Michael Kelley
University of Maryland

Jian-Yang Li
Planetary Science Institute

Miguel de Val Borro
Planetary Science Institute

Michael Mommert
Formerly Lowell Obs.

Community

sbpy Overview
Michael S. P. Kelley, University of Maryland
LSST Solar System Science Collaboration Readiness Sprint 3, 23 June 2020

Why Python?

Ubiquitous programming language with a rich standard library.

> *Readability counts.*
>
> - Zen of Python

In 2017, when `sbpy` was proposed, Python had already taken flight within the astronomical community.

> `astropy` v0.1 released June 2012 and v2.0 July 2017

Why Python?

A great default for astro students and early career researchers.

→ See "Ubiquitous programming language."

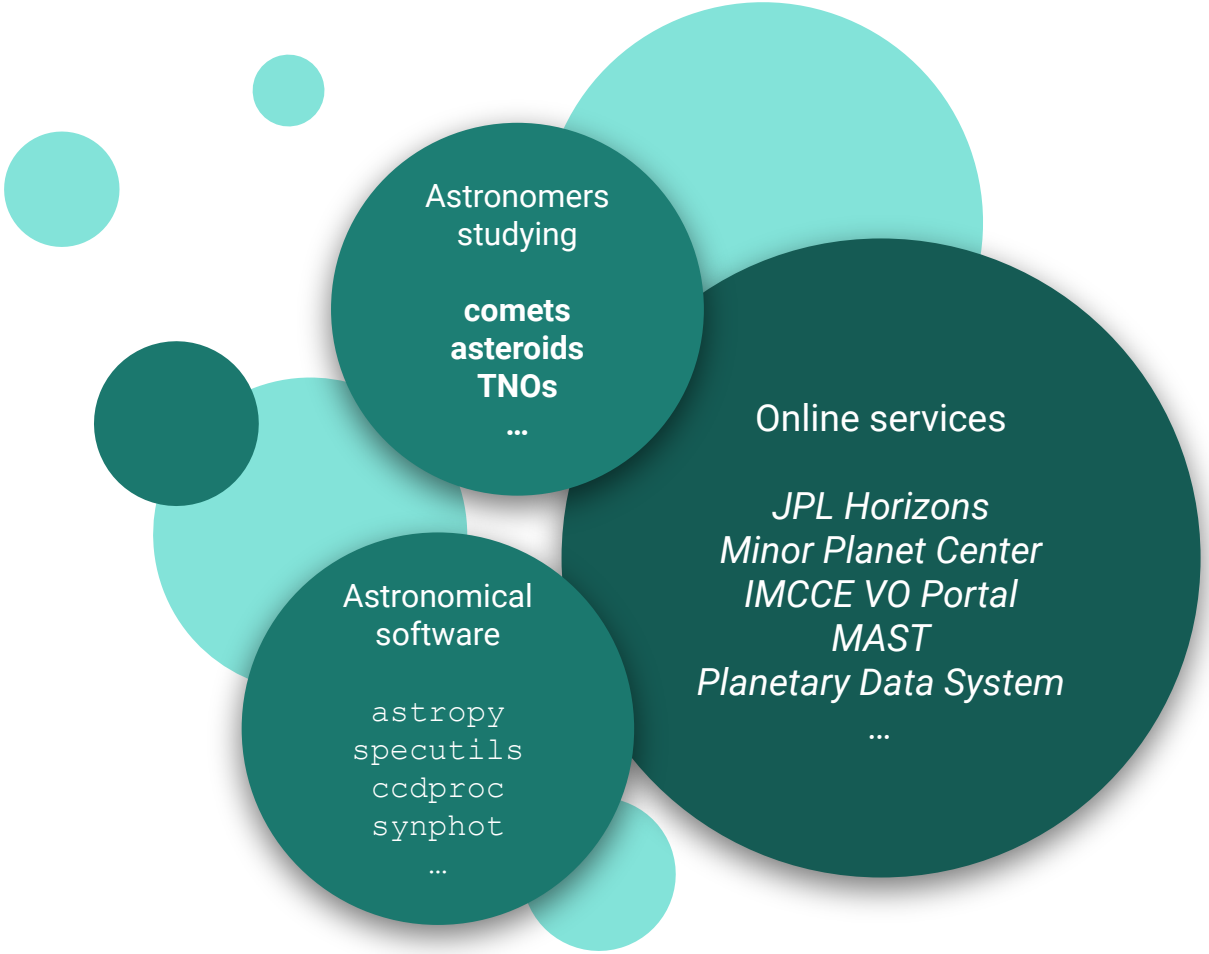
Python and astropy communities promote open source philosophies.

→ Also benefits mid- and late-career scientists (Gordon 2018, PDF/video).

Goals

The sbpy project aims to provide small-body astronomers with domain specific tools that interface with common astronomical tools and datasets.

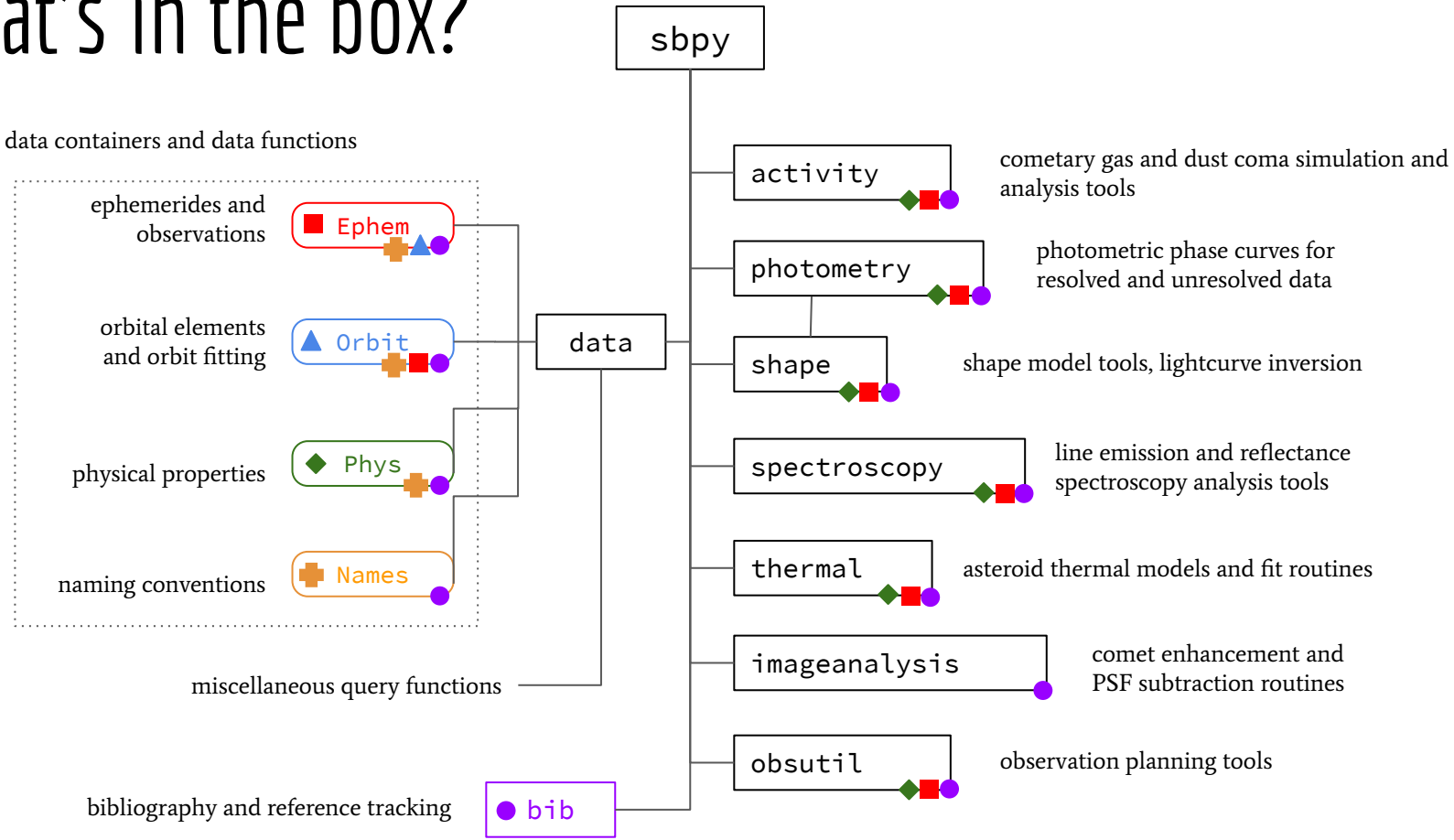
No need to develop a spectroscopy library for asteroid taxonomy, when it already exists.



Goals

OS	**Open source, with a test suite.**	• Critical for reproducibility. • Improves reliability. • Community driven.
AA	**An `astropy` affiliated module.**	• Facilitates code reuse. • Improve interoperability. • Follow interface standards.
Ex	**Extend `astropy`, `astroquery`, ...**	• Wherever appropriate, our efforts feed back into other projects.
LM	**`sbpy` is a living module.**	• Kick-started with NASA funding. • It should grow beyond our baseline package. • Version control preserves old behavior.

What's in the box?



sbpy.data: Where is 2P/Encke today?

```
In [1]: from sbpy.data import Ephem
In [2]: eph = Ephem.from_horizons('2P', id_type='designation', closest_apparition=True)
In [3]: print(eph['date', 'ra', 'dec', 'rh', 'delta', 'phase'])
<QTable length=1>
     epoch            RA       DEC           r              delta         alpha
                     deg       deg           AU              AU           deg
     object        float64   float64      float64         float64       float64
---------------- --------- -------- -------------- ---------------- -------
2459023.40490978 104.94399 23.46338 0.346145942653 1.26710086497057 37.8857

In [4]: print(eph['rh'].to('km'))
[51782695.97233311] km
```

sbpy.data: What is 2P/Encke's orbit?

```
In [1]: from sbpy.data import Orbit, Ephem
In [2]: orb = Orbit.from_mpc('2P', target_type='comet')
In [3]: print(orb['q', 'e', 'i'])
<QTable length=1>
    q          e         i
    AU                  deg
 float64    float64   float64
--------- --------- --------
0.3367202 0.8479982 11.76466
```

sbpy.data: Orbits ⟶ Ephemerides

```
In [4]: # add openorb required parameters
In [5]: orb['H'], orb['G'] = 15, 0.15
In [6]: eph = Ephem.from_oo(orb)
In [7]: print(eph['rh', 'delta', 'phase'])
<QTable length=1>
         r                Delta              alpha
         AU                 AU                deg
      float64            float64            float64
------------------ ------------------ ------------------
0.34609967304037237 1.2669227780766985 37.924127308801715
```

sbpy.calib: Built-in solar spectra

```
In [1]: from sbpy.calib import Sun

In [2]: Sun.show_builtin()
    name                                  description
------------ ------------------------------------------------------------
Castelli1996      Castelli model, scaled and presented by Colina et al. (1996)
   E490_2014                E490-00a (2014) reference solar spectrum (Table 3)
 E490_2014LR E490-00a (2014) low resolution reference solar spectrum (Table 4)
  Kurucz1993               Kurucz (1993) model, scaled by Colina et al. (1996)

In [3]: sun = Sun.from_default()

In [4]: print(sun)
<Sun: E490-00a (2014) reference solar spectrum (Table 3)>
```

sbpy.calib: Apparent brightness of the Sun

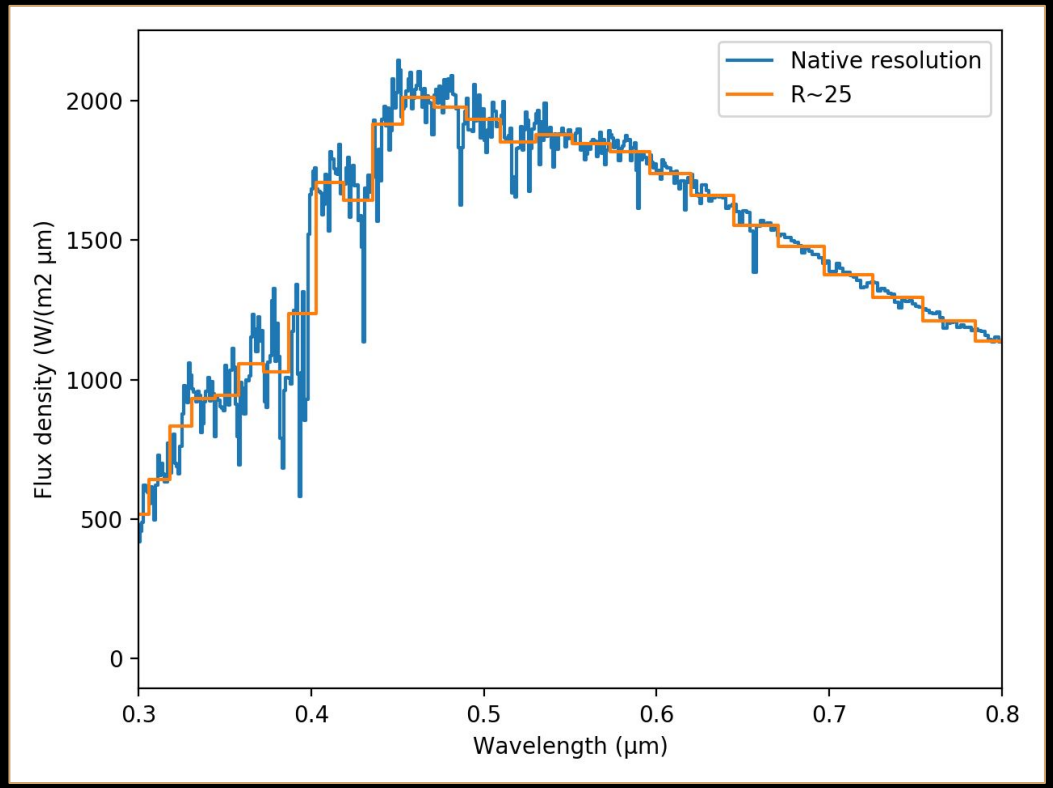
```
In [5]: from sbpy.photometry import bandpass
In [6]: import sbpy.units as sbu

In [7]: bp = bandpass('Johnson V')

In [8]: fluxd = sun.observe(bp, unit=sbu.VEGAmag)
In [9]: print(fluxd)
-26.774715028702648 mag(VEGA)
```

sbpy.calib: Solar spectra

Solar spectra can be plotted at the native resolution of the data, or rebinned. Plot the solar spectrum at the native resolution, and at a resolution of ~25...

See full example in the sbpy documentation:

https://sbpy.readthedocs.io/en/latest/sbpy/calib.html#plot-solar-spectra

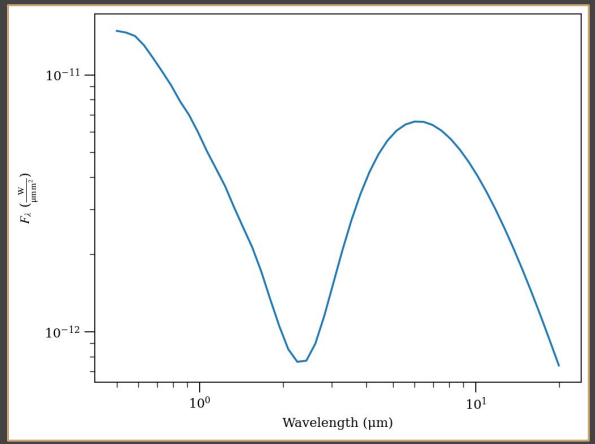


sbpy.photometry: Apparent brightness of Ceres

```
In [1]: import numpy as np
In [2]: import astropy.units as u
In [3]: from sbpy.photometry import HG
In [4]: from sbpy.data import Ephem

In [5]: ceres = HG(3.34 * u.mag, 0.12, radius = 480 * u.km, wfb = 'V')

In [6]: eph = Ephem.from_mpc('Ceres')
In [7]: print(ceres(eph['phase']))
[4.33661803] mag
In [8]: print(ceres(eph['phase']) + 5 * np.log10(eph['rh'] / u.au * eph['delta'] / u.au) *
u.mag)
[8.6749947] mag

In [9]: print(eph['V'])
[8.7] mag
```

sbpy.activity: Dust coma spectrum

```
In [1]: import numpy as np
In [2]: import astropy.units as u
In [3]: import matplotlib.pyplot as plt
In [4]: from sbpy.activity import Afrho, Efrho
In [5]: from sbpy.data import Ephem

In [6]: eph = Ephem.from_mpc('C/2020 F3')
In [7]: w = np.logspace(-0.3, 1.3) * u.um
In [8]: aper = 5 * u.arcsec

In [9]: fsca = Afrho(5000 * u.cm).to_fluxd(w, aper, eph)
In [10]: ftherm = Efrho(20000 * u.cm).to_fluxd(w, aper, eph)

In [11]: plt.plot(w, fsca + ftherm)
```



sbpy.activity: OH coma column density

```
In [1]: import astropy.units as u
In [2]: from sbpy.activity import gas
In [3]: import numpy as np

In [4]: Q = 1e28 / u.s          # production rate
In [5]: v = 0.8 * u.km / u.s    # expansion speed
In [6]: parent = gas.photo_lengthscale('H2O')
In [7]: daughter = gas.photo_lengthscale('OH')
In [8]: coma = gas.Haser(Q, v, parent, daughter)

In [9]: rho = [10, 100, 1000, 10000] * u.km
In [10]: print(coma.column_density(rho))
[7.09928015e+17 5.19060330e+17 3.28255692e+17 1.40606742e+17] 1 / m2
```

sbpy.bib: Reference tracking

```
In [1]: from sbpy import bib
In [2]: bib.track()
...  # repeat previous slide
In [12]: print(bib.to_text())
sbpy:
  software: sbpy:
      Mommert, Kelley, de Val-Borro, Li et al. 2019, The Journal of Open Source Software, Vol 4,
38, 1426
sbpy.activity.gas.core.photo_lengthscale:
  H2O photodissociation lengthscale:
      Cochran & Schleicher 1993, Icarus, Vol 105, 1, 235
  OH photodissociation lengthscale:
      Cochran & Schleicher 1993, Icarus, Vol 105, 1, 235
sbpy.activity.gas.core.Haser.__init__:
  model:
      Haser 1957, Bulletin de la Societe Royale des Sciences de Liege, Vol 43, 740
sbpy.activity.gas.core.Haser._column_density:
  model:
      Newburn & Johnson 1978, Icarus, Vol 35, 3, 360
```

What is planned?

When effort is restored this fall, on our horizons are:

- Comets: dust syndynes and synchrones, gas vectorial model, ice sublimation.
- Disk resolved photometric functions.
- NAIF Spice integration (via `spiceypy`).
- Lowell asteroid database, IMCCE tools.
- Observational tools: when to observe, finder charts.

What is planned?

Anything of general interest to astronomers studying small solar system objects.

Contributions are welcome!

File a GitHub issue to introduce your idea.

Find out more

Documentation and examples:

sbpy.readthedocs.io

Class and module docstrings →

GitHub tutorial repository.

```
In [1]: from sbpy.activity import Afrho
In [2]: Afrho?
Init signature: Afrho(value, unit=None, dtype=None, copy=None)
Docstring:
Coma dust quantity for scattered light.

``Afrho`` objects behave like `~astropy.units.Quantity`
objects
with units of length.


Parameters
----------
value : number, `~astropy.units.Quantity`
    The value(s).

unit : string, `~astropy.units.Unit`, optional
    The unit of the input value.  Strings must be parseable by
    :mod:`~astropy.units` package.

...
```

#sbpy channel in the astropy Slack workspace.

`sbpy` needs your help

- Use it, test it, break it! Then give us your feedback.
- What would you like to see/have in the future? For LSST?
- Spread the word. More users and contributions leads to better code.
- If you use it in your work, please cite it: `10.21105/joss.01426`.

Thanks