STScI | SPACE TELESCOPE SCIENCE INSTITUTE

EXPANDING THE FRONTIERS OF SPACE ASTRONOMY

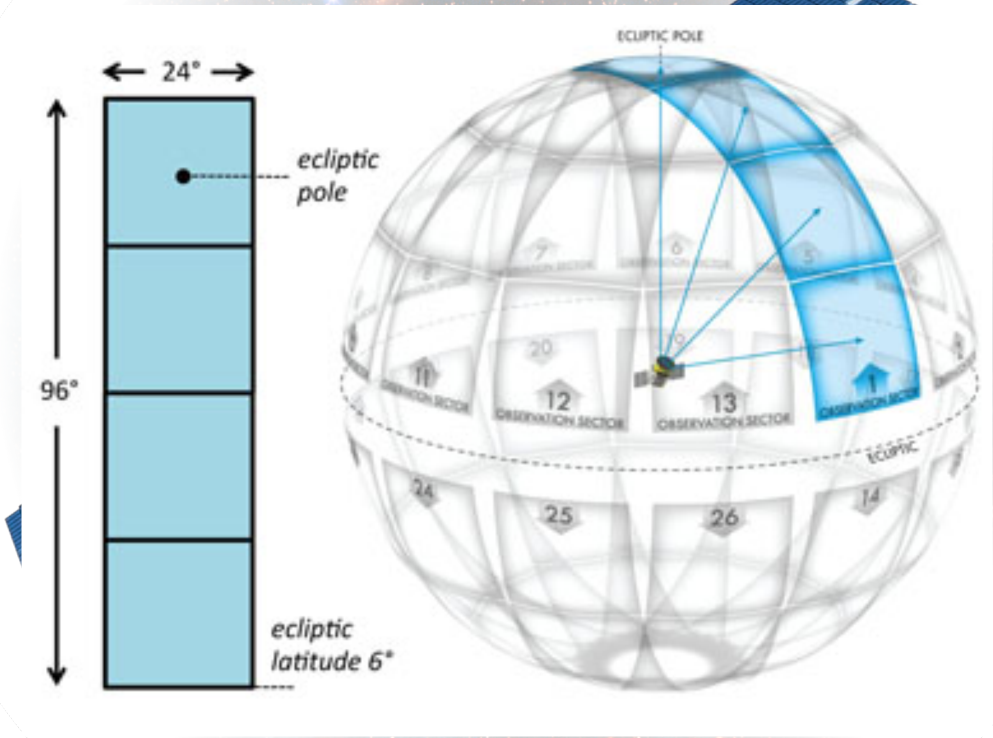
Astrocut: A cutout service for TESS full-frame image sets

Clara Brasseur

13 November, 2018

The Transiting Exoplanet Survey Satellite (TESS)

- Launched 18 April 2018
- 4 cameras each with 4 2k x 2k CCDs
  - 600-1000nm detector (blue to the near-IR)
  - Resolution is 21 arcsec/pixel
  - Total FOV: 24 x 96 deg
- 2 year mission planned
  - 1st year southern hemisphere
  - 2nd year northern hemisphere
- Mission divided into 26 sectors
  - Same pointing for entire sector
  - Sector lasts ~27 days (two orbits)

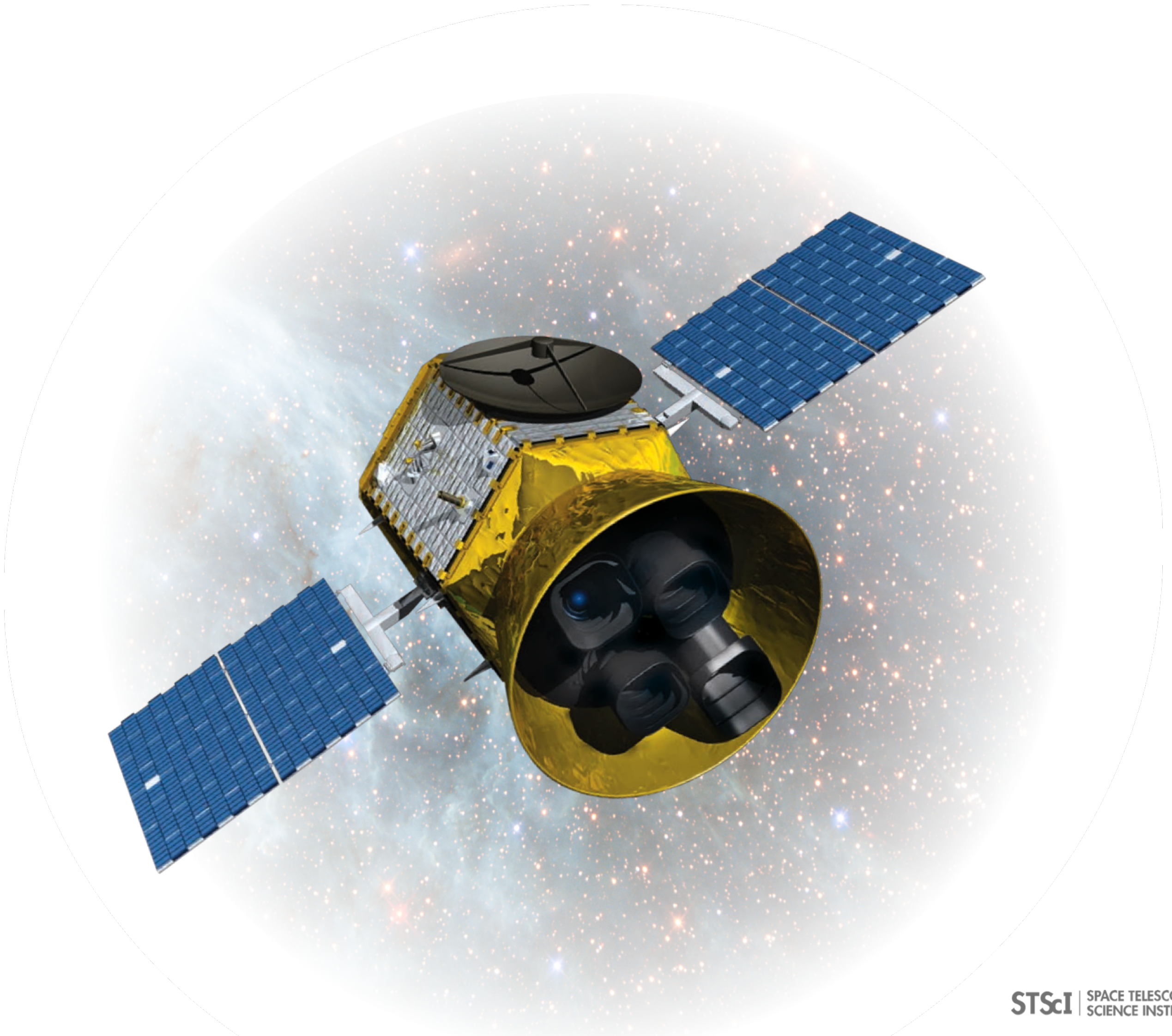


The Transiting Exoplanet Survey Satellite (TESS)

- Variety of data products:
  - Light Curves
  - Target pixel files
  - Full-frame images

- Full-frame images (FFIs):
  - One FFI per CCD (16 total)
  - ~35 MB per FFI
  - 30 minute cadence

That's a lot of data...



The Need for a cutout service

- The Tess FFIs will be a large and incredibly valuable scientific dataset
- It will be possible to do time-domain astronomy on the FFIs
- FFIs are large and scientists will need to cut out sections of interest
- One sector of FFIs for one CCD is ~50 GB

Astrocut goals:

- Provide a user friendly way to get cutouts from a sector of FFIs
- Eliminates the need for users to download the entire sector of FFIs
- Return cutout as a TESS pipeline compatible target pixel file

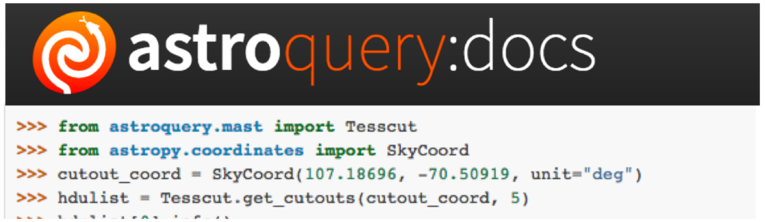
The TESScut/Astrocut Ecosystem

Astrocut:

- Underlying functionality
- What this talk is (mostly) about

TESScut web service:

- URL-based web service that runs Astrocut on STSCI servers



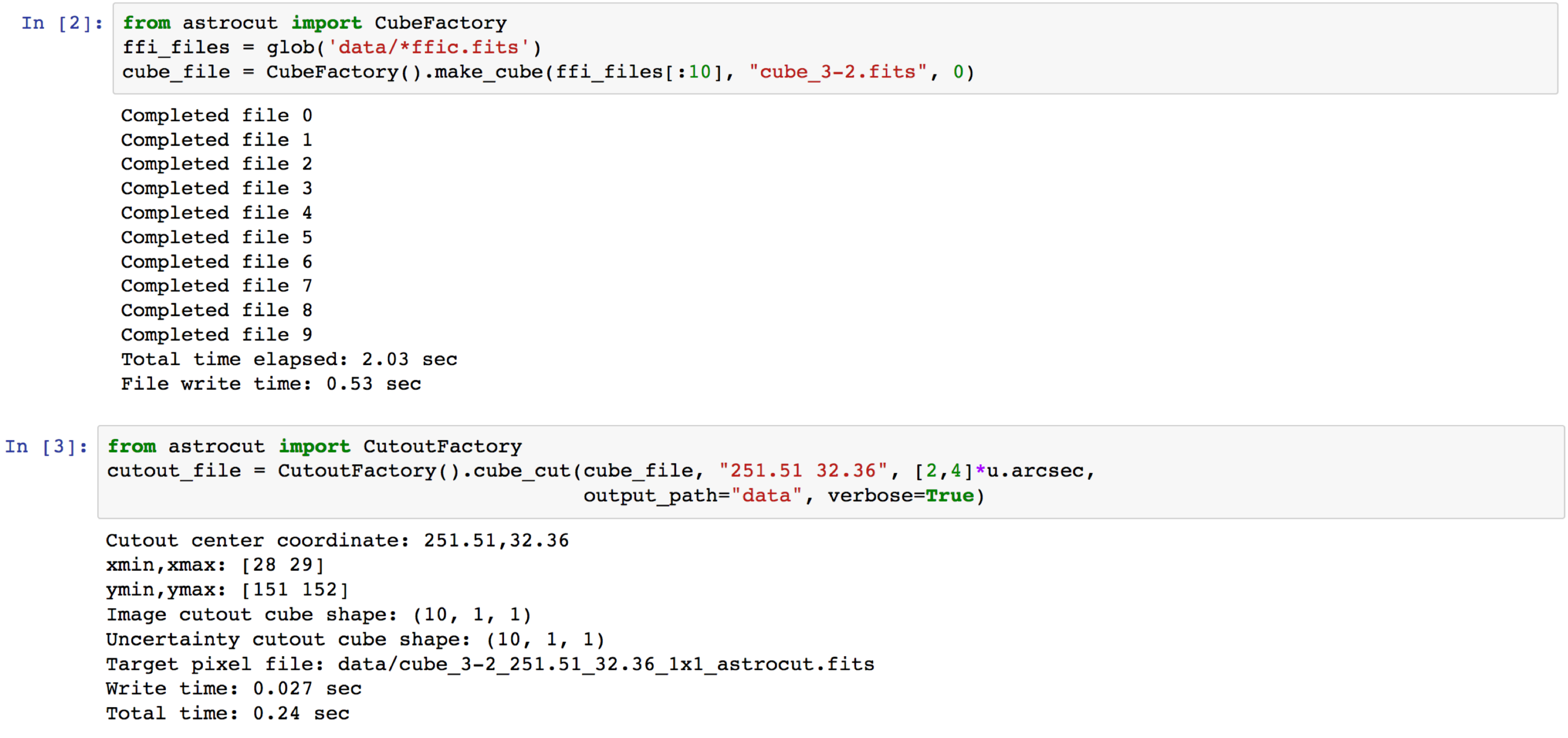
```
>>> from astroquery.mast import Tesscut
>>> from astropy.coordinates import SkyCoord
>>> cutout_coord = SkyCoord(107.18696, -70.50919, unit="deg")
>>> hdulist = Tesscut.get_cutouts(cutout_coord, 5)
```

astroquery.mast.Tesscut:

- Python wrapper around the TESScut web service

Astrocut: How it works

Two parts: Cubes and Cutouts



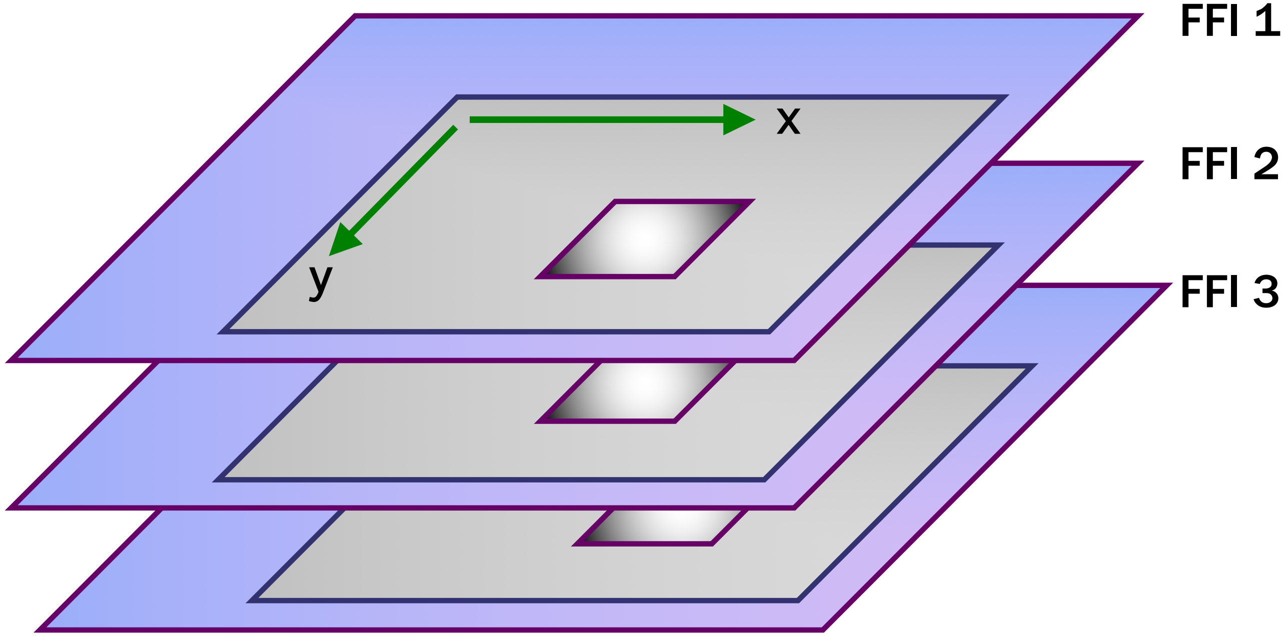
```
In [2]: from astrocut import CubeFactory
        ffi_files = glob('data/*ffic.fits')
        cube_file = CubeFactory().make_cube(ffi_files[:10], "cube_3-2.fits", 0)
```

```
Completed file 0
Completed file 1
Completed file 2
Completed file 3
Completed file 4
Completed file 5
Completed file 6
Completed file 7
Completed file 8
Completed file 9
Total time elapsed: 2.03 sec
File write time: 0.53 sec
```

```
In [3]: from astrocut import CutoutFactory
        cutout_file = CutoutFactory().cube_cut(cube_file, "251.51 32.36", [2,4]*u.arcsec,
                                               output_path="data", verbose=True)
```

```
Cutout center coordinate: 251.51,32.36
xmin,xmax: [28 29]
ymin,ymax: [151 152]
Image cutout cube shape: (10, 1, 1)
Uncertainty cutout cube shape: (10, 1, 1)
Target pixel file: data/cube_3-2_251.51_32.36_1x1_astrocut.fits
Write time: 0.027 sec
Total time: 0.24 sec
```

Astrocut: How it works



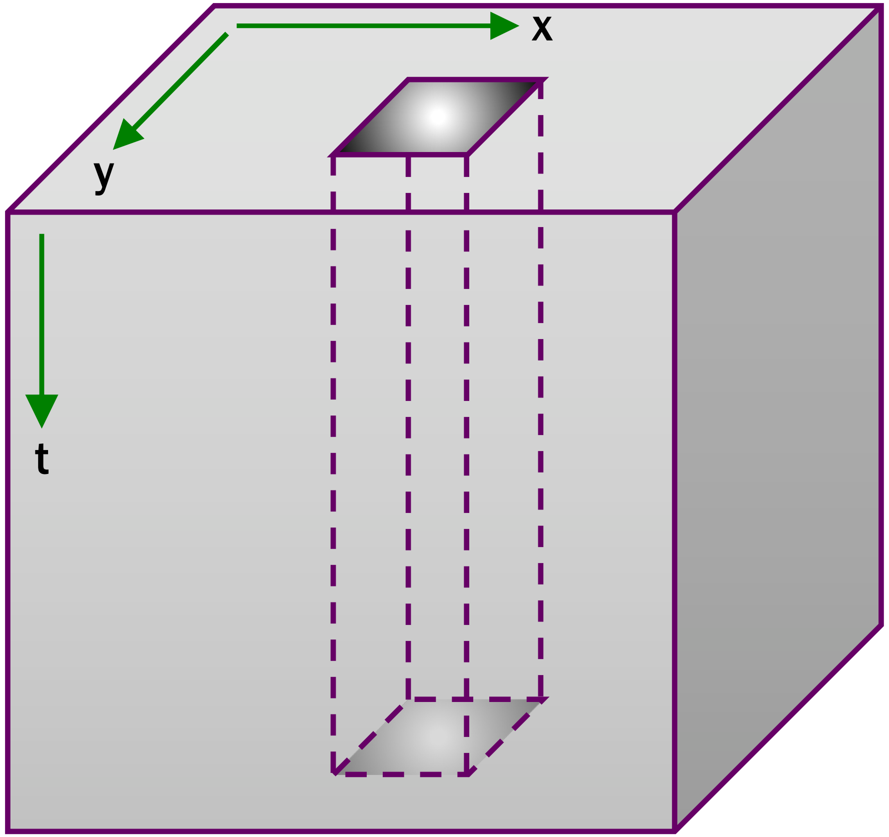
What is cubing?

- Instead of opening lots of files (~1250 per sector) every time, do it once, and then open one (BIG) file every time.
- Arrange data in file to minimize seek actions when doing cutouts.

Why Cube? Performance!

- Without cubing:
  ~25 sec for 10x10 cutout over 1348 FFIs with 8 threads
- With cubing:
  ~0.58 sec for same cutout with single thread

Astrocut: How it works



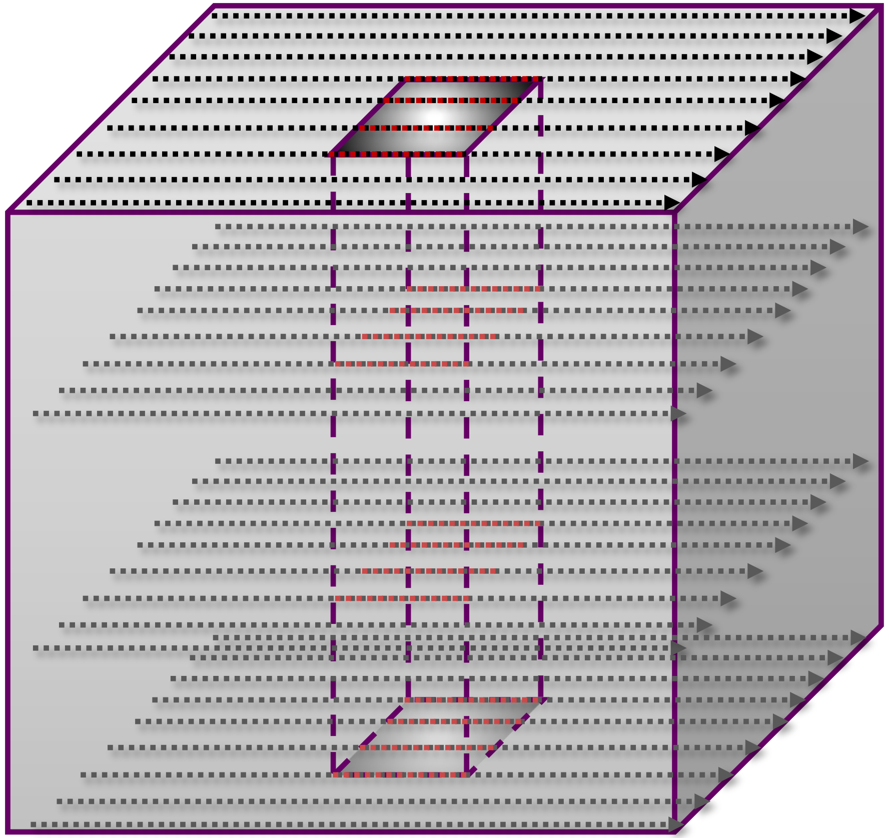
What is cubing?

- Instead of opening lots of files (~1250 per sector) every time, do it once, and then open one (BIG) file every time.
- Arrange data in file to minimize seek actions when doing cutouts.

Why Cube? Performance!

- Without cubing:
  ~25 sec for 10x10 cutout over 1348 FFIs with 8 threads
- With cubing:
  ~0.58 sec for same cutout with single thread

Astrocut: How it works



Seeks:

[illegible]

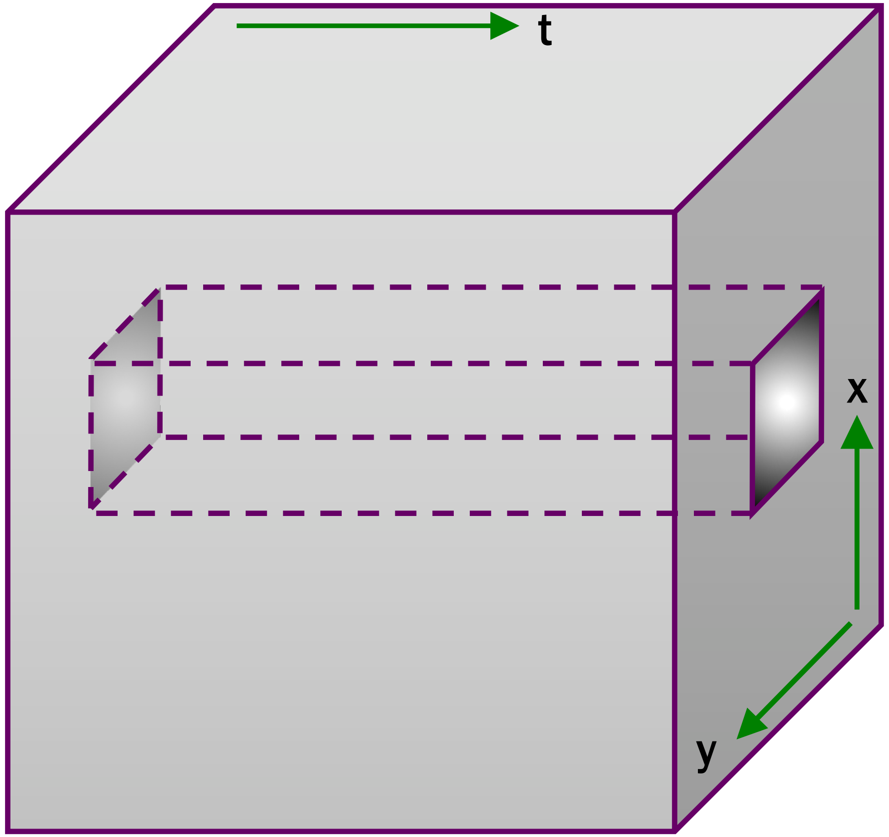
What is cubing?

- Instead of opening lots of files (~1250 per sector) every time, do it once, and then open one (BIG) file every time.
- Arrange data in file to minimize seek actions when doing cutouts.

Why Cube? Performance!

- Without cubing:
  ~25 sec for 10x10 cutout over 1348 FFIs with 8 threads
- With cubing:
  ~0.58 sec for same cutout with single thread

Astrocut: How it works



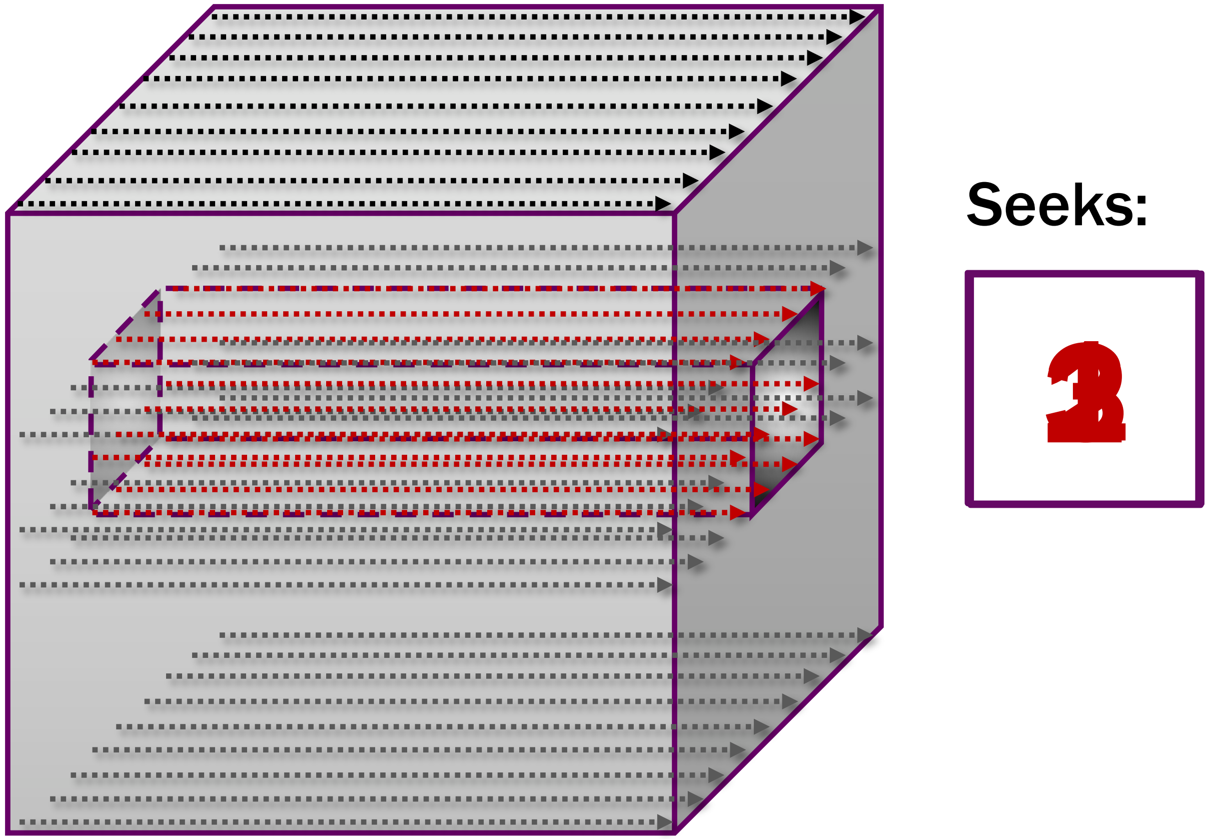
What is cubing?

- Instead of opening lots of files (~1250 per sector) every time, do it once, and then open one (BIG) file every time.
- Arrange data in file to minimize seek actions when doing cutouts.

Why Cube? Performance!

- Without cubing:
  ~25 sec for 10x10 cutout over 1348 FFIs with 8 threads
- With cubing:
  ~0.58 sec for same cutout with single thread

Astrocut: How it works



What is cubing?

- Instead of opening lots of files (~1250 per sector) every time, do it once, and then open one (BIG) file every time.
- Arrange data in file to minimize seek actions when doing cutouts.

Why Cube? Performance!

- Without cubing:
  ~25 sec for 10x10 cutout over 1348 FFIs with 8 threads
- With cubing:
  ~0.58 sec for same cutout with single thread

Astrocut versus TESScut: Use Cases

Use TESScut if:

- You want the most straightforward option
- You want a few to a handful of cutouts
- You don’t want enormous cutouts
- You don’t have access to a system with ~60 GB memory

* If you are a python user consider using Tesscut in astroquery…

Use Astrocut if:

- You want to cutout every star in the sky
- You want to make really big cutouts
- You want do do something non-standard
- You have access to a system with plenty of memory

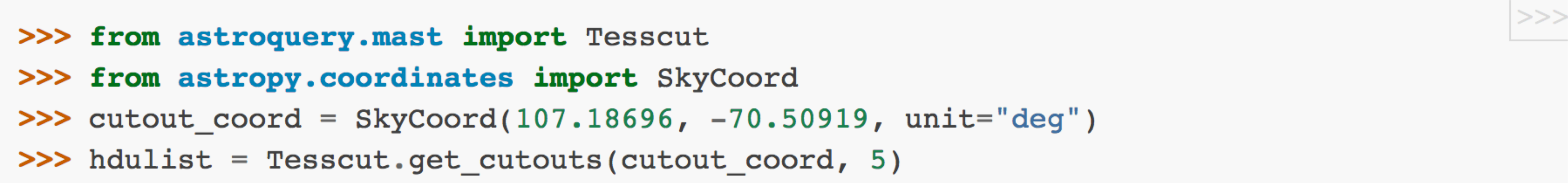
*Astrocut is open source and on github…

TESScut: A Brief Tour

- TESScut allows access to Astrocut functionality on MAST servers
- Eliminates need for users to download anything but requested cutout

Three ways to query

1. Web form: https://mast.stsci.edu/tesscut
2. HTTP GET request: https://mast.stsci.edu/tesscut/api/v0.1/astrocut?ra=250.25&dec=3.52&y=3&x=3
3. astroquery.mast.Tesscut:



```
>>> from astroquery.mast import Tesscut
>>> from astropy.coordinates import SkyCoord
>>> cutout_coord = SkyCoord(107.18696, -70.50919, unit="deg")
>>> hdulist = Tesscut.get_cutouts(cutout_coord, 5)
```

Where to get more info

Astrocut

- Docs: https://astrocut.readthedocs.io
- On github: https://github.com/spacetelescope/astrocut

TESScut

- Main site: https://mast.stsci.edu/tesscut
- API docs: https://mastdev.stsci.edu/tesscut/docs/

astroquery.mast.Tesscut

- https://astroquery.readthedocs.io/en/latest/mast/mast.html#tesscut

cbrasseur@stsci.edu

@cebrasseur