A Brief Introduction to MCMC

Brian R. Moore & Sebastian Höhna
Department of Evolution and Ecology
University of California, Bervis
Bodega Workshop, 2017

Outline

I. The Bayesian hard sell

What's the deal with priors?

Learning to embrace your inner Bayesian

II. Approximating the posterior probability with MCMC

Metropolis-Hastings algorithm

Metropolis-coupled algorithm

Maximum Likelihood *and* Bayesian Inference

What is there to disagree about?

Not much, actually:

- model-based statistical inference
- observations are random variables
- likelihood function extracts information from data to estimate parameters

Maximum Likelihood and Bayesian Inference

Lots to love about maximum-likelihood estimation

Desirable statistical properties

- consistent estimator
- asymptotically efficient estimator

Explicit with respect to model assumptions

Convenient and objective model selection/hypothesis testing framework

Some less desirable aspects of maximum-likelihood estimation

Non-intuitive meaning of likelihood

Frequentist perspective can be awkward for some inference problems

Less amenable to EDA scenarios

Accommodating uncertainty can be less than natural

Maximum Likelihood and Bayesian Inference

Maximum-likelihood perspective on parameters:

Data are random variables, but the parameters are fixed

Bayesian perspective on parameters:

Data are random variables, and so are the model parameters

If we treat the parameters are random variables, what do we have to specify?

Bayesian Inference

A priori…

We usually (*i.e.*, always) have prior beliefs, so why not be explicit about it?

- this is consistent with making assumptions clear (model-based inference)

When relevant prior information is available, it can be naturally incorporated

- this is consistent with the way we behave as rational beings

It can be non-trivial to specify our prior beliefs as probability distributions

- we might attempt to define vague priors in some cases
- we can (and should) assess the impact of our prior assumptions

Concerns about the prior sensitivity are largely philosophical

- the posterior is typically dominated by the likelihood function
- when this is not the case, the ability to detect prior sensitivity is a good thing!

Outline

I. The Bayesian hard sell

What's the deal with priors?

Learning to embrace your inner Bayesian

II. Approximating the posterior probability with MCMC

Metropolis-Hastings algorithm

Metropolis-coupled algorithm

Bayesian Inference of Phylogeny (on one slide)

$$\underbrace{\boxed{f(\text{Parameter} \mid \text{Data})}}_{\text{posterior probability}} = \frac{\overbrace{\boxed{f(\text{Data} \mid \text{Parameter})}}^{\text{likelihood function}}\ \overbrace{\boxed{f(\text{Parameter})}}^{\text{prior probability}}}{\underbrace{\boxed{f(\text{Data})}}_{\text{marginal likelihood}}}$$

I. Data

Assume an alignment, $\mathbf{X}$, of N sites for S species: $\mathbf{X} = (x_1, x_2, x_3, \ldots, x_N)$

II. Phylogenetic model parameters

1. Tree topology $\tau = (\tau_1, \tau_2, \ldots, \tau_{(2s-5)!!})$

   branch lengths $\nu = (\nu_1, \nu_2, \ldots, \nu_{(2S-3)})$

2. Model of character change $\Phi = (\theta, \pi, \alpha, T)$

   relative substitution rates $\theta = (\theta_{AC}, \theta_{AG}, \theta_{AT}, \theta_{CG}, \theta_{CT}, \theta_{GT})$

   stationary frequencies $\pi = (\pi_A, \pi_C, \pi_G, \pi_T)$

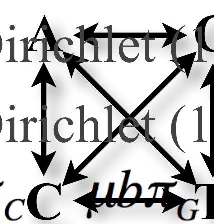
IV. Priors on parameters

~Uniform

~Dirichlet (1,...,1)

~Dirichlet (1,1,1,1,1,1)

~Dirichlet (1,1,1,1)



$$Q = q_{ij} = \begin{pmatrix} - & \mu a\pi_C & \mu b\pi_G & \mu c\pi_T \\ \mu a\pi_A & - & \mu d\pi_G & \mu e\pi_T \\ \mu b\pi_A & \mu d\pi_C & - & \mu f\pi_T \\ \mu c\pi_A & \mu e\pi_C & \mu f\pi_G & - \end{pmatrix}$$

III. Phylogenetic likelihood function

$$L(\tau, \nu, \Theta) \propto f(\mathbf{X} \mid \tau, \nu, \Theta) = \prod_{i=1}^{N} f(x_i \mid \tau, \nu, \Theta)$$

V. Posterior Probability

$$f(\tau, \nu, \Phi \mid \mathbf{X}) = \frac{f(\mathbf{X} \mid \tau, \nu, \Phi)\, f(\tau, \nu, \Phi)}{f(\mathbf{X})}$$

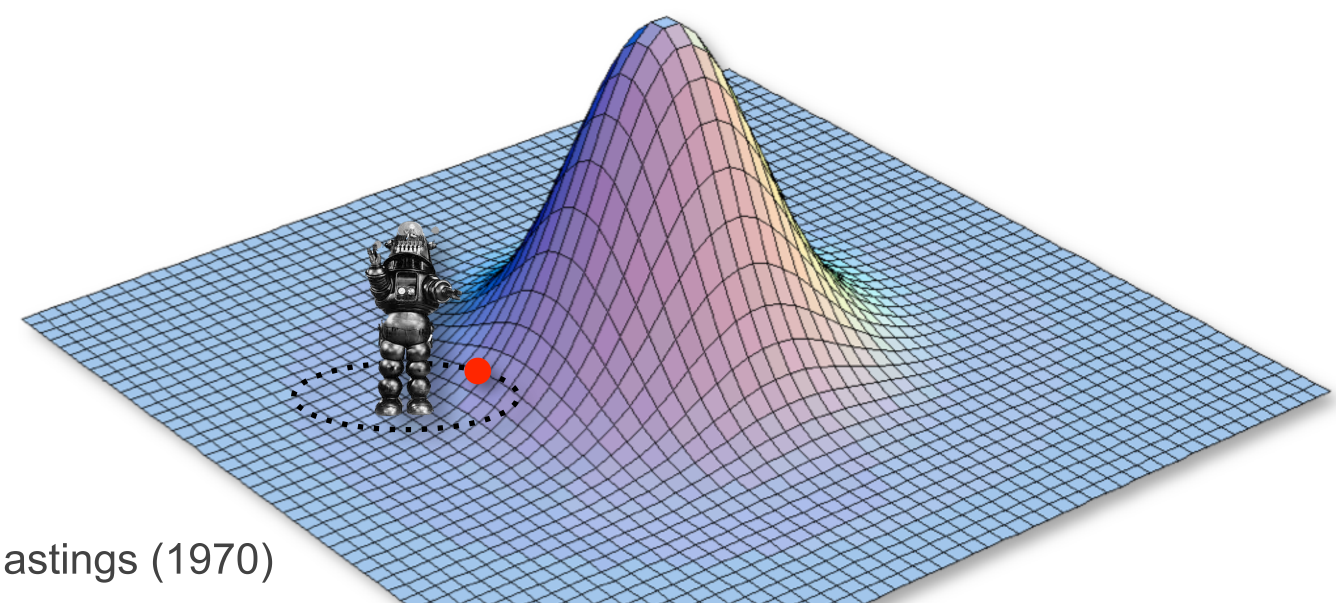
Approximating the Joint Posterior Probability Density using MCMC

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[Accept] = 1



Metropolis et al. (1953); Hastings (1970)

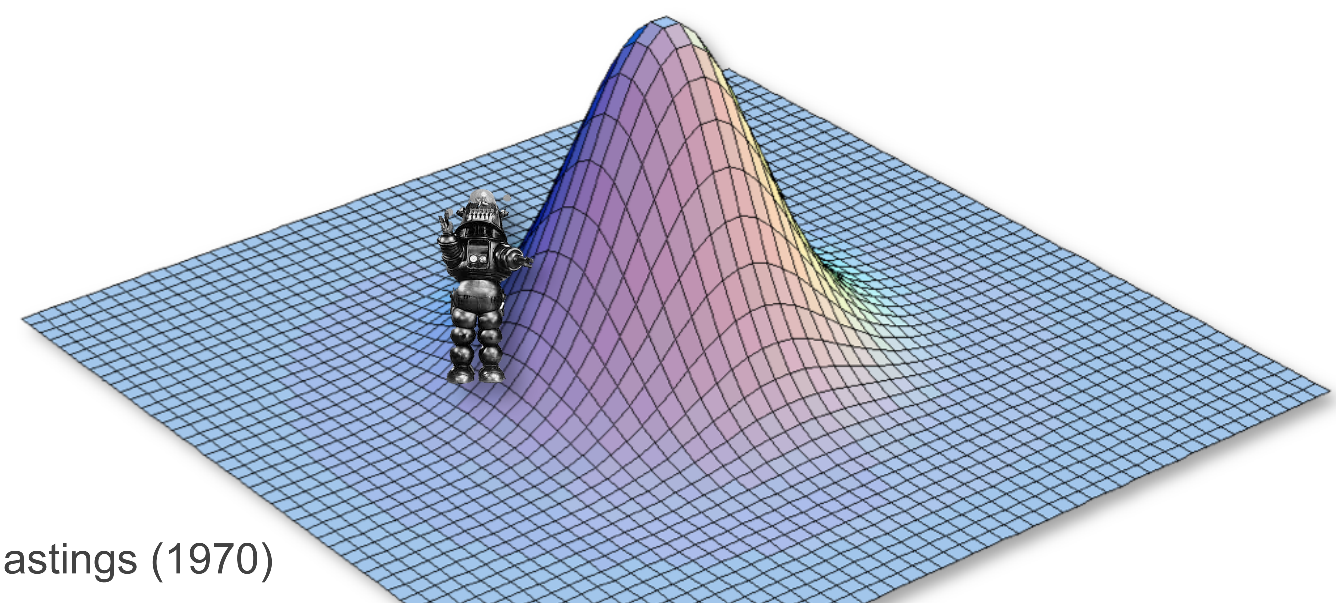
Approximating the Joint Posterior Probability Density using MCMC

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[Accept] = 1



Metropolis et al. (1953); Hastings (1970)

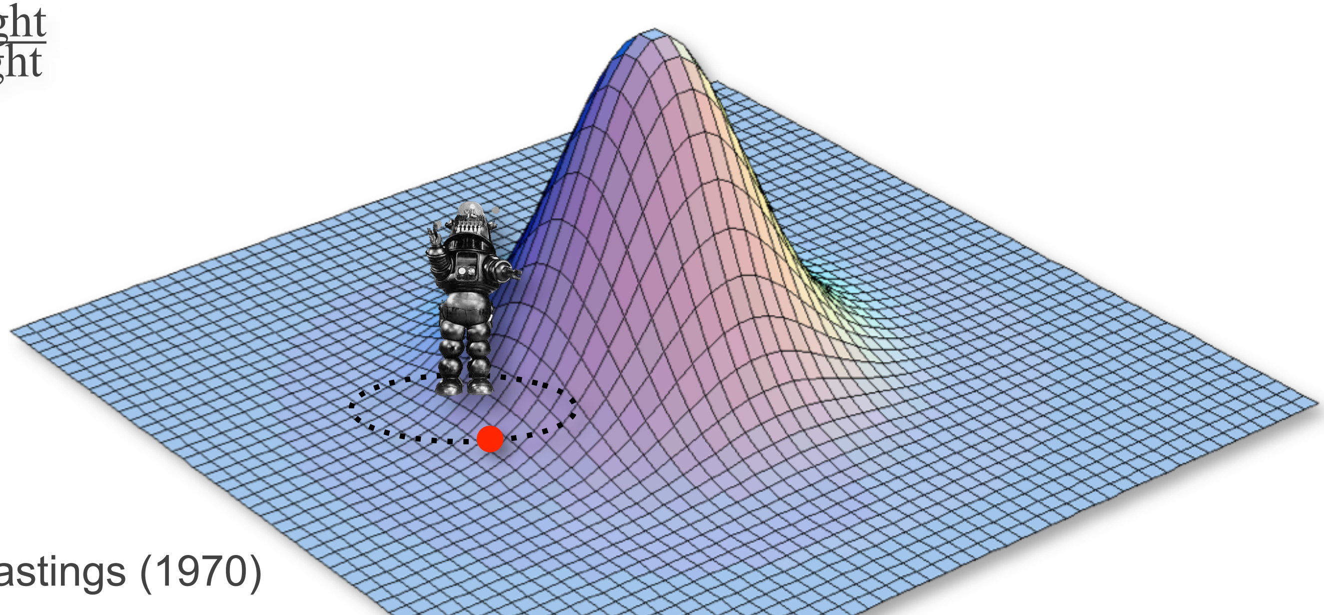
Approximating the Joint Posterior Probability Density using MCMC

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, $\mathrm{Uniform}[0,1]$

$$\Pr[\text{Accept}] = \frac{\text{new height}}{\text{old height}}$$



Metropolis et al. (1953); Hastings (1970)

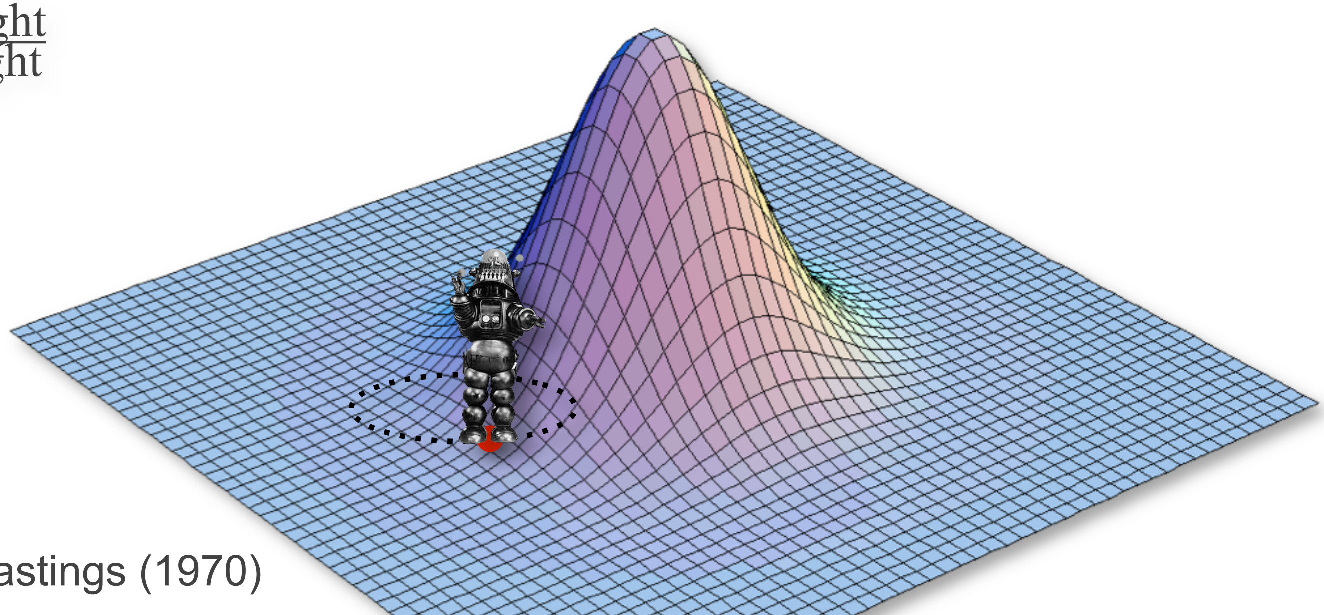
Approximating the Joint Posterior Probability Density using MCMC

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, $\mathrm{Uniform}[0,1]$
3. The proposal distribution is symmetrical, so $\Pr[A \rightarrow B] = \Pr[B \rightarrow A]$

$$\Pr[\text{Accept}] = \frac{\text{new height}}{\text{old height}}$$



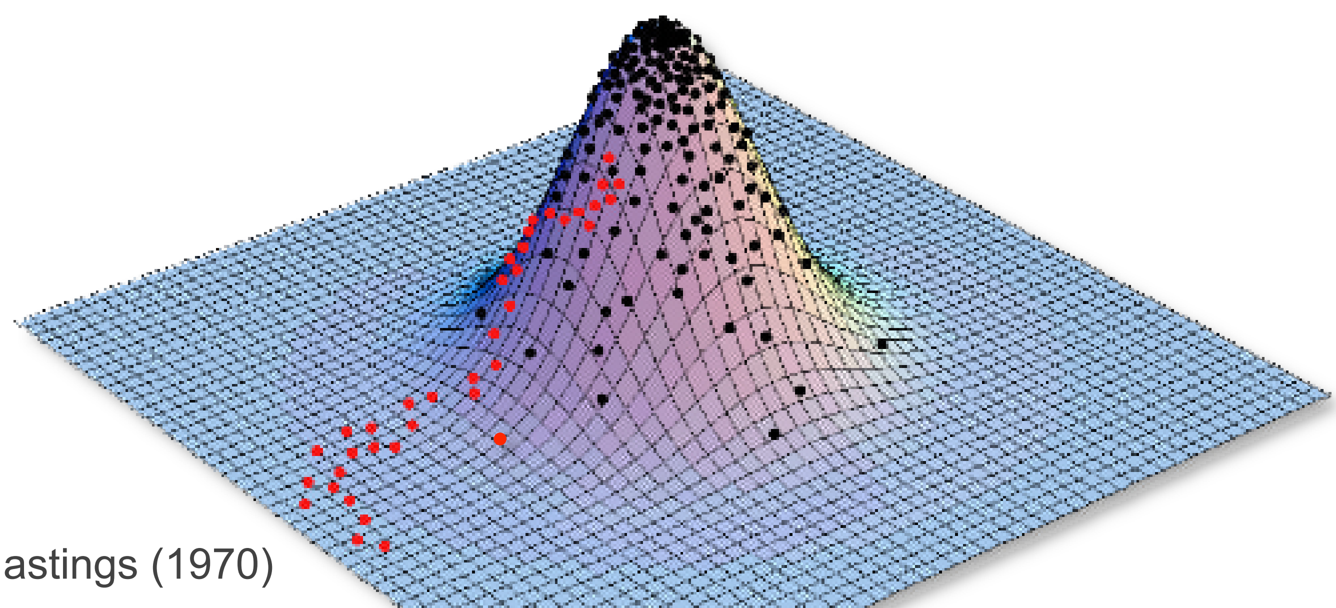
Metropolis et al. (1953); Hastings (1970)

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, $\mathrm{Uniform}[0,1]$
3. The proposal distribution is symmetrical, so $\Pr[A \rightarrow B] = \Pr[B \rightarrow A]$



Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.

2. Select a parameter to update (alter) according to the proposal probabilities

parameter

prior distribution

proposal weights

```
pi ~ dnDirichlet(pi_prior)
#moves for base frequencies
moves[++mi] =  mvSimplexElementScale(pi, alpha=10.0, tune=true, weight=1.0)

er ~ dnDirichlet(er_prior)
#moves for exchangeability rates
moves[++mi] =  mvSimplexElementScale(er, alpha=10.0, tune=true, weight=1.0)

alpha ~ dnUnif( alpha_prior_min, alpha_prior_max )
#moves for alpha-shape parameter
moves[++mi]               =  mvScale(alpha, lambda=0.8, tune=true, weight=1.0)
```

```
Running MCMC simulation
The simulator uses 48 different moves in a random move schedule with 48
moves per iteration
```

Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.

2. Select a parameter to update (alter) according to the proposal probabilities

parameter

prior distribution

proposal weights

```
pi ~ dnDirichlet(pi_prior)
#moves for base frequencies
moves[++mi] =  mvSimplexElementScale(pi, alpha=10.0, tune=true, weight=2.0)

er ~ dnDirichlet(er_prior)
#moves for exchangeability rates
moves[++mi] =  mvSimplexElementScale(er, alpha=10.0, tune=true, weight=2.0)

alpha ~ dnUnif( alpha_prior_min, alpha_prior_max )
#moves for alpha-shape parameter
moves[++mi]            =  mvScale(alpha, lambda=0.8, tune=true, weight=2.0)
```

```
Running MCMC simulation
The simulator uses 48 different moves in a random move schedule with 96
moves per iteration
```

Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.

2. Select a parameter to update (alter) according to the proposal probabilities

parameter

prior distribution

proposal weights

```
pi ~ dnDirichlet(pi_prior)
#moves for base frequencies
moves[++mi] =  mvSimplexElementScale(pi, alpha=10.0, tune=true, weight=4.0)

er ~ dnDirichlet(er_prior)
#moves for exchangeability rates
moves[++mi] =  mvSimplexElementScale(er, alpha=10.0, tune=true, weight=4.0)

alpha ~ dnUnif( alpha_prior_min, alpha_prior_max )
#moves for alpha-shape parameter
moves[++mi]               =  mvScale(alpha, lambda=0.8, tune=true, weight=4.0)
```

```
Running MCMC simulation
The simulator uses 48 different moves in a random move schedule with 192
moves per iteration
```

Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.

2. Select a parameter to update (alter) according to the proposal probabilities

parameter

prior distribution

proposal weights

```
pi ~ dnDirichlet(pi_prior)
#moves for base frequencies
moves[++mi] =  mvSimplexElementScale(pi, alpha=10.0, tune=true, weight=2.0)

er ~ dnDirichlet(er_prior)
#moves for exchangeability rates
moves[++mi] =  mvSimplexElementScale(er, alpha=10.0, tune=true, weight=4.0)

alpha ~ dnUnif( alpha_prior_min, alpha_prior_max )
#moves for alpha-shape parameter
moves[++mi]            =  mvScale(alpha, lambda=0.8, tune=true, weight=8.0)
```

```
Running MCMC simulation
The simulator uses 48 different moves in a random move schedule with 192
moves per iteration
```

Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value for the selected parameter via the proposal mechanism:
   - each parameter has a prior probability distribution of a specific form (uniform, etc.)
   - each prior probability distribution has one or more proposal mechanisms
4. Calculate the probability of accepting the proposed change:

$$R = \min\left[1, \underbrace{\boxed{\frac{f(X \mid \Theta')}{f(X \mid \Theta)}}}_{\text{likelihood ratio}} \cdot \underbrace{\boxed{\frac{f(\Theta')}{f(\Theta)}}}_{\text{prior ratio}} \cdot \underbrace{\boxed{\frac{f(\Theta \mid \Theta')}{f(\Theta' \mid \Theta)}}}_{\text{proposal ratio}}\right]$$

Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value for the selected parameter via the proposal mechanism:
   - each parameter has a prior probability distribution of a specific form (uniform, etc.)
   - each prior probability distribution has one or more proposal mechanisms
4. Calculate the probability of accepting the proposed change:
5. Generate a uniform random variable, $\mathrm{U}[0,1]$, accept if $R > U$
6. Repeat steps 2–5 an ‘adequate’ number of times

Metropolis et al. (1953); Hastings (1970)

Bayesian Inference of Phylogeny (on one slide)

$$f(\text{Parameter} \mid \text{Data}) = \frac{f(\text{Data} \mid \text{Parameter}) f(\text{Parameter})}{f(\text{Data})}$$

I. Data

Assume an alignment, $\mathbf{X}$, of N sites for S species: $\mathbf{X} = (x_1, x_2, x_3, \ldots, x_N)$

II. Phylogenetic model parameters

1. Tree topology $\tau = (\tau_1, \tau_2, \ldots, \tau_{(2s-5)!!})$ ~Uniform

   branch lengths $\nu = (\nu_1, \nu_2, \ldots, \nu_{(2S-3)})$ ~Dirichlet (1,...,1)

2. Model of character change $\Phi = (\theta, \pi, \alpha, T)$

   relative substitution rates $\theta = (\theta_{AC}, \theta_{AG}, \theta_{AT}, \theta_{CG}, \theta_{CT}, \theta_{GT})$ ~Dirichlet (1,1,1,1,1,1)

   stationary frequencies $\pi = (\pi_A, \pi_C, \pi_G, \pi_T)$ ~Dirichlet (1,1,1,1)

IV. Priors on parameters

III. Phylogenetic likelihood function

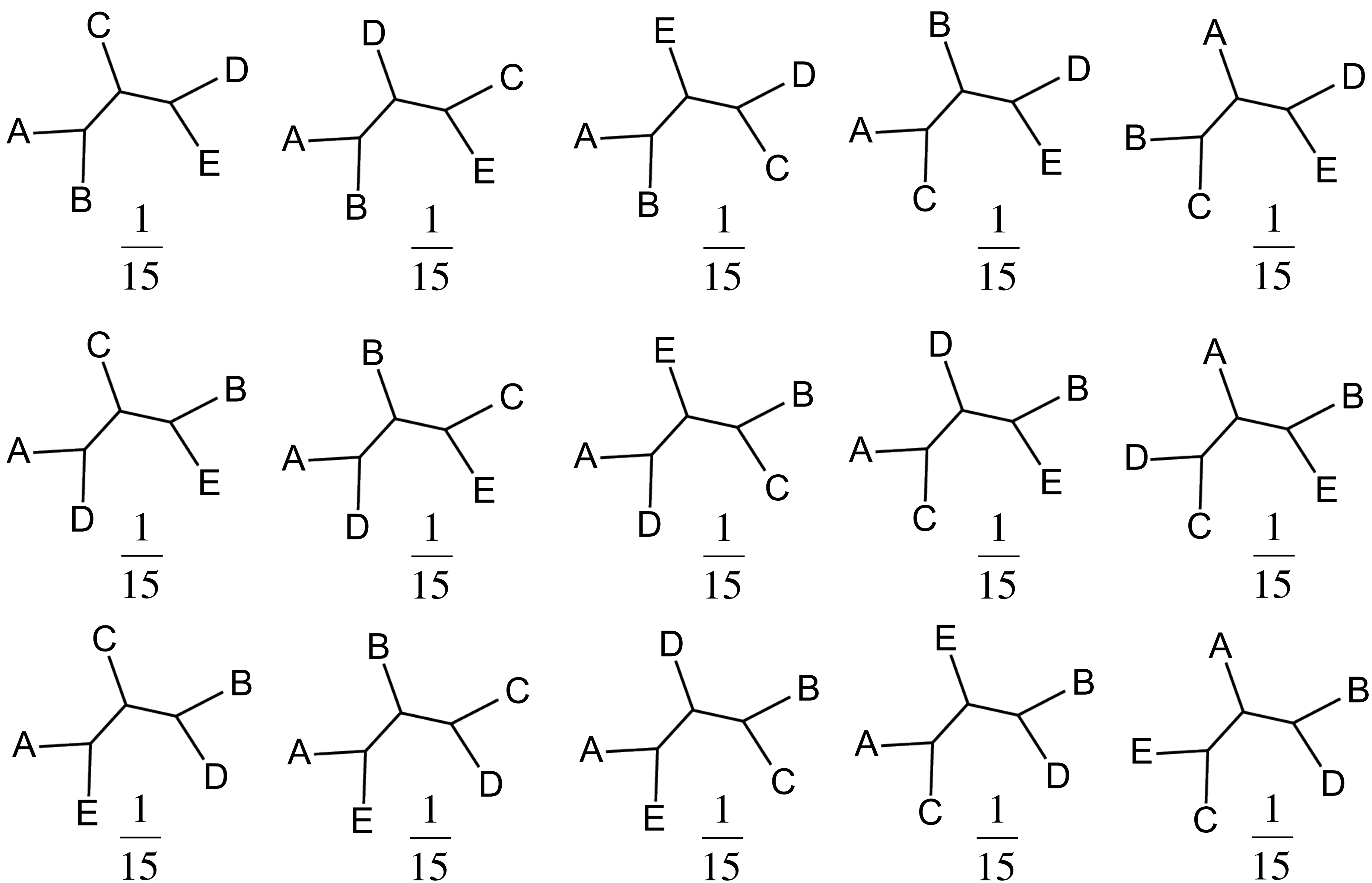
$$L(\tau, \nu, \Theta) \propto f(\mathbf{X} \mid \tau, \nu, \Theta) = \prod_{i=1}^{N} f(x_i \mid \tau, \nu, \Theta)$$

V. Posterior Probability

$$f(\tau, \nu, \Phi \mid \mathbf{X}) = \frac{f(\mathbf{X} \mid \tau, \nu, \Phi) f(\tau, \nu, \Phi)}{f(\mathbf{X})}$$

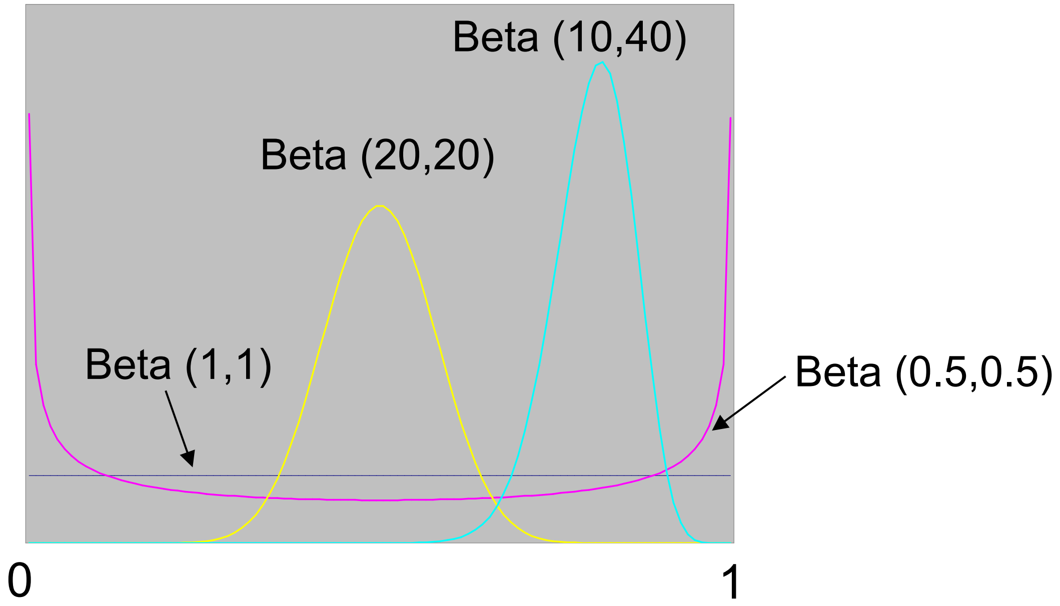
Approximating the Joint Posterior Probability Density using MCMC

Discrete uniform prior on topologies



Approximating the Joint Posterior Probability Density using MCMC

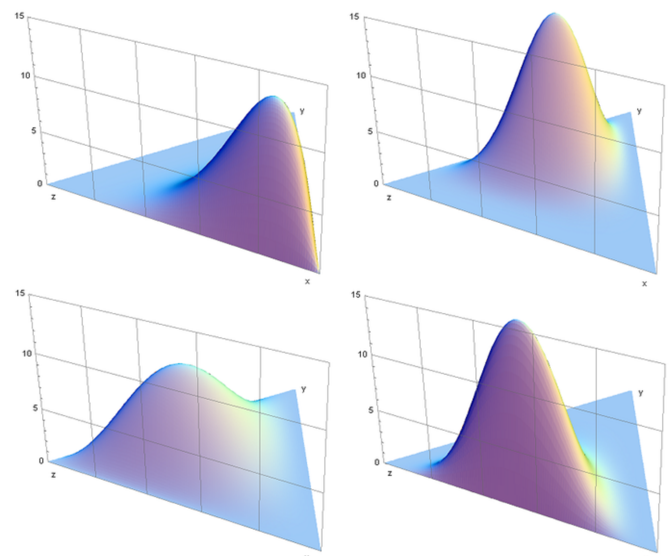
Beta prior

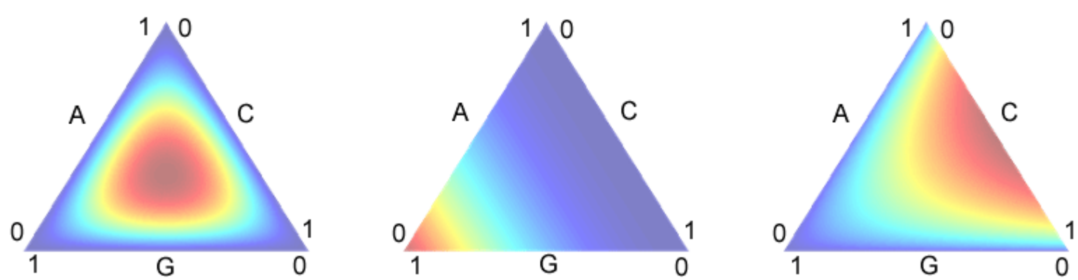


Approximating the Joint Posterior Probability Density using MCMC

Dirichlet prior

Generalization of the beta often used for proportions



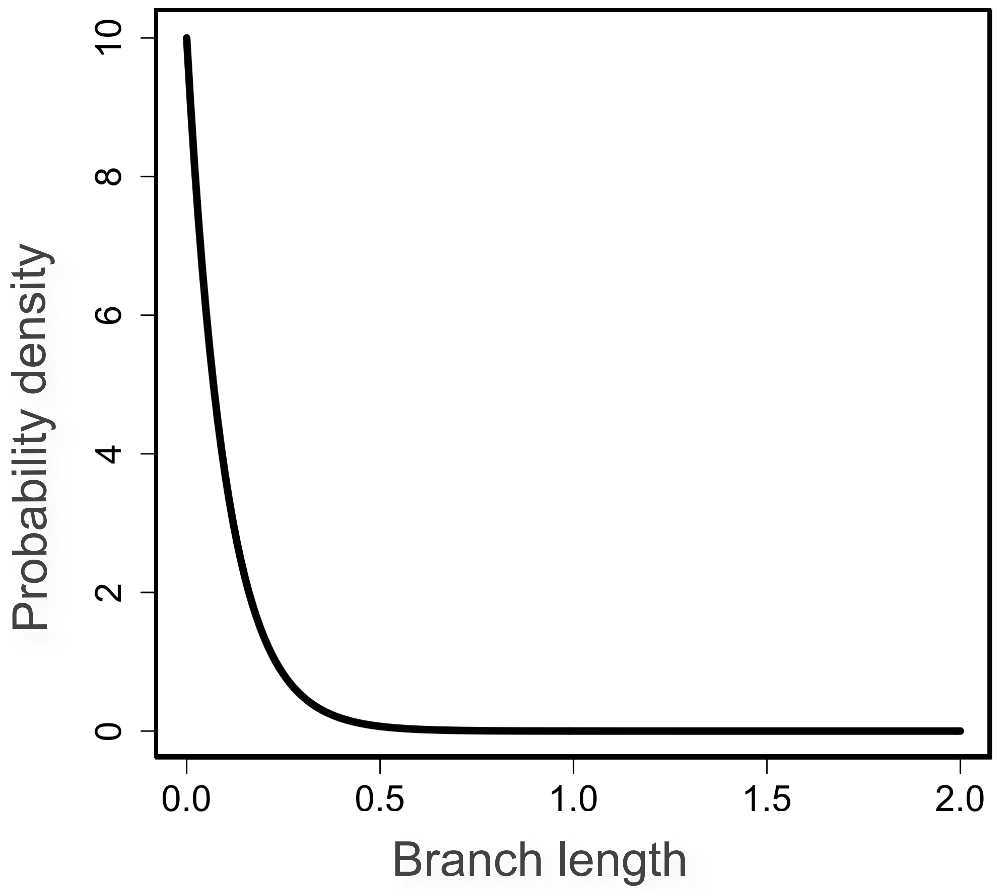


Approximating the Joint Posterior Probability Density using MCMC

Exponential priors

Often used for branch lengths

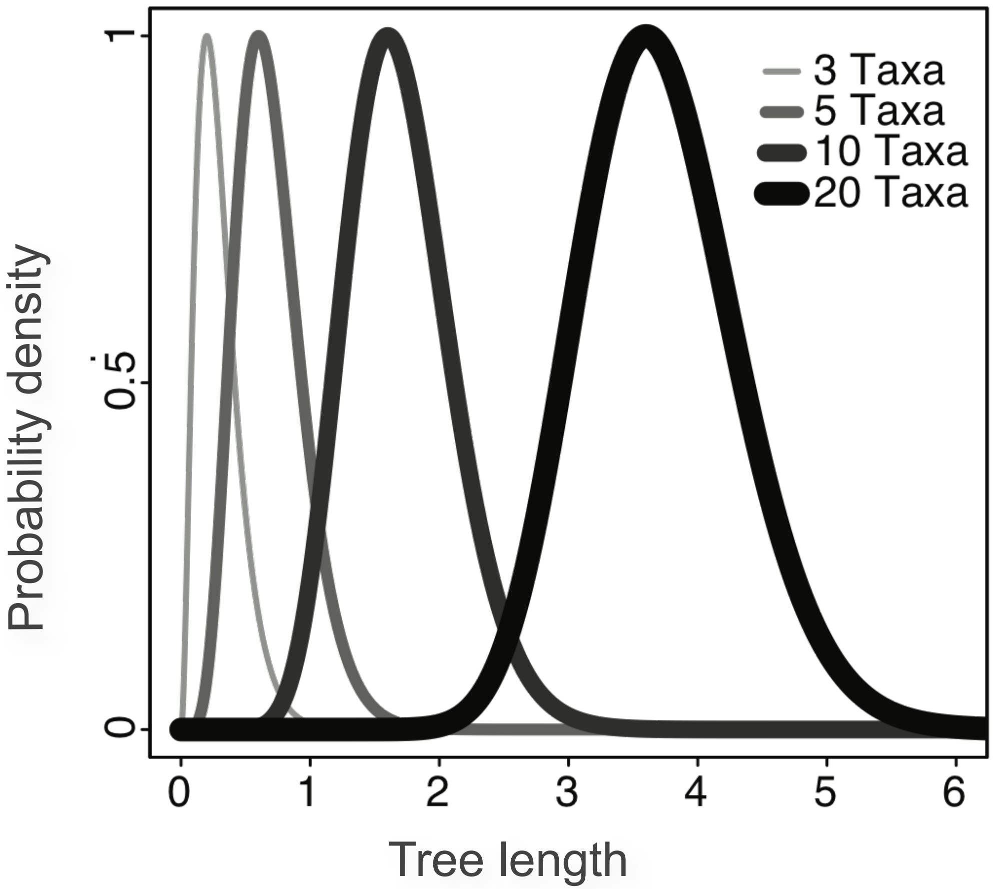
Default Exponential Branch-Length Prior (λ=10, mean=0.1)



Approximating the Joint Posterior Probability Density using MCMC

Gamma prior

May be used for tree length



Approximating the Joint Posterior Probability Density using MCMC

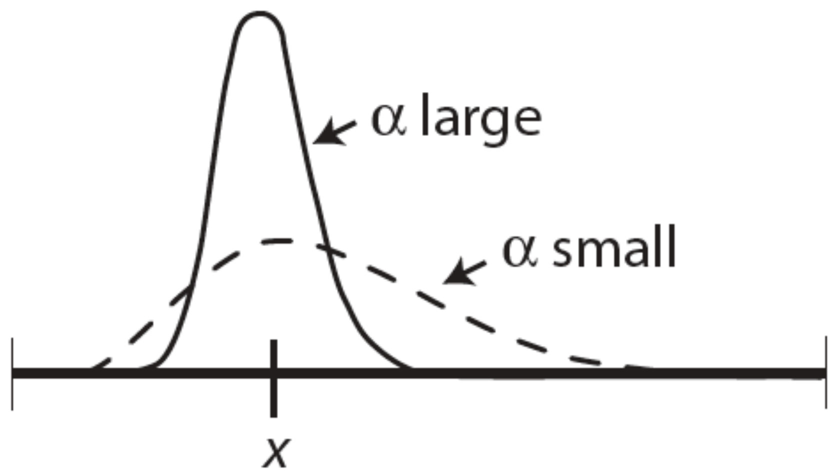
The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\Theta = (\tau, \nu)$.
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value for the selected parameter via the proposal mechanism:
   - each parameter has a prior probability distribution of a specific form (uniform, etc.)
   - each prior probability distribution has one or more proposal mechanisms
4. Calculate the probability of accepting the proposed change:
5. Generate a uniform random variable, $\mathrm{U}[0,1]$, accept if $R > U$
6. Repeat steps 2–5 an ‘adequate’ number of times

Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using MCMC

Dirichlet proposal mechanism



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

Tuning parameter: α

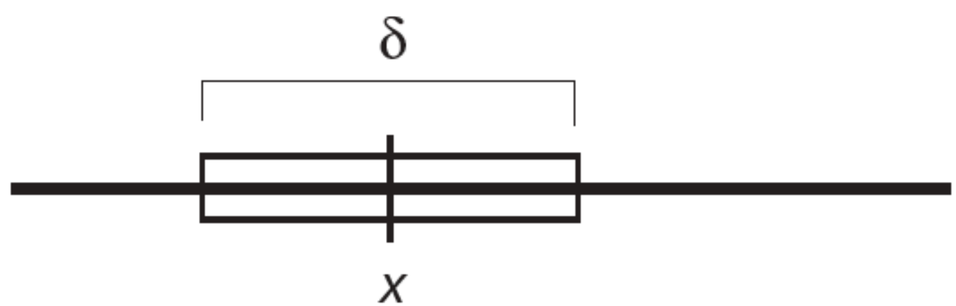
Bolder proposals: decrease α

More modest proposals: increase α

```
pi ~ dnDirichlet(pi_prior)
moves[++mi] = mvSimplexElementScale(pi, alpha=10.0, tune=true, weight=2.0)
```

Approximating the Joint Posterior Probability Density using MCMC

Sliding-window proposal mechanism

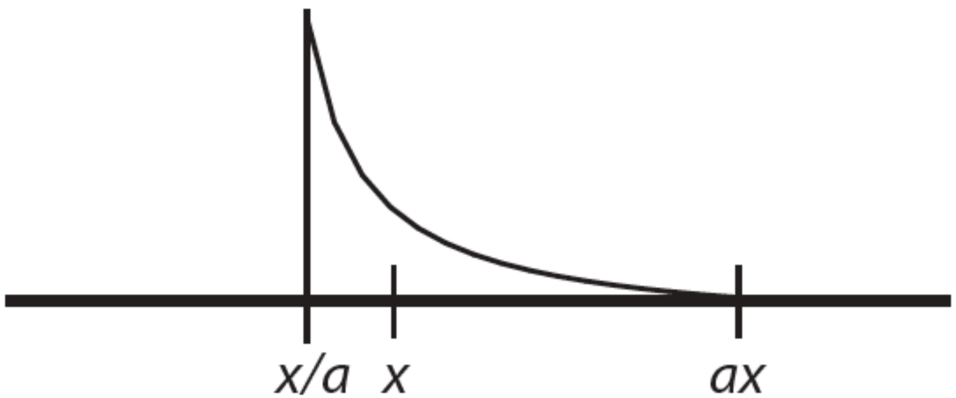


New values are picked uniformly from a sliding window of size δ centered on x.
Tuning parameter: δ
Bolder proposals: increase δ
More modest proposals: decrease δ

```
epsilon ~ dnUnif( epsilon_prior_min, epsilon_prior_max )
moves[++mi] = mvSlide(epsilon, delta=0.8, tune=true, weight=3.0)
```

Approximating the Joint Posterior Probability Density using MCMC

Multiplier proposal mechanism

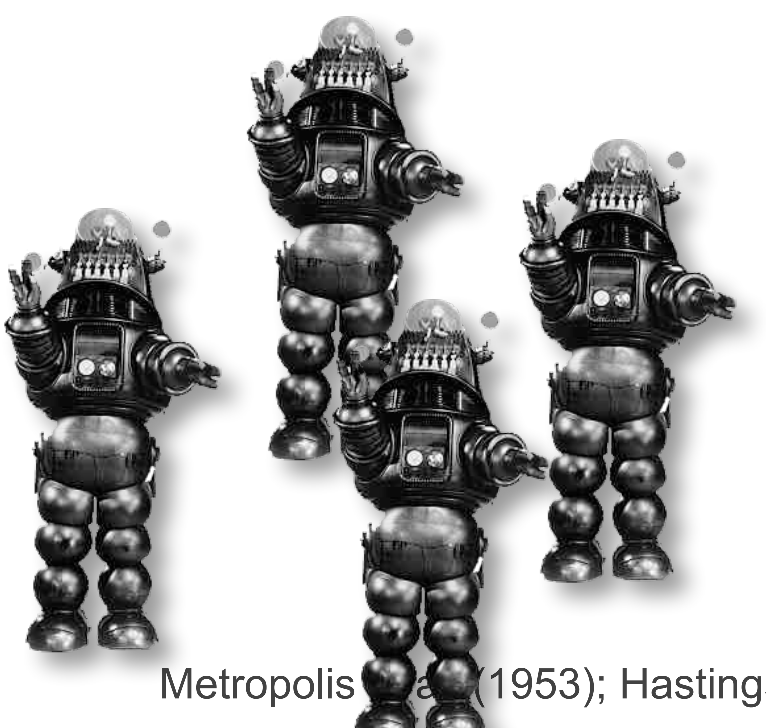


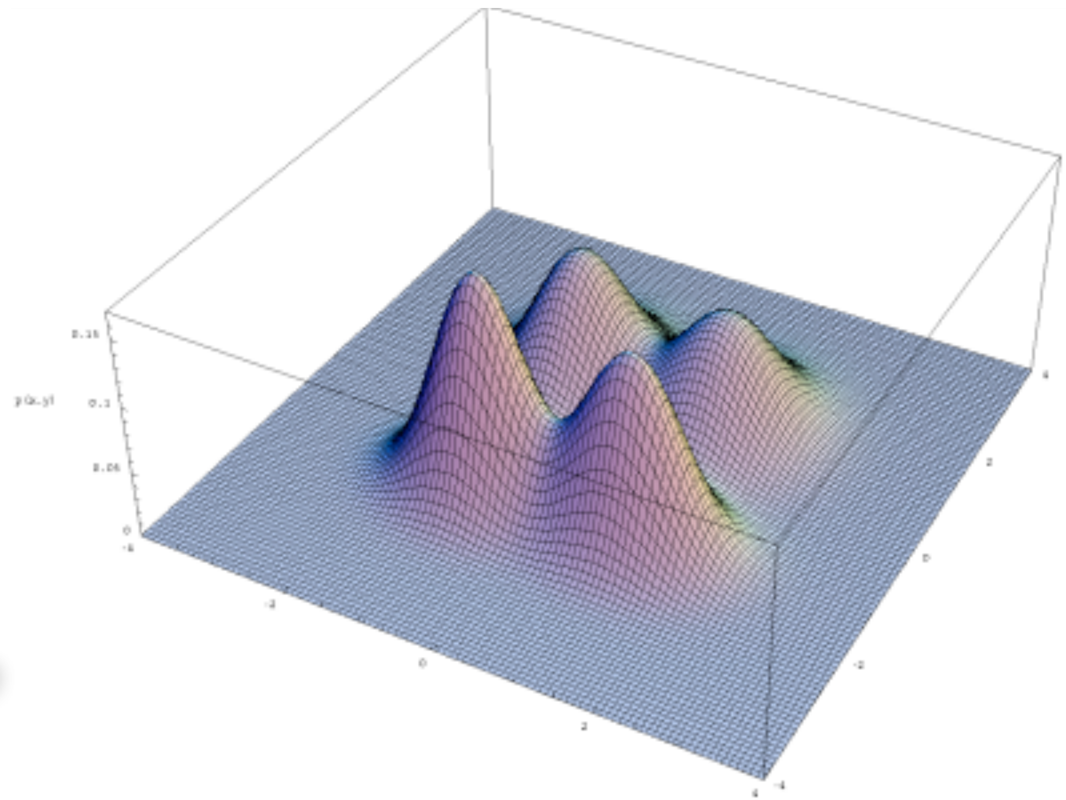
New values are picked from the equivalent of a sliding window on the log-transformed x axis.
Tuning parameter: $\lambda = 2 \ln a$
Bolder proposals: increase λ
More modest proposals: decrease λ

```
br_lens[i] ~ dnExponential(10.0)
moves[++mi] = mvScale(br_lens[i],lambda=1,tune=true,weight=1)
```

Approximating the Joint Posterior Probability Density using MCMC

Robot Squadron!!





Metropolis et al. (1953); Hastings (1970)

Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

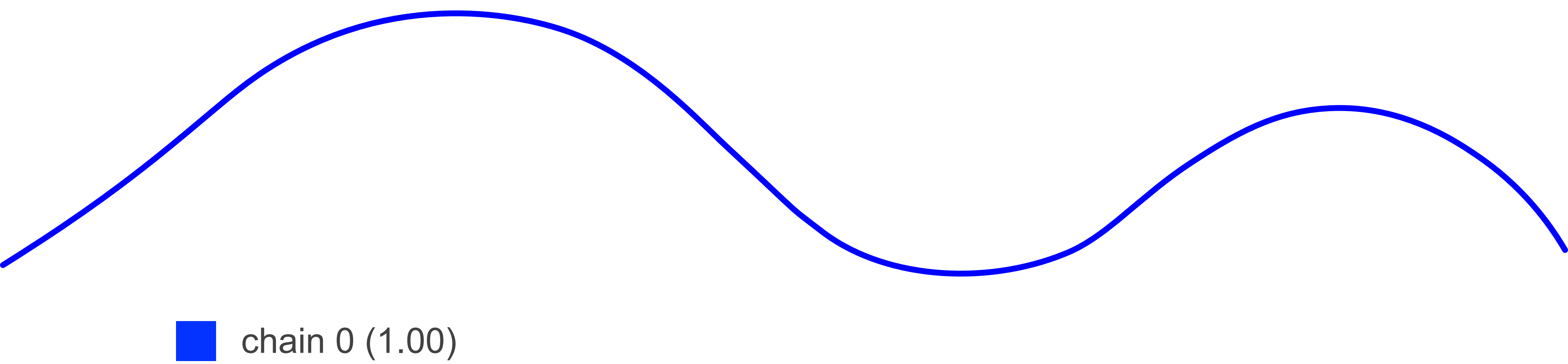
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
   - posterior of chain i is raised to a power: the heat of chain $i = 1/(1 + iT)$
   - the incremental heating 'flattens' the posterior, allowing chains to more readily traverse regions of low probability

	temperature			
chain	0.25	0.20	0.15	0.10
0	1.00	1.00	1.00	1.00
1	0.80	0.83	0.87	0.91
2	0.67	0.71	0.77	0.83
3	0.57	0.63	0.69	0.77

Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

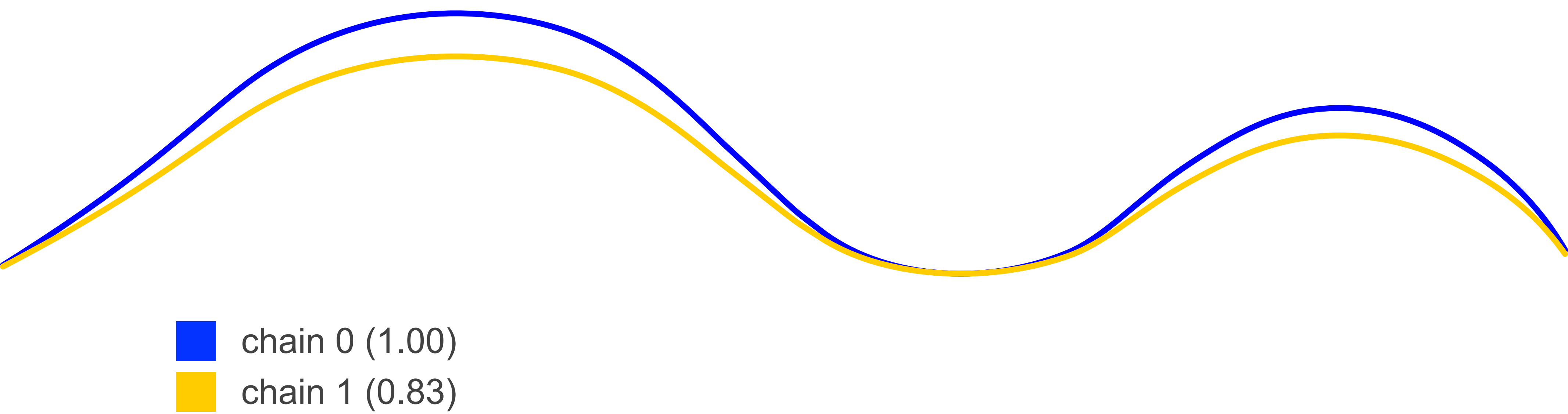
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
   - posterior of chain i is raised to a power: the heat of chain $i = 1/(1 + iT)$
   - the incremental heating 'flattens' the posterior, allowing chains to more readily traverse regions of low probability



Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

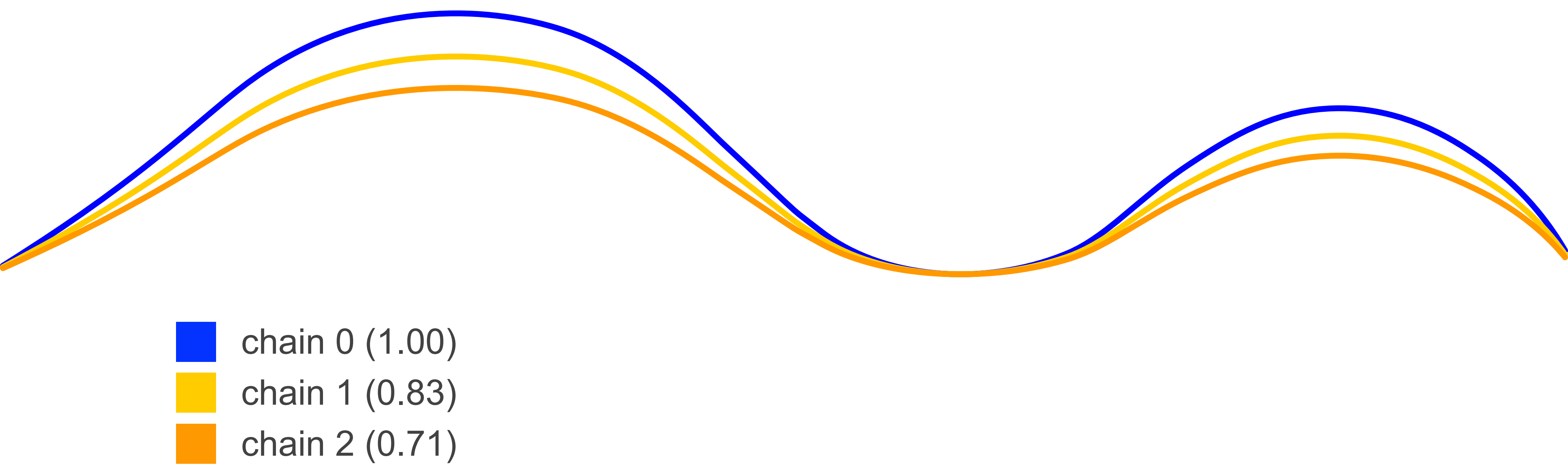
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
   - posterior of chain i is raised to a power: the heat of chain $i = 1/(1 + iT)$
   - the incremental heating 'flattens' the posterior, allowing chains to more readily traverse regions of low probability



Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

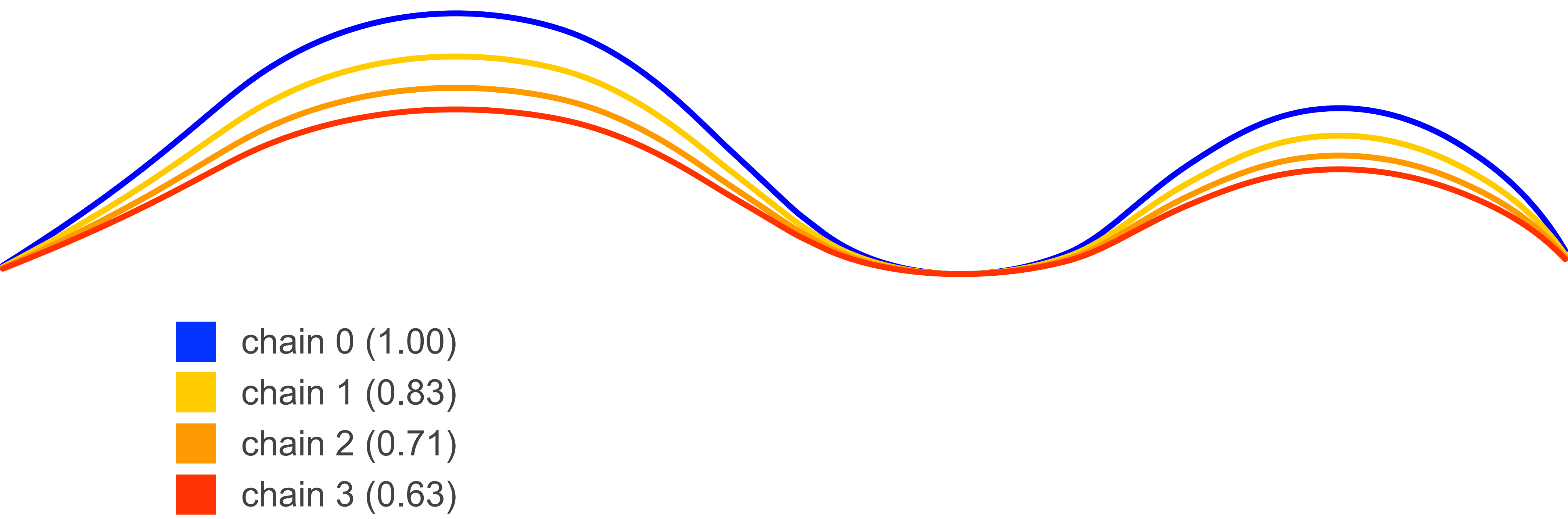
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
    - posterior of chain i is raised to a power: the heat of chain $i = 1/(1 + iT)$
    - the incremental heating 'flattens' the posterior, allowing chains to more readily traverse regions of low probability



Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

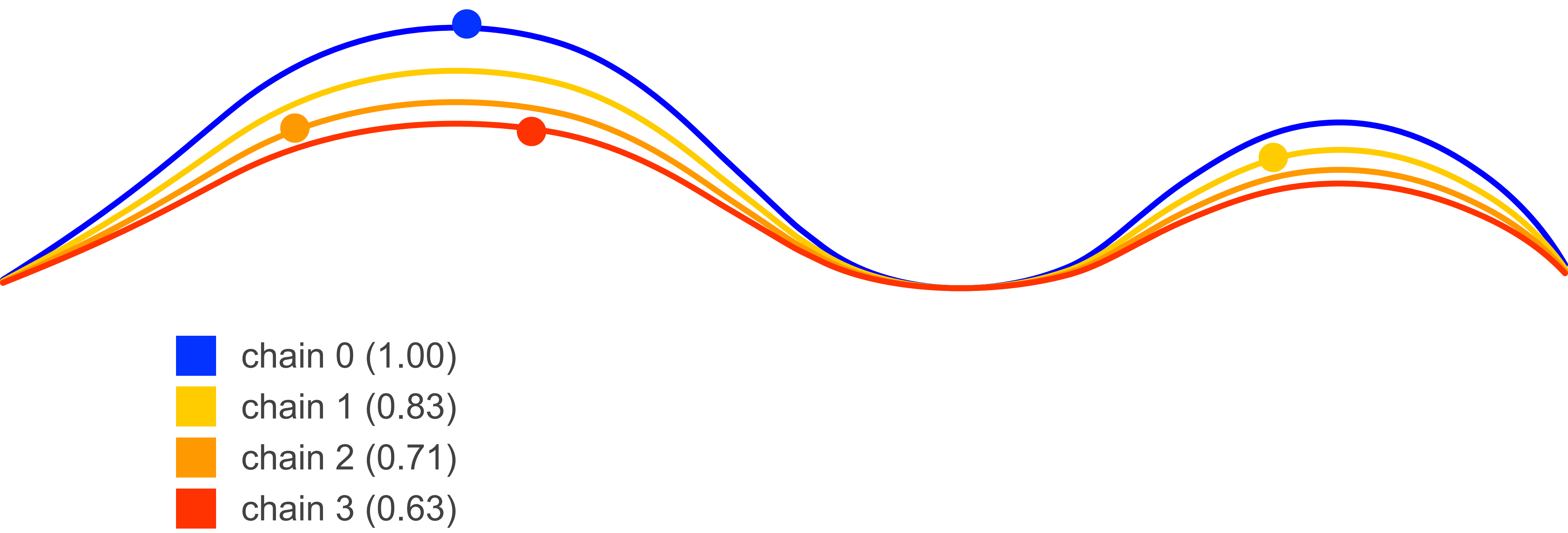
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
   - posterior of chain i is raised to a power: the heat of chain $i = 1/(1 + iT)$
   - the incremental heating 'flattens' the posterior, allowing chains to more readily traverse regions of low probability



Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC³ algorithm

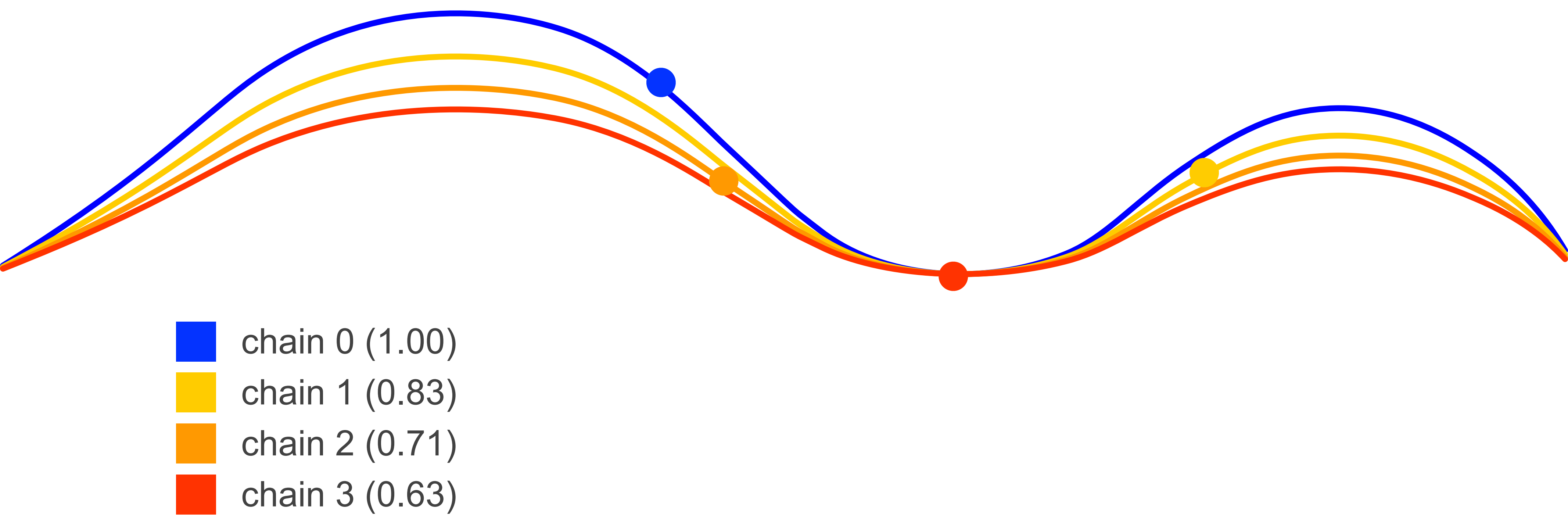
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
   - posterior of chain i is raised to a power: the heat of chain $i = 1/(1 + iT)$
   - the incremental heating 'flattens' the posterior, allowing chains to more readily traverse regions of low probability



Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

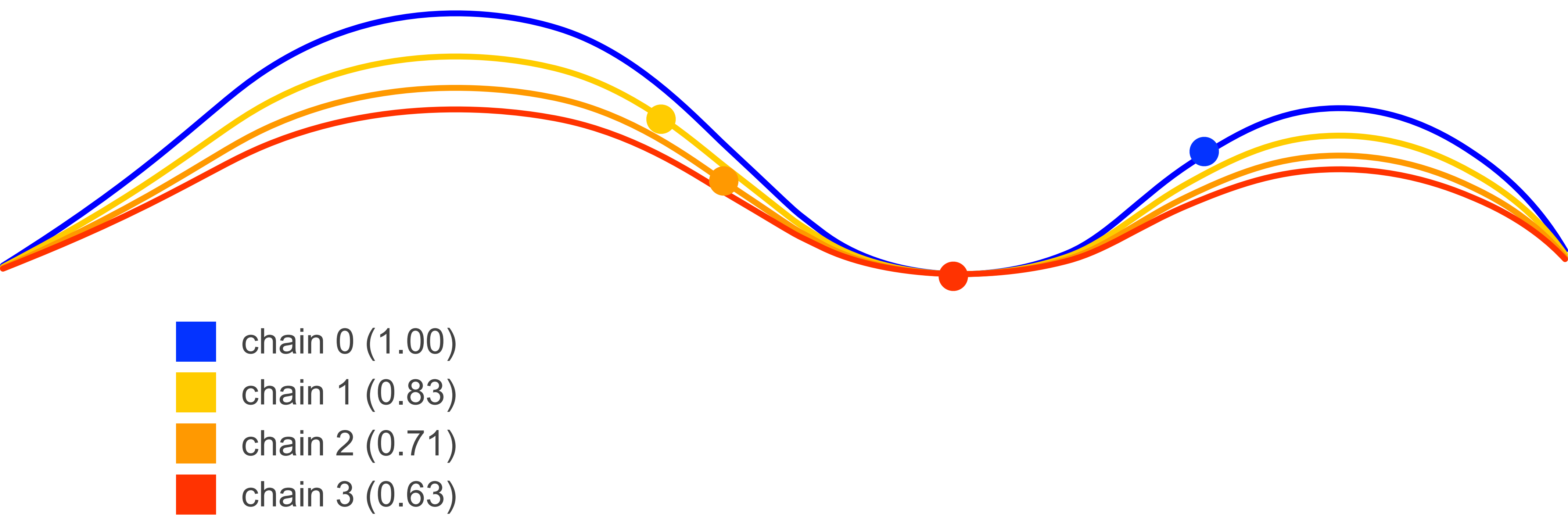
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
3. At prescribed intervals, two chains are randomly selected to swap.
   - we compute the acceptance probability of the proposed swap
   - if accepted, the chains swap positions (and in computer memory)



Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC^3 algorithm

1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold.
3. At prescribed intervals, two chains are randomly selected to swap.
4. Only samples from the cold chain are used to approximate the posterior.



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

The frequency of sampled parameter values provides a valid estimate of the posterior probability of that parameter

- *e.g.*, the frequency of a sampled clade provides an estimate of its nodal probability

We can query the joint posterior with respect to any individual parameter of interest: the marginal posterior probability

Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

Each sample includes the values of all model parameters.

[ID: 2325481386]

Gen	LnL	TL	r(A<->C)	r(A<->G)	r(A<->T)	r(C<->G)	r(C<->T)	r(G<->T)	pi(A)	pi(C)	pi(G)	pi(T)	alpha
1	-13413.769	1.313	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667	0.250000	0.250000	0.250000	0.250000	0.500000
1000	-10429.772	0.904	0.100364	0.271178	0.057126	0.095681	0.404818	0.070833	0.276201	0.173231	0.228359	0.322209	0.845634
2000	-10420.654	0.980	0.115937	0.254216	0.041309	0.051039	0.455344	0.082157	0.291050	0.181003	0.231042	0.296904	0.670406
3000	-10417.930	0.961	0.137253	0.264348	0.037891	0.056962	0.426295	0.077251	0.291050	0.181003	0.231042	0.296904	0.901480
4000	-10423.816	0.925	0.101065	0.273786	0.035266	0.067623	0.441301	0.080958	0.290603	0.185952	0.231800	0.291644	0.859284
5000	-10425.264	1.002	0.135985	0.259584	0.048509	0.057733	0.430436	0.067753	0.289106	0.189615	0.210373	0.310906	0.671675
6000	-10421.366	0.962	0.119016	0.268203	0.041284	0.062913	0.415543	0.093041	0.281133	0.187367	0.234148	0.297353	0.824395
7000	-10417.840	0.981	0.123308	0.246185	0.032588	0.070686	0.443381	0.083851	0.298478	0.186125	0.221560	0.293837	0.644508
8000	-10420.174	1.058	0.129152	0.263612	0.036846	0.061359	0.424323	0.084708	0.284539	0.192084	0.216456	0.306921	0.691606
9000	-10419.701	0.980	0.101173	0.266573	0.035445	0.072158	0.438826	0.085825	0.285541	0.188378	0.229610	0.296471	0.687021
10000	-10423.917	1.015	0.100312	0.289851	0.045985	0.059364	0.422372	0.082115	0.285505	0.176257	0.228230	0.310007	0.684473
11000	-10418.487	0.945	0.107911	0.270677	0.049322	0.063833	0.421602	0.086655	0.279829	0.188085	0.233921	0.298165	0.860128
12000	-10420.169	0.893	0.115085	0.270950	0.038203	0.070506	0.417478	0.087778	0.288131	0.191473	0.231758	0.288638	0.723312
13000	-10419.081	0.922	0.115323	0.269076	0.036184	0.069919	0.429555	0.079943	0.294340	0.187665	0.227043	0.290952	0.784700
14000	-10423.817	1.030	0.112545	0.254842	0.042601	0.077867	0.436797	0.075348	0.283706	0.189549	0.224014	0.302731	0.615981
15000	-10424.879	0.944	0.131641	0.260134	0.043160	0.069779	0.421550	0.073736	0.296187	0.175620	0.219147	0.309046	0.797970
16000	-10426.143	0.940	0.117469	0.266011	0.056463	0.049593	0.441326	0.069139	0.282578	0.203117	0.231372	0.282933	0.792757
17000	-10421.133	0.978	0.134024	0.277374	0.040419	0.056384	0.416233	0.075565	0.289061	0.187968	0.225825	0.297145	0.767063
18000	-10418.290	0.930	0.104450	0.251683	0.041434	0.063649	0.455528	0.083256	0.287086	0.189510	0.226700	0.296704	0.767072
19000	-10420.052	0.972	0.121227	0.274901	0.037023	0.083743	0.414224	0.068881	0.289061	0.187968	0.225825	0.297145	0.758345
20000	-10425.127	0.955	0.099741	0.277386	0.043745	0.069447	0.433059	0.076622	0.292229	0.197483	0.212827	0.297461	0.645034
21000	-10421.087	0.939	0.105737	0.258514	0.039941	0.094773	0.429045	0.071991	0.292778	0.192129	0.217655	0.297438	0.692877
22000	-10421.805	0.926	0.111237	0.293260	0.047595	0.061320	0.409044	0.077544	0.286897	0.197795	0.222410	0.292899	0.797696
23000	-10422.326	0.943	0.123590	0.240213	0.047236	0.048864	0.453312	0.086786	0.291024	0.187438	0.225934	0.295603	0.851381
24000	-10417.974	0.938	0.123674	0.274369	0.051414	0.065387	0.413009	0.072146	0.291024	0.187438	0.225934	0.295603	0.801620
25000	-10422.454	0.996	0.132415	0.249036	0.036744	0.063052	0.457012	0.061741	0.299053	0.171847	0.226435	0.302665	0.607659
26000	-10424.506	0.892	0.122118	0.235061	0.042240	0.063788	0.462004	0.074790	0.302331	0.170502	0.220011	0.307156	0.812245
27000	-10420.001	0.953	0.128264	0.263415	0.040470	0.058989	0.432138	0.076724	0.279181	0.190422	0.234369	0.296028	0.824956

Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

Each sample includes the values of all model parameters.

[ID: 2325481386]

Gen	LnL	TL	r(A<->C)	r(A<->G)	r(A<->T)	r(C<->G)	r(C<->T)	r(G<->T)	pi(A)	pi(C)	pi(G)	pi(T)	alpha
1	-13413.769	1.313	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667	0.250000	0.250000	0.250000	0.250000	0.500000
1000	-10429.772	0.904	0.100364	0.271178	0.057126	0.095681	0.404818	0.070833	0.276201	0.173231	0.228359	0.322209	0.845634
2000	-10420.654	0.980	0.115937	0.254216	0.041309	0.051039	0.455344	0.082157	0.291050	0.181003	0.231042	0.296904	0.670406
3000	-10417.930	0.961	0.137253	0.264348	0.037891	0.056962	0.426295	0.077251	0.291050	0.181003	0.231042	0.296904	0.901480
4000	-10423.816	0.925	0.101065	0.273786	0.035266	0.067623	0.441301	0.080958	0.290603	0.185952	0.231800	0.291644	0.859284
5000	-10425.264	1.002	0.135985	0.259584	0.048509	0.057733	0.430436	0.067753	0.289106	0.189615	0.210373	0.310906	0.671675
6000	-10421.366	0.962	0.119016	0.268203	0.041284	0.062913	0.415543	0.093041	0.281133	0.187367	0.234148	0.297353	0.824395
7000	-10417.840	0.981	0.123308	0.246185	0.032588	0.070686	0.443381	0.083851	0.298478	0.186125	0.221560	0.293837	0.644508
8000	-10420.174	1.058	0.129152	0.263612	0.036846	0.061359	0.424323	0.084708	0.284539	0.192084	0.216456	0.306921	0.691606
9000	-10419.701	0.980	0.101173	0.266573	0.035445	0.072158	0.438826	0.085825	0.285541	0.188378	0.229610	0.296471	0.687021
10000	-10423.917	1.015	0.100312	0.289851	0.045985	0.059364	0.422372	0.082115	0.285505	0.176257	0.228230	0.310007	0.684473
11000	-10418.487	0.945	0.107911	0.270677	0.049322	0.063833	0.421602	0.086655	0.279829	0.188085	0.233921	0.298165	0.860128
12000	-10420.169	0.893	0.115085	0.270950	0.038203	0.070506	0.417478	0.087778	0.288131	0.191473	0.231758	0.288638	0.723312
13000	-10419.081	0.922	0.115323	0.269076	0.036184	0.069919	0.429555	0.079943	0.294340	0.187665	0.227043	0.290952	0.784700
14000	-10423.817	1.030	0.112545	0.254842	0.042601	0.077867	0.436797	0.075348	0.283706	0.189549	0.224014	0.302731	0.615981
15000	-10424.879	0.944	0.131641	0.260134	0.043160	0.069779	0.421550	0.073736	0.296187	0.175620	0.219147	0.309046	0.797970
16000	-10426.143	0.940	0.117469	0.266011	0.056463	0.049593	0.441326	0.069139	0.282578	0.203117	0.231372	0.282933	0.792757
17000	-10421.133	0.978	0.134024	0.277374	0.040419	0.056384	0.416233	0.075565	0.289061	0.187968	0.225825	0.297145	0.767063
18000	-10418.290	0.930	0.104450	0.251683	0.041434	0.063649	0.455528	0.083256	0.287086	0.189510	0.226700	0.296704	0.767072
19000	-10420.052	0.972	0.121227	0.274901	0.037023	0.083743	0.414224	0.068881	0.289061	0.187968	0.225825	0.297145	0.758345
20000	-10425.127	0.955	0.099741	0.277386	0.043745	0.069447	0.433059	0.076622	0.292229	0.197483	0.212827	0.297461	0.645034
21000	-10421.087	0.939	0.105737	0.258514	0.039941	0.094773	0.429045	0.071991	0.292778	0.192129	0.217655	0.297438	0.692877
22000	-10421.805	0.926	0.111237	0.293260	0.047595	0.061320	0.409044	0.077544	0.286897	0.197795	0.222410	0.292899	0.797696
23000	-10422.326	0.943	0.123590	0.240213	0.047236	0.048864	0.453312	0.086786	0.291024	0.187438	0.225934	0.295603	0.851381
24000	-10417.974	0.938	0.123674	0.274369	0.051414	0.065387	0.413009	0.072146	0.291024	0.187438	0.225934	0.295603	0.801620
25000	-10422.454	0.996	0.132415	0.249036	0.036744	0.063052	0.457012	0.061741	0.299053	0.171847	0.226435	0.302665	0.607659
26000	-10424.506	0.892	0.122118	0.235061	0.042240	0.063788	0.462004	0.074790	0.302331	0.170502	0.220011	0.307156	0.812245
27000	-10420.001	0.953	0.128264	0.263415	0.040470	0.058989	0.432138	0.076724	0.279181	0.190422	0.234369	0.296028	0.824956

Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

We can query the joint distribution marginally with respect to any parameter.

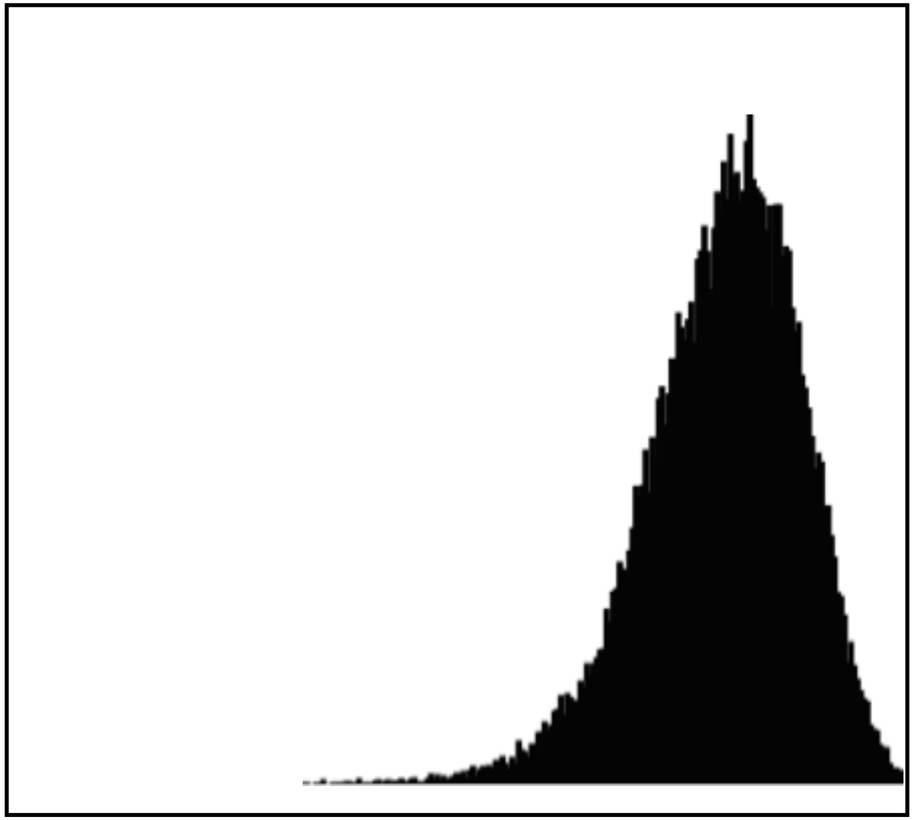
[ID: 2325481386]

Gen	LnL	TL	r(A<->C)	r(A<->G)	r(A<->T)	r(C<->G)	r(C<->T)	r(G<->T)	pi(A)	pi(C)	pi(G)	pi(T)	alpha
1	-13413.769	1.313	0.166667	0.166667	0.166667	0.166667	0.166667	0.166667	0.250000	0.250000	0.250000	0.250000	0.500000
1000	-10429.772	0.904	0.100364	0.271178	0.057126	0.095681	0.404818	0.070833	0.276201	0.173231	0.228359	0.322209	0.845634
2000	-10420.654	0.980	0.115937	0.254216	0.041309	0.051039	0.455344	0.082157	0.291050	0.181003	0.231042	0.296904	0.670406
3000	-10417.930	0.961	0.137253	0.264348	0.037891	0.056962	0.426295	0.077251	0.291050	0.181003	0.231042	0.296904	0.901480
4000	-10423.816	0.925	0.101065	0.273786	0.035266	0.067623	0.441301	0.080958	0.290603	0.185952	0.231800	0.291644	0.859284
5000	-10425.264	1.002	0.135985	0.259584	0.048509	0.057733	0.430436	0.067753	0.289106	0.189615	0.210373	0.310906	0.671675
6000	-10421.366	0.962	0.119016	0.268203	0.041284	0.062913	0.415543	0.093041	0.281133	0.187367	0.234148	0.297353	0.824395
7000	-10417.840	0.981	0.123308	0.246185	0.032588	0.070686	0.443381	0.083851	0.298478	0.186125	0.221560	0.293837	0.644508
8000	-10420.174	1.058	0.129152	0.263612	0.036846	0.061359	0.424323	0.084708	0.284539	0.192084	0.216456	0.306921	0.691606
9000	-10419.701	0.980	0.101173	0.266573	0.035445	0.072158	0.438826	0.085825	0.285541	0.188378	0.229610	0.296471	0.687021
10000	-10423.917	1.015	0.100312	0.289851	0.045985	0.059364	0.422372	0.082115	0.285505	0.176257	0.228230	0.310007	0.684473
11000	-10418.487	0.945	0.107911	0.270677	0.049322	0.063833	0.421602	0.086655	0.279829	0.188085	0.233921	0.298165	0.860128
12000	-10420.169	0.893	0.115085	0.270950	0.038203	0.070506	0.417478	0.087778	0.288131	0.191473	0.231758	0.288638	0.723312
13000	-10419.081	0.922	0.115323	0.269076	0.036184	0.069919	0.429555	0.079943	0.294340	0.187665	0.227043	0.290952	0.784700
14000	-10423.817	1.030	0.112545	0.254842	0.042601	0.077867	0.436797	0.075348	0.283706	0.189549	0.224014	0.302731	0.615981
15000	-10424.879	0.944	0.131641	0.260134	0.043160	0.069779	0.421550	0.073736	0.296187	0.175620	0.219147	0.309046	0.797970
16000	-10426.143	0.940	0.117469	0.266011	0.056463	0.049593	0.441326	0.069139	0.282578	0.203117	0.231372	0.282933	0.792757
17000	-10421.133	0.978	0.134024	0.277374	0.040419	0.056384	0.416233	0.075565	0.289061	0.187968	0.225825	0.297145	0.767063
18000	-10418.290	0.930	0.104450	0.251683	0.041434	0.063649	0.455528	0.083256	0.287086	0.189510	0.226700	0.296704	0.767072
19000	-10420.052	0.972	0.121227	0.274901	0.037023	0.083743	0.414224	0.068881	0.289061	0.187968	0.225825	0.297145	0.758345
20000	-10425.127	0.955	0.099741	0.277386	0.043745	0.069447	0.433059	0.076622	0.292229	0.197483	0.212827	0.297461	0.645034
21000	-10421.087	0.939	0.105737	0.258514	0.039941	0.094773	0.429045	0.071991	0.292778	0.192129	0.217655	0.297438	0.692877
22000	-10421.805	0.926	0.111237	0.293260	0.047595	0.061320	0.409044	0.077544	0.286897	0.197795	0.222410	0.292899	0.797696
23000	-10422.326	0.943	0.123590	0.240213	0.047236	0.048864	0.453312	0.086786	0.291024	0.187438	0.225934	0.295603	0.851381
24000	-10417.974	0.938	0.123674	0.274369	0.051414	0.065387	0.413009	0.072146	0.291024	0.187438	0.225934	0.295603	0.801620
25000	-10422.454	0.996	0.132415	0.249036	0.036744	0.063052	0.457012	0.061741	0.299053	0.171847	0.226435	0.302665	0.607659
26000	-10424.506	0.892	0.122118	0.235061	0.042240	0.063788	0.462004	0.074790	0.302331	0.170502	0.220011	0.307156	0.812245
27000	-10420.001	0.953	0.128264	0.263415	0.040470	0.058989	0.432138	0.076724	0.279181	0.190422	0.234369	0.296028	0.824956

Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

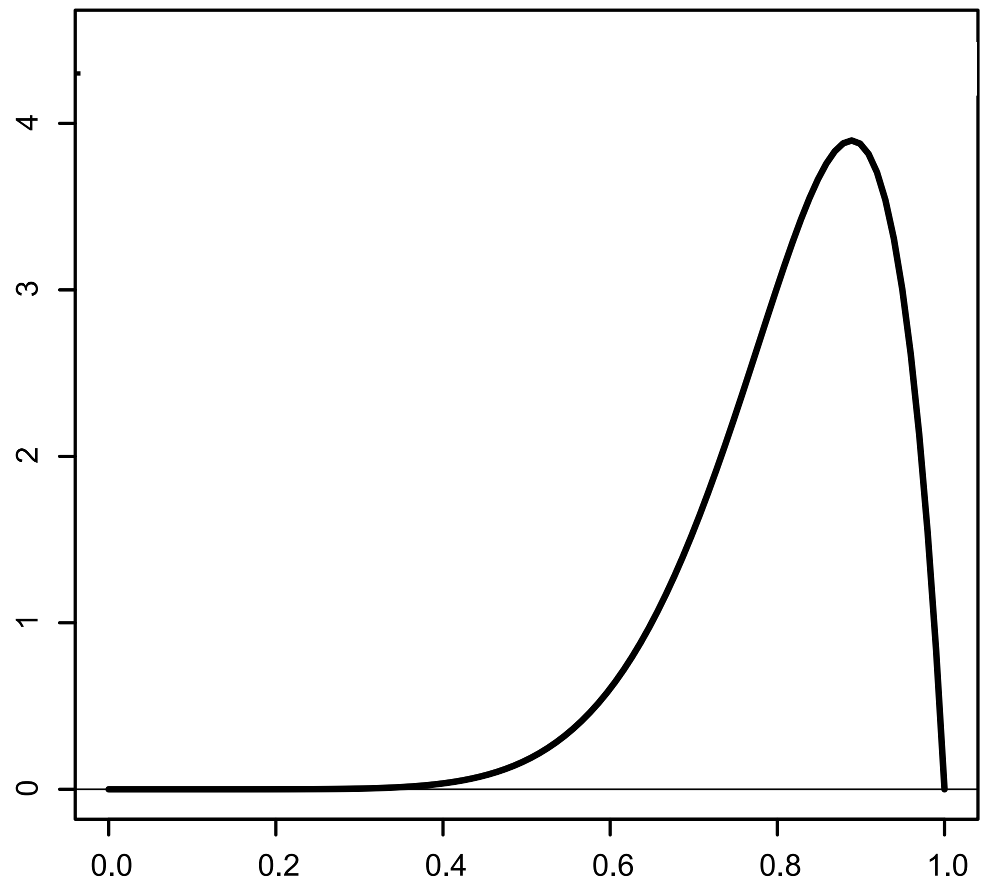
We can do this by simply constructed a histogram for any column in the file this provides an estimate of its marginal posterior probability density



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

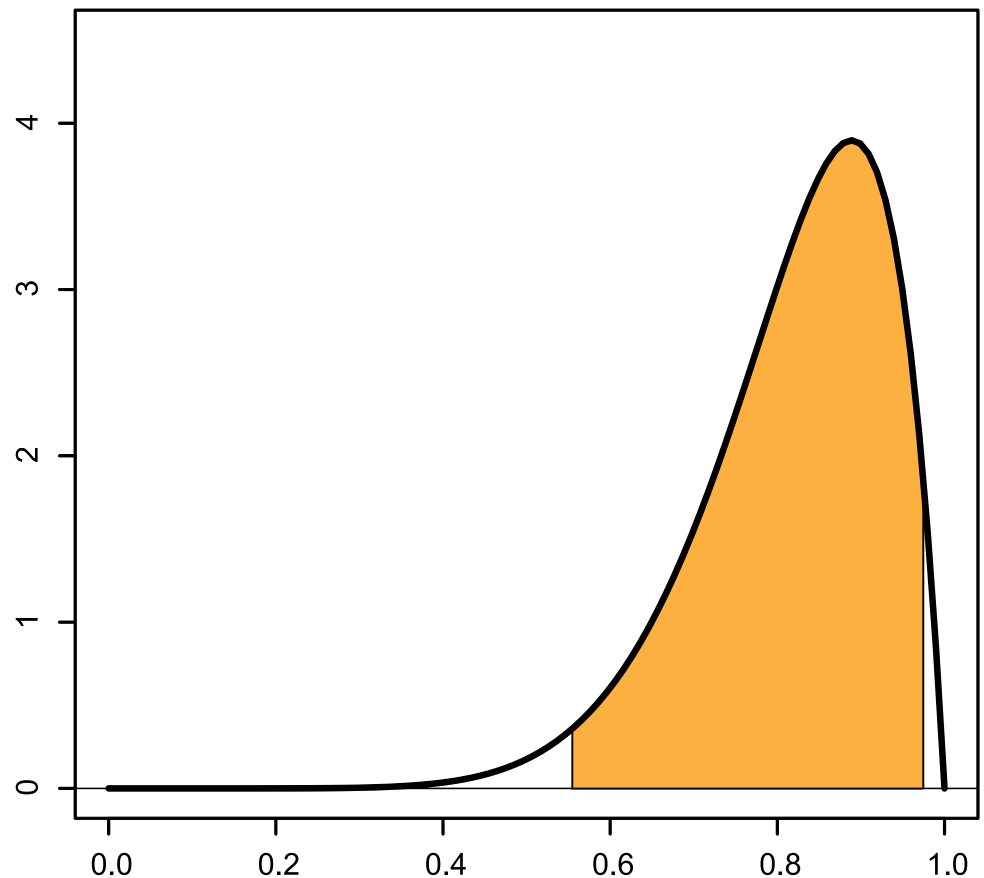
We can do this by simply constructed a histogram for any column in the file this provides an estimate of its marginal posterior probability density



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

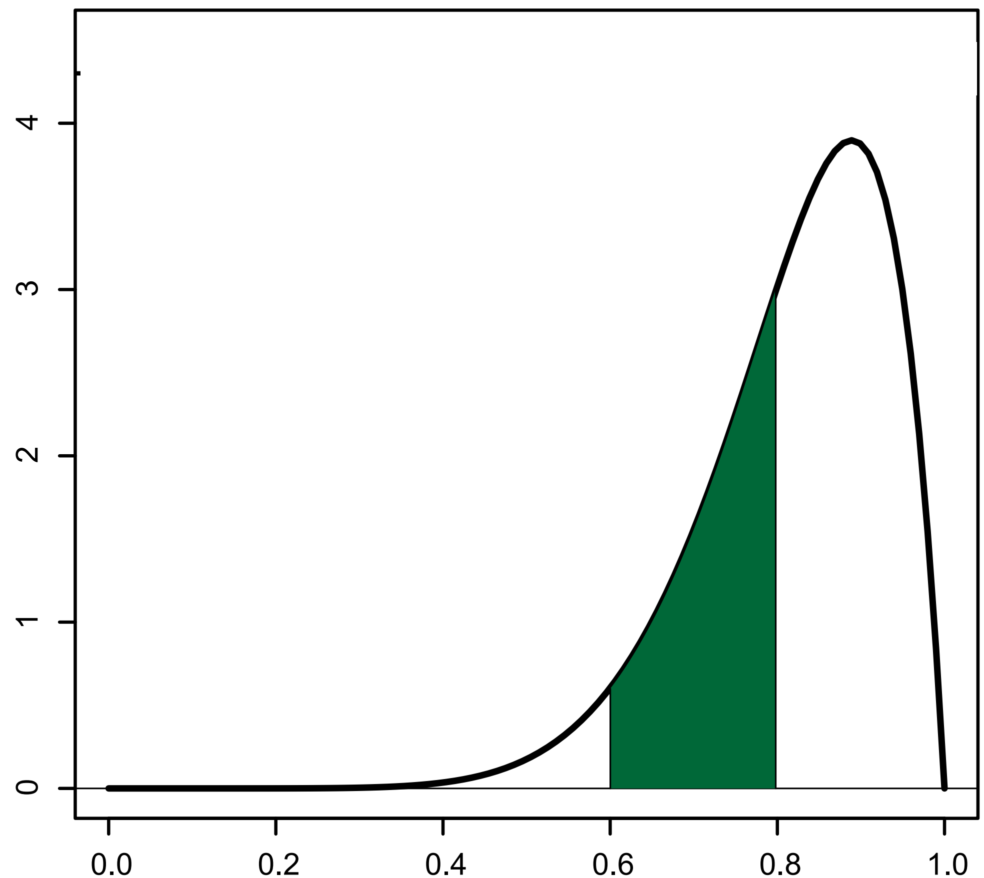
We can easily summarize aspects of the marginal posterior probability density: *e.g.*, to summarize the 95% credible interval.



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

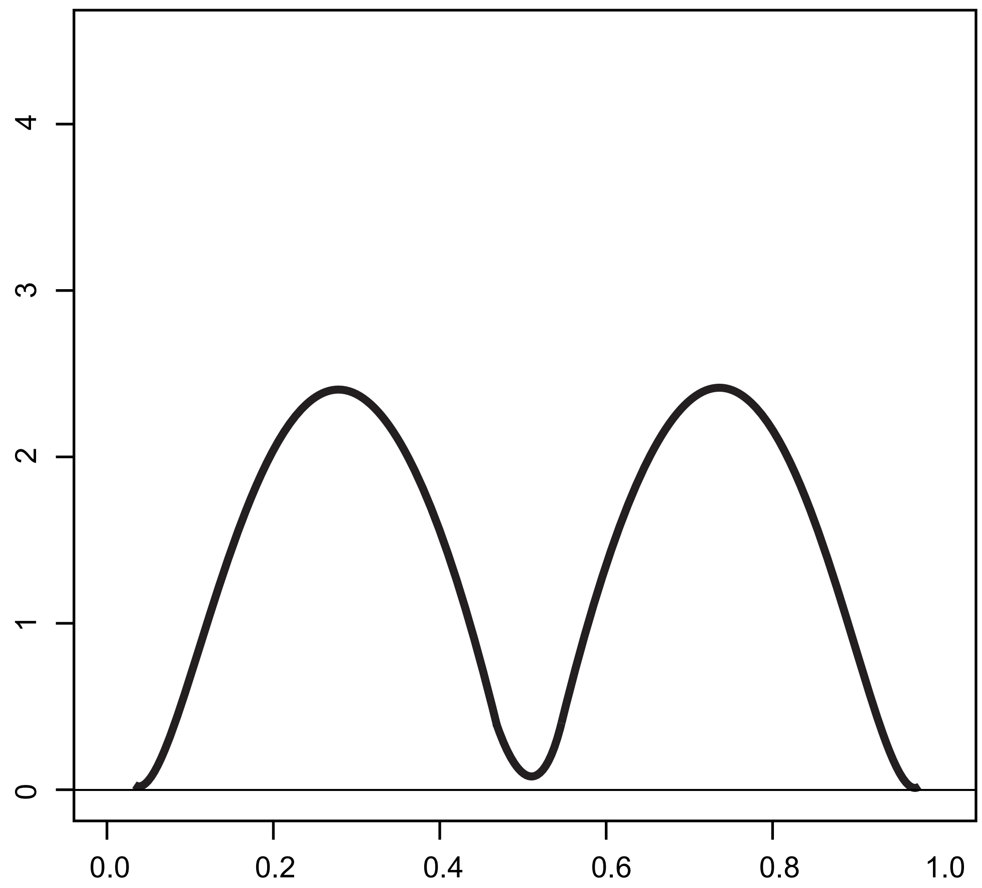
We can easily summarize aspects of the marginal posterior probability density:
e.g., or the probability within some arbitrary interval of interest (0.6–0.8).



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

We can easily summarize aspects of the marginal posterior probability density: *e.g.*, or we can summarize the highest posterior density (HPD) interval.



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

We can easily summarize aspects of the marginal posterior probability density: *e.g.*, or we can summarize the highest posterior density (HPD) interval.