



Max-Planck-Institut für Physik
(Werner-Heisenberg-Institut)

Massively parallel Markov chain Monte Carlo with BAT

Frederik Beaujean, Allen Caldwell – Max Planck Institute for Physics

Daniel Greenwald – Technical University of Munich

Daniel Kollár – CERN

Kevin Kröninger, Shabnaz Pashapour – University of Göttingen



Bayesian inference



Bayes' theorem:

$$P(\vec{\theta}|D, M) = \frac{P(D|\vec{\theta}, M)P(\vec{\theta})}{Z}$$

$\vec{\theta}$: parameters
 D : data
 M : model
 Z : evidence

Use case:

- Allow to phrase arbitrary models for any data set
- Interface to (HEP) software
- Best-fit values
- Marginalized credibility regions



Bayesian inference



Bayes' theorem:

$$P(\vec{\theta}|D, M) = \frac{P(D|\vec{\theta}, M)P(\vec{\theta})}{Z}$$

$\vec{\theta}$: parameters
 D : data
 M : model
 Z : evidence

Use case:

- Allow to phrase arbitrary models for any data set
- Interface to (HEP) software
- Best-fit values
- Marginalized credibility regions

Solution:

- C++ library relying on ROOT.
- Models implemented in (user defined) C++ classes
- Key tool: **MCMC**



BAT availability



- BAT license: GPL, download from <http://mpp.mpg.de/bat>
- Version 0.9 released 2012-05-17



The main goals of a typical data analysis are to

- compare model predictions with data,
- draw conclusions on the validity of a model as a representation of the data, and to
- extract the values of the free parameters of a model.

The **Bayesian Analysis Toolkit**, BAT, is a software package which addresses the points above. It is designed to help solve statistical problems encountered in Bayesian inference. BAT is based on Bayes' Theorem and is realized with the use of Markov Chain Monte Carlo. This gives access to the full posterior probability distribution and enables straightforward parameter estimation, limit setting and uncertainty propagation.

BAT is implemented in C++ and allows for a flexible definition of mathematical models and applications while keeping in mind the reliability and speed requirements of the numerical operations. It provides a set of algorithms for numerical integration, optimization and error propagation. Predefined models exist for standard cases. In addition, methods to judge the goodness-of-fit of a model are implemented. An interface to ROOT allows for further analysis and graphical display of results. BAT can also be run from within RooStats analysis.

Computer Physics Communications
180 (2009) 2197-2209

How does MCMC help?

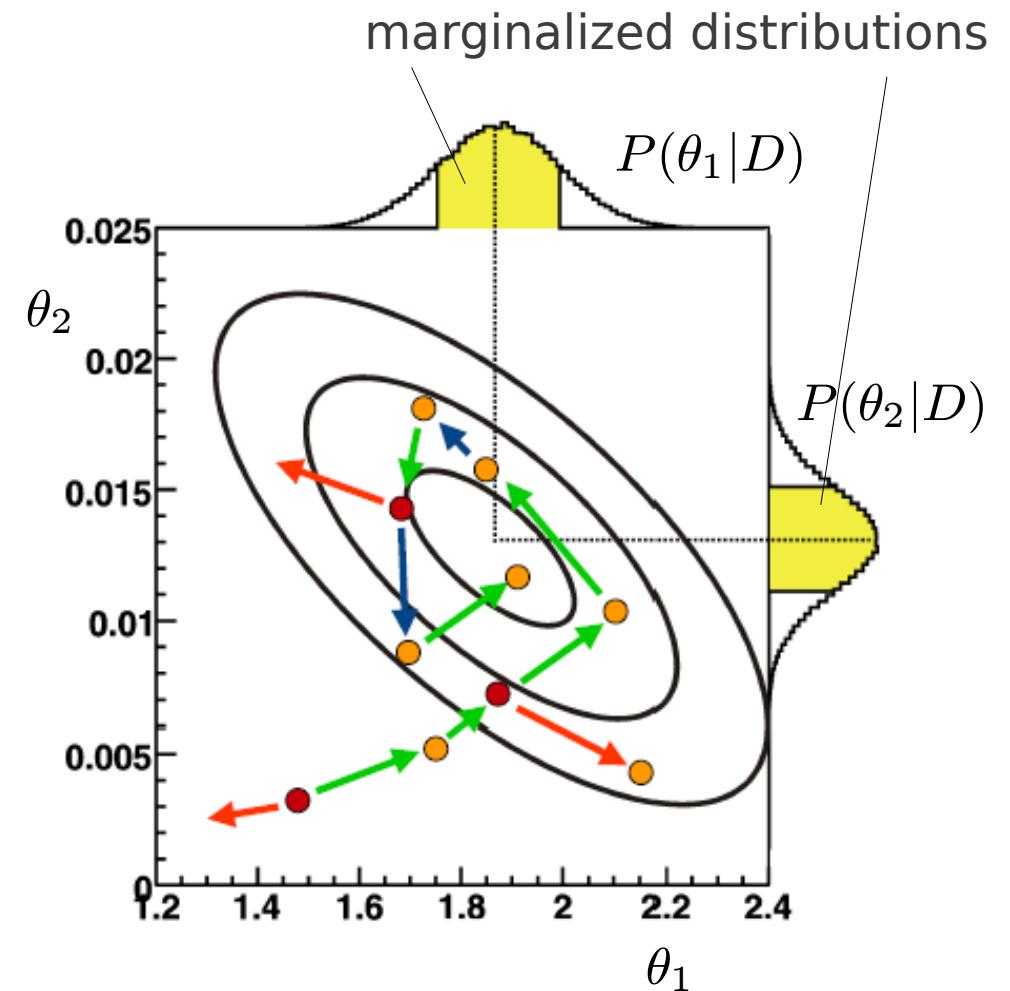
- Adaptive local random walk to sample from posterior

$$\vec{\theta} \sim P(D|\vec{\theta})P(\vec{\theta})$$

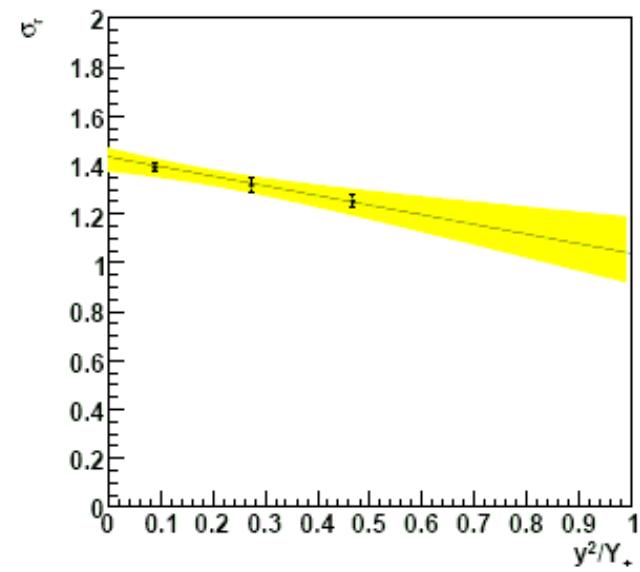
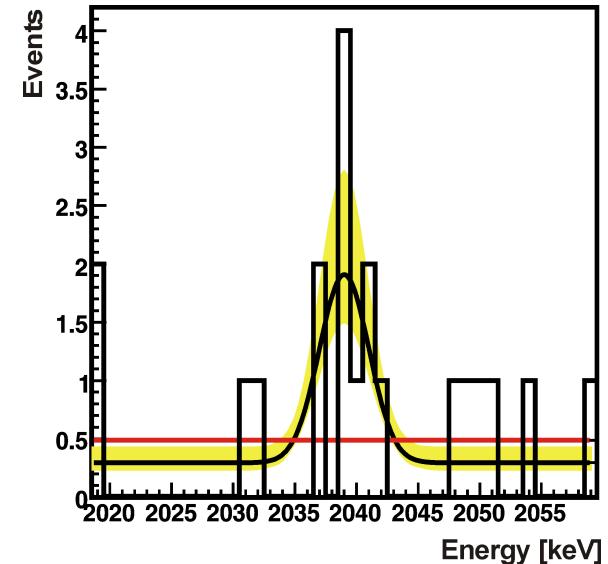
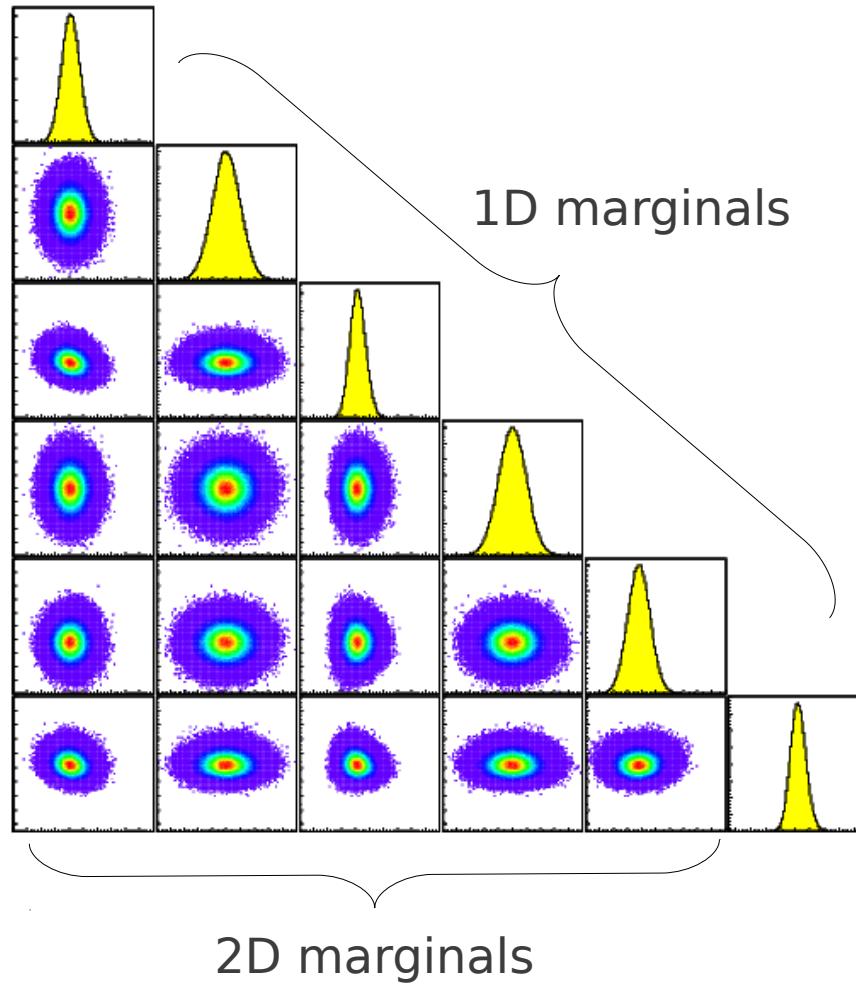
- Marginalization of posterior:

$$P(\theta_i|D) = \int d\vec{\theta}_{j \neq i} P(D|\vec{\theta})P(\vec{\theta})$$

- Fill histograms with just one/two coordinates while sampling



Graphical output





MCMC and convergence



- Run multiple chains
- Draw starting points uniformly in parameter box
- Convergence if chains mix

Gelman & Rubin R value:

- Calculate average variance of all chains

$$W = \frac{1}{m} \frac{1}{n-1} \sum_{j=1}^m \sum_{i=1}^n (\theta_i - \bar{\theta}_j)^2$$

- Estimate variance of target distribution

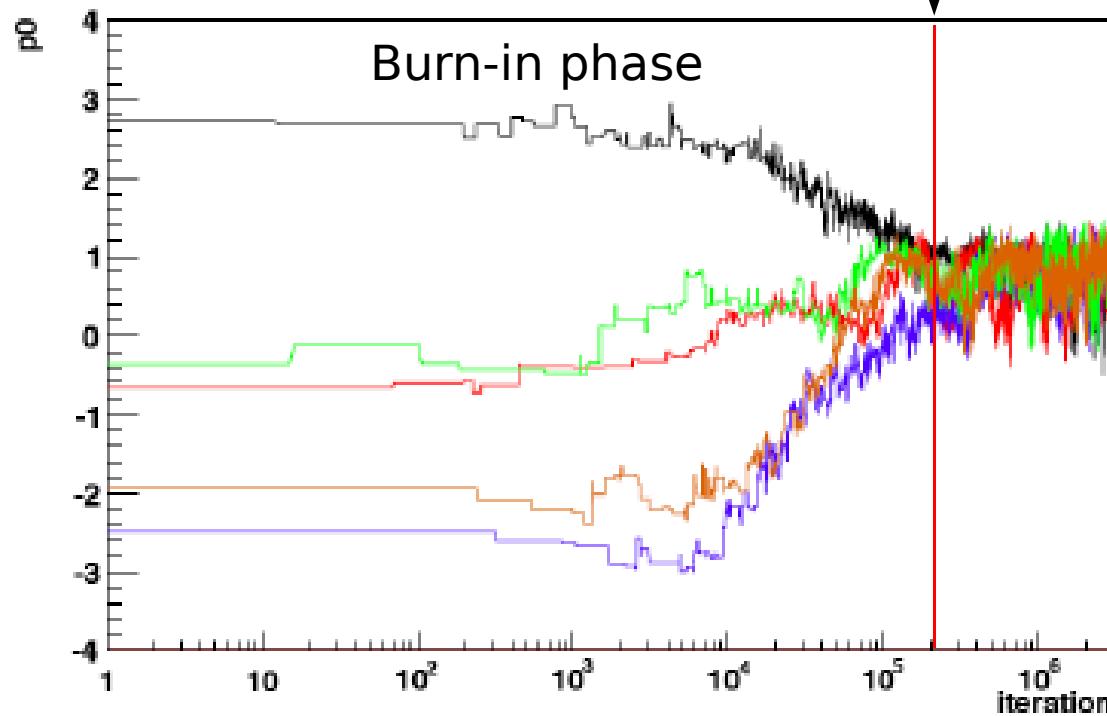
$$V = (1 - \frac{1}{n})W + \frac{1}{m-1} \sum_{j=1}^m (\bar{\theta}_j - \bar{\theta})^2$$

- Calculate ratio and compare with stopping criterion (relaxed version):

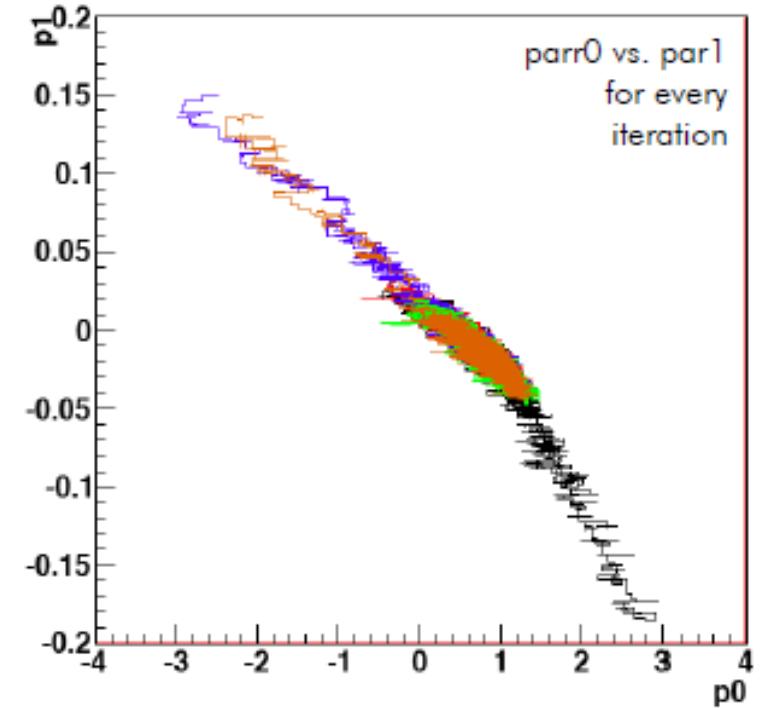
$$R = \sqrt{\frac{V}{W}} < 1.x, x = 0.1 \text{ default}$$

Gelman & Rubin, StatSci 7, 1992

Convergence à la Gelman & Rubin

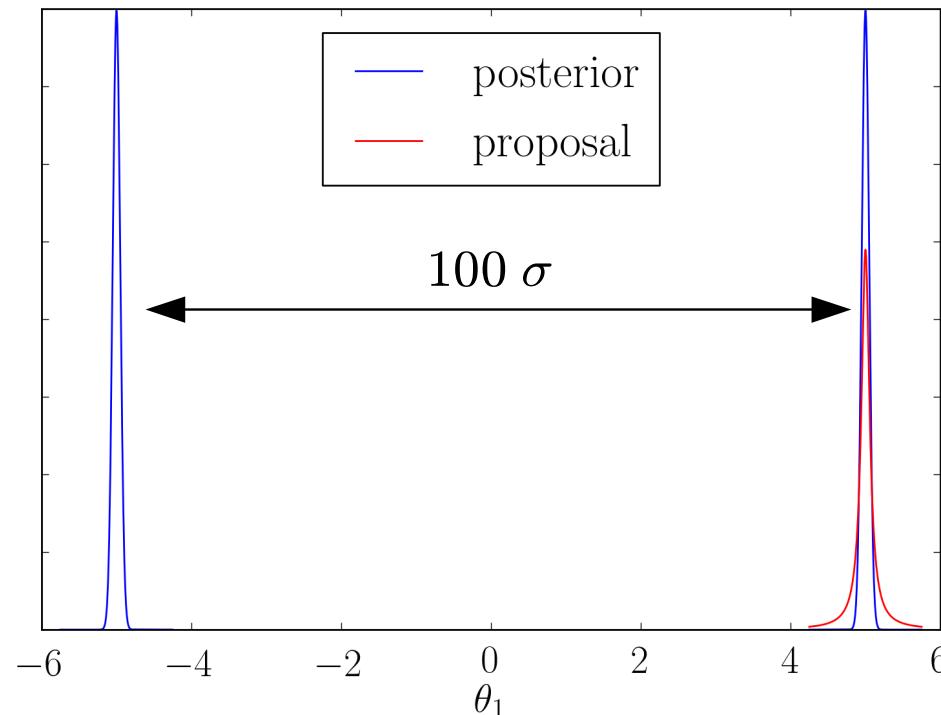


Parameter 0 vs. iteration



Parameter 1 vs. parameter 0

Multimodality



What if there are two far-away modes?



Importance Sampling



- Basic importance sampling

$$\int P = \int \frac{P}{q} q = \mathbb{E}_q\left[\frac{P}{q}\right] \approx \frac{1}{N} \sum_{i=1}^N \frac{P(\vec{\theta})}{q(\vec{\theta})} = \frac{1}{N} \sum_{i=1}^N w_i, \quad \vec{\theta} \sim q(\vec{\theta})$$

- Maximum efficiency if $P(\vec{\theta}) = q(\vec{\theta})$
- How to choose a good proposal $q(\vec{\theta})$?

Adaptive importance sampling:

- Proposal: multivariate mixture density
- Minimize Kullback-Leibler divergence $KL(P||q)$ with Expectation-Maximization

$$KL(P||q) = \int P \log \frac{P}{q}$$

- Massively parallelize
- Need very good initial proposal q^1 in 30D or nearly all components die

$$q(\vec{\theta}) = \sum_{j=1}^m \alpha_j q_j(\vec{\theta} | \vec{\mu}, \Sigma)$$

α : weight
 $\vec{\mu}$: mean
 Σ : covariance
 q_j : Gauss, Student t

For step $t \geq 1$:
 Draw N samples $\vec{\theta}_i$ from q^t
 Compute w_i
 if $KL(P||q^t) \approx KL(P||q^{t-1})$:
 stop
 else:
 Update $q^t \rightarrow q^{t+1}$ based on
 $\{(\vec{\theta}_1, w_1), \dots, (\vec{\theta}_N, w_N)\}$

Cappé et al. (2008), Kilbinger et al.(2009), pmclib



The best of both worlds



MCMC: local

- Same local proposal adjusts well to many problems
- Single chain maps out local features

PMC: global

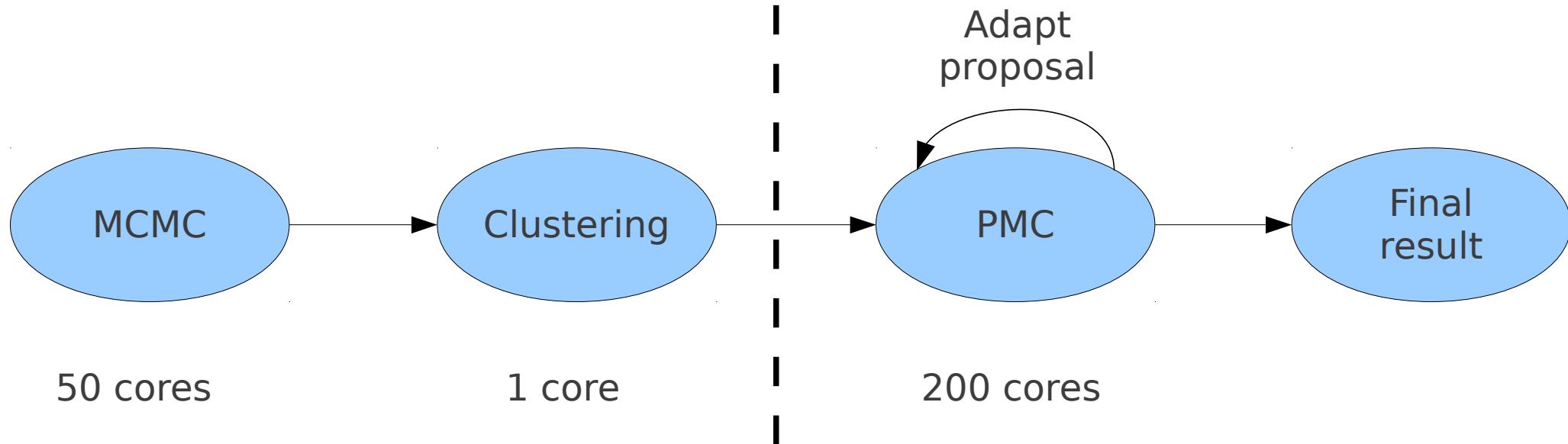
- Massively parallel
- Handles multimodality, degeneracy

MCMC: local

- Same local proposal adjusts well to many problems
- Single chain maps out local features

PMC: global

- Massively parallel
- Handles multimodality, degeneracy

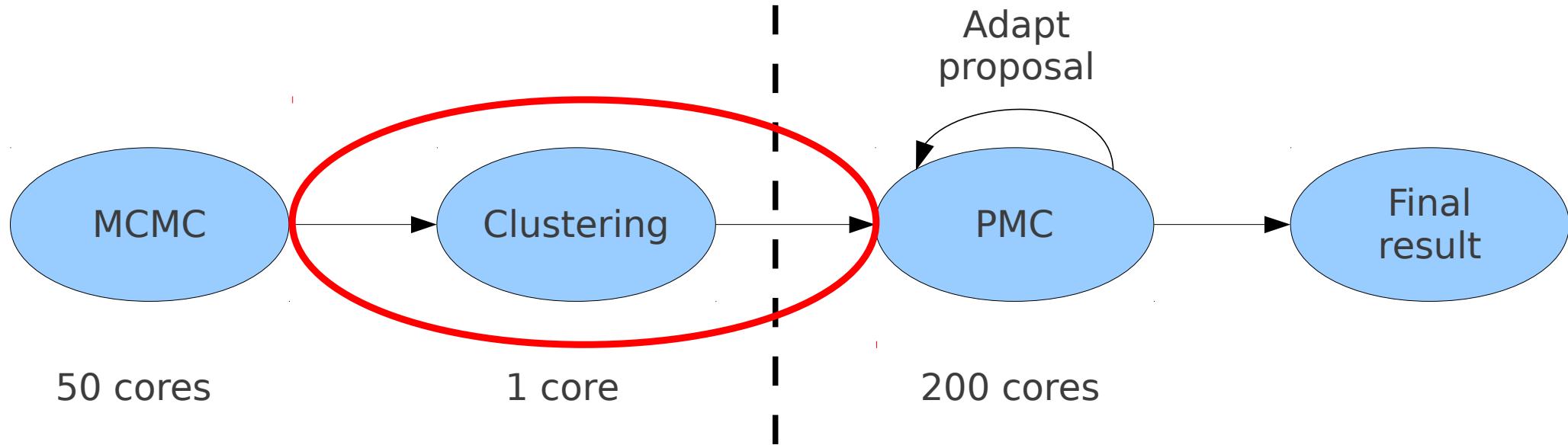


MCMC: local

- Same local proposal adjusts well to many problems
- Single chain maps out local features

PMC: global

- Massively parallel
- Handles multimodality, degeneracy



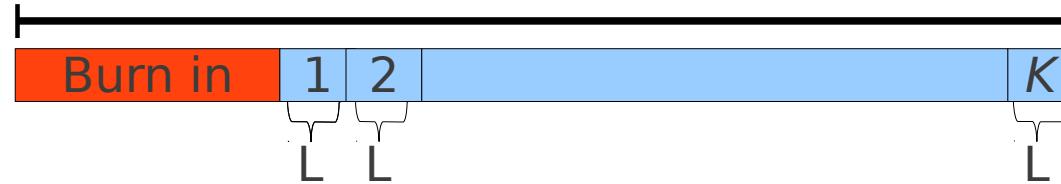


Initial proposal

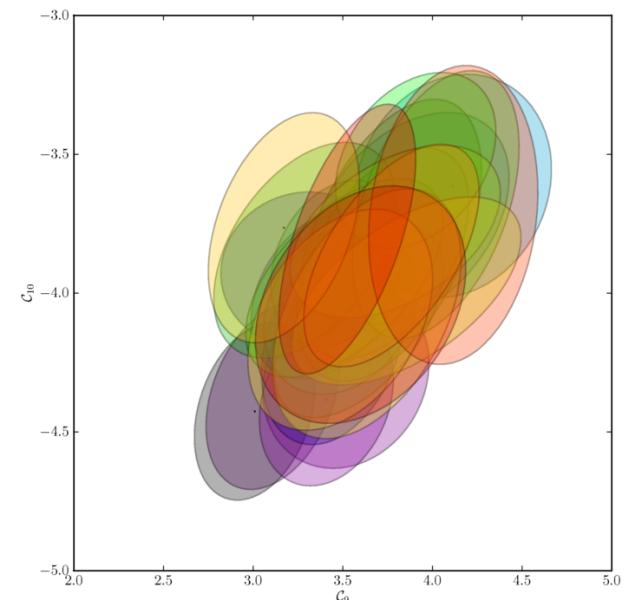


- 1) Split up each chain into patches, one multivariate component per patch
- 2) Use hierarchical clustering to reduce #components
 - a) Clustering requires its own good initial guess!
- 3) One component per output cluster, equal weights = initial proposal for PMC

Chain with N samples



- Patch length L : 500 - 1000 steps
small to cover local features
- Multivariate density from patch mean
and covariance
- Ex: $N=60000$, $L=1000$,
burn-in=6000, $K=54$





Hierarchical clustering



- Have Gaussian mixture with M components $f(\vec{\theta}) = \sum_{l=1}^M \beta_l f_l(\vec{\theta} | \vec{\mu}_l, \Sigma_l)$
- Want only $m < M$ components $q(\vec{\theta}) = \sum_{j=1}^m \alpha_j q_j(\vec{\theta} | \vec{\mu}_j, \Sigma_j)$
- Select q minimizing the distance measure

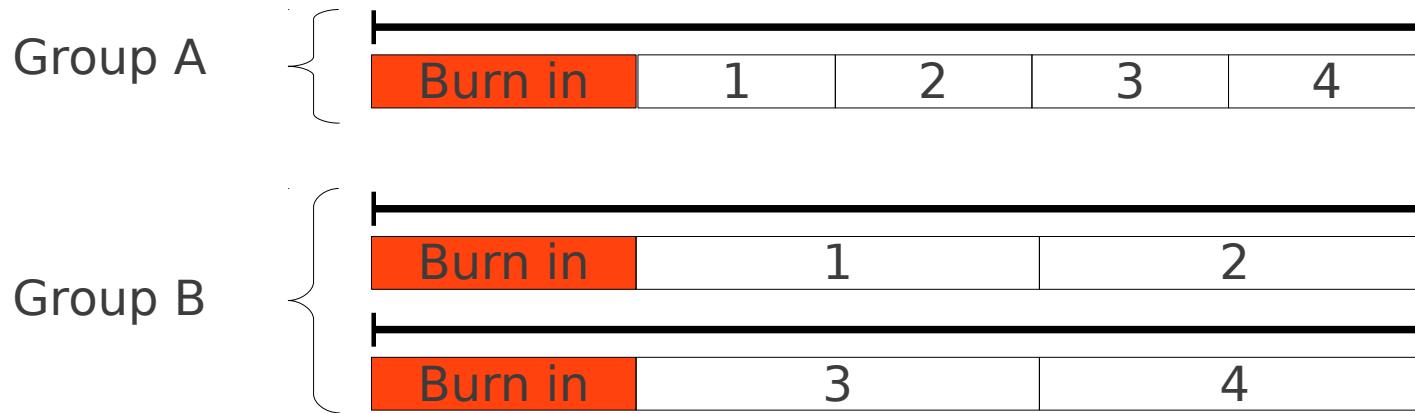
$$d(f, q) = \sum_{l=1}^M \beta_l \min_{j=1}^m KL(f_l || q_j)$$

- Fast: work at level of components, not samples
- Expectation maximization: local(!) minimum of $d(f, q)$
- Again: **need good guess** for initial clusters $q(\vec{\theta})$

Goldberger & Roweis, Hierarchical clustering of a mixture model. (2004)



Hierarchical clustering initial guess



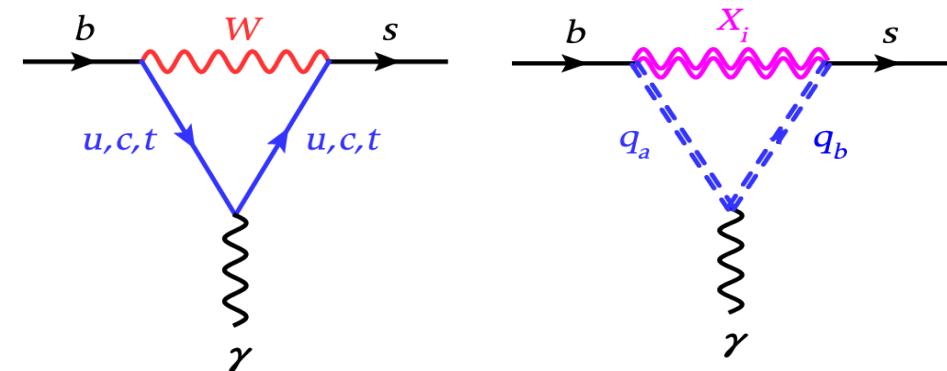
- Need $\gtrsim \dim \vec{\theta}$ components for each posterior mode
- #chains in a mode is random but doesn't reflect mode's weight
- Group chains with R value:
Add chain to group if $\forall \theta_i : R \lesssim 1.5$
- Form components from each group:
longer patches from each chain $\Rightarrow q_{init}$

Flavor changing neutral currents

parton \Leftrightarrow meson

$$b \rightarrow s\gamma \Leftrightarrow B \rightarrow K^*\gamma \quad Br \sim 10^{-5}$$

$$b \rightarrow s\bar{\mu}\mu \Leftrightarrow B \rightarrow K^{(*)}\bar{\mu}\mu \quad Br \sim 10^{-7}$$



- Effective theory: separate **high energy** from low energy scales

$$\mathcal{H}^{\text{eff}} = \frac{4G_F}{\sqrt{2}} V_{tb} V_{ts}^* \sum_{i=7,9,10} \mathcal{C}_i \mathcal{O}_i$$

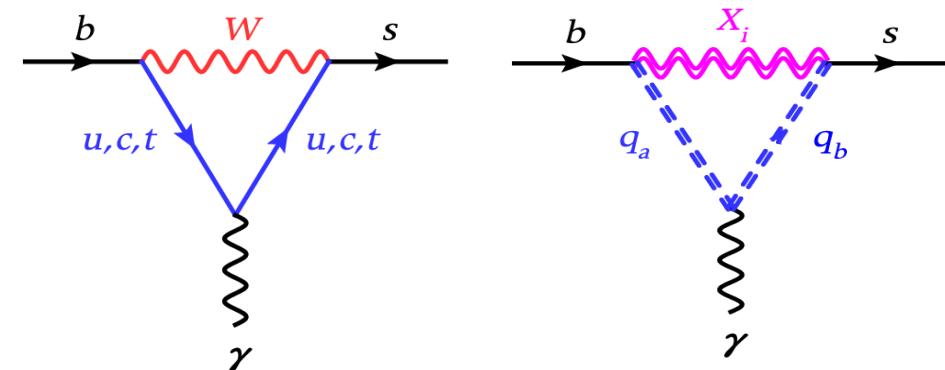
- Heavy particles (new physics) contribute to Wilson coefficient \mathcal{C}_i

Flavor changing neutral currents

parton \Leftrightarrow meson

$$b \rightarrow s\gamma \Leftrightarrow B \rightarrow K^*\gamma \quad Br \sim 10^{-5}$$

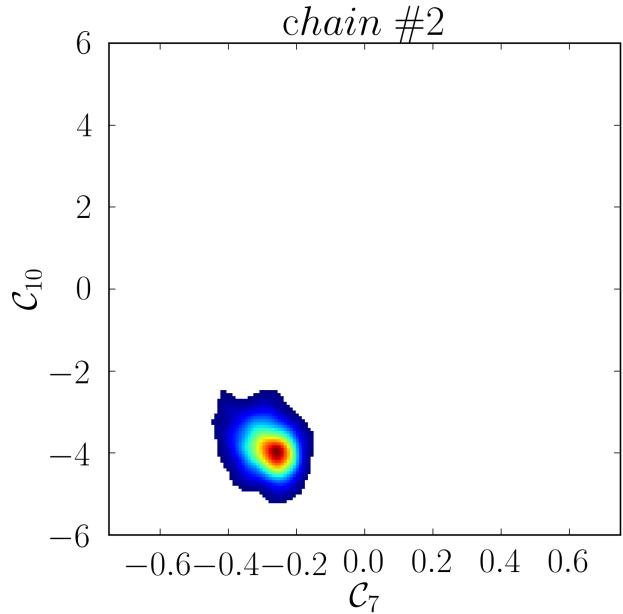
$$b \rightarrow s\bar{\mu}\mu \Leftrightarrow B \rightarrow K^{(*)}\bar{\mu}\mu \quad Br \sim 10^{-7}$$



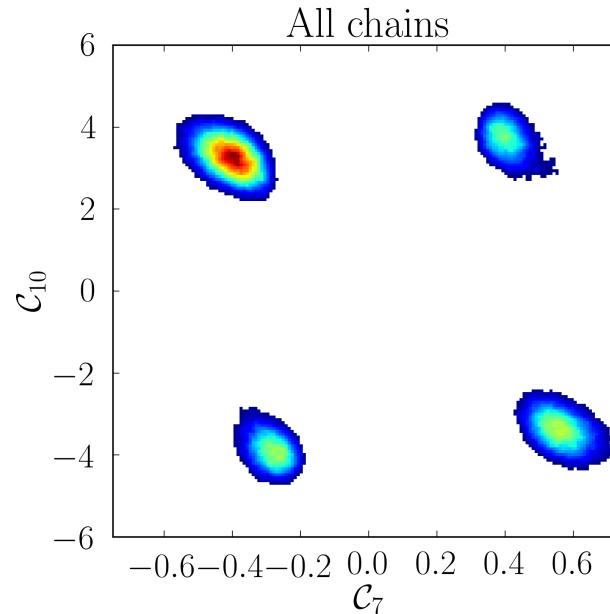
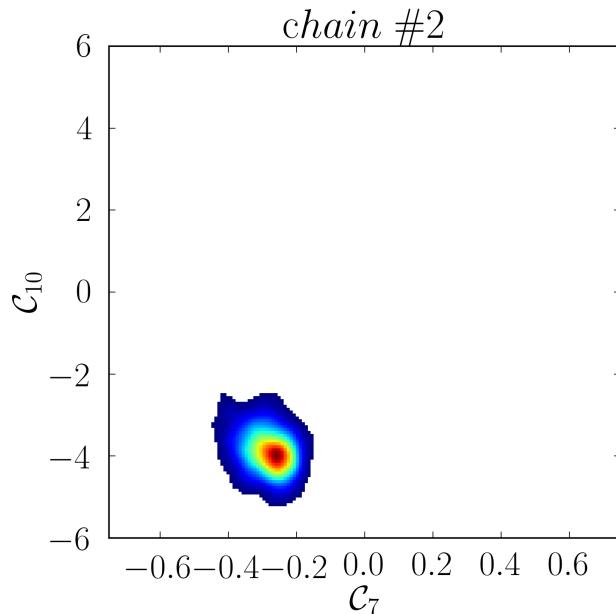
- Effective theory: separate **high energy** from low energy scales

$$\mathcal{H}^{\text{eff}} = \frac{4G_F}{\sqrt{2}} V_{tb} V_{ts}^* \sum_{i=7,9,10} \mathcal{C}_i \mathcal{O}_i$$

- Heavy particles (new physics) contribute to Wilson coefficient \mathcal{C}_i
- Extract \mathcal{C}_i in a global fit with EOS <http://project.het.physik.tu-dortmund.de/eos/>
- Tough problem: 28 nuisance parameters, 3 Wilson coefficients
22 observables, 59 measurements
0.2 s for one likelihood evaluation, need $\gtrsim 5 \cdot 10^6$ samples



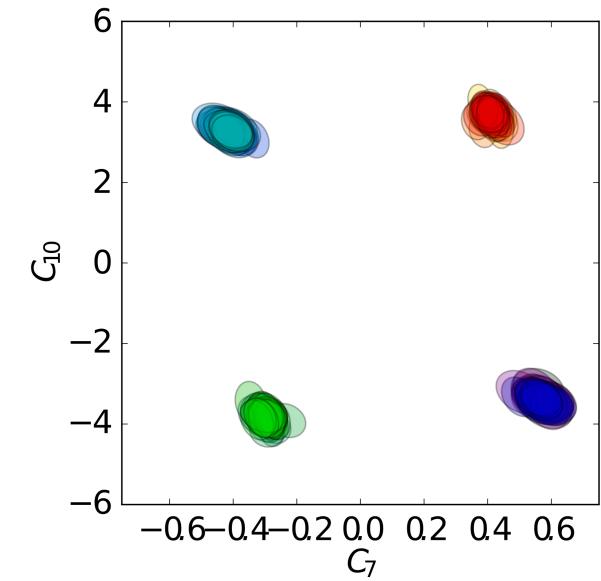
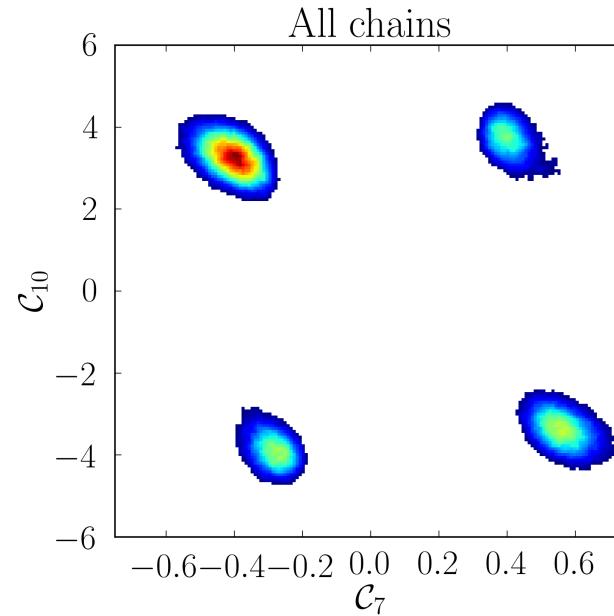
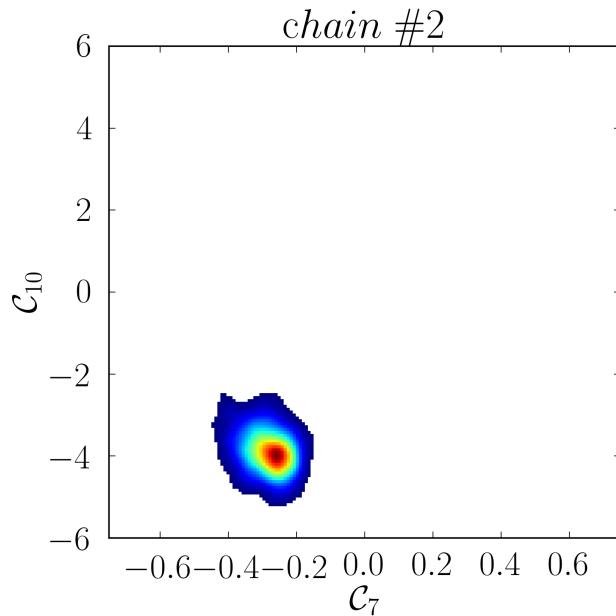
Single chain – single mode



Single chain - single mode

50 chains - four groups

Multimodal: chains do not mix

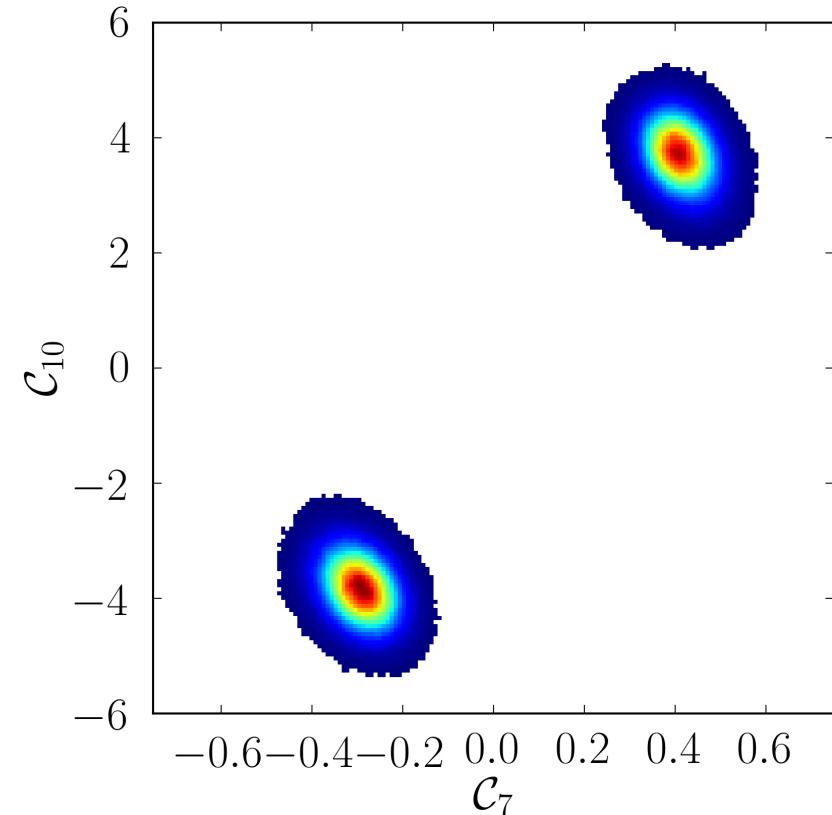
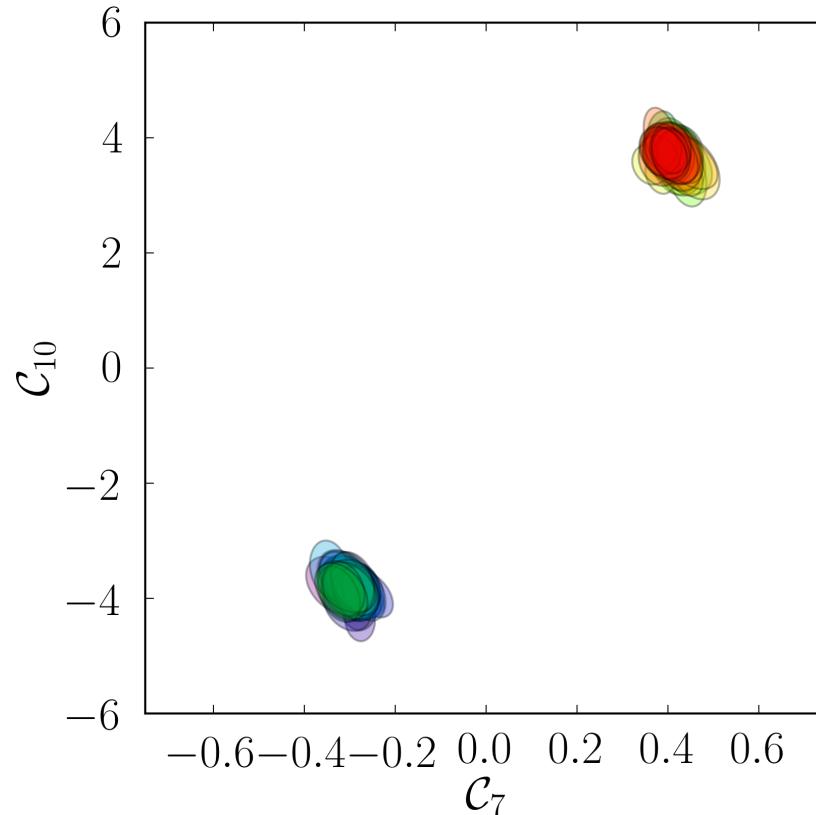


Single chain - single mode

50 chains - four groups

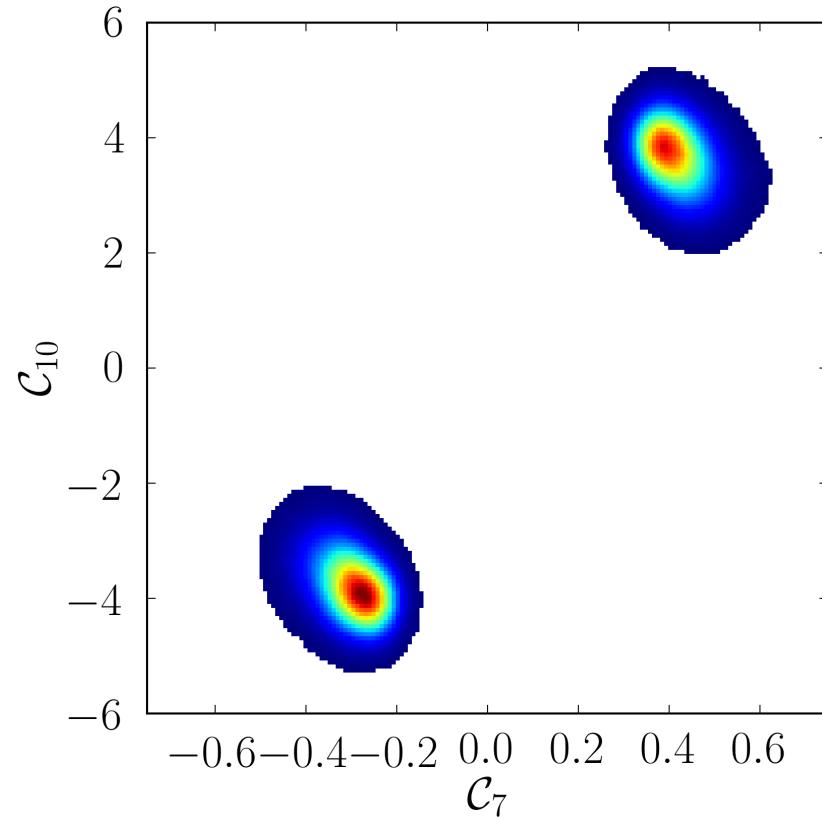
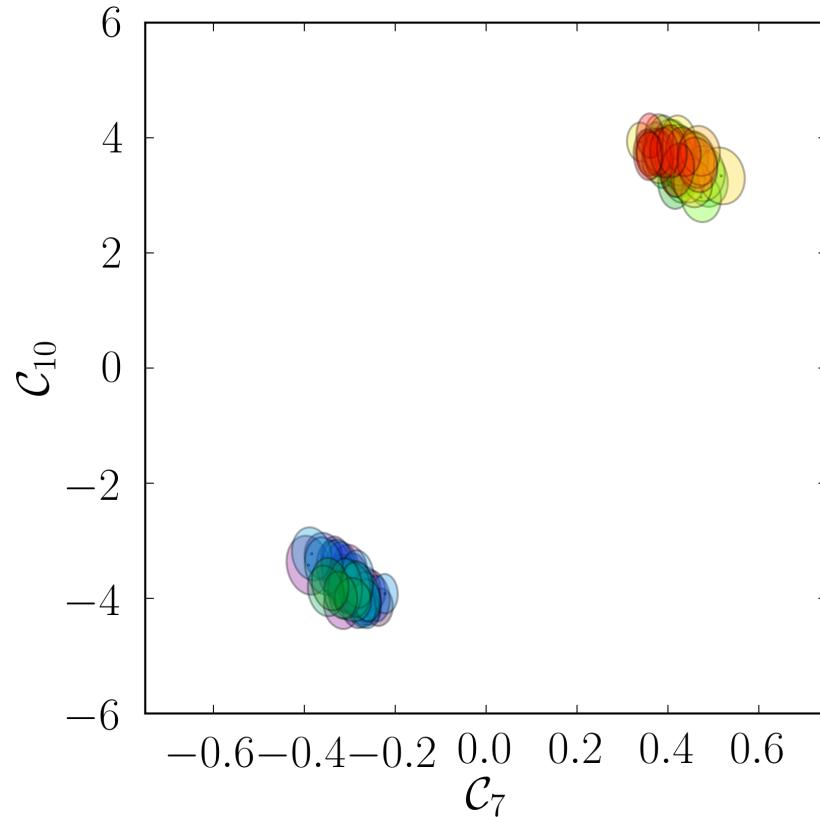
35 components per group

Multimodal: chains do not mix



- Only two modes survive
- others suppressed by factor of 10^{10}
- 500 000 samples per step

- Note:
- Smoothed density
 - Outliers removed



- Components adjusted
- Proposal function converged

- Final step: 2 000 000 samples
- PMC ran for only ~ 10 h

Towards a massively parallel, nearly black-box Monte Carlo algorithm coping with multiple modes:

MCMC + clustering + PMC