Astronomical Image Reduction With

Erik Tollerud

@eteq

STScI | SPACE TELESCOPE SCIENCE INSTITUTE

Project Scientist, Data Analysis Tools Branch
Astropy Coordination Committee

Easy/Basic/Lazy

Astronomical Image Reduction With

Erik Tollerud

@eteq

STScI | SPACE TELESCOPE SCIENCE INSTITUTE

Project Scientist, Data Analysis Tools Branch
Astropy Coordination Committee

Why  python for this Task?

- It's easy to write code at a high-level to do low-level tasks.
- It integrates well with notebooks (which are great for "quick" reduction).
- It has 

What is Astropy?

The Astropy Project is a community effort to develop a common core package for Astronomy in Python and foster an ecosystem of interoperable astronomy packages.

What is Astropy?

The Astropy Project is a community effort to develop a common core package for Astronomy in Python and foster an ecosystem of interoperable astronomy packages.

This means both ***by*** and ***for*** the community

What is Astropy?

The Astropy Project is a community effort to develop a common core package for Astronomy in Python and foster an ecosystem of interoperable astronomy packages.

This means both *by* and *for* the community

(Professional) Astronomers help write it

What is Astropy?

The Astropy Project is a community effort to develop a common core package for Astronomy in Python and foster an ecosystem of interoperable astronomy packages.

This means both *by* and *for* the community

(Professional) Astronomers help write it

It should be useful for them as part of their day-to-day work

The astropy Project and package

> The Astropy Project is a community effort to develop a common core package for Astronomy in Python and foster an ecosystem of interoperable astronomy packages.

Core package "astropy" ≠ "Astropy Project"

The core package is what's in github repo *astropy/astropy*. I.e., what "`pip install astropy`" or "`conda install astropy`" gets you.

"Astropy Project" includes all the coordinated and affiliated packages and community.

What Unites Affiliated Packages?

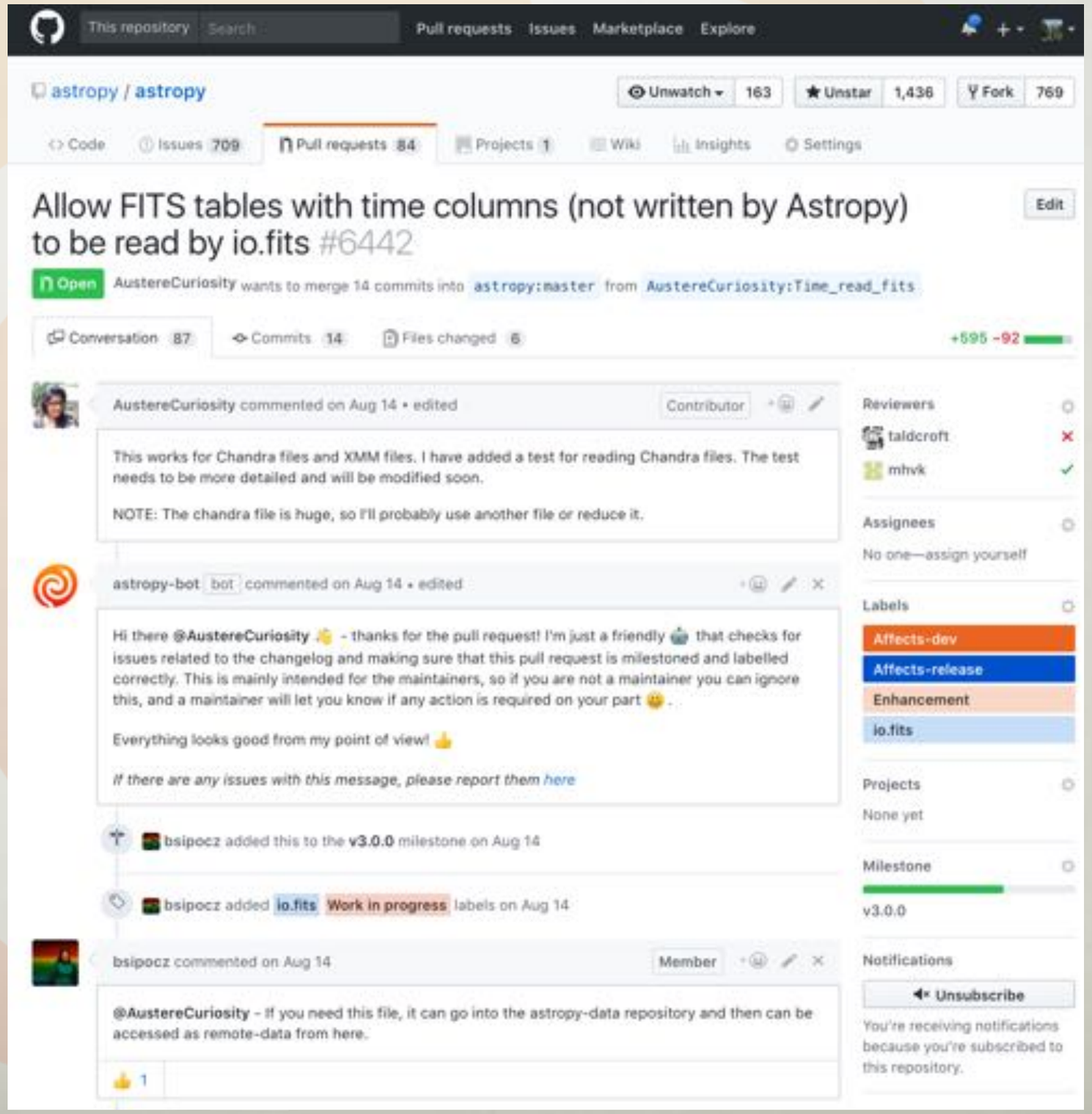
- A common goal and vision: reducing duplication and embracing good coding practices (testing, docs), open development
- Listing on http://affiliated.astropy.org (curated and reviewed by the Astropy coordination committee)
- (For many) a package template

What Unites Affiliated Packages?

- A common goal and vision: reducing duplication and embracing good coding practices (testing, docs), open development
- Listing on http://affiliated.astropy.org (curated and reviewed by the Astropy coordination committee)
- (For many) a package template

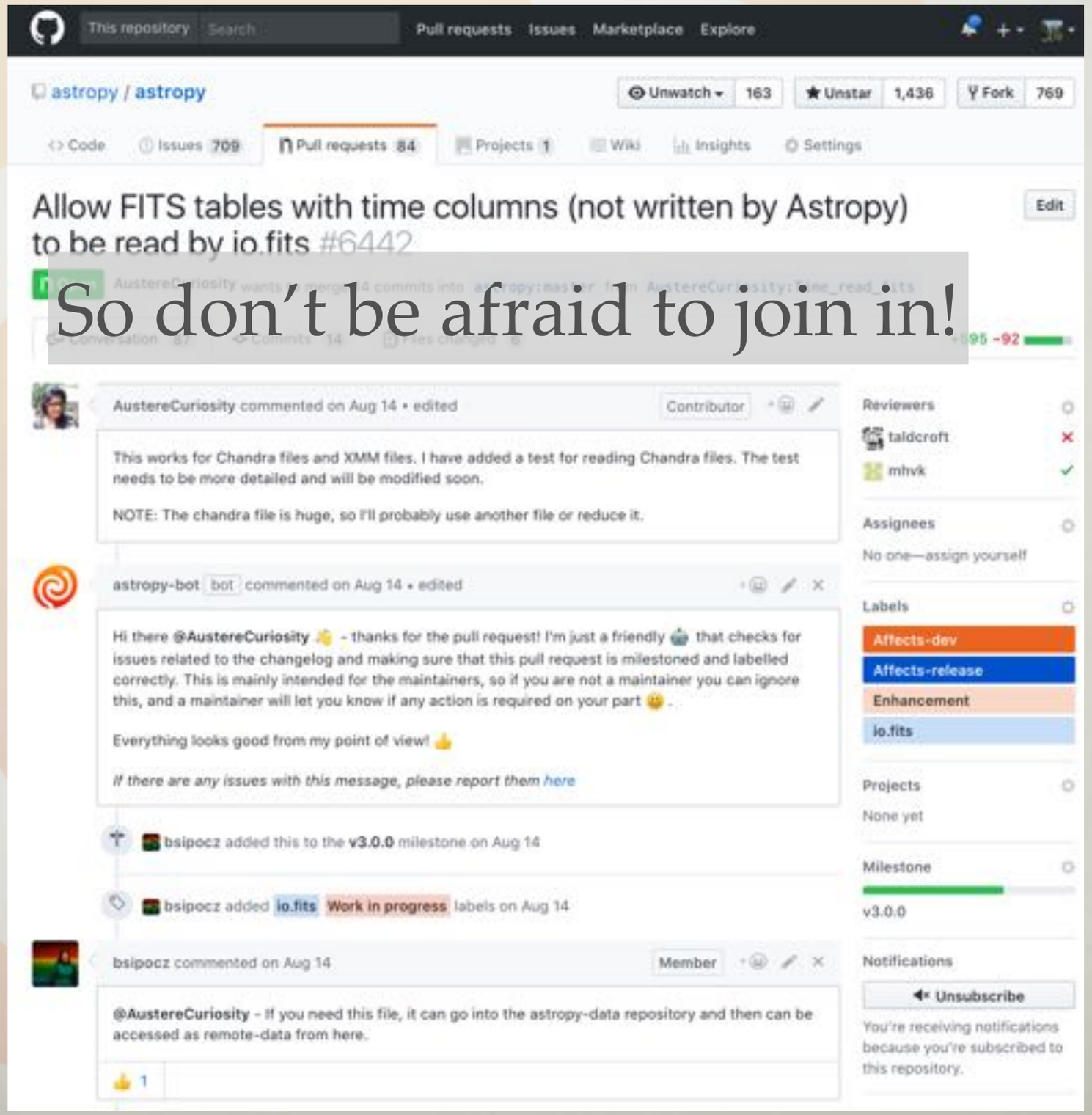
Remember: Astropy is Community Developed

GitHub



Remember: Astropy is Community Developed

GitHub



So don't be afraid to join in!

What exactly do we mean by "reduction"?

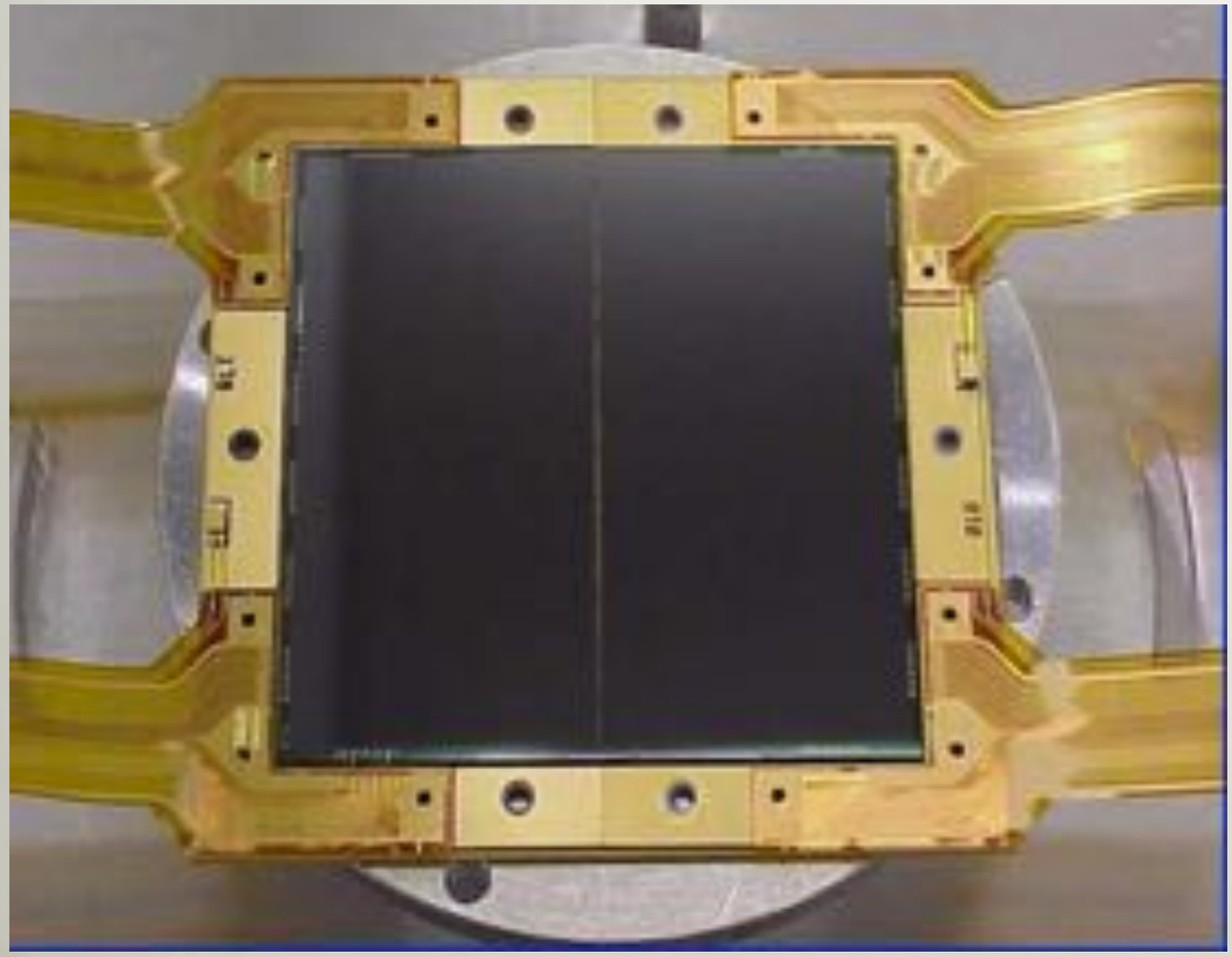
What exactly do we mean by "reduction"?

- At it's core: taking a lot of data and making it simpler (e.g., plotting an average).

What exactly do we mean by "reduction"?

- At it's core: taking a lot of data and making it simpler (e.g., plotting an average).
- In astronomy, usually we really mean "analysis": raw images/spectra -> something scientifically interpretable. For images that usually means "how much light from each object I see".

Let's Narrow the Focus: CCD Images



Raw Data

↓

Overscan

↓

Bias (or Dark)

↓

Flat

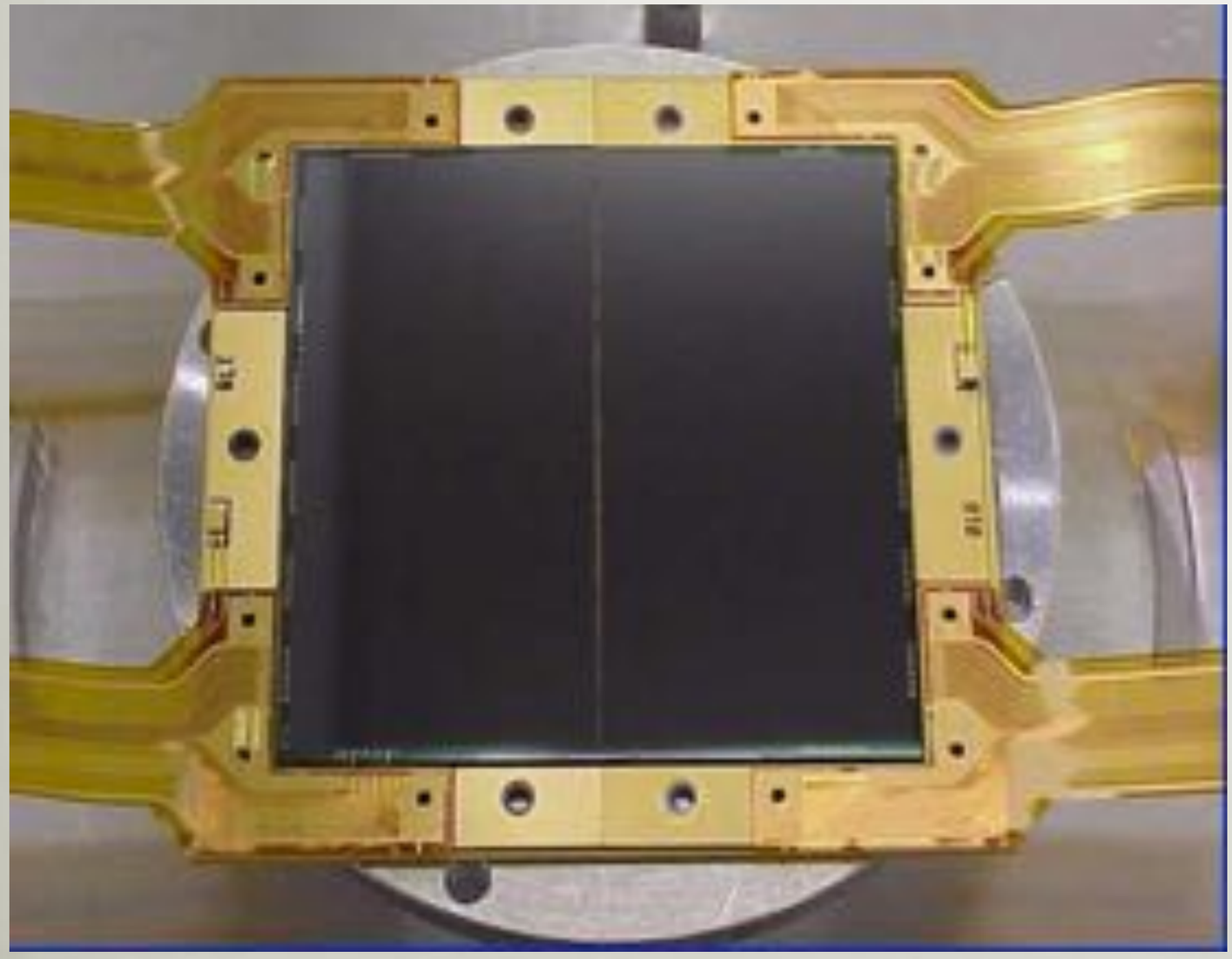
↓

Photometry

↓

Profit! (or Science!)

Let's Narrow the Focus: CCD Images



Raw Data

↓ ccdproc

Overscan

↓ ccdproc

Bias (or Dark)

↓ ccdproc

Flat

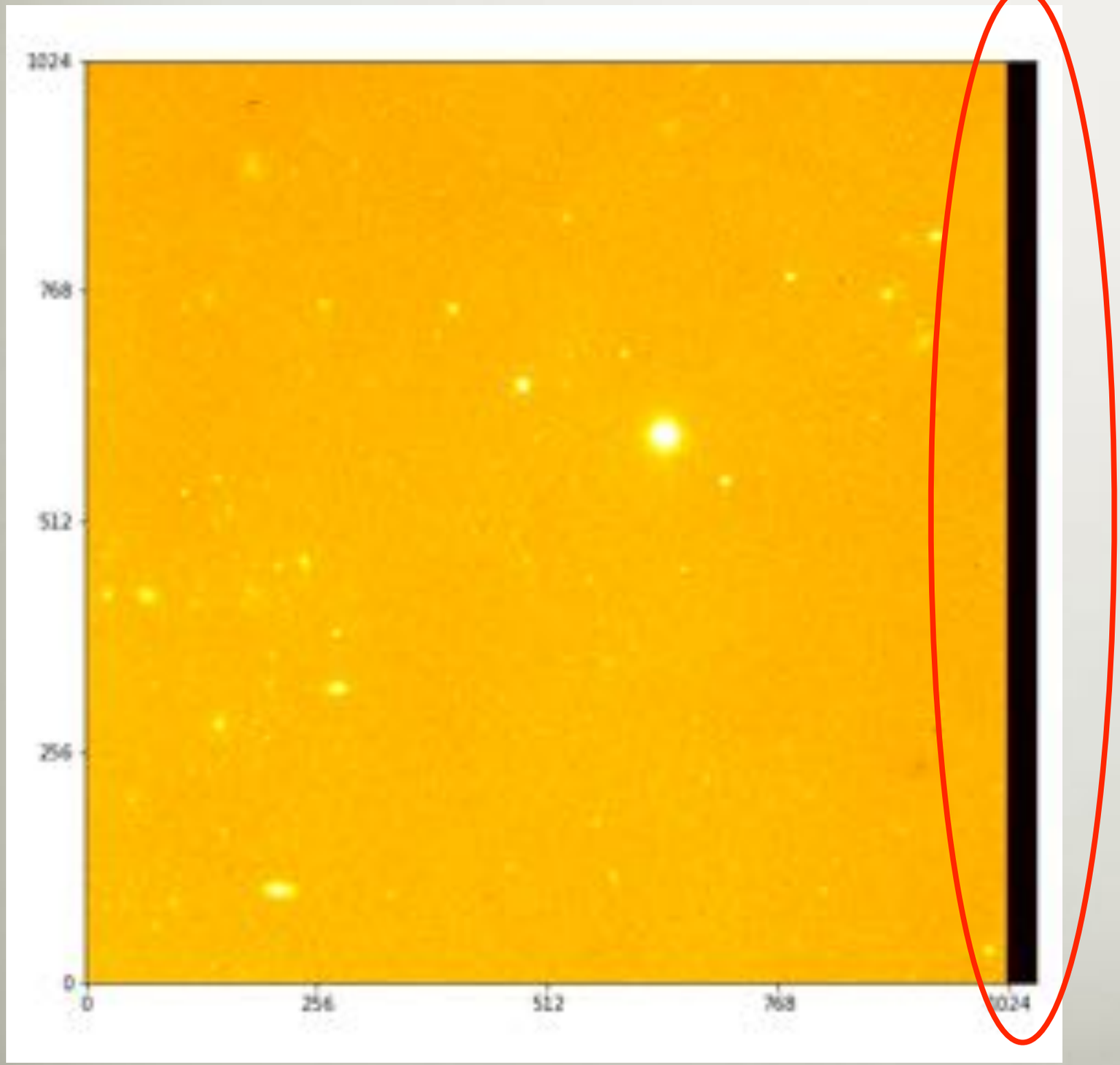
↓ ccdproc

Photometry

↓ photutils

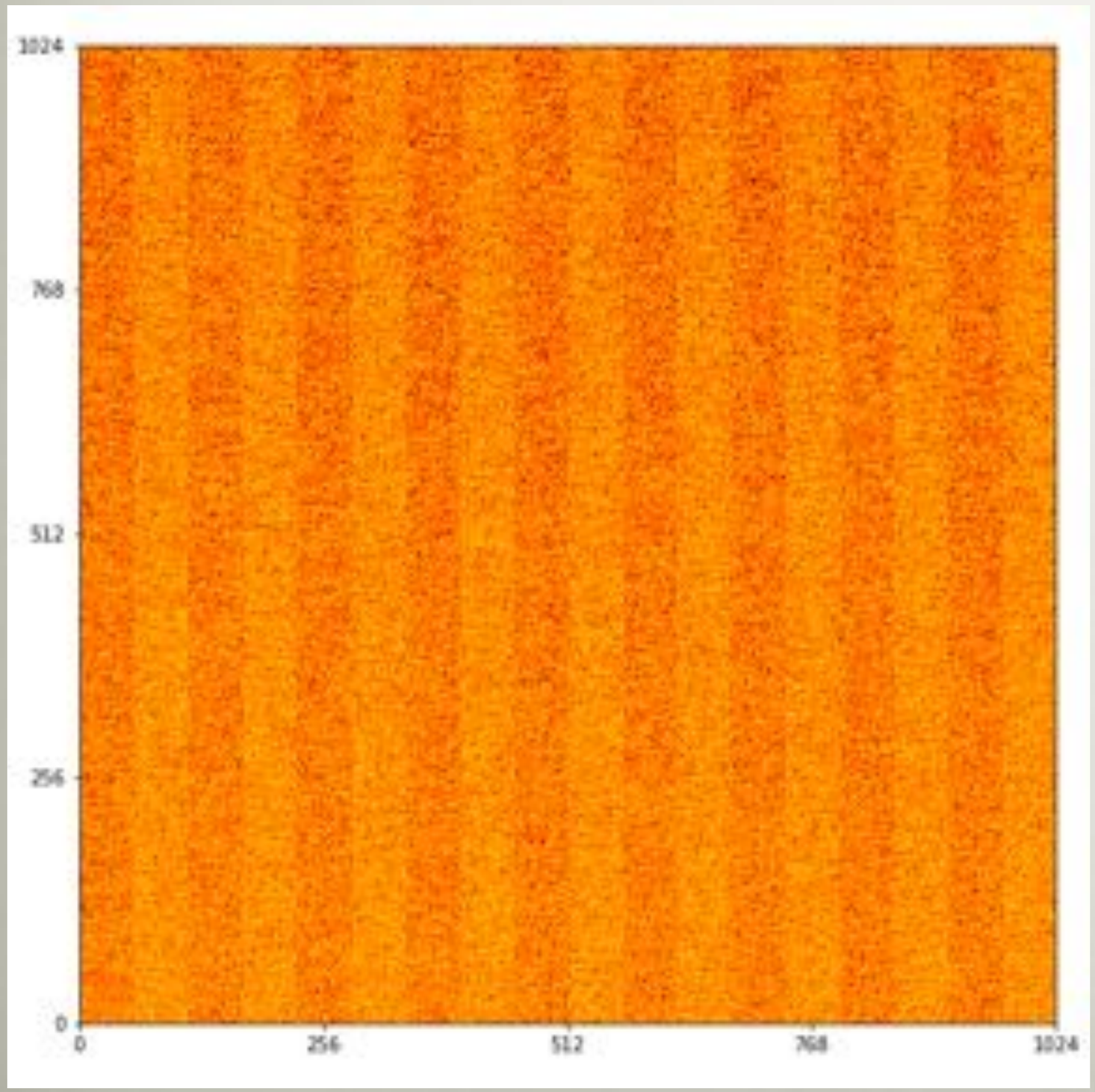
Profit! (or Science!)

Overscan



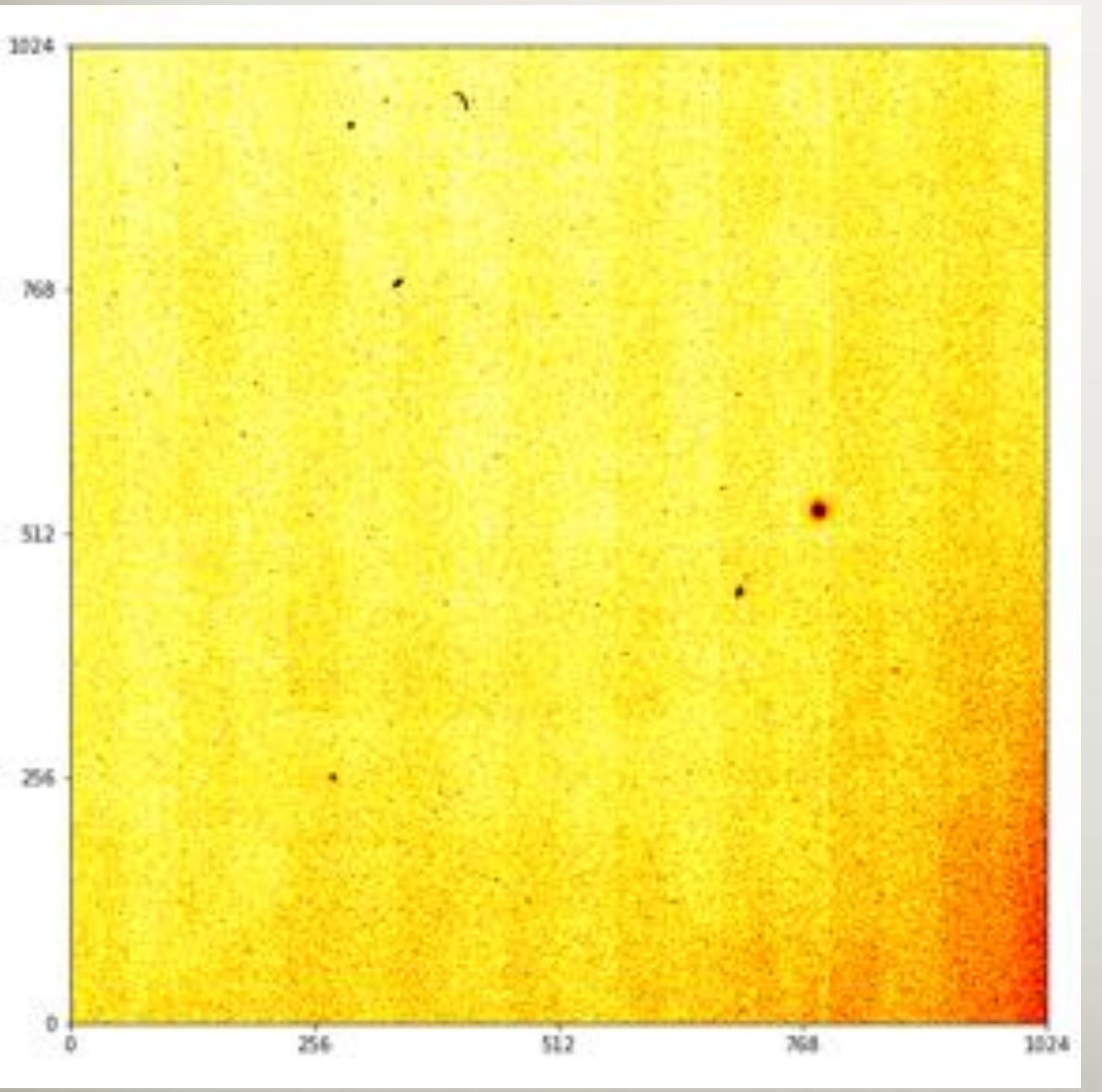
- A “fake” part of the image where the readout electronics were run past the end of the chip.
- Subtracting this removes the column-by-column bias.

Bias



- An unexposed image is required.
- Catches pixel-by-pixel bias variation.
- Sometimes replaces overscan, but otherwise must be overscan-corrected.
- Sometimes “darks” used also/instead

Flats

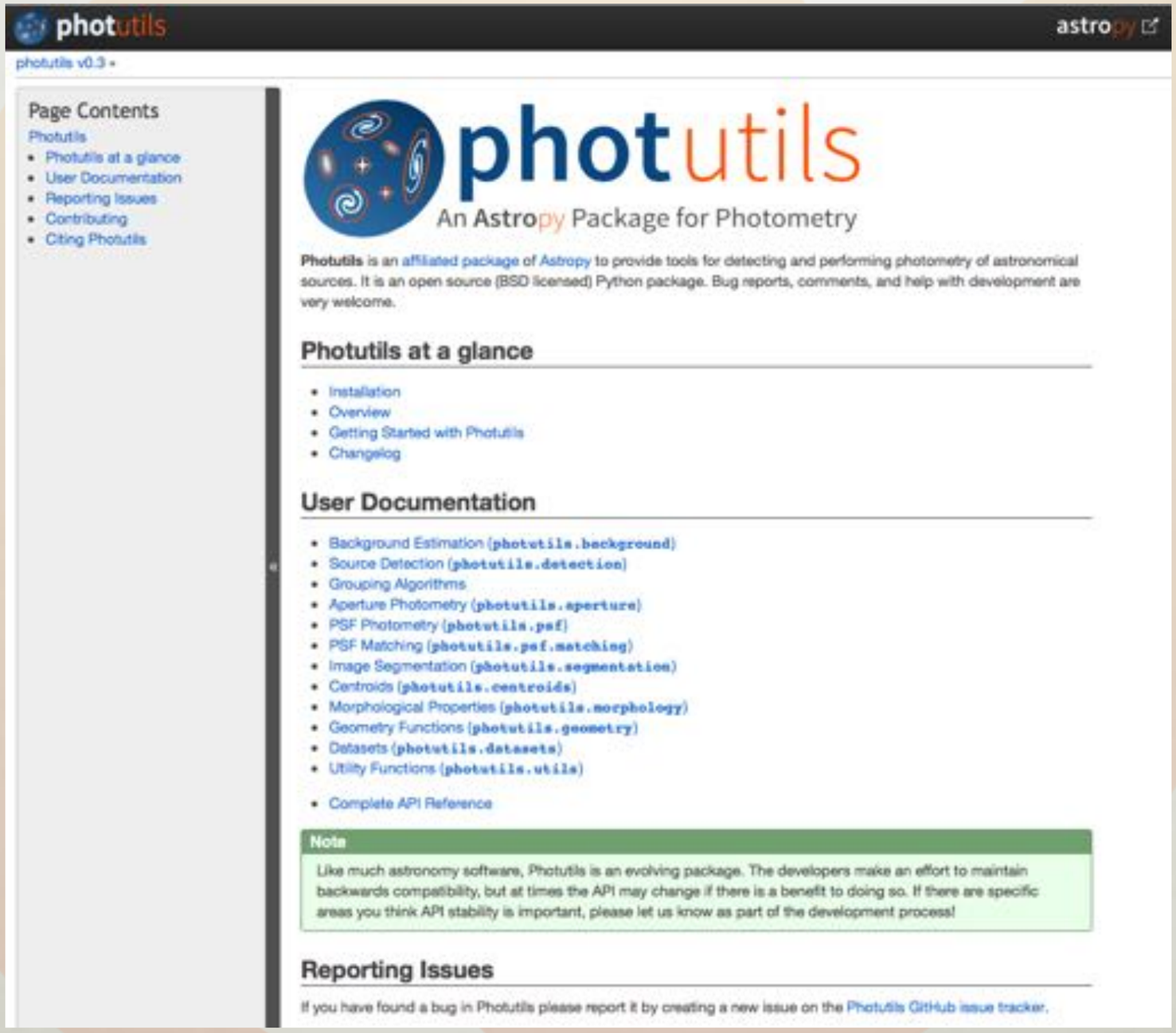


- An exposure of a lit (often de-focused) screen is required.
- Removes optical and pixel-to-pixel sensitivity artifacts.
- Filter-dependent
- Must have better S/N than science exposures!

Photometry

- The aforementioned corrections yield an image that *should* be proportional to the light actually coming from the sky.
- To truly reduce the data to a table of "how much light is coming from all the things", need to measure the "total" from each object from that image.
- Measuring light: "photometry"

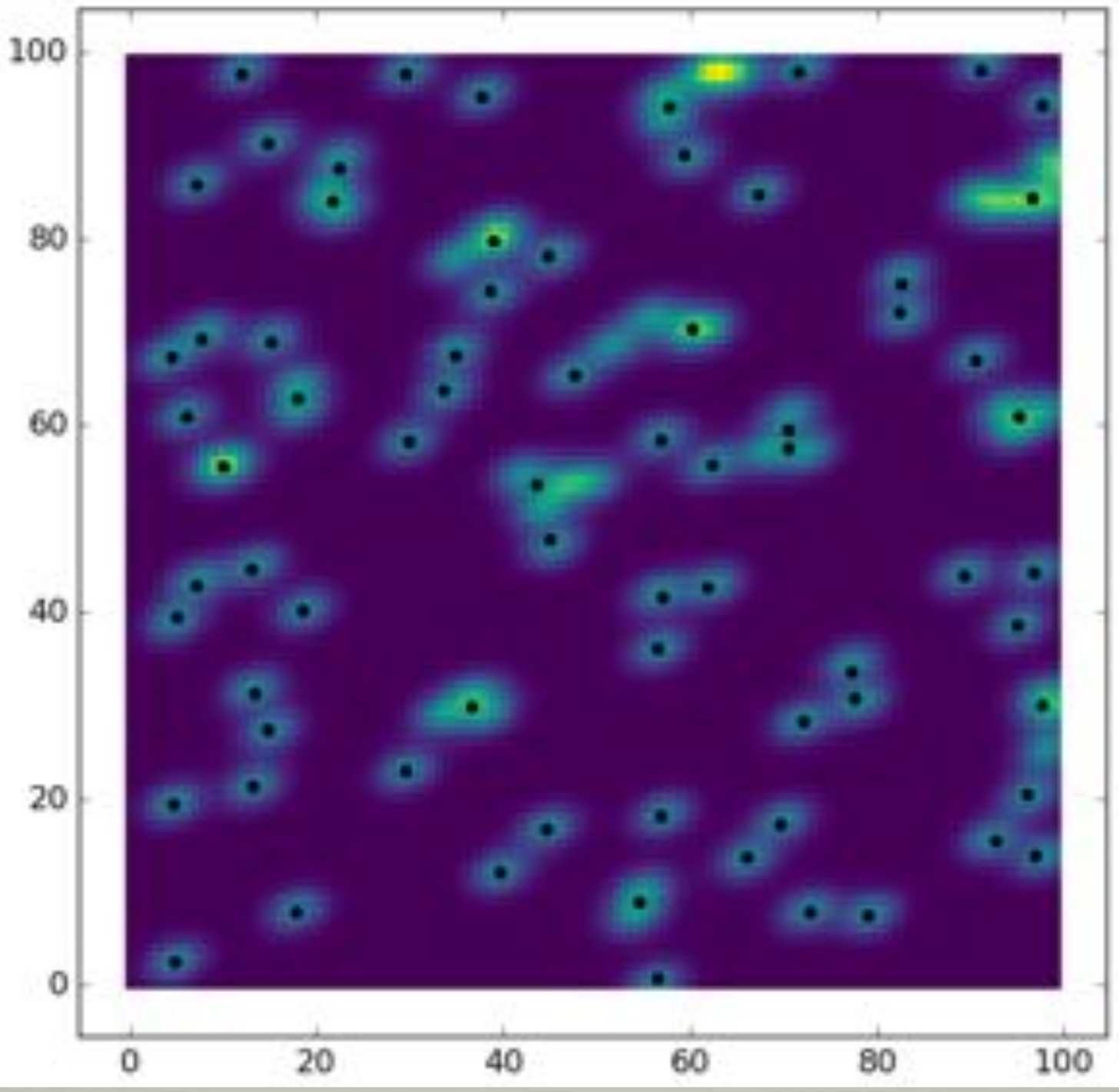
Photutils: Community Photometry Tools



http://photutils.readthedocs.io

Photutils: Object-Finding

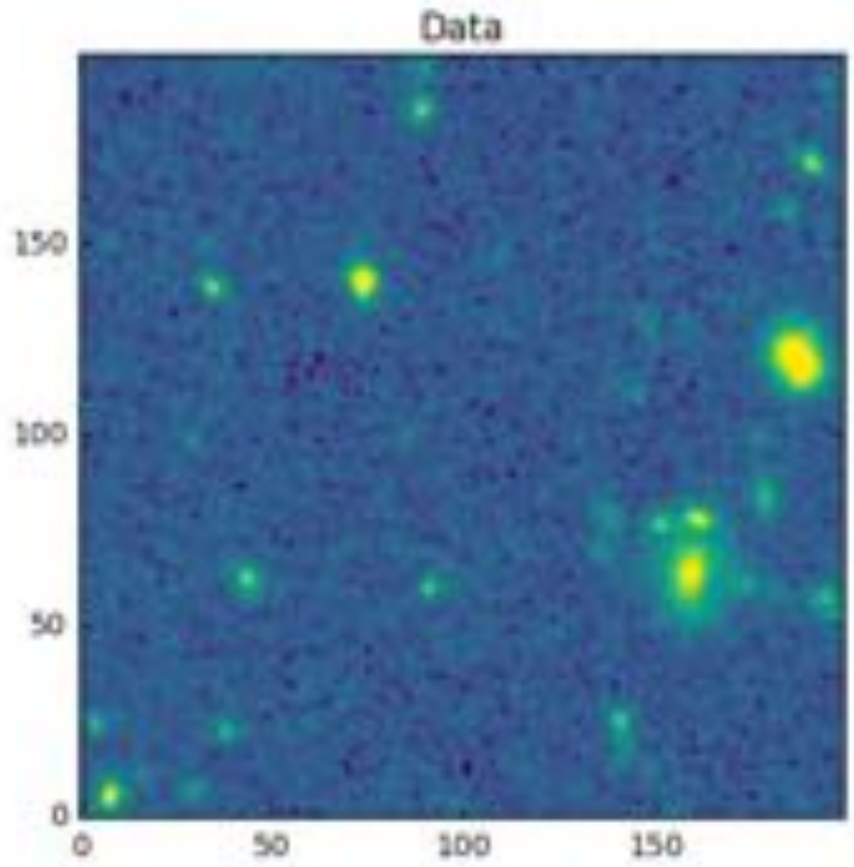
```
In [21]: star_finder = photutils.findstars.DAOStarFinder(threshold=bkg_var/2,  fwhm=5)
found_stars = star_finder(im)

plt.imshow(im)
plt.scatter(found_stars['xcentroid'], found_stars['ycentroid'], color='k')
None
```

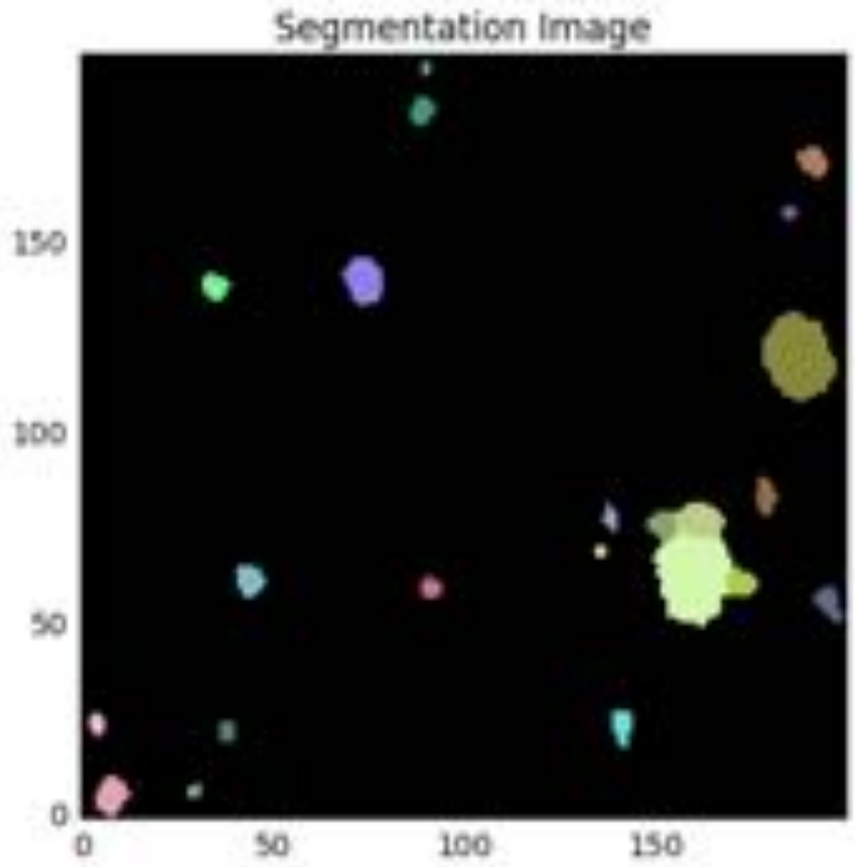


Photutils: Object-Finding

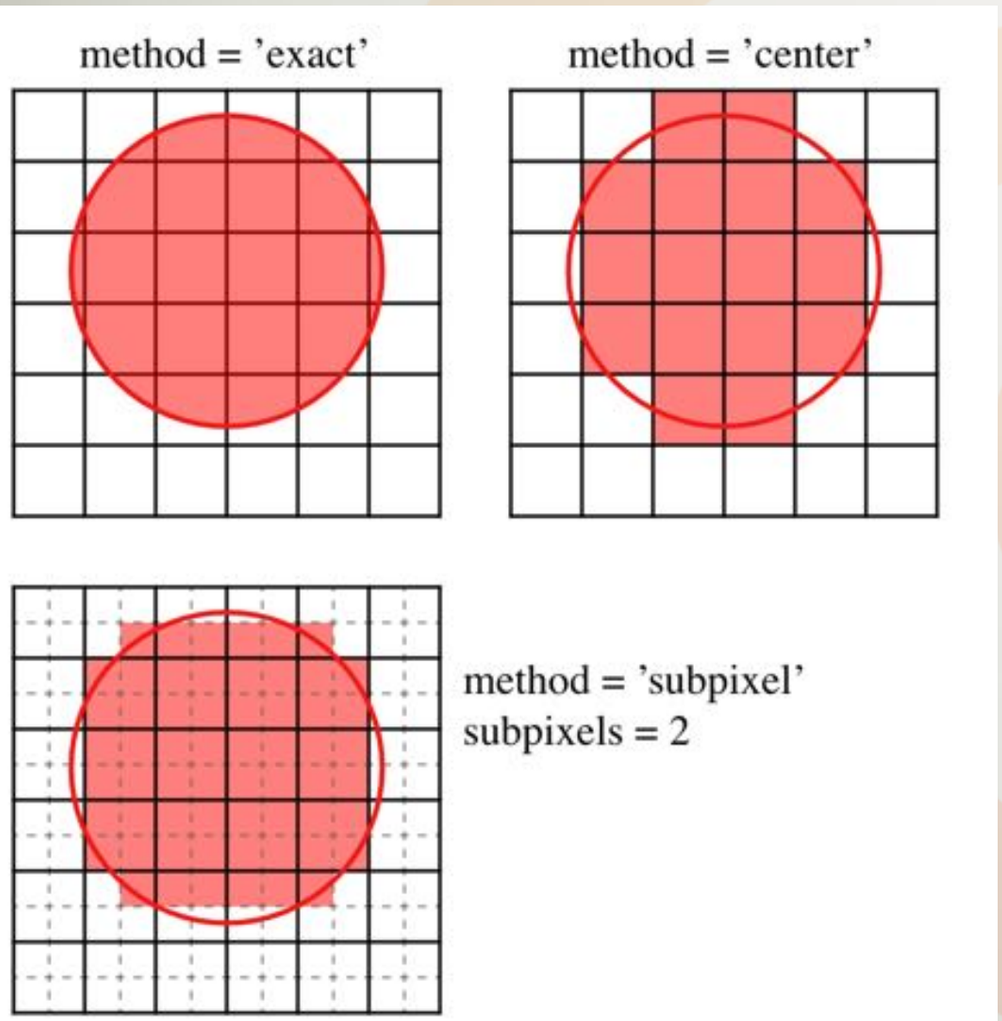
In [61]:
```
from photutils import deblend_sources

segm2 = deblend_sources(data, segm, npixels, filter_kernel=kernel,
                        contrast=0.001, nlevels=32)

fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(10, 8))
ax1.imshow(scale_image(data, scale='sqrt', percent=99.5))
ax1.set_title('Data')
ax2.imshow(segm2, cmap=rand_cmap)
ax2.set_title('Segmentation Image')

print('Found {0} sources'.format(segm2.max))
```

```
Found 22 sources
```





Photutils:
Aperture Photometry



In [41]:
```
from photutils import CircularAnnulus

positions = [(90.73, 59.43), (73.63, 139.41), (43.62, 61.63)]

aper = CircularAperture(positions, r=3)
bkg_aper = CircularAnnulus(positions, r_in=10., r_out=15.)
apers = [aper, bkg_aper]
```

Now, perform the photometry.

In [42]:
```
phot = aperture_photometry(data, apers)
phot.rename_column('aperture_sum_0', 'aperture_sum')
phot.rename_column('aperture_sum_1', 'annulus_sum')
phot
```

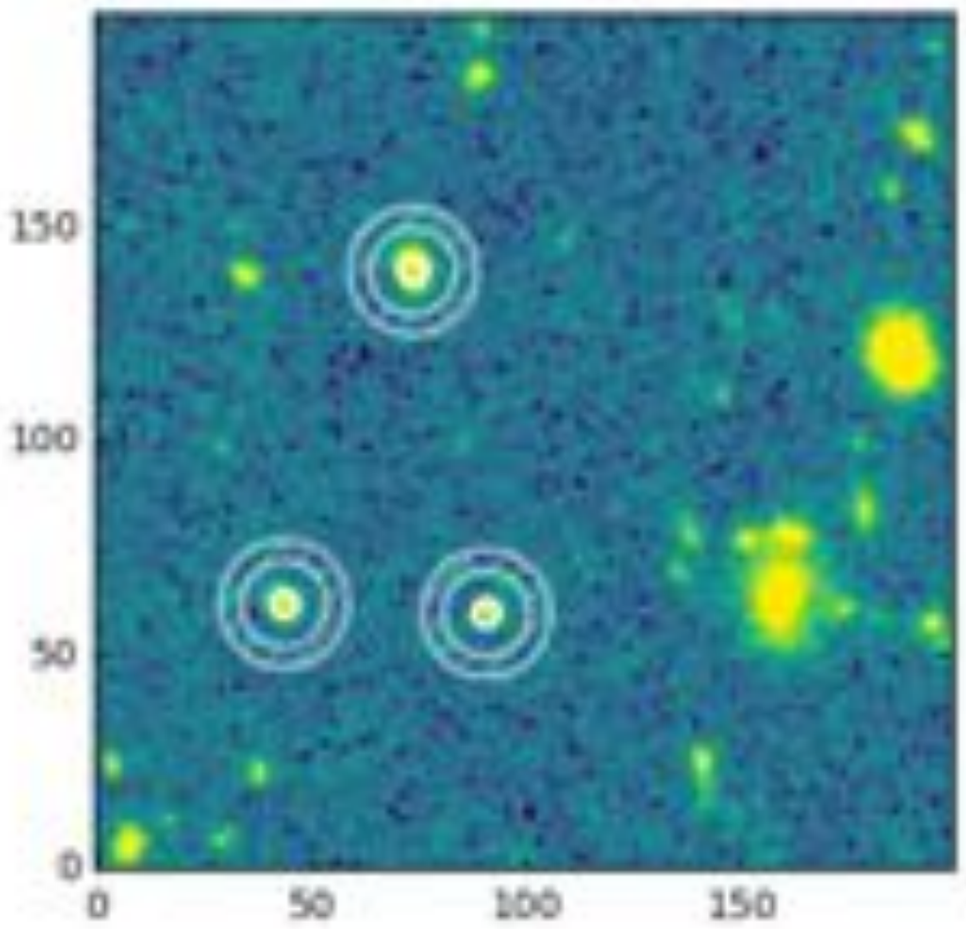
Out[42]: <QTable length=3>

id	xcenter	ycenter	aperture_sum	annulus_sum
	pix	pix		
int64	float64	float64	float64	float64
1	90.73	59.43	0.0866436609693	0.0199107563833
2	73.63	139.41	0.390646538117	0.0358905305285
3	43.62	61.63	0.130109734904	0.0166684757391

Photutils: Aperture Photometry

Apertures can plot themselves

In [46]:
```
plt.imshow(scale_image(data, scale='sqrt', percent=98.))

aper.plot(color='white', lw=2)
bkg_aper.plot(color='white', lw=2, hatch='//', alpha=0.5)
```



Photutils: PSF Photometry

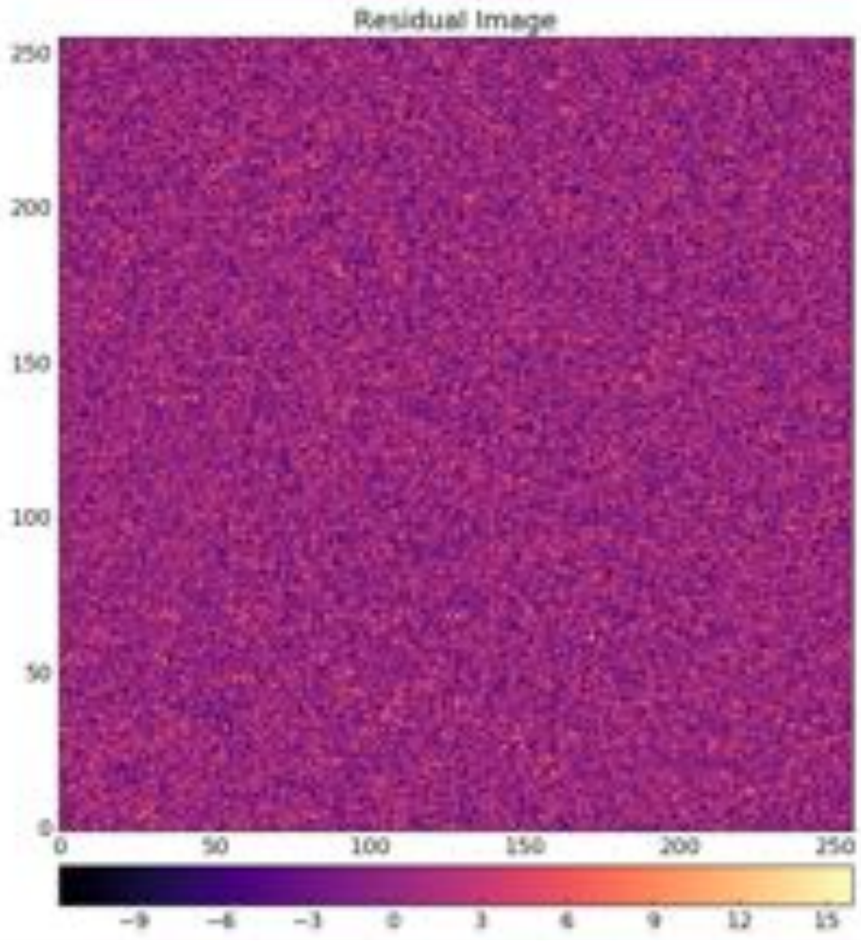
In [20]:
```
from astropy.table import Table
positions = Table(names=['x_0', 'y_0'], data=[starlist['x_mean'], starlist['y_mean']])
```

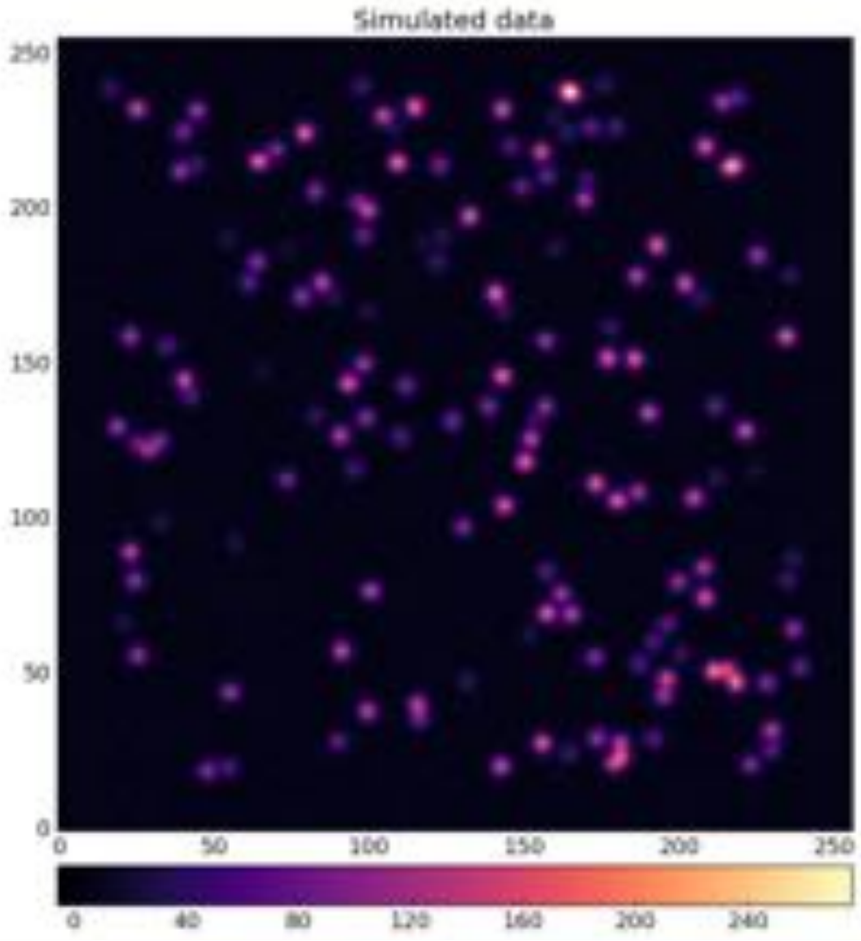
In [21]:
```
photometry_results = basic_photometry(image=image, positions=positions)
```

```
WARNING: Both positions and finder are different than None, which is ambiguous. finder is going to be ignored. [photu
tils.psf.photometry]
```

In [22]:
```
plt.subplot(1,2,1)
plt.imshow(basic_photometry.get_residual_image())
plt.title('Residual Image')
plt.colorbar(orientation='horizontal', fraction=0.046, pad=0.04)

plt.subplot(1,2,2)
plt.imshow(image)
plt.title('Simulated data')
plt.colorbar(orientation='horizontal', fraction=0.046, pad=0.04)
```

Out[22]:
```
<matplotlib.colorbar.Colorbar at 0x11488be48>
```





Now go do it yourself!

LSSTC-DSFP-Sessions/Session5/Day1/python_imred.ipynb

Note: substantial downloads required at beginning… may need to share the files if you can.