

PySpi for Persistent Sources

Möller, Julius

May 2023

Contents

1	General Procedure	2
2	Spimodfit Procedure	2
2.1	Fitting the Crab Nebula	2
2.2	Fitting other Sources	3
2.3	Fitting Custom Data	3
2.4	Spimodfit Background	4
3	Results: Crab Nebula	4
4	Results: Simulated Source	4
5	Background Analysis	9
6	Results: Pure Simulation	11
6.1	Consistency	11
6.1.1	Dynamic Binning	11
6.2	Different Powerlaws	11
6.3	A Second Source	11
6.4	Energy Range	11
6.5	Number of Energy Bins	11
6.6	Cluster Size	11

1 General Procedure

In contrast to previous approaches to working with INTEGRAL SPI, this method does not attempt to fit the background via physical models, and instead uses a profile likelihood to eliminate its effect in the source model fit. The underlying assumption that allows this to work is that background count rates in each energy bin remain constant for temporally and spatially near SPI Science Windows (SCWs). Here we distinguish between source counts, which describe all detected events directly originating from astronomical objects considered in our source model and background counts, which describe all other detector events.

This works in the following way: once we have split our set of SCWs into clusters within which we claim the assumption of constant background to be true and have defined a source model, a likelihood value can be calculated for each energy bin of each detector in each cluster. If one energy bin of one detector in one SCW with measuring time t has a source count rate s and a background count rate b , the probability P of measuring C counts follows a Poisson distribution:

$$P(C|b, s, t) = \frac{(t(b + s))^C \exp(-t(b + s))}{C!} \quad (1)$$

Equivalent energy bins in the same cluster are assumed to have the same background count rate b , so that the probability of measuring C_1 counts in one SCW and C_2 counts in another SCW is simply:

$$P(C_1, C_2|b, s_1, t_1, s_2, t_2) = P(C_1|b, s_1, t_1) \cdot P(C_2|b, s_2, t_2) \quad (2)$$

The likelihood \mathcal{L} for any energy bin of a detector within a cluster depends on all equivalent energy bins in that cluster, i.e. one term for each SCW in the cluster. If we have cluster of size two, the likelihood is described by:

$$\ln \mathcal{L}(C_1, C_2, t_1, t_2|b, s_1, s_2) = \ln P(C_1|b, s_1, t_1) + \ln P(C_2|b, s_2, t_2) \quad (3)$$

This expression allows us to solve for the probability distribution of b . A full and mathematically correct treatment of the background parameter b turns out to be very cumbersome with only marginal benefits. Hence, the profile likelihood is used by simply solving for the background count rate value b_M which maximizes the likelihood. This is easily done by setting the derivative of equation 3 to zero and solving for b . For a cluster size of two one finds:

$$b_M = \frac{1}{2} \left[\frac{C_t}{t_t} - s_t + \sqrt{\left(\frac{C_t}{t_t} - s_t \right)^2 + 4 \left(\frac{C_1 s_2 + C_2 s_1}{t_t} - s_1 s_2 \right)} \right] \quad (4)$$

where $C_t = C_1 + C_2$, $s_t = s_1 + s_2$, and $t_t = t_1 + t_2$. Using this maximum likelihood background b_M finally allows us to compute a likelihood value that is only dependent on the measured counts and the source model:

$$\ln \mathcal{L}(C_1, C_2, b_M, t_1, t_2|s_1, s_2) = \ln P(C_1|b_M, s_1, t_1) + \ln P(C_2|b_M, s_2, t_2) \quad (5)$$

The process for clusters with sizes larger than two is analogous. One such likelihood value exists for every energy bin of every detector in every cluster, such that the total likelihood value for any given source model is simply the sum of the logarithmic likelihoods over all energy bins, detectors, and clusters. Thus the source model may be fitted to the count data by applying an algorithm to maximize the likelihood.

2 Spimodfit Procedure

In order to be able to compare the results obtain using PySpi and the method described above, spimodfit will be used with the same dataset. For the sake of limiting sources of potential errors, the procedure for obtaining results using spimodfit will be described in more detail than might otherwise be deemed necessary. In general, the spimodfit cookbook found here was followed very closely. In all of the following sections only the SE Data was used by Spimodfit with an energy up to 600keV, compared to a possible 2MeV in the data used by Pyspi.

2.1 Fitting the Crab Nebula

1. On the ga05us.mpe.mpg.de PC, head into the directory:

```
/home/jmoeller/cookbook/SPI_cookbook/examples/Crab
```

2. In the file `spiselectscw.cookbook_dataset_02_0020-0600keV_SE.par` specify which revolution is to be used via the `fits_revolutions_list` and `revolutions_cond_value` parameters. Run the script:


```
./submit-spiselectscw_ga05us.sh cookbook_dataset_02_0020-0600keV_SE &
```
3. In the file `background_model_SE_02.pro` change the parameters `spidir`, `scw_file`, and `bgdir` to point towards the correct directories in the newly created directory `cookbook_dataset_02_0020-0600keV_SE`. Run this script:


```
idl idl-startup.pro background_model_SE_02.pro
```
4. In the file `spimodfit.fit_Crab_SE_02.par` change the parameters `counts_input_file`, `pointing_input_file`, `ebounds_input_file`, `deadtime-dol`, `gti-dol`, and `background_input_file` to point to the correct directories. Furthermore, specify the source catalogue directory:


```
source-cat-dol,s,h,"/home/jmoeller/cookbook/SPI_cookbook/cats/cat_crab.fits.gz"
```

 Leave everything else untouched and run the script:


```
./submit-spimodfit_v3.2_ga05us.sh fit_Crab_SE_02 clobber &
```
5. Inside the directory `fit_Crab_SE_02` run the script `./spimodfit_rmfgn.csh`. Back in the previous directory, edit the parameters `response_file` and `spectrum_file_01` in the file `adjust4threeML_SE_02.pro` to point at the correct directories, and run the script:


```
idl idl-startup.pro adjust4threeML_SE_02.pro
```
6. For convenience, transfer the files `specra_Crab.fits` and `spectral_response.rmfi.fits` onto local machine. Here the script found in


```
/home/jmoeller/cookbook/SPI_cookbook/spectral_analysis/3ml_fit_Crab_SE.py
```

 is used extract the fit parameters, where the source model can be changed and active measurements (i.e. energy range) can be changed as pleased.

2.2 Fitting other Sources

If we want to fit another source instead of the Crab, the following changes are made to the previously described steps:

1. Access the SPI source catalogue: `/home/jmoeller/cookbook/SPI_cookbook/cats/spi_cat.fits.gz`
2. Create a new file with only one (or more, depending on how many sources we want to fit) row from the source catalogue, representing one source. Edit the entries `RA_OBJ`, `DEC_OBJ`, and `NAME` to the coordinates and name of the new source. Other entries in the row are not used and irrelevant for the fit.
3. Place the new file in the same directory as the original catalogue.
Now repeat steps from section 2.1 with the following differences:
4. In step 2 from section 2.1 also change the parameters `select_PtgX_masks_chi_list_1` and `select_PtgX_masks_psi_list_1` to the coordinates of the new source. Note that these are galactic coordinates.
5. In step 4 from section 2.1 change `source-cat-dol` to instead point to the new source file.
6. The new spectrum output file will be names according to the new source name, which has to be reflected in the parameter `spectrum_file_01` from step 5 in section 2.1.

2.3 Fitting Custom Data

If we want to use Spimodfit to fit our own custom data, for example for simulated sources, we can once again repeat the steps from section 2.1 with one difference. After step 2 is completed the directory

```
/home/jmoeller/cookbook/SPI_cookbook/examples/Crab/cookbook_dataset_02_0020-0600keV_SE/spi
```

is created, which includes the relevant data files to be used in the fit. Here we can edit `evts_det_spec.fits.gz` to our custom data, which obviously has to take into consideration which SCW, detector, and energy bin each table entry corresponds to.

2.4 Spimodfit Background

For every fit, Spimodfit creates a background model. After step 3 in section 2.1 is completed the directory

/home/jmoeller/cookbook/SPI_cookbook/examples/Crab/cookbook_dataset_02_0020-0600keV_SE/spi/bg-e0020-0600

is created, containing the files `output_bgmodel-conti.fits.gz` and `output_bgmodel-lines.fits.gz`. These contain the expected number of counts for each energy bin from each detector in each SCW. In order to obtain a usable background model one can Poisson distribute the files and then add them together. Since some elements are negative, one can take the Poisson distribute the absolute values and multiply the result with the sign of the original number.

3 Results: Crab Nebula

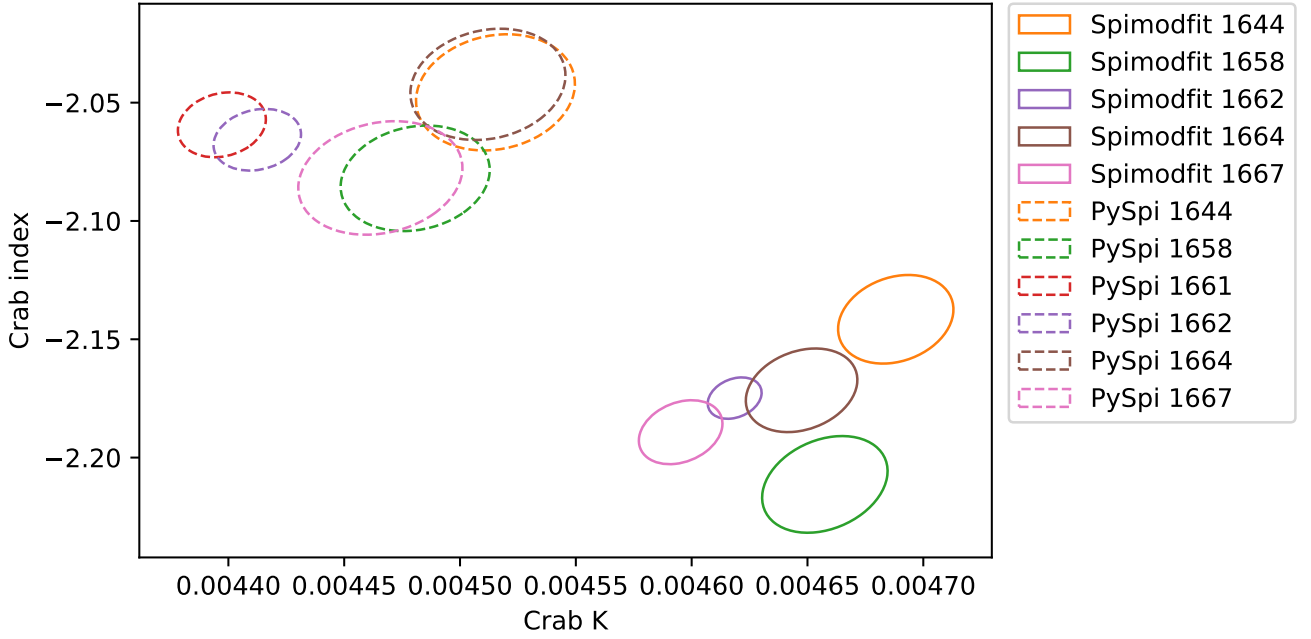


Figure 1

Figure 1 shows one standard deviation confidence intervals of several near SPI Revolutions, as fitted using PySpi and Spimodfit. Revolution 1661 was only fitted using PySpi since spimodfit rejected all SCWs due to "General Expression". The source model is a simple powerlaw

$$f(x) = K \frac{x^{index}}{piv} \quad (6)$$

where K , measured in $\text{keV}^{-1}\text{s}^{-1}\text{cm}^{-2}$ is the differential flux at the pivot value $piv = 40\text{keV}$, x is the photon energy in keV, and $index$ is the index of the powerlaw.

Both methods used the same energy range of 20-81.5keV, although it is worth pointing out that they use independent algorithms to select which SCWs are used in the fit. PySpi used a cluster size of two.

Although the two methods obtain different values for the index and normalization, we see comparable sizes of uncertainties as well as similar levels of compactness within the values from different revolutions.

The data used by PySpi was also subjected to a Posterior Predictive Check (PPC) analysis, allowing for revolution pairs that did not match the source model to be manually removed. This substantially improved the consistency of the fits.

4 Results: Simulated Source

The downside to fitting a real astronomical object such as the Crab nebula is that one can never be exactly sure what the true source parameter values should be. This is why it can be advantageous to simulate a source, feed

the simulated data back into the fit, and compare the produced source parameters to the now known true values. However, not only a simulated source is required, but a background as well. There are different approaches one may take to realize this. The first approach is to use an entirely real dataset from a SPI revolution as background, and add a simulated source on top:

1. Choose a data set to use as background. This was done once with real data, in this case a single SPI revolution. For this to work well, there should be as few bright sources within the field of view as possible, since these will interfere with the fit if not included in the source model. This step was also performed using the Spimodfit background spectrum, as described in section 2.4. Alternatively, one can also use a constant background spectrum for all SCWs. This may be realized by choosing one SCW and duplicating its counts to all other SCWs. The counts have to be redistributed according to a Poisson distribution and the lifetimes of the other SCWs has to be adjusted as well.
2. Define spectrum of the source to be simulated, as well as position. In this case powerlaws as shown in equation 6 are used, and the position of the sources is chosen to be central to as many SCWs in the revolution as possible.
3. Use the SPI Instrument Response Function (IRF) to calculate the expected source count rates in each energy bin in each detector for each SCW. Multiply by these be the respective lifetime (effectively the active measuring time) of the detector in the SCW, and finally draw a sample of a Poisson distribution with the respective means to attain the measured counts of the simulated source for each energy bin in each detector in each SCW.
4. These source count rates are then added to the background spectrum to acquire the total counts.

This dataset can then be used to fit the simulated source with both PySpi and Spimodfit. When using real data in step 1 this method produces a very realistic background, which in some cases it may be too realistic by including unwanted sources in the field of view that interfere with the fit. This is why it can be beneficial to use the Spimodfit background spectrum or a constant background spectrum instead. The results of this for revolutions 0374 and 1380 are shown in figures 2 and 3. Spimodfit was not given the data with a constant background spectrum, since its approach to modelling the background does not allow it to deal with arbitrary spectra.

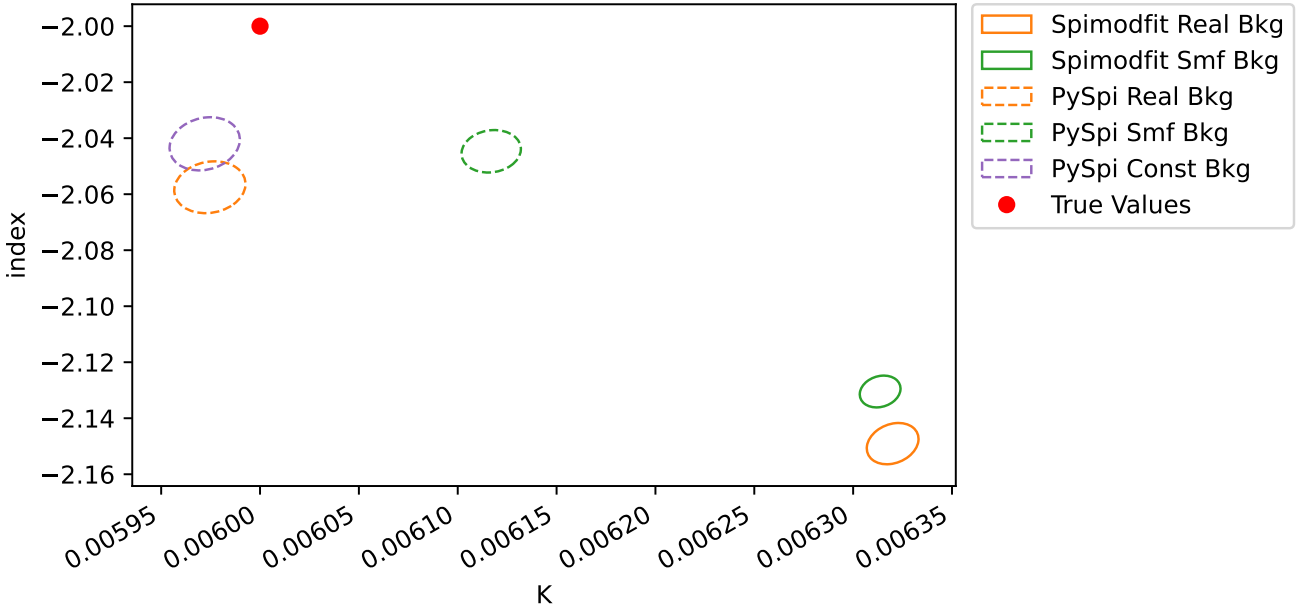


Figure 2: Simulated Powerlaw source with $K = 0.006 \text{ keV}^{-1} \text{ s}^{-1} \text{ cm}^{-2}$, $piv = 40 \text{ keV}$, and $index = -2$ for the different methods of generating a background spectrum fitted using PySpi and Spimodfit. This is based on revolution 0374 with the equatorial coordinates of the source at $RA = 10$, $DEC = -40$. As always, the ellipses show the confidence intervals of one standard deviation.

In general we see PySpi consistently perform better than Spimodfit. A couple of curious results stand out. One is that PySpi performs consistently worse with the Spimodfit background spectrum. In theory, one might not expect this since the Spimodfit background spectrum should not include any point sources in SPIs field of view, so that the fundamental assumption of constant background in SCW clusters should be well met. Another curiosity is that

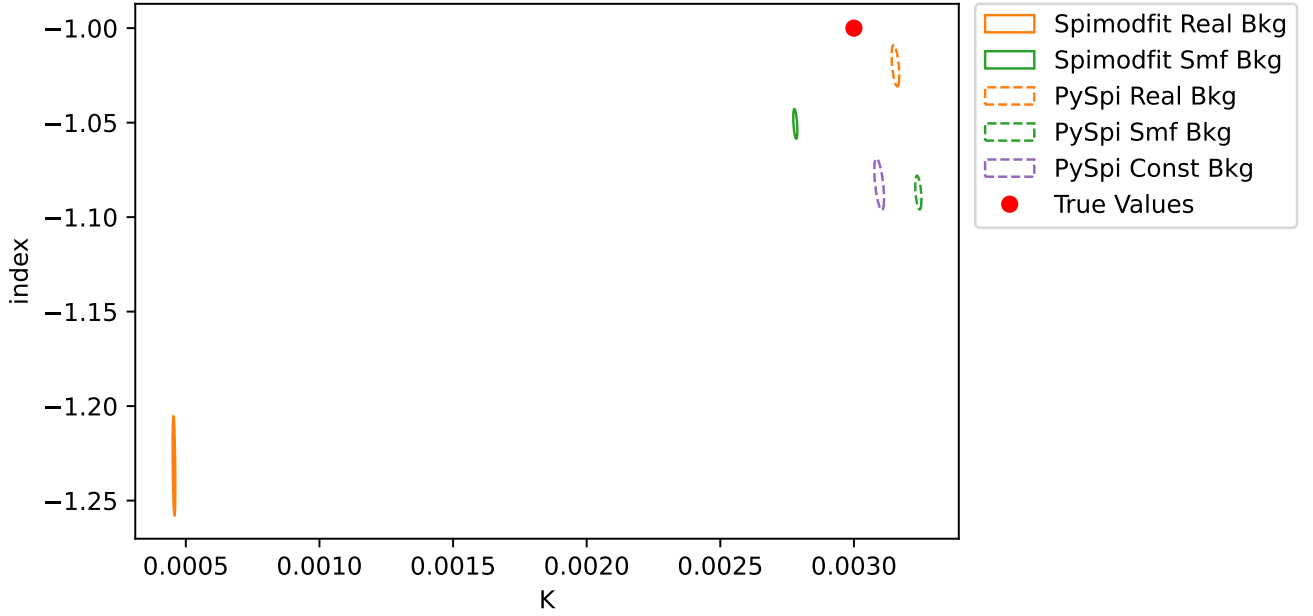


Figure 3: Simulated Powerlaw source with $K = 0.003\text{keV}^{-1}\text{s}^{-1}\text{cm}^{-2}$, $piv = 40\text{keV}$, and $index = -1$ for the different methods of generating a background spectrum fitted using PySpi and Spimodfit. This is based on revolution 1380 with the equatorial coordinates of the source at $RA = 155$, $DEC = 75$.

the effect of using Spimodfit with the data produced using the Spimodfit background spectrum has inconsistent results. For revolution 0374 this makes almost no difference to using the real data as background, where as in revolution 1380 the results improve dramatically. Finally, it is very curious how PySpi with a constant background spectrum does not perform as well as one might expect, since the fundamental assumption of constant background is perfectly met. This demonstrates systematic inaccuracies with the PySpi approach to fitting, which are further explored in section 6.

Finally, the count and source model spectra produced by the Spimodfit fits are shown in figures 4 to 7. Two possible signs of errors are the first energy bin is always significantly higher than predicted by the source model, and that the uncertainties for revolution 1380 with real background data are very high.

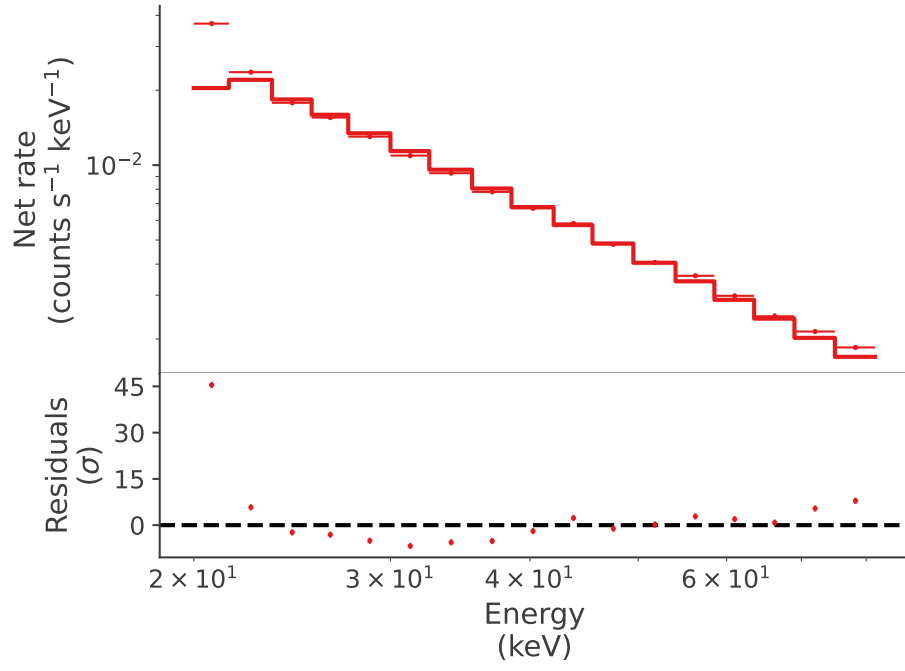


Figure 4: Spimodfit spectrum and source model for revolution 0374 using the simulated source and the real background spectrum.

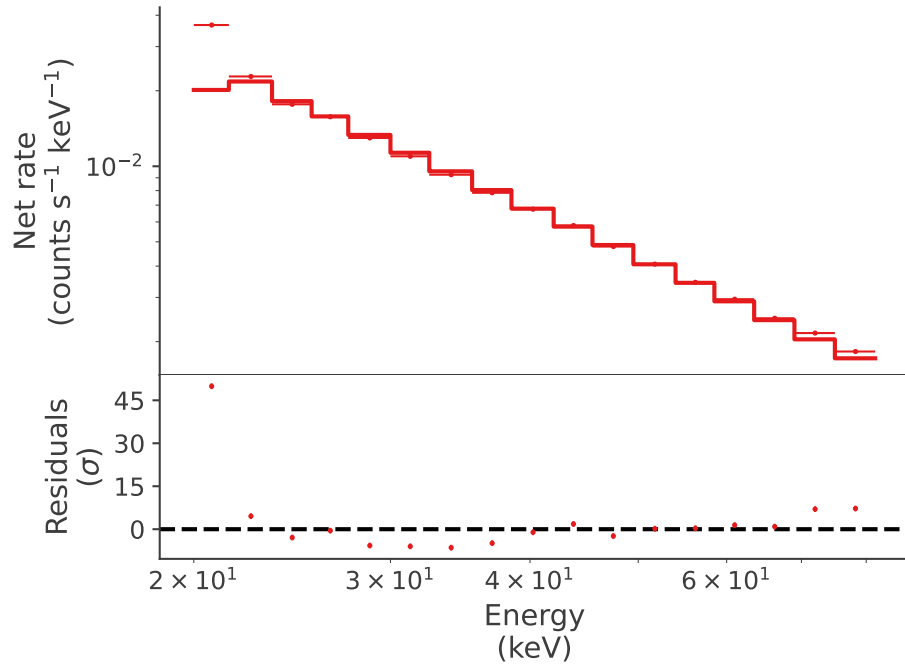


Figure 5: Spimodfit spectrum and source model for revolution 0374 using the simulated source and the spimodfit generated background spectrum.

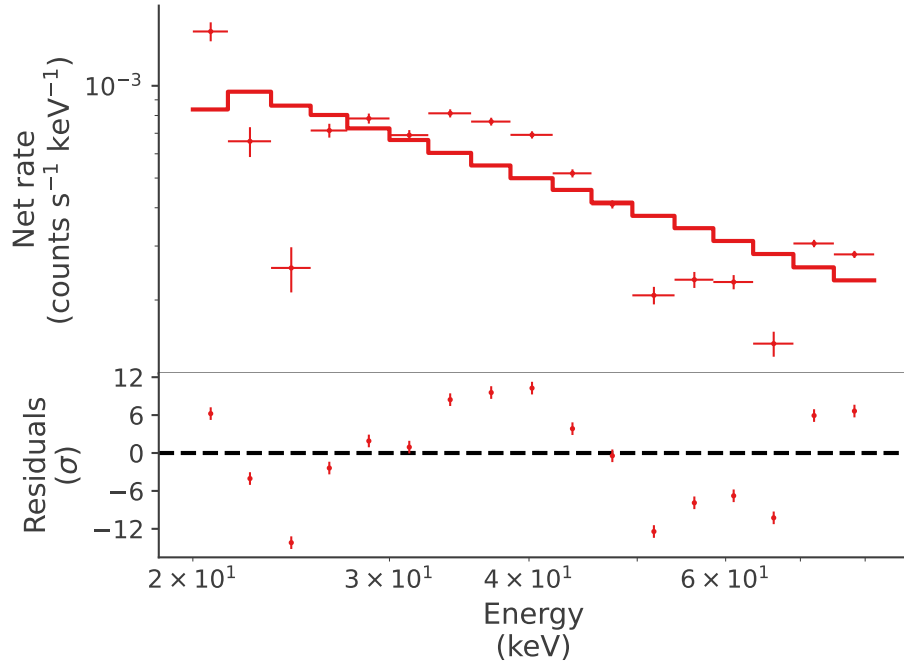


Figure 6: Spimodfit spectrum and source model for revolution 1380 using the simulated source and the real background spectrum.

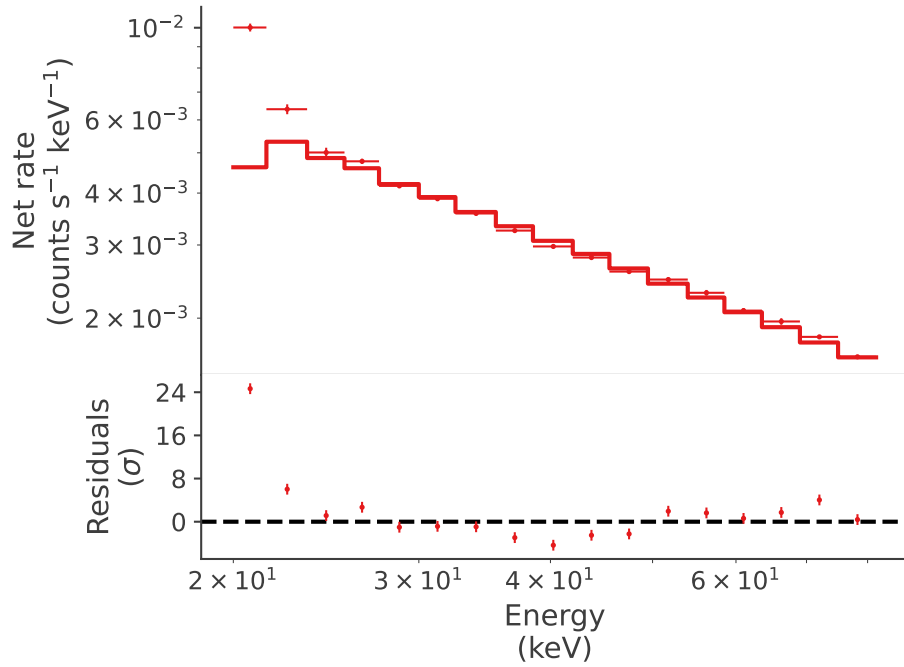


Figure 7: Spimodfit spectrum and source model for revolution 1380 using the simulated source and the spimodfit generated background spectrum.

5 Background Analysis

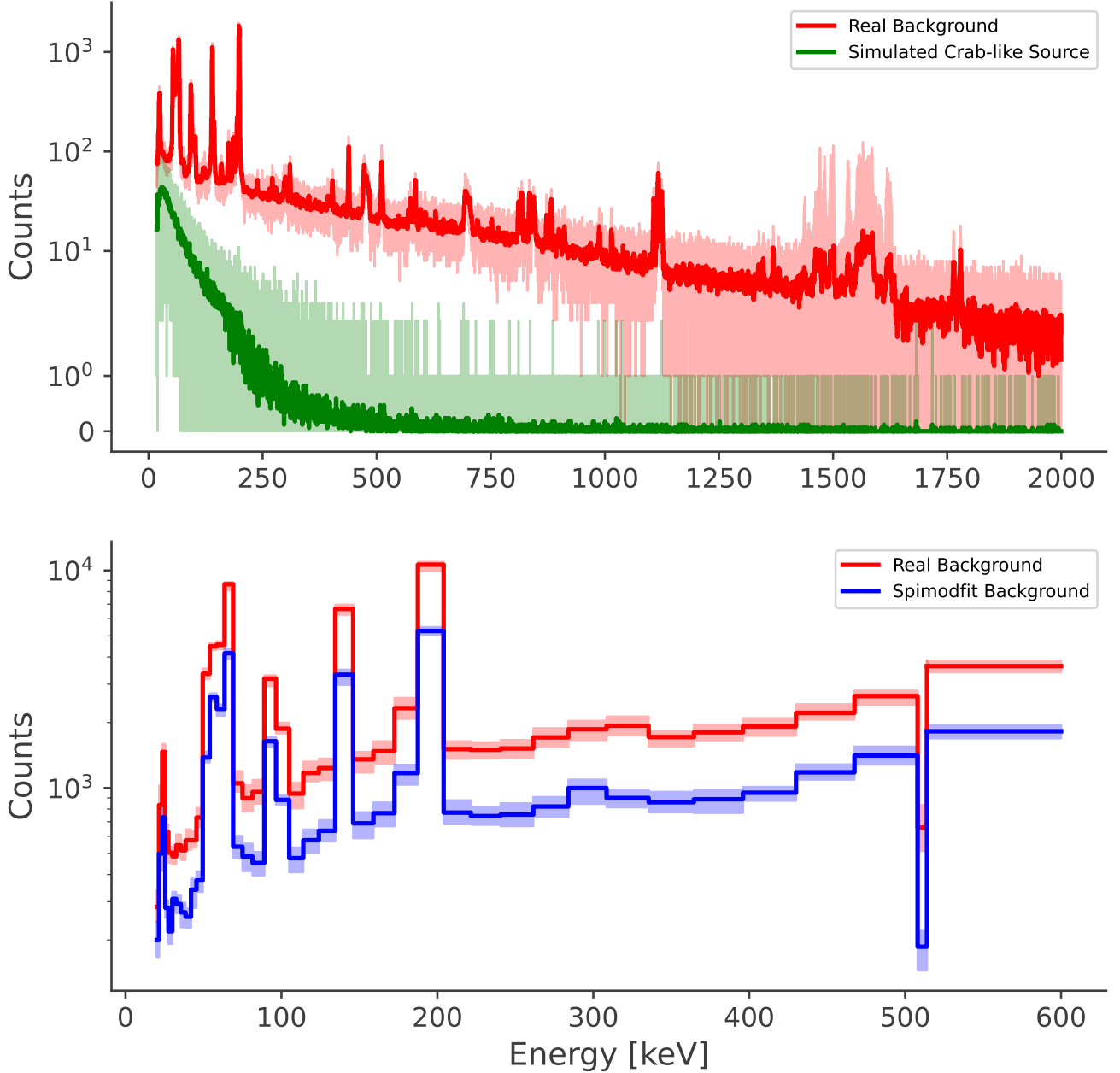


Figure 8: Background spectrum of SCW 03740002 in comparison to count rates from a simulated source with a powerlaw spectrum similar to that of the Crab nebula. In the bottom plot, the spectrum is rebinned so that it may be compared to the spimodfit background spectrum of the same SCW. The shaded areas show the range over all detectors, whereas the solid line shows the mean over all detectors.

The counts from SCW 03740002 are used very often as an example background spectrum, and duplicated to other SCWs in the revolution for a constant background. Hence, it can be beneficial to visualize this spectrum, as was done in figure 8. Two things worth noting are that the counts from the simulated Crab-like source are significantly lower than those from the background, especially at higher energies, and that the Spimodfit background spectrum, although very similar in shape, is consistently lower than the real background by a roughly constant factor.

While we are analyzing data spectra that we consider to be background-like (i.e. no point-like source in SPIs field of view), there is a very insightful statistical test we can do. After pairing up SCWs into clusters within a SPI revolution, we can calculate the variance of the counts in each energy bin of each detector of each SCW cluster, and then compare this to the variance we would expect if the two counts were Poisson sampled based on a common underlying background count-rate. The definition of the Bessel corrected variance of the two counts is standard, and the expected variance is also relatively straightforward to compute given the fact that variances of independent

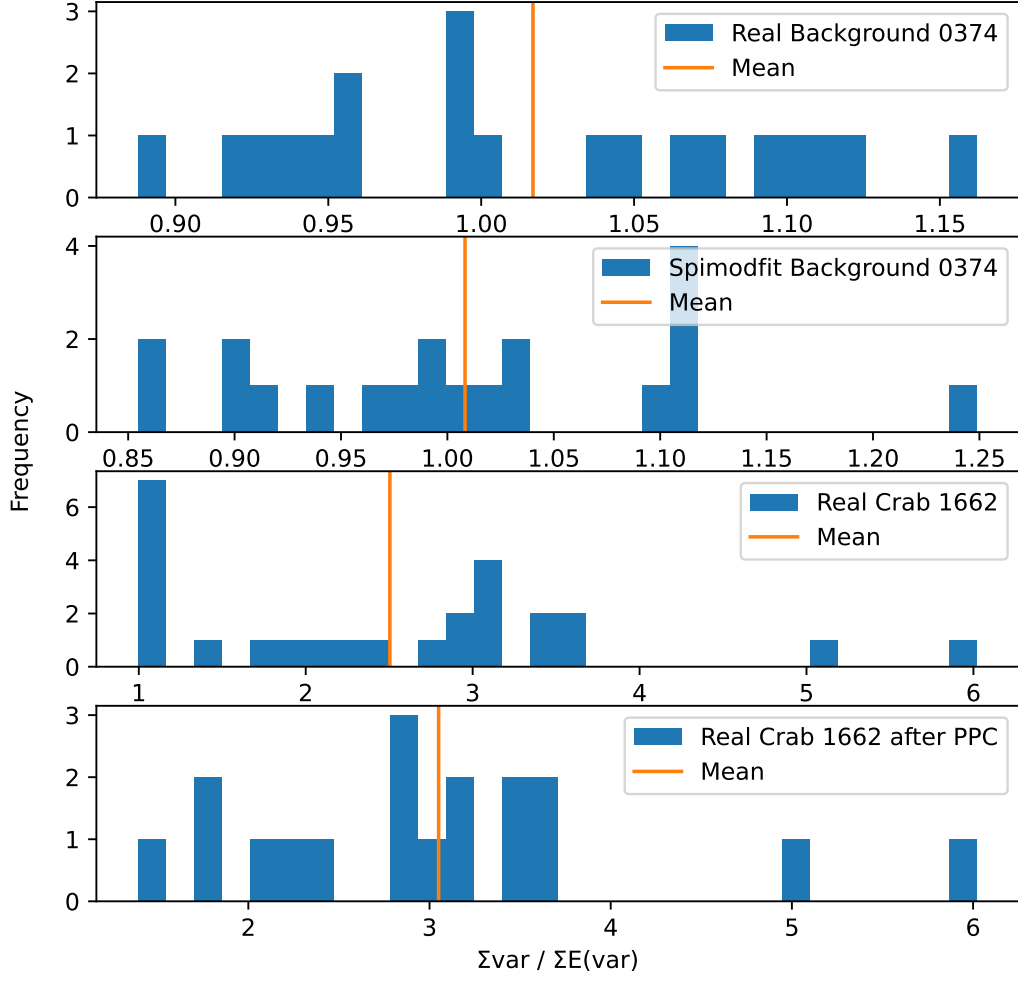


Figure 9: A histogram of the ratio of the variances to the expected variances based of the SCW clusters for different data sets. (see text)

distributions add together:

$$E(\text{var}) = (rt_1 - m)^2 + (rt_2 - m)^2 + \frac{r(t_1 + t_2)}{2} \quad (7)$$

where $r = \frac{C_1 + C_2}{t_1 + t_2}$ is the average count-rate in both energy bins, $m = \frac{C_1 + C_2}{2}$ is the average number of counts, and t_1 and t_2 are the lifetimes of the respective SCWs. We may then sum the variance and expected variance over all energy bins and all detectors for each cluster of SCWs and take their ratio to get an estimate of how well the fundamental assumption of a constant background count-rate within SCW clusters is met. A ratio $\text{var}/E(\text{var})$ around 1 means that the variances between the two SCWs are entirely explained by Poisson sampling with a common count-rate, in comparison to a ratio larger than 1 indicating a source in SPIs field of view, hence non-common count-rates for identical energy bins in the clustered SCWs.

This was plotted in figure 9 for the real data from revolution 0374 and the Spimodfit generated background spectrum for revolution 0374. We see that both are closely distributed around 1, meaning that the fundamental assumption of constant background rates within SCW clusters is well met, both within the real data and the spimodfit generated background.

Additionally, the same analysis was performed on revolution 1662, featuring many SCWs with the Crab Nebula

within 10 degrees of the center of SPIs field of view. As expected, we see that the ratio of the variances is now considerably higher. Curiously, there are still several clusters with a ratio close to 1, implying that there are no visible sources in SPIs field of view. Since the Crab nebula is always within 10 degrees of the field of view, this is impossible. It is unclear what happened in these SCW clusters, but whatever the cause, any sources are completely drowned out. A PPC analysis was conducted on this revolution, with notable outlier SCW clusters being removed from the fit. As expected, this removes exactly those SCW clusters with a variance ratio close to 1. This highlights the importance of performing a PPC analysis when using PySpi.

6 Results: Pure Simulation

6.1 Consistency

6.1.1 Dynamic Binning

6.2 Different Powerlaws

6.3 A Second Source

6.4 Energy Range

6.5 Number of Energy Bins

6.6 Cluster Size