

QUEST-La Silla AGN Variability Survey: Reduction Pipeline (*python version*)

Paula Sánchez

July 2016

Questions and comments to psanchez@das.uchile.cl

1 Overview

1.1 TELESCOPE AND CAMERA DESCRIPTION

Schmidt telescopes are the instrument of choice for large area sky surveys because of their large field of view. The QUEST Camera was designed to operate at the 48” Samuel Oschin Schmidt Telescope at the Palomar Observatory. Since 2009 the camera has been located in the 1-m Schmidt telescope of the European Southern Observatory (ESO) at La Silla, Chile. This telescope is one of the largest Schmidt configurations in the southern hemisphere, situated in a dry site with dark skies and good seeing. Having a nearly identical optical configuration to the Palomar Schmidt, the QUEST camera was installed at La Silla without any changes to its front-end optics. The survey uses the Q-band filter.

Telescope pointing and camera exposure are coordinated by a master scheduling program. A remote operator of one of the larger telescopes at the site (the ESO 3.6m) decides when conditions are appropriate for opening the telescope, and sends a remote command each night to enable the control software to open the dome. The control software automatically closes the dome whenever another nearby telescope (the 2.2m) is closed, when the sun rises, or when the remote operator sends a command to close. The remote operator can monitor and control the state of the Schmidt telescope via a web-based interface.

The CCD camera is located at the prime focus of the telescope about 3 m from the primary mirror. The camera consists of 112 CCDs arranged in four rows or “*fingers*” of 28 CCDs each, and covers $4.6^\circ \times 3.6^\circ$ (north–south by east–west) on the sky. The fingers are flagged A, B, C, and D and the columns of CCDs from 1 to 28.

Several CCDs have areas of high dark current owing to defective, electron-emitting pixels in the CCD substructure or to a bad readout amplifier. By subtracting dark calibration images, the unfavorable influence of these defects on source detection is largely eliminated. However, about 16% of the CCDs are useless because they are permanently off, randomly turn on and off, or have large defective areas which make it impossible to obtain an acceptable PSF, all of which hamper the astrometric solution due to the low number of stars detected and fake detections. The effective sky area covered by the functioning CCDs is $\sim 7.5 \text{ deg}^2$.

1.2 This pipeline

This pipeline is based on the methods described in Cartier et al. 2015:

For a typical night we take darks of 10, 60, and 180 seconds, as well as morning and evening twilight flats. The exposure times for our science images are either 60 or 180 seconds, with two image-pairs obtained per field per night.

Darks are combined to obtain a master dark for each detector and each exposure time. Then we subtract the master dark of the appropriate exposure time from the science images. The pixel-to-pixel variations of the science frames are corrected dividing them by the master flats obtained as we describe below. Finally, bad pixels in the science frames are identified with bad pixel masks. We construct bad pixel masks for each detector, for a certain period. To construct the bad pixel mask we use a median filter for each detector, taking as input the ratio between a high count twilight flat and a low count twilight flat.

Given the large size of the camera and the large field of view of the telescope, to obtain dome flats using a uniformly illuminated panel was unfeasible. Furthermore, the automatic operation of the telescope complicates a careful acquisition of dome flats. Instead twilight flats were preferred because they can be easily acquired automatically. Typically twilight flats are taken at the beginning (evening flats), and the end (morning flats) of the observing night. On many occasions flats were not useful because of moon illumination gradients, a closed dome, or cloudy conditions. To avoid these problems and to obtain a good pixel-to-pixel variation correction we select, from dark subtracted and trimmed twilight flats, the flats with mean counts above the average obtained over approximately two weeks of observations. Then we median combine these twilight flats and normalize them to obtain master flats for each detector. Typically we combine more than 50 twilight flats per detector.

The scripts described in this manual, are designed to be used in a computer (or cluster) with several cores. The user has to change the number of cores when it is specified in the manual, according to the number of cores available in their system. In our particular case, we work in a cluster with 64 cores and a memory of 129175MB.

The scripts were designed for the following folder structure: all the images are stored in a main folder (e.g. */home/main_folder/*), inside of this folder, for every epoch with available images there is a folder with the name of the epoch (e.g. */home/main_folder/20130614/*). Inside of the epoch folders there are folders for every image taken during the night (e.g. */home/main_folder/20130614/images073647s.041*). There are four types of images, sky or science (s), dark (d), morning flat (m), and evening flat (e). Inside every image folder we find fits files for every CCD, there are 112 fits files in total (e.g. */home/main_folder/20130614/images073647s.041/20130614073647s.B25.fits*).

2 Usage Instructions

The scripts used in this pipeline can be copied in any folder in your system, however, as will be explained soon, you will need to specify the location of the images. A good idea is to have a folder called "scripts" (e.g. /home/main_folder/scripts/) where you can copy the six scripts that compose the pipeline.

To run the scripts you need to have the following python packages installed: numpy, pyfits, astropy, ccdproc, multiprocessing, scikit-image, scipy, glob, os and time, besides astrometry.net and SExtractor. The parameters of every code can be changed in two ways. You can open the code and change the parameters below the expression "modify these and only these variables". You can also give the parameters in the command line, when you run the code. For instance, you can write in the terminal "dark_comb_parallel.py -p 201306 -n 28 -d /home/main_folder/".

It is important to note that not all the parameters have to change every time you run the code, probably you will only change the variables related with the period of time where you want to do the reduction process.

2.1 Master Darks

The script **dark_comb_parallel.py** generates the master darks for a certain period. The script was designed to work monthly, i.e. to generate all the master darks for a certain month. There are three variables that you can modify:

- period (e.g. period='201306' or -p 201306): specify for which month you want to generate the master darks.
- ncores (e.g. ncores=28 or -n 28): number of cores used to combine the darks. A high fraction of the total is not recommended, since some processes use multiple cores.
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.

After the code runs, you can find inside every epoch folder a new folder called "darks" (e.g. /home/main_folder/20130614/darks/). This folder contains the master darks for every CCD, for different exposure times (e.g. /home/main_folder/20130614/darks/Dark_180s_A03.fits)

2.2 Flats: master dark subtraction

The scripts **flat_dark_sub_m.py** and **flat_dark_sub_e.py**, subtract the master darks for every morning and evening flat respectively. The scripts were designed to work monthly. There are three variables that you can modify:

- period (e.g. period='201306' or -p 201306): specify for which month you want to subtract master darks from the flats.
- ncores (e.g. ncores=56 or -p 201306): number of cores used. A high fraction of the total is not recommended, since some processes use multiple cores.
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.

After the code runs, you can find inside every epoch folder a new folder called "flats" (e.g. /home/main_folder/20130614/flats/). This folder contains the master dark subtracted flats for every CCD (e.g. /home/main_folder/20130614/flats/DS.20130614104242m.C12.fits)

2.3 Master Flats

The script **flat_comb_parallel.py** generates the master flats for a certain period. The script was designed to work for every two weeks, in order to follow the methods described in the overview, therefore, you will need to run the code twice, first with '1st' (for the first two weeks of the month) and then with '2nd' (for the last two weeks of the month). There are six variables that you can modify:

- period (e.g. period='2013061st' or period='2013062nd' or -p 2013061st): specify for which period of the month you want to generate the master flats. If you are working in the first two weeks of a certain month use '1st', on the other hand, if you are working in the last two weeks use '2nd'.
- start_epoch (e.g. start_epoch=20130601 or -s 20130601): initial epoch considered in the period to combine, notice that in this case the variable is a number, not a string.
- final_epoch (e.g. final_epoch=20130615 or -f 20130615): final epoch considered in the period to combine, notice that in this case the variable is a number, not a string.
- ncores_rf (e.g. ncores_rf=15 or -r 15): number of cores used to read the flat info. 15 or less is recommended.
- ncores_cf (e.g. ncores_cf=28 or -c 28): number of cores used to combine the flats. It is not recommended to use a high fraction of the cores available.
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.

After the code runs, you can find inside the main folder a new folder called "MasterFlats" (in the case it was not created before), and inside of this one, another new folder with the name of the period (e.g. /home/main_folder/MasterFlats/2013061st/). This folder contains the master flats for every CCD (e.g. /home/main_folder/MasterFlats/2013061st/Flat_2013061st_C11.fits)

2.4 Reduction of the science images

The script **reduce_science_img.py** trim, subtract the master dark, and flat-field the science images in a certain month. There are four variables that you can modify:

- period (e.g. period='201306' or -d 201306): specify for which month you want to reduce the science images.
- ncores (e.g. ncores=56 or -n 56): number of cores used in the reduction process.
- mid_epoch (e.g. mid_epoch=20130615 or -m 20130615): epoch that defines the end of the 1st period and the start of the second period (for the flat correction). Notice that in this case the variable is a number, not a string.
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.

After the code runs, you can find inside every science image folder a new folder called "reduced" (e.g. /home/main_folder/20130614/images073647s.041/reduced/). This folder contains the reduced science images for every CCD, these images have in their names the prefix "R_" (e.g. /home/main_folder/20130614/images073647s.041/reduced/R_20130614073647s.B25.fits).

2.5 Astrometry

The script **astrometry_parallel.py** computes the astrometry for every image, using the open source codes **astrometry.net** and SExtractor. There are four variables that you can modify:

- period (e.g. period='201306' or -d 201306): specify for which month you want to compute the astrometric solution.
- ncores (e.g. ncores=56 or -n 56): number of cores used in the reduction process.
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.
- defa (e.g. defa='/home/main_folder/default_files' or -defa '/home/main_folder/default_files'): where are located the default files needed to run SExtractor.

After the code ends, you will find in the "reduced" folders for every science image, new files with the extension ".new.fits" and "cat.fits", the first ones correspond to the reduce science images with the astrometric solution, and the second ones correspond to the catalogs needed in the calibration step. It is important to note that some times the images have a really bad quality and "astrometry.net" cannot find a astrometric solution.

2.6 pre-Calibration

The script **sel_sci_images.py** creates a txt file with the log information of all the science images for a certain period. The julian date and the field can be find in this files. Run this code before the calibration process. There are three variables that you can modify:

- period (e.g. period='201306' or -p 201306): specify for which month you want to compute the calibration.
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.
- defa (e.g. defa='/home/main_folder/default_files/datalogs' or -defa /home/main_folder/default_files/datalogs) where are located the observing logs.

At the end, you will find in the folder "defa" a new file called "sci_images_period.txt" (e.g sci_images_201306.txt).

2.7 Calibration

The script **calibration_parallel.py** computes the calibration of the science images, using external photmetric catalogs (in our case using SDSS and DES data). The code compute the calibration using the method explained in Cartier et al. 2015. To correct the non-linearities we inspected the residuals of the difference between the q instrumental magnitudes and the $(g + r)_{SDSS}$ (or $(g + r)_{DES}$) magnitudes as a function of q . A linear fit as function of q is significantly better representation of the residuals than a constant zero point. For this reason we decided to use a linear expression of the following form to fit the residuals:

$$q - (g + r)_{SDSS} = \alpha + \beta \times q \quad (1)$$

To avoid distortions in the fit due to very bright and faint sources we only used objects in the interval $13.5 < q < 17.5$, and we sigma-clipped the fit to eliminate outliers

The code computes the calibration for our five extragalactic fields, for COSMOS, Stripe82 and XMM-LSS we use SDSS photometry, and for ElaisS1 and ECDFS we use DES photometry. It really important to run in advance **astrometry_parallel.py** and **sel_sci_images.py**, since the data products of these two codes are used in the process. There are four variables that you can modify:

- period (e.g. period='201306' or -p 201306): specify for which month you want to compute the calibration.
- ncores (e.g. ncores=28 or -n 28): number of cores used in the reduction process.
- main_path (e.g. main_path='/home/main_folder/' or -d '/home/main_folder/'): where all the images are stored.
- defa (e.g. defa='/home/main_folder/default_files' or -defa /home/main_folder/default_files): where are located the default files needed to run SExtractor.

After the code ends, you will find in the "reduced" folders for every science image, new files with the extension ".phot.fits", which correspond to the calibrated catalogs.

2.8 Masks

The script **gen_masks.py** generates the Masks for a certain period. The Masks have the same size of the science images, and are constructed in a way that bad pixels have a value of 1 and good pixels a value of 0. The script was designed to work in a period between two epochs specified by the user. There are five variables that you can modify:

- period (e.g. period='201306' or -p 201306): specify the period for which you want to generate the Masks. It can be a month or a whole year.
- start_epoch (e.g. start_epoch=20130601 or -s 20130601): initial epoch considered in the period to combine, notice that in this case the variable is a number, not a string.
- final_epoch (e.g. final_epoch=20130630 or -f 2013063): final epoch considered in the period to combine, notice that in this case the variable is a number, not a string.
- ncores (e.g. ncores=14 or -n 14): number of cores used to combine the flats and generate the masks. It is not recommended to use a high fraction of the cores available. In this case, we use 14 of 64
- main_path (e.g. main_path='/home/main_folder/' or -d /home/main_folder/): where all the images are stored.

After the code runs, you can find inside the main folder a new folder called "Masks" (in the case it was not created before), and inside of this one, another new folder with the name of the period (e.g. /home/main_folder/Masks/201306/). This folder contains the Masks for every CCD (e.g. /home/main_folder/Masks/201306/Mask_201306_B15.fits). Besides masks, fake variance maps are generated (e.g. /home/main_folder/Masks/201306/VarMap_201306_D06.fits), these are maps with the same size of the science images, and are constructed in a way that bad pixels have a value of 10^{30} and good pixels a value of 1. The maps were designed to be used when you run SExtractor (Bertin & Arnouts 1996), to let the code know which pixels have to be avoided.

2.9 Usage example

Lets suppose you want to reduce the data for the period 201306, then you will need to do the following:

- run **dark_comb_parallel.py** with the variables: `period='201306'`, `ncores=28`,
`main_path='/home/main_folder/'`
- run **flat_dark_sub_m.py** with the variables: `period='201306'`, `ncores=56`,
`main_path='/home/main_folder/'`
- run **flat_dark_sub_e.py** with the variables: `period='201306'`, `ncores=56`,
`main_path='/home/main_folder/'`
- run **flat_comb_parallel.py** with the variables: `period='2013061st'`, `start_epoch=20130601`,
`final_epoch=20130615`, `ncores_rf=15`, `ncores_cf=28`,
`main_path='/home/main_folder/'`
- run again **flat_comb_parallel.py**, but now with the variables: `period='2013062nd'`,
`start_epoch=20130616`, `final_epoch=20130630`, `ncores_rf=15`, `ncores_cf=28`,
`main_path='/home/main_folder/'`
- run **reduce_science_img.py** with the variables: `period='201306'`, `ncores=56`, `mid_epoch=20130615`
`main_path='/home/main_folder/'`
- run **astrometry_parallel.py** with the variables: `period='201306'`, `ncores=28`,
`main_path='/home/main_folder/'`, `defa='/home/main_folder/default_files'`
- run **sel_sci_images.py** with the variables: `period='201306'`, `ncores=56`,
`main_path='/home/main_folder/datalogs/'`
- run **calibration_parallel.py** with the variables: `period='201306'`, `ncores=28`,
`main_path='/home/main_folder/'`, `defa='/home/main_folder/default_files/'`
- run **gen_masks.py** with the variables: `period='201306'`, `start_epoch=20130601`,
`final_epoch=20130630`, `ncores_rf=14`, `ncores_gm=14`,
`main_path='/home/main_folder/'`

For the special case of the Masks, you may want to consider a larger period, like a year, in that case, adjust your variables accordingly.