Gammapy and CTOOLS hands-on

Gonzalo Rodriguez
Sezione INFN di Tor Vergata Roma

What are GammaLib-CTOOLS & Gammapy?

- There are both a high-level data analysis package for gamma-ray astronomy

- At first there have been development for CTA analysis

- But they have develop abstracts libraries that can handle data for different experiment.

What are GammaLib-CTOOLS & Gammapy?

- So far…

GammaLib-CTOOLS:

- COMPTEL
- Fermi/LAT
- Cherenkov telescopes (CTA, H.E.S.S., MAGIC, VERITAS)

Gammapy:

- Fermi/LAT
- Cherenkov telescopes (CTA, H.E.S.S., MAGIC)

Documentation

GammaLib-CTOOLS:

- http://gammalib.sourceforge.net/
- http://cta.irap.omp.eu/ctools/index.html

Gammapy:

- http://gammapy.org/

Both have also regular coding sprints for user and developers

Documentation

They both have very good tutorials:

(gammapy mainly notebooks)

There are forums for user and developers

Active mailing list

Regular eZuce meetings…

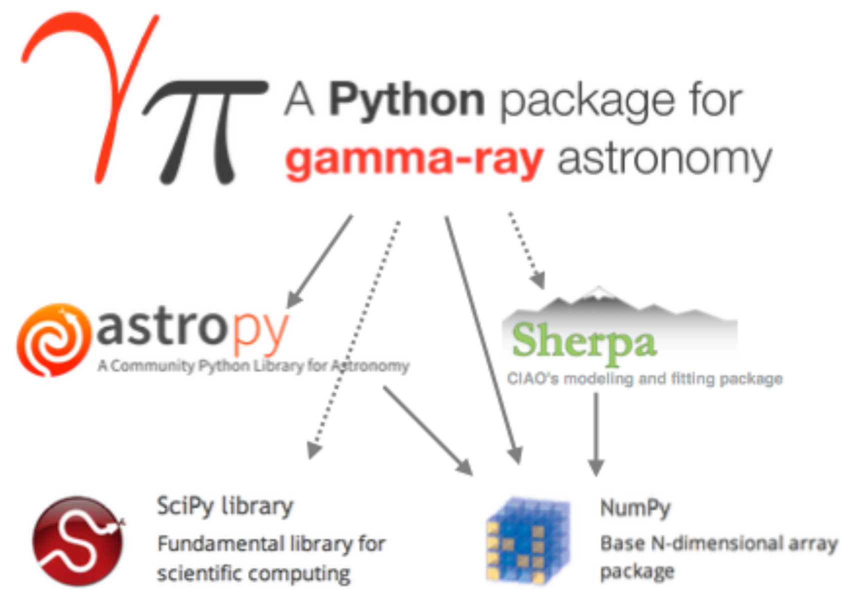
Also for DC-1 analysis (howtos, notebooks)

Different development philosophy
(From user point of view)

Gammapy is a python package built on Numpy & Astropy

There are both python standard Packages widely use for the Python scientific community.

In any case,
you have to program in python



python™

Different development philosophy

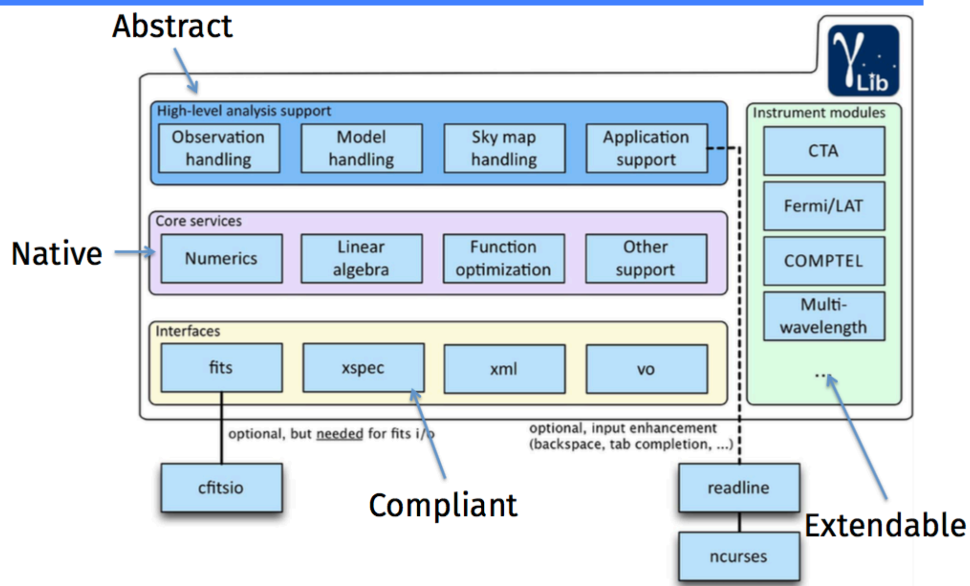
(From user point of view)

GammaLib & CTOOLS are develop using C++

They need just one external package (cfitsio)

They have python support

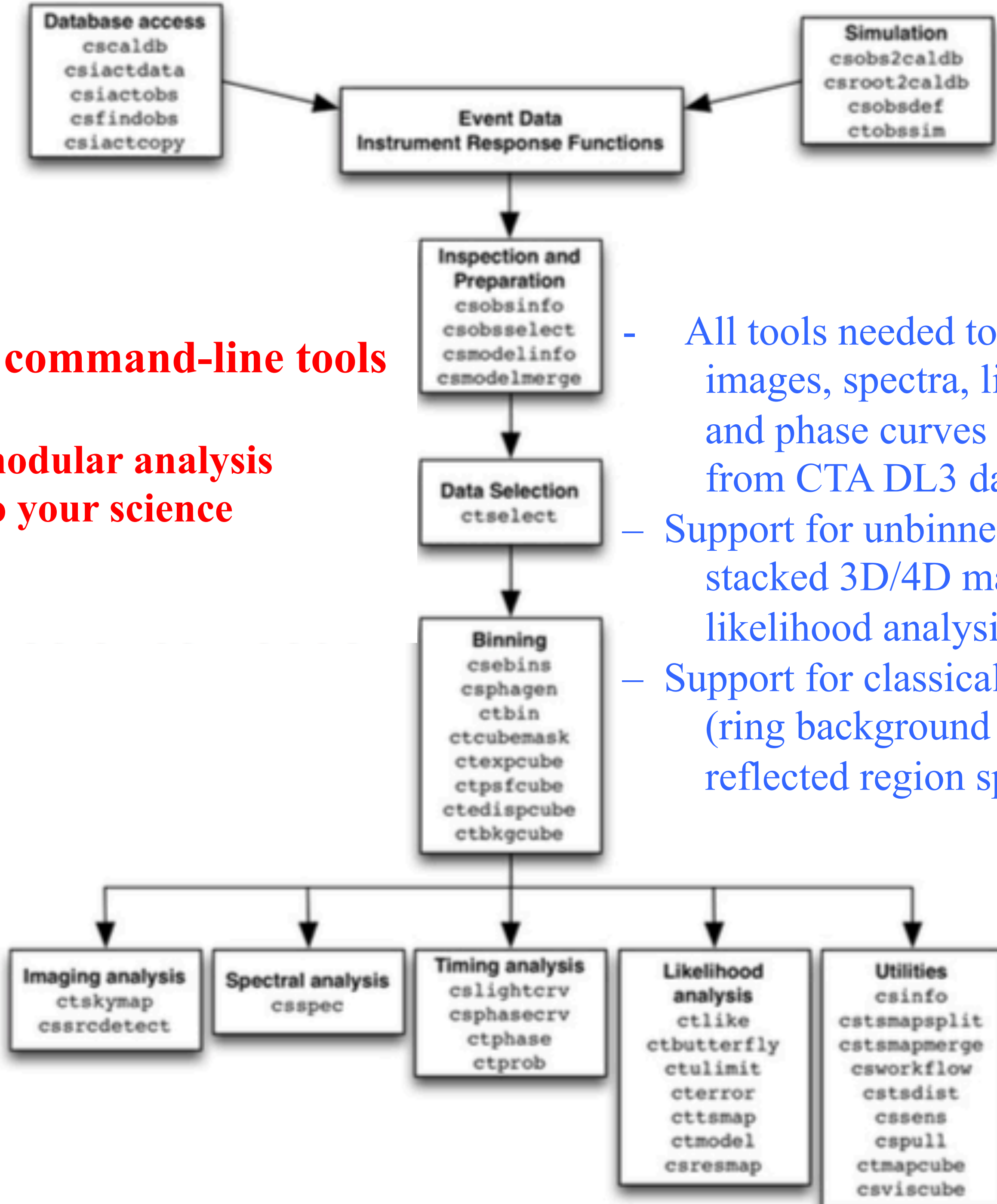
For the user <u>is not mandatory to know C++ or python</u>



CTOOLS:

They are a set of command-line tools

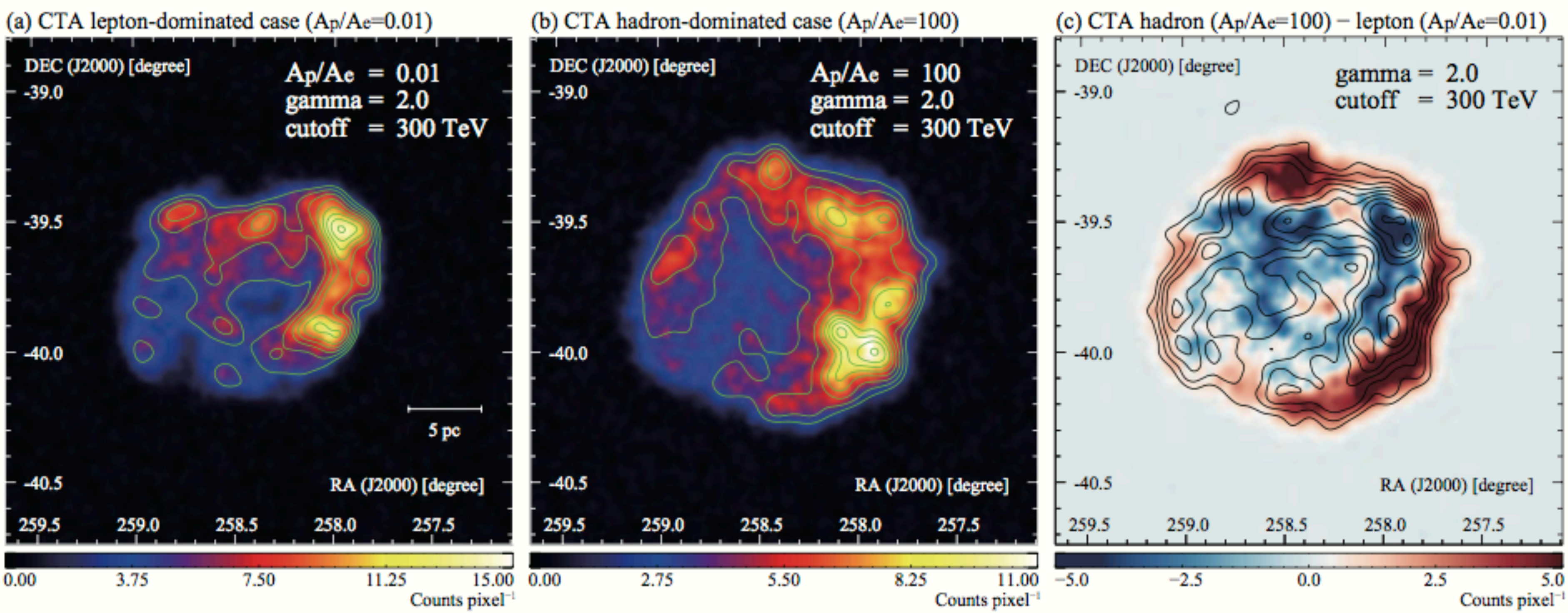
Bricks to build a modular analysis
Workflow suited to your science



- All tools needed to generate images, spectra, light and phase curves from CTA DL3 data
- Support for unbinned, binned and stacked 3D/4D maximum likelihood analysis
- Support for classical IACT analysis (ring background sky maps, reflected region spectra)

Some CTOOLS results

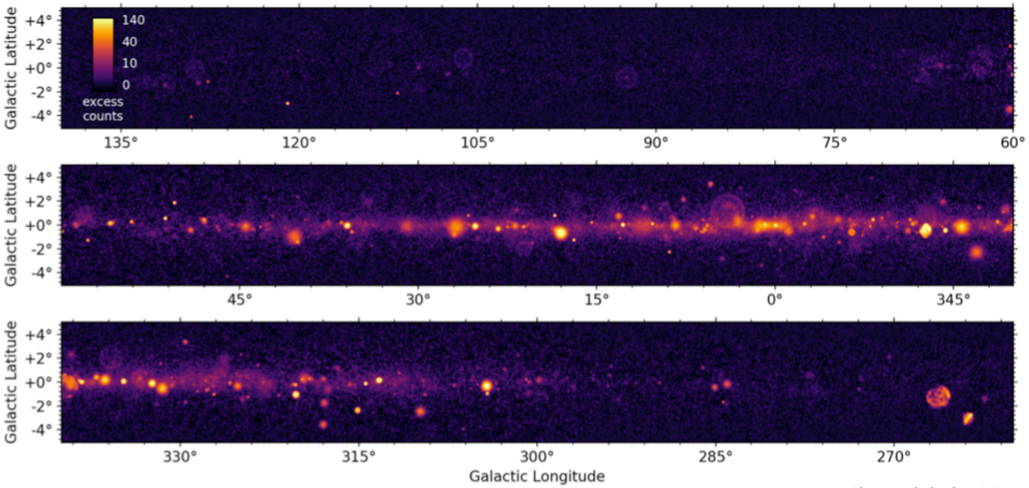
Comparison of hadronic versus leptonic models of RX J1713.7-3946 as seen by CTA



Credits: Acero et al. 2017, ApJ, 840, 74

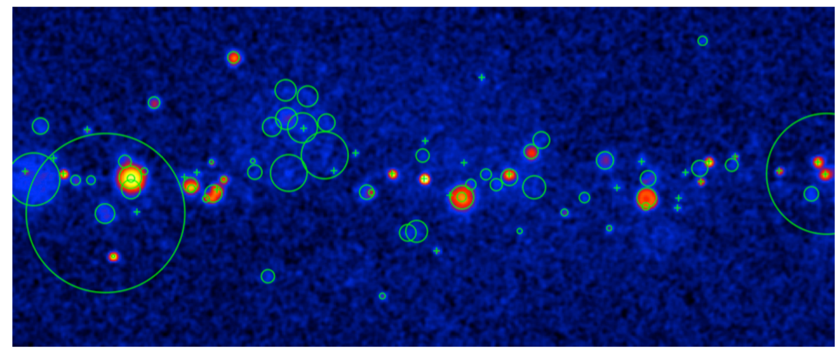
Some CTOOLS results

Sky map of the inner regions of the Galactic Plane Survey KSP above 1 TeV (DC-1)



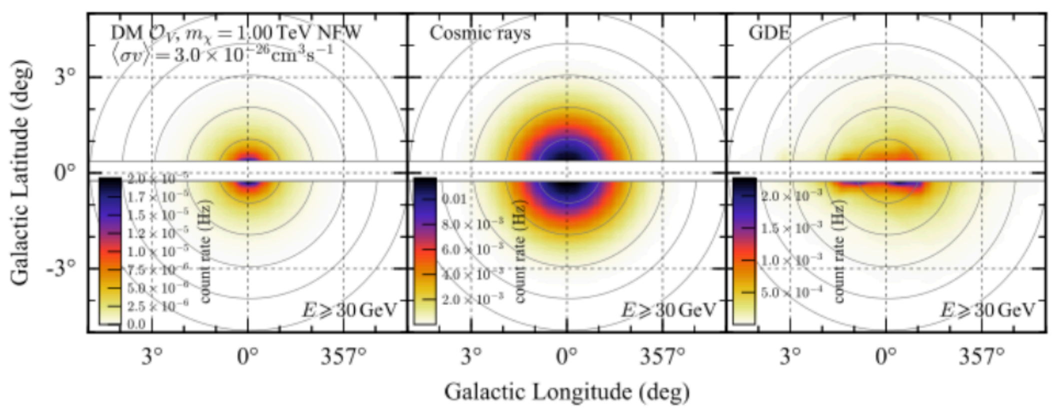
Credits: Luigi Tibaldo

Catalogue of sources extracted from the Galactic Plane Survey KSP (DC-1)



Credits: Josh Cardenzana

Predicted Dark Matter Profile in comparison to cosmic-ray background and Galactic diffuse emission as seen by CTA

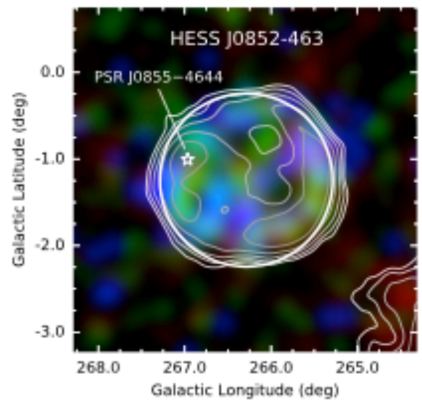


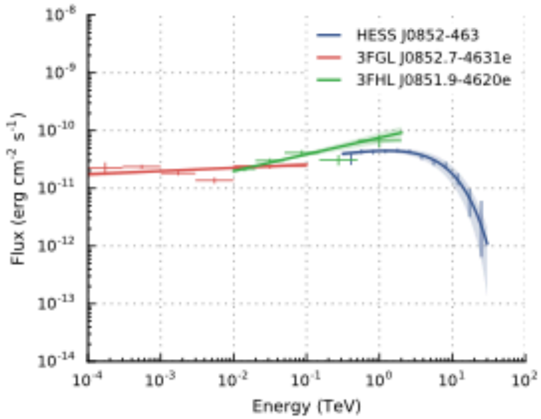
Credits: Balasz et al. 2017, PRD, 96, 3002

Some Gammapy results

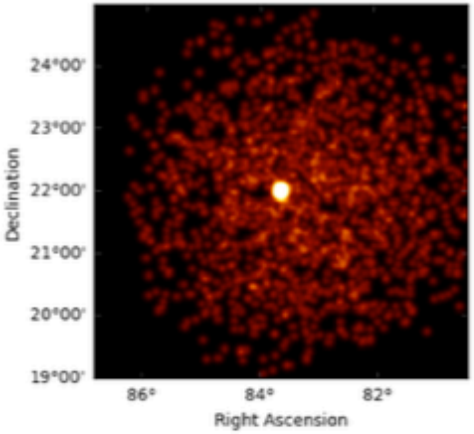
Gammapy is developed and tested using real data from HESS, MAGIC, Fermi-LAT

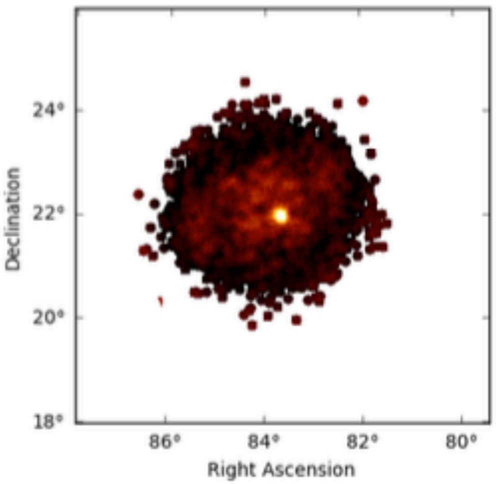
Vela Junior SNR
with Fermi-LAT & HESS

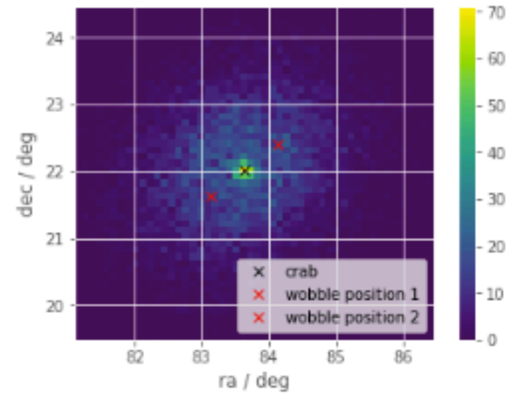




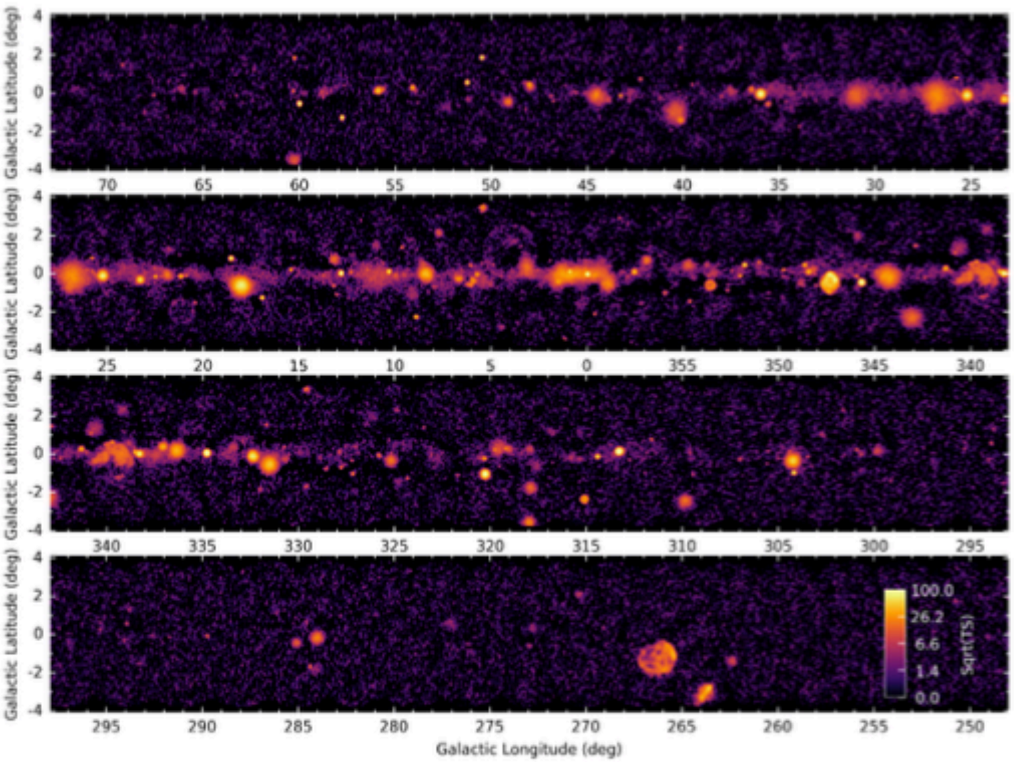
Crab nebula with
existing IACTs

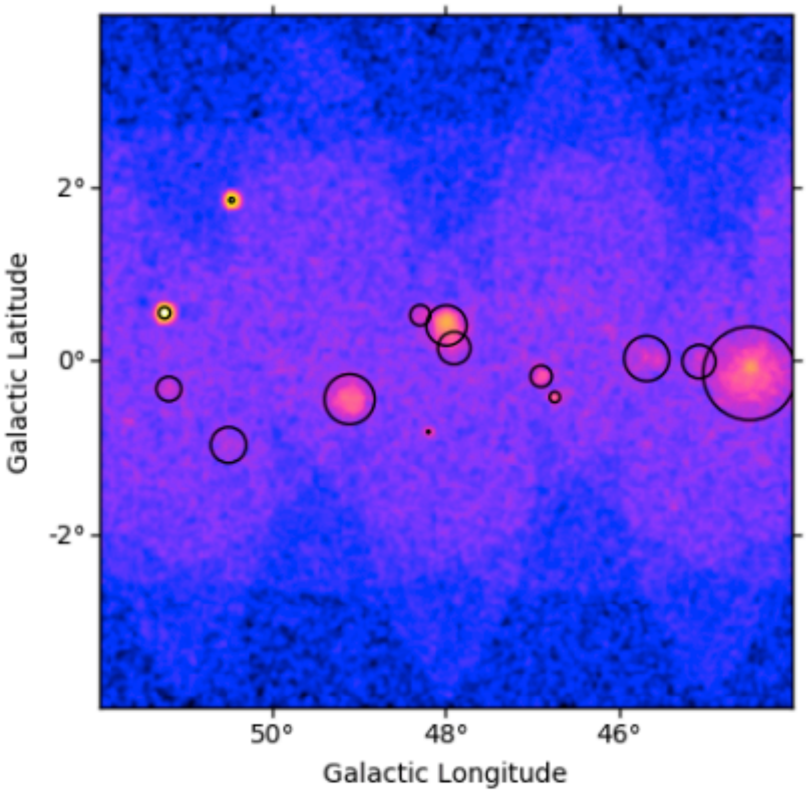






Some Gammapy results

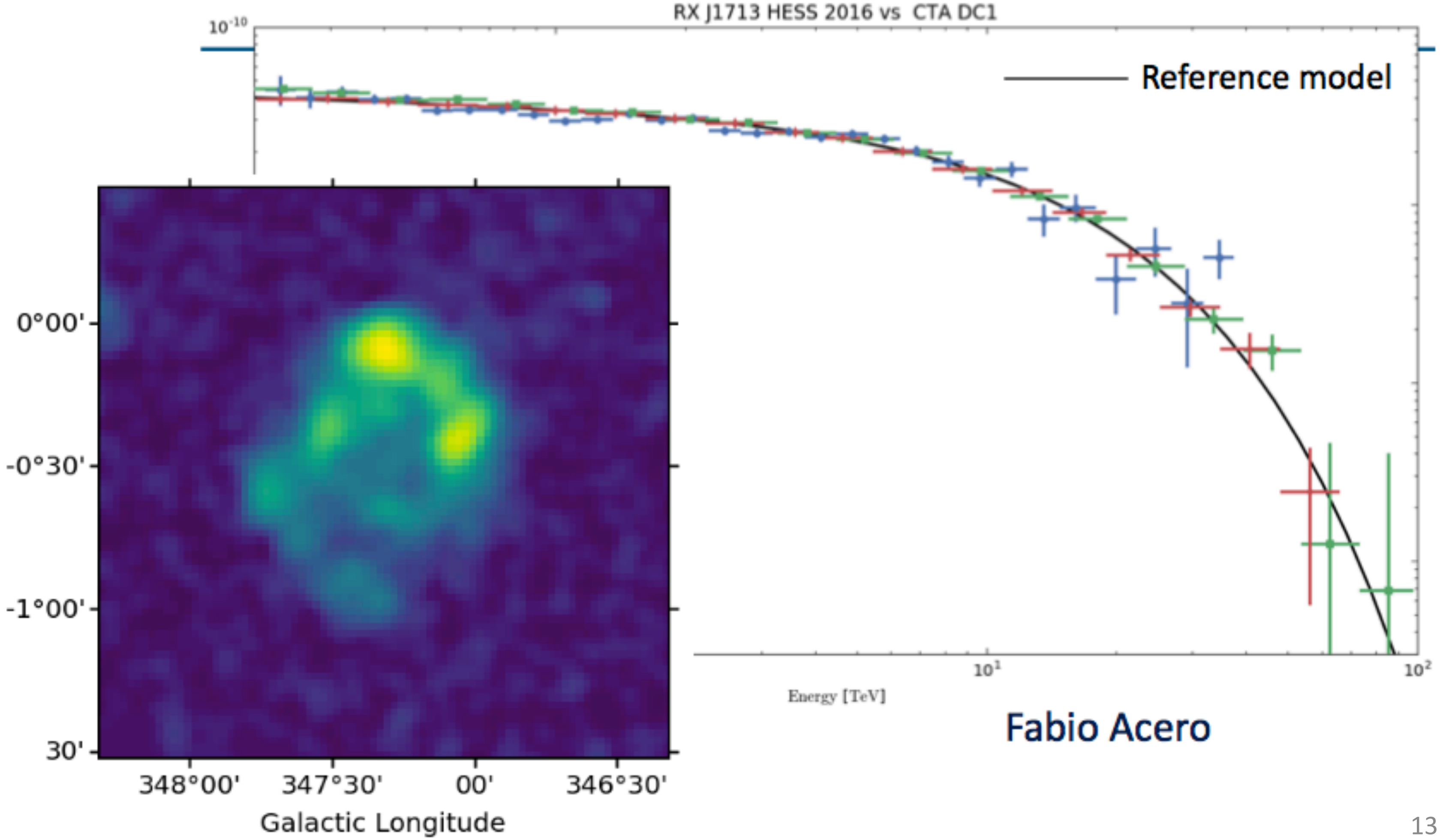




Roberta Zanin, Axel Donath, Christoph Deil & Yves Gallant

Some Gammapy results

CTA DC-1 analysis of RX J1713.7-3946



Fabio Acero

First Data Challenge

- The first data challenge was simulated with CTOOLS
- Data can be analyse with both CTOOLS and Gammapy
- Very useful to test both frameworks and to gives feedback to develops groups.

First Data Challenge

- 11/Dec/2017 Gammapy Hand-Ons at Legnaro
  - We study the PWN MSH 15-52
    - http://tevcat.uchicago.edu/?mode=1;id=95
  - It was done using python with a jupyter notebook
  - Please, check the indico page:
    - https://agenda.infn.it/conferenceDisplay.py?confId=13990
- I will repeat the analysis using CTOOLS.

CTOOLS WORKFLOW
PWN MSH 15-52

$ **csobsselect**

Input event list or observation definition XML file [obs.xml] $CTADATA/obs/obs_gps_baseline.xml
Pointing selection region shape (CIRCLE|BOX) [CIRCLE]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [CEL] GAL
Galactic longitude of selection centre (deg) (0-360) [184.56] 320.33
Galactic latitude of selection centre (deg) (-90-90) [-5.79] -1.19
Radius of selection circle (deg) (0-180) [5.0] 3.0
Output observation definition XML file [outobs.xml] obs_msh_1552.xml

$ **csobsinfo debug=yes**
Input event list, counts cube, or observation definition XML file [obs.xml] obs_msh_1552.xml
Output DS9 region file [ds9.reg]

CTOOLS WORKFLOW
PWN MSH 15-52

$ **ctskymap**

Input event list or observation definition XML file [obs_selected.xml]
First coordinate of image center in degrees (RA or galactic l) (0-360) [0.0] 320.33
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [0.0] -1.19
Projection method (AIT|AZP|CAR|GLS|MER|MOL|SFL|SIN|STG|TAN) [CAR]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [GAL]
Image scale (in degrees/pixel) [0.02]
Size of the X axis in pixels [400]
Size of the Y axis in pixels [400]
Lower energy limit (TeV) [0.1]
Upper energy limit (TeV) [100.0]
Background subtraction method (NONE|IRF|RING) [NONE] IRF
Output skymap file [skymap.fits] skymap_irf.fits

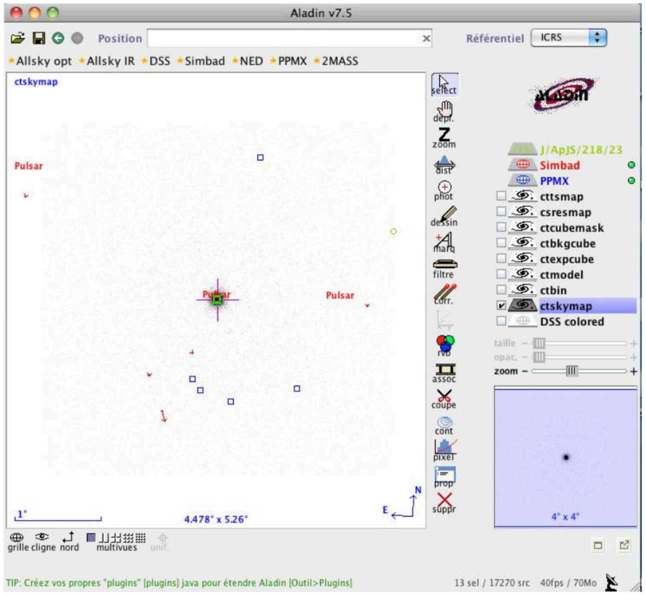
CTOOLS WORKFLOW

PWN MSH 15-52

$ **ctskymap**

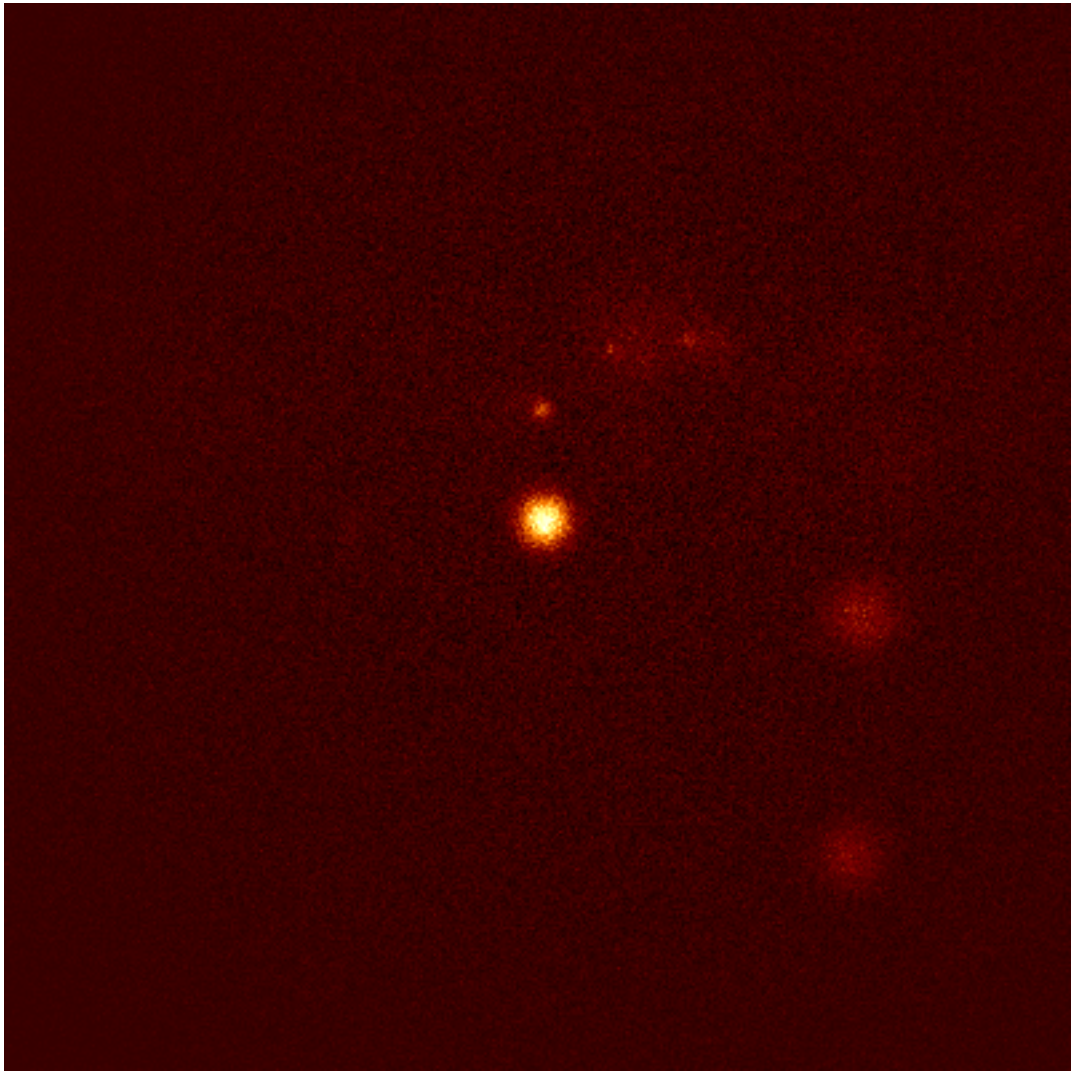
Input event list or observation definition XML file [obs_selected.xml]
First coordinate of image center in degrees (RA or galactic l) (0-360) [0.0] 320.33
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [0.0] -1.19
Projection method (AIT|AZP|CAR|GLS|MER|MOL|SFL|SIN|STG|TAN) [CAR]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [GAL]
Image scale (in degrees/pixel) [0.02]
Size of the X axis in pixels [400]
Size of the Y axis in pixels [400]
Lower energy limit (TeV) [0.1]
Upper energy limit (TeV) [100.0]
Background subtraction method (NONE|IRF|RING) [NONE] IRF
Output skymap file [skymap.fits] skymap_irf.fits

You can use Aladin to publish the image:
ctskymap publish=yes



CTOOLS WORKFLOW

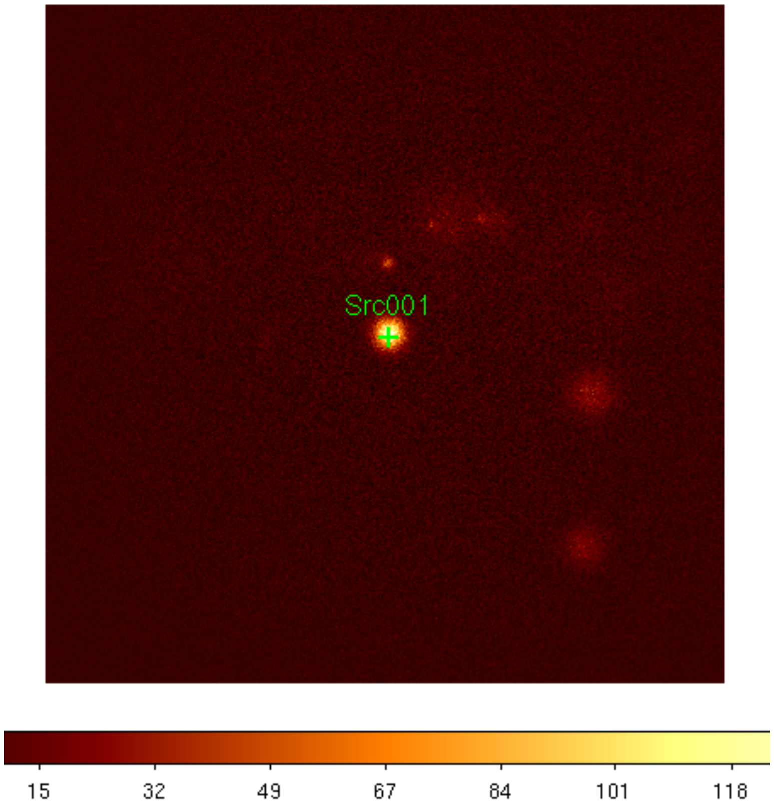
PWN MSH 15-52



CTOOLS WORKFLOW

PWN MSH 15-52

$ **cssrcdetect**
Input sky map file [skymap.fits] skymap_irf.fits
Source model type (POINT) [POINT]
Background model type (NONE|IRF|AEFF|CUBE) [NONE]
Detection threshold (Gaussian sigma) [5.0] 20.0
Output model definition XML file [models.xml]
Output DS9 region file [ds9.reg]



CTOOLS WORKFLOW

PWN MSH 15-52

$ **ctbin**
Input event list or observation definition XML file [obs_selected.xml]
First coordinate of image center in degrees (RA or galactic l) (0-360) [320.33033806]
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [-1.19304588]
Projection method (AIT|AZP|CAR|GLS|MER|MOL|SFL|SIN|STG|TAN) [CAR]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [GAL]
Image scale (in degrees/pixel) [0.02]
Size of the X axis in pixels [300]
Size of the Y axis in pixels [300]
Algorithm for defining energy bins (FILE|LIN|LOG) [LOG]
Start value for first energy bin in TeV [0.1]
Stop value for last energy bin in TeV [100]
Number of energy bins (1-200) [30]
Output counts cube file [cntcube.fits]

CTOOLS WORKFLOW
PWN MSH 15-52

$ **ctexpcube**
Input event list or observation definition XML file [obs_selected.xml]
Input counts cube file to extract exposure cube definition [cntcube.fits]
Output exposure cube file [expcube.fits]

$ **ctpsfcube**
Input event list or observation definition XML file [obs_selected.xml]
Input counts cube file to extract PSF cube definition [NONE]
First coordinate of image center in degrees (RA or galactic l) (0-360) [320.33033806]
Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [-1.19304588]
Projection method (AIT|AZP|CAR|GLS|MER|MOL|SFL|SIN|STG|TAN) [CAR]
Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [GAL]
Image scale (in degrees/pixel) [1.0]
Size of the X axis in pixels [10]
Size of the Y axis in pixels [10]
Lower energy limit (TeV) [0.1]
Upper energy limit (TeV) [100.0]
Number of energy bins [30]
Output PSF cube file [psfcube.fits]

CTOOLS WORKFLOW

PWN MSH 15-52

$ **ctbkgcube**
Input event list or observation definition XML file [obs_selected.xml]
Input counts cube file to extract background cube definition [cntcube.fits]
Input model definition XML file [models.xml]
Output background cube file [NONE] bkgcube.fits
Output model definition XML file [NONE] stacked_models.xml

$ **ctlike**
Input event list, counts cube or observation definition XML file [cntcube.fits]
Input exposure cube file (only needed for stacked analysis) [expcube.fits]
Input PSF cube file (only needed for stacked analysis) [psfcube.fits]
Input background cube file (only needed for stacked analysis) [bkgcube.fits]
Input model definition XML file [stacked_models.xml]
Output model definition XML file [stacked_results.xml]

CTOOLS WORKFLOW

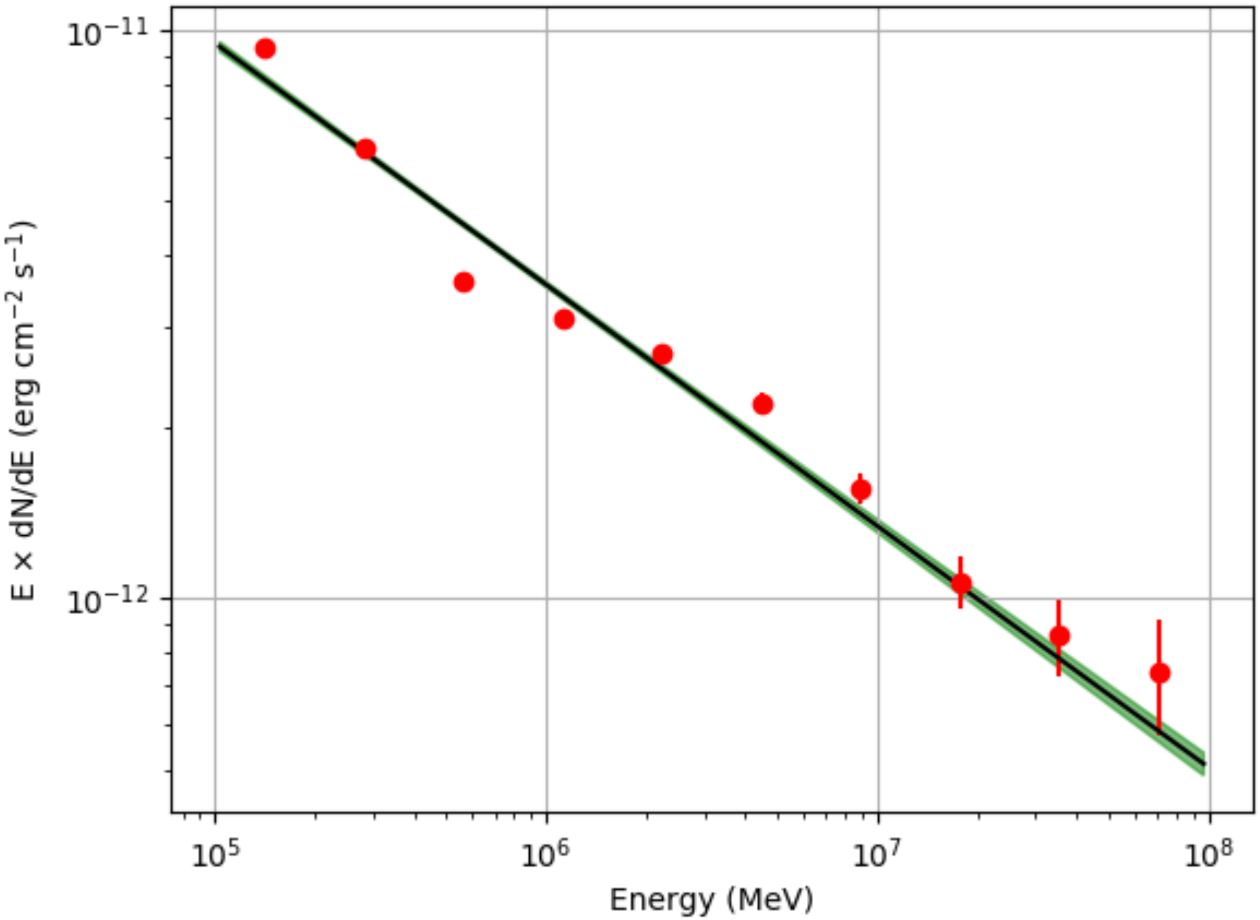
PWN MSH 15-52

$ **ctbutterfly**
Input event list, counts cube or observation definition XML file [cntcube.fits]
Input exposure cube file (only needed for stacked analysis) [NONE] expcube.fits
Input PSF cube file (only needed for stacked analysis) [NONE] psfcube.fits
Input background cube file (only needed for stacked analysis) [NONE] bkgcube.fits
Source of interest [crab] Src001
Input model definition XML file [crab.xml] stacked_results.xml
Start value for first energy bin in TeV [10e-6] 0.1
Stop value for last energy bin in TeV [3000e-6] 100
Output ASCII file [butterfly.txt] butterfly_src001.txt

$ **csspec**
Input event list, counts cube, or observation definition XML file [cntcube.fits]
Input exposure cube file (only needed for stacked analysis) [expcube.fits]
Input PSF cube file (only needed for stacked analysis) [psfcube.fits]
Input background cube file (only needed for stacked analysis) [bkgcube.fits]
Input model definition XML file [stacked_results.xml]
Source name [Src001]
Binning algorithm (LIN|LOG|FILE) [LOG]
Lower energy limit (TeV) [0.1]
Upper energy limit (TeV) [100.0]
Number of energy bins (0=unbinned) [31] 10
Output spectrum file [spectrum.fits] spectrum_src001.fits

CTOOLS WORKFLOW

PWN MSH 15-52



CTOOLS WORKFLOW

PWN MSH 15-52

$ **csresmap**
Input event list, counts cube, or observation definition XML file [cntcube.fits]
Input model cube file (generated with ctmodel) [NONE]
Input exposure cube file (only needed for stacked analysis) [expcube.fits]
Input PSF cube file (only needed for stacked analysis) [psfcube.fits]
Input background cube file (only needed for stacked analysis) [bkgcube.fits]
Input model definition XML file [stacked_results.xml]
Residual map computation algorithm (SUB|SUBDIV|SUBDIVSQRT|SIGNIFICANCE) [SIGNIFICANCE]
Output residual map file [resmap.fits]

CTOOLS WORKFLOW

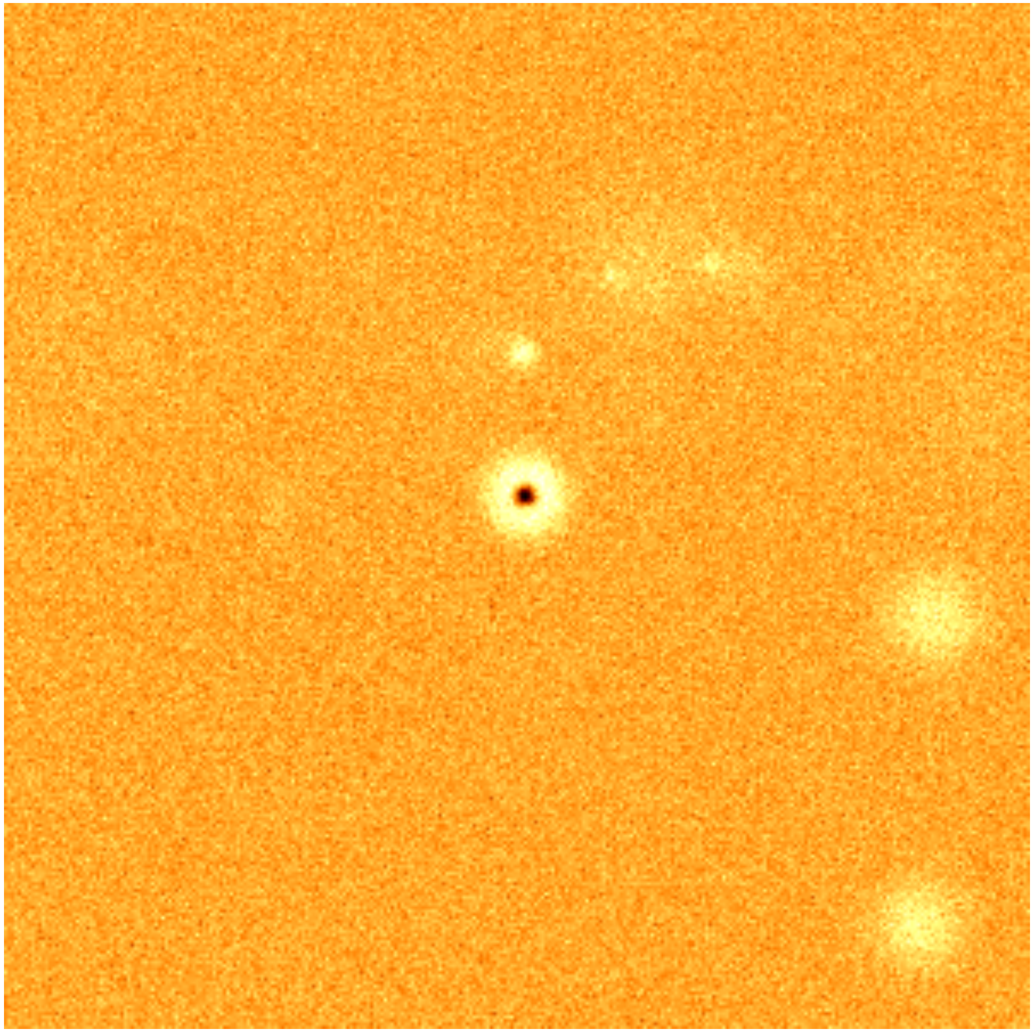
PWN MSH 15-52

It is not a point source

We have to used a extended source

In this case it is a radial gaussian source

We change the xml file! And repeat…



CTOOLS WORKFLOW

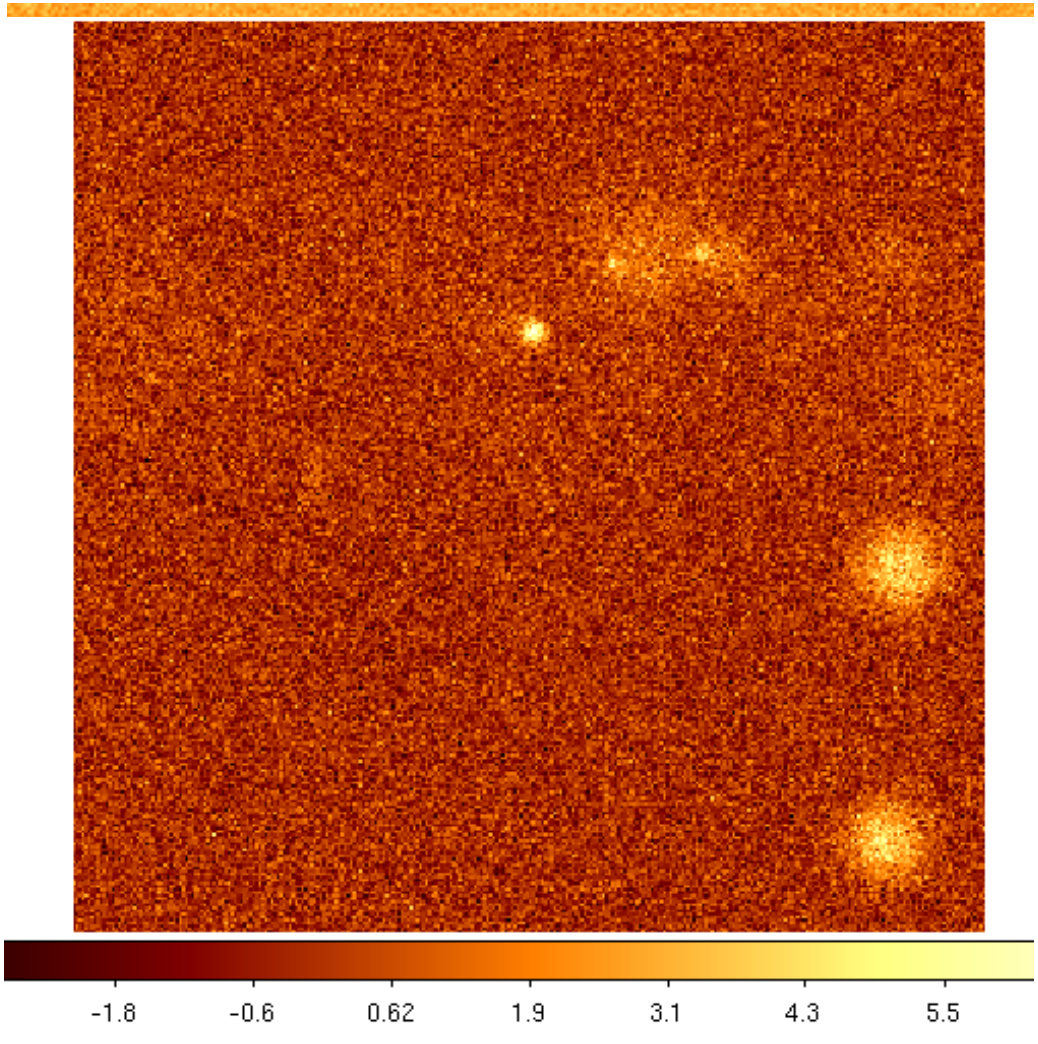
PWN MSH 15-52

$ **ctlike**
Input event list, counts cube or observation definition XML file [cntcube.fits]
Input exposure cube file (only needed for stacked analysis) [expcube.fits]
Input PSF cube file (only needed for stacked analysis) [psfcube.fits]
Input background cube file (only needed for stacked analysis) [bkgcube.fits]
Input model definition XML file [stacked_models.xml] stacked_gauss_models.xml
Output model definition XML file [stacked_results.xml] stacked_gaus_results.xml

$ **csresmap**
Input event list, counts cube, or observation definition XML file [cntcube.fits]
Input model cube file (generated with ctmodel) [NONE]
Input exposure cube file (only needed for stacked analysis) [expcube.fits]
Input PSF cube file (only needed for stacked analysis) [psfcube.fits]
Input background cube file (only needed for stacked analysis) [bkgcube.fits]
Input model definition XML file [stacked_gaus_results.xml]
Residual map computation algorithm (SUB|SUBDIV|SUBDIVSQRT|SIGNIFICANCE) [SIGNIFICANCE]
Output residual map file [resmap_gauss.fits]

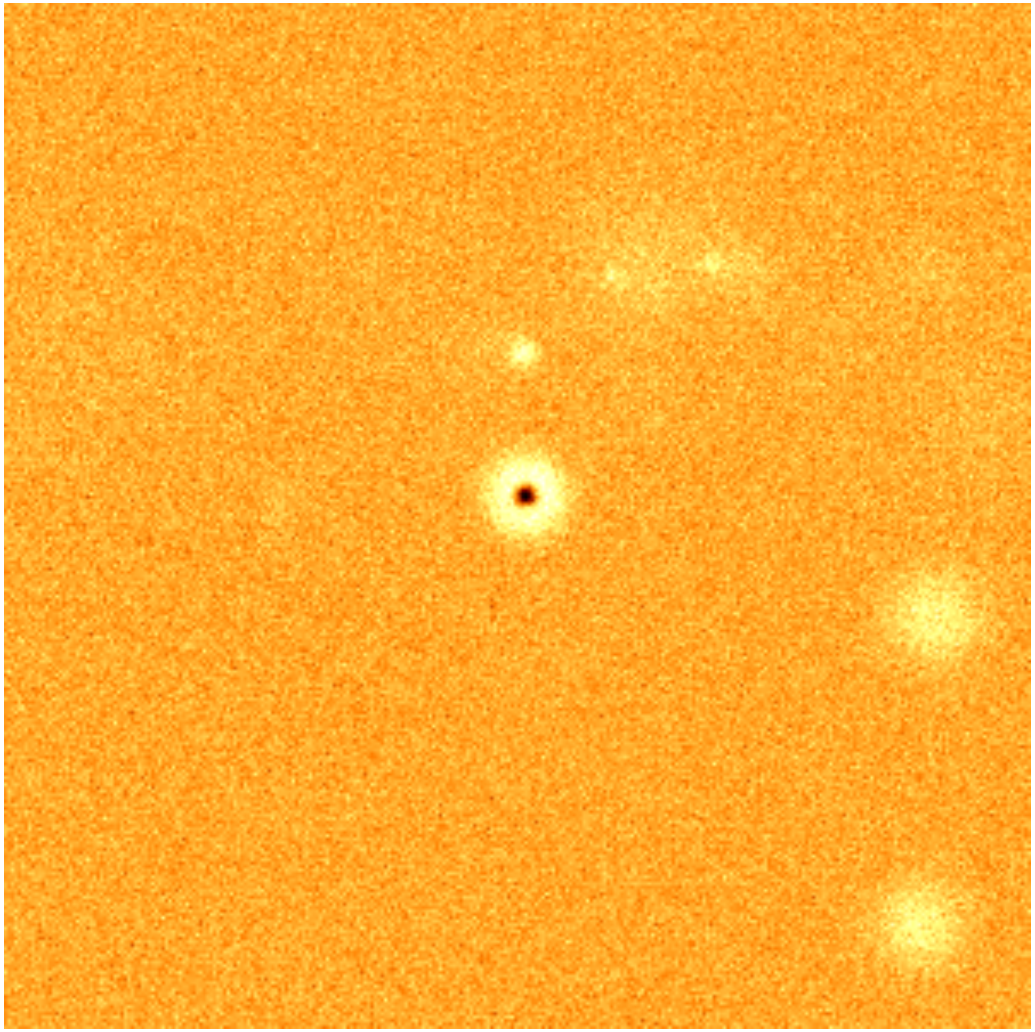
CTOOLS WORKFLOW

PWN MSH 15-52



CTOOLS WORKFLOW

PWN MSH 15-52



SKYMAP WITH GAMMAP

PWN MSH 15-52

Open a jupyter notebook or your favorite editor

- Load the python libraries
- Load the DC-1 data index

```
from gammapy.data import DataStore
import astropy.units as u
from astropy.coordinates import SkyCoord

data_store = DataStore.from_dir('$CTADATA/index/gps')
glat_target = -1.19304588
glon_target = 320.33033806
max_offset = 3 * u.deg
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Load the observations within a 3 deg circle from the target

```
glat_target = -1.19304588
glon_target = 320.33033806
max_offset = 3 * u.deg

table = data_store.obs_table

pos_obs = SkyCoord(table['GLON_PNT'], table['GLAT_PNT'],
frame='galactic', unit='deg')
pos_target = SkyCoord(glon_target, glat_target, frame='galactic',
unit='deg')
offset = pos_target.separation(pos_obs).deg
mask = (offset < max_offset.value)
table = table[mask]
print('Number of selected observations: ', len(table))
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Load the observations within a 3 deg circle from the target

```
glat_target = -1.19304588
glon_target = 320.33033806
max_offset = 3 * u.deg

table = data_store.obs_table

pos_obs = SkyCoord(table['GLON_PNT'], table['GLAT_PNT'], frame='galactic', unit='deg')
pos_target = SkyCoord(glon_target, glat_target, frame='galactic’,unit='deg')
offset = pos_target.separation(pos_obs).deg
mask = (offset < max_offset.value)
table = table[mask]

obs_list = data_store.obs_list(obs_ids_now)
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- Define on ON region

```
from regions import CircleSkyRegion

on_radius = 0.3 * u.deg
on_region = CircleSkyRegion(center=pos_target, radius=on_radius)
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- Define the reference Image (Canvas)

```
from gammapy.image import SkyImage
xref = pos_target.galactic.l.value
yref = pos_target.galactic.b.value
size = 10 * u.deg
binsz = 0.02 # degree per pixel
npix = int((size / binsz).value)
print(npix)
ref_image = SkyImage.empty(
    nxpix=500, nypix=500, binsz=binsz,
    xref=xref, yref=yref,
    proj='CAR', coordsys='GAL',
)
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- Estimate background using a ring surrounded the target

```
exclusion_mask = ref_image.region_mask(on_region)
exclusion_mask.data = 1 - exclusion_mask.data
exclusion_mask.plot()

from gammapy.background import RingBackgroundEstimator
bkg_estimator = RingBackgroundEstimator(
    r_in=0.6 * u.deg,
    width=0.2 * u.deg,
)
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- Estimate background using a ring surrounded the target

```
from gammapy.image import IACTBasicImageEstimator
image_estimator = IACTBasicImageEstimator(
    reference=ref_image,
    emin=200 * u.GeV,
    emax=100 * u.TeV,
    offset_max=3 * u.deg,
    background_estimator=bkg_estimator,
    exclusion_mask=exclusion_mask,
)
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- We are ready to make the skymap

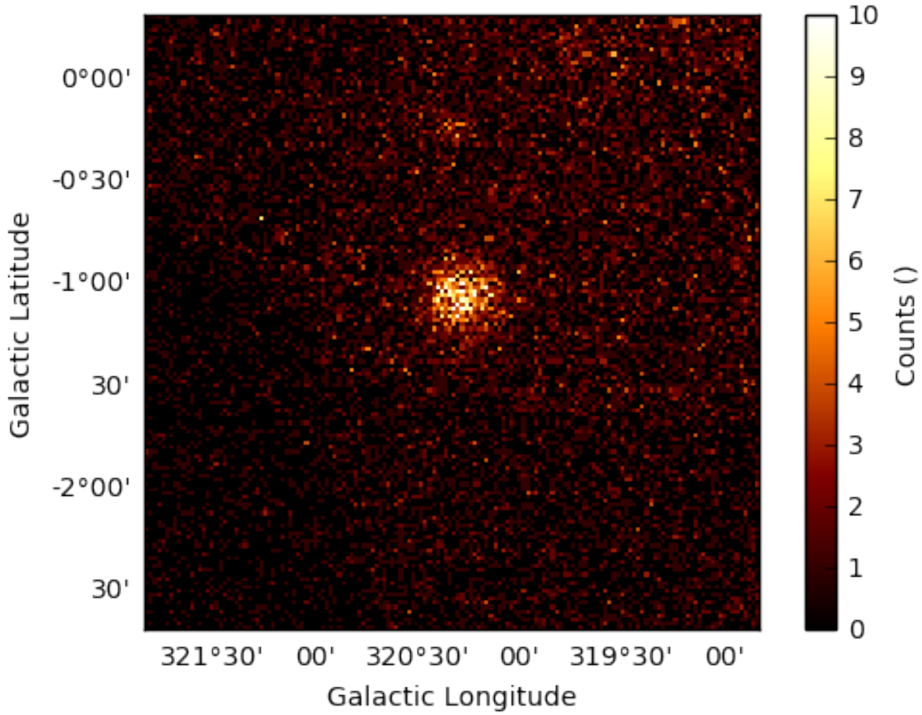
```
images = image_estimator.run(obs_list)

cts_image = images['counts']
cts_image.show(vmin=0,vmax=10,add_cbar=True)

bkg_image = images['background']
bkg_image.show(vmin=0,vmax=4,add_cbar=True)
```

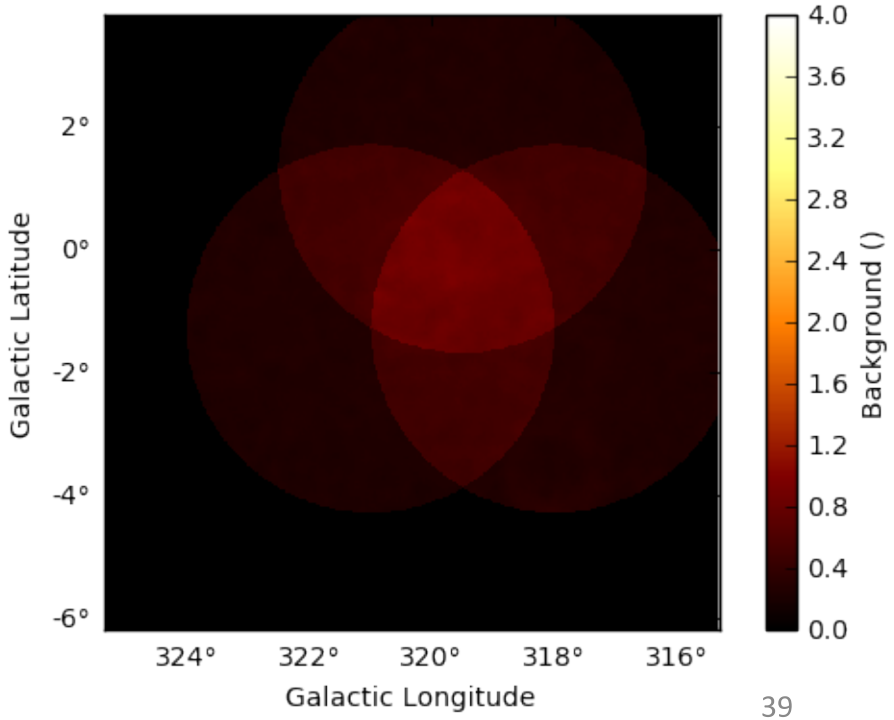
SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- We are ready to make the skymap





SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- Lets significance map by using the Li&Ma significance

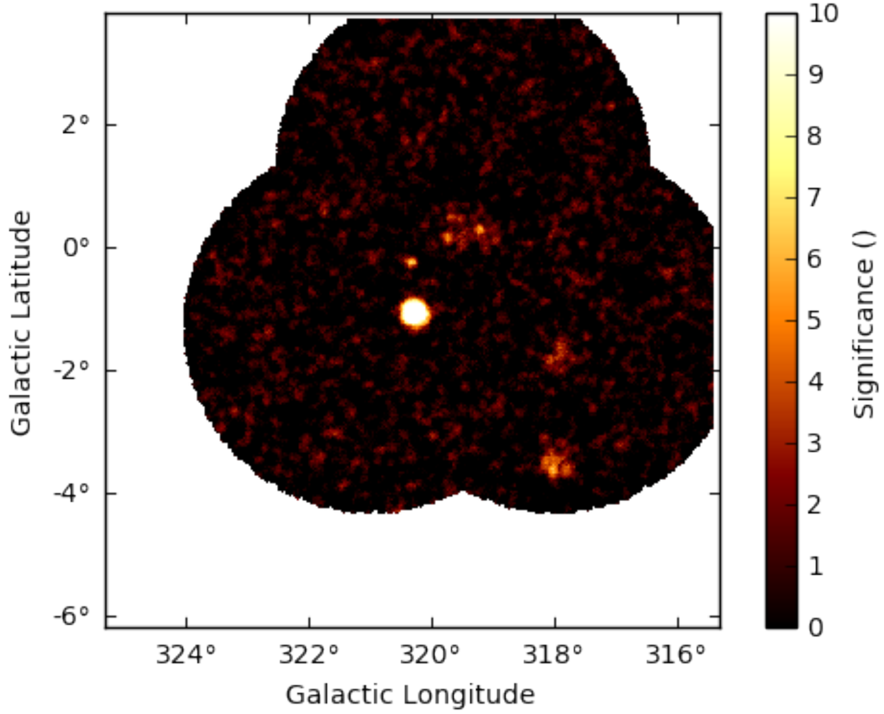
```
from astropy.convolution import Tophat2DKernel
from gammapy.detect import compute_lima_image

kernel = Tophat2DKernel(4)

lima_image = compute_lima_image(cts_image,bkg_image,kernel)['significance']

lima_image.show(vmin=0,vmax=10,add_cbar=True)
```

SKYMAP WITH GAMMAP

PWN MSH 15-52

- Make the SKYMAP

- Lets significance map by using the Li&Ma significance



Summary & Conclusions

GammaLib+CTOOLS and Gammapy are a high-level data analysis package for gamma-ray astronomy

Both are work in progress

They are very well documented

Gammapy is a python library

CTOOLS is a set of user-friendly command-lines tools that also support python

They are being used in the first Data Challenge

This first Data Challenge is used to comparer both frameworks

And also to improve their analysis algorithms, debugging, ecc.