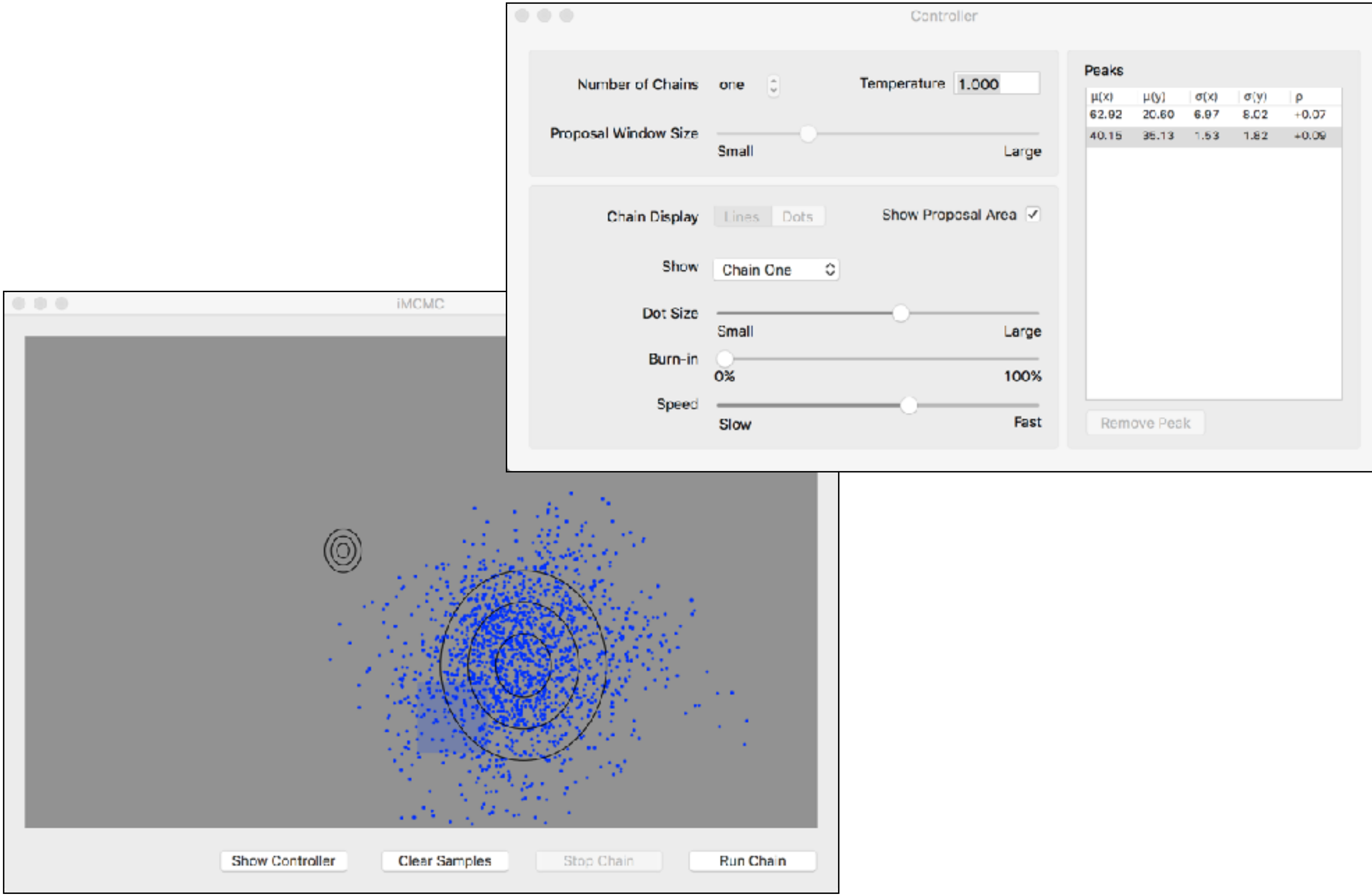
Review of MCMC

What happens when you change…

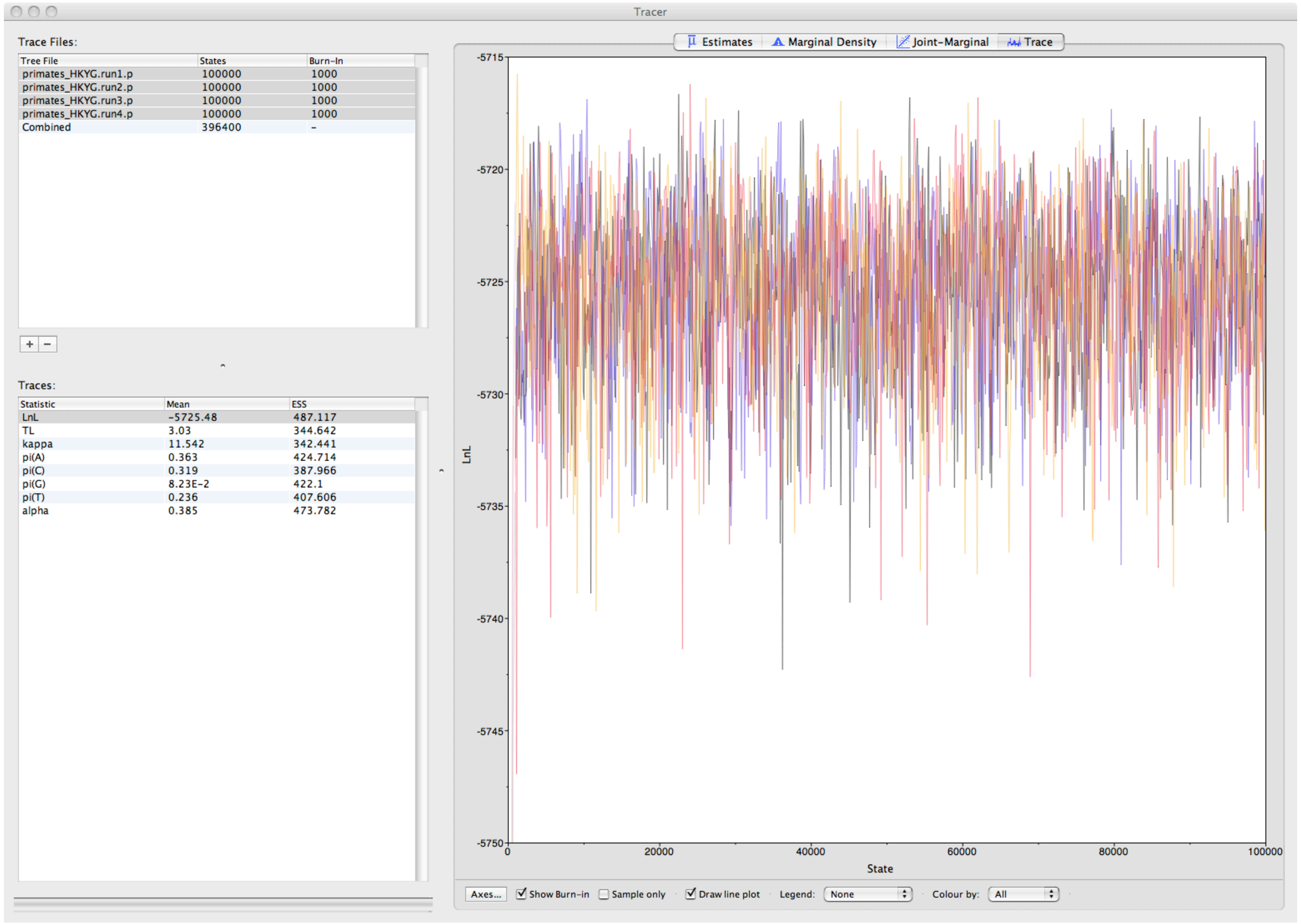
- The size of the proposal distribution
- The number of generations
- The burn-in

iMCMC

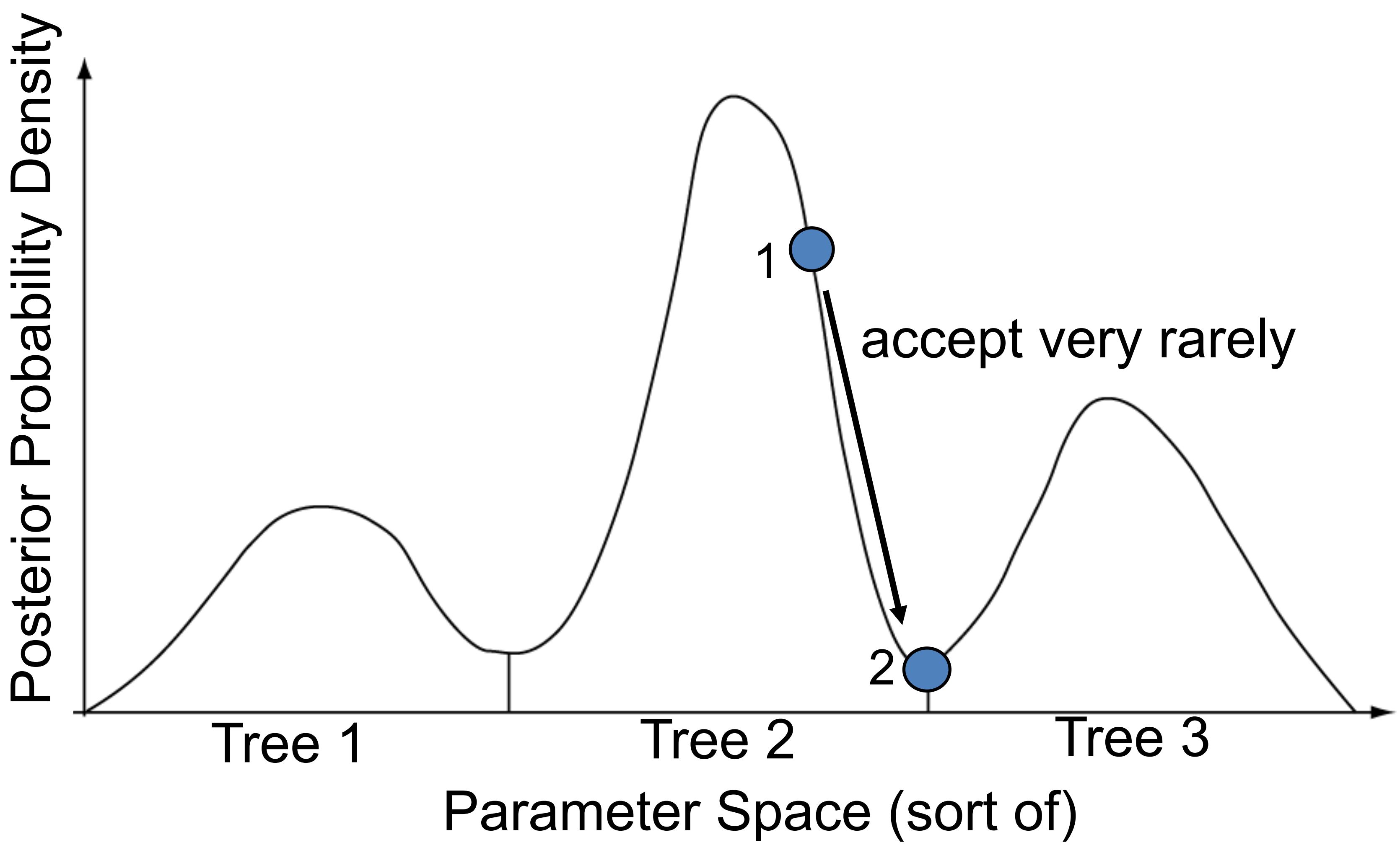


Controller
Number of Chains
one
Temperature
1.000
Proposal Window Size
Small
Large
Chain Display
Lines
Dots
Show Proposal Area
Show
Chain One
Dot Size
Small
Large
Burn-in
0%
100%
Speed
Slow
Fast
Peaks
μ(x) μ(y) σ(x) σ(y) ρ
62.92 20.60 6.97 8.02 +0.07
40.15 35.13 1.53 1.82 +0.09
Remove Peak
iMCMC
Show Controller
Clear Samples
Stop Chain
Run Chain

Convergence of Scalars - Tracer



Metropolis Coupling

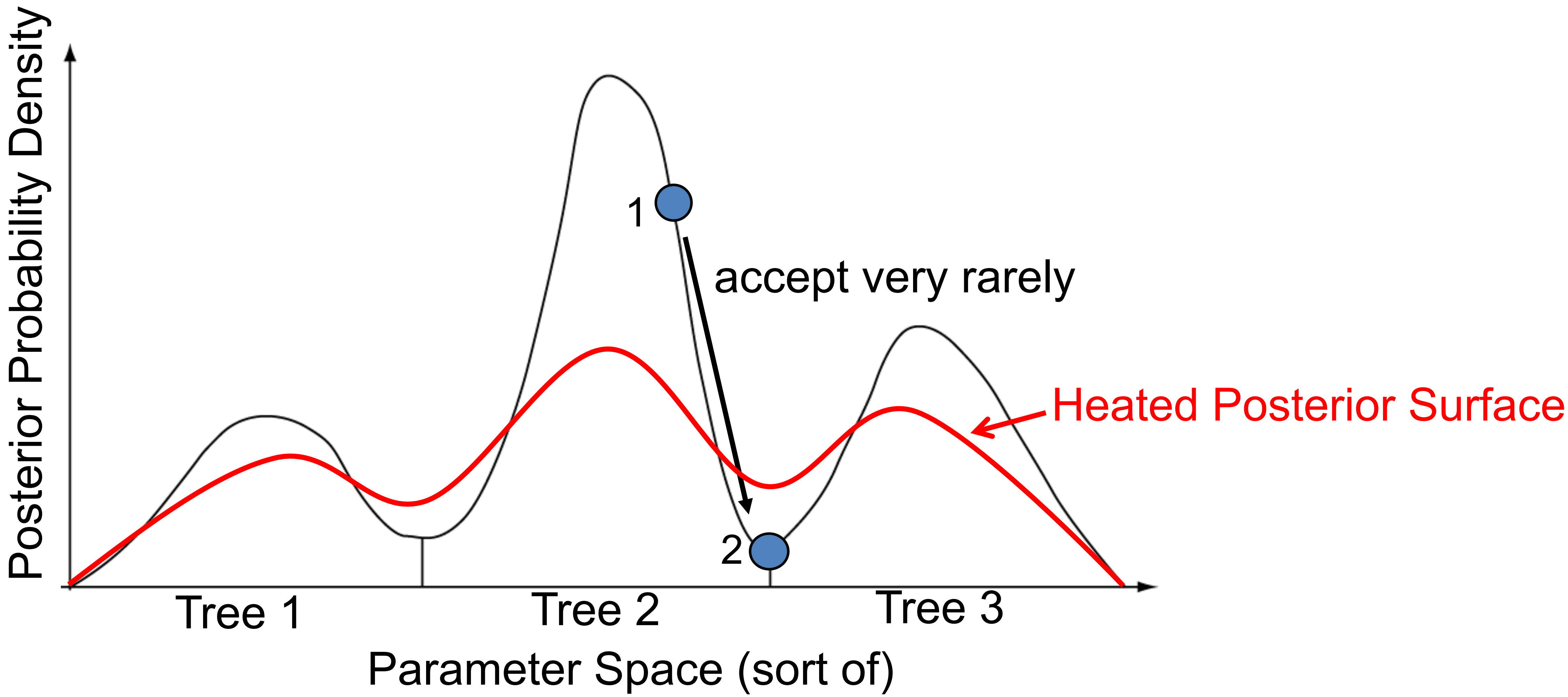


This slide "borrowed" from F. Ronquist

Metropolis Coupling

- Same rules as regular MCMC, but now there are multiple chains with different ‘temperatures’.
- ‘Heated’ chains sample a ‘melted’ version of the posterior
- Only difference is that heated chains raise the ratio of posterior densities to (1-temp) when deciding whether to accept a move.

$r^{(1\text{-temp})}$ approaches 1 (flat surface) as temp. increases

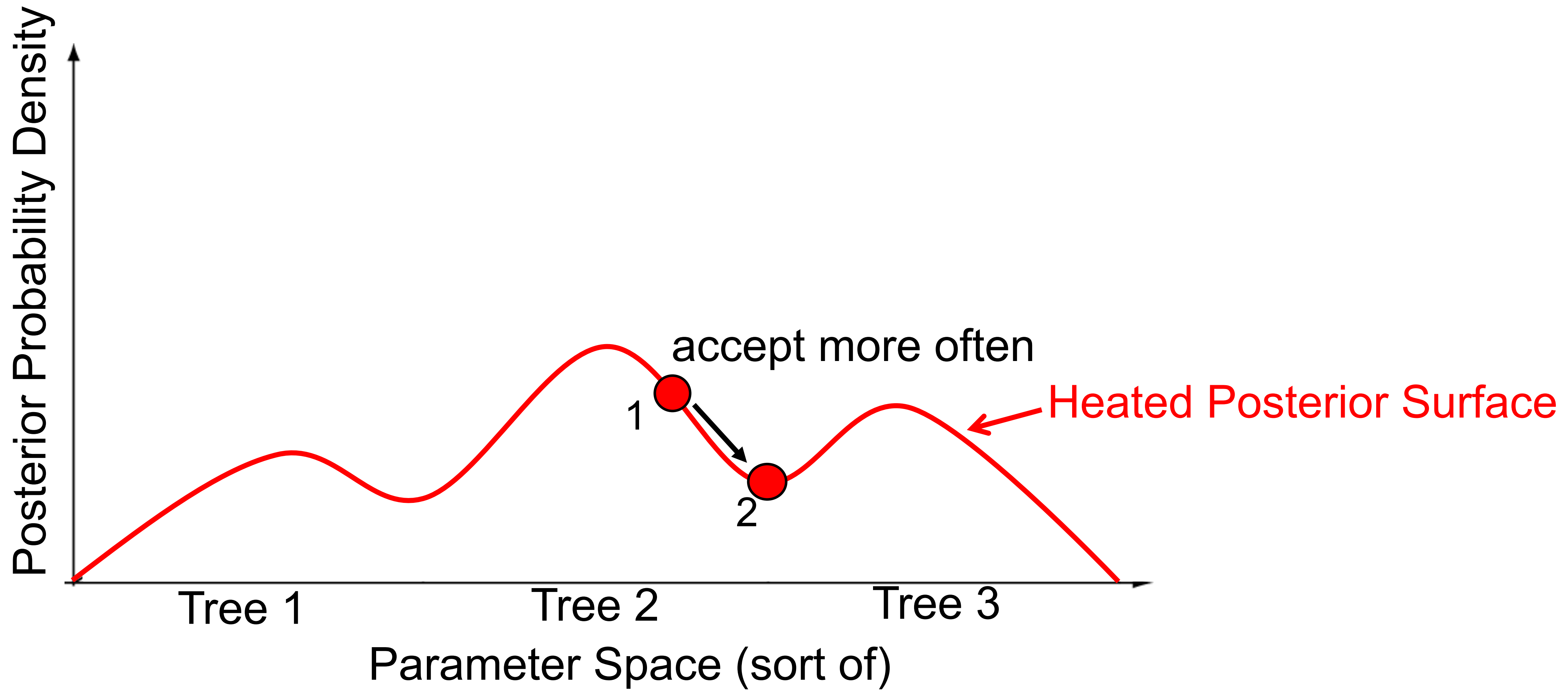


This slide “borrowed” from F. Ronquist

Metropolis Coupling

- Same rules as regular MCMC, but now there are multiple chains with different 'temperatures'.
- 'Heated' chains sample a 'melted' version of the posterior
- Only difference is that heated chains raise the ratio of posterior densities to (1-temp) when deciding whether to accept a move.

$r^{(1\text{-temp})}$ approaches 1 (flat surface) as temp. increases



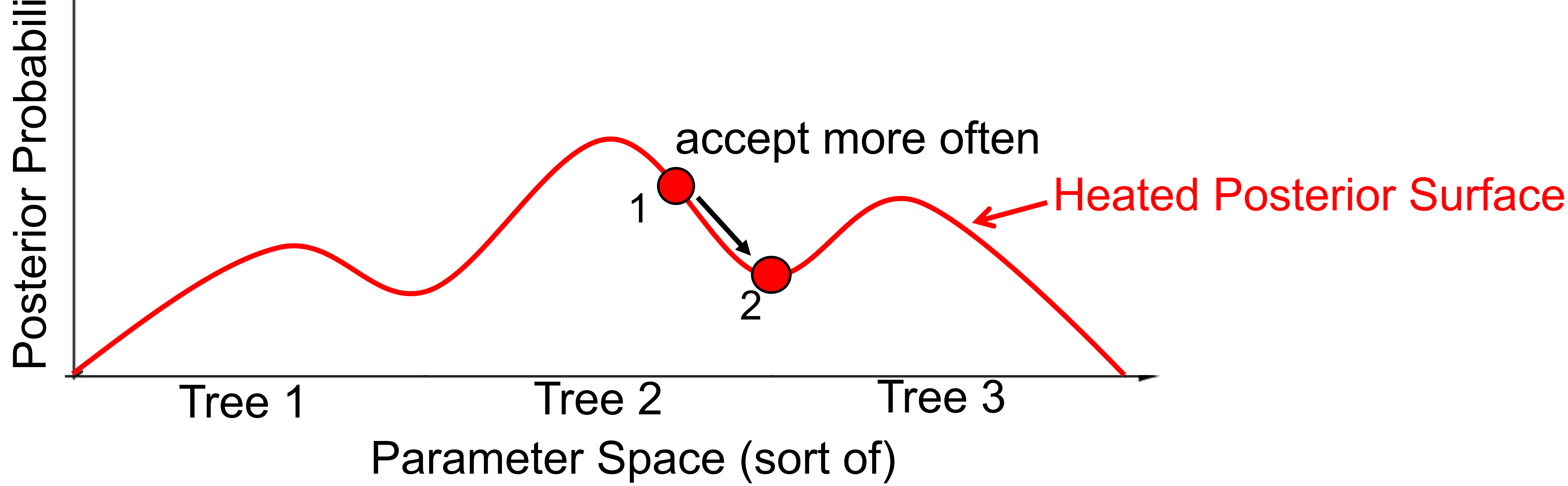
This slide "borrowed" from F. Ronquist

Metropolis Coupling

- Same rules as regular MCMC, but now there are multiple chains with different ‘temperatures’.
- ‘Heated’ chains sample a ‘melted’ version of the posterior
- Only difference is that heated chains raise the ratio of posterior densities to (1-temp) when deciding whether to accept a move.

$r^{(1\text{-temp})}$ approaches 1 (flat surface) as temp. increases

- Samples only recorded from the ‘cold’ chain
- Heated chains are ‘scout’s. Occasionally propose to trade places with the cold chain.



This slide “borrowed” from F. Ronquist

Graphical Models

Graphical Models

Graphical models provide a means of depicting the dependencies among parameters in probabilistic models.

The "graphical" part of the name graphical models has to do with their basis on graphs, and not anything to do with drawing.

But it is convenient that graphical models can also be drawn, to clearly depict the structure of the model.

Graphical Model Syntax

Symbol

$$\boxed{x}$$

Type

Constant Node

Description

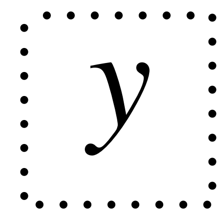
Constant nodes are like standard variables in a programming language. They are assigned fixed values.

Rev Example

```
x <- 2.3
```

Graphical Model Syntax

Symbol



Type

Deterministic Node

Description

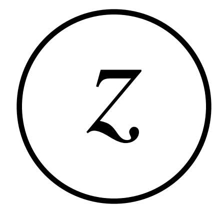
Deterministic nodes depend on the values of other nodes, but in a deterministic (fixed) way. They have no probability distribution intrinsically associated with them.

Rev Example

```
y  :=  2 * x
```

Graphical Model Syntax

Symbol



Type

Stochastic Node

Description

Stochastic nodes represent random variables and take on values according to some probability distribution. These distributions may have parameters defined in other nodes in the model graph. The type of distribution associated with a node is often not written explicitly when a model is drawn, but it must be specified.

Rev Example

```
z ~ dnBernoulli(0.5)
```

Graphical Model Syntax

Symbol

Type

Clamped Stochastic Node

Description

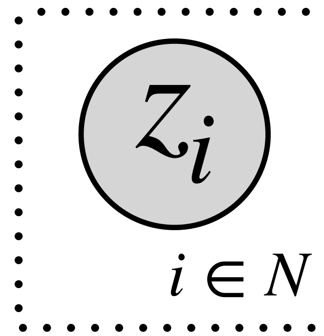
Data can be viewed as the observed outcome of a stochastic node. Clamping involves assigning a set of observations to an associated stochastic node. Clamping data allows the values of other parameters in the model to be inferred.

Rev Example

```
z  ~  dnBernoulli(0.5)
      z.clamp(1)
```

Graphical Model Syntax

Symbol



Type

Plate

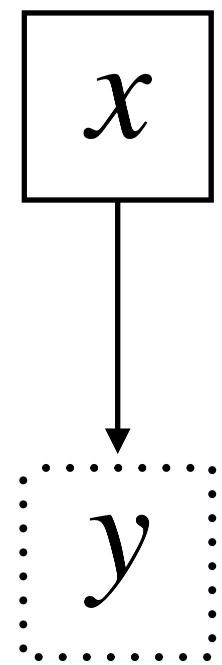
Description

Plates simply represent repetition and make it easier to depict graphical models with a repetitive structure. For instance, the example above shows *N* instances of the clamped variable, *z*, with each instance indexed by *i*. Plates can be used with any type of node, but are commonly used with clamped nodes representing data.

Rev Example

```
N <- 10
for (i in 1:N){
     z  ~  dnBernoulli(0.5)
     z.clamp(1)}
```

Super Simple Models

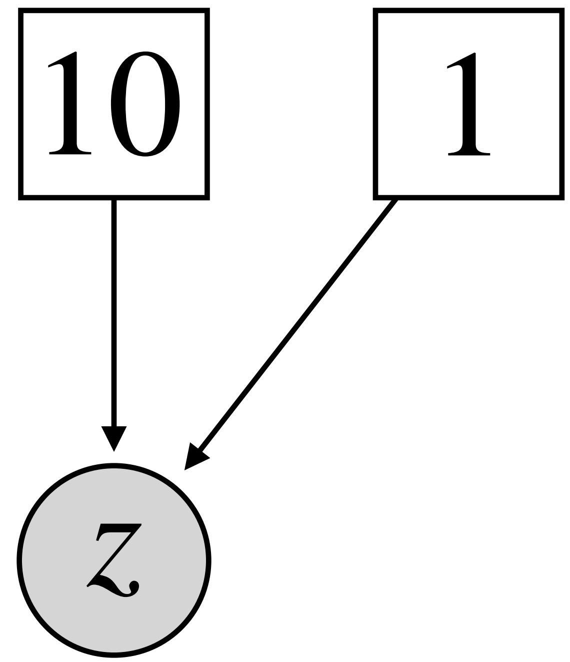



```
x <- 2
y := x / 4
```

What's the structure of this model?

What would happen if x was set to 16?

Super Simple Models



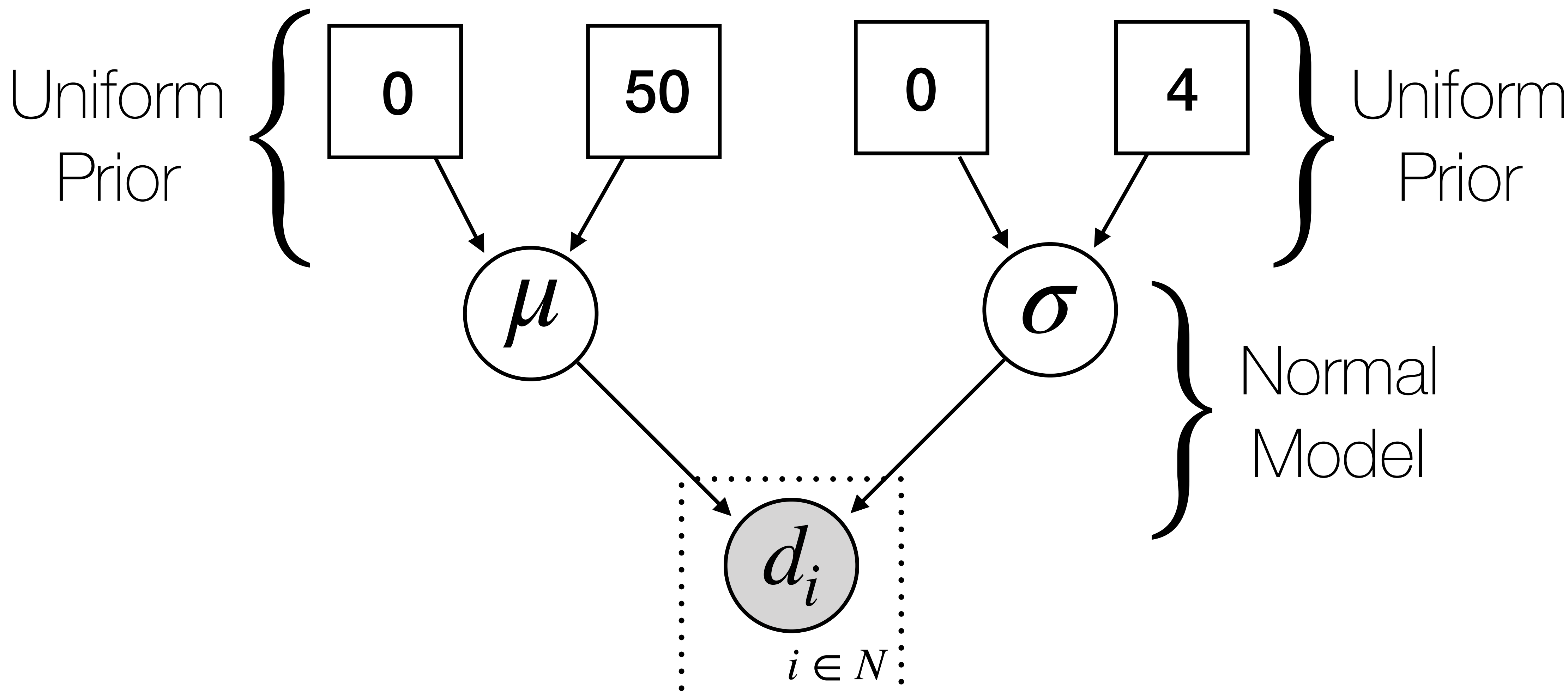

```
z ~ dnNormal(10,1)
z.clamp(9.6)
```

What's the structure of this model?
What can we infer about this model

Intro. to Graphical Models Jupyter Notebook (Part 1)

Normal Model

(from last week)

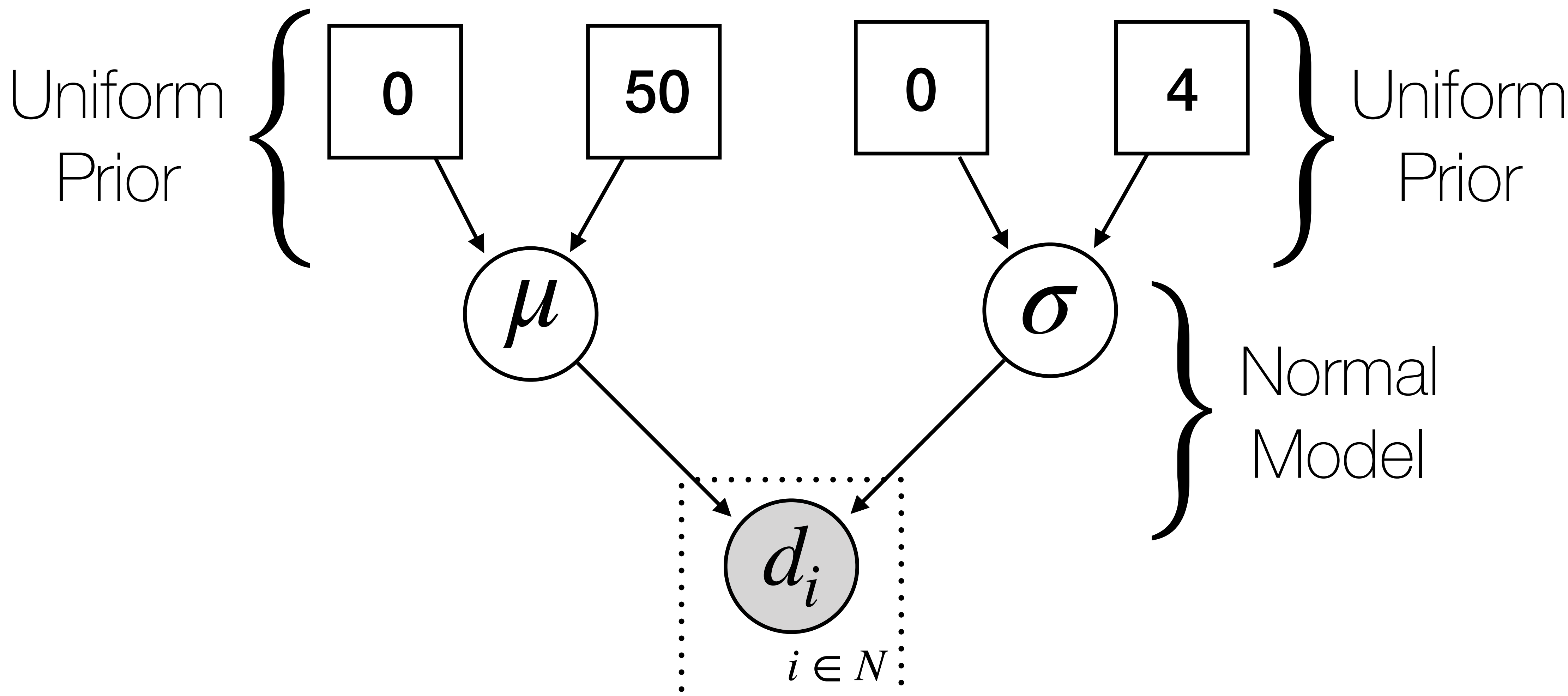


$d =$ observed data

$$d = \{10.1, 9.7, ..., 11.5\}$$

Where are the constant nodes? The deterministic nodes?
The stochastic nodes? The clamped stochastic nodes?

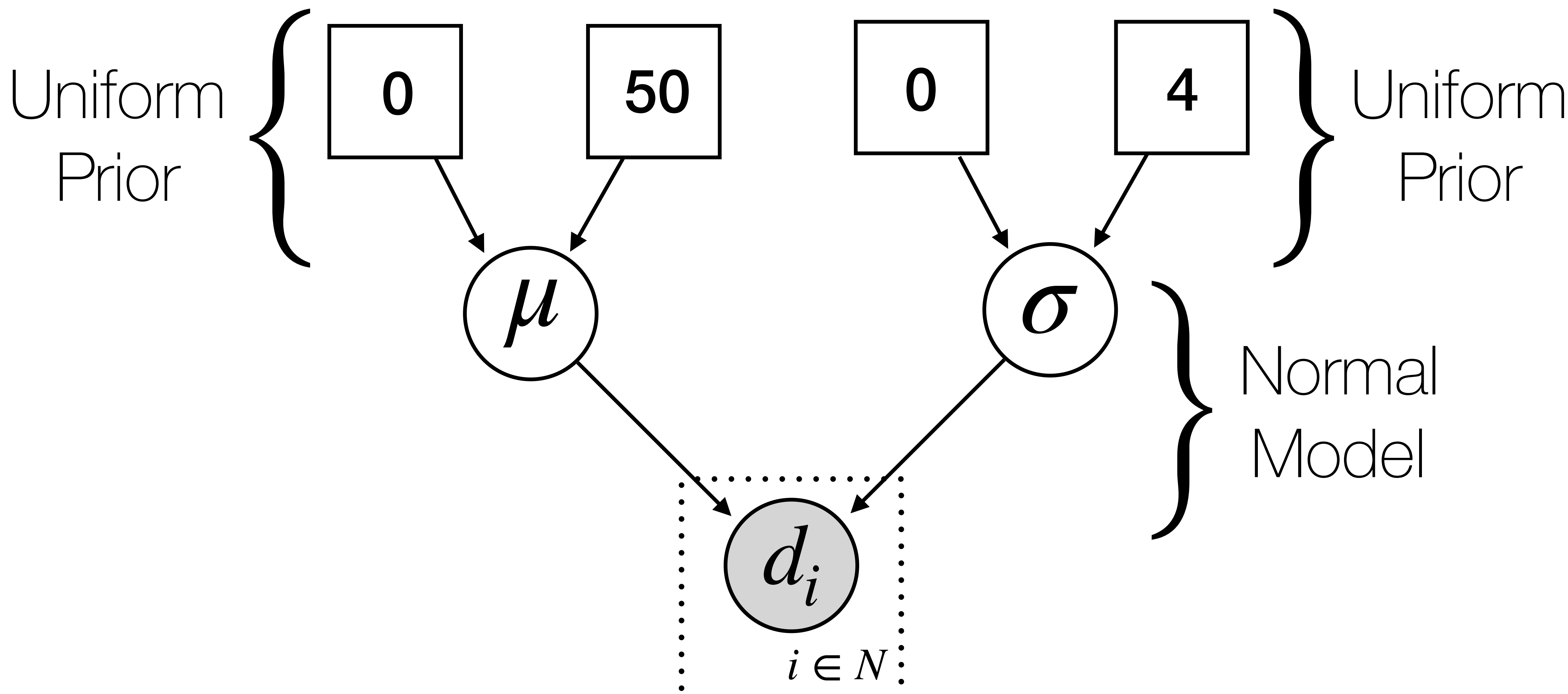
What parameters could we infer?



$d =$ observed data

$$d = \{10.1, 9.7, \ldots, 11.5\}$$

How would we write out this model in RevBayes?
Give it a try!
Start at the top and work your way down.
Make up your own data.



$d =$ observed data

$$d = \{10.1, 9.7, ..., 11.5\}$$

Intro. to Graphical Models Jupyter Notebook (Part 2)