

# Phylogenetics

Introduction to Bayesian  
inference

RL-V3 MPP

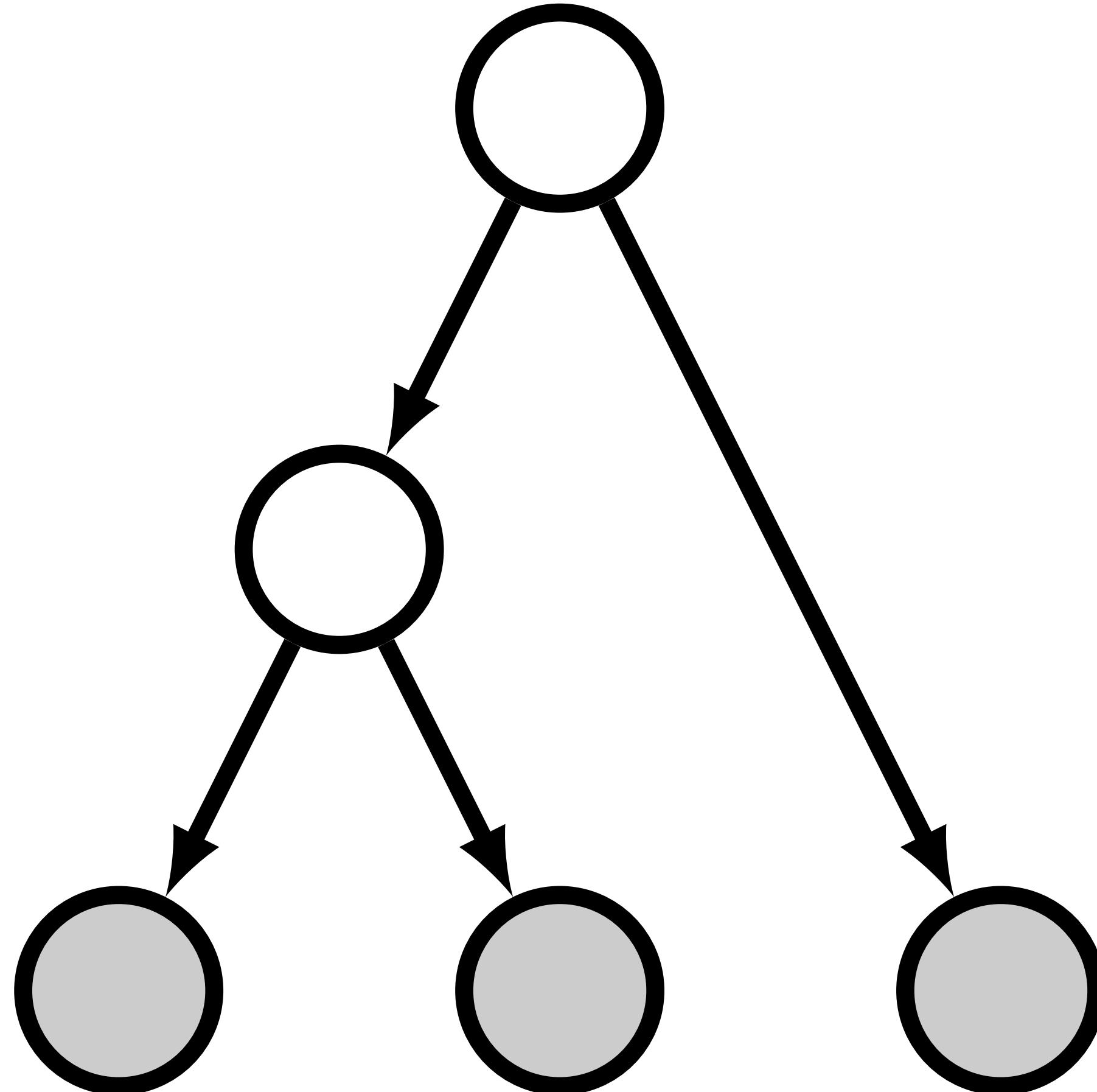
Rachel Warnock

15.04.25



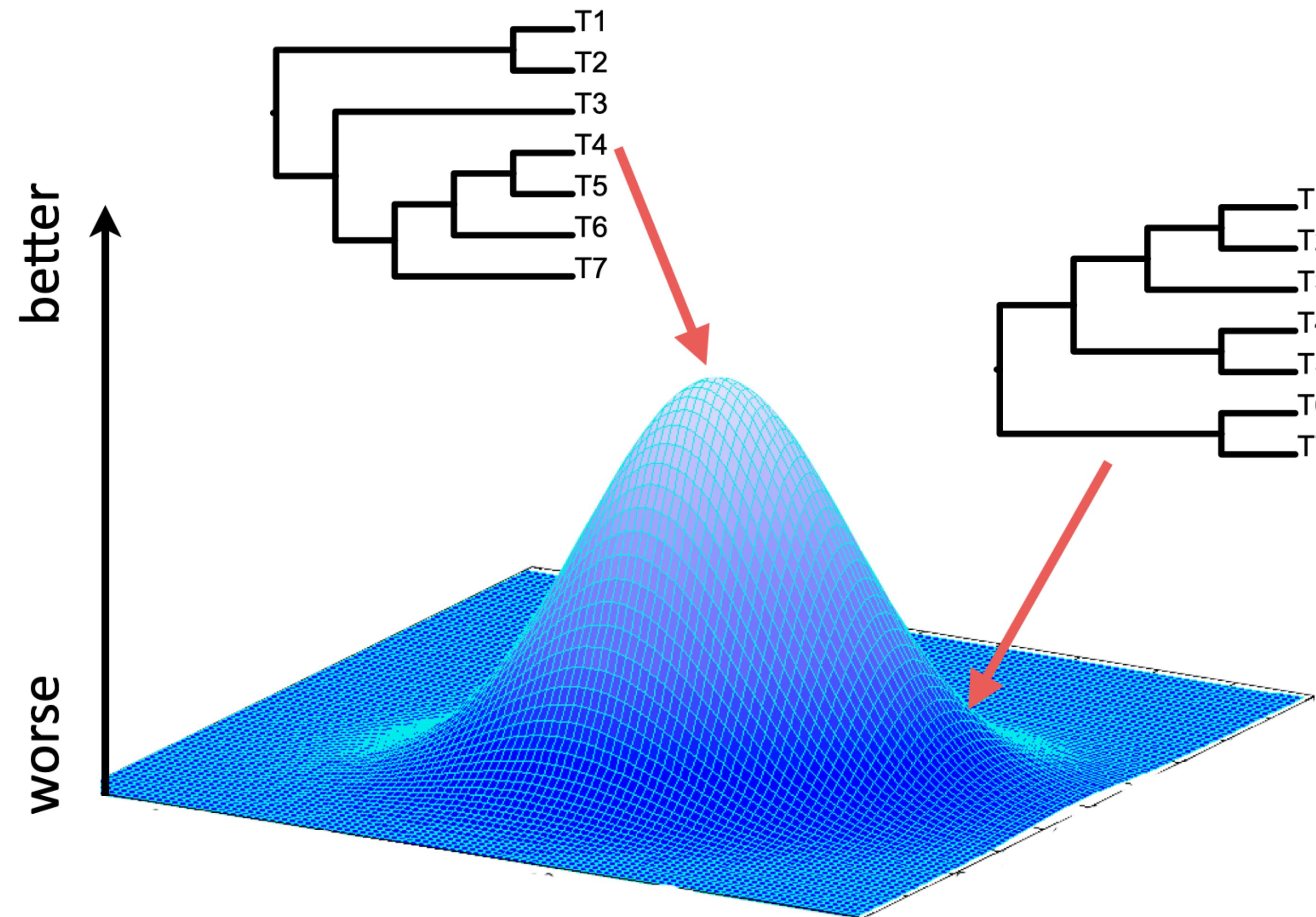
# Objectives

- Graphical models
- RevBayes
- Bayesian inference
- MCMC



# Recap

# How do we find the ‘best’ tree?



# It depends how you measure ‘best’

---

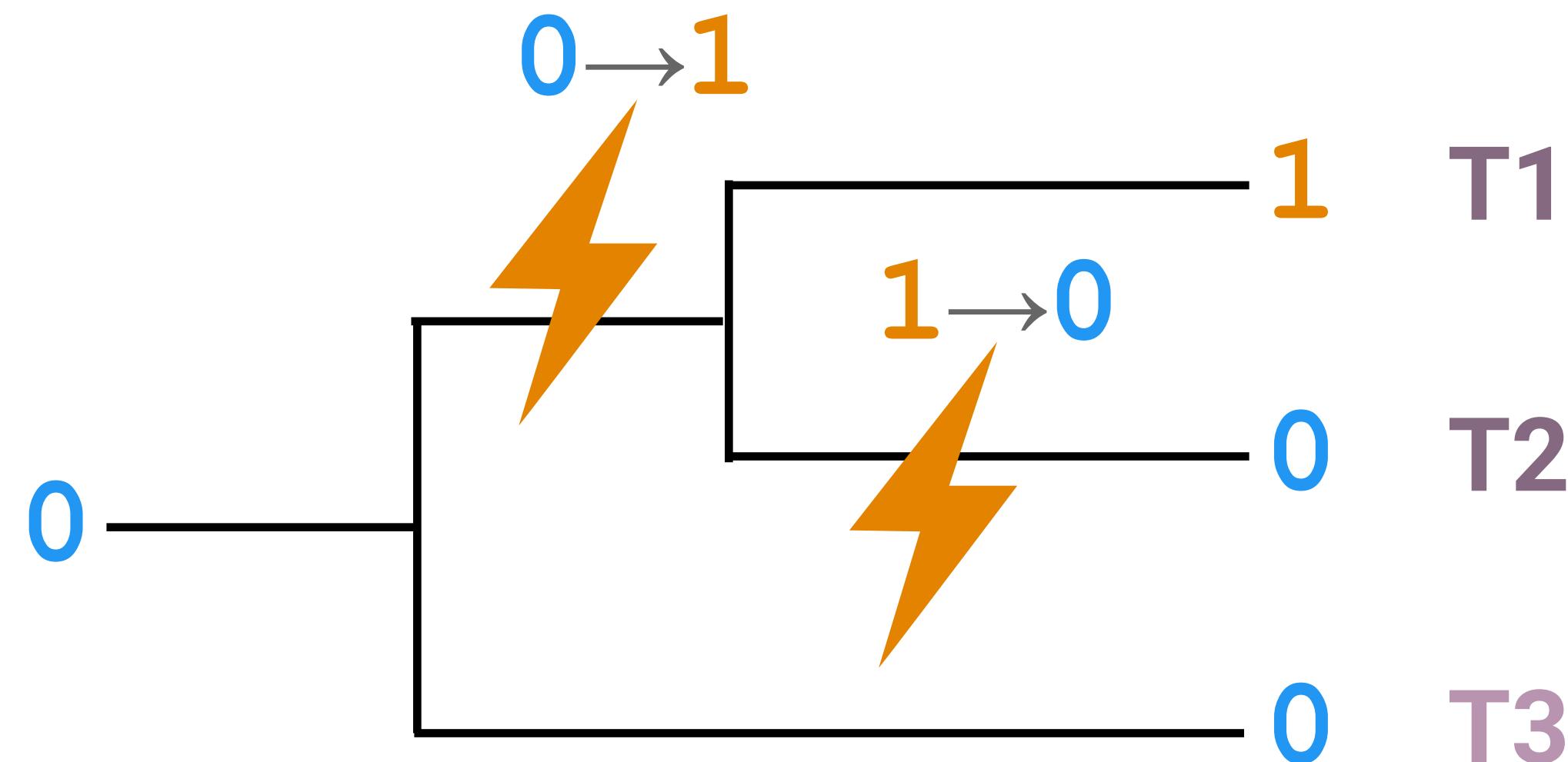
Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

---

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 ranger of applications in paleobio)

# It depends how you measure ‘best’

---

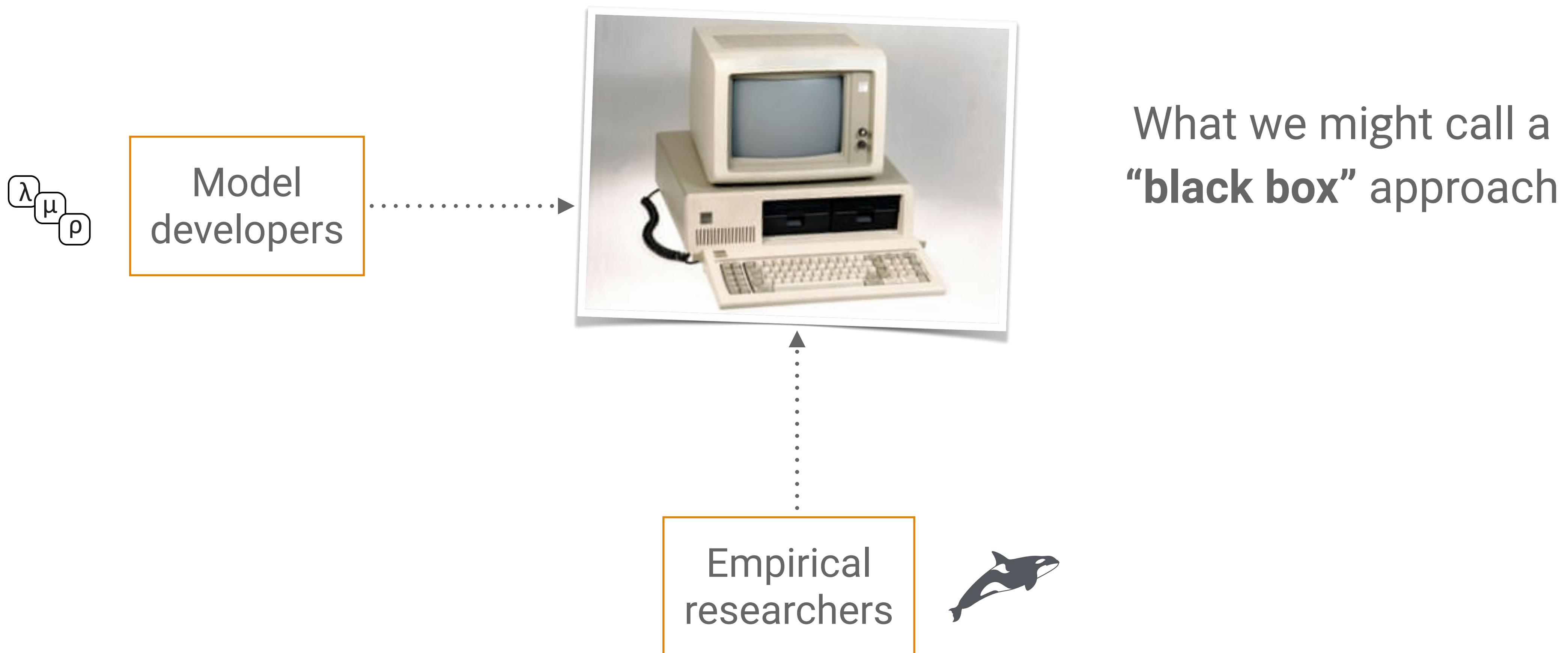
Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

Both maximum likelihood and Bayesian inference are model-based approaches

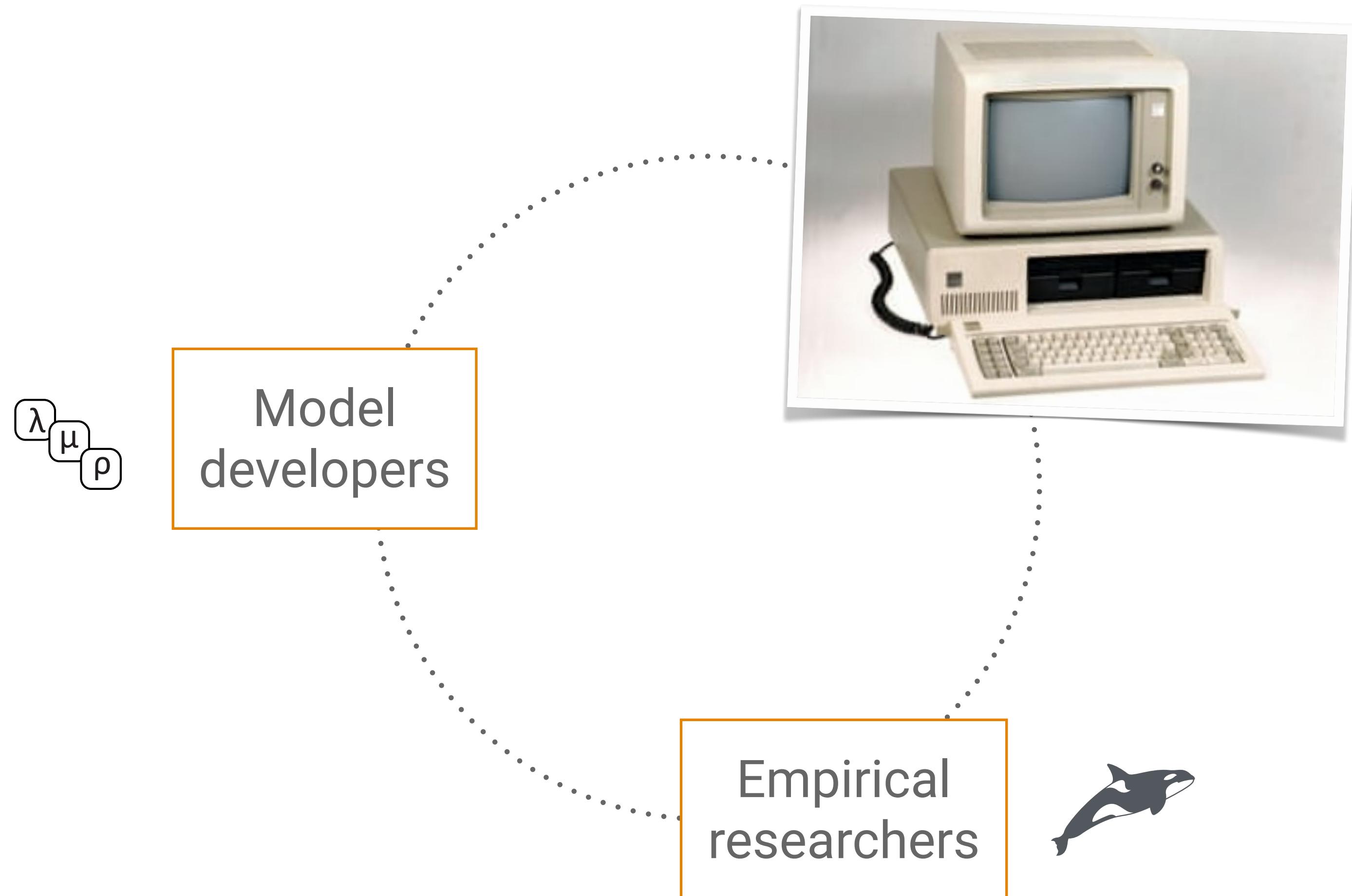
Note these are not the only approaches to tree-building but they are the most widely used

# RevBayes

# Phylogenetic inference – the old way



# 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](https://revbayes.github.io)

The screenshot shows a web browser window displaying the RevBayes GitHub page ([revbayes.github.io](https://revbayes.github.io)). The page features a navigation bar with links for Download, Tutorials, Documentation, Interfaces, Workshops, Jobs, and Developer. A small portrait of a man in historical attire is positioned next to the Download link. The main content area contains a phylogenetic tree diagram on the left and descriptive text on the right.

**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](#)

The screenshot shows the Beast2.org website. At the top left is the Beast logo (a black mouse). To its right is the text "Beast2" in blue and "Bayesian evolutionary analysis by sampling trees". Below this is a navigation bar with links: ABOUT (which is highlighted), BOOK, CITATION, TUTORIALS, FAQ, BLOG, and DEVELOPERS. A search icon is located at the top right. On the left side, there's a sidebar titled "RECENT POSTS" with several entries. The main content area has a section titled "ABOUT" with a detailed description of what BEAST 2 is. Below this is a "Download" section with links for Windows, Mac OS X, and Linux.

The screenshot shows the taming-the-beast.org website. At the top left is a small icon of a mouse in a cage. To its right is the text "Taming the BEAST". Below this is a brief description: "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!](#)". The page is organized into sections: "news", "recently updated tutorials", "workshops", and "resources". Each section contains several items with titles, dates, and authors.

[www.beast2.org](http://www.beast2.org)

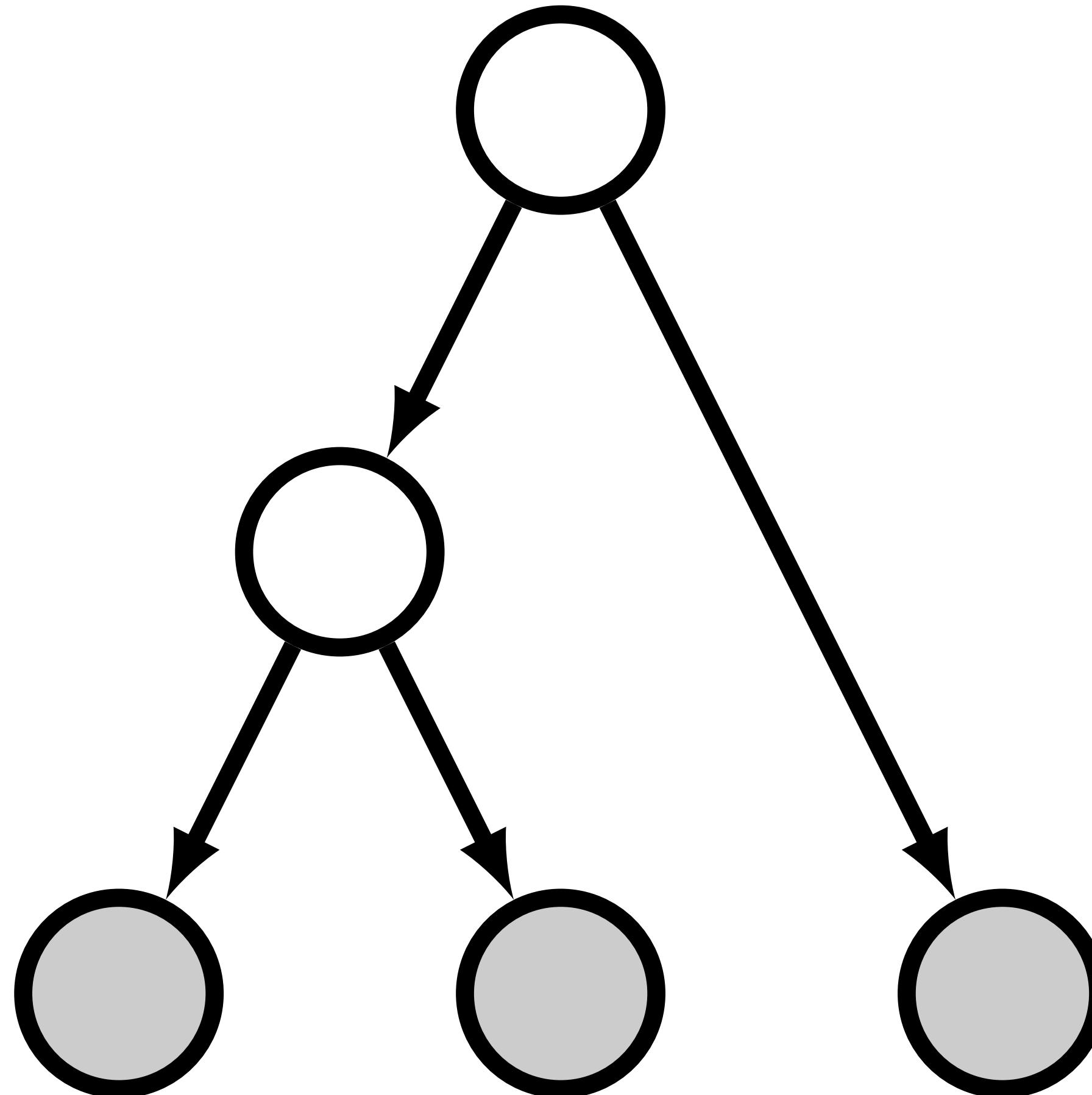
[taming-the-beast.org](http://taming-the-beast.org)

# Graphical models

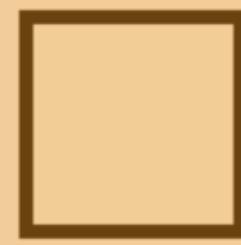
# Graphical models

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

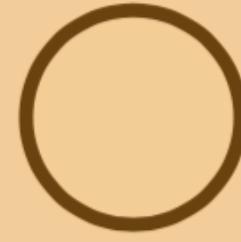
Depict the conditional dependence structure of parameters and other random variables



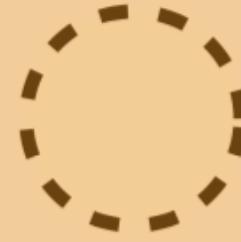
# Types of variables (nodes)



**a) Constant node**



**b) Stochastic node**



**c) Deterministic node**



**d) Clamped node  
(observed)**

**a. fixed value variables**

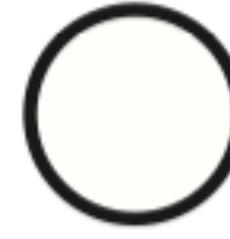
**b. random variables that depend on other variables**

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

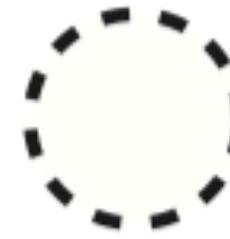
**d. observed stochastic variables (data)**



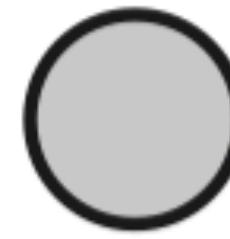
**a) Constant node**



**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)**

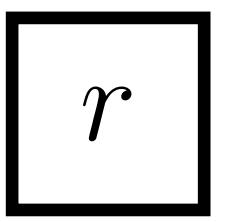
**e. repetition over multiple variables (equivalent to a loop)**

# Specifying graphical models using the Rev syntax

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

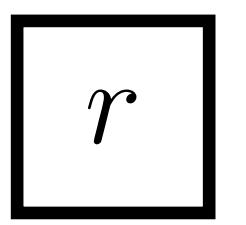
Operator	Variable
<code>&lt;-</code>	constant variable
<code>~</code>	stochastic variable
<code>:=</code>	deterministic variable
<code>node.clamp(data)</code>	clamped variable
<code>=</code>	inference ( <i>i.e.</i> , non-model) variable
<code>for(i in 1:N){...}</code>	plate

a)

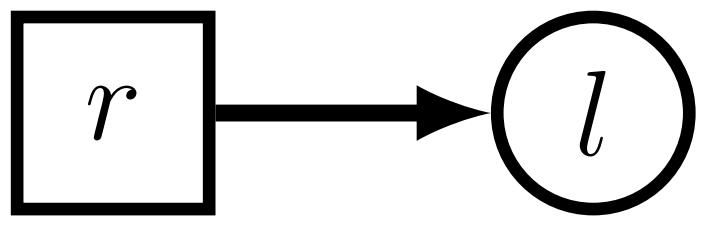


```
# constant node  
r <- 10
```

a)



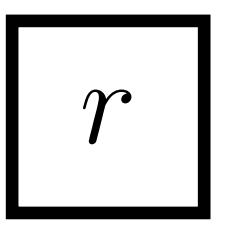
b)



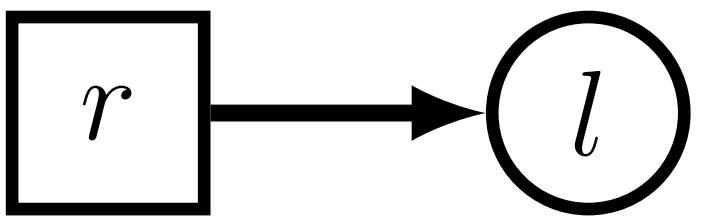
```
# constant node  
r <- 10
```

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

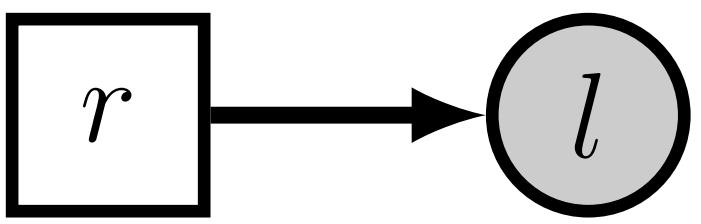
a)



b)



c)

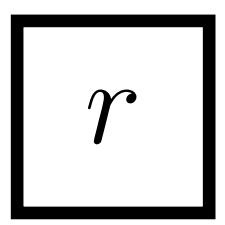


```
# constant node  
r <- 10
```

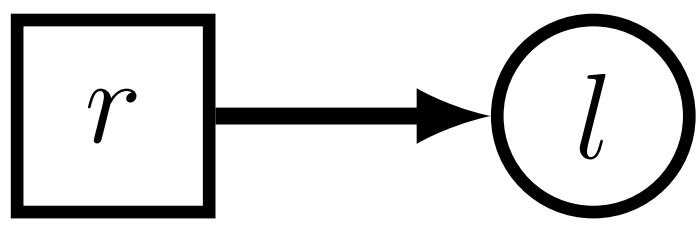
```
# stochastic node  
l ~ dnExp(r)
```

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

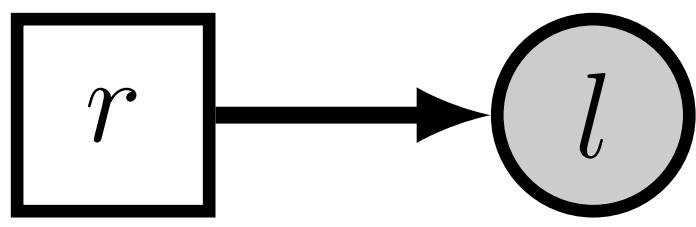
a)



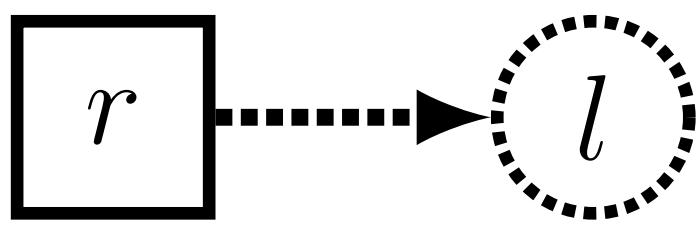
b)



c)



d)



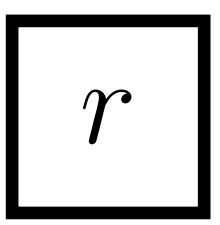
```
# constant node  
r <- 10
```

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

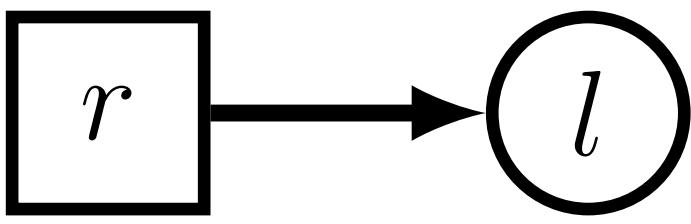
```
# stochastic node (observed)  
l.clamp(0.1)
```

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

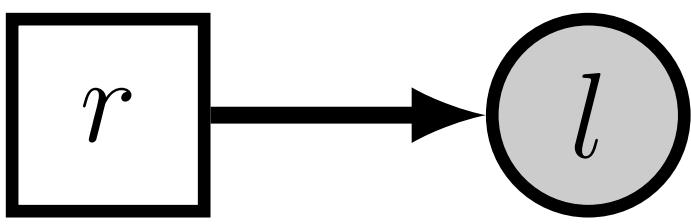
a)



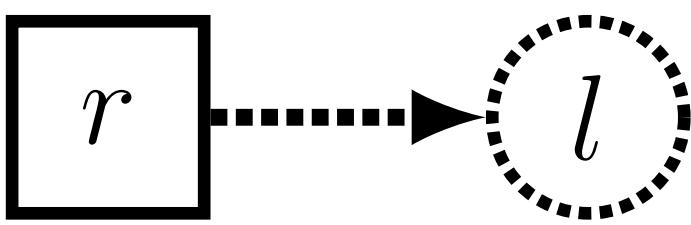
b)



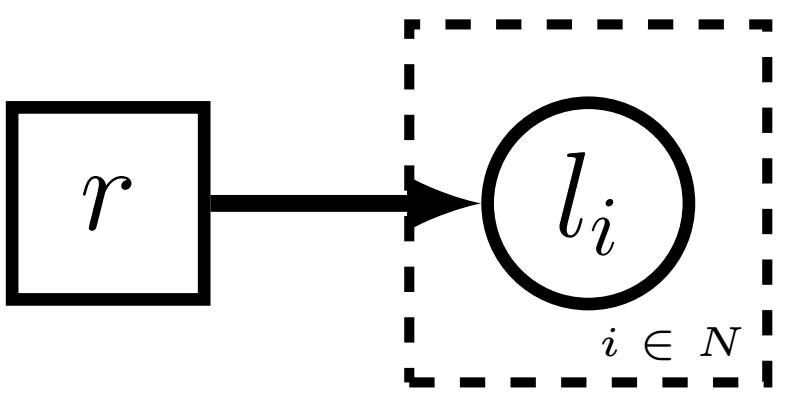
c)



d)



e)



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

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

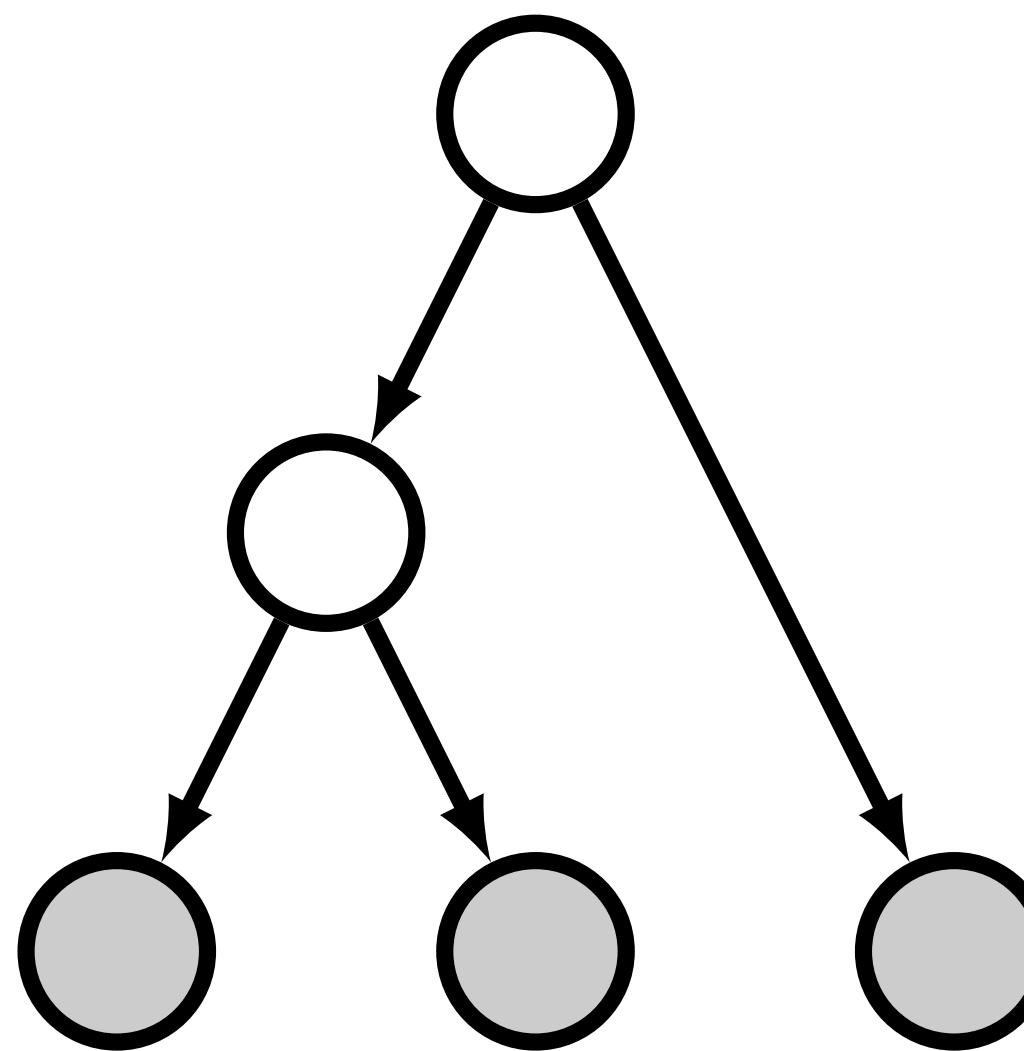
```
# stochastic node (observed)  
l.clamp(0.1)
```

```
# 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

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

Marginal probability

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

# Bayes' theorem

posterior

Reflects our combined knowledge based on the likelihood and the priors

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

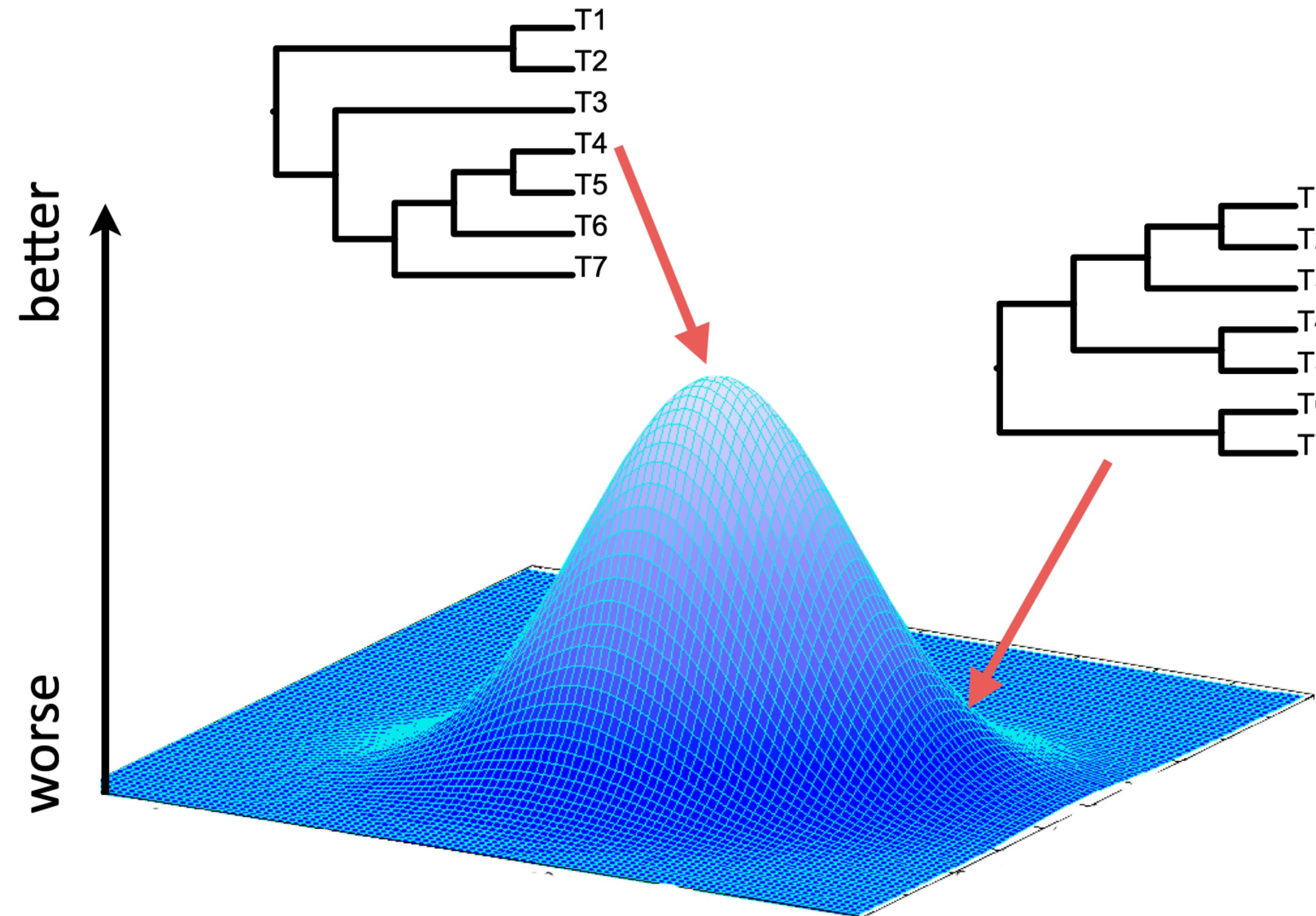
$$\frac{\Pr(\text{data} \mid \text{model}) \Pr(\text{model})}{\Pr(\text{data})}$$

# Bayes' theorem

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

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

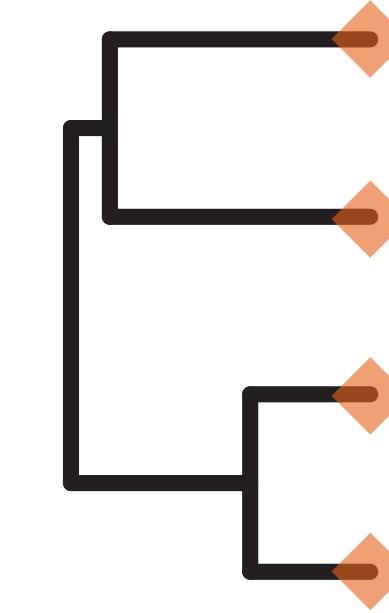
Note these are not the only approaches to tree-building but they are the most widely used

# Components used to infer trees

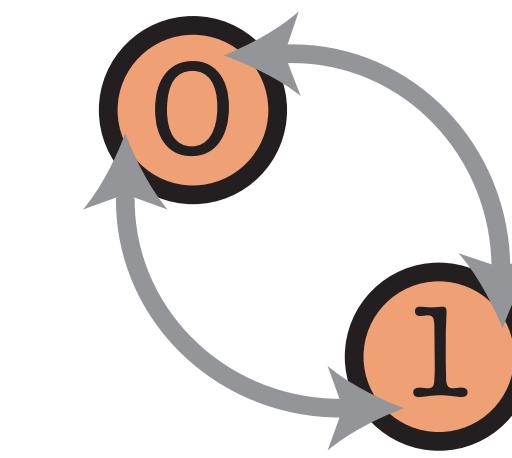
without considering time

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

data  
sequences or  
characters



tree  
topology and  
branch lengths



substitution  
model

# Bayesian tree inference

$$\text{posterior} \quad P(E \mid \text{0101...}, \text{1101...}, \text{0100...}) = \frac{\text{likelihood} \quad P(\text{0101...} \mid E) \quad P(E)}{\text{priors} \quad P(\text{0101...}, \text{1101...}, \text{0100...})}$$

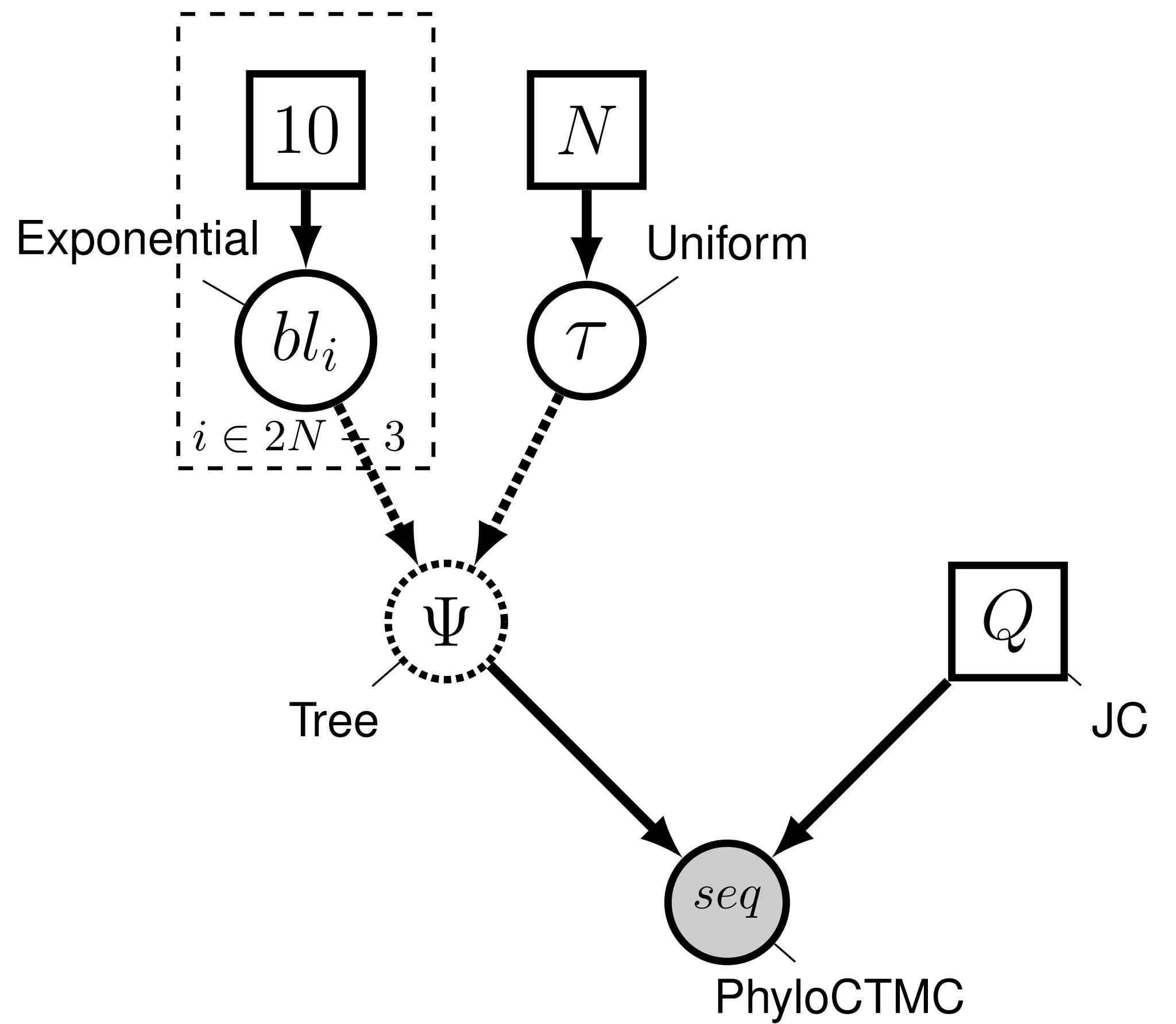
Diagram illustrating the components of Bayesian tree inference:

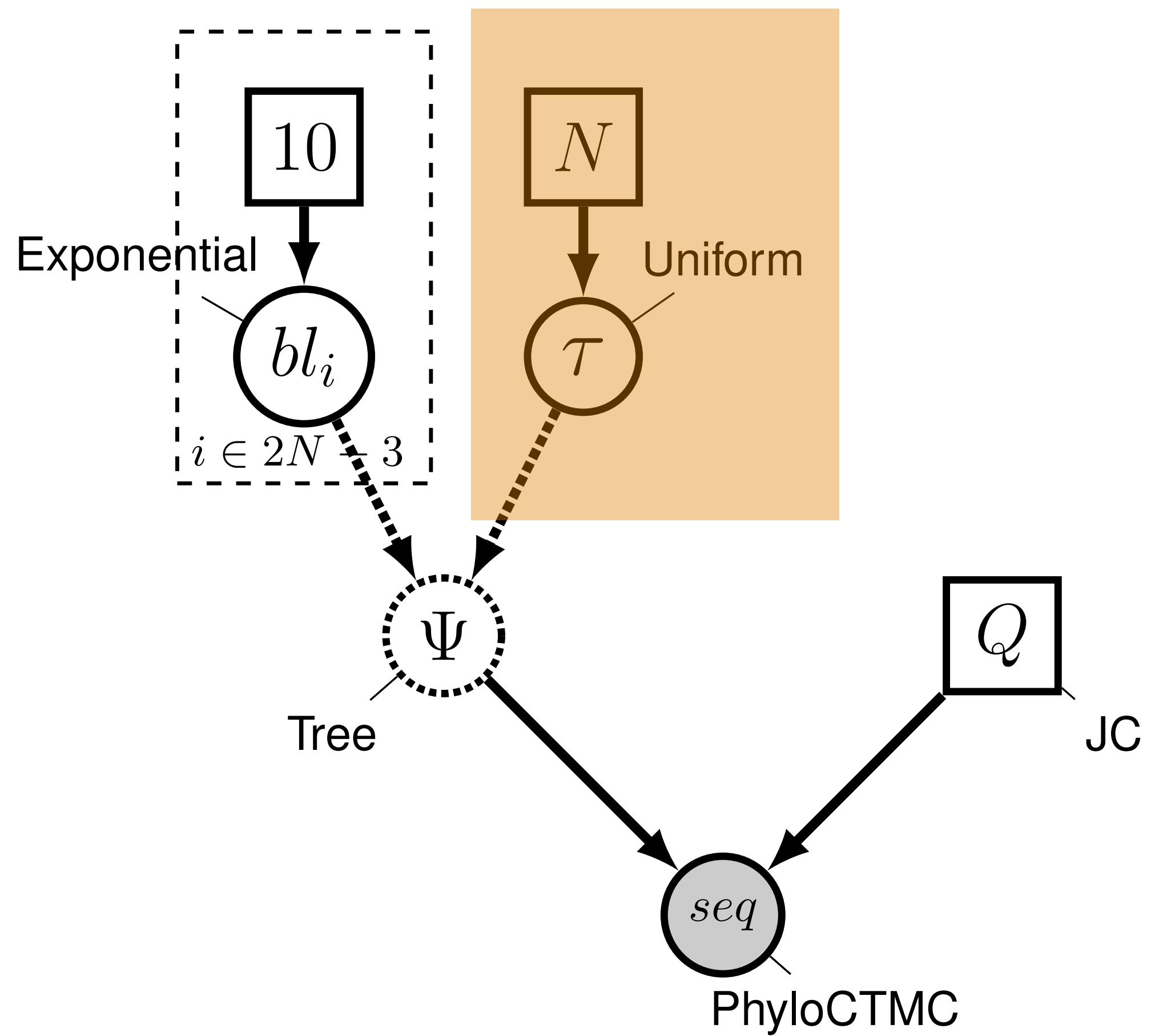
- posterior**:  $P(E \mid \text{0101...}, \text{1101...}, \text{0100...})$
- likelihood**:  $P(\text{0101...} \mid E)$
- priors**:  $P(E)$
- marginal probability**:  $P(\text{0101...}, \text{1101...}, \text{0100...})$

The diagram shows a phylogenetic tree with two terminal nodes. The left node is labeled '0' and the right node is labeled '1'. Arrows indicate the direction of evolution from root to leaves. The tree structure is as follows:

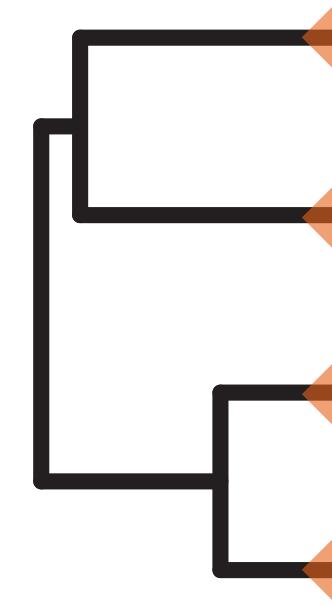
- Root branches into two nodes.
- Left branch leads to node '0'.
- Right branch leads to node '1'.
- Node '0' has three children, which further branch into four leaves.
- Node '1' has two children, which further branch into three leaves.

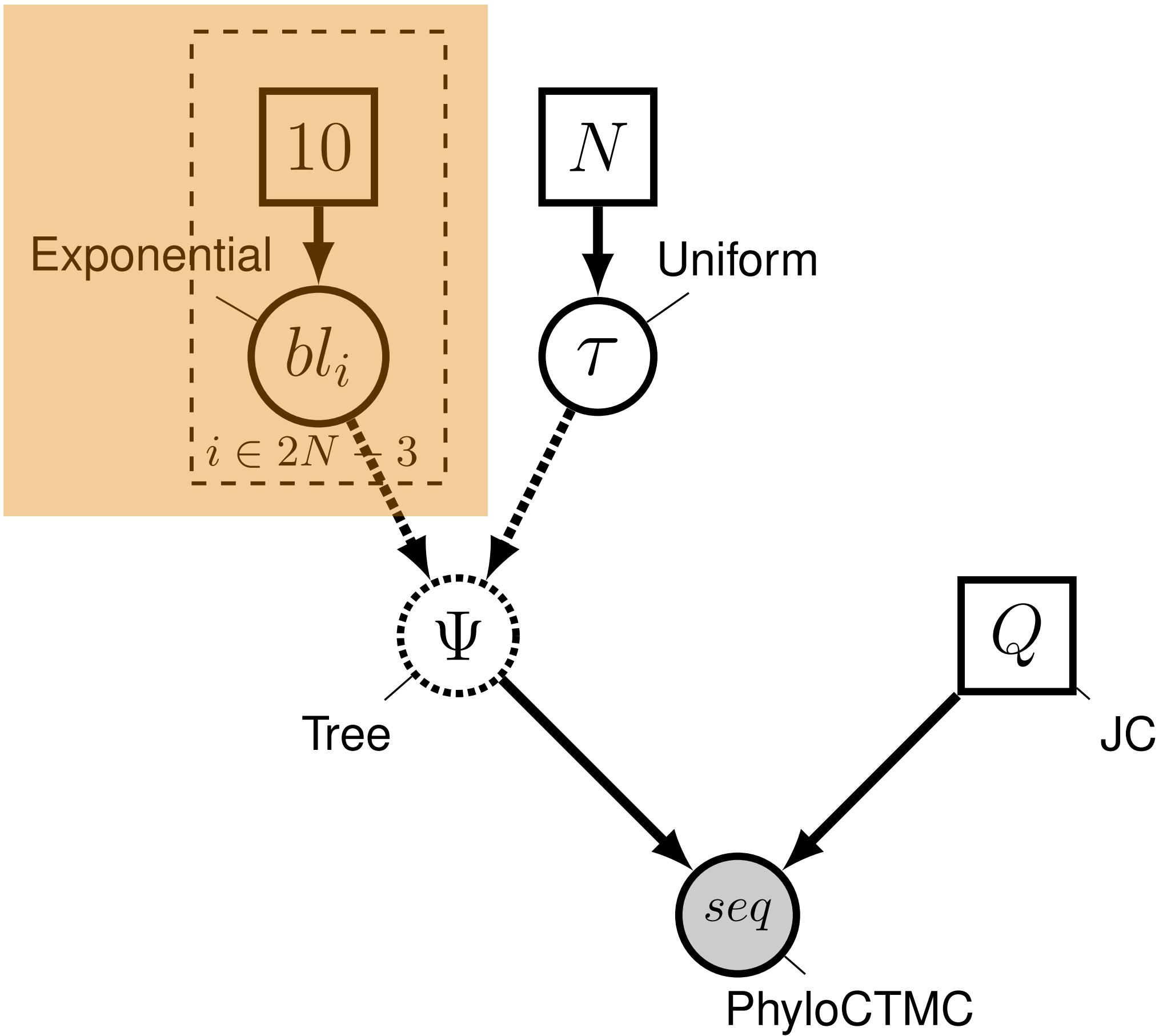
Sequence labels are provided for the leaves: 0101..., 1101..., and 0100... for the first group of leaves, and 0101..., 1101..., and 0100... for the second group of leaves.



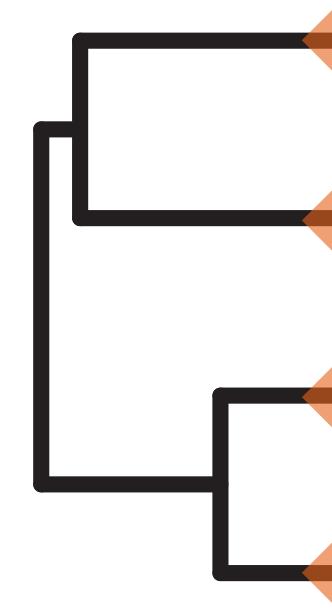


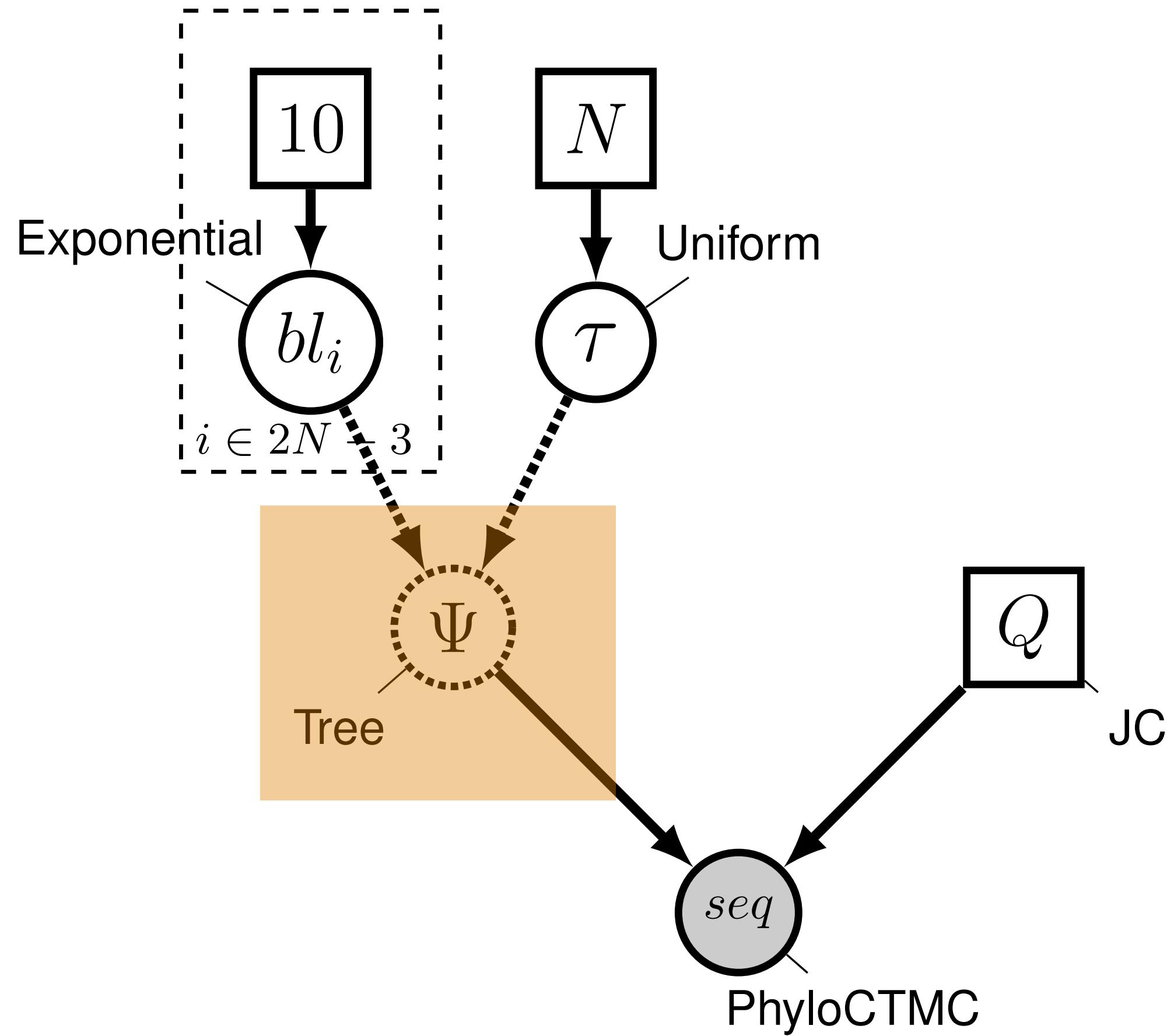
prior on the tree  
topology



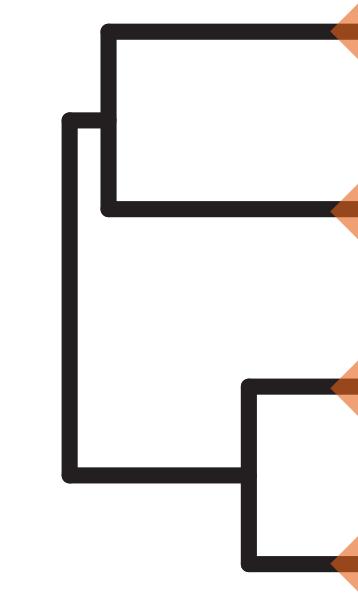


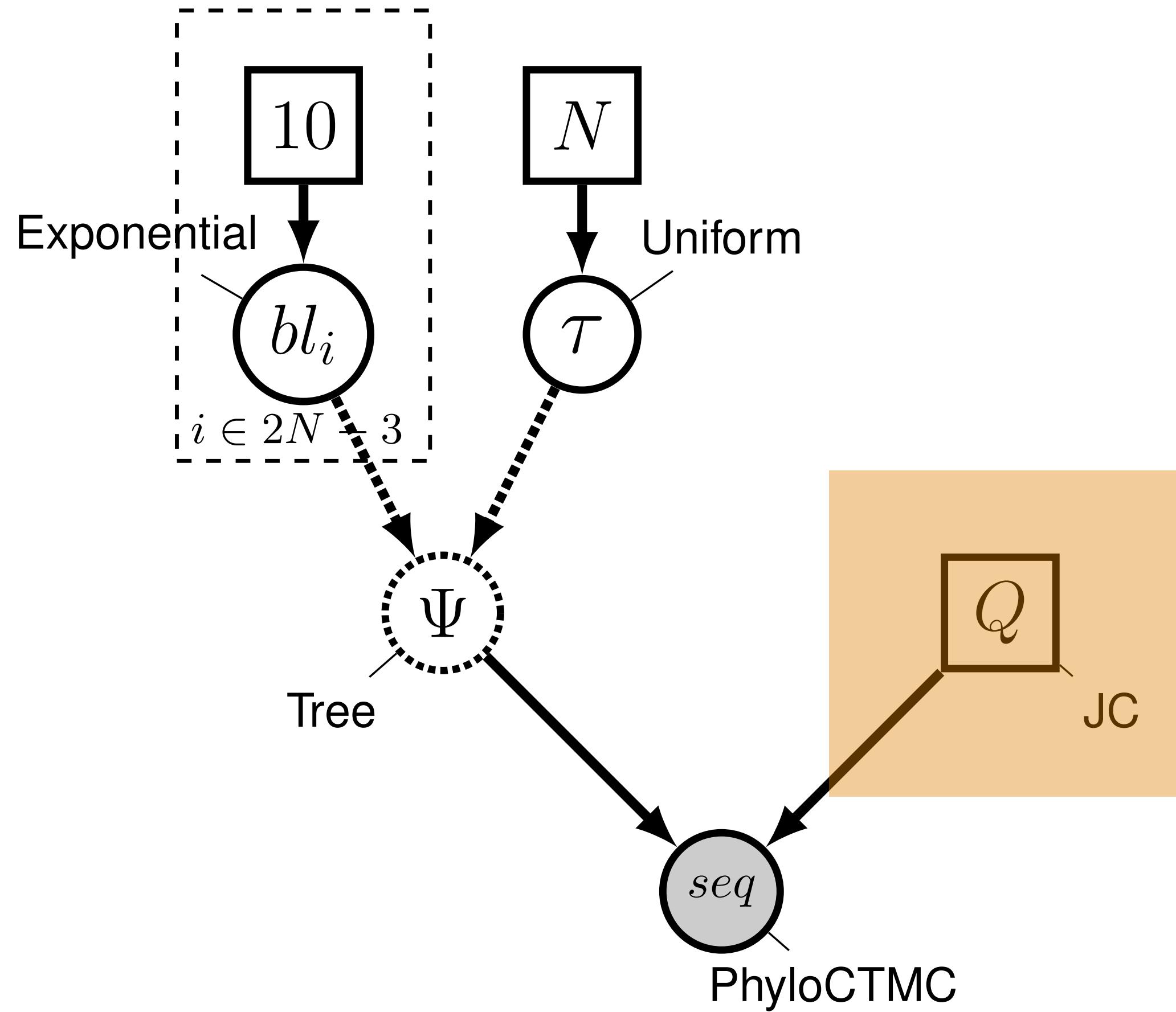
prior on the  
branch lengths



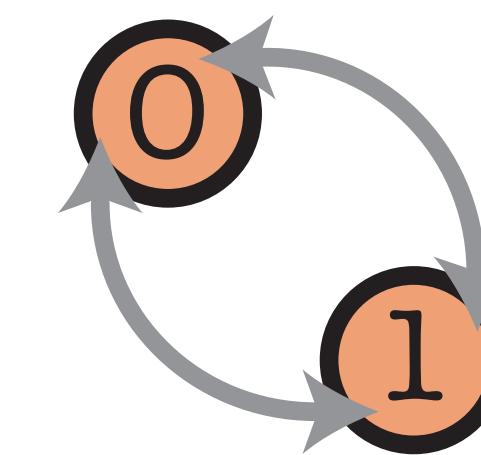


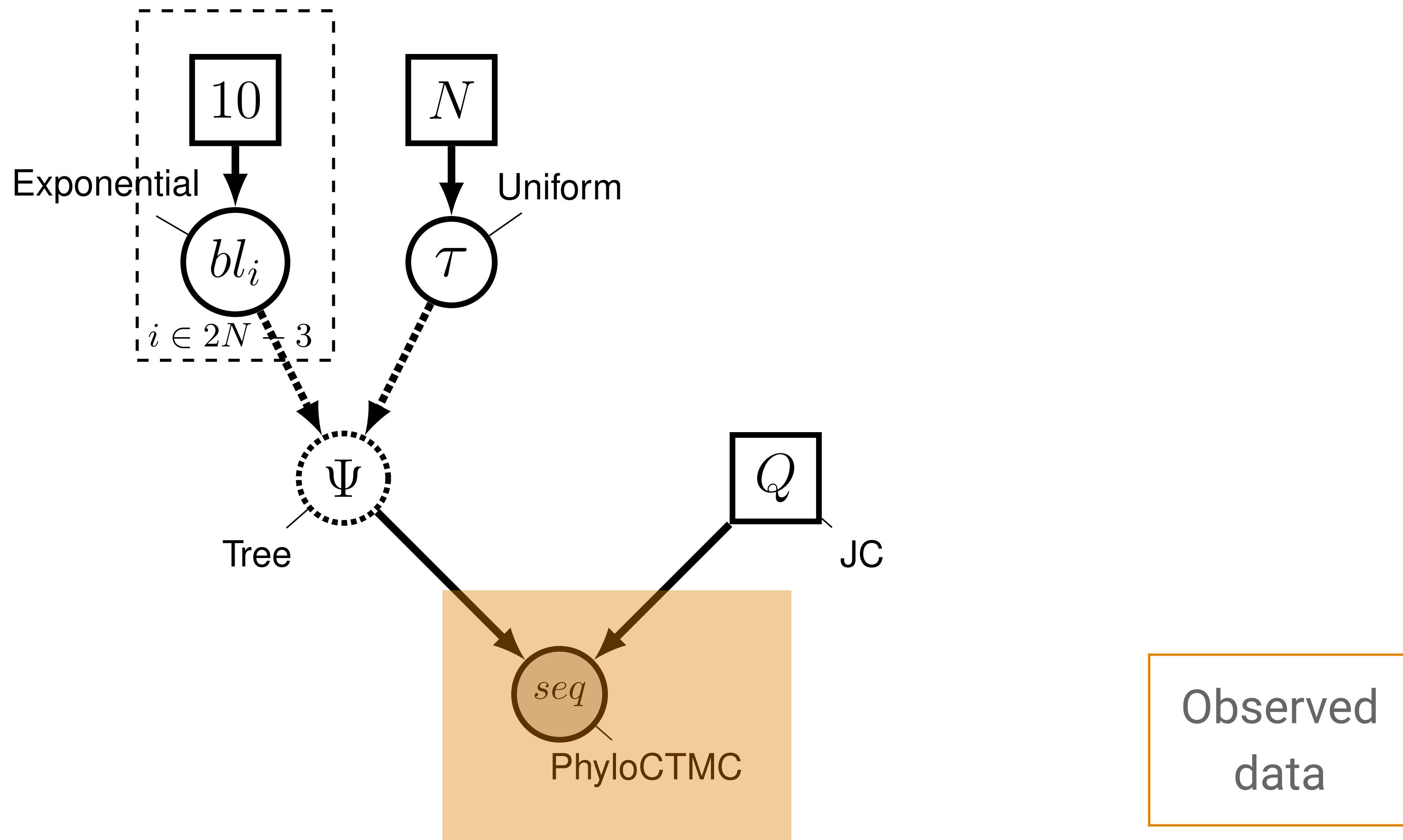
we can combine  
the topology and  
branch lengths



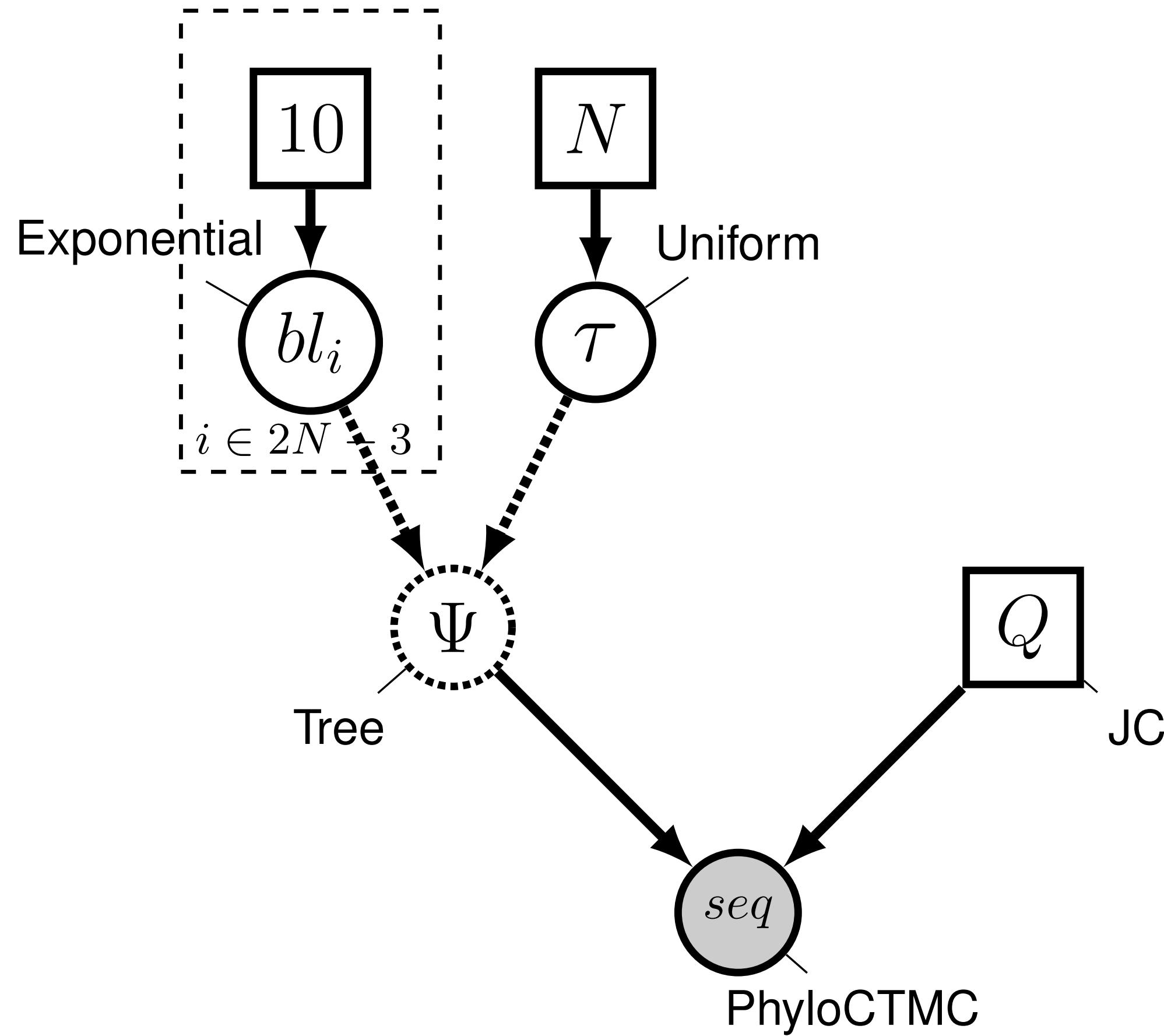


Substitution  
model





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



```

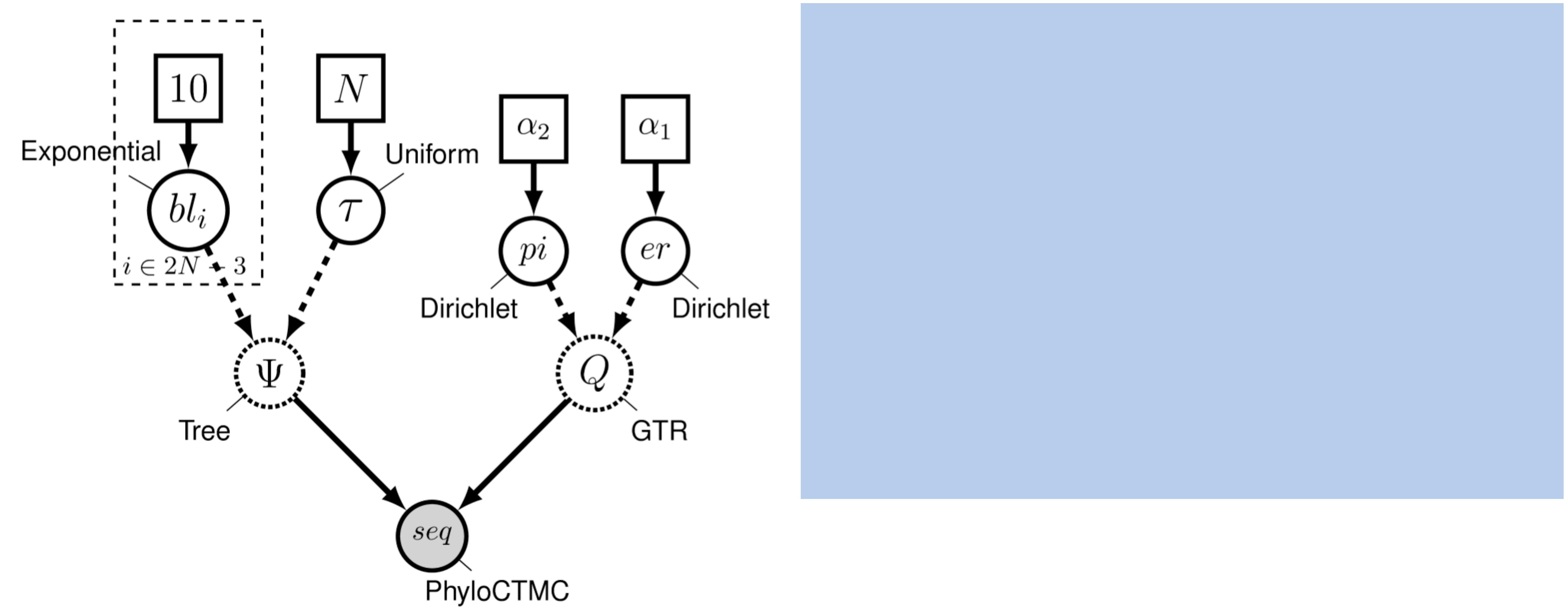
for (i in 1:n_branches) {
  bl[i] ~ dnExponential(10.0)
}
topology ~ dnUniformTopology(taxa)
psi := treeAssembly(topology, bl)

Q <- fnJC(4)

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

seq.clamp( data )

```



# Introduction to MCMC

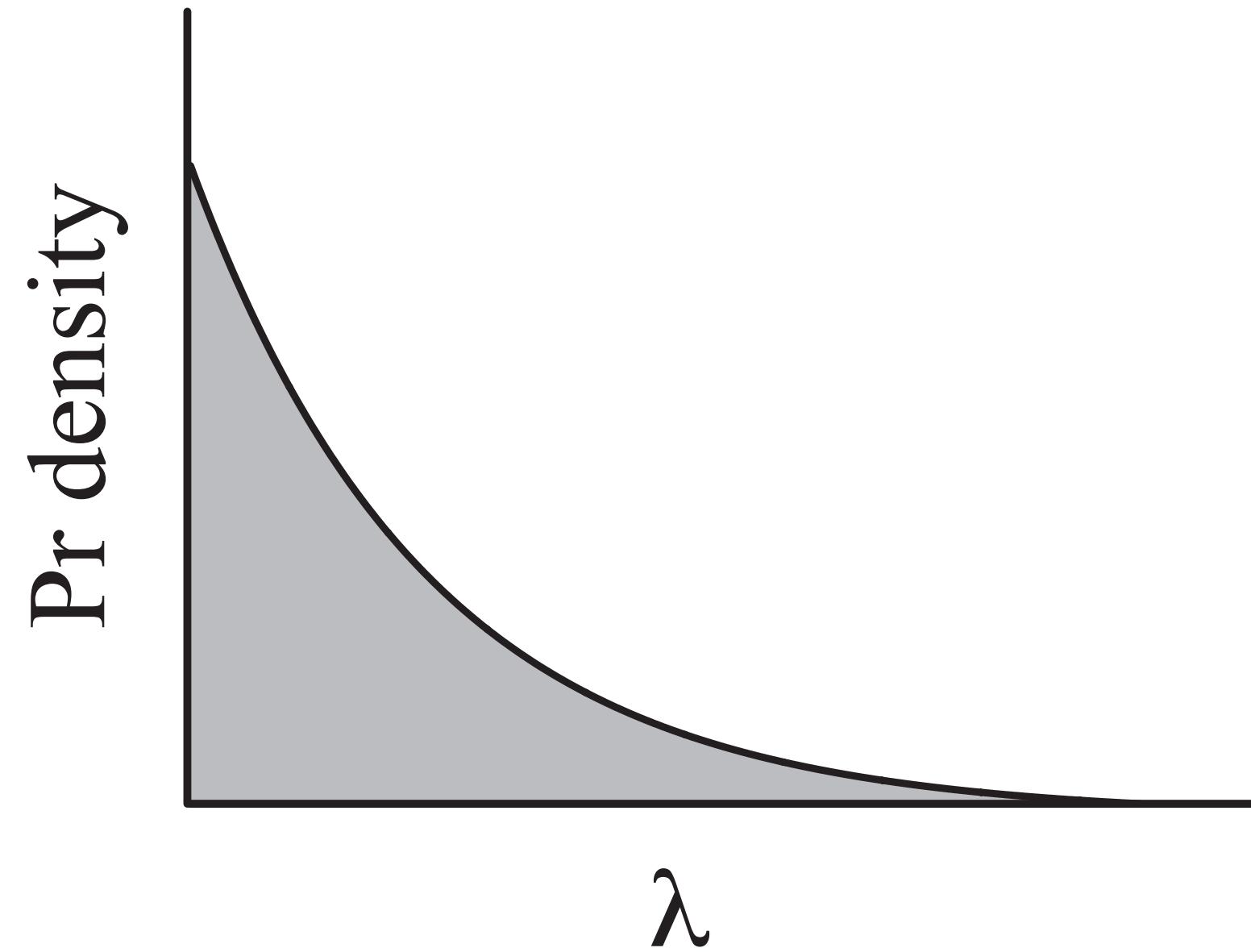
# Probabilities vs. probability densities

In phylogenetics, probabilities are not normally discrete (i.e., represented by a single value)

We're often dealing with a lot of uncertainty and typically work with **probability densities**

Probability densities introduce some complexity

# Probabilities vs. probability densities



$\lambda$  is drawn from an exponential distribution with mean  $\delta$

The x-axis represents the value of our parameter  $\lambda$

The y-axis does have a value but it is not so easily interpretable

The distribution height reflects the relative probability of a given range of values

# Bayesian tree inference

$$\text{posterior} \quad P(E \mid \text{0101...}, \text{1101...}, \text{0100...}) = \frac{\text{likelihood} \quad P(\text{0101...}, \text{1101...}, \text{0100...} \mid E, \text{prior})}{\text{priors} \quad P(E)}$$

marginal probability

# Bayesian tree inference

$$= \frac{P(\text{0101...} | \text{E} \circlearrowleft \text{O} \circlearrowright \text{1}) P(\text{E} \circlearrowleft \text{O} \circlearrowright \text{1})}{\int P(\text{0101...} | \text{E} \circlearrowleft \text{O} \circlearrowright \text{1}) P(\text{E} \circlearrowleft \text{O} \circlearrowright \text{1}) d\text{E} \circlearrowleft \text{O} \circlearrowright \text{1}}$$

this part is incredibly difficult to calculate!

# What is Markov chain Monte Carlo (MCMC)?

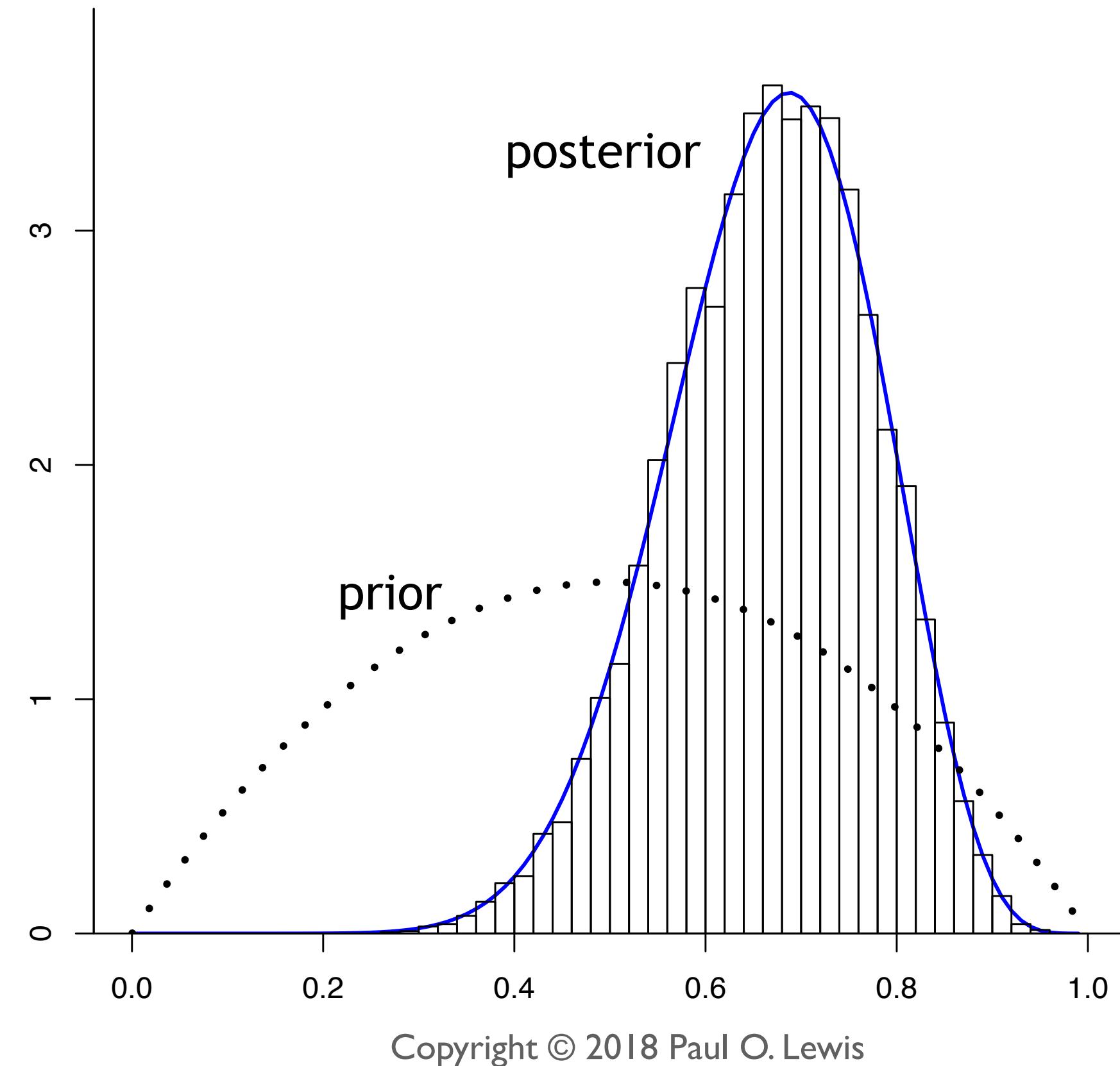
A group of algorithms for approximating the posterior distribution (also known as samplers)

**Markov chain** means the progress of the algorithm doesn't depend on its past

**Monte Carlo** (named for the casino in Monaco) methods estimate a distribution via random sampling

We use this algorithm to visit different regions the parameter space. The number of times a given region is visited will be in proportion to its posterior probability

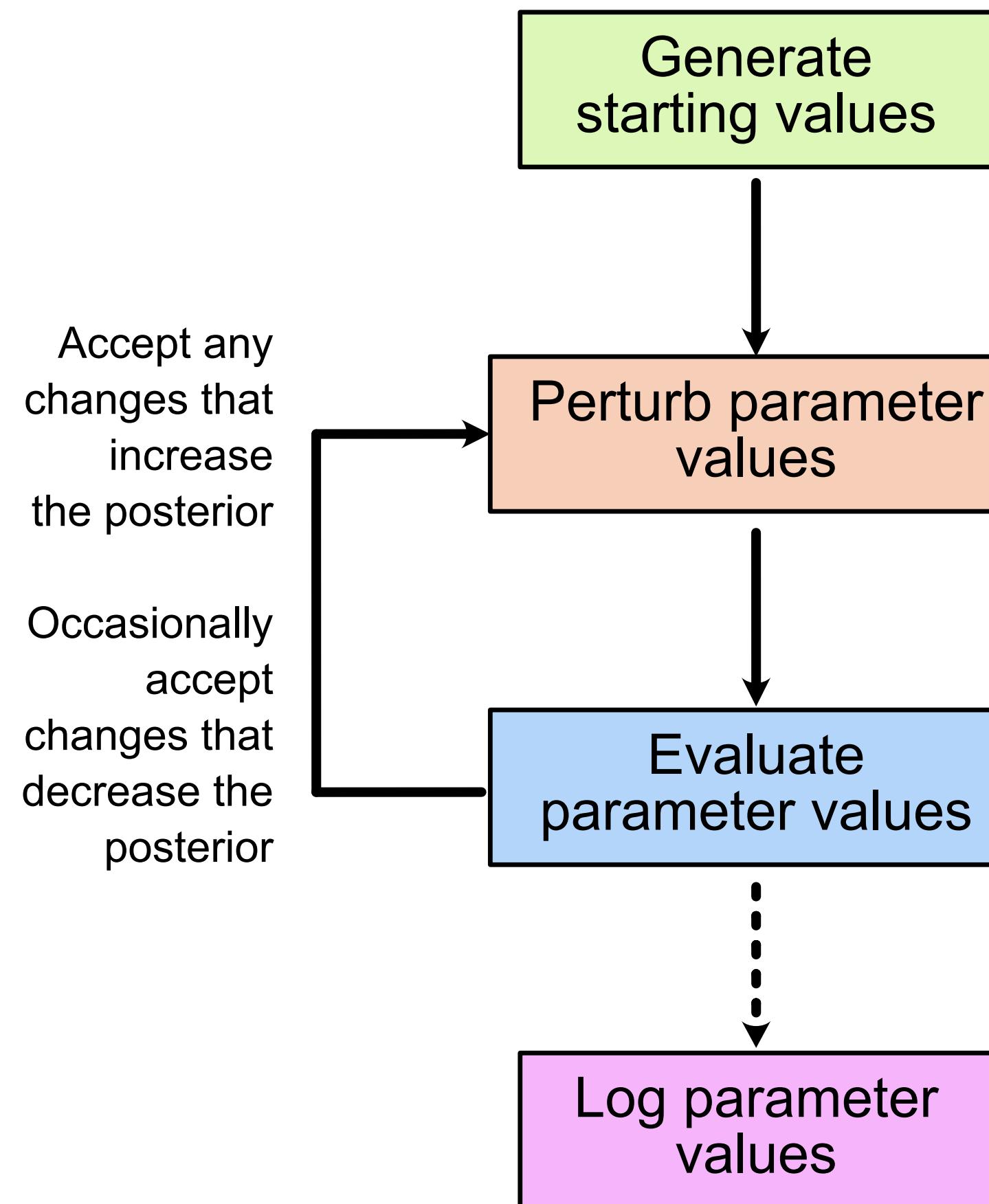
# What is Markov chain Monte Carlo (MCMC)?



The aim is to produce a  
histogram that provides a good  
approximation of the posterior

# The Metropolis-Hastings algorithm

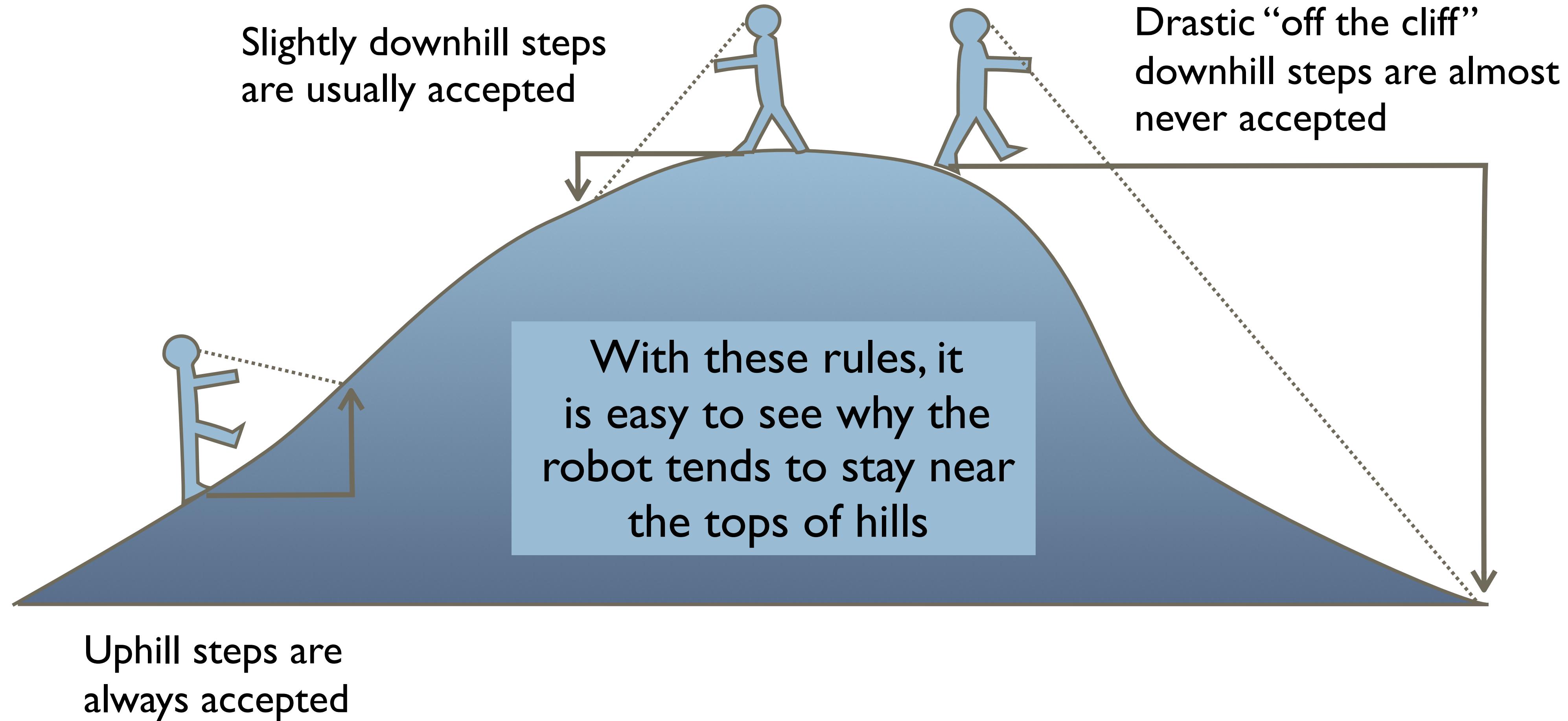
## Flowchart

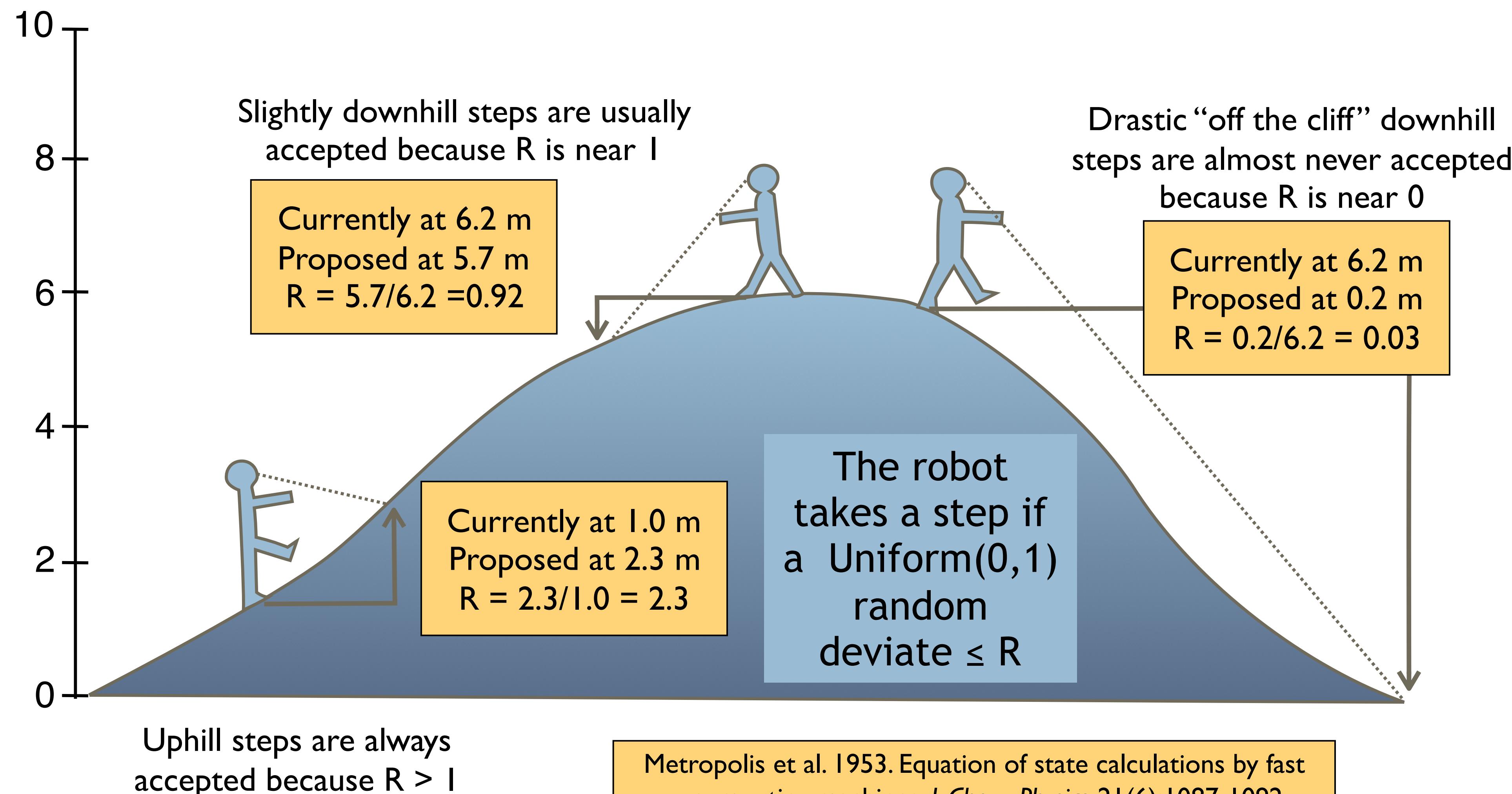


## Pseudocode

```
initialize starting values;  
  
for i in mcmc steps  
do  
    propose new parameter values;  
    calculate the Hastings ratio R;  
  
    if( R > 1 )  
        accept the new values;  
    else  
        accept the new values with Pr = R;  
  
    store the values with frequency j;  
done
```

# MCMC robot's rules





When calculating the ratio ( $R$ ) of posterior densities, the marginal probability of the data cancels.

$$\frac{p(\theta^* | D)}{p(\theta | D)} = \frac{\frac{p(D | \theta^*) p(\theta^*)}{p(D)}}{\frac{p(D | \theta) p(\theta)}{p(D)}} = \frac{p(D | \theta^*) p(\theta^*)}{p(D | \theta) p(\theta)}$$

Posterior  
odds

Apply Bayes' rule to  
both top and bottom

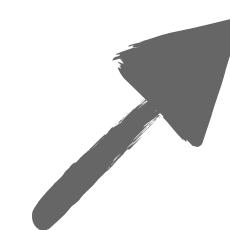
Likelihood  
ratio

Prior  
odds

# Hastings ratio

new parameter  
values

$$R = \frac{P(\text{E}^* | \text{0101... 1101... 0100...})}{P(\text{E} | \text{0101... 1101... 0100...})}$$



=

=

$$\frac{\cancel{P(\text{0101... 1101... 0100...})} P(\text{E}^* | \text{0101... 1101... 0100...}) P(\text{E}^*)}{\cancel{P(\text{0101... 1101... 0100...})} P(\text{E} | \text{0101... 1101... 0100...}) P(\text{E})}$$

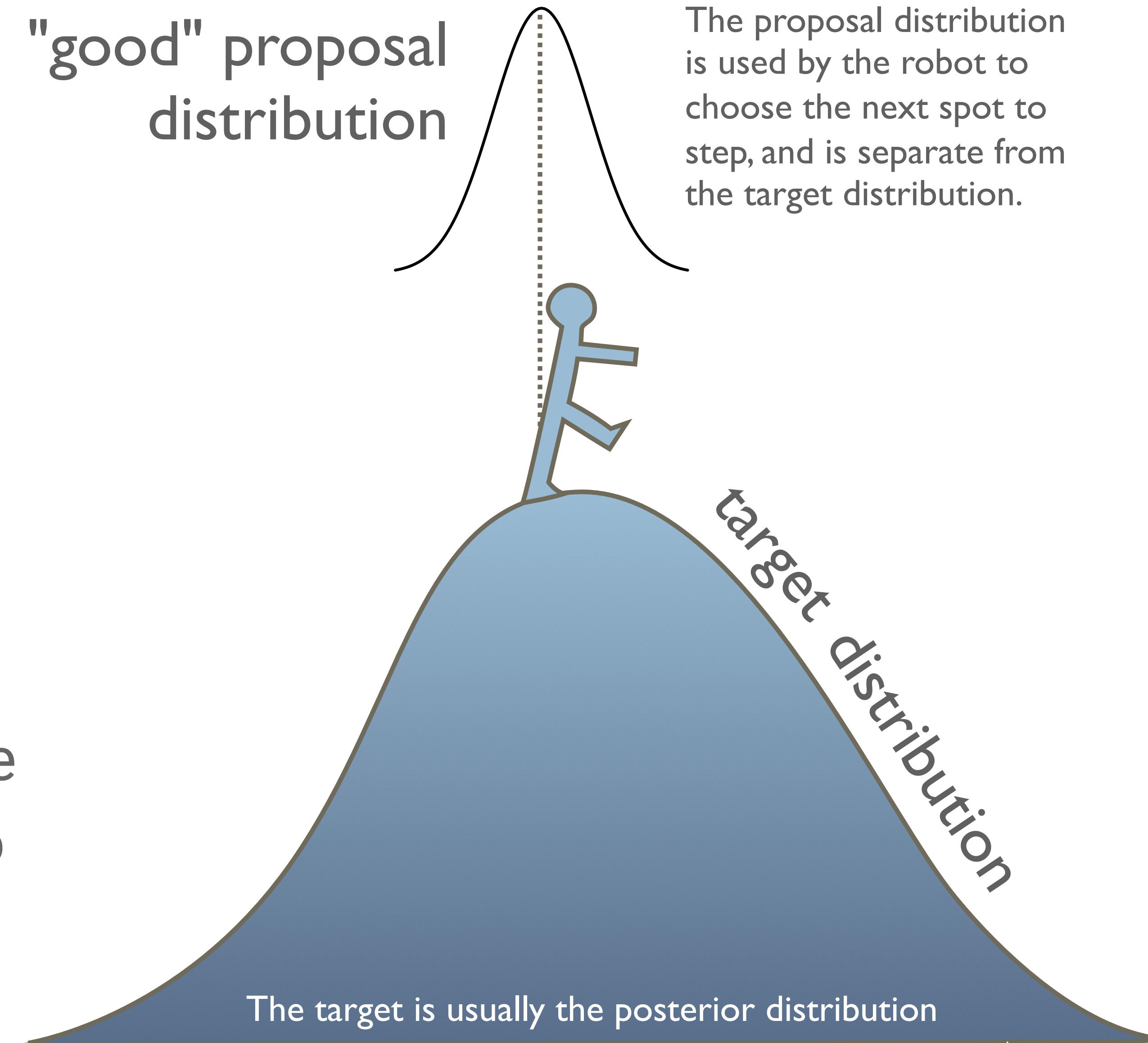
The marginal probability of the data cancels out

All we're left to calculate is the likelihood ratio and the prior odds ratio

# Moves

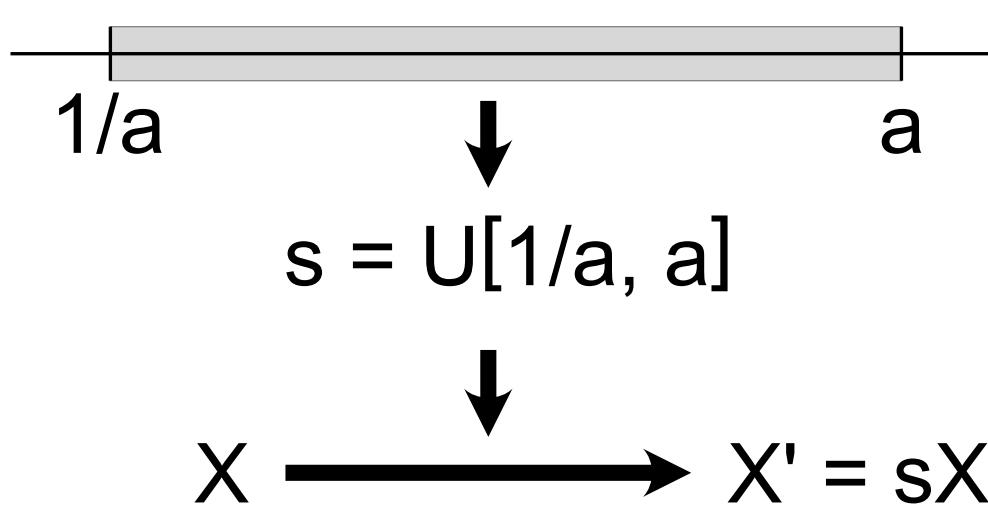
Known as **moves**,  
**proposals** or **operators** –  
how we move across the  
parameter space

i.e., how do we select the  
next set of parameters to  
evaluate

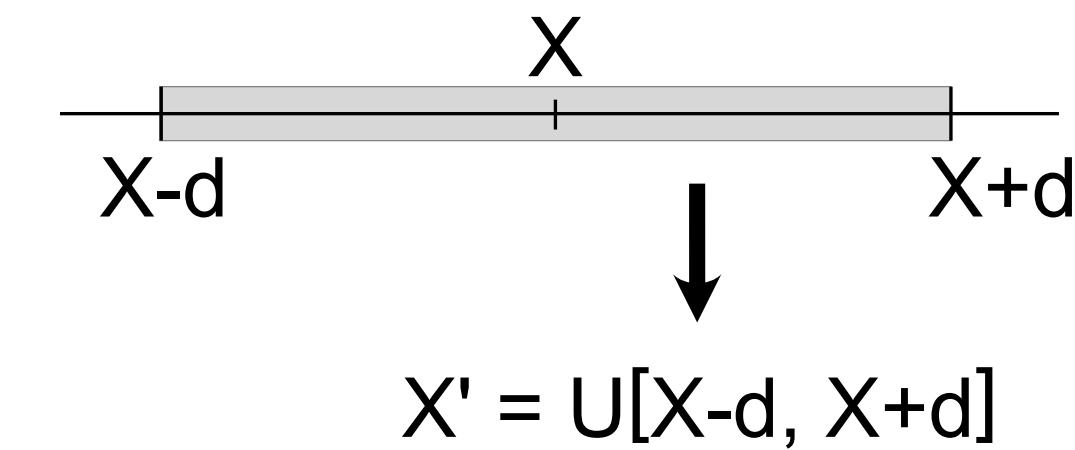


# Moves

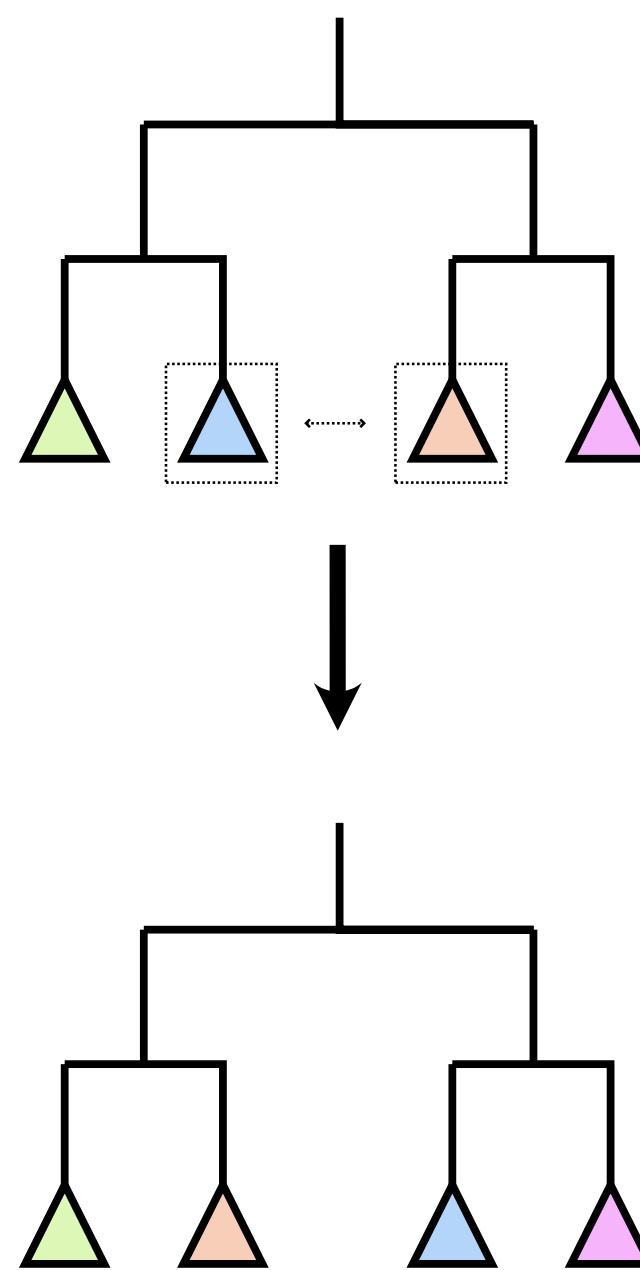
Scaling move



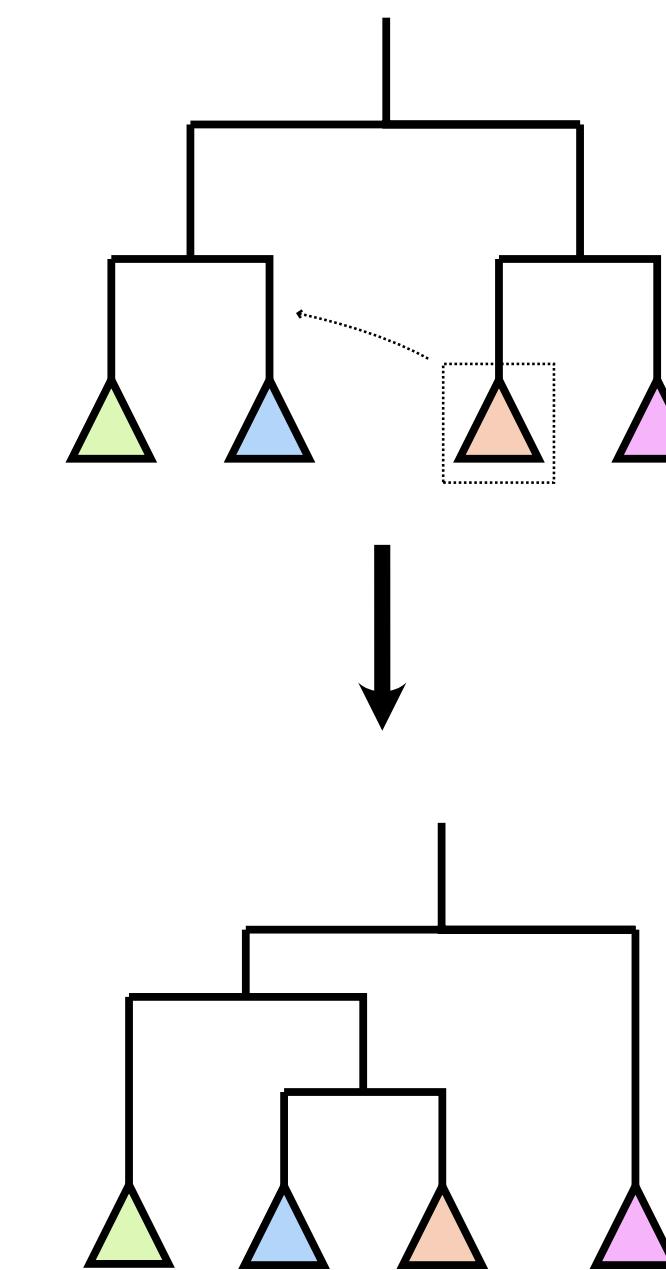
Sliding move



SubTree eXchange  
(STX) move



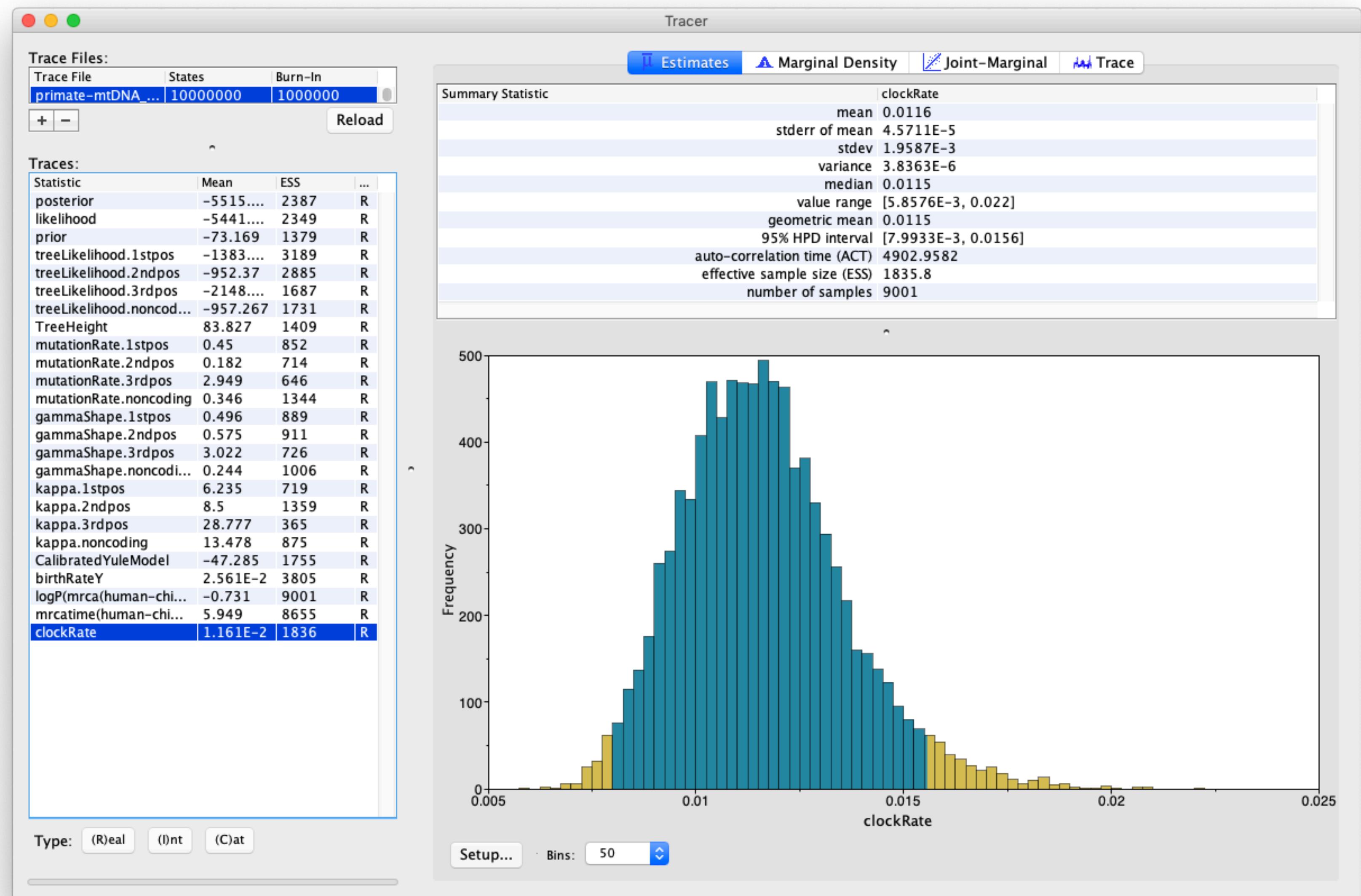
Subtree Pruning and  
Regrafting (SPR) move



# MCMC Robot demo

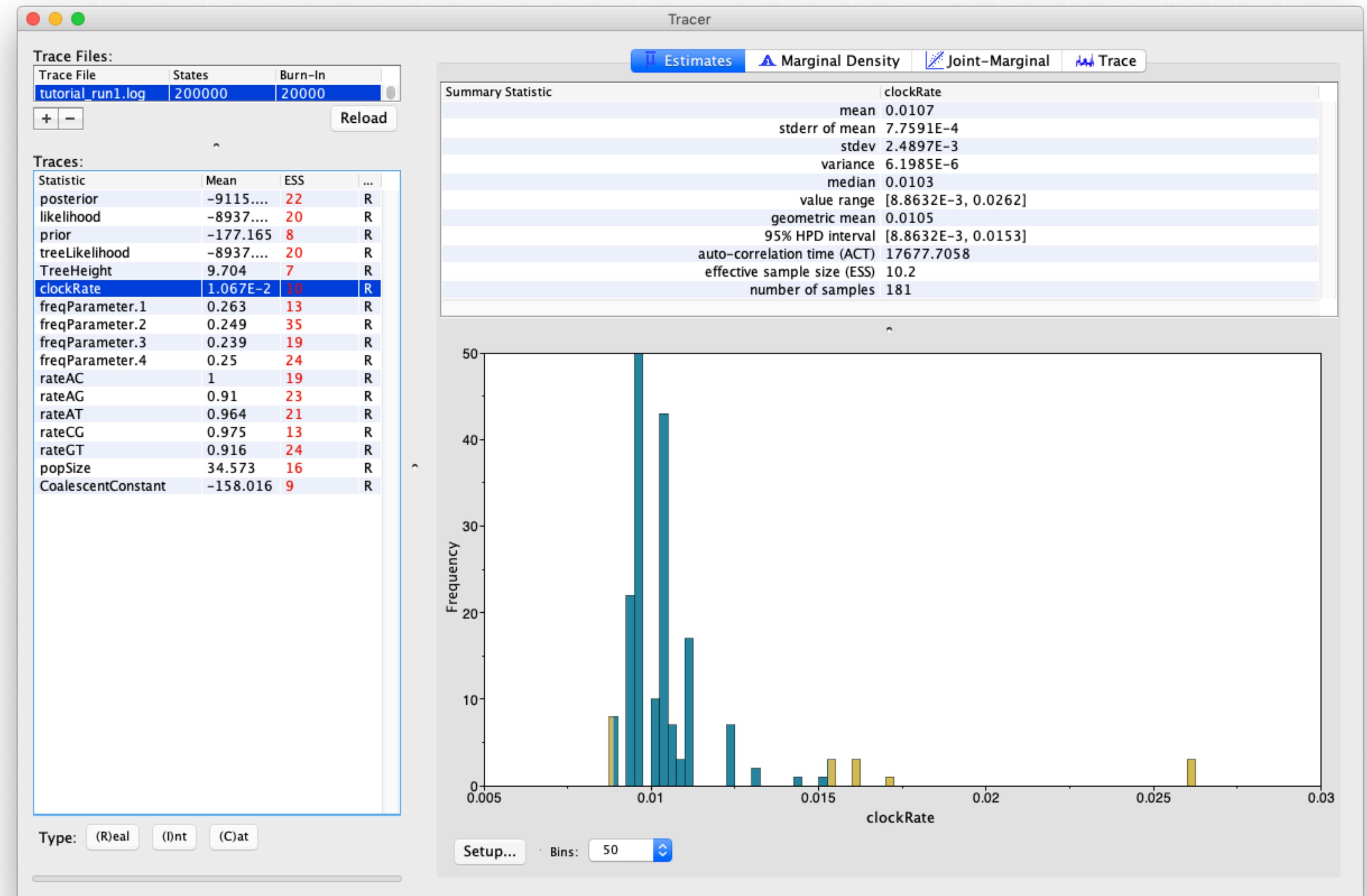
# Summarising the posterior

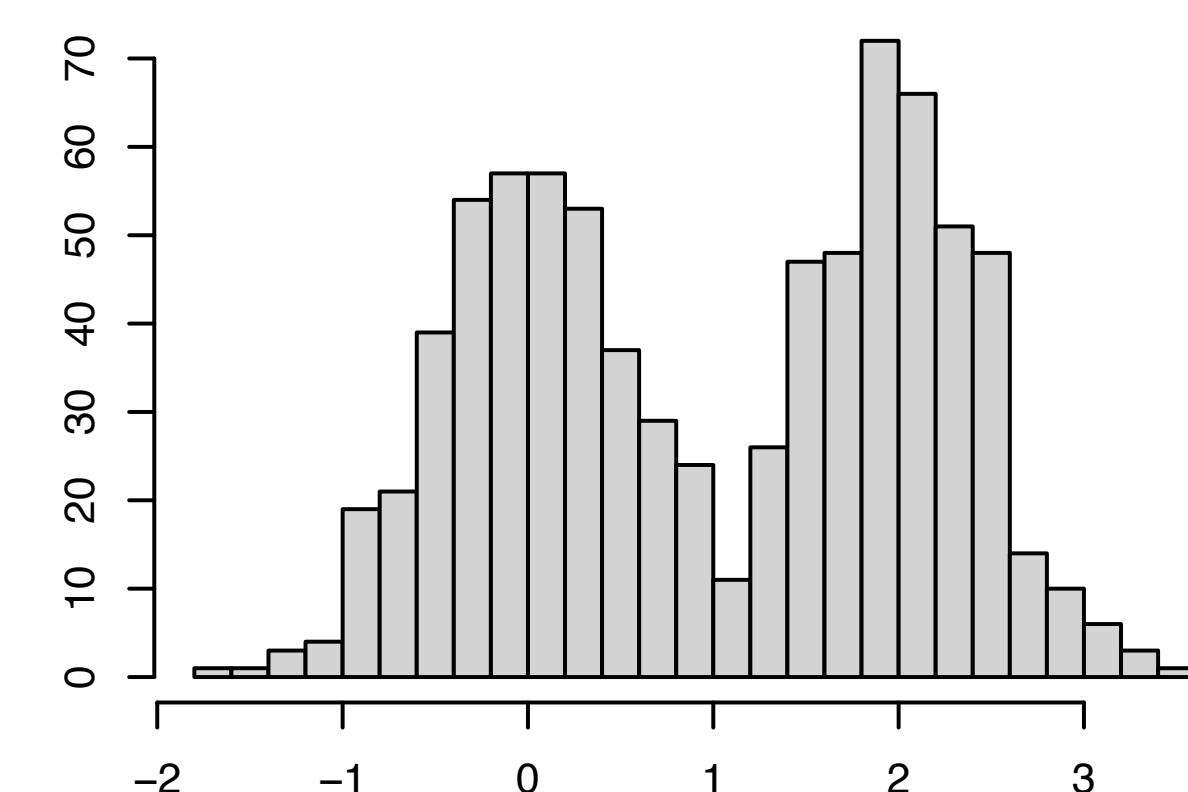
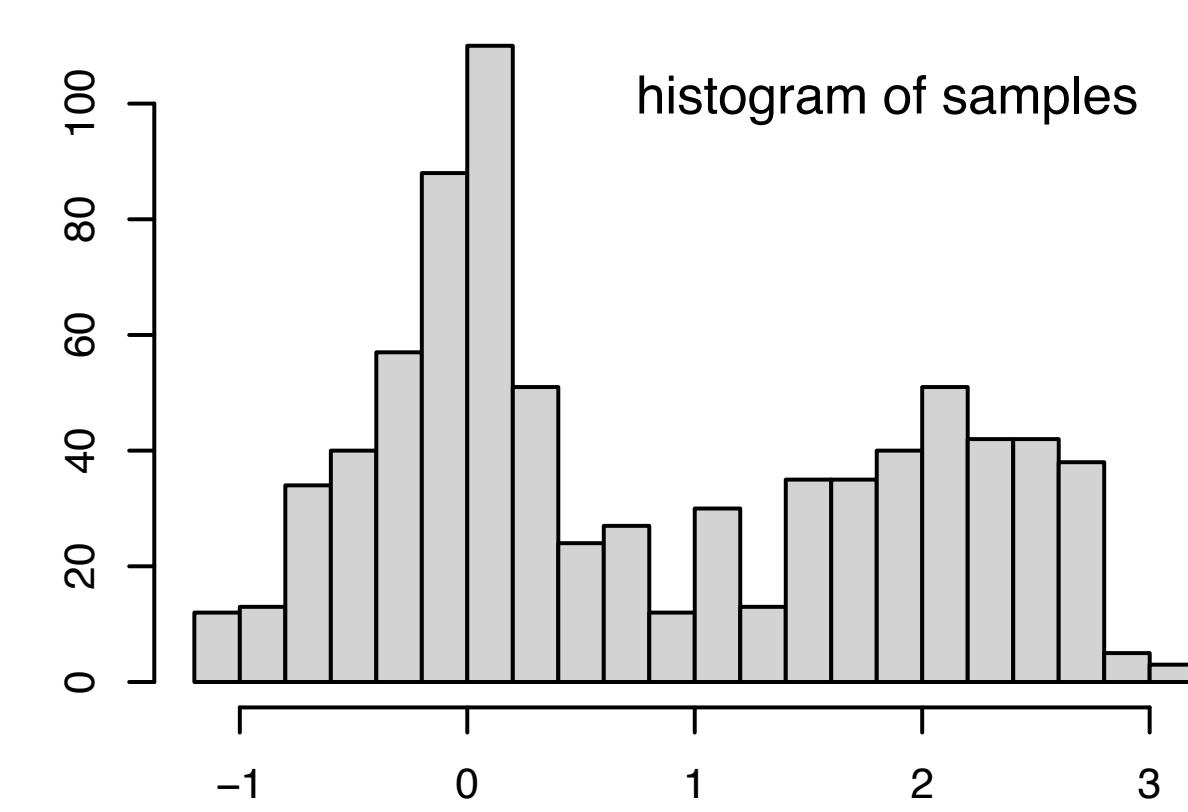
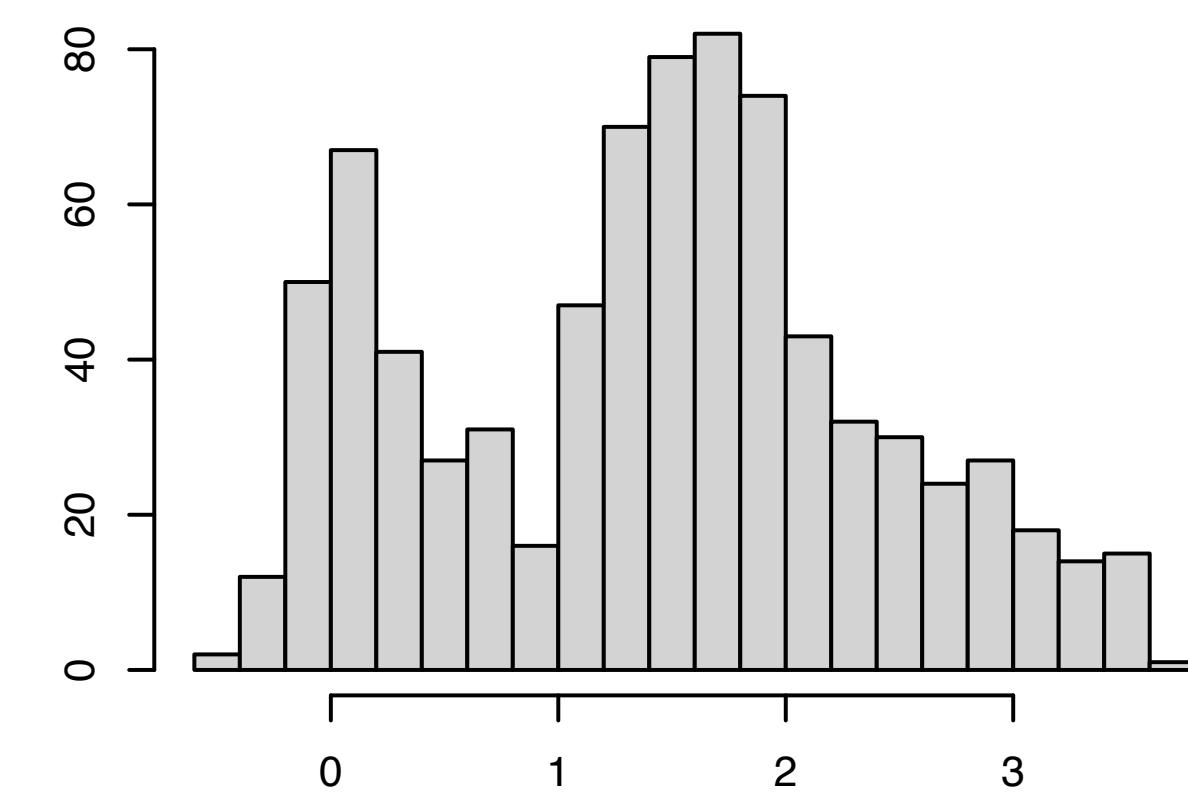
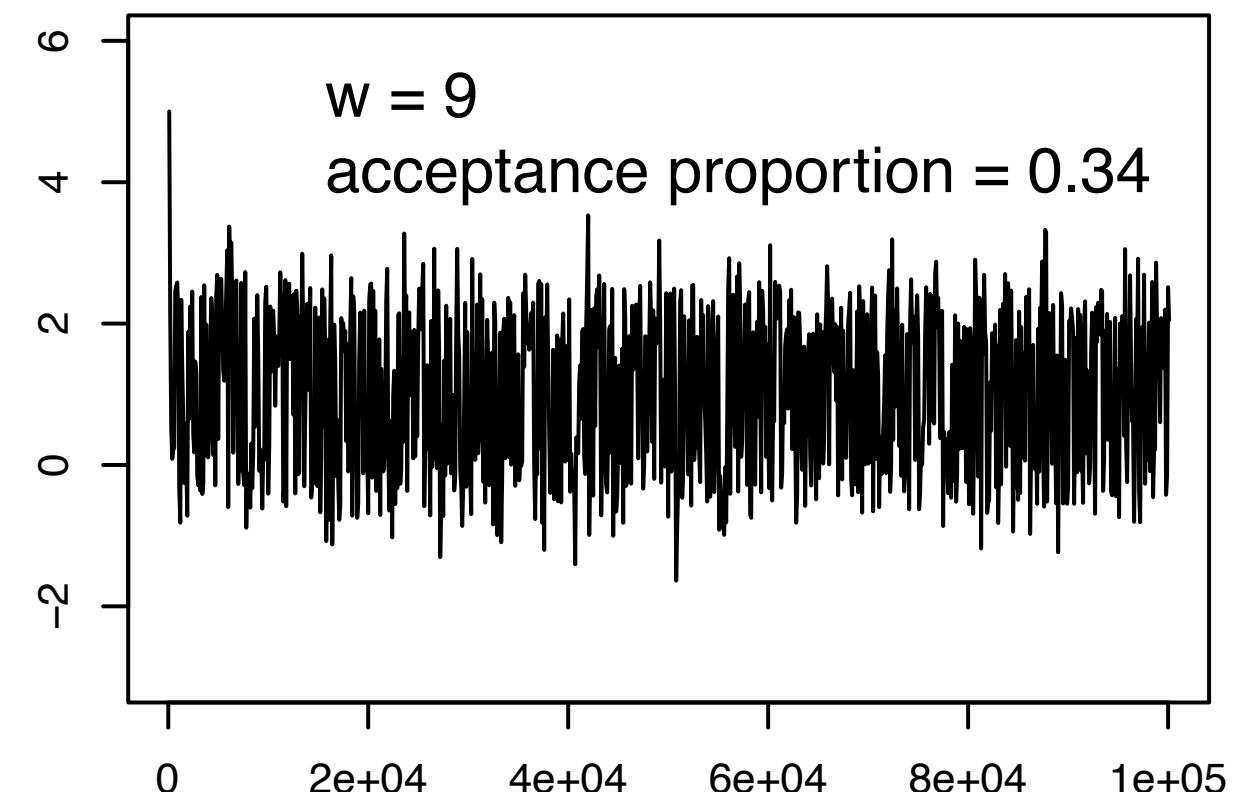
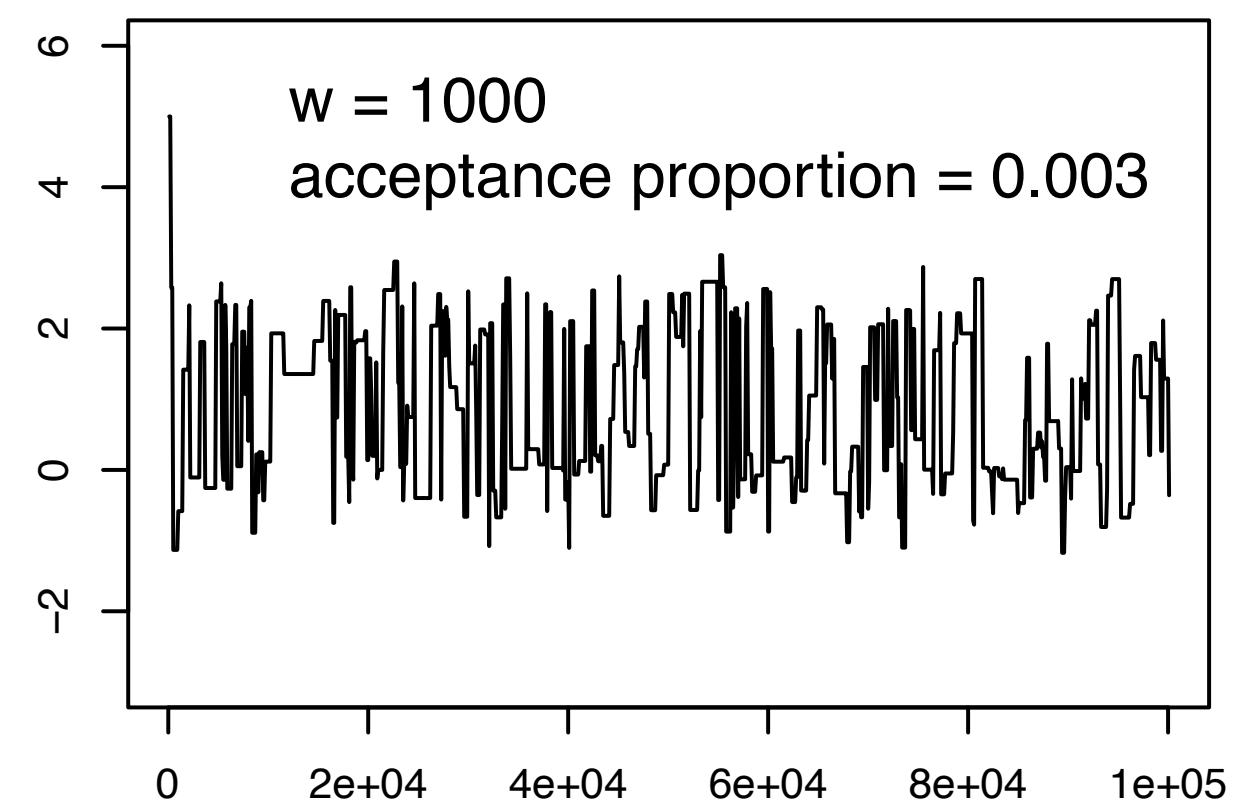
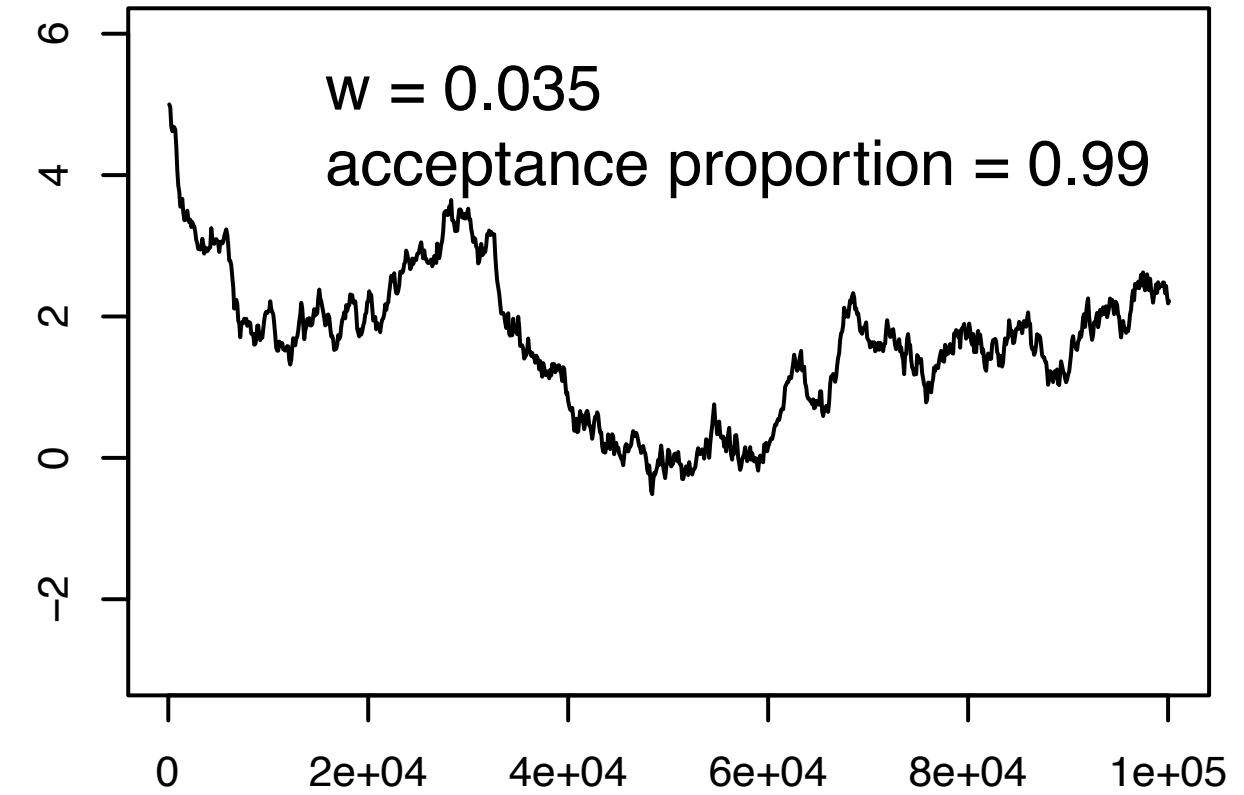
Tracer is an amazing program for exploring MCMC output



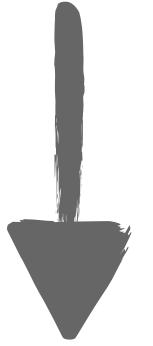
# Summarising the posterior

Tracer is an amazing program for exploring MCMC output





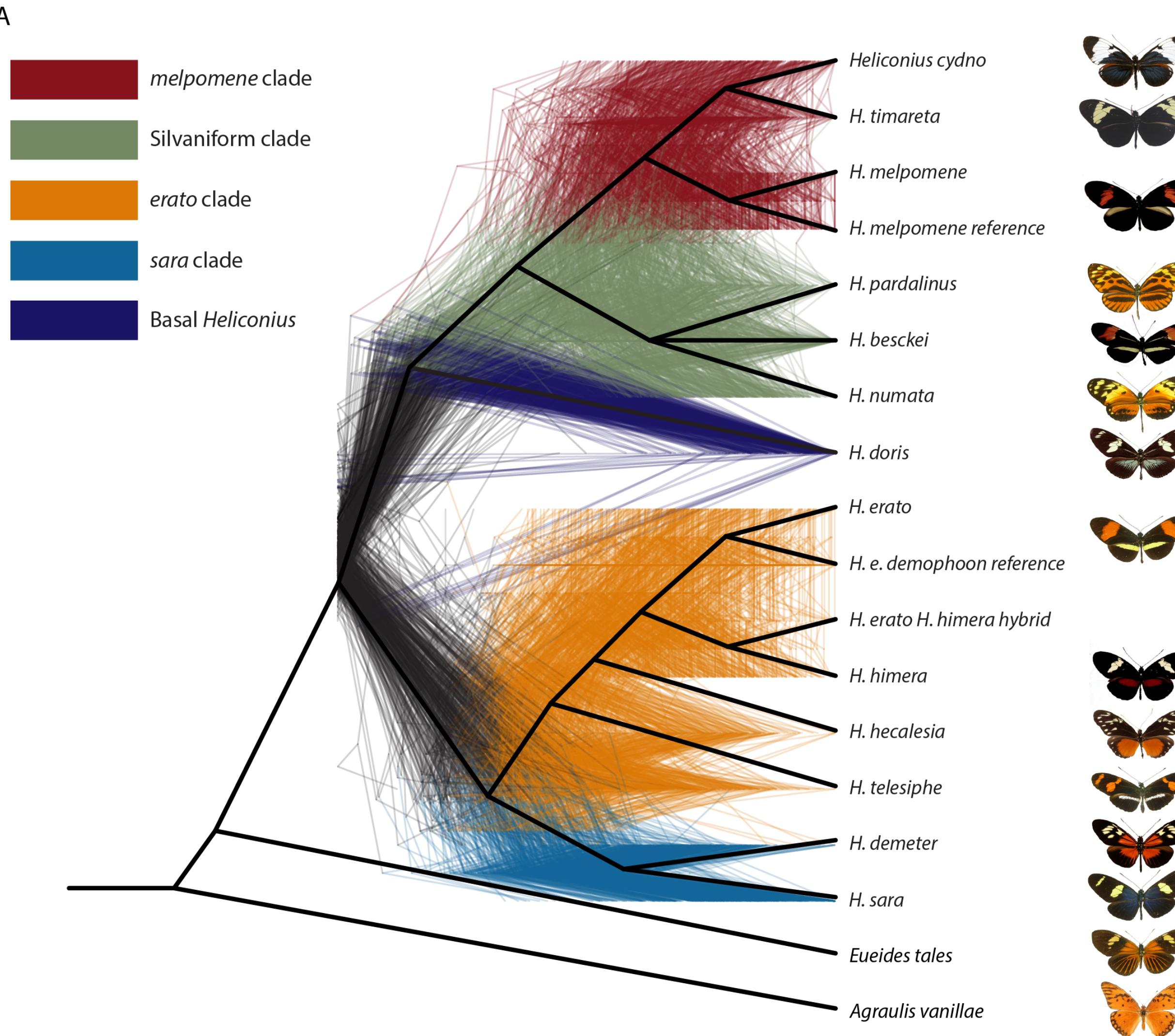
Better!



# Summarising the posterior

Summarising trees is  
much more challenging

Presenting a single  
summary tree can be  
misleading



# Summarising the posterior

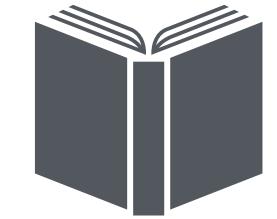
**Maximum clade credibility (MCC) tree** – the tree in the posterior sample that has the highest posterior probability (i.e., clade support) across all nodes

The **95% highest posterior density (HPD)** – the shortest interval that contains 95% of the posterior probability. The Bayesian equivalent of the 95% confidence interval

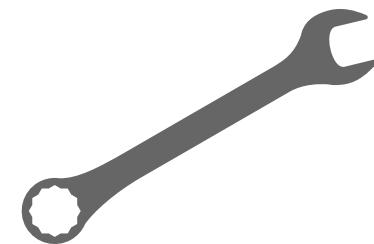
**Marginal posterior density** – the probability of a parameter regardless of the value of the others, represented by the histogram

# Exercise

# Reading



Bayesian Inference of Phylogeny and Its Impact on Evolutionary Biology *Huelsenbeck et al. (2001)*



[MCMC Robot App](#) by Paul Lewis



For later: Practical guidelines for Bayesian phylogenetic inference using Markov chain Monte Carlo (MCMC) *Barido-Sottani et al. (2024)*