

Short overview of MCMC methods in phylogenetics

Bastien Boussau

Bastien.boussau@univ-lyon1.fr

@bastounette

With many slides stolen from
Brian R. Moore, UC Davis



Bayesian inference

Bayes theorem:
$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

We are interested in the posterior probability of the parameter values: we want to find the most probable parameter values given the data.

Therefore we need to get the distribution for $P(\theta|D)$.

→ *Smart sampling using Markov Chain Monte Carlo (MCMC)*

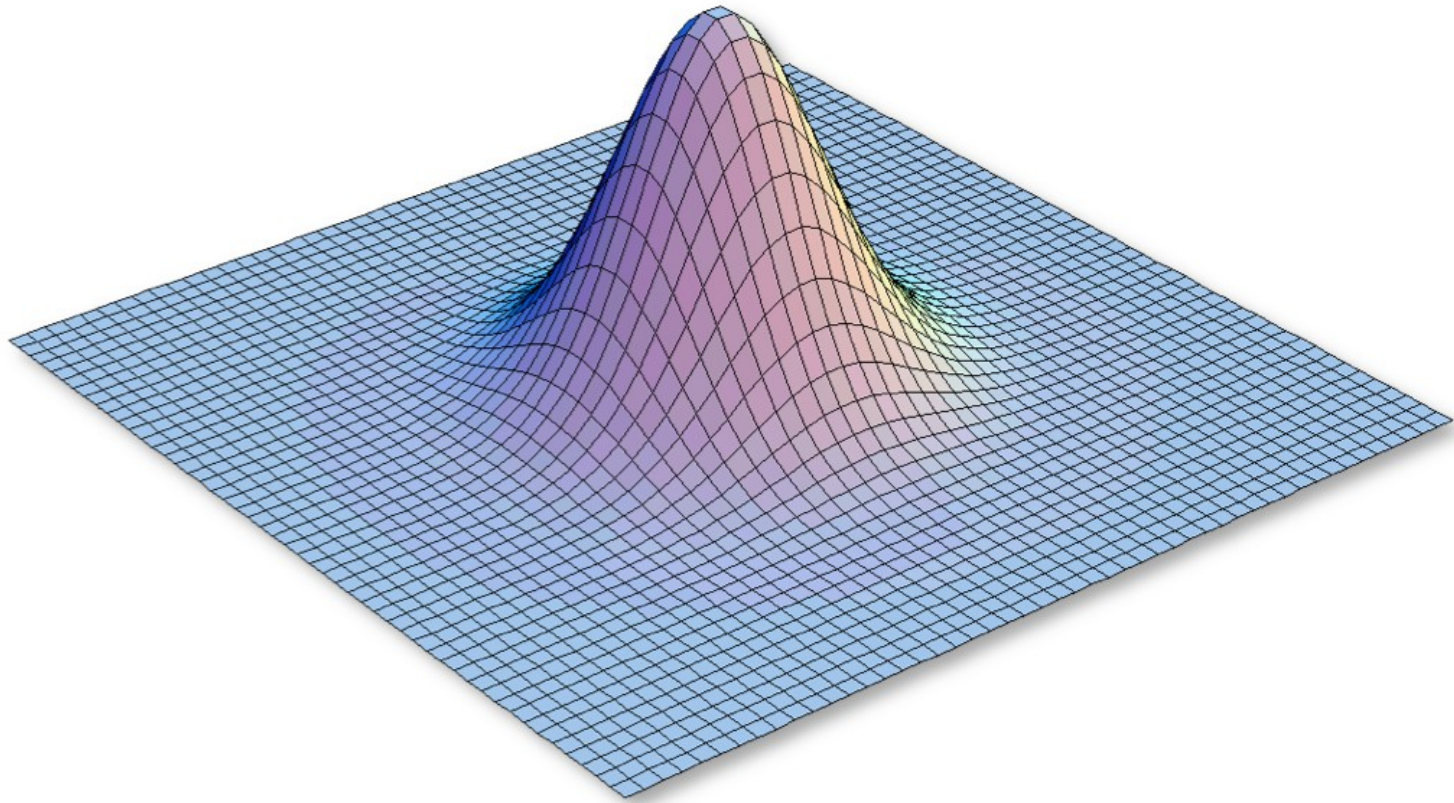
Smart sampling: MCMC sampling of $P(\theta/D)$

Aim: Sampling trees and parameter values with a frequency equal to their probability.

Idea: do a random sampling whose expectation is the target probability distribution. As a result, most samples should be in the high probability areas.

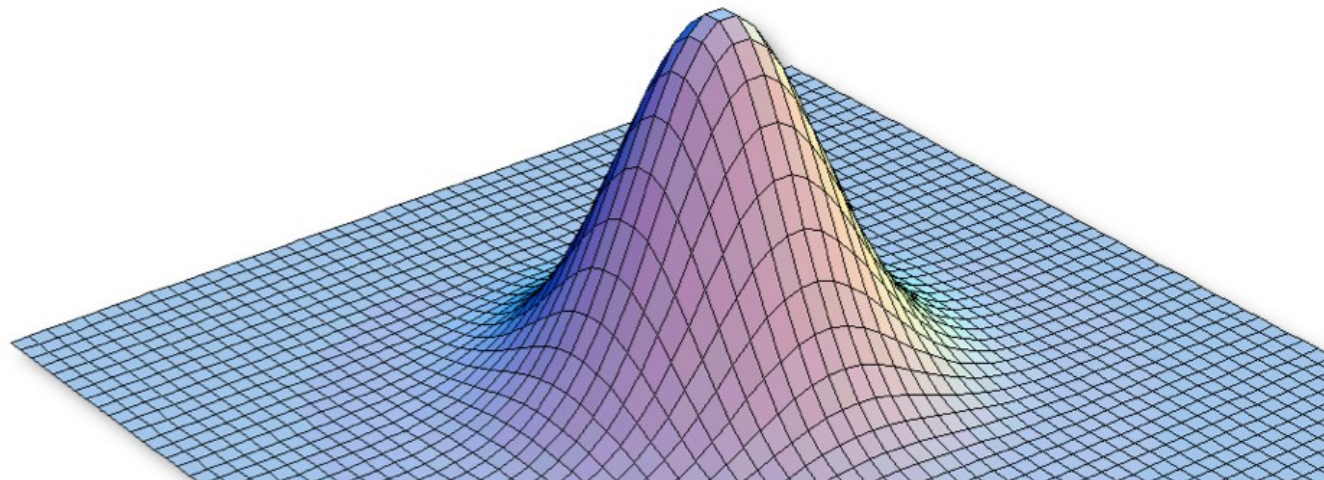
This is obtained by a Markov chain whose equilibrium value is the target distribution. Methods known as ***Markov Chain Monte Carlo (MCMC)***.

Approximating the Joint Posterior Probability Density using MCMC



Approximating the Joint Posterior Probability Density using MCMC

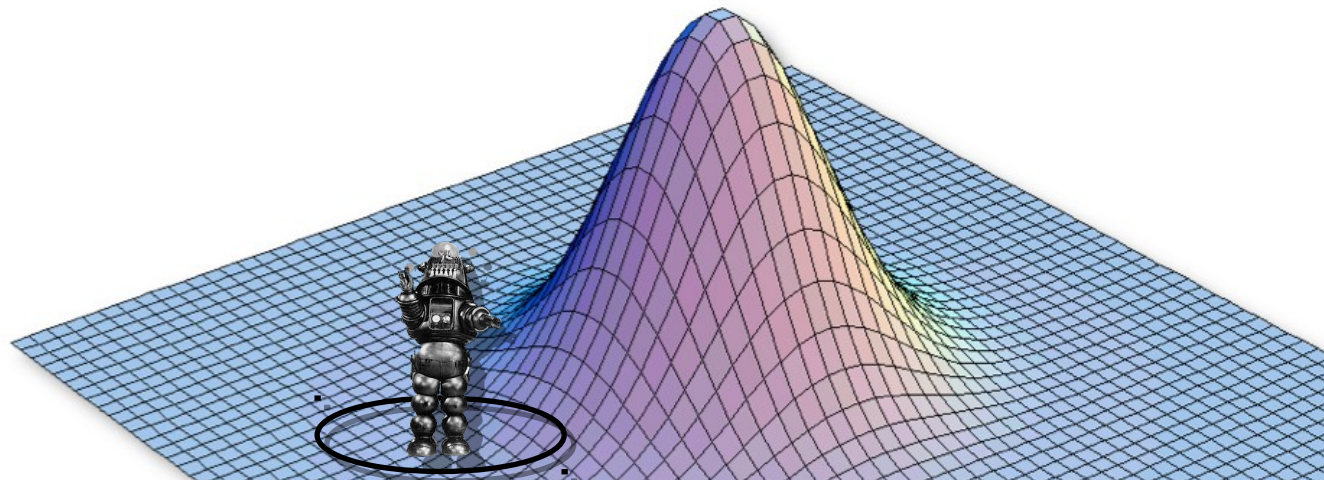
Programming our MCMC robot...



Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

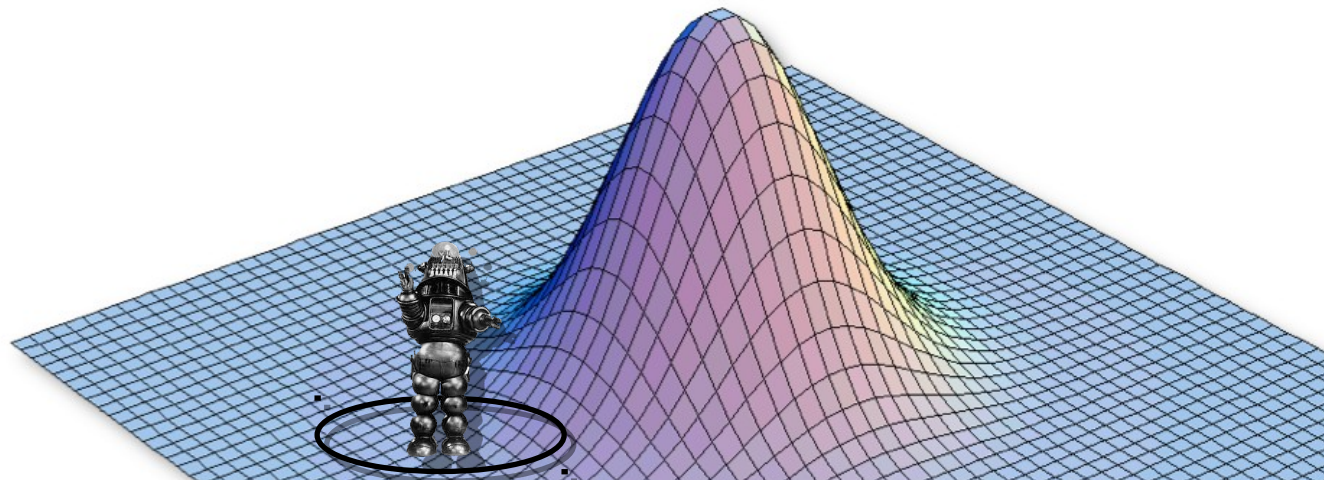


Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step



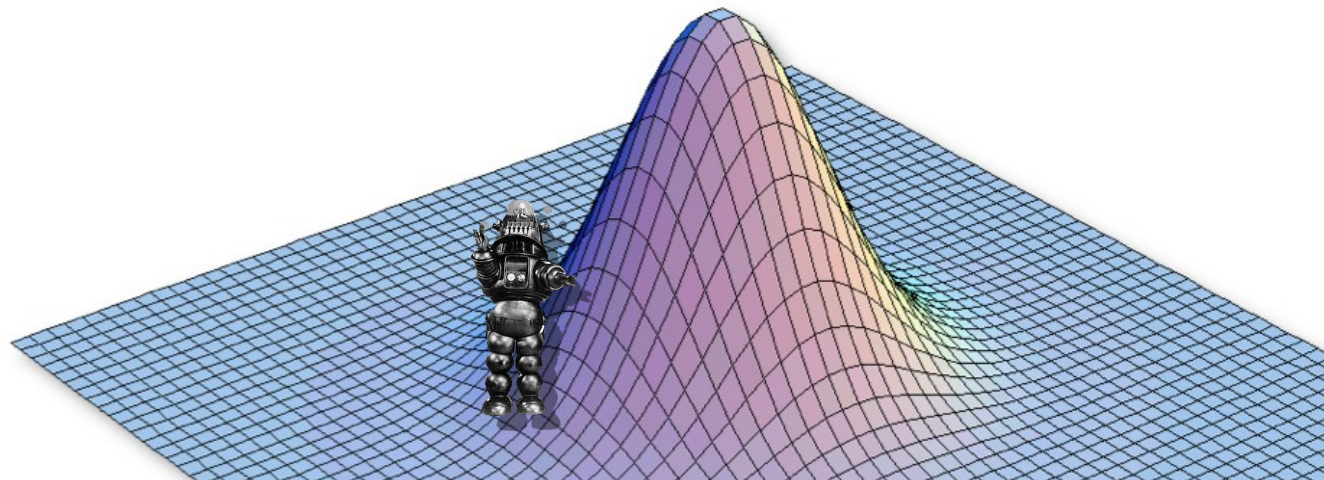
Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step

$$\Pr(\text{Accept}) = 1$$

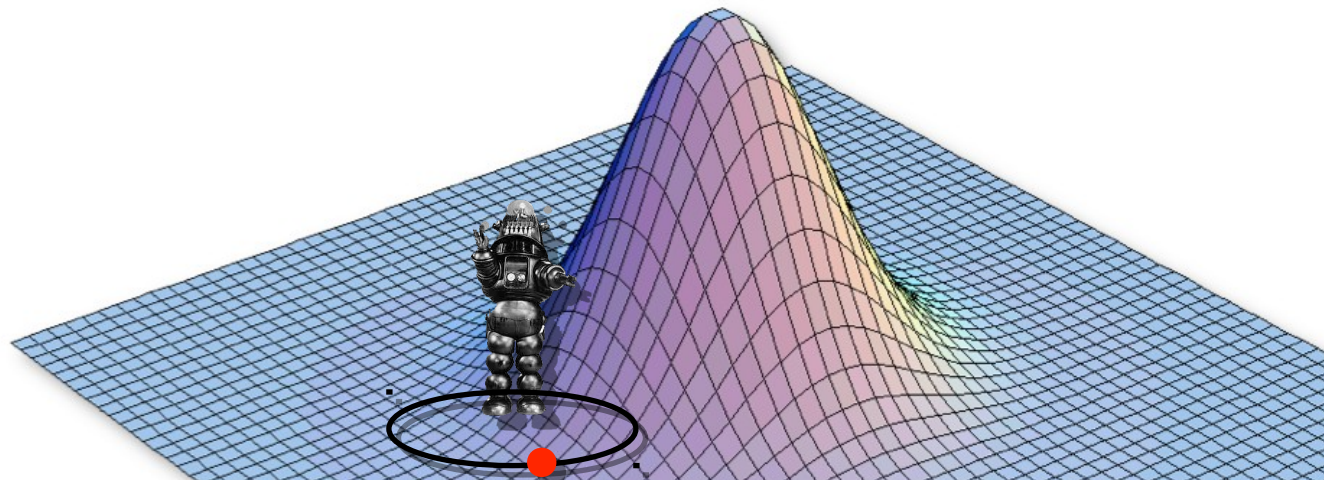


Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step
2. If the proposed step will take the robot downhill, it divides the elevation of the proposed location by the current location, and it only takes the step if the quotient is less than a uniform random variable, $\text{Uniform}[0,1]$



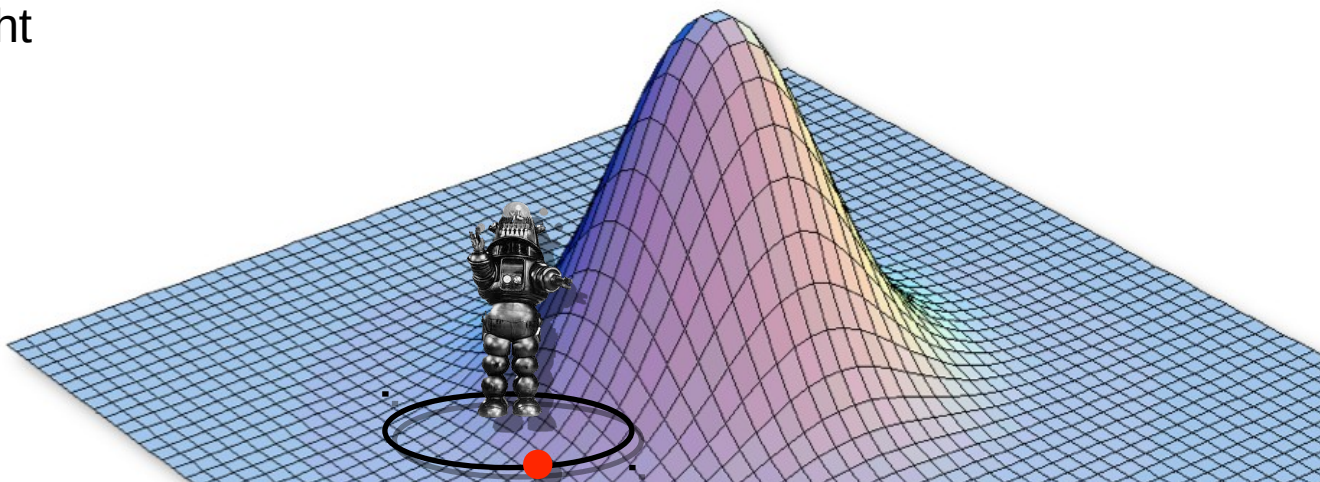
Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step
2. If the proposed step will take the robot downhill, it divides the elevation of the proposed location by the current location, and it only takes the step if the quotient is less than a uniform random variable, $\text{Uniform}[0,1]$

$$\text{Pr}(\text{Accept}) = \text{new height}/\text{old height}$$



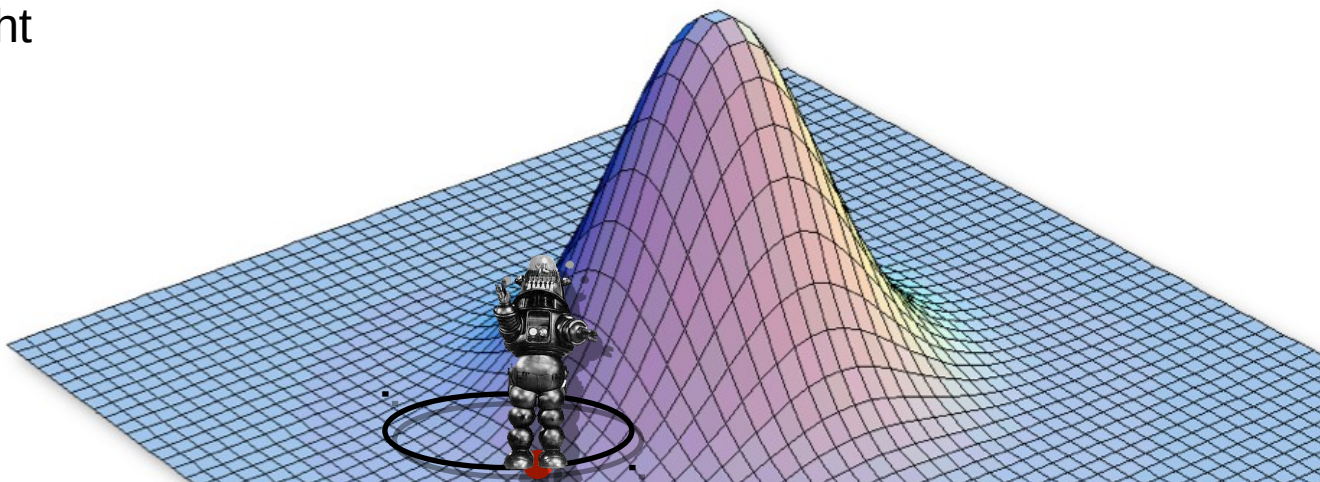
Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step
2. If the proposed step will take the robot downhill, it divides the elevation of the proposed location by the current location, and it only takes the step if the quotient is less than a uniform random variable, $\text{Uniform}[0,1]$
3. The proposal distribution is symmetrical, so $P(\theta \rightarrow \theta') = P(\theta' \rightarrow \theta)$

$\text{Pr}(\text{Accept}) = \text{new height}/\text{old height}$

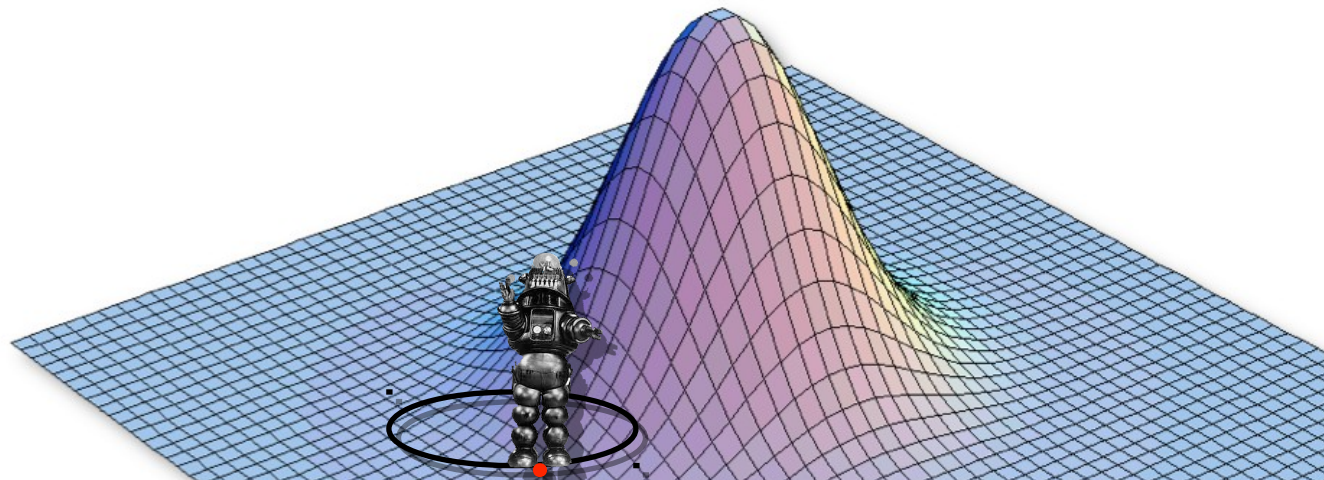


Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step
2. If the proposed step will take the robot downhill, it divides the elevation of the proposed location by the current location, and it only takes the step if the quotient is less than a uniform random variable, $\text{Uniform}[0,1]$
3. The proposal distribution is symmetrical, so $P(\theta \rightarrow \theta') = P(\theta' \rightarrow \theta)$

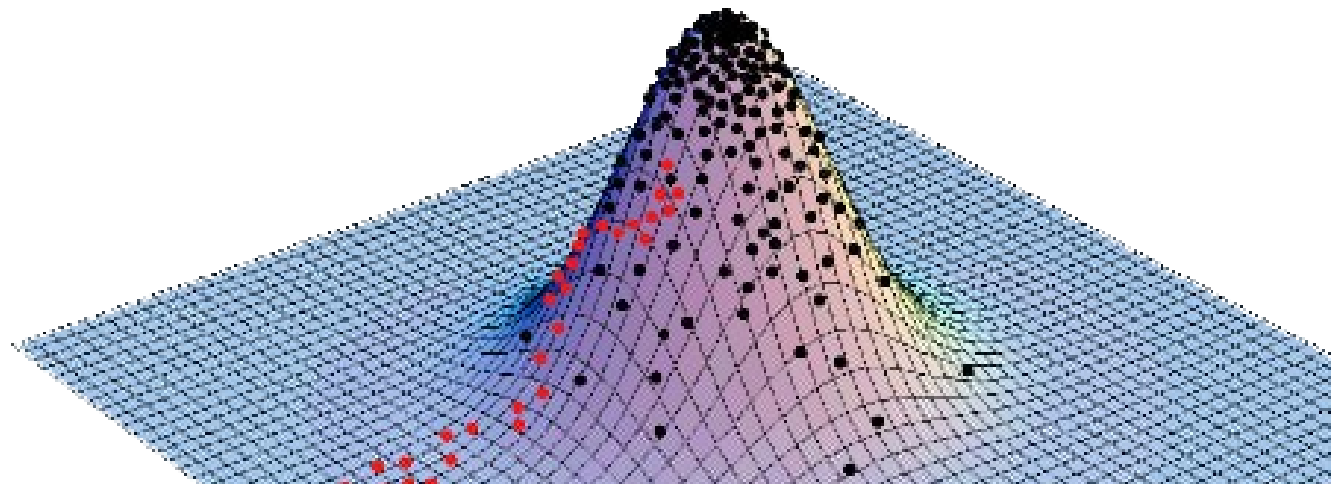


Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

1. If the proposed step will take the robot uphill, it automatically takes the step
2. If the proposed step will take the robot downhill, it divides the elevation of the proposed location by the current location, and it only takes the step if the quotient is less than a uniform random variable, $\text{Uniform}[0,1]$
3. The proposal distribution is symmetrical, so $P(\theta \rightarrow \theta') = P(\theta' \rightarrow \theta)$



MCMC in phylogenetics

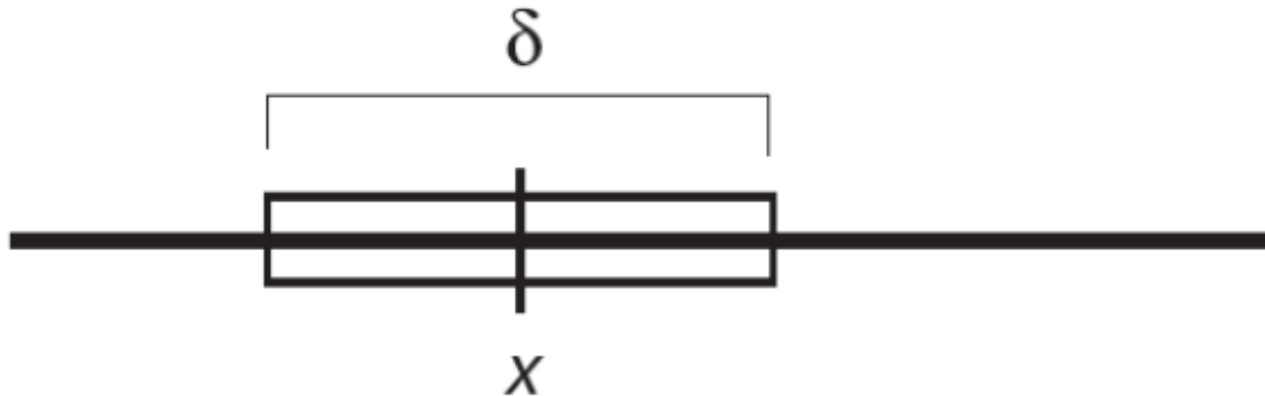
- MCMC is an approach for sampling from a probability distribution
- It is often used in the Bayesian setting, where one wants to sample from a Posterior probability distribution
- In phylogenetics, one uses it to sample:
 - Phylogenetic trees (when we don't care about dates):
 - topologies
 - branch lengths
 - Chronograms (when dates are of interest)
 - Rates:
 - Rates of exchangeability
 - Site-wise rates of evolution
 - Birth-death rates
 - ...
 - Frequencies:
 - ACGT equilibrium frequencies
 - Root frequencies
 - Other parameters...

MCMC in phylogenetics

Parameter	Prior (example)	Move (example)
Topology	Uniform...	NNI, SPR, TBR...
Branch lengths	Exponential, Gamma+Dirichlet...	Scaling, Sliding...
Chronogram	Birth-Death, Coalescent...	SS, FNPR, NodeTimeSlide...
Rates of exchangeability	Dirichlet...	Dirichlet moves...
Site-wise rates of evolution	(Discretized) Gamma...	Scaling, Sliding...
Birth-death rates	Lognormal, Exponential...	Scaling, Sliding...
ACGT equilibrium frequencies	Dirichlet...	Dirichlet moves...
Root frequencies	Dirichlet...	Dirichlet moves...

Sliding move

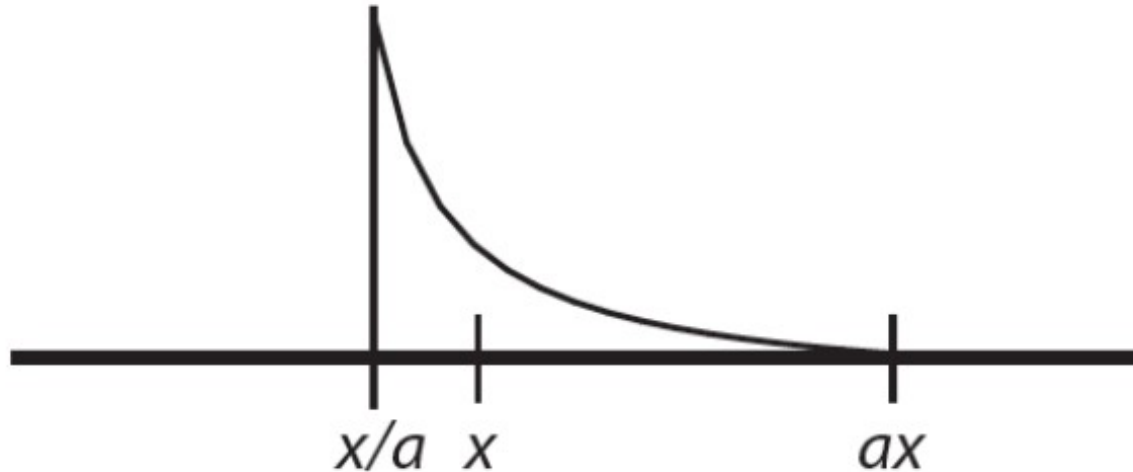
- New values are picked uniformly in an interval of size δ centered on the current value



- Increase δ : bolder moves
- Decrease δ : more modest moves

Scaling move

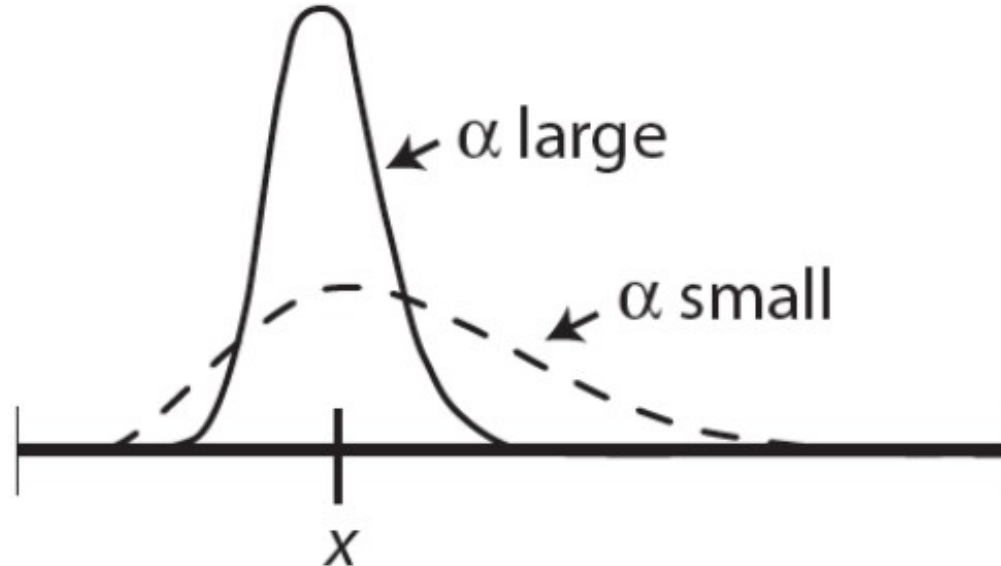
- Like a sliding window, but on the log-transformed parameter



- Tuning parameter: $\lambda = 2 \ln a$
- Increase λ : bolder moves
- Decrease λ : more modest moves

Dirichlet move

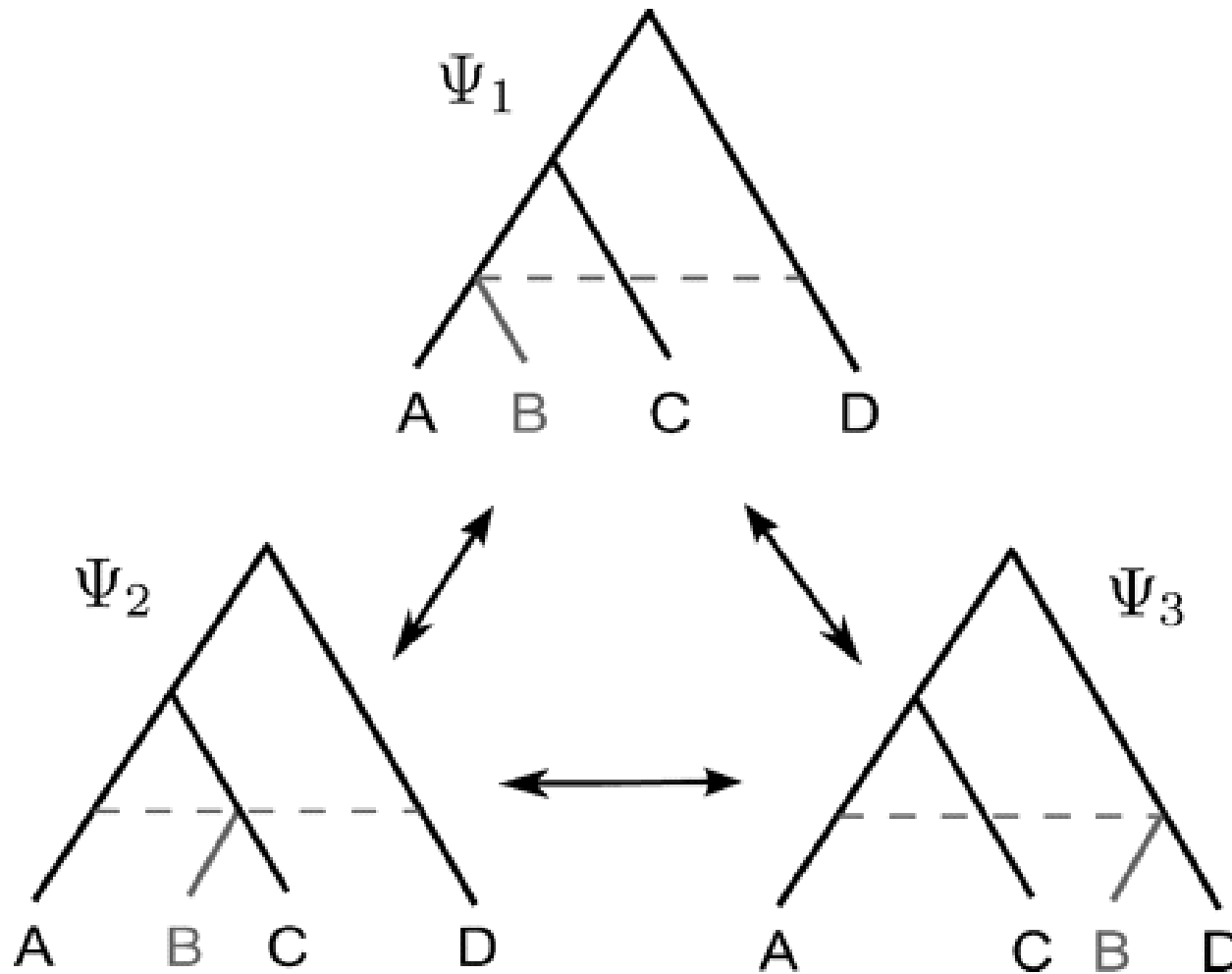
- New values are picked from a Dirichlet (or Beta) distribution centered on x



- Tuning parameter: α
- Bolder proposals: decrease α
- More modest proposals: increase α

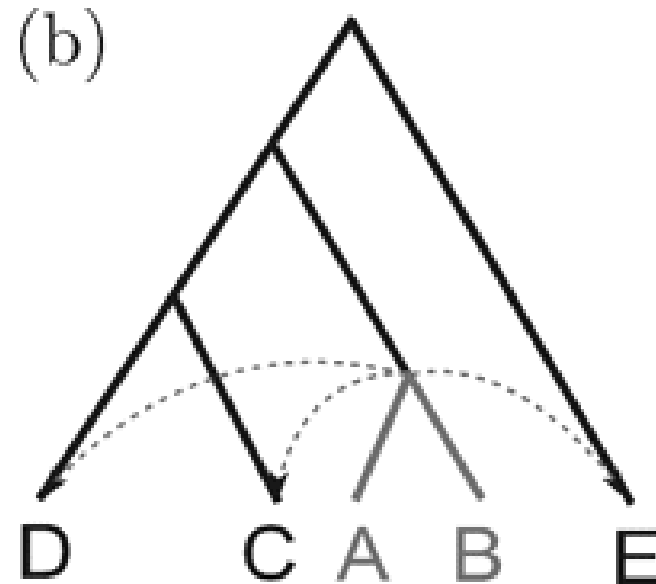
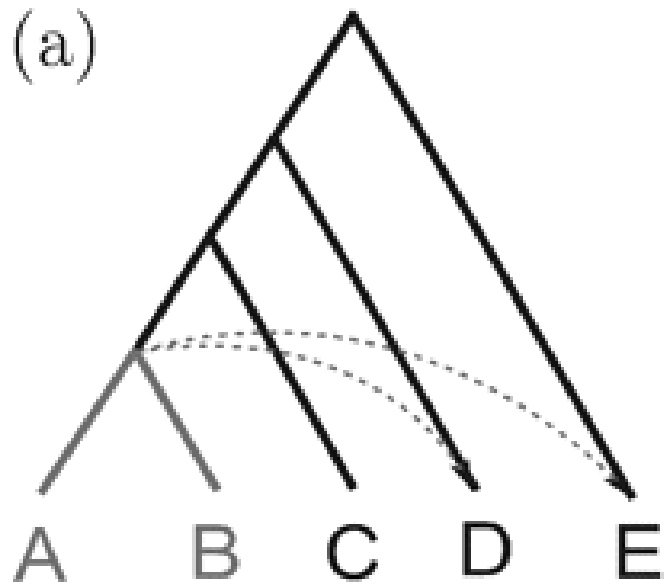
Chronogram moves

- Fixed Node height Prune and Regraft (FNPR)



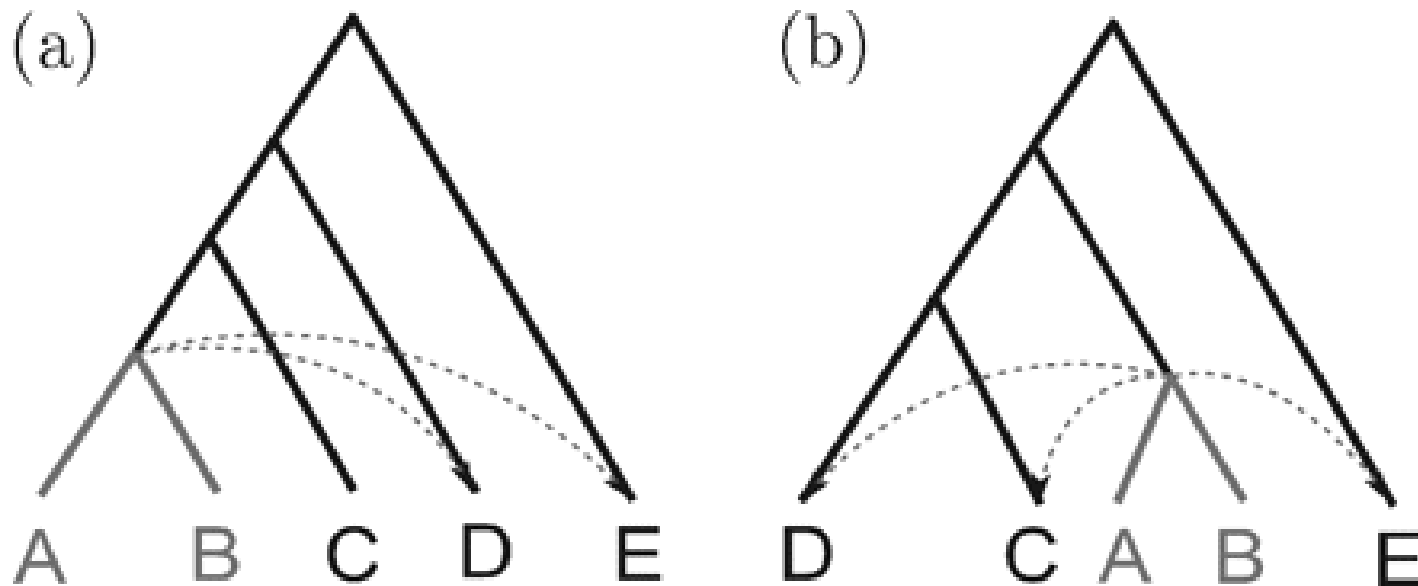
Chronogram moves (2)

- Subtree swap



Chronogram moves (2)

- Subtree swap



Here the Hastings ratio needs to be computed to account for the bias in transition probabilities between (a) and (b)

Conclusion

- MCMC is an approach for sampling from a probability distribution
- It is often used in the Bayesian setting, where one wants to sample from a Posterior probability distribution
- In phylogenetics, one uses it to sample:
 - Phylogenetic trees: topologies and branch lengths
 - Divergence times
 - Rates
 - Other parameters...
- Different moves have been designed for each of these parameters
- Some joint moves operate on several parameters at the same time
- Efficiency of the MCMC relies heavily on those moves
- Other algorithms like Metropolis Coupled Markov Chain Monte Carlo (MC³) have been proposed and can help