Python Astro: Dark, flat corrections

Author: A. Klotz
Updated : 2022-02-24

Contents:

- Load and save FITS images
- Get keyword values of the header
- Visualize images and get pixel values with a cursor
- Make a median stack of images
- Apply dark and flat corrections
- Extract sources from an image

Level: Beginners
Python prerequisite: Spyder, Numpy, Matplotlib
Astro prerequisite: Dark, Flat, FITS

Load and save FITS images - Direct programming

```
from astropy.io import fits

# Set the name of the FITS file
fitsname = "oneimage.fits"

# Load the image in memory
hdu_list = fits.open(fitsname, memmap=False)
hdu = hdu_list[0]
array = hdu.data
hdu_list.close()

# Load the image in memory
hdu.data = array
hdu.writeto(fitsname, overwrite=True)
```

HDU = Header Data Unit

We select the first HDU

We extract the numpy array

Easy but difficult to build a long program by this way (lot of copy/paste)

Load and save FITS images - Using a class

```
from astropy.io import fits

class FitsIma:

    # === Attributes
    array = None
    hdu = None

    # === Methods
    def savefits( self, fitsname: str ) :
        self.hdu.data = self.array
        self.hdu.writeto(fitsname, overwrite=True)

    def loadfits( self, fitsname: str ) :
        hdu_list = fits.open(fitsname, memmap=False)
        self.hdu = hdu_list[0]
        self.array = self.hdu.data
        hdu_list.close()

if __name__ == "__main__":
    test = 1

    if test == 1:
        # Just load and save an image
        ima = FitsIma()
        fitsname = "oneimage.fits"
        ima.loadfits(fitsname)
        ima.savefits(fitsname)
```

Attributes are variables shared in all methods

A function defined inside a class is called method

The following code is executed only when the script is launched directly.

It becomes easy to build a long program by this way

Managing FITS images - Completing the class

The script python_astro_step2.py
provides a FitsIma class
able to make dartk and flat corrections

To compute a master dark from 5 darks (dark60s-1.fits to dark60s-5.fits):

```
stack_median("dark60s-", 5, ".fits")
ima = FitsIma()
ima.loadfits("dark60s-median.fits")
ima.visu()
```

Managing FITS images - Completing the class

The script python_astro_step2.py
provides a FitsIma class
able to make dartk and flat corrections

To compute a master flat from 5 flats (flat-1.fits to flat-5.fits):

```
dark = FitsIma()
dark.loadfits("dark5s-median.fits")
flat = FitsIma()
for i in range(1,6):
    flat.loadfits(f"flat-{i}.fits")
    flat.sub(dark)
    flat.ngain(10000)
    ima.savefits(f"flat_{i}.fits")
stack_median("flat_", 5, ".fits")
ima.loadfits("flat_median.fits")
ima.visu()
```

Managing FITS images - Completing the class

The script python_astro_step2.py
provides a FitsIma class
able to make dartk and flat corrections

To make corrections of a raw image by dark and flat and extract sources:

```
ima = FitsIma()
ima.loadfits("rawimage.fits")
dark = FitsIma()
dark.loadfits("dark60s-median.fits")
flat = FitsIma()
flat.loadfits("flat-median.fits")
ima.sub(dark)
ima.div(flat)
ima.array *= np.mean(flat.array)
stars = ima1.sextractor(2)
np.savetxt("sources.txt", stars)
ima.savefits("processedimage.fits")
```