SiPhyNetwork: An R package for Simulating Phylogenetic Networks

Joshua A. Justison[1,*], Claudia Solis-Lemus[2], and Tracy A. Heath[1]

[1] *Department of Ecology Evolution and Organismal Biology, Iowa State University, Ames, 50011, USA*

[2] *Wisconsin Institute for Discovery and Department of Plant Pathology, University of Wisconsin-Madison, Madison, WI, 53706, USA*

[*] *Correspondence to be sent to Josh Justison at 2200 Osborn Dr, Ames, Iowa 50011, USA. Emails to be sent to justison@iastate.edu*

Abstract

1. Gene flow is increasingly recognized as an important macroevolutionary process. The many mechanisms that contribute to gene flow (*e.g.,* introgression, hybridization, lateral gene transfer) uniquely affect the diversification of dynamics of species, making it important to be able to account for these idiosyncrasies when constructing phylogenetic models. Existing phylogenetic-network simulators for macroevolution are limited in the ways they model gene flow.
2. We present SiPhyNetwork, an R package for simulating phylogenetic networks under a birth-death-hybridization process.
3. Our package unifies the existing birth-death-hybridization models while also extending the toolkit for modeling gene flow. This tool can create patterns of reticulation such as hybridization, lateral gene transfer, and introgression.
4. Specifically, we model different reticulate events by allowing events to either add, remove, or keep constant the number of lineages. Additionally, we allow reticulation events to be trait-dependent, creating the ability to model the expanse of isolating mechanisms that prevent gene flow. This tool makes it possible for researchers to model many of the complex biological factors associated with gene flow in a phylogenetic context.

1 Introduction

Interspecific gene flow—the movement of genetic material across species boundaries—is observed throughout the tree of life (Mallet et al., 2016). Interspecific gene flow (gene flow hereafter) processes such as the transmission of

genes without vertical inheritence to parents (lateral-gene transfer), interbreeding between species (hybridization), and backcrossing between hybrids and their parental lineages (introgression), can operate across wide ranges of both genetic and taxonomic scales. These dynamic processes can facilitate the sharing of genetic material as small as single genes or as large as whole chromosomes or genomes. Further, exchanges happen not only between closely related populations and species complexes, but also between organisms from different kingdoms of life. These events can have a profound effect on species and lineages at micro- and macroevolutionary scales, playing a significant role in mimicry complexes (Smith and Kronforst, 2013; Enciso-Romero et al., 2017), invasion ecology (Rhymer and Simberloff, 1996; Viard et al., 2020), insecticide resistance (Norris et al., 2015), and adaptive radiations (Grant and Grant, 2019; Meier et al., 2017, 2019; Moest et al., 2020). Regardless of the mode of gene flow and the many ways in which it may affect reticulate species, it is clear that it is an important factor in shaping macroevolutionary patterns (Stebbins, 1959; Bock, 2010; Taylor and Larson, 2019).

With the increased accessibility and availability of genomic data for lineages across the tree of life, an increasing number of studies have found the signatures of historical reticulation and gene flow (Taylor and Larson, 2019). These studies use a wide range of methods for detecting gene flow and describing the processes responsible for generating patterns of reticulation. The available approaches for untangling past gene flow vary widely in how these events are characterized and the scope and scale to which they can be applied—some methods estimate the presence of gene flow, while others seek to estimate when gene flow occurred and which parts of the genome have reticulate histories (see Payseur and Rieseberg, 2016). Recent advances allow researchers to directly estimate gene flow through phylogenetic network inference, with approaches ranging from parsimony and distance-based methods to model-based likelihood and Bayesian methods (Elworth et al., 2019).

Although the field has made significant progress in the development of phylogenetic-network inference methods, the tools for simulating networks under relevant macroevolutionary processes remain limited. Simulated data are vital for validating the accuracy and examining the performance of statistical methods. Moreover, simulation tools can also be useful in empirical studies for hypothesis testing and evaluating model adequacy. Due to the limited availability of phylogenetic network simulators, network-based simulation studies often rely on using bifurcating trees with randomly added reticulate edges (*e.g.,* Bastide et al., 2018; Hejase et al., 2018), simulating sequence data from empirically derived networks (*e.g.,* Wen et al., 2016), or by using an arbitrary fixed phylogenetic network (*e.g.,* Solís-Lemus and Ané, 2016; Wen and Nakhleh, 2018). While these approaches to creating datasets with known attributes are useful for testing specific scenarios and highlighting core features of methods, these networks are not generated by biologically relevant or stochastic model–based processes, limiting the range of conditions that can be explored and rendering the conclusions less generalizable.

Birth-death processes are often used to describe macroevolutionary patterns (Kendall, 1948; Nee, 2006) and, consequently, are also commonly used in phylogenetic simulators (*e.g.,* Stadler, 2011; Höhna, 2013; Höhna et al., 2015; Hagen and Stadler, 2018). To further model and simulate under important mechanisms of biological systems,

many extensions of the birth-death process have been developed. These extensions include density-dependent diversification (Rabosky and Lovette, 2008), time-dependent rates (Stadler, 2011; Höhna, 2013), lineage age-dependent rates (Hagen and Stadler, 2018), and fossilization (Barido-Sottani et al., 2019). Extensions of the birth-death process even simulate phylogenetic networks by allowing hybridization events (Morin and Moret, 2006; Woodhams et al., 2016; Zhang et al., 2018). Although some macroevolutionary simulators can incorporate gene flow, each simulator makes different assumptions about how reticulation events affect the phylogeny (Figure 1; Table 1).

[Figure 1 about here.]

Simulation tools generate phylogenetic networks using a variety of models such as the coalescent (Arenas and Posada, 2007; Kelleher et al., 2016; McKenzie and Eaton, 2020), genome evolution models (Mallo et al., 2016; Davín et al., 2020), or generate networks based on certain phylogenetic characteristics (Janssen and Liu, 2021). While these processes are often useful for modeling population genetic and genomic processes, birth-death processes are invoked to describe lineage diversification on a macroevolutionary scale. HybridSim (Woodhams et al., 2016), NetGen (Morin and Moret, 2006), and the SpeciesNetwork package in BEAST2 (Zhang et al., 2018) all simulate networks under variants of the birth-death-hybridization process, which models how lineages speciate, go extinct, and hybridize as a stochastic process. However, it is important to recognize the different model assumptions about the diversification process and different conceptualizations of hybridization found in each of the available simulators (Table 1). For example, extinction has a considerable effect on shaping biodiversity yet only NetGen explicitly models extinction events while simulating phylogenetic networks. Although, other simulators do not model the loss of species, they have additional flexibility in how they handle gene flow events by having different types of gene flow and allowing parental lineages to differentially contribute to the hybrid species. The disparity of event types and how certain events are modeled between simulators, in turn, make the evaluation of patterns in networks challenging, since the distribution of generated networks is dependent on the inputs and assumptions of available simulation software. As such, an important addition to the present toolkit is a simulator capable of generating phylogenetic networks from a unified framework, allowing for more direct comparisons between the effects of these assumptions on generated networks.

The evolution of morphological characters and other phenotypic traits can additionally lead to reproductive isolation under a variety of pre-zygotic and post-zygotic mechanisms, and consequently, act as barriers to to hybridization (Grant, 1981; Soltis and Soltis, 2009; Abbott et al., 2013). For example, studies have described traits imposing reproductive isolation in lice where body size differences cause mechanical isolation (Villa et al., 2019), butterflies selectively mating with individuals that match their mimetic coloration (Dincă et al., 2013), or many plant species where chromosomal rearrangements create post-pollination barriers (Baack et al., 2015). Even viable hybrids may become excluded on macroevolutionary scales due to low fitness of subsequent generations. This phenomenon, termed hybrid breakdown (Grant, 1981; Soltis and Soltis, 2009), can occur if the hybrid has an intermediate trait value and is unable to find a niche different from the parental lineages, resulting in lower fitness.

Although HybridSim is capable of modulating the amount of hybridization in relation to the genetic distance between lineages, no tools currently account for the effect that certain morphological characters can have on the ability for lineages to hybridize.

Here we present the R package SiPhyNetwork that enables phylogenetic network simulation under a range of biological scenarios, and extends the currently available tools by:

1. modeling the different ways that reticulation can affect lineages during gene flow events,
2. allowing simulations to have trait-dependent hybridization,
3. adapting many common tree simulation features and utility functions for networks (*e.g.,* incomplete lineage sampling, complete versus extant only phylogeny, sampling under the generalized sampling approach of Hartmann et al., 2010),
4. unifying many unique model features from other model simulators in one package (*i.e.,* asymmetric inheritance and genetic distance dependent hybridization as seen in HybridSim), and
5. providing functions for manipulating and classifying phylogenetic networks.

Although reticulation on a macroevolutionary scale is typically described with hybridization events, SiPhyNetwork does not make any specific assumptions about the mechanism of gene flow. Consequently, SiPhyNetwork can model other reticulate processes like introgression or lateral gene transfer. Overall, we sought to provide a framework for simulating evolutionary histories under a multitude of reticulate mechanisms found across the Tree of Life. We believe that this work will enable researchers to test a wide array of hypotheses about reticulate macroevolution.

[Table 1 about here.]

2 Model Components and Implementation

SiPhyNetwork is an R (R Core Team, 2022) package that simulates phylogenetic networks under a birth-death-hybridization process. Our implementation is a generalization of the constant-rate birth-death-hybridization process that allows hybridization to either add a lineage (lineage generative), remove a lineage (lineage degenerative), or keep the number of lineages constant (lineage neutral)—unifying the models of Woodhams et al. (2016) and Zhang et al. (2018) in a single framework. These types of hybridization events are also denoted m-type, y-type, and n-type reticulations respectively (Janssen and Liu, 2021). We additionally allow trait-dependent hybridization and genetic distance–dependent hybridization. SiPhyNetwork is available on the Comprehensive R Archive Network (CRAN). Alternatively, the source code and installation instructions for the development version can be accessed at `https://github.com/jjustison/SiPhyNetwork`.

Network simulation using SiPhyNetwork relies on three core functions: `sim.bdh.age()`, `sim.bdh.taxa.ssa()`, and `sim.bdh.taxa.gsa()`, collectively referred to as the `sim.bdh()` functions. All three functions use the same simulation algorithm but have different stopping conditions (discussed below). With the exception of arguments setting the stopping condition, each simulation function takes the same set of arguments to specify the model (Table 2). These functions generate `evonet` objects from the `ape` software package (Paradis and Schliep, 2019), which are themselves extensions of the `phylo` objects used for phylogenetic trees. These objects can then be stored in the extended Newick format (Cardona et al., 2008) so they can be used for downstream macroevolutionary analyses (*e.g.,* Solís-Lemus et al., 2017; Bastide et al., 2018) or visualization (*e.g.,* Vaughan, 2017; Schliep et al., 2021). In the following sections, we demonstrate how the arguments are used for each component of the birth-death-hybridization model. Further details about the R implementation and examples can be found in the "SiPhyNetwork Introduction" vignette that is released with the package and included in the supplementary materials.

[Table 2 about here.]

2.1 Diversification Process

We model the branching process of the phylogeny with speciation, extinction, and hybridization events. The lineage diversification process has exponentially distributed waiting times for the events, with constant-rate parameters λ for speciation, μ for extinction, and ν for hybridization. Since hybridization requires two lineages, we consider the rate of hybridization (ν) on each species pair, whereas speciation (λ) and extinction (μ) are rates on each lineage. For a phylogeny with N taxa, a rate on each species pair means that hybridization events occur at an effective rate of $\binom{N}{2}\nu$. The overall waiting time until the next event of the birth-death-hybridization process is exponentially distributed with rate

$$N\lambda + N\mu + \binom{N}{2}\nu,$$

where the probability for each event is weighted by its effective rate for N taxa (*i.e.,* $N\lambda$ for speciation, $N\mu$ for extinction, and $\binom{N}{2}\nu$ for hybridization). In SiPhyNetwork users specify a value for each rate (`lambda`, `mu`, and `nu`) by providing values in the `sim.bdh()` functions.

For each hybridization event, we denote the genetic contributions of the two parental lineages—also called inheritance probabilities—as γ and $1-\gamma$. Inheritance probabilities are drawn from a user-defined distribution that draws values from 0 to 1, allowing for asymmetric inheritance, where one parent contributes more genetic material, and broad flexibility to match prior beliefs about gene flow. For example, supplying a $\text{Beta}(10, 10)$ distribution can model hybrid speciation where inheritance probabilities are largely equal, while a $\text{Beta}(0.1, 0.1)$ would reflect introgression where one parental lineage often contributes a larger proportion of genetic material (Figure 2). In SiPhyNetwork, users supply a function for the `hyb.inher.fxn` argument in the `sim.bdh()` functions to draw inheritance probabilities at each hybridization event. There are several helper functions in SiPhyNetwork

that create functions for inheritance probabilities, as shown in the example below.

```
## Example Inheritance proportion sampling distributions
inheritance.fxn1 <- make.beta.draw(10,10)
inheritance.fxn2 <- make.beta.draw(0.1,0.1)
inheritance.fxn3 <- make.uniform.draw()
inheritance.fxn4 <- make.categorical.draw(inheritances= c(0.2,0.5,0.8),
                                          weights = c(0.5,0.3,0.2))
phy<- sim.bdh.age(age=2, numbsim=20,
                  lambda=1, mu=0.2, nu=0.25,
                  hybprops = c(1/3,1/3,1/3),
                  hyb.inher.fxn = inheritance.fxn1,
                  complete=FALSE)
```

Additionally, users can create their own functions to sample inheritance probabilities when hybridization events occur. The supplied function should take no arguments and return a number between 0 and 1.

```
inheritance.fxn5 <- function(){ ##inheritance is equally either 0.1, 0.5, or 0.7
    return(sample(c(0.1,0.5,0.7),1))
}
inheritance.fxn6 <-function(){ ##always equal inheritance
    return(0.5)
}
```

[Figure 2 about here.]

2.2 Hybridization Type and the Effect on the Number of Lineages

SiPhyNetwork has a versatile system for simulating many modes of gene flow by allowing for reticulation events with different macroevolutionary patterns (see Figure 1). We denote each type of event by the net change in the number of lineages as a result of the hybridization: lineage generative when gaining a lineage, lineage neutral when maintaining the same number of lineages, and lineage degenerative for when a lineage is lost. Each type of hybridization imposes different time constraints on the parent nodes (orange circles in Figure 1) that lead to the reticulate node. Both parental nodes co-occur with the reticulation for lineage generative events, only one parent occurs at the same time as the reticulate event for lineage-neutral events, and there are no time co-occurrence constraints for lineage-degenerative events. When a hybridization event occurs, it is either lineage generative, lineage neutral, or lineage degenerative, with probabilities ρ_+, ρ_0, and ρ_-, respectively.

Users have the ability to specify the probabilities for each macroevolutionary pattern, giving the flexibility to model various gene flow types and microevolutionary mechanisms. Although we do not make mechanistic assumptions about how gene flow occurs, certain processes may be better at describing a given reticulate pattern. For example, modeling hybrid speciation with lineage generative hybridization would be appropriate due to the

creation of a new hybrid lineage on the phylogeny. However, lineage-neutral and lineage-degenerative hybridization may also be valid models for hybrid speciation in the cases where genetic swamping occurs (Todesco et al., 2016) or in the presence of ghost lineages (Ottenburghs, 2020; Tricou et al., 2022). Additionally, lineage-neutral hybridization could be used to model cases of introgression or lateral-gene transfer, in which gene flow occurs but no new lineages are produced.

Existing phylogenetic network simulators allow different subsets of these reticulation patterns (Table 1). Netgen solely considers lineage-generative hybridization (Morin and Moret, 2006), HybridSim considers lineage-generative and lineage-neutral hybridization, with the latter being termed 'introgression' (Woodhams et al., 2016), and the SpeciesNetwork package considers solely lineage-degenerative hybridization (Zhang et al., 2018).

Hybrid-type probabilities are modeled in SiPhyNetwork by providing a vector of probabilities for each pattern of hybridization. Each of the elements in the vector corresponds to the probability that hybridization is generative, neutral, or degenerative, respectively. The vector is given in the `hybprops` argument of the `sim.bdh` functions. Below we show examples of different specifications for the hybrid-type proportions.

```
##Example  Hybridization Proportions

##All types equally likely
prop1 <- c(1/3, ##Lineage Generative
           1/3, ##Lineage Neutral
           1/3) ## Lineage Degenerative
##Only lineage neutral hybridization (introgression)
prop2 <- c(0,1,0)
##Proportions skewed towards lineage generative
prop3 <- c(0.5, 0.2, 0.3)

phy<- sim.bdh.age(age=2, numbsim=20,
                  lambda=1, mu=0.2, nu=0.25,
                  hybprops=prop2,
                  hyb.inher.fxn = inheritance.fxn1,
                  complete=FALSE)
```

2.3 Hybridization Success Dependent on Genetic Distance

Gene flow occurs more frequently between closely related lineages than it does for distantly related lineages (Gourbiere and Mallet, 2010; Abbott et al., 2013). We model hybridization success as a function of genetic distance between lineages in SiPhyNetwork using the approach of Woodhams et al. (2016). Hybridization events effectively become a non-stationary Poisson process with respect genetic distance. The hybridization rate changes as a function

of the genetic distance d_{ij} between taxa i and j at a given time:

$$\nu(d_{ij}),$$

where the relationship between hybridization and genetic distance is user-specified. However, in practice we use the thinning of a Poisson process to break this into two steps: (1) hybridization events between a given species pair are proposed as part of a Poisson process with rate ν and (2) proposed hybridization events are then successful with a probability that is proportional to the genetic distance between the species pair. Successful events are added to the phylogeny while unsuccessful hybridization attempts are not. A genetic distance matrix is maintained throughout the simulation and is updated at each event during the forward-in-time simulation to accurately reflect genetic distances at any given point in time. The genetic distance between two taxa i and j is the total length of edges that are not shared on the path from each taxon to the root (or a weighted summation of each path if the taxon in question has hybrid ancestry). Formally, we assume a strict molecular clock where the genetic distance at a given time is denoted as:

$$d_{ij} = \sum_{\substack{p_i \in \mathcal{P}_i \\ p_j \in \mathcal{P}_j}} \left(\prod_{e \in p_i} \gamma_e \right) \left(\prod_{e \in p_j} \gamma_e \right) \sum_{e \notin p_i \cap p_j} \ell_e \ ,$$

where $\mathcal{P}_i$ denotes the set of paths from the taxon i to the root, γ_e is the inheritance probability of the associated edge e, and ℓ_e is the edge length of e. This formulation is identical to the covariance computation of Bastide et al. (2018) with the exception that we take the sum of edge lengths that are not shared across paths, instead of taking the edges that are shared.

In SiPhyNetwork genetic-distance dependence is modeled by providing a genetic-distance function for the `hyb.rate.fxn` argument. Users have the flexibility to define any arbitrary function that relates hybridization success to genetic distance. This function takes the genetic distance as an argument and should return a number that represents the probability of hybridization success, as shown in the example below.

```
##Hybridization fails if the distance is greater than 2.5
hyb.success1 <- function(distance){
    if(distance>=2.5){
        return(0)
    } else{
        return(1)
    }
}
```

Additionally, we have implemented the same decreasing functions as Woodhams et al. (2016), *i.e.,* linear

decay, exponential decay, snowballing decay, and polynomial decay:

$$\text{Linear Decay: } f(d_{ij}) = max\left\{0, 1 - \frac{d_{ij}}{t}\right\}$$
$$\text{Polynomial Decay: } f(d_{ij}) = max\left\{0, 1 - \frac{d_{ij}}{t^s}\right\}$$
$$\text{Snowball Decay: } f(d_{ij}) = e^{-\frac{d_{ij}^2}{t}}$$
$$\text{Exponential Decay: } f(d_{ij}) = e^{-\frac{d_{ij}^s}{t}}$$

Here, s and t are values set by the user that are used to affect the shape and rate of decay for each function. We have several helper functions to create these genetic-distance dependence functions, as shown below

```
hyb.success2<-make.exp.decay(t=1,s=1)
hyb.success3<-make.linear.decay(threshold = 1)
hyb.success4<-make.stepwise(probs = c(1,0.5,0),distances = c(0.25,0.75,Inf))
hyb.success5<-make.polynomial.decay(threshold = 1,degree = 2)
```

2.4 Trait-Dependent Hybridization

In SiPhyNetwork, we implemented a general framework for modeling the complex interplay between successful hybridization and trait evolution. Our trait-dependent hybridization model has three components: a trait-evolution model, a model for trait inheritance in hybrid lineages, and rules that describe how hybridization success depends on trait values (Figure 3). The model of trait evolution specifies how continuous or discrete trait values change over time and has the flexibility to implement a number of trait evolution models (*e.g.,* Brownian motion (Felsenstein, 1985), Ornstein–Uhlenbeck (Lande, 1980), Mk (Pagel, 1994; Lewis, 2001), threshold (Felsenstein, 2005)). The model for trait inheritance specifies how the trait is inherited both at speciation events and at hybridization events. The last component permits the user to define how trait values interact to determine whether hybridization occurs. In nature, both discrete and continuous traits are known to affect rates of hybridization, thus SiPhyNetwork allows either type of phenotypic trait to affect the hybridization potential of different lineages. Likewise, in some systems, traits that are more similar may enhance the likelihood of hybridization (Dincă et al., 2013; Pereira et al., 2014), while in others, opposite trait values create novelty for hybrids to succeed (Vereecken et al., 2010).

Each component is created with user-defined functions to determine how they operate during simulation. This framework offers a great degree of flexibility for modeling biologically realistic trait-dependent hybridization scenarios. For example, one can generate networks of hybridizing lineages modulated by ploidy evolution with both allo- and auto-polylploidization; characters can evolve continuously such that a hybrid can only persist if it avoids hybrid breakdown by occupying a trait space different than that of its parental lineages (Soltis and Soltis, 2009); or hybridization could become less successful as the traits of two lineages become increasingly dissimilar. Thus, SiPhyNetwork is flexible in modeling trait-dependent hybridization by allowing the biologist to tailor the mode to

their particular system.

We model trait-dependent hybridization by supplying the optional argument `trait_model` to the `sim.bdh()` functions. The `trait_model` argument is a list that specifies each component of the trait-dependent hybridization model (Figure 3). The trait model is a named list with the following elements: `initial_states` a value for the initial state of the trait, `hyb.event.fxn` a function to determine how the trait is inherited on the hybrid lineage, `hyb.compatability.fxn` a function to determine whether hybridization can occur based on the trait values, `time.fxn` a function that determines how the traits change over time, and `spec.fxn` a function that determines how the trait is inherited at speciation events. In the example below we implement a model for ploidy evolution that considers autopolyploidy and allopolyploid hybridization. We restrict allopolyploidy events to only occur between lineages with the same ploidy.

```
initial_val<-2 ## The root starts off at 2N

###function for what happens at hybridization event
hyb_e_fxn <- function(parent_states,inheritance){
  ##For allopolyploidy we add the ploidy of both parents
  return(sum(parent_states))
}

##Function for determining whether hybridization occurs
hyb_c_fxn <-function(parent_states,hybrid_state){
  ##Hybridization occurs only when the ploidy is the same
  return(parent_states[1]==parent_states[2])
}

##Function for how the trait changes over time
t_fxn <- function(trait_states,timestep){
  ##We assume that autopolyploidy occur exponentially with rate lambda
  lambda<- 2 ##Rate of autopolyploidy

  ##The number of autopolyploidy events that occur on each lineage over the timestep
  nevents<-rpois(length(trait_states),timestep)

  ##each event doubles the ploidy
  new_states<- trait_states * (2^nevents)
  return(new_states)
}

##Function for how the trait changes at speciation events
s_fxn <-function(tip_state){
  ##Ploidy doesn't change at speciation events.
```

```
  ##Both daughter lineages have the same ploidy as the parent
  return(c(tip_state,tip_state))
}

trait_model<-make.trait.model(initial_states = initial_val,
                              hyb.event.fxn = hyb_e_fxn,
                              hyb.compatibility.fxn = hyb_c_fxn,
                              time.fxn = t_fxn,
                              spec.fxn = s_fxn)

trait_nets <-sim.bdh.age(age=2,numbsim=10,
                         lambda=1,mu=0.2,
                         nu=0.25, hybprops = hybrid_proportions,
                         hyb.inher.fxn = inheritance.fxn,
                         trait.model = trait_model)
```

More information about model capability and specific implementations can be found in the "Introduction" vignette (Supplemental files).

[Figure 3 about here.]

2.5 Extant-Only and Incomplete Sampling

[Figure 4 about here.]

Typically, phylogenetic networks do not include all extant taxa or they lack fossil specimens that can provide information about extinct lineages. We can model incomplete lineage sampling by pruning away unsampled lineages (Figure 4), leaving what is often referred to as the reconstructed or sampled phylogenetic network (Gernhard, 2008; Stadler, 2009). Indeed, it is necessary to account for incomplete sampling as it affects expected branch-length distributions (Nee et al., 1994; Stadler, 2008). Producing an extant-only phylogeny is a common feature of phylogenetic tree simulators, but not available in current phylogenetic network simulators. We extend these features to phylogenetic networks, both as a core part of phylogenetic network simulation and as a *post-hoc* operation on phylogenetic networks with utility functions. SiPhyNetwork models these processes in the `sim.bhd()` functions by setting `complete=FALSE` to eliminate all extinct taxa and setting `frac` to less than 1 for incomplete sampling of extant taxa (Figure 4). If `frac` is less than one, then that proportion of extant taxa will be sampled. Further, if the argument `stochsampling=TRUE`, then each extant taxon will be sampled with probability `frac`. If `stochsampling=FALSE`, then `frac` proportion of taxa will be sampled from the phylogeny, rounded to the nearest whole number.

```
##simulating a complete phylogeny
set.seed(4)
```

```
net <-sim.bdh.age(age=2, numbsim=1,
                  lambda=1, mu=0.2, nu=0.25,
                  hybprops = c(1/3,1/3,1/3),
                  hyb.inher.fxn = make.beta.draw(10,10))
##simulating the same phylogenetic network but only return extant tips
set.seed(4)
nets <-sim.bdh.age(age=2, numbsim=1,
                   lambda=1, mu=0.2, nu=0.25,
                   hybprops = c(1/3,1/3,1/3),
                   hyb.inher.fxn = make.beta.draw(10,10),
                   complete=FALSE)
##The extant only phylogenetic network with incomplete sampling
set.seed(4)
nets <-sim.bdh.age(age=2, numbsim=1,
                   lambda=1, mu=0.2, nu=0.25,
                   hybprops = c(1/3,1/3,1/3),
                   hyb.inher.fxn = make.beta.draw(10,10),
                   complete=FALSE, frac=0.7)
##The extant only network with each extant taxa getting sampled with probability 0.7
set.seed(4)
nets <-sim.bdh.age(age=2, numbsim=1,
                   lambda=1, mu=0.2, nu=0.25,
                   hybprops = c(1/3,1/3,1/3),
                   hyb.inher.fxn = make.beta.draw(10,10),
                   complete=FALSE, frac=0.7, stochsampling=T)
```

2.6 Sampling Strategies

Users have two options for defining a simulation's stopping condition, where the simulation ends once (1) a specified time or (2) a specified number of taxa is reached. Traditionally, phylogenies simulated to a number of taxa allow the process to continue until first reaching N taxa, known as the simple sampling approach (SSA). However, this approach does not correctly sample to a number of taxa while assuming a uniform prior on tree ages, and doing so is a not a trivial task (Hartmann et al., 2010; Stadler, 2011).

We extend the generalized sampling approach (GSA) for generating birth-death trees that was introduced by Hartmann et al. (2010) to phylogenetic network simulation, which correctly samples networks with a specified number of taxa under the birth-death-hybridization process. Briefly, if N taxa are desired under the GSA, the simulation process will continue until reaching M taxa, then phylogenies are uniformly sampled from periods with N taxa. A sufficiently large value of M (*i.e.,* $M >> N$) should be chosen such that the probability of the process returning to $\leq N$ taxa is small. The function `sim.bdh.age()` simulates the process from the origin until a specified age, while `sim.bdh.taxa.ssa()` and `sim.bdh.taxa.gsa()` are used to simulate to a specified number

of taxa under the SSA and GSA approaches, respectfully. Birth-death simulations can routinely go extinct before reaching a stopping condition or never reach a desired number of species in a tractable amount of time under certain parameterizations (*e.g.,* $\mu > \lambda$). Similarly, for the birth-death-hybridization process, the specific combinations of λ, μ, ν values will affect the probability that a simulation reaches its stopping condition.

3 Discussion

SiPhyNetwork brings much of the currently available phylogenetic-network simulation functionality into a single R package, while also extending existing models by considering various macroevolutionary patterns of reticulation and allowing for trait-dependent hybridization. The different types of gene flow (lineage generative, neutral, and degenerative) have received some attention, although primarily through the context of the timing of hybridization events (Hibbins and Hahn, 2019; Flouri et al., 2020). The constraints produced by each gene-flow type pose an interesting yet challenging problem for inference (Hibbins and Hahn, 2022). Since each type of hybridization necessitates a different number of speciation events to explain the same number of lineages, over-attributing a specific type of hybridization likely would lead to bias in diversification-rate estimates. Additionally, both sampling only extant taxa (Nee et al., 1994) and incomplete sampling (Stadler, 2008, 2009), are known change our expectations about the birth-death process and resulting distributions of bifurcating trees. However, it is not well characterized how these processes change our expectations of the birth-death-hybridization process. In fact, incomplete sampling and the presence of ghost lineages can make it particularly difficult to infer the correct reticulate pattern (Ottenburghs, 2020; Tricou et al., 2022). Additionally, failure to sample parental lineages can remove the node co-occurrence constraint in the extant-only phylogeny, making it appear as another type of hybridization when compared to the complete phylogeny (Figure 4).

The birth-death-hybridization process and other biological extensions may be able to explain certain macroevolutionary patterns. First, ancient gene flow is frequently found in empirical studies (Meier et al., 2017; Pavón-Vázquez et al., 2021; Zhang et al., 2021). Yet, under the simple birth-death-hybridization process, ancient gene flow should be rare compared to contemporary gene flow due to there being fewer species and the effective hybridization rate scaling more quickly per taxon than speciation or extinction. Hybridization dependent on genetic distance or certain characteristics may make the effective hybridization rate scale more slowly to explain the pattern of ancient gene flow. Additionally, high lineage-degenerative hybridization may have the ability to explain the slowdown in lineage accumulation, often attributed to density-dependent diversification (Rabosky and Lovette, 2008) or time-dependent diversification (Hagen and Stadler, 2018). In this case, lineage accumulation slows down because the rate of lineage-degenerative hybridization scales more quickly with the number of taxa than the rate of speciation. Eventually the lineage-degenerative hybridization would reduce the net-diversification rate to zero until reaching and revolving around some steady state of taxa.

SiPhyNetwork is a tool that facilitates our understanding of patterns of reticulate diversification. Further, the birth-death-hybridization process has many unique properties that we have only begun to explore and this work allows further characterization by sampling from the distribution of phylogenetic networks under this macroevolutionary process. Moreover, SiPhyNetwork provides a framework to test and validate inference methods under a stochastic and biologically informed model that accounts for many gene flow processes.

4 Authors' Contributions

Joshua A. Justison, Claudia Solis-Lemus, and Tracy A. Heath designed the models. Joshua A. Justison implemented the methods. Joshua A. Justison and Claudia Solis-Lemus tested the methods. Joshua A. Justison, Claudia Solis-Lemus, and Tracy A. Heath wrote the manuscript.

5 Acknowledgements

We would like to thank folks from the Heath lab and Solis-Lemus lab for helpful comments and discussion while developing this tool. We would additionally like to thank Cecile Ane and the many other users that submitted bug reports and provided feedback on the software in its early states. Lastly, we would like to thank the editors, Florian C. Boucher, and three other anonymous reviewers for helpful comments and review. This work was partially funded by the National Science Foundation [DEB-2144367 to CSL].

6 Conflict of Interest Statement

We have no conflicts of interest to declare.

7 Data Accessibility Statement

SiPhyNetwork GPL v3 open-source license held by Joshua Justison, Claudia Solis-Lemus and, Tracy A. Heath. SiPhyNetwork is tested on current and future versions of R. All code is avaliable on Github: https://github.com/jjustison/SiPhyNetwork and on Zenodo (Justison, 2023).

References

Abbott, R., D. Albach, S. Ansell, J. W. Arntzen, S. J. Baird, N. Bierne, J. Boughman, A. Brelsford, C. A. Buerkle, R. Buggs, et al. 2013. Hybridization and speciation. Journal of Evolutionary Biology 26:229–246.

Arenas, M. and D. Posada. 2007. Recodon: coalescent simulation of coding dna sequences with recombination, migration and demography. BMC Bioinformatics 8:1–4.

Baack, E., M. C. Melo, L. H. Rieseberg, and D. Ortiz-Barrientos. 2015. The origins of reproductive isolation in plants. New Phytologist 207:968–984.

Barido-Sottani, J., W. Pett, J. E. O'Reilly, and R. C. M. Warnock. 2019. FossilSim: An R package for simulating fossil occurrence data under mechanistic models of preservation and recovery. Methods in Ecology and Evolution 10:835–840.

Bastide, P., C. Solís-Lemus, R. Kriebel, K. William Sparks, and C. Ané. 2018. Phylogenetic comparative methods on phylogenetic networks with reticulations. Systematic Biology 67:800–820.

Bock, R. 2010. The give-and-take of DNA: horizontal gene transfer in plants. Trends in Plant Science 15:11–22.

Cardona, G., F. Rosselló, and G. Valiente. 2008. Extended newick: it is time for a standard representation of phylogenetic networks. BMC Bioinformatics 9:1–8.

Davín, A. A., T. Tricou, E. Tannier, D. M. de Vienne, and G. J. Szöllosi. 2020. Zombi: a phylogenetic simulator of trees, genomes and sequences that accounts for dead linages. Bioinformatics 36:1286–1288.

Dincă, V., C. Wiklund, V. A. Lukhtanov, U. Kodandaramaiah, K. Norén, L. Dapporto, N. Wahlberg, R. Vila, and M. Friberg. 2013. Reproductive isolation and patterns of genetic differentiation in a cryptic butterfly species complex. Journal of Evolutionary Biology 26:2095–2106.

Elworth, R. A., H. A. Ogilvie, J. Zhu, and L. Nakhleh. 2019. Advances in computational methods for phylogenetic networks in the presence of hybridization. Pages 317–360 *in* Bioinformatics and Phylogenetics. Springer.

Enciso-Romero, J., C. Pardo-Díaz, S. H. Martin, C. F. Arias, M. Linares, W. O. McMillan, C. D. Jiggins, and C. Salazar. 2017. Evolution of novel mimicry rings facilitated by adaptive introgression in tropical butterflies. Molecular Ecology 26:5160–5172.

Felsenstein, J. 1985. Phylogenies and the comparative method. The American Naturalist 125:1–15.

Felsenstein, J. 2005. Using the quantitative genetic threshold model for inferences between and within species. Philosophical Transactions of the Royal Society B: Biological Sciences 360:1427–1434.

Flouri, T., X. Jiao, B. Rannala, and Z. Yang. 2020. A Bayesian implementation of the multispecies coalescent model with introgression for phylogenomic analysis. Molecular Biology and Evolution 37:1211–1223.

Gernhard, T. 2008. The conditioned reconstructed process. Journal of Theoretical Biology 253:769–778.

Gourbiere, S. and J. Mallet. 2010. Are species real? The shape of the species boundary with exponential failure, reinforcement, and the "missing snowball". Evolution 64:1–24.

Grant, P. R. and B. R. Grant. 2019. Hybridization increases population variation during adaptive radiation. Proceedings of the National Academy of Sciences 116:23216–23224.

Grant, V. 1981. Plant Speciation. Columbia University Press.

Hagen, O. and T. Stadler. 2018. TreeSimGM: Simulating phylogenetic trees under general bellman–harris models with lineage-specific shifts of speciation and extinction in R. Methods in Ecology and Evolution 9:754–760.

Hartmann, K., D. Wong, and T. Stadler. 2010. Sampling trees from evolutionary models. Systematic Biology 59:465–476.

Hejase, H. A., N. VandePol, G. M. Bonito, and K. J. Liu. 2018. Fastnet: fast and accurate statistical inference of phylogenetic networks using large-scale genomic sequence data. Pages 242–259 *in* RECOMB International conference on Comparative Genomics Springer.

Hibbins, M. S. and M. W. Hahn. 2019. The timing and direction of introgression under the multispecies network coalescent. Genetics 211:1059–1073.

Hibbins, M. S. and M. W. Hahn. 2022. Phylogenomic approaches to detecting and characterizing introgression. Genetics 220:iyab173.

Höhna, S. 2013. Fast simulation of reconstructed phylogenies under global time-dependent birth–death processes. Bioinformatics 29:1367–1374.

Höhna, S., M. R. May, and B. R. Moore. 2015. TESS: an R package for efficiently simulating phylogenetic trees and performing Bayesian inference of lineage diversification rates. Bioinformatics 32:789–791.

Janssen, R. and P. Liu. 2021. Comparing the topology of phylogenetic network generators. Journal of Bioinformatics and Computational Biology 19:2140012.

Justison, J. 2023. jjustison/siphynetwork: Siphynetwork.

Kelleher, J., A. M. Etheridge, and G. McVean. 2016. Efficient coalescent simulation and genealogical analysis for large sample sizes. PLoS Computational Biology 12:e1004842.

Kendall, D. G. 1948. On the generalized "birth-and-death" process. Annals of Mathematical Statistics 19:1–15.

Lande, R. 1980. Genetic variation and phenotypic evolution during allopatric speciation. The American Naturalist 116:463–479.

Lewis, P. O. 2001. A likelihood approach to estimating phylogeny from discrete morphological character data. Systematic Biology 50:913–925.

Mallet, J., N. Besansky, and M. W. Hahn. 2016. How reticulated are species? BioEssays 38:140–149.

Mallo, D., L. De Oliveira Martins, and D. Posada. 2016. SimPhy: Phylogenomic simulation of gene, locus, and species trees. Systematic Biology 65:334–344.

McKenzie, P. F. and D. A. R. Eaton. 2020. ipcoal: An interactive Python package for simulating and analyzing genealogies and sequences on a species tree or network. Bioinformatics 36:4193–4196.

Meier, J. I., D. A. Marques, S. Mwaiko, C. E. Wagner, L. Excoffier, and O. Seehausen. 2017. Ancient hybridization fuels rapid cichlid fish adaptive radiations. Nature Communications 8:1–11.

Meier, J. I., R. B. Stelkens, D. A. Joyce, S. Mwaiko, N. Phiri, U. K. Schliewen, O. M. Selz, C. E. Wagner, C. Katongo, and O. Seehausen. 2019. The coincidence of ecological opportunity with hybridization explains rapid adaptive radiation in Lake Mweru cichlid fishes. Nature Communications 10:1–11.

Moest, M., S. M. Van Belleghem, J. E. James, C. Salazar, S. H. Martin, S. L. Barker, G. R. P. Moreira, C. Mérot, M. Joron, and N. J. Nadeau. 2020. Selective sweeps on novel and introgressed variation shape mimicry loci in a butterfly adaptive radiation. PLoS Biology 18:e3000597.

Morin, M. M. and B. M. E. Moret. 2006. NetGen: generating phylogenetic networks with diploid hybrids. Bioinformatics 22:1921–1923.

Nee, S. 2006. Birth-death models in macroevolution. Annual Review of Ecology, Evolution, and Systematics 37:1–17.

Nee, S., R. M. May, and P. H. Harvey. 1994. The reconstructed evolutionary process. Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences 344:305–311.

Norris, L. C., B. J. Main, Y. Lee, T. C. Collier, A. Fofana, A. J. Cornel, and G. C. Lanzaro. 2015. Adaptive introgression in an African malaria mosquito coincident with the increased usage of insecticide-treated bed nets. Proceedings of the National Academy of Sciences 112:815–820.

Ottenburghs, J. 2020. Ghost introgression: spooky gene flow in the distant past. BioEssays 42:2000012.

Pagel, M. 1994. Detecting correlated evolution on phylogenies: a general method for the comparative analysis of discrete characters. Proceedings of the Royal Society of London. Series B: Biological Sciences 255:37–45.

Paradis, E. and K. Schliep. 2019. ape 5.0: An environment for modern phylogenetics and evolutionary analyses in R. Bioinformatics 35:526–528.

Pavón-Vázquez, C. J., I. G. Brennan, and J. S. Keogh. 2021. A comprehensive approach to detect hybridization sheds light on the evolution of Earth's largest lizards. Systematic Biology 70:877–890.

Payseur, B. A. and L. H. Rieseberg. 2016. A genomic perspective on hybridization and speciation. Molecular Ecology 25:2337–2360.

Pereira, R. J., F. S. Barreto, and R. S. Burton. 2014. Ecological novelty by hybridization: experimental evidence for increased thermal tolerance by transgressive segregation in *Tigriopus californicus*. Evolution 68:204–215.

R Core Team. 2022. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing Vienna, Austria.

Rabosky, D. L. and I. J. Lovette. 2008. Explosive evolutionary radiations: decreasing speciation or increasing extinction through time? Evolution 62:1866–1875.

Rhymer, J. M. and D. Simberloff. 1996. Extinction by hybridization and introgression. Annual review of Ecology and Systematics 27:83–109.

Schliep, K., M. Vidal-Garcia, C. Solis-Lemus, L. Biancani, E. Ada, and L. F. Henao Diaz. 2021. tanggle: Visualization of Phylogenetic Networks. Https://klausvigo.github.io/tanggle, https://github.com/KlausVigo/tanggle.

Smith, J. and M. R. Kronforst. 2013. Do *Heliconius* butterfly species exchange mimicry alleles? Biology Letters 9:20130503.

Solís-Lemus, C. and C. Ané. 2016. Inferring phylogenetic networks with maximum pseudolikelihood under incomplete lineage sorting. PLoS Genetics 12:e1005896.

Solís-Lemus, C., P. Bastide, and C. Ané. 2017. Phylonetworks: a package for phylogenetic networks. Molecular Biology and Evolution 34:3292–3298.

Soltis, P. S. and D. E. Soltis. 2009. The role of hybridization in plant speciation. Annual Review of Plant Biology 60:561–588.

Stadler, T. 2008. Lineages-through-time plots of neutral models for speciation. Mathematical Biosciences 216:163–171.

Stadler, T. 2009. On incomplete sampling under birth–death models and connections to the sampling-based coalescent. Journal of Theoretical Biology 261:58–66.

Stadler, T. 2011. Simulating trees with a fixed number of extant species. Systematic Biology 60:676–684.

Stebbins, G. L. 1959. The role of hybridization in evolution. Proceedings of the American Philosophical Society 103:231–251.

Taylor, S. A. and E. L. Larson. 2019. Insights from genomes into the evolutionary importance and prevalence of hybridization in nature. Nature Ecology & Evolution 3:170–177.

Todesco, M., M. A. Pascual, G. L. Owens, K. L. Ostevik, B. T. Moyers, S. Hübner, S. M. Heredia, M. A. Hahn, C. Caseys, and D. G. Bock. 2016. Hybridization and extinction. Evolutionary Applications 9:892–908.

Tricou, T., E. Tannier, and D. M. de Vienne. 2022. Ghost lineages highly influence the interpretation of introgression tests. Systematic Biology 71:1147–1158.

Vaughan, T. G. 2017. Icytree: rapid browser-based visualization for phylogenetic trees and networks. Bioinformatics 33:2392–2394.

Vereecken, N. J., S. Cozzolino, and F. P. Schiestl. 2010. Hybrid floral scent novelty drives pollinator shift in sexually deceptive orchids. BMC Evolutionary Biology 10:1–12.

Viard, F., C. Riginos, and N. Bierne. 2020. Anthropogenic hybridization at sea: three evolutionary questions relevant to invasive species management. Philosophical Transactions of the Royal Society B 375:20190547.

Villa, S. M., J. C. Altuna, J. S. Ruff, A. B. Beach, L. I. Mulvey, E. J. Poole, H. E. Campbell, K. P. Johnson, M. D. Shapiro, and S. E. Bush. 2019. Rapid experimental evolution of reproductive isolation from a single natural population. Proceedings of the National Academy of Sciences 116:13440–13445.

Wen, D. and L. Nakhleh. 2018. Coestimating reticulate phylogenies and gene trees from multilocus sequence data. Systematic Biology 67:439–457.

Wen, D., Y. Yu, and L. Nakhleh. 2016. Bayesian inference of reticulate phylogenies under the multispecies network coalescent. PLoS Genetics 12:e1006006.

Woodhams, M. D., P. J. Lockhart, and B. R. Holland. 2016. Simulating and summarizing sources of gene tree incongruence. Genome Biology and Evolution 8:1299–1315.

Zhang, C., H. A. Ogilvie, A. J. Drummond, and T. Stadler. 2018. Bayesian inference of species networks from multilocus sequence data. Molecular Biology and Evolution 35:504–517.

Zhang, D., F. E. Rheindt, H. She, Y. Cheng, G. Song, C. Jia, Y. Qu, P. Alström, and F. Lei. 2021. Most genomic loci misrepresent the phylogeny of an avian radiation because of ancient gene flow. Systematic Biology 70:961–975.

Table 1: Simulation model features present in existing tools employing the birth-death-hybridization process. Reticulation type is denoted as generative (denoted as m-type in Janssen and Liu, 2021), neutral (n-type), and degenerative (y-type), referring to the net gain, stasis, or reduction in the number of lineages from an event, respectively (see Figure 1). A distribution on the inheritance probability γ enables users to specify an arbitrary distribution to determine the inheritance proportions of the parental lineages at hybridization. Hybridization dependence allows the success of hybridization to rely on either genetic distance or on a trait that evolves on the phylogenetic network.

Feature / **Tool**	**Birth**	**Death**	**Reticulation Type**			**Distribution on Inheritance Probability γ**	**Hybridization Dependence**	
			Generative	**Neutral**	**Degenerative**		**Genetic Distance**	**Trait Evolution**
NetGen	✓	✓	✓					
HybridSim	✓		✓	✓		✓	✓	
SpeciesNetwork	✓				✓	✓		
SiPhyNetwork	✓	✓	✓	✓	✓	✓	✓	✓

Table 2: Arguments of simulation functions in SiPhyNetwork

Parameter	**Description**
`age`, `n`, `m`	Stopping conditions
`numsim`	The number of simulation replicates
`lambda`	Speciation rate (λ)
`mu`	Extinction rate (μ)
`nu`	Hybridization rate (ν)
`hybprops`	Hybridization-type proportions
`hyb.inher.fxn`	A function that determines inheritance probabilities
`hyb.rate.fxn`	A function that relates genetic distance to hybridization success
`trait.model`	A list containing a model for trait evolution and hybridization-dependence rules for the traits
`frac`	The sampling fraction of extant species
`stochsampling`	A logical that if `TRUE`, then each extant taxon is sampled with probability `frac`. If `FALSE`, then a constant proportion of `frac` taxa are sampled, rounded to the nearest whole taxon
`twolineages`	A logical that if `TRUE`, starts the process with two lineages that share a common ancestor, else starts the process with one lineage
`complete`	A logical to return the complete or reconstructed network

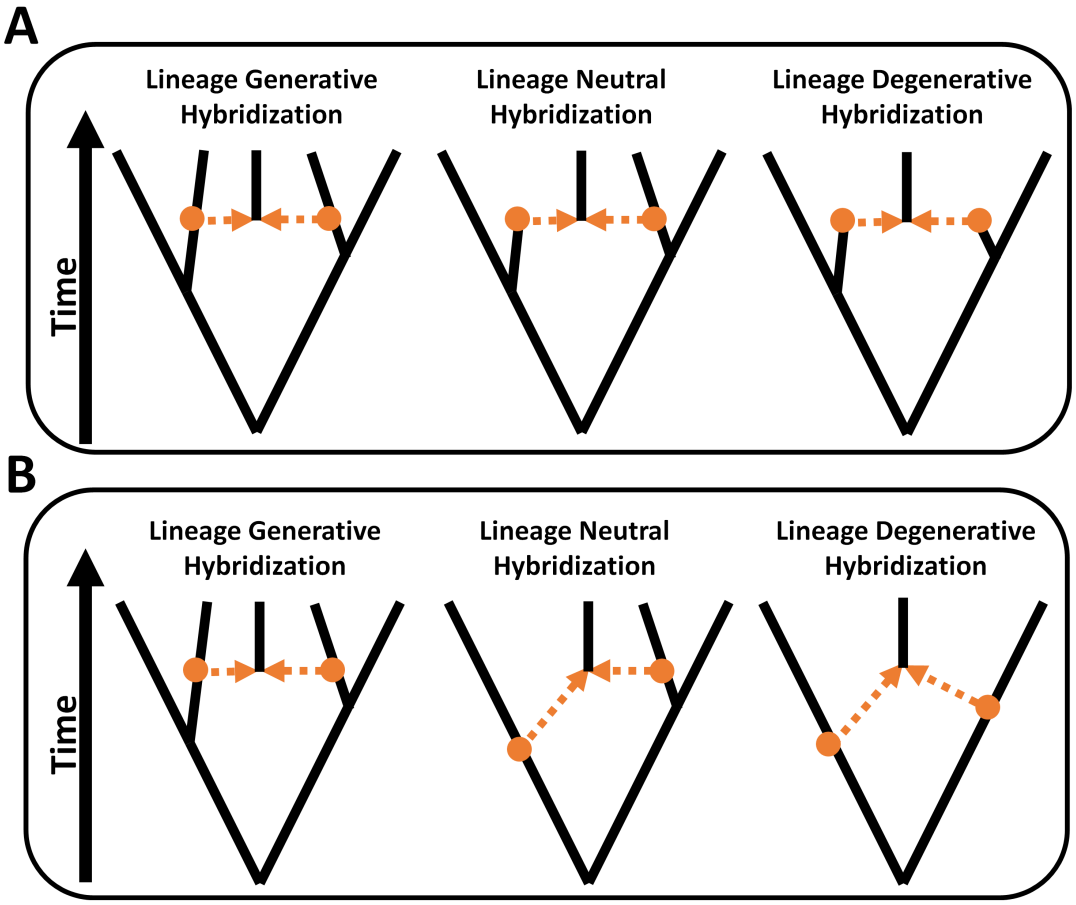

Figure 1: Macroevolutionary patterns of gene flow. Orange circles denote the parental nodes that lead to the reticulate node, while dashed orange arrows indicate the two parental lineages that contribute to gene flow. Lineage generative hybridization (m-type) occurs when a reticulation event results in a gain of one lineage. Lineage neutral hybridization (n-type) results in the a net-zero change in the number of lineages. Lineage degenerative hybridization (y-type) reduces the number of lineages by one. Reticulation events on phylogenetic networks are typically drawn using one of two conventions: A) having all parental nodes lateral to the hybrid node, or B) fusing any in-degree 1 out-degree 1 nodes, potentially placing parent nodes at different times than the hybrid node. Both depictions, however, have the same topological relationships.

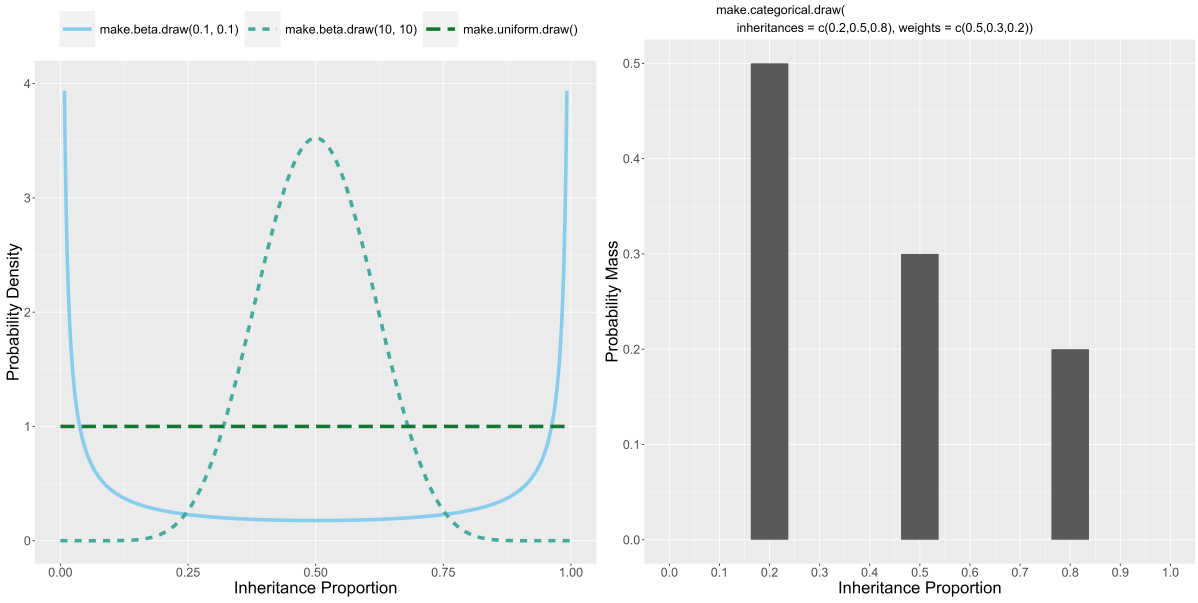

Figure 2: Examples of sampling distributions for inheritance proportions γ and the corresponding functions in R to make the distributions. Inheritance proportions are drawn from the sampling distribution at each hybridization event. Users can specify any distribution or set of values as long as the output is between 0 and 1. The distributions on the left depict continuous sampling distributions on $[0, 1]$ while the distribution on the right only draws values 0.2, 0.5, or 0.8

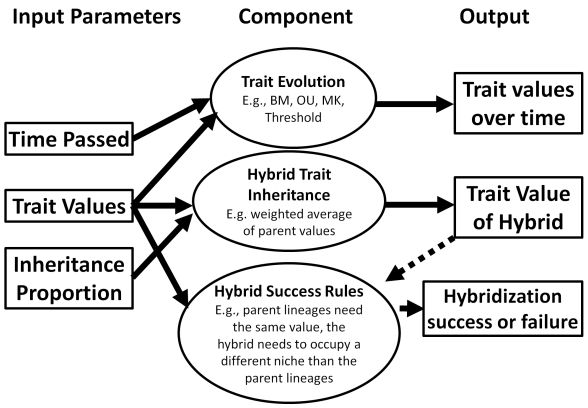

Figure 3: A schematic diagram of the SiPhyNetwork model for trait-dependent hybridization. Each column depicts the model inputs, model components, and outputs of each model, respectively. Arrows leading into each model component denote the needed inputs and the arrows leading out of each model component depict the outputs. The trait value of the hybrid—indicated with a dashed arrow—is a special output that can also be used an optional input for determining hybrid success. The model for trait-dependent hybridization has three sub-components: the trait evolution model, hybrid trait inheritance, and rules for hybridization success. The first component describes how trait values change over time for each lineage. The second models how the hybrid inherits its trait value from its parents. The last component enforces rules that determine whether the hybridization event is successful and occurs on the phylogeny or fails and does not occur. Further, if desired, the last component uses the trait value of the hybrid from the second component as an input for determining hybridization success.

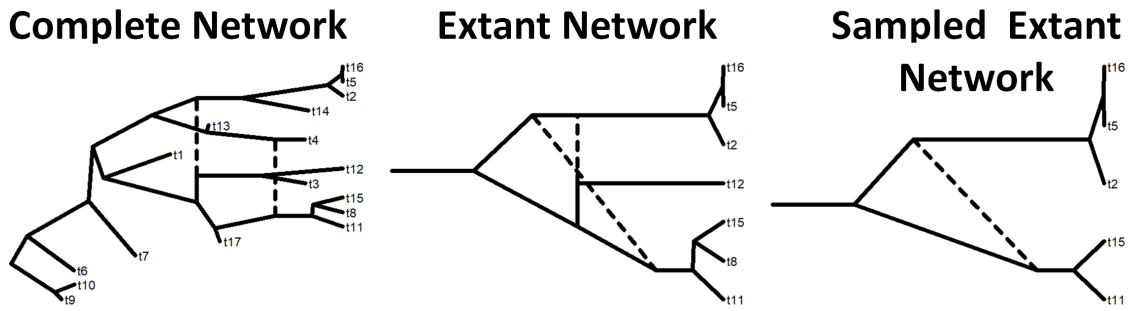

Figure 4: Lineage sampling on a phylogenetic network from SiPhyNetwork. Here we show the same phylogenetic network under three lineage-sampling procedures. The complete network represents the entire simulated history and includes all extant and extinct lineages. The extant network shows only the history of the extant lineages after pruning all extinct lineage. Interestingly, although the complete network has a lineage neutral hybridization at MRCA(t15, t8, t11), the same hybridization appears as a lineage degenerative hybridization in the extant phylogenetic network. The sampled extant network shows the reticulate history of a subset of the extant taxa after pruning all extinct lineages and the extant taxa $t8$ and $t12$ pruned. Without the hybrid lineage $t12$ in the sampled extant network, the lineage-generative hybridization is no longer observable.