

Phylogenetics

Introduction to Bayesian
inference

RL-V3 MPP

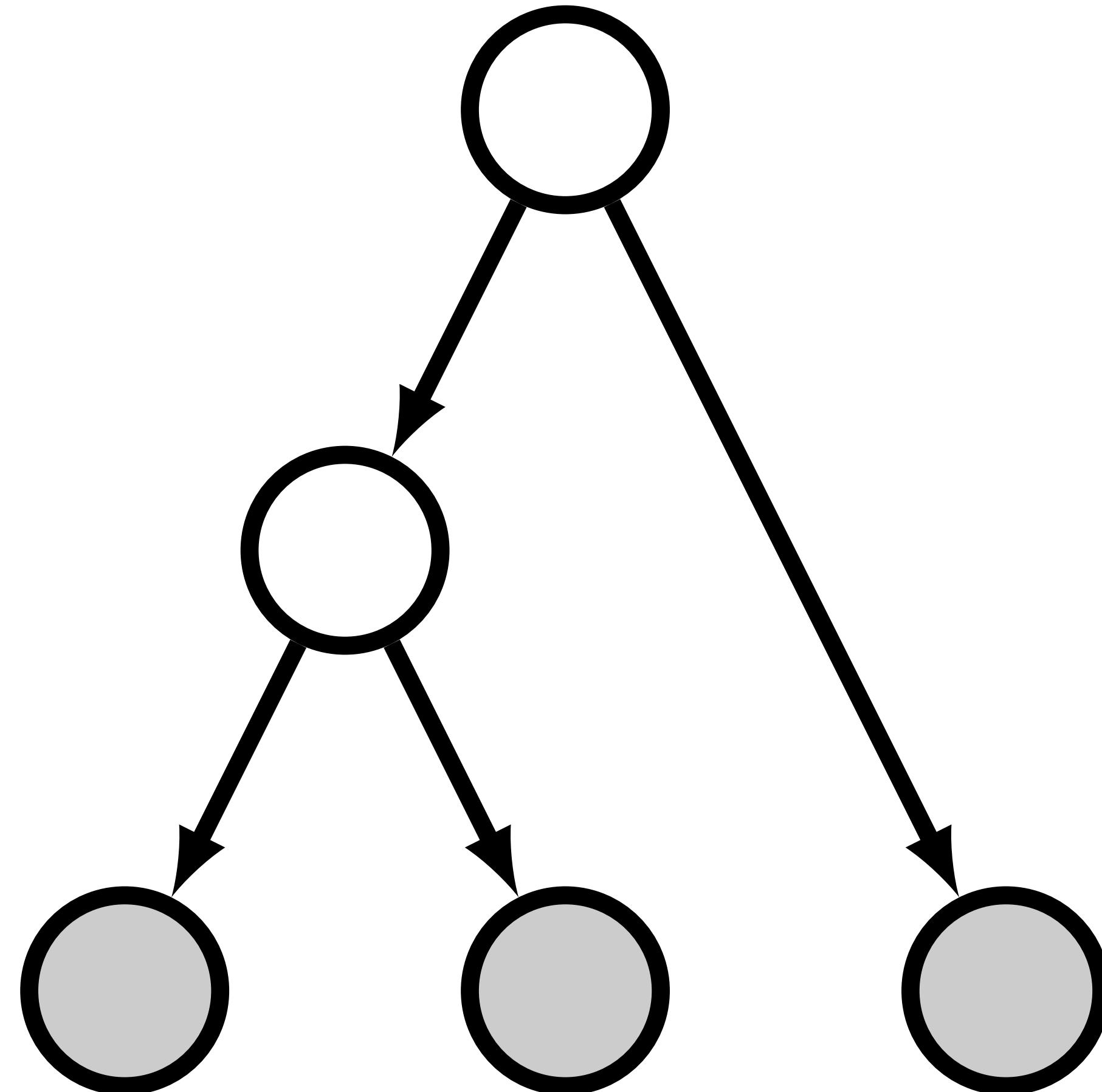
Rachel Warnock

30.04.24



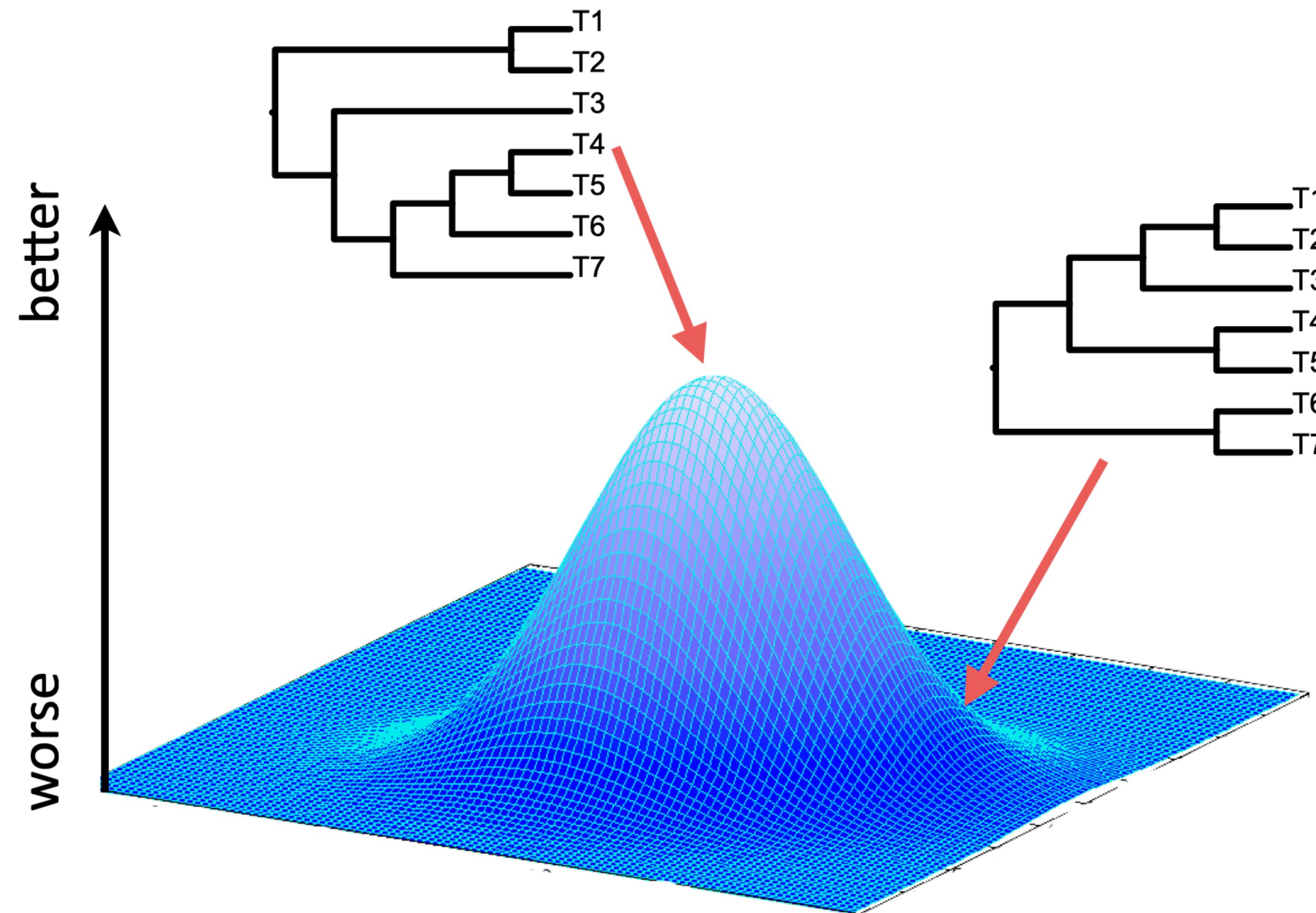
Today's objectives

- 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

Homework



Install the software [RevBayes](#)



[Phylogenetics primer part 1: tree terminology and substitution models Paul Lewis](#)

→ See the question guide

Q&A Phylogenetics primer part 1 by *Paul Lewis*

(the answers provided here are my interpretation of these concepts – answers may vary!)

1. In your own words, how would you describe the terms conditional probability and likelihood?

Conditional probability → the probability of an event, dependent on the value of some other event

Likelihood → the probability of our observations given a set of assumptions (i.e., the model) and parameter values

2. Why do we calculate likelihoods on a log scale?

→ Because likelihoods can get incredibly small – see for yourself using R!

3. How does the probability of transitioning from one character state to another (e.g., from A to T) change over time?

use the [Transition Probability](#) tool to explore this further

→ the probability of change increases with time

Felsenstein's pruning algorithm

4a. What do we need this for?

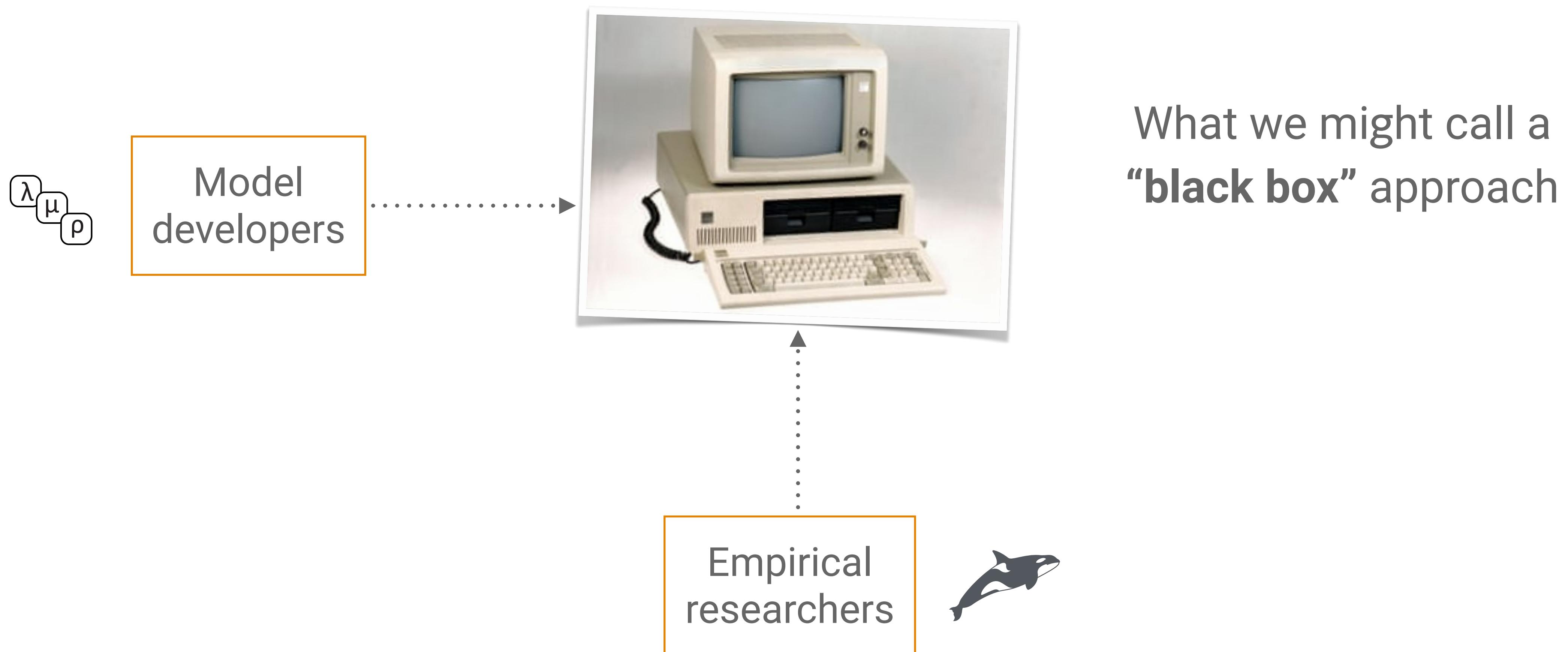
→ to calculate the likelihood of a tree (given an alignment and a substitution model, taking into account all possible ancestral states at every node)

4b. Can you describe the gist of Felsenstein's pruning algorithm?

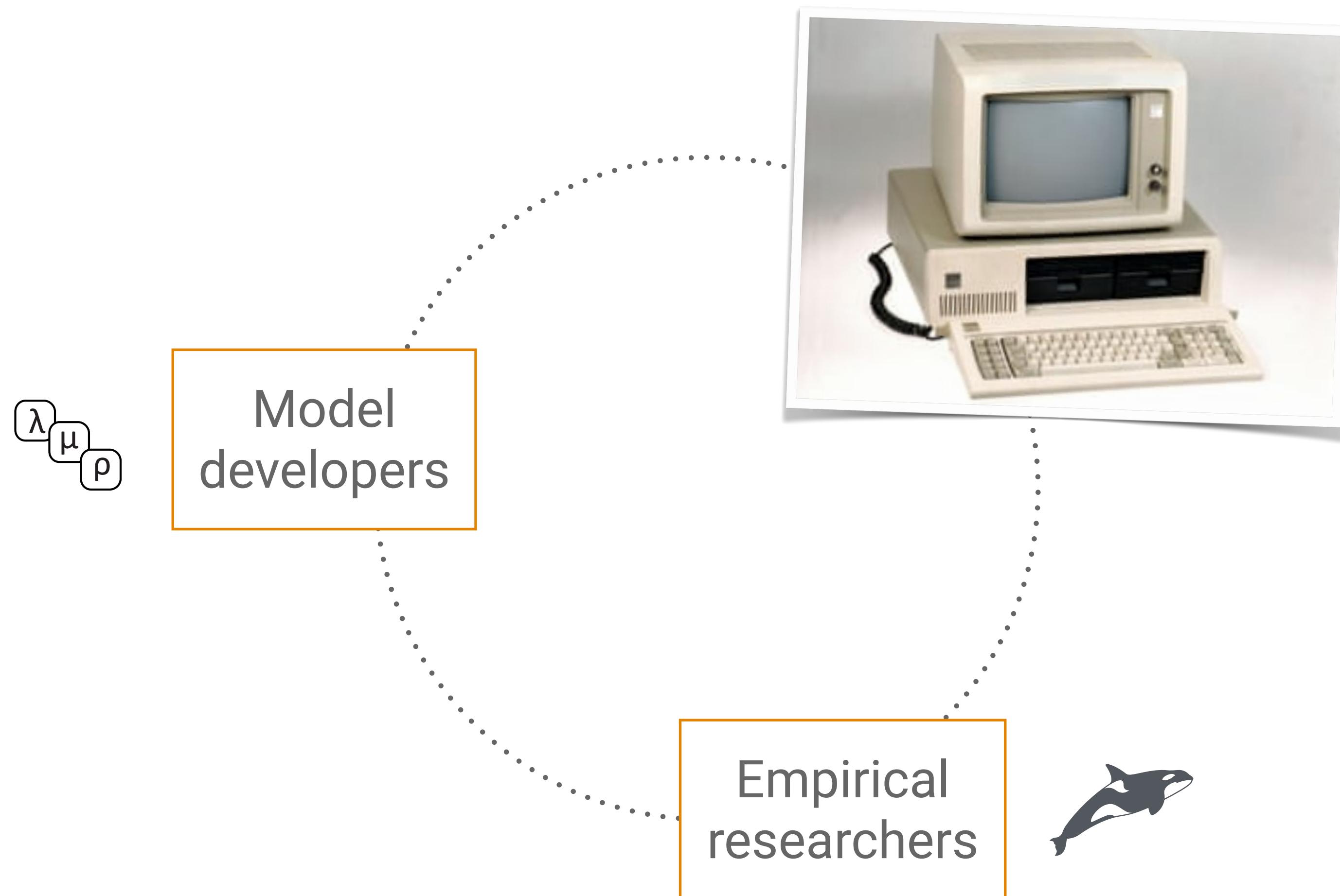
For a good description of Felsenstein's pruning algorithm see [Section 8.8](#) of Phylogenetic Comparative Methods by Harmon

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

The screenshot shows a web browser window displaying the RevBayes GitHub page (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.

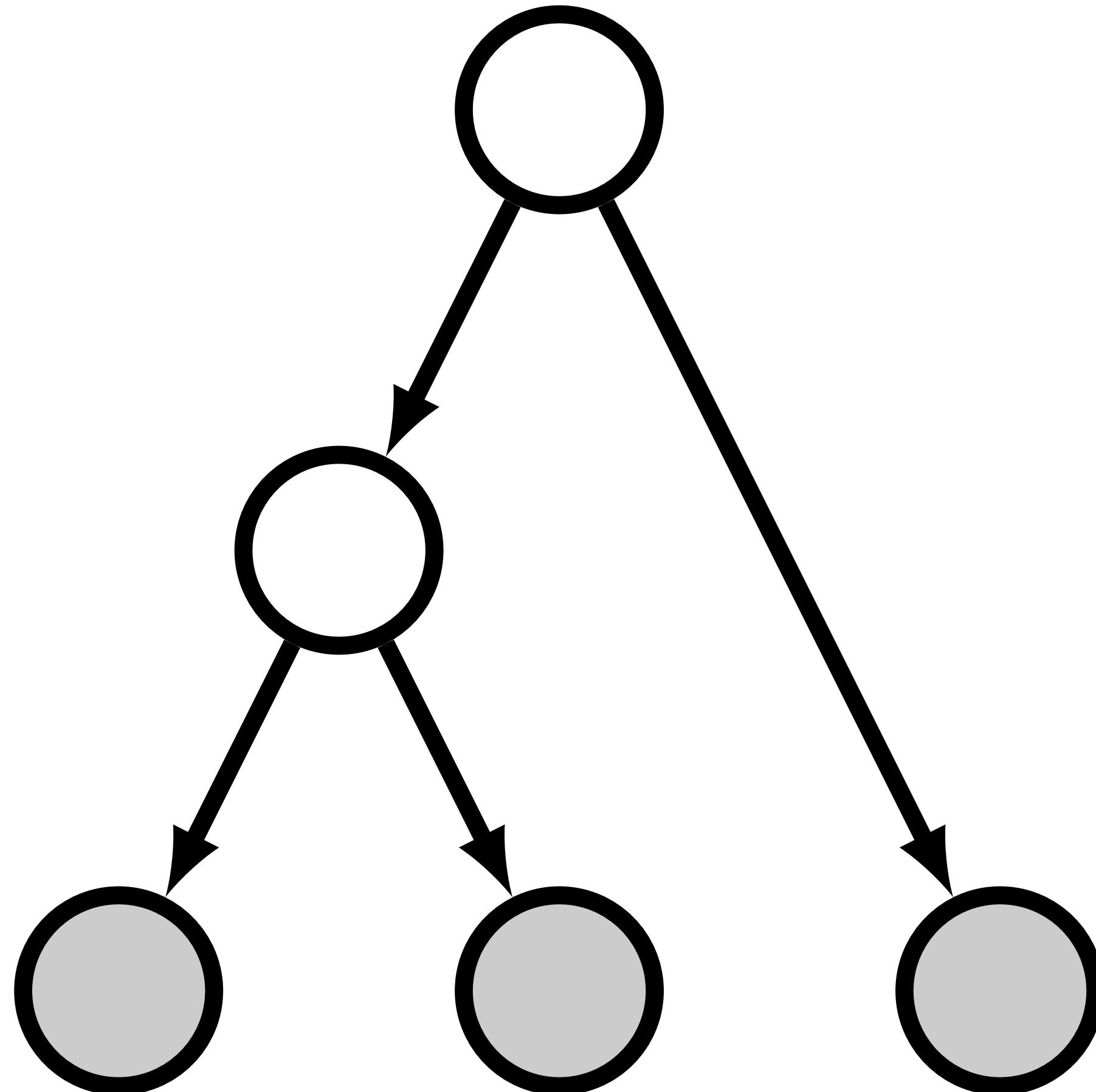
[GitHub](#) | [License](#) | [Citation](#) | [Users Forum](#)

Graphical models

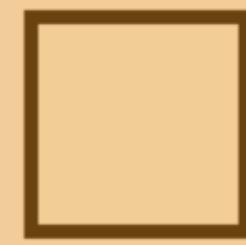
Graphical models

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

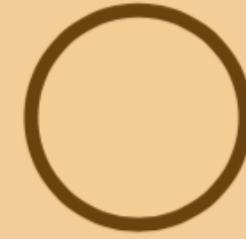
Depict the conditional dependence structure of parameters and other random variables



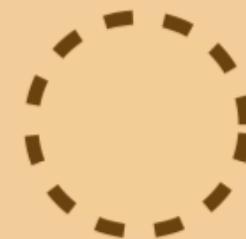
Types of variables (nodes)



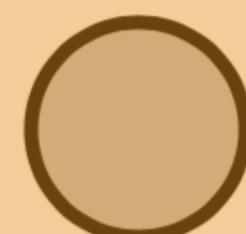
a) Constant node



b) Stochastic node



c) Deterministic node



d) Clamped node
(observed)

a. fixed value variables

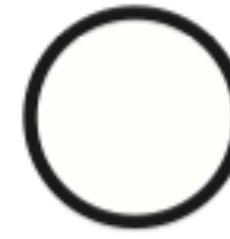
b. random variables that depend on other variables

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

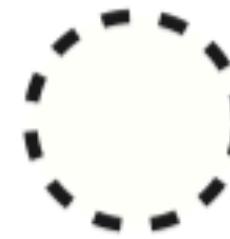
d. observed stochastic variables (data)



a) Constant node



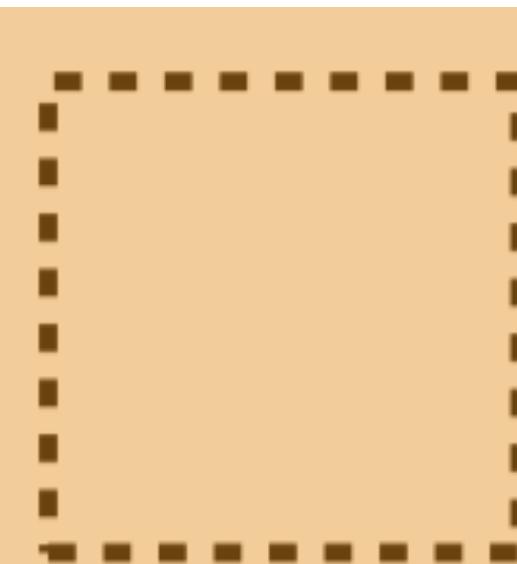
b) Stochastic node



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)

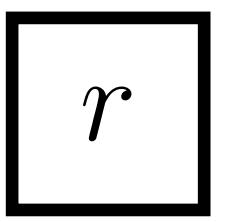
e. repetition over multiple variables (equivalent to a loop)

Specifying graphical models using the Rev syntax

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

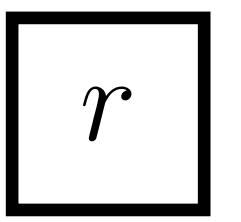
Operator	Variable
<code><-</code>	constant variable
<code>~</code>	stochastic variable
<code>:=</code>	deterministic variable
<code>node.clamp(data)</code>	clamped variable
<code>=</code>	inference (<i>i.e.</i> , non-model) variable
<code>for(i in 1:N){...}</code>	plate

a)

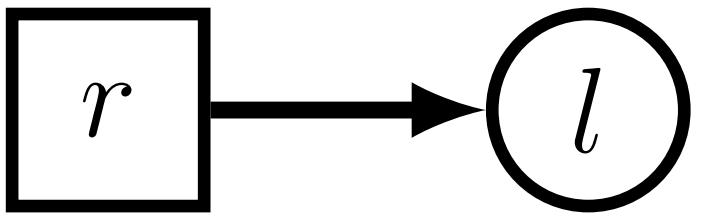


```
# constant node  
r <- 10
```

a)



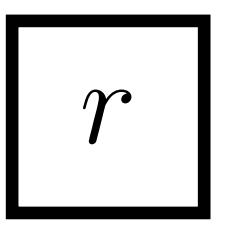
b)



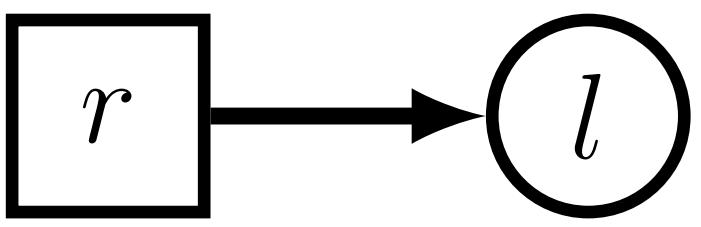
```
# constant node  
r <- 10
```

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

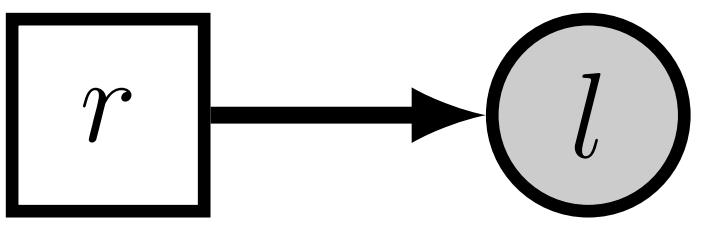
a)



b)



c)

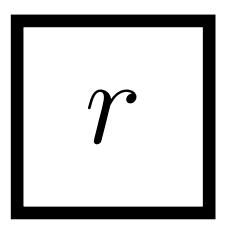


```
# constant node  
r <- 10
```

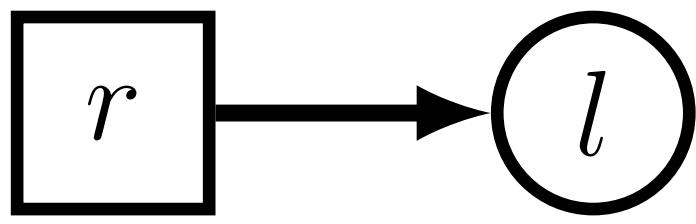
```
# stochastic node  
l ~ dnExp(r)
```

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

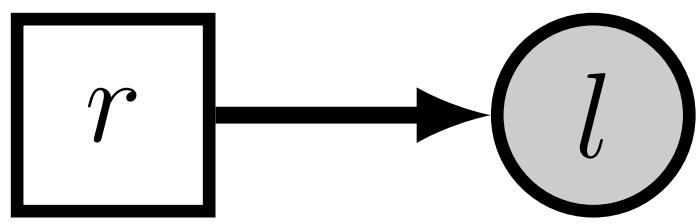
a)



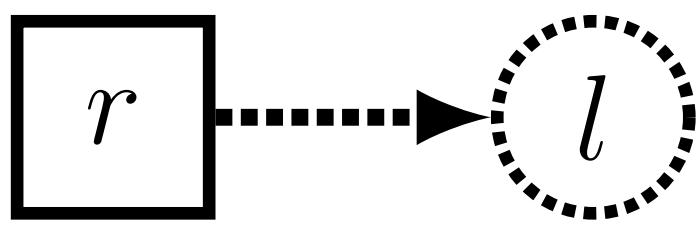
b)



c)



d)



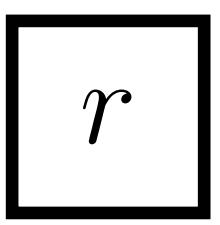
```
# constant node  
r <- 10
```

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

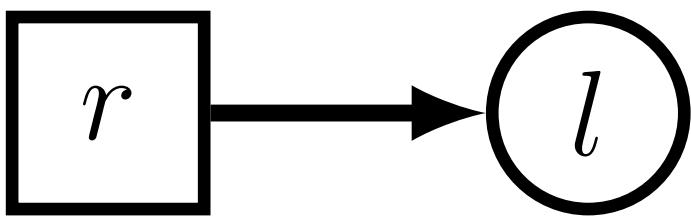
```
# stochastic node (observed)  
l.clamp(0.1)
```

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

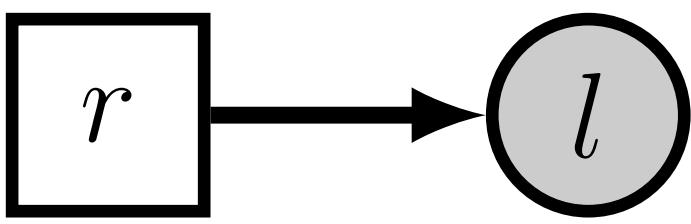
a)



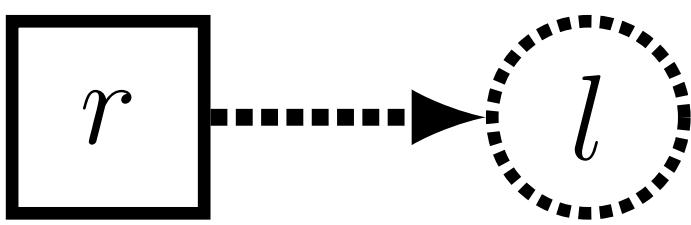
b)



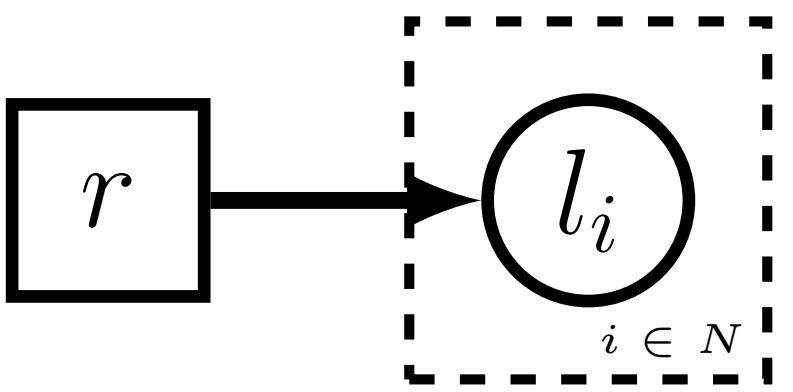
c)



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)  
}
```

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

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

Bayes' theorem

priors

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

Bayes' theorem

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

marginal probability

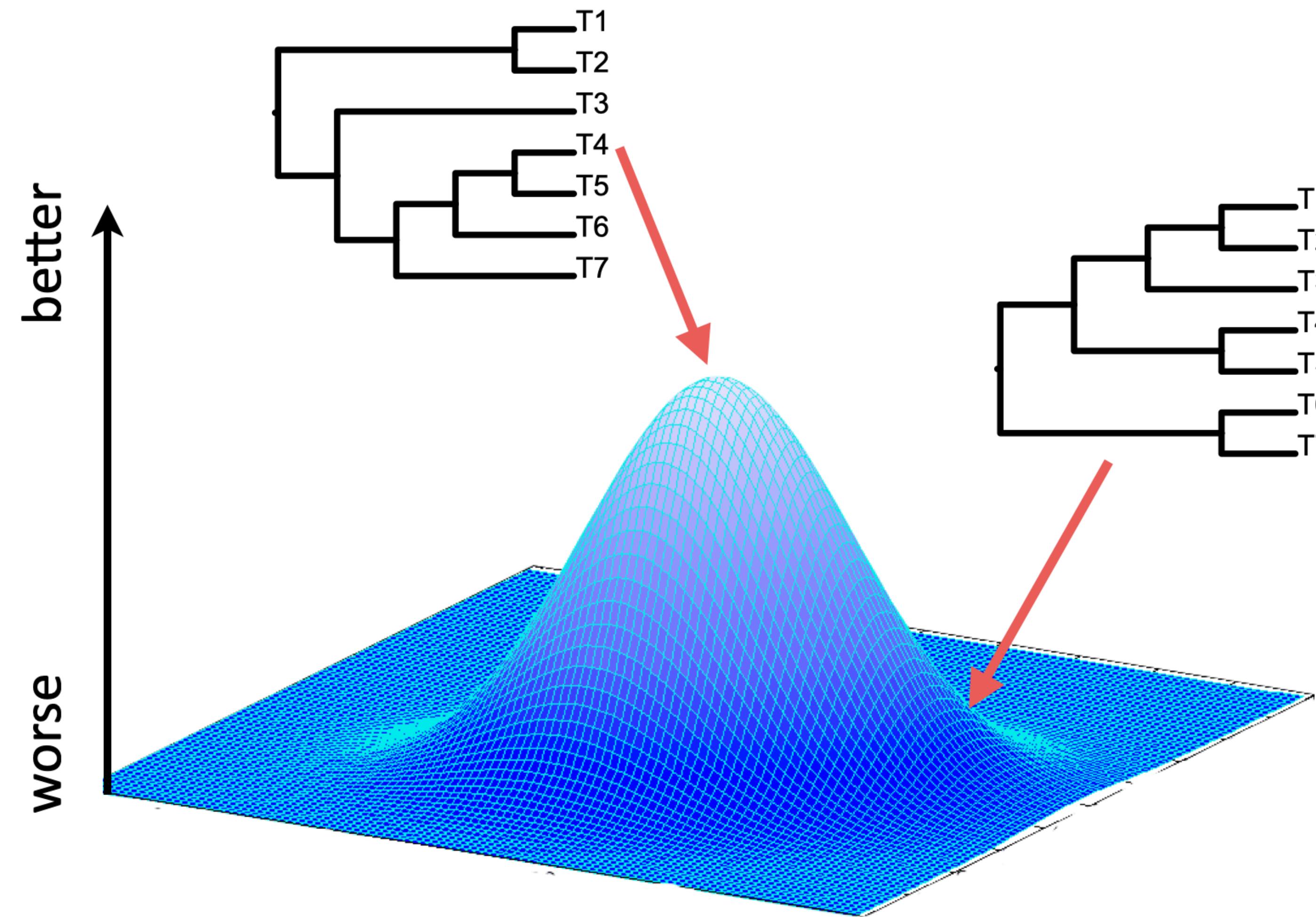
Bayes' theorem

posterior

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

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

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

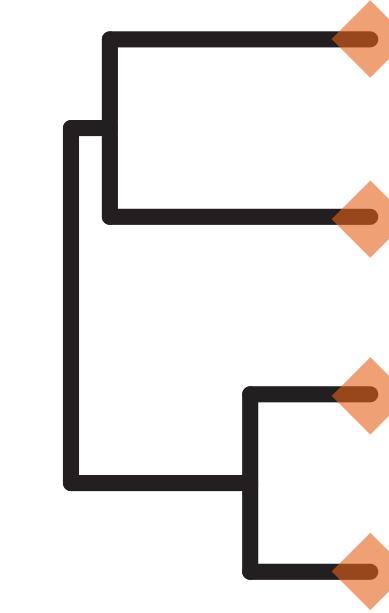
Note these are not the only approaches to tree-building but they are the most widely used

Components used to infer trees

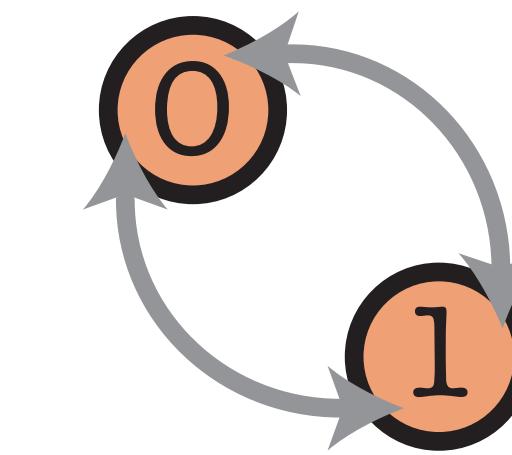
without considering time

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

data
sequences or
characters



tree
topology and
branch lengths



substitution
model

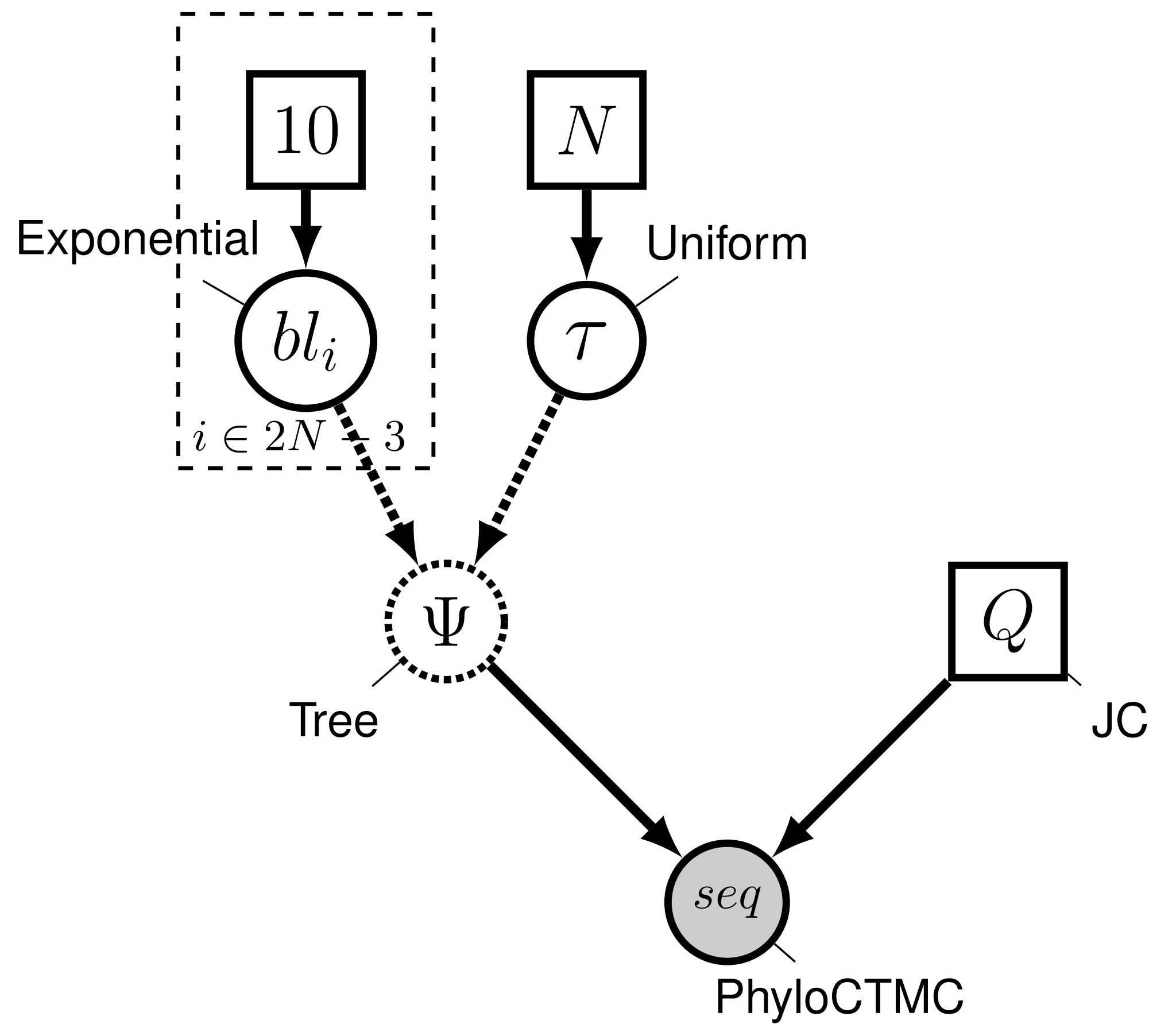
Bayesian tree inference

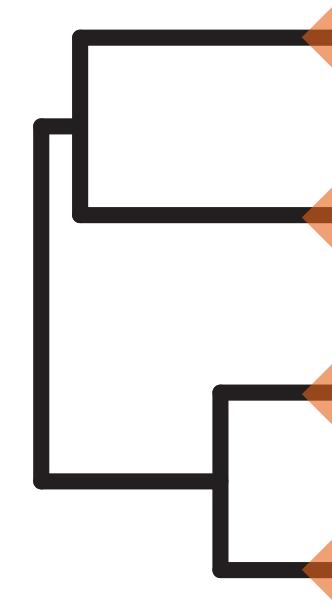
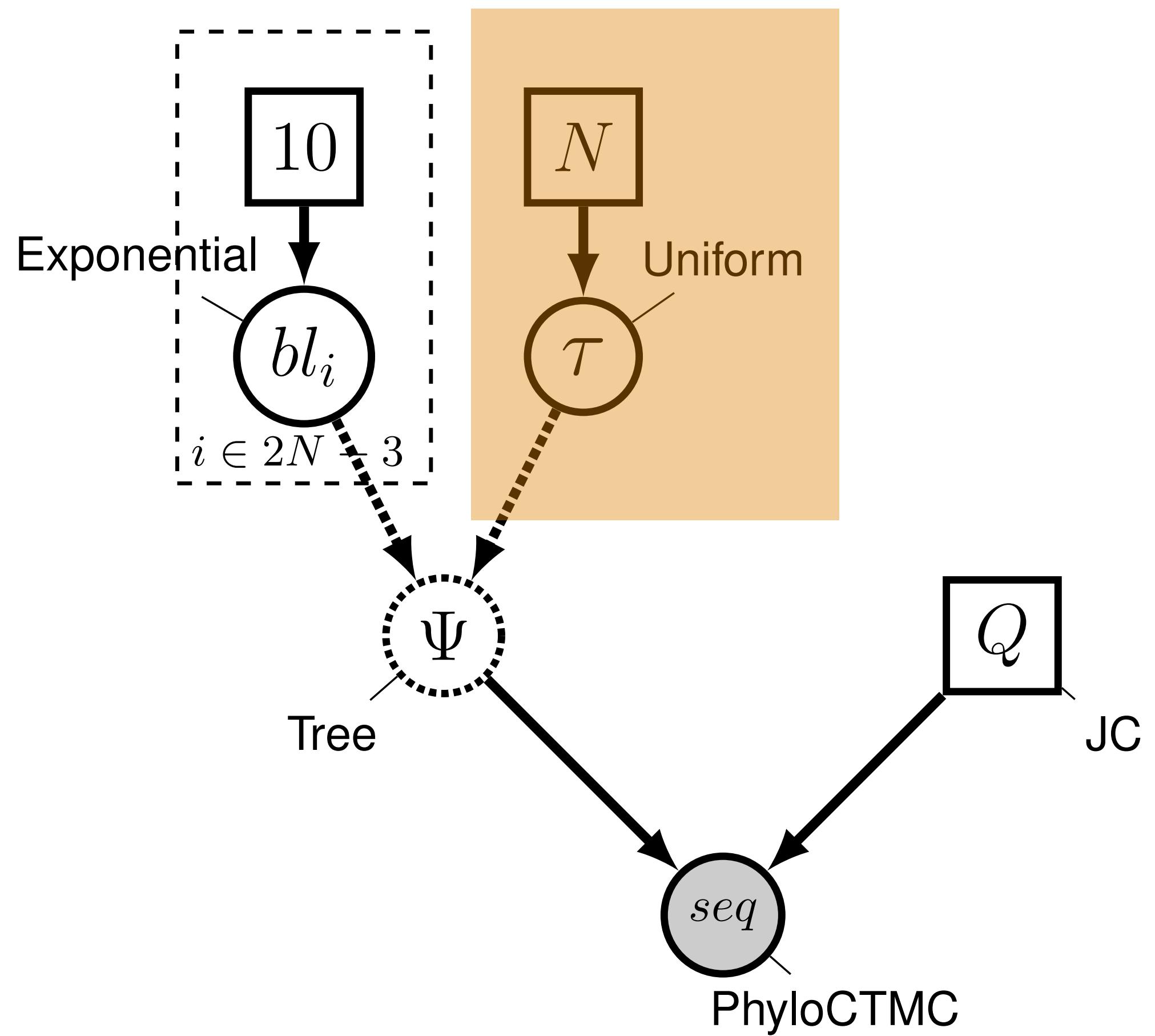
$$\text{posterior} = \frac{\text{likelihood} \cdot \text{priors}}{\text{marginal probability}}$$

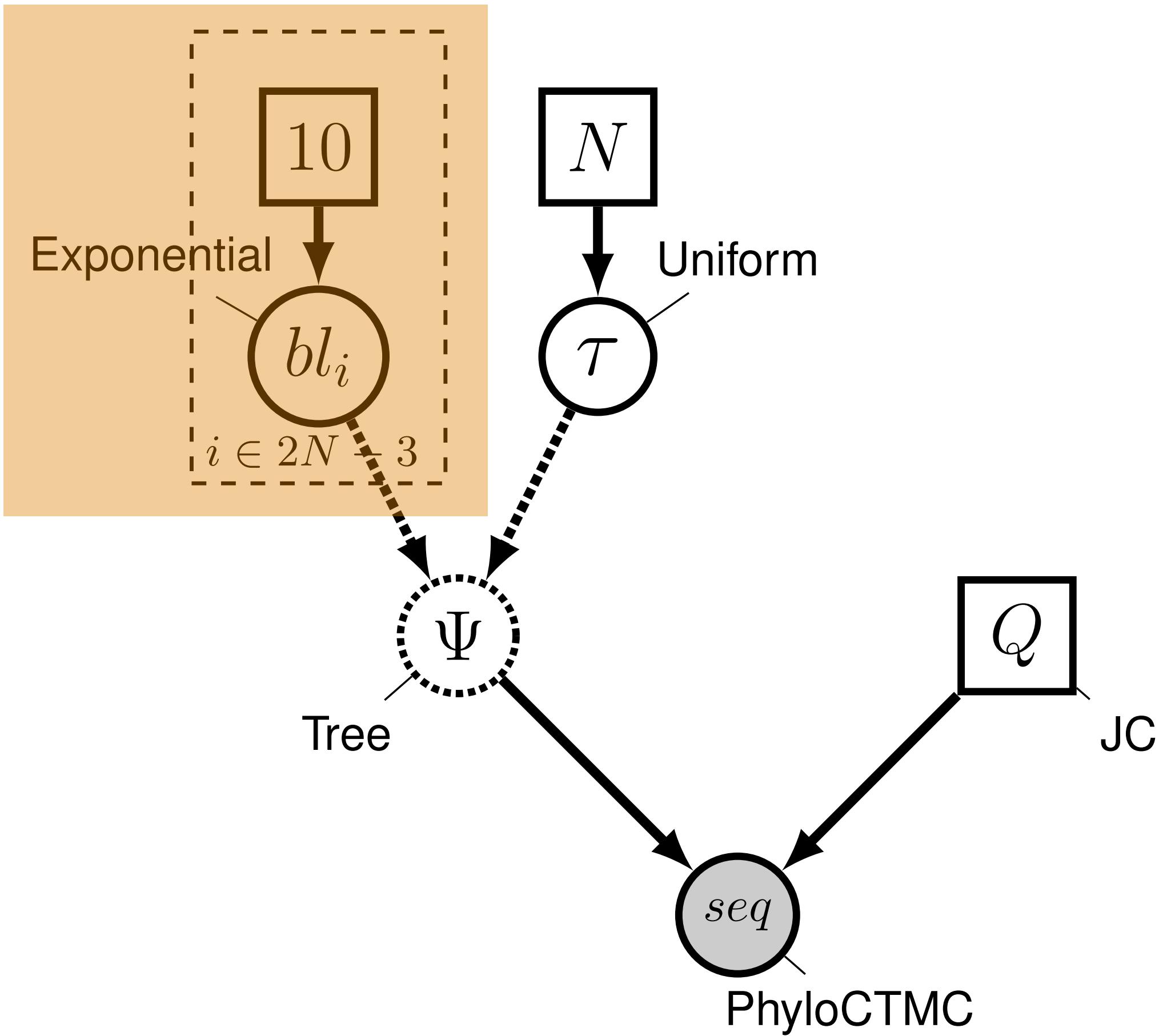
The equation illustrates the Bayesian formula for tree inference. The posterior probability is shown on the left, followed by an equals sign. To the right of the equals sign, the likelihood is represented by a tree with two leaf nodes (0 and 1) and three internal nodes. The priors are also represented by a tree with the same structure. The marginal probability is at the bottom, also represented by a tree with the same structure.

Diagram labels:

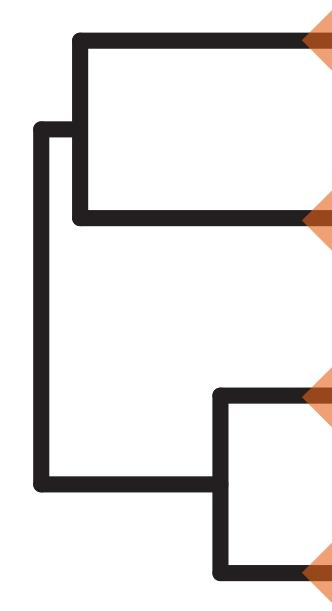
- posterior
- likelihood
- priors
- marginal probability

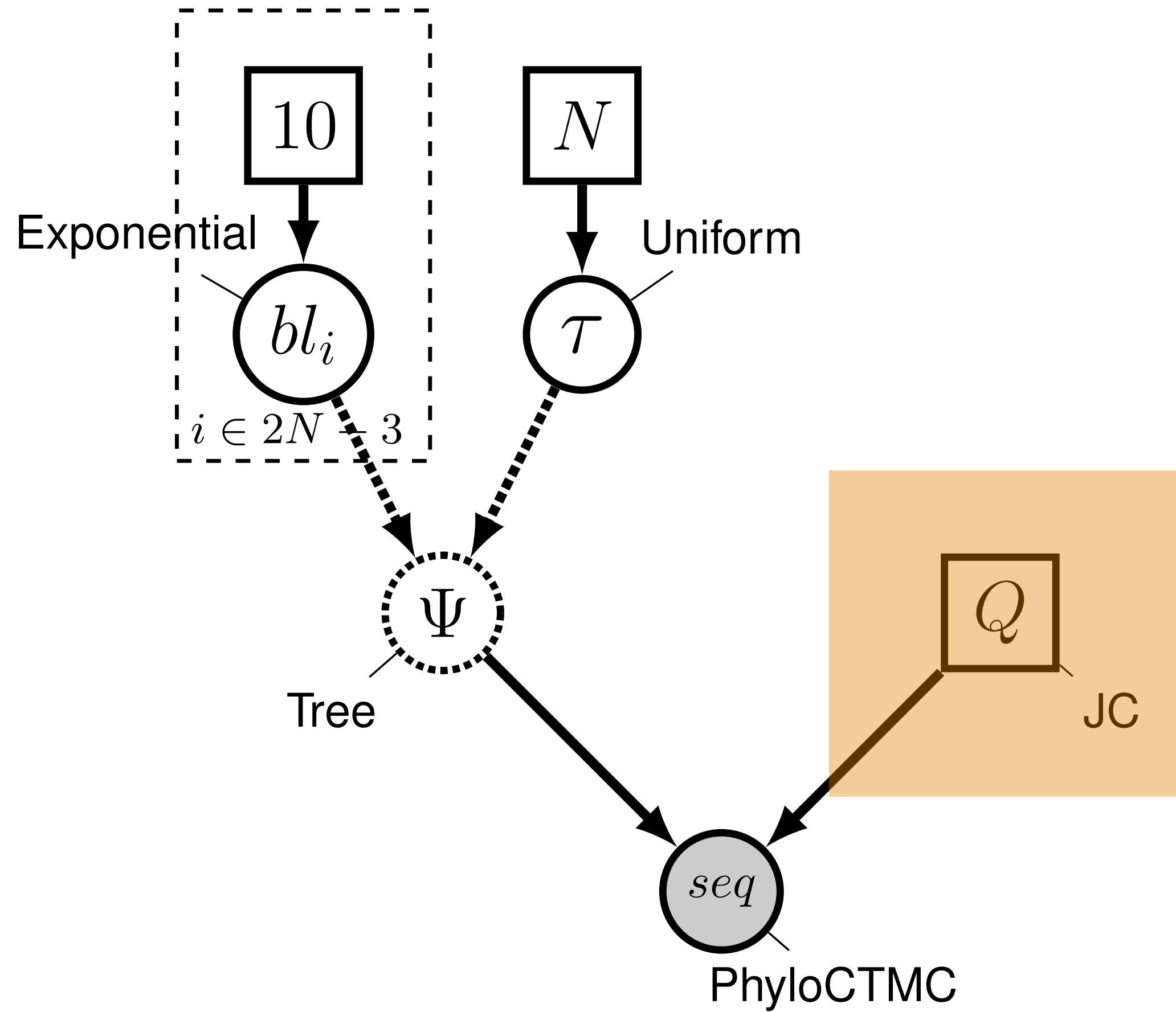




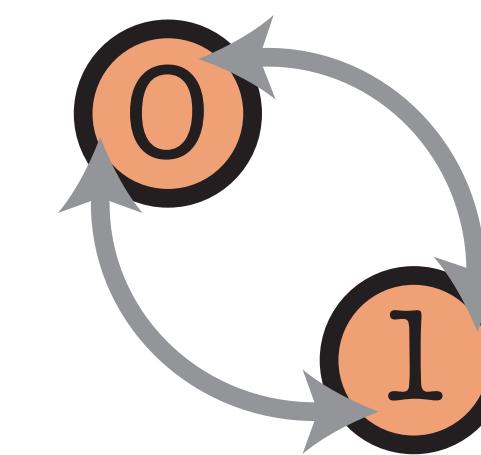


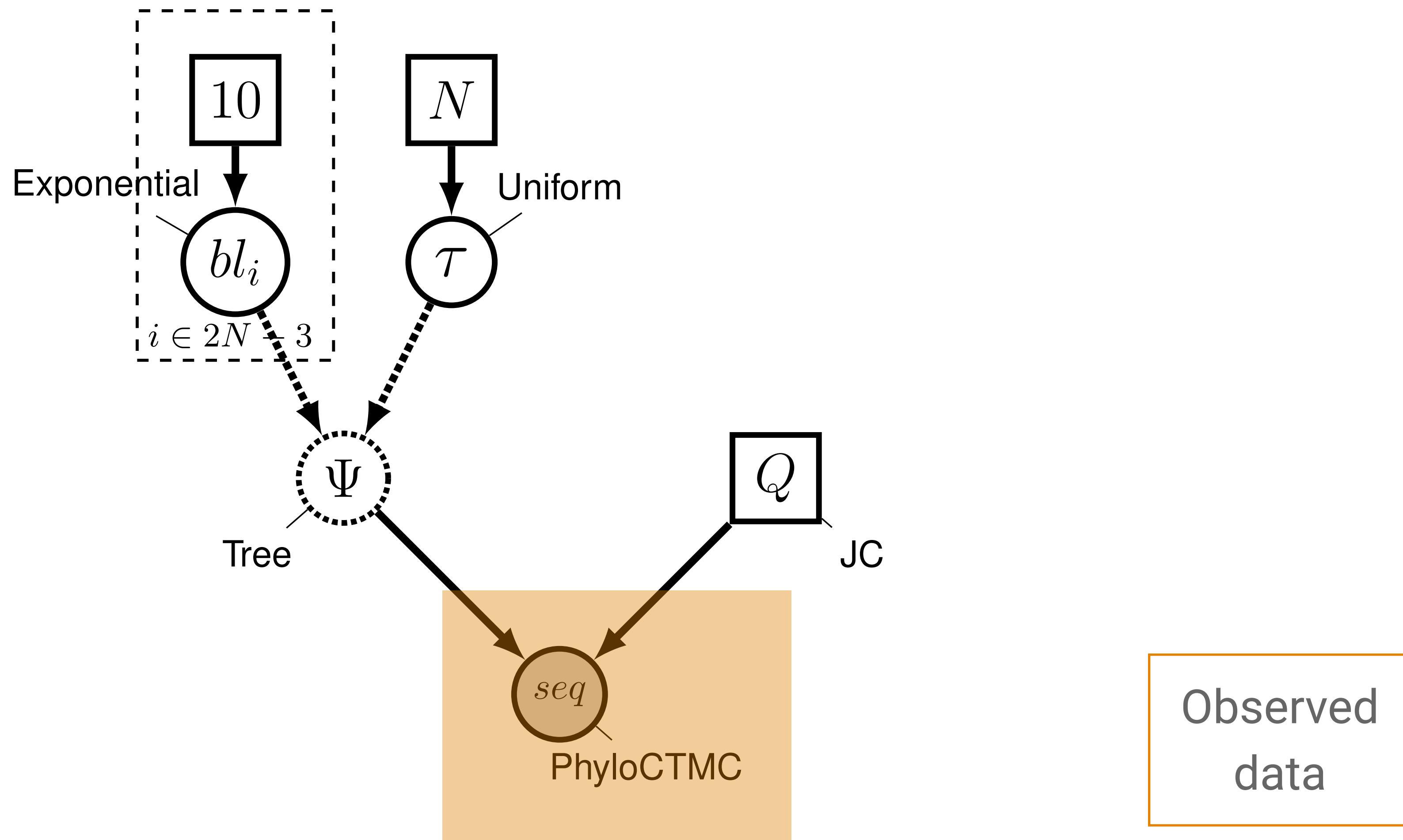
prior on the
branch lengths



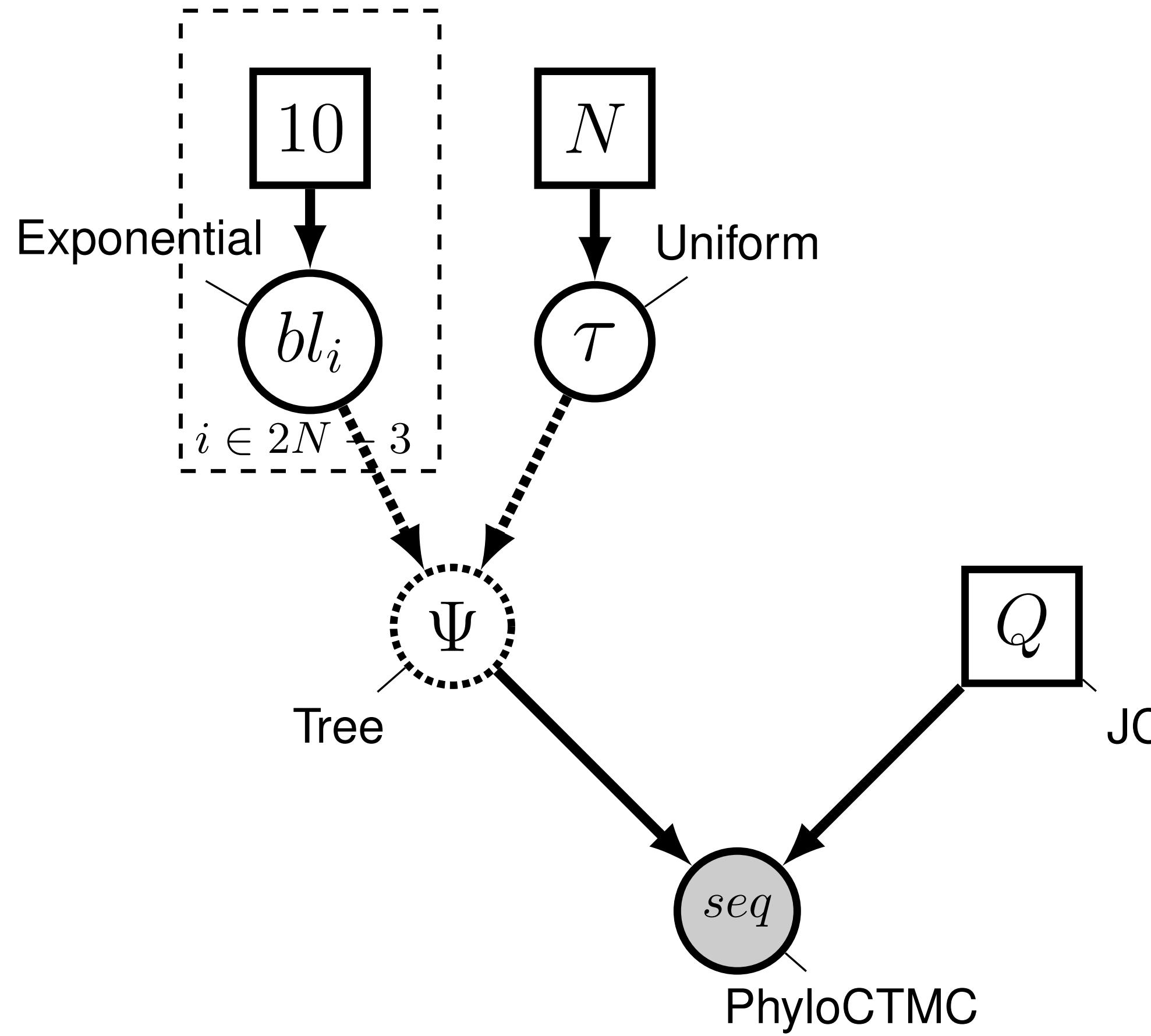


Substitution
model





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



```

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

Q_morpho <- fnJC(2)

phyMorpho ~ dnPhyloCTMC( tree=psi,
siteRates=rates_morpho, Q=Q_morpho,
type="Standard", coding="variable" )
phyMorpho.clamp( data )

```

Introduction to MCMC

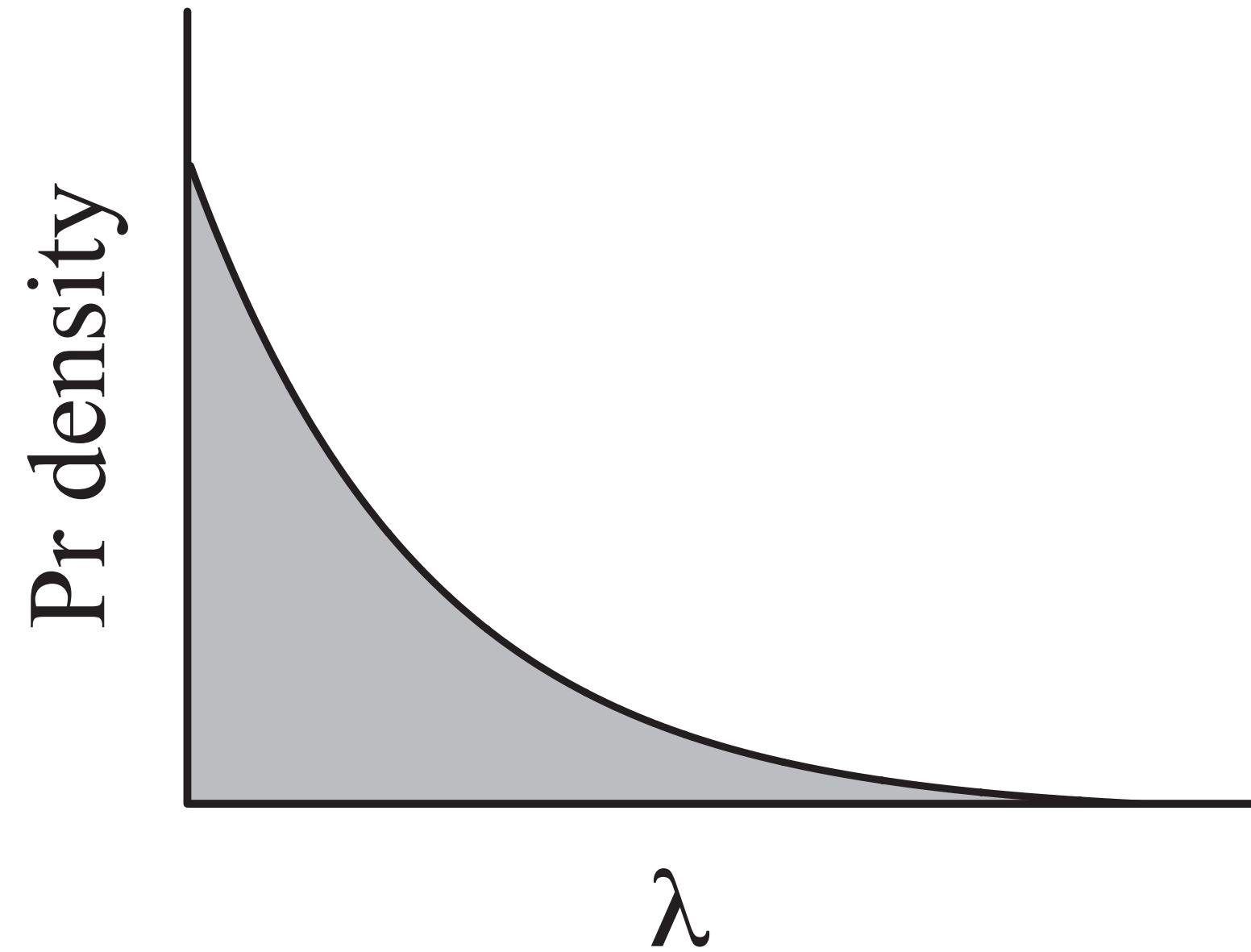
Probabilities vs probability densities

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



λ is drawn from an exponential distribution with mean δ

The x-axis represents the value of our parameter λ

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

The diagram illustrates the Bayesian formula for tree inference. On the left, the posterior probability $P(E \mid \text{0101...}, \text{1101...}, \text{0100...})$ is shown, where E is a tree with two root nodes labeled 0 and 1. The likelihood term $P(\text{0101...}, \text{1101...}, \text{0100...} \mid E, \text{prior})$ is split into two parts: the joint probability of the observed data given the tree and the prior probability of the tree itself. The prior term $P(E)$ is also shown, which is the joint probability of the tree structure given the observed data.

Bayesian tree inference

$$= \frac{P(\text{0101...} | \text{E} \circlearrowleft \text{0}) P(\text{E} \circlearrowleft \text{0})}{\int P(\text{0101...} | \text{E} \circlearrowleft \text{0}) P(\text{E} \circlearrowleft \text{0}) d\text{E}}$$

this part is incredibly difficult to calculate!

What is Markov chain Monte Carlo (MCMC)?

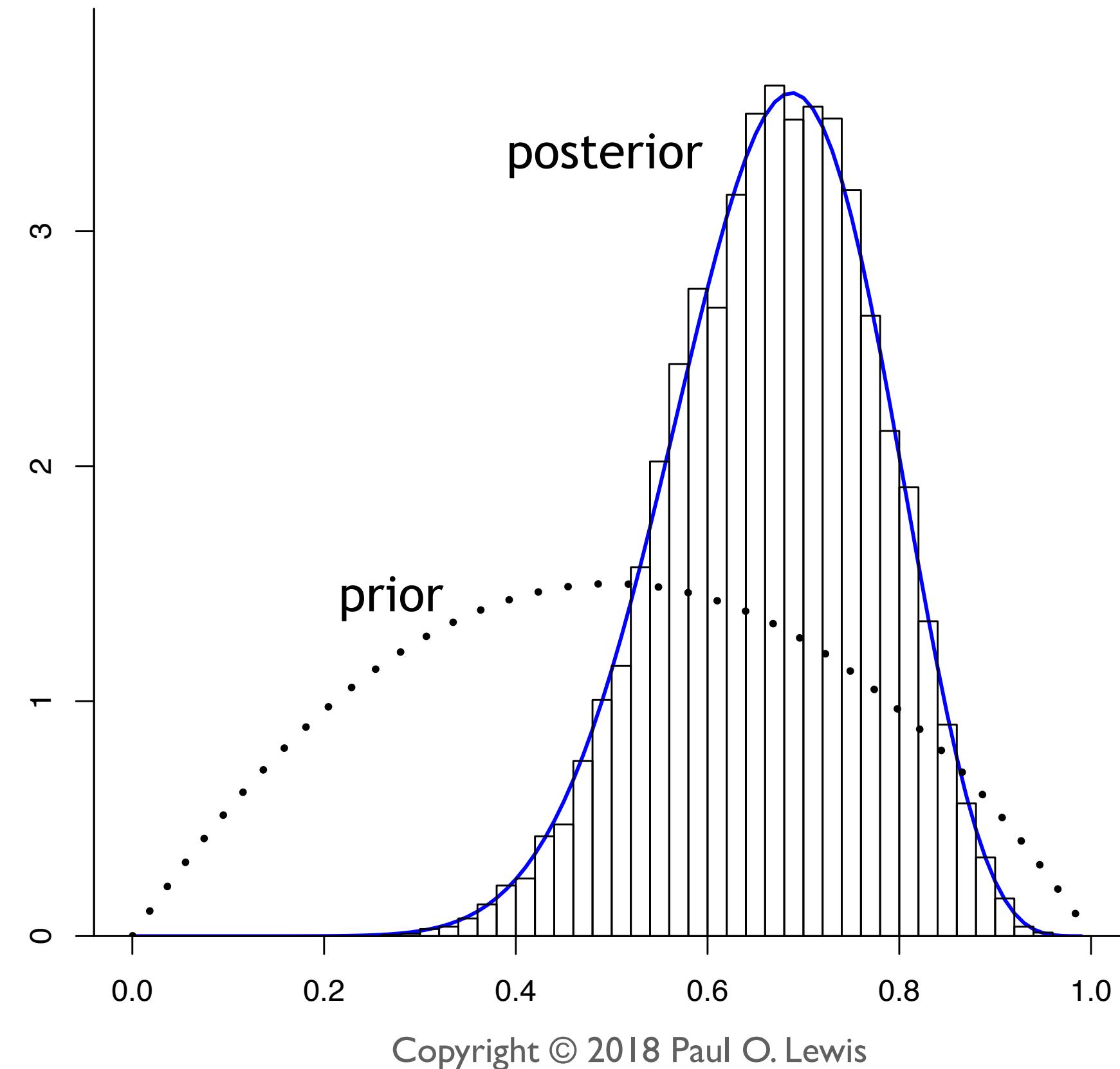
A group of algorithms for approximating the posterior distribution (also known as samplers)

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

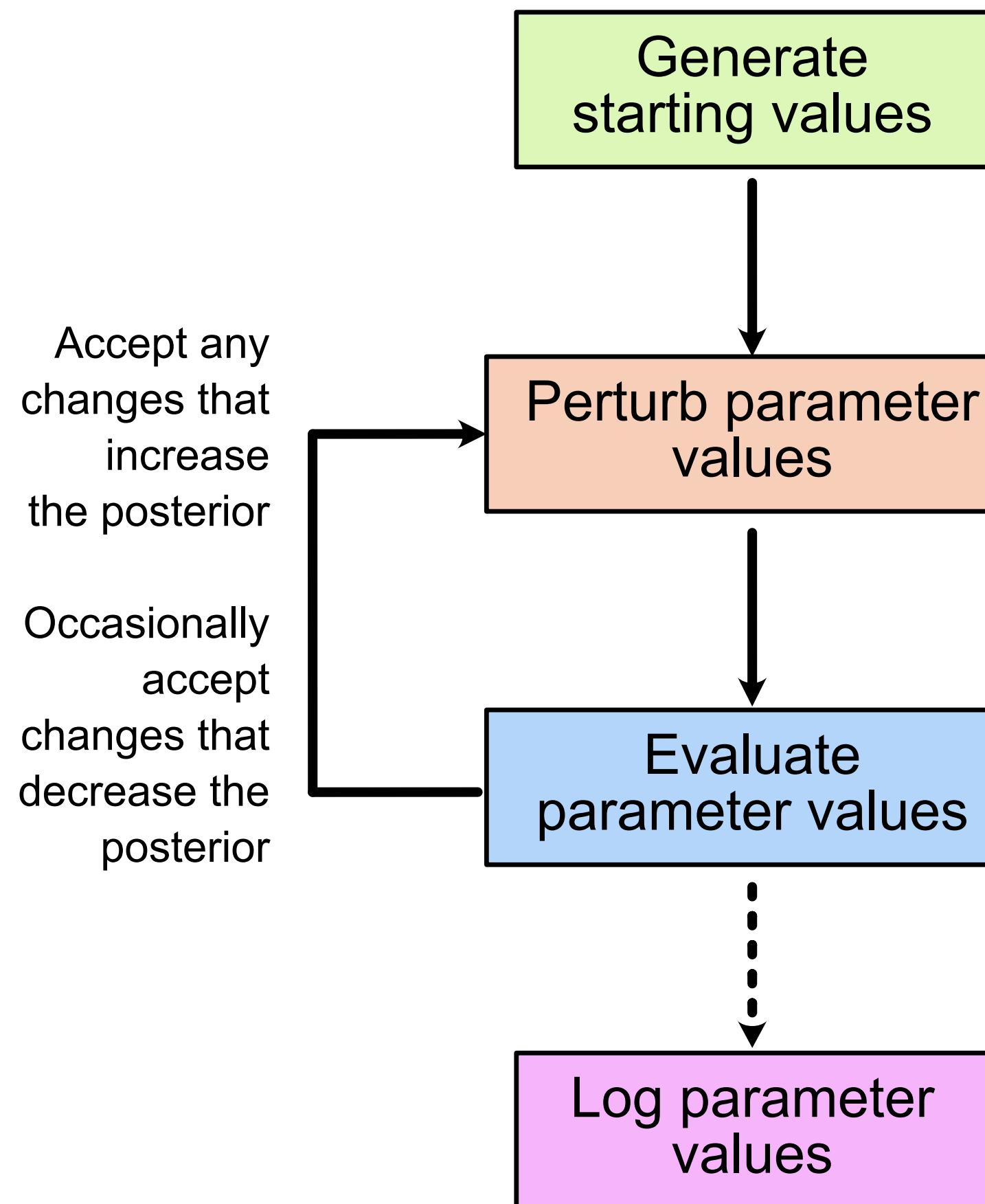
What is Markov chain Monte Carlo (MCMC)?



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

The Metropolis-Hastings algorithm

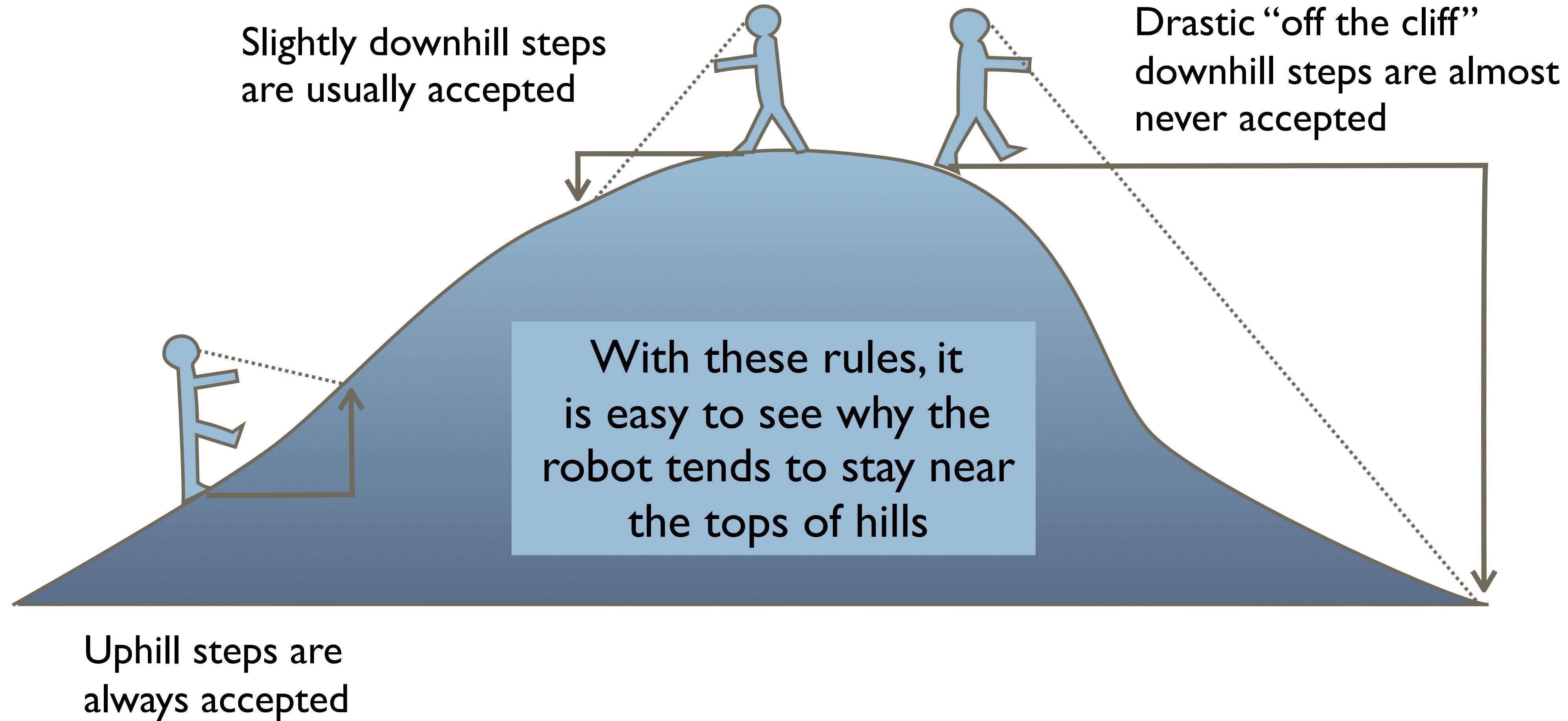
Flowchart

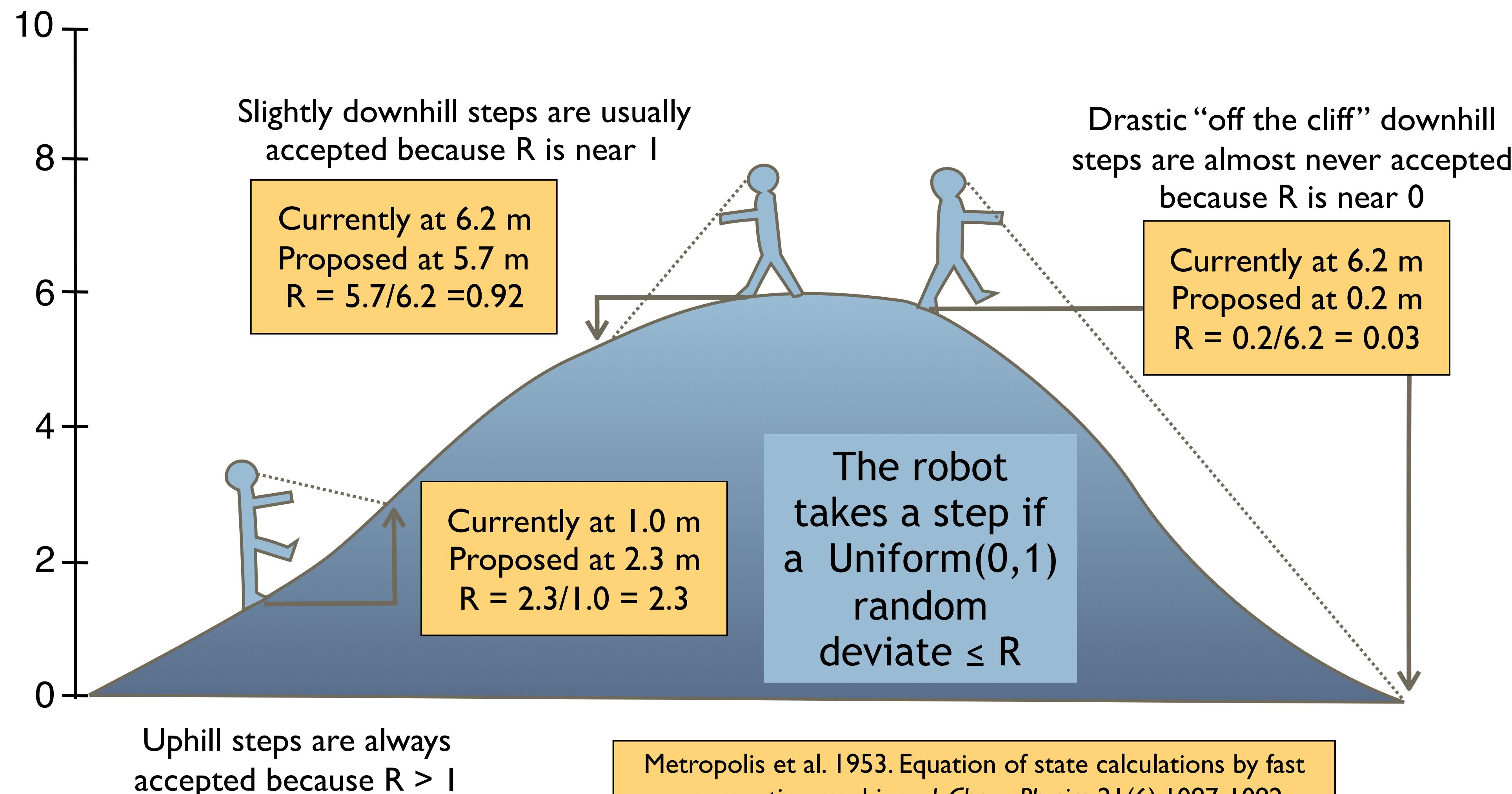


Pseudocode

```
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

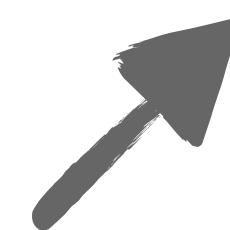
Likelihood
ratio

Prior
odds

Hastings ratio

new parameter
values

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



=

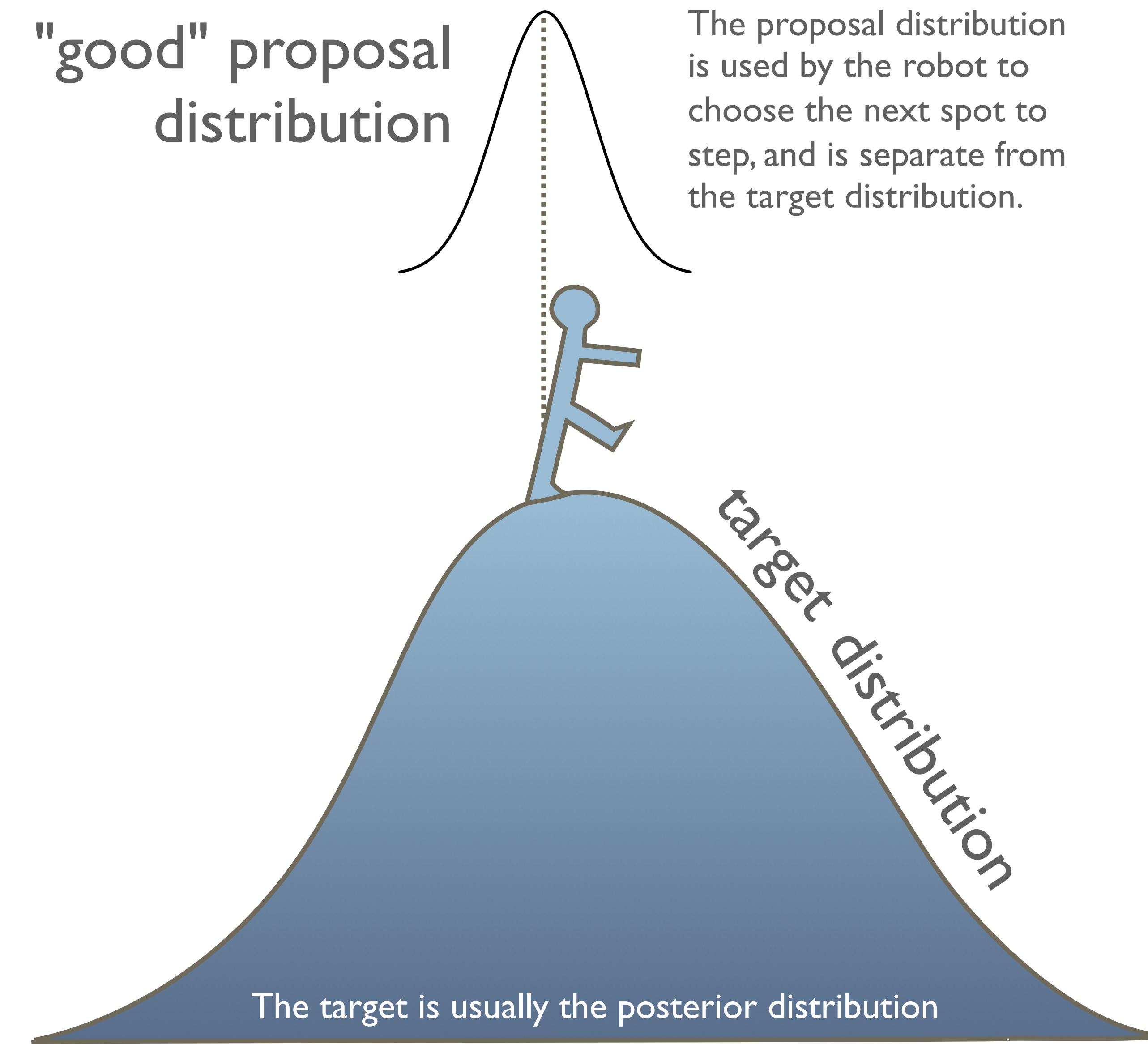
=

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

The marginal probability of the data cancels out

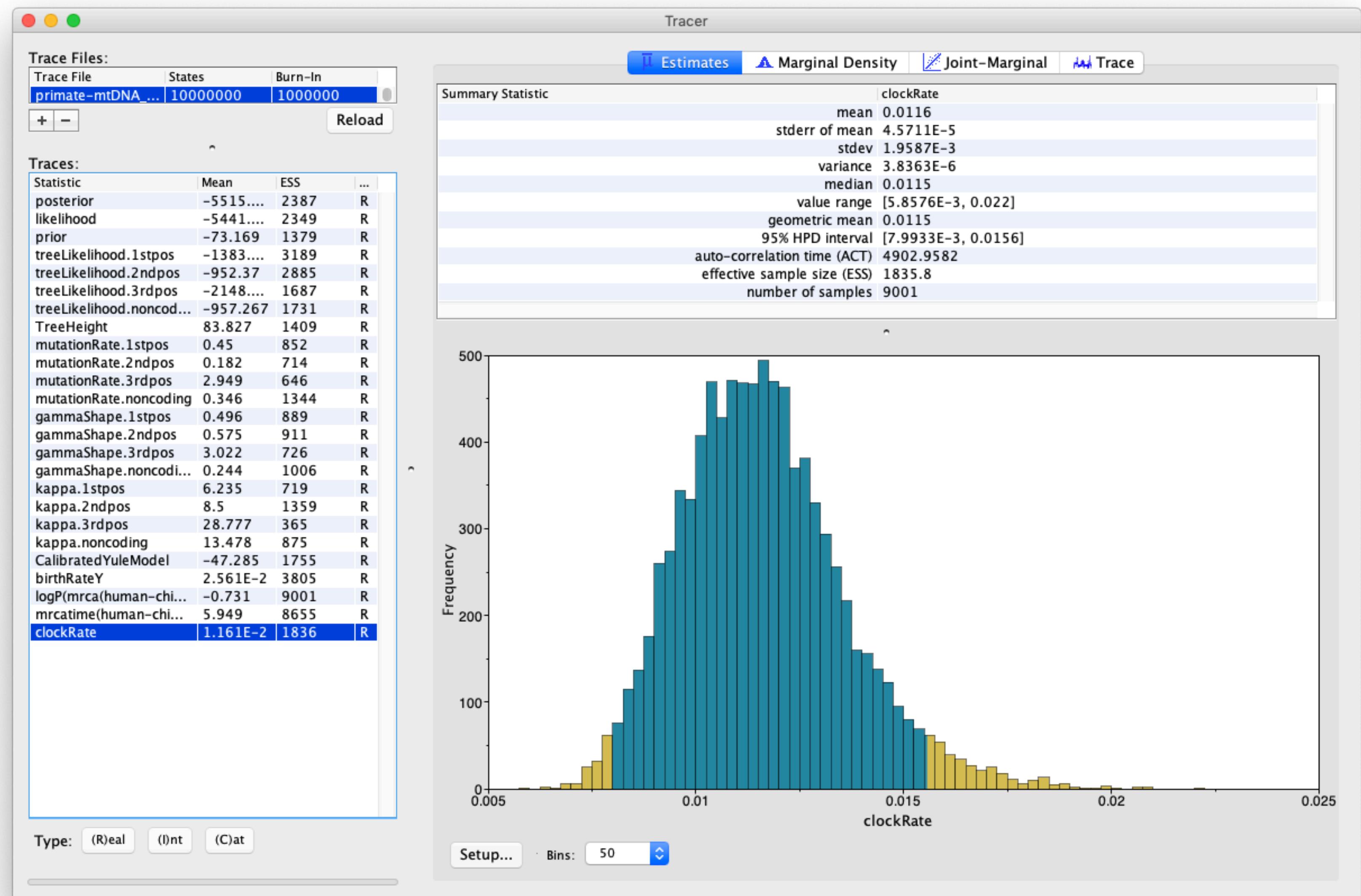
All we're left to calculate is the likelihood ratio and the prior odds ratio

Proposals



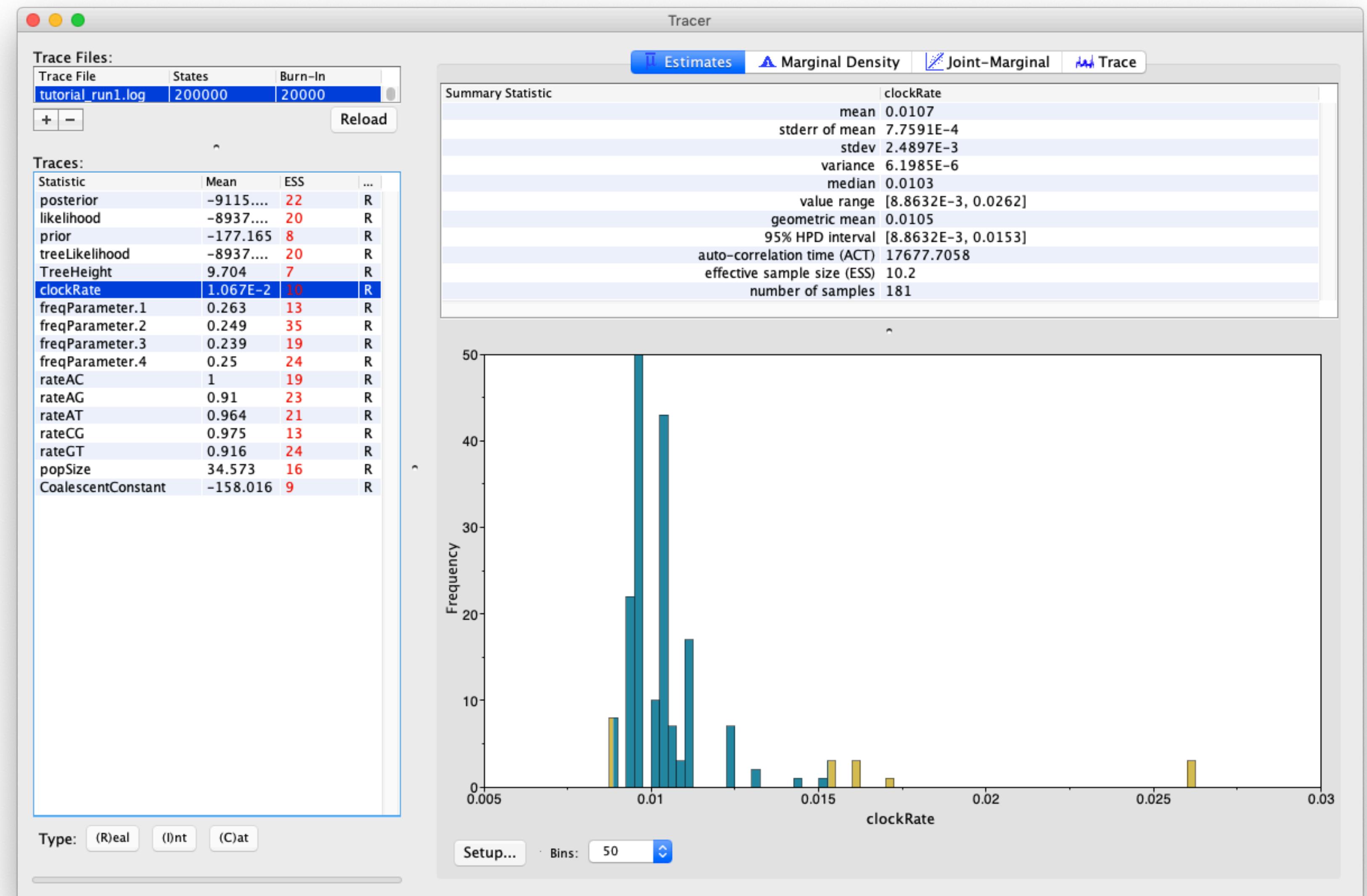
Summarising the posterior

Tracer is an amazing program for exploring MCMC output



Summarising the posterior

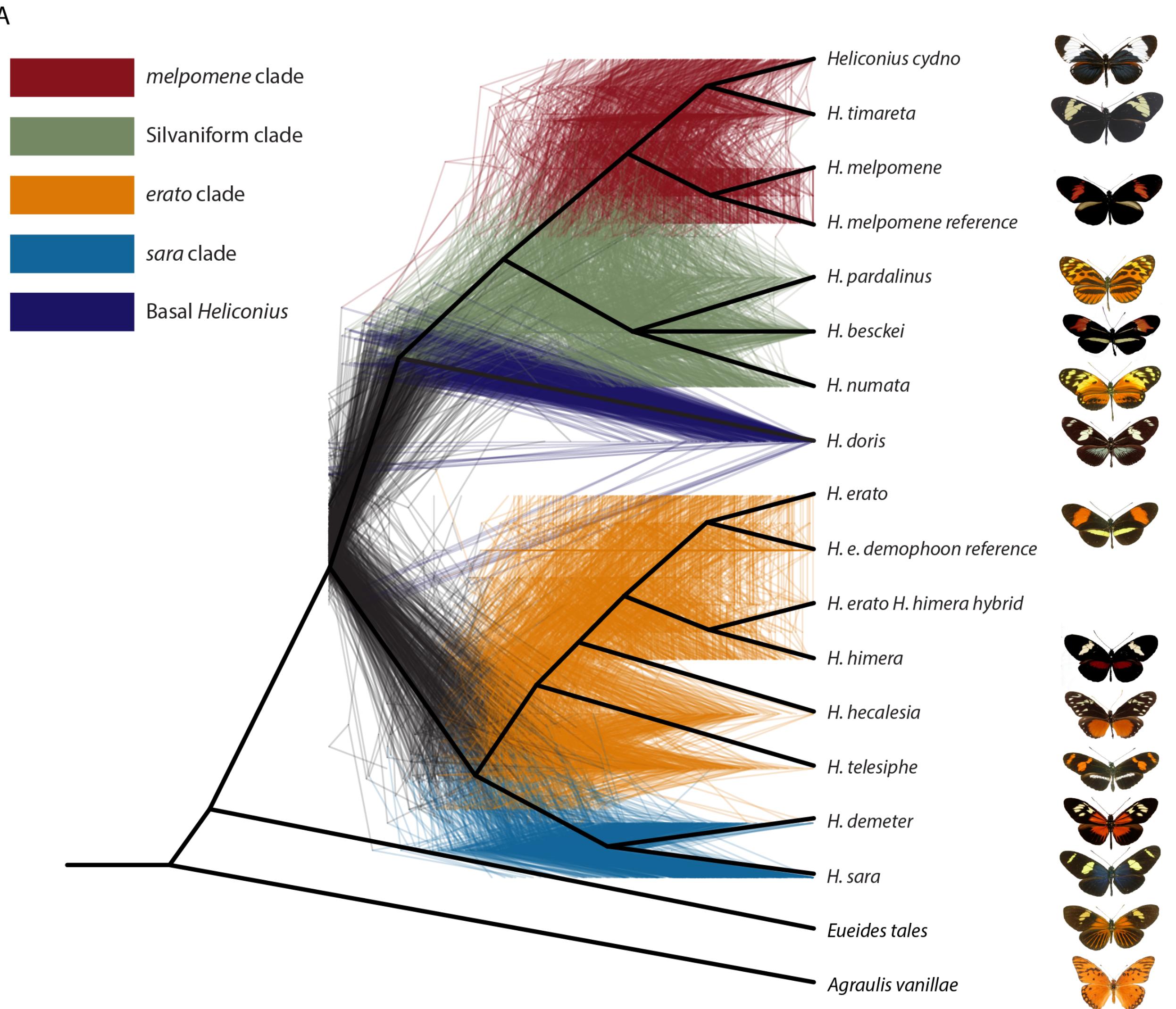
Tracer is an amazing program for exploring MCMC output



Summarising the posterior

Summarising trees is
much more challenging

Presenting a single
summary tree can be
misleading



Summarising the posterior

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

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

Exercise

Homework



[Phylogenetics primer part 2: tree likelihood and rate heterogeneity](#)

Paul Lewis

[Phylogenetics primer part 3a: introduction to Bayesian statistics](#)

Paul Lewis

→ See the question guide