Markov Chain Monte Carlo (MCMC)

Markovian: process where each step only depends on current location

Chain: succession of steps in parameter space

Monte Carlo: technique that makes use of random numbers

Viviana Acquaviva (CUNY)
vacquaviva@citytech.cuny.edu

In many cases, MCMC will be a way of doing INFERENCE on parameters.

In Bayesian terms:

Posterior probability density function

Some constant

Priors (how you expect your params to be distributed)

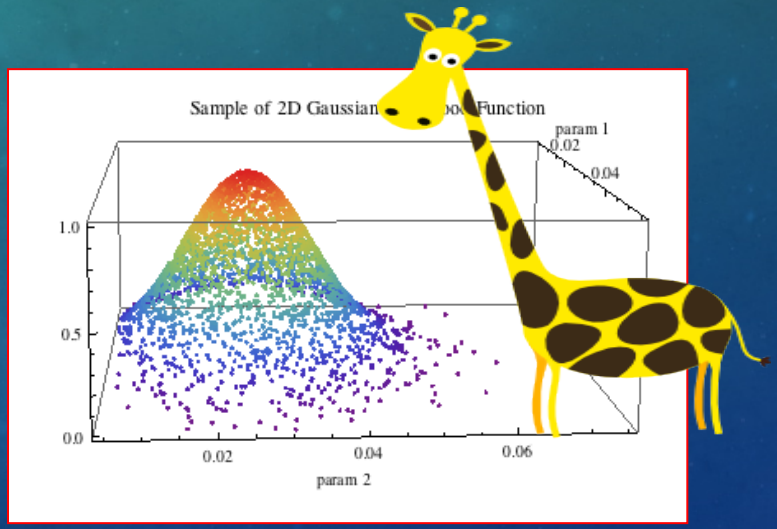
$$p(\theta \mid D) = \frac{1}{Z}\, p(D \mid \theta)\, p(\theta)$$

Likelihood (how you expect the data D to look like for certain params θ)

Do we really need to know the full Probability Distribution Function (in many D)?

Not really: we want *marginalized* constraints on parameters, which are obtained *integrating* the PDF in the other dimensions.

1D constraints are not “slices” but integrals!



Example: the expectation value (mean) of a function f is:

$$\langle f(\mathbf{x}) \rangle = \int dx_1 ... dx_n f(\mathbf{x}) p(\mathbf{x})$$

f can be, say, the Age of a galaxy and p is the PDF;
x_1 … x_n are the all the parameters: Age, Mass, Dust, z, Z, τ.

Now, imagine having a series of points r_i in parameter space *distributed identically* to the PDF.

THE HORRIBLE INTEGRAL BECOMES AN EASY SUM:

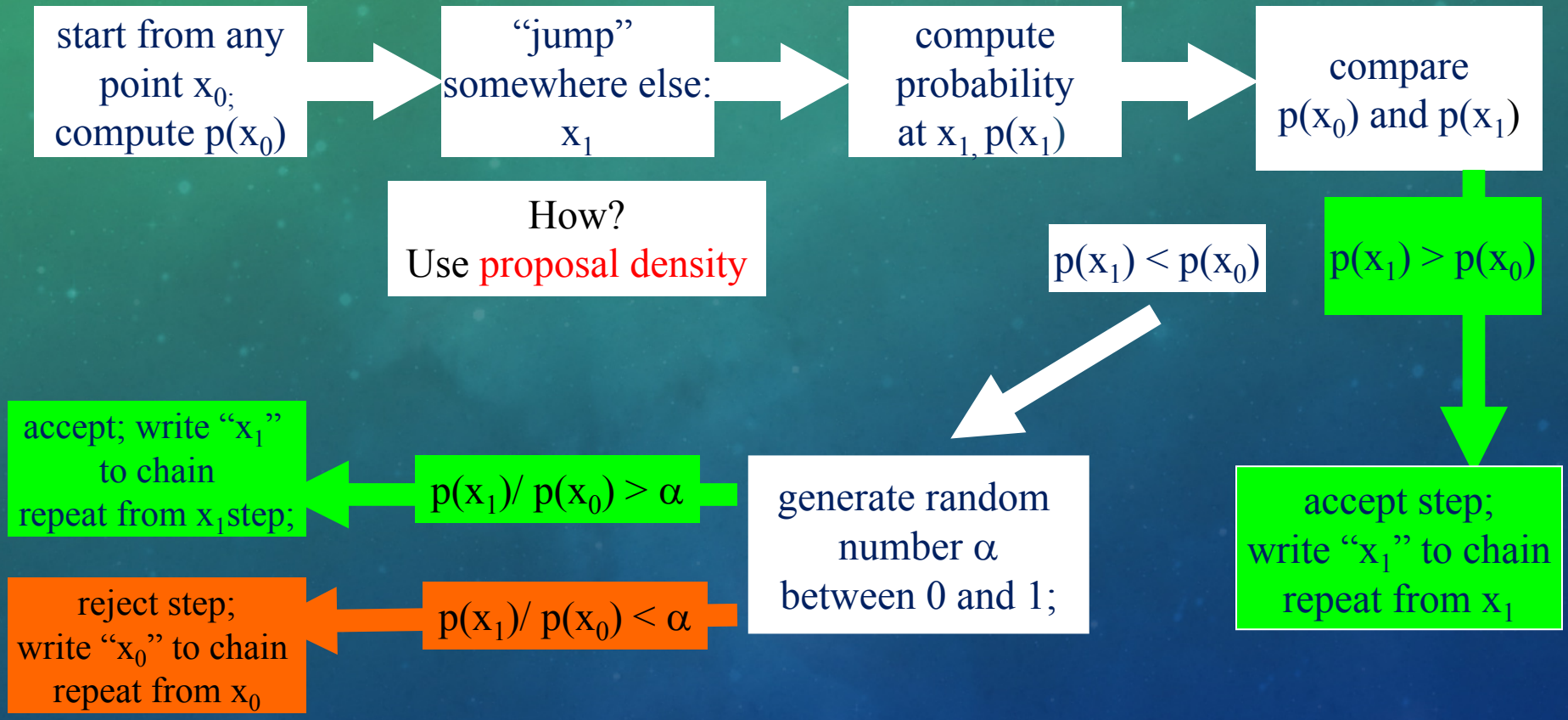
$$\langle f(\mathbf{x}) \rangle = \int dx_1 ... dx_n f(\mathbf{x}) p(\mathbf{x}) \simeq \frac{1}{R} \sum_{i=1}^{R} f(r_i)$$

MCMC PROVIDES A PRESCRIPTION TO CREATE A SERIES OF POINTS (CHAIN) DISTRIBUTED IDENTICALLY TO THE PROBABILITY DISTRIBUTION FUNCTION.

HOW?

EXPLORING THE PARAMETER SPACE THROUGH A RANDOM WALK INTENTIONALLY BIASED SO THAT THE DENSITY OF VISITED POINTS IS PROPORTIONAL TO THE PDF.

Perhaps the simplest implementation is the Metropolis algorithm:



probability $p(x) \propto$ likelihood x priors

HAPPY THINGS

EASY! As we mentioned, computing statistics =

COUNTING POINTS

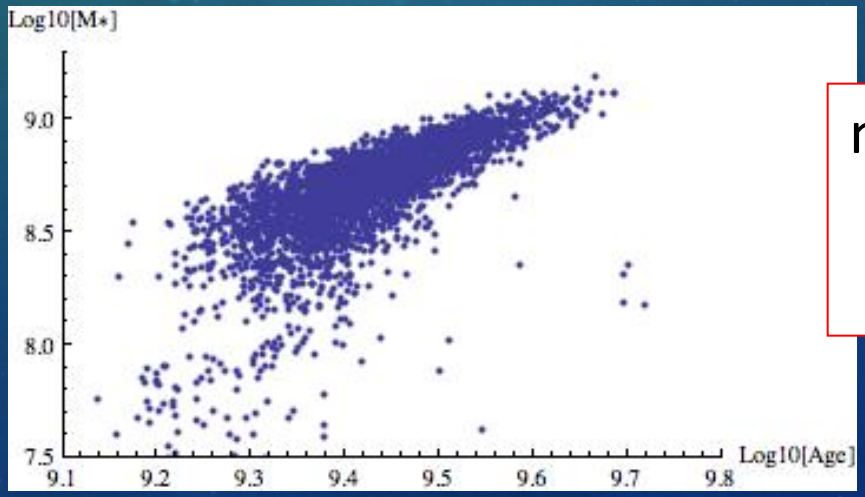
We saw already how to compute things like the mean. How about uncertainties (credible intervals?)

1. Assume Gaussian PDF and compute 68%, 95% as 1 and 2 σ deviations from best fit: WRONG

2. Integrate PDF in many dimensions to find contours enclosing 68%, 95% of total volume: PAINFUL

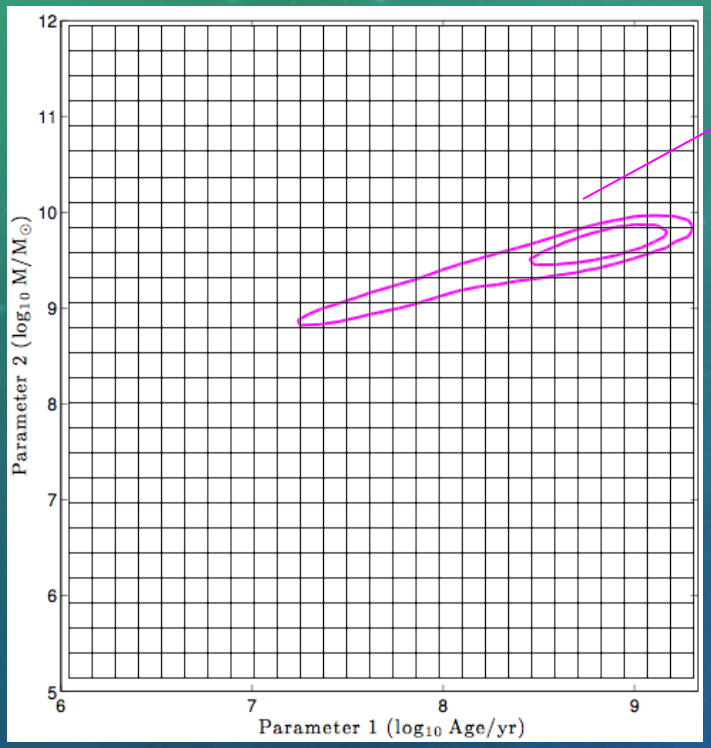
MCMC:

COUNT POINTS



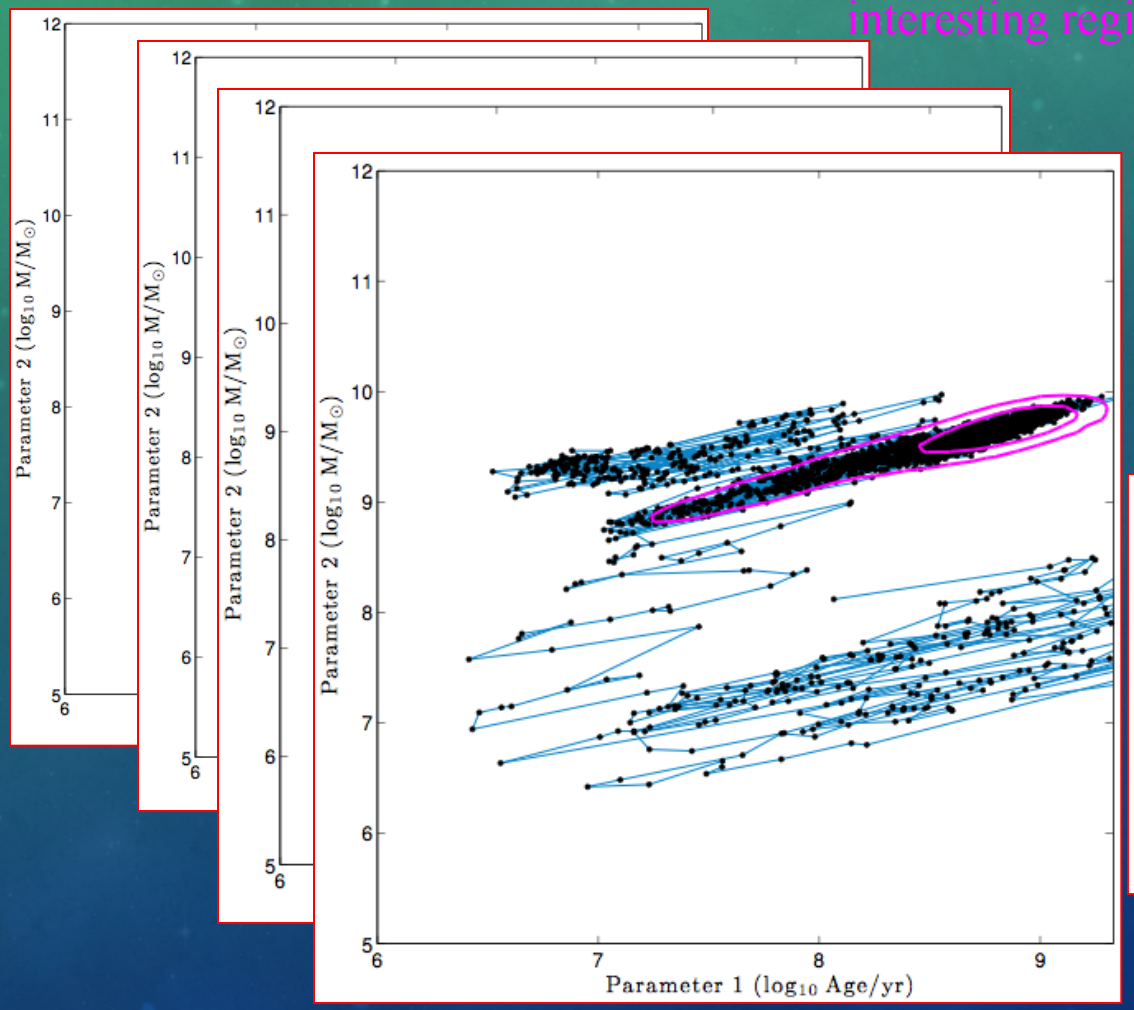
no assumptions for PDF shape; accurate

EFFICIENT SAMPLING



GRID: spend large fraction
of time in uninteresting regions

EFFICIENT SAMPLING



THE MCMC WAY:

EFFICIENT!
most time spent in informative region even if you don't previously know where it is.

UNHAPPY THINGS

The whole MCMC architecture hinges on the fact that the samples are distributed like the PDF.

How do we know that this is true?

We don't; only NECESSARY conditions.

CONVERGENCE TESTS
(did samples reach a stationary distribution?)

- ✓ do results change if I keep running?
- ✓ do results differ if I run many chains, starting in different places?

GOOD SAMPLER, BAD SAMPLER?

M-H algorithm is easy, but many knobs to turn to improve efficiency

Simple knob: proposal density (acceptance rate), want to be in Goldilock zone

More in general, error on MCMC estimates scales like sqrt (autocorrelation time/Nsamples) so small a.c. times are better (could optimize for that)

Simple MCMC (e.g. M-H) breaks in presence of multiple, separated peaks; solution not obvious