The report of the project for the Partial fulfillment of the course PHY312: Numerical Methods and Programming at IISER Bhopal

Astronomical Data Reduction and Analysis (Photometry) using Python Programming Language

By

Parth Sunilkumar Nayak
Roll no. 15118

Bikram Keshari Sahu
Roll no. 15051

Course Instructor

Dr. Nirmal Ganguli
Assistant Professor

Indian Institute of Science Education and Research (IISER) Bhopal

Date of submission
April 8, 2018

Abstract

The aim of the project was to learn how the Python programming language can be useful for cleaning and analysis of the astronomical (CCD) data. We have used Python version 2.* for our project throughout. It is very commonly believed that Python is a very user-friendly and easy-to-use yet very efficient programming language. Here we explore the use of Python for handling the CCD data in the FITS format. The data are in the form of two-dimentional pixel-arrays having the photon counts at each pixel constituting an image of a field of sky. Python's module PyFITS was used to access the FITS data in the computer-accessible format. First of all, the data are processed for various noise cancellation. This is known as "Cleaning of the data" in astronomical terminology. Once cleaned, the data are further analysed to extract the magnitudes (brightnesses) of certain specific sources (stars, galaxies, planets, etc.). This is known as "Photometry". The magnitudes can be plotted against time to see the variations over time. These plots are known as the "Light curves". The PSF (Point Spread Function) of a star in a CCD image can be approximated by a 2-D gaussian. This corresponds to the average atmospheric seeing over time. The data were collected from the IUCAA-Girawali 2m-telescope, Pune, India. We present the light curve of this data using our very preliminary methods implemented in Python. The peaks were found and the corresponding profiles were fitted with gaussians thus obtaining the FWHM and the sky signal. (The light curve here is not well-standardized, however it can be used to infer the variability status of the source.)

Contents

1 Introduction

In astronomy, the observations are done from the telescopes and the data are recorded using the CCD (Charge-Coupled Device). It works on the principle of photoelectric effect. It collects the light on its individual pixels over the exposure-time set by the observer and converts it into a two-dimentional array of counts (#photons per pixel). A FITS file constitutes such a 2-D array and some other relevant informations e.g Time of Exposure, Date and Time of observation, observatory coordinates, Right Ascension (RA) and Declination (Dec)[1] of the observed source in the form of its headers. FITS is an image format used for astronomical images, where FITS stands for Flexible Image Transfer System. It was developed by a group of astronomers in USA and Europe in late 1970s to serve the easy interchange of data between observatories and was brought under the auspices of the International Astronomical Union in 1982. More information on this can be found at: *https://fits.gsfc.nasa.gov/standard30/fits_standard30.pdf*
The following Section 2 describes the various Noise-cancellation procedures for cleaning the data. Section 3 is devoted to the techniques of photometry and the use of Python for those. We present our results in Section 4. Further applications/modifications are pointed out in Section 5 of Discussions.

2 Noise-cancellation (Cleaning)

The CCD is a semiconductor device. The data (science files/frames) recorded with them have different kinds of internal as well as external noises. These must be removed before analysing the data for our purpose. A sample of Raw science frame is shown in Figure 1. The description of these noises and the removal of them using Python is descibed below.

2.1 Bias-subtraction

The CCD is an electronic device which needs an external bias (or offset) voltage to operate. Due to this internal biasing of the CCD, there is a small amount of signal that is generated. There may be signal due to the (thermal) dark current as well. This signal is unwanted and must be subtracted from the science frames so as to get good, noise-free images. For that purpose, the "bias frames" are observed. For bias-observation, the shutter of the CCD camera is closed so that no light can fall on the CCD (essentially that is what enables us to observe the instrumental noise of the camera). Then the CCD is integrated for the same duration as the exposure time for the science frames and read to obtain a bias frame. Multiple bias frames are observed so as to be statistically sure that the master bias frame (which is obtained by median-combining all the good bias frames) can be undoubtedly used for noise reduction. The master bias frame is then subtracted from all the science frames. This whole process is called "Bias-subtraction". The reason for combining them as median (and not as mean) can be understood by a simple logic: if in one frame out of 6 (let), counts at certain pixels is *unusuallly* high/low (these are called hot/cold pixels), the mean of those pixels shifts towards the higher/lower value,

[1]RA and Dec are the astronomical coordinates of any celestial object. They are also commonly referred to as the Equatorial coordinates. For information visit: *http://www.physics.csbsju.edu/astro/sky/sky.11.html*

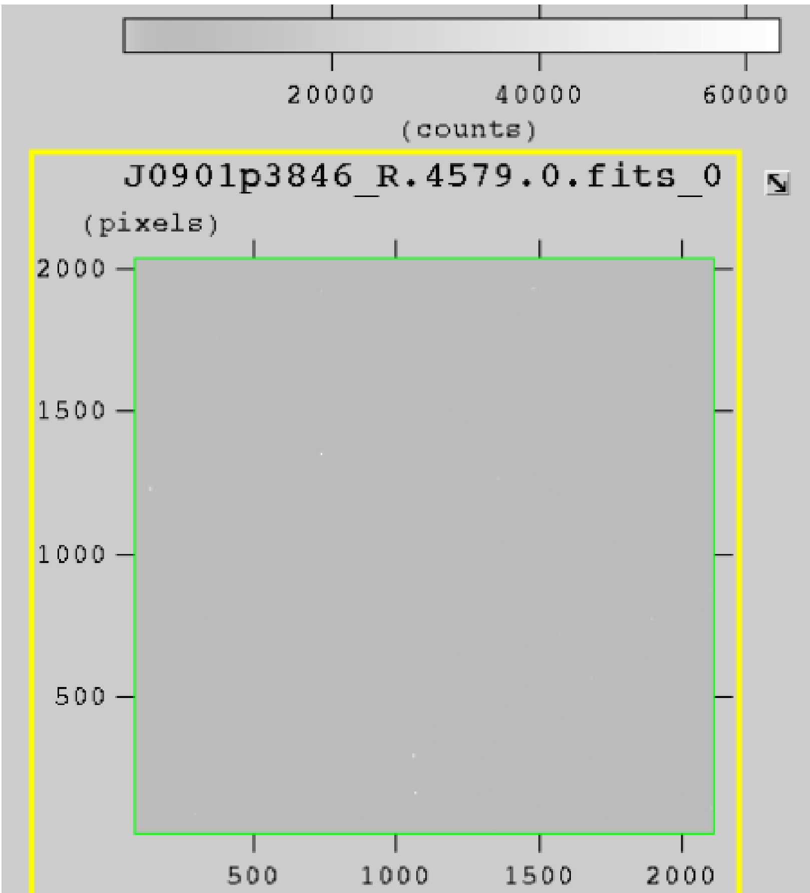


Figure 1: A raw FITS image on logarithmic scale of the source J0901+3846 previewed in NASA's fv (fits viewer). The stars are not readily visible because of certain noises.

which is essentially inadequate for bias-subtraction, however this does not occur in median combination.
The master bias frame was obtaines using our code "**mbias.py**". A module named "**statistics.py**" was created by us for all the required statistical operations on our data (the bias-subtraction was done at the time of final cleaning)[2]. The codes can be accessed here:
https://drive.google.com/drive/folders/1kYRXOKXen9uQvMYBFKUxpoQ0YgDUZGfP?usp=sharing
(with obtained permission outside of IISER Bhopal).

2.2 Flat-fielding

The collection of photons by a pixel and yielding of electrons (and hence, counts) is a random process on the quantum scale, i.e for each pixel it may slightly vary. So, a neighbouring pixel of the one yielding n electrons upon collection of m photons, may not yield the same n electrons when collects m photons. The meaning of this is that there is no uniformity of counts of the pixels. Actually this process depends upon the quantum-efficiencies of the individual pixels. If this is the case, then the raw images collected cannot give us the desired information nicely. Or else to say, this is a kind of

[2]A README file can be found along with the codes which contains all the steps to perform the data-cleaning and photometry with our package.

multiplicative noise in the frame. To get rid of this noise, the flat frames are observed. A flat frame is a frame which has the uniform amount of light fell on each pixel. The observed flat frame, as expected, does not contain a uniform count over each pixel. This way, we have a measure of the non-uniformity of the pixel-gains.
As is the case for biases, multiple flat frames are observed at the time of twilights (which is considered as the best time to observe flats), again for the same duration as the exposure-time for the science frames, and then they are median-combined to get a master flat frame. Now, the master flat is normalised (devided by the mean of the pixel-counts) so that the count oscillates around 1. The science frames (after bias-subtraction) are then divided by the normalised master flat frame for the noise-reduction. This way we make sure that the pixel gain is made uniform over the whole CCD for the flat-fielded science frames. This complete process is called "flat-fielding".
The normalized master flat-field was obtained using our code "**nmflat.py**".

Cleaning also includes removing profiles that are created by the cosmic-rays, which are not gaussian because of their instantaneity. Rather they give localised delta profiles. These are unnecessary and should be removed. They can be visually distinguished from the stars' profiles and hence we did not implement any special algorithm for that purpose. One example of cosmic ray profile is shown in Figure 2 (a). The 3-D plots were made using Python's module "**mpl_toolkits.mplot3d**".

As a whole, a clean science frame can be obtained as:

$$S_{clean} = \left(\frac{S_{raw} - B_{master}}{(F - B_{master})_{normalised,\, master}} \right)$$

where,
S is the Science-frame,
B is the Bias-frame,
F is the flat-frame,
and the subscripts have their literal meanings.

A sample of clean science frame is shown in Figure 3. The cleaning was done using our code "**clean.py**"

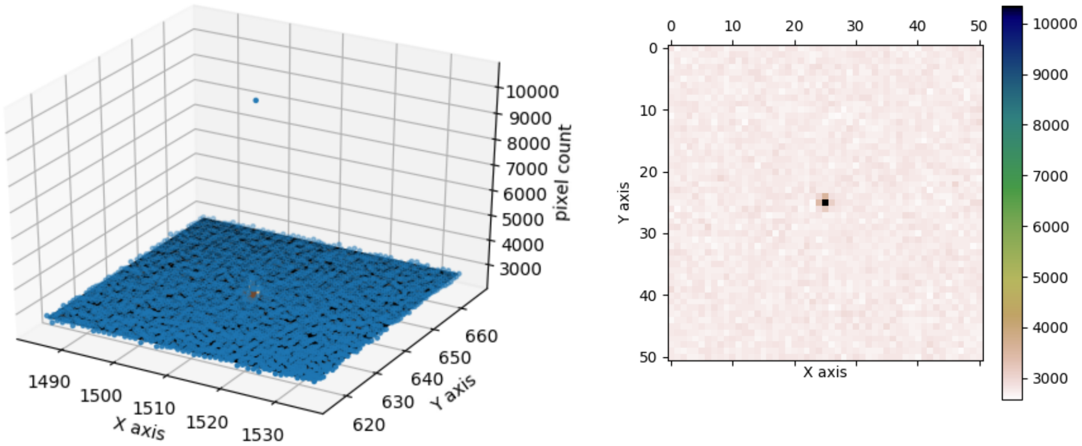
3 Photometry

Photometry is the science of measuring flux of a desired source over a wavelength-band. Flux of a source is the amount of power per unit area from the source that is recieved by the observer (CGS unit: erg s^{-1} cm^{-2}). The apparent magnitude is expressed as,

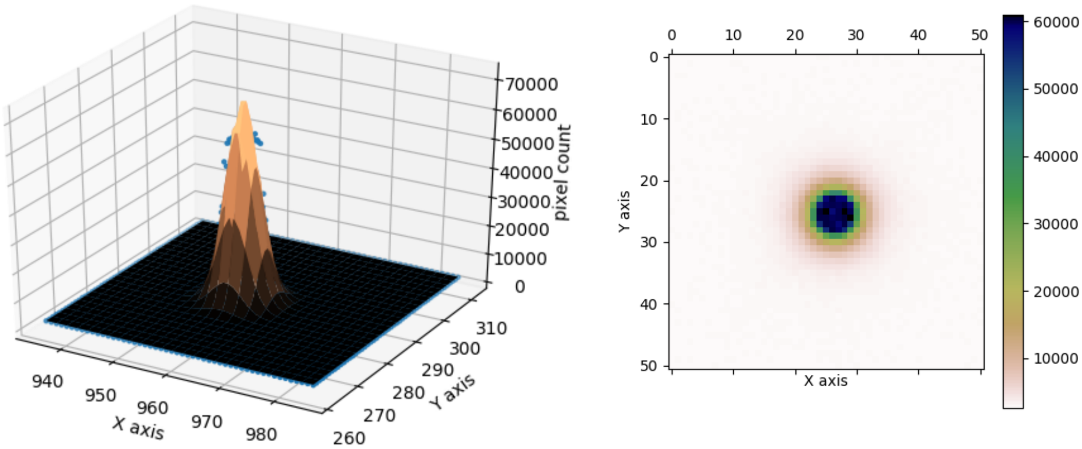
$$m_{apparent} = m_{app} = -2.5 \log_{10}(f/f_{ref})$$

where,
f =flux from the source and,
f_{ref} =flux from a standard reference source.

The star Vega in the constellation of Lyra is chosen as the standard reference source, and



(a) One of the cosmic ray delta profiles in our data



(b) One of the star's gaussian profiles in our data

Figure 2: Different kinds of profiles observed in a CCD image. Here we can visually distinguish between a cosmic ray and a star's profile.

so its apparent magnitude (hereinafter magnitude) is 0. The instrument which we use for observations and data collection cannot give us the magnitude directly, but what it does is, it gives us the information as to how many photons hit the CCD per second, which can be used as a proxy to calculate the amount of power that is recieved by the CCD. The information that a CCD gives is the instrumental magnitude, and can be derived as,

$$m_{instrumental} = m_I = -2.5\log_{10}(\#photons\ per\ second\ from\ the\ source)$$

Then, by measuring the instrumental magnitudes of some standard (known m_{app}) star, the apparent magnitude of the source can be extracted (This process is called standardization

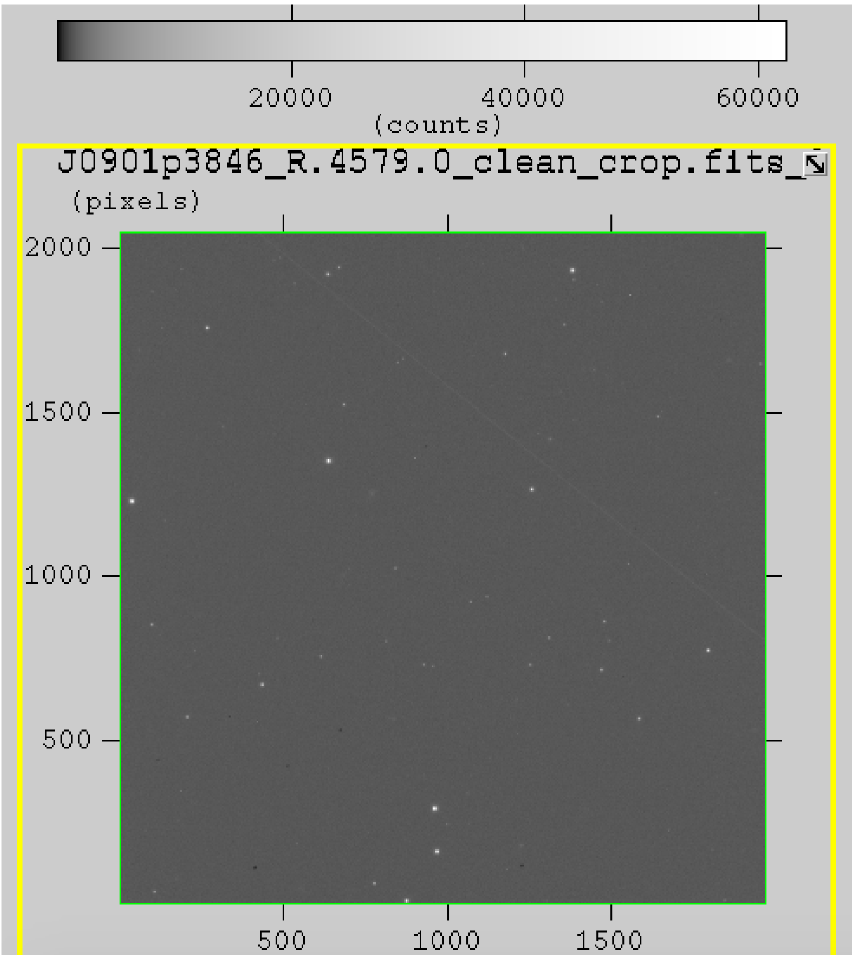


Figure 3: A cleaned FITS image on logarithmic scale of the source J0901+3846 previewed in NASA's fv (fits viewer). After noise-cancellation, now after cleaning, the stars are visible.

of the magnitude data) as follows:

$$m_I = -2.5\log_{10}(C)$$
$$m_I^* = -2.5\log_{10}(C^*)$$
$$\Rightarrow m_I - m_I^* = -2.5\log_{10}\left(\frac{C}{C^*}\right)$$

and, $f = C \times (energy\ unit)$,
hence,

$$\frac{f}{f^*} = \frac{C}{C^*}$$
$$\Rightarrow m - m^* = -2.5\log_{10}\left(\frac{f}{f^*}\right) = -2.5\log_{10}\left(\frac{C}{C^*}\right) = m_I - m_I^*$$
$$\Rightarrow m = m_I + (m^* - m_I^*)$$

where,
m stands for apparent magnitude,
C stands for count,

f stands for flux,
the subscript I is for instrumental quantities and * is for the standard source.

However, in our case, standardization was not done because we just needed to know the variations in the flux over time, which instrumental magnitude can also give. Moreover, we have used the technique called "Differential Photometry" in which we analyse the differential magnitudes of three source, therefore the standardization is not necessary.
There is another important concept of SNR or S/N (Signal to Noise Ratio) while doing photometry. It is the ratio of the desired source signal to the noise such as photon counting statistical noise (Poisson noise). A high S/N is desirable for good photometric results of the data. Even the sky which appears dark is not really dark, but has a little signal-count that is known as sky-background noise. A detailed discussion is addressed in the Aperture Photometry subsection below.

PSF (Point Spread Function): The PSF is the shape of the CCD image of a point source of light. (Real stars are not precisely points of light- they must have some finite angular size. However, except for one or two very nearby, very large stars, the angular size of every star in the sky is far smaller than the diffraction limit of our optical telescopes, so we can treat stars as unresolved points). For all research telescopes, the dominant determinant of the PSF is smearing caused by the passage of starlight through the Earths turbulent atmosphere. This smearing is called "seeing". Assuming good optics, proper focusing and tracking, the PSF should be circularly symmetric. The shape of a real PSF set by seeing is a bit complicated, but can be approximated by a 2-D Gaussian. The angular size of the PSF can be characterized in several ways. One common measure is the full width at half maximum (FWHM) which is the diameter where the flux falls to half its central value.

3.1 Aperture Photometry

To extract the instrumental magnitude of the source from a clean science-frame, a technique called Aperture Photometry is used. It is ideal for non-croweded fields but fails for heavily crowded field such as clusters, supernova remnants, etc. For crowded fields, another photometric techinque called PSF Photometry is used, but in our case the fields were not crowded, so the aperture photometry technique was implemented. As mentioned above, there is a so-called sky-background signal which is undesired. It is also recorded along with the source(star)-signal. So, essentially the star-signal observed directly is (star+sky) signal. Sky-signal may also vary by the locality, i.e it may be slightly different close to the star and going farther it may drop a bit. Therefore, to properly measure the signal from the star, we somehow need a measure of the local sky-signal in the vicinity of the star. In aperture photometry, different concentric circles around the star's Gaussian centre are drawn, which have different aperture sizes but are technically in a same aperture. It can be shown that within 4×FWHM radius aperture, nearly all (100%) of the starlight is covered (Figure 4).

Other than the aperture of radius 4×FWHM (the star-aperture), we use two more (outer) apertures (9×FWHM to 12×FWHM), which encloses the sky annulus, and are useful in estimating the sky-signal which can then be subtracted from the star-signal to get an

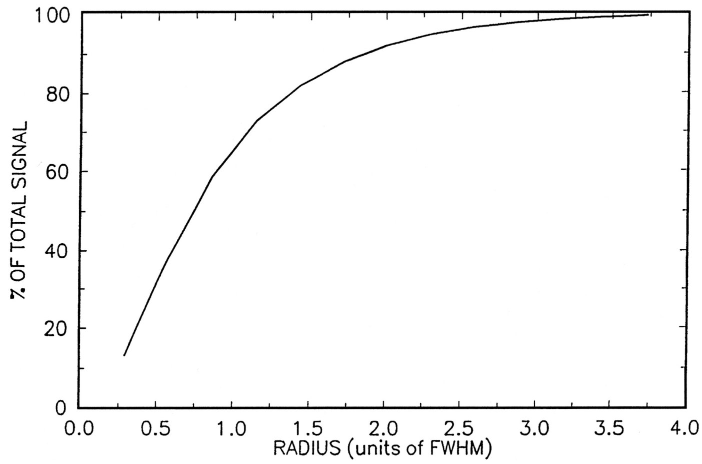


Figure 4: For any reasonable PSF approximation, the figure above shows the run of the total encircled signal with radius of the PSF in FWHM units. It can be seen that within 4×FWHM, all (100%) of the star-signal (star+sky) is included. (Adopted from: Optical Photometry by Sergio Ilovaisky (OHP), 5th NEON School, 23 July-5 August 2006).

absolute value of flux of that star ($star\text{-}signal = (star + sky) - sky$).
Figure 5 shows our plots of data only within the star-aperture (left) and the sky annulus (right).

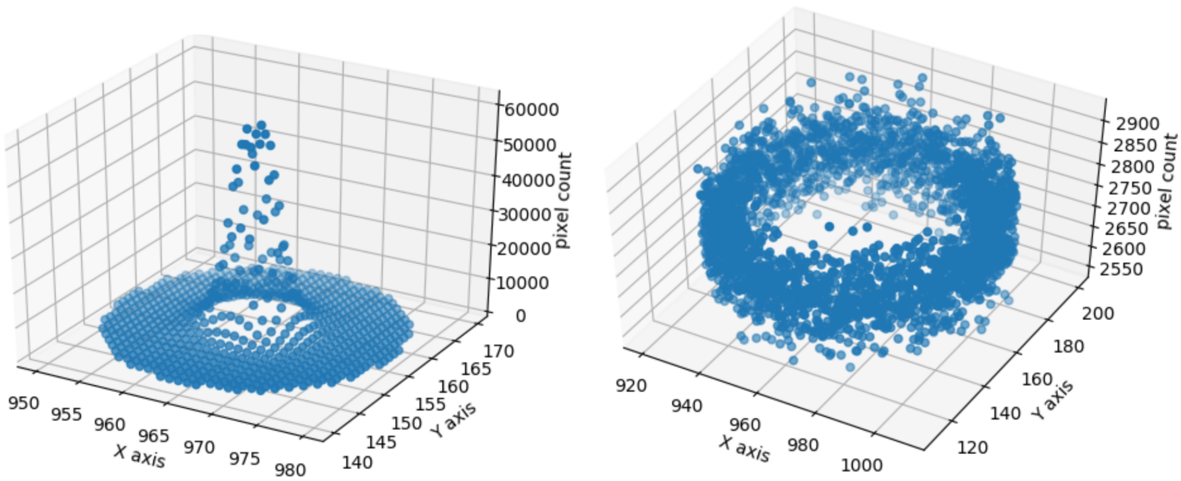


Figure 5: The data within the measurement apertures from one of our clean frames (left: star-aperture, right:sky-annulus)

Let's call the total counts in the measurement aperture N_{ap}, the area of the measurement aperture (in pixels) A_{ap}, the sky-signal per pixel S_{sky} and the exposure time of the frame (in seconds) t_{exp}, then the instrumental magnitude of the star can be obtained as:

$$\boxed{m_I = -2.5 \log_{10} \left(\frac{N_{ap} - A_{ap} S_{sky}}{t_{exp}} \right)} \tag{1}$$

where, the sky-signal is mostly the mode-combination of the set of pixel values measured within the sky annulus.

3.2 Photometry using Python

We have used Python 2.* for the photometry on our data. We first analysed all the peaks in our data above the given threshold using our module "**Analysis.py**". For gaussian fitting over the peaks we used our module "**gaussfit.py**" and, the codes "**3d_profiles.py**" and "**3dplot_target_objects.py**". Thus we got the FWHM of the gaussians. The delta profiles were discarded and only the gaussian profiles were considered. The source RA and Dec were extracted from the headers of the raw files and updated into the headers of the clean files with the code "**header_update.py**". Using the RA and Dec, we looked up the source into the SDSS (Sloan Digital Sky Survey) Data Release 14 Finding Chart tool (*http://skyserver.sdss.org/dr14/en/tools/chart/chartinfo.aspx*) (See Figure 6.) It was matched with our data visually in *fv* and was taken as Source-1 (S1) for differentail photometry. Other two stars (S2 and S3) were picked by selecting two good gaussian profiles from our plots. The aim of differential photometry is to see the variations of differential magnitude of S2 and S3 over time and, minimize it by changing S2 and S3. However, due to time constraints, we could not follow this iterative procedure, so we just selected two stars at once and performed photometry. The coordinates of these three sources were written in the file "**coordinates.dat**" for the use of our photometric codes that follow.

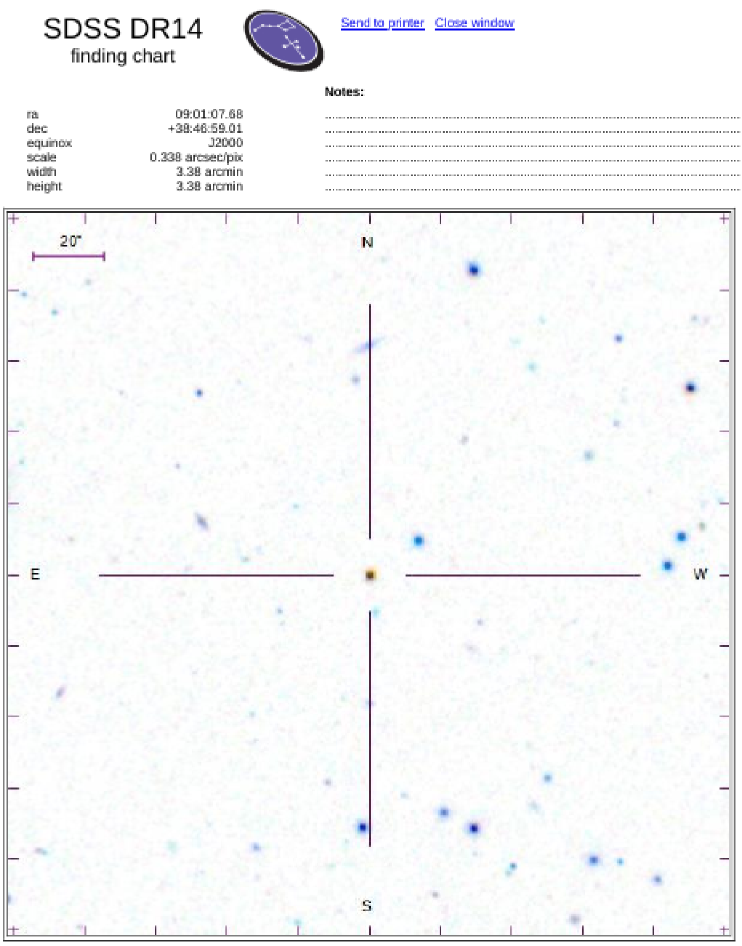


Figure 6: SDSS DR 14 Finding chart of the source J0901+3846 (RA: 09h 01m 07.6s, Dec: 38° 46' 59")

For aperture photometry, we used our modules "**photometry.py**" and "**sky.py**". As

described in subsection 3.1, the star's data was collected within an aperture of radius 4×FWHM with the help of the method "*photo*" of "photometry.py" and the sky value was obtained within the annulus of radii 9×FWHM to 12×FWHM with the help of the method "*base*" of "sky.py".
For differential photometry, we used the code "**light_curve.py**" which gets the coordinates of the desired sources from the file "coordinates.dat" and performs aperture photometry on them, extracts the instrumental magnitudes according to equation (1), derives the differential instrumental magnitudes and writes them along with the date-time of observation into a file "**light_curve_data.txt**".
For plotting the light curve of the data, we used our code "**plot_light_curve.py**". It reads the file "light_curve_data.txt". It collects the data of differential instrumental magnitudes and Date and Universal Time (UT) of observation and plots the data on appropriate X and Y axes.

4 Results

The light curve of our data is shown in Figure 7. The variations can be seen in all the three differential instrumental magnitudes (DMs). As noted earlier in subsection 3.2, we could not optimize the choice of S2 and S3 to get the minimal variations in the DM.

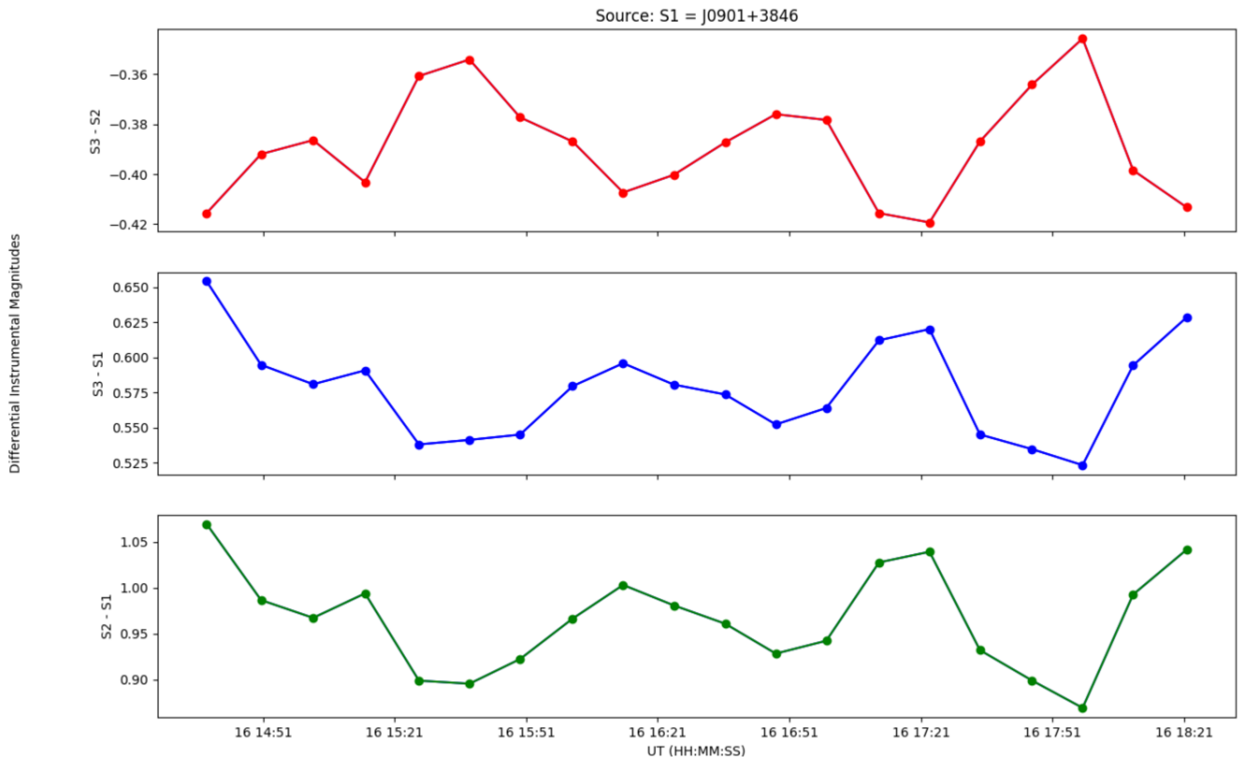


Figure 7: The light curve of our data as created by our codes. S1 corresponds to the source J0901+3846, and the pixel coordinates of S2 and S3 are as listed in "coordinates.dat" file (2nd and 3rd entries).

5 Discussions

The project that we worked on explores the preliminary techniques of astronomical data reduction using Python. There are many advanced techniques available but require a great deal of human efforts and association. We hereby make a few comments on how this basic package can be exploited for further/better analysis of data.

• The cosmic-ray removal technique (manual) used by us is not very efficient. Sometimes, it may happen that a cosmic-ray has hit a star's gaussian profile. In such cases, it may be difficult to distinguish between a delta profile and a gaussian profile, and it leads to some different count of the star's data. There are more sophisticated methods to remove cosmic noise which are implemented by a package called IRAF (Image Reduction and Analysis Facility). More information about this can be found at: *http://iraf.noao.edu/*

• As noted earlier in Section 3, the magnitudes can be found out by observing a standard star and then comparing its instrumental magnitude to the known value of apparent magnitude. However, due to resource constraints, we were not able to perform this task, but it is always a good practice to observe a standard source and standardize the instrumental data accordingly.

• There is always an associated error with any observation. It comes from various sources such as readout and integration of the CCD, observing conditions, sky backround, etc. We can see that the sky's signal in figure 5 is also not constant but fluctuates randomly. Hence, by analysing the standard deviation (σ) of the sky, we can get the information of the error bar associated with our data.

• The true analysis of variaton over time of the source can be done by applying various statistical hypotheses testing algorithms. In this procedure, the error bar associated to the data is very important.

List of Figures

References

Web manuals and Documentations:

- PyFITS 3.4 Documentation:
  https://pythonhosted.org/pyfits/
- Matplotlib 2.0.2 Documentation, mplot3d tutorial:
  https://matplotlib.org/mpl_toolkits/mplot3d/tutorial.html
- SciPy Cookbook Documentation:
  http://scipy-cookbook.readthedocs.io

Books/articles:

- W. Romanishin, An Introduction to Astronomical Photometry Using CCDs, University of Oklahoma
- Optical Photometry by Sergio Ilovaisky (OHP), 5th NEON School, 23 July-5 August 2006)