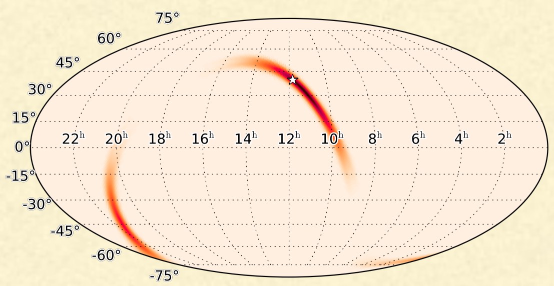
Finding gamma transients in simulated GW follow-ups with Gammapy

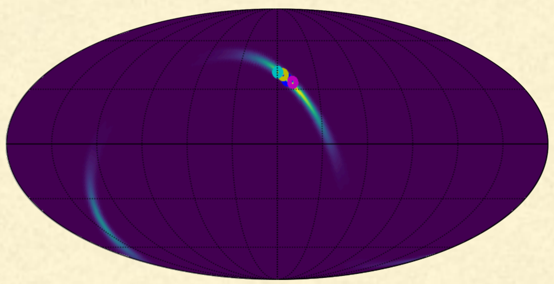
Monica Seglar-Arroyo on behalf of IRFU/CEA Paris-Saclay team

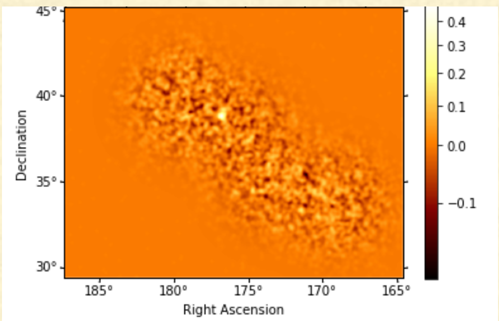
Goal

In a nutshell: Observe extended regions, following the GW reconstruction area and look for a transient signal

Simulation chain







- Detection rates dependency of injected source luminosity, GWparameters (90%rec. area..)
- Sensitivity plots
- etc

☑ Done

☑ Done

☐ In progress

☐ In the near future..

Pointing pattern covers source location?

Is source detected?

Bank of GW simulated signals + NSNS locations, distances, inclinations, detectors involved..

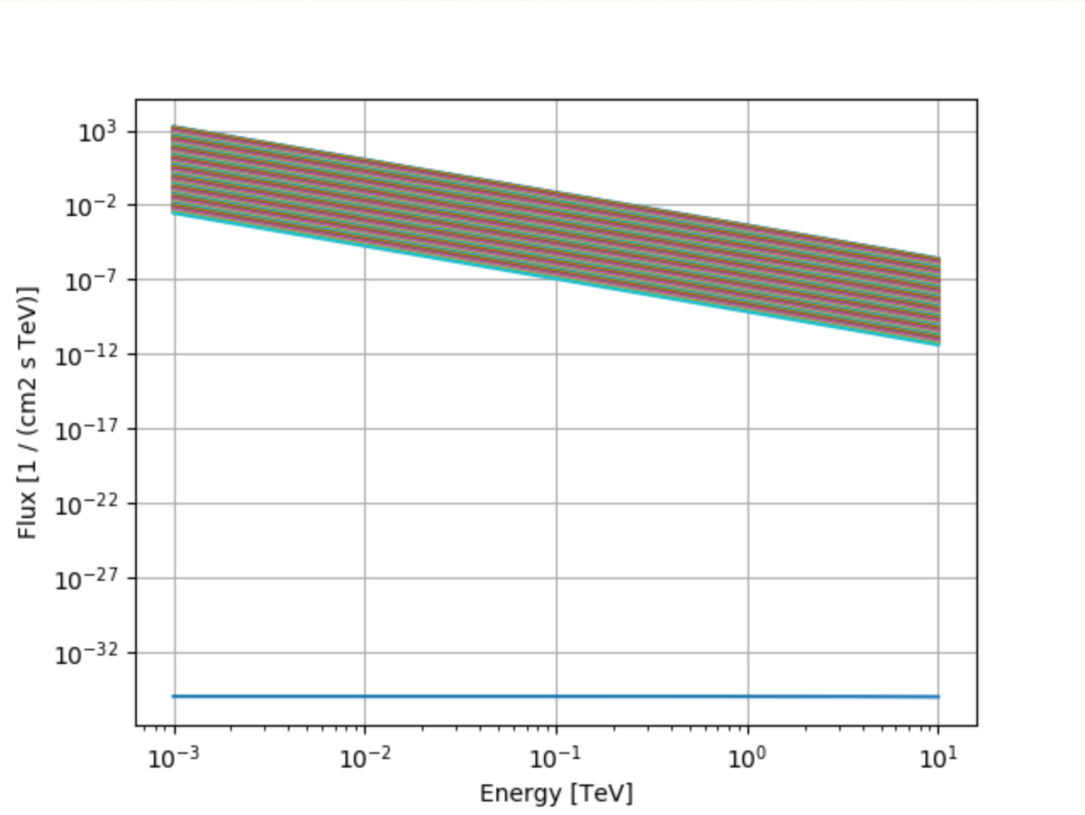
Observation algorithms considering visibility constrains, zenith angle optimisation, galaxy distribution defines pointing strategy

Input
- Phenomenological GRB model
- Source position
- Pointings
- Duration
- IRFs

Output
- Source detection and dependency of input+source parameters

Injection of a gravitational wave counterpart aka low redshift GRB

Injected source spectrum:



- Evolution with time [0, 10000s]

- Based of an interpolation to lower z of one of the brightest GRBs observed: GRB090510
  - E_{iso} 10^{52} ergs
  - Shows extended emission(~200s) up to GeV energies (~30GeV)
  - Opening angle 10deg, on-axis GRB

For the simulation:

- The spectrum is EBL deabsorbed.

- IRFs are chosen for a given simulation which depend on zenith angle and telescope config. Example: North_NSBx05_z20_N_LST_30m

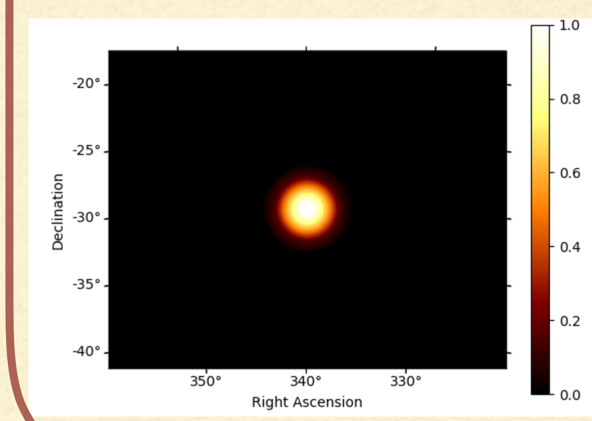
First steps on simulation

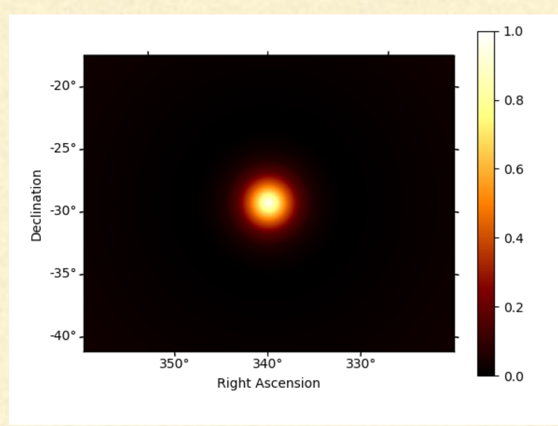
- Map simulation: single, simple cube simulation (extracted from Notebooks)

- Pointing
- Livetime
- IRFs
- Cube geometry
- Sky Model

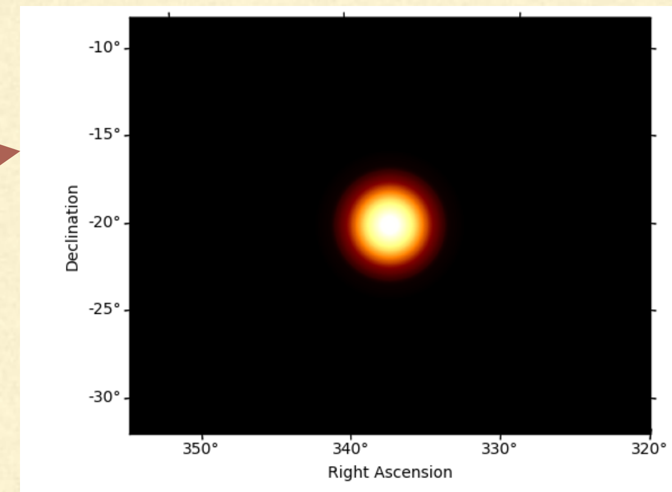
MapEvaluator

- PSF Kernel
- Background
- Energy Dispersion
- Exposure

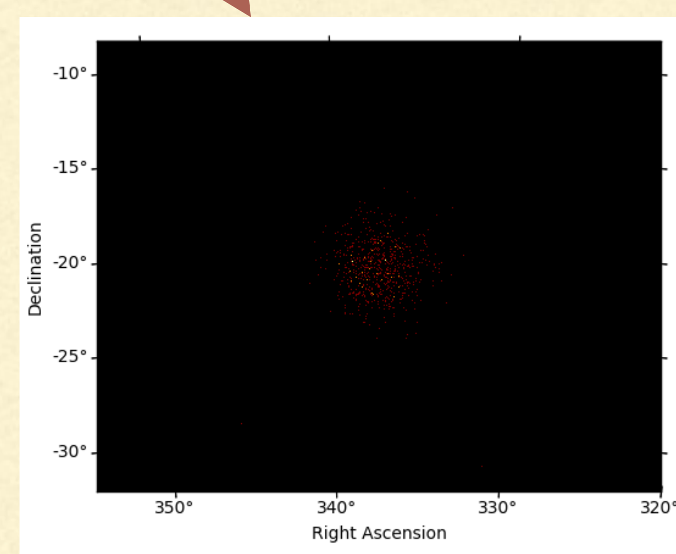




Predicted counts



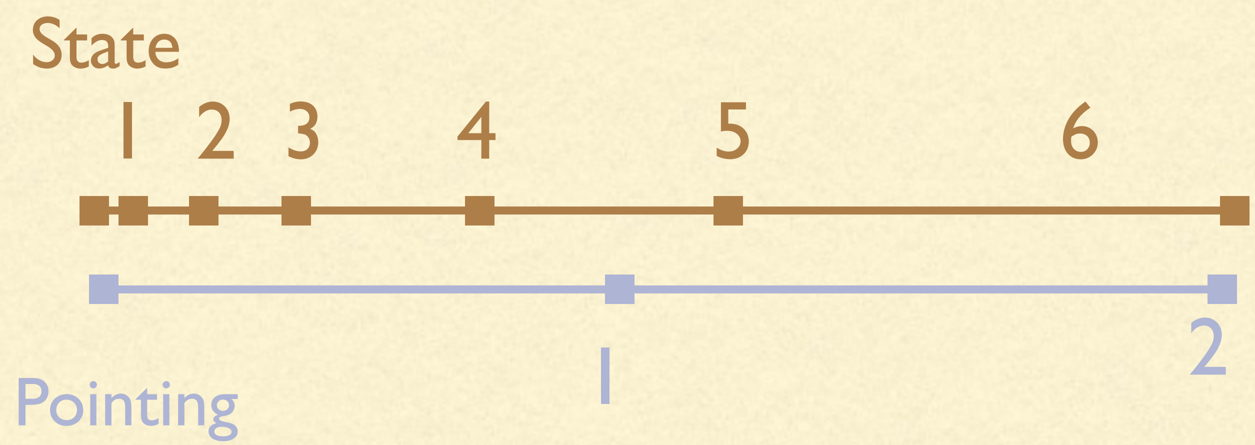
Randomise predicted counts

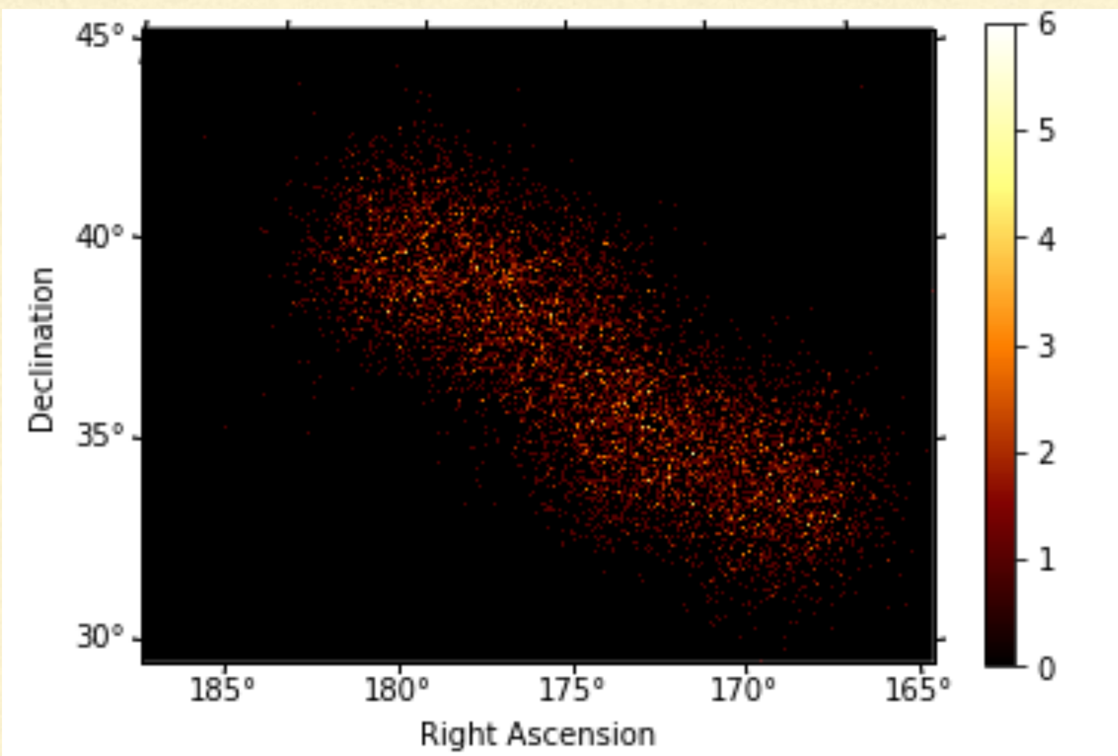


Counts

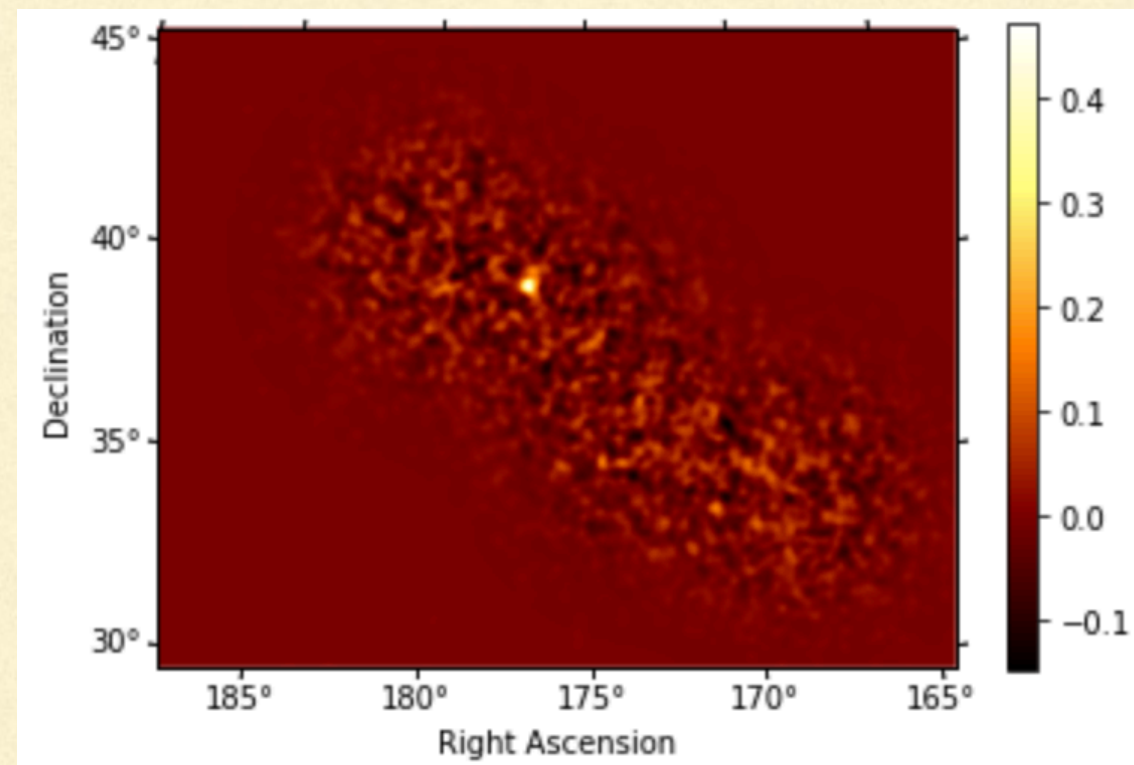
Extended observation simulation

- Combination of several observations, with several states of the source per observation





Counts



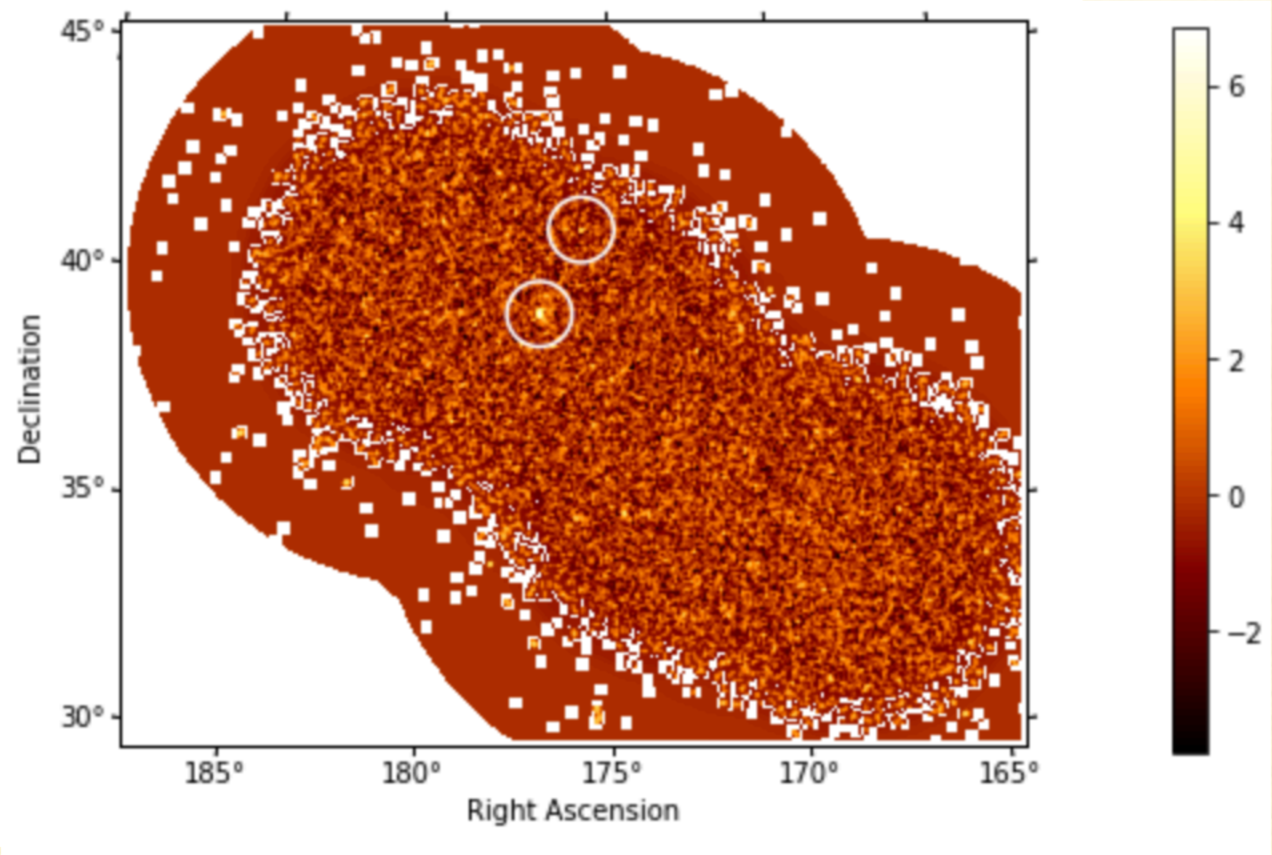
Excess map

Finding the source

1. Merge all observations together

2. Tried: → Li&Ma computation with Tophat2DKernel
   → TSMapEstimation using a Gaussian2DKernel

3. Look for peaks in significance



Sqrt TS

value	x	y	ra	dec
			deg	deg
float64	int64	int64	float64	float64
6.8406	463	475	176.94583	39.50629
4.0593	509	568	175.74726	41.36974

Next step: TIME

- This is the approach I found by looking into Jupiter Notebook tutorials + some docu.

- **Time-related** improvements:

  - Looking for a better way to simulate an evolving source
    - I do a simulation of every source state and add them up, not sure if in the most efficient way.

  - Looking for nice clustering technique that includes time evolution
    - Clustering algorithms in (x,y,t)
    - Wavelets
    - ?

  - I am working with **maps** and I dont have a list of simulated photons with times, etc. => Possible solution would be to use MapEventSampler explained by Fabio Pintore including a temporal evolution in sky_model as input option ?

Thanks!