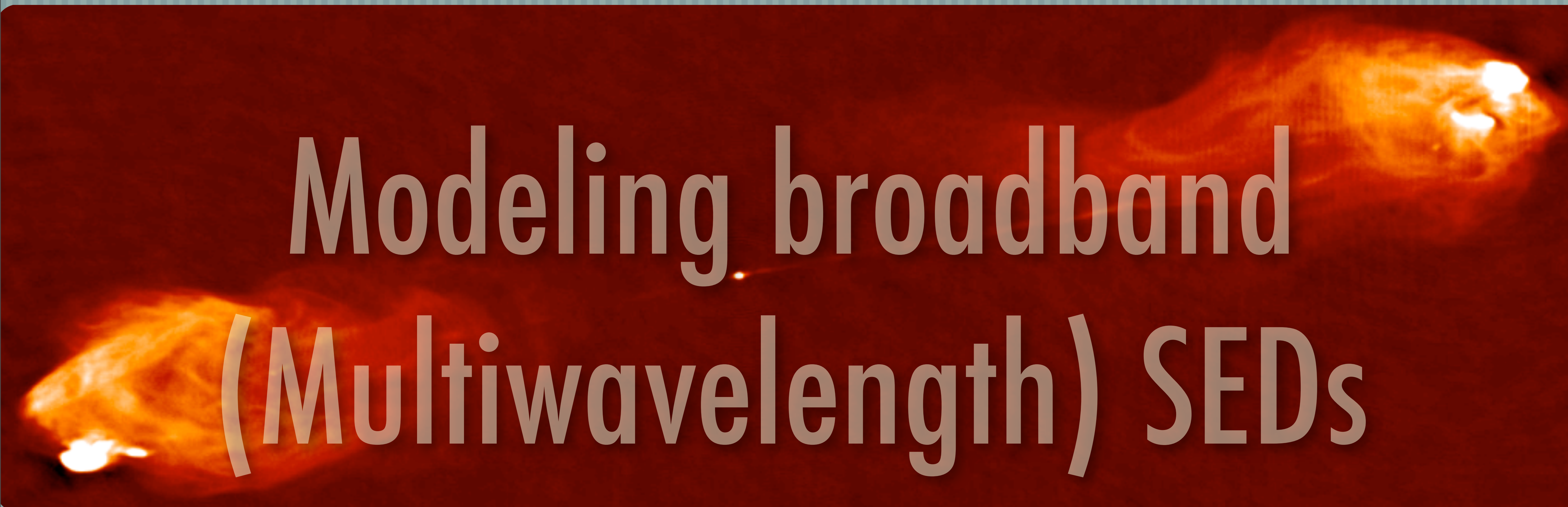
Modeling broadband (Multiwavelength) SEDs



ITN MW School, 9/7/10, A'dam

Rough outline of next few hours

- Short introduction: why we want to model MW SEDs
- What you need to do to combine X-ray and non X-ray data with Isis
- Hands on practice fitting MW SED with simple models, adding more physical components
- How to SLIRP customized models
- Hands on practice fitting MW SED with "slirped" model

Why do we want to do MW fitting?

Many astrophysical objects radiate over multiple wavebands

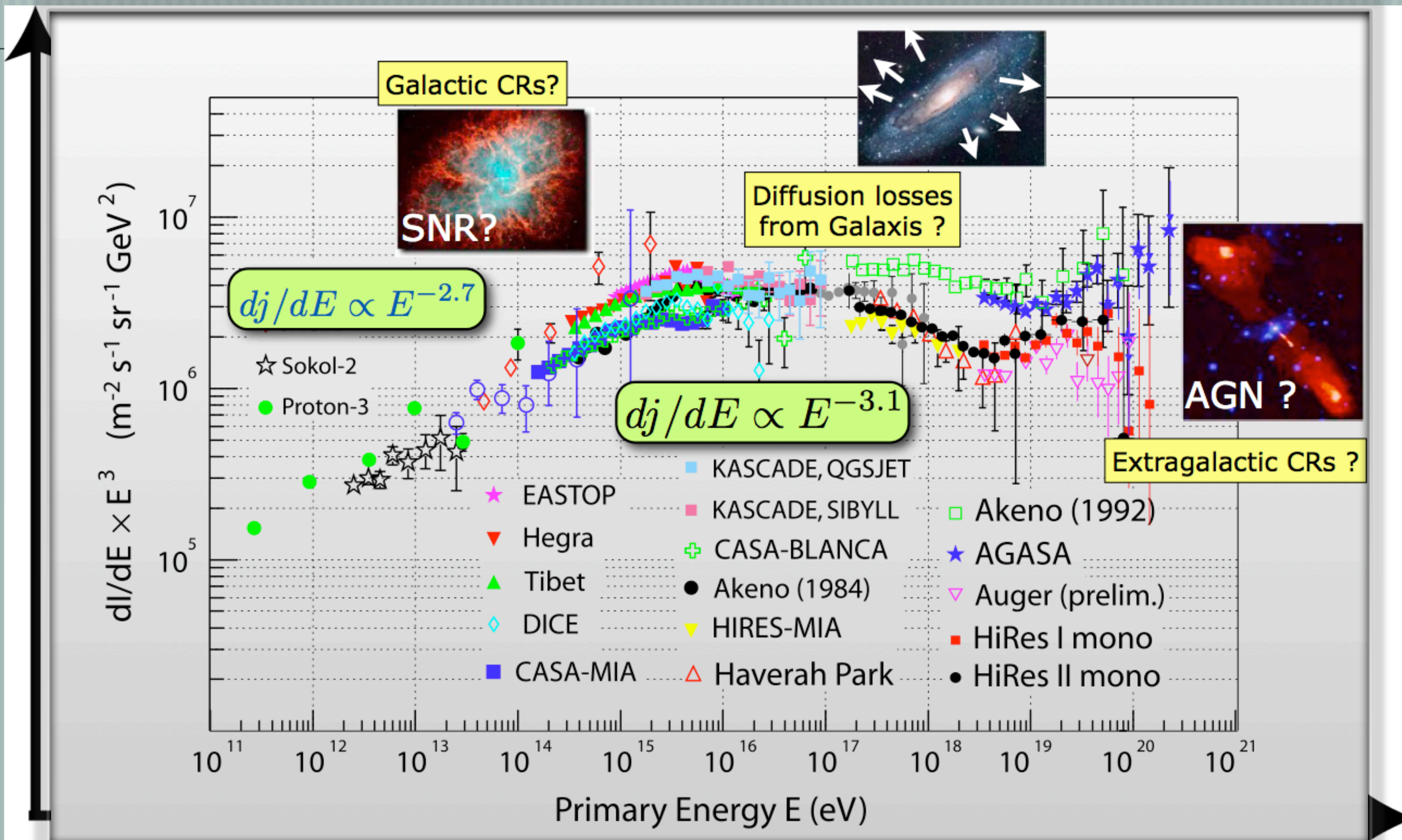
- Either via the same process, or interlinked processes (one band depends on another)
- To understand and constrain the source physics, we need to fit over the "broadband" spectrum, which can, in the most extreme case, extend from radio through the gamma-rays
- Mostly nonthermal because thermal processes, by definition, usually are associated with a single T or limited T range

Example: synchrotron radiation

Can extend over as many orders of magnitude as the underlying particle distribution, and be extended (in range) by source geometry

- Common process in HEA objects: particle acceleration
- Acceleration leads to power-law tails on particle distributions

Example: synchrotron radiation



Example: synchrotron radiation

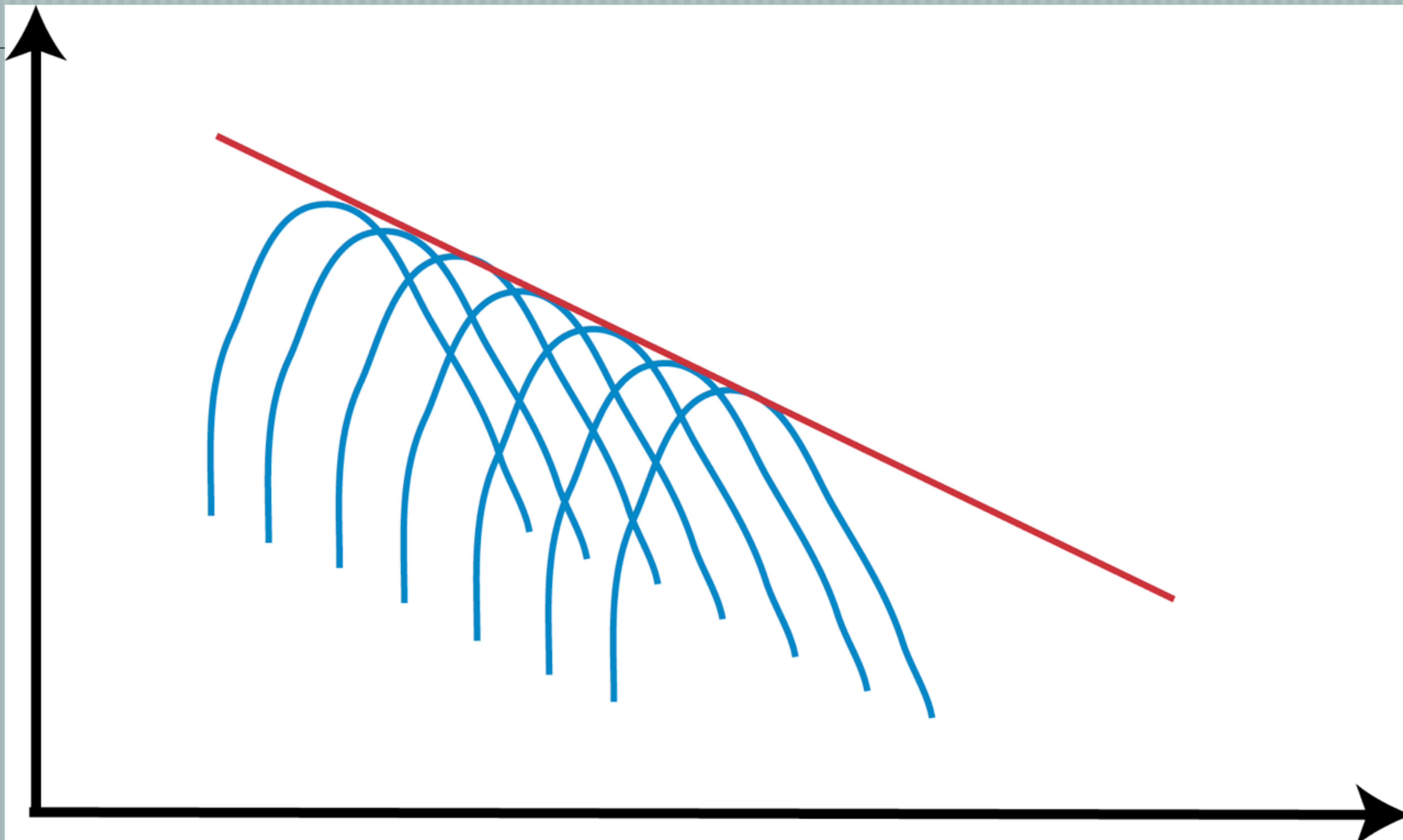
Can extend over as many orders of magnitude as the underlying particle distribution, and be extended (in range) by source geometry

* Common process in HEA objects: particle acceleration
* Acceleration leads to power-law tails on particle distributions
* Each electron radiates synchrotron emission that peaks around

$$\nu_{\mathrm{c}} = \frac{3}{2}\gamma\nu_B \sin\alpha$$

* Convolve this with a power-law distribution and you get a power-law distribution of radiation

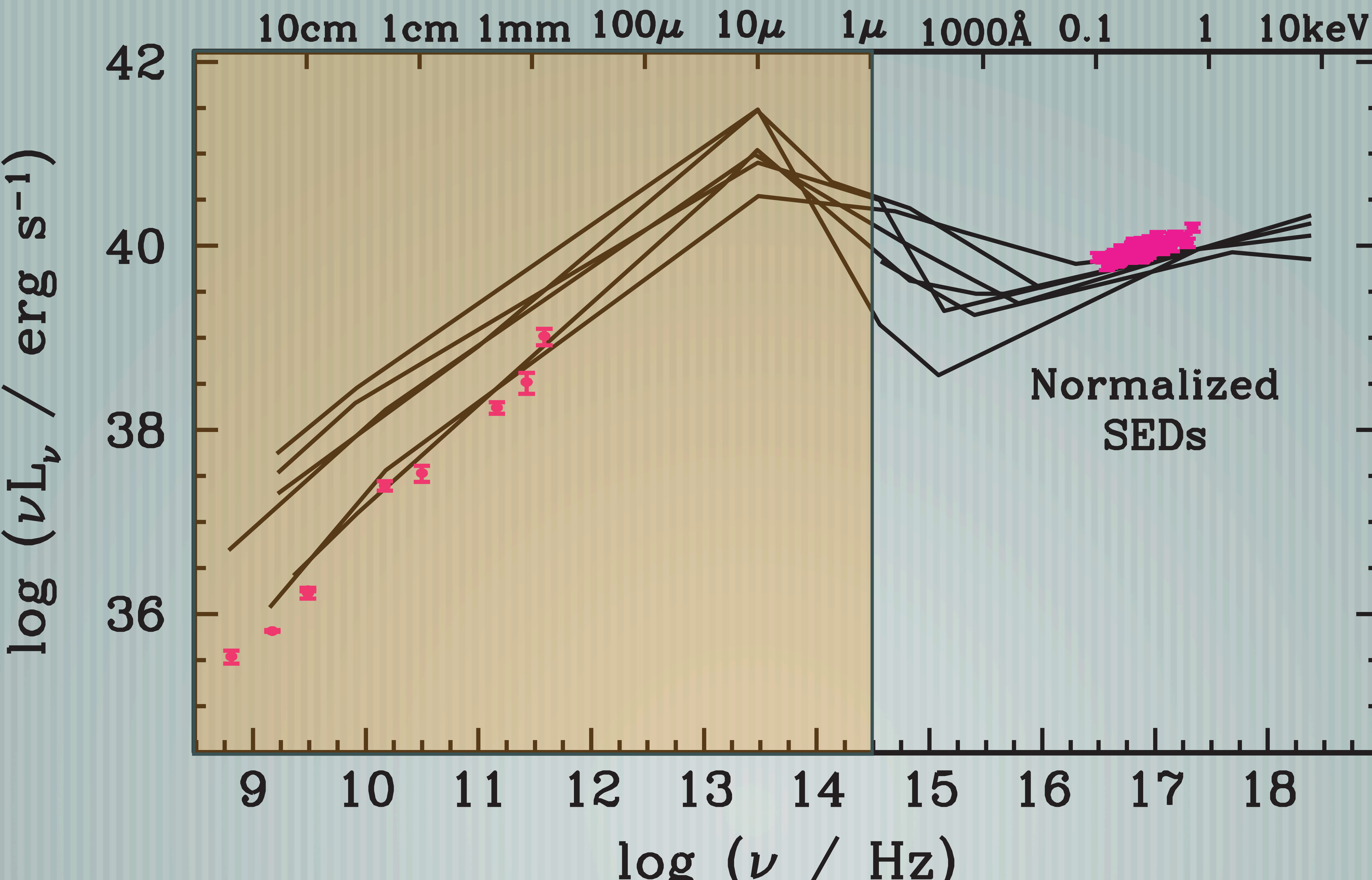
Example: synchrotron radiation



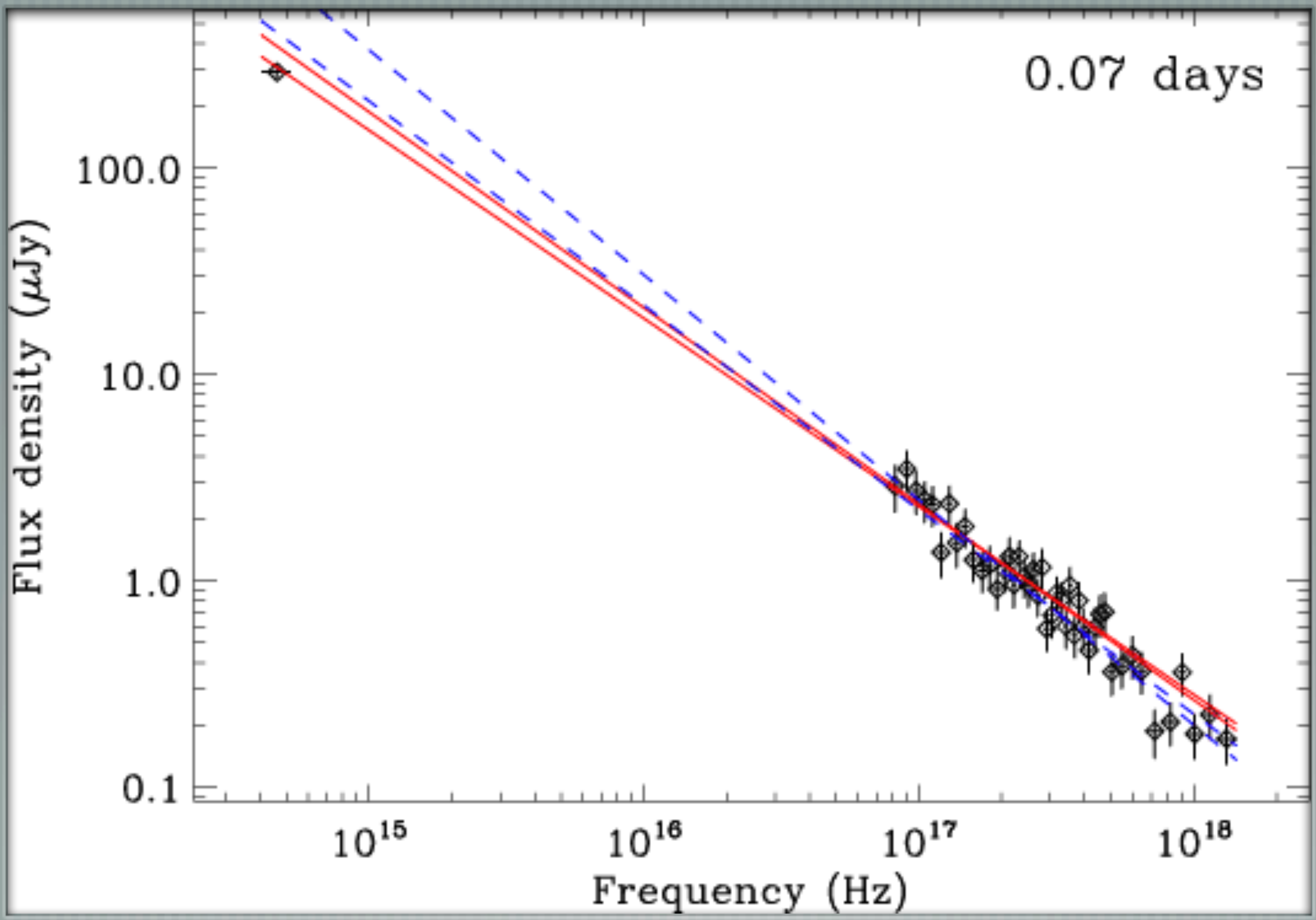
e)

w

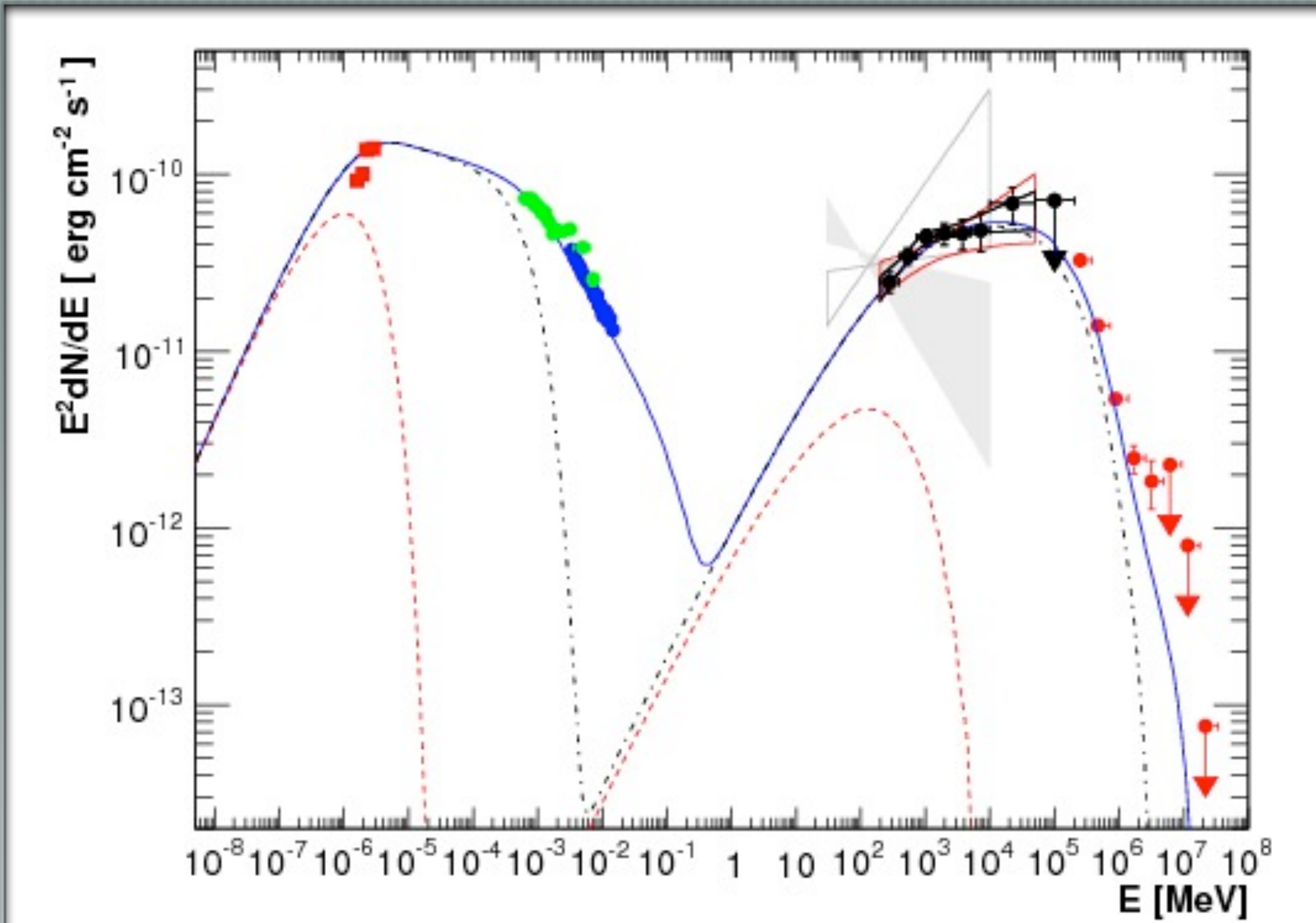
Synchrotron from LLAGN/LINERs:



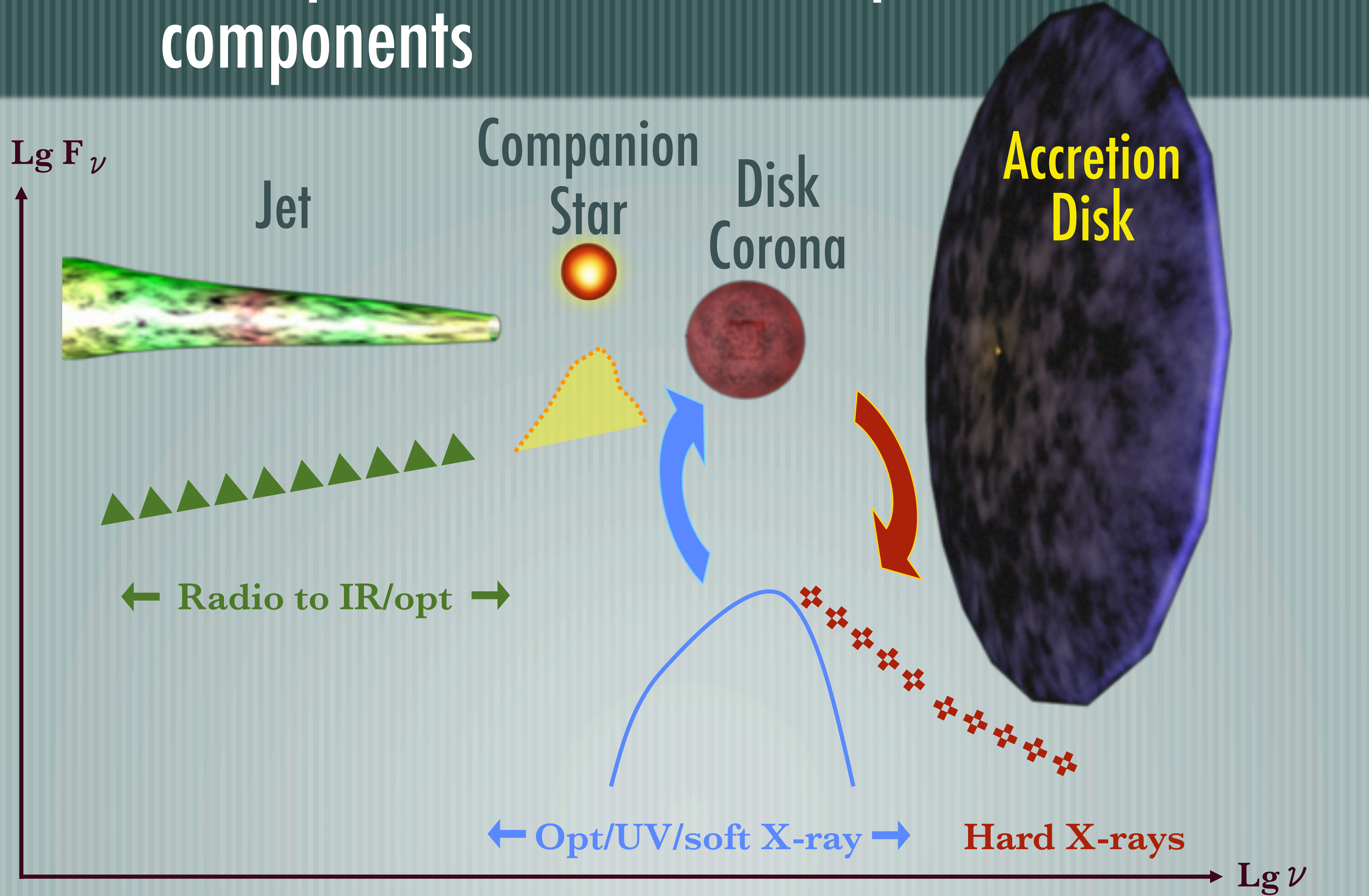
Synchrotron from GRB afterglow:



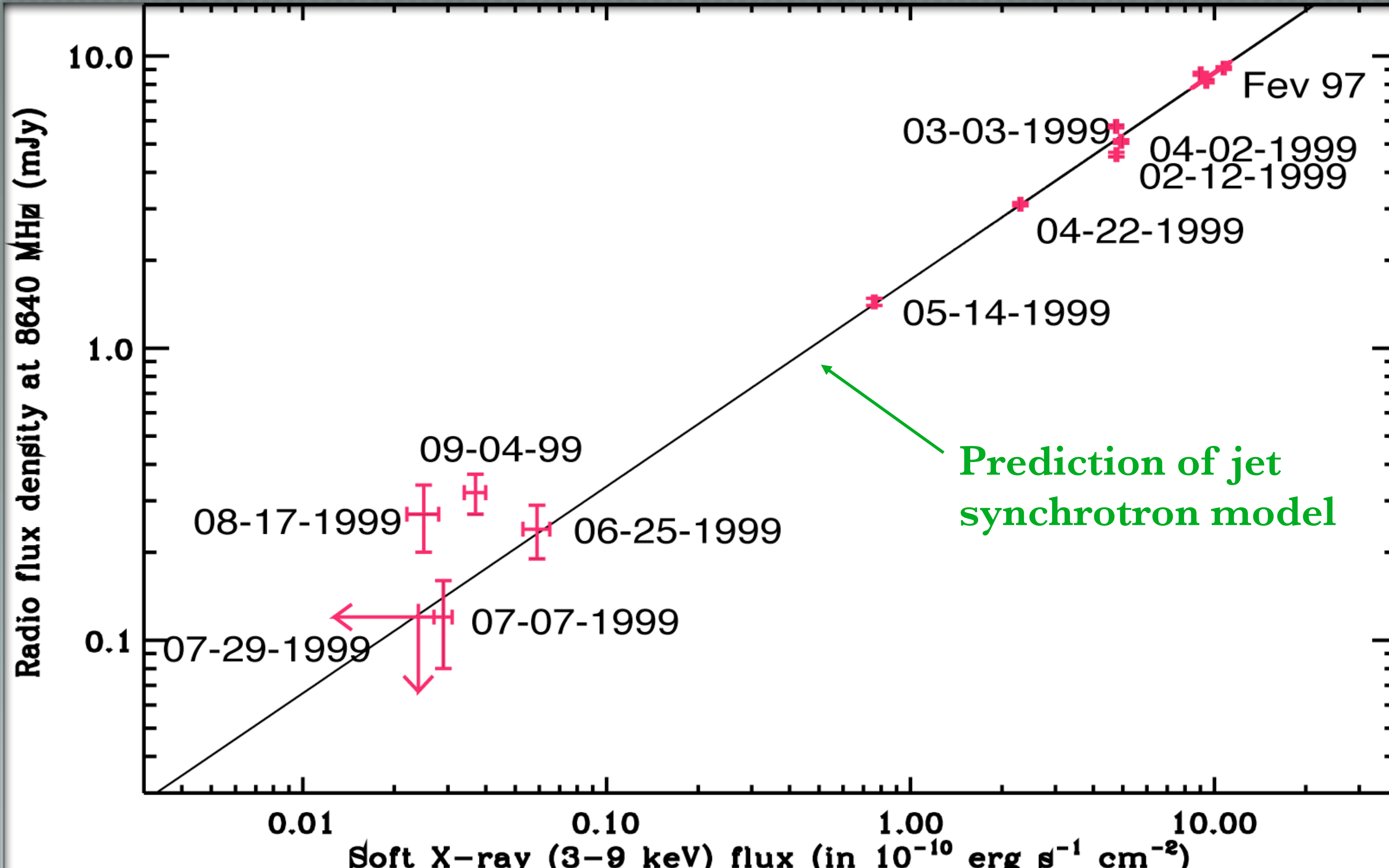
MW dependence: blazar synchrotron + SSC



Example: XRB Hard state spectral components

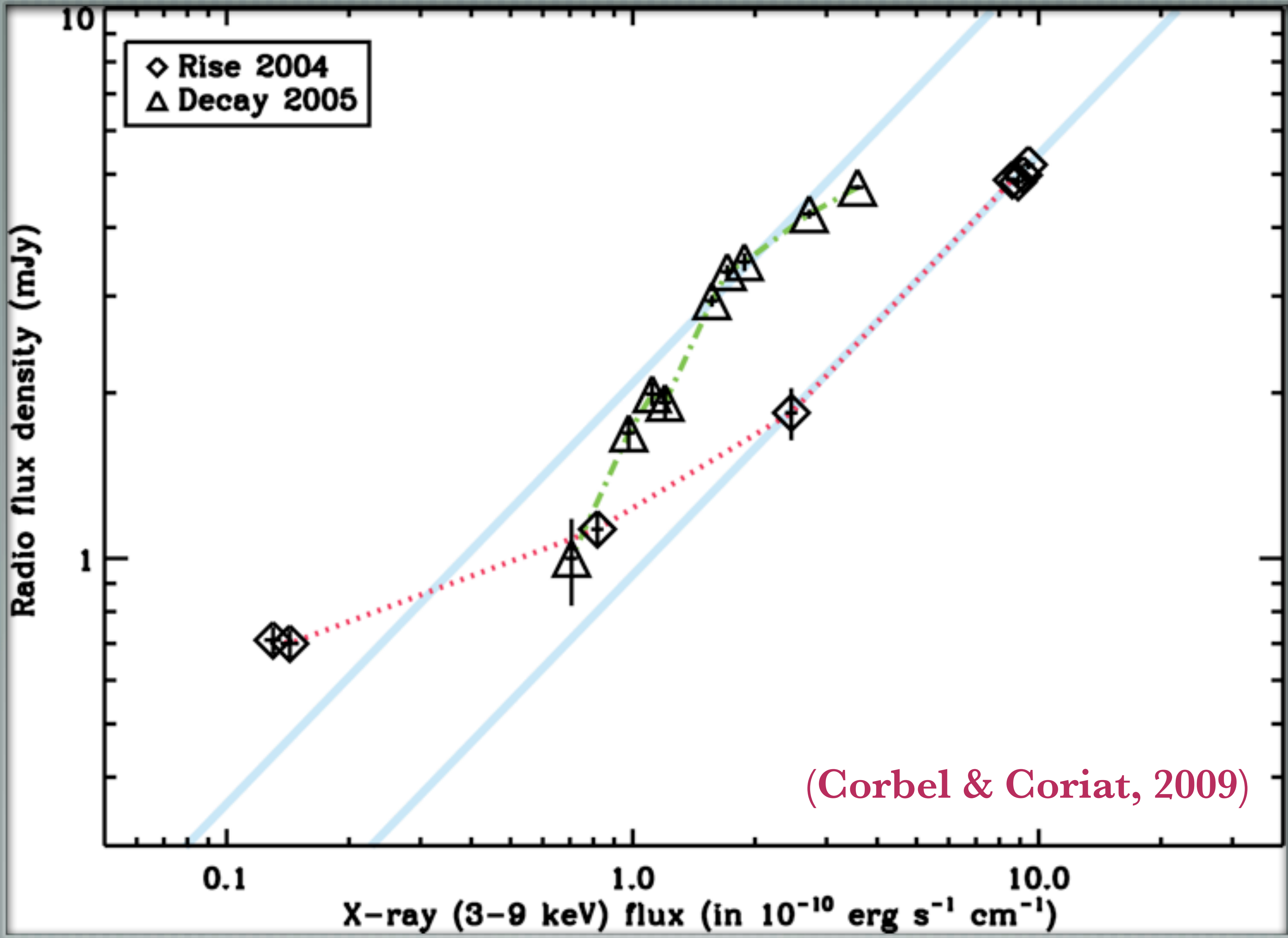


XRB hard state – Radio/Xray correlation



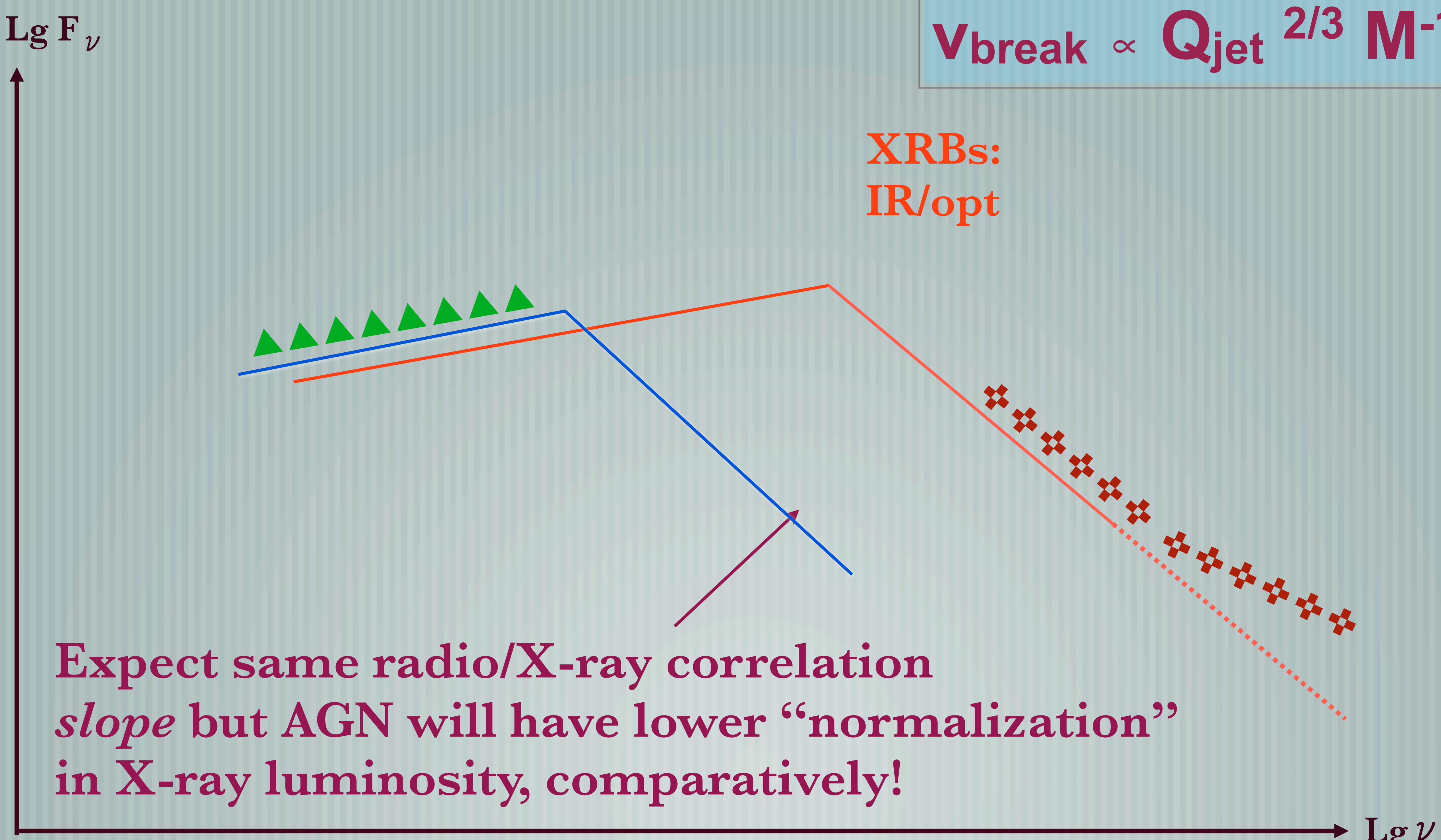
(Corbel ea. 2000, 2003; Markoff et. 2003, Gallo ea. 2003)

GX339-4: key source to study MW correlations, inflow/outflow connections



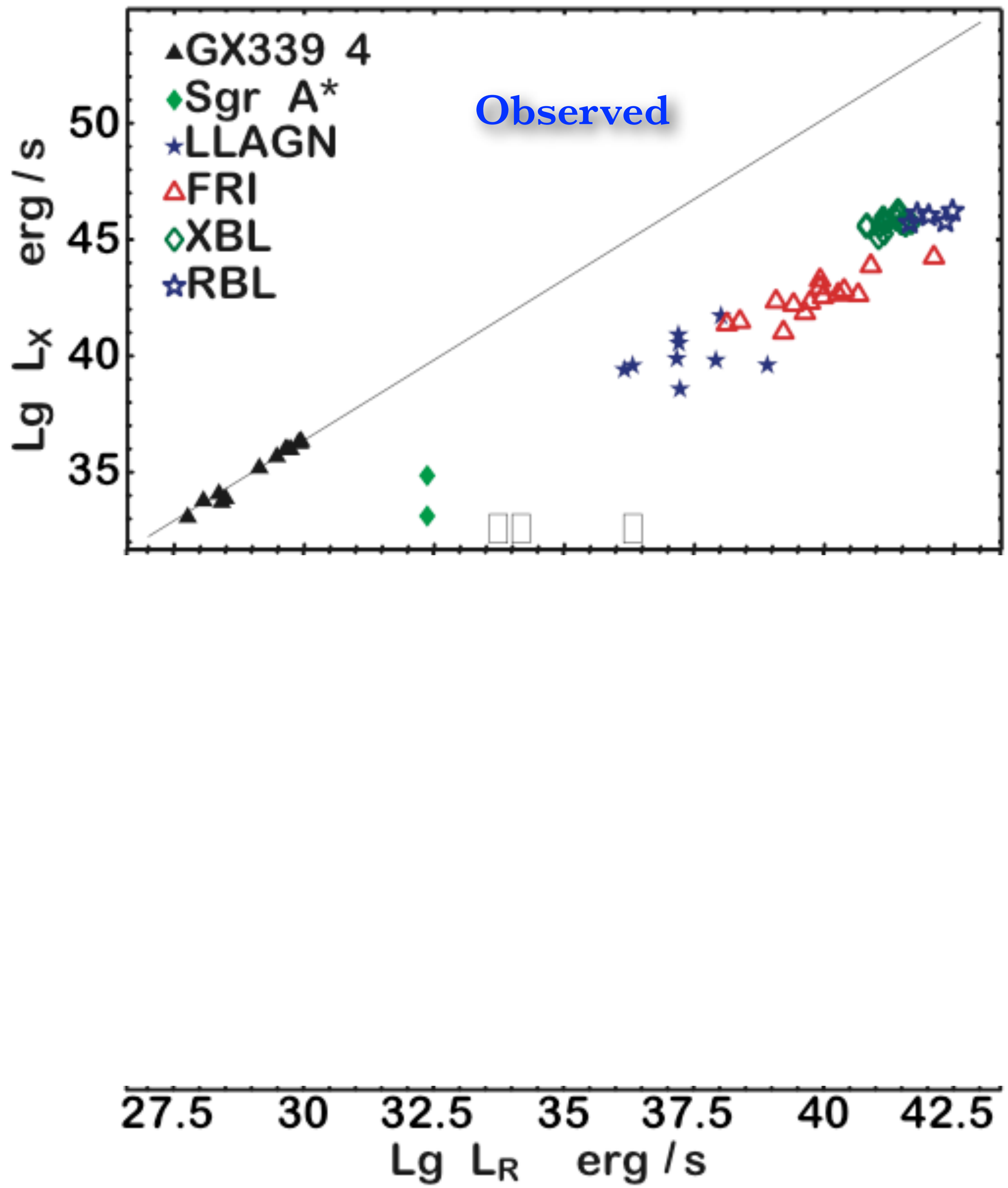
Mass scaling of jet break frequency

$\nu_{break} \propto Q_{jet}^{2/3} M^{-1}$



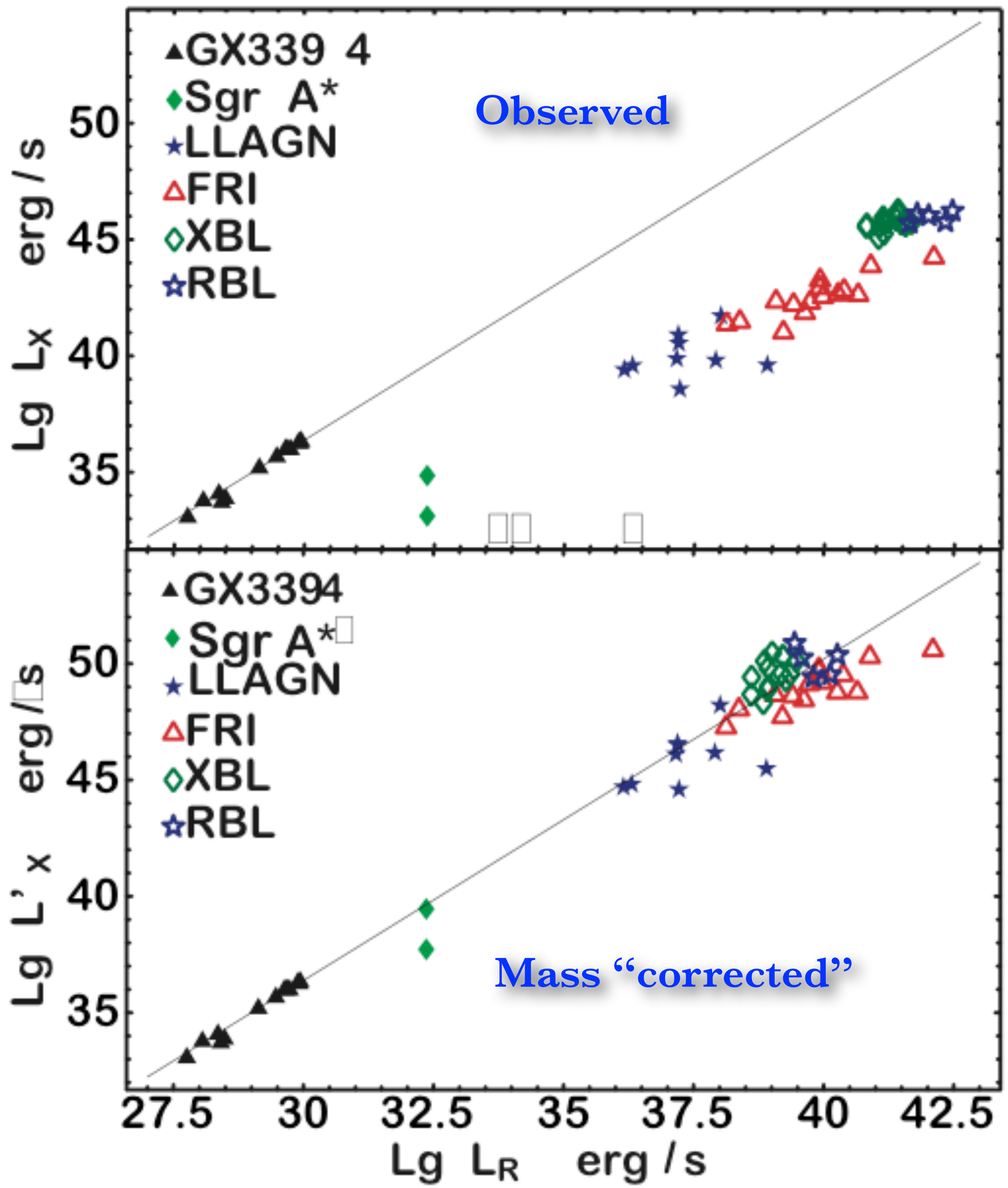
Expect same radio/X-ray correlation *slope* but AGN will have lower "normalization" in X-ray luminosity, comparatively!

Fundamental plane of black hole accretion



(Markoff et al. 2003, Merloni,Heinz & diMatteo 2003, Falcke, Körding & Markoff 2004, Markoff 2005, Merloni et al. 2006, Körding et al. 2006)

Fundamental plane of black hole accretion



(Markoff et al. 2003, Merloni,Heinz & diMatteo 2003, Falcke, Körding & Markoff 2004, Markoff 2005, Merloni et al. 2006, Körding et al. 2006)

Why Isis?

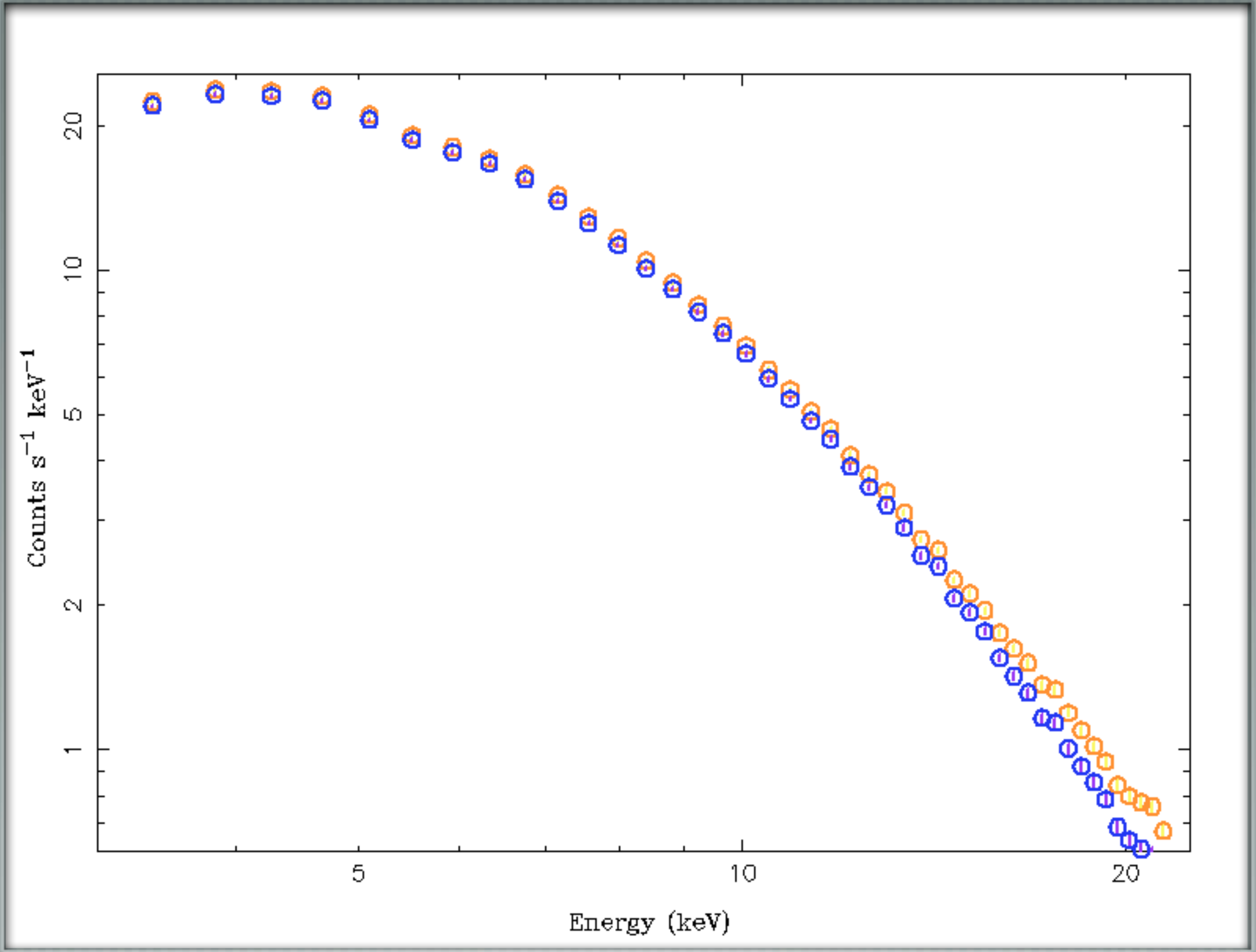
- At this point, probably Xspec12 can handle MW data and models better, but several years ago it was hopeless
- Isis is very programmable and flexible, if you can think of it, it can likely be done
- For non X-ray multiwavelength data it's especially nice because it can be imported as an ascii file

Getting started: PCA data (set 1)

- Load up Isis and then load the PCA data (try on your own first)
- Set .005 systematics on PCA and group min S/N=4.5, "notice" 3-22 keV, and (over)plot the data (cts/s/energy) with and w/out background

```
isis> variable pid = load_data("pca.pha");
isis> set_sys_err_frac(pid,Double_Type[129]+0.005);
isis> group(pid;min_sn=4.5,bounds=3.,unit="kev");
isis> notice_values(pid,3.,22.;unit="kev");
isis> xlog;ylog;point_size(1.5);
isis> set_plot_widths(;d_width=5,de_width=5);
isis> plot_data(pid;dsym=4,dcol=8,decol=7,bkg=1);
isis> plot_data(pid;dsym=4,dcol=4,decol=12,oplt=1);
```

PCA background features

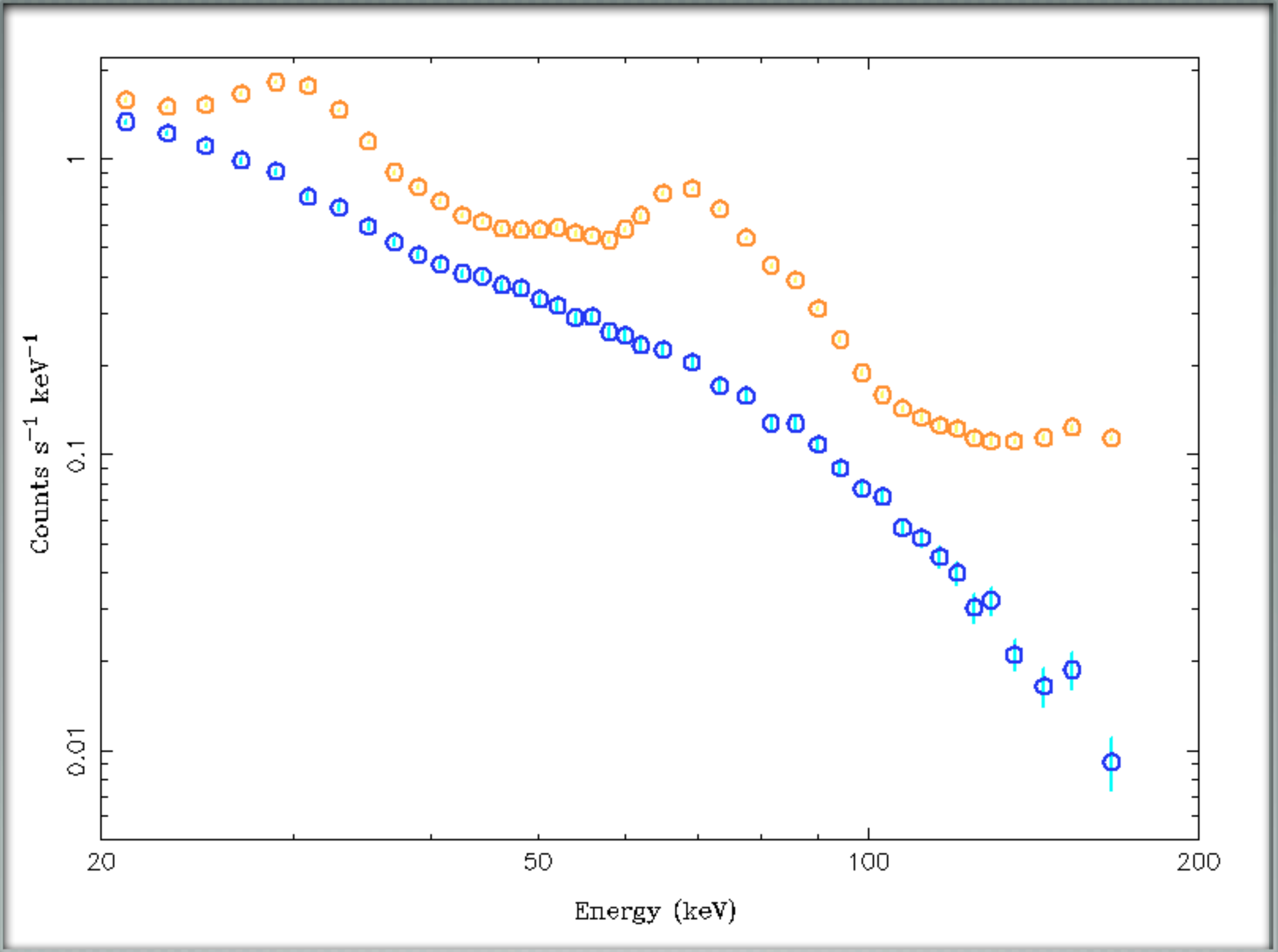


Now we'll add HEXTE data

- Treat similarly as PCA, group min S/N=4.5 but "bounds" now 20 keV and "notice" 20-200 keV
- Can call it anything you want! Just keep track which is which.
- Make the same two plots (will need to change x/y ranges)

```
isis> variable hid = load_data("hxt.pha");
isis> group(hid;min_sn=4.5,bounds=20.,unit="kev");
isis> notice_values(hid,20.,200.;unit="kev");
isis> plot_data(hid;xrange={20,200},yrange={0.005,2.2},
dsym=4,dcol=8,decol=7, bkg=1);
isis> plot_data(hid;dsym=4,dcol=4,decol=5,oplt=1);
```

HEXTE background features



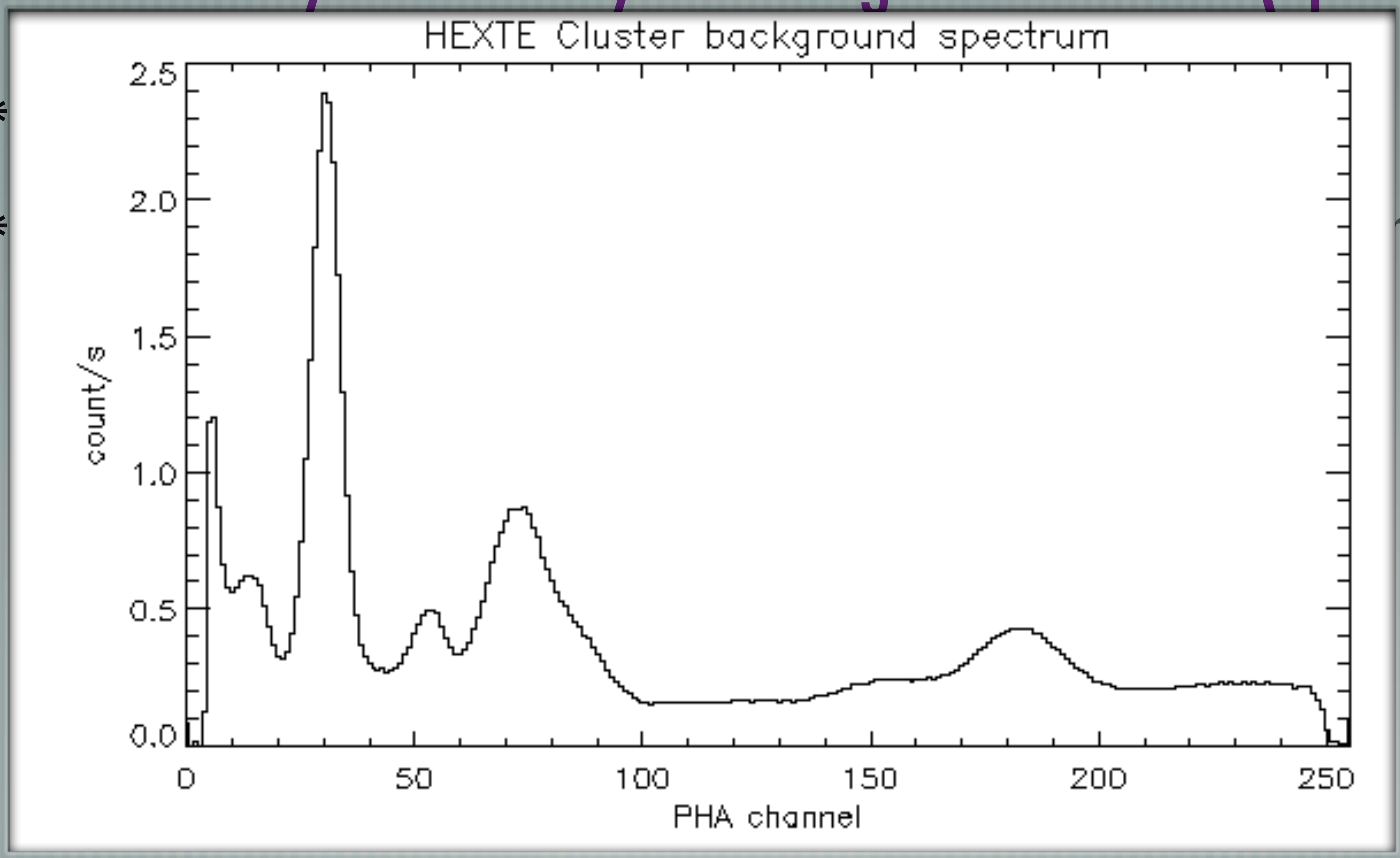
HEXTE background features

Line features all real, result from radioactive decays initiated by cosmic rays striking the detector (spallation)

- 30 keV line due to iodine and thallium decaying via K-capture
- Between 60–80 keV many features from interactions with iodine, tungsten and lead isotopes in the detector

HEXTE background features

- Line features all real, result from radioactive decays initiated by cosmic rays striking the detector (spallation)



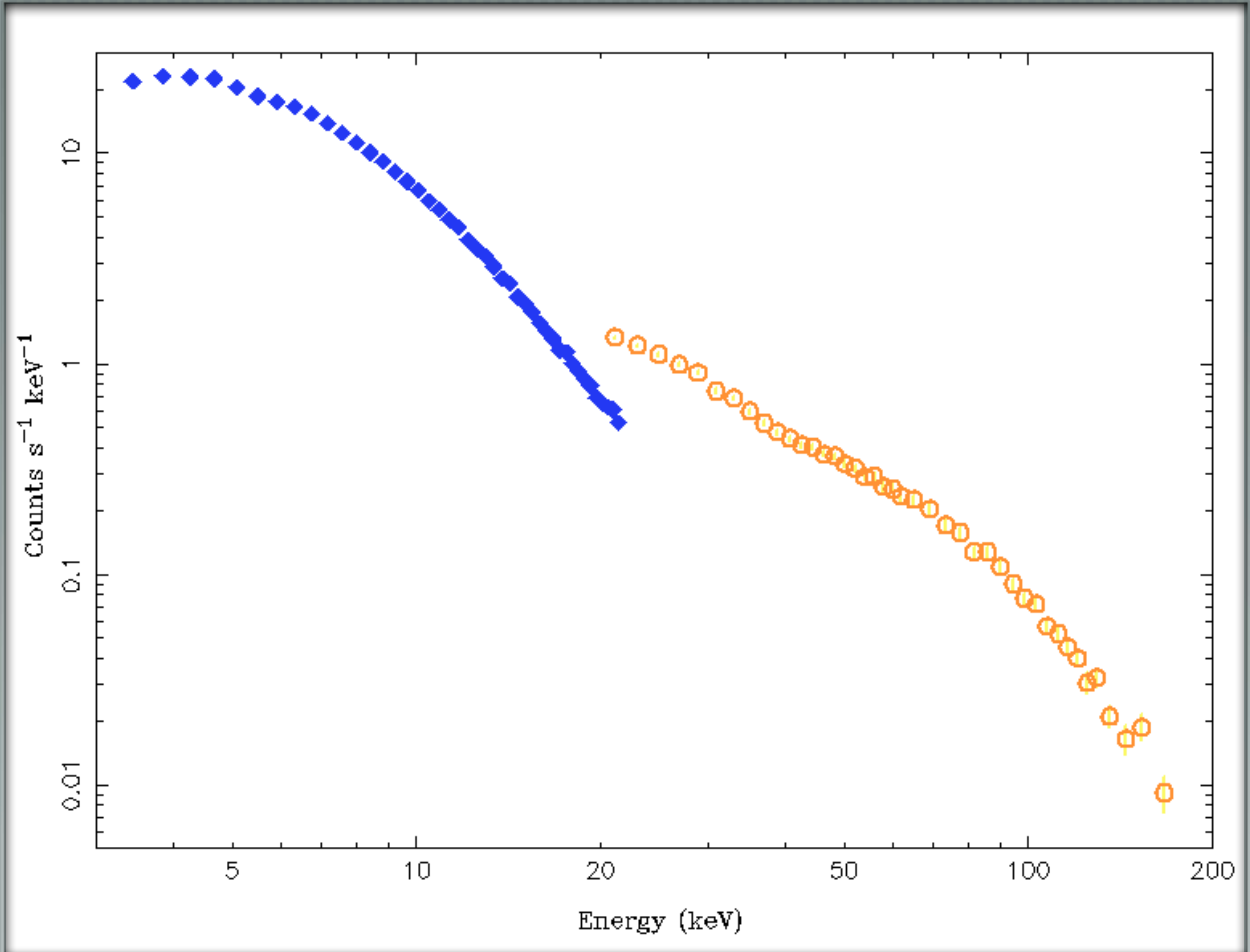
ne,

Plot both data sets: no bkgds

- Now plot both data sets on the same plot, using different colors for each (you've seen this before...)

```
isis> plot_data({pid,hid};dsym={-4,4},dcol={4,8}, decol=
{5,7},xrange={3,200},yrange={0.005,30});
```

PCA + HEXTE in counts/s/keV

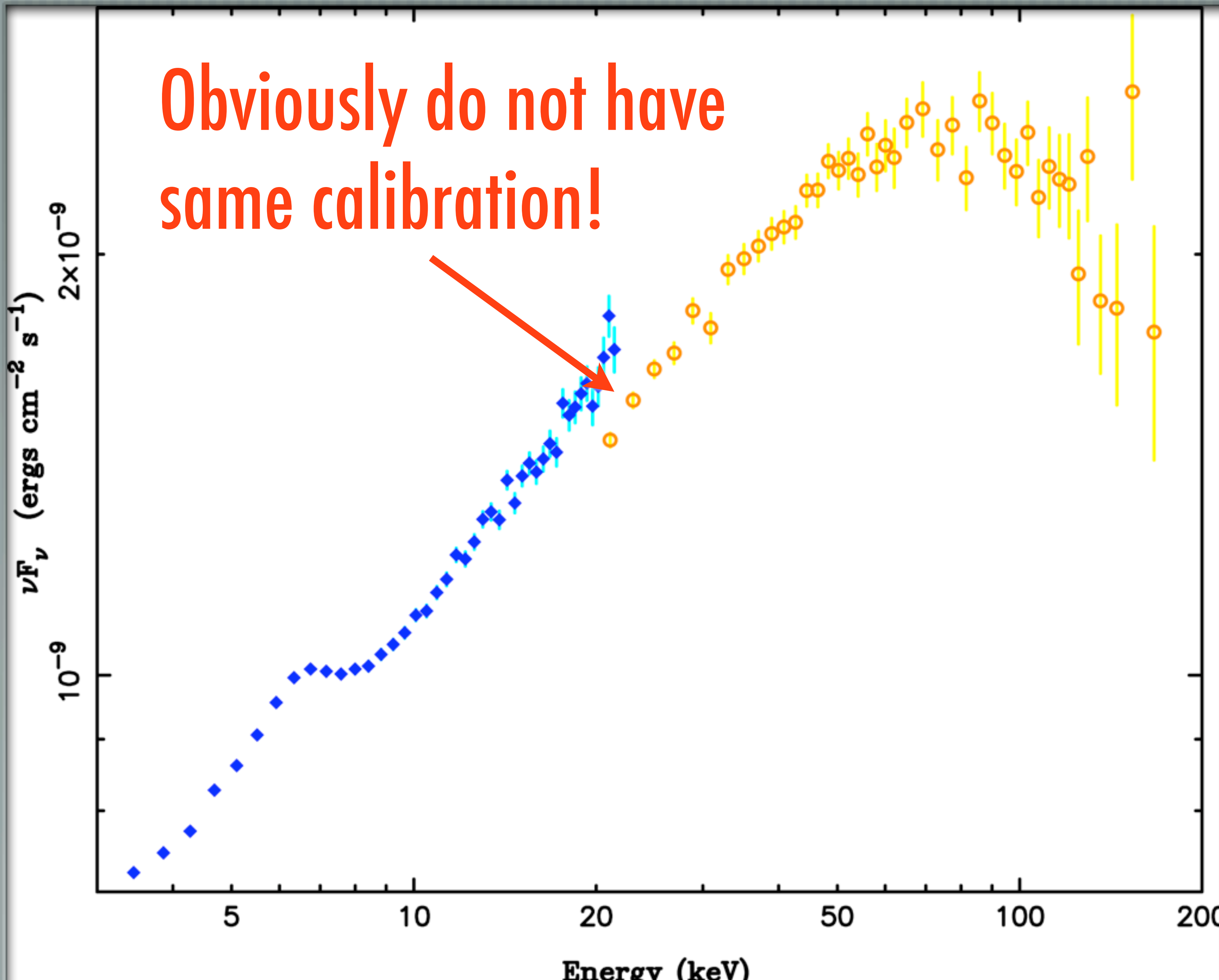


Plot both data sets: unfolded

- Now do the same thing for an unfolded (physical units) plot, with units of keV and ergs/cm^2/s

```
isis> fancy_plot_unit("kev","ergs");
isis> plot_unfold({pid,hid};dsym={-4,4},dcol={4,8},
decol={5,7},xrange={3,200},yrange=
{7.e-10,3.e-9},power=3);
```

Plot both data sets: unfolded (flux)



Now let's include radio and IR!

- Take a look at the content of the scripts "read_radio" and "load_radio"
- Input file is ASCII, 3 cols: Freq(Hz), Flux (mJy), errFlux(mJy)
- For any data w/out ARF & RMF, assumes RMF is diagonal with values =1, ARF = 1 cm^2 and "exposure" = 1sec. So we need to convert the input flux into counts/cm^2/s (for XSPEC!)

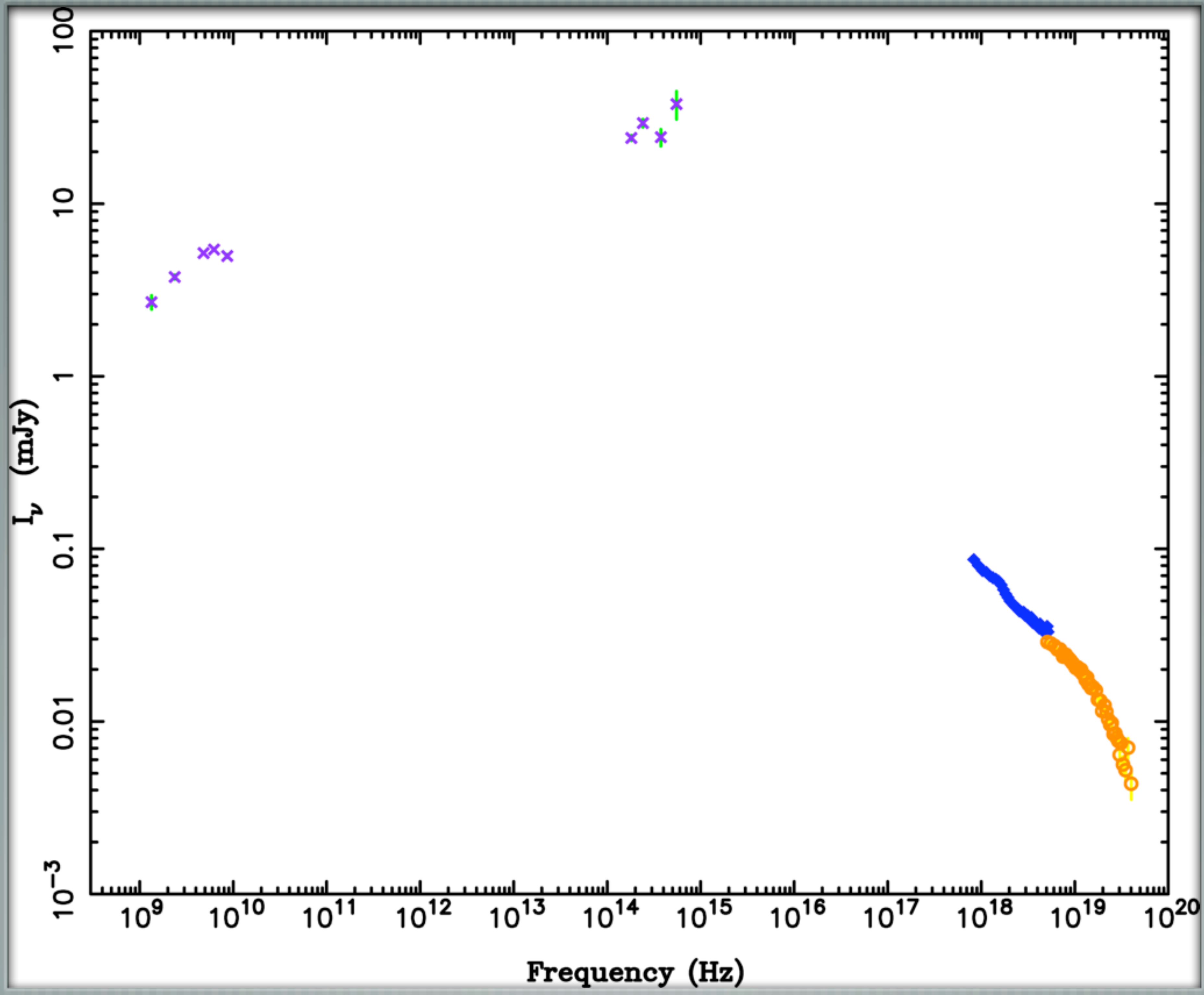
```
isis> variable rid_struct = read_radio("53082_radio-
OIR.dat",1000);
isis> variable rid = load_radio(rid_struct);
```

Plot the entire broadband spectrum

- First do plot_data, switch x axis units to “hz” and y axis to “mjy” and do plot_unfold with power=2 (flux units), you may need to set x and y ranges by hand if {NULL,NULL} doesn’t work

```
isis> fancy_plot_unit("hz","mjy");
isis> popt.dsym={5,-4,4};
isis> popt.dcol={12,4,8};
isis> popt.decol={3,5,7};
isis> popt.bkg={0,0,0};
isis> popt.mcol={0,0,0};
isis> popt.xrange={3.e8,1.e20};
isis> popt.yrange={0.001,100.};
isis> popt.power=2.;
isis> plot_unfold({rid,pid,hid},popt);
```

Plot unfolded xray + radio + IR data



Before modeling, need to grid data

- Isis internally makes a grid for each new dataset, and would evaluate model once for each. Gridding tells Isis explicitly what grids to use
- To group data from different instruments to have the same normalization, gridding is essential. Also reduces model runs.
- For using a reflection model gridding is also essential because model needs info up to MeV to calculate results at 10keV, but PCA data only goes to 30keV, and HEXTE data to 300 keV.
- Protocol called "caching" tells Isis to only evaluate model once and then apply to all grids, can save a lot of time for complex models.

Set radio and PCA to same norm.

- We will *choose* to have radio and PCA have same normalization, by putting them on the same grid. Given PCA's very good statistics, this is probably fine within 10% (smaller than typical radio/IR errors)
- To test consequences, would need to redo fits with radio and HEXTE on same grid and incorporate this into erros
- These are shorthand for a script that defines a logarithmic grid (see utilities for details)

```
isis> usr_grid([rid,pid],-9,3,0.001,1);
isis> usr_grid([hid],0,3,0.001);
```

Define some finger-saving aliases

- You can also set these in your .isisrc file so that it's automatically set:

```
isis> alias("eval_counts","ec");
isis> alias("list_par","lp");
isis> alias("newpar", "np");
isis> alias("set_par", "sp");
isis> alias("fit_counts", "fit");
```

- And since we'll be plotting models, add settings for plots

```
isis> set_plot_widths(;m_width=5, r_width=5,
re_width=5);
```

First let’s ignore the radio though...

- Type “exclude(rid)” to ignore radio data for now, and set the fit_fun model to be an absorbed broken powerlaw (bknpower)
- IMPORTANT: You want to multiply your model by constant (Isis_Active_Dataset) as discussed in Mike Nowak’s slides from Monday (see slide 15)

```
isis> exclude(rid);
isis> fit_fun("constant(Isis_Active_Dataset)*phabs
(1)*bknpower(1)");
isis> list_par;
```

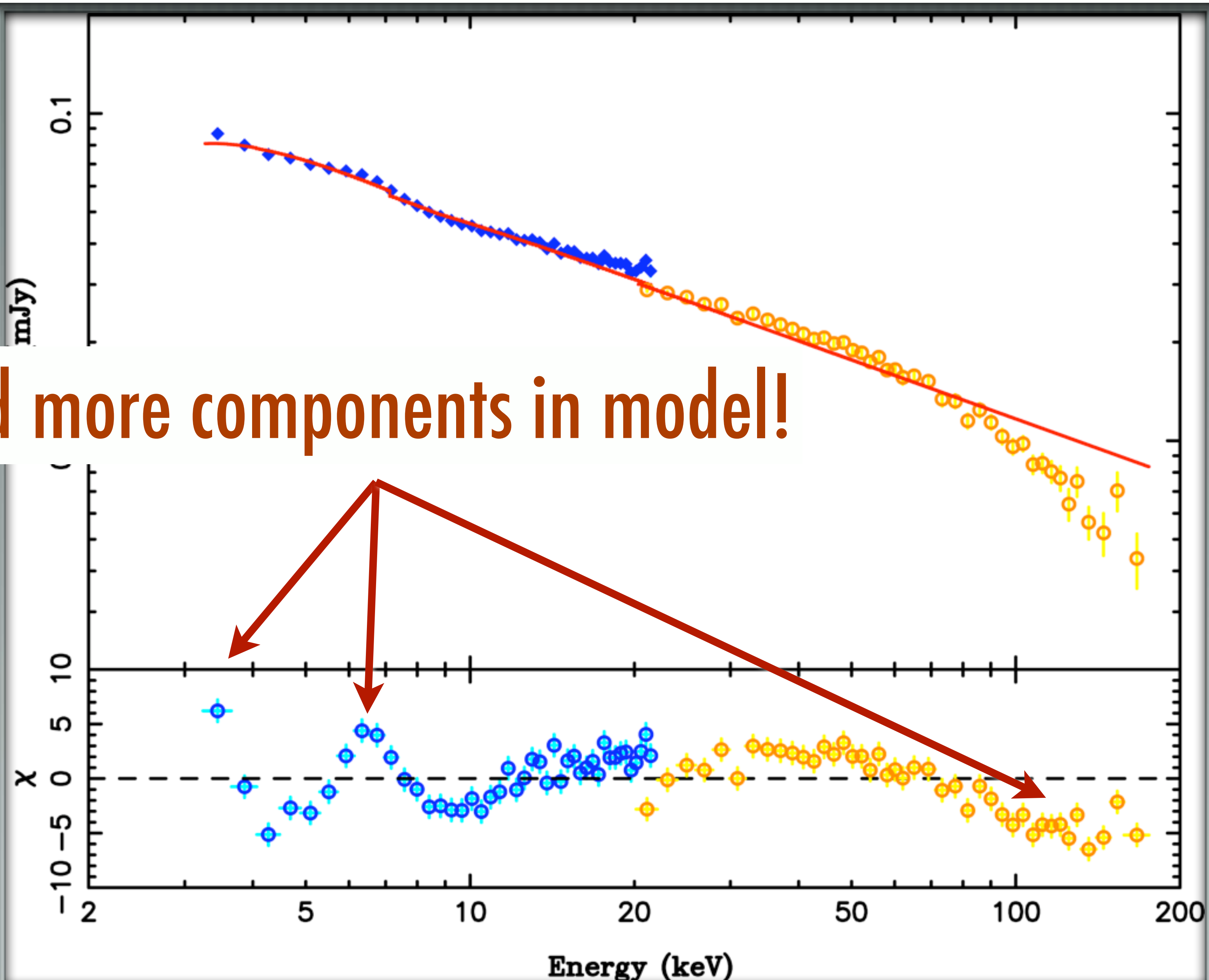
Optimize parameters, then fit

- Fix absorption to “typical” value as discussed in last couple days
- Fix constant(1), PCA normalization to 1, and limit HEXTE constant to .9-1.1 range (that reflects the very good overall calibration)
- Make some initial guesses for break energy and slopes, renorm_counts and fit_counts, see how good you can get, and plot with residuals

```
isis> set_par(2,0.6,1);set_par(1,1,1);
isis> set_par(7,1.,0,.9,1.1);
isis> popt.dsym={-4,4};popt.dcol={4,8};
isis> popt.decol={5,7};popt.mcol={2,2};
isis> popt.rcol={4,8};popt.recol={5,7};
isis> popt.xrange={NULL,NULL};
isis> popt.yrange={NULL,NULL,-10,10};
isis> popt.res=1; plot_unfold({pid,hid},popt);
```

Fitting broken PL + absorption

```
isis>plot_unfold({pid,hid};dsym={-4,4},dcol={4,8},decol={5,7},rcol=
{4,8},recol={5,7},xrange={2,200},yrange={0.002,.2,-10,10},res=1);
```



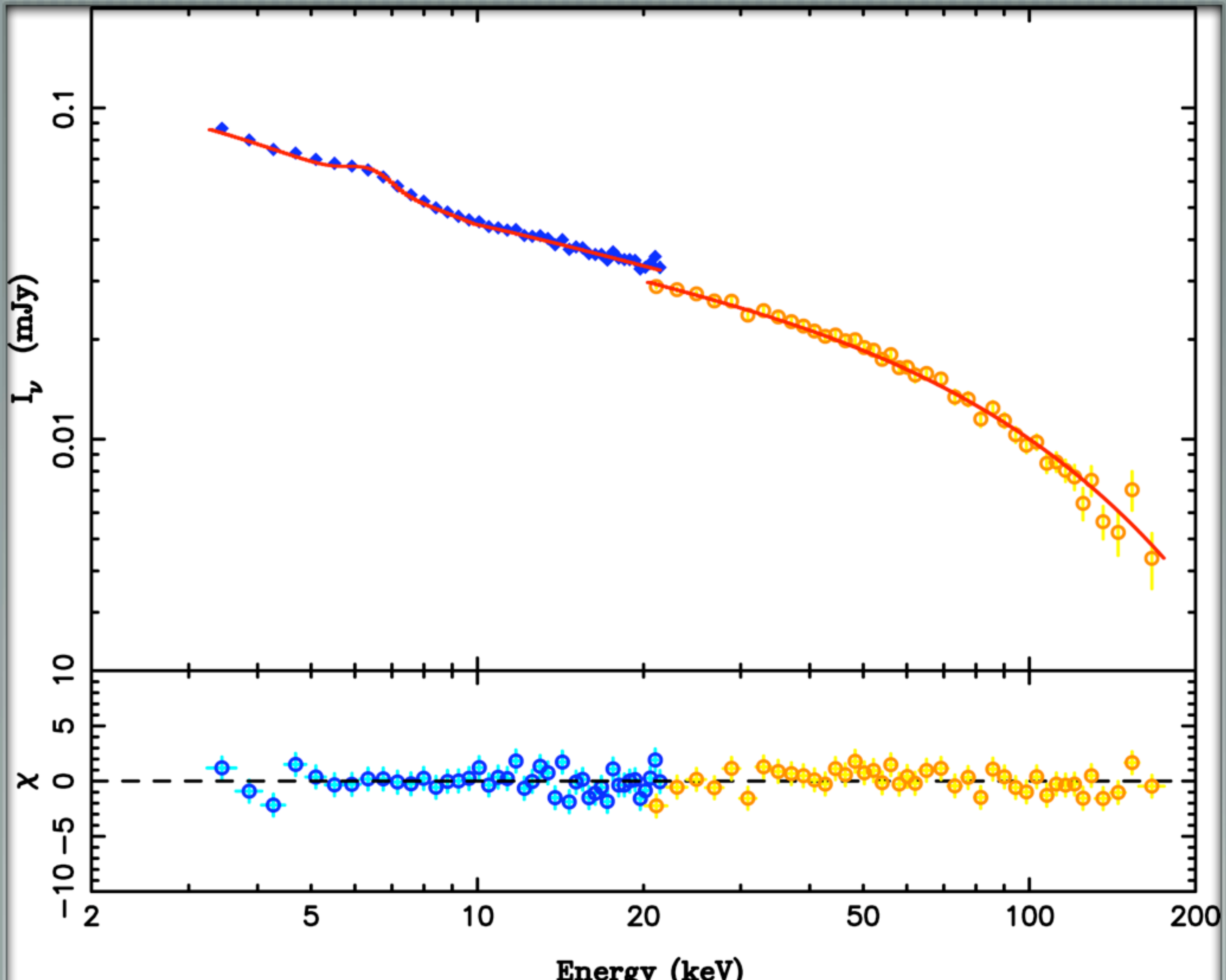
Need more components in model!

Add gaussian line, disk black body & cutoff

- Remember "vector math" of models. Are each of these additive or multiplicative?
- You've seen the line and disk black body, the cutoff model we'll use is called "highecut" – construct a new fit_fun using these three now!
- Use spectral features to give good starting values (i.e. disk temperature around 1 keV, cutoff at high energies...)

```
isis> fit_fun("constant(Isis_Active_Dataset)*phabs
(1)*highecut(1)*(bknpower(1)+diskbb(1)
+ gaussian(1))");
```

Better fit, but what about the radio...?



Let's re-notice the radio and refit

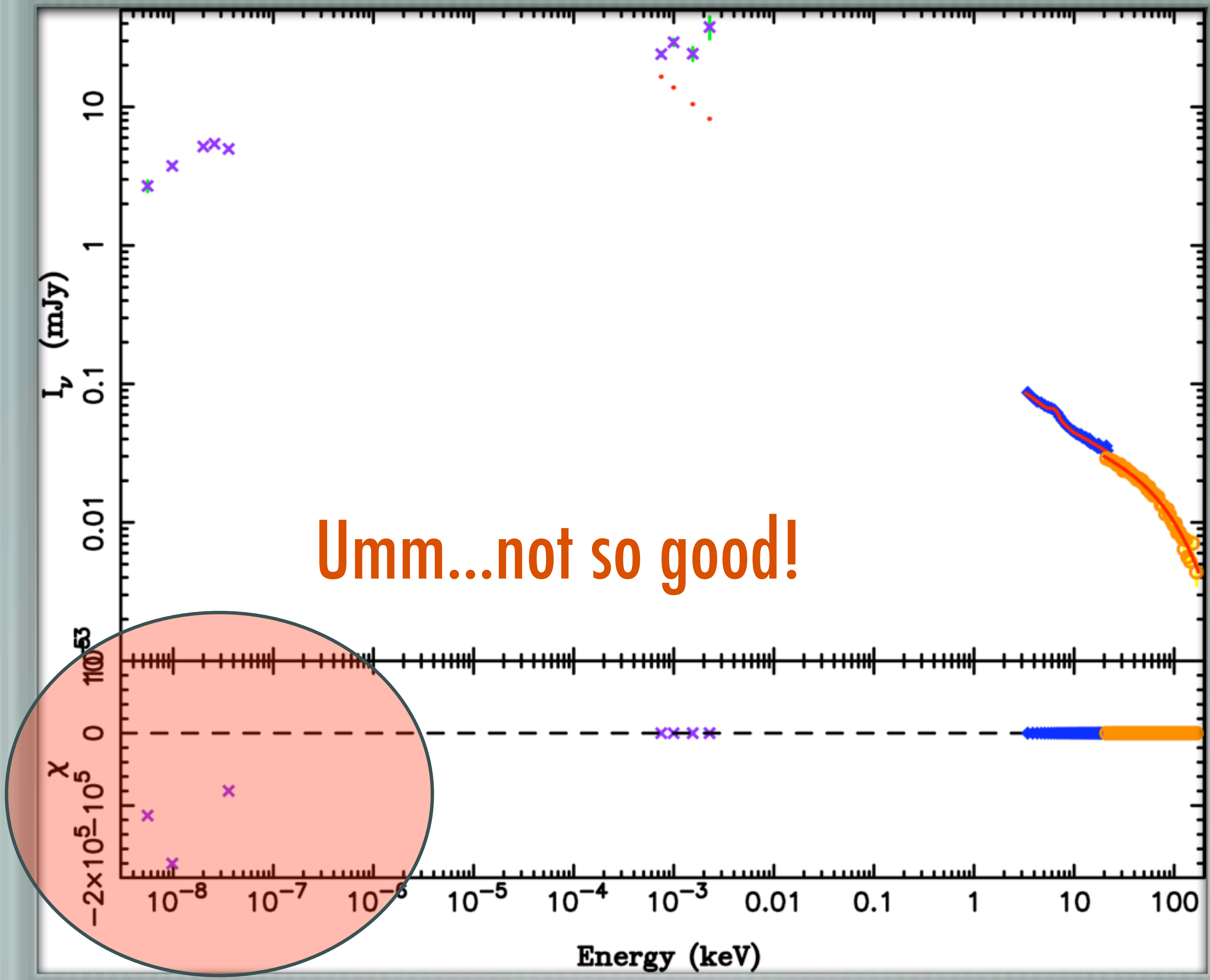
- First redefine popt for 3 data sets:

```
isis> popt.dsym={5,-4,4};
isis> popt.dcol={12,4,8};
isis> popt.decol={3,5,7};
isis> popt.rcol={12,4,8};
isis> popt.recol={3,5,7};
isis> popt.mcol={2,2,2};
isis> popt.xrange={3.e-9,250};
isis> popt.yrange={1.e-3,50,-2.e5,2.e5};
isis> popt.power=2;
isis> popt.rsym={5,-4,4};
isis> popt.res=1;
```

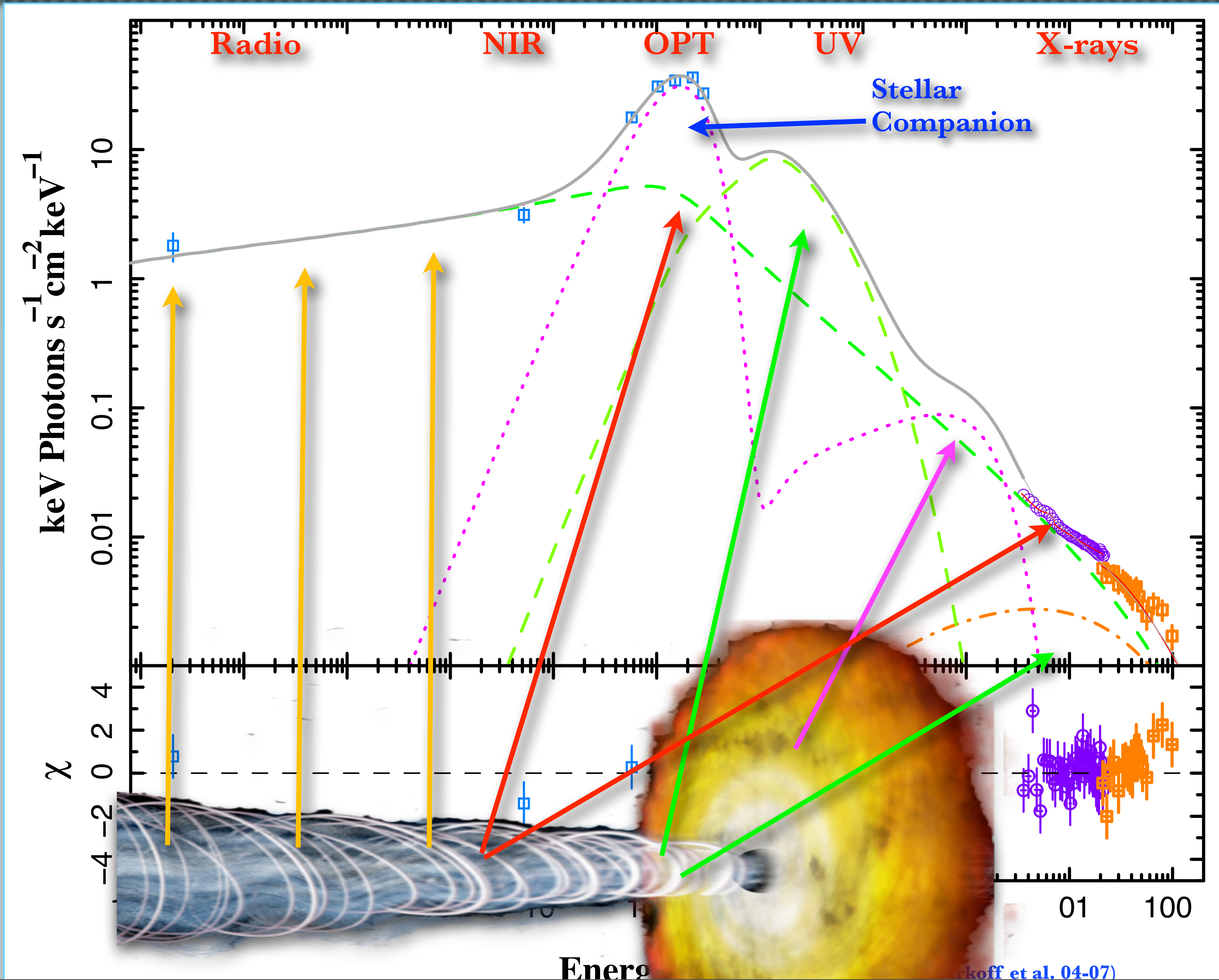
Let's re-notice the radio and re-evaluate the fit

```
isis> include(rid);
isis> lp;  % you will see a new constant
isis> sp(15,1,1);
isis> tie(15,1);
isis> () = eval_counts;
 Parameters[Variable] = 15[14]
            Data bins = 95
           Chi-square = 1.610517e+11
   Reduced chi-square = 1.988292e+09
isis> plot_unfold({rid,pid,hid},popt);
```

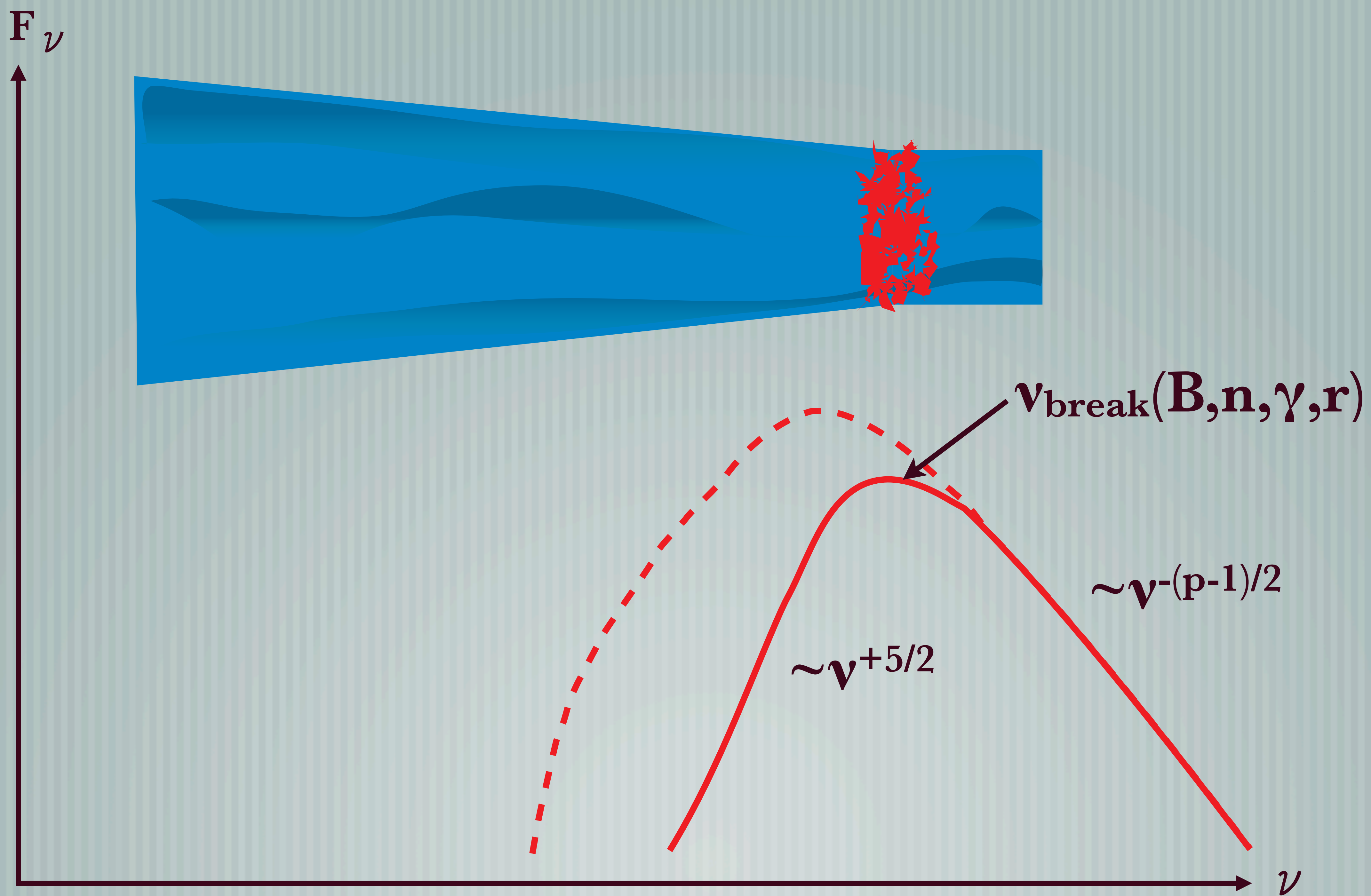
Fitting abs + bknpl + diskbb + line (xray)



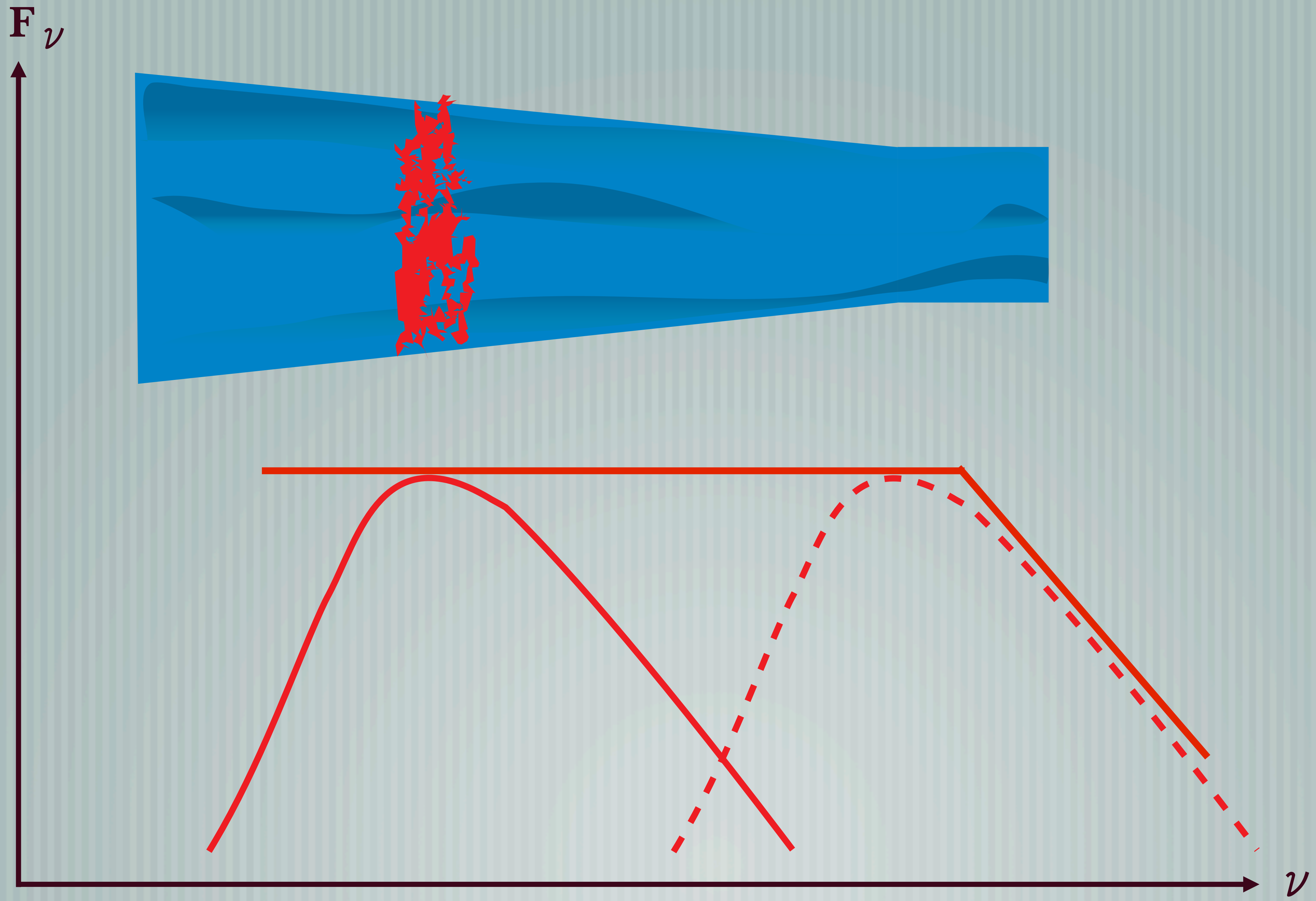
MW dependence: “microquasar”



Understanding compact jet spectra



Understanding compact jet spectra



So based on physics, try new pl indices

- Go back to simple absorbed PL with cutoff, start by giving a slightly inverted index for the first PL

```
isis> fit_fun(“constant(Isis_Active_Dataset)*phabs(1)
*highecut(1)*bknpower(1)");
isis> lp;
isis> set_par(2,.6,1);
isis> set_par(5,.4);
isis> set_par(6,.82);
isis> set_par(7,2.5e-4,0,1.e-5,1.e5);
isis> set_par(8,1.6);
isis> ec;
isis> renorm_counts;
```

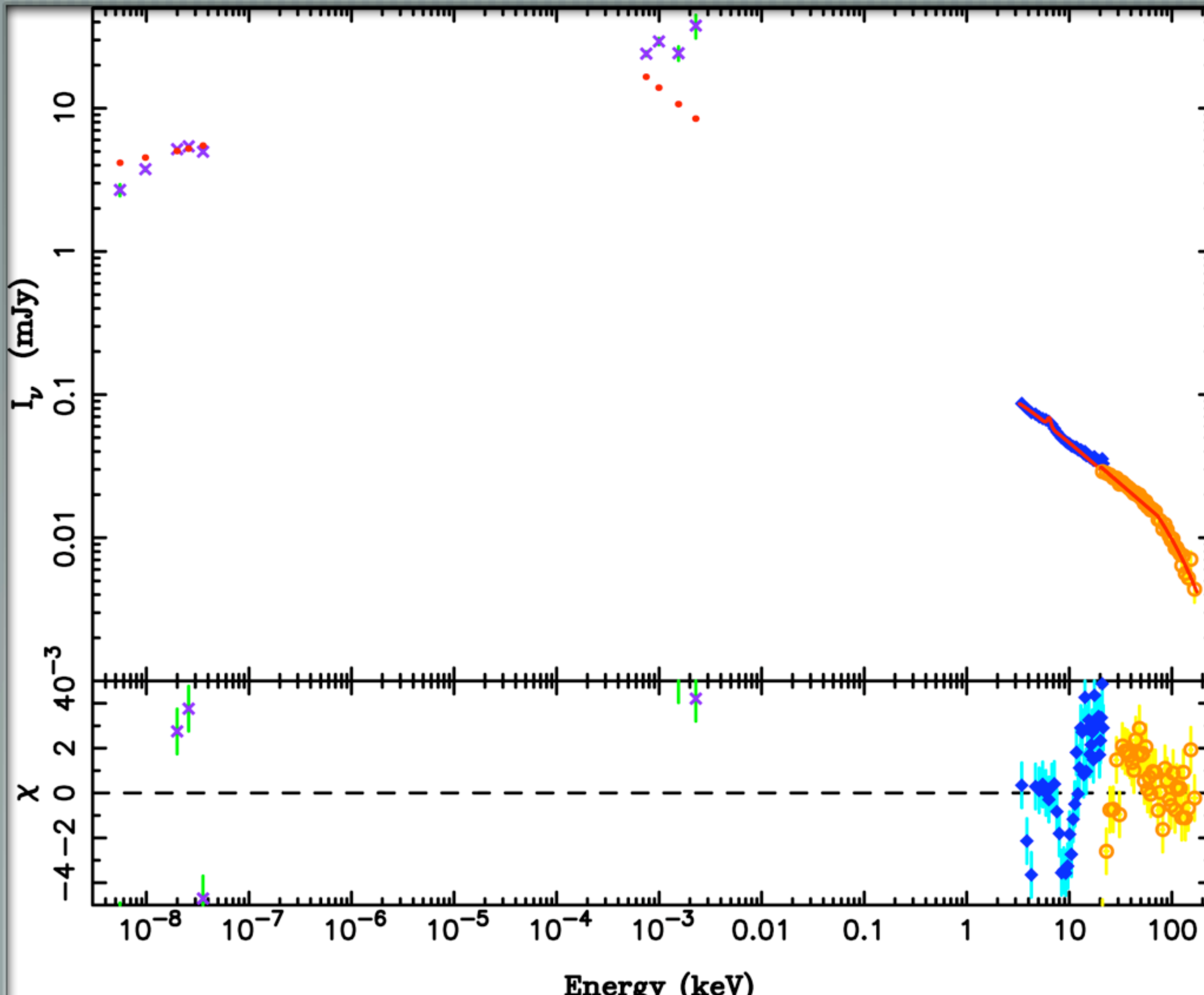
So based on physics, try new pl indices

- Now add back in the Gaussian and blackbody disk

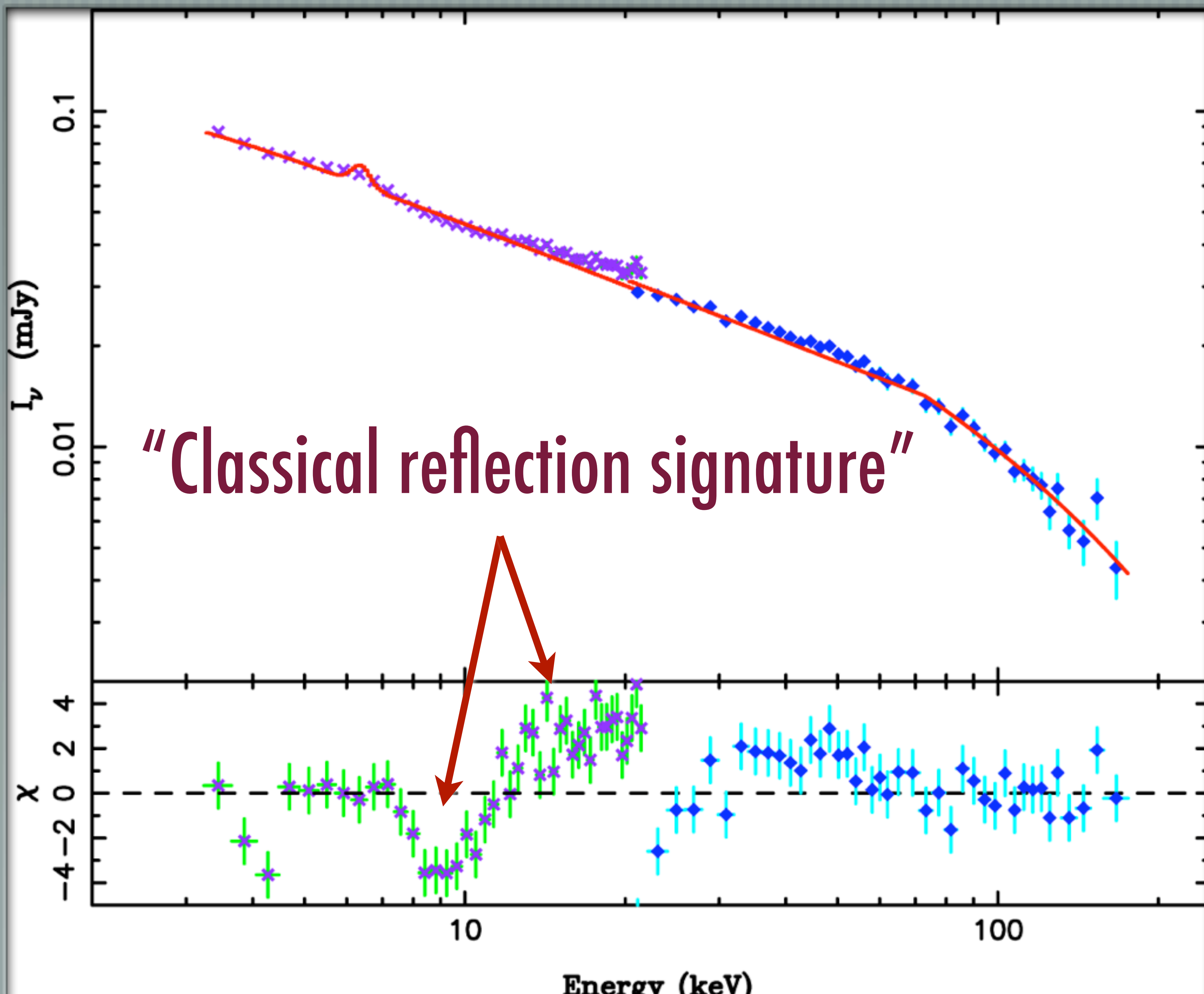
```
isis> fit_fun("constant(Isis_Active_Dataset)*phabs(1)
*highecut(1)*(bknpower(1)+diskbb(1)+gaussian(1))");
isis> renorm_counts;
isis> fit;
```

- Should get a reasonable fit value at this point

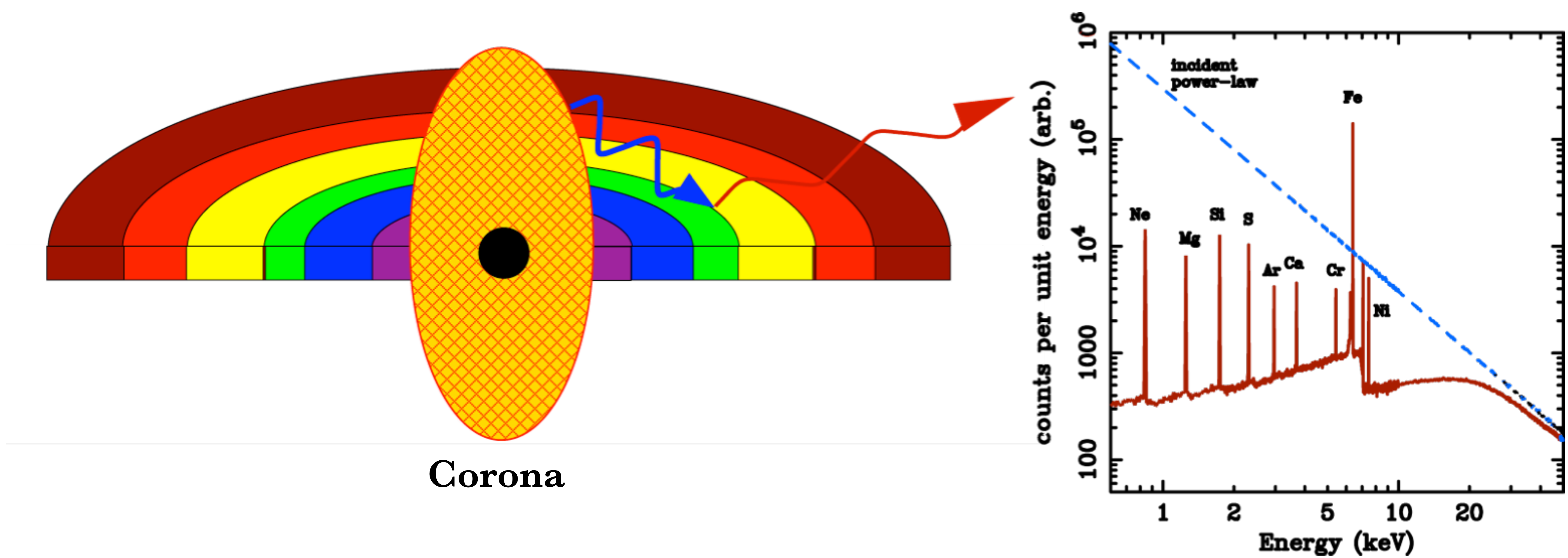
Fit now better adapted to radio



But now look at X-ray residuals...



Reflection of hard X-rays off cooler disk

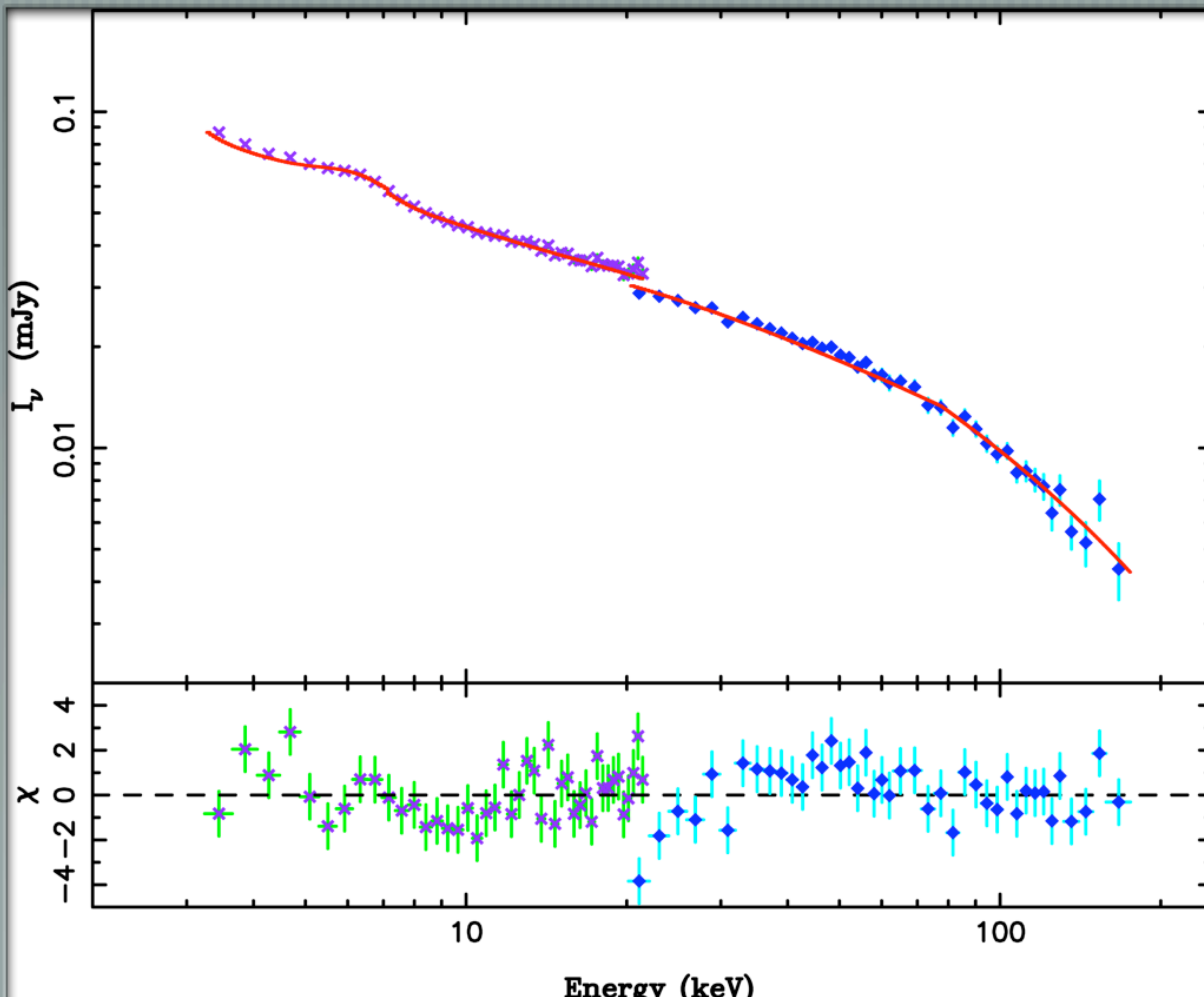


Add reflection, a convolution model

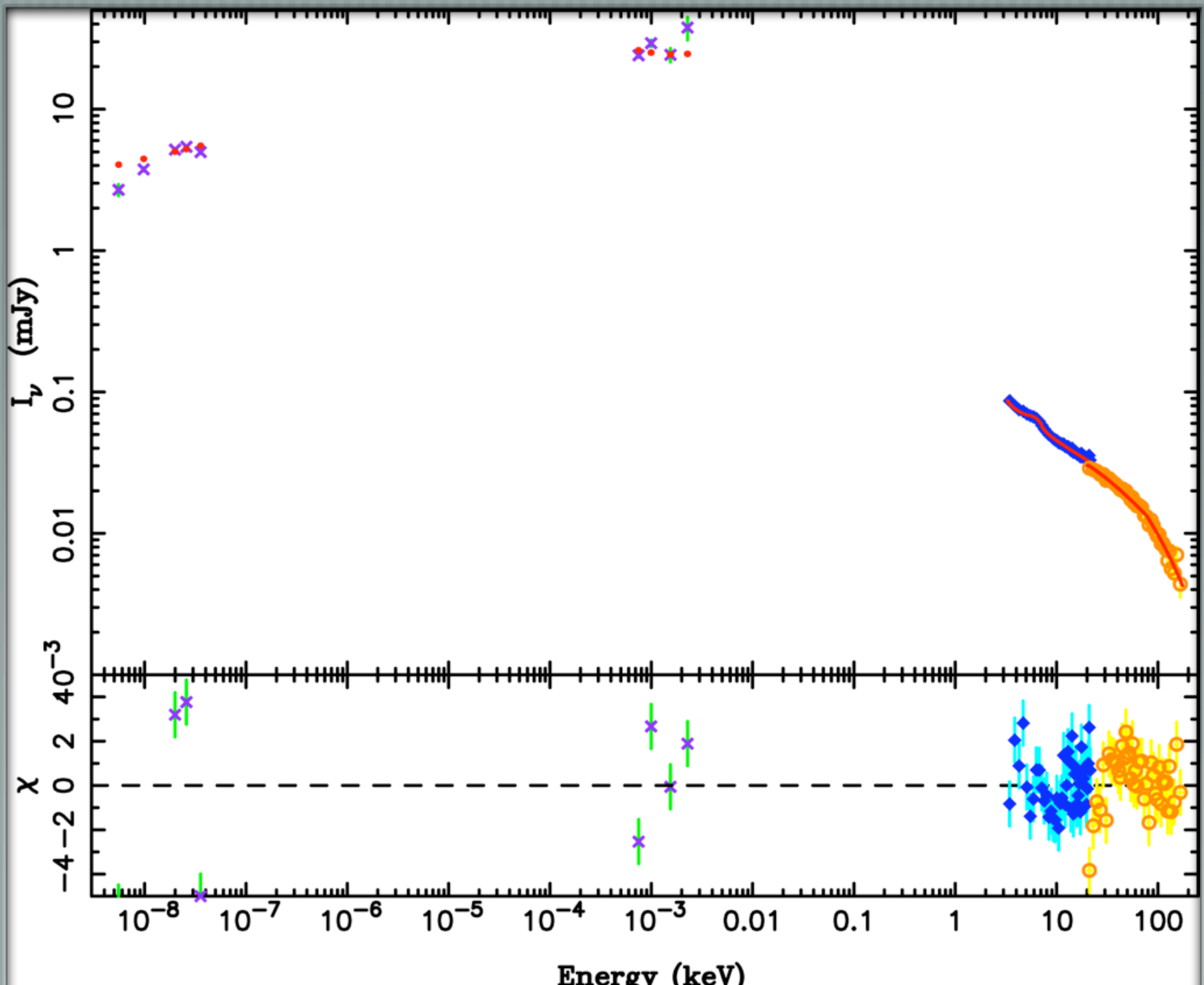
- The reflection should be convolved with all things contributing to the hard X-ray flux
- Takes form: reflect(1, convolved models..)

```
isis>  fit_fun("constant(Isis_Active_Dataset)*phabs
(1)*reflect(1,highecut(1)*(bknpower(1)+diskbb(1)+
gaussian(1)))");
```

Including 27% reflection improves fit a lot



Not a terrible fit, for a simple model!



How to include a table model

- Sometimes you want to include a model component that is not an analytical expression, for instance an individual star model, or a galactic stellar population model
- We'll use an example of a single star model, built from data in "star.dat". You can look into these scripts

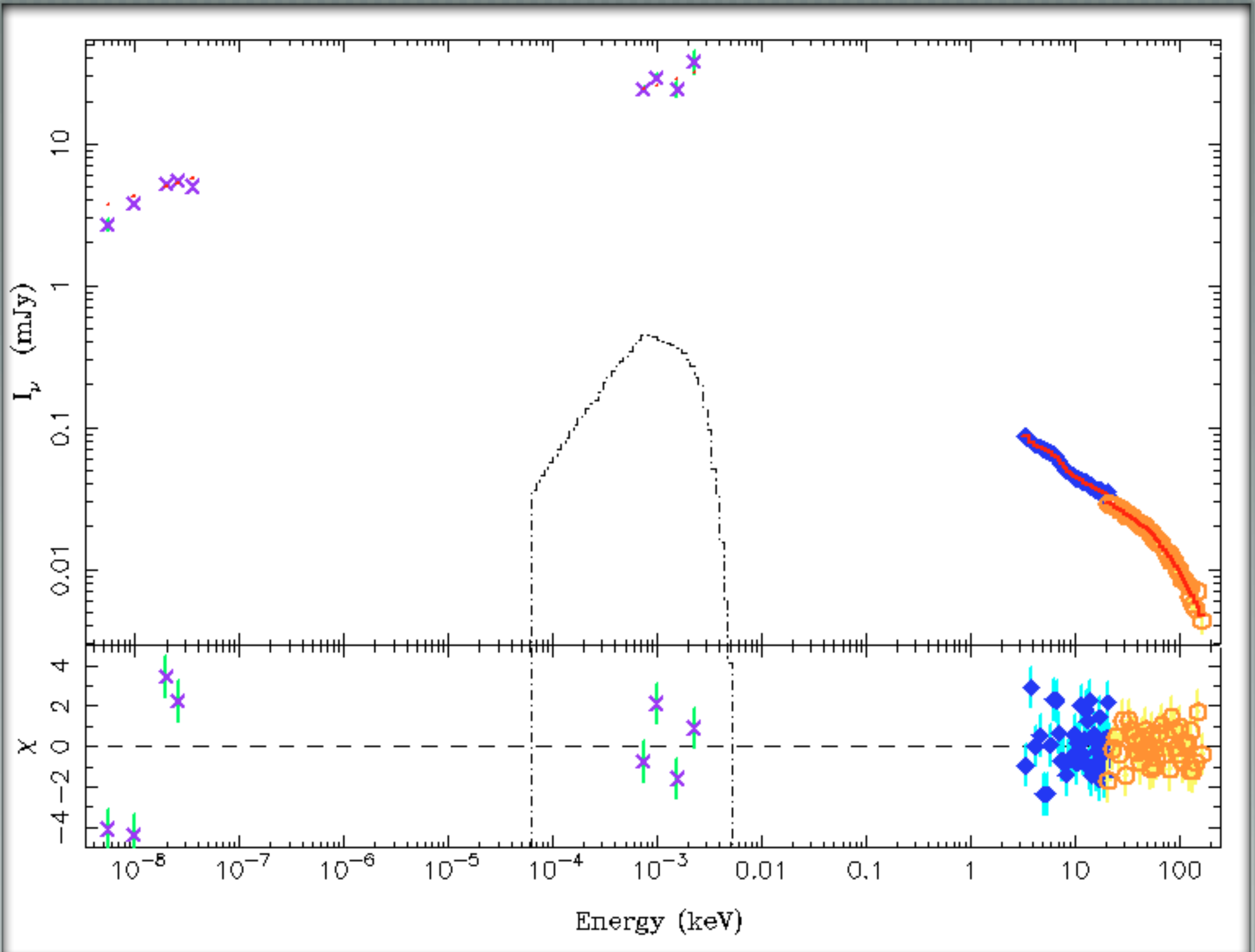
```
isis> ()= evalfile("star_model.sl");
isis> ()= evalfile("eval_fun2_kev.sl");
isis> ()= evalfile("log_grid.sl");
isis> (lo,hi) = log_grid(1.e-6,1.e-2,100);
isis> variable starcomp=eval_fun2_keV
("star",lo,hi,get_par("star(1).norm"));
```

Include model “star” in your fit_fun

- Change the fit_fun to introduce this additive component

```
isis> fit_fun("constant(Isis_Active_Dataset)
*phabs(1)*reflect(1,highecut(1)*(bknpower
(1)+diskbb(1)+gaussian(1)+ star(1)))");
isis> fit;
```

Include model “star” in your fit_fun



Save all your commands/parameters

```
isis> save_input("mwfitting.sl"); %history!!
isis> save_par("mwfitting.par");
isis> quit;
```

Now suppose you want to use your own model as input code?

- Any model written as a C or Fortran subroutine can be SLIRP'ed into Isis, i.e., converted into a S-lang module
- If you want the model to potentially work in Xspec, there are some constraints on the input/outputs which require a "translation" wrapper for the grid (see simplejet.sl)
- Follow these steps to use an over-simplified HD jet model, "simplejet" written in Fortran

```
unix> slirp -make simplejet.f
unix> make
unix> make test
```

Set up paths in .isisrc to be able to "find" the new model

- For simplicity we'll just add the current directory to isis load and module paths, but in general you may want to have a separate directory where you keep your custom models
- Inside .isisrc add the lines (obviously sub in your own path!):

```
prepend_to_isis_load_path("/Users/sera/meetings/ITN/2ndschool/SEDexercises/simplejet");

prepend_to_isis_module_path("/Users/sera/meetings/ITN/2ndschool/SEDexercises/simplejet");
```

Assuming that worked, load model

- Go back into Isis, using your saved input files as a reference now, and reload radio/IR, PCA and HEXTE data
- Set the grids as before
- Now load the new model:

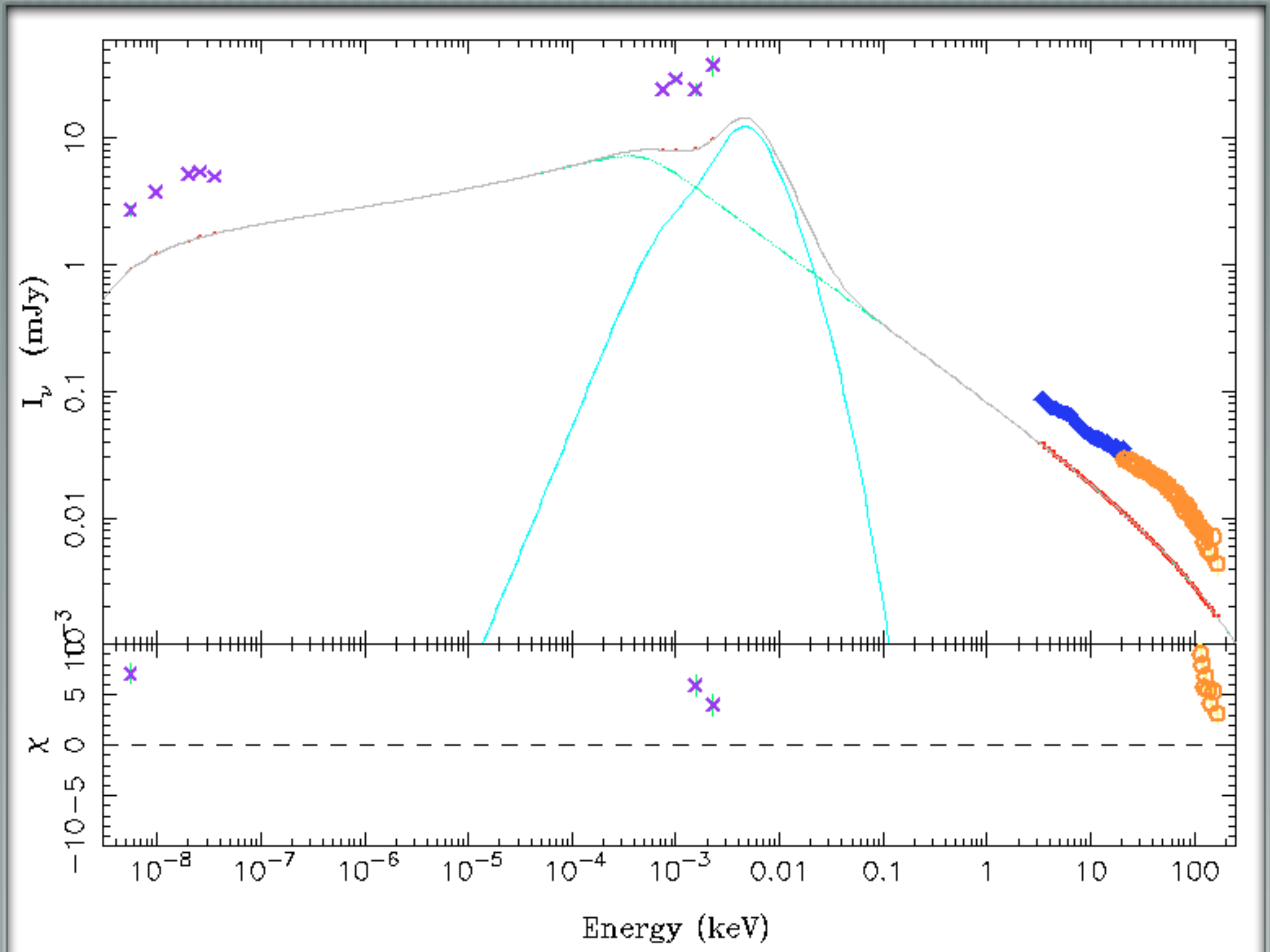
```
(or)
isis> .load load_data_SED

isis> () = evalfile("simplejet.sl");
isis> fit_fun("simplejet");
isis> list_par;
```

Let's look at the input parameters

```
isis> lp;
simplejet
 idx  param              tie-to  freeze          value          min          max
  1  simplejet(1).norm      0      1                 1            0      1000000
  2  simplejet(1).mbh       0      1                 7            1        1e+09  msun
  3  simplejet(1).jetrat    0      0              0.03        1e-07            1  L_edd
  4  simplejet(1).pspec     0      0               2.2          1.7            4
  5  simplejet(1).zsh       0      0               100            5        10000  r_g
  6  simplejet(1).r0        0      0                15            2          100  r_g
  7  simplejet(1).incl      0      1                55            0           90  deg
  8  simplejet(1).eltemp    0      0             2e+10        2e+09        5e+11  K
  9  simplejet(1).plfrac    0      0              0.75       0.0001            1
 10  simplejet(1).dkpc      0      1                 2          0.1        10000  kpc
 11  simplejet(1).plotsw    0      1                 1            0            1
 12  simplejet(1).fsc       0      0            0.0036        1e-07        0.036
 13  simplejet(1).zmax      0      1              13.5           13           16  lg cm
 14  simplejet(1).equip     0      0                 2          0.1          100
```

Should give a fit like:



Time to cache model

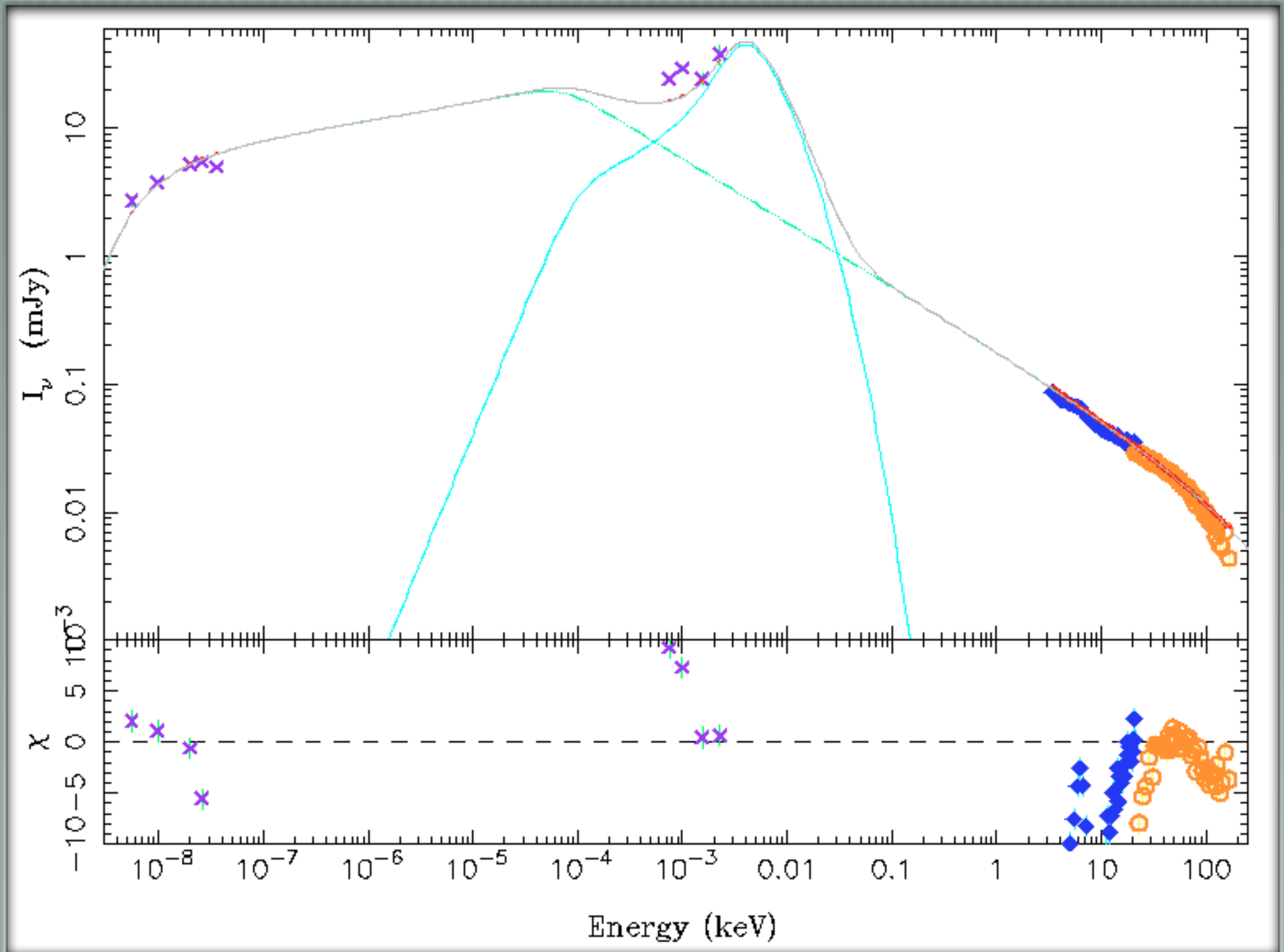
- Note that every time you eval_counts, the model is run twice. That's because pca/radio are already cached on one grid, but hexte has its own grid and the model is run for each grid
- Before fitting, to save time, let's cache the model on all grids

```
isis> loval = _A(10^[-9:3:0.001]);
isis> hival = make_hi_grid (loval);
isis> cache_fun("simplejet",loval,hival);
isis> fit_fun("simplejet_cache");
```

Now try to improve your fit, using simplejet_cache in place of bknpower

- Using your saved input file as a reference, start with constant on Isis_Active_Dataset, absorption, simplejet and a gaussian
- Freeze absorption at “typical value” as before, give good Fe line starting values, tie constant(1) to constant(3) and fix to one (all as before) and limit range for constant(2) to between 0.9 - 1.1
- renorm_counts and do fit_counts etc., play around between subplex and lmdif fit methods
- Eventually add diskbb and reflection, and if you feel adventurous you can also add a second single blackbody in the IR/opt

Here's my fit with just simplejet_cache



Here's my fit with simplejet_cache, ecut, gaussian, diskbb, reflection