Diffuse X-Ray Emission Cleaning Algorithm

Carter Rhea

January 28, 2019

1 Introduction

Instead of simply following the common data analysis workflow which creates a Level II data product from a Level I data product, I will follow the recommendations [1].Because our object is so X-ray faint (which we can tell from completing a preliminary reduction using `chandra_repro`), we must take extra care to reduce our data and ensure that we are accurately seperating the source counts from the background counts; thus, there are several additional constraints which we must contend with...

The basic cleaning flow is as follows:

1. Update Aspect File with correct positioning
2. Discard times exhibiting background flares
3. Remove streak events
4. Create new badpixel file
5. Process event with new badpixel file (Level 1.5)
6. Construct Level II event file by filtering on grade, status, and GTI

In addition to the following steps we must *reset* the *evt1.fits* file to allow us to apply custom cleaning; this is quickly done with the following command: `acis_clear_status_bits`.

1.1 Astromety

Because we are looking at a faint extended object over several epochs, it is worthwhile to ensure the astrometric alignment is as accurate as possible (though *Chandra* data in general is accurate up to a %90); we do this to ensure that the extraction aperatures use the exact same well-calibrated celestial position. Following [?] (and in turn the classic astromety thread from CXC), we align our observation with the known location of *SpARCS1049+56* and then correct the aspect file using `CIAO` – *wcs_update*. The structure is as follows[2]:

1. Compute Source Location in Observation using `dmstat`
2. Use `dmcopy` to calculate the physical coordinates of the source
3. Calculate the offset by subtracting the Measured Position from the Reference Position
4. Apply changes to aspect file and event file using `wcs_update`

To do this requires first running `fluximage` on the *evt2.fits* file to calculate an exposure map and broad-threshed image.

[1] https://arxiv.org/abs/1003.2397

[2] See CXC suggestions for `reproject_aspect`

1.2 Discard Background Flares

Since we have already analysed the background for flare events, we can now simply apply those times we created with the `lc_clean` command...

```
dmcopy "evt2.fits[@lc_ccd.gti]" evt2_lc.fits
```

1.3 Remove Streak Events

We will search for a remove streak events **before** we create our new bad-pixel file b ecause the CXC team and [?] suggest that doing so enhances detection of streaks in faint and diffuse images[3] with the command `destreak`.

1.4 Bad Pixel File

We must create a bad pixel file that will through out pixels with a bad bais value and known bad pixels as well as hot pixels and afterglows.

1. Use `acis_build_badpix` to identify known bad pixels on the CCDs and those with bad bias values
2. Find hot pixels and detect afterglow events using `acis_detect_afterglow`
3. Mark nearby pixels and sort the bad pixel list using again `acis_build_badpix`

We also consider using a different bitflag – note that the bitflag reads right to left instead of left to right. Please see figure 1 for details on each individual bit.

Bit	Reason a pixel or column is identified as "bad"
0	Known bad pixel
1	Known bad column
2	Bias-parity error. The start time is specified in the column TIME.
3	Bias = 4095 adu.
4	Bias = 4094 adu.
5	CHIPX = 1 or 1024 (i.e. inactive columns). [obsolete]
6	In timed-exposure mode observations, CHIPY = 1 or 1024 (i.e. inactive rows). [obsolete]
7	User specified
8	The pixel is horizontally, vertically or diagonally adjacent to a bad pixel.
9	For TIMED VFAINT mode observations, CHIPX = 2 or 1023 or CHIPY = 2 or 1023 (i.e. avoid having a 5 x 5 event island overlap the edge of a CCD). [obsolete]
10	For TIMED VFAINT mode observations, the pixel is horizontally, vertically or diagonally adjacent to a pixel for which bit eight is set to one. [obsolete]
11	CHIPX = 512 or 513. Events in these columns are often due to cosmic rays.
12	CHIPX = 256, 257, 768 or 769. Events in these columns are often due to cosmic rays.
13	FEP0 problem
14	Hot pixel
15	Cosmic-ray afterglow. The start and stop times of the afterglow are specified in the columns TIME and TIME_STOP, respectively.
16	Bad bias value
17-31	Unused

Figure 1: Bax pixel bit-by-bit explanation[4]

Since we are dealing with a faint source, we choose to use the following bitflag:

00000000000000022221100020022212

Adoption of this particular bitflag ensures that, in the case that errant bad columns appear through a faint source, adjacent columns are not also flagged; thus, we retain a higher number of source counts.

[3] http://cxc.harvard.edu/ciao/threads/createL2/

1.5 VFAINT Background Cleaning

I believe that our observation is an excellent candidate for *VFAINT background cleaning*[5] due to the lack of bright features within the region of interest[6].

Simply put, the *VFAINT background cleaning* algorithm capitalizes on the fact that in `VFAINT` mode *Chandra* creates 5×5 pixel islands rather than the standard 3×3 pixel islands thus enabling a more thorough detection of cosmic ray events by examining the pulse heights in the larger islands. However, this method has proven to often lead to the rejection of good events in especially bright regions of the image. Nonetheless, our image lacks strong – bright – feautures and is therefore an excellent candidate.

1.6 Final Steps and Comments

Using our newly created bad pixel file we can run `acis_process_events` to create a Level 1.5 event fits, *evt1_5.fits*. We can now create our Level II event file by filtering bad grades using `dmcopy` and applying our good time intervals (GTIs) with the following two commands...

```
dmcopy "evt1_5.fits[EVENTS][grade=0,2,3,4,6,status=0]" flt_evt1_5.fits
dmcopy "flt_evt1_5.fits[EVENTS][@acis_flt1.fits][cols -phas]" repro_evt2.fits
```

While this approach may seem aggressive, several studies have demonstrated its efficacy in cleaning faint and extended X-ray sources[?][?][?].

1.7 Programs to Run

In order to streamline the process please follow this list of events in order:

1. Run `CCD_split.py`; this will create a *Background* folder containing individual CCD fits files.

2. Inspect CCD of interest and locate sources. Places source in *.reg* file for exclusion. Ensure that the *.reg* file is in the *Background* folder.

3. Create clean background fits file with the `dmcopy` (see 1.2).

4. Run `CreateLightcurve.py` to general good time file (*gti*).

5. Finally run `FaintCleaning.py` to clean data!

[5] http://cxc.harvard.edu/ciao/why/aciscleanvf.html
[6] http://cxc.harvard.edu/cal/Acis/Cal_prods/vfbkgrnd/index.html