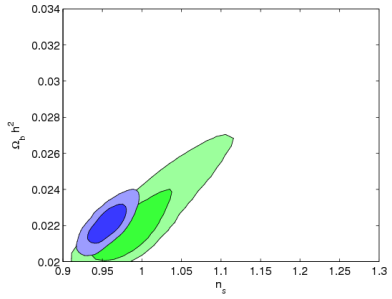
Computing posteriors

- For 2 parameters, a grid is usually possible
  - Marginalise by numerically integrating along each axis of the grid



- For >>2 parameters it is not feasible to have a grid (e.g. 10 points in each parameter direction, 12 parameters = 10^{12} likelihood evaluations)
- We can instead *sample* the distribution

Sampling

- Draw samples with a probability proportional to the *target distribution* (likelihood, or the posterior, or something else)
- Approximate target by a sum of Dirac delta functions:

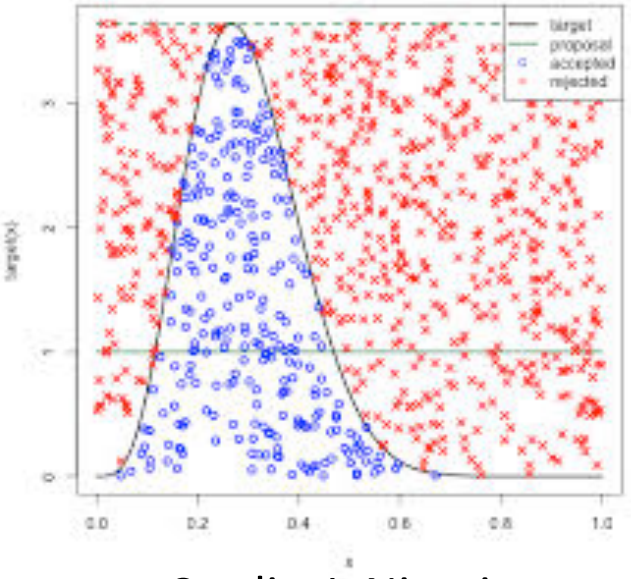
$$p(\theta) \simeq \frac{1}{N} \sum_{\text{samples } i} \delta^D(\theta - \theta_i)$$

- Mean of any function is given approximately by

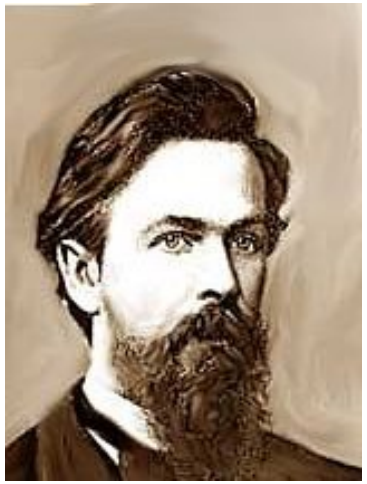
$$\langle f \rangle = \int f(\theta)\, p(\theta) d\theta \simeq \frac{1}{N} \sum_{\text{samples } i} f(\theta_i)$$

Advantages of sampling

- Points may be concentrated where the target is high
- Converges as number of samples $\longrightarrow \infty$
- e.g. Rejection sampling:
- Very inefficient in general

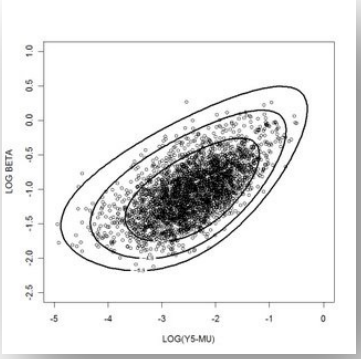


Credit: J. Niemi



MCMC
Markov Chain
Monte Carlo

Aim of MCMC: generate a set of points in the parameter space whose distribution function is the same as the target density.



MCMC follows a Markov process - i.e. the next sample depends on the present one, but not on previous ones.

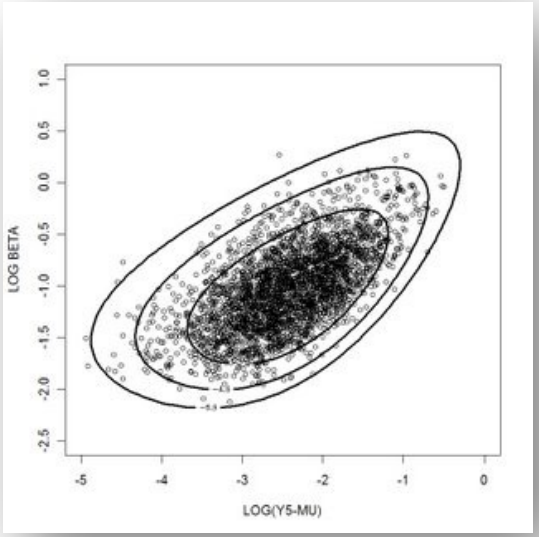
Target density

The *target density* is approximated by a set of delta functions (you may need to normalise):

$$p(\boldsymbol{\theta}) \simeq \frac{1}{N} \sum_{i=1}^{N} \delta(\boldsymbol{\theta} - \boldsymbol{\theta}_i)$$

and we can estimate the mean of any function f by

$$f(\boldsymbol{\theta}) \simeq \frac{1}{N} \sum_{i=1}^{N} f(\boldsymbol{\theta}_i).$$



Metropolis-Hastings algorithm

Proposal Distribution

$q(\theta^*|\theta)$

$$p(acceptance) = min\left[1, \frac{p(\theta^*)q(\theta^*|\theta)}{p(\theta)q(\theta|\theta^*)}\right]$$

Metropolis algorithm (special case):

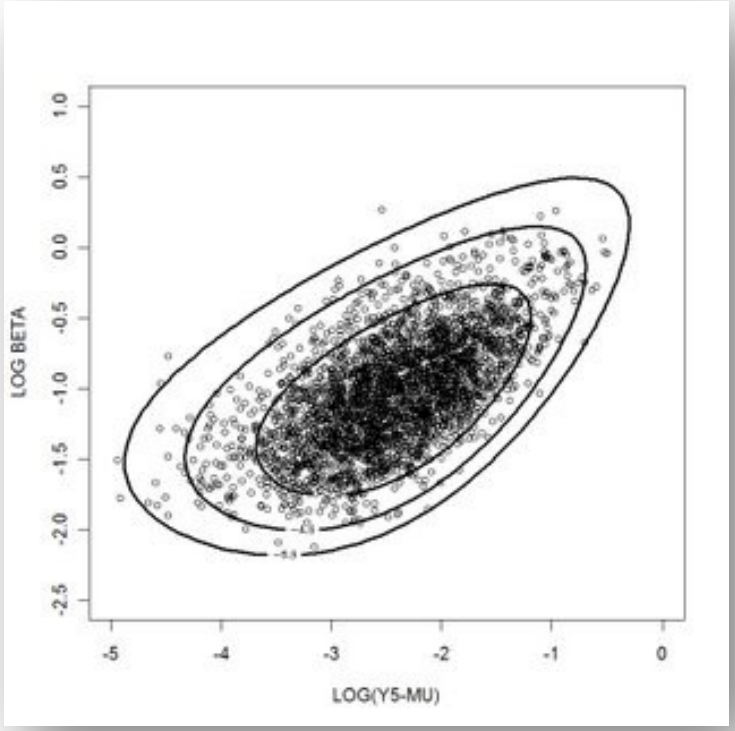
$$min\left[1, \frac{p(\theta^*)}{p(\theta)}\right]$$

MCMC Algorithm

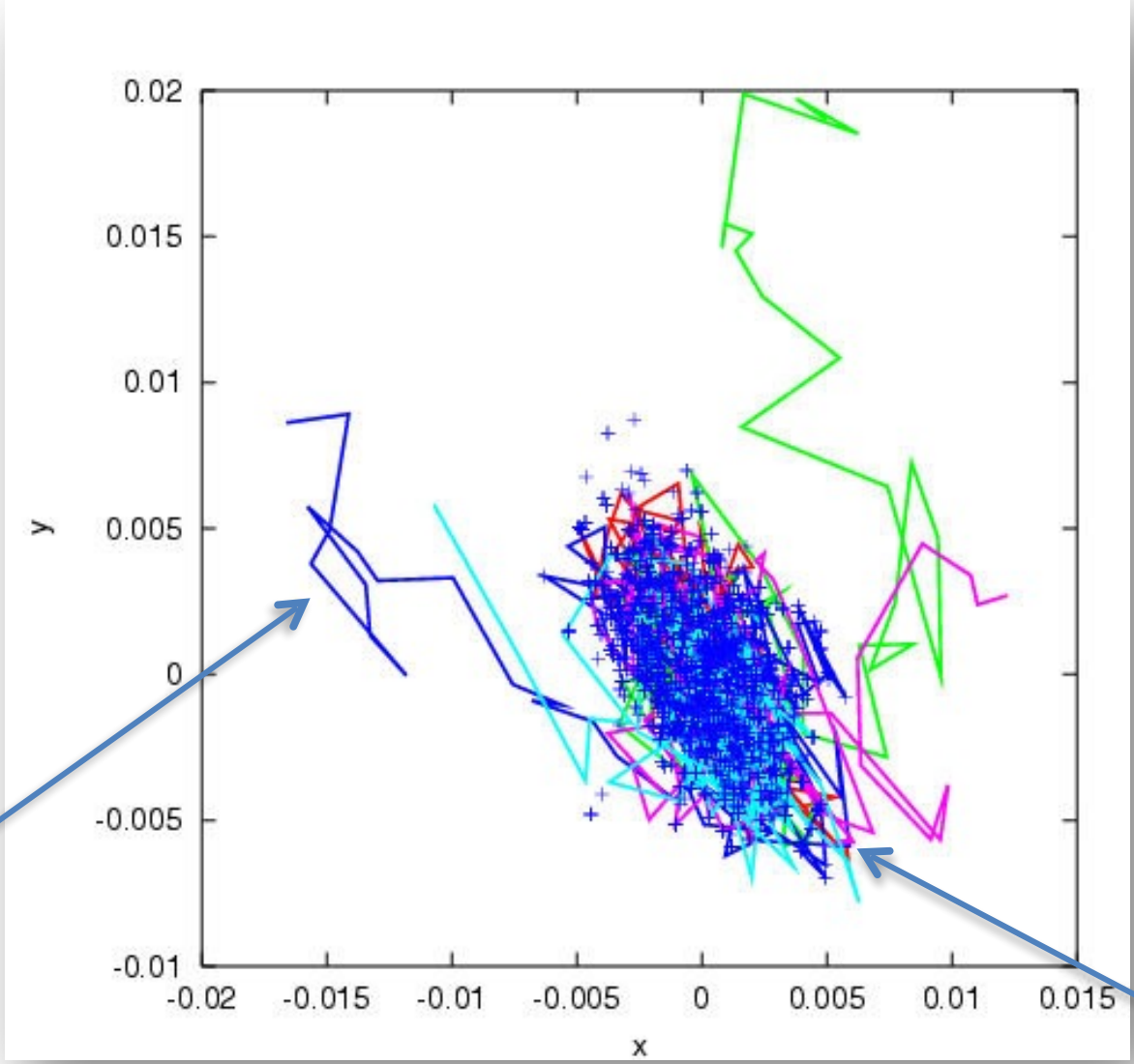
- Choose a random initial starting point in parameter space, and compute the target density.
- Repeat:
  - Generate a step in parameter space from a proposal distribution, generating a new trial point for the chain.
  - Compute the target density at the new point, and accept it (or not) with the Metropolis-Hastings algorithm.
  - If the point is not accepted, the previous point is repeated in the chain.
- End Repeat:

The proposal distribution

- Too small, and it takes a long time to explore the target
- Too large and almost all trials are rejected
- q ~ `Fisher size' is good.



Burn-in

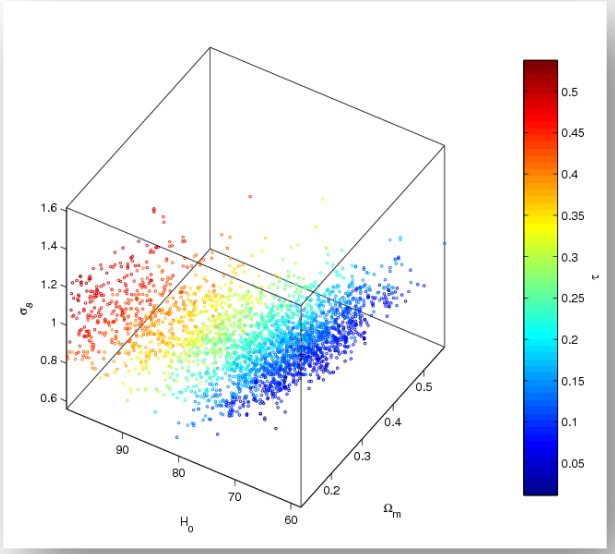


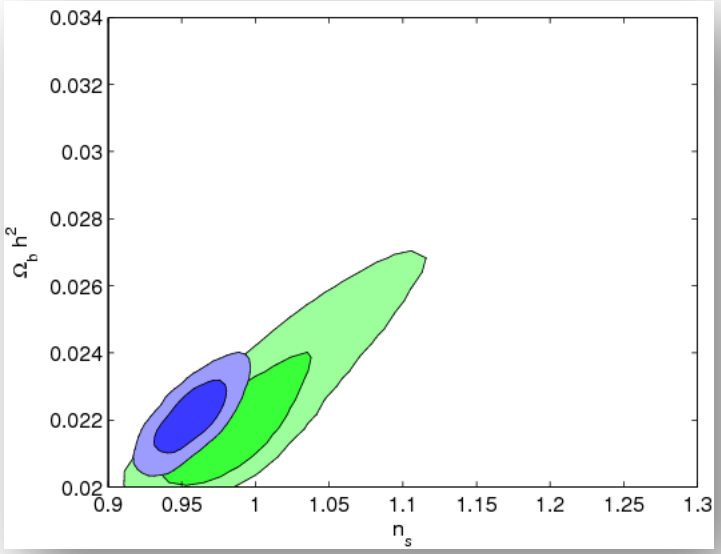
"Burn-in"

Points are correlated

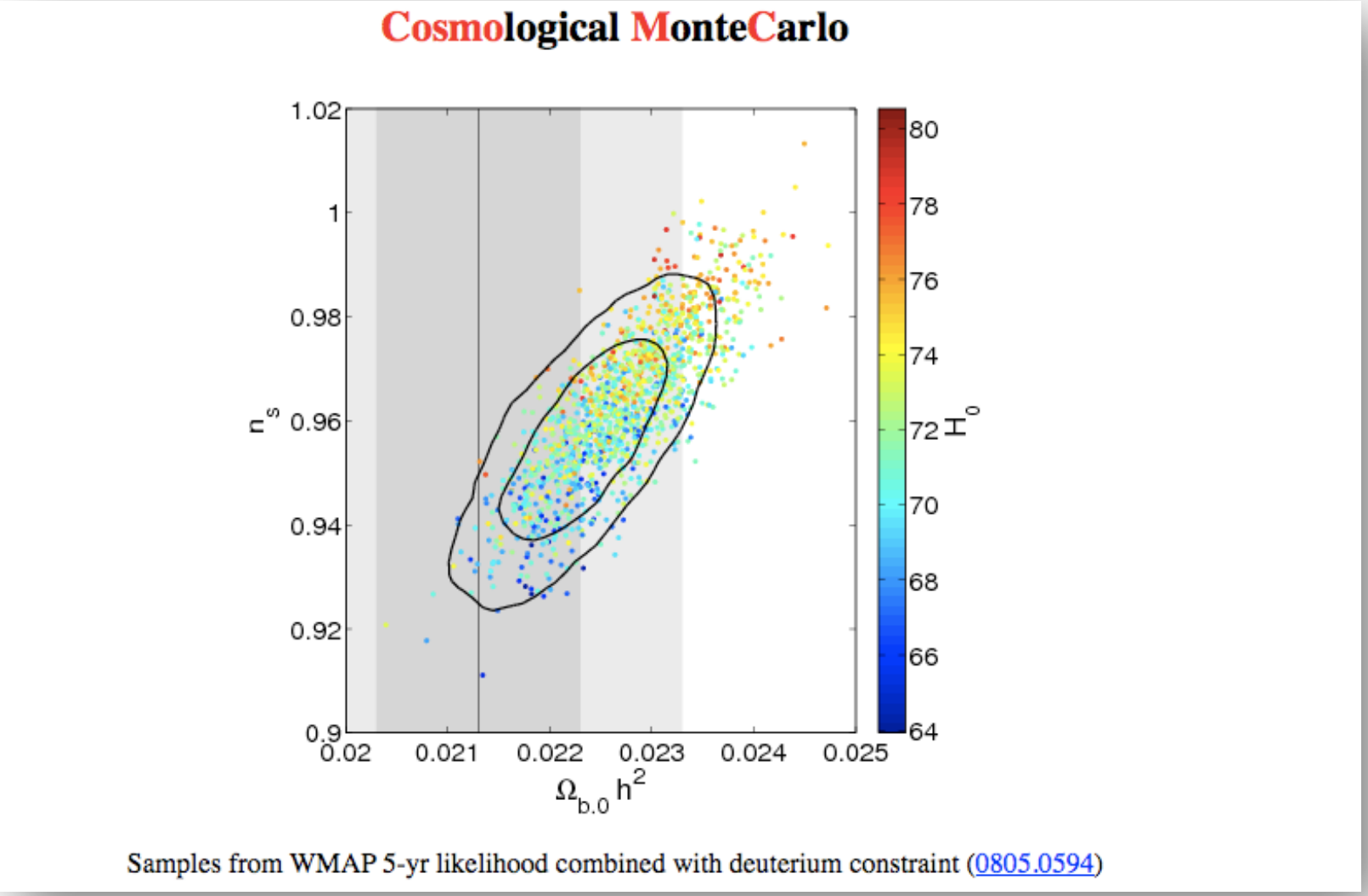
Marginalisation

- Marginalisation is trivial
  - Each point in the chain is labelled by all the parameters
  - To marginalise, just ignore the labels you don't want



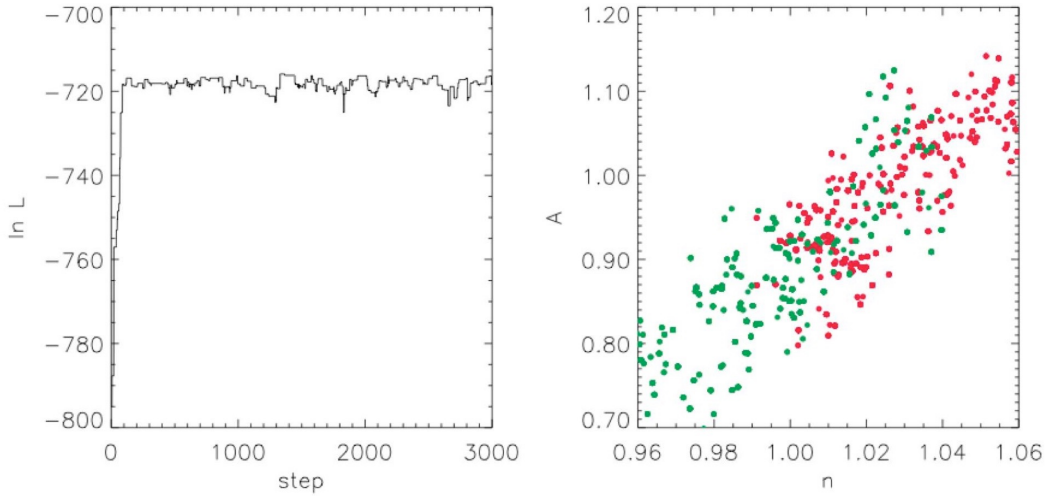


CosmoMC

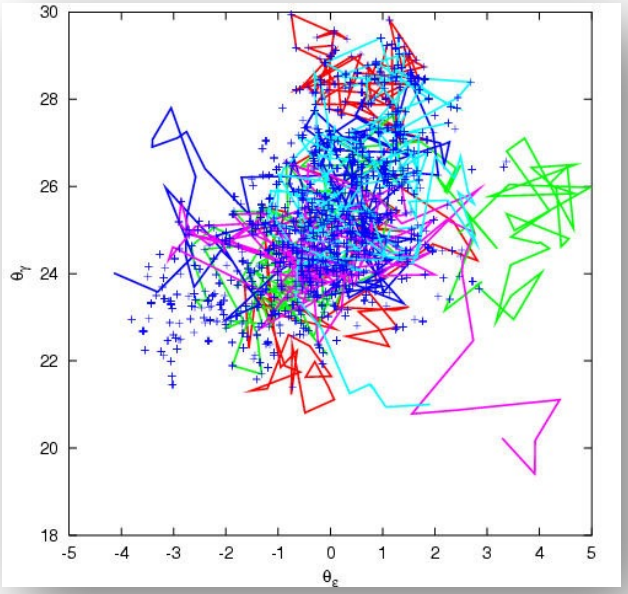


http://cosmologist.info/cosmomc/

Convergence



Verde et al 2003



You must use a convergence test.
Gelman-Rubin test is most common

Gelman-Rubin test (1992)

- Run M chains, each of length N, from different starting points
- Basic idea: compare different estimates of the variance of the points. If the chains have converged, they should agree well

Gelman-Rubin algorithm

and Rubin Diagnostics

and Rubin diagnostics (Gelman and Rubin; 1992; Brooks and Gelman; 1997) are based on analyzing multiple simulated MCMC chains by comparing
s within each chain and the variance between chains. Large deviation between these two variances indicates nonconvergence.

$\theta^t\}$, where $t = 1, \ldots, n$, to be the collection of a single Markov chain output. The parameter θ^t is the tth sample of the Markov chain. For notational si
med to be single dimensional in this section.

you have M parallel MCMC chains that were initialized from various parts of the target distribution. Each chain is of length n (after discarding the bu
θ^t, the simulations are labeled as θ^t_m, where $t = 1, \ldots, n$ and $m = 1, \ldots, M$. The between-chain variance B and the within-chain variance W are calc

$$= \frac{n}{M-1}\sum_{m=1}^{M}(\bar{\theta}^{\cdot}_m - \bar{\theta}^{\cdot}_{\cdot})^2, \text{ where } \bar{\theta}^{\cdot}_m = \frac{1}{n}\sum_{t=1}^{n}\theta^t_m, \ \bar{\theta}^{\cdot}_{\cdot} = \frac{1}{M}\sum_{m=1}^{M}\bar{\theta}^{\cdot}_m$$

$$= \frac{1}{M}\sum_{m=1}^{M}s^2_m, \text{ where } s^2_m = \frac{1}{n-1}\sum_{t=1}^{n}(\theta^t_m - \bar{\theta}^{\cdot}_m)^2$$

erior marginal variance, $\mathrm{var}(\theta|\mathbf{y})$, is a weighted average of W and B. The estimate of the variance is

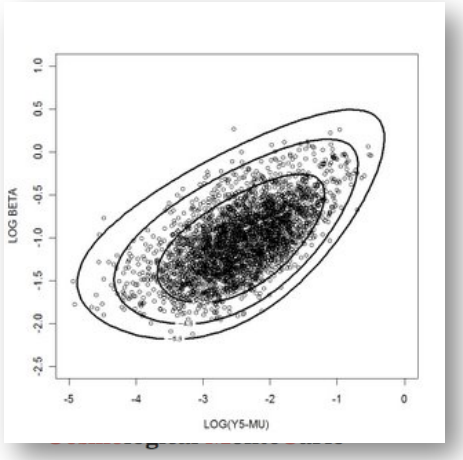
$$\frac{n-1}{n}W + \frac{M+1}{nM}B$$

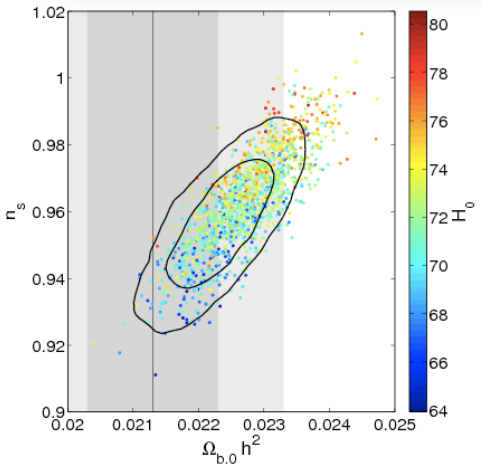
hains have reached the target distribution, this posterior variance estimate should be very close to the within-chain variance W. Therefore, you would
e ratio $\widehat{V}/W$ be close to 1. The square root of this ratio is referred to as the *potential scale reduction factor* (PSRF). A large PSRF indicates that the
-chain variance is substantially greater than the within-chain variance, so that longer simulation is needed. If the PSRF is close to 1, you can conclud
he M chains has stabilized, and they are likely to have reached the target distribution.

From SAS STAT user guide

Combining Chains - Importance Sampling

- This is NOT TRIVIAL
- For 2 independent experiments, the likelihoods L1 and L2 *multiply*
- For small numbers of parameters, chains can be smoothed and multiplied
- For many dimensions, can do *IMPORTANCE SAMPLING*: compute L2 at the positions of the L1 chain, and give the point a weight L2.





Samples from WMAP 5-yr likelihood combined with deuterium constraint (0805.0594)

(Importance sampling draws from a distribution different from the target)

Gibbs Sampling

- Sampling from a high-dimensional space without symmetries is very hard to do efficiently
- Gibbs sampling can be applied if *conditional* distributions are known, i.e.,

$$p(\theta_i | \text{all other } \theta_j)$$

- Then one can cycle through each parameter in turn
- Can be especially useful for hierarchical models (see later)
- Can sometimes combine with Metropolis-Hastings
- JAGS: http://mcmc-jags.sourceforge.net/
- BUGS: www.mrc-bsu.cam.ac.uk/software/bugs/

Hybrid (Hamiltonian) Monte Carlo

- We would like to increase the acceptance rate to improve efficiency, and explore the target distribution efficiently (‘good mixing’)
- We have a hard problem in many dimensions. Solution:
- Make things harder! add in M auxiliary variables, one for each parameter in the model.
- Imagine each of the parameters in the problem as a coordinate.
- Target distribution ➔ effective potential
- For each coordinate HMC generates a generalised momentum.
- It then samples from the extended target distribution in 2M dimensions.

HMC

- HMC explores this 2M-dimensional space by treating the problem as a dynamical system, and evolving the phase space coordinates by solving the dynamical equations.
- Finally, it ignores the momenta (marginalising, as in MCMC), and this gives a sample of the original target distribution.
- May help with decorrelating the points in the chain.
- Invented by particle physicists (Duan et al 1987)

Theory

- Potential $U(\theta) = -\ln p(\theta)$
- For each θ_α, generate a *momentum* u_α (gaussian distributed)
- K.E. $K = u^T u/2$
- Define a $H(\boldsymbol{\theta}, \mathbf{u}) \equiv U(\boldsymbol{\theta}) + K(\mathbf{u})$

- and define an *extended target density*

$$p(\boldsymbol{\theta}, \mathbf{u}) = \exp\left[-H(\boldsymbol{\theta}, \mathbf{u})\right]$$

Magic of HMC

William Rowan Hamilton

- Evolve as a *dynamical system*

$$\dot{\theta}_\alpha = u_\alpha$$

$$\dot{u}_\alpha = -\frac{\partial H}{\partial \theta_\alpha}$$

- H remains constant, so extended target density is uniform – *all points get accepted!*
- Also, you can make big jumps – good mixing, if you generate a new u each time

Complications

- Evolving the system takes time. Take big steps.
- Need derivatives of U = - ln p
- We may not have them, and *might approximate* U (from a short MCMC)

$$U = \frac{1}{2}(\theta - \theta_0)_\alpha C^{-1}_{\alpha\beta}(\theta - \theta_0)_\beta$$

- H is therefore not constant
- Use Metropolis-Hastings. Accept new point with probability

$$min\left\{1, \exp\left[-H(\boldsymbol{\theta}^*, \mathbf{u}^*) + H(\boldsymbol{\theta}, \mathbf{u})\right]\right\}$$

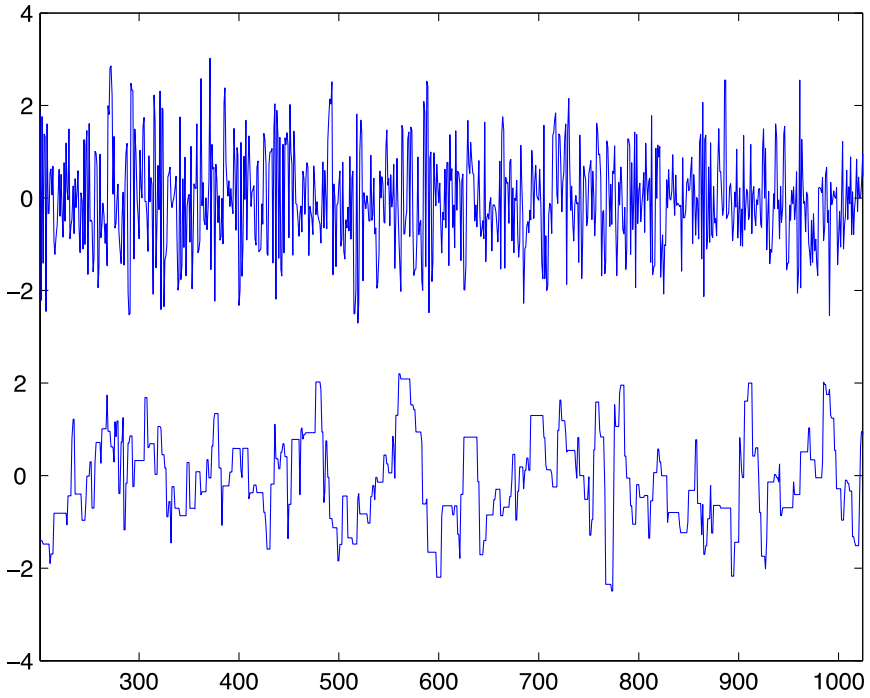
Algorithm

Hamiltonian Monte Carlo

1: initialize $\mathbf{x}_{(0)}$
2: for i = 1 to $N_{samples}$
3: $\mathbf{u} \sim \mathcal{N}(0, 1)$
4: $(\mathbf{x}^*_{(0)}, \mathbf{u}^*_{(0)}) = (\mathbf{x}_{(i-1)}, \mathbf{u})$
5: for j = 1 to N
6: make a leapfrog move: $(\mathbf{x}^*_{(j-1)}, \mathbf{u}^*_{(j-1)}) \rightarrow (\mathbf{x}^*_{(j)}, \mathbf{u}^*_{(j)})$
7: end for
8: $(\mathbf{x}^*, \mathbf{u}^*) = (\mathbf{x}_{(N)}, \mathbf{u}_{(N)})$
9: draw $\alpha \sim (0, 1)$
10: if $\alpha < \min\{1, e^{-(H(\mathbf{x}^*, \mathbf{u}^*) - H(\mathbf{x}, \mathbf{u}))}\}$
11: $\mathbf{x}_{(i)} = \mathbf{x}^*$
12: else
13: $\mathbf{x}_{(i)} = \mathbf{x}_{(i-1)}$
14: end for

From Hajian 2006

HMC vs MCMC



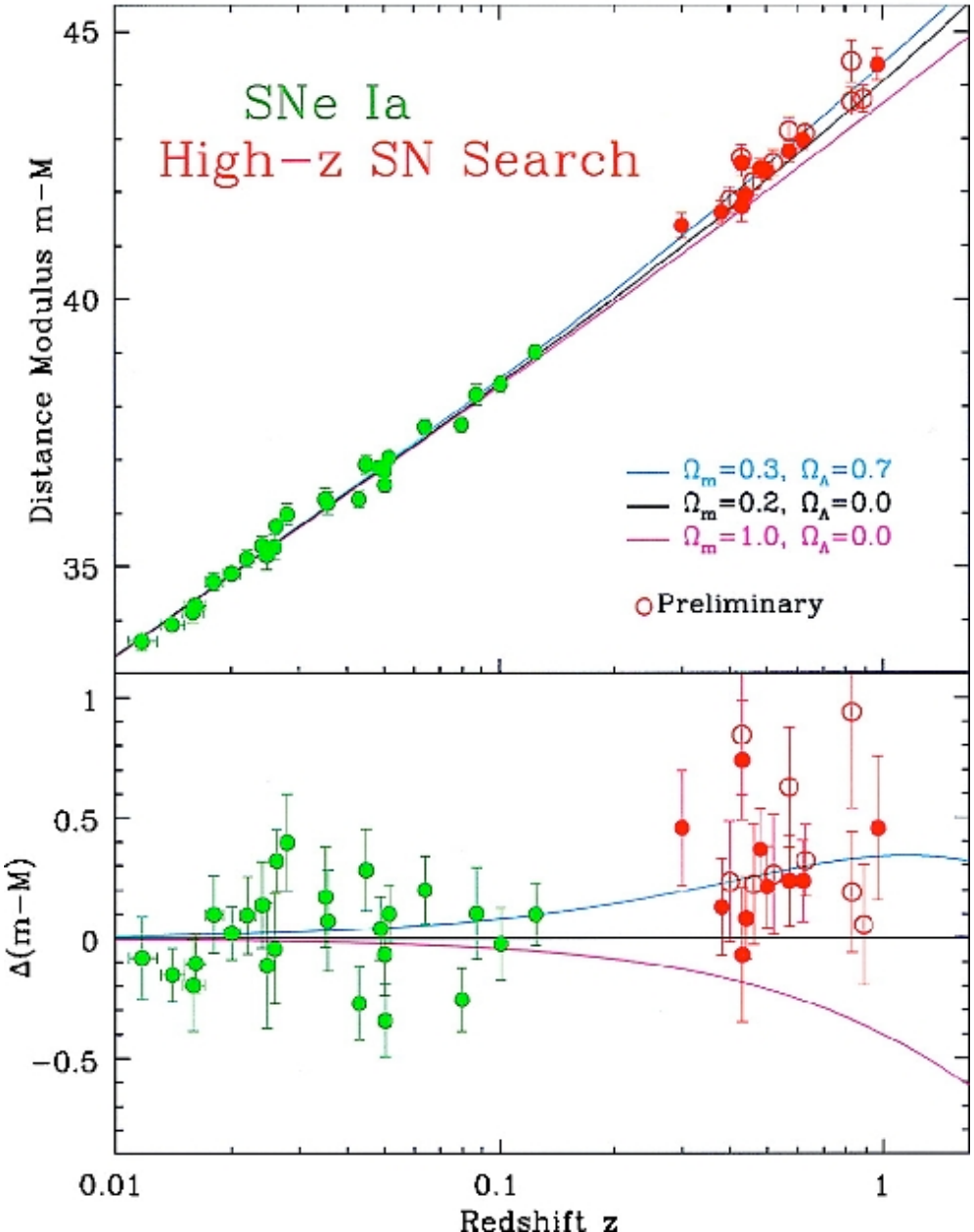
Typical speed-ups: factor 4.

Supernova distances

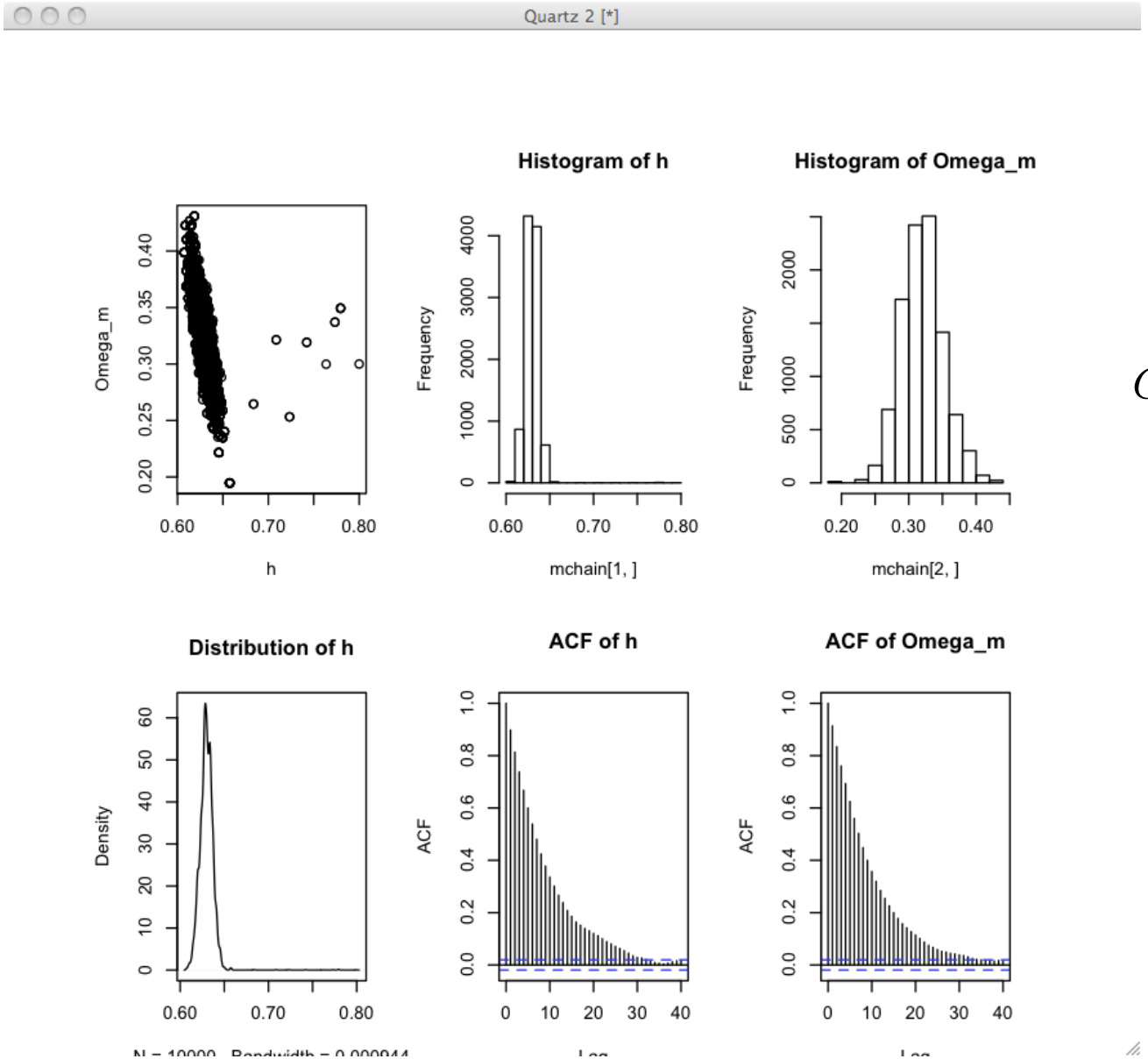
- Luminosity Distance depends on Ω_m and H_0 (for flat Universe)

$$f = \frac{L}{4\pi D_L^2}$$

$$D_L = \frac{c(1+z)}{H_0} \int_0^z \frac{dz'}{\sqrt{\Omega_m (1+z')^3 + 1 - \Omega_m}}$$



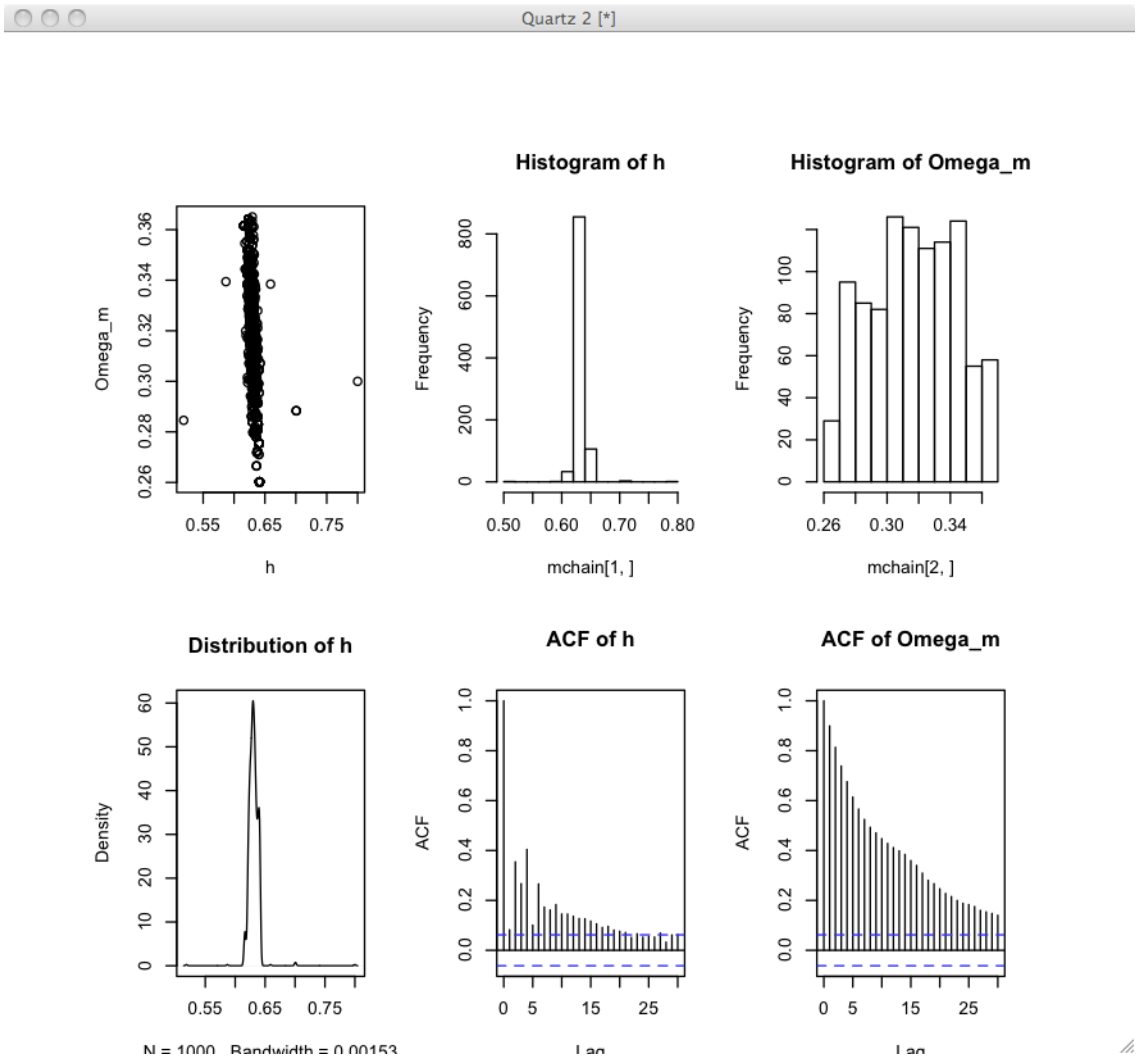
MCMC run



$$C = \begin{pmatrix} 7.1 \times 10^{-5} & -1.9 \times 10^{-4} \\ -1.9 \times 10^{-4} & 1.0 \times 10^{-3} \end{pmatrix}$$

Acceptance rate 0.15

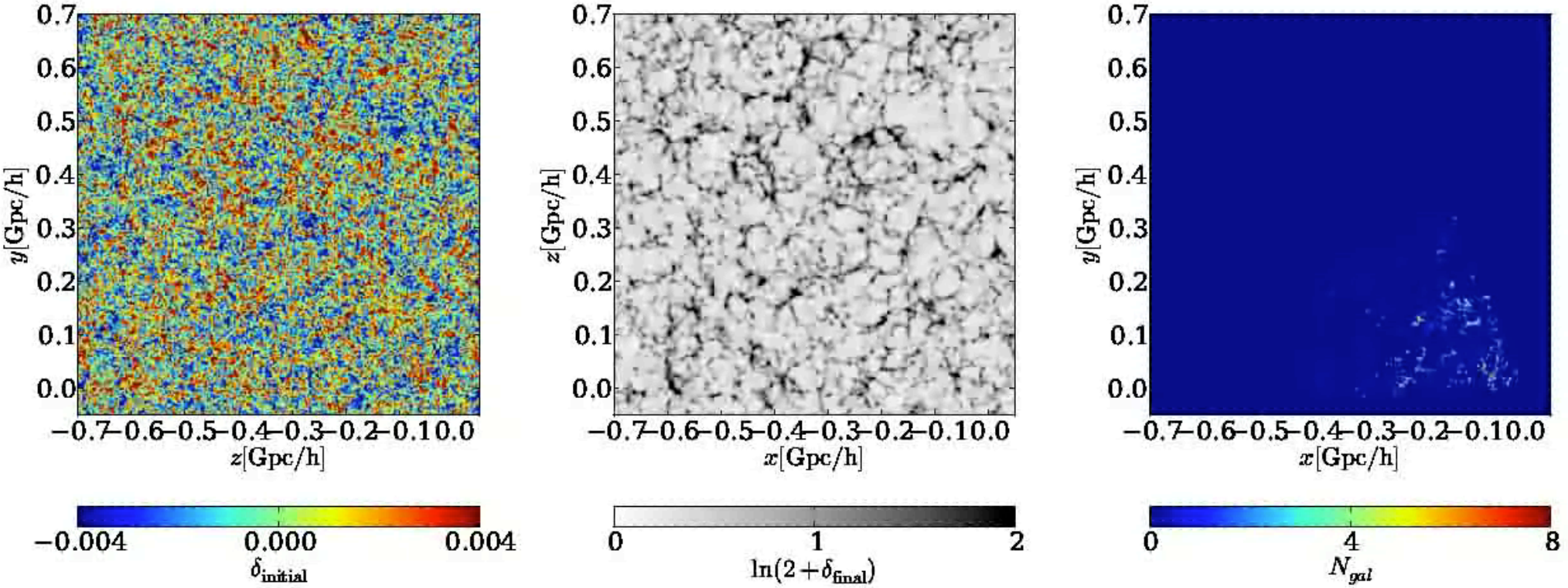
HMC



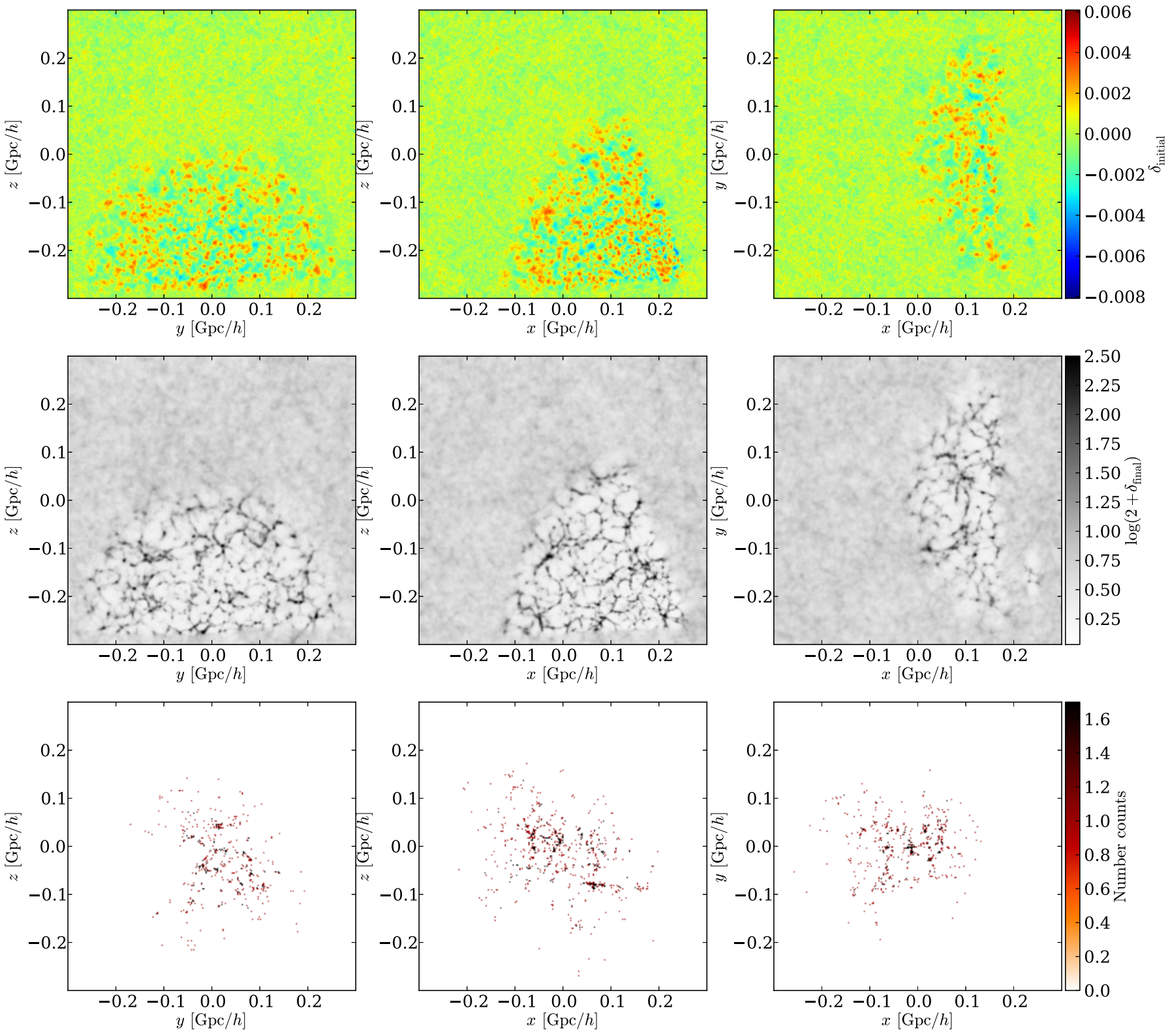
Acceptance rate 0.4

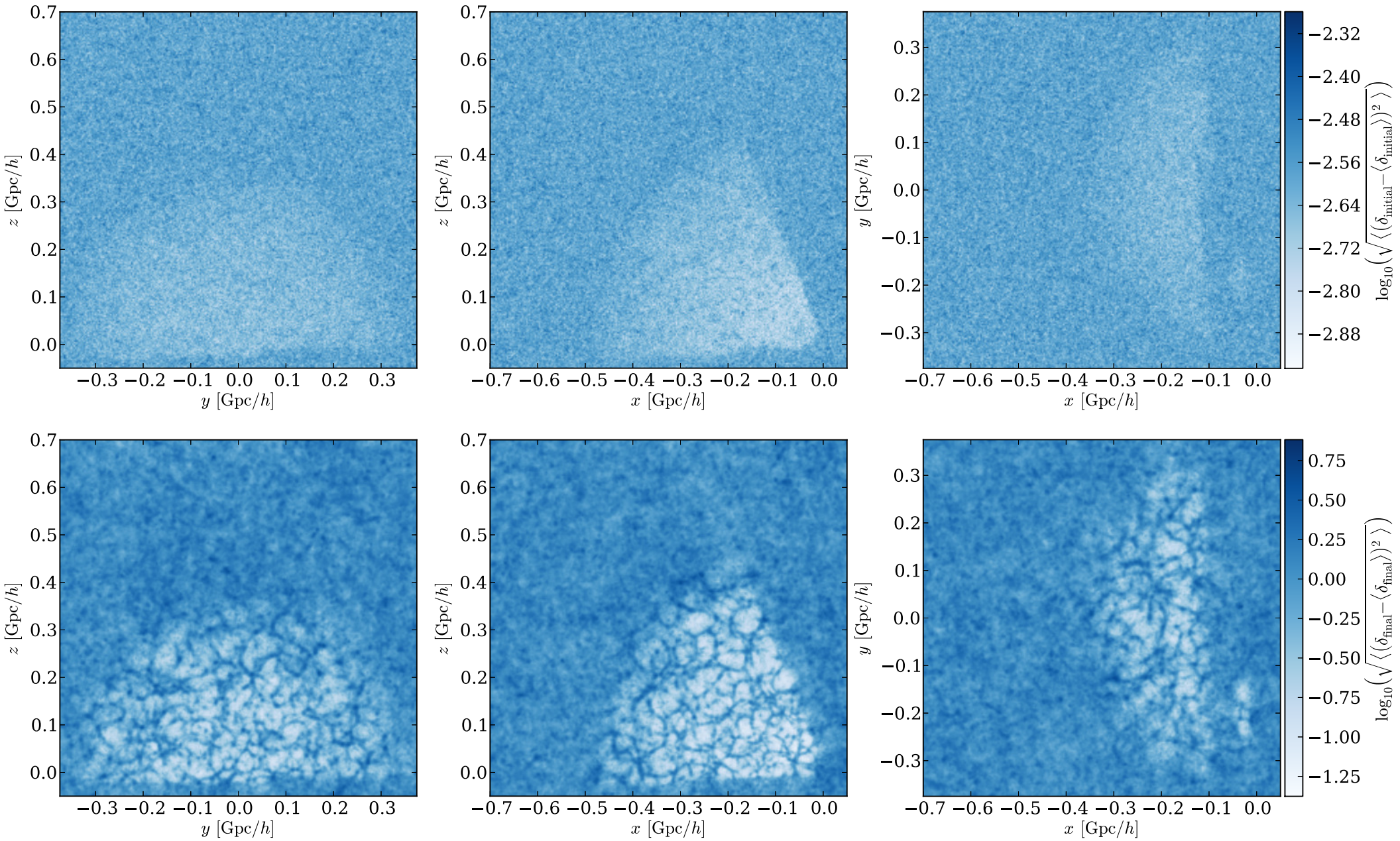
Still work to do…

HMC with millions of parameters



J. Jasche


z [Gpc/h]
y [Gpc/h]
x [Gpc/h]
δinitial
log(2+δfinal)
Number counts


z [Gpc/h]
y [Gpc/h]
x [Gpc/h]
$\log_{10}\left(\sqrt{\langle(\delta_{\rm initial}-\langle\delta_{\rm initial}\rangle)^2\rangle}\right)$
$\log_{10}\left(\sqrt{\langle(\delta_{\rm final}-\langle\delta_{\rm final}\rangle)^2\rangle}\right)$