Calibration, Systematic Errors, and AstroStatistics

Vinay Kashyap
CXC / SAO
CHASC

we need statistics because we want error bars

"Four score plus-or-minus seven years ago ..."

– Abraham Lincoln
(Gettysburg address first draft)

Outline

I. Statistics as a black box: the promise and perils of χ^2 and model fitting

II. Bayesian probabilities: conceptual clarity vs computational complexity

III. Systematic errors: if you can't beat 'em, join 'em

“The only thing we have to chisquare is chisquare itself.”

– Franklin D. Roosevelt
(after the stock market went downhill like a simplex)

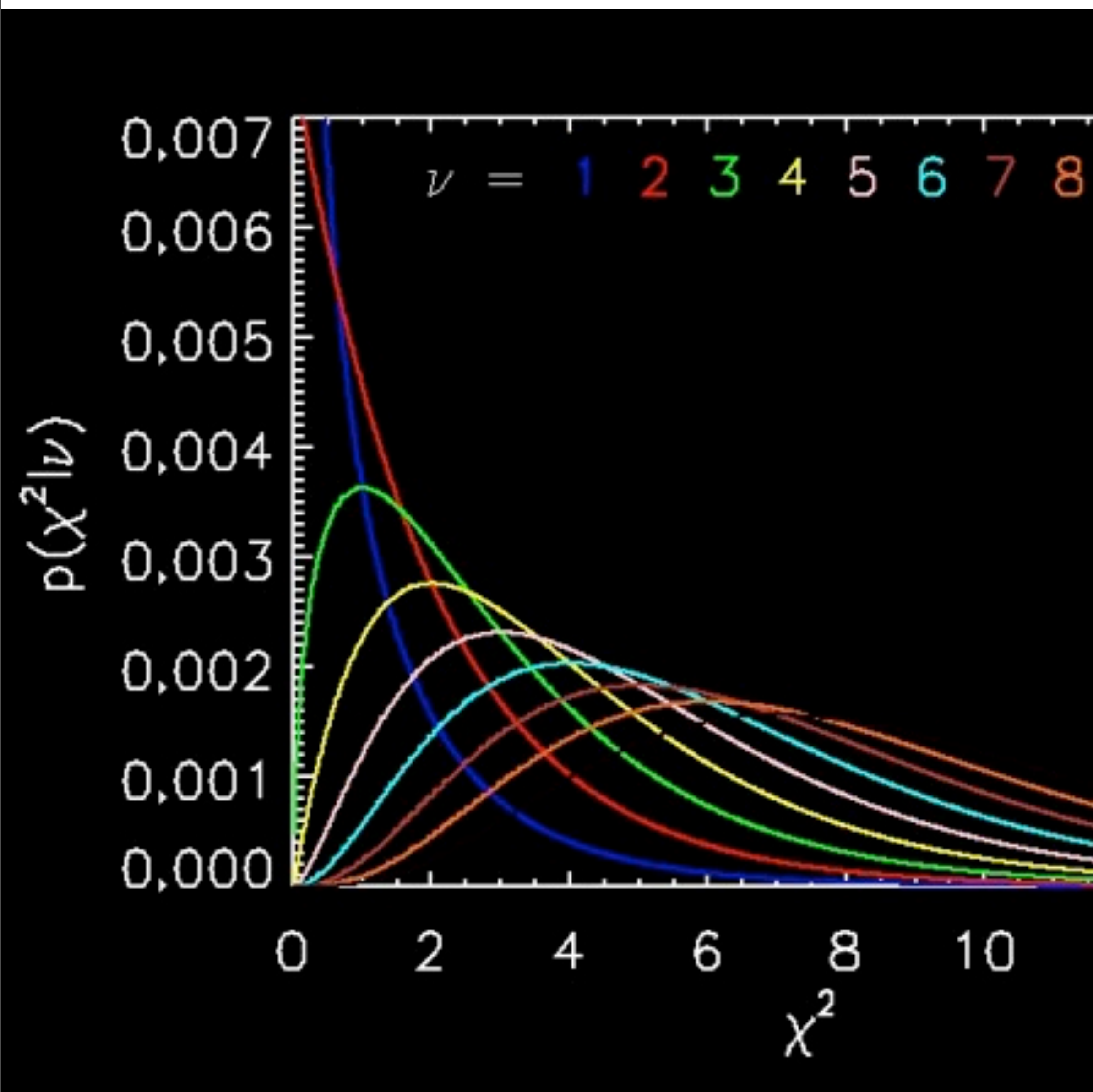
χ^2

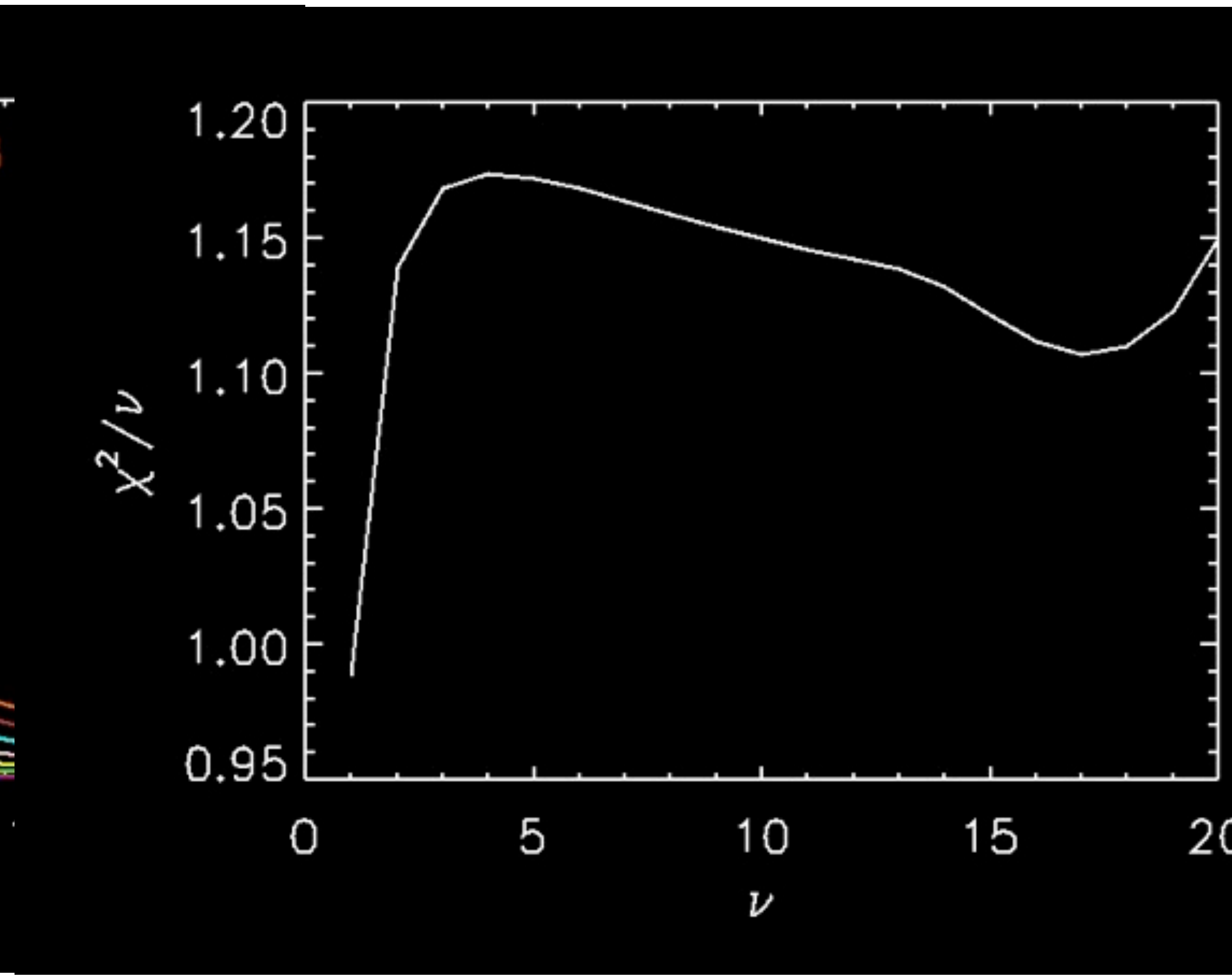
- the statistic vs the goodness-of-fit
  - you can always use the statistic; it is the interpretation that is tricky

statistic vs goodness-of-fit

$$\chi^2 = \sum_i (D_i - M_i)/\sigma_i^2 \equiv -2 \ln \mathscr{L}_G$$

$$p(\chi^2 \,|\nu) = \gamma(\chi^2; \nu/2, 1/2) \text{ ; mean } = \nu \text{ ; variance } = 2\,\nu$$

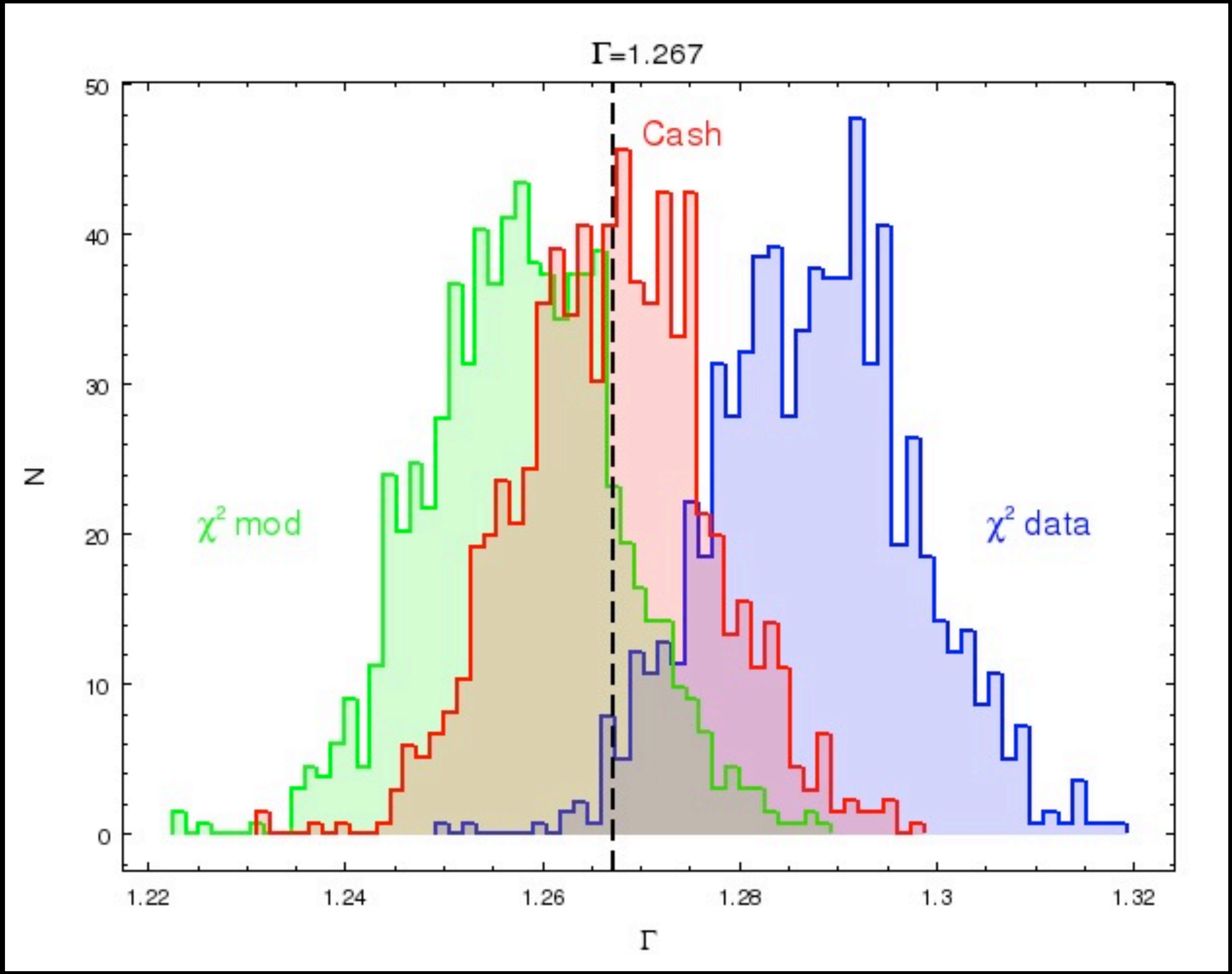




χ^2

- the statistic vs the goodness-of-fit
  - you can always use the statistic; it is the interpretation that is tricky
- reduced χ^2
  - $\chi^2 \sim \nu \pm \sqrt{2\nu}$
  - if errors are incorrect, or deviations are non-linear, or bins are not independent, the χ^2 statistic is not distributed as the χ^2 distribution

scary plot #1: bias



If the likelihood is not appropriate, you may not get the *best* fit.

[A. Siemiginowska]

Poisson likelihood and `cstat`

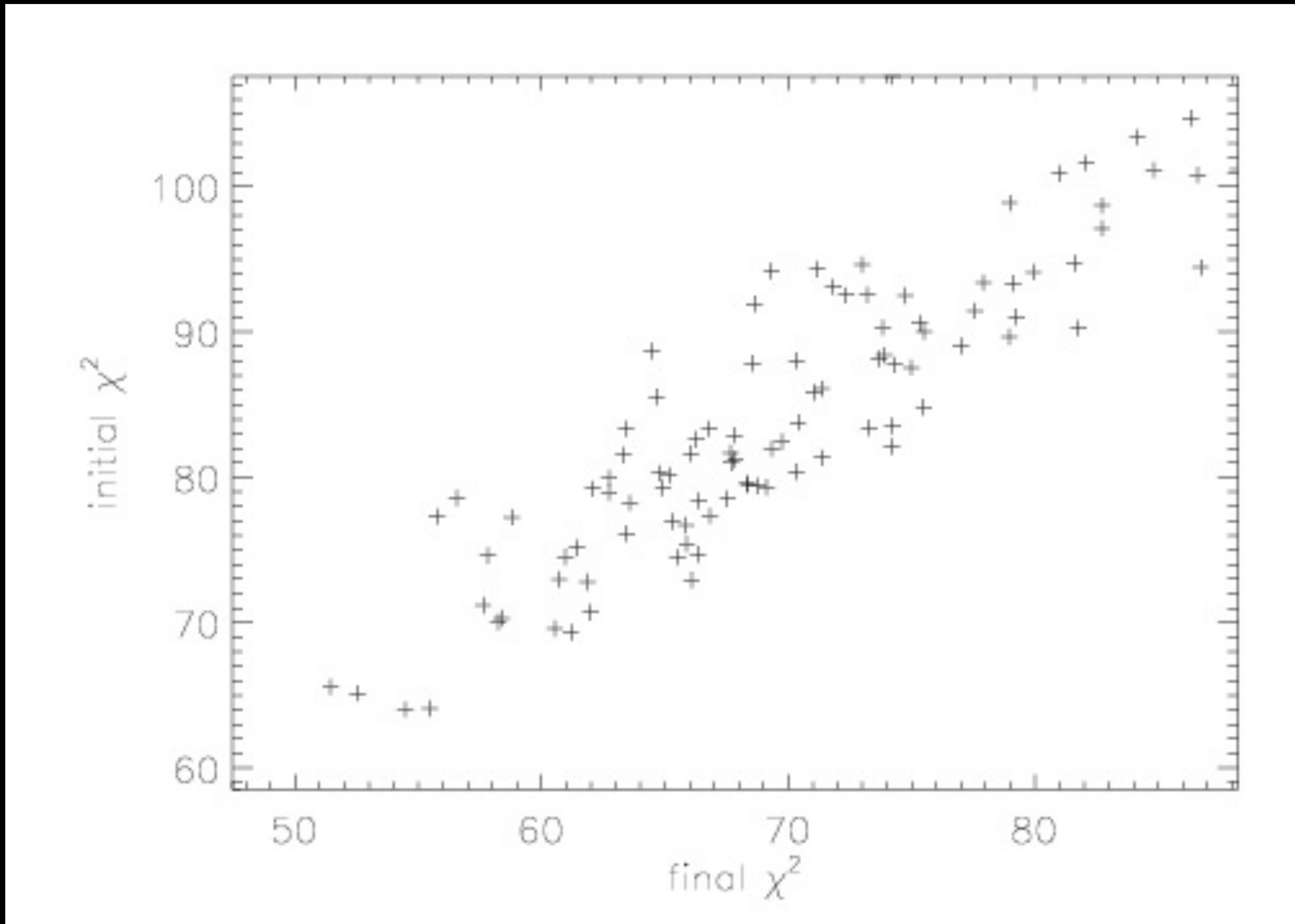
- $\mathscr{L}_P = \Pi_i \, M_i^{D_i} \, e^{-M_i} / \Gamma(D_i)$

- $-2 \ln \mathscr{L}_P = 2 \sum_i ((M_i - D_i) + D_i (\ln D_i - \ln M_i))$

issues with fitting: convergence

"Are we there yet?"

scary plot #2: stopping rule

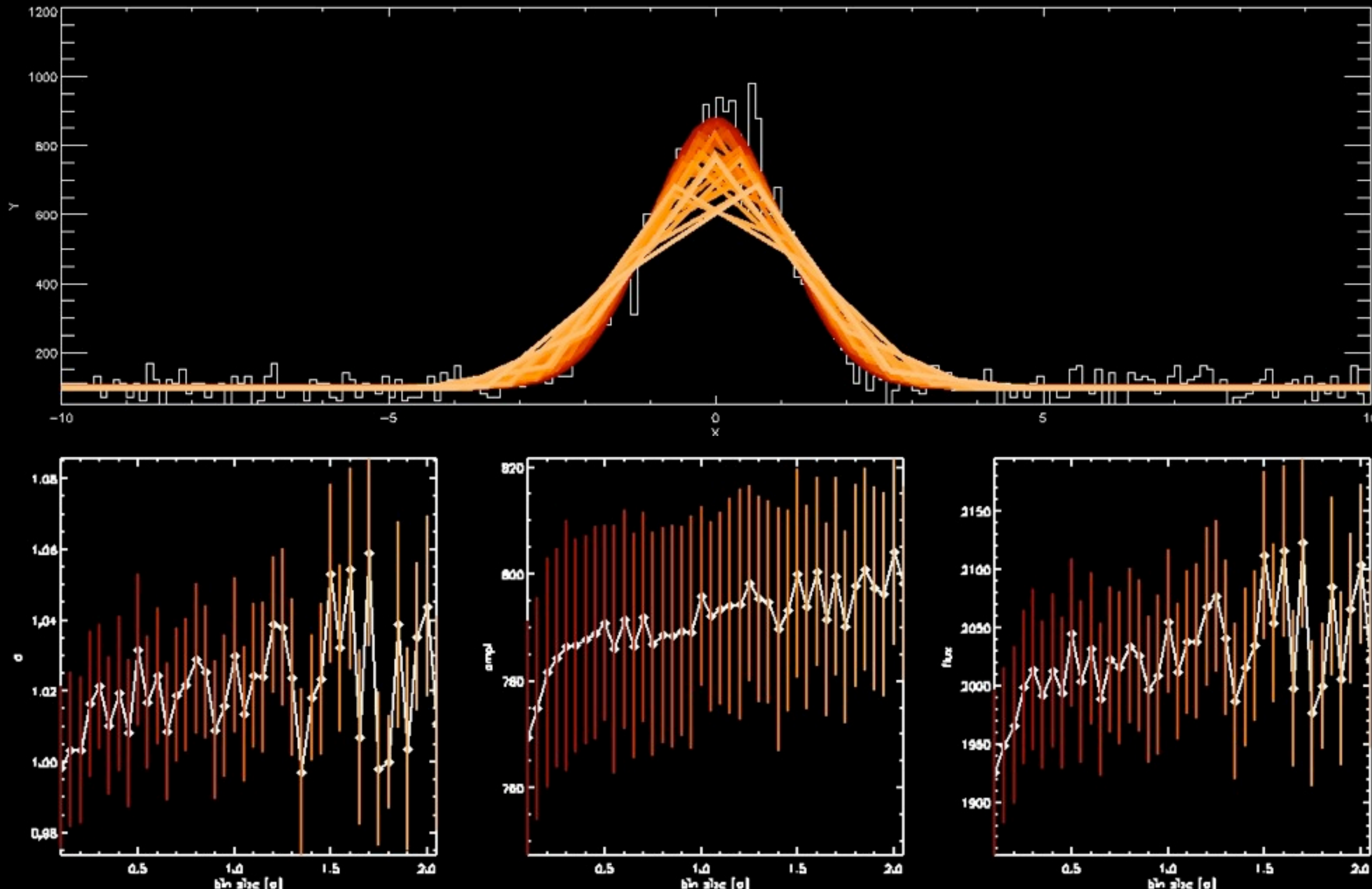


issues with fitting: binning

"To bin, or not to bin. That is the question."

– Hamlet, Prince of Denmark
(he eventually decided to bin. what a tragedy.)

scary plot #3: binning

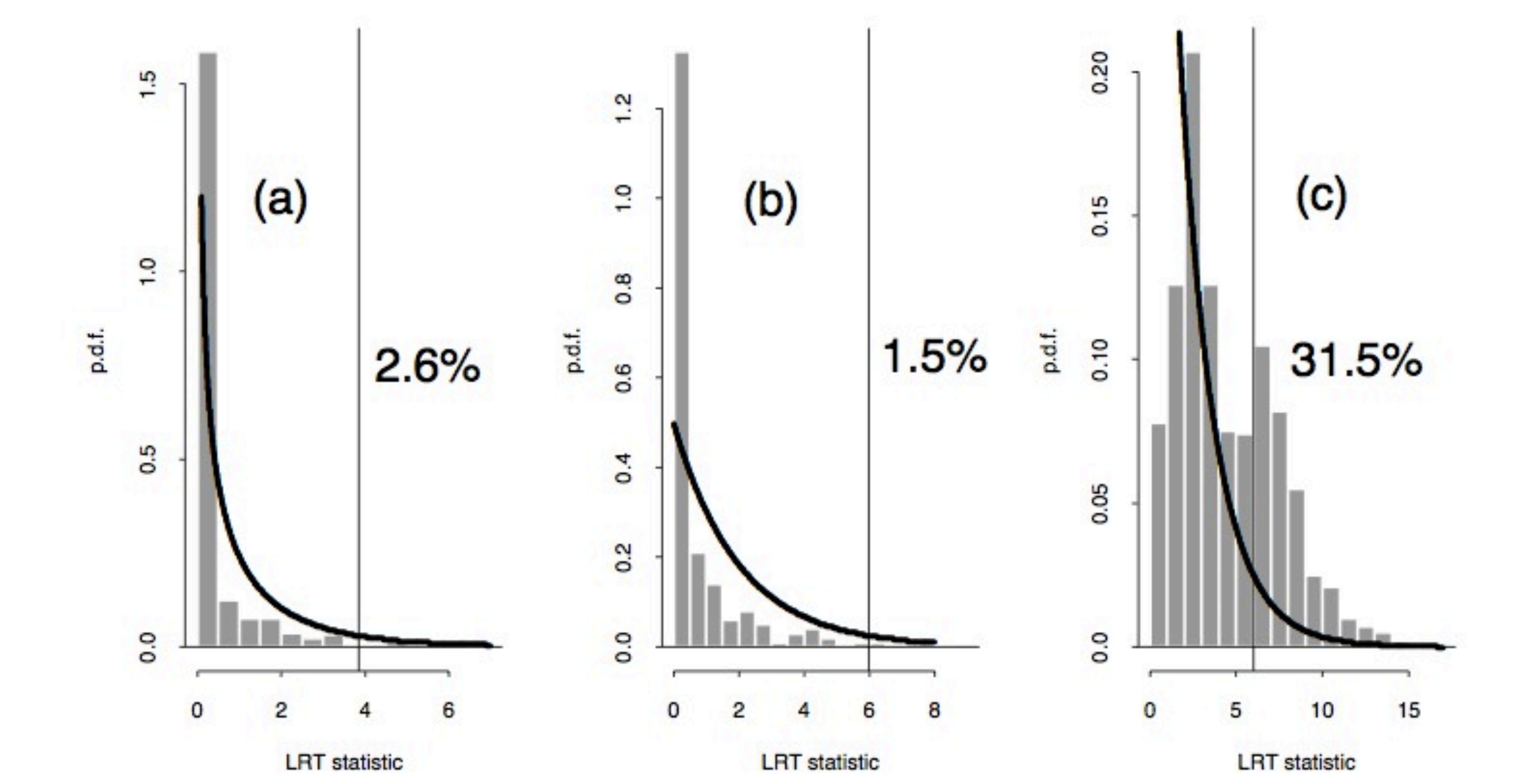


issues with fitting: model comparison

"Is that a spectral line I see before my eyes?
Come, let me flux thee."

– MacBeth, Thane of Glamis
(soon realizes he may have gotten a best-fit, but not necessarily a *good* fit.)

scary plot #4: F-test



narrow emission line | broad emission line at nominal 5% | broad absorption line

You end up not finding as many emission lines as you could, and finding way more absorption lines than you should.

[Protassov et al, 2002, ApJ, 571, 545]

alleviation

- Avoid binning as much as possible
- Use data-appropriate statistics, e.g., Cash or cstat with Poisson counts
  - if there is no help for it, calibrate the statistic
- Avoid subtracting the background, model it instead
- fit in multiple iterative steps, nudging solution to peak of likelihood surface
- Use MCMC to explore parameter space

II. Bayesian Statistics
or as we should say, Laplacian Statistics

"Are you feeling *lucky*, punk?"

– Dirty Harry
(little known fact: the Rev.Bayes kept a 44 gauge shotgun)

II. Bayesian Statistics
or as we should say, Laplacian Statistics

- probability vs ensembles
- *a priori* and conditional probability
- Bayes' Theorem
- MCMC

frequentist vs Bayesian

- the data are fixed, and the parameters have uncertainties (data are not ensembles resulting from true fixed parameter)
- uncertainty is range in parameter values that encompasses a certain probability (unlike confidence interval that overlaps true value a certain fraction of the time)
- prior assumptions are explicit, and probability distributions can be daisy chained

probability calculus

"Alea jacta est"

– Gaius Iulius Caesar

(the theory of conditional probabilities goes back 2060 years)

probability calculus

- jargon: $p(\theta)$, $p(\theta|D)$, $p(A+\bar{A})=1$
- marginalization:

  $p(x|z) = \int dy\ p(xy|z)$
- Bayes' Theorem:

  $p(\theta D) = p(\theta)\ p(D|\theta) = p(D)\ p(\theta|D)$

  $p(\theta|D) = p(\theta)\ p(D|\theta)\ /\ p(D)$

Markov Chain Monte Carlo

- directed and efficient Monte Carlo
- An ordered chain of parameter values that depends only on the previous state

  $p(\theta_i \mid \theta_{i-1}, \theta_{i-2}, .., \theta_1) = p(\theta_i \mid \theta_{i-1})$

- sample θ_i, accept for sure if likelihood improves, and with small probability even if it doesn't

Markov Chain Monte Carlo

- directed and efficient Monte Carlo
- An ordered chain of parameter values that depends only on the previous state

  $p(\theta_i \mid \theta_{i-1}, \theta_{i-2}, .., \theta_1) = p(\theta_i \mid \theta_{i-1})$

- sample θ_i, accept for sure if likelihood improves, and with small probability even if it doesn't

"One small step in iteration, one probability distribution for parameter."

– Neil Armstrong
(shooting for the Moon)

Markov Chain Monte Carlo

- directed and efficient Monte Carlo
- An ordered chain of parameter values that depends only on the previous state

  $p(\theta_i \mid \theta_{i-1}, \theta_{i-2}, .., \theta_1) = p(\theta_i \mid \theta_{i-1})$

- sample θ_i, accept for sure if likelihood improves, and with small probability even if it doesn't
- histogram of parameter values asymptotically gives marginalized posterior probability distribution
- combines acceptance, rejection, and importance sampling, and preferentially samples from higher probability regions

MCMC: the Surgeon General's warning

- no guarantee of convergence: run multiple chains from different starting points
- chain must be homogeneous and ergodic
- always check the trace of the parameters
    - has it reached an asymptotic limit?
    - are the draws independent?
    - is the sampling efficient?

III. Systematic Errors

- the main focus of this group is to identify and eliminate systematic errors
- but not always possible. in that case, can we somehow include them as part of analysis?

  (yes)
- make available representative samples of ARFs, RMFs, PSFs, and we will take care of the rest

Main Uncertainties in Instrument Response: Chandra ACIS–S

HRMA
- Geometry
- Obscuration
- Reflectivity
- Scattering

ACIS OBF
- Transmittance
- Contamination
- Uniformity

ACIS CCD
- QE
- Gain
- Pulse Height Distribution

random variations of input parameters

$\mu(.)$: multiplicative perturbative functions

$\Omega(\sigma)$: truncated Gaussian

– μ_H

– sample contam models

– vignetting $V(\theta)$ from

$\mu_V(E,\theta) = \Omega(\sigma_V)(1-V(\theta))$

$+ \theta\ \Omega(\sigma_S)(1-R_{DW}/R)$

$\sigma_V,\sigma_S = 0.2$

– $\mu_{OBF}(E)$

– Contamination Layer

$\ln(\mu_{CL}(E)) = -\sum_X \Omega(\sigma_X)\tau_X$

$X \equiv$ C,O,F,Fl

$\mu_{CL}(0.7\ \mathrm{keV}) < 0.05$

– $\mu_{QE}(E)$

– 13% in CCD depletion depth and 20% in SiO_2 thickness

– $\Omega(\sigma_G)$, $\sigma_G=1\%$ @0.7 keV, 0.5% @1.5 keV, 0.2% @≥4 keV

[J. Drake]

FLASHBACK FROM IACHEC 2010

coming soon to a console near you

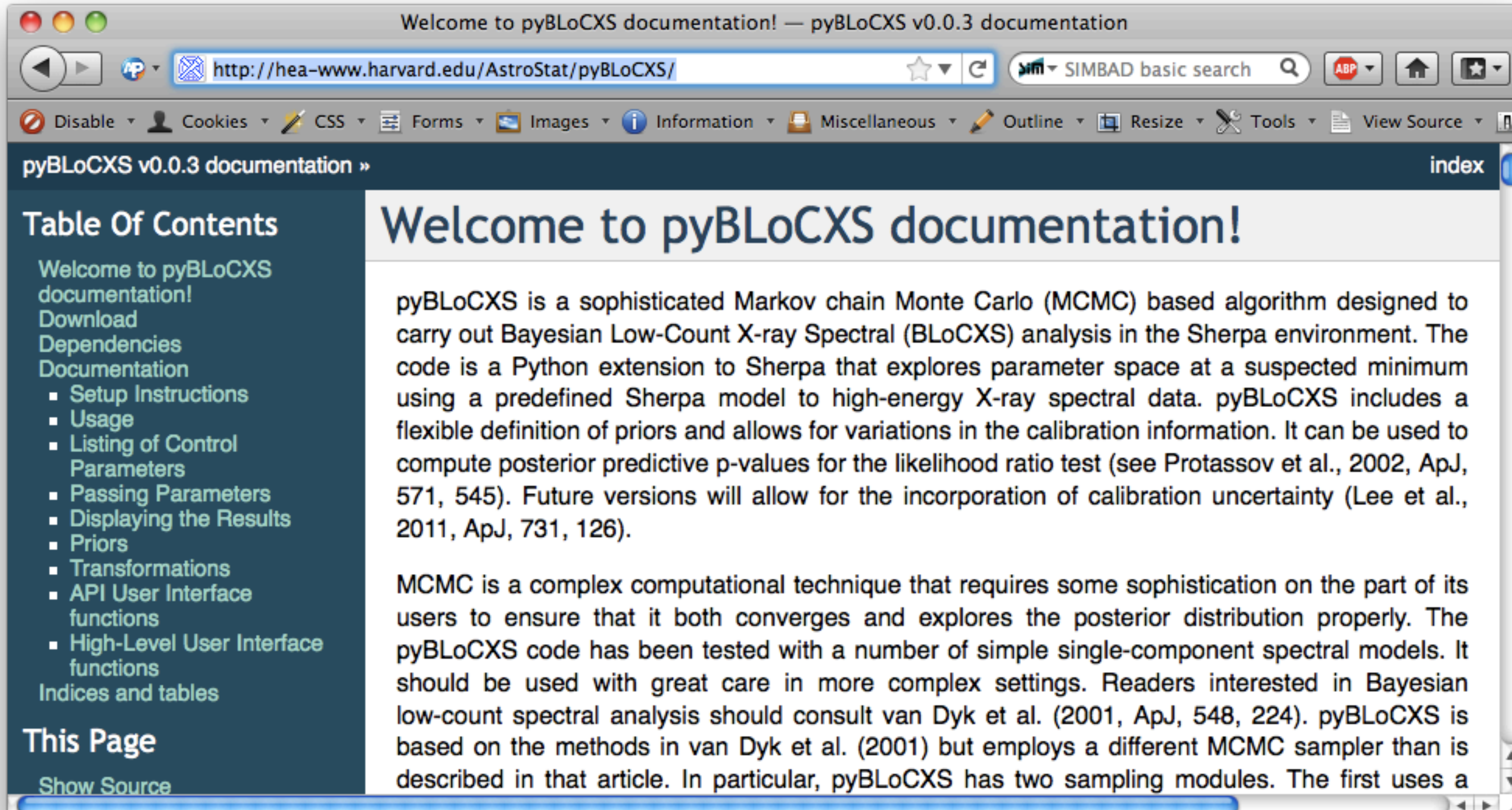
pyBLoCXS

pyBLoCXS

- MCMC fitting engine in CIAO/Sherpa
- python code

pyBLoCXS

- MCMC fitting engine in CIAO/Sherpa
- python code



Welcome to pyBLoCXS documentation! — pyBLoCXS v0.0.3 documentation

http://hea-www.harvard.edu/AstroStat/pyBLoCXS/

pyBLoCXS v0.0.3 documentation » index

Table Of Contents

This Page

Show Source

Welcome to pyBLoCXS documentation!

pyBLoCXS is a sophisticated Markov chain Monte Carlo (MCMC) based algorithm designed to carry out Bayesian Low-Count X-ray Spectral (BLoCXS) analysis in the Sherpa environment. The code is a Python extension to Sherpa that explores parameter space at a suspected minimum using a predefined Sherpa model to high-energy X-ray spectral data. pyBLoCXS includes a flexible definition of priors and allows for variations in the calibration information. It can be used to compute posterior predictive p-values for the likelihood ratio test (see Protassov et al., 2002, ApJ, 571, 545). Future versions will allow for the incorporation of calibration uncertainty (Lee et al., 2011, ApJ, 731, 126).

MCMC is a complex computational technique that requires some sophistication on the part of its users to ensure that it both converges and explores the posterior distribution properly. The pyBLoCXS code has been tested with a number of simple single-component spectral models. It should be used with great care in more complex settings. Readers interested in Bayesian low-count spectral analysis should consult van Dyk et al. (2001, ApJ, 548, 224). pyBLoCXS is based on the methods in van Dyk et al. (2001) but employs a different MCMC sampler than is described in that article. In particular, pyBLoCXS has two sampling modules. The first uses a

pyBLoCXS

- MCMC fitting engine in CIAO/Sherpa
- python code
- http://hea-www.harvard.edu/AstroStat/pyBLoCXS/
- capabilities
  - fit any Sherpa model
  - include log, normal, flat, or user-defined priors
  - multiple chains
  - parameter transformations
- coming soon
  - calibration uncertainty sampler