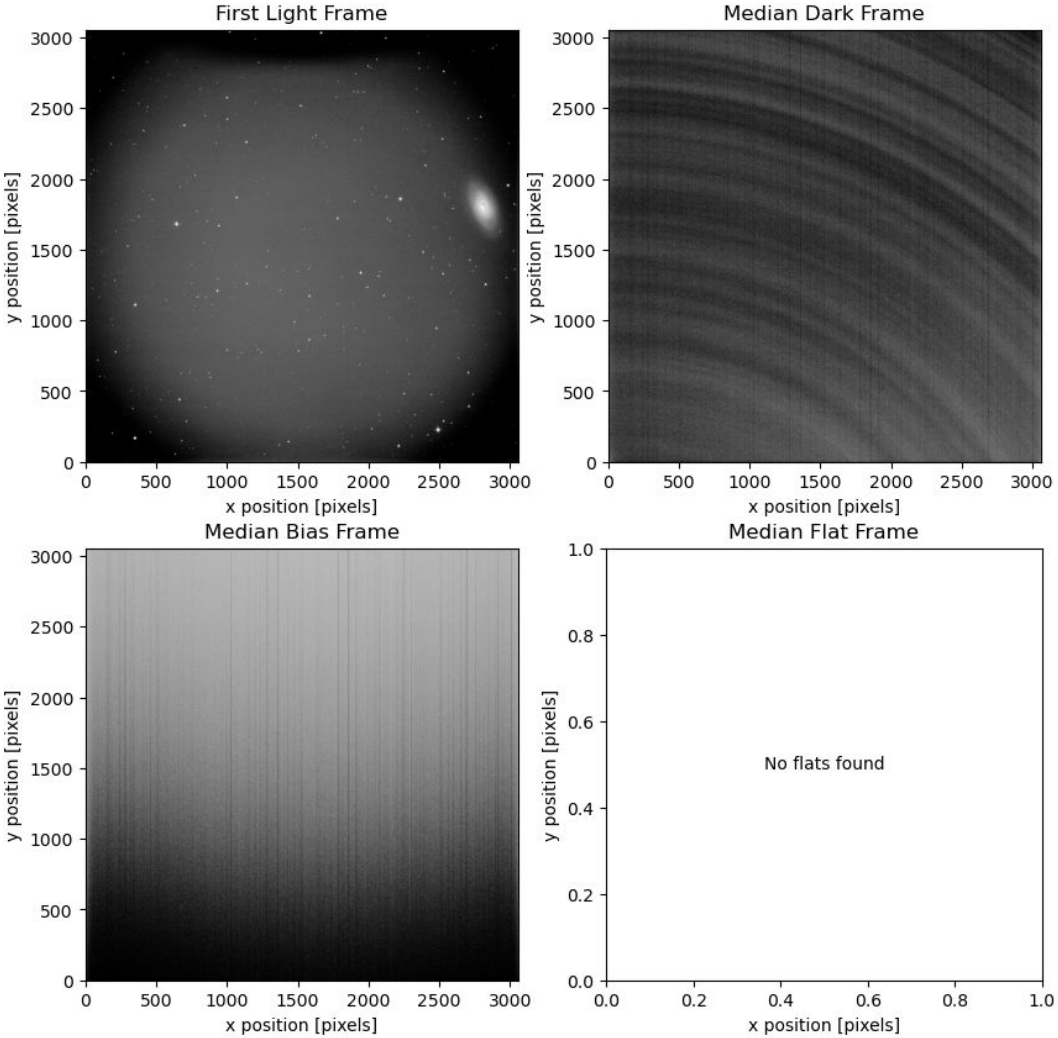
# NEO Follow-Up

Data Reduction and Star Detection

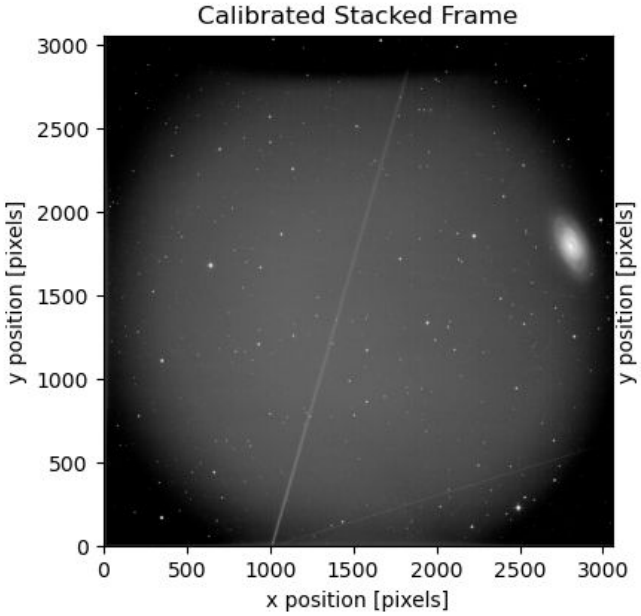
# Data Reduction Steps

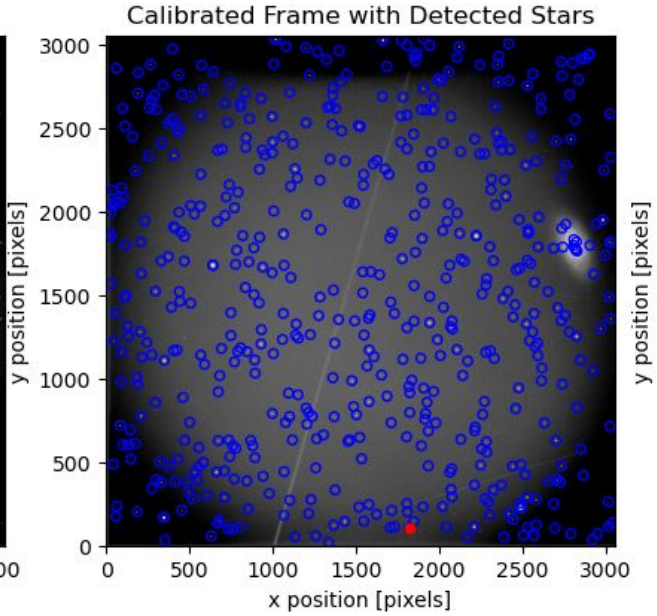
1. Calculate the **median bias/dark/flat** frames
    a. Load in the fits images
    b. np.median
2. Align the light frames, then calculate the **median light** frame
    a. import astroalign as aa
    b. Select the reference frame
    c. Match all other frames to the reference frame
    d. np.median
3. **calibrated_image** = (median_light - median_bias - median_dark) / median_flat

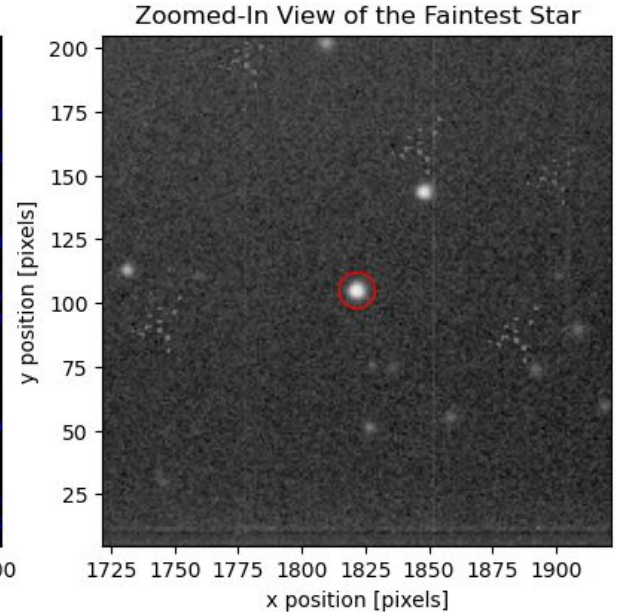

First Light Frame
Median Dark Frame
Median Bias Frame
Median Flat Frame
No flats found
x position [pixels]
y position [pixels]
0
500
1000
1500
2000
2500
3000
0.0
0.2
0.4
0.6
0.8
1.0

# Star Detection

1. **Find all stars** in the calibrated image
    a. from photutils import DAOStarFinder
    b. daofind = DAOStarFinder(fwhm=detection_fwhm, threshold=detection_sigma_threshold*std)
2. **Plate solve** the calibrated image
    a. ast = AstrometryNet()
    b. ast.api_key = astrometry_api_key
    c. wcs_header = ast.solve_from_image(fit_image_path, force_image_upload=True)
3. **SIMBAD Query**
    a. Query every star found in step 1
    b. Get V mag info
    c. Return the faintest star




(<Row index=0>

| MAIN_ID | RA | DEC | RA_PREC | DEC_PREC | COO_ERR_MAJA | COO_ERR_MINA | COO_ERR_ANGLE | COO_QUAL | COO_WAVELENGTH | COO_BIBCODE | FLUX_V |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | "h:m:s" | "d:m:s" | | | mas | mas | deg | | | | mag |
| object | str13 | str13 | int16 | int16 | float32 | float32 | int16 | str1 | str1 | object | float32 |
| SDSS J125433.37+212336.2 | 12 54 33.3841 | +21 23 36.248 | 14 | 14 | 0.670 | 0.508 | 90 | A | 0 | 2020yCat.1350....0G | 20.79 |