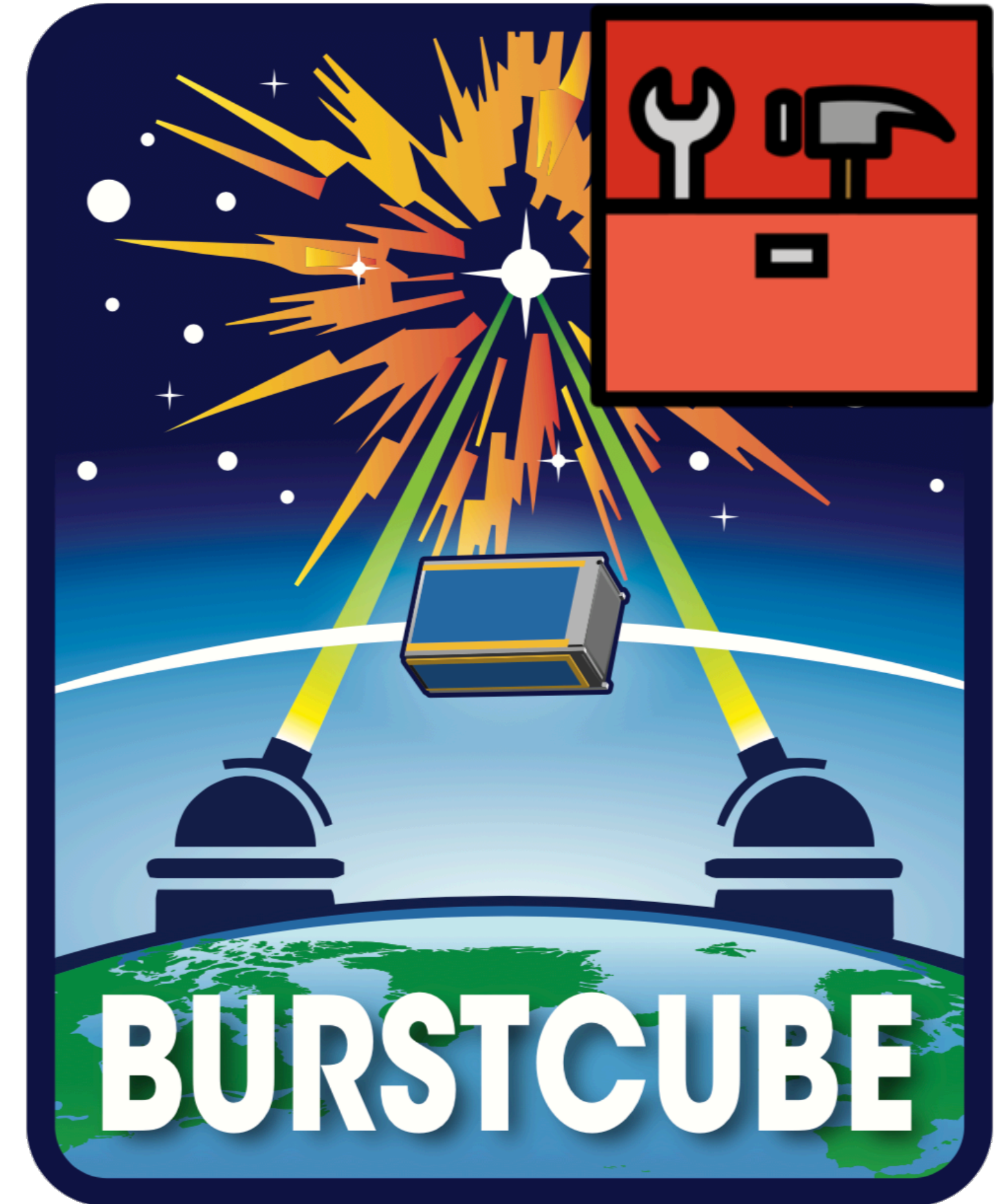


```
import bctools
```

status

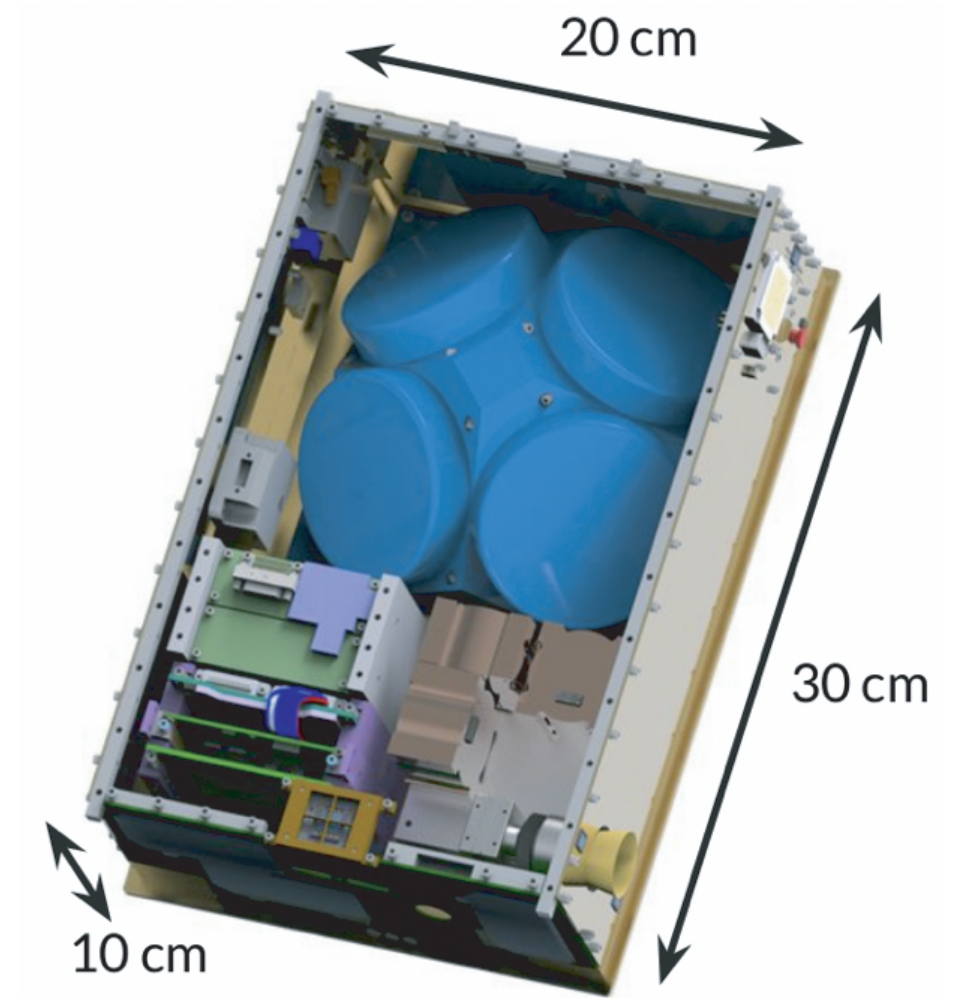


Israel Martinez - Sep 6, 2022 - GRB SmallSats, Brno

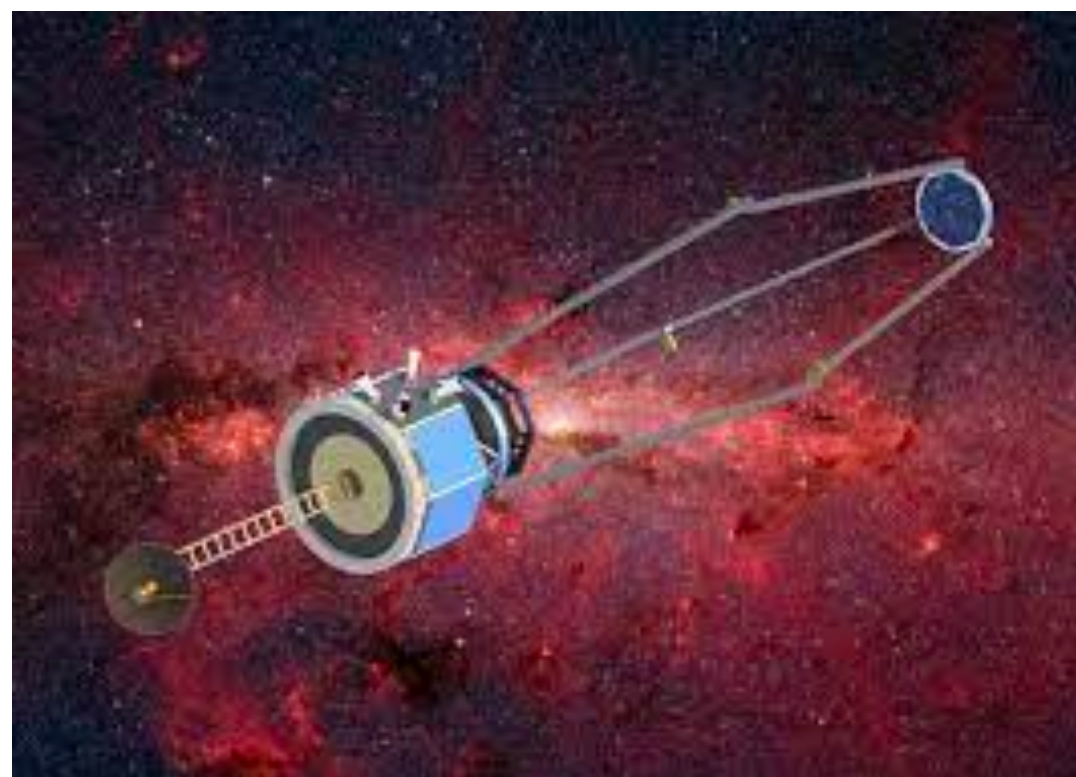


# What is [bctools](#)?

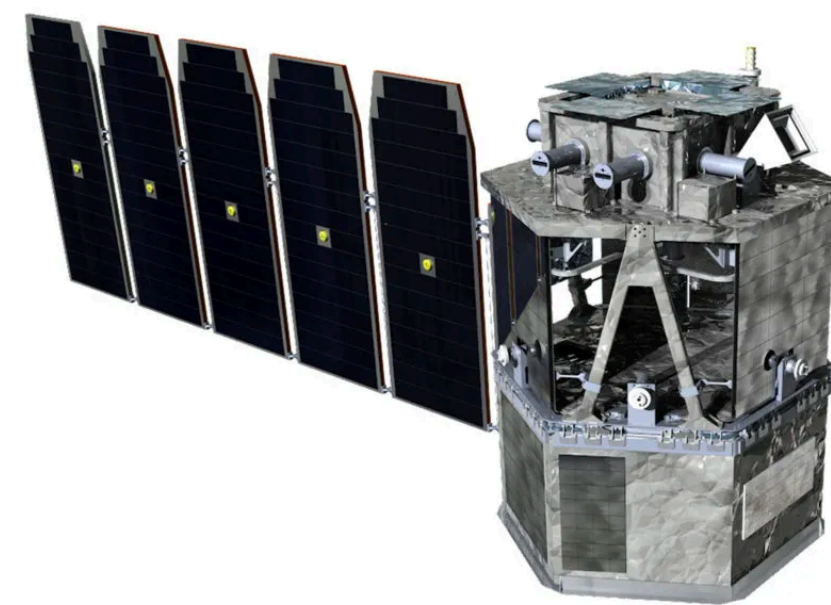
- BurstCube's ground analysis software
- Designed to do common tasks for count-based instruments
- Using GBM data tools as the starting point. Adding:
  - Detector response generator
  - Localization
- Detector agnostic. Some parts used in:



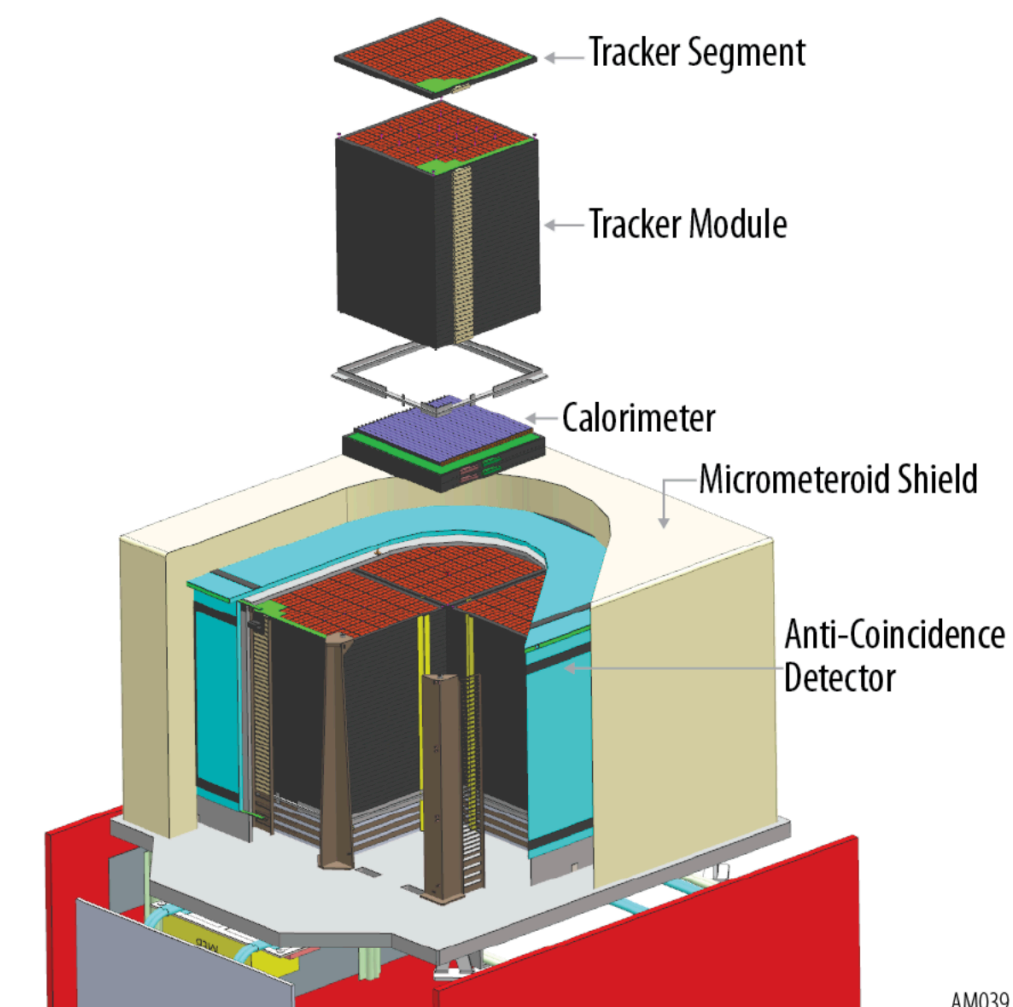
BurstCube



GECCO



COSI

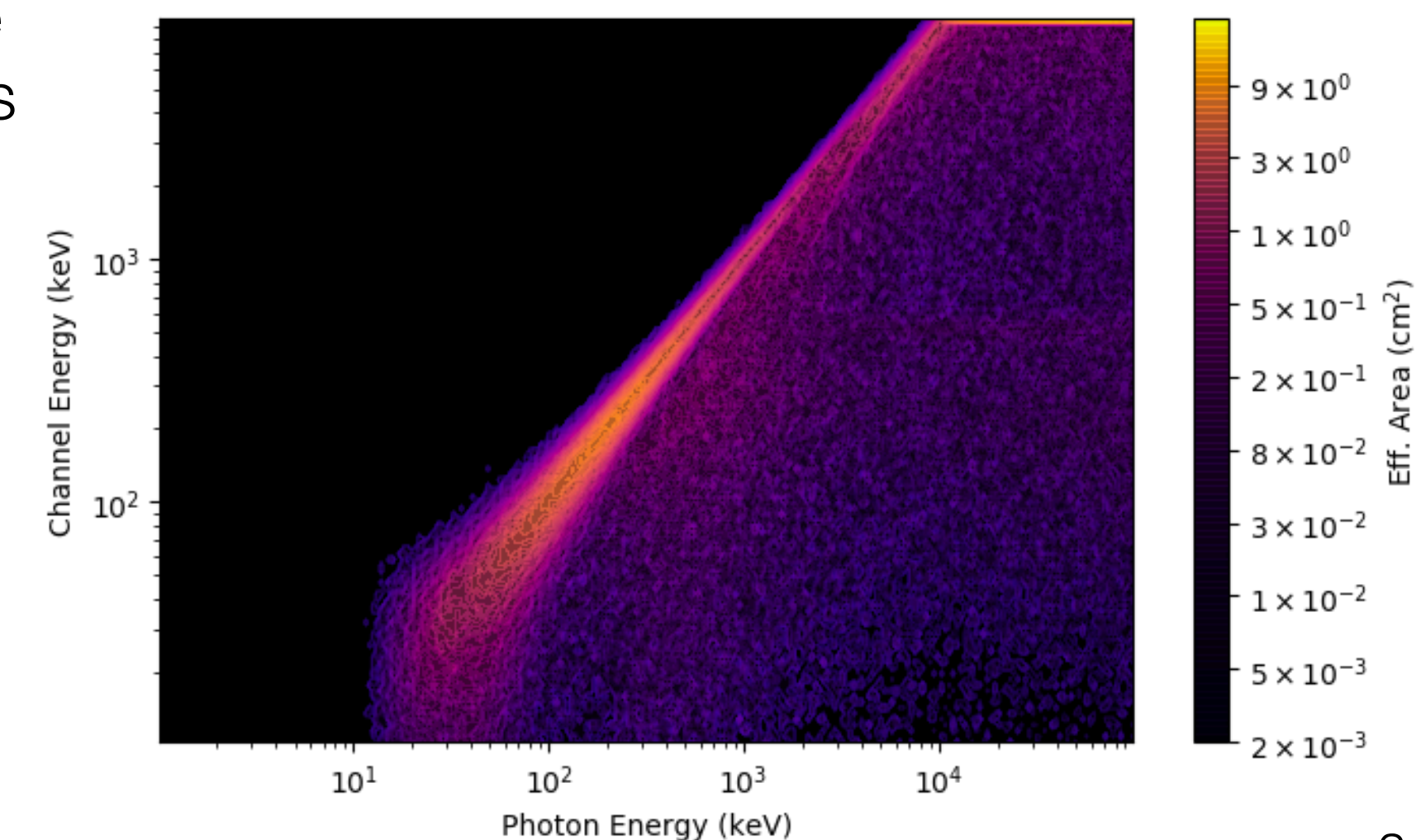
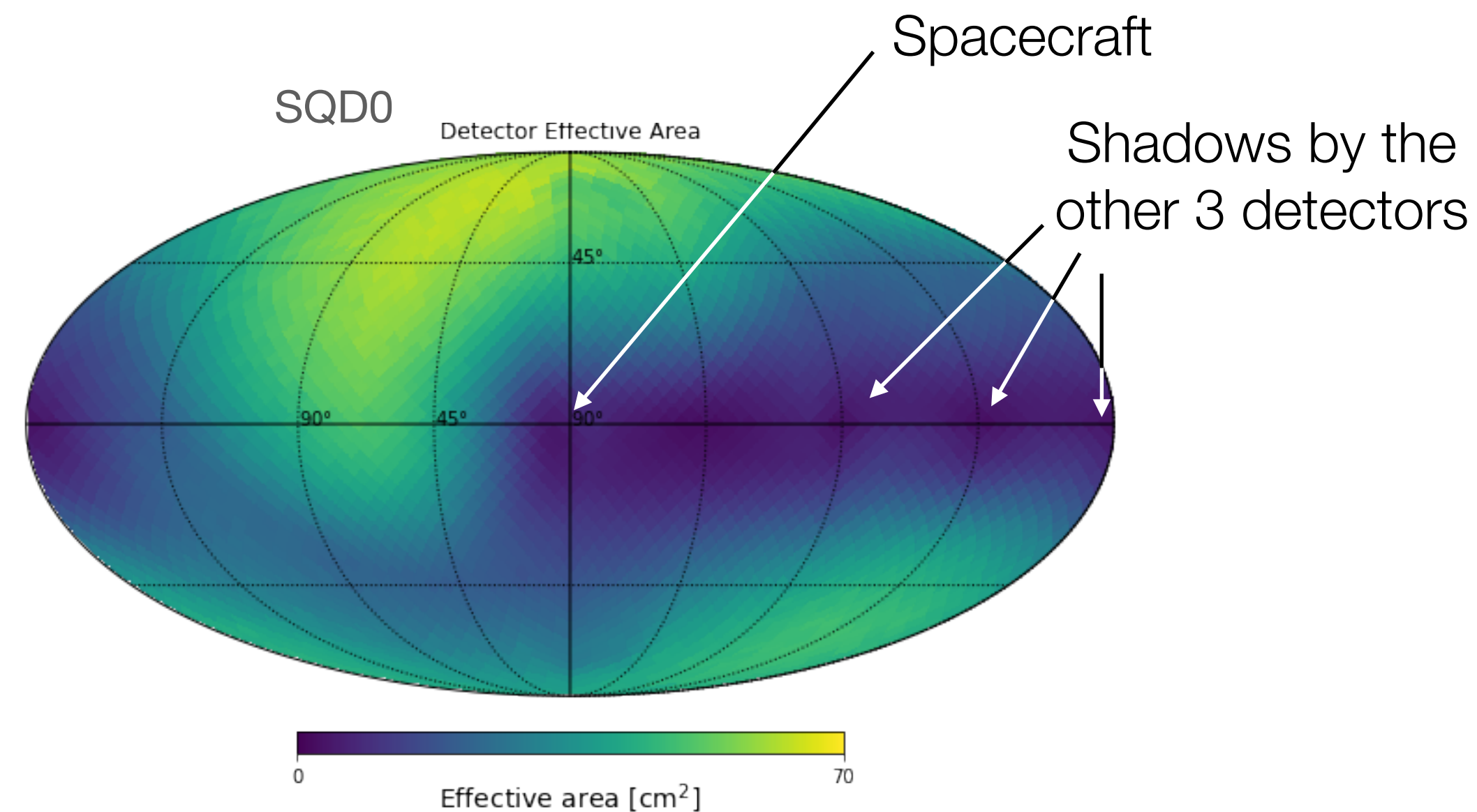
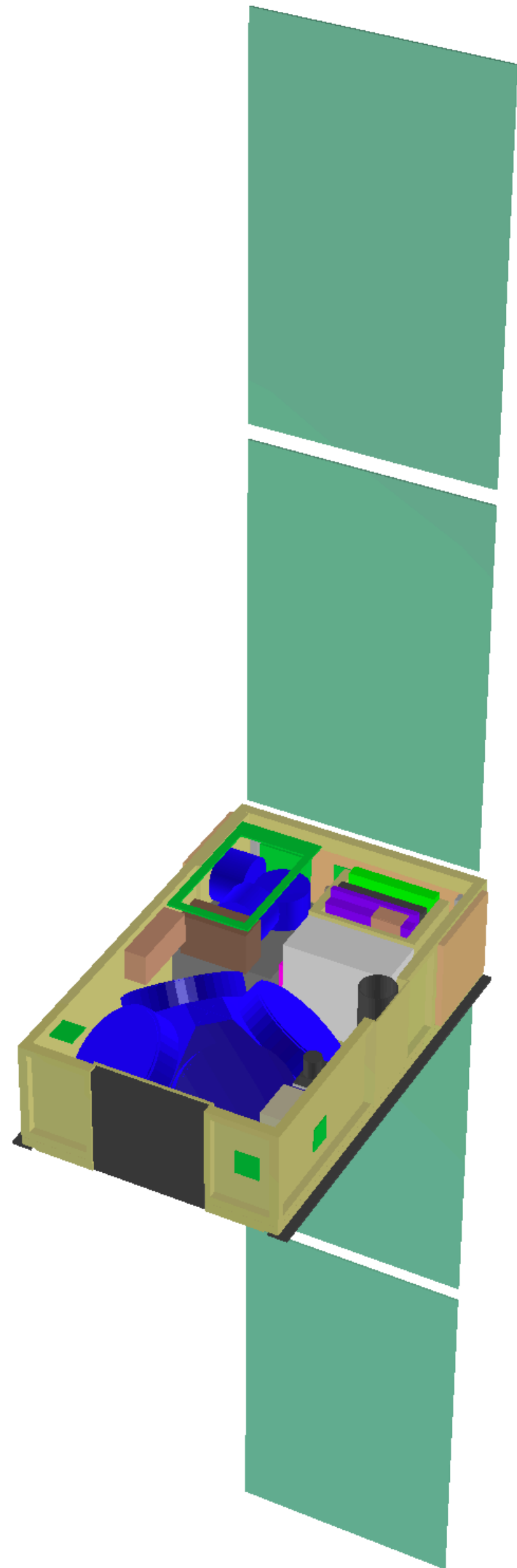


AMEGO-X



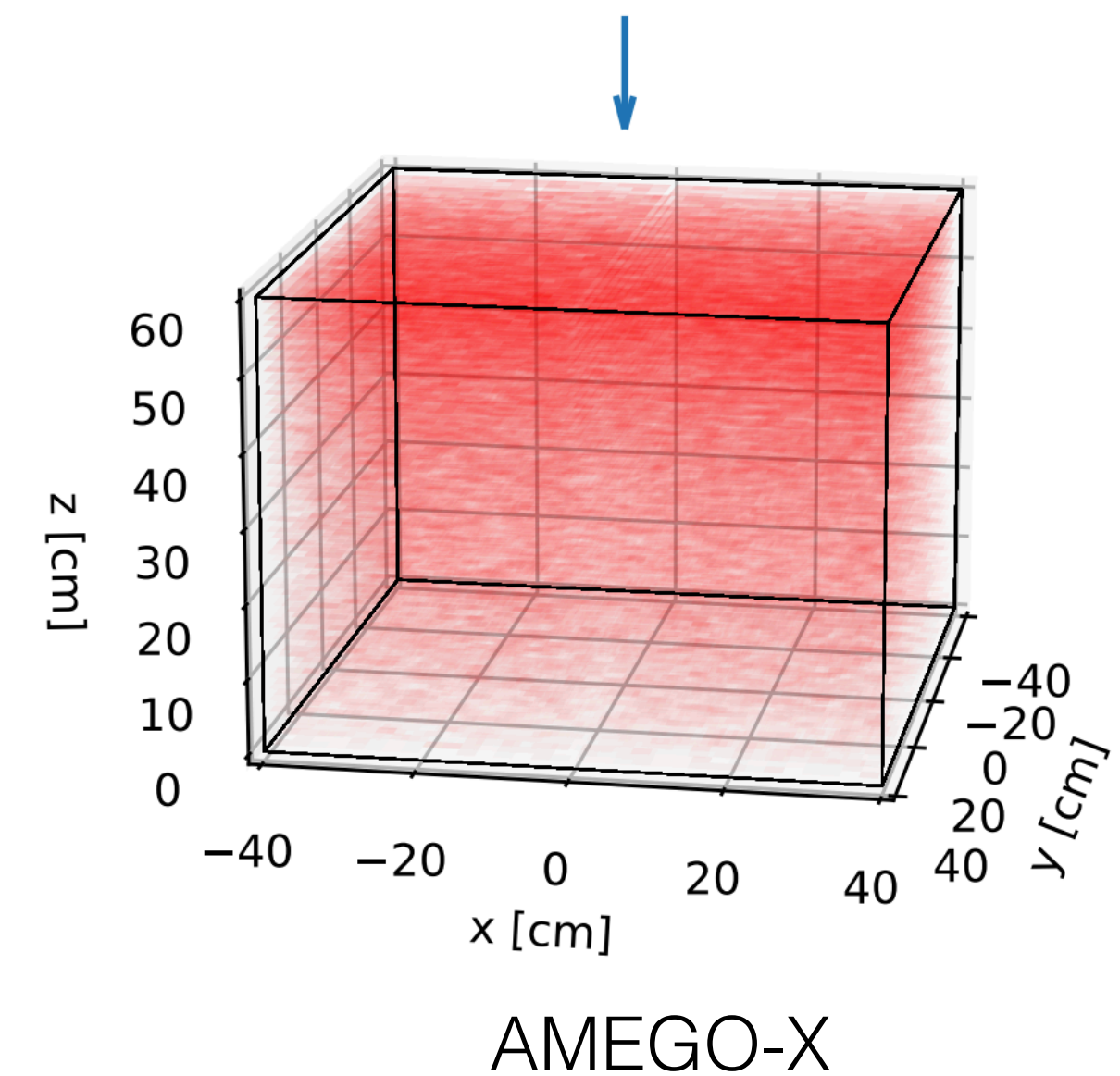
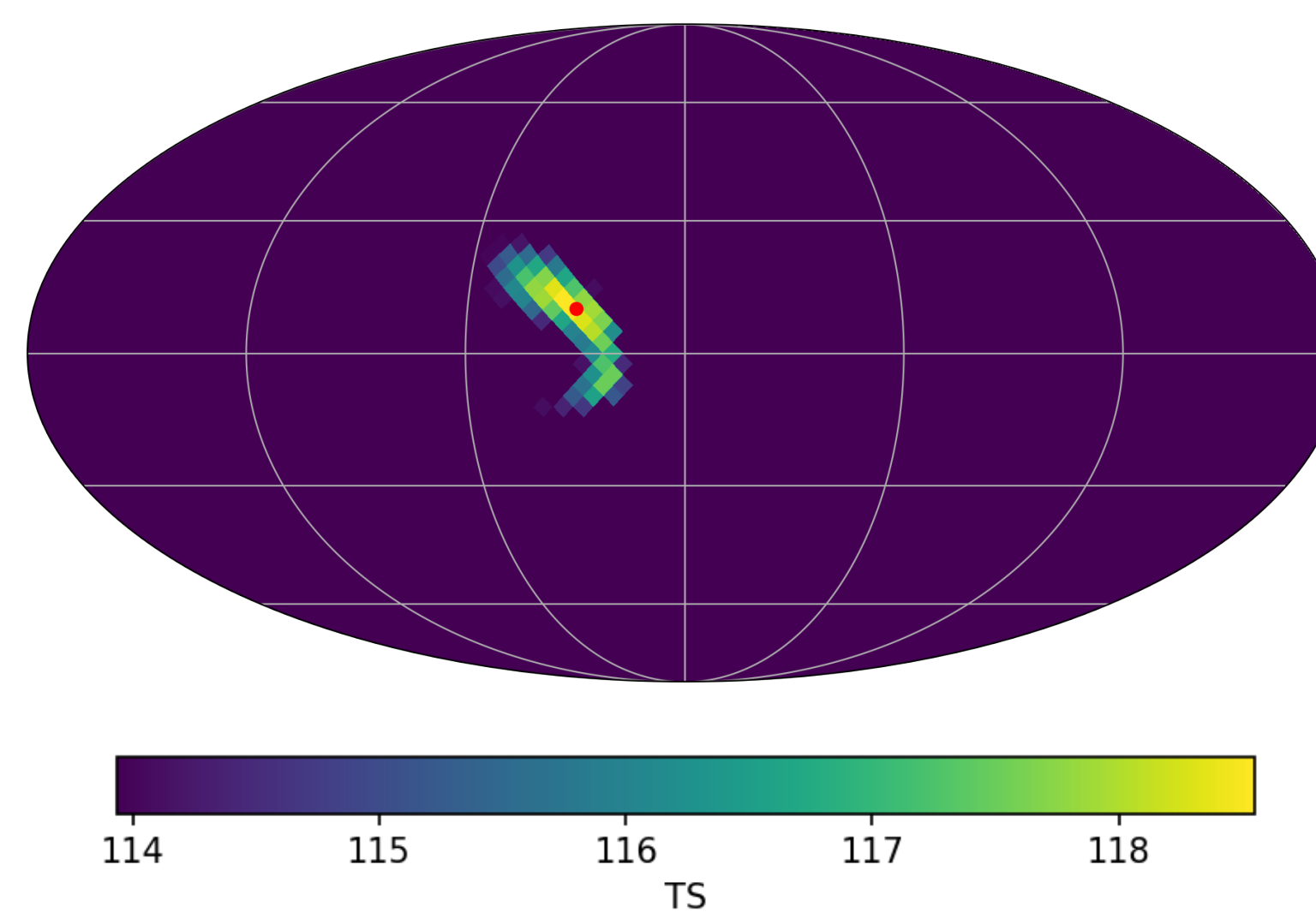
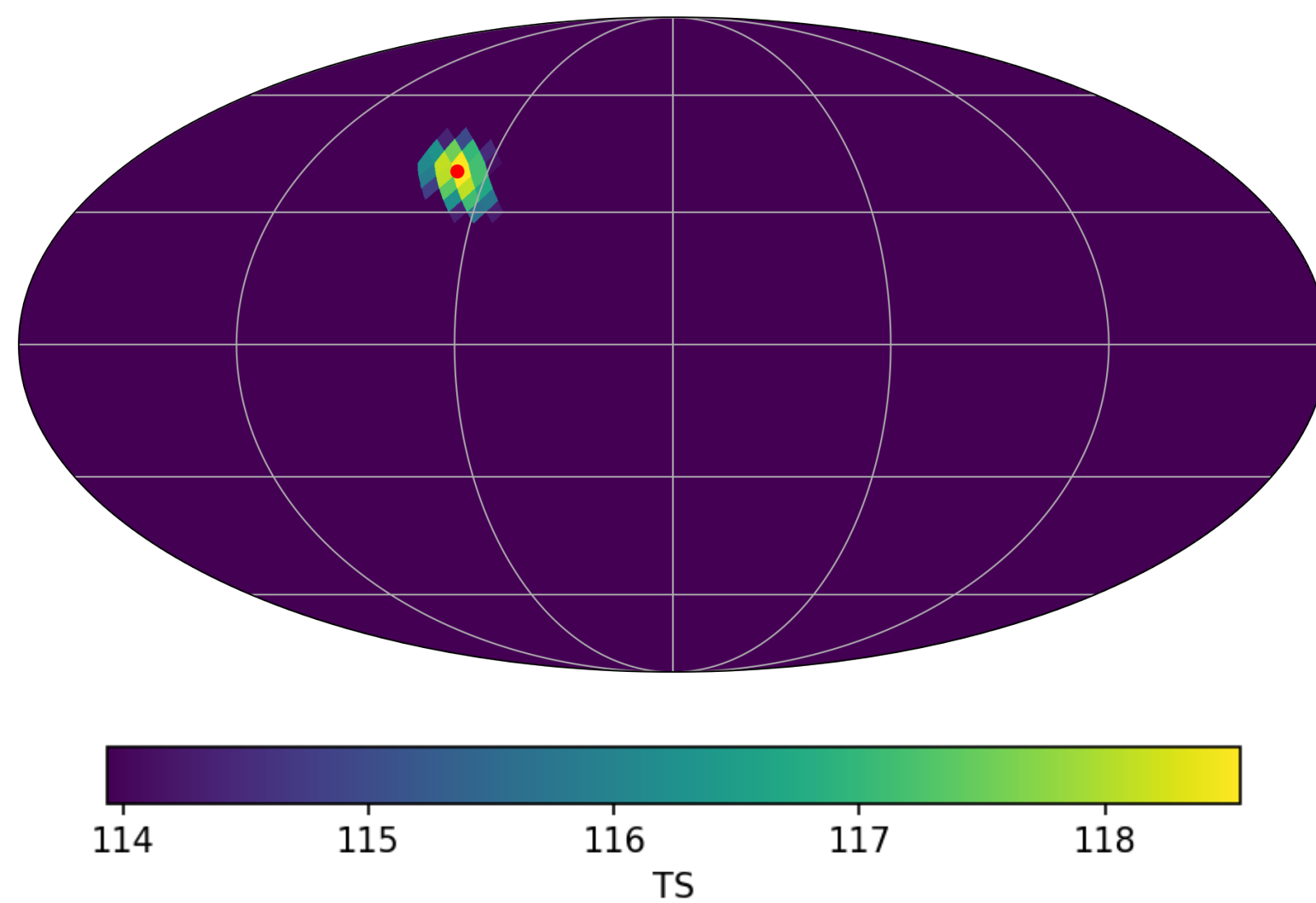
# Detector response generator and handling

- Built upon MEGAlib (a wrapper for GEANT4)
  - Needs mass model as input
  - Simulates all attenuation and internal scattering effects
- Multi-dimensional matrix encoding the effective area and energy migration:
  - For each detector and each direction
- First step for spectral fits and source localization



# Localization (& sub-threshold search)

- Source localization using the aggregate signal (i.e. like BATSE and GBM)
- Coherent Poisson-based maximum-likelihood analysis
  - As opposed to  $\chi^2$ , it can be used with low counts
  - Doubles as a sub-threshold search.
- Returns HEALPix maps

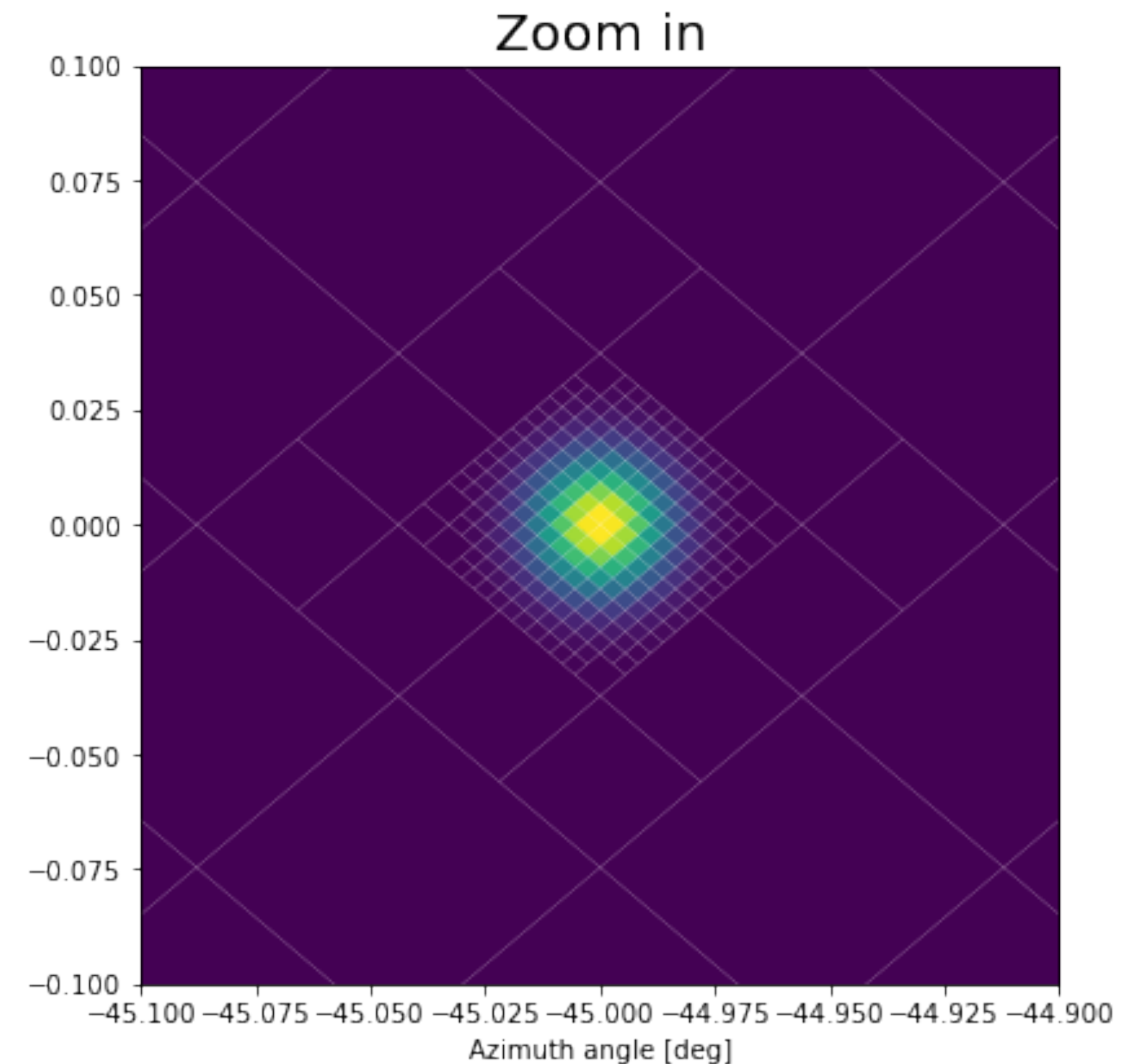
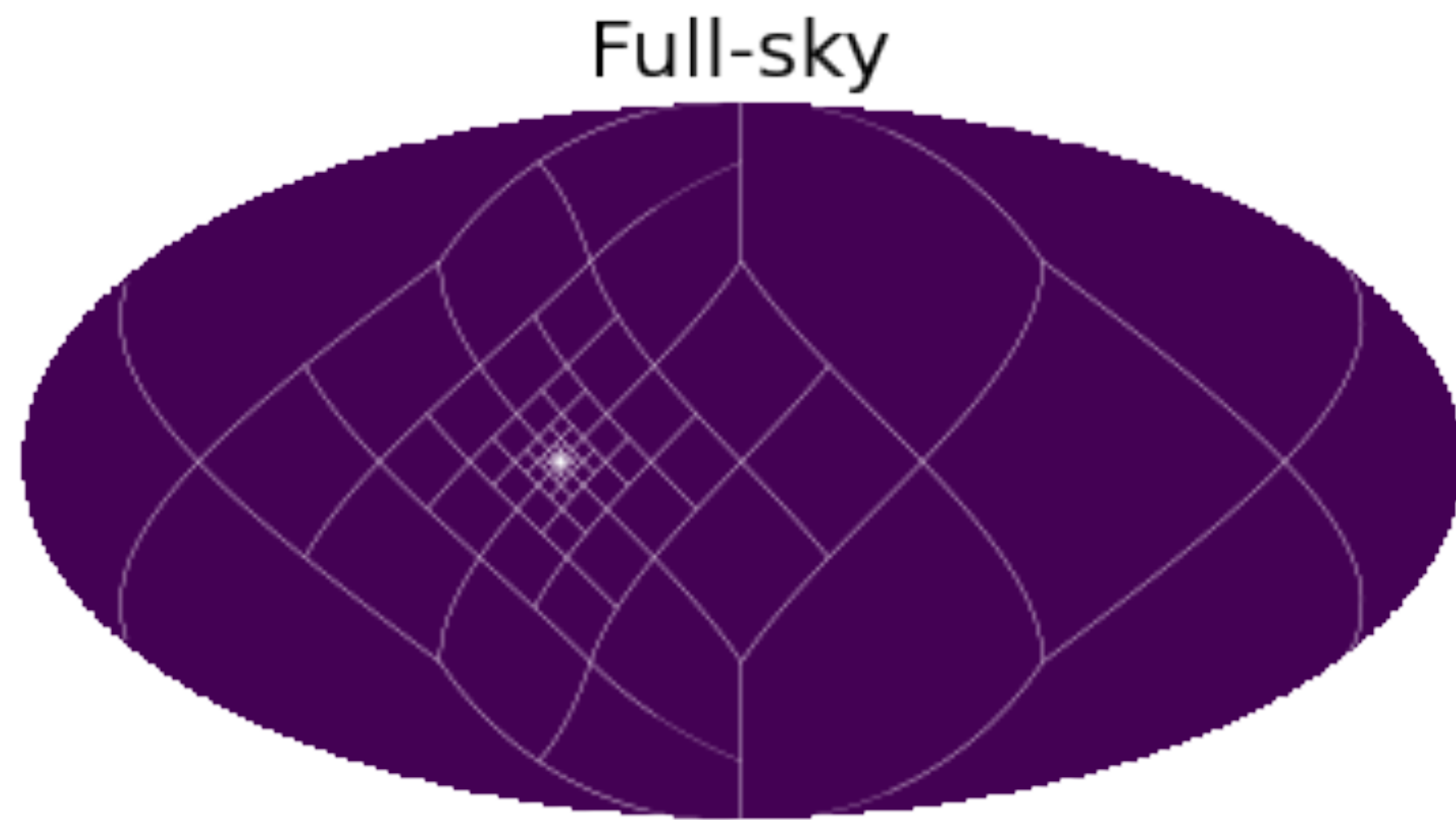


See <https://doi.org/10.3847/1538-4357/ac7ab2>



# Derived project: [mhealpy](#)

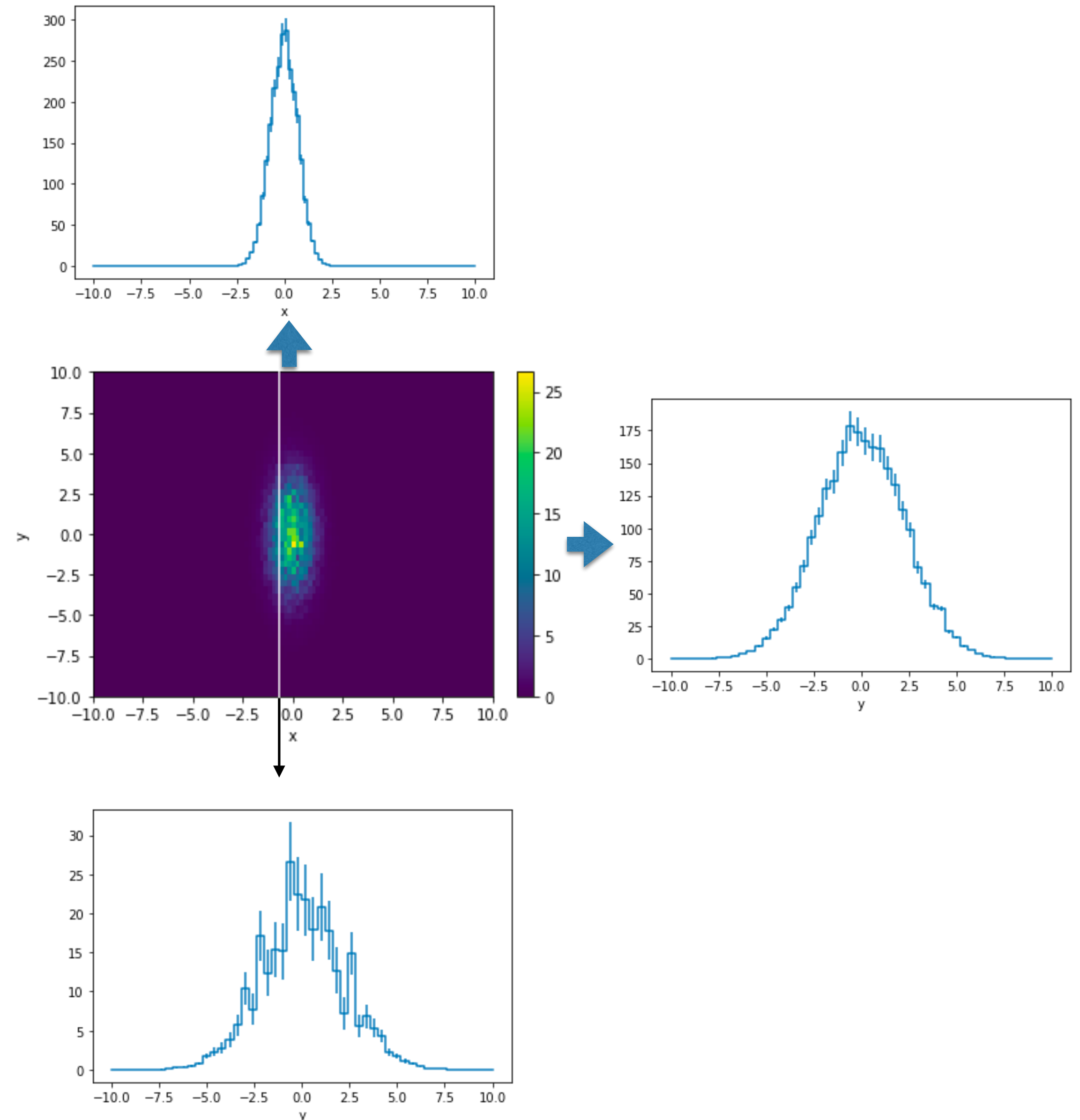
- An object oriented wrapper for healpy (a wrapper for HEALPix)
- Handles multi-resolutions maps (grid agnostic)
- Other features:
  - Binary operations between maps
  - World Coordinate System (WCS) compatibility
  - Compatible with astropy coordinates and units



See <https://doi.org/10.3847/1538-3881/ac6260>

# Derived project: [histpy](#)

- ROOT-like histogram (the good stuff only...)
- Allows for common operations when building and handling a detector response:
  - Filling
  - Weighting
  - Project, slice, concatenate
  - Convolutions
  - Tracking under/overflow
- Other features:
  - Sparse matrices (specially important for MeV instruments)
  - Compatible with astropy units
  - Binary operations
  - Plotting



# Other derived projects

## scoords

- Transformations to and from spacecraft coordinates
- Custom frame class extends all of astropy's machinery
- Multiple conventions to specify the attitude

```
from scoords import SpacecraftFrame, Attitude
import astropy.units as u

from astropy.coordinates import ICRS

attitude = Attitude.from_quat([0,1,0,1])

c = SpacecraftFrame(lon = 10*u.deg,
                    lat = 45*u.deg,
                    attitude = attitude)

c.transform_to(ICRS())
```

```
<ICRS Coordinate: (ra, dec) in deg
(9.85107612, -44.13602946)>
```

## yayc

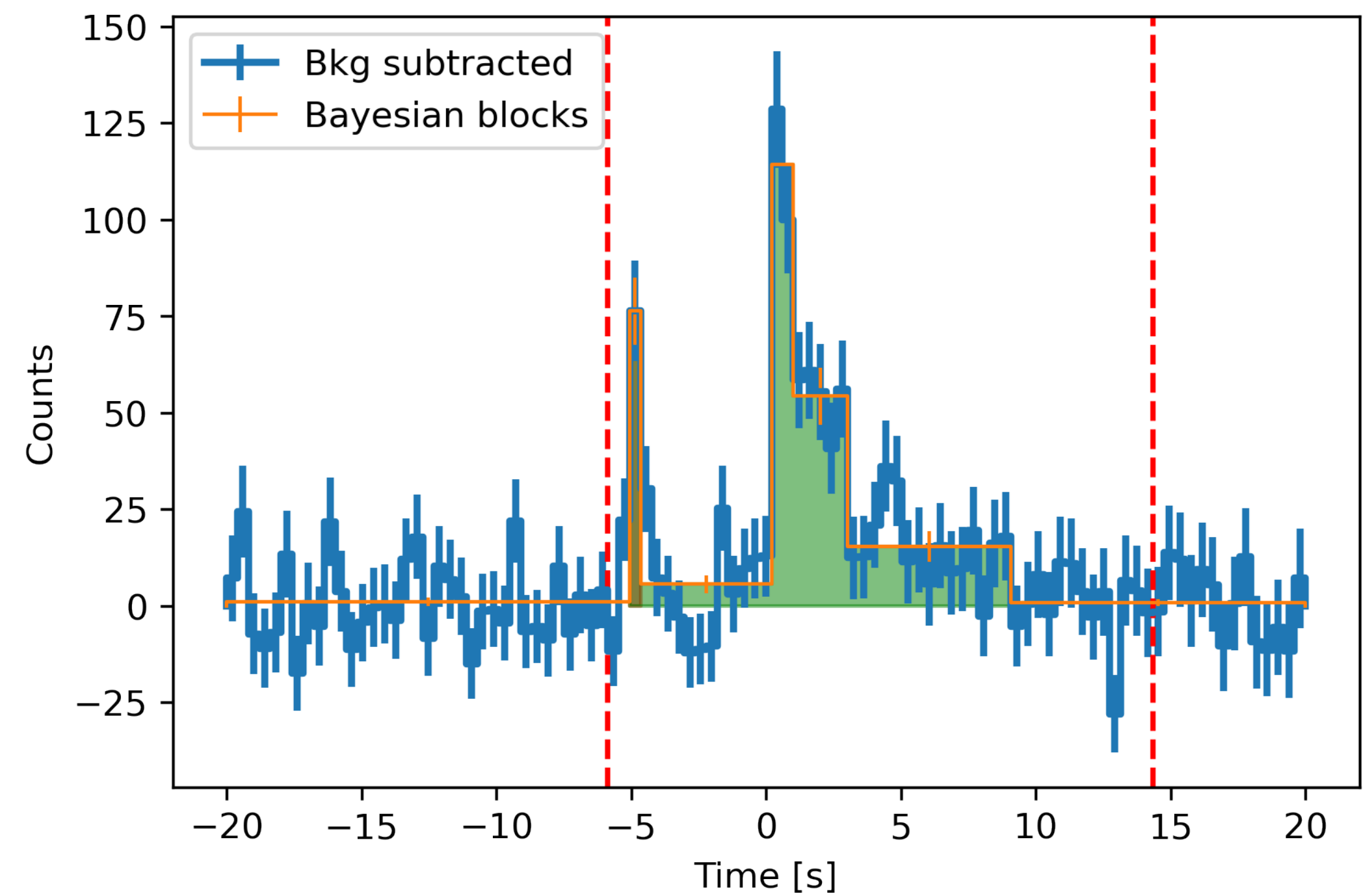
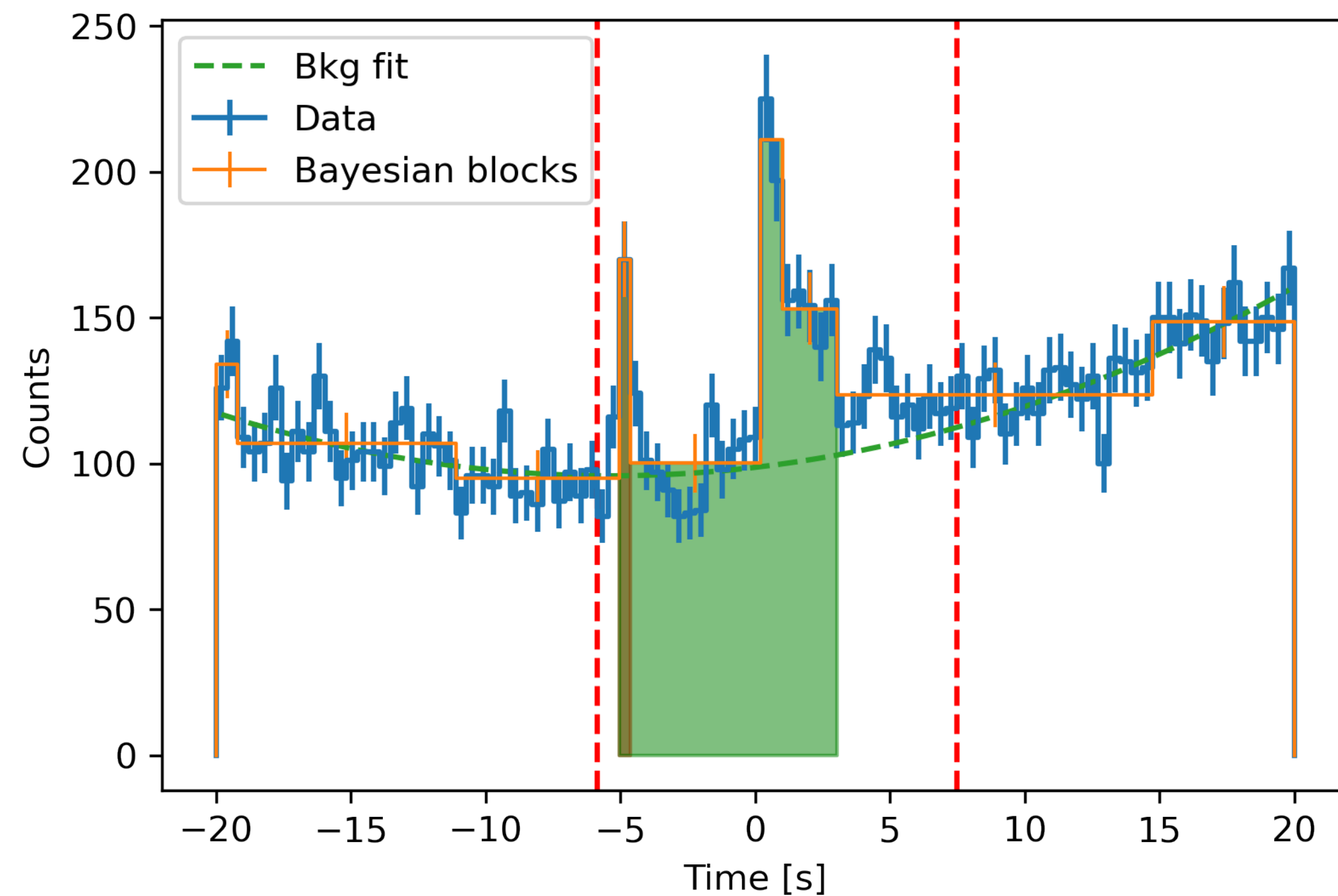
- Yet another YAML configurator
- Handles paths relative to the config file
- Allows user to easily override any option in the command line

```
35 # Detector effects such as energy resolution and efficiency
36 # See SimDetectorEffect class for options
37 detector_effects:
38   order: ["efficiency", "energy_resolution"] # Names must match entries in 'effects'
39   effects:
40     efficiency:
41       name: "SimEfficiencyFromPoints"
42       args:
43         energy: [1.50E+01, 2.00E+01, 3.00E+01, 4.00E+01]
44         efficiency: [ 0.00, 0.20, 0.50, 1.00]
45     energy_resolution:
46       name: "SimEnergyResolutionFromFit"
47       args:
48         # These coefficients are the ones reported in ICRC 2019 proceedings,
49         # multiply by 10 so they are in keV (not MeV) and fractional (not %)
50         # You can also specify one list per detector
51         coeffs: [4.00E-02, 1.93E+00, 1.00E+01]
52
```

# Things that are still in progress



- Event duration calculation
  - A variation of the Bayesian blocks algorithms seems to work pretty well
- Event classification
  - Not much progress on this yet. Ideas welcome.





# Final remarks

- bctools is open-source and detector agnostic
- bctools is still under development
  - If something seems useful to you, let's talk!
- We were funded to merge bctools and GBM data tools (among other things)
  - Let's join forces

