Phylogenetics

Introduction to Bayesian inference

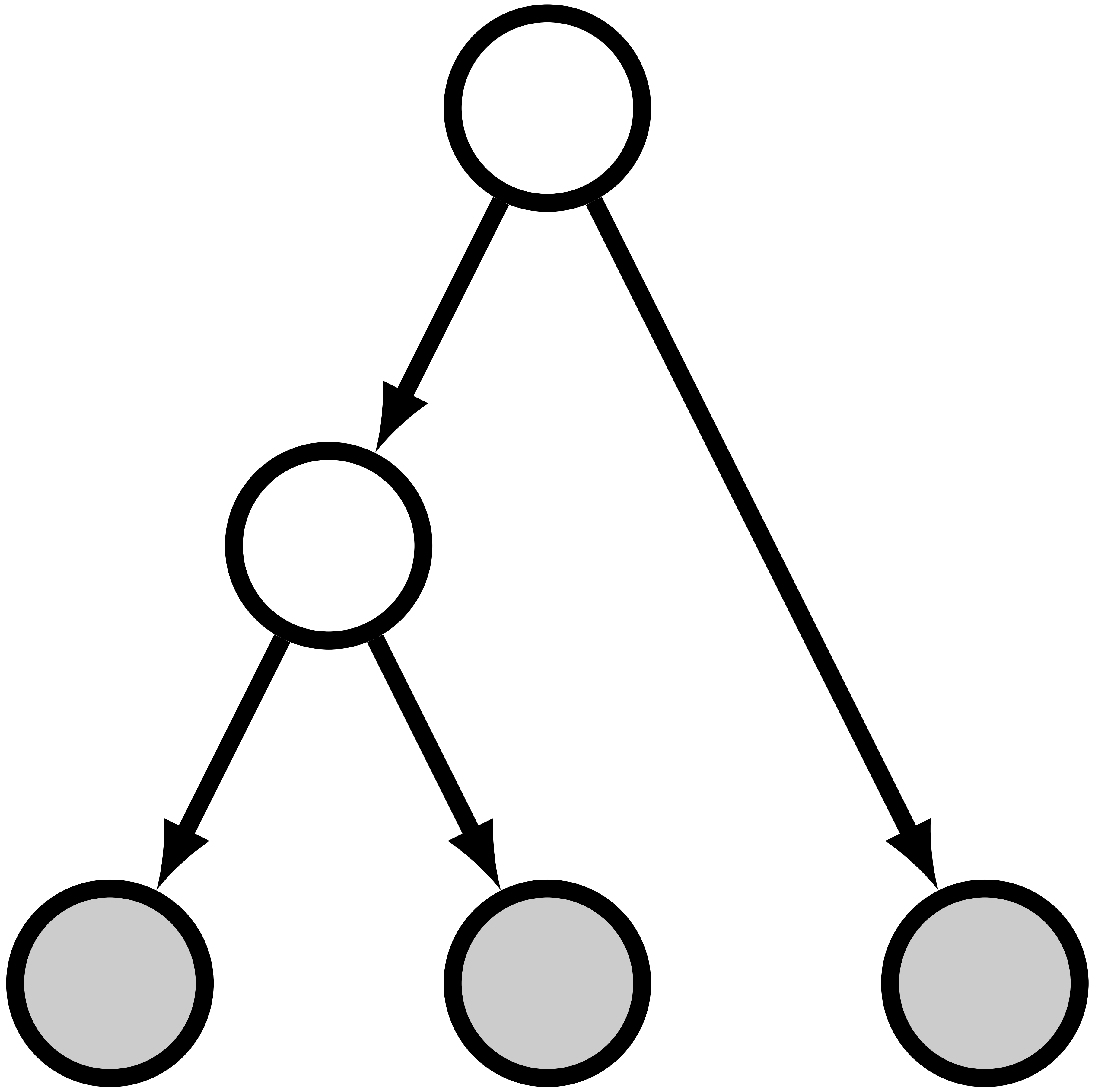
RL-V3 MPP

Rachel Warnock

02.06.26

Objectives

- Bayesian inference
- MCMC
- MrBayes



Recap

How do we find the 'best' tree?

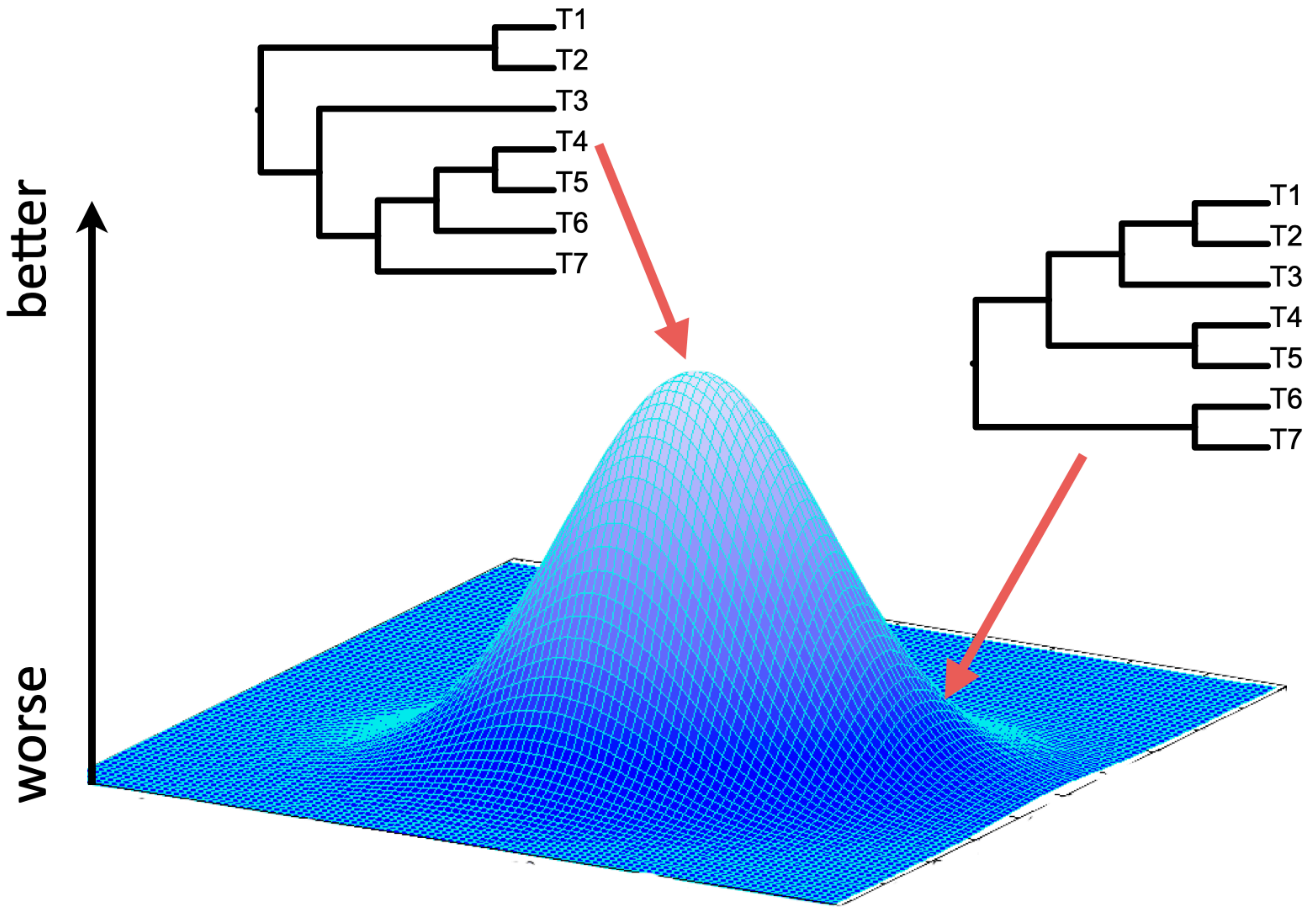


Image source Tracy Heath

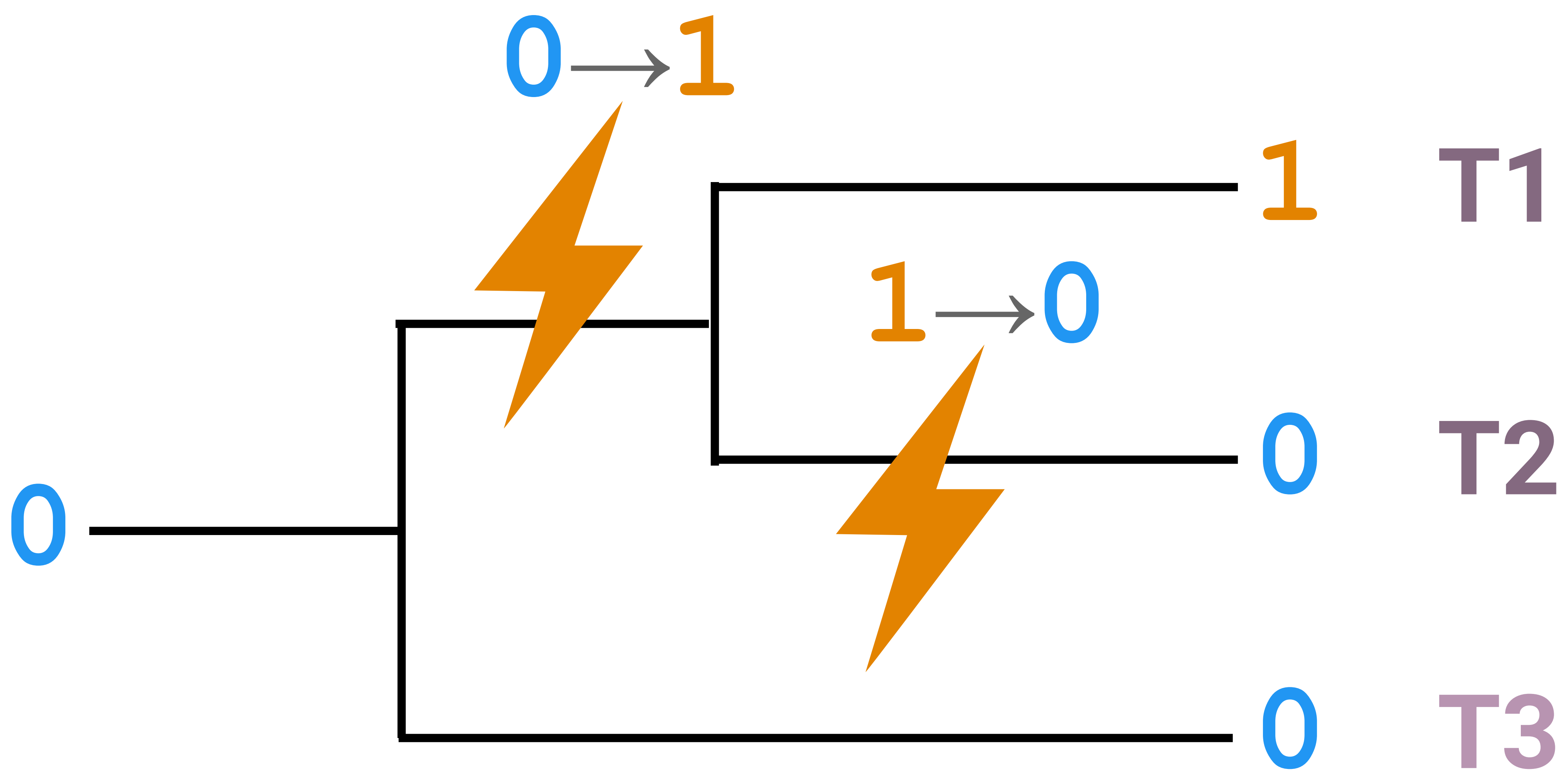
It depends how you measure ‘best’

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 → 1 → 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 ranger of applications in paleobio)

It depends how you measure ‘best’

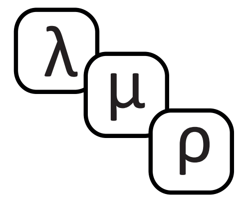
Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

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

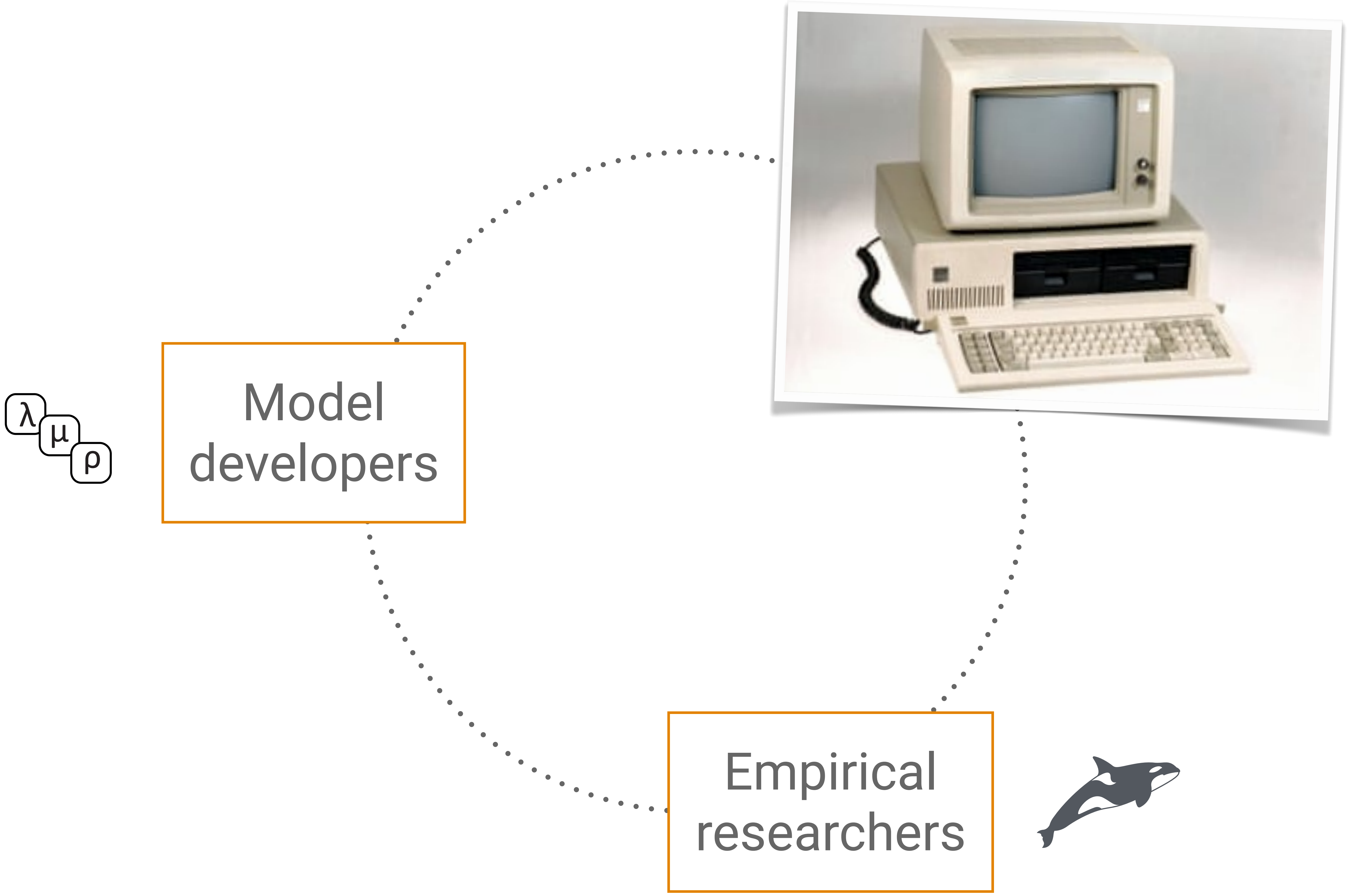


Model developers

What we might call a **"black box"** approach

Empirical researchers

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

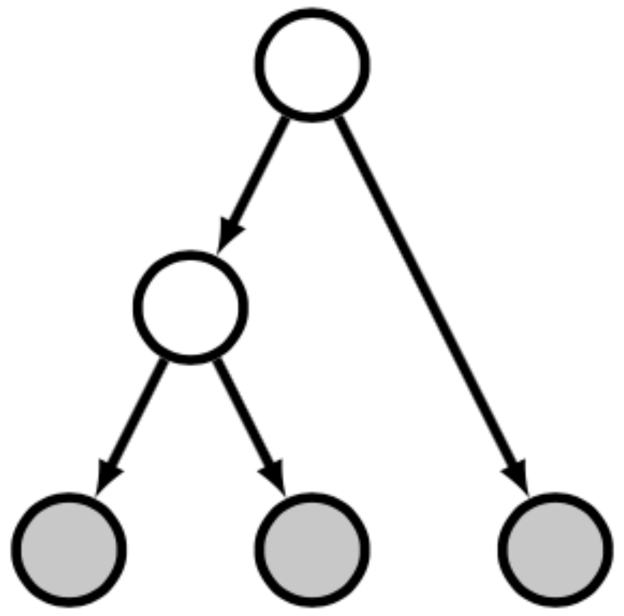
Designed with extendability and flexibility in mind

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

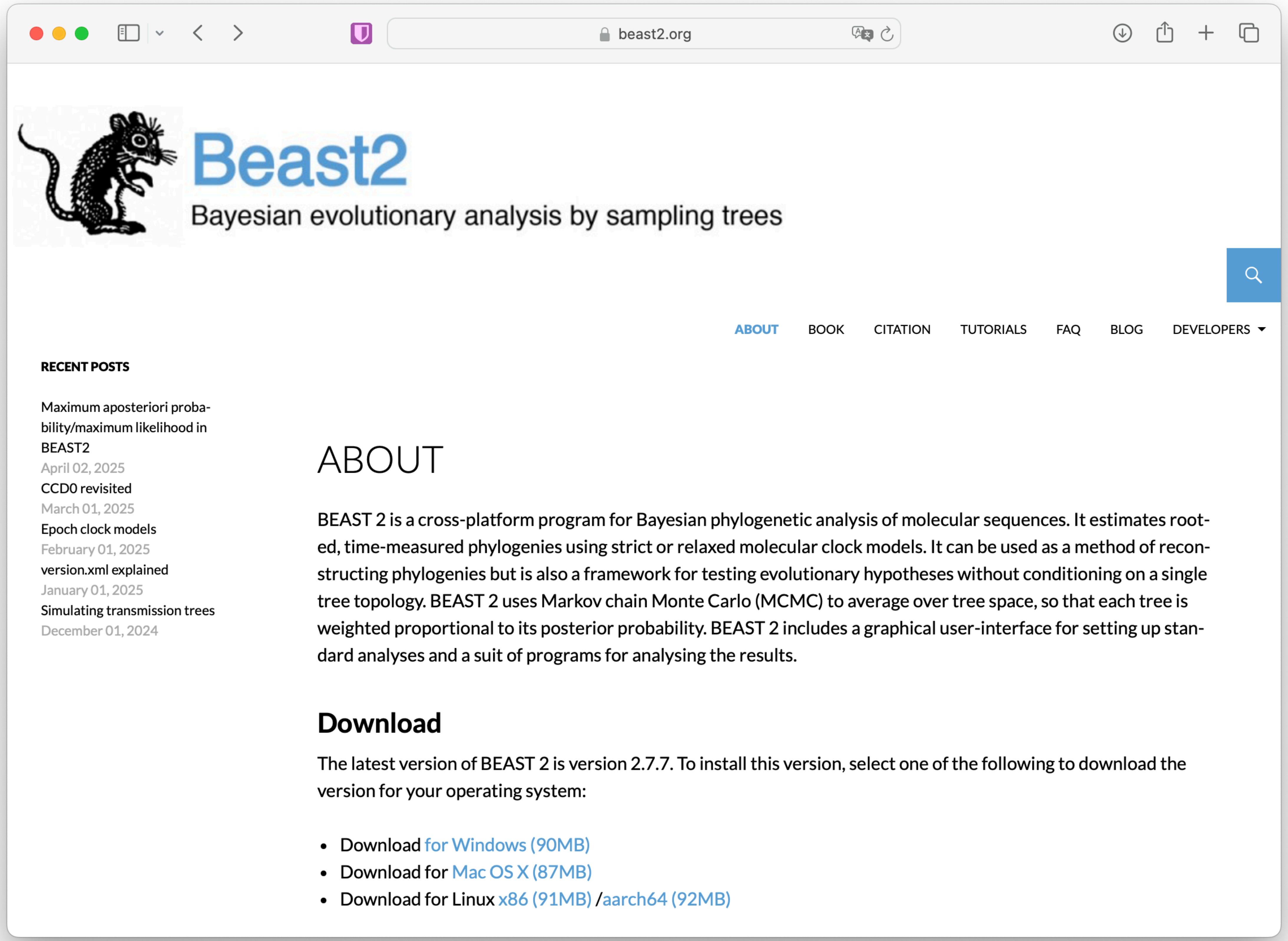
RevBayes provides an interactive environment for statistical computation in phylogenetics. It is primarily intended for modeling, simulation, and Bayesian inference in evolutionary biology, particularly phylogenetics. However, the environment is quite general and can be useful for many complex modeling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

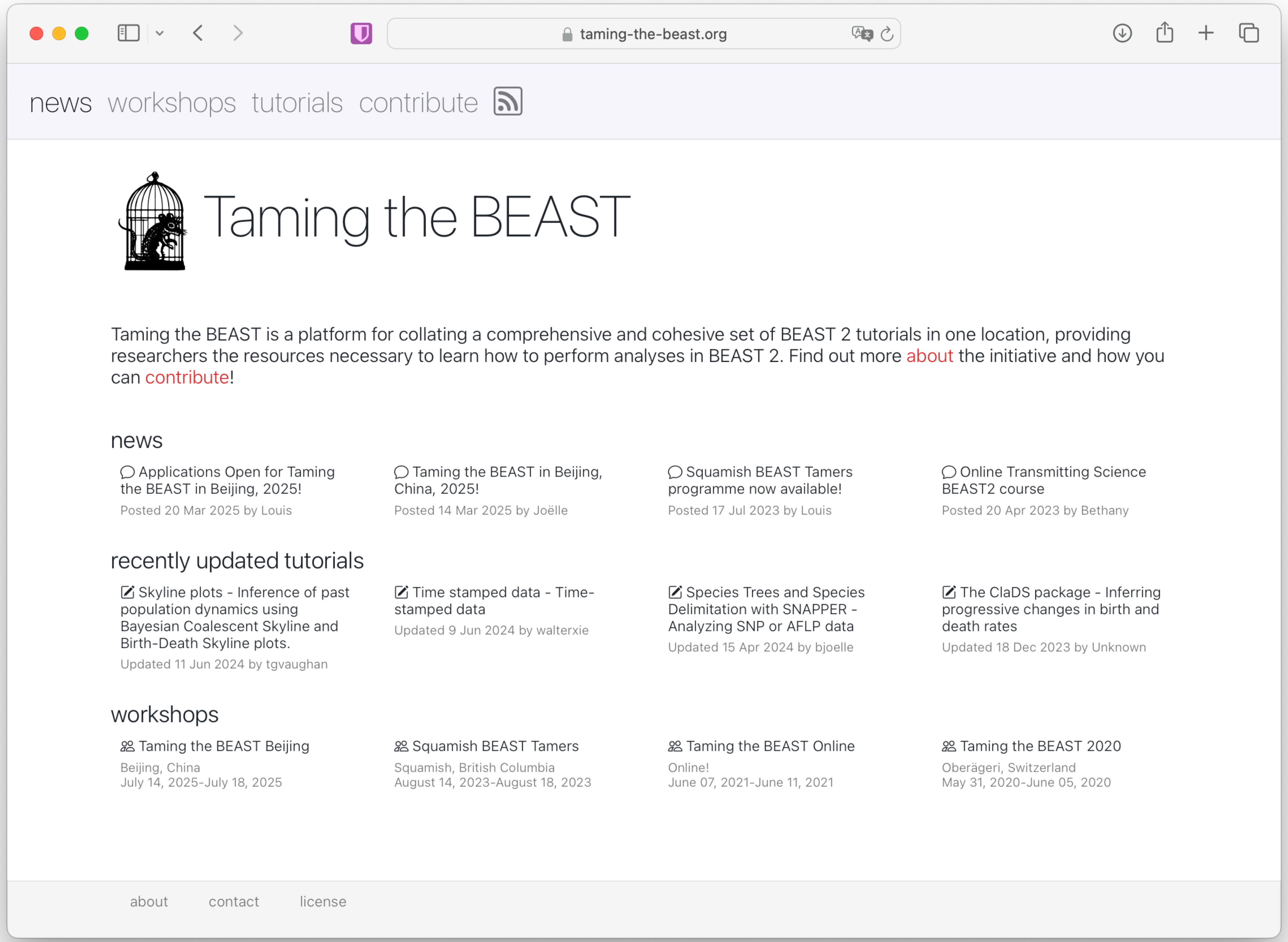
RevBayes uses its own language, Rev, which is a probabilistic programming language like JAGS, STAN, Edward, PyMC3, and related software. However, phylogenetic models require inference machinery and distributions that are unavailable in these other tools.

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





www.beast2.org

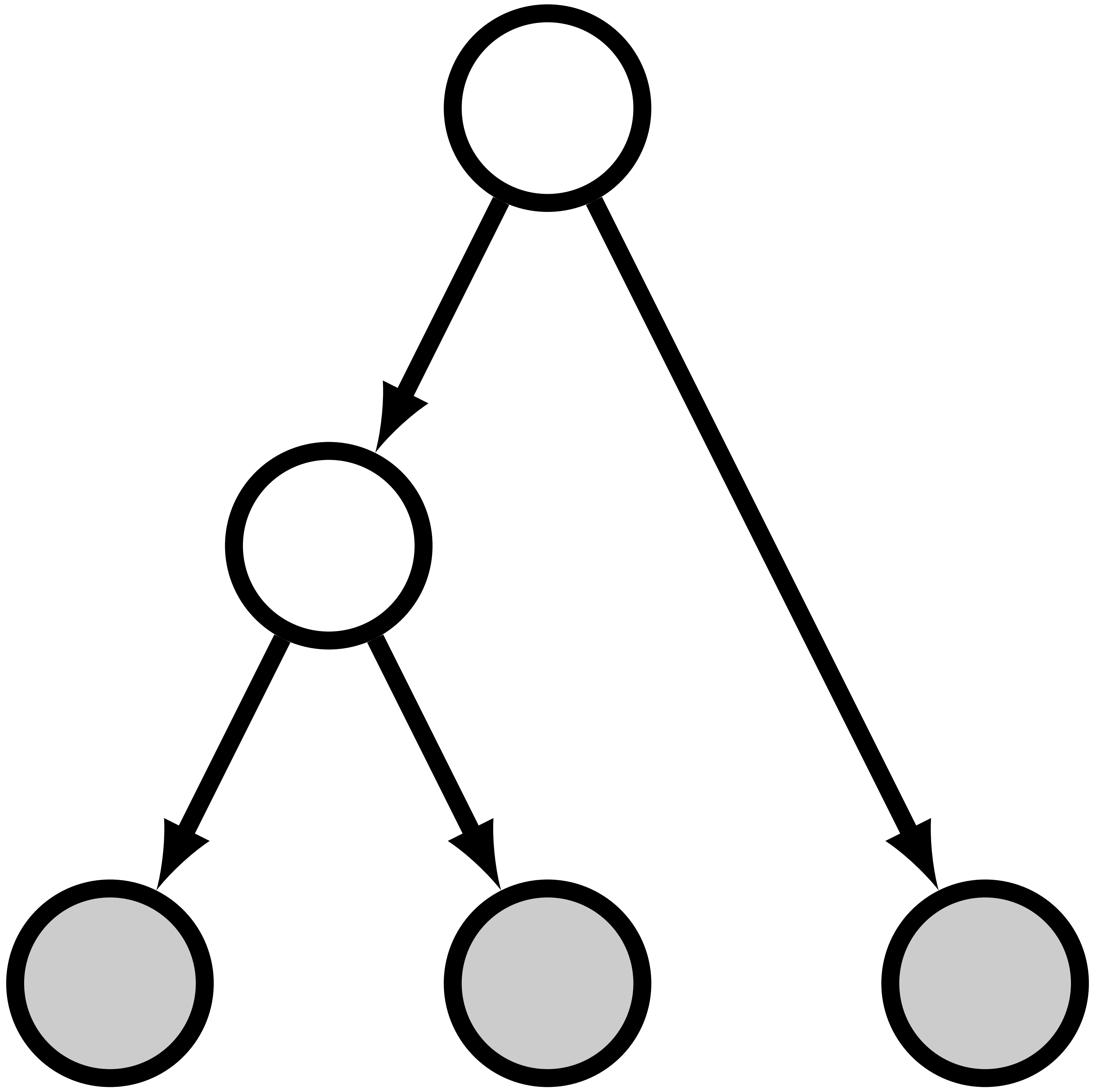
taming-the-beast.org

Graphical models

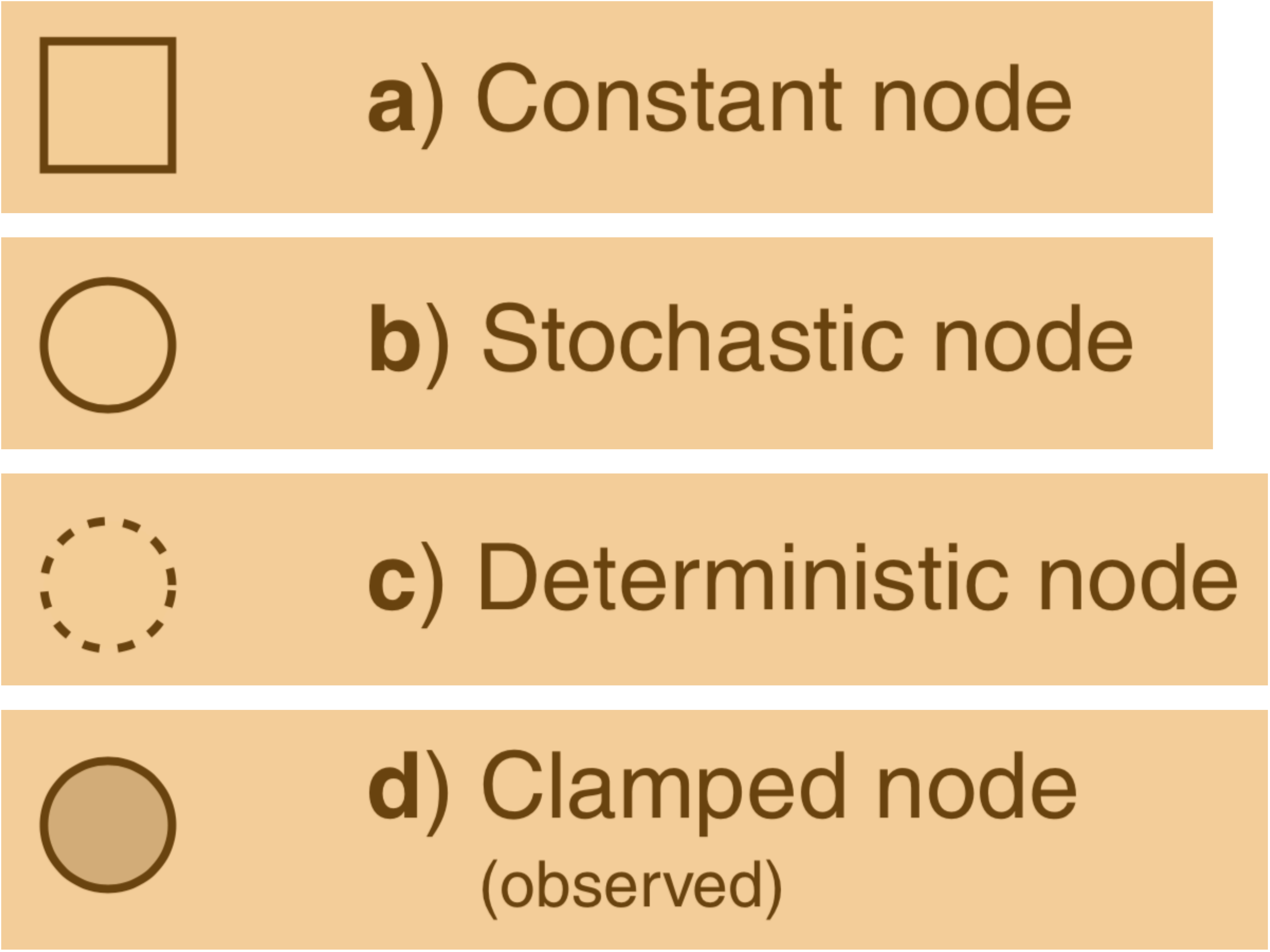
Graphical models

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)

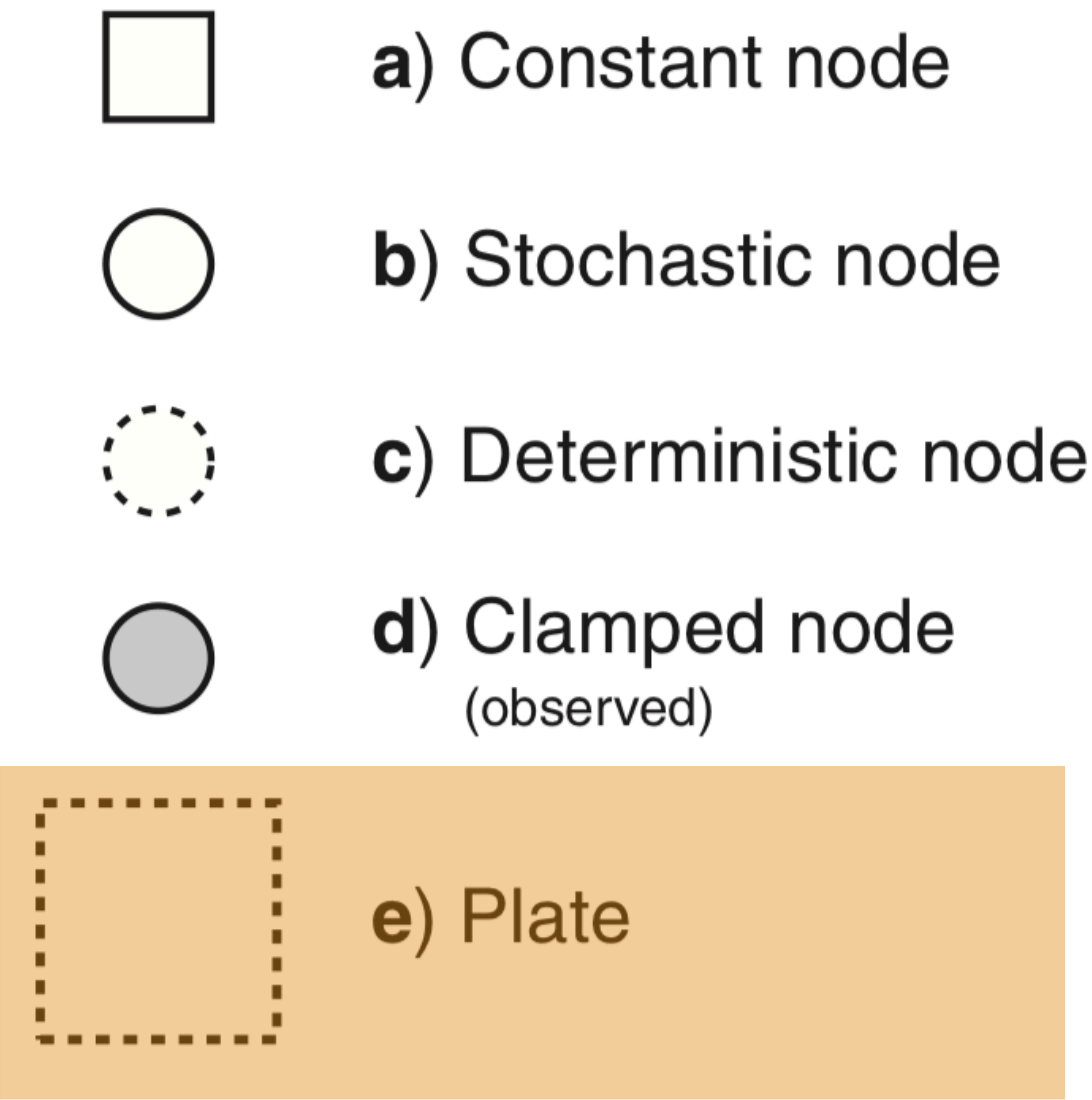


a. fixed value variables

b. random variables that depend on other variables

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

d. observed stochastic variables (data)



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.

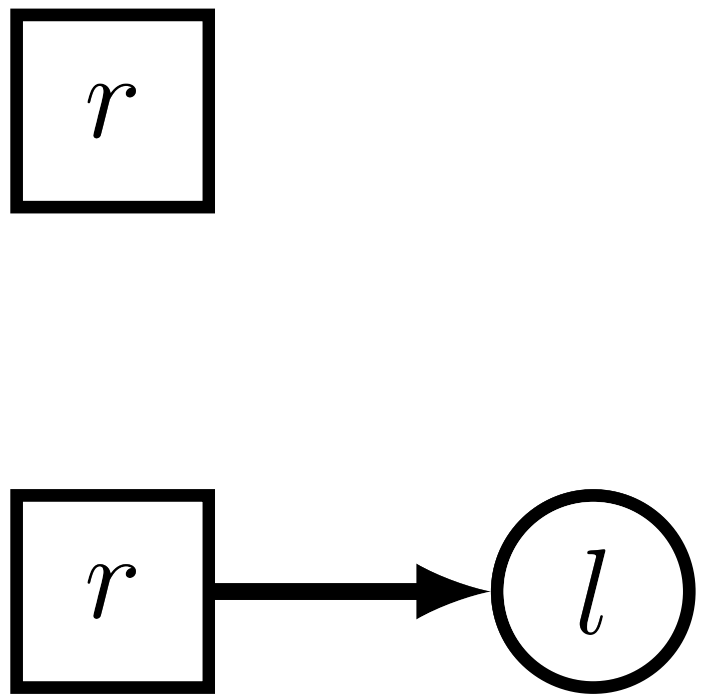
Operator	**Variable**
`<-`	constant variable
`~`	stochastic variable
`:=`	deterministic variable
`node.clamp(data)`	clamped variable
`=`	inference (*i.e.*, non-model) variable
`for(i in 1:N){...}`	plate

a) $\boxed{r}$

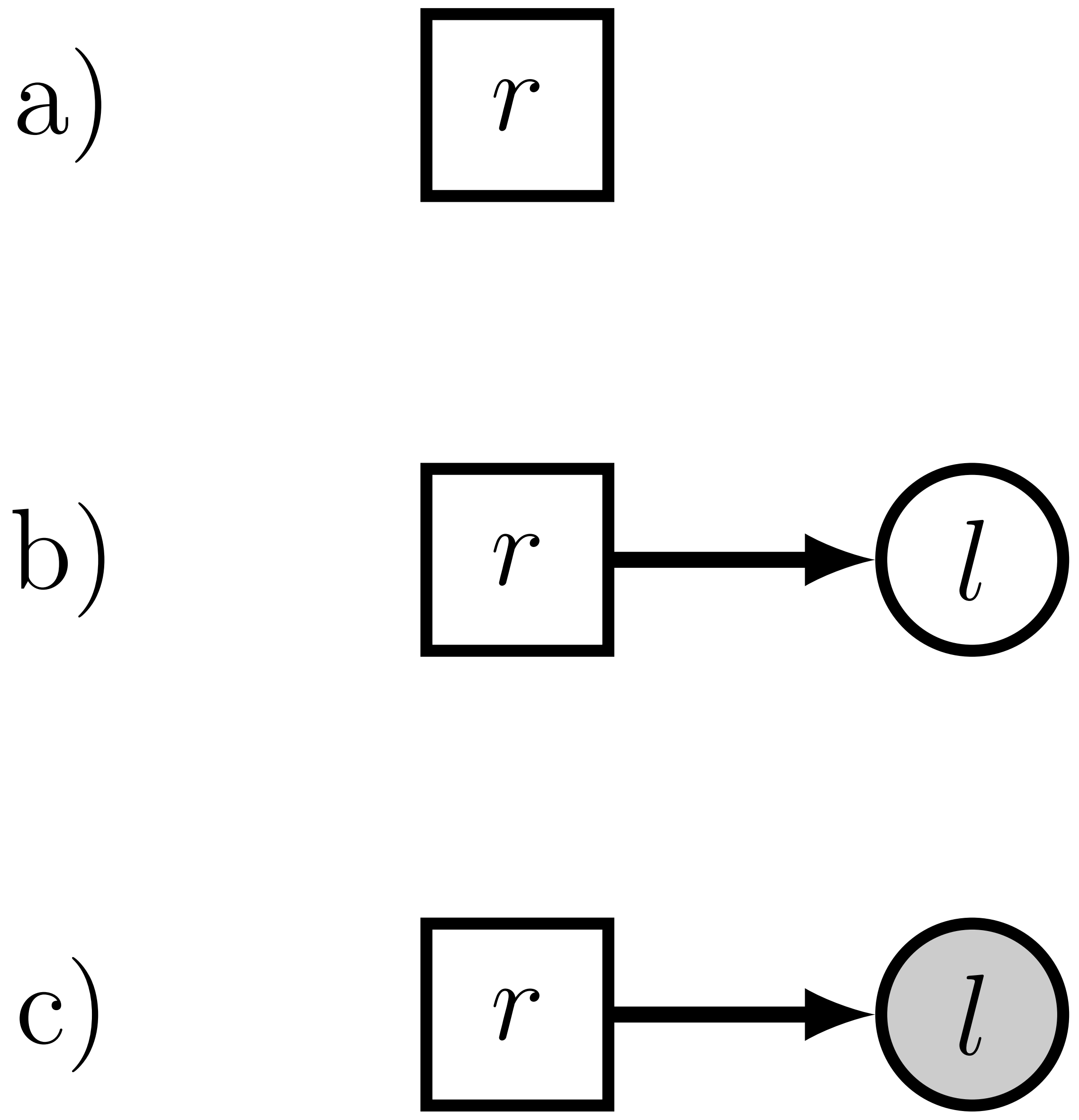
```
# constant node
r <- 10
```

a)

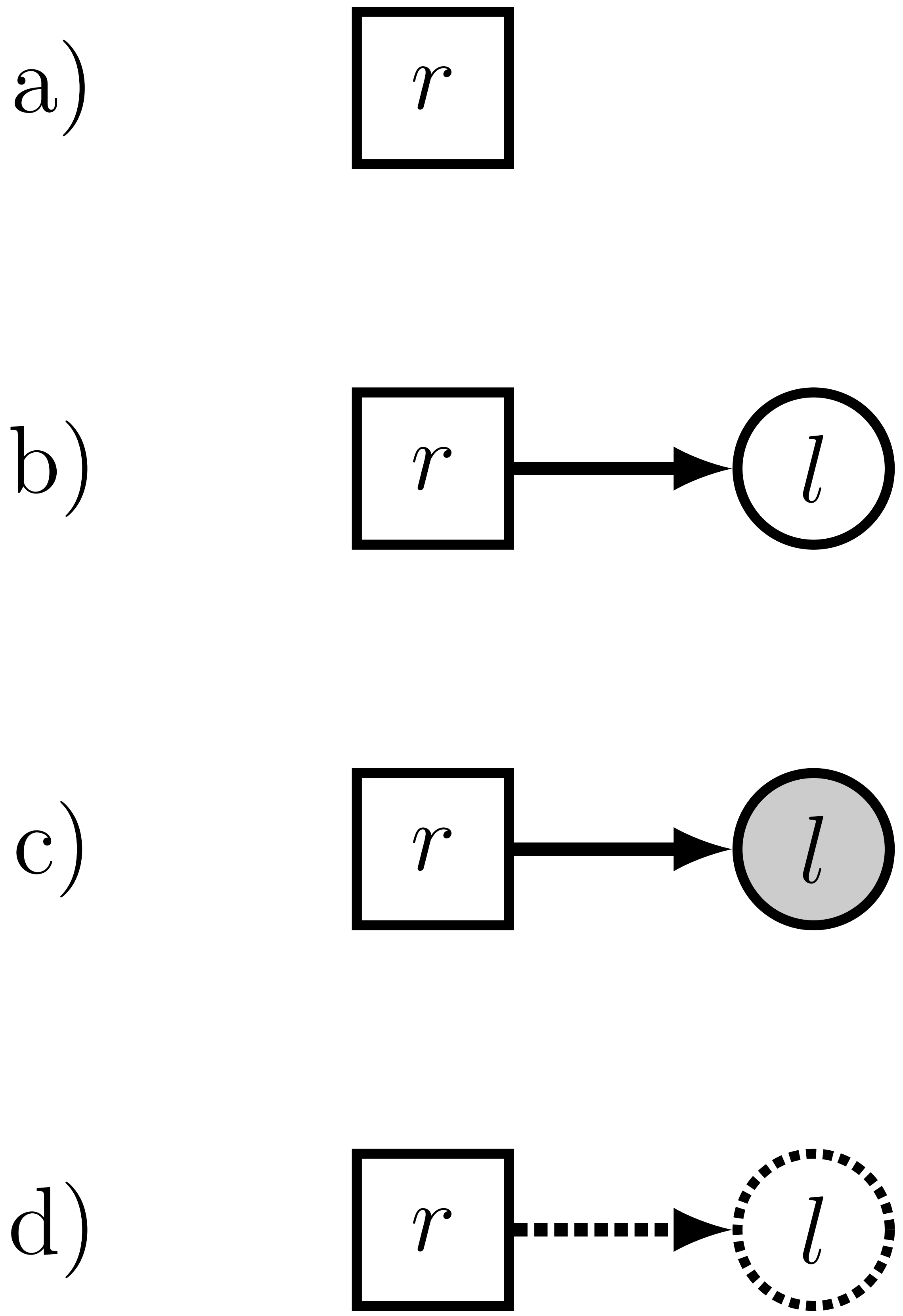
b)



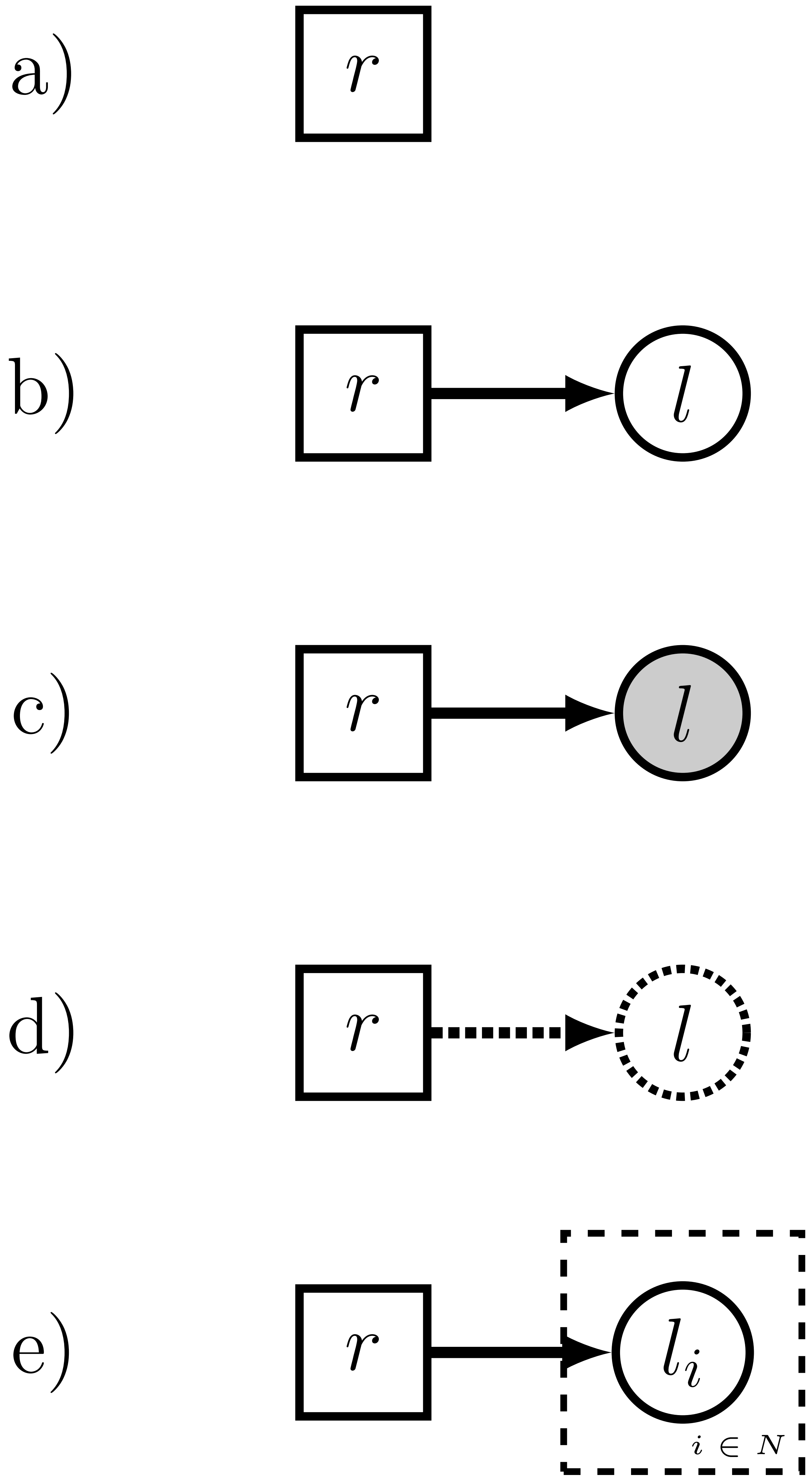

```
# constant node
r <- 10


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




```
# constant node
r <- 10

# stochastic node
l ~ dnExp(r)

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



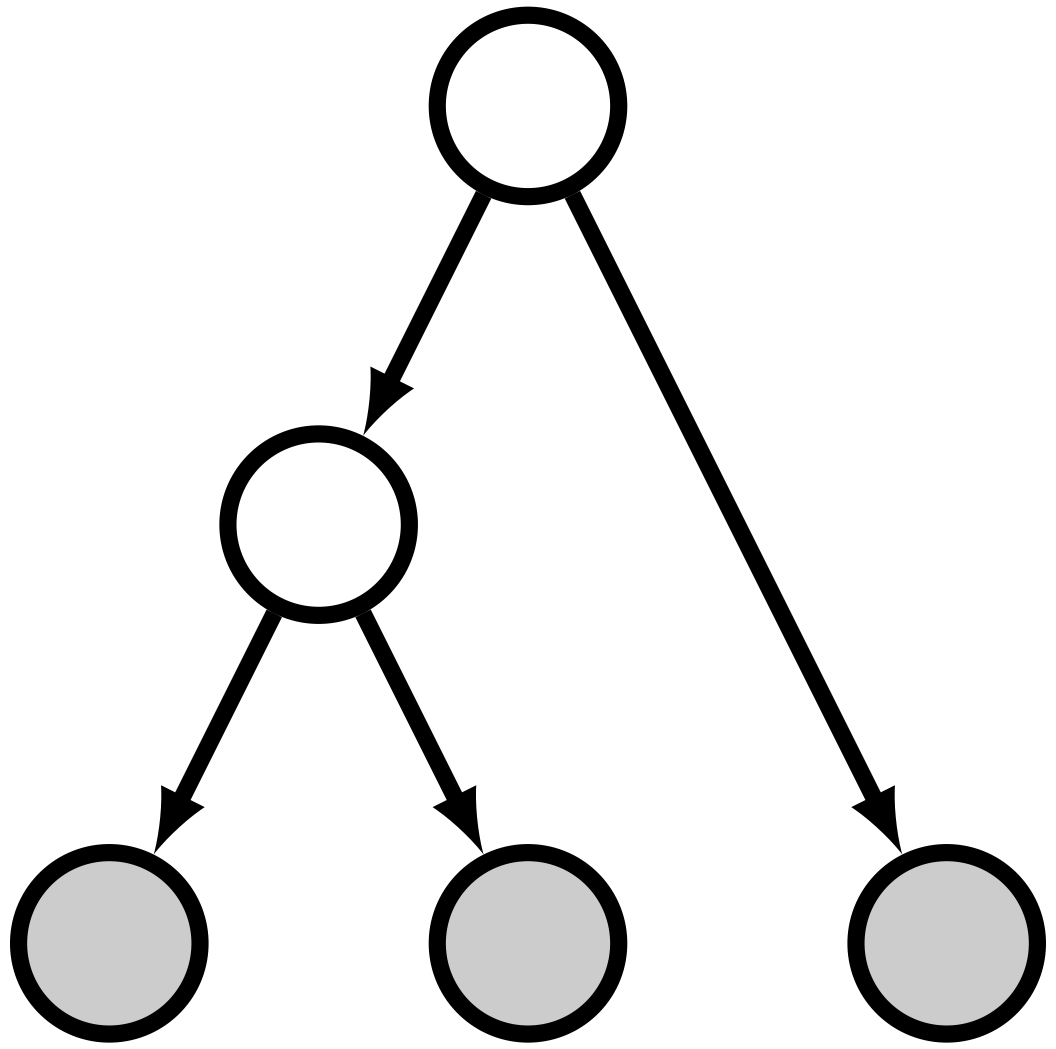

```
# constant node
r <- 10

# 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

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

Likelihood

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

Bayes' theorem

This represents our prior knowledge of the model parameters

Priors

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

Bayes' theorem

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

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

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

Bayes' theorem

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

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

How do we find the ‘best’ tree?

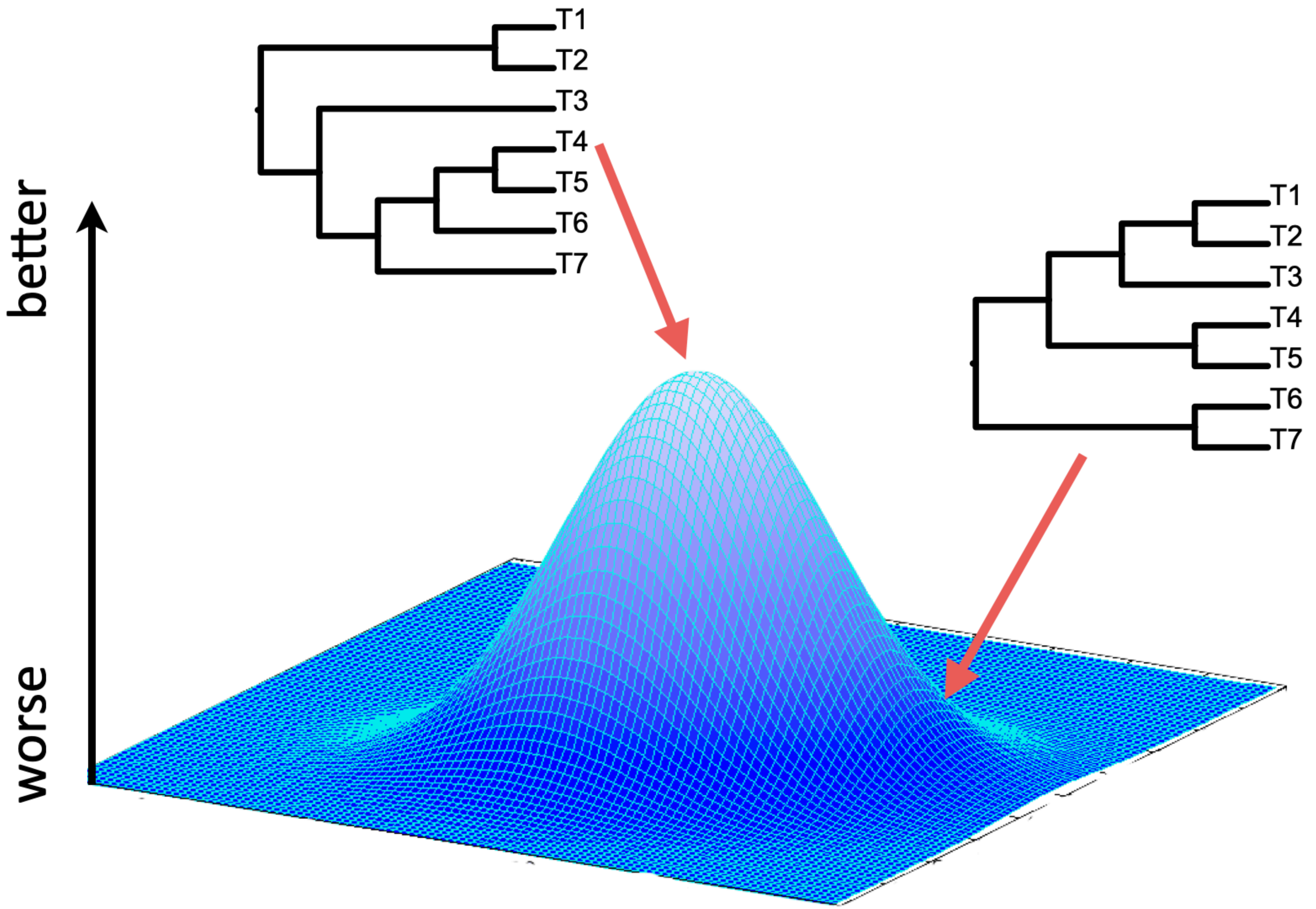


Image source Tracy Heath

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

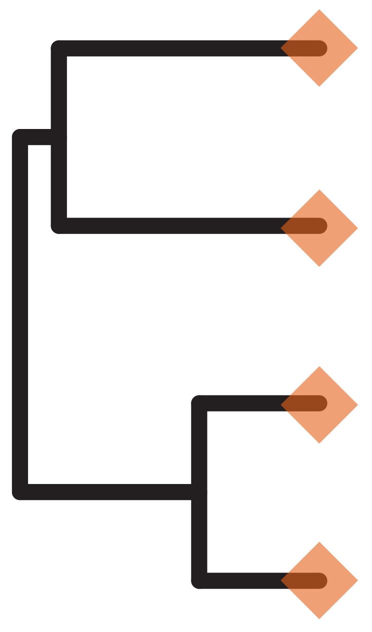
Note these are not the only approaches to tree-building but they are the most widely used

Components used to infer trees

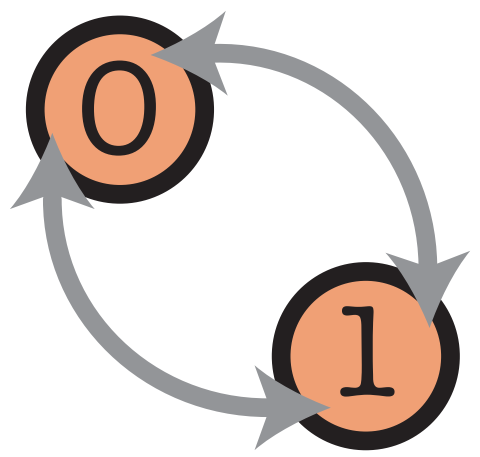
without considering time

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

data
sequences or
characters

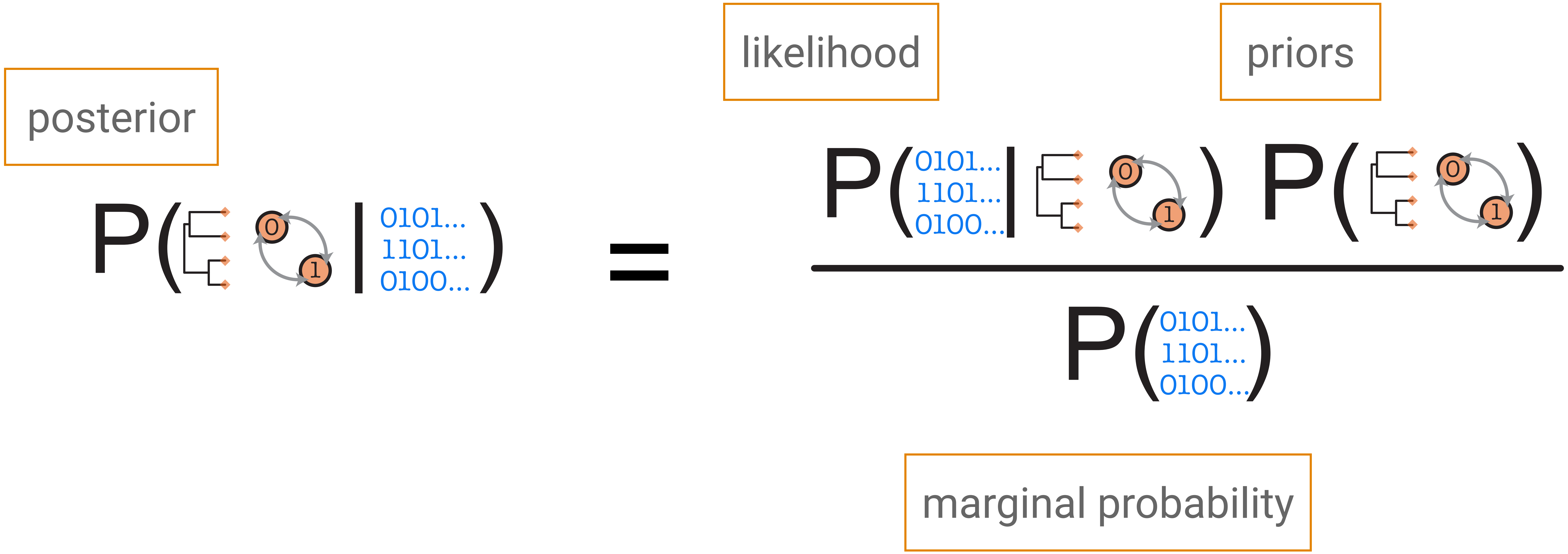


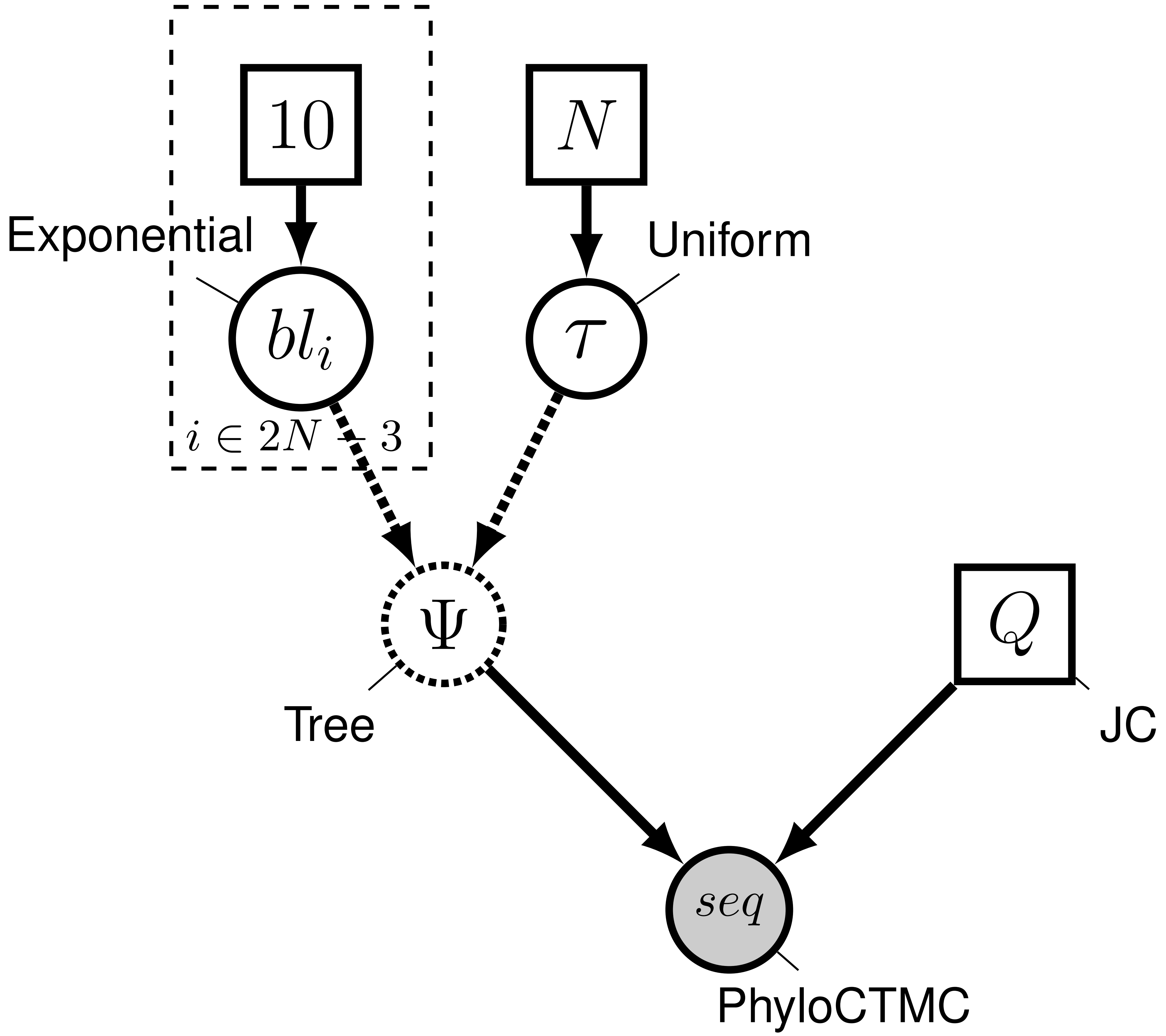
tree
topology and
branch lengths

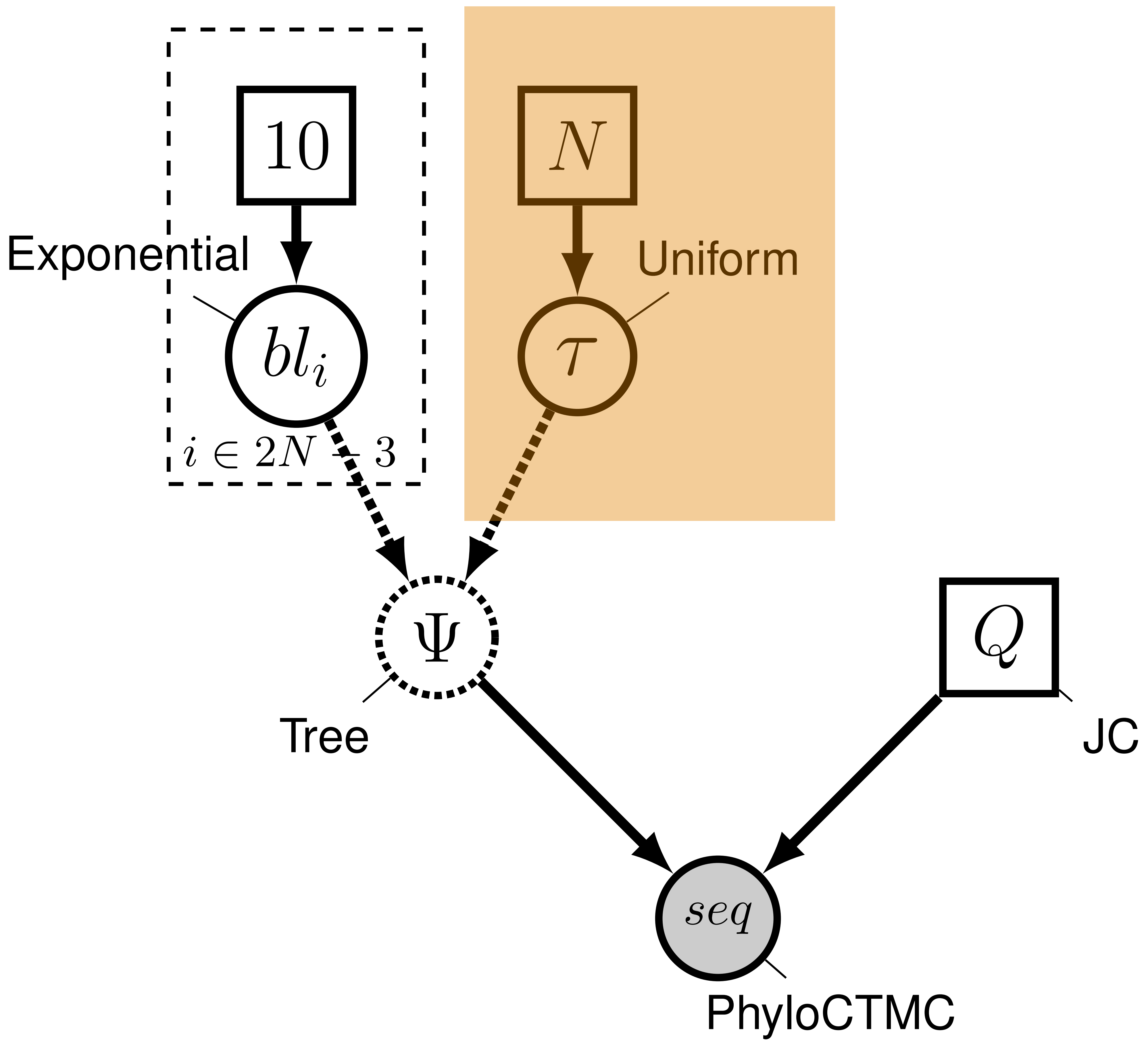


substitution
model

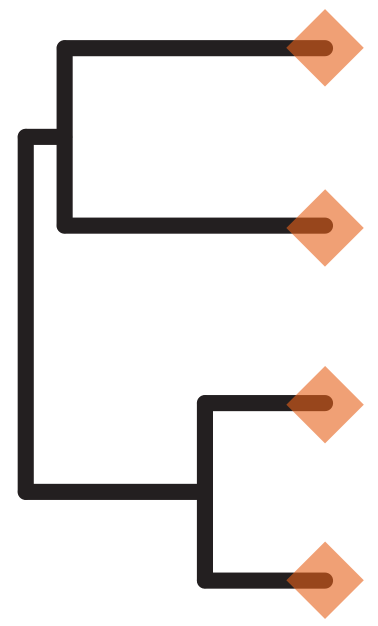
Bayesian tree inference

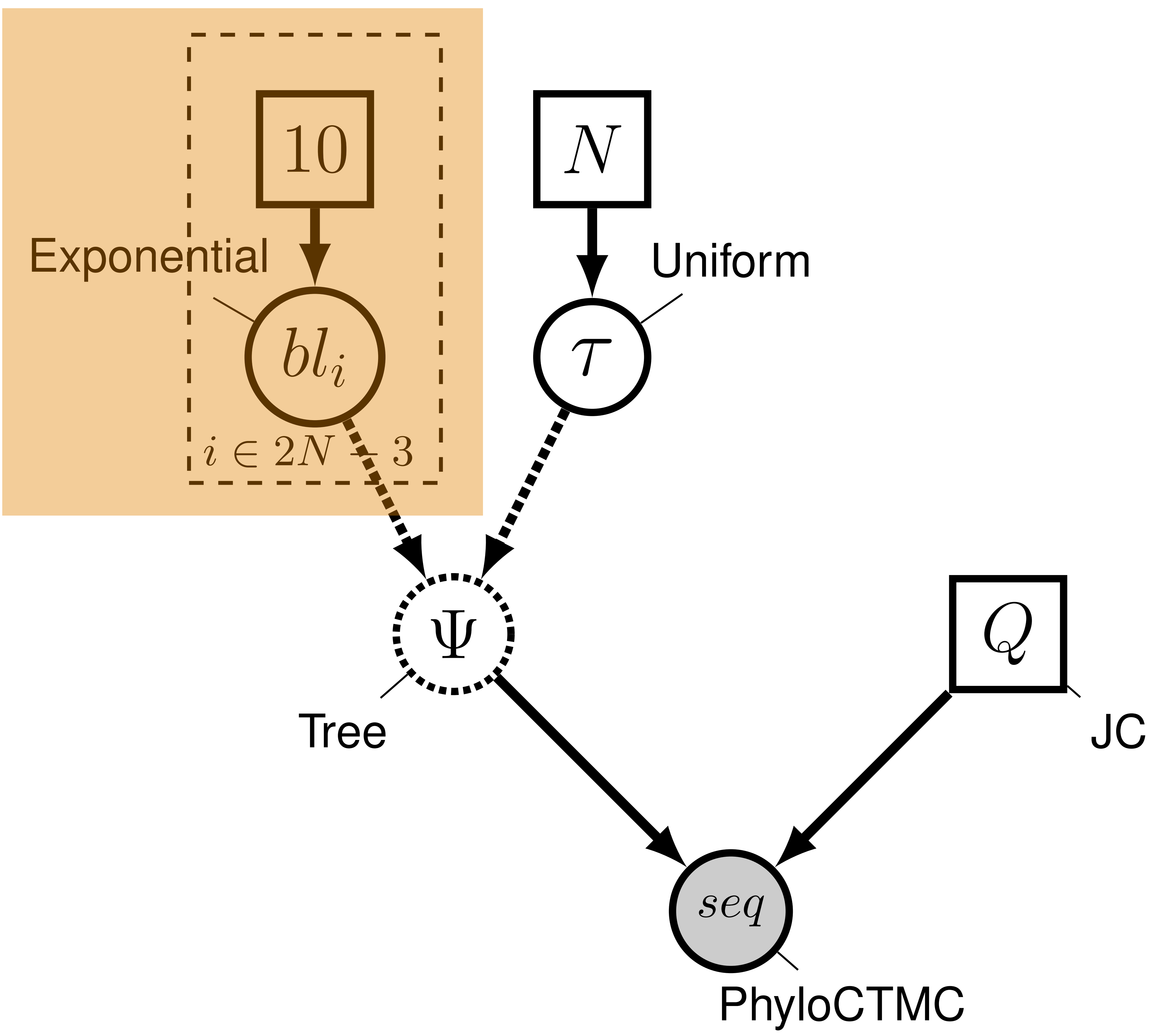



10
N
Exponential
Uniform
bl_i
τ
$i \in 2N-3$
Ψ
Q
Tree
JC
seq
PhyloCTMC

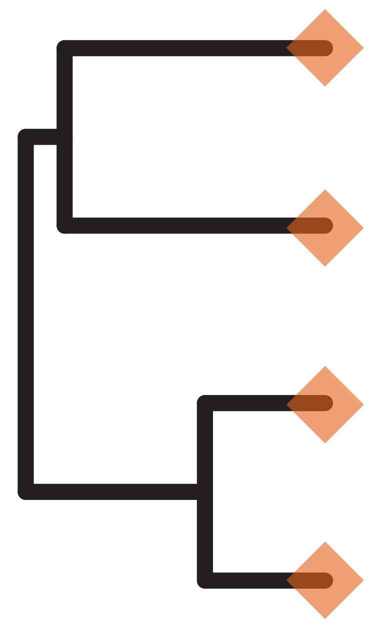


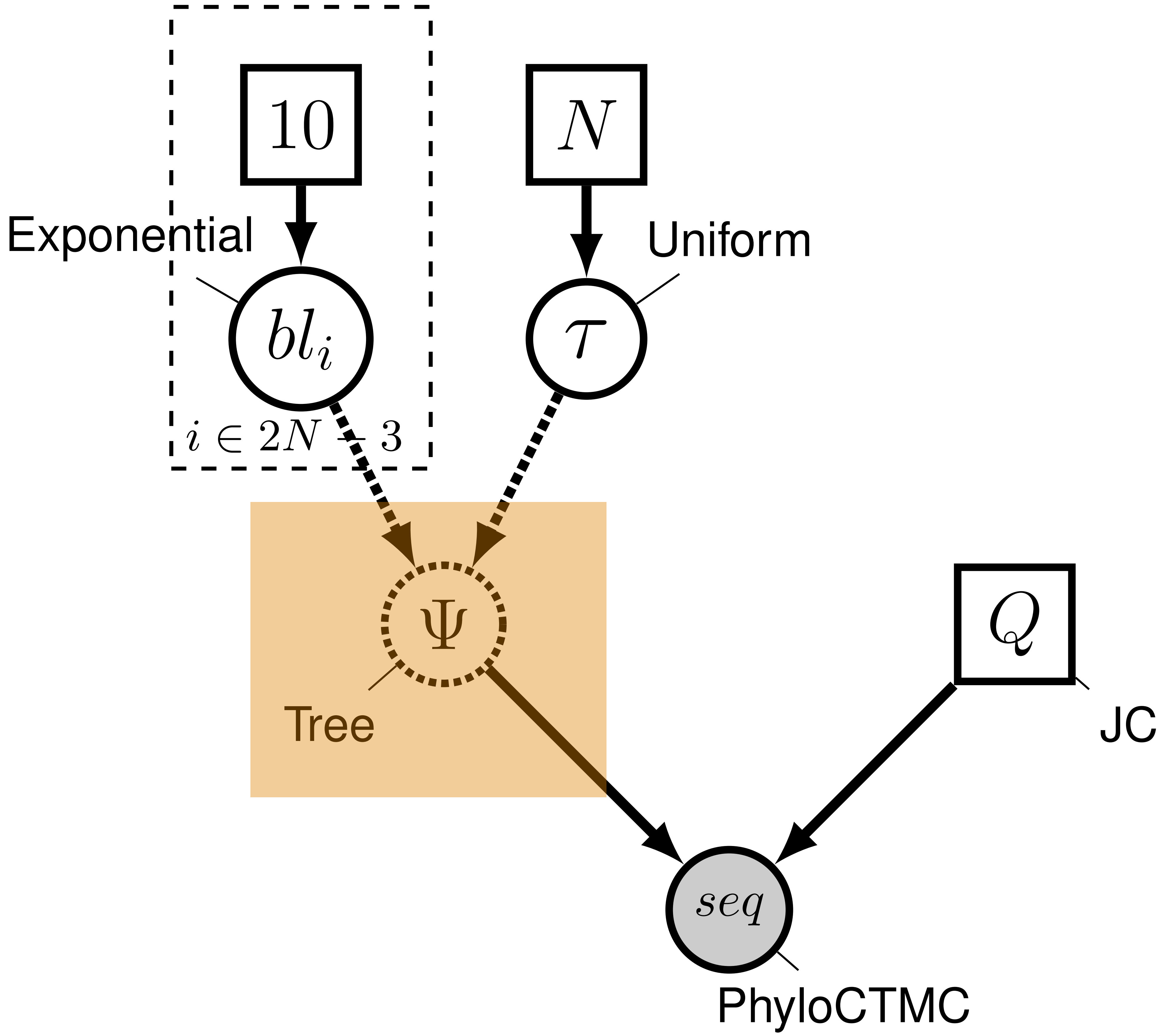
prior on the tree topology



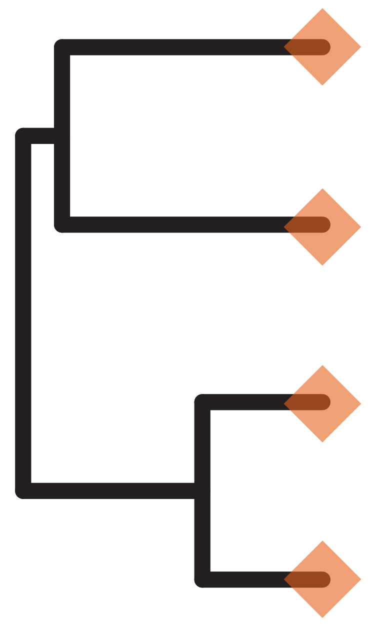


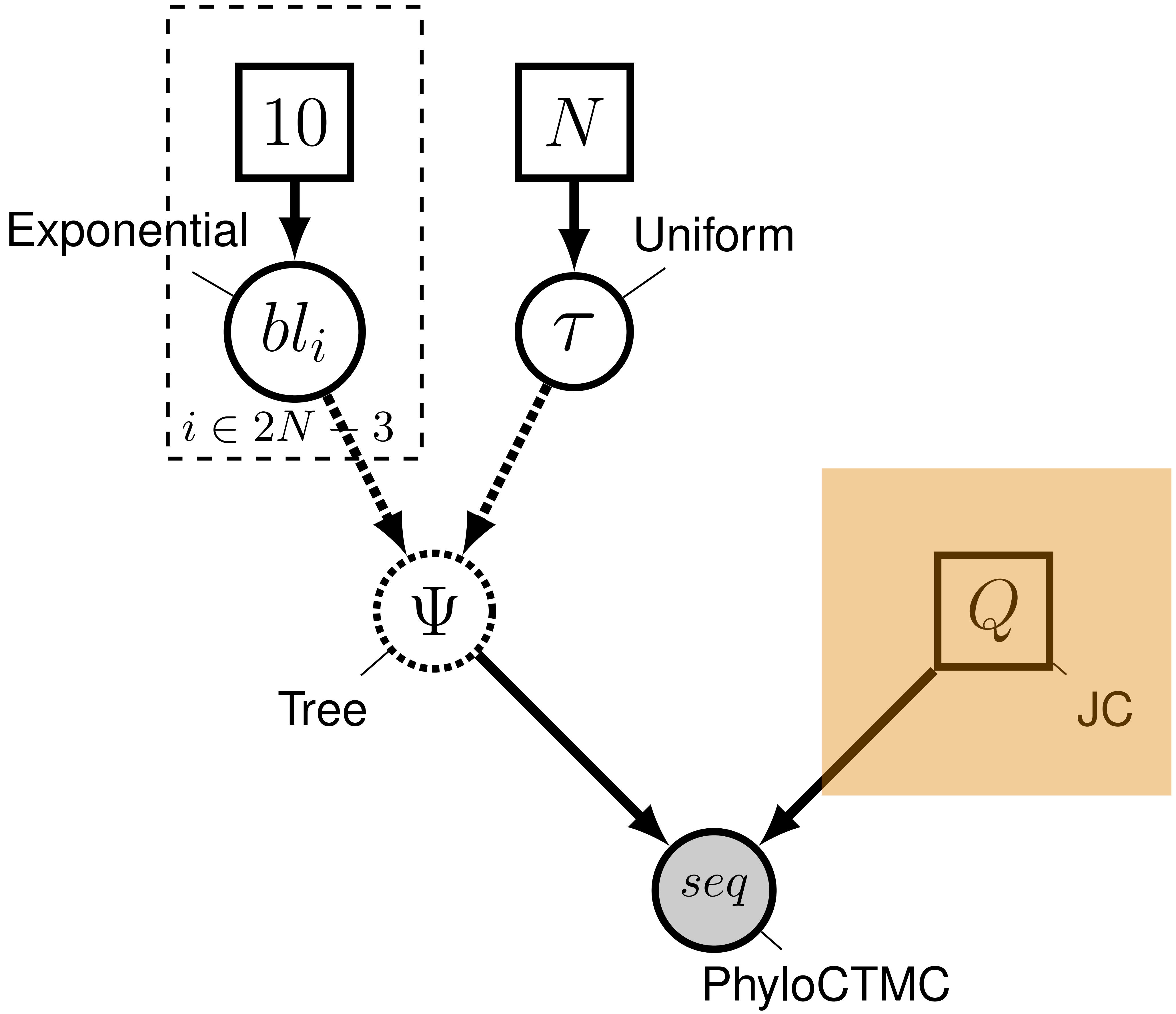
prior on the branch lengths



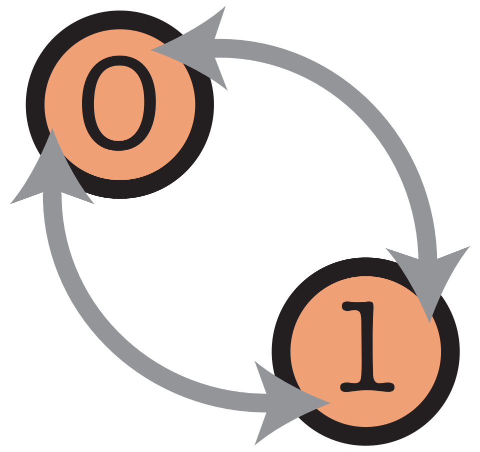


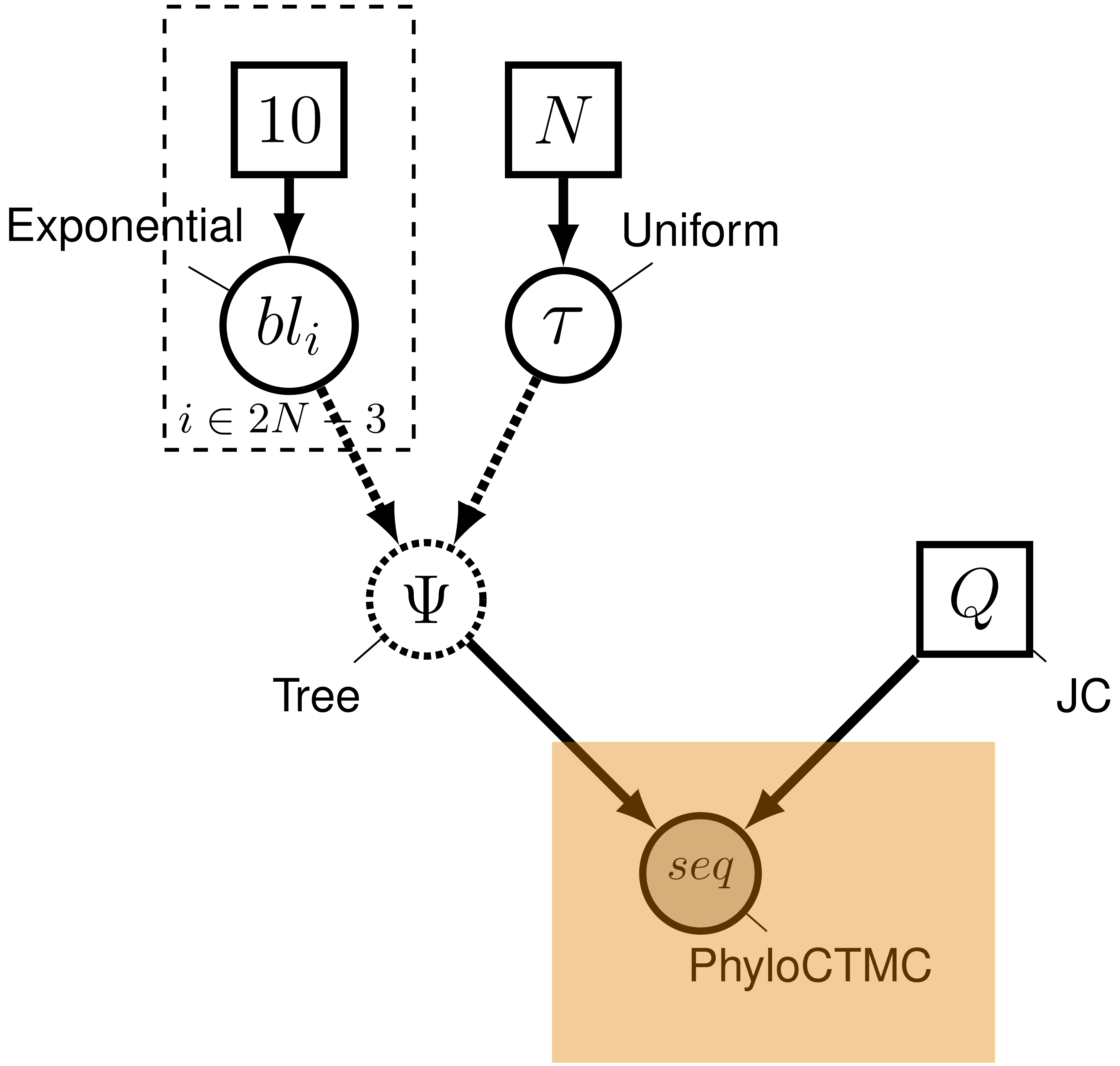
we can combine the topology and branch lengths





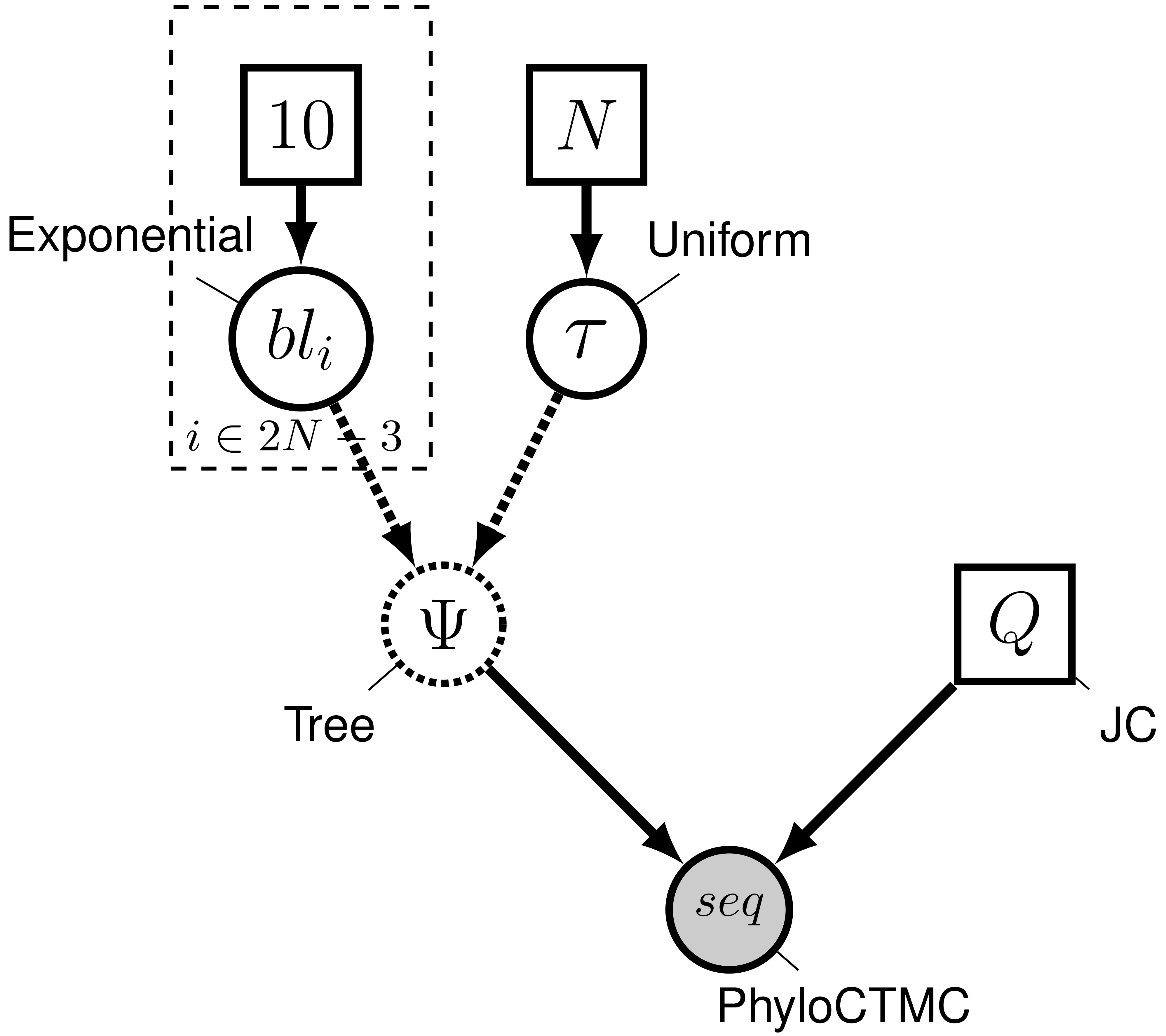
Substitution model



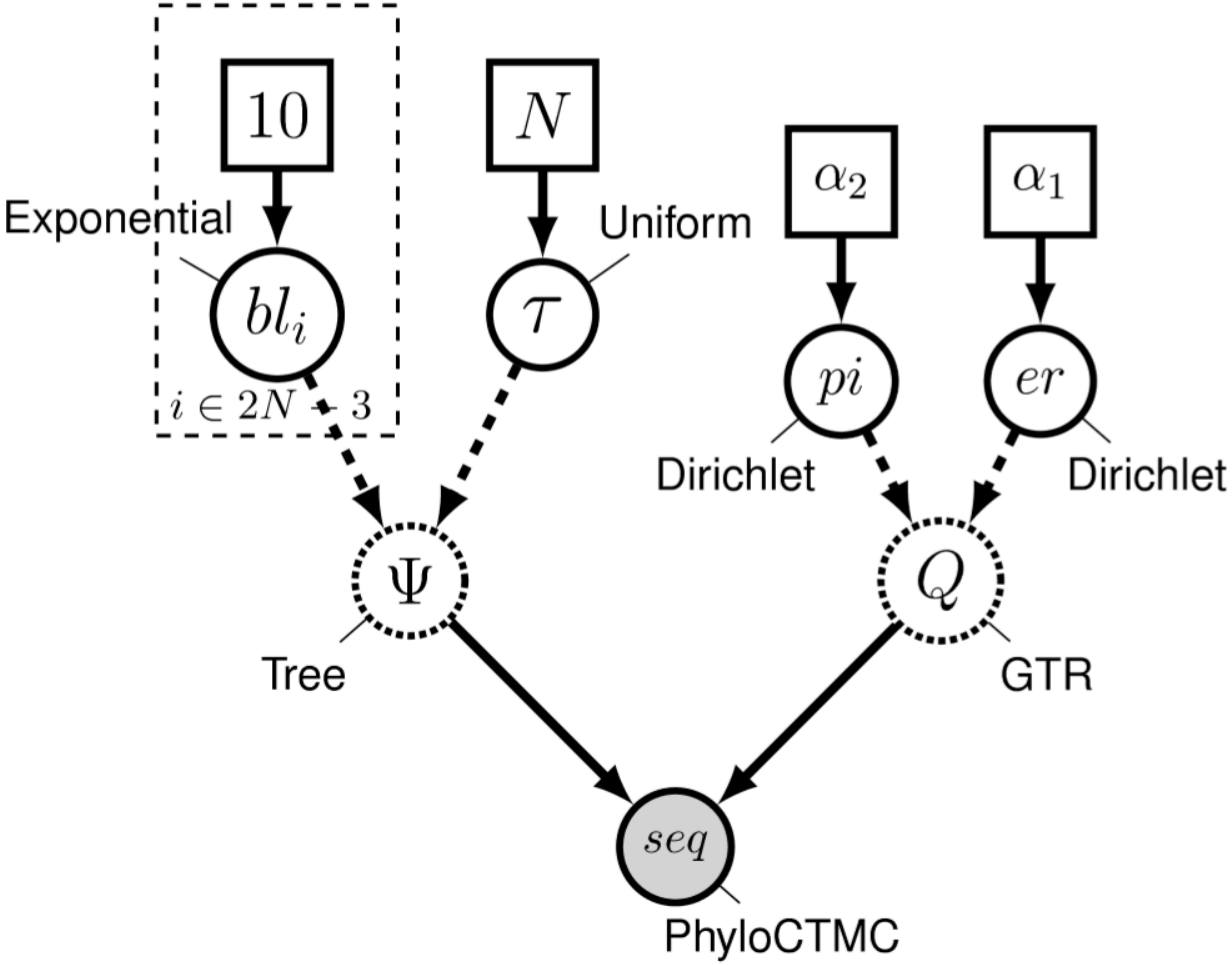


Observed data

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 )
```


10
N
α_2
α_1
Exponential
Uniform
bl_i
τ
pi
er
$i \in 2N - 3$
Dirichlet
Dirichlet
Ψ
Q
Tree
GTR
seq
PhyloCTMC