Markov Chain Monte Carlo and using Eryn sampler

LISA Analysis Tools Workshop
15-18 April 2024
16/04/2024

Argyro Sasli
Aristotle University of Thessaloniki
asasli@auth.gr

LISA CONSORTIUM

ARISTOTLE UNIVERSITY OF THESSALONIKI

Introduction

Eryn: A multi-purpose sampler for Bayesian inference

paper: N. Karnesis et al (2023)
GitHub page: https://github.com/mikekatz04/Eryn
zenodo
pip installable!

- Advanced **Markov Chain Monte Carlo** (MCMC) sampler
- **Parallel Tempering**
- **Multiple model types**
- Unknown counts within multiple model types using Reversible Jump MCMC techniques (RJMCMC)

tomorrow by N. Karnesis

Markov Chain Monte Carlo (MCMC)

MCMC Techniques

- **Monte Carlo:** A technique that uses random numbers and probability to solve complex problems.

- **Markov Chain:** A process where the next state depends only on the current state (a "state" refers to the assignment of values to the parameters). It is based on the idea of creating a chain of points of the parameter space, using a combination of (i) random walk and (ii) selection of points based on their relative probability.

- The goal of the MCMC is to assess the *posterior* probability on parameters θ from a model $\mathbb{M}$, given some data y.

prior

posterior

$$p(\vec{\theta}|y) = \frac{p(y|\vec{\theta},\mathcal{M})\,p(\vec{\theta},\mathcal{M})}{p(y|\mathcal{M})}$$

Likelihood

Evidence

Markov Chain Monte Carlo (MCMC)

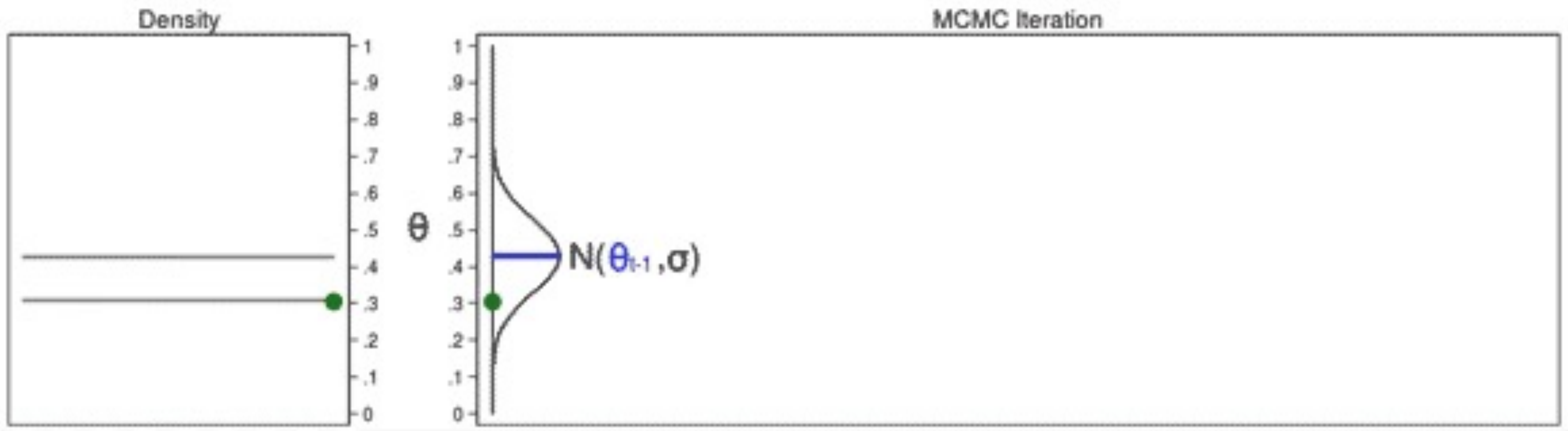
Performing MCMC (Metropolis-Hastings algorithm)

1. Define the problem
2. Define the likelihood function
3. Start with one set of parameters θ_t
4. Propose a new based on a proposal distribution (θ_{t+1})
5. Accept it based on a probability.
6. Repeat steps 4 and 5

Markov Chain Monte Carlo (MCMC)

Performing MCMC (Metropolis-Hastings algorithm)

1. Define the problem
2. Define the likelihood function
3. Start with one set of parameters θ_t
4. Propose a new based on a proposal distribution (θ_{t+1})
5. Accept it based on a probability.
6. Repeat steps 4 and 5



Step 1: $r(\theta_{new}, \theta_{t-1}) = \dfrac{\text{Posterior}(\theta_{new})}{\text{Posterior}(\theta_{t-1})} = \dfrac{\text{Beta}(1,1,0.306) \times \text{Binomial}(10,4,0.306)}{\text{Beta}(1,1,0.429) \times \text{Binomial}(10,4,0.429)} = 0.834$

Step 2: Acceptance probability $\alpha(\theta_{new}, \theta_{t-1}) = \min\{r(\theta_{new}, \theta_{t-1}), 1\} = \min\{0.834, 1\} = 0.834$

Step 3: Draw $u \sim \text{Uniform}(0,1) = 0.617$

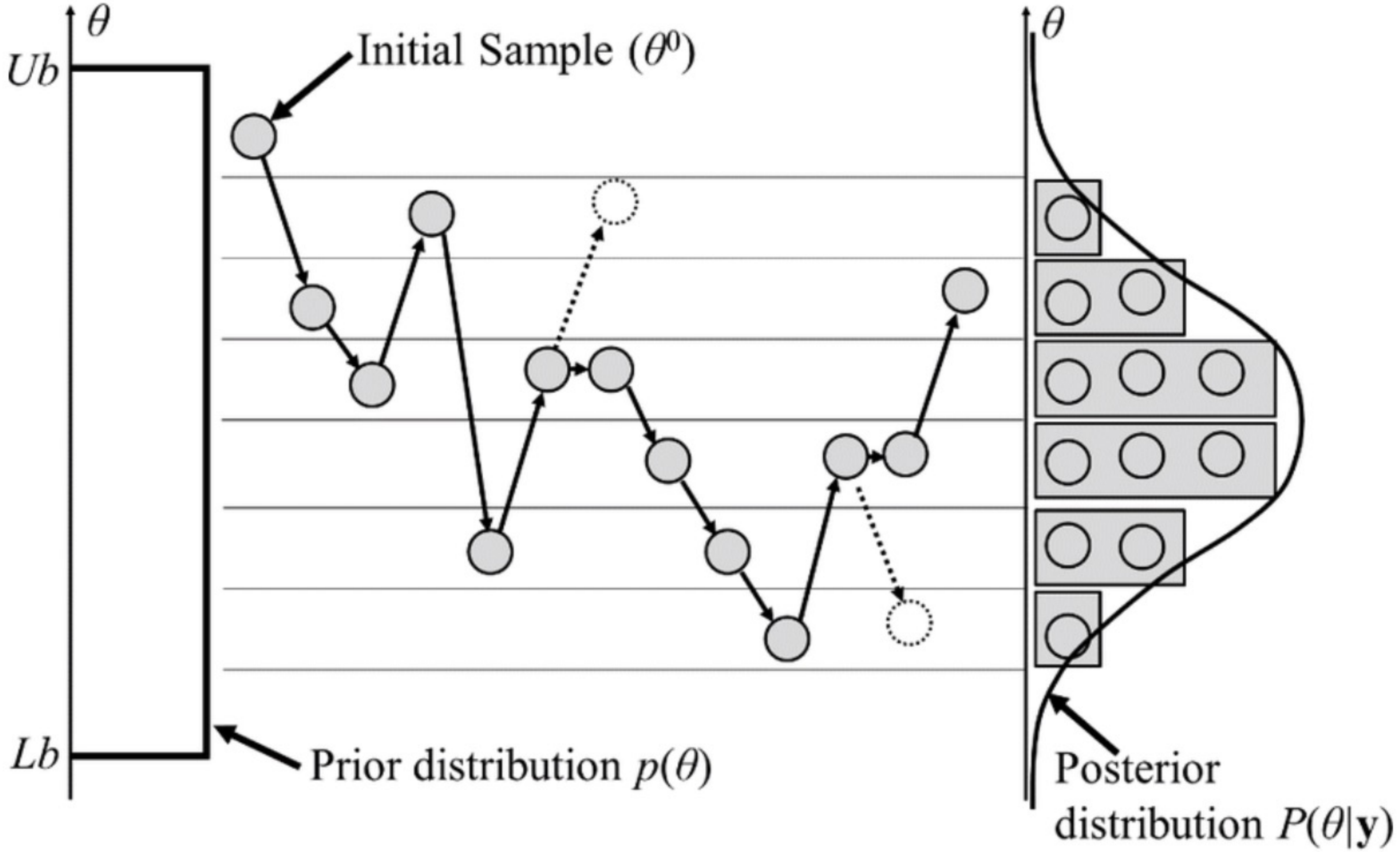
Step 4: If $u < \alpha(\theta_{new}, \theta_{t-1}) \rightarrow$ If $0.617 < 0.834$ Then $\theta_t = \theta_{new} = 0.306$
Otherwise $\theta_t = \theta_{t-1} = 0.429$

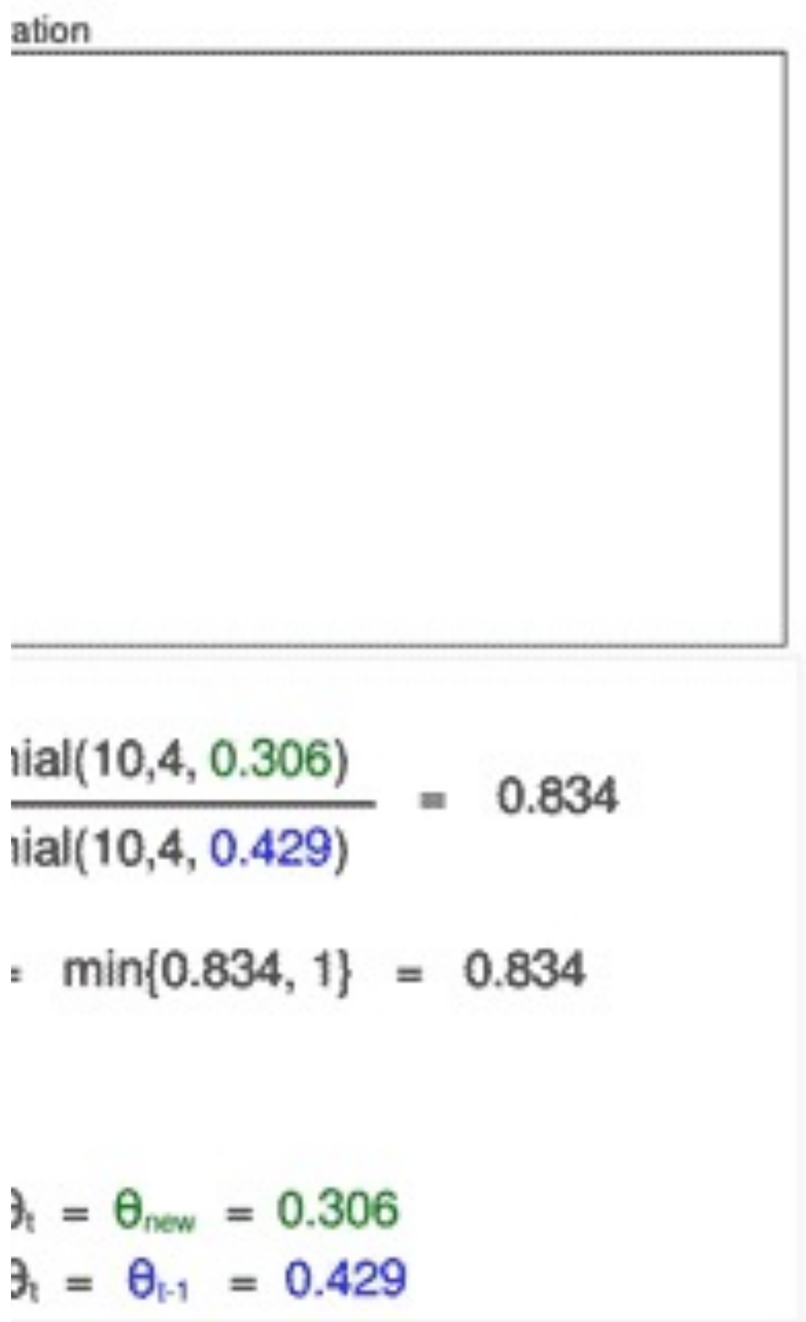
Credits: https://blog.stata.com/

Markov Chain Monte Carlo (MCMC)

Performing MCMC

1. Define
2. Define t
3. Start with c
4. Propose a r
   (
5. Accept it
6. Repeat steps

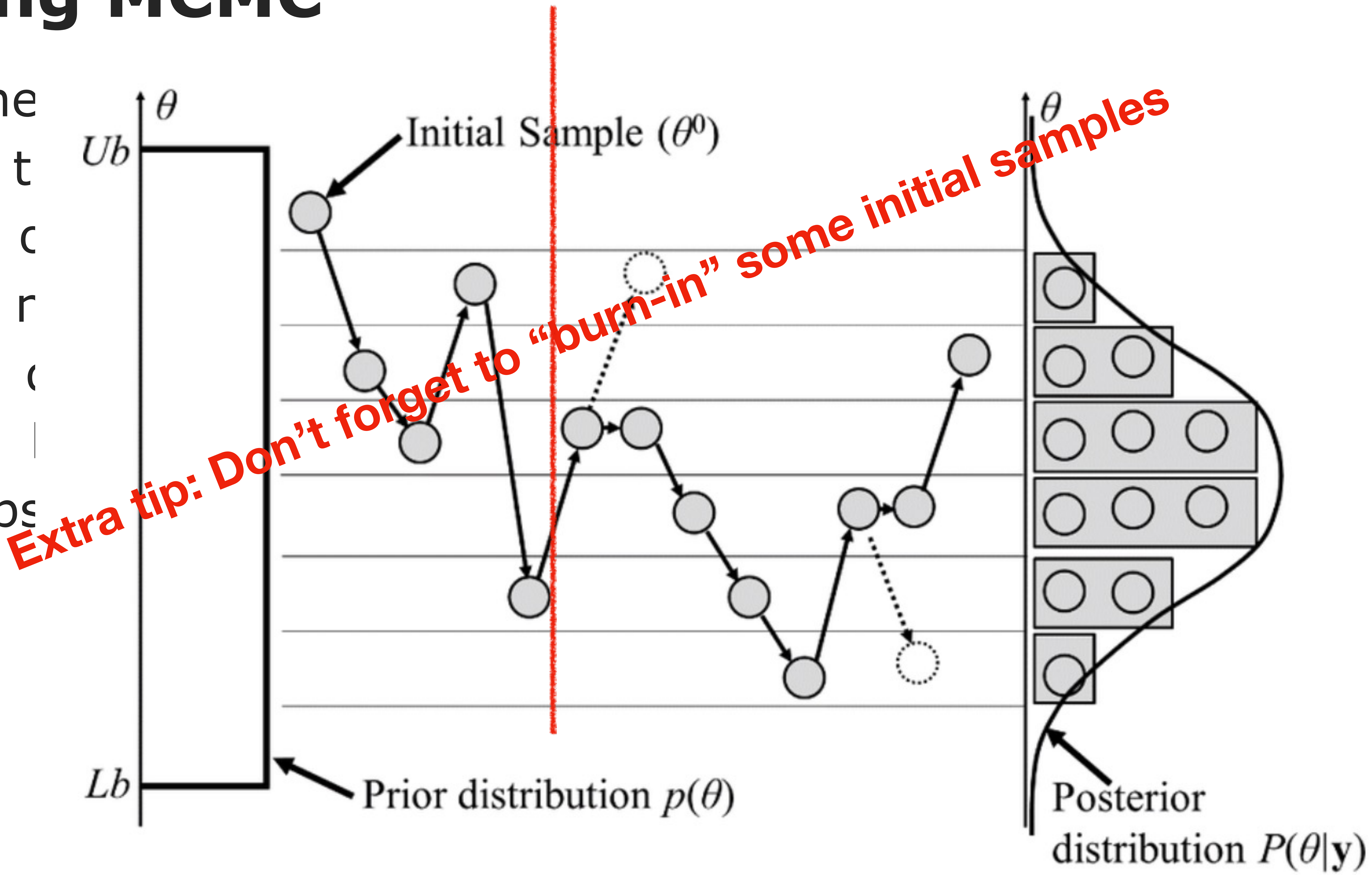


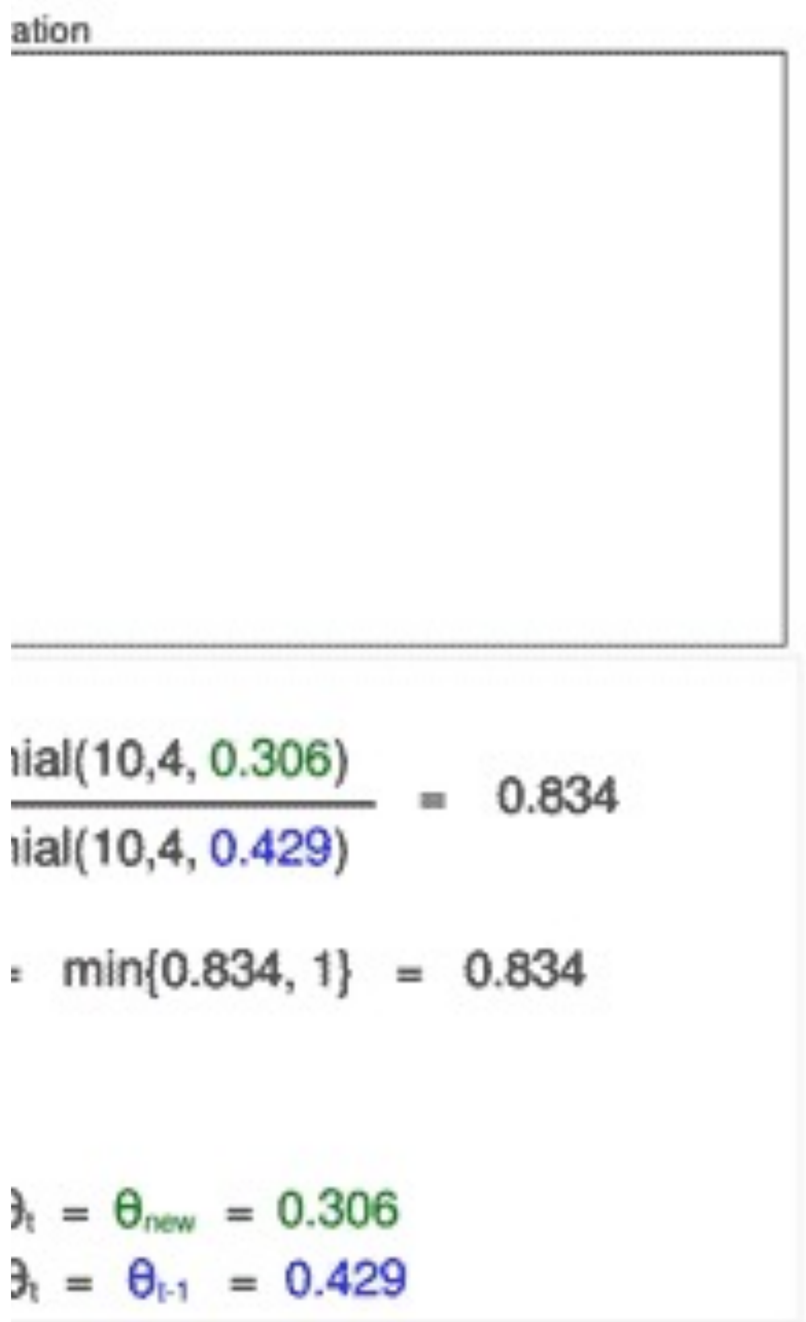


Markov Chain Monte Carlo (MCMC)

Performing MCMC

1. Define
2. Define t
3. Start with c
4. Propose a r
   (
5. Accept it
6. Repeat steps





Markov Chain Monte Carlo (MCMC)

…with Eryn

Such routines are already adopted in **Eryn**, so basically you don't have to do a lot, just define some things!

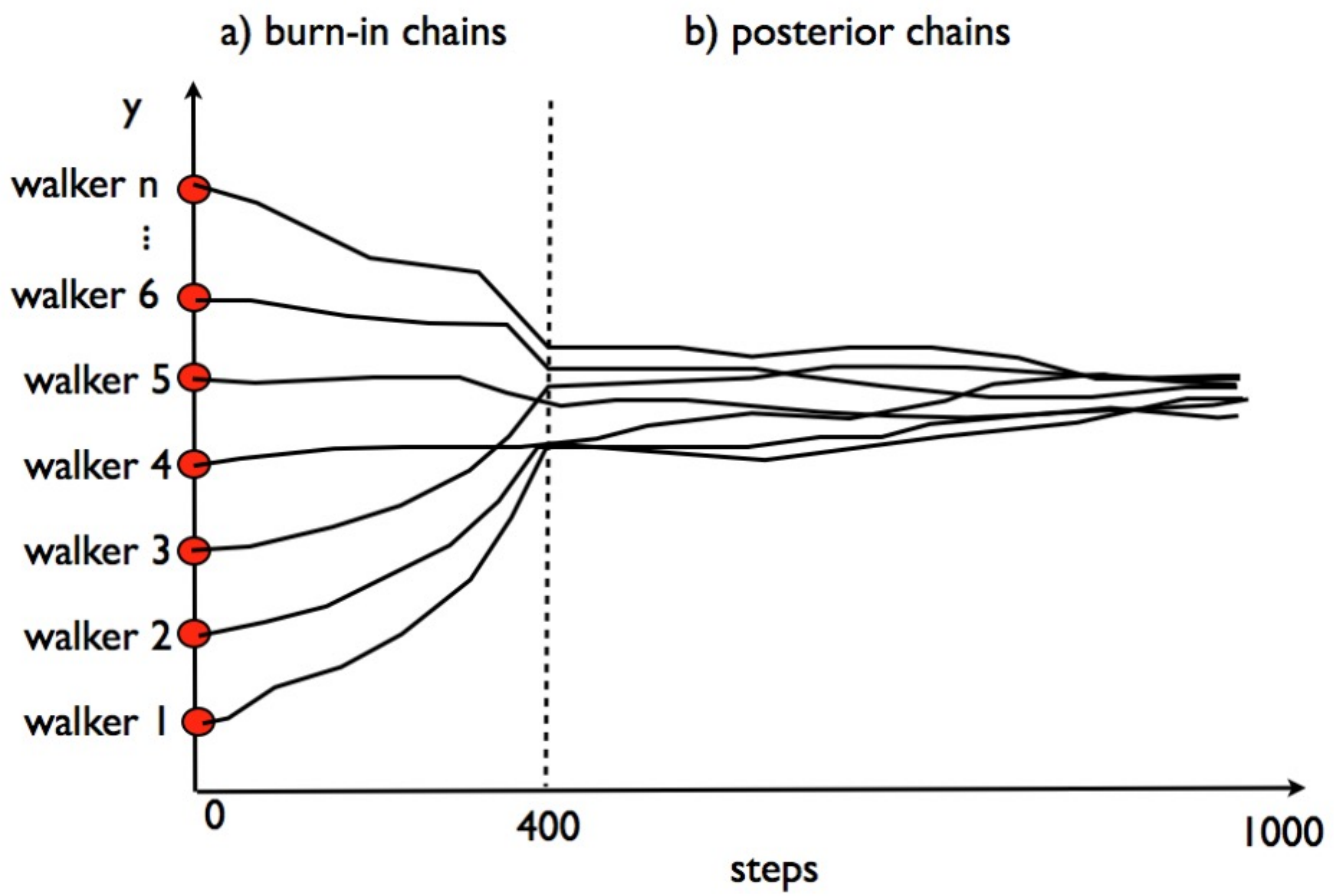
1. Define the log-likelihood $p(y \mid \vec{\theta}) = (2\pi\sigma)^{-1/2} \times e^{-y^2/2} \implies \log(p(y \mid \vec{\theta})) = -\chi^2/2 - \frac{1}{2}\log(2\pi\sigma)$
2. Define the prior (proposal distribution) `from eryn.prior import uniform_dist, ProbDistContainer`
3. Define the sampler `from eryn.ensemble import EnsembleSampler`
4. Choose start point
5. Put the starting point into a State object `from eryn.state import State`
6. Run mcmc with **run_mcmc**

Markov Chain Monte Carlo (MCMC)

...with Eryn

Extra features:

- **Walkers (*trees*)**: The members of the ensemble. The proposals for new steps for each walker then depend on the relative locations of the other walkers. This allows the group of walkers to evolve like an amoeba through parameter space, ***stretching and shrinking*** in ways that follow the posterior.



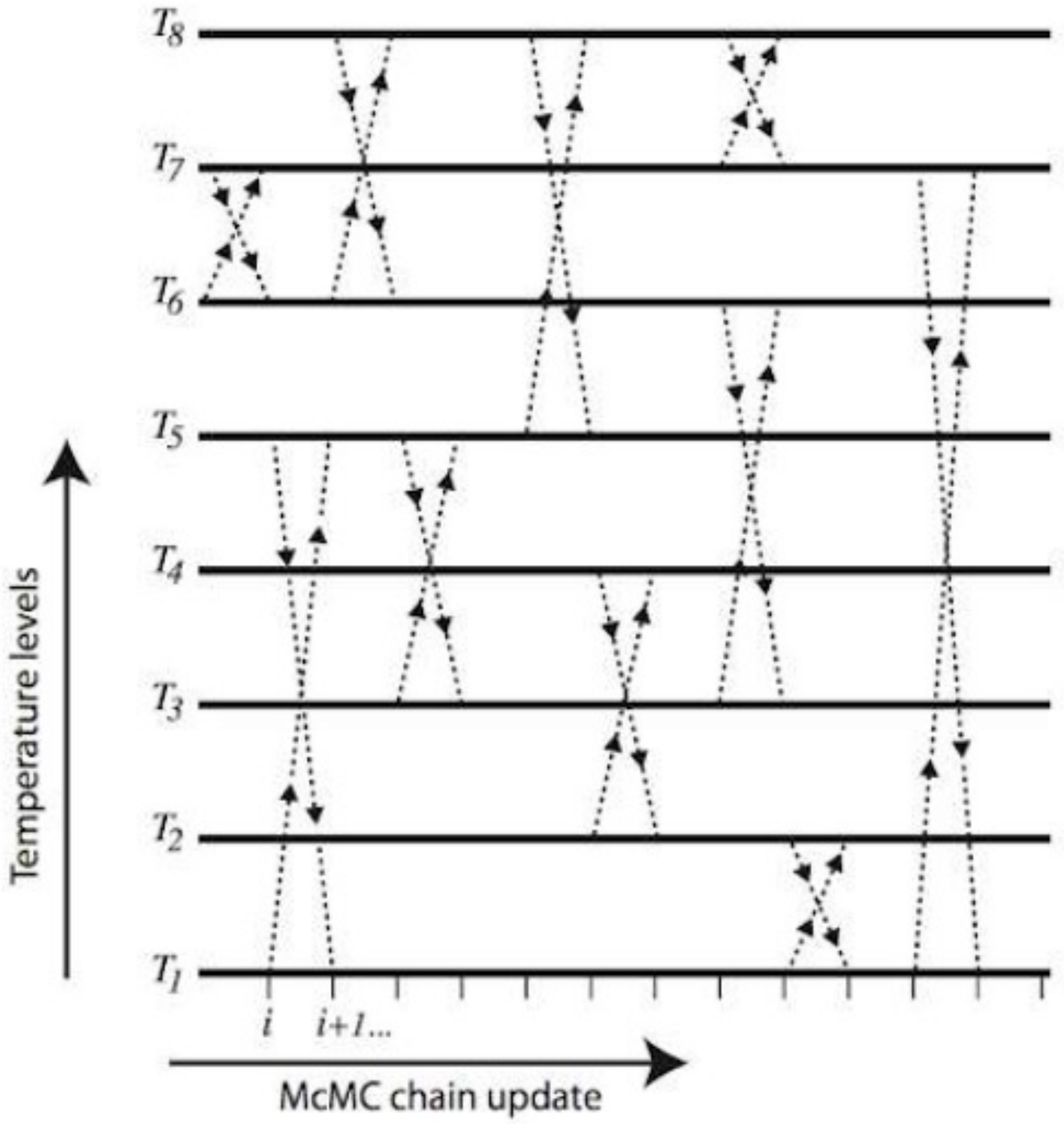
Credits: Dr. Veselina Kalinova, *Lecture notes on Regression: MCMC*

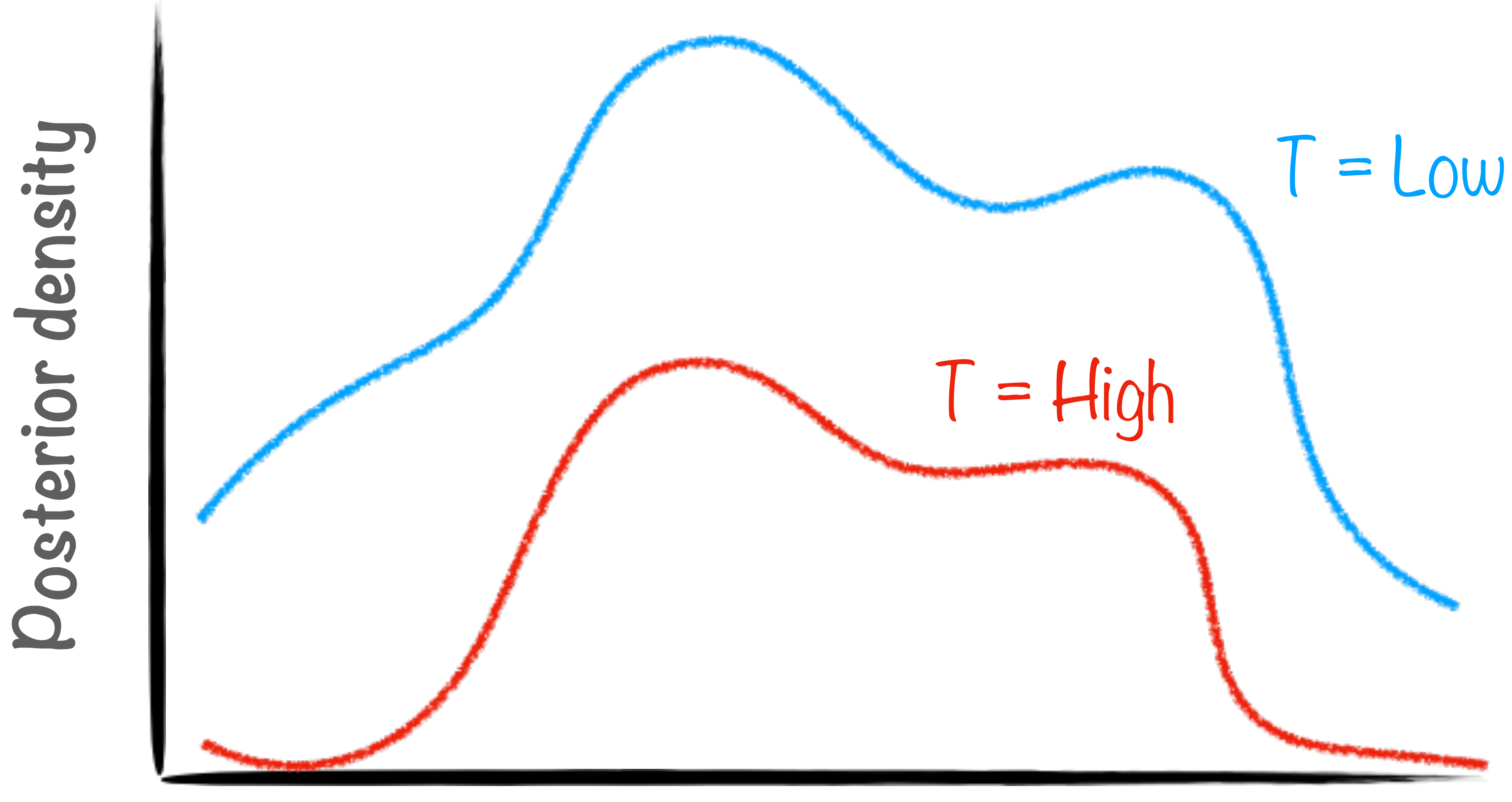
Markov Chain Monte Carlo (MCMC)

...with Eryn

Extra features:

- **Temperatures (forest)**: For each different temperature T, the posterior density is transformed to a density with a different temperature. These chains periodically exchange information. This helps to properly sample distributions with multiple posterior modes.





$$\ln p_\beta(\vec{\theta}, \mathcal{M}) \propto \underbrace{\beta}_{1/T} \ln p(\vec{\theta}, \mathcal{M})$$

Markov Chain Monte Carlo (MCMC)

...with Eryn

Extra features:

- **Evidence**: Is computed via the *thermodynamic integration* $\mathcal{Z}_{i,\beta} = \int p(y|\vec{\theta}, \mathcal{M}_i)^{\beta} p(\vec{\theta})\, \mathrm{d}\vec{\theta}.$

  Conduce to identify the model best supported by the observed data.

- **Models (branches)**

- RJMCMC (leaves)

tomorrow by N. Karnesis

Markov Chain Monte Carlo (MCMC)

...with Eryn

Extra features:

- **Evidence**: Is computed via the *thermodynamic integration* $\mathcal{Z}_{i,\beta} = \int p(y|\vec{\theta}, \mathcal{M}_i)^\beta p(\vec{\theta})\, \mathrm{d}\vec{\theta}.$

  Conduce to identify the model best supported by the observed data.

- **Models (branches)**
- RJMCMC (leaves)

tomorrow by N. Karnesis

Let's start coding with Eryn!

Thank you!