ZTF24aahgqwk in NGC 3443

Observation Notes

Typically a session has 60 30-second exposures in each of r' and g', but starting with 2024-04-21, there are 120 of g', because g' images were getting fainter.

ZTF24aahgwk Observation Log

In [ ]:
```
import os
import numpy as np
from astropy import units as u
from astropy.nddata import CCDData
# from astropy.io import fits
from ccdproc import ImageFileCollection, combine, subtract_dark, flat_correct # Combiner
import matplotlib.pyplot as plt
%matplotlib inline

# filters

filters = ['g', 'r']
filter_full_names = ["Sloan g'", "Sloan r'"]

# exposure durations

light_exposure = 30 * u.second
flat_exposure = 0.1 * u.second
dark_exposure = light_exposure  # our method presumes this equality
bias_exposure = flat_exposure  # our method presumes this equality
```

Combine the Calibration Images into Masters

Calibration Images

The calibration images are in ~/2024 Sessions/2024-04-12/. In turn, ~/2024 Sessions is actually a soft link to /Volumes/Astronomy Data/2024 Sessions/2024 Sessions.

In [ ]:
```
# calibration directory

calibration_date = '2024-04-12'

calibration_directory = os.path.join(os.path.expanduser('~'), '2024 Sessions', calibration_date)

# subdirectory for the 30-second darks

dark_directory = os.path.join(calibration_directory, 'dark')

# subdirectories for the 0.1-second g and r flats

flat_directories_by_filter = {filter:os.path.join(calibration_directory, 'flat', filter)
                              for filter in filters}

# subdirectory for the biases (TheSky Professional Edition may indicate that these are 0.1-second darks)

bias_directory = os.path.join(calibration_directory, 'bias')

# Trimmed image reader utility (needed because our images have a final row of zeros and four columns of N

def delete_last_rows_and_columns(arr, rows_to_delete, columns_to_delete):
    row_count = np.shape(arr)[0]
    arr = np.delete(arr, slice(row_count - rows_to_delete, row_count), 0)
    column_count = np.shape(arr)[1]
    arr = np.delete(arr, slice(column_count - columns_to_delete, column_count), 1)
    return arr
```

```
def trimmed_image_reader(file):
    img = CCDData.read(file, unit=u.adu)
    data = img.data
    trimmed_data = delete_last_rows_and_columns(data, 1, 4)
    img.data = trimmed_data
    return img

# darks

dark_files = ImageFileCollection(dark_directory).files_filtered(include_path='True')
darks = [trimmed_image_reader(file) for file in dark_files]

# flats by filter

flat_files_by_filter = {filter:ImageFileCollection(flat_directory).files_filtered(include_path='True')
                        for filter, flat_directory in flat_directories_by_filter.items()}
flats_by_filter = {filter:[trimmed_image_reader(file) for file in flat_files]
                   for filter, flat_files in flat_files_by_filter.items()}

# biases

bias_files = ImageFileCollection(bias_directory).files_filtered(include_path='True')
biases = [trimmed_image_reader(file) for file in bias_files]

# Combine darks, flats, and biases

method = 'median'  # alternatively, the method can be 'average'

master_dark = combine(darks, method=method)
master_flats_by_filter = {filter:combine(flats, method=method)
                          for filter, flats in flats_by_filter.items()}
master_bias = combine(biases, method=method)

# Perform dark subtraction of the master flats

master_flats_subtracted_by_filter = {filter:subtract_dark(master_flat,
                                                          master_bias,
                                                          data_exposure=flat_exposure,
                                                          dark_exposure=bias_exposure,
```

```
                                                scale=False)
                                    for filter, master_flat in master_flats_by_filter.items()}
```

Load, Calibrate, Align, and Stack Lights

The lights we are currently examining are in ~/2024 Sessions/2024-04-17/.

As a next step before adding the entire observation series, I will be adding

2024-04-10, 2024-04-11, and 2024-04-13

and

2024-04-21 and 2024-04-22.

In [ ]:
```
observation_date = '2024-04-17'

observation_directory = os.path.join(os.path.expanduser('~'), '2024 Sessions', observation_date)

# subdirectories for the 30-second g and r lights

light_directories_by_filter = {filter:os.path.join(observation_directory, filter) for filter in filters}

# lights by filter

light_files_by_filter = {filter:ImageFileCollection(light_directory).files_filtered(include_path='True')
                         for filter, light_directory in light_directories_by_filter.items()}
lights_by_filter = {filter:[trimmed_image_reader(file) for file in light_files]
                    for filter, light_files in light_files_by_filter.items()}

lights_subtracted_by_filter = {filter:[subtract_dark(light,
                                                     master_dark,
                                                     data_exposure=light_exposure,
                                                     dark_exposure=dark_exposure,
                                                     scale=False) for light in lights]
                               for filter, lights in lights_by_filter.items()}

# Perform flat division

lights_calibrated_by_filter = {filter:[flat_correct(light, master_flats_subtracted_by_filter[filter])
                                       for light in lights]
                               for filter, lights in lights_subtracted_by_filter.items()}
```

In [ ]:
```
# in this phase of the analysis, the aligned directories are written to not read from

# first create the aligned directories

aligned_directories_by_filter = {filter:os.path.join(light_directory, 'aligned')
                                 for filter, light_directory in light_directories_by_filter.items()}

for aligned_directory in aligned_directories_by_filter.values():
    if not os.path.exists(aligned_directory):
        os.makedirs(aligned_directory)
```

In [ ]:
```
master_flats_subtracted_by_filter['r']
```

In [ ]:
```
master_flats_by_filter['r']
```

In [ ]:
```
master_bias
```

In [ ]:
```

```