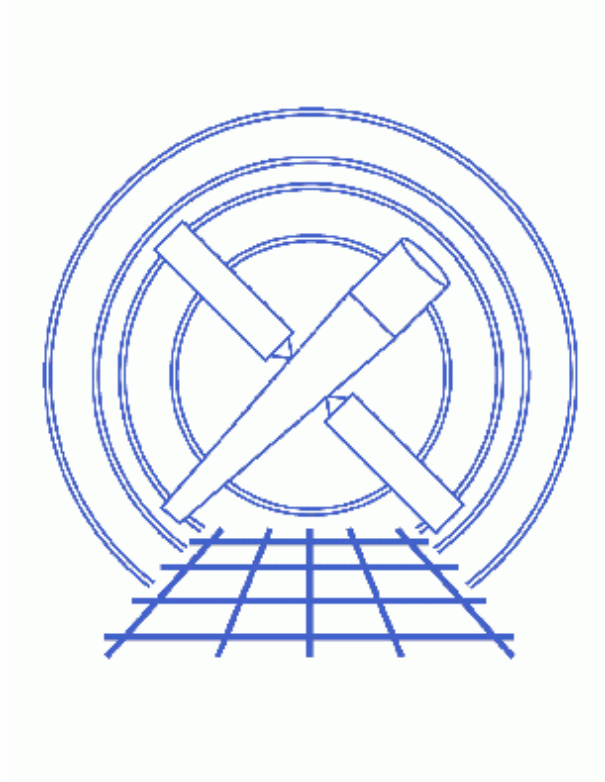


# Filtering Lightcurves



## ***CIAO 3.4 Science Threads***

# Table of Contents

- *Get Started*
- *Remove bright/variable sources from the dataset*
- *Create the lightcurve (dmextract)*
- *Analyze the lightcurve (analyze\_ltcv.sl)*
  1. Using dmgti
  2. Using a DM filter
- *Parameter files:*
  - ◆ dmextract
  - ◆ dmgti
- *History*
- *Images*
  - ◆ ACIS-S3 observation of a field
  - ◆ "Source" regions
  - ◆ Background lightcurve

---

# Filtering Lightcurves

*CIAO 3.4 Science Threads*

## Overview

**Last Update:** 1 Dec 2006 – updated for CIAO 3.4: `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`; ChIPS version

### **Synopsis:**

It may be necessary to filter your data to remove periods of anomolous background levels, such as the flares seen in ACIS observations. The CIAO package includes the dmextract tool to calculate the lightcurve of a dataset (or region within a dataset).

### **Purpose:**

To analyze the output of `dmextract` in order to clean your dataset of periods of anomolous background rates.

### **Read this thread if:**

you are working with an ACIS imaging observation and would like to check for unusual background rates. The algorithm used to detect flares is simple, so it may not be sufficient in all cases. *If you intend to use the ACIS blank-sky background files*, the cleaning described in this thread *is not* conservative enough. Instead, use the alternative method described in the Using the ACIS "Blank-Sky" Background Files thread.

### **Related Links:**

- Analysis Guide: ACIS Data Preparation
- Using the ACIS "Blank-Sky" Background Files thread: an alternative method of flare detection.

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

---

## Get Started

**Sample ObsID used:** 1712 (ACIS-S, 3C 273)

**File types needed:** `evt2`

## Filtering Lightcurves – CIAO 3.4


This thread uses the `analyze_ltcrv.sl` script. The most recent version of `analyze_ltcrv.sl` is v1.5 (25 June 2003):

```
unix% grep Id $ASCDS_CONTRIB/share/slsh/local-packages/analyze_ltcrv.sl
% $Id: analyze_ltcrv.sl,v 1.5 2003/06/25 22:11:23 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 script installed or need to update to a newer version, please refer to the [Scripts page](#).

For this thread we shall restrict attention to the ACIS-S3 chip, and [the 0.3 to 10 keV energy range](#):

```
unix% punlearn dmcop
unix% dmcop "acisf01712N002_evt2.fits[energy=300:10000,ccd_id=7]" evt2_c7.fits
```

An image of the data is shown in [Figure 1](#) .


---

## Remove bright/variable sources from the dataset

To avoid any background variations due to sources in the field we first filter out regions of high/variable emission from the dataset. There are a number of ways to create a list of regions that should be excluded – for instance you may wish to use the output from one of the [source detection programs](#). In this thread we shall use visual estimation, as an example.

Using `ds9`, we chose several regions and saved them in CIAO format as `exclude.reg`, which looks like:

```
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)
```

The chosen regions are shown in [Figure 2](#)  (note that we ignored several obvious point sources in this example).

---

## Create the lightcurve (dmextract)

We can now create a lightcurve of this background region using the CIAO `dmextract` tool. The best way to bin the lightcurve depends on the dataset and your scientific objectives; for this example we use a bin length of 200 seconds:

```
unix% punlearn dmextract
unix% pset dmextract "evt2_c7.fits[exclude sky=region(exclude.reg)][bin time>::200]"
unix% pset dmextract outfile=lc_c7.fits
unix% pset dmextract opt=ltcl
unix% dmextract
Input event file (evt2_c7.fits[exclude sky=region(exclude.reg)][bin time>::200]):
Enter output file name (lc_c7.fits):
```

The contents of the parameter file may be checked with [plist dmextract](#).

---

## Analyze the lightcurve (analyze\_ltcrv.sl)

Here we use the `analyze_ltcrv.sl` script to analyze the lightcurve. It performs an iterative sigma-clipping algorithm, removing those points that fall outside  $\pm 3$  sigma from the mean at each iteration until all data points are within  $\pm 3$  sigma. This algorithm is robust but not perfect; it can easily "overclean" a noisy lightcurve and should not be used blindly. The output is a graph of the lightcurve, showing suggested excluded points in red and accepted points in green, together with a list of the accepted time periods and their lengths.

The script is used within [ChIPS](#):

```
unix% chips

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

chips> () = evalfile("analyze_ltcrv.sl")
chips> analyze_ltcrv("lc_c7.fits")
((time > 77377570.949648) && (time < 77399570.949648)) ; 22.00 ksec
((time > 77404770.949648) && (time < 77406770.949648)) ; 2.00 ksec

chips> exit
```

The script also produces a plot [showing the lightcurve](#) , where the red points indicate the suggested periods to reject.

The output time periods can then be used to filter the event list, either by using [dmgti](#) to create a [GTI](#) file, or directly within a [DM filter expression](#):

### 1. Using dmgti

The time limits can be used by `dmgti` to create a GTI file, as shown below:

```
unix% punlearn dmgti
unix% pset dmgti infile=lc_c7.fits
unix% pset dmgti outfile=lc_c7.gti
unix% pset dmgti userlimit = \
"((time > 77377570.949648) && (time < 77399570.949648))|((time > 77404770.949648) && (time < 77406770.949648))"
unix% dmgti
Input MTL file (lc_c7.fits):
Output GTI file (lc_c7.gti):
User defined limit string ((time > 77377570.949648) && (time < 77399570.949648))|((time > 77404770.949648) && (time < 77406770.949648)):
```

The contents of the parameter file may be checked with [plist dmgti](#).

The output of `dmgti`, here `lc_c7.gti`, can then be used to filter an event list:

```
unix% dmcop "evt2_c7.fits[@lc_c7.gti]" evt2_c7_clean1.fits
```

## Filtering Lightcurves – CIAO 3.4

```
unix% dmlist evt2_c7.fits subspace
-----
Data subspace for block EVENTS: Components: 1 Descriptors: 15
-----

--- Component 1 ---
  1 time                Real8                TABLE GTI7
                                           77379630.9496500045: 77407432.5496758968
  2 ccd_id              Int2                 7:7
...

unix% dmlist evt2_c7_clean1.fits subspace
-----
Data subspace for block EVENTS: Components: 1 Descriptors: 15
-----

--- Component 1 ---
  1 time                Real8                TABLE GTI7
                                           77379630.9496500045: 77399570.9496479928
                                           77404970.9496479928: 77406770.9496479928
  2 ccd_id              Int2                 7:7
...

```

## 2. Using a DM filter

Alternatively, you can use filter the event list directly:

```
unix% dmcopy \
  "evt2_c7.fits[time=77377570.949648:77399570.949648,77404770.949648:77406770.949648]" \
  evt2_c7_clean2.fits

unix% dmlist evt2_c7_clean2.fits subspace
-----
Data subspace for block EVENTS: Components: 1 Descriptors: 15
-----

--- Component 1 ---
  1 time                Real8                TABLE GTI7
                                           77379630.9496500045: 77399570.9496479928
                                           77404770.9496479928: 77406770.9496479928
  2 ccd_id              Int2                 7:7
...

```

The two filtering methods produce slightly different results, which can be seen by looking at the cleaned exposure times:

```
unix% dmkeypar evt2_c7_clean1.fits EXPOSURE echo+
21365.98128797

unix% dmkeypar evt2_c7_clean2.fits EXPOSURE echo+
21662.182507332

```

## Filtering Lightcurves – CIAO 3.4

The difference is due to the way the range endpoints are handled in the two cases. The `dmgti` call explicitly gives open intervals, e.g. `(time > a) && (time < b)`, so the endpoints aren't included. Data Model range filters on real-valued columns include the lower bound but exclude the upper one, so `time=a:b` is interpreted as `(time >= a) && (time < b)`. If you change the `dmgti` expression to include the lower bounds:

```
unix% pset dmgti \  
      userlimit="((time >= 77377570.949648) && (time < 77399570.949648))|((time >= 77404770.949648
```

then the results will be closer.

---

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

```
#-----  
#  
# DMEXTRACT -- extract columns or counts from an event list  
#  
#-----  
      infile = evt2_c7.fits[exclude sky=region(exclude.reg)][bin time>:::200] Input event file  
      outfile = lc_c7.fits          Enter output file name  
      (bkg = )                      Background region file or fixed background (counts/pixel/s) subtra  
      (error = gaussian)           Method for error determination(poisson|gaussian|<variance file>)  
(bkgerror = gaussian)           Method for background error determination(poisson|gaussian|<varian  
(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: phal  
(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 = lc_c7.fits          Input MTL file  
      outfile = lc_c7.gti         Output GTI file  
      userlimit = ((time > 77377570.949648) && (time <  
77399570.949648))|((time > 77404770.949648) && (time <  
77406770.949648)) User defined limit string  
      (mtlfile = none)            Optional output smoothed/filtered MTL file  
(lkupfile = none)              Lookup table defining which MTL columns to check against (NONE|non  
      (smooth = yes)             Smooth the input MTL data?  
(clobber = no)                 Clobber output file if it exists?  
(verbose = 0)                  Debug level  
      (mode = ql)
```

## History

- 14 Dec 2004 updated for CIAO 3.2: path for script
- 08 Dec 2005 updated for CIAO 3.3: default value of `dmextract error` and `bkgerror` parameters is "gaussian"
- 01 Dec 2006 updated for CIAO 3.4: `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`; ChIPS version

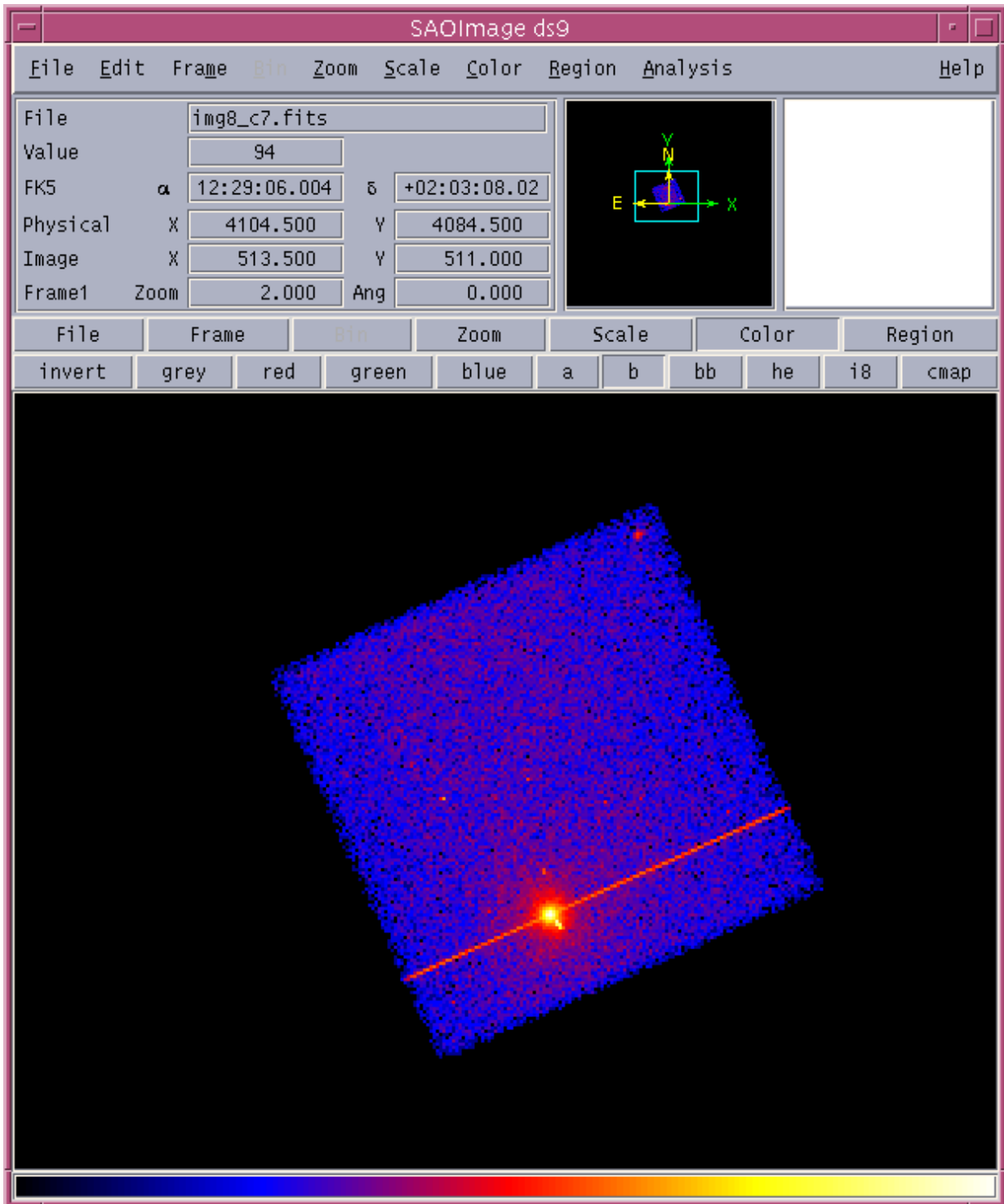
---

URL: [http://cxc.harvard.edu/ciao/threads/filter\\_ltrv/](http://cxc.harvard.edu/ciao/threads/filter_ltrv/)

Last modified: 1 Dec 2006

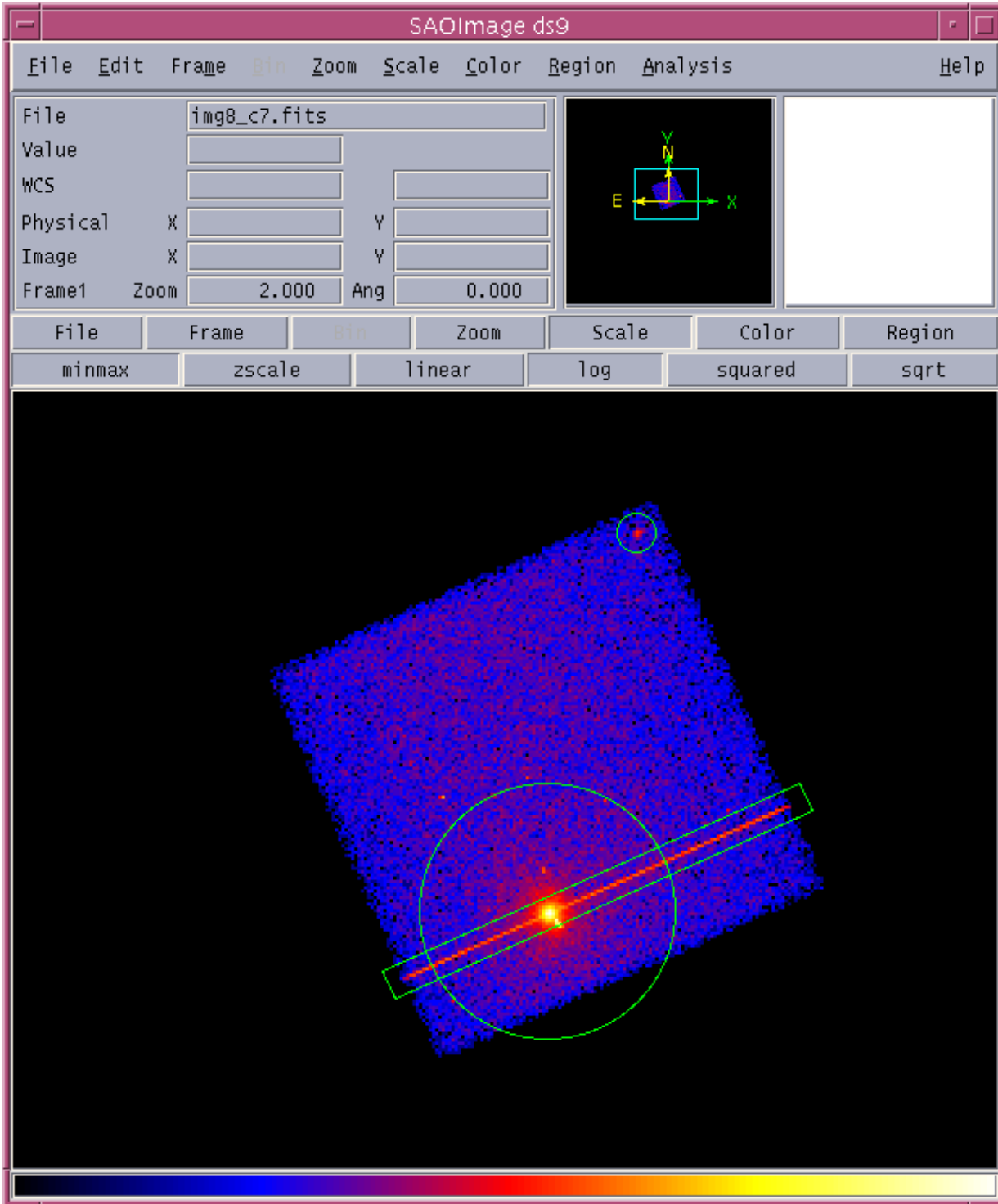


**Image 1: ACIS–S3 observation of a field**



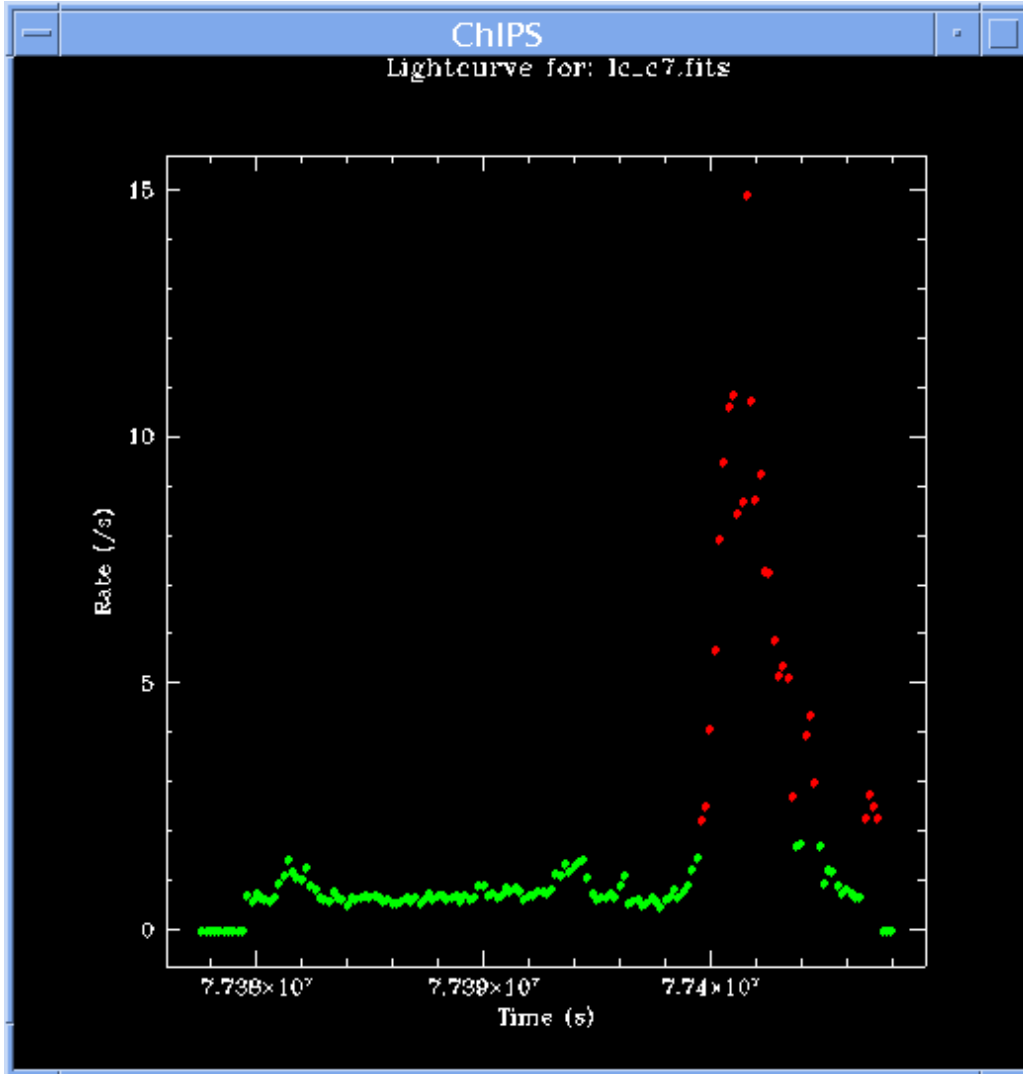
## Image 2: "Source" regions

For this example, the regions were chosen to contain most of the source emission.



### Image 3: Background lightcurve

The red points indicate those times that lie outside the suggested good-time intervals(s) – the background flare here is very strong.



## Filtering Lightcurves – CIAO 3.4