation (Mallet, 2007).

Horizontal Gene Transfer is common in bacteria and archaea, and have recently been hypothesized to also play a role in the evolution of multicellular organisms (Soucy et al., 2015). It relies on many mechanisms (Soucy et al., 2015), that can involve direct transmission between organisms through a contact (conjugation), assimilation of exogenous DNA from the environment (transformation) or transmission through a predatory vector, such as a virus (transduction).

Because they happen quite frequently, and because they are important drivers of diversification and trait evolution, these events cannot be ignored. Adding some horizontal edges, they transform the usual phylogenetic tree into a *phylogenetic network*. In the next two sections, we give a formal definition of a phylogenetic network, and browse the main statistical methods that aim at inferring them.

### 4.1.2 Formal Definition and Properties

We only consider here rooted phylogenetic networks, and use the definition given in Solís-Lemus & Ané (2016).

**Definition 4.1.1** (Rooted Binary Directed Network). A reticulate network  $N = (V, E)$  is a connected directed acyclic graph with vertices  $V = \{\rho\} \cup V_L \cup V_H \cup V_T$  and edges  $E = E_H \cup E_T$  such that:

- the root  $\rho$  has an in-degree 0, and out-degree 2;
- vertices in  $V_L$  (leaves) have an in-degree 1, and out-degree 0;
- vertices in  $V_T$  (tree nodes) have an in-degree 1 and out-degree 2;
- vertices in  $V_H$  (hybrid nodes) have an in-degree 2 and out-degree 1;
- edges in  $E_T$  (tree branches) have a tree node child;

- edges in  $E_H$  (hybrid branches) have a hybrid node child.

We then define phylogenetic networks the same way we did for phylogenetic trees (see Section 1.1.1).

**Definition 4.1.2** (Phylogenetic Network). A *phylogenetic network* (on  $X$ )  $\mathcal{N}$  is a pair  $(N, \phi)$ , where  $N = (V, E)$  is a network, and  $\phi : X \rightarrow V_L$  is a bijection between a set of labels  $X$  and the set  $V_L$  of the leaves of  $N$ .

The definition above is quite general, and can encompass some fairly intricate networks. However, for identifiability reasons (Solís-Lemus & Ané, 2016), we will limit ourselves to “simple” networks, that have clearly separated cycles. An example of such a network is given Figure 4.1.1.

**Definition 4.1.3** (Level-1 Network). A Level-1 network (also called a cactus network, see e.g. Alexeev & Alekseyev 2016) is a network such that any given edge belongs to at most one cycle.

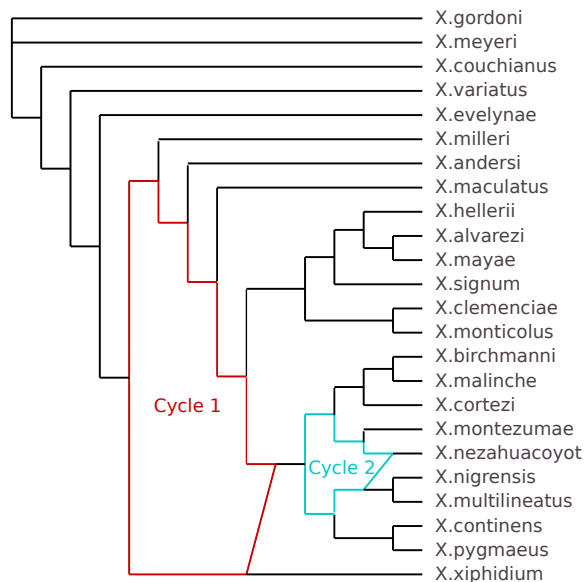


Figure 4.1.1 – Phylogenetic Network of the *Xiphophorus* Fish family, as inferred by Solís-Lemus & Ané (2016). It has two hybridization events. This is a level-1 network: cycle 1 (red) and cycle 2 (cyan) are disjoint. The figure is plotted with julia package PhyloNetworks, and slightly modified with a vector graphic editor.

For phylogenetic trees, we defined a preorder of the nodes, that allowed for efficient traversal of the tree (see Def. 1.1.3 in Section 1.1.1). We can define a similar order for networks, called the *topological sorting* (Kahn, 1962).

**Definition 4.1.4** (Topological Sorting). A topological sorting of the nodes of a rooted network  $N = (V, E)$ , also called a preorder, is such that any node comes after all its parents: for any two nodes numbered  $i$  and  $j$ , if there is an oriented path going from  $i$  to  $j$ , then  $i \leq j$ . Such an ordering can be obtained in a linear time in the number of nodes and edges (Kahn, 1962).

Finally, as mentioned above, a hybrid species inherits its genetic material from its two parents with given proportions, that might vary from one case to another. These proportions are represented by an extra parameter  $\boldsymbol{\gamma}$ , that represents *inheritance probabilities*.

**Definition 4.1.5** (Weighted Network). A weighted phylogenetic network  $(\mathcal{N}, \boldsymbol{\gamma})$  is a phylogenetic network where each edge  $e \in E$  is weighted by an inheritance probability factor  $\gamma_e$ , that is such that:

- for any edge  $e \in E$ ,  $\gamma_e \in [0, 1]$ ;
- for any tree edge  $e \in E_T$ ,  $\gamma_e = 1$ ;
- for any hybrid node with parents edges  $e_a$  and  $e_b$ ,  $\gamma_{e_a} + \gamma_{e_b} = 1$ .

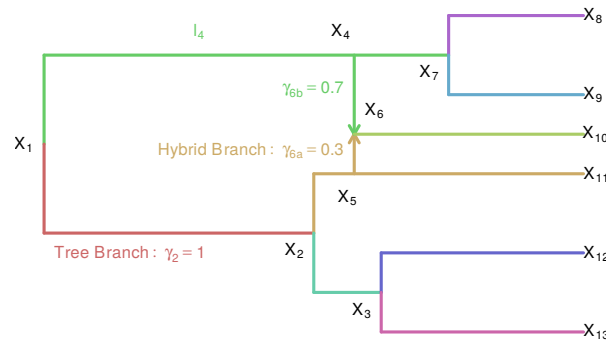


Figure 4.1.2 – A dated weighted binary rooted network with six tips and one hybridization event. Inheritance probabilities at the hybridization event are  $\gamma_{6a}$  and  $\gamma_{6b}$ , with  $\gamma_{6a} + \gamma_{6b} = 1$ . Inheritance probabilities on tree branches are equal to 1.

### 4.1.3 Inference

**Gene Trees and Species Tree.** The methods for tree inference we presented in Section 1.3.4 mostly rely on models that are well suited for *nonrecombined* loci of the genome, i.e. parts of genes that form a block that was not subject to recombination. The genetic history of such a locus can always be represented as a tree (Maddison, 1997). In a given genetic sequence, we can find many of those nonrecombined loci, and we can infer a so-called *gene tree* for each of them. All those trees are unlikely to be exactly concordant. The *species tree* (or phylogenetic tree) can be seen as a hull that constrains the form of all the gene trees. It represents the branching pattern induced by *speciation* events (Maddison, 1997).

This can be seen using the *coalescent* theory (Kingman, 1982), that can describe the probability distribution of a gene tree. In its simplest form, this model, that is backward in time (from the tips to the root), allows for coalescence events to happen with an exponential law (see e.g. Wakeley, 2009, for an introduction). The genes trees are assumed to follow such a process, but constrained by the species tree, as two lineages in two different branches of the species tree cannot merge. See Figure 4.1.3 for an example.

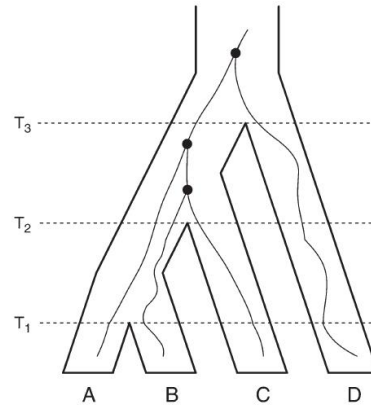


Figure 4.1.3 – Incomplete Lineage Sorting (ILS). The species tree (hull) constrains the form of the gene tree (line): for instance, lineages from *A* and *B* cannot coalesce before time  $T_1$ . However, they can still coalesce after time  $T_2$ , as shown here, so that *B* and *C* merge before *A*. This example of ILS makes the gene tree  $((A,(B,C)),D)$  (in Newick format) discordant with the species tree  $((A,B),C),D)$ . Figure taken from [Kubatko \(2009\)](#).

**Incomplete Lineage Sorting (ILS).** One source of discordance between the gene trees and the species tree is Incomplete Lineage Sorting (ILS, also referred to as deep coalescence). As presented in [Figure 4.1.3](#), this phenomenon happens when two lineages do not coalesce in their most recent common ancestral branch, but instead in an older branch where a third lineage has already joined them. This is perfectly explained by the coalescent theory, and can produce some incongruent gene trees. Given a species tree, the gene tree distribution under this coalescent process has been derived by [Degnan & Salter \(2005\)](#).

**Phylogenetic Networks and ILS.** If the species are not related by a phylogenetic tree, but by a phylogenetic network instead, the same model can be used. However, its combinatorial complexity explodes with the number of hybrids, as gene trees can “turn right or left” at each reticulation. See [Figure 4.1.4](#) for an example.

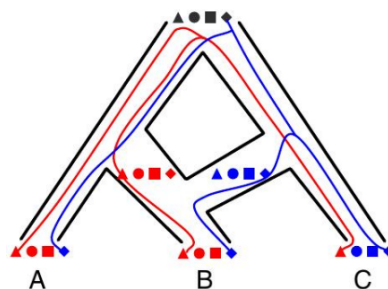


Figure 4.1.4 – ILS on a phylogenetic network. Four independent loci (shapes) are illustrated. Two gene trees are depicted for the triangle and diamond loci. Both agree on a same tree shape  $(A,(B,C))$ . However, the red tree is obtained through ILS, whereas the blue one is due to hybridization. Figure slightly modified from [Yu et al. \(2014\)](#).

Both ILS and reticulation events are hence a source of discordance among gene trees. Ignoring one source or another leads to inconsistent inferences ([Kubatko, 2009](#); [Solís-](#)

Lemus et al., 2016). Similarly to the tree case, the gene tree distribution under this coalescent process given a species network can be derived (Kubatko, 2009).

**Phylogenetic Network Inference.** From the model presented above, it is possible to design a maximum likelihood method to infer a species phylogenetic network from a number of gene trees. A rigorous framework has been derived by Yu et al. (2012, 2014), and implemented in the stand-alone software PhyloNet. Because of the combinatorial complexity of the problem, it is however difficult to infer trees with more than 10 tips, and 4 hybridization events (Solís-Lemus & Ané, 2016). To tackle this computational issue, Solís-Lemus & Ané (2016) proposed a pseudo-likelihood approach, computed assuming that all the quartets are independent from one another. Yu & Nakhleh (2015) described a similar approach, but based on triplets. These relaxations improve the size of the datasets that can be analysed, but the number of hybridization events, as well as the number of species, that can be dealt with is still quite limited (in their article, Solís-Lemus & Ané 2016 deal with networks with at most 15 species, and 3 hybridization events).

**Dating.** In section 1.3.3, we saw that expressing the branch length in real elapsed time was a difficult problem for phylogenetic trees. As expected, this problem does not get easier for phylogenetic networks, and is in fact still an open problem. This will hamper our attempts to apply the models of continuous traits evolution that we describe in the remaining of the section to a biological dataset.

## 4.2 Continuous Trait Evolution on a Network

As in previous chapters, we will assume from now on that the network is known, along with all branch lengths and inheritance probabilities. Given that, as we saw in the previous section, the state of the art methods are still quite limited, this is a strong assumption, but we are confident that these kind of phylogenetic networks will improve and get larger in the coming years.

To model the evolution of a trait on the network, we adopt a framework similar to the one presented in the introductory chapter (Section 1.2.1). The only difference is that, in addition to a process describing the dynamic of the trait and to the split rule after speciation, we need here an extra merging rule for hybridization. It is more difficult to make this merging rule generic, as it will depend on the parameters of the process at hand. Even when the process is fixed, there might be several natural candidate merging rules, each giving birth to a different joint law of the observations at the tips. In the following, we focus on the Brownian Motion process, with a weighted average merging rule.

### 4.2.1 Brownian Motion on a Network

We use the following formal definition of a BM on a network, inspired by Definition 1.2.1 of Section 1.2.1.

**Definition 4.2.1** (Brownian Motion on a Network). Let  $\mathcal{N} = (N, \phi)$  be a phylogenetic network as defined in 4.1.2, with  $N = (E, V)$  a rooted directed network. Assume that each edge  $e \in E$  of the network has an associated branch length  $\ell_e$ , and inheritance



probability  $\gamma_e$ . Given a preorder numbering of the vertices, denote by  $(X_i)_{1 \leq i \leq |V|}$  the sequence of random variables, taking its values in  $\mathbb{R}$ , describing the trait of each node. The law of  $(X_i)_{1 \leq i \leq |V|}$  is recursively defined by:

- $X_1 \sim \mathcal{N}(\mu, s^2)$ : the root is Gaussian with mean  $\mu$  and variance  $s^2$ . It can also be taken fixed ( $s^2 = 0$ ).
- Let  $e \in E$  be a branch, with child node  $i$ , and parent node  $\text{pa}(i)$ . On this branch, the traits evolve according to a Brownian Motion  $(W_t^e, 0 \leq t \leq t_e)$  with variance  $\sigma^2$ , independently from other species, conditionally on  $W_0^e = X_{\text{pa}(i)}$ .
- At node  $i$ , define:
  - $X_i = W_{\ell_e}^e$  if  $i$  is a *tree* node with only one parent edge  $e$ .
  - $X_i = \gamma_{e_a} W_{\ell_{e_a}}^{e_a} + \gamma_{e_b} W_{\ell_{e_b}}^{e_b}$  if  $i$  is an *hybrid* node, with parent edges  $e_a$  and  $e_b$ .
- Iterate down the network.

An example of such a process is presented Figure 4.2.2.

*Remark 4.2.1* (Non-zero hybrid branch lengths). Note that here, we use a network with hybrid edges having a non-zero length ( $\ell_{e_a}$  and  $\ell_{e_b}$  in the definition). This might seem contradictory, as we defined hybridization events as happening between contemporary species. These apparently impossible gene flows might appear for two reasons. First, as we saw earlier, the dating of the network is itself a difficult and still open issue, so the branch length might not be accurate. Second, even if all the branch lengths are known without error, this kind of event might still appear on the network, due to extinct or un-sampled species (see Fig. 4.2.1). It is straightforward to show that it would be equivalent to define the BM on the “original”, complete network with only zero-length hybrid branches, and with all extinct species.

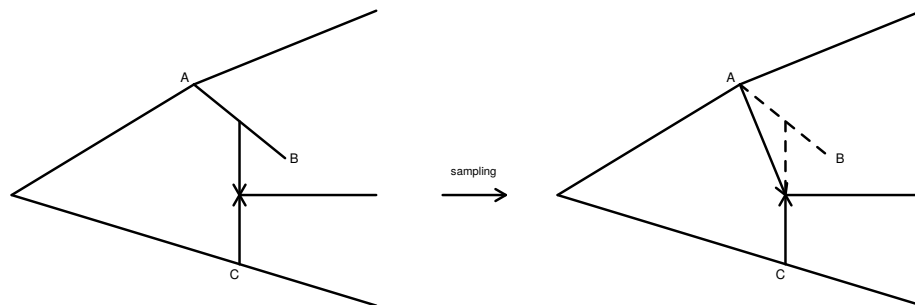


Figure 4.2.1 – Non-zero hybrid branch length case. Species B goes extinct or is un-sampled. In the sampled network, everything happens as if the event involved non-contemporary species A and C.

The previous formal definition 4.2.1 is given to make the link with Section 1.2.1, but is not straightforward to use as such. As the underlying process is a Brownian Motion, it is straightforward to derive the following alternative characterization of the process.

**Proposition 4.2.1** (BM on a Network). *Using the notations of Definition 4.2.1, the joint law of  $\mathbf{X} = (X_i)_{1 \leq i \leq |V|}$  can also be described as follow:*

- $X_1 \sim \mathcal{N}(\mu, s^2)$ : the root is Gaussian with mean  $\mu$  and variance  $s^2$ . It can also be taken fixed ( $s^2 = 0$ ).
- At node  $i$ , define:
  - If  $i$  is a tree node, with parent node  $a$  and parent branch  $e_a$ , take:

$$X_i = X_a + \sqrt{\ell_{e_a}} \epsilon_a,$$

with  $\epsilon_a \sim \mathcal{N}(0, 1)$  a standard normal variable, independent from all other node variables.

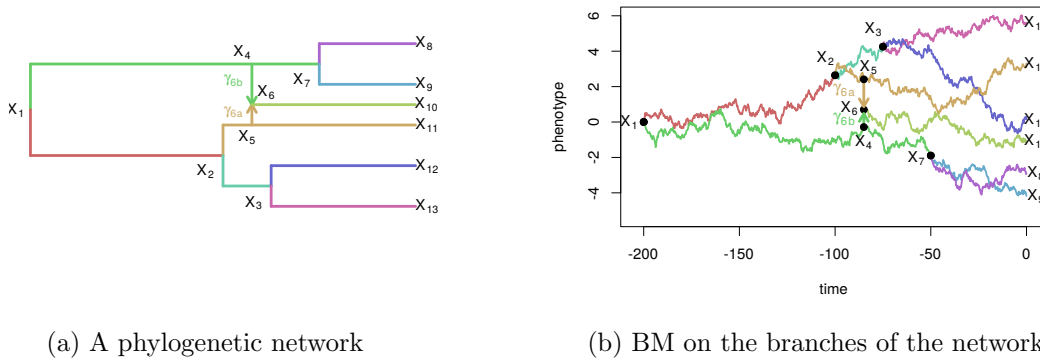
- If  $i$  is an hybrid node, with parent nodes  $a$  and  $b$ , and parent branches  $e_a$  and  $e_b$  take:

$$X_i = \gamma_{e_a} \left( X_a + \sqrt{\ell_{e_a}} \epsilon_a \right) + \gamma_{e_b} \left( X_b + \sqrt{\ell_{e_b}} \epsilon_b \right)$$

with  $\epsilon_a$  and  $\epsilon_b$  two independent standard normal variables, independent from all other node variables.

- Iterate down the network.

This recursive definition is correct provided that the nodes are ordered in a preorder.



(a) A phylogenetic network

(b) BM on the branches of the network

Figure 4.2.2 – Realization of a univariate BM process (with  $\mu = 0$  and  $\sigma^2 = 0.04$ ) on a calibrated network. The colors of the branches (left) match with the colors of the distinct processes (right). Only tip values are observed (at time  $t = 0$ ). For simplicity reasons, the two hybrid branches were chosen to have a length equal to 0, but this need not to be the case (see Fig.4.2.1). Inheritance probabilities at the hybridization event are  $\gamma_{6a}$  and  $\gamma_{6b}$ , with  $\gamma_{6a} + \gamma_{6b} = 1$ .

**Biological Interpretation.** We already discussed the biological interpretation of a BM on a tree in the introduction (see Section 1.4.1.3). The basic idea is that the BM can represent the drift of a polygenic trait, that is, a trait which value depends on the additive expression of many genes. With this interpretation in mind, it is natural to think that the trait on a hybrid, that inherits a proportion  $\gamma_a$  of its genetic material from a parent  $a$ , and  $\gamma_b = 1 - \gamma_a$  from its other parent  $b$ , is a weighted average of the traits of its two parents. This is the assumption we make for our null model. We will see in Section 4.2.3 how this model can be refined, taking into account some of the non standard effects that might occur during a hybridization.

**Possible Other Merging Rules.** The merging rule we chose (weighted average) is nice from a mathematical point of view, as it allows us to carry all the computations analytically. However, that is not the only merging rule one could think of for a BM. For instance, another possible merging rule could be to choose the trait of one of the parents, with a probability given by the inheritance probability parameters  $\gamma$ . This model could be suited to model traits that would inherit all their coding genes from only one of their two parents. However, it introduces a discrete law in the stochastic process, and hence this model is more difficult to study. We will not look further into that direction here, but it could be a starting point for new developments.

### 4.2.2 Joint Law

For the BM on a tree, we were able to describe the joint law of the trait values at all the nodes of the tree, with a covariance between node  $i$  and  $j$  equal to  $\sigma^2 t_{ij}$ , where  $t_{ij}$  is the time elapsed between the root and the most recent common ancestor (mrca) of  $i$  and  $j$  (see Equation 1.4 in Section 1.4.1.1). Our goal here is to derive a similar expression for a trait evolving on a network. An intuition of how this is going to work can be given by re-writing  $t_{ij}$  as:

$$t_{ij} = \sum_{e \in p_i \cap p_j} \ell_e \quad (4.1)$$

where  $p_i$  is is path going from the root to node  $i$ :

$$p_i = \{e \in E : \text{pa}(e) \in \text{anc}(i)\}$$

where  $\text{pa}(e)$  is the parent node of edge  $e \in E$ , and  $\text{anc}(i)$  is the set of all ancestor nodes of  $i$  (as defined for a tree in Section 1.1.1). This formula just expresses that the variance is proportional to the total length of the path shared between the two nodes, that precisely ends at the mrca. On a network, the difficulty is coming from the fact that there is *not a unique path* going from the root to a given node. Indeed, if there is a hybrid among the ancestors of node  $i$ , then a path might go “right” or “left” of the hybrid loop to go from the root to  $i$ .

The general formula was first derived by [Pickrell & Pritchard \(2012\)](#) in the context of population genomics. It is similar to Equation (4.1), just summing over all possible paths, each weighted by their inheritance probabilities.

**Proposition 4.2.2** (Variance Matrix [Pickrell & Pritchard, 2012](#)). *Assume that  $\mathbf{X}$  is the random vector of the traits at the nodes of a network, as defined in 4.2.1 (with a fixed root). Then its variance matrix is equal to  $\sigma^2 \mathbf{V}$ , with:*

$$V_{ij} = \sum_{\substack{p_i \in \mathcal{P}_i \\ p_j \in \mathcal{P}_j}} \left( \prod_{e \in p_i} \gamma_e \right) \left( \prod_{e \in p_j} \gamma_e \right) \sum_{e \in p_i \cap p_j} \ell_e$$

where  $\mathcal{P}_i$  denotes the set of all the paths going from the root to node  $i$ .

This closed formula is compact, and can help us understand the problem, but it is not practical to compute. Using the characterization of the process given in 4.2.1, we can derive an iterative way to compute this covariance matrix, in just one traversal of the tree.

**Proposition 4.2.3.** *Assume that  $\mathbf{X}$  is the random vector of the traits at the nodes of a network, as defined in 4.2.1 (with a fixed root). Let  $i \in V$  be a node of the network. Then:*

- If  $i = 1$  then  $i$  is the root, and  $V_{ii} = 0$ .
- If  $i$  is a tree node, denote by  $a$  the parent of  $i$ , and by  $\ell_{e_a}$  the length of the branch  $e$  going from  $a$  to  $i$ . Then:

$$\begin{cases} V_{ij} = V_{aj} & \forall 1 \leq j \leq i-1 \\ V_{ii} = V_{aa} + \ell_{e_a} \end{cases} \quad (4.2)$$

- If  $i$  is a hybrid node, denote by  $a$  and  $b$  the parents of  $i$ , by  $\ell_{e_a}$  and  $\ell_{e_b}$  the lengths of the branches  $e_a$  and  $e_b$  going from  $a$  or  $b$  to  $i$ , and by  $\gamma_{e_a}$  and  $\gamma_{e_b}$  the associated inheritances probabilities. Then:

$$\begin{cases} V_{ij} = \gamma_{e_a} V_{aj} + \gamma_{e_b} V_{bj} & \forall 1 \leq j \leq i-1 \\ V_{ii} = \gamma_{e_a}^2 (V_{aa} + \ell_{e_a}) + \gamma_{e_b}^2 (V_{bb} + \ell_{e_b}) + 2\gamma_{e_a} \gamma_{e_b} V_{ab} \end{cases} \quad (4.3)$$

*Proof.* Let  $i \geq 2$ , and  $j \leq i$ . Because of the preorder, there is no directed path from  $i$  to  $j$  if  $i \neq j$ . We use the same notations than in the proposition.

- If  $i$  is a tree node, then  $X_i = X_a + \epsilon$ , with  $\epsilon \sim \mathcal{N}(0, \ell_{e_a})$ ,  $\epsilon$  independent of the values  $X_j$  in the current network ( $j < i$ ). Then:

$$\text{Cov}[X_i; X_j] = \begin{cases} \text{Cov}[X_a; X_j] & \text{if } j < i \\ \text{Cov}[X_a; X_a] + \ell_{e_a} & \text{if } j = i \end{cases}$$

- If  $i$  is a hybrid node, then  $X_i = (\gamma_{e_a} X_a + \gamma_{e_b} X_b) + (\gamma_{e_a} \epsilon_a + \gamma_{e_b} \epsilon_b)$ , with  $\epsilon_k \sim \mathcal{N}(0, \ell_{e_k})$ , and  $\epsilon_k$  independent of the all values  $X_j$  in the current network ( $j < i$ ) for  $k \in \{a, b\}$ . Then:

$$\text{Cov}[X_i; X_j] = \begin{cases} \gamma_{e_a} \text{Cov}[X_a; X_j] + \gamma_{e_b} \text{Cov}[X_b; X_j] & \text{if } j < i \\ \gamma_{e_a}^2 (\text{Cov}[X_a; X_a] + \ell_{e_a}) + \gamma_{e_b}^2 (\text{Cov}[X_b; X_b] + \ell_{e_b}) & \text{if } j = i. \\ + 2\gamma_{e_a} \gamma_{e_b} \text{Cov}[X_a; X_b] \end{cases}$$

This ends the proof, because

$$\begin{cases} \mathcal{P}_i = \{(p_a, e_a) : p_a \in \mathcal{P}_a\} & \text{if } i \text{ is a tree node} \\ \mathcal{P}_i = \{(p_a, e_a) : p_a \in \mathcal{P}_a\} \cup \{(p_b, e_b) : p_b \in \mathcal{P}_b\} & \text{if } i \text{ is a hybrid node} \end{cases}$$

and because, if  $i \neq j$ ,  $p_j$  cannot go through  $i$ , so  $p_j$  cannot go through  $e_a$ , or  $e_a$  or  $e_b$ .  $\square$

**Link Between the Tree and the Network Variance Matrix.** Assume for simplicity reasons that we have a phylogenetic network with only one hybridization event, with transitions probabilities  $\gamma$  and  $1 - \gamma$ . Then the network has only two underlying trees, obtained by setting  $\gamma$  respectively to 0 or 1 (so that all the genes come from one side or another of the hybrid). One natural question to ask is: can we compute the network variance matrix from both the trees variances matrices? As we already know how to compute efficiently these matrices for a tree, that might be an easy way to get the network matrix, without using the algorithm presented above. It is indeed possible to get a simple formula, as stated in the next proposition.

**Proposition 4.2.4.** *Let  $\mathcal{N}$  be a rooted binary network with only one hybridization event. Denote by  $p$  the hybrid node, and by  $s$  the MRCA of  $a$  and  $b$ , that are the direct parents of  $p$  (where  $p$  stand for pit, and  $s$  for source, see [Alexeev & Alekseyev 2016](#)). Assume that the hybrid branch going from  $a$  to  $p$  has weight  $\gamma$  and hybrid branch going from  $b$  to  $p$  has weight  $1 - \gamma$ . Then the variance matrix  $\mathbf{V}(\gamma)$  of the network can be obtained from the variances matrices  $\mathbf{V}(0)$  and  $\mathbf{V}(1)$  of the two underlying subtrees as:*

$$\mathbf{V}(\gamma) = \gamma\mathbf{V}(1) + (1 - \gamma)\mathbf{V}(0) - 2\gamma(1 - \gamma)[t_p - t_s]\mathbf{D}(p) \quad (4.4)$$

where  $t_p$  and  $t_s$  are the times elapsed between the root and nodes  $p$  and  $s$ , and  $\mathbf{D}(p)$  is the matrix of nodes descending of  $p$ : for any two nodes  $i$  and  $j$ ,

$$\mathbf{D}(p)_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ are descendants of } p \\ 0 & \text{otherwise.} \end{cases}$$

This proposition is proven in Appendix 4.B, as a corollary of a more general case. It has a natural form, and one could think of using it for the matrix computation. However, it only deals with the case of a network with only one hybridization event. In the same appendix, we derive the general formula for a level-1 network with any number  $h$  of hybrids. However, this formula involves a sum over all the  $2^h$  underlying trees on the network. It is hence not practical to use, and the algorithm derived above in Proposition 4.2.3 is still the best option to compute the variance matrix of a network. The form we derived here might however be useful for other purposes, such as a more systematic study of the hybridization test described below (Section 4.3.2).

### 4.2.3 Shifted BM and Heterosis

In the model we described, the trait of an ancestor was passed on to its children directly, so that the overall expectation of the trait is constant equal to  $\mu$  for all the nodes of the network. However, similarly to the model we used on a tree, some events might happen, that induce a sudden change in the value of the trait on a given branch. We define the shifted BM as a slightly modified version of Definition 4.2.1.

**Proposition 4.2.5** (Shifted BM on a Network). *Using the notations of Definition 4.2.1, the joint law of  $\mathbf{X} = (X_i)_{1 \leq i \leq |V|}$  is defined as follow:*

- $X_1 \sim \mathcal{N}(\mu, s^2)$ : the root is Gaussian with mean  $\mu$  and variance  $s^2$ . It can also be taken fixed ( $s^2 = 0$ ).
- At node  $i$ , define:

- If  $i$  is a tree node, with parent node  $a$  and parent branch  $e_a$ , take:

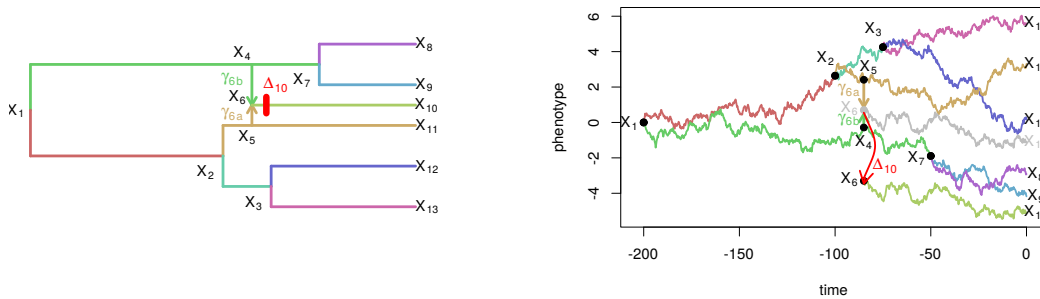
$$X_i = X_a + \sqrt{\ell_{e_a}} \epsilon_a + \Delta_{e_a},$$

with  $\epsilon_a \sim \mathcal{N}(0,1)$  a standard normal variable, independent from all other node variables, and  $\Delta_{e_a}$  a fixed shift value (possibly equal to 0) associated with branch  $e_a$ .

- If  $i$  is an hybrid node,  $X_i$  is defined as in 4.2.1 (no shift are allowed on hybrid branches).
- Iterate down the tree.

This recursive definition is correct provided that the nodes are ordered in a preorder.

*Remark 4.2.2.* In this definition, we forbade shifts on hybrid branches. This can be done without loss of generality. Indeed, a hybrid connects three branches, two in and one out. A shift on any of those three branches would impact the same set of nodes (apart from the hybrid itself), and hence would produce the same data at the tips. The position of a shift on these three branches is consequently not identifiable. By convention, we assume that this shift happens on the branch going out of the hybrid, which is a tree-like branch (it has only one parent and daughter node).



(a) A phylogenetic network with heterosis

(b) BM on the branches of the network

Figure 4.2.3 – Realization of a univariate BM process (with  $\mu = 0$  and  $\sigma^2 = 0.04$ ) on a calibrated network, with heterosis. The shift occurs right after the hybridization event, and changes the trajectory of the BM from the grey one to the colored one.

We present an example of heterosis in Figure 4.2.3. As the shifts are fixed, they do not impact the variance matrix, that remains unchanged. As in the tree case, we expect that most of the shifts  $\Delta_e$  on the branches will be zero. However, and contrary to the tree case, we have some pre-defined candidate branches for those shifts to occur on. Indeed a shift might have a different biological interpretation, depending on where it occurs:

- If it occurs on a “regular” tree-branch, i.e. a branch that is not coming out of a hybrid node, then it has the same interpretation as before. The shift can then represent the effect of a sudden environmental change, that has an impact on the ecological niche of the species studied.

- If it occurs on a branch going out of a hybrid node, then it can be seen as an effect of *heterosis*. Heterosis, or hybrid vigor, is a well documented effect (see, e.g. [Fiévet et al., 2010](#); [Chen, 2013](#), for recent developments, and a review) that happens when two distinct species are hybridized. The hybrid species might then exhibit a shift in some of its traits, making it particularly fit (hybrid vigor) or ill-fit (hybrid depression) to its environment. Such a shift can hence be seen as a component of hybridization: the new species inherits its trait as a weighted average of the traits of its two parents, plus a shift representing heterosis.

*Remark 4.2.3 (Identifiability).* We are not discussing here identifiability problems that are likely to arise, as in the tree case (see Section 2.3 in Chapter 2). A careful analysis of this question should be the focus of future work. However, we won't try to infer the position of the shifts here, and rather impose their position by hand and *a priori*. We hence assume that the user “knows what he's doing” and will not use non-identifiable configurations. When the shift positions are fixed, it is however easy to check that the corresponding regression matrix has full rank (see next section).

We explain in the next section how these shifts can be easily included in the model.

#### 4.2.4 Phylogenetic Regression Model

Now that, given a network, we are able to compute the matrix  $\mathbf{V}$  giving the covariance structure of the observations  $\mathbf{Y}$  at the tips of the network, we can use the linear regression framework, as defined for models on trees in Section 1.4.2.2 of the introduction.

$$\mathbf{Y} = \mathbf{U}\boldsymbol{\theta} + \sigma^2\mathbf{E} \quad \text{with} \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{V}) \quad (4.5)$$

where  $\mathbf{U}$  is a  $n \times q$  matrix of regressors, and  $\boldsymbol{\theta}$  a vector of  $q$  coefficients. In the simple case of a BM with root value fixed equal to  $\mu$ , we simply get  $\mathbf{U} = \mathbf{1}_n$ , and  $\boldsymbol{\theta} = \mu$ . Matrix  $\mathbf{U}$  can also contain some explanatory traits variable of interest. As in Section 2.2.3 (see also Section 1.4.3), we can also include some shifts on the branches of the network, and take them into account using an equivalent of the incidence matrix we used for the tree.

**Definition 4.2.2.** The incidence matrix  $\mathbf{U}$  of a network  $N = (E, V)$  with ordered nodes is defined as, for any two nodes  $i$  and  $j$ :

$$U_{ij} = \sum_{p \in \mathcal{P}_{j \rightarrow i}} \prod_{e \in p} \gamma_e \quad (4.6)$$

where  $\mathcal{P}_{j \rightarrow i}$  is the set of all the paths going from  $j$  to  $i$  (respecting the orientation of the network). Note that, if  $i$  is not a descendant of  $j$ , then  $\mathcal{P}_{j \rightarrow i} = \emptyset$  and  $U_{ij} = 0$ .

Associated with this incidence matrix, we define the vector of the shifts on the branches by identifying the branches with their descending node (when possible), as in the tree case.

**Definition 4.2.3.** The vector  $\boldsymbol{\Delta}$  represents all the shifts on the network, numbered by nodes. For any node  $i$ , if  $i$  is tree-like, then it represents the edge ending at  $i$ . If  $i$  is hybrid, then it artificially represents both the hybrid edges ending at  $i$ . As, for



identifiability reasons, shifts are forbidden on hybrid edges (see Remark 4.2.2), these coefficients will not be used anyway. We get, for any node  $i$ :

$$\Delta_i = \begin{cases} \mu & \text{if } i = 1 \text{ (root node)} \\ \delta_i & \text{if } i \text{ is tree-like and there is a shift on edge ending at } i \\ 0 & \text{if } i \text{ is tree-like and there are no shift on the edge ending at } i \\ 0 & \text{if } i \text{ is a hybrid} \end{cases}$$

note that we put the ancestral mean  $\mu$  of the BM on the fictive branch ending at the root of the tree.

*Remark 4.2.4.* It is straightforward to see that, when the network actually reduces to a tree, then this definition is compatible with the definition of an incidence matrix for a tree we gave in Section 2.2.3. Compared to the incidence matrix of a plain tree, the incidence matrix of a network can have non-binary entries: some of its coefficients are given by the transmission probabilities  $\gamma$ .

Using these definitions, the joint law of a shifted BM on a network can be expressed in the linear model framework. See Example 4.2.1 for the incidence matrix and shift vector associated with Figure 4.2.3.

**Proposition 4.2.6.** *Let  $\mathbf{X}$  be the vector of the traits at the nodes of a network  $N = (E, V)$  with ordered nodes. Assume that  $\mathbf{X}$  is the result of a shifted BM as defined in 4.2.5, with vector of shifts  $\Delta$ . Then its variance matrix is given by  $\sigma^2 \mathbf{V}$ , with  $\mathbf{V}$  defined in 4.2.2, and its expectation by:*

$$\mathbb{E}[\mathbf{X}] = \mathbf{U}\Delta$$

*Proof.* We show this equality recursively. Assume that the nodes are numbered in pre-order (from the root to the tips). Denote by  $\mathbf{U}^i$  the  $i^{\text{th}}$  row-vector of  $\mathbf{U}$ . Then, as the root is its only descendant:

$$\mathbb{E}[X_1] = \mu = \Delta_1 = \mathbf{U}^1 \Delta$$

Then, assuming that the property is true for its parents, let's show it for node  $i$ .

- If  $i$  is tree-like, then denote by  $a$  its unique parent, and by  $e_a$  the edge linking the two nodes. We have, for any node  $k$  above  $i$ :  $\mathcal{P}_{k \rightarrow i} = \{(p_a, e_a) : p_a \in \mathcal{P}_{k \rightarrow a}\}$ , so that, from definition 4.2.2:

$$U_{ik} = \begin{cases} U_{ak} & \forall k \neq i \\ 1 & \text{if } k = i \end{cases}$$

hence

$$\mathbb{E}[X_i] = \mathbb{E}[X_a] + \Delta_i = \mathbf{U}^a \Delta + \Delta_i = \mathbf{U}^i \Delta$$

- If  $i$  is a hybrid, then denote by  $a$  and  $b$  its two parents, by  $e_a$  and  $e_b$  the corresponding edges, with coefficients  $\gamma_{e_a}$  and  $\gamma_{e_b}$ . Then for any node  $k$  above  $i$ , we have:  $\mathcal{P}_{k \rightarrow i} = \{(p_a, e_a) : p_a \in \mathcal{P}_{k \rightarrow a}\} \cup \{(p_b, e_b) : p_b \in \mathcal{P}_{k \rightarrow b}\}$ , so that, from definition 4.2.2:

$$U_{ik} = \begin{cases} \gamma_a U_{ak} + \gamma_b U_{bk} & \forall k \neq i \\ 1 & \text{if } k = i \end{cases}$$

hence, as no shift can occur on the hybrid branches ( $\Delta_i = 0$  by convention):

$$\mathbb{E}[X_i] = \gamma_{e_a} \mathbb{E}[X_a] + \gamma_{e_b} \mathbb{E}[X_b] = \gamma_{e_a} \mathbf{U}^a \Delta + \gamma_{e_b} \mathbf{U}^b \Delta = \mathbf{U}^i \Delta$$

□

*Remark 4.2.5.* Note that the proof above also gives us a way of computing the incidence matrix of a network in just one preorder traversal of the network.

Keeping only the tips of the network, it is straightforward to get the following corollary.

**Corollary 4.2.1.** *Using the same notations as in Proposition 4.2.6, let  $\mathbf{T}$  be the matrix of  $\mathbf{U}$  with only lines corresponding to tips of the network. Then the expectation of the vector of traits  $\mathbf{Y}$  at the tips of the tree is given by:*

$$\mathbb{E}[\mathbf{Y}] = \mathbf{T}\Delta.$$

*Example 4.2.1 (Incidence Matrix and Shift Vector).* The incidence matrix  $\mathbf{T}$  associated with the network presented Figure 4.2.3 is given by:

$$\mathbf{U} = \begin{array}{c} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_7 \\ X_8 \\ X_9 \\ X_{10} \\ X_{11} \\ X_{12} \\ X_{13} \end{array} \begin{array}{c} X_1 \\ X_2 \\ X_3 \\ X_4 \\ X_5 \\ X_7 \\ X_8 \\ X_9 \\ X_{10} \\ X_{11} \\ X_{12} \\ X_{13} \end{array} \left( \begin{array}{cccccccccccc} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \hline 1 & \cdot & \cdot & 1 & \cdot & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & 1 & \cdot & 1 & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \gamma_{6b} & \gamma_{6a} & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot \\ 1 & 1 & \cdot & \cdot & 1 & \cdot & \cdot & \cdot & \cdot & 1 & \cdot & \cdot & \cdot \\ 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 & \cdot \\ 1 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \end{array} \right) \mathbf{T}$$

(where zeros are replaced with dots to improve readability). The associated shift vector and expectation vector at the tips are:

$$\Delta = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 7 \\ 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \left( \begin{array}{c} \mu \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \Delta_{10} \\ \cdot \\ \cdot \\ \cdot \end{array} \right) \quad \mathbf{T}\Delta = \begin{array}{c} 8 \\ 9 \\ 10 \\ 11 \\ 12 \\ 13 \end{array} \left( \begin{array}{c} \mu \\ \mu \\ \mu + \Delta_{10} \\ \mu \\ \mu \\ \mu \end{array} \right)$$

Note that node  $X_6$ , that is a hybrid node, is excluded from the matrix and shift vectors (because shifts on hybrid branches are excluded, see Remark 4.2.2).

### 4.3 Tests of Phylogenetic Signal

The phylogenetic linear regression framework gives us all the tools needed to test some hypothesis about the distribution of the traits observed at the tips of a network. We describe here two tests on the variance structure, and one on the expectation structure, to test for the impact, respectively, of the network and of heterosis.

#### 4.3.1 Pagel's $\lambda$ on a Network

Recall that Pagel's  $\lambda$  transform (Pagel, 1999) could be seen as a modification of the branch lengths of a tree, giving more importance to external edges, i.e. edges that are leading to a tip (see Section 1.4.2.3 in the introduction). We want to define here the same transformation on a network. For a network with consistent branch lengths representing real time, all the paths going from the root to a given node  $i$  have the same length, that can be defined as:

$$t_i = \frac{1}{|\mathcal{P}_i|} \sum_{p_i \in \mathcal{P}_i} \sum_{e \in p_i} \ell_e = \sum_{e \in p_i} \ell_e \quad , \quad \forall p_i \in \mathcal{P}_i$$

Using this definition, Pagel's  $\lambda$  transformation on the branch lengths is the same as in the tree case.

**Definition 4.3.1** (Pagel's  $\lambda$  transform of the branch lengths). Let  $e$  be a branch of the network, with child node  $i$ , parent node  $\text{pa}(i)$ , and length  $\ell_e$ . Then its transformed length is given by:

$$\ell_e(\lambda) = \begin{cases} \lambda \ell_e & \text{if } i \text{ is an internal node} \\ \ell_e + (1 - \lambda)t_{\text{pa}(i)} = \lambda \ell_e + (1 - \lambda)t_i & \text{if } i \text{ is a tip} \end{cases} \quad (4.7)$$

The interpretation is then similar: when  $\lambda$  decreases to zero, the network structure is less and less important, until it's completely gone for  $\lambda = 0$ , and all the tips are independent. However, its impact on the matrix  $\mathbf{V}$  is not completely similar, and cannot be written in the simple form we used on a tree (see Equation 1.11 in the introduction). The structure matrix of the variance  $\mathbf{V}(\lambda)$  of such a transformed network is given by the following proposition.

**Proposition 4.3.1** (Pagel's  $\lambda$  effect on the variance). *The structure of the variance matrix of a BM running on a network transformed by a parameter  $\lambda$  is given by:*

$$\begin{cases} V(\lambda)_{ij} = \lambda V_{ij} & \text{for any two nodes } i \text{ and } j, i \neq j \\ V(\lambda)_{ii} = \lambda V_{ij} & \text{for any internal node } i \\ V(\lambda)_{ii} = \lambda V_{ii} + (1 - \lambda)t_i & \text{for any tree tip } i \\ V(\lambda)_{ii} = \lambda V_{ii} + (\gamma_{e_a}^2 + \gamma_{e_b}^2)(1 - \lambda)t_i & \text{for any hybrid tip } i \text{ with parent branches } e_a \text{ and } e_b \end{cases}$$

*Proof.* From the general formula given in Proposition 4.2.2, the first two equations (for non-diagonal elements and internal nodes) are straightforward (all the branch lengths included in the paths of the sum are multiplied by  $\lambda$ ). Let's now prove the last two. Take  $i$  a tip node of the network.

- If  $i$  is a tree node, with parent node  $a$  and parent branch  $e_a$ , then, from the recursive formula (4.2), its variance is proportional to:

$$V_{ii}(\lambda) = V(\lambda)_{aa} + \ell_{e_a}(\lambda) = \lambda V_{aa} + \lambda \ell_{e_a} + (1 - \lambda)t_i = \lambda V_{ii} + (1 - \lambda)t_i$$

- If  $i$  is a hybrid tip, with parent nodes  $a$  and  $b$ , and parent branches  $e_a$  and  $e_b$ , then, from the recursive formula 4.3, its variance is proportional to:

$$\begin{aligned} V(\lambda)_{ii} &= \gamma_{e_a}^2 (V(\lambda)_{aa} + \ell_{e_a}(\lambda)) + \gamma_{e_b}^2 (V(\lambda)_{bb} + \ell_{e_b}(\lambda)) + 2\gamma_{e_a}\gamma_{e_b} V(\lambda)_{ab} \\ &= \lambda V_{ii} + (\gamma_{e_a}^2 + \gamma_{e_b}^2)(1 - \lambda)t_i \end{aligned}$$

□

The extra-diagonal elements are still just multiplied by  $\lambda$  (as in Equation 1.11). But, the variance of a trait is impacted by any hybridization event that happened in its history. So, when reducing the impact of the network structure, the variances are going to change too, in opposition with the tree case. From these equations, we can see that any ancestral hybridization event is forgotten (which is coherent with the “star network” representation), but that present day (tips) hybridizations remain. It is not clear however whether “hybrid tips” should be allowed or not.

Once the model is defined, we can use the same method as in the tree case to test for “phylogenetic signal”, using a maximum-likelihood ratio test for  $\mathcal{H}_0 : \lambda = 0$  (no structure) vs  $\mathcal{H}_1 : \lambda > 0$ . There are no closed form estimate for the  $\lambda$  parameter, and a numerical optimization needs to be carried out. The interpretation of this test is similar to the one given in the tree case (Section 1.4.2.3).

A systematic study of the features of this test is still to be carried on. Because the generalization is quite straightforward, it is natural to think that this test will have the same strengths and flaws as its equivalent on phylogenetic trees.

## 4.3.2 Can we Detect Hybridization Events ?

### 4.3.2.1 Description of a Hybridization Effect Test

In the previous section, we tested whether the data contained any structure at all. We might want to be more precise, and test the form of this structure. Namely, we could want to test whether a tree is enough to explain the data, against the full network. In other words, such a test would give us an indication of whether the trait we are studying was impacted or not by the hybridization events. Indeed, the “inheritance probability” coefficients  $\gamma$  associated with an hybridization event just gives us roughly the proportion of the genome that the hybrid inherits from its parents. It might happen that all the genes coding for the trait we are studying are all on the “same side” of the genome, and hence that the trait does not “see” the hybridization, and behaves as if the transmission were purely vertical.

On a network with several hybridization events, testing for all one by one would not be practical. To keep things simple, in the same spirit as Pagel’s  $\lambda$  transformation, we introduce a single tuning parameter, that controls all the hybridization events.

**Definition 4.3.2** (Network Tuning Parameters  $\lambda$ ). Take a rooted phylogenetic network  $\mathcal{N}$  with  $h$  hybrids. Denote by  $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_h)$  the vector of *minor* coefficients for each hybrid (i.e.  $\gamma_i \leq 1/2, \forall 1 \leq i \leq h$ ). The phylogenetic model is then defined by the couple  $(\mathcal{N}, \boldsymbol{\gamma})$  of a network topology and minor inheritance probabilities. Indeed, all other inheritance probabilities are either 1 (for tree edges) or  $1 - \gamma_{\text{mate}(e)}$ , where  $\text{mate}(e)$  is the minor parent leading to the same hybrid node as the major hybrid edge  $e$ . The tuning parameter  $\lambda$  acts on all the minor hybrids in the following way:

$$\mathcal{N}(\lambda) = (\mathcal{N}, \lambda\boldsymbol{\gamma})$$

The parameter  $\lambda$  quantifies the importance of the hybridization events on the trait:

- If  $\lambda = 1$  then the original inferred network is unchanged.
- If  $\lambda = 0$  then the model is the *major tree* extracted from the network: the hybridization have no influence on the trait.
- If  $\lambda < 0$  or  $\lambda > 1$ , then the trait of the hybrid might go outside of the interval of the traits of its two parents. This might be one way to model heterosis (but see next section for a more natural way).

The structure matrix  $\mathbf{V}(\lambda\boldsymbol{\gamma})$  can be obtained with the same recursive algorithm as before, for any fixed  $\lambda$ . When there is only one hybridization event in the network, some closed form formulas can be derived (see Appendix 4.B). In any case, a numerical optimization on  $\lambda$  allows us to use the maximum likelihood ratio test, the same way we did for Pagel's test. Here, we can take  $\mathcal{H}_0 : \lambda = 0$  (the tree is enough) vs  $\mathcal{H}_1 : \lambda \neq 0$  (some events occurred at the hybridizations). An alternative way could be to take the inferred network as the null ( $\mathcal{H}_0 : \lambda = 1$ ).

#### 4.3.2.2 Empirical Power Study

To have an idea of how this test behaves, we did an empirical study of its power for some given fixed networks. We chose four different networks, all sharing the same backbone symmetric phylogenetic tree with 32 tips and total height 1. We added 1 to 8 hybridization events. In all scenarios, the number of tips that have a hybrid in their ancestors is fixed equal to 8. The network used, along with the empirical results on 100,000 simulations are shown Figure 4.3.1.

The main conclusion is that the test has a very small power, that is hardly above the imposed level. This means that the test is not efficient at all in detecting hybridization effects on the traits, and never rejects the null hypothesis, even when the network is strongly marked ( $\gamma = 0.5$ ). The only case where the power seems to be a little bit less miserable is for the network with many recent hybridization events (leftmost, see Fig. 4.3.1). This would hint that recent hybridization events have a stronger impact on a continuous trait.

The fact that this test fails is not that surprising. Indeed, we are trying to detect the fine effects induced by hybridization events on the variance matrix structure, with only one trait observed at the tips of the network. One natural thing to do would be to study the effect of the number of (independent) traits observed at the tips on the power of this test. Because most dataset do not have such independent traits observations, and because the test on heterosis presented below revealed more powerful and natural, we did not push this analysis further in the present work.

### 4.3.3 Test of Heterosis

#### 4.3.3.1 Description of the Test

In this section, we use the phylogenetic linear model defined in (4.5), and use the incidence matrix defined in 4.2.2 to model heterosis as shifts on branches that come out of a hybrid.

Assume that the network has  $h$  hybrids, and denote by  $\mathbf{N}$  the submatrix of the incidence matrix  $\mathbf{T}$  that has  $n$  rows corresponding to the tips, and  $h$  columns corresponding

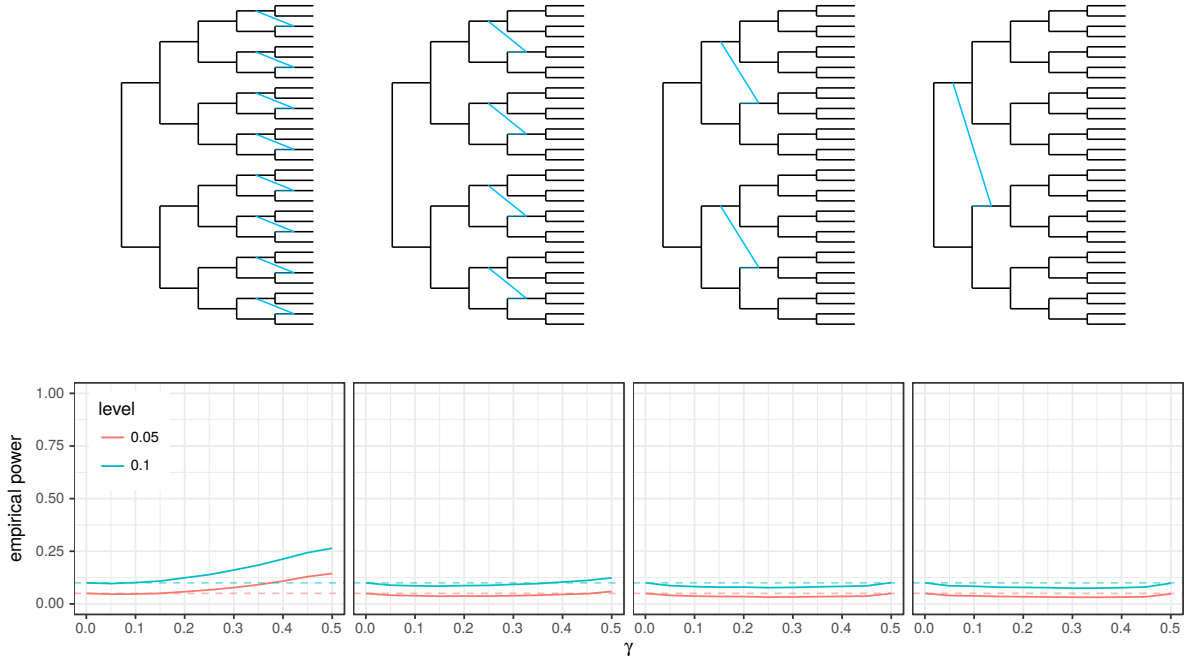


Figure 4.3.1 – Empirical power of the LRT of hybridization, for four different network topologies (shown above), and two different test levels (colors). All hybridization events on the original networks had minor weights  $\gamma$ . The empirical distribution of the statistic under  $\mathcal{H}_0$  (no hybridization event,  $\gamma = 0$ ) and various alternative ( $0 < \gamma \leq 0.5$ ) were obtained on 100,000 simulations. The mean and variance of the BM were fixed, respectively, to 0 and 1. Dotted lines show the levels of the test, that should be below the power.

to the  $h$  branches linking a hybrid to its child. Further define  $\tilde{\mathbf{N}}$  the column vector with size  $n$  containing the row sums of  $\mathbf{N}$ : for any tip  $i$ ,  $\tilde{N}_i = \sum_{k=1}^h N_{ik}$ . Then, from Proposition 4.2.6, the shifted BM can be written as:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \tilde{\mathbf{N}}b + \mathbf{N}\mathbf{d} + \mathbf{E} \quad , \quad \mathbf{d} \text{ s.t. } \sum_{k=1}^h d_k = 0 \quad , \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{V}) \quad (4.8)$$

where  $\mathbf{X}$  is a given matrix of regressors, with associated coefficients  $\boldsymbol{\beta}$ . For a simple BM with initial root value  $\mu$ , we can take  $\mathbf{X} = \mathbf{1}_n$  and  $\boldsymbol{\beta} = \mu$ .

When written this way, the problem of testing for heterosis just amounts to testing the fixed effects  $b$  and  $\mathbf{d}$ . Indeed, we can write the following assumptions on the hybrids events, with their counterparts in our linear model framework.

	Hypotheses	Linear Model
$\mathcal{H}_0$	No heterosis	$b = 0$ and $\mathbf{d} = \mathbf{0}$
$\mathcal{H}_1$	Single effect heterosis	$b \neq 0$ and $\mathbf{d} = \mathbf{0}$
$\mathcal{H}_2$	Multi effect heterosis	$b \neq 0$ and $\mathbf{d} \neq \mathbf{0}$

Hypothesis  $\mathcal{H}_0$  corresponds to the null model where the trait of the hybrids are just inherited from their parent's through a weighted average.  $\mathcal{H}_1$  is the case where all hybridization events result in the same heterosis effect, the trait being shifted by a

coefficient  $b$ . Finally,  $\mathcal{H}_2$  is the case where each hybridization event  $k$  is impacted by its own heterosis event, with a shift  $b + d_k$ .

These tests on fixed effects are very classic in the statistics literature (see e.g. [Lehman, 1986](#); [Searle, 1987](#)). The tests we write are all exact and uniformly most powerful among all invariant tests. They are also admissible against all invariant alternatives ([Lehman, 1986](#)). In our case, we can write the following two Fisher statistics:

$$F_{10} = \frac{\|\mathbf{Y} - \text{Proj}_{\mathbf{X}} \mathbf{Y}\|_{\mathbf{V}^{-1}}^2 - \|\mathbf{Y} - \text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]} \mathbf{Y}\|_{\mathbf{V}^{-1}}^2}{\|\mathbf{Y} - \text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]} \mathbf{Y}\|_{\mathbf{V}^{-1}}^2} \quad (4.9)$$

$$F_{21} = \frac{\|\mathbf{Y} - \text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]} \mathbf{Y}\|_{\mathbf{V}^{-1}}^2 - \|\mathbf{Y} - \text{Proj}_{[\mathbf{X} \ \mathbf{N}]} \mathbf{Y}\|_{\mathbf{V}^{-1}}^2}{\|\mathbf{Y} - \text{Proj}_{[\mathbf{X} \ \mathbf{N}]} \mathbf{Y}\|_{\mathbf{V}^{-1}}^2} \quad (4.10)$$

where  $\text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]}$  denotes the projection on the linear space spanned by the columns of matrices  $\mathbf{X}$  and  $\bar{\mathbf{N}}$ . These statistics follow a noncentral Fisher distribution:

$$\frac{N - r_{[\mathbf{X} \ \bar{\mathbf{N}}]}}{r_{[\mathbf{X} \ \bar{\mathbf{N}}]} - r_{\mathbf{X}}} F_{10} \sim \mathcal{F} \left( r_{[\mathbf{X} \ \bar{\mathbf{N}}]} - r_{\mathbf{X}}, N - r_{[\mathbf{X} \ \bar{\mathbf{N}}]}, \frac{b^2}{2\sigma^2} \|(\mathbf{I} - \text{Proj}_{\mathbf{X}}) \bar{\mathbf{N}}\|_{\mathbf{V}^{-1}}^2 \right) \quad (4.11)$$

$$\frac{N - r_{[\mathbf{X} \ \mathbf{N}]}}{r_{[\mathbf{X} \ \mathbf{N}]} - r_{[\mathbf{X} \ \bar{\mathbf{N}}]}} F_{21} \sim \mathcal{F} \left( r_{[\mathbf{X} \ \mathbf{N}]} - r_{[\mathbf{X} \ \bar{\mathbf{N}}]}, N - r_{[\mathbf{X} \ \mathbf{N}]}, \frac{1}{2\sigma^2} \|(\mathbf{I} - \text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]}) \mathbf{N} \mathbf{d}\|_{\mathbf{V}^{-1}}^2 \right) \quad (4.12)$$

where  $r_{[\mathbf{X} \ \mathbf{N}]}$  is the rank of the matrix obtained by pasting the columns of  $\mathbf{X}$  and  $\mathbf{N}$  together. The noncentral coefficient depends on the network topology, through the metric defined by  $\mathbf{V}$ , and through the regression matrix  $\mathbf{N}$ . We will study it for several symmetric networks in the following section.

As noted previously, we did not study the identifiability of such a model carefully. For the tests above to be meaningful, one could expect that the regression matrix  $\mathbf{N}$  has full rank. This verification is left to the user. For small level-1 networks such as the ones that can be inferred by state of the art methods, this is however likely to be the case.

#### 4.3.3.2 Theoretical Power Study

As the distribution of the statistics above are fully known, it is possible to conduct a theoretical study of the power of the two tests of heterosis presented above.

**Test  $\mathcal{H}_0$  vs  $\mathcal{H}_1$ .** We first study the theoretical power to detect a single heterosis effect. The non-central coefficient of the Fisher statistic  $F_{01}$  ([4.11](#)) depends on the size of the effect  $b$ , the variance  $\sigma^2$  of the BM, the topology of the network (through  $\mathbf{N}$  and  $\mathbf{V}$ ), and the value of the inheritance probabilities  $\gamma$  (through  $\mathbf{V}$ ). In the following, we fix the scaling variance factor  $\sigma^2 = 1$ . We use the same symmetric backbone tree with total height 1 as before, and add one single hybridization event at various heights (see [Fig. 4.3.2](#), top line). The leftmost topology has a very recent hybridization event, that only affects one tip, while the rightmost has a very ancient one, that affects 8 tips. We fix the inheritance probability  $\gamma = 0.4$  in all topology, as this parameter revealed to have only but a small impact on the power of the test (not shown). The variation of the power for these four networks with the size of the effect  $b$  is presented [Figure 4.3.2](#) (bottom line) for three test levels.



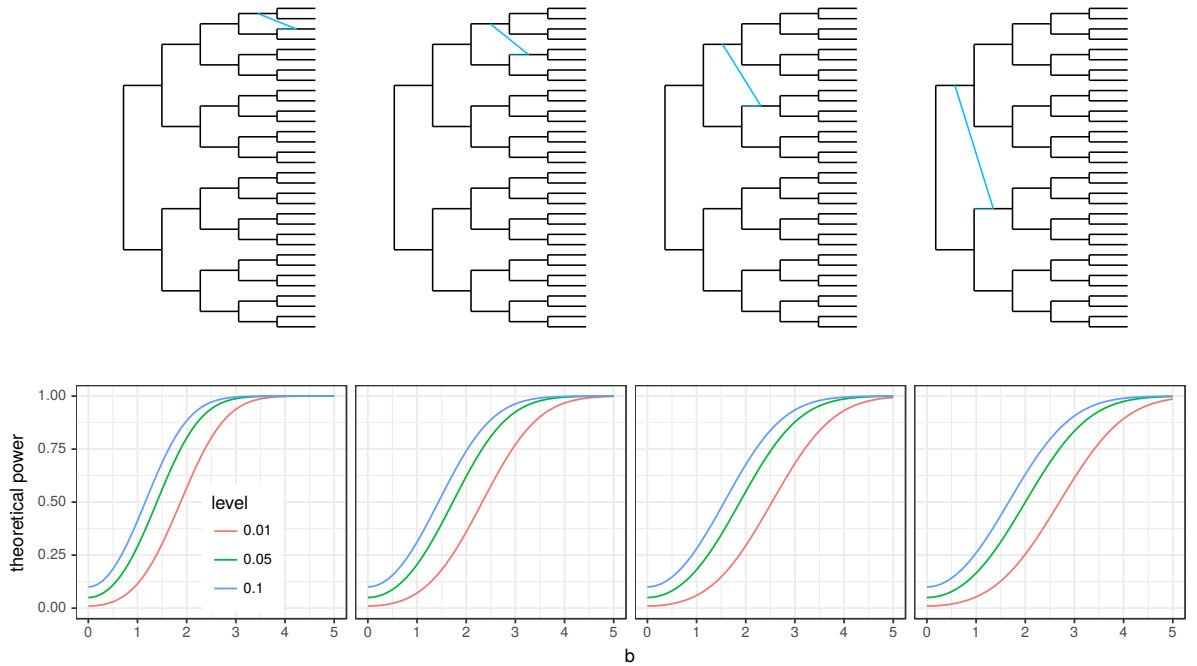


Figure 4.3.2 – Theoretical power of the single effect heterosis test  $\mathcal{H}_0$  vs  $\mathcal{H}_1$ , for four different networks topologies with inheritance probability  $\gamma = 0.4$ , and a BM with variance  $\sigma^2 = 1$ .

As expected, the power improves with the effect size, reaching approximately 1 for  $b = 5$  in all scenarios. In addition, the heterosis effect seems easier to detect for recent hybridization events, even if they affect a lower number of tips. One intuition for that is that ancient hybridization effects are “diluted” by the variance of the BM, and are hence harder to detect, even if they affect more tips.

Note that here, we used a scenario slightly different from the one presented in Figure 4.3.1: each network has only one hybridization event. We used this scenario to stress out the impact of recent hybridization event: even when they are affecting only one tip, their signal is much stronger. Taking the previous scenario, that adds such recent hybridization events in the already favorable case, only strengthen the effect.

**Test  $\mathcal{H}_1$  vs  $\mathcal{H}_2$ .** To study the power of this test, we use the same networks topologies as before, with  $\gamma = 0.4$ , keeping only networks with more than one hybridization event (as we want to test for heterogeneity). The non-central coefficient of the Fisher statistic  $F_{12}$  (4.12) depends on the same parameters as before, and of the heterogeneity of the effects through vector  $\mathbf{d}$ . Here, denoting  $h$  the number of hybrids, and  $h = q \cdot 2 + r$  the euclidean division of  $h$  by 2, we take vector  $\mathbf{d} = b\mathbf{d}^u$  as proportional to unit vector  $\mathbf{d}^u$ , defined as:

$$d_i^u = \begin{cases} 1/f & \text{if } 1 \leq i \leq q \\ -f & \text{if } q < i \leq h \end{cases} \quad \text{with} \quad f = \sqrt{\frac{q}{q+r}}.$$

This form is convenient, as, first, it ensures us that the sum of coefficients is 0, and, second, it allows us to re-write the non-central coefficient in (4.12) as:

$$\frac{1}{2\sigma^2} \|(\mathbf{I} - \text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]})\mathbf{N}\mathbf{d}\|_{\mathbf{V}^{-1}}^2 = \frac{b}{2\sigma^2} \|(\mathbf{I} - \text{Proj}_{[\mathbf{X} \ \bar{\mathbf{N}}]})\mathbf{N}\mathbf{d}^u\|_{\mathbf{V}^{-1}}^2.$$

The heterogeneity is hence controlled by coefficient  $b$ , that we vary between 0 and 5 (fixing  $\sigma^2 = 1$ ).

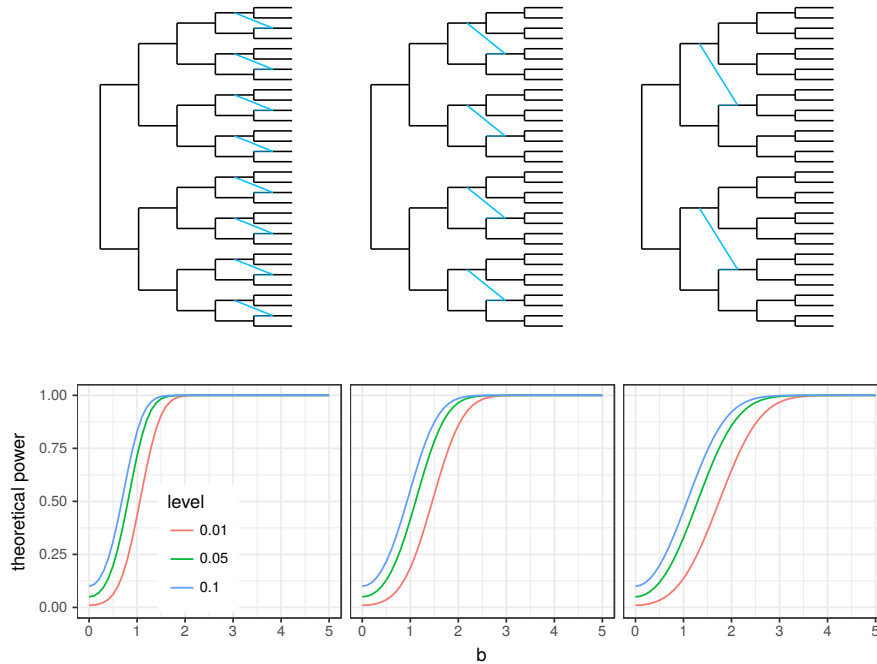


Figure 4.3.3 – Theoretical power of the heterogeneity heterosis effect test  $\mathcal{H}_1$  vs  $\mathcal{H}_2$ , for three different networks topologies with inheritance probability  $\gamma = 0.4$ , and a BM with variance  $\sigma^2 = 1$ .

The results, presented Figure 4.3.3, are similar to the ones commented for the single effect: the test is more powerful for a high heterogeneity coefficient  $b$ , and for recent hybridization event.

## 4.4 The Julia package PhyloNetworks

In this section, we briefly describe the PhyloNetworks package (Solís-Lemus et al., 2017).

**Julia.** Because they are computationally intensive, the methods for phylogenetic network inference need to be implemented in an efficient way. Solís-Lemus & Ané (2016) decided to use the new programming language Julia (Bezanson et al., 2017). Julia is a high-level programming language similar to R, Matlab or Python, that is also high-performance. It is still under rapid development, and a long-lasting stable version is yet to come. With 1465 registered packages (as of July 13, 2017, see [julialang.org](http://julialang.org)), Julia provides an ever growing community of users with an interactive and comprehensive

computing environment. The language and the packages are all developed on GitHub ([github.com](https://github.com)), which makes them open to contributions from any interested researcher.

**Inferring and Manipulating Networks.** Solís-Lemus & Ané (2016) made their method for network inference available through a Julia package `PhyloNetworks` ([github.com/crs14/PhyloNetworks.jl](https://github.com/crs14/PhyloNetworks.jl)). This package was designed to be useful not only for network inference, but also for interactive network manipulation and visualization. This package has for ambition to become the standard tool to deal with phylogenetic networks on Julia.

**Trait Evolution on Networks.** We implemented several functions and data structures to analyse continuous traits evolving on networks using the methods presented above. The functions we wrote are fully integrated in the Julia environment, and take advantage of the numerous tools it offers. In particular, we build on the package `GLM` (Bates, 2016), which is part of the `JuliaStats` project (JuliaStats, 2016). The implementation has been written to be flexible, so that models other than the BM could be easily cast in this linear regression framework in the future. See Appendix 4.A for the manual pages corresponding to the functions developed for the continuous traits study.

**Testing and Documentation.** The package uses the continuous integration tool Travis CI ([travis-ci.org](https://travis-ci.org)), to run an extensive set of unitary tests, and to automatically deploy the documentation (<http://crs14.github.io/PhyloNetworks.jl/stable/>). The manual pages are automatically built at each new update (push to the master branch), thanks to the package `Documenter` (Hatherly & Piibeleht, 2017). It is written in a “Julia markdown” (.jmd) format, so that all the examples shown are re-run at each build with the last version of the functions, thanks to package `Weave` (Pastell, 2017). Associated with the Git tracking system, all these tools are there to help us keep this collaborative package up-to-date and working smoothly.

## 4.5 Perspectives

To link this section with the rest of this manuscript, that is dealing with shift detection for OU processes, one could think of two natural extensions of the model presented above: shift detection for the BM, and OU modeling.

### 4.5.1 Shift Detection

In this chapter, we only studied shifts on fixed branches, representing heterosis. However, similar to the tree case, as the phylogenetic networks become larger, shift detection on other branches might be needed. Because we forbade shifts on hybrid branches for identifiability reasons, this problem is actually quite similar to the one we tackled in Chapters 2 and 3. Using the notations of Proposition 4.2.5, denote by  $\mathbf{X}$  the vector of traits measured at the node of a network  $N = (V, E)$  issued from a shifted BM. We identify an edge with its ending node for tree edges, and for a hybrid edge ending at node  $i$ , we denote by  $(i, 1)$  the major edge (with  $\gamma_{i,1} > 0.5$ ) and  $(i, 2)$  the minor edge ( $\gamma_{i,2} = 1 - \gamma_{i,1}$ ). Denoting  $\text{pa}(i)$  the parent nodes of node  $i$  ( $|\text{pa}(i)| = 1$  if  $i$  is a tree node, and  $|\text{pa}(i)| = 2$

if  $i$  is a hybrid), we get the following completed model from Proposition 4.2.5:

$$\begin{cases} X_i \mid X_{\text{pa}(i)} \sim \mathcal{N}(X_{\text{pa}(i)} + \Delta_i, \ell_i \sigma^2) & \text{if } |\text{pa}(i)| = 1 \\ X_i \mid (X_{\text{pa}(i,1)}, X_{\text{pa}(i,2)}) \sim \mathcal{N}(\gamma_{i,1} X_{\text{pa}(i,1)} + \gamma_{i,2} X_{\text{pa}(i,2)}, (\gamma_{i,1}^2 \ell_{i,1} + \gamma_{i,2}^2 \ell_{i,2}) \sigma^2) & \text{if } |\text{pa}(i)| = 2. \end{cases}$$

The likelihood of the completed dataset can then be written as:

$$\log p_{\theta}(\mathbf{X}) = \prod_{i \in V} p_{\theta}(X_i \mid \mathbf{X}_{\text{pa}(i)}) = \prod_{i: |\text{pa}(i)|=1} p_{\theta}(X_i \mid X_{\text{pa}(i)}) \prod_{i: |\text{pa}(i)|=2} p_{\theta}(X_i \mid X_{\text{pa}(i,1)}, X_{\text{pa}(i,2)})$$

where each term of the product is just a Gaussian distribution. Similarly to the approach we took in the tree case (see Section 2.4.1), an EM could be written to maximize the likelihood when the number of shifts is fixed. The conditional expectation of the complete log-likelihood given the data at the tips is straightforward to write.

At the M step, the parameters would be estimated the same way they were for trees. In particular, as shifts are only allowed on tree branches, we recover the sum of costs that makes their positioning easy (see Section 2.C.3).

At the E step, as everything is Gaussian, we can still compute the conditional expectations and variances explicitly (using the same equations as in Section 2.C.1). However, it is not clear that an efficient algorithm such as the upward-downward (see Section 3.C.2) could be used here. Indeed, the hybridization events introduce some loops in the graphical model, that need to be handled separately. It could be possible to deal with those cycles through “variable elimination”, merging all the nodes in the cycle into one new synthetic multivariate node (Peyrard et al., 2015).

Finally, the identifiability and model selection problems remain. As in the tree case, we would need some combinatorial tools to measure the space of solutions. These are yet to be studied. Alexeev & Alekseyev (2016) recently proposed an extension of the problem of counting the number of convex coloring of the tips (see Sections 1.1 and 2.3.2) to a network, that could be used for model selection.

## 4.5.2 Ornstein-Uhlenbeck

As for trait evolution on trees, other processes than the BM could be used to model the dynamical evolution of the trait. The Ornstein-Uhlenbeck naturally comes into mind. As pointed out earlier (see Section 4.2.1), the salient difference is that we need to define a *merging rule* for the traits of hybrids. For an OU with one single optimum value over the whole tree, the weighted average merging rule could be adapted. Problems might arise if the OU is allowed to have several optima on the tree. What should be the optimum value of a hybrid species, whose parents have different optima  $\beta_a$  and  $\beta_b$ ? Should we also take the weighted average? Or choose one of the two values? Or maybe define a brand new optimum for the hybrid species?

All of these choices could be legitimate from a biological point of view, depending on the species and traits studied. However, further work would be needed to see which merging rule would make the computations feasible from a mathematical and computational point of view. In particular, it could be interesting to find a merging rule that could allow, in the ultrametric case, to see the OU as a BM on a re-scaled network, as we did in the tree case (see Sections 1.4.2.4 and 3.2.4).

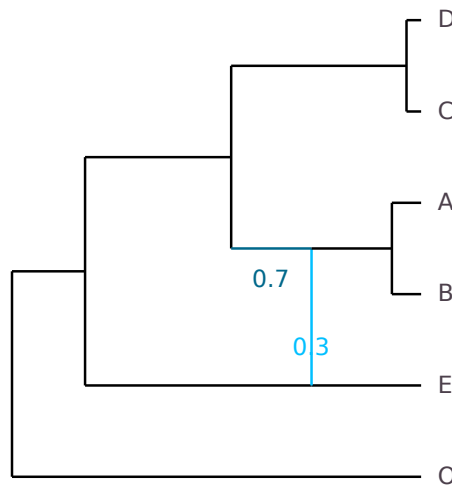
## Appendix

### 4.A Documentation: Continuous Trait Evolution

This appendix presents the main functions for trait analysis in the PhyloNetworks package. It is extracted from the online documentation ([http://crsl4.github.io/PhyloNetworks.jl/stable/man/trait\\_tree/](http://crsl4.github.io/PhyloNetworks.jl/stable/man/trait_tree/)). It was written in collaboration with Cécile Ané and Claudia Solís-Lemus.

We assume a fixed network, correctly rooted, with branch lengths proportional to calendar time. Here, we consider the true network that was used in the previous sections, and which is ultrametric (all the tips are contemporary).

```
plot(truenet, useEdgeLength=true, showGamma=true)
```



#### 4.A.1 Trait simulation

We start by generating continuous traits to study. We simulate three traits on the network (two independent, one dependent), using a Brownian Motion (BM) model of trait evolution on the network. We start by choosing the parameters of the BM (ancestral mean and variance), by creating objects of class `ParamsBM<:ParamsProcess`.

```
params_trait1 = ParamsBM( 2, 0.5) # BM with mean 2 and variance 0.5
params_trait2 = ParamsBM(-2, 1)  # BM with mean -2 and variance 1.0
```

We then simulate the independent traits according to these parameters, using function `simulate` (fixing the seed, for reproducibility).

```
srand(18480224)
sim1 = simulate(truenet, params_trait1) # simulate a BM on truenet
sim2 = simulate(truenet, params_trait2)
```

This creates objects of class `TraitSimulation`, from which we can extract the data at the tips, thanks to the method `getindex(::TraitSimulation, ::Symbol)`.

```
trait1 = sim1[:Tips] # trait 1 at the tips (data)
trait2 = sim2[:Tips]
```

This extractor creates an `Array` with one column, and as many lines as the number of tips there are in the phylogeny. It is sorted in the same order as the tips of the phylogeny

used to simulate it. If needed, we could also extract the simulated values at the internal nodes in the network:

```
sim1[:InternalNodes]
```

Finally, we generate the last trait correlated with trait 1 (but not trait 2), with phylogenetic noise.

```
srand(18700904)
noise = simulate(truenet, ParamsBM(0, 0.1)) # phylogenetic residuals
trait3 = 10 + 2 * trait1 + noise[:Tips] # trait to study. independent of trait2
```

## 4.A.2 Phylogenetic regression

Assume that we measured the three traits above, and that we wanted to study the impact of traits 1 and 2 on trait 3. To do that, we can perform a phylogenetic regression.

In order to avoid confusion, the function takes in a `DataFrame`, that has an extra column with the names of the tips of the network, labeled `tipNames`. Here, we generated the traits ourselves, so they are all in the same order.

```
julia> using DataFrames

julia> dat = DataFrame(trait1 = trait1, trait2 = trait2, trait3 = trait3,
                      tipNames = tipLabels(sim1))
```

```
6x4 DataFrames.DataFrame
| Row | trait1 | trait2 | trait3 | tipNames |
-----|-----|-----|-----|-----|
| 1 | 4.08298 | -7.34186 | 16.673 | "D" |
| 2 | 3.10782 | -7.45085 | 15.0831 | "C" |
| 3 | 2.17078 | -3.32538 | 14.4522 | "A" |
| 4 | 1.87333 | -4.26472 | 13.9712 | "B" |
| 5 | 2.8445 | -5.96857 | 16.417 | "E" |
| 6 | 5.88204 | -1.99388 | 22.0269 | "O" |
```

Phylogenetic regression / ANOVA is based on the `GLM` package, with the network as an extra argument, using function `phyloNetworklm`.

```
julia> fitTrait3 = phyloNetworklm(@formula(trait3 ~ trait1 + trait2), dat, truenet)
DataFrames.DataFrameRegressionModel{PhyloNetworks.PhyloNetworkLinearModel,Array{
  Float64,2}}
```

```
Formula: trait3 ~ 1 + trait1 + trait2
```

```
Model: BM
```

```
Parameter(s) Estimates:
```

```
Sigma2: 0.034712
```

```
Coefficients:
```

```

      Estimate Std.Error t value Pr(>|t|)
(Intercept)  11.9564   1.15462 10.3552  0.0019
trait1       1.69111   0.183047 9.23868  0.0027
trait2       0.170664  0.155645  1.0965  0.3530
```

```
Log Likelihood: -2.9851753461
```

```
AIC: 13.9703506922
```

From this, we can see that the intercept, the coefficient for trait 1 and the variance of the noise are correctly estimated (given that there are only 6 taxa). In addition, the

Student test for the coefficient associated with trait 2 has a high p-value, which means that this coefficient is not significantly different from 0. This is consistent with the way we simulated trait 3.

The function returns an object of type `PhyloNetworkLinearModel<:LinPredModel`. It is a subtype of the GLM type `LinPredModel`, which means that all base functions from Julia `StatsBase` can be applied to it. See the documentation for this type for a list of all functions that can be used. Some functions allow the user to retrieve directly the estimated parameters of the BM, and are specific to this object.

```
julia> sigma2_estim(fitTrait3) # estimated variance of the BM
0.034711959298062325
```

```
julia> mu_estim(fitTrait3) # estimated root value of the BM
11.956367929622921
```

## 4.A.3 Ancestral State Reconstruction

### 4.A.3.1 From known parameters

If we assume that we know the exact model of evolution that generated the traits, we can do ancestral trait reconstruction. Here, we simulated trait 1 ourselves, so we can use the true process, with the true parameters. In other words, we can reconstruct the state at the internal nodes, given the values at the tips, the known value at the root and the known BM variance.

```
ancTrait1 = ancestralStateReconstruction(truenet, trait1, params_trait1)
```

Function `ancestralStateReconstruction` creates an object with type `ReconstructedStates`. Several extractors can be applied to it:

```
julia> expectations(ancTrait1) # predictions
13x2 DataFrames.DataFrame
```

Row	nodeNumber	condExpectation
1	-5	3.55615
2	-7	2.08473
3	5	2.42943
4	-4	2.61415
5	-8	2.56143
6	-3	2.26785
7	-2	2.0
8	1	4.08298
9	2	3.10782
10	3	2.17078
11	4	1.87333
12	6	2.8445
13	7	5.88204

```
julia> stderr(ancTrait1) # associated standard errors
7-element Array{Float64,1}:
 0.312339
 0.429933
 0.812157
 0.985996
 1.00992
 0.807042
```

0.0

```

julia> predint(ancTrait1) # prediction interval (default to 95%)
13x2 Array{Float64,2}:
 2.94398  4.16832
 1.24207  2.92738
 0.837628 4.02123
 0.681629 4.54666
 0.582023 4.54084
 0.686076 3.84962
 2.0      2.0
 4.08298  4.08298
 3.10782  3.10782
 2.17078  2.17078
 1.87333  1.87333
 2.8445   2.8445
 5.88204  5.88204

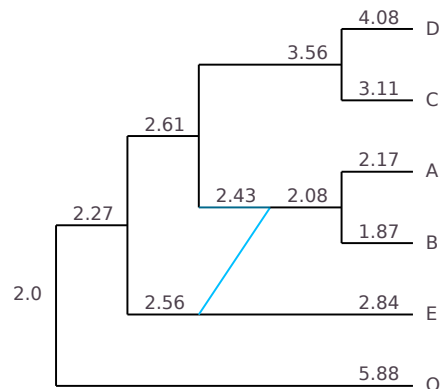
```

We can plot the ancestral states or prediction intervals on the tree, using the `nodeLabel` argument of the `plot` function.

```

ancExpe = expectationsPlot(ancTrait1); # format expected ancestral states for the
plot(truenet, nodeLabel = ancExpe)

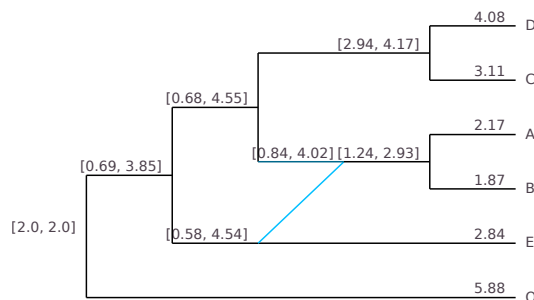
```



```

ancInt = predintPlot(ancTrait1) # format the prediction intervals for the plot
plot(truenet, nodeLabel = ancInt)

```



As we know the true ancestral states here, we can compare them to our estimation.

```

julia> predictions = DataFrame(
  infPred=predint(ancTrait1)[1:7, 1],
  trueValue=sim1[:, :InternalNodes],
  supPred=predint(ancTrait1)[1:7, 2])

```



```
7x3 DataFrames.DataFrame
| Row | infPred | trueValue | supPred |
-----|-----|-----|-----|
| 1 | 2.94398 | 2.74233 | 4.16832 |
| 2 | 1.24207 | 2.24355 | 2.92738 |
| 3 | 0.837628 | 1.38334 | 4.02123 |
| 4 | 0.681629 | 1.50076 | 4.54666 |
| 5 | 0.582023 | 2.84188 | 4.54084 |
| 6 | 0.686076 | 1.76745 | 3.84962 |
| 7 | 2.0 | 2.0 | 2.0 |
```

### 4.A.3.2 From estimated parameters

In real applications though, we do not have access to the true parameters of the process that generated the data. We can estimate it using the previous function. To fit a regular BM, we just need to do a regression of trait 1 against a simple intercept:

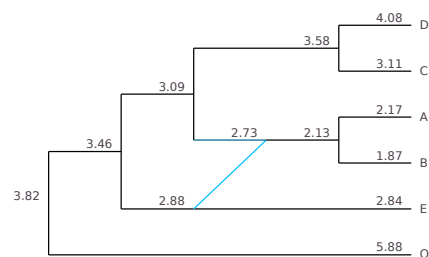
```
fitTrait1 = phyloNetworklm(@formula(trait1 ~ 1), dat, truenet)
```

We can then apply the `ancestralStateReconstruction` function directly to the fitted object:

```
ancTrait1Approx = ancestralStateReconstruction(fitTrait1)
```

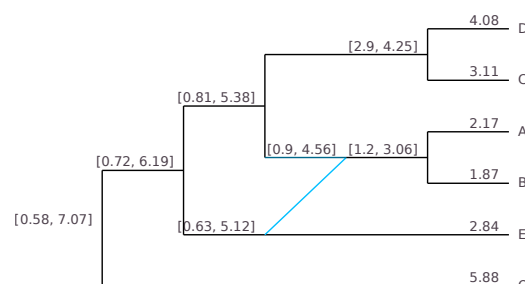
The prediction intervals ignore the fact that we estimated the process parameters, so they are less accurate and the function throws a warning. The output is an object of the same `ReconstructedStates` type as earlier, and the same extractors can be applied to it:

```
plot(truenet, nodeLabel = expectationsPlot(ancTrait1Approx))
```



For convenience, the two steps described above (fitting against the intercept, and then do ancestral state reconstruction) can be done all at once with a single call of the function `ancestralStateReconstruction` on a `DataFrame` with the trait to reconstruct, and the tip labels:

```
datTrait1 = DataFrame(trait1 = trait1, tipNames = tipLabels(sim1))
ancTrait1Approx = ancestralStateReconstruction(datTrait1, truenet)
plot(truenet, nodeLabel = predintPlot(ancTrait1Approx))
```

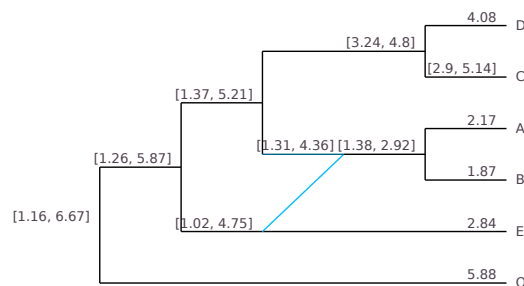


This produces the exact same results.

#### 4.A.3.3 Data imputation

Note that there is no theoretical difference between an internal node, for which we could not measure the value of the trait, and a missing value at a tip of the network. Consequently, the previous `ancestralStateReconstruction` function can be used to do data imputation. To see this, let's add some missing values in trait 1.

```
datTrait1[[2], :trait1] = NA # second row: for taxon C
ancTrait1Approx = ancestralStateReconstruction(datTrait1, truenet)
plot(truenet, nodeLabel = predintPlot(ancTrait1Approx))
```



In the plotting function, a prediction interval is shown for the missing values.

#### 4.A.3.4 With known predictors

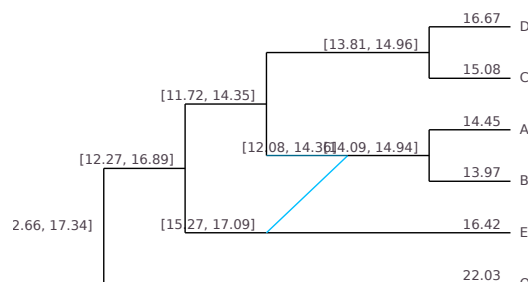
At this point, it might be tempting to apply this function to trait 3 we simulated earlier as a linear combination of trait 1 and a phylogenetic noise. However, this cannot be done directly:

```
ancTrait3 = ancestralStateReconstruction(fitTrait3) # Throws an error !
```

This is because the model we used to fit the trait (a regression with one predictor and an intercept) is not compatible with the simple model of Brownian evolution that we assumed for the ancestral state reconstruction. As the predictor used is not known for ancestral states, it is not possible to reconstruct the trait for this particular model.

The only option we have is to provide the function with the predictor's ancestral states, if they are known. They are known indeed in this toy example that we generated ourselves, so we can reconstruct our trait doing the following:

```
ancTrait3 = ancestralStateReconstruction(fitTrait3,
    [ones(7, 1) sim1[:InternalNodes] sim2[:InternalNodes]])
plot(truenet, nodeLabel = predintPlot(ancTrait3))
```



where we provided the ancestral predictors as a matrix, containing the intercept, and the known predictor at the nodes. The user must be very careful with this function, as no check is done for the order of the predictors, that must be in the same order as the internal nodes of the phylogeny. As ancestral predictors are often unknown, the use of this functionality is discouraged.

#### 4.A.4 Phylogenetic ANOVA

As mentioned above, the `phyloNetworklm` function is based on the `lm` function from **GLM**. This means that it inherits from most of its features, and in particular, it can handle formulas with factors or interactions. For example, in lizards, we might want to do a regression of toe length against body length and the region where each species is found, where this region is coded into 4 categories (say). We might also want to include an interaction effect between body length and region. (This model has no biological basis. It is just meant to show the possibilities of the function).

To illustrate the use of categorical predictors of particular interest in a network with reticulations, let's assume that some heterosis took place after the hybridization event, so that tips "A" and "B" have larger mean compared to the others.

```
delta = 5.0; # value of heterosis
underHyb = [(n == "A" || n == "B") for n in tipLabels(sim1)] # tips under hybrid
underHyb
```

```
6-element Array{Bool,1}:
```

```
false
false
 true
 true
false
false
```

```
for i in 1:length(trait3)
    underHyb[i] && (trait3[i]+=delta) # add delta to tips A and B
end
trait3 # changed: +5 was added by the previous loop to A and B
```

```
6-element Array{Float64,1}:
```

```
16.673
15.0831
19.4522
18.9712
16.417
22.0269
```

The categorical variable `underHyb` separates tips "A" and "B" from the others. We need to mark it as a factor, not a numerical variable, i.e. as a `PooledDataArray`.

```
dat = DataFrame(trait1 = trait1, trait2 = trait2, trait3 = trait3,
               underHyb = underHyb,
               tipNames = tipLabels(sim1))
dat[:underHyb] = PooledDataArray(dat[:underHyb])
```

Now we can include this factor in the regression.

```
julia> fitTrait = phyloNetworklm(@formula(trait3 ~ trait1 + underHyb), dat, truenet)
DataFrames.DataFrameRegressionModel{PhyloNetworks.PhyloNetworkLinearModel,Array{
Float64,2}}
```

```
Formula: trait3 ~ 1 + trait1 + underHyb
```

```
Model: BM
```

```
Parameter(s) Estimates:
```

```
Sigma2: 0.0484988
```

```
Coefficients:
```

	Estimate	Std.Error	t value	Pr(> t )
(Intercept)	11.0616	1.19414	9.26324	0.0027
trait1	1.72504	0.240787	7.16418	0.0056
underHyb: true	5.07354	0.837326	6.05922	0.0090

```
Log Likelihood: -3.9885372687
```

```
AIC: 15.9770745374
```

In this case, the categorical variable indicating which tips are descendants of the reticulation event is indeed relevant, and the heterosis effect is recovered.

This is a very simple example of how to include heterosis, but some general functions to test for it, on networks with more than one hybrid, are also available.

#### 4.A.5 Pagel's Lambda

One classical question about trait evolution is the amount of "phylogenetic signal" in a dataset, that is, the importance of the tree structure to explain variation in the observed traits. One way of doing measuring that is to use Pagel's lambda transformation of the branch lengths. This model assumes a BM on a tree where the internal branches are multiplied by a factor  $\lambda$ , while the external branches are modified so that the total height of the tree is constant. Hence,  $\lambda$  varies between 0 (the tree has no influence on the data) and 1 (the tree is unchanged). Using the same branch length transformations, this model can be straightforwardly extended to phylogenetic networks.

We can illustrate this with the predictor trait we used earlier. We use the same function as before, only indicating the model we want to use:

```
fitPagel = phyloNetworklm(@formula(trait1 ~ 1), dat, truenet, model="lambda")
```

As it is indeed generated according to a plain BM on the phylogeny, the estimated  $\lambda$  should be close to 1. It can be extracted with function `lambda_estim`:

```
julia> lambda_estim(fitPagel)
0.907356122898758
```

## 4.B Decomposition of the Covariance Matrix

In this section, we derive a general formula linking the covariance matrix of a network to the covariance matrices of its underlying trees. We start by dealing with only one hybridization event, and then show how to extend the formula to any number of hybrids.

Assume that we have a network  $\mathcal{N}$ , that has at least one hybridization event (but that might have more). Take  $p$  the pit of this hybridization, with transmissions  $\gamma$  from its first parent  $a$ , and  $1 - \gamma$  from its other parent  $b$ . We want to express  $\mathbf{V}(\gamma)$  the covariance matrix of the network, using  $\mathbf{V}(\gamma = 0)$  and  $\mathbf{V}(\gamma = 1)$  the covariance matrices of the network where this hybridization event is suppressed.

**Proposition 4.B.1.** *The following proposition holds:*

$$\mathbf{V}(\gamma) = \gamma\mathbf{V}(1) + (1 - \gamma)\mathbf{V}(0) - \gamma(1 - \gamma)\left[\mathbf{V}(1)_{pp} - \mathbf{V}(1)_{ab} + \mathbf{V}(0)_{pp} - \mathbf{V}(0)_{ab}\right]\mathbf{D}(p) \quad (4.13)$$

where  $\mathbf{D}(p)$  is the matrix of nodes descending of  $p$ :

$$\forall i, j \in \mathcal{N}, \mathbf{D}(p)_{ij} = \begin{cases} 1 & \text{if } i \text{ and } j \text{ descendants of } p \\ 0 & \text{otherwise} \end{cases}$$

*Proof.* Let  $i$  and  $j$  be two nodes of  $\mathcal{N}$ . We have three distinct cases:

- $i$  and  $j$  are not descendants of  $p$ . Then, by definition, no path going from  $i$  or  $j$  to the root will go through  $p$ , and, from the general formula of the covariance matrix, we can see that:

$$\mathbf{V}(\gamma)_{ij} = \mathbf{V}(0)_{ij} = \mathbf{V}(1)_{ij}$$

and the formula holds.

- $i$  is a descendant of  $p$ , but not  $j$ . Remark that any path going from  $i$  to the root must go through  $a$  or  $b$ , exclusively, so that the set of path is partitioned as  $\mathcal{P}_i = \mathcal{P}_i^{\gamma=0} \cup \mathcal{P}_i^{\gamma=1}$  where  $\mathcal{P}_i^{\gamma=0}$  and  $\mathcal{P}_i^{\gamma=1}$  are the set of path going from  $i$  to the root when the topology is such that, respectively,  $\gamma = 0$  or  $\gamma = 1$ . Then using the general formula, we get the following formula:

$$\begin{aligned} \mathbf{V}(\gamma)_{ij} &= \sum_{\substack{p_i \in \mathcal{P}_i \\ p_j \in \mathcal{P}_j}} \left( \prod_{e \in \mathcal{P}_i} \gamma(e) \right) \left( \prod_{e \in \mathcal{P}_j} \gamma(e) \right) \sum_{e \in \mathcal{P}_i \cap \mathcal{P}_j} \ell_e \\ &= \gamma \sum_{\substack{p_i \in \mathcal{P}_i^{\gamma=1} \\ p_j \in \mathcal{P}_j}} \left( \prod_{e \in \mathcal{P}_i} \gamma(e) \right) \left( \prod_{e \in \mathcal{P}_j} \gamma(e) \right) \sum_{e \in \mathcal{P}_i \cap \mathcal{P}_j} \ell_e \\ &\quad + (1 - \gamma) \sum_{\substack{p_i \in \mathcal{P}_i^{\gamma=0} \\ p_j \in \mathcal{P}_j}} \left( \prod_{e \in \mathcal{P}_i} \gamma(e) \right) \left( \prod_{e \in \mathcal{P}_j} \gamma(e) \right) \sum_{e \in \mathcal{P}_i \cap \mathcal{P}_j} \ell_e \\ &= \gamma\mathbf{V}(1)_{ij} + (1 - \gamma)\mathbf{V}(0)_{ij} \end{aligned}$$

- $i$  and  $j$  are descendants of  $p$ . We write the following decomposition:

$$\begin{aligned} \text{Cov}[X_i; X_j] &= \mathbb{E}\left[\text{Cov}[X_i; X_j \mid X_p]\right] + \text{Cov}\left[\mathbb{E}[X_i \mid X_p]; \mathbb{E}[X_j \mid X_p]\right] \\ &= \text{Cov}[X_i; X_j \mid X_p] + \text{Var}[X_p] \end{aligned}$$

Then, from the recursive formulas:

$$\begin{aligned} \text{Var}[X_p] &= \gamma^2(\text{Cov}[X_a; X_a] + \ell_1) + (1 - \gamma)^2(\text{Cov}[X_b; X_b] + \ell_2) + 2\gamma(1 - \gamma)\text{Cov}[X_a; X_b] \\ &= \gamma^2\text{Var}[X_p \mid \gamma = 1] + (1 - \gamma)^2\text{Var}[X_p \mid \gamma = 0] + 2\gamma(1 - \gamma)\text{Cov}[X_a; X_b] \\ &= \gamma\text{Var}[X_p \mid \gamma = 1] + (1 - \gamma)\text{Var}[X_p \mid \gamma = 0] \\ &\quad + \gamma(1 - \gamma)\left[2\text{Cov}[X_a; X_b] - \text{Var}[X_p \mid \gamma = 1] - \text{Var}[X_p \mid \gamma = 0]\right] \end{aligned}$$

(because  $\gamma^2 = \gamma - \gamma(1 - \gamma)$  and  $(1 - \gamma)^2 = 1 - \gamma - \gamma(1 - \gamma)$ ). As  $a$  and  $b$  are not descendants of  $p$ , from the above,  $\mathbf{V}(\gamma)_{ab} = \mathbf{V}(0)_{ab} = \mathbf{V}(1)_{ab}$ . In addition,  $\text{Cov}[X_i; X_j \mid X_p]$  is not dependent on  $\gamma$  anymore, so that:

$$\text{Cov}[X_i; X_j \mid X_p] = \gamma \text{Cov}[X_i; X_j \mid X_p, \gamma = 1] + (1 - \gamma) \text{Cov}[X_i; X_j \mid X_p, \gamma = 0]$$

and:

$$\text{Cov}[X_i; X_j \mid X_p, \gamma = 1] + \text{Var}[X_p \mid \gamma = 1] = \text{Cov}[X_i; X_j \mid \gamma = 1]$$

Putting everything together, we finally get the advertised formula:

$$\mathbf{V}(\gamma)_{ij} = \gamma \mathbf{V}(1)_{ij} + (1 - \gamma) \mathbf{V}(0)_{ij} - \gamma(1 - \gamma) [\mathbf{V}(1)_{pp} - \mathbf{V}(1)_{ab} + \mathbf{V}(0)_{pp} - \mathbf{V}(0)_{ab}]$$

□

**Corollary 4.B.1.** *If  $\mathcal{N}$  is tree like everywhere but in  $p$  (i.e. if there is only one hybridization event), then the formula simplifies to:*

$$\mathbf{V}(\gamma) = \gamma \mathbf{V}(1) + (1 - \gamma) \mathbf{V}(0) - 2\gamma(1 - \gamma) [t_p - t_s] \mathbf{D}(p) \quad (4.14)$$

where  $s$  is the source of the hybridization event.

This corollary is precisely Proposition 4.4 of the main text.

*Proof.* If  $\mathcal{N}$  is tree-like everywhere but in  $p$ , then:

$$\text{Cov}[X_a; X_b] = \sigma^2 t_s \text{ and } \text{Var}[X_p \mid \gamma = 1] = \text{Var}[X_p \mid \gamma = 0] = \sigma^2 t_p$$

hence the simplification. □

**Proposition 4.B.2.** *Assume that  $\mathcal{N}$  is a level-1 network. If there are  $k$  hybrid species  $p_1, \dots, p_k$  in the network  $\mathcal{N}$ , with parents  $(a_1, b_1), \dots, (a_k, b_k)$  and coefficients  $\gamma_1, \dots, \gamma_k$ , then the variance matrix can be written as a linear combination of variance matrices of trees:*

$$\mathbf{V}(\gamma_1, \dots, \gamma_k) = \sum_{a \in \{0,1\}^k} \prod_{i=1}^k \gamma_i(a_i) \left[ \mathbf{V}(a) - \sum_{j=1}^k \gamma_j(a_j - 1) (\mathbf{V}(a)_{p_j p_j} - \mathbf{V}(a)_{a_j b_j}) \mathbf{D}(p_j) \right]$$

where, for any  $1 \leq i \leq k$ ,  $\gamma_i(1) = \gamma_i(-1) = \gamma_i$ , and  $\gamma_i(0) = (1 - \gamma_i)$ .

We show this formula by recurrence, using the following lemma:

**Lemma 4.B.1.** *Assume that  $\mathcal{N}$  is a level-1 network. Take  $p_1$  and  $p_2$  two hybrid nodes, with parents  $(a_1, b_1)$  and  $(a_2, b_2)$ . Denote by  $\text{des}(p_1)$  the set of all descendants of  $p_1$ . The following holds:*

$$p_2 \in \text{des}(p_1) \iff a_2 \in \text{des}(p_1) \text{ and } b_2 \in \text{des}(p_1)$$

*Proof.* The reciprocal assumption simply follows the definition of descending nodes, and holds for any kind of network. Let's focus on the direct implication. Assume that  $p_2$  is a descendant of  $p_1$ , and that  $a_2$  is not. We show that this contradicts the level-1 assumption. Indeed, as  $p_2$  is a descendant of  $p_1$ , there exists a path  $t$  going from the root to  $p_2$  and going through  $p_1$ . Similarly, as  $a_2$  is not a descendant of  $p_1$ , there exists a path  $t'$  going from the root to  $a_2$  and not going through  $p_1$ . Then, the set of edges  $(t \cup t') \setminus (t \cap t')$  defines a cycle above  $p_2$ . But  $p_1$  is in this cycle, hence some edges are in two cycles, which contradicts the level-1 assumption. This ends the proof. □

*Proof of proposition 4.B.2.* We show the formula by recurrence. If there is only one hybrid, then, from proposition 4.B.1, we have:

$$\begin{aligned} \mathbf{V}(\gamma_1) = & \gamma_1 \left[ \mathbf{V}(1) - (1 - \gamma_1) \left( \mathbf{V}(1)_{p_1 p_1} - \mathbf{V}(1)_{a_1 b_1} \right) \mathbf{D}(p_1) \right] \\ & + (1 - \gamma_1) \left[ \mathbf{V}(0) - \gamma_1 \left( \mathbf{V}(0)_{p_1 p_1} - \mathbf{V}(0)_{a_1 b_1} \right) \mathbf{D}(p_1) \right] \end{aligned}$$

and the formula holds. Assume now that it holds for  $k-1$  hybrids, and let's prove it for  $k$ . From the recurrence, we get:

$$\mathbf{V}(\gamma_1, \dots, \gamma_k) = \sum_{a \in \{0,1\}^{k-1}} \prod_{i=1}^{k-1} \gamma_i(a_i) \left[ \mathbf{V}(a, \gamma_k) - \sum_{j=1}^{k-1} \gamma_j(a_j - 1) \left( \mathbf{V}(a, \gamma_k)_{p_j p_j} - \mathbf{V}(a, \gamma_k)_{a_j b_j} \right) \mathbf{D}(p_j) \right]$$

If  $a \in \{0,1\}^{k-1}$  we have, from proposition 4.B.1:

$$\begin{aligned} \mathbf{V}(a, \gamma_k) = & \gamma_k \left[ \mathbf{V}(a, 1) - (1 - \gamma_k) \left( \mathbf{V}(a, 1)_{p_k p_k} - \mathbf{V}(a, 1)_{a_k b_k} \right) \mathbf{D}(p_k) \right] \\ & + (1 - \gamma_k) \left[ \mathbf{V}(a, 0) - \gamma_k \left( \mathbf{V}(a, 0)_{p_k p_k} - \mathbf{V}(a, 0)_{a_k b_k} \right) \mathbf{D}(p_k) \right] \end{aligned}$$

and, for any  $1 \leq j \leq k-1$ :

$$\begin{aligned} \mathbf{V}(a, \gamma_k)_{p_j p_j} - \mathbf{V}(a, \gamma_k)_{a_j b_j} = & \gamma_k \left( \mathbf{V}(a, 1)_{p_j p_j} - \mathbf{V}(a, 1)_{a_j b_j} \right) + (1 - \gamma_k) \left( \mathbf{V}(a, 0)_{p_j p_j} - \mathbf{V}(a, 0)_{a_j b_j} \right) \\ & - \gamma_k (1 - \gamma_k) \left( \mathbf{V}(a, 1)_{p_k p_k} - \mathbf{V}(a, 1)_{a_k b_k} \right) \left( \mathbf{D}(p_k)_{p_j p_j} - \mathbf{D}(p_k)_{a_j b_j} \right) \\ & - \gamma_k (1 - \gamma_k) \left( \mathbf{V}(a, 0)_{p_k p_k} - \mathbf{V}(a, 0)_{a_k b_k} \right) \left( \mathbf{D}(p_k)_{p_j p_j} - \mathbf{D}(p_k)_{a_j b_j} \right) \end{aligned}$$

but, from lemma 4.B.1, for any  $1 \leq j \leq k-1$ ,  $\mathbf{D}(p_k)_{p_j p_j} - \mathbf{D}(p_k)_{a_j b_j} = 0$ . Indeed, if  $p_j$  is a descendant of  $p_k$ , then  $a_j$  and  $b_j$  are too, and  $\mathbf{D}(p_k)_{p_j p_j} = \mathbf{D}(p_k)_{a_j b_j} = 1$ , and, if  $p_j$  is a not descendant of  $p_k$ , then  $a_j$  and  $b_j$  are not either, and  $\mathbf{D}(p_k)_{p_j p_j} = \mathbf{D}(p_k)_{a_j b_j} = 0$ . Hence:

$$\mathbf{V}(a, \gamma_k)_{p_j p_j} - \mathbf{V}(a, \gamma_k)_{a_j b_j} = \gamma_k \left( \mathbf{V}(a, 1)_{p_j p_j} - \mathbf{V}(a, 1)_{a_j b_j} \right) + (1 - \gamma_k) \left( \mathbf{V}(a, 0)_{p_j p_j} - \mathbf{V}(a, 0)_{a_j b_j} \right)$$

and:

$$\begin{aligned} \mathbf{V}(\gamma_1, \dots, \gamma_k) = & \sum_{a \in \{0,1\}^{k-1}} \prod_{i=1}^{k-1} \gamma_i(a_i) \\ & \times \left( \gamma_k \left[ \mathbf{V}(a, 1) - (1 - \gamma_k) \left( \mathbf{V}(a, 1)_{p_k p_k} - \mathbf{V}(a, 1)_{a_k b_k} \right) \mathbf{D}(p_k) \right. \right. \\ & \quad \left. \left. - \sum_{j=1}^{k-1} \gamma_j(a_j - 1) \left( \mathbf{V}(a, 1)_{p_j p_j} - \mathbf{V}(a, 1)_{a_j b_j} \right) \mathbf{D}(p_j) \right] \right. \\ & \quad \left. + (1 - \gamma_k) \left[ \mathbf{V}(a, 0) - \gamma_k \left( \mathbf{V}(a, 0)_{p_k p_k} - \mathbf{V}(a, 0)_{a_k b_k} \right) \mathbf{D}(p_k) \right. \right. \\ & \quad \quad \left. \left. - \sum_{j=1}^{k-1} \gamma_j(a_j - 1) \left( \mathbf{V}(a, 0)_{p_j p_j} - \mathbf{V}(a, 0)_{a_j b_j} \right) \mathbf{D}(p_j) \right] \right) \end{aligned}$$

and the announced formula follows.  $\square$

**Corollary 4.B.2.** *Assume that  $\mathcal{N}$  is a level-1 network, and that there are exactly  $k$  hybrid species  $p_1, \dots, p_k$  in the network  $\mathcal{N}$ , i.e. that the rest of the network is tree-like. Then the formula can be simplified to:*

$$\mathbf{V}(\gamma_1, \dots, \gamma_k) = \sum_{a \in \{0,1\}^k} \prod_{i=1}^k \gamma_i(a_i) \mathbf{V}(a) - 2 \sum_{j=1}^k \gamma_j(1 - \gamma_j) (t_{p_j} - t_{s_j}) \mathbf{D}(p_j)$$

where, for any  $1 \leq i \leq k$ ,  $s_j$  is the MRCA of  $a_j$  and  $b_j$  (source of the hybrid).

*Proof.* If there are exactly  $k$  hybrids in  $\mathcal{N}$ , then, for any  $a \in \{0,1\}^k$ ,  $\mathbf{V}(a)$  is the variance matrix of a tree. Hence  $\mathbf{V}(a)_{p_j p_j} = t_{p_j}$ , and  $\mathbf{V}(a)_{a_j b_j} = t_{s_j}$  for any  $1 \leq j \leq k$ , and the second term of the formula becomes:

$$\begin{aligned} M &= \sum_{a \in \{0,1\}^k} \prod_{i=1}^k \gamma_i(a_i) \sum_{j=1}^k \gamma_j(a_j - 1) (\mathbf{V}(a)_{p_j p_j} - \mathbf{V}(a)_{a_j b_j}) \mathbf{D}(p_j) \\ &= \sum_{a \in \{0,1\}^k} \prod_{i=1}^k \gamma_i(a_i) \sum_{j=1}^k \gamma_j(a_j - 1) (t_{p_j} - t_{s_j}) \mathbf{D}(p_j) \\ &= \sum_{j=1}^k \left[ \sum_{a \in \{0,1\}^k} \prod_{i=1}^k \gamma_i(a_i) \gamma_j(a_j - 1) \right] (t_{p_j} - t_{s_j}) \mathbf{D}(p_j) \end{aligned}$$

but:

$$\begin{aligned} \sum_{a \in \{0,1\}^k} \prod_{i=1}^k \gamma_i(a_i) \gamma_j(a_j - 1) &= \sum_{a \in \{0,1\}^k} \prod_{i \neq j} \gamma_i(a_i) \gamma_j(a_j) \gamma_j(a_j - 1) \\ &= \sum_{a \in \{0,1\}^{k-1}} \prod_{i \neq j} \gamma_i(a_i) \sum_{a_j \in \{0,1\}} \gamma_j(a_j) \gamma_j(a_j - 1) \\ &= \left( \sum_{a \in \{0,1\}^{k-1}} \prod_{i \neq j} \gamma_i(a_i) \right) (2\gamma_j(1 - \gamma_j)) \\ &= 2\gamma_j(1 - \gamma_j) \left( \prod_{i \neq j} (\gamma_i(0) + \gamma_i(1)) \right) \\ &= 2\gamma_j(1 - \gamma_j) \end{aligned}$$

Hence,  $M = \sum_{j=1}^k 2\gamma_j(1 - \gamma_j) (t_{p_j} - t_{s_j}) \mathbf{D}(p_j)$ , and we get the announced formula.  $\square$

**Other Form for the Variance Matrix.** From the proof of proposition 4.B.1, we can state a more precise formula:

**Proposition 4.B.3.** *For a level-1 network:*

$$\mathbf{V}(\gamma)_{ij} = \begin{cases} \mathbf{V}(0)_{ij} & \text{if } i \text{ and } j \text{ are not descendants of } p \\ \gamma \mathbf{V}(1)_{ij} + (1 - \gamma) \mathbf{V}(0)_{ij} & \text{if } i \text{ or } j \text{ is descendant of } p \\ \mathbf{V}(0)_{ij} - 2\gamma(1 - \gamma)(t_p - t_s) & \text{if } i \text{ and } j \text{ are both descendants of } p \end{cases} \quad (4.15)$$



*Proof.* The first two cases follow directly from the proof of proposition 4.B.1. The last case follows from the fact that, for a level-1 network,  $\mathbf{V}(0)_{pp} = \mathbf{V}(1)_{pp}$ . Indeed, for a level-1 network, all the nodes between the source  $s$  and the pit  $p$  are tree-like. Hence,  $\mathbb{V}\text{ar}[X_p | \gamma = 1] = \mathbb{V}\text{ar}[X_s] + \ell_{s \rightarrow p} = \mathbb{V}\text{ar}[X_p | \gamma = 0]$ , where  $\ell_{s \rightarrow p} = t_p - t_s$  is the time elapsed between  $s$  and  $p$ , that do not depends on  $\gamma$ . As  $\mathbb{C}\text{ov}[X_a; X_b] = \mathbb{V}\text{ar}[X_s]$ , the formula follows.  $\square$



# Chapter 5

## Extensions and Perspectives

### Contents

---

<b>5.1</b>	<b>Dealing with Tree and Trait Uncertainty</b> . . . . .	<b>197</b>
5.1.1	Simulation Studies . . . . .	197
5.1.2	Including Trait Measurement Errors . . . . .	201
5.1.3	Factor Analysis . . . . .	202
<b>5.2</b>	<b>Convergence and Sparsity</b> . . . . .	<b>203</b>
5.2.1	Convergence and Fused-ANOVA . . . . .	203
5.2.2	Sparse Number of Shifted Traits . . . . .	205
<b>5.3</b>	<b>Non-Ultrametric Trees</b> . . . . .	<b>205</b>
5.3.1	Identifiability and Model Selection . . . . .	205
5.3.2	Inference of the OU . . . . .	209
<b>5.4</b>	<b>Sampling Scheme and Missing Data</b> . . . . .	<b>212</b>

---

In this section, we try to alleviate some of the (numerous) assumptions we made, and explore their consequences on our framework. These developments are still preliminary but could serve as inspiration for future work on the subject.

### 5.1 Dealing with Tree and Trait Uncertainty

As stated in the introductory chapter (Sections 1.2.1 and 1.4.5.1), we assumed throughout this work that both the tree (topology and branch lengths) and the tip data were known without any uncertainty. In this section, we first try to assess, through simulations, the bias introduced when these assumptions are not fulfilled. We then briefly review some of the adaptations that could be made to account for the trait measurement errors.

#### 5.1.1 Simulation Studies

In this section, we try to assess the impact of measurement errors or tree misspecifications on parameters estimation, when they are not explicitly accounted for.

**Experimental Design.** We used here the same simulation scheme than in the multivariate analysis of Chapter 3 (see Section 3.4). The base scenario had 3 shifts, with fixed positions and values, on a 160 taxa tree (see Fig. 3.4.1, top-left). The base scaling factor of 1.25 was applied to the shift values. Four traits were simulated on the tree, with a scalar selection strength matrix with  $\alpha = 1$ , a tip stationary variance  $\gamma^2 = 1$ , and a base

correlation  $r_d = 0.4$ . We added the three parameters described below, that we varied one by one. First, we added some measurement errors at the tips. To do that, we simulated a matrix  $\mathbf{Y}$  of traits at the tips according to the model at hand, and then generated a matrix of observations  $\mathbf{Y}_{\text{obs}}$ , adding a noise independently to each tip:

$$\mathbf{Y}_{\text{obs}}^i \sim \mathcal{N}(\mathbf{Y}^i, \mathbf{P}) \quad \text{with} \quad \mathbf{P} = \begin{pmatrix} e & r_e & r_e & r_e \\ r_e & e & r_e & r_e \\ r_e & r_e & e & r_e \\ r_e & r_e & r_e & e \end{pmatrix}.$$

- $e$  is a diagonal measurement error variance on the tip traits, that took 9 values between 0 and 5. ( $r_e$  was fixed to 0 when  $e$  varied). It was applied independently to all the tip measurements. When  $e = 0$  the model is correctly specified, and when  $e > \gamma^2 = 1$ , the error variance is larger than the tree-induced variance, so we expect the method to behave poorly.
- $r_e$  is a correlation error factor for the measurement at the tips, that took 5 values between 0 and 0.8. When  $r_e$  varied,  $e$  was fixed to 0.5. This parameter is used to assess the impact of the measurement error correlations on the method, for a fixed level of noise.

The third parameter impacted the branch lengths of the trees:

- $1/l$  is the parameter of a gamma distribution, that took 6 values between 0 and 1. We used this parameter to alter the original simulation tree. For a given  $l$  value, we drew a parameter  $\delta \sim \text{Gamma}(l, 1/l)$ , so that  $\delta$  had expectation 1 and variance  $1/l$ . We then applied Pagel's  $\delta$  transform (see Section 1.4.2.3 of the introductory chapter) to the tree, in order to alter its branch lengths, while keeping the tree ultrametric. Traits were simulated under this altered tree, whereas parameter inference was performed using the original tree. When  $1/l = 0$ , then the variance is formally equal to 0, and both trees are equal. When  $1/l$  increases, the variance increases, and the altered tree moves away from the original (observed) one. See Figures 1.4.9 and 5.1.4 for examples of several altered versions of an original tree.

**Effects of Measurement Error.** In Figure 5.1.1, we first analyse the effects of measurement error  $e$  on the estimations. As expected, the measurement error progressively dilutes the phylogenetic signal, with a sharp decrease of performances when  $e$  is larger than  $\gamma^2 = 1$ . When the error is large, then the shifts tend to be missed, and the variance is over-estimated. In addition, the selection strength is also over-estimated.

This is consistent with the conclusions of Cooper et al. (2016), who found that the OU was erroneously favored over the BM for simulated datasets with added measurement errors. Indeed, as we saw in Section 1.4.2.4 of the introductory chapter, the univariate OU can be seen as a tree transformation, that extends the terminal branches when  $\alpha$  increases, reducing the phylogenetic signal (see Fig. 1.4.10). When the measurement errors are not accounted for explicitly in the model, they act like longer terminal branches, and have a tendency to dilute the phylogenetic signal. They hence favor artificially large values of  $\alpha$ . This is another evidence that this  $\alpha$  parameter, that was initially introduced as a selection strength parameter, should be interpreted with caution (see also Section 3.6.1 of Chapter 3).

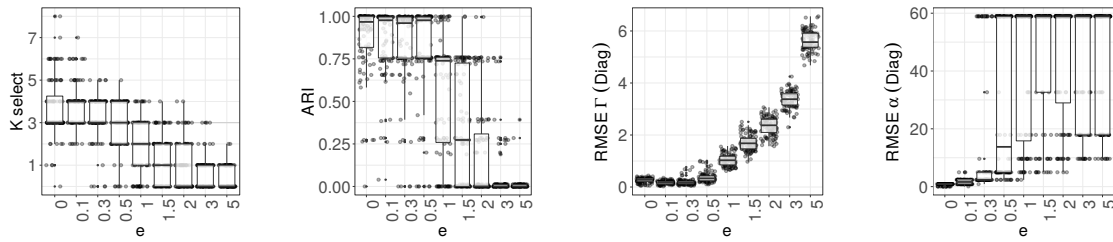


Figure 5.1.1 – Estimated number of shifts, ARI and Root Mean Square Error (RMSE) of the diagonal coefficient of stationary variance matrix  $\Gamma$  and selection strength  $\alpha$ , for PhylogeneticEM with the default model selection criterion, when the diagonal measurement errors  $e$  increase.

**Effects of Measurement Error Correlations.** On Figure 5.1.2, we see that increasing the correlations of the measurement errors, for a fixed level of noise ( $e = 0.5$ ), does not substantially degrade the estimations. The intensity of the measurement error, rather than its structure, hence alters the results.

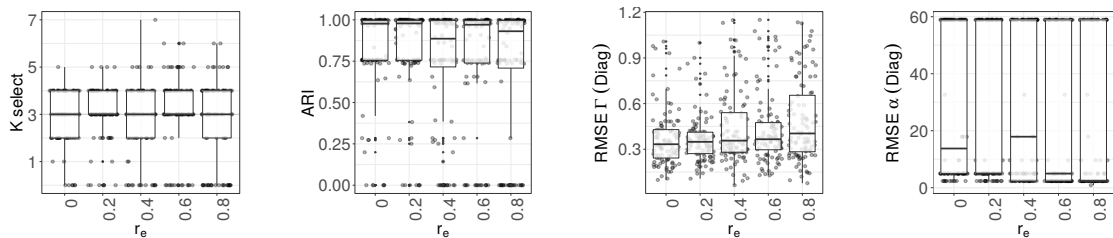


Figure 5.1.2 – Estimated number of shifts, ARI and Root Mean Square Error (RMSE) of the diagonal coefficient of stationary variance matrix  $\Gamma$  and selection strength  $\alpha$ , for PhylogeneticEM with the default model selection criterion, when the correlations  $r_e$  of the measurement error increase (with  $e = 0.5$  fixed).

**Effects of Tree Misspecification.** On Figure 5.1.3, we show the impact of a tree misspecification on the parameters estimation. It seems that misspecified branch lengths tend to lead to an increase in the number of false-positives, with an ARI that drops when the simulation tree differs more and more from the tree used in the analysis. Here, we present the results of two different model selection criteria. In blue (“LINselect ml”) is the default criterion, that we’ve been using until now. In red (“LINselect lsq”) is an alternative criterion, where the best solution in  $\alpha$ , for a given number of shifts, is taken to be the one minimizing the least squares criterion, instead of the maximum likelihood. It seems that this criterion is more robust to the tree misspecification, as it selects for less shifts. However, the ARI scores obtained using this criterion are not dramatically improved, compared to the default selection criterion.

On Figure 5.1.4, we show two typical scenarios, with, on the left, the original tree (with  $1/l = 0.75$ ), and, on the right, the solution found on the wrong observed tree. The first line presents a “good” scenario, where the transformation factor  $\delta$  is greater than 1, so that most of the variation occurs on tip branches. In such a scenario, clades are not clearly marked, and the structured differences are mainly due to the shifts. The right solution is almost found in that case. On the second line, we show a “bad” scenario, where  $\delta < 1$ , so that most of the variation on the modified tree happens on ancestral

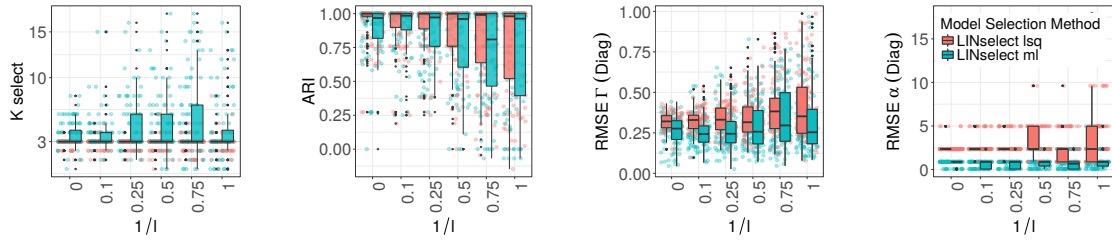


Figure 5.1.3 – Estimated number of shifts, ARI and Root Mean Square Error (RMSE) of the diagonal coefficient of stationary variance matrix  $\mathbf{\Gamma}$  and selection strength  $\alpha$ , for PhylogeneticEM with the default model selection criterion (blue), or an alternative one (red) when the tree modification factor  $1/l$  increases.

branches. This tends to create well separated clades, with highly structured variations of the trait. When using the original tree for inference, these variations cannot be explained by the tree, as the clades are not that well separated anymore, and the method makes up for this by adding numerous shifts.

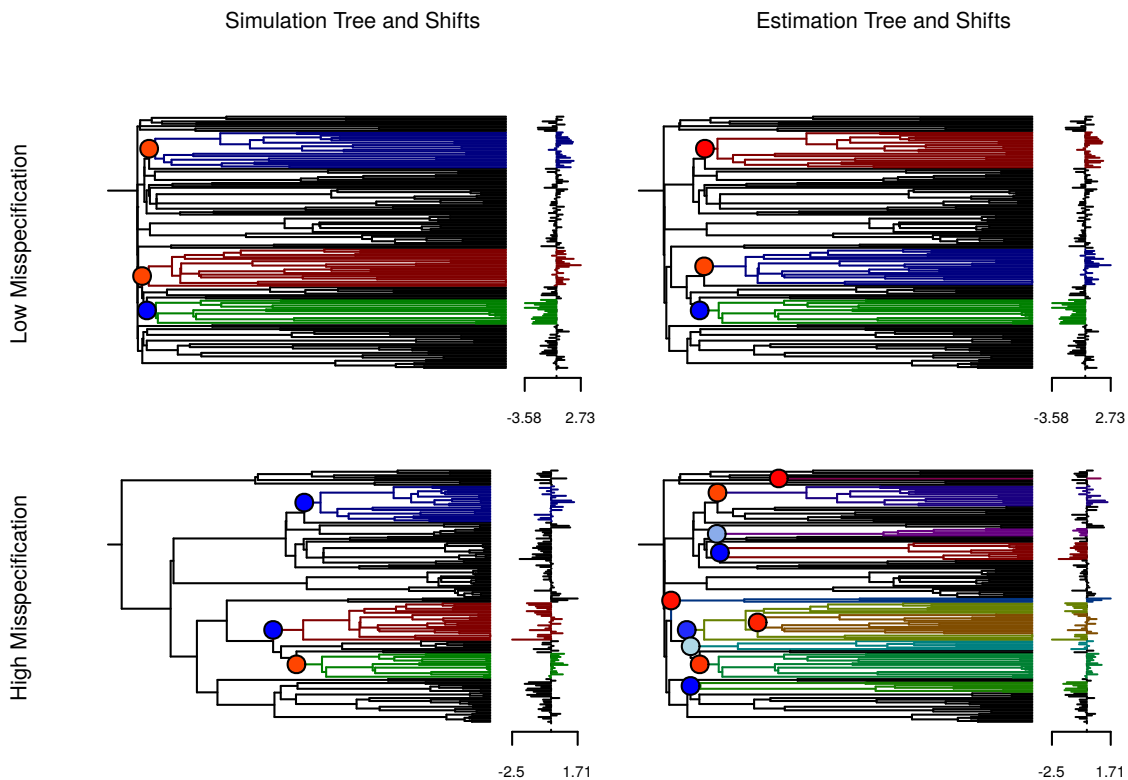


Figure 5.1.4 – Simulation scenario with the true tree and shifts (left) and inferred solutions with the wrong tree (right), with  $1/l = 0.75$ . The upper line shows a “good” scenario (the true solution is almost perfectly recovered), and the bottom line a “bad” one (many shifts are added to make up for the lost clade structure of the tree).

These simulations show us that the tree branch length misspecification can have an unexpected and deleterious impact on shift detection. Going further, errors in the topology of the tree are likely to alter the results even more. There are no simple ways to take these tree uncertainties into account in our framework. This is one of the strength

of Bayesian methods, that perform tree inference and trait analysis at once, rather than using a two step procedure (see Section 1.4.5.3 of the introductory chapter).

### 5.1.2 Including Trait Measurement Errors

As presented in the introductory chapter (Section 1.4.5.1), several methods already exist to cope with measurement errors. These methods could be easily adapted to our framework. Indeed, errors just add an extra layer to our hierarchical modeling: as exposed in Section 1.4.5.1, we link each observation on the tree with its species by a zero-length branch. What we used to consider as tips are now internal nodes, and the observations, for each species, are linked to them by zero length branches. Using the same notations as before (see Section 3.2), we denote by  $\mathbf{X} = (\mathbf{Z}, \mathbf{Y})$  the matrix of (non-observed) traits at the internal nodes and tips of the tree, and by  $\mathbf{Y}_o$  the matrix of observed traits. We assume that there are  $m$  internal nodes numbered from 1 to  $m$ , (1 is the root),  $n$  tips numbered from  $m+1$  to  $m+n$ , and  $n_o$  observations, numbered from  $m+n+1$  to  $m+n+n_o$ . See Figure 5.1.5 for an illustration of these notations. The model can then be written as:

$$\begin{aligned} \mathbf{X}^1 &\sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Gamma}) && \text{for the root,} \\ \mathbf{X}^j \mid \mathbf{X}^{\text{pa}(j)} &\sim \mathcal{N}(\mathbf{X}^{\text{pa}(j)} + \boldsymbol{\Delta}^j, \ell_j \mathbf{R}) && \text{for nodes } 2 \leq j \leq m+n, \\ \mathbf{Y}_o^i \mid \mathbf{Y}^{\text{pa}(i)} &\sim \mathcal{N}(\mathbf{Y}^{\text{pa}(i)}, \mathbf{P}) && \text{for observations } m+n+1 \leq i \leq m+n+n_o. \end{aligned}$$

To keep things simple, we only present details for the BM.

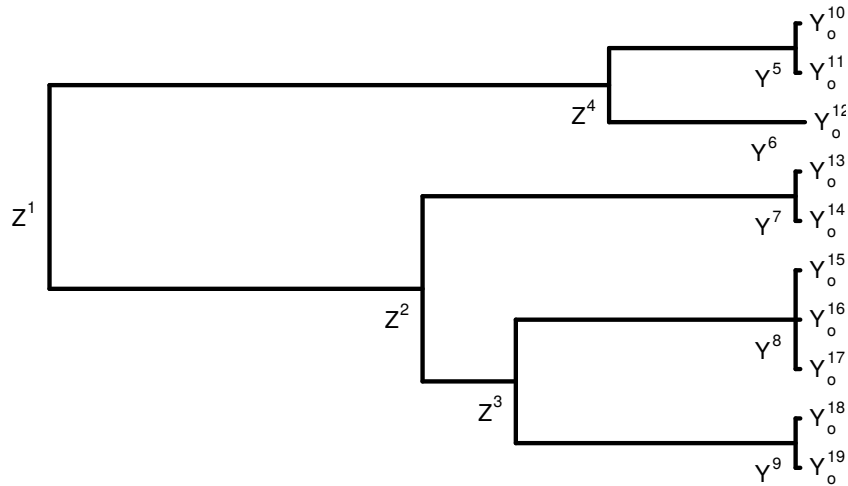


Figure 5.1.5 – A phylogenetic tree with 5 tips and 4 internal nodes, and several measurements at each tips. For instance, species 5 was measured twice, while species 6 was only measured once. Terminal branches linking tips (noted  $\mathbf{Y}$ ) with observations (noted  $\mathbf{Y}_o$ ) have length 0.

Using the graphical model as in Section 2.4, we can factorize the likelihood of the completed dataset  $p_{\theta}(\mathbf{Z}, \mathbf{Y}, \mathbf{Y}_o)$  as a product of Gaussian densities over the edges of the tree. We are hence able to write the same EM algorithm as before. As in Section 3.C,

the expectation given the observation can be written as:

$$\begin{aligned}
-2\mathbb{E}[\log p_{\theta}(\mathbf{X}) \mid \mathbf{Y}_o] &= p(m+n)\log 2\pi + p \sum_{j=2}^{m+n} \log \ell_j \\
&+ \log |\mathbf{\Gamma}| + \text{tr} \left\{ \mathbf{\Gamma}^{-1} \mathbb{V}\text{ar} \left[ \mathbf{X}^1 \mid \mathbf{Y}_o \right] \right\} + \left\| \mathbb{E} \left[ \mathbf{X}^1 \mid \mathbf{Y}_o \right] - \boldsymbol{\mu} \right\|_{\mathbf{\Gamma}^{-1}}^2 \\
&+ (m+n-1)\log |\mathbf{R}| + \sum_{j=2}^{m+n} \ell_j^{-1} \text{tr} \left\{ \mathbf{R}^{-1} \mathbb{V}\text{ar} \left[ \mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}_o \right] \right\} \\
&+ \sum_{j=2}^{m+n} \ell_j^{-1} \left\| \mathbb{E} \left[ \mathbf{X}^j - \mathbf{X}^{\text{pa}(j)} \mid \mathbf{Y}_o \right] - \boldsymbol{\Delta}^j \right\|_{\mathbf{R}^{-1}}^2 \\
&+ n_o \log |\mathbf{P}| + \sum_{i=m+n+1}^{m+n+n_o} \text{tr} \left\{ \mathbf{P}^{-1} \mathbb{V}\text{ar} \left[ \mathbf{Y}_o^i - \mathbf{Y}^{\text{pa}(i)} \mid \mathbf{Y}_o \right] \right\} \\
&+ \sum_{i=m+n+1}^{m+n+n_o} \left\| \mathbb{E} \left[ \mathbf{Y}_o^i - \mathbf{Y}^{\text{pa}(i)} \mid \mathbf{Y}_o \right] \right\|_{\mathbf{P}^{-1}}^2.
\end{aligned}$$

At the M step, the estimators for the parameters are the same than in the case without error, and we can get an estimation of the measurement error matrix in a similar way:

$$n_o \mathbf{P}^{(h+1)} = \sum_{i=m+n+1}^{m+n+n_o} \mathbb{V}\text{ar}^{(h)} \left[ \mathbf{Y}^{\text{pa}(i)} \mid \mathbf{Y}_o \right] + \left( \mathbb{E}^{(h)} \left[ \mathbf{Y}_o^i - \mathbf{Y}^{\text{pa}(i)} \mid \mathbf{Y}_o \right] \right) \left( \mathbb{E}^{(h)} \left[ \mathbf{Y}_o^i - \mathbf{Y}^{\text{pa}(i)} \mid \mathbf{Y}_o \right] \right)^T.$$

At the E step, we can write a similar upward-downward algorithm, using the graphical model presented in Figure 5.1.5. All the propagation formula we wrote can be readily applied, using the following model, adapted from Equation (3.9) of Section 3.C.2:

$$\begin{aligned}
\forall j \in \llbracket 2, m+n \rrbracket &\left\{ \begin{array}{l} \mathbb{E} \left[ \mathbf{X}^j \mid \mathbf{X}^{\text{pa}(j)} \right] = m_j(\mathbf{X}^{\text{pa}(j)}) = \mathbf{Q}_j \mathbf{X}^{\text{pa}(j)} + \mathbf{r}_j \\ \mathbb{V}\text{ar} \left[ \mathbf{X}^j \mid \mathbf{X}^{\text{pa}(j)} \right] = \boldsymbol{\Sigma}_j \end{array} \right. \\
\forall i \in \llbracket m+n+1, m+n+n_o \rrbracket &\left\{ \begin{array}{l} \mathbb{E} \left[ \mathbf{Y}_o^i \mid \mathbf{Y}^{\text{pa}(i)} \right] = \mathbf{Y}^{\text{pa}(i)} \\ \mathbb{V}\text{ar} \left[ \mathbf{Y}_o^i \mid \mathbf{Y}^{\text{pa}(i)} \right] = \mathbf{P}. \end{array} \right.
\end{aligned}$$

The missing data can be handled the same way as before.

Once implemented, this method to deal with measurement errors should improve the robustness of the algorithm to trait uncertainty. Some tests would be needed to assess the quality of the estimation of the measurement error, especially when there are only but a few species with multiple measurements. In such situations, it might be more robust to estimate the measurement error beforehand, based on some prior information on the way the data was gathered, and then to fix  $\mathbf{P}$  during the inference step (as in Ives et al. 2007).

### 5.1.3 Factor Analysis

Interestingly, the framework developed above could also be used in a factor analysis. The idea of factor analysis is to reduce the  $p$  observed traits to a smaller number  $q < p$  of



hidden “factors” that would capture the dynamic of trait evolution. The  $q$  factors would then evolve on the tree as, for instance, a BM with no correlation. The observed “real” traits would then be obtained as a linear combination of these factors, plus an error. This hierarchical model would read, using the notations and numbering defined above (see Fig. 5.1.5):

$$\begin{aligned} \mathbf{F}^1 &\sim \mathcal{N}(\boldsymbol{\mu}_F, \boldsymbol{\Gamma}_F) && \text{for the root,} \\ \mathbf{F}^j \mid \mathbf{F}^{\text{pa}(j)} &\sim \mathcal{N}(\mathbf{F}^{\text{pa}(j)} + \boldsymbol{\Delta}^j, \ell_j \mathbf{I}_q) && \text{for nodes } 2 \leq j \leq m+n, \\ \mathbf{Y}_o^i \mid \mathbf{F}^{\text{pa}(i)} &\sim \mathcal{N}(\mathbf{F}^{\text{pa}(i)} \mathbf{L}, \mathbf{P}) && \text{for observations } m+n+1 \leq i \leq m+n+n_o. \end{aligned}$$

Here,  $\mathbf{F}$  is the  $(m+n) \times q$  matrix of ancestral and current factors, that evolve according to an independent BM on the tree with shifts  $\boldsymbol{\Delta}$  (size  $(m+n) \times q$ ), and ancestral  $q$ -dimensional expectation and variance  $\boldsymbol{\mu}_F$  and  $\boldsymbol{\Gamma}_F$ .  $\mathbf{L}$  is a  $q \times p$  matrix of loadings, and  $\mathbf{P}$  is a  $p \times p$  covariance error matrix, accounting for representation and measurement errors simultaneously.

When written this way, the factor model is very similar to the BM with errors presented above, and the upward-downward algorithm for moments computation at the E step could be readily used here. Two main difficulties might arise. First, when the number of latent trait  $q$  is fixed, the M step would need to estimate  $\mathbf{P}$  and  $\mathbf{L}$ , which might cause some identifiability problems. The classical constraints on the matrix of loadings (upper-triangular matrix with positive diagonal) would need to be analysed in this context. The second problem would then be the selection of  $q$  itself. Indeed, one would need to design an adequate model selection criterion to select for the right number of shifts  $K$  and the right number of factors  $q$ . In addition, trying to infer the model for every couple  $(K, q)$  might become computationally burdensome.

We did not pursue this direction, but the model could be interesting to study. A version of it was already analysed in a Bayesian framework (without shift) by [Tolkoff et al. \(2017\)](#). We saw in Section 3.5.2 of Chapter 3 that our method was somehow sensitive to the number of traits considered in the analysis. This modelling, that tries to select for only a small number of independent factors, could make the method more robust to this kind of trait overflow.

## 5.2 Convergence and Sparsity

In Section 1.5.3.2 (see also Sections 2.C.4 and 3.C.3), we saw that, using the linear formulation (see e.g. Equation (3.2)) of the problem, a (group)-lasso penalty could be used to select for non-zero lines of the shift matrix  $\boldsymbol{\Delta}$ , and hence the position of the shifts. Here, we show how different penalties can induce some other desired structural constraints on the shifts, such as convergence, or sparsity in the number of shifted traits. In all this section, for the sake of clarity, we assume that all the tips are independent (i.e. related by a star-tree). If the tree-induced correlations are known, it is straightforward to reduce theoretical considerations to that simple case (see Section 3.C.3).

### 5.2.1 Convergence and Fused-ANOVA

In all the developments above, we made the strong assumption that there was no homoplasy (see Sections 2.3.1 and 3.2.1). However, in many cases, evolutionary biologists are

actually interested in these convergence phenomena (see e.g. [Mahler et al. 2013](#); [Aristide et al. 2016](#)). The definition of the term “convergence” is fluctuating in the literature (see [Stayton 2015](#) for a review). Here, we simply define convergence as two distinct clades having the same optimal trait values. In our framework, as any shift produces a new optimum, this situation is forbidden, and the two clades will typically be found to have very similar, but still different, optima.

If we want to account for convergence, we need to count the complexity of a model not by its number of shifts, but by its number of “regimes”, i.e. the number of truly different optima, no matter how many shift. This means that we need to find a new way of navigating through the models. Indeed, in our EM approach, we split the space of models by their number of shifts, and this partition might not be relevant anymore. In addition, we cannot use the results exposed in Section 1.1 for model selection, as we did in Section 2.3.1. Another penalty criterion, based on a yet to compute complexity, would also be required.

As an alternative to the rigorous, full likelihood approach, we propose here a simple way to detect convergence *a posteriori*, using a fused-ANOVA penalty ([Chiquet et al., 2017](#)), that uses a penalty similar to the fused-lasso (see Section 1.5.3.2) to ensure that only a few regimes are actually different. Assume that we found a homoplasy-free solution with  $K$  shifts thanks to our previous method, so that we can classify all the tips of the tree in one of the  $K+1$  groups created. More specifically, we know a map  $\kappa : \llbracket 1, n \rrbracket \rightarrow \llbracket 1, K+1 \rrbracket$  that to a tip  $i$  associate its group  $\kappa(i)$ . Each group  $k$  has  $n_k$  members, with  $\sum_{k=1}^{K+1} n_k = n$ . Let  $\mathbf{Y}$  be the  $n \times p$  matrix of observations, and  $\boldsymbol{\beta}$  the  $(K+1) \times p$  matrix of distinct optimal values of the regimes. Then, the fused-ANOVA is the solution to the following minimization problem ([Chiquet et al., 2017](#)):

$$\operatorname{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^{(K+1) \times p}} \left\{ \sum_{i=1}^n \|\mathbf{Y}^i - \boldsymbol{\beta}^{\kappa(i)}\|_2^2 + \lambda \sum_{\substack{1 \leq k, l \leq K+1 \\ k \neq l}} \omega_{k,l} \Omega(\boldsymbol{\beta}^k - \boldsymbol{\beta}^l) \right\}. \quad (5.1)$$

Here,  $\Omega$  is a given norm on  $\mathbb{R}^p$ , that penalizes large differences between regimes values, and  $(\omega_{kl})_{1 \leq k, l \leq K+1}$  are weights, that need to be chosen properly. [Chiquet et al. \(2017\)](#) offer several forms for these norm and weights that have some good theoretical properties. In particular, if  $\omega_{kl} = n_k n_l$  and  $\Omega$  is an  $\ell_q$ -norm, with  $q \in \mathbb{N}^* \cup \{+\infty\}$ , then the path of solution  $\{\boldsymbol{\beta}(\lambda) : \lambda > 0\}$  has no split. In other words, when  $\lambda$  increases, the penalty becomes stronger, so that groups are progressively merged together, in such a way that if two groups are merged for a given  $\lambda_1$ , then they are still merged for any  $\lambda_2 \geq \lambda_1$ .

The choice of  $\lambda$  itself remains a problem, as we don't have any good model selection criterion available yet. Cross validation could be used as a first guess.

An alternate way is to use the lasso penalty from the beginning, instead of the EM algorithm. Using the same notations, the corresponding minimization problem is:

$$\operatorname{argmin}_{\Delta \in \mathbb{R}^{(m+n) \times p}} \left\{ \sum_{i=1}^n \|\mathbf{Y}^i - (\mathbf{T}\Delta)^i\|_2^2 + \lambda^1 \sum_{1 \leq j \leq m+n} \omega_j^1 \|\Delta^j\|_2 + \lambda^2 \sum_{1 \leq k, l \leq n} \omega_{k,l}^2 \Omega((\mathbf{T}\Delta)^k - (\mathbf{T}\Delta)^l) \right\}.$$

However, the problem of calibrating the tuning parameters  $\lambda^1$  and  $\lambda^2$  remains.

It is also possible to write an EM to maximize the penalized likelihood directly. However, it is not clear that this new criterion can be optimized efficiently at the M step. The discrete shift location algorithm should in particular probably be revised.

## 5.2.2 Sparse Number of Shifted Traits

Another strong assumption we made in Chapter 3 is that, in the multivariate case, all the traits shift at the same time. This assumption might be impairing when the number of traits grows, and can only be valid for carefully pre-selected traits that are assumed to shift in a synchronized way. We saw on the lizard example (Section 3.5.2, Fig. 3.5.2) that the number of shifts detected tends to decrease when there are more traits included in the analysis. Intuitively, the more traits one includes, the more “unlikely” they are to shift together, and the more “costly” including a shift becomes.

As in the previous section, designing an exact EM algorithm to deal with this problem might be difficult. Indeed, models in this setting would be indexed not only by the number of shifts, but also by the number of non-zero components in each of them, so that the maximization step might become prohibitive, and the model selection inefficient.

Again, one easy way to deal with this problem would be to simply add a sparsity inducing penalty. The following minimization problem could be solved:

$$\operatorname{argmin}_{\Delta \in \mathbb{R}^{(m+n) \times p}} \left\{ \sum_{i=1}^n \|\mathbf{Y}^i - (\mathbf{T}\Delta)^i\|_2^2 + \lambda^1 \sum_{1 \leq j \leq m+n} \omega_j^1 \|\Delta^j\|_2 + \lambda^2 \|\Delta\|_1 \right\}.$$

Where we used a “sparse-group-sparse” penalty (see Section 1.5.3.2). As previously, the problem of selecting the two tuning parameters remains.

## 5.3 Non-Ultrametric Trees

In Chapter 2 and 3, we made the assumption that the tree was ultrametric. This assumption is reasonable in many cases, as one usually only has access to trait measurements for extant species. However, such a framework does not allow us to deal with *fossil* data points that might be available for some traits of some species. When available, these fossils provide the researcher with unique insights on the process, and should not be ignored. For instance, and as expected, incorporating them in an ancestral trait reconstruction study is known to significantly improve the estimates (Finarelli et al., 2006; Albert et al., 2009). Non-ultrametric trees are also common in some other biological fields, such as virology, where organisms evolve very fast, and can be sampled over long periods of time (Faria et al., 2011).

In the developments above, this assumption was used twice: for the identifiability and the computation of the models complexity in Chapter 2; and for tree re-scaling in Chapter 3. Note that in both problems, this assumption allowed us to somehow reduce the OU to a more manageable BM.

As the OU has a dynamical component, we expect that having un-synchronized data points will generally improve the identifiability of the model. This has already been observed on simulation studies (Slater et al., 2012). In the next two sections, we review the changes induced by non-ultrametric trees, and expose some possible solutions to deal with such trees.

### 5.3.1 Identifiability and Model Selection

When assessing identifiability in Chapter 2, the fact that we only dealt with ultrametric trees allowed us to reduce each regime to a single color (through Lemma 2.2.1 and

Proposition 2.3.2), and hence to use the combinatorial results of Section 1.1. This is not true anymore, and has several consequences, listed below.

**Identifiability of the Root Value.** In Section 1.4.1.2, we saw that, on an ultrametric tree with height  $h$ , the root expectation  $\mu$  and the root optimal value  $\beta_0$  of an OU (without any shift) only appeared on the tips through the linear combination  $\lambda = e^{-\alpha h}\mu + (1 - e^{-\alpha h})\beta_0$ , so that only  $\lambda$  was identifiable, and not  $\beta_0$  and  $\mu$  separately (see Fig. 1.4.3). On a non-ultrametric tree, we have at least two tips  $i$  and  $j$  such that  $t_i \neq t_j$ , so that the expectations on these two tips are, respectively,  $e^{-\alpha t_i}\mu + (1 - e^{-\alpha t_i})\beta_0$  and  $e^{-\alpha t_j}\mu + (1 - e^{-\alpha t_j})\beta_0$ . Formally, we hence get a system with two independent equations, so that the two parameters become identifiable.

**Identifiability of Shifts Position.** We use a simple example to show how a configuration that was not identifiable on an ultrametric tree can become identifiable on a non-ultrametric tree.

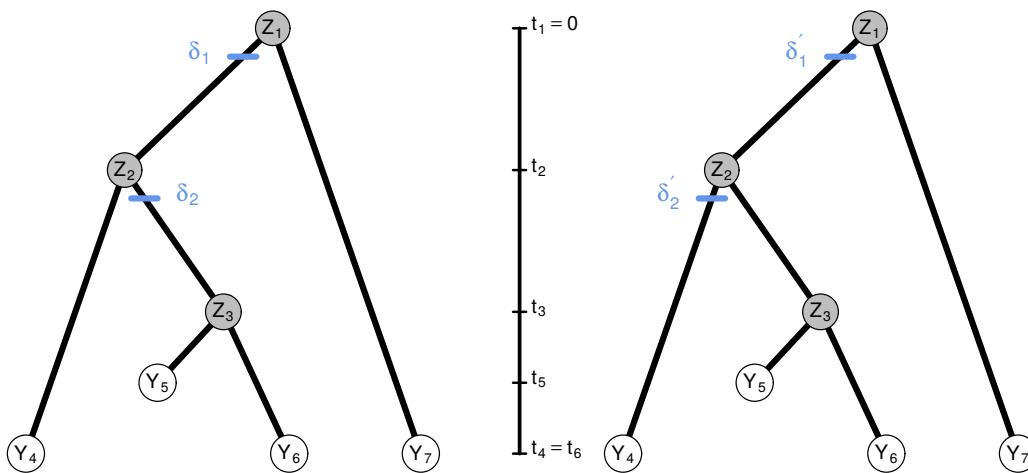


Figure 5.3.1 – A non-ultrametric tree, with two shifts configurations that are equivalent for a BM, but not for an OU.

The two shifts configurations presented in Figure 5.3.1, are not identifiable for a BM (or an OU on an ultrametric tree), but actually are on a non-ultrametric tree. Indeed, using the notation of Figure 5.3.1, we have, in the first configuration (taking  $\mu = \beta_0 = 0$ , and using Equation (2.5) of Section 2.2.3):

$$\begin{cases} \mathbb{E}[Y_4] = \delta_1(1 - e^{-\alpha t_4}) \\ \mathbb{E}[Y_5] = \delta_1(1 - e^{-\alpha t_5}) + \delta_2(1 - e^{-\alpha(t_5-t_2)}) \\ \mathbb{E}[Y_6] = \delta_1(1 - e^{-\alpha t_6}) + \delta_2(1 - e^{-\alpha(t_6-t_2)}) \end{cases} \quad (5.2)$$

and, in the second configuration:

$$\begin{cases} \mathbb{E}[Y_4] = \delta'_1(1 - e^{-\alpha t_4}) + \delta'_2(1 - e^{-\alpha(t_4-t_2)}) \\ \mathbb{E}[Y_5] = \delta'_1(1 - e^{-\alpha t_5}) \\ \mathbb{E}[Y_6] = \delta'_1(1 - e^{-\alpha t_6}) \end{cases} \quad (5.3)$$

To prove that the two configuration produce the same trait distribution at the tips, we need to show that the induced expectations at the tips are equal (shifts do not impact the variances). If the tree is ultrametric, then  $t_4 = t_5 = t_6$ , and the last two equations are the same. We can hence take:

$$\begin{cases} \delta'_1 = \delta_1 + \delta_2 \frac{1 - e^{-\alpha(t_5-t_2)}}{1 - e^{-\alpha t_5}} \\ \delta'_2 = -\delta_2 \frac{1 - e^{-\alpha(t_5-t_2)}}{1 - e^{-\alpha t_5}} \frac{1 - e^{-\alpha t_4}}{1 - e^{-\alpha(t_4-t_2)}} = -\delta_2 \end{cases}$$

to get two configurations that produce the same distribution of the trait at the tips (note that when  $\alpha \rightarrow +\infty$ , then  $\delta'_1 = \delta_1 + \delta_2$  and  $\delta'_2 = -\delta_2$ , so that we find back the BM solution). However, when  $t_5 \neq t_6$ , then the three equations become linearly independent, and having only two free parameters, do not have any solution. Intuitively, when the two branches below the shift  $\delta_2$  do not have the same length, because of the non-linearity of the actualization factor, it is impossible to mimic the effect of the shifts with a different value of  $\delta_1$ .

Note that this same example can be used to show that even non parsimonious shift configurations can become identifiable on a non-ultrametric tree. On Figure 5.3.2, we

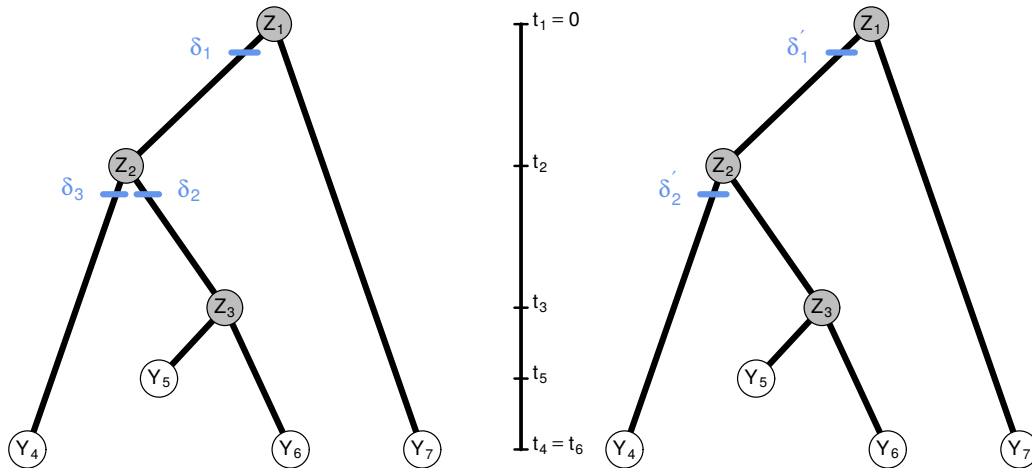


Figure 5.3.2 – A non-ultrametric tree, with a “non parsimonious” solution on the left that cannot be reduced to the “parsimonious” one on the right for an OU.

show a non-parsimonious solution with three shifts, that used to be forbidden, as it could be replaced by a configuration with only two shifts. Writing equations similar to (5.2) and (5.3), we can show that the system of equation only has a solution when the tree

is ultrametric. In general, it is hence impossible to get the tips trait distribution on the left with only two shifts.

This has an important implication on the linear model view of the problem. Recall from Equation 2.5 that, for a univariate OU model, the vector of traits  $\mathbf{Y}$  measured at the tips of a tree with  $n$  tips and  $m$  internal nodes, is such that:

$$\mathbf{Y} = \mathbf{T}(\alpha)\Delta + \mathbf{E}$$

where  $\mathbf{T}(\alpha) = \mathbf{T} - \mathbf{A}\mathbf{T}\mathbf{B}$ , with  $\mathbf{T}$  the incidence matrix of the tree, and  $\mathbf{A} = \text{Diag}(e^{-\alpha t_i}, m+1 \leq i \leq m+n)$ ,  $\mathbf{B} = \text{Diag}(0, e^{\alpha t_{\text{pa}(i)}}, 2 \leq i \leq m+n)$  diagonal matrices of sizes  $n$  and  $m+n$  representing the actualization factors.  $\mathbf{E}$  is an error vector with a variance structure defined by the tree and the OU model. Note that, when the tree is ultrametric with height  $h$ , then  $\mathbf{A} = e^{-\alpha h}\mathbf{I}_n$ , and we get  $\mathbf{T}(\alpha) = \mathbf{T}\mathbf{W}(\alpha)$ , with  $\mathbf{W}(\alpha)$  invertible, as in Equation (2.6). In general however,  $\mathbf{T}$  cannot be factorized, so that  $\mathbf{T}(\alpha)$  does not directly inherit its properties from  $\mathbf{T}$ .

In particular, Proposition 2.3.6 that ensured us, in the ultrametric case, that a shift configuration was parsimonious if and only if the corresponding columns of the regression incidence matrix  $\mathbf{T}$  were linearly independent cannot be used anymore. On a non-ultrametric tree, it is easy to show that only the direct implication is still true: if a shift configuration is parsimonious, then the corresponding columns of  $\mathbf{T}(\alpha)$  are linearly independent. As shown by the example above, the converse statement is false.

This leads us to a new definition of what is an *acceptable* model with  $K$  shifts: it is not just a parsimonious model, but a model that is such that the corresponding columns of  $\mathbf{T}(\alpha)$  are linearly independent. A general study of these models would be needed. As in Section 2.3, we would like to know, first, the size of an equivalent class (as in Propositions 2.3.3), and, second, the number of truly different models one has for a fixed number of shifts (as in Proposition 2.3.5). From the simple examples above, we see that adding fossils, as expected, allows us to choose between different scenarios that were not identifiable before. Hence, the number of distinguishable models should be bigger. In addition, we can have the intuition that identifiability will highly depend on the number of non-synchronized tips below each shift. It is hence probable that no general formula can be easily derived in this case.

**Model Selection.** The fact that we are not able in general to compute the true number of different models might impair our model selection procedure, that took this number explicitly into account (see Proposition 2.4.1 in Section 2.4.2). One simple solution is to use the natural bound we have of this number of models. For a tree with  $m$  internal nodes and  $n$  tips, the (unknown) number of different models  $N_K^I$  is bounded by the number of possible allocations of  $K$  shifts on the  $m+n-1$  internal branches:

$$N_K^I \leq \binom{m+n-1}{K}.$$

This bound is actually sufficient to derive a penalty that fulfills the conditions of the LINselect model selection (Theorem 1.5.2), recalled in the introductory chapter (Section 1.5.4), and used in Chapter 2 (Section 2.4.2). To see this, let's assume, for the sake of clarity, that we are in the univariate case, and that we already corrected for the phylogeny (for a known  $\alpha$ ). We are then in the classical linear model setting for model selection:

$$\mathbf{Y} = \mathbf{s} + \gamma\mathbf{E} \quad , \quad \mathbf{E} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$$

with models  $(S_\eta)_{\eta \in \mathcal{M}}$  that are linear sub-spaces of  $\mathbb{R}^n$  defined by the shift positions on the tree, with dimension  $D_\eta = |\eta| = K_\eta + 1$ .

The crucial point, when applying Theorem 1.5.2, is to find a set of weights  $(L_\eta)_{\eta \in \mathcal{M}}$  such that:

- $\sum_{\eta \in \mathcal{M}} (D_\eta + 1)e^{-L_\eta} = \Omega' < +\infty$ ;
- $L_\eta$  is of the order than  $D_\eta$ .

Taking the proof of Proposition 2.4.1 (see Section 2.E), we can see that this can be achieved by just replacing  $N_K^I$  by its upper bound in the expression of the weights:

$$L_K = \log \binom{m+n-1}{K} + 2\log(K+2), \forall K \in \llbracket 0, p-1 \rrbracket.$$

Indeed, we get:

$$\Omega' = \sum_{\eta \in \mathcal{M}} (D_\eta + 1)e^{-L_\eta} = \sum_{K=0}^{p-1} N_K^I (K+2)e^{-L_K} \leq \sum_{K=0}^{p-1} \binom{m+n-1}{K} (K+2)e^{-L_K} \leq \sum_{K=0}^{p-1} \frac{1}{K+2} \leq \log(n)$$

and, using the same inequalities as in Section 2.E:

$$L_K = \log \binom{n+m-1}{K} + 2\log(K+2) \leq (K+1)(2 + \log(2) + \log(n)) \leq p(2 + \log(2) + \log(n))$$

All the subsequent derivations of Section 2.E then hold. This proves that Proposition 2.4.1 still holds when we choose a more stringent set of weights  $L_\eta$ . The practical properties of such a penalty would need to be evaluated on simulations. For example, because we use an upper bound of the number of models, we expect that the criterion might have a tendency to over-penalize models with a great number of shifts.

### 5.3.2 Inference of the OU

We saw in previous chapters that the selection strength parameter  $\alpha$ , that represents a dynamic aspect of the evolution model, was rather poorly estimated when using data coming from a single time point, at the tips of an ultrametric tree (see Section 2.5, Fig. 2.5.2 and Section 3.4). Having access to fossil data points should break synchronicity, and hence help us estimate this  $\alpha$  parameter. In Slater et al. (2012), the authors indeed showed on some simulations that including some fossil information on the analysis helped to discriminate between several dynamical models of evolution, including the BM, AC/DC, and OU.

However, the inference method of Chapter 3 heavily relies on the equivalency between an OU and a BM on a re-scaled tree, that only holds for an ultrametric tree (see Sections 1.4.2.4 and 3.2.4). We hence need to design a new method to explicitly deal with the OU in the multivariate case. Using the EM framework, we can readily see that the E step is not a problem. In Section 3.C, we indeed described a quite general upward-downward algorithm, that can deal with a broad class of Gaussian processes, including the general OU, on any dated tree.

The M step would then be the limiting point. In the univariate case, we showed that the efficient shift allocation algorithm that worked for the BM could not be used anymore,



because of the non-independent increments of the OU (see Sections 2.4, 2.C.3 and 3.C). We hence resorted to heuristics in order to raise, if not maximize, the objective function at each step of the EM. This leads to a Generalized EM algorithm (GEM Dempster et al., 1977), that behaved well in the univariate case, but was quite slow compared to the re-scaling tree (see Section 2.5). We sketch here two simple heuristics that could be used in the multivariate case for the OU. The first one, based on a lasso penalty, is a direct adaptation from the univariate case. The second one, based on a binary segmentation algorithm, is a promising and possibly fast alternative.

**A Lasso Based Heuristic.** In the multivariate setting, with a scalar OU (scOU), an heuristic based on a lasso penalty can be readily adapted from the univariate case, developed in Section 2.C.3. At iteration  $(h+1)$  of the M step, the optimal shift location boils down to the minimization of the following sum of costs over all the branches of the tree:

$$C_{OU} = \sum_{j=1}^{m+n} \left(1 - e^{-2\alpha\ell_j}\right)^{-1} \left\| \mathbb{E}^{(h)}[\mathbf{D}^j \mid \mathbf{Y}] - E_j \boldsymbol{\beta}^j \right\|_{(\Gamma^{(h)})^{-1}}^2$$

where  $\mathbf{D}^j = \mathbf{X}^j - e^{-\alpha\ell_j} \mathbf{X}^{\text{pa}(j)}$  and  $E_j = (1 - e^{-2\alpha\ell_j})$  (we formally set  $\ell_1 = +\infty$ ).  $\Gamma^{(h)}$  is the root stationary variance estimate obtained at the previous M step, and  $\mathbb{E}^{(h)}$  is taken according to the previous estimates of the parameters, and was computed at the E step.  $\boldsymbol{\beta} = \mathbf{U}\boldsymbol{\Delta}$  is the  $(m+n) \times p$  matrix of optimal values on each branches. The optimization must be conducted by finding the  $K$  non-zero lines of  $\boldsymbol{\Delta}$ , and their associated values. Define matrix  $\mathbf{F}$  (size  $(m+n) \times p$ ) and diagonal matrix  $\mathbf{A}$  (size  $m+n$ ) by, for any  $1 \leq j \leq m+n$ :

$$\begin{aligned} \mathbf{F}^j &= \left(1 - e^{-2\alpha\ell_j}\right)^{-1/2} \mathbb{E}[\mathbf{D}^j \mid \mathbf{Y}] \\ \mathbf{A}_{jj} &= \left(1 - e^{-2\alpha\ell_j}\right)^{-1/2} E_j. \end{aligned}$$

Then, the objective function to be minimized can be re-written as:

$$C_{OU} = \sum_{j=1}^{m+n} \left\| \mathbf{F}^j - (\mathbf{A}\mathbf{U}\boldsymbol{\Delta})^j \right\|_{(\Gamma^{(h)})^{-1}}^2. \quad (5.4)$$

This sum can be seen as the least squares minimization associated with the linear regression model:

$$\mathbf{F} = \mathbf{A}\mathbf{U}\boldsymbol{\Delta} + \mathbf{E}, \quad \text{with } \mathbf{E}^j \sim \mathcal{N}(\mathbf{0}, \Gamma^{(h)}) \text{ i.i.d.}$$

A sparse estimation of  $\boldsymbol{\Delta}$  can then be obtained using a lasso regression, using a sparse-group penalty, as explained in Section 1.5.3.2.

**A Binary Segmentation Based Heuristic.** Segmentation, or shift detection for data points displayed on a line, has received a lot of attention these last few decades (see e.g. Eckley et al. 2011; Fryzlewicz 2014 for a review). Among all the numerous algorithms designed to tackle this problem, the binary segmentation heuristic, although very simple, has proven to be very efficient (Rigaille, 2015). It is rather easy to implement, runs quite fast, and finds very reasonable solutions, when compared to the exact one.

In classical segmentation, we assume that we have an ordered sequence data  $\mathbf{Y}_{1:n} = (Y_1, \dots, Y_n)$ , and that we are able to compute a *cost*  $C(Y_{r:s})$  of any continuous sequence of data (with  $1 \leq r \leq s \leq n$ ). The cost might be associated to the least squares, or the



maximum likelihood, depending on the underlying model for the observations. The goal is to find the  $K$  change points  $\mathbf{v}_{1:K} = (v_1, \dots, v_K)$  such that

$$\sum_{k=1}^{K+1} C(\mathbf{Y}_{(v_{k-1}+1):v_k})$$

is minimal (where  $v_0 = 0$ , and  $v_{K+1} = n$ ). Classical binary segmentation relies on the fact that, given a segment of data  $\mathbf{Y}_{r:s}$  (with  $1 \leq r < s \leq n$ ), we know how to find  $\hat{v}_{r:s} \in \llbracket r, s \rrbracket$  that minimizes the total cost of the segment by introducing one change point:

$$\hat{v}(\llbracket r, s \rrbracket) = \operatorname{argmin}_{v \in \llbracket r, s \rrbracket} \left\{ C(\mathbf{Y}_{r:v}) + C(\mathbf{Y}_{(v+1):s}) \right\} \leq C(\mathbf{Y}_{r:s}).$$

Denote also  $\hat{C}(\llbracket r, s \rrbracket) = \min_{v \in \llbracket r, s \rrbracket} \left\{ C(\mathbf{Y}_{r:v}) + C(\mathbf{Y}_{(v+1):s}) \right\}$ , with  $\hat{C}(\llbracket r, r \rrbracket) = +\infty$  (a unique point cannot be split). The idea is then to do a step-wise optimization, that adds shifts one by one: we first find  $\hat{v}(\llbracket 1, n \rrbracket)$  the best split point on the all segment, and then split one of the two segments produced, getting the best split between  $\hat{v}(\llbracket 1, \hat{v}(\llbracket 1, n \rrbracket) \rrbracket)$  and  $\hat{v}(\llbracket \hat{v}(\llbracket 1, n \rrbracket) + 1, n \rrbracket)$ , and so on, until  $K$  shifts are found. See Algorithm 5.3.1 for a more formal description (this presentation is inspired by Eckley et al., 2011). Note

---

**Algorithm 5.3.1** Binary Segmentation

---

```

 $\mathcal{S} \leftarrow \{\llbracket 1, n \rrbracket\}$ 
for  $k \in \llbracket 1, K \rrbracket$  do
   $\hat{I} \leftarrow \operatorname{argmin}_{I \in \mathcal{S}} \hat{C}(I)$ 
   $v_k \leftarrow \hat{v}(\hat{I})$ 
   $\mathcal{S} \leftarrow (\mathcal{S} \setminus \hat{I}) \cup \{\llbracket \min(\hat{I}), v_k \rrbracket, \llbracket v_k + 1, \max(\hat{I}) \rrbracket\}$ 
end for
return  $\mathbf{v} = (v_1, \dots, v_K)$ 

```

---

that this algorithm has no guaranty to converge to the global minimum. An efficient implementation of it relies on the efficient computation of the costs  $\hat{C}(I)$  for the needed intervals  $I$  of  $\llbracket 1, n \rrbracket$ .

In our problem, the data points do not lie on a segment, but at the nodes of a tree. Hence, instead of considering the cost of intervals, we need to consider the cost of *connex* sets of nodes on the tree. Let  $I$  be such a connex set of nodes. Using Equation (5.4), the cost of  $I$  is obtained by minimizing the least squares in the optimal value  $\hat{\beta}_I$  that is common to all the nodes in  $I$ :

$$\begin{aligned} C(I) &= \sum_{j \in I} \|\mathbf{F}^j - A_{jj} \hat{\beta}_I\|_{(\Gamma^{(h)})^{-1}}^2 \\ &= \sum_{j \in I} \|\mathbf{F}^j\|_{(\Gamma^{(h)})^{-1}}^2 - \left( \sum_{j \in I} A_{jj}^2 \right) \|\hat{\beta}_I\|_{(\Gamma^{(h)})^{-1}}^2 \quad \text{with} \quad \hat{\beta}_I = \left( \sum_{j \in I} A_{jj}^2 \right)^{-1} \sum_{j \in I} A_{jj} \mathbf{F}^j. \end{aligned}$$

As the nodes are naturally ordered by the tree, finding the best split of a connex subset  $I$  into two connex subsets  $I_{\hat{v}(I)}$  and  $I_{-\hat{v}(I)}$  using the costs above is straightforward. The previous binary splitting heuristic could then be applied to the tree. The efficiency of the algorithm will depend on how fast we can compute the costs involved. Because, on a

tree, a connex subset of nodes  $I$  is just a subtree, minus one or several sub-subtrees, all the sums involved can be computed from the quantities  $\sum_{j \in T_i} A_{jj}^2$  and  $\sum_{j \in T_i} A_{jj} \mathbf{F}^j$ , where  $T_i = \text{des}(i)$  is the set of descendants of node  $i$  (with  $1 \leq i \leq m+n$ ). These base sums could be computed in one postorder traversal of the tree.

We only sketched the heuristic here, and some work would be needed to figure out the details of the algorithm, and find an efficient way to implement it. Note that we wrote here the costs for a scalar OU, but that similar expressions could be obtained for the general OU (starting from expressions given in Section 1.A.3). This binary heuristic might hence also be useful for the extension of the method to a full OU.

The two methods presented above are heuristics: they raise, if not maximize, the objective function at each M step. In linear segmentation problems, *dynamic programming* can be used to minimize the sum of costs efficiently (see e.g. Lebarbier, 2005). It might be possible to adapt these kind of methods to segmentation on a tree, using *nonserial* dynamic programming (Bertele & Brioschi, 1972). We mention this possible direction to be comprehensive, but we did not look much into it yet.

## 5.4 Sampling Scheme and Missing Data

One of the strengths of our method is that it can readily handle missing data. In Section 3.4 (see Fig. 3.4.6), we showed that the method was robust to a great proportion of randomly chosen missing data. However, we did not study the impact that *structured* missing data could have on the analysis. A simple example that might come in mind is a trait missing for an entire clade of the tree. In that case, any shift occurring somewhere on the clade is likely to be missed by any shift detection method. Such a pattern might be due to the lack of information collected on this given clade, or to the fact that this particular trait is difficult to measure on the organisms of the clade, for instance because of its extremely small values. Some statistical tools have been developed to explicitly model the process giving rise to missing data. In this last section, we briefly recall the main ideas of these methods, and show in an informal way how they can be included in our framework.

**Statistical Framework.** The data *sampling scheme* describes the way the data was collected. In the statistical literature, sampling schemes are usually classified into three categories, based on the kind of structure of missing data they produce (Rubin, 1976; Little & Rubin, 2002). The are defined as follows.

**Definition 5.4.1** (MCAR, MAR, NMAR, Little & Rubin 2002, Eq. (1.1) to (1.3)). Let  $\mathbf{Y}$  be a  $n \times p$  be the matrix of the complete data, with  $n$  individuals and  $p$  traits, and  $\mathbf{M}$  the  $n \times p$  missing data indicator matrix: for any individual  $i$ ,  $1 \leq i \leq n$ , and trait  $l$ ,  $1 \leq l \leq p$ ,  $Y_{il}$  is actually measured if and only if  $M_{il} = 1$ . Assume that the conditional distribution  $p_\psi(\mathbf{M} | \mathbf{Y})$  of  $\mathbf{M}$  given  $\mathbf{Y}$  is described by parameters  $\psi$ .

The data are called missing completely at random (MCAR) if missingness does not depend on the data:

$$p_\psi(\mathbf{M} | \mathbf{Y}) = p_\psi(\mathbf{M}).$$

Denoting  $\mathbf{Y}_{\text{miss}}$  the missing components of  $\mathbf{Y}$ , and  $\mathbf{Y}_{\text{obs}}$  the observed components, the data are called missing at random (MAR) if missingness only depends on the observed data:

$$p_\psi(\mathbf{M} | \mathbf{Y}) = p_\psi(\mathbf{M} | \mathbf{Y}_{\text{obs}}).$$

Finally, the mechanism is called not missing at random (NMAR) if the distribution of  $\mathbf{M}$  also depends on the missing values of  $\mathbf{Y}$ .

In other words, if the mechanism is MCAR, the sampling scheme does not depend on the values of the traits whatsoever: there is an independent rule controlling which traits of which species are measured, on the whole tree. At the other side of the spectrum, if the mechanism is NMAR, then the sampling scheme does depend on the values of observed *and* missing traits. This would be the case in the simple example presented above, where a trait that is “small” is less likely to be measured.

The great strength of this methodology is that we can explicitly take into account any information we have on the sampling scheme, in order to improve our estimation of the parameters. Before going any further, let’s take an example of such a NMAR sampling rule.

**Definition 5.4.2** (Size Censored Sampling). Assume that each trait  $l$ ,  $1 \leq l \leq p$ , has a smaller probability of being measured if it is below a given threshold  $c_l$ . The conditional sampling distribution  $p_\psi(\mathbf{M} | \mathbf{Y})$  is then defined by:

$$\begin{cases} \mathbb{P}[M_{il} = 1 | Y_{il} < c_l] = \rho_s \\ \mathbb{P}[M_{il} = 1 | Y_{il} \geq c_l] = \rho_l \end{cases}$$

where  $0 \leq \rho_s \leq \rho_l \leq 1$  are the probability of the traits being measured, when they are smaller or larger than the threshold.

One can imagine many other sampling schemes. In designing one, the researcher should try, first, to mimic the actual sampling process going on, and, second, to choose one that can lead to tractable computations in its subsequent analysis. If more down to earth, the second condition is crucial for a method to be tractable. In the following, we sketch the main points of the inference method, and show how it can be applied to the problem of shift detection.

**General Setting.** When dealing with an explicit sampling scheme, the quantity of interest is not the likelihood of the sole data  $p_\theta(\mathbf{Y}_{\text{obs}})$  anymore, but the joint likelihood  $p_{\theta,\psi}(\mathbf{Y}_{\text{obs}}, \mathbf{M})$  of the data and the sampling design. This quantity needs to be optimized in  $\theta$  and  $\psi$ , that lie in a product space.

Recall that  $\mathbf{Z}$  is the  $m \times p$  matrix of un-observed traits at the internal nodes of the tree. When the sampling scheme was not accounted for, we saw that the completed likelihood  $p_\theta(\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{miss}}, \mathbf{Z})$  was easy to write (it could be decomposed in a product of Gaussian, see Section 2.4). Following Tabouy et al. (2017), we decompose the likelihood of the completed dataset as follow:

$$p_{\theta,\psi}(\mathbf{Y}_{\text{obs}}, \mathbf{M}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}) = p_\psi(\mathbf{M} | \mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}) p_\theta(\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}).$$

Then, setting up an EM algorithm framework, we need to look at the conditional expectation of the completed log-likelihood given the observations:

$$\begin{aligned} \mathbb{E} \left[ \log p_{\theta,\psi}(\mathbf{Y}_{\text{obs}}, \mathbf{M}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}) \mid \mathbf{Y}_{\text{obs}}, \mathbf{M} \right] &= \mathbb{E} \left[ \log p_\psi(\mathbf{M} | \mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}) \mid \mathbf{Y}_{\text{obs}}, \mathbf{M} \right] \\ &+ \mathbb{E} \left[ \log p_\theta(\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}) \mid \mathbf{Y}_{\text{obs}}, \mathbf{M} \right]. \end{aligned} \quad (5.5)$$

Note that, in general, the maximization must be conducted jointly in  $(\theta, \psi)$ , as the expectation is taken according to the distribution of  $(\mathbf{Y}_{\text{obs}}, \mathbf{M})$ , that depends on both

these parameters. However, if the sampling is MAR (or MCAR), then the right hand side of this equation simplifies to  $\log p_{\boldsymbol{\psi}}(\mathbf{M} \mid \mathbf{Y}_{\text{obs}}) + \mathbb{E}[\log p_{\boldsymbol{\theta}}(\mathbf{Y}_{\text{obs}}, \mathbf{Y}_{\text{miss}}, \mathbf{Z}) \mid \mathbf{Y}_{\text{obs}}]$  and the two terms of the sum can be optimized separately respectively in  $\boldsymbol{\psi}$  and  $\boldsymbol{\theta}$ . Furthermore, the second term of the sum is exactly the one we used when ignoring the sampling scheme. So, if the sampling is MAR, then the problem is the same as before, and the sampling scheme does not bring us any information (see also Prop. 1 in [Tabouy et al. 2017](#)).

The interesting case is then the NMAR case: the maximization problem is actually impacted by the sampling scheme, and we hope that this extra information will help us infer the parameters. Taking the example of the size censored sampling (Def. 5.4.2), we review the consequences of a NMAR sampling scheme on the EM algorithm. These developments are quite informal, and would require some more work to assess their relevance and feasibility.

**E step.** In the “upward-downward” framework of Section 3.C.2, the fact that we had no information on the trait values of the missing tips data was reflected in the degenerate Gaussian distribution we took at each tip  $i$  of the tree for  $f_{\mathbf{Y}_{\text{obs}}^i \mid \mathbf{Y}^i}(\mathbf{Y}_{\text{obs}}^i; \mathbf{a})$ . For a measured trait, it is a Gaussian with a zero variance (i.e. a Dirac on the observed value), and for an unobserved trait, a Gaussian with infinite variance. When the sampling is not independent from the trait value, the very fact that we did not measure a trait gives us an information on its value. We now need to compute the distribution of:

$$f_{\mathbf{Y}_{\text{obs}}^i, \mathbf{M}^i \mid \mathbf{Y}^i}(\mathbf{Y}_{\text{obs}}^i, \mathbf{M}^i; \mathbf{a}) = f_{\mathbf{Y}_{\text{obs}}^i \mid \mathbf{M}^i, \mathbf{Y}^i}(\mathbf{Y}_{\text{obs}}^i; \mathbf{M}^i, \mathbf{a}) f_{\mathbf{M}^i \mid \mathbf{Y}^i}(\mathbf{M}^i; \mathbf{a}).$$

The first term of the product is the same Dirac as before, and the second term is given by the sampling scheme, that defines the distribution of  $\mathbf{M}^i \mid \mathbf{Y}^i$ . However, recall that the upward-downward algorithm we wrote heavily depends on all the distributions being Gaussian like. It is not clear whether the size censored sampling scheme (Def. 5.4.2) can be cast into this framework or not. If we want to keep this algorithm, we hence would need to design a maybe more regular sampling scheme, that would give us a Gaussian like distribution for  $(\mathbf{Y}_{\text{obs}}^i, \mathbf{M}^i) \mid \mathbf{Y}^i$  (or maybe a mixture of Gaussians).

If this framework cannot be adapted, other paths might be explored. There is a vast literature describing EM algorithm adaptations when the E step is not tractable. As in [Tabouy et al. \(2017\)](#), the Variational EM algorithm might be a promising alternative (see also [Jaakkola 2001](#); [Wainwright & Jordan 2007](#); [Robin 2014](#)).

**M step.** At the M step, the maximization in  $\boldsymbol{\theta}$  and  $\boldsymbol{\psi}$  are independent, as we can see in Eq. (5.5). The estimation on  $\boldsymbol{\theta}$  can hence be carried on as previously. The maximization in  $\boldsymbol{\psi}$  will again depend on the sampling scheme chosen. For the size censored scheme, if the thresholds  $c_k$  are fixed, then the maximization in  $\rho_s$  and  $\rho_l$  is straightforward (and very similar to the “double standard sampling” of [Tabouy et al. 2017](#)). However, the optimization in these thresholds is more difficult, and one would probably need to design some heuristics to tackle it. Then again, this size censored scheme is not carved in stone, and it might be possible to think of another, more easily tractable, strategy.

## Chapitre 6

### Résumé substantiel

L'écologie évolutive a pour objet l'étude de la diversité des organismes biologiques. Pour s'y confronter, nul besoin de s'attarder sur la distance incommensurable qui sépare par exemple, d'une part, *Pyrolobus fumarii*, une archée thermophile prospérant dans la fournaise des cheminées hydrothermale sous-marines de la dorsale atlantique (Blöchl et al., 1997), et, d'autre part, *Cypripedium calceolus*, une espèce d'orchidées « délicates et charmantes, palpitantes et frileuses » (Huysmans, 1922). Il suffit de s'étendre sur une pelouse par une belle après midi de printemps, et de s'abîmer dans la contemplation des diverses parures arborées par les élytres des nombreux membres de la famille des *Coccinellidae*, depuis les plus courantes en Europe, comme *Coccinella septempunctata*, rouge avec sept points noirs, ou *Psyllobora vigintiduopunctata*, jaune avec vingt points noirs, jusqu'à l'américaine *Brachiacantha ursina*, noire avec dix points jaunes<sup>1</sup>.



Lorsque l'on s'intéresse aux variations exhibées par un trait au sein d'un groupe d'espèces, l'une des questions principale que l'on peut se poser est celle du rôle joué par le *hasard*. Celui-ci suffit-il à lui seul pour expliquer toute la diversité observée ? Ou bien faut-il lui chercher d'autres causes, comme des contraintes environnementales, géographiques ou climatiques ? Toute tentative de réponse à ces questions passe nécessairement par une définition préalable de cette notion de hasard. Ne voir dans celui-ci qu'un coup de dès instantané constitue un prémisses à des raisonnements potentiellement spéculatifs. « L'univers m'embarrasse, et je ne puis songer / Que cette horloge existe et n'ait point d'horloger. » s'exclamait Voltaire (1772), et, à sa suite, tous les partisans de la théorie du « Grand Horloger » au 18<sup>ème</sup> siècle (voir également Rousseau, 1762a; Paley, 1802). Une hypothèse qui peut sembler de prime abord naturelle est de supposer que les traits phénotypiques sont apparus pour chaque espèce aléatoirement et indépendamment. Cette

<sup>1</sup>Crédits photographiques : voir note page 15.

hypothèse ne tient cependant pas à un examen approfondi, et peut conduire à des conclusions erronées. C'est qu'elle ne prend pas en compte le fait que les espèces actuelles ne sont précisément *pas* indépendantes les unes des autres. D'après la théorie de l'évolution (Darwin, 1859), il existe en effet un *arbre phylogénétique* reliant toutes les espèces entre elles, et permettant d'expliquer leurs relations de parenté. Il est alors naturel de faire l'hypothèse que les traits de deux espèces proches, c'est-à-dire dont l'ancêtre commun est relativement récent, seront plus semblables que ceux de deux espèces éloignées, dont la relation de parenté est plus distante. L'un des objectifs principaux des *Méthodes Phylogénétiques Comparatives* est de rendre explicites les hypothèses faites sur l'évolution des traits au cours du temps, afin de proposer un *modèle nul* raisonnable pour la répartition actuelles de ces traits parmi les espèces.

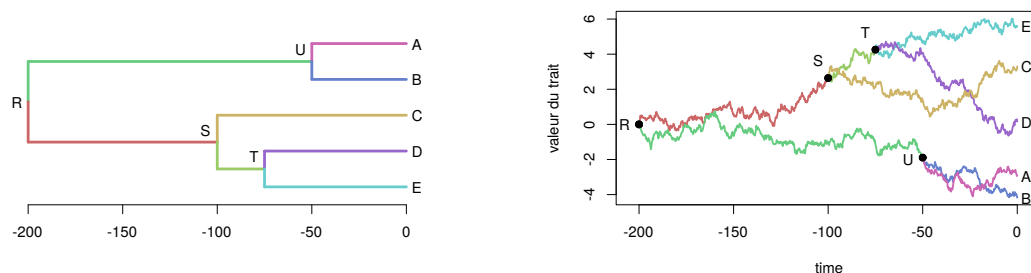
### Méthodes Phylogénétiques Comparatives

Les Méthodes Phylogénétiques Comparatives (PCM, de l'anglais *Phylogenetic Comparative Methods*) utilisent un modèle défini par deux composantes principales. Premièrement, on suppose que l'on a accès à un arbre phylogénétique daté liant les espèces actuelles entre elles. Cet arbre donne les relations de parentés, ainsi que les dates des événements de spéciations qui ont eu lieu au cours de l'histoire. Deuxièmement, on choisit un modèle dynamique décrivant l'évolution des traits quantitatifs considérés au cours du temps. Les traits en question sont des caractères continus, comme le poids, la taille, ou encore la couleur d'une fleur. Si l'on fait l'hypothèse que l'arbre phylogénétique est donné, le « hasard » est entièrement défini par ce modèle d'évolution des traits au cours du temps, que l'on choisit souvent dans une classe de *processus stochastiques*. Un tel processus permet de *quantifier* les variations du trait modélisé. Le processus stochastique le plus simple que l'on peut envisager est le *mouvement brownien* (BM, de l'anglais *Brownian Motion*). Un trait suivant un tel modèle d'évolution n'a aucune tendance, et a des incréments gaussiens indépendants.

La distribution attendue du trait dans la population d'espèces actuelles est alors obtenue en combinant ces deux ingrédients de la manière suivante. Le trait d'une espèce ancestrale donnée évolue au cours du temps comme un BM. Lorsqu'un événement de spéciation survient, les deux espèces filles héritent de la valeur du trait de leur mère. Chacune voit ensuite son trait évoluer comme un BM, indépendamment l'une de l'autre. Il est important de noter que, même si l'on suppose que les deux espèces filles sont indépendantes, le simple fait qu'elles aient hérité leur trait d'un même ancêtre commun introduit des corrélations entre leurs traits respectifs. Par exemple, supposons que, par hasard, le trait de l'espèce mère a dérivé vers des valeurs extrêmes. Si le trait considéré est la taille, l'espèce est, disons, particulièrement grande. Ses deux enfants commenceront alors leur évolution en étant de grande taille, et elles auront une grande probabilité de se ressembler pendant encore un long moment, étant plus grande que la plupart des autres espèces du groupe. Si le processus d'évolution du trait est un BM, on peut quantifier ces corrélations : la covariance entre les traits de deux espèces actuelles est proportionnelle à leur temps d'évolution partagé, c'est-à-dire au temps qui s'est écoulé entre la racine de l'arbre et leur plus récent ancêtre commun.

Tout l'art des PCM réside dans la définition correcte du modèle d'évolution dynamique du trait, et dans l'étude du type de distribution qu'il produit aux feuilles de l'arbre, pour les espèces observées.





(a) Arbre phylogénétique daté. La position verticale des espèces actuelles (feuilles) est arbitraire. (b) BM sur les branches de l'arbre. L'axe vertical donne la valeur du trait (unité arbitraire).

Réalisation d'un BM univarié sur un arbre phylogénétique daté. Les couleurs des branches (à gauche) sont associées aux couleurs de différents processus (à droite). Par exemple, le trait de l'espèce ancestrale rouge évolue d'une valeur de 0 (nœud  $R$  au temps  $-200$ ) jusqu'à une valeur de 2.6 (nœud  $S$ , au temps  $-100$ ). Seules les espèces actuelles (à  $t = 0$ ), aux feuilles de l'arbre, sont observées. Les espèces violette et bleu ( $A$  et  $B$ ) ont hérité du trait de leur ancêtre vert ( $U$ ) assez récemment, ils ont donc une grande probabilité de se ressembler encore au temps présent. Les valeurs de leur traits sont marginalement corrélées (mais indépendantes conditionnellement à  $U$ ).

### Détection de sauts

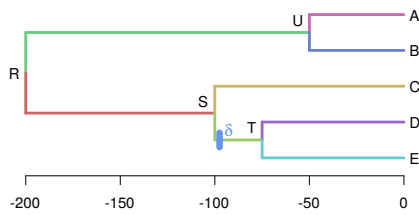
La distribution du trait aux feuilles de l'arbre obtenu par ce processus peut être vue comme un modèle nul, qui décrit les corrélations entre les espèces, ainsi que les intervalles de variation du trait, que l'on peut s'attendre à observer sous le coup du « hasard » seul. Si la distribution observée diffère de manière significative de la distribution attendue, on peut alors être amené à penser qu'un événement particulier a contribué à façonner l'histoire du trait étudié.

Dans ce manuscrit, on s'intéresse plus particulièrement aux *sauts* qui peuvent survenir à certains moments de l'histoire évolutive du trait considéré. De tels sauts sont caractérisés par un changement brutal de la valeur du trait, et peuvent avoir plusieurs causes biologiques, comme une migration vers un nouvel environnement ou un changement climatique rapide. Une espèce ancestrale affectée par un tel saut va transmettre la valeur de son trait à sa progéniture, si bien que, parmi les espèces actuelles, tous ses descendants vont hériter du changement découlant de cet événement. Un exemple d'une telle situation est présenté dans la figure qui suit.

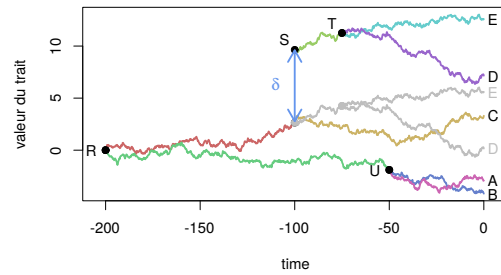
Bien entendu, plusieurs sauts peuvent avoir lieu dans l'histoire d'un groupe d'espèces. L'un des buts principaux de cette thèse est de retrouver, premièrement, le nombre et, subséquentment, la position de ces sauts sur l'arbre phylogénétique considéré. Ces deux questions posent chacune des problèmes statistiques différents, qui sont exposés dans les deux paragraphes suivants.

### Identifiabilité

L'un des aspects importants du problème à garder à l'esprit est que, bien que l'on ait défini un modèle dynamique d'évolution du trait au cours du temps, on ne peut mesurer l'état du système qu'à un instant donné, pour les espèces observées aujourd'hui. Seules les dernières valeurs prises par le processus stochastique, sur les feuilles de l'arbre, sont



(a) Un arbre phylogénétique daté.

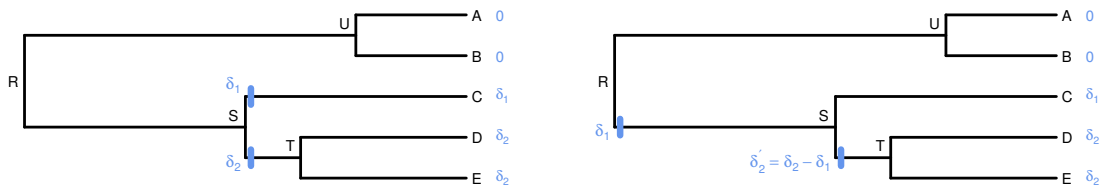


(b) BM sur les branches de l'arbre.

Même processus, mais avec un saut sur la branche vert clair (entre  $S$  et  $T$ ). Les feuilles  $D$  et  $E$  sont affectées par le saut ancestral : leur trait a une valeur bien plus grande que celle attendue en l'absence de saut (en gris).

donc accessibles. Dans une telle situation, on s'attend à voir émerger des problèmes d'*identifiabilité*.

L'ajout de sauts ne fait qu'empirer la situation. Sur la figure suivante, on montre qu'il est possible de construire facilement deux scénarios distincts, avec des sauts se produisant sur des branches différentes, qui pourtant donnent exactement la même distribution attendue du trait aux feuilles de l'arbre. Il est impossible de discriminer ces deux scénarios, donnant deux histoires biologiquement différentes de l'évolution du trait, en se basant uniquement sur les données accessibles aux feuilles de l'arbre. On dit qu'ils ne sont pas identifiables. L'étude et la quantification de ces problèmes d'identifiabilité est cruciale. D'un point de vue statistique, les ignorer nous conduirait à utiliser des modèles mathématiquement mal définis, ce qui poserait des problèmes pour leur inférence. De plus, d'un point de vue biologique, il est important de comprendre ce que les données collectées peuvent – et ne peuvent pas – nous dire sur l'histoire évolutive du trait considéré.



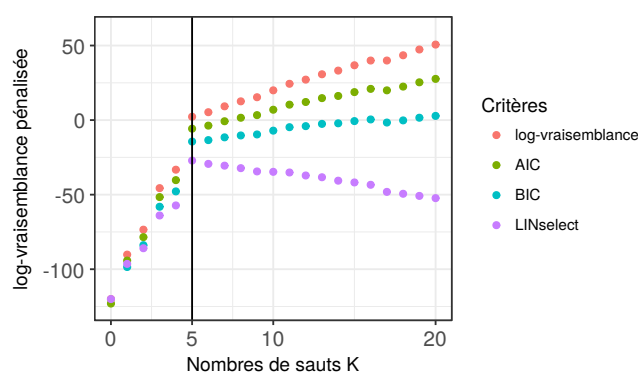
Deux scénarios équivalents. On suppose que le trait évolue suivant un BM avec des sauts marqués sur les branches de l'arbre, et partant d'une valeur ancestrale de 0. L'espérance du trait aux feuilles est indiquée en bleu. Les deux scénarios, bien que donnant la même distribution du trait, ne sont pas équivalents d'un point de vue biologique. Dans le scénario de gauche, les deux enfants de l'espèce  $S$  sont chacun affectés par un événement extrême, qui conduit à deux sauts dans la valeur de leurs traits respectifs. Dans celui de droite, c'est l'espèce ancestrale qui subit d'abord un saut, transmis en l'état à son premier enfant  $C$ , tandis que son second enfant s'ajuste avec un deuxième saut.



## Sélection de modèle

Une fois que l'on a déterminé la collection de modèles identifiables que l'on peut utiliser, il reste à trouver une méthode pour choisir le « meilleur » de ces modèles, au vu des données collectées. Cette inférence se fait en deux étapes. Premièrement, si l'on se donne un nombre arbitraire  $K$  de sauts à répartir sur l'arbre, on montre que l'on est capable de trouver le modèle à  $K$  sauts qui maximise la vraisemblance des données. Cependant, ce nombre de sauts est lui-même inconnu. La deuxième étape consiste donc à choisir un nombre de sauts approprié. Le critère de vraisemblance que l'on a utilisé à l'étape précédente ne peut pas être utilisé en l'état pour ce problème. En effet, il est facile de montrer que la vraisemblance croît avec le nombre de paramètres d'un modèle, si bien que qu'une solution  $K + 1$  sauts aura toujours une vraisemblance plus grande qu'une solution à  $K$  sauts. Utiliser ce critère reviendrait ainsi à toujours choisir, quelles que soient les données étudiées, la solution avec le plus de sauts possibles, c'est à dire celle présentant un saut pour chaque espèce de l'arbre. Une telle solution correspondrait typiquement à un cas de sur-ajustement du modèle aux données, et n'apporterait aucune information sur le processus biologique à l'œuvre.

Pour ne garder que les sauts dit *significatifs*, il est courant d'utiliser un critère de *vraisemblance pénalisée*. La pénalité doit être choisie de manière à compenser, entièrement et uniquement, l'augmentation *mécanique* de la vraisemblance avec le nombre de sauts. Le principe de cette méthode est illustré dans la figure qui suit. Cet exemple simple montre l'importance du choix de la pénalité, qui doit être adaptée au problème considérée. En particulier, on voit sur ce graphique que des critères utilisés de manière courante, comme l'AIC ou le BIC, ne sont pas à même de corriger les effets du sur-ajustement. On s'attachera ici à dériver une pénalité prenant en compte les spécificités de la collection de modèles considérés, dont la structure particulière a été décrite dans la section précédente. Grâce à la théorie de la sélection de modèle, cette pénalité hérite de bonnes garanties théoriques.



Profil de vraisemblance typique. Les données utilisées ont été générées par simulation, en utilisant un BM courant sur un arbre de 64 espèces, et présentant  $K_{\text{true}} = 5$  sauts bien marqués. Chaque point rouge montre la vraisemblance obtenue en fixant le nombre  $K$  de sauts autorisés, pour  $K$  variant de 0 à 20. Comme attendu, cette vraisemblance est strictement croissante. Le rôle d'une pénalité est de créer un critère ayant un maximum en  $K = 5$  (ligne verticale), afin que l'on puisse retrouver le vrai nombre de sauts (inconnu dans les cas d'application, mais connu ici, puisque les données sont simulés suivant un modèle fixé). Les critères standards AIC et BIC (en vert et bleu) sont tous les deux strictement croissants, et ne peuvent donc pas être utilisés ici. En revanche, le critère LINselect (en violet) que nous proposons d'utiliser a bien un maximum en  $K = 5$ , qui nous permet de retrouver la bonne solution dans cet exemple jouet.

Les modèles et méthodes statistiques présentés ici sont à la base de ces travaux de thèse. Dans ce document, on s'attache d'abord à en présenter les fondations (Chapitre 1), avant de les développer et d'en présenter des extensions à d'autres processus stochastiques (Chapitres 2 et 3) ou à d'autres structures de parentés entre les espèces (Chapitre 4).

## Chapitre 1 : Contexte

Le premier chapitre s'attache à dresser un panorama des outils convoqués dans le reste du manuscrit. Trois thèmes principaux sont abordés. Tout d'abord, on s'intéresse aux propriétés mathématiques des arbres phylogénétiques, qui servent de support aux développements ultérieurs, et on rappelle quelques résultats sur la structure de l'espace des caractères associés aux nœuds d'un tel arbre (section 1.1). Dans un deuxième temps, on décrit un certain nombre de modèles classiques d'évolution dynamique de traits, discrets ou quantitatifs, le long d'un arbre phylogénétique (sections 1.2, 1.3 et 1.4). Enfin, on évoque quelques résultats statistiques découlant de la théorie de la sélection de modèle, qui seront utiles dans la suite (section 1.5).

### Arbres phylogénétiques, caractères convexes et parcimonie

En se basant essentiellement sur les livres de [Semple & Steel \(2003\)](#) et [Felsenstein \(2004\)](#), on rappelle les définitions et propriétés suivantes. Tout d'abord, on appelle arbre phylogénétique un arbre dont les feuilles sont identifiées par un label distinctif (comme un nom d'espèce).

**Définition** (Arbre phylogénétique). Un *arbre binaire raciné*  $T = (V, E)$  est un graphe connecté acyclique, avec  $V$  un ensemble de nœuds, et  $E$  d'arêtes, tel que tous les nœuds internes sont de degrés 3, sauf un, la racine, de degré 2. Les nœuds externes, de degré 1, sont les *feuilles* de l'arbre.

Un *arbre phylogénétique* (sur  $X$ )  $\mathcal{T}$  est un couple  $(T, \phi)$ , où  $T$  est un arbre, et  $\phi : X \rightarrow L$  une bijection depuis un ensemble de labels  $X$ , vers l'ensemble  $L$  des feuilles de  $T$ .

Pour chacune des ces espèces identifiées aux feuilles de l'arbre, on mesure un trait, ou caractère donné. On se limite ici aux caractères dit convexes, qui sont le résultat d'une évolution où chaque innovation est unique (la même innovation ne peut pas apparaître deux fois indépendamment, on dit aussi qu'il n'y a pas d'homoplasie).

**Définition** (Caractère convexe). Un *caractère discret complet*  $\chi : X \rightarrow C$  est une application de l'ensemble des feuilles d'un arbre phylogénétique (identifiées par leurs labels) vers un ensemble de caractères  $C$ . Si  $|\chi(X)| = r$ ,  $\chi$  est un *caractère à  $r$  états*.

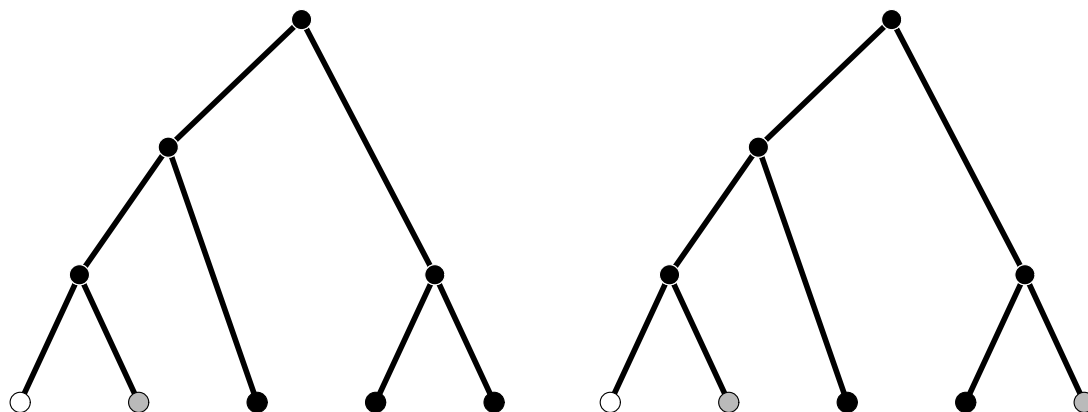
Une *extension* de  $\chi$  sur  $\mathcal{T}$  est une application  $\bar{\chi} : V \rightarrow C$  telle que  $\bar{\chi} \circ \phi = \chi$ . On note  $\text{Ex}(\chi, \mathcal{T})$  l'ensemble de toutes les extensions de  $\chi$  sur  $\mathcal{T}$ .

$\chi$  est dit *convexe* s'il existe une extension  $\bar{\chi}$  telle que, pour tout  $c$ ,  $c \in C$ , le sous-graphe de  $T$  induit par  $\{v \in V \mid \bar{\chi}(v) = c\}$  est connecté.

La proposition suivante porte sur l'énumération des caractères convexes sur une phylogénie. Elle nous sera utile lors de notre étude d'identifiabilité.

**Proposition** ([Steel \(1992\)](#), proposition 1, item 4). *Le nombre de caractères complets à  $r$  états sur un arbre phylogénétique binaire ayant  $n$  feuilles est donné par :*

$$\binom{2n - r - 1}{r - 1}.$$



(a) Caractère convexe, sans homoplasie.

(b) Caractère non-convexe, convergences.

Un arbre phylogénétique associé à un caractère convexe (gauche) ou non convexe (droite). On présente ici un caractère discret, pouvant prendre trois valeurs. Le trait de chaque nœud est représenté par sa couleur (noir, gris ou blanc).

Un caractère donné peut avoir un grand nombre d'extensions. On s'intéresse souvent à celles qui vérifient une propriété de minimalité, en ce qu'elles induisent un nombre minimal de changements. On dit qu'elles sont parcimonieuses, et le nombre de changements associé est le score de parcimonie du caractère. Il vérifie la propriété suivante.

**Proposition** (Semple & Steel, 2003, Proposition 5.1.3). *Soit  $\chi$  un caractère à  $r$  états sur un arbre phylogénétique  $\mathcal{T}$ . Alors :*

$$\min_{\bar{\chi} \in \text{Ex}(\chi, \mathcal{T})} |\{(u, v) \in E \mid \bar{\chi}(u) \neq \bar{\chi}(v)\}| \geq r - 1,$$

avec égalité si et seulement si  $\chi$  est convexe sur  $\mathcal{T}$ .

Plusieurs algorithmes classiques de programmation dynamique, comme ceux de Fitch ou Sankoff permettent d'obtenir une extension parcimonieuse d'un caractère donné en un temps  $O(|X| \times |C|)$  (voir Felsenstein, 2004, pour une introduction).

### Modèles d'évolution et modèles à variables latentes

Dans la section précédente, on s'est intéressé aux propriétés purement combinatoires des caractères attribués aux feuilles d'un arbre. Ici, on introduit un modèle d'évolution dynamique des caractères au cours du temps, nous permettant d'étudier plus précisément les mécanismes de l'évolution. Un tel modèle, stochastique, nous permet de voir les caractères mesurés comme des variables aléatoires, et ainsi de re-formuler le problème en terme de modèle à variables latentes, où les variables cachées sont les nœuds internes de l'arbre.

**Définition** (Modèle d'évolution de trait générique). Soit  $T = (E, V)$  ayant une racine  $\rho$ , tel que chacune de ses branches  $e \in E$  a une longueur  $\ell_e$ . Soit  $\mathbf{X}$  le vecteur des variables aléatoires décrivant les valeurs prises par les traits aux nœuds de l'arbre, à valeurs dans

un espace de caractères arbitraire  $C$  (indifféremment discret ou continu). La loi de  $\mathbf{X}$  est définie par :

- $\mathbf{X}_\rho \sim \mathcal{D}(\boldsymbol{\theta}_1)$  : la racine suit une loi donnée  $\mathcal{D}$ , de paramètres  $\boldsymbol{\theta}_1$ .
- Soit  $e \in E$  une branche, allant de  $\text{pa}(i)$  à  $i$ . Sur cette branche, le trait évolue comme un processus stochastique  $(\mathbf{W}_t^e, 0 \leq t \leq \ell_e)$  de loi  $\mathcal{P}(\boldsymbol{\theta}_e)$ , et ceux indépendamment des autres espèces, conditionnellement à  $\mathbf{W}_0^e = \mathbf{X}_{\text{pa}(i)}$ .
- Au nœud  $i$ , on définit  $\mathbf{X}_i = \mathbf{W}_{\ell_e}^e$ .
- On itère le long de l'arbre jusqu'à ce que tous les nœuds soient visités.

Un exemple d'un tel modèle a été présenté en introduction pour un trait continu, en prenant comme processus  $\mathcal{P}$  le BM. Lorsque l'espace d'état est discret, de nombreux modèles peuvent être décrits, qui permettent de modéliser par exemple l'évolution des bases aminées d'une séquence d'ADN. Ces modèles sont à la base des méthodes modernes pour inférer un arbre phylogénétique par maximum de vraisemblance.

Le modèle général décrit repose sur l'hypothèse fondamentale que les espèces évoluent de manière indépendantes les unes des autres après une spéciation. Cette hypothèse, bien que peu réaliste d'un point de vue biologique, est nécessaire pour le traitement mathématique simple du modèle. D'autres hypothèses doivent être faites sur le processus  $\mathcal{P}$ , en fonction de l'espace d'états  $C$  choisi. Si l'on fait l'hypothèse générique que le processus est *markovien* on peut alors montrer que la loi de  $\mathbf{X}$  peut être obtenue comme le résultat d'un modèle graphique orienté, tel que défini comme suit.

**Définition** (Modèle graphique orienté). Un vecteur  $\mathbf{X}$  de variables aléatoires sur un espace  $C$  suit un *modèle graphique orienté* s'il se situe aux nœuds d'un graphe acyclique orienté, et est tel que sa distribution jointe peut se factoriser de la manière qui suit :

$$p_{\boldsymbol{\theta}}(\mathbf{X}) = \prod_{i \in V} p_{\boldsymbol{\theta}}(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)})$$

où  $\text{pa}(i)$  est l'ensemble de tous les parents directs de  $i$  dans le graphe, et  $\boldsymbol{\theta}$  un vecteurs de paramètres de la distribution. Par convention, si  $\text{pa}(i) = \emptyset$  (par exemple, à la racine d'un arbre), on prend :  $p_{\boldsymbol{\theta}}(\mathbf{X}_i \mid \mathbf{X}_{\text{pa}(i)}) = p_{\boldsymbol{\theta}}(\mathbf{X}_i)$ .

En pratique,  $\mathbf{X}$  se divise en deux composantes,  $\mathbf{Y}$ , les traits aux feuilles de l'arbre, observées, et  $\mathbf{Z}$  les traits aux nœuds internes de l'arbre, pour des espèces ancestrales, non observées. On est alors dans le cadre d'un modèle graphique à variables latentes. C'est dans ce cadre, bien étudié, que nous nous plaçons pour notre étude statistique.

### Sélection de modèle

Le principe général de la sélection de modèle par vraisemblance pénalisée a été exposé plus haut dans cette introduction. On renvoie à Giraud (2014) pour une introduction construite à ces méthodes statistiques. Dans la suite, on se base sur la méthode LINselect, présentée dans Baraud et al. (2010). Cette méthode s'applique à des problèmes écrits sous forme de régression linéaire :

$$\mathbf{Y} = \boldsymbol{\mu} + \sigma^2 \mathbf{E}$$

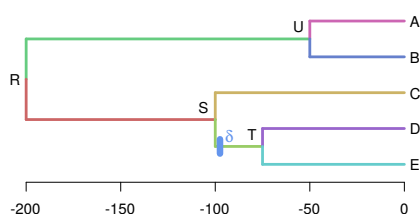
avec  $\mathbf{E}$  un vecteur de variables gaussiennes i.i.d., et  $\sigma^2$  un paramètre de variance *inconnu*. Les différents modèles sont représentés par les espace vectoriels de divers dimensions dans lequel on autorise  $\boldsymbol{\mu}$  à évoluer.

## Chapitre 2 : Détection de sauts pour des processus univariés

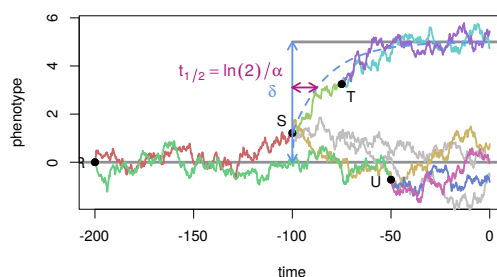
Ce chapitre a fait l'objet d'une publication dans le « Journal of the Royal Statistical Society » (Bastide, Mariadassou & Robin, 2017b). Il s'attaque au problème de la détection de sauts adaptatifs dans le cas univarié : seul un trait est mesuré pour chaque espèce. Cette étude se fait en trois étapes principales. On s'intéresse tout d'abord à la définition et aux propriétés des modèles utilisés, avant d'en étudier l'identifiabilité en détails, puis d'en proposer une procédure d'inférence statistique, que l'on valide grâce à une campagne de simulations, et dont on étudie les résultats sur un jeu de données classique.

### Modèle

Le modèle d'évolution du trait est le même que celui décrit ci-dessus, en prenant comme espace d'états  $C$  la droite réelle, et comme processus  $\mathcal{P}$  le BM, ou le processus d'Ornstein-Uhlenbeck (OU). Un trait  $(X_t)_{t \geq 0}$  suivant ce processus a pour l'équation différentielle stochastique :  $dX_t = \alpha(\beta - X_t)dt + \sigma dB_t$ , où  $(B_t)_{t \leq 0}$  est le mouvement brownien et  $\sigma^2$  un paramètre de variance. La partie déterministe de l'équation décrit un mouvement de rappel vers une valeur centrale,  $\beta$ , interprétée comme la valeur optimale du trait dans un environnement donné, avec une vitesse contrôlée par le paramètre de rappel élastique  $\alpha$ . Ce paramètre s'interprète plus facilement en considérant le temps de demi-vie phylogénétique  $t_{1/2} = \ln(2)/\alpha$  (Hansen, 1997). C'est le temps nécessaire au trait pour parcourir la moitié de la distance qui le sépare de l'optimum. On peut comparer ce temps avec la hauteur totale  $h$  de l'arbre phylogénétique sur lequel il évolue. Si  $t_{1/2}$  est grand par rapport à  $h$ , cela signifie que le trait, durant son histoire évolutive, n'a jamais le temps d'arriver à son optimum : la sélection est de faible intensité. À l'inverse, si  $t_{1/2}$  est petit par rapport à  $h$ , le trait converge rapidement vers son optimum, et la force de sélection est grande. Un exemple de ce modèle sur un arbre simple est présenté dans la figure suivante.



(a) Un arbre phylogénétique daté.



(b) OU sur les branches de l'arbre.

Processus OU avec saut courant sur les branches de l'arbre. L'évolution de la moyenne du trait après le saut est indiqué en pointillé. Le temps de demi-vie phylogénétique nécessaire au trait pour parcourir en moyenne la moitié de la distance le séparant de l'optimum est indiqué en bordeaux. Ici, la force de sélection est grande par rapport à la taille de l'arbre.

À la différence du BM, les sauts adaptatifs sur l'OU n'ont pas lieu directement sur la valeur du trait, mais sur la valeur de l'optimum  $\beta$ , comme on le voit sur la figure ci-dessus. Un tel saut peut traduire un changement de conditions environnementales, qui

induit un nouvel équilibre optimal, vers lequel le trait converge, de manière continue, et avec une vitesse contrôlée par la force de sélection  $\alpha$ . Ce modèle OU, qui admet un état stationnaire et induit une variance bornée, a été proposé pour modéliser une évolution stabilisatrice.

Comme on l'a vu dans le chapitre précédent, la définition du modèle en terme de loi conditionnelle d'une espèce fille sachant la valeur de sa mère nous permet de reformuler le problème en terme de modèle graphique. Une autre vision, complémentaire, peut être obtenue en écrivant la loi des feuilles sous forme d'une régression linéaire.

**Proposition** (Modèle linéaire). *On suppose que les nœuds internes de l'arbre sont numérotés de 1 à  $m$ , et ses feuilles de  $m+1$  à  $m+n$ . Soit  $\mathbf{T}$  la matrice d'incidence de l'arbre, de taille  $n \times (m+n)$ , telle que, pour  $1 \leq i \leq n$  et  $1 \leq j \leq m+n$ ,  $T_{ij}$  vaut 1 si la feuille  $m+i$  est descendante du nœud  $j$ , et 0 sinon. Soit  $\mathbf{Y}$  le vecteur des traits aux feuilles de l'arbre. Pour un modèle d'OU, on a :*

$$\mathbf{Y} = \mathbf{TW}(\alpha)\Delta + \frac{\sigma^2}{2\alpha}\mathbf{E}$$

où  $\Delta$  est un vecteur (parcimonieux) de taille  $m+n$ , représentant les sauts sur les branches de l'arbre : pour tout nœud  $i$ ,  $\Delta_i$  vaut 0 s'il n'y a pas de saut sur la branche menant à  $i$ , et la valeur de ce saut sinon.  $\mathbf{W}(\alpha)$  est une matrice diagonale d'expression connue, et  $\mathbf{E}$  est un vecteur gaussien, dont la matrice de variance  $\mathbf{V}(\alpha)$ . Ces deux quantités ne dépendent que de  $\alpha$ , et des paramètres (fixés) de l'arbre phylogénétique.

Cette vision du problème nous permet d'appliquer les outils de sélection de modèle évoqués précédemment, à condition de caractériser avec précision les modèles considérés, ce que l'on s'attache à faire dans la section suivante.

## Identifiabilité

On s'intéresse ici à deux problèmes complémentaires. Premièrement, étant donnée une allocation de sauts sur les branches de l'arbre, on cherche à énumérer toutes les allocations distinctes qui lui sont équivalentes, au sens où elles produisent un modèle de même vraisemblance, comme exposé dans l'introduction. Cette question répond au problèmes d'identifiabilité. Deuxièmement, étant donné un nombre  $K$  de sauts fixés, on compte le nombre de modèles réellement distincts qui possèdent  $K$  sauts. Ce nombre nous donne la complexité de cette classe de modèle, dont on a besoin dans l'étape de sélection de modèle.

Ces problèmes peuvent s'étudier en utilisant le formalisme des caractères convexes introduit dans le chapitre précédent. Les deux points fondamentaux sont, d'une part, de se limiter aux allocations *parcimonieuses* de sauts sur l'arbre, et d'autre part, de représenter une classe d'équivalence pour une allocation donnée à  $K$  sauts par la classification en  $K+1$  groupes qu'elle induit aux feuilles.

La première question (énumération d'une classe d'équivalence) peut alors être réglée par un algorithme récursif, adapté de l'algorithme de Sankoff de programmation dynamique pour trouver le coût parcimonieux d'un caractère évoqué précédemment.

La seconde question (décompte du nombre de classes d'équivalences à  $K$  sauts) revient à un décompte des caractères convexes. Pour un arbre binaire, la proposition rappelée dans l'introduction nous permet ainsi de répondre à la question. On constate en particulier que la formule close exhibée ne dépend pas de la topologie de l'arbre. Pour

un arbre non binaire (c'est-à-dire, comportant certains nœuds ayant plus de deux espèces filles, ou polytomies), on peut écrire un algorithme récursif sur l'arbre permettant trouver ce nombre, qui en général, dans ce cas, dépend de la topologie de l'arbre considéré.

### Inférence

Pour l'inférence, on utilise les deux visions complémentaires du modèle. La vision en terme de modèle graphique nous permet d'écrire un algorithme de « Expectation Maximization » (EM) pour maximiser la vraisemblance à nombre de sauts  $K$  fixés. Comme le modèle est gaussien, et que les données sont situées aux nœuds d'un arbre, un algorithme efficace, semblable à un algorithme de « upward-backward », ou à un filtre de Kalman, permet de calculer toutes les quantités nécessaires à l'étape E en seulement deux parcours de l'arbre. À l'étape M de maximisation, c'est l'optimisation en la position des sauts sur les branches de l'arbre, combinatoire, qui peut poser problème. Du fait de l'indépendance des incréments du BM, cette optimisation revient cependant à la simple minimisation d'une somme de coûts indépendants, associés à chaque branches, et dont il suffit d'annuler les  $K$  plus grands. Lorsque l'on utilise un OU, cette méthode n'est plus applicable, et l'on doit recourir à des heuristiques pour augmenter, si ce n'est maximiser, la quantités cible (Generalized EM).

La seconde vision, en terme de régression linéaire, nous est utile à deux reprises. Premièrement, elle nous permet de trouver une bonne initialisation pour l'EM, qui y est notoirement sensible. Cette initialisation se base sur une pénalité de type LASSO, qui permet d'obtenir un vecteur de sauts  $\Delta$  parcimonieux. Pour chaque valeur de  $K$ , on règle le paramètre de régularisation de telle sorte à obtenir exactement  $K$  coefficients non nuls.

Deuxièmement, comme mentionné précédemment, cette formulation en terme de régression linéaire nous permet d'adapter des outils de sélection de modèle à notre problème. Cette deuxième étape, dont le principe a été expliqué schématiquement dans l'introduction, nous permet de choisir, parmi toutes les solutions obtenues par l'EM, celle ayant un nombre de sauts adapté. La méthode choisie garantie, pour le BM ou lorsque  $\alpha$  est fixé pour l'OU, une inégalité de type oracle.

La méthode complète est implémentée dans le paquet R *PhylogeneticEM*, disponible sur le CRAN. Elle a été testée sur un grand nombre de scénarios simulés, et utilisée sur un jeu de données classique, relatifs à la famille des *chéloniens*, comportant toutes les espèces de tortues connues aujourd'hui.

## Chapitre 3 : Détection de sauts pour des processus multivariés

Dans ce chapitre, on s'attache à étendre la méthode précédente à des traits multivariés : pour chaque espèce, on mesure non pas une seule mais plusieurs caractéristiques. Ce chapitre a fait l'objet d'une soumission dans « *Systematic Biology* » ayant, au moment où ce manuscrit pars à l'impression, reçu un avis positif pour une publication sous réserve de révisions mineures (Bastide, Ané, Robin & Mariadassou, 2017a).

L'extension au BM multivarié se fait de manière assez naturelle. Lors de l'algorithme EM, l'étape de maximisation M peut se traiter de manière identique. Pour l'étape E, on peut encore écrire un algorithme efficace, capable de gérer la présence de données manquantes, ce qui est particulièrement appréciable pour le traitement des jeux de données écologiques considérés. Enfin, la méthode de sélection de modèle peut également être étendue, bien que les garanties théoriques tombent dans ce cas. Le critère peut alors être



vu comme une simple heuristique, dont la pertinence a été évaluée par un grand nombre de simulations.

Le méthode d'inférence pour le BM multivarié, efficacement implémentée dans le paquet `PhylogeneticEM`, est ainsi rapide et complète. Contrairement aux méthodes proposées jusqu'à présent, elle ne fait pas l'hypothèse que les traits sont indépendants les uns des autres. Cette hypothèse est particulièrement fautive sur beaucoup de jeux de données écologiques. Pour tenter de décorréliser les traits, une adaptation phylogénétique de l'ACP (en anglais, pPCA) était utilisée jusqu'à présent. Cependant, lorsque les traits ont subis des sauts au cours de leur histoire, on peut montrer que cette méthode est biaisée, et peut fausser l'analyse. On s'attend ainsi à ce que notre méthode, qui en fait l'économie, donne de meilleurs résultats.

L'extension à l'OU multivarié est plus problématique. En toute généralité, le paramètre de force de sélection,  $\alpha$ , se transforme en matrice  $\mathbf{A}$ , difficile à gérer. La loi produite par ce processus général, bien que toujours gaussienne, n'est ainsi pas facile à étudier, et les interactions entre les matrices  $\mathbf{A}$  et  $\Sigma$  de variance mal connues, et propres à générer des problèmes d'identifiabilité. Dans ce travail, on décide de simplifier le problème en ne considérant que des matrices  $\mathbf{A}$  scalaires, c'est-à-dire égales à  $\alpha$  fois la matrice identité. Cette hypothèse est assez restrictive, car elle suppose que tous les traits sont attirés par leurs divers optimums indépendamment, et avec une même vitesse. On ne fait en revanche pas d'hypothèse sur la matrice de variance  $\Sigma$ , si bien que les traits sont tout de même autorisés à évoluer de manière corrélée.

Lorsque toutes les données sont mesurées pour des espèces actuelles, à un seul moment dans l'histoire, on montre que l'OU est équivalent à un BM courant sur un arbre dont les branches ont été renormalisées, par une transformation dépendant de  $\alpha$ . À  $\alpha$  connu, il est alors possible de bénéficier de la méthode multivariée développée pour le BM. Il suffit alors de reproduire cette analyse sur une grille en  $\alpha$ , avant de sélectionner la meilleure solution en terme de vraisemblance. L'étape de sélection de modèle utilise ensuite une heuristique similaire.

De même que précédemment, la performance de la méthode, sous divers scénarios explorant plusieurs formes de violations aux hypothèses du modèle, a été évaluée par simulations, et utilisée pour analyser des jeux de données classiques issus de la littérature.

## Chapitre 4 : Évolution de traits sur des réseaux phylogénétiques

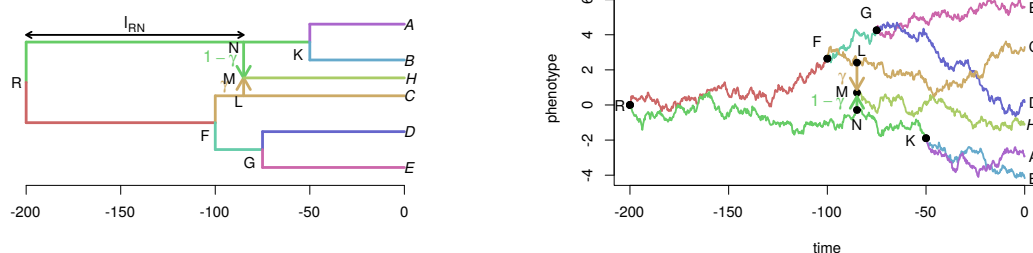
Ce chapitre s'éloigne de la thématique de la détection de ruptures pour explorer d'autres formes de relations de parentés entre les espèces, représentées non plus par un arbre, mais par un réseau phylogénétique. Il a fait l'objet d'une soumission dans un journal de biologie à comité de relecture (Bastide, Solís-Lemus, Kriebel, Sparks & Ané, 2017c). Les outils développés ici ont été intégrés au paquet `Julia PhyloNetworks`, dont la présentation a été publiée dans « *Molecular Biology and Evolution* » (Solís-Lemus, Bastide & Ané, 2017). La présentation que l'on en fait ici est adaptée du résumé long de l'exposé présenté aux 49<sup>èmes</sup> Journées de Statistiques de la SFdS (Avignon, 2017).

### Réseau phylogénétique

Les liens de parentés entre espèces sont représentés de manière classique par un arbre phylogénétique. Cependant, cette représentation arborescente ne tient pas compte des événements d'hybridations, ou de transferts de gènes horizontaux, qui peuvent modifier



substantiellement les relations de filiation entre les espèces présentes. On a alors recours à un *réseau phylogénétique* pour représenter ces liens. Un réseau phylogénétique est un graphe acyclique dirigé et raciné, dont les feuilles représentent les espèces actuelles observées, et les nœuds internes des espèces ancestrales. Les nœuds internes peuvent avoir un seul parent (filiation arborescente) ou bien deux parents (hybridation). Le réseau est calibré en temps, si bien que la longueur des branches arborescentes représente un temps évolutif. L'événement d'hybridation étant instantané, les branches menant à un nœud hybride sont supposées de longueur nulle. Un paramètre  $\gamma$  leur est cependant associé, représentant une proportion de patrimoine génétique transmis par chacun des deux parents. Un exemple de réseau phylogénétique présentant un seul événement d'hybridation est présenté ci-dessous. Sur cet exemple, le nœud hybride  $M$  a hérité d'une proportion  $\gamma$  de ses gènes de l'espèce  $L$ , et le reste  $1 - \gamma$  de l'espèce  $N$ . Plusieurs méthodes d'inférence ont été développées ces dernières années (voir par exemple [Yu et al. \(2014\)](#), [Solís-Lemus & Ané \(2016\)](#)), et ce type de réseaux phylogénétiques commence à être disponible pour un certain nombre de groupes d'espèces. Dans toute la suite, on suppose que le réseau est connu et fixé.



(a) Réseau phylogénétique. Le nœud  $L$  est hybride. (b) Variation du trait en fonction du temps. Seule  $\ell_{RN}$  est la longueur de la branche allant de  $R$  à  $N$ . la valeur du processus aux feuilles est observée.

FIGURE 6.0.1 – Réseau phylogénétique d'un ensemble d'espèces contemporaines, et modélisation de l'évolution d'un caractère par un mouvement Brownien.

### Évolution d'un trait

Pour modéliser l'évolution d'un trait quantitatif, on utilise de même que précédemment un mouvement Brownien (BM) courant sur les branches du réseau phylogénétique liant les espèces entre elles, comme présenté sur la figure ci-dessus. Le processus est défini de la manière suivante :

- Sur une branche donnée, le trait évolue au cours du temps suivant un mouvement Brownien.
- Lors d'une spéciation (nœud arborescent), le processus se divise en deux Browniens indépendants, partants du même point et avec les mêmes paramètres, courant chacun sur une des deux branches filles.
- Lors d'une hybridation, le trait hybride est obtenu en faisant la moyenne pondérée par le coefficient  $\gamma$  des traits de ses deux parents, puis évolue suivant un Brownien indépendant, et avec les mêmes paramètres.

Comparé au modèle d'évolution sur un arbre, le point nouveau est la règle de fusion adoptée aux points d'hybridations. Ici, la fusion par moyenne pondérée a été choisie, car elle conduit à des calculs simples, et est en accord avec la modélisation d'un trait multi-loci par le BM.

Grâce à ce modèle, il est possible d'étendre les Méthodes Comparatives Phylogénétiques à des espèces liés par un réseau, plutôt que par un arbre phylogénétique. Le point central dans cette adaptation est le calcul de la matrice de variance induite par ce nouveau modèle. La covariance entre les traits  $Y_i$  et  $Y_j$  de deux espèces  $i$  et  $j$  aux feuilles du réseau phylogénétique s'écrit (Pickrell & Pritchard (2012)) :

$$\text{Cov}[Y_i; Y_j] = \sigma^2 V_{ij} = \sigma^2 \sum_{\substack{p_i \in \mathcal{P}_i \\ p_j \in \mathcal{P}_j}} \left( \prod_{e \in p_i} \gamma_e \right) \left( \prod_{e \in p_j} \gamma_e \right) \sum_{e \in p_i \cap p_j} \ell_e$$

où  $\sigma^2$  est la variance du mouvement Brownien,  $\mathcal{P}_i$  est l'ensemble des chemins allant de la racine au nœud  $i$ , et, pour une arrête  $e$ ,  $\gamma_e$  est le coefficient de transmission génétique ( $\gamma_e = 1$  pour toutes les arrêtes arborescentes), et  $\ell_e$  est la longueur de l'arrête, en temps phylogénétique.

Cette formule close, impliquant une somme sur un nombre potentiellement grand de chemins, ne peut pas être utilisée directement pour calculer efficacement la matrice de covariance. On montre qu'il est possible de trier les nœuds du réseau de telle sorte à ce que la matrice  $\mathbf{V}$  puisse être calculée récursivement en un parcours du réseau, depuis la racine jusques aux feuilles.

### Évolution transgressive

L'hétérosis, ou vigueur hybride, est un phénomène bien connu en génétique, qui rend possible la naissance d'un hybride ayant un caractère exceptionnellement grand (ou petit) par rapport à ses deux parents. Dans notre modèle, le trait hybride est alors obtenu comme la moyenne pondérée des traits espèces parentes, comme précédemment, plus un saut d'une valeur  $b$ . De la même manière que pour le BM sur un arbre, on montre qu'il est alors possible de ré-écrire le problème sous la forme d'un modèle linéaire à effets fixes, et ainsi de replacer la question dans un cadre statistique bien connu. Ce modèle est semblable à celui exposé précédemment, à la différence que la position potentielle des sauts est connue d'avance : ceux-ci ne peuvent avoir lieu que sur les branches suivant un événement d'hybridation. Ceci nous permet d'écrire un test de Fisher pour tester la nullité de ces sauts d'hétérosis, dont on sait exprimer la puissance théorique.

## Chapitre 5 : Extensions et Perspectives

Dans ce dernier chapitre, on explore quelques pistes d'extensions, en essayant d'ébaucher des solutions simples lorsque cela est possible, et en identifiant les difficultés principales lorsqu'elles se présentent.

Dans tous nos développements méthodologiques, nous avons supposé disposer de mesures sans erreurs sur un arbre connu parfaitement. La non prise en compte de ces sources d'incertitudes peut avoir des conséquences néfastes sur le résultats de nos analyses, que l'on tente de quantifier à l'aide d'une série de simulations. Si la mauvaise connaissance de l'arbre est difficile à prendre en compte, les erreurs ou incertitudes de mesure des traits aux feuilles de l'arbre sont en revanche mieux étudiées dans la littérature. On montre ici

comment elles pourraient être incorporées dans notre cadre de travail. Il est intéressant de noter que ces adaptations, vues sous un autre angles, peuvent également être adoptées pour réaliser une analyse à facteurs d'un jeu de données multivarié.

Lorsque l'on s'intéresse à la détection de sauts, deux phénomènes, jusqu'à présent négligés, peuvent être intéressants à étudier : la convergence évolutive, et la parcimonie dans le nombre de traits impactés par chaque saut. Une solution simple pour incorporer ces contraintes au modèle serait d'ajouter des pénalités structurelles adéquates. Deux espèces distinctes sont dites *convergentes* si elles atteignent un même régime évolutif de manière indépendante. un critères de type « fused-ANOVA », qui pénaliserait la présence de deux régimes distincts mais proches, et favoriserait leur fusion en un seul, pourrait être adaptée pour étudier ce phénomène. D'autre part, dans le cas multivarié, on a supposé que tous les traits étaient indifféremment impactés par chaque saut. Cela revient à faire l'hypothèse que tous les traits inclus dans l'analyse sont pertinents pour l'étude des régimes adaptatifs, et peut rendre la méthode sensible à l'ajout de traits « neutres » brouillant le signal. Une pénalité LASSO de type « sparse group sparse », qui privilégierait les solutions dans lesquels un petit nombre seulement de traits seraient impliqués dans chaque saut, pourrait rendre la méthode plus robuste à ce type de bruit, et rendre l'analyse moins dépendante du choix arbitraire des traits inclus ou non.

Jusqu'à présent, nous avons également fait l'hypothèse forte que toutes les mesures provenaient d'espèces observées à un seul instant dans l'histoire, au temps présent. Cela revient à supposer que tous les arbres considérés étaient *ultramétriques*, en ignorant toutes les données fossiles éventuelles. Ces sources de données, bien que rares, sont précieuses, et peuvent nous donner des renseignements sur la nature de la dynamique de l'évolution des traits au cours du temps. Par exemple, en terme d'identifiabilité, deux solutions que l'on considérerait équivalentes sur un arbre ultramétrique peuvent devenir distinguable lorsque l'on inclue des données fossiles. Ce pouvoir discriminant est cependant très dépendant de la position du fossile sur l'arbre. Une étude systématique de l'identifiabilité des configurations de sauts sur l'arbre, telle que celle menée au chapitre 2, serait donc plus complexe, et les résultats dépendants de la topologie des arbres considérés. D'un point de vue pratique, la rupture de l'hypothèse d'ultramétrie rend l'astuce de changement d'échelle utilisée au chapitre 3, pour étendre les résultats du BM à l'OU de manière efficace, caduque. On propose ici plusieurs heuristiques pour pallier ce problème.

Enfin, dans une dernière partie, on tente de jeter les bases d'une méthode permettant d'intégrer les données manquantes de manière plus satisfaisante. En effet, on a considéré jusqu'ici que les mesures étaient manquantes de manière complètement aléatoire, pour toutes les espèces et pour tous les traits. Or, l'échantillonnage en lui-même peut avoir une certaine structure. Par exemple, des phénomènes de censure peuvent être observés, induisant des données manquantes pour toute une série de traits, par exemple très petits, donc difficiles à mesurer. La prise en compte de ces biais d'échantillonnage peuvent ainsi nous renseigner sur la nature des traits manquants, et ainsi améliorer notre analyse. On étudie ainsi pour finir quelques unes des conséquences de l'ajout de ce modèle d'échantillonnage sur notre méthode.

## Chapitre 6 : Résumé substantiel

This last short chapter, written in French, describes the context and main results presented in this manuscript. It can be read independently from the rest, and is highly redundant with the introduction, that precisely ended this way.



# Bibliography

- Akaike H. 1969. Fitting autoregressive models for prediction. *Annals of the Institute of Statistical Mathematics*. 21:243–247.
- Albert JS, Johnson DM, Knouft JH. 2009. Fossils provide better estimates of ancestral body size than do extant taxa in fishes. *Acta Zoologica*. 90:357–384.
- Alexeev N, Alekseyev MA. 2016. Combinatorial Scoring of Phylogenetic Networks. In: Computing and Combinatorics. COCOON, pp. 560–572.
- Aristide L, dos Reis SF, Machado AC, Lima I, Lopes RT, Perez SI. 2016. Brain shape convergence in the adaptive radiation of New World monkeys. *Proceedings of the National Academy of Sciences*. 113:2158–2163.
- Aristide L, Rosenberger AL, Tejedor MF, Perez SI. 2015. Modeling lineage and phenotypic diversification in the New World monkey (Platyrrhini, Primates) radiation. *Molecular Phylogenetics and Evolution*. 82:375–385.
- Arlot S, Massart P. 2009. Data-driven calibration of penalties for least-squares regression. *The Journal of Machine Learning Research*. 10:245–279.
- Baraud Y, Bouvier A, Giraud C, Huet S. 2013. *LINselect: {R} Package for Estimator Selection*. .
- Baraud Y, Giraud C, Huet S. 2009. Gaussian model selection with an unknown variance. *Annals of Statistics*. 37:630–672.
- Baraud Y, Giraud C, Huet S. 2010. Estimator selection in the Gaussian setting. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*. 50:1092–1119.
- Bartoszek K, Glémin S, Kaj I, Lascoux M. 2016. The Ornstein-Uhlenbeck process with migration: evolution with interactions. *arXiv e-print*. .
- Bartoszek K, Pienaar J, Mostad P, Andersson S, Hansen TF. 2012. A phylogenetic comparative method for studying multivariate adaptation. *Journal of Theoretical Biology*. 314:204–215.
- Bartoszek K, Sagitov S. 2015. Phylogenetic confidence intervals for the optimal trait value. *Journal of Applied Probability*. 52:1115–1132.
- Bastide P, Ané C, Robin S, Mariadassou M. 2017a. Inference of Adaptive Shifts for Multivariate Correlated Traits. *bioRxiv*. .

- Bastide P, Mariadassou M, Robin S. 2017b. Detection of adaptive shifts on phylogenies by using shifted stochastic processes on a tree. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 79:1067–1093.
- Bastide P, Solís-Lemus C, Kriebel R, Sparks KW, Ané C. 2017c. Phylogenetic Comparative Methods on Phylogenetic Networks with Reticulations. *Submitted*. .
- Bates D. 2016. Generalized linear models in Julia. [\url{https://github.com/JuliaStats/GLM.jl}](https://github.com/JuliaStats/GLM.jl).
- Baudry JP, Maugis C, Michel B. 2012. Slope heuristics: overview and implementation. *Statistics and Computing*. 22:455–470.
- Beaulieu JM, Jhwueng DC, Boettiger C, O’Meara BC. 2012. Modeling stabilizing selection: Expanding the Ornstein-Uhlenbeck model of adaptive evolution. *Evolution*. 66:2369–2383.
- Bertele U, Brioschi F. 1972. Nonserial Dynamic Programming. Orlando, FL, USA: Academic Press, Inc.
- Bezanson J, Edelman A, Karpinski S, Shah VB. 2017. Julia: A Fresh Approach to Numerical Computing. *SIAM Review*. 59:65–98.
- Birgé L, Massart P. 2001. Gaussian model selection. *Journal of the European Mathematical Society*. 3:203–268.
- Birgé L, Massart P. 2007. Minimal Penalties for Gaussian Model Selection. *Probability Theory and Related Fields*. 138:33–73.
- Blöchl E, Rachel R, Burggraf S, Hafenbradl D, Jannasch HW, Stetter KO. 1997. *Pyrolobus fumarii*, gen. and sp. nov., represents a novel group of archaea, extending the upper temperature limit for life to 113°C. *Extremophiles*. 1:14–21.
- Blomberg SP, Garland T, Ives AR. 2003. Testing for Phylogenetic Signal in Comparative Data: Behavioral Traits Are More Labile. *Evolution*. 57:717–745.
- Boussau B, Gouy M, Gascuel O. 2006. Efficient Likelihood Computations with Nonreversible Models of Evolution. *Systematic Biology*. 55:756–768.
- Brault V, Baudry JP, Maugis C, Michel B. 2012. capushe: Capushe, Data-Driven Slope Estimation and Dimension Jump. R package version 1.0.
- Butler MA, King AA. 2004. Phylogenetic Comparative Analysis: A Modeling Approach for Adaptive Evolution. *The American Naturalist*. 164:683–695.
- Cavalli-Sforza LL, Edwards AWF. 1967. Phylogenetic Analysis: Models and Estimation Procedures. *Evolution*. 21:550.
- Chen ZJ. 2013. Genomic and epigenetic insights into the molecular bases of heterosis. *Nature Reviews Genetics*. 14:471–482.
- Chiquet J, Gutierrez P, Rigai G. 2017. Fast Tree Inference With Weighted Fusion Penalties. *Journal of Computational and Graphical Statistics*. 26:205–216.

- Clavel J, Escarguel G, Merceron G. 2015. mvmorph : an r package for fitting multivariate evolutionary models to morphometric data. *Methods in Ecology and Evolution*. 6:1311–1319.
- Cook LM, Grant BS, Saccheri IJ, Mallet J. 2012. Selective bird predation on the peppered moth: the last experiment of Michael Majerus. *Biology Letters*. 8:609–612.
- Cooper N, Thomas GH, Venditti C, Meade A, Freckleton RP. 2016. A cautionary note on the use of Ornstein Uhlenbeck models in macroevolutionary studies. *Biological Journal of the Linnean Society*. 118:64–77.
- Cressler CE, Butler MA, King AA. 2015. Detecting adaptive evolution in phylogenetic comparative analysis using the ornstein-uhlenbeck model. *Systematic Biology*. 64:953–968.
- Cybis GB, Sinsheimer JS, Bedford T, Mather AE, Lemey P, Suchard MA. 2015. Assessing phenotypic correlation through the multivariate phylogenetic latent liability model. *The Annals of Applied Statistics*. 9:969–991.
- Darwin C. 1859. *On the Origin of Species*. London: John Murray.
- Davis CC, Latvis M, Nickrent DL, Wurdack KJ, Baum DA. 2007. Floral Gigantism in Rafflesiaceae. *Science*. 315:1812.
- Degnan JH, Salter La. 2005. Gene tree distributions under the coalescent process. *Evolution; international journal of organic evolution*. 59:24–37.
- Dempster A, Laird N, Rubin DB. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society Series B Methodological*. 39:1–38.
- Donoho DL, Johnstone JM. 1994. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*. 81:425–455.
- Drummond AJ, Suchard MA, Xie D, Rambaut A. 2012. Bayesian Phylogenetics with BEAUti and the BEAST 1.7. *Molecular Biology and Evolution*. 29:1969–1973.
- Drury J, Clavel J, Manceau M, Morlon H. 2016. Estimating the Effect of Competition on Trait Evolution Using Maximum Likelihood Inference. *Systematic Biology*. 65:700–710.
- Duchen P, Leuenberger C, Szilágyi SM, Harmon LJ, Eastman JM, Schweizer M, Wegmann D. 2017. Inference of Evolutionary Jumps in Large Phylogenies using Lévy Processes. *Systematic Biology*. 00:1–14.
- Eastman JM, Alfaro ME, Joyce P, Hipp AL, Harmon LJ. 2011. A Novel comparative method for identifying shifts in the rate of character evolution on trees. *Evolution*. 65:3578–3589.
- Eastman JM, Wegmann D, Leuenberger C, Harmon LJ. 2013. Simpsonian 'Evolution by Jumps' in an Adaptive Radiation of Anolis Lizards. *arXiv e-print*. .
- Eckley IA, Fearnhead P, Killick R. 2011. Analysis of changepoint models. In: Barber D, Cengil AT, Chiappa S, editors, *Bayesian Time Series Models*, Cambridge: Cambridge University Press, January, pp. 205–224.

- Eldredge N, Gould SJ. 1972. Punctuated equilibria: an alternative to phyletic gradualism.
- Faria NR, Suchard MA, Rambaut A, Lemey P. 2011. Toward a quantitative understanding of viral phylogeography. *Current Opinion in Virology*. 1:423–429.
- Felsenstein J. 1973a. Maximum Likelihood and Minimum-Steps Methods for Estimating Evolutionary Trees from Data on Discrete Characters. *Systematic Biology*. 22:240–249.
- Felsenstein J. 1973b. Maximum-likelihood estimation of evolutionary trees from continuous characters. *American Journal of Human Genetics*. 25:471–492.
- Felsenstein J. 1981. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*. 17:368–376.
- Felsenstein J. 1985. Phylogenies and the Comparative Method. *The American Naturalist*. 125:1–15.
- Felsenstein J. 2004. Inferring Phylogenies.
- Felsenstein J. 2005. Using the quantitative genetic threshold model for inferences between and within species. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 360:1427–1434.
- Felsenstein J. 2008. Comparative Methods with Sampling Error and Within-Species Variation: Contrasts Revisited and Revised. *The American Naturalist*. 171:713–725.
- Felsenstein J. 2012. A Comparative Method for Both Discrete and Continuous Characters Using the Threshold Model. *The American Naturalist*. 179:145–156.
- Felsenstein J, Churchill GA. 1996. A hidden Markov Model approach to variation among sites in rate of evolution. *Molecular Biology and Evolution*. 13:93–104.
- Fiévet JB, Dillmann C, de Vienne D. 2010. Systemic properties of metabolic networks lead to an epistasis-based model for heterosis. *Theoretical and Applied Genetics*. 120:463–473.
- Finarelli JA, Flynn JJ, Oakley T. 2006. Ancestral State Reconstruction of Body Size in the Caniformia (Carnivora, Mammalia): The Effects of Incorporating Data from the Fossil Record. *Systematic Biology*. 55:301–313.
- Fitch WM. 1971. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology*. 20:406–416.
- Fitch WM, Markowitz E. 1970. An improved method for determining codon variability in a gene and its application to the rate of fixation of mutations in evolution. *Biochemical Genetics*. 4:579–593.
- Fitzjohn RG. 2010. Quantitative traits and diversification. *Systematic Biology*. 59:619–633.
- Fitzjohn RG. 2012. Diversitree: Comparative phylogenetic analyses of diversification in R. *Methods in Ecology and Evolution*. 3:1084–1092.



- Fitzjohn RG, Maddison WP, Otto SP. 2009. Estimating trait-dependent speciation and extinction rates from incompletely resolved phylogenies. *Systematic Biology*. 58:595–611.
- Fraley C, Raftery AE, Murphy TB, Scrucca L. 2012. mclust Version 4 for R: Normal Mixture Modeling for Model-Based Clustering, Classification, and Density Estimation. Technical report, Department of Statistics, University of Washington.
- Freckleton RP. 2012. Fast likelihood calculations for comparative analyses. *Methods in Ecology and Evolution*. 3:940–947.
- Fryzlewicz P. 2014. Wild binary segmentation for multiple change-point detection. *The Annals of Statistics*. 42:2243–2281.
- Galtier N. 2001. Maximum-Likelihood Phylogenetic Analysis Under a Covarion-like Model. *Molecular Biology and Evolution*. 18:866–873.
- Galtier N, Gouy M. 1998. Inferring pattern and process: maximum-likelihood implementation of a nonhomogeneous model of DNA sequence evolution for phylogenetic analysis. *Molecular Biology and Evolution*. 15:871–879.
- Gill MS, Tung Ho LS, Baele G, Lemey P, Suchard MA. 2016. A Relaxed Directional Random Walk Model for Phylogenetic Trait Evolution. *Systematic biology*. p. syw093.
- Giraud C. 2014. Introduction to high-dimensional statistics. Chapman & Hall/CRC Monographs on Statistics & Applied Probability.
- Goldberg EE, Lancaster LT, Ree RH. 2011. Phylogenetic Inference of Reciprocal Effects between Geographic Range Evolution and Diversification. *Systematic Biology*. 60:451–465.
- Goolsby EW, Bruggeman J, Ané C. 2017. Rphylopars : fast multivariate phylogenetic comparative methods for missing data and within-species variation. *Methods in Ecology and Evolution*. 8:22–27.
- Grafen A. 1989. The Phylogenetic Regression. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 326:119–157.
- Grafen A. 1992. The uniqueness of the phylogenetic regression. *Journal of Theoretical Biology*. 156:405–423.
- Grandvalet Y, Chiquet J, Ambroise C. 2012. Sparsity by Worst-Case Quadratic Penalties. *arXiv e-print*. .
- Hadfield JD, Nakagawa S. 2010. General quantitative genetic methods for comparative biology: phylogenies, taxonomies and multi-trait models for continuous and categorical characters. *Journal of Evolutionary Biology*. 23:494–508.
- Hansen TF. 1997. Stabilizing Selection and the Comparative Analysis of Adaptation. *Evolution*. 51:1341.
- Hansen TF, Bartoszek K. 2012. Interpreting the Evolutionary Regression: The Interplay Between Observational and Biological Errors in Phylogenetic Comparative Studies. *Systematic Biology*. 61:413–425.

- Hansen TF, Houle D. 2004. Evolvability, Stabilizing Selection, and the problem of stasis. In: Phenotypic integration: studying the ecology and evolution of complex phenotypes, pp. 130–154.
- Hansen TF, Martins EP. 1996. Translating Between Microevolutionary Process and Macroevolutionary Patterns: The Correlation Structure of Interspecific Data. *Evolution*. 50:1404.
- Hansen TF, Orzack SH. 2005. Assessing Current Adaptation and Phylogenetic Inertia as Explanations of Trait Evolution: The Need for Controlled Comparisons. *Evolution*. 59:2063–2072.
- Hansen TF, Pienaar J, Orzack SH. 2008. A Comparative Method for Studying Adaptation to a Randomly Evolving Environment. *Evolution*. 62:1965–1977.
- Harmon LJ, Losos JB, Jonathan Davies T, et al. (19 co-authors). 2010. Early Burst of Body Size and Shape Evolution are Rare in Comparative Data. *Evolution*. 64:no–no.
- Hasegawa M, Kishino H, Yano Ta. 1985. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*. 22:160–174.
- Hatherly M, Piibeleht M. 2017. A documentation generator for Julia. [\url{https://github.com/JuliaDocs/Documenter.jl}](https://github.com/JuliaDocs/Documenter.jl).
- Hiscott G, Fox C, Parry M, Bryant D. 2016. Efficient Recycled Algorithms for Quantitative Trait Models on Phylogenies. *Genome Biology and Evolution*. 8:1338–1350.
- Ho LST, Ané C. 2013a. A Linear-Time Algorithm for Gaussian and Non-Gaussian Trait Evolution Models. *Systematic Biology*. 63:397–408.
- Ho LST, Ané C. 2013b. Asymptotic theory with hierarchical autocorrelation: Ornstein–Uhlenbeck tree models. *The Annals of Statistics*. 41:957–981.
- Ho LST, Ané C. 2014. Intrinsic inference difficulties for trait evolution with Ornstein–Uhlenbeck models. *Methods in Ecology and Evolution*. 5:1133–1146.
- Hubert L, Arabie P. 1985. Comparing partitions. *Journal of Classification*. 2:193–218.
- Huelsenbeck JP, Alfaro ME, Suchard MA. 2011. Biologically inspired phylogenetic models strongly outperform the no common mechanism model. *Systematic Biology*. 60:225–232.
- Huelsenbeck JP, Larget B, Swofford D. 2000. A compound poisson process for relaxing the molecular clock. *Genetics*. 154:1879–92.
- Huelsenbeck JP, Rannala B. 2003. Detecting Correlation between Characters in a Comparative Analysis with Uncertain Phylogeny. *Evolution*. 57:1237–1247.
- Hunt G, Bell MA, Travis MP. 2008. Evolution toward a New Adaptive Optimum: Phenotypic Evolution in a Fossil Stickleback Lineage. *Evolution*. 62:700–710.
- Hunt G, Rabosky DL. 2014. Phenotypic Evolution in Fossil Species: Pattern and Process. *Annual Review of Earth and Planetary Sciences*. 42:421–441.

- Huysmans JK. 1922. *À rebours*. Paris: Georges Crès.
- Huysmans JK. 1930. *Against the Grain* (Translation J. Howard). A. & C. Boni.
- Ingram T, Mahler DL. 2013. SURFACE: Detecting convergent evolution from comparative data by fitting Ornstein-Uhlenbeck models with stepwise Akaike Information Criterion. *Methods in Ecology and Evolution*. 4:416–425.
- Ives AR, Midford PE, Garland T, Oakley T. 2007. Within-Species Variation and Measurement Error in Phylogenetic Comparative Methods. *Systematic Biology*. 56:252–270.
- Jaakkola TS. 2001. Tutorial on Variational Approximation Methods. In: *Advanced mean field methods: theory and practice*, MIT press, pp. 129–159.
- Jaffe AL, Slater GJ, Alfaro ME. 2011. The evolution of island gigantism and body size variation in tortoises and turtles. *Biology Letters*. 7:558–561.
- Jetz W, Thomas G, Joy J, Hartmann K, Mooers A. 2012. The global diversity of birds in space and time. *Nature*. 491:444–448.
- Jukes TH, Cantor CR. 1969. Evolution of protein molecules. In: *Mammalian protein metabolism*, volume III, pp. 21–132.
- JuliaStats. 2016. Julia Statistics. [\url{https://github.com/JuliaStats}](https://github.com/JuliaStats).
- Kahn AB. 1962. Topological sorting of large networks. *Communications of the ACM*. 5:558–562.
- Khabbazian M, Kriebel R, Rohe K, Ané C. 2016. Fast and accurate detection of evolutionary shifts in Ornstein-Uhlenbeck models. *Methods in Ecology and Evolution*. 7:811–824.
- Kimura M. 1980. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*. 16:111–120.
- Kingman J. 1982. The coalescent. *Stochastic Processes and their Applications*. 13:235–248.
- Kishino H, Hasegawa M. 1989. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea. *Journal of Molecular Evolution*. 29:170–179.
- Kollo T, Neudecker H. 1993. Asymptotics of Eigenvalues and Unit-Length Eigenvectors of Sample Variance and Correlation Matrices. *Journal of Multivariate Analysis*. 47:283–300.
- Kubatko LS. 2009. Identifying hybridization events in the presence of coalescence via model selection. *Systematic Biology*. 58:478–488.
- Lanave C, Preparata G, Saccone C, Serio G. 1984. A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution*. 20:86–93.

- Lande R. 1976. Natural Selection and Random Genetic Drift in Phenotypic Evolution. *Evolution*. 30:314.
- Landis MJ, Schraiber JG, Liang M. 2013. Phylogenetic analysis using Lévy processes: Finding jumps in the evolution of continuous traits. *Systematic Biology*. 62:193–204.
- Lartillot N. 2014. A phylogenetic Kalman filter for ancestral trait reconstruction using molecular data. *Bioinformatics*. 30:488–496.
- Lebarbier E. 2005. Detecting multiple change-points in the mean of Gaussian process by model selection. *Signal Processing*. 85:717–736.
- Lebarbier E, Mary-Huard T. 2006. Une introduction au critère BIC : fondements théoriques et interprétation. *Journal de la Société française de Statistiques*. 147:39–57.
- Lehman EL. 1986. Testing Statistical Hypotheses. Springer Texts in Statistics. New York, NY: Springer New York.
- Lemey P, Rambaut A, Welch JJ, Suchard MA. 2010. Phylogeography takes a relaxed random walk in continuous space and time. *Molecular Biology and Evolution*. 27:1877–1885.
- Li H, Homer N. 2010. A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*. 11:473–483.
- Liow LH, Reitan T, Harnik PG. 2015. Ecological interactions on macroevolutionary time scales: clams and brachiopods are more than ships that pass in the night. *Ecology Letters*. 18:1030–1039.
- Little RJA, Rubin DB. 2002. Statistical Analysis with Missing Data. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- Losos JB. 1990. Concordant evolution of locomotor behaviour, display rate and morphology in Anolis lizards.
- Lynch M. 1991. Methods for the Analysis of Comparative Data in Evolutionary Biology. *Evolution*. 45:1065–1080.
- Maddison WP. 1997. Gene Trees in Species Trees. *Systematic Biology*. 46:523.
- Maddison WP, Midford PE, Otto SP. 2007. Estimating a binary character's effect on speciation and extinction. *Systematic biology*. 56:701–710.
- Mahler DL, Ingram T, Revell LJ, Losos JB. 2013. Exceptional Convergence on the Macroevolutionary Landscape in Island Lizard Radiations. *Science*. 341:292–295.
- Mahler DL, Revell LJ, Glor RE, Losos JB. 2010. Ecological opportunity and the rate of morphological evolution in the diversification of greater Antillean anoles. *Evolution*. 64:2731–2745.
- Mallet J. 2005. Hybridization as an invasion of the genome. *Trends in Ecology & Evolution*. 20:229–237.
- Mallet J. 2007. Hybrid speciation. *Nature*. 446:279–283.

- Manceau M, Lambert A, Morlon H. 2016. A unifying comparative phylogenetic framework including traits coevolving across interacting lineages. *Systematic Biology*. p. syw115.
- Mardia KV, Kent JT, Bibby JM. 1979. Multivariate analysis. Probability and mathematical statistics. Academic Press.
- Massart P. 2007. *Concentration Inequalities and Model Selection*. 1896.
- Meredith RW, Janecka JE, Gatesy J, et al. (22 co-authors). 2011. Impacts of the Cretaceous Terrestrial Revolution and KPg Extinction on Mammal Diversification. *Science*. 334:521–524.
- Meucci A. 2009. Review of Statistical Arbitrage, Cointegration, and Multivariate Ornstein-Uhlenbeck. *SSRN Electronic Journal*. p. 20.
- Mond B, Pečarić J. 2000. On Inequalities Involving The Hadamard Product of Matrices. *The Electronic Journal of Linear Algebra*. 6:56–61.
- Moore BR, Höhna S, May MR, Rannala B, Huelsenbeck JP. 2016. Critically evaluating the theory and performance of Bayesian analysis of macroevolutionary mixtures. *Proceedings of the National Academy of Sciences*. 113:9569–9574.
- Mossel E, Steel M. 2004. A phase transition for a random cluster model on phylogenetic trees. *Mathematical Biosciences*. 187:189–203.
- O'Meara BC. 2012. Evolutionary Inferences from Phylogenies: A Review of Methods. *Annual Review of Ecology, Evolution, and Systematics*. 43:267–285.
- Pagel M. 1999. Inferring the historical patterns of biological evolution. *Nature*. 401:877–884.
- Paley W. 1802. *Natural Theology or Evidences of the Existence and Attributes of the Deity*. London: R. Faulder.
- Paradis E, Claude J, Strimmer K. 2004. APE: Analyses of Phylogenetics and Evolution in R language. *Bioinformatics*. 20:289–290.
- Pastell M. 2017. Weave.jl: Scientific Reports Using Julia. *The Journal of Open Source Software*. 2.
- Pennell MW, Eastman JM, Slater GJ, Brown JW, Uyeda JC, FitzJohn RG, Alfaro ME, Harmon LJ. 2014. geiger v2.0: an expanded suite of methods for fitting macroevolutionary models to phylogenetic trees. *Bioinformatics*. 30:2216–2218.
- Pennell MW, FitzJohn RG, Cornwell WK, Harmon LJ. 2015. Model Adequacy and the Macroevolution of Angiosperm Functional Traits. *The American Naturalist*. 186:E33–E50.
- Pennell MW, Harmon LJ. 2013. An integrative view of phylogenetic comparative methods: connections to population genetics, community ecology, and paleobiology. *Annals of the New York Academy of Sciences*. 1289:90–105.

- Peyrard N, de Givry S, Franc A, Robin S, Sabbadin R, Schiex T, Vignes M. 2015. Exact and approximate inference in graphical models: variable elimination and beyond. *arXiv e-print*. .
- Pickrell JK, Pritchard JK. 2012. Inference of Population Splits and Mixtures from Genome-Wide Allele Frequency Data. *PLoS Genetics*. 8:e1002967.
- Pybus OG, Suchard MA, Lemey P, et al. (11 co-authors). 2012. Unifying the spatial epidemiology and molecular evolution of emerging epidemics. *Proceedings of the National Academy of Sciences*. 109:15066–15071.
- R Core Team. 2017. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria.
- Rabiner L. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*. 77:257–286.
- Rabosky DL. 2014. Automatic detection of key innovations, rate shifts, and diversity-dependence on phylogenetic trees. *PLoS ONE*. 9.
- Rabosky DL, Donnellan SC, Grundler M, Lovette IJ. 2014. Analysis and Visualization of Complex Macroevolutionary Dynamics: An Example from Australian Scincid Lizards. *Systematic Biology*. 63:610–627.
- Rabosky DL, Mitchell JS, Chang J. 2017. Is BAMM Flawed? Theoretical and Practical Concerns in the Analysis of Multi-Rate Diversification Models. *Systematic Biology*. 66:477–498.
- Rabosky DL, Santini F, Eastman J, Smith SA, Sidlauskas B, Chang J, Alfaro ME. 2013. Rates of speciation and morphological evolution are correlated across the largest vertebrate radiation. *Nature Communications*. 4:1–8.
- Reitan T, Schweder T, Henderiks J. 2012. Phenotypic evolution studied by layered stochastic differential equations. *The Annals of Applied Statistics*. 6:1531–1551.
- Revell LJ. 2009. Size-correction and principal components for interspecific comparative studies. *Evolution*. 63:3258–3268.
- Revell LJ. 2012. phytools: An R package for phylogenetic comparative biology (and other things). *Methods in Ecology and Evolution*. 3:217–223.
- Rigail G. 2015. A pruned dynamic programming algorithm to recover the best segmentations with  $K_{\max}$  change-points. *Journal de la Société française de Statistiques*. 156:180–205.
- Robin S. 2014. Models with Hidden Structure: Application to Genomics. Technical report.
- Rose JP, Kriebel R, Sytsma KJ. 2016. Shape analysis of moss (Bryophyta) sporophytes: Insights into land plant evolution. *American Journal of Botany*. 103:652–662.
- Rousseau JJ. 1762a. *Émile ou De l'éducation*. Paris: Nicolas Bonaventure Duchesne.

- Rousseau JJ. 1762b. *Emile* (Translation Barbara Foxley). London: J.M. Dent and Sons 1921.
- Rousseeuw P, Croux C, Todorov V, Ruckstuhl A, Salibián-Barrera M, Verbeke T, Koller M, Maechler M. 2014. *robustbase: Basic Robust Statistics*.
- Rubin DB. 1976. Inference and missing data. *Biometrika*. 63:581–592.
- Sagitov S, Bartoszek K. 2012. Interspecies correlation for neutrally evolving traits. *Journal of Theoretical Biology*. 309:11–19.
- Sankoff D. 1975. Minimal Mutation Trees of Sequences. *SIAM Journal on Applied Mathematics*. 28:35–42.
- Searle SR. 1987. *Linear Models for Unbalanced Data*. Wiley series edition.
- Self SG, Liang KY. 1987. Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests Under Nonstandard Conditions. *Journal of the American Statistical Association*. 82:605–610.
- Semple C, Steel M. 2003. *Phylogenetics*. Oxford University Press, oxford lec edition.
- Shi JJ, Rabosky DL. 2015. Speciation dynamics during the global radiation of extant bats. *Evolution*. 69:1528–1545.
- Silvestro D, Kostikova A, Litsios G, Pearman PB, Salamin N. 2015. Measurement errors should always be incorporated in phylogenetic comparative analysis. *Methods in Ecology and Evolution*. 6:340–346.
- Simon N, Friedman J, Hastie T, Tibshirani R. 2013. A Sparse-Group Lasso. *Journal of Computational and Graphical Statistics*. 22:231–245.
- Simpson GG. 1944. *Tempo and Mode in Evolution*. A Wartime book. Columbia University Press.
- Slater GJ, Harmon LJ, Alfaro ME. 2012. *Integrating fossils with molecular phylogenies improves inferences of trait evolution*. .
- Solís-Lemus C, Ané C. 2016. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. *PLoS Genetics*. 12:e1005896.
- Solís-Lemus C, Bastide P, Ané C. 2017. *PhyloNetworks: a package for phylogenetic networks*. *Molecular Biology and Evolution*. .
- Solís-Lemus C, Yang M, Ané C. 2016. Inconsistency of Species Tree Methods under Gene Flow. *Systematic Biology*. 65:843–851.
- Soucy SM, Huang J, Gogarten JP. 2015. Horizontal gene transfer: building the web of life. *Nature Reviews Genetics*. 16:472–482.
- St John K. 2016. Review Paper: The Shape of Phylogenetic Treespace. *Systematic Biology*. 66:syw025.

- Stadler T. 2011. Simulating Trees with a Fixed Number of Extant Species. *Systematic Biology*. 60:676–684.
- Stayton CT. 2015. What does convergent evolution mean? The interpretation of convergence and its implications in the search for limits to evolution. *Interface focus*. 5:20150039.
- Steel M. 1992. The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification*. 9:91–116.
- Stuart YE, Campbell TS, Hohenlohe PA, Reynolds RG, Revell LJ, Losos JB. 2014. Rapid evolution of a native species following invasion by a congener. *Science*. 346:463–466.
- Tabouy T, Barbillon P, Chiquet J. 2017. Inference under Missing Data Conditions in the Stochastic Block Model. *arXiv e-print*. .
- Tamura K, Nei M. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. *Molecular biology and evolution*. 10:512–26.
- Thomas GH, Cooper N, Venditti C, Meade A, Freckleton RP. 2014. Bias and measurement error in comparative analyses: a case study with the Ornstein Uhlenbeck model. Technical report.
- Tibshirani R. 1996. Regression Selection and Shrinkage via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*. 58:267–288.
- Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. 2005. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 67:91–108.
- Tolkoff MR, Alfaro ML, Baele G, Lemey P, Suchard MA. 2017. Phylogenetic Factor Analysis. *arXiv e-print*. .
- Tuffley C, Steel M. 1997. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bulletin of mathematical biology*. 59:581–607.
- Uyeda JC, Caetano DS, Pennell MW. 2015. Comparative Analysis of Principal Components Can be Misleading. *Systematic Biology*. 64:677–689.
- Uyeda JC, Harmon LJ. 2014. A Novel Bayesian Method for Inferring and Interpreting the Dynamics of Adaptive Landscapes from Phylogenetic Comparative Data. *Systematic Biology*. 63:902–918.
- Voltaire. 1772. Les Cabales. In: Œuvres Complètes, Texte établi par Louis Moland, Paris: Garnier, chapter 10, pp. 177 – 186.
- Wainwright MJ, Jordan MI. 2007. Graphical Models, Exponential Families, and Variational Inference. *Foundations and Trends® in Machine Learning*. 1:1–305.
- Wakeley J. 2009. Coalescent Theory: An Introduction. reewood Village Roberts and Company Publishers.



- Weston S. 2014a. doParallel: Foreach parallel adaptor for the parallel package.
- Weston S. 2014b. foreach: Foreach looping construct for R.
- Wickham H. 2007. Reshaping Data with the {reshape} Package. *Journal of Statistical Software*. 21:1–20.
- Wickham H. 2009. ggplot2. New York, NY: Springer New York, springer edition.
- Yang Z. 1993. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Molecular biology and evolution*. 10:1396–401.
- Yang Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods. *Journal of Molecular Evolution*. 39:306–314.
- Yang Z. 1995. A space-time process model for the evolution of DNA sequences. *Genetics*. 139:993–1005.
- Yu Y, Degnan JH, Nakhleh L. 2012. The Probability of a Gene Tree Topology within a Phylogenetic Network with Applications to Hybridization Detection. *PLoS Genetics*. 8:e1002660.
- Yu Y, Dong J, Liu KJ, Nakhleh L. 2014. Maximum likelihood inference of reticulate evolutionary histories. *PNAS*. 111:16448–16453.
- Yu Y, Nakhleh L. 2015. A maximum pseudo-likelihood approach for phylogenetic networks. *BMC Genomics*. 16:S10.
- Yuan M, Lin Y. 2006. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 68:49–67.
- Zuckerkandl E, Pauling L. 1965. Molecules as documents of evolutionary history. *Journal of theoretical biology*. 8:357–366.

**Titre :** Modèles de processus stochastiques avec sauts sur arbres : application à l'évolution adaptative sur des phylogénies

**Mots clefs :** Sélection de Modèle, Écologie Comparative, Phylogénie, Orstein-Uhlenbeck

**Résumé :** Le projet s'inscrit dans la dynamique de systématisation statistique qui s'opère aujourd'hui dans le champ de l'écologie comparative. Les différents traits quantitatifs d'un jeu d'espèces échantillonné peuvent être vus comme le résultat d'un processus stochastique courant le long d'un arbre phylogénétique, ce qui permet de prendre en compte des corrélations issues d'histoires évolutives communes. Certains changements environnementaux peuvent produire un déplacement de niches évolutive, qui se traduisent par un saut dans la valeur du processus stochastique décrivant l'évolution au cours du temps du trait des espèces concernées.

Parce qu'on ne mesure la valeur du processus dynamique qu'à un seul instant, pour les espèces actuelles, certains scénarii d'évolution ne peuvent être reconstitués, ou présentent des problèmes d'identifiabilité, que l'on étudie avec soin. On construit ici un modèle

à données incomplètes d'inférence statistique, que l'on implémente efficacement. La position des sauts est détectée de manière automatique, et leur nombre est choisi grâce à une procédure de sélection de modèle adaptée à la structure du problème, et pour laquelle on dispose de certaines garanties théoriques.

Un arbre phylogénétique ne prend pas en compte les phénomènes d'hybridation ou de transferts de gènes horizontaux, qui sont fréquents dans certains groupes d'organismes, comme les plantes ou les bactéries. Pour pallier ce problème, on utilise alors un réseau phylogénétique, pour lequel on propose une adaptation du modèle d'évolution de traits quantitatifs décrit précédemment. Ce modèle permet d'étudier l'hétérosis, qui se manifeste lorsqu'un hybride présente un trait d'une valeur exceptionnelle par rapport à celles de ses deux parents.

**Title :** Shifted stochastic processes evolving on trees: application to models of adaptive evolution on phylogenies

**Keywords :** Model selection, Comparative ecology, Phylogeny, Orstein-Uhlenbeck

**Abstract :** This project is aiming at taking a step further in the process of systematic statistical modeling that is occurring in the field of comparative ecology. A way to account for correlations between quantitative traits of a set of sampled species due to common evolutionary histories is to see the current state as the result of a stochastic process running on a phylogenetic tree. Due to environmental changes, some ecological niches can shift in time, inducing a shift in the parameters values of the stochastic process modeling trait evolution. Because we only measure the value of the process at a single time point, for extant species, some evolutionary scenarios cannot be reconstructed, or have some identifiability issues, that we carefully study. We construct an incomplete-data model for statistical inference, along with an efficient implementation. We

perform an automatic shift detection, and choose the number of shifts thanks to a model selection procedure, specifically crafted to handle the special structure of the problem. Theoretical guarantees are derived in some special cases.

A phylogenetic tree cannot take into account hybridization or horizontal gene transfer events, that are widely spread in some groups of species, such as plants or bacterial organisms. A phylogenetic network can be used to deal with these events. We develop a new model of trait evolution on this kind of structure, that takes non-linear effects such as heterosis into account. Heterosis, or hybrid vigor or depression, is a well studied effect, that happens when a hybrid species has a trait value that is outside of the range of its two parents.