NuSTAR Calibration Coffee

#toomanyphotons

2024/10/02 BG

CALDB/Software Updates

Updates since last Calibration Coffee

- No major updates since last Cal Coffee
  - Some issues with intermediate CALDB releases
    - https://nustarsoc.caltech.edu/NuSTAR_Public/NuSTAROperationSite/file/release_20240325.txt
  - Use 20240325 or later
- NuSTARDAS v2.1.4 released for single-laser operations
  - But we hope you never have to use it
  - https://nustarsoc.caltech.edu/NuSTAR_Public/NuSTAROperationSite/file/NuSTARDAS_2.1.4-release.txt

Covered Today

- #toomanyphotons (credit Felix Fuerst, circa 2013)
  - How does NuSTAR handle high count rates?
  - What processing steps should we use?
  - What impact does this have on my data?

- NuSTAR is an **imaging** telescope
  - What do I do when the entire FoV is dominated by source photons?

Bright source example

Background

- NuSTARDAS data provide a number of filtering options, which may or may not be useful for your science application
- For many sources the data span a case where you are source dominated vs background dominated
- In this example, the source is very soft and dominates over the background at low energies, while the background dominates above ~30 keV.
- **NB:** Default processing is specifically tuned for ***faint*** sources, so there are software cuts made on the data that remove a small (~few %) of source photons for bright sources.

`nupipeline` processing
For bright sources - the STATUS bit

- The NuSTAR readout allows for individual events to be veto'd if they look like electronic noise

- This is done during `nupipeline` processing by populating various bits in the STATUS 16-bit word associated with each event

- Several of these are tuned to filter out "retriggering" events, which also flag real photons if the source is bright enough

- **Only** works if done during the initial `nupipeline` call (a.k.a., "Level 1" processing)

`nupipeline` processing

For bright sources - STATUS bits

Bit-wise (least significant bit on the right

STATUS[15] = Unused
STATUS[14] = Unused
STATUS[13] = Unused
STATUS[12] = PI out of range
STATUS[11] = Fails reset cut
STATUS[10] = Fails prior cut
STATUS[9] = Fails prior/reset cut (FPMA only)
STATUS[8] = Fails baseline cut
STATUS[7] = Fails depth Cut
STATUS[6] = Neighbor of hot/flickering pixel
STATUS[5] = Hot/flickering pixel
STATUS[4] = Edge pixel
STATUS[3] = Neighbor of pixels below
STATUS[2] = User bad pixel
STATUS[1] = Bad pixel from on-board disabled pixel
STATUS[0] = Bad pixel from CALDB file

IDL and python swap the order, so that indexed version:

STATUS[0] = Unused
STATUS[1] = Unused
STATUS[2] = Unused
STATUS[3] = PI out of range
STATUS[4] = Fails reset cut
STATUS[5] = Fails prior cut
STATUS[6] = Fails prior/reset cut (FPMA only)
STATUS[7] = Fails baseline cut
STATUS[8] = Fails depth Cut
STATUS[9] = Neighbor of hot/flickering pixel
STATUS[10] = Hot/flickering pixel
STATUS[11] = Edge pixel
STATUS[12] = Neighbor of pixels below
STATUS[13] = User bad pixel
STATUS[14] = Bad pixel from on-board disabled pixel
STATUS[15] = Bad pixel from CALDB file

“Standard” Processing (x = ignore)
“STATUS==b000000000x0xx000"

“Fast” Processing (x = ignore)
"STATUS==b0000xxx00xxxx000"

`nupipeline` processing

Only for the brightest, hardest sources

For bright sources - STATUS bits

Bit-wise (least significant bit on the right

STATUS[15] = Unused
STATUS[14] = Unused
STATUS[13] = Unused
STATUS[12] = PI out of range
STATUS[11] = Fails reset cut
STATUS[10] = Fails prior cut
STATUS[9] = Fails prior/reset cut (FPMA only)
STATUS[8] = Fails baseline cut
STATUS[7] = Fails depth Cut
STATUS[6] = Neighbor of hot/flickering pixel
STATUS[5] = Hot/flickering pixel
STATUS[4] = Edge pixel
STATUS[3] = Neighbor of pixels below
STATUS[2] = User bad pixel
STATUS[1] = Bad pixel from on-board disabled pixel
STATUS[0] = Bad pixel from CALDB file

IDL and python swap the order, so that indexed version:

STATUS[0] = Unused
STATUS[1] = Unused
STATUS[2] = Unused
STATUS[3] = PI out of range
STATUS[4] = Fails reset cut
STATUS[5] = Fails prior cut
STATUS[6] = Fails prior/reset cut (FPMA only)
STATUS[7] = Fails baseline cut
STATUS[8] = Fails depth Cut
STATUS[9] = Neighbor of hot/flickering pixel
STATUS[10] = Hot/flickering pixel
STATUS[11] = Edge pixel
STATUS[12] = Neighbor of pixels below
STATUS[13] = User bad pixel
STATUS[14] = Bad pixel from on-board disabled pixel
STATUS[15] = Bad pixel from CALDB file

“Standard” Processing (x = ignore)
“STATUS==b000000000x0xx000"

“Fast” Processing (x = ignore)
"STATUS==b0000xxx00xxxx000"

Part 1 Summary

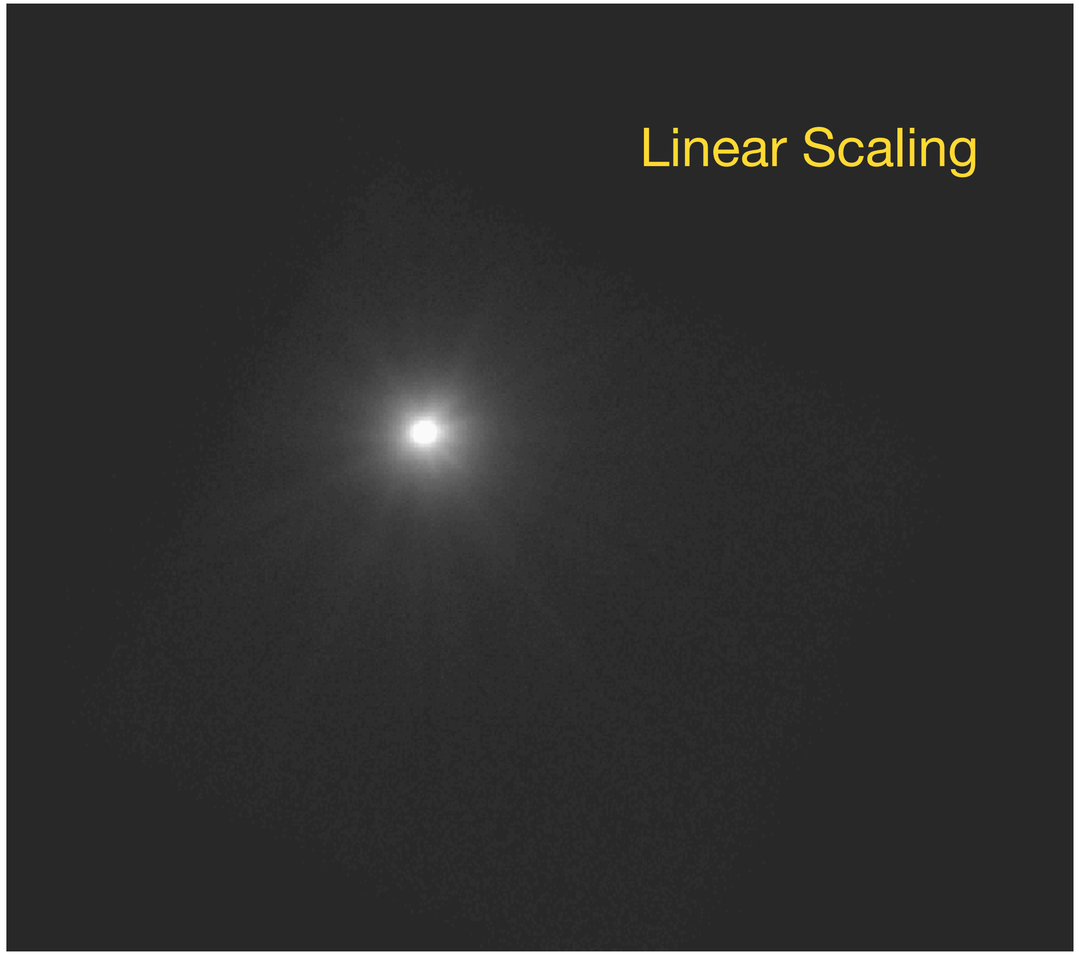
Please read the FAQs on HEASARC

- Use the correct STATUS bit for your analysis (and report what you did in your papers)
  - Remember, “default” values tuned for faint sources, not for bright sources!!
- If your source is >50 cps, should probably use the bright source filter
- Additional “black belt” analyses for turning off the depth cut and the shield veto
  - Not covered here…

Part 2: The bright source IMAGING problem

#toomanyphotons

- You’ve got a bright source



The problem

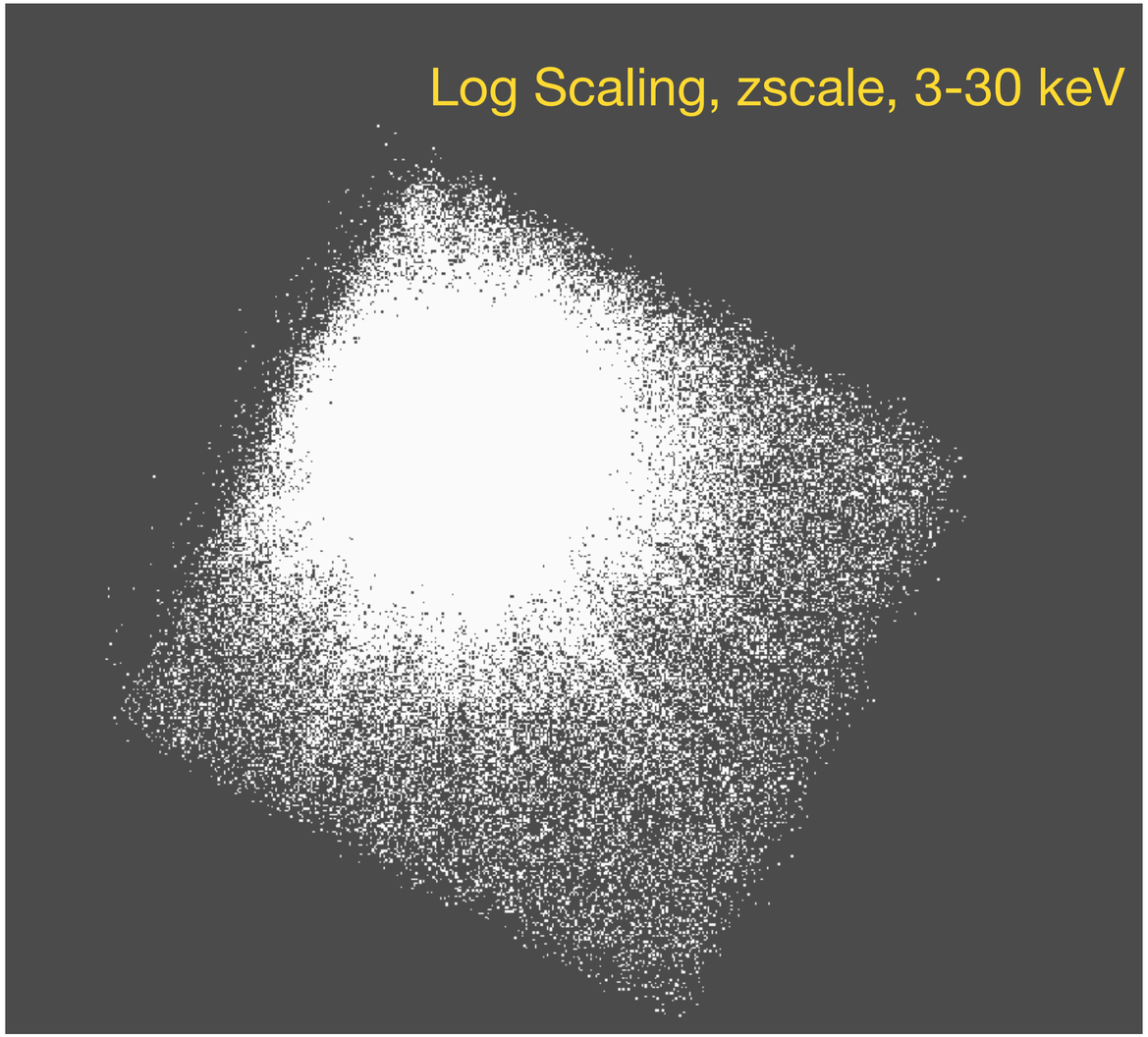
#toomanyphotons

- You’ve got a bright source
- It dominates DET0, what now?

The problem

#toomanyphotons

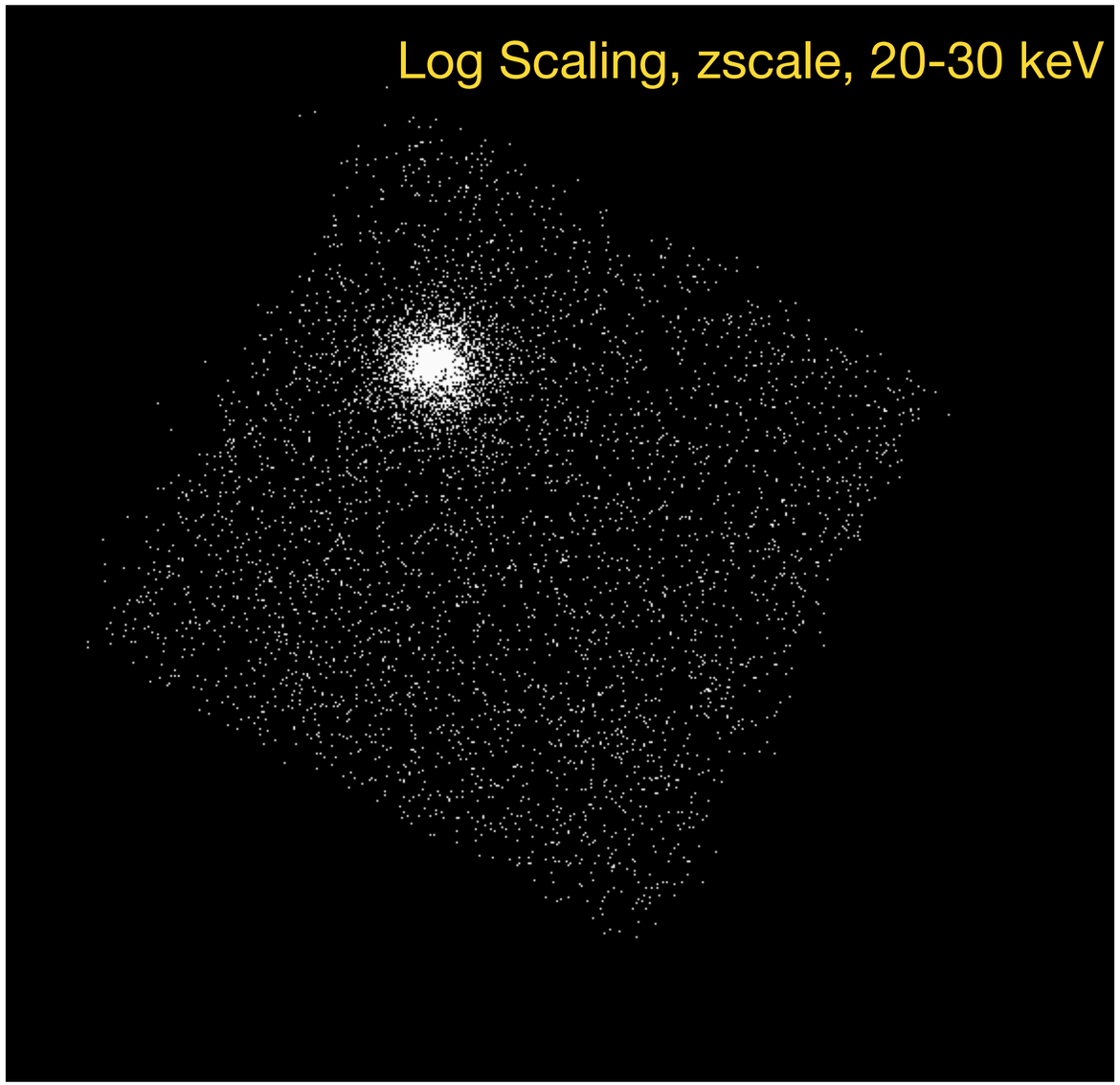
- You’ve got a bright source
- It dominates DET0, what now?
- It actually dominates most of the FoV (at 3-30 keV)



The problem

#toomanyphotons

- You’ve got a bright source
- It dominates DET0, what now?
- It actually dominates most of the FoV (at 2-30 keV)
- But not at 20-30 keV, where the background matters



Solution(s)

What now?

- Option 1: Use a background region somewhere else and subtract the background

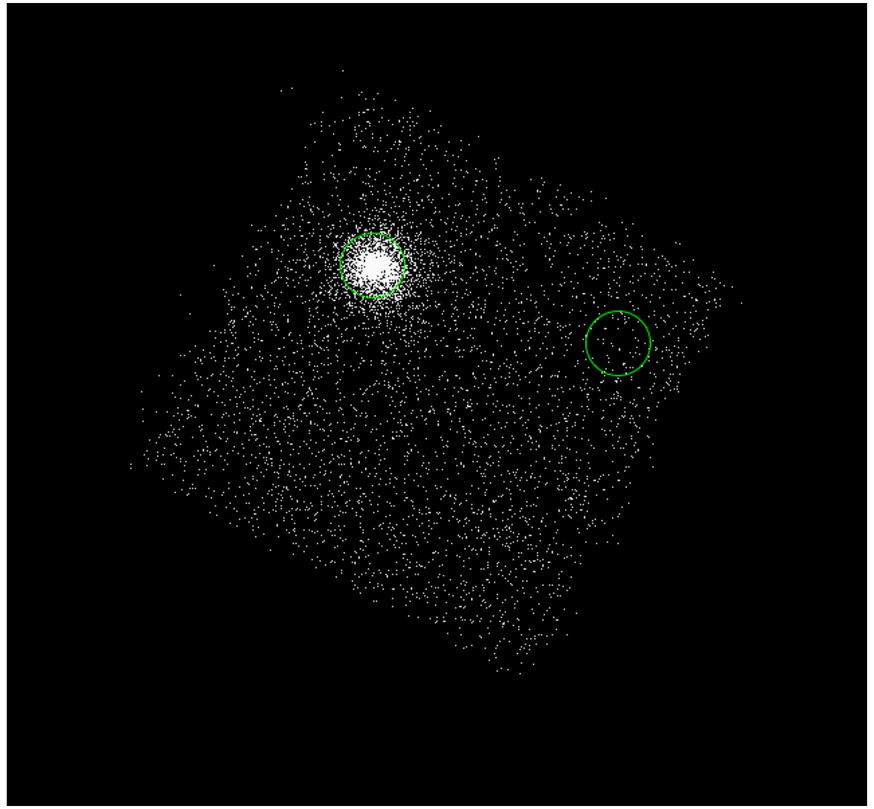
- Option 2: Use a background **spectral** model and the data in the source region

- Option 3: Use a background **spatial-spectral** model, the data in the source region, and the data in one or more background regions somewhere else (a.k.a., `nuskybgd`)

Option 1 Example

Background subtraction

- Pros:
  - This is easy to do
  - You can tune this to avoid contamination from the source (maybe)
  - `nuproducts` automagically scales the background by the relative sizes of the region
    - (I used the same size here, you don't have to)
- Cons:
  - Poor statistical sampling in background region
  - Not necessarily the same background as in your source region (because different CZT detectors, different parts of the background patterns)

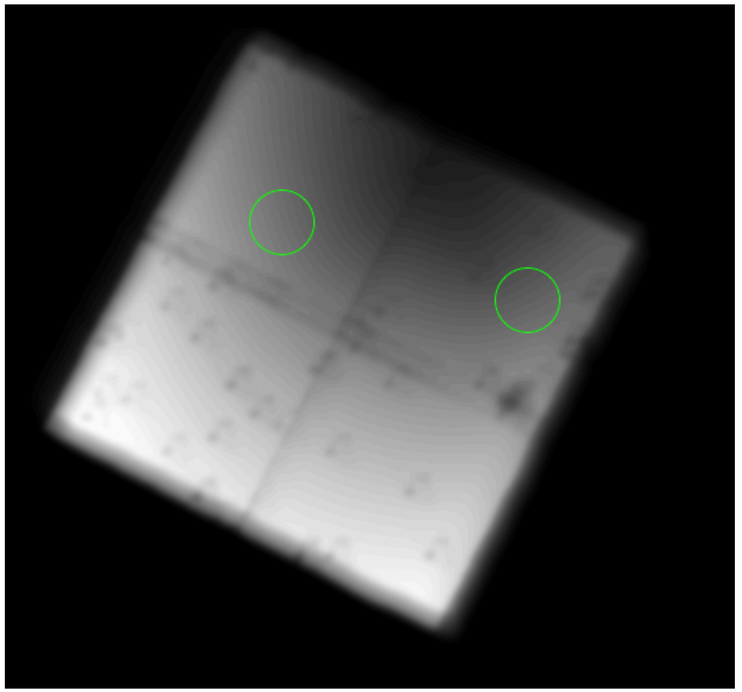


Option 1 Example

Background subtraction

- Pros:
  - This is easy to do
  - You can tune this to avoid contamination from the source (maybe)
  - `nuproducts` automagically scales the background by the relative sizes of the region
    - (I used the same size here, you don't have to)
- Cons:
  - Poor statistical sampling in background region
  - Not necessarily the same background as in your source region (because different CZT detectors, different parts of the background patterns)

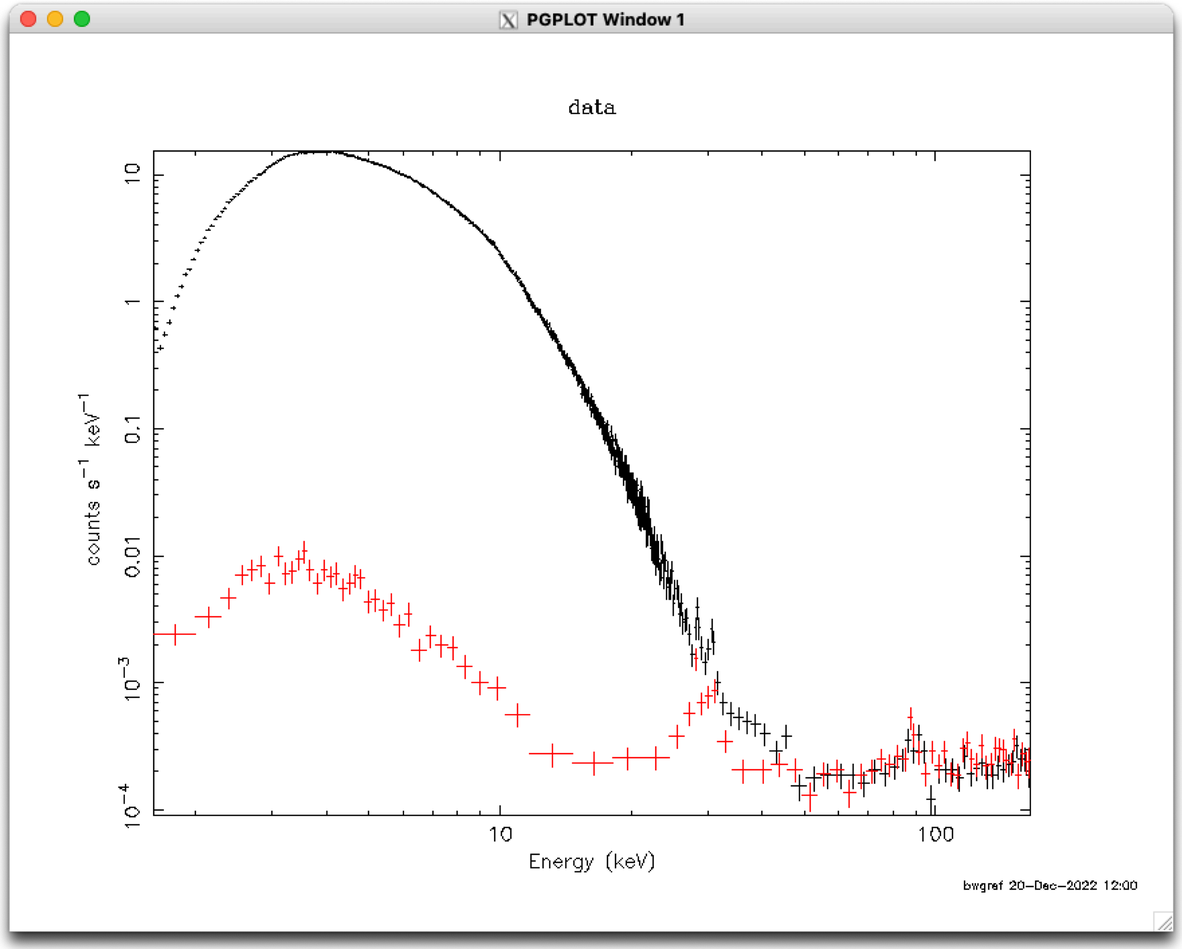
Aperture CXB model spatial variations



Option 1 Example

Background subtraction

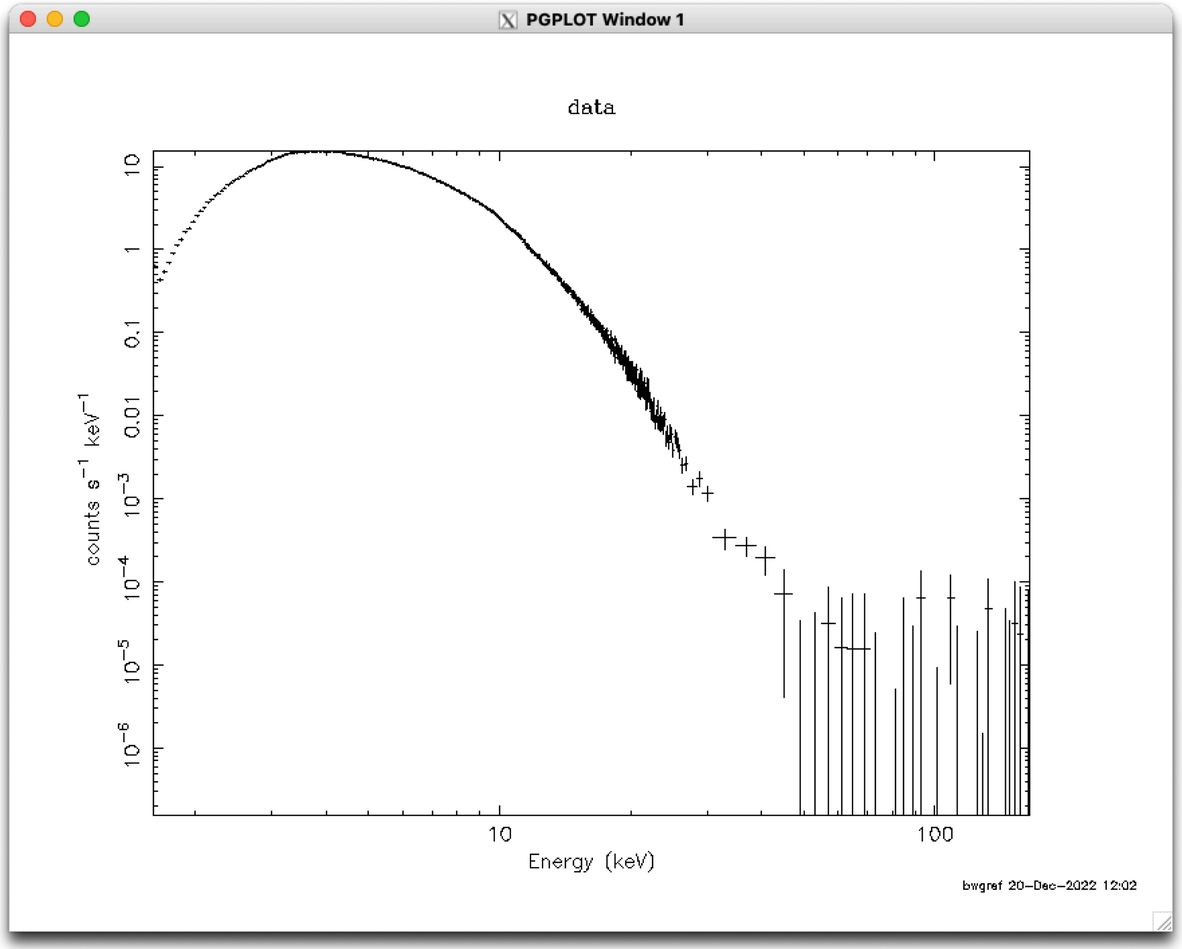
- Black is spectrum from source region
- Red is spectrum from background region loaded in separately
  - NB: They only match because I used the same size region
- Good match above ~50 keV



Option 1 Example

Background subtraction

- Background-subtracted spectra look pretty good
- Easy to run with this, except hard to constrain the hard tail because of low statistics
- Can get some artifacts near CsI lines near 30 keV region

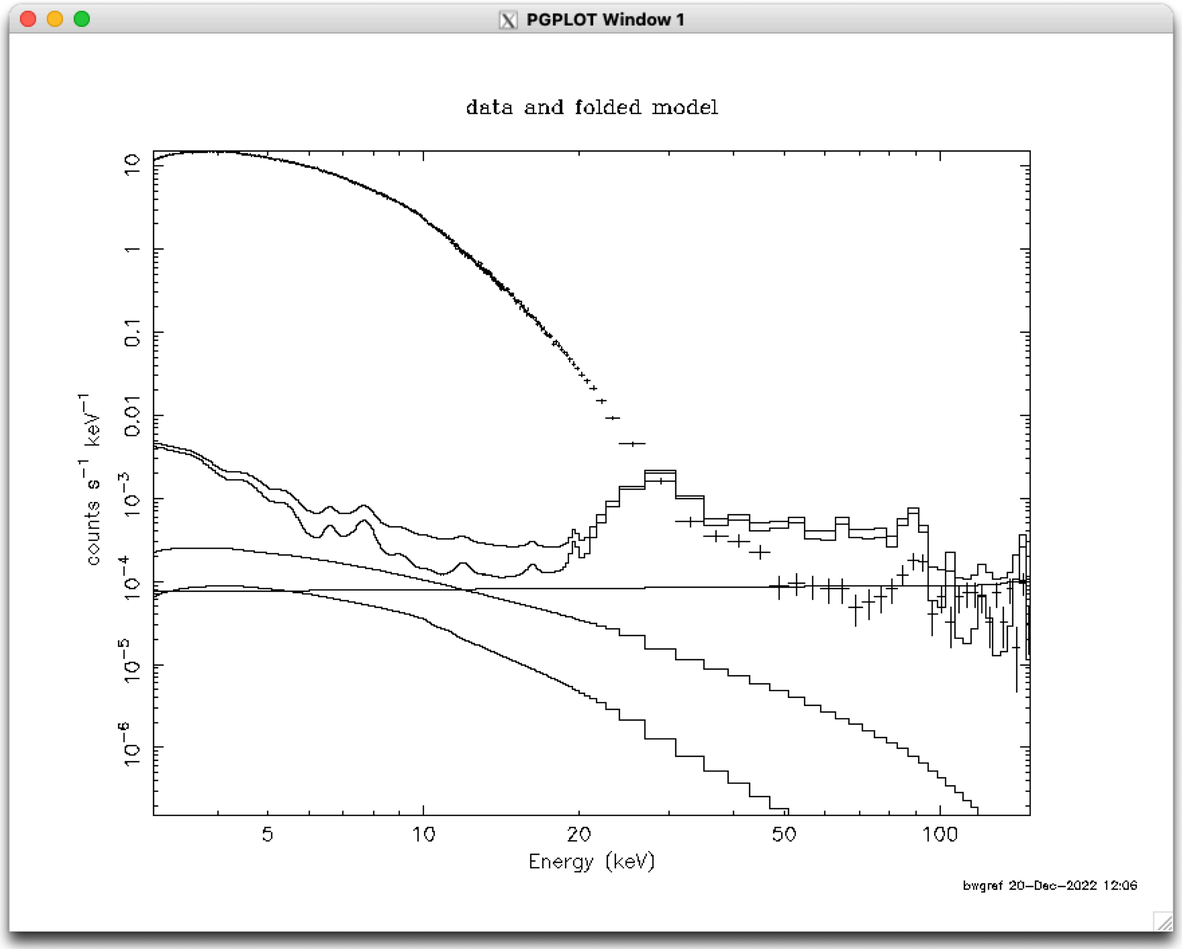


Option 2 Example

Background Fitting

- Pros:
  - Better statistical treatment
  - We know what the spectral shapes are supposed to be and how it varies in time, so can make educated choices about the line ratios
  - Avoids any possible self-subtraction of the source across the full bandpass
- Cons:
  - Have to install nuskybgd-py and pyXspec
    - Steep learning curve
  - Have to iteratively fit the background

Default nuskybgd model



Option 2 Example

Background Fitting

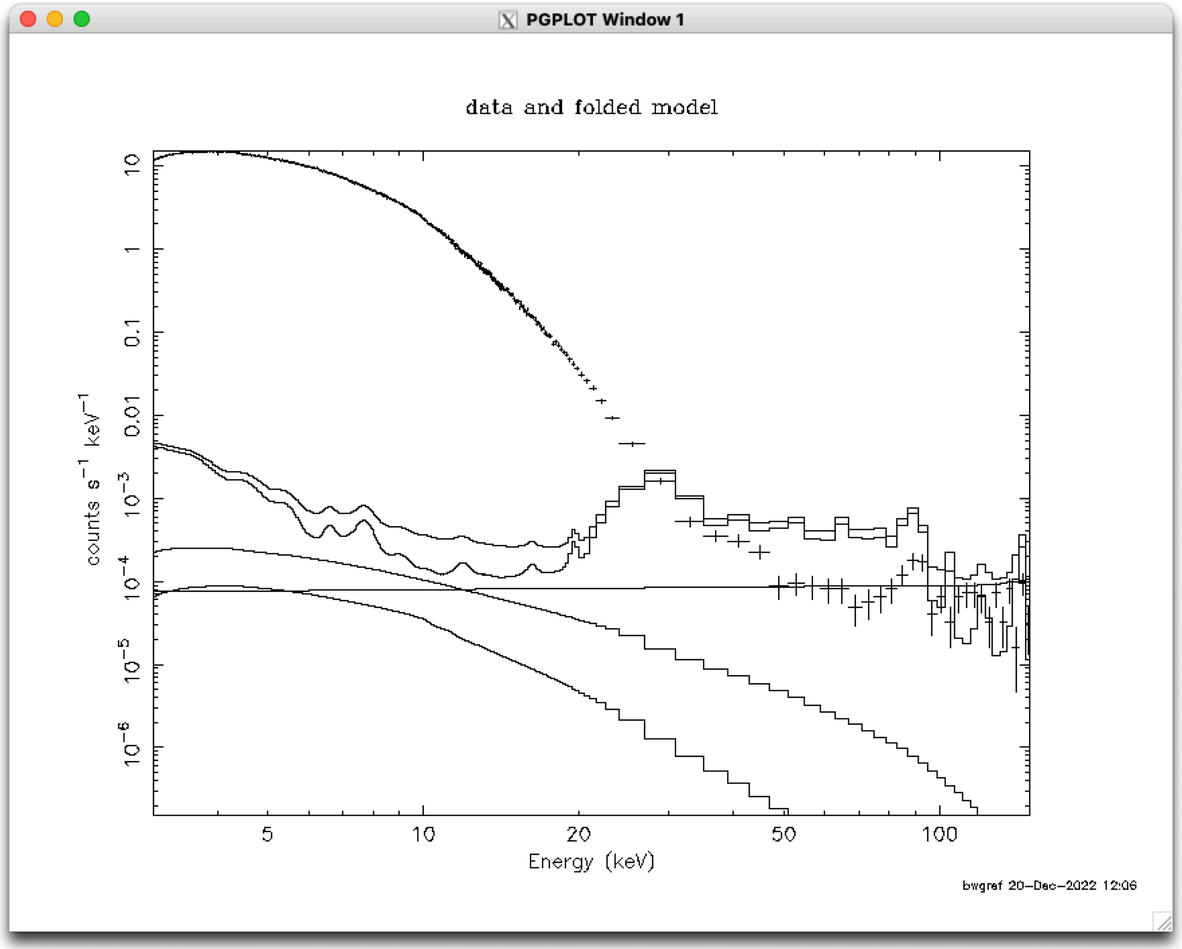
- Step 0: Run nuskybgd with just the source region (see nuskybgd-py walkthrough guide)
- Step 1: load background model and set correct statistics

```
@fpma_start.xcm
statistic cstat
parallel leven 8
parallel error 8

# Zero out solar components for now

newpar intbgd:4 0. -0.1
newpar intbgd:7 0. -0.1
newpar intbgd:10 0. -0.1
newpar intbgd:13 0. -0.1

freeze apbgd:3
freeze fcxb:3
freeze intbgd:16
```



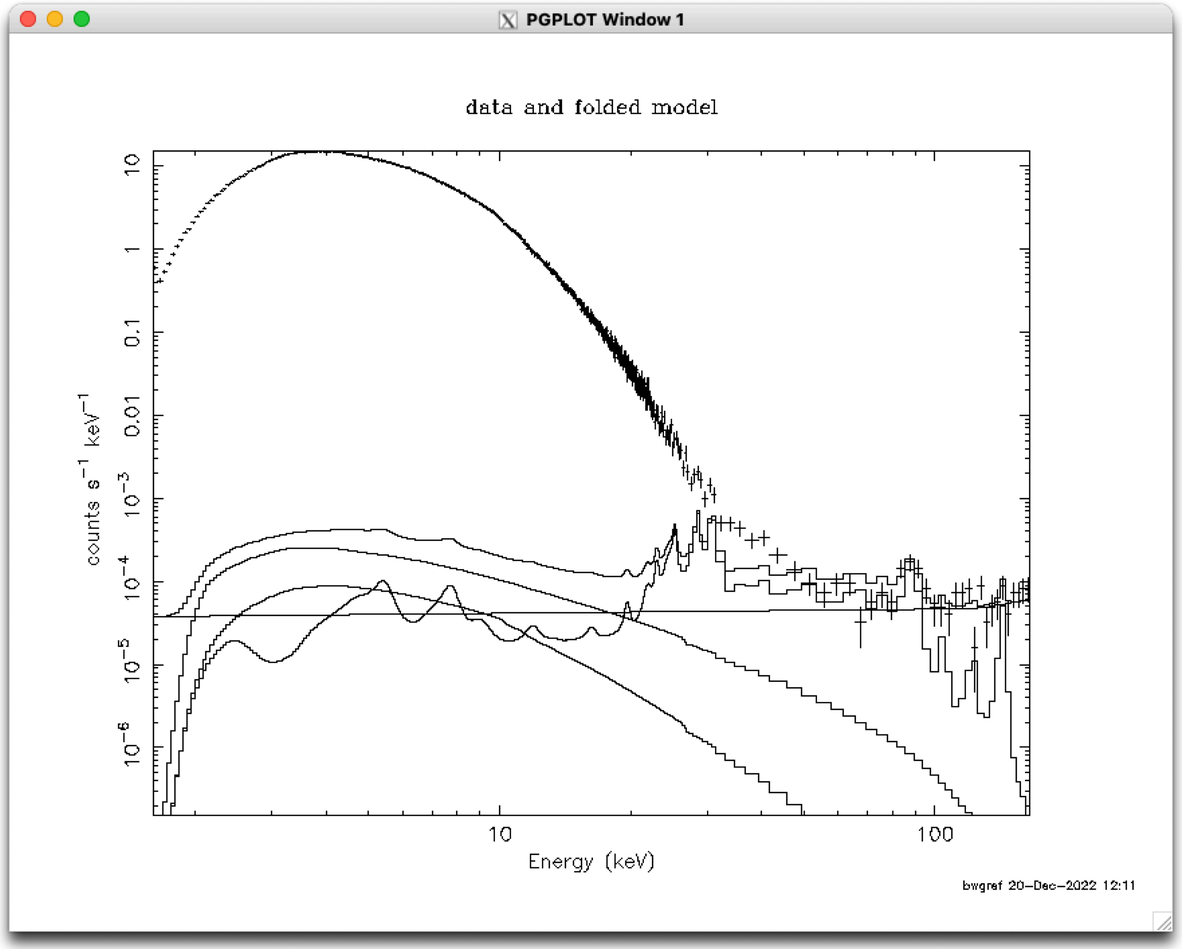
Option 2 Example

Background Fitting

- Step 2, fit continuum and line normalization

```
ign *:0.-105.
fit 1000
notice *:80.-105.


thaw intbgd:16
fit 1000
```

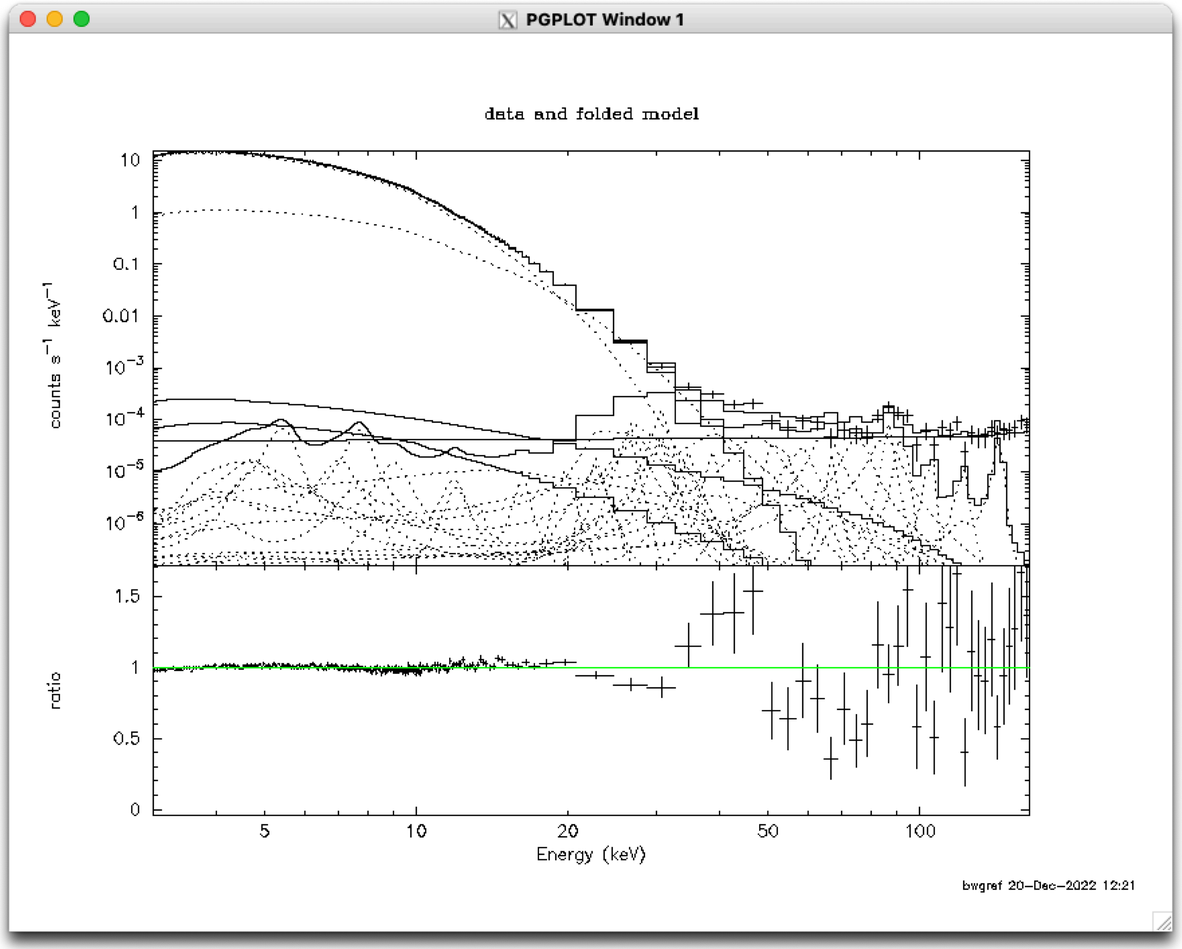


Option 2 Example

Background Fitting

- Step 3: Add source model and fit as per usual
- Pros:
  - Independent of spatial variations in background
  - Allows formal use of C-stat (not W-stat)
  - Don’t have to worry about poor sampling in background region

Two compTT models

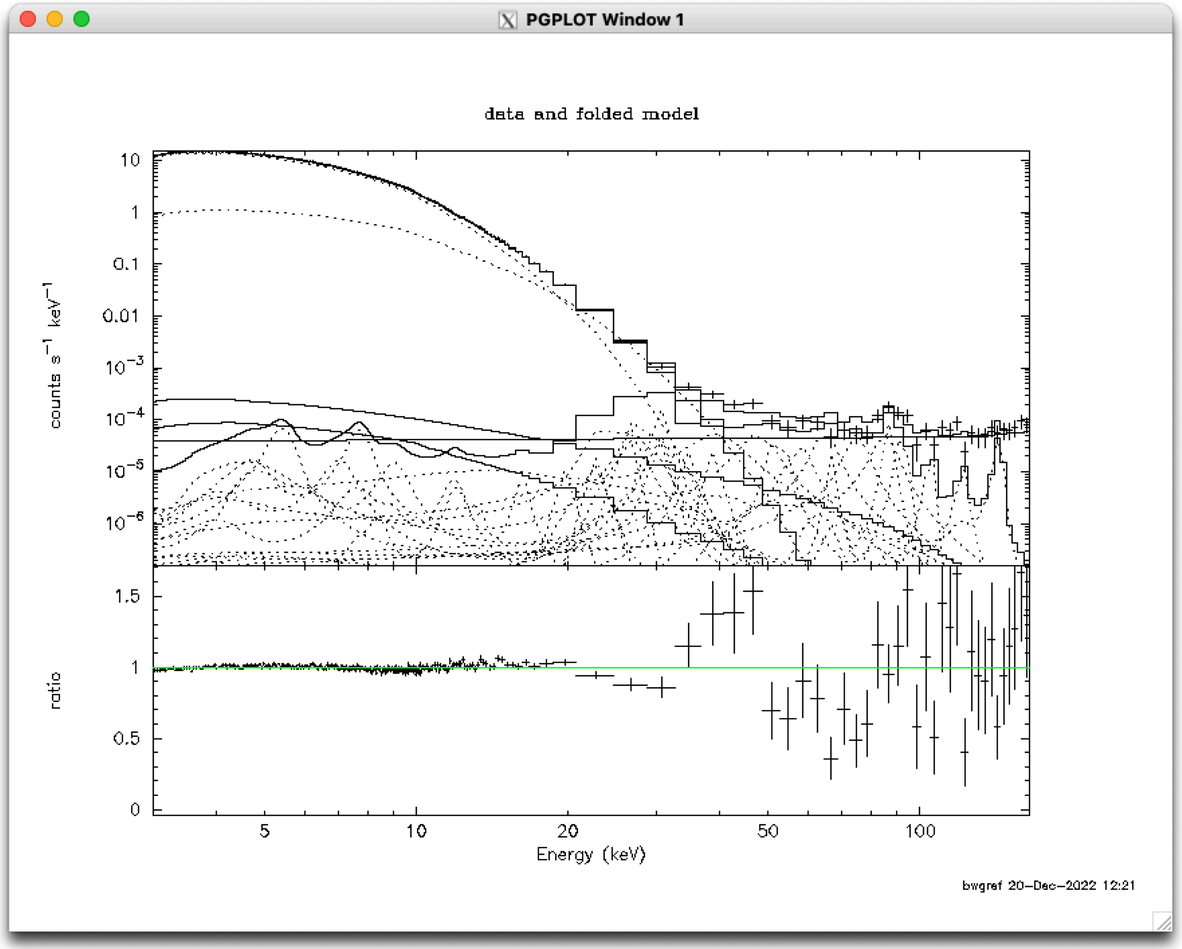


Option 2 Example

Background Fitting

- Cons:
- Shape of CXB spectrum may be degenerate with source spectrum
  - Line normalization component usually degenerate with source model around ~25 keV
  - Not (generally) possible to keep background parameters free
- C-stat / d.o.f. no longer appropriate check of **source** goodness of fit
  - Can be mitigated with binning strategies, but C-stat is now telling you about the goodness of the combined **source+background** fit
  - Fit shown has a C-stat / d.o.f. of 2634 / 4061 but is definitely not a good fit in the region where the source dominates
    - Mostly meaningless because of the number of bins above 50 keV where the background dominates.

Two compTT models

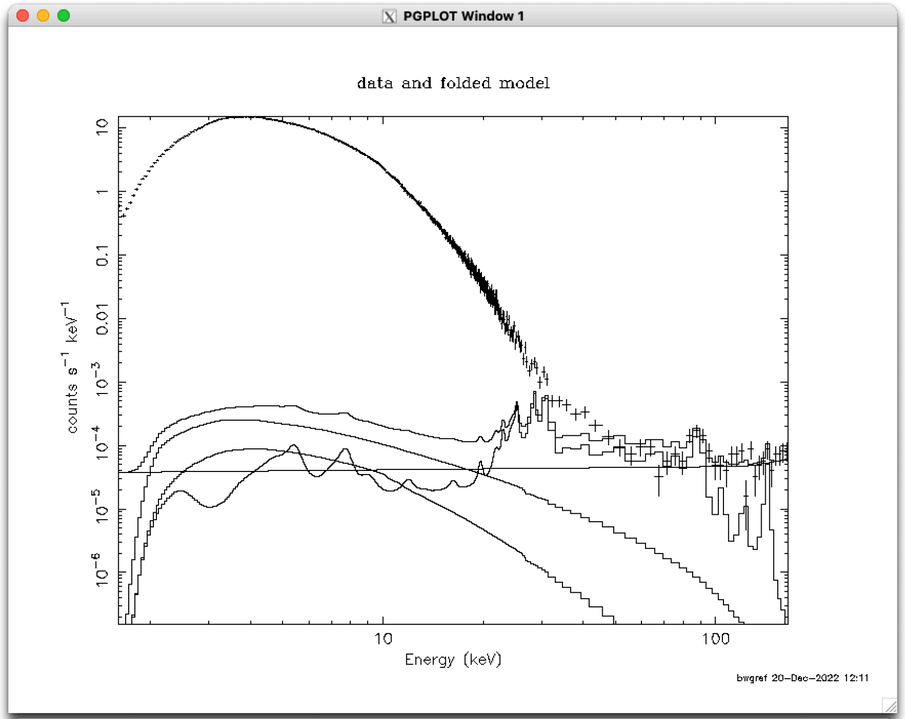


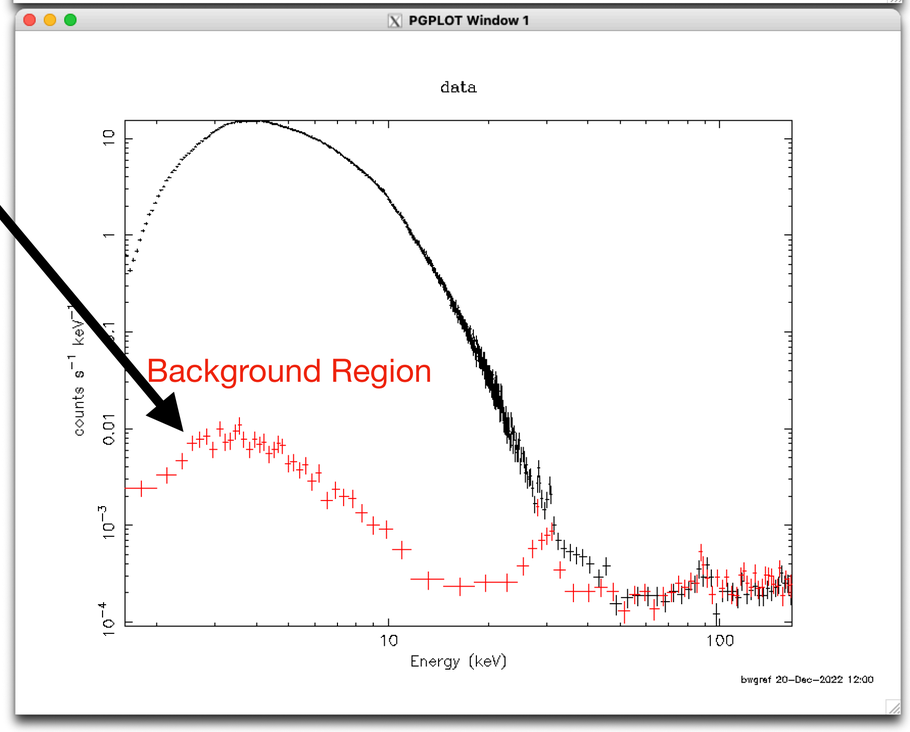
Option 2 Example

A warning for background subtraction

- Note **shape** of background model below 10 keV compared with “subtracted” background is very different

- Model expectation is ~1e-3 cts / sec / keV, background region is at ~0.01 cts / sec / keV, so 10x more counts in background region than expected

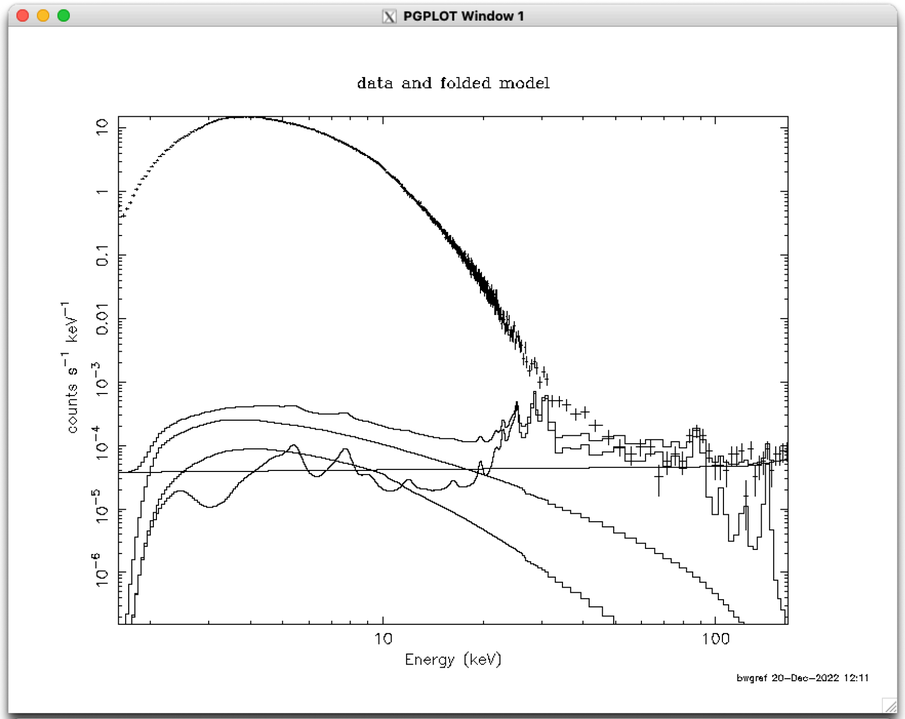


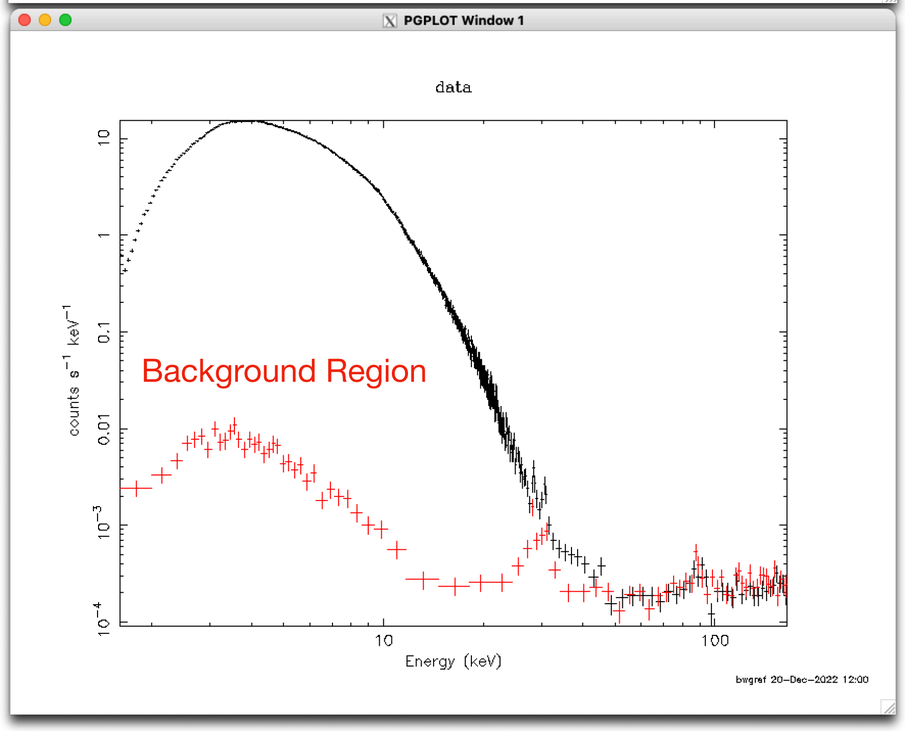


Option 2 Example

A warning for background subtraction

- Likely culprit is **source** photons in the **background** region
    - Source photons are coming from the extreme wings of the PSF, so will different effective area response than on-axis in source region
    - Can result in biasing background-subtracted shape at the ~few % level
- Could also be solar X-rays?
    - There was an M-class flare during this observation
    - You can see it in the nucalcsaa background report
    - In real life, filter out the solar flare before you begin



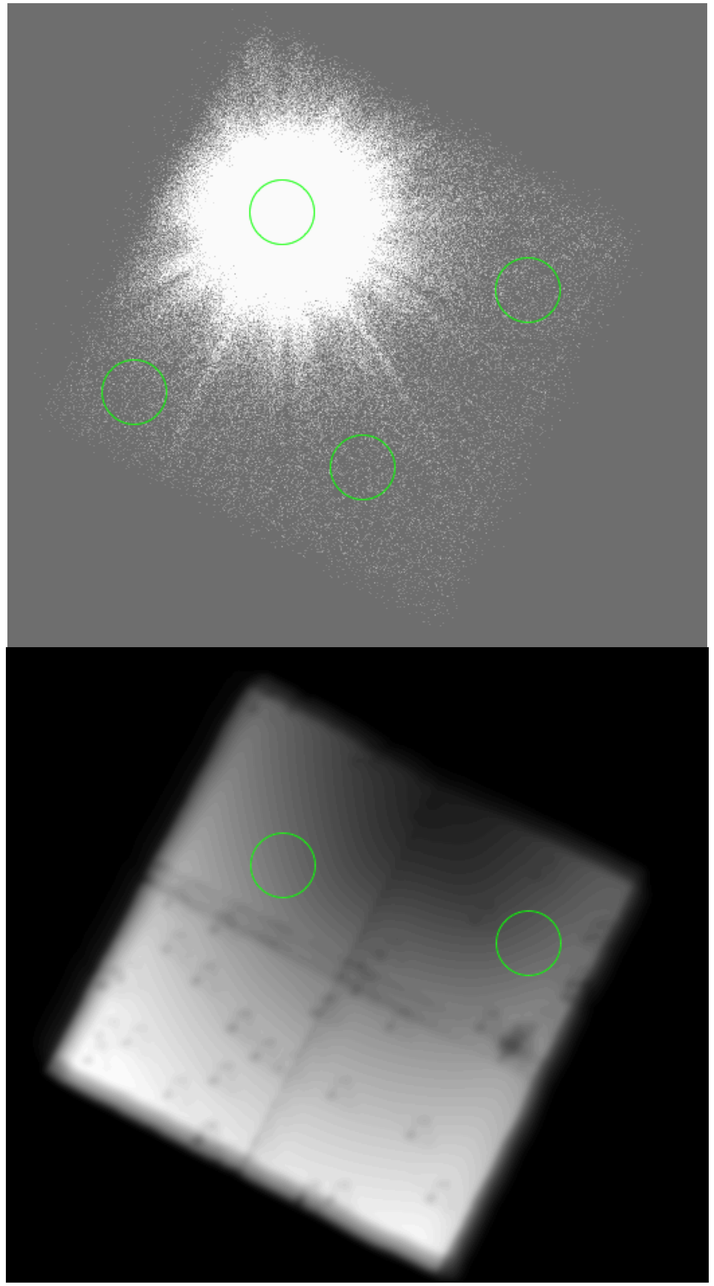


Option 3 Example

nuskybgd fitting

- Don’t bother in this case
- Used to constrain normalization of the aperture CXB spatial model
  - In this case, source is bright over the whole FoV below 20 keV
  - Probably no leverage over the background model…
- More useful in the faint-source case where you need to constrain the CXB normalization to avoid biasing the spectral fits

3-20 keV with source and background regions



Part 2 Summary

Wrapping up for the bright source

- In this case, background subtraction looks like it's good enough (probably?)
  - Useful because people know how to do it
  - Also useful of use of out-of-date statistical tools to determine goodness of fit

Part 2 Summary

Wrapping up for the bright source

- **However**
  - In background subtracted spectrum looks like source is only detected to ~40 keV
  - In the background model version it looks like there is source flux in 40-50 keV range

Part 2 Summary

Wrapping up for the bright source

- Easy to see that source is still present by eye in 40-50 keV image
  - **So, the background-subtracted spectrum is fooling us…**
  - Could be that internal background norm is different on the other detector (we know this happens), or just unlucky?
- Low-energy spectral shape in the background region is different from the model
  - May or may not be important
  - Your mileage may vary. Check, check, and double check before you start playing with detailed spectral models.

40-50 keV Image, Smoothed