pyBLoCXS

Vinay Kashyap

Chandra X-Ray Center / Harvard-Smithsonian Center for Astrophysics

Strategy to deal with known errors in effective areas

Vinay Kashyap

Hyunsook Lee, Jin Xu, David van Dyk, Aneta Siemiginowska, Jeremy Drake, Pete Ratzlaff, Andreas Zezas, Xiao-Li Meng, Rima Izem, Alanna Connors

Demo & WG

- Monday *7:30pm* at *Meadow*
  - demo of current capabilities: generating and compressing ARF library, fitting with pyBLoCXS
- Monday *8:30pm* at *Meadow*
  - meeting of Cal Uncertainties Working Group

Spectral model
parameter fitting

Bayesian framework
(pyBLoCXS)

Calibration
uncertainty

select cal subset

Calibration

Spectral model parameter fitting

Calibration uncertainty

Bayesian framework (pyBLoCXS)

select cal subset

Standard analysis (θ±δθ; XSPEC, Sherpa, etc.)

Calibration

Spectral model
parameter fitting

pyBLoCXS: compute $p(\theta|data,A_o)$

Calibration

Spectral model parameter fitting

van Dyk et al. 2001

Bayesian framework (pyBLoCXS*)

* **py**thon-based **B**ayesian **Lo**w-**C**ounts **S**pectral analysis package

pragmatic Bayes: compute $p(\theta|data,\{A\})\ p(\{A\})$

Calibration

Spectral model parameter fitting

Calibration uncertainty

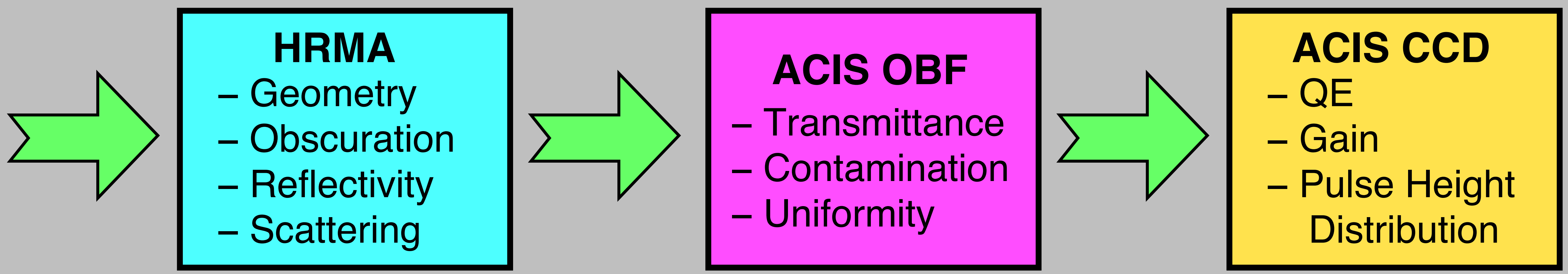
$\{A\} \rightarrow$ AREF

Lee et al. 2011

Bayesian framework (pyBLoCXS*)

* **py**thon-based **B**ayesian **Lo**w-**C**ounts **S**pectral analysis package

Main Uncertainties in Instrument Response: Chandra ACIS–S



random variations of input parameters

$\mu(.)$: multiplicative perturbative functions

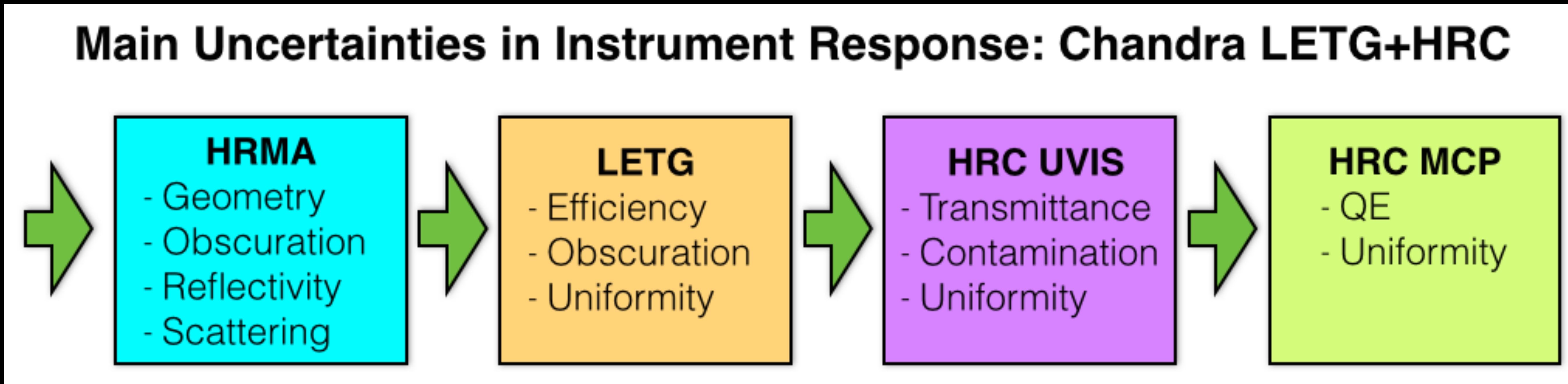
$\Omega(\sigma)$: truncated Gaussian

– 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
$\mu_{CL}(0.7\ \text{keV}) < 0.05$

– $\mu_{QE}(E)$
– 13% in CCD depletion depth and 20% in SiO_2 thickness
– $\Omega(\sigma_G)$, σ_G=1% @0.7 keV, 0.5% @1.5 keV, 0.2% @≥4 keV

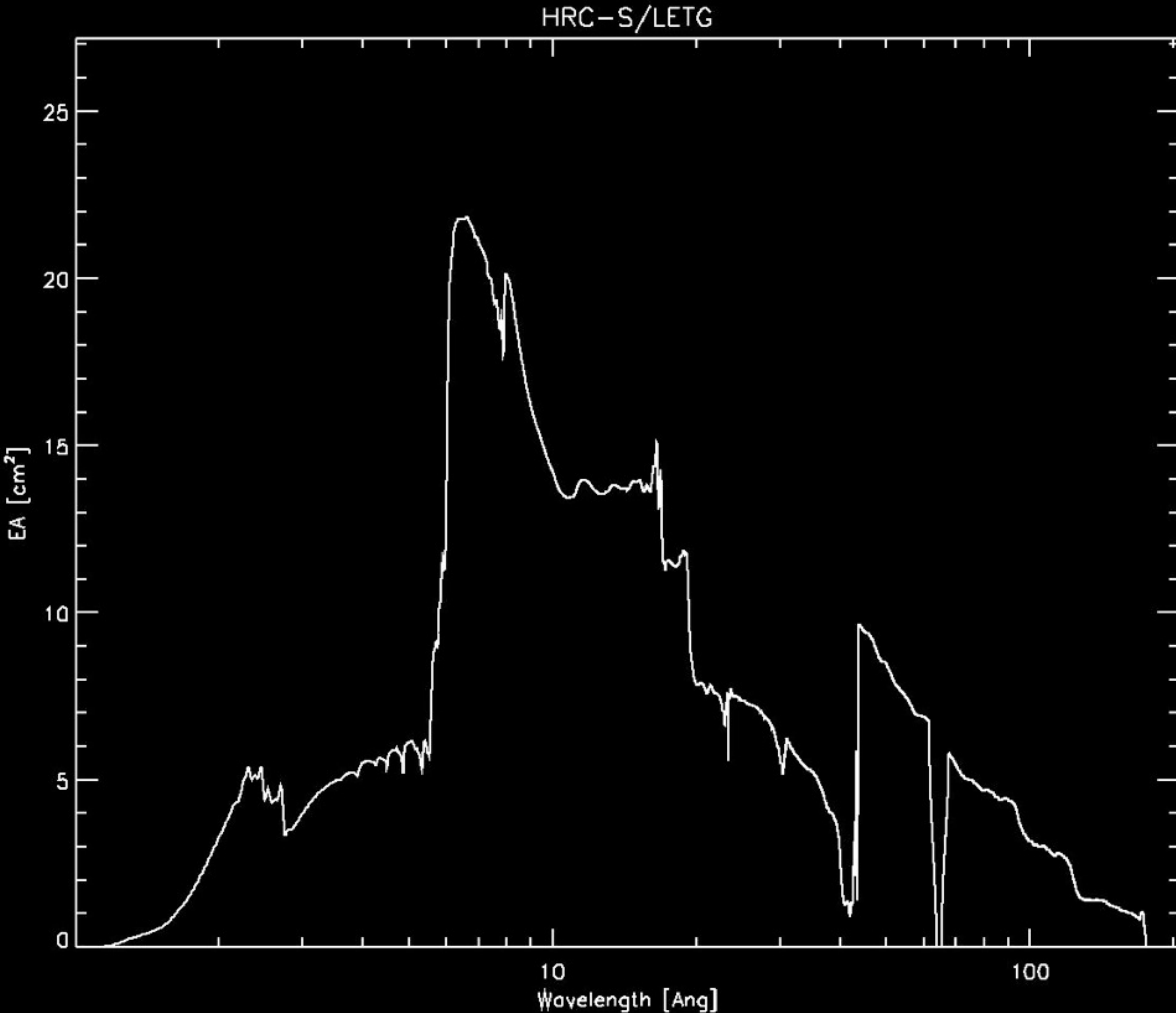
[Jeremy Drake]

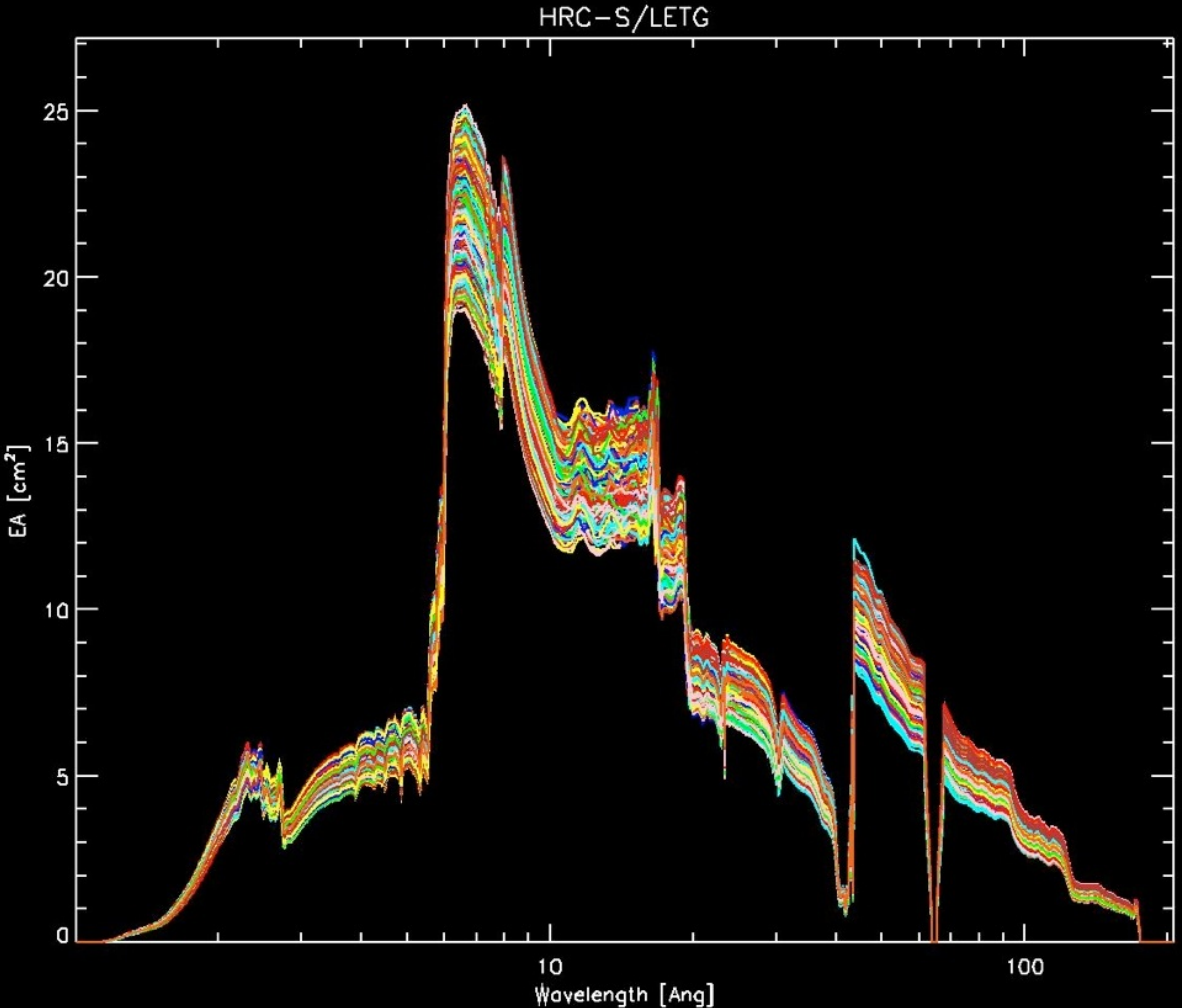


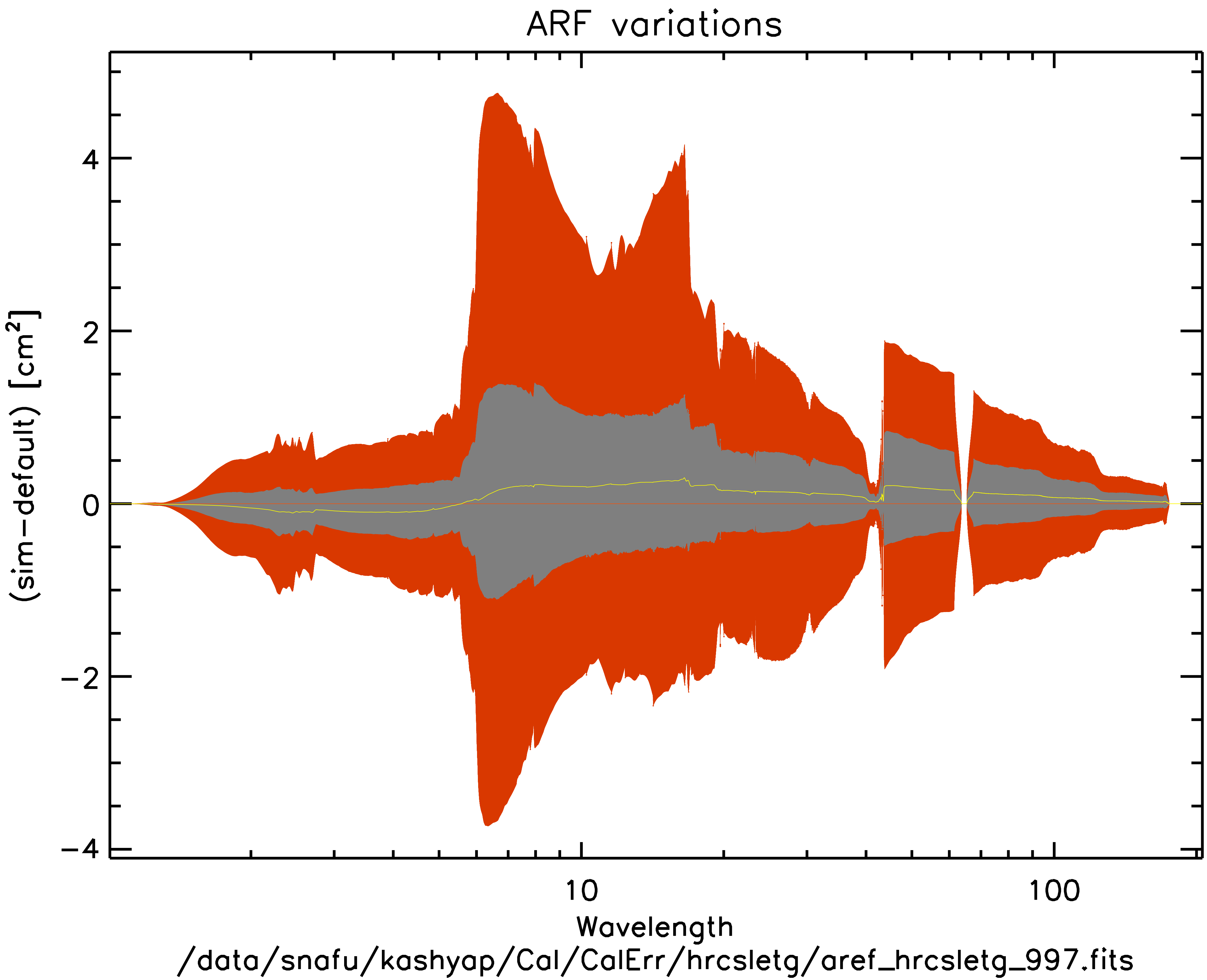
random variations of input parameters

$\mu(.)$: multiplicative perturbative functions

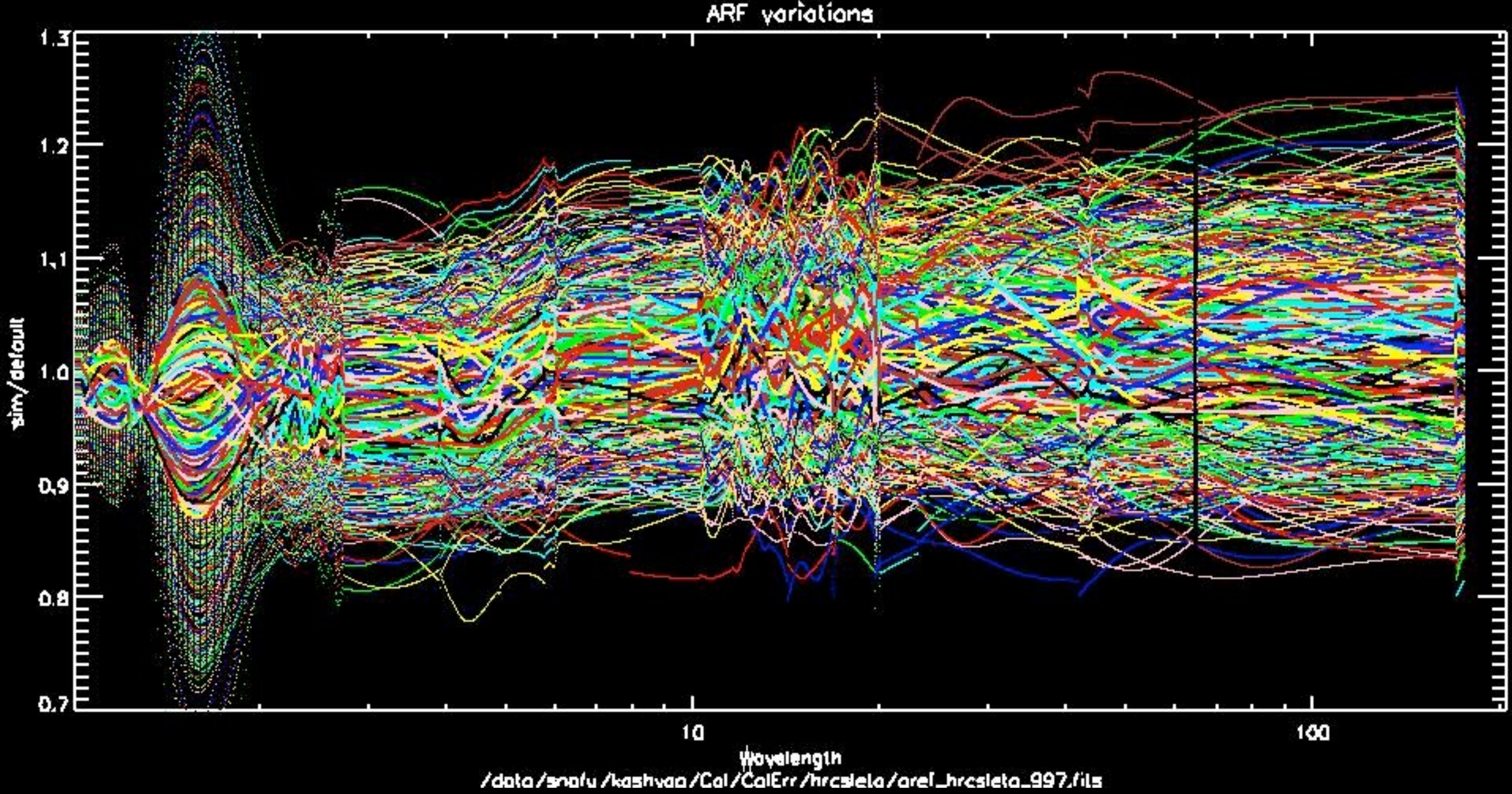
$\Omega(\sigma)$: truncated Gaussian

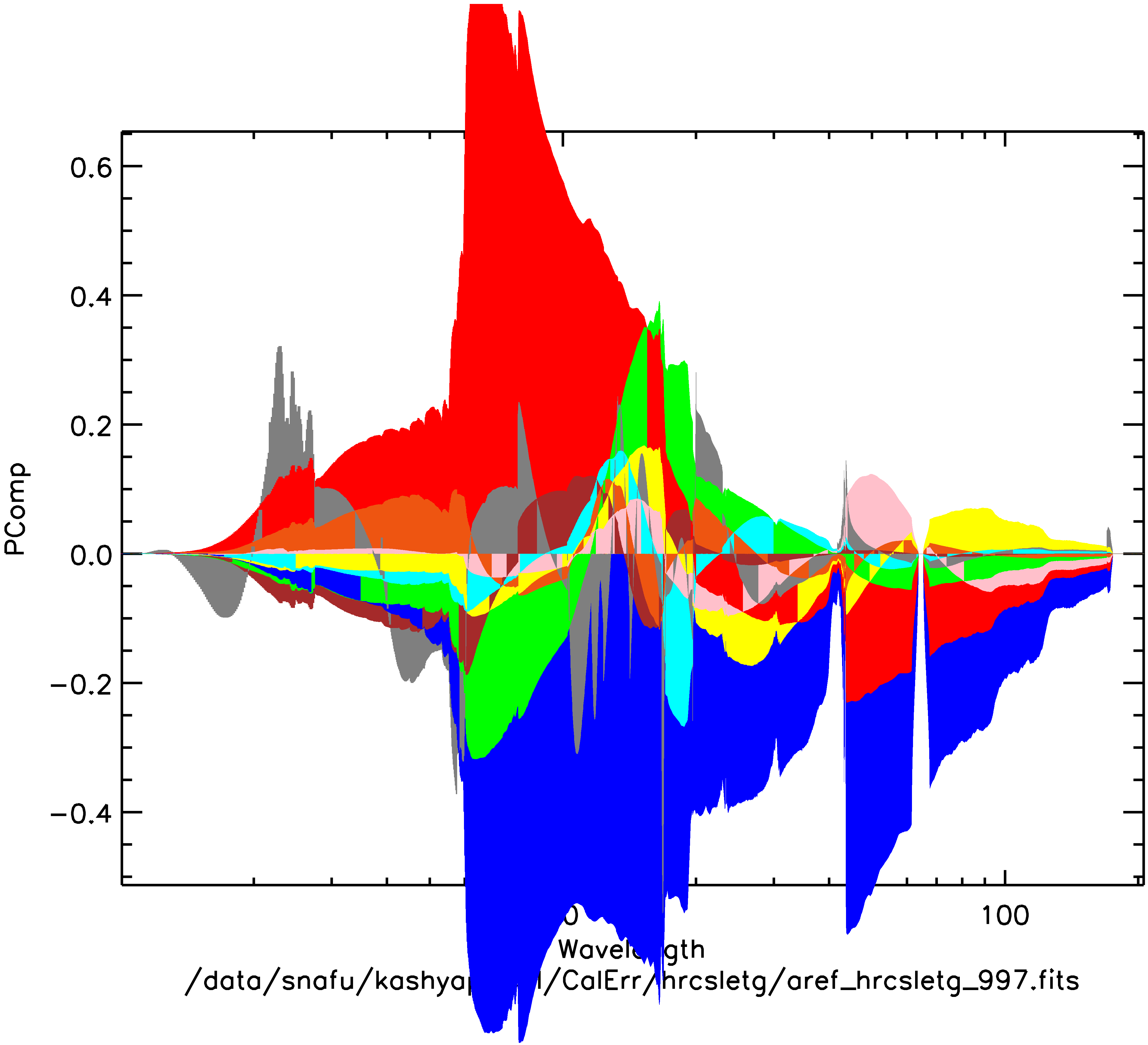

HRC-S/LETG
EA [cm^2]
Wavelength [Ang]
0
5
10
15
20
25
10
100


HRC-S/LETG
EA [cm²]
Wavelength [Ang]
25
20
15
10
5
0
10
100

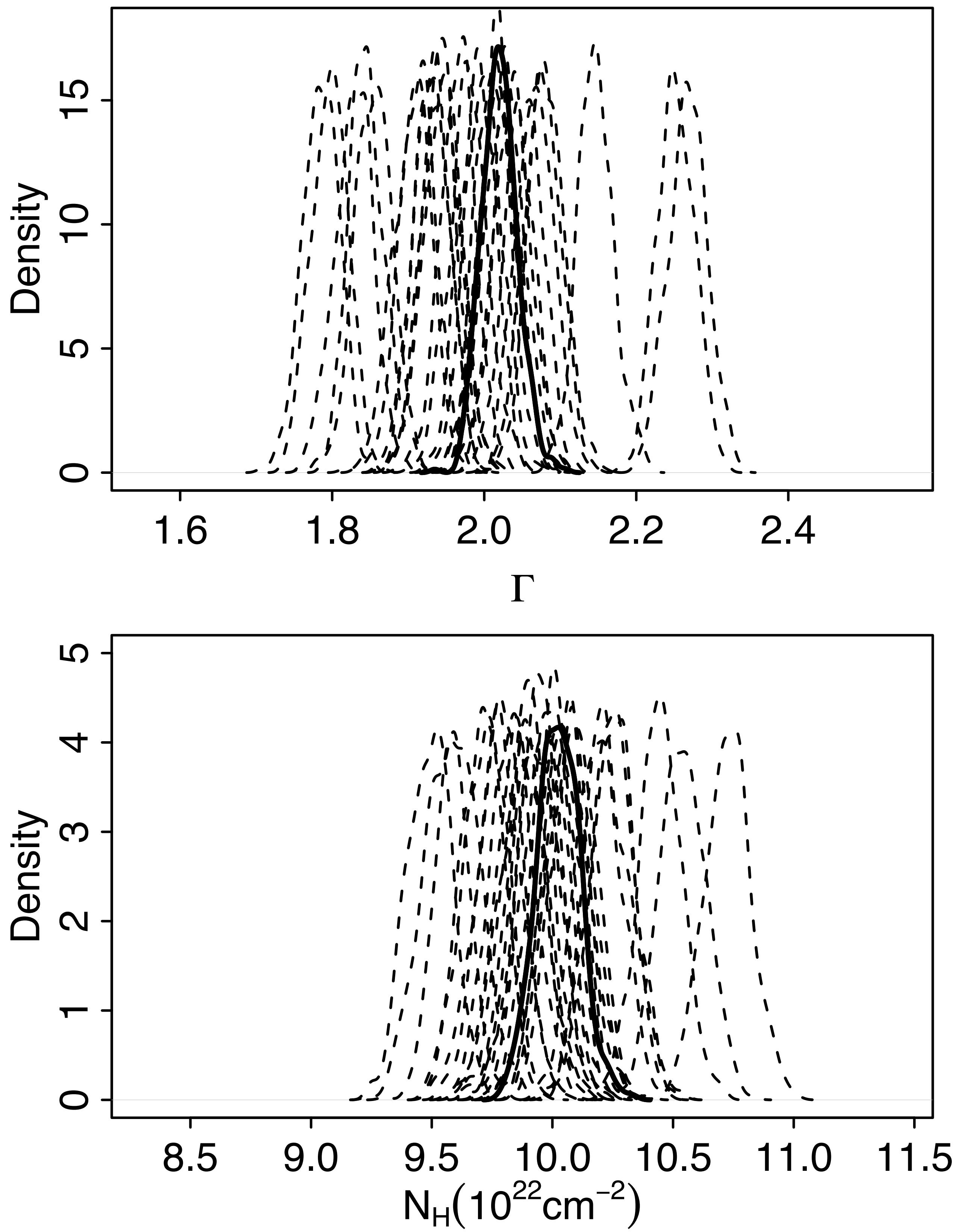


ARF variations
(sim-default) [cm^2]
4
2
0
−2
−4
10
100
Wavelength
/data/snafu/kashyap/Cal/CalErr/hrcsletg/aref_hrcsletg_997.fits


ARF variations
1.3
1.2
1.1
1.0
0.9
0.8
0.7
sim/default
10
100
Wavelength
/data/snafu/kashyap/Cal/CalErr/hrcsleta/arel_hrcsleta_997.fits



PComp
0.6
0.4
0.2
0.0
−0.2
−0.4
100
/CalErr/
hrcsletg/aref_hrcsletg_997.fits



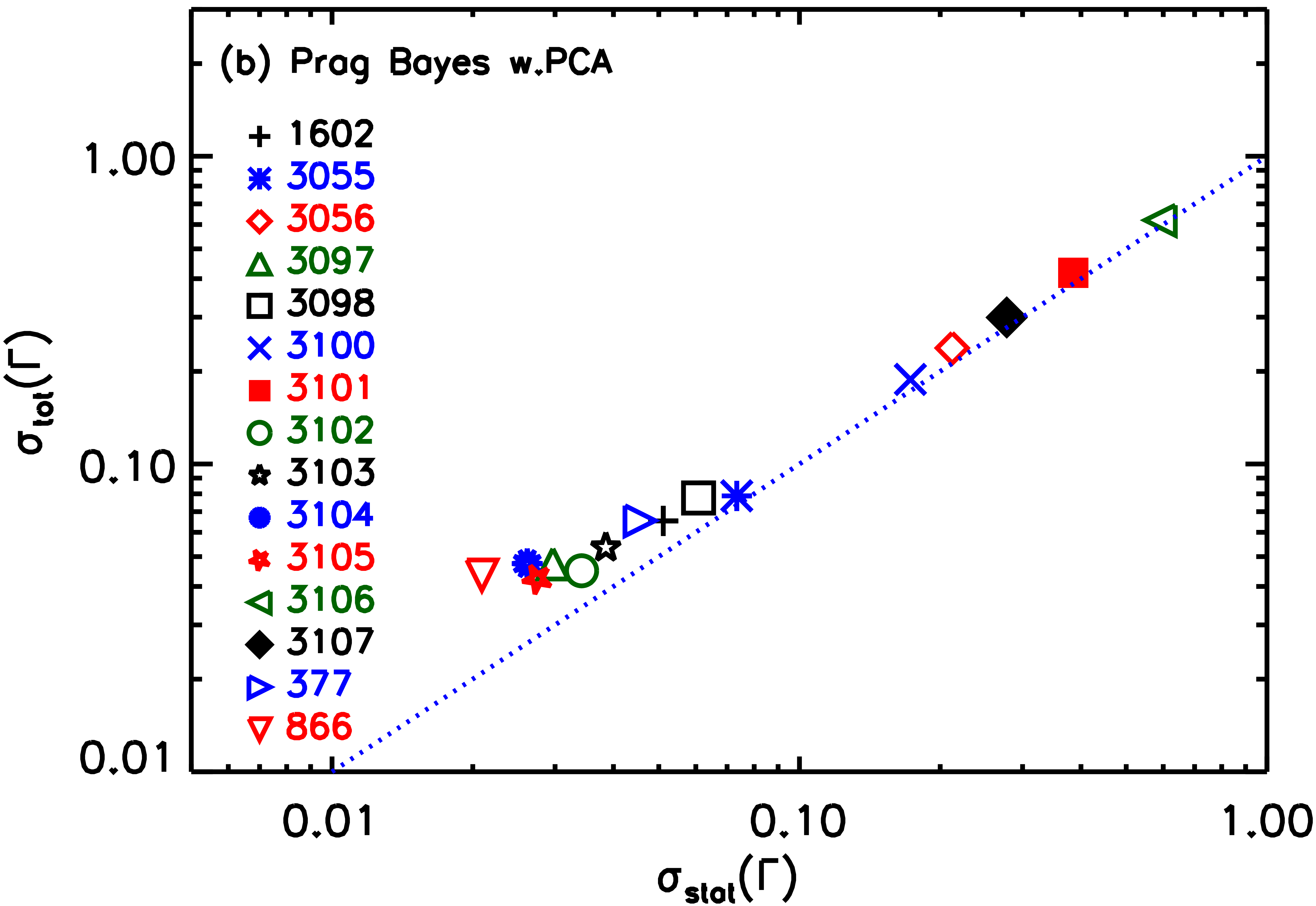
Effect of Cal uncertainty on parameter estimates

(Lee et al. 2011)

- absorbed power-law
- simulation with 10^5 ct
- posterior probability density functions
- for default ARF (solid curve) and 30 randomly drawn ARFs (dashed lines)

Effect of Cal uncertainty on parameter uncertainty

(Lee et al. 2011)



pragmatic Bayes: compute $p(\theta|data,\{A\})\ p(\{A\})$

Calibration

Spectral model parameter fitting

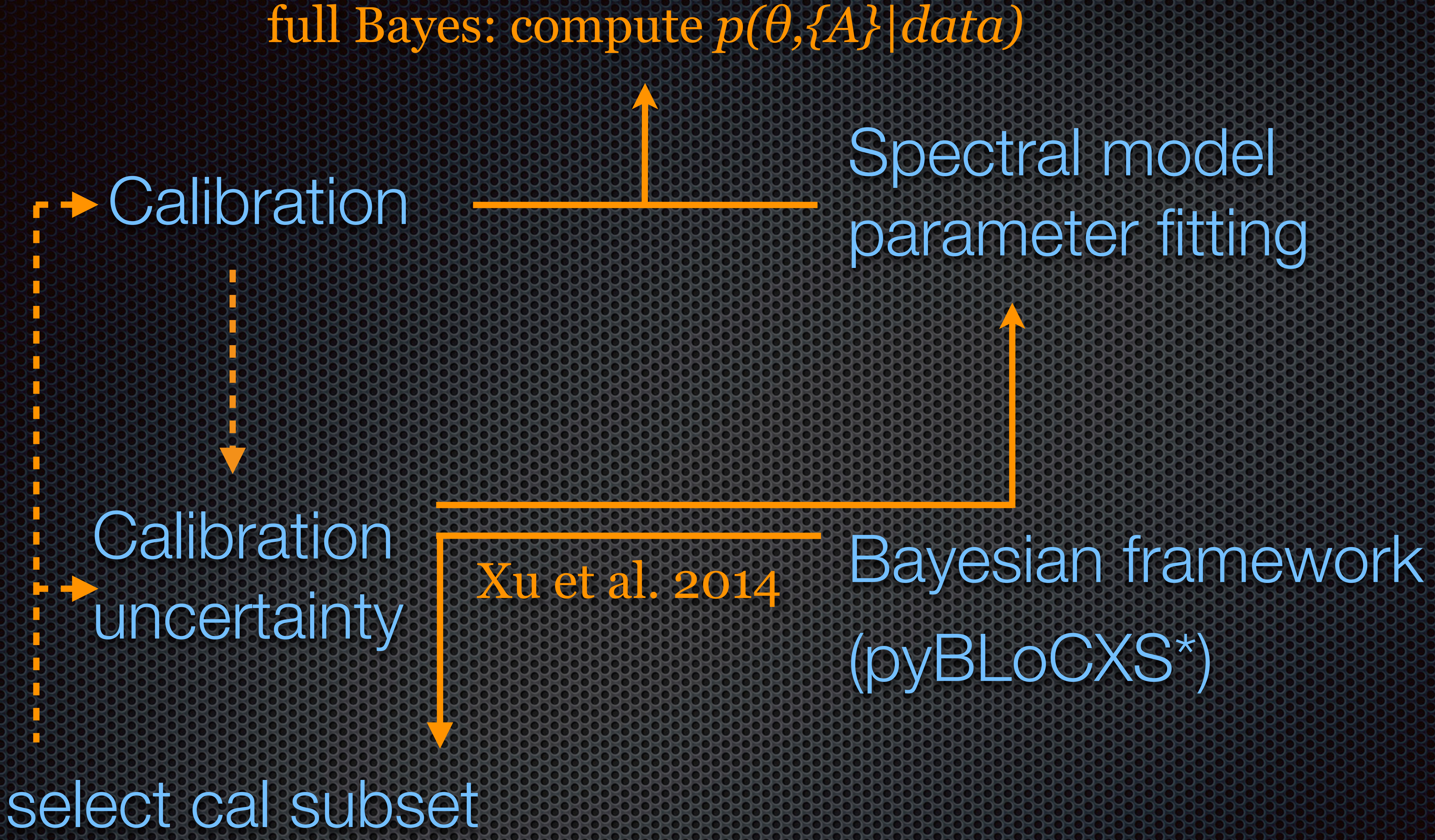
$\{A\} \rightarrow$ AREF

Lee et al. 2011

Calibration uncertainty

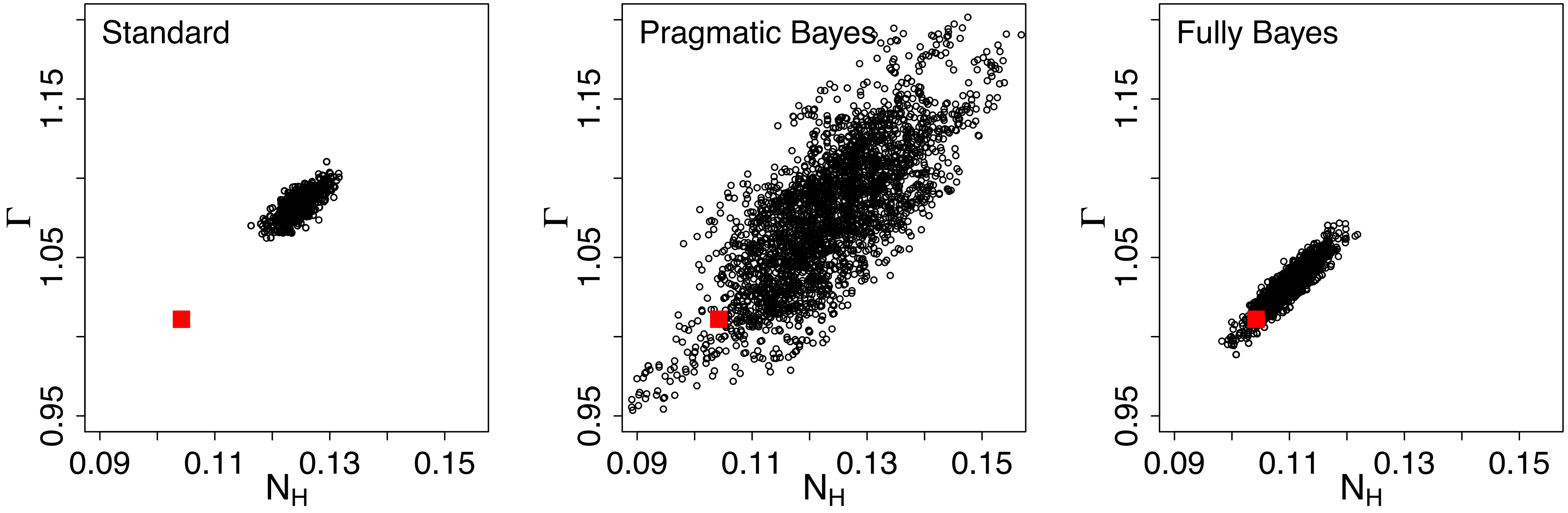
Bayesian framework (pyBLoCXS*)

* **py**thon-based **B**ayesian **Lo**w-**C**ounts **S**pectral analysis package



* **py**thon-based **B**ayesian **Lo**w-**C**ounts **S**pectral analysis package

how does it work?

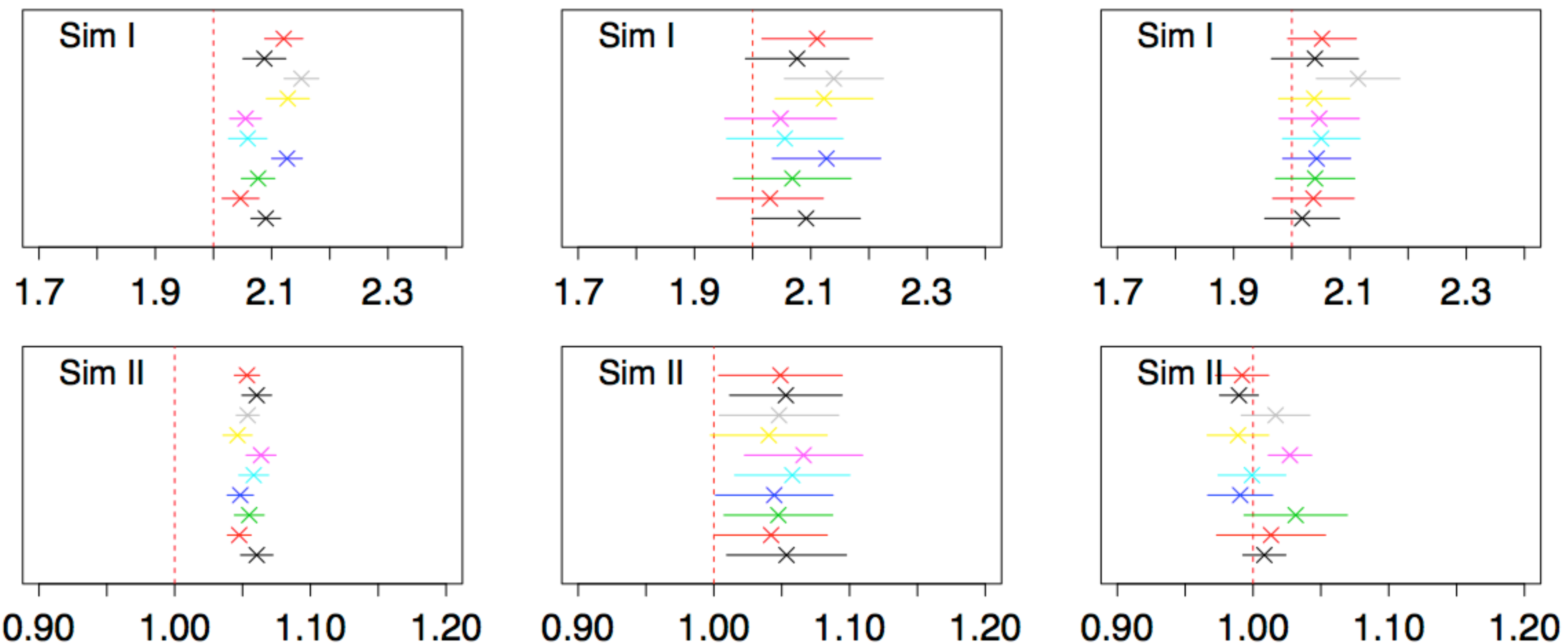


full Bayes analysis

(Xu et al. 2014)

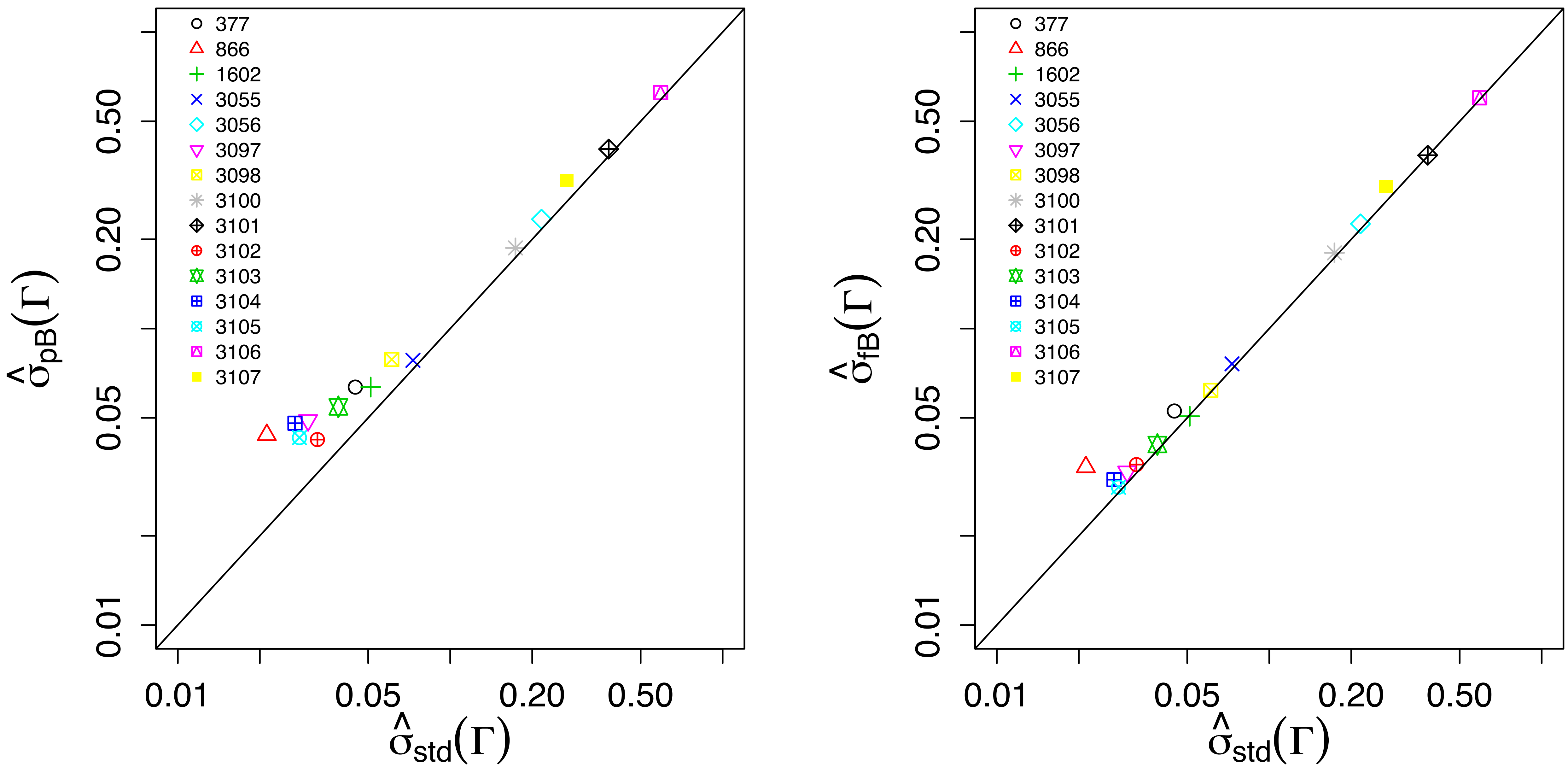
two absorbed power-law simulations with 10^5 counts
for spectra generated using ``bad'' ARF

standard | pragmatic Bayes | full Bayes



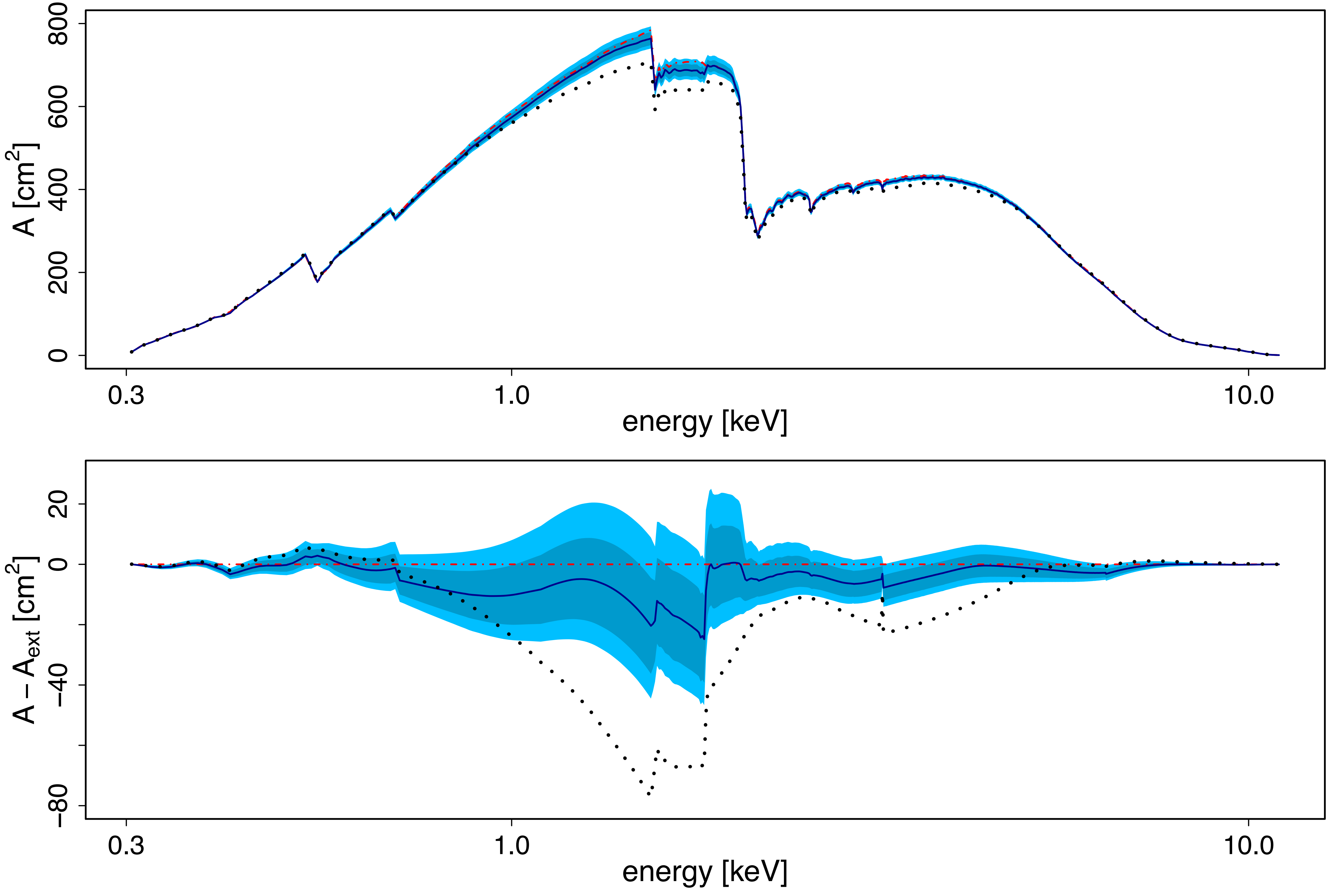
full Bayes analysis effect on parameter uncertainties

(Xu et al. 2014)



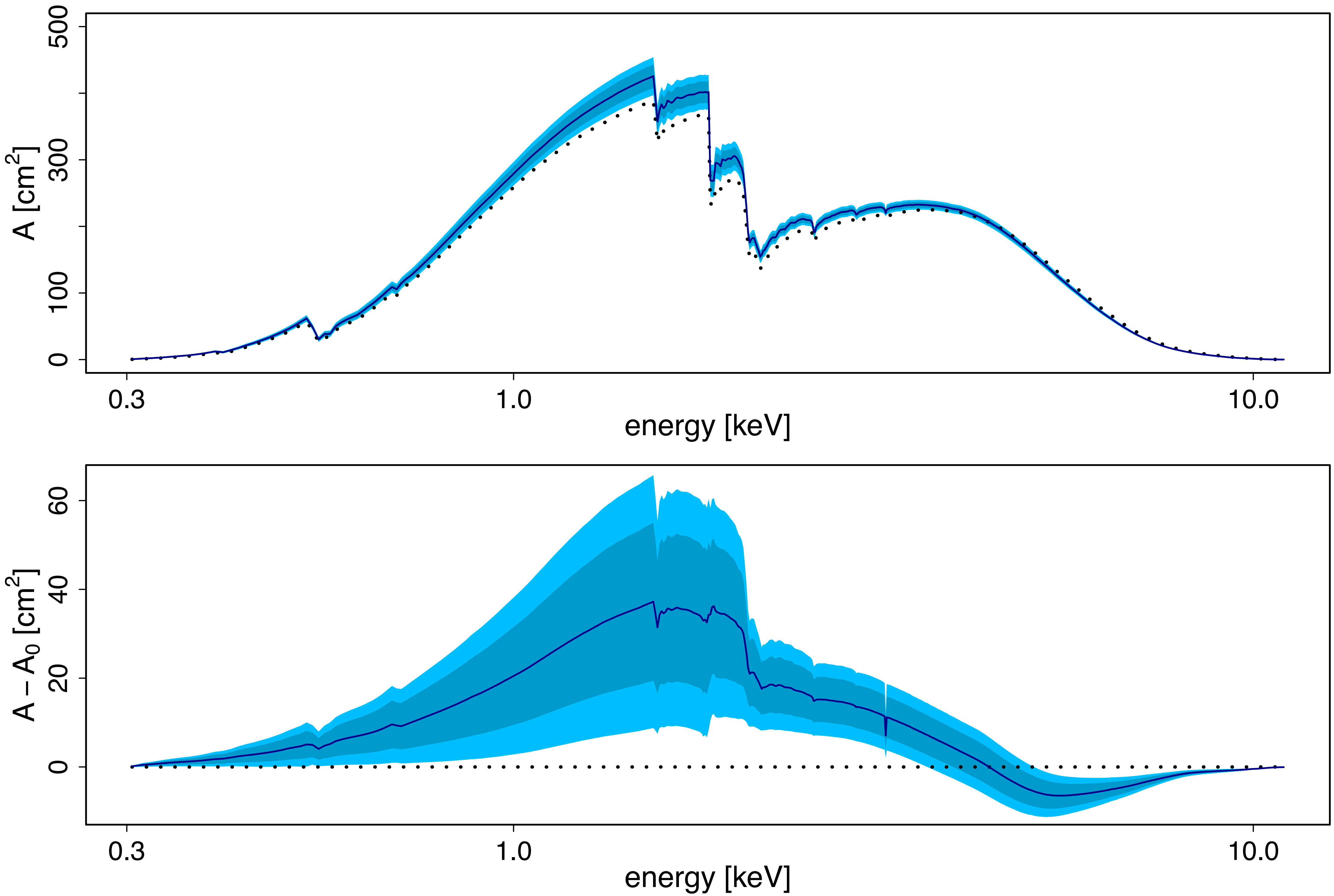
full Bayes effect on effective areas (simulations)

(Xu et al. 2014)



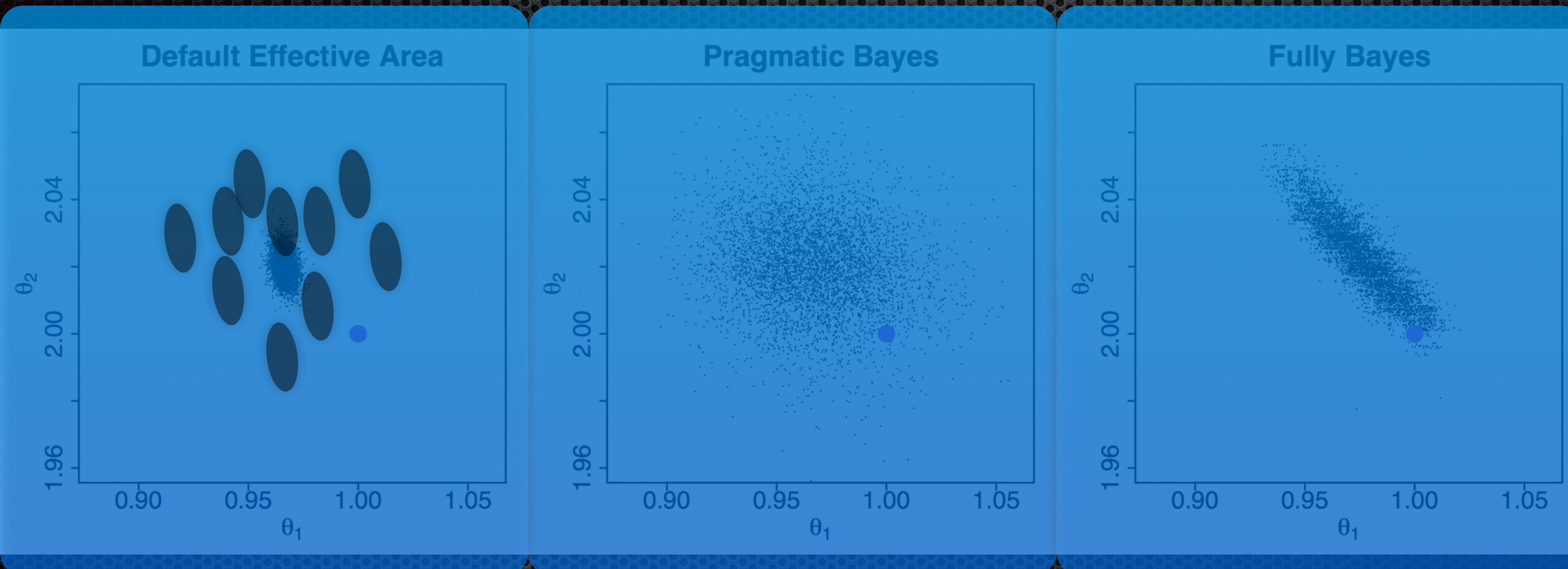
full Bayes effect on effective areas (ζ Ori)

(Xu et al. 2014)



fitting to simulated data

$$f(\varepsilon;\theta) = \theta_3\, \varepsilon^{-\theta_1}\, e^{-\theta_2\, \sigma(\varepsilon)}$$



$p(\theta|D,A_0)$ $p(A)\, p(\theta|D,A)$ $p(A,\theta|D)$

$p(\theta|D,A_i)$ $p(A(\theta'),\theta|D)$

(David van Dyk & Jin Xu)

Demo & WG

- Monday *7:30pm* at *Meadows*
  - demo of current capabilities: generating and compressing ARF library, fitting with pyBLoCXS
- Monday *8:30pm* at *Meadow*
  - meeting of Cal Uncertainties Working Group