ColorLCAPlot

August 9, 2023

0.0.1 Visualization of the photometry chains

This notebook has some simple visualizations of the flux+LCA chains derived for each TNO in this data release.

The key idea is that each chain has several realizations of the 5-dimensional (f_g, f_r, f_i, f_z, A) parameter space, where each f_b is the mean flux in band b (normalized to 30 au from the observer and the Sun, and with zeropoint defined to be 30 mag) and A the lightcurve amplitude. These can be projected into different parameters, for example: At band b, the absolute magnitude (at 1 au from the Sun and observer) is

$$H_b = -2.5\log_{10}(f_b) + 30 - 5\log_{10}(30\,\mathrm{au}\cdot 30\,\mathrm{au}).$$

The color between band b and band b' is

$$b - b' = -2.5\log_{10}(f_b/f_{b'}).$$

The peak-to-peak (magnitude space) lightcurve amplitude is

$$\Delta m = 2.5\log_{10}\left(\frac{1+A}{1-A}\right)$$

Let's start with a few package imports, and reproduce the figures in the paper

[1]:
```
import numpy as np
import astropy.table as tb
import matplotlib.pyplot as pl
```

Let's start with the figures for 2016 SD_{106}:

[7]:
```
data_sd = tb.Table.read('../fluxes/2016SD106.hdf5', 'data')
chain_sd  = tb.Table.read('../fluxes/2016SD106.hdf5', 'samples')
```

What is actually inside these tables?

For the flux measurements, we have, for each observation (and each object): - `EXPNUM`: exposure number - `BAND`: band - `PHASE`: phase angle of the measurement (in radians) - `PHASEPA`: position angle of the phase - `RANGE`: topocentric distance (in AU) - `SOLARD`: heliocentric distance (in AU) - `TDB`: time of the observation (in years after J2000.0, TDB scale) - `RA`: right ascension (degrees) - `DEC`: declination (degrees) - `FLUX_30AU`: flux scaled to 30 au - `FLUXERR_30AU`: uncertainity in the flux, also scaled to 30 au - `CLIPPED`: boolean variable that indicates if the observation was clipped from in the fitting procedure - `MPC`: MPC identifier for the object

[8]:
```
data_sd.show_in_notebook()
```

[8]: <IPython.core.display.HTML object>

And for the chains: - `flux_grizY`: flux in the *grizY* bands. This processing did not use the *Y* band data, so these are all 0. - `lca`: the lightcurve amplitude - `logP`: log of the probability of this value (only needed for bookkeeping purposes)

[9]:
```
chain_sd.show_in_notebook()
```

[9]: <IPython.core.display.HTML object>

Now onto the visualization - we will compare the colors of this object to those measured by Chen et al (2022)

[6]:
```
#subplot_mosaic allows some really nice plots
fig, ax = pl.subplot_mosaic(
    '''
    AAAA
    BCDD
    ''')

### flux measurements
for i in ['g', 'r', 'i', 'z']:
    mean = np.mean(chain_sd[f'flux_{i}'])
    a = data_sd[data_sd['BAND'] == i]
    ax['A'].errorbar(2000+a['TDB'], a['FLUX_30AU']/mean, yerr=a['FLUXERR_30AU']/
 ↪mean, fmt='.', color=BAND_COLORS[i],
                    label=r'${' + i + '}$')

ax['A'].set_xlim(2013,2019)
ax['A'].set_xlabel('Observing date (years)')
ax['A'].set_ylabel('Relative flux')
ax['A'].legend(ncol=4, loc=3,columnspacing=0.5,bbox_to_anchor=(0.2, 0.05))
ax['A'].axhline(1, linestyle='--', alpha=0.4)
ax['A'].set_ylim(0.,1.5)
ax['A'].grid()

### absolute magnitude
Hr = -2.5 * np.log10(chain_sd['flux_r']) - 10 * np.log10(30) + 30
ax['B'].hist(Hr, bins=100, histtype='step', density=False, color='k', alpha=0.6)
ax['B'].set_xlabel(r'$H_r$ (mag)')
ax['B'].set_ylabel('Samples')
ax['B'].set_ylim(0,120)
ax['B'].set_yticks([0,30,60,90,120])#,['','','','','']
ax['B'].grid()

### lightcurve amplitude - flux space
```

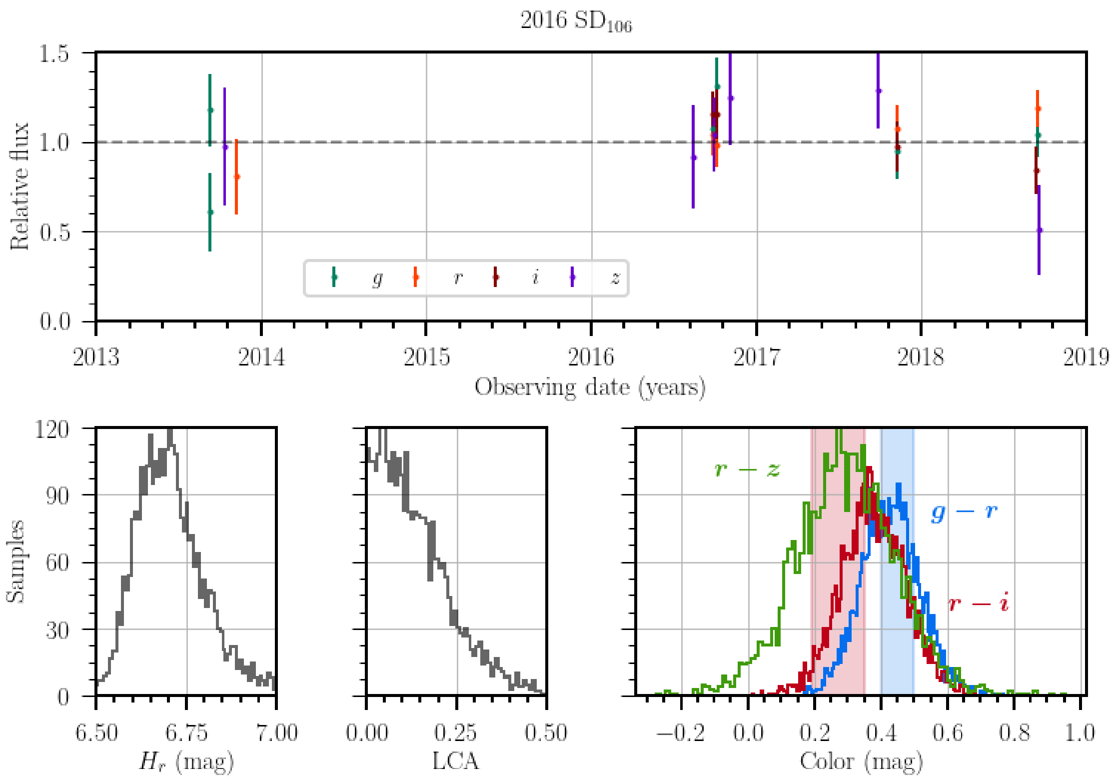
```python
ax['C'].hist(chain_sd['lca'], bins=100, histtype='step', density = False, color='k', alpha=0.6)
ax['C'].set_xlabel(r'LCA')
ax['C'].set_yticks([0,30,60,90,120],['','','','',''])
ax['C'].set_ylim(0,120)
ax['C'].grid()
ax['C'].set_xlim(0,0.5)
ax['C'].set_xticks([0,  0.25, 0.5, ])#, [0,25,50])

### colors

ax['D'].hist(-2.5 * np.log10(chain_sd['flux_g']/chain_sd['flux_r']), histtype='step', bins=100, density=False, label=r'$g-r$')
ax['D'].hist(-2.5 * np.log10(chain_sd['flux_r']/chain_sd['flux_i']), histtype='step', bins=100, density=False, label=r'$r-i$')
ax['D'].hist(-2.5 * np.log10(chain_sd['flux_r']/chain_sd['flux_z']), histtype='step', bins=100, density=False, label=r'$r-z$')

# these correspond to measurements from Chen et al (2022)
ax['D'].axvspan(0.4, 0.5, alpha=0.2, color='xkcd:cerulean blue')
ax['D'].axvspan(0.19, 0.35, alpha=0.2, color='xkcd:scarlet')
ax['D'].set_xlabel('Color (mag)')
ax['D'].set_ylim(0,120)
ax['D'].set_yticks([0,30,60,90,120],['','','','',''])
ax['D'].grid()
ax['D'].text(0.55, 80, r'$\boldsymbol{g-r}$', fontsize=18, color='xkcd:cerulean blue')
ax['D'].text(0.6, 40, r'$\boldsymbol{r-i}$', fontsize=18, color='xkcd:scarlet')
ax['D'].text(-0.1, 100, r'$\boldsymbol{r-z}$', fontsize=18, color='xkcd:grass green')
ax['B'].set_xticks([6.5,6.75, 7])
ax['B'].set_xlim(6.5,7)
ax['D'].set_xticks([-0.2, 0, 0.2,0.4, 0.6, 0.8, 1.0,])


fig.subplots_adjust(hspace=0.4, wspace=0.5)
fig.set_size_inches(12,8)
pl.suptitle(r'2016 SD$_{106}$',y=0.93)
pl.show()
```



Now, for (612620) 2003 SQ_{317}. We'll be comparing with the $\Delta m = 0.85$ measured by Lacerda et al (2014).

Since

$$\Delta m = 2.5 \log_{10} \left(\frac{1+A}{1-A} \right) \implies A = \frac{10^{0.4\Delta m} - 1}{10^{0.4\Delta m} + 1},$$

[13]:

```
Delta_m_Lacerda = 0.85
A_Lacerda = (10**(0.4 * Delta_m_Lacerda) - 1)/(10**(0.4 * Delta_m_Lacerda) + 1)

print(f'A = {A_Lacerda:.4f}')
```

```
A = 0.3726
```

Load the data and plot

[15]:

```
data_sq = tb.Table.read('../fluxes/2003SQ317.hdf5', 'data')
chain_sq = tb.Table.read('../fluxes/2003SQ317.hdf5', 'samples')
```

[17]:

```
fig, ax = pl.subplot_mosaic(
    '''
    AAAA
    BCDD
    ''')
```

```
### flux measurements

for i in ['g', 'r', 'i', 'z']:
    mean = np.mean(chain_sq[f'flux_{i}'])
    a = data_sq[data_sq['BAND'] == i]
    ax['A'].errorbar(2000+a['TDB'], a['FLUX_30AU']/mean, yerr=a['FLUXERR_30AU']/
 ↪mean, fmt='.', color=BAND_COLORS[i],
                     label=r'${' + i + '}$')

ax['A'].set_xlim(2013,2019)
ax['A'].set_ylim(0,2)
ax['A'].set_xlabel('Observing date (years)',)
ax['A'].set_ylabel('Relative flux', )
ax['A'].legend(ncol=5, loc=3,columnspacing=0.5)
ax['A'].axhline(1, linestyle='--', alpha=0.4)

#Bands from Lacerda measurement
ax['A'].fill_between([2013, 2019], 1. + A_Lacerda, 1-A_Lacerda, alpha=0.2,␣
 ↪color='k')
ax['A'].set_yticks([0,0.5, 1,1.5,2])

### absolute magnitude
ax['B'].hist(-2.5 * np.log10(chain_sq['flux_r']) - (10 * np.log10(30)) + 30,␣
 ↪bins=100, histtype='step', density=False, color='k', alpha=0.6)
ax['B'].set_xlabel(r'$H_r$ (mag)', )

ax['B'].set_ylabel('Samples', )
ax['B'].grid()
ax['B'].set_yticks([0, 50, 100, 150])
ax['B'].set_yticks([0, 50, 100, 150])

### lca
ax['C'].hist(chain_sq['lca'], bins=100, histtype='step', density = False,␣
 ↪color='k', alpha=0.6)
ax['C'].set_xlabel(r'LCA', )
ax['C'].grid()
ax['C'].set_xlim(0, 0.5)
ax['C'].set_xticks([0,0.25, 0.5])
ax['C'].set_yticks([0, 50, 100, 150], ['', '', '', ''])

# Lacerda measurement
ax['C'].axvline(A_Lacerda, linestyle='--')

### colors
ax['D'].hist(-2.5 * np.log10(chain_sq['flux_g']/chain_sq['flux_r']),␣
 ↪histtype='step', bins=100, density=False, label=r'$g-r$')
```

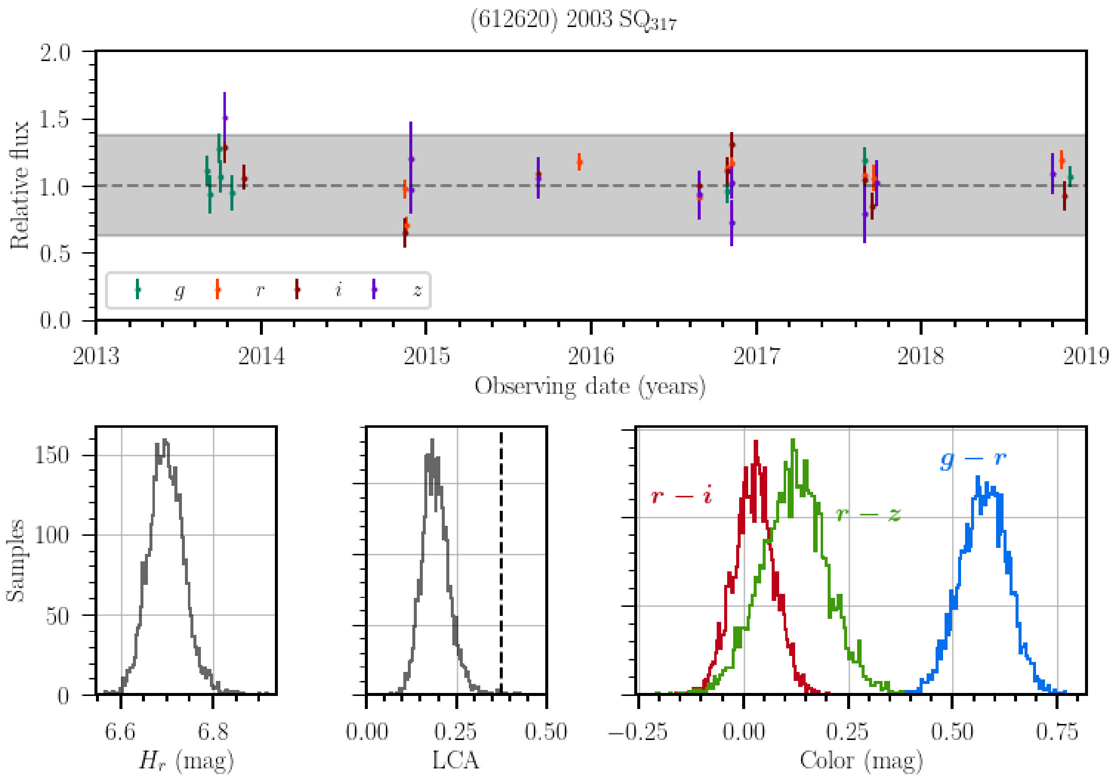
```python
ax['D'].hist(-2.5 * np.log10(chain_sq['flux_r']/chain_sq['flux_i']), histtype='step', bins=100, density=False, label=r'$g-i$')
ax['D'].hist(-2.5 * np.log10(chain_sq['flux_r']/chain_sq['flux_z']), histtype='step', bins=100, density=False, label=r'$g-z$')

ax['D'].set_xlabel('Color (mag)', )
ax['D'].grid()
ax['D'].text(0.47, 130, r'$\boldsymbol{g-r}$', fontsize=18, color='xkcd:cerulean blue')
ax['D'].text(-0.22, 110, r'$\boldsymbol{r-i}$', fontsize=18, color='xkcd:scarlet')
ax['D'].text(0.22, 100, r'$\boldsymbol{r-z}$', fontsize=18, color='xkcd:grass green')
ax['D'].set_yticks([0, 50, 100, 150], ['', '', '', ''])



fig.subplots_adjust(hspace=0.4, wspace=0.5)
fig.set_size_inches(12,8)
pl.suptitle(r'(612620) 2003 SQ$_{317}$',y=0.93)

pl.show()
```



In principle, we could produce 814 of these. Let's… not do that in this notebook.