Imperial College London

# Simulated inapplicable data predict false reconstruction of evolutionary trees

Hovig Artinian

August 2020

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science at Imperial College London

Formatted in the journal style of Systematic Biology

Submitted for the **MSc in Computational Methods in Ecology and Evolution**

## Declaration

# 1 Abstract

*Keywords: simulation method, inapplicable data, hierarchical dependency, morphology, phylogenetic analysis, evolutionary trees*

## 2 Introduction

In the last few decades, technological advancements in computational biology have enabled high-throughput sequencing of genes and whole genomes. As a result, there is much interest in bioinformatics research focused on developing mathematical models to decipher genetic information in the efforts of understanding the stochastic behaviour of nature.

In evolutionary biology, scientists use probabilistic models for phylogenetic reconstruction of evolutionary trees. These models aim to describe the evolutionary dynamics underlying biological processes in a mathematical context. Many of these models are incorporated in computer programs, such as PAUP* (Swofford 2002), to increase efficiency and speed in processing and analyzing large datasets. An adequate amount of methods have been developed within a statistical framework for molecular sequence data using such probabilistic models to infer phylogenies (Jukes & Cantor 1969, Felsenstein 1981, Hasegawa et al. 1985), while those for morphological data still lag. The latter, however, is just as important. Palaeontologists and archaeologists, for example, are interested in studying ancient taxa (species), where fossil records are the only accessible data, leading to a better understanding of the relationships between ancient and living, descendent, taxa (Wiens 2004). Morphological data also plays a role in molecular clock studies, where fossil records are used to detect calibration points – divergence time between species (Lee 1999). Despite the importance of morphology, molecular models are still in use to analyze such data (O'Reilly et al. 2016, Lewis 2001). These models, however, do not account for the peculiarities that exist in morphological data, leading to undesirable results (Goloboff et al. 2019).

In morphology, it is often the case that the state of a character is dependent on the state of another character. A character can be any morphological trait or anatomical feature of species; the equivalent of a DNA sequence site in molecular evolution. Maddison (1993) was first to present the classic red-tail, blue-tail case: species with no tail have no tail colour. Therefore, the state of character "tail colour" can only be determined in the presence of the "tail" character. If tail is absent, tail colour becomes inapplicable since it cannot be in any state. This phenomenon is known as hierarchical dependency.

Obtaining these inapplicable data experimentally is not simple since they are not directly observed in nature; therefore, gathering raw morphological data becomes a challenge. Failing to reflect inapplicable data in our observations gives rise to problems when encoding them as input for computer software like PAUP* to perform statistical analyses. In a given dataset, when a dependent character, such as tail colour, becomes inapplicable, its state cannot be assigned a logical value; they are treated as missing data, leading to both computational complexities and biological incongruities (Nixon & Davis 1991, Hawkins et al. 1997, Brazeau 2011). An example of the latter is when an ancestor with no tail is assigned a state for tail colour (Platnick et al. 1991). Due to these limitations, it is necessary to resort to computational simulation methods to generate data (Pagel 1994, Nielsen 2002). Such simulation methods are available for sequence data (Fletcher & Yang 2009, Butterfield et al. 2004), but none exist in the literature that are robust and satisfactory for morphological data.

In this work, I develop a simulation method to generate morphological data with hierarchical dependencies among characters. I aim to explore the consequences of having such a logical

internal inconsistency in a given dataset and whether or not they add up to a significant problem when molecular models are used for analysis. I expect failure to capture such configurations will distort the evolutionary tree being reconstructed by overestimating the branch lengths, leading to false recovery of its topology.

# 3 Methods

In this section, I describe the overall workflow of my project. Prior to that, however, I provide some preliminary information about probabilistic evolutionary models to give some context behind the workings of my simulation method.

## 3.1 Preliminary Information

In any phylogenetic analysis being performed within a statistical framework, a probabilistic evolutionary model has 3 components:

- Rate matrix ($Q$)- gives the rates of a single mutation event for all states at an infinitesimal period of time
- Transition probabilities ($P(t)$) - give the probabilities of transitioning from one state to another at any given time $t$ in the form of a matrix.
- Stationary distribution ($\pi$) - gives the relative frequencies of the states at equilibrium

From a likelihood-based approach, probabilistic evolutionary models use transition probabilities to try to mimic the evolutionary processes occuring from one node of an evolutionary tree to another across a lineage, i.e. branch. Programs like PAUP* can reconstruct all possible evolutionary trees and assign probabilities for each of them. The goal is to find the maximum likelihood tree, i.e. the tree that is the most likely to explain the given data. This process leads to estimation of parameters of interest, such as branch lengths, topology, and rates of evolution.

Calculating the likelihood of a tree is simply the product of the likelihood of its lineages. This is done by computing the transition probabilities at each branch by solving the following differential equation:

$$\frac{dP(t)}{dt} = P(t)Q \tag{1}$$

Note that both $P(t)$ and $Q$ are in matrix form. Therefore, the solution to this equation, given that $P(0) = I$, is a power series that always converges:

$$P(t) = e^{Qt} = I + Qt + \frac{(Qt)^2}{2!} + \frac{(Qt)^3}{3!} + ... \tag{2}$$

$P(t)$ is a matrix of transition probabilities at a given continuous value $t$. For an evolutionary tree, $t$ is no longer represented as time, but as branch length. The length of a branch is defined as the average number of mutation events per character across a branch. For very large values of $t$, the transition probability matrix will converge and reach equilibrium, with probabilities of the

states following the stationary distribution. Each element of the matrix, $P_{ij}(t)$, is the probability of $j$ being the state at the end of a branch of length $t$, given $i$ as the starting state of that branch.

Details on the derivation of the differential equation can be found in the Supplementary Information.

## 3.2 Project Workflow

The workflow for my project can be summarized in 3 main phases, as illustrated in Figure 1.

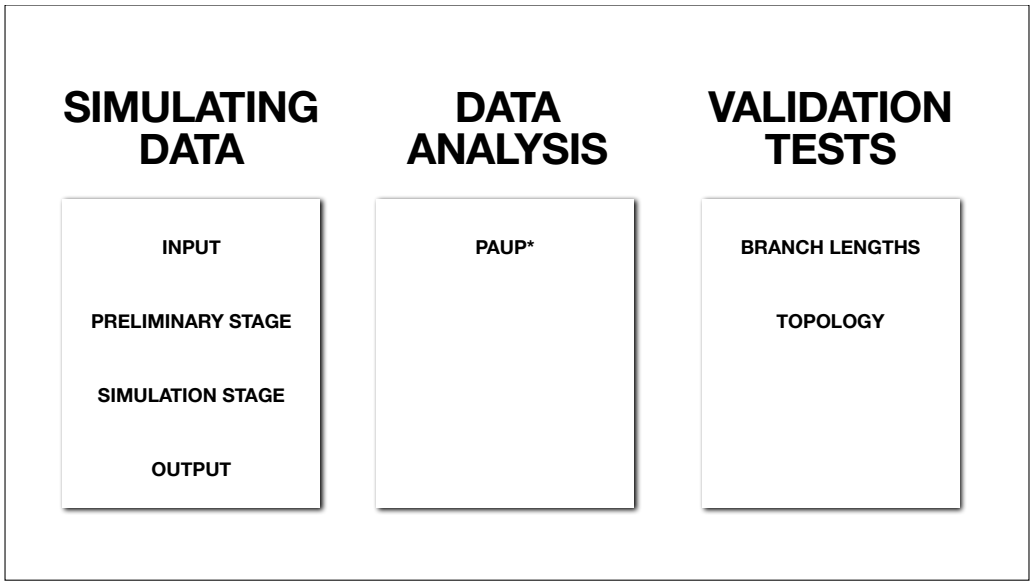


Figure 1

## 3.3 Simulating Data

In statistical inference, we are usually given the data and the model, and we try to predict and give estimates to the parameters of interest. To simulate data, however, we need to approach the situation from the opposite end of the statistical framework. We start with given estimated parameters and we reverse engineer the process by using the model to sample and generate data. Further detail is given in the simulation stage of this phase.

My simulation method is comprised of 4 parts:

1. **Input** - I start my simulation method by accepting some inputs from the user:
    - Nexus or Newick files - include one or more evolutionary trees with estimated branch lengths and topology (parameters of interest)
    - Number of simulations - number of datasets the user wants to produce
    - Number of characters - number of discrete morphological data the user wants to produce
    - Frequency of dependent characters - proportion of characters that are going to be hierarchically-dependent on other characters

- Rate matrix - which will be used during transition probability calculations. These rates can be estimated from a previous statistical analysis for example
- Number of states - to set the characters either as binary (0 or 1) or multistate (more than 2)
- Stationary distribution - which is usually based on the model being used

2. **Preliminary stage** - Once the method receives the inputs, I perform some preliminary tasks needed for the simulations. These involve:

   - extracting estimated branch lengths and topology from the input tree files. The latter is done by extracting the nodes in a postorder manner
   - calculating the transition probabilities for each lineage using equation 2, which involves the rate matrix and the estimated branch lengths
   - separating the characters into ”independent” and ”dependent” characters based on the provided number of characters and frequency of dependent characters. For example, if there are 200 characters, and the frequency of dependent characters is 0.1, then my dataset will include 20 dependent characters.

3. **Simulation stage** - For simplicity, I will describe the process of generating data for one simulation. This can be replicated for as many simulations as the user specifies.

   - **Sampling independent characters** - For each independent character, I start at the root, where I sample from the stationary distribution, under the assumption that probabilities of the states of a character converge at the ancestral node of a tree. Once the ancestral state is determined, I traverse the tree all the way up to the tip nodes to determine the states of the character at the rest of the nodes. At each internal node, I sample a state of the character from the probability distribution given by the relative transition probability matrix at that branch. Once the states of all independent characters are determined, I move on to sampling mutations for each lineage.
   - **Mapping mutational histories** - The method I use for mapping mutational histories is very similar to the one proposed by Nielsen (2002). Instead of using DNA sequences, I deal with discrete data. For each independent character, I start from the state of the parent node at each lineage of the tree and simulate mutational events. I assume an exponential rate for each event; therefore I draw waiting times for each mutation from an exponential distribution. When the total waiting time exceeds the branch length at that lineage, the simulation terminates. This process is repeated until a mapping is found where the last mutated state is equal to the descendent state.
   - **Sampling dependent characters** - Once the mutational histories are determined, the states of the dependent characters can be sampled. For each dependent character, states are sampled depending on the mutational history of the parent character. For example, consider the ”tail” character with two possible states: absent (0) and

present (1). Assume its mutational history at a certain lineage is ”0-1-0-1”, then the state of the dependent character ”tail color” will have to be resampled twice - only when the ”tail” character is in the ”present” state - from the transition probability matrix relative to the lineage.

WORK ON THIS - When parent character is absent, probability of subordinate character being in state “missing” is one, and probability of it being observed is 0. Once the character becomes present, the probabilities switch, and there is a further step, where u calculate the probabilities of the states of the subordinate character.

4. **Output** - After the states of all characters have been sampled, the ones at the tip nodes are taken as values for the characters of each taxon and saved as a dataset in a Nexus file format as output. The columns of the dataset are the characters and the rows are the taxa. For *n* simulations, *n* datasets are produced.

Reading Nexus and Newick tree files, as well as extracting nodes and branch lengths, were performed using the package *DendroPy* (Sukumaran & Holder 2010) in Python v3.8.2 (van Rossum & Drake 2009).

## 3.4 Data Analysis

To perform a statistical analysis, I generate data using my simulation method, followed by plotting and comparing basic statistical measures. The latter was done using R v4.0.1 (R Core Team 2020). The model maximum likelihood tree I use for comparison is the one used by Lewis (2001) shown in Figure 2. It is a 4-taxon unrooted tree with two long and two short branches. The taxa are labeled as Taxon A, Taxon B, Taxon C, and Taxon D, and the branch lengths of their relative branches are [0.2, 0.05, 0.2, 0.05], respectively. The branch length of the internal branch, Branch X, is 0.05.

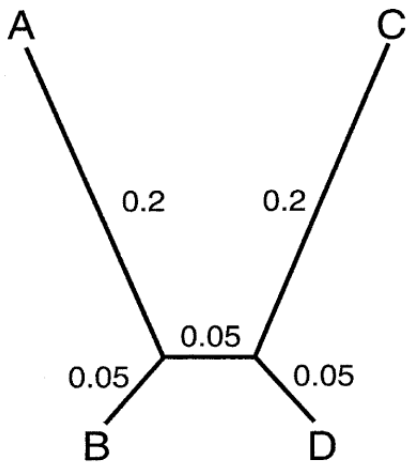


Figure 2: Model maximum likelihood tree with estimated branch lengths (values next to the branches) and topology used as reference for the statistical analysis section of this project. Adapted from ”A Likelihood Approach to Estimating Phylogeny from Discrete Morphological Character Data” by Paul O. Lewis, 2001, *Systematic Biology*, 50(6), pp. 913–925, *https://doi.org/10.1080/106351501753462876*

### 3.4.1 Model-fitting

I generate 2 sets of data. The first includes 1000 simulations of 500 independent characters and the 4 taxa from the model reference tree, while the second includes 1000 simulations of 500 characters, of which 100 (20%) are dependent characters. All characters have binary states and the same rate of evolution (= 1). I use a simple JC69 (Jukes & Cantor 1969) model (say it belongs to the Markov model family, introduce that briefly), so the stationary distribution is equal for all states, i.e. 50-50 for binary states in this situation.

Next, I feed these datasets into PAUP* v4.0a (Swofford 2002) to generate maximum likelihood trees with estimated branch lengths and topologies.

### 3.4.2 Validation Tests

I compare the average branch lengths of the simulated data of the two sets with that of the reference tree and plot them in a graph. I also compare the frequency of incorrect recovery of topology of the two sets in reference to the topology of the model tree.

## 4 Results

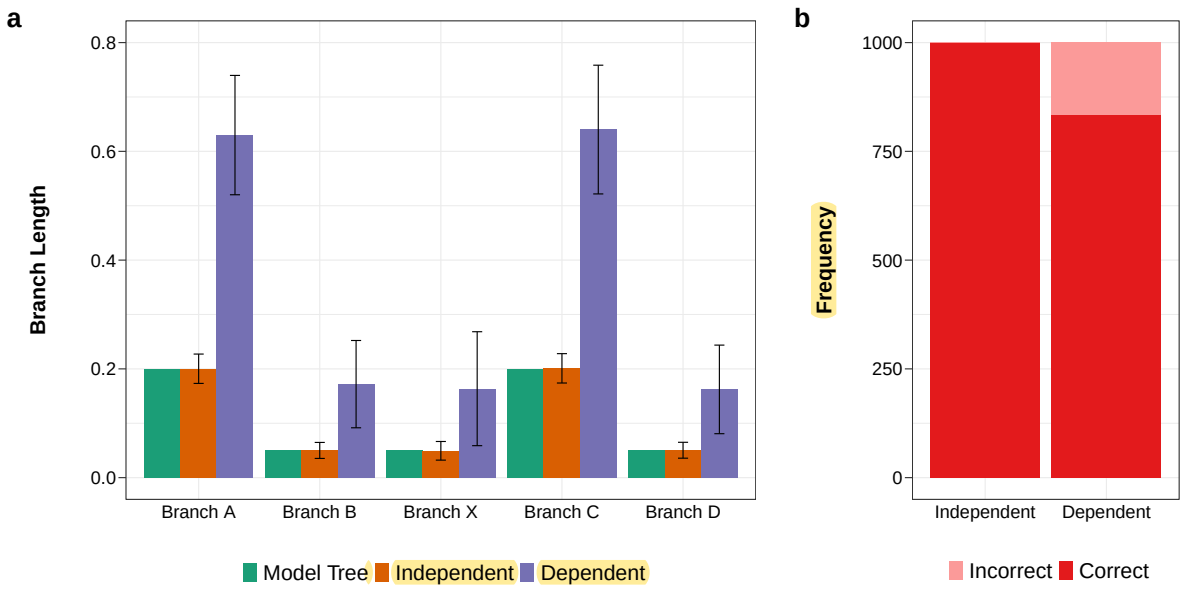


Figure 3: **Validation tests for two sets of simulations.** The first set is comprised of 1000 datasets of 500 independent characters, while the second one contains 500 characters, of which 100 (20%) are dependent characters. **(a)** Comparison of average branch lengths of the two sets with the reference tree at each lineage. Error bars are given by the standard deviations of the means. **(b)** Frequency of incorrect recovery of topology in the two sets.

The average branch lengths of the first set of simulations (all independent characters) for all branches are almost identical to the reference tree (Figure 3a) with mean values of 0.20024 ($\pm$0.02699), 0.05002 ($\pm$0.01471), 0.04934 ($\pm$0.01714), 0.20101 ($\pm$0.02700), 0.05038 ($\pm$0.01461), respectively, and low variance. Those of the second set, on the other hand, (with 20% dependent characters) are signficantly higher than those of the model tree, with average

values of 0.62992 (±0.10973), 0.17194 (±0.08026), 0.16356 (±0.10477), 0.64008 (±0.11836), 0.16225 (±0.08145), respectively, with high variance.

The first set also outperformed the second set in the number of times the correct topology was recovered (Figure 3b). Its success rate was 100%, i.e. PAUP* was able to correctly recover the topology of the model tree with all 1000 simulated datasets. That rate drops to 83.4% for the second set.

## 5 Discussion

The close prediction of estimated branch lengths using the first set (Figure 3a) is an indication of the accuracy of my simulation method ~~since Lewis (2001) used independent discrete characters for reconstructing the reference tree as well~~. ~~Moreover,~~ the fact that PAUP* was able to recover the correct topology 100% of the time (Figure 3b) further emphasises that. Therefore, the distorted results using the second set has to do with the model being used.

The JC69 model, similar to most other DNA sequence models, is not familiar with the hierarchical nature of discrete characters since it assumes all sequence sites/characters in a dataset are independent. So, when such characters are introduced in a given dataset, it treats them as missing data. Current programs for phylogenetic analysis disregard branches that have missing data, which affects the likelihood calculations of the tree. When those branches are disregarded, other branches further away from each other that should not have much effect on each other start to be more closely correlated. Such an event affects the likelihood calculations of the lineages, which, in turn, affect the likelihood calculations of possible trees. As a result, undesirable topologies become more probable than other more likely ones, as was observed when the second set of simulated data was used (Figure 3b). The success rate (83.4%) was higher than expected, but that can be due to the limited number of possible topologies. The model tree was a 4-taxon case where only 3 (unrooted) topologies could exist: (A, B, (C, D)), (A, (B, C), D), and ((A, C), B, D). For larger trees with many taxa, I expect much higher effects from overestimating the branch lengths, leading to more frequent incorrect recovery of topology. The overestimation of branch lengths is due to the correlation bias that arises among distant branches.

Branch length is calculated as the expected number of substitutions along a branch per character. For example, if a branch length is 0.01, then for 200 characters in a given dataset, only 2 characters are mutated and have an effect on the tree topology. As a branch gets longer, the more mutations we expect. The longer the branch, the more likely it is that more changes have occured in between two branches, leading to different topology possibilities. Out of all these possibilities, there is going to be one optimal solution for the tree once the data is given. Therefore, branch length estimation is key; overestimation will lead to bias when inferring tree topology.

Branch length estimation also has an effect on hierarchical relationships between independent and dependent characters. A dependent character can only exist when its parent character is present. My simulation method incorporates mutational histories of parent characters because they allow accurate representation of how dependent characters evolve. Across a

long branch, there are much more mutation events of a parent character, allowing its dependent character to evolve more frequently. For example, consider the tail-tail colour case. Across a lineage with more than one mutation event, tail will alternate between absent and present. Each time tail is present, the state of the tail colour will be resampled. Therefore, the states of tail colour between two of such events will have no historical connections to each other. If all the tails evolved recently, we expect a more random distribution of tail colours. But, if we have two clades that are quite close together, we expect a higher chance of finding same tail colour at the tips since it is less likely that a mutation event has occured. So, in effect, we are just evolving the dependent character over a short history, depending on the true history of the underlying parent character. The lower the probability of mutation events, the higher the probability that dependent characters of neighbouring nodes share a common history. In a way, dependent characters are not actually evolving along the tree; their evolution is more localized, at nodes where their parent characters were last present.

Recently, some attempts have been made to develop morphological models. Tarasov (2019) proposes quite a few in his paper; however, there were only two characters being studied when simulation were performed using these models. With real empirical datasets, comprised of hundreds of characters, softwares like PAUP* will not be able to perform well, due to the matrix structure of the models, which incorporate all characters. For large datasets, processing matrices becomes computationally exhaustive.

# 6 Conclusion

In this project, I developed a simulation method that generates inapplicable data. I simulated data and used them in a statistical framework. I showed that using a model, developed for DNA sequence data, leads to overestimation of branch lengths, affecting tree topology recovery. Efforts in improving the encoding of inapplicable data does not seem to be the optimal solution. It is long overdue that robust morphological models - ones that can account for the hierarchical nature of discrete characters - be developed to enable proper analysis of discrete character data. In doing so, phylogeny studies will become more robust as they will be able to incorporate both molecular and morphological models in their analyses.

# 7 Limitations & Future work

My simulation method generates characters with binary states only. A possible extention could be to allow multistate characters to exist as well. Most morphological characters, however, have binary states. So, the results shown are still acceptable.

Time between each mutational event was drawn from an exponential distribution in my simulation method. While genetic bases are accepted to be evolving at an exponential rate, it is not necesseraly the case for morphological data. I also set the rates of evolution of all characters as 1, but that is not biologically accurate; we know that some characters evolve faster than others. A possible extension to my method would be to allow different rates of evolution across characters. One way is to draw from a discrete gamma distribution for the

relative rates of discrete morphological data, which is assumed to be an appropriate distribution (Yang 1994).

Another extension of my method could be to allow rate of evolution of independent characters be affected by the number of subordinate characters they have. This could be possible by drawing hierarchical relationships from a Poisson distribution, where most dependent characters are in a one-to-one relationship with their parent characters, but there would also be few that are dependent on the same parent character. It would be expected that having the latter case in the dataset would lead to high correlation, leading to overestimation of branch lengths if currently-available models are used for analysis.

## References


Brazeau, M. D. (2011), 'Problematic character coding methods in morphology and their effects'.

Butterfield, A., Vedagiri, V., Lang, E., Lawrence, C., Wakefield, M. J., Isaev, A. & Huttley, G. A. (2004), 'PyEvolve: A toolkit for statistical modelling of molecular evolution', *BMC Bioinformatics* .

Felsenstein, J. (1981), 'Evolutionary trees from DNA sequences: A maximum likelihood approach', *Journal of Molecular Evolution* .

Fletcher, W. & Yang, Z. (2009), 'INDELible: A flexible simulator of biological sequence evolution', *Molecular Biology and Evolution* .

Goloboff, P. A., Pittman, M., Pol, D. & Xu, X. (2019), 'Morphological data sets fit a common mechanism much more poorly than DNA sequences and call into question the Mkv model', *Systematic Biology* .

Hasegawa, M., Kishino, H. & aki Yano, T. (1985), 'Dating of the human-ape splitting by a molecular clock of mitochondrial DNA', *Journal of Molecular Evolution* .

Hawkins, J. A., Hughes, C. E. & Scotland, R. W. (1997), 'Primary Homology Assessment, Characters and Character States'.

Jukes, T. H. & Cantor, C. R. (1969), Evolution of protein molecules BT - Mammalian protein metabolism, *in* 'Mammalian protein metabolism'.

Lee, M. S. (1999), 'Molecular clock calibrations and metazoan divergence dates', *Journal of Molecular Evolution* **49**(3), 385–391.

Lewis, P. O. (2001), 'A likelihood approach to estimating phylogeny from discrete morphological character data', *Systematic Biology* **50**(6), 913–925.

Maddison, W. P. (1993), 'Missing data versus missing characters in phylogenetic analysis', *Systematic Biology* **42**(4), 576–581.

Nielsen, R. (2002), Mapping mutations on phylogenies, *in* 'Systematic Biology'.

Nixon, K. C. & Davis, J. I. (1991), 'POLYMORPHIC TAXA, MISSING VALUES AND CLADISTIC ANALYSIS', *Cladistics* .

O'Reilly, J. E., Puttick, M. N., Parry, L., Tanner, A. R., Tarver, J. E., Fleming, J., Pisani, D. & Donoghue, P. C. (2016), 'Bayesian methods outperform parsimony but at the expense of precision in the estimation of phylogeny from discrete morphological data', *Biology Letters* .

Pagel, M. (1994), 'Detecting correlated evolution on phylogenies: A general method for the comparative analysis of discrete characters', *Proceedings of the Royal Society B: Biological Sciences* .

Platnick, N. I., Griswold, C. E. & Coddington, J. A. (1991), 'ON MISSING ENTRIES IN CLADISTIC ANALYSIS', *Cladistics* .

R Core Team (2020), 'R: A language and environment for statistical computing.'.

Sukumaran, J. & Holder, M. T. (2010), 'DendroPy: A Python library for phylogenetic computing', *Bioinformatics* .

Swofford, D. L. (2002), 'Phylogenetic Analysis Using Parsimony * (and other methods). Version 4'.

Tarasov, S. (2019), 'Integration of Anatomy Ontologies and Evo-Devo Using Structured Markov Models Suggests a New Framework for Modeling Discrete Phenotypic Traits', *Systematic Biology* .

van Rossum, G. & Drake, F. L. (2009), *Python 3 Reference Manual.*

Wiens, J. J. (2004), 'The role of morphological data in phylogeny reconstruction'.

Yang, Z. (1994), 'Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: Approximate methods', *Journal of Molecular Evolution* .

# 8 Supplementary Information

- derivation of differential equation dP(t)/dt = P(t)Q
- PAUP* code
- derivation of Figure 1 differential equations