

MCMC Simulation and Diagnosis



Brian R. Moore

Department of Evolution & Ecology
University of California, Davis
Bodega Phylogenetic Workshop, 2019

Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm

Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

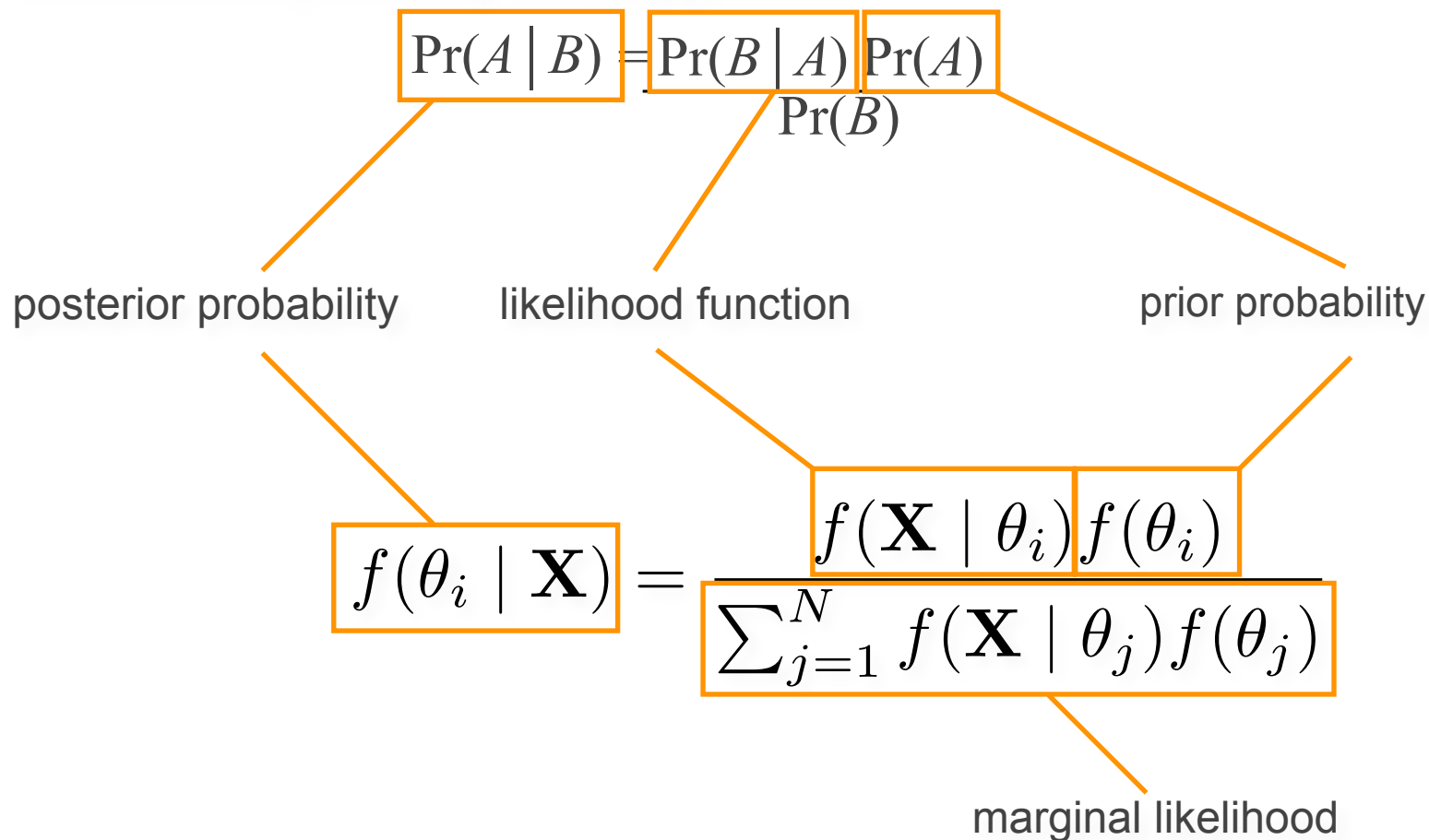
General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains

Bayesian Inference

Bayes Theorem

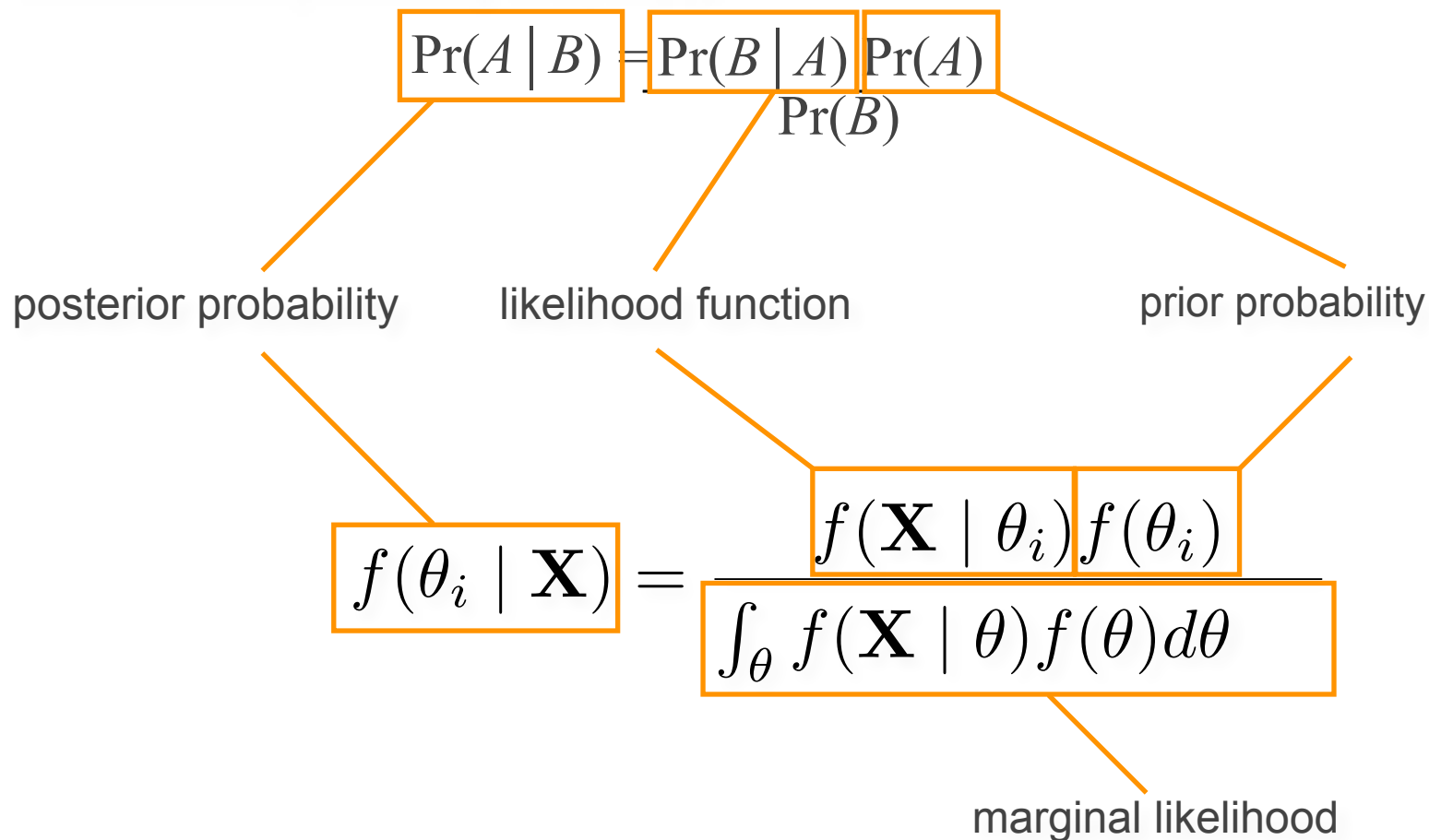
The posterior probability of observing A given that B has occurred, $\Pr(A | B)$, is proportional to the product of the conditional probability of $\Pr(A | B)$ and the unconditional probability of A , $\Pr(A)$.



Bayesian Inference

Bayes Theorem

The posterior probability of observing A given that B has occurred, $\Pr(A | B)$, is proportional to the product of the conditional probability of $\Pr(A | B)$ and the unconditional probability of A , $\Pr(A)$.



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: $\mathbf{X} = (x_1, x_2, x_3, \dots, x_N)$

II. Phylogenetic model parameters

1. Tree topology $\tau = (\tau_1, \tau_2, \dots, \tau_{(2S-5)!!})$
branch lengths $\nu = (\nu_1, \nu_2, \dots, \nu_{(2S-3)})$

\sim Uniform

\sim Dirichlet (1,...,1)

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

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

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

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

Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm

Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains

Approximating the Joint Posterior Probability Density using MCMC

Programming our MCMC robot...

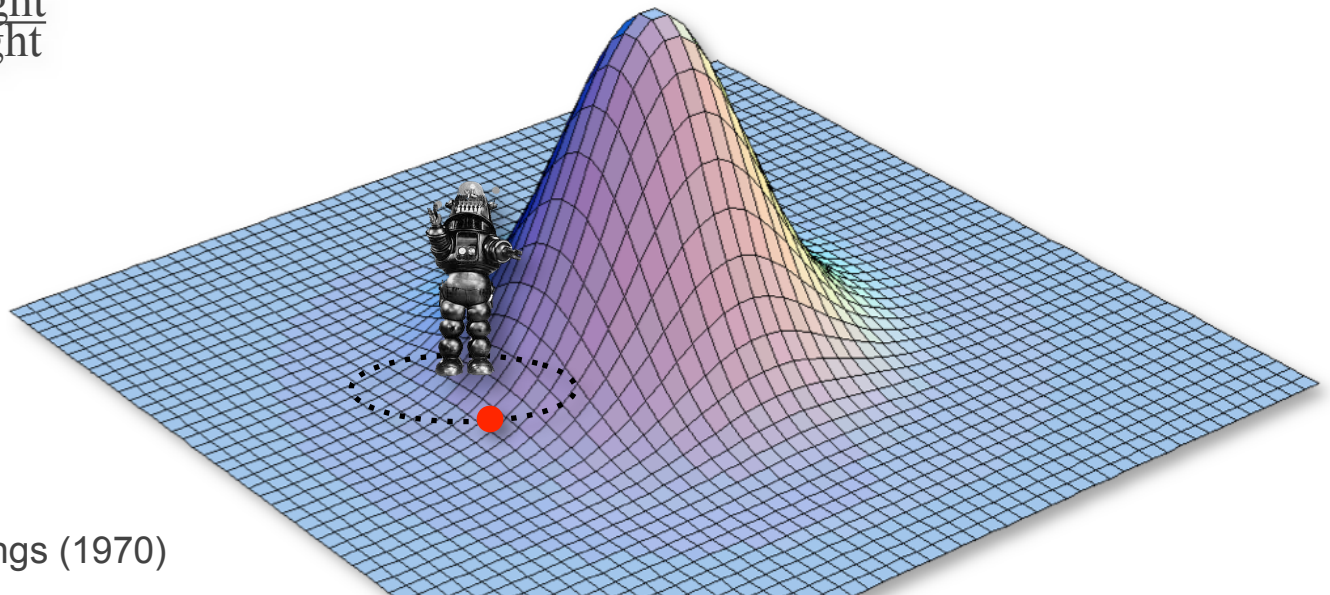
Our robot parachutes into a random location in the joint posterior density and will explore parameter space by following these simple rules:

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

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

< U step

> U stay

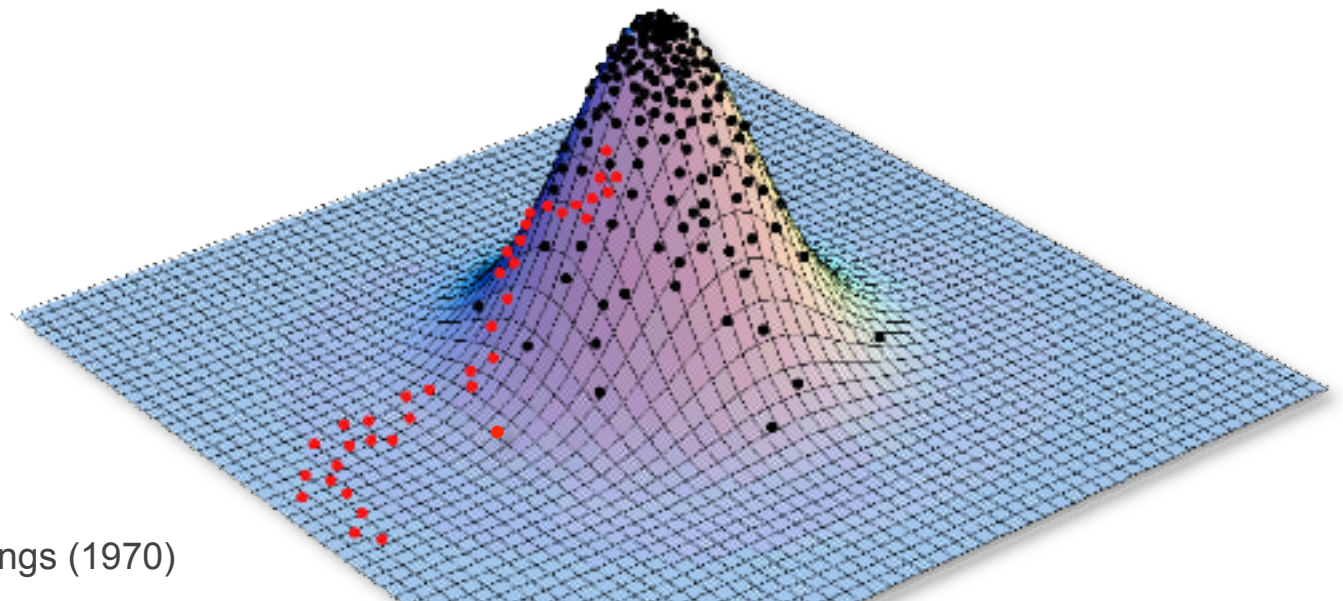


Approximating the Joint Posterior Probability Density using MCMC

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 $\Pr[A \rightarrow B] = \Pr[B \rightarrow A]$



Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

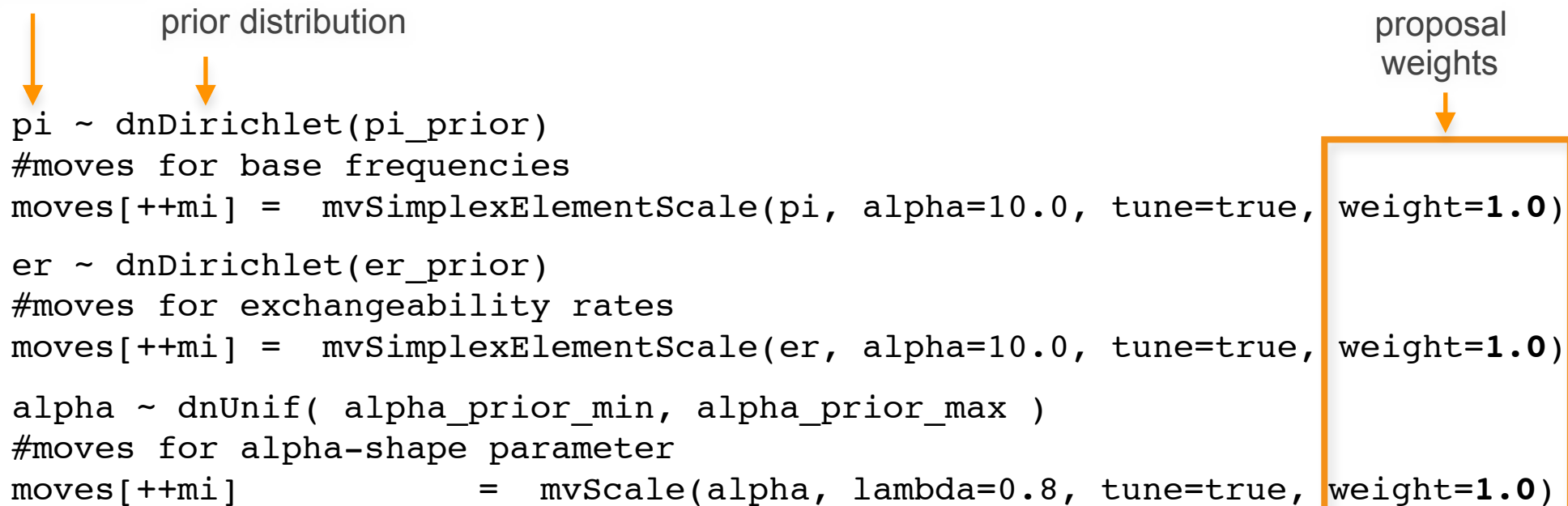
1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities

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)
```



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, $\theta = \{\tau, \nu, \Phi\}$
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

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, $\theta = \{\tau, \nu, \Phi\}$
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

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, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value, θ' , 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:

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

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

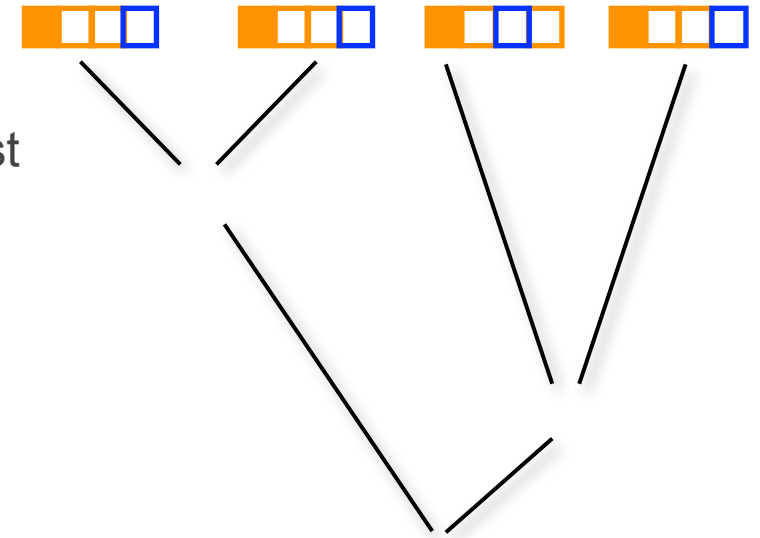
1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value, θ' , for the selected parameter via the proposal mechanism:
 - 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:
 - How do we calculate the likelihood for a given parameter value, θ, θ' ?

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

How Do We Calculate Site Likelihoods?

The Felsenstein Pruning Algorithm

Then we need to repeat the entire process for each of the more inclusive nodes toward the root, where the conditional likelihoods of the tips are first recorded...



$$\mathcal{L}_i^{(Anc)} = \left(\sum_{j \in (A, C, G, T)} p_{ij}(\nu_L) \mathcal{L}_j^{(L)} \right) \times \left(\sum_{k \in (A, C, G, T)} p_{ik}(\nu_R) \mathcal{L}_k^{(R)} \right)$$

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

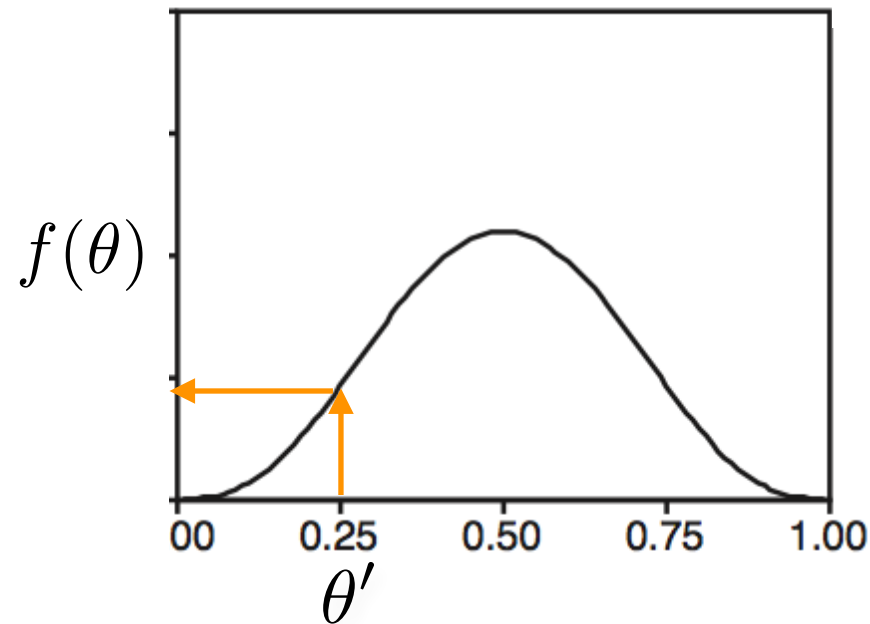
1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value, θ' , for the selected parameter via the proposal mechanism:
 - 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:
 - How do we calculate the prior probability for a given parameter value, θ, θ' ?

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

Approximating the Joint Posterior Probability Density using MCMC

The prior for each parameter is specified

We can just look up the prior probability of a given parameter value θ, θ'



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, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value, θ' , 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:
 - *i.e.*, we decide how to explore the posterior probability density based on the ratio of the posterior probabilities of the current and proposed parameter values, θ, θ'

$$R = \min \left[1, \frac{f(\mathbf{X}|\theta')}{f(\mathbf{X}|\theta)} \cdot \frac{f(\theta')}{f(\theta)} \right]$$

posterior probability of
proposed state
posterior probability of
current state

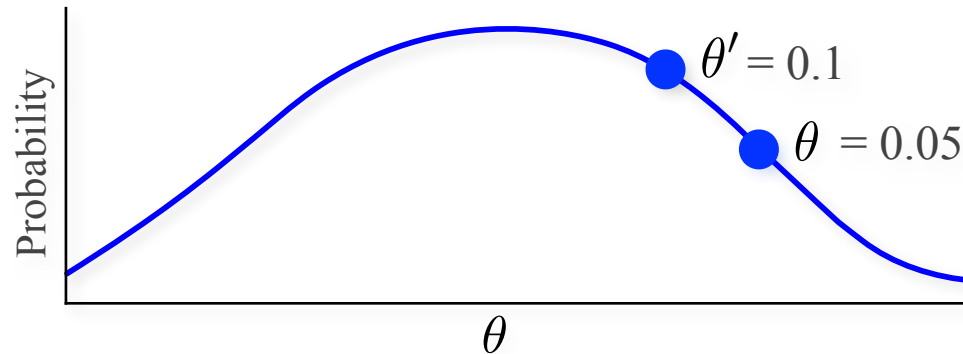
Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities
3. Propose a new value, θ' , 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
5. Generate a uniform random variable, $U[0,1]$, accept if $R > U$
6. Repeat steps 2–5 an ‘adequate’ number of times

Approximating the Joint Posterior Probability Density using MCMC

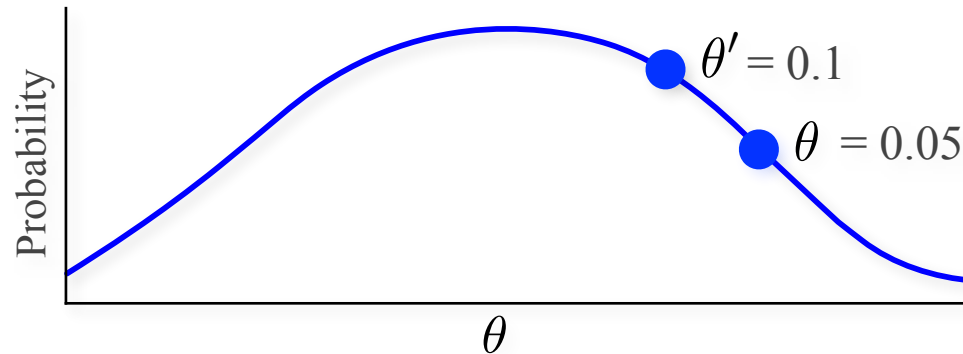
The Metropolis-Hastings algorithm



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

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

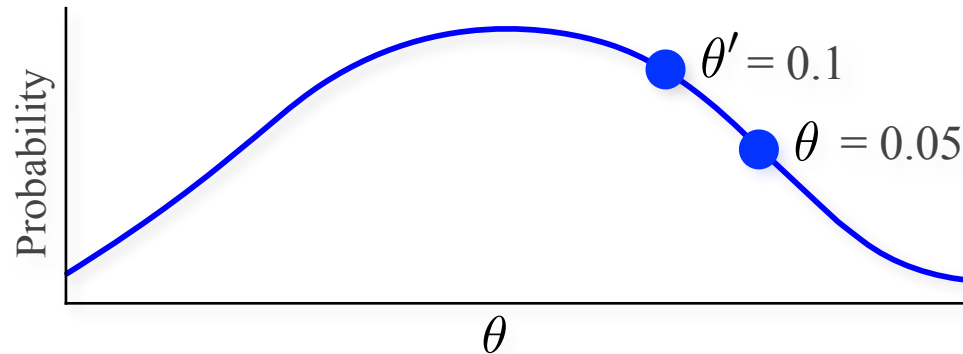


$$R = \min \left[1, \frac{f(\theta' | \mathbf{X})}{f(\theta | \mathbf{X})} \right]$$

posterior
ratio

Approximating the Joint Posterior Probability Density using MCMC

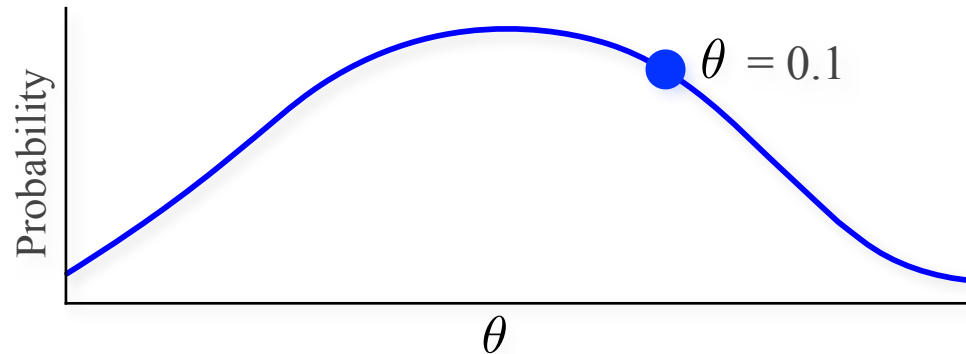
The Metropolis-Hastings algorithm



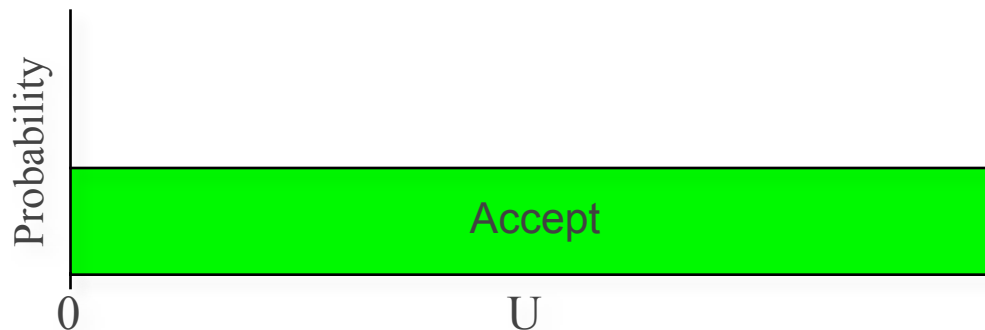
$$R = \min \left[1, \frac{f(\theta' | \mathbf{X})}{f(\theta | \mathbf{X})} \right] = \frac{0.1}{0.05}$$

Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

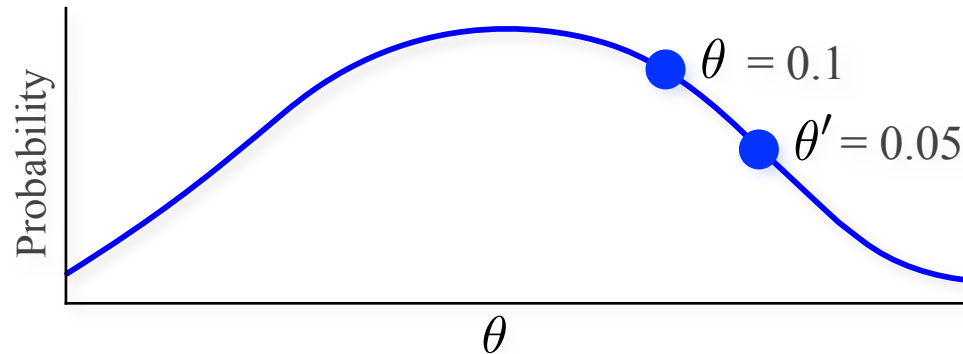


$$R = \min \left[1, \frac{f(\theta' | \mathbf{X})}{f(\theta | \mathbf{X})} \right] = \frac{0.1}{0.05}$$

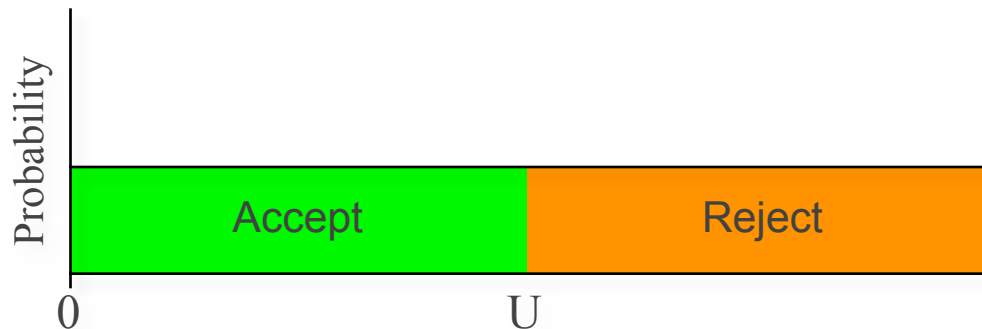


Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

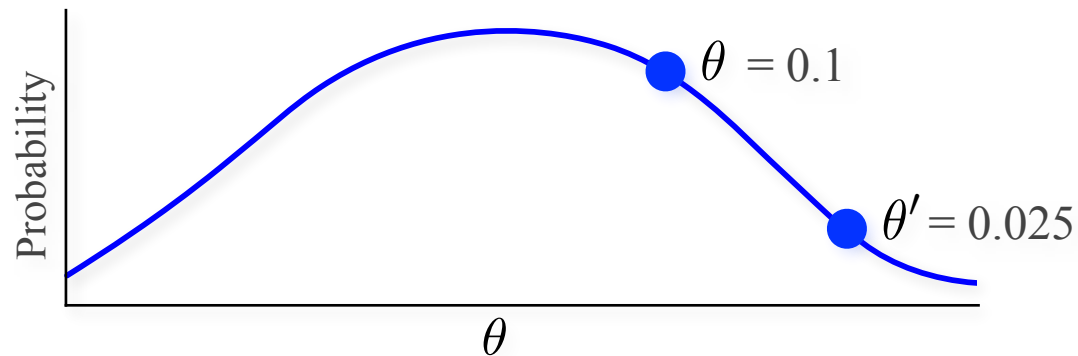


$$R = \min \left[1, \frac{f(\theta' | \mathbf{X})}{f(\theta | \mathbf{X})} \right] = \frac{0.05}{0.1} = 0.5$$



Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm




$$R = \min \left[1, \frac{f(\theta' | \mathbf{X})}{f(\theta | \mathbf{X})} \right] = \frac{0.025}{0.1} = 0.25$$



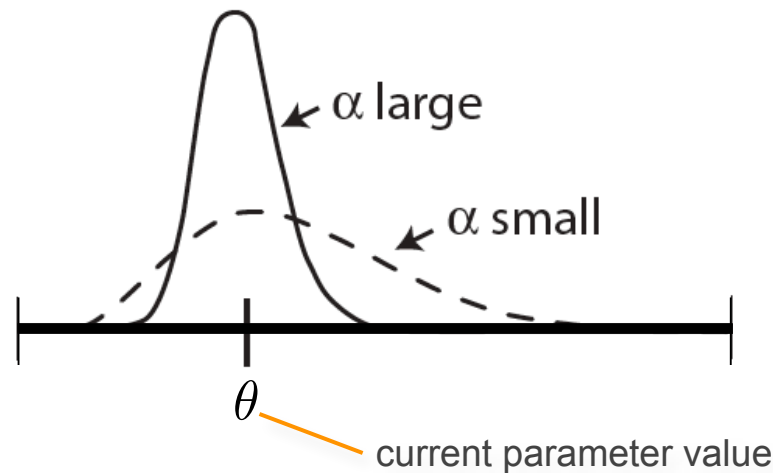
Approximating the Joint Posterior Probability Density using MCMC

The Metropolis-Hastings algorithm

1. Initialize the chain with some random values for all parameters, including the tree with branch lengths, $\theta = \{\tau, \nu, \Phi\}$
2. Select a parameter to update (alter) according to the proposal probabilities
-  3. **Propose a new value**, θ' , 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
5. Generate a uniform random variable, $U[0,1]$, accept if $R > U$
6. Repeat steps 2–5 an ‘adequate’ number of times

Approximating the Joint Posterior Probability Density using MCMC

Dirichlet proposal mechanism



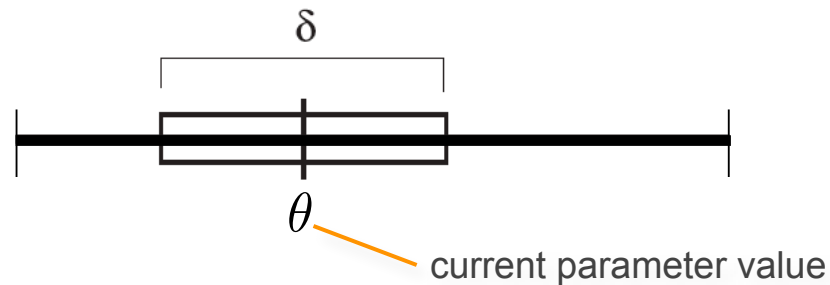
Propose a new value for a parameter with a Dirichlet prior probability density

- the proposal density is controlled by the tuning parameter, α
- when α is large, smaller changes will be proposed
- when α is small, larger changes will be proposed

```
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



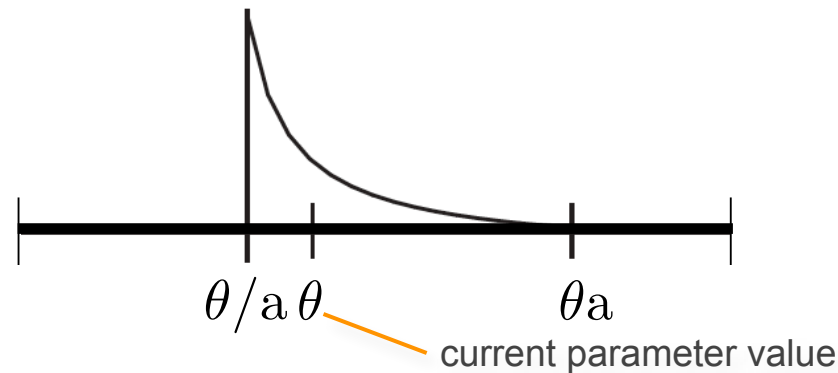
Propose a new value for a parameter with a uniform prior probability density

- the proposal density is controlled by the tuning parameter, δ
- when δ is large, larger changes will be proposed
- when δ is small, smaller changes will be proposed

```
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



Propose a new value for a parameter with a exponential prior probability density

- the proposal density is controlled by the tuning parameter, $\lambda = 2\ln a$
- when λ is large, larger changes will be proposed
- when λ is small, smaller changes will be proposed

Equivalent to sliding-window proposal with log-transformed x axis

Works well when changes to small parameter values have a larger impact on probability of data than changes in large parameter values

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


Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm



Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains

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 (unmodified).
 - posterior of chain i is raised to a power, β_i : the heat of chain $i = 1/(1 + iT)$

temperature		
chain	0.25	
0	1.00	$\beta_0 = 1/(1 + 0 \cdot 0.25)$
1	0.80	$\beta_1 = 1/(1 + 1 \cdot 0.25)$
2	0.67	$\beta_2 = 1/(1 + 2 \cdot 0.25)$
3	0.57	$\beta_3 = 1/(1 + 3 \cdot 0.25)$

- the incremental heating successively ‘flattens’ the posterior visited by each chain by making the acceptance probability of the i^{th} chain more ‘permissive’:

$$R_i = \min \left[1, \left(\frac{f(\mathbf{X}|\theta')}{f(\mathbf{X}|\theta)} \cdot \frac{f(\theta')}{f(\theta)} \right)^{\beta_i} \cdot \frac{f(\theta|\theta')}{f(\theta'|\theta)} \right]$$

- this allows heated chains to more readily traverse regions of low probability.
- the degree of incremental heating is controlled by the temperature parameter, T .

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 (unmodified).
 - posterior of chain i is raised to a power, β_i : the heat of chain $i = 1/(1 + iT)$

	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

- the incremental heating successively ‘flattens’ the posterior visited by each chain by making the acceptance probability of the i^{th} chain more ‘permissive’:

$$R_i = \min \left[1, \left(\frac{f(\mathbf{X}|\theta')}{f(\mathbf{X}|\theta)} \cdot \frac{f(\theta')}{f(\theta)} \right)^{\beta_i} \cdot \frac{f(\theta|\theta')}{f(\theta'|\theta)} \right]$$

- this allows heated chains to more readily traverse regions of low probability.
- the degree of incremental heating is controlled by the temperature parameter, T .

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 (unmodified).
 - posterior of chain i is raised to a power, β_i : the heat of chain $i = 1/(1 + iT)$

		temperature			
	chain	0.25	0.20	0.15	0.10
cold chain	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

- the incremental heating successively ‘flattens’ the posterior visited by each chain by making the acceptance probability of the i^{th} chain more ‘permissive’:

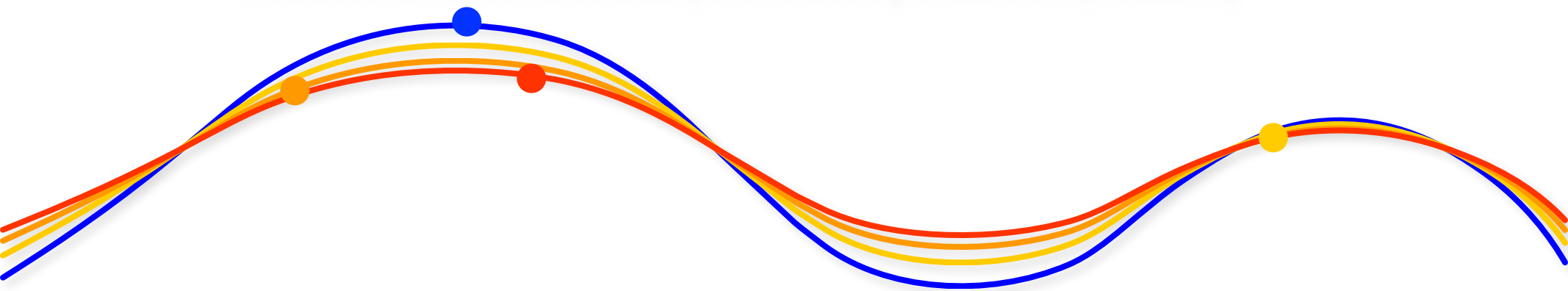
$$R_i = \min \left[1, \left(\frac{f(\mathbf{X}|\theta')}{f(\mathbf{X}|\theta)} \cdot \frac{f(\theta')}{f(\theta)} \right)^{\beta_i} \cdot \frac{f(\theta|\theta')}{f(\theta'|\theta)} \right]$$

- this allows heated chains to more readily traverse regions of low probability.
- the degree of incremental heating is controlled by the temperature parameter, T .
- samples are only collected by the ‘cold’ chain (*i.e.*, the undistorted posterior).

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 (unmodified).
 - posterior of chain i is raised to a power, β_i : the heat of chain $i = 1/(1 + iT)$
 - the cold chain samples the true posterior, whereas the heated chains sample successively 'flattened' distortions of the posterior
 - heated chains to more readily traverse regions of low probability



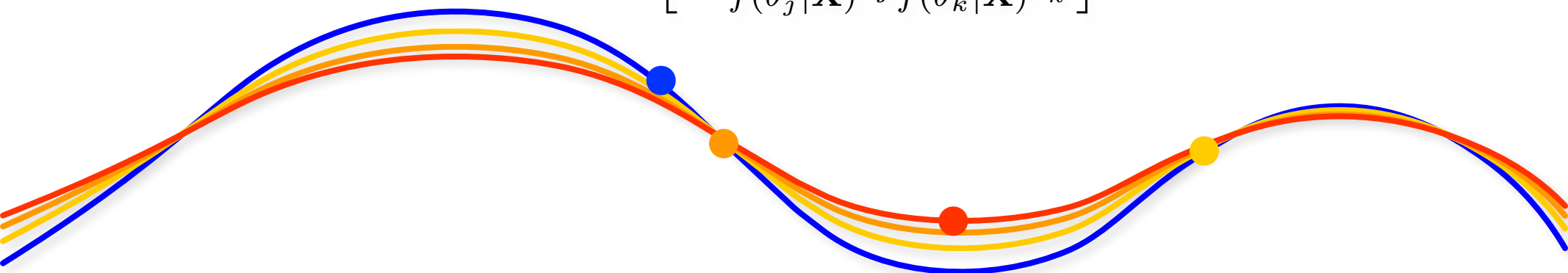
- chain 0 ($\beta = 1.00$)
- chain 1 ($\beta = 0.83$)
- chain 2 ($\beta = 0.71$)
- chain 3 ($\beta = 0.63$)

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 (unmodified).
3. At prescribed intervals, two chains are randomly selected to swap.
 - we compute the acceptance probability of swapping the two chains.

$$R = \min \left[1, \frac{f(\theta_k | \mathbf{X})^{\beta_j} f(\theta_j | \mathbf{X})^{\beta_k}}{f(\theta_j | \mathbf{X})^{\beta_j} f(\theta_k | \mathbf{X})^{\beta_k}} \right]$$

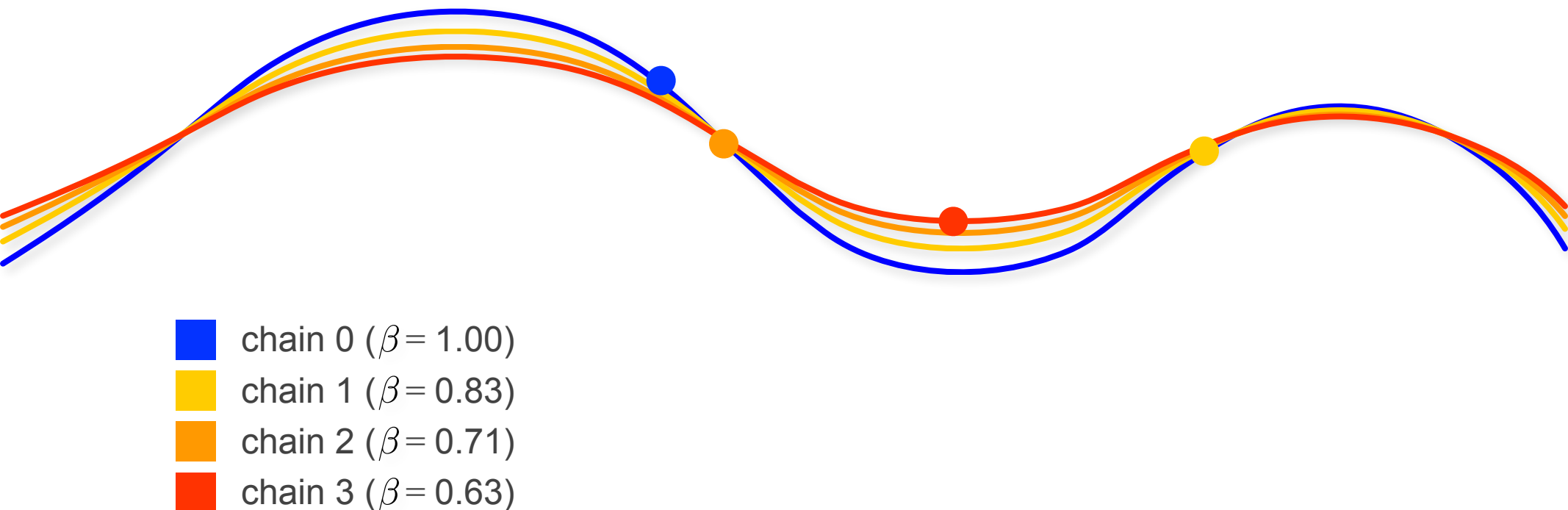


- chain 0 ($\beta = 1.00$)
- chain 1 ($\beta = 0.83$)
- chain 2 ($\beta = 0.71$)
- chain 3 ($\beta = 0.63$)

Approximating the Joint Posterior Probability Density using Metropolis-Coupled MCMC

The MC³ algorithm

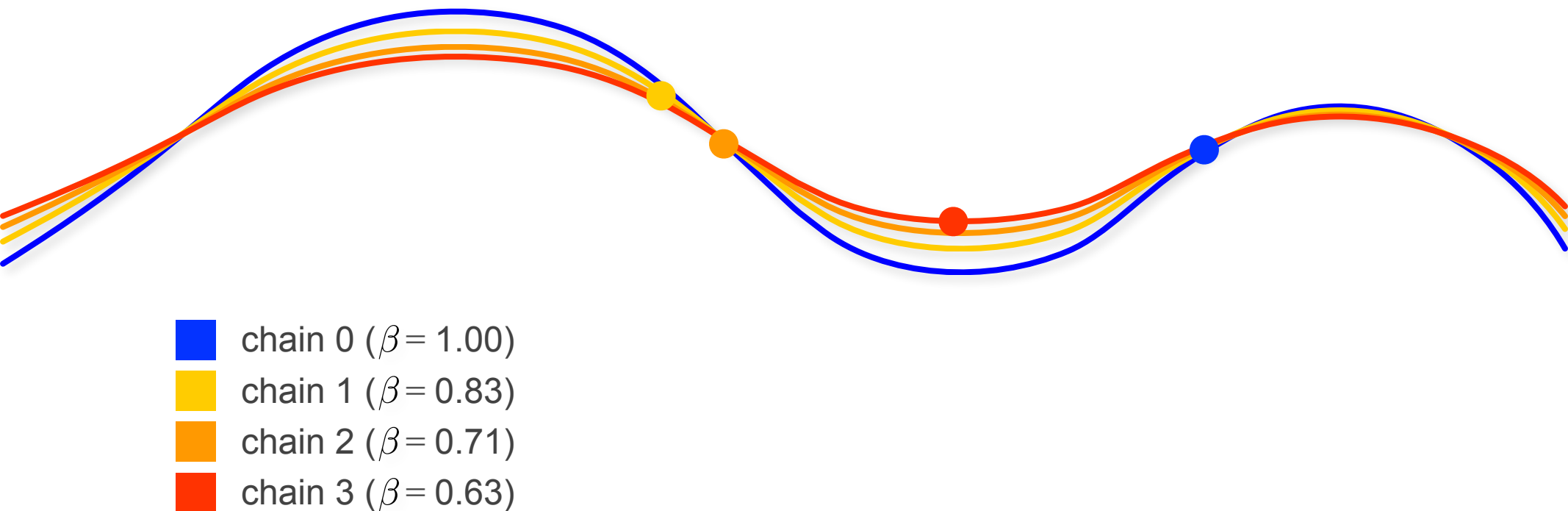
1. Initialize N independent M-H MCMC chains with random values for all parameters.
2. The chains are incrementally heated, such that the first chain is cold (unmodified).
3. At prescribed intervals, two chains are randomly selected to swap.
 - we compute the acceptance probability of swapping the two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



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 (unmodified).
3. At prescribed intervals, two chains are randomly selected to swap.
4. Only samples from the cold chain are used to approximate the posterior.



Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm



Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains

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 row in our log file—with values of all model parameters—is a sample from the *joint* posterior probability density.

[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	0.166667	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 column in our log file—with values for a single model parameter—is a sample from the *marginal* posterior probability density.

[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	0.166667	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.

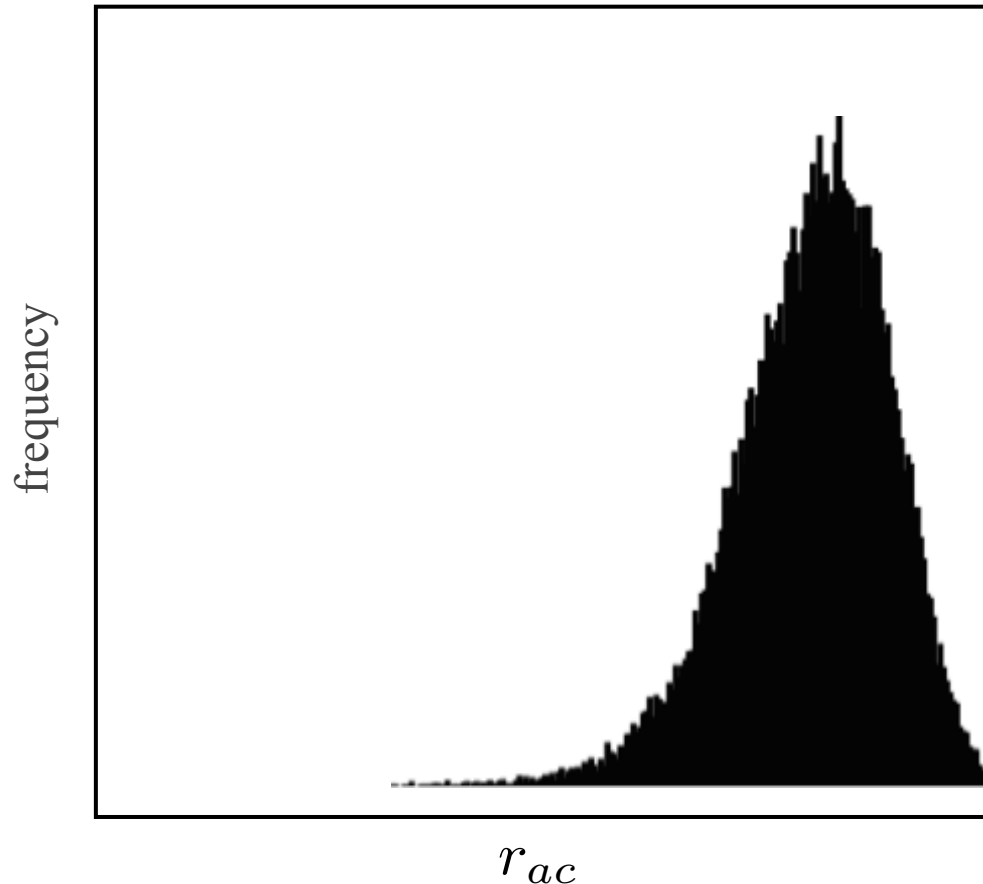
[ID: 2325481386]

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	0.166667	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

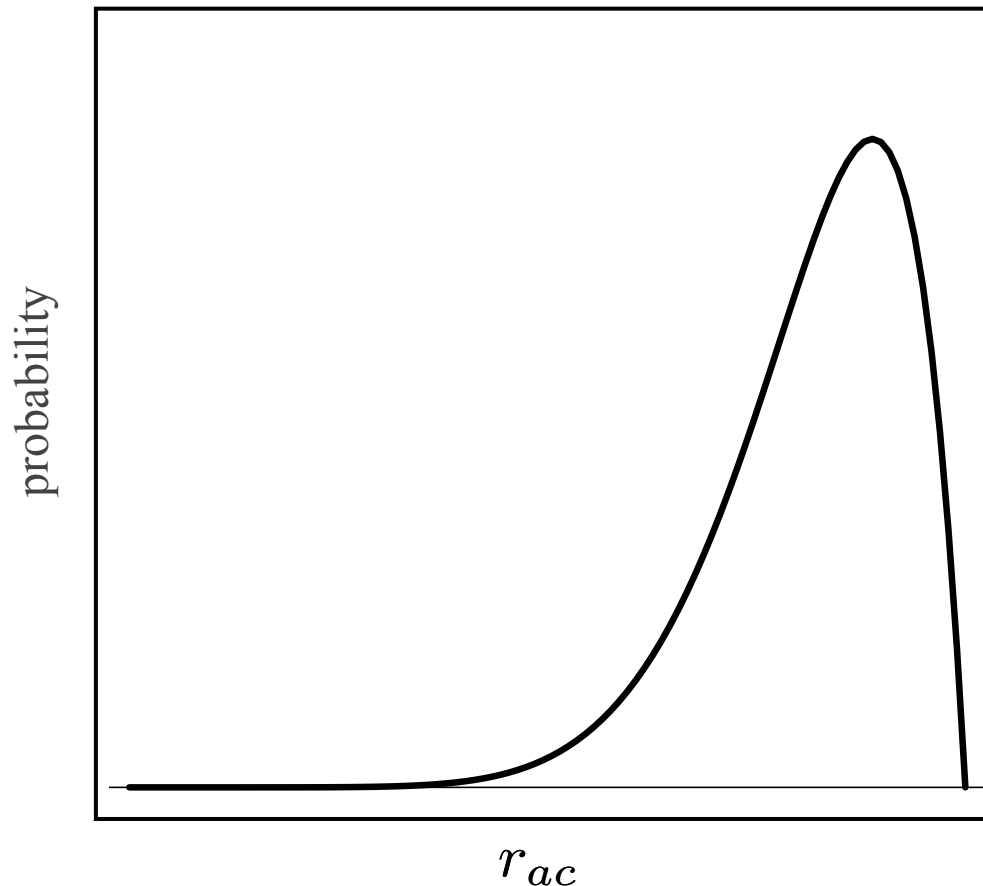
We can do this by simply constructed a histogram for any column in the file
this provides an estimate of its marginal posterior probability density



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

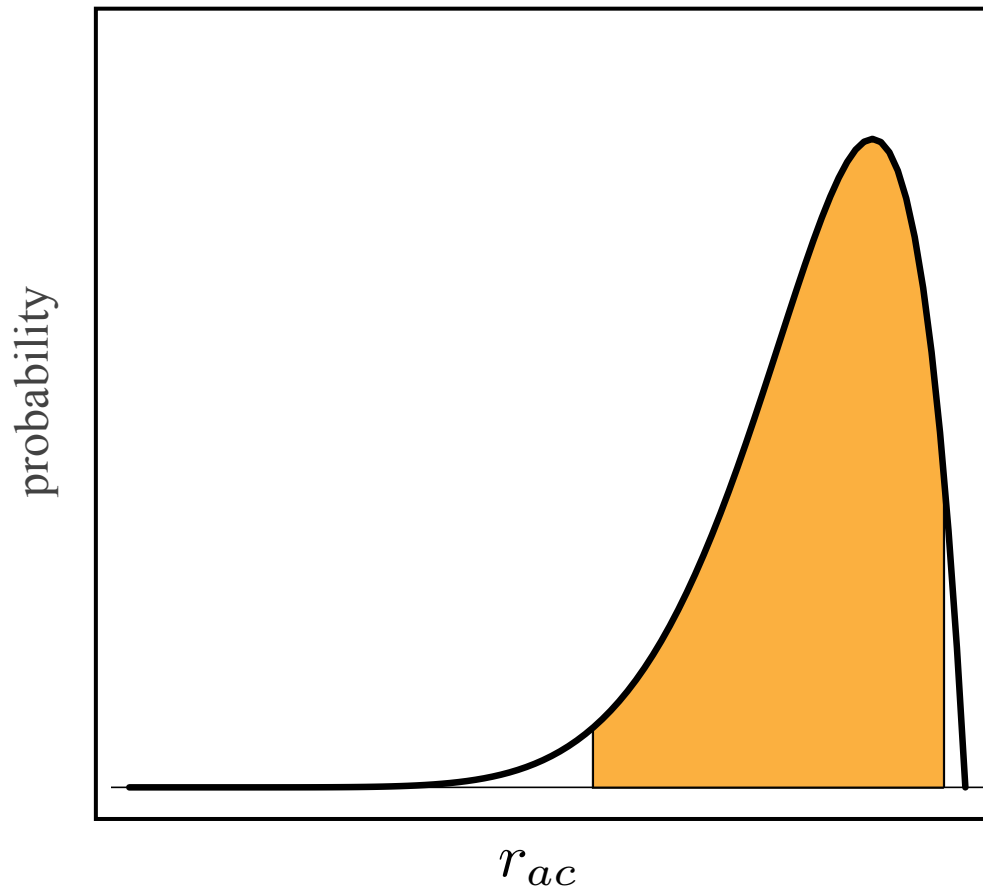
We can do this by simply constructed a histogram for any column in the file
this provides an estimate of its marginal posterior probability density



Approximating the Joint Posterior Probability Density using MCMC

Samples from the MCMC simulation approximate the joint posterior

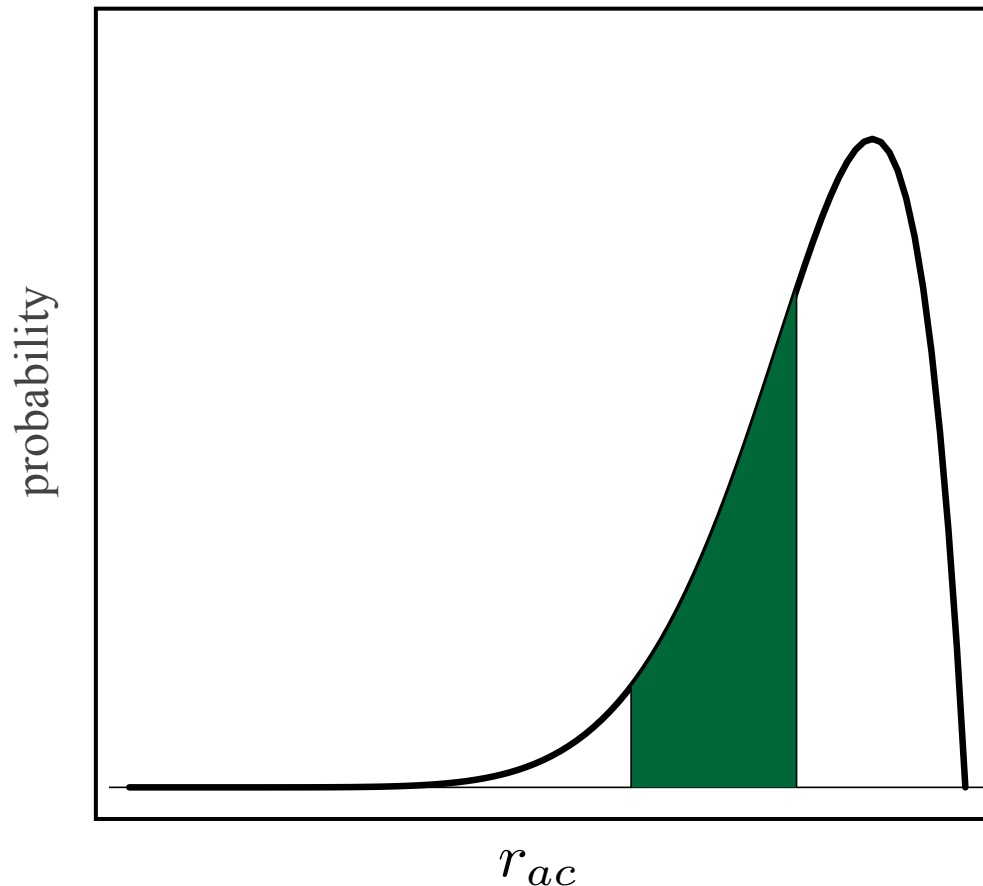
We can easily summarize aspects of the marginal posterior probability density:
e.g., to summarize the 95% credible interval.



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).



Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm

Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains

Approximating the Joint Posterior Probability Density using MCMC

MCMC in theory and practice

MCMC in theory...

an **appropriately constructed** and **adequately run** chain is guaranteed to provide an arbitrarily precise description of the joint stationary density

MCMC in practice...

although a given sampler may work well in most cases, all samplers will fail in some cases, and is not guaranteed to work for any particular case

Q. When do we know that the MCMC provides an accurate approximation for a given empirical analysis?

A.

NEVER!

Approximating the Joint Posterior Probability Density using MCMC

MCMC performance

It is not sufficient to merely be deeply concerned about MCMC performance...
you need to be **completely obsessed** about it!
for **any** Bayesian inference based on MCMC
particularly for complex models/inference problems



WE
ARE
HERE

careless



WE
SHOULD
BE HERE

careful



I'LL
BE
HERE

paranoid

Approximating the Joint Posterior Probability Density using MCMC

Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review

Mary Kathryn COWLES and Bradley P. CARLIN

A critical issue for users of Markov chain Monte Carlo (MCMC) methods in applications is how to determine when it is safe to stop sampling and use the samples to estimate characteristics of the distribution of interest. Research into methods of computing theoretical convergence bounds holds promise for the future but to date has yielded relatively little of practical use in applied work. Consequently, most MCMC users address the convergence problem by applying diagnostic tools to the output produced by running their samplers. After giving a brief overview of the area, we provide an expository review of 13 convergence diagnostics.

...under simulation, all MCMC diagnostics may fail to detect the exact problems that they were specifically designed to identify...

...therefore, it is critical to use a combination of tools to detect MCMC failure

KEY WORDS: Autocorrelation; Gibbs sampler; Metropolis-Hastings algorithm.

Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm

Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics



IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains

Assessing MCMC Performance: Three Main Issues

1. Convergence

Has the chain (robot) successfully targeted the stationary distribution?

2. Mixing

Is the chain (robot) efficiently integrating over the joint posterior probability?

3. Sampling intensity

Have we collected enough samples to adequately describe the posterior probability distribution?

Assessing MCMC Performance: Based on Single Chains

1. Convergence diagnostics

Time-series plots of parameter estimates

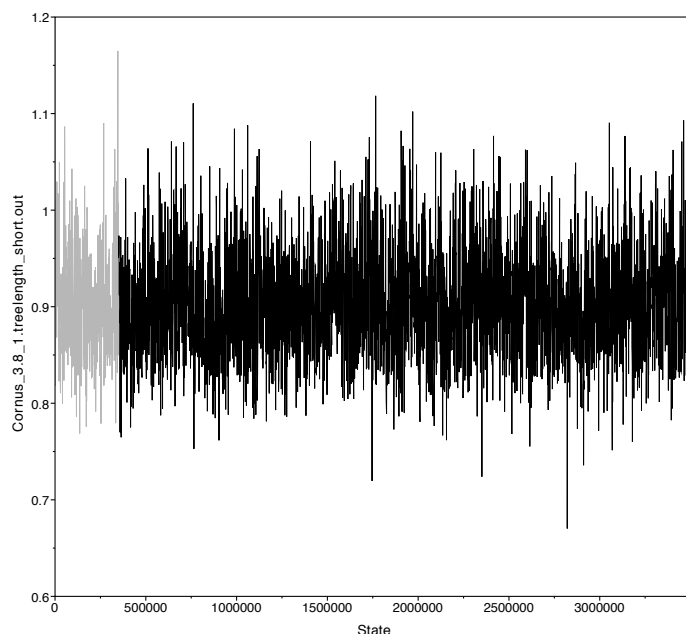
Continuous parameters (e.g., substitution rates)

- some parameters are more reliable than others
- steps may occur!

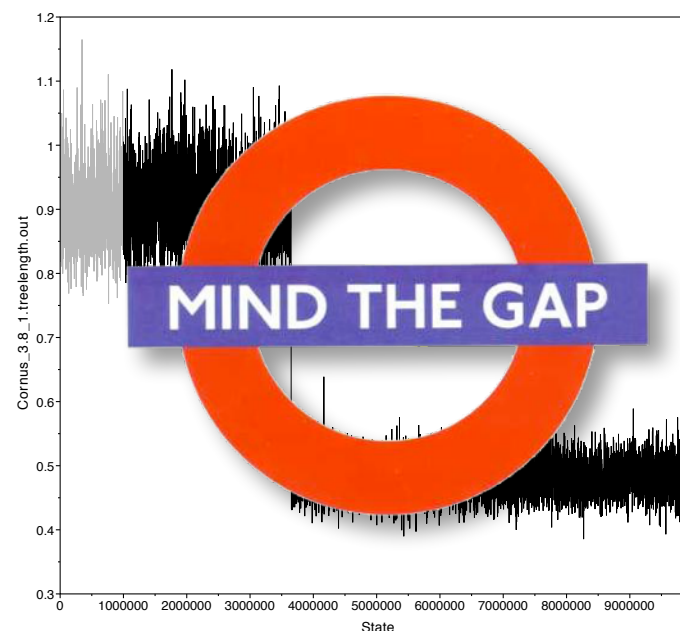
Assessing MCMC Performance: Based on Single Chains

Example: Tracer plots of tree-length at two stages of a single MrBayes run

all looks good...



until it doesn't



fast*

slow*



*somewhat data-set dependent

Assessing MCMC Performance: Based on Single Chains

1. Convergence diagnostics

Time-series plots of parameter estimates

Continuous parameters (*e.g.*, substitution rates)

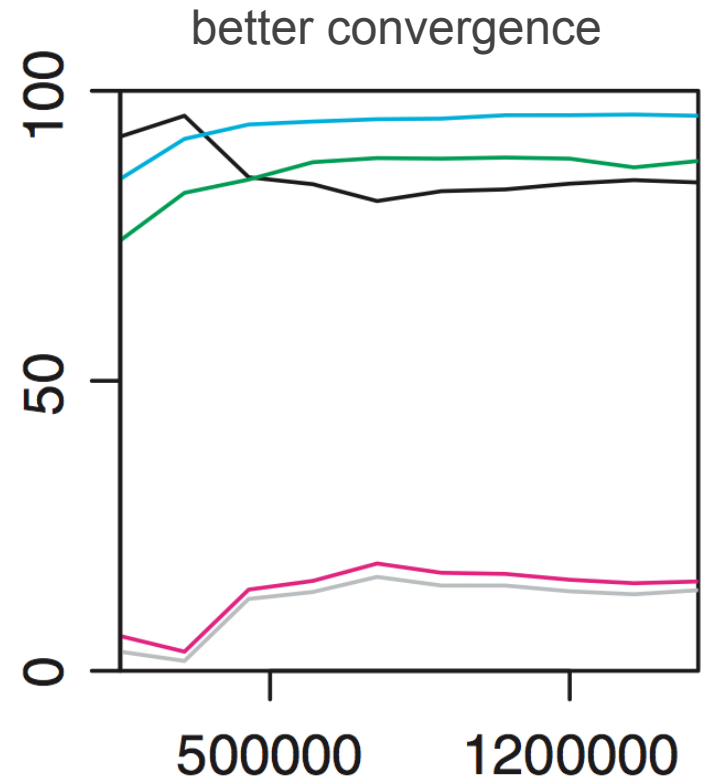
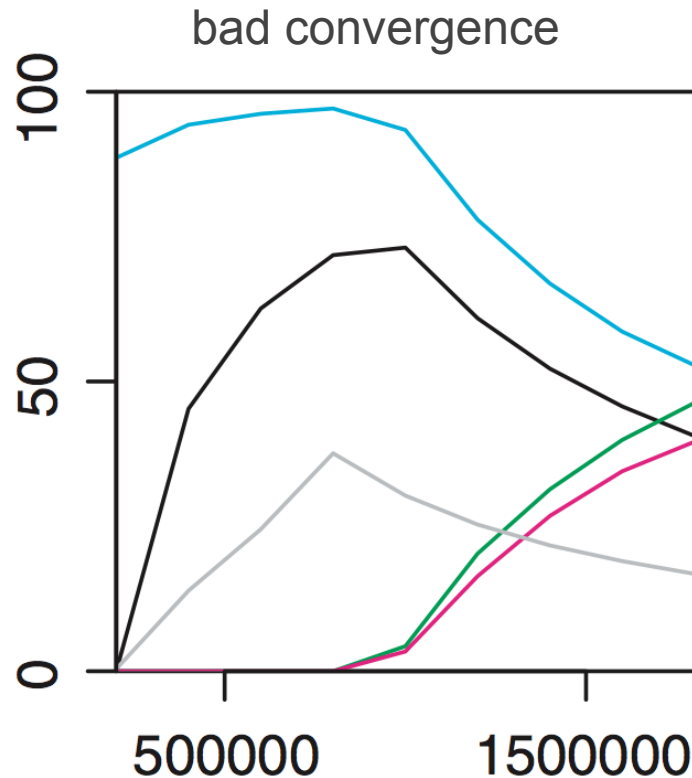
- some parameters are more reliable than others
- steps may occur!

Discrete parameters (*e.g.*, bi-partitions)

- some parameters are more reliable than others
- steps may occur!

Assessing MCMC Performance: Based on Single Chains

Example: AWTY plots of cumulative bi-partition frequency of 5 nodes



Assessing MCMC Performance: Based on Single Chains

1. Convergence diagnostics

Time-series plots of parameter estimates

Geweke diagnostic (coda, BOA)

Continuous or discrete parameters

- A test for equality of the means of the first and last part of a Markov chain (by default the first 10% and the last 50%)
- If the samples are drawn from the stationary distribution, the two means should equal and Geweke's statistic has an asymptotically standard normal distribution

Assessing MCMC Performance: Based on Single Chains

1. Convergence diagnostics

Time-series plots of parameter estimates

Geweke diagnostic (coda, BOA)

Heidelberg-Welch diagnostic (coda, BOA)

Continuous or discrete parameters

- uses the Cramer-von Mises statistic to test the null hypothesis that the sampled values come from a stationary distribution
- This test is successively applied, first to the whole chain, then after discarding the first 10%, 20%, ... of the samples until either the null hypothesis is accepted, or 50% of the chain has been discarded
- The latter outcome constitutes “failure” of the test and indicates that a longer run is needed
- Otherwise, the number of iterations to keep and the number to discard (burn-in) are reported

Assessing MCMC Performance: Based on Single Chains

1. Convergence diagnostics

Time-series plots of parameter estimates

Geweke diagnostic (coda, BOA)

Heidelberg-Welch diagnostic (coda, BOA)

(many others)

Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Form of the time-series plots of parameter estimates

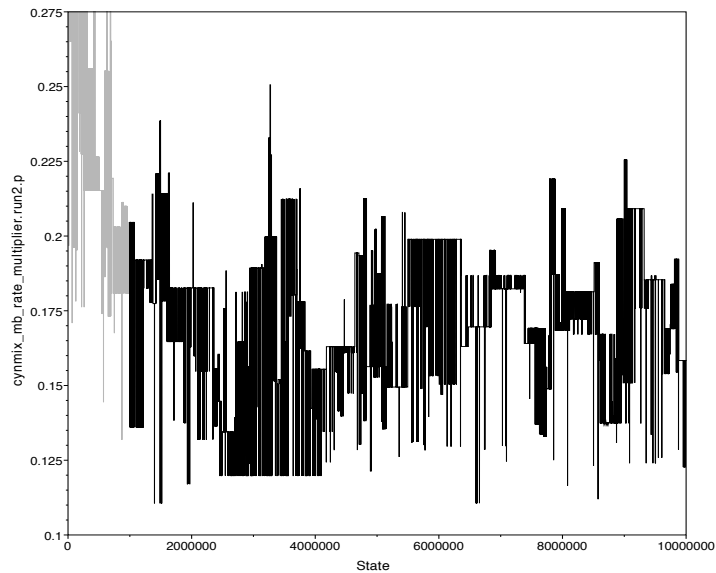
Continuous parameters (e.g., substitution rates)

- warm and fuzzy caterpillars

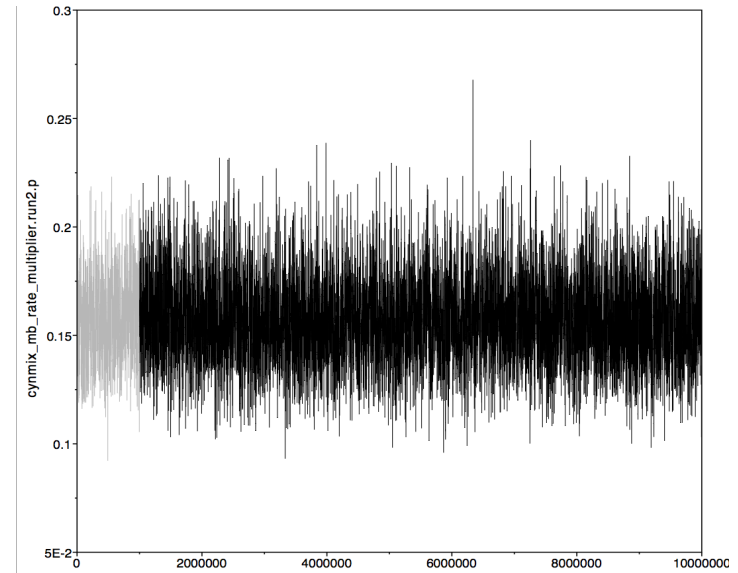
Assessing MCMC Performance: Based on Single Chains

Example: Tracer plots of relative-rate multipliers from two MrBayes runs

bad mixing



better mixing



Assessing MCMC Performance: Based on Single Chains

Example: Tracer plots of relative-rate multipliers from two MrBayes runs

bad mixing



better mixing



Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Form of the time-series plots of parameter estimates

Continuous parameters (e.g., substitution rates)

- warm and fuzzy caterpillars

Acceptance rates of parameter updates

Continuous and discrete parameters (MrBayes, BEAST)

- rates should ideally fall in the ~20–70% range

Assessing MCMC Performance: Based on Single Chains

Example: Tracer plots of relative-rate multipliers from two MrBayes runs

bad mixing



Acceptance rates for the moves in the "cold" chain of run 1:

With prob. Chain accepted changes to

13.61 % param. 1 (revmat) with Dirichlet proposal

.

.

.

0.04 % param. 34 (rate multiplier) Dirichlet proposal

6.59 % param. 35 (topology and branch lengths) TBR

14.06 % param. 35 (topology and branch lengths) LOCAL

better mixing



Acceptance rates for the moves in the "cold" chain of run 1:

With prob. Chain accepted changes to

33.30 % param. 1 (revmat) with Dirichlet proposal

.

.

.

19.13 % param. 34 (rate multiplier) Dirichlet proposal

17.40 % param. 35 (topology and branch lengths) TBR

29.76 % param. 35 (topology and branch lengths) LOCAL

Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Form of the time-series plots of parameter estimates

Continuous parameters (e.g., substitution rates)

- warm and fuzzy caterpillars

Acceptance rates of parameter updates

Continuous and discrete parameters (MrBayes, BEAST)

- rates should ideally fall in the ~20–70% range
- acceptance rates can be controlled by varying the scale of the tuning parameters for the relevant proposal mechanisms
- to increase acceptance rates, decrease scale of tuning parameter (and vice versa)

parameter

prior distribution

tuning
parameter

proposal
weights

↓
pi ~ dnDirichlet(pi_prior)
#moves for base frequencies

moves[++mi] = mvSimplexElementScale(pi, alpha=10.0, tune=true, weight=1.0)



Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Form of the time-series plots of parameter estimates

Continuous parameters (e.g., substitution rates)

- warm and fuzzy caterpillars

Acceptance rates of parameter updates

Continuous and discrete parameters (MrBayes, BEAST)

- rates should ideally fall in the ~20–70% range

Form of the marginal posterior probability densities

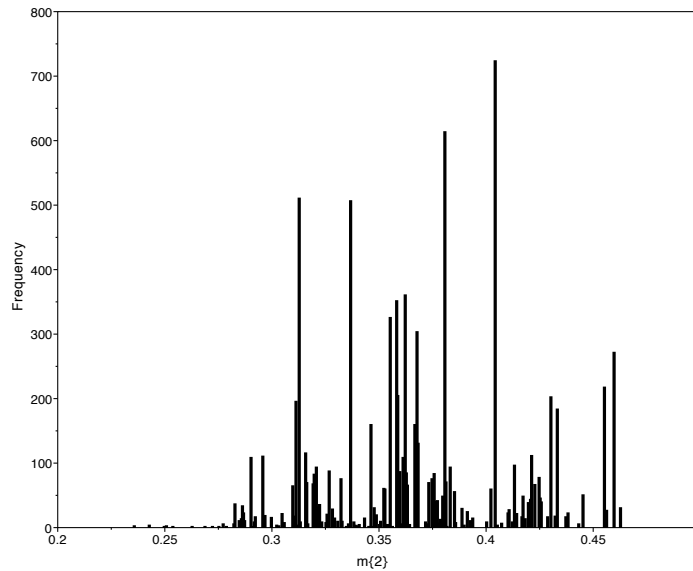
Continuous parameters (e.g., substitution rates)

- beware of porcupine roadkill!

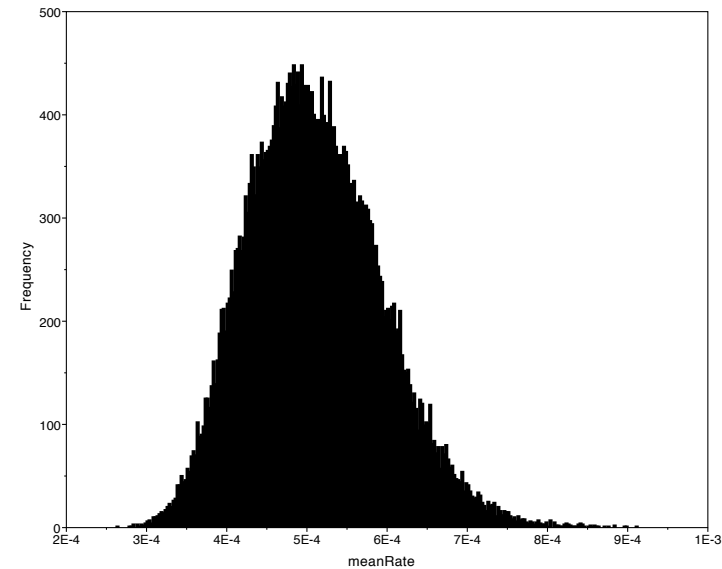
Assessing MCMC Performance: Based on Single Chains

Example: Tracer plots of relative-rate multipliers from two MrBayes runs

bad mixing



better mixing



Acceptance rates for the moves in the "cold" chain of run 1:

With prob. Chain accepted changes to
13.61 % param. 1 (revmat) with Dirichlet proposal

⋮

0.04 % param. 34 (rate multiplier) Dirichlet proposal
6.59 % param. 35 (topology and branch lengths) TBR
14.06 % param. 35 (topology and branch lengths) LOCAL

Acceptance rates for the moves in the "cold" chain of run 1:

With prob. Chain accepted changes to
33.30 % param. 1 (revmat) with Dirichlet proposal

⋮

19.13 % param. 34 (rate multiplier) Dirichlet proposal
17.40 % param. 35 (topology and branch lengths) TBR
29.76 % param. 35 (topology and branch lengths) LOCAL

Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Form of the time-series plots of parameter estimates

Continuous parameters (e.g., substitution rates)

- warm and fuzzy caterpillars

Acceptance rates of parameter updates

Continuous and discrete parameters (MrBayes, BEAST)

- rates should ideally fall in the ~20–70% range

Form of the marginal posterior probability densities

Continuous parameters (e.g., substitution rates)

- beware of porcupine roadkill!

qualitative
diagnostics

Autocorrelation time (ACT) of parameter samples

Effective sample size (ESS) of parameter samples

quantitative
diagnostics

Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Autocorrelation time (ACT) of parameter samples

The lag (number of cycles) it takes for autocorrelation in parameter values to break down.

The lag k autocorrelation ρ_k is the correlation every draw and its k^{th} lag:

$$\rho_k = \frac{\sum_{i=1}^{n-k} (x_i - \bar{x})(x_{i+k} - \bar{x})}{\sum_{i=1}^n (x_i - \bar{x})^2}$$

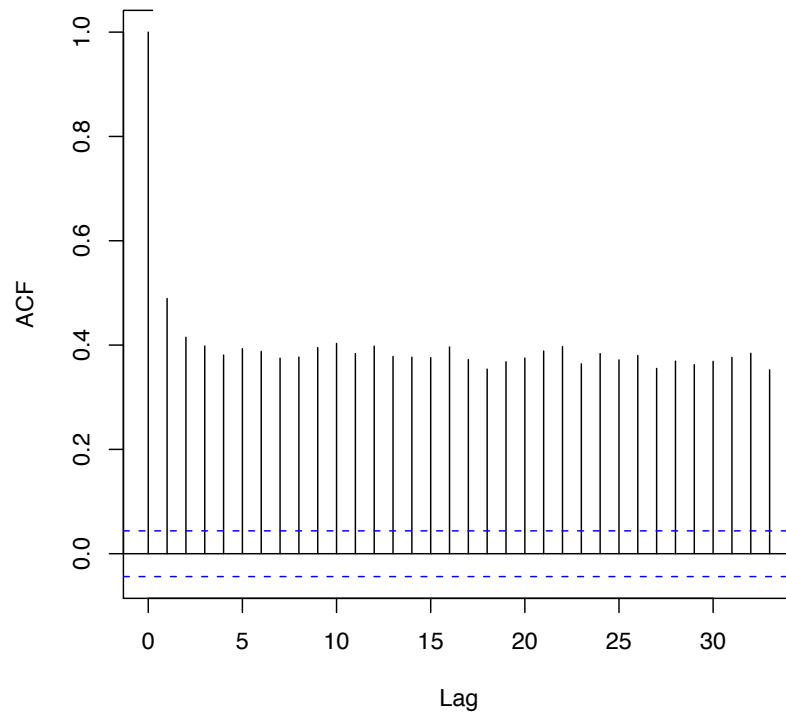
We would expect the k^{th} lag autocorrelation to be smaller as k increases (our 1st and 100th draws should be less correlated than our 1st and 2nd draws).

If autocorrelation is still relatively high for higher values of k , this indicates high degree of correlation between our draws and slow mixing.

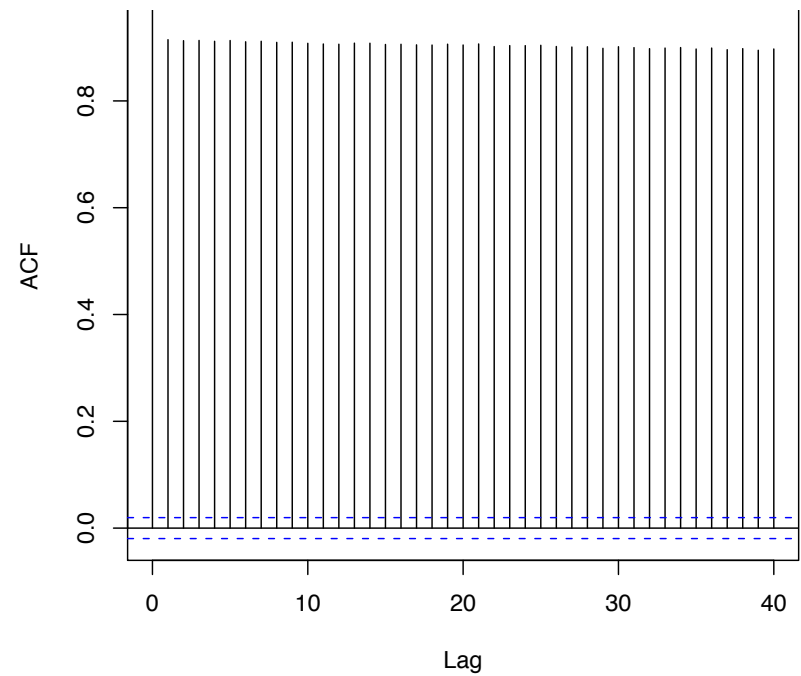
Assessing MCMC Performance: Based on Single Chains

Example: Autocorrelation time (ACT) of parameter samples

efficient mixing



slow mixing



Assessing MCMC Performance: Based on Single Chains

2. Mixing diagnostics

Effective Sample Size (ESS) diagnostic

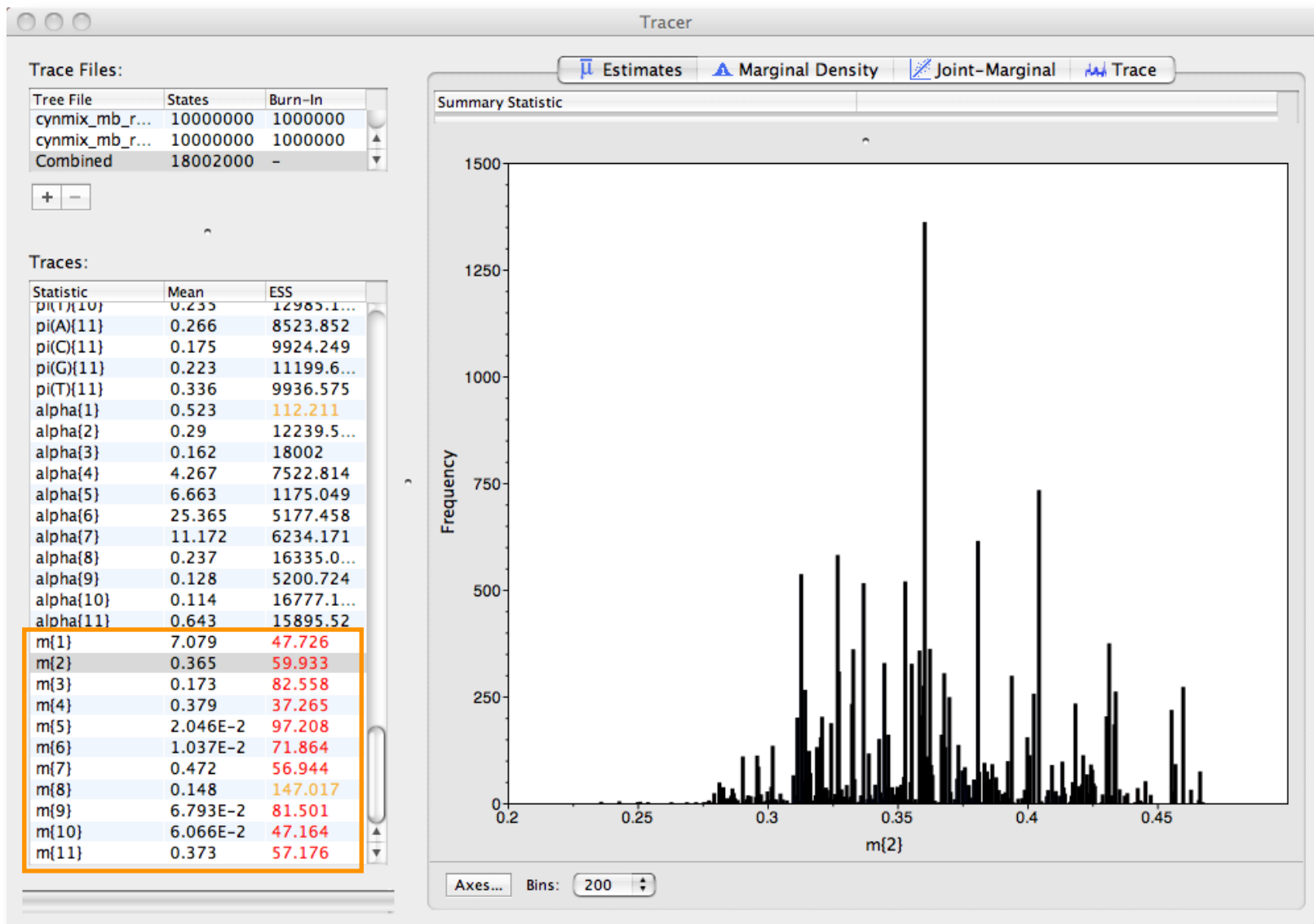
Continuous or discrete parameters

- number of samples/autocorrelation time (ACT)

Assessing MCMC Performance: Based on Single Chains

Example: ESS values for relative-rate multipliers from two RevBayes runs

poor mixing



Assessing MCMC Performance: Based on Single Chains

3. Sample-size diagnostics

Form of the marginal posterior probability densities

Continuous parameters

- brother of porcupine roadkill
- ensure SAE compliance!

Assessing MCMC Performance: Based on Single Chains

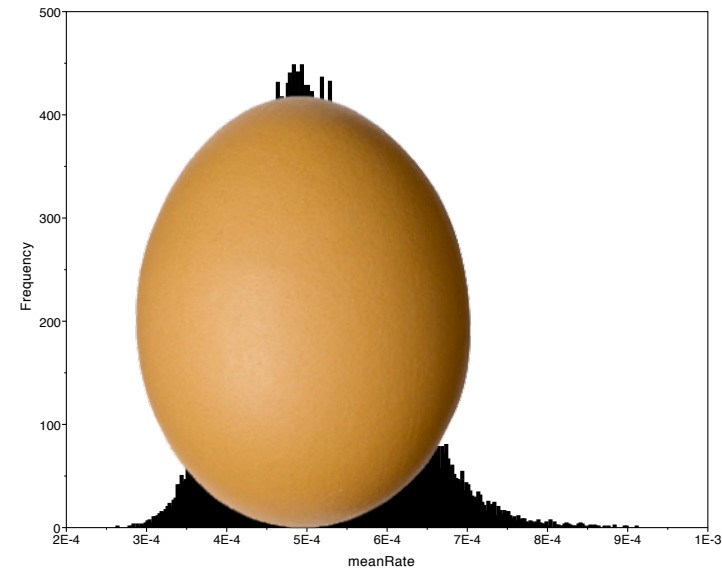
Example: Parameter estimates for mean-rate multipliers from BEAST runs

poor sampling



1M cycles

better sampling



40M cycles

inadequate chain length/poor mixing

all continuous parameters should be SAE

KDE SAE does not count (use histogram render)

Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm

Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains



Assessing MCMC Performance: Diagnostics Based on the Prior

Estimating under the prior...

Marginal posterior densities for parameters are updated versions of the corresponding prior probability densities: they are updated by the information in the data via the likelihood function.

$$\begin{array}{c} \text{posterior probability} \\ f(\tau, \nu, \Phi \mid \mathbf{X}) \end{array} = \frac{\begin{array}{c} \text{likelihood function} \\ f(\mathbf{X} \mid \tau, \nu, \Phi) \end{array} \begin{array}{c} \text{prior probability} \\ f(\tau, \nu, \Phi) \end{array}}{\begin{array}{c} f(\mathbf{X}) \\ \text{marginal likelihood} \end{array}}$$

Assessing MCMC Performance: Diagnostics Based on the Prior

Estimating under the prior...

Marginal posterior densities for parameters are updated versions of the corresponding prior probability densities: they are updated by the information in the data via the likelihood function.

We can compare the marginal prior densities to their posterior counterparts to help identify weak parameters.

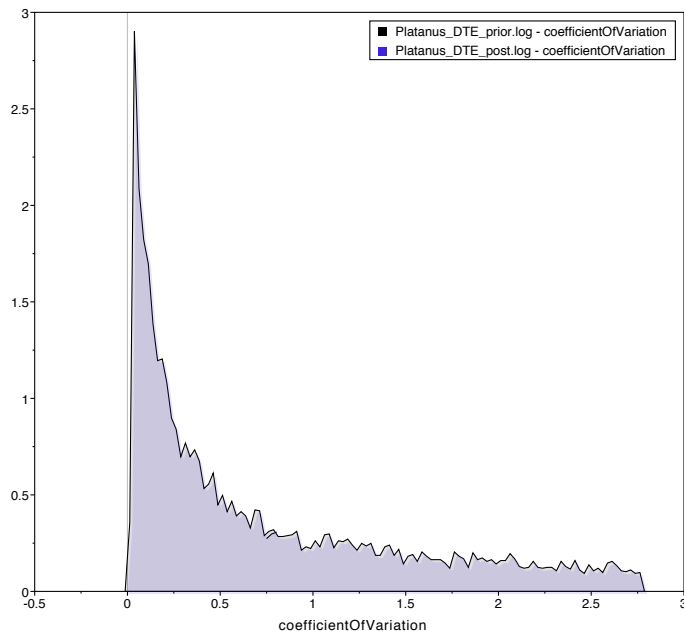
MCMC can be run to target the joint prior probability density either by estimating with no data or by forcing the likelihood function return 1.

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

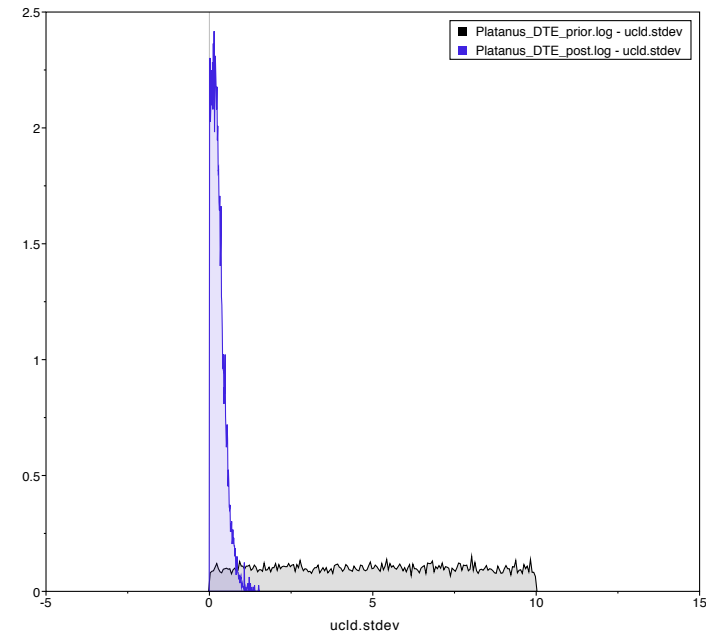
Assessing MCMC Performance: Diagnostics Based on the Prior

Does the marginal prior resemble the marginal posterior?

weak parameter



strong parameter



Strong departure of marginal prior and posterior is always good news

Similarity between the marginal prior and posterior may indicate:

- good news (you chose a great prior)
- bad news (no information in your data to estimate the parameter)

Outline

I. Bayesian inference, redux

The anatomy and meaning of Bayes theorem

II. Numerical algorithms for Bayesian inference

Markov-chain Monte Carlo (MCMC)

- Metropolis-Hastings algorithm
- Metropolis-Coupled algorithm

Summarizing posterior samples

III. Diagnosing MCMC performance

Motivation and overview of the basics

IV. MCMC Diagnostics

General strategies:

- diagnostics based on single chains
- diagnostics based on the prior
- diagnostics based on multiple, replicate chains



Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Compare estimates from multiple independent chains

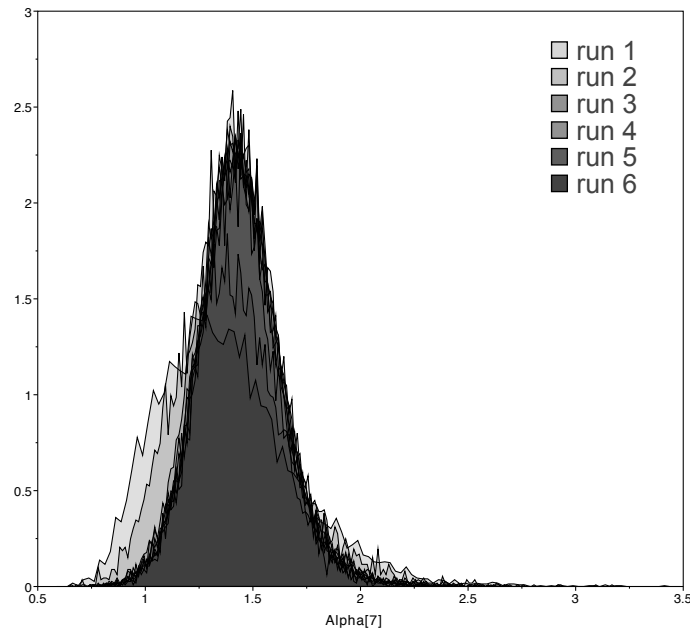
Form of the marginal posterior densities for all parameters

Continuous parameters

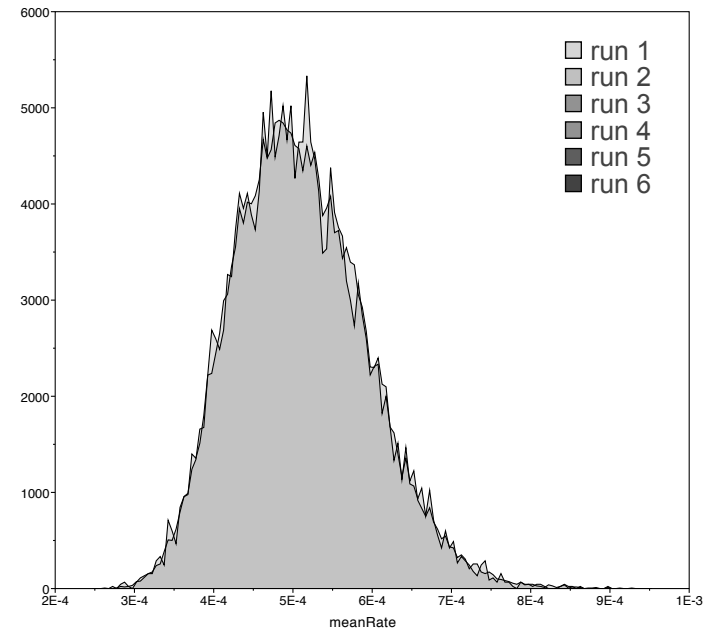
Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Example: Tracer plots of marginal densities from multiple RevBayes runs

bad convergence



better convergence



Parameter estimates from replicate independent MCMC analyses should be effectively identical.

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Compare estimates from multiple independent chains

Form of the marginal posterior densities for all parameters

Continuous parameters

PSRF (Gelman–Rubin) diagnostic

Continuous and discrete parameters

1. Run $m \geq 2$ chains of length $2c$ from overdispersed starting values.
2. Discard the first n draws of each chain.
3. Calculate the within-chain and between-chain variance.
4. Calculate the estimated variance of the parameter as a weighted sum of the within-chain and between-chain variance.
5. Calculate the PSRF.

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Example: PSRF values for relative-rate multipliers from two MrBayes runs

bad convergence			95% Cred. Interval		Median	PSRF *
Parameter	Mean	Variance	Lower	Upper		
TL{all}	4.921609	2.998138	2.836000	7.295000	5.056000	9.084
kappa{4,5}	3.095696	0.054125	2.667623	3.587024	3.085271	1.000
alpha{5}	1.006544	0.087721	0.606472	1.738482	0.950093	1.000
pinvar{1}	0.307396	0.009357	0.095913	0.471070	0.316173	1.000
m{1}	0.264226	0.009315	0.146502	0.421870	0.244468	5.507
m{2}	0.040919	0.000227	0.022205	0.065884	0.037425	5.279
m{3}	2.721453	7.157157	0.039001	5.544253	5.030560	69.564
m{4}	2.125810	3.568002	0.199137	4.044249	3.917338	150.012
m{5}	0.188768	0.004373	0.109303	0.295129	0.170624	5.749

better convergence			95% Cred. Interval		Median	PSRF *
Parameter	Mean	Variance	Lower	Upper		
TL{all}	0.073893	0.000034	0.063000	0.086000	0.074000	1.000
kappa{2,3}	3.236308	0.366904	2.199024	4.587719	3.190195	1.000
m{1}	1.285838	0.028345	0.980634	1.630387	1.278161	1.000
m{2}	1.423906	0.015507	1.182596	1.664627	1.423610	1.000
m{3}	0.589346	0.005341	0.453175	0.736459	0.587617	1.001

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Compare estimates from multiple independent chains

Form of the marginal posterior densities for all parameters

Continuous parameters

PSRF (Gelman–Rubin) diagnostic

Continuous and discrete parameters

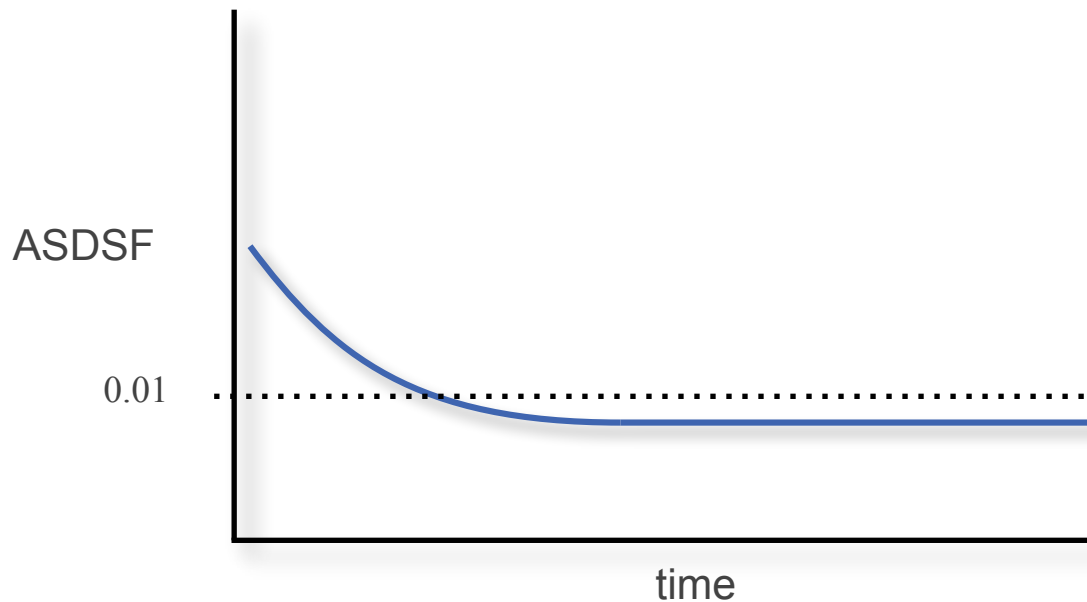
Comparing independent samples of trees

ASDSF: similarity of trees sampled by paired, independent chains

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Example: ASDSF

The overall similarity of the trees sampled by two independent, simultaneous MCMC analyses



Terminate chain when $\text{ASDSF} < 0.01$

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Compare estimates from multiple independent chains

Form of the marginal posterior densities for all parameters

Continuous parameters

PSRF (Gelman–Rubin) diagnostic

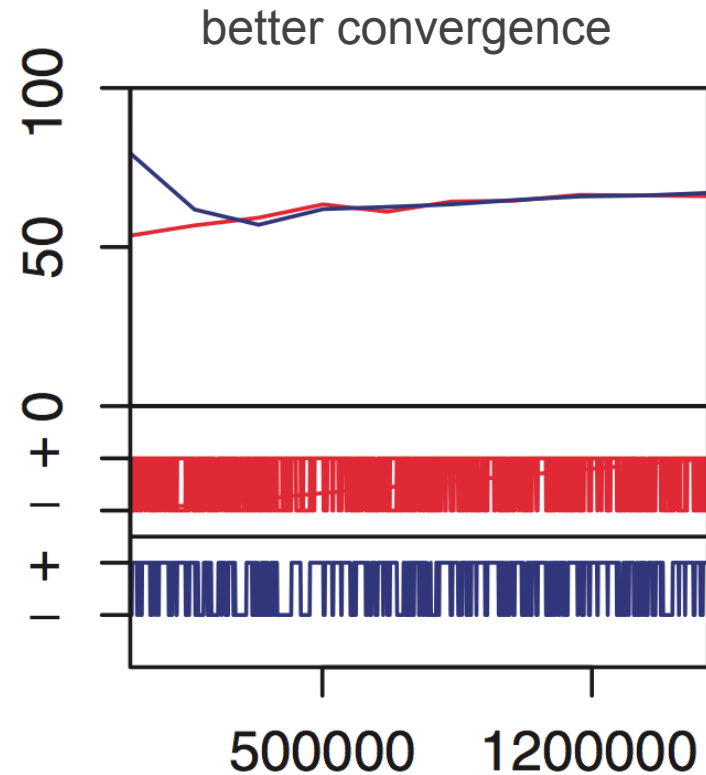
Continuous and discrete parameters

Comparing independent samples of trees

ASDSF: similarity of trees sampled by paired, independent chains

split frequencies & presence/absence plots

Example: split frequencies & presence/absence in AWTY



Track the frequency of a single node in trees sampled by two independent chains

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

Compare estimates from multiple independent chains

Form of the marginal posterior densities for all parameters

Continuous parameters

PSRF (Gelman–Rubin) diagnostic

Continuous and discrete parameters

Comparing independent samples of trees

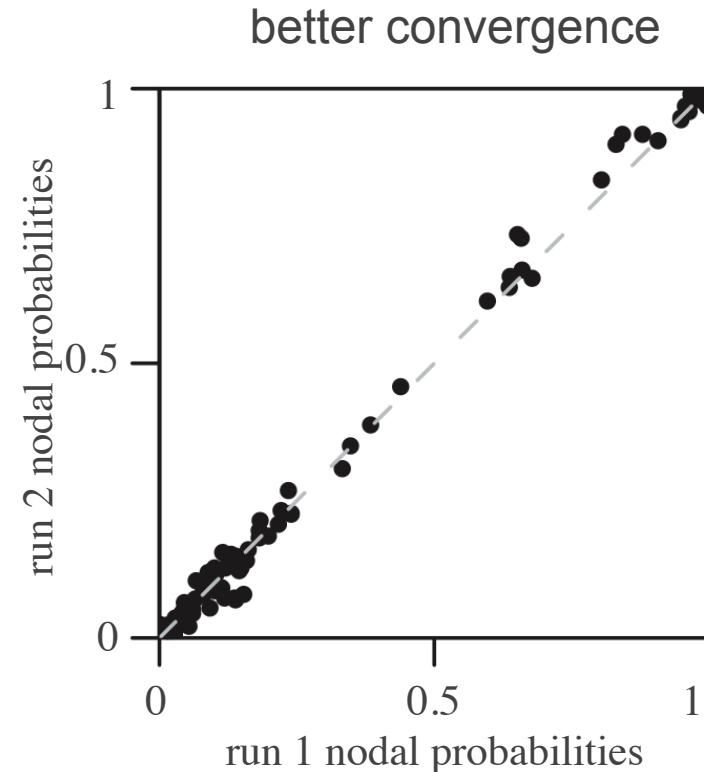
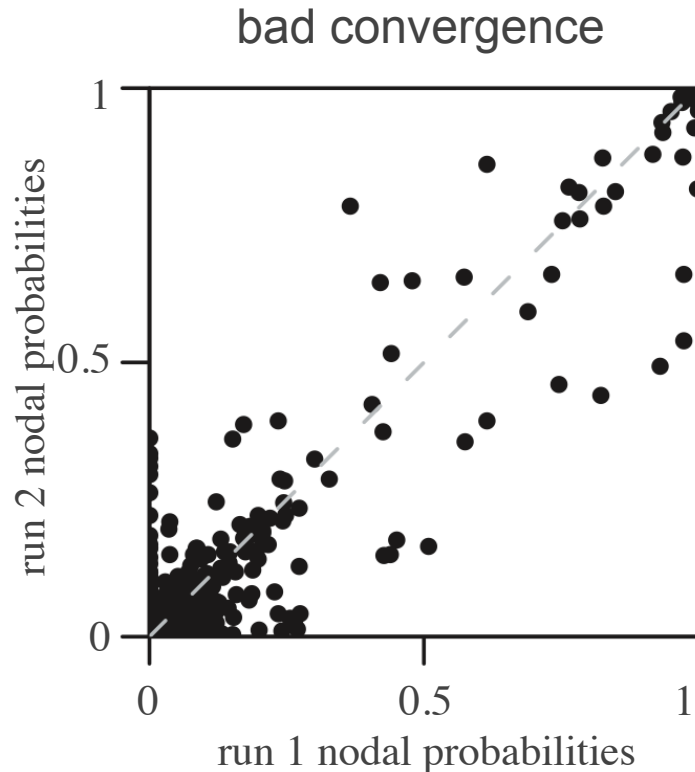
ASDSF: similarity of trees sampled by paired, independent chains

split frequencies & presence/absence plots

Compare-tree plots

Assessing MCMC Performance: Diagnostics Based on Multiple Runs

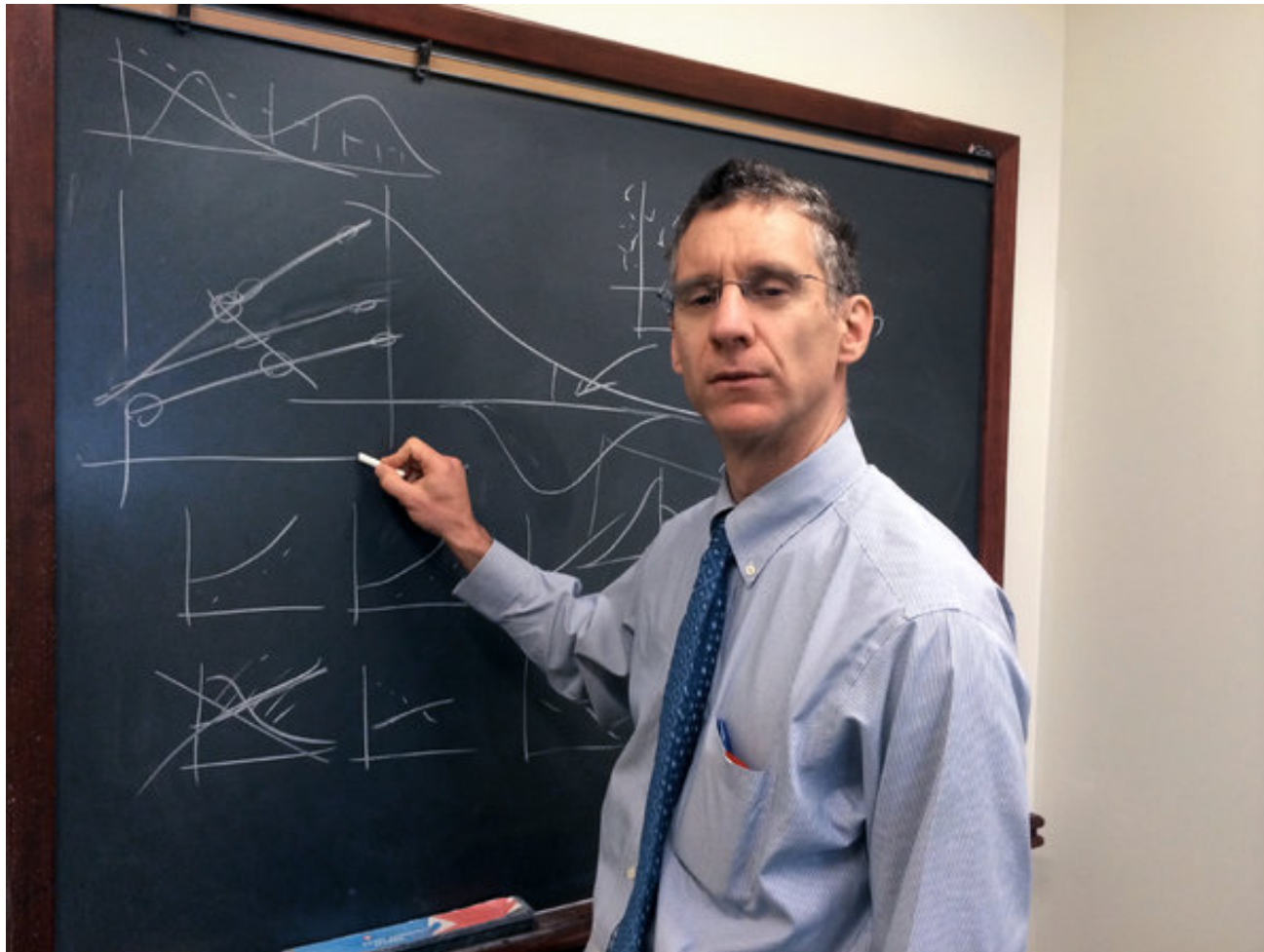
Example: 'comparetrees' plot of trees sampled by two MrBayes runs



Compare estimates of node probabilities estimated by two independent chains

Summary: Some General Strategies for Assessing MCMC Performance

“You can never be absolutely certain that the MCMC is reliable, you can only identify when something has gone wrong.” Andrew Gelman (hero)



Summary: Some General Strategies for Assessing MCMC Performance

1. When do you need to assess MCMC performance?

ALWAYS

2. When should you assess the performance of individual runs?

ALWAYS

3. Which diagnostics should you use to assess individual runs?

ALL that are relevant for the models/parameters you are estimating under

4. When is a single run sufficient to assess MCMC performance?

NEVER

5. When should you estimate under the prior?

WHENEVER POSSIBLE (and be wary of programs where it is not possible)

Summary: Some General Strategies for Assessing MCMC Performance

6. When should you use Metropolis-Coupling?

Whenever you cannot be certain that standard MCMC is adequate
i.e., **ALWAYS** (and be wary of programs where it is not possible)

7. When should you perform multiple independent MCMC runs?

ALWAYS (and be wary of pseudo-independence)

8. Which diagnostics should you use to assess multiple runs?

ALL that are relevant for the models/parameters you are estimating under

9. How many independent MCMC runs are sufficient?

AS MANY AS POSSIBLE (*i.e.*, as many as you think your data/problem deserve)

10. How long should you run each MCMC analysis?

AS LONG AS POSSIBLE (*i.e.*, as long as you think your data/problem deserve)

Assessing MCMC Performance: Software Tools

Software	Manual/visual	Split frequencies	PSRF	ESS	Geweke test	H-W test	S-Stationarity	M-Stationarity
AWTY	x	x	-	-	-	-	-	-
BOA	x	-	x	x	x	x	-	-
CODA	x	-	x	x	x	x	-	-
MrBayes	-	x	x	x	-	-	-	-
PhyloBayes	-	x	-	-	-	-	-	-
RevBayes	x	x	x	x	x	x	x	x
Tracer	x	-	-	x	-	-	-	-

Software tools are scattered across many programs

Diagnosis is largely manual/by visual inspection

Use of the methods is time consuming

Use of the methods is vague and virtual

Assessing MCMC Performance: Software Tools



BONSAI

Bayesian Output Needs Semi-Automated Inspection

Mike May <https://bitbucket.org/mrmay/bonsai/overview>

Semi-automated analysis using diverse diagnostic tools
Generates an automated report (sup. mat.)
Flags suspicious parameters
R package

Assessing MCMC Performance: Software Tools



```
install.packages('< path to bonsai >', repos=NULL, type='source', dependencies=TRUE)
library(bonsai)
```

```
# Start by naming our project.
```

```
project <- '< project name >'
```

```
# Run bonsai on a set of posterior and prior samples by specifying
# the directories directly to the log files.
```

```
posteriors <- c('< path to posterior log file 1 >',
                '< path to posterior log file 2 >',
                '< path to posterior log file 3 >',
                '< path to posterior log file 4 >')
```

```
priors <- c('< path to prior log file 1 >',
            '< path to prior log file 2 >',
            '< path to prior log file 3 >',
            '< path to prior log file 4 >')
```

```
path <- '< path to output diretory >'
```

```
# Make an object of class bonsai.
```

```
bonsai_object <- bonsai(project=project, path=path,
                        posterior.paths=posteriors,
                        prior.paths=priors)
```

```
# Then use the member function runBonsai to make a bonsai report.
bonsai_object$runBonsai()
```

```
# We can also point bonsai at a directory that contains a mix of
# posterior and prior log files without specifying the paths to
# each log file explicitly. As long as the path for each log file
# in the directory contains 'posterior' or 'prior', and all the
# log files have the same parameters, bonsai will collate all the
# log files automatically.
```

```
bonsai_object <- bonsai(project=project, path=path)
bonsai_object$runBonsai()
```

Assessing MCMC Performance: Software Tools



It summarizes issues for
the entire MCMC project.

BONSAI generates a report that highlights potential MCMC pathologies.

2 Posterior numerical parameters

2.1 Summary

There are a total of 15 parameters. They are: Posterior, Likelihood, Prior, TL, er[1], er[2], er[3], er[4], er[5], er[6], pi[1], pi[2], pi[3], pi[4], pinvar

2.2 Flags

2.2.1 Critical flags

- Run 2: Parameter pi[3] has critically low p-value for Geweke's diagnostic ($p = 0.002$)
- Parameters Likelihood and Posterior are strongly correlated ($\rho = 1$)
- Parameters TL and Prior are strongly correlated ($\rho = -1$)

2.2.2 Major flags

- Run 2: Parameter Prior has very low p-value for Geweke's diagnostic ($p = 0.023$)
- Run 2: Parameter TL has very low p-value for Geweke's diagnostic ($p = 0.023$)
- Parameters Prior and Posterior are correlated ($\rho = 0.292$)
- Parameters TL and Posterior are correlated ($\rho = -0.292$)
- Parameters pinvar and Posterior are correlated ($\rho = -0.268$)
- Parameters Prior and Likelihood are correlated ($\rho = 0.275$)
- Parameters TL and Likelihood are correlated ($\rho = 0.275$)

Assessing MCMC Performance: Software Tools



BONSAI generates a report that highlights potential MCMC pathologies.

2.3.6 `er[2]`

It also reports results for each individual parameter.

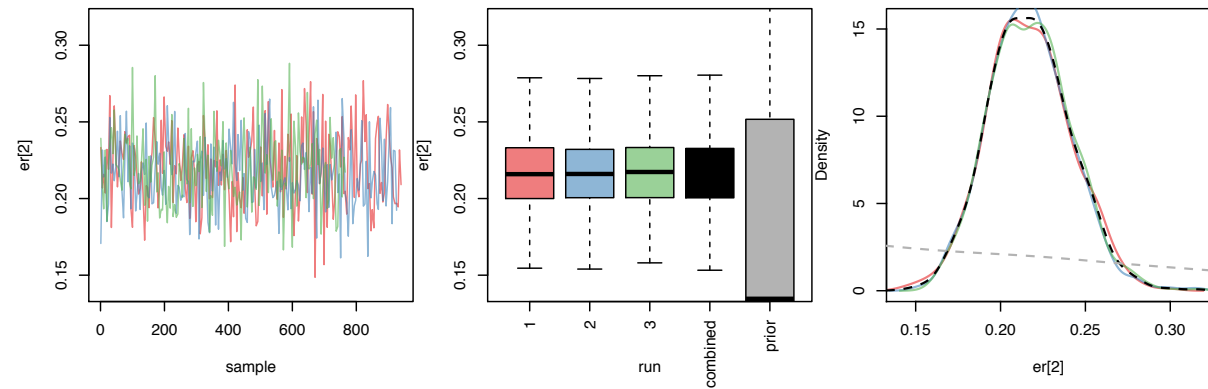


Figure 6: Parameter plots

	Mean	Lower 95% HPD	Upper 95% HPD	ESS	Geweke	KL
Run 1	0.22	0.17	0.27	890.73	0.26	1.62
Run 2	0.22	0.17	0.26	833.69	0.09	1.65
Run 3	0.22	0.17	0.27	558.97	0.42	1.65
Combined runs	0.22	0.17	0.27	2283.39		

Table 8: Parameter table

Major flags

- Parameters `er[2]` and `er[4]` are correlated ($\rho = -0.291$)
- Parameters `er[2]` and `er[5]` are correlated ($\rho = -0.317$)

Assessing MCMC Performance: Software Tools



BONSAI generates a report that highlights potential MCMC pathologies.

2.3.6 `er[2]`

It also reports results for each individual parameter.

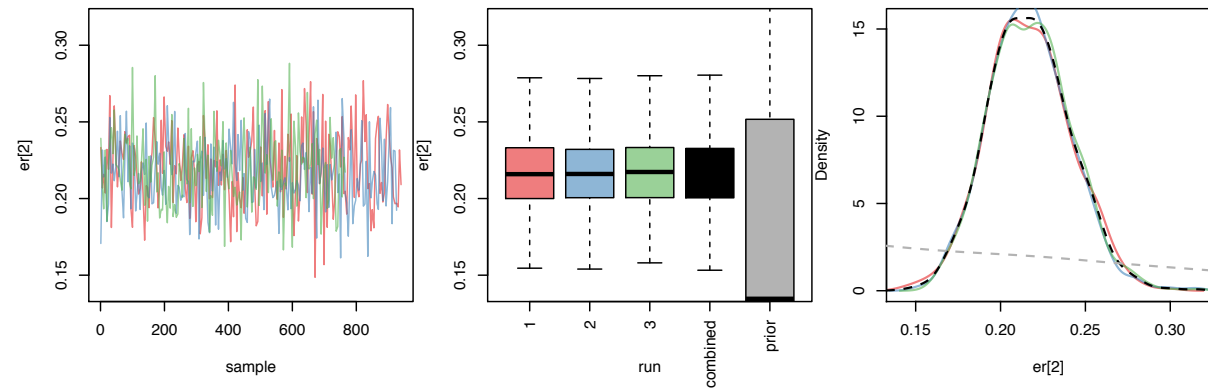


Figure 6: Parameter plots

summary stats

	Mean	Lower 95% HPD	Upper 95% HPD	ESS	Geweke	KL
Run 1	0.22	0.17	0.27	890.73	0.26	1.62
Run 2	0.22	0.17	0.26	833.69	0.09	1.65
Run 3	0.22	0.17	0.27	558.97	0.42	1.65
Combined runs	0.22	0.17	0.27	2283.39		

Table 8: Parameter table

Major flags

- Parameters `er[2]` and `er[4]` are correlated ($\rho = -0.291$)
- Parameters `er[2]` and `er[5]` are correlated ($\rho = -0.317$)

Assessing MCMC Performance: Software Tools



BONSAI generates a report that highlights potential MCMC pathologies.

2.3.6 `er[2]`

It also reports results for each individual parameter.

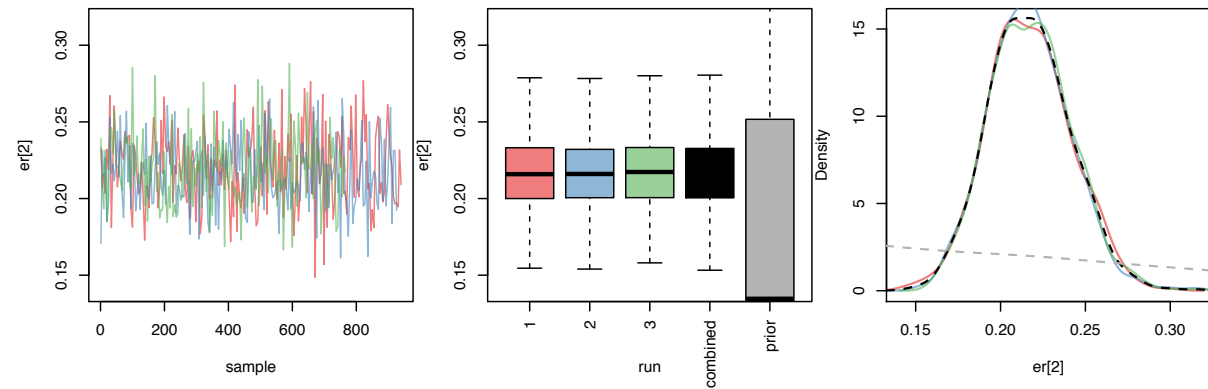


Figure 6: Parameter plots

	Mean	Lower 95% HPD	Upper 95% HPD	number of samples	ESS	Geweke	KL
Run 1	0.22	0.17	0.27		890.73	0.26	1.62
Run 2	0.22	0.17	0.26		833.69	0.09	1.65
Run 3	0.22	0.17	0.27		558.97	0.42	1.65
Combined runs	0.22	0.17	0.27		2283.39		

Table 8: Parameter table

Major flags

- Parameters `er[2]` and `er[4]` are correlated ($\rho = -0.291$)
- Parameters `er[2]` and `er[5]` are correlated ($\rho = -0.317$)

Assessing MCMC Performance: Software Tools



BONSAI generates a report that highlights potential MCMC pathologies.

2.3.6 `er[2]`

It also reports results for each individual parameter.

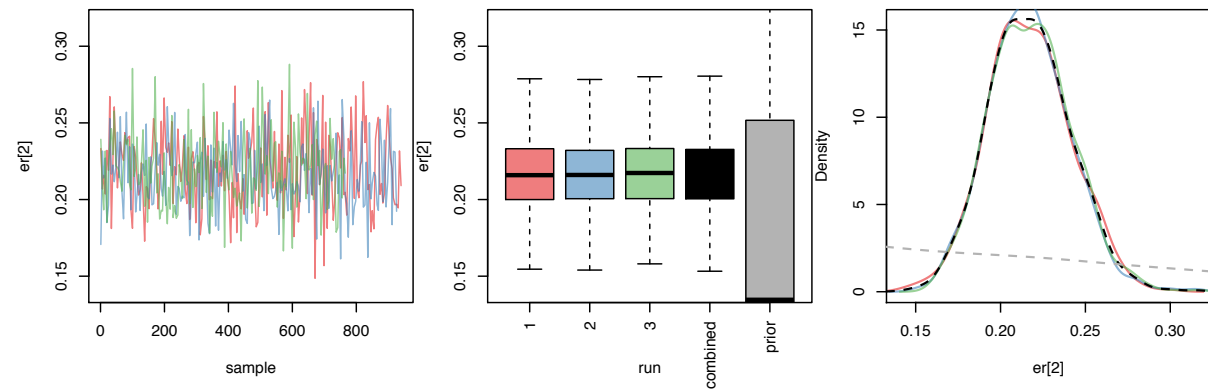


Figure 6: Parameter plots

convergence diagnostics

	Mean	Lower 95% HPD	Upper 95% HPD	ESS	Geweke	KL
Run 1	0.22	0.17	0.27	890.73	0.26	1.62
Run 2	0.22	0.17	0.26	833.69	0.09	1.65
Run 3	0.22	0.17	0.27	558.97	0.42	1.65
Combined runs	0.22	0.17	0.27	2283.39		

Table 8: Parameter table

Major flags

- Parameters `er[2]` and `er[4]` are correlated ($\rho = -0.291$)
- Parameters `er[2]` and `er[5]` are correlated ($\rho = -0.317$)

Assessing MCMC Performance: Software Tools



BONSAI generates a report that highlights potential MCMC pathologies.

2.3.6 `er[2]`

It also reports results for each individual parameter.

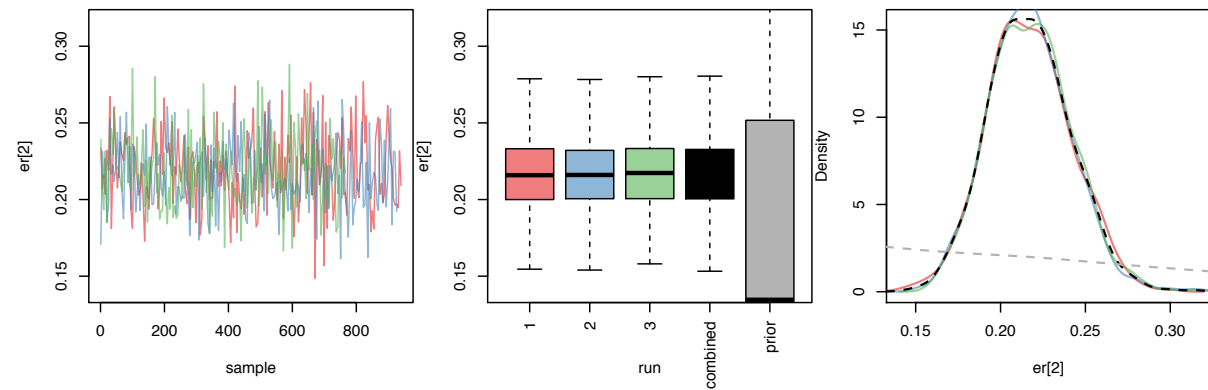


Figure 6: Parameter plots

	Mean	Lower 95% HPD	Upper 95% HPD	ESS	Geweke	KL
Run 1	0.22	0.17	0.27	890.73	0.26	1.62
Run 2	0.22	0.17	0.26	833.69	0.09	1.65
Run 3	0.22	0.17	0.27	558.97	0.42	1.65
Combined runs	0.22	0.17	0.27	2283.39		

Table 8: Parameter table

Major flags

- Parameters `er[2]` and `er[4]` are correlated ($\rho = -0.291$)
- Parameters `er[2]` and `er[5]` are correlated ($\rho = -0.317$)

Assessing MCMC Performance: Software Tools



BONSAI generates a report that highlights potential MCMC pathologies.

Additionally, it identifies correlations among parameters.

