Alex's Quick Start Guide to the ARCONS Pipeline

Created: Feb 3, 2016

0. Set up your system

Download the pipeline to your home (or wherever) directory

```
git clone https://github.com/bmazin/ARCONS-pipeline.git
```

Add the following variables to your .bash_profile

```
MKID_BEAMMAP_PATH=/ScienceData/sci3gamma/beamimage_sci3gamma.h5
MKID_RAW_PATH=/ScienceData
MKID_PROC_PATH=/Scratch
MKID_CAL_LOOKUP=/Scratch/calLookup/lookup.h5
PYTHONPATH=$HOME/ARCONS-pipeline
export MKID_BEAMMAP_PATH
export MKID_RAW_PATH
export MKID_PROC_PATH
export MKID_CAL_LOOKUP
export PYTHONPATH=$PYTHONPATH:$PYTHONPATH/util
PATH = $PYTHONPATH:/usr/local/epd/bin:/usr/local/Trolltech/Qt-4.8.3/bin:
$HOME/bin:$PATH
export PATH
```

1. Decide the target you want to analyze

The list of targets and other information can be found on the google drive Palomar folder:

https://drive.google.com/folderview?id=0B6vM_l0m98k9VDYyVzdmcWZQT1k&usp=sharing

Additionally, a copy of the detailed observation log is on the Turk Science hard drive. You'll need to ssh into Turk and access the /ScienceData/ directory. Ex. /ScienceData/PAL2014/Pal2014-2ObservingLog.pdf

Identify the date and times when your target was observed (some targets are observed on multiple days). For the purposes of this log, we'll examine X2 in M82, the pulsar previously identified as a ULX intermediate mass black hole.

X2 in M82	ULX and pulsar	20141021	11:48 – 12:45

2. Preview the observations in ObsFileViewer

ObsFileViewer.py is the new and improved quicklook.py GUI made in 2015. To use it, navigate to your local pipeline directory and use the following command from the terminal:

```
python ~/ARCONS-pipeline/quicklook/ObsFileViewer.py
```

This will open the GUI. Load an obs file by clicking “File -> Load Obs File...” For example, I'll choose (somewhat randomly) /PAL2014/20141021/obs_20141022-120843.h5. Check under “Windows -> Header Info” that the target is correct and look at the description if there is any.

Load the calibration files (wavelength cal, flat field cal, flux cal, time mask etc...) by clicking

"File -> Load cal files..." The appropriate files should be populated by default from the calLookUp file located in the system path $MKID_CAL_LOOKUP. Select "Windows -> Image Plot Parameters" to adjust which cals you want to apply to the preview image. Then press the "Plot" button on the GUI.

You can examine an individual pixel by clicking "Windows -> New Plot Window" and selecting the data you want (timestream, spectrum, etc..)

3. Initialize hot pixel mask

If the obs file you're analyzing doesn't have a timemask file you need to create one. You do this with the hotpixel code found in /hotpix/hotPixels.py. Here is an example python script:

```
from util.ObsFile import ObsFile
import hotpix.hotPixels as hp
fname = '/ScienceData/PAL2014/20141021/obs_20141022-120843.h5'
outputFileName = '/Scratch/timeMasks/20141021/timeMask_20141022-120843.h5
obs = ObsFile(fname)
obs.loadBeammapFile('/ScienceData/PAL2014/beammap_SCI6_B140731-
Boba_20141118flip.h5')  #It's important to load the correct beammap file!
obs.loadAllCals(timeMaskPath='')
obs.setWvlCutoffs(wvlLowerLimit=3500, wvlUpperLimit=8000)           # You can play
around with the wavelength limits but the longer wavelengths (after 9000 A) get noisy
hp.findHotPixels(obsFile=obs, outputFileName=outputFileName, timeStep=1,
startTime=0, endTime=-1, badTimeBuffer = 5.,fwhm=2.5, boxSize=5, nSigmaHot=3.0,
nSigmaCold=1.5, display=False, weighted=True, fluxWeighted=False, maxIter=5,
dispMinPerc=0.0, dispMaxPerc=98.0, diagnosticPlots=False, useLocalStdDev=None,
useRawCounts=False, bkgdPercentile=10.0, deadTime=100.e-6)
```

4. Update time mask by hand with ObsFileViewer.py

Click on a pixel that might be hot and open the light curve window. Highlight the times that it is hot, select reason: Hot Pixel, and hit mask out selection