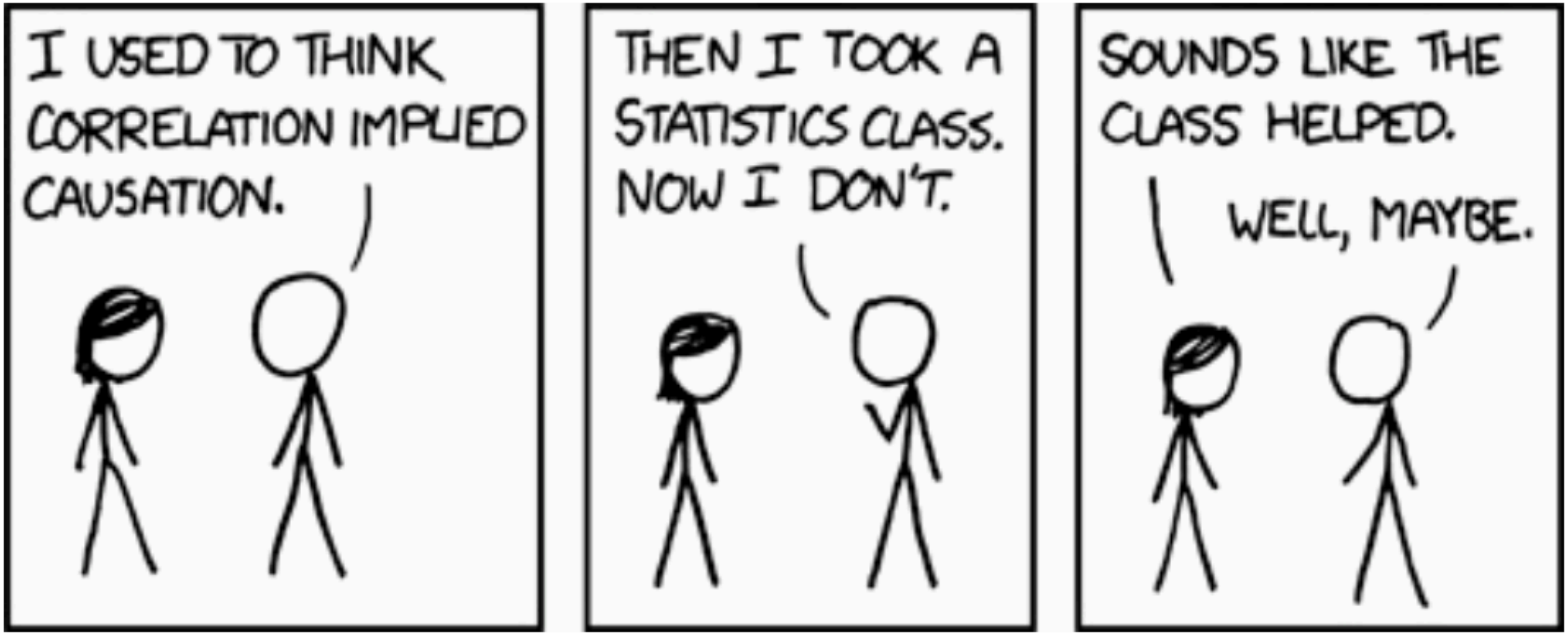

I USED TO THINK CORRELATION IMPLIED CAUSATION.
THEN I TOOK A STATISTICS CLASS. NOW I DON'T.
SOUNDS LIKE THE CLASS HELPED.
WELL, MAYBE.

Australian National University

Inference from Data. Bootstrapping and Monte-Carlo techniques

Michael Ireland (RSAA)

Bayesian Basics…

- The intuitive *frequentist* definition of probability is that P(A) is a number between 0 and 1 representing the limit for an infinite number of identical experiments of the fraction of experiments that achieve result A.
- An infinite number of experiments is not very practical.
- The *Bayesian* approach is to be more nuanced – probability represents the likelihood of a *statement* being true, even if the idea of an approximately infinite number of experiments seems irrelevant (e.g. Cosmology, where there is only one Universe).
- If we want to get pedantic, by a *statement* we really mean that an *elementary event* is part of a *set.*
- **Example**: $\Lambda>0$. Out of the set of all possible universes our Universe (the event) is part of the set with $\Lambda>0$.

Assumptions and Basic Stats

- Strict Bayesians do not believe in “Unconditional probability”.
- The probability of an event always depends on conditions. E.g. a dice roll has well-defined probability if the dice is fair (not weighted) and the throw is reasonable.
- We will write P(A) as a probability given assumptions defined elsewhere
- The probability of one statement given another is written:
  P(A|B)
- *Conditional* probability is very useful in writing and reading science papers, as it enables information beyond a paper’s scope to be used, and a reader to make their own conclusions.
- **Examples**:
  a) $P(A|A)=1$
  b) If $A \cup B=C$ and $AB=\emptyset$, then $P(A|C) + P(B|C)=1$ and $P(A|B)=0$.
  c) $P(AB) = P(A|B)P(B)$

Bayes' Theorem

- The key to Bayesian probability is Bayes' theorem, which can be written:

$$P(A|D) = \frac{P(A)P(D|A)}{P(D)} \text{ or}$$

$$P(A_k|D) = \frac{P(A_k)P(D|A_k)}{\Sigma_k P(A_k)P(D|A_k)} \text{ for mutually exclusive } A_k$$

- Derived in any good textbook, D can be any event, but is written as D because it is typically a particular set of data.
- P(A) is the *prior* and P(A|D) is the *posterior.*
- With many data sets D_j, Bayes' theorem can be repeated, with one posterior becoming the next prior.

Australian National University

Bayes' Theorem with Probability Densities

- In astrophysics, many parameterizations are *continuous*, meaning that our probabilities are really n-dimensional probability densities, e.g.:

$$P(x < X < x + dx) = f(x)dx$$

$$f(x_0|D) = \frac{f_p(x_0)P(D|x_0)}{\int f_p(x)P(D|x)dx}$$

- Data are often (approximated by) Normal distributions, i.e.:

$$P(D|x) \propto \exp(-\chi^2/2)$$

$$\chi^2 = \Sigma_k \frac{(m_k(x) - d_k)^2}{\sigma_k^2}$$

Likelihood

- You'll often hear of *likelihood* instead of *probability*. The conventional definition for a continuous random variable θ is:

$$
\begin{aligned}
L(\theta|\{D_k\}) &= f(\{D_k\}|\theta) \\
&= \Pi_k f(D_k|\theta) \text{ for independent data} \\
&= \exp(-\chi^2/2) \text{ independent data, Normally distributed errors}
\end{aligned}
$$

- Note that *likelihood* isn't normalised.
- The *Bayesian* likelihood needs a prior (e.g. last example):

$$L(\theta|\{D_k\}) = f_{\mathrm{pr}}(\theta) f(\{D_k\}|\theta)$$

Uninformed Priors

- As the last example shows… ignoring priors can give the wrong answer. There are some typical examples (read up on *Jeffreys priors* if you want a formal derivation).
- Unbounded numbers that can be positive or negative – a *Uniform* distribution, i.e. no need to write anything down.
- Scale factors that have to be positive – a *Logarithmic* distribution with: *f(a) α 1/a*
- Angles on a sphere, e.g. inclination in [0,180], a sinusoidal distribution with: *f(i) = sin(i)/2.*

Inverse Problems with Uncertain Data

- Often a data set is reasonably removed from what we're trying to learn. E.g.
  1. Observed positions of a binary star on the sky can be determined from an orbital solution… but an orbital solution is non-trivially determined from the measurements of positions.
  2. Interferometric measurements can be determined from a true object brightness distribution, but (esp. with self-cal etc) there is not necessarily a unique image corresponding to interferometric data.
  3. A CMB power spectrum and SNIa laws can be determined from a cosmology, but there is no formula to invert this.

Marginalisation

- Most inverse problems are phrased as problems of computing likelihood.
- Sometimes, many of the parameters are nuisance parameters, and the term P(D|M) involves the probability product rule and marginalisation, i.e..

$$P(D|M) = \int P(D, \phi|M) d\phi,$$

$$P(D|M) = \int P(D|\phi, M) P(\phi|M) d\phi$$

Australian National University

Comparing Models

- If you have two models to compare, often the probability ratio is more intuitively useful than the probability.
- If only 2 models are being considered, then:

$$P(M_1|I) = \frac{P(D|M_1, I)P(M_1|I)}{P(D|M_1, I)P(M_1|I) + P(D|M_2, I)P(M_2|I)}$$

- More generally, we can consider a probability ratio (*odds ratio*) *R*, and a *Bayes' factor K*:

$$R = \frac{P(M_1|D)}{P(M_2|D)} = \frac{P(D|M_1)}{P(D|M_2)}\frac{P(M_1)}{P(M_2)} = \frac{P(M_1)}{P(M_2)}K$$

Australian National University

Famous Example: Tegmark (2004)

PHYSICAL REVIEW D **69**, 103501 (2004)

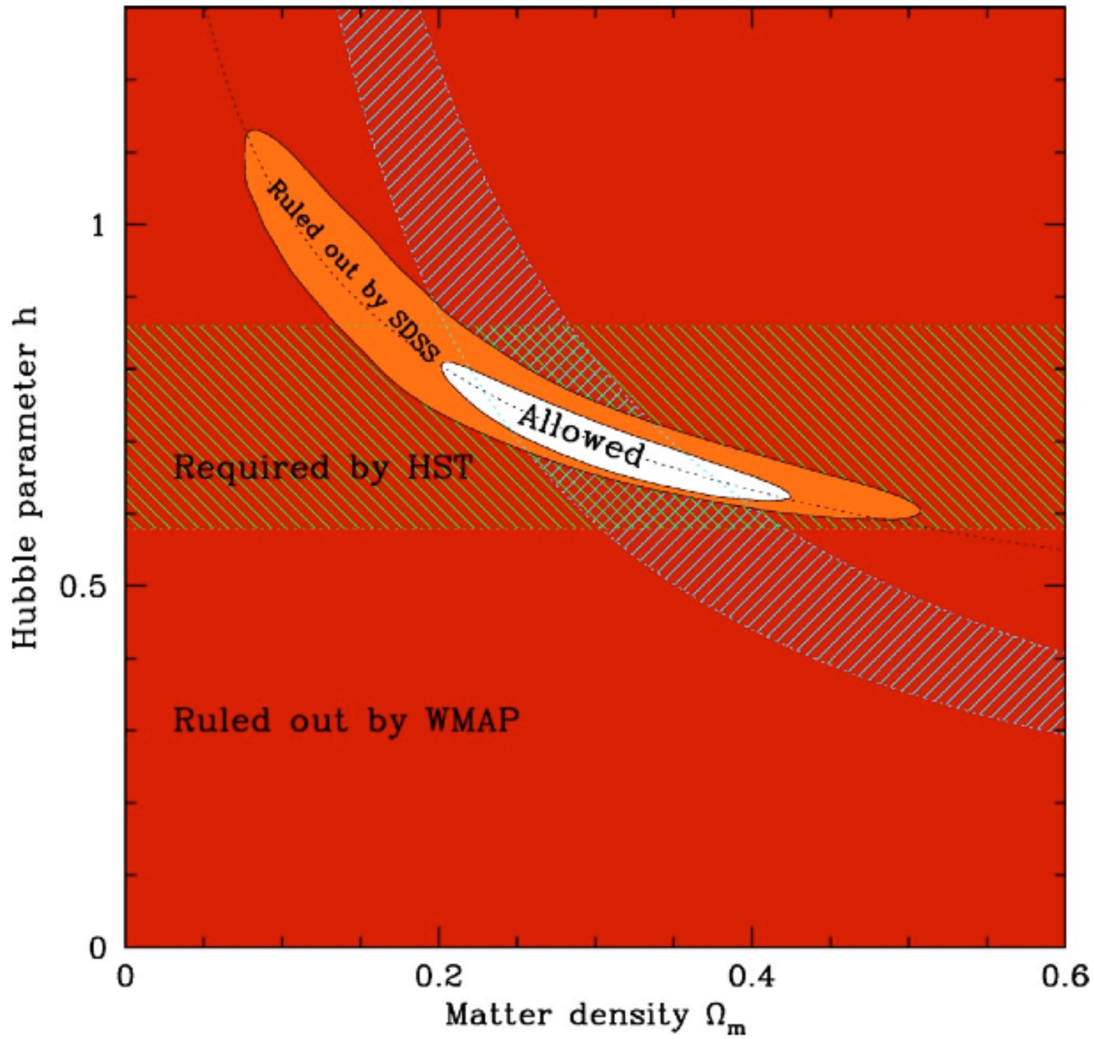
Cosmological parameters from SDSS and WMAP

Max Tegmark,[1,2] Michael A. Strauss,[3] Michael R. Blanton,[4] Kevork Abazajian,[5] Scott Dodelson,[6,7] Havard Sandvik,[1]

a cosmological constant without tilt ($n_s = 1$), running tilt, tensor modes, or massive neutrinos. Adding SDSS information more than halves the WMAP-only error bars on some parameters, tightening 1σ constraints on the Hubble parameter from $h \approx 0.74^{+0.18}_{-0.07}$ to $h \approx 0.70^{+0.04}_{-0.03}$, on the matter density from $\Omega_m \approx 0.25 \pm 0.10$ to $\Omega_m \approx 0.30 \pm 0.04$ (1σ) and on neutrino masses from < 11 to < 0.6 eV (95%). SDSS helps even more when

- One of many early-2000s papers on Bayesian cosmological parameters, taking many data sets together and marginalising over unknown data.

Famous Example: Tegmark (2004)



… and marginalisation…

$$P(D|M) = \int P(D, \phi|M) d\phi,$$

$$P(D|M) = \int P(D|\phi, M) P(\phi|M) d\phi$$

(marginalisation can also be for continuous random variables)

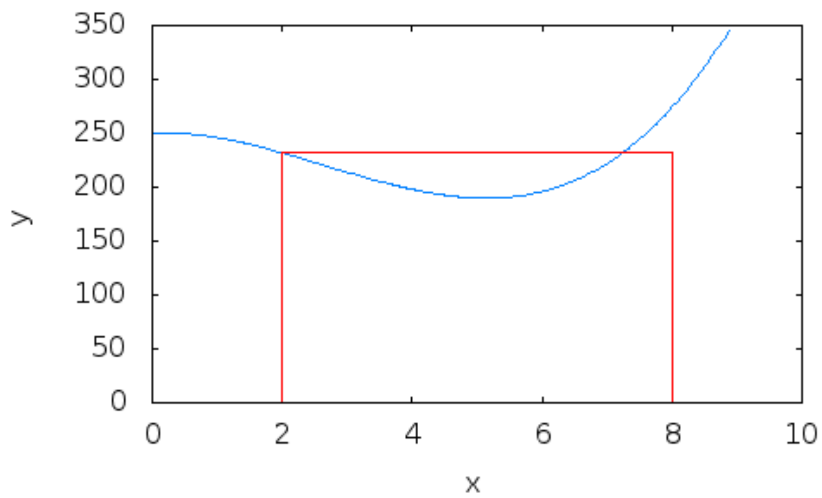
- The catch is that integrals are often highly multidimensional. How can we compute them efficiently?

(NB "parameters" above sometimes *Φ, sometimes θ)*

Monte-Carlo Integration

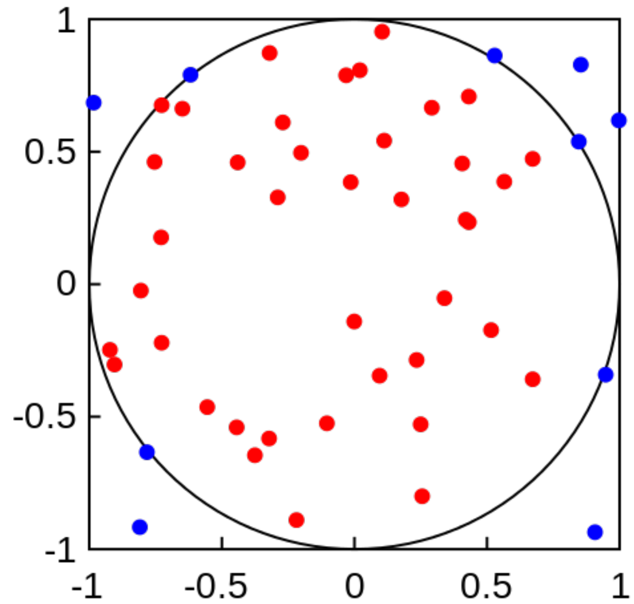
- If we want to integrate a function *f* of a real variable over an interval, we can approximate the integral by a sum:

$$\int_a^b f(x)dx \approx \frac{1}{M}\sum_{i=1}^{M} f(x_i) \text{ for } \{x_i\} \in [a,b]$$



- If we choose the x values regularly, this is *rectangle* integration (similar to the trapezoidal rule).
- If we choose the x values randomly (Uniformly distributed), this is *Monte-Carlo* integration.

Monte-Carlo Integration



- Monte-Carlo integration is obviously useless in 1D.
- In N computations in M dimensions, the error in a trapezoidal-rule like integration is proportional to $N^{-2/M}$.
- Monte-Carlo uncertainties just go as $N^{-1/2}$.
- This means that in more than 4 dimensions, Monte-Carlo is a good idea.
- E.g. Volume of a 10-dimensional hypersphere of radius 1. Should be $\pi^5/120$. *Python exercise 2…*

(point out the problem… most the points lie outside the hypersphere)

Monte-Carlo Integration with Non-Uniform distributions

- Integrals that are a product of a complex function and a probability distribution can be computed like:

$$\int_{-\infty}^{\infty} f(x)g(x)dx \approx \frac{1}{M}\sum_{i=1}^{M} f(x_i) \text{ for } \{x_i\} \text{ distrubuted as } x_i \leftarrow G \sim g(x)$$

- This may seem easy if e.g. g is a Gaussian, but how far can we take the idea of complex distrubutions for our $\{x_i\}$?

Australian National University

Monte-Carlo Markov Chains

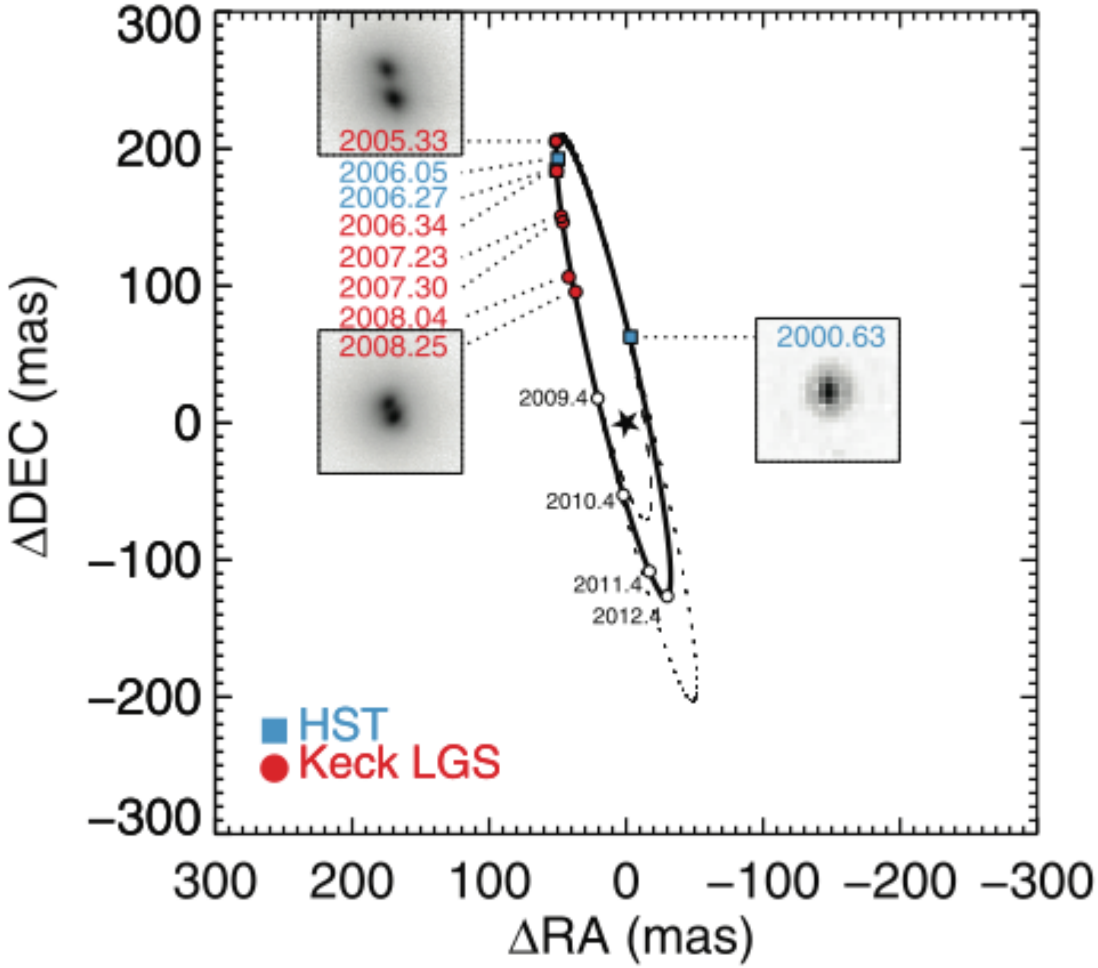
- A *Markov Process* is something where the future depends on the present but not the past [P(future | present) = P(future | present,past)]
- A *Markov Chain* is a discrete Markov process where the next step in the sequence (of numbers or vectors) depends only on the present step.
- *Markov Chain Monte Carlo* is a way of creating a Markov Chain where, in the limit of infinite time, the distribution of parameter vectors θ match the posterior likelihood.

$$L(\theta|\{D_k\}) = f_{\rm pr}(\theta) f(\{D_k\}|\theta)$$

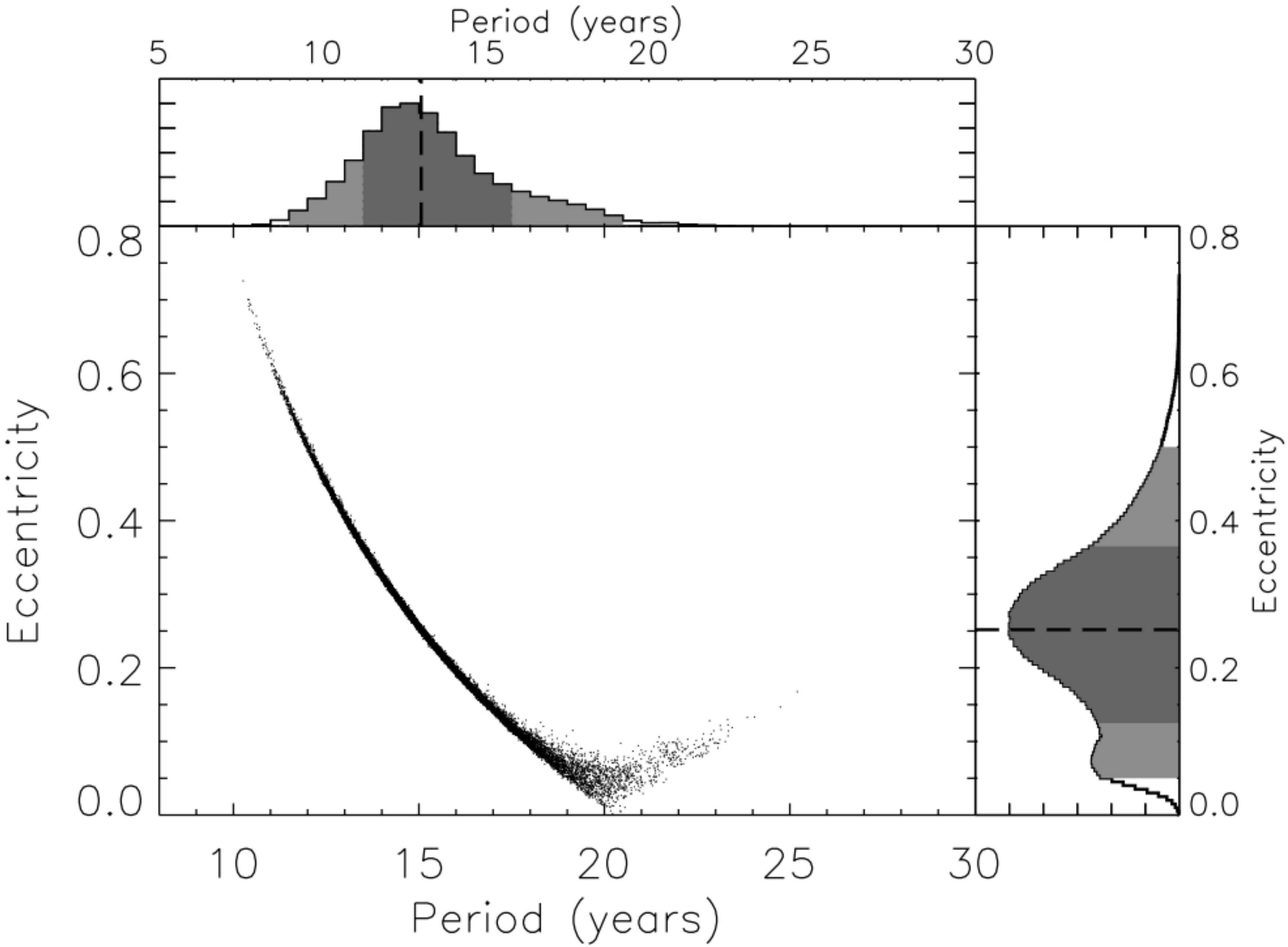
Personal Example: 2MASS J1534-2952AB

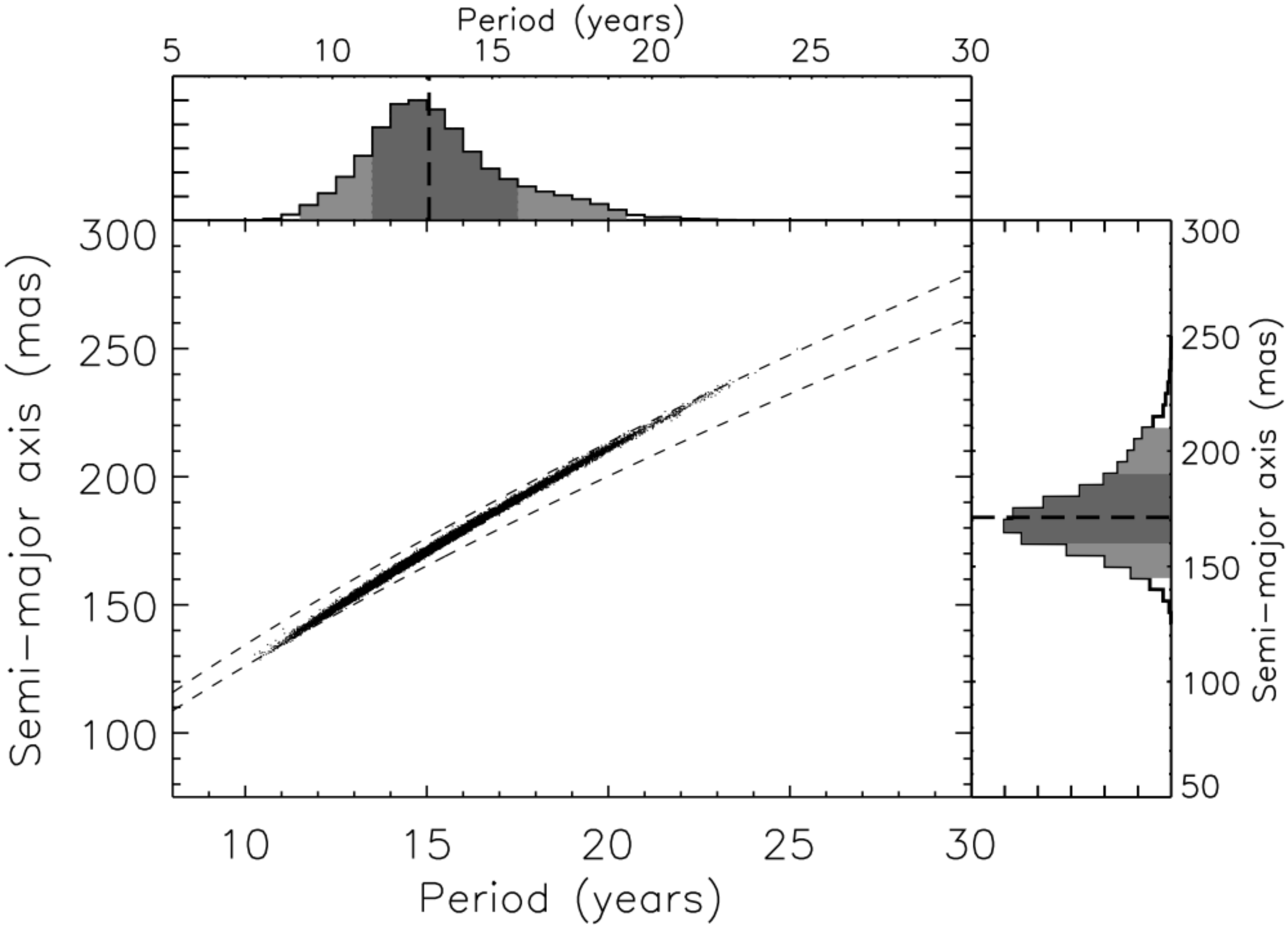
- From less than half an orbit, can we find the dynamical mass a T dwarf binary?

(NB paper submitted and accepted without 2008 data)

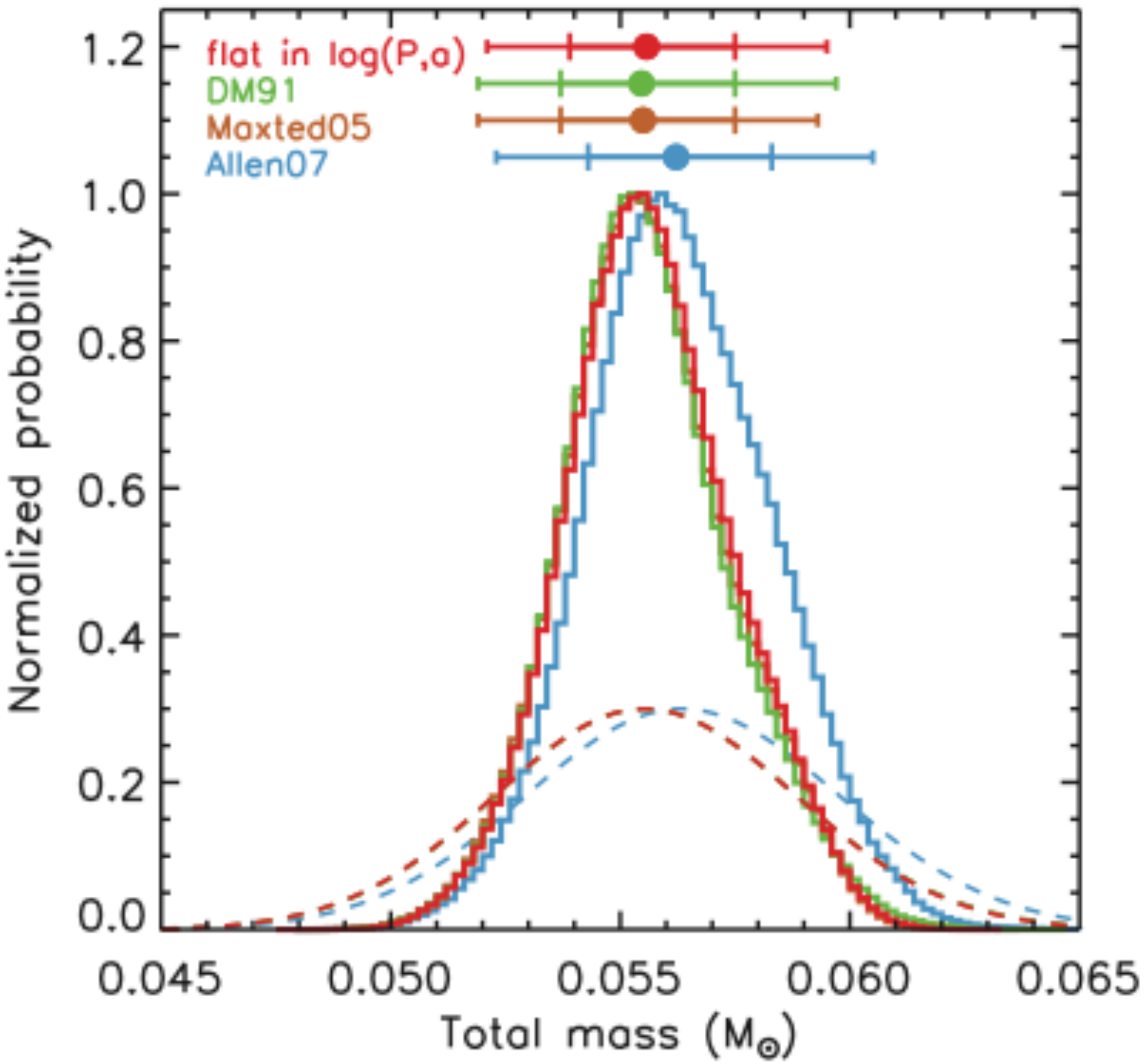


http://adsabs.harvard.edu/abs/2008ApJ...689..436L


Period (years)
5
10
15
20
25
30
0.8
0.6
0.4
0.2
0.0
Eccentricity
10
15
20
25
30
Period (years)
0.8
0.6
0.4
0.2
0.0
Eccentricity



Period (years)
5
10
15
20
25
30
300
250
200
150
100
Semi−major axis (mas)
10
15
20
25
30
Period (years)
300
250
200
150
100
50
Semi−major axis (mas)



- Dynamical mass $M=a^3/P^2$ changes little with different orbits.
- $M = f(\theta)$
- Parallax dominated the uncertainties.
- Different "reasonable" priors gave nearly the same answer.

Metropolis-Hastings with Gibbs Sampler…

- The simplest way to compute a chain is with the MH algorithm.
- The simplest MH variant is the *Gibbs Sampler*, where each dimension *k* has its own step size s_k and our parameters θ are approximated by the chain ***X***(t).
- Note that this algorithm always goes "downhill" and sometimes "uphill" in chi-squared space.

1. Randomly choose a dimension $k \in \{1, ..., N\}$ and direction $D \in \{-1, 1\}$.
2. Create a new trial element $\mathbf{Y} = \{X_1(t), ..., X_k(t) + D \times s_k, ...X_N(t)\}$.
3. Compute $q \leftarrow \frac{L(\mathbf{Y}|D)}{L(\mathbf{X}(t)|D)}$
4. Get a random number $r \leftarrow R \sim [0, 1]$
5. **if** $r \leq q$ **then**
6. $\quad \mathbf{X}(t+1) \leftarrow \mathbf{Y}$
7. **else**
8. $\quad \mathbf{X}(t+1) \leftarrow \mathbf{X}(t)$
9. **end if**

Australian National University

Metropolis-Hastings Algorithm

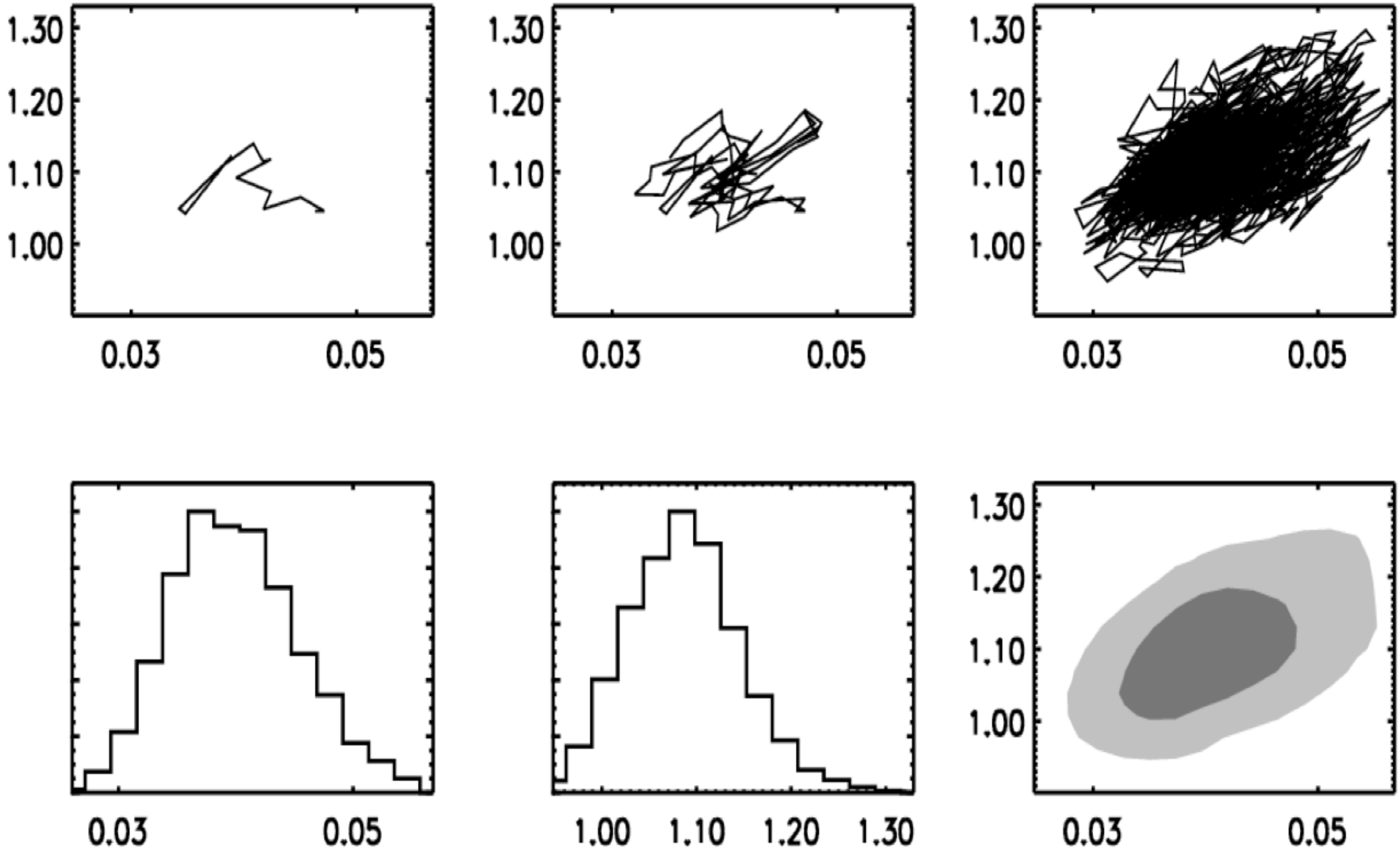
- The general MH algorithm can make e.g. variable step sizes.
- E.g. from http://arxiv.org/pdf/1202.3665v4.pdf, with:

$$X \sim \theta$$

$$p(X) \sim L(\theta|\{D_k\}) = f_{\mathrm{pr}}(\theta) f(\{D_k\}|\theta)$$

Algorithm 1 The procedure for a single Metropolis-Hastings MCMC step.

Draw a proposal $Y \sim Q(Y; X(t))$
$q \leftarrow [p(Y)\,Q(X(t); Y)]/[p(X(t))\,Q(Y; X(t))]$ *// This line is generally expensive*
$r \leftarrow R \sim [0, 1]$
if $r \leq q$ **then**
    $X(t+1) \leftarrow Y$
else
    $X(t+1) \leftarrow X(t)$
end if

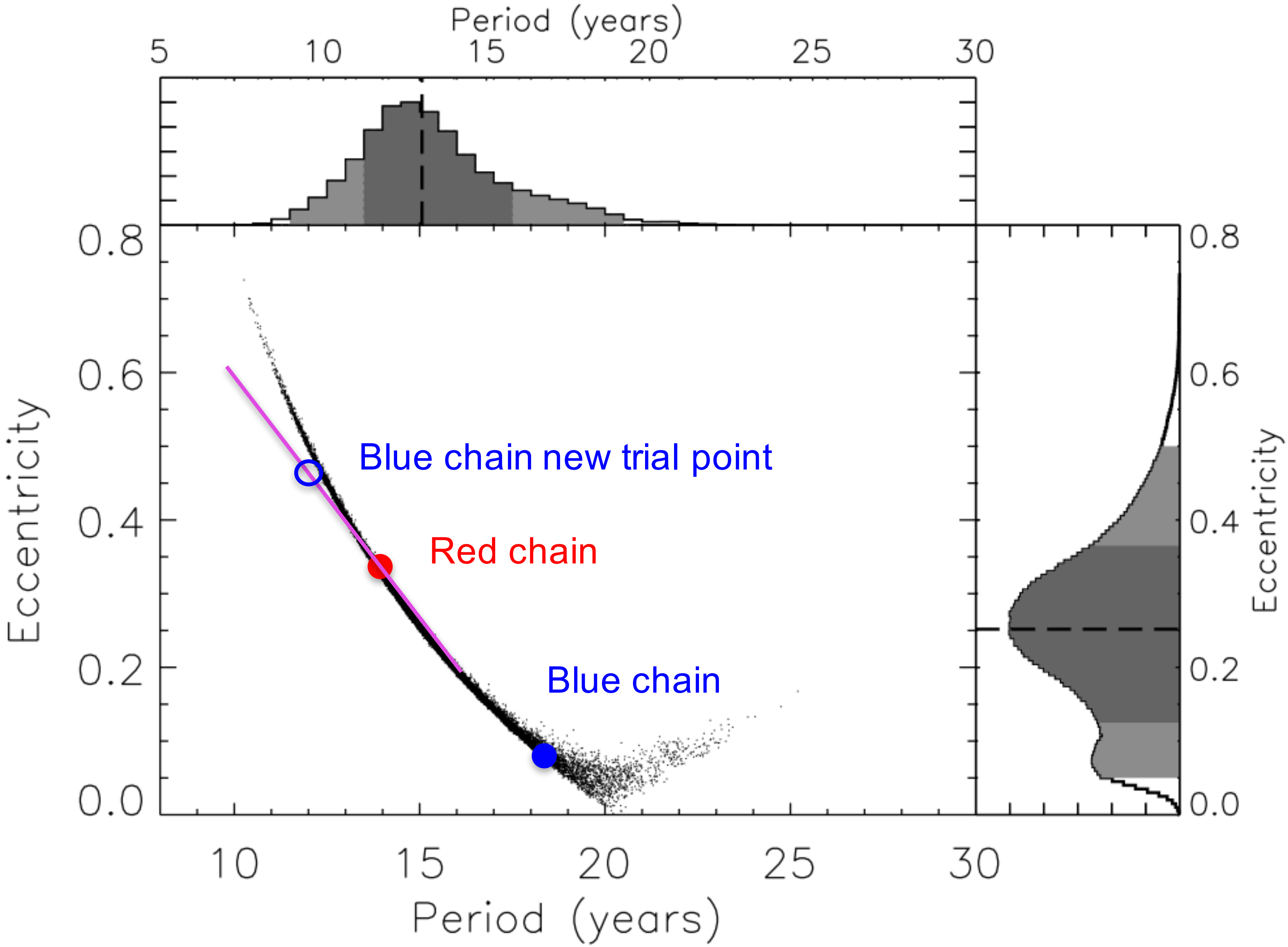

1.30
1.20
1.10
1.00
0.03
0.05
1.30
1.20
1.10
1.00
0.03
0.05
1.30
1.20
1.10
1.00
0.03
0.05
0.03
0.05
1.00
1.10
1.20
1.30
1.30
1.20
1.10
1.00
0.03
0.05

Tricks with Metropolis-Hastings

- Unless you know you start at the global minimum (and arguably even if you do), MCMC requires a “burn-in” time to randomize the starting location.
- Step sizes in the Gibbs sampler can’t be too large or too small – for optimal convergence, you want to accept new steps about half the time.
- Finding *credible intervals* which are *Bayesian confidence intervals* requires care in wording. E.g. no standard on using the posterior mean, median or mode (maximum likelihood/MAP) for the “best guess” parameter.
- To get reliable results, you have to make sure the chain runs for many *correlation lengths*.
- If you have multiple solutions in totally different parts of parameter space, you need a better algorithm or *annealing*.
- Complex distribution and *pretty* plots need more steps.

Affine-Invariant Monte-Carlo

- In the Liu/Dupuy/Ireland work, we used a trial chain, then chose new Metropolis-Hastings directions as linear combinations of parameters using principle component analysis on the trial chains.
- This works, but is regarded as dodgy because the algorithm as a whole violates the Markov property.
- A better idea is to find an algorithm that works equally well on any linear combination of parameters.
- These are trickier to code… but luckily other people have coded them for us! E.g. *emcee* which is in anaconda.


Period (years)
5
10
15
20
25
30
0.8
0.6
0.4
0.2
0.0
Eccentricity
Blue chain new trial point
Red chain
Blue chain
10
15
20
25
30
Period (years)
0.8
0.6
0.4
0.2
0.0
Eccentricity

Insert Python Example Here

emcee
The MCMC Hammer

emcee is an extensible, pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo (MCMC) Ensemble sampler. It's designed for Bayesian parameter estimation and it's really sweet!

Feedback

Feedback is greatly appreciated. If

emcee

Seriously Kick-Ass MCMC

`emcee` is an MIT licensed pure-Python implementation of Goodman & Weare's Affine Invariant Markov chain Monte Carlo (MCMC) Ensemble sampler and these pages will show you how to use it.

This documentation won't teach you too much about MCMC but there are a lot of resources available for that (try this one). We also published a paper explaining the `emcee` algorithm and implementation in detail.

emcee has been used in quite a few projects in the astrophysical literature and it is being actively developed on GitHub.

Summary

- Integrals in Bayesian inverse problems are often stupidly difficult to compute. The solution is *Monte-Carlo* integration.
- In most situations, *Monte-Carlo Markov Chain* integration is fastest, because it computes $P(D|\theta)$ for parameters θ only in the region where they are most likely given the data D.
- Although writing your own code for the Metropolis-Hastings algorithm is super-fun and relatively easy, once you get to *affine invariant ensemble MCMC*, and you want parallelizable code, it is easier to use pre-made tools e.g. `emcee`.