# Bayesian methods for ecological and environmental modelling

Trainers:

Lindsay Flynn Banin, David Cameron, Pete Henrys & Peter Levy

Edinburgh

23-27 June 2025



UK Centre for Ecology & Hydrology

# Session 2b
# Markov Chain Monte Carlo (MCMC)

**David Cameron**

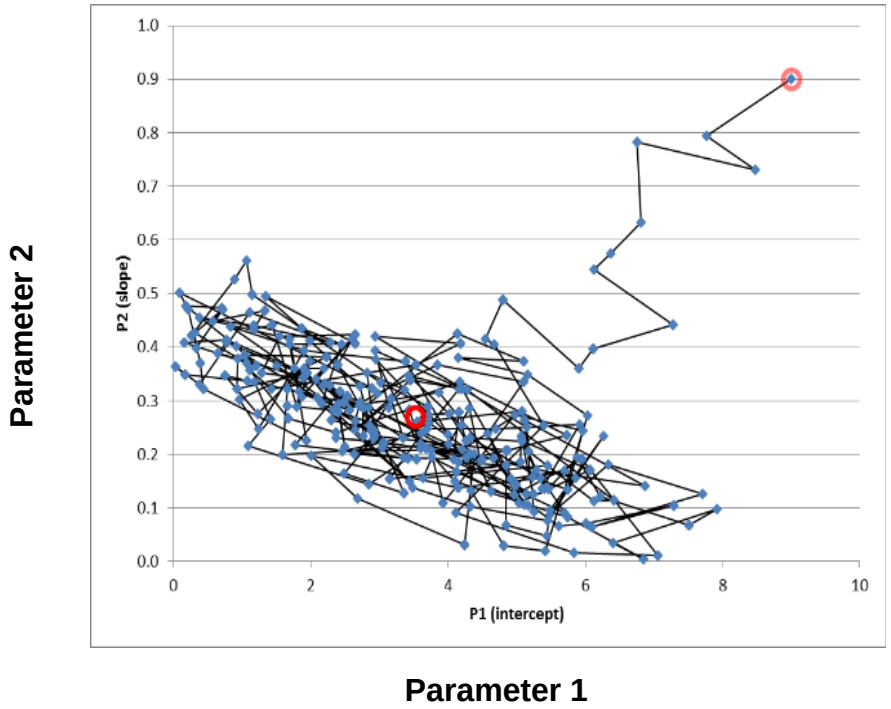
UK Centre for
Ecology & Hydrology

# How can we calculate the posterior P(θ|y) distribution?

We have: (1) a prior pdf P(θ) for our model's parameters, (2) new data y. We also know how to calculate the likelihood P(y|θ).

Bayes Theorem tells us that the posterior P(θ|y) is proportional to P(θ) times P(y|θ), but how do we calculate that?

Answer: we do not "calculate" a formula for P(θ|y), but we sample from it using "Markov Chain Monte Carlo" (MCMC)



# Plan

How to sample any probability distribution p(θ) (leaving inference/Bayes to one side for just now)

- Simple random sampling example (“Accept-Reject” using random (Monte Carlo) sampling)
- Why MCMC? - curse of dimensionality!
- What is MCMC and how does it work?
- Simplest MCMC algorithm (Metropolis)

# Plan

Application of MCMC to inference/Bayes

- Posterior is just another probability distribution $P(\theta|y)$ that can be sampled
- Posterior $\propto$ Prior x Likelihood (or $P(\theta|y) \propto P(\theta) \times P(y|\theta)$))
- Simple linear model example (Excel Spreadsheet)

# Plan

MCMC practicalities

- Convergence
- MCMC proposal functions
- What to do with the posterior MCMC sample/chain

# R code demonstrations: Sampling a probability p(θ) distribution

On the following slides I will demonstrate sampling from probability distributions using R

We will be switching between this presentation and the webpage https://nerc-ceh.github.io/beem/ae/ae-2b-mcmc.html which contains the R-code. You may want to cut and paste the code into an R session or just watch

At the end of the webpage there are also some optional examples which are provided as extra R code examples of MCMC for your interest

No practical session however, since MCMC is our main method of sampling posterior distributions there are many practicals/examples of MCMC (e.g. process-based models practical) throughout the course

# Example of Monte Carlo sampling that is not MCMC

- Θ is the unit square (so sampling space is 2-dimensional)
- p(θ) = c > 0 within a distance 1 from the origin, and 0 outside
- MC: Generate 5000 candidates uniformly within the unit square, and then discard the ones too far from the origin (Accept – Reject algorithm)

**R-code**:

```
n          <- 5000
xproposed <- cbind( runif(n,0,1), runif(n,0,1) )
iaccept    <- xproposed[,1]^2 + xproposed[,2]^2 < 1
x          <- xproposed[iaccept,]
```

Problems with such non-MCMC sampling methods:

- Inefficient if sampling space is high-dimensional

# The curse of dimensionality

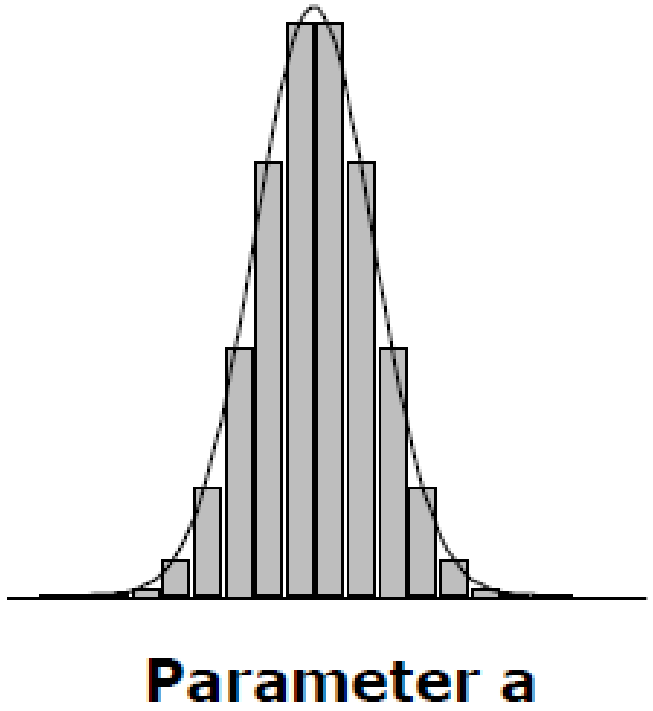
Sampling probability p(a) where a has one dimension

A sample size of 20 is sufficiently large to calculate:

- **Quantiles**
- **Mean etc**

However:

If “a” has many dimensions (n) (e.g. 10 model parameters) with 20 points per dimension, you have a sample size of $20^n$ or $20^{10}$!



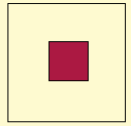
# How to address the curse of dimensionality?

Large areas are likely to be unimportant - particularly for high dimensional space

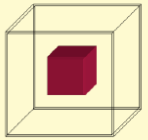
Space needing sampled generally proportionally smaller as we increase dimensions. Can we make use of this?

Need a method that ‘zones in’ on the important part of space and samples eg MCMC

- 1d: 1/3



- 2d: 1/9



- 3d: 1/27
- 1/3^n → most of the space is empty in high dimensions

# What is MCMC?

**MCMC** = Markov Chain Monte Carlo sampling

**Markov process:** random walk where the jump from θ(i) to θ(i+1) is independent of θ(0) .. θ(i-1) i.e. next value θ(i+1) depends ONLY on the last value θ(i) according to some transition probability

$p(\theta(i) \rightarrow \theta(i+1))$

**Markov chain**: chain of values of a Markov process. This is our sampling of the probability distribution p(θ).

# What is MCMC?

**MCMC** = Markov Chain Monte Carlo sampling

Method for sampling from a probability distribution:

- A function $p(\theta)$ on a sampling space $\Theta = \{\theta\}$
- $p(\theta) \geq 0$
- $\int p(\theta)d\theta = 1$

Important application:

- $\Theta$ is the parameter space of a model
- $p(\theta)$ represents uncertainty about parameter values

MCMC is a subset of Monte Carlo (MC) sampling

- There are direct and iterative MC sampling methods
- MCMC is iterative: it builds up the sample one-by-one

# How does MCMC work?

We build our sample from p(θ) step-by-step.

Pseudo-code for all MCMC algorithms:

- Start anywhere in sampling space Θ
- Random walk through Θ in such a way that the set of visited points is the desired representative sample

# What do you need to know to run a MCMC?

Fortunately we only need to know the ratio of the probability of where we are in Θ say $\theta_i$ and where we propose to go next θ' (proposal or candidate point)

**Required:**

- For all $\{\theta_i, \theta'\}$: be able to calculate ratio $\alpha = p(\theta')/p(\theta_i)$

**Terminology**: $\alpha$ also known as "transition probability"

Time for a quick analogy - visiting countries in geographical space i.e. a two dimensional Θ

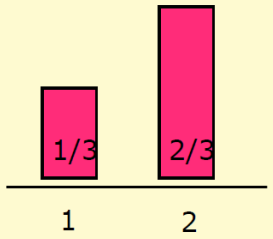
# Concept of MCMC by analogy

A politician wants to visit all European countries

To be efficient, they want to spend time in each country proportional to its population size

How should they decide whether to spend more time in the country they are currently in or move onto the next country?

- Imagine two countries
  - P(1) = 500.000 ; P(2) =1.000.000
- Try:
  - p(1 → 2) = 1
  - p(2 → 1) = P(1) / P(2) = 0.5




# How does MCMC work?

THE JOURNAL OF CHEMICAL PHYSICS VOLUME 21, NUMBER 6 JUNE, 1953

## Equation of State Calculations by Fast Computing Machines

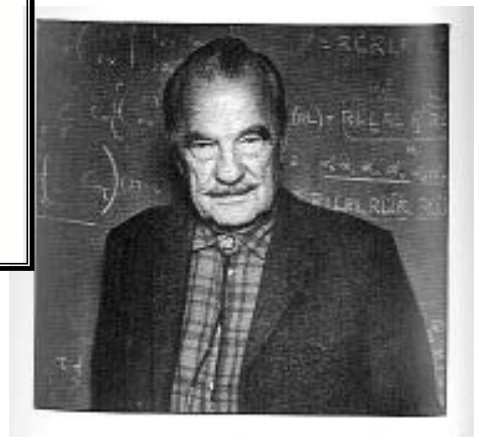
NICHOLAS METROPOLIS, ARIANNA W. ROSENBLUTH, MARSHALL N. ROSENBLUTH, AND AUGUSTA H. TELLER,
*Los Alamos Scientific Laboratory, Los Alamos, New Mexico*

AND

EDWARD TELLER,* *Department of Physics, University of Chicago, Chicago, Illinois*

(Received March 6, 1953)

A general method, suitable for fast computing machines, for investigating such properties as equations of state for substances consisting of interacting individual molecules is described. The method consists of a modified Monte Carlo integration over configuration space. Results for the two-dimensional rigid-sphere system have been obtained on the Los Alamos MANIAC and are presented here. These results are compared to the free volume equation of state and to a four-term virial coefficient expansion.



5. If $\alpha < 1$, accept the candidate with probability $\alpha$,
6. Otherwise reject the candidate: $\theta^{(i+1)} \leftarrow \theta^{(i)}$
7. Go to step 2 unless the desired sample size is reached

# Application of MCMC to inference/ Bayes

Specific application: sampling the posterior probability distribution $P(\theta|y)$

Just like sampling any other probability distribution using MCMC

Use the same MCMC Metropolis algorithm

Remember Posterior $\propto$ Prior x Likelihood (or $P(\theta|y) \propto P(\theta) \times P(y|\theta)$)

# Application of MCMC to inference/ Bayes

So at each MCMC step:

Calculate Prior $P(\theta)$ and Likelihood $P(y|\theta)$ at candidate point and form Metropolis ratio

Metropolis ratio = $(P(\theta)' \times P(y|\theta)') / (P(\theta)_i \times P(y|\theta)_i)$

where $(\ldots)'$ = candidate point and $(\ldots)_i$ = current point

# Application of MCMC to inference/ Bayes

So at each MCMC step:

Calculate prior $P(\theta)$ and likelihood $P(y|\theta)$ at candidate point and form Metropolis ratio

Metropolis ratio = $(P(\theta)' \times P(y|\theta)') / (P(\theta)_i \times P(y|\theta)_i)$
where $(\ldots)'$ = candidate point and $(\ldots)_i$ = current point

**Important:** For practical (numerical rounding) reasons we often take logarithms hence:

Metropolis ratio = $[ \log (P(\theta)') + \log(P(y|\theta)') ] - [ \log(P(\theta)_i) + \log(P(y|\theta)_i) ]$

Or in R code (0 = current point, 1 = candidate point -- see optional examples)

```
logalpha <- (logPrior1 + logL1) – (logPrior0 + logL0)
```

# Example: Using MCMC to do Bayesian linear regression

- **Say we have a model $y = \alpha + \beta * x$, but we do not know the appropriate values of $\theta = \{\alpha, \beta\}$**
  - **Prior:** Uncertainty about parameters θ, expressed as P(θ) = { Uniform[0,10], Uniform[0,1] }
- **Say we then get some data: y(x=10) = 6.09 ± 1.83, y(x=20) = 8.81 ± 2.64, y(x=30) = 10.66 ± 3.27**
  - **Likelihood:** Uncertainty about measurement error expressed as Gaussian distributions with mean μ=0 and standard deviations $\sigma_i$=1.83, 2.64, 3.27

Peter Levy will be saying much more about Linear models later in the course

# A demonstration of MCMC using Excel

I am now going to show you an **Excel spreadsheet** that demonstrates the Bayesian inference of slope and intercept of a linear model using MCMC

The link for the spreadsheet

https://nerc-ceh.github.io/beem/ae/ae-2b-excel.xls

You may want to download and follow along or just watch

Having a ‘play’ with this spreadsheet now or later will give you a good practical understanding of how MCMC works

If there is an issue with running macros in your version of Excel. I’ve created an alternative you can also makes changes in and rerun.

https://nerc-ceh.github.io/beem/ae/ae-2b-mcmc-linear.html

# Convergence

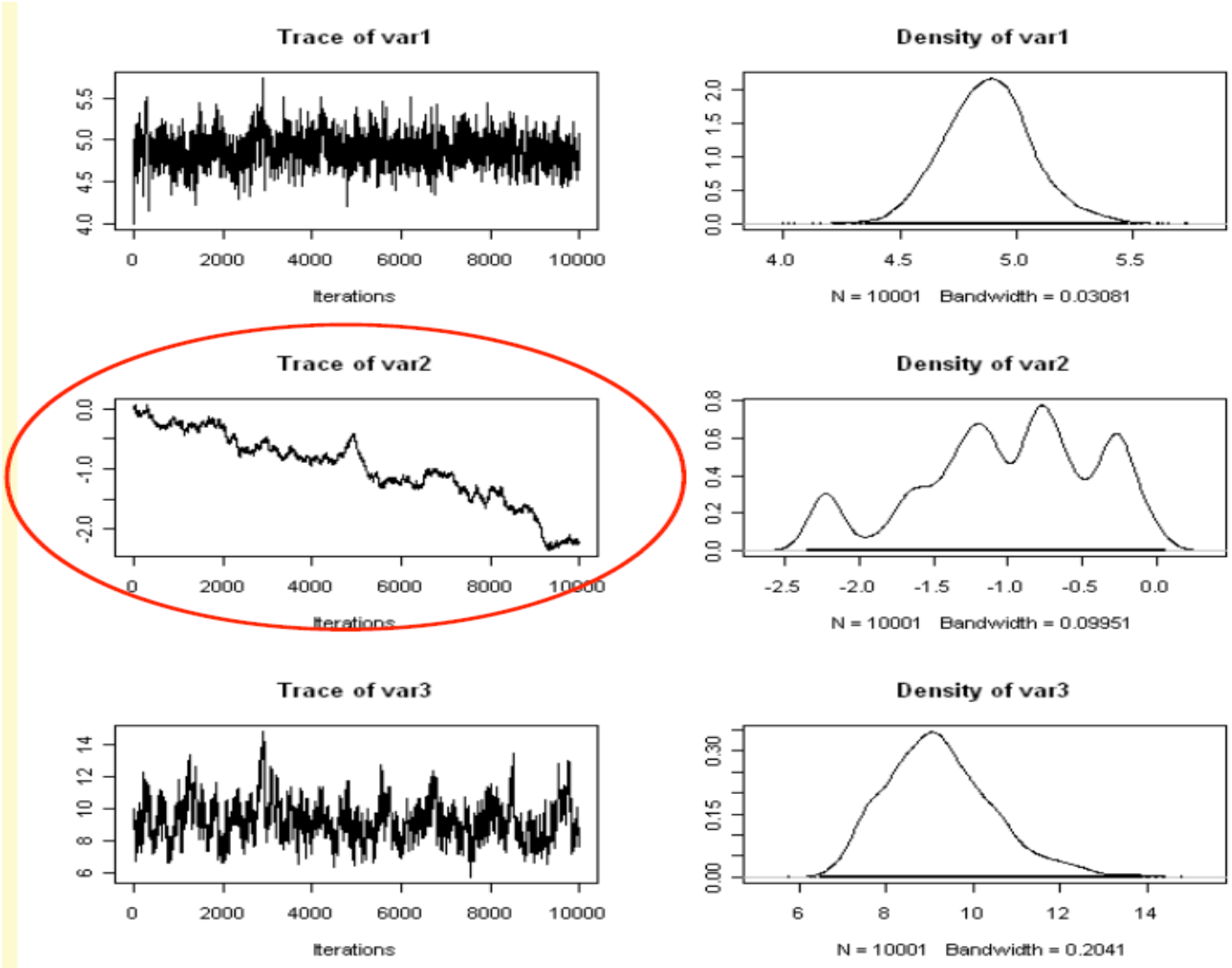
Convergence = representative sample using MCMC of the target posterior distribution

If done correctly, MCMC will always converge to a posterior distribution for $n \rightarrow inf$

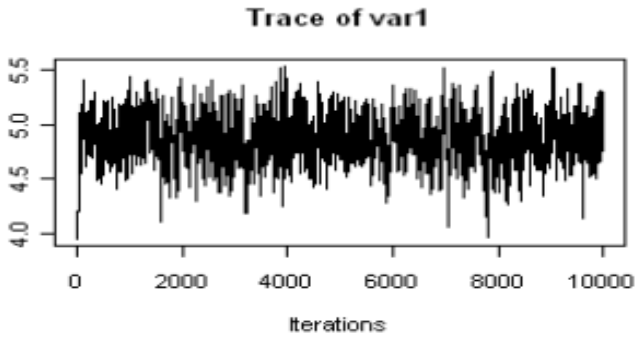
- Provided it can get anywhere in parameter space
- Better to use proposal functions with wide/ unlimited support such as Gaussian or exponential

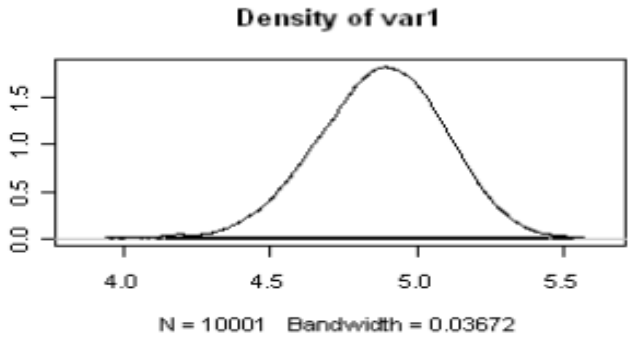
Question: How can we know the chain has converged?

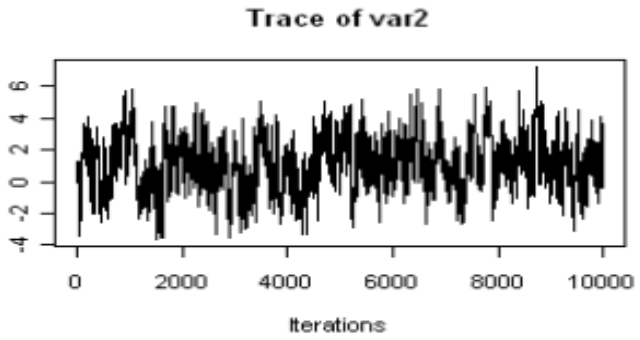
# MCMC chain trace plots showing lack of convergence

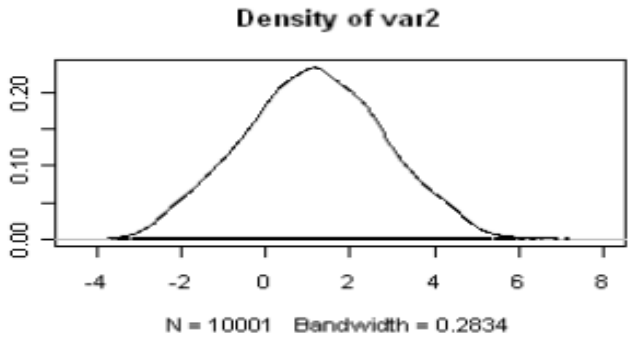


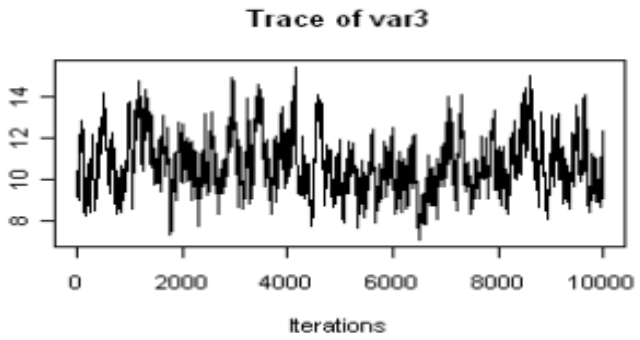
# MCMC chain trace plots showing convergence

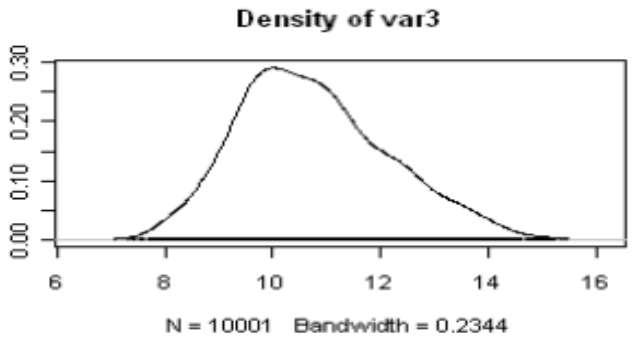


# Answer

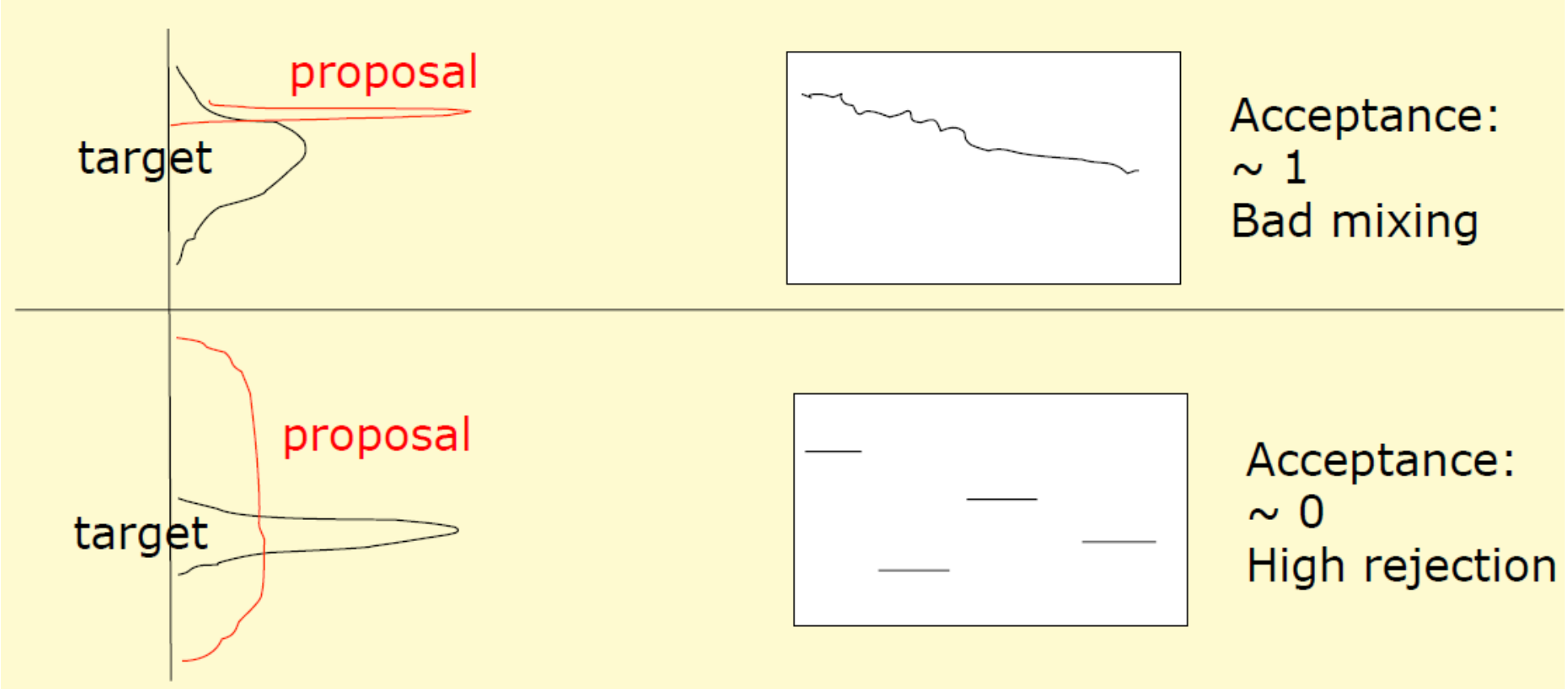
You can often see convergence by eye, but:

It is better to use a convergence test such as Gelman-Rubin

- compares variance within chains and between multiple chains that start in a different part of the space
- Gelman-Rubin should be close to 1 (R-hat)

# How proposal functions can influence speed of convergence

When an MCMC chain is slow to converge we say that is has ‘bad mixing’



# What to do with the posterior sample?

Remove the burn-in:

- part of the chain determined by initial conditions
- ensure starting point is unimportant
- can be up to about half of the chain

Use as a new prior when new calibration data is available

Posterior model predictions:

- sample from posterior chain (eg every $10^{th}$ point)
- run the model with the sample

Don’t summarise MCMC chain if not necessary

NEVER use summary statistics (e.g. stdev) for making model predictions (you will lose higher order correlations!)

# Recap/ Summary/ What we covered in this session

MCMC is an efficient way to sample a probability distribution

Simplest algorithm Metropolis

Can use it to sample $P(\theta|y) \propto P(\theta) P(y|\theta)$ (Prior x Likelihood)

Steps:

- Choose prior and likelihood distributions
- Sample with MCMC
- Check convergence (eg Gelman Rubin)
- Rerun with longer chain if not converged
- Remove burn-in subsample posterior chain
- Create model predictions by rerunning model with subsample

**Never** use summary statistics of chain to make predictions

# Any Questions?

# Thank You!

# Session 2b
# Markov Chain Monte Carlo (MCMC)

**David Cameron**

UK Centre for
Ecology & Hydrology

# Bayesian methods for ecological and environmental modelling

Trainers:

Lindsay Flynn Banin, David Cameron, Pete Henrys & Peter Levy

Edinburgh

23-27 June 2024

UK Centre for Ecology & Hydrology

# Example of MCMC - Metropolis

Same example as before:

- Θ is the unit square
- p(θ) = c > 0 within a distance 1 from the origin, and 0 outside

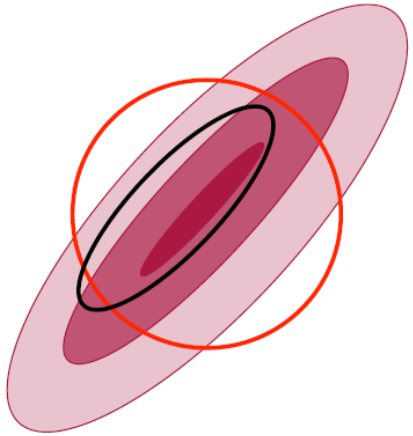
R-code:

```
pratio <- function(xa,xb) { (xa[1]^2 + xa[2]^2 < 1)  &&
                                (xa[1]>=0) && (xa[2]>=0) }
n  <- 500 ; x <- vector("list",n) ; x[[1]] <- c(0,0)
for (i in 2:n) {
   xproposed <- x[[i-1]] + runif(2,-1,1)
   if ( runif(1,0,1) < pratio( xproposed, x[[i-1]]) )
     x[[i]] <- xproposed
   else
     x[[i]] <- x[[i-1]]
   end }
```

**Proposal function: runif(2,-1,1)**

# Importance of the proposal function

Important to have a proposal function that “fits” to the target distribution



- Change proposal width
- Change functional form
- Change correlation structure

Rule of thumb: 20-30% acceptance is good

But possible to tune more specifically

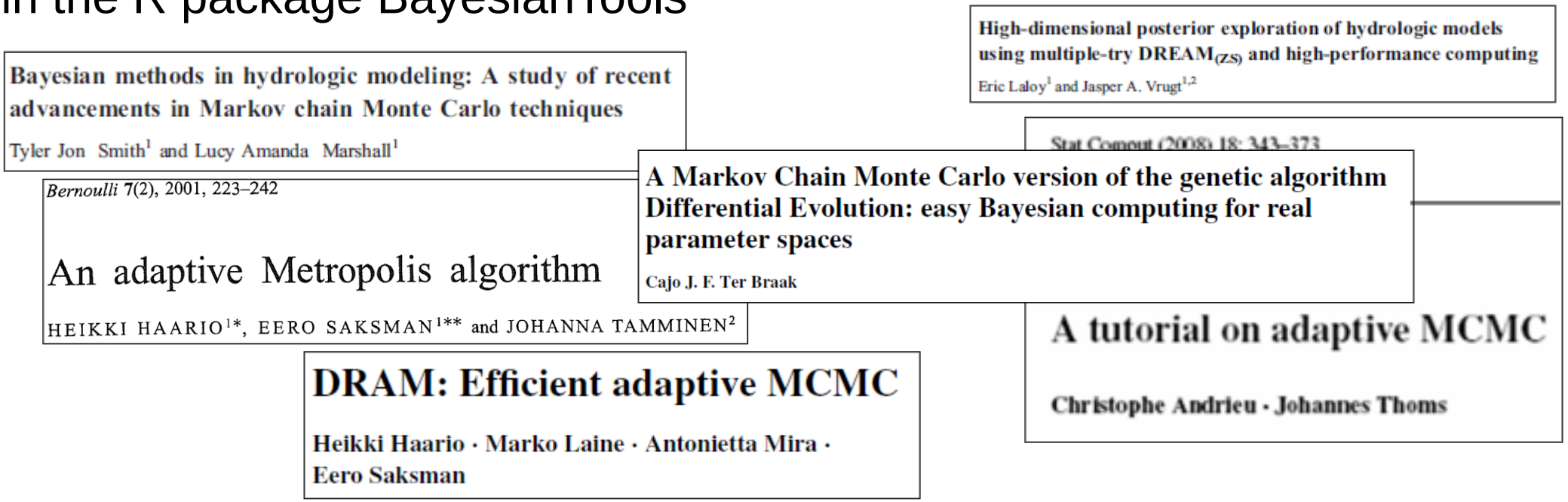
- eg dependent on the parameter area
- Must not change after burn-in

# Alternatives to Metropolis algorithm?

Metropolis is the simplest but many other algorithms exist in the literature and as R packages

For a particular Bayesian calibration an alternative could be significantly more efficient

We will be using other algorithms in this course. For example the ‘DEzs’ algorithm in the R package BayesianTools



# FAQ – Bayesian Calibration & MCMC


HOME ABOUT BROWSE SEARCH LATEST ADDITIONS POLICIES STATISTICS HELP CONTACT US ORCID
STAFF LOGIN

Bayesian Calibration (BC) and Bayesian Model Comparison (BMC) of process-based models: Theory, implementation and guidelines

**Van Oijen, Marcel**. 2008 *Bayesian Calibration (BC) and Bayesian Model Comparison (BMC) of process-based models: Theory, implementation and guidelines.* NERC/Centre for Ecology & Hydrology, 16pp.

Before downloading, please read NORA policies.

Text
BC&BMC_Guidance_2008-12-18_Final.pdf
Download (222kB)


https://nora.nerc.ac.uk/id/eprint/6087/

See example FAQ questions on the next two slides

# FAQ – Bayesian Calibration & MCMC example questions in document 1

- *Which BC algorithm should I use?*

- *Where can I read more about the BC methods?*

- *Can BC handle correlations between parameters?*

- *What do we do if measurement errors are not independent?*

- *Does it matter whether we use all data at the same time or in steps?*

- *The example codes are in Matlab and R: what if I work in a different programming language?*

# FAQ – Bayesian Calibration & MCMC example questions in document 2

- *What distribution should I use for the prior pdf?*

- *What is burn-in and do I need it?*

- *How do I make the proposal algorithm of my MCMC method more efficient?*

- *Is there any point in using more than one chain in MCMC?*

- *What do I do if the MCMC crashes?*

- *What do I do if the MCMC does not converge?*

- *What do I do if the sample from the posterior parameter pdf looks wrong, or if the posterior predictions of model output variables look wrong?*