# Algorithmic Description of Simulator

January 23, 2024

## 1 Overview

There are three steps to using the simulator:

1. A network is generated from the provided tree (i.e. reticulate edges are added).
2. Characters are generated according to the character class specifications.
3. Characters are evolved according to a particular evolution model.

The README in the Github repository describes how to use the command line arguments; in this document, we focus on the mathematical description of the simulator, which is adapted from the description of the simulator in Barbançon et al. [2013].

## 2 Network Generation

The parameters for network generation are:

- $\boldsymbol{N}$: The number of reticulate edges to create.
- $\boldsymbol{\epsilon}$: The minimum amount of time between the pair of edges between which we consider drawing a reticulate edge and their least common ancestor.
- $\boldsymbol{\alpha_{trm}}, \boldsymbol{\beta_{trm}}$: The shape parameters of the Beta distribution from which the transmission strength of a reticulate edge is drawn.

We first generate *candidate* pairs of edges between which a reticulate edge may be drawn. This is done by considering each pair of edges $e_1$ and $e_2$. The *overlap interval* between $e_1$ and $e_2$ is the interval $[t_1, t_2]$ such that for any $t_1 \leq t \leq t_2$, both edges are in existence at time $t$. We also find the time $t_0$ of the least common ancestor, which is the node furthest from the root that is an ancestor to both edges, of $e_1$ and $e_2$. An edge is a candidate if and only if both edges overlap in time and $t_1 - t_0 > \boldsymbol{\epsilon}$.

Next, we assign a score $S(\pi) \geq 0$ for each pair of edges $\pi$ using the function

$$S(\pi) = -\log\left(\frac{t_1 - t_0}{t_2 - t_0}\right)$$

We then draw (without replacement) from the distribution of scores until we have $\boldsymbol{N}$ pairs. To do this, we first draw $U_0 \sim U(0, \Sigma_i S(\pi_i))$. The interval where the random number falls corresponds to the pair of edges between which a contact edge will be drawn. This pair is then removed from the distribution, and the process repeats until we have $\boldsymbol{N}$ pairs.

After selecting the edges between which we will draw a contact edge, we decide where on the overlap interval $[t_1, t_2]$ to place the edge. We draw a random number $U_1 \sim U(0, 1)$. The time $t_c$ of the contact edge, $t_1 \leq t_c \leq t_2$, is given by

$$t_c = t_0 + (t_1 - t_0) \cdot \exp\left(U_1 \cdot \log\left(\frac{t_2 - t_0}{t_1 - t_0}\right)\right)$$

A reticulate edge is added at this contact time: this requires splitting the original edges and making new intermediate nodes at the contact points.

Finally, the strength $\kappa_e$ (transmission probability factor) of edge $e$ is calculated as $\kappa_e \sim B(\boldsymbol{\alpha_{trm}}, \boldsymbol{\beta_{trm}})$.

# 3 Character Generation

Each character belongs to a Character Class, which is group of characters with a shared set of properties. The parameters for a Character Class are:

- $\boldsymbol{N}$: The number of characters in this class.
- $\boldsymbol{\sigma_{dlc}} \in [0, \infty)$: Degree to which character class deviates from lexical clock (failure to be ultrametric).
- $\boldsymbol{\sigma_{het}} \in [0, \infty)$: Degree to which character class violates rates-across-sites assumption (degree of heterotachy).
- $\boldsymbol{type_{dlc}} \in \{\text{Individual}, \text{Uniform}\}$: Whether or not all characters in the class should have the same deviation from lexical clock (Uniform) or different (Individual).
- $\boldsymbol{H}$: Height of character class, i.e. the factor by which to multiply edge lengths (controls how "tall" the character class is).
- $\boldsymbol{\alpha_{trm}}, \boldsymbol{\beta_{trm}}$: The shape parameters of the Beta distribution from which the transmission strength of a particular character is drawn.

- Any additional parameters specific to the model of evolution.

To account for deviation from lexical clock and heterotachy, we just need to calculate edge length modifiers $elm(c, e)$ for each character $c$ and edge $e$. If $\boldsymbol{type_{dlc}} = \text{Uniform}$ (and no heterotachy), then $elm(c, e) = elm(e)$.

The edge length modifiers $elm(c, e)$ are determined as follows. For each edge $e$ and character $c$, draw $X_{c,e} \sim N(0, \boldsymbol{\sigma_{dlc}}^2)$ and $Y_{c,e} \sim N(0, \boldsymbol{\sigma_{het}}^2)$. If $\boldsymbol{type_{dlc}} = \text{Uniform}$, set $X_{c,e} = X_e$ (i.e. draw one number for all characters in the class). Then, define

$$elm(c, e) = \boldsymbol{H} \cdot \exp\left(X_{c,e} - \frac{\boldsymbol{\sigma_{dlc}}^2}{2}\right) \cdot \exp\left(Y_{c,e} - \frac{\boldsymbol{\sigma_{het}}^2}{2}\right)$$

We draw the site transmission factor $\pi_c$ in a similar manner to $\kappa_e$: $\pi_c \sim B(\boldsymbol{\alpha_{trm}}, \boldsymbol{\beta_{trm}})$.

This process is repeated for each character until we have $\boldsymbol{N}$ characters within the Character Class.

# 4 Character Evolution

The following algorithm for character evolution accounts for reticulate edges, and it is general enough to work with a wide variety of models of character evolution. Specifically, the models should require that each character evolves independently.

1. Draw the root state.
2. Then, evolve genetically, stopping at any *reticulate* nodes.
3. Identify the reticulate edges that the evolution stopped at in step (2); there must be at least one if step (2) stopped before finding all the leaves.
4. For each reticulate edge, determine whether or not there is transmission, and in which direction. So, draw $U_0 \sim U(0, 1)$. If $U_0 < \kappa_e \times \pi_c$, then transmission occurs.
5. Continue steps (2) through (4) until states have been selected for all the leaves (i.e. until step (2) does not halt due to finding reticulate edges). Character evolution is complete.

When step (2) says "evolve genetically", this means applying some model of genetic evolution to the phylogenetic tree in a top-down fashion. In this way, this algorithm is abstracted away from the particular evolution model. Thus, to describe such an evolution model, one just has to describe (1) how the root is chosen, (2) how the state of a daughter language is determined from the parent language on a single edge (typically this involves sampling from some distribution), and (3) how a reticulate edge is resolved. In the code, this corresponds to 3 abstract functions that must be overriden for each particular evolution algorithm.

There are two evolution models described here: the one with homoplasy [Warnow et al., 2006], and the one with polymorphism described in Anonymous [2024]. The following two sections treat each of these models.

### 4.0.1 The Model of Warnow et al. [2006]

Below is a description of the algorithm for the genetic evolution model presented in Warnow et al. [2006]. The additional parameters for the Character Class related to evolution are:

- $\boldsymbol{PML} \in \{\text{Phonological}, \text{Morphological}, \text{Lexical}\}$: The type of character.
- $\boldsymbol{h_{root}} \in [0,1]$: The probability that the root is in the homoplastic state. For a phonological character, this is always 0.
- $\boldsymbol{h} \in [0,1]$: The homoplasy factor used in infinitesimal rate calculation. Not used for phonological characters.

We can first define the infinitesimal character rates such that

$$\begin{aligned} q(h^*, h^*) + q(h^*, n') &= 1 \\ q(n, n) + q(n, n') + q(n, h^*) &= 1 \end{aligned}$$

Here, $n$ and $n'$ are non-homoplastic states, and $h^*$ is the sole homoplastic state. We define

$$\begin{aligned} q(h^*, h^*) &= 0, \\ q(h^*, n') &= 1 \\ q(n, n) &= 0 \\ q(n, n') &= \begin{cases} 1 - \boldsymbol{h} & \text{if } \boldsymbol{PML} \in \{\text{Morphological}, \text{Lexical}\} \\ 0 & \text{if } \boldsymbol{PML} = \text{Phonological} \end{cases} \\ q(n, h^*) &= \begin{cases} \boldsymbol{h} & \text{if } \boldsymbol{PML} \in \{\text{Morphological}, \text{Lexical}\} \\ 1 & \text{if } \boldsymbol{PML} = \text{Phonological} \end{cases} \end{aligned}$$

A root state is chosen as follows: draw $U_0 \sim U(0,1)$. If $U_0 < \boldsymbol{h_{root}}$, then $c_{root} = h^*$. Else, $c_{root} = n$.

Suppose we wish to evolve edge $e$, and $c_1$ is the state of the parent and $c_2$ is the state of the child. The goal is to determine $c_2$ given $c_1$. For this model, all that matters is whether or not $c_1$ is in the homoplastic or non-homoplastic state. Define:

$$\begin{aligned} p(n, h^*) &= q(n, h^*) \cdot \frac{1 - \exp(-elm(c,e) \cdot len(e) \cdot [q(h^*, n) + q(n, h^*)])}{q(h^*, n) + q(n, h^*)} \\ p(n, n') &= \frac{q(h^*, n') + q(n, h^*) \cdot \exp(-elm(c,e) \cdot len(e) \cdot [q(h^*, n) + q(n, h^*)])}{q(h^*, n) + q(n, h^*)} - \\ &\qquad \exp(-elm(c,e) \cdot len(e) \cdot (1 - q(n, n))) \\ p(n, n) &= \exp(-elm(c,e) \cdot len(e) \cdot (1 - q(n, n))) \\ p(h^*, h^*) &= \frac{q(n, h^*) + q(h^*, n') \cdot \exp(-elm(c,e) \cdot len(e) \cdot [q(h^*, n) + q(n, h^*)])}{q(h^*, n) + q(n, h^*)} \\ p(h^*, n) &= \frac{q(h^*, n') \cdot (1 - \exp(-elm(c,e) \cdot len(e) \cdot [q(h^*, n) + q(n, h^*)]))}{q(h^*, n) + q(n, h^*)} \end{aligned}$$

To determine $c_2$ from $c_1$, draw $U_1 \sim U(0,1)$. First consider that $c_1$ is non-homoplastic. If $0 < U_1 < p(n, h^*)$, set $c_2 = h^*$. If $p(n, h^*) < U_1 < p(n, h^*) + p(n, n')$, set $c_2 = n'$. Otherwise, set $c_2 = n$. If $c_1$ is homoplastic, then if $0 < U_1 < p(h^*, h^*)$, set $c_2 = h^*$. Otherwise set $c_2 = n'$. That's it!

To resolve a reticulate edge, pick a direction uniformly at random, and copy the state in the elected direction (i.e. replace the state of the target language with the one being transmitted across the edge).

### 4.0.2 The Polymorphism Model of Anonymous [2024]

In this model, the key point is that there is no substitution: all lexical change occurs due to some (however brief) period of polymorphism, after which some states disappear (but the number of states never goes below 1). The parameters for the character class related to evolution are provided below. For each character class these parameters may be set separately.

- $\boldsymbol{ML} \in \{\text{Morphological}, \text{Lexical}\}$: The type of character. We cannot have phonological characters with this model because it doesn't make sense to speak of polymorphism for phonological characters.
- $\boldsymbol{h_{root}} \in [0,1]$: The probability that the root is in the homoplastic state.
- $\boldsymbol{h} \in [0,1]$: The homoplasy factor used to determine whether a new state (result of a birth) is homoplastic (i.e. a state previously seen for this character).
- $\boldsymbol{\lambda}$: The birth rate for the birth-death model.
- $\boldsymbol{\mu}$: The death rate for the birth-death model.

We generate the root state in the same way as in Section 4.0.1. This means that the root is never polymorphic. Alternatively we could have a root edge prior to the root along which evolution occurs. To evolve an edge $e$, suppose $c_1$ is the set of states of the parent, and $c_2$ is the set of states of the child. The goal is to determine $c_2$ given $c_1$.

Suppose $t_0$ is the date of the parent and $t$ is the date of the child. To account for character scaling, we set $t_0 \leftarrow elm(c,e) \cdot t_0$ and $t \leftarrow elm(c,e) \cdot t$ for the following calculations. Here is the basic plan. We first draw the *waiting time* $t_w$ until the next event (birth or death), so the next event will occur at time $t_0 + t_w$. Then, we decide whether or not the event is a birth or death, and adjust the set of states accordingly. We then set $t_0 \leftarrow t_0 + t_w$ and repeat this process until we realize the next event will occur after time $t$. At this point, we can set $c_2$ equal to the set of states produced by the most recent event.

More specifically, we have a linear birth-death model with birth rate $\boldsymbol{\lambda}$ and death rate $\boldsymbol{\mu}$. Therefore, the waiting time $t_w$ between events is distributed $t_w \sim \text{Exp}(\boldsymbol{\lambda} + i\boldsymbol{\mu})$, where $i$ is the number of states at $t_0$. Once a waiting time $t_w$ has been drawn, the probability of the event being a birth is $\boldsymbol{\lambda}/(\boldsymbol{\lambda} + i\boldsymbol{\mu})$, and the probability of a death is $i\boldsymbol{\mu}/(\boldsymbol{\lambda} + i\boldsymbol{\mu})$ If the event is a birth, we add either the (distinct) homoplastic state with probability $\boldsymbol{h}$, or a new state with probability $1 - \boldsymbol{h}$. If the event is a death, we remove one of the states uniformly at random.

There are a few important caveats. If $i = 1$ at any point, then we have only one state and so we cannot permit a death. In this case, we draw $t_w \sim \text{Exp}(\boldsymbol{\lambda})$.The other issue occurs if we pick a waiting time such that $t_0 + t_w > t$. In other words, the problem arises when a waiting time is drawn such that the new event would occur at a time later than the child node $-$ this would imply that an event would occur simultaneously on the two independent edges leaving from the child node.

To solve this issue, we first calculate the probability $P(t_0 + t_w \leq t)$ as

$$P(t_0 + t_w \leq t) = F_{t_w}(t - t_0) = 1 - e^{-L(t-t_0)}$$

where $L = \boldsymbol{\lambda} + i\boldsymbol{\mu}$ and $F_{t_w}(x)$ is the cdf of $t_w \sim \text{Exp}(L)$. We then draw $U_0 \sim U(0,1)$, and if $U_0 < P(t_0 + t_w \leq t)$, then a new event will occur on this edge (before the child node). We have to be careful to draw $t_w \sim \text{Exp}(L)$ such that $t_w \leq t - t_0$. To ensure this, we define a pdf $\widetilde{f}_{t_w}(x) = Af_{t_w}(x)$ where $A$ is a scaling factor such that

$$\int_0^{t-t_0} \widetilde{f}_{t_w}(x)\,dx = \int_0^{t-t_0} Af_{t_w}(x)\,dx = 1$$

It turns out that

$$\widetilde{f}_{t_w}(x) = \frac{L \cdot e^{-Lx}}{1 - e^{-L(t-t_0)}}$$

To actually draw from this distribution, we use the probability integral transform. Draw $U_1 \sim U(0,1)$. Then we solve for $x$ in $\widetilde{F}_{t_w}(x) = U_1$. We find that

$$x_{draw} = \frac{1}{L} \cdot \log\left(\frac{C}{C - U_1}\right) \quad \text{where} \quad C = \frac{1}{1 - e^{-L(t-t_0)}}$$

On the other hand, if $U_0 > P(t_0 + t_w \leq t)$, then the next event will occur at a time after the child node. In this case, we set $c_2$ equal to the set of states at $t_0$. As in the previous case, we have to ensure that $t_w \sim \text{Exp}(L)$ is drawn such that $t_w > t - t_0$ for each of the edges leaving from the child node. We use a similar technique to the one above to find that

$$\widetilde{f}_{t_w}(x) = Le^{L(t-t_0-x)}$$

Importantly, we see that this is just $\text{Exp}(\boldsymbol{\lambda} + i\boldsymbol{\mu})$ shifted by $t - t_0$. Therefore, we can just set $t_0 \leftarrow t$ and continue to evolve as normal along the next edge.

# References

Anonymous. Addressing polymorphism in linguistic phylogenetics. *Under review*, 2024.

F. Barbançon, S. N. Evans, L. Nakhleh, D. Ringe, and T. Warnow. An experimental study comparing linguistic phylogenetic reconstruction methods. *Diachronica*, 30(2):143–170, 2013.

T. Warnow, S. N. Evans, D. Ringe, and L. Nakhleh. A stochastic model of language evolution that incorporates homoplasy and borrowing. In P. Forster and C. Renfrew, editors, *Phylogenetic Methods and the Prehistory of Languages*, pages 75–90. Cambridge: McDonald Institute for Archaeological Research, 2006.