Import Statements

In [2]:
```python
import custom_cmap
import os
import sys
from PIL import Image
from IPython.display import display
import matplotlib.pyplot as plt
import matplotlib as mpl
import pandas as pd
from moria import reduce
from astroquery.ogle import Ogle
from astropy.coordinates import SkyCoord
from astropy import units as u
from pathlib import Path
from astropy.visualization import PercentileInterval
from astropy.coordinates import SkyCoord
from astropy import units as u
from re import A



from astropy.io import fits
from astropy.visualization import LogStretch, ImageNormalize
import plotly.express as px
import numpy as np
from astropy.visualization import ImageNormalize, AsinhStretch, SqrtStretch, LogStretch, PowerStretch

plt.rcParams.update({'font.size':25})
plt.rc('text', usetex=True)
%config InlineBackend.figure_format = 'retina'
%load_ext autoreload
%autoreload 2
```

```
/Users/tmbhadra/miniforge3/lib/python3.12/site-packages/astroquery/ogle/__init__.py:40: UserWarning: Experi
mental: OGLE has not yet been refactored to have its API match the rest of astroquery.
  warnings.warn("Experimental: OGLE has not yet been refactored to have its "
```

Understanding the main directory.

The directory structure for your data analysis should be organized as follows.

00.DATA
01.XYM
02.CMD
03.LOC_TRANS
04.PSF_EXTRACT
05.COORD_TRANS(OPTIONAL)
06.FIT
07.CALIBRATION
This directory structure has been set up in "MORIA/data".

All necessary scripts are included within the corresponding folders under MORIA/data. Therefore, the simplest way to run MORIA on your target is to copy all eight folders from MORIA/data to the location where you intend to perform your analysis. Begin by placing your exposures in data/00.DATA.

This notebook involves manual steps to calibrate HST with OGLE-III.

NOTE: chmod +x program.src is a useful command to use whenever permissions are denied for a script

In [3]:
```
#This is the directory where you are processing your data. This does not point at MORIA.
directory = os.getcwd()
```

Step 0

We assume you ran the output_stacks.ipynbm, cmd_diagram.ipynb, creating_psf.ipynb, fitting_psfs.ipynb (in that order) notebooks correctly

Step 1: Enter coordinates of your target

In [4]:
```
ogle_ra = "17:51:31.86"
ogle_dec = "-29:33:55.0"

ogle_coord = SkyCoord(ogle_ra, ogle_dec, unit = (u.deg, u.deg))
ogle_ra_deg = np.float64(ogle_coord.ra)
ogle_dec_deg = np.float64(ogle_coord.dec)

ogle_band = "I"
```

The goal here is to calibrate the HST photometry to the OGLE-III database. To obtain location of your event you will first need to obtain the OGLE-III field number, chip number, and pixel coordinates at the OGLE web page using the “OGLE Field Finder” under the “Sky Coverage”. MORIA automates this with the scripts below.

Once MORIA does that, it chooses the appropriate OGLE-III catalog file at http://www.astrouw.edu.pl/ogle/ogle3/maps/blg/maps/. It will download the appropriate photometry map for your event from here.

For example, consider OGLE-2012-BLG-0563: the OGLE-III catalog file is blg226.7.map, and the OGLE reference image corresponding to this photometry catalog file is blg226.I.7.fts, which can be obtained from http://www.astrouw.edu.pl/ogle/ogle3/maps/blg/ref_images/ .

The routines in this module compare the HST photometry to the OGLE photometry in order to enable the HST photometry to be calibrated to OGLE. Special efforts are required to avoid blending effects, where several HST stars will be merged together to make an object that is considered to be a single star in the lower angular resolution OGLE images. Also, PSF-fitting magnitudes are dependent on the details of the PSF model. T

hus, we will need to re-reduce the photometry of the selected HST calibration candidate stars with the same PSF used for multi-star fitting.

This is accomplished with a program that is similar to the multi-star fitting program used in module 06.TRI_FIT_final. It only does single star fits, but it does them for all the HST calibration candidate stars that were identified in module 02.CMD_final.

Step 2: Download the OGLE map and reference image

In [5]:
```
ogle_field_number, ogle_chip_number = reduce.get_chip_number(ogle_ra_deg, ogle_dec_deg)

reduce.download_ogle_map_and_reference(
    directory=directory,
    ogle_field_number=ogle_field_number,
    ogle_chip_number=ogle_chip_number,
    ogle_band=ogle_band,
    destination_subdir="07.CALIBRATION",
    overwrite=False,
)
```

```
OGLE field candidates: [{'field_name': 'BLG194.1', 'field_prefix': 'BLG', 'field_number': 194, 'chip_number': 1, 'x': 2066.5, 'y': 1538.96}]
```

Out[5]:
```
{'map_url': 'http://www.astrouw.edu.pl/ogle/ogle3/maps/blg/maps/blg194.1.map.bz2',
 'ref_url': 'http://www.astrouw.edu.pl/ogle/ogle3/maps/blg/ref_images/blg194.I.1.fts.bz2',
 'map_path': '/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION/BLG194.1.map',
 'ref_path': '/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION/blg194.I.1.fts'}
```

Step 3: Designing calibration input

When calibrating the module below, we require running the program psf_star_mags_mcmc.xOg using the script run_psf_star_Imags_mcmc.src and the input file IN_psf_star_mags_mcmc_I. We will first design the input for this script.

The inputs in this file are:

1. The PSF file name (already entered)
2. The input file name tag (already entered)
3. The output file name tag (already entered)
4. The number of Markov Chain steps (e.g. 100,000)
5. The maximum size of MCMC coordinate steps in pixels and the error bar fudge factor (e.g. 0.02)
6. The maximum distance in x and y from the star center for pixels to be included in the fit, and the $\chi 2$ threshold to define an outlier pixel. (e.g. 16)
7. The sky model to use: 0 for annulus, 1 for fit sky on the same pixels as the PSF peak. The annulus sky model is needed for determining the PSF model, but it is more sensitive to blending than the fit sky model. (e.g. 1.0)

8. A list of star numbers to produce PIX_SHOW files for, ending with a 0 to exit the program. These PIX_SHOW are the same as the ones produced in module 06, and they are useful to tell you how well each star was fit. But, they are large files that are time consuming to produce, and it is unlikely that you'll want them for every star. (e.g. 10)

In [6]:
```
# Default priors you can use:
mcmc_defaults = {
    "markov_chain_steps": 100000,
    "maximum_size_mcmc": 0.02,
    "fudge": 1.0,
    "maximum_distance_x": 2.5,
    "maximum_distance_y": 2.5,
    "chi2cut": 16,
    "sky_model": 1,
    "star_numbers_pix_show": 10,
}
```

In [7]:
```
reduce.calibration_input_file_one(directory)
```

```
File 'IN.psf_star_mags_mcmc_I' successfully created.
```

Step 4: Recreate calibration input for V magnitude

Repeat this procedure for the V band using script run_psf_star_Vmags_mcmc.src, which uses the input file IN_psf_star_mags_mcmc_V.

In [8]:
```
reduce.calibration_input_file_two(directory)
```

```
File 'IN.psf_star_mags_mcmc_V' successfully created.
```

Step 5: Create new calibration map

When you finish running this step, you should see two new files in 07.CALIBRATION:

1. MATCHUP.F814W_cal.XYM
2. MATCHUP.F814W_cal_only.XYM

In [10]:
```
reduce.calibration_new_matchup(directory)
```

```
/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION
/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION/run_psf_star_Imags_mcmc.src
Finished run_psf_star_Imags_mcmc
/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION
/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION/run_psf_star_Vmags_mcmc.src
Finished run_psf_star_Vmags_mcmc
/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION
/Users/tmbhadra/Documents/Work/NASA/test_moria/07.CALIBRATION/run_cal_star_num_2_MATCHUP.src
Finished cal_star_num_2_MATCHUP
Finished cal_star_num_2_MATCHUP
```

Step 6

This is a manual step. Notice that the 07.CALIBRATION Setup has four IN.* files in it already. The first is IN.cal_star_num_2_MATCHUP which is used to re-create matchup files used for calibration. Then we have IN.fit_HST_Iogle_col_1 and IN.fit_HST_Iogle_col_2 used in Step 6 below. We also have IN.VI_HST_ogle_man_match4_backup.

As of now, you need to manually edit IN.VI_HST_ogle_man_match4_backup. This step instructs you on how to edit that file.

An important part of this input file is a list of stars that have been matched in the OGLE reference image and the HST image, outputq_F814W.fits. At this point, navigate to the 07.CALIBRATION folder using your preferred terminal and run 'python starlist2reg.py'. You will be prompted to enter the name of your starlist (e.g. 'blg194.1.map') and the x/y position columns in the starlist. If you don't know what these are, enter the number 3 for 'X column' and 4 for 'Y column'.

Then, open 'outputq_F814W.fits' in DS9. 'starlist2reg.py' created a '.reg' file that you can load to DS9. These will overlay the positions from the OGLE map on your HST images. Choose ones that look best by eye and edit 'IN.VI_HST_ogle_man_match4_backup.'

In 'IN.VI_HST_ogle_man_match4', edit the first row to your matching radius in arcseconds; enter the coordinates for your event in the second row; change your map number in seven. Then, you need to enter information about the matching calibrataed stars. The first two entries are the HST x and y coordinates (found in the MATCHUP file), then the OGLE x and y coordinates (found in the starlist), followed by the HST mangitudes and then the OGLE magnitudes for those stars. Loading the '.reg' file into DS9 makes this process quicker. However, this is the only part of the pipeline that has been left manual. Once the 'IN.VI_HST_ogle_man_match4' file has been created, proceed with the notebook.

In [13]:
```
reduce.calibration_hst_ogle_match(directory)
```

Step 7

The final calibration program is used to select constraints to put on the parameters in the VI_HST_ogle_Cal_matches4.dat file to select the calibration stars and calibrate the HST photometry. This is done program fit_HST_IV_ogle_col.xOg.

This program calculates both I band and V band offsets between the calibrated OGLE photometry and the HST photometry, as well as 2-color fits to calibrate both the HST I and V bands to the OGLE I and V band.

The results of these calibration attempts have been routed to the run_fit_HST_IV_ogle_col_1.log and run_fit_HST_IV_ogle_col_2.log files.

If this step fails, you should open ``VI_HST_ogle_Cal_matches4.dat" and remove the spacing between the columns with headers Vo-Vhfs and Ig_c2Vmx

In [ ]:
```
reduce.fit_calibration(directory)
```

You can now open the log_files present in 07.CALIBRATION/log_files to note down the OGLE-III calibrated HST magnitudes

In [ ]:
```

```