LinguaPhylo

A probabilistic model specification language for reproducible phylogenetic analyses

Alexei Drummond, Kylie Chen, Fabio K Mendes, Dong Xie

BEAST: Bayesian evolutionary analysis by sampling trees

Alexei J Drummond*[1,2] and Andrew Rambaut[3]

BMC Evol Biol, 2007

>12700 citations

Bayesian Phylogenetics with BEAUti and the BEAST 1.7

Alexei J. Drummond,*[,1,2] Marc A. Suchard,*[,3,4] Dong Xie,[1,2] and Andrew Rambaut*[,5]

Mol Biol Evol, 2012

>9700 citations

BEAST 2: A Software Platform for Bayesian Evolutionary Analysis

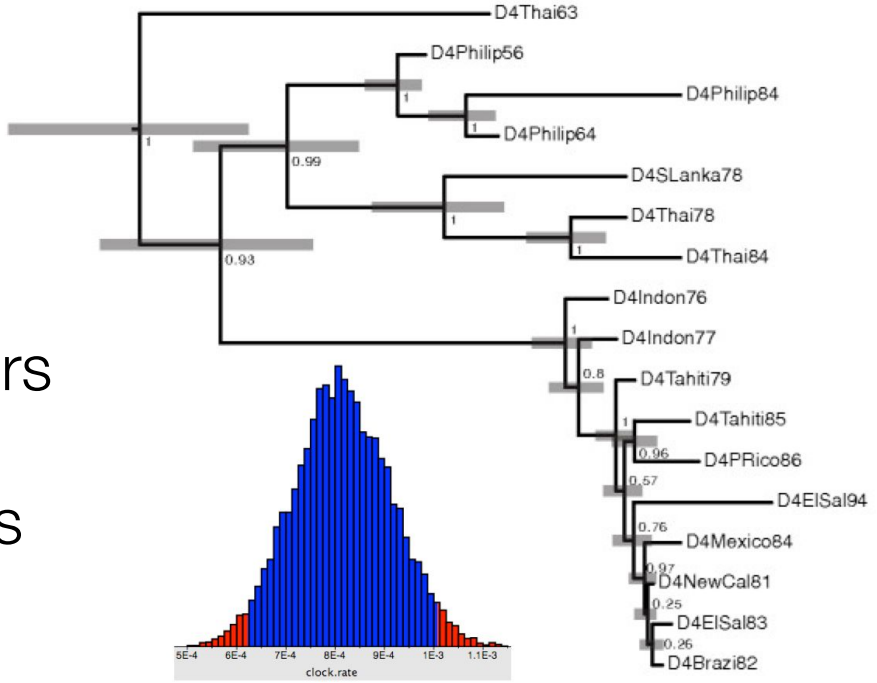
Remco Bouckaert[1*], Joseph Heled[1], Denise Kühnert[1,2], Tim Vaughan[1,3], Chieh-Hsi Wu[1], Dong Xie[1], Marc A. Suchard[4,5], Andrew Rambaut[6], Alexei J. Drummond[1,7*]

PLoS Comp Bio, 2014

>5100 citations

Estimation of

- evolutionary trees
- divergence times
- evolutionary rates
- evolutionary model parameters
- population history
- infectious disease parameters
- migration rates
- more…



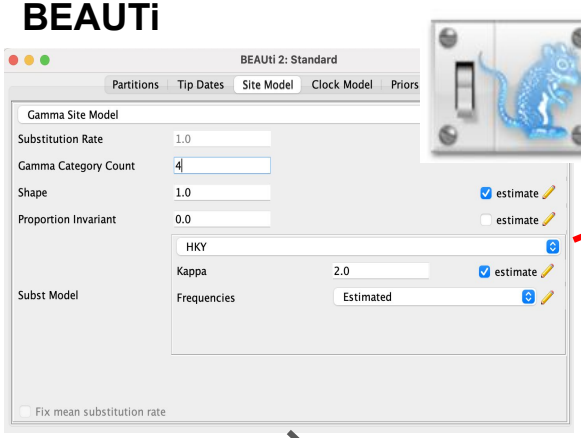
> 35,000 scientific studies have used this software since 2007;
> 2,000 studies last year alone (so far in 2023)

Bayesian phylogenetic inference software

- **MrBayes** (2003-): >79,000 citations
- **BEAST1 + 2** (2007-): >35,000 citations
- PhyloBayes (2007-): >1900 citations
- BayesPhylogenies (2006-): >1000 citations
- BPP (2015?-): >800 citations
- **RevBayes** (2016-): >500 citations
- Phycas (2005?-): >300 citations

Analysis Pipeline for BEAST 2

BEAUTi



BEAST2 XML

```
<state id="State" spec="State">
    <parameter id="lambda" spec="parameter.RealParameter" lower="0.0" name="stateNode">9.591500353792183</parameter>
    <stateNode id="psi" spec="beast.util.TreeParser" IsLabelledNewick="true" newick="(((14:0.06962779760393568,(6:0.05
        <taxonset id="TaxonSet" spec="TaxonSet"> ... </taxonset>
    </stateNode>
</state>
<distribution id="posterior" spec="util.CompoundDistribution">
    <distribution id="prior" spec="util.CompoundDistribution">
        <distribution id="lambda.prior" spec="beast.math.distributions.Prior" x="@lambda">
            <distr id="LogNormalDistributionModel" spec="beast.math.distributions.LogNormalDistributionModel">
                <parameter id="RealParameter" spec="parameter.RealParameter" name="M">3.0</parameter>
                <parameter id="RealParameter1" spec="parameter.RealParameter" name="S">1.0</parameter>
            </distr>
        </distribution>
        <distribution id="YuleModel" spec="beast.evolution.speciation.YuleModel" birthDiffRate="@lambda" tree="@psi"/>
    </distribution>
    <distribution id="likelihood" spec="util.CompoundDistribution">
        <distribution id="D.treeLikelihood" spec="ThreadedTreeLikelihood" tree="@psi">
            <data idref="D"/>
            <siteModel id="SiteModel" spec="SiteModel">
                <substModel id="JukesCantor" spec="JukesCantor"/>
            </siteModel>
            <branchRateModel id="StrictClockModel" spec="beast.evolution.branchratemodel.StrictClockModel">
                <parameter id="RealParameter2" spec="parameter.RealParameter" name="clock.rate">1.0</parameter>
            </branchRateModel>
        </distribution>
    </distribution>
</distribution>
```

Paper

- Method description
- Supplementary info
- Figures
- Estimated quantities in text

Post-processing tools

- Tracer
- TreeAnnotator
- FigTree
- DensiTree
- IcyTree
- etc…

BEAST 2.7

Output files

- Log file
- Trees file

Analysis Pipeline for RevBayes

Rev Script

```
…
### diversification = birth_rate - death_rate
### assume an exponential prior distribution
### and apply a scale proposal
diversification ~ dnExponential(10.0)
moves[mi++] = mvScale(diversification,lambda=1.0,tune=true,weight=3.0)

### turnover = death_rate / birth_rate
### this parameter can only take values between 0 and 1
### use a Beta prior distribution
### and a slide move
turnover ~ dnBeta(2.0, 2.0)
moves[mi++] = mvSlide(turnover,delta=1.0,tune=true,weight=3.0)

### the parameters of the BDP include birth and death rates
### these are deterministic variables of the diversification & turnover
##### create a variable to ensure the rates are always positive (RealPos)
denom := abs(1.0 - turnover)
##### birth_rate = diversification / (1 - turnover)
birth_rate := diversification / denom
##### death_rate = (turnover * diversification) / (1 - turnover)
death_rate := (turnover * diversification) / denom

### rho is the probability of sampling species at the present
### fix this to 0.068, since there are ~147 described species of caniforms (bears,
dogs, mustelids, pinnipeds, etc.)
### and we have sampled 10
rho <- 0.068

### the root age is an independent stochastic node with a lognormal prior
### the lognormal distribution is off-set by the age of the canid fossil in the genus
Hesperocyon
### the observation time of Hesperocyon is ~ 38 Mya
tHesperocyon <- 38.0
### the mean of the lognormal distribution is set to 15 Mya older than the observed
fossil
### when offset by tHesperocyon, this gives a mean root_time of 49 Mya
mean_ra <- 11.0
stdv_ra <- 0.25
### the lognormal distribution is parameterized by mu which is a function of the
mean and standard deviation
mu_ra <- ln(mean_ra) - ((stdv_ra*stdv_ra) * 0.5)
root_time ~ dnLnorm(mu_ra, stdv_ra, offset=tHesperocyon)
…
```

Paper

- **Method description**
- **Supplementary info**
- **Figures**
- **Estimated quantities in text**

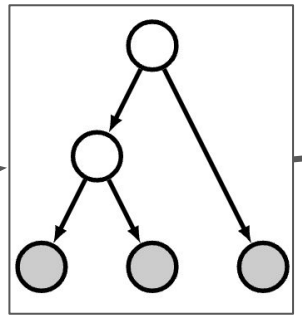
Post-processing tools

- **summarize()**
- **mapTree()**
- **Tracer**
- **etc**

Output files

- **Log file**
- **Trees file**
- **Map tree**
- **etc**

RevBayes



A phylogenetic model

To date the divergence time for *Y. pestis* and nodes within the *Y. pestis* clade we performed Bayesian Markov Chain Monte Carlo simulations using BEAST-2.3.0 (Bouckaert et al., 2014) and the BEAGLE library v2.1.2 (Ayres et al., 2012). We used the codon-partitioned supermatrix that included the two closest *Y. pseudotuberculosis* clades, with unlinked substitution models, GTR+G+I with eight gamma rate categories and unlinked clock models. Dates were set as years ago with the RISE509, RISE505, Justinian and Black Death samples set to 4,761, 3,701, 1,474, and 667 years ago, respectively. All unknown dates were set to 0 years ago. We followed previous work (Cui et al., 2013, Wagner et al., 2014) and applied a lognormal relaxed clock, assuming a constant population size.

https://www.sciencedirect.com/science/article/pii/S0092867415013227

Another phylogenetic model

RAxML was used to construct maximum likelihood phylogenies for each segment (33). The general-time reversible model of nucleotide substitution was assumed, and the rates were assumed to vary according to the discrete Gamma distribution with four rate categories. … Bayesian phylogenetic trees were constructed for each segment using Bayesian Evolutionary Analysis Sampling Trees Software (BEAST) (35). The model of substitution and rate variation in each tree was the same as in the corresponding RAxML tree, with the additional assumption of a log-normally distributed molecular clock. BEAST trees were initiated with the RAxML majority rule consensus tree to reduce the computation time required for the MArcov Chain Monte Carlow (MCMC) sampler to reach the stationary distribution. Maximum clade credibility (MCC) trees were constructed using 2,000 equidistant samples from the final 30% of the posterior sample of trees (n = 25 × 106 total samples) using a python script and tree annotator (35). The resulting MCC trees were partitioned into clades using a median branch length distance threshold of 0.20 (24). Briefly, clades with median root-to-tip distances equal to or less than 20% of the median root-to-tip distances of all clades in the tree were selected as clusters. Clades were then manually merged, if necessary, based on the branching posterior and reported classification (19, 20, 25). Each clade was assigned a unique clade ID.

https://www.pnas.org/content/112/2/548.short

EVOLUTIONARY ECOLOGY

A new time tree reveals Earth history's imprint on the evolution of modern birds

Santiago Claramunt* and Joel Cracraft*

10.1126/sciadv.1501005

The genomic data set was analyzed under a single GTR+ gamma substitution model. Bayesian inference was performed using BEAST 2 (44). Rate heterogeneity across lineages was modeled with a relaxed lognormal clock (39). Priors on rates of substitution and clocks were set to defaults (as in Beauti 2.1). We ran analyses using a birth-death tree prior with incomplete sampling that takes into account the fact that we sampled a small fraction of all avian diversity (Supplementary Materials).

https://advances.sciencemag.org/content/advances/1/11/e1501005.full.pdf

Supplementary Materials for

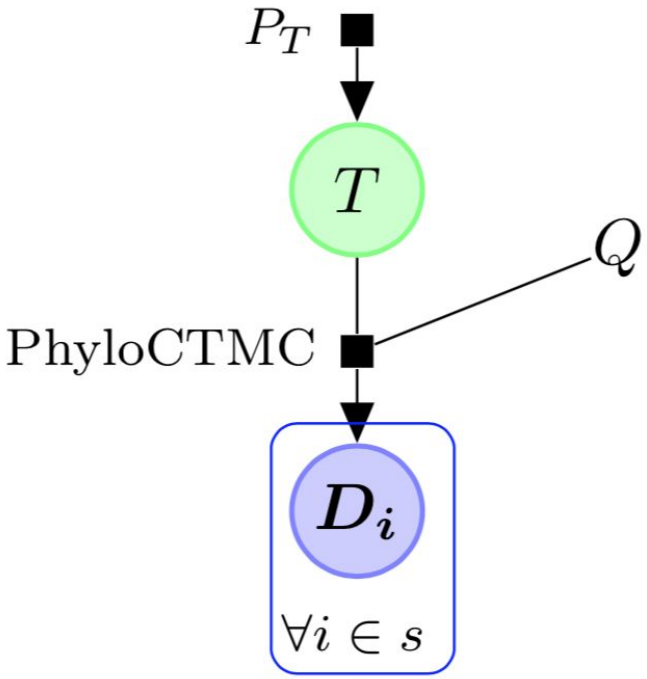
A new time tree reveals Earth history's imprint on the evolution of modern birds

Santiago Claramunt and Joel Cracraft

Published 11 December 2015, *Sci. Adv.* **1**, e1501005 (2015)
DOI: 10.1126/sciadv.1501005

We ran the main analyses using a birth-death tree prior with incomplete sampling (158) that takes into account that we sampled a small fraction of all avian species diversity; using the 'birth-death skyline contemporary' prior, part of the BEAST BDSKY 1.1.0 add-on (159), we set a uniform prior (0, infinity) for the birth/total death ratio (effective reproduction number R0), a uniform prior (0, infinity) for the "total death" rate ("become uninfectious") prior, and a prior for the sampling probability (rho) to take into account uncertainty regarding the total number of extant avian species: we set the parameters of a beta distribution so that it resulted in a bellshaped distribution with the 5% quantile at 0.023, the proportion of the currently assumed 10,000 avian species in the RAG dataset (0.0048 for the genome dataset), and the 95% quantile at 0.008 (0.0016 for the genome dataset), corresponding to assuming about 30,000 extant avian species.

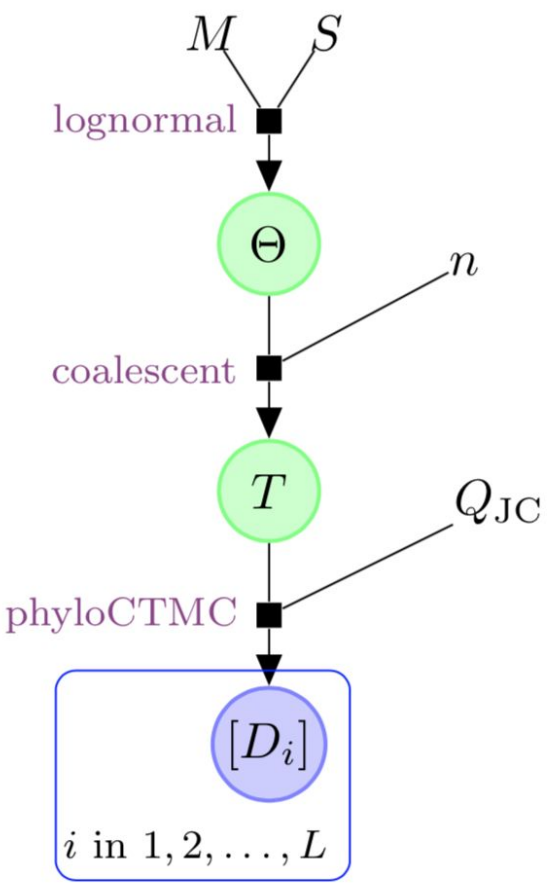
A phylogenetic model



$$P(T|\boldsymbol{D}) \propto \prod_i \Pr(\boldsymbol{D_i}|T)P(T)$$

A simple phylogenetic model

(a) Graphical model



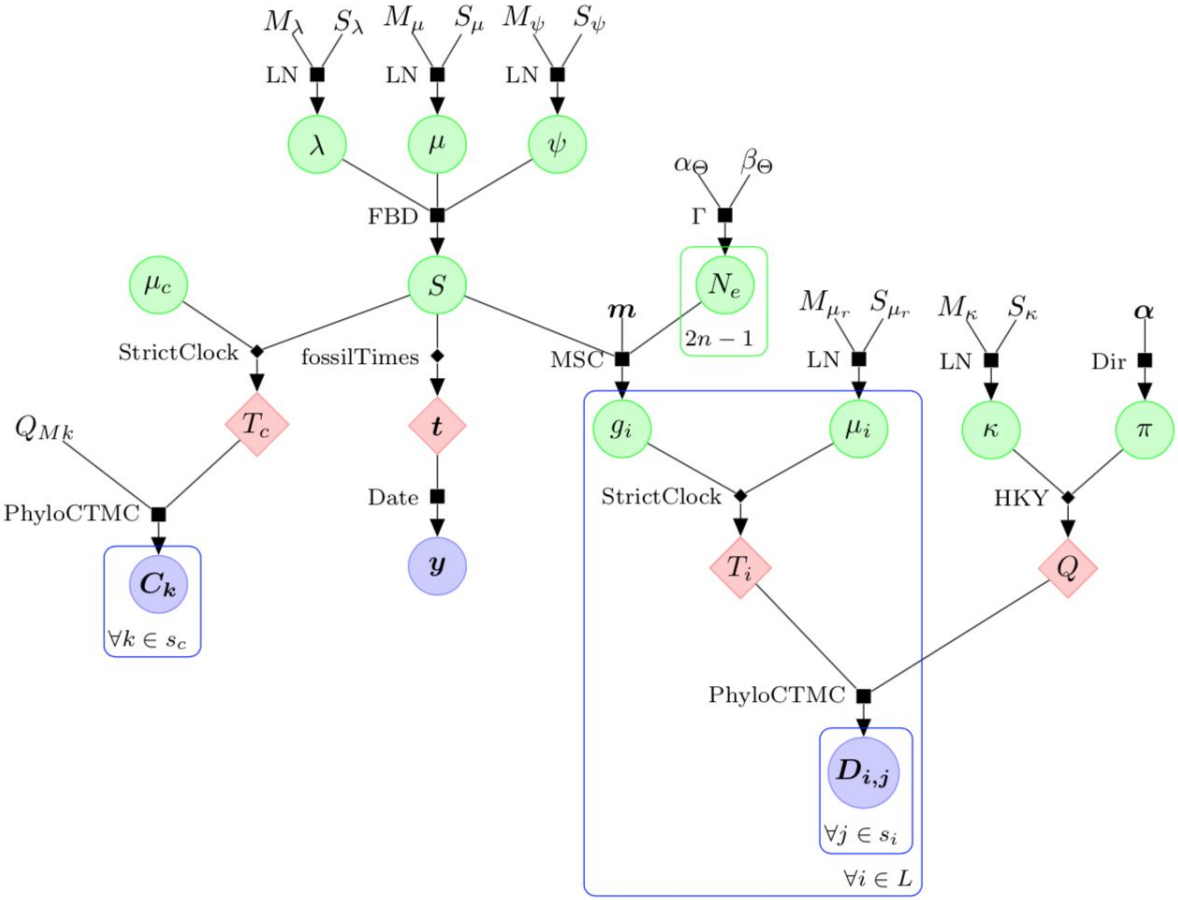
(b) Canonical phylogenetic posterior

$$P(T, \Theta|\boldsymbol{D}) = \frac{1}{P(\boldsymbol{D})} \prod_i \left[\Pr(D_i|T)\right] P(T|\Theta)P(\Theta)$$

(c) Probabilistic modelling language (EPML)

```
theta ~ lognormal(M=-5, S=1)
T ~ coalescent(theta, n=10)
D ~ phyloCTMC(T, Q=jukesCantor(), L=1000)
```

A phylogenetic model



$$P(S, \boldsymbol{N_e}, \boldsymbol{g}, \boldsymbol{\mu}, \lambda, Q | \boldsymbol{D}, \boldsymbol{C}, \boldsymbol{y}) \propto \prod_i \left[P(g_i | S, \boldsymbol{N_e}) \prod_j \Pr(\boldsymbol{D_{i,j}} | g_i \mu_i, Q) \right] \times$$
$$\prod_k \Pr(\boldsymbol{C_k} | \mu_c S, Q_{Mk}) \times$$
$$P(\boldsymbol{y}|S) P(S|\lambda) P(\lambda) P(\boldsymbol{N_e}) P(\boldsymbol{\mu}) P(\mu_c) P(Q)$$

Reproducible and reusable research

Some good/necessary trends in science:

- The data science notebook paradigm (e.g. Juypter / R notebooks)
- Everything open source, open access, open data and online
- Live science
- Not just reproducible but reusable
  - What happens if we change a key assumption in the model?
  - What happens if we add a little more data, or use different data?

The reality is that most research isn't reproducible

- Only a degree of reproducibility and reusability
  - If you have the XML file then you might be able to run it again
  - Need to find the version of the software that was used
  - No way of telling if the XML that was used to produce figures in the paper is the same as the XML that is provided in supplemental information
  - XML is very difficult for a normal person to read
    - Might be better for other formats (NEXUS?) but still rather complex and non-normalized
  - Users understanding of the model they ran is imperfect, leading to errors and omissions in model descriptions in methods section of paper
  - Can be hard to reuse XML on a new data set.

LinguaPhylo language

Language features

The LPhy language provides a simple syntax for specifying and simulating under different evolutionary models.

```
data {
  L = 200;
  taxa = taxa(names=1:10);
}
model {
  Θ ~ LogNormal(meanlog=3.0, sdlog=1.0);
  ψ ~ Coalescent(theta=Θ, taxa=taxa);
  D ~ PhyloCTMC(L=L, Q=jukesCantor(), tree=ψ);
}
```

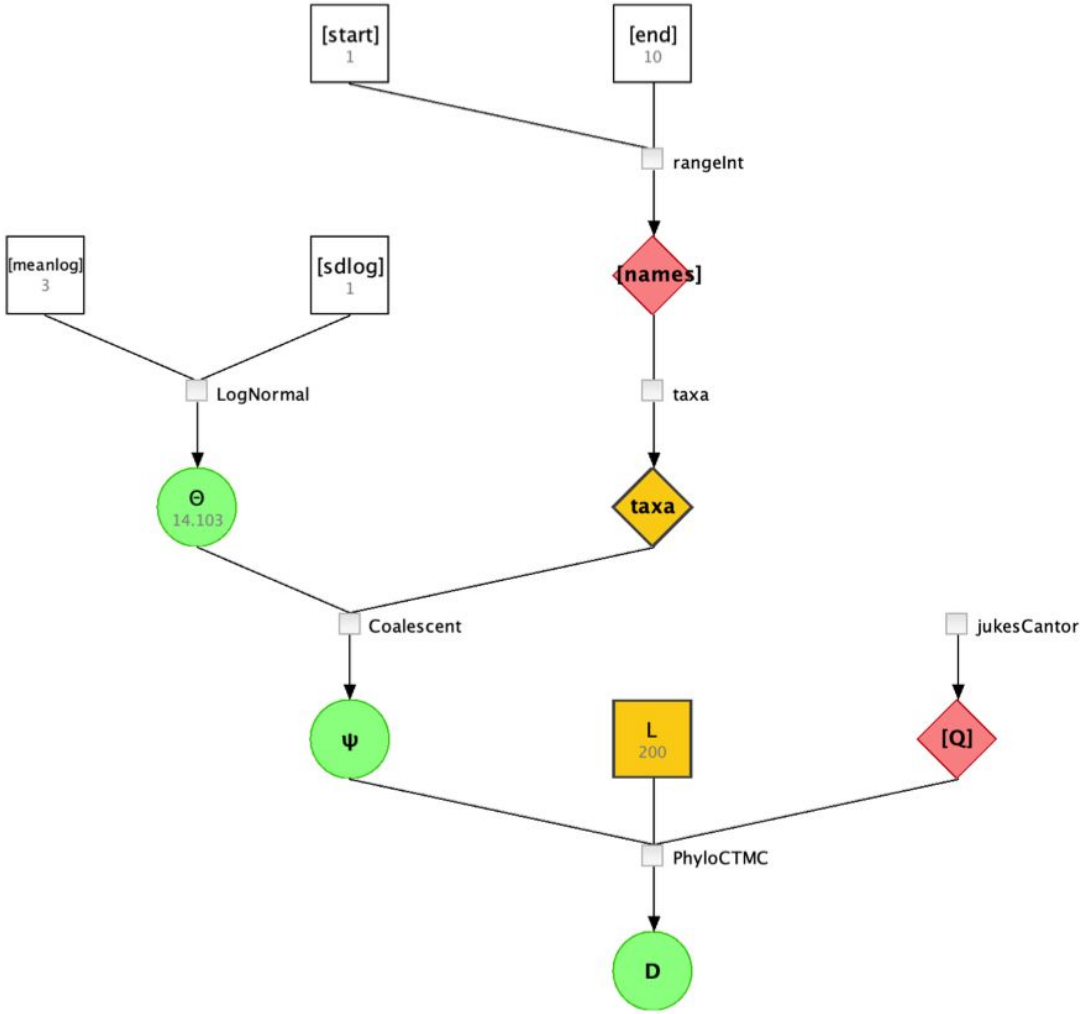
Unicode characters, such as greek letters, can be used for variable names to aid in consistency of model specification across code and mathematical descriptions.

Language features

```
data {
  L = 200;
  taxa = taxa(names=1:10);
}
model {
  Θ ~ LogNormal(meanlog=3.0, sdlog=1.0);
  ψ ~ Coalescent(theta=Θ, taxa=taxa);
  D ~ PhyloCTMC(L=L, Q=jukesCantor(), tree=ψ);
}
```

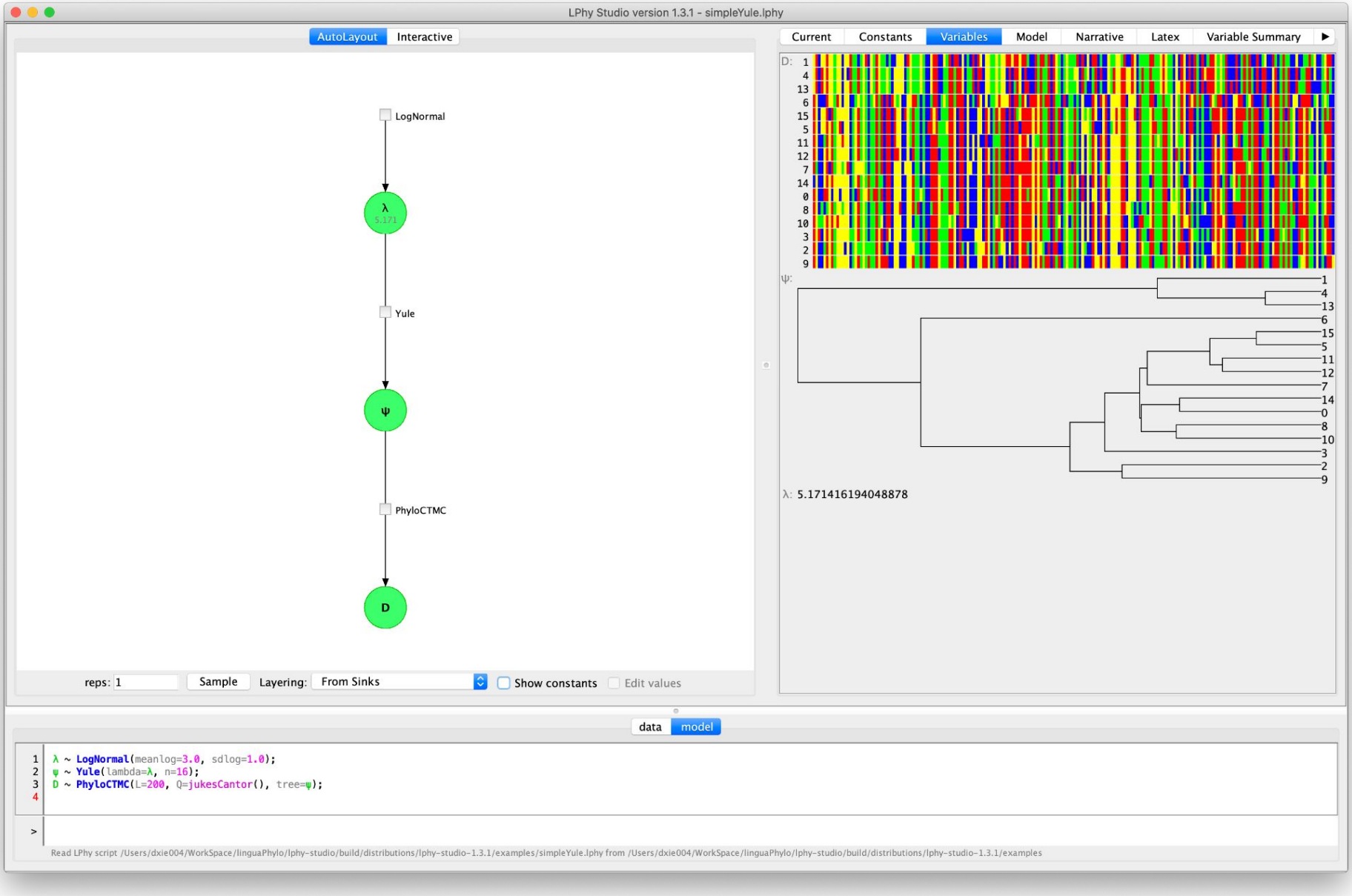
This listing above specifies a complete phylogenetic model using only five lines of code inside two blocks. Constant nodes with fixed values are shown in magenta. The first block specifies “200” nucleotide sites in “L” and ten different taxa named ‘1’ to ‘10’ in “taxa”. The second block declares the stochastic nodes or random variables in green, and their generative distributions in blue.

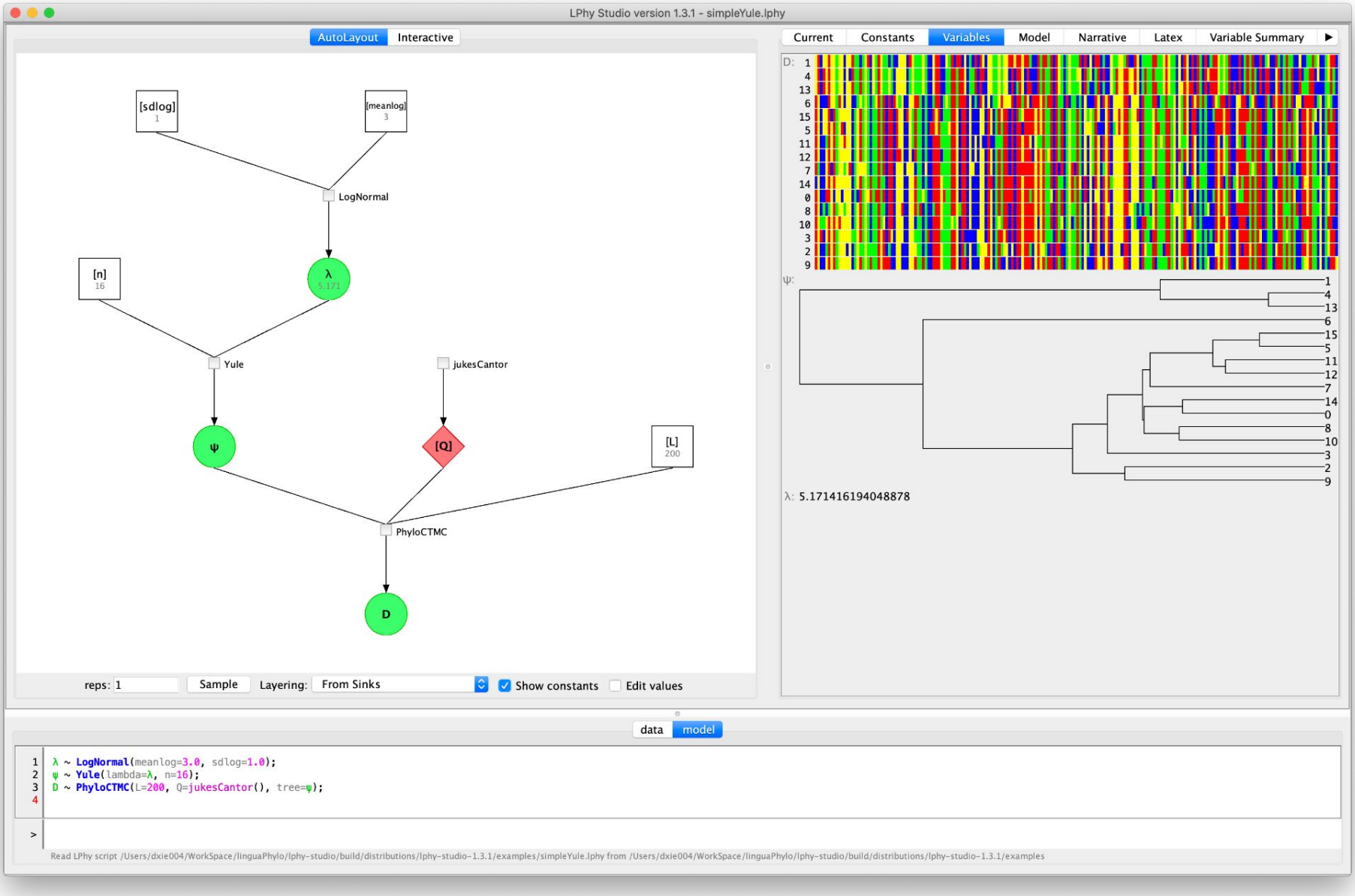
Graphical model representation



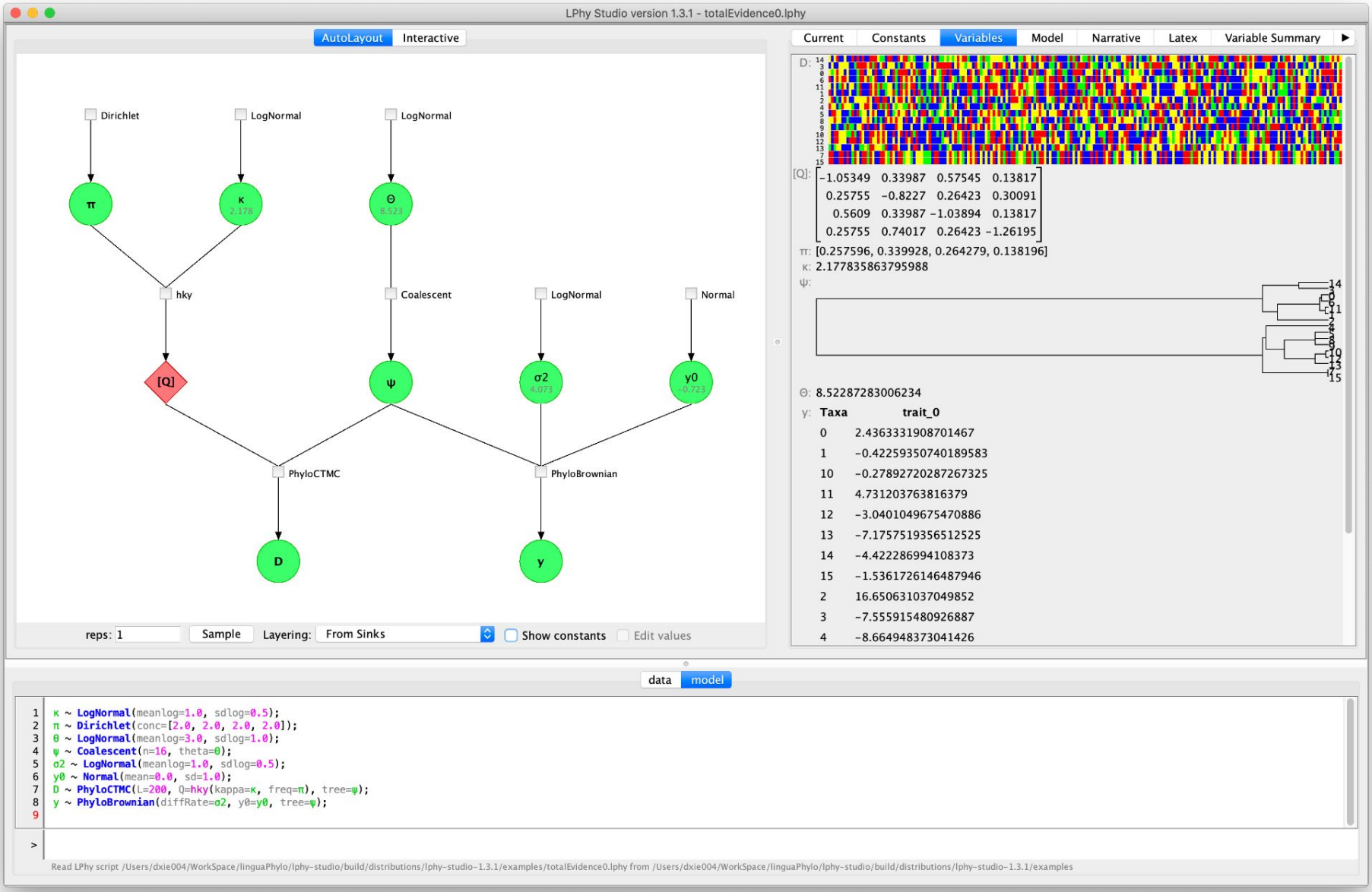
Factor nodes: nodes representing a generative distribution (e.g. Coalescent) or a deterministic function (e.g. rangeInt) with zero or more inputs and one output.

Value nodes: The result of a factor (random variables generated by a distribution, or functions of previous values), and constant nodes.

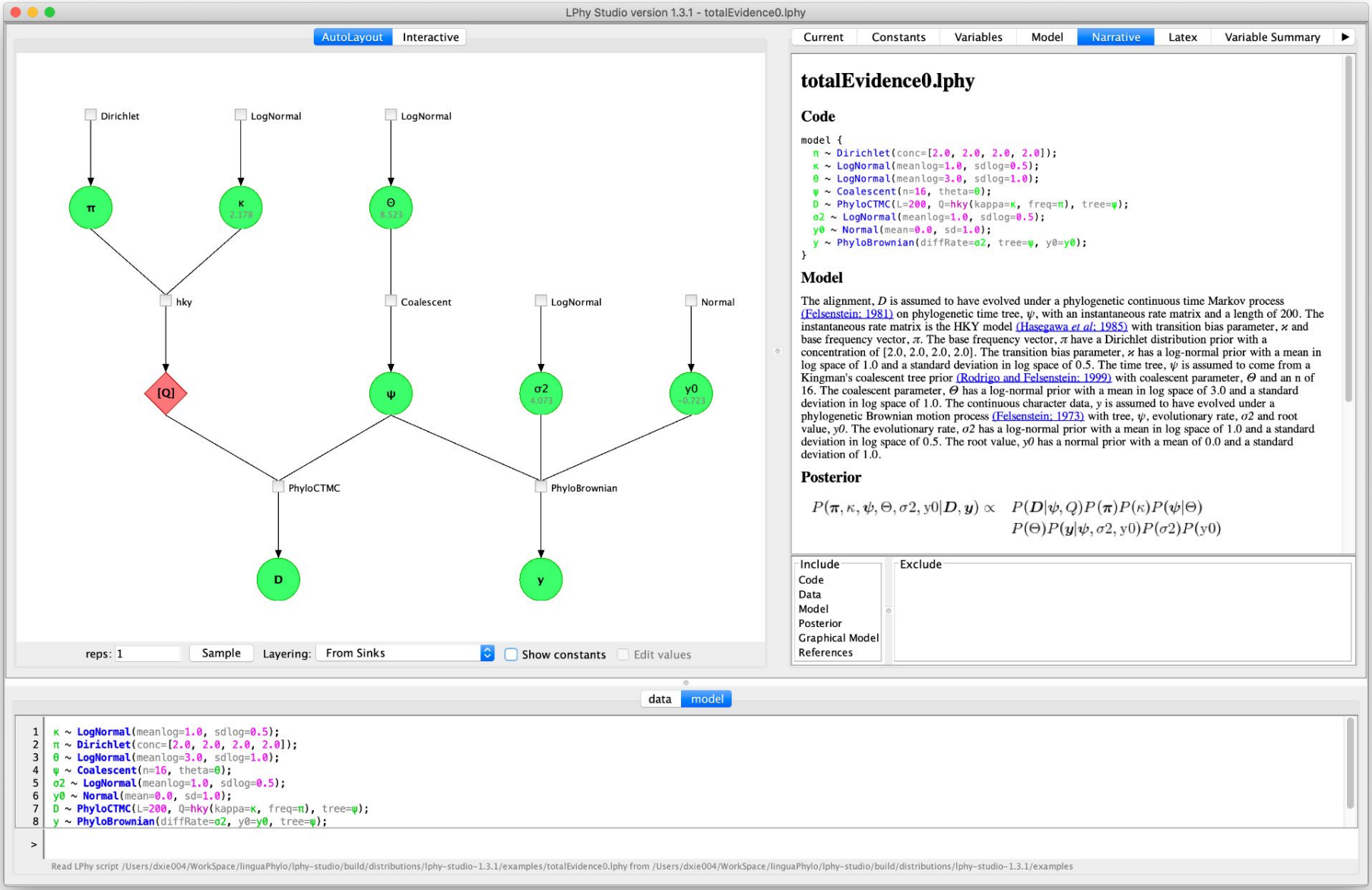

LPhy Studio version 1.3.1 - simpleYule.lphy
AutoLayout
Interactive
LogNormal
λ
5.171
Yule
ψ
PhyloCTMC
D
reps: 1
Sample
Layering: From Sinks
Show constants
Edit values
Current
Constants
Variables
Model
Narrative
Latex
Variable Summary
D: 1 4 13 6 15 5 11 12 7 14 0 8 10 3 2 9
ψ: 1 4 13 6 15 5 11 12 7 14 0 8 10 3 2 9
λ: 5.171416194048878
data
model
1 λ ~ LogNormal(meanlog=3.0, sdlog=1.0);
2 ψ ~ Yule(lambda=λ, n=16);
3 D ~ PhyloCTMC(L=200, Q=jukesCantor(), tree=ψ);
4
>
Read LPhy script /Users/dxie004/WorkSpace/linguaPhylo/lphy-studio/build/distributions/lphy-studio-1.3.1/examples/simpleYule.lphy from /Users/dxie004/WorkSpace/linguaPhylo/lphy-studio/build/distributions/lphy-studio-1.3.1/examples


LPhy Studio version 1.3.1 - simpleYule.lphy
AutoLayout
Interactive
[sdlog]
1
[meanlog]
3
LogNormal
[n]
16
λ
5.171
Yule
jukesCantor
ψ
[Q]
[L]
200
PhyloCTMC
D
reps: 1
Sample
Layering: From Sinks
Show constants
Edit values
Current
Constants
Variables
Model
Narrative
Latex
Variable Summary
D:
1
4
13
6
15
5
11
12
7
14
0
8
10
3
2
9
ψ:
λ: 5.171416194048878
data
model
1 λ ~ LogNormal(meanlog=3.0, sdlog=1.0);
2 ψ ~ Yule(lambda=λ, n=16);
3 D ~ PhyloCTMC(L=200, Q=jukesCantor(), tree=ψ);
4
>
Read LPhy script /Users/dxie004/WorkSpace/linguaPhylo/lphy-studio/build/distributions/lphy-studio-1.3.1/examples/simpleYule.lphy from /Users/dxie004/WorkSpace/linguaPhylo/lphy-studio/build/distributions/lphy-studio-1.3.1/examples

Integrative total-evidence models



Integrative total-evidence models



Vectorization / array programming

Named variables in LPhy can be scalars or vectors. Any generator can be vectorized to produce a vector of i.i.d. values by using the replicates keyword:

```
κ ~ LogNormal(meanlog=0.5, sdlog=1.0, replicates=3);
```

Vectorization can also be applied to a distribution that already produces vectors. In which case, the output will be a matrix as in the following example, where the major dimension has size 3, with each element of the π random variable being a vector of base frequencies.

```
π ~ Dirichlet(conc=[2.0, 2.0, 2.0, 2.0], replicates=3);
```

Finally, vectorization can be coerced simply by passing a vector of elements instead of a single element to one or more of the inputs of a generator:

```
Q = hky(kappa=κ, freq=π);
```

Parametric Distributions

LPhy core has the familiar parametric distributions available as generative distributions:

- Beta
- Cauchy
- Dirichlet
- Exp
- Gamma
- InverseGamma
- LogNormal
- MVN
- Normal
- Uniform
- Weibull
- Bernoulli
- Geometric
- Poisson

Version 1.4 has some core phylogenetic distributions

Tree models

- BirthDeathSerialSampling
- Coalescent
- MultispeciesCoalescent
- SkylineCoalescent
- StructuredCoalescent
- SimFossilsPoisson
- Yule
- …

Data sampling distributions

- ErrorModel
- PhyloBrownian
- PhyloCTMC
- …

Version 1.4 has some core functions

Rate matrices

- BinaryRateMatrix
- GTR
- TN93
- HKY
- F81
- K80
- JukesCantor

Phylogenetic functions

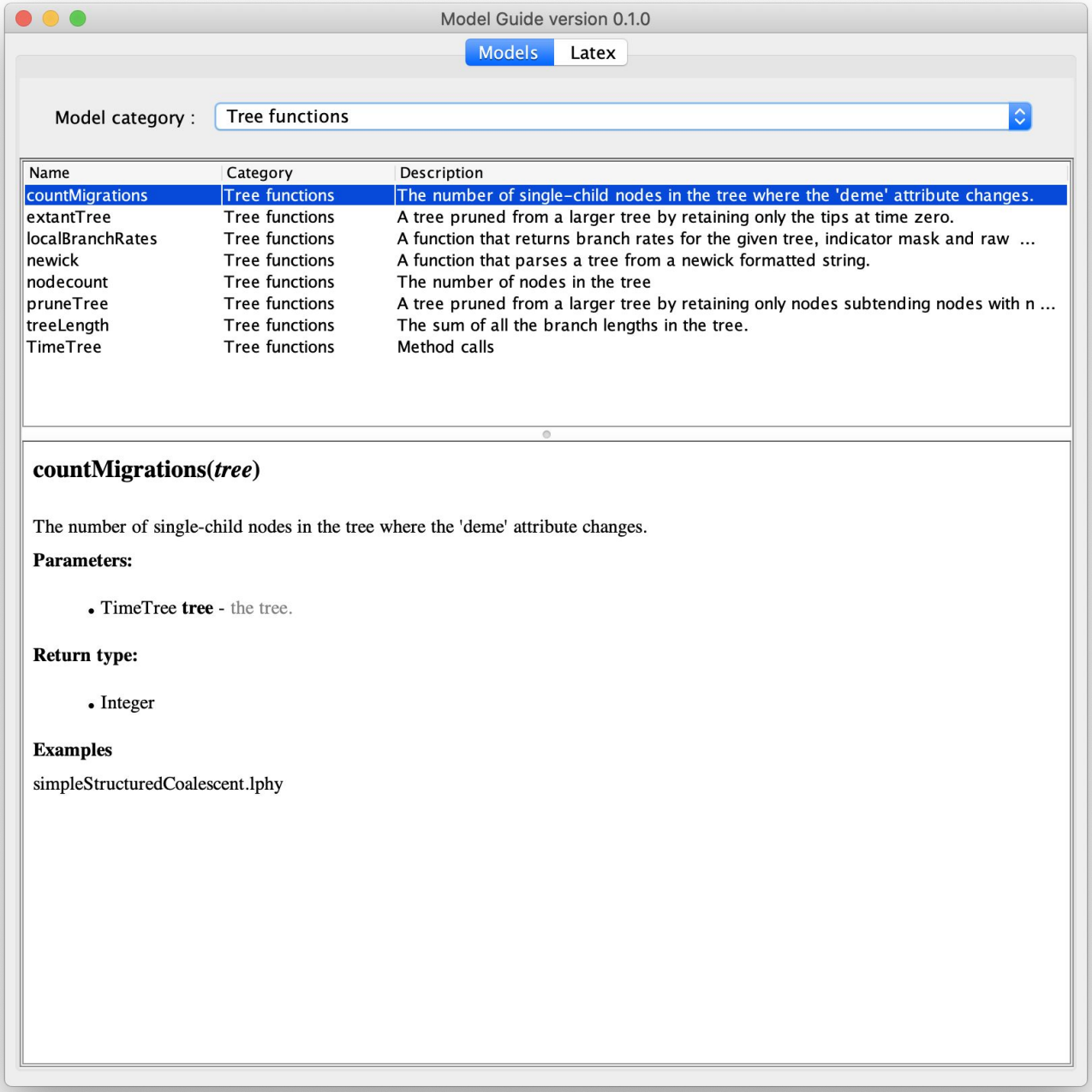
- Newick

Standard functions

- Exp
- Floor
- Log
- Rep
- …

The LPhy 1.3.1 Doc is available at

https://github.com/LinguaPhylo/linguaPhylo/blob/1.3.1/lphy/doc/index.md


Model Guide version 0.1.0
Models
Latex
Model category :
Tree functions
Name
Category
Description
countMigrations
Tree functions
The number of single-child nodes in the tree where the 'deme' attribute changes.
extantTree
Tree functions
A tree pruned from a larger tree by retaining only the tips at time zero.
localBranchRates
Tree functions
A function that returns branch rates for the given tree, indicator mask and raw ...
newick
Tree functions
A function that parses a tree from a newick formatted string.
nodecount
Tree functions
The number of nodes in the tree
pruneTree
Tree functions
A tree pruned from a larger tree by retaining only nodes subtending nodes with n ...
treeLength
Tree functions
The sum of all the branch lengths in the tree.
TimeTree
Tree functions
Method calls
countMigrations(tree)
The number of single-child nodes in the tree where the 'deme' attribute changes.
Parameters:
TimeTree tree - the tree.
Return type:
Integer
Examples
simpleStructuredCoalescent.lphy

Coalescent model

The simplest coalescent model LPhy implements is the constant-population size coalescent, which can be extended to generate serially sampled (heterochronous) data

```
ψ ~ Coalescent(theta=Θ, taxa=taxa(names=["a", "b", "c", "d"],
  ages=[0.0, 1.0, 2.0, 3.0]));
```

The taxa argument is optional, and only needed if you want to give the taxa specific names, as in the case when doing inference against specific input data.

Structured coalescent model

The structured coalescent generalizes the constant-population size coalescent by allowing multiple demes, each of which are characterized by a distinct population size (in the simplest case this population size does not change through time).

```
M = migrationMatrix(theta=[0.1, 0.1], m=[1.0, 1.0]);
g ~ StructuredCoalescent(M=M, n=[15, 15]);
```

Skyline coalescent model

The following script specifies a Skyline coalescent model with 10 coalescent intervals (hence 11 taxa), governed by four distinct population sizes.

```
g ~ SkylineCoalescent(theta=[0.1, 0.2, 0.3, 0.4], groupSizes=[4,3,2,1]);
```

Here, “g” is a stochastic node in the PGM, with its value sampled from the 160 SkylineCoalescent generative distribution. Ten coalescent intervals are defined 161 through the “groupSizes” argument: the first four coalescent intervals will be drawn 162 assuming a “theta” of 0.1, the next three intervals with “theta” equal to 0.2, and so on.

Birth-death-serial-sampling model

Birth-death models are commonly used in macroevolution as sampling distributions for species trees.

```
ages = [0.0, 1.0, 2.0, 3.0, 4.0];
tree ~ BirthDeathSerialSampling(lambda=1, mu=0.5, rho=0.1,
psi=1, rootAge=5, ages=ages);
```

Other tree models including different conditions on birth-death models, the fossilized birth-death model and the Yule process are also implemented.

Substitution models

Substitution models consist of continuous-time Markov chains (CTMC) used to model the evolution of discrete characters, such as nucleotides and amino acid residues.

LPhy implements a general formulation of a phylogenetic CTMC, known as the GTR model, under which several nested models can be specified. The first line below constructs the instantaneous rate matrix (Q) for an HKY model, which is then used to in PhyloCTMC, the generative distribution over sequence alignment data (D):

```
Q = hky(kappa=2.0, freq=[0.2, 0.25, 0.3, 0.25]);
Θ ~ LogNormal(meanlog=3.0, sdlog=1.0);
ψ ~ Coalescent(theta=Θ, taxa=taxa);
D ~ PhyloCTMC(L=200, Q=Q, tree=ψ);
```

Evolutionary clock models

The simplest clock model is the strict clock, under which the evolutionary rate remains constant over the entire tree. Specifying a strict molecular clock can be done by specifying the “mu” parameter in the PhyloCTMC distribution (default is 1.0)

```
λ ~ LogNormal(meanlog=3.0, sdlog=1.0);
ψ ~ Yule(lambda=λ, n=16);
D ~ PhyloCTMC(L=200, Q=jukesCantor(), tree=ψ, mu=0.5);
```

Relaxed phylogenetic clock models

More realistic clock models like the uncorrelated relaxed clock model assume the rate for each branch is drawn according to a parametric distribution. For example, a relaxed clock with rates drawn from a log-normal distribution can be constructed as follows:

```
λ ~ LogNormal(meanlog=3.0, sdlog=1.0);
ψ ~ Yule(lambda=λ, n=16);
branchRates ~ LogNormal(sdlog=0.5, meanlog=-0.25, replicates=ψ.branchCount());
D ~ PhyloCTMC(L=200, Q=jukesCantor(), branchRates=branchRates, tree=ψ);
```

Documentation

The art of constructing generative distribution specifications is to do so in the most canonical and intuitive way possible. However there may still be ambiguity or uncertainty, so good documentation is essential (https://linguaphylo.github.io).

SkylineCoalescent distribution

SkylineCoalescent(Double[] theta, Integer[] groupSizes, Integer n, Taxa taxa, Double[] ages)

The skyline coalescent distribution over tip-labelled time trees. If no group sizes are specified, then there is one population parameter per coalescent event (as per classic skyline coalescent of Pybus, Rambaut and Harvey 2000)

Parameters

- Double[] **theta** - effective population size, one value for each group of coalescent intervals, ordered from present to past. Possibly scaled to mutations or calendar units. If no groupSizes are specified, then the number of coalescent intervals will be equal to the number of population size parameters.
- Integer[] **groupSizes** - A tuple of group sizes. The sum of this tuple determines the number of coalescent events in the tree and thus the number of taxa. By default all group sizes are 1 which is equivalent to the classic skyline coalescent.
- Integer **n** - number of taxa.
- Taxa **taxa** - Taxa object, (e.g. Taxa or Object[])
- Double[] **ages** - an array of leaf node ages.

Return type

- TimeTree

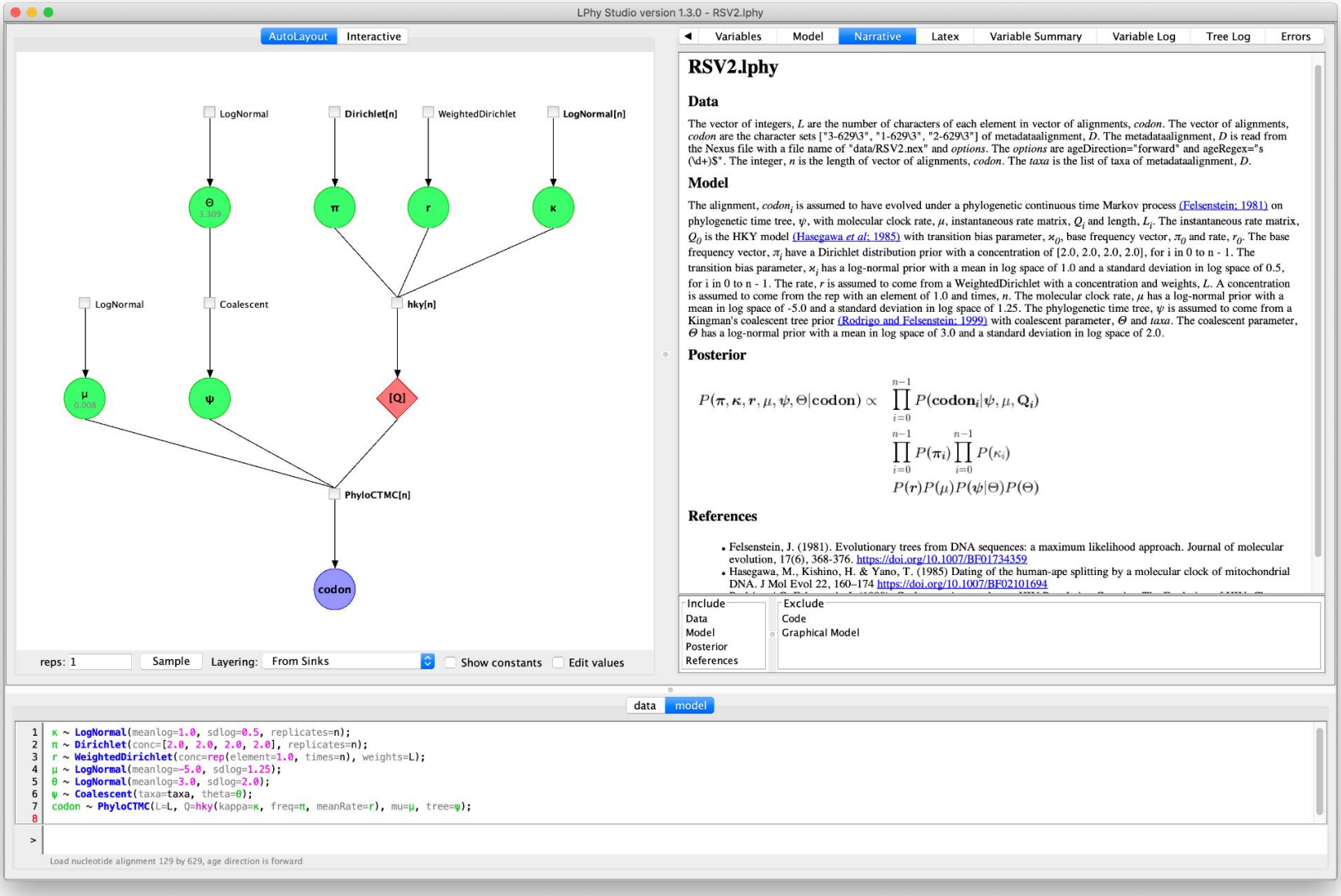
Reference

Drummond, A. J., Rambaut, A., Shapiro, B, & Pybus, O. G. (2005). Bayesian coalescent inference of past population dynamics from molecular sequences. Molecular biology and evolution, 22(5), 1185-1192.http://doi.org/10.1093/molbev/msi103

Inference and Data clamping

```
data {
    options = {ageDirection="forward", ageRegex="s(\d+)"};
    nexusFilePath = "tutorials/data/RSV2.nex";
    D = readNexus(file=nexusFilePath, options=options);
    codon = D.charset(["3-629\3", "1-629\3", "2-629\3"]);
    n = 3;
    L = [209, 210, 210];
    taxa = D.taxa();
}
model {
    π ~ Dirichlet(replicates=n, conc=[2.0, 2.0, 2.0, 2.0]);
    κ ~ LogNormal(sdlog=0.5, meanlog=1.0, replicates=n);
    r ~ WeightedDirichlet(conc=rep(element=1.0, times=n), weights=L);
    μ ~ LogNormal(meanlog=-5.0, sdlog=1.25);
    Θ ~ LogNormal(meanlog=3.0, sdlog=2.0);
    ψ ~ Coalescent(taxa=taxa, theta=Θ);
    Q = hky(kappa=κ, freq=π, meanRate=r);
    codon ~ PhyloCTMC(L=L, Q=Q, mu=μ, tree=ψ);
}
```

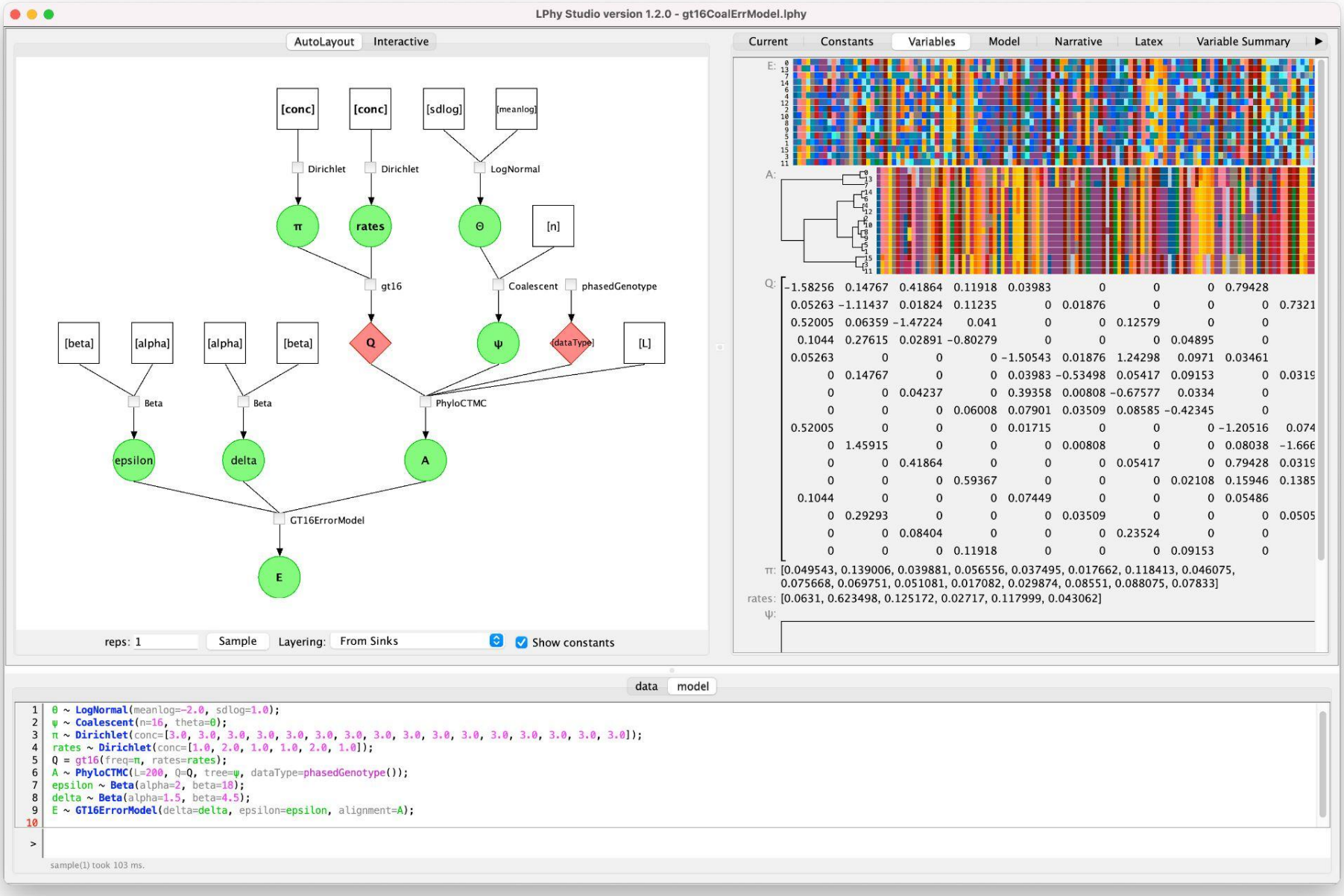
LPhyStudio


LPhy Studio version 1.3.0 - RSV2.lphy
AutoLayout
Interactive
LogNormal
Dirichlet[n]
WeightedDirichlet
LogNormal[n]
Θ
3.309
π
r
κ
LogNormal
Coalescent
hky[n]
μ
0.008
ψ
[Q]
PhyloCTMC[n]
codon
reps: 1
Sample
Layering:
From Sinks
Show constants
Edit values
Variables
Model
Narrative
Latex
Variable Summary
Variable Log
Tree Log
Errors
RSV2.lphy
Data
The vector of integers, L are the number of characters of each element in vector of alignments, codon. The vector of alignments, codon are the character sets ["3-629\3", "1-629\3", "2-629\3"] of metadataalignment, D. The metadataalignment, D is read from the Nexus file with a file name of "data/RSV2.nex" and options. The options are ageDirection="forward" and ageRegex="s(\d+)$". The integer, n is the length of vector of alignments, codon. The taxa is the list of taxa of metadataalignment, D.
Model
The alignment, codon_i is assumed to have evolved under a phylogenetic continuous time Markov process (Felsenstein; 1981) on phylogenetic time tree, ψ, with molecular clock rate, μ, instantaneous rate matrix, Q_i and length, L_i. The instantaneous rate matrix, Q_0 is the HKY model (Hasegawa et al; 1985) with transition bias parameter, κ_0, base frequency vector, π_0 and rate, r_0. The base frequency vector, π_i have a Dirichlet distribution prior with a concentration of [2.0, 2.0, 2.0, 2.0], for i in 0 to n - 1. The transition bias parameter, κ_i has a log-normal prior with a mean in log space of 1.0 and a standard deviation in log space of 0.5, for i in 0 to n - 1. The rate, r is assumed to come from a WeightedDirichlet with a concentration and weights, L. A concentration is assumed to come from the rep with an element of 1.0 and times, n. The molecular clock rate, μ has a log-normal prior with a mean in log space of -5.0 and a standard deviation in log space of 1.25. The phylogenetic time tree, ψ is assumed to come from a Kingman's coalescent tree prior (Rodrigo and Felsenstein; 1999) with coalescent parameter, Θ and taxa. The coalescent parameter, Θ has a log-normal prior with a mean in log space of 3.0 and a standard deviation in log space of 2.0.
Posterior
$P(\boldsymbol{\pi}, \boldsymbol{\kappa}, \boldsymbol{r}, \mu, \boldsymbol{\psi}, \Theta | \mathbf{codon}) \propto \prod_{i=0}^{n-1} P(\mathbf{codon_i} | \boldsymbol{\psi}, \mu, \mathbf{Q_i}) \prod_{i=0}^{n-1} P(\boldsymbol{\pi_i}) \prod_{i=0}^{n-1} P(\kappa_i) P(\boldsymbol{r}) P(\mu) P(\boldsymbol{\psi} | \Theta) P(\Theta)$
References
Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of molecular evolution, 17(6), 368-376. https://doi.org/10.1007/BF01734359
Hasegawa, M., Kishino, H. & Yano, T. (1985) Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. J Mol Evol 22, 160–174 https://doi.org/10.1007/BF02101694
Include
Data
Model
Posterior
References
Exclude
Code
Graphical Model
data
model
1 κ ~ LogNormal(meanlog=1.0, sdlog=0.5, replicates=n);
2 π ~ Dirichlet(conc=[2.0, 2.0, 2.0, 2.0], replicates=n);
3 r ~ WeightedDirichlet(conc=rep(element=1.0, times=n), weights=L);
4 μ ~ LogNormal(meanlog=-5.0, sdlog=1.25);
5 θ ~ LogNormal(meanlog=3.0, sdlog=2.0);
6 ψ ~ Coalescent(taxa=taxa, theta=θ);
7 codon ~ PhyloCTMC(L=L, Q=hky(kappa=κ, freq=π, meanRate=r), mu=μ, tree=ψ);
8
>
Load nucleotide alignment 129 by 629, age direction is forward

Simulation example

```
model {
  π  ~ Dirichlet(conc=[3.0, 3.0, 3.0, 3.0,
                       3.0, 3.0, 3.0, 3.0,
                       3.0, 3.0, 3.0, 3.0,
                       3.0, 3.0, 3.0, 3.0]);
  rates ~ Dirichlet(conc=[1.0, 2.0, 1.0, 1.0, 2.0, 1.0]);
  Q = gt16(rates=rates, freq=π);
  Θ ~ LogNormal(meanlog=-2.0, sdlog=1.0);
  ψ ~ Coalescent(n=16, theta=Θ);
  A ~ PhyloCTMC(L=200, Q=Q, dataType=phasedGenotype(), tree=ψ);
  delta ~ Beta(alpha=1.5, beta=4.5);
  epsilon ~ Beta(alpha=2, beta=18);
  E ~ GT16ErrorModel(alignment=A, delta=delta, epsilon=epsilon);
}
```

Model in LPhyStudio



Method section generation

The alignment, $\boldsymbol{E}$ is assumed to come from a GT16ErrorModel. The alignment, $\boldsymbol{A}$ is assumed to have evolved under a phylogenetic continuous time Markov process [37] on phylogenetic time tree, $\boldsymbol{\psi}$, with instantaneous rate matrix, $\boldsymbol{Q}$, a length of 200 and a dataType. The instantaneous rate matrix, $\boldsymbol{Q}$ is the general time-reversible rate matrix on phased genotypes [35] with relative rates, **rates** and base frequencies, $\boldsymbol{\pi}$. The base frequencies, $\boldsymbol{\pi}$ have a Dirichlet distribution prior with a concentration of [3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0, 3.0]. The relative rates, **rates** have a Dirichlet distribution prior with a concentration of [1.0, 2.0, 1.0, 1.0, 2.0, 1.0]. The dataType is the phased genotype data type. The phylogenetic time tree, $\boldsymbol{\psi}$ is assumed to come from a Kingman's coalescent tree prior [21] with coalescent parameter, Θ and an n of 16. The coalescent parameter, Θ has a log-normal prior with a mean in log space of -2.0 and a standard deviation in log space of 1.0. The delta has a Beta distribution prior with an alpha of 1.5 and a beta of 4.5. The epsilon has a Beta distribution prior with an alpha of 2 and a beta of 18.

LinguaPhylo is fully extensible by a Java API

```
public interface LPhyExtension {

    /**
     * @return the list of new {@link GenerativeDistribution} implemented
     *         in the LPhy extension.
     */
    List<Class<? extends GenerativeDistribution>> getDistributions();

    /**
     * @return the list of new {@link Func} implemented in the LPhy extension.
     */
    List<Class<? extends Func>> getFunctions();

    /**
     * @return the map of new {@link SequenceType} implemented in the LPhy exten
     *         The string key is a keyword to represent this SequenceType.
     *         The keyword can be used to identify and initialise the
     *         corresponding sequence type.
     */
    Map<String, ? extends SequenceType> getSequenceTypes();

}
```

GenerativeDistribution interface

```
12  public interface GenerativeDistribution<T> extends Generator<T> {
13
14      /**
15       * The {@link RandomVariable} must be re-wrapped
16       * to ensure correct behaviour downstream.
17       * @return {@link RandomVariable} to connect to this {@link GenerativeDistribution}
18       */
19      RandomVariable<T> sample();
```

Func abstract class

```
/**
 * @return a value generated by this generator.
 */
Value<T> generate();
```

LPhyBEAST

- Takes an LPhy script as input
- Generates BEAST 2.6 XML as output
- Can produce many model combinations that BEAUti can't
- Not all valid LPhy scripts are supported by BEAST 2.6

Future directions

- Supporting BEAST module developers to provide LPhy/LPhyBEAST support
- LPhyRevBayes
- LPhyMrBayes
- Integration of LPhy into Data Science notebook frameworks like Jupyter Notebooks and/or Stencila
- A formal language for defining the analytical outputs and figures of a particular analysis using a Bayesian model?