github.com/mshemuni/

octans

Observed **C**alcula**T**ed diagr**A**m a**N**d lightcurve**S**

Mohammed Niaei[1,2] Ozgur Basturk[3]

[1] Ankara University, Graduate School of Natural & Applied Sciences
[2] TÜBİTAK National Academic Network and Information Center (ULAKBİM)
[3] Ankara University, Faculty of Science, Astronomy & Space Sciences Dept.

What is it?

Octans is a software package designed to **simplify** the acquisition and analysis of **light curves** to derive their **timings of extrema** with **different techniques.**

How it does what it does

Portal

- XMatch
  - Xmatchs the given coordinate
- Retrieve light curve
  - Kepler
  - TESS
  - ASAS
  - ...

And the best part, It returns **XLightCurves**

Portal Example

Retrieve a list of light curves

```
from octans import Portal

# portal = Portal.from_coordinates(281.28812083, 42.45108092)
portal = Portal.from_name("kepler-8")
xlcs = portal.kepler()
```

How it does what it does

XLightCurve

- Analyse
  - Smooth light curves
  - Find boundaries
  - Fold
  - Time & Flux Correction
- Minima
  - Curve fit
  - Periodogram
  - Kwee - van Woerden (1956)
  - Cross correlation with a template
  - Thoroughgood (2004) and more...
- Visualize
  - Display light curves

XLightCurve Example

Load **light curve data**

```
from octans import XLightCurve
import pandas as pd

data = pd.read_csv("Light/Curve/File")

xlc = XLightCurve(
    data["TIME"], data["FLUX"], data["FLUX_ERROR"]
)
```

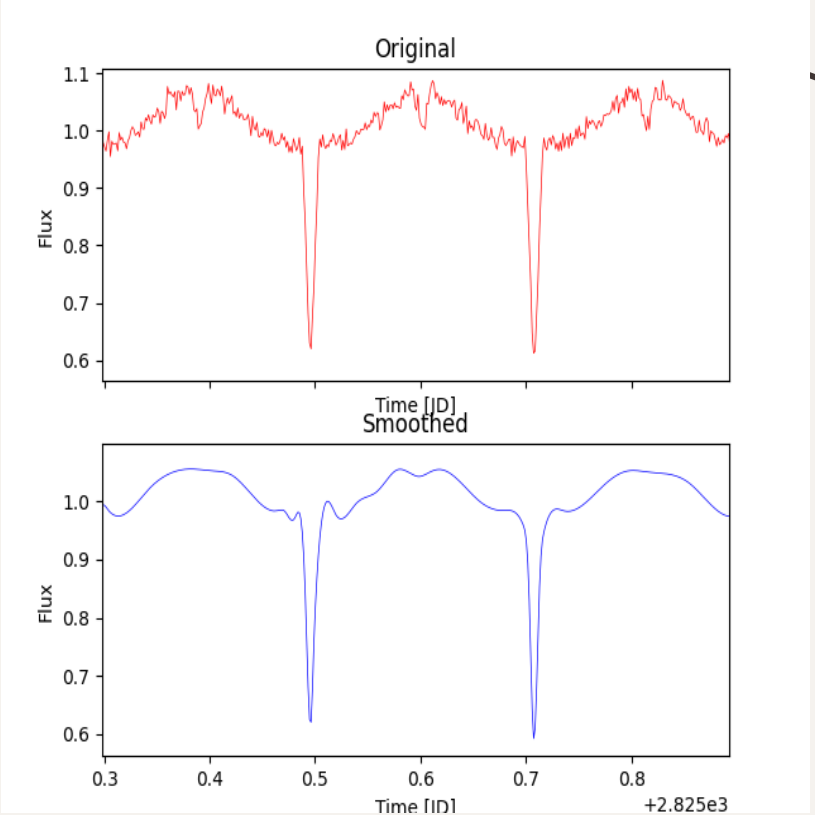
XLightCurve Example

Let's **smooth** a light curve

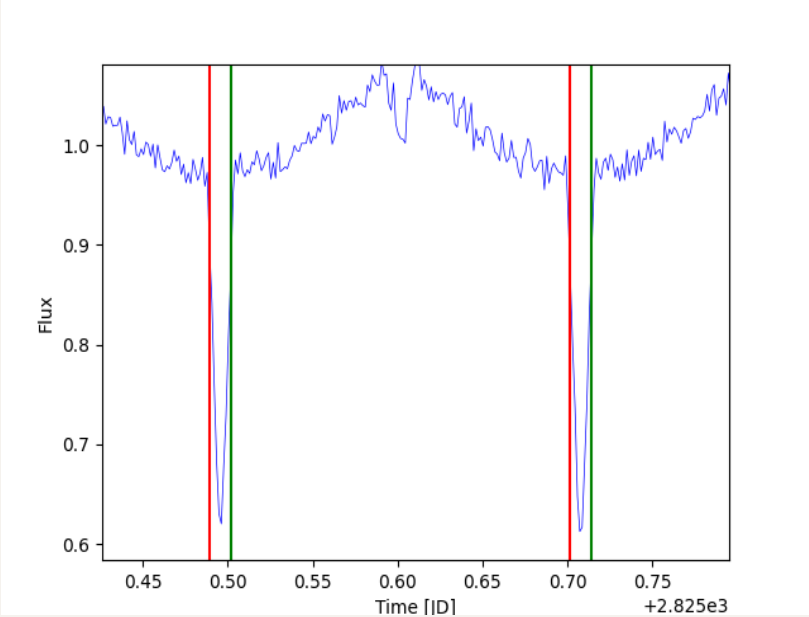
```
smooth_xlc = xlc.smooth_b_spline(s=3)
fig, (ax1, ax2) = plt.subplots(1, 2, sharex=True)

xlc.plot(ax=ax1, color="red")
smooth_xlc.plot(ax=ax2, color="blue")

plt.show()
```



XLightCurve Example



or **find boundaries**

```
boundaries = xlc.boundaries_extrema()

fig, ax = plt.subplots()
xlc.plot(ax=ax, color="blue")

for boundary in boundaries:
    ax.axvline(boundary[0], color="red")
    ax.axvline(boundary[1], color="green")

plt.show()
```

XLightCurve Example

and **find timings of minima**

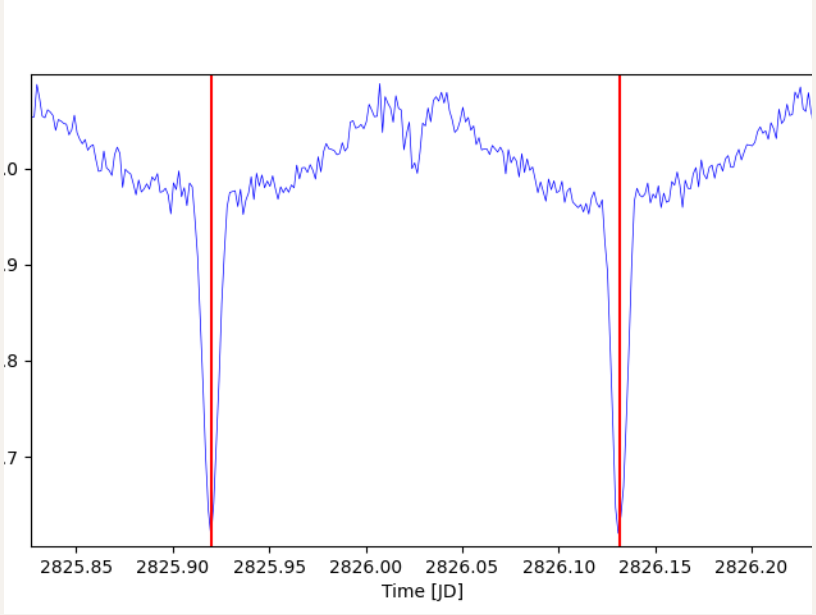
```
minimas = xlc.minima_fit()

fig, ax = plt.subplots()
xlc.plot(ax=ax, color="blue")

for minima in minimas[3:5]:
    ax.axvline(minima.time.jd, color="red")

#Minima1: 2825.9197446528474 ± 0.0015278394522997567
#Minima2: 2826.1316128322815 ± 0.0006916204743035943

plt.show()
```



What to do next?

* Robust handling of **measurement uncertainties**
* **More methods** for extrema measurements
* Improved **GUI**
* **Documentation**
* **O-C Portal**. ability to retrieve minimas from databases and create publish-quality O-C diagrams

Every help appreciated...

Thank You!

Do you have any questions?
niaei@pardus.org.tr
https://github.com/mshemuni/octans/issues

Poster Number: **FP04**