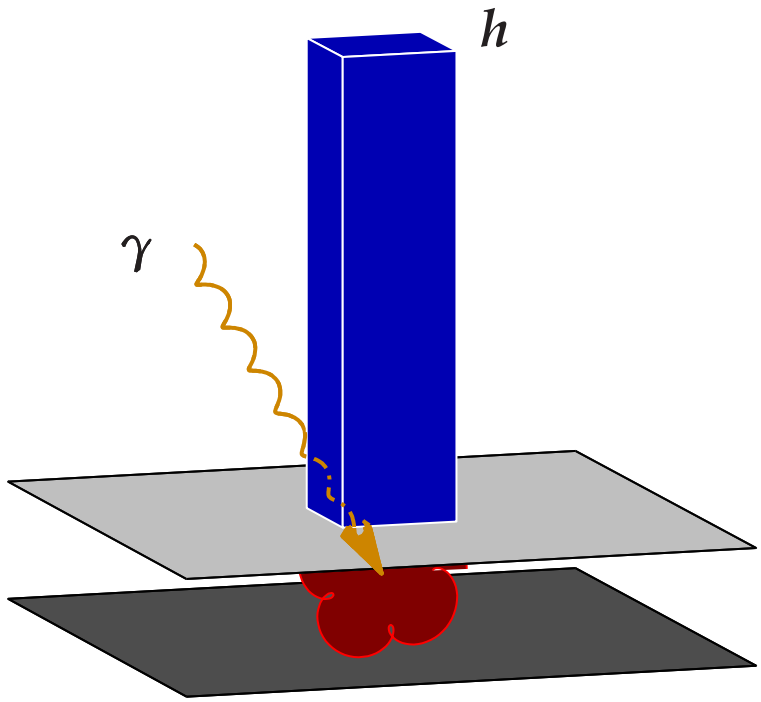
Pile-up Modeling

John E. Davis
<davis@space.mit.edu>

Outline

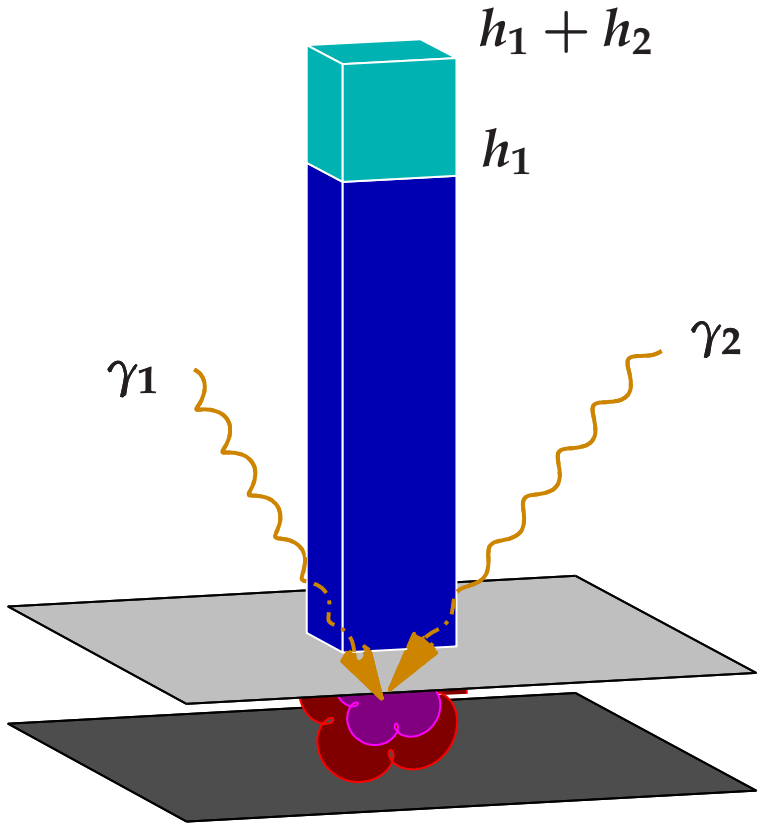
- Event Detection
- Grade Migration
- The “Standard Model”
- The Pile-up Model
- Data Preparation
- Using the model in `sherpa` and `isis`
- Examples

When a photon is absorbed in the silicon of a CCD, a charge cloud of electron-hole pairs is formed ($\approx$ 3.65 eV per pair).

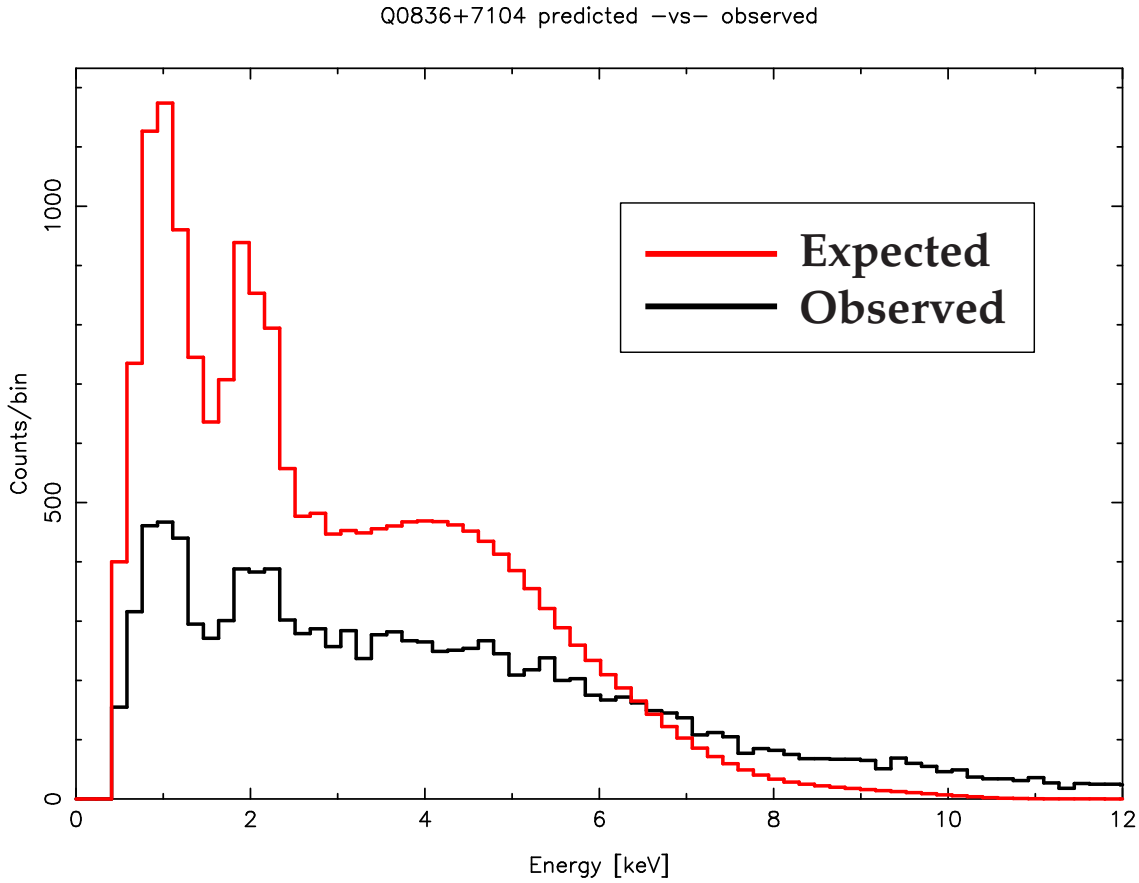


The pulse-height is a measure of the number of pairs in the cloud.

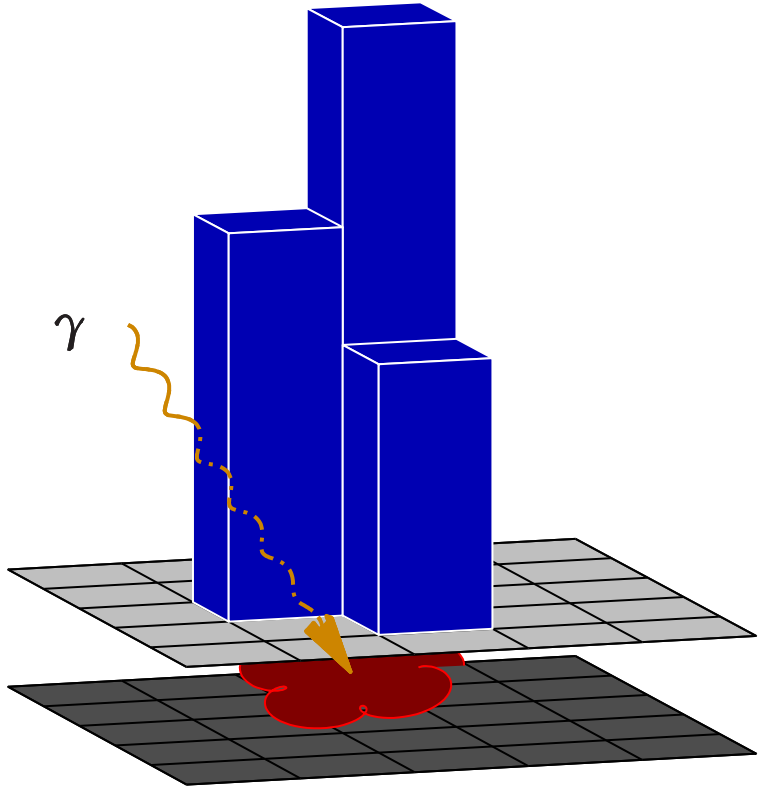
For a bright source, there is a non-negligible probability for two or more photons to arrive in the same region during an integration time. The detector will be unable to distinguish the two events. This phenomena is called “pile-up”.



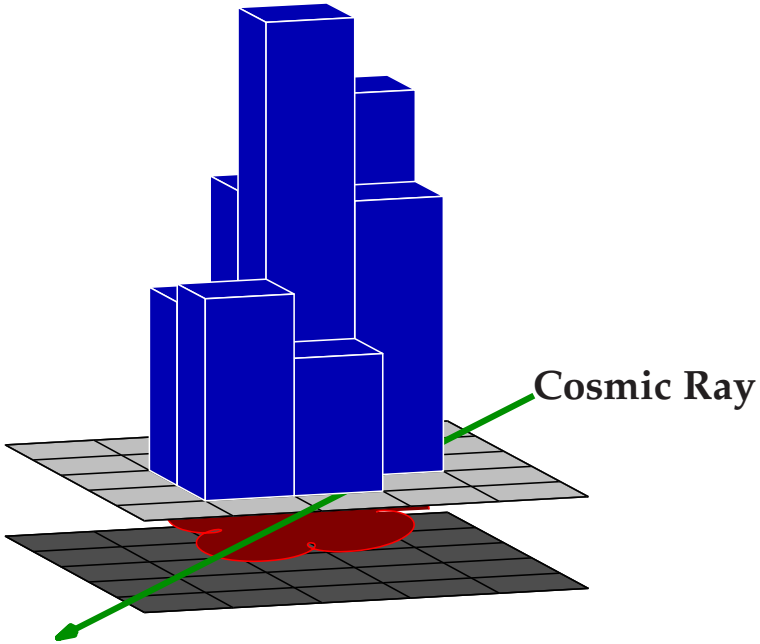
Pile-up manifests itself by a lower event detection rate and an energy spectrum distorted towards higher energies.

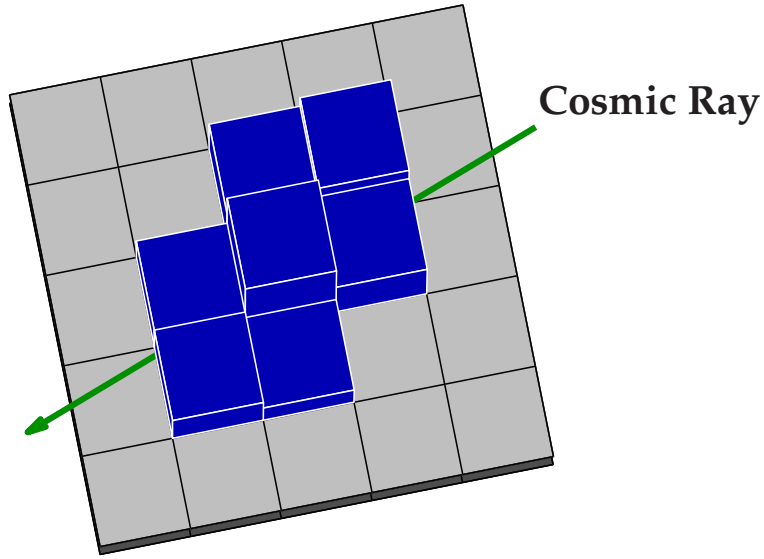


The charge cloud is not necessarily confined to a single pixel.



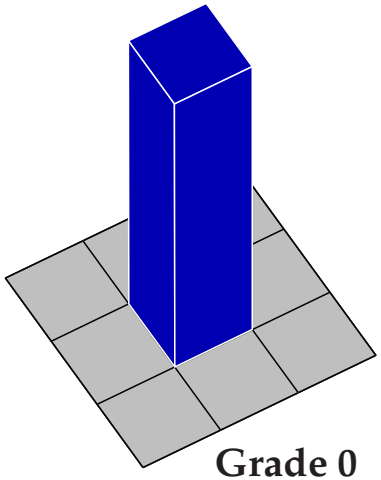
Cosmic rays also produce charge clouds!

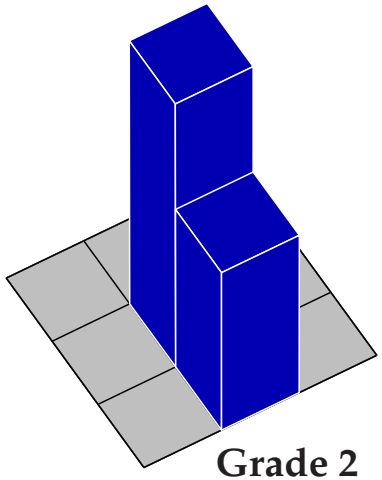


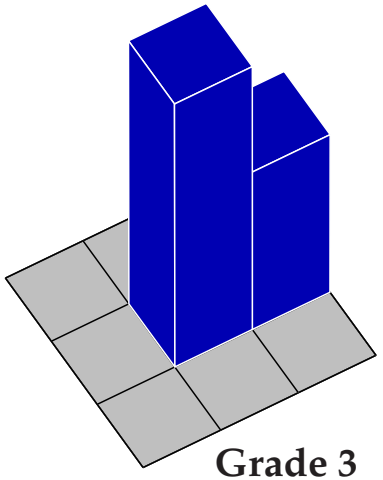


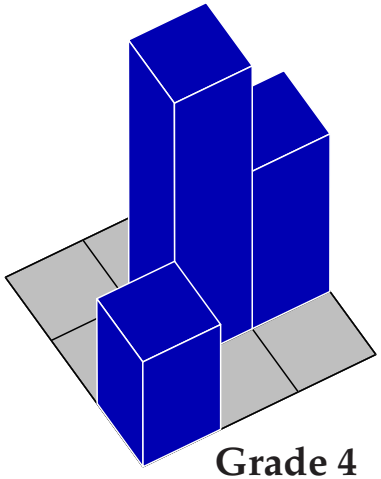
Fortunately, the charge cloud patterns, or “grades”, from cosmic rays are different from the patterns produced by X-rays.

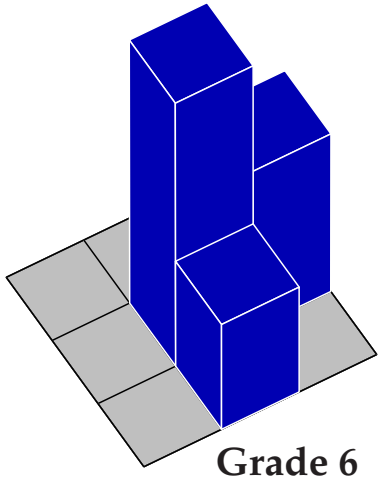
X-rays are more likely to produce events with “good grades”.

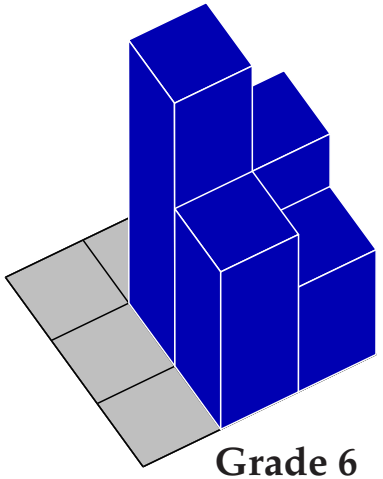




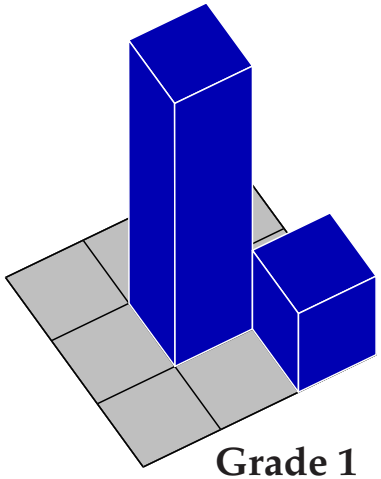


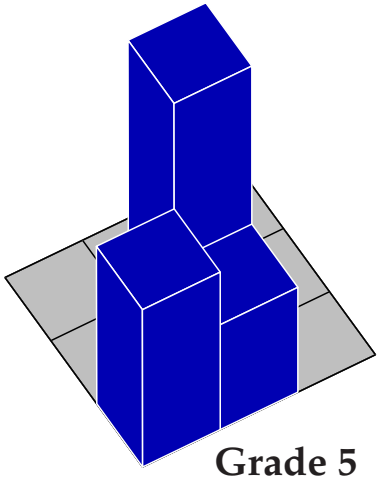


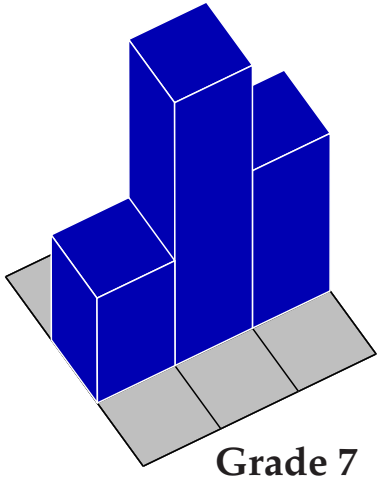


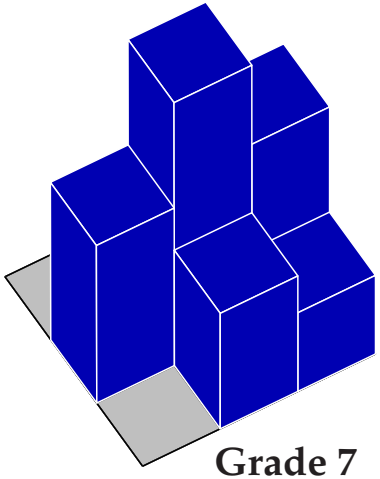


Cosmic rays are more likely to produce events with “bad grades”.

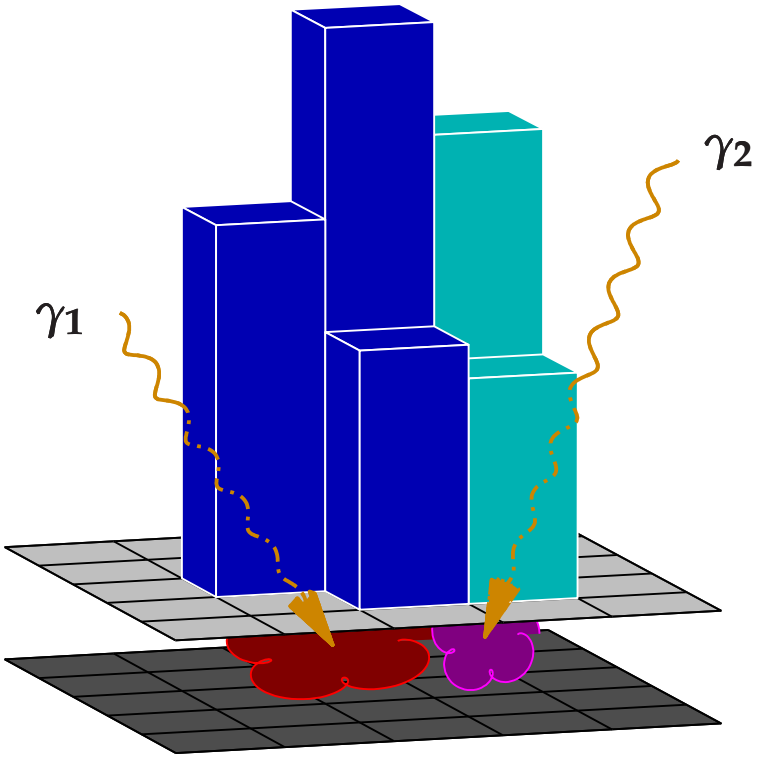


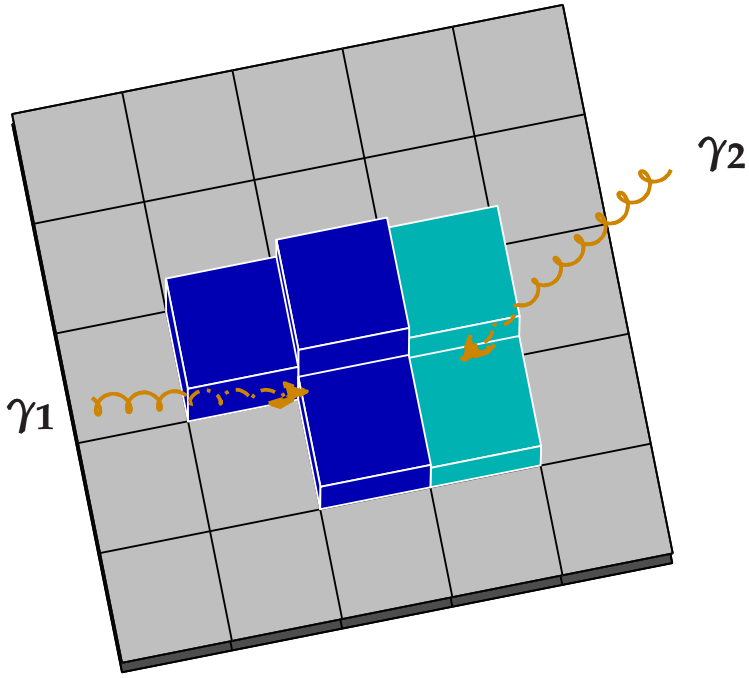






Pile-up can also produce events with “bad grades”.





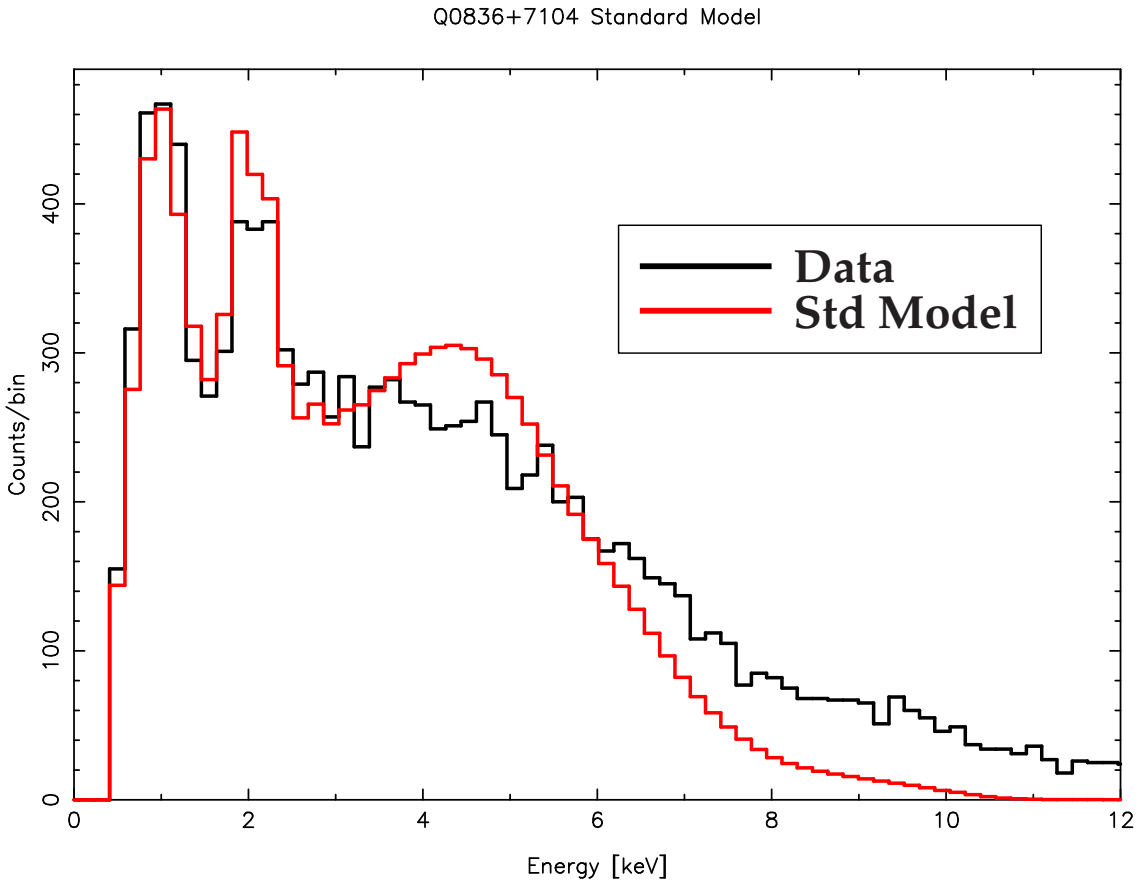
This effect is called “Grade Migration”.

The "Standard Model"

$$C(h) = (N\tau) \int \mathrm{d}E \; R(h,E) A(E) s(E)$$

$C(h)$	The number of counts in pulse-height bin h.	`dmextract`
$s(E)$	Incident source flux	`sherpa`
$A(E)$	Effective area, or "ARF"	`mkarf`
$R(h,E)$	Detector redistribution matrix, or "RMF"	`mkrmf`
τ	CCD frame time	$\approx$ `TIMEDEL`
N	Total number of CCD frames	$N\tau$ = `EXPOSURE`

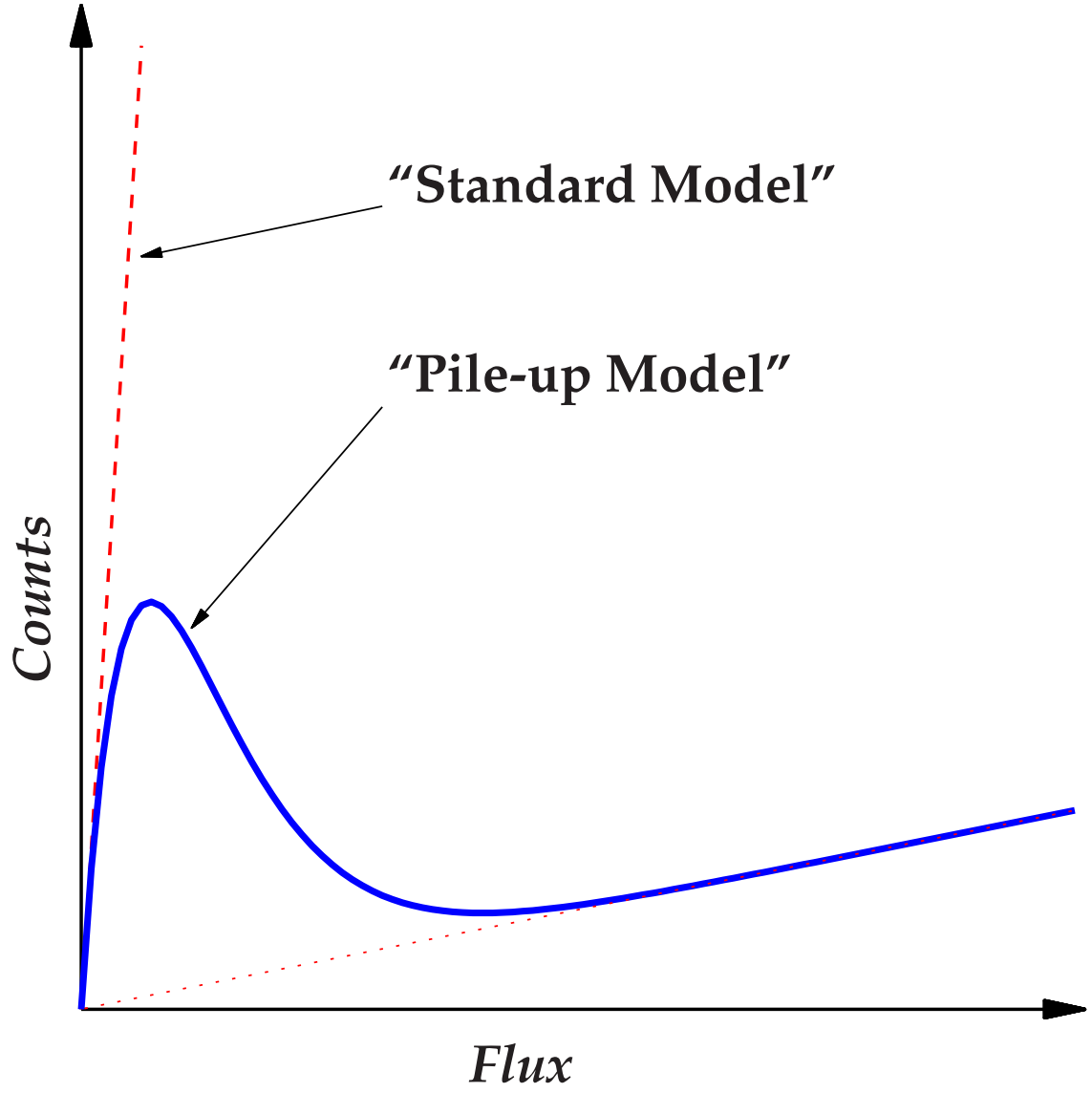
Pile-up Modeling



The Pile-up Model

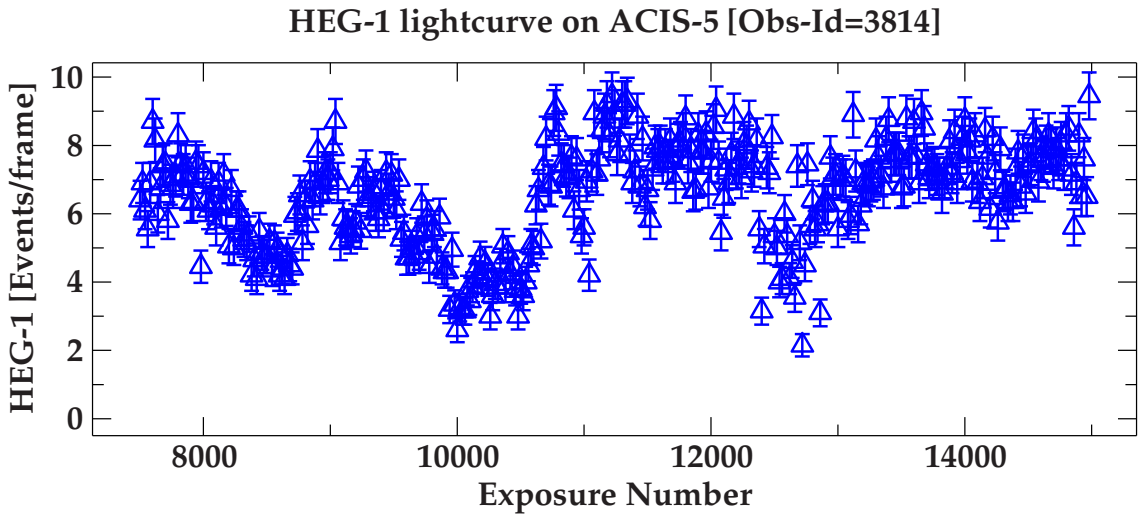
$$C(h) = (N\tau)(1-f)\int dE\ R(h,E)A(E)s(E)$$
$$+Nne^{-(\tau/\bar{g}_0)\int dE\ A(E)fs(E)/n}\sum_{p=1}^{\infty}\alpha^{p-1}\int dE\ R(h,E)\frac{[\tau A(E)fs(E)/n]^{*p}}{p!}$$

- n The number of regions where pile-up occurs.
- f Total PSF fraction enclosed by the n pile-up regions.
- α Grade-migration survival probability.
- $\bar{g}_0$ Average branching ratio into "good" grades.
- $*p$ Convolution product ($\int_p \cdots \int$)

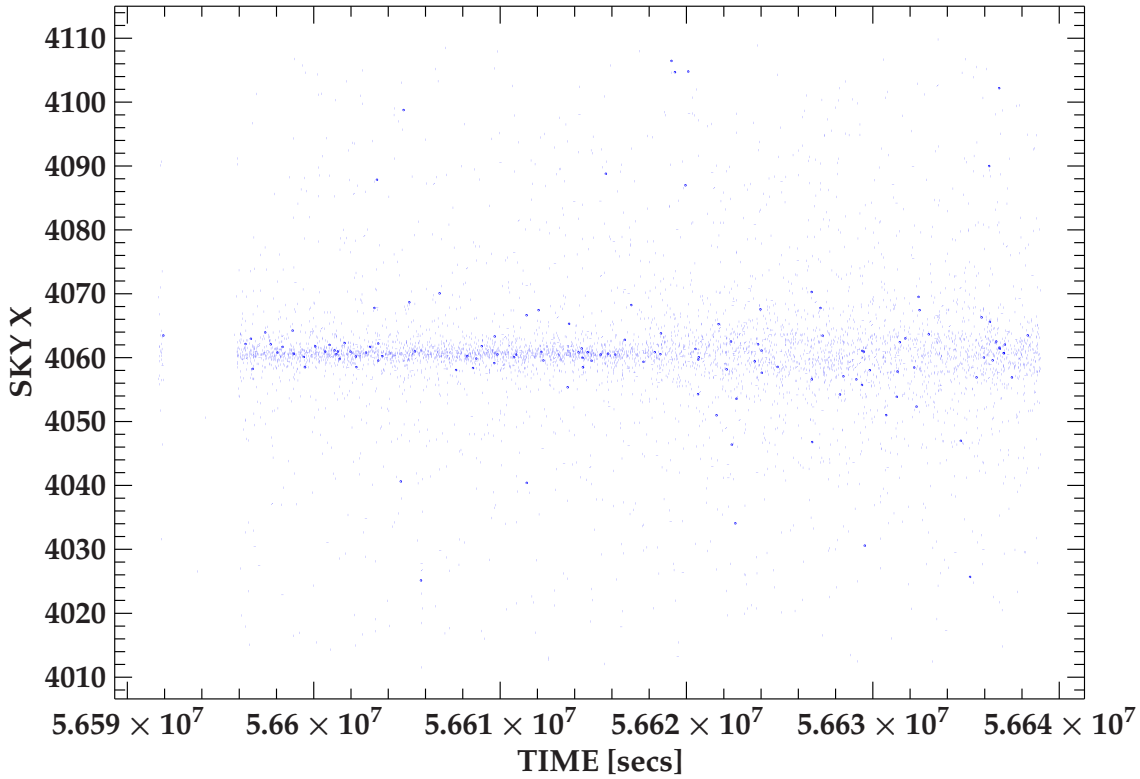

"Standard Model"
"Pile-up Model"
Counts
Flux

Data Preparation

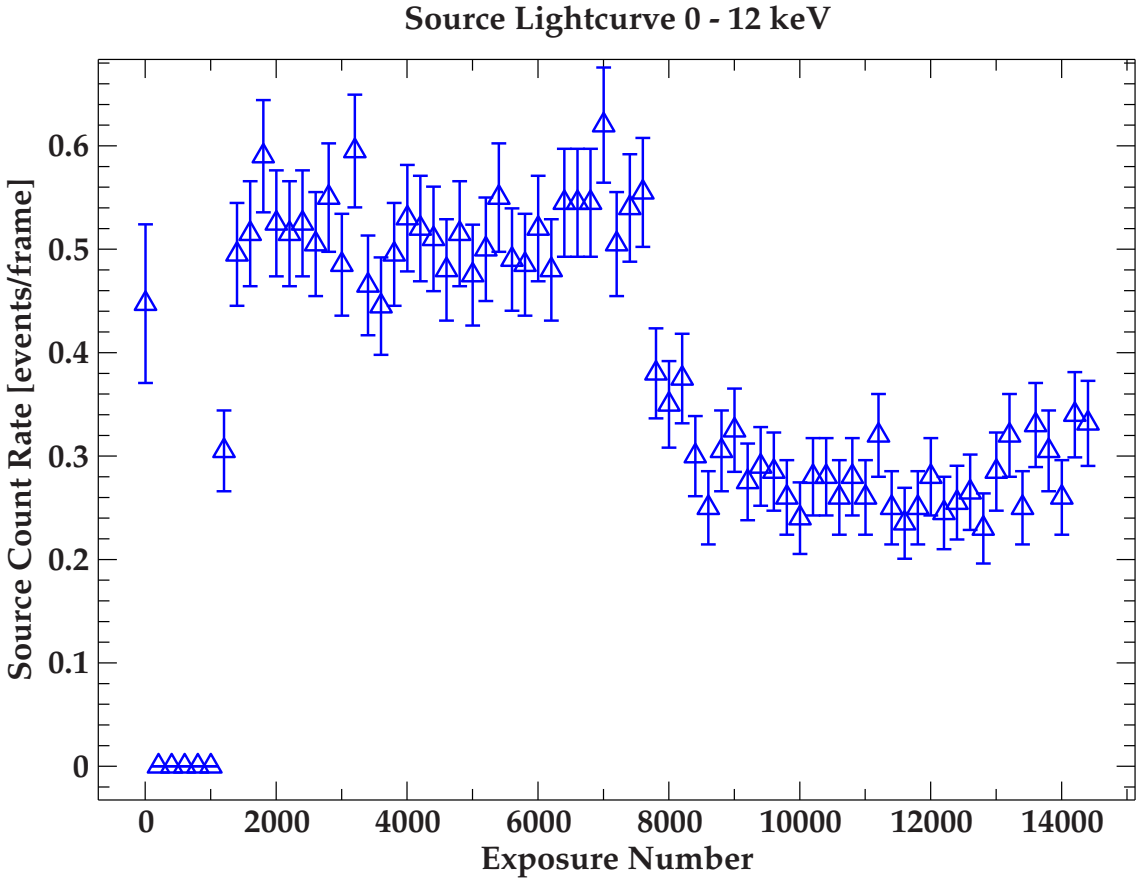
Know your lightcurve! The pile-up model assumes that photon arrival times are Poisson distributed. Use time-intervals where the incident flux is constant. Use either the counts in the wings of the PSF or in the readout streak to generate the lightcurve.



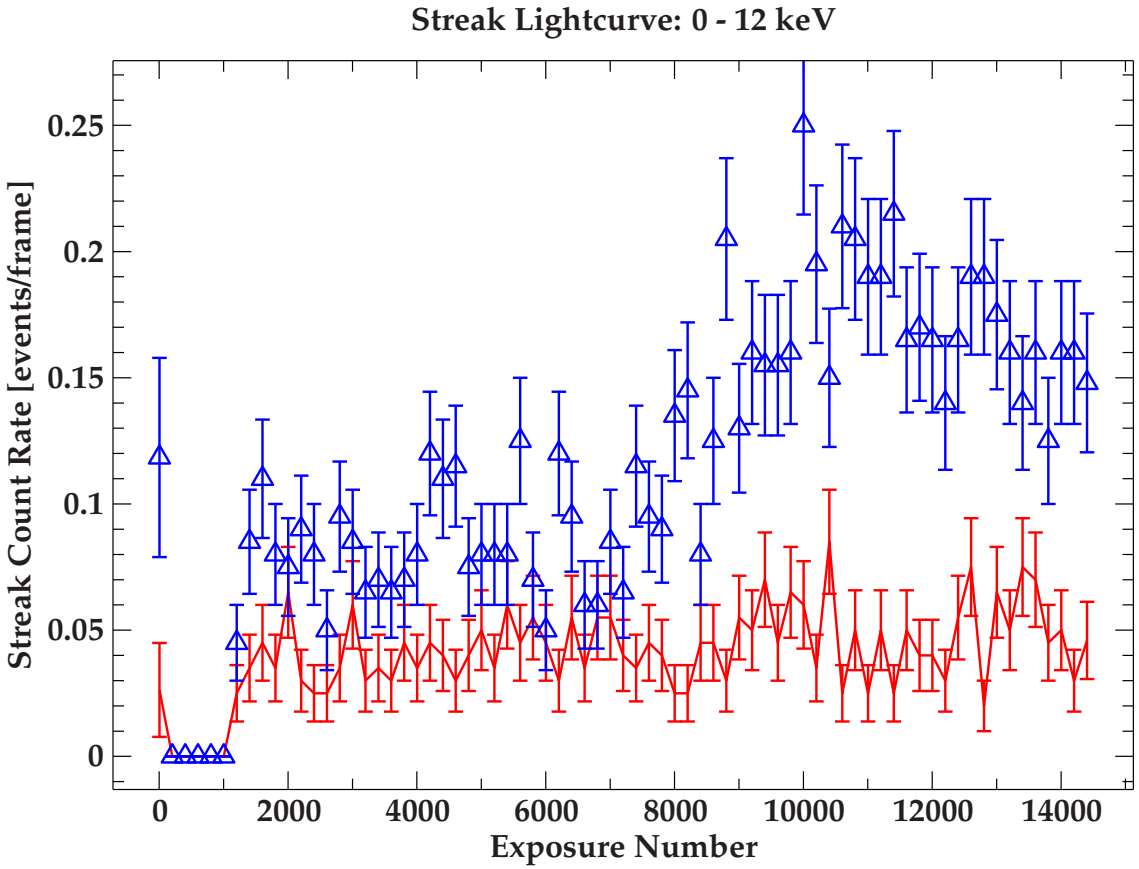
II Peg Light Curve *OBSID 1451, HETG/ACIS-S, Claude Canizares*



II Peg Light Curve *OBSID 1451, HETG/ACIS-S, Claude Canizares*

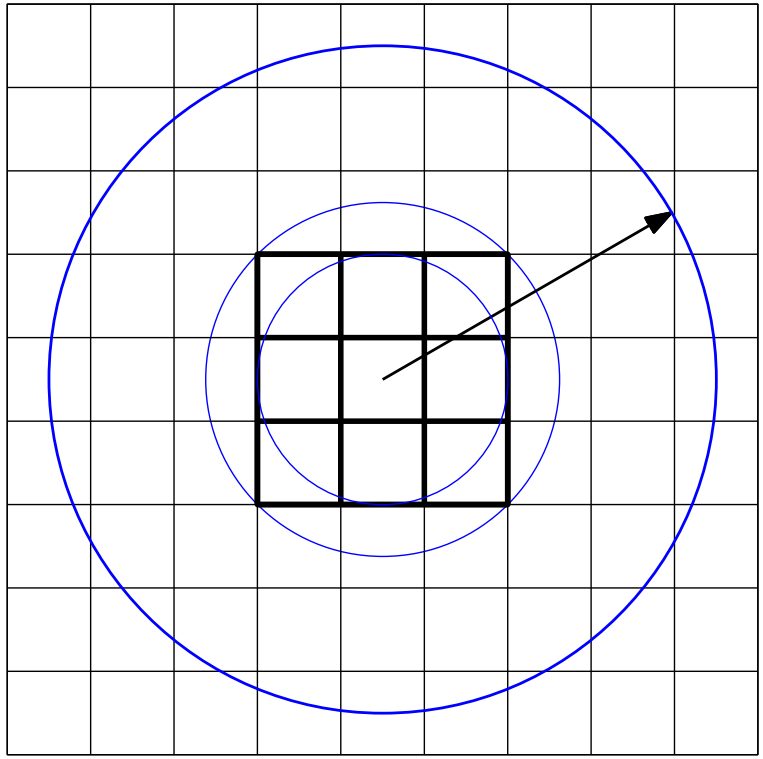


II Peg Light Curve



Use the right extraction region

For an on-axis point source, it is recommended that a circular region with a 2 arc-second radius be used (4 ACIS pixels). A larger region will decrease the signal to noise ratio.



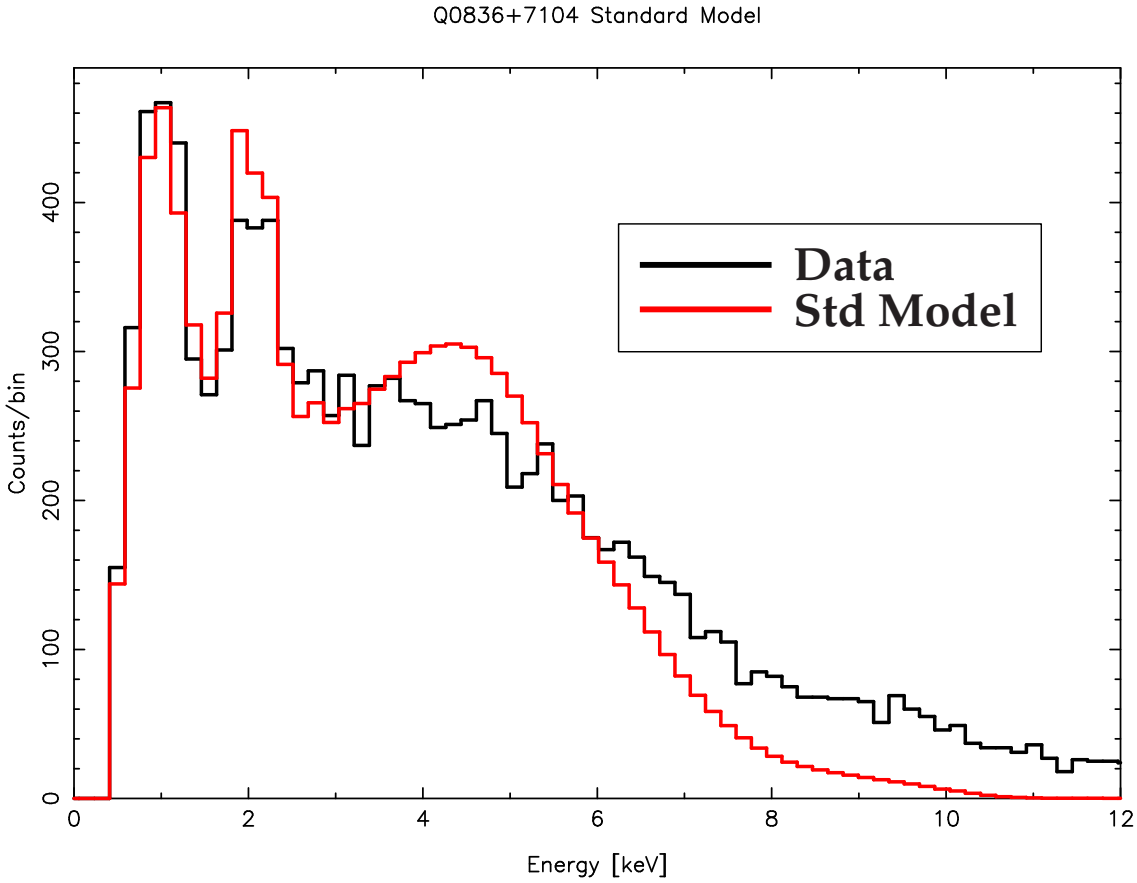
Using the pile-up model in sherpa

```
    .
    .
sherpa> set_pileup_model(jdpileup.jdp);
sherpa> jdp.f.min=0.85;
sherpa> jdp.f.max=0.95;
sherpa> jdp.f=0.9;
sherpa> jdp.n = 1
sherpa> jdp.alpha = 0.5
sherpa> jdp.f = 0.95
sherpa> thaw (jdp.f, jdp.alpha);
sherpa> freeze (jdp.n);
sherpa> fit;
sherpa> print(jdp);
```

Using the pile-up model in isis

```
    .
    .
isis> set_kernel (1, "pileup");
isis> set_par ("pileup<1>.nregions", 1, 1);
isis> set_par ("pileup<1>.alpha", 0.5, 0);
isis> set_par ("pileup<1>.psffrac", 0.9, 0, 0.85, 0.95);
isis> fit_counts;
isis> print_kernel (1);
```

Example 1: Q0836+7104 *OBSID 1450, HETG/ACIS-S, Claude Canizares*



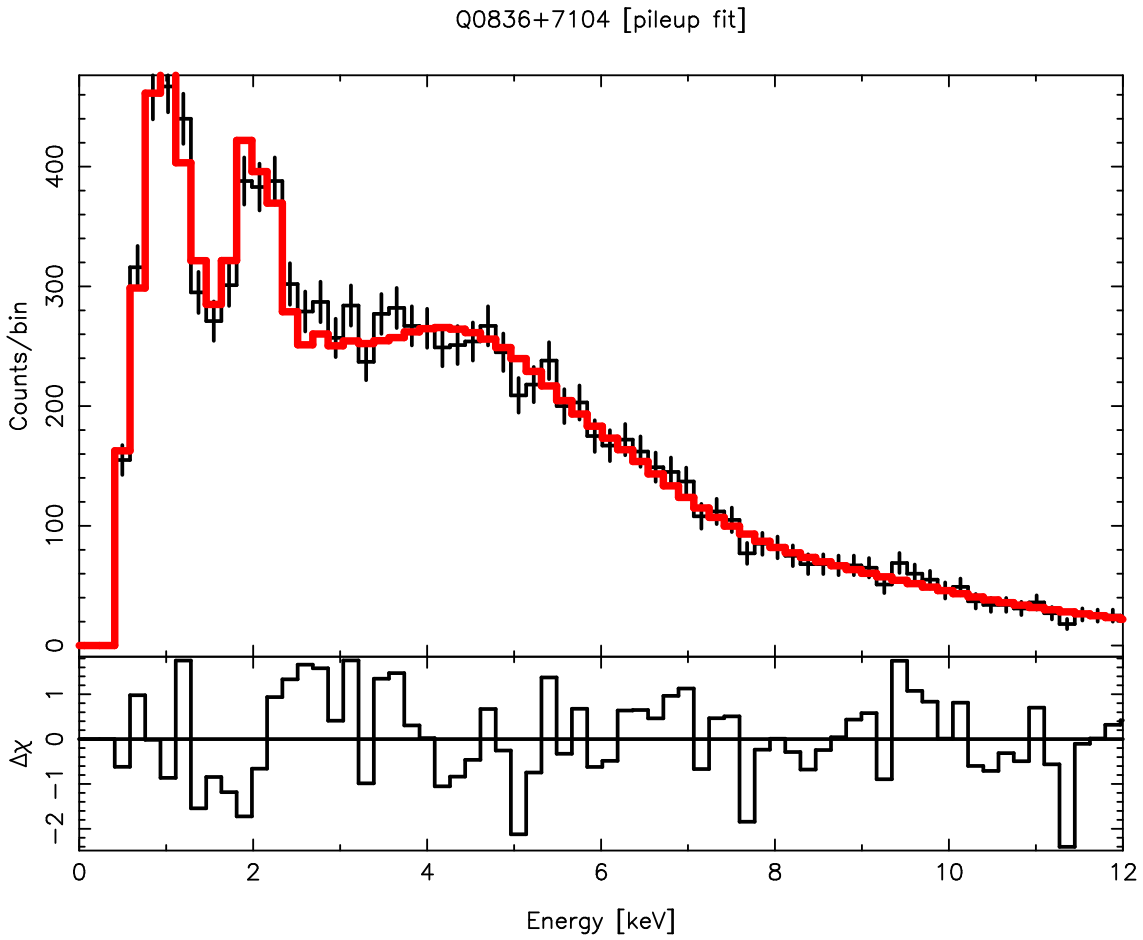
Pile-up Fit

```
isis> fit_counts;

 Parameters[Variable] = 7[5]
 Data bins = 67
 Chi-square = 63.1
 Reduced chi-square = 1.018

isis> list_par;
phabs(1)*powerlaw(1)
 idx  param              tie-to  freeze        value          min          max
  1  phabs(1).nH             0     0       0.03381944            0          0.2
  2  powerlaw(1).norm        0     0      0.002399927            0         0.01
  3  powerlaw(1).PhoIndex    0     0         1.374511            0            3
  4  pileup<1>.nregions      0     1                1            1           10
  5  pileup<1>.g0            0     1                1            0            1
  6  pileup<1>.alpha         0     0        0.8728455         0.35            1
  7  pileup<1>.psffrac       0     0        0.9304663          0.9            1
```

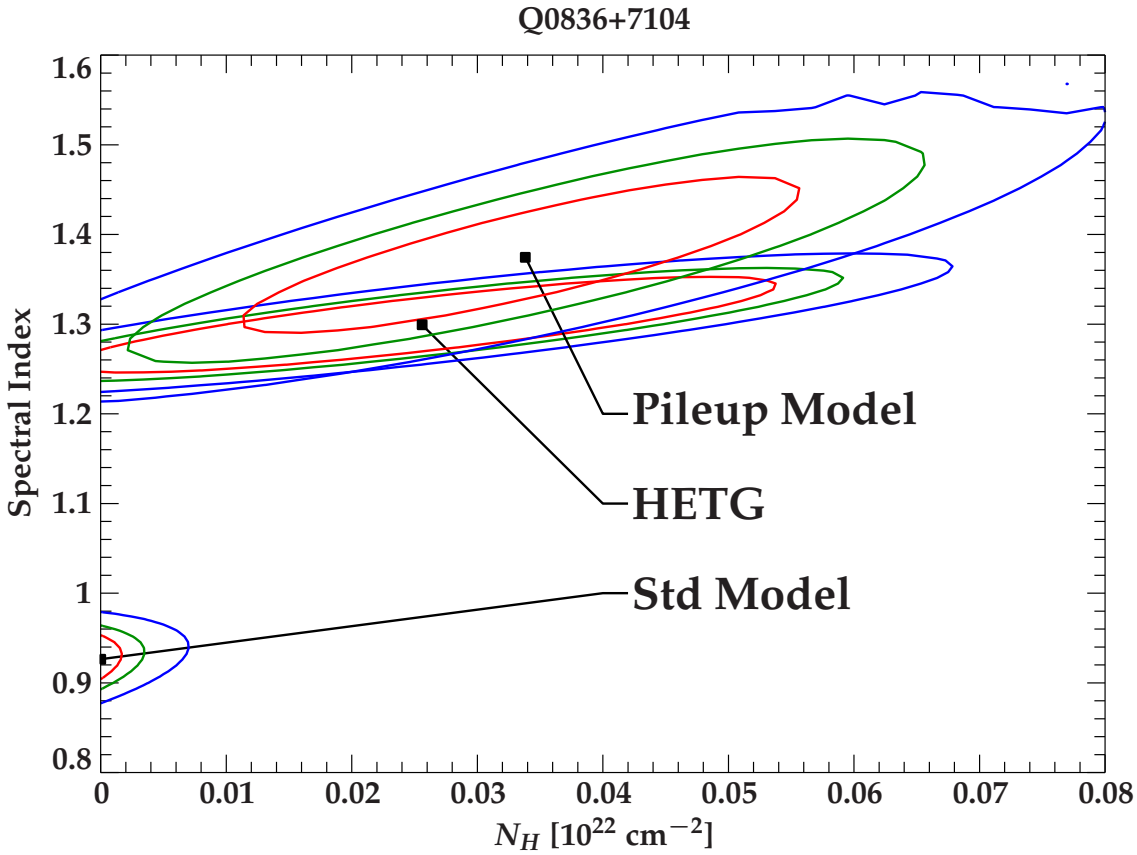
Spectral Fit



Pile-up Fraction

```
isis> print_kernel(1);
 1: 0.367879     0.625742
 2: 0.184252     0.273553
 3: 0.061522     0.0797253
 4: 0.0154067    0.0174266
 5: 0.00308658   0.00304732
 6: 0.000515305  0.000444061
 7: 7.37403e-05  5.54651e-05
 8: 9.23323e-06  6.06186e-06
 *** pileup fraction: 0.374258
```

Confidence Contours



Example 2: NGC 4579 *OBSID 807, ACIS-S, Michael Eracleous*

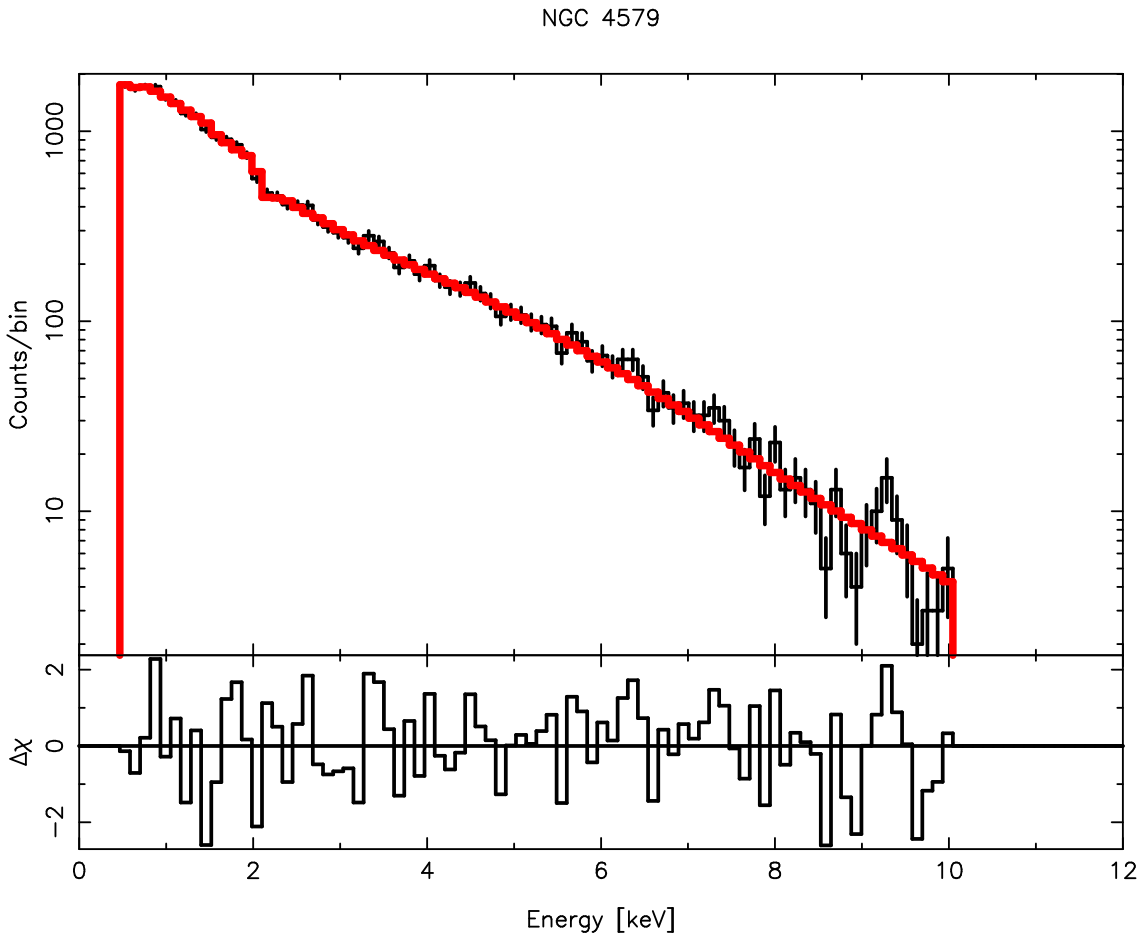
```
isis> fit_counts;

 Parameters[Variable] = 7[5]
            Data bins = 82
          Chi-square = 103.6
   Reduced chi-square = 1.346

isis> list_par;

phabs(1)*powerlaw(1)
 idx  param              tie-to  freeze        value          min          max
  1  phabs(1).nH             0     0       0.02727911            0          0.2
  2  powerlaw(1).norm        0     0      0.001469524            0          0.1
  3  powerlaw(1).PhoIndex    0     0         1.764569            1            3
  4  pileup<1>.nregions      0     1                1            1           10
  5  pileup<1>.g0            0     1                1          0.8            1
  6  pileup<1>.alpha         0     0        0.6937202            0            1
  7  pileup<1>.psffrac       0     0         0.902147          0.8            1
```

Spectral Fit

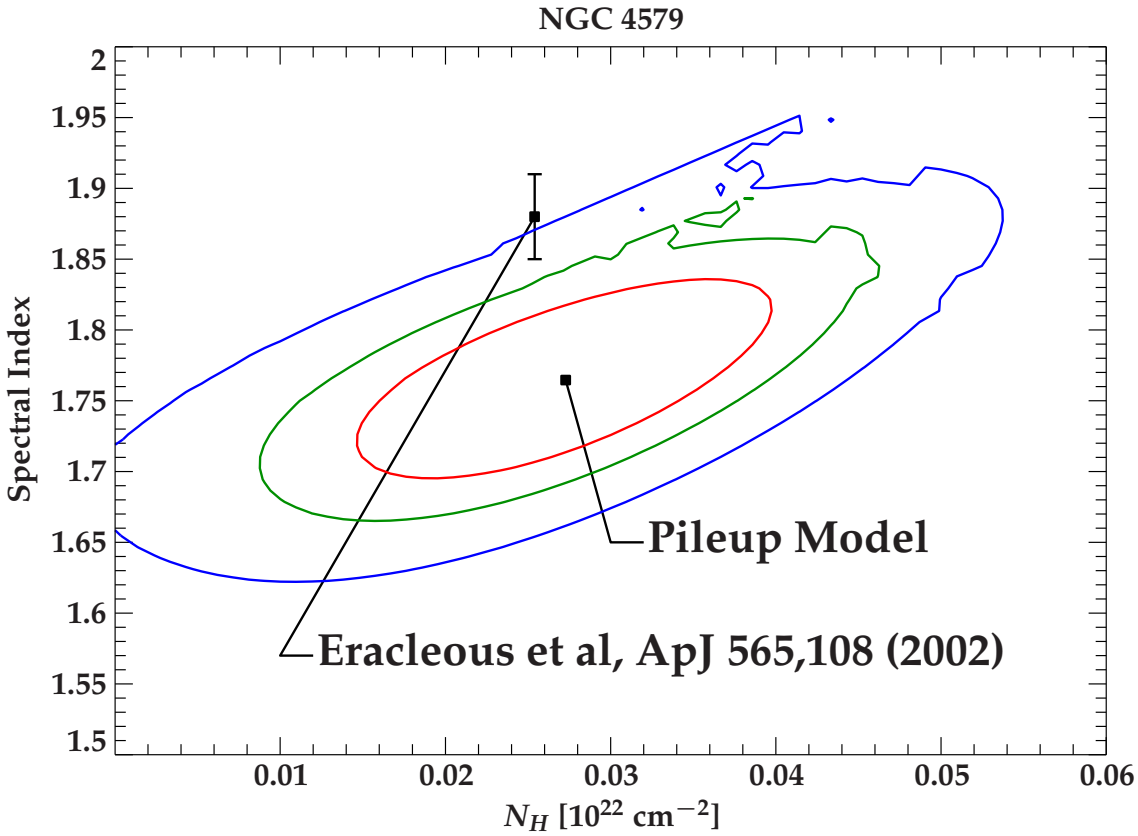


Pile-up Fraction

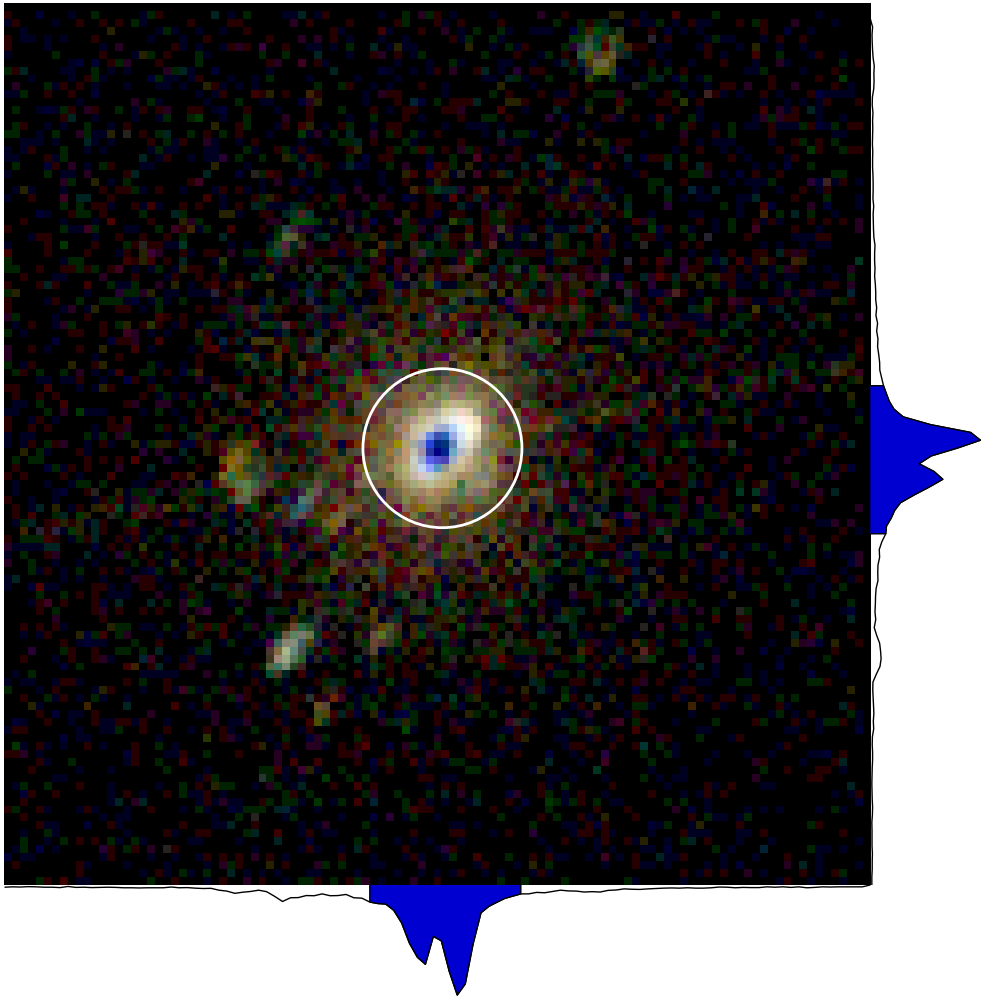
```
isis> print_kernel(1);

 1: 0.367835     0.688744
 2: 0.186795     0.242635
 3: 0.0632391    0.0569848
 4: 0.0160571    0.0100375
 5: 0.00326167   0.00141443
 6: 0.000552117  0.000166095
 7: 8.01078e-05  1.67181e-05
 8: 1.01701e-05  1.47239e-06
 *** pileup fraction: 0.311256
```

Confidence Contours



Example 3: Nucleus of M81 *OBSID 735, ACIS-S, Douglas Swartz*



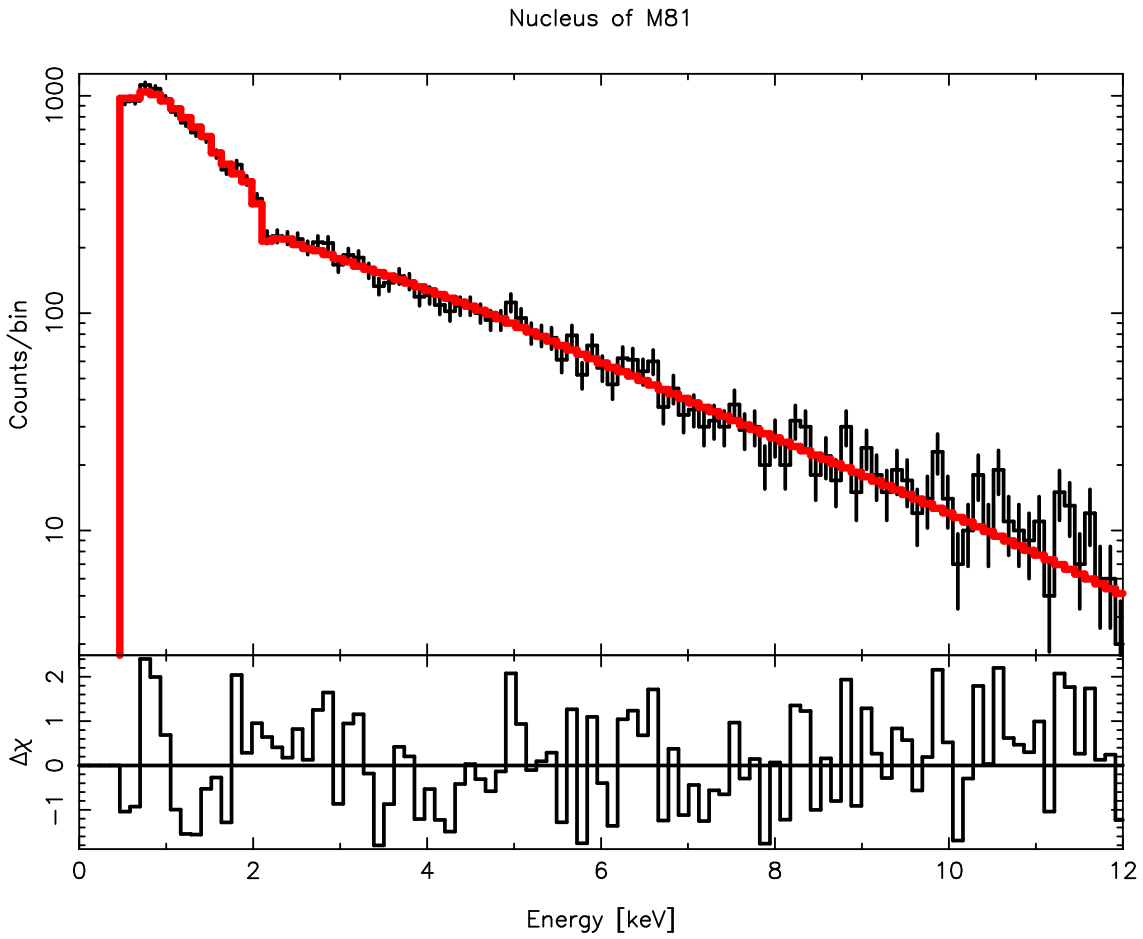

```
isis> fit_counts;

 Parameters[Variable] = 7[5]
            Data bins = 99
          Chi-square = 120.8
   Reduced chi-square = 1.285

isis> list_par;

phabs(1)*powerlaw(1)
 idx  param              tie-to  freeze        value          min          max
  1  phabs(1).nH             0     0       0.08063485            0            1
  2  powerlaw(1).norm        0     0      0.006880267            0          0.1
  3  powerlaw(1).PhoIndex    0     0          1.92351            0            3
  4  pileup<1>.nregions      0     1                3            1           10
  5  pileup<1>.g0            0     1                1            0            1
  6  pileup<1>.alpha         0     0        0.4810092            0            1
  7  pileup<1>.psffrac       0     0        0.9578905            0            1
```

Spectral Fit



Pile-up Fraction

```
isis> print_kernel(1);

 1: 0.0241037    0.207965
 2: 0.0652543    0.270813
 3: 0.117772     0.235102
 4: 0.159419     0.153076
 5: 0.172633     0.0797342
 6: 0.155786     0.03461
 7: 0.1205       0.0128769
 8: 0.0815551    0.0041921
 9: 0.0490641    0.0012131
 10: 0.0265656   0.000315941
 11: 0.0130762   7.48036e-05
 12: 0.00590006  1.62349e-05
 13: 0.00245736  1.04828e-05
 *** pileup fraction: 0.792035
```

Confidence Contours