Refining a Markov Chain Monte Carlo Algorithm for Fitting Neutron Reflectometry Data

Aaron Schankler

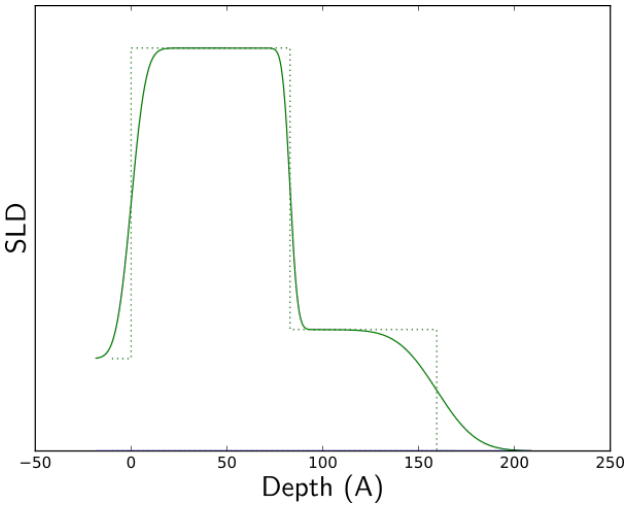
Haverford College

August 2016

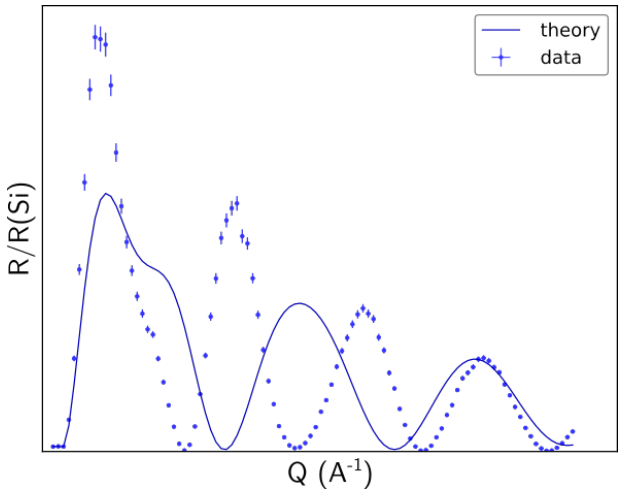
Mentor: Paul Kienzle

Neutron reflectometry is an inverse fitting problem

- Reflectivity is sensitive to the thickness and composition of layers in an interface
- Well understood but not reversible
  - Can calculate the expected reflectivity from a depth profile
  - Cannot calculate the depth profile from a reflectivity measurement
- We build a parametrized model of the surface structure
- Vary the parameters until expected reflectivity matches the data



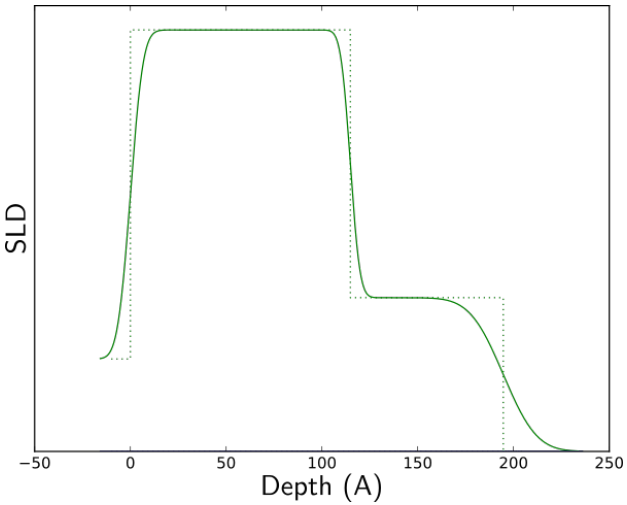
(a) Depth profile



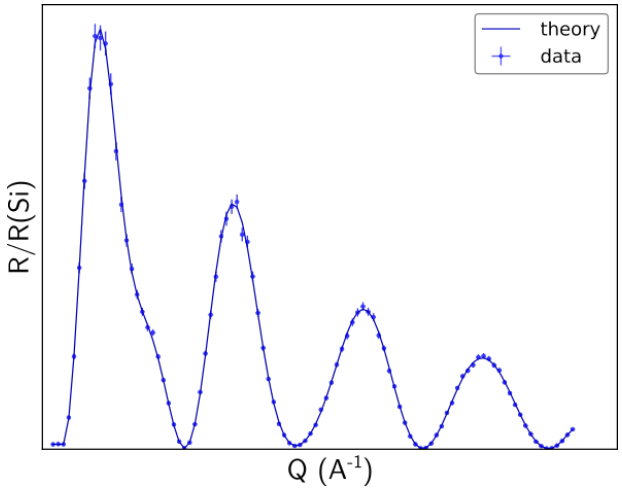
(b) Reflectivity

Neutron reflectometry is an inverse fitting problem

- Reflectivity is sensitive to the thickness and composition of layers in an interface
- Well understood but not reversible
  - Can calculate the expected reflectivity from a depth profile
  - Cannot calculate the depth profile from a reflectivity measurement
- We build a parametrized model of the surface structure
- Vary the parameters until expected reflectivity matches the data

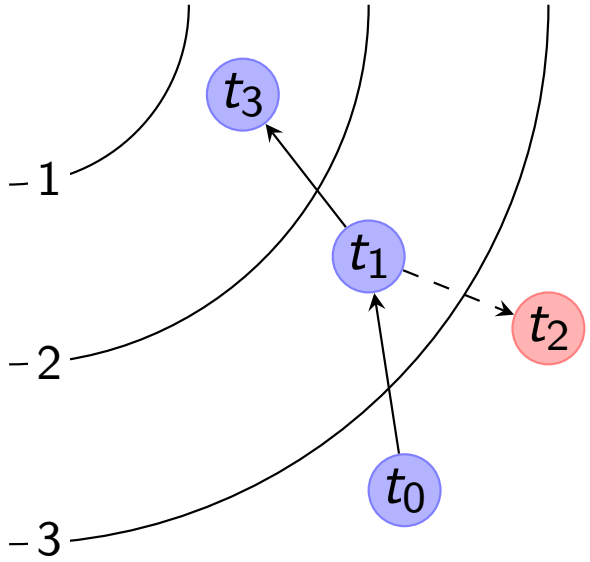


(a) Depth profile



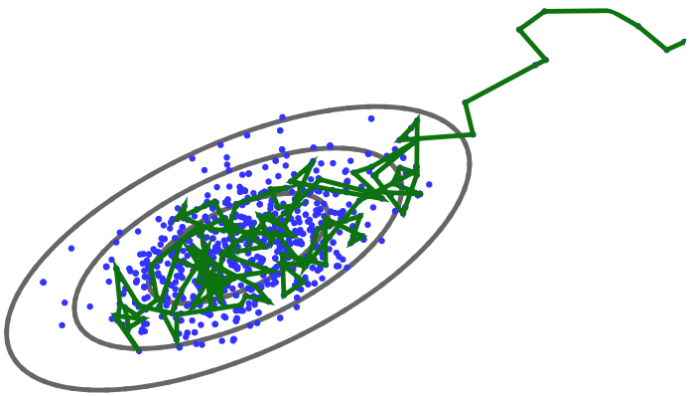
(b) Reflectivity

Markov Chain Monte Carlos can address inverse problems



- Markov chain Monte Carlos (MCMC) are random algorithms
- The chains accumulate a history of many discrete steps
- Can move "uphill" and escape local optima
  - MCMCs are robust optimizers

MCMC gives more information than just the optimum

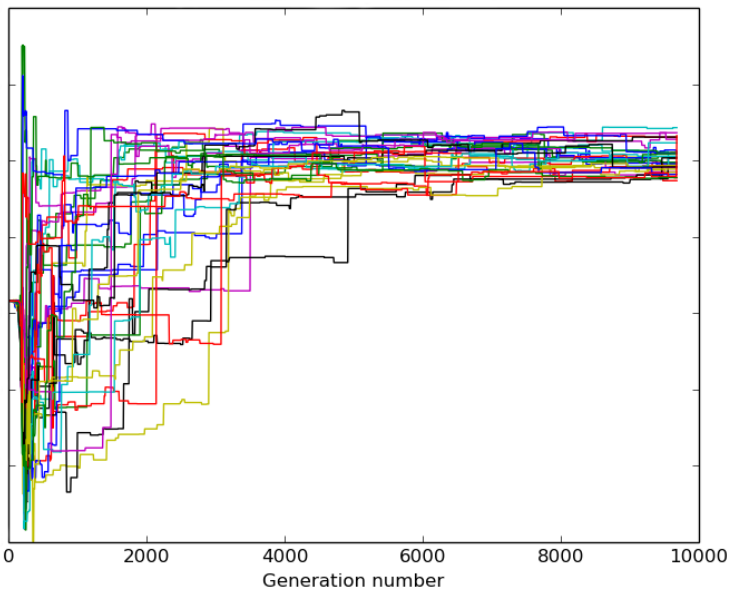


Iain Murray, Neural Information Processing Systems

Confrence, 2015

- After many steps, the chain's history reflects the probability of a set of parameters
  - Can be used to determine parameter correlation and credible intervals
- A longer run gives a more precise result
- Results are provably correct in the limit

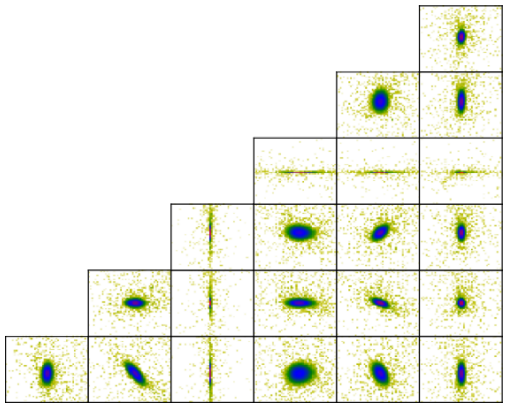
MCMC often requires a “burn in” period



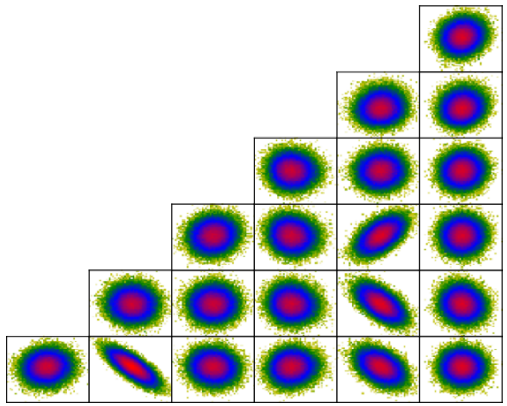
- The starting values are often not typical of values later in the chain
- They can distort statistics at the end of a run
- Discarding some initial samples can eliminate this effect

An automated test for burn in simplifies interpretation

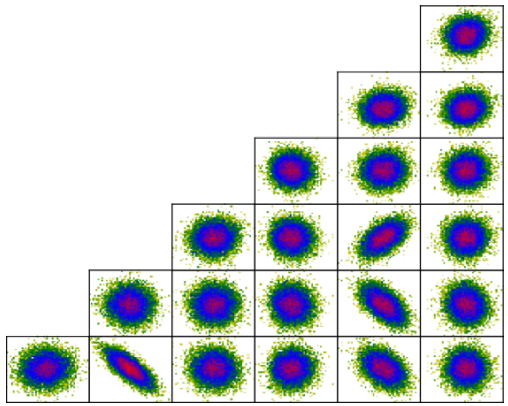
- A short burn in distorts results, while an overly long burn in is wasteful
- Determining the correct amount of burn in requires user guesswork
- Automated burn selection gives good results while simplifying the fitting process



(a) Including burn in



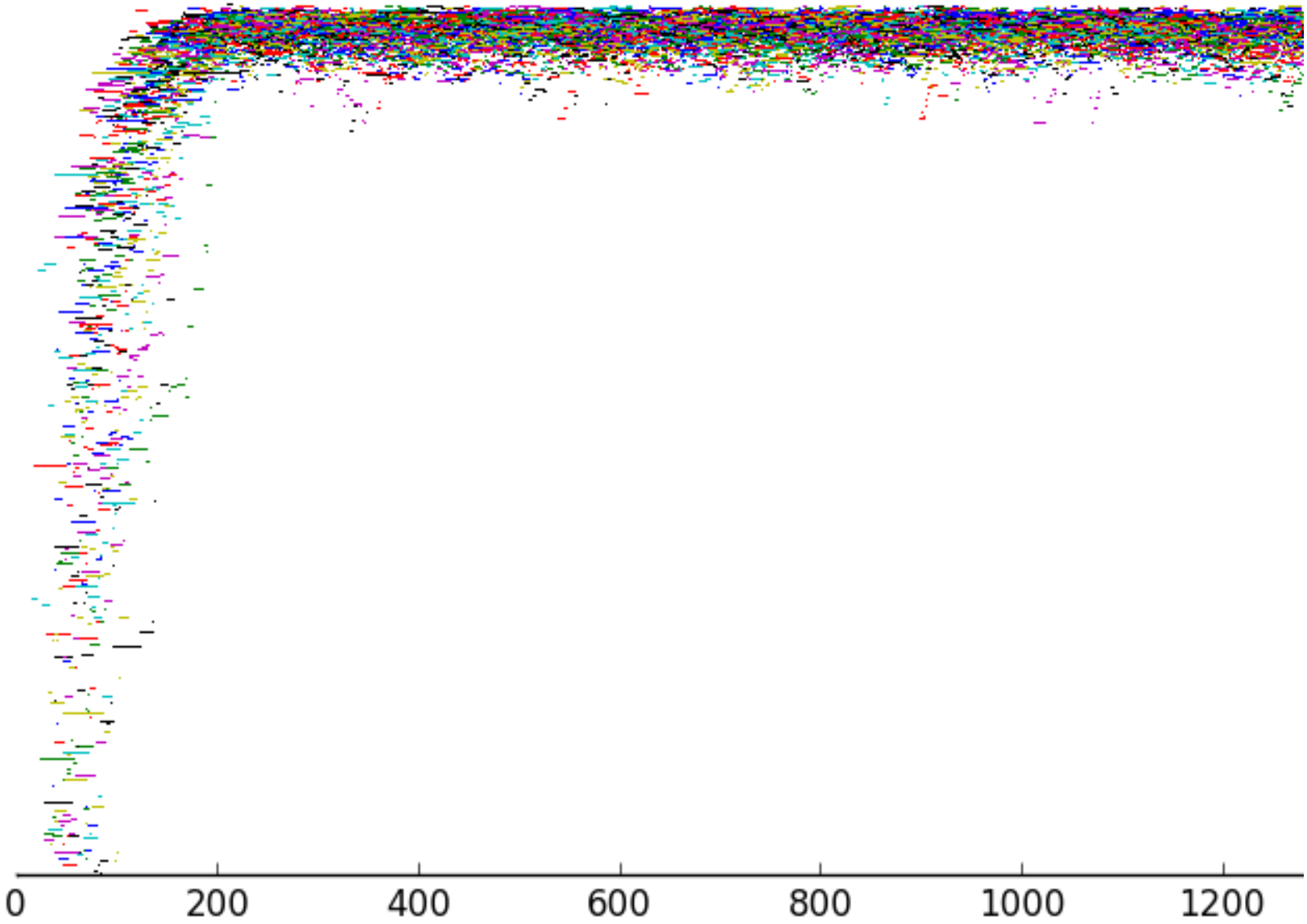
(b) First 3% removed

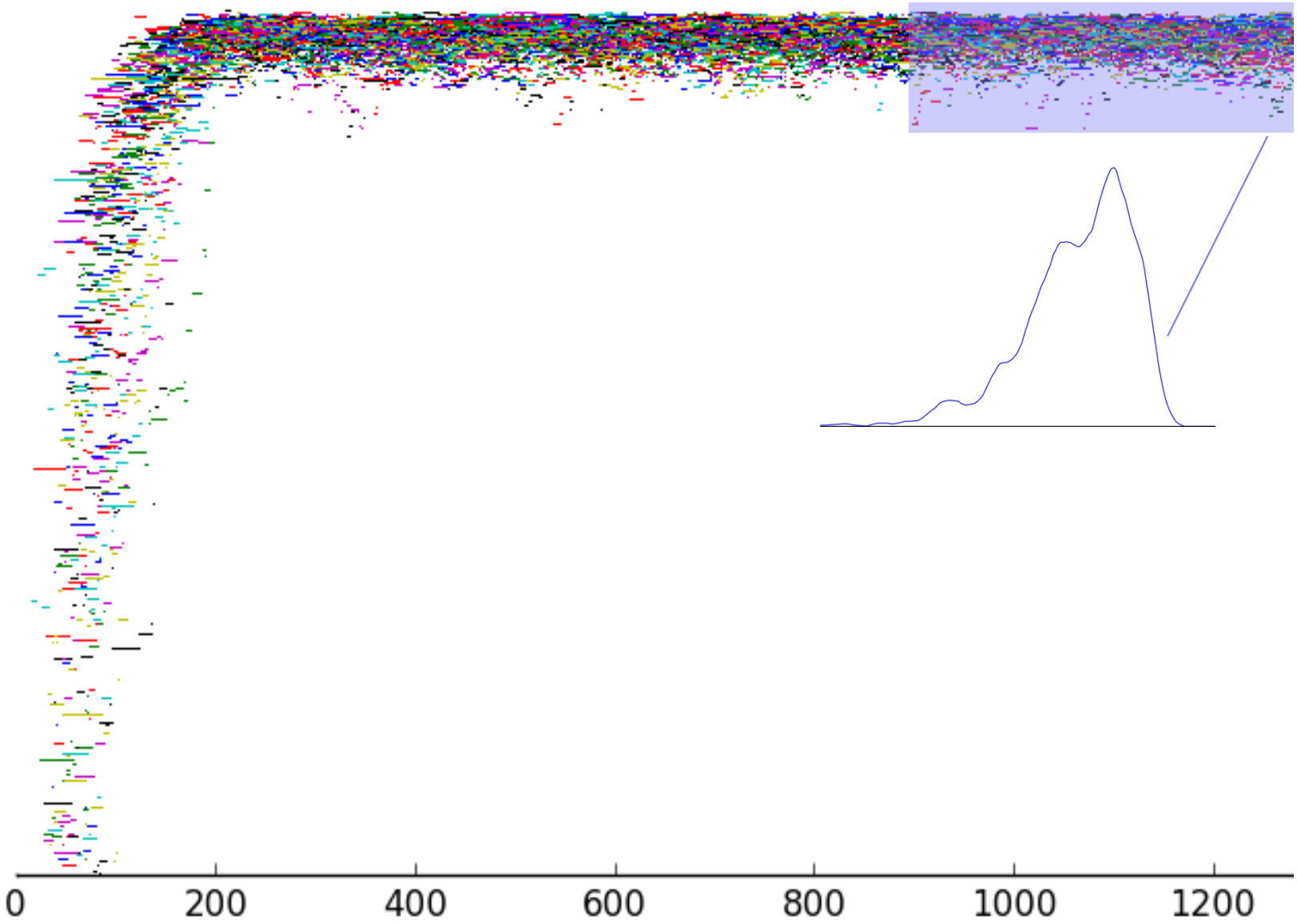


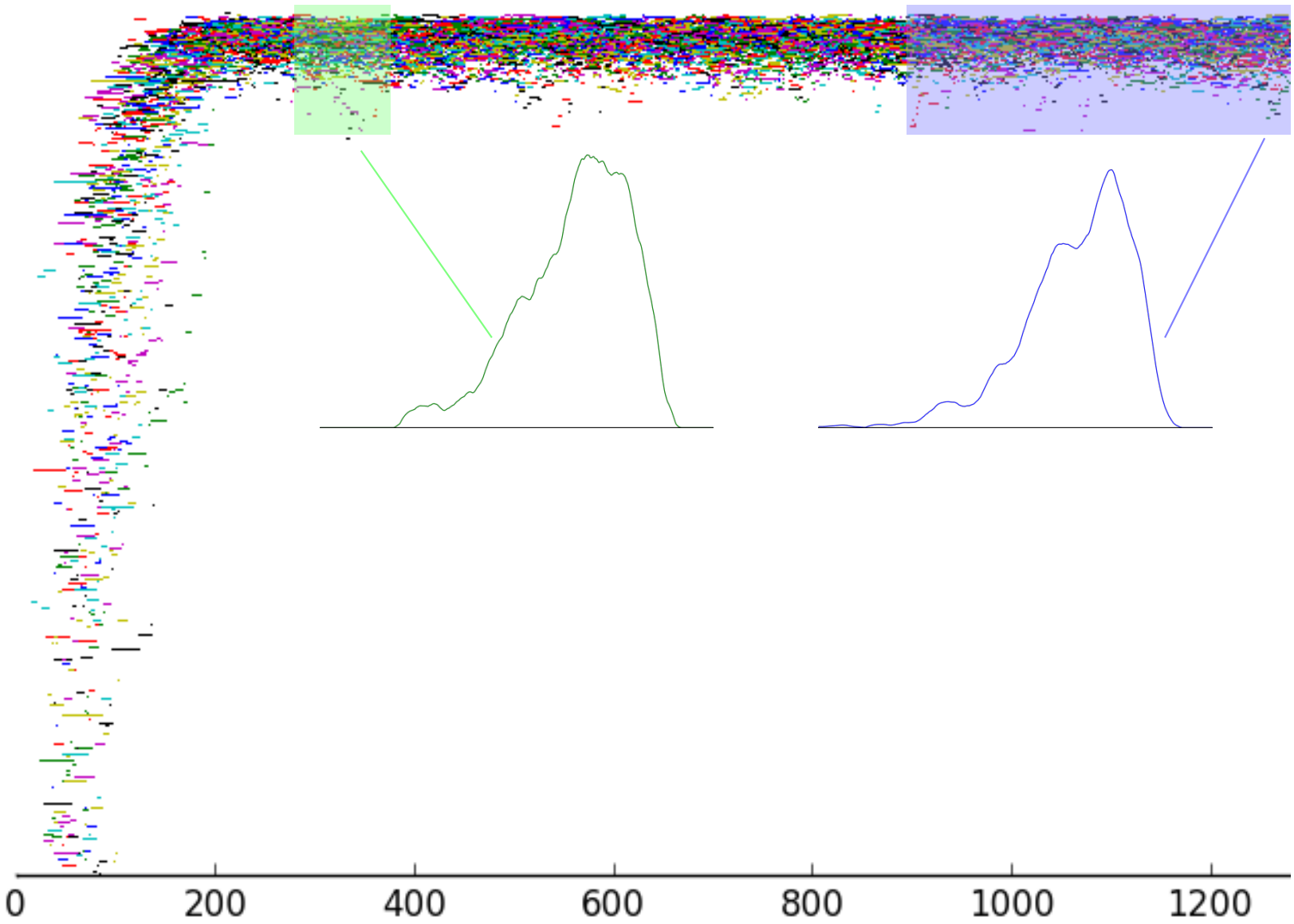
(c) First 90% removed

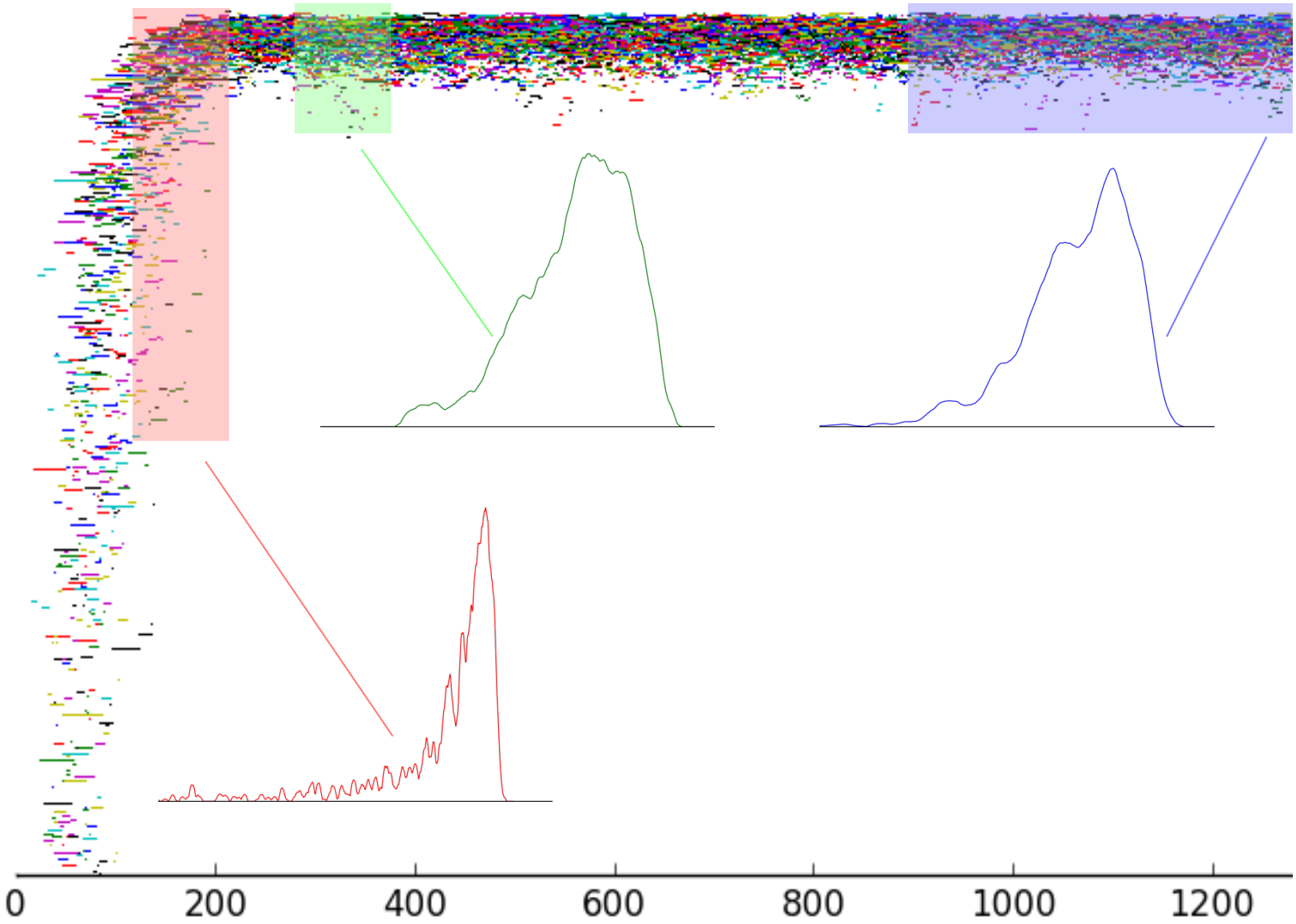
To determine the end of burn in we test for stationarity

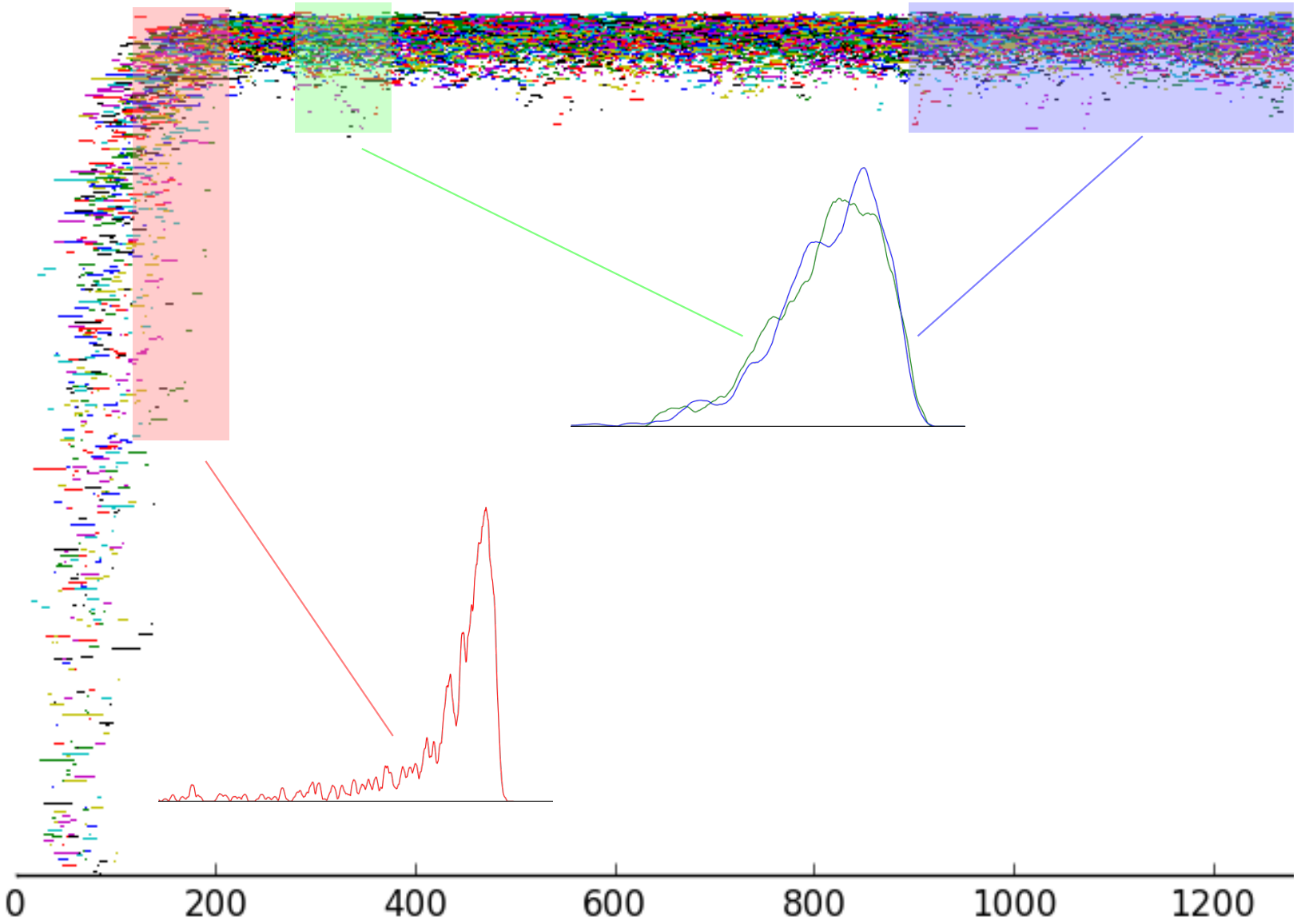
- In a converged chain, we expect the beginning to "look like" the end
  - Formally, we expect the distribution of the points to remain the same
  - Difficult to test in high dimensional search spaces
- Instead we calculate the *log probability* of each point
  - This collapses points in many dimensional spaces into a single dimension
  - Testing for stationarity in a single dimension becomes tractable

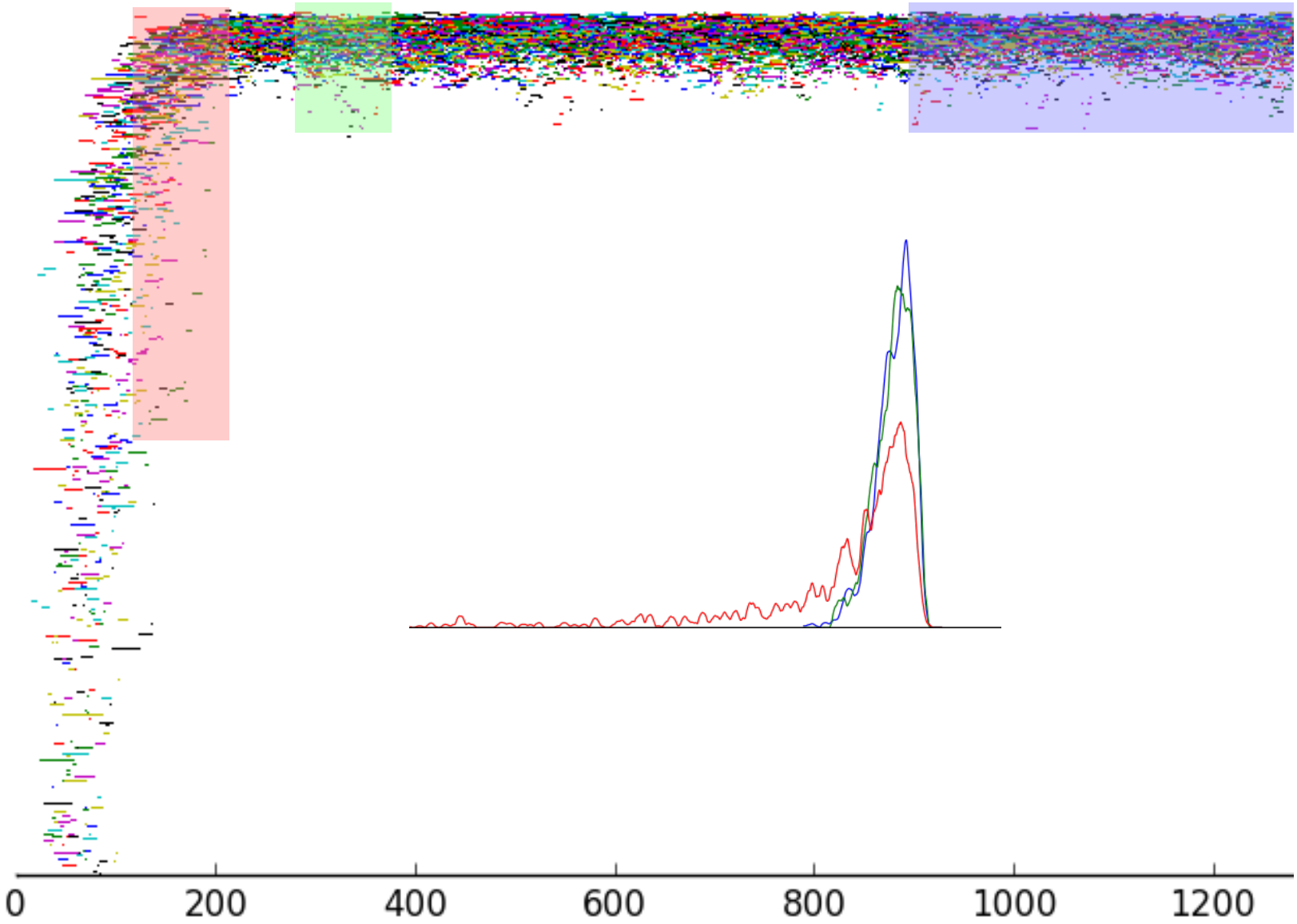

0
200
400
600
800
1000
1200


0
200
400
600
800
1000
1200

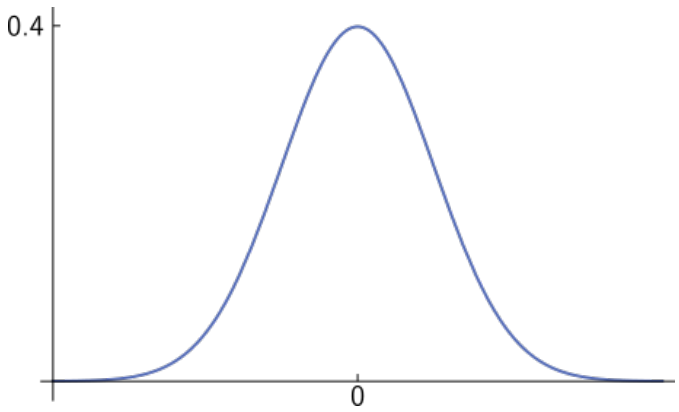

0
200
400
600
800
1000
1200


0
200
400
600
800
1000
1200

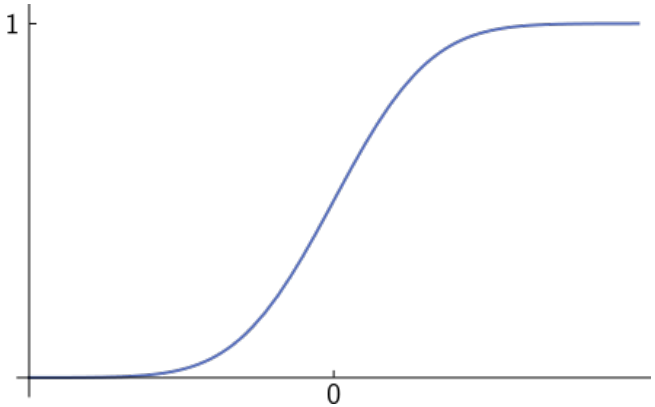

0
200
400
600
800
1000
1200


0
200
400
600
800
1000
1200

Formalizing distribution fit tests



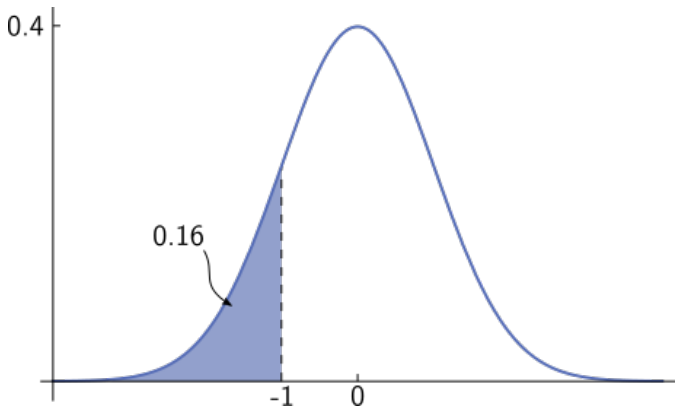
(a) Normal PDF



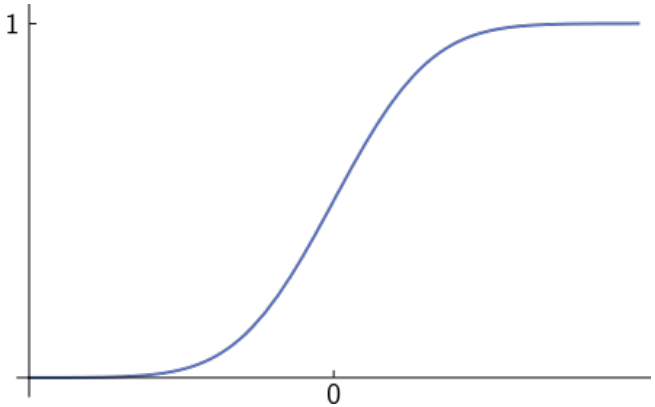
(b) Normal CDF

- We draw random subsets of points to compare
  - Reduces autocorrelation
  - Reduces the power of the test
- Test for differences in distribution by comparing observed CDFs

Formalizing distribution fit tests



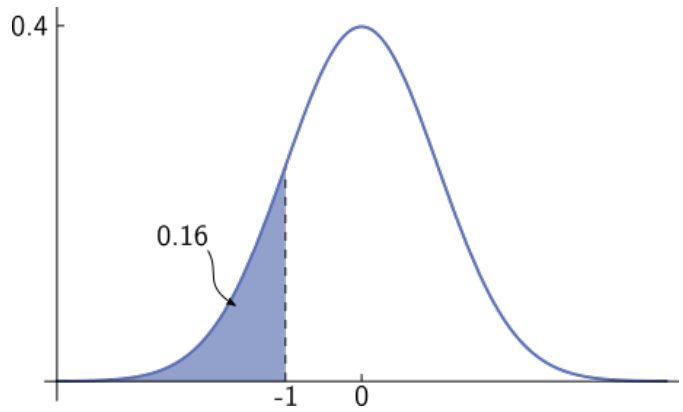
(a) Normal PDF



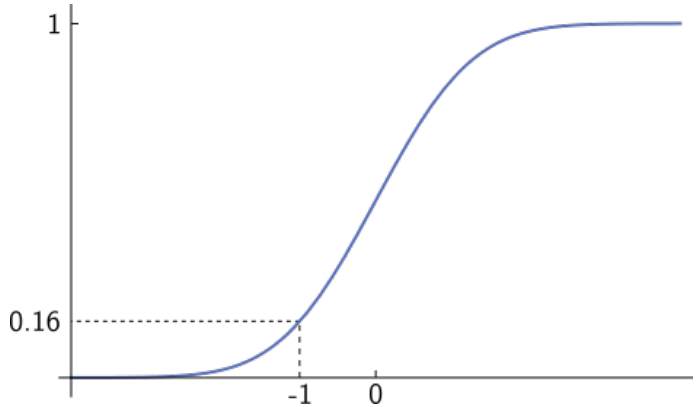
(b) Normal CDF

- We draw random subsets of points to compare
  - Reduces autocorrelation
  - Reduces the power of the test
- Test for differences in distribution by comparing observed CDFs

Formalizing distribution fit tests



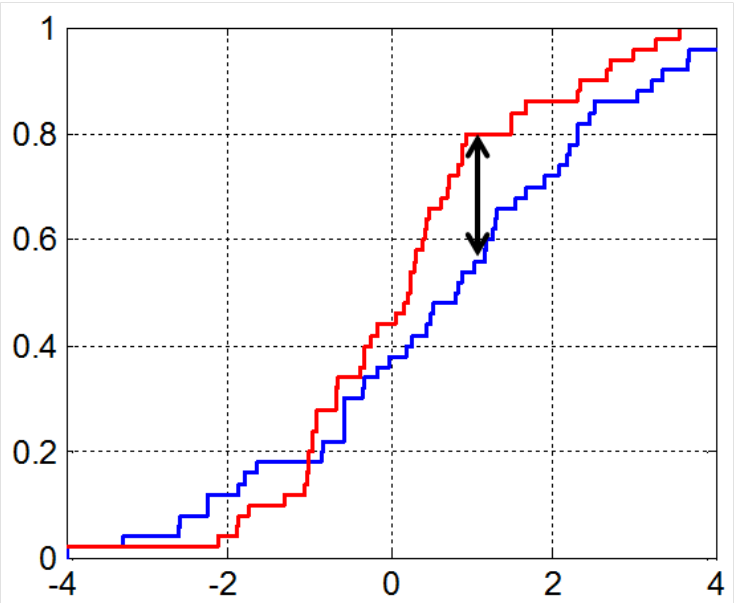
(a) Normal PDF



(b) Normal CDF

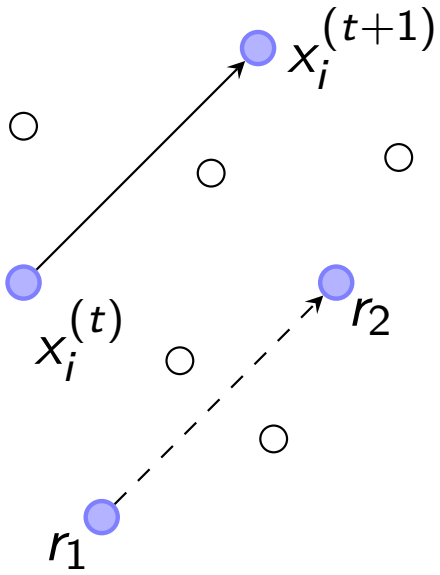
- We draw random subsets of points to compare
  - Reduces autocorrelation
  - Reduces the power of the test
- Test for differences in distribution by comparing observed CDFs

Formalizing distribution fit tests



- We draw random subsets of points to compare
  - Reduces autocorrelation
  - Reduces the power of the test
- Test for differences in distribution by comparing observed CDFs

The DREAM sampler is an effective MCMC

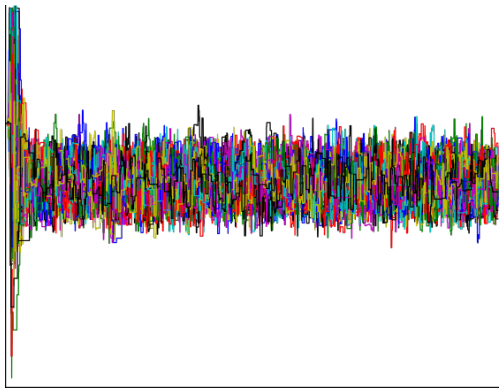


$$x_i^{(t+1)} = x_i^{(t)} + \gamma(r_2 - r_1)$$

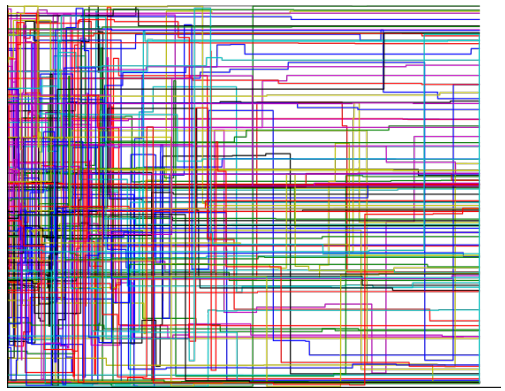
- A population of many chains is run simultaneously
- Uses a *differential evolution* algorithm to propose steps
- Running multiple chains
  - Better explores the parameter space
  - Preserves multimodal systems

DREAM does not perform well on all problems

- In high dimensional problems the chains are too far apart to "mix" effectively
  - Part of the curse of dimensionality
  - In these problems, chains reject almost all proposed points



(a) Good mixing, convergent fit



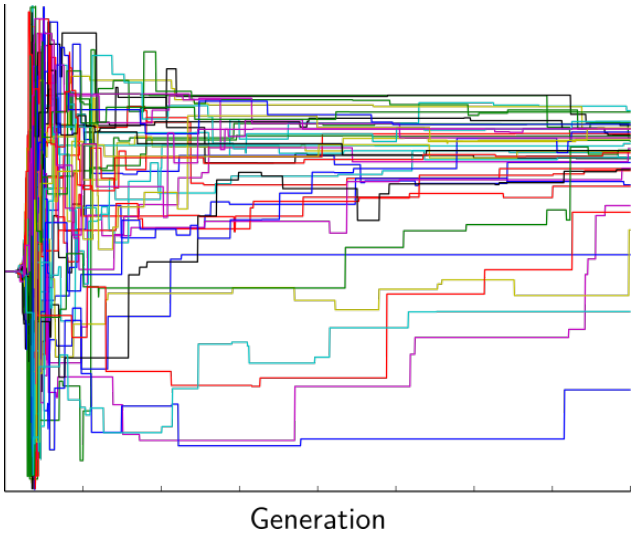
(b) Bad mixing, stuck fit

A test for stuckness

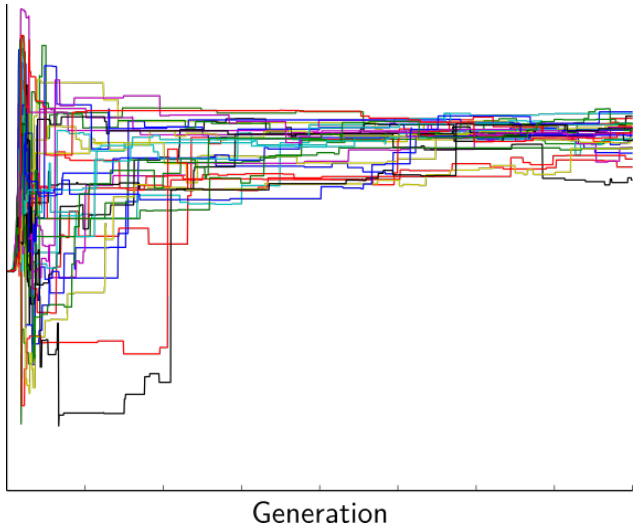
- The test considers the variance within a chain, as compared to overall variance
  - Low variability within a chain indicates a static chain
  - Many static chains indicates a stuck fit
- If stuckness is detected, the algorithm can be changed to address it
  - For example, by using another optimization algorithm to bring chains closer together

A more robust variant of DREAM

- An alternative to adaptively addressing stuckness is to use a more robust algorithm
- The literature describes the $DREAM_{(ZS)}$ algorithm, a variant of *DREAM* which bases its steps off of the history of every chain
- Reported to be more effective on multimodal fits and less prone to outlier chains

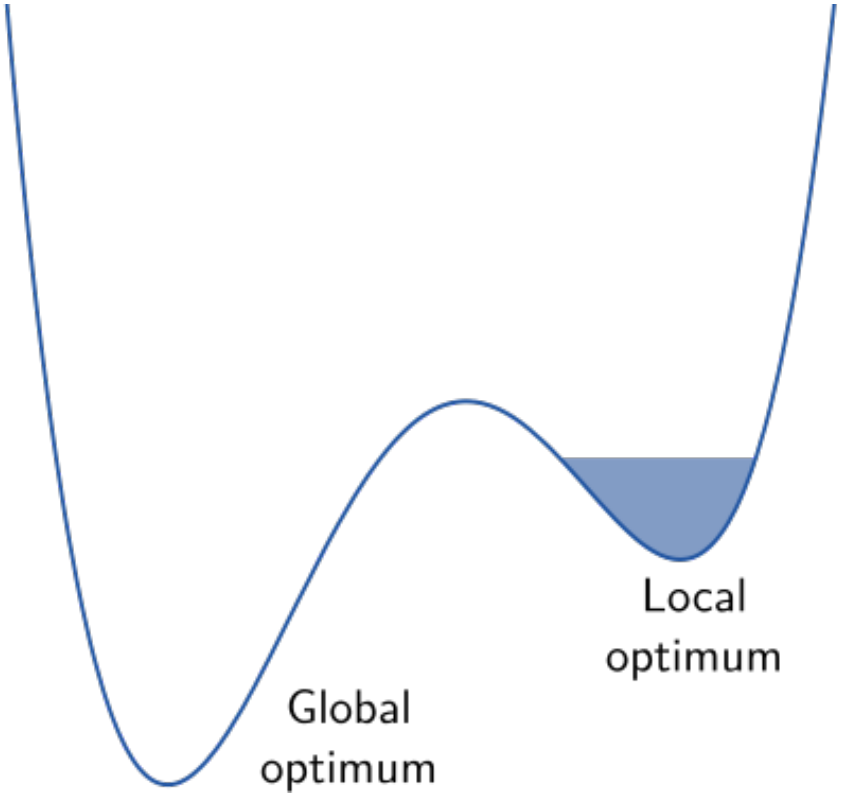


(a) Fit with *DREAM*



(b) Fit with $DREAM_{(ZS)}$

Multimodal search spaces present difficulties



- The search can get temporarily stuck in non-global optima
  - Called *metastability*
  - It is difficult to identify a metastable chain
- Non-optimal modes may not be preserved
  - Other modes affect confidence regions
  - *DREAM* can "tunnel" through potential barriers revisit modes

Acknowledgements

This project would not have been possible without the support of my mentor Paul Kienzle. I would also like to acknowledge the support of NCNR SURF directors Joe Dura and Julie Borchers. The SURF program at the NCNR is sponsored in part by the National Science Foundation through a CHRNS (Center for High Resolution Neutron Scattering) grant.