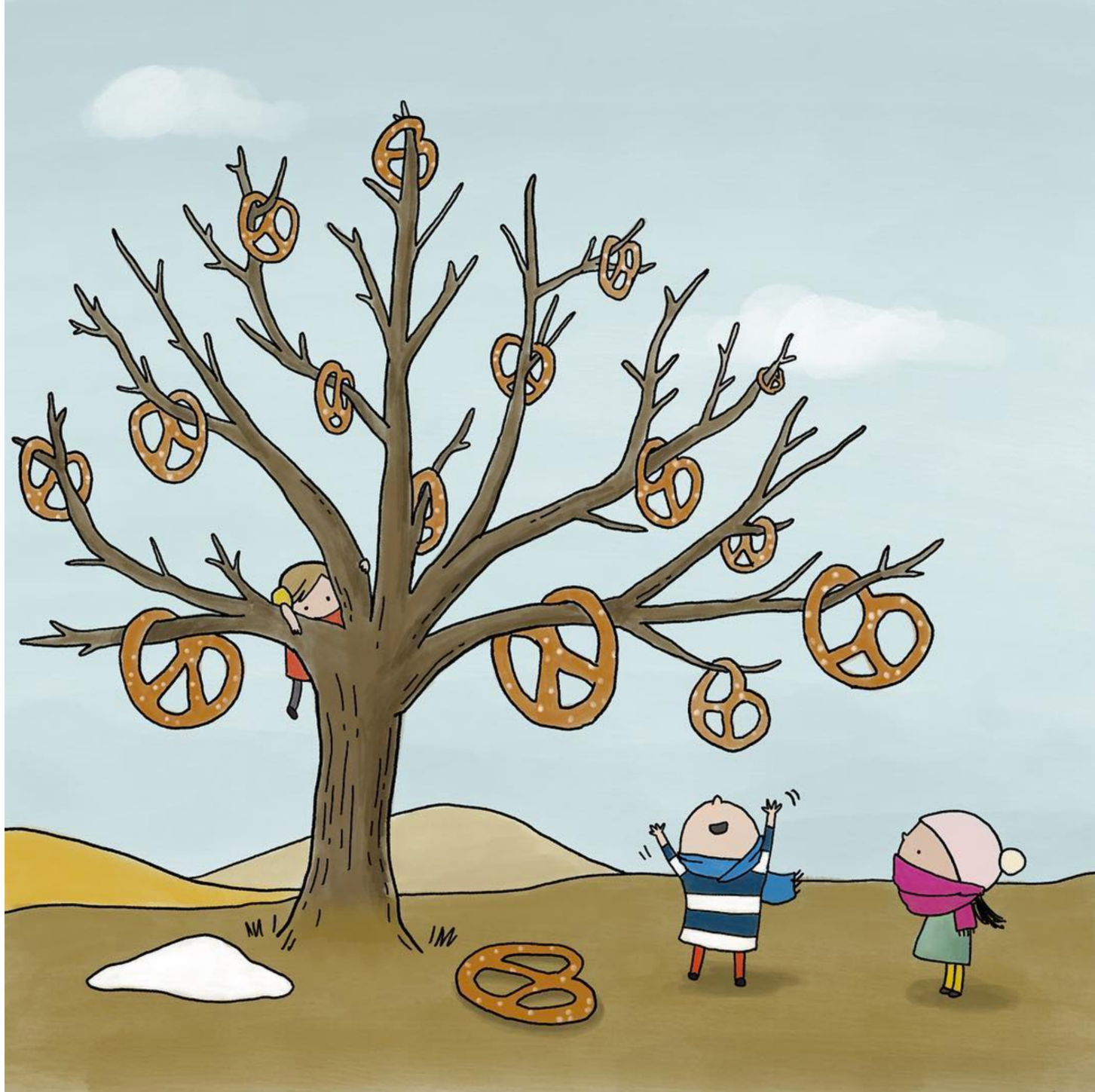
Phylogenetics workshop

Introduction to Bayesian tree inference

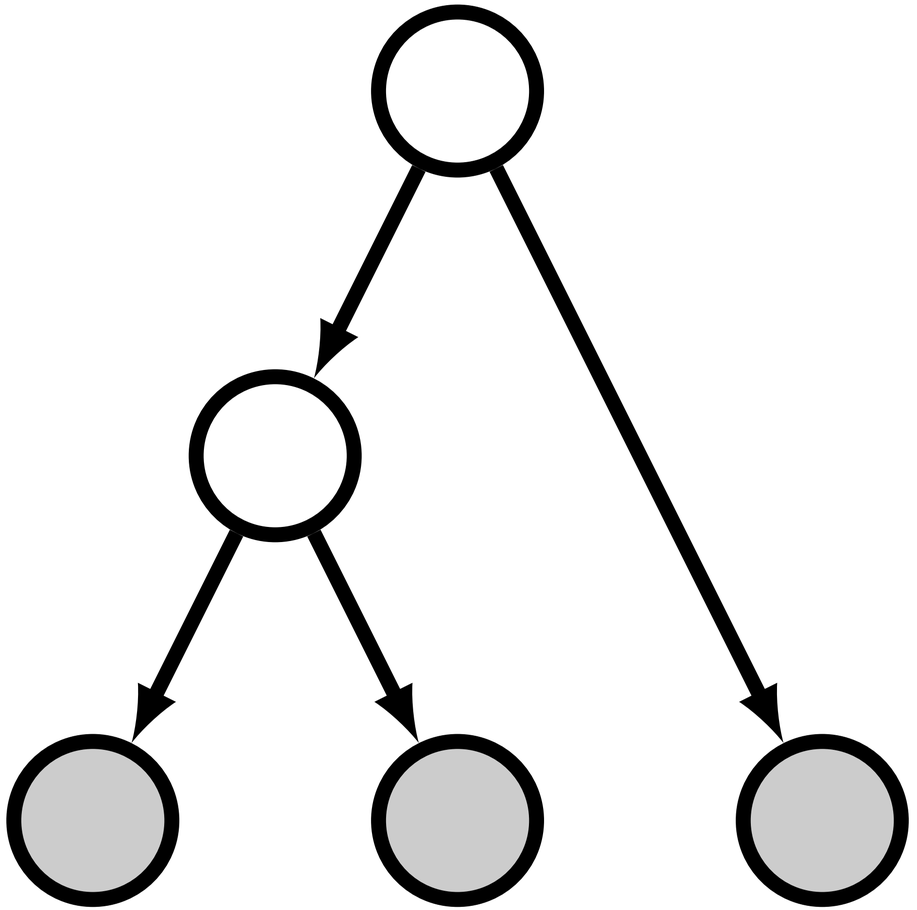
Rachel Warnock & co

10.12.24



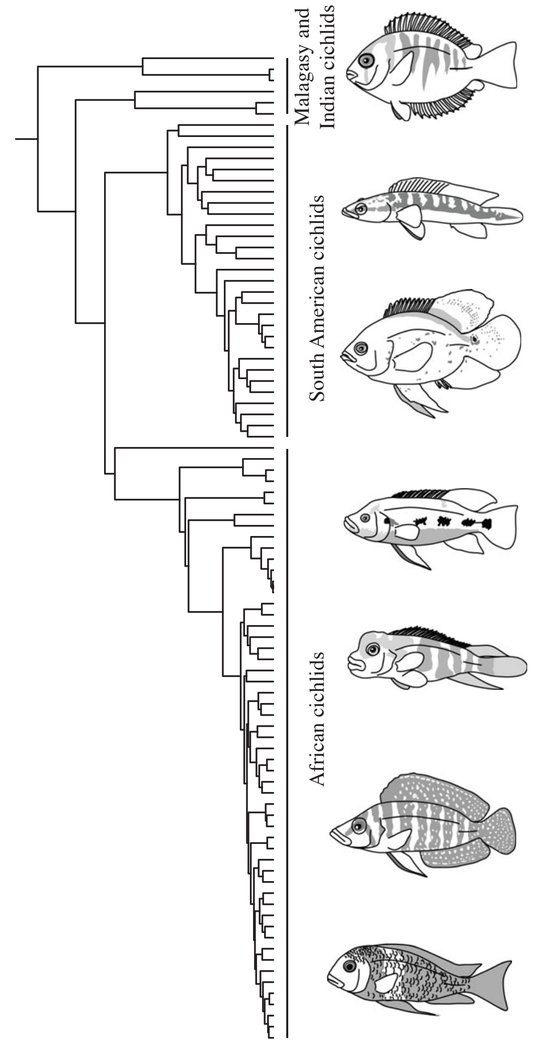
Part 1 objectives

- Bayesian inference
- MCMC
- RevBayes



What can we learn from trees?

- Evolutionary relationships



What can we learn from trees?

- Evolutionary relationships
- Timing of diversification events
- Geological context
- Rates of phenotypic evolution
- Diversification rates

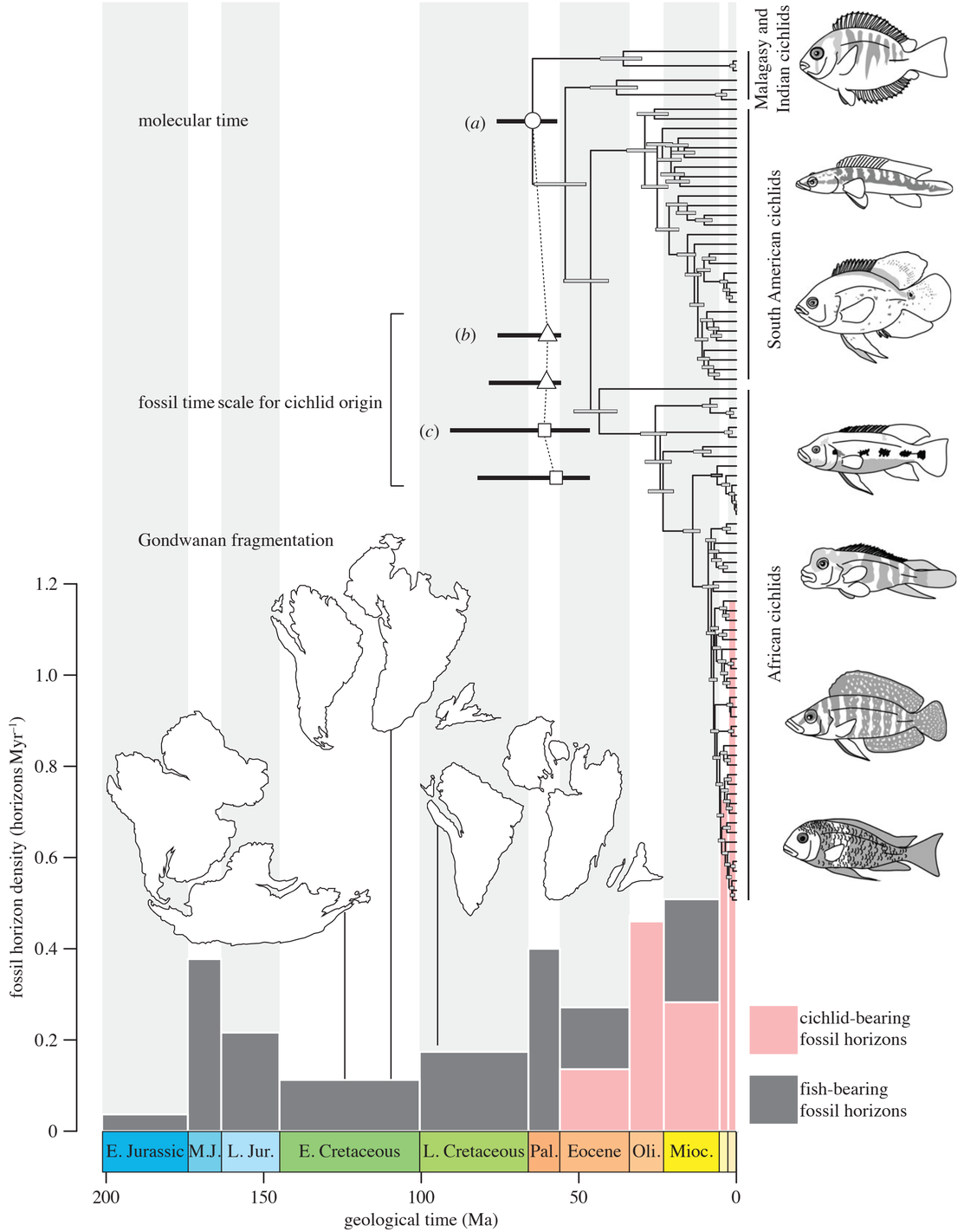


Image adapted from *Friedmann et al.* (2013)

How do we find the ‘best’ tree?

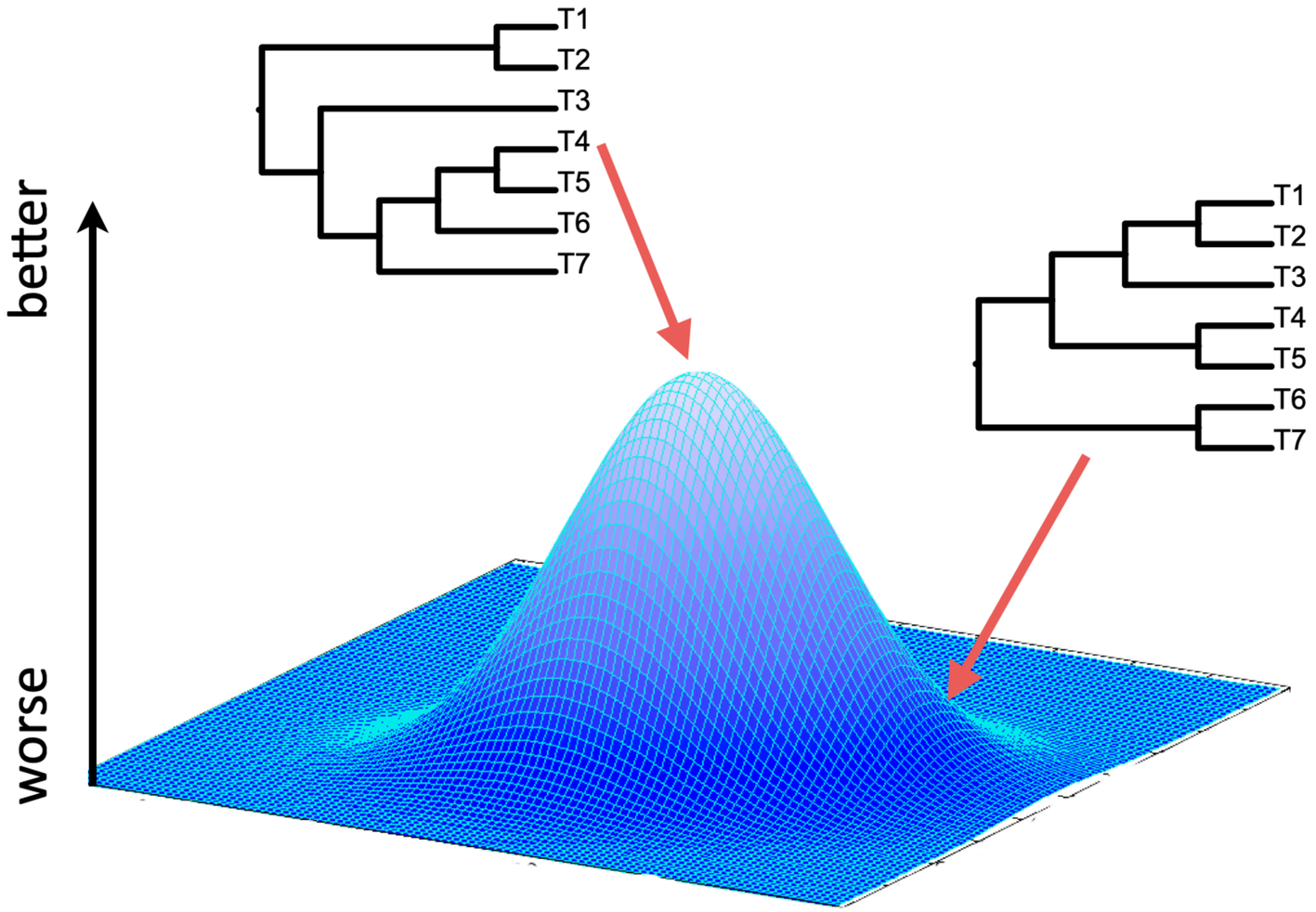


Image source Tracy Heath

It depends how you measure ‘best’

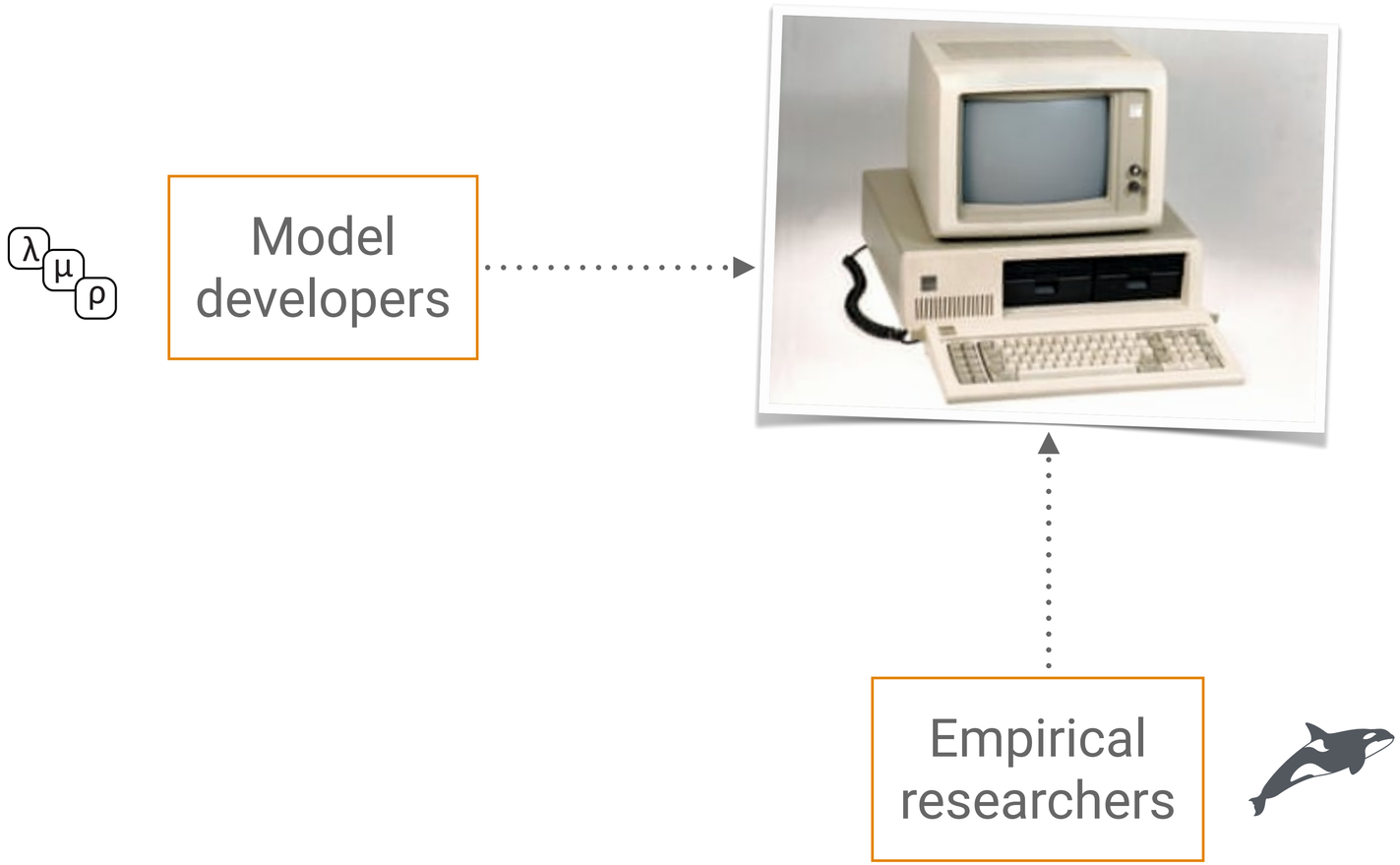
Method	Criterion (tree score)
Maximum parsimony	Minimum number of changes
Maximum likelihood	Likelihood score (probability), optimised over branch lengths and model parameters
Bayesian inference	Posterior probability, integrating over branch lengths and model parameters

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

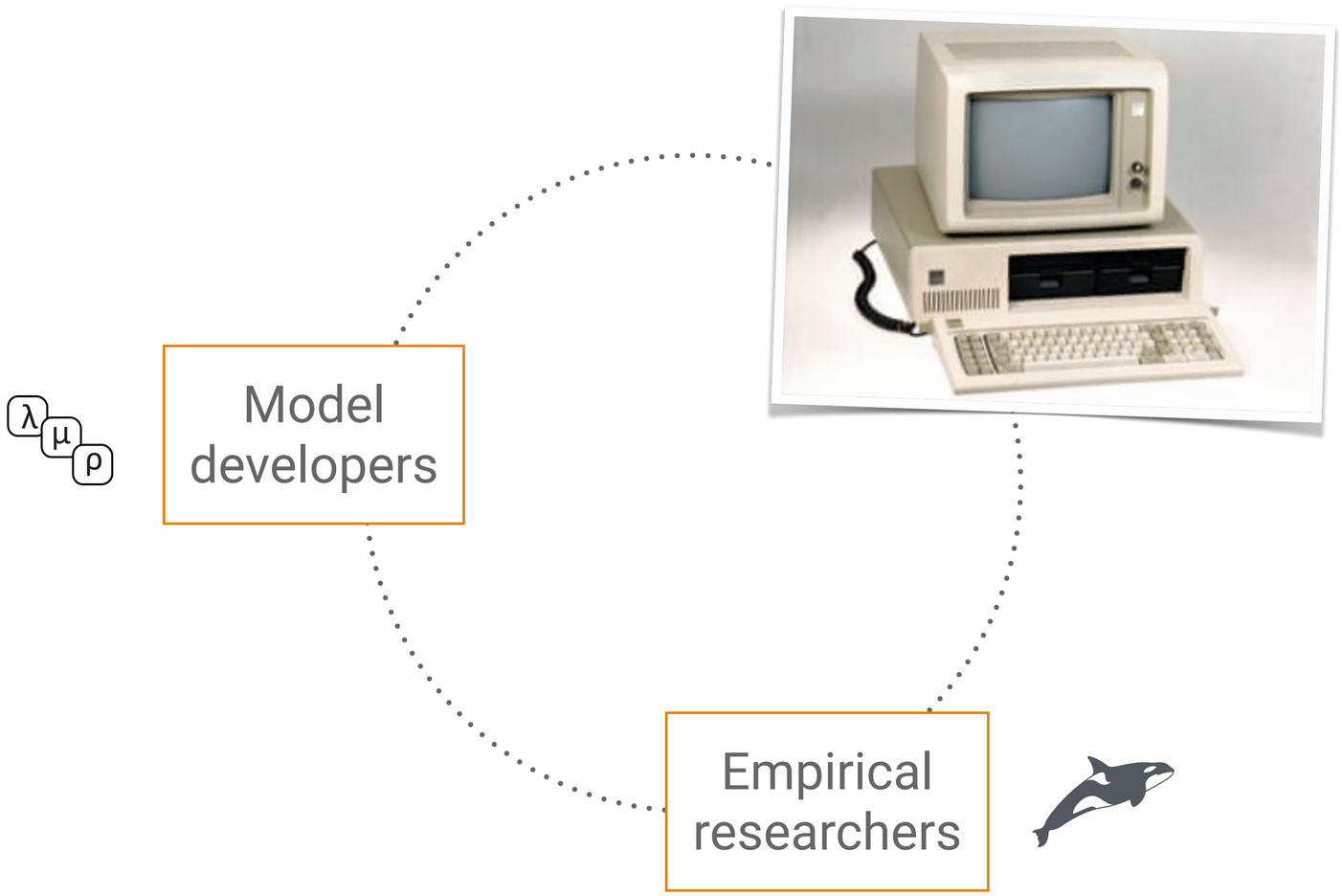
RevBayes

Phylogenetic inference — the old way



What we might call a **“black box”** approach

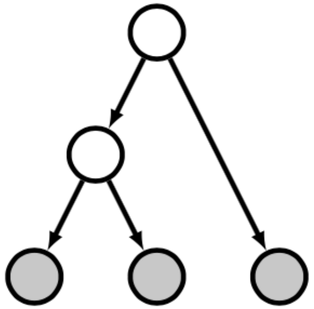
Phylogenetic inference — a better way?



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

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

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

Bayesian tree inference

Bayes' theorem

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

Bayes' theorem

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

Likelihood

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

Bayes' theorem

This represents our prior knowledge of the model parameters

Priors

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

Bayes' theorem

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

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

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

Bayes’ theorem — the gist

3 points to remember

We can include explicit models that describe the evolutionary and sampling processes

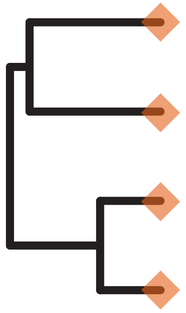
The priors can be used to incorporate existing knowledge

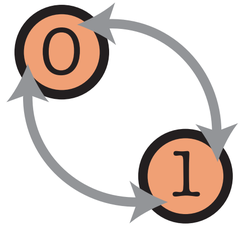
The outcome is not a point estimate, but a distribution of plausible parameter values and trees that reflect the uncertainty

Components used to infer trees

without considering time

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



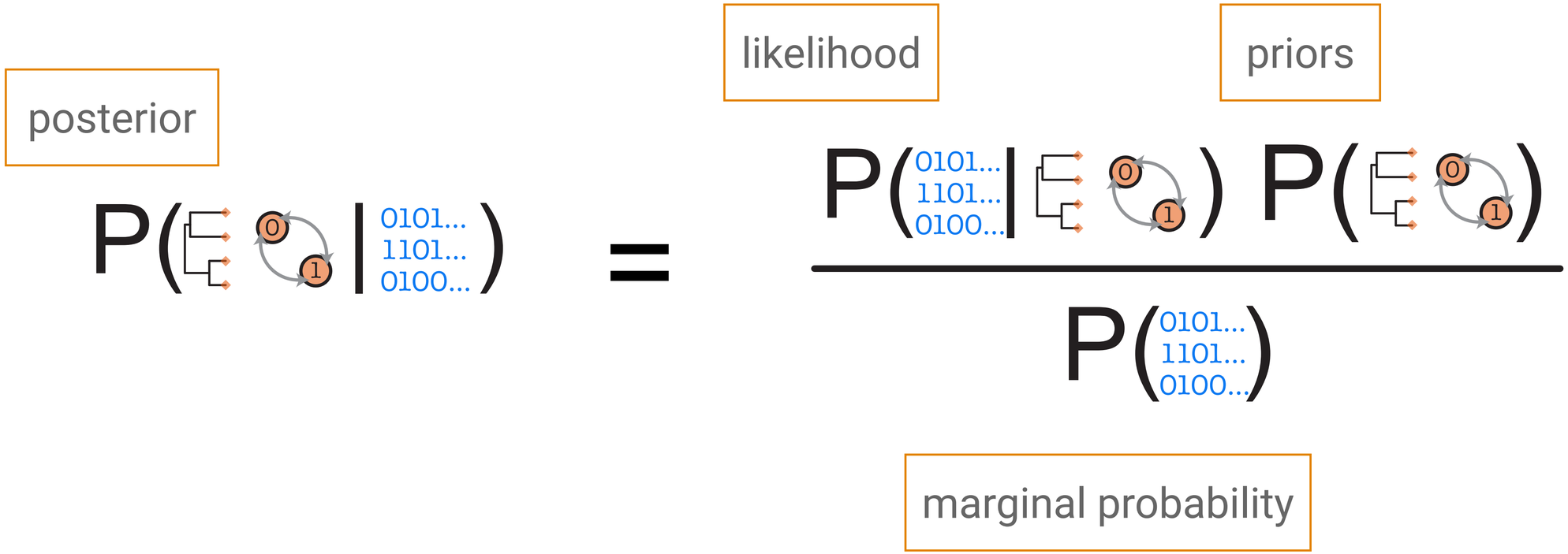


data
sequences or characters

tree model
topology and branch lengths

substitution model

Bayesian tree inference



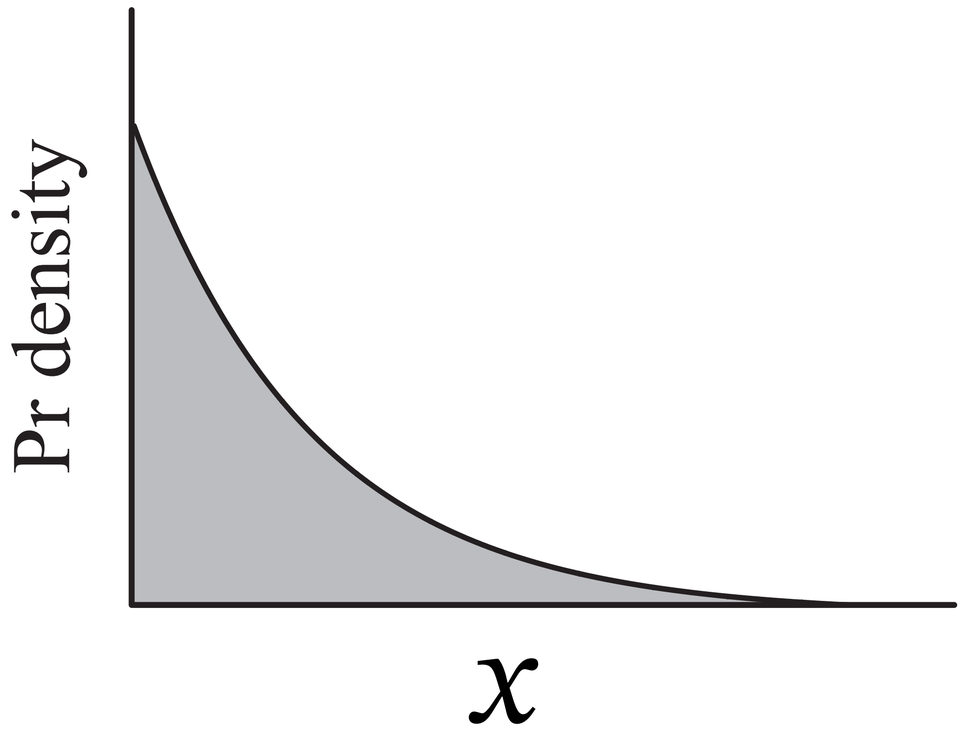
Introduction to MCMC

Probabilities vs probability densities

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

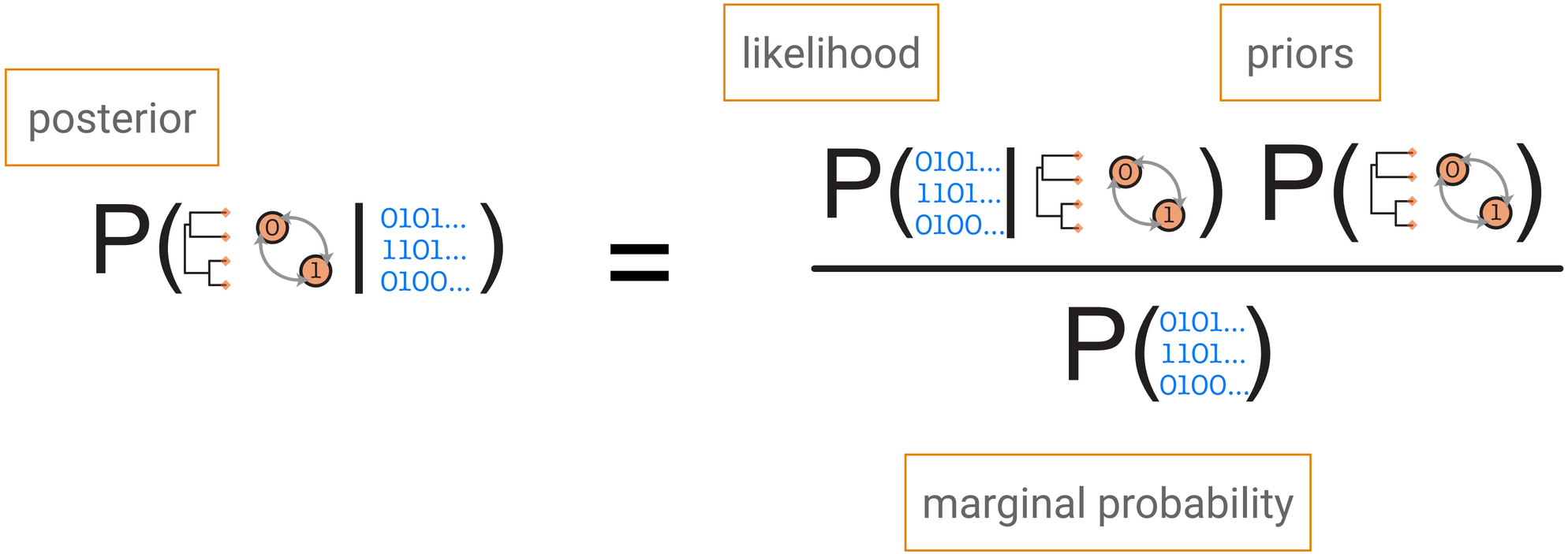
We're often dealing with a lot of uncertainty and typically work with probability densities

Probability densities introduce some complexity

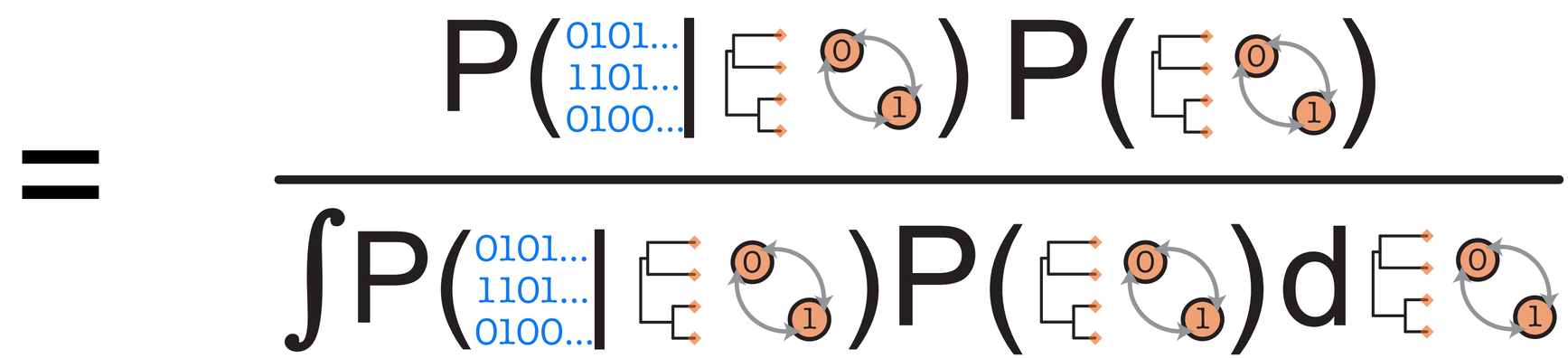


The distribution height reflects the relative probability of a given range of values

Bayesian tree inference



Bayesian tree inference



this part is incredibly difficult to calculate!

What is Markov chain Monte Carlo (MCMC)?

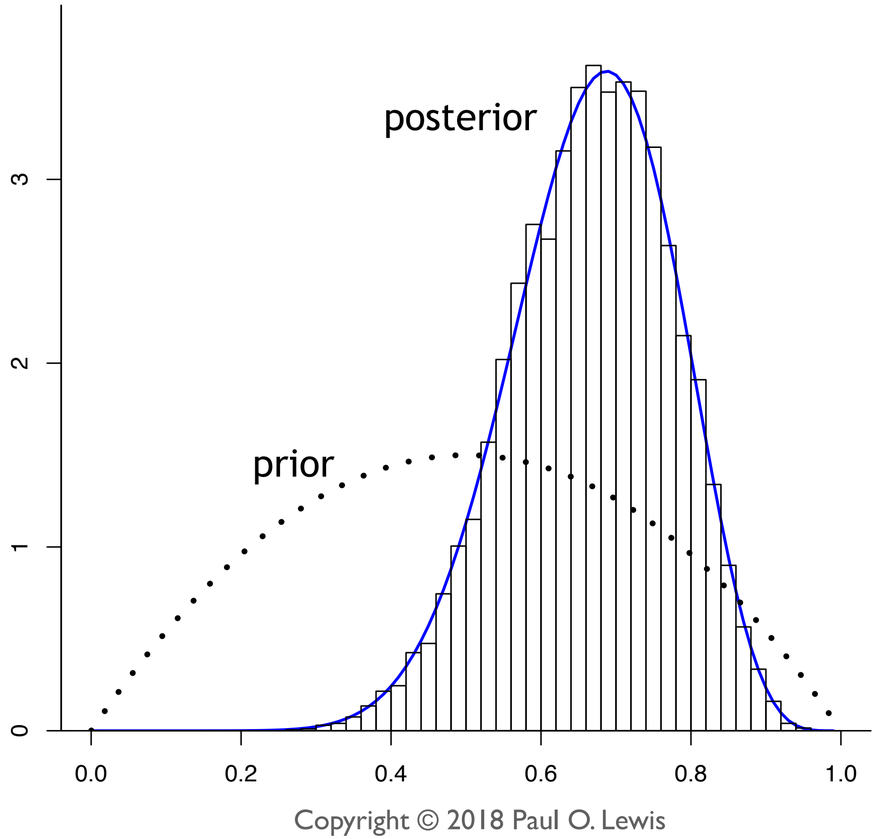
A group of algorithms for approximating the posterior distribution

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

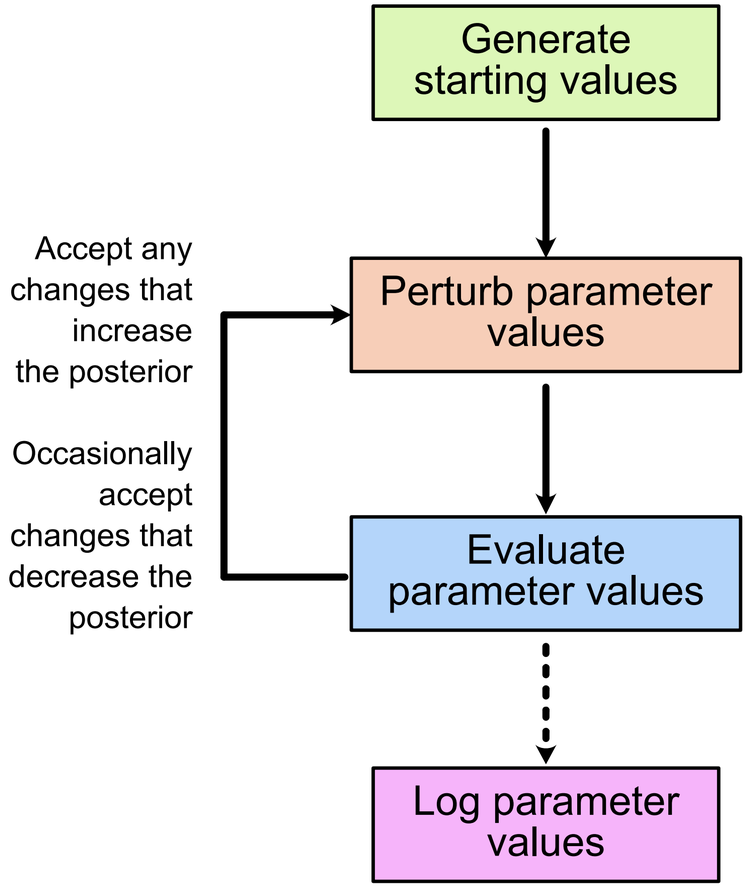
What is Markov chain Monte Carlo (MCMC)?



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

The Metropolis-Hastings algorithm

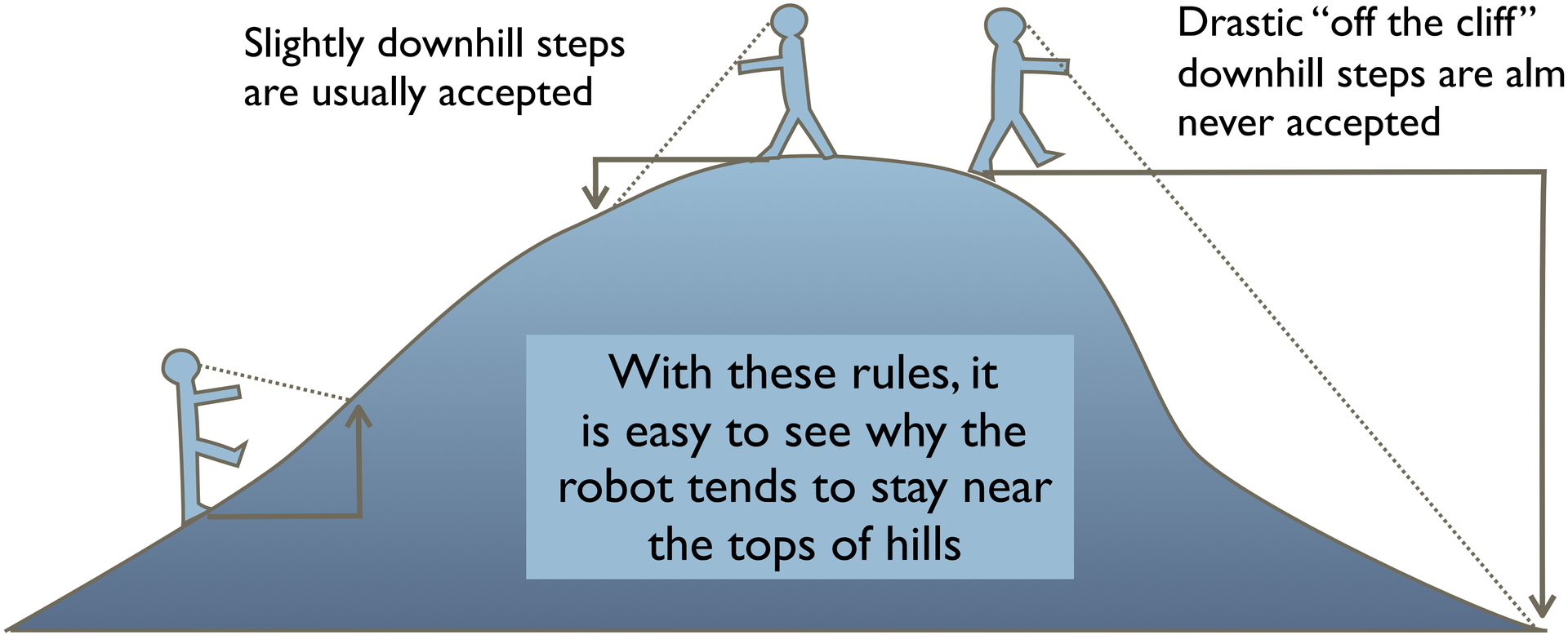
Flowchart

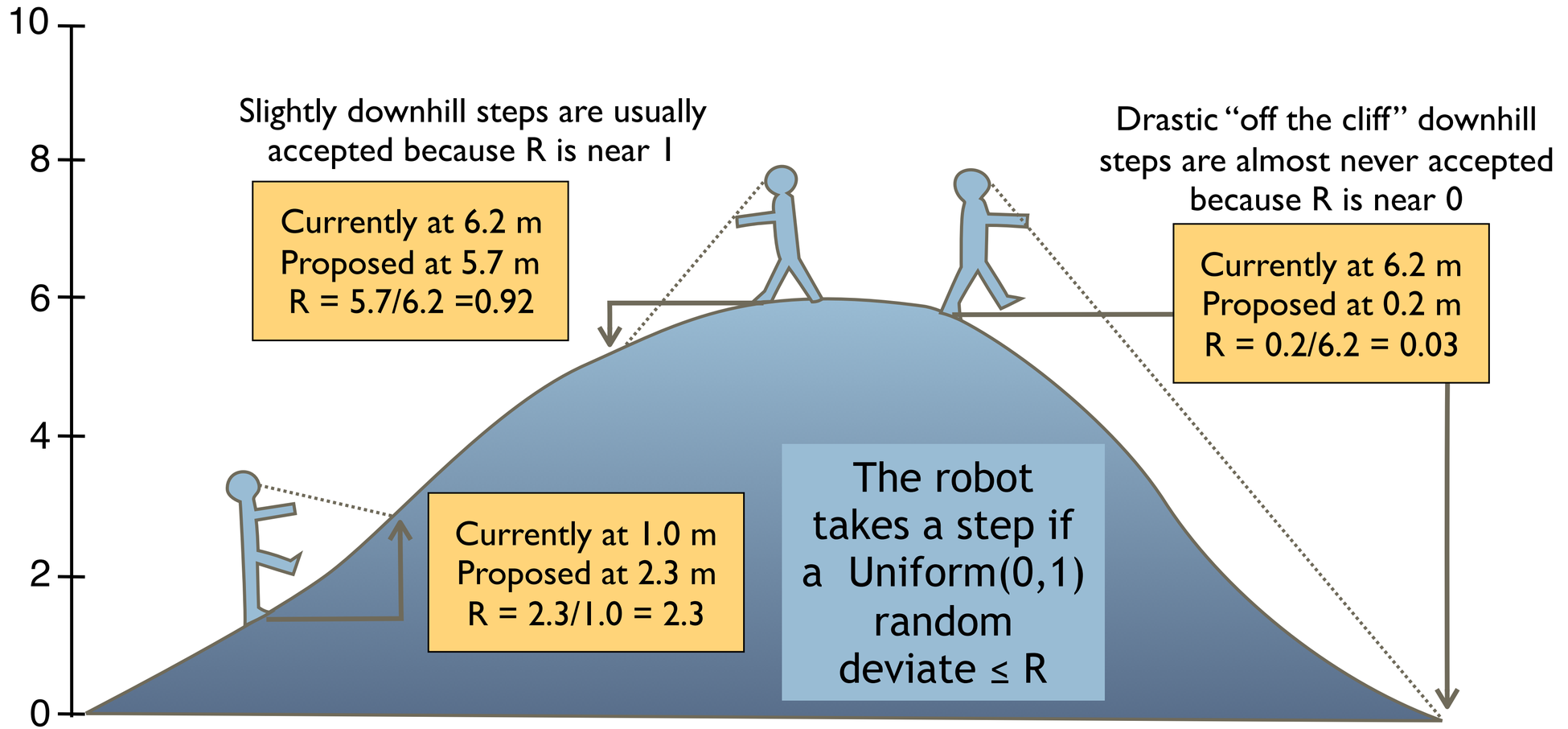


Pseudocode

```
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





Metropolis et al. 1953. Equation of state calculations by fast computing machines. *J. Chem. Physics* 21(6):1087-1092.

Hastings ratio

new parameter values

$$R = \frac{P(\text{tree}^*, \text{rates}^* \mid \text{0101... 1101... 0100...})}{P(\text{tree}, \text{rates} \mid \text{0101... 1101... 0100...})}$$

$$= \frac{P(\text{0101... 1101... 0100...} \mid \text{tree}^*, \text{rates}^*)\, P(\text{tree}^*, \text{rates}^*)}{P(\cancel{\text{0101... 1101... 0100...}})}$$

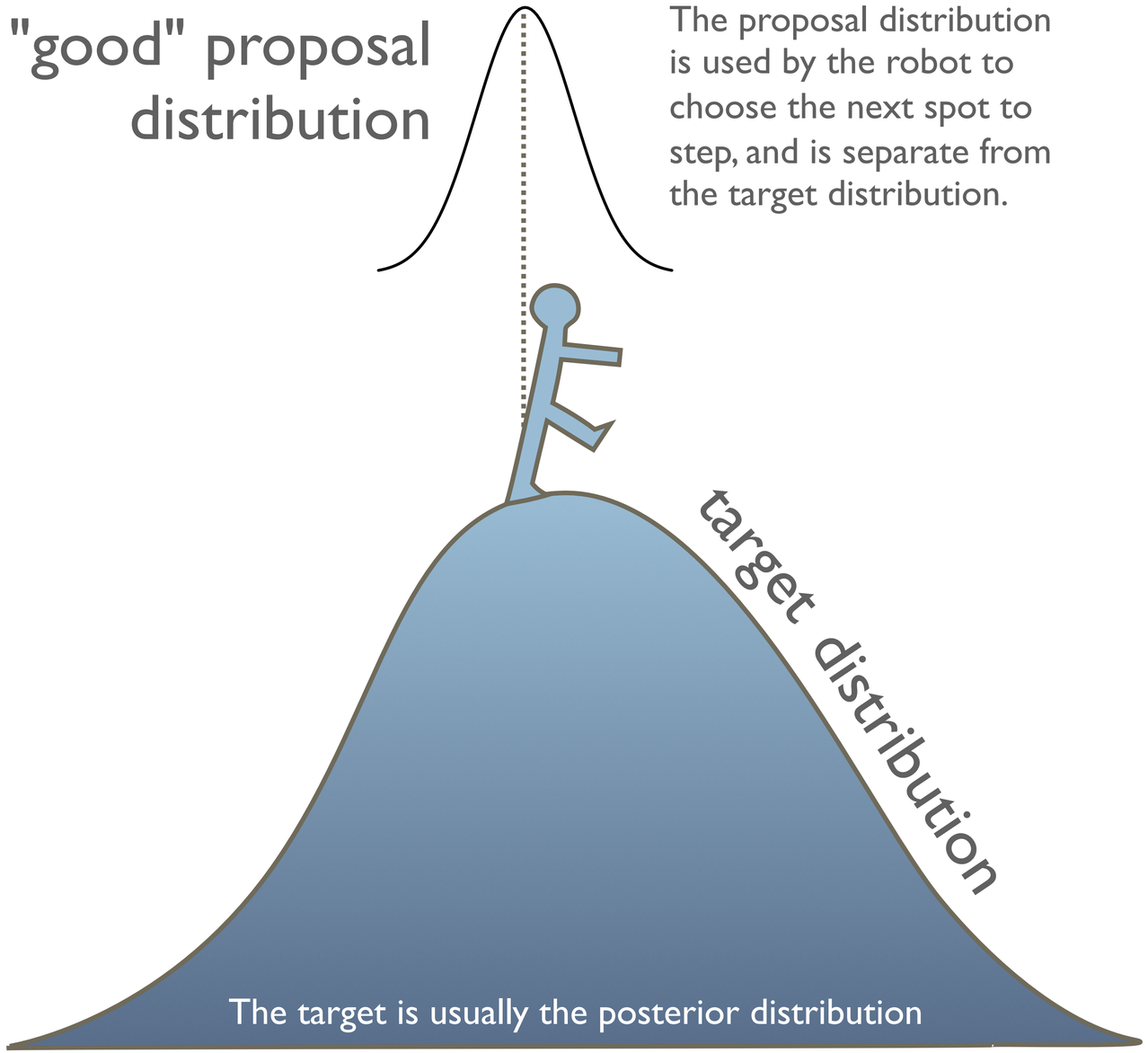
$$\frac{P(\text{0101... 1101... 0100...} \mid \text{tree}, \text{rates})\, P(\text{tree}, \text{rates})}{P(\cancel{\text{0101... 1101... 0100...}})}$$

The marginal probability of the data cancels out

$$= \frac{P(\text{0101... 1101... 0100...} \mid \text{tree}^*, \text{rates}^*)\, P(\text{tree}^*, \text{rates}^*)}{P(\text{0101... 1101... 0100...} \mid \text{tree}, \text{rates})\, P(\text{tree}, \text{rates})}$$

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

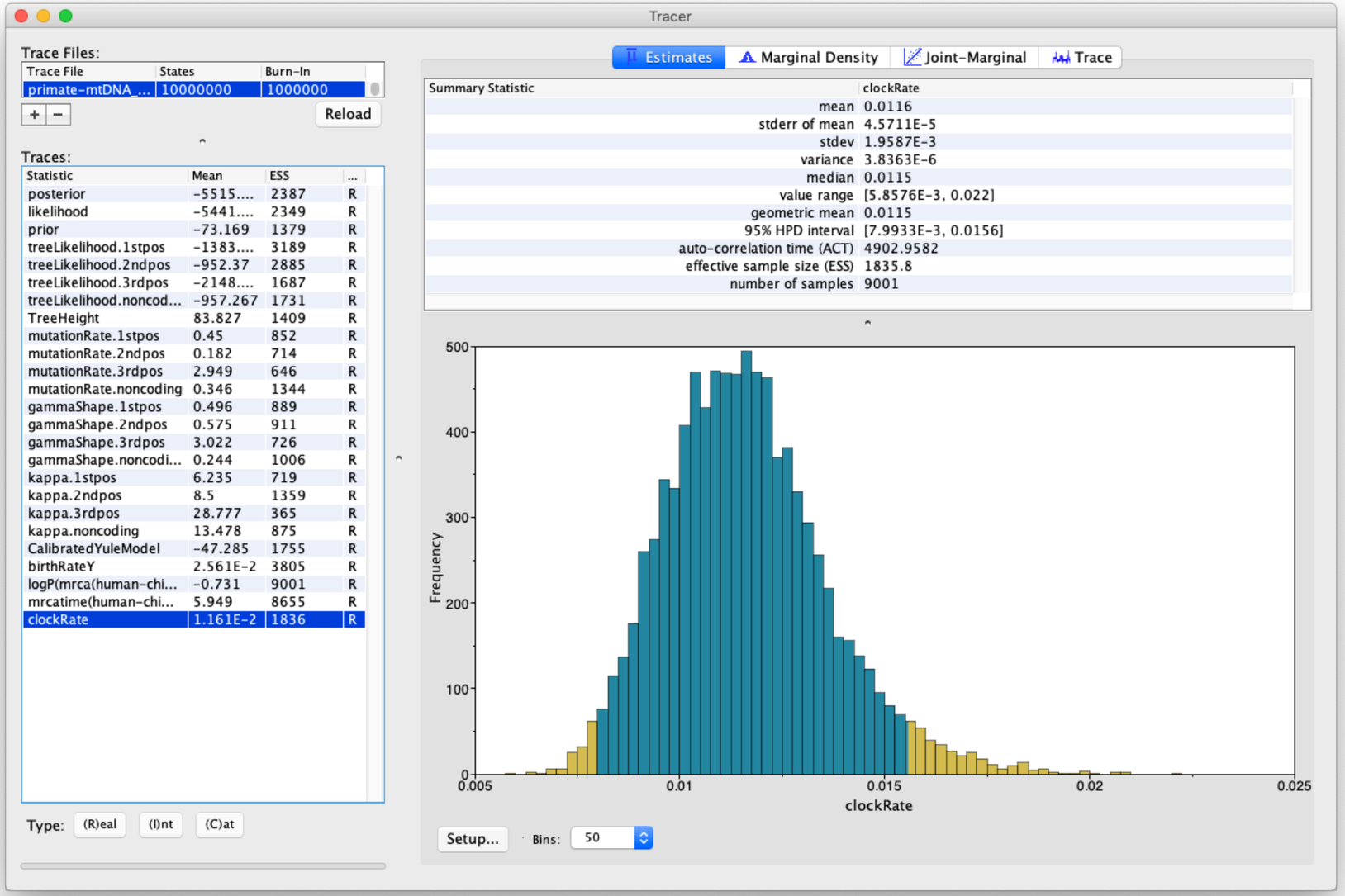
Proposals



Summarising the posterior

Tracer

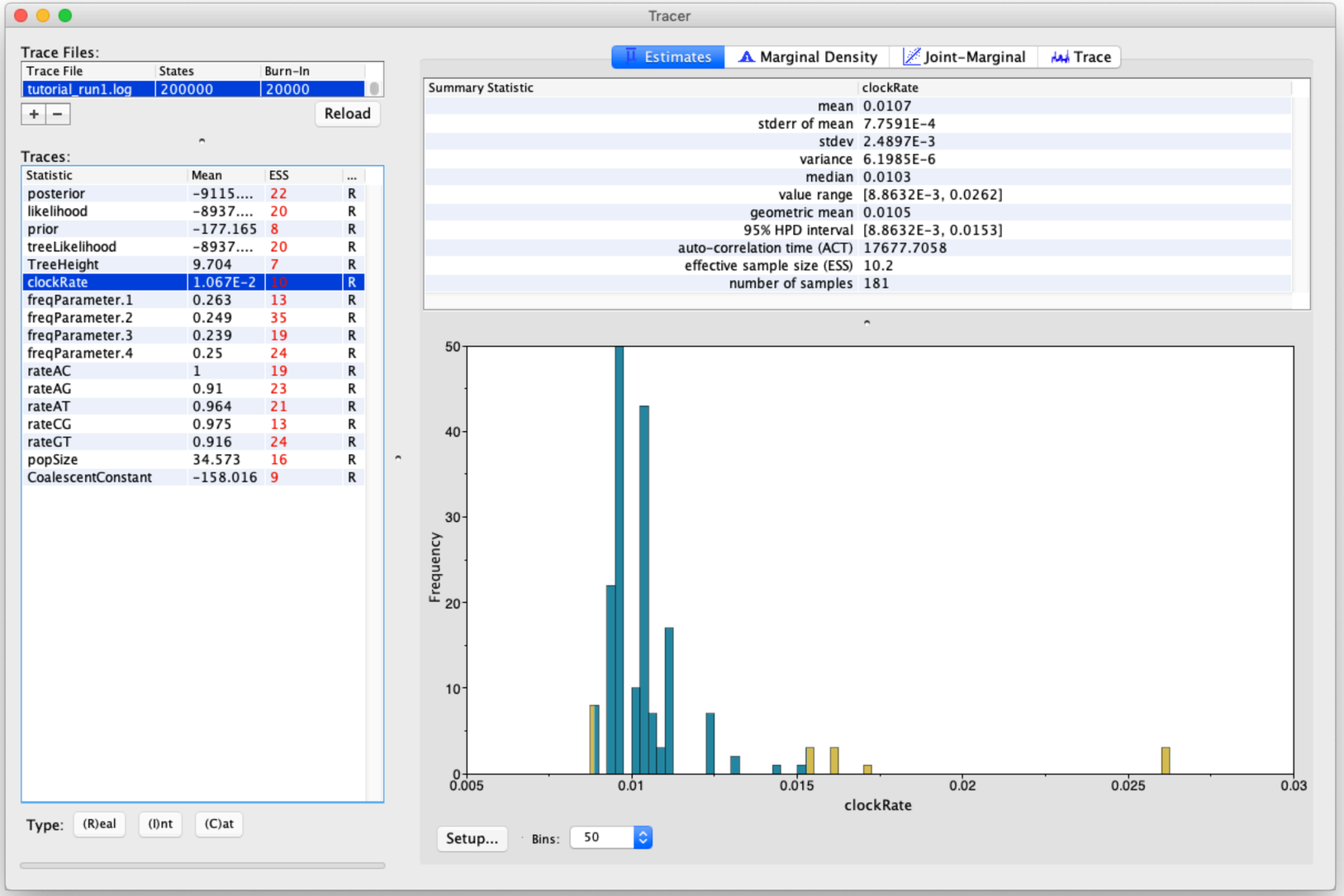
This is an example of good convergence



Summarising the posterior

Tracer

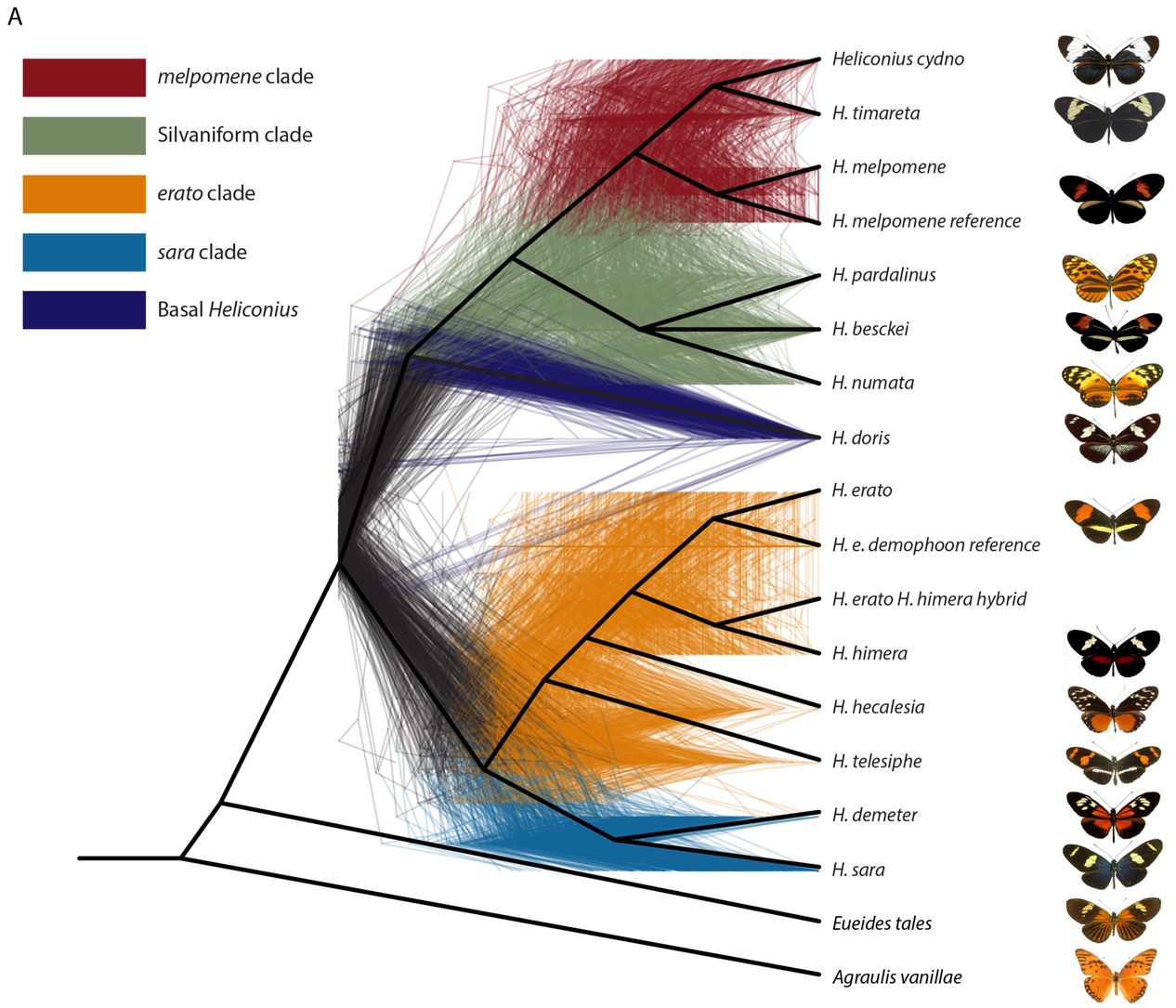
This is an example of poor convergence



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

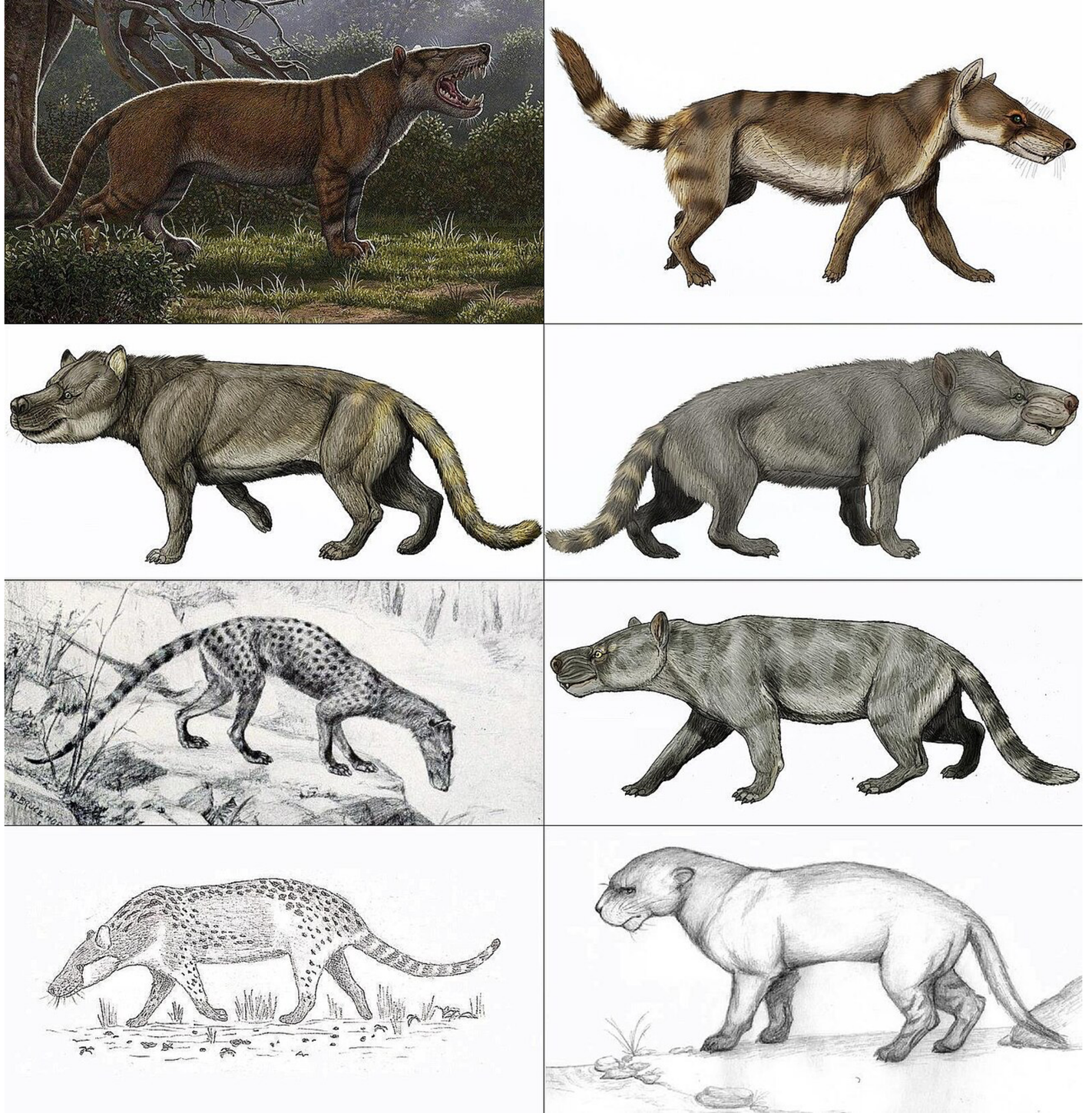
Marginal posterior density — the probability of a parameter regardless of the value of the others, represented by the histogram

Exercise overview

Hyaenodonts

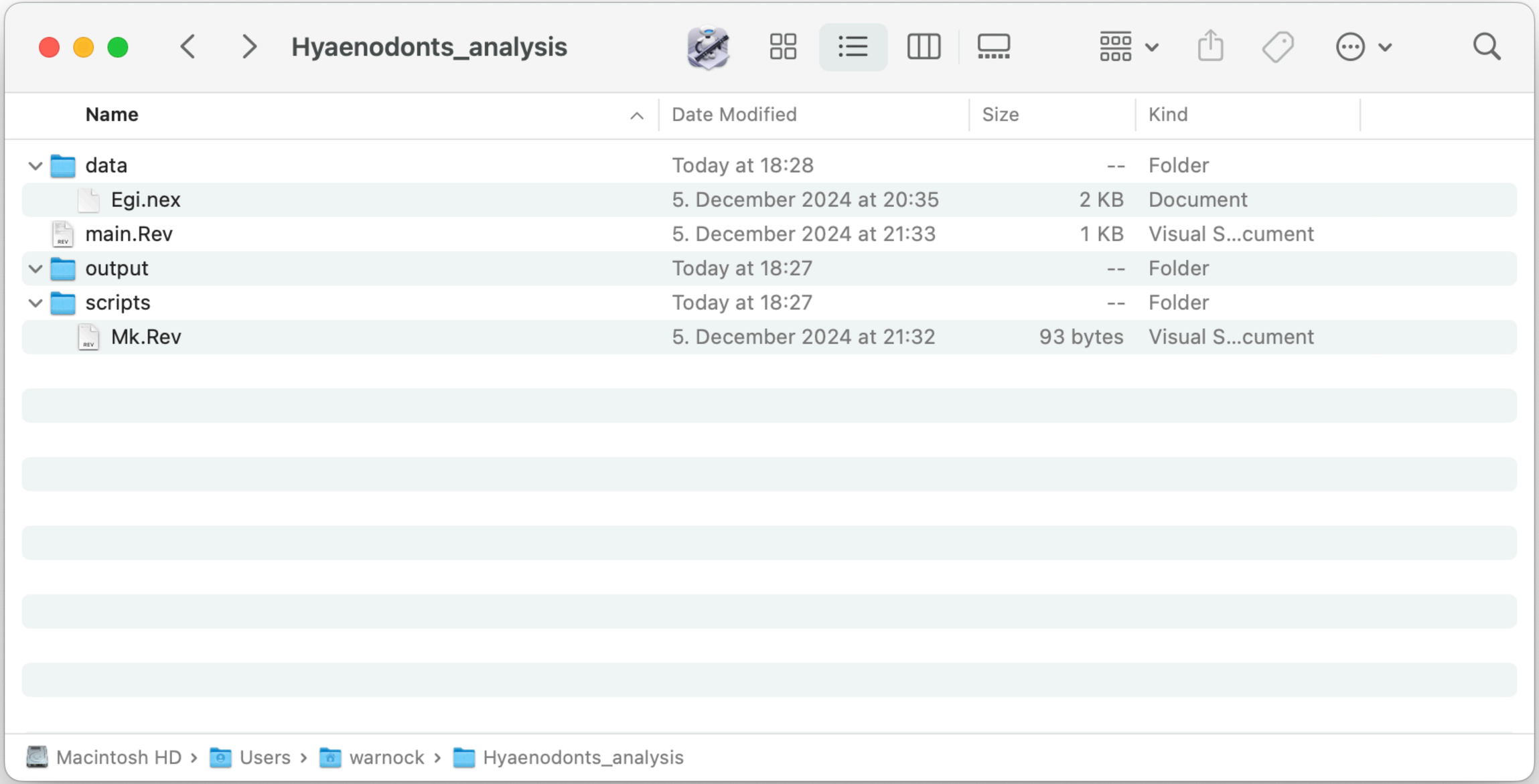
12 taxa and 65 characters

Extinct order of hyper-carnivorous placentals



Data from *Egi et al.* (2007)
Image source Wikipedia

```
#NEXUS

[File downloaded from graemetlloyd.com]

BEGIN DATA;
DIMENSIONS  NTAX=15 NCHAR=65;
  FORMAT DATATYPE = STANDARD SYMBOLS= " 0 1 2 3 4" MISSING=? GAP=- ;
MATRIX

missing_ORIG_Procerberus_and_Cimolestes  00000001000000000001000?00100000000000011000000010000000?0000000000
missing_ORIG_Proviverra_and_Lesmesodon   100????0?00110001001010??000101311102 1??20?1100001021010000110000
Allopterodon_ORIG_Allopterodon           10121110110110111111110?1100112100011110101100000212100002010000 0
Leonhardtina_ORIG_Leonhardtina           11121210210011110001000?1100012000011021202212000112000012020000 1
Eurotherium_ORIG_Eurotherium             20110010101100101111110211001100001110012011112201021100102200 0?0
Prodissopsalis_ORIG_Prodissopsalis       30100010111100011111?1112??0?00100011100101111102010011001102000 00
Cynohyaenodon_ORIG_Cynohyaenodon          11111001011110211111111100011000000120102011010200110101110200000
no_age_data_ORIG_Paracynohyaenodon        11111001011111211111110111 0?11100001201020111102011211011 0?200000
Brychotherium_ORIG_Brychotherium         10110????11112?11111110210012001000120202012102212011201 02?300111
Metasinopa_ORIG_Metasinopa               21???????00112101110110??11110020001202020111112121111111002000 10
Dissopsalis_ORIG_Dissopsalis             42010012112201111120222101?1?02000022100111111212011101010300110
Anasinopa_ORIG_Anasinopa                 311??01111221010112011 0??000000100002020012111121211011111020101 0
Paratritemnodon_ORIG_Paratritemnodon     10010112110112101110122010?0?11??00211011020112121101010010000 10
Masrasector_ORIG_Masrasector             10100002?00112?01111001??10110100000202001110111121100011011010 10
Kyawdawia_ORIG_Kyawdawia                 42010212111012101111111 0010???00?0?12010011201111 21?000110110001 0
;
END;
```


Hyaenodonts_analysis
Name
Date Modified
Size
Kind
data
Today at 18:28
--
Folder
Egi.nex
5. December 2024 at 20:35
2 KB
Document
main.Rev
5. December 2024 at 21:33
1 KB
Visual S...cument
output
Today at 18:27
--
Folder
scripts
Today at 18:27
--
Folder
Mk.Rev
5. December 2024 at 21:32
93 bytes
Visual S...cument
Macintosh HD
Users
warnock
Hyaenodonts_analysis

Files

Data

- **Egi.nex** (the nexus file, i.e., the character data)

Scripts

- **main.Rev** (used to read in the data, set up the tree model & MCMC settings)
- **Mk.Rev** (used the model of character evolution)

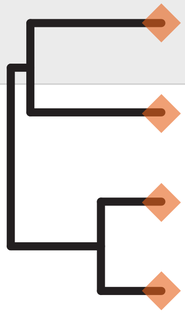
main.Rev
part 1

```
# read in character data
morpho <- readDiscreteCharacterData("data/Egi.nex")


# set up some useful variables
num_taxa <- morpho.ntaxa() # number of taxa
num_branches <- 2 * num_taxa - 3 # number of branches in an unrooted tree
taxa <- morpho.taxa() # list of taxon names


moves    = VectorMoves()
monitors = VectorMonitors()
```

main.Rev part 2

```
# prior on the tree topology
topology ~ dnUniformTopology(taxa)

moves.append( mvNNI(topology, weight = num_taxa) ) # nearest neighbour interchange
moves.append( mvSPR(topology, weight = num_taxa/10.0) ) # subtree pruning and regrafting

# prior on the branch length
for (i in 1:num_branches) {
   br_lens[i] ~ dnExponential(10.0)
   moves.append( mvScale(br_lens[i]) )
}

phylogeny := treeAssembly(topology, br_lens)
TL := sum(br_lens)
```

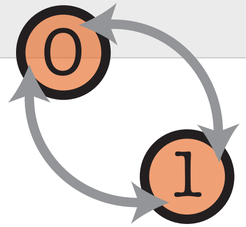
prior on the tree topology & branch lengths



Mk.Rev

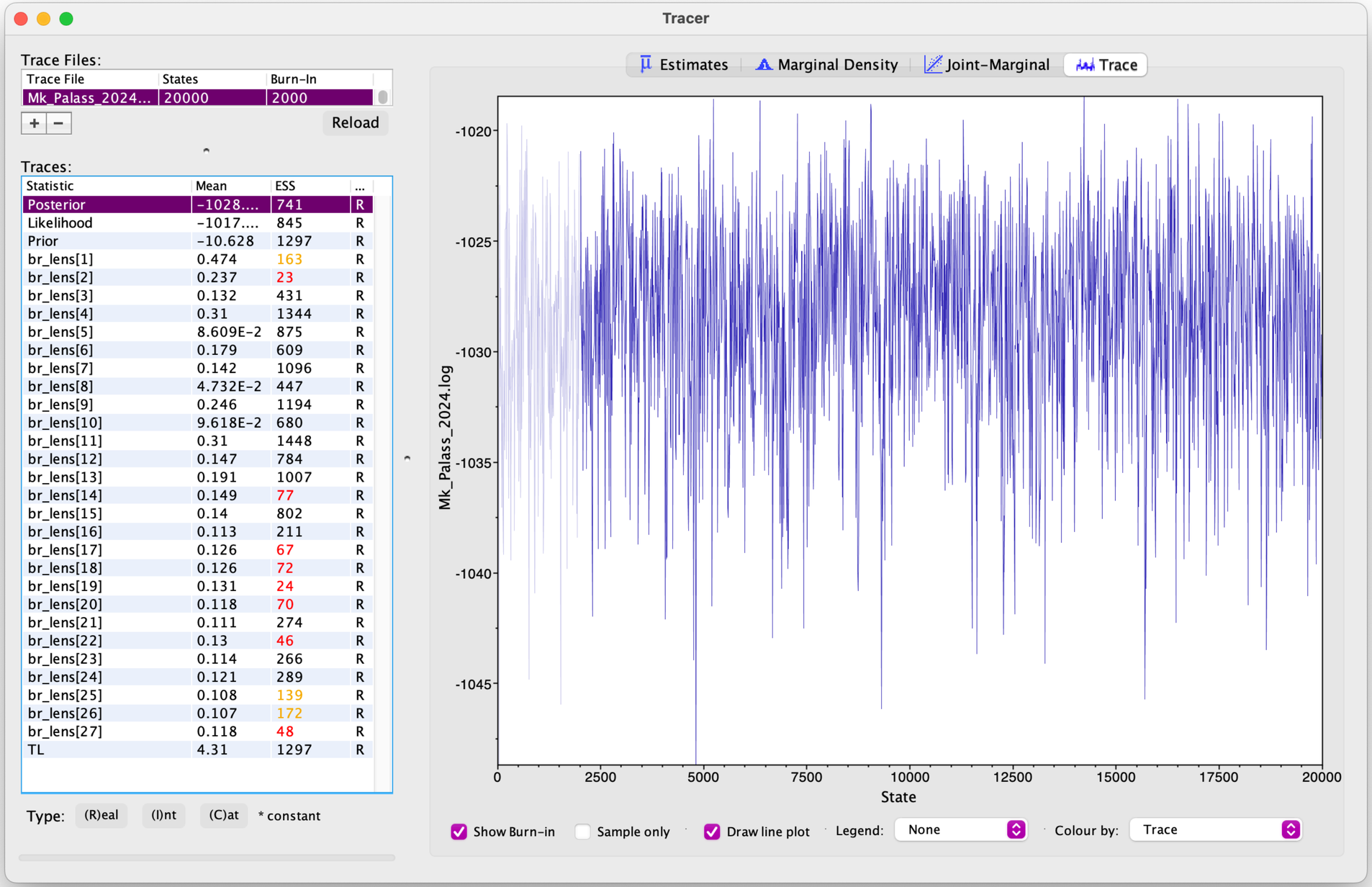
```
# define the Q matrix
Q <- fnJC(5)

# bring everything together
seq ~ dnPhyloCTMC(tree = phylogeny, Q = Q, type = "Standard")

seq.clamp(morpho)
```

Substitution model



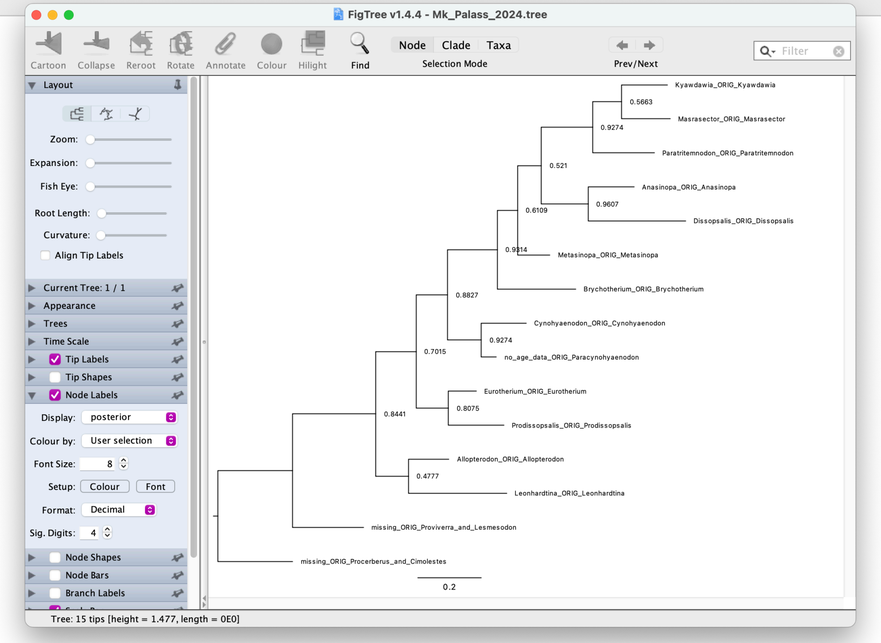
main.Rev
part 3

```
source("scripts/Mk.Rev")


mymodel = model(phylogeny)


# parameters printed to file
monitors.append( mnModel(filename = "output/Mk_Palass_2024.log", printgen = 10) )
# trees printed to file
monitors.append( mnFile(filename = "output/Mk_Palass_2024.trees", printgen = 10,
phylogeny) )
# parameter values printed to screen during the MCMC
monitors.append( mnScreen(printgen = 100, TL) )


mymcmc = mcmc(mymodel, monitors, moves)
mymcmc.run(generations = 20000)
```


Tracer
Trace Files:
Trace File
States
Burn-In
Mk_Palass_2024...
20000
2000
Reload
Traces:
Statistic
Mean
ESS
...
Posterior
-1028....
741
R
Likelihood
-1017....
845
R
Prior
-10.628
1297
R
br_lens[1]
0.474
163
R
br_lens[2]
0.237
23
R
br_lens[3]
0.132
431
R
br_lens[4]
0.31
1344
R
br_lens[5]
8.609E-2
875
R
br_lens[6]
0.179
609
R
br_lens[7]
0.142
1096
R
br_lens[8]
4.732E-2
447
R
br_lens[9]
0.246
1194
R
br_lens[10]
9.618E-2
680
R
br_lens[11]
0.31
1448
R
br_lens[12]
0.147
784
R
br_lens[13]
0.191
1007
R
br_lens[14]
0.149
77
R
br_lens[15]
0.14
802
R
br_lens[16]
0.113
211
R
br_lens[17]
0.126
67
R
br_lens[18]
0.126
72
R
br_lens[19]
0.131
24
R
br_lens[20]
0.118
70
R
br_lens[21]
0.111
274
R
br_lens[22]
0.13
46
R
br_lens[23]
0.114
266
R
br_lens[24]
0.121
289
R
br_lens[25]
0.108
139
R
br_lens[26]
0.107
172
R
br_lens[27]
0.118
48
R
TL
4.31
1297
R
Type:
(R)eal
(I)nt
(C)at
* constant
Estimates
Marginal Density
Joint-Marginal
Trace
Mk_Palass_2024.log
-1020
-1025
-1030
-1035
-1040
-1045
0
2500
5000
7500
10000
12500
15000
17500
20000
State
Show Burn-in
Sample only
Draw line plot
Legend:
None
Colour by:
Trace

main.Rev
part 4

```
# read the tree file back in
treetrace = readTreeTrace("output/Mk_Palass_2024.trees", treetype = "non-clock")

# generate an MAP summary tree
map_tree = mapTree(treetrace, "output/Mk_Palass_2024.tree")
```



Exercise

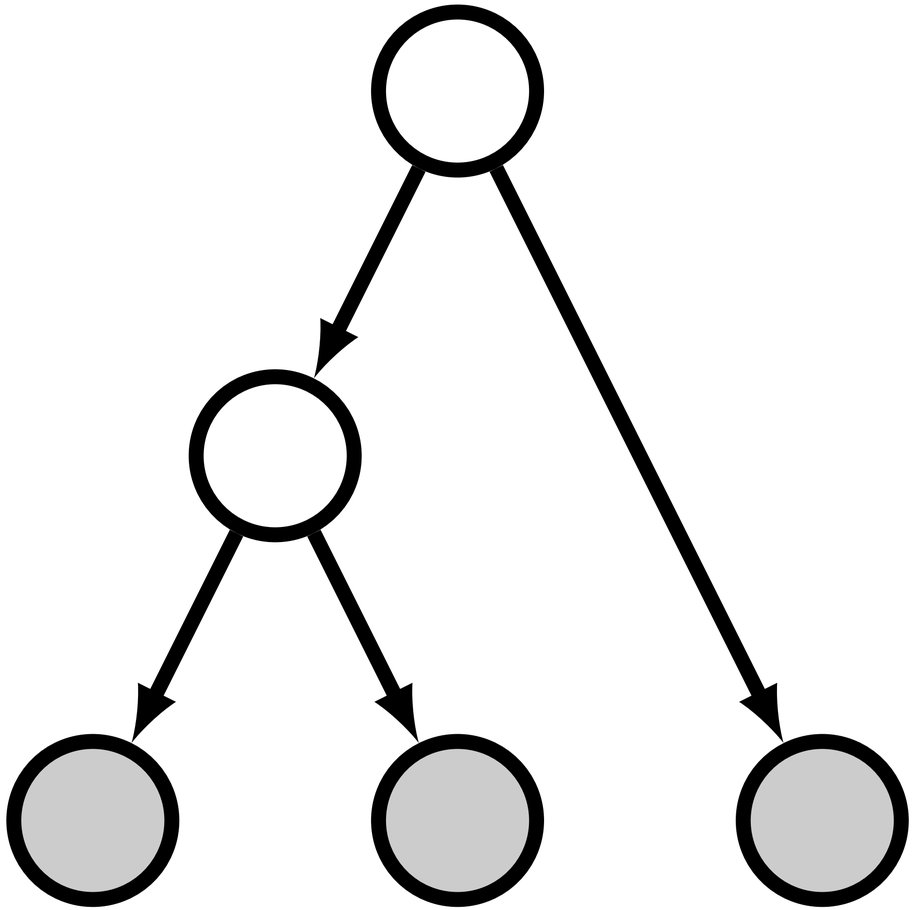
Extra slides

Graphical models

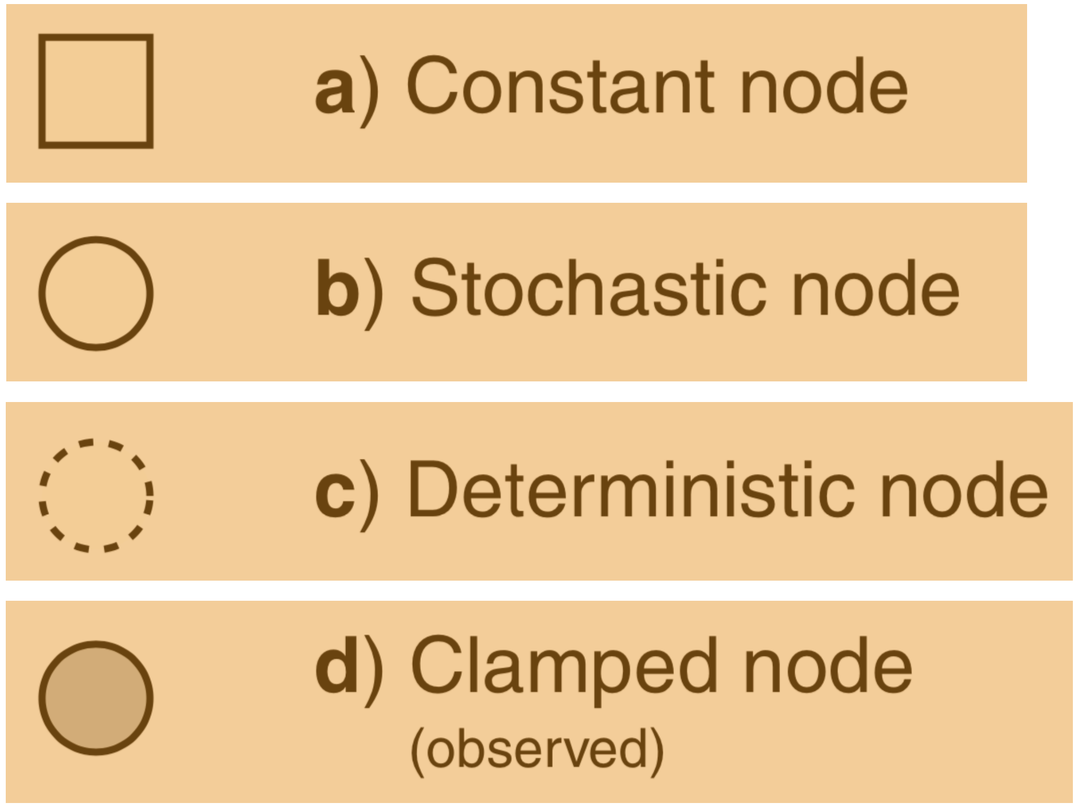
Graphical models

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)

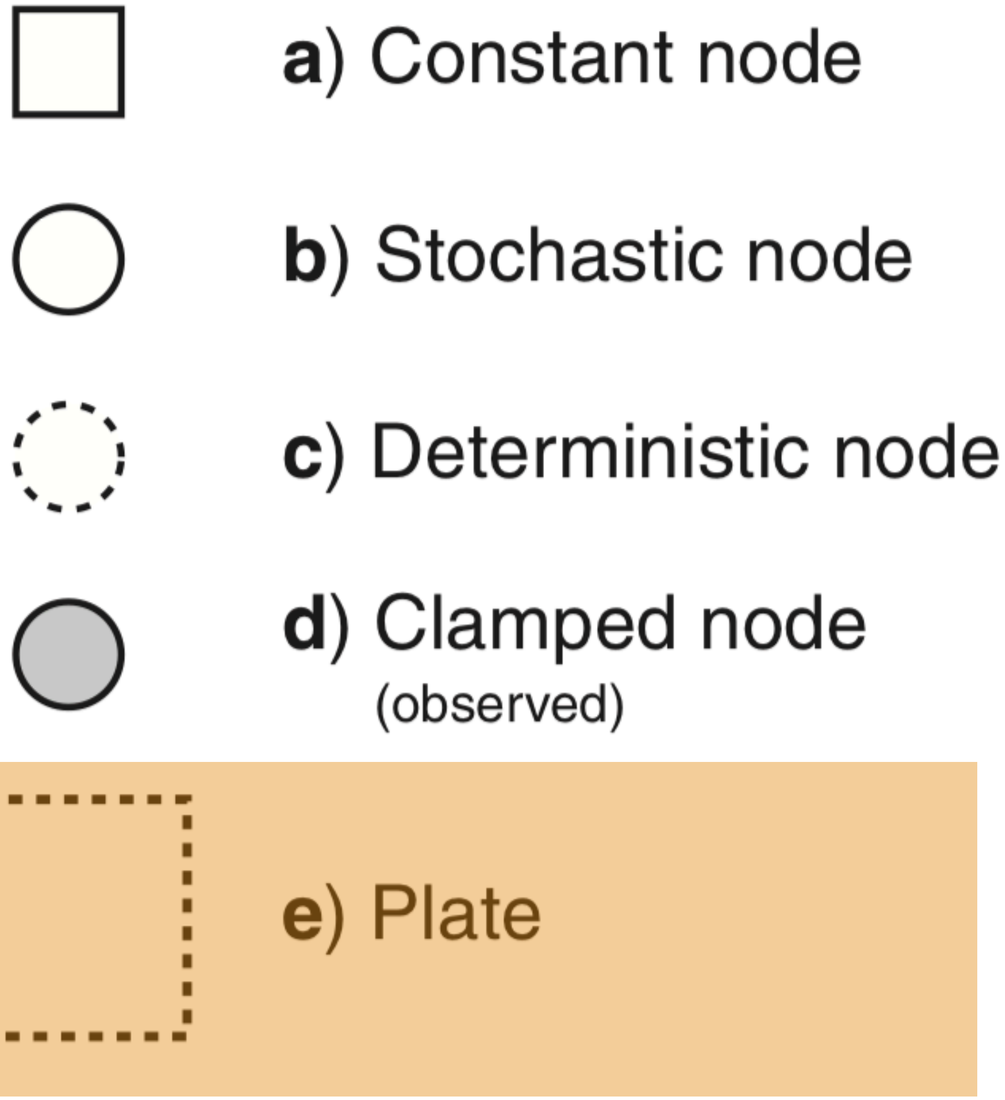


a. fixed value variables

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.

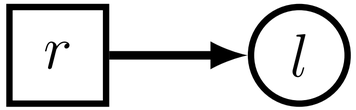
Operator	**Variable**
`<-`	constant variable
`~`	stochastic variable
`:=`	deterministic variable
`node.clamp(data)`	clamped variable
`=`	inference (*i.e.*, non-model) variable
`for(i in 1:N){...}`	plate

a)


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

a)

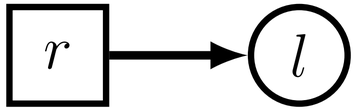
b)



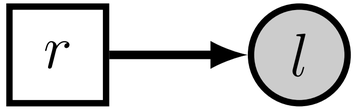

```
# constant node
r <- 10


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

a)

b)



c)

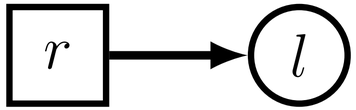



```
# constant node
r <- 10


# stochastic node
l ~ dnExp(r)


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

a)

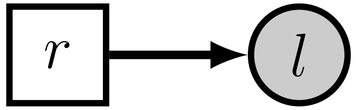

```
# constant node
r <- 10
```

b)



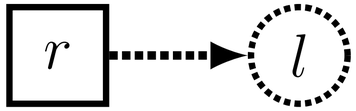

```
# stochastic node
l ~ dnExp(r)
```

c)




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

d)

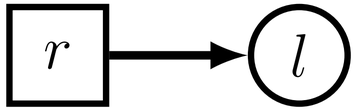



```
# deterministic node
l := exp(r)
```

a)

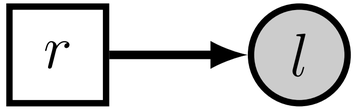

```
# constant node
r <- 10
```

b)



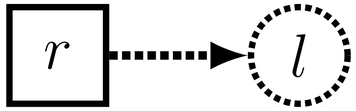

```
# stochastic node
l ~ dnExp(r)
```

c)



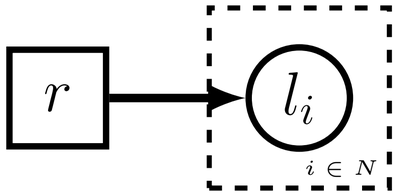

```
# stochastic node (observed)
l.clamp(0.1)
```

d)




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

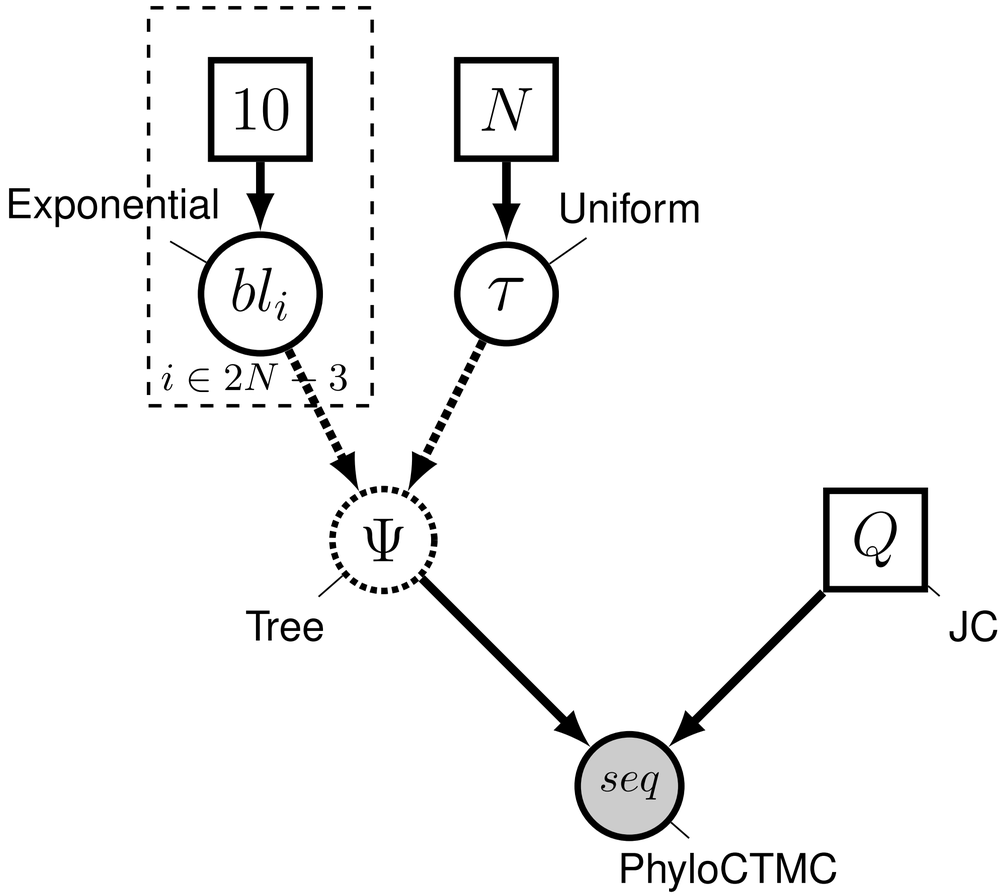
e)

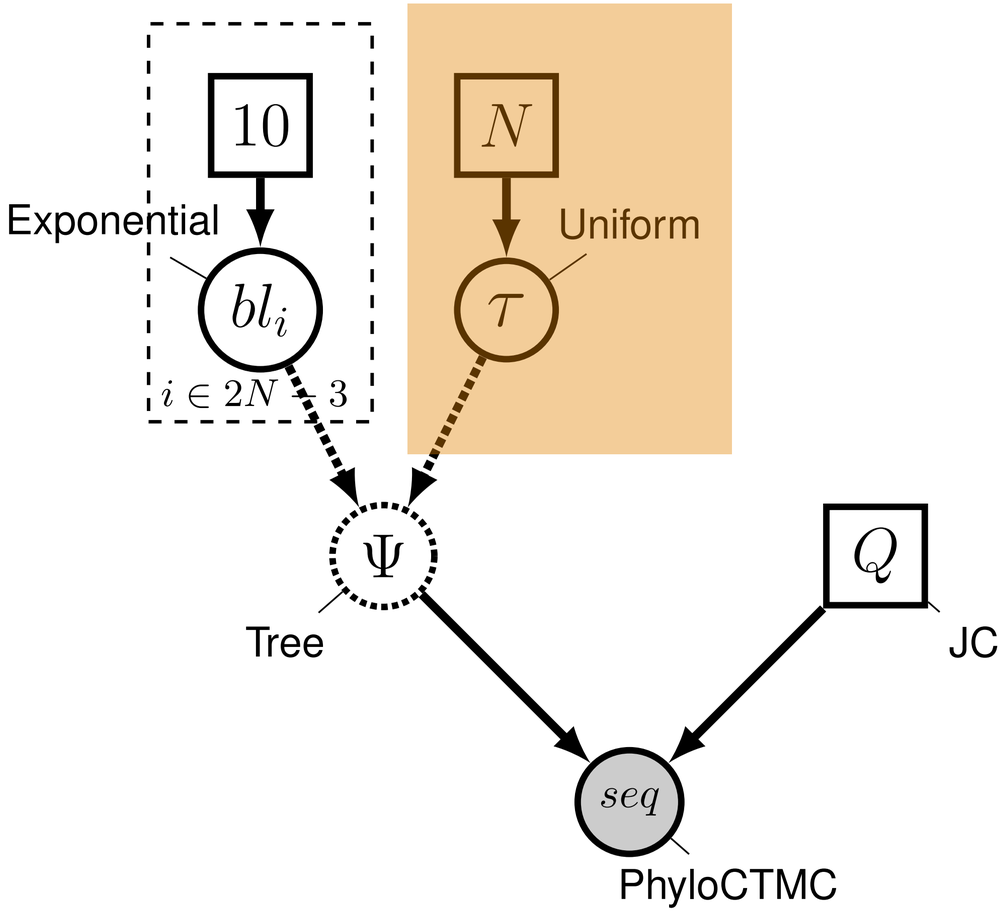



```
# stochastic nodes (iid)
for (i in 1:N) {
    l[i] ~ dnExp(r)
}
```

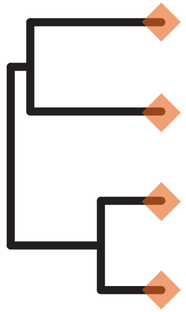
Graphical model example (DNA)

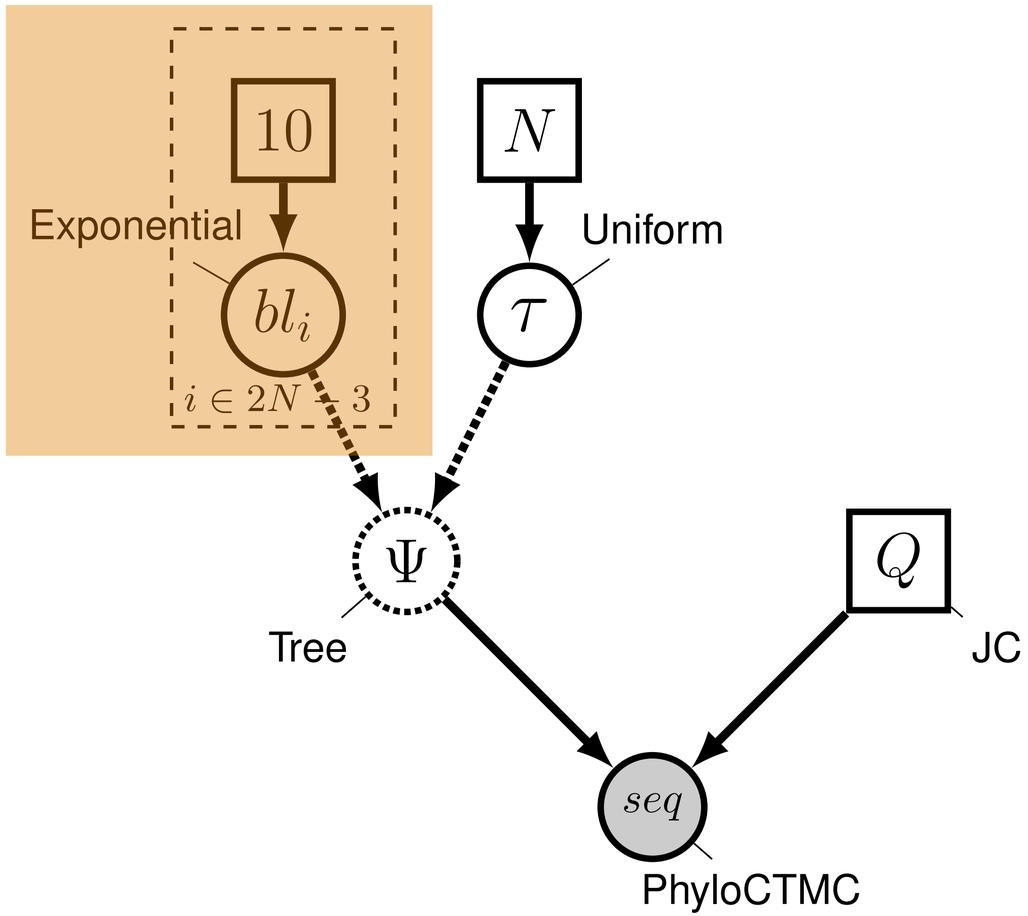
$


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

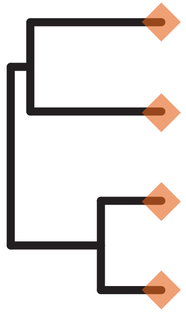


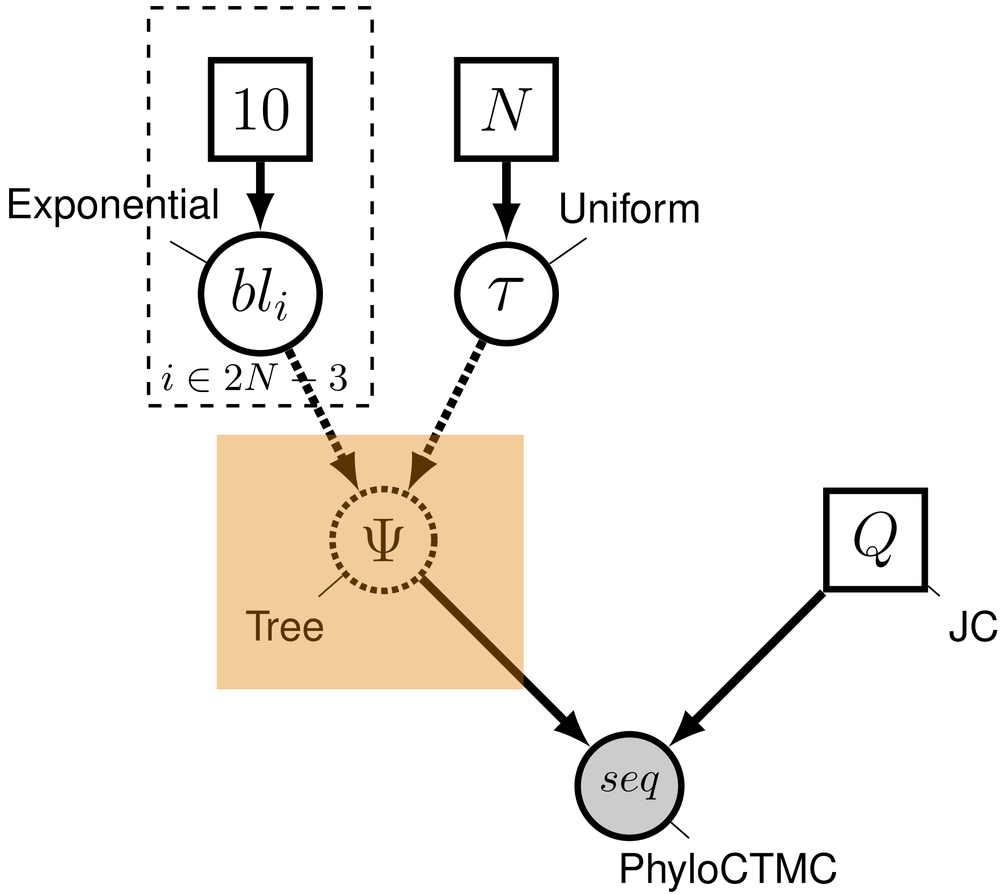
prior on the tree topology



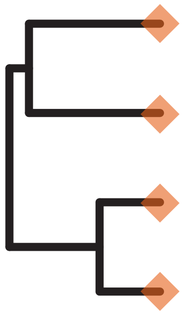


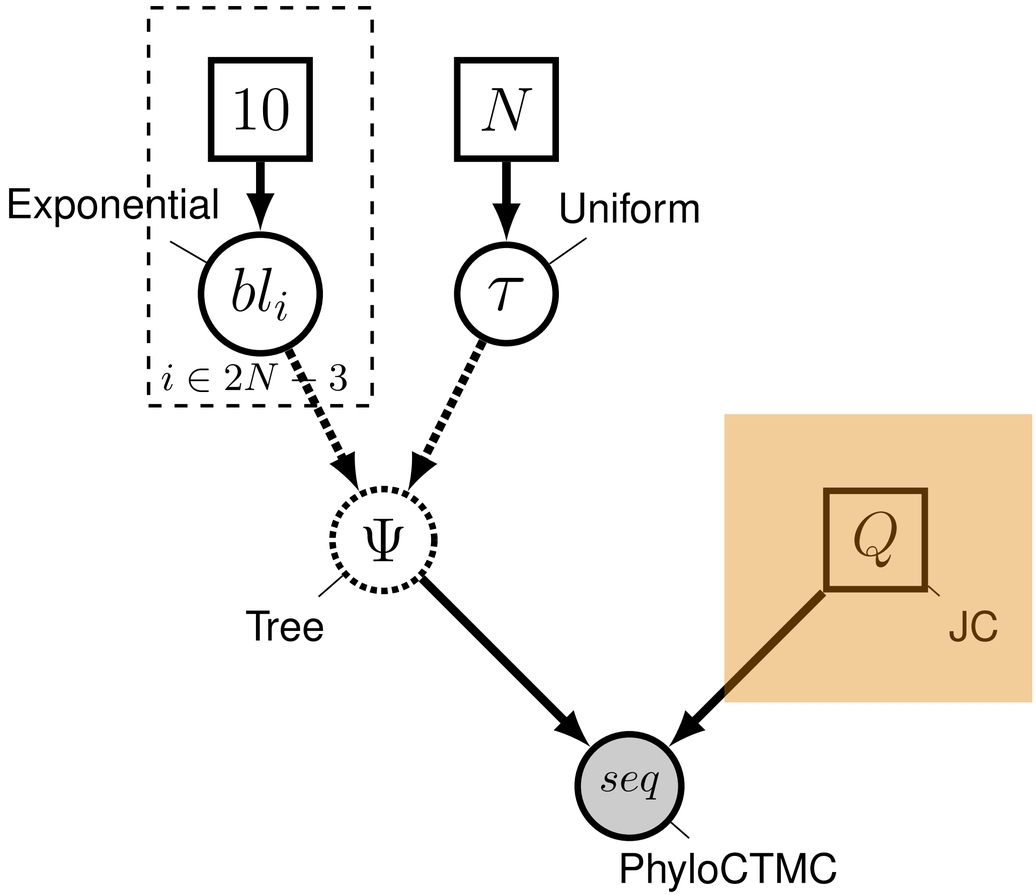
prior on the branch lengths



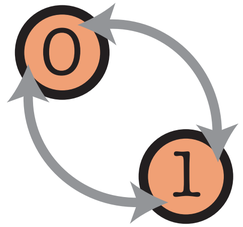


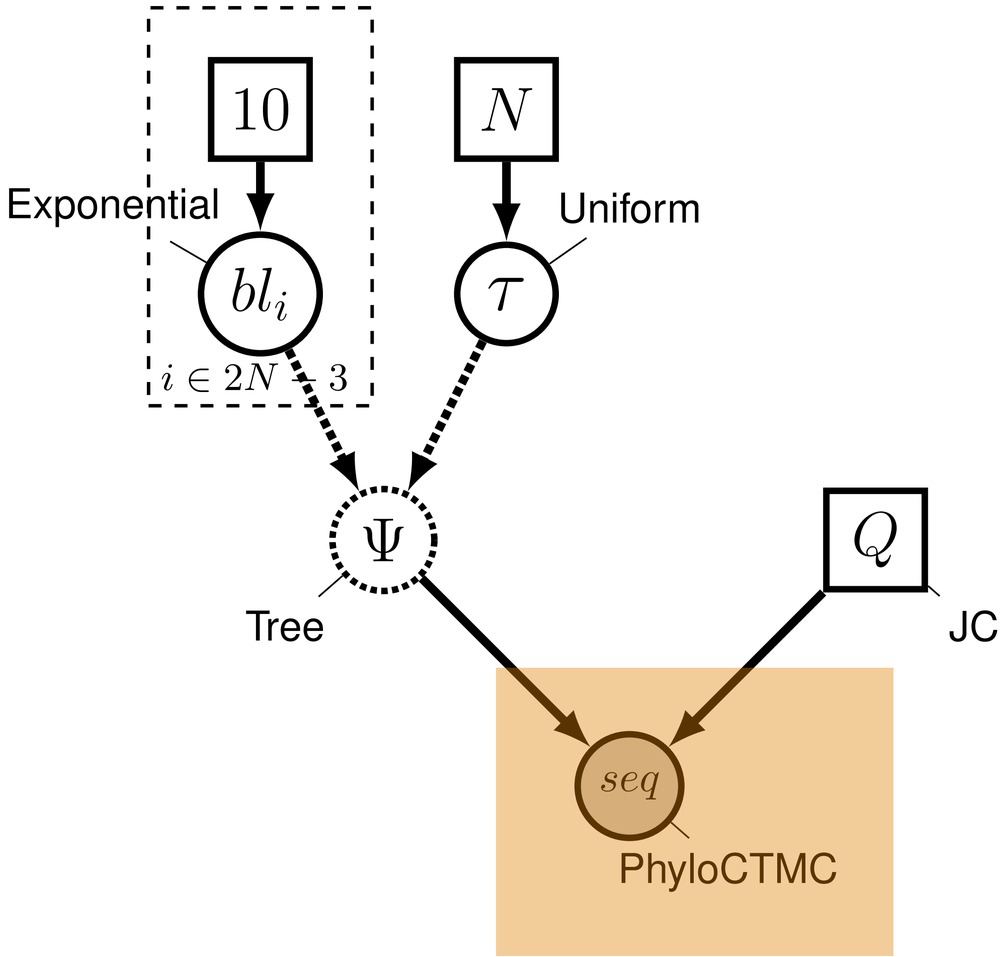
we can combine the topology and branch lengths





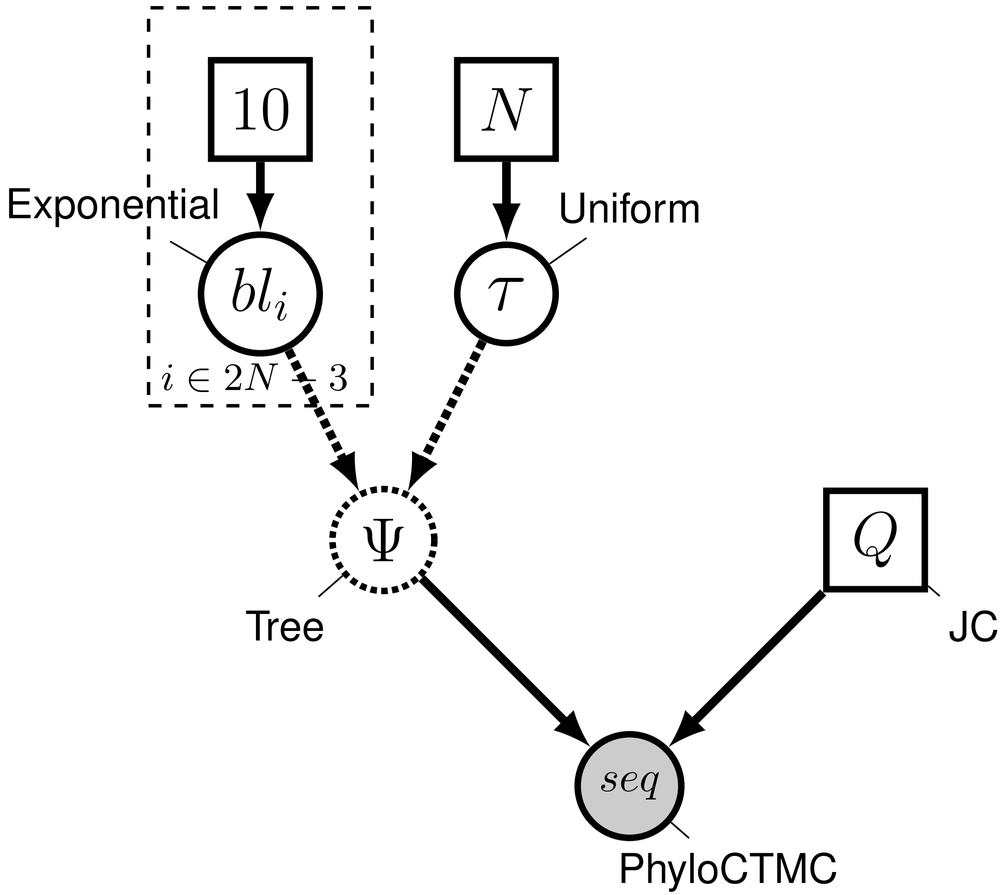
Substitution model



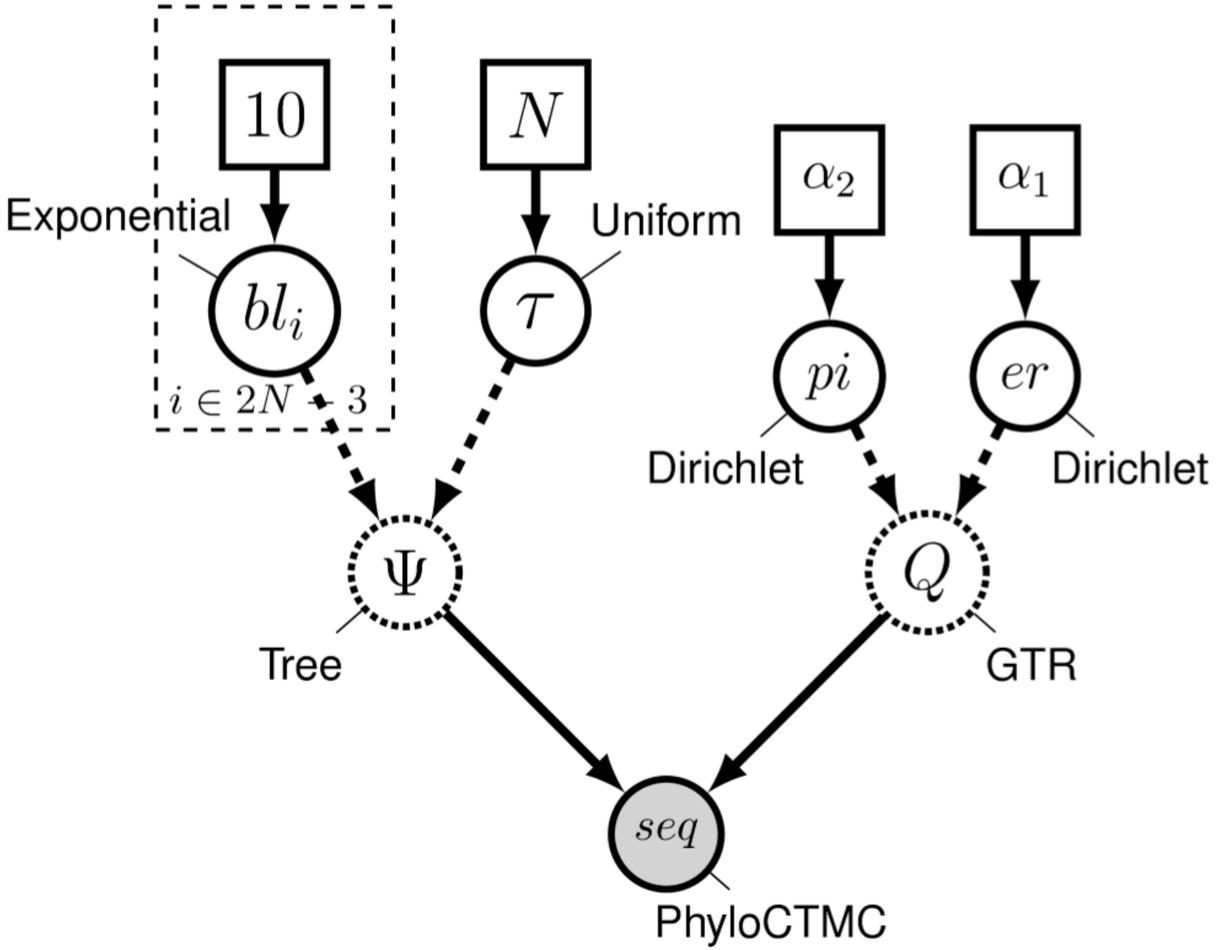


Observed data

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 )
```




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


alpha1 <- v(1,1,1,1,1,1)
alpha2 <- v(1,1,1,1)
er ~ dnDirichlet( alpha1 )
pi ~ dnDirichlet( alpha2 )
Q := fnGTR(er, pi)
```