Full Photometry Procedure

Abi Chown

Abstract

The procedure outlined in this document takes you through the full photometry pipeline, starting from CCD data, all the way through to plotting a light curve for a variable star. The parameters used here are not universal so should not be applied to every situation. They work on Spitzer Data from channels 1 and 2, that is wavelengths of $3.6\mu m$ and $4.5\mu m$. The IRAC (Infrared Array Camera) on board Spitzer has a pixel size of 1.2". The "raw" images that are the starting point of the pipeline are Level 2 BCD data so have already been flat-fielded etc.

1 File Setup

The files downloaded from the Spitzer Heritage Archive are given not very useful names, namely the folders are called 'r' followed by a string of numbers corresponding to the AOR key. Inside each of these directories are many subdirectories sorting the files into file type and channel they correspond to.

To make the file organisation easier to understand with more intuitive file names, I have organised my files in a hierechial way as depicted by Figure 1. This structure allows one to navigate easily through the data. In order to get my data into this format, I have created the script `file_setup.py`. First, this script creates a list of all the files in the original unsorted directory that need to be sorted. Two functions, `find_bcd_a_home` and `find_pbcd_a_home`

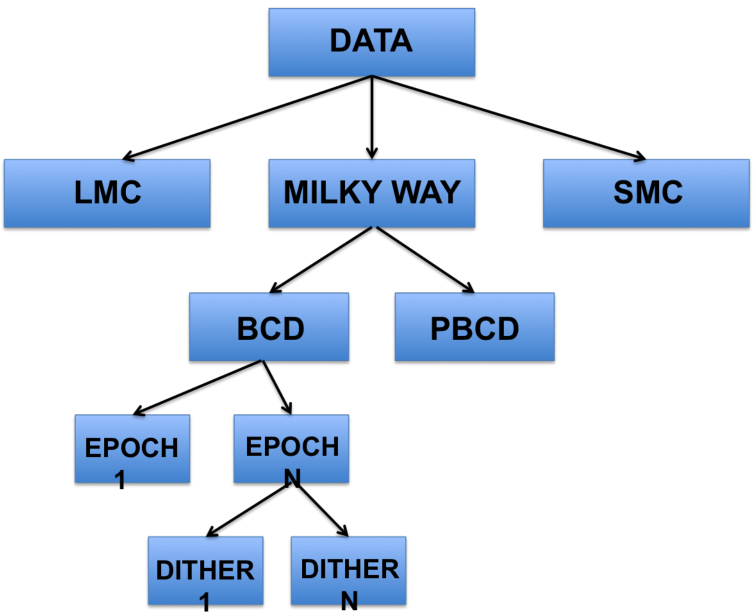

Figure 1: File structure for the data

then sort all the BCD and PBCD files respectively into their new organisation system. They do this by finding relevant keywords from the FITS headers such as target name, epoch number, galaxy and dither position. These keywords are not only used to sort the file into its correct place, but they also change the name of the file to something more comprehensible, given by `target_channel_epoch_dither_cbcd.fits` for BCD images and `target_channel_epoch_dither_munc.fits` where `munc` can be replaced by `maic` or `mcov`.

Write this notebook up as a script

Now that the files are in a more logical structure, each file is then converted to data counts using the Jupyter Notebook `convert_to_counts.ipynb`. The FITS images are in units of MJy/sr and so Equation 1 is applied, where $fluxconv$ and $exptime$ are obtained from the image header.

$$flux_{DN} = \frac{flux_{MJy/sr} \times exptime}{fluxconv} \tag{1}$$

The filenames are then amended to reflect this change of unit, namely `_dn` is added between the `cbcd` and the `.fits`.

Now that the files have been sorted into a better file system and their data converted to counts, we are now ready to use DAOPHOT to perform the aperture photometry.

2 Aperture Photometry

DAOPHOT.opt and PHOTO.opt

Script aperphot.py for finding stars and performing aperture photometry on them.