PHY319ObservingProject

May 8, 2018

Python module to align different frames: 1. Take all R-band images. 2. Identify x,y positions of the same bright stars in all images (either automatically or by-eye). 3. Work out the x and y shift between the images. 4. Shift images so the stars align (Have the same x-y positions). 5. Mean-combine the images to increase the S/N. (Also to try median combine) 6. Do the same for Halpha images. The galaxy will start to appear in Halpha after adding more images. This will increase the S/N. 7. Subtract the R-band. Before that, the R-band image should be normalised. On normalising, the stars in the Halpha image disappear after subtraction. This leaves with the galaxy in Halpha.

```
In [1]: #Reduction of images of M61
        from astropy.io import fits
        import os
        import numpy as np
        from astropy.visualization import (MinMaxInterval, SqrtStretch, ImageNormalize)
        # display plots in notebook
        %matplotlib inline
        import matplotlib.pyplot as plt
        import ccdproc
        from astropy import units as u

        #assigning directories
        data_dir = '../../assignments/Session5/M61/raw_data'
        destination_dir = '../../assignments/Session5/M61/calib_data'

        images = ccdproc.ImageFileCollection(location=data_dir,keywords=['type','exposure','filt
```

```
In [2]: print(images.summary) #the image types of the images are fine
        #hdu_list_b1 = fits.open('../../assignments/Session5/M61/raw_data/m61_h_150_2.fits')
        #header_b1 = hdu_list_b1[0].header
        #print(header_b1['type'])
```

```
       file         type   exposure filter xbinning
----------------- -------- -------- ------ --------
 m61_h_150_1.fits SCIENCE    150.0      H        2
m61_h_150_10.fits SCIENCE    150.0      H        2
m61_h_150_11.fits SCIENCE    150.0      H        2
m61_h_150_12.fits SCIENCE    150.0      H        2
m61_h_150_13.fits SCIENCE    150.0      H        2
```

```
m61_h_150_14.fits SCIENCE    150.0      H        2
m61_h_150_15.fits SCIENCE    150.0      H        2
m61_h_150_16.fits SCIENCE    150.0      H        2
m61_h_150_17.fits SCIENCE    150.0      H        2
m61_h_150_18.fits SCIENCE    150.0      H        2
m61_h_150_19.fits SCIENCE    150.0      H        2
 m61_h_150_2.fits SCIENCE    150.0      H        2
m61_h_150_20.fits SCIENCE    150.0      H        2
 m61_h_150_3.fits SCIENCE    150.0      H        2
 m61_h_150_4.fits SCIENCE    150.0      H        2
 m61_h_150_5.fits SCIENCE    150.0      H        2
 m61_h_150_6.fits SCIENCE    150.0      H        2
 m61_h_150_7.fits SCIENCE    150.0      H        2
 m61_h_150_8.fits SCIENCE    150.0      H        2
 m61_h_150_9.fits SCIENCE    150.0      H        2
 m61_r_240_1.fits SCIENCE    240.0      R        2
 m61_r_240_2.fits SCIENCE    240.0      R        2
```

```
In [3]: calib_frames = ccdproc.ImageFileCollection(location=destination_dir, keywords=['type','e

        print(calib_frames.summary)
```

```
           file             type     exposure   filter xbinning
------------------------- ------- ------------- ------ --------
        Bflatnorm1x1.fits     SKY 2.54617976223      B        1
        Bflatnorm2x2.fits     SKY 2.54617976223      B        2
        Bflatnorm3x3.fits     SKY 2.54617976223      B        3
        Hflatnorm1x1.fits     SKY 21.1649156994      H        1
        Hflatnorm2x2.fits     SKY 21.1649156994      H        2
        Hflatnorm3x3.fits     SKY 21.1649156994      H        3
        Iflatnorm1x1.fits     SKY 3.75051017342      I        1
        Iflatnorm2x2.fits     SKY 3.75051017342      I        2
        Iflatnorm3x3.fits     SKY 3.75051017342      I        3
        Rflatnorm1x1.fits     SKY 2.20479702354      R        1
                      ...     ...           ...    ...      ...
m61_h_150_20_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_2_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_3_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_4_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_5_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_6_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_7_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_8_reduced.fits SCIENCE         150.0      H        2
 m61_h_150_9_reduced.fits SCIENCE         150.0      H        2
 m61_r_240_1_reduced.fits SCIENCE         240.0      R        2
 m61_r_240_2_reduced.fits SCIENCE         240.0      R        2
Length = 43 rows
```

```
In [3]: def getCalibs(hdu):
            filtname = hdu.header['filter']
            xbin = hdu.header['xbinning']
            dark, bias, flat = None, None, None
            for hdu in calib_3184_3.ccds(xbinning=2, ccd_kwargs=dict(unit=u.adu)):
                if hdu.header['type'] == 'DARK':
                    dark = hdu
                elif hdu.header['type'] == 'BIAS':
                    bias = hdu
                elif hdu.header['type'] == 'SKY' and hdu.header['filter'] == filtname:
                    flat = hdu
            return dark, bias, flat
```

```
In [7]: for ccd, fname in images.ccds(return_fname=True, ccd_kwargs=dict(unit=u.adu)):
            dark, bias, flat = getCalibs(ccd)
            bias_subtracted = ccdproc.subtract_bias(ccd, bias)
            dark_subtracted = ccdproc.subtract_dark(bias_subtracted, dark, exposure_time='exposu
            reduced_image = ccdproc.flat_correct(dark_subtracted, flat)
            root, ext = os.path.splitext(fname)
            ofname = os.path.join(destination_dir, root) + '_reduced.fits'
            reduced_image.write(ofname)
```

```
WARNING: FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa.
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi


INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as


INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi


INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as


INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi


INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi
```

```
INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi

INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as

INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi

INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as

INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi

INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as

INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi

INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as

INFO:astropy:using the unit adu passed to the FITS reader instead of the unit adu in the FITS fi

INFO: using the unit adu passed to the FITS reader instead of the unit adu in the FITS file. [as
```

```
In [5]: os.listdir(destination_dir)
```

```
Out[5]: ['m61_h_150_20_reduced.fits',
         'Iflatnorm3x3.fits',
         'Rflatnorm1x1.fits',
         'bias3x3.fits',
         'Bflatnorm2x2.fits',
         'dark3x3.fits',
         'Hflatnorm3x3.fits',
         'Vflatnorm3x3.fits',
         'Rflatnorm2x2.fits',
         'M61_calibratedimages.zip',
         'Bflatnorm1x1.fits',
         'Hflatnorm2x2.fits',
         'm61_h_150_10_reduced.fits',
         'm61_h_150_11_reduced.fits',
         'm61_h_150_13_reduced.fits',
         'Bflatnorm3x3.fits',
         'dark2x2.fits',
         'm61_h_150_12_reduced.fits',
         'bias2x2.fits',
         'Vflatnorm1x1.fits',
         'm61_r_240_1_reduced.fits',
         'm61_h_150_17_reduced.fits',
         'm61_h_150_8_reduced.fits',
         'm61_h_150_9_reduced.fits',
         'm61_h_150_16_reduced.fits',
         'Iflatnorm2x2.fits',
         'm61_h_150_14_reduced.fits',
         'm61_r_240_2_reduced.fits',
         'm61_h_150_15_reduced.fits',
         'dark1x1.fits',
         'm61_h_150_1_reduced.fits',
         'm61_h_150_2_reduced.fits',
         'Hflatnorm1x1.fits',
         'm61_h_150_3_reduced.fits',
         'm61_h_150_19_reduced.fits',
         'm61_h_150_6_reduced.fits',
         'm61_h_150_7_reduced.fits',
         'm61_h_150_18_reduced.fits',
         'm61_h_150_5_reduced.fits',
         'Rflatnorm3x3.fits',
         'Iflatnorm1x1.fits',
         'm61_h_150_4_reduced.fits',
         'Vflatnorm2x2.fits',
         'bias1x1.fits']
```

```
In [3]: #reduction of images of ngc3184_1
        #assigning directories
        data_dir_3184_1 = '../../assignments/Session5/NGC3184_1/raw_data_1'
        destination_dir_3184_1 = '../../assignments/Session5/NGC3184_1/calib_data_1'

        images_3184_1 = ccdproc.ImageFileCollection(location=data_dir_3184_1,keywords=['type','e
        #print(images_3184_1.summary)
        calib_3184_1 = ccdproc.ImageFileCollection(location=destination_dir_3184_1,keywords=['ty
        #print(calib_3184_1.summary)
```

```
      file        type    exposure    filter xbinning
----------------- ---- ------------- ------ --------
Bflatnorm1x1.fits  SKY 2.71539647751      B        1
Bflatnorm2x2.fits  SKY 2.71539647751      B        2
Bflatnorm3x3.fits  SKY 2.71539647751      B        3
Hflatnorm1x1.fits  SKY 21.2001404396      H        1
Hflatnorm2x2.fits  SKY 21.2001404396      H        2
Hflatnorm3x3.fits  SKY 21.2001404396      H        3
Iflatnorm1x1.fits  SKY 4.24281279695      I        1
Iflatnorm2x2.fits  SKY 4.24281279695      I        2
Iflatnorm3x3.fits  SKY 4.24281279695      I        3
```

```
Rflatnorm1x1.fits  SKY 2.51441818049      R        1
Rflatnorm2x2.fits  SKY 2.51441818049      R        2
Rflatnorm3x3.fits  SKY 2.51441818049      R        3
Vflatnorm1x1.fits  SKY 3.39509675375      V        1
Vflatnorm2x2.fits  SKY 3.39509675375      V        2
Vflatnorm3x3.fits  SKY 3.39509675375      V        3
     bias1x1.fits BIAS          0.03      H        1
     bias2x2.fits BIAS          0.03      H        2
     bias3x3.fits BIAS          0.03      H        3
     dark1x1.fits DARK          60.0      H        1
     dark2x2.fits DARK          60.0      H        2
     dark3x3.fits DARK          60.0      H        3
```

```
In [8]: for ccd, fname in images_3184_1.ccds(return_fname=True, ccd_kwargs=dict(unit=u.adu)):
            dark, bias, flat = getCalibs(ccd)
            bias_subtracted = ccdproc.subtract_bias(ccd, bias)
            dark_subtracted = ccdproc.subtract_dark(bias_subtracted, dark, exposure_time='exposu
            reduced_image = ccdproc.flat_correct(dark_subtracted, flat)
            root, ext = os.path.splitext(fname)
            ofname = os.path.join(destination_dir_3184_1, root) + '_reduced.fits'
            reduced_image.write(ofname)
```

```
WARNING: FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa.
```

```
        ---------------------------------------------------------------------------

        OSError                                   Traceback (most recent call last)

        <ipython-input-8-a7af521dd2af> in <module>()
          6     root, ext = os.path.splitext(fname)
          7     ofname = os.path.join(destination_dir_3184_1, root) + '_reduced.fits'
    ----> 8     reduced_image.write(ofname)
          9 os.listdir(destination_dir_3184_1)


        /ext/anaconda3/lib/python3.5/site-packages/astropy/nddata/mixins/ndio.py in write(self,
         35         formats.
         36         """
    ---> 37         io_registry.write(self, *args, **kwargs)


        /ext/anaconda3/lib/python3.5/site-packages/astropy/io/registry.py in write(data, format,
        558 
        559     writer = get_writer(format, data.__class__)
    --> 560     writer(data, *args, **kwargs)
        561 
        562 


        /ext/anaconda3/lib/python3.5/site-packages/astropy/nddata/ccddata.py in fits_ccddata_wri
        569     hdu = ccd_data.to_hdu(hdu_mask=hdu_mask, hdu_uncertainty=hdu_uncertainty,
        570                           hdu_flags=hdu_flags)
    --> 571     hdu.writeto(filename, **kwd)
        572 
        573 


        /ext/anaconda3/lib/python3.5/site-packages/astropy/utils/decorators.py in wrapper(*args,
        485                         # one with the name of the new argument to the function
        486                         kwargs[new_name[i]] = value
    --> 487             return function(*args, **kwargs)
        488 
        489         return wrapper


        /ext/anaconda3/lib/python3.5/site-packages/astropy/io/fits/hdu/hdulist.py in writeto(sel
        865         # file object that's open to write only, or in append/update modes
        866         # but only if the file doesn't exist.
    --> 867         fileobj = _File(fileobj, mode='ostream', overwrite=overwrite)
        868         hdulist = self.fromfile(fileobj)
        869         try:


        /ext/anaconda3/lib/python3.5/site-packages/astropy/utils/decorators.py in wrapper(*args,
        485                         # one with the name of the new argument to the function
        486                         kwargs[new_name[i]] = value
    --> 487             return function(*args, **kwargs)
        488 
        489         return wrapper


        /ext/anaconda3/lib/python3.5/site-packages/astropy/io/fits/file.py in __init__(self, fil
        173             self._open_fileobj(fileobj, mode, overwrite)
        174         elif isinstance(fileobj, str):
    --> 175             self._open_filename(fileobj, mode, overwrite)
        176         else:
        177             self._open_filelike(fileobj, mode, overwrite)


        /ext/anaconda3/lib/python3.5/site-packages/astropy/io/fits/file.py in _open_filename(sel
        515
        516         if mode == 'ostream':
    --> 517             self._overwrite_existing(overwrite, None, True)
        518
        519         if os.path.exists(self.name):


        /ext/anaconda3/lib/python3.5/site-packages/astropy/io/fits/file.py in _overwrite_existin
        412                     os.remove(self.name)
        413             else:
    --> 414                 raise OSError("File {!r} already exists.".format(self.name))
        415
        416     def _try_read_compressed(self, obj_or_name, magic, mode, ext=''):


        OSError: File '../../assignments/Session5/NGC3184_1/calib_data_1/ngc3184_h_240_1_reduced
```

```
In [9]: os.listdir(destination_dir_3184_1)
```

```
Out[9]: ['Bflatnorm1x1.fits',
         'Vflatnorm3x3.fits',
         'Rflatnorm2x2.fits',
         'bias3x3.fits',
         'Hflatnorm3x3.fits',
         'Bflatnorm2x2.fits',
         'Rflatnorm1x1.fits',
         'Iflatnorm3x3.fits',
         'dark3x3.fits',
         'Iflatnorm1x1.fits',
         'Rflatnorm3x3.fits',
         'Vflatnorm2x2.fits',
         'dark1x1.fits',
         'bias1x1.fits',
         'Hflatnorm1x1.fits',
         'dark2x2.fits',
         'Vflatnorm1x1.fits',
         'Iflatnorm2x2.fits',
         'Bflatnorm3x3.fits',
         'Hflatnorm2x2.fits',
         'ngc3184_h_240_1_reduced.fits',
         'bias2x2.fits']
```

```
In [2]: #reduction of images of ngc3184_2
        #assigning directories
        data_dir_3184_2 = '../../assignments/Session5/NGC3184_2/raw_data_2'
        destination_dir_3184_2 = '../../assignments/Session5/NGC3184_2/calib_data_2'
        images_3184_2 = ccdproc.ImageFileCollection(location=data_dir_3184_2,keywords=['type','e
        print(images_3184_2.summary)
        calib_3184_2 = ccdproc.ImageFileCollection(location=destination_dir_3184_2,keywords=['ty
        print(calib_3184_2.summary)
```

```
        file           type  exposure filter xbinning
-------------------- ------- -------- ------ --------
ngc3184_h_240_2.fits SCIENCE    240.0      H        3
    file     type exposure filter xbinning
------------ ---- -------- ------ --------
bias1x1.fits BIAS     0.03      H        1
bias2x2.fits BIAS     0.03      H        2
bias3x3.fits BIAS     0.03      H        3
dark1x1.fits DARK     60.0      H        1
dark2x2.fits DARK     60.0      H        2
dark3x3.fits DARK     60.0      H        3
```

```
In [7]: for ccd, fname in images_3184_2.ccds(return_fname=True, ccd_kwargs=dict(unit=u.adu)):
            dark, bias, flat = getCalibs(ccd)
            bias_subtracted = ccdproc.subtract_bias(ccd, bias)
            dark_subtracted = ccdproc.subtract_dark(bias_subtracted, dark, exposure_time='exposu
            reduced_image = ccdproc.flat_correct(dark_subtracted, flat)
            root, ext = os.path.splitext(fname)
            ofname = os.path.join(destination_dir_3184_2, root) + '_reduced.fits'
            reduced_image.write(ofname)
```

```
WARNING: FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa.
```

```
        ---------------------------------------------------------------------------

        ValueError                                Traceback (most recent call last)

        <ipython-input-7-652e23d1a9e0> in <module>()
          1 for ccd, fname in images_3184_2.ccds(return_fname=True, ccd_kwargs=dict(unit=u.adu))
          2     dark, bias, flat = getCalibs(ccd)
    ----> 3     bias_subtracted = ccdproc.subtract_bias(ccd, bias)
          4     dark_subtracted = ccdproc.subtract_dark(bias_subtracted, dark, exposure_time='ex
          5     reduced_image = ccdproc.flat_correct(dark_subtracted, flat)


        /ext/anaconda3/lib/python3.5/site-packages/ccdproc/log_meta.py in wrapper(*args, **kwd)
         94         # Grab the logging keyword, if it is present.
         95         log_result = kwd.pop(_LOG_ARGUMENT, True)
```

```
---> 96         result = func(*args, **kwd)
     97 
     98         if not log_result:


    /ext/anaconda3/lib/python3.5/site-packages/ccdproc/core.py in subtract_bias(ccd, master)
    572                                "image".format(ccd.unit, master.unit))
    573         else:
--> 574             raise e
    575 
    576     result.meta = ccd.meta.copy()


    /ext/anaconda3/lib/python3.5/site-packages/ccdproc/core.py in subtract_bias(ccd, master)
    565 
    566     try:
--> 567         result = ccd.subtract(master)
    568     except ValueError as e:
    569         if 'operand units' in str(e):


    /ext/anaconda3/lib/python3.5/site-packages/astropy/nddata/ccddata.py in inner(self, oper
     47             _config_ccd_requires_unit = False
     48             result = self._prepare_then_do_arithmetic(op, operand,
---> 49                                                       operand2, **kwargs)
     50             # Wrap it again as CCDData so it checks the final unit.
     51             _config_ccd_requires_unit = True


    /ext/anaconda3/lib/python3.5/site-packages/astropy/nddata/mixins/ndarithmetic.py in _pre
    612 
    613         # Now call the _arithmetics method to do the arithmetics.
--> 614         result, init_kwds = operand._arithmetic(operation, operand2, **kwargs)
    615 
    616         # Return a new class based on the result


    /ext/anaconda3/lib/python3.5/site-packages/astropy/nddata/mixins/ndarithmetic.py in _ari
    244         # Then calculate the resulting data (which can but not needs to be a
    245         # quantity)
--> 246         result = self._arithmetic_data(operation, operand, **kwds2['data'])
    247 
    248         # Determine the other properties


    /ext/anaconda3/lib/python3.5/site-packages/astropy/nddata/mixins/ndarithmetic.py in _ari
    320         else:
    321             result = operation(self.data * self.unit,
--> 322                                 operand.data * operand.unit)
    323
    324         return result


    ValueError: operands could not be broadcast together with shapes (490,728) (736,1092)
```

```
In [2]: #reduction of images of ngc3184_3
        #assigning directories
        data_dir_3184_3 = '../../assignments/Session5/NGC3184_3/raw_data_3'
        destination_dir_3184_3 = '../../assignments/Session5/NGC3184_3/calib_data_3'

        images_3184_3 = ccdproc.ImageFileCollection(location=data_dir_3184_3,keywords=['type','e
        print(images_3184_3.summary)
        calib_3184_3 = ccdproc.ImageFileCollection(location=destination_dir_3184_3,keywords=['ty
        print(calib_3184_3.summary)
```

```
        file           type  exposure filter xbinning
-------------------- ------- -------- ------ --------
ngc3184_r_240_1.fits SCIENCE    240.0      R        2
ngc3184_r_240_2.fits SCIENCE    240.0      R        2
ngc3184_r_240_3.fits SCIENCE    240.0      R        2
       file       type    exposure    filter xbinning
----------------- ---- ------------- ------ --------
Bflatnorm1x1.fits  SKY 2.54617976223      B        1
Bflatnorm2x2.fits  SKY 2.54617976223      B        2
Bflatnorm3x3.fits  SKY 2.54617976223      B        3
Hflatnorm1x1.fits  SKY 21.1649156994      H        1
Hflatnorm2x2.fits  SKY 21.1649156994      H        2
Hflatnorm3x3.fits  SKY 21.1649156994      H        3
Iflatnorm1x1.fits  SKY 3.75051017342      I        1
Iflatnorm2x2.fits  SKY 3.75051017342      I        2
Iflatnorm3x3.fits  SKY 3.75051017342      I        3
Rflatnorm1x1.fits  SKY 2.20479702354      R        1
Rflatnorm2x2.fits  SKY 2.20479702354      R        2
Rflatnorm3x3.fits  SKY 2.20479702354      R        3
Vflatnorm1x1.fits  SKY 3.00486467614      V        1
Vflatnorm2x2.fits  SKY 3.00486467614      V        2
Vflatnorm3x3.fits  SKY 3.00486467614      V        3
     bias1x1.fits BIAS          0.03      H        1
     bias2x2.fits BIAS          0.03      H        2
     bias3x3.fits BIAS          0.03      H        3
     dark1x1.fits DARK          60.0      H        1
     dark2x2.fits DARK          60.0      H        2
     dark3x3.fits DARK          60.0      H        3
```

```
In [4]: for ccd, fname in images_3184_3.ccds(return_fname=True, ccd_kwargs=dict(unit=u.adu)):
```

```
            dark, bias, flat = getCalibs(ccd)
            bias_subtracted = ccdproc.subtract_bias(ccd, bias)
            dark_subtracted = ccdproc.subtract_dark(bias_subtracted, dark, exposure_time='exposu
            reduced_image = ccdproc.flat_correct(dark_subtracted, flat)
            root, ext = os.path.splitext(fname)
            ofname = os.path.join(destination_dir_3184_3, root) + '_reduced.fits'
            reduced_image.write(ofname)
```

```
WARNING: FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa. [astropy.wcs.wcs]
WARNING:astropy:FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa.
```

```
In [11]: #loading the calibration object NGC6720
         ccd_ngc6720 = fits.open('../../assignments/Session5/NGC6720/ngc6720_h_30_1.fits')
         ccd_data6720 = ccd_ngc6720[0].data
         print(ccd_data6720.mean())

         #fig,axes = plt.subplots(figsize=(8,8))
         #Bins = 3000
         #axes.hist(ccd_data6720.ravel(),Bins)
         #axes.set_xlim(0,700)
         #plt.show()
```

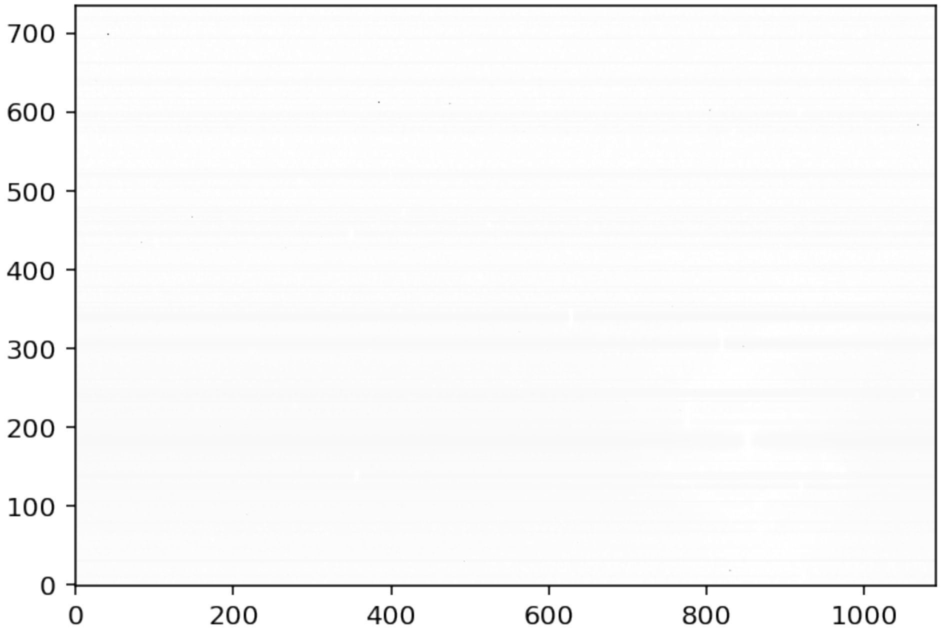
```
290.118368271
```

```
        ---------------------------------------------------------------------------

        NameError                                 Traceback (most recent call last)

        <ipython-input-11-ddbc50053616> in <module>()
          3 ccd_data6720 = ccd_ngc6720[0].data
          4 print(ccd_data6720.mean())
    ----> 5 print(median(ccd_data6720))
          6 #fig,axes = plt.subplots(figsize=(8,8))
          7 #Bins = 3000


        NameError: name 'median' is not defined
```

```
In [20]: #Scikit learn module used
         from sklearn.preprocessing import normalize
         #?normalize
         normed_data = normalize(mean_r_data, axis=1, norm='max')
         print(normed_data)
```

```
[[ 0.27124596  0.30585245  0.52528778 ...,  0.64014297  0.54018972
   0.50692661]
 [ 0.20176851  0.34586829  0.47074685 ...,  0.71502778  0.63944739
   0.41248747]
 [ 0.21111121  0.35209332  0.55122093 ...,  0.64131794  0.73883172
   0.61950779]
 ...,
 [ 0.31939159  0.31181076  0.55997743 ...,  0.85751808  0.67299692
   0.67654348]
 [ 0.41664443  0.35075512  0.64554666 ...,  0.64457353  0.88232864
   0.78651874]
 [ 0.40494603  0.3099271   0.76234955 ...,  0.58667189  0.53818709
   0.62189351]]
```

Out[20]:



```
In [4]: #Normalize the rows
        def normalize_rows(a):
            row_sums = a.sum(axis=1)[:,np.newaxis]
            return a/row_sums

        mean_normed_r_data = normalize_rows(mean_r_data)
        print(mean_normed_r_data)
```

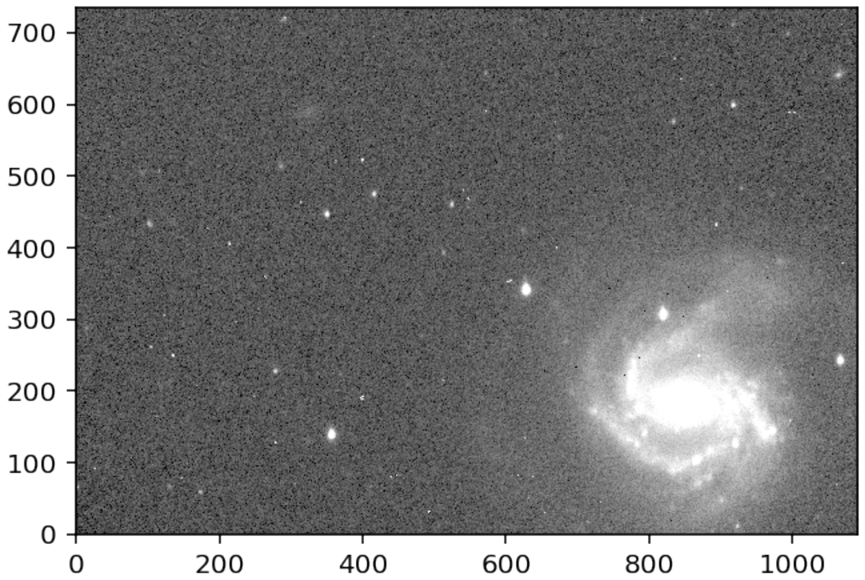
```
[[ 0.0003847   0.00043378  0.00074499 ...,  0.00090789  0.00076613
   0.00071895]
```

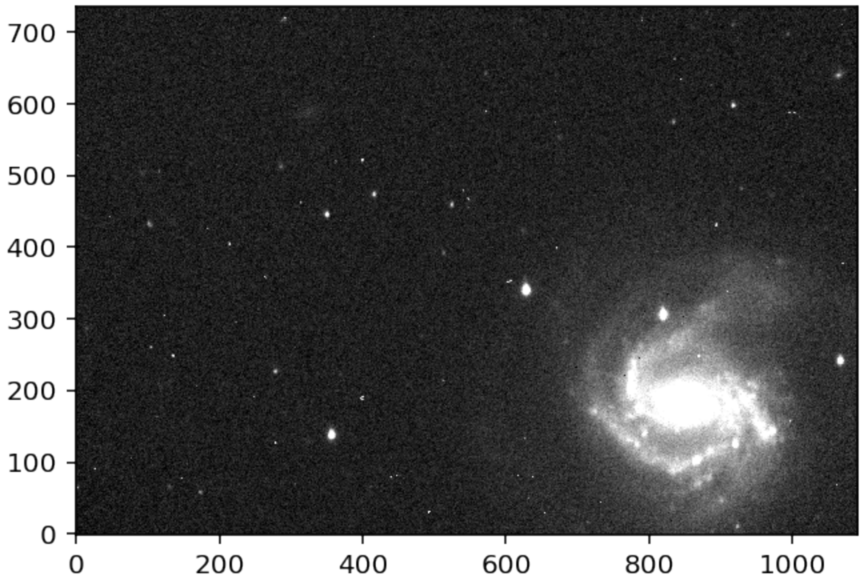
```
 [ 0.0002778   0.0004762   0.00064814 ...,  0.00098447  0.00088041
   0.00056792]
 [ 0.00030275  0.00050492  0.00079048 ...,  0.00091969  0.00105953
   0.00088841]
 ...,
 [ 0.00045213  0.0004414   0.00079271 ...,  0.00121391  0.0009527
   0.00095772]
 [ 0.00053014  0.0004463   0.0008214  ...,  0.00082016  0.00112268
   0.00100077]
 [ 0.00057377  0.00043914  0.00108019 ...,  0.00083127  0.00076257
   0.00088117]]
```

In [9]:
```
#Normalizing the R-band image
#matplotlib normalization
mean_r_ccd = fits.open('../../assignments/Session5/M61/calib_data/m61_mean_r.fits')
#print(mean_r_ccd[0].data)
mean_r_data = mean_r_ccd[0].data
mean_r_header = mean_r_ccd[0].header

#create a ImageNormalize object
norm = ImageNormalize(mean_r_data, interval=MinMaxInterval(),stretch=SqrtStretch())

fig,ax = plt.subplots(figsize=(8,8), nrows=2)
vmin, vmax = np.percentile(mean_r_data,[1,99])
implot = ax[0].imshow(mean_r_data, cmap='gray', origin='lower',vmin=vmin, vmax=vmax, nor
implot1 = ax[1].imshow(mean_r_data, cmap='gray', origin='lower', vmin=vmin, vmax=vmax)
plt.show()
print(type(implot))
```

```
/ext/anaconda3/lib/python3.5/site-packages/astropy/visualization/stretch.py:112: RuntimeWarning:
  np.sqrt(values, out=values)
/ext/anaconda3/lib/python3.5/site-packages/matplotlib/colors.py:504: RuntimeWarning: invalid val
  xa[xa < 0] = -1
```

Out[9]:





```
<class 'matplotlib.image.AxesImage'>
```

```
---------------------------------------------------------------------------

AttributeError                            Traceback (most recent call last)

<ipython-input-9-3765cbc575b4> in <module>()
 15 plt.show()
```

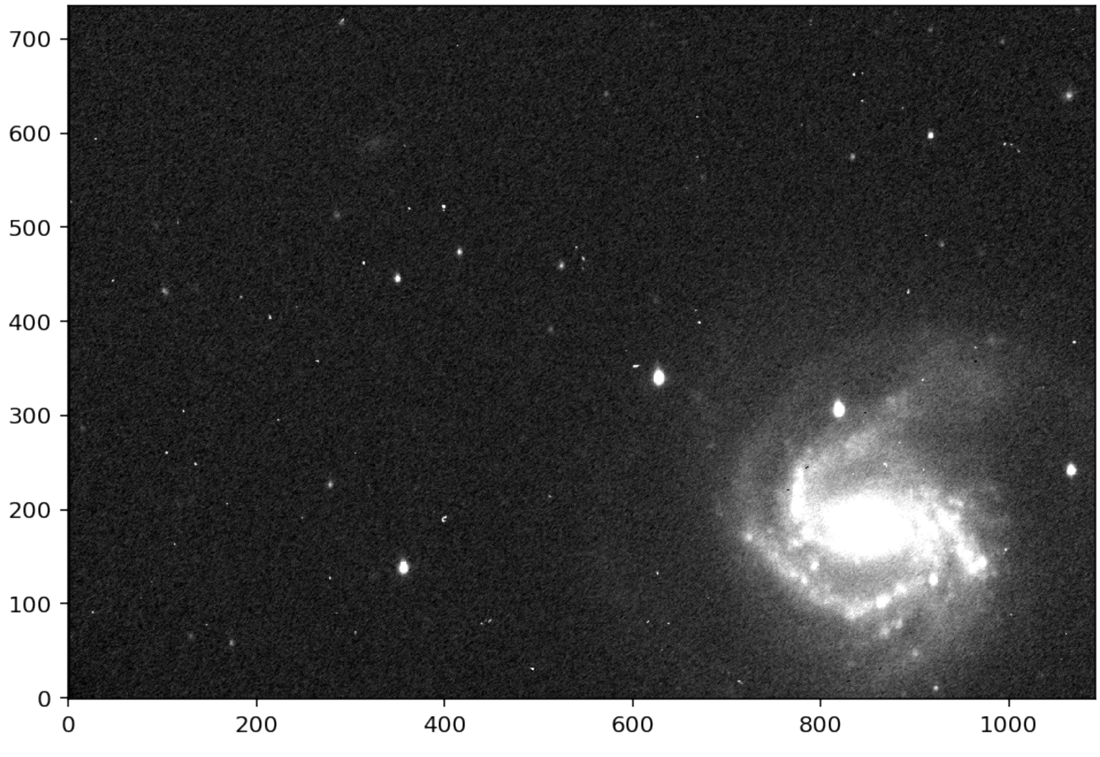
```
     16 print(type(implot))
---> 17 print(implot.get_data())


    AttributeError: 'AxesImage' object has no attribute 'get_data'
```

```
In [74]: #combining the images for R-band and h-alpha
         #destination_dir_align = '../../assignments/Session5/M61/calib_aligned'
         #calib_images = ccdproc.ImageFileCollection(location=destination_dir_align,keywords=['e

         #R-band
         #hdu_r_1 = fits.open('../../assignments/Session5/M61/calib_aligned/aligned_m61_r_240_1_
         #ccd_r_data_1 = fits.getdata('../../assignments/Session5/M61/calib_aligned/aligned_m61_
         #ccd_r_data_2 = fits.getdata('../../assignments/Session5/M61/calib_aligned/aligned_m61_

         """The mean combined r-band and h-alpha images are loaded.
            The r-band image is normalised using the maximum value of the counts in the image.""
         #os.listdir(destination_dir)
         #create a function to normalize each column of the data
         def normalize_data(data):
             rows,cols = data.shape
             for j in range(cols):
                 data[:,j] /= abs(data[:,j]).max()
                 j=j+1
             new_data = data
             return new_data
         mean_r_ccd = fits.open('../../assignments/Session5/M61/calib_data/m61_mean_r.fits')
         #print(mean_r_ccd[0].data)
         mean_r_data = mean_r_ccd[0].data
         mean_r_header = mean_r_ccd[0].header

         print(mean_r_data)
         #filepath = os.path.join(destination_dir,'m61_mean_r_norm.fits')
         #mean_r_ccd1 = fits.open('../../assignments/Session5/M61/calib_data/m61_mean_r_norm.fit
         #print(mean_r_ccd1[0].data)
         #mean_r_data1 = fits.getdata(filepath)

         #mean_r_norm = normalize_data(mean_r_data)
         #print(mean_r_norm)
         #hdu = fits.PrimaryHDU(data=mean_r_norm, header=mean_r_header)
         #hdu.writeto('m61_mean_r_norm.fits',clobber=True)

         fig,axis = plt.subplots(figsize=(8,8))
         Nbins=800
         #axis.hist(mean_r_data.ravel(),Nbins)
         #axis.set_xlim(-2,5)
         #axis.set_ylim(0,5000)
```

```
#implot = axis[0].imshow(mean_r_norm,cmap='gray',origin='lower')
vmin, vmax = np.percentile(mean_r_data,[1,99])
implot1 = axis.imshow(mean_r_data,cmap='gray',origin='lower',vmin=vmin,vmax=vmax)
axis.grid(False)
plt.show()
```

```
[[  77.41653442   87.29360199  149.92282104 ...,  182.70373535
   154.17599487  144.68234253]
 [  55.77303314   95.60522461  130.12426758 ...,  197.64862061
   176.75662231  114.02015686]
 [  60.9186554   101.60072327  159.06137085 ...,  185.05993652
   213.19869995  178.76635742]
 ...,
 [  85.8229599    83.78593445  150.47021484 ...,  230.42166138  180.8394165
   181.79240417]
 [ 101.39308167   85.35849762  157.09790039 ...,  156.86108398  214.7203064
   191.40435791]
 [ 109.49563599   83.80293274  206.13598633 ...,  158.6335144   145.5234375
   168.1572876 ]]
```

Out[74]:


700
600
500
400
300
200
100
0
0
200
400
600
800
1000