

BatAnalysis: A Comprehensive Python Pipeline for Swift BAT Data Analysis

Tyler Parsotan & BAT Team

Outline

- Introduction to Swift and BAT
- BAT Survey Data
 - Overview of the data
 - Past Analyses
 - Usefulness
- The BatAnalysis Python Package
 - How to analyze BAT survey data
 - Verifying with the Crab Nebula Pulsar
- Example Analyses of:
 - NGC 2992
 - MAXI J0637-430
 - GRB 221009A

The Neil Gehrels Swift Observatory

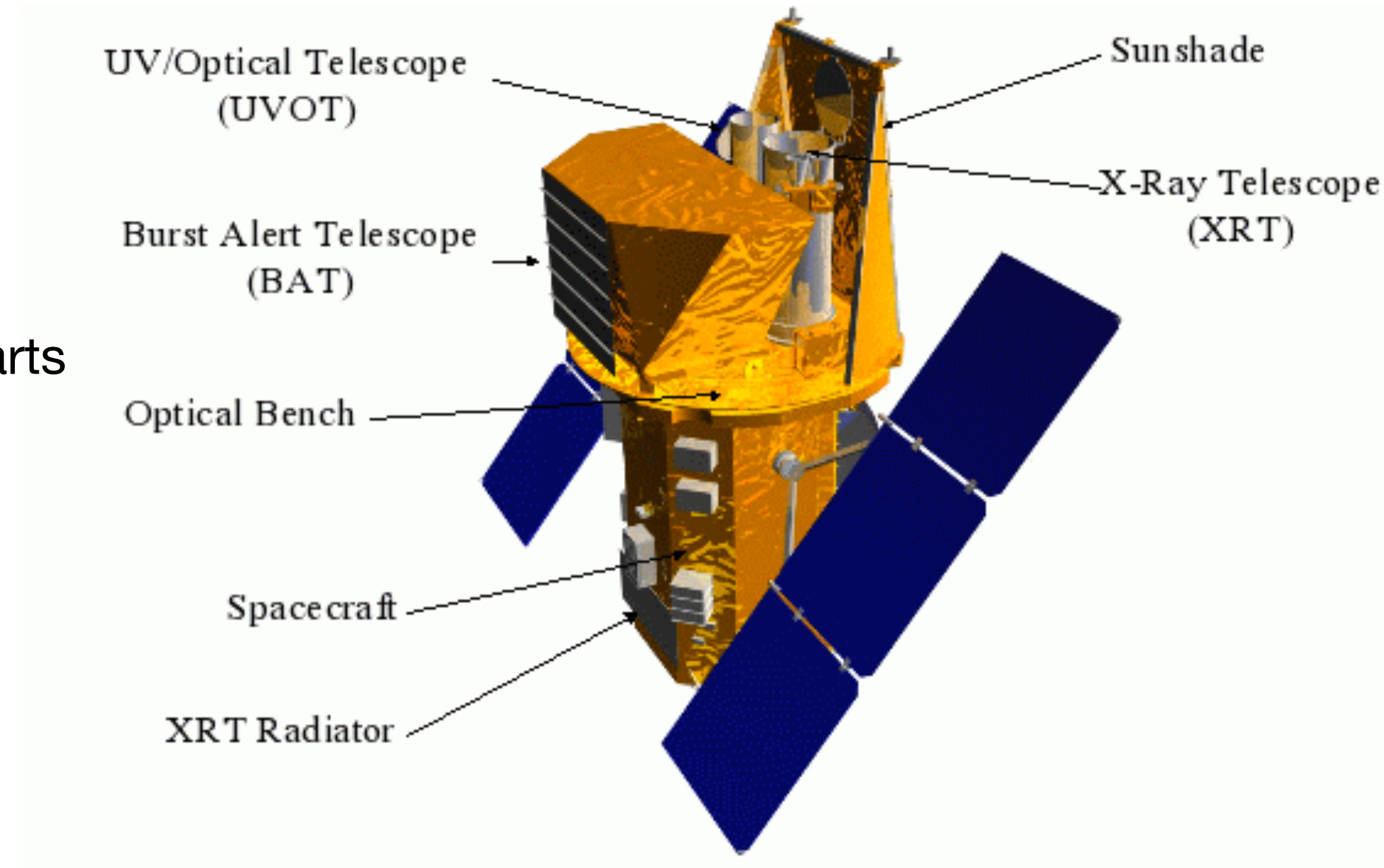
- Launched in 2004
- Overview of operations:
 1. BAT detects a Gamma Ray Burst (GRB)
 2. Autonomous slewing to the GRB
 3. XRT and UVOT observe the field to detect afterglow counterparts

OR

 1. There is a ToO/many point plan for XRT/UVOT to observe a source
 2. BAT is pointed towards that source too collecting survey data

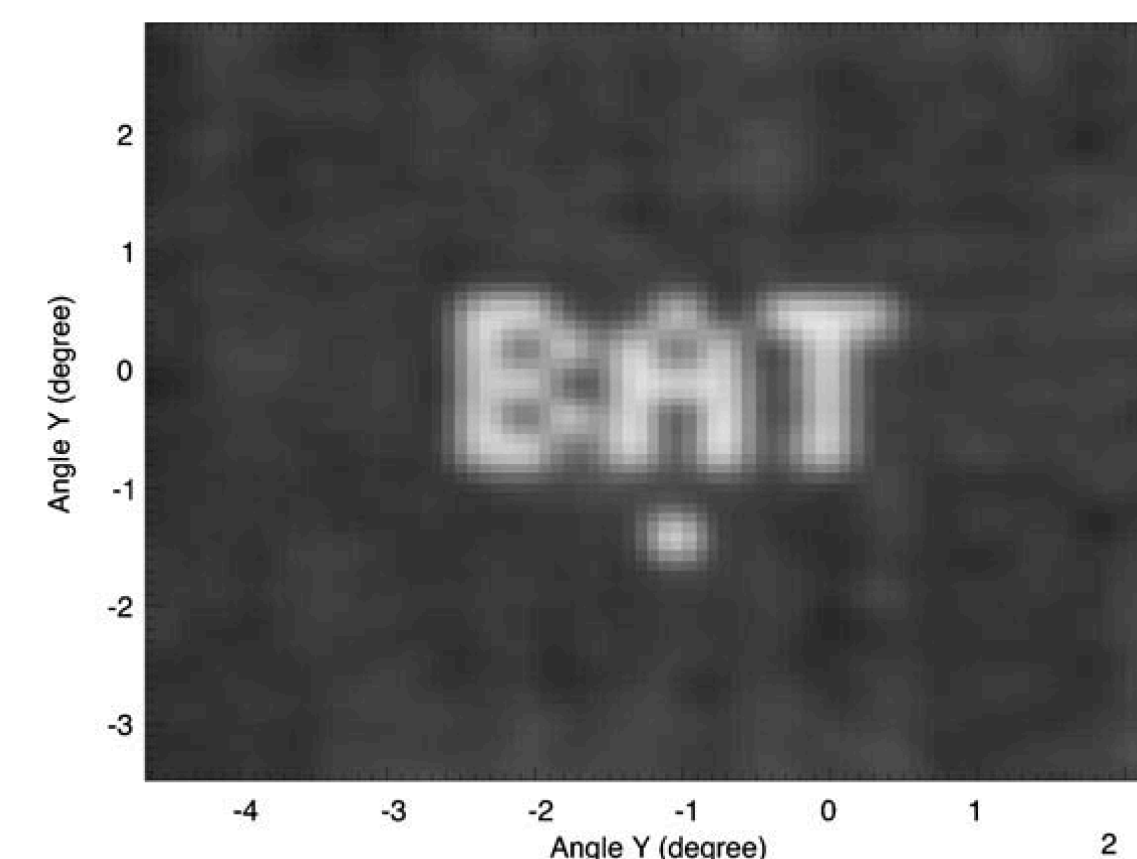
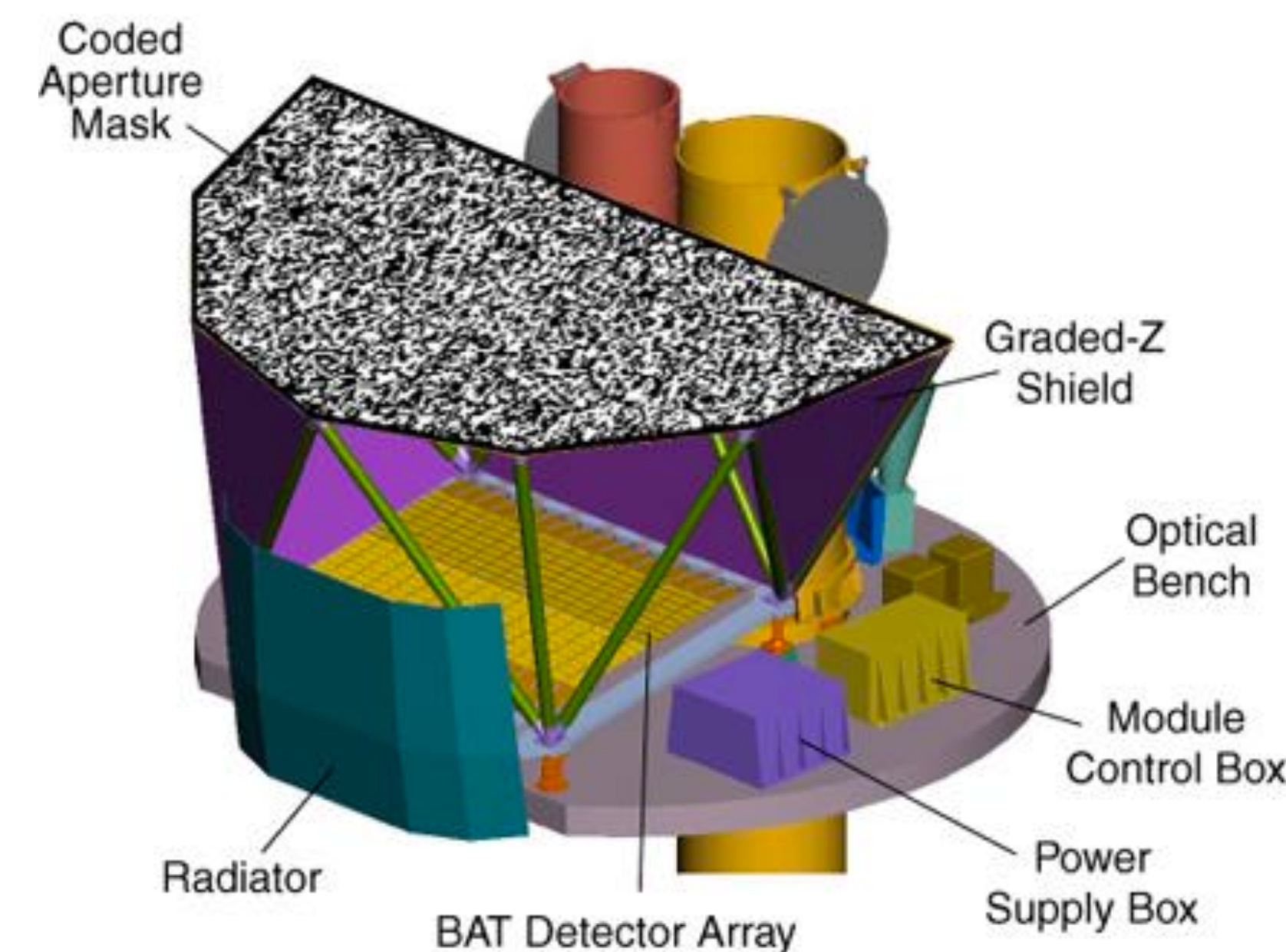
OR

 1. BAT is surveying the sky normally, collecting survey data in between slews
- Over 1500 GRBs have been detected



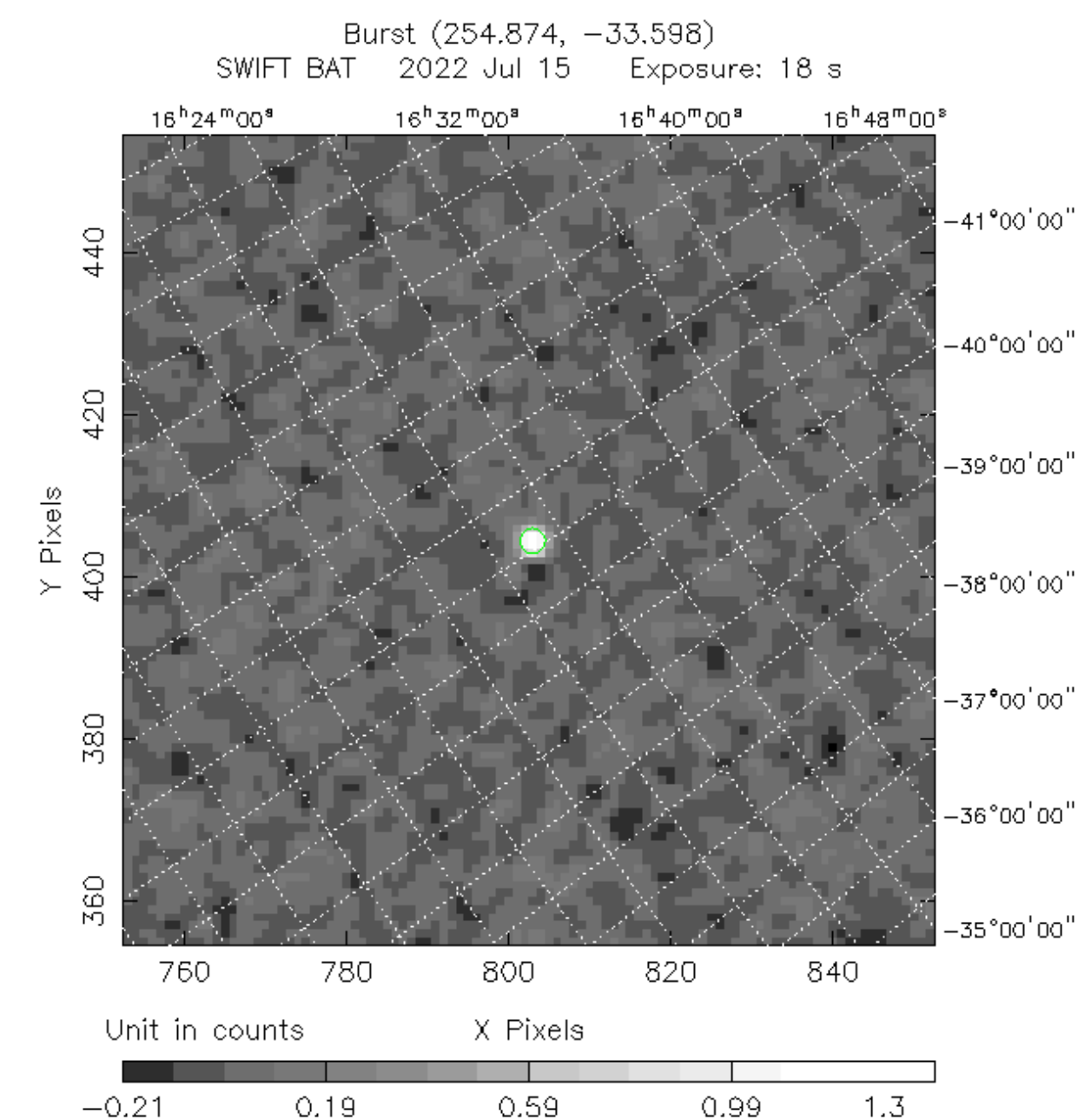
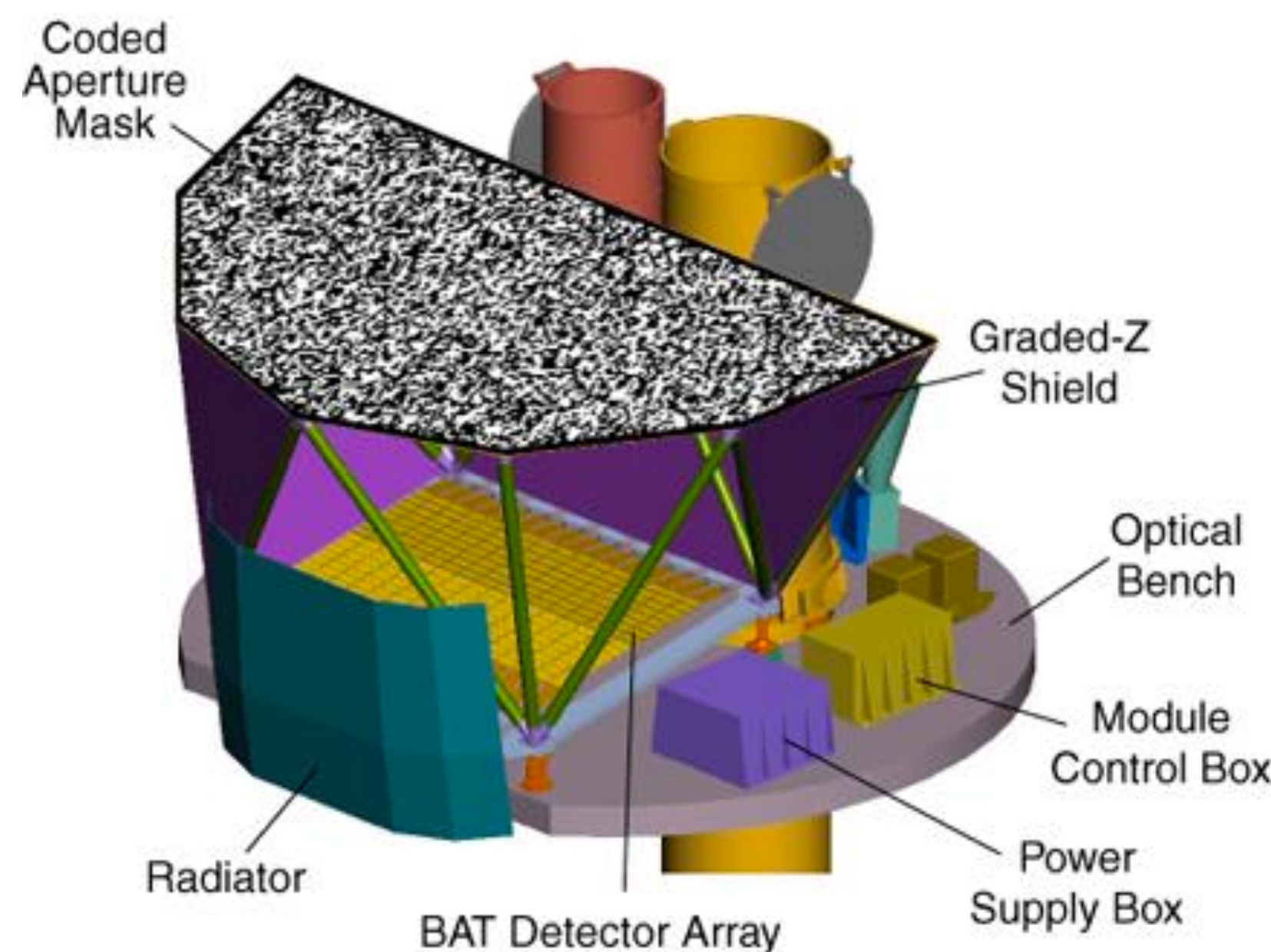
The BAT

- Uses coded mask techniques to maximize:
 - FOV ($\sim 60 \times 120^\circ$)
 - Localization Capabilities of GRBs (~ 3 arcmin)
- GRBs \rightarrow time tagged event (TTE) data
 - The highest quality data of each photon's direction and energy
 - Intensive to store and transfer to the ground
- Event data can be used to localize transients even if they do not trigger BAT or are located outside of the BAT FOV



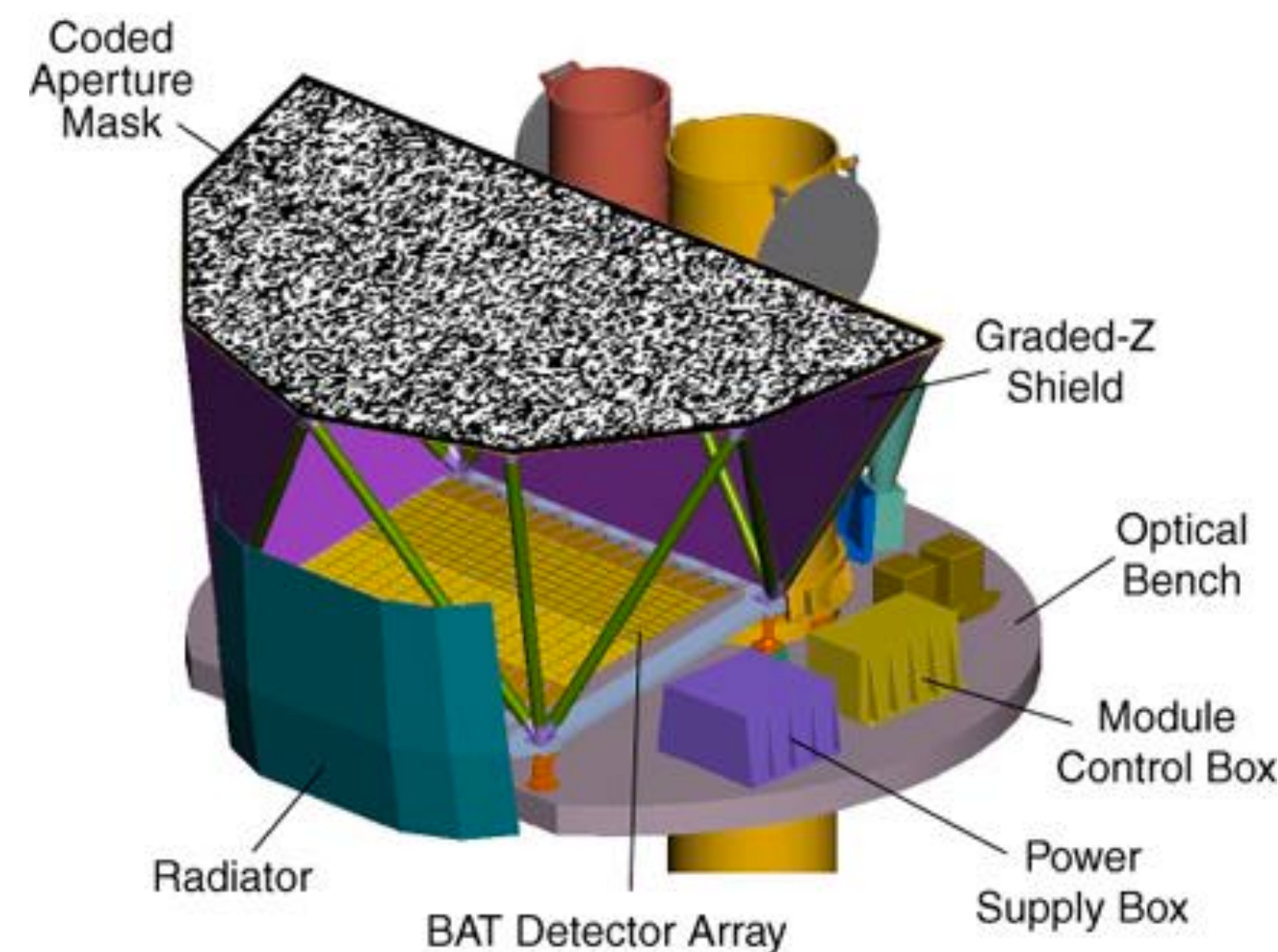
The BAT

- Uses coded mask techniques to maximize:
 - FOV ($\sim 60 \times 120^\circ$)
 - Localization Capabilities of GRBs (~ 3 arcmin)
- GRBs \rightarrow time tagged event (TTE) data
 - The highest quality data of each photon's direction and energy
 - Intensive to store and transfer to the ground
- Event data can be used to localize transients even if they do not trigger BAT or are located outside of the BAT FOV



The BAT

- When its not triggering on GRBs BAT is surveying the sky in the 14-195 keV energy range
- Cannot store all the event data, therefore we compress it
- Create Detector Plane Histograms (DPHs)
 - 80 channel histograms of photon counts
- Accumulates DPHs in time intervals of ~300 seconds



BAT survey data is >90% of all the data produced by BAT, by volume.

It is the least used and has tons of possibilities

BAT TTE data is collected when:

1) BAT triggers

OR

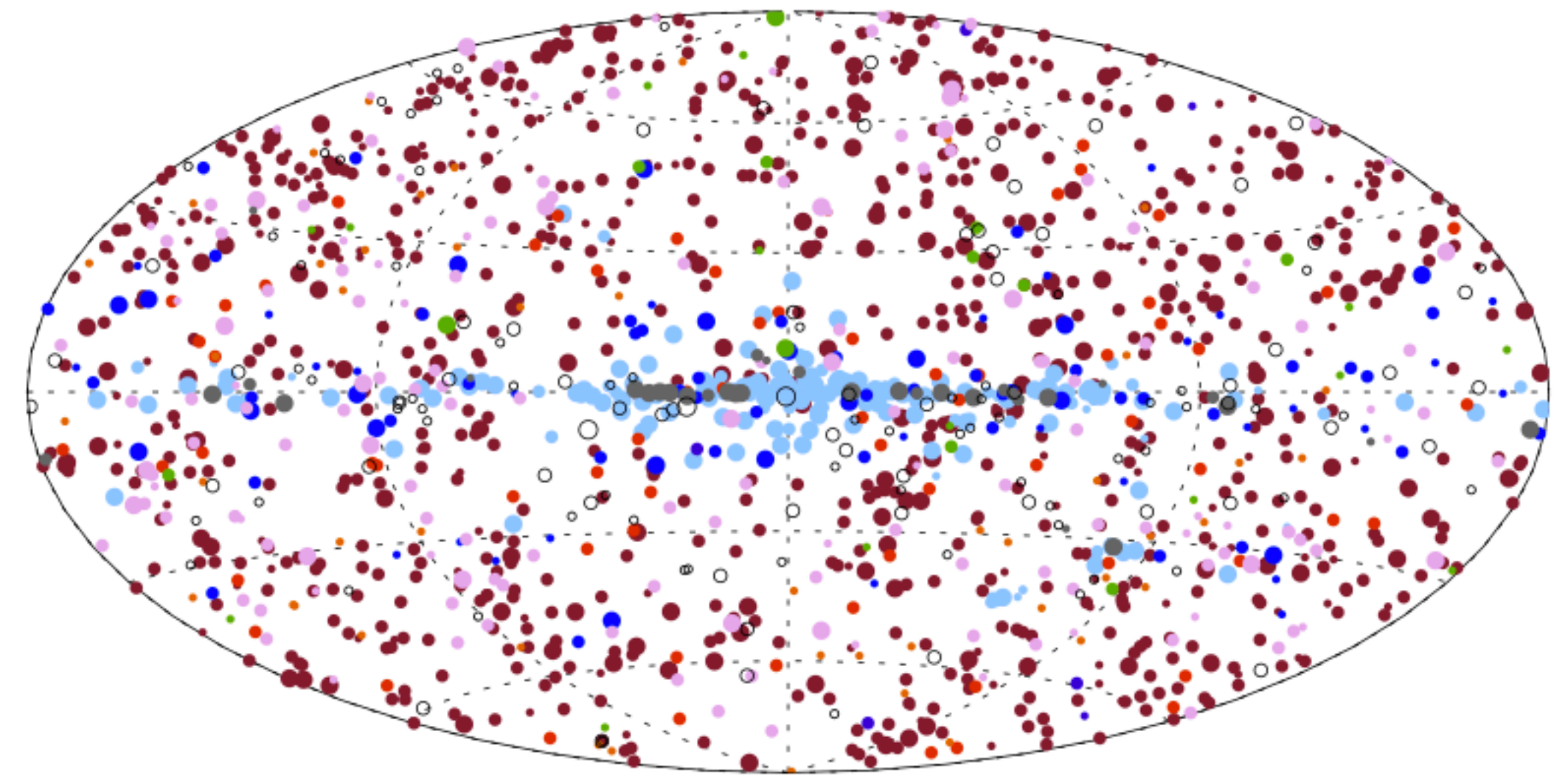
2) when there is an external trigger and we can tell BAT to save the data around that time period

N-Month Survey Catalog

where $N=\{22, 70, 105, 157\}$

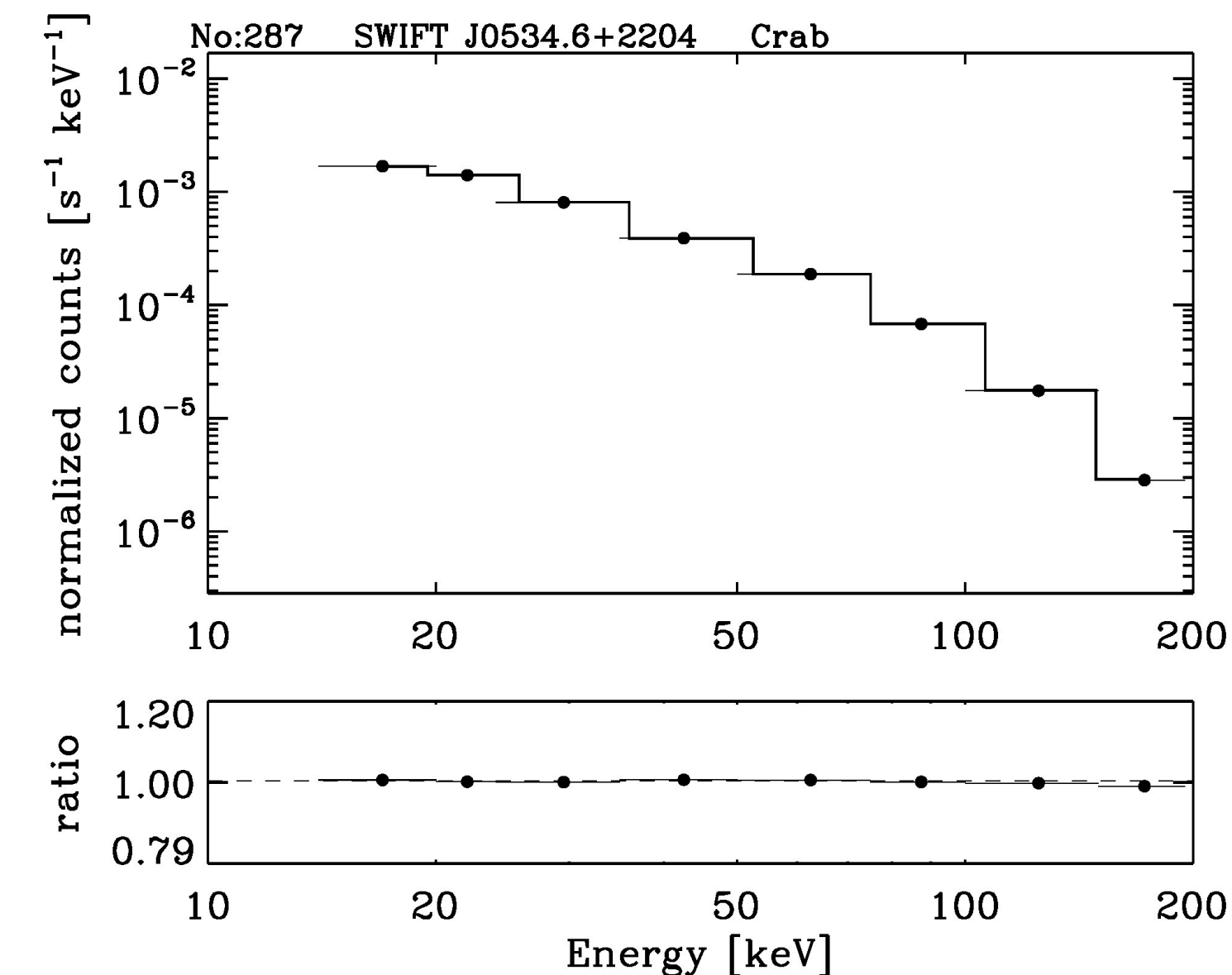
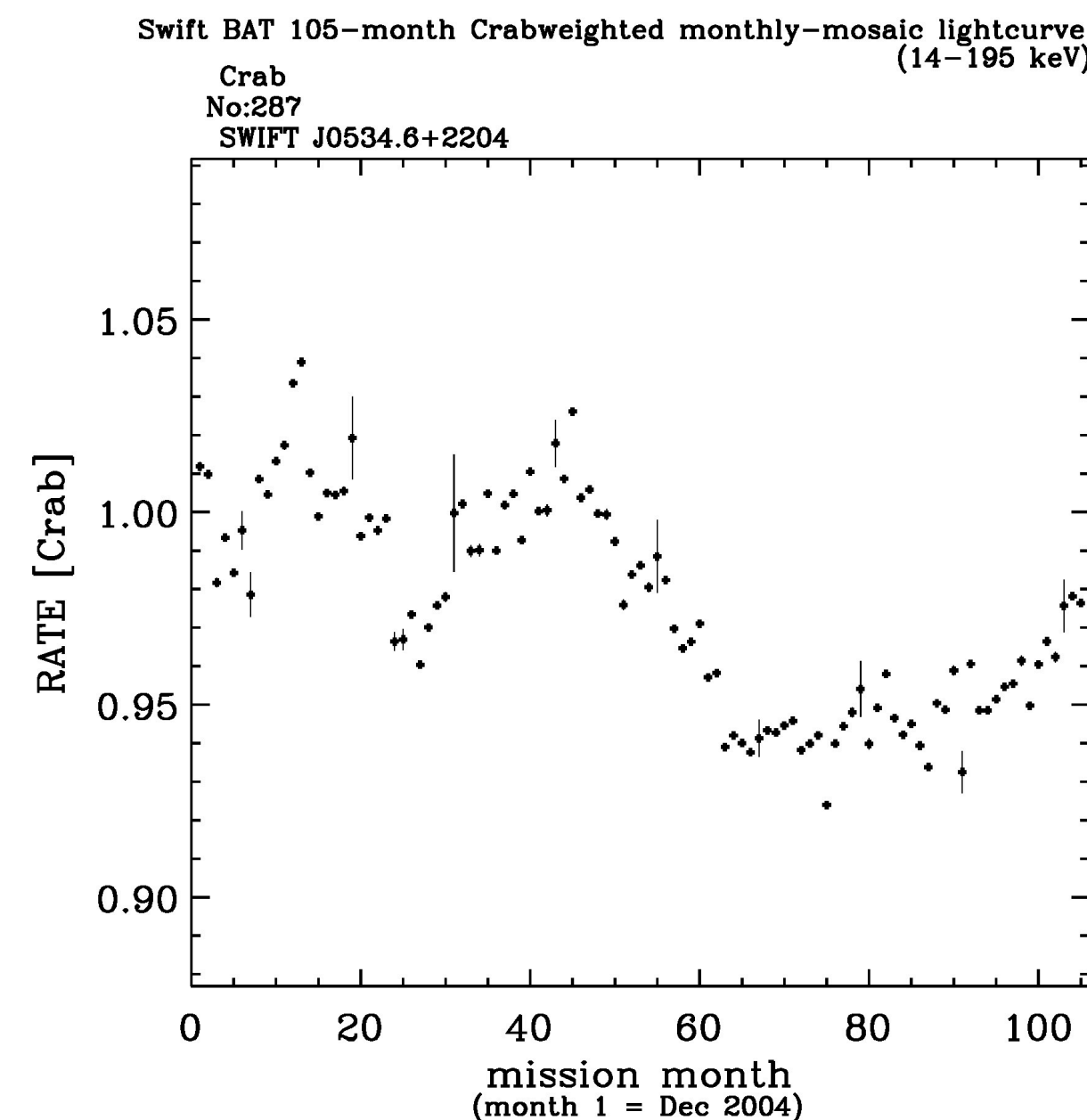
- Systematic analyses of BAT Survey data
- Analyze known sources to produce
 - Monthly mosaiced light curves
 - N-month mosaiced Spectra
- Discover new sources and (attempt to) identify them based on multi-wavelength follow-up

<http://swift.gsfc.nasa.gov/results/bs22mon/>
<http://swift.gsfc.nasa.gov/results/bs70mon/>
<http://swift.gsfc.nasa.gov/results/bs105mon/>



○ Unidentified ● Unknown AGN ● Seyfert Galaxies ● CVs/Stars ● X-ray Binaries
 ● LINER ● Galaxy Clusters ● Beamed AGN ● Pulsars/SNR

There are a total of 1632 sources identified in the 105 month survey where 422 are new detections

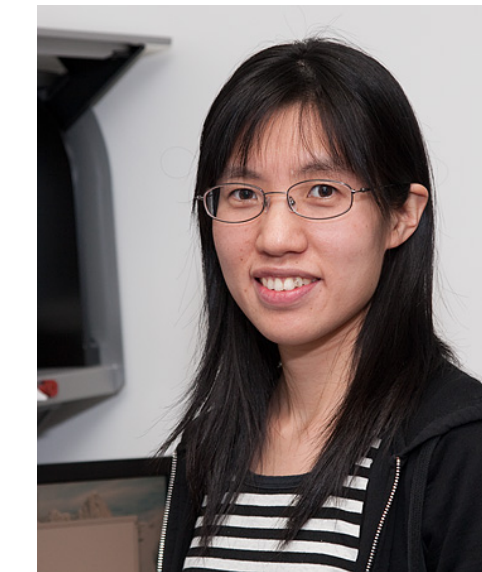


What if we want to analyze a source on a different time scale?

**What if we want to look in archival data to place upper limits
on a newly detected transient?**

The BAT ANALYSIS Package

github.com/parsotat/BatAnalysis

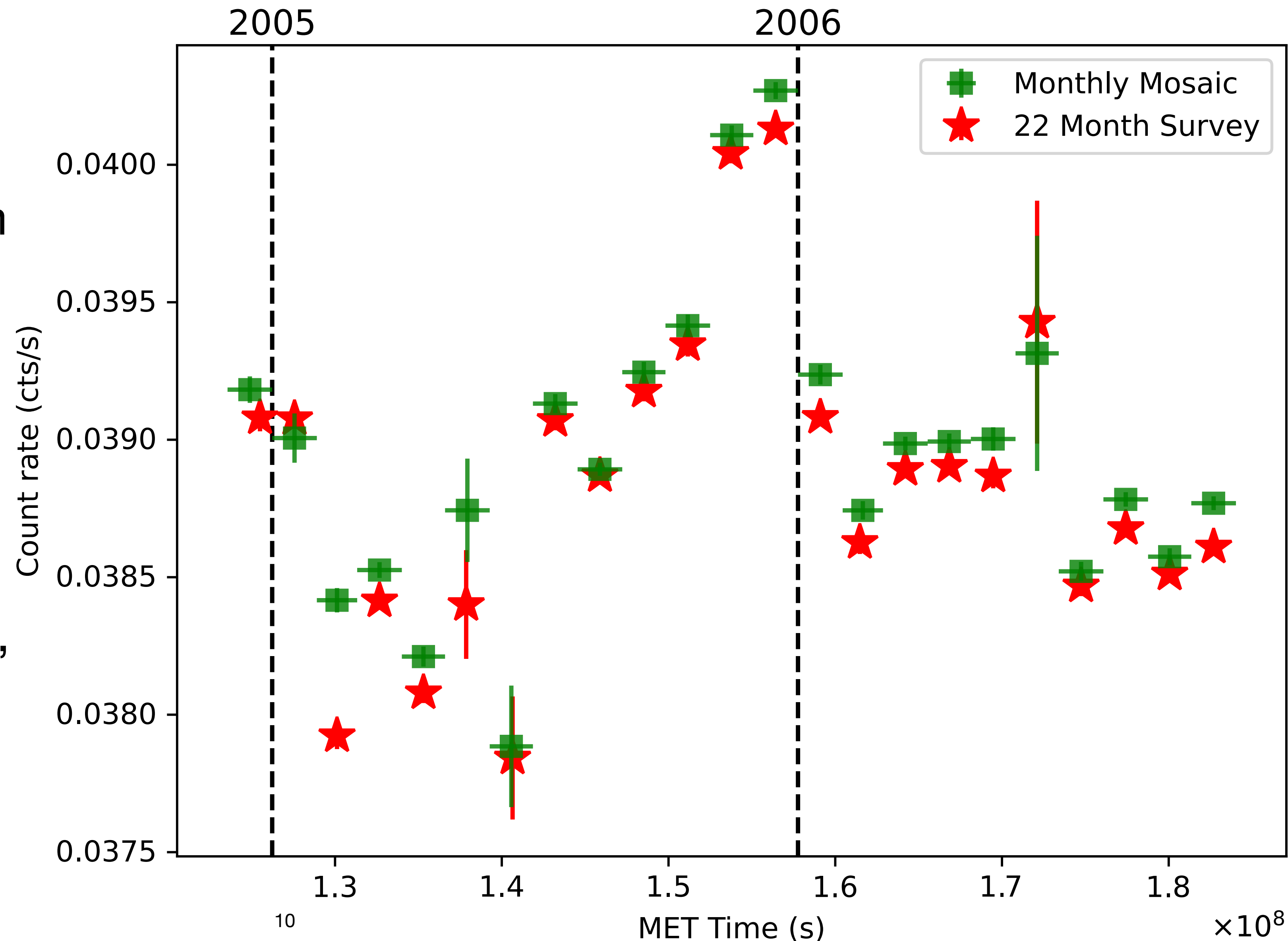


The screenshot shows the GitHub repository page for `parsotat/BatAnalysis`. The repository is public and has 3 watchers, 1 fork, and 0 stars. The main branch is selected, with 2 other branches and 1 tag. A notification indicates that the main branch is not protected. The repository description is "A python HEASOFT wrapper for processing Swift-BAT data." The file list includes `batanalysis`, `notebooks`, `.gitignore`, `README.md`, `requirements.txt`, and `setup.py`. The latest release is `BatAnalysis v0.0.1`, marked as the latest version.

File/Folder	Description	Last Commit
<code>batanalysis</code>	Fixed error with removing facet header value when it no longer exists...	last week
<code>notebooks</code>	Updated NGC script.	yesterday
<code>.gitignore</code>	Initlaized BatAnalysis repo.	3 months ago
<code>README.md</code>	Updated README.	last week
<code>requirements.txt</code>	Added concatenation function to collect all information for all objec...	last month
<code>setup.py</code>	updated setup.py and __init__.py to read version from a file and prep...	2 weeks ago

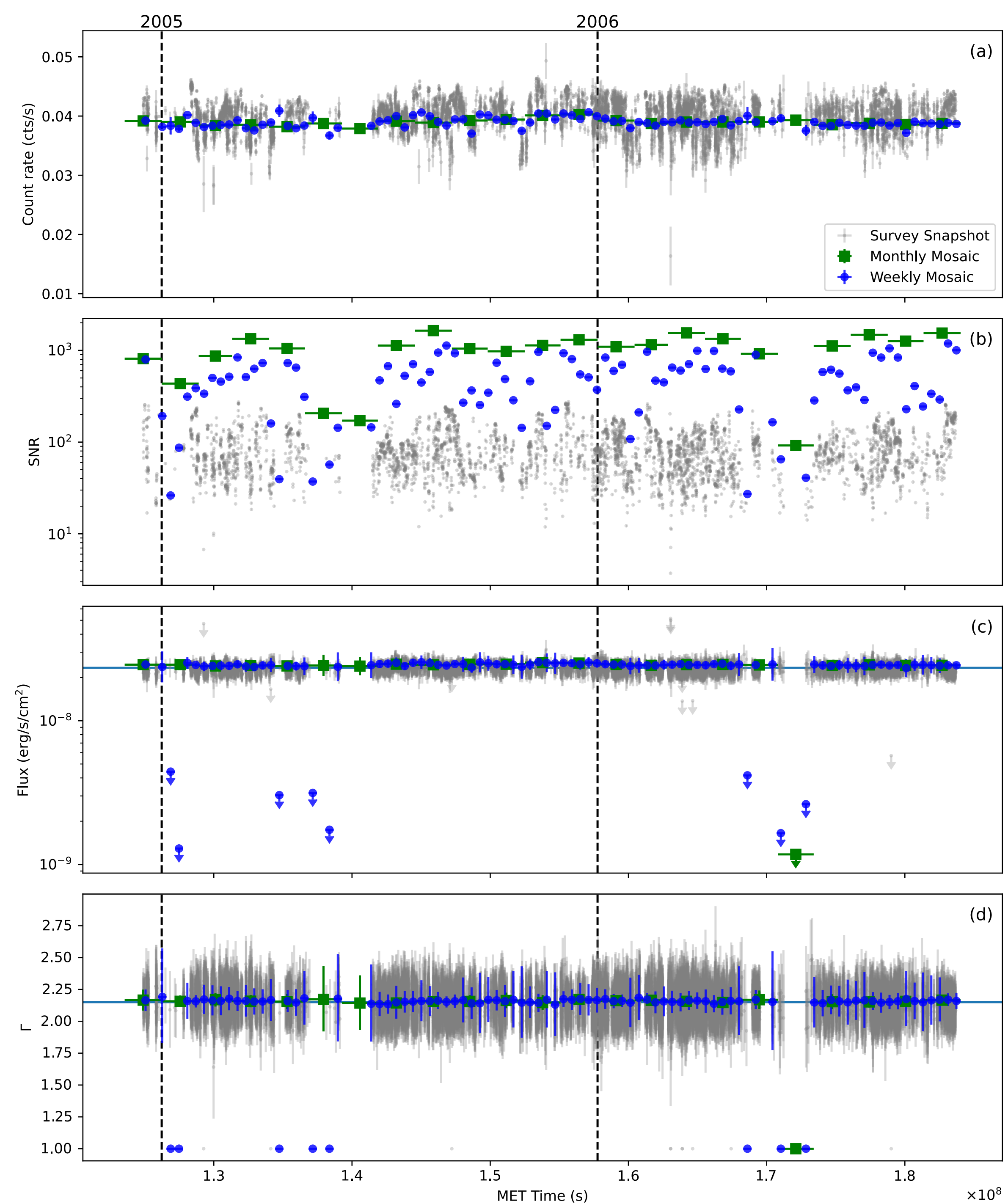
The BatAnalysis Code Reproduces Prior Survey Catalog Results

- 22 month survey:
 - Creates mosaic images from all survey data from Dec 2004 to Oct 2006
- BatAnalysis
 - Queries HEASARC for data
 - Filters data for ones where the Crab is “seen” by 1000 cm^2 of the BAT (~19% partial coding)



The BatAnalysis Code Allows for a More Comprehensive Picture

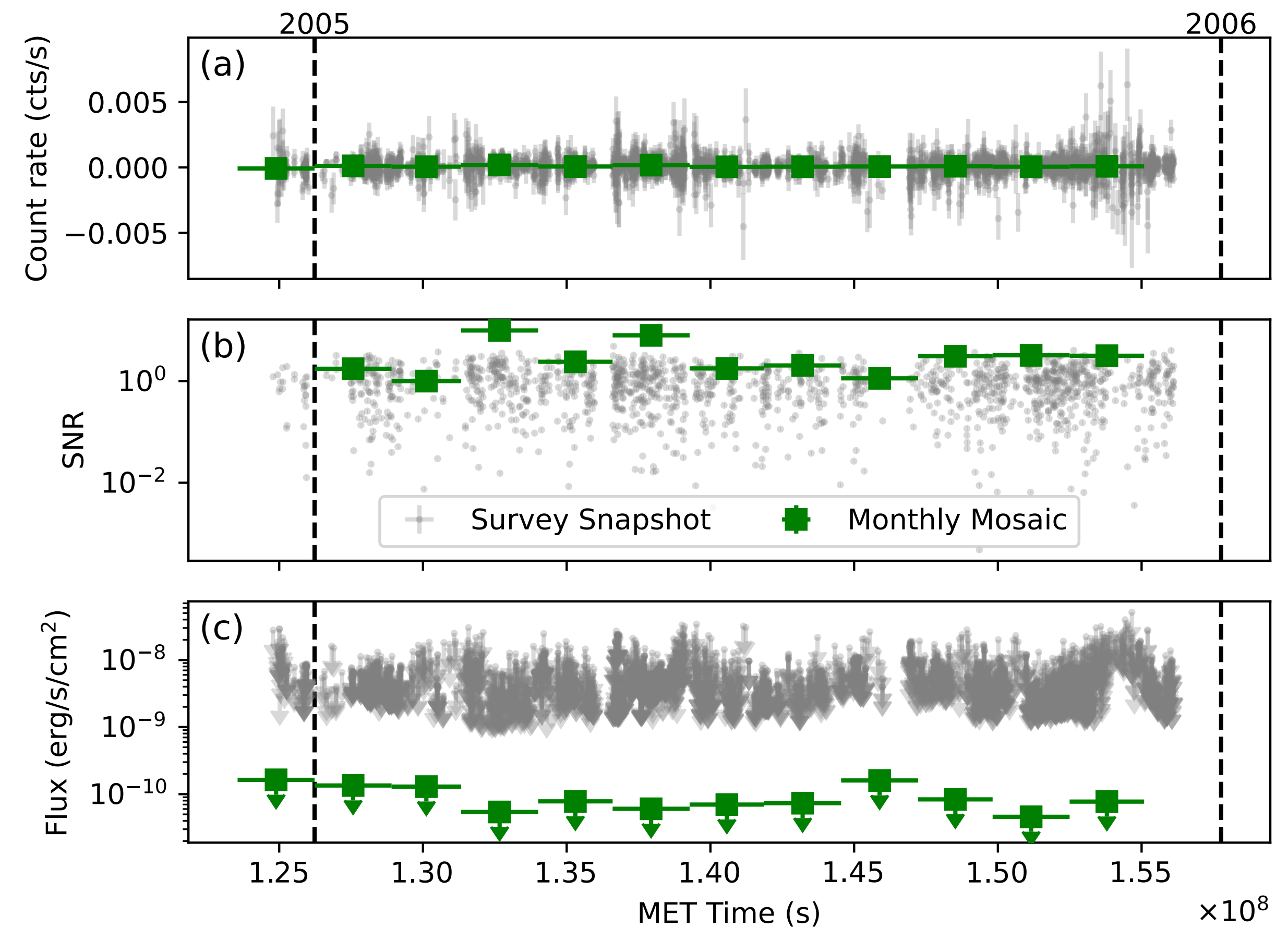
- BatAnalysis
 - Can fit each survey/mosaic spectrum with a simple power law
 - If spectra are not well fit or if a detection is not obtained, the spectra are used to calculate upper limits



BatAnalysis Allows for Custom Mosaicing

AGN NGC 2992

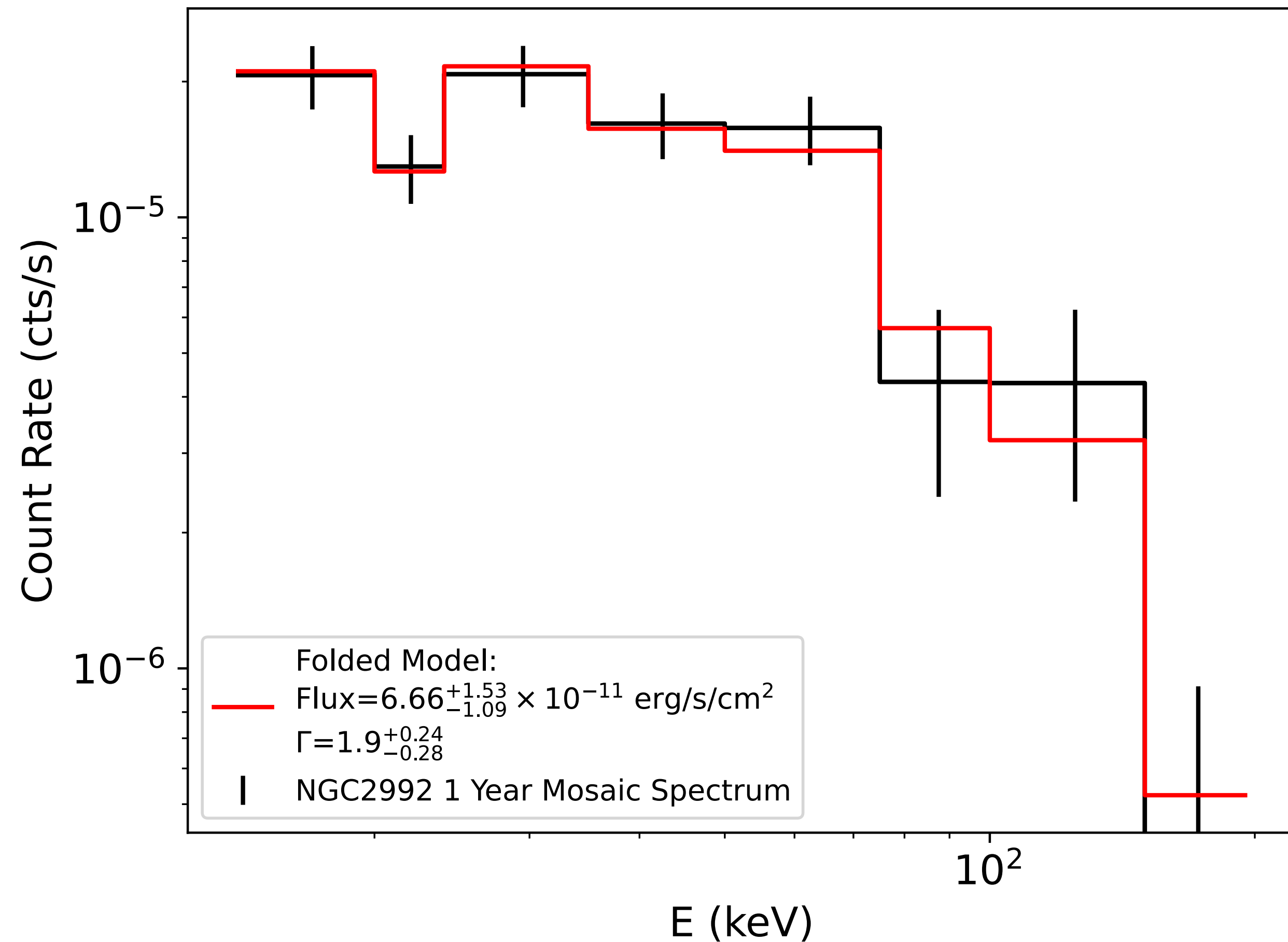
- Analyzed data from Dec 2004 - Dec 2005
- Obtained upper limits for each individual survey dataset and the monthly mosaics
- The year long mosaic image has a detection of SNR~12 in 14-195 keV energy band



BatAnalysis Allows for Custom Mosaicing

AGN NGC 2992

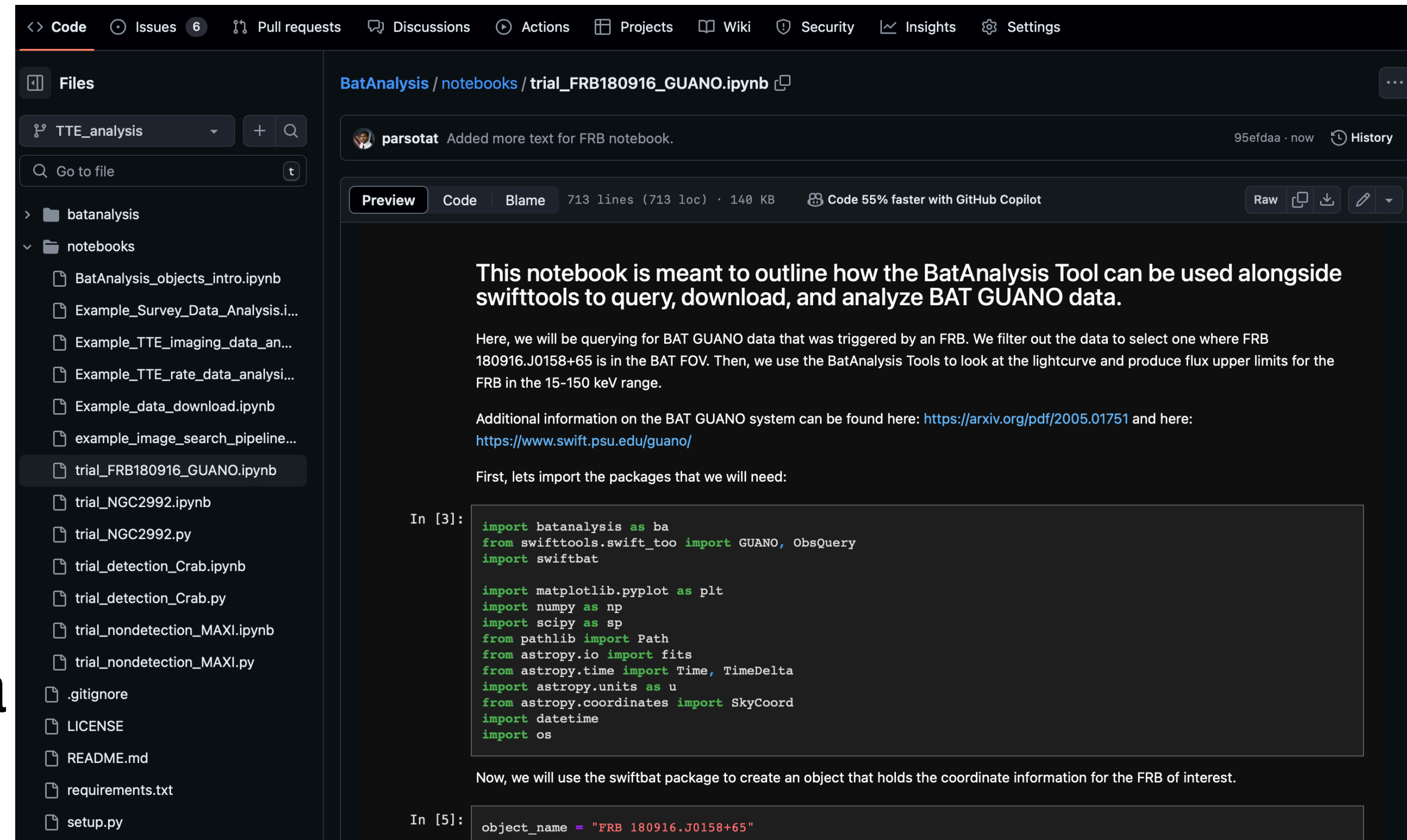
- Analyzed data from Dec 2004 - Dec 2005
- Obtained upper limits for each individual survey dataset and the monthly mosaics
- The year long mosaic image has a detection of SNR~12 in 14-195 keV energy band
- The spectral fit show agreement with prior analyses



BETA: BatAnalysis Allows for Custom Event Data Analysis

BAT GUANO Data

- Currently working on this portion of the package
- The beta version is available for testing with only one example analysis.
- There are notebooks that give a general overview of BAT event data files and ways to manipulate the data



The screenshot shows a GitHub repository for 'BatAnalysis' with a notebook titled 'trial_FRB180916_GUANO.ipynb'. The notebook content is as follows:

This notebook is meant to outline how the BatAnalysis Tool can be used alongside swifttools to query, download, and analyze BAT GUANO data.

Here, we will be querying for BAT GUANO data that was triggered by an FRB. We filter out the data to select one where FRB 180916.J0158+65 is in the BAT FOV. Then, we use the BatAnalysis Tools to look at the lightcurve and produce flux upper limits for the FRB in the 15-150 keV range.

Additional information on the BAT GUANO system can be found here: <https://arxiv.org/pdf/2005.01751> and here: <https://www.swift.psu.edu/guano/>

First, lets import the packages that we will need:

```
In [3]: import batanalysis as ba
from swifttools.swift_tool import GUANO, ObsQuery
import swiftbat

import matplotlib.pyplot as plt
import numpy as np
import scipy as sp
from pathlib import Path
from astropy.io import fits
from astropy.time import Time, TimeDelta
import astropy.units as u
from astropy.coordinates import SkyCoord
import datetime
import os
```

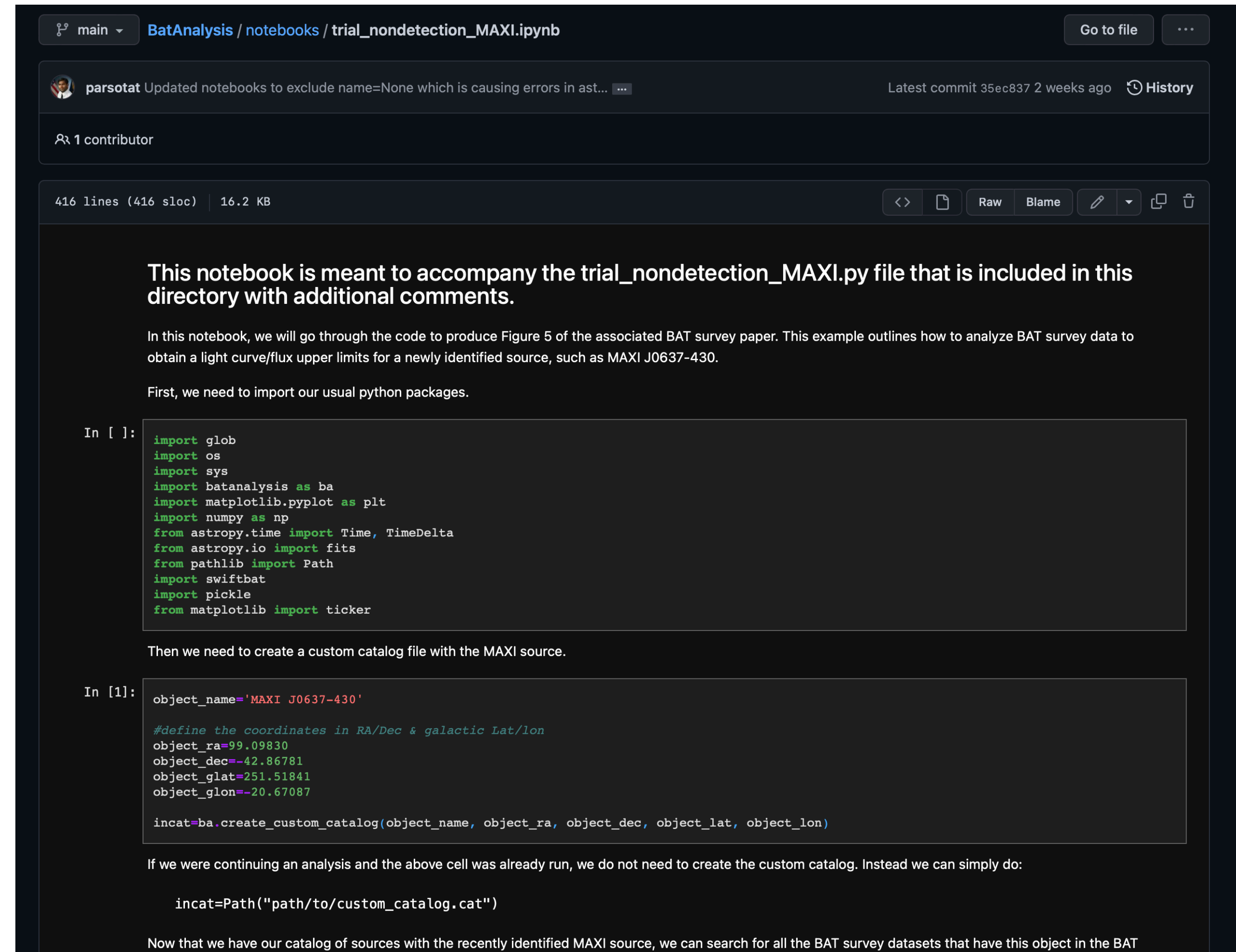
Now, we will use the swiftbat package to create an object that holds the coordinate information for the FRB of interest.

```
In [5]: object_name = "FRB 180916.J0158+65"
```

BatAnalysis Survey Documentation

<https://github.com/parsotat/BatAnalysis/tree/main/notebooks>

- Documentation is currently provided as Jupyter notebooks
 - Allow the results to be reproduced
 - Broadly explains what each step of an analysis is doing
 - A word of caution:
 - Some of these analyses can take a VERY large amount of space (eg the full 22 month Crab data & BatAnalyses produced files are ~2.5 TB in total)
 - The NGC analysis takes up ~815 GB
 - The MAXI analysis takes up ~330 GB
 - The BOAT analysis takes up ~60 GB of storage
 - The parallelized mosaic analyses capability can use a lot of memory at ~10 GB per process



main BatAnalysis / notebooks / trial_nondetection_MAXI.ipynb

parsotat Updated notebooks to exclude name=None which is causing errors in ast... Latest commit 35ec837 2 weeks ago History

1 contributor

416 lines (416 sloc) 16.2 KB

This notebook is meant to accompany the trial_nondetection_MAXI.py file that is included in this directory with additional comments.

In this notebook, we will go through the code to produce Figure 5 of the associated BAT survey paper. This example outlines how to analyze BAT survey data to obtain a light curve/flux upper limits for a newly identified source, such as MAXI J0637-430.

First, we need to import our usual python packages.

```
In [ ]: import glob
import os
import sys
import batanalysis as ba
import matplotlib.pyplot as plt
import numpy as np
from astropy.time import Time, TimeDelta
from astropy.io import fits
from pathlib import Path
import swiftbat
import pickle
from matplotlib import ticker
```

Then we need to create a custom catalog file with the MAXI source.

```
In [1]: object_name='MAXI J0637-430'

#define the coordinates in RA/Dec & galactic Lat/lon
object_ra=99.09830
object_dec=-42.86781
object_glat=251.51841
object_glon=-20.67087

incat=ba.create_custom_catalog(object_name, object_ra, object_dec, object_lat, object_lon)
```

If we were continuing an analysis and the above cell was already run, we do not need to create the custom catalog. Instead we can simply do:

```
incat=Path("path/to/custom_catalog.cat")
```

Now that we have our catalog of sources with the recently identified MAXI source, we can search for all the BAT survey datasets that have this object in the BAT

BatAnalysis TTE Documentation

https://github.com/parsotat/BatAnalysis/tree/TTE_analysis/notebooks

- Documentation is currently provided as Jupyter notebooks
 - 2 Primary Notebooks outlining the imaging capabilities and the traditional rate capabilities
- This is still the beta version and development is ongoing so be wary of any bugs.

The screenshot shows a GitHub repository for 'BatAnalysis' with a file browser on the left and a Jupyter notebook viewer on the right. The notebook is titled 'Example_TTE_rate_data_analysis.ipynb' and contains the following text:

BAT TTE Rate Analysis Overview

This notebook gives a brief overview of the capabilities of the BatAnalysis package to dynamically analyze BAT Time-tagged Event (TTE) data.

Installation instructions for Beta testers:

Thanks to all who are willing to test the TTE portion of BatAnalysis. Here are some quick instructions for getting the code and installing it for access in python:

- `git clone -b TTE_analysis https://github.com/parsotat/BatAnalysis.git`
- `cd BatAnalysis`
- if BatAnalysis is already installed: `pip uninstall BatAnalysis`
- `pip install -e .`

Then in a jupyter notebook or an ipython session `import batanalysis as ba` should work.

Any issues that get brought up will be pushed to the github branch. To get these changes simply do `git pull` in the BatAnalysis directory and the next time you do `import batanalysis as ba` the changes will be implemented.

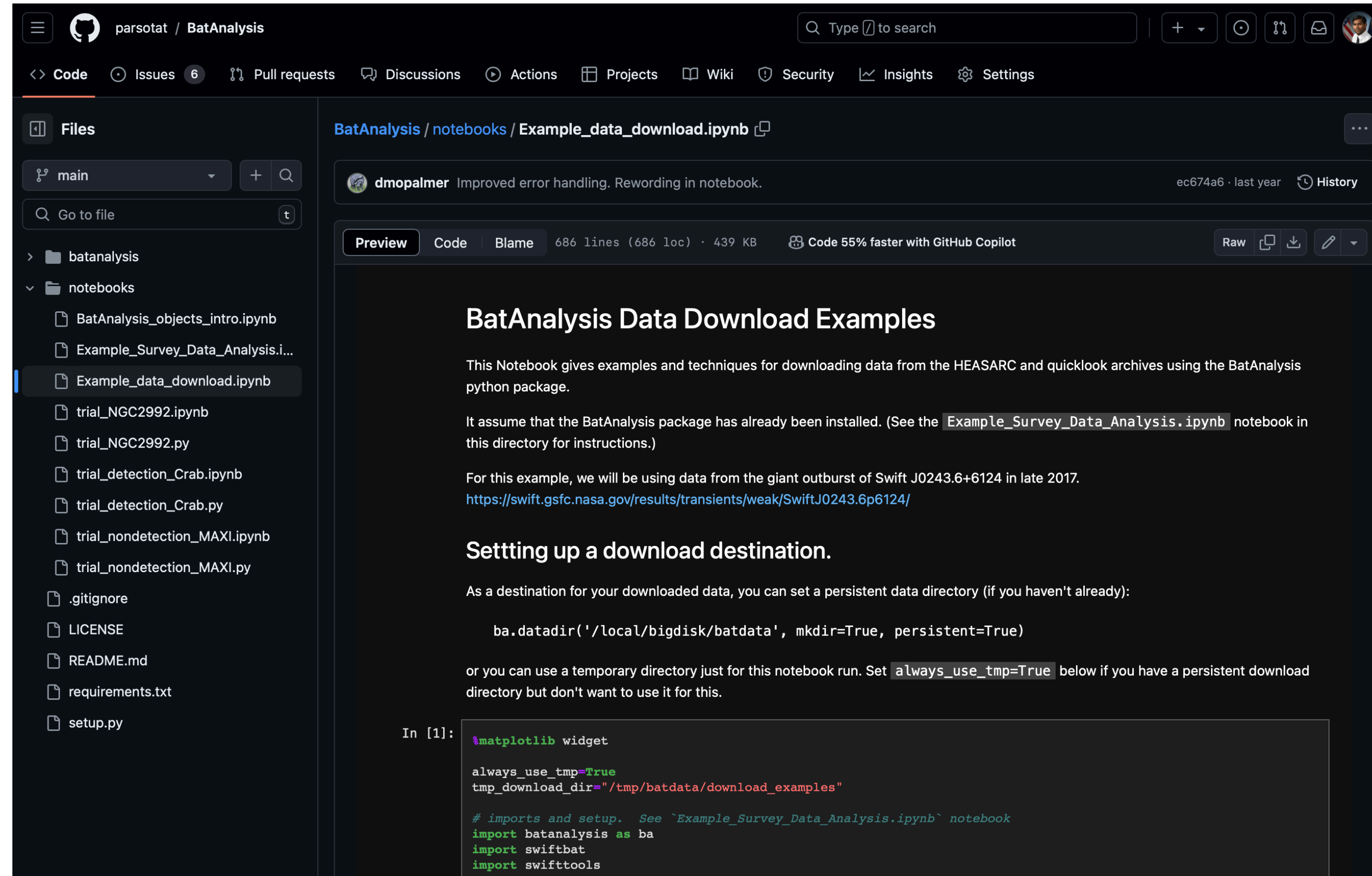
Here, we will analyze a BAT triggered GRB to exhibit the various capabilities that the BatAnalysis tool offers to interact with the TTE data, allowing users to download data, create lightcurves which can be dynamically rebinned in time and energy, and also create Pulse Height Amplitude (PHA) files for any arbitrary time bin at arbitrary energy binning.

First, we will look at GRB 211211A, a bright long GRB that was determined to be from a merger of compact objects. This GRB was detected by BAT and the refined analyses can be found here:
<https://swift.gsfc.nasa.gov/results/batgrbcats/GRB211211A/web/GRB211211A.html>

Data Querying and Download Documentation

https://github.com/parsotat/BatAnalysis/blob/main/notebooks/Example_data_download.ipynb

- Various Examples show different ways to query data programmatically
- This notebook walks through the use of swifttools and various files that can be downloaded
- Additional documentation can be found on the swifttools website: https://www.swift.psu.edu/too_api/



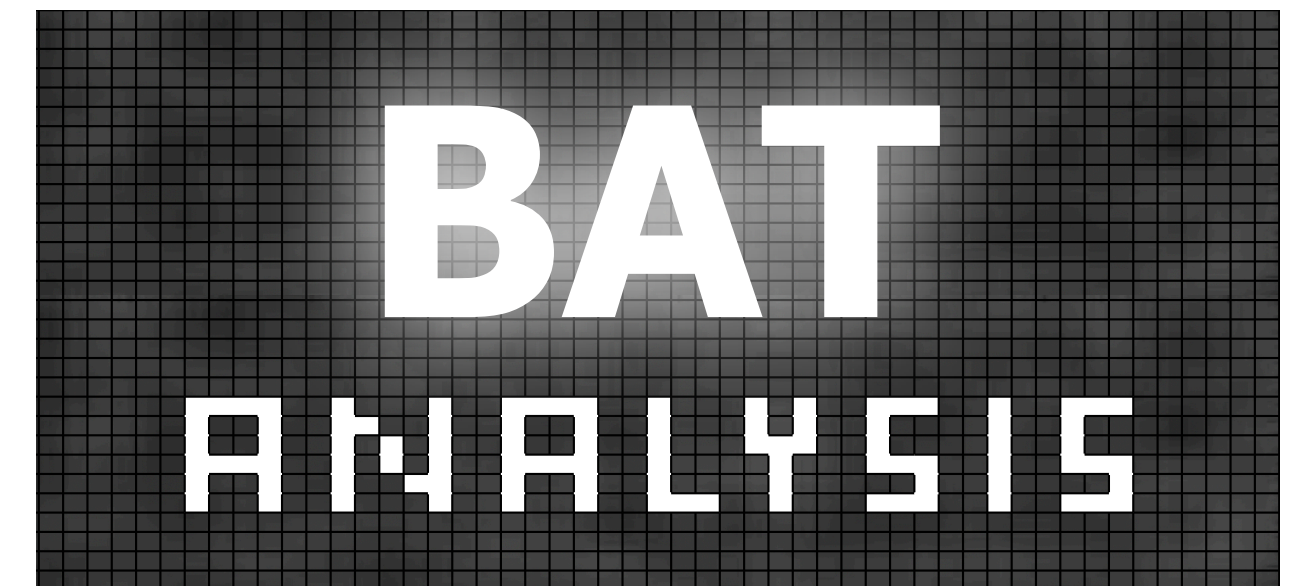
The screenshot shows the GitHub interface for the repository 'parsotat / BatAnalysis'. The file browser on the left lists the directory structure, including 'batanalysis' and 'notebooks'. The 'notebooks' directory contains several files, with 'Example_data_download.ipynb' selected. The main content area displays the notebook's preview, which includes a title 'BatAnalysis Data Download Examples', an introduction to the notebook's purpose, and instructions for setting up a download destination. The code section shows the following Python code:

```
In [1]: %matplotlib widget
always_use_tmp=True
tmp_download_dir="/tmp/batdata/download_examples"

# imports and setup. See `Example_Survey_Data_Analysis.ipynb` notebook
import batanalysis as ba
import swiftbat
import swifttools
```

Summary

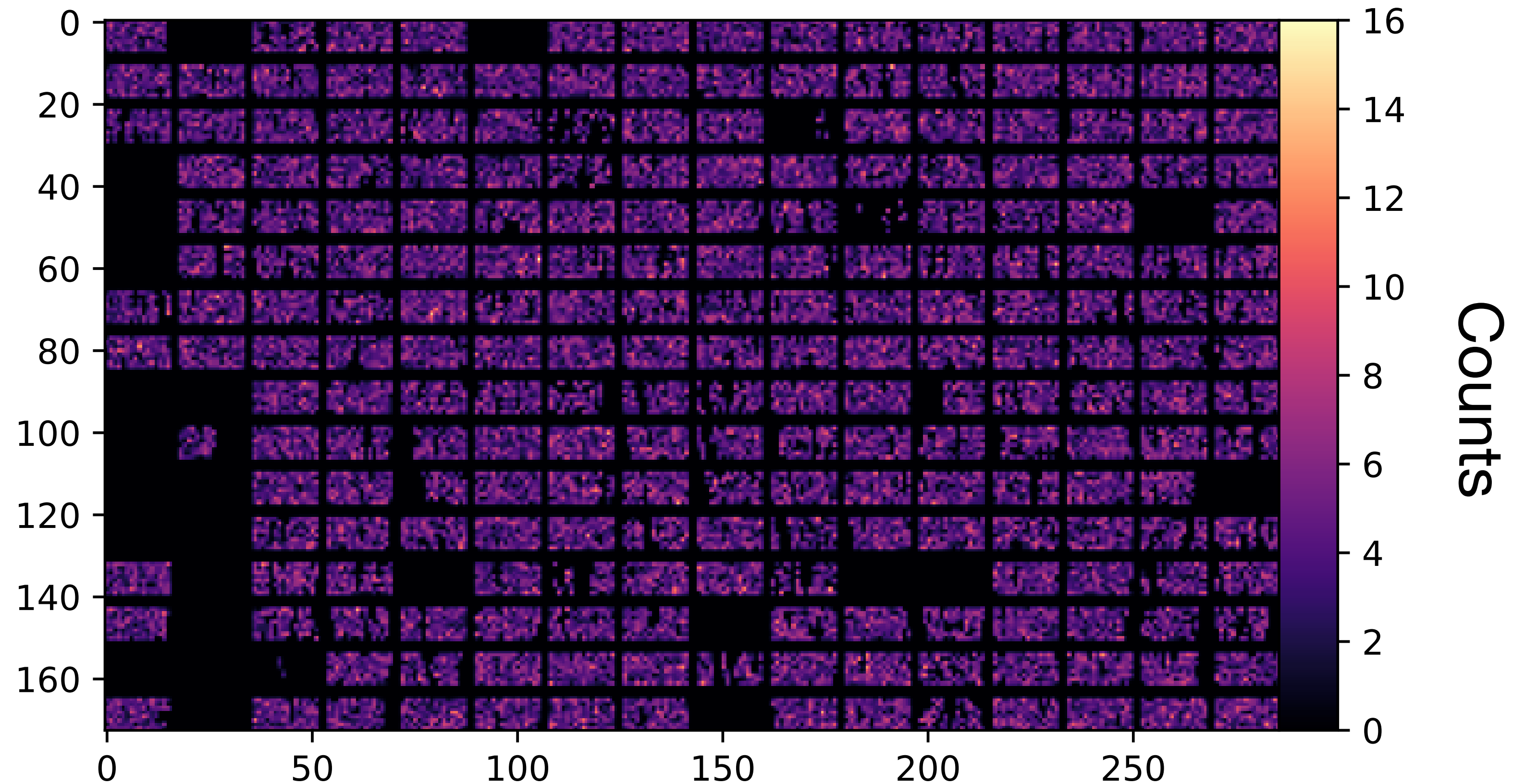
- The BatAnalysis package unlocks the potential of BAT data for a number of different analyses
- The package is open source and can be used by anyone
- Software such as this is dependent on other open source efforts including:
 - Swifttools (https://www.swift.psu.edu/too_api/)
 - The swift_bat package (https://github.com/lanl/swiftbat_python)
 - HEASoftpy
 - Astropy & Astroquery
- Feedback from users such as yourselves
 - If issues are encountered users should open a GitHub issue outlining the problem and any relevant code
 - If the documentation is not clear please let me know by opening an GitHub issue as well so we can clarify things



Survey Observation Processing Steps

1. Raw Data Processing

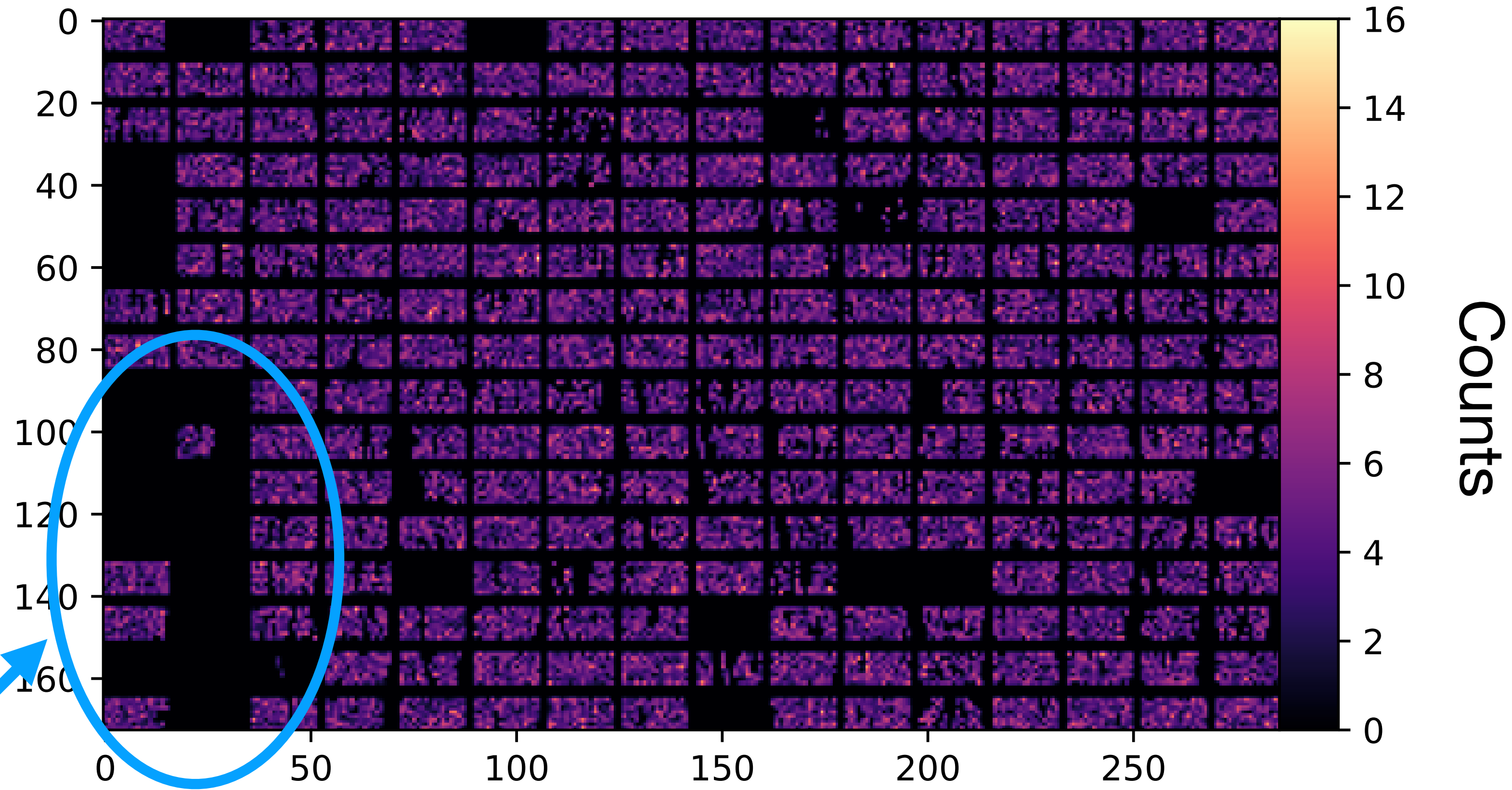
DPH (Channel i of 80)



Survey Observation Processing Steps

1. Raw Data Processing

DPH (Channel i of 80)

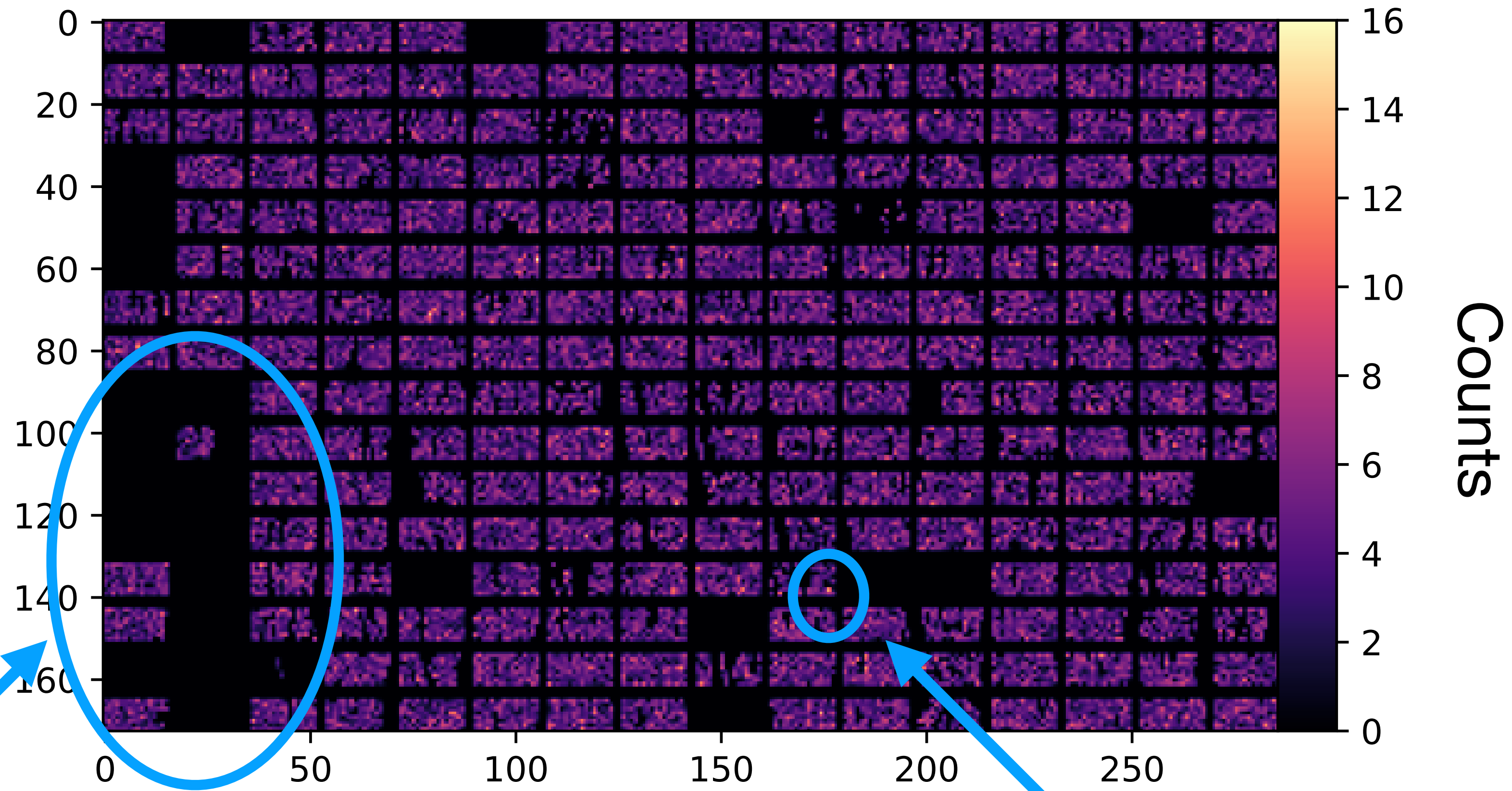


Disabled Detectors

Survey Observation Processing Steps

1. Raw Data Processing

DPH (Channel i of 80)



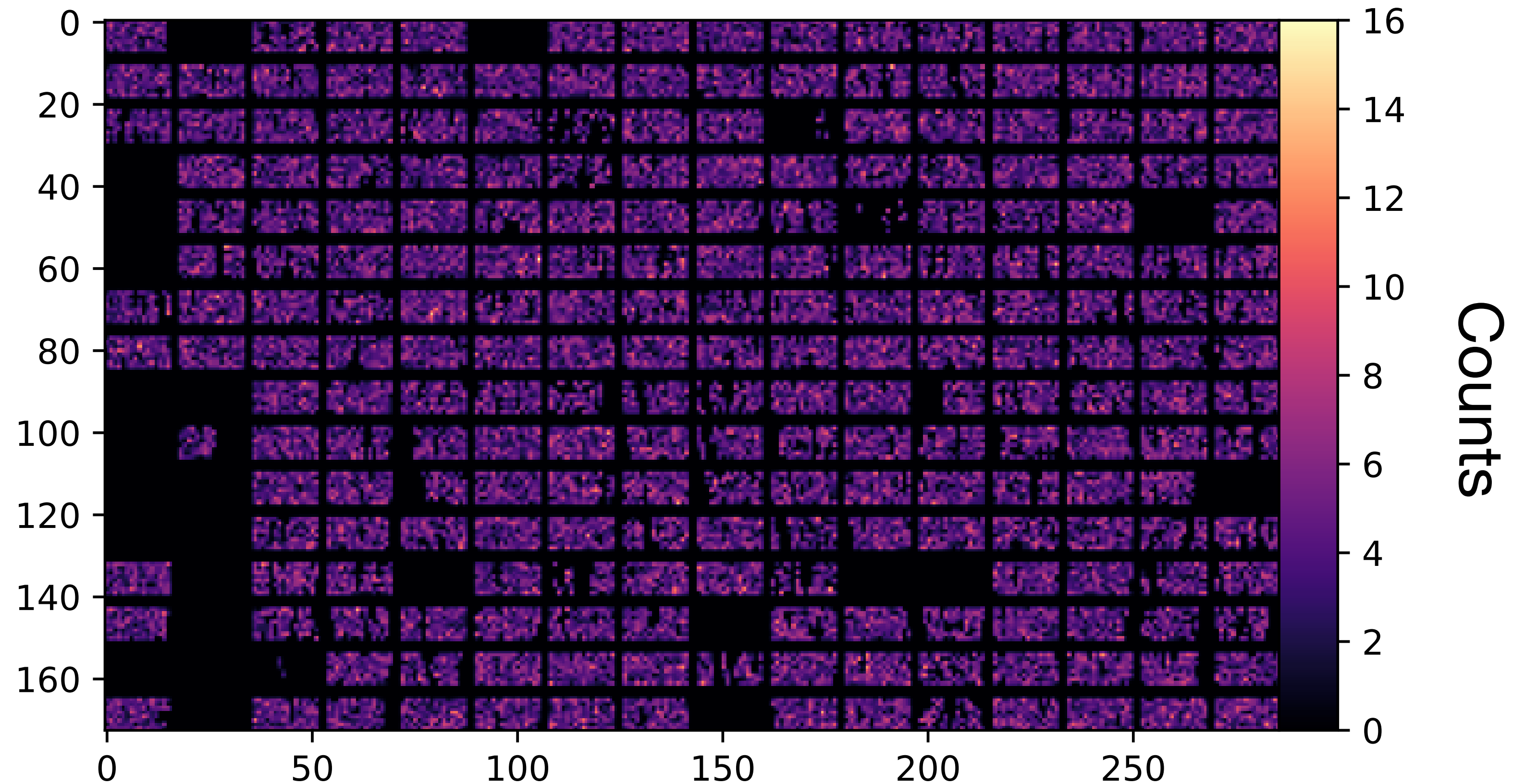
Disabled Detectors

Hot Pixel?

Survey Observation Processing Steps

1. Raw Data Processing

DPH (Channel i of 80)

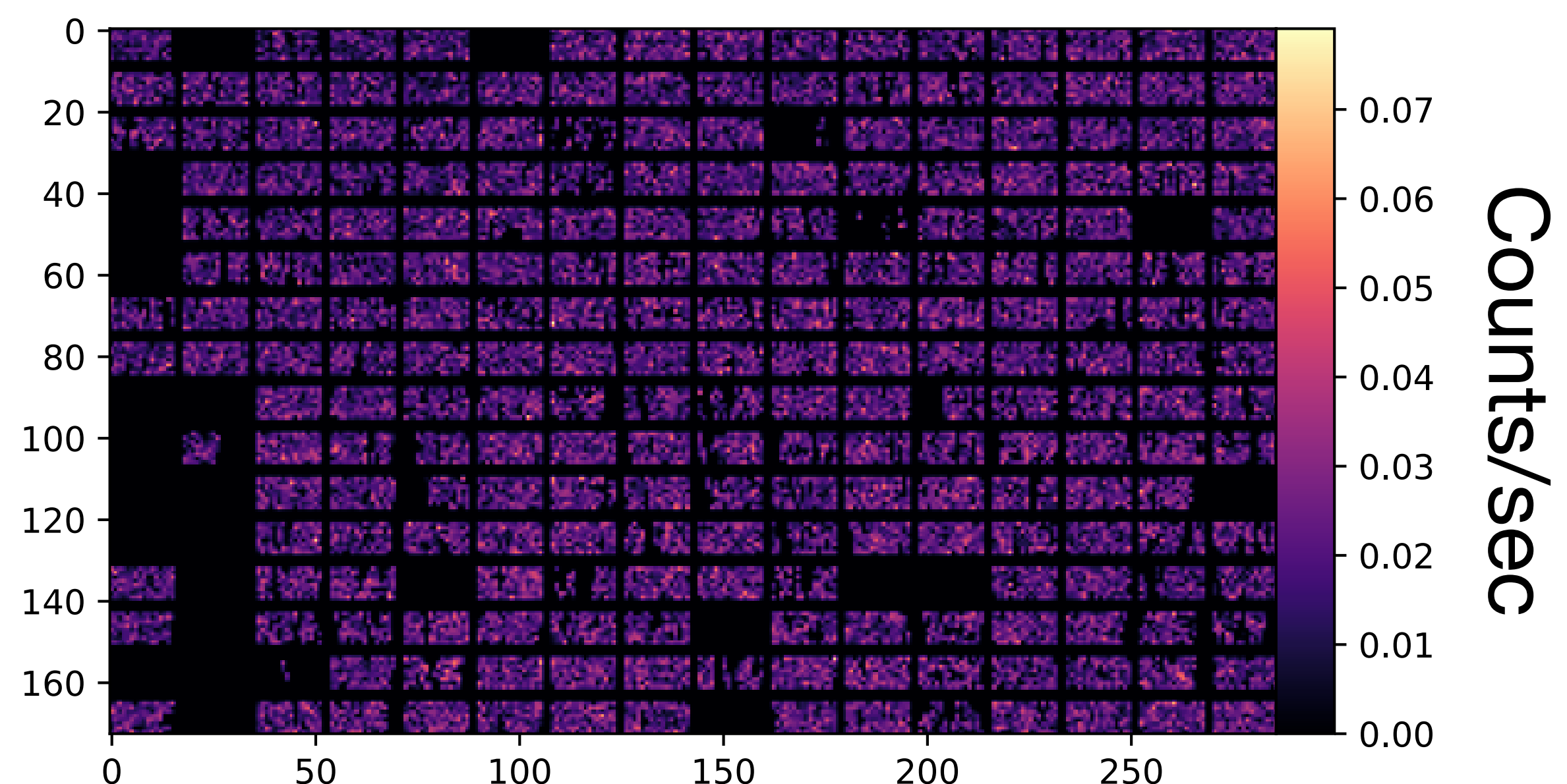
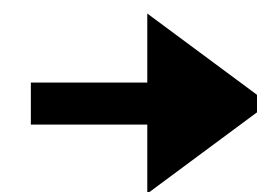
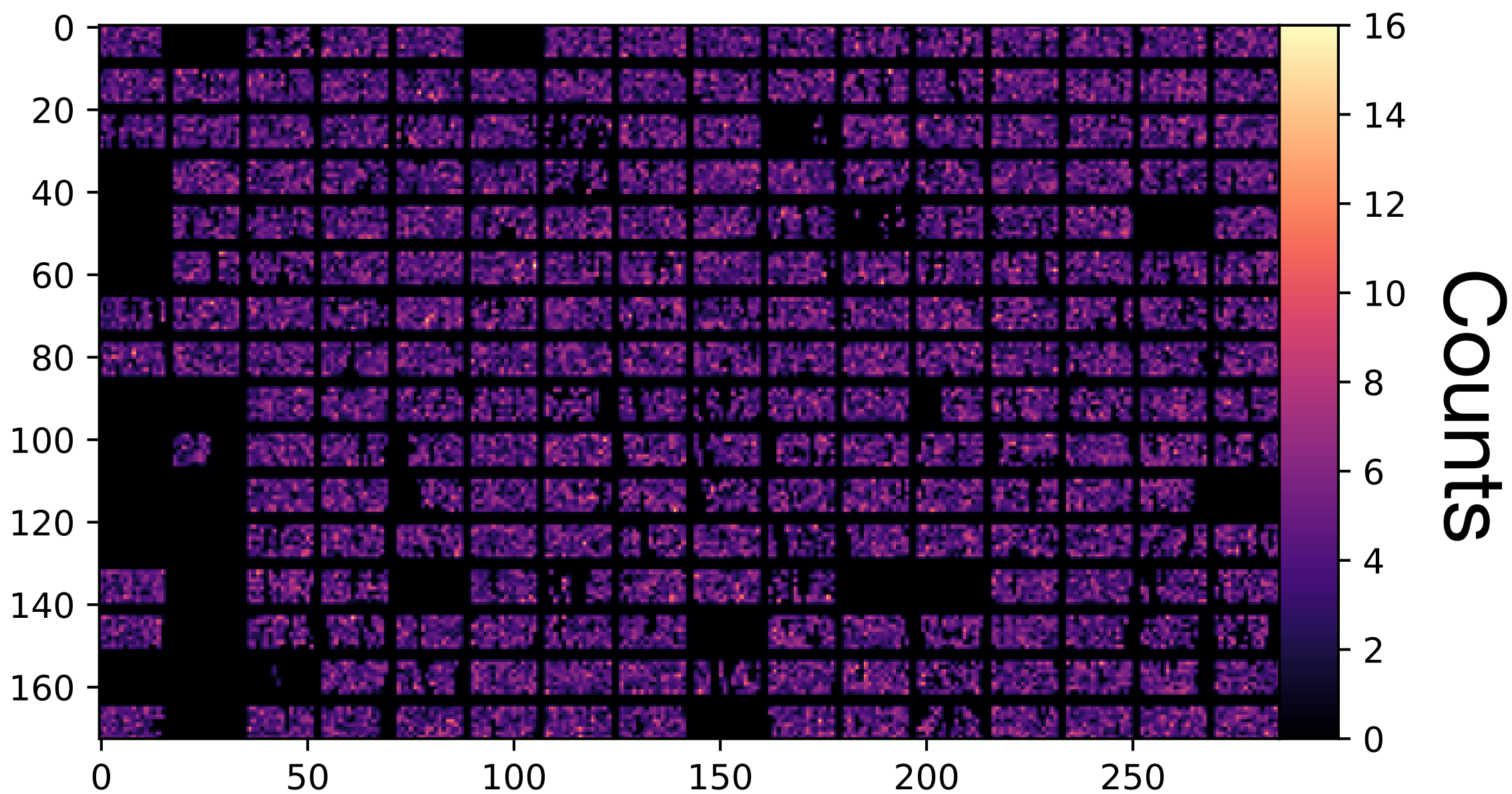


Survey Observation Processing Steps

1. Raw Data Processing

DPH (Channel i of 80)

DPI (Energy Bin i of 8)

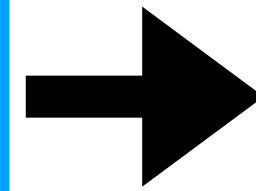
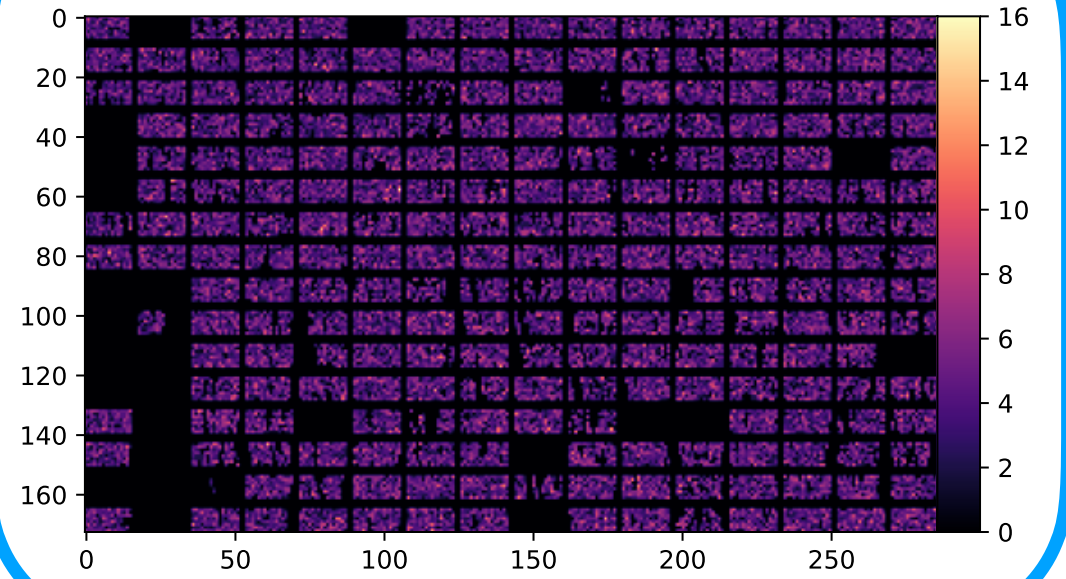


- Rebin data into 8 energy bins: 20-24, 24-35, 35-50, 50-75, 75-100, 100-150, & 150-195 keV

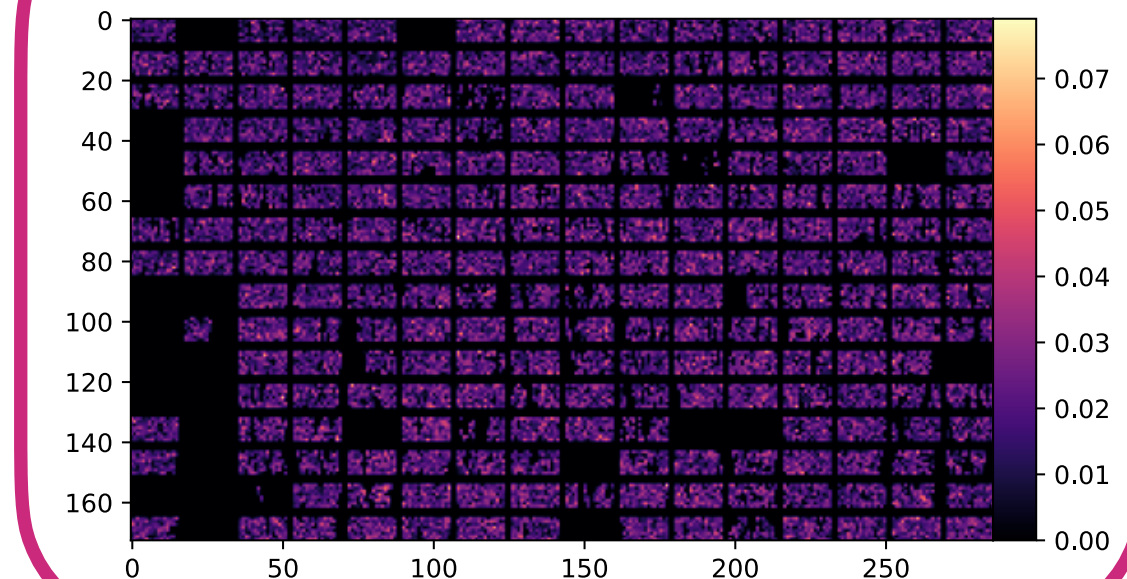
Survey Observation Processing Steps

1. Raw Data Processing

DPH

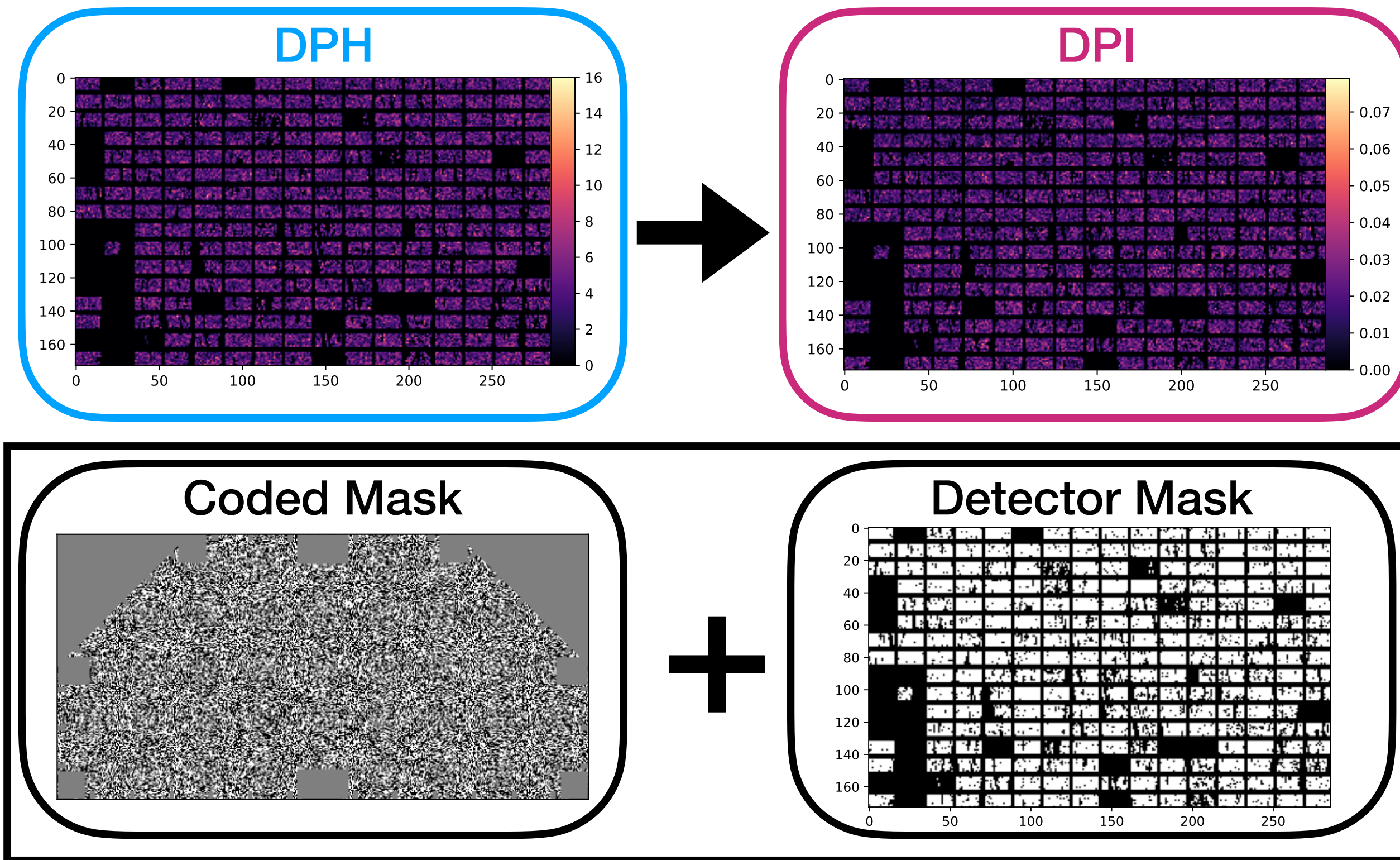


DPI



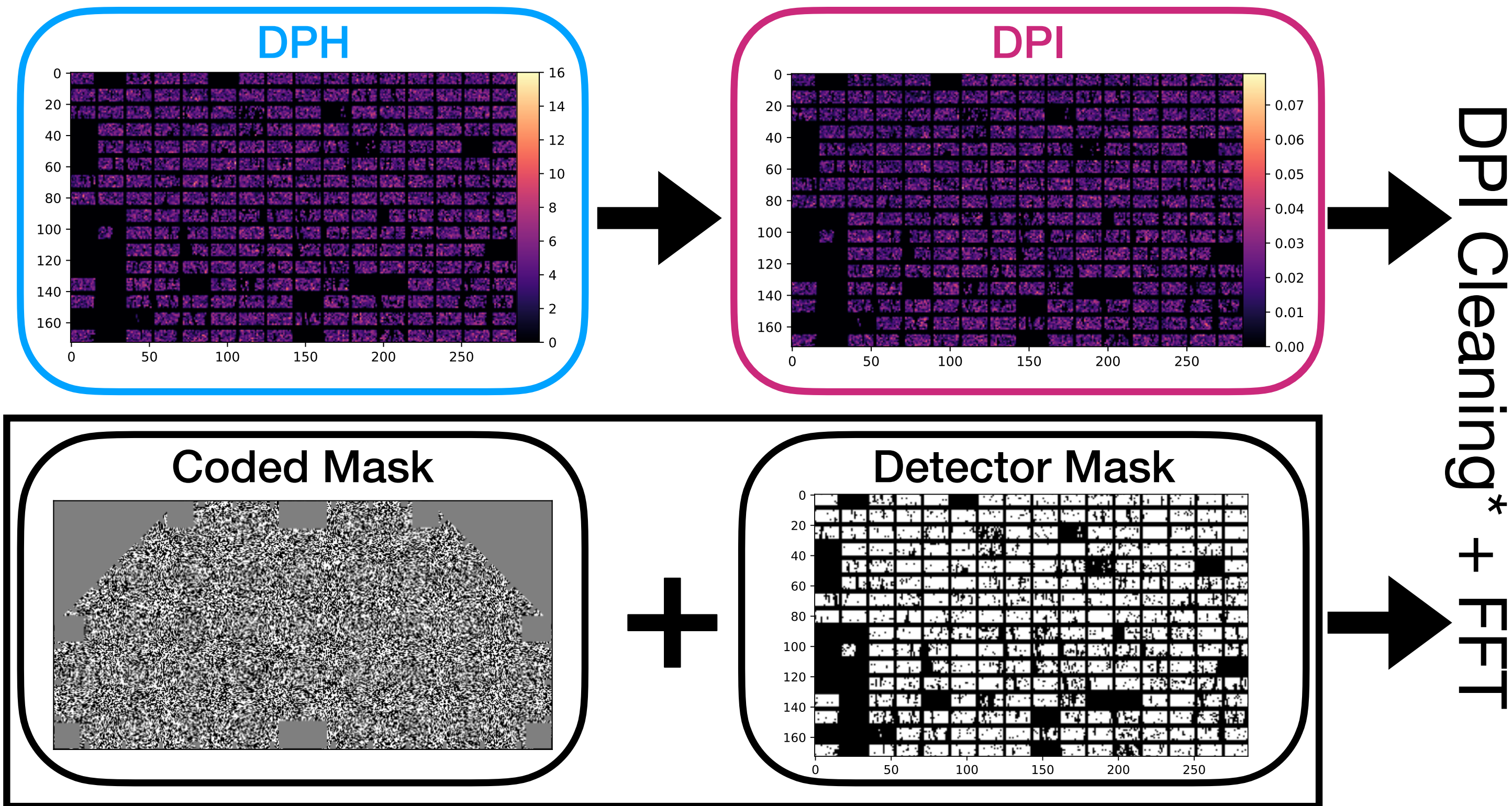
Survey Observation Processing Steps

1. Raw Data Processing



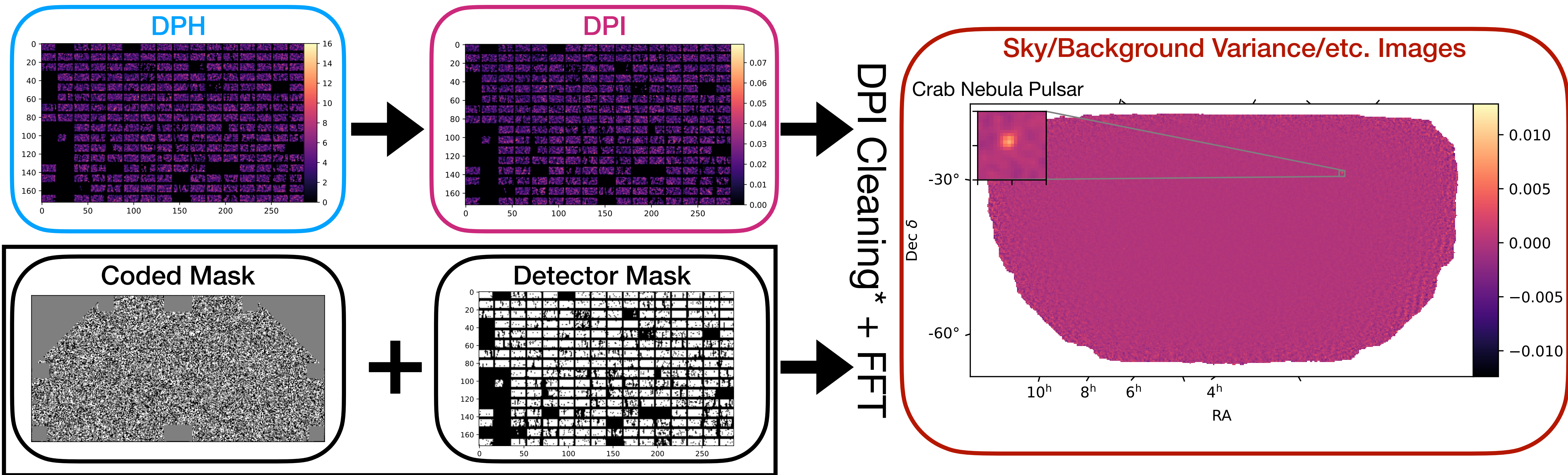
Survey Observation Processing Steps

1. Raw Data Processing



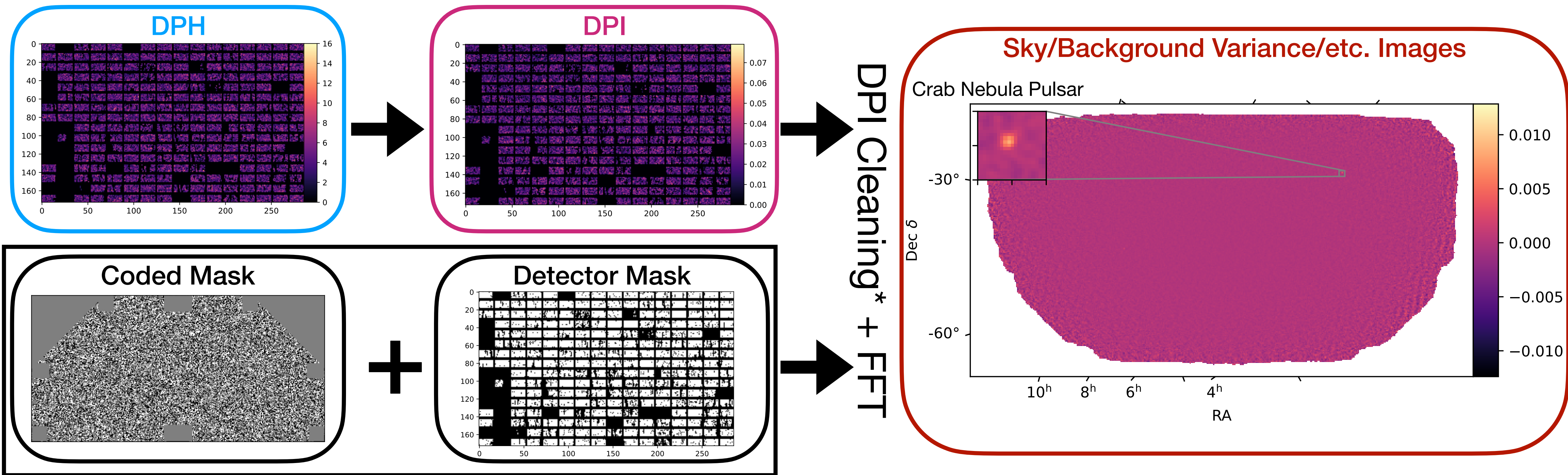
Survey Observation Processing Steps

1. Raw Data Processing



Survey Observation Processing Steps

1. Raw Data Processing

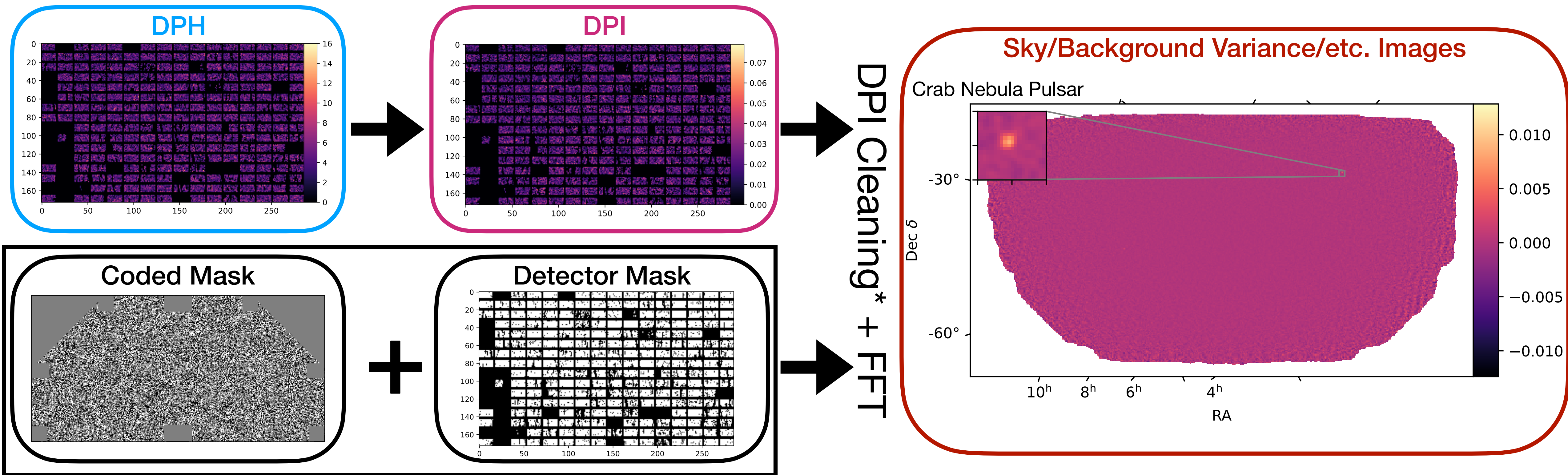


DPI Cleaning:

- Cleans counts from bright sources with ray tracing
- Cleans noisy pixels from DPIs

Survey Observation Processing Steps

1. Raw Data Processing



DPI Cleaning:

- Cleans counts from bright sources with ray tracing
- Cleans noisy pixels from DPIs

Pattern Noise Map Subtraction:

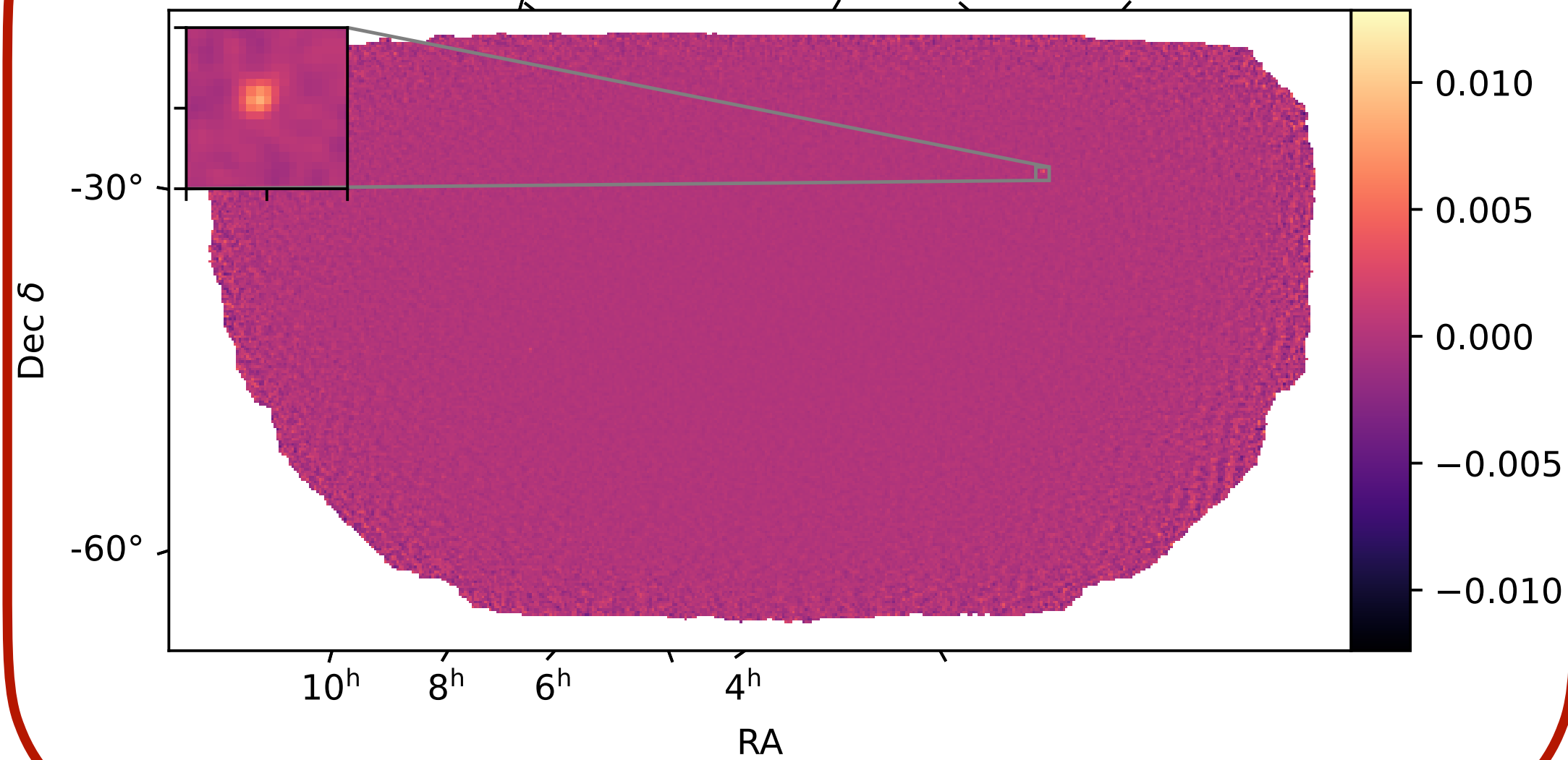
- Subtracts off persistent detector noise that builds up over time

Survey Observation Processing Steps

2. Image Processing - Individual Survey Datasets

Sky/Background Variance/etc. Images

Crab Nebula Pulsar

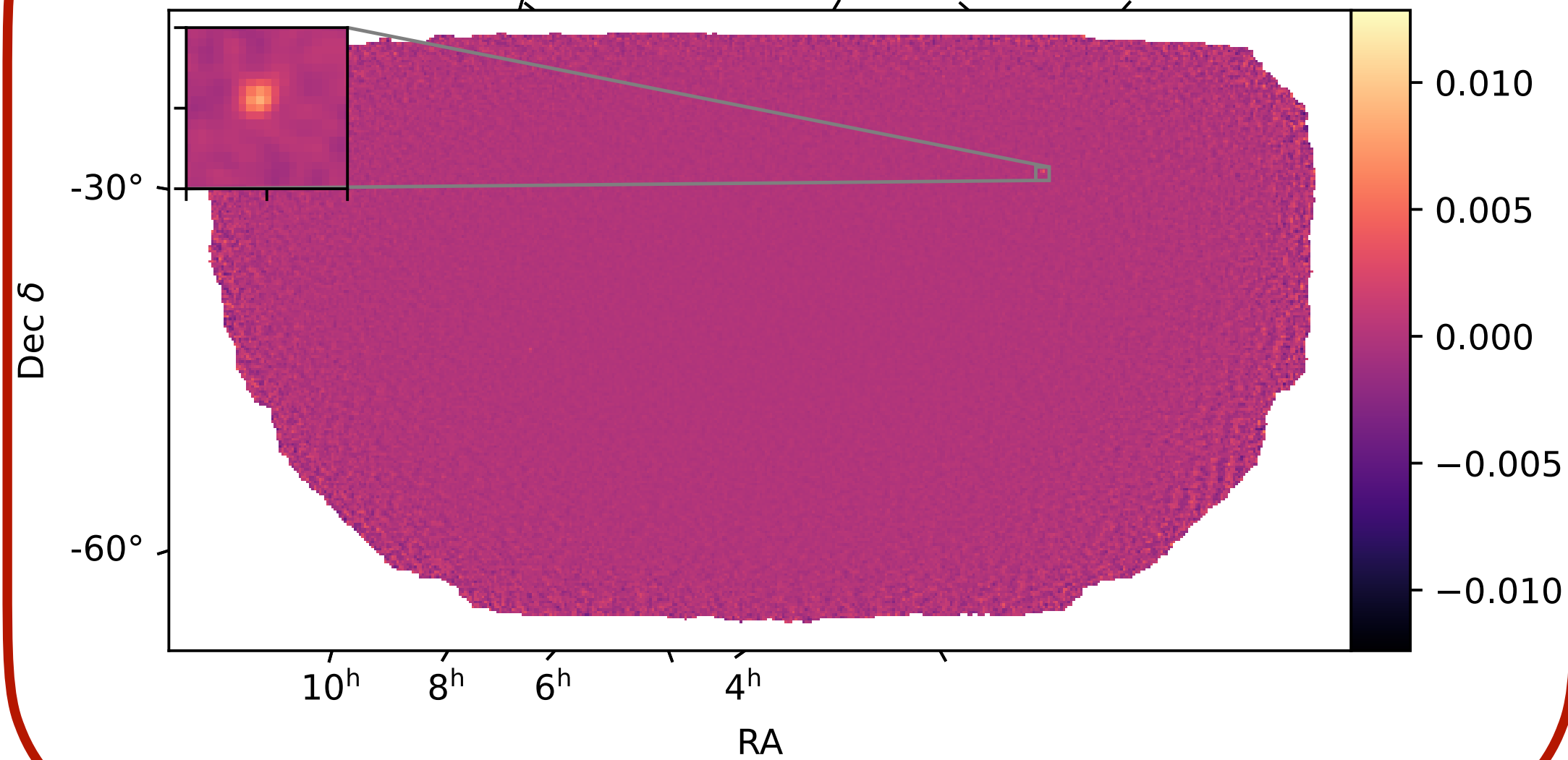


Survey Observation Processing Steps

2. Image Processing - Individual Survey Datasets

Sky/Background Variance/etc. Images

Crab Nebula Pulsar

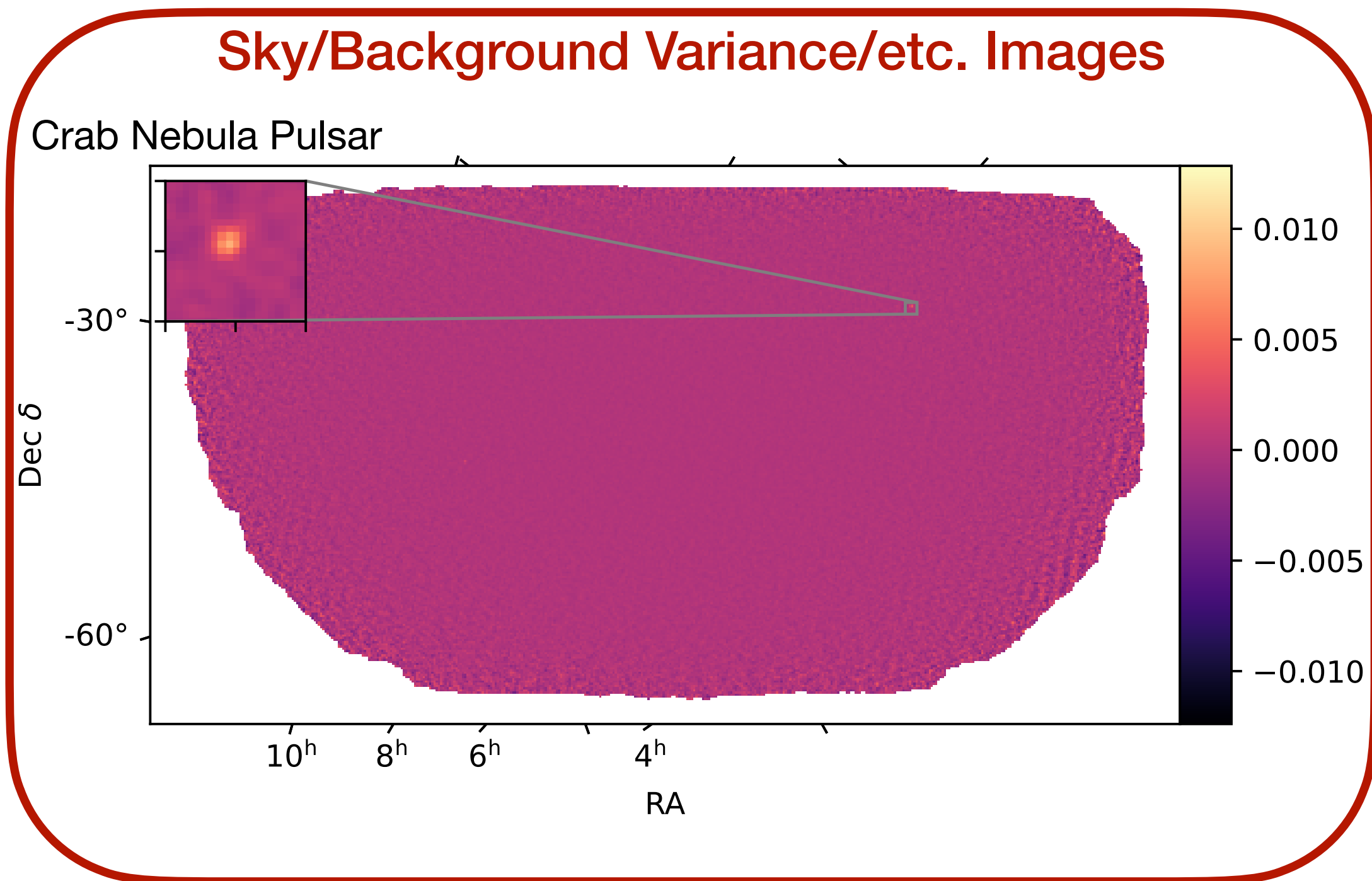


batcelldetect



Survey Observation Processing Steps

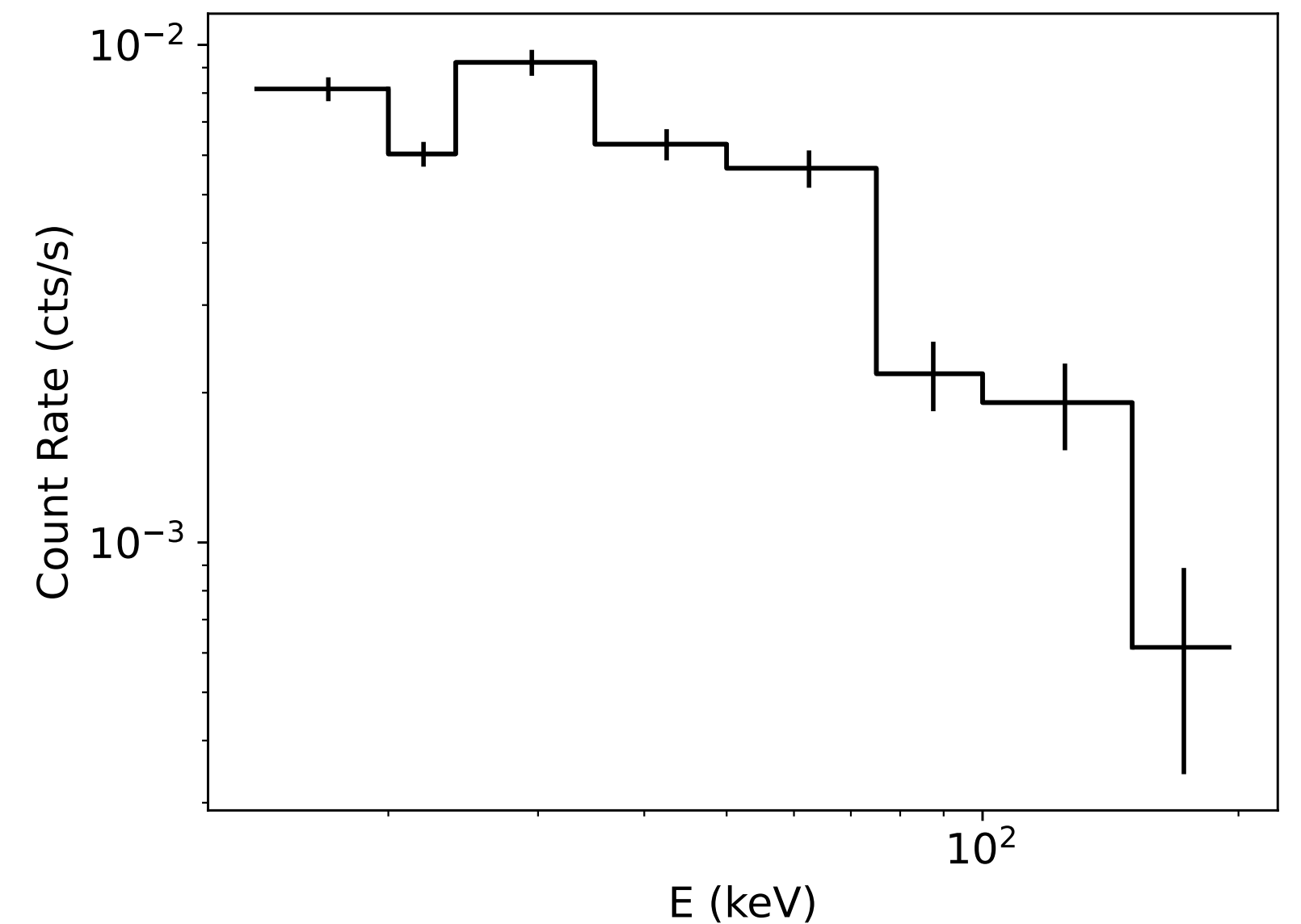
2. Image Processing - Individual Survey Datasets



batcelldetect

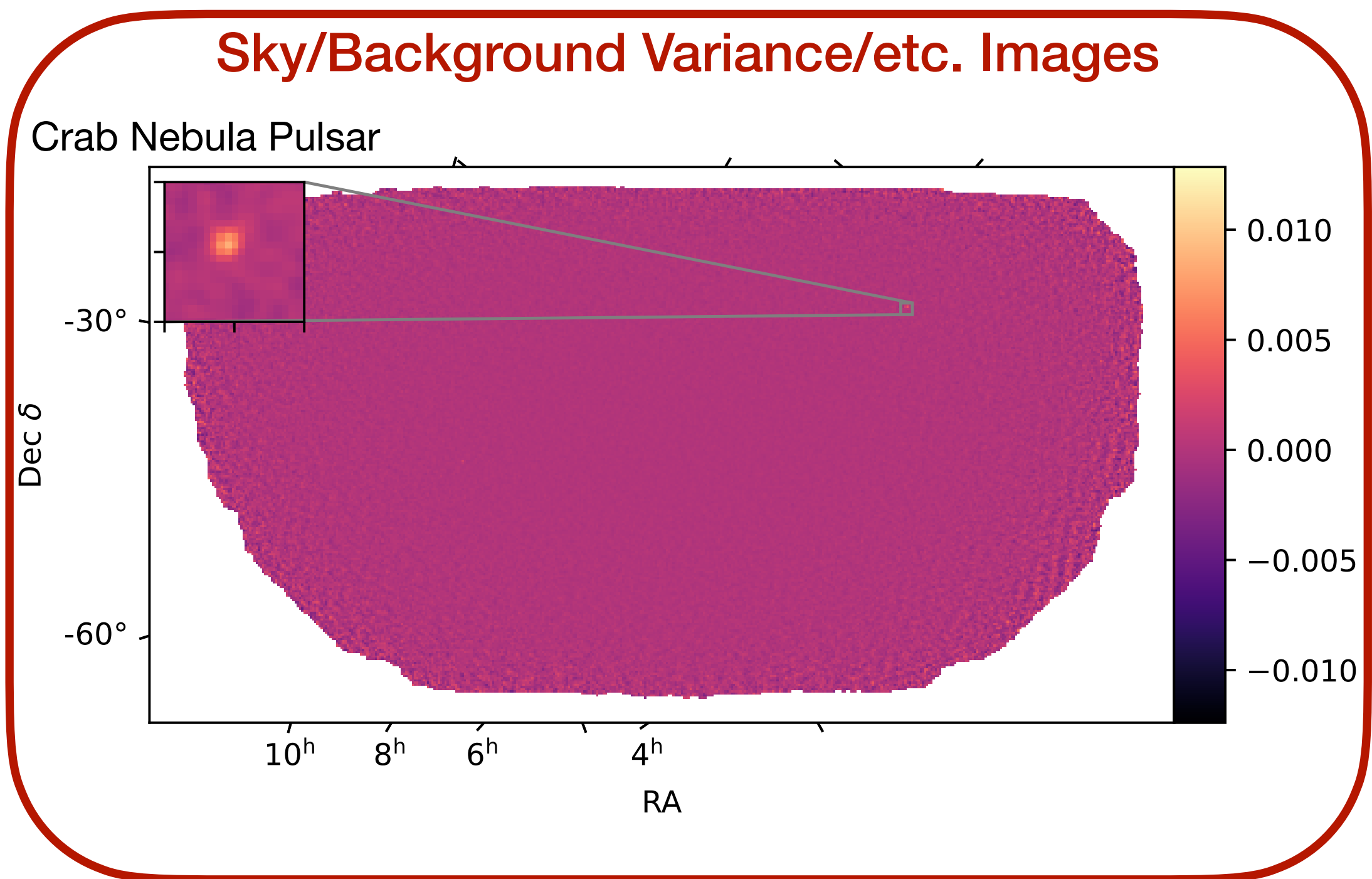


- For Each Source that may be found:
 - Count rate in each energy bin
 - SNR of the detection
 - Local background variation



Survey Observation Processing Steps

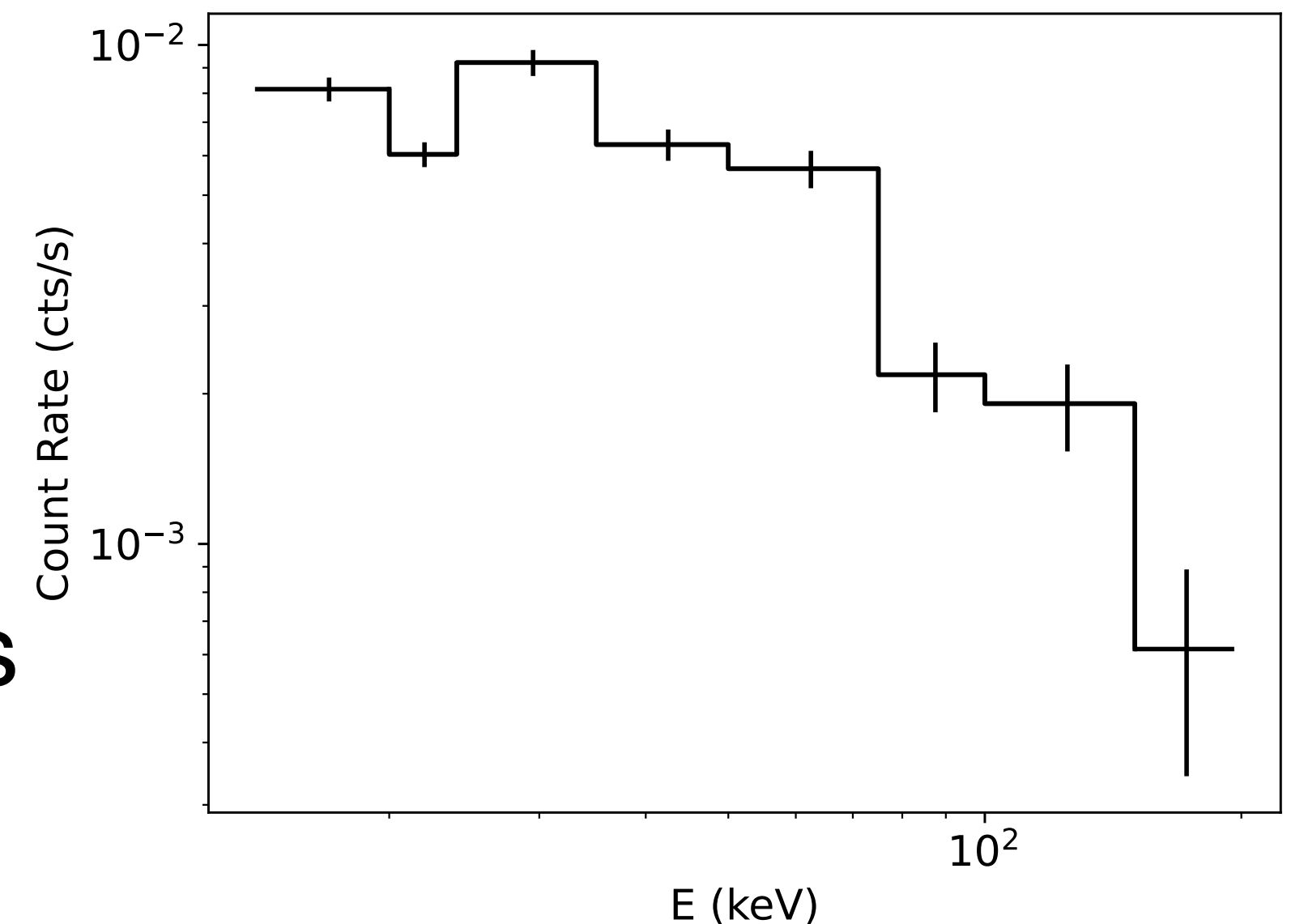
2. Image Processing - Individual Survey Datasets



batcelldetect



- For Each Source that may be found:
 - Count rate in each energy bin
 - SNR of the detection
 - Local background variation

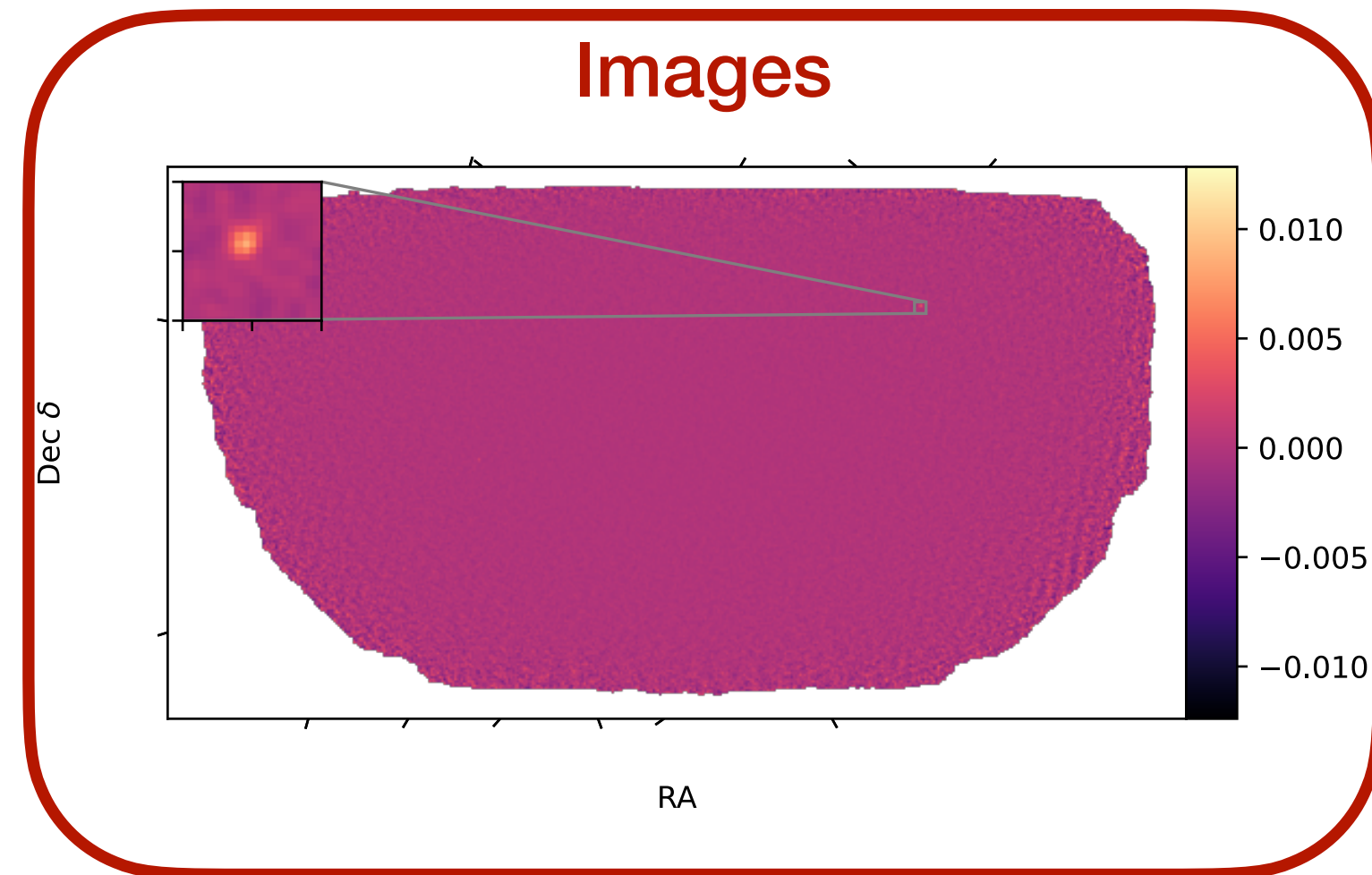


Batcelldetect: Uses a sliding annulus to detect sources and measure the local background counts

Survey Observation Processing Steps

2. Image Processing - Mosaic Survey Datasets

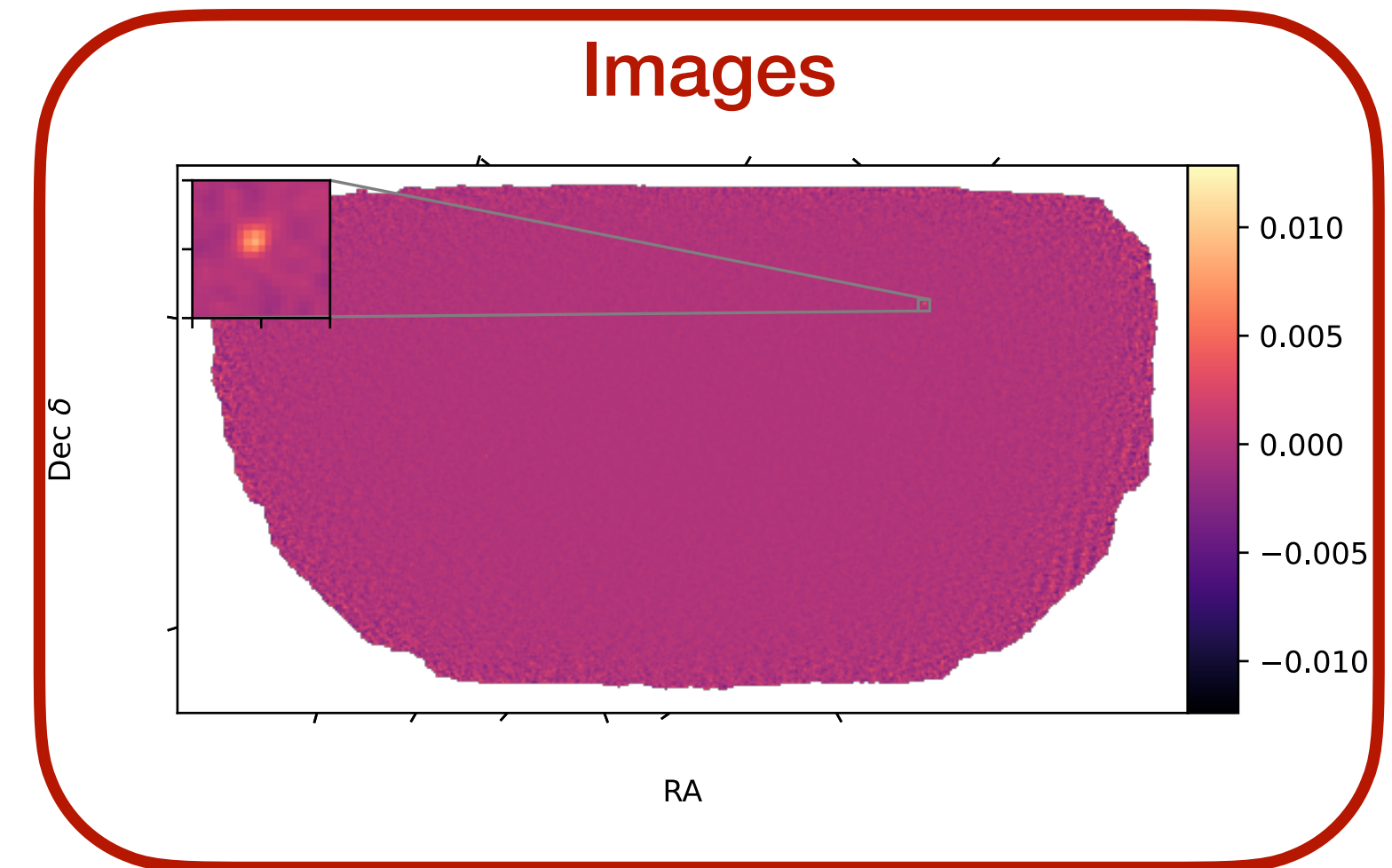
Observation 1



?

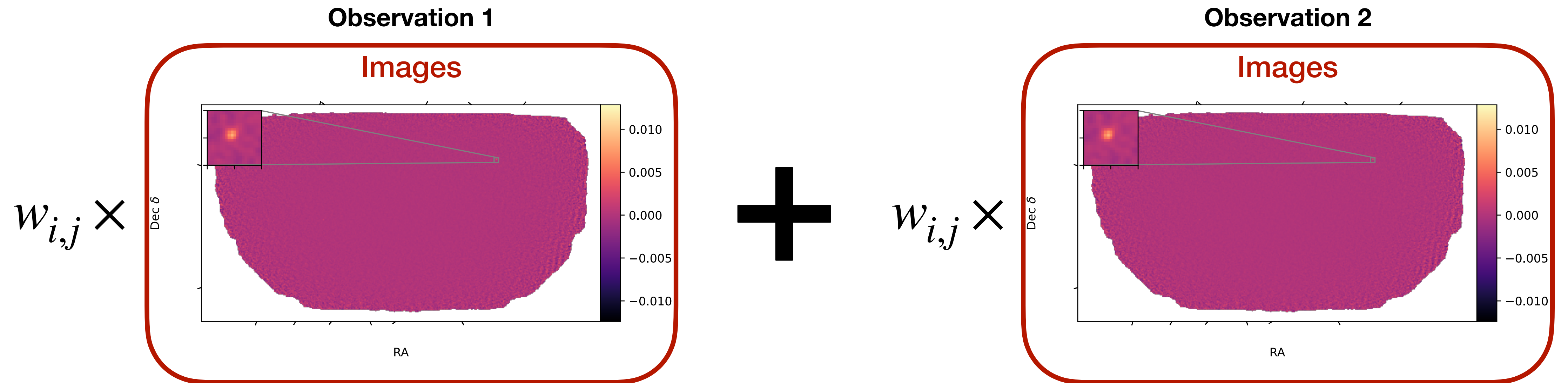
+

Observation 2



Survey Observation Processing Steps

2. Image Processing - Mosaic Survey Datasets



$$w_{i,j} \propto \sigma_{i,j}^{-2}$$

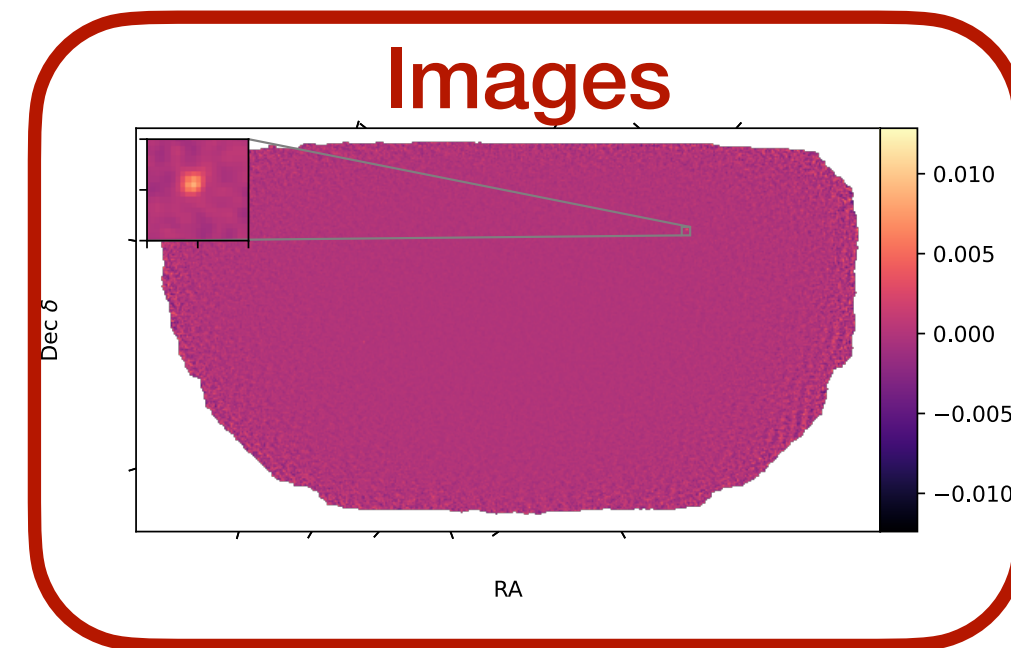
This suppresses counts in noisy images

Survey Observation Processing Steps

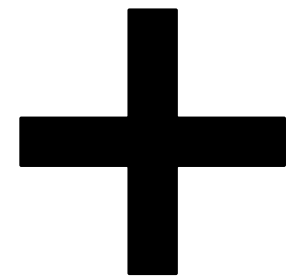
2. Image Processing - Mosaic Survey Datasets

Observations within some time bin

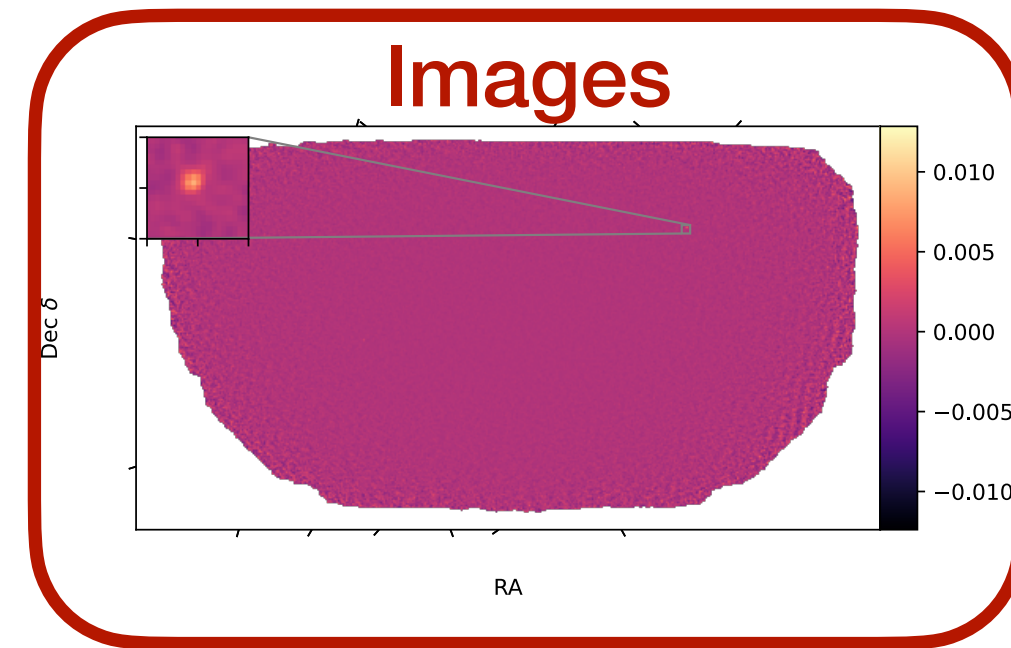
Observation 1



$w_{i,j} \times$



Observation N

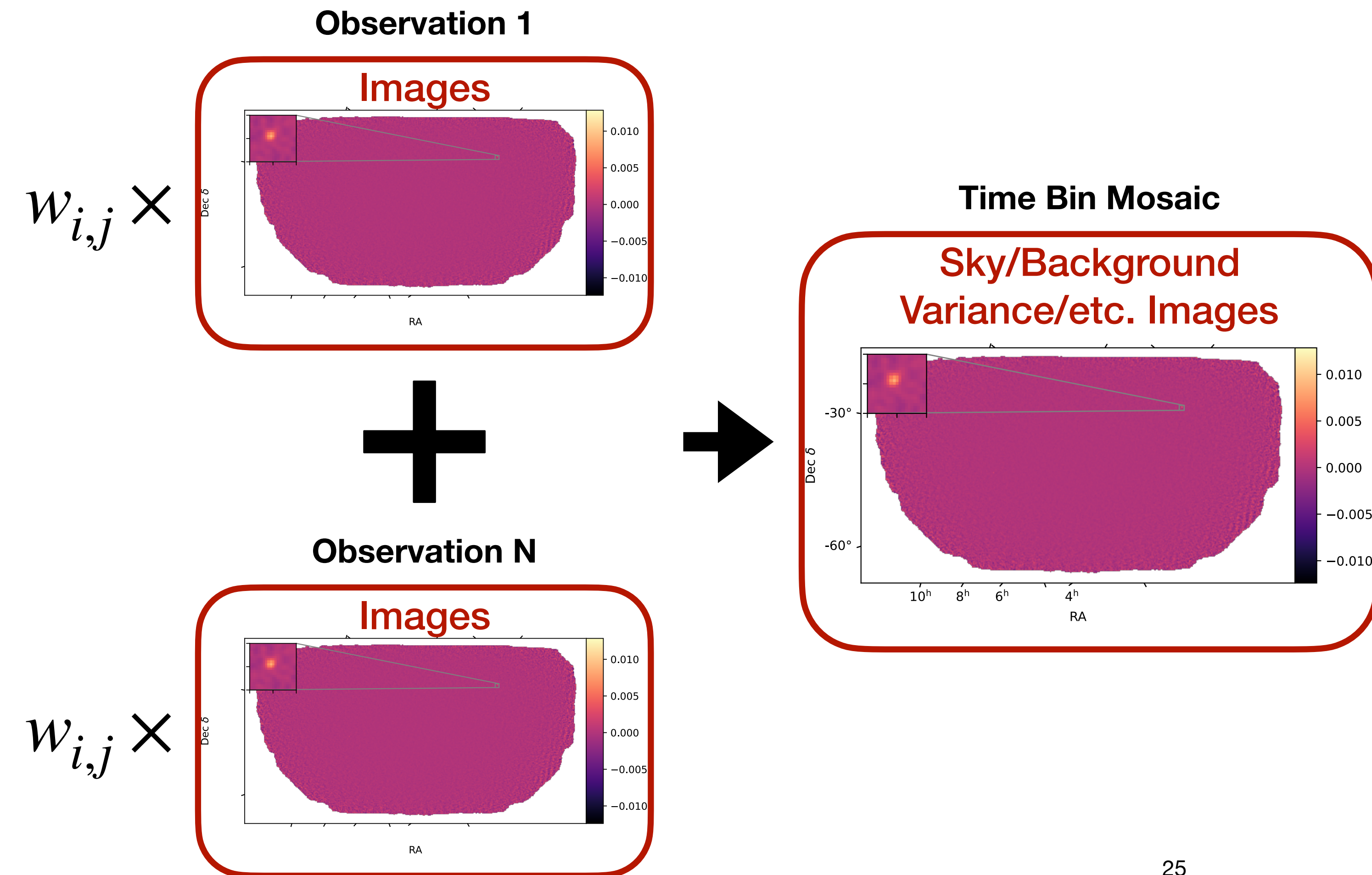


$w_{i,j} \times$

Survey Observation Processing Steps

2. Image Processing - Mosaic Survey Datasets

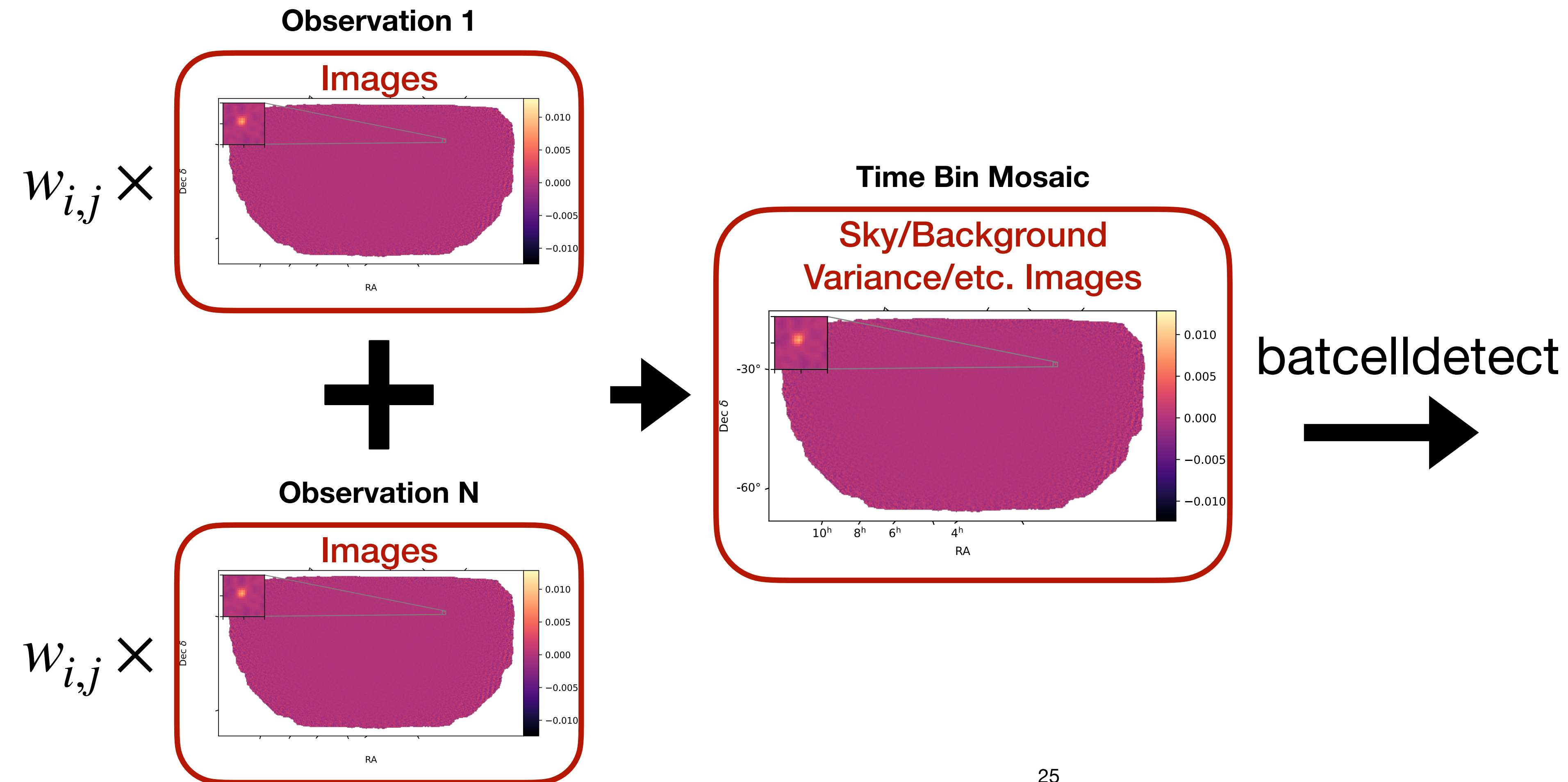
Observations within some time bin



Survey Observation Processing Steps

2. Image Processing - Mosaic Survey Datasets

Observations within some time bin



Survey Observation Processing Steps

2. Image Processing - Mosaic Survey Datasets

Observations within some time bin

