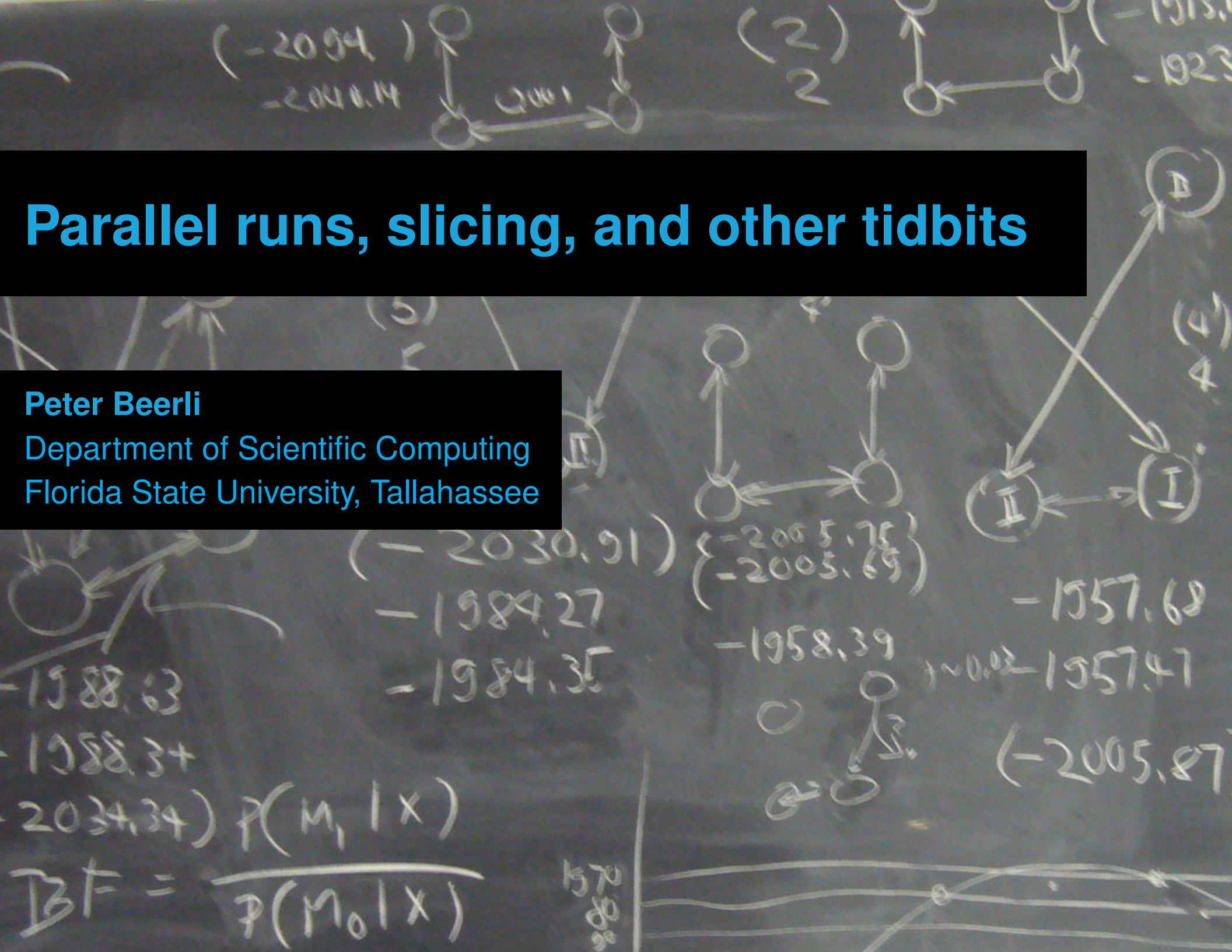


# Parallel runs, slicing, and other tidbits

Peter Beerli

Department of Scientific Computing  
Florida State University, Tallahassee



# Overview

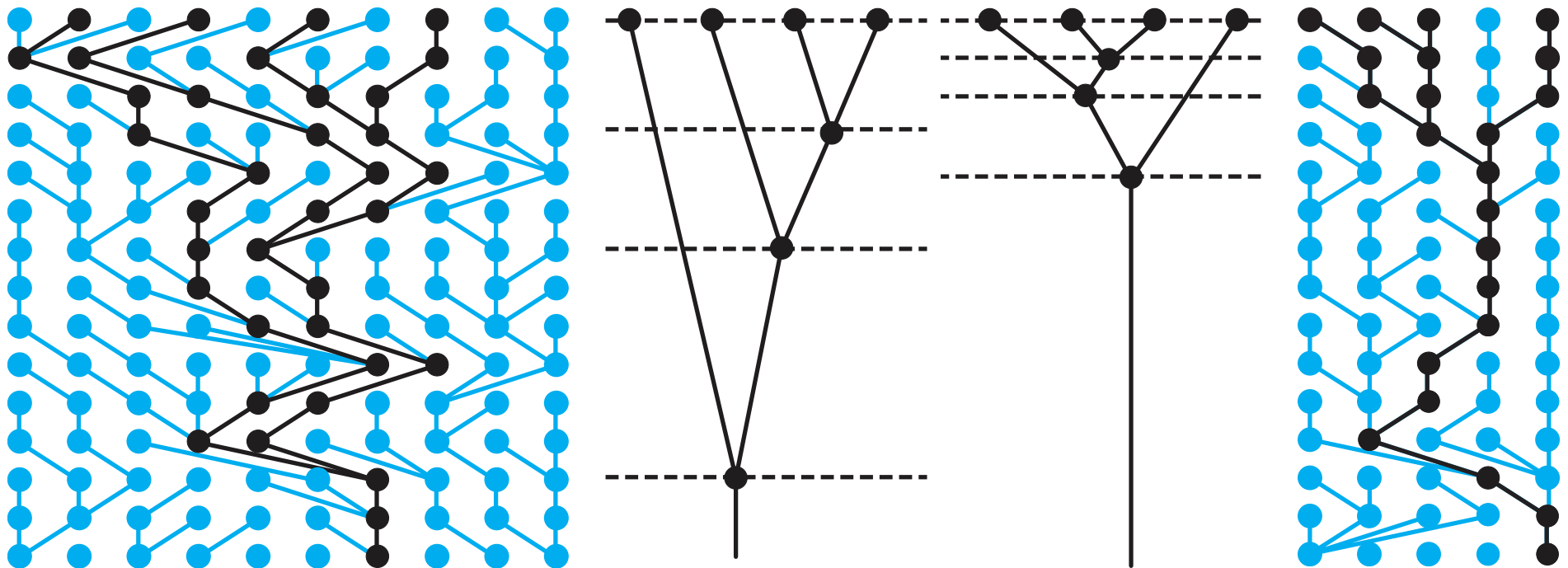
1. Changes through time affect the overall estimates
2. Slice sampling (quick)
3. Replication and parallel runtime (longer)
4. Thermodynamic integration (quick)



# Average of parameters over long time

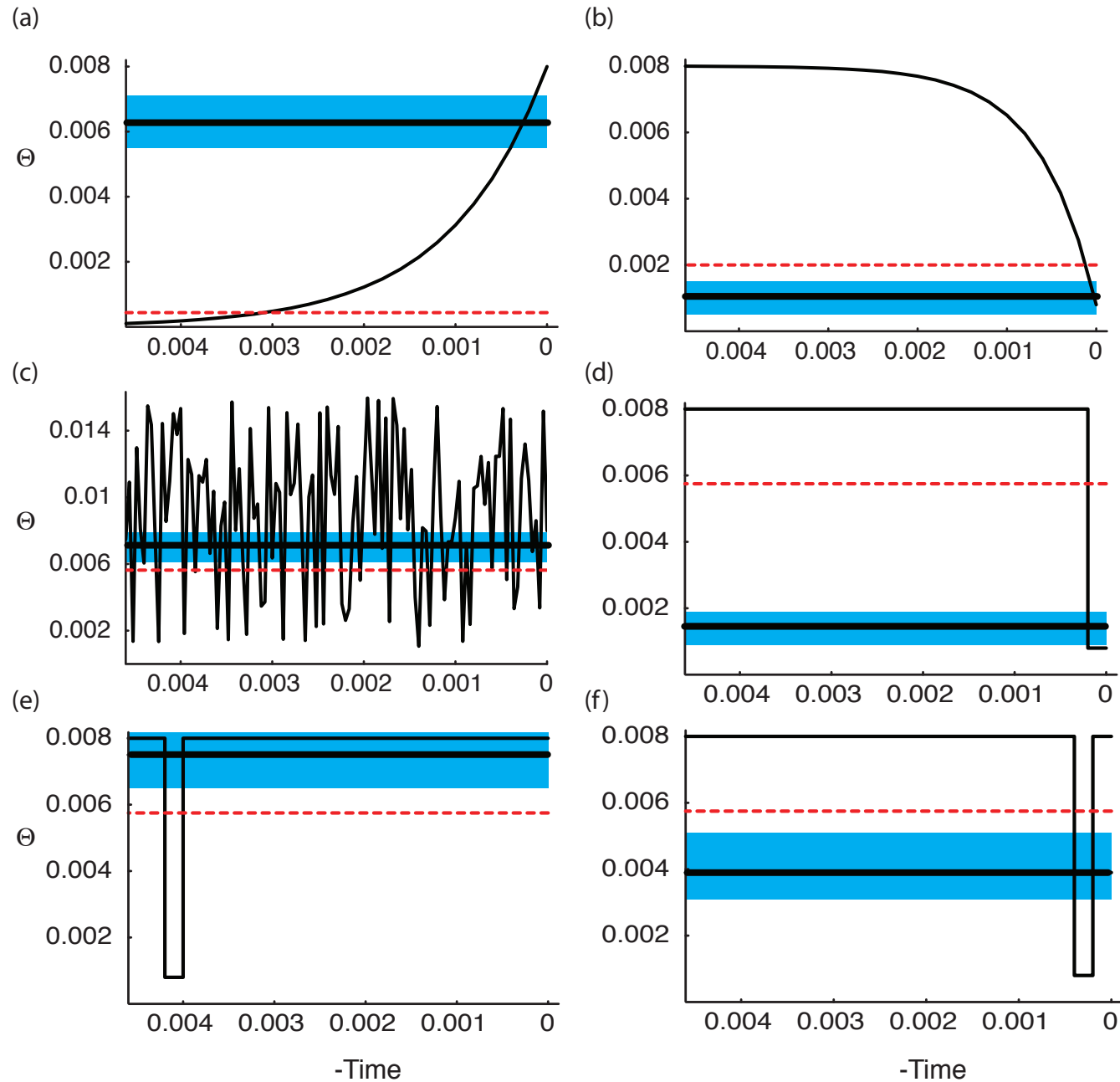
Researchers from the frequency-based camp claim that the coalescence-based methods are working on an evolutionary time-scale and therefore are not really usable in a conservation genetics or management context.

There is some truth to this claim because the time scale for the genealogies is in generations and with large populations such genealogies are deep, but ...



# Average of parameters over long time

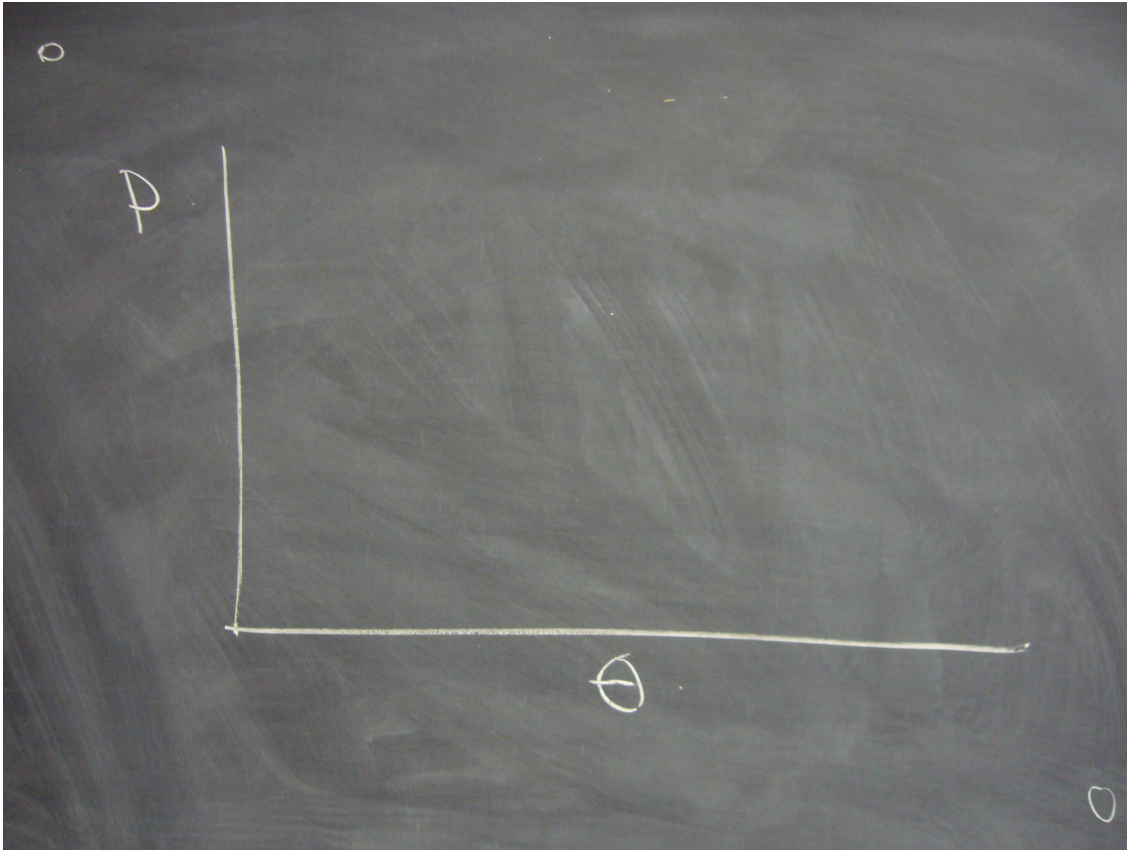
- True value
- MIGRATE estimate
- Support interval
- - - Harmonic mean





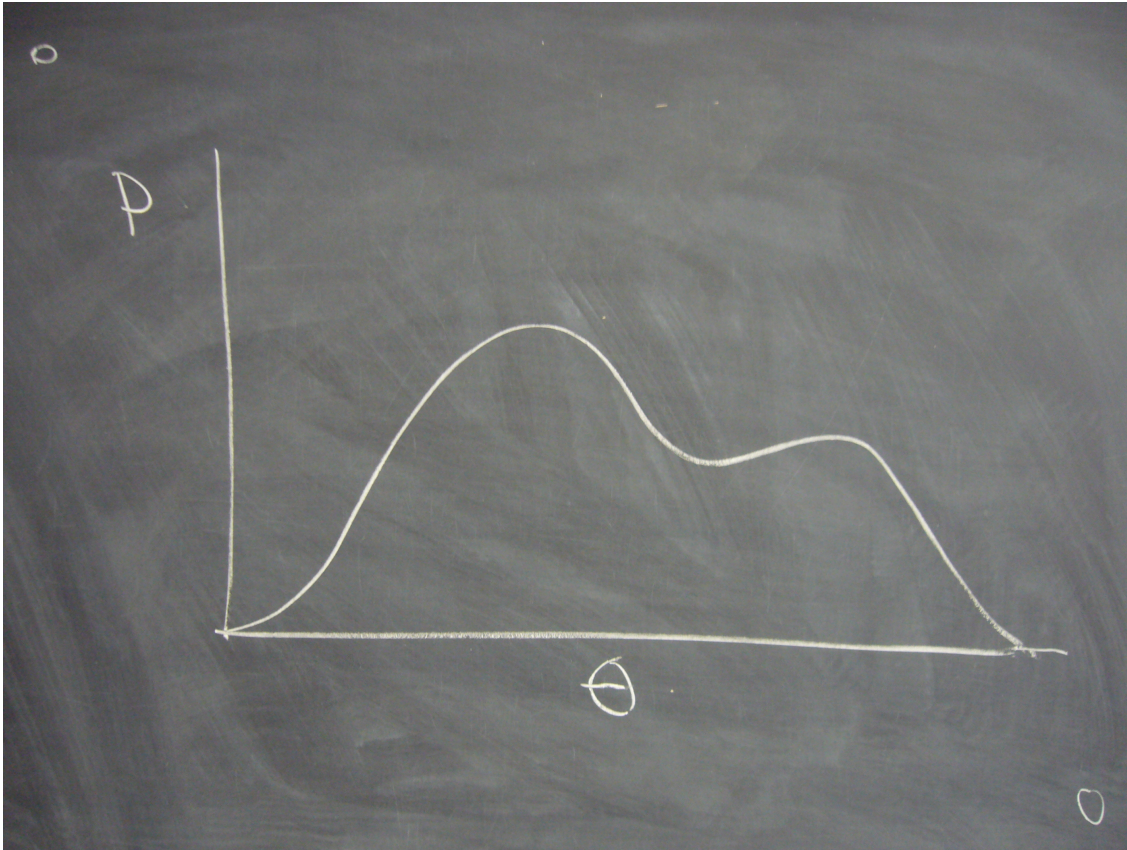


we can calculate any point on the posterior probability distribution up to a constant.

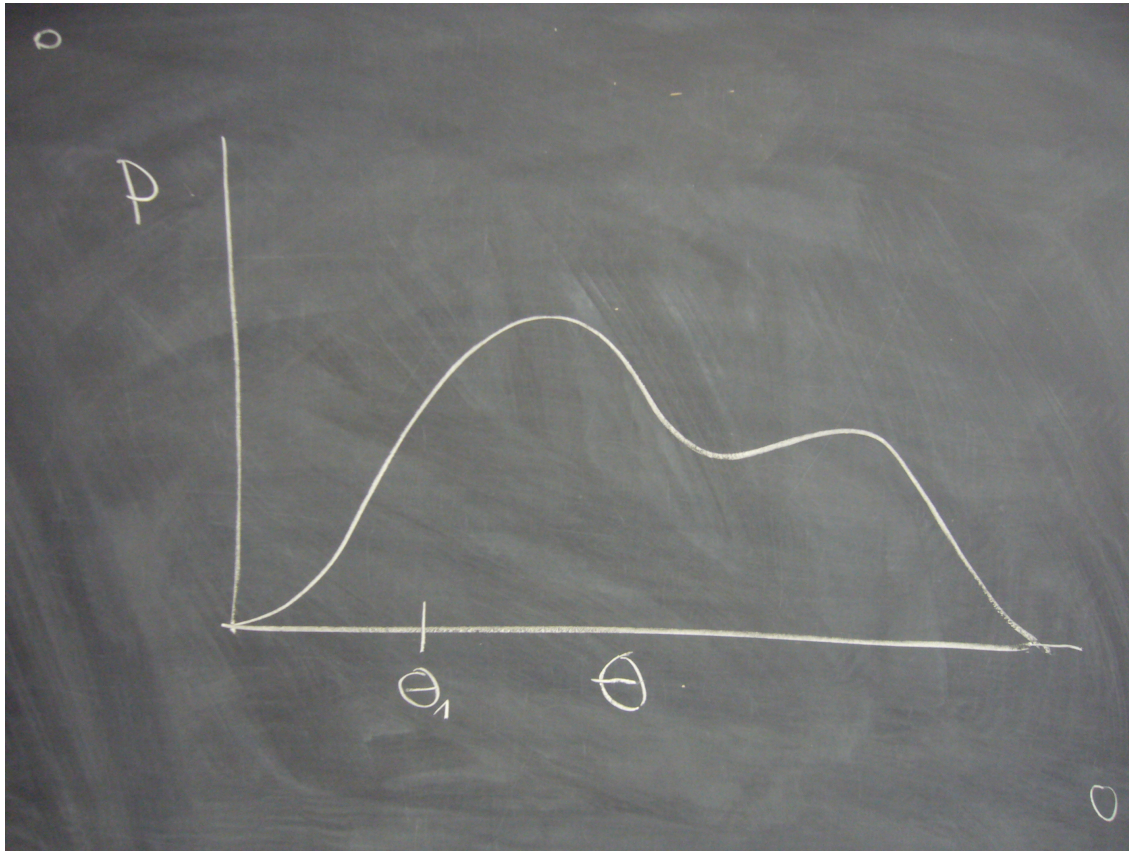


we can calculate any point on the posterior probability distribution up to a constant.



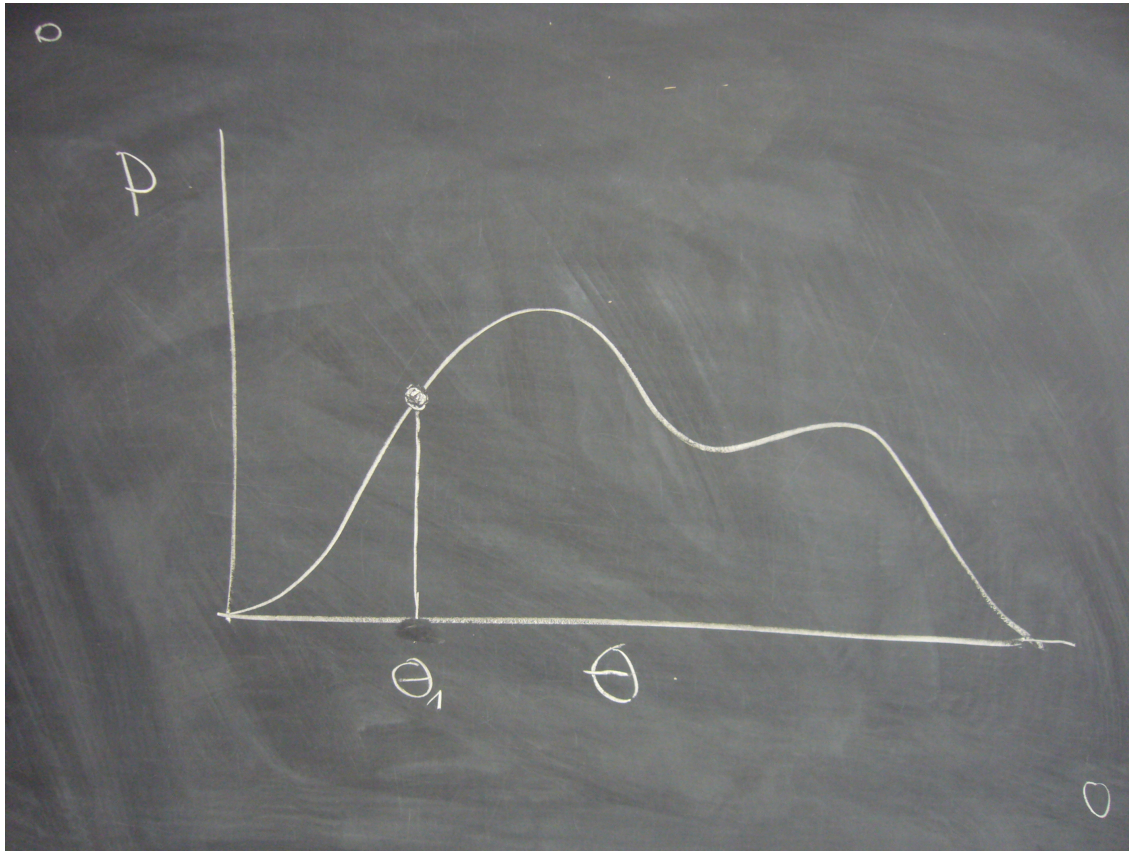


we can calculate any point on the posterior probability distribution up to a constant.

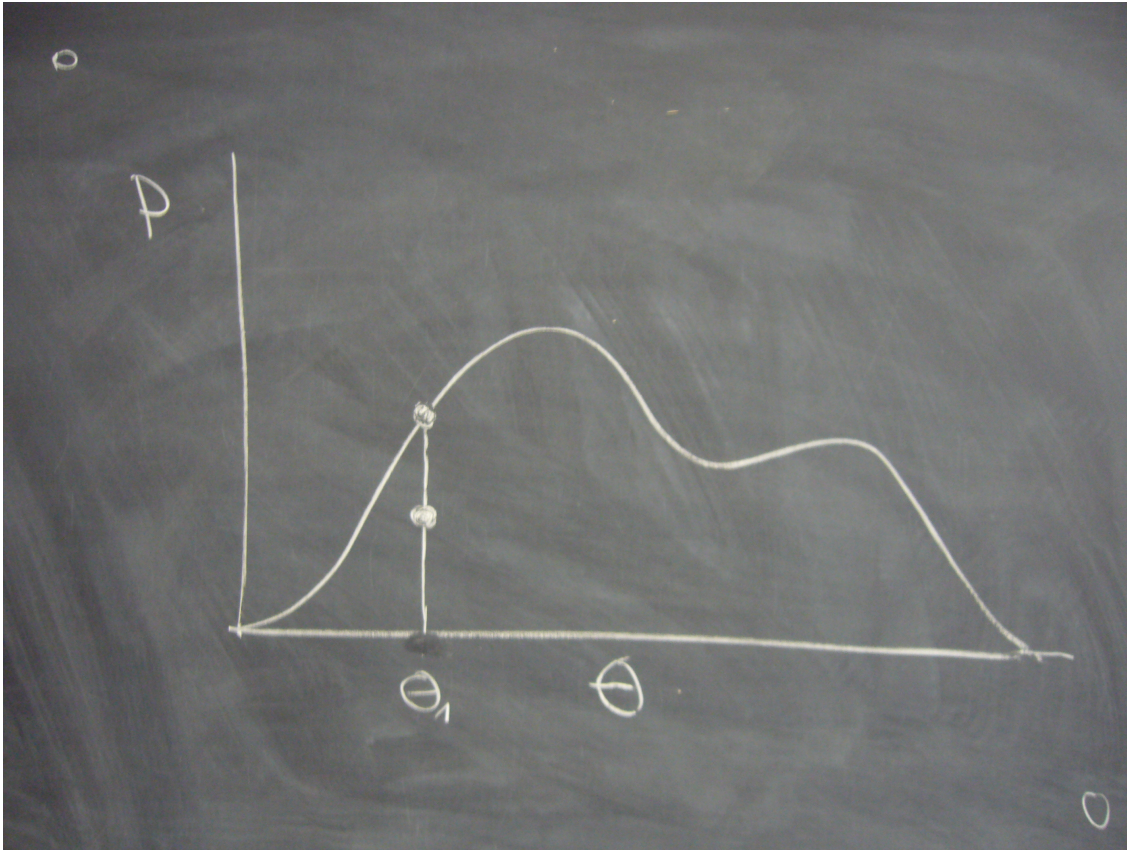


we can calculate any point on the posterior probability distribution up to a constant.



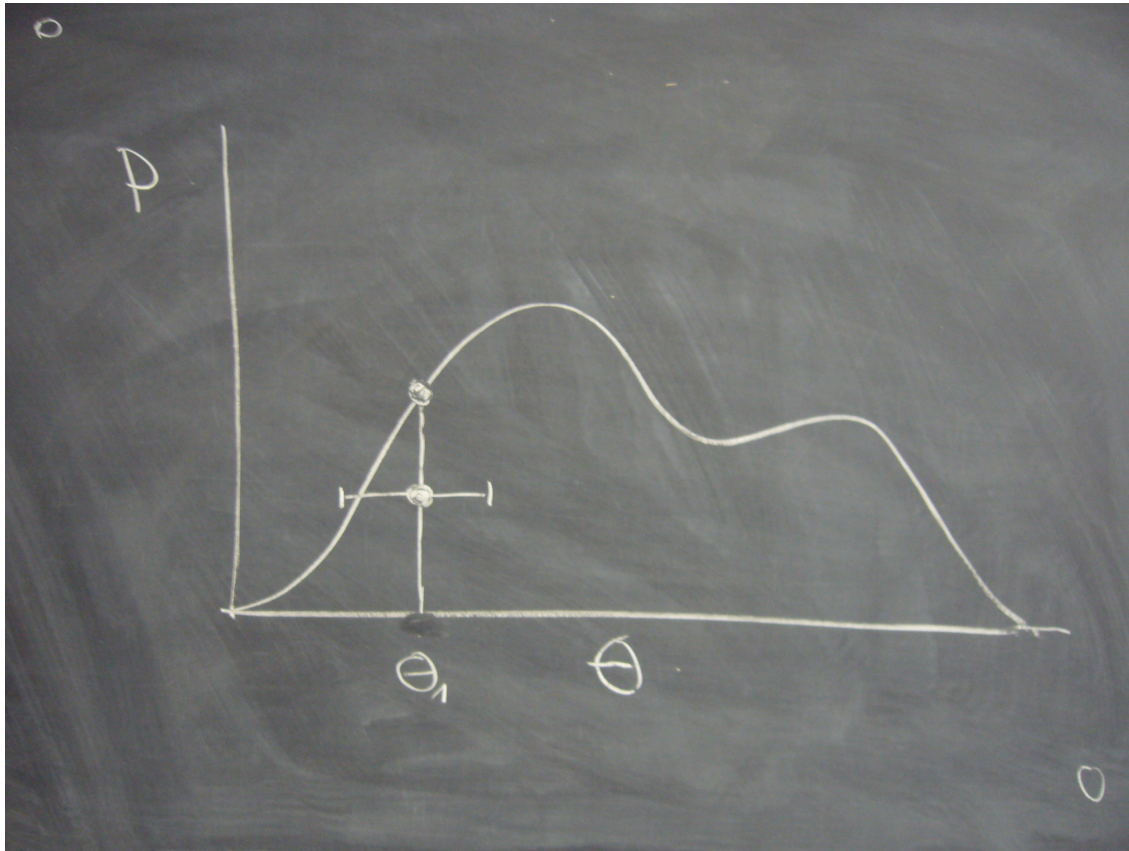


we can calculate any point on the posterior probability distribution up to a constant.

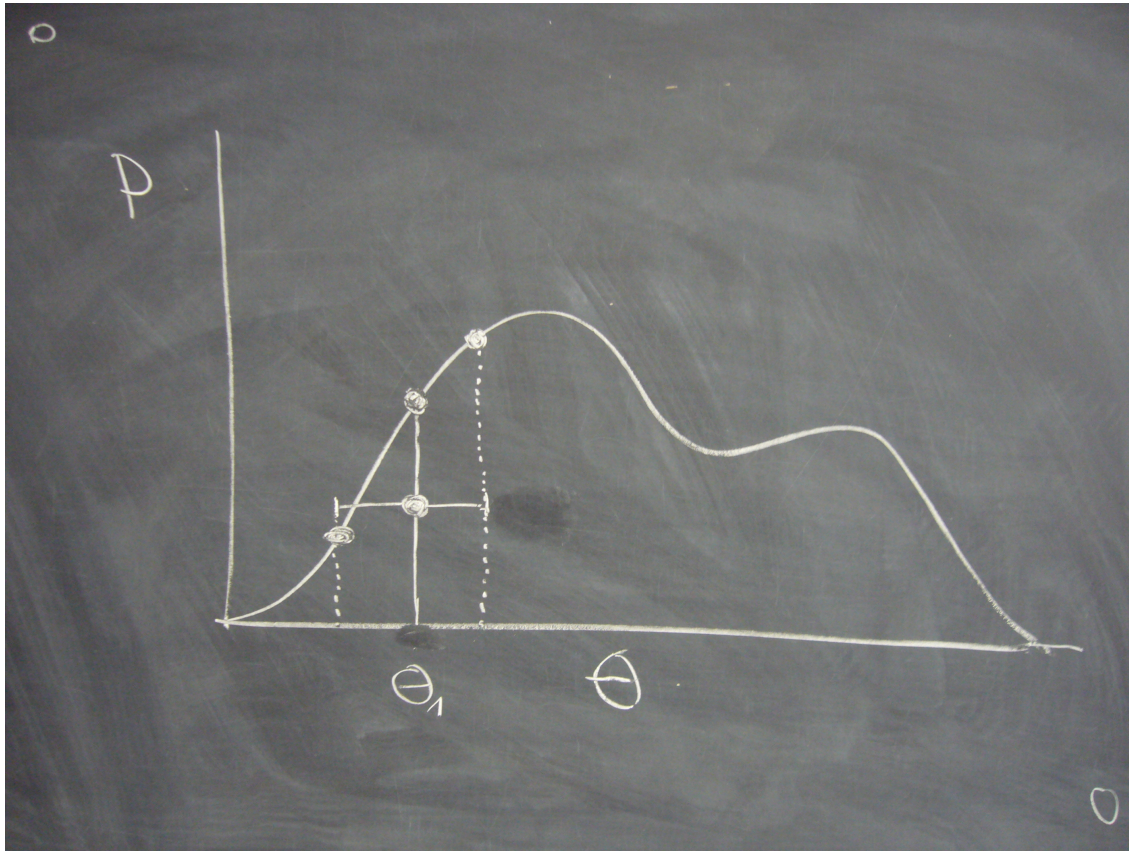


we can calculate any point on the posterior probability distribution up to a constant.



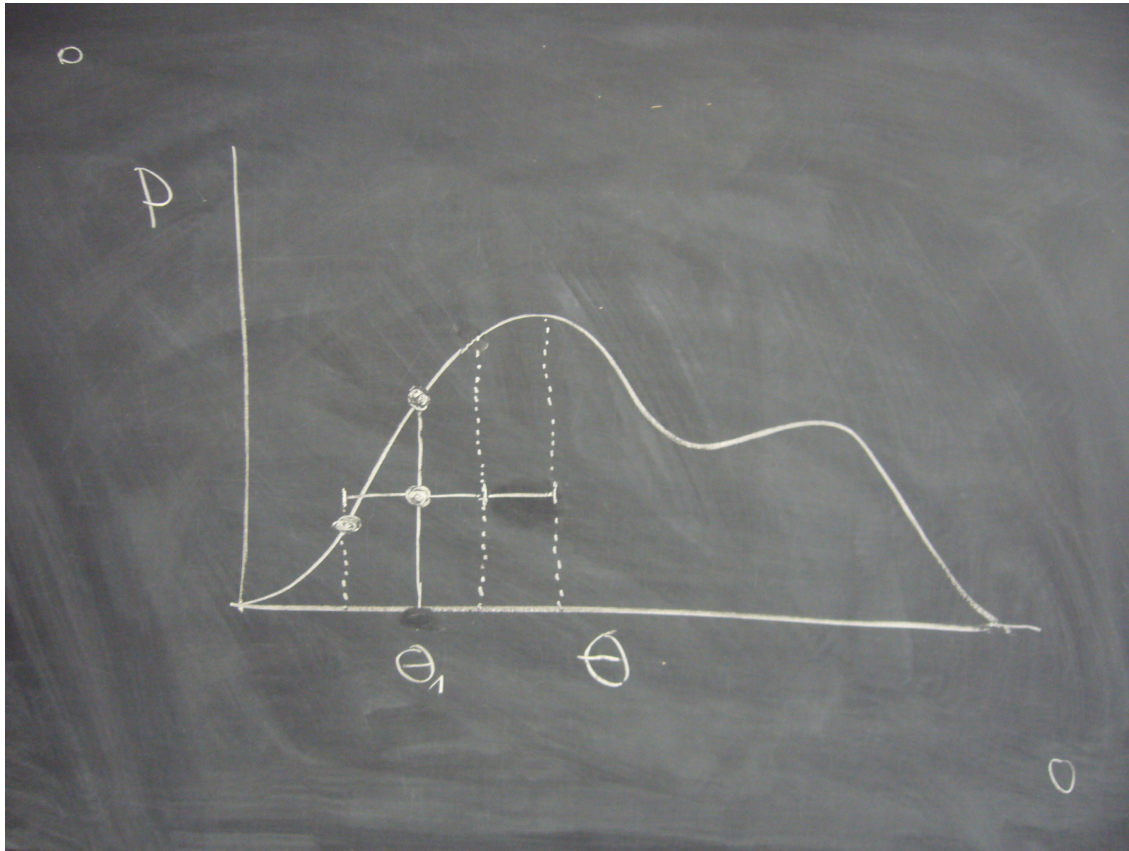


we can calculate any point on the posterior probability distribution up to a constant.

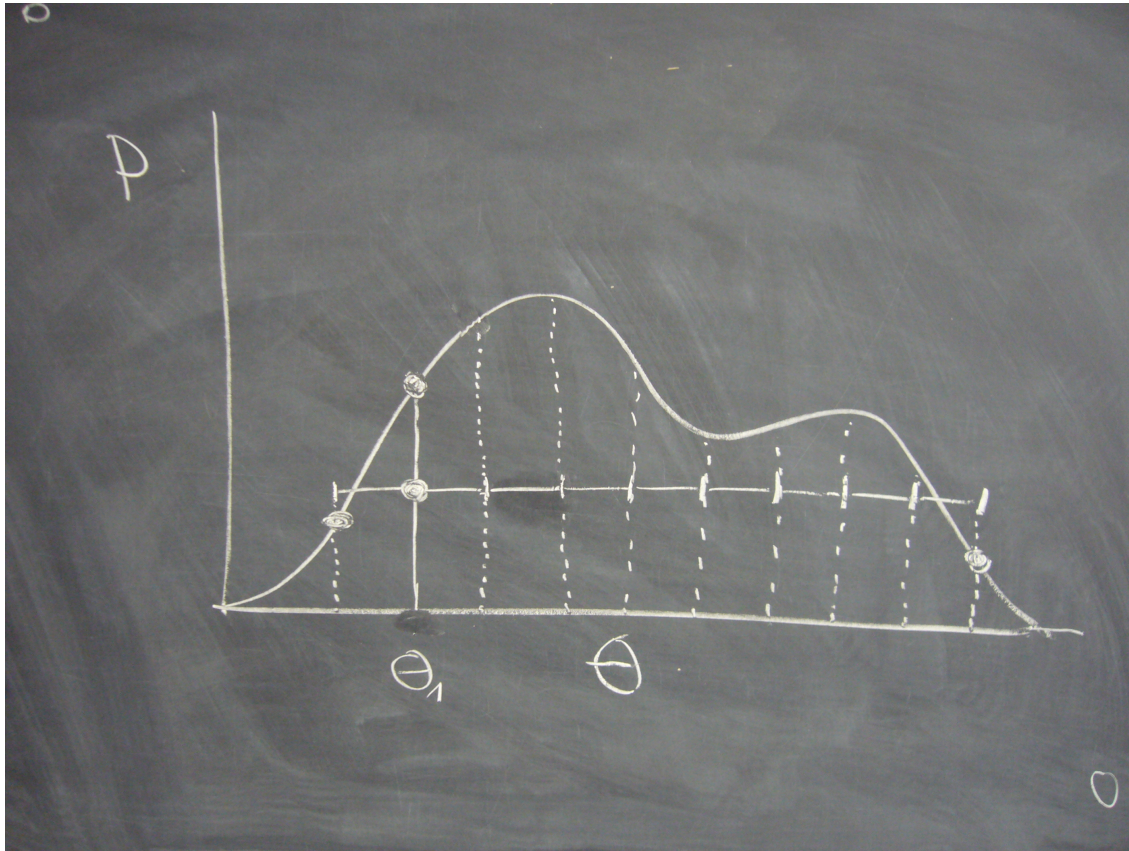


we can calculate any point on the posterior probability distribution up to a constant.



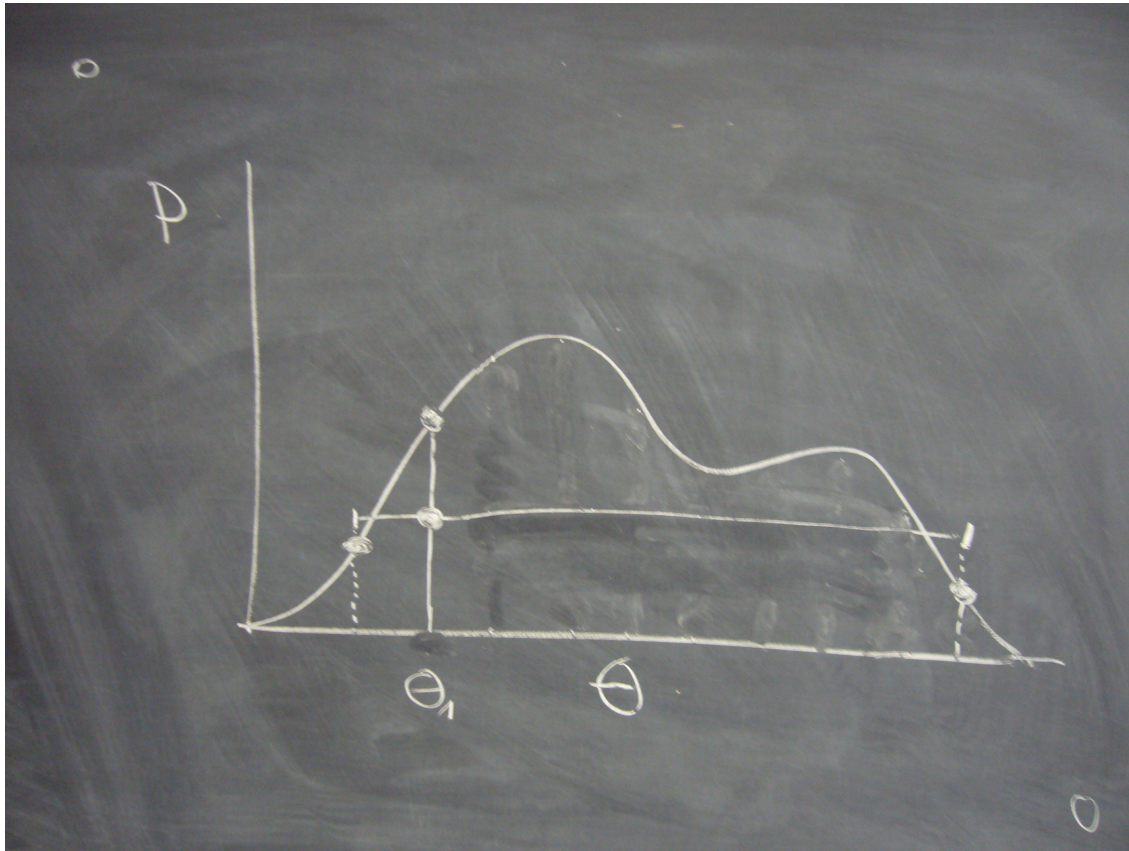


we can calculate any point on the posterior probability distribution up to a constant.

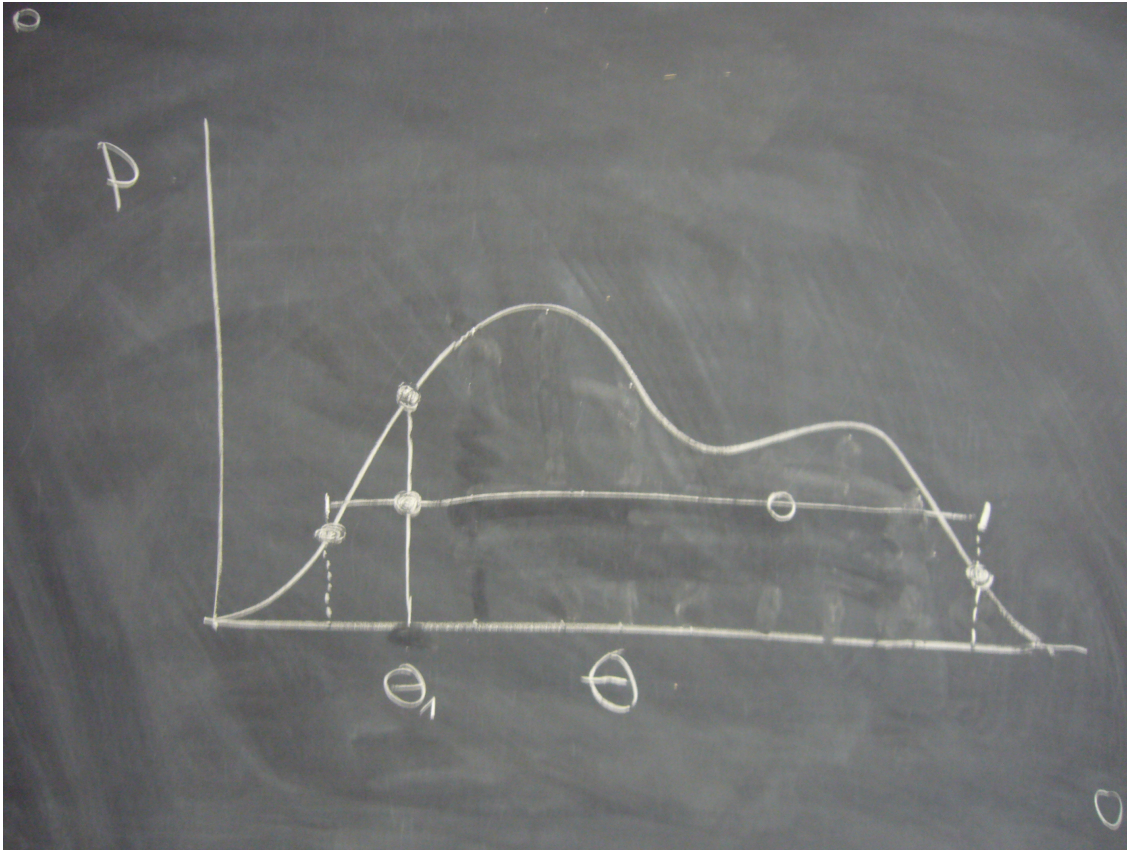


we can calculate any point on the posterior probability distribution up to a constant.



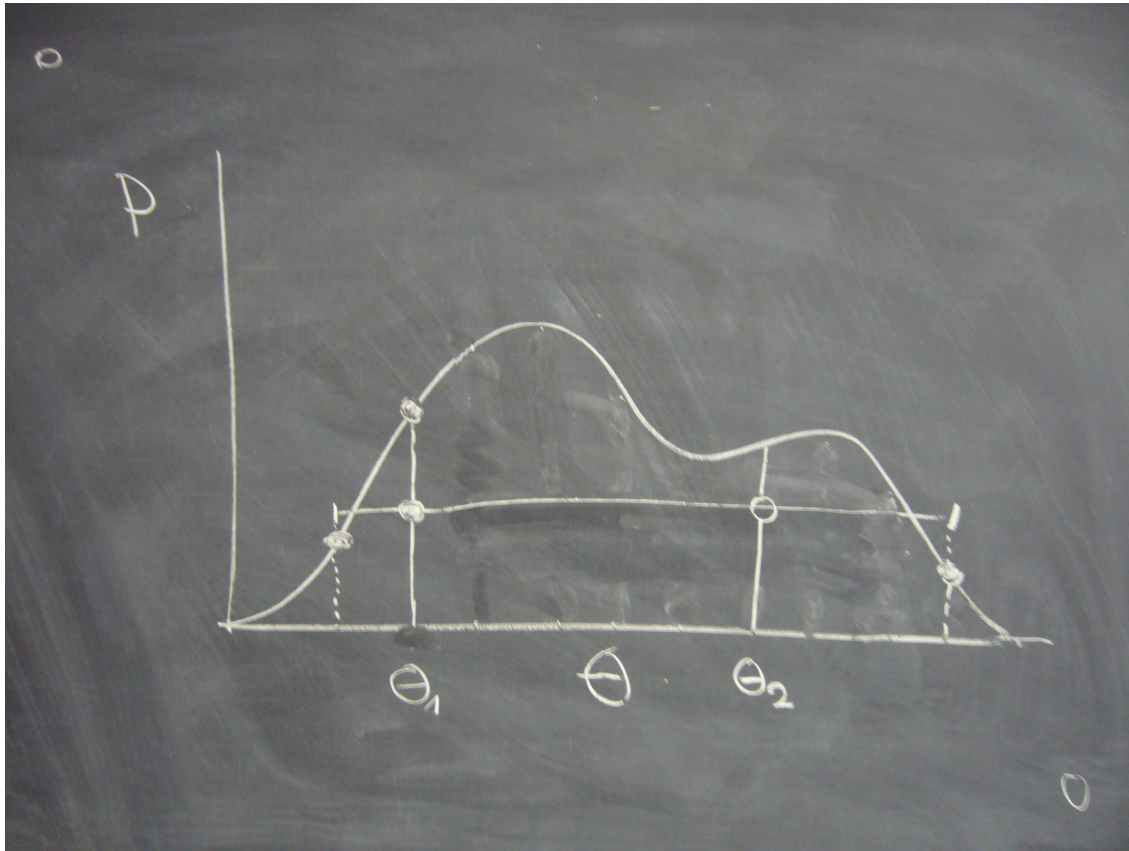


we can calculate any point on the posterior probability distribution up to a constant.

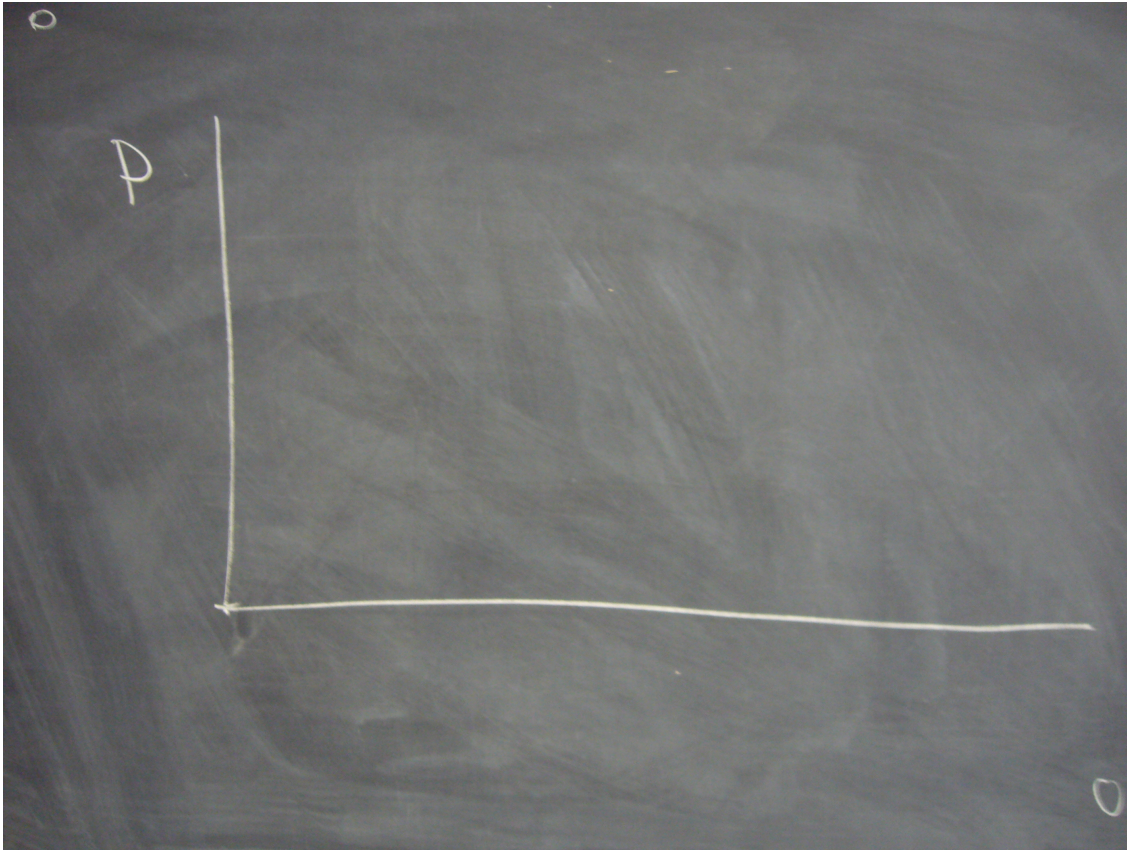


we can calculate any point on the posterior probability distribution up to a constant.



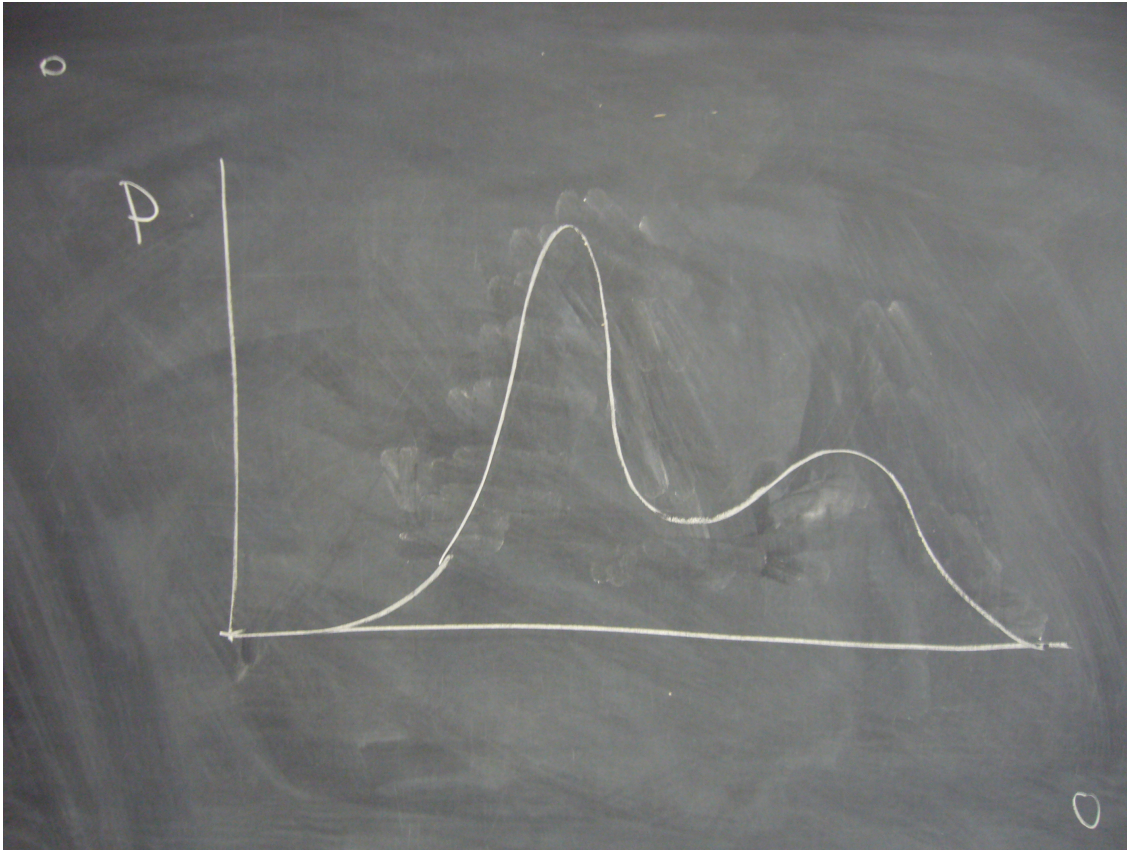


we can calculate any point on the posterior probability distribution up to a constant.



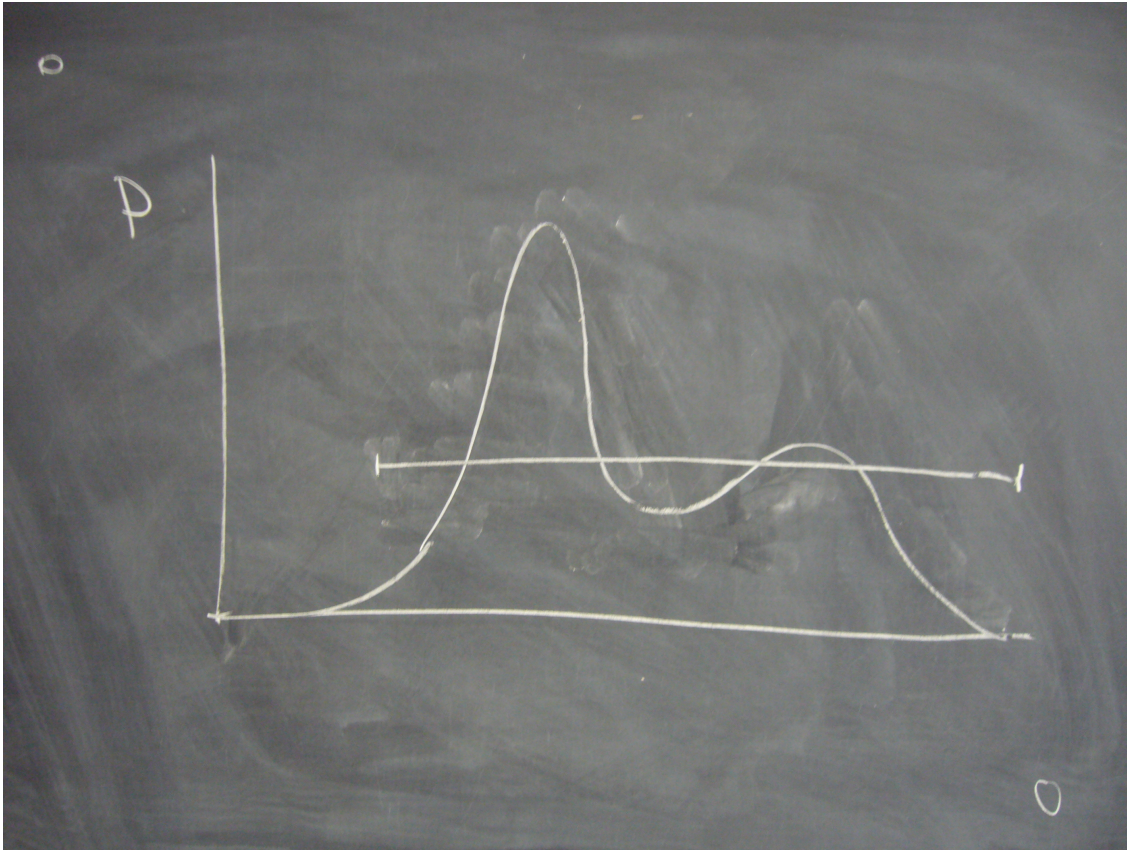
we can calculate any point on the posterior probability distribution up to a constant.



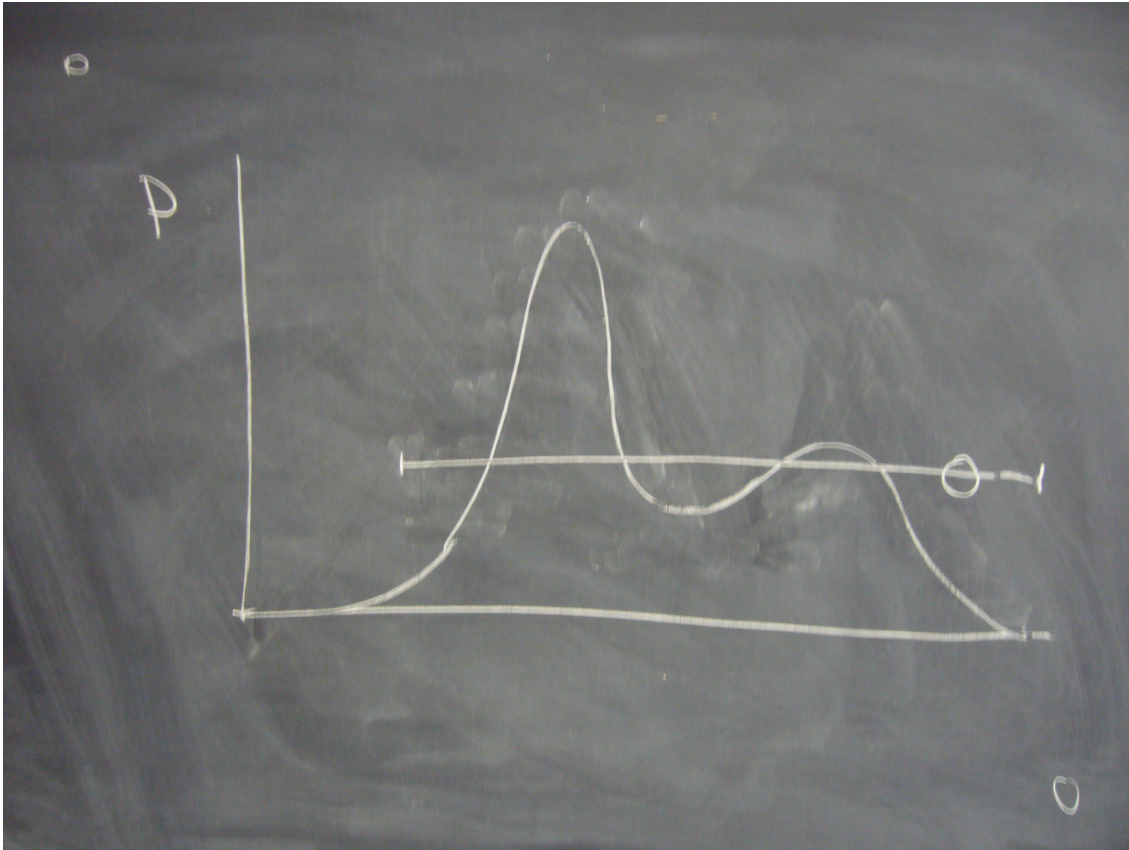


we can calculate any point on the posterior probability distribution up to a constant.



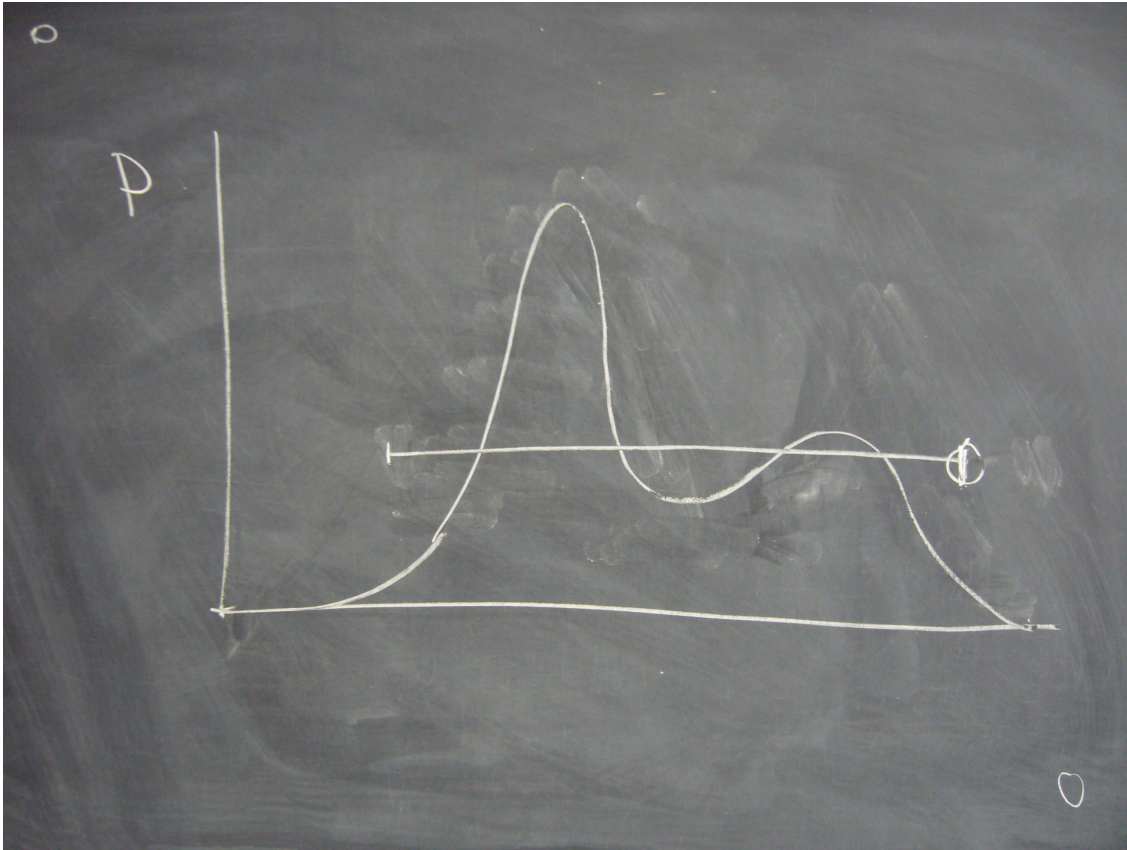


we can calculate any point on the posterior probability distribution up to a constant.



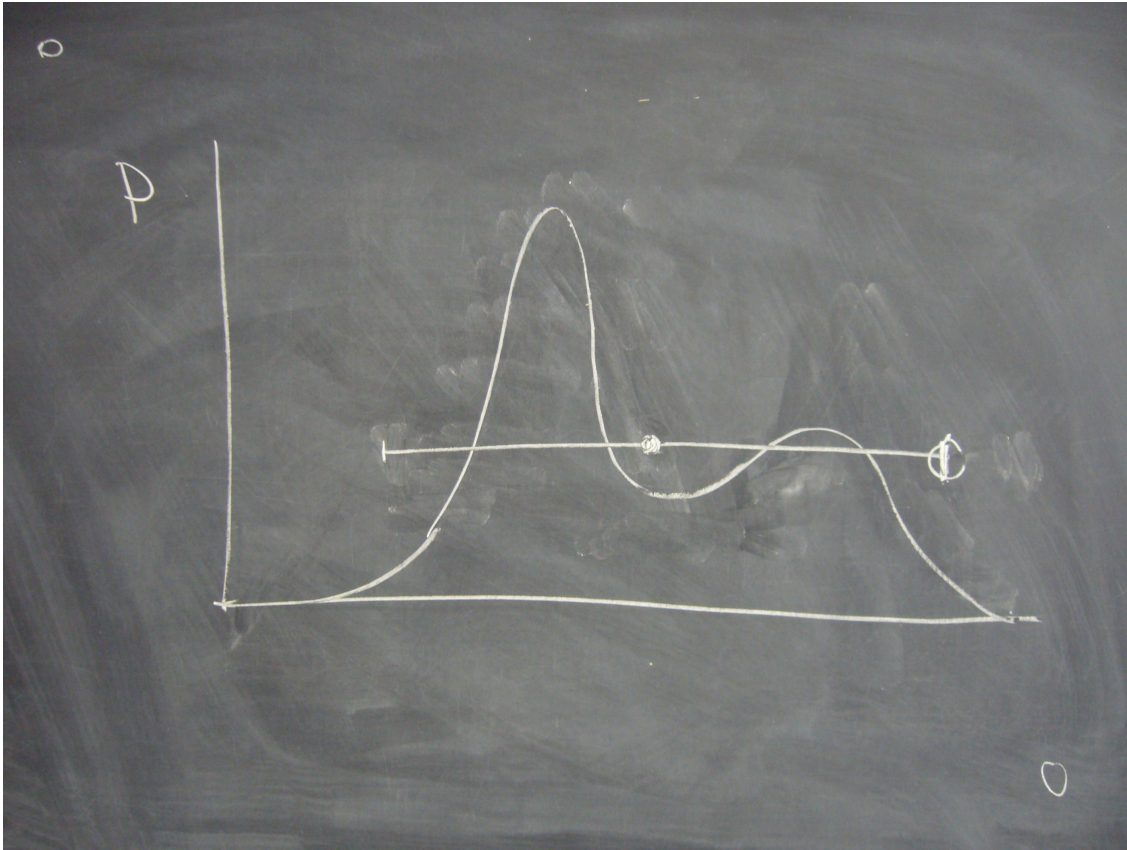
we can calculate any point on the posterior probability distribution up to a constant.





we can calculate any point on the posterior probability distribution up to a constant.



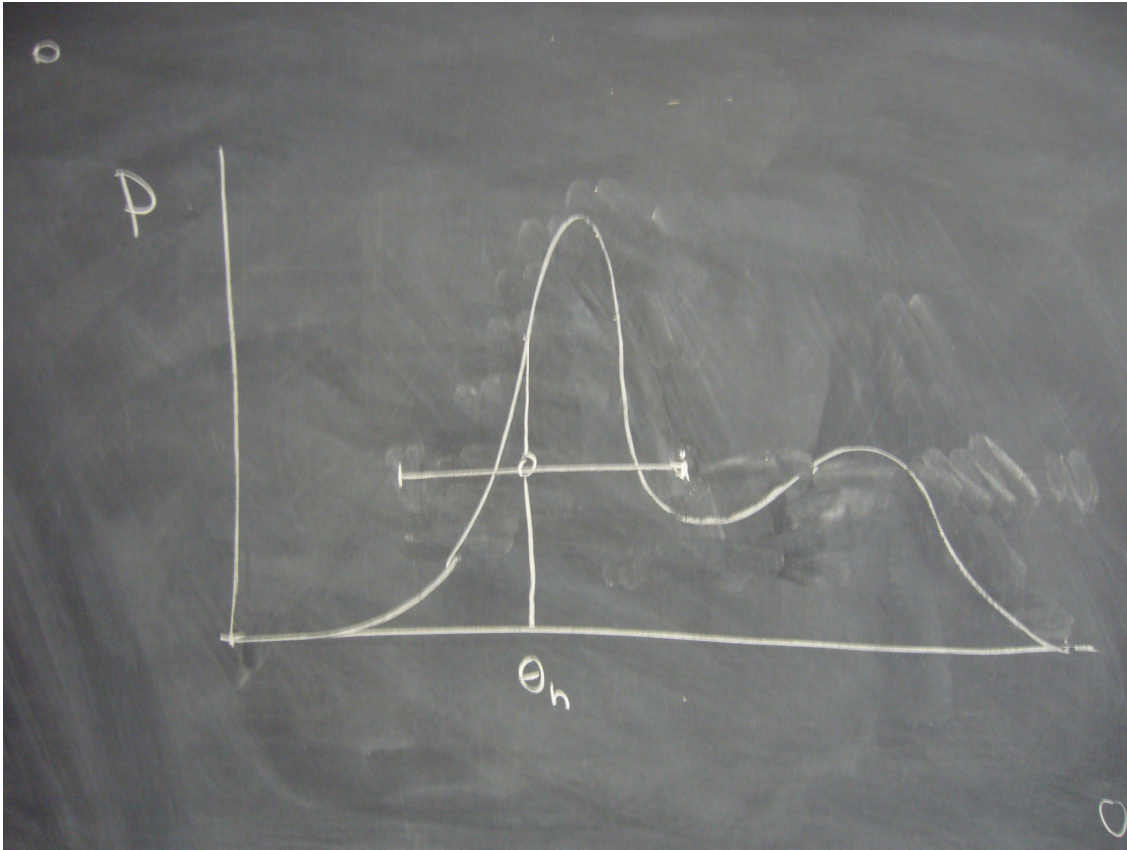


we can calculate any point on the posterior probability distribution up to a constant.



we can calculate any point on the posterior probability distribution up to a constant.





we can calculate any point on the posterior probability distribution up to a constant.



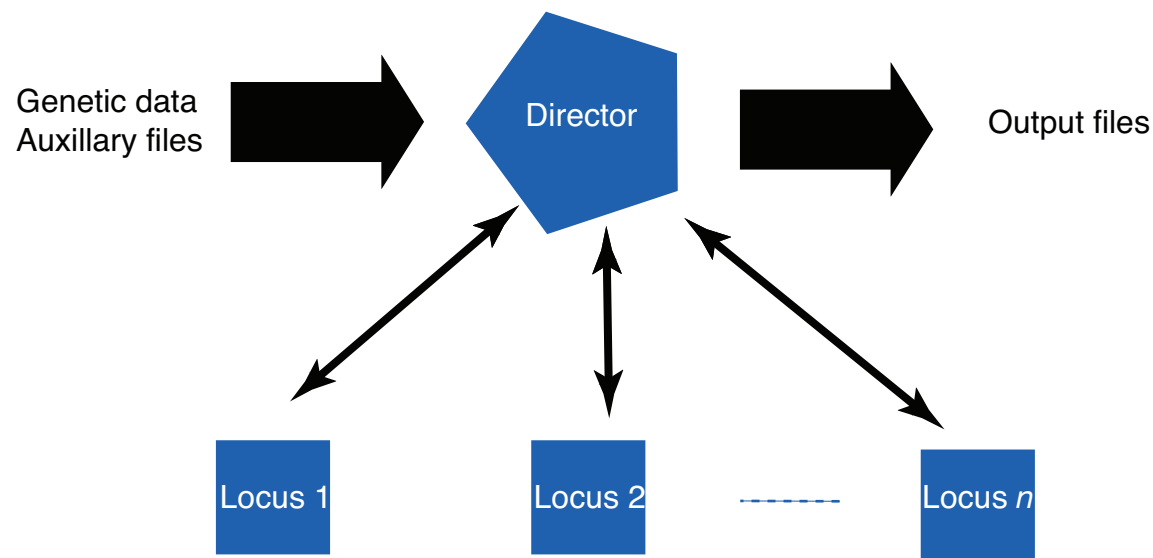


MCMC works perfectly fine when run infinitely long. It is rather difficult to know when the (finite) run has converged and is sampling from the distribution of interest **and** is reaching all important parts. Several methods are used to improve convergence and sampling:

- ◆ Improve the changing procedure between the MCMC steps (in MIGRATE: Slice sampling)
- ◆ Use Metropolis-coupled MCMC to improve finding peaks in the distribution. [this is needed to do BF]
- ◆ Program optimization can improve runtime considerably.
- ◆ Run several analyses in parallel

Each locus is completely independent, therefore can run on a different computer. Embarrassingly simple parallel computing can be done by splitting up data set and gathering “results” from individual nodes by “hand”. This gets really tedious with 100+ loci.

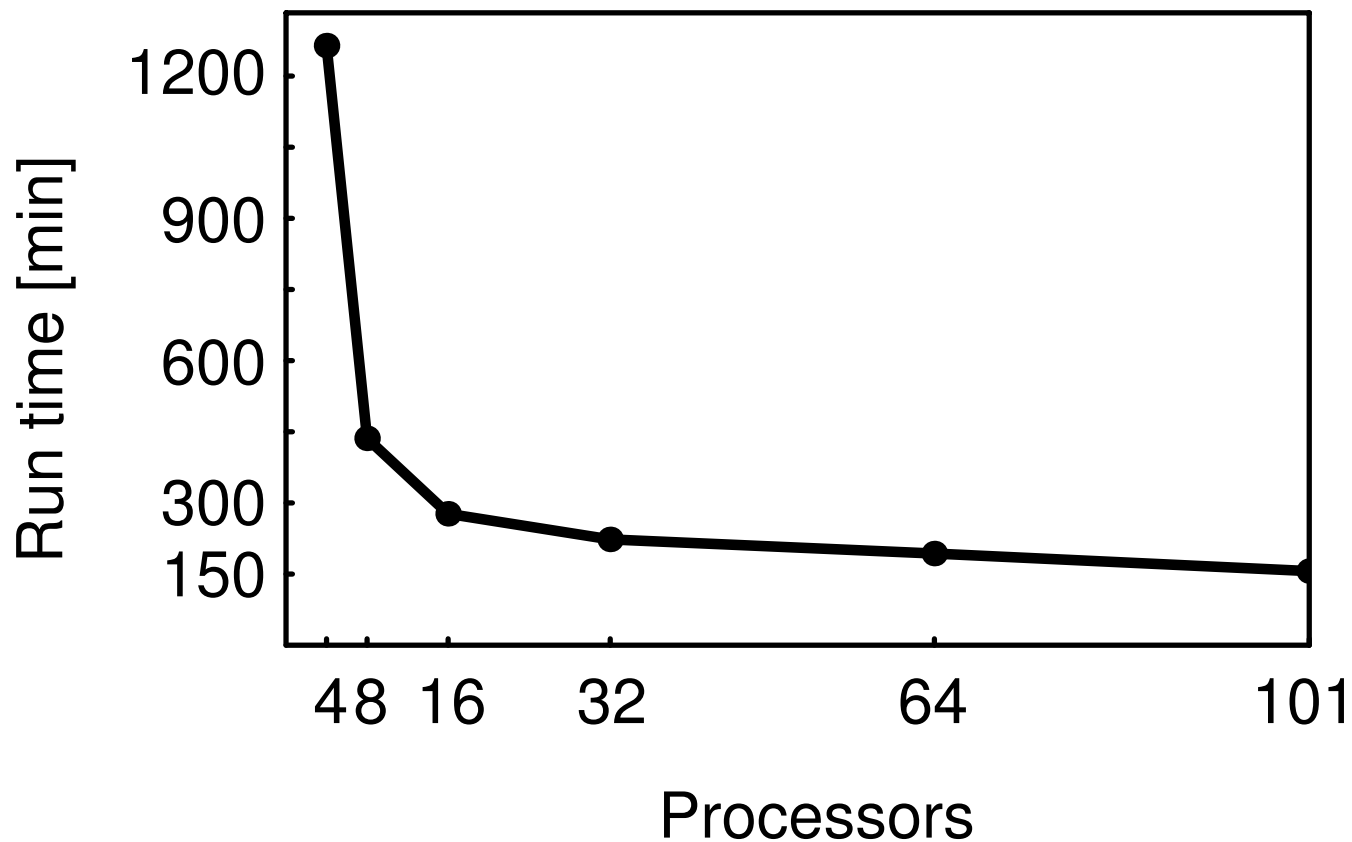
MIGRATE uses a more sophisticated strategy (MPI) and can use a cluster of (loosely) connected computer nodes. With more loci than nodes a load balancing scheme is used.





# Speed up

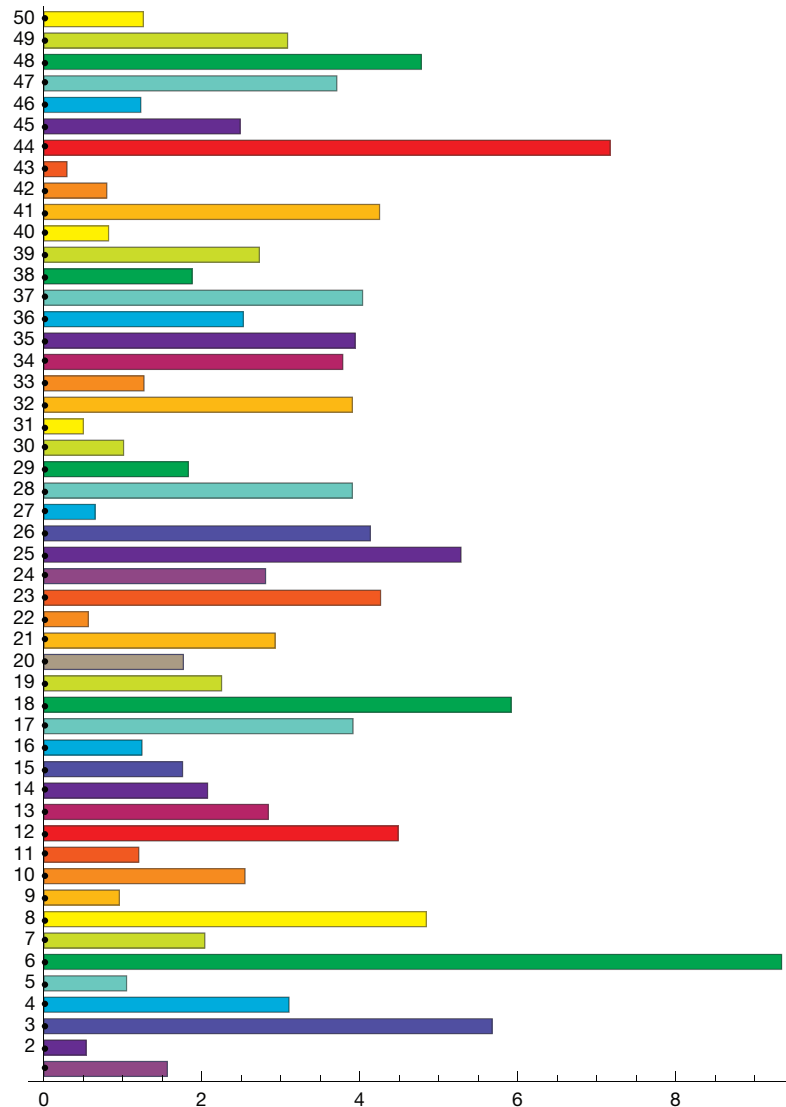
Estimation of 9 parameters in a 3 population migration model using data from a total of 100 loci, distributed over 4, 8, 16, 32, 64, 101 computer nodes.



# Speed up

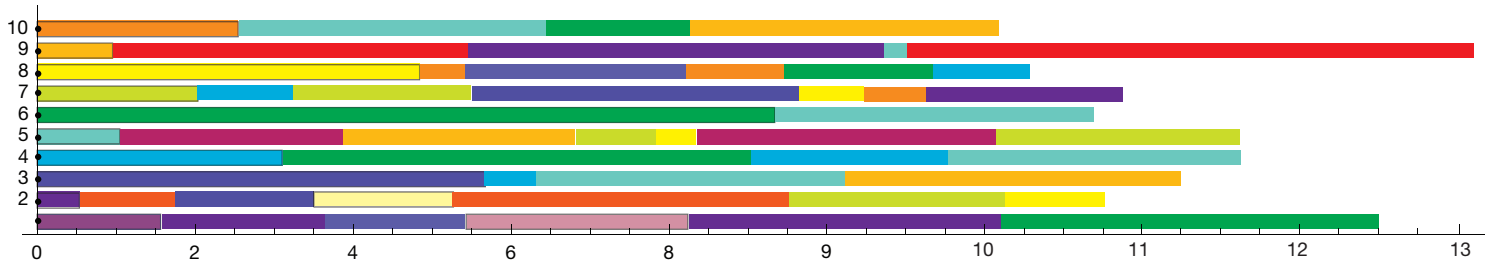
Speed of total run depends on the “slowest” locus (here out of 50)

51 nodes

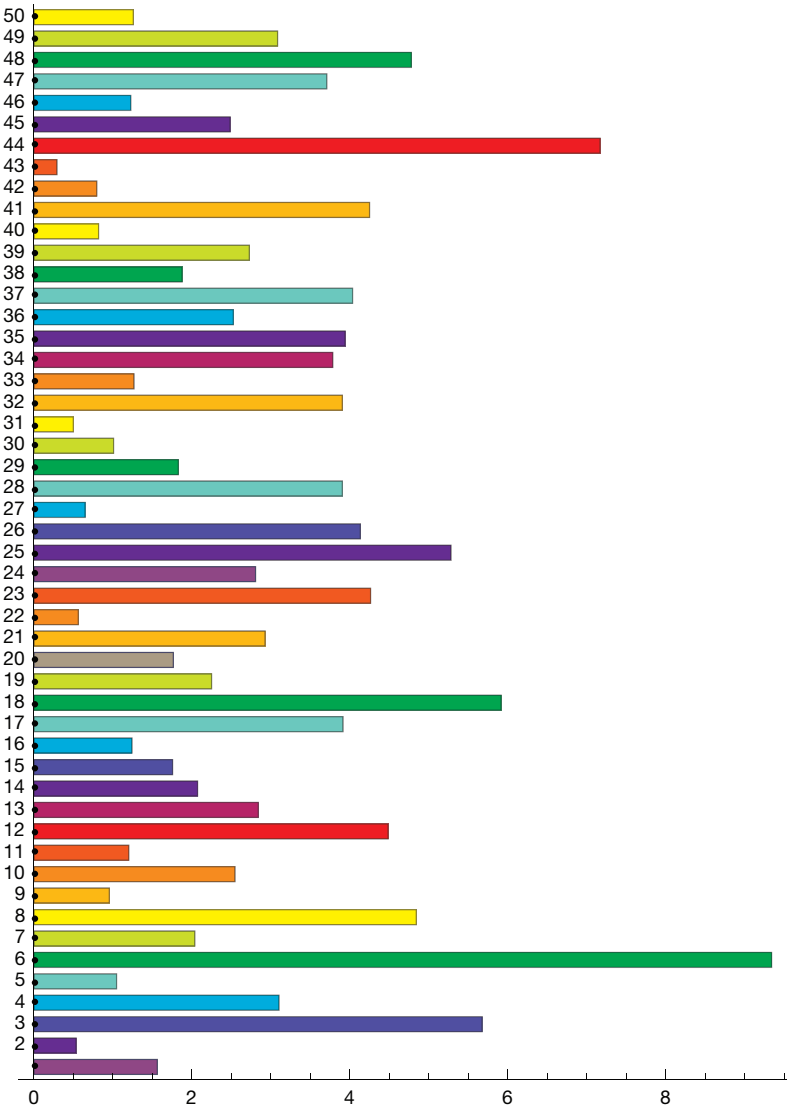


# Speed up

11 nodes



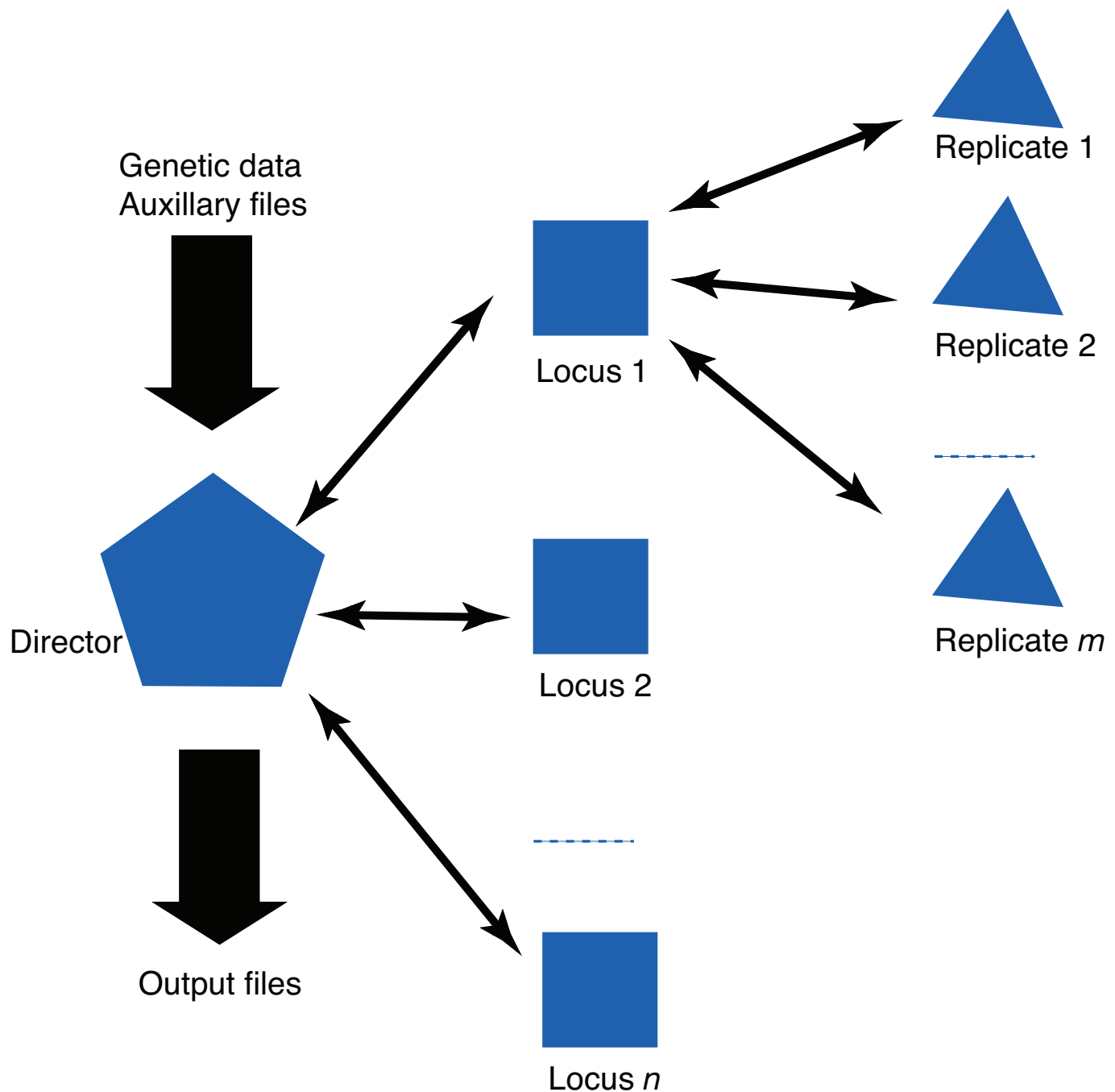
51 nodes



Load-balancing allows more effective use of all available compute nodes.



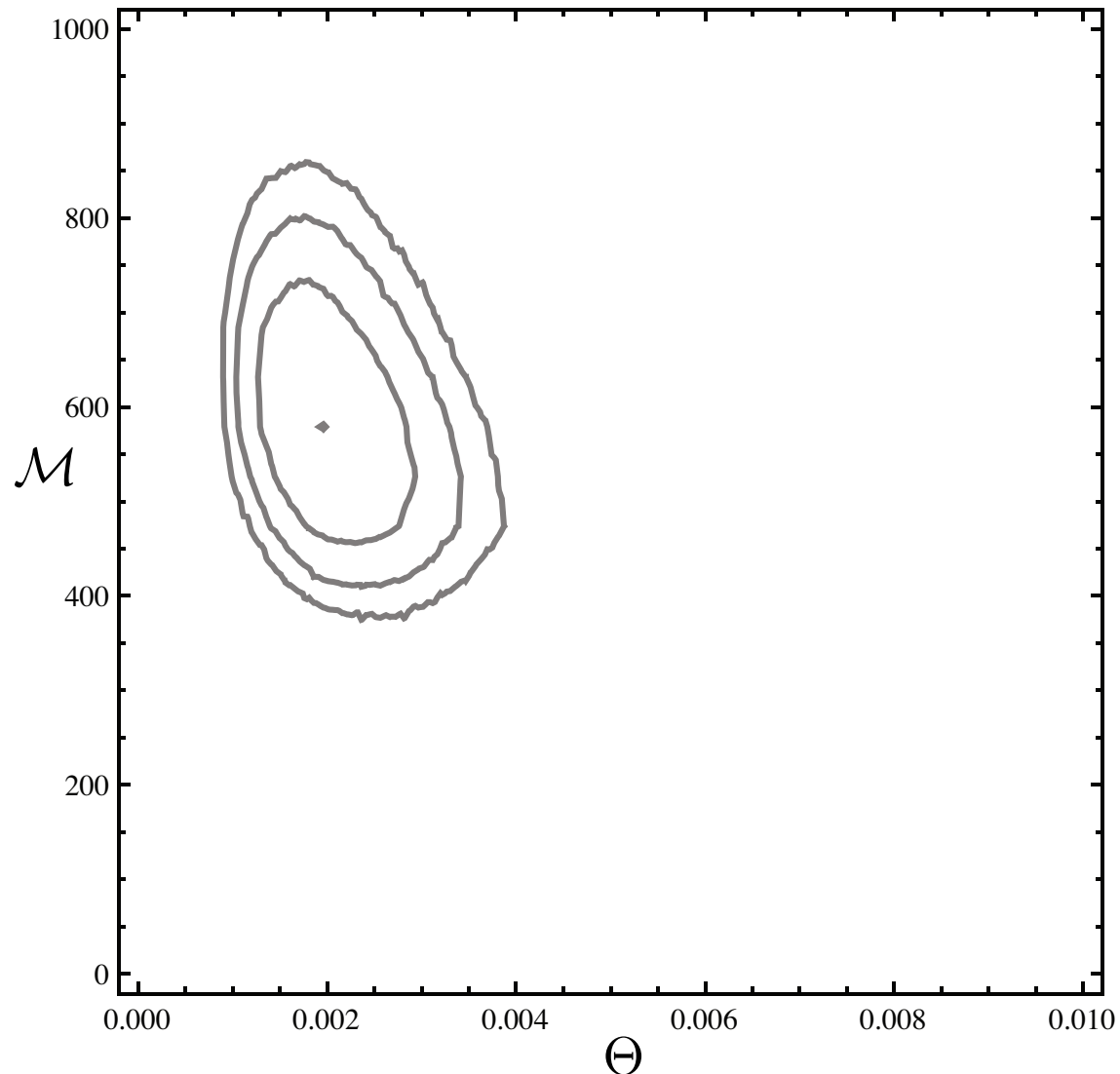
# Speed up even more?



# Run time versus accuracy

One long run

Posterior density for a 2-parameter model (population size and gene flow) A run for  $50 \times 10^6$  steps (sampling 3 quantities: 2 parameters and genealogies) took about 20 hours.



One long run:  
all samples used (no burn-in)

Contour lines are at 50%, 95%, and 99% credibility level

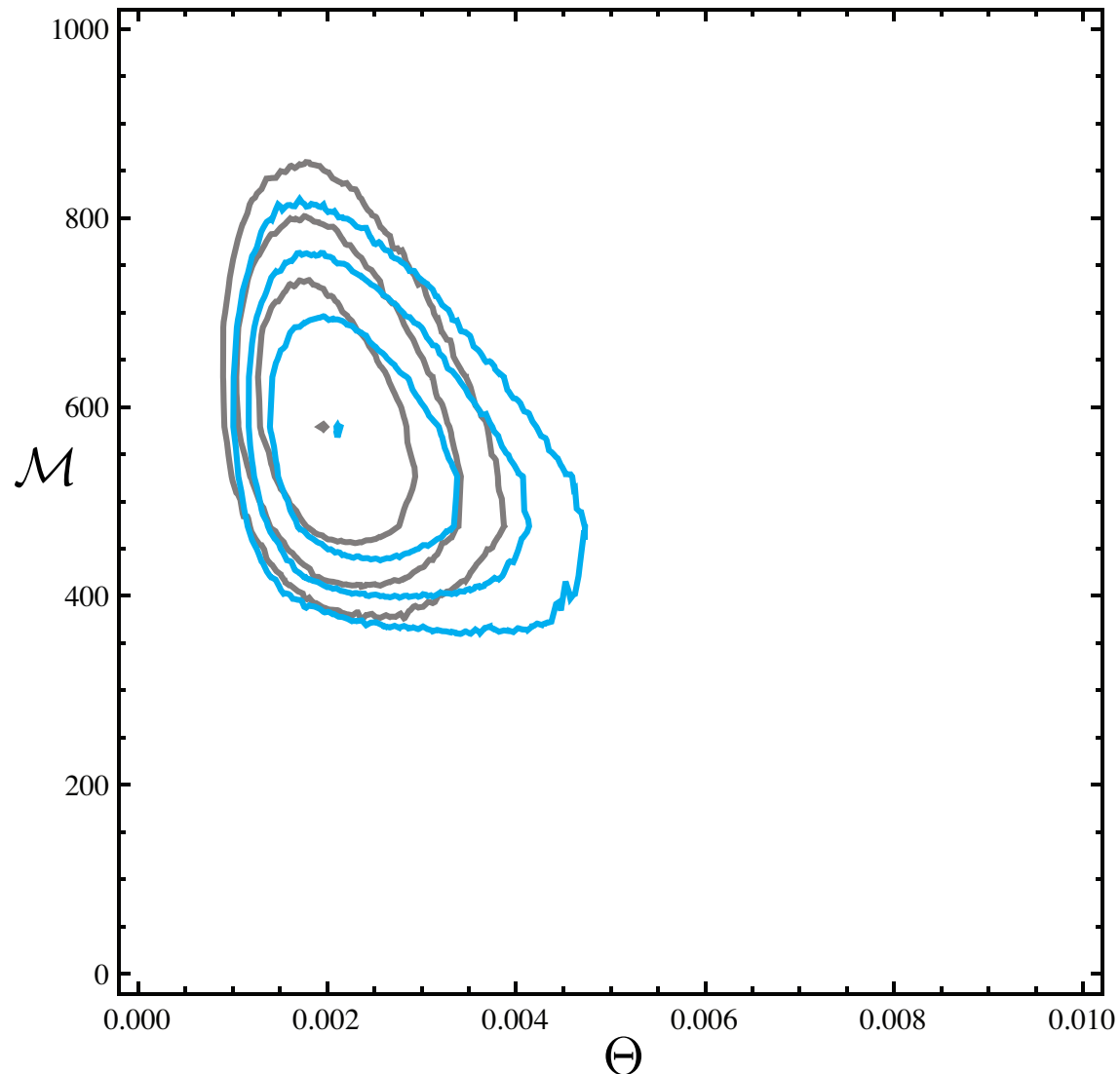
$\Theta = 4N_e\mu$  (population size scaled by mutation rate)

$\mathcal{M} = \frac{m}{\mu}$  (immigration rate scaled by mutation rate)

# Run time versus accuracy

10 replicated runs

Posterior density for a 2-parameter model (population size and gene flow) 10 runs each for  $5 \times 10^6$  steps took about 2 hours.



One long run: 20 hours  
all samples used (no burn-in)  
10 replicates: 2 hours  
all samples used (no burn-in)

Contour lines are at 50%, 95%, and 99% credibility level

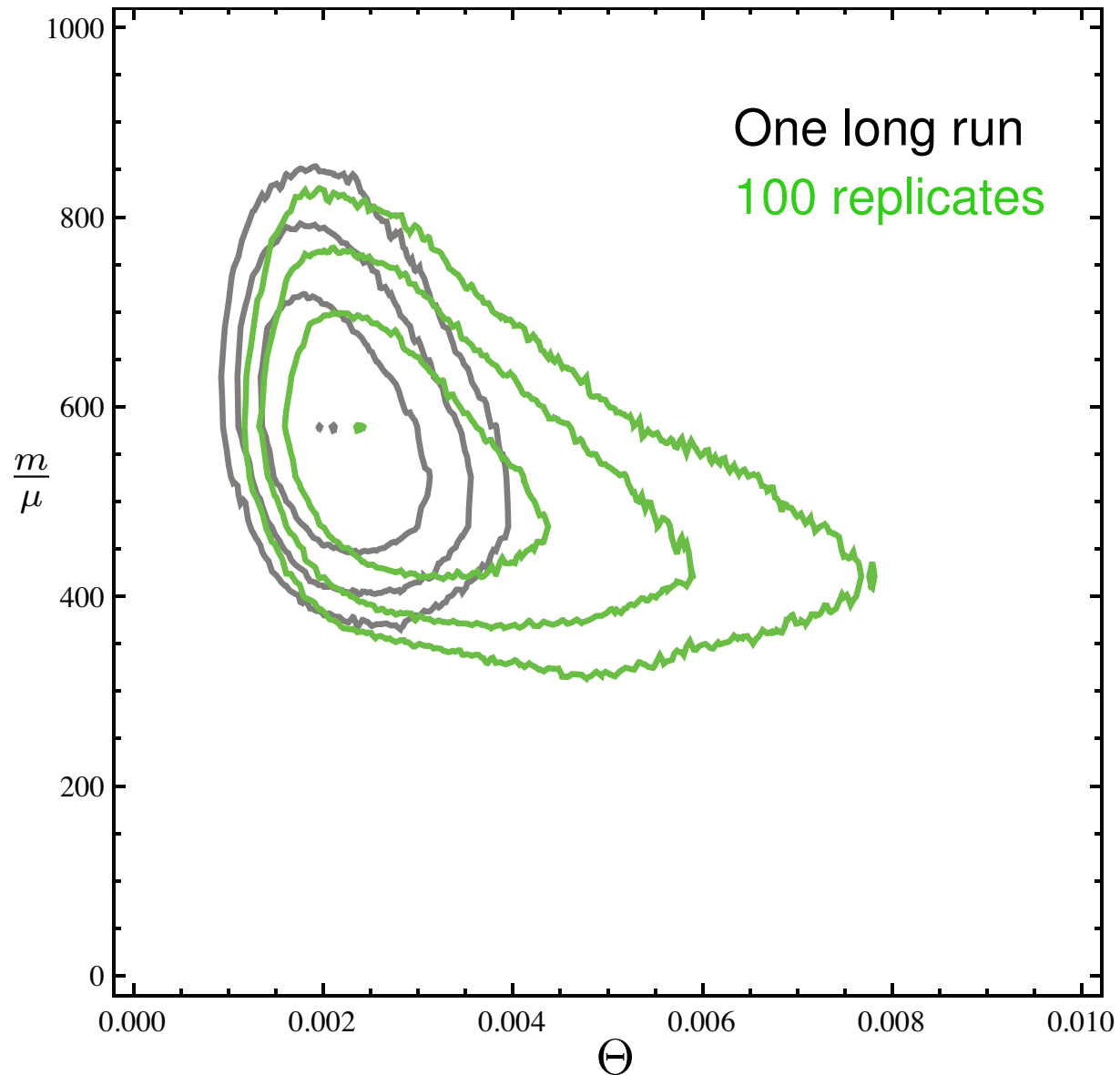
$\Theta = 4N_e\mu$  (population size scaled by mutation rate)

$\mathcal{M} = \frac{m}{\mu}$  (immigration rate scaled by mutation rate)



# Accuracy

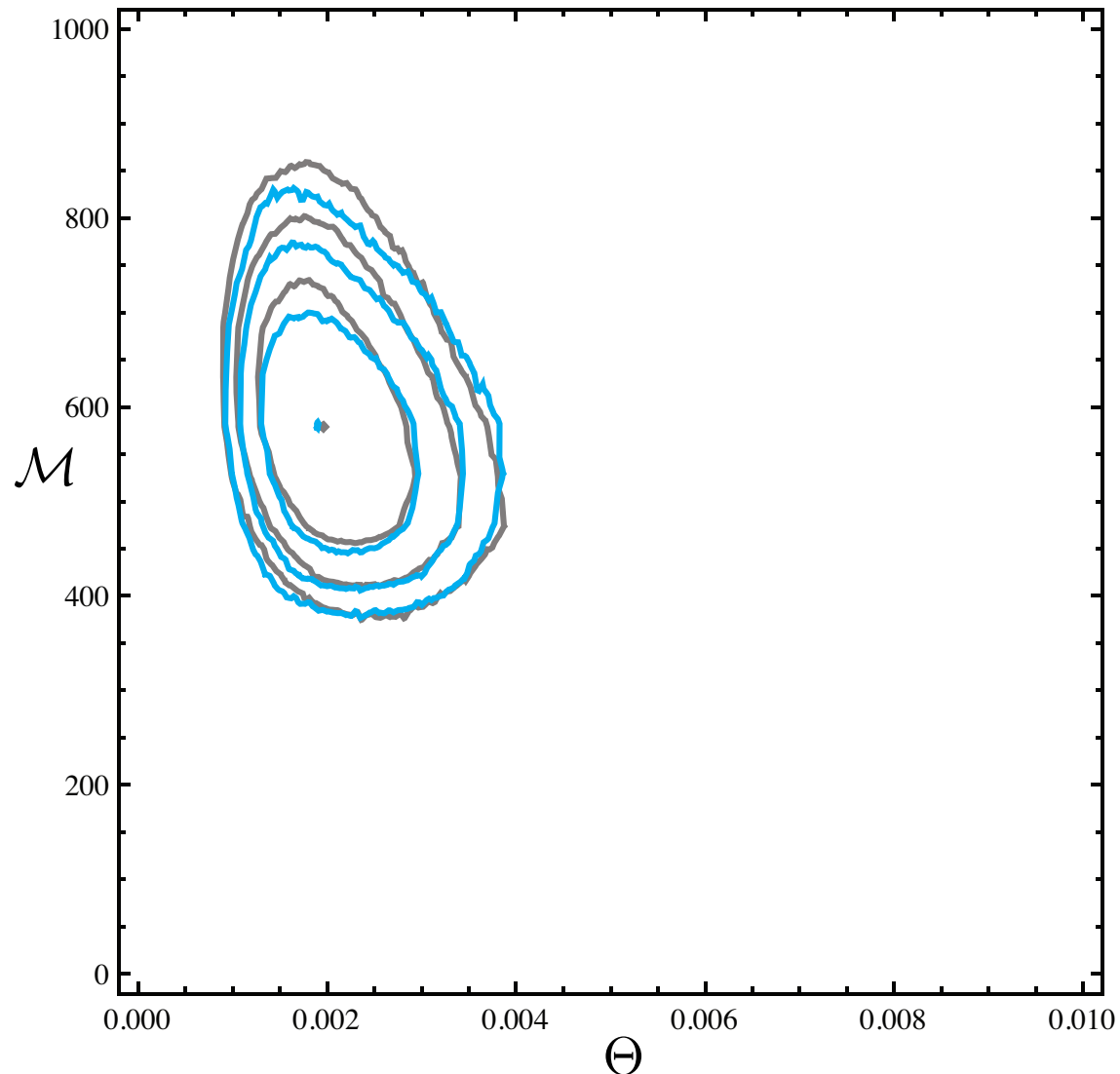
Posterior densities for a 2-parameter model (size and gene flow) for different runs (all with a total of  $50 \times 10^6$  steps).



# Run time versus accuracy

10 replicated runs\*

Posterior density for a 2-parameter model (population size and gene flow) 10 runs each for  $5 \times 10^6$  steps took about 2 hours.



One long run: 20 hours  
all samples used (burn-in)  
10 replicates: 2 hours  
first 50% of samples discarded

Contour lines are at 50%, 95%, and 99% credibility level

$\Theta = 4N_e\mu$  (population size scaled by mutation rate)

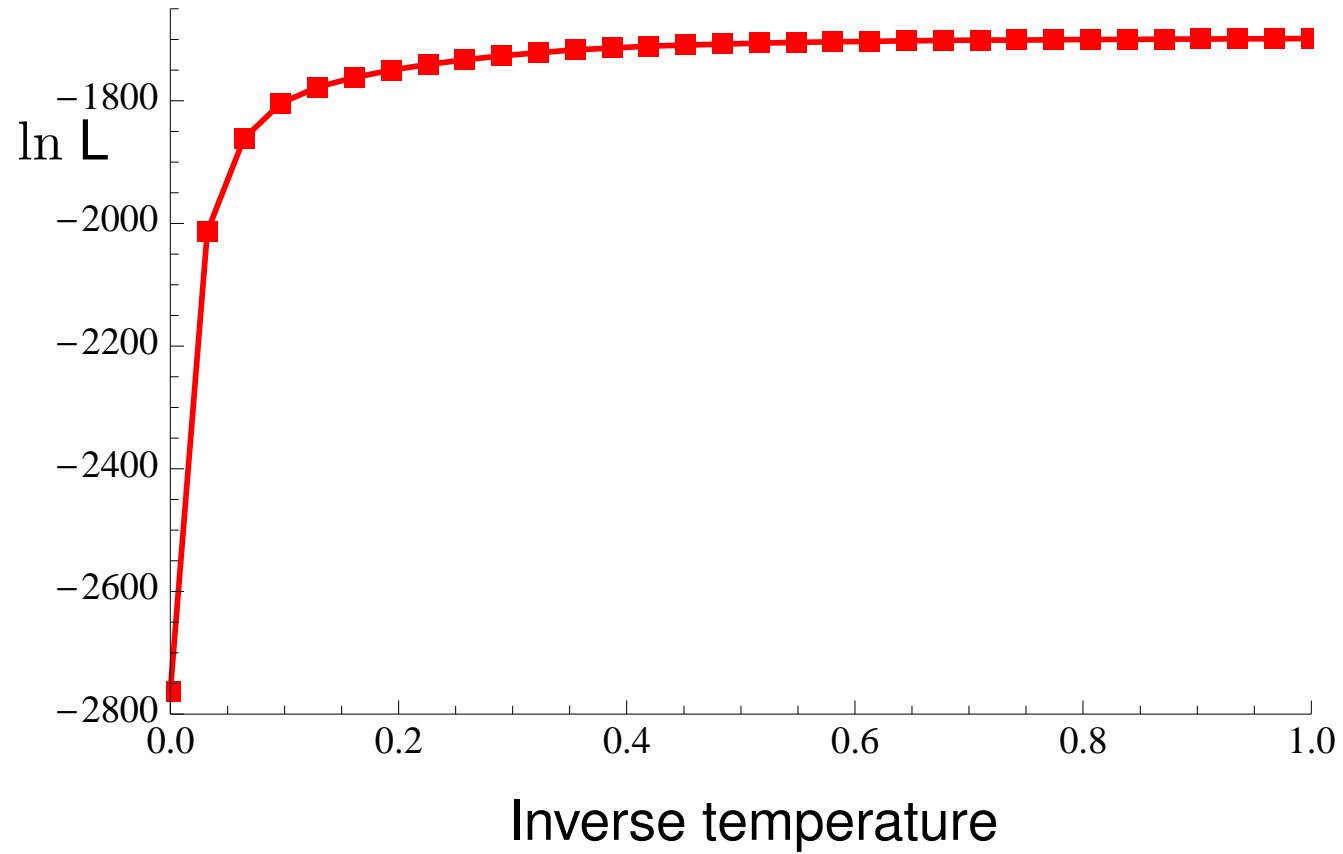
$\mathcal{M} = \frac{m}{\mu}$  (immigration rate scaled by mutation rate)

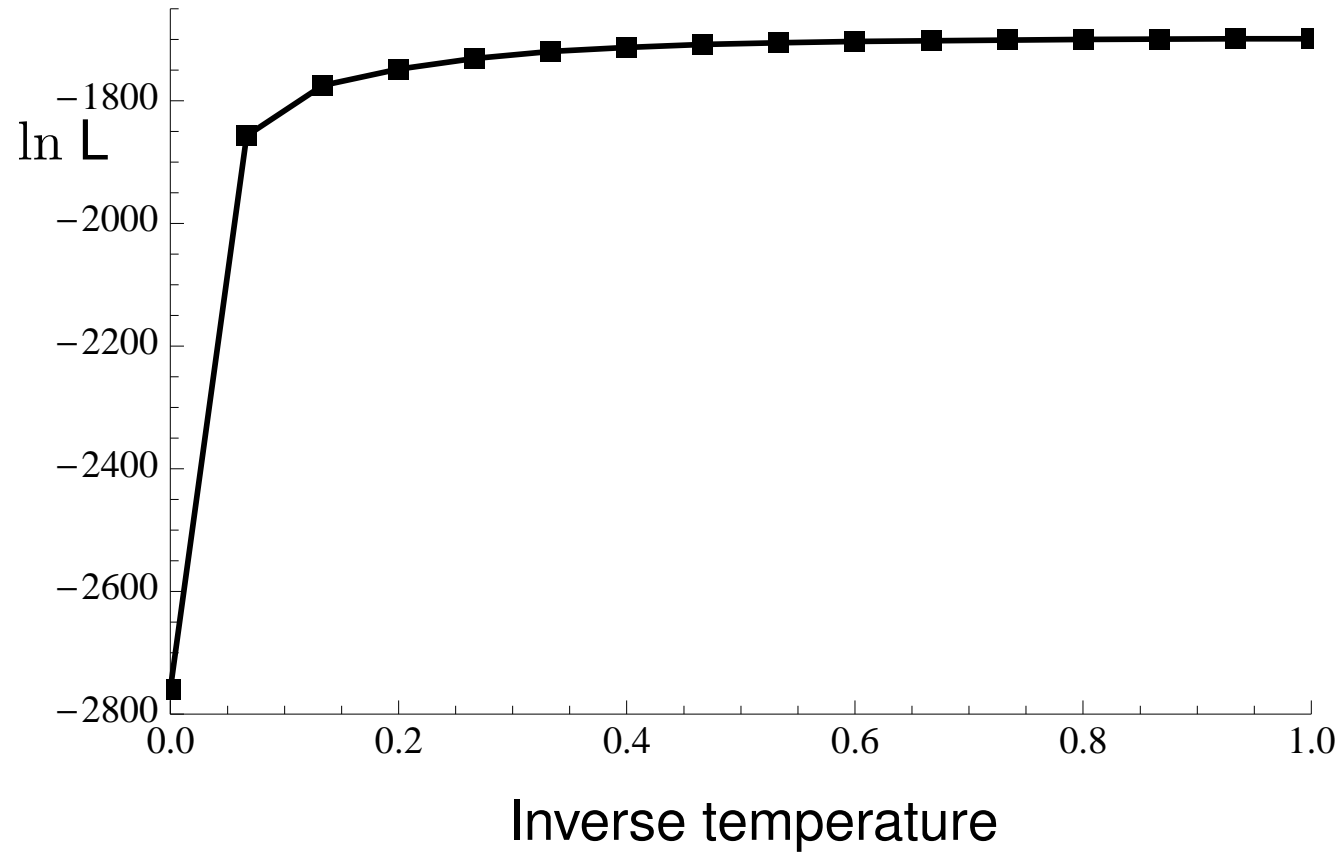




# Thermodynamic integration

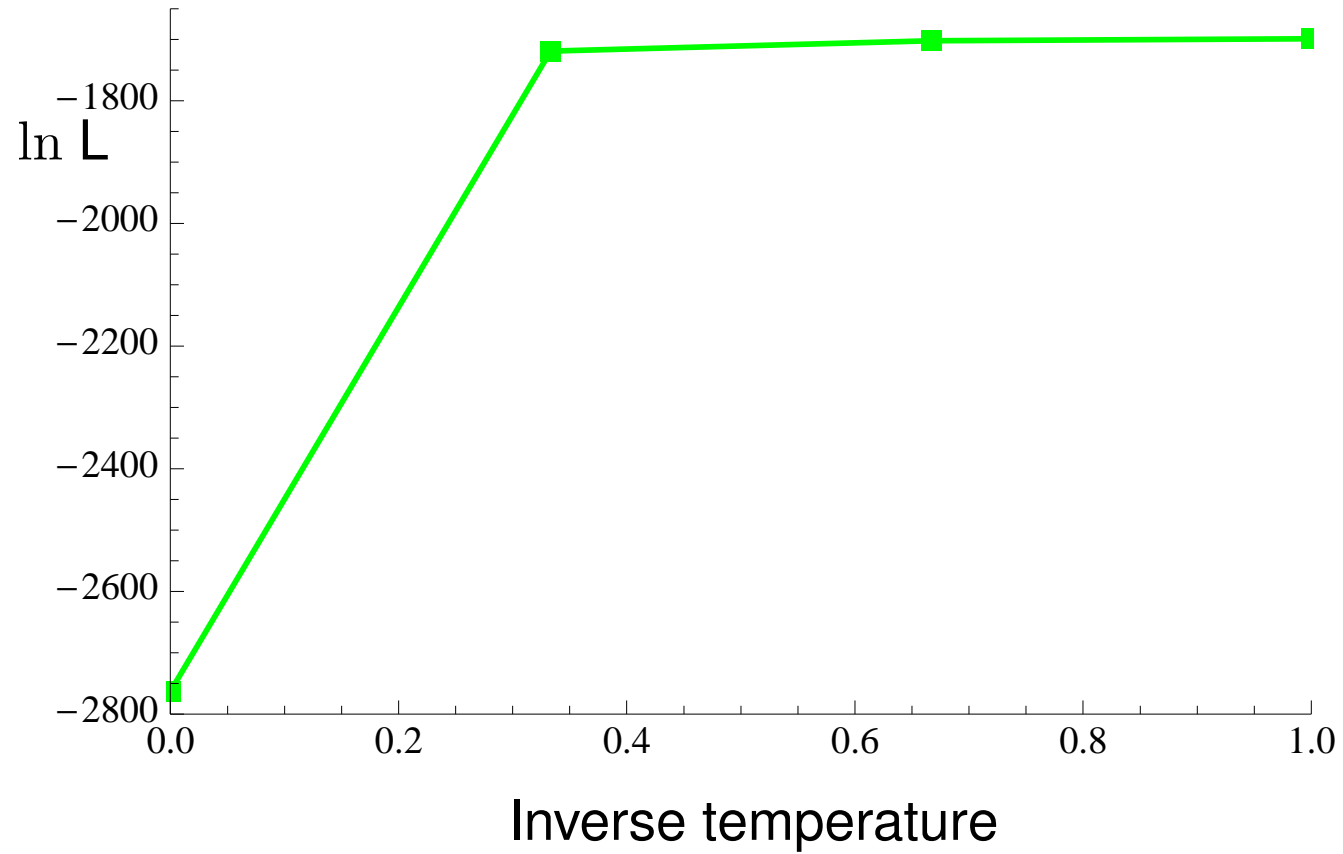
Marginal  
likelihood





# Thermodynamic integration

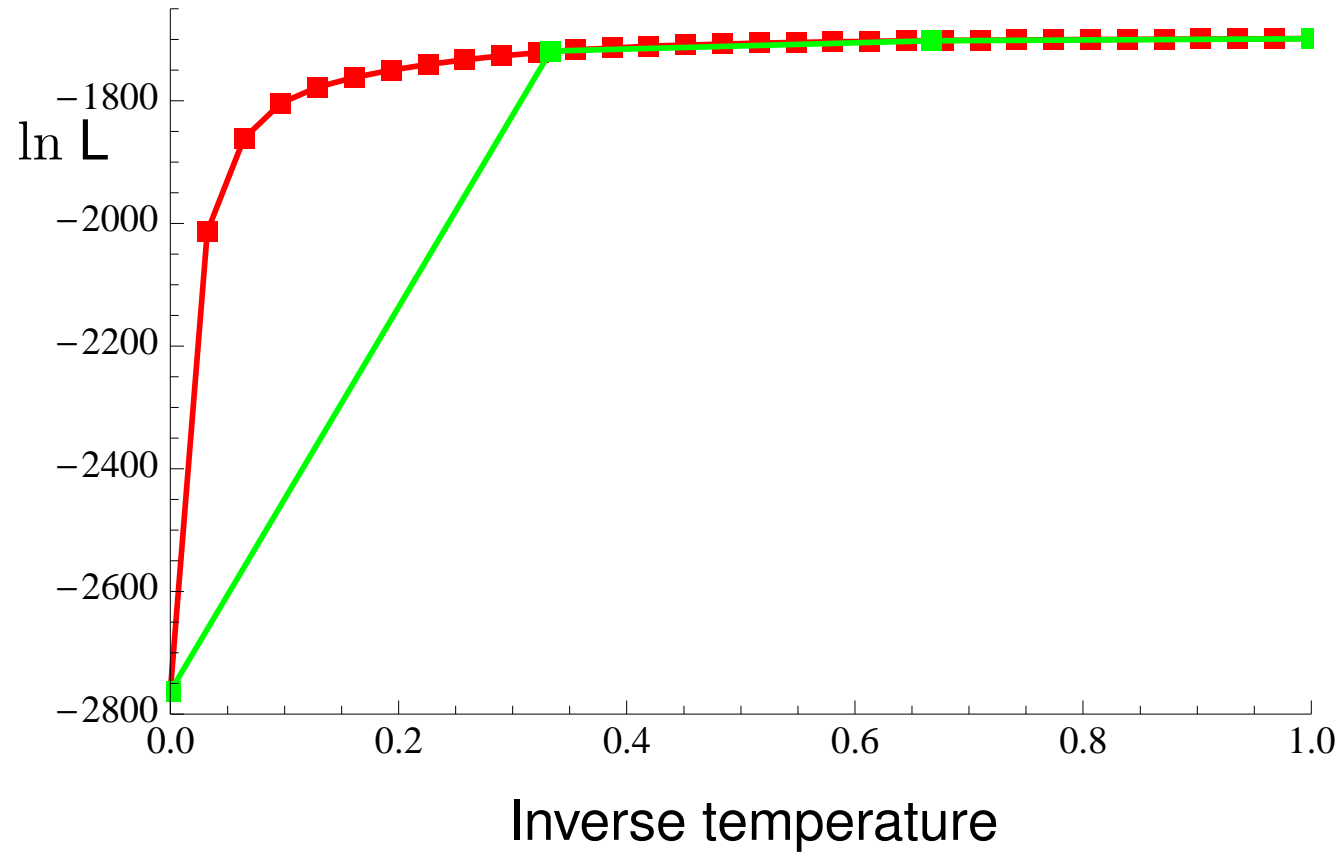
Marginal  
likelihood





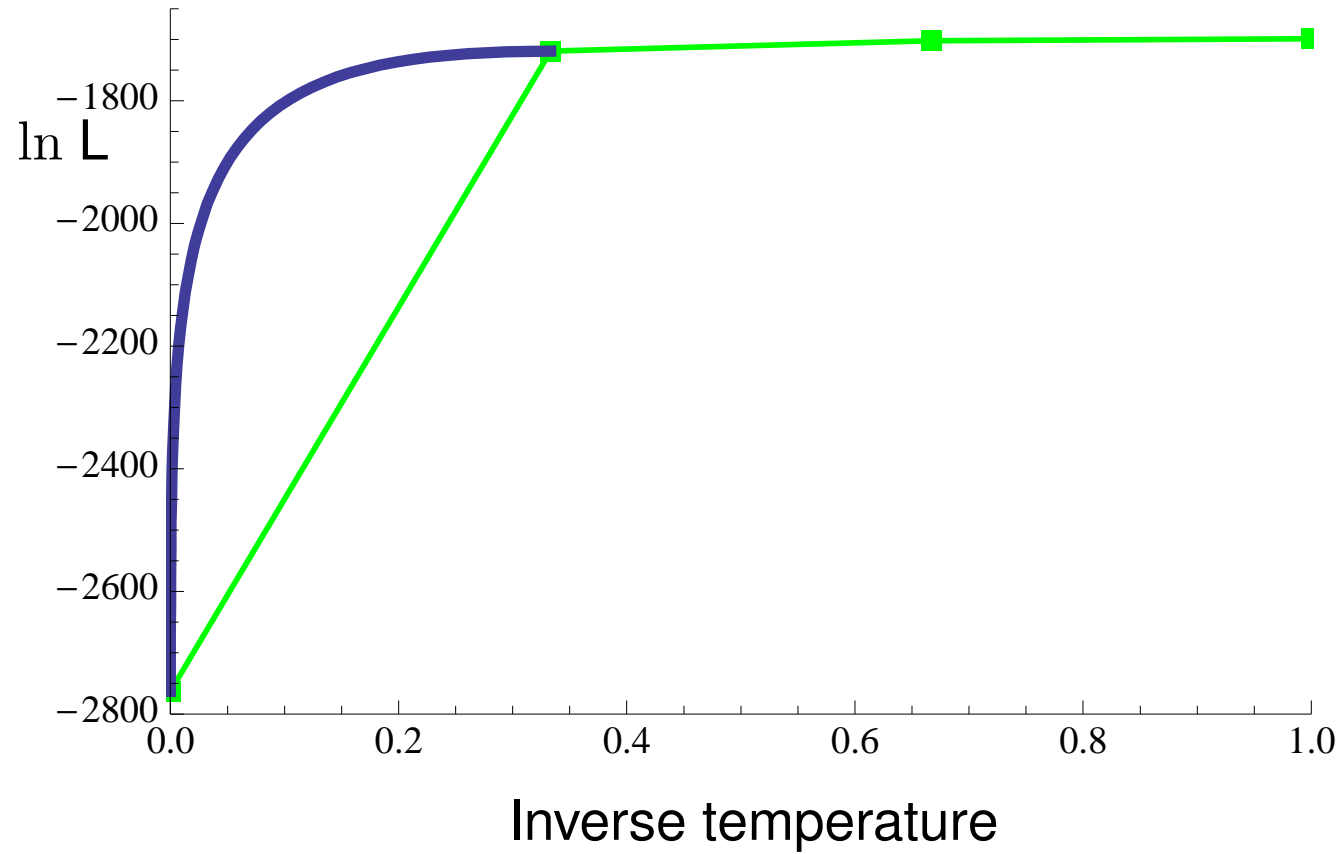
# Thermodynamic integration

Marginal  
likelihood



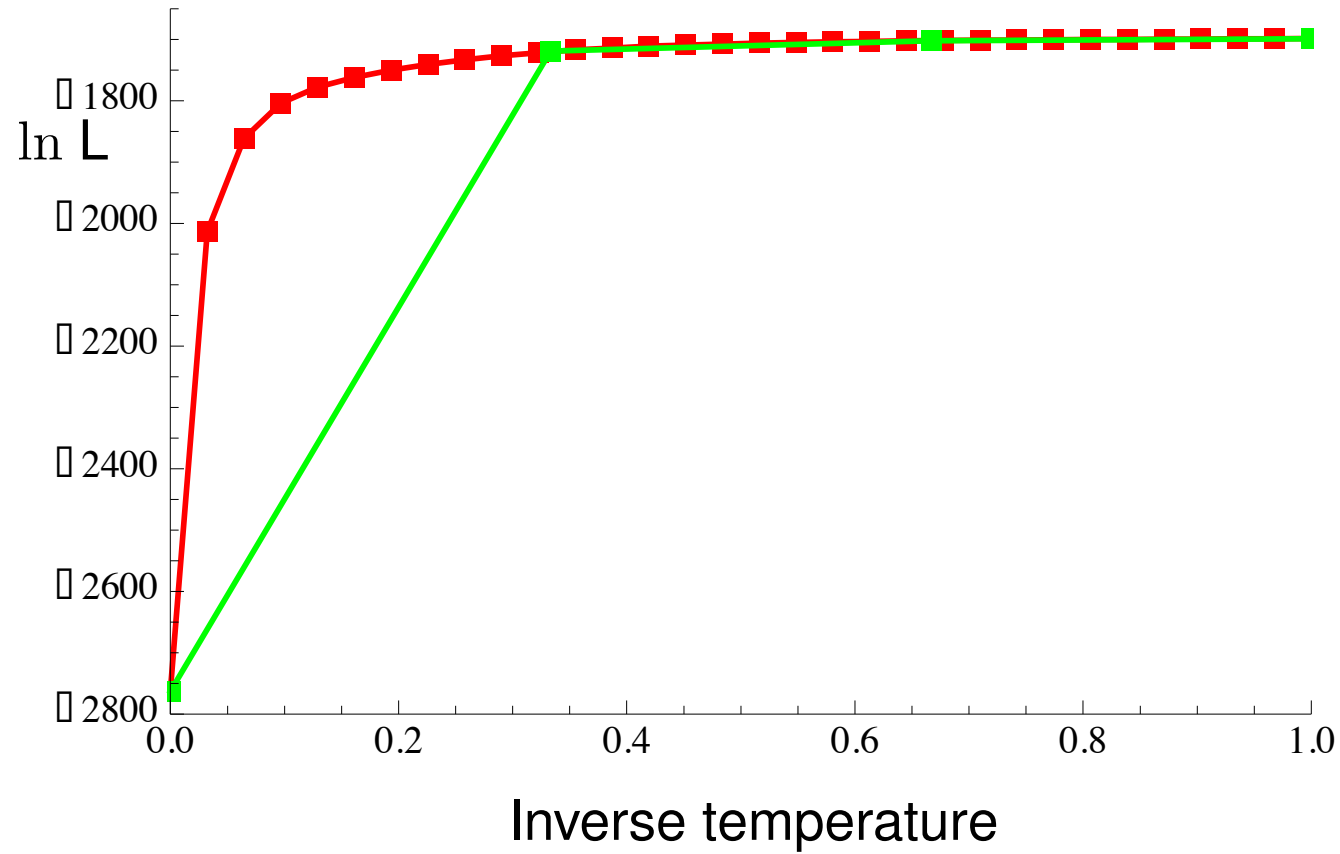
# Thermodynamic integration

Marginal  
likelihood



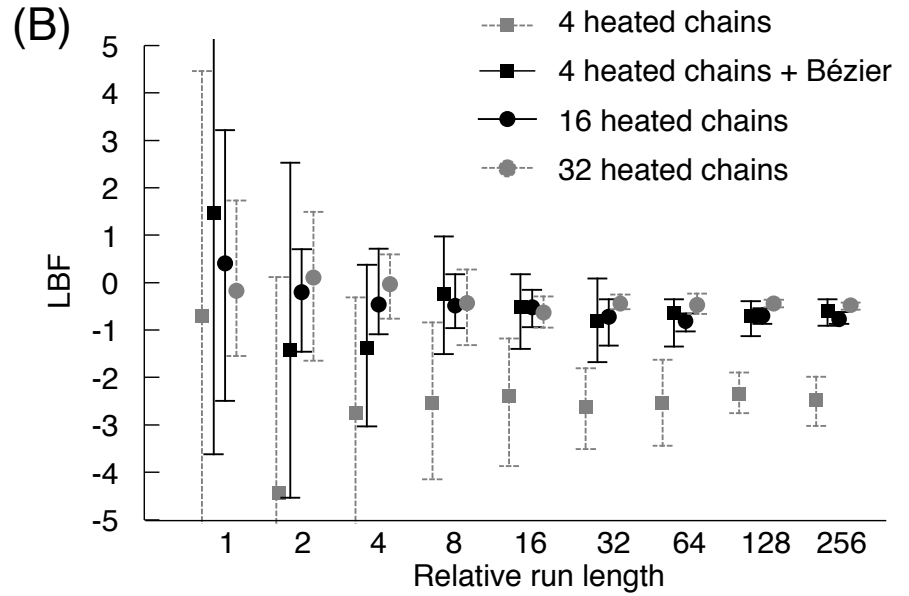
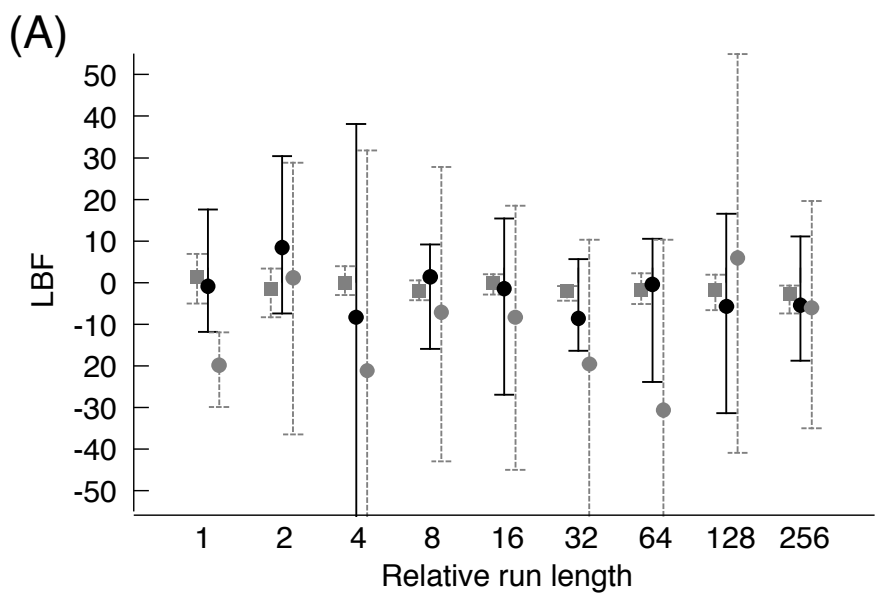
# Thermodynamic integration

Marginal  
likelihood





# Harmonic versus Thermodynamic



# Model comparison

With a criterium such as likelihood we can compare nested models. Commonly we use a likelihood ratio test (LRT) or Akaike's information criterion (AIC) to establish whether phylogenetic trees are statistically different or mutation models have an effect on the outcome, etc.

Kass and Raftery (1995) popularized the **Bayes Factor** as a Bayesian alternative to the LRT.

# Approximative Likelihood ratio test

=====  
Likelihood ratio tests  
=====

Over all loci  
Legend for the LRT tables

Null-Hypothesis: your test model		Log(likelihood) of test model
=same=		Log(likelihood) of full model
full model (the model under which the		Likelihood ratio test value
genealogies were sampled)		Degrees of freedom of test
[Theta values are on the diagonal of the		Probability*
Migration matrix, migration rates are		Probability**
specified as M]		Akaike's Information Criterion***
		Number of parameters used

\*) Probability under the assumption that parameters have range -Inf to Inf  
 \*\*) Probability under the assumption that parameters have range 0 to Inf  
 \*\*\*) AIC: the smaller the value the better the model  
       [the full model has AIC=-10.504683, num(param)=4]

H0: 0.5471 61.081 0.0000 0.0417		LnL(test) = -93.070073
= 0.5471 61.081 23.517 0.0417		LnL(full) = 9.252341
[ *, *, 0, *,]		LRT = 204.644829
		df = 1
		Prob = 0.000000
		Probc = 0.000001
		AIC = 192.140146
		num(param)= 3

H0: 0.5471 0.0000 23.517 0.0417		LnL(test) = -930.914874
= 0.5471 61.081 23.517 0.0417		LnL(full) = 9.252341
[ *, 0, *, *,]		LRT = 1880.334431
		df = 1
		Prob = 0.000000
		Probc = 0.000001
		AIC = 1867.829748

# Bayes factor

In a Bayesian context we could look at the **posterior odds ratio** or equivalently the **Bayes factors**.

$$p(M_1|X) = \frac{p(M_1)p(X|M_1)}{p(X)}$$

$$\frac{p(M_1|X)}{p(M_2|X)} = \frac{p(M_1)}{p(M_2)} \times \frac{p(X|M_1)}{p(X|M_2)}$$

$$\text{BF} = \frac{p(X|M_1)}{p(X|M_2)} \quad \text{LBF} = 2 \ln \text{BF} = 2 \ln \left( \frac{p(X|M_1)}{p(X|M_2)} \right)$$

The magnitude of BF gives us evidence against hypothesis  $M_2$

$$\text{LBF} = 2 \ln \text{BF} = z \quad \left\{ \begin{array}{ll} 0 < |z| < 2 & \text{No real difference} \\ 2 < |z| < 6 & \text{Positive} \\ 6 < |z| < 10 & \text{Strong} \\ |z| > 10 & \text{Very strong} \end{array} \right.$$



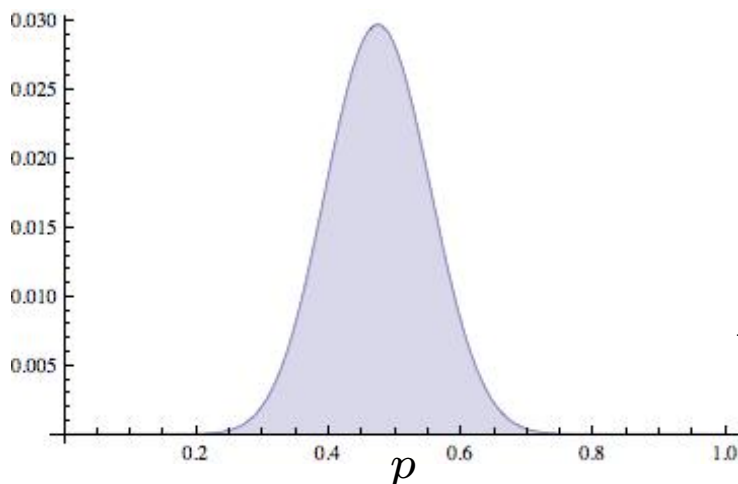
# Bayes factor versus Likelihood ratio test

Likelihood ratio test compares the maximum likelihoods of two models (hypotheses), whereas the BF compares the marginal likelihoods (=integrated over all parameter values) of two models.

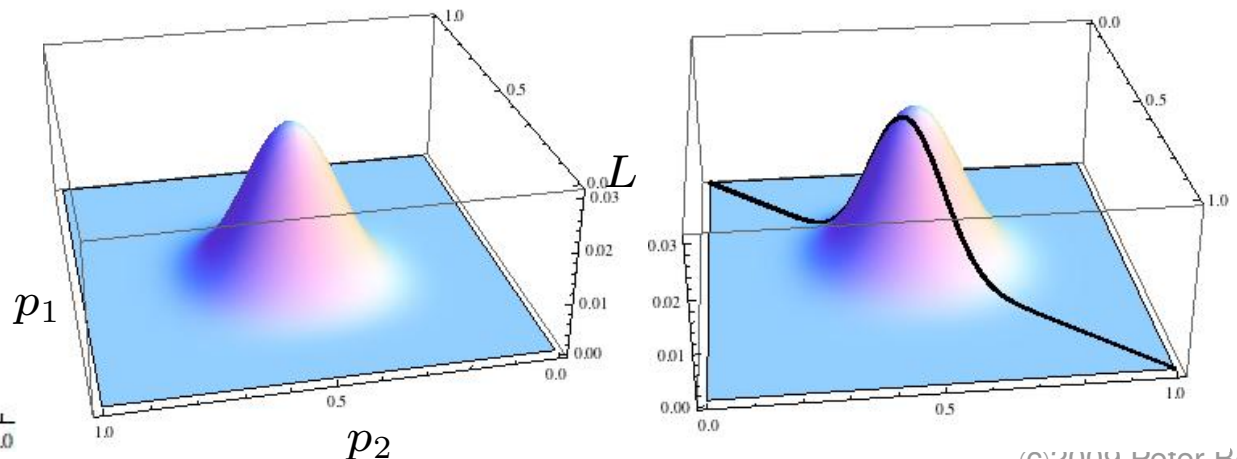
Example: Coin tossing experiment with two identically looking coins thrown simultaneously. (Data: coin 1: 9 heads out of 20, coin2 10 heads out of 20)

Model 1 assumes that both coins are similar (both loaded or not):  $p = p_1 = p_2$   
 Model 2 assumes that the coins are independent of each other:  $p_1 \neq p_2$

Model 1



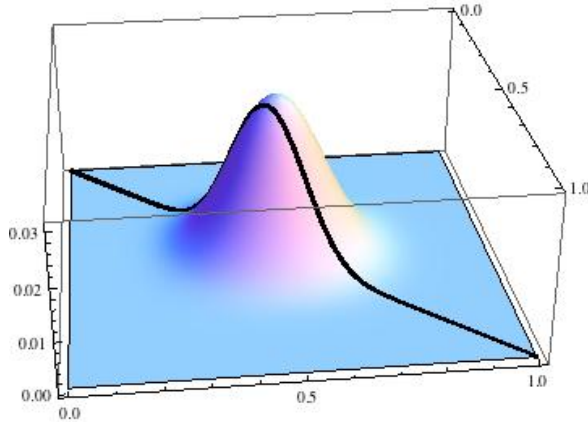
Model 2



# Bayes factor versus Likelihood ratio test

BF integrates over all values for each model

$$\text{LBF} = 2 \ln \left( \frac{L(M_1)}{L(M_2)} \right) = 2 \ln \left( \frac{0.0058}{0.0023} \right) = 1.87$$



In contrast LRT compares the maximum values.

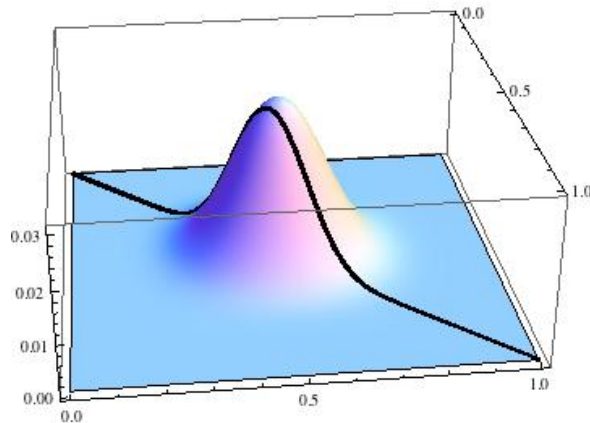
$$\text{LRT} = -2 \ln \left( \frac{0.0300}{0.0312} \right) = 0.10$$

# Bayes factor versus Likelihood ratio test

*Data set 1: 9 and 10 heads*

$$\text{LBF} = 2 \ln \left( \frac{0.0058}{0.0023} \right) = 1.87$$

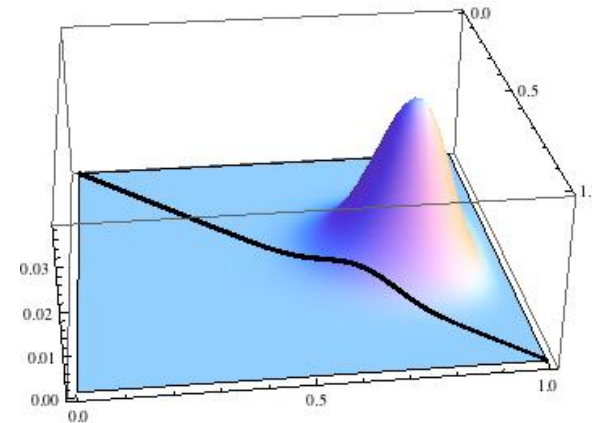
$$\text{Prob}(M_1) = 0.87; \text{Prob}(M_2) = 0.13$$



*Data set 2: 16 and 10 heads*

$$\text{LBF} = 2 \ln \left( \frac{0.0009}{0.0023} \right) = -1.76$$

$$\text{Prob}(M_1) = 0.15; \text{Prob}(M_2) = 0.85$$



$$\text{LRT} = -2 \ln \left( \frac{0.0300}{0.0312} \right) = 0.10$$

$$\text{AIC}(M_1) = -2 \ln(0.0300) + 2 \times 1 = 9.01$$

$$\text{AIC}(M_2) = -2 \ln(0.0312) + 2 \times 2 = 10.93$$

$$\text{LRT} = -2 \ln \left( \frac{0.0051}{0.0384} \right) = 4.053$$

$$\text{AIC}(M_1) = -2 \ln(0.0051) + 2 = 12.6$$

$$\text{AIC}(M_2) = -2 \ln(0.0384) + 4 = 10.5$$