RevBayes: Bayesian phylogenetics using probabilistic graphical models and an interpreted model specification language

SEBASTIAN HÖHNA

DEPARTMENT OF EVOLUTION & ECOLOGY, UC DAVIS
DEPARTMENT OF STATISTICS, UC BERKELEY

RevBayes

- Why a new software???
  - No inherited problems
  - Extendable software
  - We need a general a flexible modeling framework
  - It needs to be fast
  - We need to be comfortable to develop in it

➡ **A new software provides more opportunities but is also more work.**

Ambitions

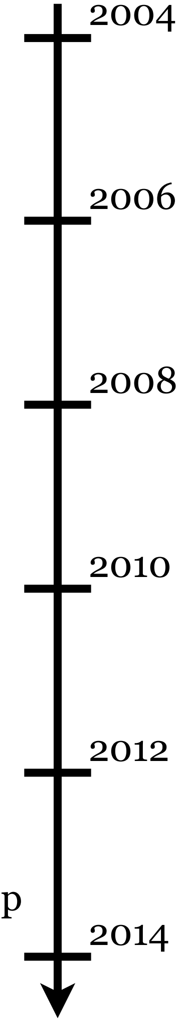
Bastien Boussau

Ambitions

Your wildest dreams will come true!

Fredrik Ronquist

History



2004, Work starts with NIH grant to J.P. Huelsenbeck, F. Ronquist and B. Larget. P. van der Mark starts PostDoc.

May 2005, Release of MrBayes 3.1

Spring 2005, Meeting in San Diego about goals and key concepts (J.P. Huelsenbeck, F. Ronquist, B. Larget. P. van der Mark and Donald Simon).
Codename: MrBayes 4

Summer 2005 and early 2006, next meetings (now including M. Suchard). Focus shifts to R-like model specification

April 2009, S. Höhna starts PhD and joint the project

August 2009, Sourceforge repository created and first lines written.

June 2011, Rewrite of RevBayes

August 2011, Developers workshop in Berkeley.

June 2012, Rewrite of RevBayes

March 2013, Alpha testing during a workshop in Groningen.

August 2014, First official user workshop at NESCent. RevBayes is ready at a beta stage.

Aims for RevBayes

1) General and flexible model specification

   a) Availability of (common) models

   b) Extendability

Aims for RevBayes

1) General and flexible model specification
   - a) Availability of (common) models
   - b) Extendability

2) Easy to learn
   - a) Well structured model specification
   - b) Explicit models
   - c) Documentation, examples and tutorials

Aims for RevBayes

1) General and flexible model specification

   a) Availability of (common) models

   b) Extendability

2) Easy to learn

   a) Well structured model specification

   b) Explicit models

   c) Documentation, examples and tutorials

3) Fast & Efficient

   a) Fast likelihood calculators

   b) Efficient (MCMC) algorithms, e.g., tree proposals

Aims for RevBayes

1) General and flexible model specification

a) Availability of (common) models

b) Extendability

2) Easy to learn

a) Well structured model specification

b) Explicit models

c) Documentation, examples and tutorials

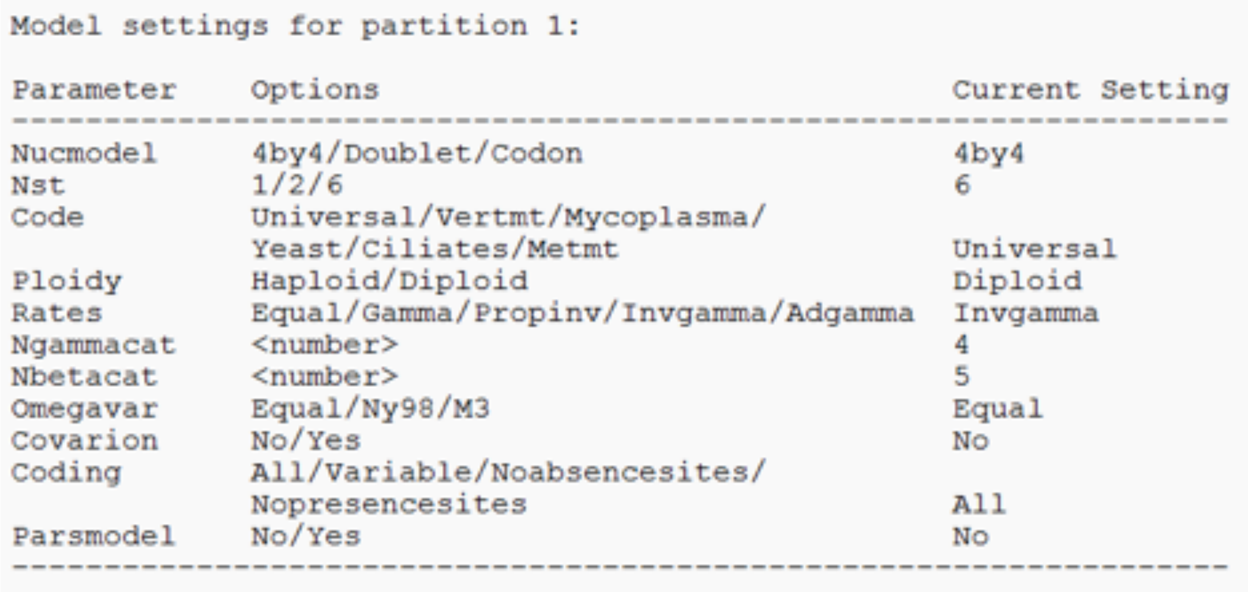
3) Fast & Efficient

a) Fast likelihood calculators

b) Efficient (MCMC) algorithms, e.g., tree proposals

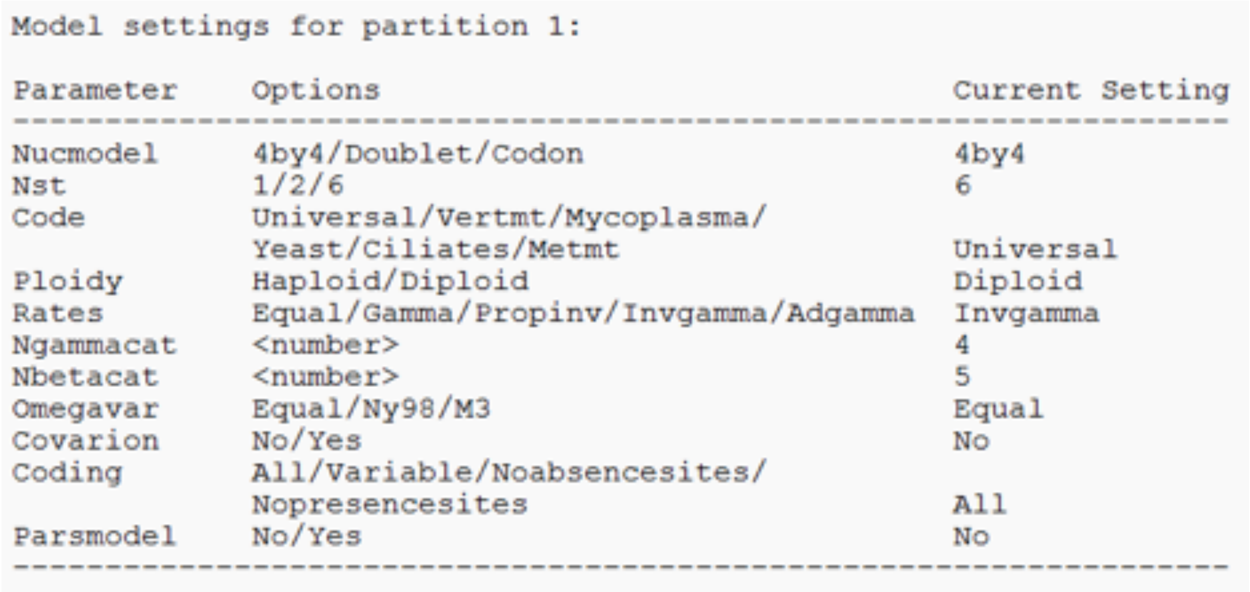
Graphical Models & interpreted interface & C++

Specifying a model in MrBayes

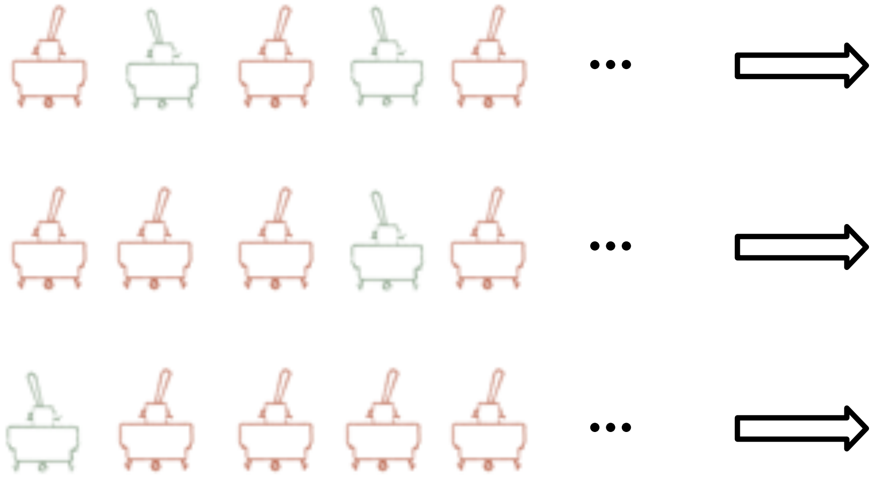


```
Model settings for partition 1:

Parameter    Options                                 Current Setting
--------------------------------------------------------------------
Nucmodel     4by4/Doublet/Codon                      4by4
Nst          1/2/6                                   6
Code         Universal/Vertmt/Mycoplasma/
             Yeast/Ciliates/Metmt                    Universal
Ploidy       Haploid/Diploid                         Diploid
Rates        Equal/Gamma/Propinv/Invgamma/Adgamma    Invgamma
Ngammacat    <number>                                4
Nbetacat     <number>                                5
Omegavar     Equal/Ny98/M3                           Equal
Covarion     No/Yes                                  No
Coding       All/Variable/Noabsencesites/
             Nopresencesites                         All
Parsmodel    No/Yes                                  No
--------------------------------------------------------------------
```

Specifying a model in MrBayes



```
Model settings for partition 1:

Parameter     Options                                      Current Setting
-----------------------------------------------------------------------
Nucmodel      4by4/Doublet/Codon                           4by4
Nst           1/2/6                                        6
Code          Universal/Vertmt/Mycoplasma/
              Yeast/Ciliates/Metmt                         Universal
Ploidy        Haploid/Diploid                              Diploid
Rates         Equal/Gamma/Propinv/Invgamma/Adgamma         Invgamma
Ngammacat     <number>                                     4
Nbetacat      <number>                                     5
Omegavar      Equal/Ny98/M3                                Equal
Covarion      No/Yes                                       No
Coding        All/Variable/Noabsencesites/
              Nopresencesites                              All
Parsmodel     No/Yes                                       No
-----------------------------------------------------------------------
```



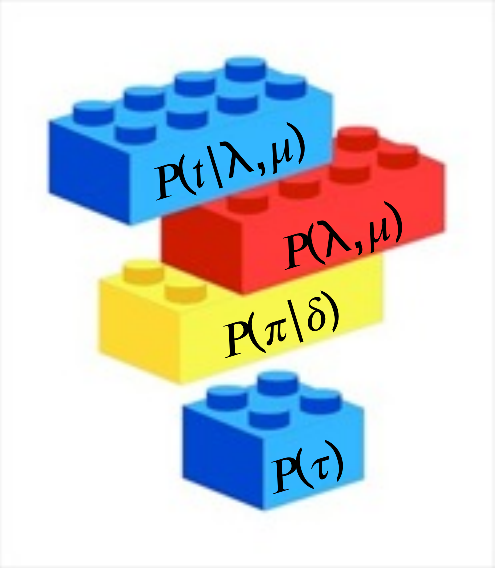
HKY+G - Substitution Model

4-Gamma Categories

Constant-Clock Model

Bottom Up Design

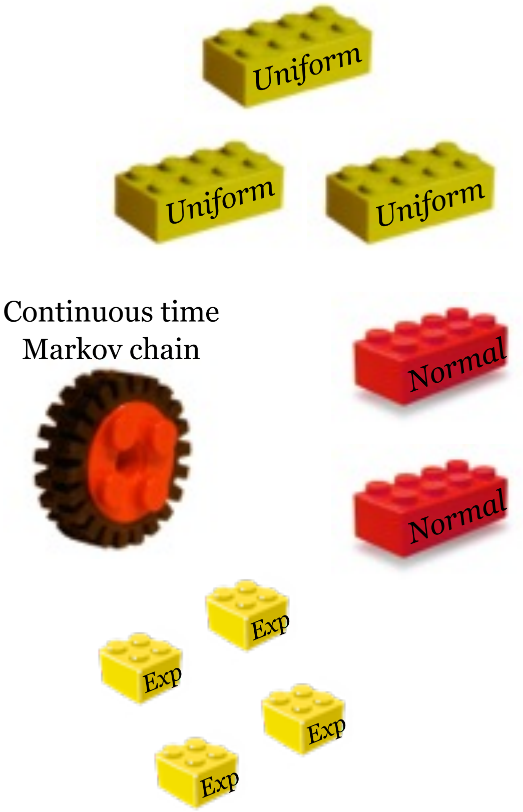
- Standard building blocks:
  - Distributions:
    - Uniform Distribution
    - Normal Distribution
    - Exponential Distribution
    - Gamma Distribution
    - ...
  - Functions:
    - Addition
    - Multiplication
    - Exponentiation
    - ...
- Phylogenetically inspired
  - Distributions
    - Tree priors
    - Substitution processes
    - ...
  - Functions
    - Rate matrix
    - ...



- Assemble model from probability distributions and functions

Bottom Up Design

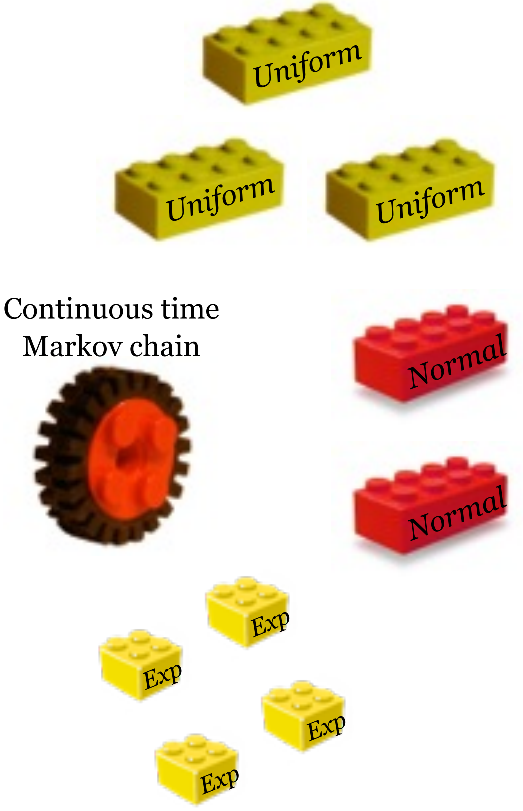
Building blocks:



Assembled model:

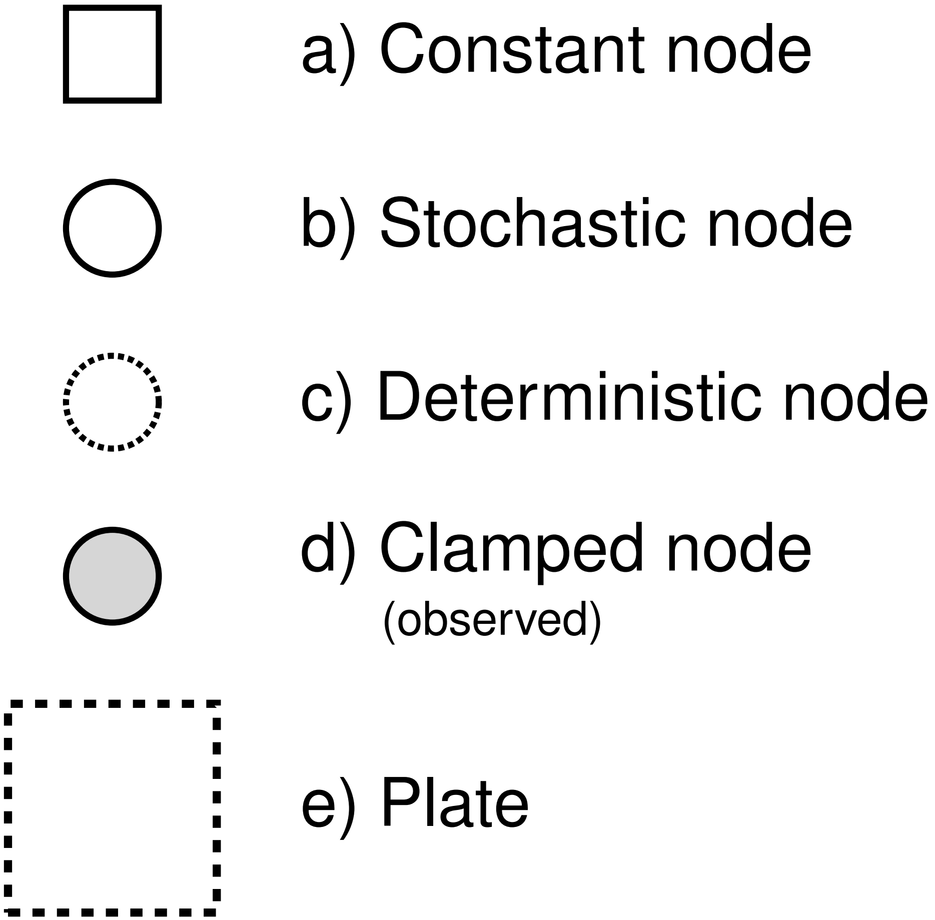
Bottom Up Design

Building blocks:



Assembled model:

Graphical Models Notation



a) Constant node

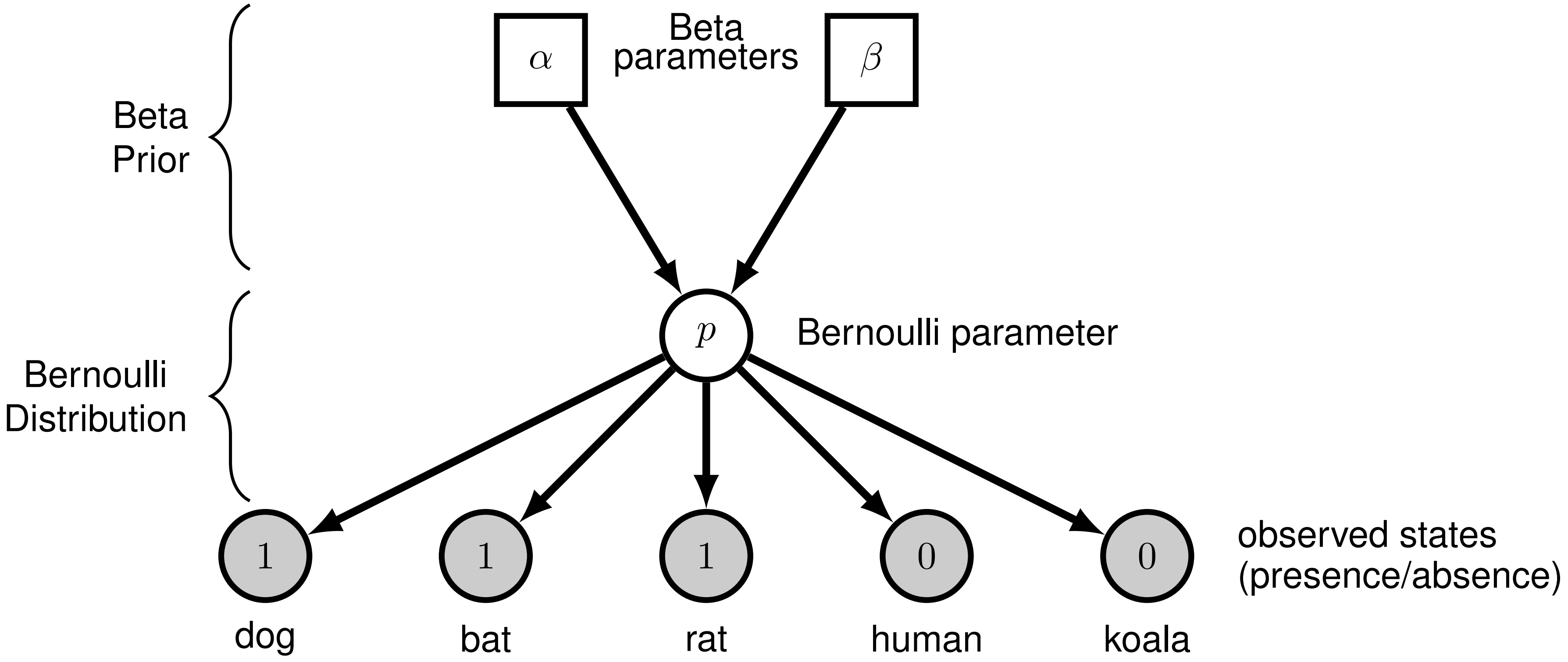
b) Stochastic node

c) Deterministic node

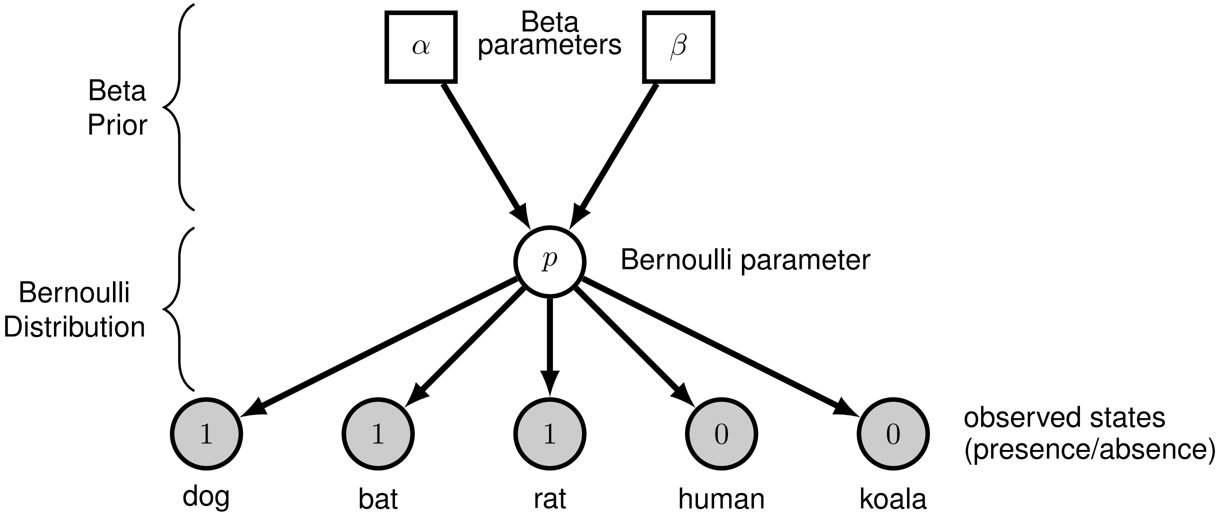
d) Clamped node
(observed)

e) Plate

A simple graphical model



A simple graphical model



Beta Prior

Bernoulli Distribution

α

Beta parameters

β

p

Bernoulli parameter

1 dog

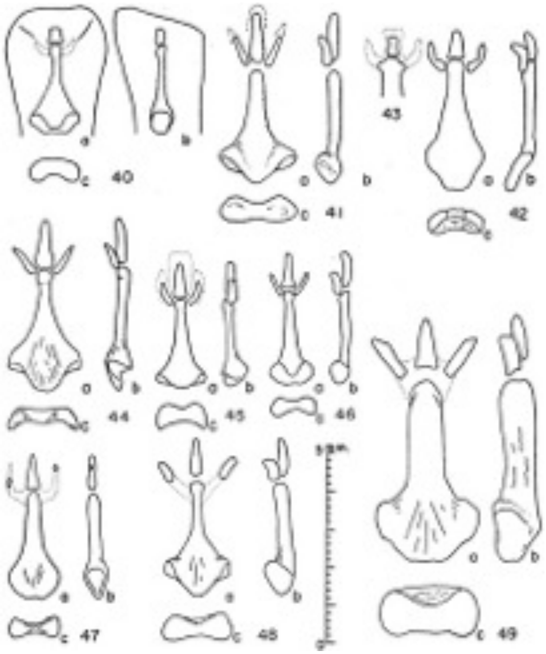
1 bat

1 rat

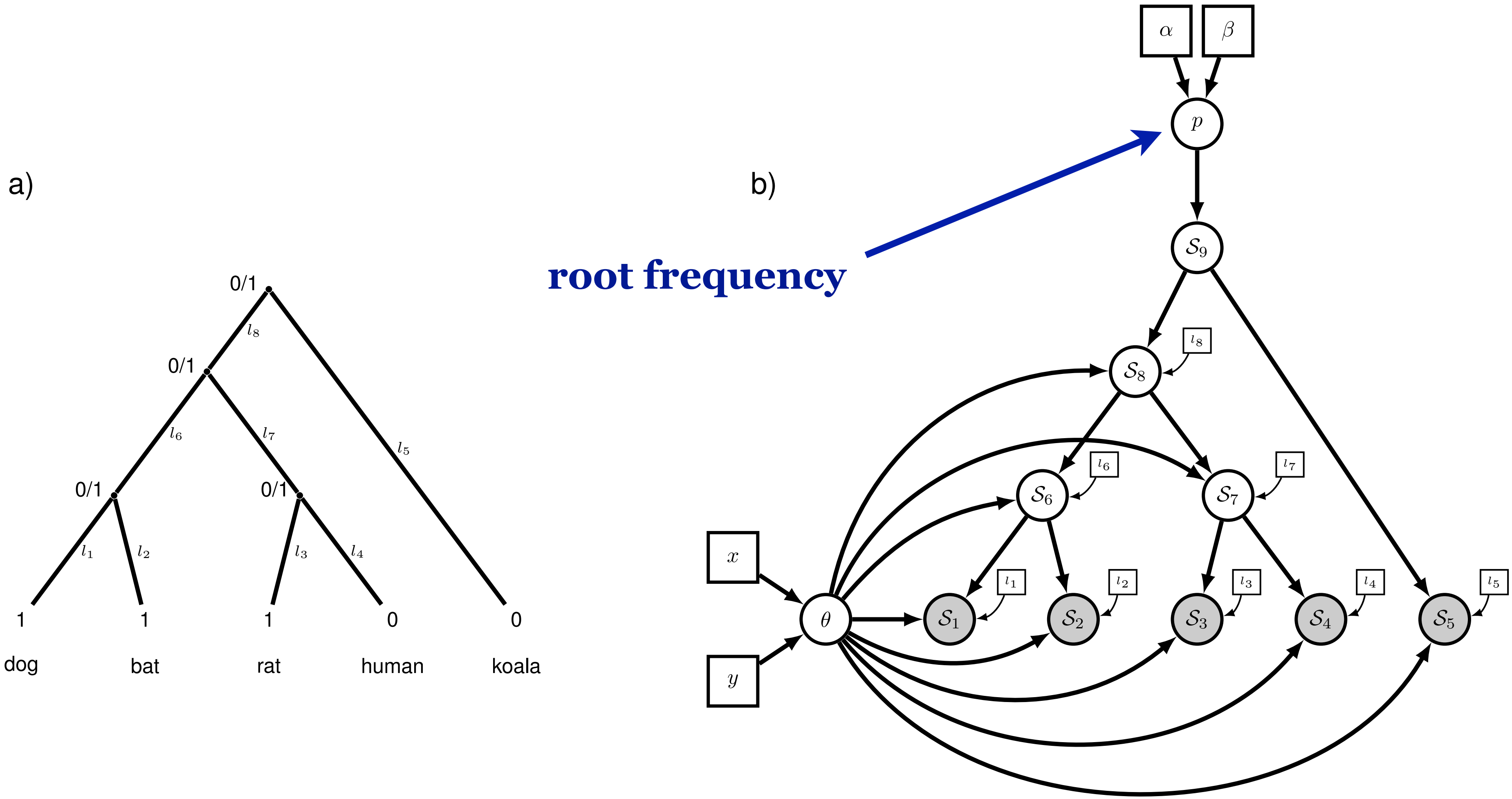
0 human

0 koala

observed states (presence/absence)

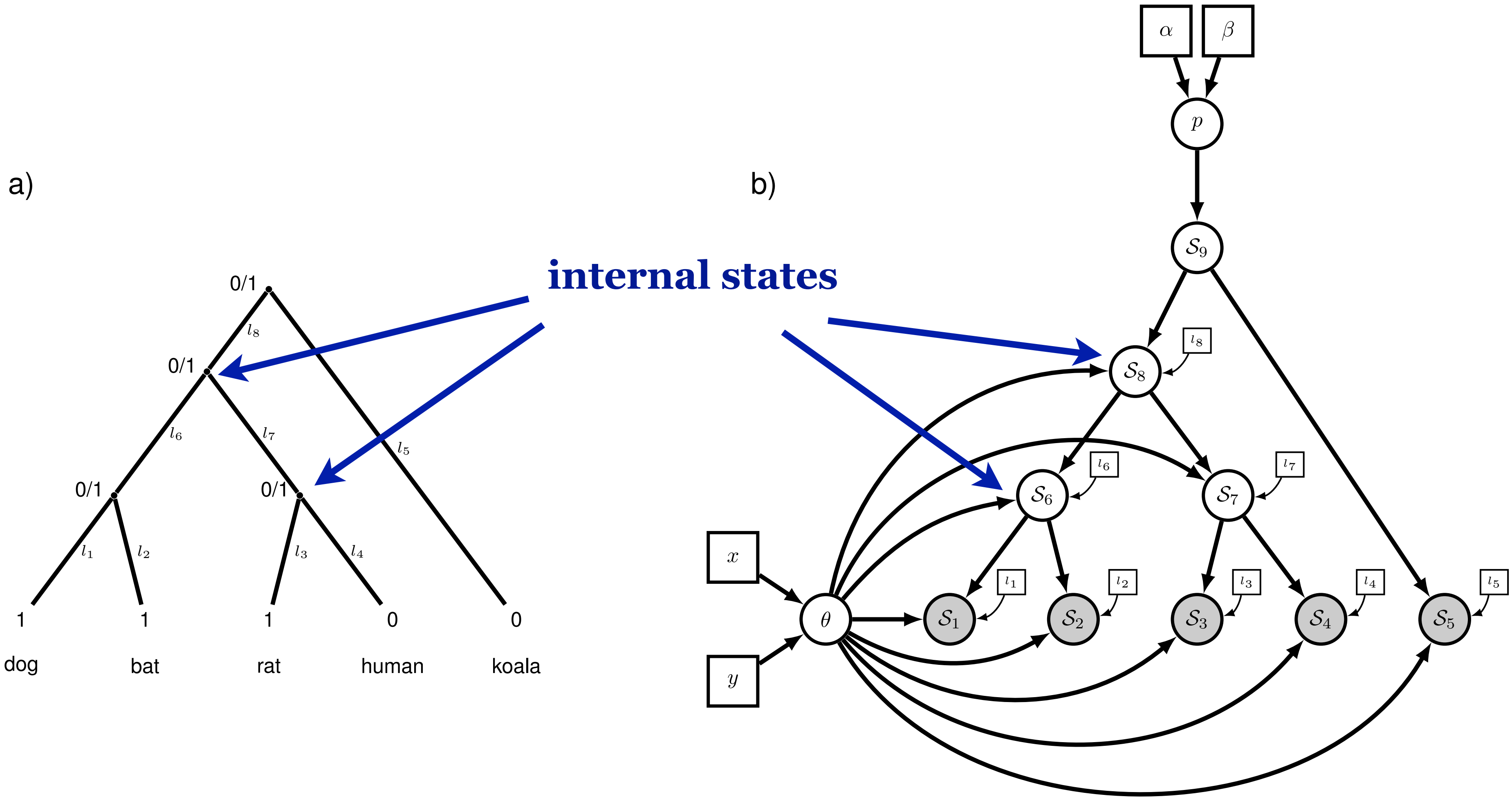


A graphical tree model

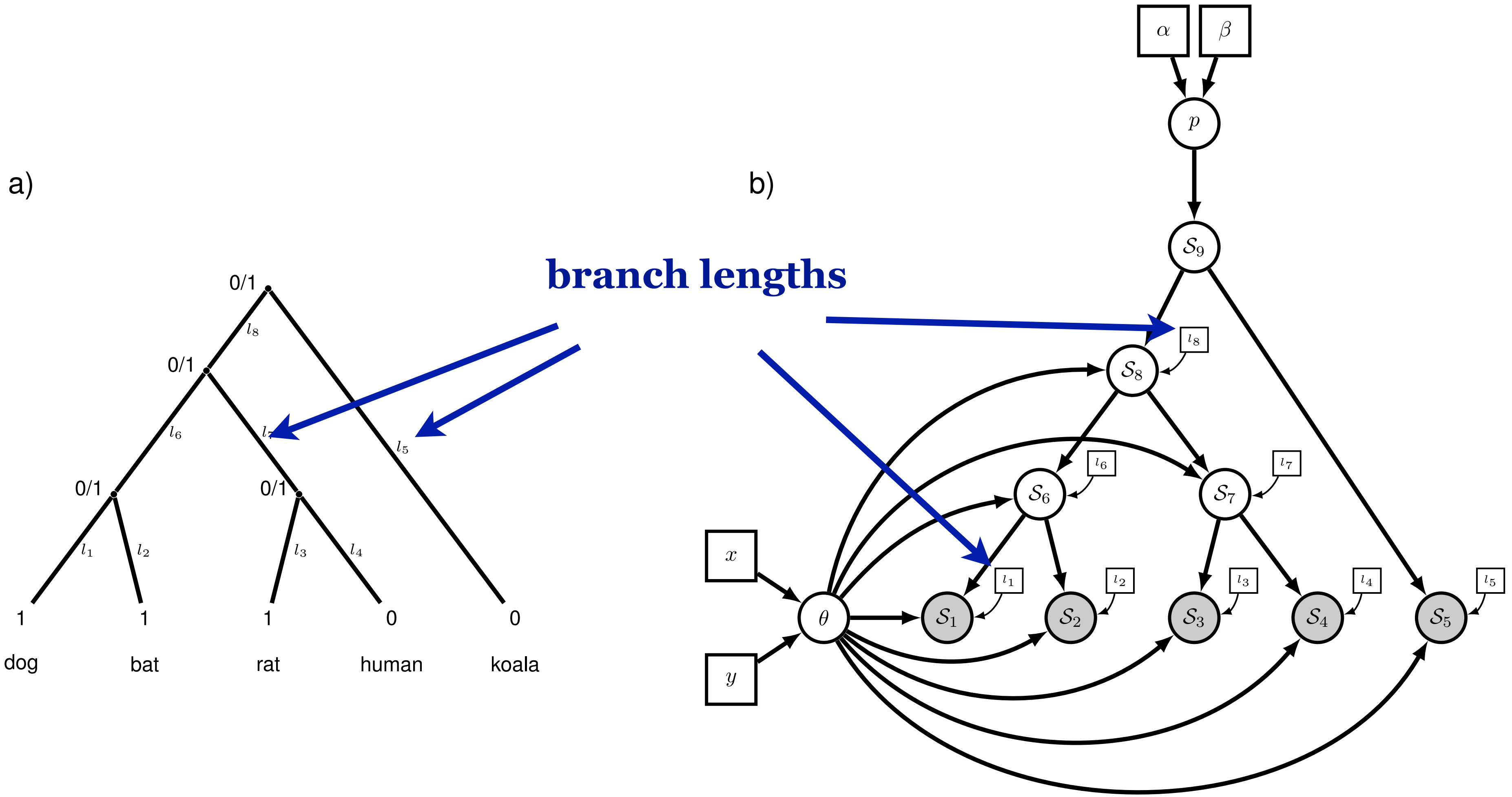


root frequency

A graphical tree model

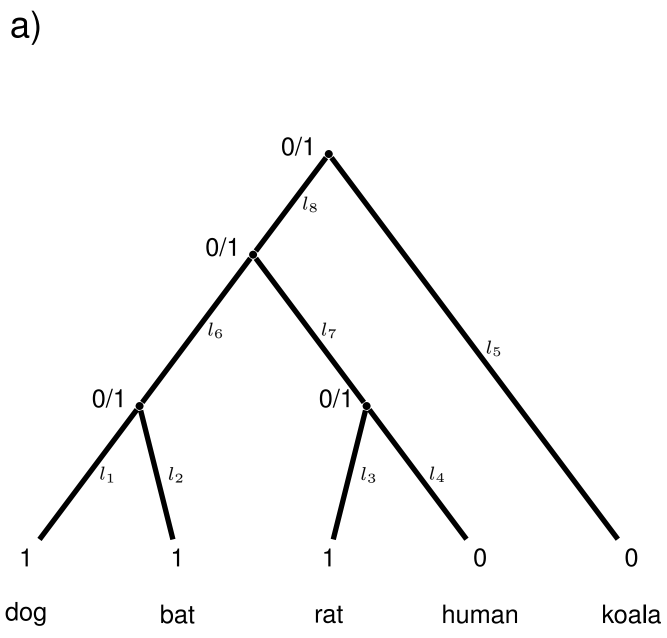

internal states

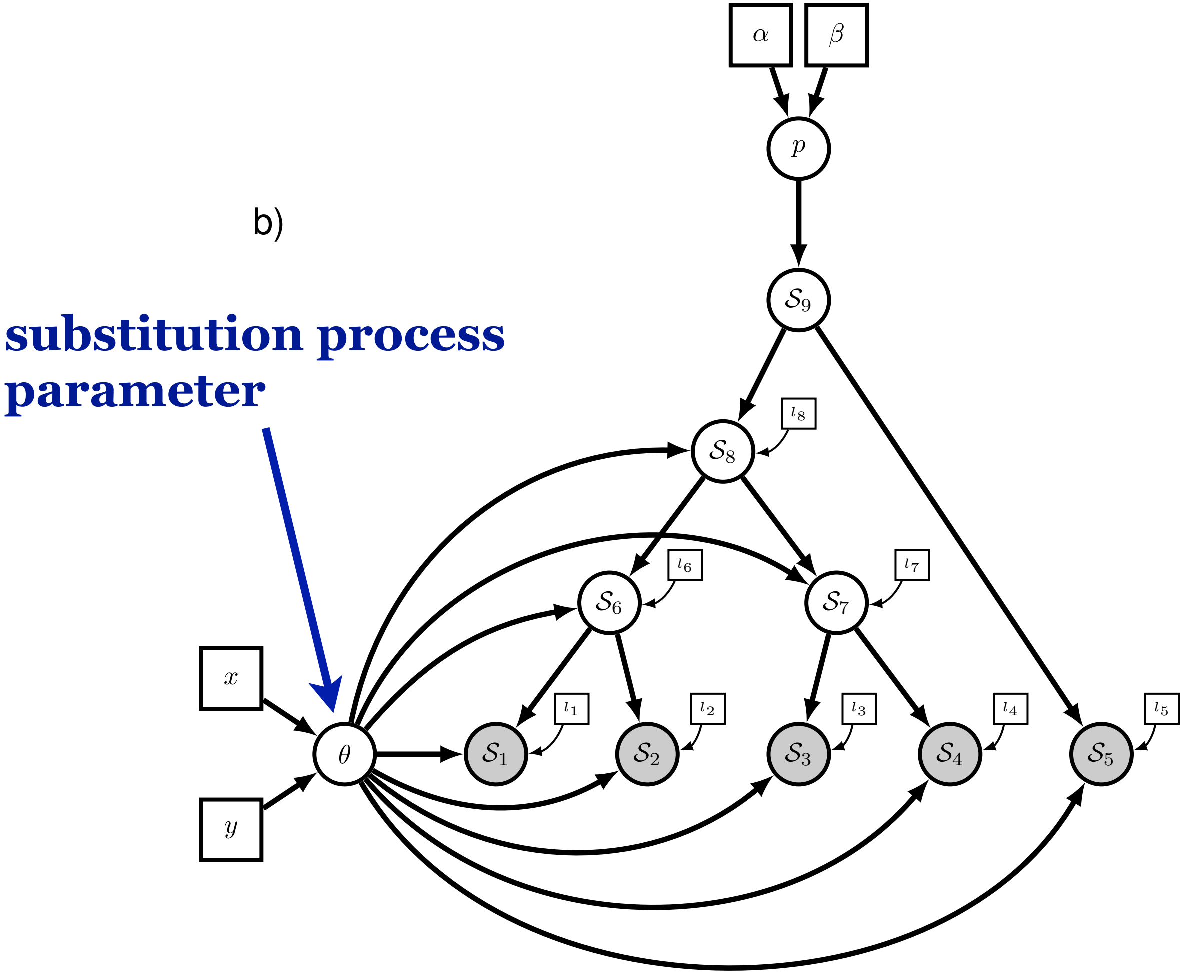
A graphical tree model



branch lengths

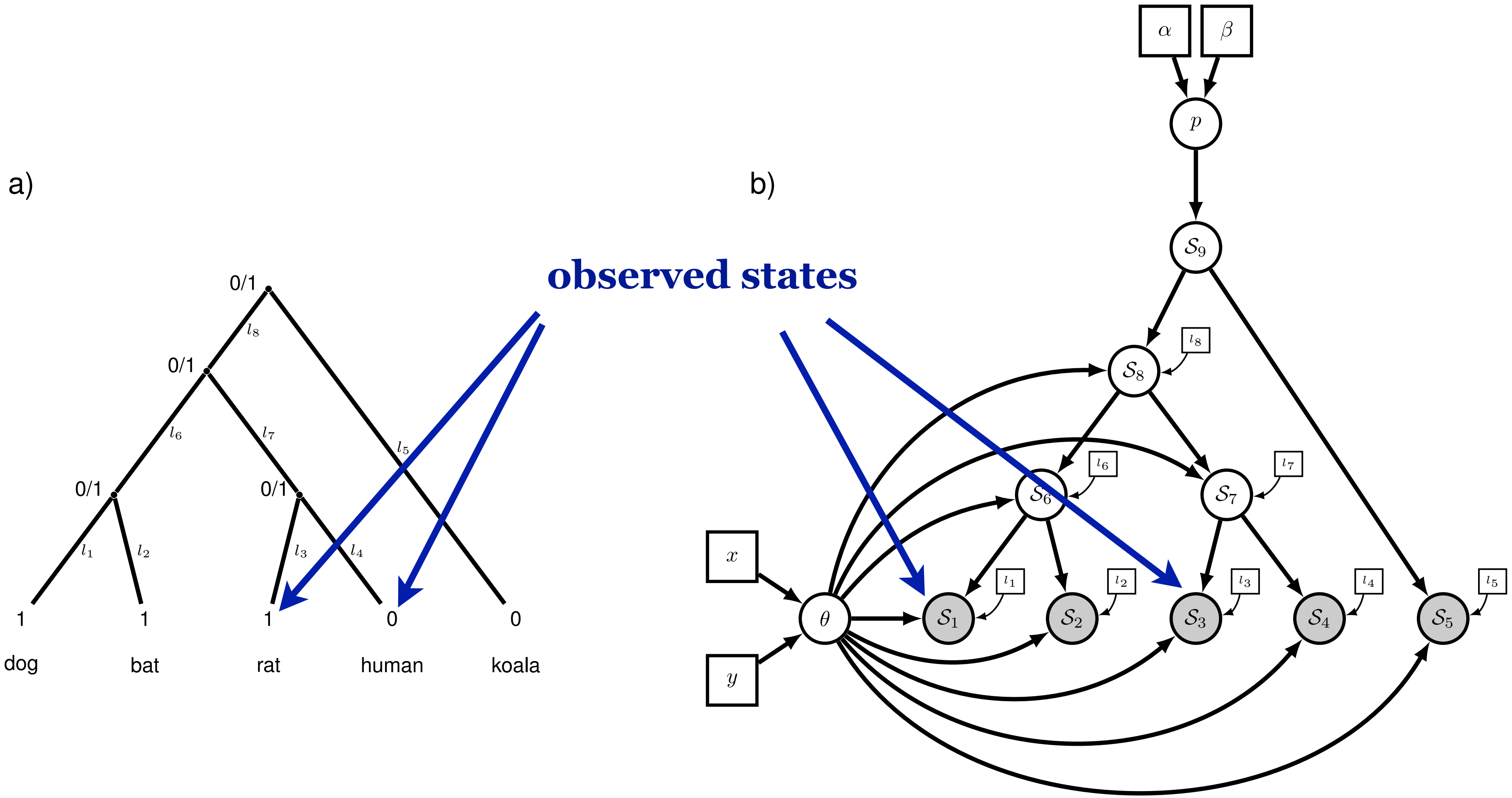
A graphical tree model





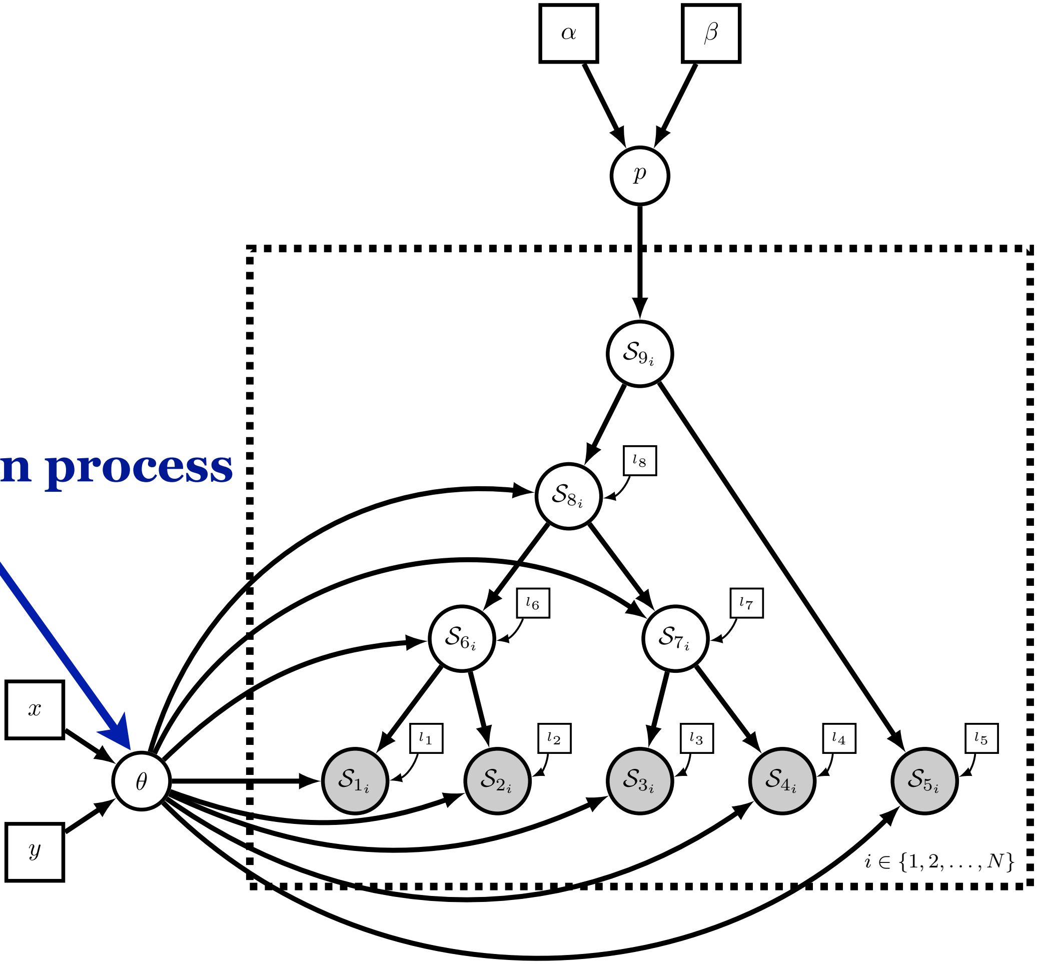
substitution process parameter

A graphical tree model



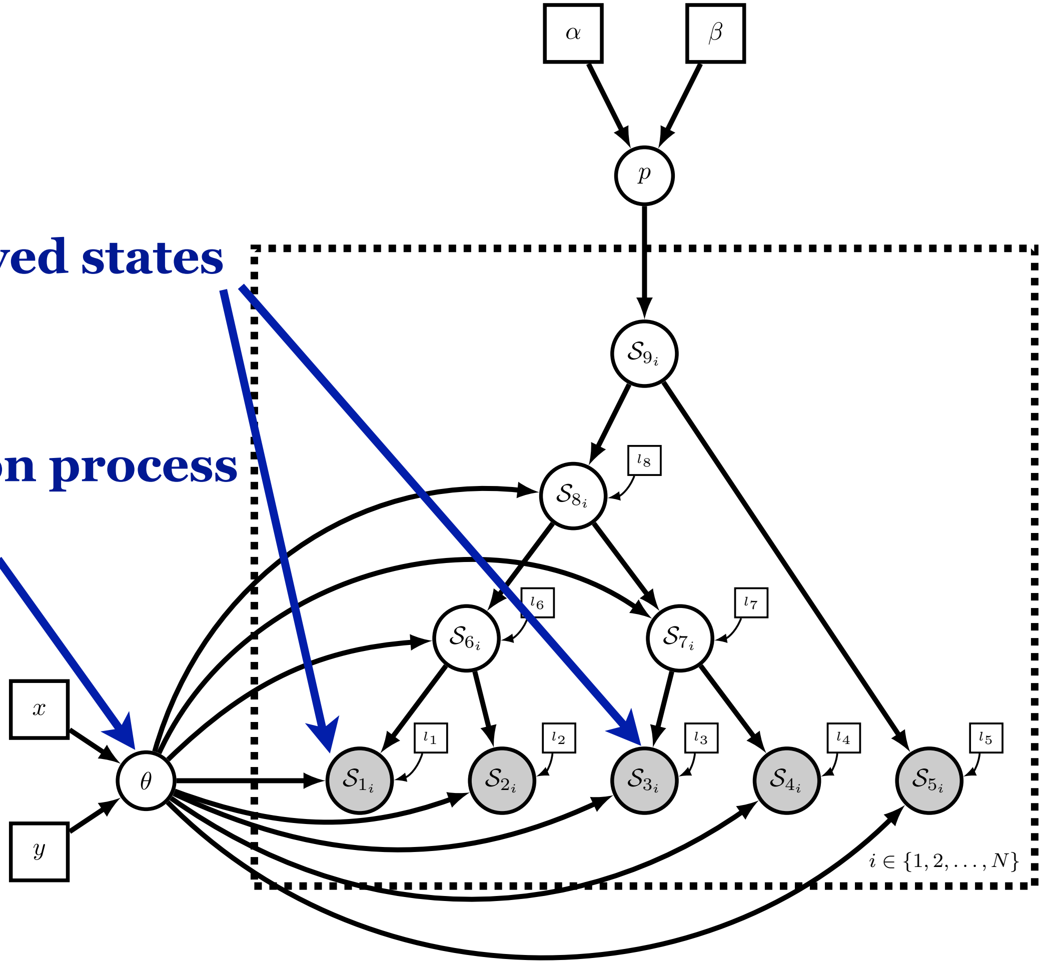
observed states

A graphical model for discrete characters



substitution process parameter

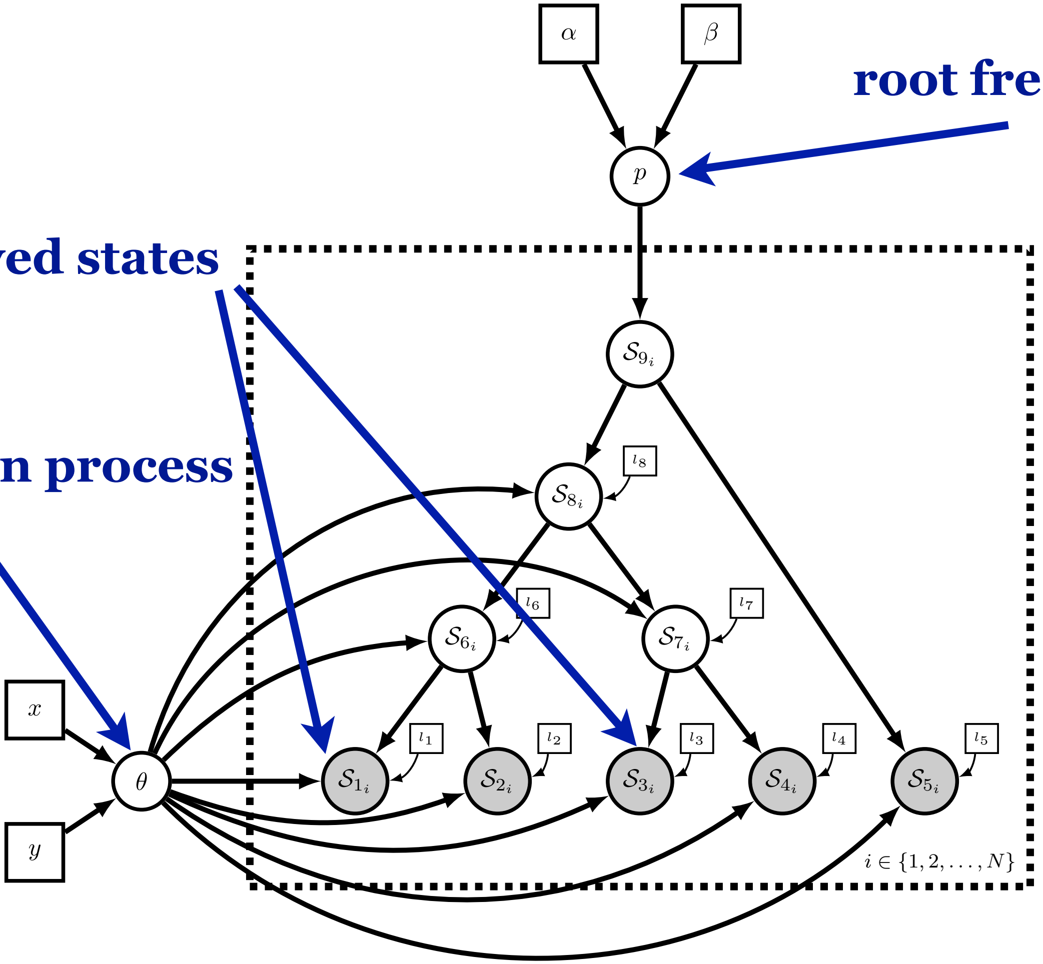
A graphical model for discrete characters



observed states

substitution process parameter

A graphical model for discrete characters

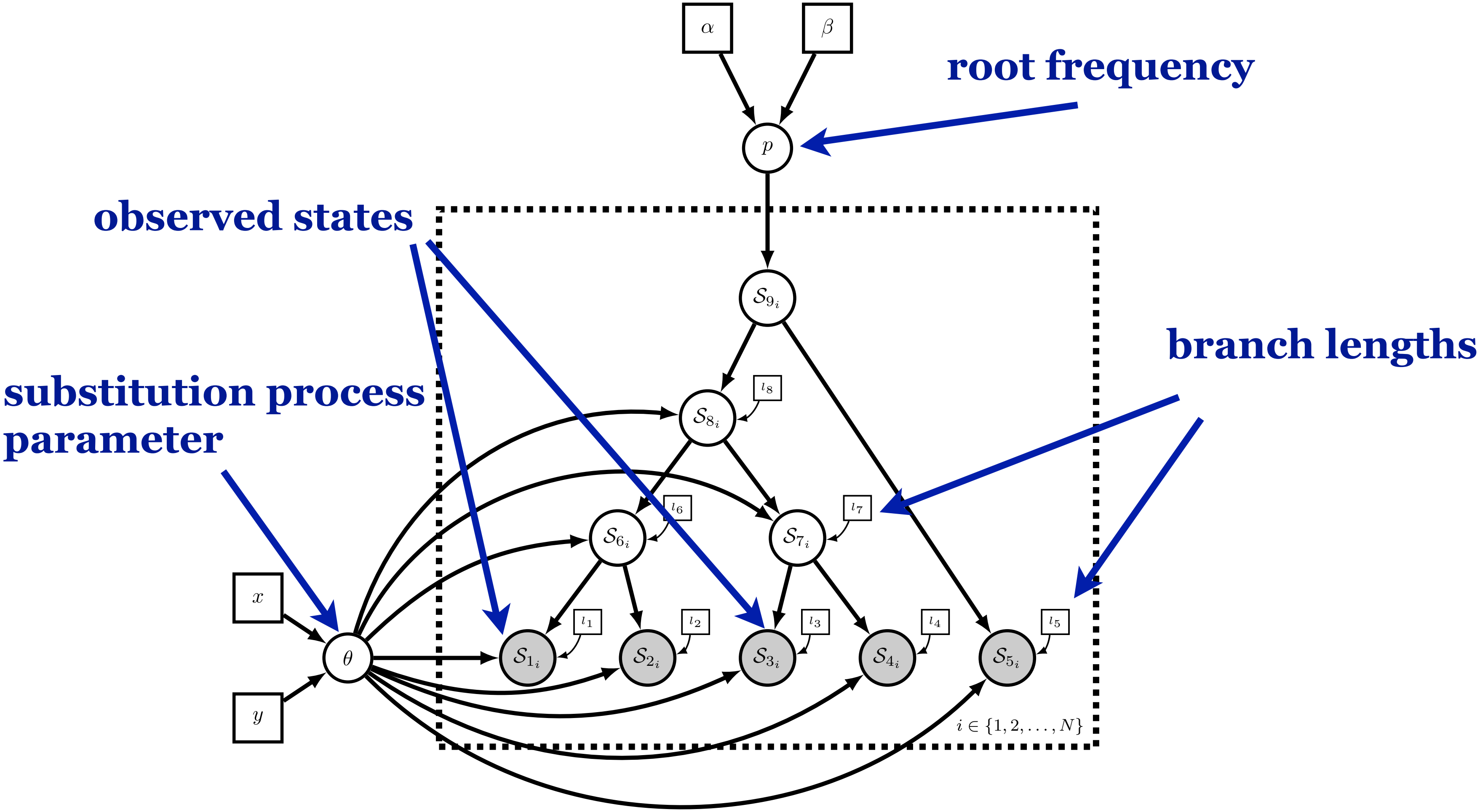


root frequency

observed states

substitution process parameter

A graphical model for discrete characters



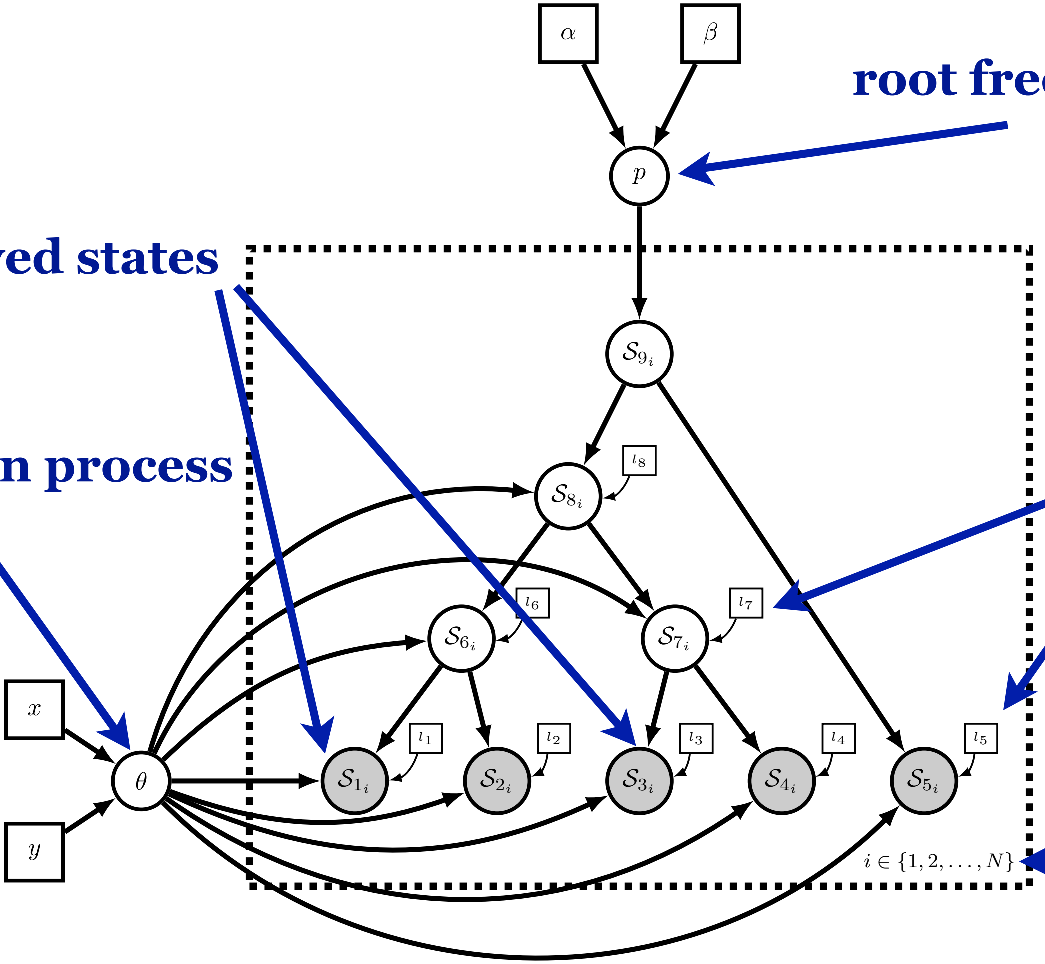
root frequency

observed states

branch lengths

substitution process parameter

A graphical model for discrete characters



root frequency

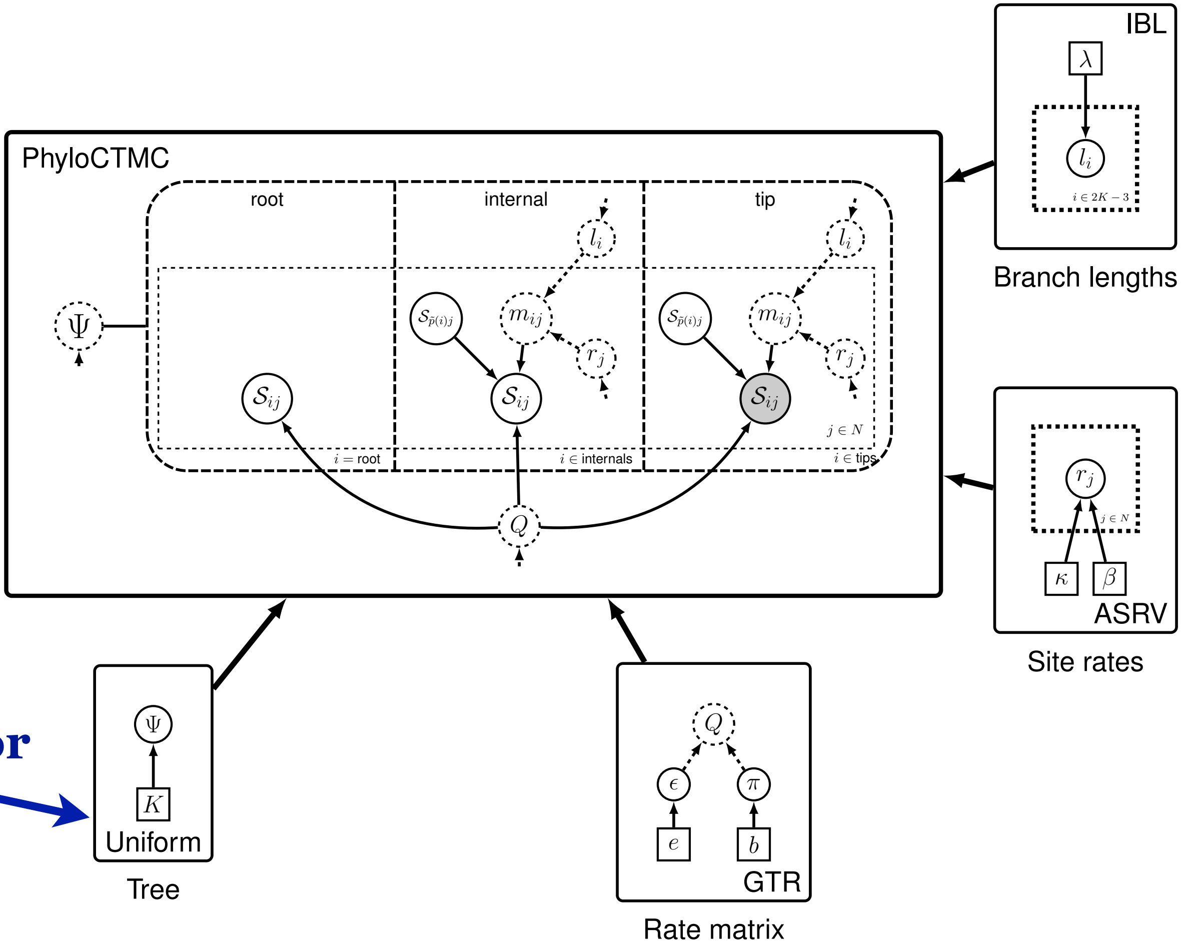
observed states

substitution process parameter

branch lengths

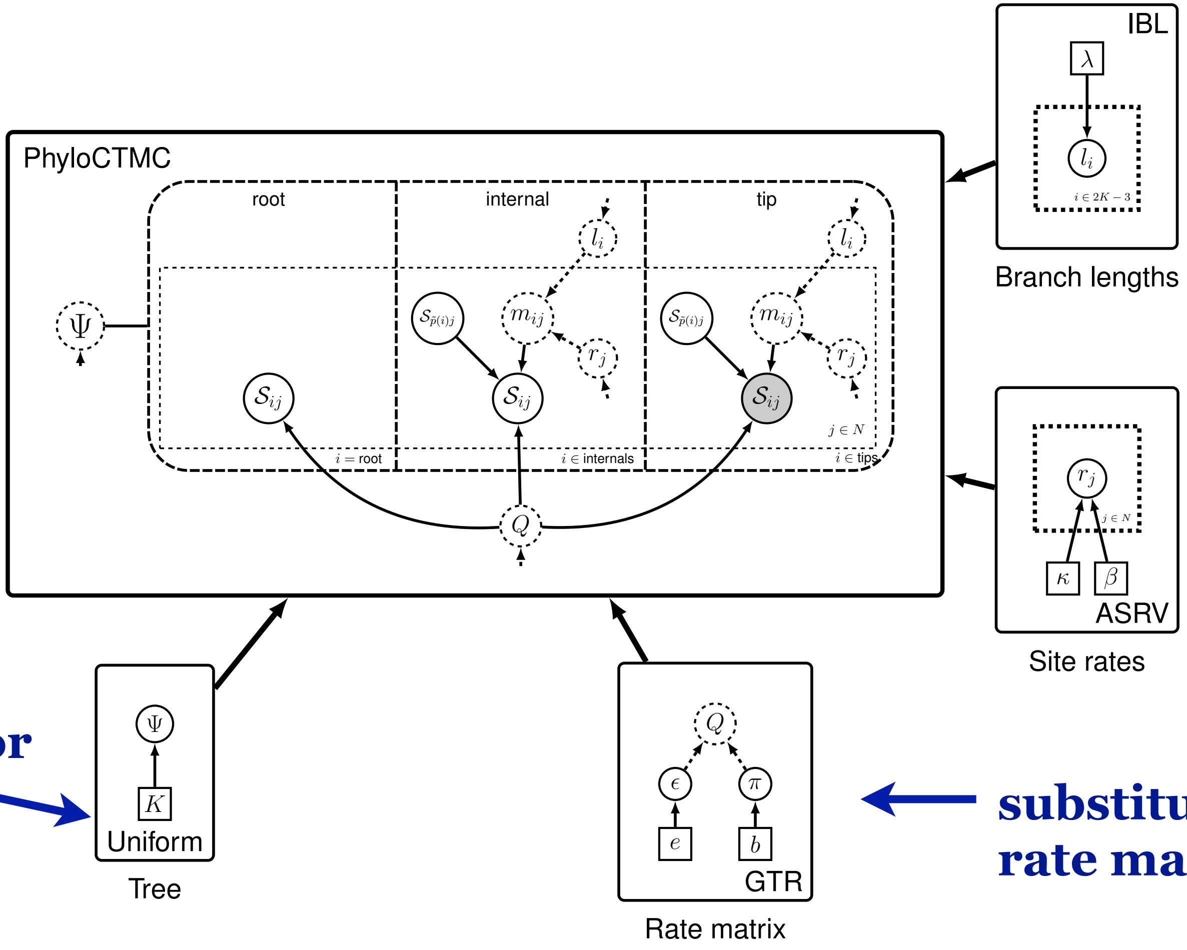
repeated for n sites

A compact/modular representation



tree prior

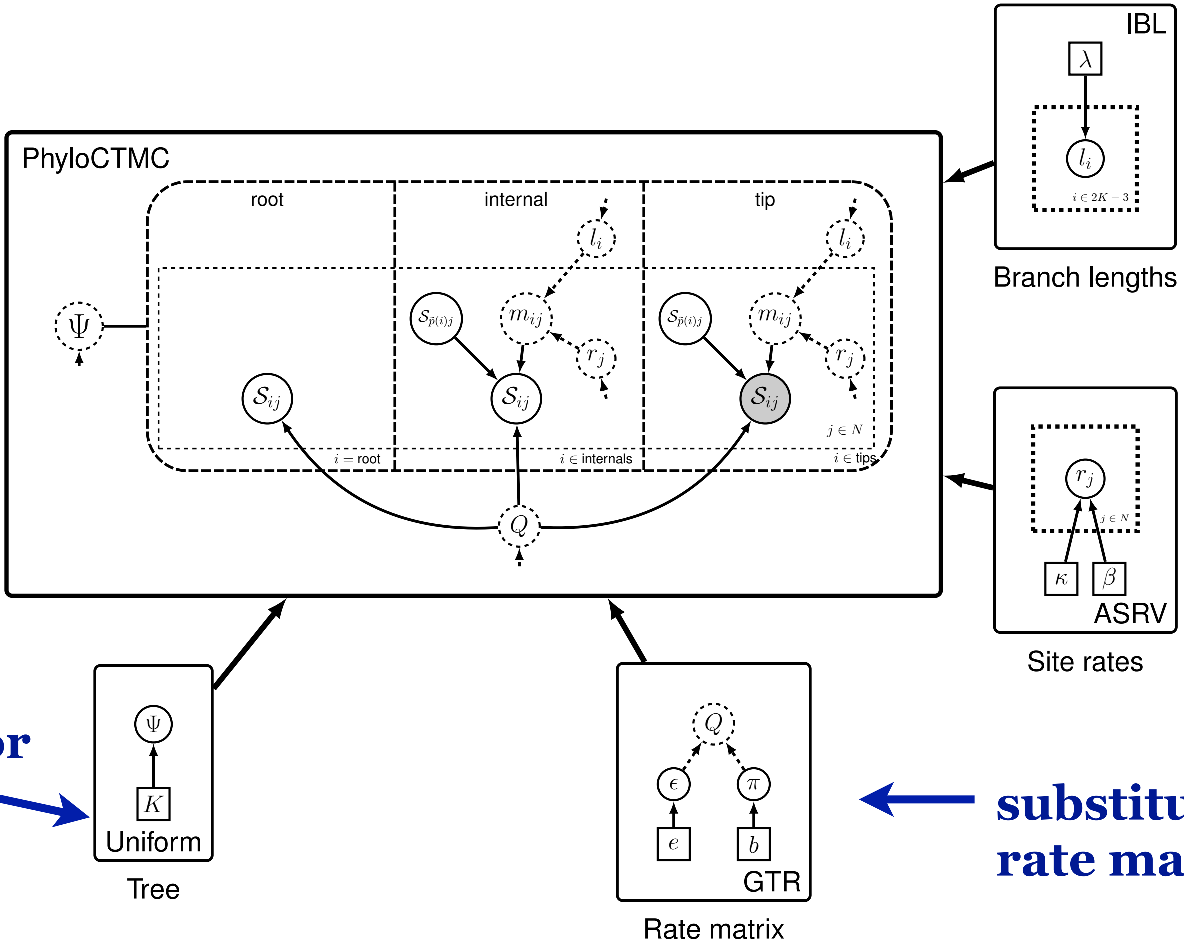
A compact/modular representation



tree prior

substitution rate matrix

A compact/modular representation

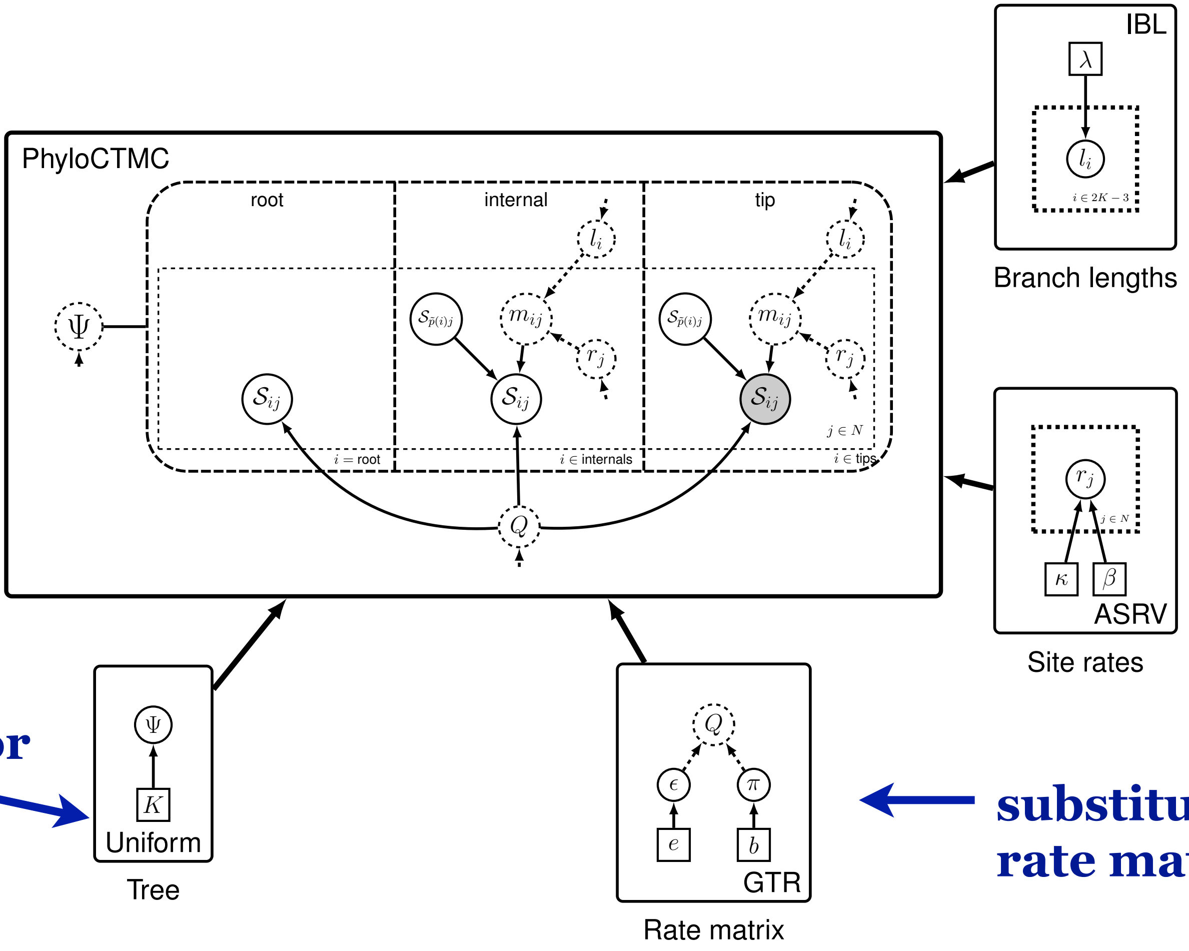


gamma distributed per site rates

tree prior

substitution rate matrix

A compact/modular representation



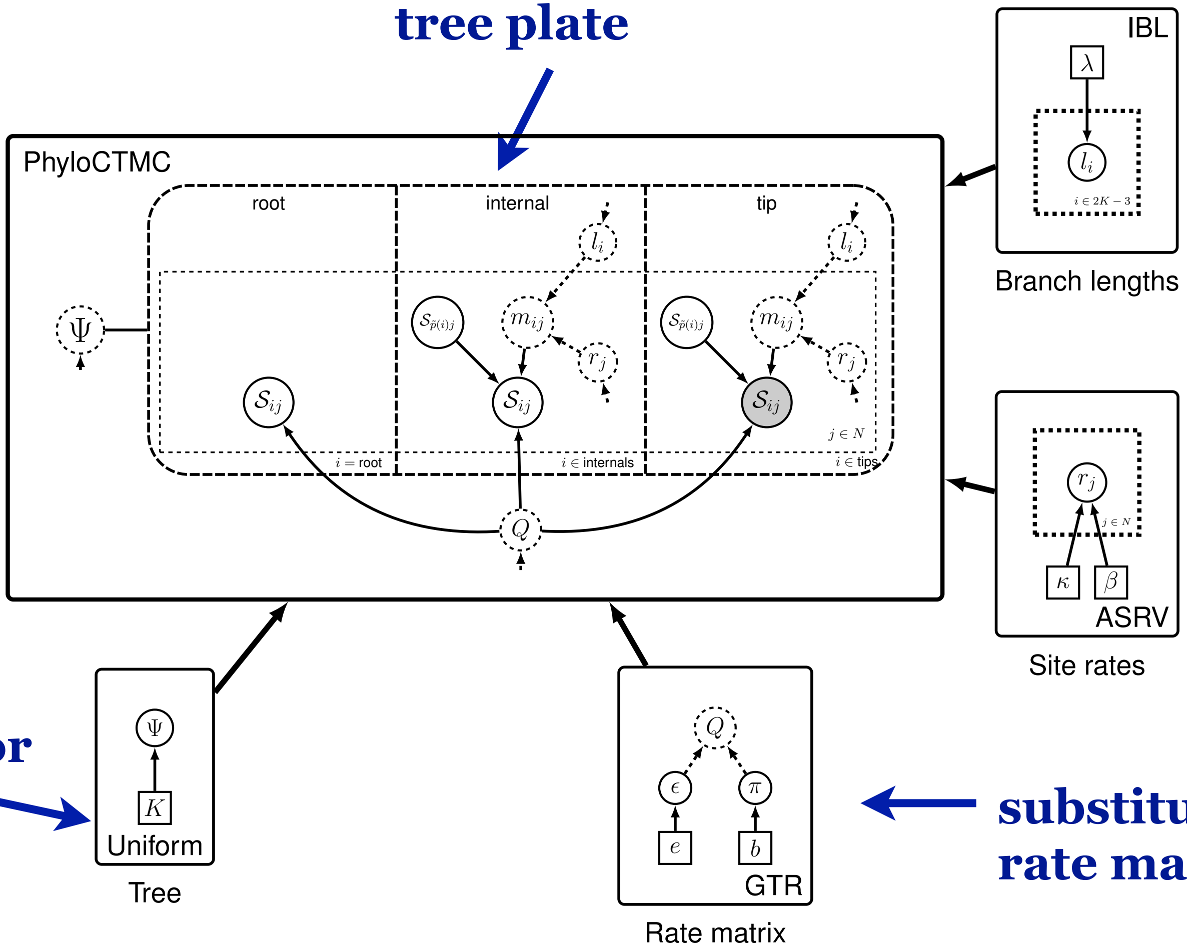
iid branch lengths

gamma distributed per site rates

tree prior

substitution rate matrix

A compact/modular representation



iid branch lengths

tree plate

gamma distributed per site rates

tree prior

substitution rate matrix

Features Available in RevBayes

- There are many features available now.
- Currently we are at a beta testing stage.
- The release will be submitted after this workshop.
- RevBayes is still under

Available distributions/functions/models

- Standard distributions:
  - Uniform distribution
  - Normal distribution
  - Exponential distribution
  - Gamma distribution
  - Lognormal distribution
  - Geometric distribution
  - …

Available distributions/functions/models

- Standard distributions:
  - Uniform distribution
  - Normal distribution
  - Exponential distribution
  - Gamma distribution
  - Lognormal distribution
  - Geometric distribution
  - …
- Tree priors:
  - Uniform topology (unrooted)
  - Uniform node-age
  - constant-rate birth-death
  - diversity-dependent pure-birth
  - constant population-size coalescent

Available distributions/functions/models

- Standard distributions:
  - Uniform distribution
  - Normal distribution
  - Exponential distribution
  - Gamma distribution
  - Lognormal distribution
  - Geometric distribution
  - …
- Tree priors:
  - Uniform topology (unrooted)
  - Uniform node-age
  - constant-rate birth-death
  - diversity-dependent pure-birth
  - constant population-size coalescent
- Substitution models:
  - Jukes Cantor
  - Felsenstein 81
  - HKY85
  - General time reversible (GTR)
  - Empirical Amino Acid (mtRev, …)
  - Coala
  - …

Available distributions/functions/models

- Standard distributions:
  - Uniform distribution
  - Normal distribution
  - Exponential distribution
  - Gamma distribution
  - Lognormal distribution
  - Geometric distribution
  - …

- Tree priors:
  - Uniform topology (unrooted)
  - Uniform node-age
  - constant-rate birth-death
  - diversity-dependent pure-birth
  - constant population-size coalescent

- Substitution models:
  - Jukes Cantor
  - Felsenstein 81
  - HKY85
  - General time reversible (GTR)
  - Empirical Amino Acid (mtRev, …)
  - Coala
  - …

- Rate variation among sites:
  - Any mixture you want (e.g., gamma)!

Available distributions/functions/models

- Clock models:
  - strict clock
  - iid clock rates (e.g., independent gamma rates)
  - mixture distributions (e.g., UCLN and UCE)
  - autocorrelated lognormal
  - RLC
  - DPP
  - …

Available distributions/functions/models

- Clock models:
  - strict clock
  - iid clock rates (e.g., independent gamma rates)
  - mixture distributions (e.g., UCLN and UCE)
  - autocorrelated lognormal
  - RLC
  - DPP
  - …

- Gene-tree species-tree:
  - constant population-size multispecies coalescent

Available distributions/functions/models

- Clock models:
  - strict clock
  - iid clock rates (e.g., independent gamma rates)
  - mixture distributions (e.g., UCLN and UCE)
  - autocorrelated lognormal
  - RLC
  - DPP
  - …
- Gene-tree species-tree:
  - constant population-size multispecies coalescent
- Additional models:
  - Branch-heterogeneous substitution models (e.g., possibility to specify any substitution process per branch).

Available distributions/functions/models

- Clock models:
  - strict clock
  - iid clock rates (e.g., independent gamma rates)
  - mixture distributions (e.g., UCLN and UCE)
  - autocorrelated lognormal
  - RLC
  - DPP
  - …
- Gene-tree species-tree:
  - constant population-size multispecies coalescent

- Additional models:
  - Branch-heterogeneous substitution models (e.g., possibility to specify any substitution process per branch).
- Inference:
  - Metropolis-Hastings (MCMC and reversible jump MCMC)
  - Metropolis-coupled MCMC
  - Power-posteriors (Path-sampling and stepping-stone-sampling)

Performance Study: Primates

Primates:

- 12 taxa
- 898 sites
- 412 patterns

MCMC:

- burnin of 10^5
- chain length of 10^6
- only substitution model parameters are updated

	HKY	HKY+G	GTR	GTR+G
BEAST v1.8	95.8	325.5	110.3	354.9

*** MrBayes used two runs because the single run does not allow to set tree proposals to 0.

Performance Study: Primates

Primates:

- 12 taxa
- 898 sites
- 412 patterns

MCMC:

- burnin of 10^5
- chain length of 10^6
- only substitution model parameters are updated

	HKY	**HKY+G**	**GTR**	**GTR+G**
BEAST v1.8	95.8	325.5	110.3	354.9
MrBayes 3.2***	212.2	530.6	208.4	513.9

*** MrBayes used two runs because the single run does not allow to set tree proposals to 0.

Performance Study: Primates

Primates:
- 12 taxa
- 898 sites
- 412 patterns

MCMC:
- burnin of 10^5
- chain length of 10^6
- only substitution model parameters are updated

	HKY	HKY+G	GTR	GTR+G
BEAST v1.8	95.8	325.5	110.3	354.9
MrBayes 3.2***	212.2	530.6	208.4	513.9
RevBayes (general implementation)	152.2	640.1	202.9	693

*** MrBayes used two runs because the single run does not allow to set tree proposals to 0.

Performance Study: Primates

Primates:

- 12 taxa
- 898 sites
- 412 patterns

MCMC:

- burnin of 10^5
- chain length of 10^6
- only substitution model parameters are updated

	HKY	HKY+G	GTR	GTR+G
BEAST v1.8	95.8	325.5	110.3	354.9
MrBayes 3.2***	212.2	530.6	208.4	513.9
RevBayes (general implementation)	152.2	640.1	202.9	693
RevBayes (char specific)	92.6	269.8	120.8	326.4

*** MrBayes used two runs because the single run does not allow to set tree proposals to 0.

Performance Study: Primates

Primates:

- 12 taxa
- 898 sites
- 412 patterns

MCMC:

- burnin of 10^5
- chain length of 10^6
- only substitution model parameters are updated

	HKY	**HKY+G**	**GTR**	**GTR+G**
BEAST v1.8	95.8	325.5	110.3	354.9
MrBayes 3.2***	212.2	530.6	208.4	513.9
RevBayes (general implementation)	152.2	640.1	202.9	693
RevBayes (char specific)	92.6	269.8	120.8	326.4
RevBayes (SSE double precision)	65.1	246.7	114.8	302.6

*** MrBayes used two runs because the single run does not allow to set tree proposals to 0.

Performance Study: MCMC Shortcuts

Primates:

- 12 taxa
- 898 sites
- 412 patterns

Cetaceans:

- 71 taxa
- 1140 sites
- 578 patterns

MCMC:

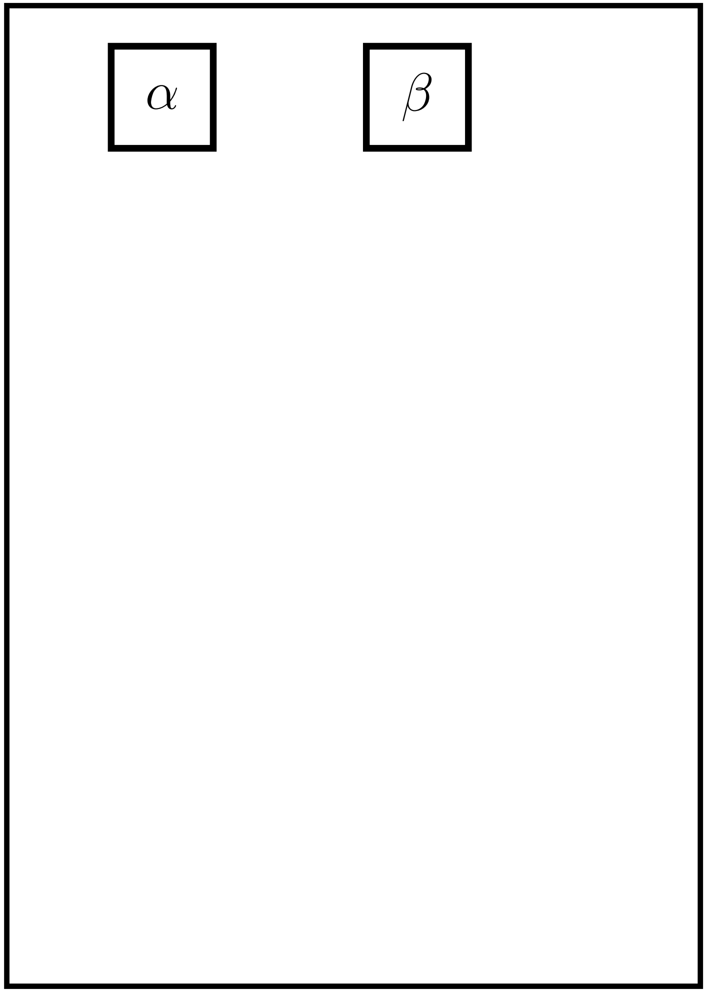
- burnin of 10^5
- chain length of 10^6
- only topology or node ages are updated

	Narrow	**NodeSlide**	**Narrow**	**NodeSlide**
BEAST v1.8	3:19	3:41	6:49	10:40
RevBayes (SSE double precision)	1:29	1:38	4:18	5:29
	Primates		**Cetaceans**	

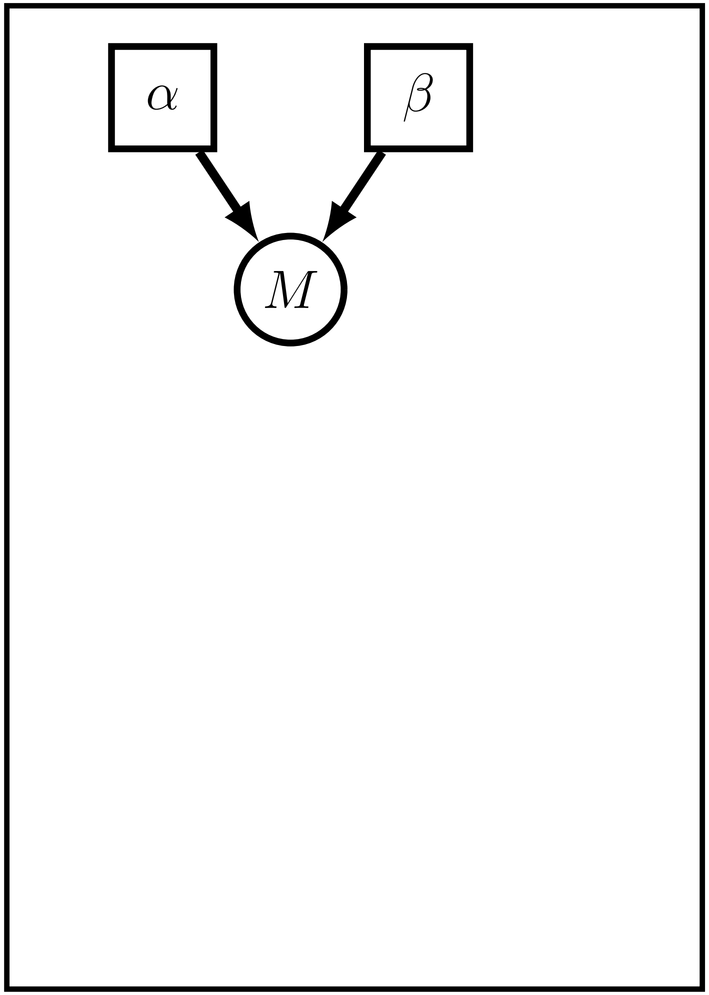
A brief intro to Rev

- Rev - The computing language used within RevBayes:
  - is an interactive environment
  - basic syntax is inspired by 'R' (and partially by BUGS)
  - aimed to built graphical models
  - provides standard 'easy-to-use' math-functions

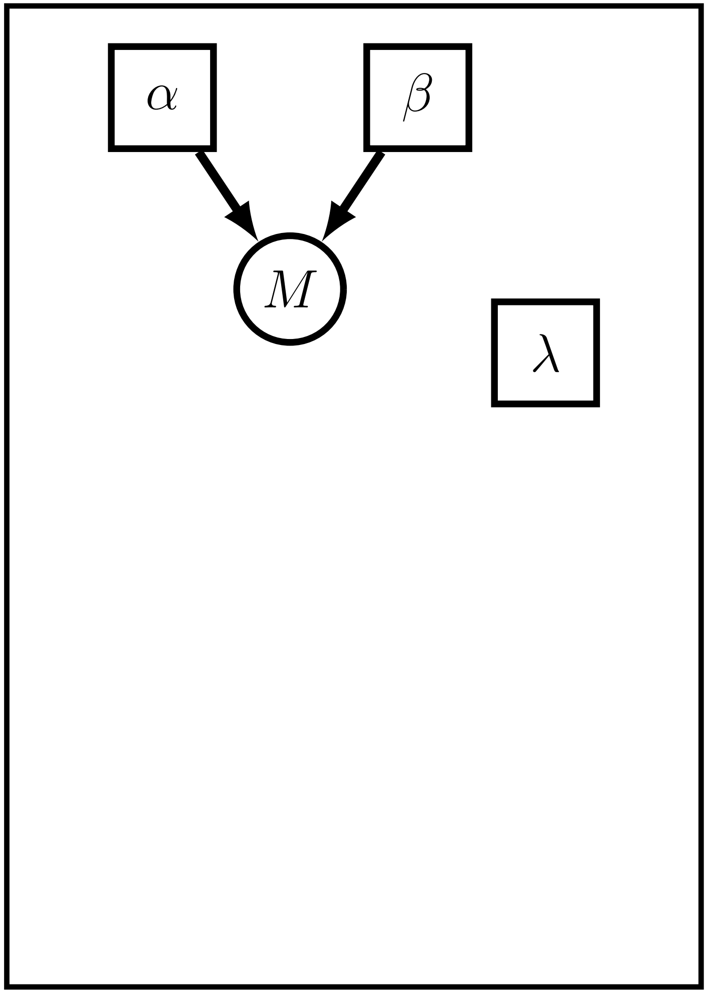
```
observations <- [<your data go here>]
```



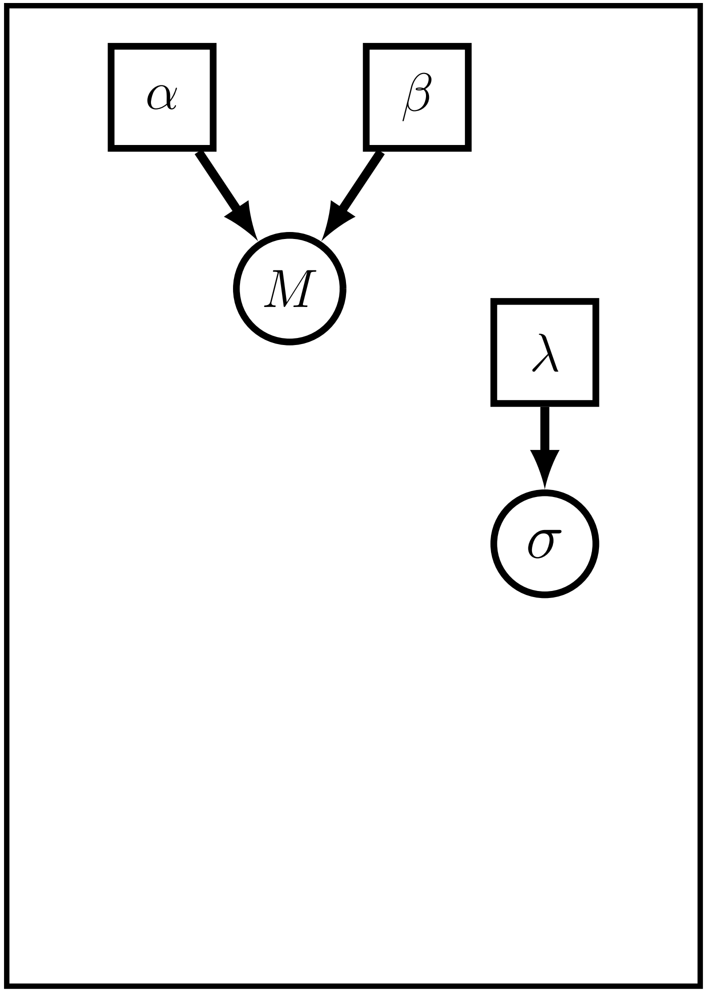
```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
```



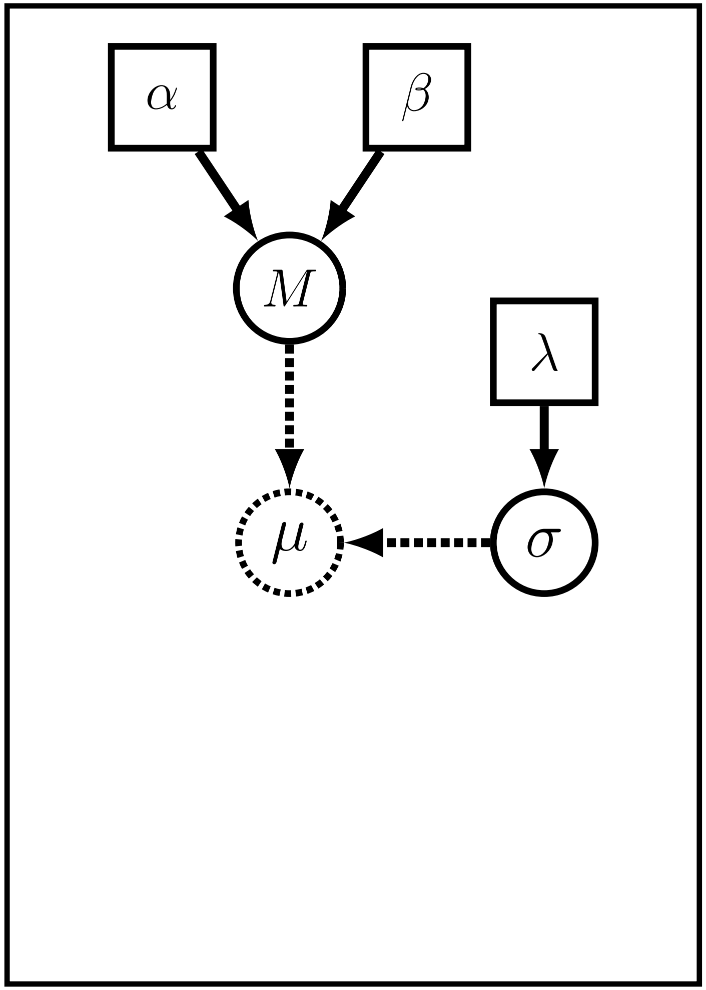
```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
M ~ dnGamma(alpha, beta)
```



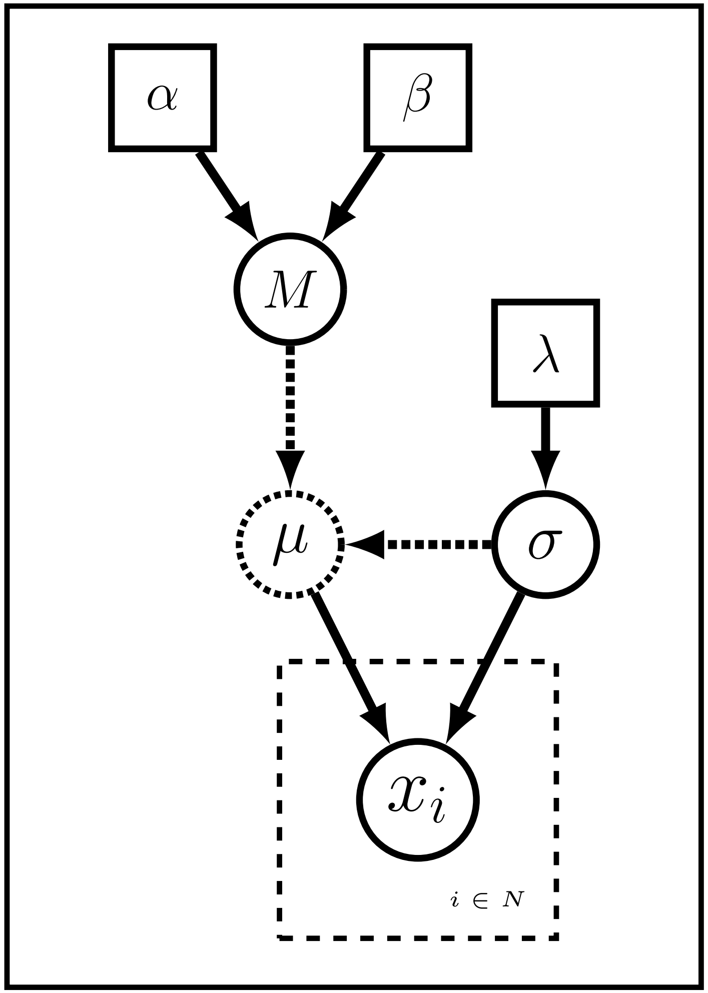
```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
M ~ dnGamma(alpha, beta)


lambda <- 1.0
```



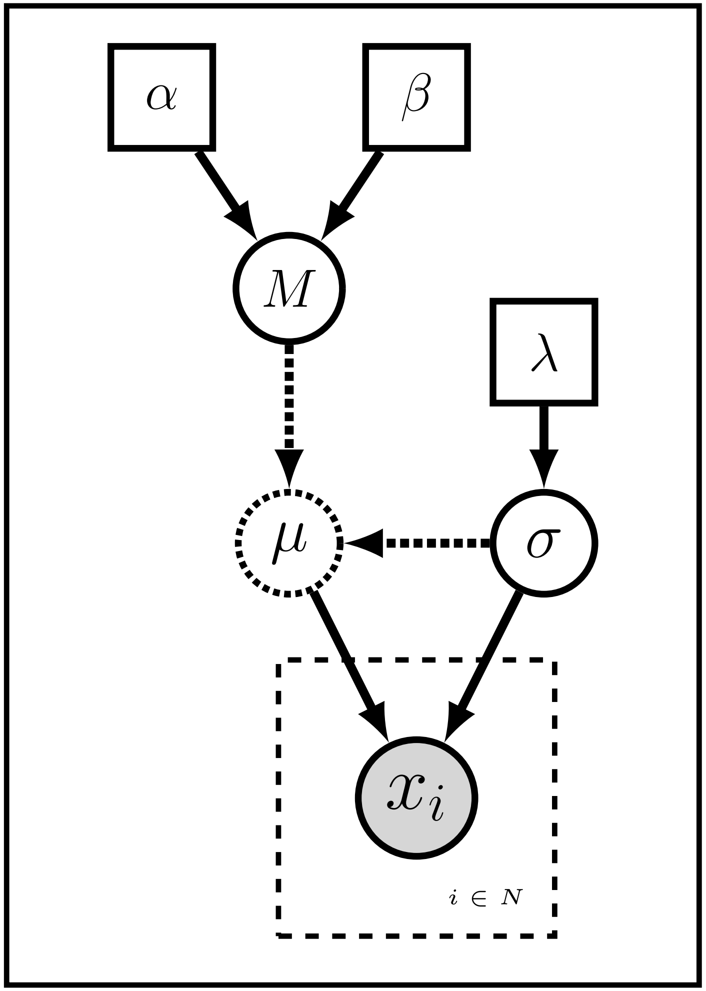
```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
M ~ dnGamma(alpha, beta)


lambda <- 1.0
sigma ~ dnExponential(lambda)
```



```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
M ~ dnGamma(alpha, beta)


lambda <- 1.0
sigma ~ dnExponential(lambda)


mu := ln(M) - (power(sigma, 2.0) / 2.0)
```



```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
M ~ dnGamma(alpha, beta)


lambda <- 1.0
sigma ~ dnExponential(lambda)


mu := ln(M) - (power(sigma, 2.0) / 2.0)


N <- observations.size()
for( i in 1:N ){
  x[i] ~ dnLnorm(mu, sigma)

}
```



```
observations <- [<your data go here>]

alpha <- 3.0
beta <- 1.0
M ~ dnGamma(alpha, beta)


lambda <- 1.0
sigma ~ dnExponential(lambda)


mu := ln(M) - (power(sigma, 2.0) / 2.0)


N <- observations.size()
for( i in 1:N ){
  x[i] ~ dnLnorm(mu, sigma)
  x[i].clamp(observations[i])
}
```