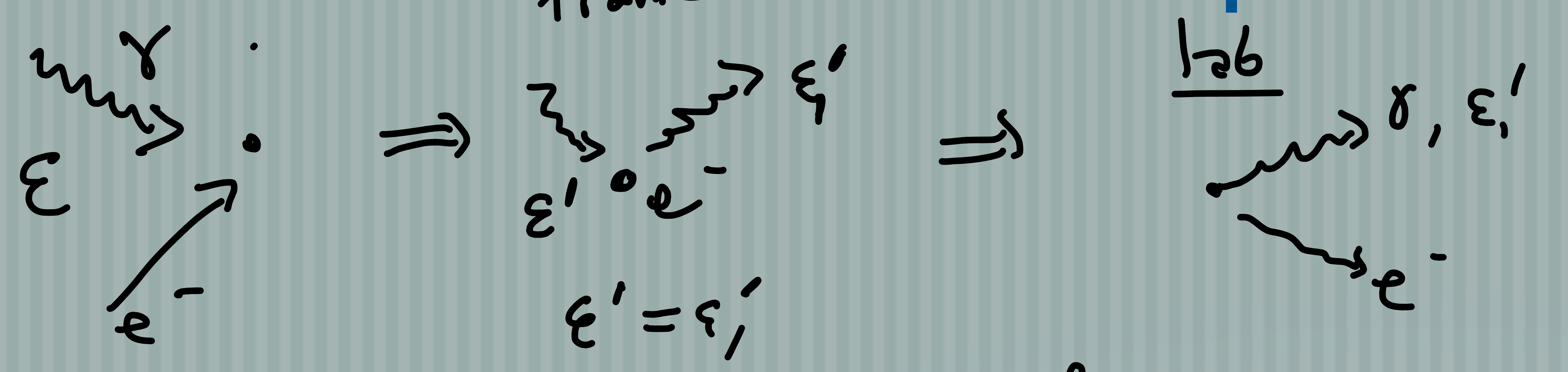
A bit of OoM Follow up + "showcase" for modeling broadband (Multiwavelength) SEDs

Sera Markoff (API/GRAPPA, U Amsterdam)

Rough outline

- Recap of IC: another Sgr A* practice problem
- Why it's tricky to *statistically* fit X-ray and non X-ray data simultaneously
- Intro to Isis code, step-by-step through tutorial to showcase the basics:
- fitting MWL SED with simple models, adding more physical components
- SLIRPing your own models into Isis to fit data

Quick recap on inverse Compton



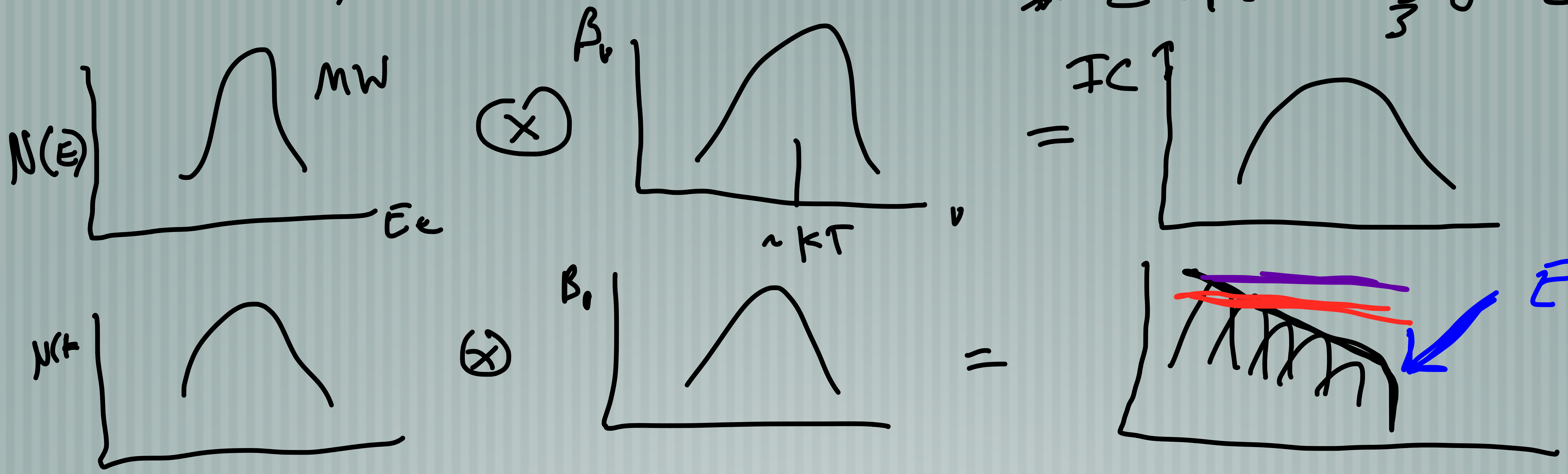
$\varepsilon : \varepsilon_1' : \varepsilon_1$

$1 : \gamma : \gamma^2$

$* \ \langle \varepsilon_1 \rangle \sim \frac{4}{3} \gamma^2 \varepsilon$

$\tau \ll 1$

$\tau \lesssim 1$



not a lot of change

$E_{e,max}$

$\tau \uparrow$: How effect slope of PL → hardening

$\Delta \varepsilon = \frac{\varepsilon_1 - \varepsilon}{\varepsilon}$ goes up? : → hardening

Quick recap on inverse Compton II

$$y = \frac{\langle \varepsilon_1 - \varepsilon \rangle}{\langle \varepsilon \rangle} \equiv \left\langle \frac{\Delta\varepsilon}{\varepsilon_{sc}} \right\rangle \langle N \rangle = \eta N$$

$$\begin{cases} N \sim \max(\tau, \tau^2) \\ \eta \sim 16 \left(\frac{kT}{mc^2} \right)^2 \end{cases}$$

relativistic Thermal

After N scatters $\quad \frac{\varepsilon_1}{\varepsilon} \sim (1+\eta)^N \sim 1 + \eta N + \frac{(\eta N)^2}{2!} + \dots$

$$\sim e^{N\eta} \sim e^{y}$$

At least PL regime $\Rightarrow$ $y \lessapprox 1$, $\quad I_{ph.} \sim \varepsilon^{1/y}$

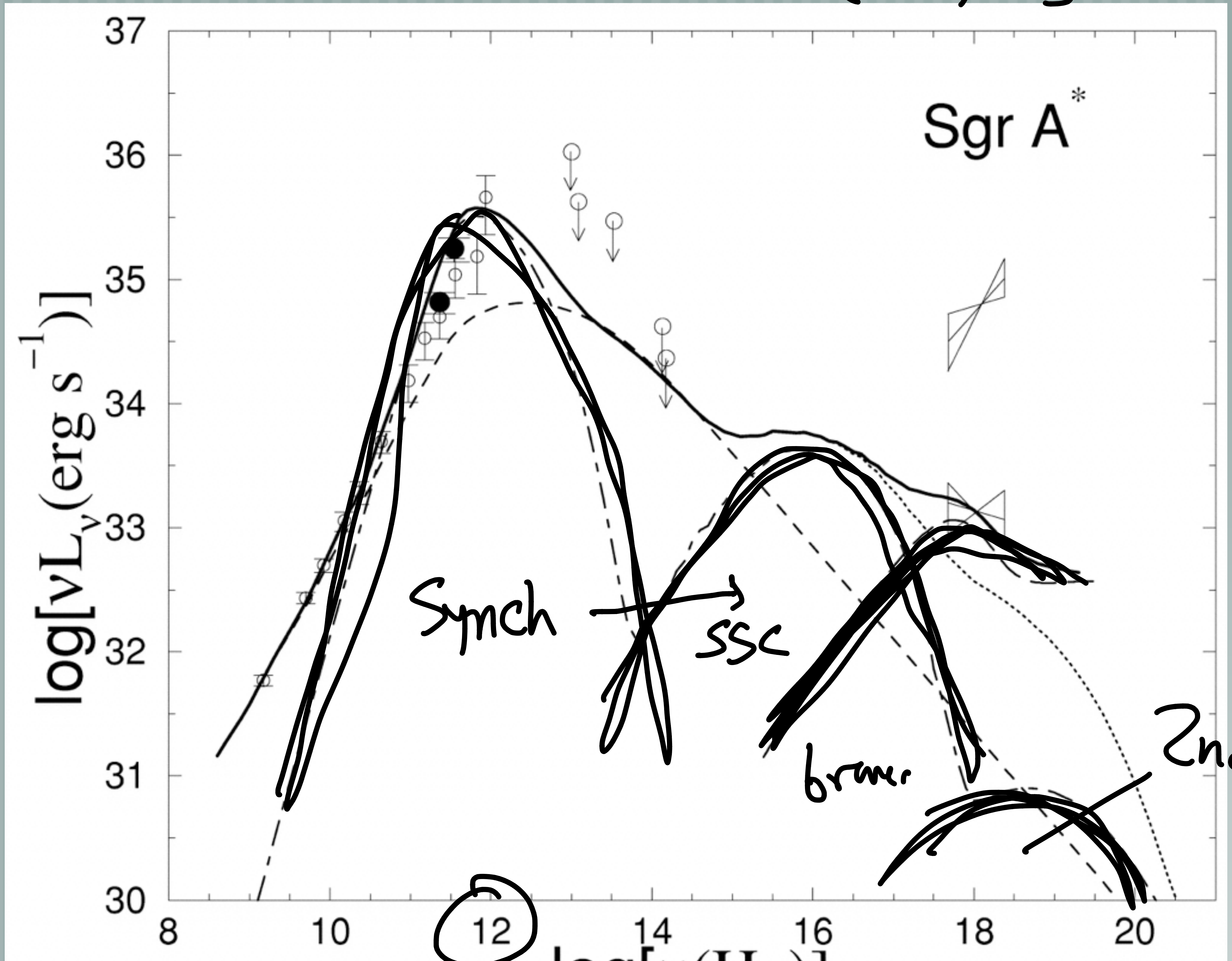
Guess what? Sgr A* again!!

$M = 4\times10^{6} M_\odot$

$\sigma_T = 6.7\times10^{-25}\ \mathrm{cm}^2$

$1.5\times10^{5}(M/M_\odot) \sim r_g$

$y \equiv 16\left(\frac{kT}{mc^2}\right)^2 \tau$



Yuan et al. 2003

$\tau ? \ll 1$

$\frac{\langle \epsilon_1 \rangle}{\epsilon} \sim 10^{16-12} \sim 10^{4} \propto \gamma^2$

$\langle \gamma \rangle_e \sim 100$

$L_{tot,syn} \sim 10^{35.5} = \frac{4}{3}\sigma_T \gamma^2 \beta^2 \frac{U_B}{8\pi} n \cdot Vol$

know $B \cong 100\,G \Longrightarrow n \sim 10^{4}$

$\tau = n\sigma R = 10^{4} \cdot 7\cdot10^{-25} \cdot 10\cdot10^{5}\cdot10^{6.6}$

$= 10^{4+0.8-25+1+5+6.6} \cong 10^{-8}$

$y ? = 16\cdot\left(\frac{kT}{mc^2}\right)^2 10^{-8}$

$\gamma \cong \frac{3kT}{mc^2} \simeq 16\cdot(30)^2\cdot10^{-8}$

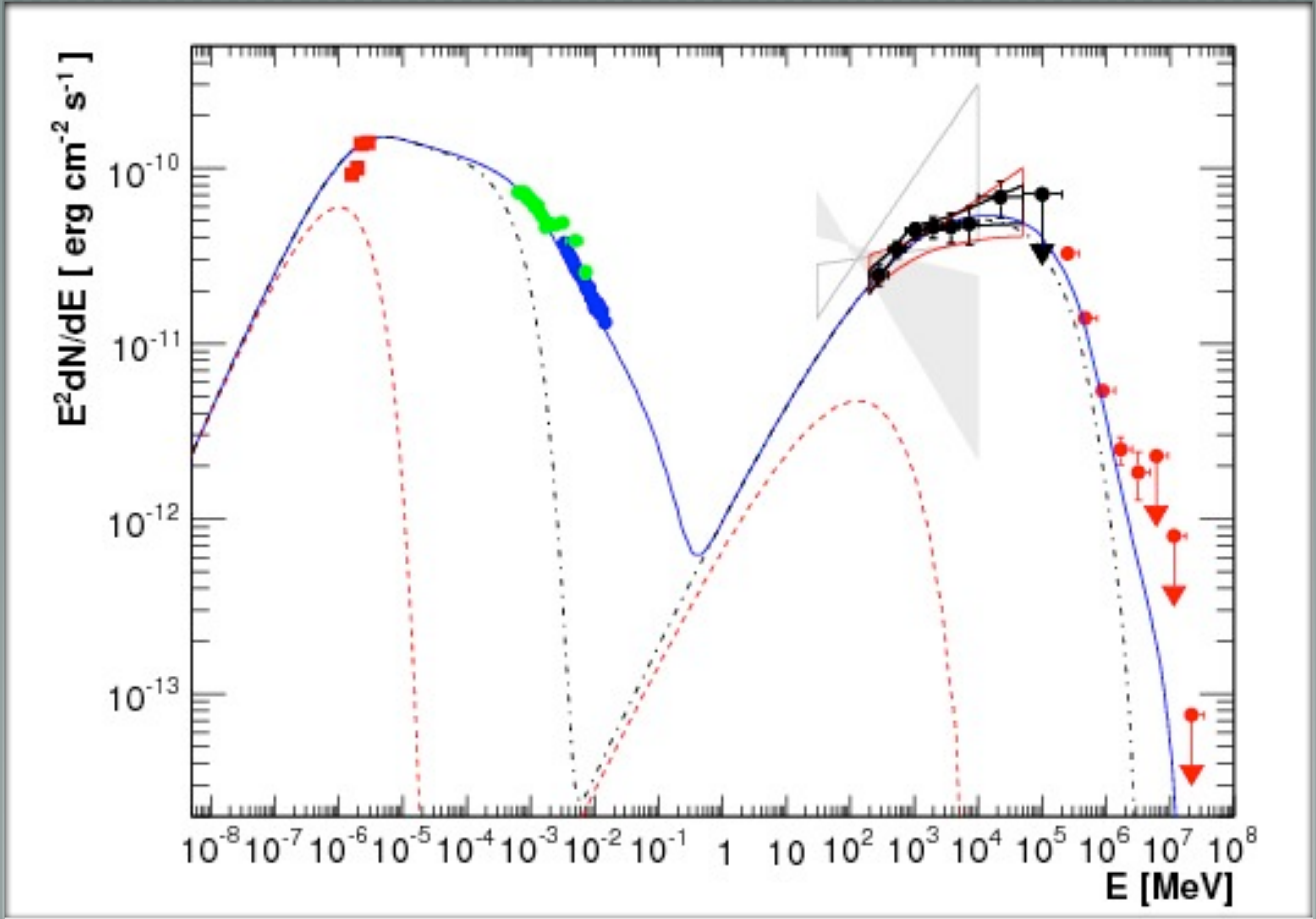
$y \sim 10^{1.15+1.5\cdot2-8} \sim 10^{1.15+3-8} \sim 10^{-3.8}$

Why do we want to do MWL fitting?

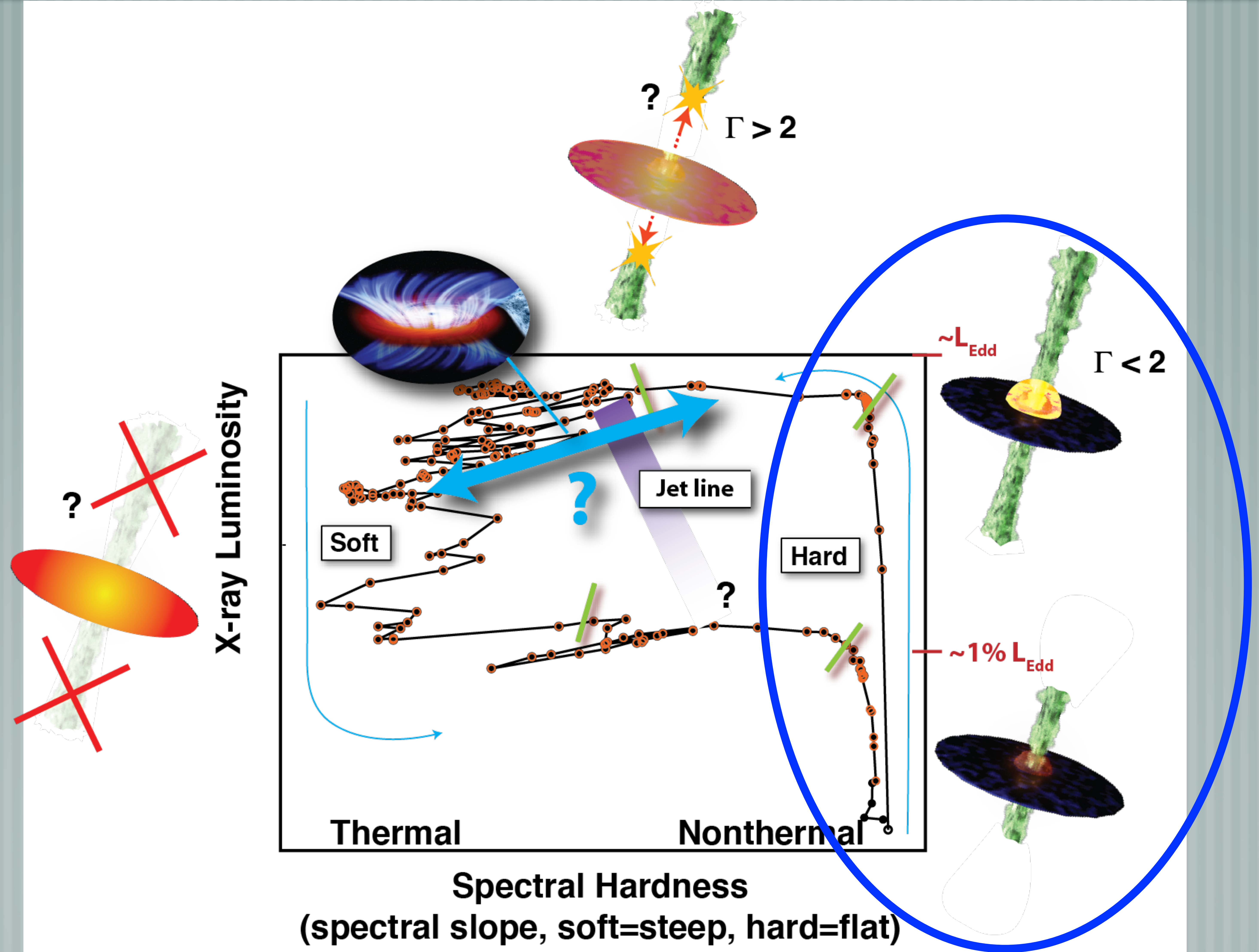
Most accretion powered astrophysical objects radiate over multiple wavebands

- Either via the same process, or interlinked processes like synchrotron and synchrotron self-Compton (SSC)
- To understand and constrain the source physics, we need to fit over the "broadband" spectrum, which can, in the most extreme cases, extend from radio through the gamma-ray

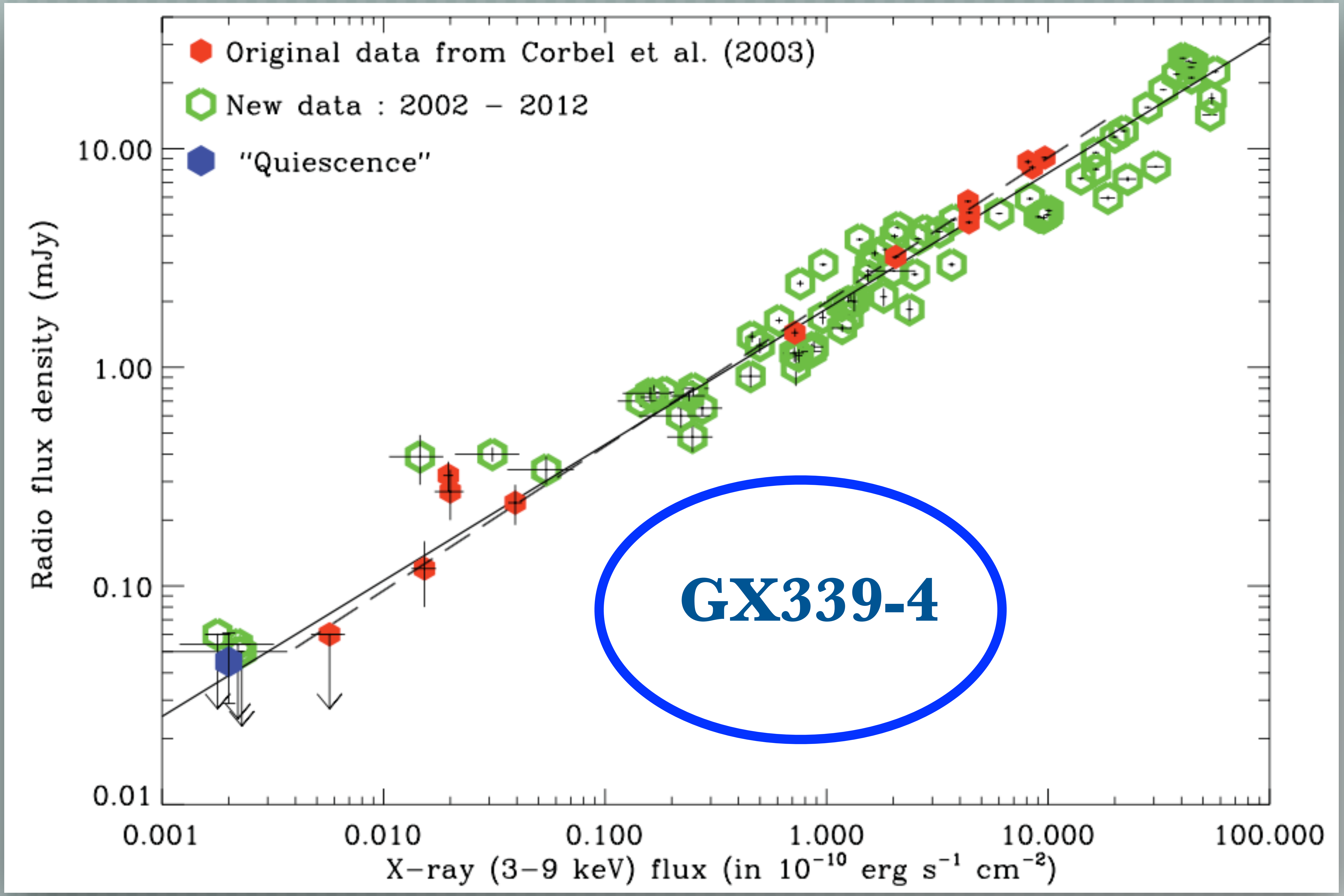
MWL dependence: blazar synchrotron + SSC

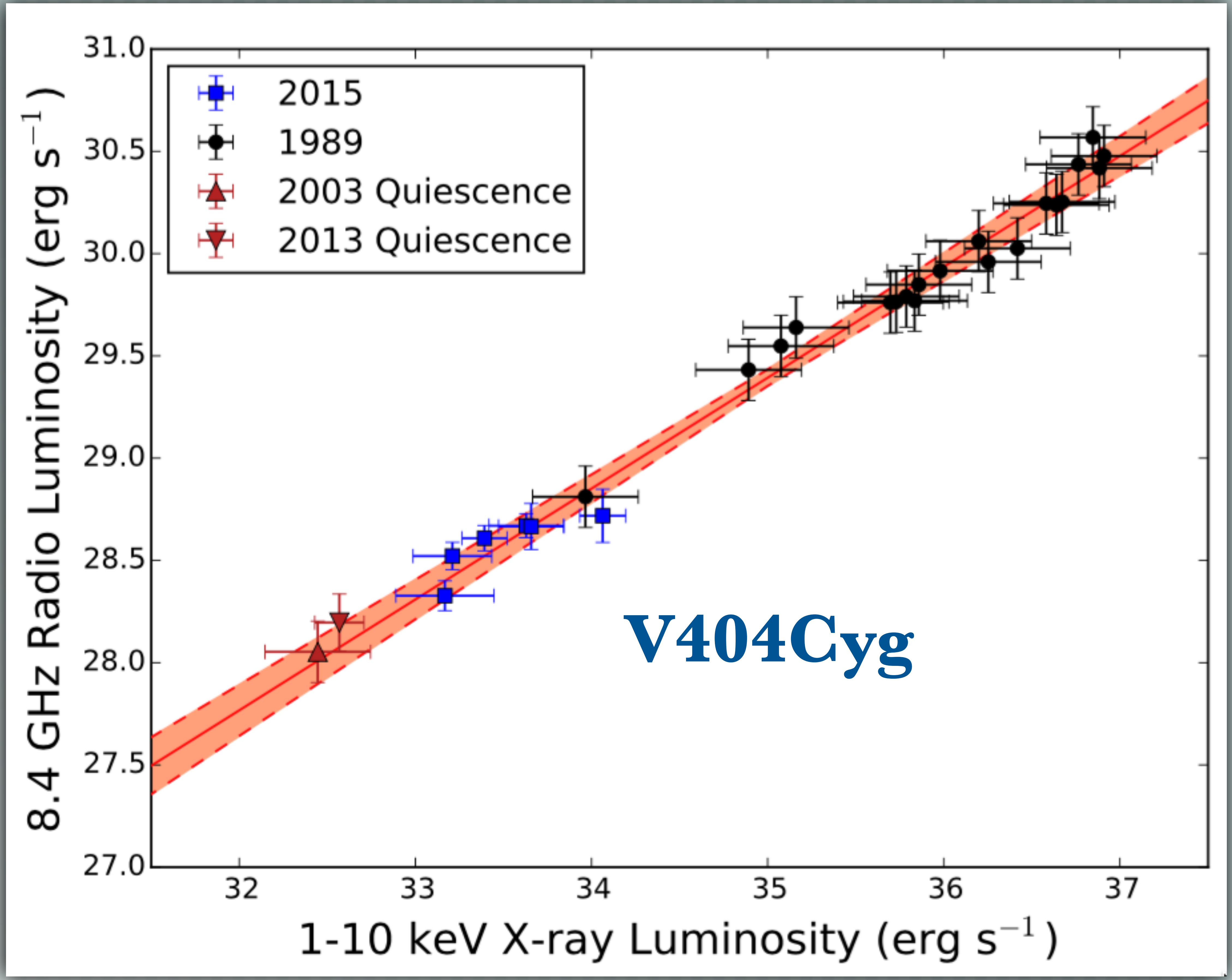


Today: performing a real black hole fit: GX339-4



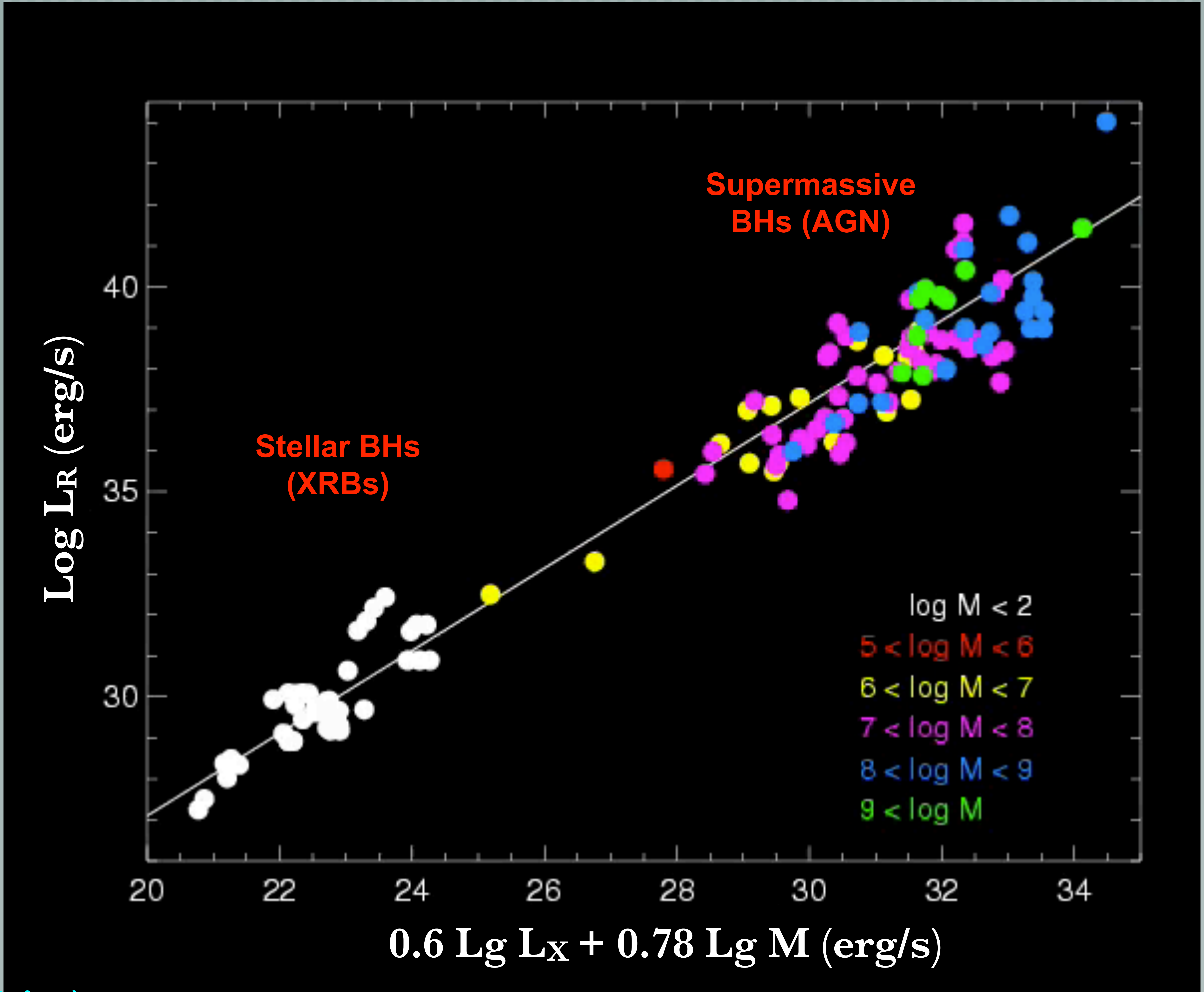
Lec I: Black hole XRBs have “built in” radio/Xray coupling





(SM++01,03,05; Corbel++2008; Hynes++2009; Corbel++2013; Rana++2016, Plotkin++2016)

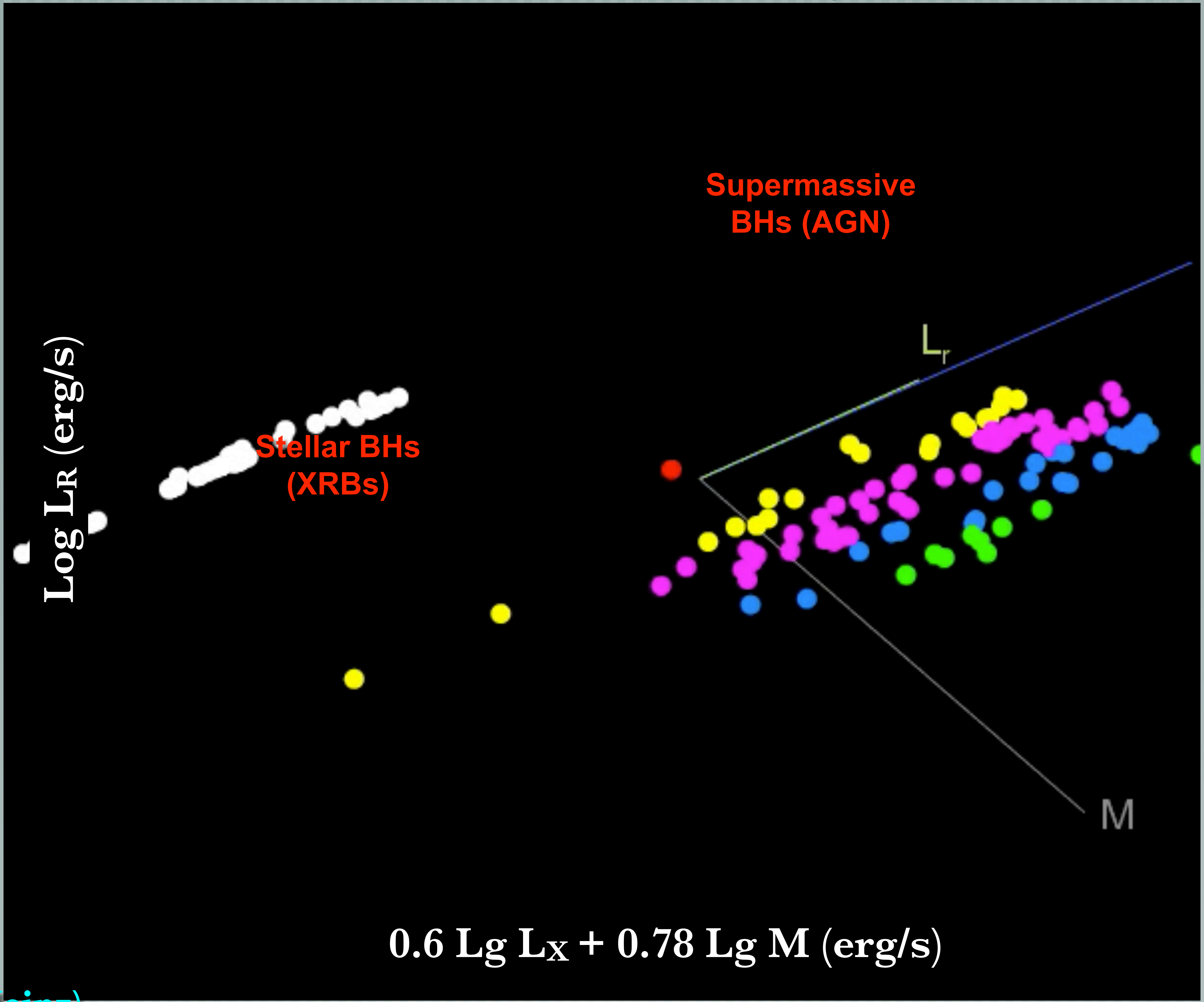
Fundamental Plane of Black Hole Accretion: connecting (low $\dot{M}$) black holes of all masses



(SM et al. 2003; Merloni,Heinz & diMatteo 2003; Falcke, Körding & SM 2004; SM 2005; Merloni et al. 2006; Körding et al. 2006; Gültekin et al. 2009; Plotkin, SM et al. 2011)

(movie courtesy of S. Heinz)

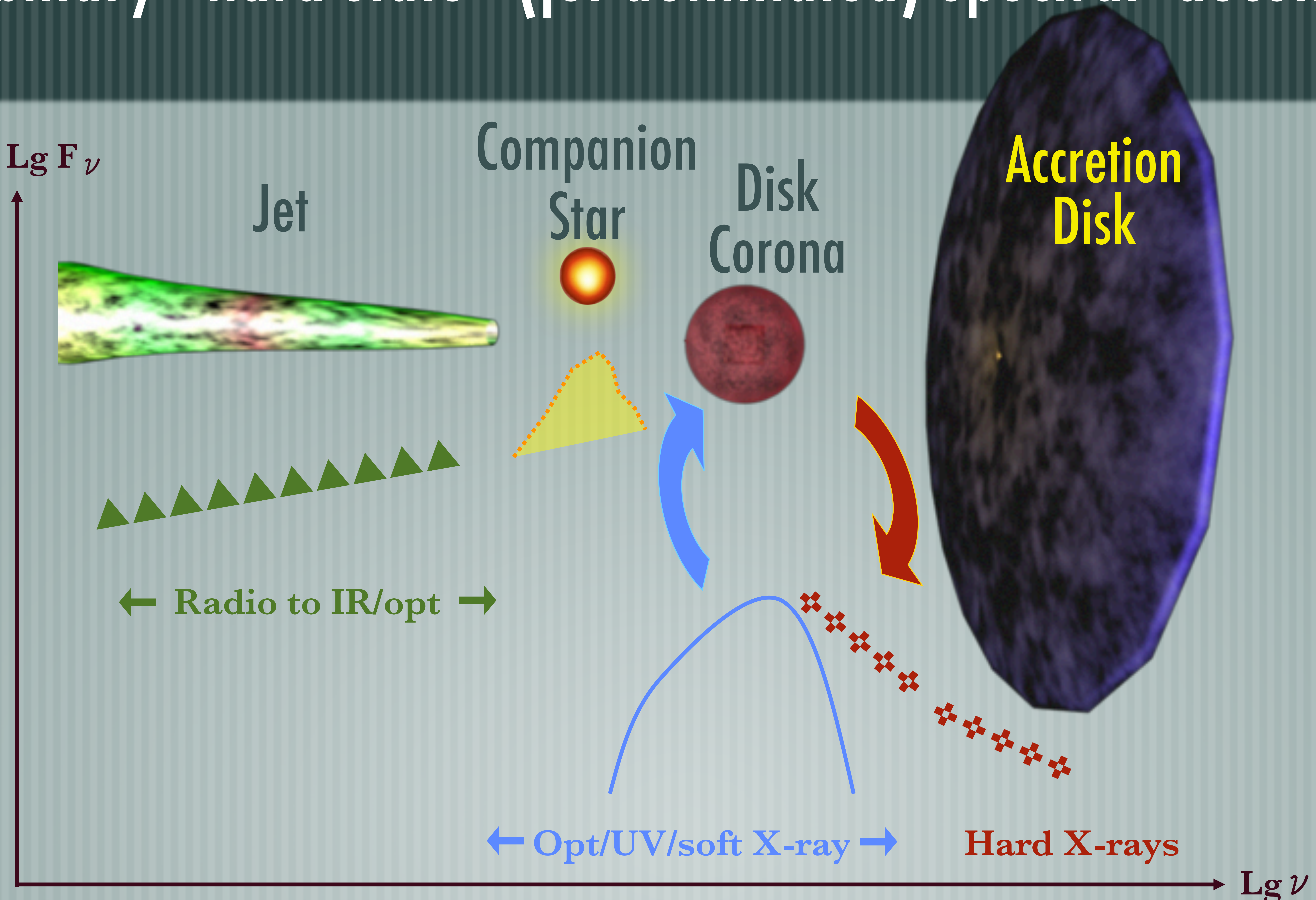
Fundamental Plane of Black Hole Accretion: connecting (low $\dot{M}$) black holes of all masses



(SM et al. 2003; Merloni,Heinz & diMatteo 2003; Falcke, Körding & SM 2004; SM 2005; Merloni et al. 2006; Körding et al. 2006; Gültekin et al. 2009; Plotkin, SM et al. 2011)

(movie courtesy of S. Heinz)

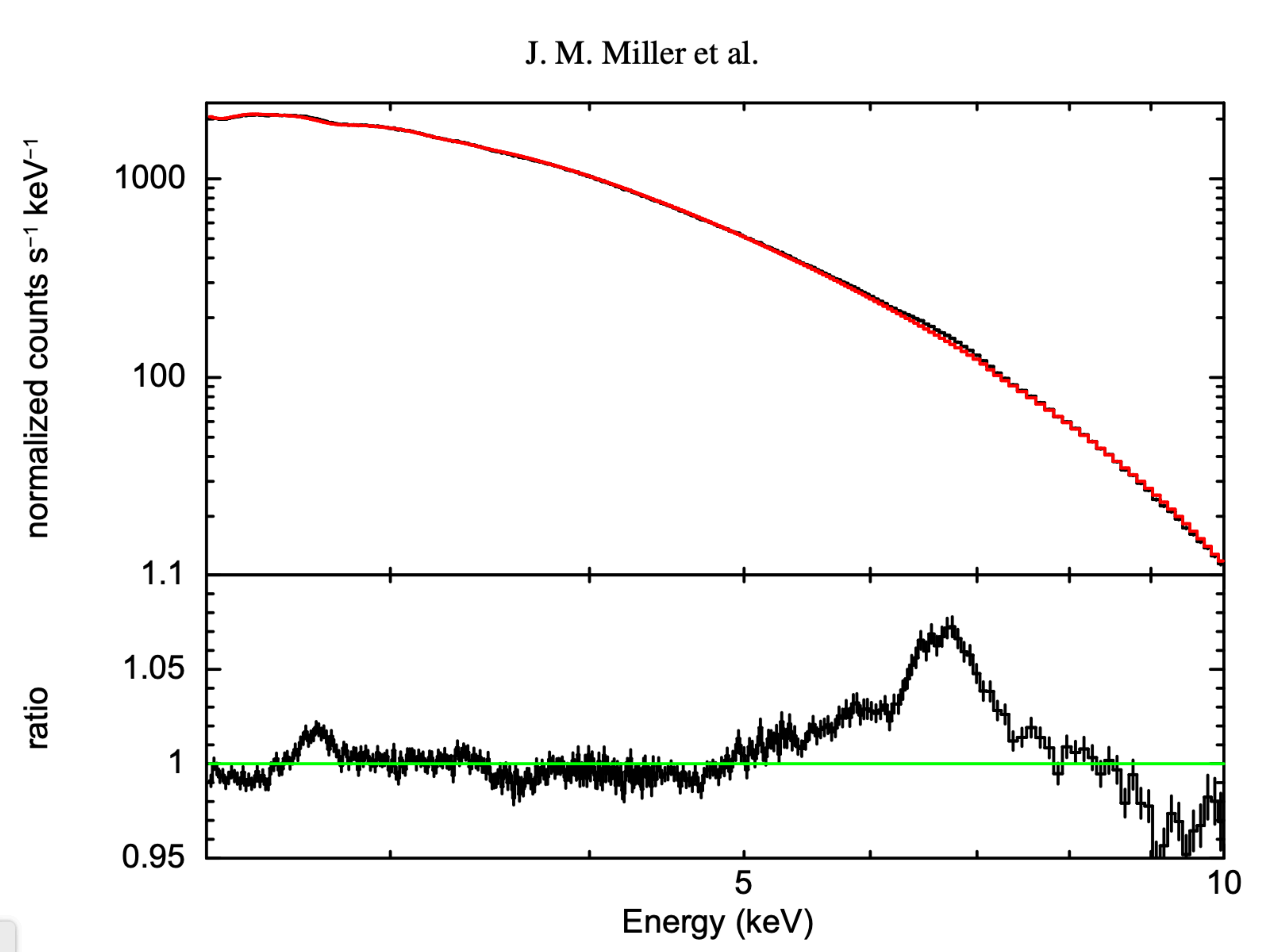
X-ray binary "hard state" (jet dominated) spectral decomposition

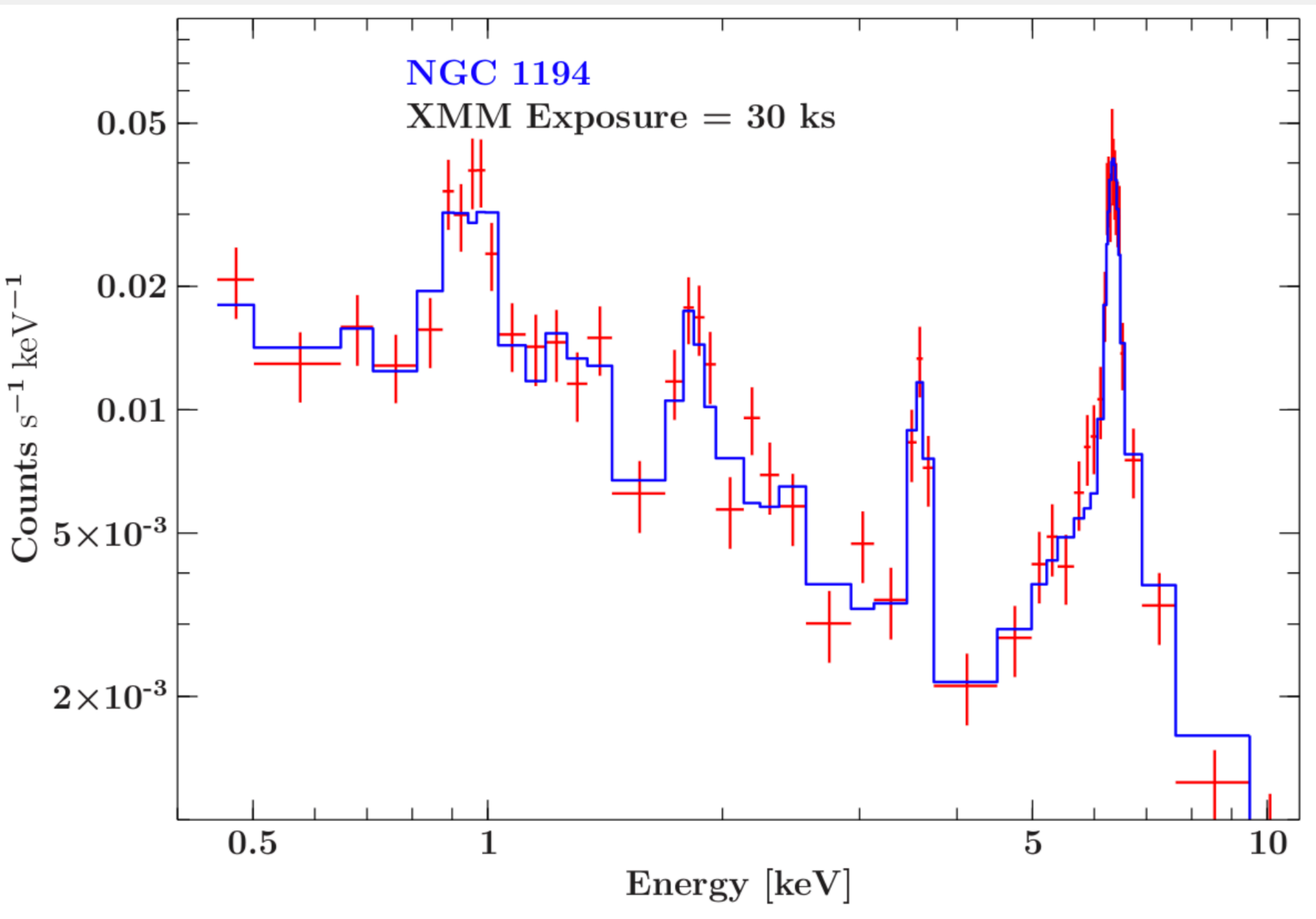


Why can’t you just fit MWL data the same way?

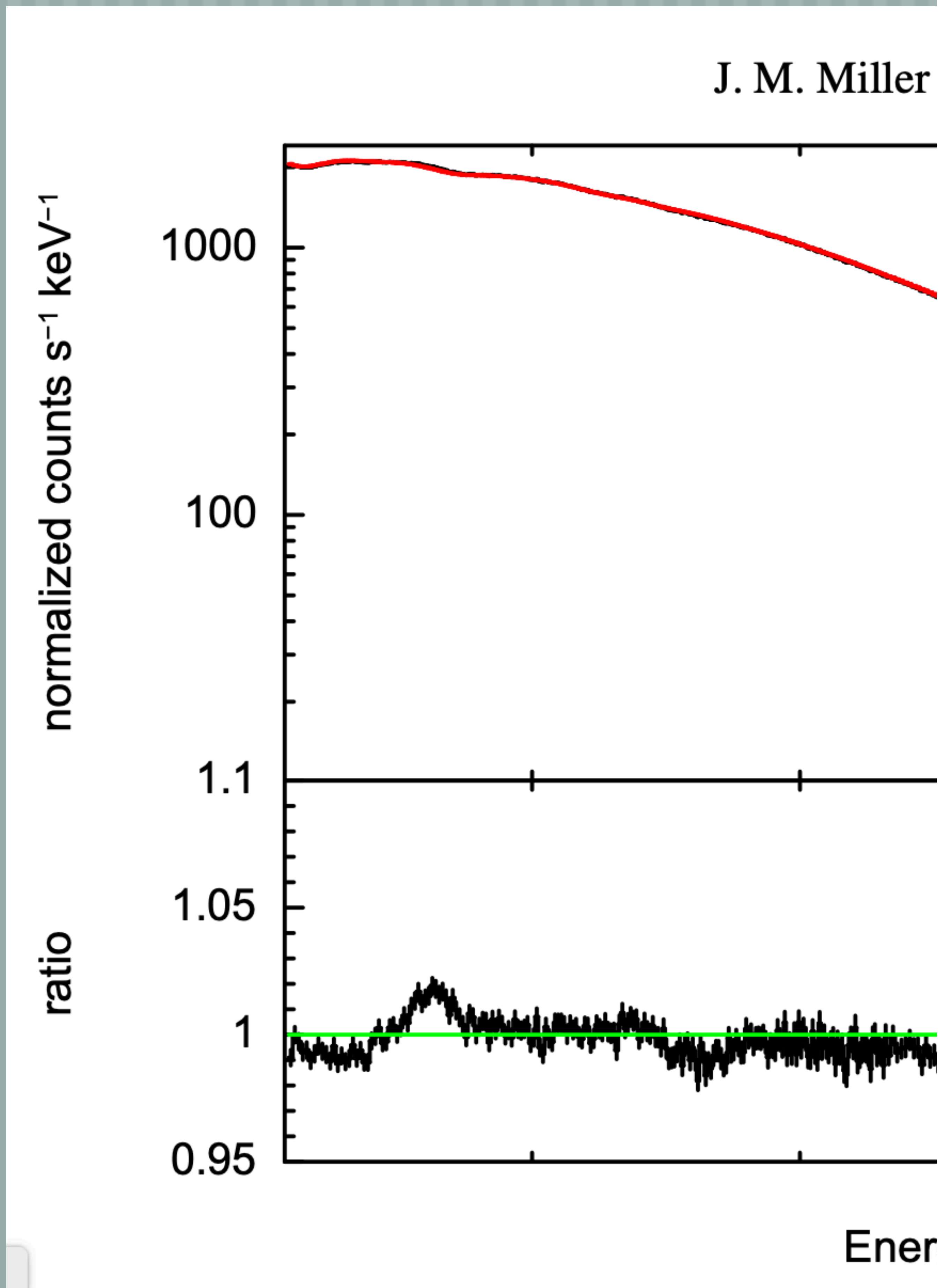
- Unlike radio and optical/IR data, the way photons interact with the detector needs to be accounted for in the modeling
- Means you can’t ‘extract’ data points with error bars without *already* assuming a model, which you don’t want to do!
- The proper way to model X-ray and in some cases gamma-ray data is to “forward-fold” the model into “detector space”, basically you simulate what the detector would see
- Need specialised software with knowledge about the detectors! Currently: Sherpa, Xspec, Isis are the main ones

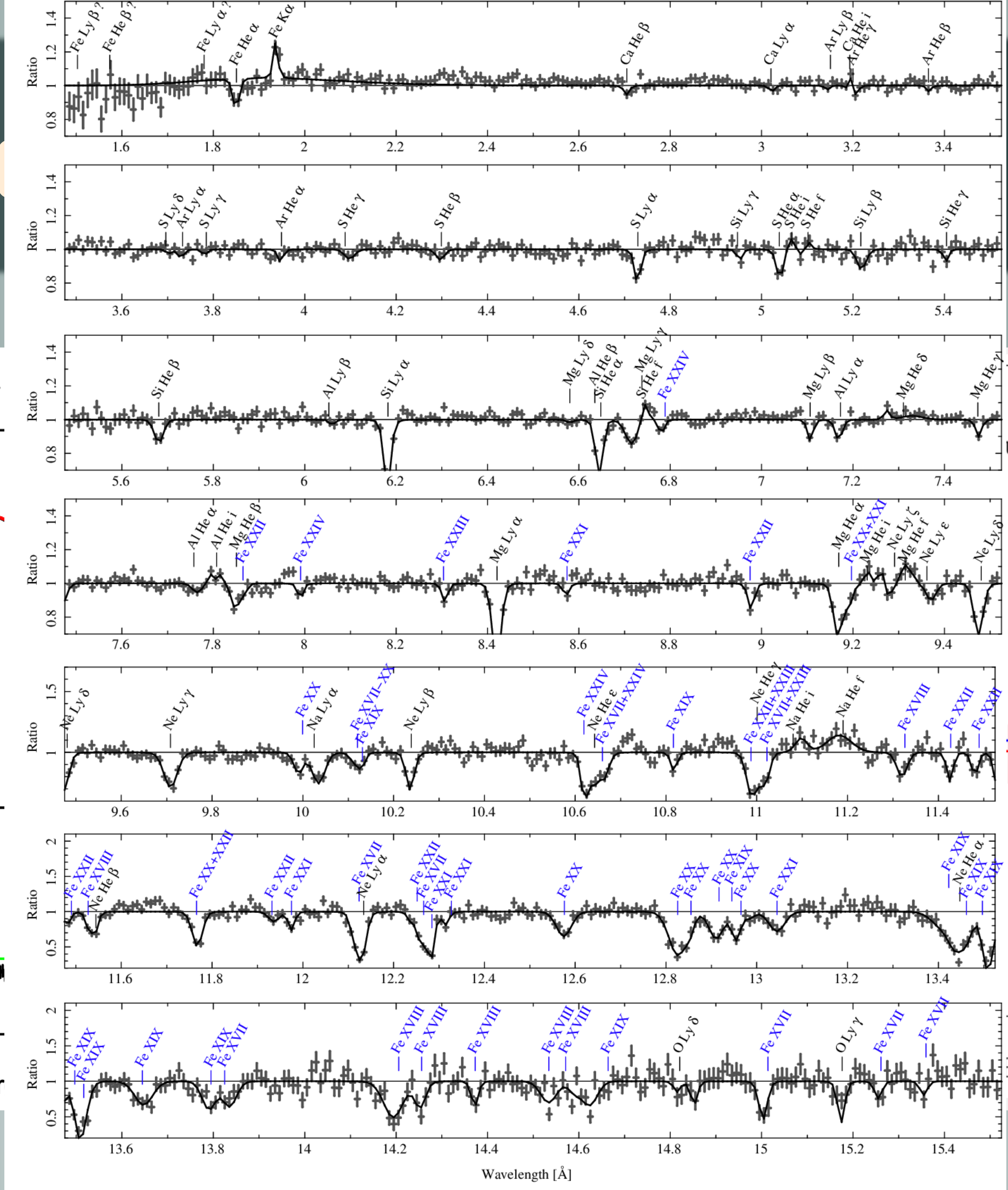
Essential for, e.g., black hole spin, outflows

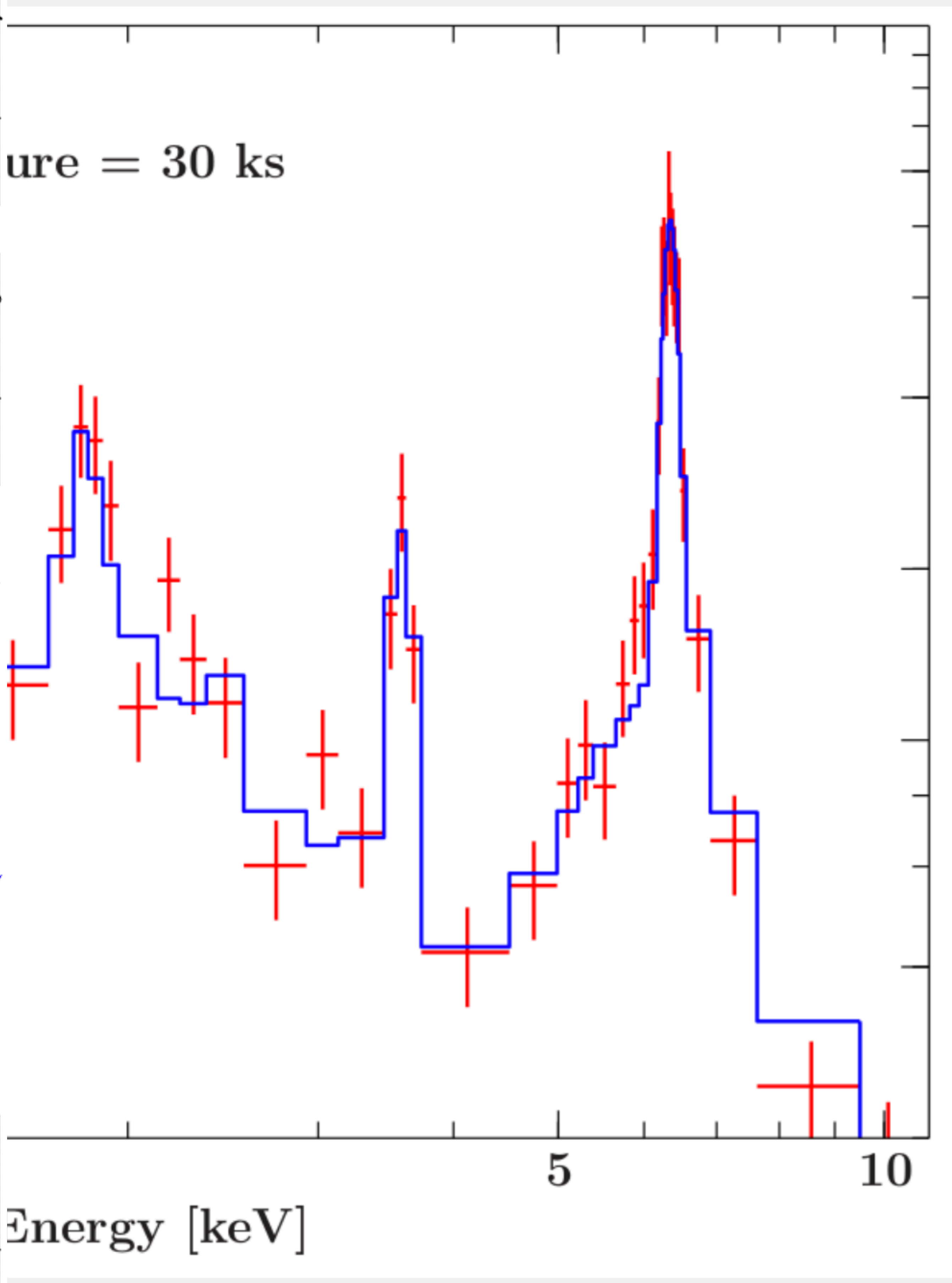




Essenti utflows







Why Isis?

- It's not well documented, it's in a 'niche' language (s-lang) that was very cult among developers in the 90s (…)
- It is not particularly "user-friendly"
- BUT, once you learn it, Isis is very programmable and flexible, if you can think of it, it can likely be done
- It's the only code that you don't have to recompile to include your own custom models: key for theoretical modelling!!!
- It's trivial to combine X-ray with radio/optical/IR data provided via an ASCII file

What we're going to do today

Go step by step through a basic "showcase" for how to use Isis, if you want to learn more see the following resources & tutorials:

- http://www.sternwarte.uni-erlangen.de/isis/
- https://space.mit.edu/cxc/isis/
- https://www.sternwarte.uni-erlangen.de/wiki/index.php/Isis_tutorial
- https://www.sternwarte.uni-erlangen.de/wiki/index.php/Category:Isis_/_Slang

Getting started with Isis

If you would do this on your own, you would use the links on the prior pages to:

1. Carefully read the install instructions, which involves establishing a compatible set of downloading the massive heasoft libraries which includes xspec, downloading and installing slang, gsl, etc, and building step by step until you build isis
2. Download the isisscripts from Remeis Observatory and install them, set up your .isisrc file to find them, as well as setting folders where you put the other models you install later
3. We've done most of this on the VM but you will have to do a couple steps still to get things set up

Getting started: preparing tutorial on VM

- Log onto VM, git pull to update ATAmaterials/day11
- In VM *home* directory edit file *.isisrc* to add line:
  Minimum_Stat_Err = 1.e-30; at end, then:

```
> cd /data/mwl_tutorial
> rm *simplejet*
> mv /data/ATAmaterials/day11/simplejet.tar.gz .
> tar xvf simplejet.tar.gz
>sudo apt install libgsl-dev
```

- Have editor handy for opening load*.pl script files for fast track running commands you'll see here

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

- Any model written as a C/C++ or Fortran subroutine can be SLIRP'ed into Isis, i.e., converted into a S-lang module
- If you want the model to "talk" to Xspec as well, 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, synchrotron only HD jet model, "simplejet" written in fortran from my phd time!

```
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 /home/ataschool/.isisrc add the lines:

```
prepend_to_isis_load_path(“/home/ataschool/data/mwl_tutorial”);

prepend_to_isis_module_path(“/home/ataschool/data/mwl_tutorial”);
```

Loading RXTE PCA data (set 1)

- Now run Isis (>isis) and then load the PCA data. Can call variable anything you want, just remember!
- Set .005 systematics on PCA and group min S/N=4.5, "notice" 3-22 keV

```
variable datapath = "/data/mwl_tutorial/";
variable pid = load_data(datapath+"pca.pha");
set_sys_err_frac(pid,0.005);
group(pid;min_sn=4.5,bounds=3.,unit="kev");
% why min_sn=4.5? Chi2 stat per bin -> need min 20-25 counts/bin
% -> Poisson -> Chi2 stat, while Poisson uncertainty per bin
% S/N = S/sqrt(S) = sqrt(S) -> sqrt(20)=4.5
xnotice_en(pid,3,22);
list_data;
```

Plotting

- Plotting is usually via pgplot which is a bit kludgy, can also hack to use xfig
- To select color/symbols, see: `pg_info;`

Plotting "folded" RXTE PCA data (set 1)

Plot the data first folded with detector features

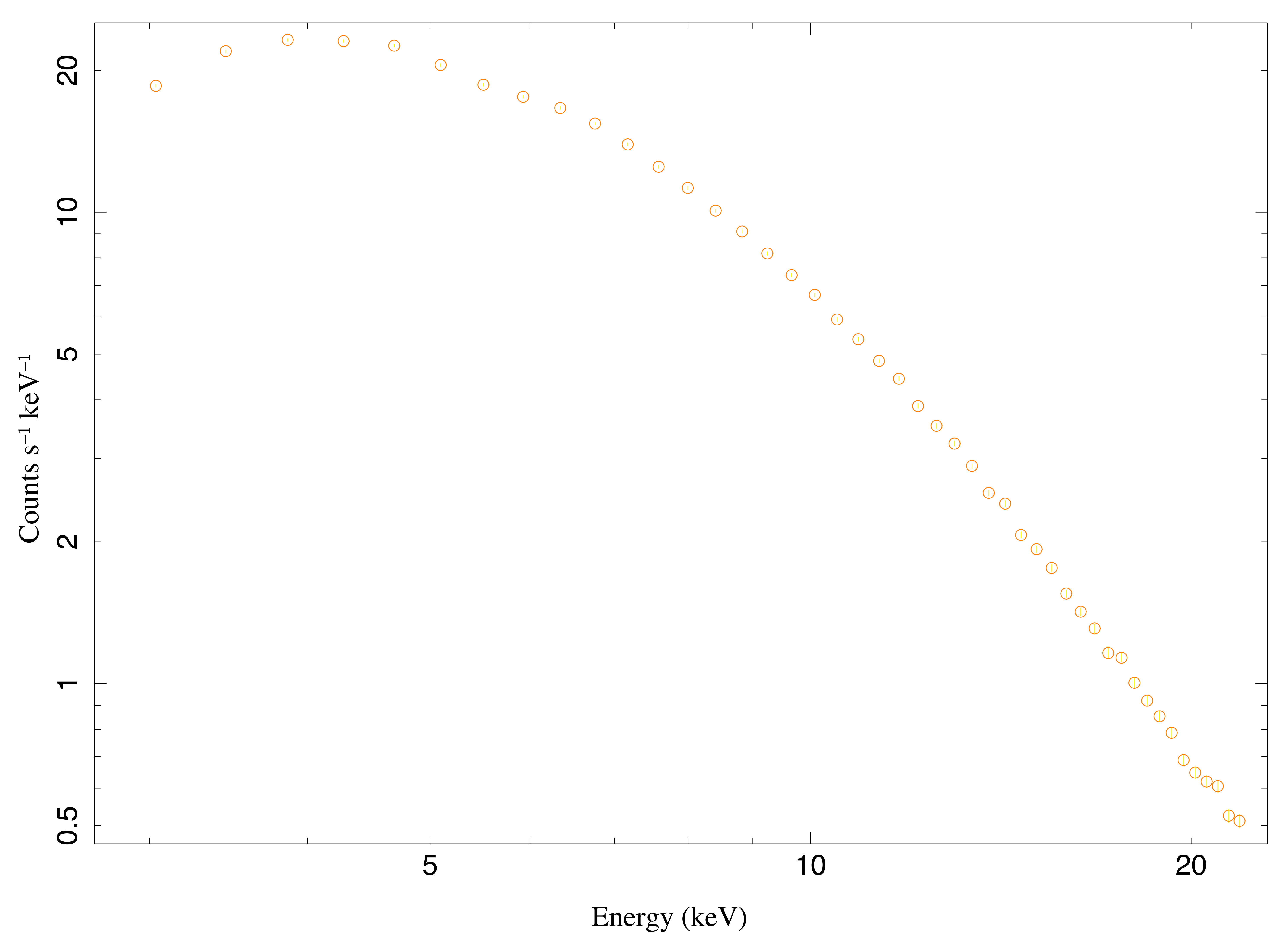
To learn about units, folding/unfolding data see also Remeis Tutorials and Mike Nowak's lectures from our summer school 9 (!!) years ago:

http://www.black-hole.eu/media/summerschool2/X-ray_Spectra_Part_I.pdf

http://www.black-hole.eu/media/summerschool2/X-ray_Spectra_Part_II.pdf)

```
xlog;ylog;d_width=5;set_plot_widths(;m_width=5, r_width=5, re_width=5,
d_width=5, r_width=5, re_width=5);
% different ways to plot data:
plot_counts(pid;dsym=4,dcol=8,decol=8); % Counts / bin
plot_data(pid;dsym=4,dcol=8,decol=8); % Counts / s / keV
% spectrum folded with detector features: intrinsic spectrum folded by
% detector response, i.e., effective area [how sensitive at what energy due to
% mirror effects etc], redistribution function [what channel belongs to which
% energy] Advantage: see detector features (bumps around absorption edges,
% less counts at edges of energy range (see shape effective area)
```

Data folded still w/PCA detector

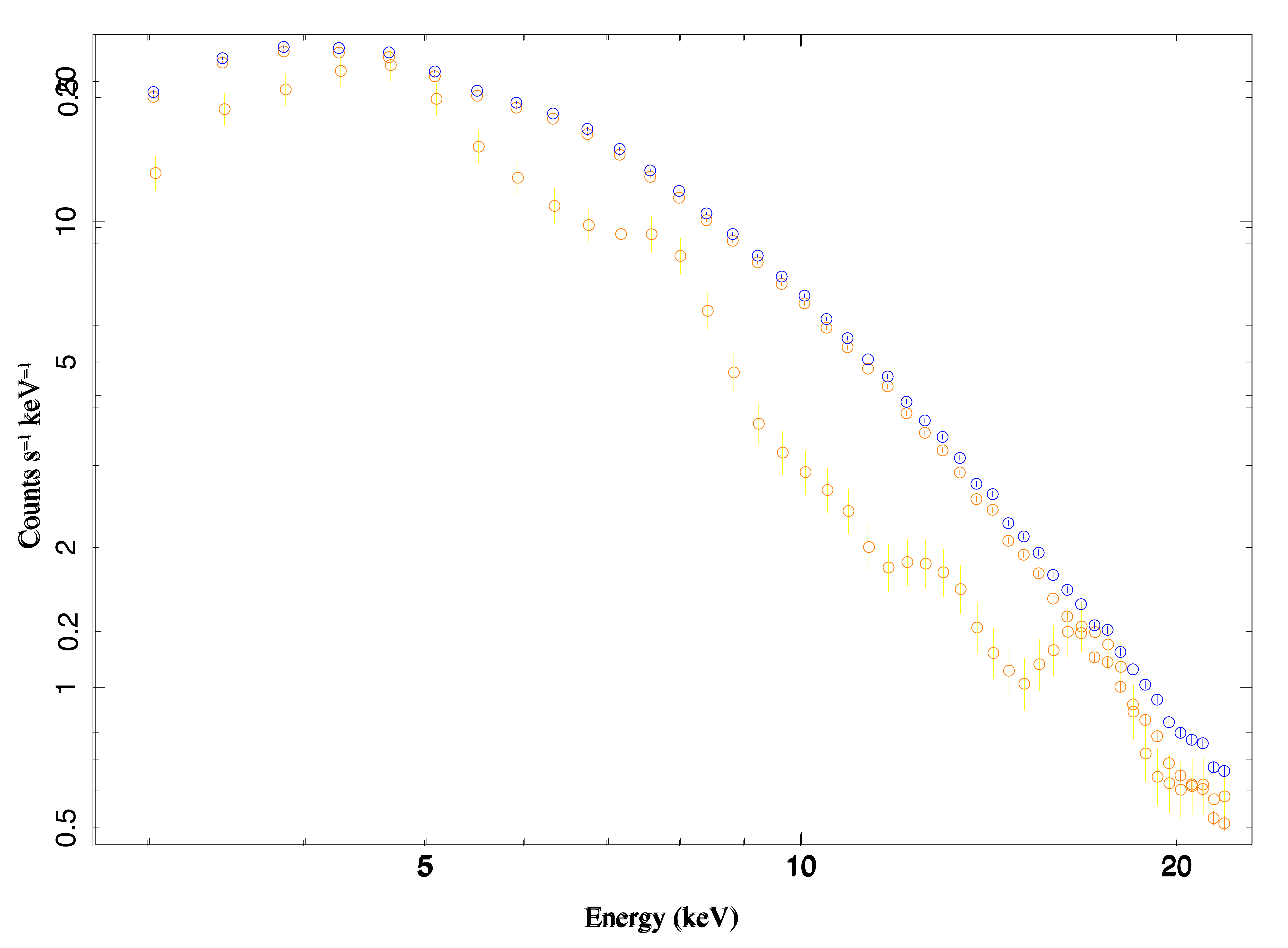


Plotting the background

- Plot the background emission only, or overplot with and without background

```
plot_data(pid;dsym=4,dcol=8,decol=8,bkg=-1);
% bkg=0 is the ‘default’ so we didn’t specify before but that’s with background subtracted off
plot_data(pid;dsym=4,dcol=4,decol=4,bkg=1,oplt=1);
% we can also overplot data without bkg + data with bkg
```

Data w/ and w/out background

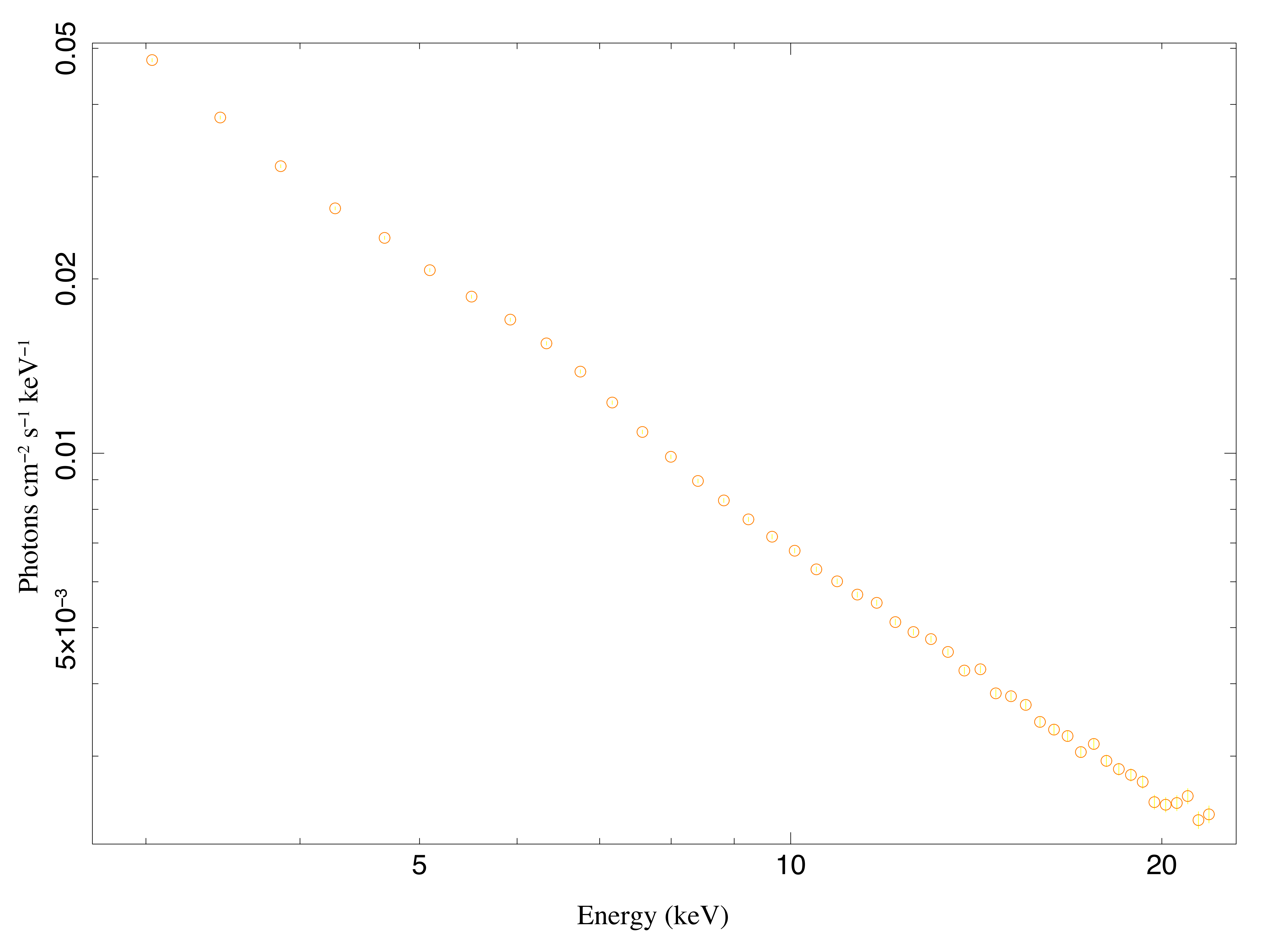


Plotting unfolded RXTE PCA data (set 1)

- Plot the data now 'unfolded' from the PCA detector

```
plot_unfold(pid;dsym=4,dcol=8,decol=8); % Photons / cm2 / s / keV
% spectrum unfolded from detector response -> source-intrinsic spectrum
% Model of intrinsic spectrum only as good as we know detector,
% unfolding not always successful, sometimes still features at
% absorption edges of effective area
% good to get idea what model we want to use (spectral model defined in
% source space of course)
```

Data unfolded from PCA detector

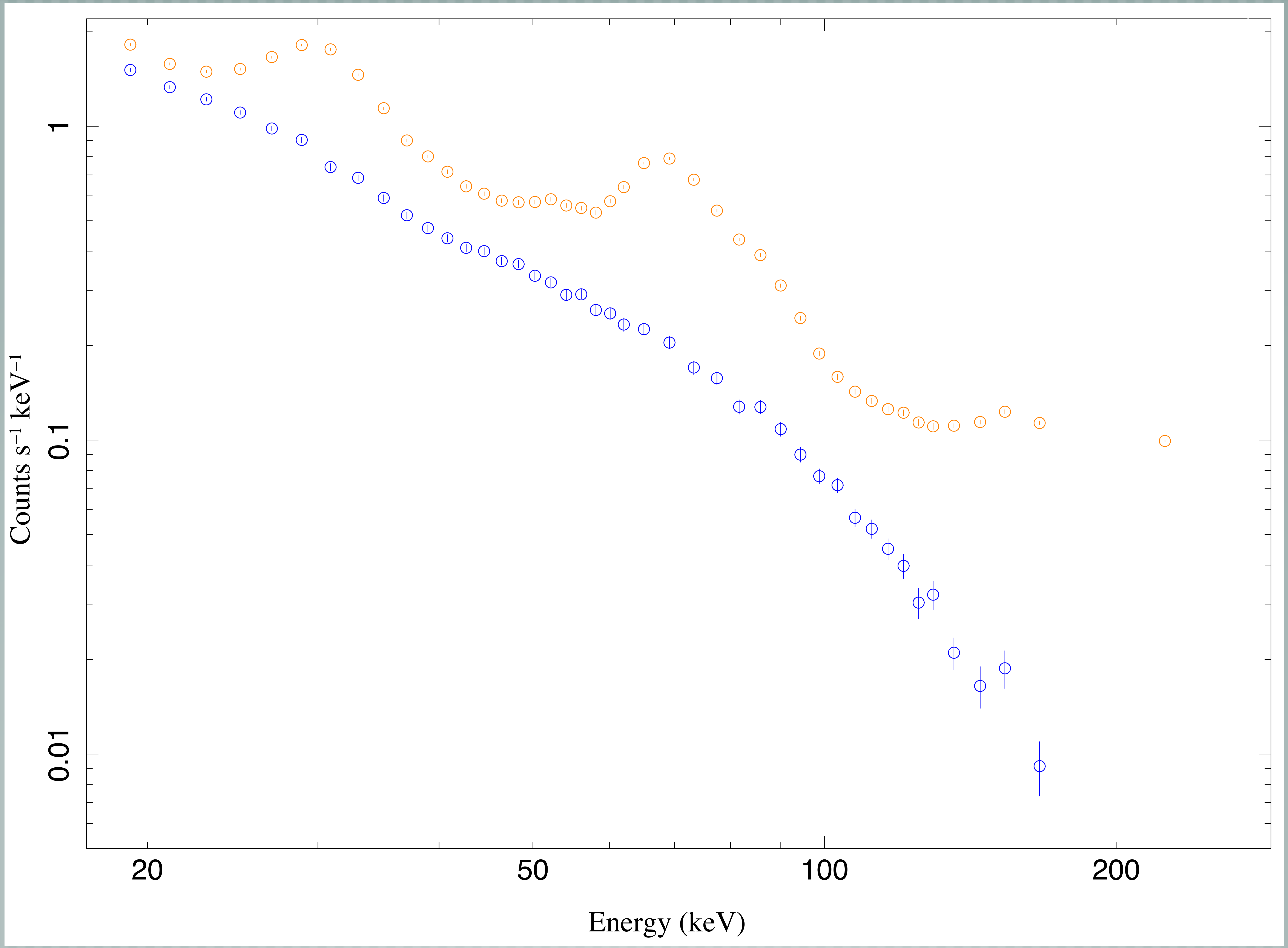


Now let's add HEXTE data also from RXTE

- Treat similarly as PCA, group min S/N=4.5 but "bounds" now 20 keV and we "notice" 20-200 keV
- Make the same two plots (will need to change x/y ranges)

```
variable hid = load_data(datapath+"hxt.pha");
group(hid;min_sn=4.5,bounds=20.,unit="kev");
xnotice_en(hid,20,200);
list_data;
plot_data(hid;dsym=4,dcol=8,decol=8, bkg=1,
yrange={0.005,2.2});
% plot data+bkg
plot_data(hid;dsym=4,dcol=4,decol=4, bkg=0, oplt=1,
yrange={0.005,2.2}); % plot data-bkg
```

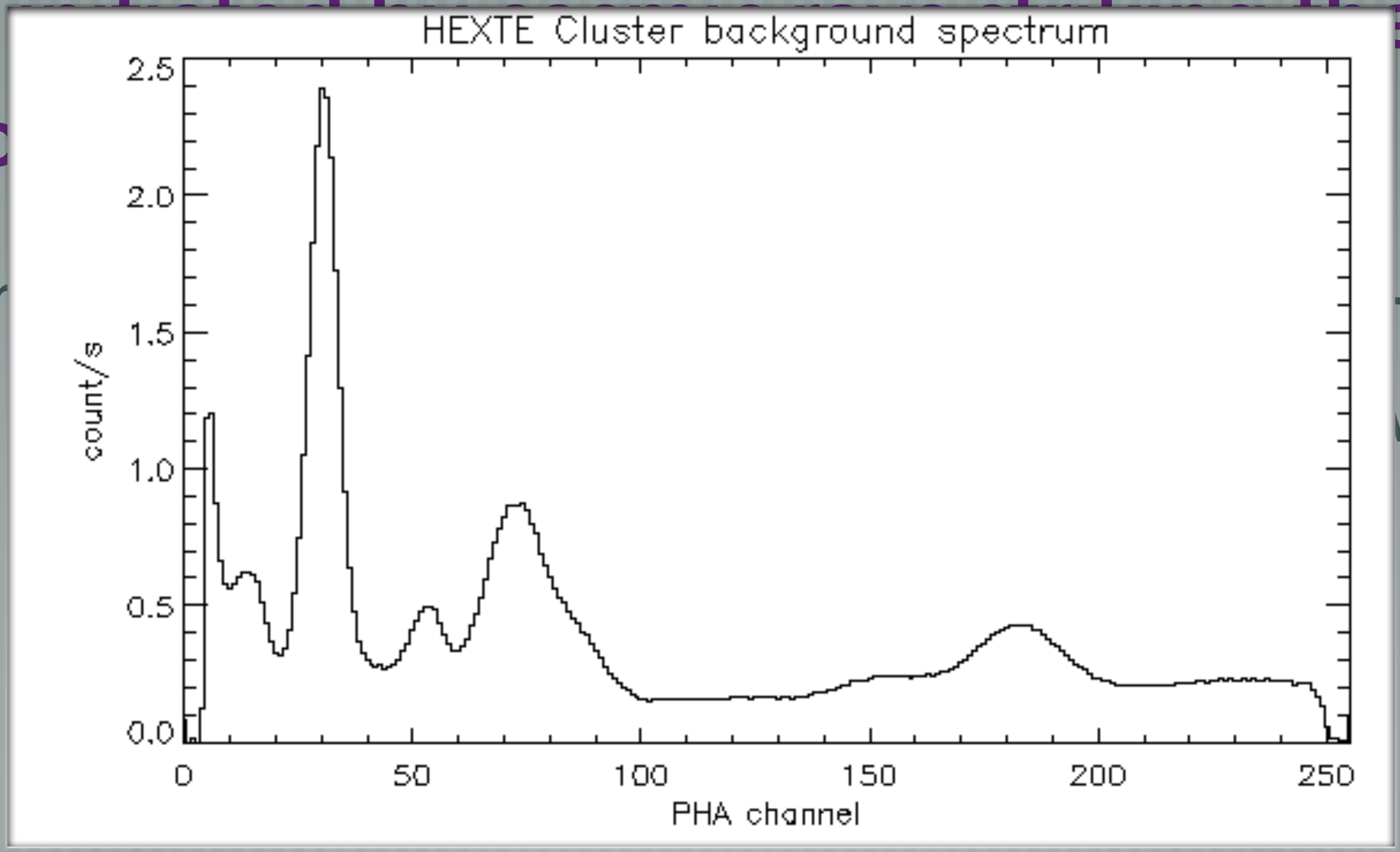
HEXTE background features (set2)



HEXTE background features

(Broad) line features all real, result from radioactive decays initiated by cosmic rays striking the detecto

- 30 keV lir -capture
- Between with iodine, tungsten

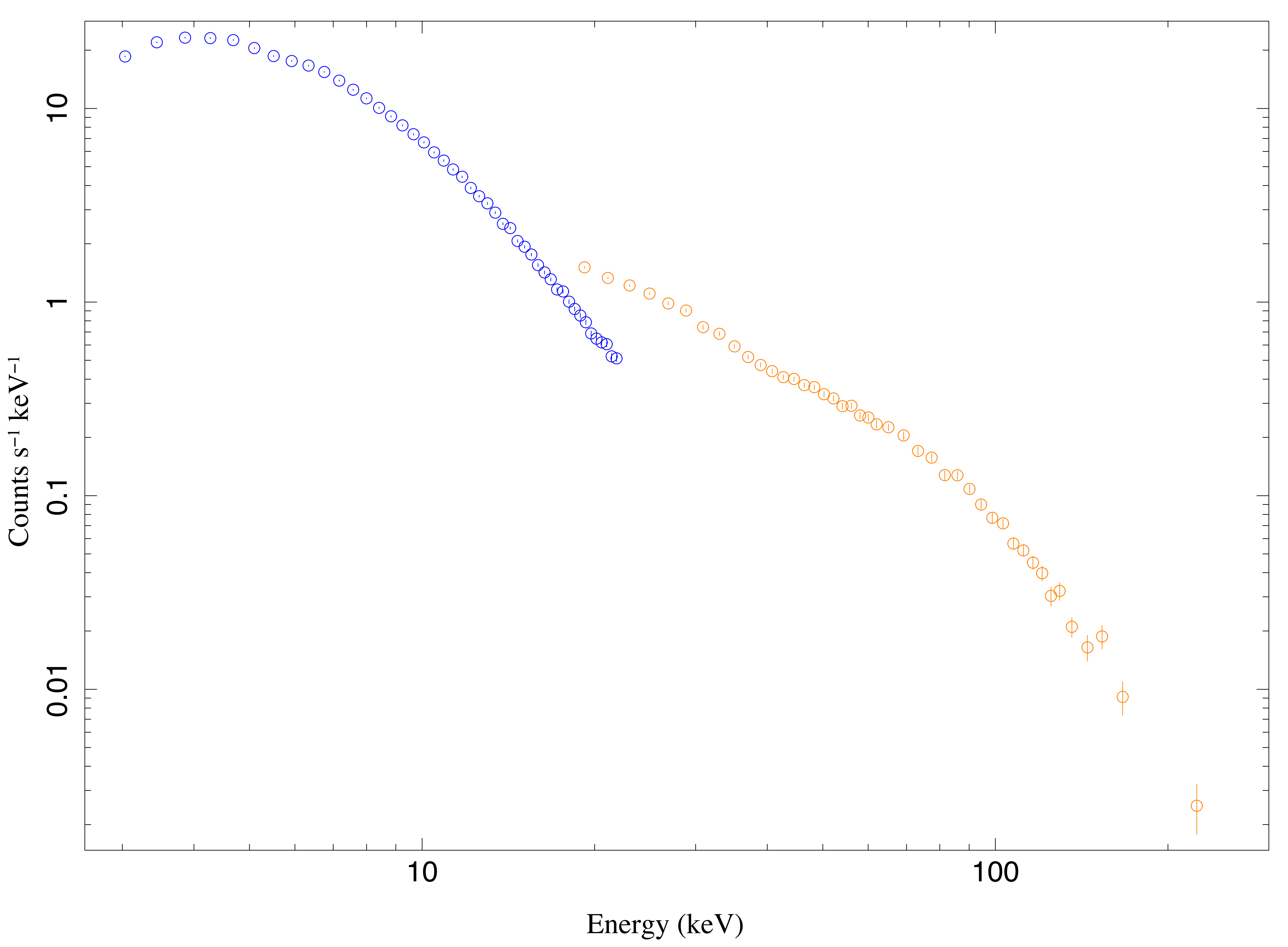


Plot both data sets: folded, no backgrounds

- Now plot both data sets on the same plot, using different colors for each (you've seen this syntax already...). This is in 'detector space'

```
plot_data({pid,hid};dsym={4,4},dcol={4,8}, decol={4,8});
```

PCA + HEXTE in counts/s/keV

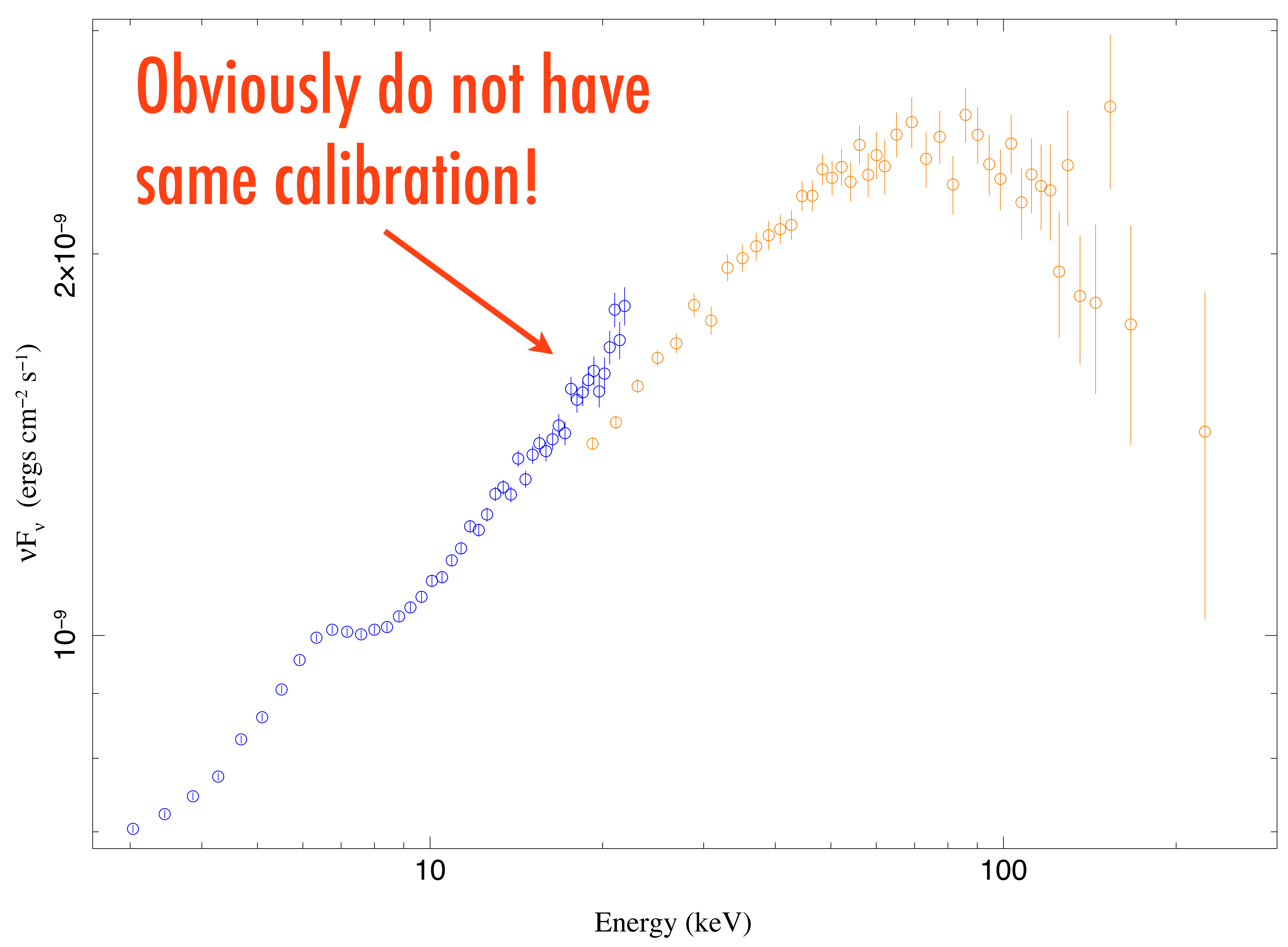


Plot both data sets: now unfolded

- Now with unfolding we can also convert to physical units plots, with units of , e.g., keV and ergs/cm^2/s
- Notice we can now do error bars!

```
fancy_plot_unit("kev","ergs");
plot_unfold({pid,hid};dsym={4,4},dcol={4,8}, decol={4,8});
```

Plot both data sets: unfolded (flux)



Now let's include radio and IR!

- Take a look at the content of the isisscripts: "load_radio2"
- Input file is ASCII, 3 cols: Freq(Hz), Flux (mJy), errFlux(mJy)
- For any data without response files ARF & RMF, assumes RMF is diagonal with values =1, ARF = 1 cm^2 and "exposure" = 1sec. So we convert the input flux into counts/cm^2/s (for compatibility with XSPEC!)

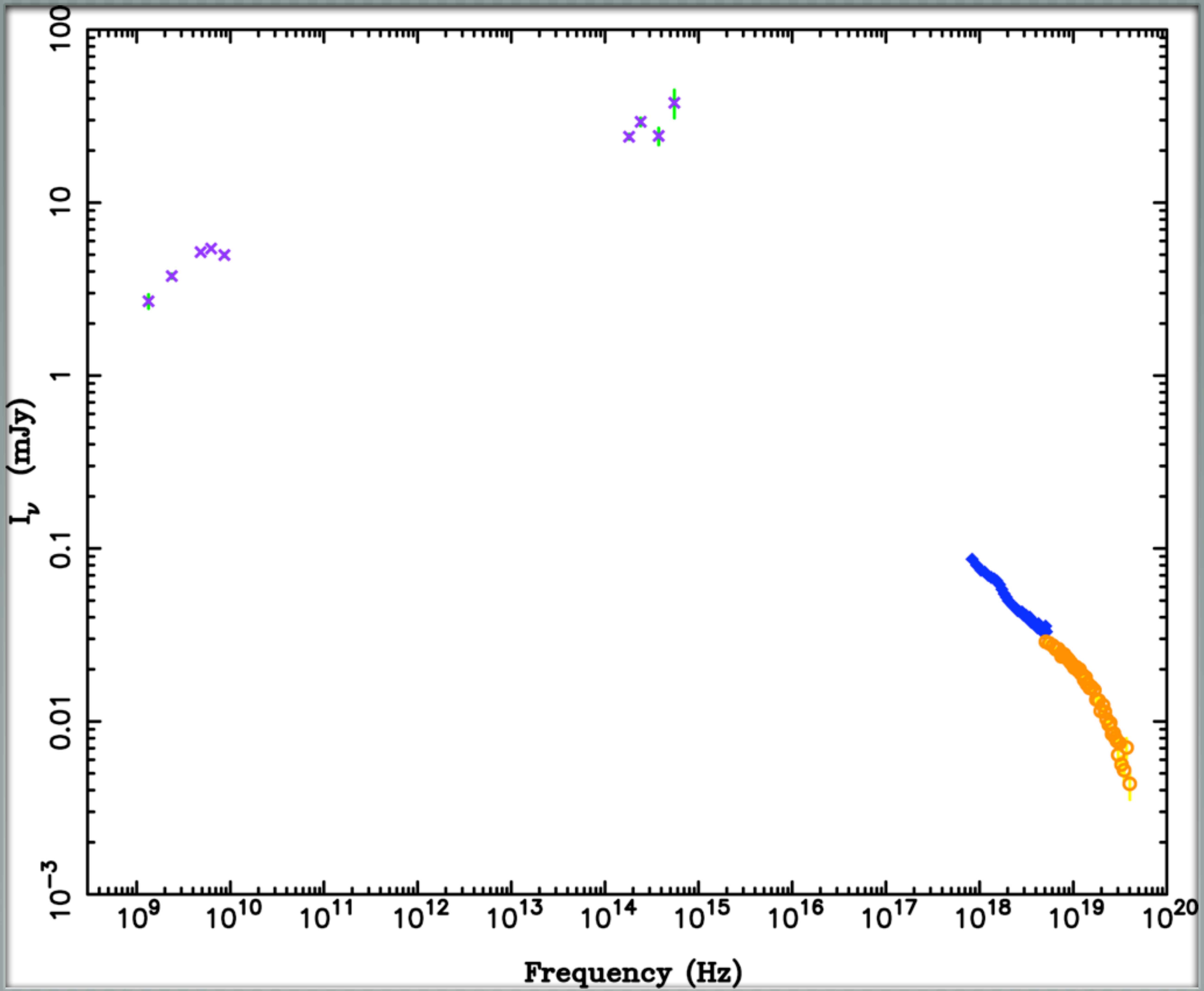
```
rid=load_radio2(datapath+"53082_radio-OIR.dat");
list_data;
```

Now we can plot the entire broadband spectrum

- Let's plot the data in "typical" radio astronomy units for jets. switch x axis units to "hz" and y axis to "mjy"

```
fancy_plot_unit("hz","mjy");
plot_unfold({rid,pid,hid};dsym={4,4,4},dcol={6,4,8},
decol={6,4,8},xrange={3e8,1e20});
```

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 (slow). Gridding tells Isis explicitly what grids to use
- To group data from different instruments to have the same normalization, gridding is essential.
- 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, essential to save time for complex models.

Set radio and PCA to same normalisation

- 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 scripts/webpage for details)

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

First let's ignore the radio though...

- We'll exclude the radio data 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), see linked tutorials to know why

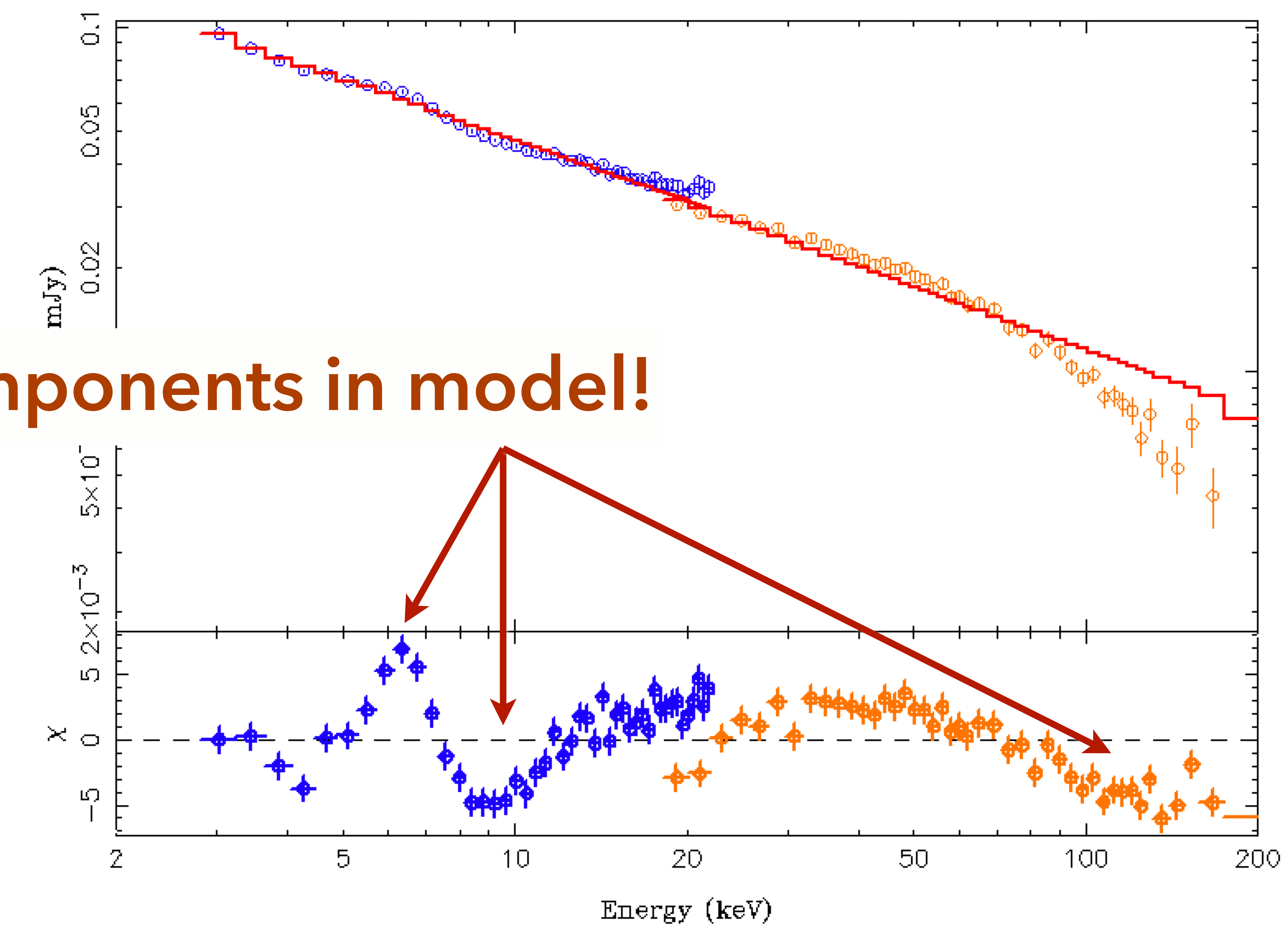
```
exclude(rid);
fit_fun("constant(Isis_Active_Dataset)*phabs(1)*bknpower(1)");
list_par;
```

Optimize model parameters, then fit

- Fix absorption to “typical” value
- Fix constant(1)=PCA normalization to 1, and limit HEXTE constant to 0.9-1.1 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

```
set_par("phabs(1).nH",0.6,1);
set_par("constant(1).factor",1,1);
% freeze detector constant of soft X-rays to 1, hard X_ray dataset free to
vary
fit_counts; fancy_plot_unit("kev","mjy");
plot_unfold({pid,hid};dsym={4,4},dcol={4,8},decol={4,8},rcol={4,8},recol={4,
8},xrange={2,200},res=1);
```

Fitting broken PL + absorption



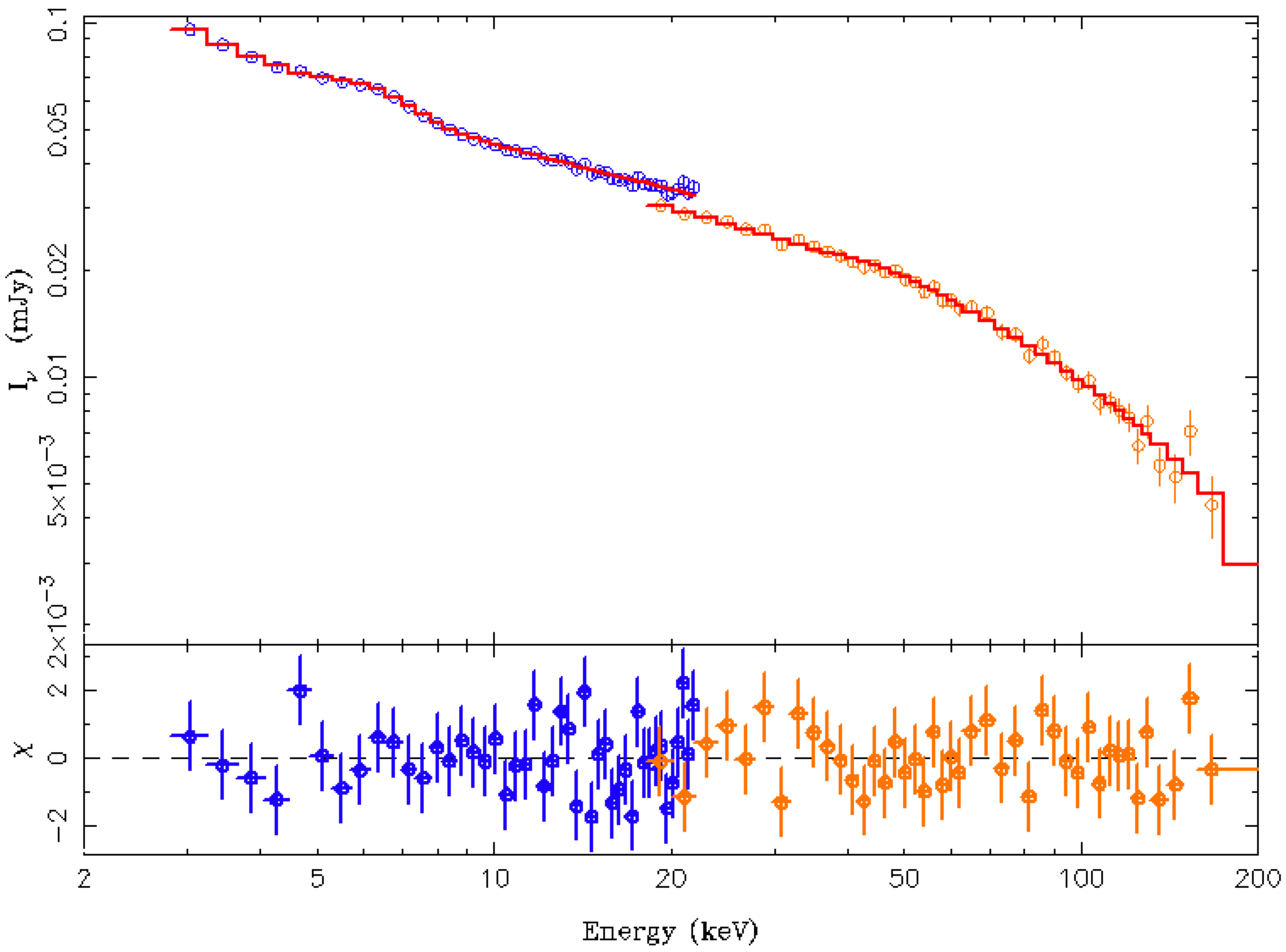
Need more components in model!

Add gaussian line, disk black body & cutoff

- Always check the “vector math” of models. Are each of these additive or multiplicative?
- The cutoff model we’ll use is called “highecut”, it *multiplies* the two *additive* models, similar to the absorption model
- Use spectral features to give good starting values (i.e. disk temperature around 1 keV, cutoff at high energies…)

```
fit_fun("constant(Isis_Active_Dataset)*phabs(1)*highecut(1)*(bknpo
wer(1)+diskbb(1)+ gaussian(1))”);
list_par;
fit_counts;
plot_unfold({pid,hid};dsym={4,4},dcol={4,8},decol={4,8},rcol={4,8}
,recol={4,8},xrange={2,200},res=1);
```

Better fit, but what about more physical models?

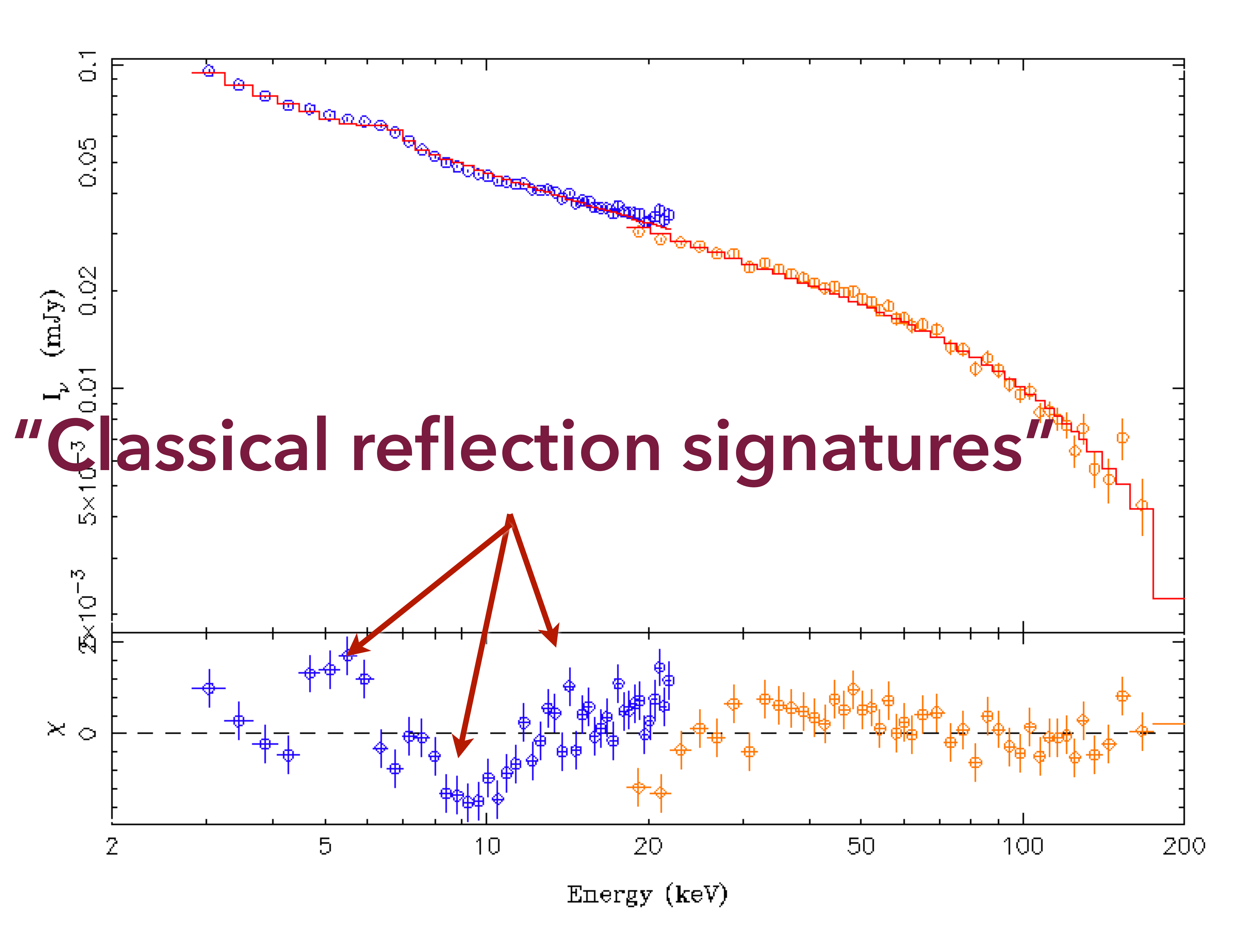


Use real inverse Compton model!

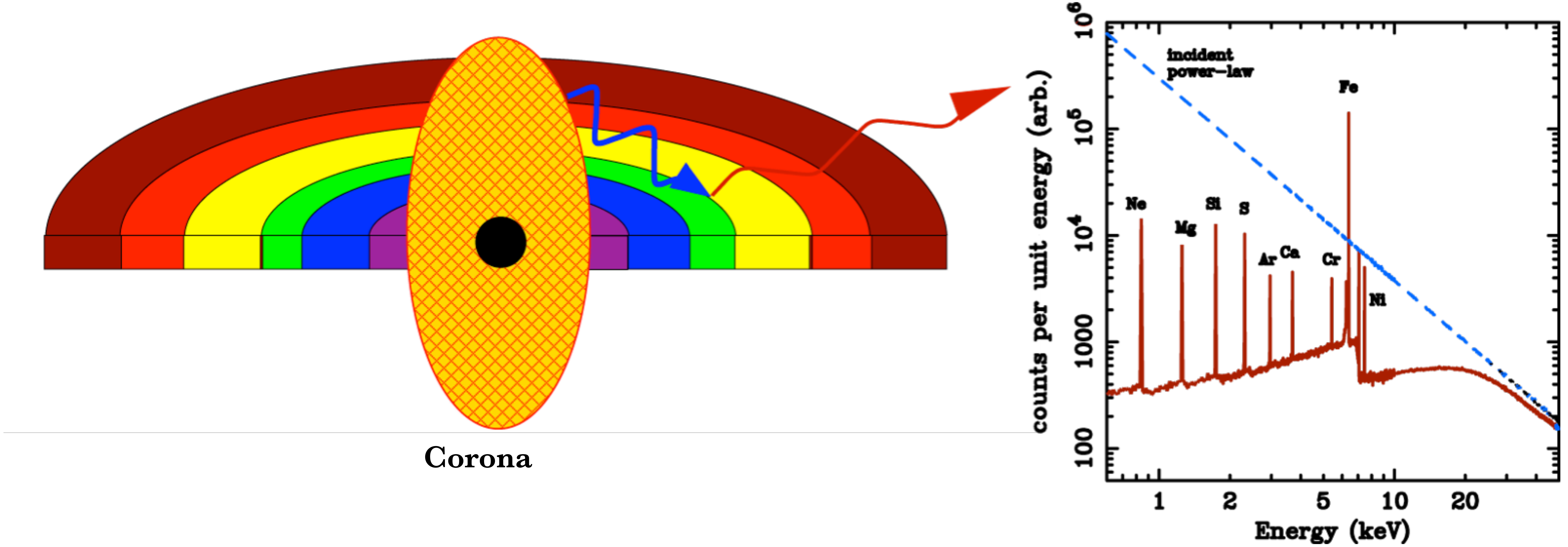
- We see a power-law with a cutoff: this is exactly what we expect from inverse Compton scattering with y≤1.   What does the cutoff represent?

```
fit_fun("constant(Isis_Active_Dataset)*phabs(1)*(gaussian(1)+nthcomp(1))");
list_par;
fit_counts;
```

Not as good a fit (reality is complicated)



Reflection of hard X-rays off cooler disk

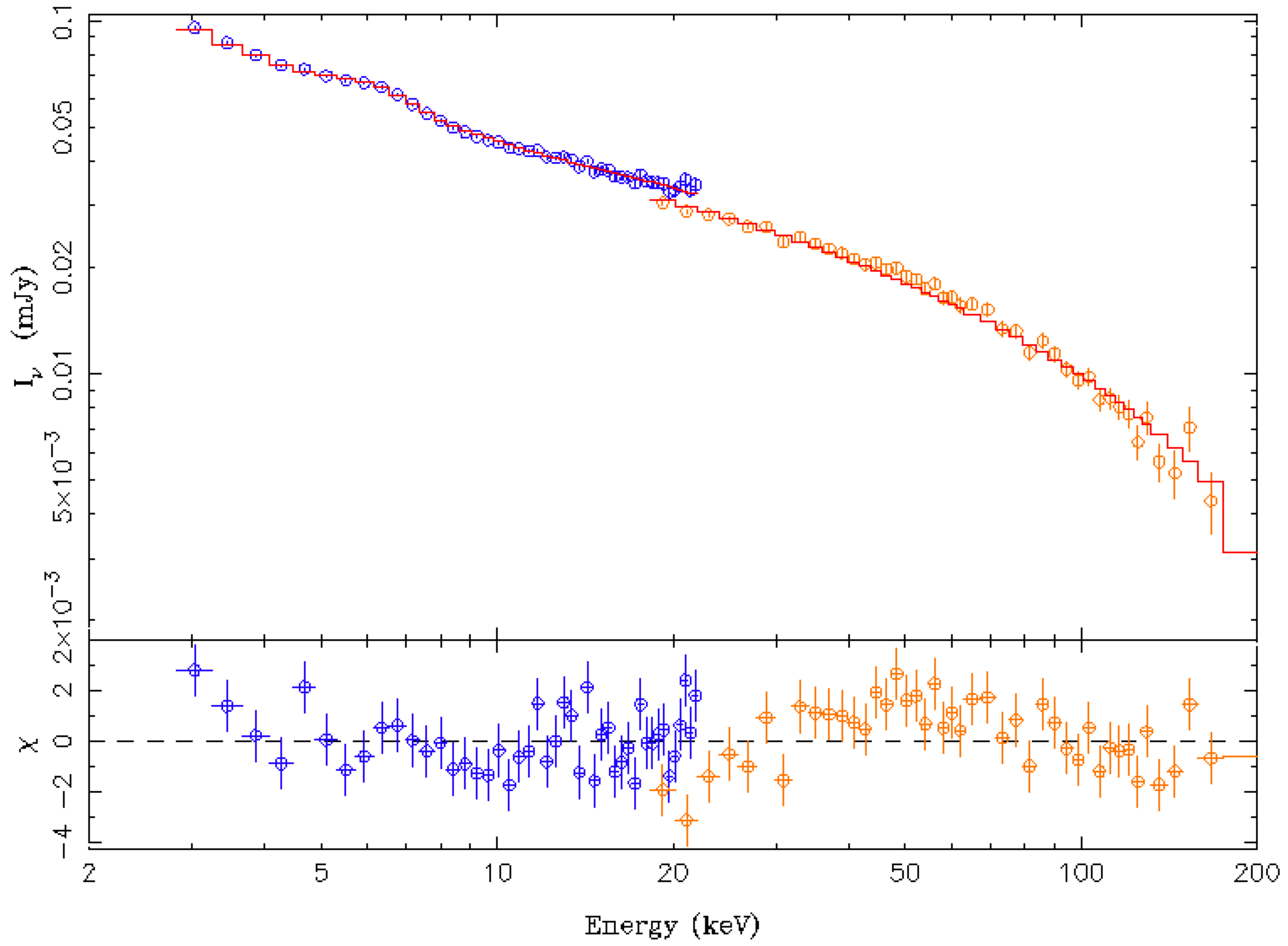


Add a disk reflection model

- To calculate relativistic reflection is a very complicated business, and very model dependent. This is just one simple model, as a demonstration, that does not include the iron line self-consistently so we keep that model
- Note that reflection is a *convolution* model applied to the source models: reflect(1, source models…)

```
fit_fun("constant(Isis_Active_Dataset)*phabs(1)*reflect(1,
(nthcomp(1)+gaussian(1)))");
list_par;
fit_counts;
```

Better fit, needs more robust method, but what about the radio/IR...?

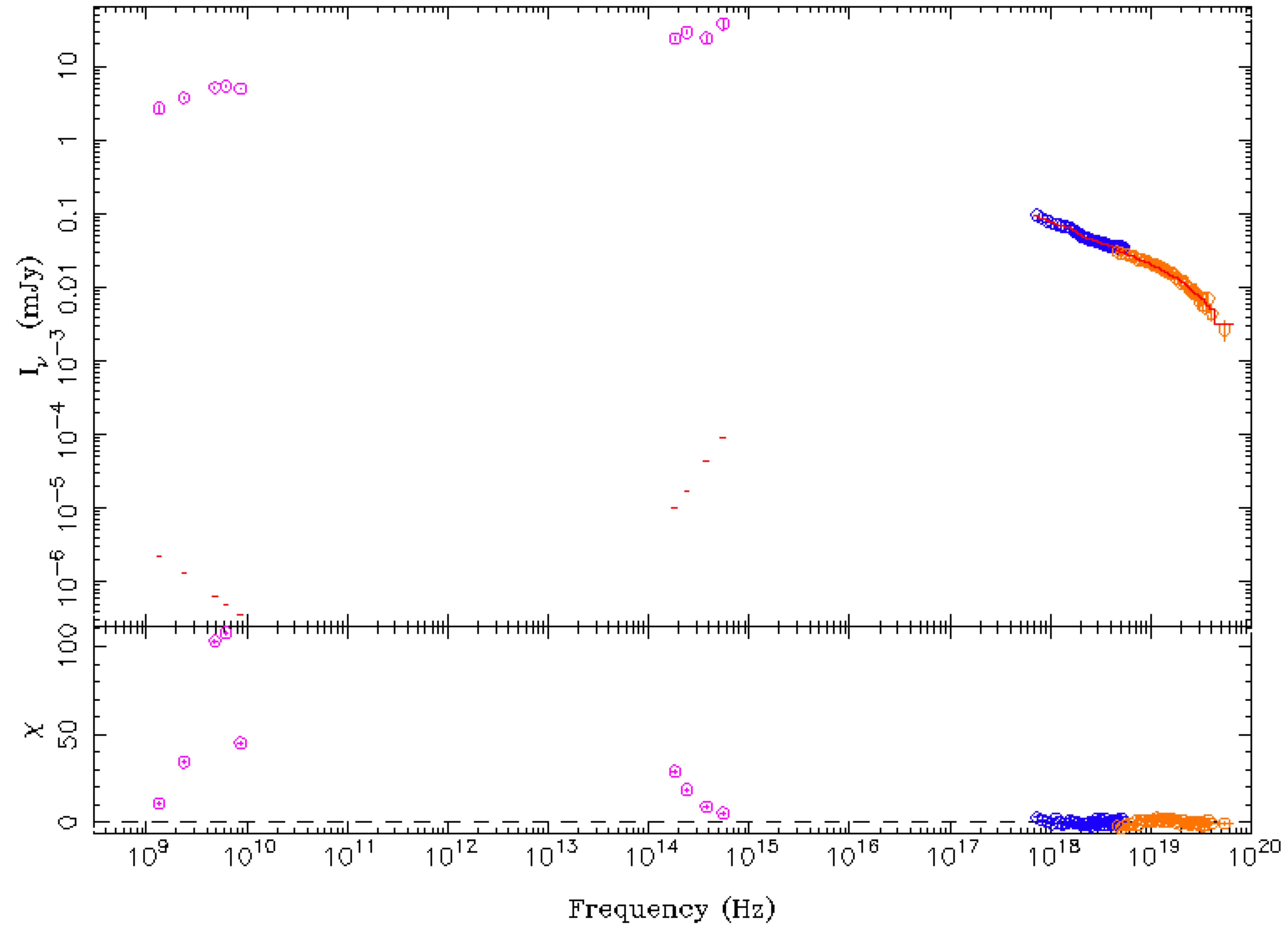


Let’s re-notice the radio and refit

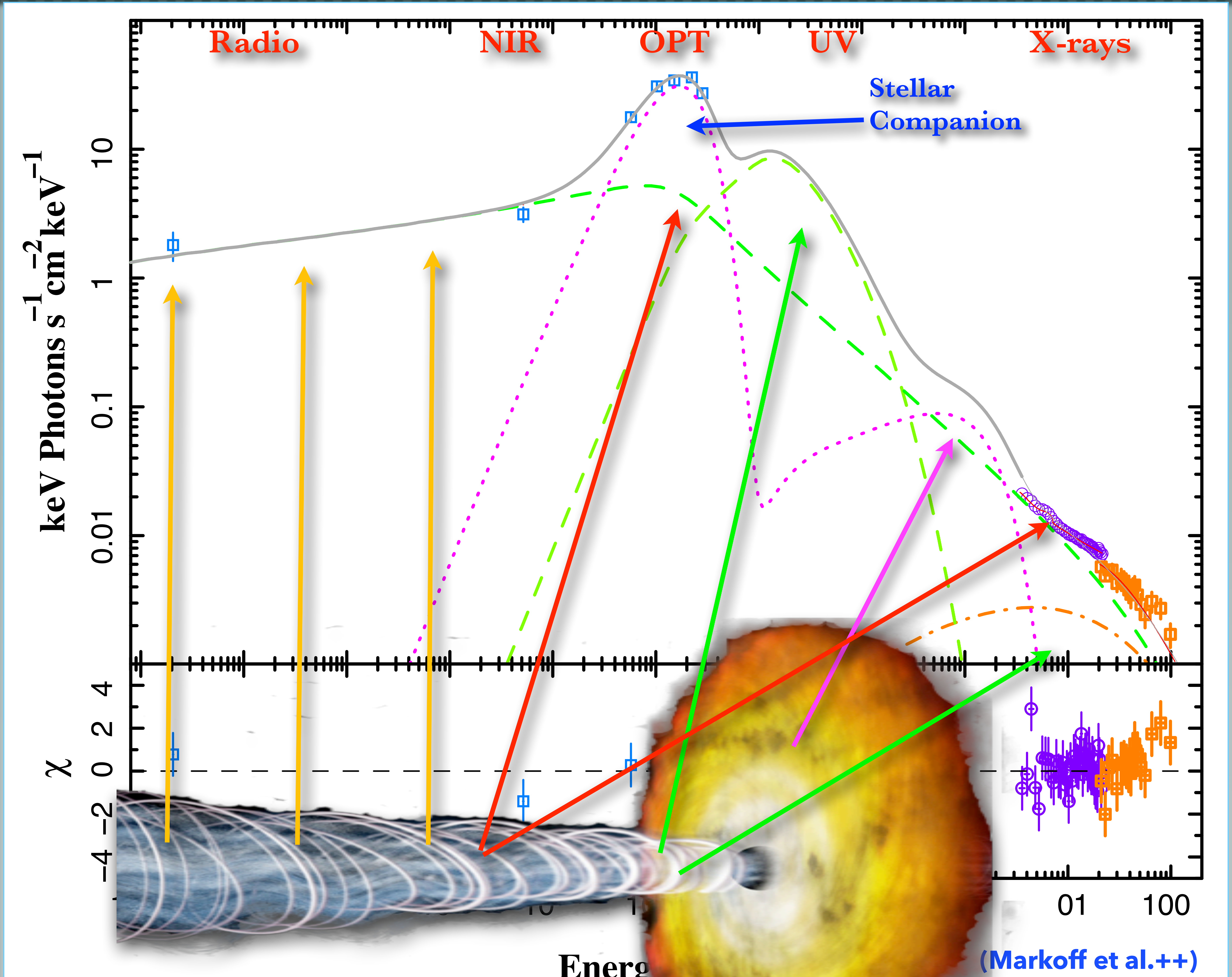
- Include radio/IR again, set constant normalisation to match PCA, change units to more typical “radio astronomy” units to study jets
- Just evaluate model (not a fit!) at same values we had for just X-rays

```
include(rid);
list_par;  %you see a new constant
set_par(“constant(3).factor”,1,1);
tie(18,1);
fancy_plot_unit(“hz","mjy");
eval_counts;
plot_unfold({rid,pid,hid};dsym={4,4,4},dcol={6,4,8},decol={6,4,8},xran
ge={3e8,1e20},res=1);
```

Hmm....X^2>300, not so good...



MW dependence: “microquasar”

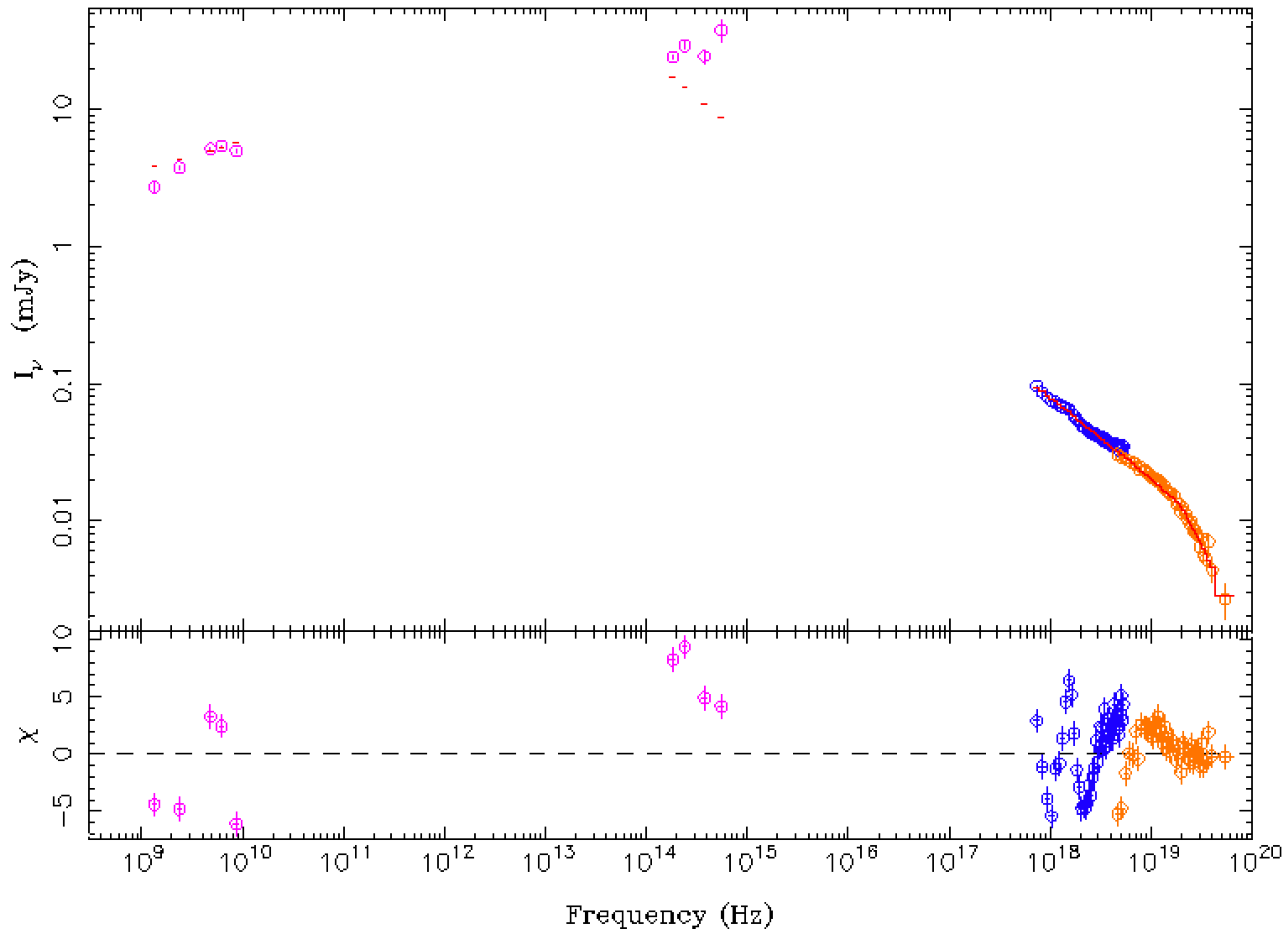


Start with a simple bknpower model again

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

```
fit_fun("constant(Isis_Active_Dataset)*phabs(1)*highecut(1)*(gaussian(
1) + bknpower(1))");
list_par;
set_par(2,.6,1);  %fix absorption to measured value
set_par(3,75); %set cutoff energy
set_par(8,1.e4); %give starting normalisation
set_par(9,0.75); %inverted PL to fit radio spectrum
set_par(10,0.001,0,1.e-5,1.e5); %give starting value/reset break range
set_par(11,1.6);
eval_counts;
renorm_counts; fit_counts;
plot_unfold({rid,pid,hid};dsym={4,4,4},dcol={6,4,8},decol={6,4,8},xran
ge={3e8,1e20},res=1);
```

Broken PL + cutoff + Gaussian = not terrible

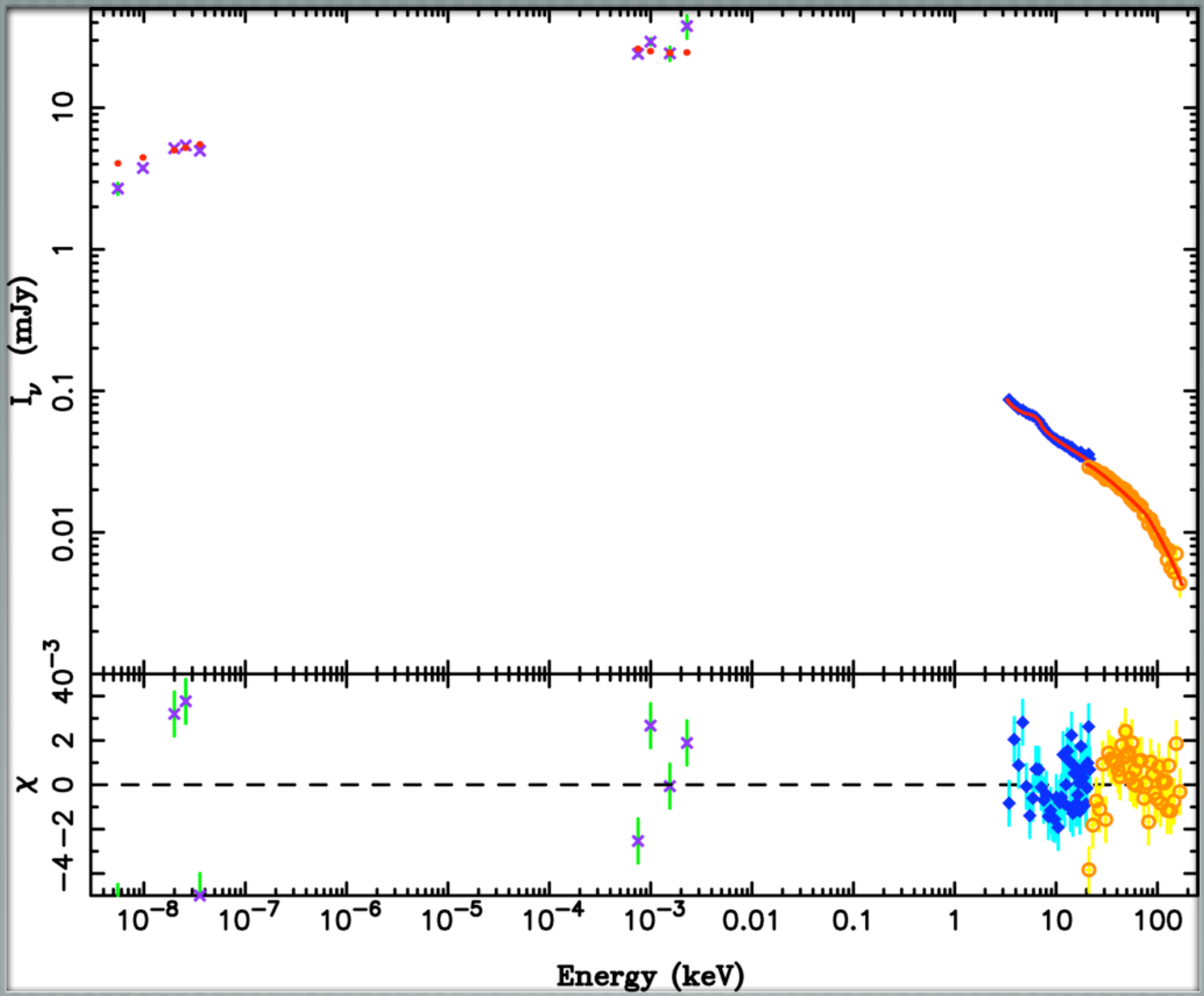


Can add more components to improve (we won't now...)

- You could at this point add in a blackbody as well for the missing flux, assuming it's stellar or disk related:

```
fit_fun("constant(Isis_Active_Dataset)*phabs(1)*highecut(1)*(bknpowe
r(1)+diskbb(1)+gaussian(1))");
isis> renorm_counts;
isis> fit_counts;
```

Not a terrible fit, for a simple model!



Save all your commands/parameters!

```
save_input(“mwfitting.sl”); %history!!
save_par(“mwfitting.par”);
```

Assuming that worked, load model into isis

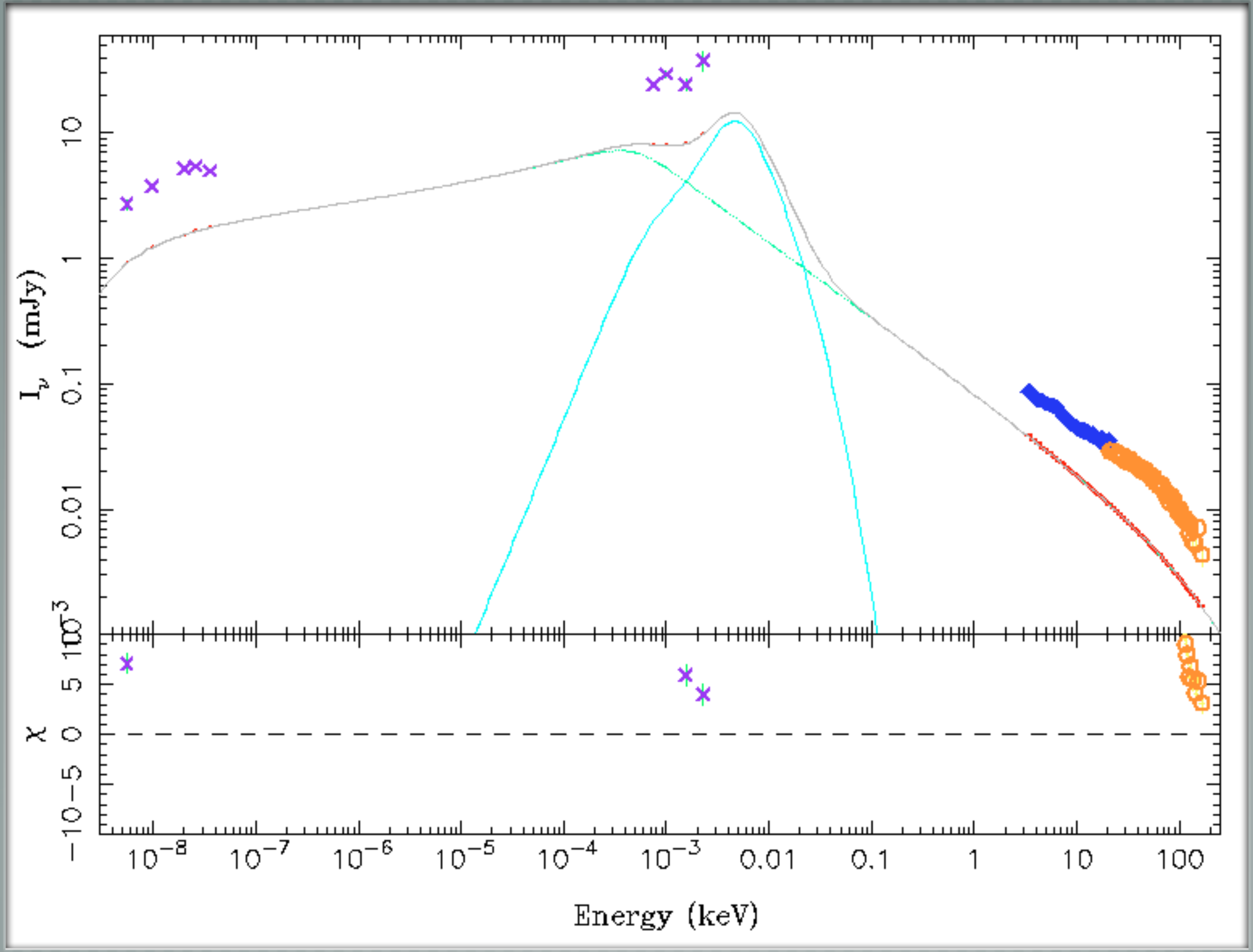
- Go back into Isis, using your saved input files as a reference now, and reload radio/IR, PCA and HEXTE data
- Set the grids as before
- Now load the new model to see if it works in Isis (this is usually where the problems occur…):

```
evalfile(“simplejet.sl”);
fit_fun(“simplejet”);
list_par;
```

Let's look at the input parameters

```
isis> list_par;
simplejet
 idx  param                tie-to  freeze           value        min        max
  simplejet
 idx  param                tie-to  freeze           value        min        max
  1  simplejet(1).norm        0      1                  1          0    1000000
  2  simplejet(1).mbh         0      1                 10          1      1e+09  msun
  3  simplejet(1).jetrat      0      0              0.001      1e-07          1  L_edd
  4  simplejet(1).pspec       0      0                  2        1.7          4
  5  simplejet(1).zsh         0      0                 10          5      10000  r_g
  6  simplejet(1).r0          0      0                  5          2        100  r_g
  7  simplejet(1).hratio      0      1                1.5        0.1        100
  8  simplejet(1).incl        0      1                 40          0         90  deg
  9  simplejet(1).eltemp      0      0              2e+10      2e+09      5e+11  K
 10  simplejet(1).plfrac      0      0               0.75     0.0001          1
 11  simplejet(1).dkpc        0      1                  1        0.1      10000  kpc
 12  simplejet(1).comsw       0      1                  0          0          1
 13  simplejet(1).plotsw      0      1                  1          0          1
 14  simplejet(1).fsc         0      0             0.0036      1e-07      0.036
 15  simplejet(1).zmax        0      1                 14         13         16  lg cm
 16  simplejet(1).equip       0      0                  2        0.1        100
```

Run it, plot it, should give something like:



Time to cache model

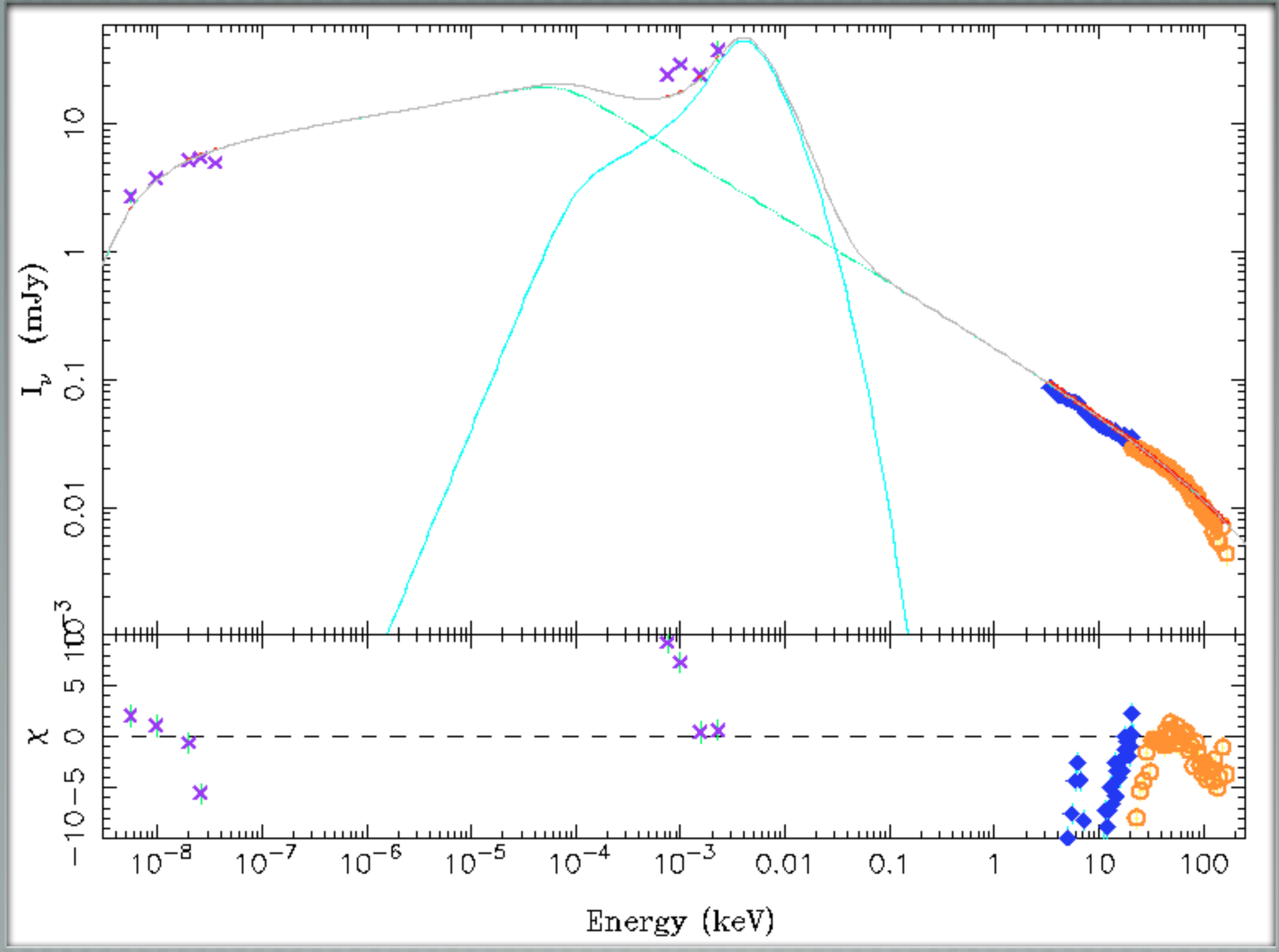
- Note that every time you eval_counts, the model is run twice. That's because pca/radio are already cached on one grid, but hexte has its own grid and the model is run for each grid
- Before fitting, to save time, let's cache the model on all grids
- Note that now you have to reimport a new set of parameters

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