

Phylogenetics

Introduction to Bayesian
inference

RL-V3 MPP

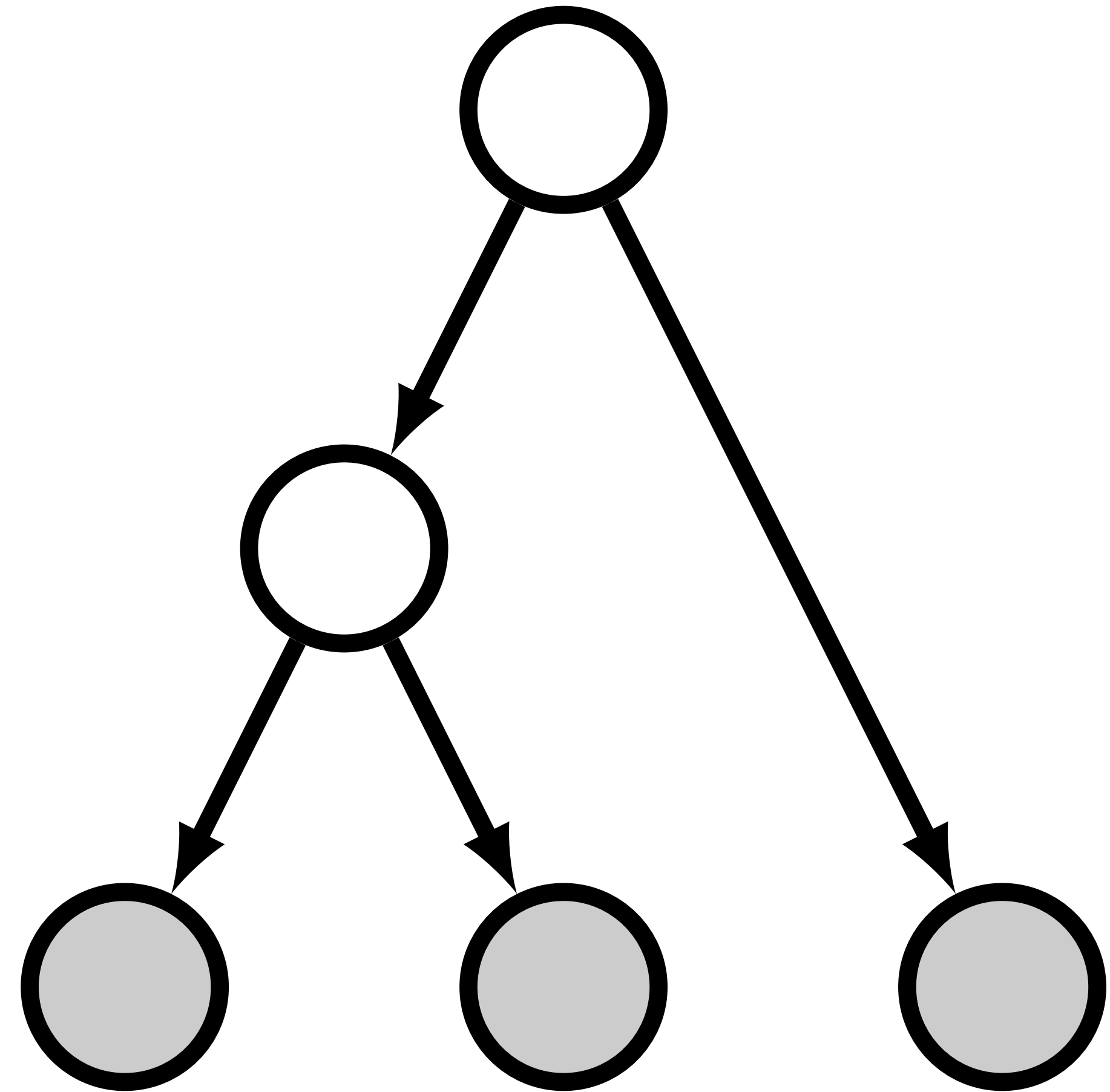
Rachel Warnock

02.06.26



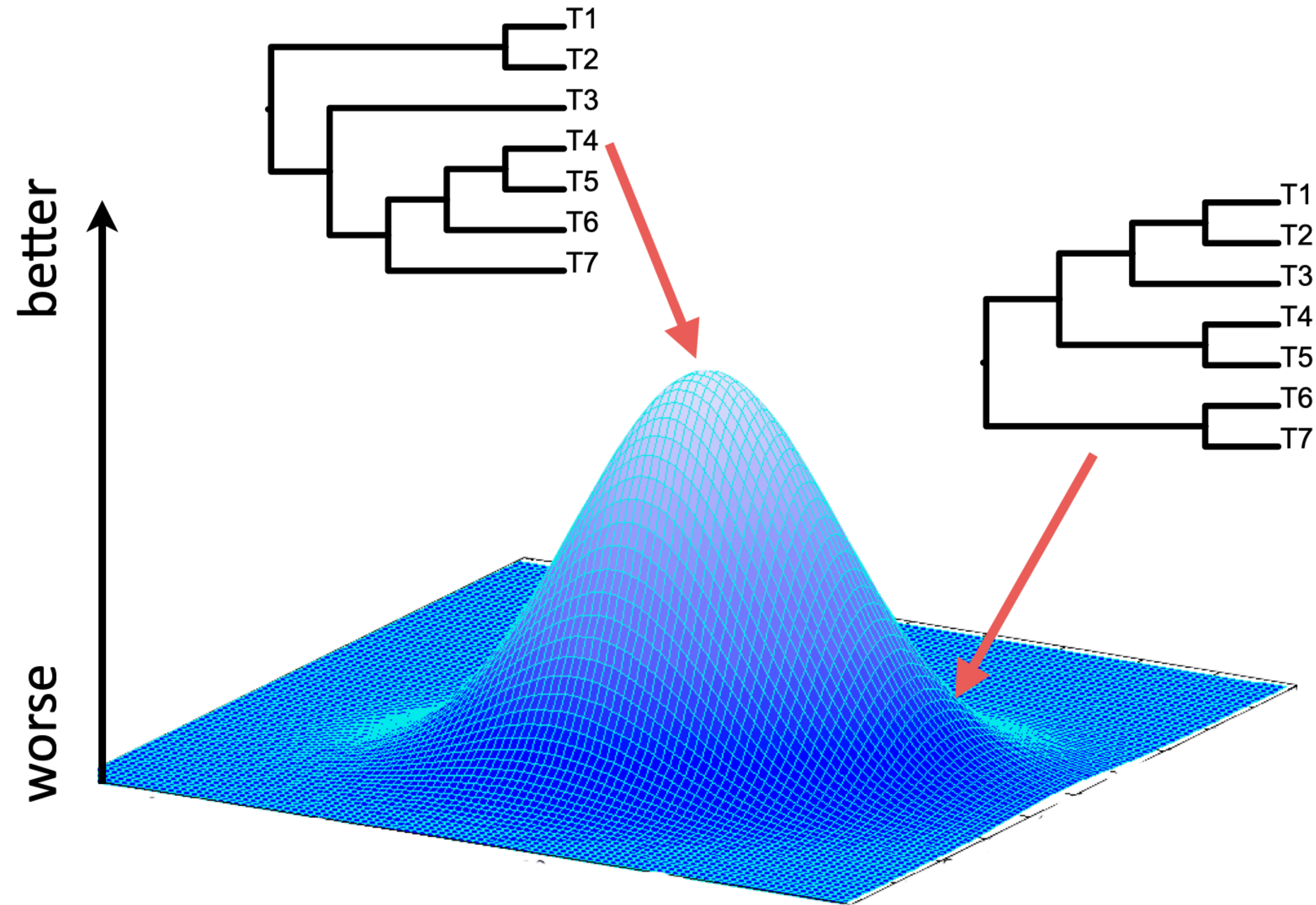
Objectives

- Bayesian inference
- MCMC
- MrBayes



Recap

How do we find the 'best' tree?



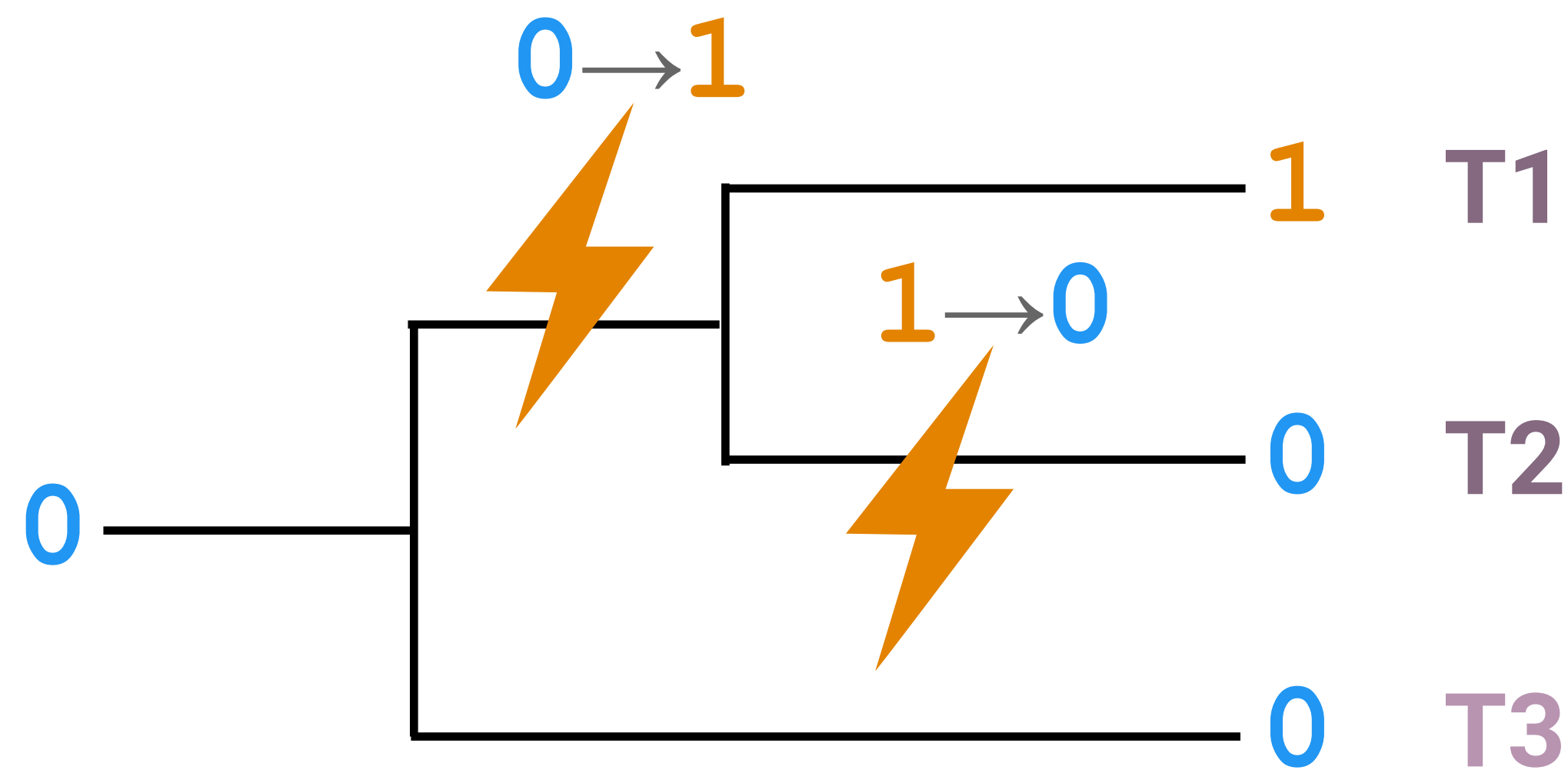
It depends how you measure 'best'

Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

Both maximum likelihood and Bayesian inference are model-based approaches

Note these are not the only approaches to tree-building but they are the most widely used

Convergence and parsimony



Hypothetical tree showing multiple transitions at the same character

Parsimony will always favour the tree with the smallest number of changes

The method does not account for multiple transitions (or “hits”), e.g.,
 $0 \rightarrow 1 \rightarrow 0$

We can only invoke convergent evolution *ad hoc* after inference

Recap: parsimony

Parsimony does not make **explicit** assumptions about the evolutionary process (although it makes **implicit** assumptions)

It has been demonstrated that in some scenarios parsimony is **statistically inconsistent**. The issue is known as **long branch attraction**

Model-based approaches on the other hand make **explicit** assumptions about evolutionary processes (+ have a wider range of applications in paleobio)

It depends how you measure 'best'

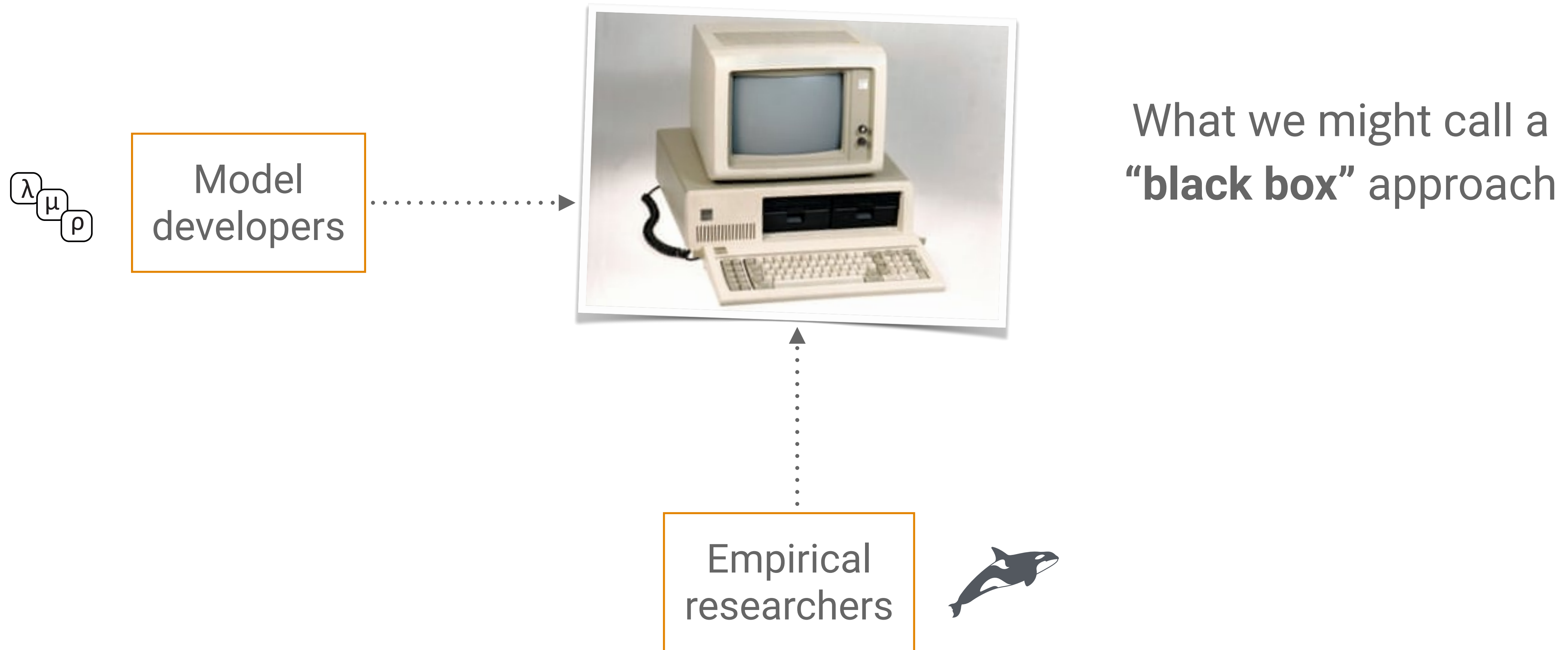
Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

Both maximum likelihood and Bayesian inference are model-based approaches

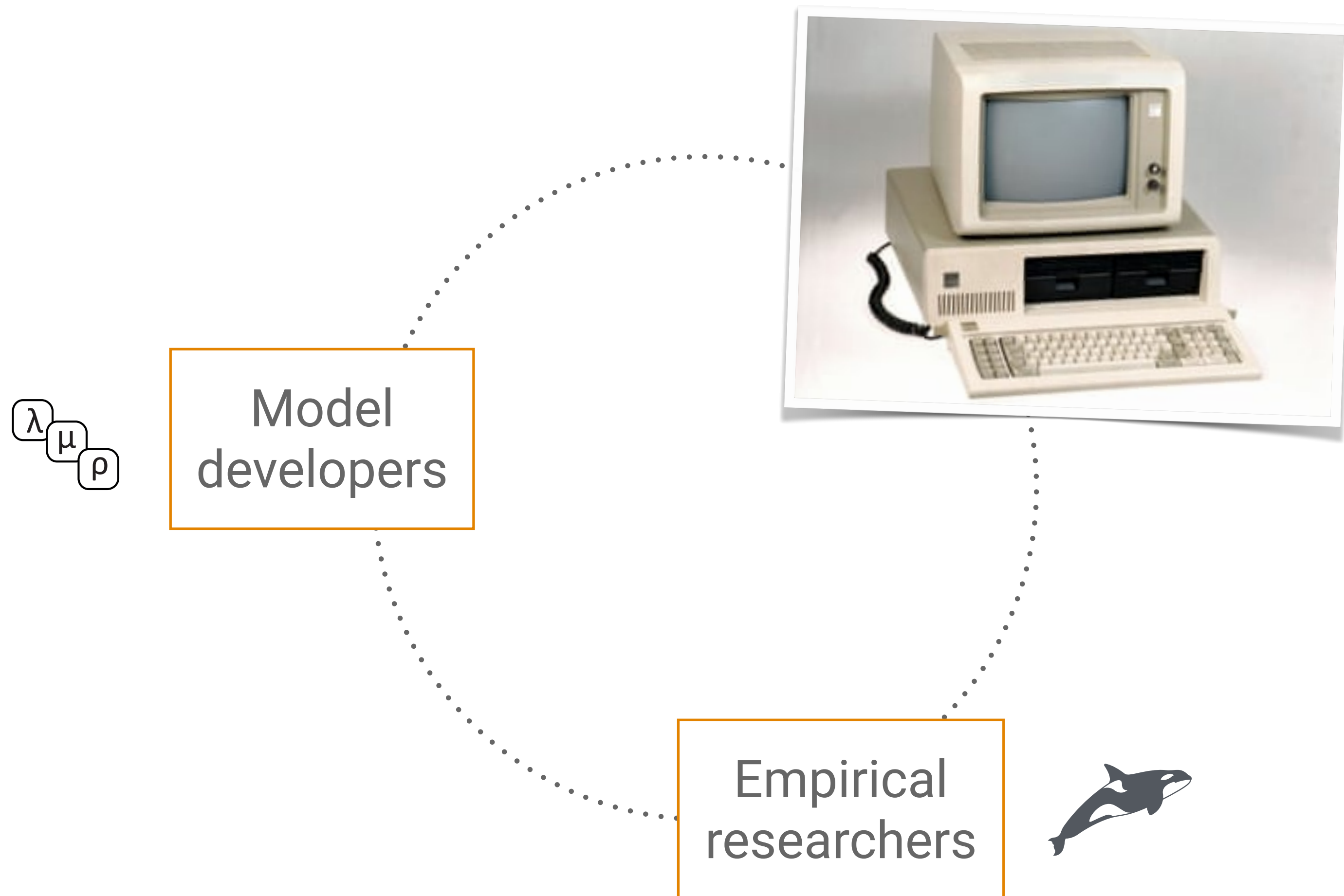
Note these are not the only approaches to tree-building but they are the most widely used

RevBayes

Phylogenetic inference – the old way



Phylogenetic inference – a better way?



The goal is to bring researchers with different expertise together, increase transparency, and do better research

RevBayes



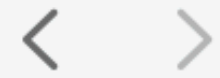
Named after Reverend Bayes, “descended” from the software MrBayes

Designed with extendability and flexibility in mind

Rev language, similar to R, and uses a graphical modelling framework

Developed and supported by a large international team of developers

revbayes.github.io



revbayes.github.io



Download

Tutorials

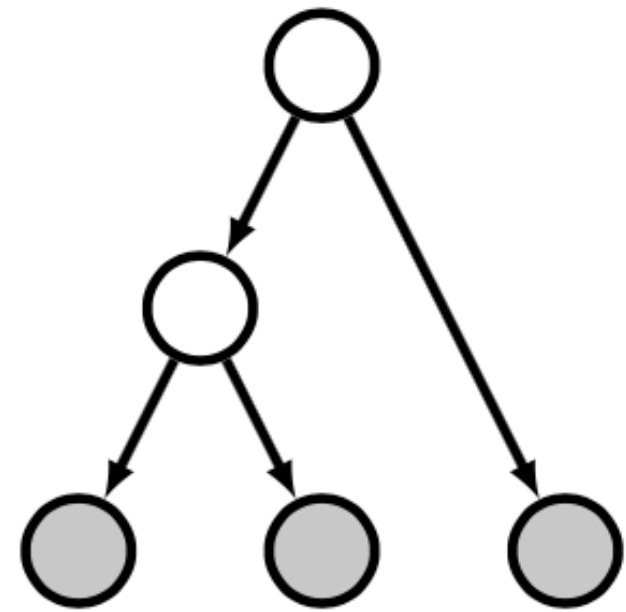
Documentation

Interfaces

Workshops

Jobs

Developer



RevBayes

Bayesian phylogenetic inference using probabilistic graphical models and an interpreted language

About

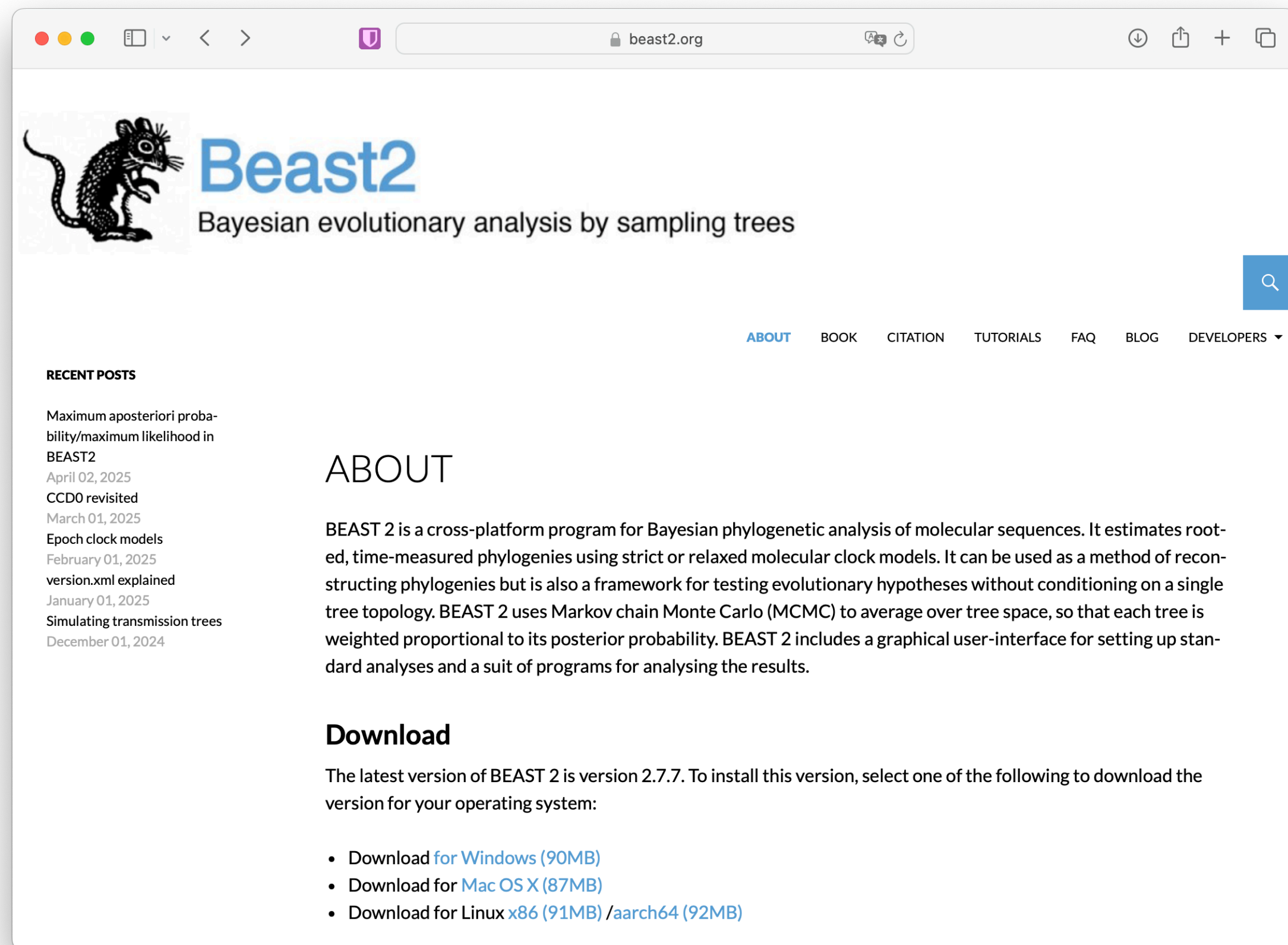
RevBayes provides an interactive environment for statistical computation in phylogenetics. It is primarily intended for modeling, simulation, and Bayesian inference in evolutionary biology, particularly phylogenetics. However, the environment is quite general and can be useful for many complex modeling tasks.

RevBayes uses its own language, Rev, which is a probabilistic programming language like [JAGS](#), [STAN](#), [Edward](#), [PyMC3](#), and related software. However, phylogenetic models require inference machinery and distributions that are unavailable in these other tools.

The Rev language is similar to the language used in R. Like the R language, Rev is designed to support interactive analysis. It supports both functional and procedural programming models, and makes a clear distinction between the two. Rev is also more strongly typed than R.

RevBayes is a collaboratively [developed](#) software project.

[GitHub](#) | [License](#) | [Citation](#) | [Users Forum](#)



Maximum a posteriori probability/maximum likelihood in BEAST2
April 02, 2025

CCDO revisited
March 01, 2025

Epoch clock models
February 01, 2025

version.xml explained
January 01, 2025

Simulating transmission trees
December 01, 2024

ABOUT

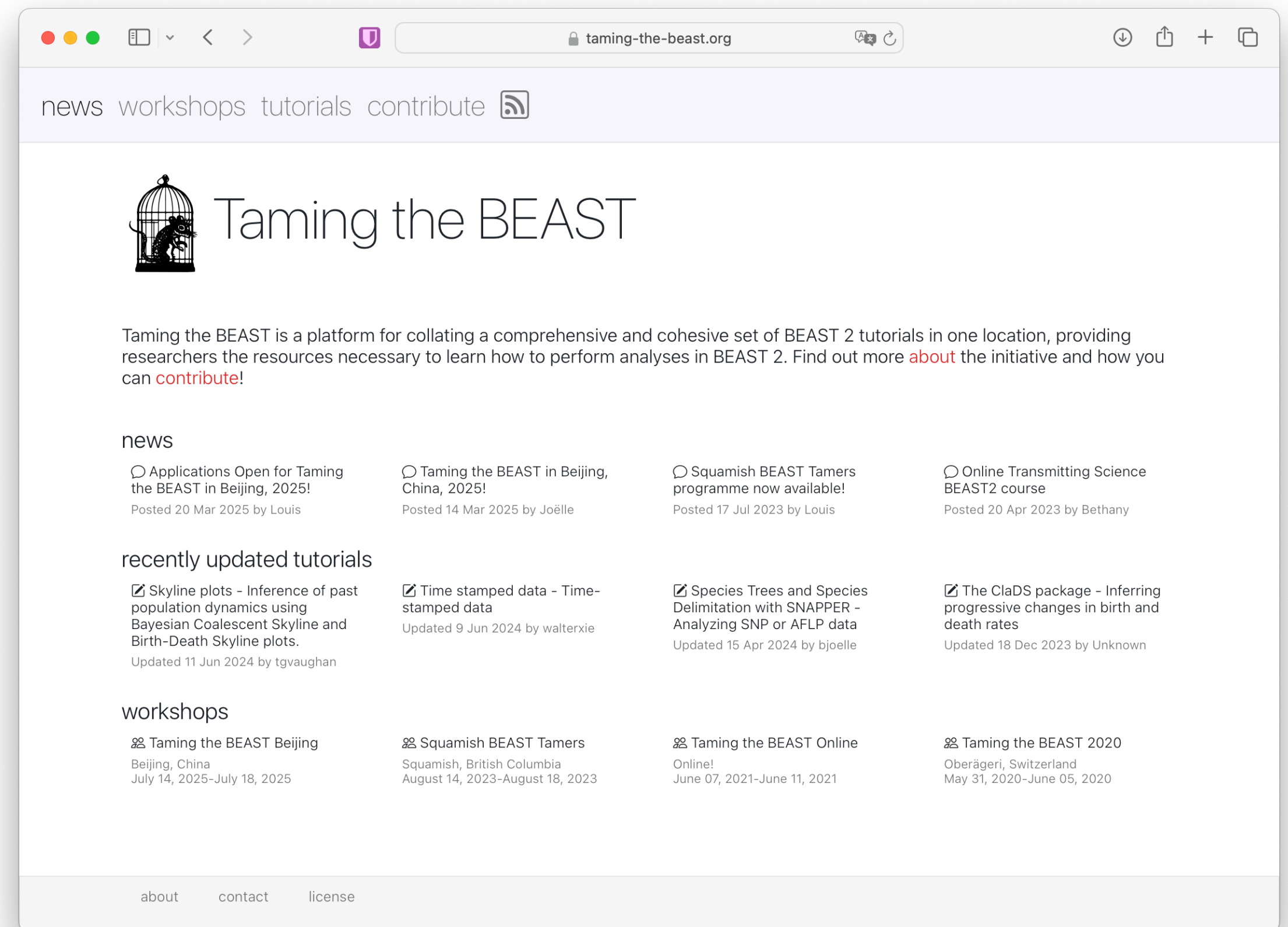
BEAST 2 is a cross-platform program for Bayesian phylogenetic analysis of molecular sequences. It estimates rooted, time-measured phylogenies using strict or relaxed molecular clock models. It can be used as a method of reconstructing phylogenies but is also a framework for testing evolutionary hypotheses without conditioning on a single tree topology. BEAST 2 uses Markov chain Monte Carlo (MCMC) to average over tree space, so that each tree is weighted proportional to its posterior probability. BEAST 2 includes a graphical user-interface for setting up standard analyses and a suit of programs for analysing the results.

Download

The latest version of BEAST 2 is version 2.7.7. To install this version, select one of the following to download the version for your operating system:

- Download for Windows (90MB)
- Download for Mac OS X (87MB)
- Download for Linux x86 (91MB) / aarch64 (92MB)

www.beast2.org



Taming the BEAST is a platform for collating a comprehensive and cohesive set of BEAST 2 tutorials in one location, providing researchers the resources necessary to learn how to perform analyses in BEAST 2. Find out more [about](#) the initiative and how you can [contribute](#)!

news

- Applications Open for Taming the BEAST in Beijing, 2025!
Posted 20 Mar 2025 by Louis
- Taming the BEAST in Beijing, China, 2025!
Posted 14 Mar 2025 by Joëlle
- Squamish BEAST Tamers programme now available!
Posted 17 Jul 2023 by Louis
- Online Transmitting Science BEAST2 course
Posted 20 Apr 2023 by Bethany

recently updated tutorials

- Skylines plots - Inference of past population dynamics using Bayesian Coalescent Skyline and Birth-Death Skyline plots.
Updated 11 Jun 2024 by tgvaughan
- Time stamped data - Time-stamped data
Updated 9 Jun 2024 by walterxie
- Species Trees and Species Delimitation with SNAPPER - Analyzing SNP or AFLP data
Updated 15 Apr 2024 by bjoelle
- The ClADS package - Inferring progressive changes in birth and death rates
Updated 18 Dec 2023 by Unknown

workshops

- Taming the BEAST Beijing
Beijing, China
July 14, 2025-July 18, 2025
- Squamish BEAST Tamers
Squamish, British Columbia
August 14, 2023-August 18, 2023
- Taming the BEAST Online
Online!
June 07, 2021-June 11, 2021
- Taming the BEAST 2020
Oberägeri, Switzerland
May 31, 2020-June 05, 2020

about contact license

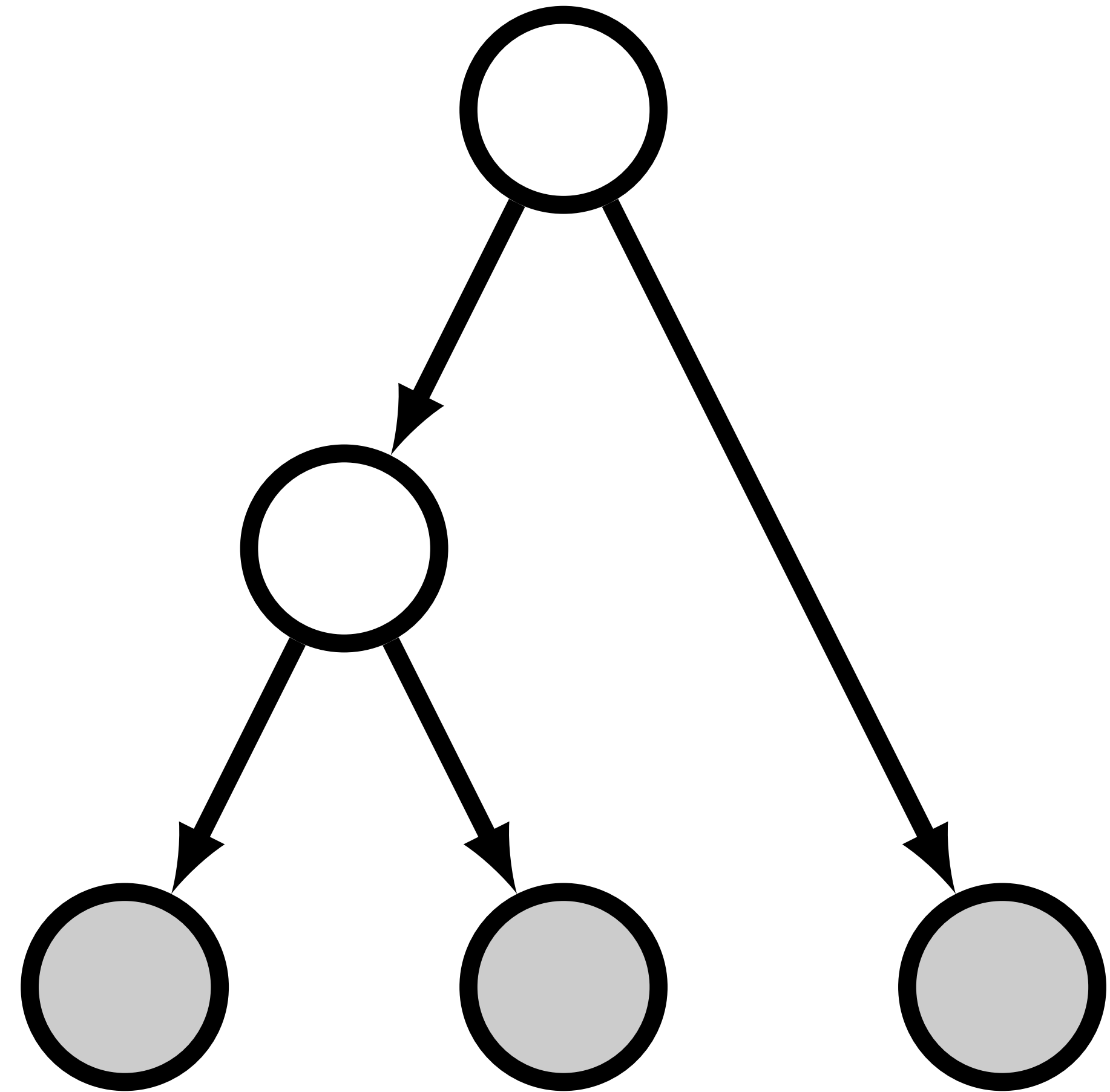
taming-the-beast.org

Graphical models

Graphical models

Provide tools for visually and computationally representing complex, parameter-rich models

Depict the conditional dependence structure of parameters and other random variables



Types of variables (nodes)



a) Constant node

a. fixed value variables



b) Stochastic node

b. random variables that depend on other variables



c) Deterministic node

c. variables determined by a function applied other variables (transformations)



d) Clamped node
(observed)

d. observed stochastic variables (data)



a) Constant node



b) Stochastic node



c) Deterministic node



d) Clamped node
(observed)



e) Plate

a. fixed value variables

b. random variables that depend on other variables

c. variables determined by a function applied other variables (transformations)

d. observed stochastic variables (data)

e. repetition over multiple variables (equivalent to a loop)

Specifying graphical models using the Rev syntax

Table 1: Rev assignment operators, clamp function, and plate/loop syntax.

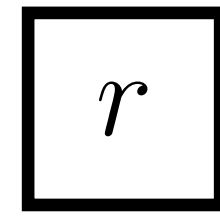
Operator	Variable
<code><-</code>	constant variable
<code>~</code>	stochastic variable
<code>:=</code>	deterministic variable
<code>node.clamp(data)</code>	clamped variable
<code>=</code>	inference (<i>i.e.</i> , non-model) variable
<code>for(i in 1:N){...}</code>	plate

a)

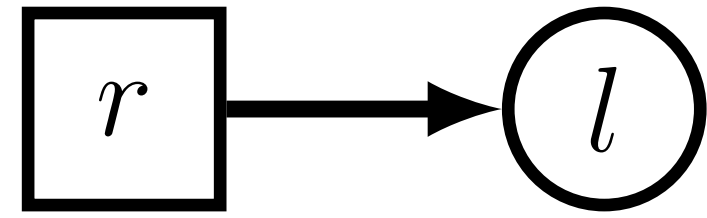
r

```
# constant node  
r <- 10
```

a)

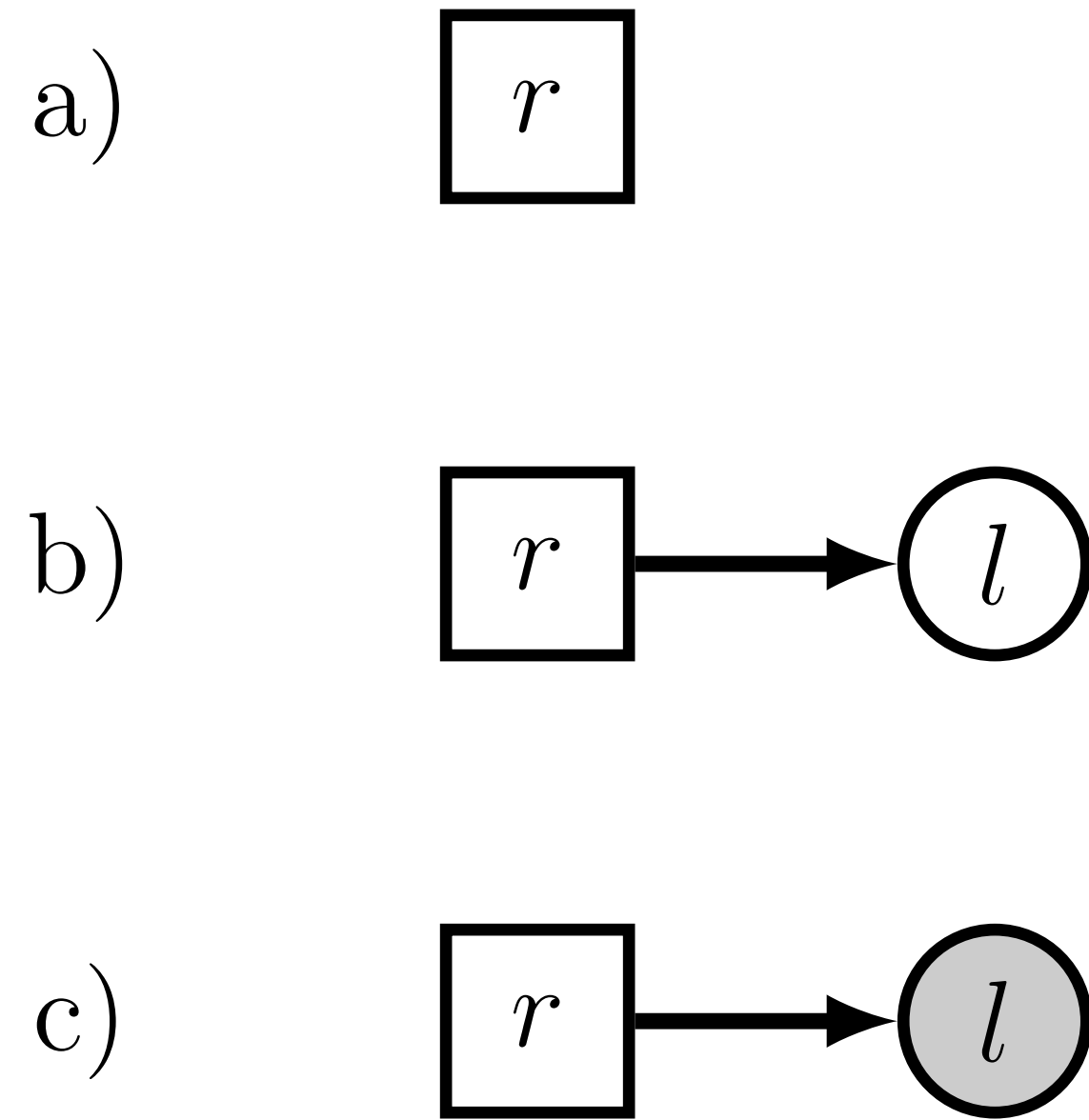


b)



```
# constant node  
r <- 10
```

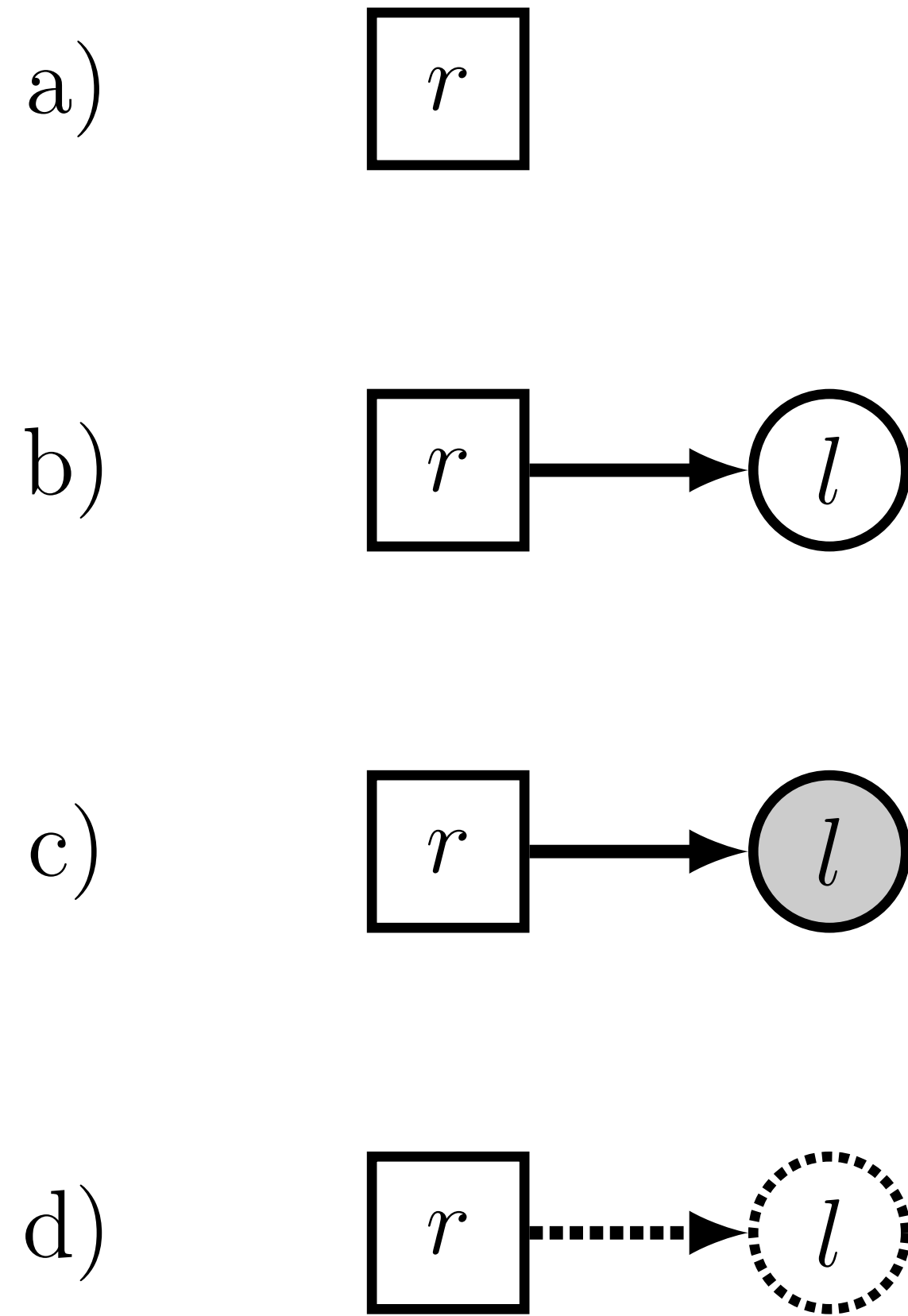
```
# stochastic node  
l ~ dnExp(r)
```



```
# constant node  
r <- 10
```

```
# stochastic node  
l ~ dnExp(r)
```

```
# stochastic node (observed)  
l.clamp(0.1)
```

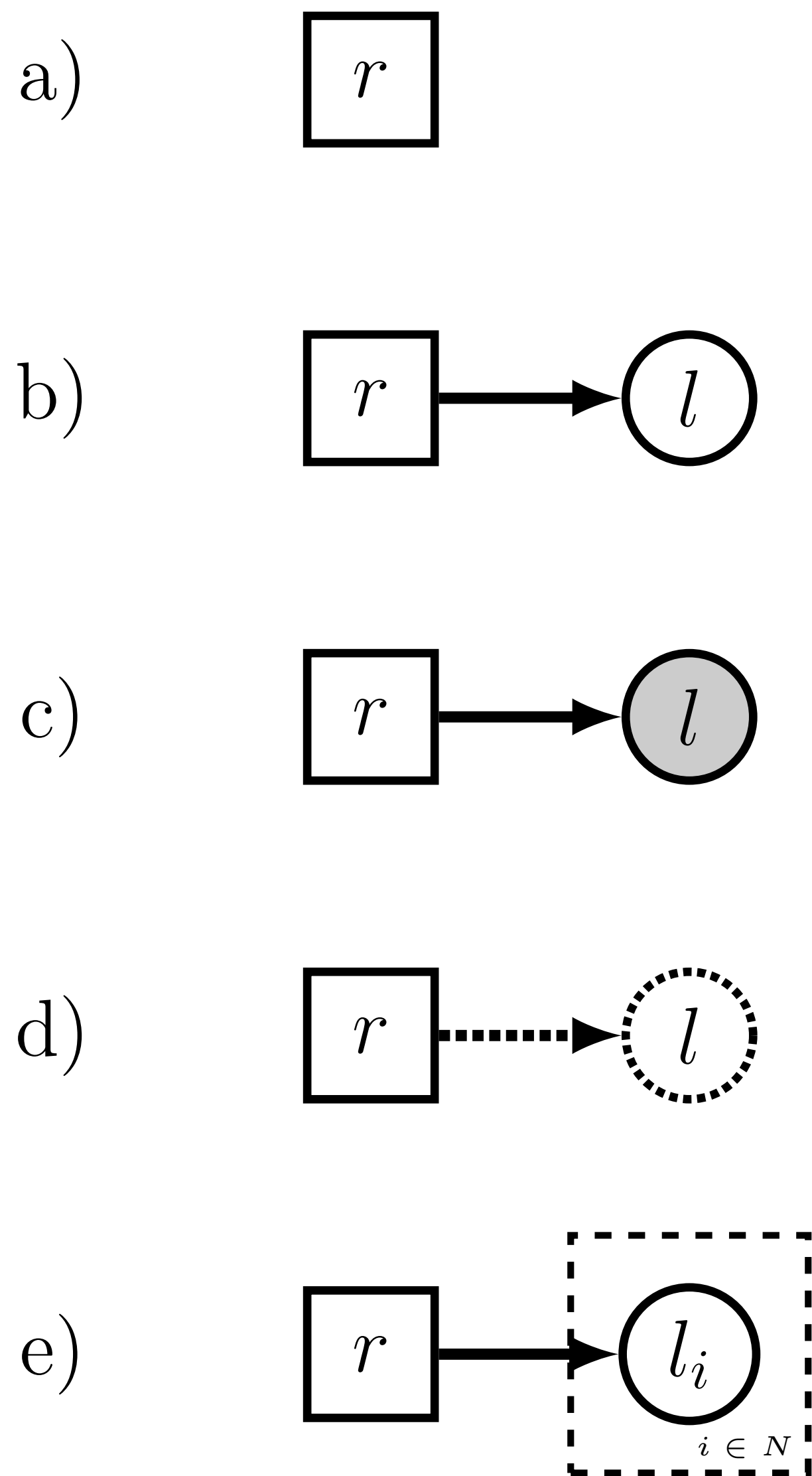


```
# constant node  
r <- 10
```

```
# stochastic node  
l ~ dnExp(r)
```

```
# stochastic node (observed)  
l.clamp(0.1)
```

```
# deterministic node  
l := exp(r)
```



```
# constant node
r <- 10
```

```
# stochastic node
l ~ dnExp(r)
```

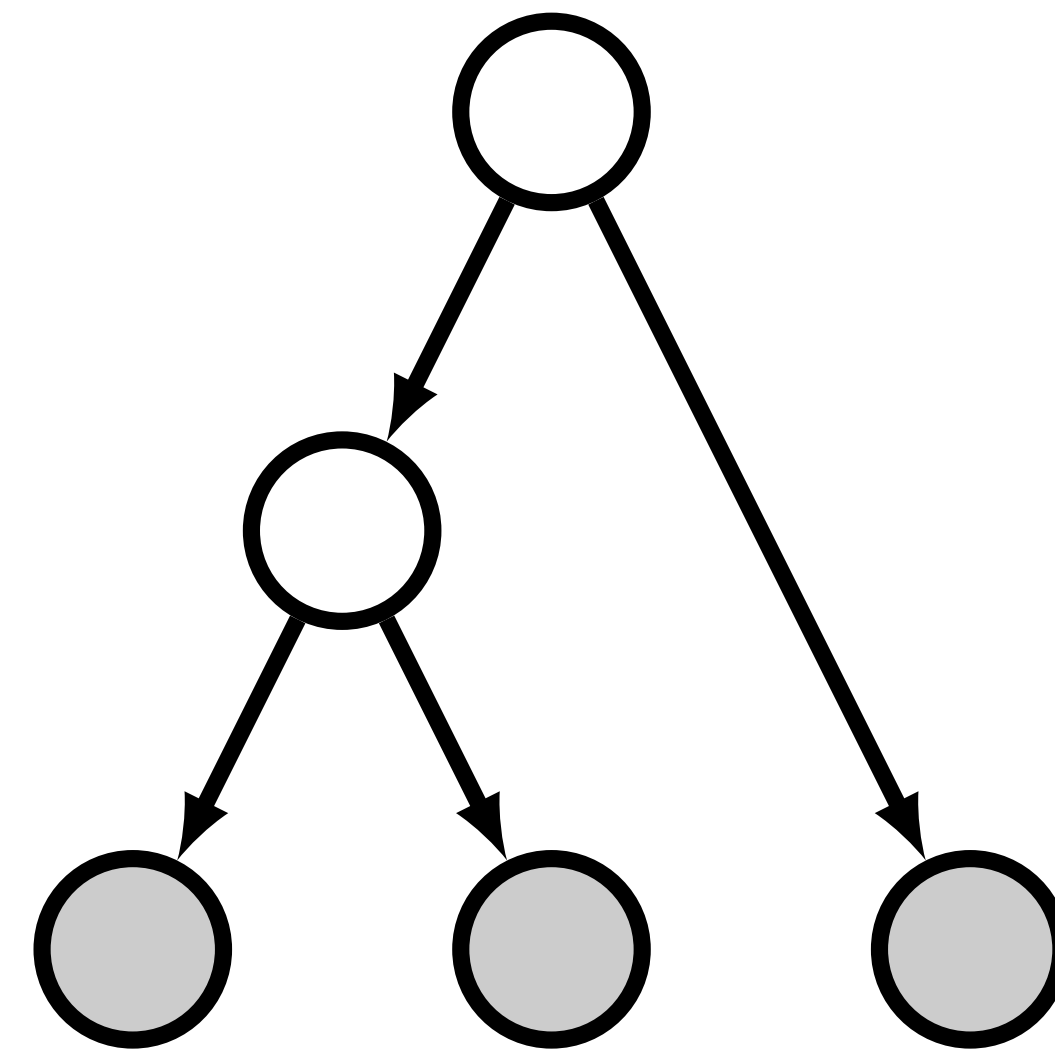
```
# stochastic node (observed)
l.clamp(0.1)
```

```
# deterministic node
l := exp(r)
```

```
# stochastic nodes (iid)
for (i in 1:N) {
  l[i] ~ dnExp(r)
}
```

Running RevBayes demo

Exercise



Bayesian tree inference

Bayes' theorem

$$\Pr(\text{model} \mid \text{data}) = \frac{\Pr(\text{data} \mid \text{model}) \Pr(\text{model})}{\Pr(\text{data})}$$

Bayes' theorem

Likelihood

The probability of the data given the model assumptions and parameter values

$$\Pr(\text{model} \mid \text{data}) = \frac{\Pr(\text{data} \mid \text{model}) \Pr(\text{model})}{\Pr(\text{data})}$$

Bayes' theorem

Priors

This represents our prior knowledge of the model parameters

$$\Pr(\text{model} \mid \text{data}) = \frac{\Pr(\text{data} \mid \text{model}) \Pr(\text{model})}{\Pr(\text{data})}$$

Bayes' theorem

$$\Pr(\text{model} \mid \text{data}) = \frac{\Pr(\text{data} \mid \text{model}) \Pr(\text{model})}{\Pr(\text{data})}$$

$\Pr(\text{data})$

Marginal probability

The probability of the data, given all possible parameter values. Can be thought of as a normalising constant

Bayes' theorem

Reflects our combined knowledge based on the likelihood and the priors

posterior

$\Pr(\text{model} \mid \text{data}) =$

$\Pr(\text{data} \mid \text{model}) \Pr(\text{model})$

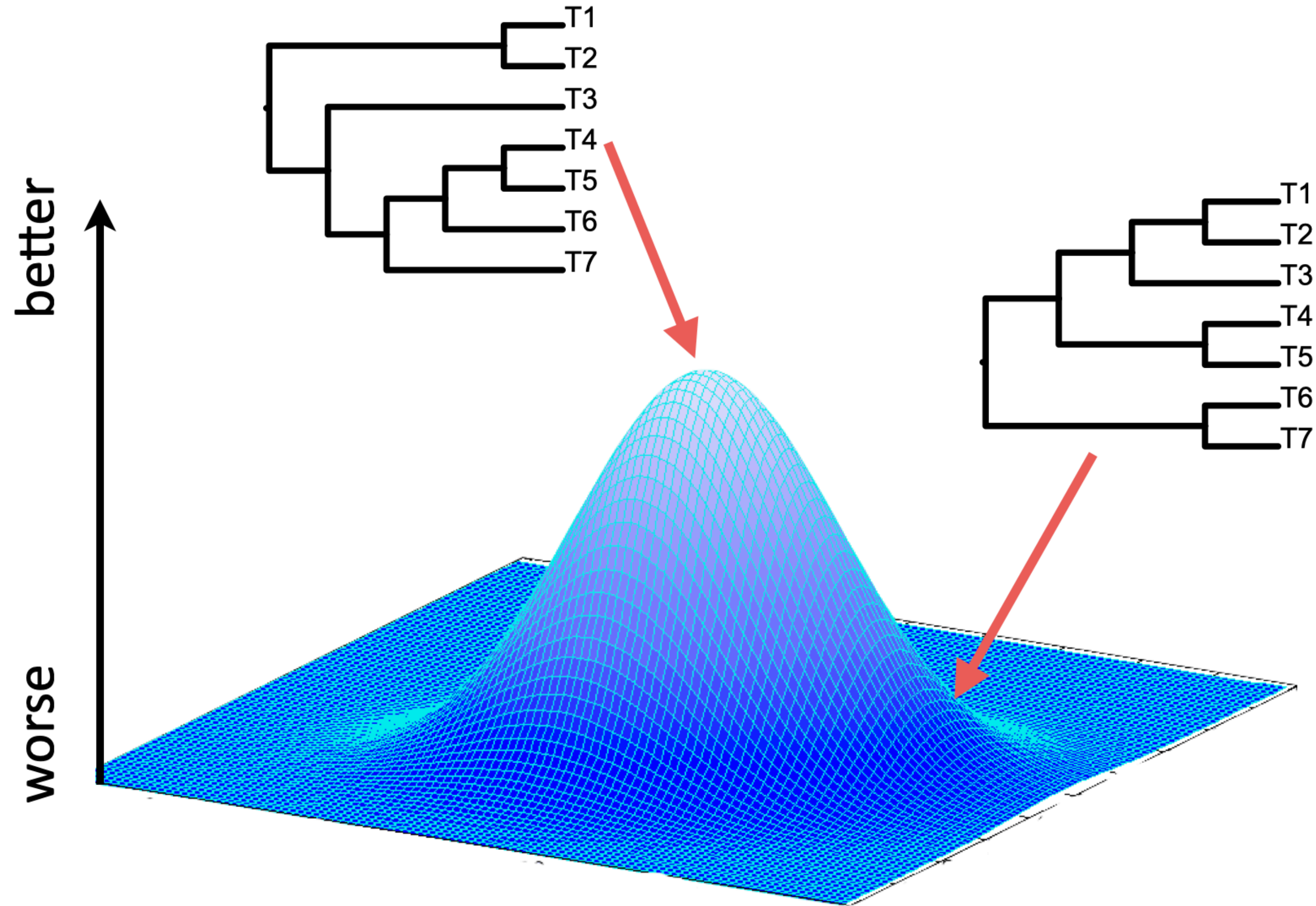
$\Pr(\text{data})$

Bayes' theorem

$$\Pr(\text{model} \mid \text{data}) \propto \Pr(\text{data} \mid \text{model}) \Pr(\text{model})$$

The posterior is proportional to the product of the prior and the likelihood

How do we find the 'best' tree?



It depends how you measure 'best'

Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

Both maximum likelihood and Bayesian inference are model-based approaches

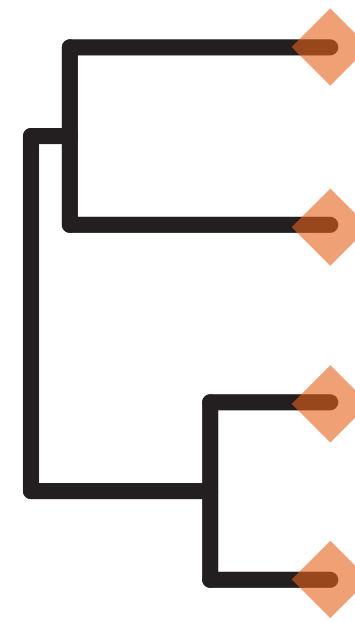
Note these are not the only approaches to tree-building but they are the most widely used

Components used to infer trees

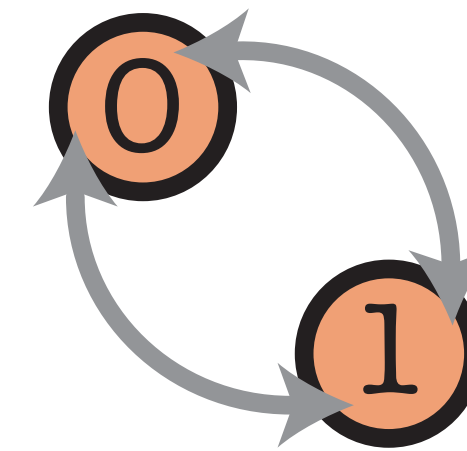
without considering time

0101...
1101...
0100...

data
sequences or
characters



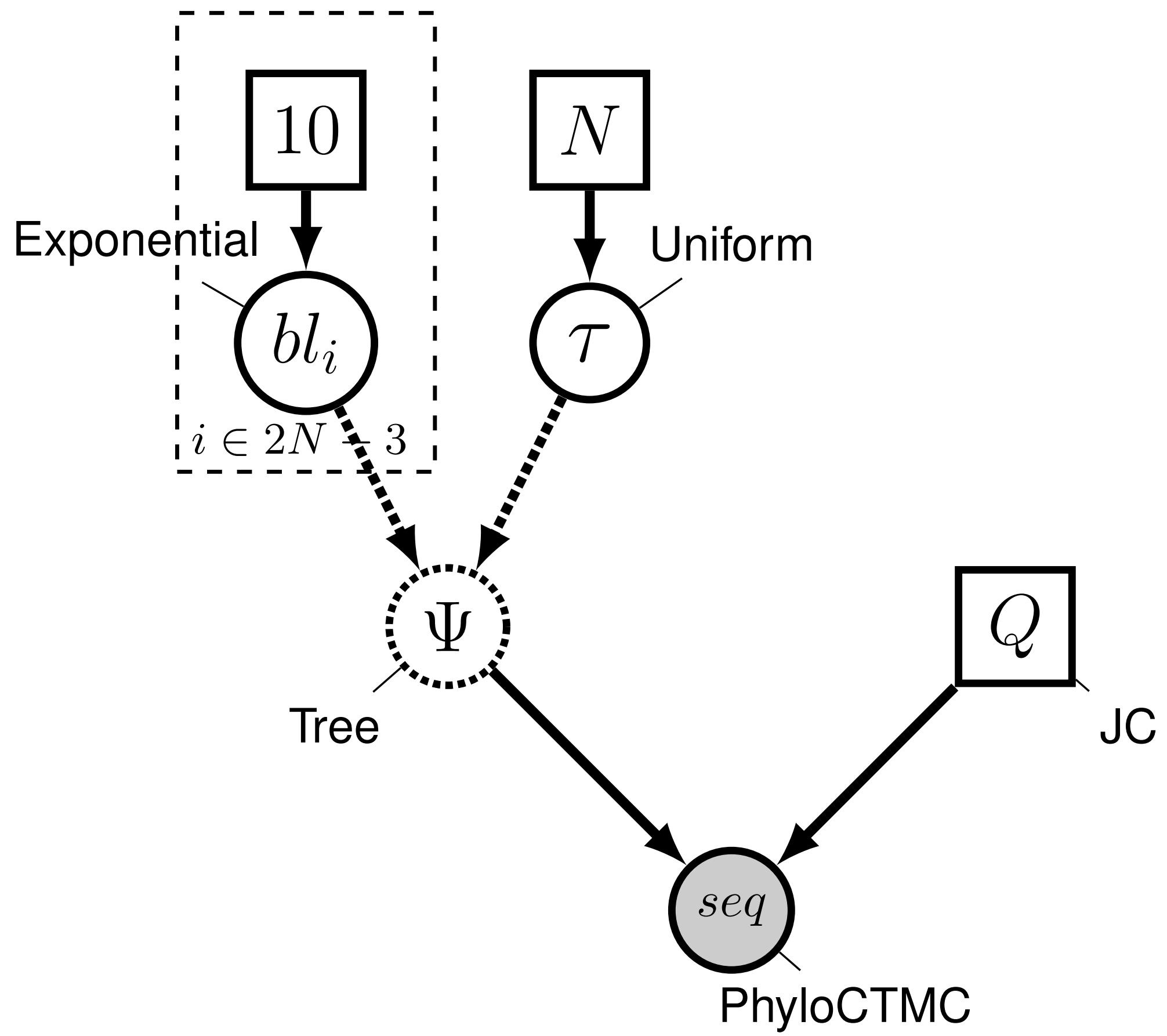
tree
topology and
branch lengths



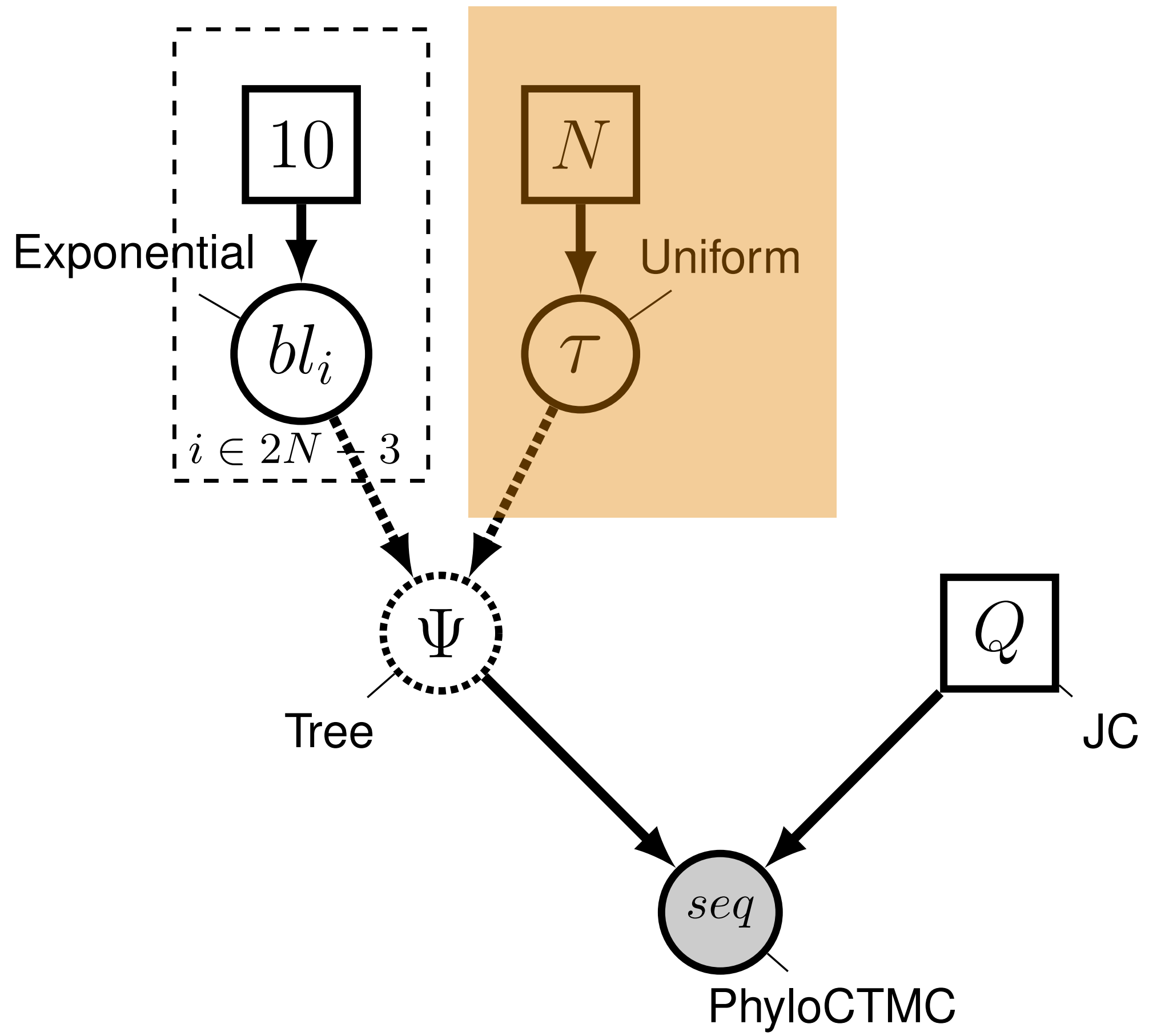
substitution
model

Bayesian tree inference

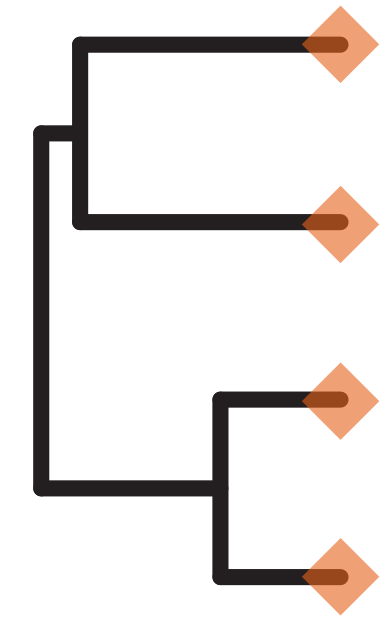
$$\begin{array}{c} \text{posterior} \\ \boxed{\phantom{\text{posterior}}} \\ P(\text{tree} \mid \text{data}) \end{array} = \frac{\begin{array}{c} \text{likelihood} \\ \boxed{\phantom{\text{likelihood}}} \\ P(\text{data} \mid \text{tree}) \end{array} \begin{array}{c} \text{priors} \\ \boxed{\phantom{\text{priors}}} \\ P(\text{tree}) \end{array}}{\begin{array}{c} \text{marginal probability} \\ \boxed{\phantom{\text{marginal probability}}} \\ P(\text{data}) \end{array}}$$

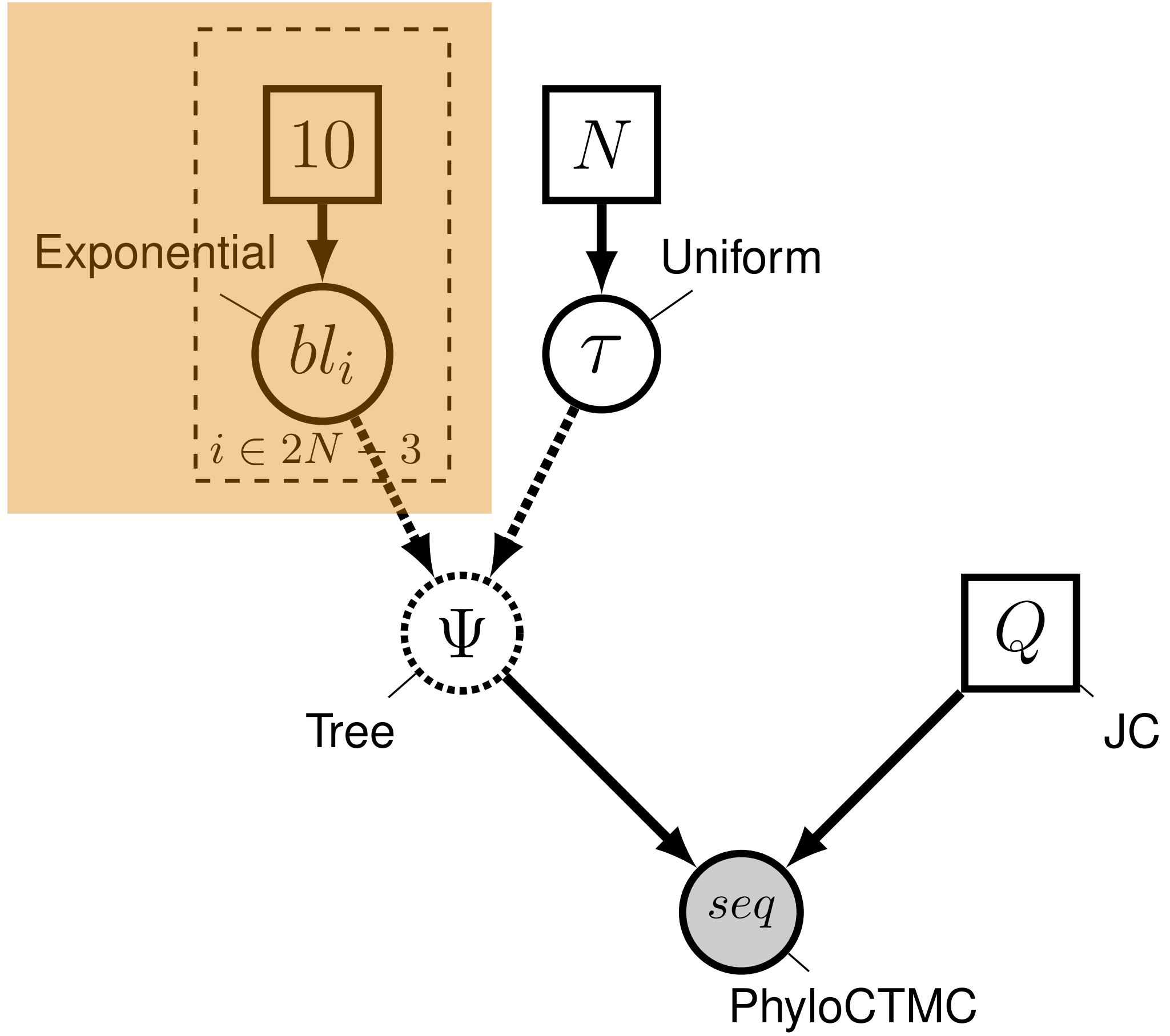


s

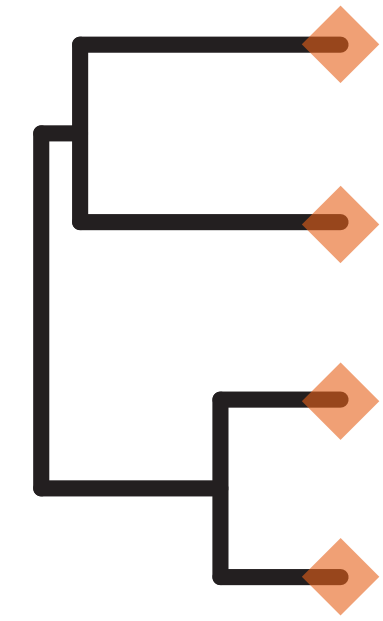


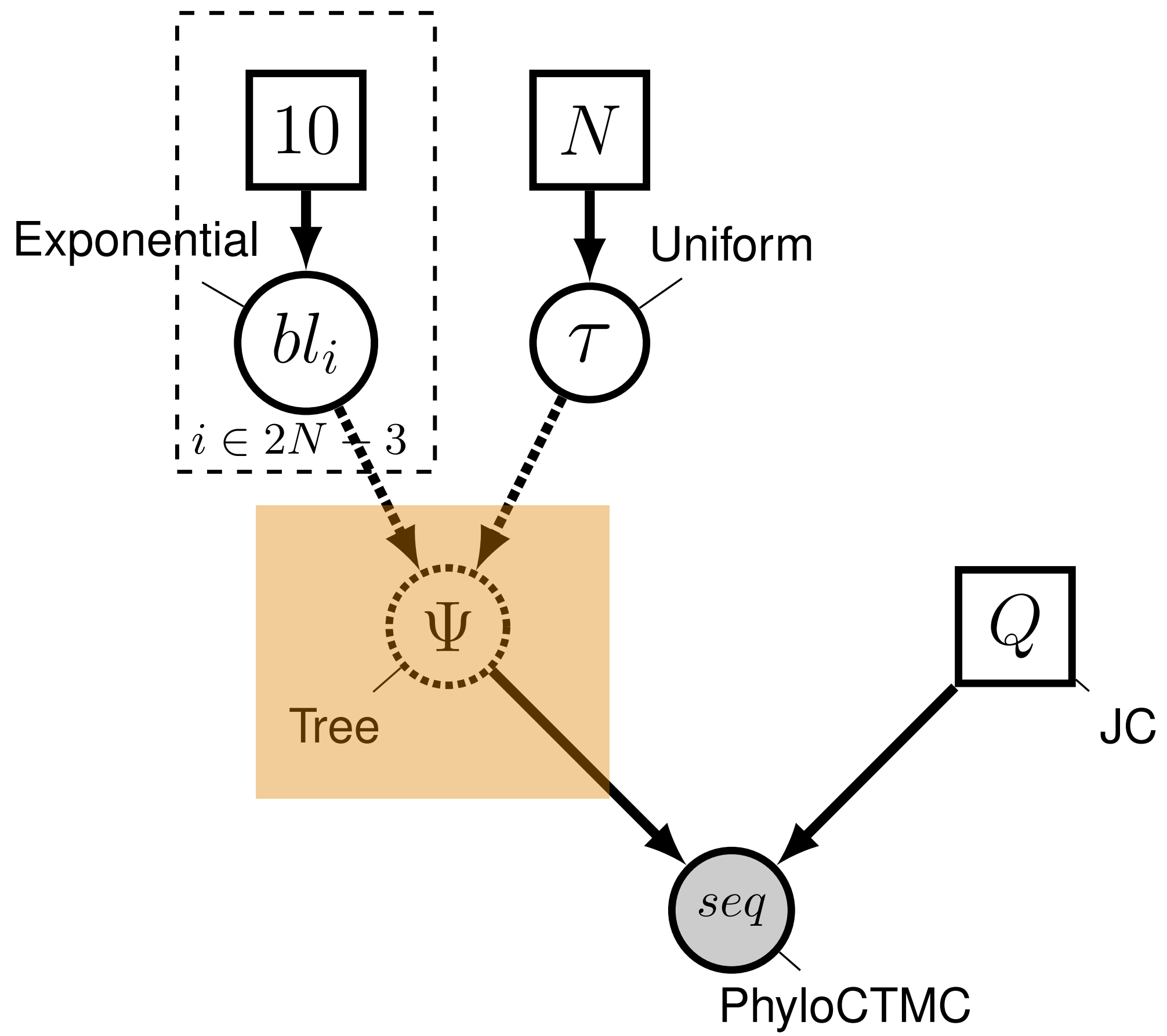
prior on the tree topology



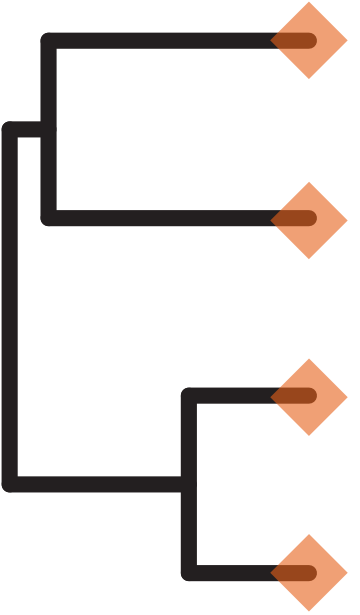


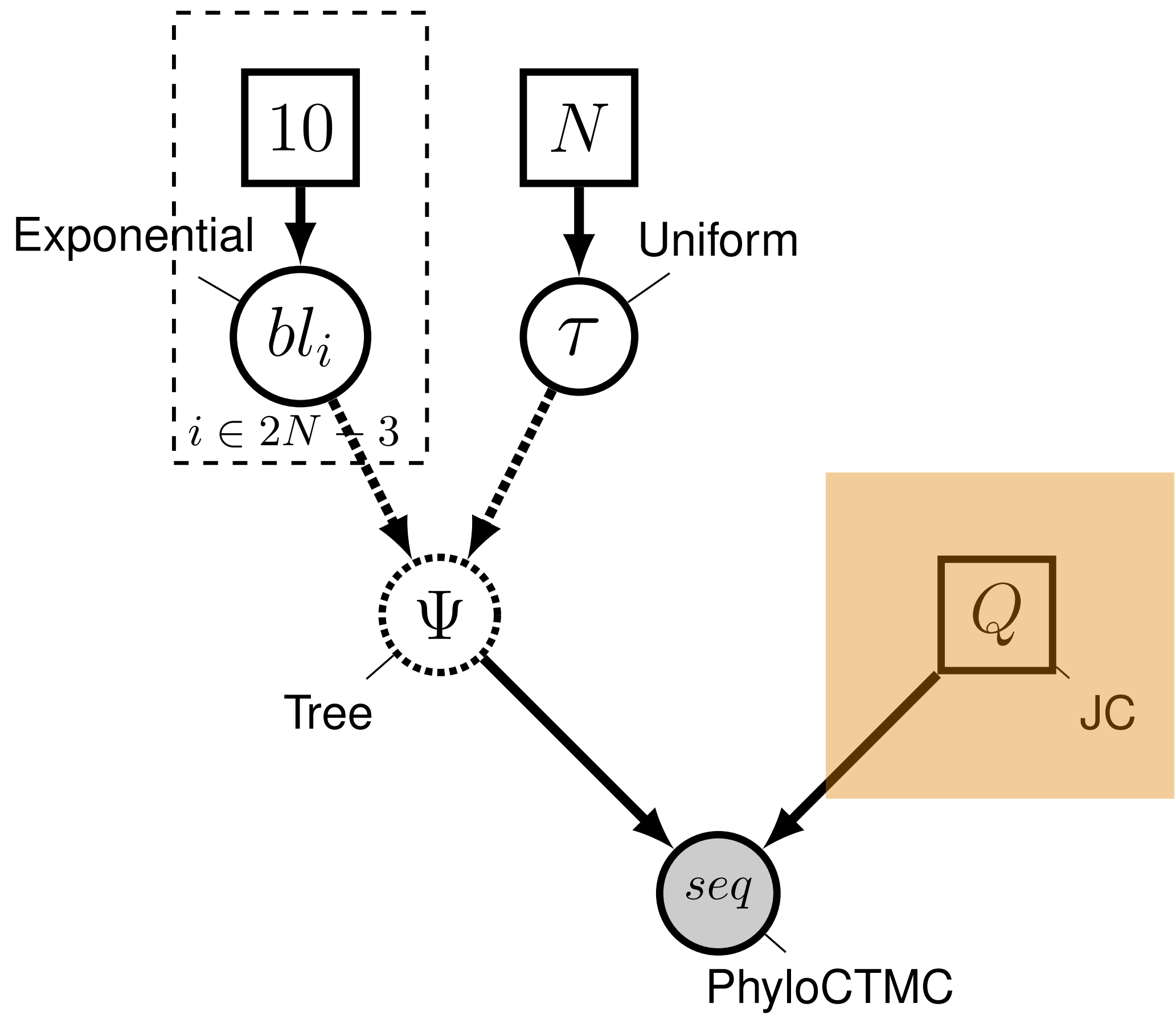
prior on the branch lengths



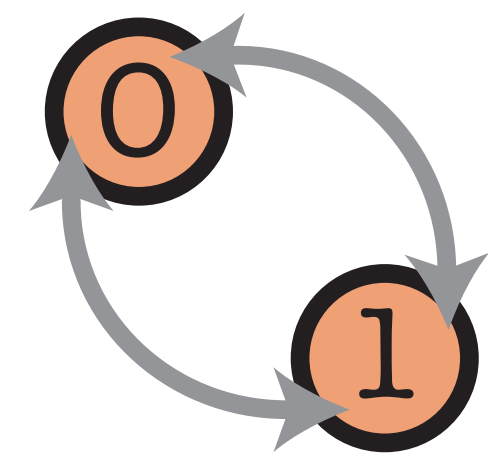


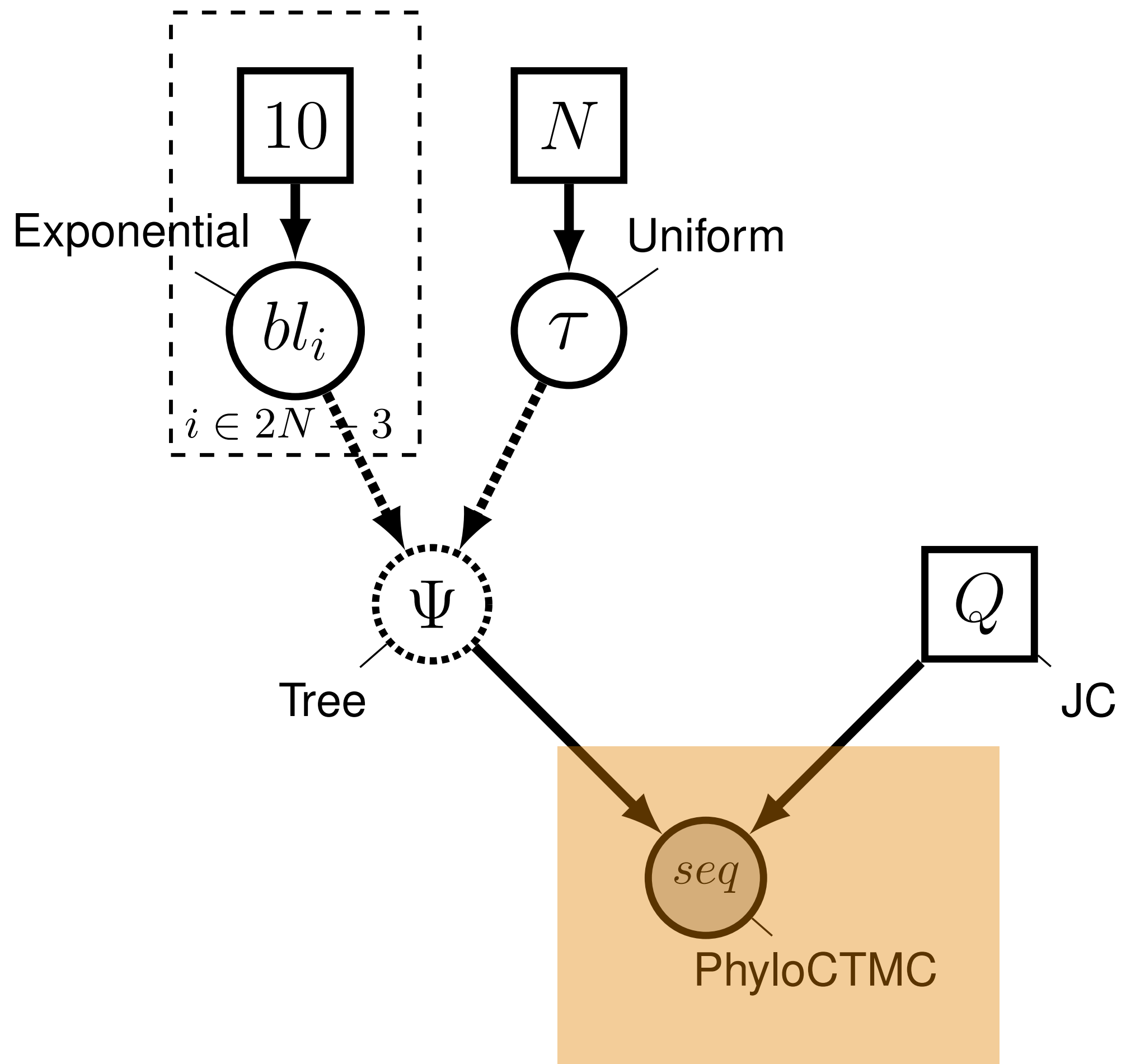
we can combine the topology and branch lengths





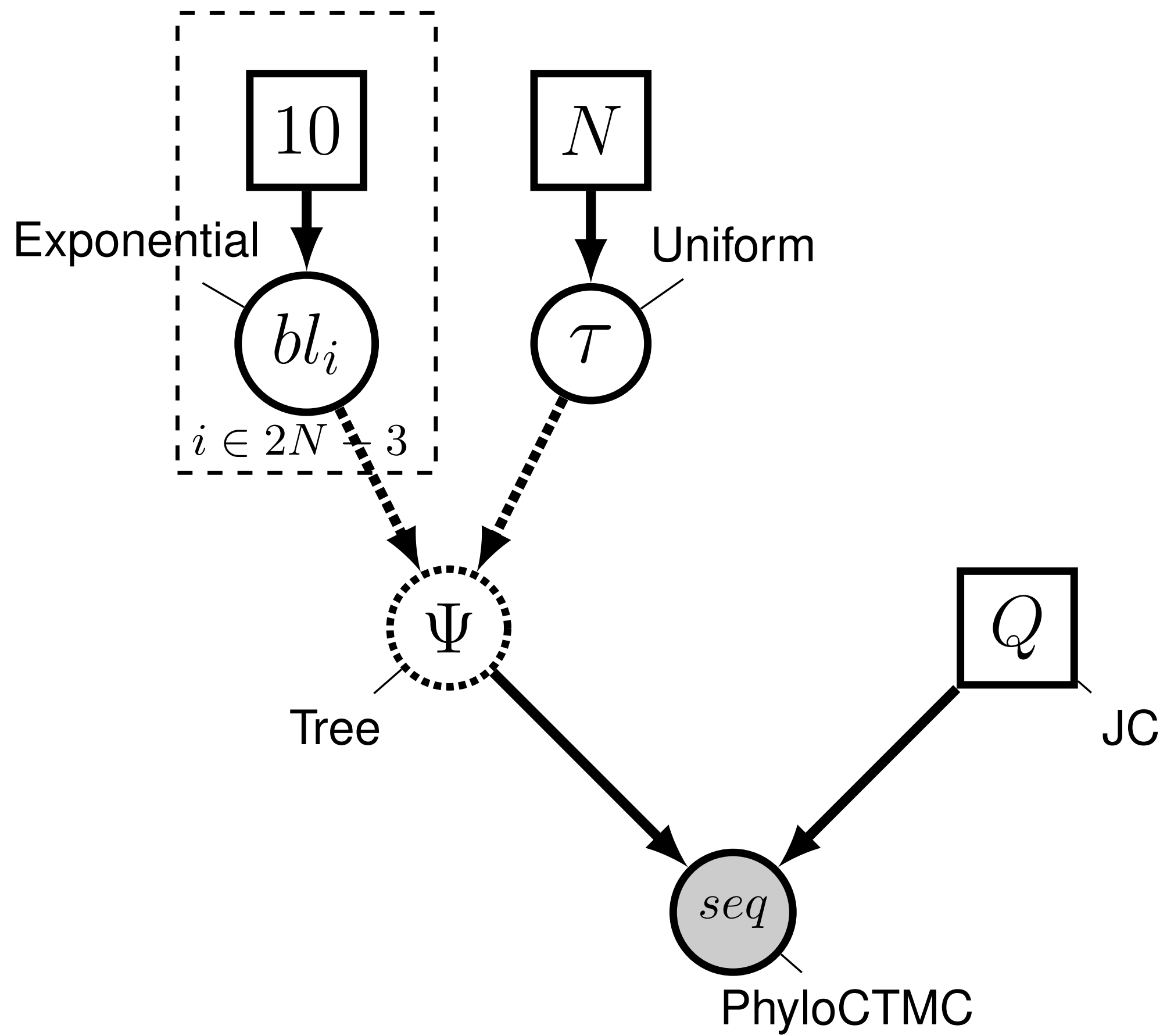
Substitution model





Observed data

0101...
1101...
0100...



```

for (i in 1:n_branches) {
  bl[i] ~ dnExponential(10.0)
}
topology ~ dnUniformTopology(taxa)
psi := treeAssembly(topology, bl)

Q <- fnJC(4)

seq ~ dnPhyloCTMC( tree=psi, Q=Q, type="DNA" )
seq.clamp( data )

```

