CAO Data Pipeline

Pre-Alpha

**Summary**

The goal of this pipeline is to prepare a batch of .fits file images for photometric analysis. I will be using .fits and images interchangeably throughout this user guide. There are four fundamental tasks that need to be completed for this to be happen:

1. Initial photometric analysis: Due to the nature of CCD observations, many files can be considered bad data. In the case of the CAO, many CCD images are skewed by wind while the image was captured. These files cannot be used in light curve analysis. The initial analysis is done using SExtractor.
2. Astrometry analysis: Utilizing the blind astrometry software astrometry.net, we can determine the exact coordinate of the field contained in each .fits file. This is crucial for the next task of stacking images.
3. Stacking images: With astrometry data for each image, the pipeline uses Scamp to stack respective images.
4. Data storage: Every time one of the previous tasks is finished, data of image name and status needs to be stored. MySQL is used to create a database that stores all data. PyMySQL is used to programmatically write and retrieve data.

After these tasks are completed for a batch of .fits files, SExtractor can be used again to gather all photometric data needed to perform light curve analysis on said batch.

**Program Flow**

The program itself is a set of scripts that automate the programs stated above in an organized and efficient manner while communicating to and from the database. All scripts are written in Python 3.4. The script files and contained methods are as follows:

1. Get\_files.py: This script contains one method and depends on the OS and Glob python modules:
   1. check\_files(): verifies that the user input is a valid directory and returns a list of .fit(s) or .FIT(S) file names.
2. Target\_Data.py: This script is a class dependent upon the numpy module. Any instance of this class will have the following properties/methods:
   1. target\_data: This is a variable that loads a list of targets into an array. This target list must be have three columns of comma separated values in the order ‘target name’, ‘RA in hour angles’, ‘Declination in degrees’. NOTE! Currently the location of the .txt file used to import the target data is hard coded into the script. This will be changed later
   2. bytes\_to\_str(): Converts data type of an array/list from ‘Byte’ to ‘str’.
   3. target\_dict(): Used to convert two lists into a dictionary. This is used in Astrometry\_automation.py for the script\_loop() function.
   4. coord\_lookup(): finds coordinates of a target from input dictionaries.
3. Astrometry\_automation.py: Dependent upon the subprocess and decimal python modules as well as Get\_files and Target\_Data.
   1. RA\_dict: dictionary with key values of target name and data value of RA.
   2. Dec\_dict: dictionary with key values of target name and data value of declination
   3. script\_loop(): Uses RA\_dict, Dec\_dict and functions from Target\_Data to loop through a list of .fit(s)/.FIT(S) files created from check\_files(). Each iteration calls astrometry.net and runs the program on a file.
4. SQL\_Connection: IN PROGRESS
5. Photometry\_verification: IN PROGRESS
6. Swarp: NOT STARTED.

Please refer to pipeline diagram for a graphical representation of the pipelin’s proposed structure.