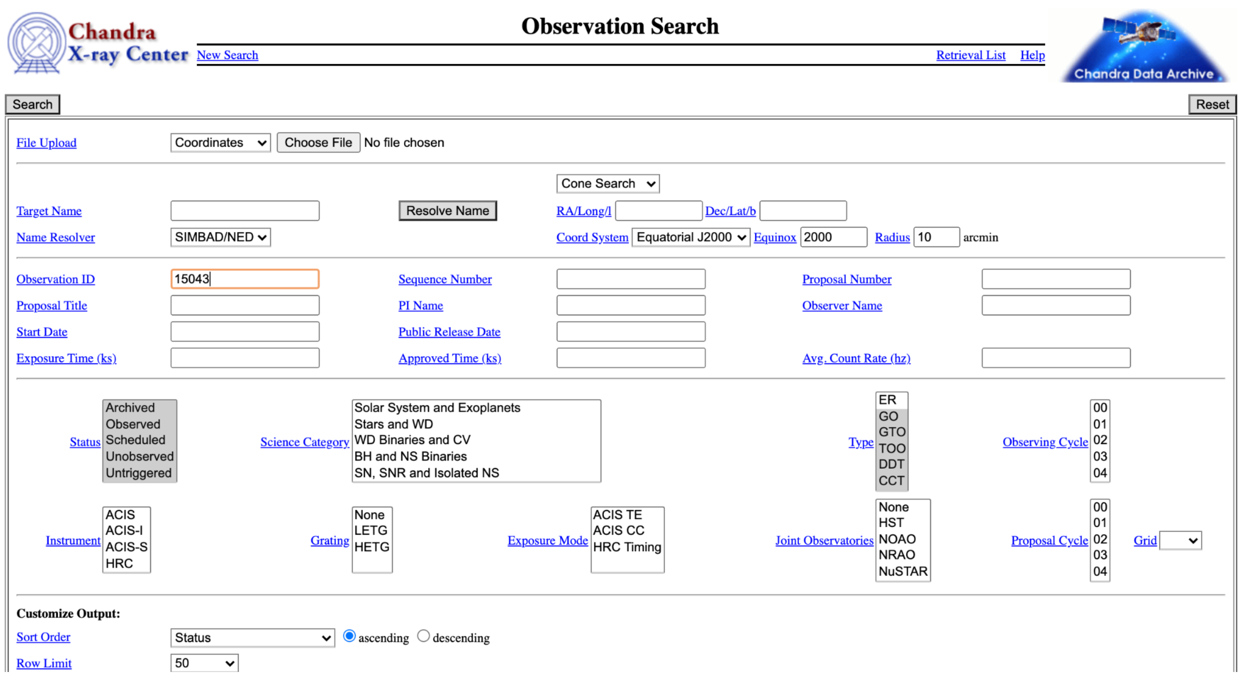
Guide to Analyzing Flares

Let's learn how to analyze Sgr A* flares using CIAO! We will go over how to process raw data downloaded from the Chandra Archive (either using ChaSeR or in the command line) and how to run the Bayesian Blocks code adapted for Sgr A* analysis by Chloe Robeyns and Elisa Jacquet. Before starting this guide make sure you have the latest version of [Anaconda]() and [CIAO]() installed.

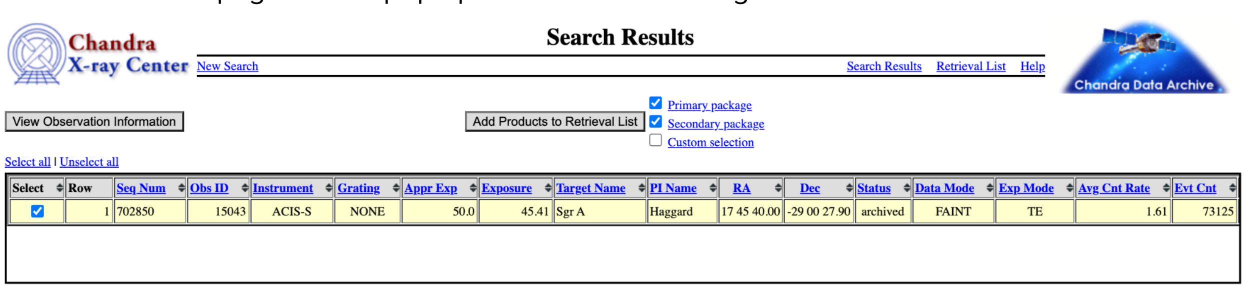
If you have a Windows machine, you will have to install Anaconda for Ubuntu first. You can find it in your Windows Store by searching for Ubuntu and then installing Anaconda: https://www.digitalocean.com/community/tutorials/how-to-install-anaconda-on-ubuntu-18-04-quickstart

1. Downloading data using ChaSeR

Access ChaSeR here: https://cda.harvard.edu/chaser/. Input your ObsID of interest (e.g. 15043) in the Observation ID field and click Search.



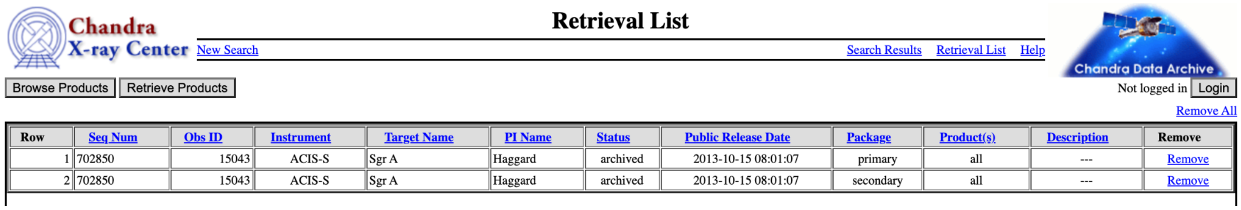
A Search Results page should pop up that looks something like this:



Select	Row	Seq Num	Obs ID	Instrument	Grating	Appr Exp	Exposure	Target Name	PI Name	RA	Dec	Status	Data Mode	Exp Mode	Avg Cnt Rate	Evt Cnt
☑	1	702850	15043	ACIS-S	NONE	50.0	45.41	Sgr A	Haggard	17 45 40.00	-29 00 27.90	archived	FAINT	TE	1.61	73125

Here, we've found one observation of Sgr A* archived under ObsID 15043. If you scroll to the right you should find additional information such as the observation start date (2013-09-14

00:03:46) and the science category (Active Galaxies and Quasars). Make sure the observation you want is selected and that the Primary package and Secondary package are selected. Click “Add Products to Retrieval List”.



Row	Seq Num	Obs ID	Instrument	Target Name	PI Name	Status	Public Release Date	Package	Product(s)	Description	Remove
1	702850	15043	ACIS-S	Sgr A	Haggard	archived	2013-10-15 08:01:07	primary	all	---	Remove
2	702850	15043	ACIS-S	Sgr A	Haggard	archived	2013-10-15 08:01:07	secondary	all	---	Remove

You can go back and search for more observations by going back to “Browse Products” and adding them to the retrieval list so that you download them all at the same time. When you’re ready to download, click “Retrieve Products”. Since it takes a couple minutes to retrieve everything and queue the download, I usually put my email in and wait for a notification. When you get the Chandra Data Notification email, click the link and download the data.

The data will be downloaded in a .tar file so to extract the contents open a terminal window and type `tar xvf file_name` (e.g. `tar xvf 23665`). Now all the files will have a .gz extension so to decompress type `gunzip file_name` (e.g. `gunzip 23665`). To decompress all the files in a directory, navigate to that directory using `cd directory_name` and then type `gunzip *.gz`.

2. Downloading data using the command line

To download data from the command line, start the CIAO software (https://cxc.harvard.edu/ciao/threads/ciao_startup/) and type `download_chandra_obsid` followed by the ObsID you want to download. For example, to download ObsID 23665 I would type `download_chandra_obsid 15043`.

3. Copy data to McGill server

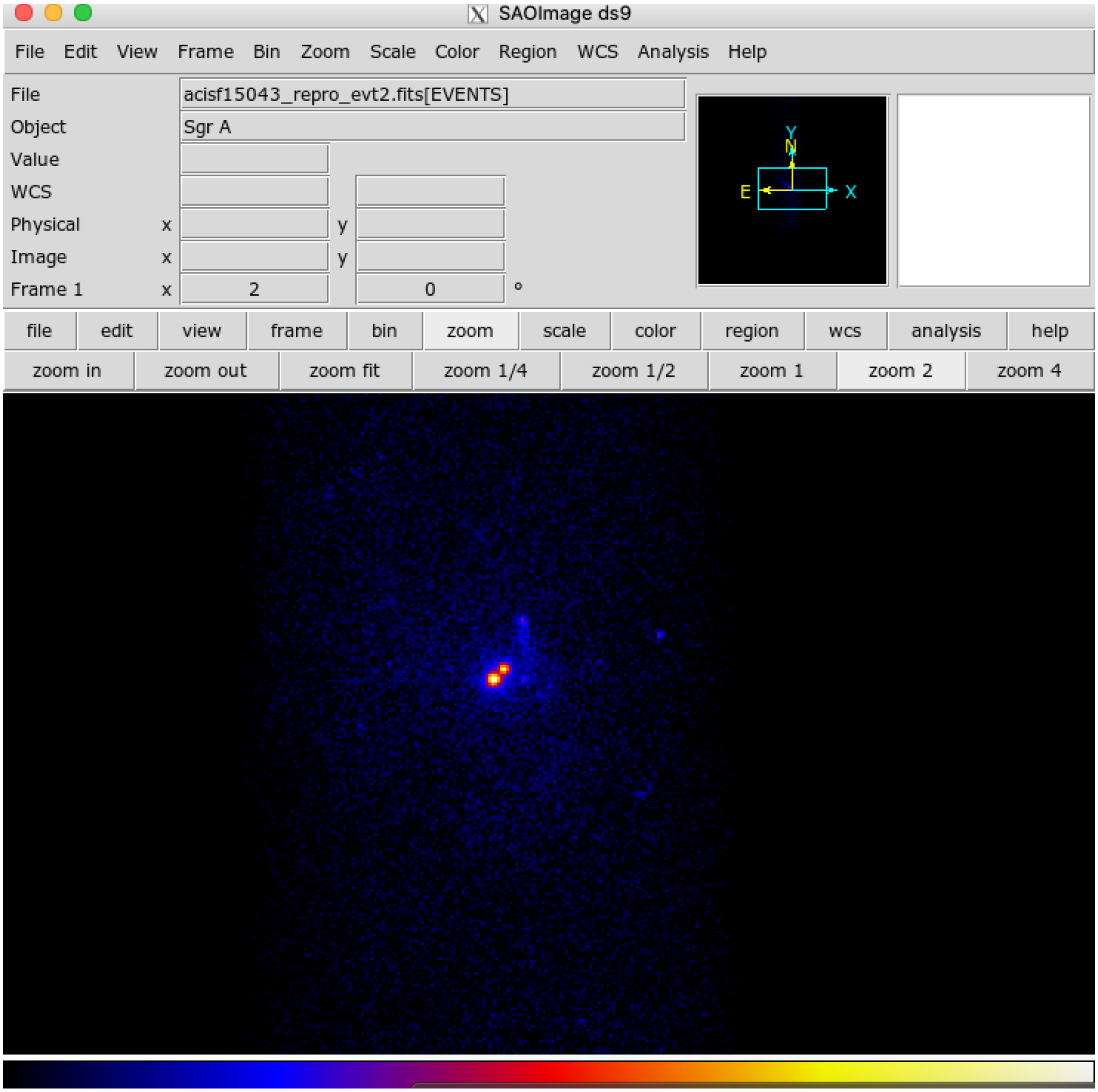
This step is applicable only if you’re using the McGill server. To copy your downloaded data to irulan using the command line type
`scp -r local_directory server_directory`
Note the space between the two directories
(e.g. `scp -r ~/chandra_data/15043 user@physics.mcgill.ca:/home/zark/user/chandra_data`).
You can use `pwd` to print your current working directory.

4. Reprocess the data

To keep everything organized I like to move all of my downloaded files into a folder/directory (I’ll be using these two terms interchangeably) named “chandra_data”. To begin, open a terminal window and start the CIAO software. Use `cd` to change directories to where your data is (e.g. `cd ~/chandra_data/15043)`. Type `ls` to list the contents of the directory. Your directory should contain two folders named “primary” and “secondary”. Stay in this directory and type `chandra_repro` which reprocesses the data. It will create a new folder named “repro”. Navigate to that directory with `cd repro`.

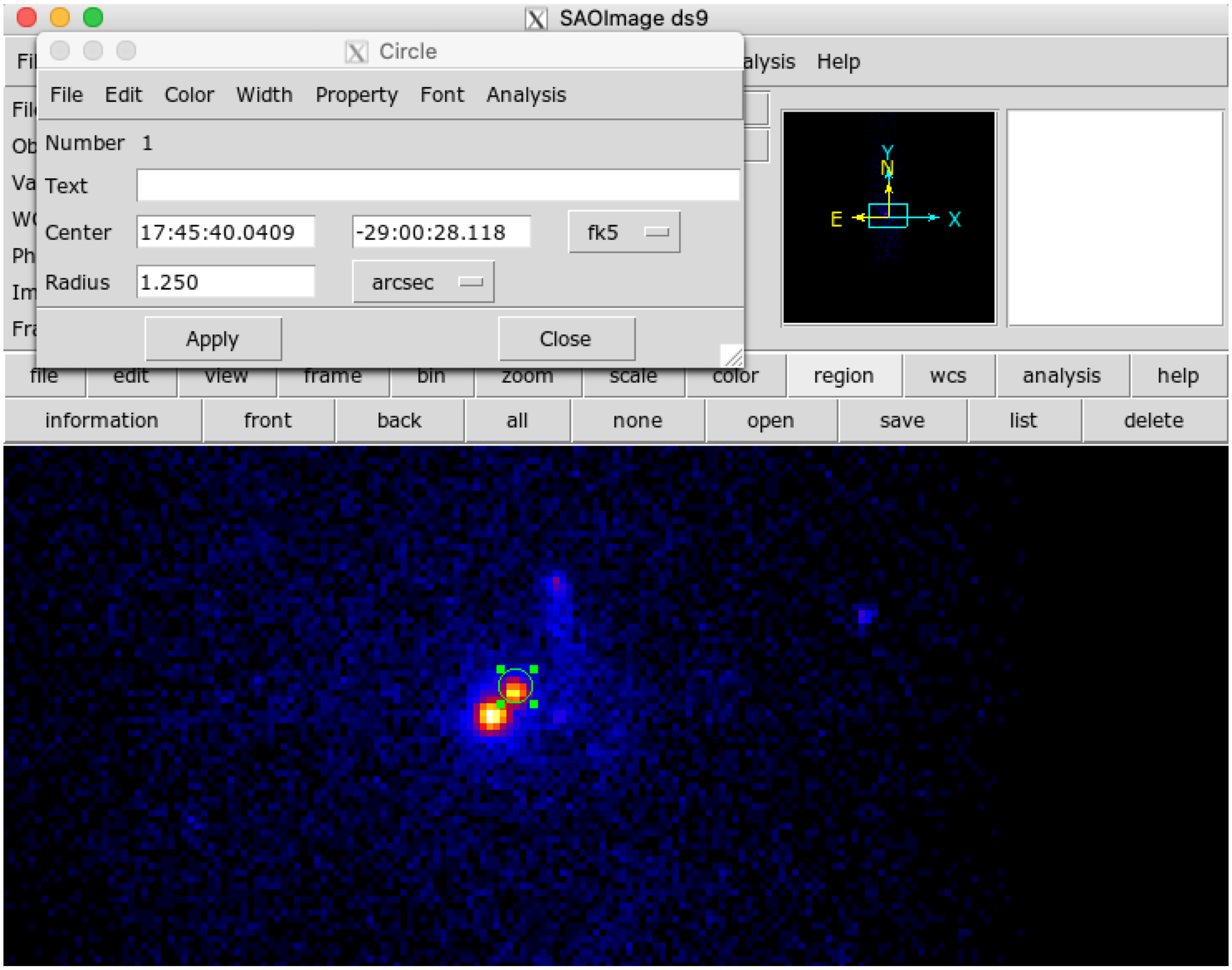
5. Define the regions

We’ll be working with the evt2.fits named something like “acisf15043_repro_evt2.fits”. We’ll use SAOImage ds9 (https://cxc.harvard.edu/ciao/threads/ds9/) to visualize the data.
In the terminal type `ds9 file_name_evt2.fits &` (e.g. `ds9 acisf15043_repro_evt2.fits &`). Or if it doesn’t work you can just type `ds9` in the terminal then File → Open → `~/chandra_data/15043/repro/acisf15043_repro_evt2.fits` and click “OK”. A window should pop up that looks something like this:

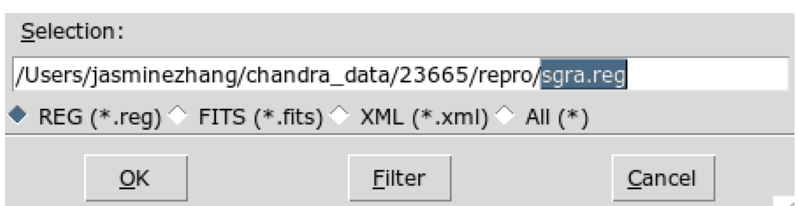


I've changed the colour map with color → map and the scale with scale → log.

Define the source region with edit → region. Left-click and drag on the image in the ds9 display to create a circular region. You can choose the shape and the color of the region with region → shape or region → color. Double click inside the circle to input the centre and radius of the region. In the "Center" field input RA = 17:45:40.0409 and Dec = -29:00:28.118. In the drop down menu beside the "Radius" field select "arcsec" and give the region a radius of 1.25". To define the region as a source or a background, click Property → Source/Background. Here is a source. Click "Apply" to save your changes and then "Close" the pop-up.

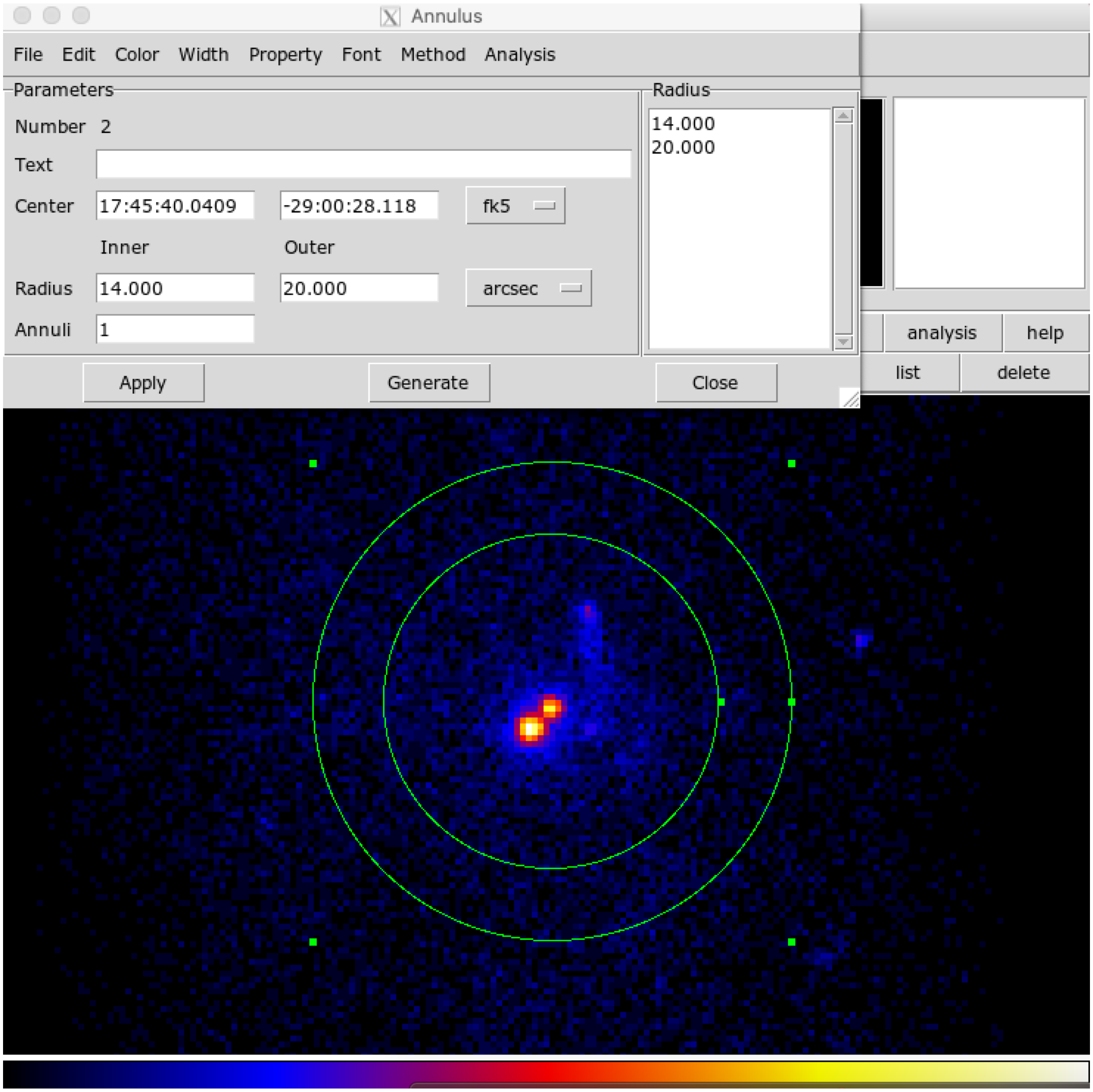


To save, click region → save. Change the file name to sgra.reg. After choosing "OK", set the format to "CIAO" and the coordinate system to "Physical".



Next we'll define a background region. Make sure you delete your Sgr A* region before starting this step. Then in the top menu go to region → shape → annulus.

Left-click and drag on the image in the ds9 display to create an annular region with the same centre as Sgr A*, an inner radius of 14", and an outer radius of 20". Click "Generate" then "Apply". Define the region as a background by doing Property → Background. "Close" the pop-up then region → save as bkg.reg. After choosing "OK", set the format to "CIAO" and the coordinate system to "Physical" like we did for the source region.



6. Determine which chips are being used

In the repro directory, so by typing: `cd repro`, use [dmstat]() to determine the correct chip. In the terminal type:

```
unix% punlearn dmstat
unix% dmstat "acisf15043_repro_evt2.fits[sky=region(sgra.reg)][cols ccd_id]"
```

```
ccd_id
    min:     7           @:     1
    max:     7           @:     1
   mean:     7
  sigma:     0
    sum:     23807
   good:     3401
   null:     0

unix% dmstat "acisf15043_repro_evt2.fits[sky=region(bkg.reg)][cols ccd_id]"
ccd_id
    min:     7           @:     1
    max:     7           @:     1
   mean:     7
  sigma:     0
    sum:     22176
   good:     3168
   null:     0
```

Make sure to replace the example filename (highlighted in orange) with the name of the file you're working with. The regions I am working with are all located on chip 7. Take note of which regions you're working with.

7. Create a background-subtracted lightcurve

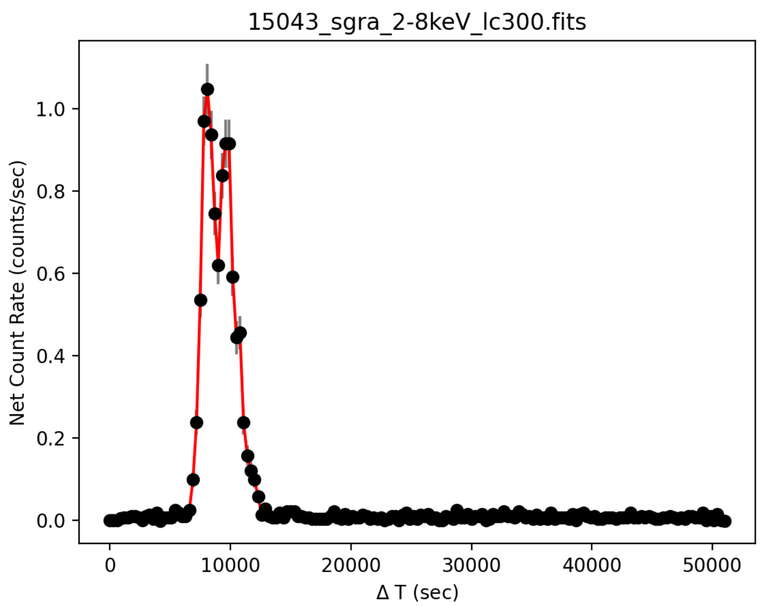
Make sure you're in the `repro` directory. Type `ls` to verify that you have two files named "sgra.reg" for the source and "bkg.reg" for the background. We'll use [`dmextract`]() to create a 2-8 keV lightcurve binned in time intervals of 300 s. Check that you're using the correct filename and ccd_id for your ObsID. Change the ObsID in the outfile to the one you're working with.
Be careful to write `pset dmextract` and `infile` on the same line!

```
unix% punlearn dmextract
unix% pset dmextract
infile="acisf15043_repro_evt2.fits[ccd_id=7,energy=2000:8000,sky=region(sgra.reg)][
bin time=::300]"
unix% pset dmextract outfile="15043_sgra_2-8keV_lc300.fits"
unix% pset dmextract bkg="acisf15043_repro_evt2.fits[ccd_id=7,sky=region(bkg.reg)]"
unix% pset dmextract opt="ltc1"
unix% dmextract
Input event file
(acisf15043_repro_evt2.fits[ccd_id=7,energy=2000:8000,sky=region(sgra.reg)][bin
time=::300]):
Enter output file name (15043_sgra_2-8keV_lc300.fits):
```

The lightcurve may be plotted with matplotlib. This isn't necessary but it may be useful to check if the lightcurve contains a flare.

```
unix% python

>>> from pycrates import read_file
>>> import matplotlib.pylab as plt
>>> tab = read_file("15043_sgra_2-8keV_lc300.fits")
>>> dt = tab.get_column("dt").values
>>> rate = tab.get_column("net_rate").values
>>> erate = tab.get_column("err_rate").values
>>> plt.errorbar(dt, rate, yerr=erate, marker="o", color="red",
mfc="black",mec="black", ecolor="grey")
>>> plt.xlabel("$\Delta$ T (sec)")
>>> plt.ylabel("Net Count Rate (counts/sec)")
>>> plt.title("15043_sgra_2-8keV_lc300.fits")
>>> plt.show()
```



To exit python from the terminal type `exit()` or `quit()`.

8. Create a filtered events file

We'll use [dmcopy] to copy a "virtual file" filtered by our source region and our energy band to a physical disk file.

```
unix% punlearn dmcopy
unix% pset dmcopy
infile="acisf15043_repro_evt2.fits[EVENTS][sky=region(sgra.reg)][energy=2000:8000]"
unix% pset dmcopy outfile="15043_sgra_2-8keV_evt.fits"
unix% pset dmcopy option="all"
unix% dmcopy
```

```
Input dataset/block specification
(acisf15043_repro_evt2.fits[EVENTS][sky=region(sgra.reg)][energy=2000:8000]):
Output dataset name (15043_sgra_2-8keV_evt.fits):
```

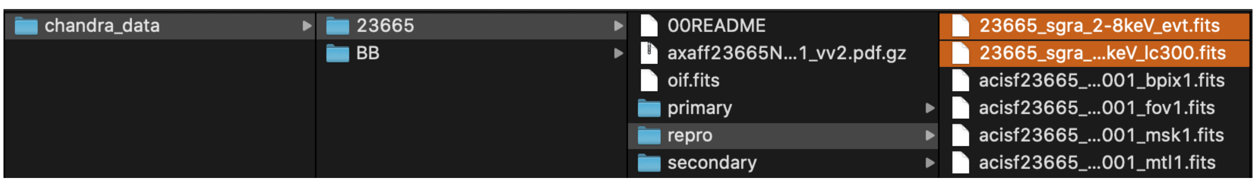
9. Save data from McGill server to laptop

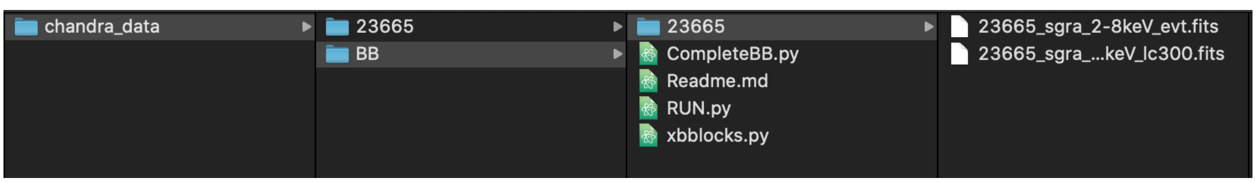
This step is applicable only if you're using the McGill server. Open the terminal but don't log into choco or irulan yet. Type `scp -r server_directory local_directory` (e.g. `scp -r user@physics.mcgill.ca:/home/zark/user/chandra_data ~/chandra_data/23665`).

10. Getting ready to use Bayesian Blocks

Download the CompleteBB.py, xbblocks.py, and RUN.py scripts from this Github in a text format but still written with a py extension:
https://github.com/chloerobeyns/SgrAFlareDatabase/tree/master/Bayesian%20Blocks%20Code

Move all three scripts into your "chandra_data" folder from Step 4. I like to keep them in a folder named "BB" to keep them separate from the downloaded files. In the "BB" folder create a new folder named by your ObsID. Copy the lightcurve file (e.g. "15043_sgra_2-8keV_lc300.fits") created in Step 7 and the events file (e.g. "15043_sgra_2-8keV_evt.fits") created in Step 8 into this folder which can be found in the "repro" folder.





From the command line:

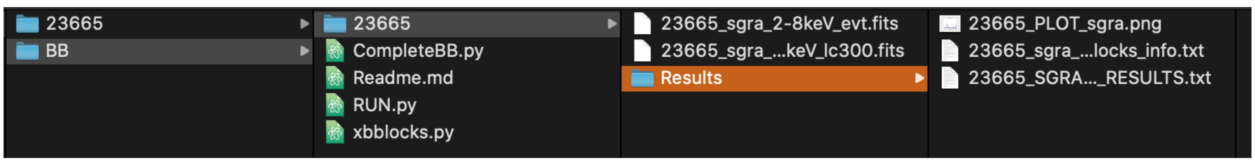
```
unix% cd ~/chandra_data
unix% mkdir BB
unix% mv ~/downloads/CompleteBB.py ~/downloads/RUN.py ~/downloads/xbblocks.py
~/chandra_data/BB
unix% cd BB
unix% mkdir 15043
unix% cp 15043/repro/15043_sgra_2-8keV_lc300.fits BB/15043
unix% cp 15043/repro/15043_sgra_2-8keV_evt.fits BB/15043
```

Note that in the third line above I am moving my python scripts from my "downloads" folder to my

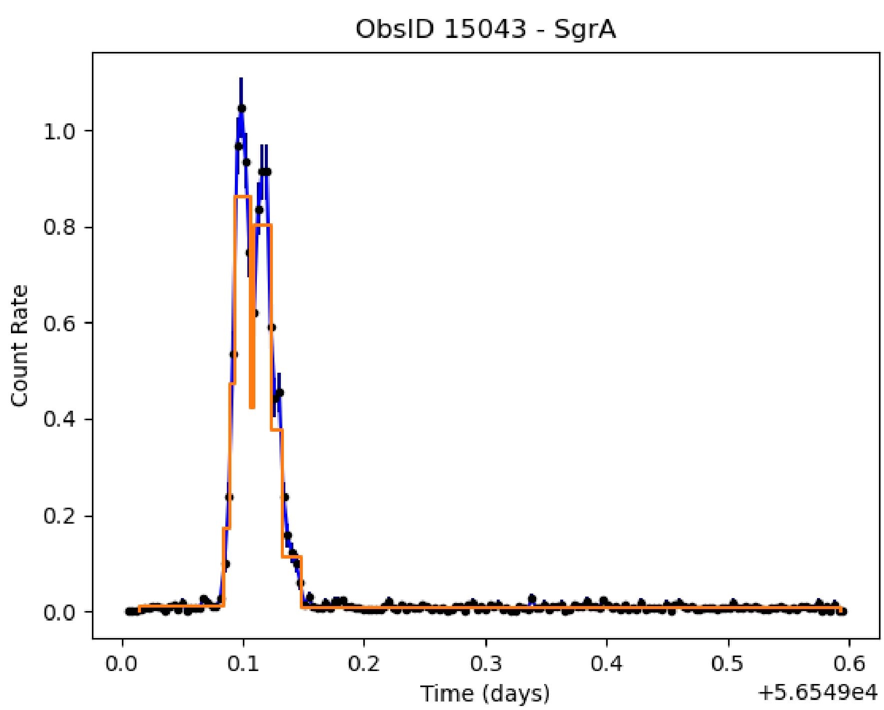
“BB” folder. Double check that you’re putting in the correct location for where your python files are!

11. Run Bayesian Blocks

In the terminal navigate to the “BB” directory (e.g. `cd ~/chandra_data/BB`) and then type `python RUN.py ObsID False` (e.g. `python RUN.py 15043 False`). If you want to run the Bayesian Blocks code on two sets of data run the command `python RUN.py first_ObsID False second_ObsID False`. This can be done for any number of ObsIds given that they each have a folder containing the proper lightcurve and events files. The False is for a non magnetar region, explanations are on the Readme.md file in the previous Github https://github.com/chloerobeyns/SgrAFlareDatabase/tree/master/Bayesian%20Blocks%20Code



After the code runs it will create a "Results" folder in the directory as the lightcurve and events files. In this "Results" directory there will be a plot of the lightcurve plotted with Bayesian blocks, a .txt file with information about each block, and a .txt file containing information for the flare table.



15043_sgra_bayesianBlocks_info.txt

```
# p0 = 0.05
# timesys = TT
# tstarts = 56549.01465
#tstops  = 56549.59415
# n = 3104
56549.01465 56549.08481   61 0.070165 869.379 113.064
56549.08481 56549.08985   75 0.00504331 14871.2 3755.61
56549.08985 56549.09380  161 0.00394328 40829 8420.48
56549.09380 56549.10687  973 0.0130779 74400.3 6253.68
56549.10687 56549.10959   99 0.00271818 36421.4 15391.2
56549.10959 56549.12302  932 0.0134276 69409.5 3517.01
56549.12302 56549.13328  334 0.0102576 32561.2 2653.51
56549.13328 56549.14878  151 0.0155 9741.93 1825.2
56549.14878 56549.59415  318 0.445373 714.008 47.2956
```

15043_SGRA_TABLE_RESULTS.txt

```
------------------------------------------------------------------------------------

BASIC INFORMATION

------------------------------------------------------------------------------------

Obs_ID: 15043

Obs Date: 2013-09-14T00:04:52

Telescope: CHANDRA

Instrument: ACIS

Exposure (ks): 50.069314654111864

Quiescent Count Rate (10^-3 ct/s): 8.264 +/- 0.463

------------------------------------------------------------------------------------

FLARE NUMBER  1

------------------------------------------------------------------------------------

Start Time:  56549.08481 (MJD)

End Time:  56549.14878 (MJD)

Duration:  5526.823968 (s)

Count Rate (mean):  0.49 +/- 0.0094 (ct/s)

Count Rate (max):  1.05 +/- 0.06 (ct/s)

Energy:  206.1 +/- 4.02 10^37 ergs

Luminosity:  37.29 +/- 0.73 10^34 erg/s

Flux:  49.44 +/- 0.97 10^-12 erg/s/cm2

Fluence:  2725.0 ct
```