Phylogenetic Dimensionality Reduction in Non-linear Geometries

Marcus Teller
Supervisor: Stefan H. Sommer

Spring 2022

Abstract

This project analyses how dimensionality reduction can be performed on highly covariate biological traits. The geometry of the spaces in which these traits reside are often highly non-linear and thus a more novel approach than classical dimensionality reduction algorithms is required. The goal of these algorithms is to minimize the distorting effects of the implicit linear approximations performed on the surface of curved geometries. We first investigate the deep theory connected to both non-linear geometries, as well as the continuous trait models that model evolution as a stochastic process. Following this theory, we then propose two new algorithms; The first is a simple adaptation of a previously know algorithm of Polly et al. (2013) to curved domains. The second is similar to ordinary PCA except that it considers small increments of a possibly non-linear brownian motion process, whilst implicitly removing the effects of biological covariance. We apply both our two new methods, as well as previously known methods to a range of synthetic datasets for visual comparison. Finally we apply our method to a dataset of traits from birds, AVONET (Tobias et al., 2022), and conclude that it does provide a similar estimate to the algorithm of Polly et al. (2013), but has additional flexibility built in.

Contents

1 Introduction

If you take any introductory course or education in machine learning, chances are that you have probably seen PCA. It gives the user a convenient way of visualizing very high dimensional data in a lower dimensional space. This is far from the only dimensionality reduction algorithm, but as we will later discover, it synergizes well with normally distributed variables. There are however limitations that make PCA unideal when we have correlated samples, phylogenetic trees will be one such case.

When a set of species evolve in an isolated system under some evolutionary process, it is evident that some species go extinct, while other evolve new traits, or refine others. The traits of one species can be assumed to be passed on to the next generation, with some variation between generations. We call the record of species and their evolution a phylogeny. Essentially, it is a simple tree, that carries information about the time of which species split up and form new ones. As is the case under evolution, some traits are naturally advantageous to reproduction. Individuals of a species that posses such traits will thus have a higher probability of passing those exact traits onto the next generation.

Sometimes, there are two incompatible traits that are both more advantageous than the norm in the previous generation. A simple example of this would be a species of bird which at some point develops one type of beak specialized in hunting prey A, and another specialized break for hunting prey, B. A single individual cannot have both beaks at once (or at least it probably would not be something expected), however it is still better to have either beak, than the type of beak previous generations had. This leads to what is commonly referred to as disruptive evolution in biology, and is what can cause one species to become two over time. Another common reason for one species to become two, is an isolating factor, where suddenly not cross-breeding can occur. This would be the case if a set of individuals from a species moved elsewhere geographically for example.

As we will see throughout this project, one of the most popular ways of modelling evolution, is as a stochastic process that has an underlying gaussian distribution. We think of each species as developing randomly but independently, until certain splits occur which transform a single species into the starting point of two or more new species. A large downside which we will see is central to this project, is that when we consider any two species with some ancestor, their traits are not independent. A simple example of this, is the probability of the closest relative to some bird species having a beak. It is very unlikely that the relative itself does not have a beak, as it was most likely formed as a part of a common ancestor. We think of this trait as being due to common descent, where their beak was evolved over a lineage of ancestors, and then passed onto both species.

From a mathematical perspective, this makes visualization of traits that can be measured numerically, much harder. The problem is that whilst some traits may have natural correlation, we might over- or underestimate that correlation, due to the correlation that comes solely from the common ancestor of two species. Typically when performing PCA, we assume that all input vectors are independent samples of some correlated variables. In this project, we will see two methods of circumventing this.

We assume the reader of this project to have masters level knowledge of mathematical analysis and data science. We will only introduce the absolute basics of biology, as the focus of this project will be on modelling the evolutionary process mathematically. For this reason, we do not assume the reader knows the basics of biology.

First, we will see a method that aims to remove the covariance that comes from two species sharing a common ancestor. We present this method based on Polly et al. (2013) in section 5.2. We also propose a completely new way of performing PCA on phylogenetic trees, which considers the many small scale changes in a species. Whilst the result should not differ from the algorithm of Polly et al. (2013), we do conjecture that this method is more robust to the distortions that arise when dealing with highly non-linear data.

As we will see throughout this project, the non-linearity of traits is something that is central to the analysis of said traits. It is common, and also the case with Polly et al. (2013), to use landmark points to represent morphological traits of an individual in a species. The set of landmarks on a skull for example, does not have any immediate vector space structure. Taking the mean of two landmark sets does not guarantee that the result itself is a meaningful landmark configuration.

Although this example will be slightly exaggerated to convey the idea, it does illustrate the problem with blindly applying linear statistics to non-linear data. Consider the set of squares with side length 2, with center at $(0, 0)$. We could consider these as 4 landmarks points for some very square species, which we wish to compare. Say now that we sample two squares t_1, t_2, and calculate their mean in a pairwise manner over each point $\frac{1}{2}(t_1 + t_2)$. The result is visualized in figure 1, and as we can see, the mean square is suddenly scaled differently than other squares, giving what could have been a possibly wrong conclusion. The reason why we say this example is exaggerated, is because it is commonplace to first align points using procrustes. This does help in this instance where all points can be aligned perfectly, but that is not always the case.

In section 5.4 we show how the method of phylogenetic projections can be applied to non-linear data, by adapting it to a geometric structure called a manifold. The resulting algorithm is similar to the algorithm of section 5.2, with only key parts exchanged. We also briefly discuss how one might

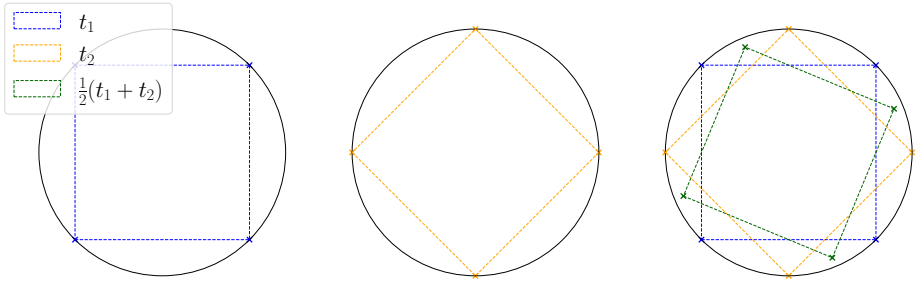

Figure 1: The mean of two squares does not preserve area

generalize our own algorithm to these non-linear geometries, and why it should be better suited.

In this project we introduce mathematical concepts that help us treat these kind of data. We will use a structure called a manifold and wish to introduce it in a way such that the reader feels somewhat comfortable with its application. Choosing what to cover and what not to cover, has been a hard choice. The overall aim when introducing new concepts is to cover them fairly rigorously if they are of use to the project. If we feel that a concept is mostly auxiliary or perhaps only used briefly, we do lack some rigor. An example of this is the definition of second-countability in the introduction of a manifold. Even though it is at the core of the definition of a manifold, we did not feel that it was important enough for the conceptualization of a manifold. The notion of local homeomorphisms are absolutely central in this project, and for that reason we cover it more deeply.

After introducing the notion of a manifold. We will investigate how we can mathematically model the process of evolution, as as stochastic process. We show its general definition both in linear and non-linear geometries, and its deep connection with the gaussian distribution.

We then show how we can use apply this model to subsets of the evolutionary tree called phylogenies, and go into details about different ways of viewing and parameterizing distributions over them.

We then look at 4 different variants of principal component analysis, each adapted either to a non-linear geometry or to highly covariate data, or both. We provide our own algorithm, which is adapted to both settings at once. After this we propose another algorithm which works locally rather than globally, which should reduce the inherent estimation error that arises when approximating curvature with flat spaces.

2 Representing data as points on a manifold

When dealing with data in machine learning or statistics, we often make many assumptions about the structure of the space in which data points reside. Let $x_1, x_2, ..., x_n \in U$ be a set of data points sampled from some space U. For many practical purposes, $U = \mathbb{R}^d$, which has many convenient properties, such as $\frac{1}{n}\sum x_i \in U$, $x_i + x_j \in U$ and $cx_i \in U$. Properties similar to these are often used in many statistical algorithms but as will be seen throughout this project, is not something that we can generally assume. To see an example of one such case, consider $U = \mathbb{S}^1 = \{v \mid ||v||^2 = 1, v \in \mathbb{R}^2\}$. One could imagine the elements of this set as being the directions of movements observed in some 2d ants. Say now, we observe n ants and sample their movement at any given time. A natural question could be, what is their average direction of travel? The answer to this question is expected to be in U, but $\frac{1}{n}\sum x_i$ is not guaranteed to be.

One may also notice if we only sampled two ants going in opposite directions (one north, one south), the answer to this question is ambiguous, since both east and west could represent what we would intuitively call average direction. The space in this example however, does have significantly more structure than an arbitrary set, it simply lacks the familiar linear structure. A concept that will be formalized later is the concept of local structure, which is innate to the geometric object we call manifolds. If we consider any point $x_i \in \mathbb{S}^1$, we can imagine zooming really far in and observing the space around x_i. If zoomed far enough, this space resembles a line, which we know has linear properties. This, as we will see later, becomes a key concept of manifolds.

The introduction of manifolds is motivated by examples like these, where linear structure is too harsh of a requirement, but where we still want to quantify the geometry of the space. This project aims to examine and provide a detailed description of classical algorithms in this new setting, as well as fill in the mathematical detail required for understanding them. Our choice of data will mostly be of biological nature, as it is modeled very well by riemannian manifolds.

2.1 An introduction to Topology

To understand how we formalize the local structure of a manifold, we need a notion of both neighborhoods and continuous maps. In this section we will give a brief introduction to the absolute basics of topology. It is assumed the reader is somewhat familiar with the notion of metric spaces, as well as basic topological concepts such as open sets and ordinary continuous maps on the form $\mathbb{R}^m \to \mathbb{R}^n$.

In topology we are interested in defining the notion of open sets of an ambient space X. We do this by simply defining any subset $U \subseteq X$ to be open iff $U \in \mathcal{T}$ where $\mathcal{T}$ is called the topology on X (Munkres, 2000).

Definition 2.1 (Toplogy)**.** We say any collection of subsets τ is a topology on X iff

1. $\{\varnothing, X\} \subseteq \tau$.
2. Any possibly infinite union of elements in τ, is itself an element of τ.
3. Any finite intersection of elements in τ, is itself an element of τ

Construction of topologies is of little interest for this project, however we encourage the reader to simply think of a topology as a collection of all the open sets on X. With this, we can be very precise when defining locality around a point $x \in X$. A very convenient definition, is the notion of a neighborhood.

Definition 2.2 (Neighborhood)**.** Consider the set X equipped with topology $\mathcal{T}$. Any open set $U \in \mathcal{T}$ is a neighborhood of $x \in X$ iff $x \in U$.

This definition is used widely throughout topology, and will be central in the definition of a manifold. Notice that there is no restriction on the "size" of a neighborhood, we simply require it to be open and contain a specific x.

The choice of $\mathcal{T}$ will lead to various different properties of X, but for our purpose we can think of $\mathcal{T}$ as the collection of all sets we would "usually" consider open in a standard metric space (in this case $\mathcal{T}$ is called the metric topology on X). We say the pair $(X, \mathcal{T})$ is a topological space, and when we define open mappings, it will be between two topological spaces.

Definition 2.3 (Continuity)**.** Consider topological spaces X and Y. We define any map $f : X \to Y$ to be continuous iff $f^{-1}(V)$ is open in X where V is open in Y

That is, we define any function whose preimage of an open set, is itself open. This definition tells us that the continuity of a function does not only depend on the function itself, but also on the topology on the spaces it maps between. A relevant example is where the topology on X is the discrete topology, defined as $\mathcal{T}_X = \mathcal{P}(X)$, where $\mathcal{P}$ is the power set. In this topology, every subset of X is open, and thus every map from X is continuous, since the preimage of any map from X is always open.

We are now ready to define a key concept in manifolds; the homeomorphism. One of the classical examples people think of when they hear topology, is the statement that a coffee cup is the same as a donut. While the statement itself is mystifying, what is really meant is that there exists a bijective map between the two spaces which preserves topological structure. A way of thinking about this is that we can use homeomorphisms to define equivalences between

two different topological spaces. Loosely speaking they are what form the equality signs between topological spaces. To see why, we must first define a homeomorphism.

Definition 2.4 (Homeomorphism)**.** A bijection $f : X \to Y$ is said to be a homeomorphism iff f and f^{-1} is continuous.

An equivalent statement is the following

Remark. Any bijection $f : X \to Y$ where $f(U) \in \mathcal{T}_Y \Leftrightarrow U \in \mathcal{T}_X$ is a homeomorphism.

Proof. Since f is a bijection we have $(f^{-1})^{-1} = f$. Let $U \in \mathcal{T}_X$, then f^{-1} is continuous since for any $U \in \mathcal{T}_X$ its preimage under the inverse is the same as $f(U)$ which is assumed to be open. f is shown continuous in the same way. □

We say that homeomorphisms preserve structure, because any open set in X will have an equivalent open set in Y. We can think of this as mapping the entire space to another, where all the open sets are preserved. This preserves structure, because in topology, spaces are structured by their open sets. More specifically properties of spaces are expressed in terms of the respective topologies.

We say two spaces X and Y are homeomorphic iff there exists a homeomorphism between them. This condition can be relaxed, and we will mostly use a relaxation of this by defining local homeomorphisms.

Definition 2.5 (Local Homeomorphism)**.** A function $f : X \to Y$ is a local homeomorphism if for every point $x \in X$, there exists a neighborhood U of x, where U is homeomorphic with an open subset of Y.

We note that quite trivially every homeomorphism is a local homeomorphism. We also say that two spaces X and Y are locally homeomorphic if there exists a local homeomorphism between them.

2.2 Defining a manifold

With these basic topological tools, we are mostly ready to define a manifold. We define a manifold $\mathcal{M}$ to be a topological space that is both hausdorff and second-countable, we did not cover these definition and refer the reader to Munkres (2000). In essence they are constraints, such that the topology is well-behaved. More importantly for this project will be the last condition

Definition 2.6 (Manifold)**.** We say a second-countable hausdorff space $\mathcal{M}$ is an n dimensional manifold if it is locally homeomorphic to $\mathbb{R}^n$

While this definition may not give an immediate intuition of what a manifold is, we can revisit the 2d ants of the introduction. If you asked any ant on planet earth whether they think earth is flat or has curvature, they would

most likely think it is flat. From their perspective, they are living in a plane, and probably have no reason to believe otherwise. Since ants are really small, we can imagine looking at an ant a on the sphere $\mathcal{M} = \mathbb{S}^2$. Since $\mathbb{S}^2$ is a manifold, we know that there must exist a neighborhood U around a which is homeomorphic to a subset of $\mathbb{R}^2$. Intuitively this means that from the ants perspective if we zoom in to the neighborhood U (and ants are really small so zooming makes sense) the structure of the space is equivalent to a subset of a plane.

Lets say two ants a and b live in two different countries sharing a common border. We might ask which way a needs to go to visit its friend, ant b. To solve this problem, a natural and very familiar construction is made; the notion of a chart. The idea is to differentiably map open sets on the surface of $\mathcal{M}$ to open sets of $\mathbb{R}^d$, such that any navigation in $\mathcal{M}$ can be done by navigating in charts. For a to navigate to b, it now simply needs to find a chart that maps a set U which contains both a and b to $\mathbb{R}^2$. We define charts the following way;

Definition 2.7 (Chart)**.** A chart on a d-dimensional manifold $\mathcal{M}$ is a C^r mapping $\phi : U \to V$ where U is open in $\mathcal{M}$ and V is open in $\mathbb{R}^d$, with C^r inverse ϕ^{-1}.

This is the general definition of a chart, however for this project we will almost exclusively deal with differentiable manifolds, and as a consequence put the additional requirement on any chart ϕ, that both ϕ and ϕ^{-1} are differentiable (ie. $r \geq 1$). These charts capture the essence of definition 2.6, as they are the equivalent of the local homeomorphisms (now diffeomorphisms since we often require they are differentiable). With these charts, we now have a way to treat neighborhoods of points of the manifold as if they were open sets of $\mathbb{R}^d$.

In the ant analogy, we can imagine every ant country as having their own chart, specifying how we could navigate around in the specific country to which the respective chart belongs. This could be problematic if no charts overlap, as navigating over any border would be impossible. For this reason it is reasonable to require charts to both cover $\mathcal{M}$, but also that we can compose maps without distortion or ambiguity at their overlaps. We call a collection of such charts an Atlas, which we define the following way

Definition 2.8 (Atlas)**.** We say a family of charts $(\phi_i)_{i=1..N}$, $\phi_i : U_i \to V_i$ is an atlas of $\mathcal{M}$ iff

1. *(Covering)* For any $x \in \mathcal{M}$, there exists a chart ϕ_i with $x \in U_i$.
2. *(Compatibility)* For any charts ϕ_i, ϕ_j with $U_i \cap U_j \neq \varnothing$. The composition $\phi_i \circ \phi_j^{-1} : \phi_j(U_i \cap U_j) \to \mathbb{R}^d$ must be differentiable.

This definition together with definition 2.7, gives us a way to represent any neighborhood of $x \in \mathcal{M}$, as a neighborhood of $\mathbb{R}^d$. This allows us to obtain local coordinate systems, which we know from definition 2.8 must be compatible.

By this construction we come very close to a coordinate system on $\mathcal{M}$, except that it is manifested by composition of local systems. More interestingly we gain properties of $\mathbb{R}^d$, pulled back to $\mathcal{M}$ via its atlas. We first introduced the topological manifold where we simply required neighborhoods of points to be homeomorphic to $\mathbb{R}^d$. This is the case where charts are C^r maps with $r = 0$. In a similar fashion we say for $r = \infty$ that $\mathcal{M}$ is a smooth manifold, and for $r \geq 1$ that $\mathcal{M}$ is a differentiable manifold.

2.3 Curves and tangents

Generalizing the notion of a straight line is something that will be of great importance to us. Notice that charts themselves are not guaranteed to encode directions as we might imagine. Thus moving in one direction is not as simple as simply moving in a fixed direction in a chart, and them mapping back to a manifold. To do this, we will need to define what *straight* is in a possibly curved geometry.

We will start by introducing the notion of a curve on a manifold $\mathcal{M}$, which we then extend with the definition of a *straight* curve called a geodesic. If we start with an ordinary curve $\gamma : I \to \mathbb{R}^k$, where I is sub sub interval of $\mathbb{R}$, we know from ordinary analysis that we can find its tangent by taking its coordinate-wise derivative $(\gamma_1'(t), \ldots, \gamma_k'(t))$.

The problem arises when we consider a curve $\gamma : I \to \mathcal{M}$. We can no longer make immediate sense of the coordinate-wise derivatives, since any element $x \in \mathcal{M}$ is simply some abstract algebraic entity. We saw in the previous section how we can induce differentiability to the manifold via charts. We are going to do exactly that now, and notice that the composition $\phi \circ \gamma : I \to \mathbb{R}^d$, is an ordinary curve in $\mathbb{R}^d$ if ϕ is an ordinary curve in $\mathbb{R}^d$. It is worth noting that due to the covering assumption and the openness of the domain of all charts, we can guarantee that there is a sequence of charts in the atlas that covers the co-domain of γ. For simplicity we encourage to simply think of γ as being mapped under a single chart.

Using these constructions, we can obtain all tangent vectors in a point $x \in \mathcal{M}$ which we call the tangent space $T_x\mathcal{M}$, by considering the set of all curves going through x, and taking their euclidean derivatives. Since many different curves may have identical derivatives, we would technically need to define each tangent as an equivalence class over curves. Another immediate problem with this is that have a very explicit presence of charts. We want all tangents to be completely independent of the choice of charts, which requires some work to show that this is in fact the case under this construction.

We will mention another way of defining tangents to a point $x \in \mathcal{M}$, which may seem less intuitive at a first glance. As in Pennec et al. (2020), we consider the set of tangent vectors around a point x, as a vector space of differential

operators. Around a point x, we can induce a basis via a chart, which is simply consists of derivations with respect to each coordinate in the chart $\frac{\partial}{\partial \phi(x)_i}$. These do indeed form a vector space, and any vector tangent to the manifold can be thought of as a linear combination of these.

Central to the definition of a geodesic is the concept of curve length. The problem however, is that any tangent of a curve γ, is in the tangent space around some x, which we haven't defined any metric on. The riemannian metric is exactly that.

Definition 2.9 (Riemannian Metric)**.** For a manifold $\mathcal{M}$, a riemannian metric in a point $x \in \mathcal{M}$ is a positive definite bilinear map $\langle \cdot, \cdot \rangle_x : T_x\mathcal{M} \to T_x\mathcal{M}$. It induces a norm on a tangent v of x, given by $||v||^2 = \langle v, v \rangle_x$ (Pennec et al., 2020).

We say that a manifold that is both smooth, and has such a metric, is a riemannian manifold. This solves a series of problems, since we now have a way of measuring the instantaneous *speed* of a curve going through a point x. With this we can simply define curve length as we would in an euclidean domain

Definition 2.10 (Curve Length)**.** For a curve $\gamma : I \to \mathcal{M}$ and a riemannian manifold $\mathcal{M}$, the length of γ is defined to be

$$\mathcal{L}(\gamma) = \int_I ||\gamma'(t)||_{\gamma(t)} \mathrm{d}t$$

With this, we can now define the distance between two points s, t as the infimum of lengths of all curves between the two. Locally, these shortest paths will be straight lines, just as intuition will expect.

There are two ways of trying to define the manifold version of a straight line, which we call the geodesic. The first approach, is to define it as a curve that minimizes distance locally. This is inline with the intuition, but the locality makes it a bit hard to conceptualize.

Another way is to define a geodesic as a curve that minimizes an energy functional. The intuition is that a straight line is the most efficient way between points on the surface. For a curve γ the energy functional $\mathcal{E}$ is defined as

$$\mathcal{E}(\gamma) = \frac{1}{2} \int_a^b ||\gamma'||^2_{\gamma(t)} \mathrm{d}t$$

Which gives rise to the definition of a geodesic.

Definition 2.11 (Geodesic)**.** Any curve γ is a geodesic iff it is a minimum of $\mathcal{E}(\gamma)$.

This has a connection with curve length, as any critical point of $\mathcal{E}$ is also a critical point of $\mathcal{L}$. Furthermore geodesics have the property of being self-parallel,

or in other words, they do not have acceleration (second order derivatives are zero).

We are now ready to introduce the notion of exponential and logarithm maps, which will be important when representing points on the manifold as vectors, and tying into the vector-space structure of tangent spaces.

Given a point $x \in \mathcal{M}$, and a tangent vector $v \in T_x\mathcal{M}$, we can imagine shooting out a curve γ from x, with velocity v, and see where it ends up at time $t = 1$.

Definition 2.12 (The Exponential map)**.** Given a point $x \in \mathcal{M}$, and a tangent vector $v \in T_x\mathcal{M}$, the exponential map at a point x, is the point $\mathrm{Exp}_x(v) = \gamma(1)$ where γ is the unique geodesic with $\gamma(0) = x$ and $\gamma' = v$.

Similarly we can define its inverse

Definition 2.13 (The Logarithmic map)**.** Given points $x, y \in \mathcal{M}$, the tangent vector $v = \mathrm{Log}_x(y)$ is the smallest tangent in $T_x\mathcal{M}$ that satisfies $\mathrm{Exp}_x(v) = y$.

These two maps are convenient for mapping between tangent spaces and the manifold. In a sense we can suddenly linearize neighborhoods of points, via the logarithmic mapping, since they give a first order vector representation of the local geometry. This way, it can be used to obtain a representative vector space for the local geometry of a point.

2.4 The LDDMM framework and a representation of landmarks

We have now introduced a range of new definitions that all relate to manifold data, but we haven't really described how this construction could look in practice. In our specific case, we will give an introduction to how one might use shape data in the context of non-linear statistics.

Shape data can take many forms, a common one being a set of k landmarks embedded in an embedding space $\Omega = \mathbb{R}^d$. It could also be a closed curve tracing the outline of some shape γ. This makes it hard to endow structure into a single underlying representation for every single shape instance, and would lead to non-generalization of algorithms since they would be implemented for a specific shape representation.

The idea of the LDDMM framework, is to work on the deformation group, rather than on the representations of shape directly. Instead of asking, how similar a set of landmarks is to a curve, which would be a very hard question to answer due to the different representations. We instead ask questions about the deformation that transforms a set of landmarks to lie on top of said curve.

We define a diffeomorphism to be any map $\phi : \Omega \to \Omega$ that is differentiable, and has a differentiable inverse. We often refer to the diffeomorphisms as actions, as they can alter a shape by considering said shapes representation in the warped

domain under ϕ. Imagine some landmark representation $q = (x_1, \ldots, x_n)$, where $x_i \in \Omega$. Then, we can apply an action ϕ to q by simply applying the action to each landmark. We usually denote the application of an action to a shape representation via the notation $\phi.q$. In this example we have

$$\phi.q = (\phi(x_1), \ldots, \phi(x_n))$$

As mentioned, we might also have a shape represented as some closed curve $\gamma : \mathbb{S}^1 \to \Omega$. We now perform the exact same operation, except to every single point of the curve γ. This is done by considering the curve that is the composition $\phi.\gamma = \phi \circ \gamma$. The space of these actions on Ω which we denote $\mathrm{Diff}(\Omega)$, is itself a manifold, something that we will use frequently.

This differentiable manifold which we will refer to as the diffeomorphism group $\mathrm{Diff}(\Omega)$, can also be endowed with a riemannian metric, which lets us define the distance between two shapes, as being the geodesic length between two deformations in $\mathrm{Diff}(\Omega)$. We do this by having a template shape, and then representing each other shape, as being a deformation of the template.

The beauty of this representation, is that we can now construct all of our algorithms to work on $\mathrm{Diff}(\Omega)$, and then apply them to any shape representation, be it a set of landmarks, or a dataset of curves. As an example we may have two shapes q_1, q_2, and ask which action ϕ transforms q_1 into q_2, that is to find a ϕ such that $\phi.q_1 = q_2$. It turns out that this can be framed as an optimization problem to find a geodesic in $\mathrm{Diff}(\Omega)$, which illustrates how powerful this abstraction is.

The notion of geodesics is important when working with the LDDMM framework for problems such as the above mentioned. For this reason simplifications of the energy functional $\mathcal{E}$ can ease analysis greatly. We do not introduce the notion of lie groups, however we do note that they are both manifolds and groups where the tangent space around the identity map Id has desirable properties. To use the properties, we need to find a way to transport tangent vectors of arbitrary tangent spaces $T_\phi\mathrm{Diff}(\Omega)$ to the tangent space $T_{\mathrm{Id}}\mathrm{Diff}(\Omega)$. First we consider the notion of a right translation, which we define to be $R_\phi\psi = \psi \circ \phi$. Along with its pushforward $(R_\phi)_*$ which is the derivative of R_ϕ. With this there now exists a unique vector $h \in T_{\mathrm{Id}}\mathrm{Diff}(\Omega)$ for each $u \in T_\phi\mathrm{Diff}(\Omega)$ were $(R_\phi)_* h = u$. This induces an inner product on each $T_\phi\mathrm{Diff}(\Omega)$ which is simply the inner product in $T_{\mathrm{Id}}\mathrm{Diff}(\Omega)$ (which is defined) of the translated vectors. This gives what we call a right-invariant metric, and as we will see when introducing brownian motion, invariance of a metric can simplify definitions greatly.

Take any curve $\phi(t)$ in $\mathrm{Diff}(\Omega)$, now we can associate with it a family of vector fields by translating the time-dependent tangents $v(t)$ of $\phi(t)$ to $T_{\mathrm{Id}}\mathrm{Diff}(\Omega)$ via $v(t) = (R_{\phi^{-1}(t)})_* \phi'(t)$. Now, we can simply define the energy of any curve as a function of $v(t)$, since it as mentioned previously is an element of what we call the lie algebra $T_{\mathrm{Id}}\mathrm{Diff}(\Omega)$, and thus already has a norm defined. We can define

the simple squared norm $l(v) = ||v||^2$, and we thus get the energy equation again in terms of elements of a lie algebra

$$\mathcal{E}(\phi) = \frac{1}{2}\int_a^b l(v(t))\mathrm{d}t$$

Via a series of simplifications, this gives rise to what is called the Euler-Poincaré equations which produce a new interpretation of the structure of $\mathrm{Diff}(\Omega)$. Although we do not show these explicitly, we note that we can construct diffeomorphisms from a vector field $m(t)$ on Ω, which we call the momentum field. It is closely related to the velocity field $v(t)$, and for the landmark case, the momentum field arising as solution to a range of problems, will be a local initial momentum of each landmark point.

3 Brownian Motion

A tool which will be used repeatedly in this project, is Brownian motion. It was originally developed as a model used to describe the movement of small particles in fluid, which collided with other objects (Resnick, 1992). In this project, we will use Brownian motion as a model for how traits evolve over time. It has an underlying connection with the normal distribution, which makes it convenient for the manifold settings, where it gives us a natural way to sample from a normal distribution. Brownian motion also has an array of useful properties, which has been central in previous analysis of evolution, but also some of which we will use when introducing our own projection algorithms.

We will start by defining what a stochastic process is, following the definition of Ross (1996). Consider a simple ant, each day, it wakes up and takes exactly one step to the right, or one step to the left. Let p be the probability that the ant walks right, and $1-p$ the probability that it walks left. We are now interested in observing where the ant ends up at any timestep $n \geq 0$. We can think of the decision of the ant at timestep $t > 0$, X_t to be 1 if it walks right, and -1 if it walks left. If we want to know the location of the ant at timestep n, we simply sum up all of the decisions it made at each timestep up until time n. Thus, we define the posititon at time n, $S_n = \sum_{t=1}^{n} X_t$. Intutitively, the set of all S_n's will act as a traceroute for the ant. If we run this experiment with $p = 1/2$, and plot S_n vs n, we get figure 2. What we see, is that the ant ended up somewhere far on the left. We call the set of all of these steps, $S = \{S_n, n \geq 0\}$ a stochastic process. In other words, we define a stochastic process to be the set of some random variables (will be rigorously defined later on). In our case, each increment is trivially countable since the index set $I = \mathbb{N}_0$ used to index S_n is itself countable. This is however not generally the case.

We will now look at what we will refer to as continuous-time stochastic processes, where the index set I is a compact connected metric space. With this,

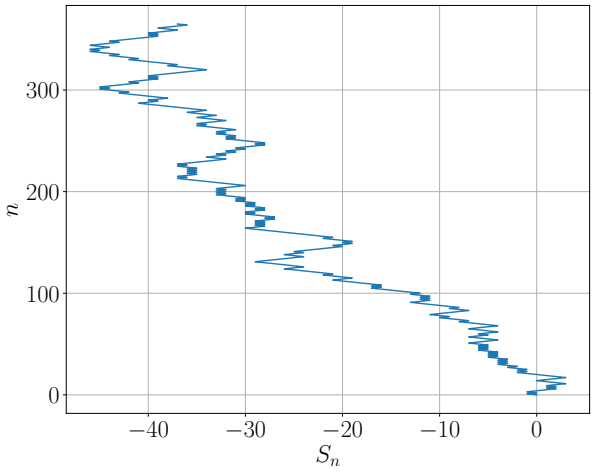

Figure 2: Plot of S_n

we can also generalize the stochastic process to be any set $X = \{X(t), t \in I\}$. Before we get any further, we will first introduce the notion of a random variable similar to Ross (1996), as it often is taught in a very vague sense.

Definition 3.1 (Random Variable)**.** Given a probability space $(\Omega, \mathcal{F}, P)$, a random variable $X : \Omega \to \mathbb{R}$ is a function that maps elements of the sample space Ω to a real value. We also require that it has a measurable probability of some set of events, which is done by the additional requirement that for each real x, $\{\omega : X(\omega) \leq x\} \in \mathcal{F}$.

This definition may seem alienating for those unfamiliar with measure theoretic concepts, who most likely where taught that random variables just magically took on different values. Intuitively we can think of random variables as being functions that lets us pull back values of $\mathbb{R}$ such that they can be assigned a probability by P. It turns out that this concept gives lead to the probability of a random variable X taking on values in some measurable set $A \subset \mathbb{R}$. We say the the probability of X taking on values in A is given as the push-forward measure $p_X(A) = P(X^{-1}(A))$. This is rather intuitive, we simply find all the elements in the sample space that makes X take on values in A, and then measure that exact set. We are often interested in what we call the probability distribution of some rv. X we define it as the probability that X is at most some other value x.

Definition 3.2 (Distribution of a random variable)**.** Given a probability space $(\Omega, \mathcal{F}, P)$, the distribution $F(x)$ of a random variable X, is given as

$$F(x) = P(X \leq x) = P(X^{-1}(\{a : a \leq x\}))$$

And now elegantly, we can define the probability density for a random variable X.

Definition 3.3 (Distribution of a random variable)**.** Given a probability space $(\Omega, \mathcal{F}, P)$, the probability density $f(x)$ of a random variable X, is the function f that satisfies

$$p_X(B) = \int_{X^{-1}(B)} \mathrm{d}P = \int_B f \mathrm{d}\lambda$$

Where λ is the standard lebesgue measure. If no such function exist, we say X is a discrete random variable. If f does exist, we say X is a continuous random variable.

The solution to this equation is given by the radon-nikodym theorem, and is

$$f = \frac{\mathrm{d}(P.X^{-1})}{\mathrm{d}\lambda}$$

Where $\frac{\mathrm{d}y}{\mathrm{d}x}$ is the radon-nikodym derivative. This pops right out of the theorem because we have a change of variable with respect to the push forward measure $P.X^{-1}$. For those new to measure theory, it is natural to think of the probability density of a random variable as the ordinary derivative of the distribution.

With these elementary definitions in place, we can now be a bit more precise. When talking about a stochastic process, we define it the following way.

Definition 3.4 (Stochastic Process)**.** Given a probability space $(\Omega, \mathcal{F}, P)$ and an index set I, a stochastic process X is a collection of random variables

$$X = \{X(t, \omega),\, t \in I\}$$

where $\omega \in \Omega$ and $X(t, \cdot)$ is a random variable.

Although this definition is more mathematically robust, it is slightly more heavy to work with. For simplicity, we will abbreviate and simply write $X = \{X(t),\, t \in I\}$, where we hide that $X(t)$ is in fact also a function of ω. This is similar to how we often simply abbreviate random variables X or Y, when we infact mean $X(\omega)$ and $Y(\omega)$. We recommend the reader to not think too hard about the importance of the measure theoretic foundation, but rather think of it as a mathematical practicality in some cases. It simply gives the required rigorousness, and lets us work under a well studied framework that is measure theory.

This does give a more clear sense of what it means to sample a path (Lindgren et al., 2013). Essentially, the entire process is parameterized by some $\omega \in \Omega$, this means that we can pick out one such ω, and obtain the entire process by evaluation of the random variables. This is impractical when working with processes computationally, since single steps cannot necessarily be evaluated at a time. We will look at a class of stochastic processes which have independent

increments. This allows for much easier sampling, since each increment can be sampled independently.

Definition 3.5 (Independent Increments)**.** Given a probability space $(\Omega, \mathcal{F}, P)$ and an index set I. A stochastic process $X = \{X(t), t \in I\}$ has independent increments if and only if for any finite increasing subsequence $(t_i)_{i \leq n} \subseteq I$, the differences.

$$X(t_2) - X(t_1), \ldots, X(t_n) - X(t_{n-1})$$

are independent random variables.

Processes with this property have a series of advantages from both an analysis point, but also in terms of the practicality of dealing with them computationally. Another property that we want is what we call stationary increments, we define it as so;

Definition 3.6 (Stationary Increments)**.** Given a probability space $(\Omega, \mathcal{F}, P)$ and an index set I. A stochastic process $X = \{X(t), t \in I\}$ has stationary increments iff for $s, t \in I$ where $s \leq t$

$$X(t) - X(s) = X(t - s)$$

With all of these new tools, we are now ready to define Brownian motion.

3.1 Brownian motion in $\mathbb{R}^d$

Lets revisit the ant process visualized in figure 2, and lets imagine we have gotten a bit bored of watching the ant constantly, as it only moves once per day. To make it a bit more exciting, we decided that it would be way more interesting if it moved half as far, but twice a day! Thus if we let the index set $I_n = \mathbb{N}_{\leq n}$ we now have $S_n = \sum_{t \in I_n} X(t)/2$. Lets imagine this was a great success, and the experiment got significantly more exciting. We then ask ourselves if we could make it move even more, we thus decide that we can scale this arbitrarily. That is, if X_t for $t \in I$ is a random variable that is 1 if the ant moves right and -1 if it moves left. We then decide to let it move Δx units per Δt time. In the case before we had $\Delta x = \Delta t = 1/2$. We now let $I = \mathbb{N}_0$, and as before have $X_t = \pm 1$ for $t \in I$. For any $\Delta = \Delta x = \Delta t$ we let

$$S^{\Delta}(t) = \sum_{i \leq t/\Delta} \Delta \cdot X_i$$

This way, we can first sample the random increments X_t, and then afterward, investigate how the scaling affects the process. Simulating the follow processes for different Δ is shown on figure 3 This figure illustrates the problem with letting $\Delta x = \Delta t$. We see that

$$E[S^{\Delta}(t)] = \Delta \sum_{i \leq t/\Delta} E[X_i] = 0$$

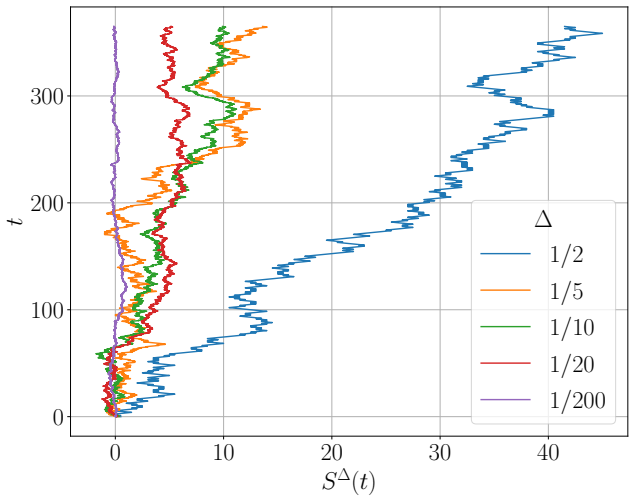

Figure 3: Plot of $S^{\Delta}(t)$

And now

$$\begin{aligned}\mathrm{Var}(S^{\Delta}(t)) &= \Delta^2 \sum_{i \leq t/\Delta} \mathrm{Var}(X_i) \\ &= \Delta^2 \sum_{i \leq t/\Delta} 1 = \Delta^2 \frac{t}{\Delta} = \Delta \cdot t\end{aligned}$$

The problem here is that if we let $\Delta \to 0$, then $\mathrm{Var}(S^{\Delta}(t)) \to 0$, and the process is completely trivial. If, however we do separate the terms, such that we define

$$X(t) = \Delta x \sum X_{t/\Delta t}$$

Then similarly

$$\begin{aligned}E[X(t)] &= 0 \\ \mathrm{Var}(X(t)) &= (\Delta x)^2 \frac{t}{\Delta t}\end{aligned}$$

We see here that if we have

$$\frac{(\Delta x)^2}{\Delta t} = \sigma^2 \Longrightarrow \Delta x = \sigma\sqrt{\Delta t}$$

then suddenly we have control over the variance of our process. By the central limit theorem, we get that in fact $X(t)$ is normally distributed with variance $\sigma^2 t$, if we let the number of steps tend to infinity. This is a highly important property, because we have now found a way to create gaussian random variables (out of seemingly thin air) by simply letting the number of random

steps X_t go to infinity, and Δt go to 0. What we got is what we call Brownian motion. The existence of this limit is not completely trivial, and we refer to Ross (1996) for a proof. We can now come with a rigorous definition;

Definition 3.7 (Brownian Motion)**.** Any stochastic process $X = \{X(t), t \in \mathbb{R}^+\}$ is said to be Brownian motion iff.

1. $X(0) = 0$
2. X has stationary independent increments
3. $X(t)$ is normally distributed with variance $\sigma^2 t$

If X is Brownian motion, then $X(t)$ will be a continuous function $X : I \to \mathbb{R}$ (Ross, 1996). This also makes it clear why we required I to be compact and connected, since there would be a range of topological complications otherwise. Intuitively, X is nowhere differentiable. It is in a sense similar to the weierstrass function, where we can think of it as a fractal, allowing us to zoom infinitely far in on the function. This also makes sense in the ant analogy; if we keep asking the ant to move more often, then at its limit, it's going to move continuously. If we conduct the same experiment as in figure 3, we will get something (figure 4) that is much closer to what we expected when taking the limit of a random walk. There are a series of properties that makes Brownian

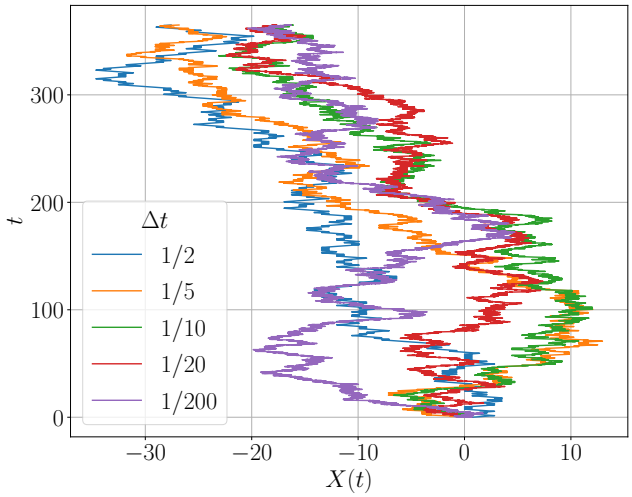

Figure 4: Plot of $X(t)$ for different Δt

motion convenient to work with, the first of which being rather trivial, but central to our later analysis of projections on trees.

Lemma 3.1 (Brownian normalization (Ross, 1996))**.** If $X(t)$ is Brownian motion with variance $\sigma_1^2 t$, then $X'(t)$ has variance $\sigma_2^2 t$ where

$$X'(t) = \frac{\sigma_2}{\sigma_1} X(t)$$

This lemma also gives lead to what we will refer to as standard Brownian motion. We define it as any Brownian motion $X(t)$ with variance t. This is convenient to work with, since any Brownian motion with variance $\sigma^2 t$ can be converted to standard Brownian motion by simply dividing out σ via lemma 3.1. Because of this, we will often omit the σ when talking about Brownian motion, because it usually does not affect computation, but simply creates notational overhead.

A result of the requirement of independent increments, is that given a state $X(c)$, any future state $X(c+t)$ conditioned on $X(c)$ will be independent of all past states $X(p)$ for $p < c$ (Ross, 1996). This is useful in the analysis of phylogenies, as it implies that knowledge of any species in the tree, lets us consider the subtree rooted at that exact species, independently of the rest of the tree.

As we will see more of in section 4, we can model the lineage of a certain species, as a Brownian motion process, with m branching points $t_1, ..., t_m$. Where $f(x)$ is the standard normal density

$$f_t(x) = \frac{1}{\sqrt{2\pi t}} \exp\left(-x^2/2t\right)$$

We obtain the joint density for a sequence of m branching points $X(t_1), .., X(t_m)$ from Ross (1996):

$$f(X(t_1), ..., X(t_m)) = \\ f_{t_1}(X(t_1)) f_{t_2-t_1}(X(t_2) - X(t_1)) \dots f_{t_m - t_{m-1}}(X(t_m) - X(t_{m-1}))$$

This is actually the definition of a gaussian process, which means that the following also holds for Brownian motion.

Lemma 3.2 (Covariance of Brownian pairs)**.** Where $X(t)$ is Brownian, the covariance of $X(u)$ and $X(v)$ with $u < v$ is given as

$$\mathrm{Cov}(X(u), X(v)) = u$$

This lemma is proven by simply using the independent increments property. This is however important since for any two species, it implies that the covariance of two species u and v which have an ancestral relationship, will have a covariance that is proportional to the time of the ancestor. It also implies (although not as directly), that two different species with a common ancestor, have a covariance that is equal to the time of their first common ancestor.

3.2 Brownian motion on a manifold

In this section, we will show that Brownian motion arises as a solution to certain differential equations, defined on the surface of a manifold. The problem

here is construction. Claiming the process that solves a certain PDE is the one we are interested in, does not necessarily give a feasible way of realizing such motion.

The differential equation we are interested in, is the heat equation. It turns out, that if we have a single point of heat $\mu \in \mathbb{R}^d$, then the resulting dispersion of heat after one unit of time, will be the density of the normal distribution centered at μ. We write this as a differential equation where $p(t, y)$ is the heat in point y, at time t. If the heat at time 0 is all centered at a single point μ, then $p(0) = \delta_\mu$, with δ_μ being the point indicator function which is 0 everywhere except $\delta_\mu(\mu) = 1$. The heat equation is then written as

$$\partial_t p(t, y) = \frac{1}{2}\Delta p(t, y) \tag{1}$$

Where $\Delta p(t, y)$ denotes the laplacian of the function $p(t, y)$ which is the sum of double derivatives with respect to each coordinate in y. By Pennec et al. (2020), with solution p, the density $p(1)$ is the density of $\mathcal{N}(0, \mathrm{Id})$. Now, due to the connection between Brownian motion and the normal distribution, we get that for any Brownian motion $X(t)$, the density of a sample endpoint at time t is given as a solution to equation (1) directly as $p(t)$.

To give an intuition of Brownian motion on a riemannian manifold, we can use the generalization of the Laplace operator, which we call the Laplace-Beltrami operator Δ_g, where g is the riemannian metric. The heat equation is now similar, but instead specified on the surface of a manifold $\mathcal{M}$

$$\partial_t p(t, y) = \frac{1}{2}\Delta_g p(t, y) \tag{2}$$

This way, we can think of Δ_g as adapting the transition density to work on curved geometries. This gives us a concrete way of writing isotropic gaussians on the surface of manifolds, but it does not yet allow us to write up general form gaussians with covariance matrices Σ.

Another problem with this formulation arises. We cannot guarantee that the solution to this equation is proper however, by Hsu (2008) it may happen that

$$\int_{\mathcal{M}} p(t, y)\, \mathrm{d}y < 1 \tag{3}$$

The consequence is that Brownian motion may not run forever, but will have a stopping time e. As Hsu (2008) notes, this can only happen if $\mathcal{M}$ is not compact, and gives rise to the notion of stochastically complete manifolds. We will assume that all manifolds in question are compact, and thus we do not run into this issue.

When thinking of variables drawn from anisotropic gaussian densities $\mathcal{N}(\mu, \Sigma)$ in $\mathbb{R}^d$, we can describe them as a linear combination of isotropic gaussians via

the standard euclidean linear latent model(Pennec et al., 2020). We get that if

$$y = \mu + Wx + \epsilon \tag{4}$$

Where W is a $d \times r$ matrix, $x \sim \mathcal{N}(0, \mathrm{Id}_r)$ is what we refer to as the latent variable and $\epsilon \sim \mathcal{N}(0, \sigma^2 \mathrm{Id}_d)$ is some isotropic noise. Then $y \sim \mathcal{N}(\mu, \Sigma)$ with $\Sigma = WW^\top + \sigma^2 \mathrm{Id}_d$. We have just seen previously how to generalize isotropic gaussians to a manifold setting, however summing them is not defined, nor is the transformation Wx, as there is no global coordinate system.

The next step becomes slightly more involved depending on which type of manifold we are dealing with. We are now looking for a way to make sense of the sum of two gaussian variables, aswell as the matrix product Wx. If it is a Lie group, as in the case of $\mathrm{Diff}(\Omega)$, construction becomes easier. We will first show how one such construction may look.

Unlike section 2.4 where we looked at constructing a right invariant metric, we now assume we have some left-invariant metric on a Lie group G. Recall that a Lie group is simply a manifold with additional group structure, and where the tangent space around the identity element is a special vector space which we call the lie algebra $\mathfrak{g}$. Since $\mathfrak{g}$ is a tangent space, it will have some basis $e_1, \ldots, e_d$.

We want some sort of coordinate system such that for any point $y \in G$, we can view local neighborhoods in a way that aligns with $\mathfrak{g}$. We do this just like in section 2.4, but this time use a left-invariant push-forward $(L_y)_*$. This push-forward translates vectors of $\mathfrak{g}$ to T_yG. With this, we can for each basis element e_i of $\mathfrak{g}$ obtain a corresponding vector $(L_y)_*(e_i)$, which with parameterization of y becomes a vector field on G given as $X_i(y) = (L_y)_*(e_i)$. Furthermore, for any y, all vectors $X_i(y)$ will be orthogonal, and thus follow what we think of as a consistent basis for the tangent space.

The beauty of this construction is that we now completely avoid the question of finding a generalization of (4), as for a d dimensional lie algebra G, we can simply perform standard Brownian motion in $\mathbb{R}^d$, and obtain a corresponding Brownian motion in G, via the vector fields X_i. The process $y(t)$ arise as a solution to the Stratonovich equation, which is expressed only in terms of our vector fields X_i, the euclidean Brownian motion $B(t)$, and naturally the manifold Brownian motion $y(t)$.

As mentioned previously, this only works in the case of $\mathcal{M}$ being a Lie group. A case where this is not possible, is the sphere $\mathbb{S}^2$. One can consider a single vector field $X_i(y)$. It cannot be continuous by the hairy ball theorem, which has a quite simple visualization. Imagine each tangent vector $X_i(y)$ as a single hair on $\mathbb{S}^2$, it is impossible to comb the hair of the ball such that there is no points where two hairs suddenly point in different directions. This makes the construction simply impossible for this case.

The plan for writing up Brownian motion on manifolds is to reformulate (4), in a manifold setting - something we were happy to avoid on Lie groups. We cannot simply perform Brownian motion in the tangent space of μ, since the tangent space only acts as a first order approximation of the manifold surface. A better approach, is to define (4) in terms of infinitesimal changes on the manifold surface. We get the following manifold generalization

$$\mathrm{d}y(t) = W\mathrm{d}x(t) + \mathrm{d}\epsilon(t) \tag{5}$$

Where $x(t)$ and $\epsilon(t)$ is euclidean Brownian motion. We also enforce that the process starts in μ. Writing (4) as a differential equation, really only helps with the problem of summing up random variables. We still have an immediate problem of how to deal with the term $W\mathrm{d}x(t)$. The problem is that unlike Lie groups, there is no obvious way of transporting vectors in $T_\mu\mathcal{M}$ to some other point $T_y\mathcal{M}$. This is problematic because the matrix W is defined with respect to the basis of $T_\mu\mathcal{M}$.

What we seek is some sort of transport that lets us take vectors in one tangent space, and transport them to another. There is the notion of a parallel transport which takes vectors of one tangent space, and transports them along a curve. One could imagine transporting a basis of one tangent space to another by selecting a curve that goes through the point of interest. This has another problem, being that the resulting vector is path dependent. This would give differing W's depending on realization. There is a way around this problem called the Eells–Elworthy–Malliavin construction, however it requires concepts that are out of scope of this project.

4 Phylogenetic trees and random Sampling

In this section, we look at phylogenetic trees where we model the evolutionary developments of certain species or sub-species. Evolution has been known scientifically since the time of Darwin, and to analyse the evolution of groups of animals, we need a framework to describe this ever-going process. As noted by Harmon (2019), there are many factors that affect evolution. Both environmental factors such the availability of certain food sources, and random changes in a species DNA can play a deciding role. We aim to construct a model for this process that explains observations to some degree. Due to how hard it is to quantify environmental factors, our models will view evolution as a process separate from the environment.

It is under these processes that change in the biology of species occur, and where some species go extinct, others grow and evolve. Extinction and early termination is another topic that can be hard to cover in a model. We will not consider these cases, and instead simply look at the lineage of a series of species. Modeling the behaviour of evolution is important to this project.

We are essentially looking for 2 components; First, we want to model the "random" drift in one species over time, as a random process where certain traits has the same probability of changing in two same-sized time intervals. In reality a combination of genetic and environmental factors take part in this process. Modeling on a genome directly is a very hard task, and even modeling a discrete set of genomes or features does not lend itself well in a PCA setting. There are models such as the MK model (Lewis, 2001), which allow one to model discrete traits, but for our project we do not consider these traits.

Throughout this project, we will think of phylogenetic trees as a tree graph $G = (V, E)$ where the edge set E has the pair (u, v) if v is a direct descendant from u. The exact contents of V, will differ within different contexts, but this section aims to give a rigid definition of phylogenetic trees and their sampling. We can think of any $u \in V$ to represent the taxon (species) u. The exact representation of u depends on the setting, in this project we will often assume that u is represented as a d dimensional trait vector which contain values for continuous traits. In the case of lizard for example, one could imagine tail length, weight, lifespan to be a set of features we could be interested in modeling. The other setting which will appear commonly, is where u is a point of some riemannian manifold $\mathcal{M}$. For now, we will assume $u \in \mathbb{R}^d$.

4.1 Distributions of phylogenies

Most species itself follow some distribution. We generally assume that within one specific population, there is a degree of randomness to the exact trait values of any new child. This would lend itself to modeling each species as a time-dependent family of distributions. However, because representing species as distributions leads to a lot of overhead in calculations, we assume that any $u \in V$, is the mean of all traits in the population it represents. The upside of this is the great simplicity in our model since each species is a simple point estimate. Suddenly each species which represent many individuals, is collapsed into a single vector, which makes both inference and sampling easier.

The notation around these taxa may be a bit inconsistent, since at times taxa are assumed unknown. Unless otherwise stated, we assume that the leaves are all known and measured quantities, and the inner nodes in the phylogeny are unknowns. It should be noted that there exists algorithms for estimating the ancestral states (inner nodes) for a variety of different models, thus it is not unrealistic to assume ancestral states can be inferred.

As laid out so far, we are interested in giving a point estimate for the mean of all d trait values in a species, at any given time t. In other words for some species X, we are looking for a function $X : \mathbb{R}^+ \to \mathbb{R}^d$, such that $X(t')$ gives us the mean trait values for the species X at time t'. This should ring some bell, as the wished for properties are some of the properties of brownian motion.

To simplify the model, let us first consider modeling a single trait, in other words, letting $d = 1$.

When we say that we assume traits evolve under a brownian motion model, we simply mean that for a single trait X, it can be described as a brownian motion process $X(t)$ which starts at time 0, and has some scaling parameter σ. As noted earlier, we often omit σ, as we in classical PCA can assume a standardization of the data, which in our case is removing the scaling parameters. This means, that when looking at a single trait for a single species at current time t_0, it will by definition 3.7 have the distribution $X(t_0) \sim \mathcal{N}(0, t_0)$.

This model conveniently lets us approach traits with the assumption that all increments are normally distributed from the origin. The problem in these models arise when we consider two or more different species that evolve from some common ancestor. Lets now consider the simple example of figure 5. In this figure, x_1 and x_2 represent two different species, that have a common

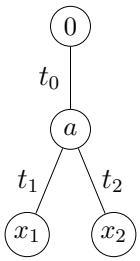

Figure 5: A simple phylogeny

ancestor a. A way to think of this is that we have $X_1(t), X_2(t)$ as two brownian motion process with $X_1(t_0+t_1) = x_1$ and likewise for x_2. Furthermore we have $X_1(t) = X_2(t)$ for $t \leq t_0$, which is equivalent to saying that the two species had identical traits before and at t_0. In reality what we really mean is not only that they had identical traits, but that they were a common species before t_0.

Because of the independent increments property, and the fact that both t_0, t_1 and t_0, t_2 are finite increasing subsequences of $\mathbb{R}^+$, we can use that $a, x_1 - a$ are independent and $a, x_2 - a$ are independent. Letting $\Delta_1 = x_1 - a$ and $\Delta_2 = x_2 - a$ we get that we can write x_1, x_2 as sums of their ancestors

$$\begin{aligned} x_1 &= a + \Delta_1 \\ x_2 &= a + \Delta_2 \end{aligned}$$

this then means that if we evaluate $\mathrm{Cov}(x_1, x_2)$ we get

$$\begin{aligned} \mathrm{Cov}(x_1, x_2) &= \mathrm{Cov}(a + \Delta_1, a + \Delta_2) \\ &= \mathrm{Cov}(a, a) + \underbrace{\mathrm{Cov}(a, \Delta_2) + \mathrm{Cov}(\Delta_1, a) + \mathrm{Cov}(\Delta_1, \Delta_2)}_{0} \\ &= \mathrm{Var}(a) = t_0 \end{aligned}$$

Where the first two 0 terms are 0 due to the independent increments property, and the last term is 0 because the increments are sampled in two independent processes. Every two species x_1, x_2 will be in an equivalent phylogeny to figure 5 if you remove all other species, thus the above implies that the covariance of any two species is their time of common ancestry, which in this example is t_0.

As described by Harmon (2019), we could also consider x_1, x_2 as being drawn as a single sample from a normal distribution. After all, we know their covariance, and we know that they are both normally distributed. Thus, we can describe the leaves x_1, x_2 as a single $x' = (x_1, x_2)^\top$, drawn from a multivariate distribution with covariance matrix C, where

$$C = \begin{bmatrix} \text{Cov}(x_1, x_1) & \text{Cov}(x_1, x_2) \\ \text{Cov}(x_2, x_1) & \text{Cov}(x_2, x_2) \end{bmatrix} = \begin{bmatrix} t_0 + t_1 & t_0 \\ t_0 & t_0 + t_2 \end{bmatrix}$$

Since the matrix C represents the entire covariance in the phylogeny, we usually refer to this as the phylogenetic covariance matrix, or just the phylogenetic matrix.

Now, given a phylogeny, we can simply calculate its covariance matrix C, and randomly draw a trait value for each leaf, by drawing from the multivariate normal given by

$$p(x) = \frac{\exp(-\frac{1}{2} x C^{-1} x^\top)}{\sqrt{(2\pi)^n \det(C)}}$$

It is very important to note that this model says nothing about the increments Δ_i, nor the internal states such as the ancestor a in our example. It only uses the phylogeny to parameterize a multivariate normal via the covariance matrix implied by the phylogeny.

Now, one could hope that we could simply do this for each of the d traits that we are modelling, and we would have something useful. This is however not the case, since that would require each trait to be independent. In reality, however, two traits such as body mass and height, will most likely be positively correlated.

Just like we saw the definition of a gaussian process in a single variable brownian motion, it is also possible to construct a multivariate brownian motion, but where we now simply require that each difference between branching points corresponds to a vector that is normally distributed with multivariate normal $\mathcal{N}(0, t\Sigma)$ where t is the increment of the index between the branching points. On a very high level, we can imagine replacing the univariate normal distributions inherent to brownian motion with a multivariate distribution with covariance matrix Σ.

The situation now gets slightly more complicated since we both have covariance between traits due to being traits on two species with some common ancestry, but also due to the traits simply having an underlying correlation.

If we extend the notion that multivariate brownian is simply the univariate brownian process with a multivariate guassian at its core, then we can as before consider one realization of a phylogeny as a single vector $x' \in \mathbb{R}^{nd}$. In this vector, the first n entries are the values of the first trait of each leaf, the next n, values of the second trait and so on. It is not too hard to see that specifying a gaussian distribution over this vector is exactly equivalent to specifying a gaussian distribution over $n \times d$ matrices, since they are equivalent under what is commonly referred to as the invertible Vec map, which takes any $n \times d$ matrix and produces an nd vector which is the stack of the columns into one vector.

If we look at it in this form, and say we want to find the distribution of some matrix $X \in \mathbb{R}^{n \times d}$ where each row is a leaf taxa, we would simply find a distribution over $\text{Vec}(X)$ as is done by Barratt (2018), where $\text{Vec}(X)$ denoets the concatenation of columns of X into one long column vector. We are going to construct our distribution with $V \in \mathbb{R}^{nd \times nd}$ as our covariance matrix, and μ as the Vec of the mean matrix. Since phylogenies do not affect the mean of any trait, we are going to have identical values for the first d entries, the next d and so on. To simplify calculations we construct a design matrix where entry i, j is 1 if $j = \lfloor \frac{i}{n} \rfloor$ and 0 otherwise. We refer to this matrix as the design matrix and it looks like a simple partitioned matrix where the first column contains n 1's followed by $n(d-1)$ 0's. In general column i contains $n(i-1)$ 0's followed by n 1's, and again followed by 0's

$$D = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{nd \times d}$$

With this we can obtain the vectorization of the mean matrix as a matrix product of the trait mean vector a.

$$\mu = Da$$

This essentially spreads the elements of a out onto each corresponding trait in the matrix vector form and does give us a correct mean matrix vector. While Revell and Harmon (2008) does not formally motivate their chosen construction of the covariance matrix V, we will try and motivate it via the definitions of Barratt (2018).

What we are given initially is the phylogenetic covariance matrix C which describes the covariance of the individual taxa. Also, we are given Σ (also denoted R in a wide range of literature), which holds the covariances between the individual traits. The question is now, given C, Σ, what is the covariance of entries in $\text{Vec}(X)$. We wish to show in agreement with Revell and Harmon (2008) that it is in fact $V = \Sigma \otimes C$ where $\otimes$ is the Kronecker product, defined

as a direct block matrix product

$$A \otimes B = \begin{bmatrix} a_{1,1}B & \dots & a_{1,m}B \\ \vdots & \ddots & \vdots \\ a_{n,1}B & \dots & a_{n,m}B \end{bmatrix}$$

and perhaps shine a light on where exactly this comes from.

As we would expect, Waal (2006) defines the $nd \times nd$ covariance matrix V as

$$V = \mathbf{E}[(\text{Vec}(X) - \mu)(\text{Vec}(X) - \mu)^\top]$$

By Waal (2006), if we can decompose V into the Kronecker product of two positive definite symmetric matrices $\Psi = (\psi_{ij}), \Sigma = (\sigma_{ij})$ such that

$$V = \Sigma \otimes \Psi$$

then the covariance of columns i and j of X, is given by $\sigma_{ij}\Psi$ and the covariance of rows i and j of X is given by $\psi_{ij}\Sigma$[1].

To show this is applicable to our case, we want to show that

$$V = \Sigma \otimes C$$

Where C is the phylogenetic matrix and Σ is the trait covariance matrix. They are both trivially p.d.s. so intuitively the covariance of column i and j of X must be $\sigma_{ij}C$ and that the covariance of row i and j must be $c_{ij}\Sigma$.

This comes from two realizations. When using a single trait model, the covariance is simply σC where σ is a scaling parameter for the trait. If we have a single taxon that is obtained via brownian motion then the covariance is $t\Sigma$ where t is the time of realization, which under a phylogeny would be C_{ii} given our taxon was taxon i.

If we consider the first simple example of a phylogeny in figure 5, but this time consider $x_1, x_2 \in \mathbb{R}^d$ as being multivariate with trait covariance matrix Σ, then their covariance must be $c_{12}\Sigma$. This follows a similar line of argument to the first analysis of figure 5, however we now have a multivariate case. Again we assume $x_1 = a + \Delta_1$ and $x_2 = a + \Delta_2$ but where a and Δ_1, Δ_2 are all drawn from multivariate distributions with covariance matrices that are scalar multiples of each other. We get these distributions from the independent increments property.

$$a \sim \mathcal{N}(0, t_0\Sigma) \qquad \Delta_1 \sim \mathcal{N}(0, t_1\Sigma) \qquad \Delta_2 \sim \mathcal{N}(0, t_2\Sigma)$$

Previously we used that covariance is additive $\text{Cov}(X + Y, Z) = \text{Cov}(X, Z) + \text{Cov}(Y, Z)$. This also holds for random vectors, but where $\text{Cov}(X, Y)$ is the

[1]The rows and columns are swapped compared to Waal (2006), due to their vectorization working on rows instead of columns

covariance matrix between traits of X and Y

$$\begin{aligned}\operatorname{Cov}(x_1, x_2) &= \operatorname{Cov}(a + \Delta_1, a + \Delta_2) \\ &= \operatorname{Cov}(a, a) + \underbrace{\operatorname{Cov}(a, \Delta_2) + \operatorname{Cov}(\Delta_1, a) + \operatorname{Cov}(\Delta_1, \Delta_2)}_{0} \\ &= \mathbf{E}[aa^\top] - \mathbf{E}[a]\mathbf{E}[a]^\top = t_0\Sigma = c_{12}\Sigma\end{aligned}$$

Since t_0 is defined to be c_{12} and the expected covariance under a multivariate distribution is the covariance matrix of the distribution. This shows that the rows of X have covariance $c_{ij}\Sigma$.

Consider now the covariance of any two traits in X. Each trait belongs to a row (taxon), let those rows be X_i, X_j. From the above we know the covariance matrix of those rows are given as $c_{ij}\Sigma$. If each of the two traits have indices r, k in their rows respectively, then position r, k in this covariance matrix must be their covariance. This implies that the covariance of any two traits X_{ir}, X_{jk} is given as $c_{ij}\sigma_{rk}$.

If we now consider the covariance matrix $\operatorname{Cov}(\operatorname{Vec}(X), \operatorname{Vec}(X))$, we can transform any index k in $\operatorname{Vec}(X)$ into an index i, j in X via $i = k \mod n$ and $j = \lfloor \frac{k}{n} \rfloor$. Substituting this in, we obtain

$$\begin{aligned}\operatorname{Cov}(\operatorname{Vec}(X), \operatorname{Vec}(X))_{k\ell} &= \operatorname{Cov}(X_{k\%n, \lfloor \frac{k}{n} \rfloor}, X_{\ell\%n, \lfloor \frac{\ell}{n} \rfloor}) \\ &= c_{k\%n, \ell\%n}\sigma_{\lfloor \frac{k}{n} \rfloor, \lfloor \frac{\ell}{n} \rfloor}\end{aligned}$$

Where we use the shorthand notation $k \mod n = k\%n$ to salvage some readability. Although this is quite hard to read and understand, some meaning is there. Since both indices of c are some indices modulo n, the c term in the covariance matrix of $\operatorname{Vec}(X)$ cycles over the entries of C. Similarly, since both indices in σ are integer divisions, the same term of σ is repeated n entries. All in all this gives us the following block matrix

$$\operatorname{Cov}(\operatorname{Vec}(X), \operatorname{Vec}(X)) = \begin{bmatrix} \sigma_{1,1}C & \dots & \sigma_{1,d}C \\ \vdots & \ddots & \vdots \\ \sigma_{1,d}C & \dots & \sigma_{d,d}C \end{bmatrix}$$

Which is identical to the Kronecker product $\Sigma \otimes C$. Notice, by the earlier observation by Waal (2006), this also implies that in fact, the covariance between any two columns of X, is $\sigma_{ij}C$, something that would have been much harder to prove. Waal (2006) now gives a closed form density for a matrix normal on this form, where[2] $V = \Sigma \otimes C$ and $M = \mathbf{E}[X]$.

$$\begin{aligned}f(X) &= \frac{\exp(-\frac{1}{2}\operatorname{tr}(C^{-1}(X - M)\Sigma^{-1}(X - M)^\top))}{\sqrt{(2\pi)^{nd}\det(C)^n\det(\Sigma)^d}} \\ &= \frac{\exp(-\frac{1}{2}\operatorname{Vec}(X - M)^\top V^{-1}\operatorname{Vec}(x - M)))}{\sqrt{(2\pi)^{nd}\det(V)}}\end{aligned}$$

[2] Σ and C are swapped because of the different vectorization again.

This gives a closed form distribution over any multivariate phylogeny, which can bed used both for inference, and for sampling.

Phylogenies on a manifold work essentially the same way. We simply replace the underlying transition density of our brownian motion models, with a model defined over the surface of some manifold we are interested in. This is essentially what makes this model so powerful, we can without too much extra effort, apply techniques to a manifold setting, by simply characterizing everything as Gaussians and by then using brownian motion.

4.2 Sampling

We have now seen that phylogenetic trees can be thought of as realizations of brownian motion, as a series of increments in a tree structure, and as a matrix distribution. The reasons for these different interpretations, are primarily related to

1. How easy their density is to write up
2. How easy they are to sample
3. How much information they carry

In the case of matrix distributions, they provide a convenient closed form density, which we can use for inference about Σ for example. Their downside is that when sampling them directly, you only obtain a single matrix X that contains data for the leaves. This is somewhat inconvenient if the aim is to perform any kind of experiment on the internal nodes of the phylogeny. For this reason, the closed form densities will mostly be used to perform MLE of different parameters, given we have observed all leaf nodes. Alternatively the distribution of nodes in a subtree at any root is also given as a matrix gaussian, and as such can be used in subtrees aswell.

To conduct some of our comparison experiments we require full trees, where we have information about all internal states in the tree. To do so, there are two steps; we must first sample the phylogeny itself, which contains information about the duration of each generation, as well as where any species splits into two. Next, we sample the increments over each edge, from a simple multivariate gaussian with covariance $t\Sigma$.

To simplify notation, for any two nodes u, v where u_t, v_t denotes their time of realization respectively, we define e_t for any edge $e = (u, v)$ to be $e_t = v_t - u_t$. We also define the increment over any edge $\Delta_e = v - u$ where by the independent increments property we have $\Delta_e \sim \mathcal{N}(0, e_t\Sigma)$. With this in place, we can think of our sampling strategy as first sampling the edge set E, and then sampling Δ_e for each edge in E. As an edge is a element of $V \times V$ where

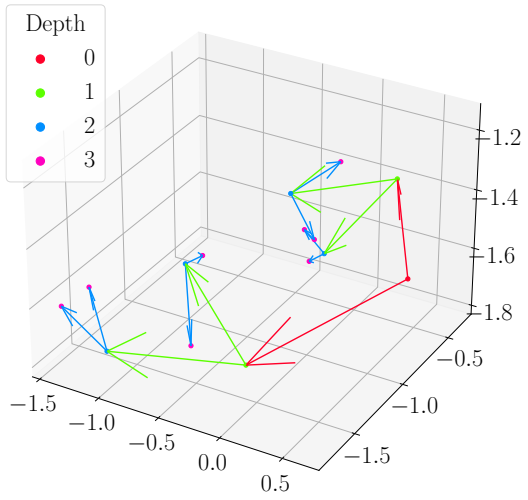

Figure 6: Sample of 3 trait phylogeny

V is the set of nodes in the phylogeny, we must first determine V before we can sample E. Our strategy is to do all of this in a recursive fashion.

Since we cannot assume real world data to be centered, we initially sample a root $r \sim \mathcal{N}(0, t'\Sigma)$ where t' has some distribution $\mathcal{D}$. We then sample b time steps t_i from $\mathcal{D}$, but multiply these timesteps with an exponential decay factor $\beta = \exp(-kd)$ where d is the depth of the node, and k is some scaling constant. This gives us b edge times, that will be the times on the b edges emanating from r. We can then sample each child v of r by sampling the increment of the edge $e = (r, v)$ from $\Delta_e \sim \mathcal{N}(0, \beta t_i \Sigma)$. Each of the b children of r can now be made roots of their own subtree, and this procedure recurses down with $d + 1$ as the new depth until some specified max depth. In practice we used either $\mathcal{D}$ as a uniform, or a log-uniform distribution.

Sampling a simple tree with this procedure with $d = b = 3$ and $\Sigma = I_3$, we get isotropically independently sampled traits with results visualized in figure 6. Each color represents the depth of the sampling, where a depth 0 means it is the first split in the tree.

These samples show the kind of data we wish to perform PCA on. Whilst the current representation is only 3 dimensional, it is not uncommon that the taxa are very high dimensional vectors. Projecting these trees into a lower dimensional space, may give intuition about the geometry of the phylogeny.

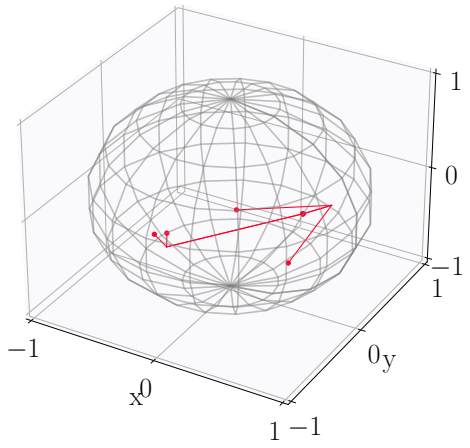

Figure 7: Sample of phylogeny with manifold valued traits

The sampling strategy we employ for manifolds, is almost identical to the the euclidean setting. We again simply let brownian motion run for βt_i time, and restart b new processes in the endpoint of the first process. An example on $\mathbb{S}^2$ can be seen in figure 7.

5 Principal Component analysis and its variants

In this section we will introduce a few variants of Principal Component Analysis (PCA), each adapted to a specific domain. Initially we will introduce the classic PCA on vectors in $\mathbb{R}^d$.

5.1 Principal Component analysis

Given n vectors $x_i \in \mathbb{R}^d$, it is often useful to project these vectors to a linear subspace of low dimensionality, such as $\mathbb{R}^2$. This allows for a visual inspection of the data, and is frequently used as a way to visualize high dimensional data. Projecting high dimensional data into a very low dimensional subspace is not trivial however, inevitably some variance amongst the original vectors x_i, is going to be lost after projection.

The simple idea behind PCA, is to first find a 1-dimensional linear subspace $P \subseteq \mathbb{R}^d$ with basis $\{\lambda_1\}$, where the projection of all x_i's into P yields the maximum variance. This process is then repeated, except we now subtract each x_i's projection onto P from x_i. We repeat this until the basis of P spans $\mathbb{R}^d$. We now have a set of basis vectors λ_i, usually referred to as principal components, where any subspace $S_m = \text{span}\{\lambda_i \mid i = 1...m\}$, will give a projection space that preserves a maximal amount of variance among all x_i's. We can then use PCA as a dimensionality reduction algorithm, mapping points in $\mathbb{R}^d$ into $\mathbb{R}^m$ by simply projecting any point x_i into S_m.

In practice this repeated process of selecting principal components is ineffective, and substituted by a decomposition of the data covariance matrix Σ. We define $\Sigma_{ij} = \text{Cov}(x_i, x_j)$, and note that the eigenvectors of Σ will form a basis that maximizes surplus variance along each axis (Abu-Mostafa et al., 2012).

This is convenient because Σ can be calculated by simply centering and transposing the data matrix and multiplying by itself, centered; $\Sigma = \frac{1}{n-1}(X - \bar{X})^\top(X - \bar{X})$. In practice this allows us to to compute all eigenvectors of Σ, and then output S_k by simply computing the span of the first k eigenvectors. The amount of variance explained by projection onto each eigen vector is its corresponding eigenvalue. An alternative and numerically more stable approach is to compute the singular value decomposition (SVD) of the centered data matrix $X - \bar{X}$. This decomposition decomposes $X - \bar{X}$ into three matrices

$$X - \bar{X} = U\Gamma V^\top \tag{6}$$

Where $U \in \mathbb{R}^{n\times d}$ has orthonormal columns, $V \in \mathbb{R}^{d\times d}$ is orthogonal and Γ is a non-negative diagonal matrix (Abu-Mostafa et al., 2012). A theorem due to Eckart and Young (1936) shows that the first k column vectors of V form a set of k principal directions, each with an explained variance of γ_i^2, where $\gamma_i = \Gamma_{ii}$, since the singular values in Γ are exactly the square roots of the eigenvalues of Σ.

If we know that our data is gaussian, that is $x_i \sim \mathcal{N}(\mu, \Sigma)$, then by definition Σ is the covariance matrix, and the problem of PCA can be reframed into a parameter estimation of Σ. This will be the approach of the following sections, since we deal with purely gaussian data.

5.2 Phylogenetic Principal Component Analysis

In cases of highly correlated data, PCA may not give results as expected. Consider a simple case of a 3-point dataset x_1, x_1, x_2. If we were to perform PCA on this simple dataset, the mean would weighed favourably in the direction of x_1, which would give an underestimation of variance in that exact direction. This is the case in phylogenetic PCA. Phylogenetic trees model the variation of species, which come from common ancestry. Under a brownian motion

model, the expected variance between two species from common descent, will be proportional to the "age" of their first common ancestor. Specifically we let $x_1, ..., x_n \in \mathcal{M}$ each represent a representation of some observed species. We refer to these as taxa, and could be any representation, such as procrustes aligned landmarks in Polly et al. (2013). Furthermore we refer to each feature in any x_i as a trait. In its original form, phylogenetic PCA is only defined for $\mathcal{M} = \mathbb{R}^d$, which we will generalize in later sections.

Our constraints on phylogenetic trees will vary slightly from the original works of Polly et al. (2013). We allow for each taxa to be sampled at a different timestep, and whilst the original works do not strictly disallow this case, they do not explicitly consider it. An example is seen in Figure 8. For our use, we

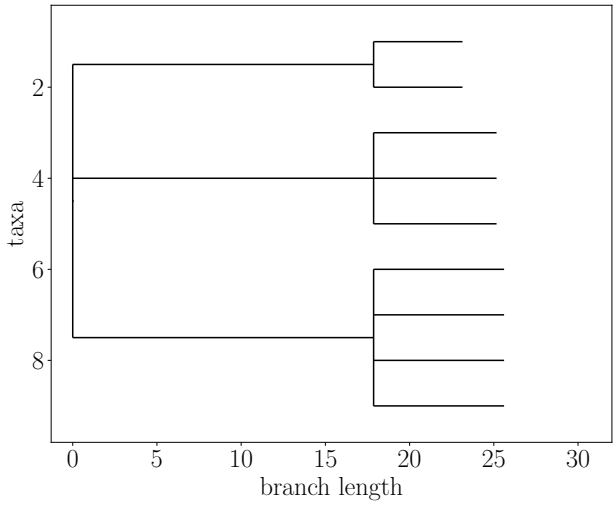

Figure 8: Example phylogeny with different taxa sample times

only use phylogenetic trees to infer covariance. We define a phylogenetic tree as a directed graph $G = (V, E)$ where $V = \mathcal{T} \cup \mathcal{A}$, where Where $\mathcal{T}$ is the set of sampled taxa, and $\mathcal{A}$ is the ancestor set, similarly to section 4.

In the case of a phylogeny with true mean $\bar{x}$, a standard PCA will skew the variance in favor of those taxa that share little branch length with others. This is the exact case of the simple 3-point dataset. Whilst discounting this case is impossible in the original works of Polly et al. (2013), because C is singular, we show a similar example where their method does work.

We will now show a simple calculated example of the effect we are describing. We will slightly modify the 3 point dataset and illustrate phylogenetic PCA

with this example. Consider a 3 point data set $x_1, x_2, x_3 \in \mathcal{T}$ with

$$C = \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

In phylogenetic PCA we are interested in finding a *true* mean of the data, where we take into consideration the covariance between taxa. Intuitively we should somehow discount the contributions of x_2 from x_1, a simple way of doing this is to set $x_1' = x_1 - 0.9x_2 - 0x_3$, and $x_2' = x_2 - 0.9x_1 - 0x_3$ however we quickly see that a mean of these variables have scaling issues, since $x_1' + x_2' = 0.1x_1 + 0.1x_2$, so a sensible answer might be how much should we scale these? One answer is to instead consider this as a system of linear equations, and take the mean of the columns of $C^{-1}X$. This will give us the case before where we have $x_1' + x_2'$, and will scale it by $\mathrm{Det}(C)^{-1}$. The interpretation of this solution, is that C specifies a linear combination of some uncorrelated variables $z_1, ..., z_n$, such that

$$X = CZ$$

Polly et al. (2013) argues that the mean of the columns in Z is an estimation of the trait values at the root. The reason it takes this form, is because X simply arises from a latent space model such as equation (4). From this point of view, the matrix C simply transforms some latent variables z_i, in a way that is specified by the phylogeny.

This only holds if each x_i is a single trait, however if we consider multiple traits, the same reasoning applies, but this time in the form of a matrix distribution. We already saw in section 4, how matrix distributions could be described by two covariance matrices, this is exactly the case here. The strength of this interpretation is that we can find an exact solution, by simply performing a maximum likelihood over the distribution of X. In other words, we are looking for an a, such that $p(X; a, \Sigma)$ is maximized. By (Polly et al., 2013) if 1 is a $n \times 1$ matrix of 1's, we have the ancestral values as a solution to the maximum likelihood problem given as:

$$a = (1^\top C^{-1} 1)^{-1} 1^\top C^{-1} X \tag{7}$$

Notice that the first term $(1^\top C^{-1} 1)^{-1}$ is simply a normalizing constant, which is a simple sum over the entries of C^{-1}. This is at the core of phylogenetic PCA and we really emphasize that the MLE in (7) is done over the entire tree in an instant. This does provide a point estimate for the true root, without any additional knowledge of the internal nodes in the tree, which have been marginalized out.

As for normal PCA, we still need to calculate the covariance matrix, which for phylogenetic PCA we will denote Σ_P. There is an obvious ambiguity here since matrix distributions is a kronecker product between two covariance

matrices, but since we know C upfront, we are interested in estimating the trait covariance matrix Σ.

The solution is again to perform a maximum likelihood estimate by maximizing $p(X; a, \Sigma)$. In line with how we found the ancestral traits, the result will again use C^{-1} to down weigh certain taxa. Again, we can think of the solution as intuitively discounting some of the covariance observed, as it is due to common descent, rather than independent variation. The closed form estimate is given as follows:

$$\Sigma_P = \frac{1}{n-1} X^\top C^{-1} X \tag{8}$$

This calculation again comes as a maximum likelihood estimate of the first term in the kronecker product $\Sigma \otimes C$. Phylogenetic PCA now follows in the same manner as before, by performing SVD on Σ_P. Because we want to perform PCA over the covariance of the features, which as described in section 4 is Σ.

The resulting principal components are now each in the trait space so to speak. We can forget that they had any phylogenetic covariation when projecting, and just project them as if they were all independent. The resulting projection will an identical phylogeny, but where the reduction has happened over the column space of the matrix distribution.

A simple example of the algorithm applied to a randomly sampled phylogeny in $\mathbb{R}^3$ can be seen in figure 9. In this experiment, $\Sigma = \mathrm{Id}_3$, we also run the

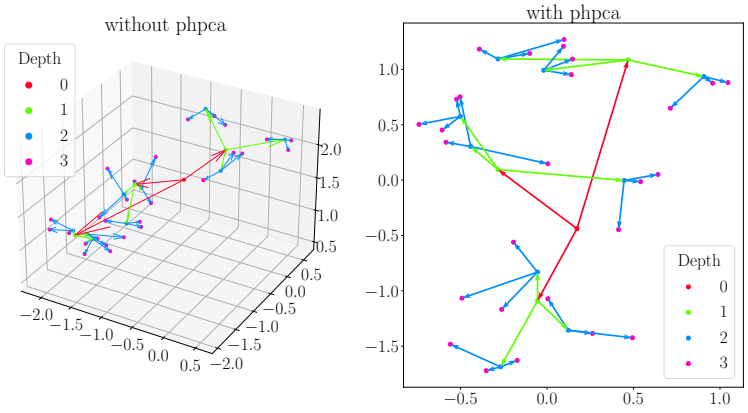

Figure 9: Result of phylogenetic PCA

experiment with

$$\Sigma = \begin{bmatrix} 1 & 0.9 & 0 \\ 0.9 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

And get the following result of figure 10

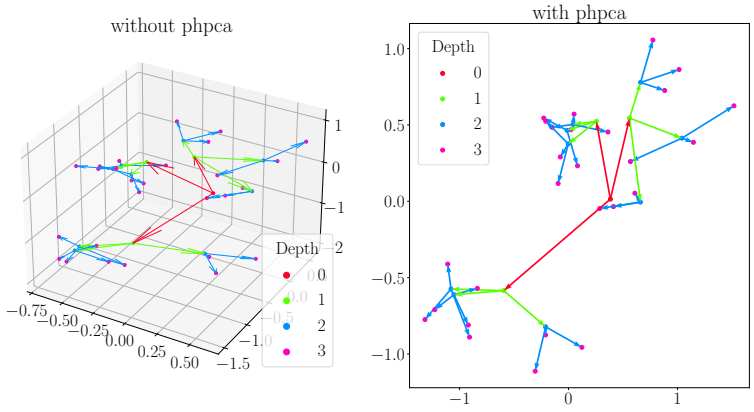

Figure 10: Result of phylogenetic PCA

5.3 Principal Geodesic Analysis

In section 5.2, we assumed all taxa to be embedded into some space with linear structure. This is, however, most likely not the case. In their original works, Polly et al. (2013) use superimposed landmark data, and use their coordinates as trait values. We argue however, that the space of landmark data is highly nonlinear and as previously suggested, using a framework as LDDMM to represent this data would be preferable.

We have seen that the direct mean of variables might not itself be an object of interest. Our solution to this is to reconsider the definition of a mean value. The most intuitive definition, is the geometric mean, where we are looking for an $\bar{x}$ that minimizes $\sum||x_i - \bar{x}||^2$ over some dataset $x_1, ..., x_n \in \mathbb{R}^d$. While this definition gives us an answer that is sound for $\mathbb{R}^d$, it still does not work for nonlinear geometries.

We do not constrain $\bar{x}$ in any way, and for that reason cannot guarantee that it itself represents an object of interest either. To circumvent this, we instead consider the dataset to be embedded into some riemannian manifold $\mathcal{M}$. This way, we can use the generalization of the geometric mean, called the Fréchet mean. Where d is the geodesic distance, we have (Pennec et al., 2020)

$$\bar{x} = \arg\min_{x \in \mathcal{M}} \sum d(x_i, x)^2$$

With this, we can now linearize each x_i around the fréchet mean and perform PCA in the tangent space $T_{\bar{x}}\mathcal{M}$. The hope is that by choosing a mean that minimizes the distances to the data points, the distortion that arises from the linearization via the logairthm map, is as small as possible.

We formalize this in a similar fashion to Fletcher et al. (2004). In section 5.1, we constructed a family of nested subspaces S_m which we defined each to be the span of the first k principal components. We want to define a similar concept on manifolds, where we find some family of submanifolds H_k, where the variance is minimized along a set of geodesics. First, we define projection onto a submanifold H to be (Fletcher et al., 2004)

$$\pi_H(x) = \arg\min_{y \in H} ||\text{Log}_x(y)||^2$$

With this, we can now frame the minimization problem for some neighborhood U of $\bar{x}$

$$v_k = \arg\max_{||v||=1} \sum ||\text{Log}_{\bar{x}}(\pi_H(x_i))||^2 \tag{9}$$

Where H is a geodesic submanifold, given as the image of a local linear neighborhood of 0 in $T_{\bar{x}}\mathcal{M}$. For a neighborhood U large enough to allow projections of all x_i, we define $H = \text{Exp}_{\bar{x}}(\text{span}(v_1, .., v_{k-1}, v) \cap U)$, and it now becomes clear that we only look for solutions in the tangent space around the mean.

This way, we can hide away the nonlinearities in the geometry by considering linearized neighborhoods around the data. We will not prove that algorithm 1 approximates the problem well, but refer the reader to Fletcher et al. (2004).

Algorithm 1 Principal Geodesic Analysis

$\bar{x} \leftarrow \arg\min_{x \in \mathcal{M}} \sum d(x_i, x)^2$
for x_i **do**
  $y_i \leftarrow \text{Log}_{\bar{x}}(x_i)$
end for
$\Sigma \leftarrow \frac{1}{n-1} Y^\top Y$
$U \Gamma V^\top \leftarrow \Sigma$
return V, Γ

In its essence, we take each data point $x_i \in \mathcal{M}$ and linearize it around $\bar{x}$ by finding minimizing geodesics. Thus PGA factors in the nonlinearities of $\mathcal{M}$ in two ways; 1. We constrain the sample mean. 2. We obtain vector representations of each x_i via a first order approximation of the geometry around the mean.

The algorithm reduces to something extremely simple, that is the standard PCA problem on the vector space $T_{\bar{x}}\mathcal{M}$. It is convenient to work with, as no extra effort is really required, except finding the fréchet mean, and linearizing locally. This also speaks to some of the limitations, as distortions will be more severe over greater distances. It is problematic because brownian motion is not preserved under arbitrary Log maps. This will be a key insight that we will use when presenting our own solution.

It is also important to note that algorithm 1 does not give a solution to (9), but rather provides an approximate solution via an approximation of the projection operator π_H. This algorithm does not have quite the same properties as the classical PCA, because we lose some of the connection to the underlying gaussianity from linearizing locally.

5.4 Phylogenetic Principal Geodesic Analysis

To account for phylogenetic covariance in principal geodesic analysis, we should consider two things; 1. How do we find the ancestral values $a \in \mathcal{M}$. 2. Is the calculation of the phylogenetic covariance matrix Σ_P unchanged from its original form in (8).

Finding the true root a suddenly has two important reasons. First it is important to choose the base point such that distortions from linearization is minimal. And secondly it is important to choose it in such a way that it does not skew surplus variance in the direction of independent taxa. It becomes a hard trade-off problem, where our algorithm chooses finding the true a which was the starting point of brownian motion.

We will sketch out an iterative minimization problem that shares a common theme of linearizing neighborhoods and performing operations in the tangent space. The method boils down to approximating the ancestral tangent a^t as the ancestral values of the data under a Log mapping, and then repeatedly projecting the ancestral tangent back onto $\mathcal{M}$.

The idea is somewhat similar to PGA, where we prefer to simply work in the tangent space of a point where distortions are not too high. The only difference being how we estimate our base point a.

Algorithm 2 Approximate Nonlinear Ancestral Values

$a_0 \leftarrow x_0$
do
  for x_i **do**
    $y_i \leftarrow \text{Log}_{a_j}(x_i)$
  end for
  $a^t \leftarrow (1^\top C^{-1}1)^{-1}1^\top C^{-1}Y$
  $a_{j+1} \leftarrow \text{Exp}_{a_j}(a^t)$
while $||a^t|| > \epsilon$

Algorithm 2 produces an approximate solution a_n for some convergence criterion ϵ. This algorithm does not prevent cycling, so selecting the stopping criterion accordingly is important.

The problem (and strength) of this algorithm is its approximation. It is a problem because a will most likely not be the true starting point of the brow-

nian motion process. We have to keep in mind that a^t is an unbiased estimator for the starting point of brownian motion, under the assumption that all the tangents y_i come from a matrix distribution defined in a tangent space. This means that the variations in our ancestral tangent will have the effects of the distortions in its estimation.

This is the idea behind iteratively sampling estimates of a^t, we hope that the effects of linearization iteratively become more and more insignificant. If we wanted to find the exact mean, one would need to perform (and first define) a maximum likelihood estimate over the distribution of phylogenies. Note that we would need a much higher dimensional manifold to represent our data in this case, since we would need to represent each phylogeny as a single point on a manifold itself.

After the ancestral root has been estimated $a \in \mathcal{M}$, all that is left to do is linearize each point $y_i = \text{Log}_a(x_i)$, and perform PCA on the vectors $y_1, \ldots, y_n$ using the true mean a. This gives us figure 11.

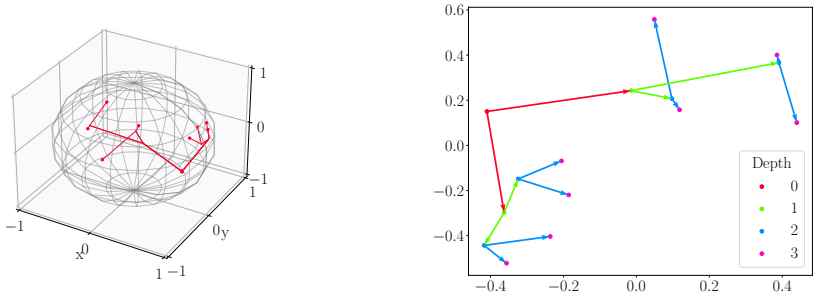

Figure 11: Left: Manifold data. Right: Projected data

Although perhaps not easily visualized on figure 11, the linearization is clearly visible. The projection looks a lot like a representation of the local neighborhood. This example is of course somewhat trivial, since the dimension of the manifold is the same as the projection space. In later examples, we will show this projection for very high dimensional manifolds.

6 Phylogenetic Projection by Increment Estimation

6.1 Underlying theory

In section 5.2 we introduced phylogenetic PCA, where we used the phylogeny matrix C, to down weigh covariance coming from common descent. We will

now introduce an alternate method which uses a similar tactic to estimate the inner nodes of the phylogeny, and project afterwards. We need a good way to provide an estimate $\hat{x}$ for any unseen node in the tree. Harmon (2019) provides both a maximum likelihood approach, as well as a bayesian method. Initially we will show the algorithm with a simple point estimate using a maximum likelihood. We will note that we only perform extremely simple and un-optimal estimates of inner nodes to demonstrate the algorithm. For any real use, we reccomend using any ancestral state reconstruction algorithm as a preprocessing step.

We are going to use C as in section 5.2, and use the likelihood obtained from brownian motion in section 4 due to Harmon (2019). Letting $V = \Sigma \otimes C$, and D the $nd \times d$ design matrix, we have

$$\mathcal{L}(x \mid a, V) = \frac{\exp\left(-\frac{1}{2}(x - Da)^\top V^{-1}(x - Da)\right)}{\sqrt{(2\pi)^{nd}\det(V)}} \tag{10}$$

Which come exactly from thee matrix distribution, just written out with a design matrix. A central observation is that any ancestor a, depends only on the vertices in the rooted subtree at a. At the root ancestor a, C is the exact same as in section 5.2, however any subtree will have a different C which is a partition of C viewed as a block matrix, except for a constant which we can subtract.

We will first introduce some notation, and afterward introduce a lemma which greatly increases computational speed. We denote the phylogeny matrix C_u to be the phylogeny matrix of the subphylogeny rooted in u and $X_{(u)}$ the datamatrix of all leaves in the subtree rooted at u.

Theorem 6.1 (Block diagonal lemma)**.** Given phylogeny $G = (V, E)$, for any internal node u with m direct descendants v^i, the phylogeny matrix C_u is given by

$$C_u = \begin{bmatrix} C_{v^1} + (v_t^1 - u_t) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & C_{v^m} + (v_t^m - u_t) \end{bmatrix}$$

Proof. First, it is clear that there must be 0s except for the block diagonal, since the covariance of any two nodes in two different subphylogenies is 0. Take now any pair x_i, x_j such that they belong to the same subphylogeny of u, rooted at v', with indicies k, l respectively in their subphylogeny. By definition $(C_{v'})_{kl}$ is the time of common ancestry, and so $(C_u)_{ij}$ should be exactly their time of common ancestry in their subphylogeny, plus the time along the edge (u, v') which is given by $(v'_t - u_t)$. Notice that two nodes x_1, x_2 in a subphylogeny $G' = (V', E')$ with path from roots $\pi_1, \pi_2 \subset E'$, which share

the first edge going out from the root (r, v_r) has covariance

$$\begin{aligned} Cov(x_1, x_2) &= \sum_{(u,v)\in\pi_1\cap\pi_2} (v_t - u_t) \\ &= ((v_r)_t - r_t) + \sum_{(u,v)\in\pi_1\cap\pi_2\setminus\{(r,v_r)\}} (v_t - u_t) \end{aligned} \tag{11}$$

The first term always exists because we assumed the two nodes shared ancestry, that is $\pi_1 \cap \pi_2 \neq \varnothing$. This gives rise to a telescoping sum, such that there is a branching node $(u, b) \in \pi_1 \cap \pi_2$, which gives $\mathrm{Cov}(x_1, x_2) = b_t - r_t$. Now, any phylogeny matrix for the sub phylogeny rooted at u in the path $\pi_1 \cap \pi_2$, simply corresponds to a sequence of terms in (11), which telescopes to $b_t - u_t$ where b is the branching node. With this, calculating the covariance in a phylogeny, is simply the covariance in the subphylogeny at any node a, plus the time between the phylogeny and subphylogeny $a_t - r_t$. Thus the adjusted phylogeny matrix becomes $(C_u)_{ij} = (C_{v'})_{kl} + (v'_t - u_t)$. □

A simple and intuitive way to think about this, is that adding another edge before the root, simply adds a single term to each entry in C which is the time along that edge. Alternatively, the time on a path in the phylogeny, is completely determined by its endpoints.

This gives rise to a recursive definition, where we estimate covariances of larger and larger subtrees of G, starting from the bottom. The idea is now to never double count any covariance, by considering only differences between nodes in G and their ancestors. Since the Covariance matrix $\Sigma \propto X^\top X$, is simply a matrix prduct, we can decompose the covariance matrix into a sum of self-products of each datapoint

Lemma 6.2 (Sum-Decomposition of Covariance matrix)**.** Given n datapoints $x_1, ...x_n \in \mathbb{R}^d$, and their corresponding data matrix $X \in \mathbb{R}^{n\times d}$, we can decompose the featurewise covariance matrix into a sum of self-products

$$\Sigma \propto X^\top X = \sum x_i^\top x_i$$

Proof. By the definition of the matrix product we have

$$\begin{aligned} (X^\top X)_{rc} &= \sum_{i=1}^{n} (X^\top)_{ri} X_{ic} \\ &= \sum_{i=1}^{n} (x_i)_r (x_i)_c \\ &= \sum_{i=1}^{n} (x_i^\top x_i)_{rc} \\ &= \left(\sum_{i=1}^{n} x_i^\top x_i \right)_{rc} \end{aligned}$$

Where the third equality is due to $(x^\top x)_{rc} = x_r x_c$ because x is a $1 \times d$ matrix and thus is an implicit sum over a single term. □

We can sum over each independent variation in G by

$$\Sigma_G = \frac{1}{|E| - 1} \sum_{(u,v) \in E} \frac{(v-u)^\top (v-u)}{v_t - u_t} \tag{12}$$

Where each u, v is either the result of an ancstral state reconstruction procedure of $\mathcal{L}$, or a leaf node which is known. As we prefaced the section with, we recommend estimating the internal nodes as a preprocessing step.

This may initially seem odd since each vertex is centered differently, however, we can imagine this process as being equivalent to calculating the covariance matrix for all variations centered in the origin. This is a direct result from the independent steps property. All edges $e = (u, v)$ can be thought of as a single increment Δ_e, where $\Delta_e \sim \mathcal{N}(u, t\Sigma)$. Now, we know both t and u, this means we can normalize each variation with $\frac{\Delta - u}{\sqrt{t}}$, where $t = v_t - u_t$. This gives equation (12) directly, as simply an unbiased estimator of Σ, given a set of increments.

Calculating Σ_G and estimating inner nodes constitutes fitting the model, and now the question of projection remains. Since we only measured independent variations, it is natural to only project data in this setting too.

Definition 6.1 (Ancestral projection)**.** Where $a_u \in \mathcal{A}$ is the direct ancestor of $u \in V \setminus \{r\}$ and r is the root of G, and π_G specifies a projection parameterized by G, we have ancestral projection $A(\pi_G)$

$$A(\pi_G)(u) = \begin{cases} A(\pi_G)(a_u) + \pi_G(u - a_u) & \text{if } u \text{ has a parent.} \\ \pi_G(u) & \text{otherwise.} \end{cases} \tag{13}$$

This definition is rather natural; to project a point u, we must first remove its variation of ancestral relations $u - a_u$, and then project. This projection however, is relative to the ancestor, so we must also add the projection $A(\pi_G)(a_u)$ to compensate. This gives rise to a top down procedure where we recursively project each estimated inner node.

Theorem 6.3 (Linear Ancestral Projection)**.** If π_G is linear, that is $\pi_G(a+b) = \pi_G(a) + \pi_G(b)$ then $A(\pi_G)(u) = \pi_G(u)$.

This is because the sum telescopes, that is $\pi_G(u - a_u) = \pi_G(u) - \pi_G(a_u)$ where each previous term had a single $\pi_G(a_u)$, and thus will cancel out.

Luckily, we have seen many examples of finding linear projection operators of this kind. We can find one by simply performing an eigen-decomposition of Σ_G as in classical PCA. Denoting U_k the matrix with the first k eigen vectors of Σ_G as columns, sorted in decreasing order by their corresponding eigenvalue.

We then form the linear projection operator $\pi_G^k(x) = xU_k$, as we would in classical PCA.

The central idea behind this algorithm, is that by looking only on the increments between each node in the phylogeny, the effect of linearization should be smaller when adapting this algorithm in a manifold setting.

We will only propose a way to adapt this algorithm to Lie groups, however it is only due to the simplicity in finding consistent bases between tangent spaces.

As before, we can imagine a set of edges $(u, v) \in G \times G$, where G is the Lie group we are considering. Again, recall that a Lie group is roughly speaking a manifold with additional structure in the tangent space of the identity element.

We propose to define the increment of an edge $e = (u, v)$ as the linearization $\text{Log}_u(v)$. Now, for each edge (u, v) we get an increment in T_uG, we can now use the left invariant fields $X_i(u)$ which arose from the push forward $(L_u)_*(e_i)$ (where e_i was a basis element for the lie algebra), to calculate the outer products $\Delta_e^\top \Delta$ in the lie algebra, and obtain an estimate of Σ. We do not provide an implementation of the manifold variant, nor proof of correctness. We simply postulate that considering only local increments could significantly reduce the effects of distortion.

6.2 Algorithm

We will now provide pseudocode for and general implementation details for the projection method described in section 6.1. As input to the algorithm, we assume we are given a phylogeny $G = (V, E)$ where only the leaves $\mathcal{T}$ have known values. We also assume we are given a scale parameter σ^2 and know d where $\mathcal{T} \subseteq \mathbb{R}^d$. We first introduce the algorithm for determining Σ_G

Algorithm 3 Fitting Phylogenetic Projection by Ancestor Estimation

1: **procedure** RECURSE_FIT($T = (V_T, E_T)$)
2: $x \leftarrow \text{ROOT}(T)$
3: **for** $(x, v) \in E_T$ **do**
4: $T' \leftarrow \text{SUBTREE}(v, T)$
5: RECURSE_FIT(T')
6: **end for**
7: $C_x \leftarrow C$ via theorem 6.1
8: $\hat{x} \leftarrow (1^\top C_x^{-1} 1)^{-1} 1^\top C_x^{-1} L_T$
9: **for** $(x, v) \in E_T$ **do**
10: $\Sigma_G \leftarrow \Sigma_G + \frac{(v-\hat{x})^\top (v-\hat{x})}{v_t - x_t}$
11: **end for**
12: **end procedure**
13: **for** $u \in \mathcal{T}$ **do**
14: $M_u \leftarrow u$
15: $\Sigma_G \leftarrow 0^{d \times d}$
16: **end for**
17: RECURSE_FIT(G)
18: $\Sigma_G \leftarrow \frac{1}{|E|-1} \Sigma_G$
19: **return** EIGEN_DECOMPOSITION(Σ_G)

Where SUBTREE(v, T) returns the subtree of T rooted at v and the data matrix L_T consists of all leaves in a subtree T

The algorithm has initialization steps on line 13-16, and calls a recursive subroutine RECURSE_FIT on the root of the phylogenetic tree. The subroutine calculates local covariances for each internal node $a \in \mathcal{A}$, by first performing a crude estimate of the internal node itself on line 8, centering and then by lemma 6.2, on line 9 we store the covariance that arises in the branching point at the current inner node we are processing.

The subroutine recurses before taking any action, which is equivalent to starting the algorithm at the leaves. This is important because (10) is completely determined by all direct descendants of any node. By performing inference bottom up, we ensure that at any branching point a, the subtree at a is completely determined, except for the root, a.

7 Results

7.1 Error Propagation

We will now show a series observations, made about the different algorithms we introduce. We start by considering the internal estimates of the phylogeny.

Since we used a simple point estimate each internal node, we expect to propagate uncertainty upwards in the tree. To verify this claim, we generated $100b$ different phylogenetic trees of species in $\mathbb{R}^3$, each with a constant timestep between generations, but with b different branching factors β, ie. each species s splits into β new ones after the constant timestep. We also expect a variance along each axis to be 1 for each generation. We then run algorithm 3 on the $100b$ trees, and for each, choose a random path from root to leaf. We then at each depth, measure the norm of the difference between the true ancestor x, and the ancestor $\hat{x}$ estimated via (10). Figure 12 shows both the mean estimation error in each internal node (dashed line) and standard deviation (filled region).

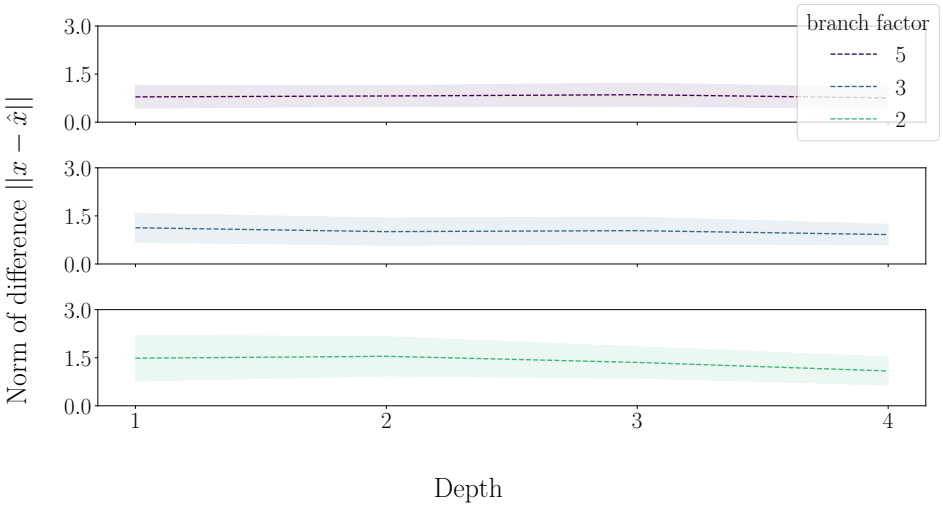

Figure 12: Uncertainty in ancestral estimation

Interestingly enough figure 12 empirically implies that error does not propagate as previously assumed. As expected however, the uncertainty is higher for lower branch factors. This is because we are trying to estimate nodes from fewer children, which as expected should give higher uncertainty. In our synthetic models we do not assume a constant time step for each generation, an example is figure 8 where the branch length decays exponentially over the generations.

As is implied by theorem 6.3, projection in this algorithm is rather simple. Whilst in principle we only project nodes with respect to their direct parents, theorem 6.3 tells us that this projection is not directly skewed by poor ancestral estimates. One may fear that poor ancestral estimates ruins the projections of the tips of the tree, in an error propagating fashion. This is however not the case. For any projection found via eigen decomposition of some covariance matrix Σ_G, the projection of the entire tree is parameterized only by Σ_G. This

implies that one could completely alter the tree, or throw away the estimates of inner nodes, without affection projection of observed nodes.

7.2 Visual results

We repeated the same experiment of figure 10, using our own propsed algorithm, and we get the result of figure 13 Next, we wanted to apply our

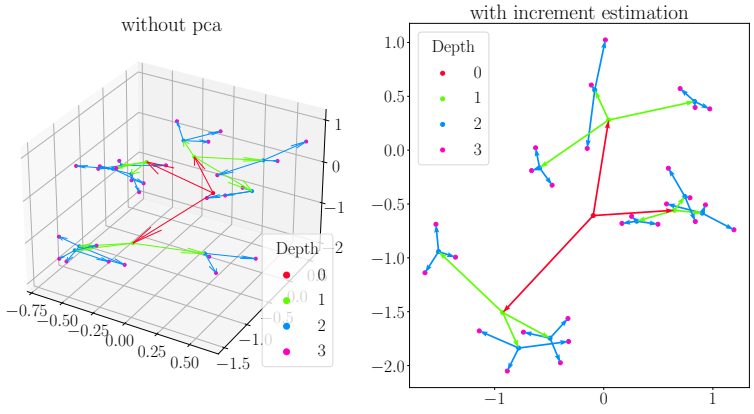

Figure 13: Projection by increment estimation

algorithms to a real world dataset. Due to the public availability we decided to attempt it on a subset of birds from the AVONET dataset (Tobias et al., 2022). We chose an arbitrary group of subspecies, with the following phylogeny seen in figure 14. The phylogeny contains a set of similar birds, with corresponding trait values. We used the 11 trait values

- Beak.Length_Culmen
- Beak.Length_Nares
- Beak.Width
- Beak.Depth
- Tarsus.Length
- Wing.Length
- Kipps.Distance
- Secondary1
- Hand-Wing.Index
- Tail.Length
- Mass

And applied both phylo PCA and our own algorithm of section 6.1. The result can be seen in figure 15 Here, we have plotted the estimated internal nodes on the right plot, and we do believe that they most likely are somewhat poor estimates of the true ancestral state values. We used the exact algorithm described, and did not preprocess with a better ancestral state reconstruction algorithm. We do believe that doing so could improve performance.

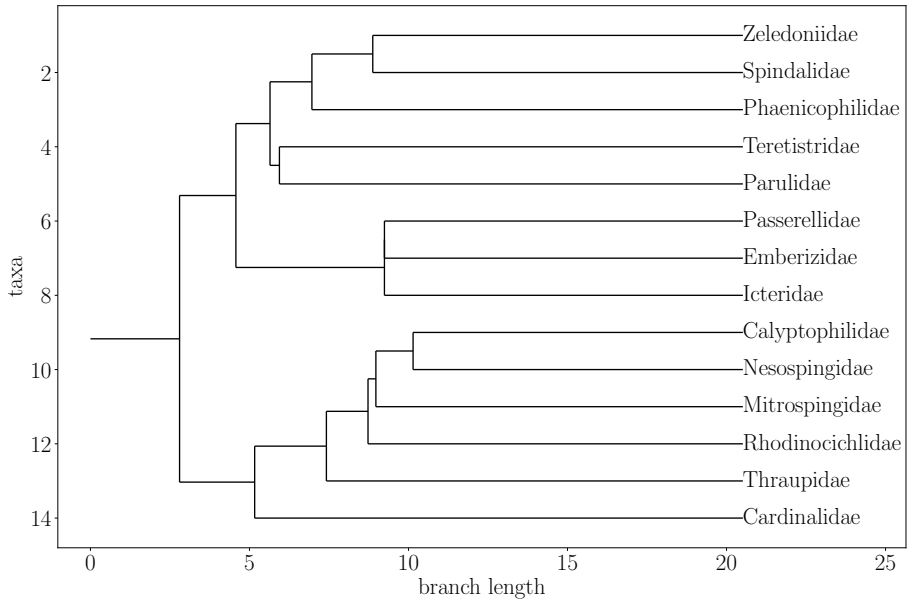

Figure 14: Subphylogeny of AVONET

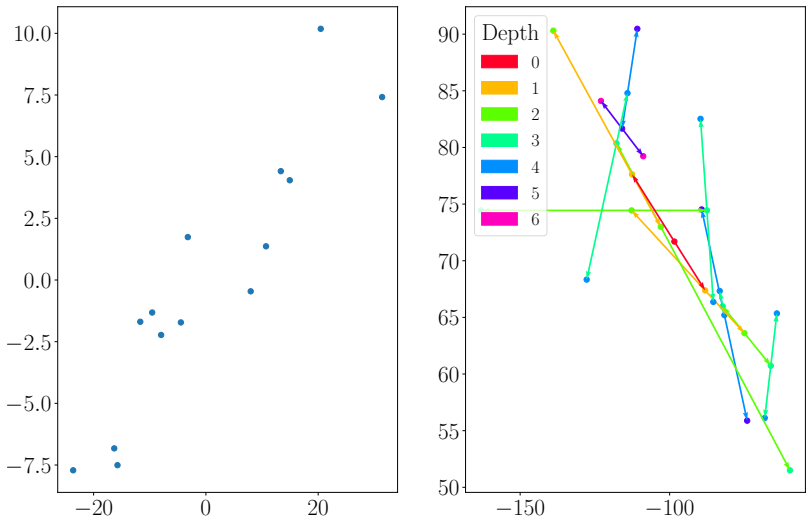

Figure 15: Projections of AVONET. Right: Phylo PCA. Left: Algorithm of section 6.1

It is interesting to see that the resemblance between the two methods is quite apparent. They both look almost identical ,except for a rotation and a scaling.

8 Future Work

There are still many open ended questions left. First a natural future project would be to provide a full implementation and generalization of the algorithm in section 6. Next an application of the algorithm on LDDMM representations of landmarks could provide interesting insight into the structure of the landmark spaces.

We also pose as an open question, whether or not the matrix W of equation (5), can be estimated from the Stratonovich equation, rather than from local linearizations of increments. If this was indeed possible, then one could construct Σ from W, without the distortions from linearization.

In general, it is hard to quantify whether or not one projection algorithm is better than another. Our adaptation in section 6 is theoretically only an improvement due to smaller effects of distortion. Besides that, the algorithm is identical with regular phylogenetic PCA. Finding an quantifying how large this improvement is depends both on the geometry of the space in which one applies the algorithm, and also on specific sampled data. If we had a simple measure that could measure the effects of these distortions, improvements would be much easier to track. In the end projections are just that, they are somewhat arbitrary, and many of the popular dimensionality reduction methods such as UMAP (McInnes et al., 2018) express what they believe is an optimal reduction problem, and optimize over it.

References

Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. *Learning From Data*. AMLBook, 2012. ISBN 1600490069.

Shane Barratt. A matrix gaussian distribution, 2018. URL https://arxiv.org/abs/1804.11010.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, Sep 1936. ISSN 1860-0980. doi: 10.1007/BF02288367. URL https://doi.org/10.1007/BF02288367.

P.T. Fletcher, Conglin Lu, S.M. Pizer, and Sarang Joshi. Principal geodesic analysis for the study of nonlinear statistics of shape. *IEEE Transactions on Medical Imaging*, 23(8):995–1005, 2004. doi: 10.1109/TMI.2004.831793.

Luke Harmon. Phylogenetic comparative methods: Learning from trees, May 2019. URL ecoevorxiv.org/e3xnr.

Elton P Hsu. A brief introduction to brownian motion on a riemannian manifold, 2008.

Paul O. Lewis. A Likelihood Approach to Estimating Phylogeny from Discrete Morphological Character Data. *Systematic Biology*, 50(6):913–925, 11 2001. ISSN 1063-5157. doi: 10.1080/106351501753462876. URL https://doi.org/10.1080/106351501753462876.

Georg Lindgren, Holger Rootzén, and Maria Sandsten. *Stationary Stochastic Processes for Scientists and Engineers*. Chapman and Hall, 10 2013. ISBN 9780429190292. doi: 10.1201/b15922.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018. URL https://arxiv.org/abs/1802.03426.

J.R. Munkres. *Topology*. Featured Titles for Topology. Prentice Hall, Incorporated, 2000. ISBN 9780131816299. URL https://books.google.dk/books?id=XjoZAQAAIAAJ.

Xavier Pennec, Stefan Sommer, and Tom Fletcher, editors. *Riemannian Geometric Statistics in Medical Image Analysis*. Academic Press, 2020. ISBN 978-0-12-814725-2. doi: https://doi.org/10.1016/B978-0-12-814725-2.00002-9.

P. David Polly, A. Michelle Lawing, Anne-Claire Fabre, and Anjali Goswami. Phylogenetic principal components analysis and geometric morphometrics. *Hystrix, the Italian Journal of Mammalogy*, 24(1):33–41, 2013. ISSN 0394-1914. doi: 10.4404/hystrix-24.1-6383. URL https://doi.org/10.4404/hystrix-24.1-6383.

Sidney I. Resnick. *Adventures in Stochastic Processes*. Birkhauser Verlag, CHE, 1992. ISBN 0817635912.

Liam Revell and Luke Harmon. Testing quantitative genetic hypotheses about the evolutionary rate matrix for continuous characters. *Evol. Ecol. Res.*, 10, 03 2008.

S.M. Ross. *Stochastic Processes*. Wiley series in probability and mathematical statistics. Wiley, 1996. ISBN 9780471120629. URL `https://books.google.dk/books?id=ImUPAQAAMAAJ`.

Joseph A. Tobias, Catherine Sheard, Alex L. Pigot, Adam J.M. Devenish, Jingyi Yang, Ferran Sayol, Montague H.C. Neate-Clegg, Nico Alioravainen, Thomas L. Weeks, Robert A. Barber, Patrick A. Walkden, Hannah E.A. MacGregor, Samuel E.I. Jones, Claire Vincent, Anna G. Phillips, Nicola M. Marples, Flavia A. Montaño-Centellas, Victor Leandro-Silva, Santiago Claramunt, Bianca Darski, Benjamin G. Freeman, Tom P. Bregman, Christopher R. Cooney, Emma C. Hughes, Elliot J.R. Capp, Zoë K. Varley, Nicholas R. Friedman, Heiko Korntheuer, Andrea Corrales-Vargas, Christopher H. Trisos, Brian C. Weeks, Dagmar M. Hanz, Till Töpfer, Gustavo A. Bravo, Vladimír Remeš, Larissa Nowak, Lincoln S. Carneiro, Amilkar J. Moncada R., Beata Matysioková, Daniel T. Baldassarre, Alejandra Martínez-Salinas, Jared D. Wolfe, Philip M. Chapman, Benjamin G. Daly, Marjorie C. Sorensen, Alexander Neu, Michael A. Ford, Rebekah J. Mayhew, Luis Fabio Silveira, David J. Kelly, Nathaniel N.D. Annorbah, Henry S. Pollock, Ada M. Grabowska-Zhang, Jay P. McEntee, Juan Carlos T. Gonzalez, Camila G. Meneses, Marcia C. Muñoz, Luke L. Powell, Gabriel A. Jamie, Thomas J. Matthews, Oscar Johnson, Guilherme R.R. Brito, Kristof Zyskowski, Ross Crates, Michael G. Harvey, Maura Jurado Zevallos, Peter A. Hosner, Tom Bradfer-Lawrence, James M. Maley, F. Gary Stiles, Hevana S. Lima, Kaiya L. Provost, Moses Chibesa, Mmatjie Mashao, Jeffrey T. Howard, Edson Mlamba, Marcus A.H. Chua, Bicheng Li, M. Isabel Gómez, Natalia C. García, Martin Päckert, Jérôme Fuchs, Jarome R. Ali, Elizabeth P. Derryberry, Monica L. Carlson, Rolly C. Urriza, Kristin E. Brzeski, Dewi M. Prawiradilaga, Matt J. Rayner, Eliot T. Miller, Rauri C.K. Bowie, René Marie Lafontaine, R. Paul Scofield, Yingqiang Lou, Lankani Somarathna, Denis Lepage, Marshall Illif, Eike Lena Neuschulz, Mathias Templin, D. Matthias Dehling, Jacob C. Cooper, Olivier S.G. Pauwels, Kangkuso Analuddin, Jon Fjeldså, Nathalie Seddon, Paul R. Sweet, Fabrice A.J. DeClerck, Luciano N. Naka, Jeffrey D. Brawn, Alexandre Aleixo, Katrin Böhning-Gaese, Carsten Rahbek, Susanne A. Fritz, Gavin H. Thomas, and Matthias Schleuning. Avonet: morphological, ecological and geographical data for all birds. *Ecology Letters*, 25(3):581–597, March 2022. ISSN 1461-023X. doi: 10.1111/ele.13898. Publisher Copyright: © 2022 The Authors. Ecology Letters published by John Wiley & Sons Ltd.

D.J. Waal. Matrix-valued distributions. *Encyclopedia of Statistical Sciences*, 5:326–333, 08 2006. doi: 10.1002/0471667196.ess1565.pub2.