Analytical paleo

Introduction to RevBayes and graphical models

Rachel Warnock

21.08.26



RevBayes

Phylogenetic inference — a better way?



Model developers

Empirical researchers

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

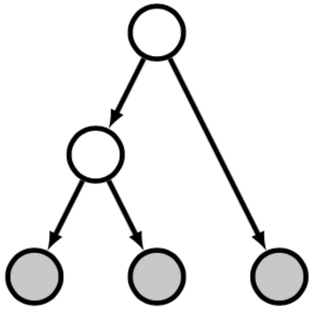
Developed and supported by a large international team of developers

revbayes.github.io

Höhna et al. (2016)

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.

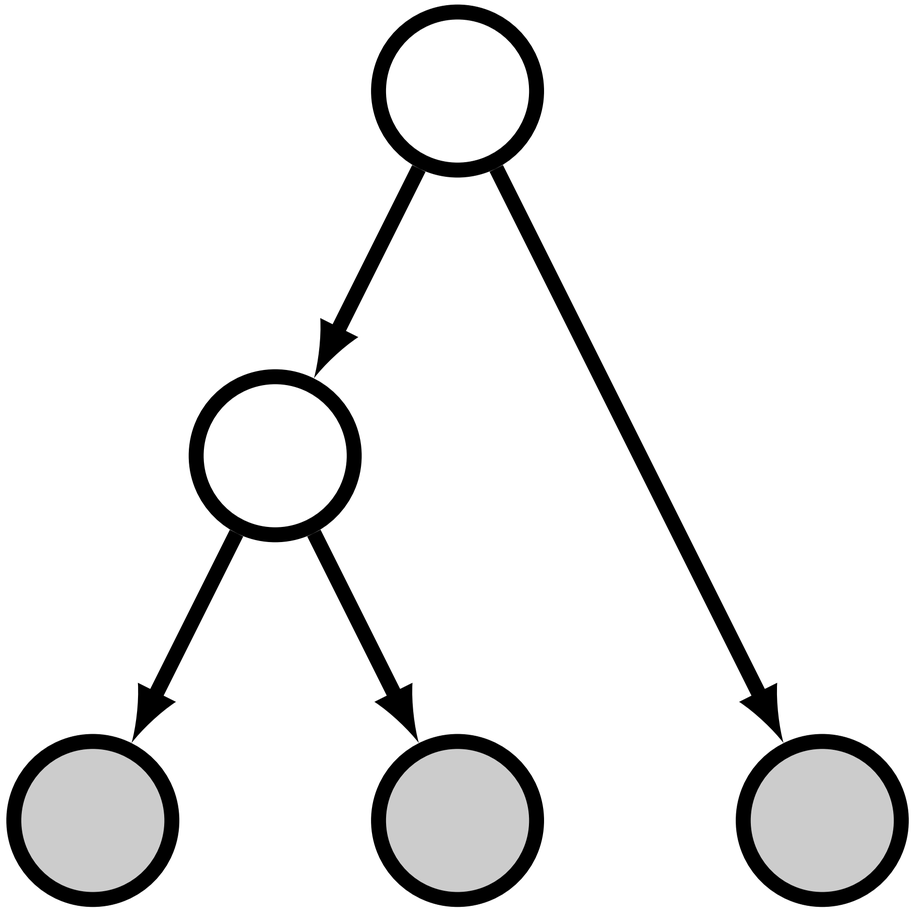
GitHub | License | Citation | Users Forum

Graphical models

Graphical models

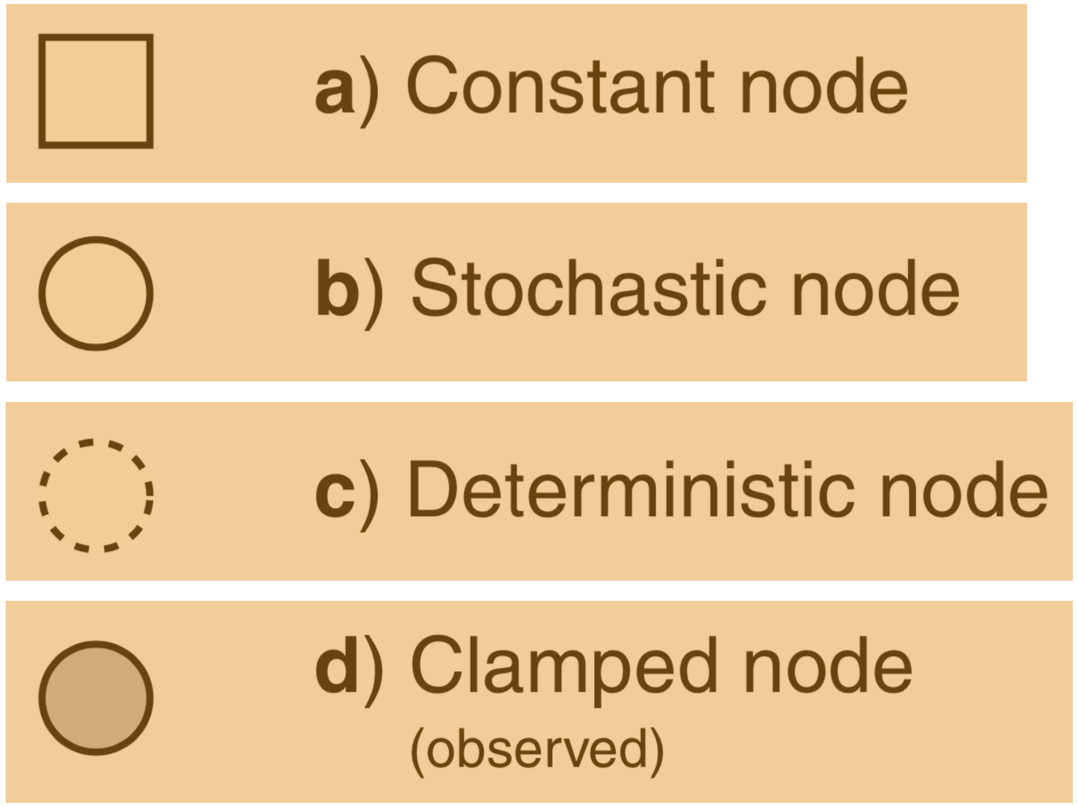
Provide tools for visually and computationally representing complex, parameter-rich models

Depict the conditional dependence structure of parameters and other random variables



Höhna et al. (2014)

Types of variables (nodes)

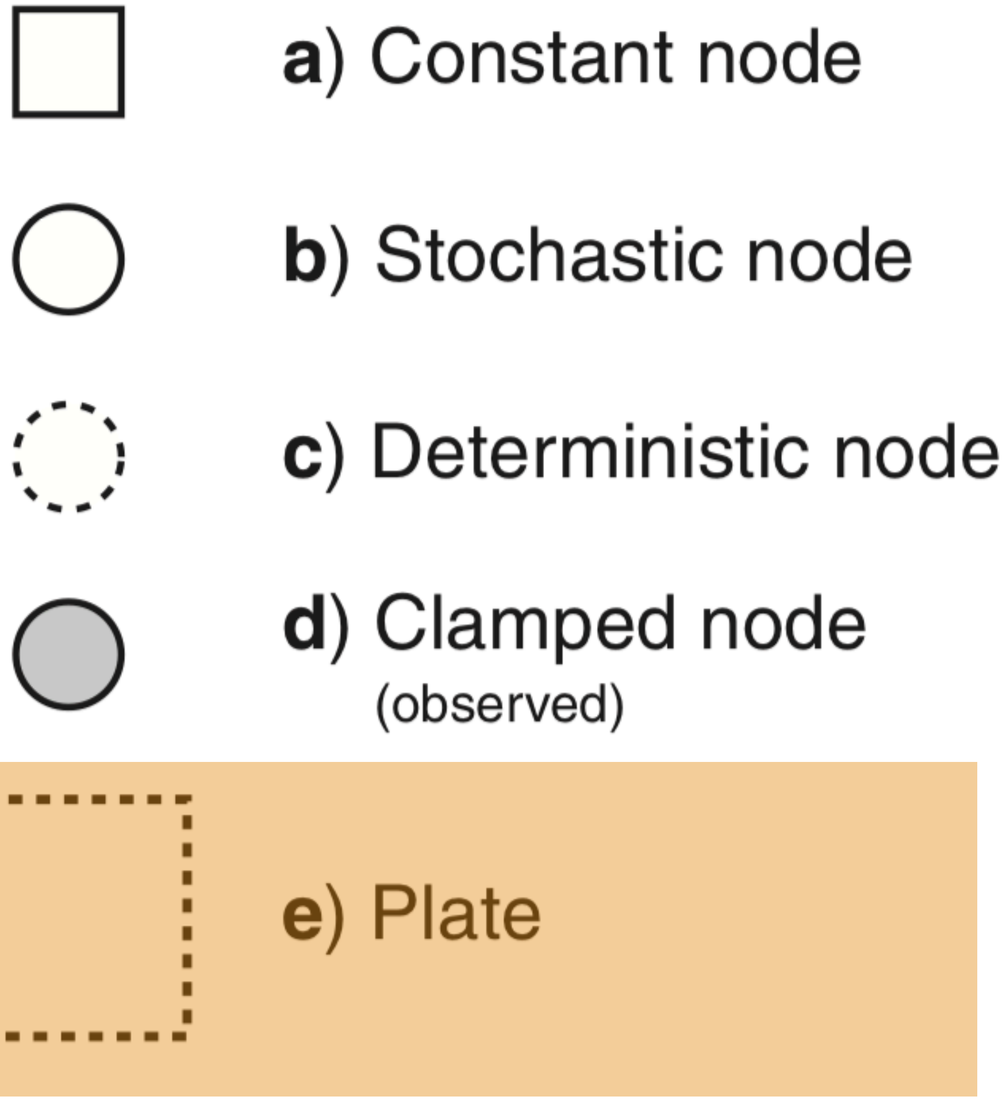


a. fixed value variables

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)

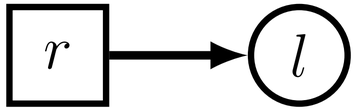



```
# constant node
r <- 10
```

a)



b)

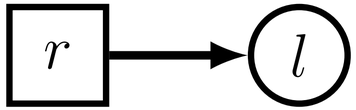



```
# constant node
r <- 10


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

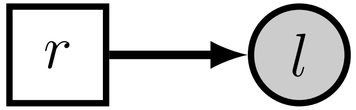
a)



b)



c)



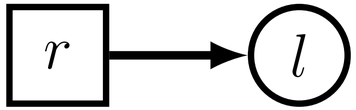

```
# constant node
r <- 10


# stochastic node
l ~ dnExp(r)


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

a)



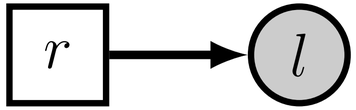

```
# constant node
r <- 10
```

b)



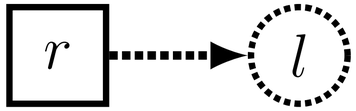

```
# stochastic node
l ~ dnExp(r)
```

c)




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

d)



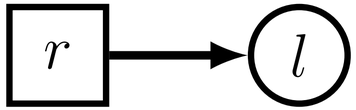

```
# deterministic node
l := exp(r)
```

a)



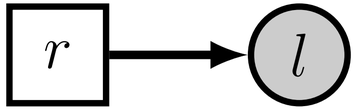

```
# constant node
r <- 10
```

b)



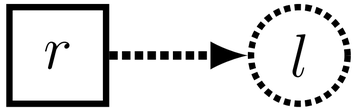

```
# stochastic node
l ~ dnExp(r)
```

c)



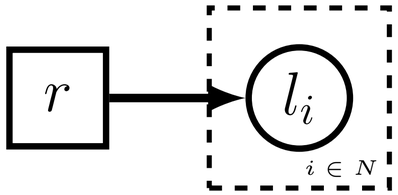

```
# stochastic node (observed)
l.clamp(0.1)
```

d)




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

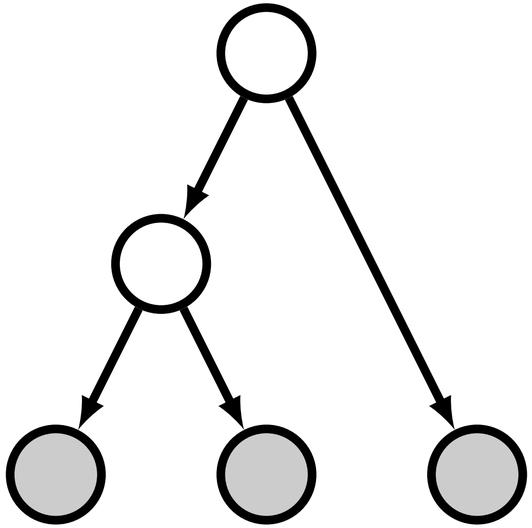
e)




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

Running RevBayes demo

Exercise



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

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

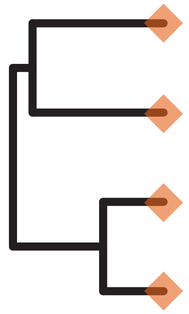
The posterior is proportional to the product of the prior and the likelihood

Components used to infer trees

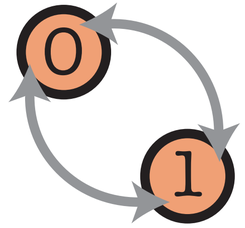
without considering time

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

data
sequences or characters

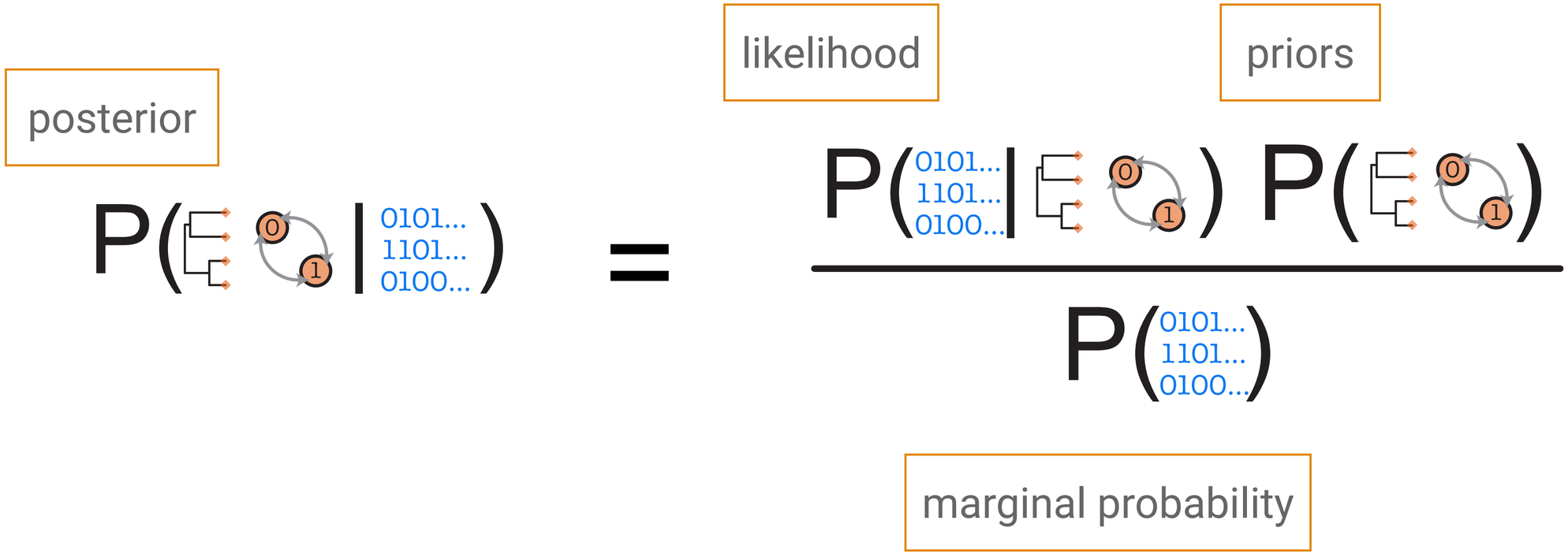


tree
topology and branch lengths

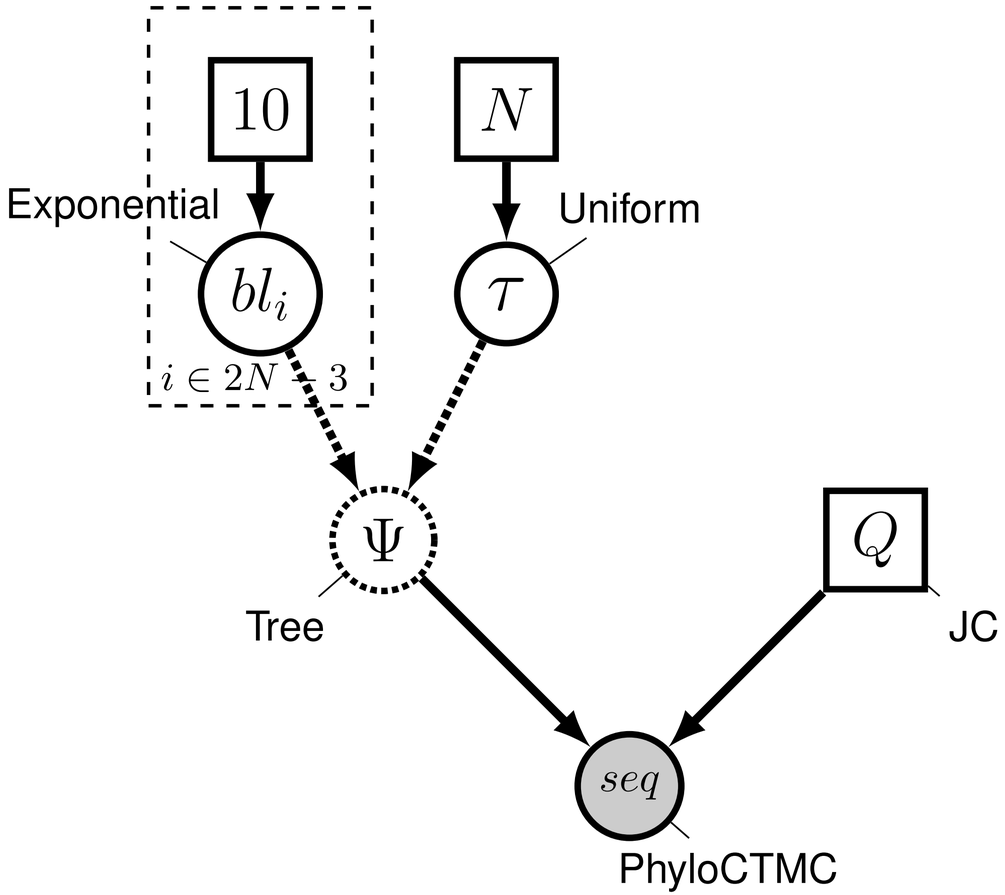


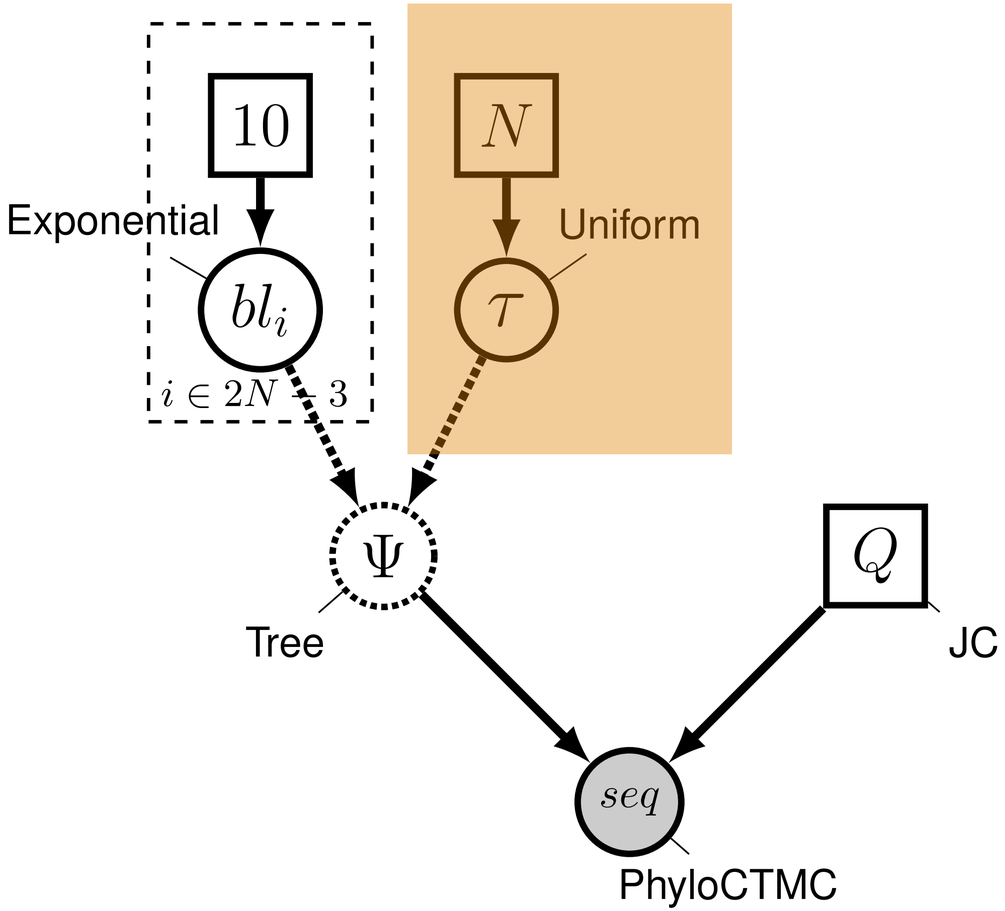
substitution model

Bayesian tree inference

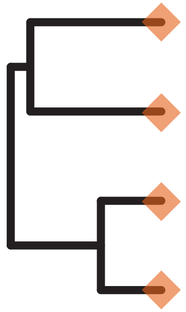


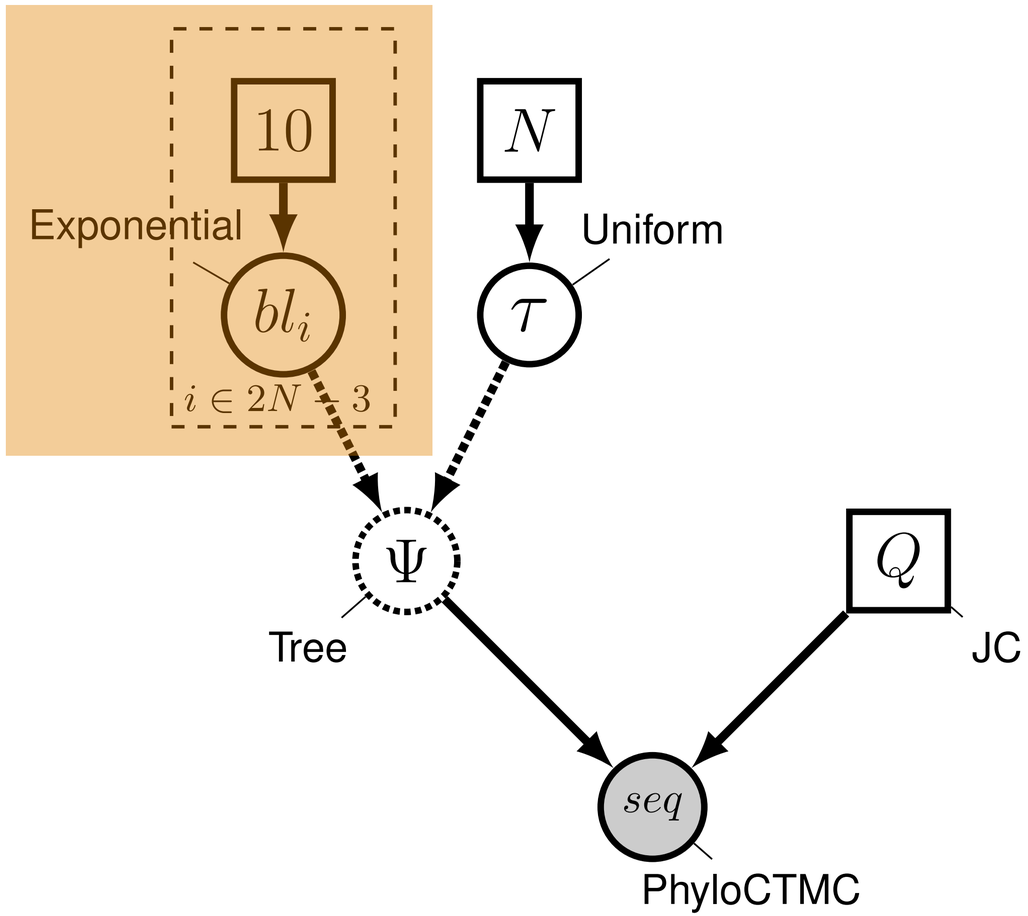
$


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

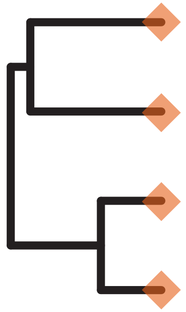


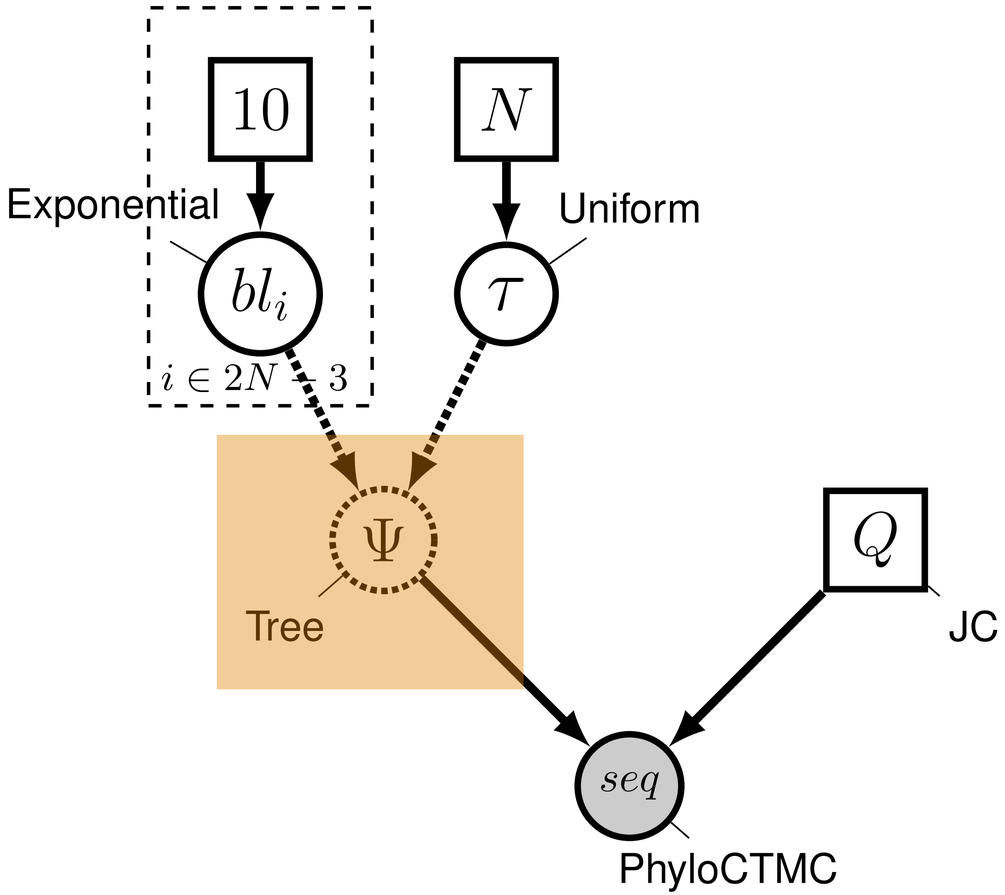
prior on the tree topology



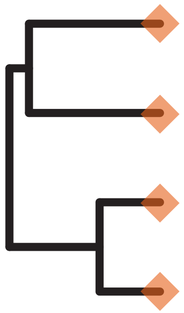


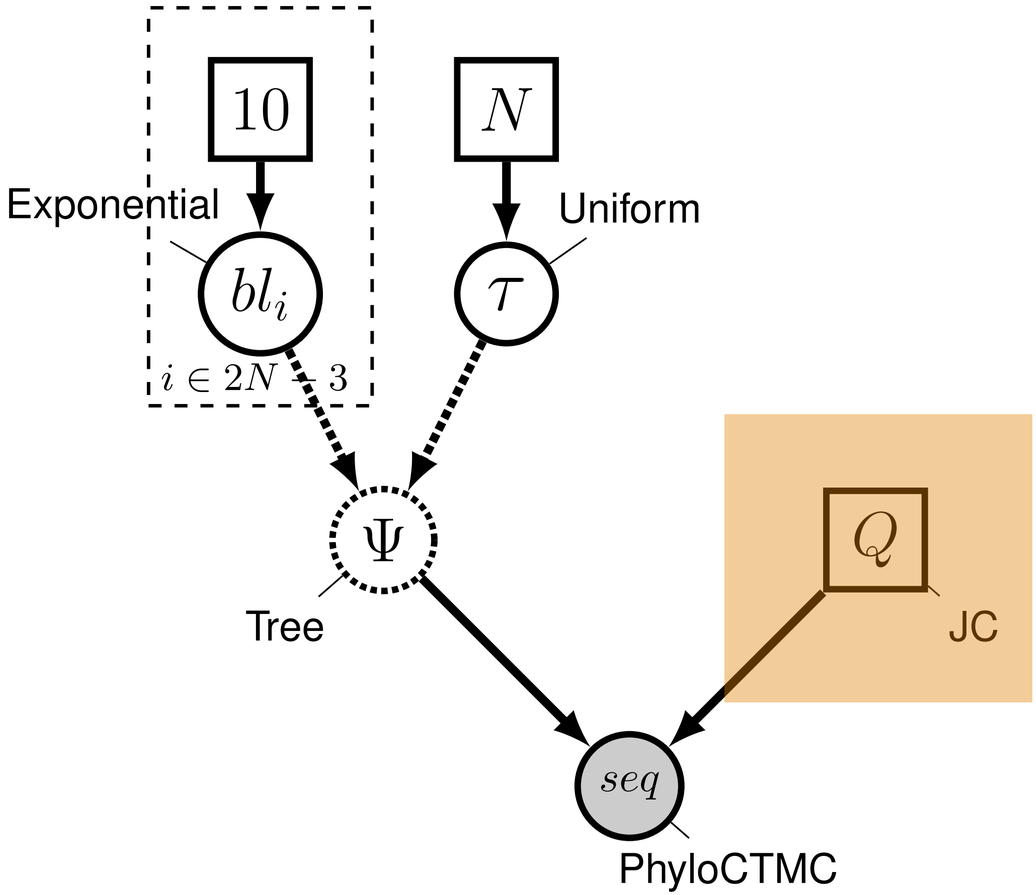
prior on the branch lengths



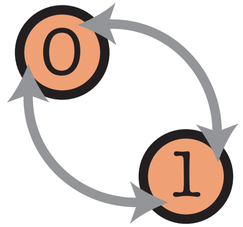


we can combine the topology and branch lengths

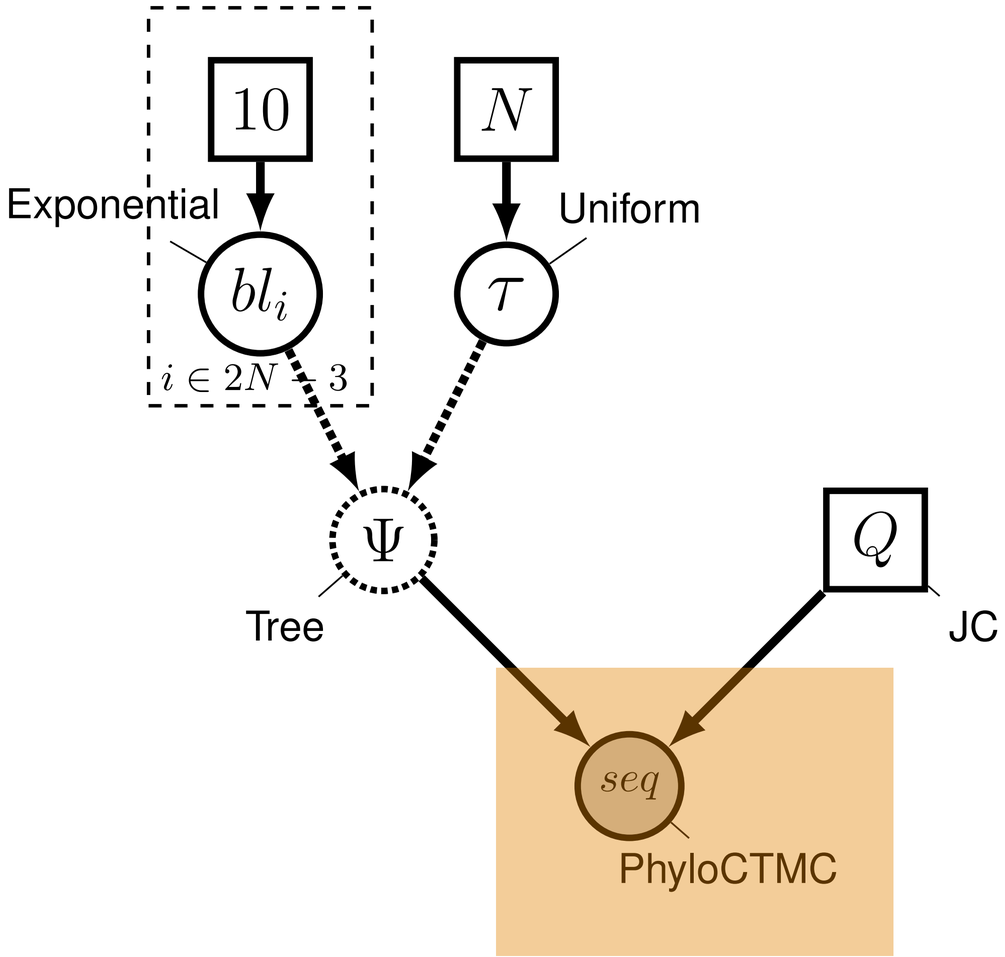




Substitution model

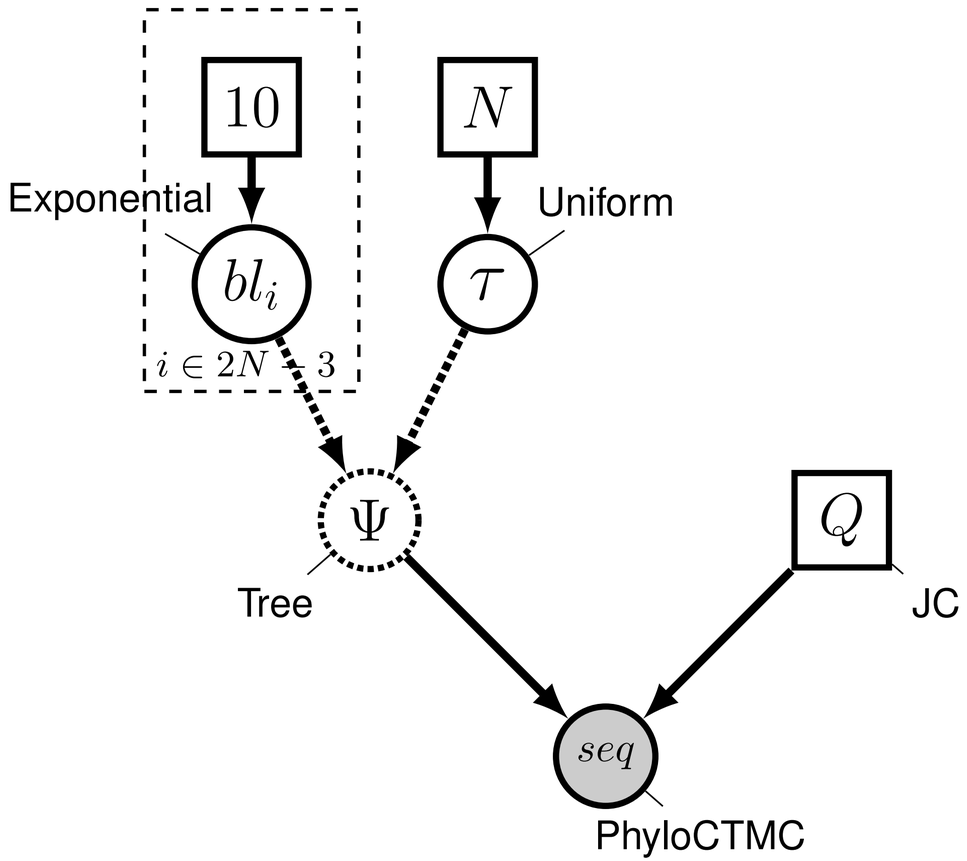


$

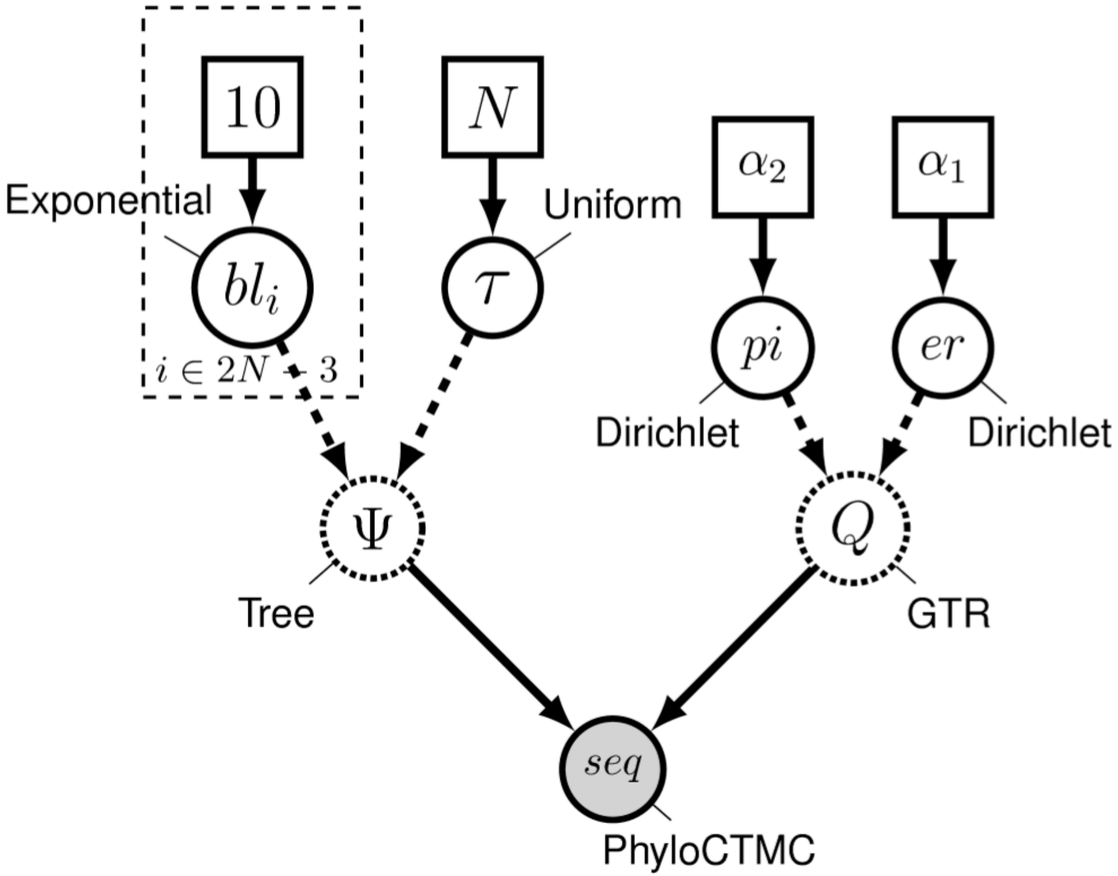


Observed data

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




```
for(i in 1:n_branches) {
  bl[i] ~ dnExp(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] ~ dnExp(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 := fnJC(er, pi)

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

seq.clamp(data)
```

Exercise

Further reading



Phylogenetic inference using RevBayes
Höhna et al. (2017)



The RevBayes tutorial pages contain a lot of useful details
revbayes.github.io