Using the ACIS "Blank–Sky" Background Files

CIAO 3.4 Science Threads

Table of Contents

Using the ACIS "Blank–Sky" Background Files

CIAO 3.4 Science Threads

Overview

Last Update: 14 Sep 2007 – updated for CALDB 3.4.1: new ACIS background files

Synopsis:

The ACIS calibration team has completed a detailed study of the ACIS background. Part of this study has produced a set of "blank–sky" datasets which can be used, for instance, for analyses where the object of interest fills the entire chip.

Purpose:

To create background spectra tailored to a specific observation for spectral fitting or images for spatial analyses.

Read this thread if:

you are analyzing an ACIS imaging observation and the source is so large that you can not estimate the background from your dataset.

Calibration Updates:

Older calibration updates for this thread are listed in the Previous CALDB updates section.

- *CALDB v3.4.1* *(14 Sep 2007):*

  This set of ACIS background files includes:

  - files for −120 C data on ACIS−5 or ACIS−7 with CTI_APP = PPPPPBPBPP header keyword value (i.e. parallel and serial CTI applied to the back–illuminated chips)
  - CTI−corrected files for ACIS−8 (ACIS−S4).
  - upgrades to the files for chips 1−3, 5−7, and 8 with ACIS−S in the focal plane. These files include the newest CTI and TGAIN corrections.

    The non−CTI−corrected background files for these chips have the TGAIN applied. This matches how observational data at −120 C without the CTI−correction would be processed.

- *CALDB v3.4.0* *(16 May 2007):*

  The newest versions of the ACIS blank−sky background files are now available in the CALDB. There are several improvements in these files:

  - The background data have been divided so that there is one chip per file. A unique background file is returned for each chip that contains data in the event file. Users should no longer see the "`WARNING: 2 CALDB files found. Using the first`" error

when looking up a background file.

♦ The CTI–corrected files also have the time–dependent gain (TGAIN) correction applied.

♦ The background files have reasonable TSTART and TSTOP values in the header. This eliminates a CALDB lookup problem that was primarily seen when using CIAO response tools, e.g. via running `specextract`.

♦ There is no TIME column. The TIME column was previously removed before reprojecting the background file, and so was never used in analysis.

♦ The the FILTER header keyword has been removed. Previous versions of the file had an incorrect filter value, which caused some CALDB lookups to fail (usually when a response tool was trying to find an effective area file).

The ACIS background files are not included in the main CALDB tarfile. There is a separate background files tarfile available from the CALDB Download page.

Related Links:

- ACIS background memos page, in particular the discussion of the ACIS background page.
- On the discrepancy between Chandra and XMM temperature profiles for A1835 (astro–ph/0205333) by Markevitch: the ACIS background is found to play a significant part in this analysis.
- Absolute measurement of the unresolved cosmic X–ray background in the 0.5–8 keV band with Chandra (astro–ph/0512542) by Hickox & Markevitch: shows how to use the ACIS–stowed background for analysis of background–critical objects, such as the Cosmic X–ray background.
- The paper "*Chandra Spectra of the Soft X–ray Diffuse Background*" by Markevitch et al. (astro–ph/0209441) presents an exploratory Chandra ACIS–S3 study of the diffuse component of the Cosmic X–ray Background in the 0.3–7 keV band for four directions at high Galactic latitudes, with emphasis on details of the ACIS instrumental background modeling.
- Analysis Guide: Extended Sources

Proceed to the HTML or hardcopy (PDF: A4 | letter) version of the thread.

Get Started

Sample ObsIDs used: 1838 (ACIS–S, G21.5–09); 1712 (ACIS–S, 3C 273)

File types needed: evt2; asol1

The 1838 event file used in this thread has been reprocessed to apply the newest gain file; the CTI correction has not been applied. In these examples, we shall restrict attention to the `ACIS-S3` chip and the 0.3 to 10 keV energy range:

```
unix% dmcopy "1838_evt2.fits[energy=300:10000,ccd_id=7]" evt2_c7.fits
unix% dmcopy "evt2_c7.fits[bin sky=8]" img8_c7.fits
```

ObsID 1712 is used as an example of a strong background flare; this is the same dataset used in the Filtering Lightcurves thread.

The `acis_bkgrnd_lookup` script requires that the input file contains a CCD_ID column, SIM_Z header keyword, and FP_TEMP header keyword. If your data has not been recently reprocess, the FP_TEMP keyword may not be included in the level 2 event file. If it is missing from the file header, get the value from the evt1.fits file and use `dmhedit` to add it to the evt2.fits file; refer to the focal plane temperature FAQ has more information.

```
unix% dmkeypar evt2_c7.fits FP_TEMP echo+
153.446014
```

Download the background files

The ACIS "blank–sky" background files are not included in the main CALDB download file, due to the substantial filesize. If you have not already installed them, you must download the background files tarfile from the CALDB Download page and install it before continuing with this thread.

The background files are installed in the $CALDB/data/chandra/acis/bcf/bkgrnd/ directory.

Download the scripts

This thread uses the acis_bkgrnd_lookup and lc_clean.sl scripts. The most recent version of acis_bkgrnd_lookup is v1.12 (29 March 2007):

```
unix% grep Id `which acis_bkgrnd_lookup`
% $Id: acis_bkgrnd_lookup,v 1.12 2007/03/29 16:43:34 egalle Exp $
```

The most recent version of lc_clean.sl is v1.9 (20 June 2003):

```
unix% grep Id $ASCDS_CONTRIB/share/slsh/local-packages/lc_clean.sl
% $Id: lc_clean.sl,v 1.9 2003/06/20 14:23:12 dburke Exp $
```

Note that $ASCDS_CONTRIB/share/slsh/local-packages/ is the default path in the standard CIAO scripts installation; see the Scripts page for more information. *Please check that you are using the most recent version before continuing.* If you do not have the scripts installed or need to update to a newer version, please refer to the Scripts page.

Finding the Correct File

Naming scheme

The ACIS background datasets are stored in the CALDB at $CALDB/data/chandra/acis/bcf/bkgrnd/. They are indexed by focal plane temperatures, aimpoint (i.e. ACIS–I or ACIS–S), and chip number:

```
unix% ls -1 $CALDB/data/chandra/acis/bcf/bkgrnd/
acis0D2000-12-01bgstow_ctiN0004.fits
acis0iD1999-09-16bkgrndN0002.fits
acis0iD2000-01-29bkgrnd_ctiN0005.fits
acis0iD2000-01-29bkgrndN0003.fits
acis0iD2000-12-01bkgrnd_ctiN0004.fits
acis0iD2000-12-01bkgrndN0002.fits
acis1D2000-12-01bgstow_ctiN0004.fits
acis1iD1999-09-16bkgrndN0002.fits
acis1iD2000-01-29bkgrnd_ctiN0005.fits
...
```

The naming scheme for CTI–corrected data is:

```
acis<chip><aimpoint>D<date>bkgrnd_ctiN<version>.fits
```

For uncorrected data, the naming is:

```
acis<chip><aimpoint>D<date>bkgrndN<version>.fits
```

The files which have "`bgstow`" in place of "`bkgrnd`" in the filename are particle–only background files created from data taken when the HRC was in the focal plane (e.g. ACIS was "stowed"). There is no ACIS–S/ACIS–I distinction for these data, so an "<aimpoint>" is not included in the filename.

Using acis_bkgrnd_lookup

The `acis_bkgrnd_lookup` script makes it easy to find an ACIS background file that matches your data. The script takes an input event file and returns a list of background files for all the ACIS chips in the file that have events on them:

```
unix% acis_bkgrnd_lookup evt2_c7.fits
/soft/ciao/CALDB/data/chandra/acis/bcf/bkgrnd/acis7sD2000-01-29bkgrndN0005.fits
```

This file only has data on ACIS–7 (ACIS–S3), so a single file is returned. The path reflects where the CALDB is installed on your system; here, $CALDB is set to `/soft/ciao/CALDB`.

In addition to being printed to the screen, the list of background files is also stored in the `outfile` parameter:

```
unix% pget acis_bkgrnd_lookup outfile
/soft/ciao/CALDB/data/chandra/acis/bcf/bkgrnd/acis7sD2000-01-29bkgrndN0005.fits
```

The input file can include a Data Model filter as long as the resulting file is still a table with a ccd_id column, such as "`acisf01838N001_evt2.fits[ccd_id=7]`". In this example, we use a spatial region to filter an event file which has been CTI–corrected. The resulting background file(s) will be chosen to match the CCDs that are covered by the circle:

```
unix% acis_bkgrnd_lookup "1838_ctiapp.fits[sky=circle(6072.5,4632.5,320)]"
/soft/ciao/CALDB/data/chandra/acis/bcf/bkgrnd/acis1sD2000-01-29bkgrnd_ctiN0005.fits
/soft/ciao/CALDB/data/chandra/acis/bcf/bkgrnd/acis3sD2000-01-29bkgrnd_ctiN0005.fits
```

Two files are returned, so the outfile parameter contains a comma–separated list:

```
unix% pget acis_bkgrnd_lookup outfile
/soft/ciao/CALDB/data/chandra/acis/bcf/bkgrnd/acis1sD2000-01-29bkgrnd_ctiN0005.fits,
/soft/ciao/CALDB/data/chandra/acis/bcf/bkgrnd/acis3sD2000-01-29bkgrnd_ctiN0005.fits
```

No matching file and identical lookup cases

There are a number of known cases where `acis_bkgrnd_lookup` cannot find a background file for the dataset and exits with this error:

```
# acis_bkgrnd_lookup: ERROR: Unable to find an ACIS background file to match <filename>
```

There are *no background files* for:

- non–CTI–corrected data on ACIS–1
- any data on ACIS–4
- CTI–corrected data on ACIS–9
- ACIS–0 data, when ACIS–S is in the focal plane (a SIM_Z limit is exceeded)

There are also a few cases that result in identical lookup results:

- For the front–illuminated (FI) chips, there is no difference between:

  ♦ CTI_APP = PPPPPNPNPP
  ♦ CTI_APP = PPPPPBPBPP

since the parallel CTI–correction is applied to all FI chips either way.
- For the back–illuminated (BI) chips, there is no difference between any of these:

  ♦ CTI_APP = NNNNNNNNN
  ♦ CTI_APP = PPPPPNPNPP
  ♦ CTI_CORR = YES
  ♦ CTI_CORR = NO

since no CTI correction is applied to BI chips for any of those configurations.

Make a Local Copy of the Background File

The background file for the `evt2_c7.fits` dataset is
`$CALDB/data/chandra/acis/bcf/bkgrnd/acis7sD2000-01-29bkgrndN0005.fits`. Make
a local copy of the file:

```
unix% cp \
      $CALDB/data/chandra/acis/bcf/bkgrnd/acis7sD2000-01-29bkgrndN0005.fits \
      bgevt2_c7.fits
```

The tailor the background file to your dataset section explains how to what data processing is required in order
to use the background in the analysis. A "cleaned" event file is needed for that step, so we first have to remove
any background flares from the event data.

Remove Background Flares from the Event File

As discussed in the ACIS background page, the background datasets were created for quiescent–background
periods. It is therefore important that you filter your dataset to remove periods of background flares. The
selection of a "true" background can be critical, as discussed in M. Markevitch's memo (astro–ph/0205333).

We use the `lc_clean.sl` script to calculate a GTI for the dataset.

Remove bright sources

The background fields were filtered using a maximum rate that is 20 percent higher than the quiescent
background level. It is therefore necessary to remove any bright sources from your dataset. For this example
we have chosen the regions to exclude by eye, using `ds9` and the image `img8_c7.fits`, saving the file in
CIAO format. The resulting region file – `sources.reg` – is shown on the image in Figure 1 📷 and listed
below:

```
unix% cat sources.reg
# Region file format: CIAO version 1.0
circle(4072.5,4252.5,400)
circle(4604.5,4804.5,48)
```

If any flux is left from the source, you should reduce the `max_scale` parameter of `lc_clean()` to account
for this residual flux. For example, compare the expected background rate from the background files with
those of your dataset within the same spatial region.

Now we create a copy of the event file, excluding the sources, using the DM filtering syntax:

```
unix% dmcopy "evt2_c7.fits[exclude sky=region(sources.reg)]" evt2_c7_bg.fits
```

Make a Local Copy of the Background File

Create the lightcurve

Note: in this example, we do not apply any energy filter to the lightcurve; we use the full energy range of the dataset (0.3–10 keV here). This is not necessarily the optimal energy band to use for your observation.

The <u>dmextract</u> tool is used to create the lightcurve of the selected region. Since the background datasets were created using a bin size of 259.28 seconds, we use the same:

```
unix% punlearn dmextract
unix% pset dmextract infile="evt2_c7_bg.fits[bin time=::259.28]"
unix% pset dmextract outfile=evt2_c7_bg.lc
unix% pset dmextract opt=ltc1
unix% dmextract
Input event file  (evt2_c7_bg.fits[bin time=::259.28]):
Enter output file name (evt2_c7_bg.lc):
```

The file evt2_c7_bg.lc contains the lightcurve for the data. You can check the parameter file that was used with plist dmextract.

Run the lc_clean.sl script

The S–Lang script lc_clean.sl selects those regions of the lightcurve that do not contain flares. The algorithm used is taken from the lc_clean program created by Maxim Markevitch, and is different from that used by the analyze_ltcrv.sl script in the Filtering Lightcurves thread.

The S–Lang script must be loaded into *ChIPS* (or *Sherpa*) before it can be run (this step is only necessary once per *ChIPS*/*Sherpa* session):

```
unix% chips

Welcome to ChIPS, version CIAO 3.4
Copyright (C) 1999-2003, Smithsonian Astrophysical Observatory

chips> () = evalfile( "lc_clean.sl" )
```

or by loading the code using the --slscript option:

```
unix% chips --slscript lc_clean.sl
```

Here we use the default parameter values, except that we set the verbose flag to 1 to produce extra screen output:

```
chips> lc->verbose = 1
chips> lc_clean( "evt2_c7_bg.lc" )
Version of code: 1.9
Parameters used to clean the lightcurve are:
  mean      = NULL
  clip      = 3
  max_scale = 1.2
  max_sigma = NULL
  minfrac   = 0.1
  outfile   = NULL
  verbose   = 1

Total number of bins in lightcurve    = 38
Max length of one bin                 = 255.997 s
Num. bins with a smaller exp. time    = 9
Number of bins with a rate of 0 ct/s = 7

Calculated an initial mean (sigma-clipped) rate of 0.489577 ct/s
```

Create the lightcurve

```
Lightcurve limits use a scale factor of 1.2 about this mean
Filtering lightcurve between rates of 0.407981 and 0.587492 ct/s
Number of good time bins (drawn in green) = 29
Mean level of filtered lightcurve = 0.488992 ct/s

chips> print postfile evt2_c7_bg.eps
```

The "print postfile" command creates a postscript version of the *ChIPS* display window, as shown in Figure 2 📷. The green dots (upper plot) and line (lower plot) indicate those regions considered to be "good" by the algorithm. For this observation the background rate appears stable.

In order to create a GTI file, we need to specify an output filename; here we use "evt2_c7_bg.gti" (and reset the verbose tag to reduce the amount of information printed to the screen), then re–run the filtering:

```
chips> lc->verbose = 0
chips> lc->outfile = "evt2_c7_bg.gti"
chips> lc_clean( "evt2_c7_bg.lc" )
Total number of bins in lightcurve   = 38
Max length of one bin                = 255.997 s
Num. bins with a smaller exp. time   = 9
Number of bins with a rate of 0 ct/s = 7

Calculated an initial mean (sigma-clipped) rate of 0.489577 ct/s
Lightcurve limits use a scale factor of 1.2 about this mean
Filtering lightcurve between rates of 0.407981 and 0.587492 ct/s
Number of good time bins (drawn in green) = 29
Mean level of filtered lightcurve = 0.488992 ct/s

Creating GTI file
Created: evt2_c7_bg.gti

chips> quit
```

Since we specified an output file, the lc_clean() function has run dmgti, using the calculated range, and created the file evt2_c7_bg.gti. You can check the parameter file that was used with plist dmgti.

An example of a strong flare

The previous example showed an essentially stable background level. ObsID 1712, which is also used in the Filtering Lightcurves, contains a major flare event. First we create a lightcurve of the background of this observation:

```
unix% dmcopy "acisf01712N002_evt2.fits[energy=300:10000,ccd_id=7]" evt2_flare.fits

unix% cat exclude.reg
# Region file format: CIAO version 1.0
rotbox(4200.3328,4137.9892,1129.5056,74.07019,24.22333)
circle(4076.5,4088.5,316)
circle(4296.5,5024.5,48)

unix% punlearn dmextract
unix% pset dmextract infile="evt2_flare.fits[exclude sky=region(exclude.reg)][bin time=::259.28]
unix% pset dmextract outfile=flare.lc
unix% pset dmextract opt=ltc1
unix% dmextract
Input event file  (evt2_flare.fits[exclude sky=region(exclude.reg)][bin time=::259.28]):
Enter output file name (flare.lc):
```

When lc_clean() is run on this lightcurve, the routine finds that the filtered lightcurve contains less than 10 percent of the original exposure, and so stops (this limit can be changed by setting the variable lc->minfrac to something other than 0.1):

An example of a strong flare   9

```
unix% chips --slscript lc_clean.sl

Welcome to ChIPS, version CIAO 3.4
Copyright (C) 1999-2003, Smithsonian Astrophysical Observatory

chips> lc_clean("flare.lc")
Total number of bins in lightcurve   = 118
Max length of one bin                = 255.997 s
Num. bins with a smaller exp. time   = 12
Number of bins with a rate of 0 ct/s = 10

Calculated an initial mean (sigma-clipped) rate of 1.66018 ct/s
Lightcurve limits use a scale factor of 1.2 about this mean
Filtering lightcurve between rates of 1.38348 and 1.99221 ct/s
Number of good time bins (drawn in green) = 4
Mean level of filtered lightcurve = 1.51955 ct/s

** ERROR **
 Fraction of bins that are good is below limit of 0.1

 lc_clean("flare.lc");
```

which produces Figure 3 .

We can use *ChIPS* commands to manipulate the plots to examine the lightcurve in more detail. In the following we switch to the first (i.e. top) plot and zoom in on the y axis (between 0 and 2 count/s) and then the histogram (plot 2) and filter on the same range in x:

```
chips> d 1 limits y 0 2
chips> d 2 limits x 0 2
```

The resulting plot is shown in Figure 4 .

The histogram of the rates suggest that the mean rate is close to 0.7 count/s; we re–run `lc_clean()`, fixing the mean to this value:

```
chips> lc->mean = 0.7
chips> lc_clean("flare.lc")
Total number of bins in lightcurve   = 118
Max length of one bin                = 255.997 s
Num. bins with a smaller exp. time   = 12
Number of bins with a rate of 0 ct/s = 10

Using a fixed mean rate of 0.7 ct/s
Lightcurve limits use a scale factor of 1.2 about this mean
Filtering lightcurve between rates of 0.583333 and 0.84 ct/s
Number of good time bins (drawn in green) = 58
Mean level of filtered lightcurve = 0.693829 ct/s
```

The resulting plot is shown in Figure 5 .

See the comments in the lc_clean.sl file for more information on changing the filtering scheme used by the routine. Maxim Markevitch's Cookbook contains advice on filtering lightcurves with strong flares.

Exit *ChIPS* before continuing:

```
chips> quit
```

Filter the event list

We now filter the event list using the new GTI information, and check to see how the exposure time has changed

```
unix% dmcopy "evt2_c7.fits[@evt2_c7_bg.gti]" evt2_c7_clean.fits

unix% dmkeypar evt2_c7.fits EXPOSURE echo+
7854.4664748687

unix% dmkeypar evt2_c7_clean.fits EXPOSURE echo+
7342.4727938257
```

Tailor the background file to your dataset

Matching calibration to the event data

The background files were generated using a gain file that may not match the one used for your observation or may not be up to date. The GAINFILE keyword in the header of the background files should match that of your source file:

```
unix% dmkeypar evt2_c7_clean.fits GAINFILE echo+
acisD2000-08-12gainN0005.fits

unix% dmkeypar bgevt2_c7.fits GAINFILE echo+
acisD2000-01-29gainN0005.fits
```

The two gain files should have the same date and version number in the filename, as shown here.

If they do not match, reprocess the background file with `acis_process_events`, specifying the exact gainfile that was used on the data.

Also, create a custom event definition that *does not contain a time definition* (e.g. take the standard eventdef and remove "d:time"). If the time specification is left in, `acis_process_events` will create a column where every event has a time equal to TSTART; this will cause problems later in the analysis.

```
unix% acis_process_events infile=bgevt2_c7.fits outfile=bgevt2_c7_newgain.fits \
      acaoffffile=NONE stop="none" doevtgrade=no apply_cti=yes apply_tgain=no \
      calculate_pi=yes \
      gainfile=$CALDB/data/chandra/acis/bcf/gain/acisD2000-08-12gainN0005.fits \
      eventdef="{s:ccd_id,s:node_id,i:expno,s:chip,s:tdet,f:det,f:sky,s:phas,l:pha,l:pha_ro,f:ene

unix% dmkeypar bgevt2_c7_newgain.fits GAINFILE echo+
acisD2000-08-12gainN0005.fits
```

`acis_process_events` will print several warnings that may be ignored. Read the Apply an ACIS Gain Map thread for more details on using this tool.

CTI and TGAIN Calibration Files

The calibration files used to apply the CTI and time−dependent gain corrections should also match in the event and background data. In practice, though, it is not currently possible to apply newer CTI and TGAIN correction the background files in the CALDB; they lack some columns required by `acis_process_events`.

The error introduced by this mismatched calibration should not be very significant for the faint, extended

objects for which people use these backgrounds.

Reproject the background data

The aspect solution of the observation is applied to the background dataset – `bgevt2_c7.fits` – by `reproject_events`, which fills in the `SKY` column of the background file. ObsID 1838 has only one asol file, which we assume to be in the current working directory. It is possible to use multiple asol files by using a stack for the aspect parameter; see this FAQ for more information.

The `random` parameter is set to a value other than −1. This tells the tool to apply a random aspect solution to each background event, thus sampling the dither of the observation.

The GTIs used are taken from the match file (i.e. the cleaned event file, `evt2_c7_clean.fits`), which means that the output file will only contain events for those chips contained in the match file.

```
unix% punlearn reproject_events
unix% pset reproject_events infile=bgevt2_c7_newgain.fits
unix% pset reproject_events outfile=bgevt2_c7_reproj.fits
unix% pset reproject_events aspect=pcadf084244404N001_asol1.fits
unix% pset reproject_events match=evt2_c7_clean.fits
unix% pset reproject_events random=0
unix% reproject_events
Input dataset/block specification (bgevt2_c7_newgain.fits):
Output dataset/block specification (bgevt2_c7_reproj.fits):
Match file (evt2_c7_clean.fits):
```

You can check the parameter file that was used with plist reproject_events.

The file `bgevt2_c7_reproj.fits` can now be used to estimate the background of your observation.

Once you have created a background spectrum or image, you may wish to change the value of the `EXPOSURE` keyword to account for the time dependence of the background. You may also wish to use the high energy data (above 10 keV) to see how the particle background normalization differs between your observation and the background.

Caveats

Creating response files

If you are just going to subtract the background spectrum (i.e. not fit it), then you do not need a separate RMF and ARF for the background spectrum. You may simply use dmextract to create the spectrum.

If you do need an RMF file for the background region:

For −120 data with the CTI correction applied, mkacisrmf should be used to make the RMF.

For all other data, use mkrmf. First run acis_fef_lookup to find the FEF file, giving it the *source event file* (*not* the background file) as the `infile` parameter. Find the coordinates of the source in chip coordinates (see the Step−by−Step Guide to Creating ACIS Spectra for Pointlike Sources thread for an example of how to do this). This filename should be used (including any DM filter) for the `infile` parameter of mkrmf.

The same FEF filename and filter should be used for the `feffile` parameter in mkwarf (i.e. instead of "CALDB").

Soft galactic flux and nH

The background datasets are constructed from a set of observations which are spread out across the sky. There is therefore no guarantee that the background at the position of your observation will match that in the background maps. An external check on the background can be made using the ROSAT All–Sky Survey R4–R5 band images. The Cookbook discusses these issues further.

Bad pixel maps

The background files were created using a more conservative list of bad pixels than the early processing of the data, but reprocessed Chandra data should match more closely.

The total area of the detector covered by hot pixels is very small, so different pixel filtering is not a concern, unless a bright hot pixel is left in the user's data. Bad columns are more important and ideally, the same bad pixel and column list should be applied to the data and the background, and also used for making the exposure maps. This excerpt was taken from the bad pixels and columns section of the Cookbook; refer to that document for more details.

Previous CALDB updates

- *CALDB v2.26* (*02 Feb 2004*)*:* New FEF files containing updated information for ACIS S3 were added to the CALDB; there are CTI–corrected and uncorrected versions available.
- *CALDB v2.21* (*14 Feb 2003*)*:* New CTI–corrected ACIS background files for period C (2000–01–29 until 2000–12–01) and period D (2000–12–01 onward) for chips 0, 1, 2, 3, and 6 (I0, I1, I2, I3, and S2). There were also new background data released relevant to S0, S1, S3, and S4, which may now be used either for CTI–corrected datasets or non–CTI–corrected datasets.
- *CALDB v2.17* (*1 Oct 2002*)*:* Updated "period C" (focal–plane temperature of –120 degrees C) background files. The changes to these files include:

    ♦ exclude North Polar Spur pointings for ACIS–S observations.
    ♦ the same observations are used for most chips to enable modeling of the soft Galactic excess.
    ♦ the lightcurves were cleaned using the 2.5–7 kev band for the BI chips
    ♦ corrected regions containing sources using the same algorithm as for the "period D" datasets.
    ♦ used an updated badpix table (adds a few columns for S4).

  Additionally, the headers have been updated to allow CALDB queries (e.g. by `quizcaldb`) to work correctly.
- *CALDB v2.12* (*14 Feb 2002*)*:* The "period D" background datasets, which include some data taken in Very Faint mode.

```
Parameters for /home/username/cxcds_param/dmextract.par


#-----------------------------------------------------------------
#
# DMEXTRACT -- extract columns or counts from an event list
#
#-----------------------------------------------------------------
        infile = evt2_c7_bg.fits[bin time=::259.28] Input event file
        outfile = evt2_c7_bg.lc     Enter output file name
```

```
      (bkg = )                 Background region file or fixed background (counts/pixel/s) subtractic
    (error = gaussian)         Method for error determination(poisson|gaussian|<variance file>)
  (bkgerror = gaussian)        Method for background error determination(poisson|gaussian|<variance :
   (bkgnorm = 1.0)             Background normalization
       (exp = )                Exposure map image file
    (bkgexp = )                Background exposure map image file
   (sys_err = 0)               Fixed systematic error value for SYS_ERR keyword
       (opt = ltc1)            Output file type: pha1
  (defaults = ${ASCDS_CALIB}/cxo.mdb -> /soft/ciao/data/cxo.mdb) Instrument defaults file
      (wmap = )                WMAP filter/binning (e.g. det=8 or default)
   (clobber = no)              OK to overwrite existing output file(s)?
   (verbose = 0)               Verbosity level
      (mode = ql)
```

Parameters for /home/username/cxcds_param/dmgti.par

```
     infile = evt2_c7_bg.lc     Input MTL file
    outfile = evt2_c7_bg.gti    Output GTI file
  userlimit = (count_rate>0.407981 && count_rate<0.587492) User defined limit string
  (mtlfile = none)             Optional output smoothed/filtered MTL file
  (lkupfile = none)            Lookup table defining which MTL columns to check against (NONE|none|<:
    (smooth = yes)             Smooth the input MTL data?
   (clobber = yes)             Clobber output file if it exists?
   (verbose = 0)               Debug level
      (mode = h)
```

Parameters for /home/username/cxcds_param/reproject_events.par

```
     infile = bgevt2_c7_newgain.fits Input dataset/block specification
    outfile = bgevt2_c7_reproj.fits Output dataset/block specification
      match = evt2_c7_clean.fits Match file
    (aspect = pcadf084244404N001_asol1.fits) Aspect file
    (random = 0)                 random seed (0 use time, -1 no randomize)
   (geompar = geom)              Parameter file for Pixlib Geometry files
   (verbose = 0)                 Debug Level(0-5)
   (clobber = no)                Clobber existing file
      (mode = ql)
```

History

05 Jan 2005 updated for CIAO 3.2: updates for CALDB 3.0.0, added Download the Background Files section

12 Jul 2005 `acis_bkgrnd_lookup` updated to version 1.11. A bug has been fixed that prevented the script from storing the name of the selected ACIS background file, or files, in its parameter file.

20 Dec 2005 updated for CIAO 3.3: default value of `dmextract` `error` and `bkgerror` parameters is "gaussian"

16 Feb 2006 added Using the background files with specextract caveat

01 Dec 2006 updated for CIAO 3.4: serial CTI released in CALDB 3.3.0 is not included in the CALDB background files; `dmgti` uses the value of the TIMEPIXR header keyword to adjust start and stop times (users may see a small shift in the time filter when compared to CIAO 3.3 because

of this); `kernel` parameter removed from `dmgti`; CIAO version in errors and warnings; ChIPS version

01 May 2007 updated for CALDB 3.4.0: new ACIS background files, acis_bkgrnd_lookup v1.12; majority of thread rewritten to reflect the new background files.

12 Jul 2007 <u>`acis_process_events` command</u> needs `apply_cti=yes`, to retain the CTI correction on the input file

14 Sep 2007 updated for CALDB 3.4.1: new ACIS background files

URL: http://cxc.harvard.edu/ciao/threads/acisbackground/   Last modified: 14 Sep 2007

Image 1: Source regions overlaid on the image
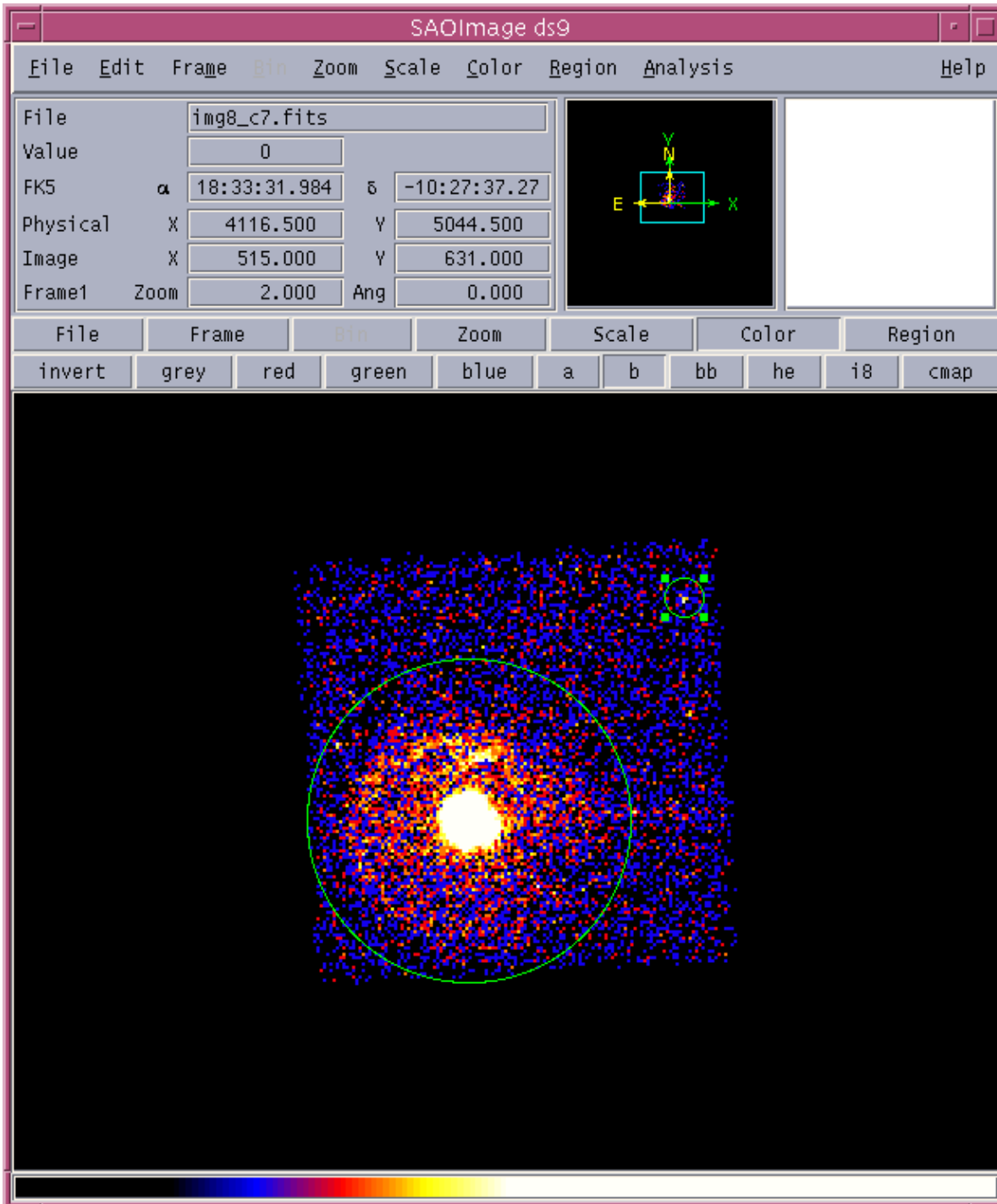
Image 2: Postscript file created by ChIPS

Lightcurve: evt2_c7_bg.lc

Image 3: Background lightcurve for a strong flare
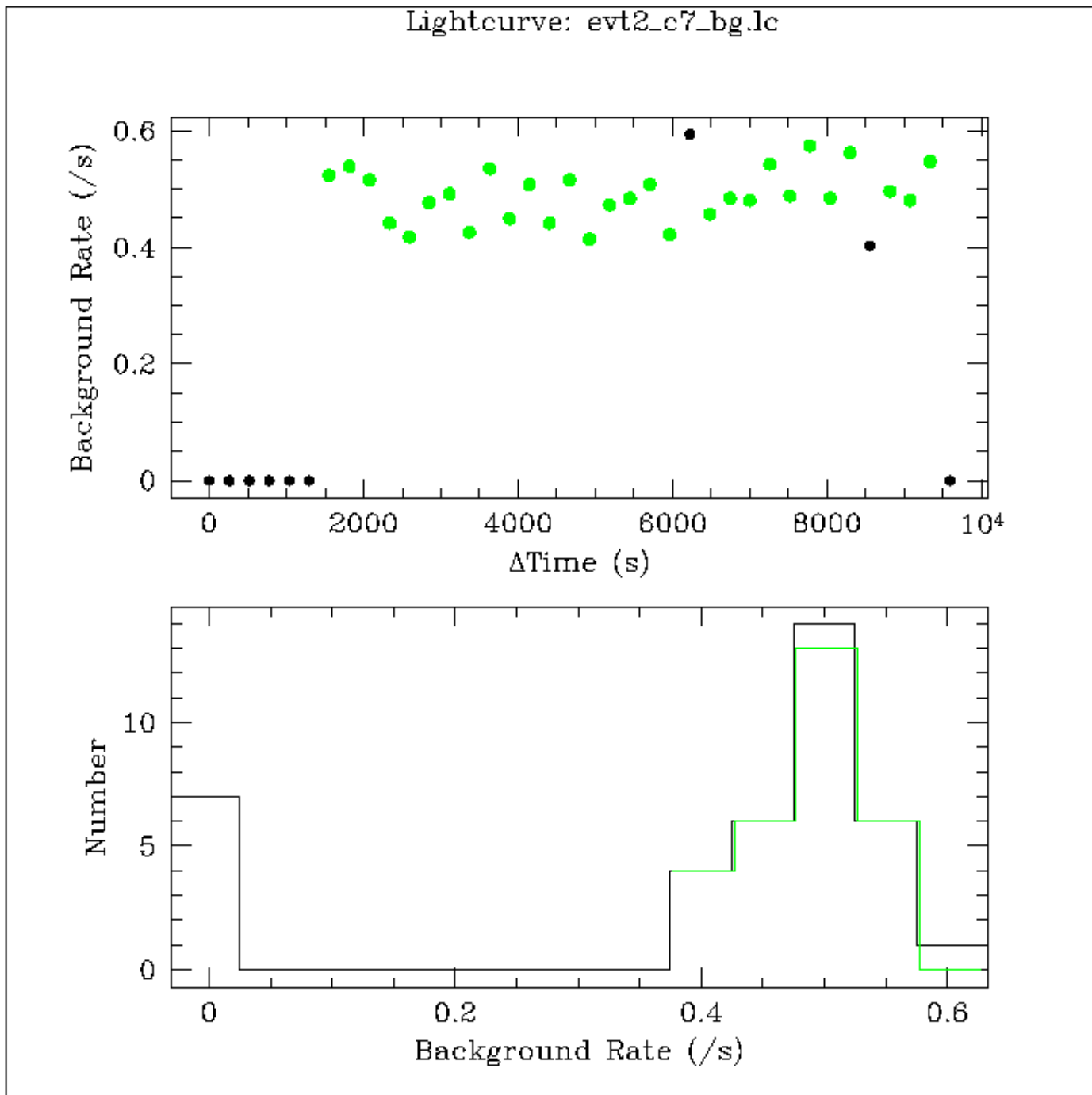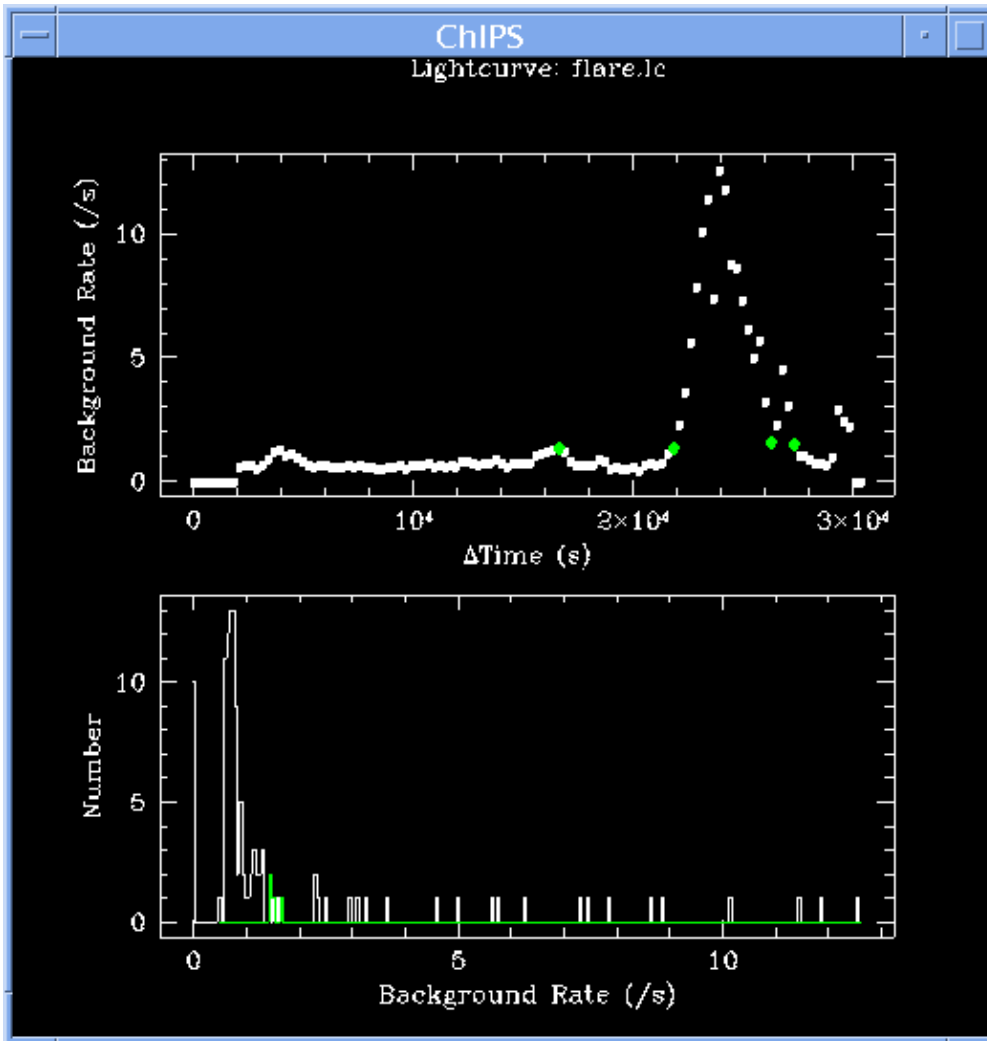

Image 3: Background lightcurve for a strong flare
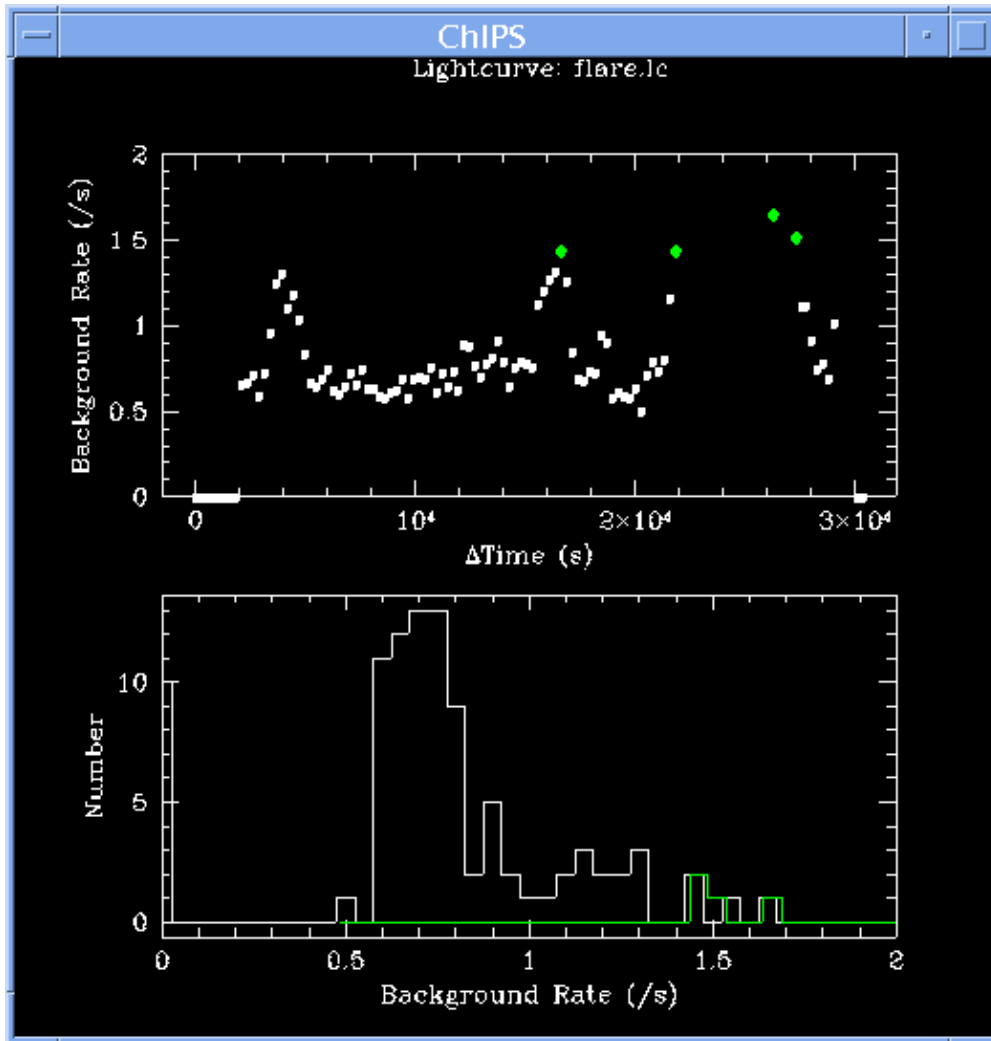
Image 4: Zoomed view of the strong flare

Image 5: lc_clean() run with mean value of count rate

Image 5: lc_clean() run with mean value of count rate