

ASTR469: Project #2, Analyzing X-ray Data

Due Friday April 14 by class time

We will have three in-class sessions to work on this assignment (with the spring holiday in between).

You will spend time outside of class writing up the project and reviewing relevant literature and class materials.

Concepts practiced: *High-energy astronomy; Emission mechanisms; Luminosities, fluxes, and brightness; Photon properties, angles; Image noise; Unix; Python; L^AT_EX.*

Description:

For this project we will be working with X-ray data from online archives. We will be doing some common X-ray data processing, including forming light curves (the X-ray intensity as a function of time), spectra, and images. **However, you will choose what object to work on and what measurement you want to make.** Remember that in the X-ray regime, we count each and every photon in part because there are so few. Each photon detected is called an “event.” You will be analyzing the events recorded by the Chandra X-ray telescope.

Project tasks:

1. **Consider choices for your target.** All of the below list are X-ray emitters and will work well for this project.

Object	Type	Chandra Obs ID
Hydra A	Galaxy cluster	4969
Perseus Cluster	Galaxy cluster	11714
30 Dorados	HII region	7263
Kepler’s SNR	Supernova Remnant	6715
Tycho’s SNR	Supernova Remnant	10095
Vela Pulsar	Pulsar	12074
M82	Galaxy	361

2. **Consider what measurement you might want to make.** It is your job this time to decide what to do with these data. Here are some ideas; note that each of these will require some background review (class materials or published works) and independent/critical thinking on your part!

- Compute the total X-ray luminosity of the object (subtract off a background by measuring counts in a source-free region the same size as your target).
- Compute (and plot) the variation of some light-measurement property as a function of distance from the center of the object.
- Does the source exhibit time-dependent variability (would you expect it to)?
- Examine the differences between the source and background emission. Compare the brightness of two objects on the list.
- In our high-energy lecture we conceptually discussed a number of emission mech-

anisms (Thermal, Synchrotron, Bremsstrahlung, Compton, Inverse compton) and indicated what conditions were required to create those emission mechanisms. Considering the type of object and your image, discuss what emission mechanism might be at play in your object to create the X-rays you observed (is there a dense gas/object? plasma? diffuse magnetic fields? relativistic electrons?).

- Estimate the X-ray spectral index by creating images of “soft” (low energy) and “hard” (high energy) emission, then examining the spatial distribution of their ratio. Officially, the spectral index α is defined as the slope of the spectrum in log space, where the measured flux S is proportional to frequency ν as $S \propto \nu^\alpha$.

3. **Write up your project.** As before, your writeup should include:

- An introduction describing the object and the telescope
- Basic observation details (the target coordinates, the observing date, the integration time, the telescope and instrument, energy range, etc.). Include a light-curve and an image of the source (creation of both are described below).
- A description of your methodology.
- Sections for results/conclusions/summary, as appropriate.

Grades for this project will be weighted more heavily toward the rigor and accuracy of your methods (i.e. did you state clearly what measurement you wanted to make, did you describe conceptually how one could make that measurement, and did you actually make that measurement correctly?), and your discussion of the measurement (interpret the results to the best of your ability).

Excluding figures, a suitable length for the words in this project in `\documentclass{article}` format would be 1–2 pages.

Below are a few procedures that will help you with data access and processing. Try all bullets below (the final one is optional).

- **Download the data.** After you pick an object(s), navigate to <http://cda.harvard.edu/chaser/>

In the ObsID field, enter the ObsID. Hit *Search* in top left. Select your source, then select *custom selection* in the top middle, then *Add products to retrieval list*. In the next screen, select only the first *evt* box, then again *Add products to retrieval list*.

Download this file, then move it to your working directory. In Linux/Mac you can unpack it with:

```
>> tar -xvf filename
```

Your windows machine may have `tar` installed - if you right click on the file you may have options. Un-tarring the file will create a directory tree. The file you want will be the `acisf(...).evt2.fits` file in the directory:

```
./obsid/primary/
```

If it has a `.gz` extension (“gunzip”), you need to unpack that file with:

```
>> gunzip filename (again, Windows may include something already).
```

- **Read the data in python.** You can actually import `fits` files directly into python. Load your data into python with:

```
>> from astropy.io import fits
>> my_events = fits.open('filename')
```

The above unpacks the whole fits file into a big, messy python data structure (which we've named `my_events`). We can extract just the X-ray event data with:

```
my_data = my_events[1].data
```

The data you just extracted (`my_data`) has a number of columns in it. You can access these columns using the column names, for instance:

```
>> print my_data['time']
```

While there are many columns, for you the important ones are:

- `time`: when a photon event was recorded (units: seconds)
- `ccd_id`: the CCD chip ID number from 1 to 7
- `x`: the detector x location (units: pixels)
- `y`: the detector y location (units: pixels)
- `energy`: the energy of the photon (units: eV)

- **Clean up the data.** X-ray emission from most sources is of constant intensity in time. Deviations from this constant intensity may indicate a cosmic ray hitting the detector or interactions with the Solar wind. These are generally not what we want, but we can determine if such events occur by examining the number of counts as a function of time.

First, determine the average count rate. You can plot the data binned as a function of time. I've left you to practice some python skills on your own in this project (think back to the lectures and to tricks you learned last project)! But I will give you a few hints you might not recognize:

HINT: histograms are necessary, and you need to have the output of these histograms. Histograms are part of the `matplotlib.pyplot` library (and many others, too).

HINT: bin the counts heavily for best results, so that histogram looks blocky. This is an option when plotting, i.e.

```
>> plt.hist(my_data['time'], bins = 100)
```

Second, determine the time intervals that deviate from the average count rate. We will want to exclude these from the data in the future. Here is one example line that includes how to combine the good data. You will have to create your own. If all time intervals are good, you can skip this step.

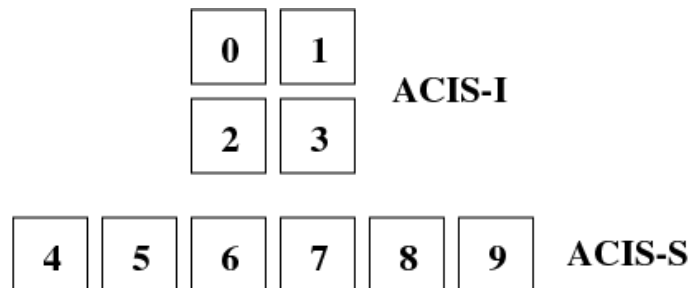
```
>> good = (my_data['time'] > 5.42169e7) & (my_data['time'] < 5.4219e7)
```

- **Look at the data.** Plot the x and y events against each other FROM THE GOOD TIME INTERVALS ONLY. Note that you're plotting actual events here, not a proper image; you in effect are making a 2D scatter plot. Try to locate your object (it may

be tough if there are a lot of counts). It should be near the center. You can probably make out the outline of the many CCD detectors of Chandra.

One thing to help make your data pretty in case you want to try it: the `matplotlib` plot takes a third argument that is quoted with the first part being the color and the second being the data point to use. For example, `'r.'` will plot red dots, `'b+'` will be blue pluses, etc.

You can modify the “good” array to include only the CCDs of interest. Here is how the CCDs are laid out (showing the CCD IDs):



Note python will flip the plots top to bottom!

- **Make an actual image.** Remember, these are individual photons events, and not pixels, so we have to make the pixels ourselves by binning. You can make big or small pixels, depending on what you want.

Instead of a 2D scatter plot, a 2D histogram allows you to see the object better. You will plot the data by first making a 2D histogram and then plotting it with `matplotlib`'s `plt.imshow()`. Numpy has a 2D histogram function called “`histogram2d`”. This creates three variables: the histogram data, the xbin values, and the ybin values:

```
>> my_hist, xbins, ybins = np.histogram2d(...)
```

Have a look at the usage of this function online and work out what inputs it needs. Specify the number of pixels with the argument `bins = #`. If you previously filtered out bad events, you can include only the “good” events with, for example:

```
>> my_data['x'][good]
```

The scaling may be off. Try to fix it with the `vmin` and `vmax` arguments to `plt.imshow()`. Once again, `imshow` may flip the image. Revert back to normal with argument “`origin=lower.`” You can also get real astronomical coordinates for the axes.

- **Make an energy spectrum.** You can make a histogram of the energy distribution of the chip with your data in it. Can you compare it to a CCD with no source in it?