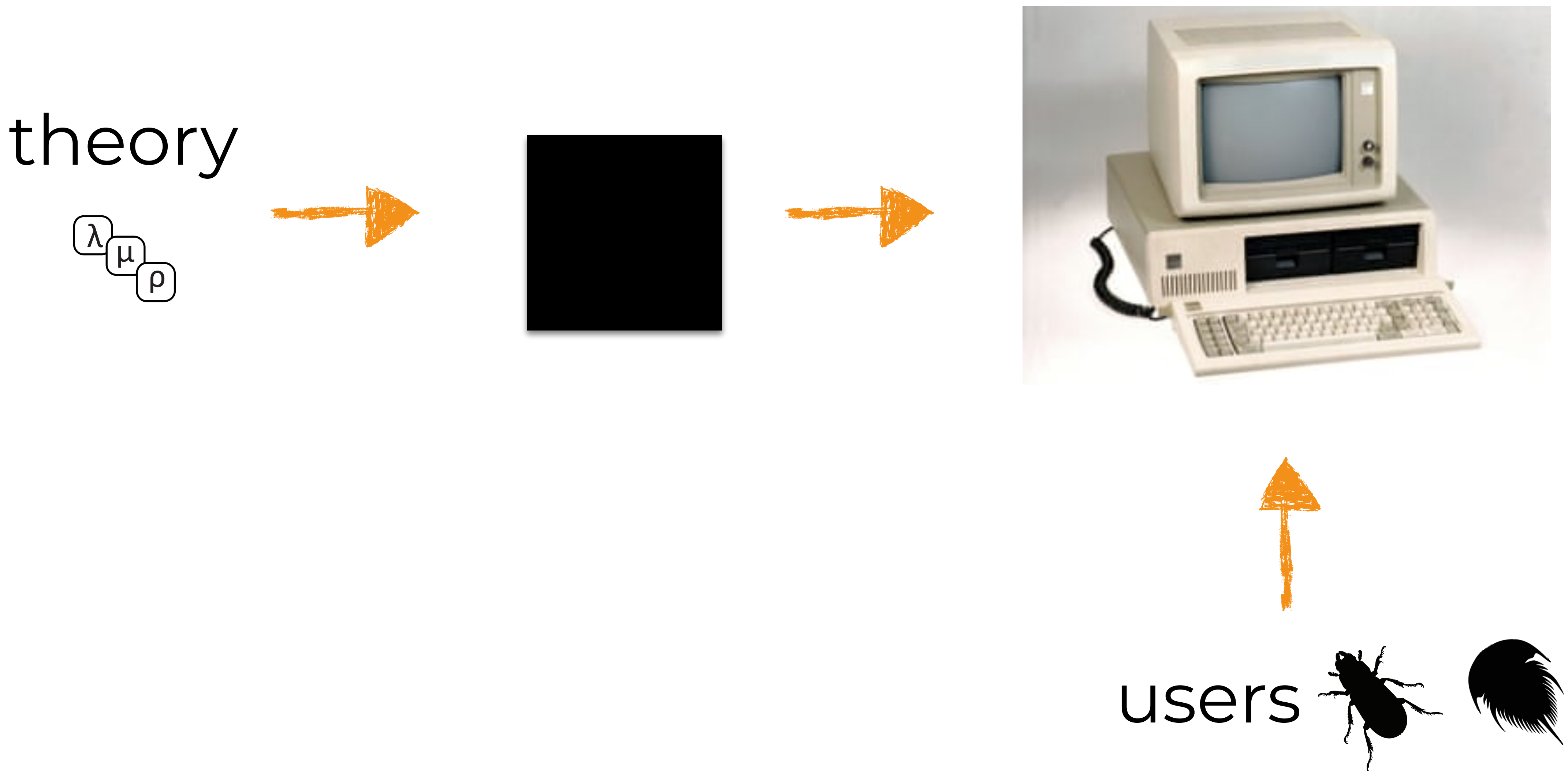
An introduction to graphical models and Bayesian phylogenetics using RevBayes

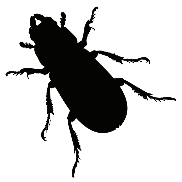
Phylogenetic inference — the old way



Phylogenetic inference — a better way?

theory

λ μ ρ

users  

Aims for RevBayes

General and flexible model specification

- Availability of (common) models
- Extendability

Easy to learn

- Well structured model specification
- Explicit models
- Documentation, examples and tutorials

Computational efficiency

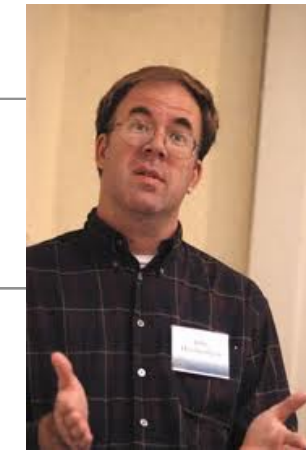
- Fast likelihood calculators
- Efficient (MCMC) algorithms

Höhna et al. 2016. Sys Bio

The RevBayes team

+ many others

Sebastian Höhna



John Huelsenbeck

Fredrik Ronquist

Tracy Heath

Michael Landis

Will Pett

April Wright

Jeremy Brown

Bastien Boussau

Brian Moore

Mike May

Will Freyman

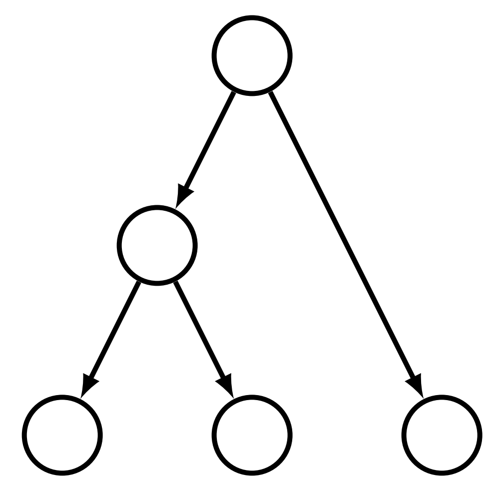
Wade Dismukes

Nicolas Lartillot

Graphical models

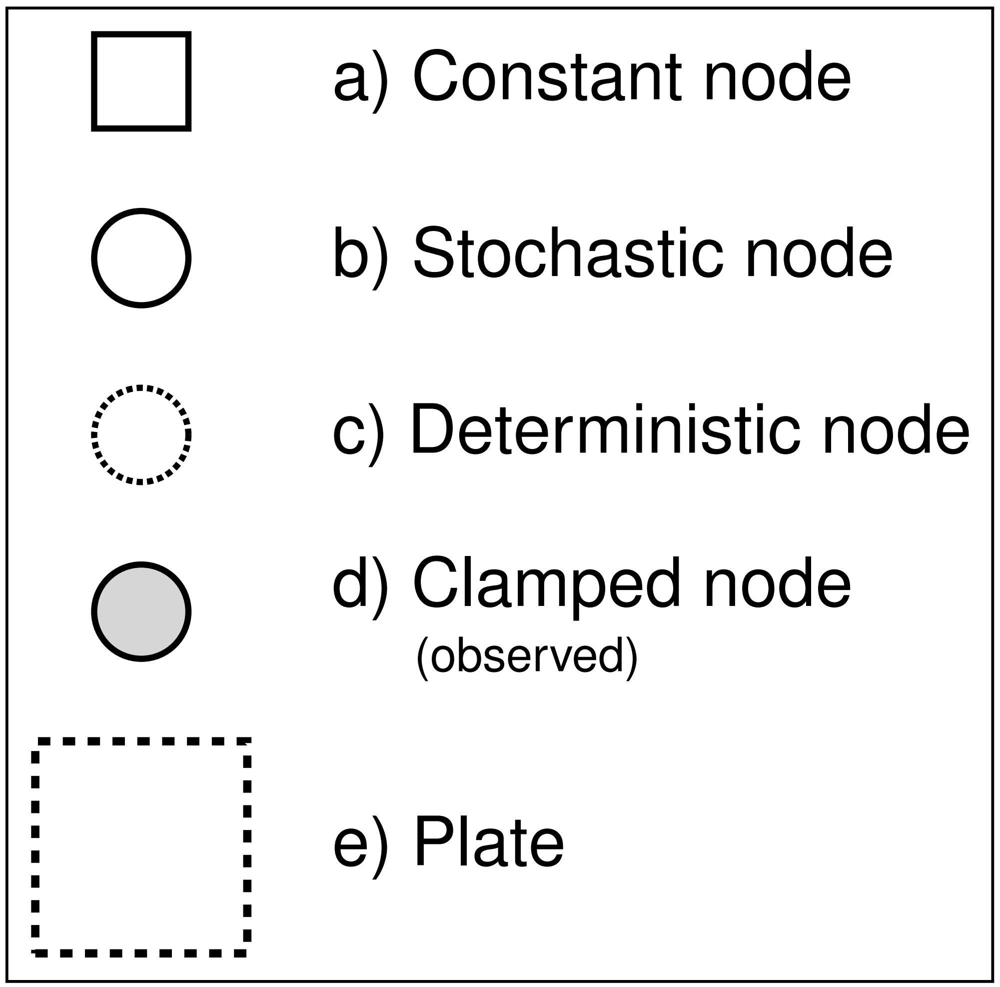
Graphical models provide tools for visually and computationally representing complex, parameter-rich probabilistic models

We can depict the conditional dependence structure of various parameters and other random variables



Höhna et al. 2014. Sys Bio

Graphical models — types of variables (nodes)



a) fixed-value variables

b) random variables that depend on other variables

c) variables determined by a specific function applied to another variable (transformations)

d) observed stochastic variables (data)

e) replication over a set of variables

Höhna et al. 2014. Sys Bio

Specifying graphical models using the Rev syntax

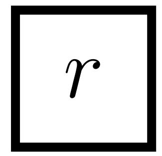
Table 1: `Rev` assignment operators, clamp function, and plate/loop syntax.

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

a) $\boxed{r}$

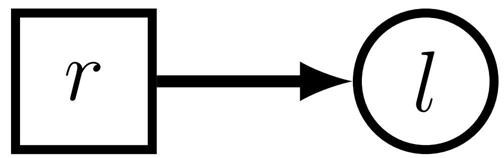
```
# constant node
r <- 10
```

a)

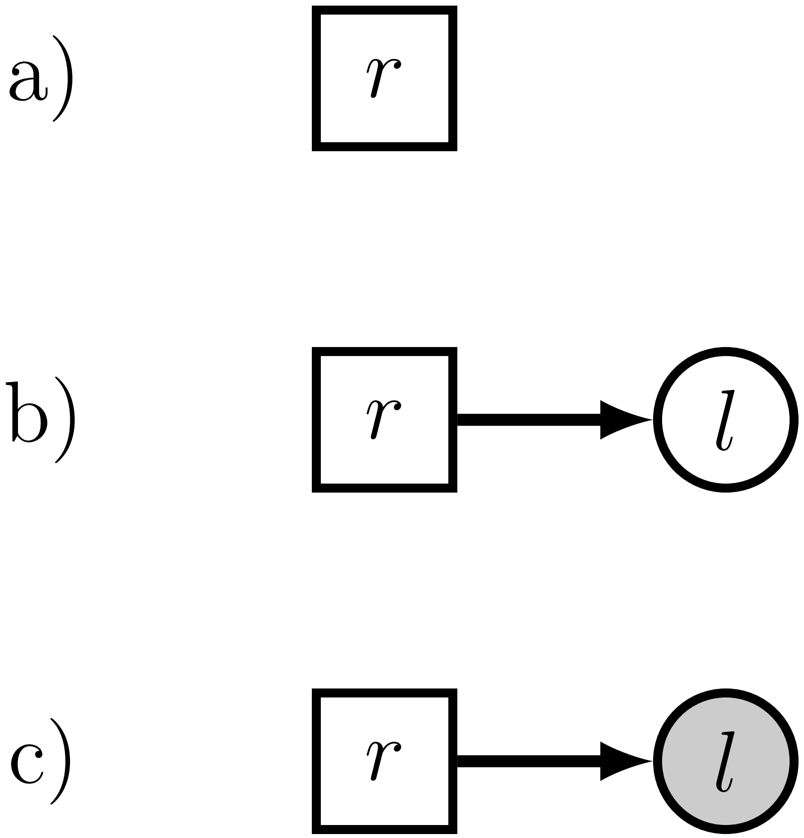



```
# constant node
r <- 10
```

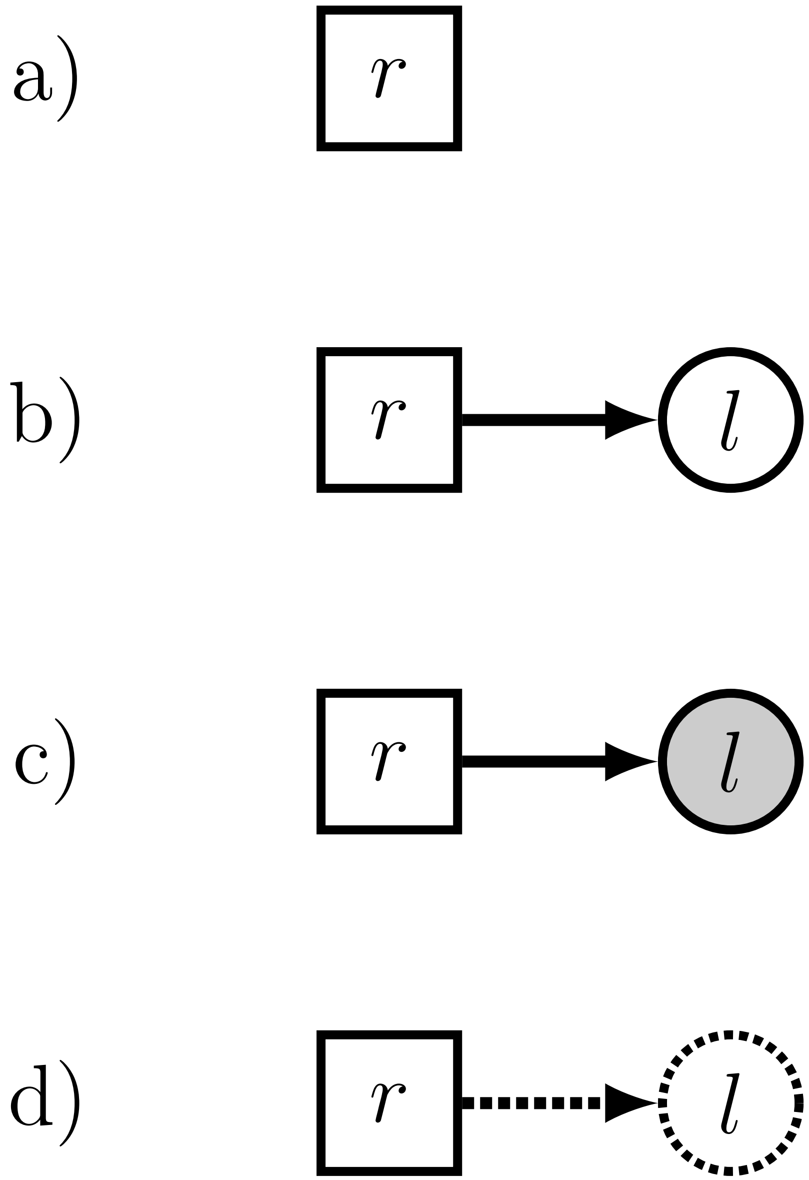
b)



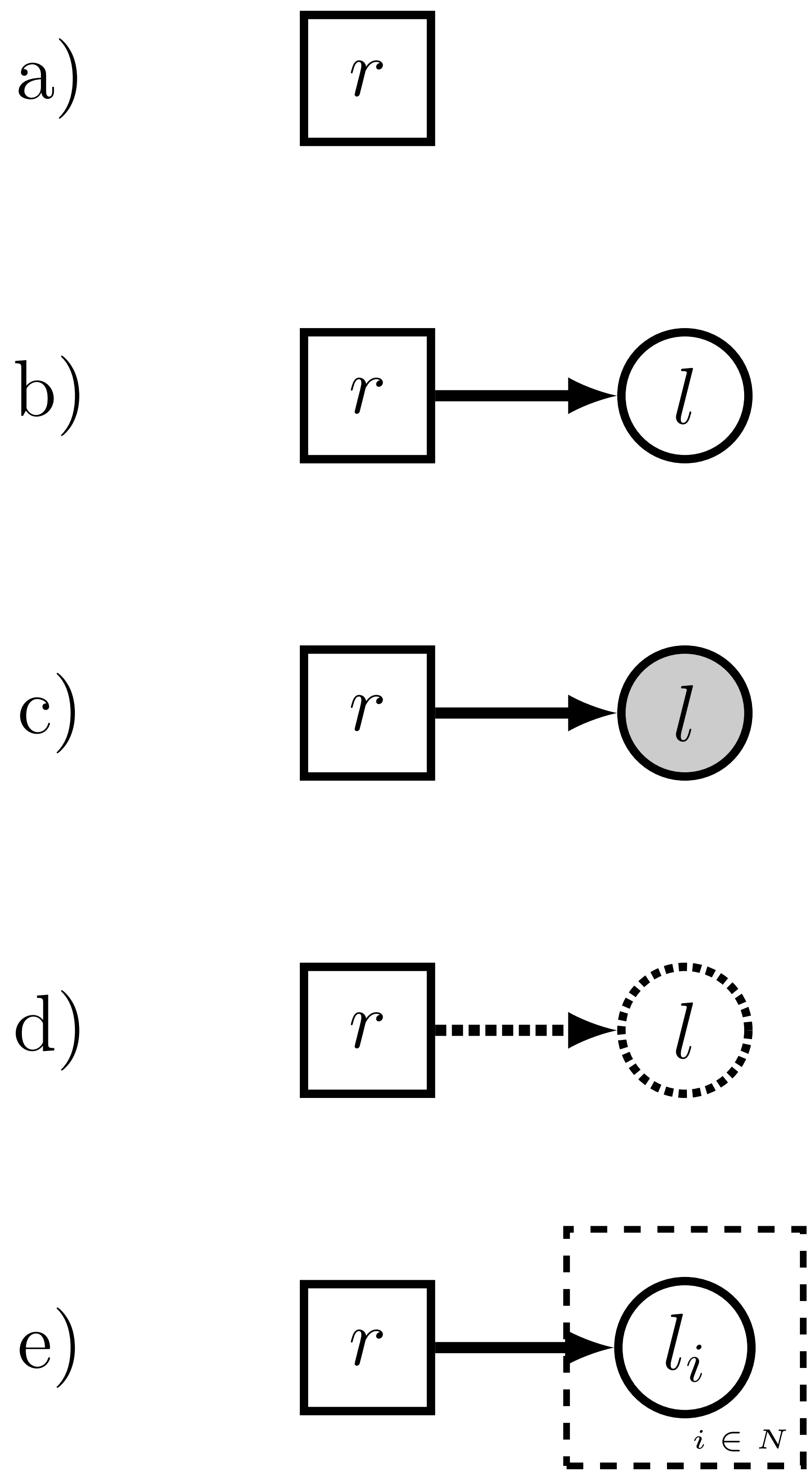

```
# stochastic node
l ~ dnExp(r)
```




```
# constant node
r <- 10


# stochastic node
l ~ dnExp(r)


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

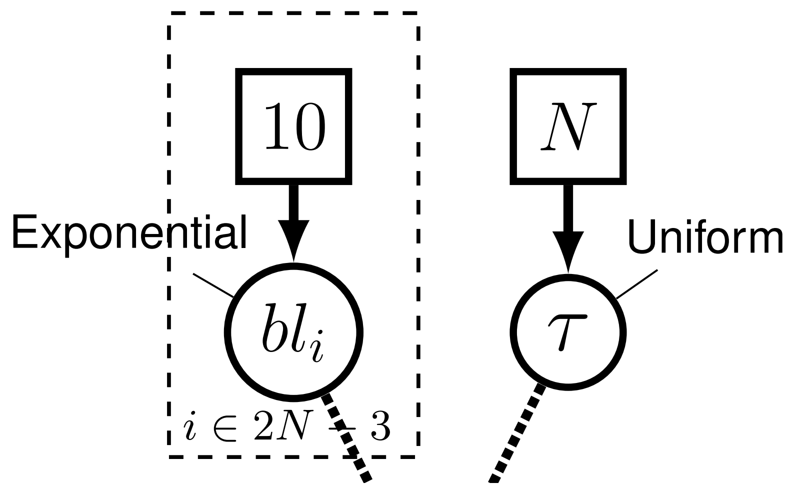



```
# constant node
r <- 10

# stochastic node
l ~ dnExp(r)

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

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



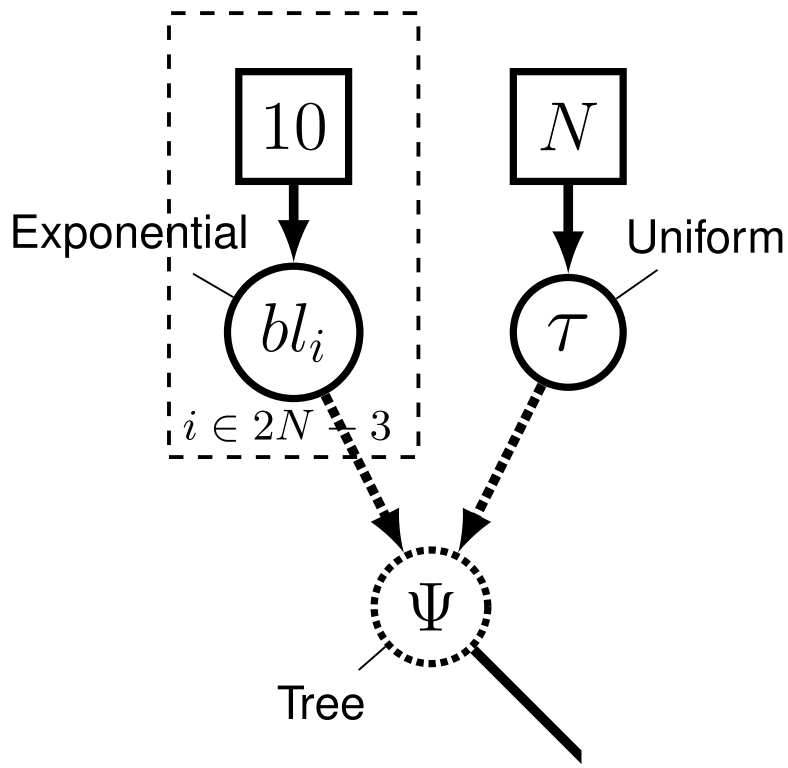

```
# constant node
r <- 10

# stochastic node
l ~ dnExp(r)

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

# deterministic node
l := exp(r)

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

Tree inference using morphology in RevBayes



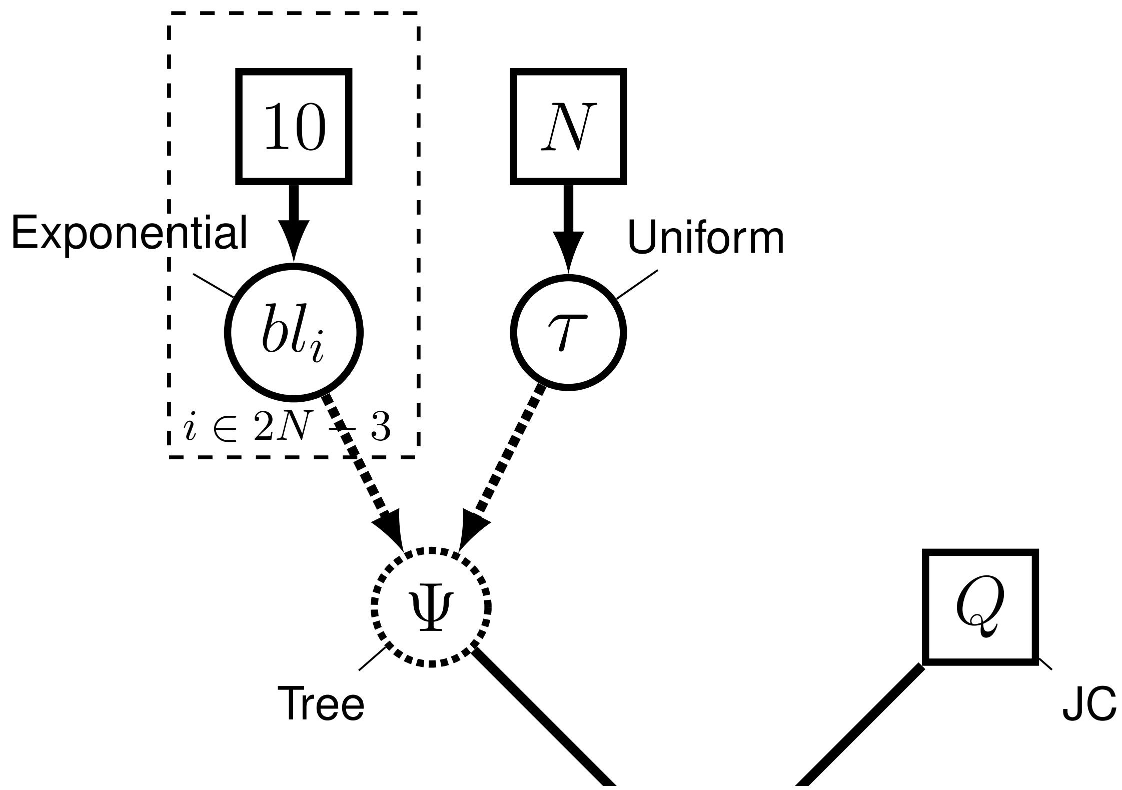
- The tree prior

Tree inference using morphology in RevBayes



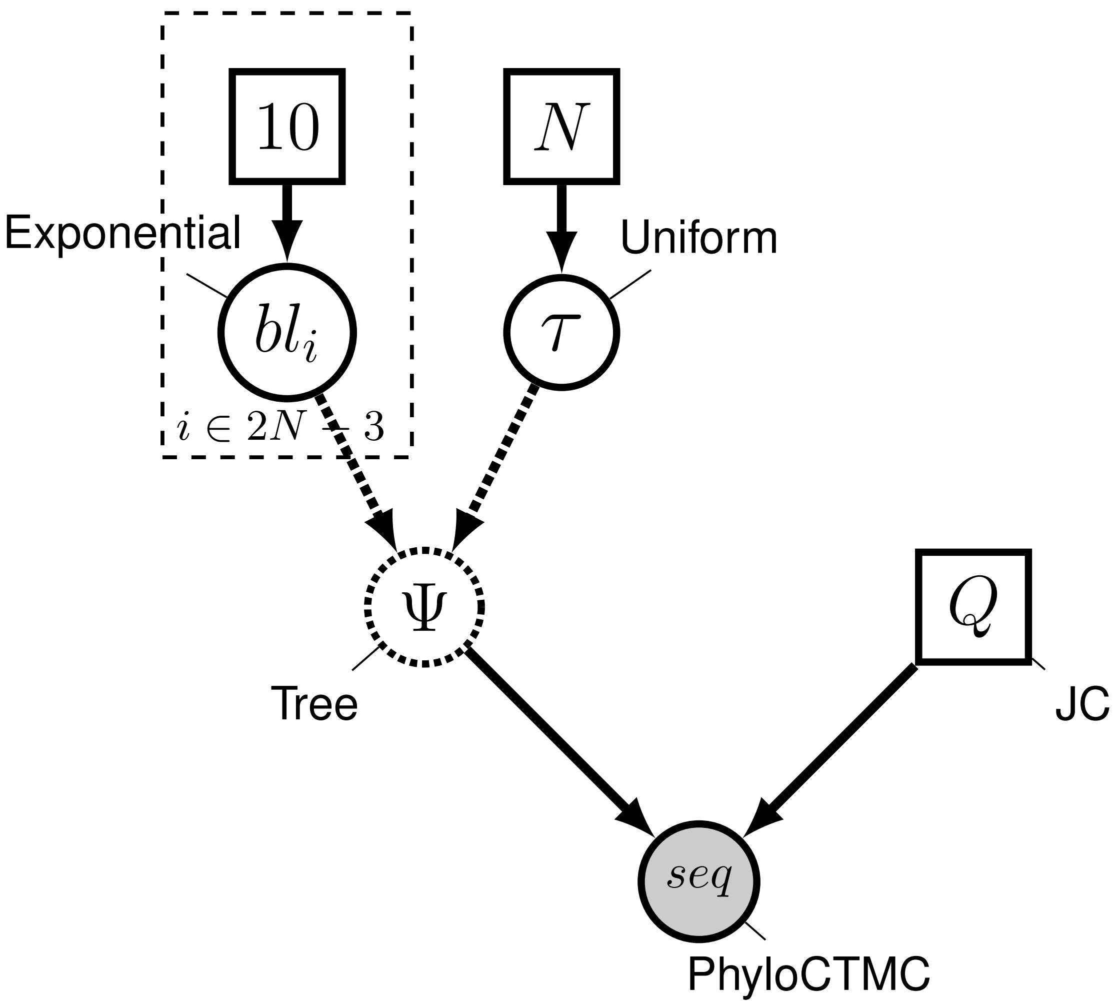
- The tree is a stochastic variable (node)

Tree inference using morphology in RevBayes



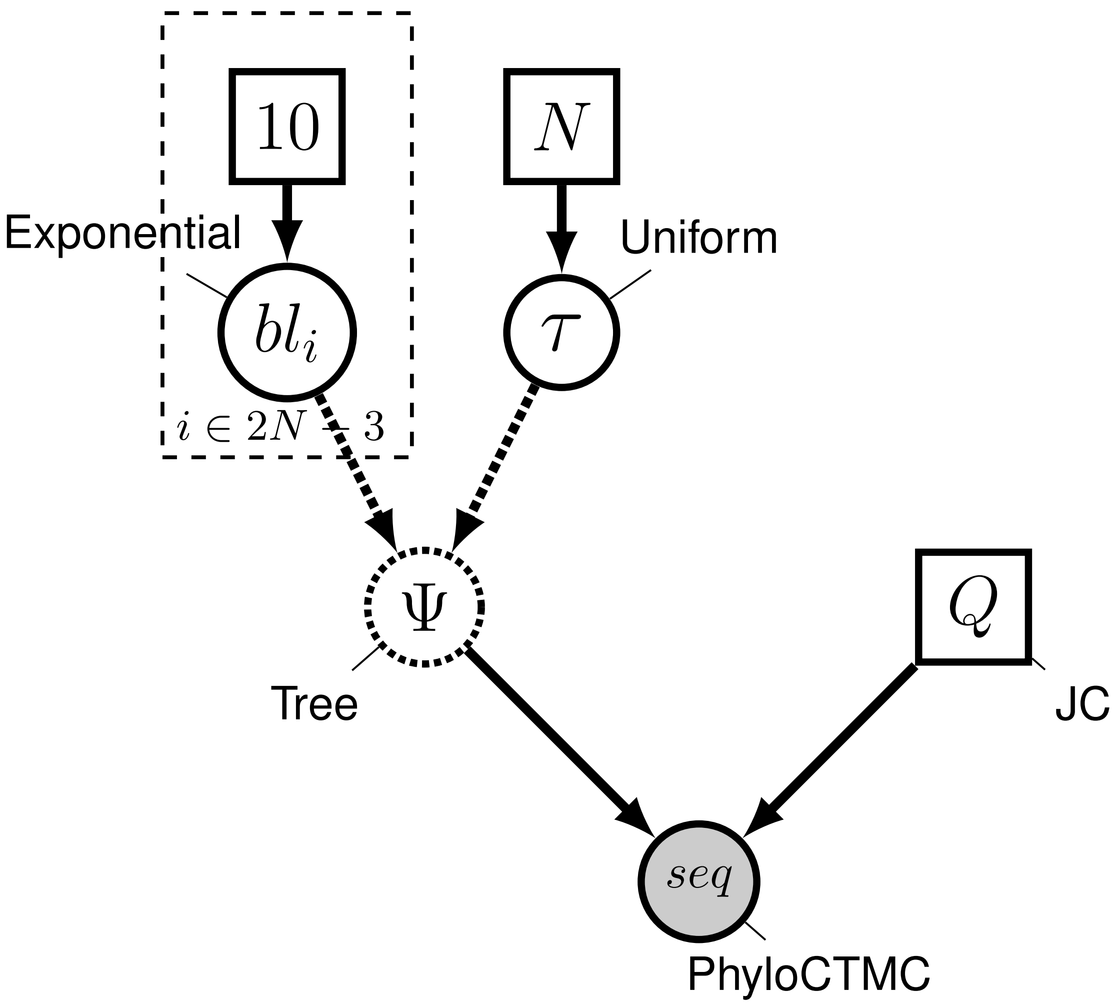
- Model of morphological evolution

Tree inference using morphology in RevBayes



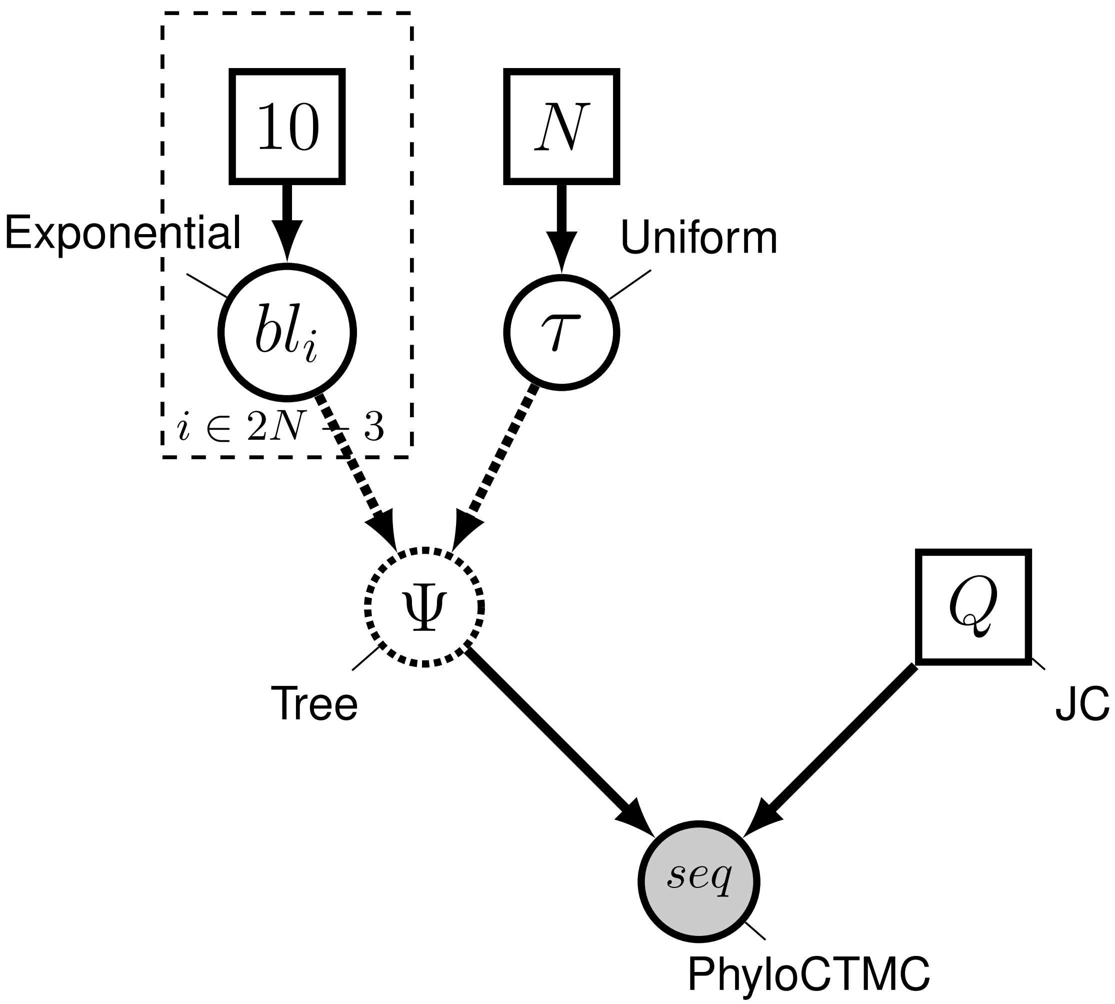
- The data is an observed stochastic variable

Tree inference using morphology in RevBayes




```
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 )
```

Tree inference using morphology in RevBayes




```
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 )
```

Correcting for ascertainment bias