aurora-asi-lib

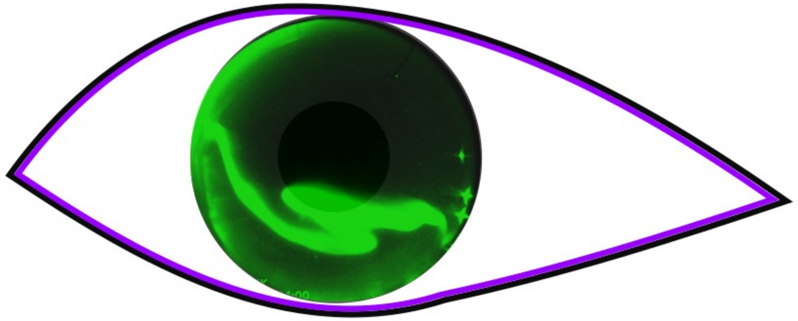
Aurora All-Sky Imager Library

Easily download, plot, animate, and analyze auroral all sky imager (ASI) data

Mykhaylo (Mike) Shumko, Bea Gallardo-Lacourt, Isaac Thompson, Alexa Halford, and Kyle Murphy

aurora-asi-lib

Aurora All-Sky Imager Library



Follow along! Install with:

```
python3 –m pip install aurora-asi-lib
```

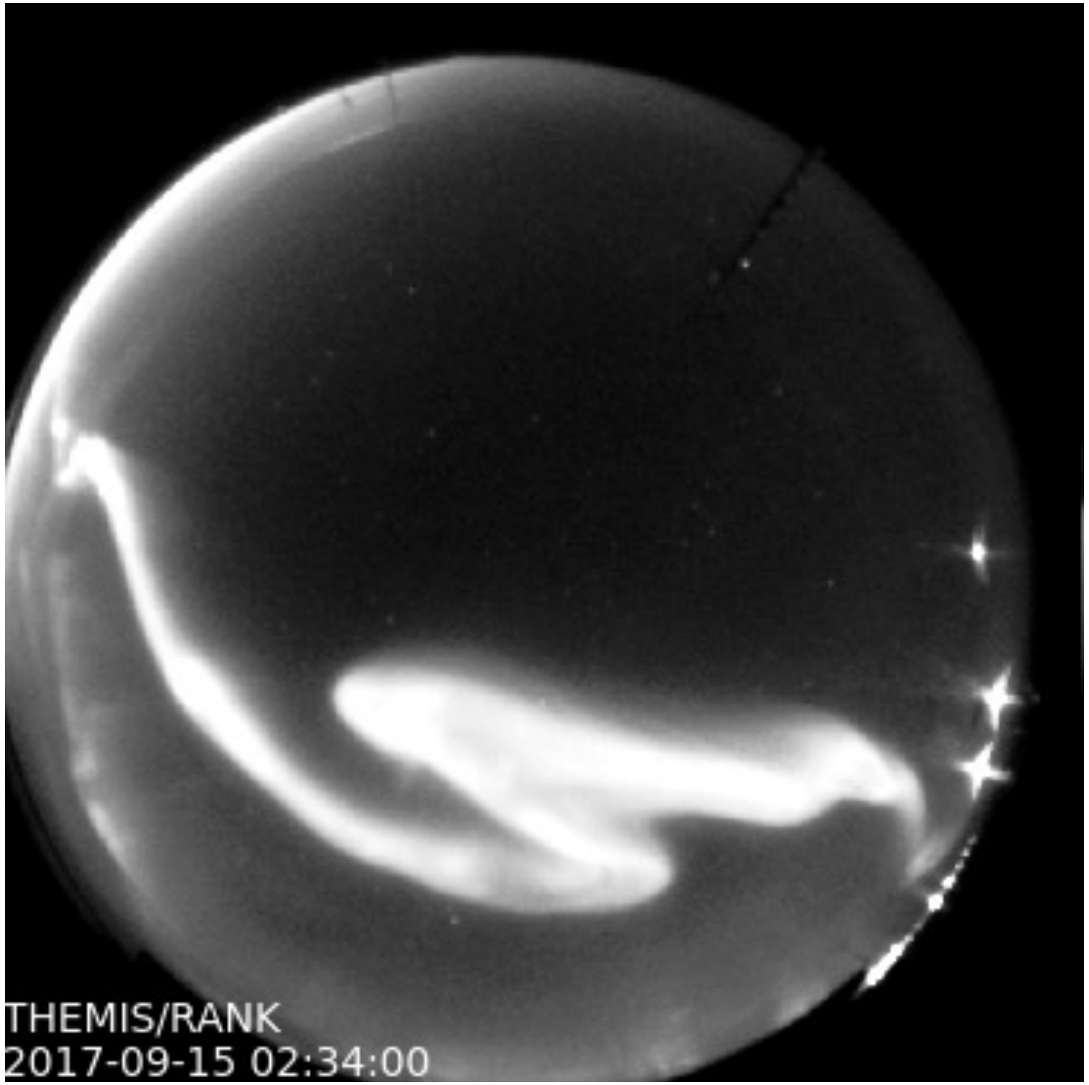
aurora-asi-lib overview

What?

A python package that enables seamless and painless handling and analysis of auroral images

Why?

Auroral researchers do similar analysis steps---our goal with asilib is to enable researchers to focus their time and energy on what matters: studying the aurora!



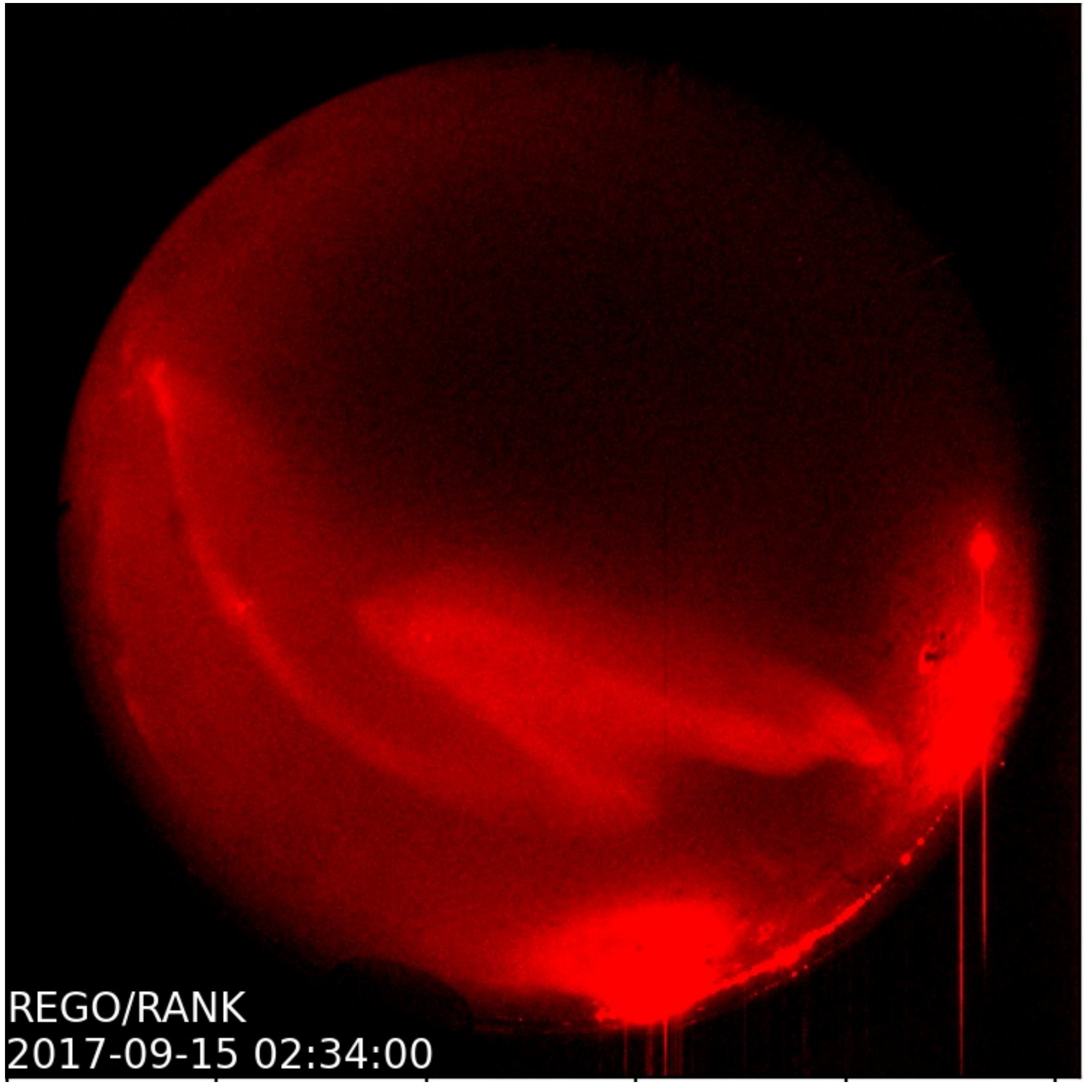
aurora-asi-lib overview

Supported camera arrays:

- **THEMIS**
- **REGO**

Once these two arrays are fully supported, we plan to add other camera arrays to asilib.

- **TREX**
- **ASI-derived models**



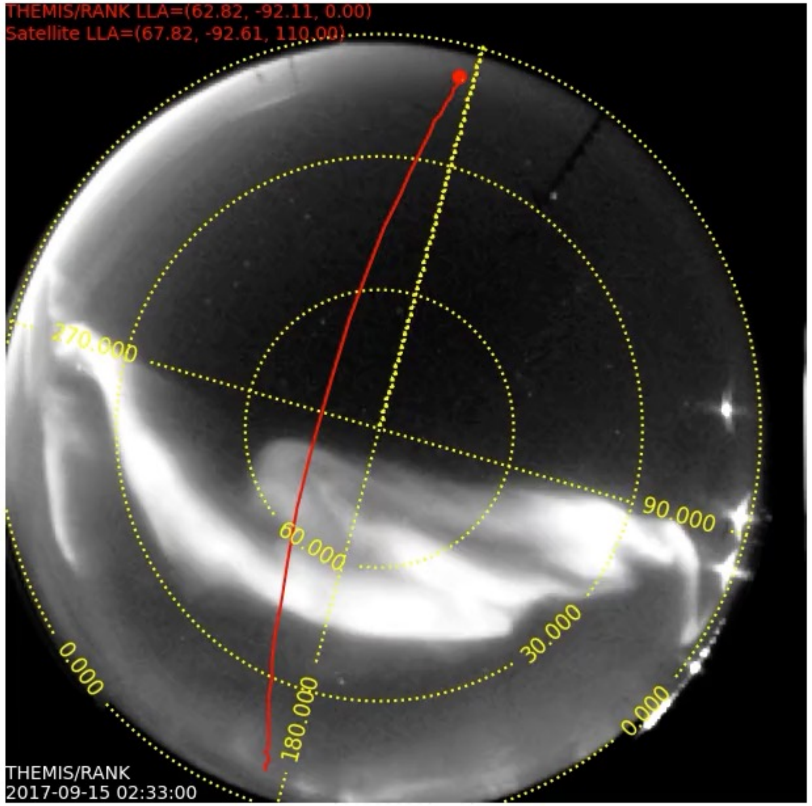
What can it do?

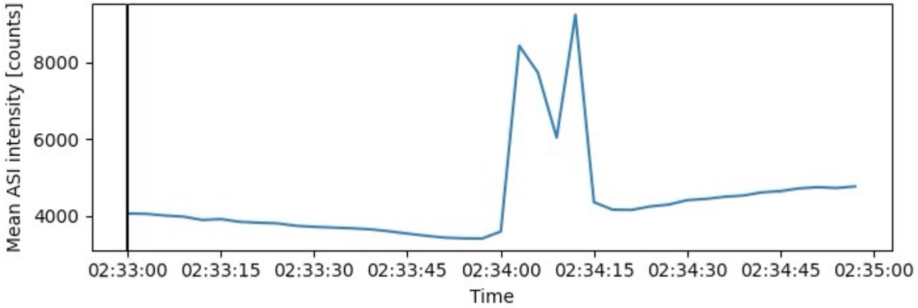
Plot one fisheye lens frame:

    asilib.plot_frame()

Make a movie:

    asilib.plot_movie()*
    asilib.plot_movie_generator()*


THEMIS/RANK LLA=(62.82, -92.11, 0.00)
Satellite LLA=(67.82, -92.61, 110.00)
270.000
90.000
60.000
30.000
0.000
0.000
180.000
THEMIS/RANK
2017-09-15 02:33:00



* Requires ffmpeg

What can it do?

Map a satellite's location:

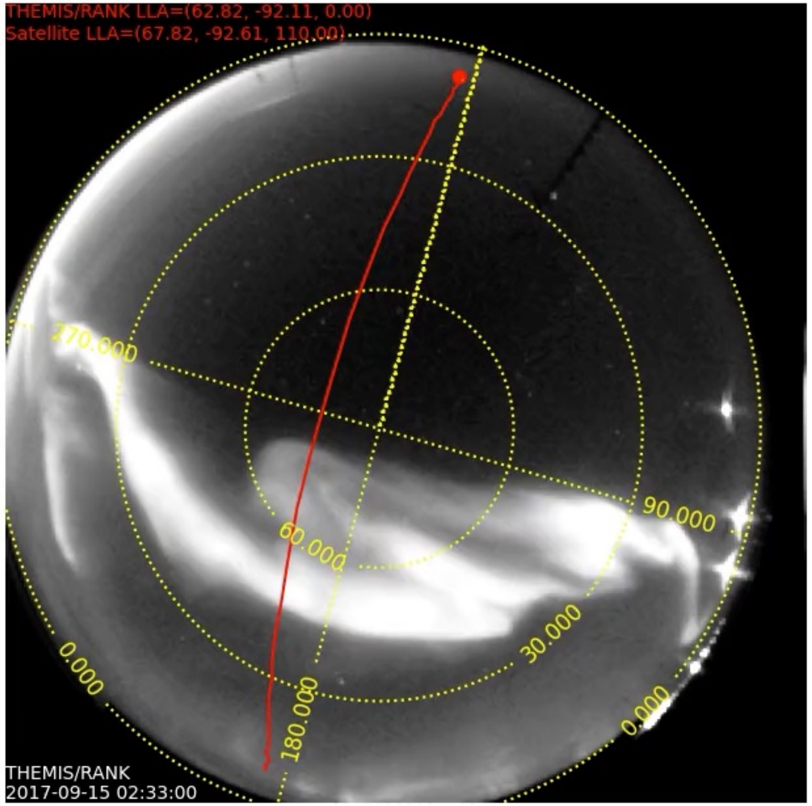
asilib.lla2azel()

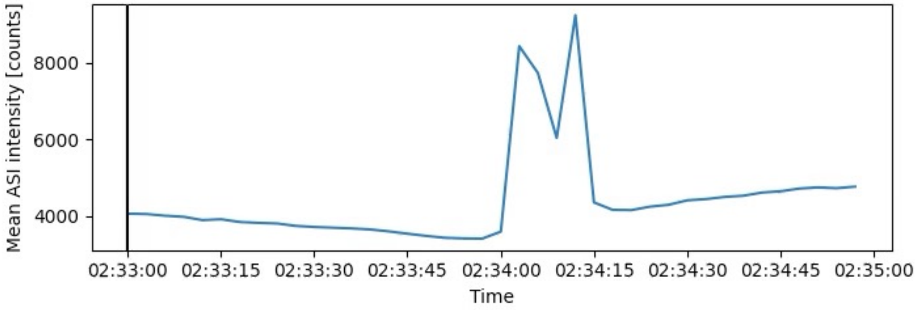
asilib.lla2footprint()*

Calculate equal areas in the image:

asilib.equal_area()

* Requires IRBEM


THEMIS/RANK LLA=(62.82, -92.11, 0.00)
Satellite LLA=(67.82, -92.61, 110.00)
270.000
90.000
60.000
30.000
0.000
180.000
0.000
THEMIS/RANK
2017-09-15 02:33:00



What can it do?

Plot a keogram:

```
asilib.plot_keogram()
```

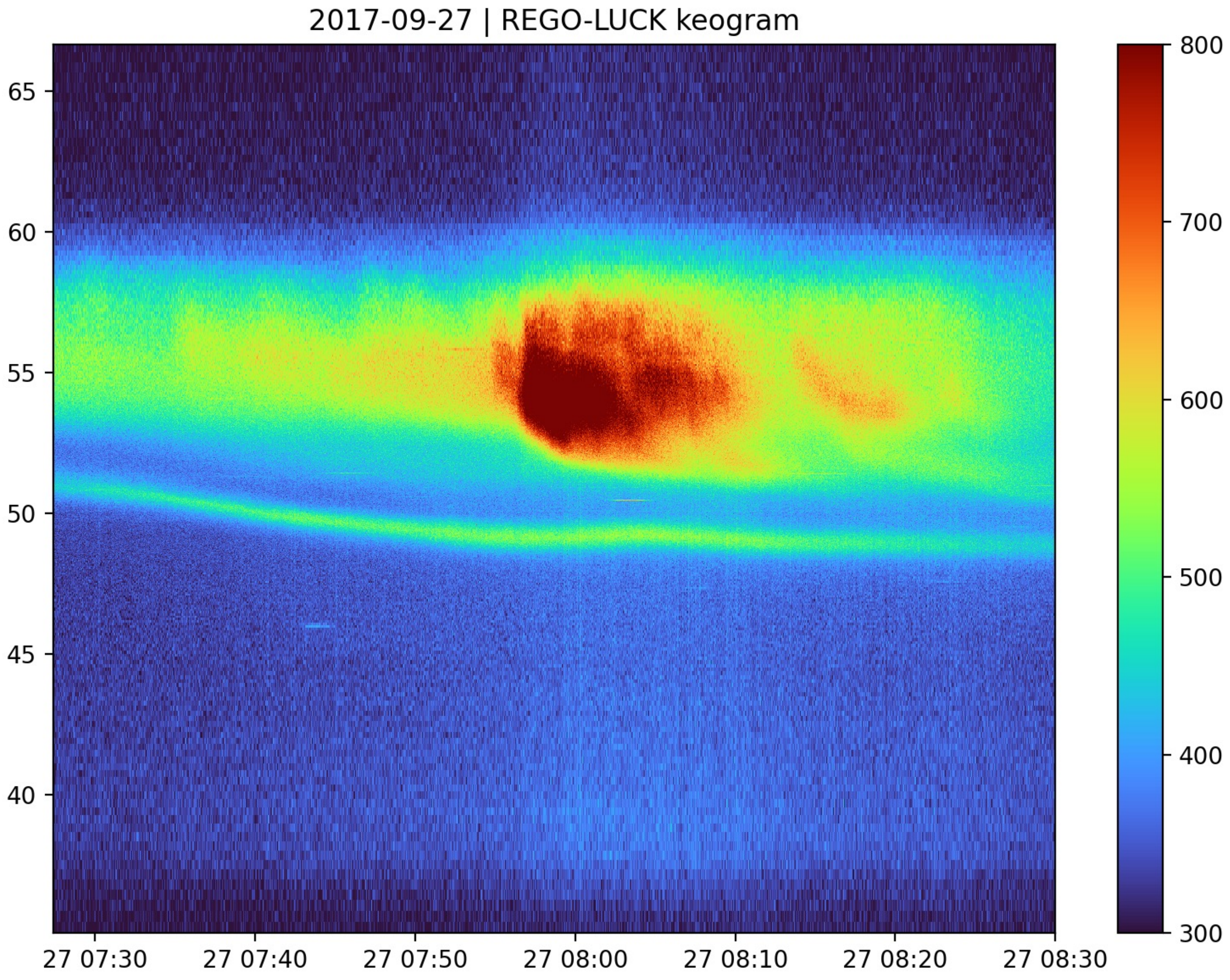
Load data

```
asilib.load_img()
asilib.load_cal()
```

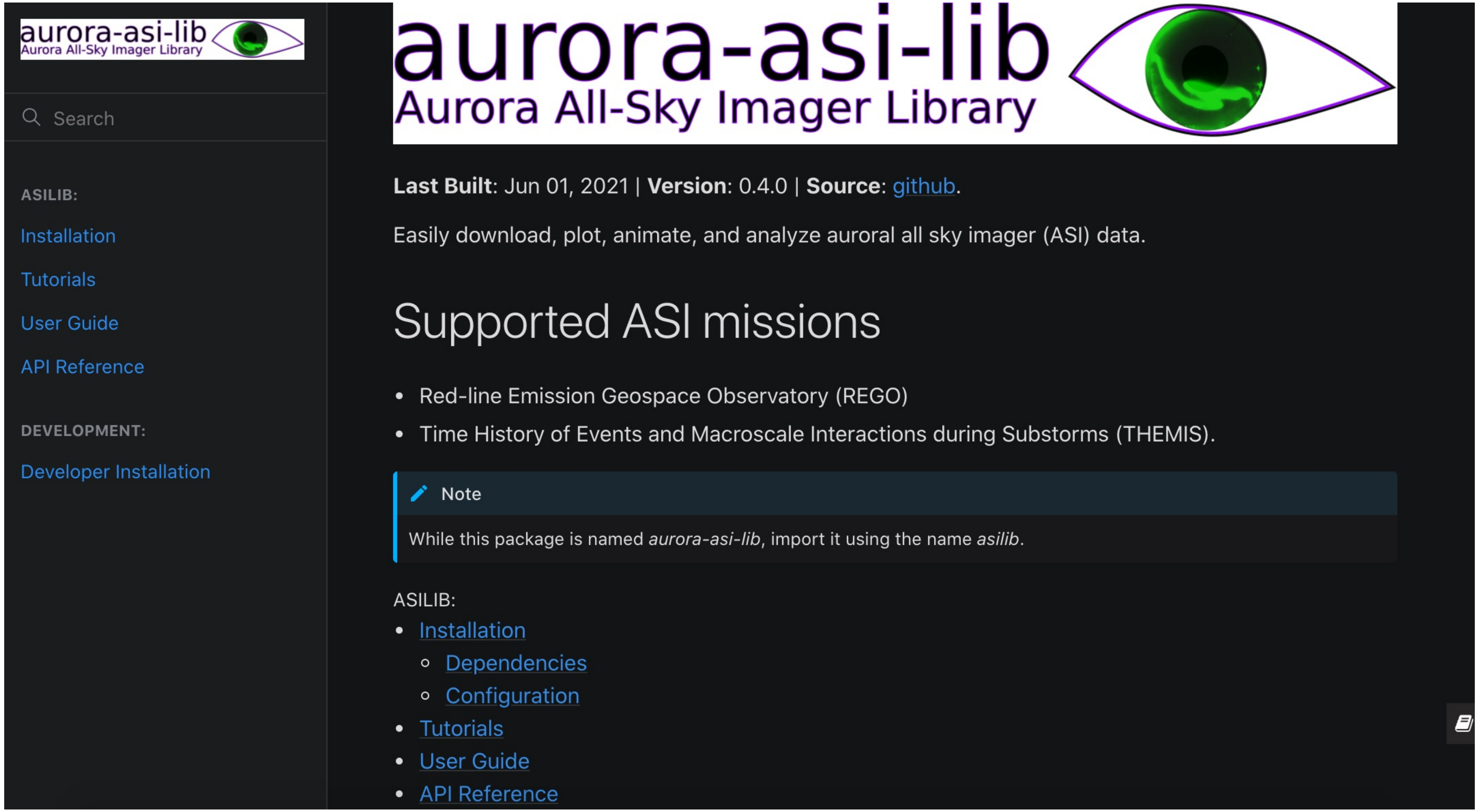
If a file is not found, one will be automatically downloaded!

Bulk download data

```
asilib.download_themis_cal()
asilib.download_themis_img()
```



Didn’t memorize the commands? Documentation: aurora-asi-lib.readthedocs.io



ASILIB:

- Installation
- Tutorials
- User Guide
- API Reference

DEVELOPMENT:

- Developer Installation

aurora-asi-lib
Aurora All-Sky Imager Library

Last Built: Jun 01, 2021 | **Version**: 0.4.0 | **Source**: github.

Easily download, plot, animate, and analyze auroral all sky imager (ASI) data.

Supported ASI missions

- Red-line Emission Geospace Observatory (REGO)
- Time History of Events and Macroscale Interactions during Substorms (THEMIS).

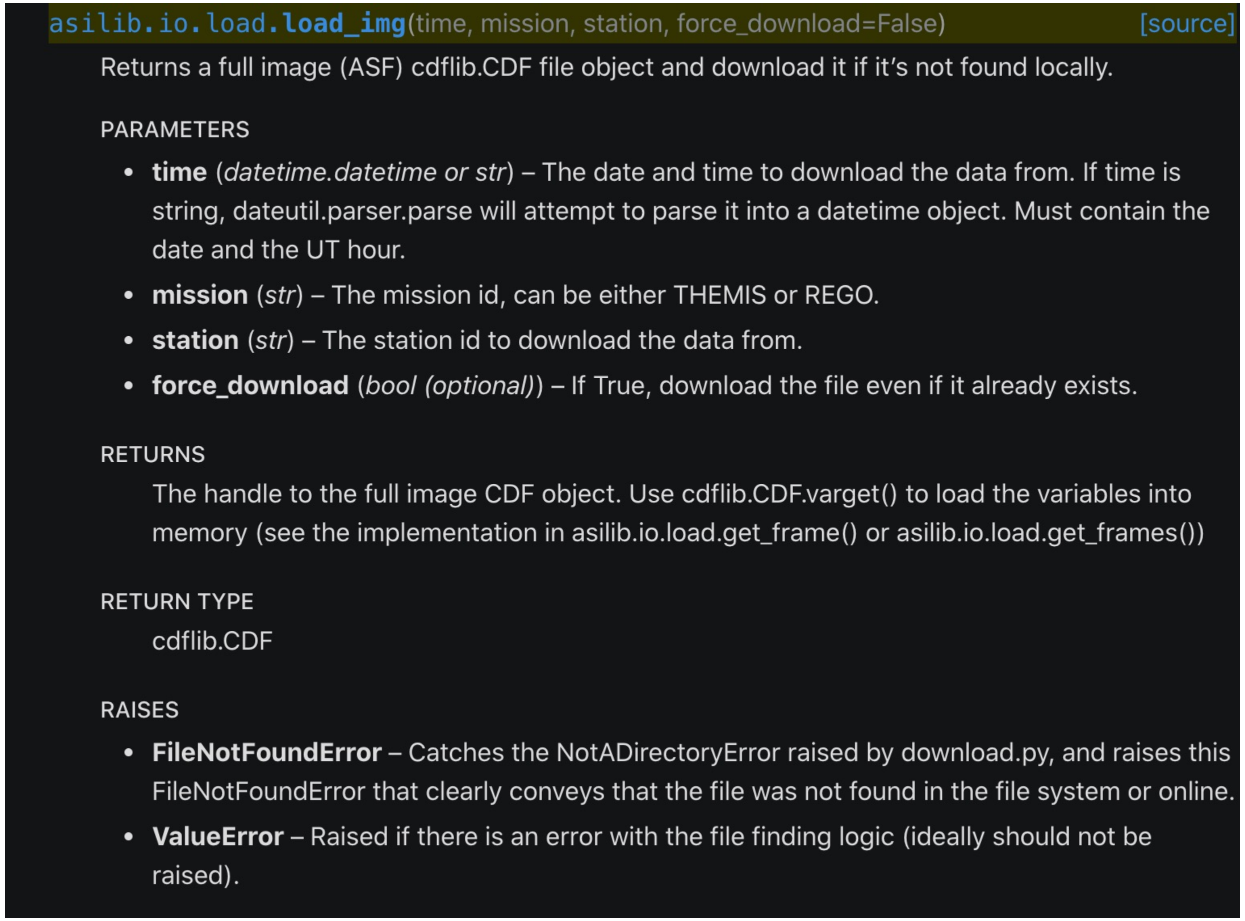
Note

While this package is named *aurora-asi-lib*, import it using the name *asilib*.

ASILIB:

- Installation
  - Dependencies
  - Configuration
- Tutorials
- User Guide
- API Reference

Didn’t memorize the commands?
Documentation: aurora-asi-lib.readthedocs.io



`asilib.io.load.load_img`(time, mission, station, force_download=False) [source]

Returns a full image (ASF) cdflib.CDF file object and download it if it’s not found locally.

PARAMETERS

- **time** (*datetime.datetime or str*) – The date and time to download the data from. If time is string, dateutil.parser.parse will attempt to parse it into a datetime object. Must contain the date and the UT hour.
- **mission** (*str*) – The mission id, can be either THEMIS or REGO.
- **station** (*str*) – The station id to download the data from.
- **force_download** (*bool (optional)*) – If True, download the file even if it already exists.

RETURNS

The handle to the full image CDF object. Use cdflib.CDF.varget() to load the variables into memory (see the implementation in asilib.io.load.get_frame() or asilib.io.load.get_frames())

RETURN TYPE

cdflib.CDF

RAISES

- **FileNotFoundError** – Catches the NotADirectoryError raised by download.py, and raises this FileNotFoundError that clearly conveys that the file was not found in the file system or online.
- **ValueError** – Raised if there is an error with the file finding logic (ideally should not be raised).

One class to rule them all

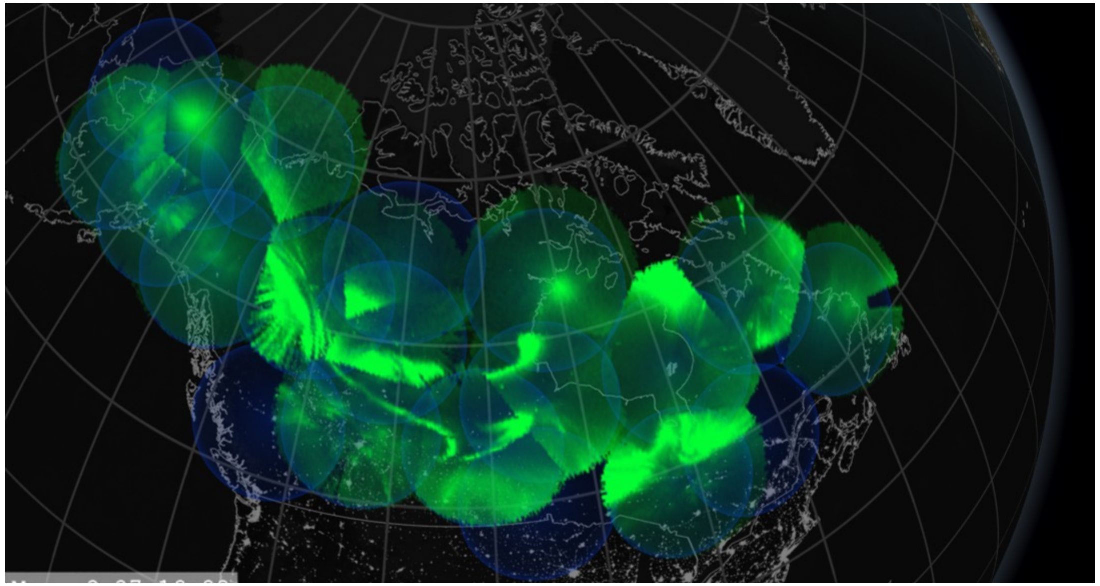
- The most usable (and fun!?) python libraries have a central class:
  - numpy.array
  - pandas.DataFrame
  - xarray.DataArray
  - pysat.Instrument
  - bs4.BeautifulSoup
  - …

One class to rule them all

- The most usable (and fun!?) python libraries have a central class:
  - numpy.array
  - pandas.DataFrame
  - xarray.DataArray
  - pysat.Instrument
  - bs4.BeautifulSoup
  - …
- And now:
  - asilib.Imager

Ongoing Development Topics

- Handle computer resources effectively
- Project the fisheye images to a map
- Unify the asilib functionality into an asilib.Imager() class
- Integrate with Aurora X
- Update the documentation with examples and tutorials
- And add other imager arrays as plugins.



We need your help! Please contact me, mykhaylo.shumko@nasa.gov if you’d like to contribute or have ideas (I am always interested in ways to improve this code)

How to get started

python3 –m pip install aurora-asi-lib (import as asilib)

Documentation: https://aurora-asi-lib.readthedocs.io

Source code: https://github.com/mshumko/aurora-asi-lib

Thank you for listening!