Imperial College London

Beyond Metropolis Sampling & Generalised Linear Models

Andrew Jaffe

ICIC Workshop 2016

WINTON

Science & Technology Facilities Council

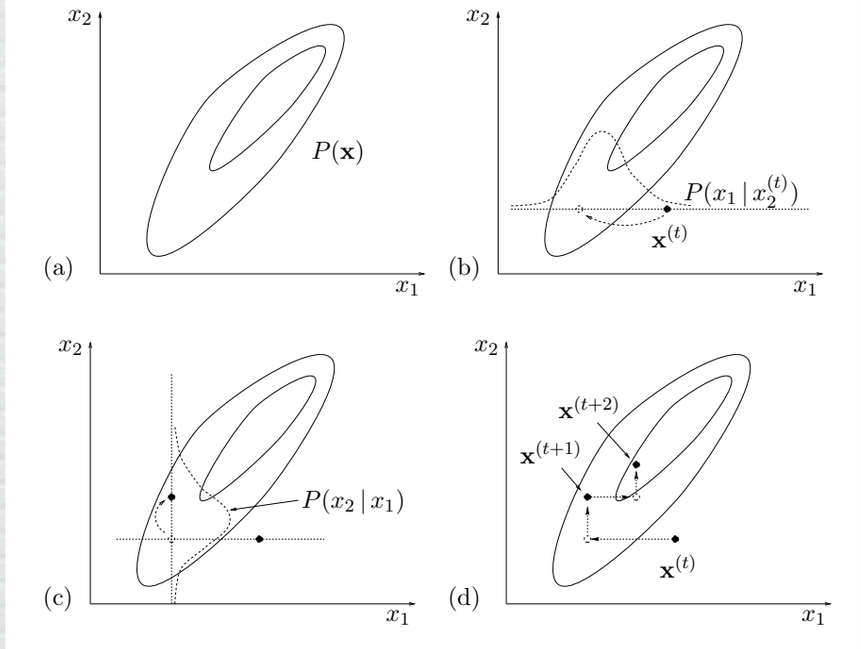
ICIC Imperial Centre for Inference & Cosmology

Sampling beyond MCMC

- Simple MCMC is a good general tool, but
  - curse of dimensionality
  - requires tuning — e.g., proposal distributions
  - inefficient
- Other sampling techniques exist
  - usually for cases when you have more information about the distributions
  - Gibbs sampling — need to have the conditional probabilities for different parameters, $\mathrm{P}(\theta_1|\theta_2,d)$
  - Hamiltonian Monte Carlo — need derivatives $\partial\mathrm{P}(\theta)/\partial\theta$

Gibbs Sampling

- Metropolis-Hastings with Proposal = conditional dist'n
  - all samples accepted
  - satisfies detailed balance
  - no adjustable parameters in the algorithm
- suited to hierarchical models (often written in terms of the conditionals)
- Algorithm:
  - $x_1^{(n+1)} \sim \mathrm{P}(x_2^{(n)}, x_3^{(n)}, \ldots)$
    $x_2^{(n+1)} \sim \mathrm{P}(x_1^{(n+1)}, x_3^{(n)}, \ldots)$
    $x_3^{(n+1)} \sim \mathrm{P}(x_1^{(n+1)}, x_2^{(n+1)}, \ldots)$
  - Should change (reverse/randomize) the order 1, 2, 3,… in successive steps
- Caveats: can fail badly if the distribution isn't aligned with the axes and/or highly curved
- *Otherwise often use "metropolis-within-Gibbs"

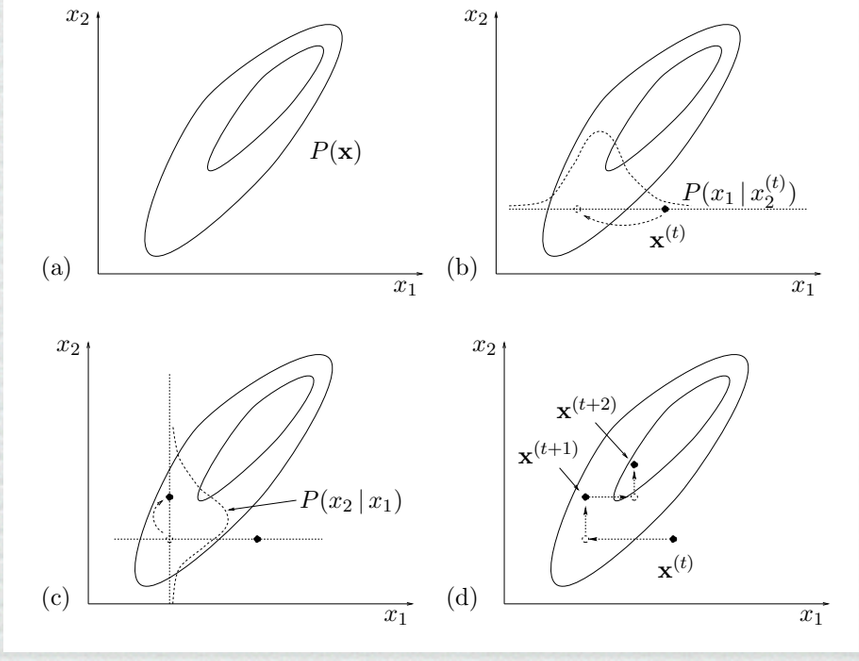


McKay, *Information Theory…*

Especially good if these can be "analytically" sampled*

Gibbs Sampling

- Algorithm:
  - $x_1^{(n+1)} \sim \mathrm{P}(x_2^{(n)}, x_3^{(n)}, \ldots)$
    $x_2^{(n+1)} \sim \mathrm{P}(x_1^{(n+1)}, x_3^{(n)}, \ldots)$
    $x_3^{(n+1)} \sim \mathrm{P}(x_1^{(n+1)}, x_2^{(n+1)}, \ldots)$
- Note that conditionals are just the full distribution with the other parameters held fixed (up to normalization).
- In a hierarchical model, get the full posterior by multiplying out all the distributions that appear
  - See Alan Heavens' talk tomorrow…



McKay, *Information Theory…*

Hamiltonian Monte Carlo (HMC)

- *(aka Hybrid Monte Carlo; Duane et al 1987)*
- Analogy with dynamical systems, which explore (***position, momentum***) phase space over time
  - Potential $U(\theta_i) = -\ln \mathrm{P}(\theta_i)$ w/ "positions" θ_i
  - KE $K(u_i) = \frac{1}{2}\boldsymbol{u}\cdot\boldsymbol{u}$ w/ "momenta" $u_i \sim \mathrm{N}(0,\sigma^2)$
  - Hamiltonian $H(\theta_i, u_i) = U(\theta_i) + K(u_i)$
  - Density $\mathrm{P}(\theta_i, u_i) = e^{-H(\theta, u)}$
    - 2N parameters!
  - Evolve as dynamical system
    - ignore (marginalize over) momenta

$$\dot{\theta}_i = \frac{\partial H}{\partial u_i} = u_i$$

$$\dot{u}_i = -\frac{\partial H}{\partial \theta_i} = \frac{\partial \ln P}{\partial \theta_i}$$

- Need to discretize the system (time derivatives)
- Values of (θ_i, u_i) at different times: proposed MC samples
- If exact dynamics, H conserved, $\Longrightarrow$ all samples accepted

$$\dot{\theta}_i = \frac{\partial H}{\partial u_i} = u_i$$

$$\dot{u}_i = -\frac{\partial H}{\partial \theta_i} = \frac{\partial \ln P}{\partial \theta_i}$$

- in practice, approximate evolution (and, e.g., numerical derivatives)
- so, accept $(\theta_i, u_i)^*$ as step $n{+}1$ with probability

$$\min\left[1, \exp\left(-H^* + H^{(n)}\right)\right]$$

HMC Algorithm (1)

- Algorithm (Hajian *PRD*75 083525, 2007)

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 $\alpha \sim \text{Uniform}(0, 1)$
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

Only propose every N timesteps

Discretisation step!

HMC Algorithm (2)

- R version (Neal, in *Handbook of MCMC*)

```
HMC = function (U, grad_U, epsilon, L, current_q)
{
  q = current_q
  p = rnorm(length(q),0,1)  # independent standard normal variates
  current_p = p

  # Make a half step for momentum at the beginning
  p = p - epsilon * grad_U(q) / 2

  # Alternate full steps for position and momentum

  for (i in 1:L)
  {
    # Make a full step for the position
    q = q + epsilon * p
    # Make a full step for the momentum, except at end of trajectory
    if (i!=L) p = p - epsilon * grad_U(q)
  }

  # Make a half step for momentum at the end.
  p = p - epsilon * grad_U(q) / 2
  # Negate momentum at end of trajectory to make the proposal symmetric
  p = -p

  # Evaluate potential and kinetic energies at start and end of trajectory

  current_U = U(current_q)
  current_K = sum(current_p^2) / 2
  proposed_U = U(q)
  proposed_K = sum(p^2) / 2

  # Accept or reject the state at end of trajectory, returning either
  # the position at the end of the trajectory or the initial position

  if (runif(1) < exp(current_U-proposed_U+current_K-proposed_K))
  {
    return (q)  # accept
  }
  else
  {
    return (current_q)  # reject
  }
}
```

Single L-step trajectory

} Leapfrog method

HMC vs Metropolis-Hastings

Two trajectories

Four trajectories

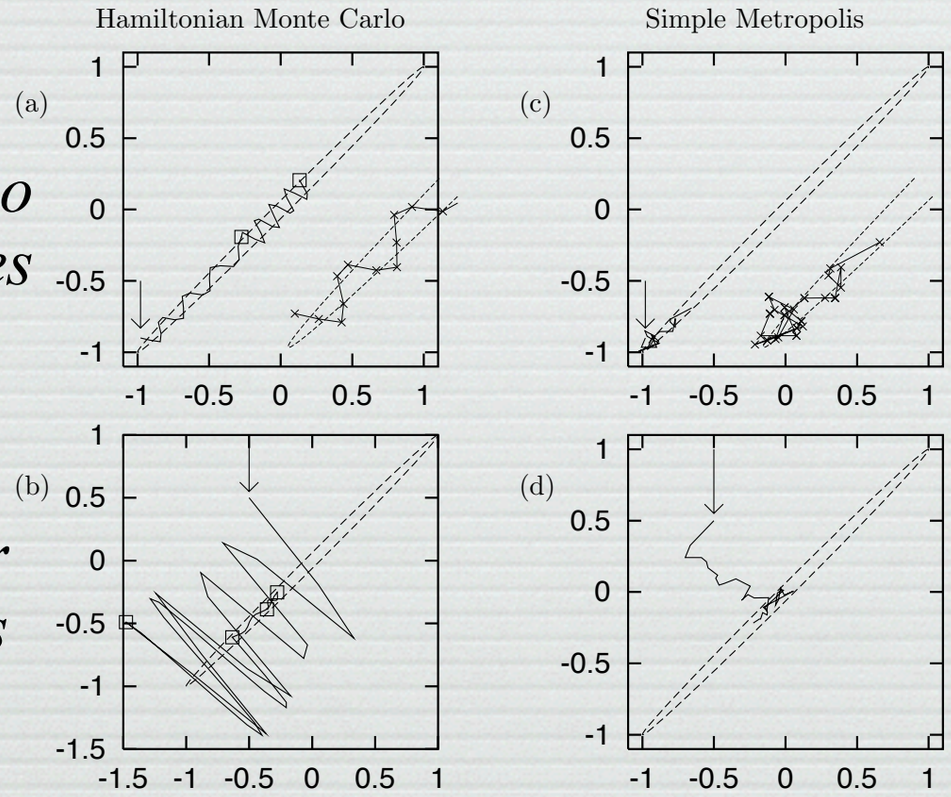


Figure 30.2. (a,b) Hamiltonian Monte Carlo used to generate samples from a bivariate Gaussian with correlation $\rho = 0.998$. (c,d) For comparison, a simple random-walk Metropolis method, given equal computer time.

MacKay,
Information Theory…

Neil,
Handbook of MCMC

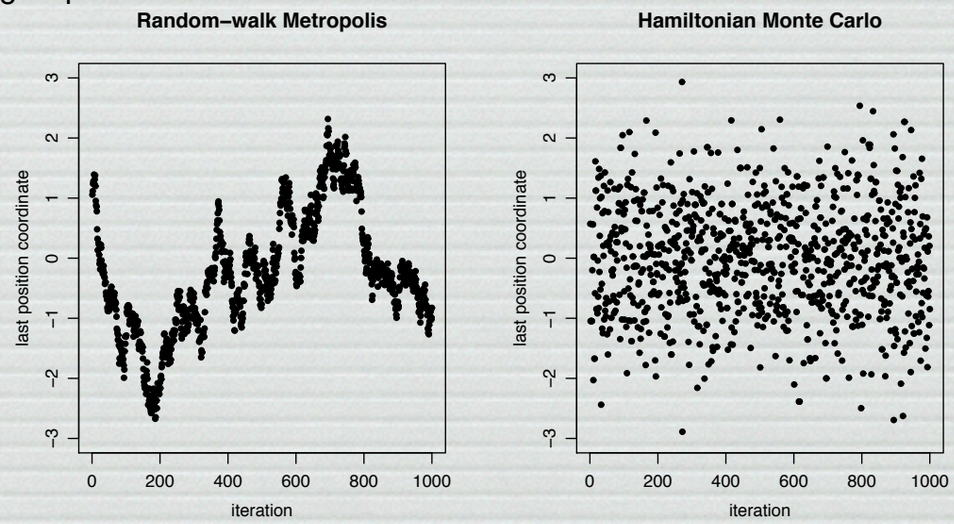
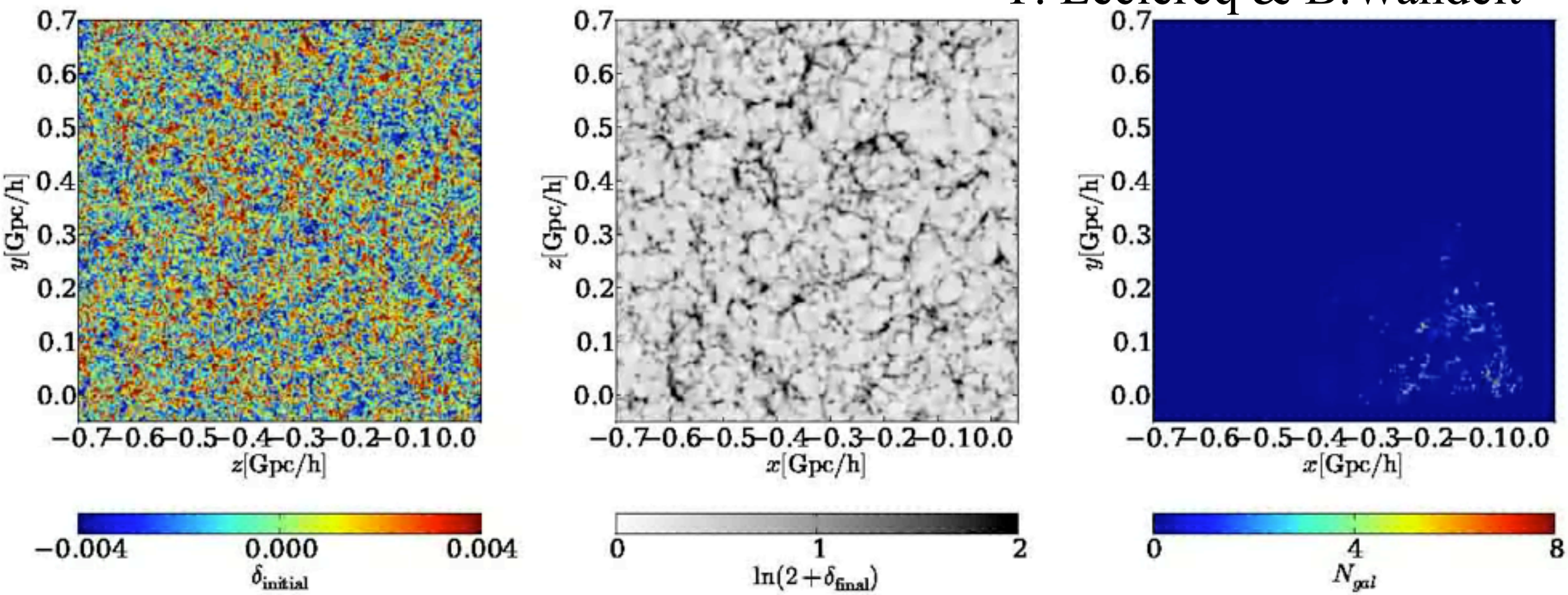


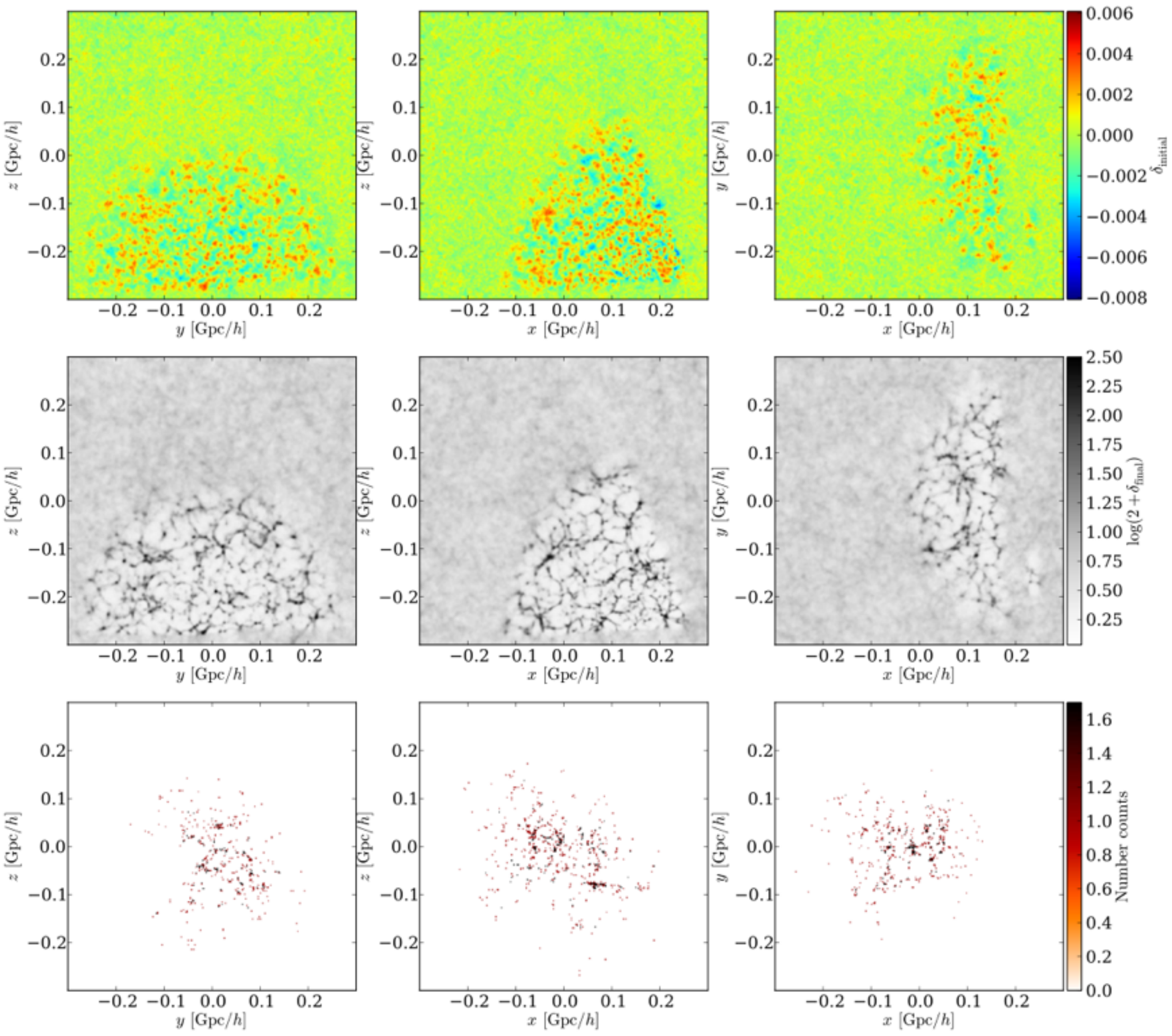
Figure 6: Values for the variable with largest standard deviation for the 100-dimensional example, from a random-walk Metropolis run and an HMC run with $L = 150$. To match computation time, 150 updates were counted as one iteration for random-walk Metropolis.

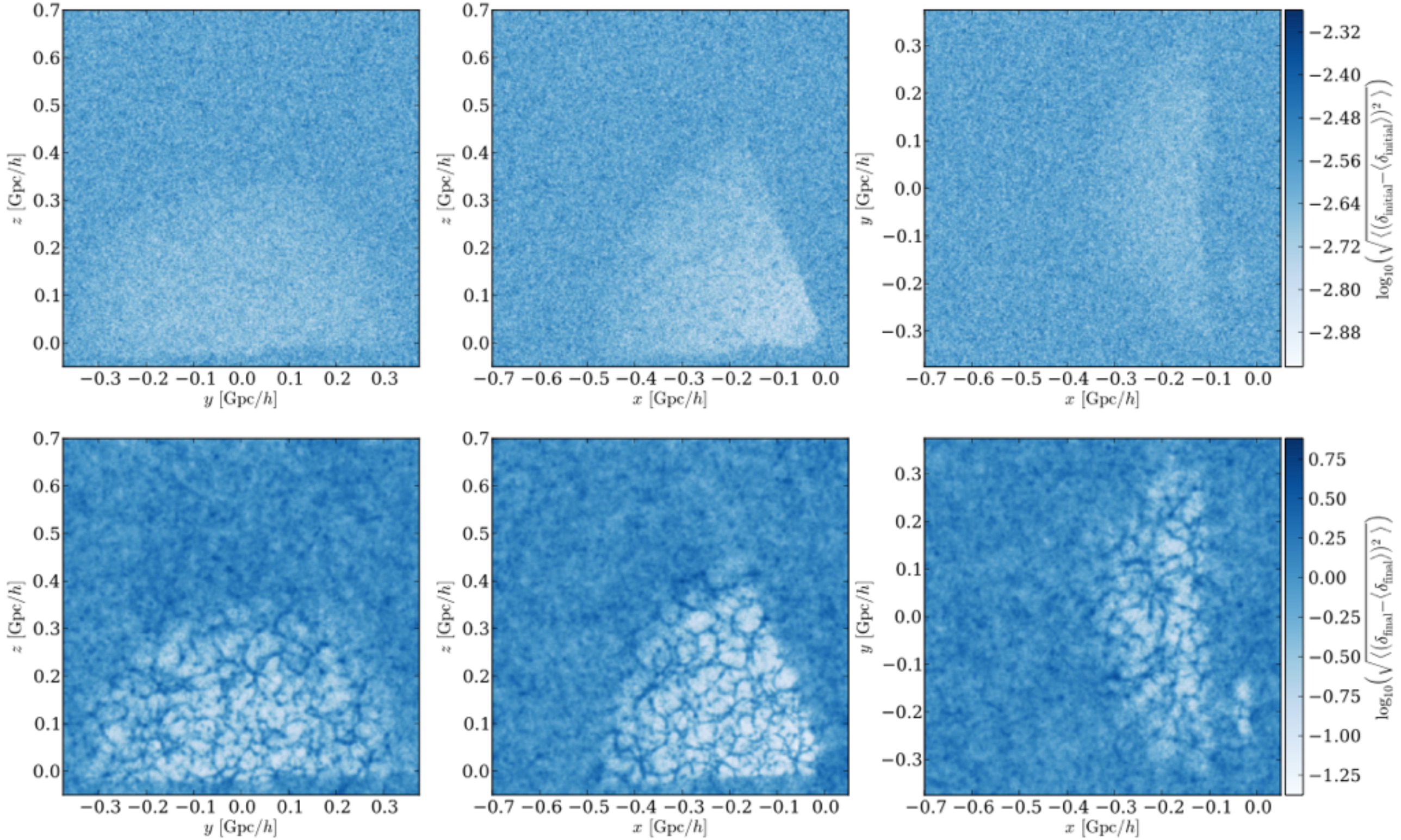
HMC with millions of parameters

- From large-scale structure observations to the *primordial* density field
  - forward physics model from primordial density to observed galaxy distribution
- Related work from Jasche, Lavaux,, Kitaura

F. Leclercq & B.Wandelt

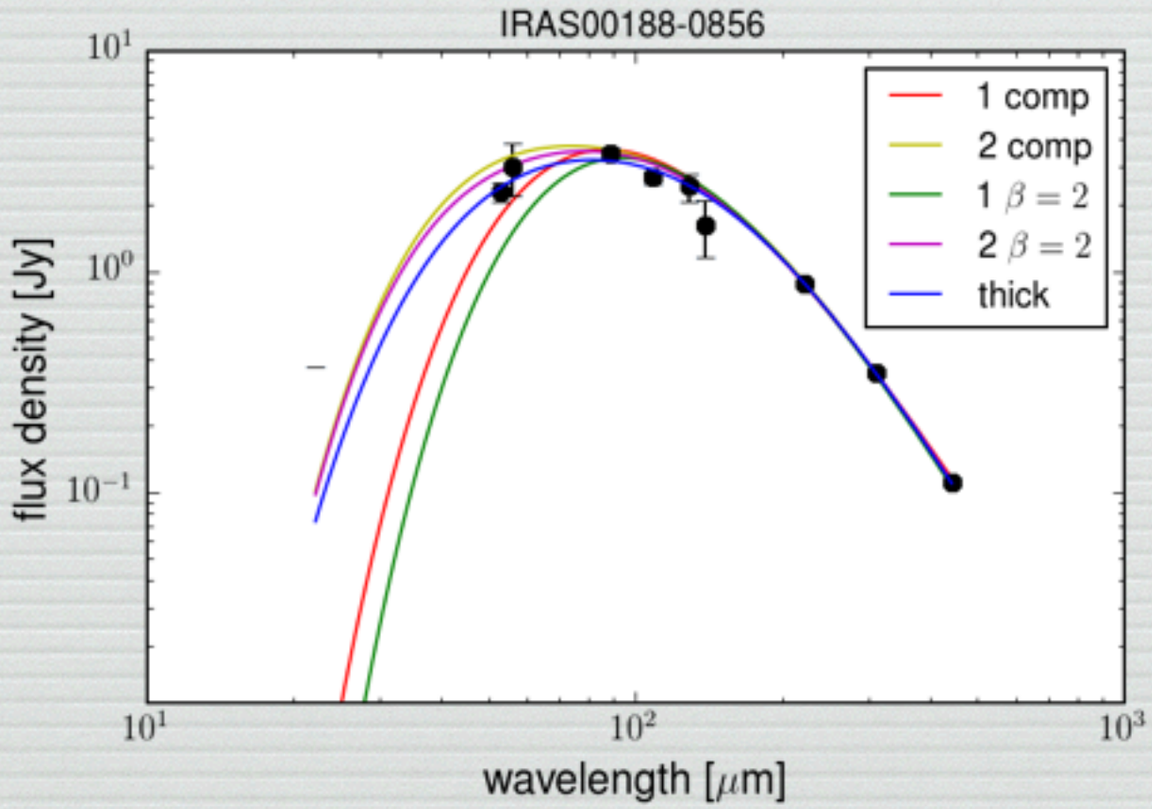


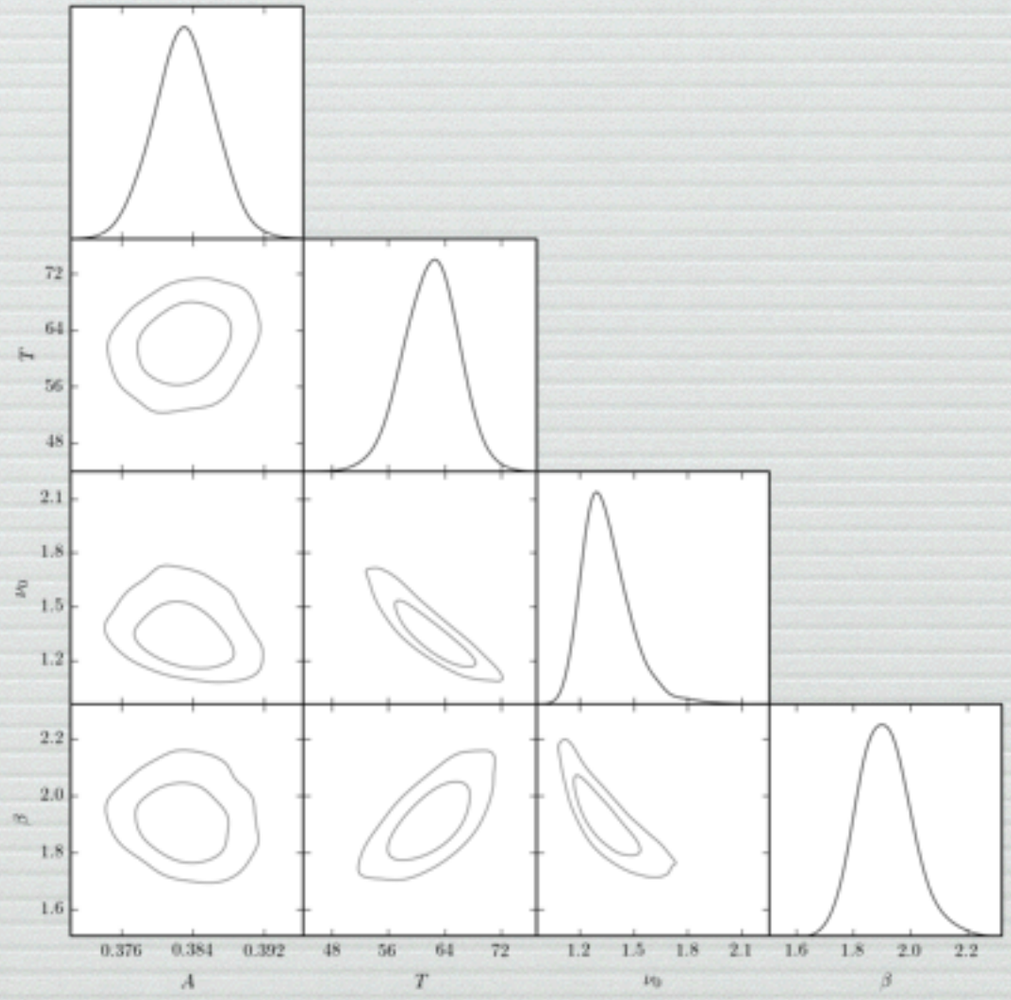

z [Gpc/h]
y [Gpc/h]
x [Gpc/h]
δ_initial
log(2+δ_final)
Number counts


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
z [Gpc/h]
−0.3 −0.2 −0.1 0.0 0.1 0.2 0.3
y [Gpc/h]
−0.7 −0.6 −0.5 −0.4 −0.3 −0.2 −0.1 0.0
x [Gpc/h]
y [Gpc/h]
−2.32
−2.40
−2.48
−2.56
−2.64
−2.72
−2.80
−2.88
$\log_{10}\left(\sqrt{\langle(\delta_{\rm initial}-\langle\delta_{\rm initial}\rangle)^2\rangle}\right)$
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
$\log_{10}\left(\sqrt{\langle(\delta_{\rm final}-\langle\delta_{\rm final}\rangle)^2\rangle}\right)$

HMC as a generic tool

- German et al, STAN (http://mc-stan.org/)
- Uses *automatic differentiation* to get derivatives for ~anything that can be built up from elementary functions
- e.g., SED fitting





Stan Code

```
data {

    int<lower=1> N_comp;    // # of greybody components
                            // (fixed model parameter)

    int<lower=1> N_band;    // number of photometric bands
    vector[N_band] nu_obs; // observed frequency
    vector[N_band] flux;    // observed flux
    vector[N_band] sigma;   // error
    real z;                 // redshift

}


transformed data {

    vector[N_band] nu;      // rest frame frequency
    nu = (1+z)*nu_obs;

}


functions {

    real greybody(real beta, real T, real nu) {
      // greybody, normalized to unit flux at nu=nu_0
        real h_over_k;
        real x;
        real nu_bar;
        real x_bar;

        nu_bar = 1000;

        h_over_k = 0.04799237;        //  K/Ghz
        x = h_over_k * nu / T;
        x_bar = h_over_k * nu_bar / T;
        return (pow(nu/nu_bar, 3+beta) *
               expm1(x_bar) / expm1(x));
    }

}
```

```
parameters {

// nb. N_comp, N_band are data
   vector<lower=0>[N_comp] amplitude;
   positive_ordered[N_comp] T;

// greybody factor
   vector<lower=0, upper=3>[N_comp] beta;

}

model {

    real fluxes[N_band, N_comp];
    vector[N_band] totalflux;

    for (band in 1:N_band) {
        for (comp in 1:N_comp) {  // vectorize over this?
            fluxes[band, comp] = amplitude[comp] *
              greybody(beta[comp], T[comp], nu[band]);
        }
        totalflux[band] = sum(fluxes[band]);
    }

    // try a proper prior on temperature;
    //    needed since ordered vectors don't have limits
    T ~ uniform(3,100);
    flux ~ normal(totalflux, sigma);

}
```

Linear models

- Very generic: $d(t_i) = \sum_p x_p f_p(t_i) + n_i$
  - (doesn't need to be function of time)

- Many simplifications if we assume Gaussian for n_i
  - Can do all integrals and derivatives analytically
  - Everything becomes linear algebra
  - Tends to agree with frequentist statistics

The Gaussian Distribution

$$P(x_i|\mu_i M_{ij} I) = \frac{1}{|2\pi M|^{1/2}} \exp\left[-\frac{1}{2}(x_i - \mu_i) M_{ij} (x_j - \mu_j)\right]$$

$$P(x|\mu\sigma I) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right]$$

$$\langle x_i \rangle = \mu_i \qquad \langle (x_i - \mu_i)(x_j - \mu_j) \rangle = M_{ij}$$

$$\langle x \rangle = \mu \qquad \langle (x-\mu)^2 \rangle = \sigma^2$$

- Moments:
  - all higher cumulants $\kappa_n = 0$
- Central Limit Theorem
  - Arises very often: sum of many independent "random variables" tends to Gaussian
    - Additive noise is often well-described as Gaussian
- Maximum Entropy
  - Bayesian interpretation: if you know only the mean and variance, Gaussian is the "least informative" consistent distribution.

Inference from a Gaussian: Averaging

- The simplest "linear model"
- Consider *data* = *signal* + *noise*,
- $d_i = s + n_i$ for data points i=1…N
  - Noise, n_i, has zero mean, known variance σ^2
    - Assign a Gaussian to $(d_i - s)$
      - Alternately: keep n_i as a parameter and marginalize over it with $p(d_i|n_i\, s\, I) = \delta(d_i\text{-}n_i\text{-}s)$
  - Prior for s (*i*.e., a and b)?
    - To be careful of limits, use Gaussian with width Σ, take $\Sigma \to \infty$ at end of calculation
      - Same answer with unifom dist'n in $(-\Sigma_1,\Sigma_2) \to (-\infty,\infty)$

→calculation

Inference from a Gaussian: Averaging

- Posterior:

$$P(s|dI) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \exp\left[-\frac{1}{2}\frac{(s-\bar{d})^2}{\sigma_b^2}\right]$$

  - best estimate of signal is average ± stdev:
    - $s = \bar{d} \pm \sigma_b = \bar{d} \pm \sigma/\sqrt{N}$
  - What if we don't know σ? try Jefferys $P(\sigma|I) \propto 1/\sigma$
    - marginalized $P(s|I) \propto [s^2 - 2s\ \langle d\rangle\ +\ \langle d^2\rangle\ ]^{-1/2}$
    - Student t or Cauchy distribution
    - (very broad distribution!)

Inference from a Gaussian: Straight-line fitting

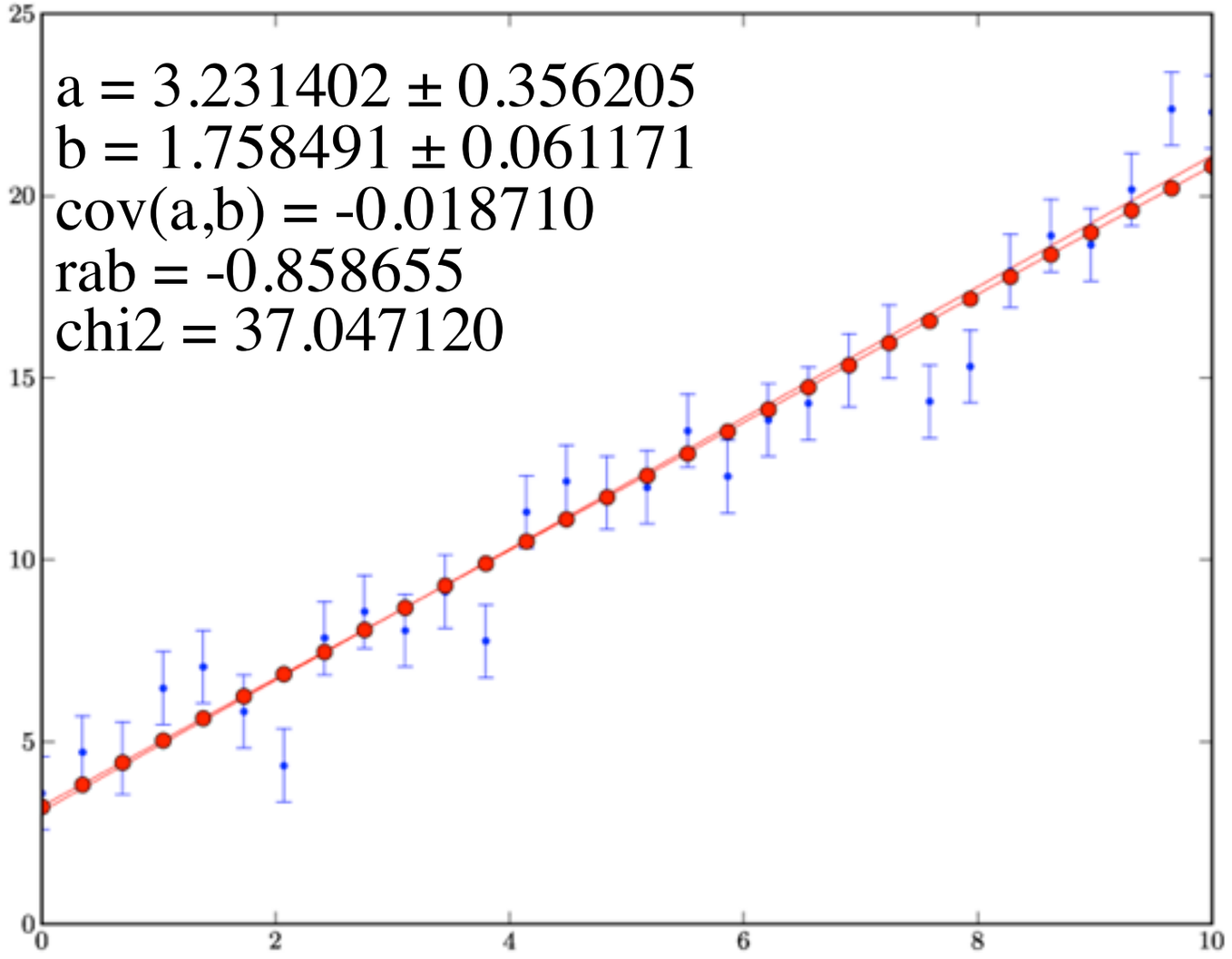
- Now consider *data* = *signal* + *noise*, where signal depends linearly on time:
  - $d_i = at_i + b + n_i$, with "iid" gaussian noise $\langle n_i \rangle = 0$; $\langle n_i^2 \rangle = \sigma^2$
- Likelihood function is

$$P(d|a,b,I) = \prod_i \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{1}{2}\frac{(d-at_i-b)^2}{\sigma^2}\right]$$

  - Multivariate gaussian in *d*
  - Linear in (a,b): also has form of a multivariate gaussian in (a,b)
    - but not a *distribution* in (a,b) until you apply Bayes' theorem and add a *prior*
  - Maximized at the value of the "least squares" est. for (a,b), with the same numerical values for the errors (& covariance)
    - (but, recall, with a very different interpretation of those errors)

Inference from a Gaussian: Straight-line fitting

- This means that for these problems you can just use usual canned routines...



General linear models (I)

- Consider $d(t_i) = \sum_p x_p f_p(t_i) + n_i$
  i.e., a sum of known functions with unknown amplitudes, plus noise — want to estimate a_p
  - e.g., linear fit: $f_0(t)=1, f_1(t)=t$
- assume zero-mean Gaussian noise, possibly correlated: $\langle n \rangle = 0$, $\langle n_i\, n_j \rangle = \mathbf{N}_{ij}$
  - typically, noise is stationary (isotropic): $\mathbf{N}_{ij}=N(t_i-t_j)$
- rewrite in matrix-vector form:

$$d_i = \sum_p A_{ip} x_p + n_i \qquad \text{with } A_{ip} = f_p(t_i)$$

- Likelihood:

$$P(d_i|x_p I) = \frac{1}{|2\pi N|^{1/2}} \exp\left[-\frac{1}{2}(d-Ax)^T N^{-1}(d-Ax)\right]$$

General linear models (II)

$$d_i = \sum_p A_{ip} x_p + n_i \qquad \text{with } A_{ip} = f_p(t_i)$$

complete the square

- Can rewrite the likelihood as

$$P(d_i|x_p I) \propto \exp\left[-\frac{1}{2}(d - A\bar{x})^T N^{-1}(d - A\bar{x})\right] \times \exp\left[-\frac{1}{2}(x - \bar{x})^T C^{-1}(x - \bar{x})\right]$$

$$\propto \exp\left[-\frac{1}{2}(d - AWd)^T N^{-1}(d - AWd)\right] \times \exp\left[-\frac{1}{2}(x - Wd)^T C^{-1}(x - Wd)\right]$$

depends on data, not params | *depends on data and params*

- with $W = (A^T N^{-1} A)^{-1} A^T N^{-1}$ and $C = (A^T N^{-1} A)^{-1}$
- Parameter-independent factor is just $e^{-\chi^2_{\rm max}}$
- Parameter-dependent factor shows that likelihood is multivariate Gaussian with mean

$$\bar{x} = Wx = (A^T N^{-1} A)^{-1} A^T N^{-1} d$$

and variance C

General linear models (III)

- In limit of an infinitely wide uniform (or Gaussian) prior on x:

$$P(x_p|dI) = \frac{1}{|2\pi C|^{1/2}} \exp\left[-\frac{1}{2}(x - Wd)^T C^{-1}(x - Wd)\right]$$

nb. normalization cancels out $e^{-\chi^2_{\rm max}}$

- Covariance matrix $\langle \delta x_p \, \delta x_q \rangle = C_{pq}$ gives error $\sigma_p^2 = C_{pp}$ if we *marginalize* all other parameters.
- Inverse covariance gives error $\sigma_p^2 = 1/C^{-1}_{pp}$ if we *fix* other parameters
  - nb. marginalization doesn't move mean (max) values *for this case*
  - cf. Fisher matrix $F \Leftrightarrow C^{-1}$

Spectrum (variance) estimation

- Gaussian prior on signal, $\langle x_p \rangle = 0$, $\langle x_p x_{p'} \rangle = S_{pp'}$
- e.g., for CMB

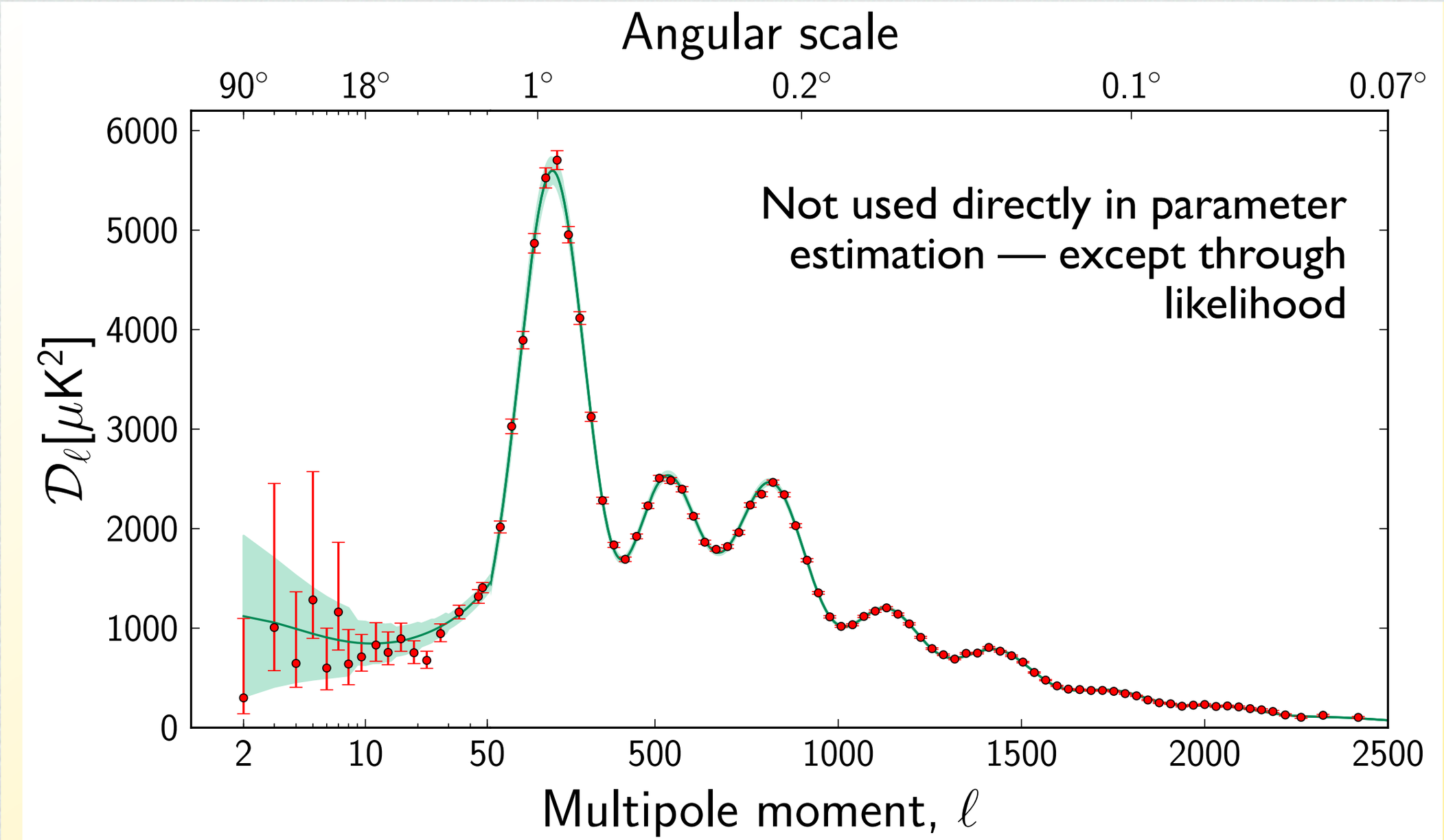
$$S_{pp'} = \sum_\ell \frac{2\ell+1}{4\pi} C_\ell B_\ell^2 P_\ell(\hat{x}_p \cdot \hat{x}_{p'})$$

- Can now marginalise over the signal, x_p:

$$P(d_p|C_\ell) = \frac{1}{|2\pi(S+N)|} \exp -\frac{1}{2} d^T (S+N)^{-1} d$$

- (i.e., data is multivariate Gaussian with variance given by the sum of signal and noise — as expected)
- In practice, this is very hard to calculate — $O(n^3)$

Example: Planck power spectrum estimates



Error band: cosmic variance estimate

error bars: cosmic + noise variance

ICIC

Wiener filters

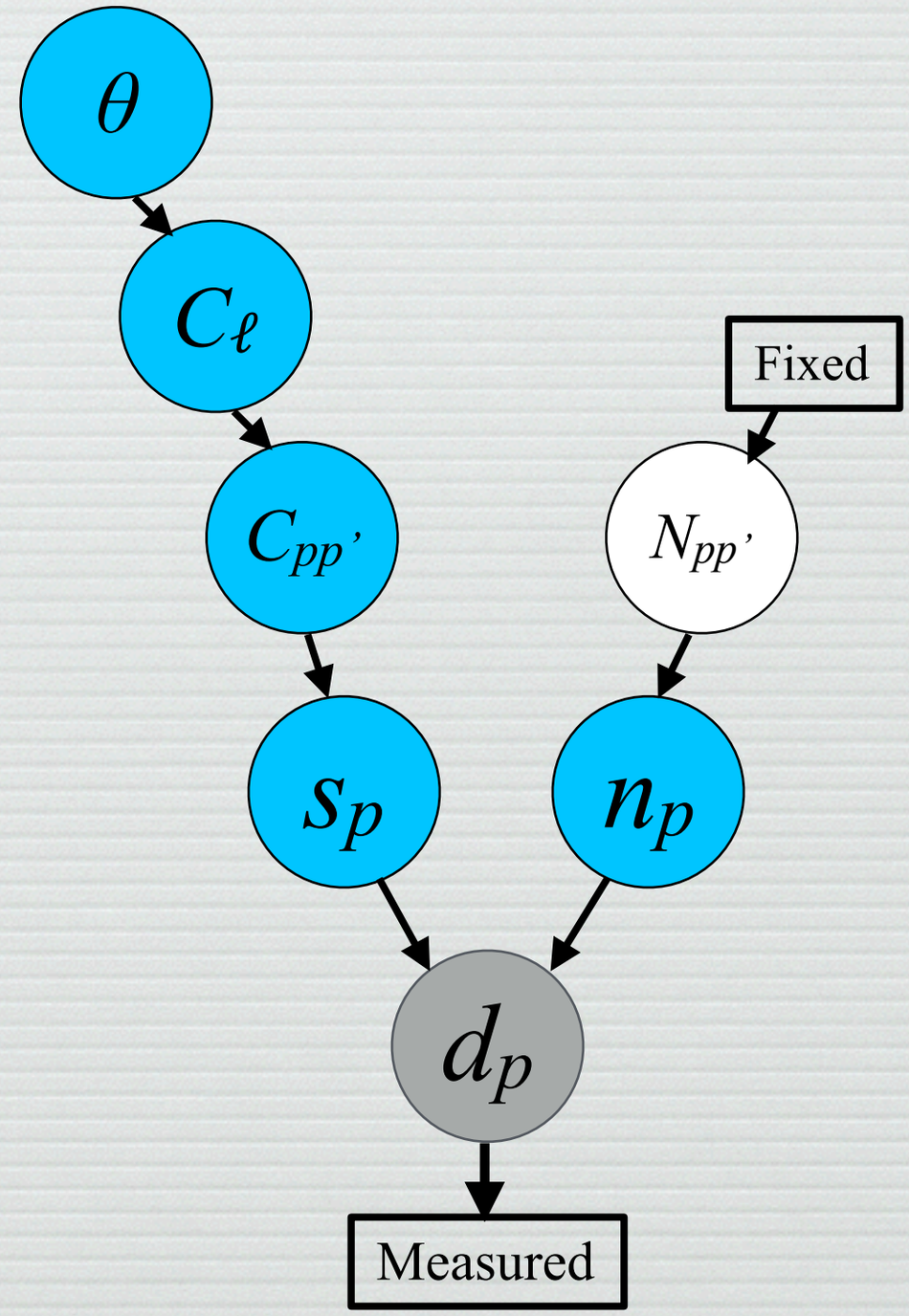
- Gaussian prior on signal, $\langle x_p \rangle = 0$, $\langle x_p x_{p'} \rangle = S_{pp'}$
- Signal and noise variances $S_{pp'}$ and $N_{pp'}$ are known
- Posterior is multivariate Gaussian

$$P(x|SNd) = \frac{1}{|2\pi M|^{1/2}} \exp\left[-\frac{1}{2}(x - Fd)^\dagger M^{-1}(x - Fd)\right]$$

- Wiener filter (mean) $Fd = S(S+N)^{-1}d$
- Wiener variance $M = S(S+N)^{-1}N = S^{-1}+N^{-1}$
- These are easy to write down, but difficult to calculate: naively, $O(n^3)$
  - (S+N) combination

General Linear models and Gibbs sampling

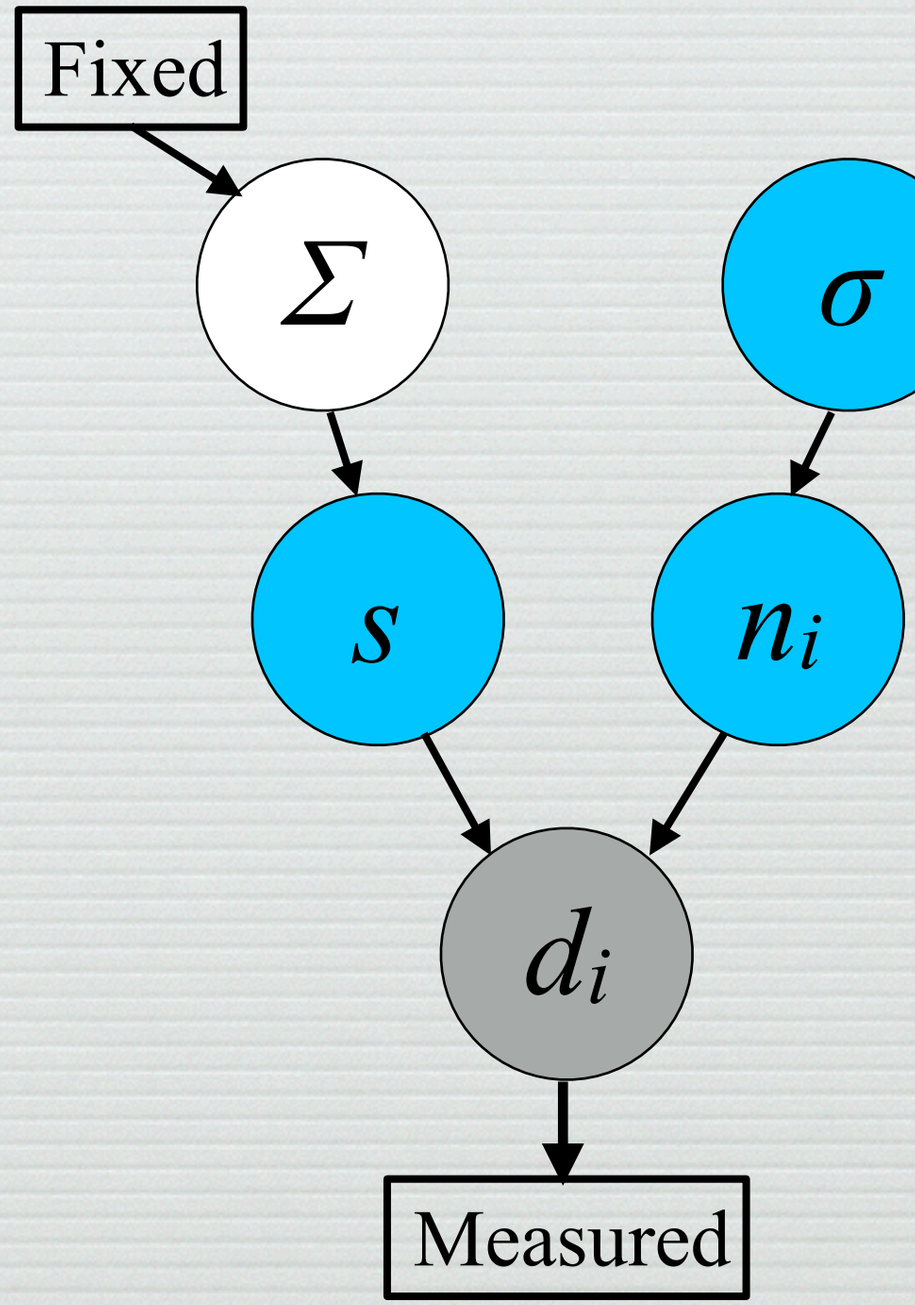
- We can write this whole problem as a ***hierarchical model.***
- Lets us estimate any/all of the parameters
- Each link corresponds to a distribution
  - possibly delta-function, e.g.,
    $C_{pp'} = C_{pp'}(C_\ell)$
    $C_\ell = C_\ell(\theta)$
  - Posterior is the product of all the distributions
  - See Alan Heavens' talk tomorrow



A toy model

- Back to our averaging problem, $d_i = s + n_i$
- P(n_i|I) = Gaussian w/ $\langle n_i \rangle = 0$, $\langle n^2 \rangle = \sigma^2$
- P(s|I) = Gaussian w/ $\langle s \rangle = 0$, $\langle s^2 \rangle = \Sigma^2$
  - Toy version of measuring a power spectrum
- But, now, take σ^2 unknown with, e.g., $P(\sigma^2) \propto 1/\sigma^v$ (improper…)

- Hierarchical model:



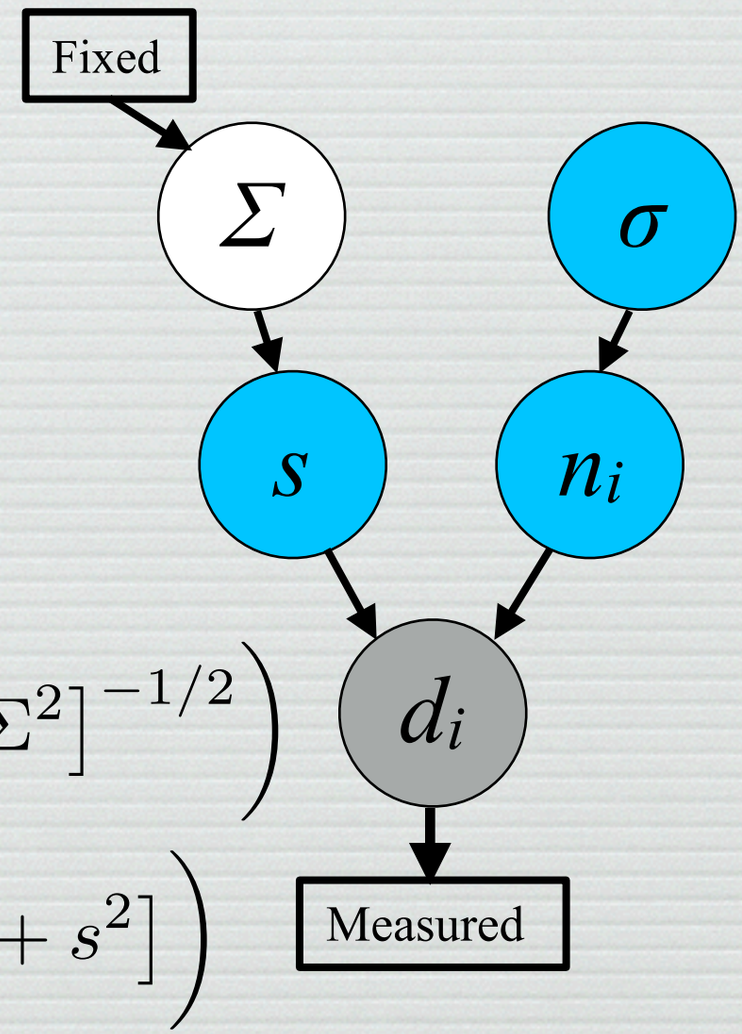
A toy model

- Back to our averaging problem, $d_i = s + n_i$

$$P(\mu, \sigma|d) = \frac{1}{\sigma^\nu} \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left[-\frac{n}{2\sigma^2}\left(\bar{d}^2 - 2s\bar{d} + s^2\right)\right] \frac{1}{(2\pi\Sigma^2)^{1/2}} \exp\left[-\frac{1}{2\Sigma^2}s^2\right]$$

$$\propto \exp\left[-\frac{1}{2}\frac{\left(s - \frac{n/\sigma^2}{n/\sigma^2+1/\Sigma^2}\bar{d}\right)^2}{\left(n/\sigma^2 + 1/\Sigma^2\right)^{-1}}\right]$$

  - Unknown noise variance σ^2, Gaussian prior on s
  - Posterior is Gaussian in s (Wiener), Gamma in $1/\sigma^2$
  - Conditionals are known for Gibbs.



- Algorithm:

$$s|\sigma, d \leftarrow \text{Normal}\left(\frac{n/\sigma^2}{n/\sigma^2+1/\Sigma^2}\bar{d}, \left[n/\sigma^2 + 1/\Sigma^2\right]^{-1/2}\right)$$

$$\sigma|s, d \leftarrow \text{InvGamma}\left(\frac{\nu+n+2}{2}, \frac{n}{2}\left[\bar{d}^2 - 2s\bar{d} + s^2\right]\right)$$

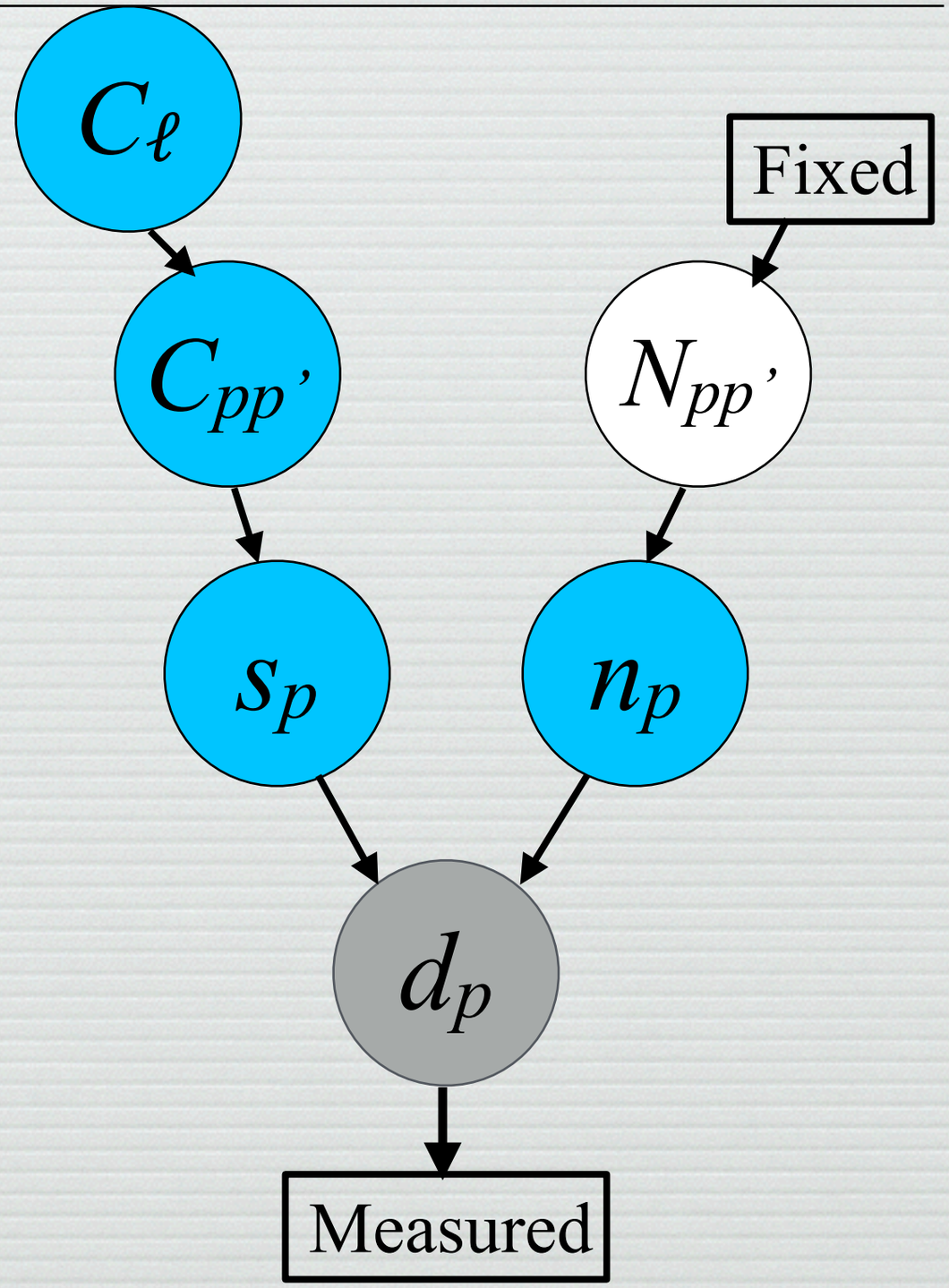
General Linear models and Gibbs sampling

- Posterior P(s|CN) ~ Gaussian in s
- So we know all conditionals: can do Gibbs

- Algorithm:
  - s|CN ⟵ Normal[Wiener mean, wiener variance]
  - C|sN ⟵ InverseWishart[…]

- Problem: Wiener step is $O(n^3)$
  - Signal Covariance is usually sparse in Fourier (Sph. Harm.) basis
  - Noise covariance is usually sparse in pixel basis

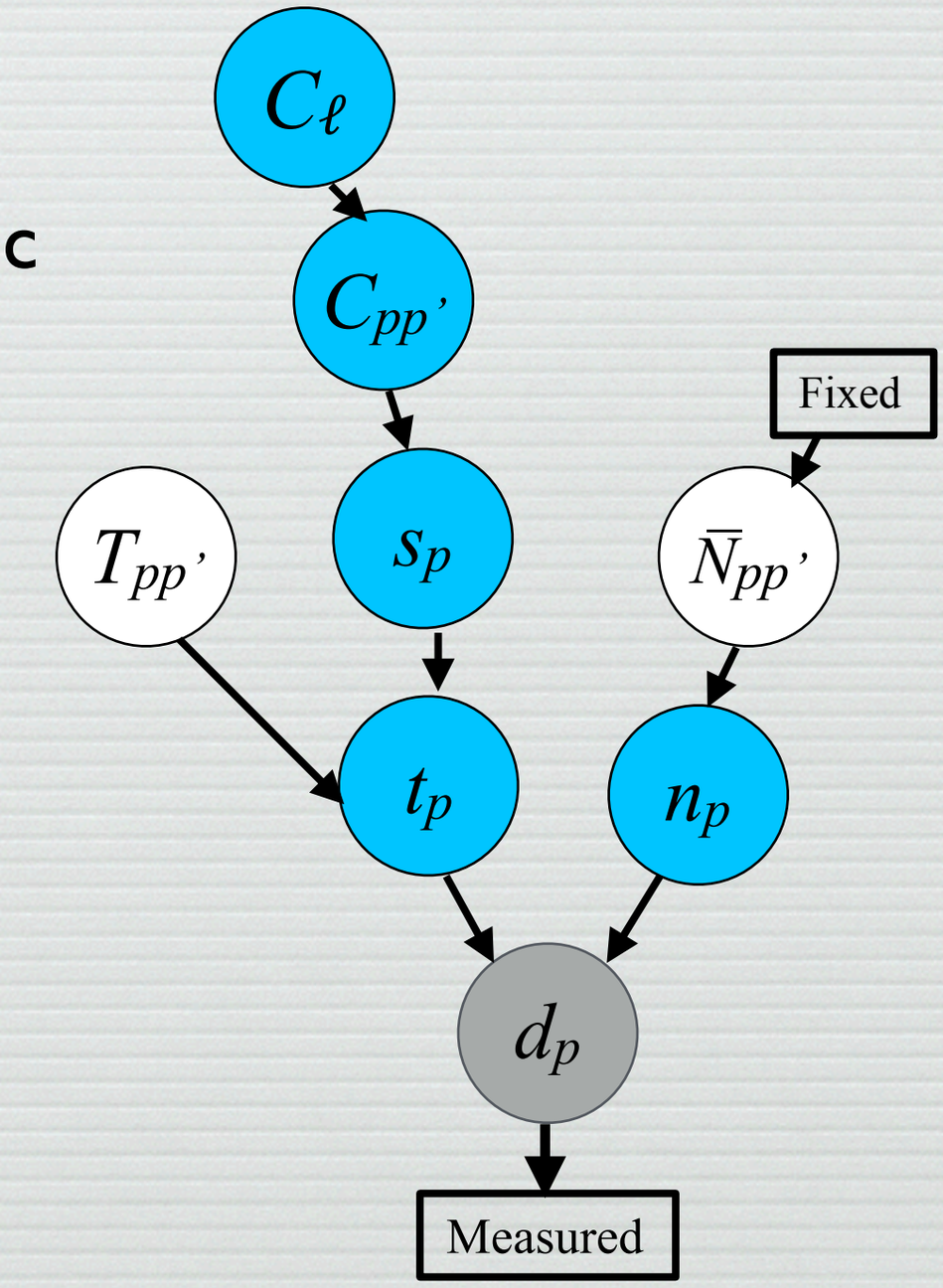
- Solution: "messenger fields"



Messenger Fields and Multivariate Wiener Filtering

- Elsner & Wandelt 2012, 2013
- Jasche & Lavaux 2015

- Add new Gaussian field t with isotropic noise variance $T=\tau I$
  - $P(t|sT) = N(t\text{-}s,T)$
- Anisotropic noise $\bar{N} = N - T$
- Algorithm now requires
  - $t|s$ ← Normal
  - $s|Ct$ ← Normal
  - $C|s$ ← InvWishart
- Where now, the two normal distributions involve $(C^{-1} + T^{-1})$ or $(\bar{N}^{-1} + T^{-1})$



Chi-squared

- The exponential factor of a Gaussian is always of the form $\exp(-\chi^2/2)$
- Likelihood: $\chi^2=\sum(\text{data}_i - \text{model}_i)^2/\sigma_i^2$
- For fixed model, χ^2 has χ^2 distribution for $\nu=N_{\text{data}}-N_{\text{parameters}}$ "degrees of freedom"
  - peaks at $\chi^2=\nu\pm\sqrt{(2\nu)}$
- model may be bad if χ^2 is too big
  - *or* too small ("overfitting" — too many parameters)
- (frequentist argument, but good rule of thumb)