Phylogenies

Bret Larget

Departments of Botany and of Statistics
University of Wisconsin—Madison

February 8, 2011

What is a Phylogeny?

Part of a statistical model for comparative studies that explains covariance among measurements of traits due to common ancestry.

Tony Ives

A connected, acyclic edge-weighted, semi-labeled graph where tip nodes are labeled to represent taxa and edge weights usually represent the expected number of nucleotide substitutions per site.

Cécile Ané

Truth.

Ken Sytsma

How do we estimate a phylogeny?

- There are a multitude of methods to estimate phylogenies from various sorts of *data*.
- Presently, the most common approaches use *multiple alignments of DNA sequence data*, but this is not always the case (especially when some taxa are represented by fossils).
- There are methods for *trait data*, *amino acid sequences*, *AFLP markers*, *restriction sites*, and others.
- When selecting a method to construct a phylogeny, it is important to understand the underlying assumptions.

Methods of Phylogenetic Reconstruction

Primary methods of phylogenetic reconstruction include these:

- Parsimony
- UPGMA
- Neighbor joining (and variants)
- Maximum likelihood
- Bayesian approaches

Here are important considerations for each with regard to finding a tree for a comparative analysis.

Parsimony

- Parsimony seeks the tree topology that requires the fewest total changes on each edge of the tree.
- Parsimony does not directly estimate edge lengths.
- For a given site, there can be multiple equally parsimonious ways to map the minimum number of changes onto a tree.
- If a tree topology is selected by parsimony, additional methods are needed to find branch lengths.
- There are conditions (especially *long branch attraction*) where the parsimony method is likely to select the incorrect tree topology.
- Evaluation of the parsimony score on a single tree is computationally fast, but searching for the single most parsimonious tree when there are many taxa requires heuristic methods that may not find the true optimal tree.

UPGMA

- UPGMA acts directly on a pairwise distance matrices among taxa; it is an algorithm for producing a tree from such a distance matrix, not a model.
- UPGMA produces rooted *ultrametric trees* (trees where all tips are equidistant from the root).
- Such trees are consistent with a *molecular clock hypothesis* in which the expected rate of nucleotide substitution is constant across all lineages.
- To use UPGMA, one needs to specify how distances between taxa are calculated; a common choice is the maximum likelihood distance between the sequences, but this also requires a selection of a maximum likelihood model.
- UPGMA and other distance methods are often used when data other than DNA sequences are used.

UPGMA (cont.)

- When the true underlying rates of nucleotide substitution are not equal, UPGMA can be biased against finding the correct tree topology.
- In formal likelihood-based tests, it is exceedingly rare with real sequence data to find examples where the molecular clock hypothesis is not strongly rejected.

Neighbor-joining

- Equivalently to UPGMA, neighbor-joining is an algorithm for producing trees directly from pairwise distances.
- Unlike UPGMA, neighbor-joining produces an *unrooted tree topology with branch lengths*.
- For comparative methods purposes, a root needs to be selected (often by using outgroups), but the resulting tree will not be ultrametric.
- Just as with UPGMA, neighbor-joining is an algorithm that makes trees rapidly from even large pairwise distance matrices, but to be a complete method requires a specification of how the distances are calculated.
- Both UPGMA and neighbor-joining lose information when reducing aligned DNA sequences to distances, and in many settings are less accurate in reconstructing the phylogeny than methods that work with sequence data directly.

Maximum Likelihood

- Maximum likelihood depends on an explicit *continuous-time Markov chain* model for how DNA sequence (or other data) changes along a tree.
- There are many variants among likelihood models that make fewer or greater restrictions among parameters.
- Similar to parsimony, maximum likelihood requires a heuristic search across tree space.
- Calculating the likelihood score for a given tree with branch lengths is about as computationally difficult as finding a parsimony score, but the need to optimize all parameter values in addition to branch lengths makes maximum likelihood more computationally intensive, especially for larger trees and large data sets.
- The resulting tree has branch lengths.
- The search for the best tree can be restricted to ultrametric trees.
- Variations include ultrametric trees with rates that can change along the tree (often in a penalized way).

Bayesian Methods

- The Bayesian paradigm differs from the other methods in that the end result is a probability distribution on tree space, not a single best tree.
- This distribution is typically represented by a large random (but not independent) sample of trees selected by *Markov chain Monte Carlo*.
- It is common for people to compute a *consensus tree* from the Bayesian sample as a single representative of the distribution.
- Bayesian methods can use the same likelihood models as used in maximum likelihood, and actually often use models richer in parameters than is feasible with maximum likelihood (for example, by partitioning data into parts, each with separate sets of parameters).
- Bayesian methods are computationally intensive for large data sets and trees; they are computationally favorable to maximum likelihood plus bootstrapping, but not for finding a single maximum likelihood tree.
- Bayesian methods can be restricted to ultrametric trees or not.

Bayesian Methods (cont.)

- To account for phylogenetic uncertainty in a comparative analysis, one approach is to use MCMC to select some trees, carryout the comparative analysis on each, and average the results.

A Famous Quote About Models

Essentially, all models are wrong, but some are useful.

George Box

The Markov Property

- Use the notation $X(t)$ to represent the base at time t.
- Formal statement:

$$\begin{aligned} &\mathrm{P}\{X(s+t) = j \mid X(s) = i, X(u) = x(u) \text{ for } u < s\} \\ &= \quad \mathrm{P}\{X(s+t) = j \mid X(s) = i\} \end{aligned}$$

- *Informal understanding: given the present, the past is independent of the future*.
- If the expression does not depend on the time s, the Markov process is called *homogeneous*.

Rate Matrix

- Positive off-diagonal rates of transition
- Negative total on the diagonal
- Row sums are zero
- Example

$$Q = \{q_{ij}\} = \begin{pmatrix} -1.1 & 0.3 & 0.6 & 0.2 \\ 0.2 & -1.1 & 0.3 & 0.6 \\ 0.4 & 0.3 & -0.9 & 0.2 \\ 0.2 & 0.9 & 0.3 & -1.4 \end{pmatrix}$$

Alarm Clock Description

- If the current state is i, the time to the next event is exponentially distributed with rate $-q_{ii}$ defined to be q_i.
- Given a transition occurs from state i, the probability that the transition is to state j is proportional to q_{ij}, namely $q_{ij} / \sum_{k \neq i} q_{ik}$.

Transition Probabilities

- For a continuous time Markov chain, the *transition matrix* whose ij element is the probability of being in state j at time t given the process begins in state i at time 0 is $P(t) = \mathrm{e}^{Qt}$.
- A probability transition matrix has non-negative values and each row sums to one.
- Each row contains the probabilities from a probability distribution on the possible states of the Markov process.

Examples

$$P(0.1) = \begin{pmatrix} 0.897 & 0.029 & 0.055 & 0.019 \\ 0.019 & 0.899 & 0.029 & 0.053 \\ 0.037 & 0.029 & 0.916 & 0.019 \\ 0.019 & 0.080 & 0.029 & 0.872 \end{pmatrix} \qquad P(0.5) = \begin{pmatrix} 0.605 & 0.118 & 0.199 & 0.079 \\ 0.079 & 0.629 & 0.118 & 0.174 \\ 0.132 & 0.118 & 0.671 & 0.079 \\ 0.079 & 0.261 & 0.118 & 0.542 \end{pmatrix}$$

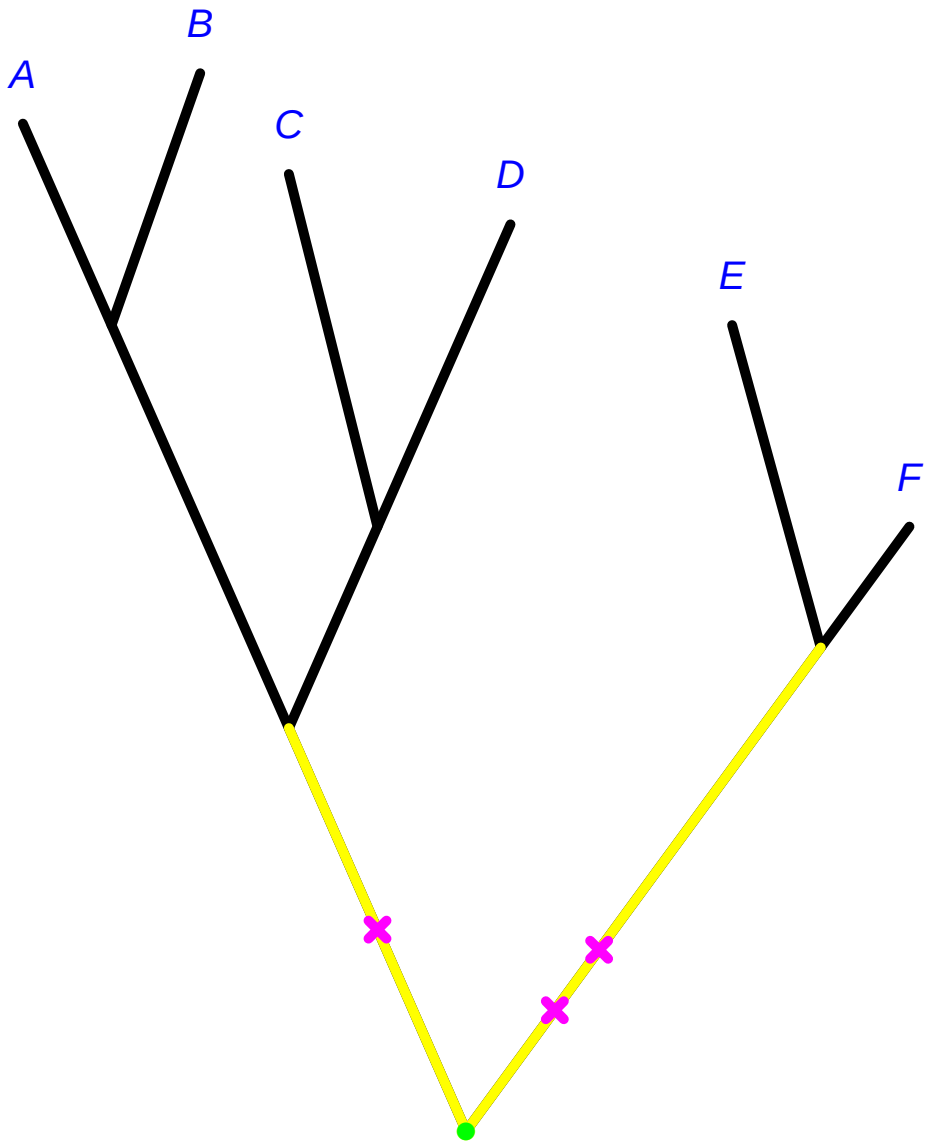
$$P(1) = \begin{pmatrix} 0.407 & 0.190 & 0.276 & 0.126 \\ 0.126 & 0.464 & 0.190 & 0.219 \\ 0.184 & 0.190 & 0.500 & 0.126 \\ 0.126 & 0.329 & 0.190 & 0.355 \end{pmatrix} \qquad P(10) = \begin{pmatrix} 0.200 & 0.300 & 0.300 & 0.200 \\ 0.200 & 0.300 & 0.300 & 0.200 \\ 0.200 & 0.300 & 0.300 & 0.200 \\ 0.200 & 0.300 & 0.300 & 0.200 \end{pmatrix}$$

The Stationary Distribution

- Well behaved continuous-time Markov chains have a *stationary distribution*, often designated π (not the constant close to 3.14 related to circles).
- When the time t is large enough, the probability $P_{ij}(t)$ will be close to π_j for each i. (See $P(10)$ from earlier.)
- The stationary distribution can be thought of as a long-run average—over a long time, the proportion of time the state spends in state i converges to π_i.

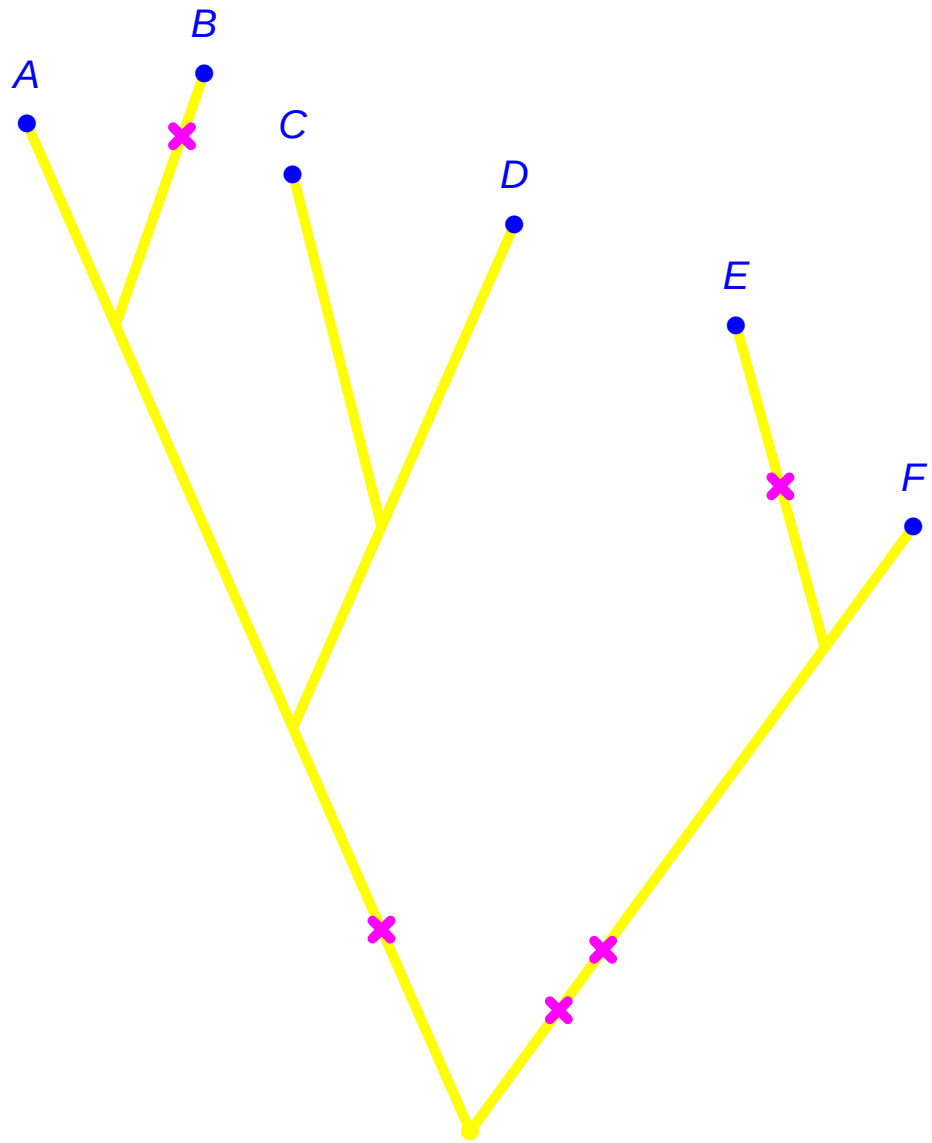
Markov Models on a Tree

- The value of the character at the root is either thought to be fixed and unknown or drawn from a probability distribution (typically the stationary distribution).
- Given the value at an internal node, the Markov process splits and continues independently up each edge.
- Each edge has a corresponding probability distribution.

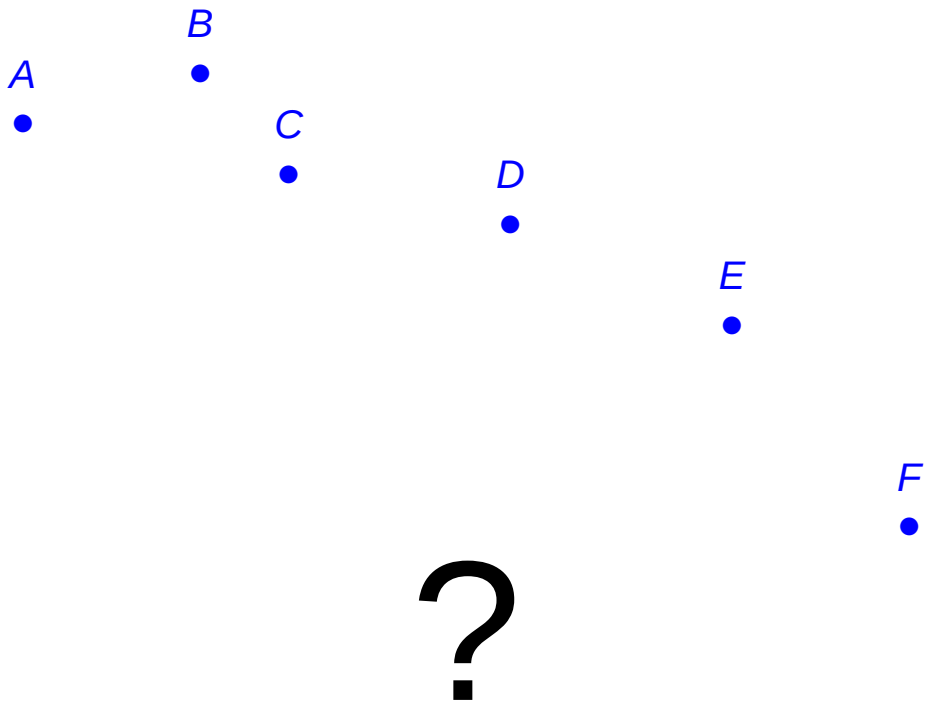


Model (cont.)

- The process continues to cover the entire tree.



Model (cont.)



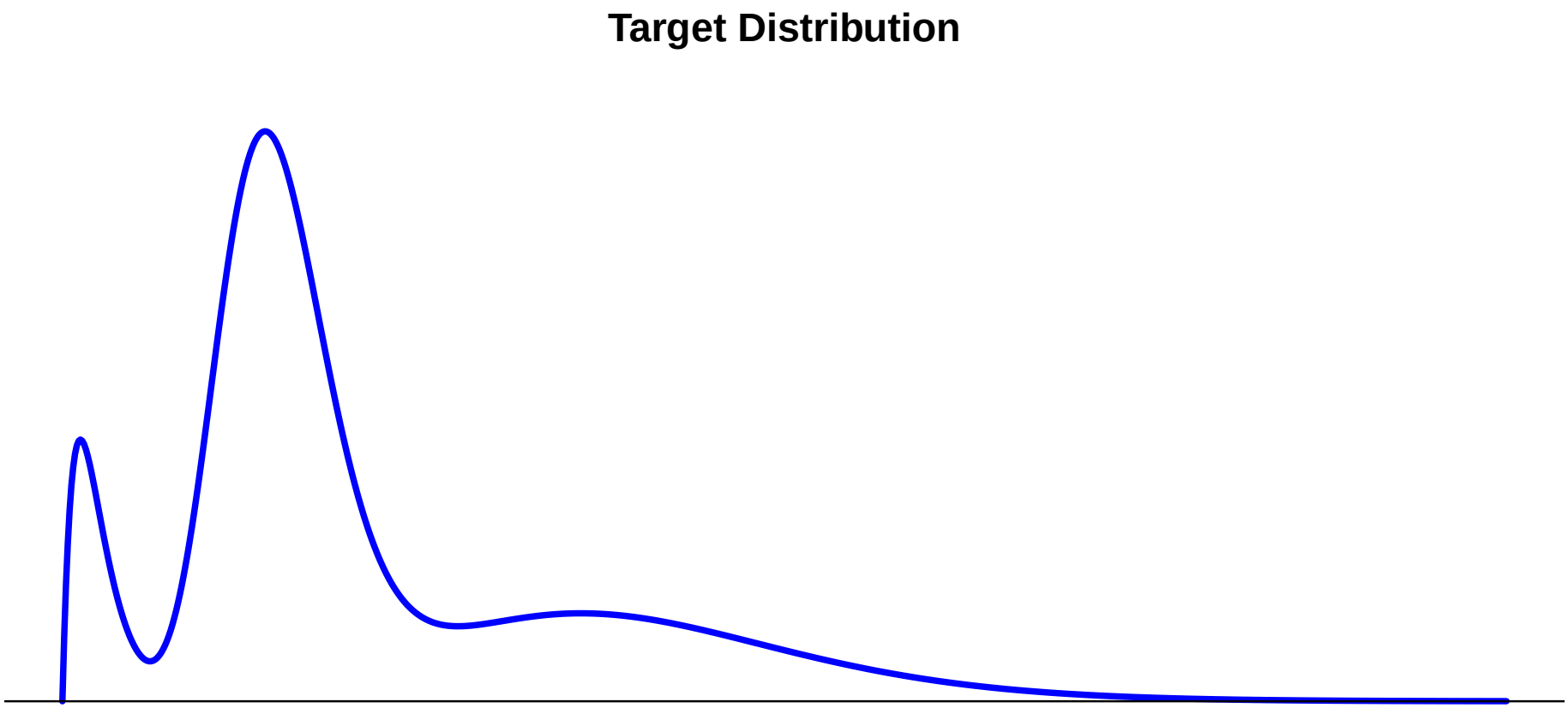
- We only observe data at the tips.
- The tree topology (shape and leaf labels), the edge lengths, and the history of genetic changes are unobserved.

Markov chain Monte Carlo

- *Markov chain Monte Carlo* (MCMC) is a very general method to sample from probability distributions by means of simulation.
- A *Markov chain* is a sequence of random variables where the distribution of each random variable depends only on the value of the previous random variable.
- Given the present, the future is independent of the past.
- The term *Monte Carlo* signifies a computer simulation of random numbers.
- We first demonstrate MCMC with an example.

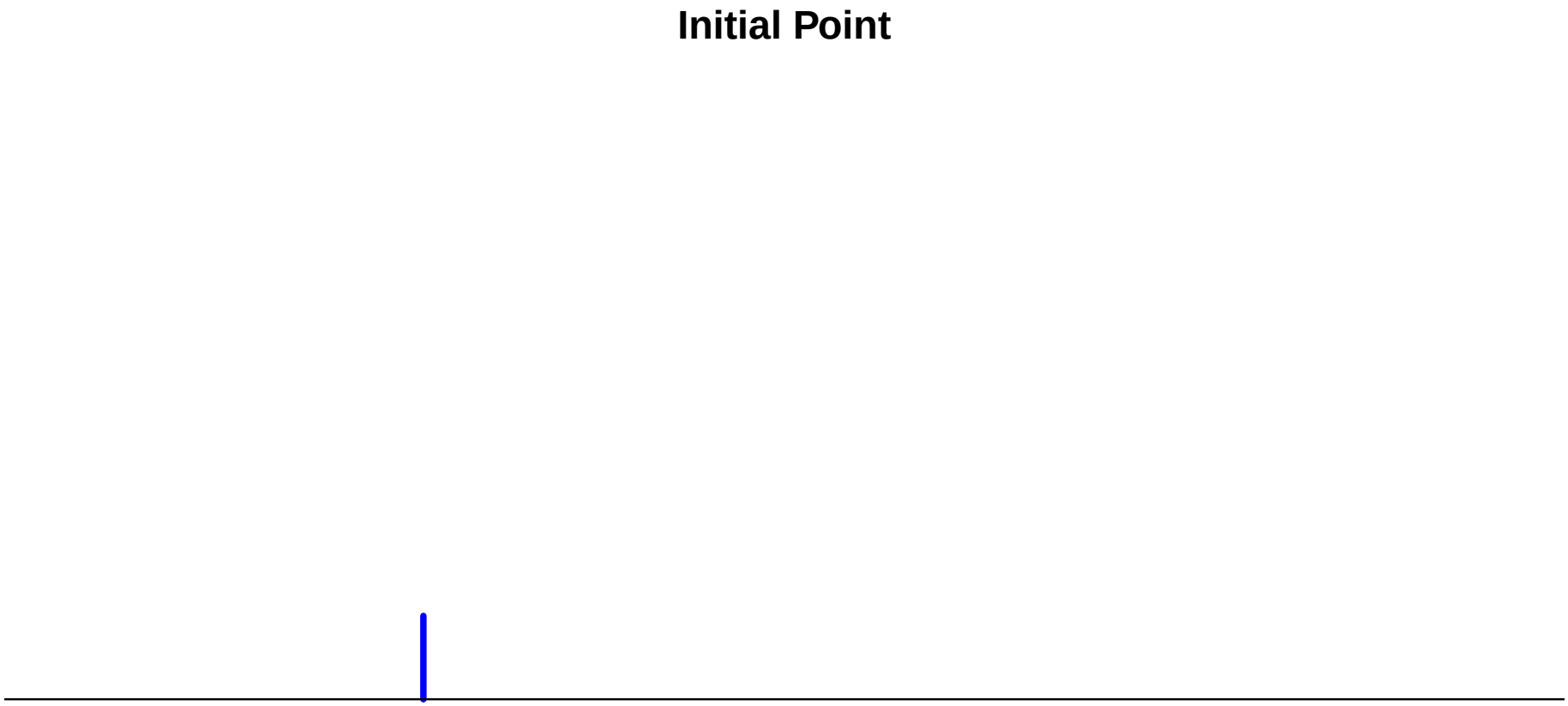
Example

- We have a function $h(\theta)$ from which we want to sample.
- We only need to know h up to a normalizing constant.



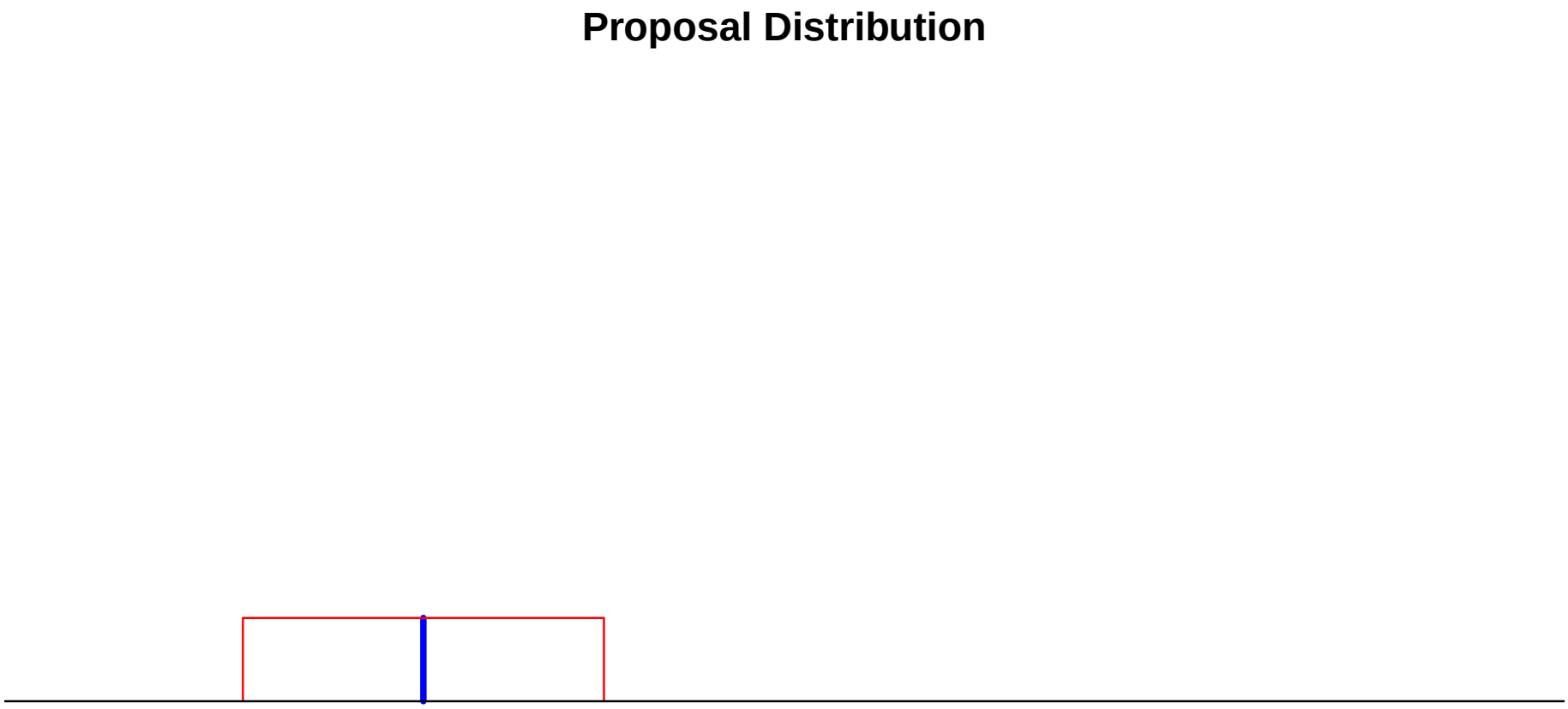
Initial Point

- We begin the Markov chain at a single point.
- We evaluate the value of h at this point.



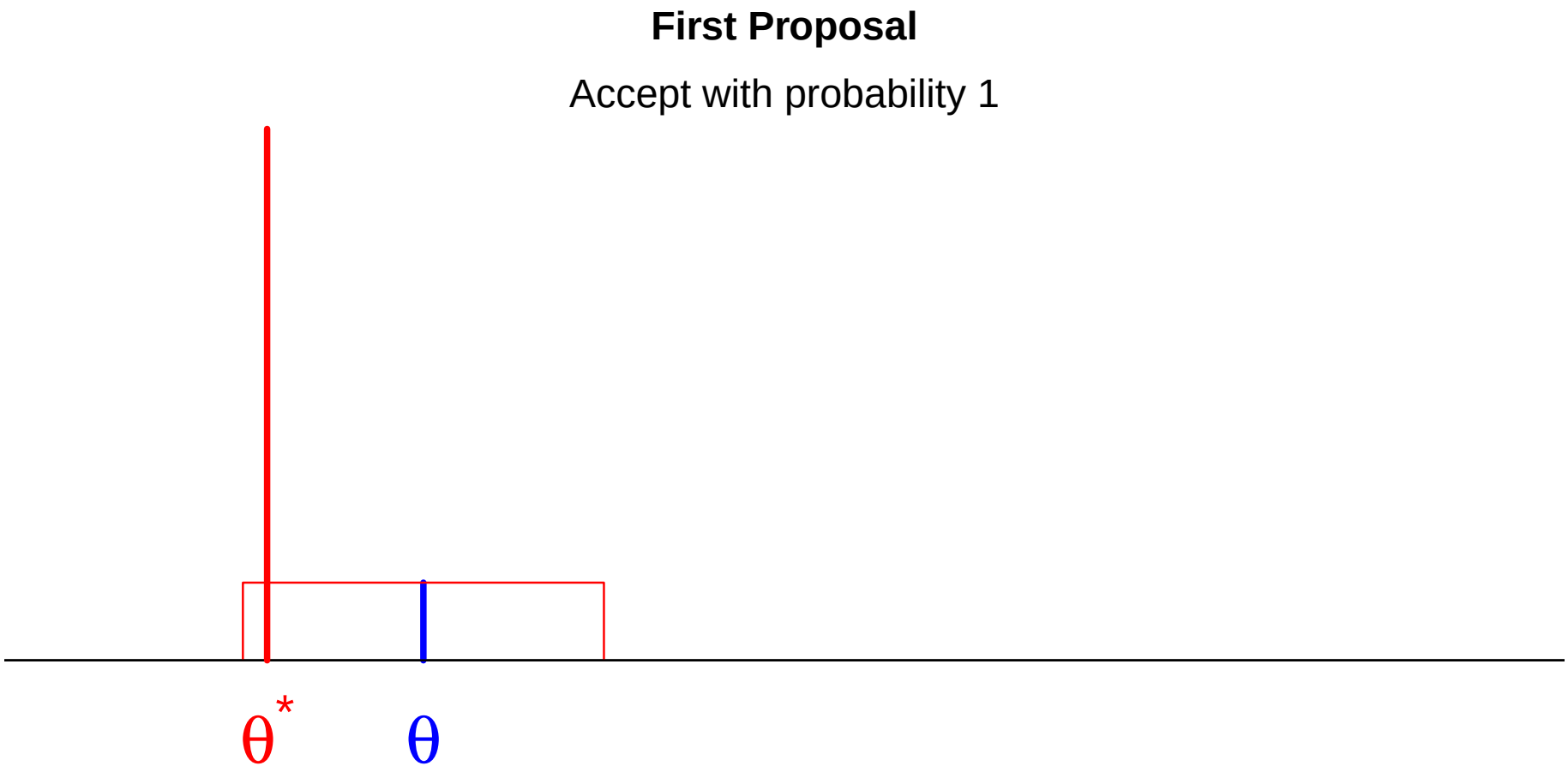
Proposal Distribution

- Given our current state, we have a proposal distribution for the next candidate state.



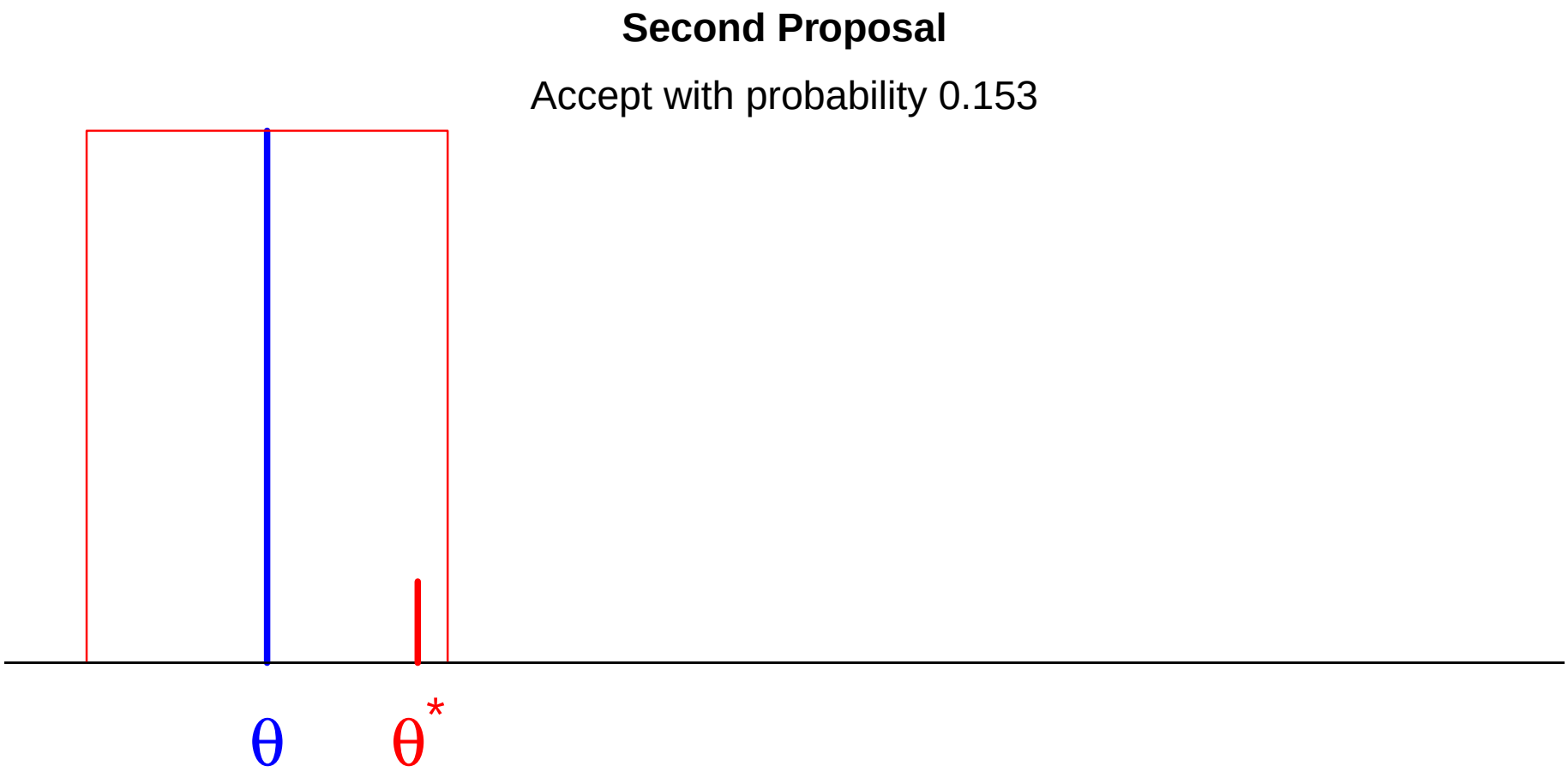
First Proposal

- We propose a *candidate* new point.
- Current state θ; Proposed state θ^*
- This proposal is accepted.



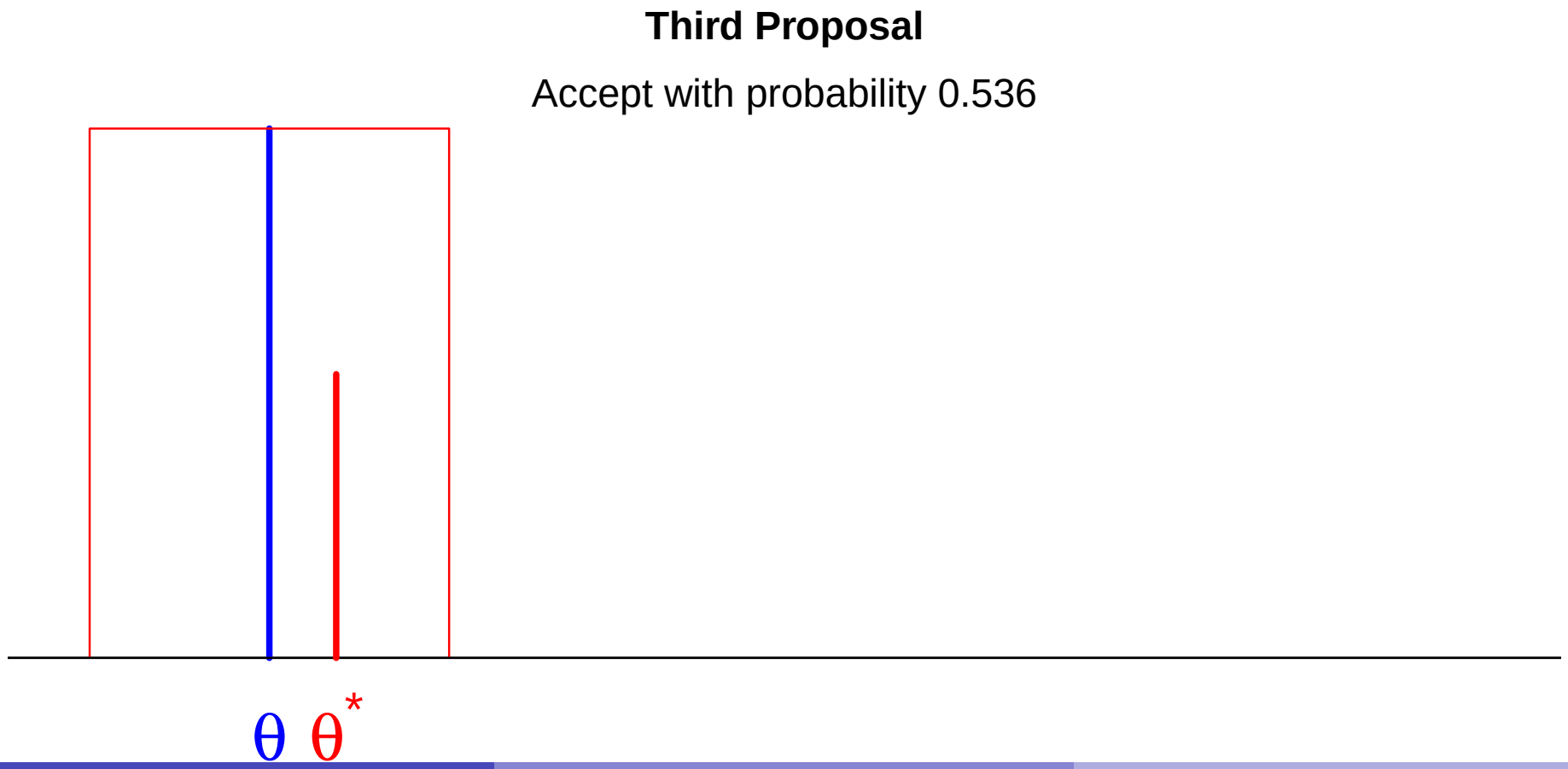
Second Proposal

- The proposal was accepted, so proposed state becomes current.
- Current state θ; Proposed state θ^*; Make another proposal.
- This proposal is rejected.



Third Proposal

- The proposal was rejected, so proposed state *is sampled again* and remains current.
- Current state θ; Proposed state θ^*; Make another proposal.
- This proposal is accepted.

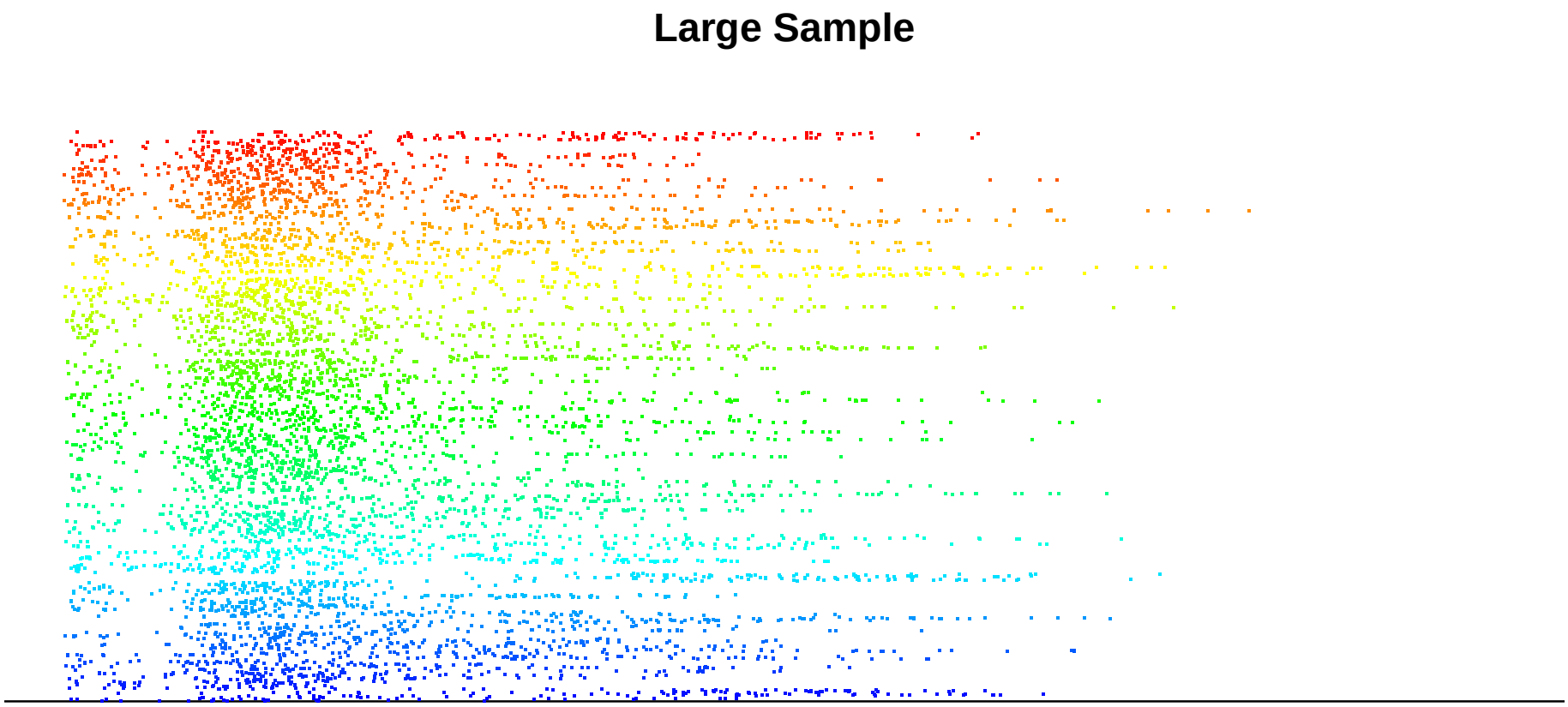


Beginning of Sample

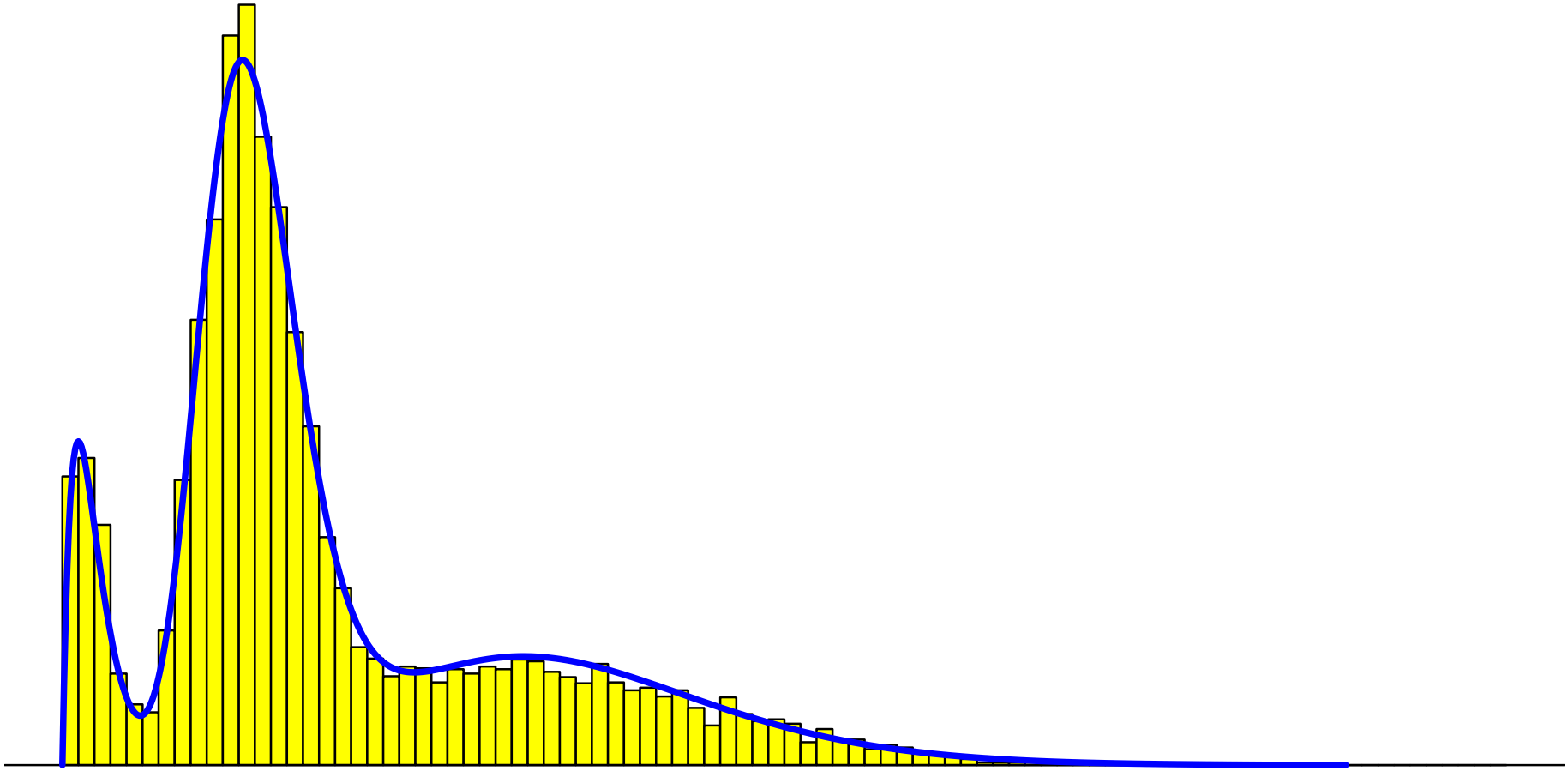
- The first four sample points.
- Vertical position is random to separate points at the same point.

Larger Sample

- Repeat this for 10,000 proposals and show the sample.



Comparison to Target



Things to Note

- The resulting sample mimics the target sample very well.
- The shape of the proposal distribution *did not depend on the target distribution at all*: almost any type of proposal method would have worked.
- There is a lot of *autocorrelation*: MCMC produces *dependent samples*.
- The acceptance probabilities depend on the proposal distributions and *relative* values of the target.
- Summaries of the sample are *good estimates* of corresponding target quantities:
  - The sample mean converges to the mean of the target.
  - The sample median converges to the median of the target.
  - The sample tail area above 1.0 converges to the relative area above 1.0 in the target.

Connections to the Paper

- In the paper, the model for the data includes:
  - A tree and a prior probability distribution on the tree.
  - A model for the evolution of discrete characters along the tree.
- The paper assumes that one or more trees have been sampled from the posterior distribution using MCMC from other data.
- The state space of the Markov chain includes:
  - full histories of character mapping onto each tree;
  - parameters for the substitution process.
- The basic principles of MCMC in this setting are identical to the previous simple example, but *the details are substantially more complicated*.

Possible Inferences

- The paper addresses methods to infer:
  - The process of evolution of the trait;
  - Frequency of cooccurence of trait values relative to a null hypothesis of independence;
  - Ancestral states.

Addressing Questions

When considering more than a single change along a branch on a tree, is the assumption that a morphological character can change back and forth multiple times between character states always true? Are the biological aspects of the morphological characters really being considered when the probability of the changes is based on molecular models of evolution?

- The methods described here assume nondirectional change processes.
- The methods could be extended by assuming, say, an ancestral state for each trait with a one-way mutation process.
- Such a Markov process would not be time-reversible.

Addressing Questions

As long branch lengths are problematic for character assignment/changes on a tree, are there cases where a branch length is sufficiently long that the character state can't be determined? If so, how is this reflected in the output data?

- In the model, long branches would correspond to weak information about the state of the trait in descendants, even given information from the ancestors.
- Taxa separated by long branches would be essentially independent.
- In an analysis of cooccurence of trait values, long branches can provide increased power to detect correlated evolution as the phyologeny would explain very little.

Addressing Questions

I don't have a particular question about the paper, because I could not understand very well how the stochastic mapping should be used. For example, I did not understand how Tables 5, 6, 7, and 8 should be interpreted.

- Table 5, for example, shows a summary of the number of events mapped onto the tree.
- In this example, each event was either a gain or a loss.
- A single mapping of a history onto a tree will contain some number of gains and losses.
- Each number is a posterior probability that the actual history had this many gains and losses.
- This is evaluated for two separate prior distributions.
- The posterior distributions are similar with one gain and two losses being most probable, but there is considerable uncertainty.

Table 5

TABLE 5. The probability distribution for the number of gains and losses for the aphid data. The maximum posterior probability estimates of the number of gains and losses are underlined. The 95% credible set of reconstructions is indicated by the bold numbers. Results for the low-rate prior on the tree length: $E(T) = 1$, $SD(T) = 5$. Results for the high-rate prior on the tree length: $E(T) = 10$, $SD(T) = 10$.

No. gains	No. losses										
	0	1	2	3	4	5	6	7	8	9	10
Low-rate prior											
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
1	0.000	**0.050**	**0.292**	**0.071**	**0.035**	**0.004**	**0.003**	0.001	0.001	0.001	0.000
2	**0.003**	**0.013**	**0.019**	**0.112**	**0.042**	**0.030**	**0.005**	**0.004**	0.001	0.001	0.001
3	**0.003**	**0.002**	**0.003**	**0.008**	**0.054**	**0.029**	**0.022**	**0.006**	**0.002**	0.001	0.001
4	**0.003**	0.000	0.001	0.001	**0.002**	**0.028**	**0.017**	**0.012**	**0.003**	**0.002**	0.001
5	0.001	0.001	0.000	0.000	0.001	**0.002**	**0.018**	**0.010**	**0.009**	0.001	**0.002**
6	0.000	0.000	0.000	0.000	0.000	0.000	0.001	**0.009**	**0.007**	**0.004**	0.001
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	**0.003**	**0.003**	**0.002**
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.002**	**0.002**
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
High-rate prior											
0	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.001**	0.000	0.000
1	0.000	**0.036**	**0.213**	**0.057**	**0.036**	**0.003**	**0.004**	**0.002**	**0.001**	0.000	0.000
2	**0.002**	**0.010**	**0.019**	**0.110**	**0.040**	**0.035**	**0.006**	**0.005**	**0.003**	**0.001**	0.000
3	**0.002**	**0.002**	**0.005**	**0.010**	**0.058**	**0.026**	**0.023**	**0.005**	**0.003**	**0.002**	**0.001**
4	**0.001**	**0.001**	**0.001**	**0.002**	**0.006**	**0.034**	**0.020**	**0.016**	**0.003**	**0.003**	**0.001**
5	0.000	**0.001**	**0.001**	**0.001**	**0.001**	**0.003**	**0.020**	**0.014**	**0.012**	**0.003**	**0.003**
6	0.000	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.013**	**0.008**	**0.008**	**0.002**
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.008**	**0.006**	**0.006**
8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.004**	**0.005**
9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**0.001**	**0.004**
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Addressing Questions

What software package(s) should I use to implement this method on my data?

- I suggest SimMap. How about a software demo?