Max-Planck-Institut für Kernphysik

Data Analysis for Gamma-ray astronomy

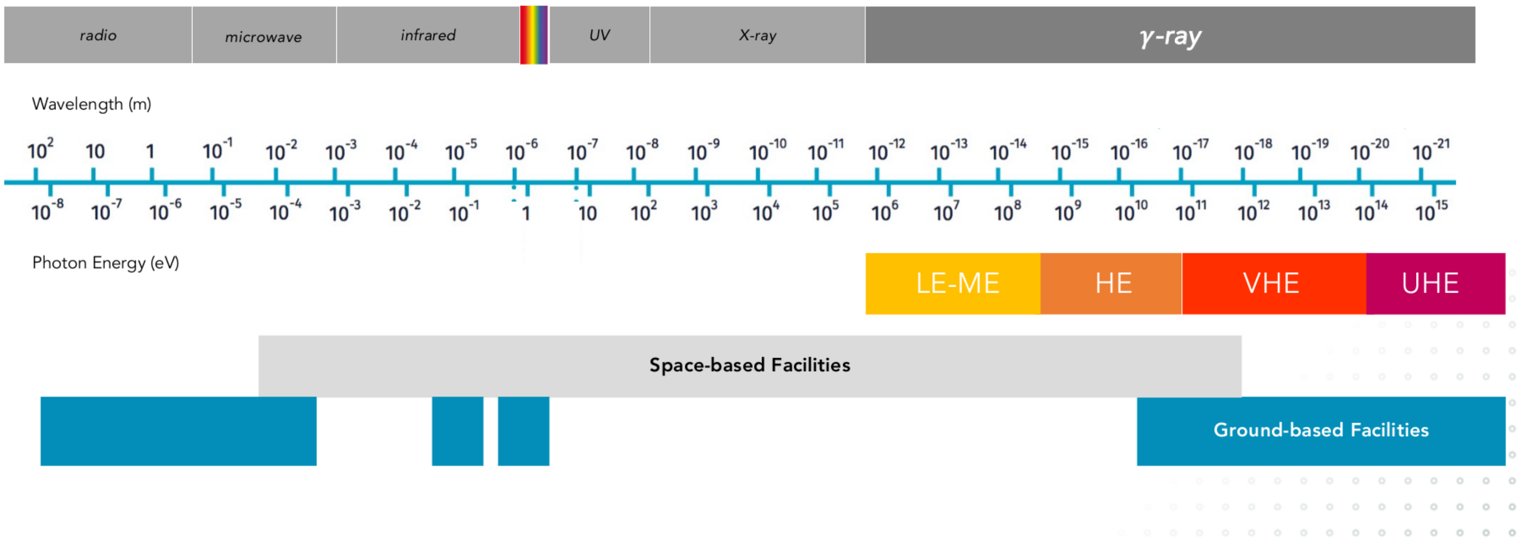
Gammapy Workshop

December 2024

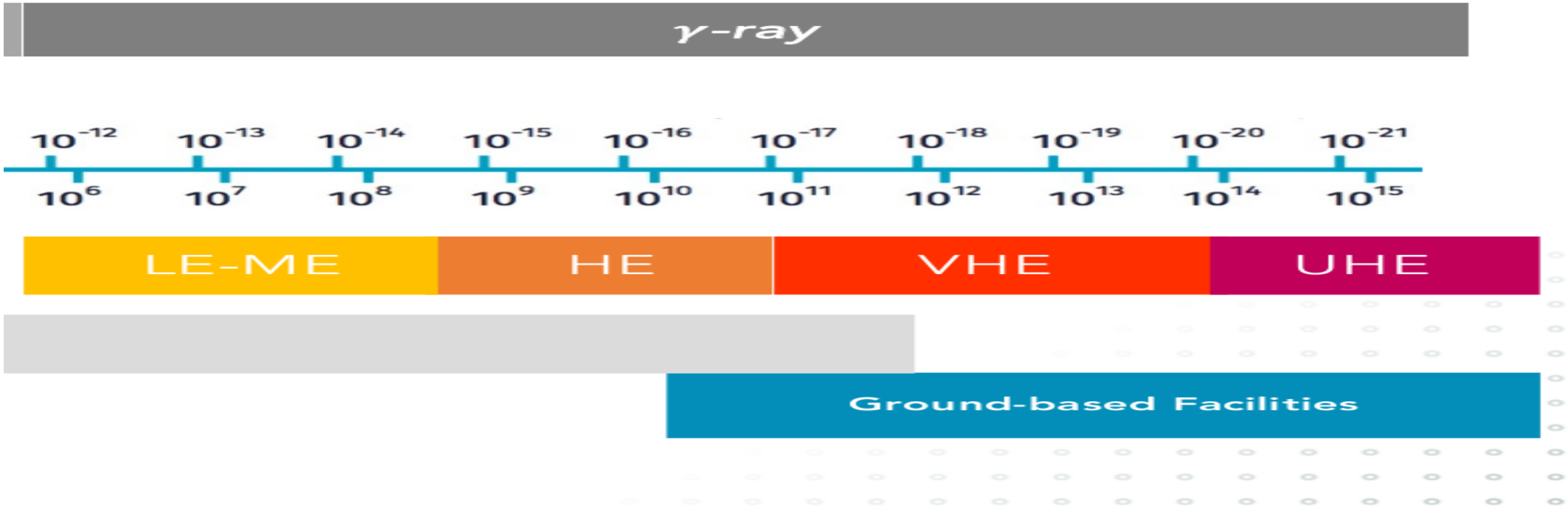
Laura Olivera-Nieto

A **Python** package for **gamma-ray** astronomy

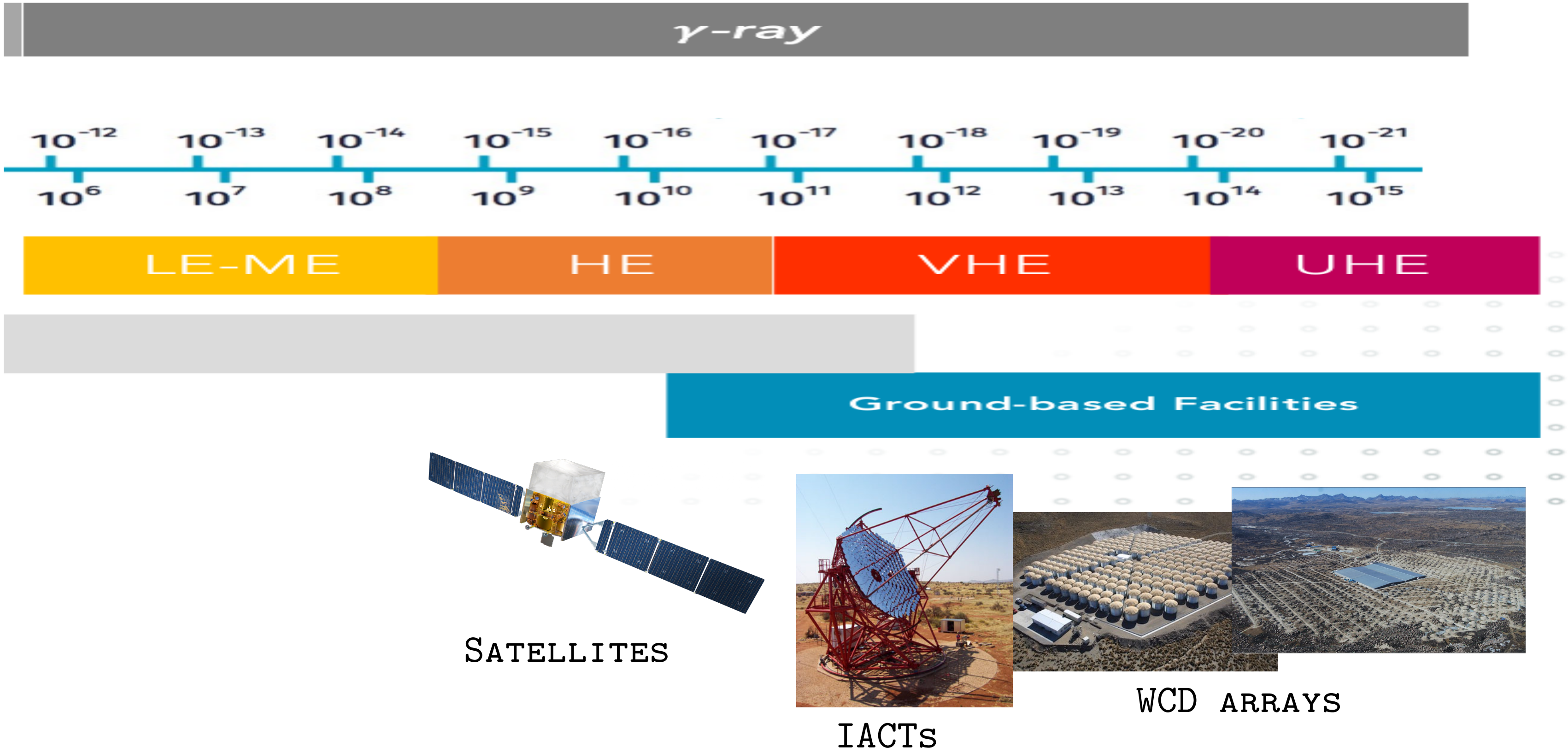
Gamma-ray astronomy and this talk



Gamma-ray astronomy and this talk



Gamma-ray astronomy and this talk



Gamma-ray astronomy and this talk

- Very hard to cover all topics/existing software in ~2h
- 3 different types of instrument, subtleties and technicalities specific to each (as you have seen already!)
- In this talk I will instead try to focus on the **THINGS IN COMMON**
- For that I will rely heavily on Gammapy — A **Python** package for **gamma-ray** astronomy
- Not the only existing package of course but the only one dedicated to gamma-ray as a whole

Data Analysis

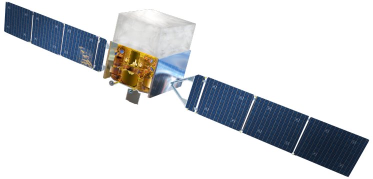
DATA ANALYSIS

DATA ANALYSIS

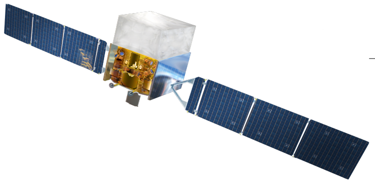
- WHAT DOES THE DATA OF GAMMA-RAY INSTRUMENTS LOOK LIKE?
- WHAT DO I NEED TO GO FROM DATA TO PHYSICAL QUANTITIES?
- HOW TO DETERMINE THE VALIDITY OF DATA?
- WHAT IS "DATA REDUCTION"?

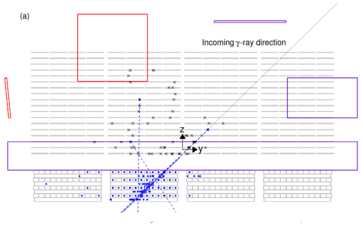
DATA ANALYSIS

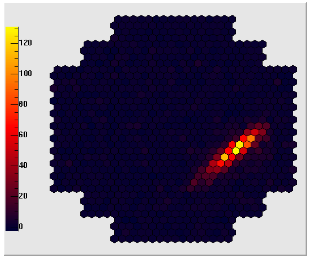
- **WHAT DOES THE DATA OF GAMMA-RAY INSTRUMENTS LOOK LIKE?**
- WHAT DO I NEED TO GO FROM DATA TO PHYSICAL QUANTITIES?
- HOW TO DETERMINE THE VALIDITY OF DATA?
- WHAT IS "DATA REDUCTION"?

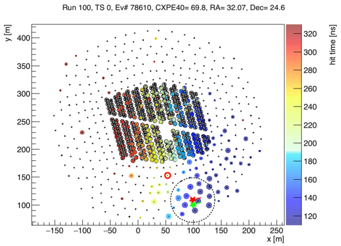


Data-taking

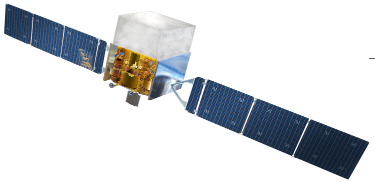


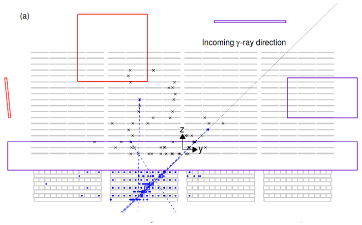

(a)
Incoming γ-ray direction

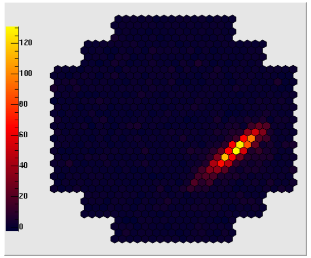

120
100
80
60
40
20
0

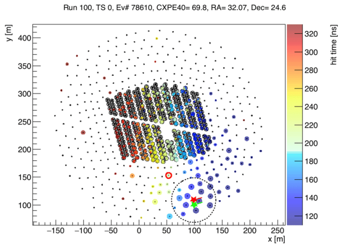

Run 100, TS 0, Ev# 78610, CXPE40= 69.8, RA= 32.07, Dec= 24.6
y [m]
x [m]
hit time [ns]

Data-taking | **Reconstruction**









EVENT ID

ENERGY

DIRECTION

TIME

"GAMMANESS"

Real Data from a gamma-ray instrument

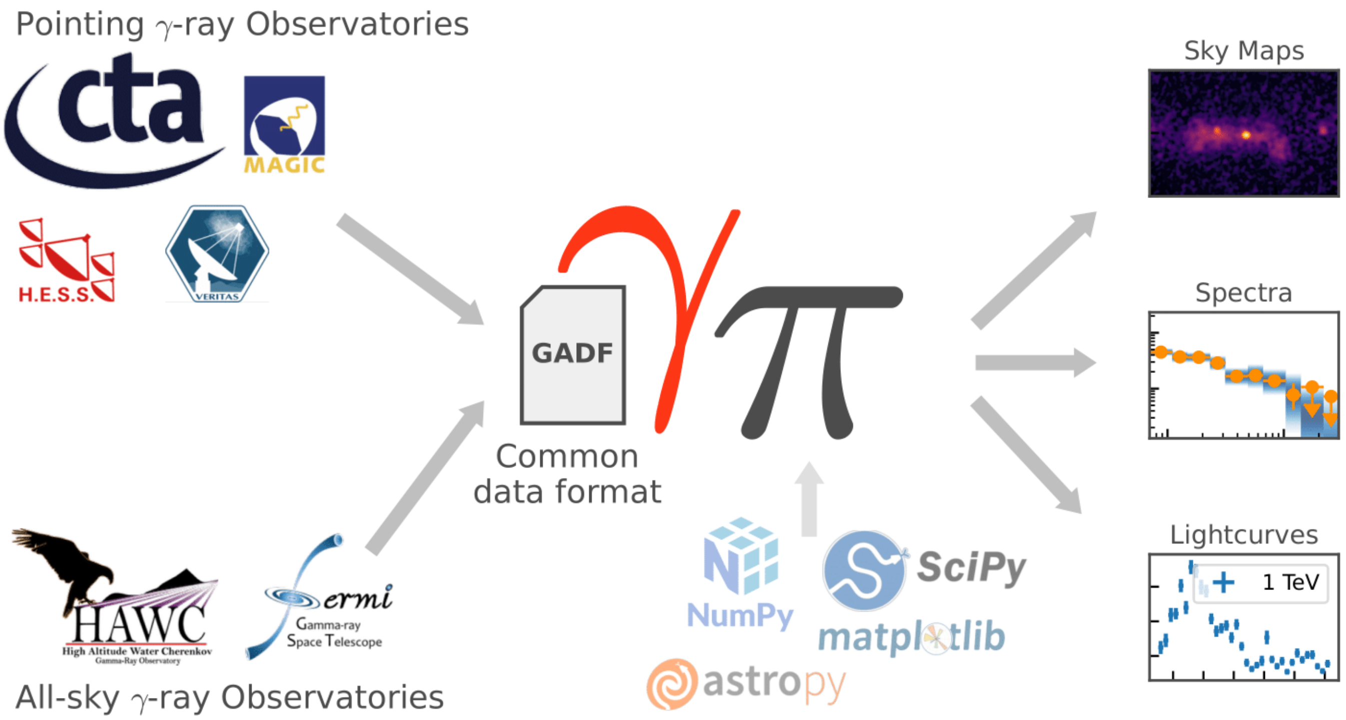
EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
18184891531583	139533040.89130569	284.92114	2.3147857	8.494793
18210661335308	139533064.76603198	284.68036	2.1020594	1.2389338
18214956302833	139533070.2015779	284.57544	2.641703	3.212592
18223546236968	139533075.5970521	285.67664	1.9790567	0.50208473
18223546237257	139533077.386261	286.9335	1.7020291	19.793955
18227841204705	139533082.1508355	285.97955	1.0685997	1.273117
18245021073483	139533096.00451803	284.84094	3.5419352	25.999714
18257905975463	139533108.47312307	283.92752	2.3060772	0.81107205
18262200943041	139533114.4788592	285.01868	4.477947	0.6513306
18270790877414	139533121.14376855	284.47827	4.6750755	0.5971853
18296560681602	139533147.5147109	284.61392	3.4823265	0.76477724
18300855648487	139533149.4298861	286.3041	2.4164367	0.96899956
18313740550377	139533161.2780261	284.2054	3.1820278	0.5589061
18322330485363	139533171.584491	284.23578	1.9863799	0.83051723
18335215387058	139533182.66867948	288.14926	1.5052232	1.3813052
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Can you tell me which one?

A unified format and tool for gamma-ray astronomy

- In the past decade or so, there has been a big effort to define a standard format to use when storing gamma-ray astronomy data
- By format I mean which columns/quantity names and so on
- Learned from existing standards, such as x-ray data and Fermi
- Result: Gamma-astro-data-format
- If all data looks the same, we can all share a tool!
- Result: Gammapy

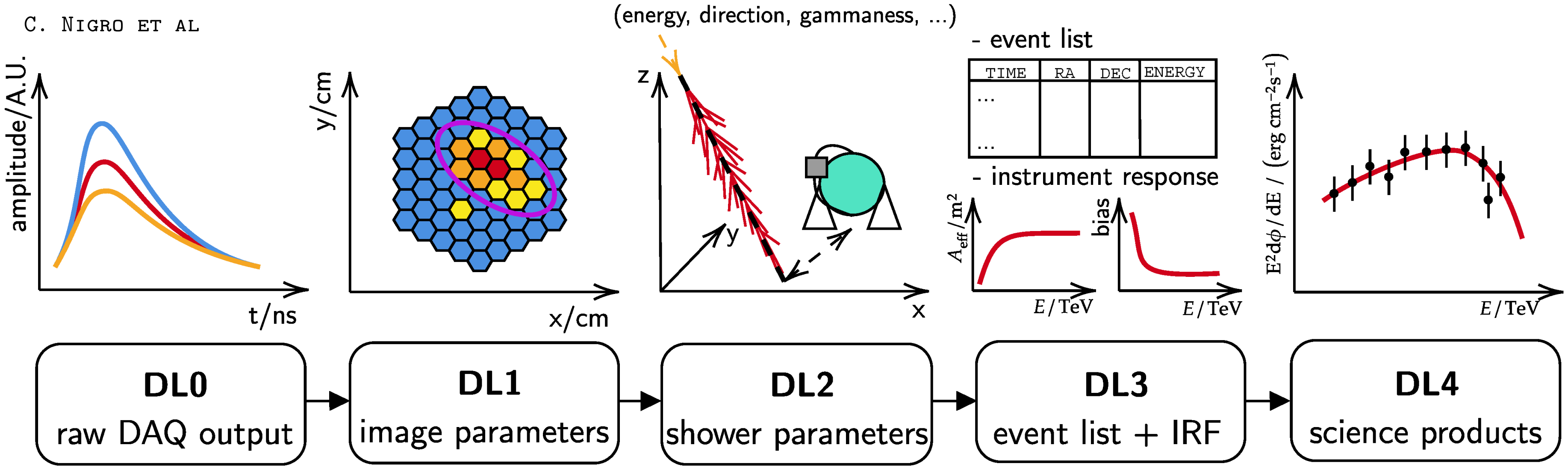
A unified format and tool for gamma-ray astronomy



A. Donath et al

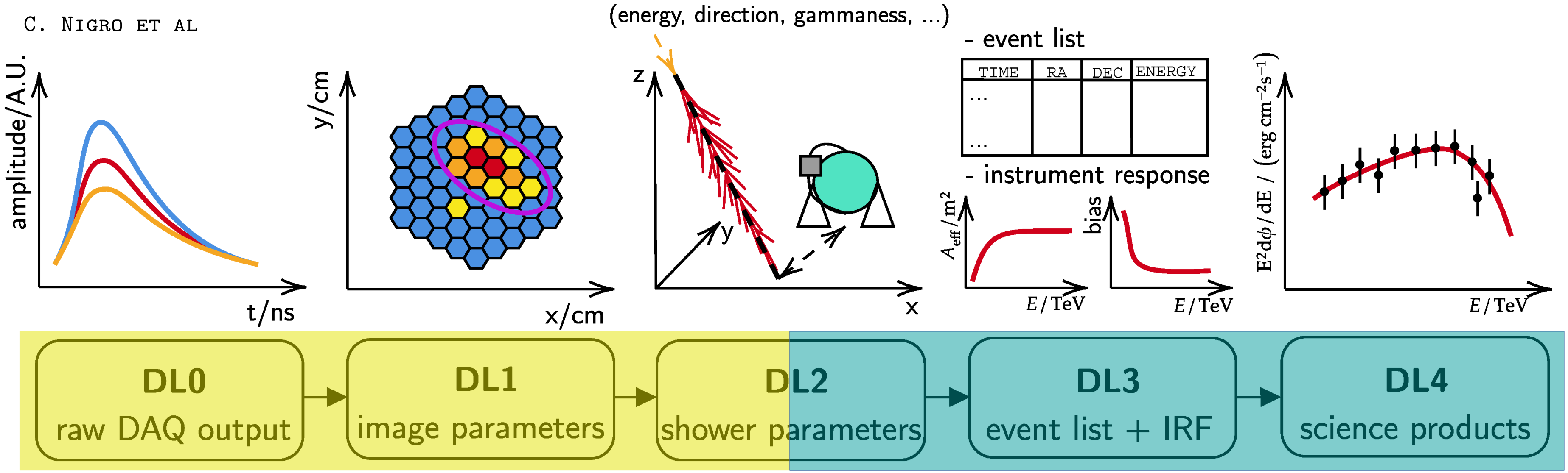
A unified format and tool for gamma-ray astronomy

C. Nigro et al



A unified format and tool for gamma-ray astronomy

C. Nigro et al



HIGHLY INSTRUMENT SPECIFIC **INSTRUMENT AGNOSTIC**

Event List

List of gamma-like events

EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
18184891531583	139533040.89130569	284.92114	2.3147857	8.494793
18210661335308	139533064.76603198	284.68036	2.1020594	1.2389338
18214956302833	139533070.2015779	284.57544	2.641703	3.212592
18223546236968	139533075.5970521	285.67664	1.9790567	0.50208473
18223546237257	139533077.386261	286.9335	1.7020291	19.793955
18227841204705	139533082.1508355	285.97955	1.0685997	1.273117
18245021073483	139533096.00451803	284.84094	3.5419352	25.999714
18257905975463	139533108.47312307	283.92752	2.3060772	0.81107205
18262200943041	139533114.4788592	285.01868	4.477947	0.6513306
18270790877414	139533121.14376855	284.47827	4.6750755	0.5971853
18296560681602	139533147.5147109	284.61392	3.4823265	0.76477724
18300855648487	139533149.4298861	286.3041	2.4164367	0.96899956
18313740550377	139533161.2780261	284.2054	3.1820278	0.5589061
18322330485363	139533171.584491	284.23578	1.9863799	0.83051723
18335215387058	139533182.66867948	288.14926	1.5052232	1.3813052

Event List

List of gamma-like events

EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
18184891531583	139533040.89130569	284.92114	2.3147857	8.494793
18210661335308	139533064.76603198	284.68036	2.1020594	1.2389338
18214956302833	139533070.2015779	284.57544	2.641703	3.212592
18223546236968	139533075.5970521	285.67664	1.9790567	0.50208473
18223546237257	139533077.386261	286.9335	1.7020291	19.793955
18227841204705	139533082.1508355	285.97955	1.0685997	1.273117
18245021073483	139533096.00451803	284.84094	3.5419352	25.999714
18257905975463	139533108.47312307	283.92752	2.3060772	0.81107205
18262200943041	139533114.4788592	285.01868	4.477947	0.6513306
18270790877414	139533121.14376855	284.47827	4.6750755	0.5971853
18296560681602	139533147.5147109	284.61392	3.4823265	0.76477724
18300855648487	139533149.4298861	286.3041	2.4164367	0.96899956
18313740550377	139533161.2780261	284.2054	3.1820278	0.5589061
18322330485363	139533171.584491	284.23578	1.9863799	0.83051723
18335215387058	139533182.66867948	288.14926	1.5052232	1.3813052
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Good Time Intervals (GTI)

Time ranges when the instrument was taking data

START	STOP
Time	Time
53524.96611324074	53524.985685

Event List

List of gamma-like events

EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
18184891531583	139533040.89130569	284.92114	2.3147857	8.494793
18210661335308	139533064.76603198	284.68036	2.1020594	1.2389338
18214956302833	139533070.2015779	284.57544	2.641703	3.212592
18223546236968	139533075.5970521	285.67664	1.9790567	0.50208473
18223546237257	139533077.386261	286.9335	1.7020291	19.793955
18227841204705	139533082.1508355	285.97955	1.0685997	1.273117
18245021073483	139533096.00451803	284.84094	3.5419352	25.999714
18257905975463	139533108.47312307	283.92752	2.3060772	0.81107205
18262200943041	139533114.4788592	285.01868	4.477947	0.6513306
18270790877414	139533121.14376855	284.47827	4.6750755	0.5971853
18296560681602	139533147.5147109	284.61392	3.4823265	0.76477724
18300855648487	139533149.4298861	286.3041	2.4164367	0.96899956
18313740550377	139533161.2780261	284.2054	3.1820278	0.5589061
18322330485363	139533171.584491	284.23578	1.9863799	0.83051723
18335215387058	139533182.66867948	288.14926	1.5052232	1.3813052

Good Time Intervals (GTI)

Time ranges when the instrument was taking data

START	STOP
Time	Time
53524.96611324074	53524.985685

These lists are prepared by the people from each instrument and represent "science-ready" data or "Data level 3"

Data Analysis

- What does the data of gamma-ray instruments look like?
- **What do I need to go from data to physical quantities?**
- How to determine the validity of data?
- What is "data reduction"?

From "events" to physics

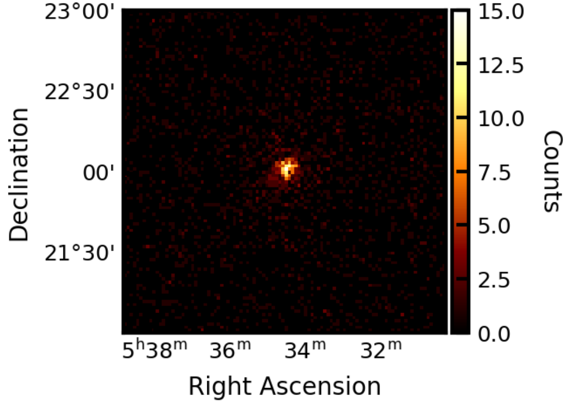
Let's say you have a list of events taken by a gamma-ray instrument when observing a source, and you want to study that source.

From "events" to physics

THIS IS DATA REDUCTION

Let's say you have a list of events taken by a gamma-ray instrument when observing a source, and you want to study that source.

You can make a map! Which is just a 2D histogram of the sky coordinates

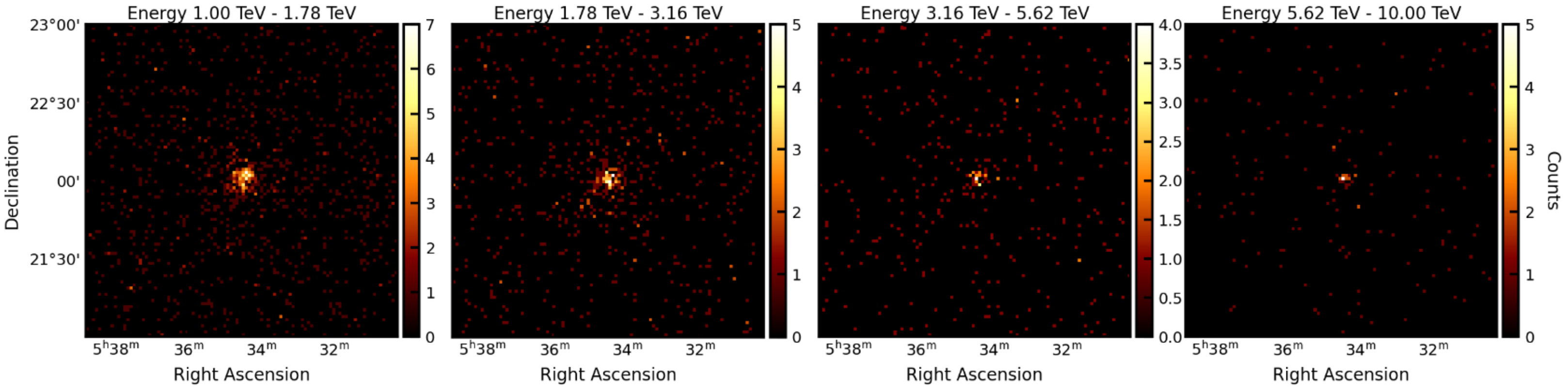


`dataset.counts.reduce_over_axes().plot()`

From "events" to physics

Let's say you have a list of events taken by a gamma-ray instrument when observing a source, and you want to study that source.

You can make a map! Which is just a 3D histogram of the sky coordinates and energy*



`dataset.counts.plot_grid()`

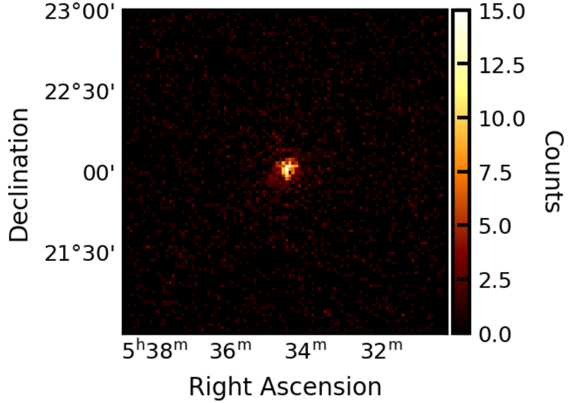
* or time, or some other quantity...

From "events" to physics

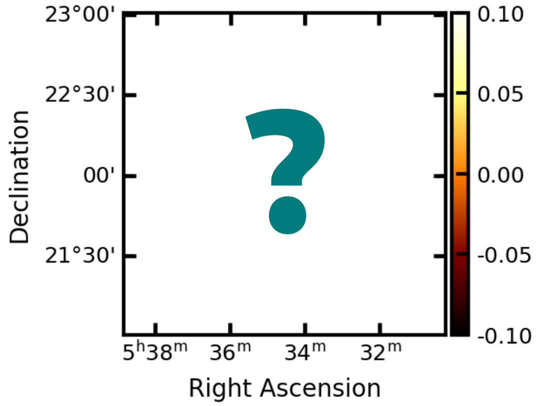
Let's say you have a list of events taken by a gamma-ray instrument when observing a source, and you want to study that source.

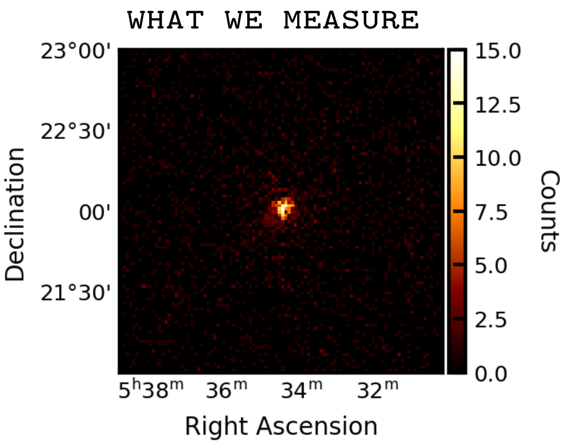
You can make a map! Which is just a 3D histogram of the sky coordinates and energy

What we measure

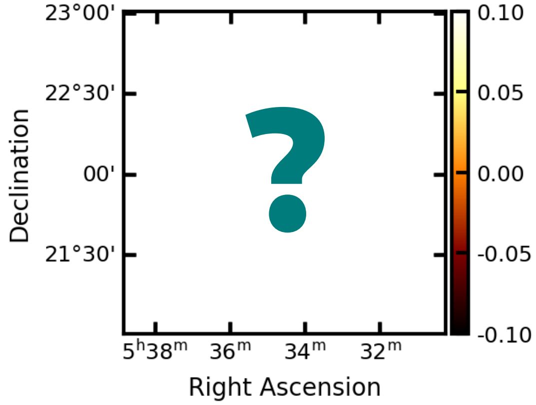


What is the "truth"?





What is the "truth"? →



Measuring is not perfect! Instruments introduce biases, inaccuracies,...

- We measure **X counts** from the source → how **bright** is it?
- The counts "blob" has a spatial extent → what is the **actual size** of the source?
- We only see a source in the middle → but did we **observe** other parts of the map as much?
- The measured counts have an energy distribution → what is the **spectrum** of the source?

Events are not enough! We also need the **instrument response**

Instrument response functions

GAMMA RAY SOURCE $\frac{dN_\gamma}{dE_{true}\,dx_{true}}$ ⟷ "GAMMA-LIKE" EVENTS $\frac{dN_C}{dE_{reco}\,dx_{reco}}$

- How many of the arriving gamma-rays do we detect?
- How many do we miss-classify?
- How wrong do we get their energy?
- How wrong do we get their direction?
- How much background do we let through?

Instrument response functions

Gamma ray source $\frac{dN_{\gamma}}{dE_{true}\,dx_{true}}$ ⟷ "Gamma-like" events $\frac{dN_{C}}{dE_{reco}\,dx_{reco}}$

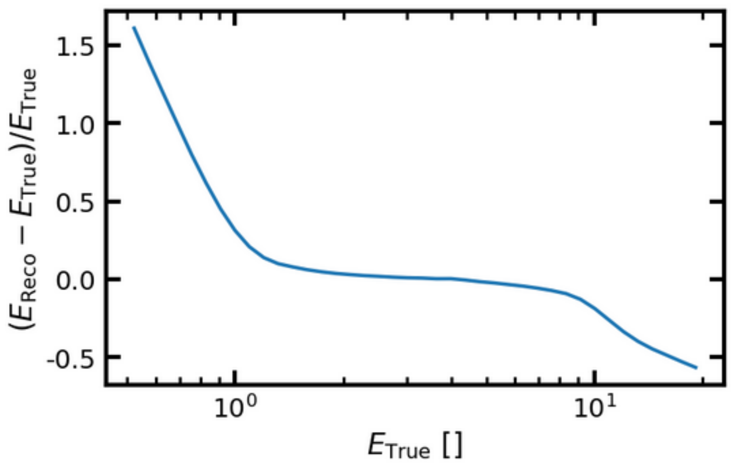
simulations!

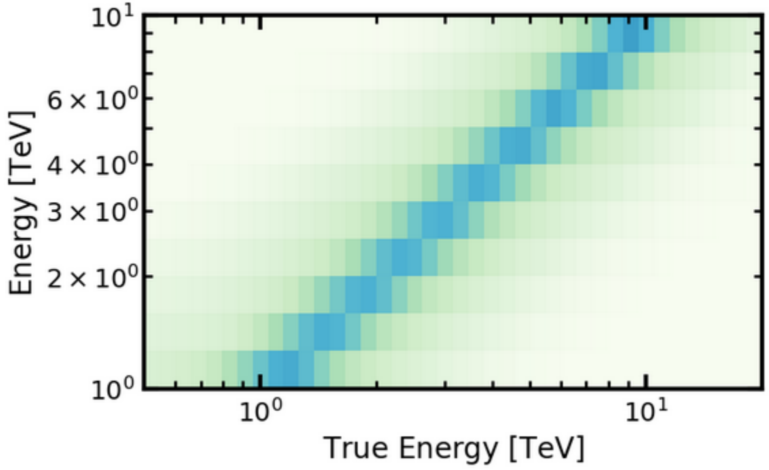
- How many of the arrivi… …tect?
- Ho…
- …rong do we get their energy?
- How wrong do we get their direction?
- How much background do we let through?

Energy dispersion

$EDISP(E_{true}, E_{reco})$

For each true gamma-ray energy, what is the probability that the event gets assigned a certain reconstructed energy?



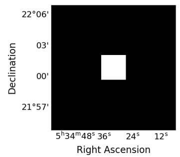


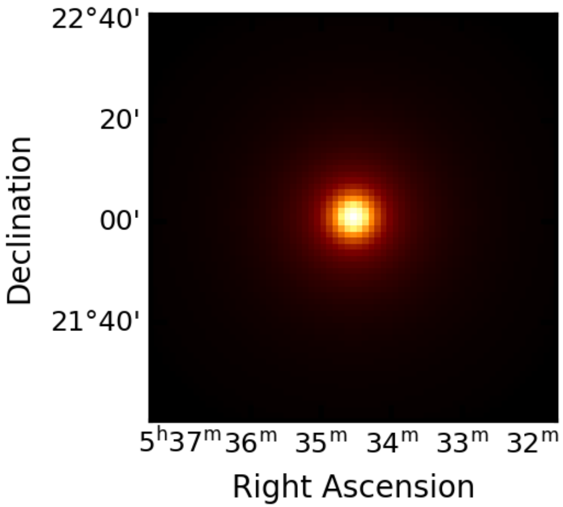
`DATASET.EDISP.PEEK()`

Point-Spread function

$PSF(E_{true}, x_{reco}, x_{true})$

For each true gamma-ray arriving direction, what is the probability that the event gets assigned a certain reconstructed direction ?



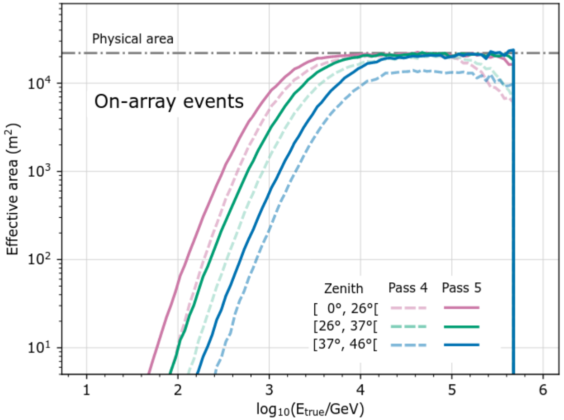


Effective Area

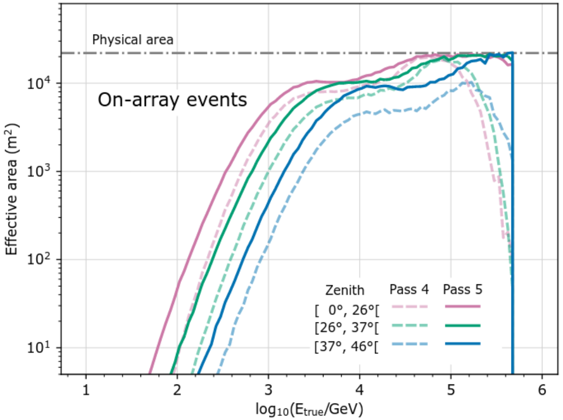
$$Aeff\left(E_{true}\right)$$

Detection probability of the gamma-ray (due to energy threshold + bad classification + instrumented area)

often multiplied by **livetime** to obtain "effective exposure" in units of m^2s



(a) Trigger conditions



(b) Gamma/hadron cuts

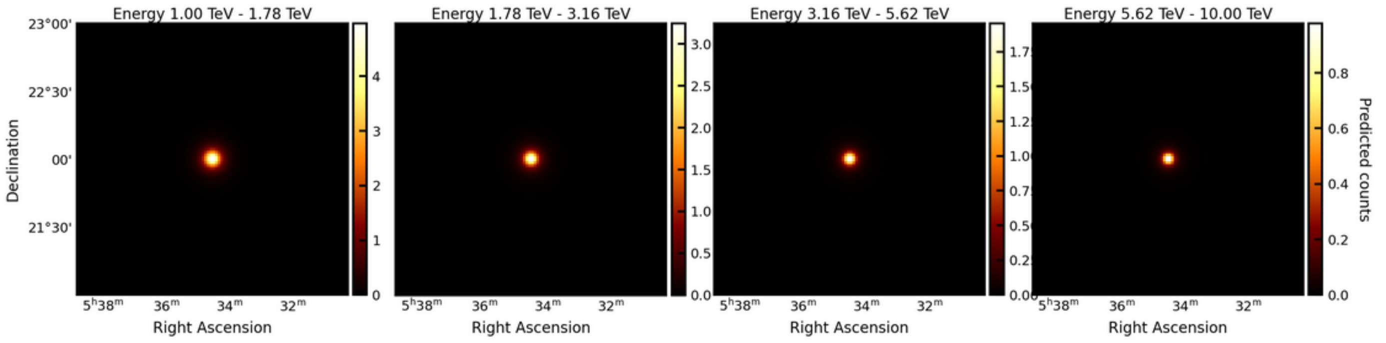
S. Yun-Cárcamo

So now we know how a source should look like!

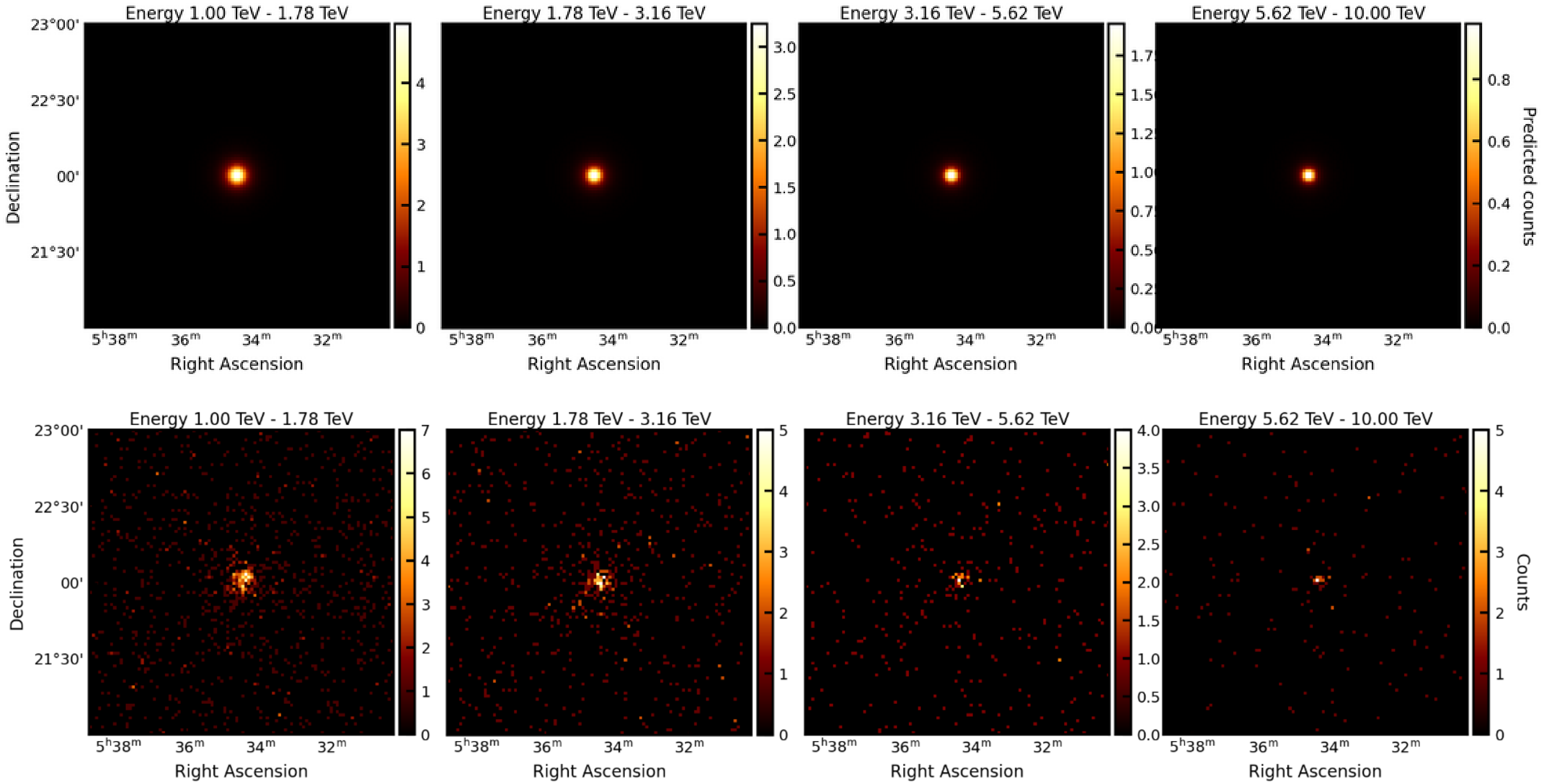
How would a **point source** with a **Crab like spectrum** look like in our detector?

- From the **spectrum** we know the flux (counts/s/TeV/cm2 in true energy) $\left(\frac{dN_\gamma}{dE_{true}}(x_{true})\right)$
- With **PSF** we shift from true to reconstructed position
- With **AEFF** and **livetime** we go from flux to counts (counts/TeV in true energy)
- With **EDISP** we shift from true to reconstructed energy (counts/TeV in reconstructed energy)
- We now can predict what we would observe if the model was a good description of reality!

```
dataset.npred_signal().plot_grid()
```

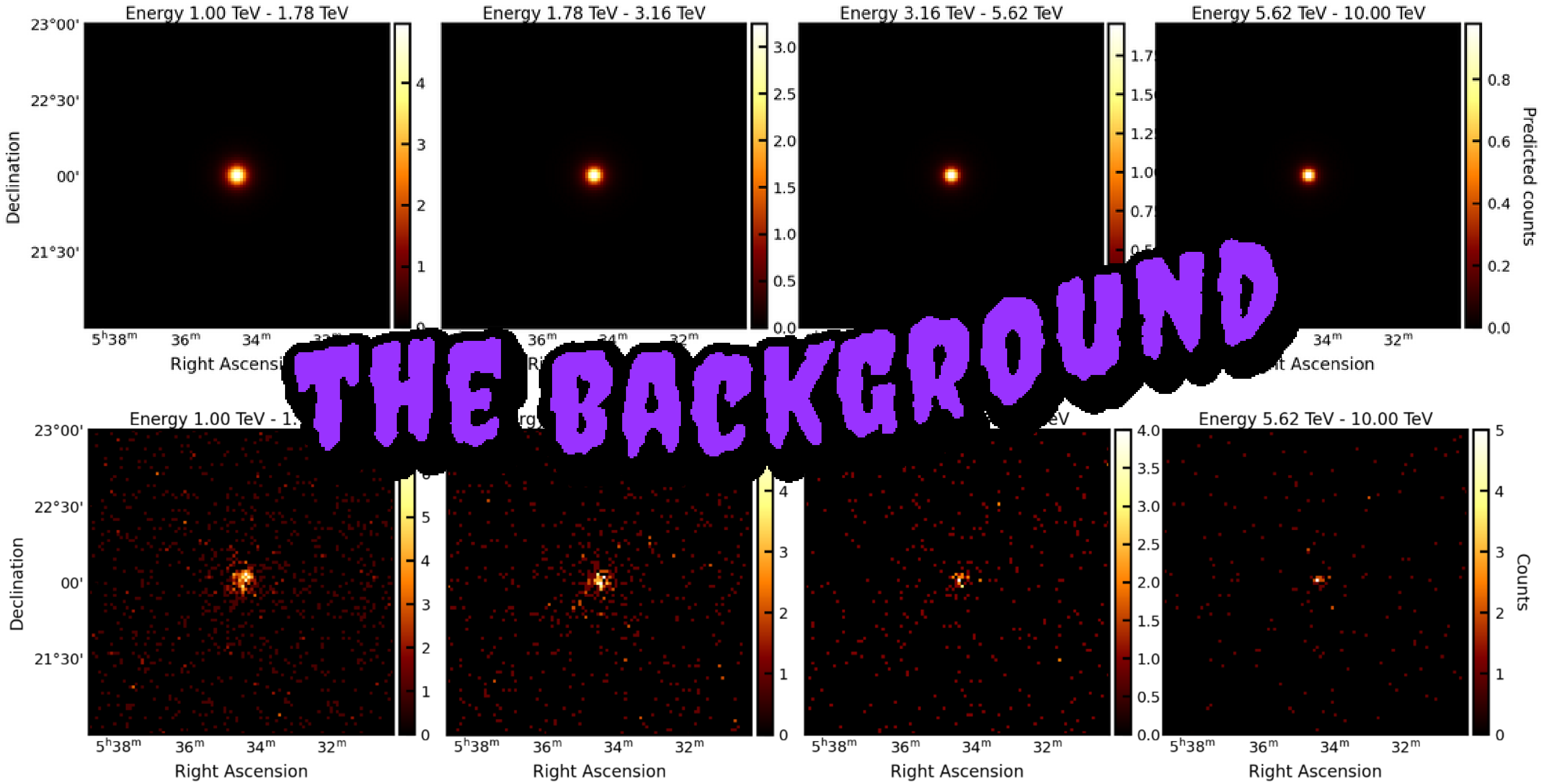


Is that all?



Energy 1.00 TeV - 1.78 TeV
Energy 1.78 TeV - 3.16 TeV
Energy 3.16 TeV - 5.62 TeV
Energy 5.62 TeV - 10.00 TeV
Declination
Right Ascension
Predicted counts
Counts

Is that all?



Energy 1.00 TeV - 1.78 TeV
Energy 1.78 TeV - 3.16 TeV
Energy 3.16 TeV - 5.62 TeV
Energy 5.62 TeV - 10.00 TeV
Declination
Right Ascension
Predicted counts
Energy 5.62 TeV - 10.00 TeV
Counts
THE BACKGROUND

Background

$N_{bkg}(E_{reco})$

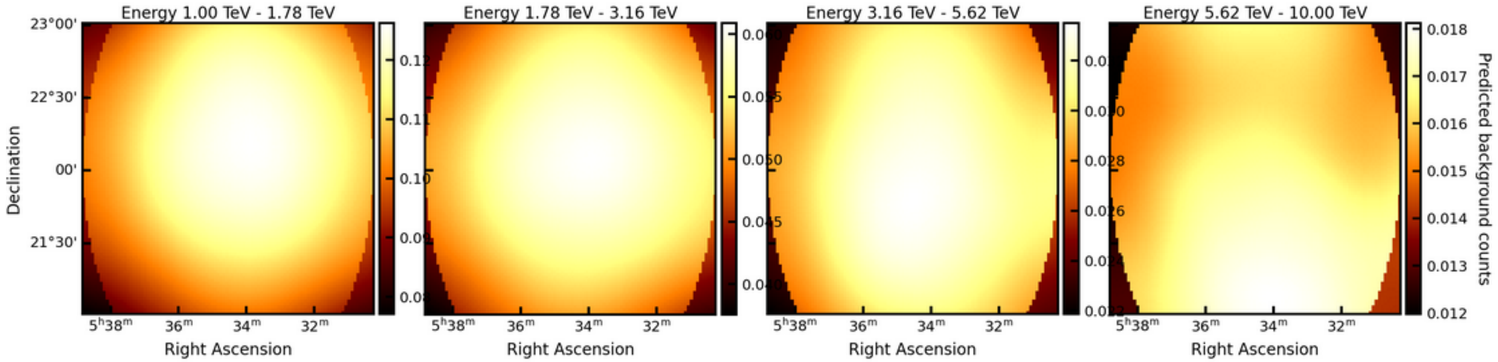
See Mohrmann et al 2019 to see how IACTs "caught up" with the other instruments ;-)

How many background events do we expect to mis-classify as signal?

A little bit different: usually derived from data of regions with no sources

- "Easy" in wide-field instruments, as the whole sky is rather empty in the TeV range
- Challenge for pointed instruments with small field of view: usually need to **assume where you expect your source to be (bias!!!)**

`dataset.background.plot_grid()`



Background

$N_{bkg}(E_{reco})$

See Mohrmann et al 2019 to see how IACTs "caught up" with the other instruments ;-)

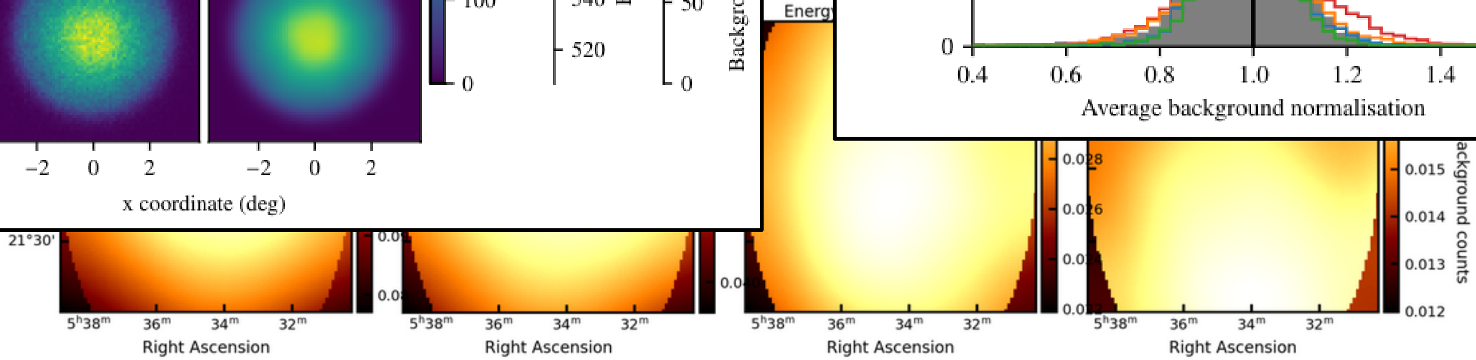
How many background events do we expect to ... y as signal?

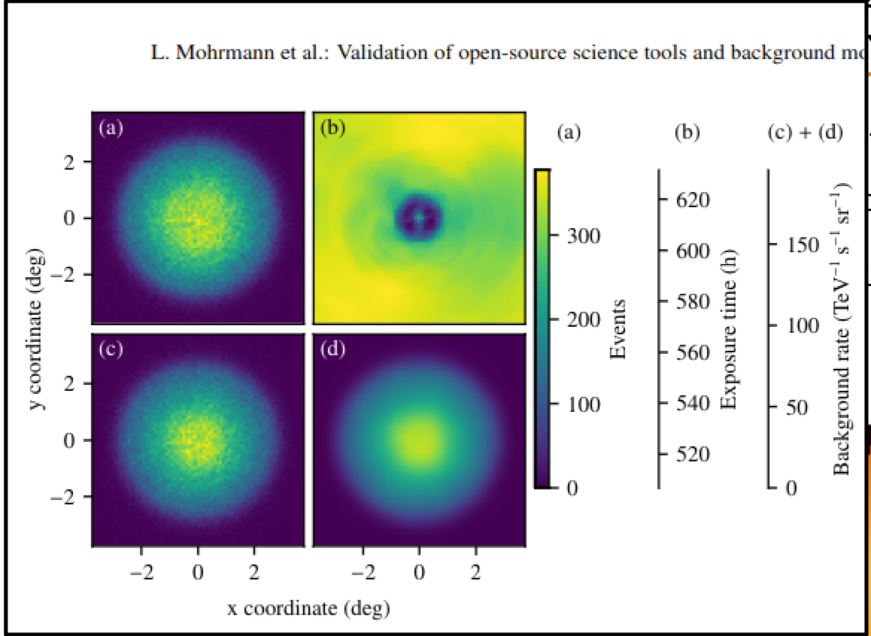
ATA

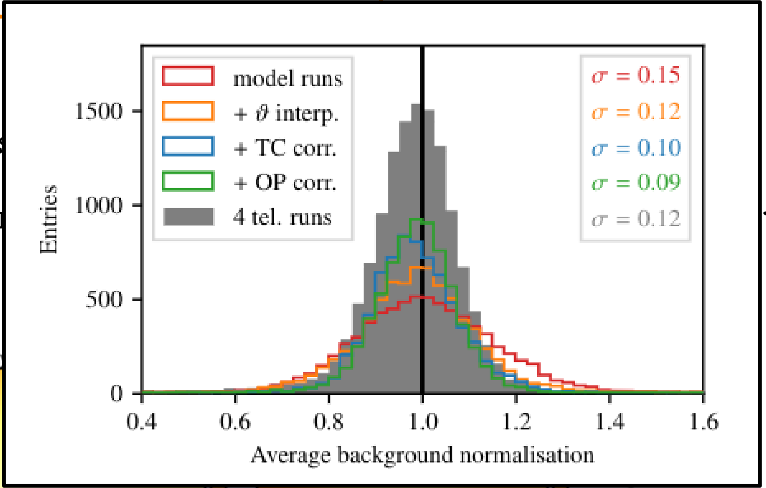
OLE

L FI

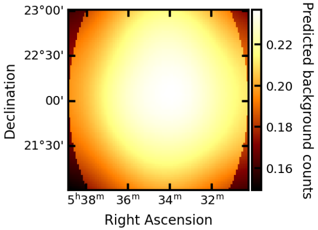
YOU

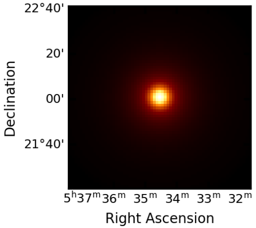


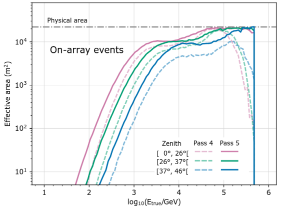




Now all together

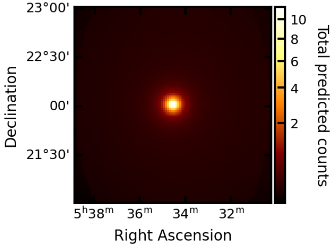


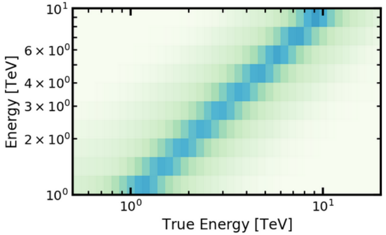


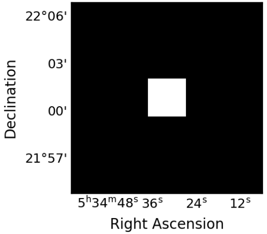


(b) Gamma/hadron cuts

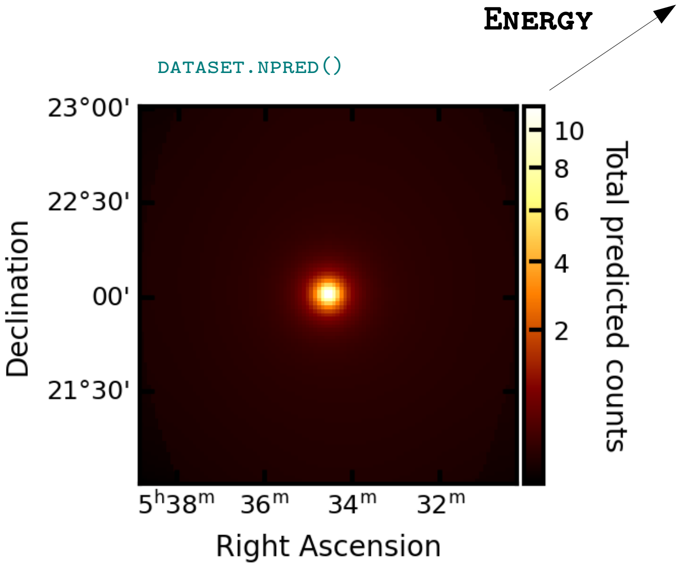
$$N_C(E_{reco}, x_{reco}) = N_{bkg}(E_{reco}, x_{reco}) + \int dx_{true} \int dE_{true}\, EDISP(E_{true}, E_{reco}) \times PSF(E_{true}, x_{reco}, x_{true}) \times Aeff(E_{true}) \times t_{live} \times \left(\frac{dN_\gamma}{dE_{true}}(x_{true}) \right)$$



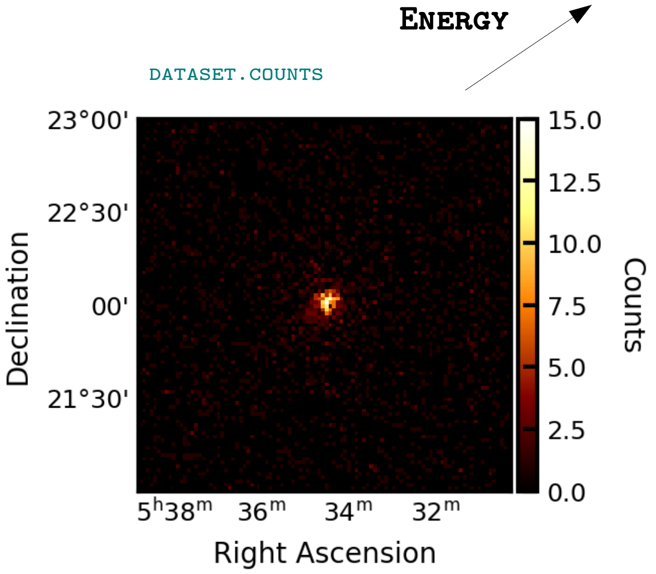




`DATASET.NPRED()`



`DATASET.COUNTS`



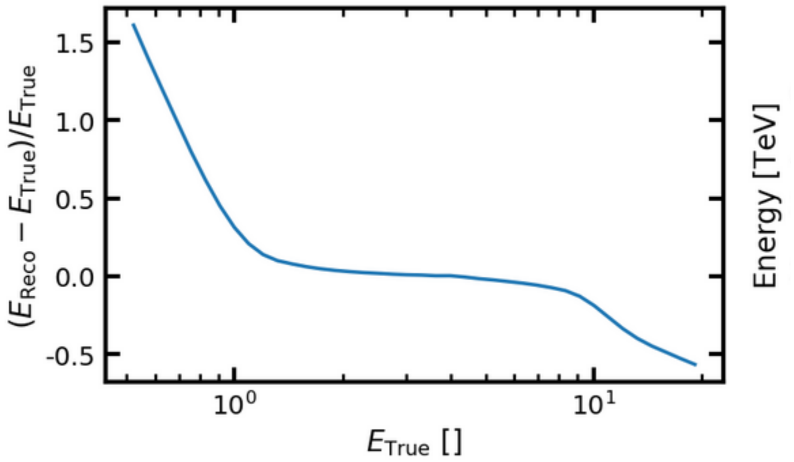
NOW WE CAN FIT A MODEL TO THE DATA BY COMPARING PREDICTION TO OBSERVATION!

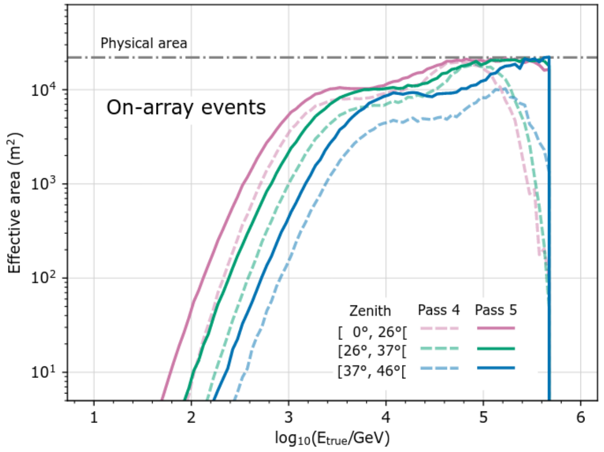
Data Analysis

- What does the data of gamma-ray instruments look like?
- What do I need to go from data to physical quantities?
- **How to determine the validity of data?**
- What is "data reduction"?

In which range is my data "valid"?

The IRFs also provide a very important tool in selecting our analysis range of validity





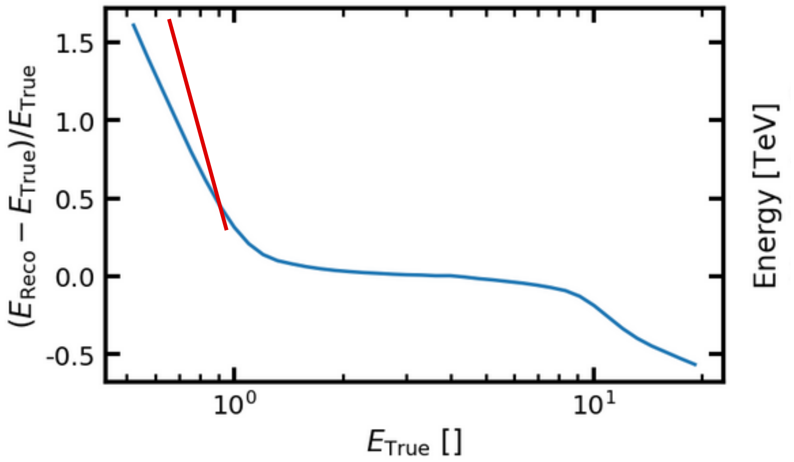
(b) Gamma/hadron cuts

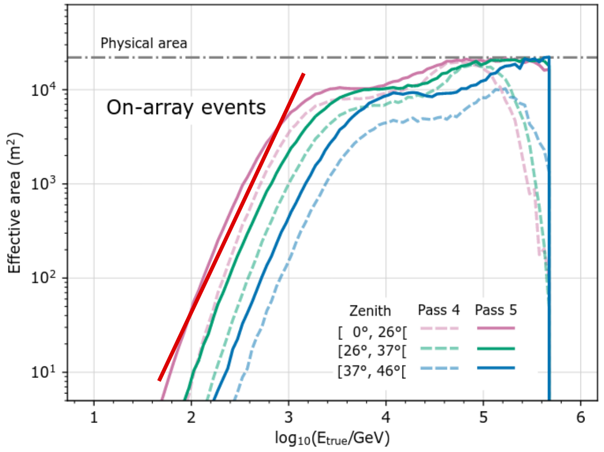
IN WHICH RANGE IS MY DATA "VALID"?

THE IRFs ALSO PROVIDE A VERY IMPORTANT TOOL IN SELECTING OUR ANALYSIS RANGE OF VALIDITY

REMEMBER: THEY ARE MADE WITH SIMULATIONS. SO THEY RELY HEAVILY ON **MC/DATA CONSISTENCY**

→ RANGES WHERE IRFs CHANGE RAPIDLY CAN BE DANGEROUS!





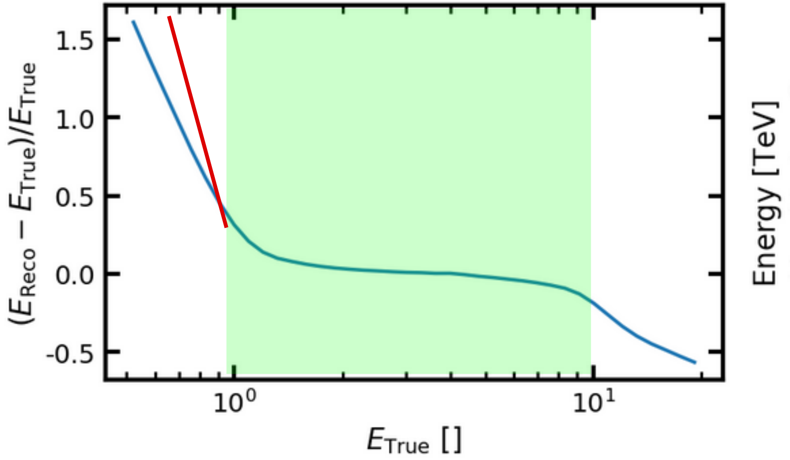
(b) Gamma/hadron cuts

In which range is my data "valid"?

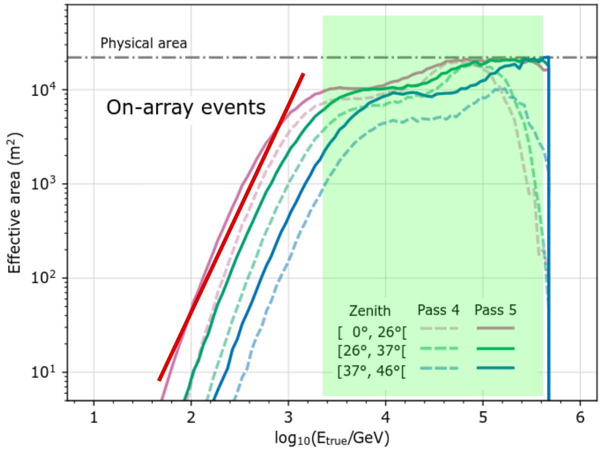
The IRFs also provide a very important tool in selecting our analysis range of validity

Remember: they are made with simulations. So they rely heavily on **MC/data consistency**

→ ranges where IRFs change rapidly can be dangerous!



`dataset.mask_safe.plot_grid()`



(b) Gamma/hadron cuts

Data Analysis

- What does the data of gamma-ray instruments look like?
- What do I need to go from data to physical quantities?
- How to determine the validity of data?
- **What is "data reduction"?**

Select and read data

```
from gammapy.data import DataStore

# Read bundle of event list, GTI and their corresponding IRF
data_store = DataStore.from_dir("$GAMMAPY_DATA/hess-dl3-dr1")

# Select runs around the Crab (only needed for pointing observations!)
selection = dict(
    type="sky_circle",
    frame="icrs",
    lon="83.633 deg",
    lat="22.014 deg",
    radius="5 deg",
)
selected_obs_table = data_store.obs_table.select_observations(selection)
observations = data_store.get_observations(selected_obs_table["OBS_ID"])

# Look at data
observations[0].events.table # event table
observations[0].GTI.table # event table

observations[0].aeff.peek() #aeff
observations[0].edisp.peek() #edisp
observations[0].psf.peek() #psf
observations[0].bkg.peek() #bkg
```

IN "DETECTOR" COORDINATES
(OFFSET, ZENITH...)

Reduce data

GO FROM DETECTOR TO SKY COORDINATES AND BUNDLE

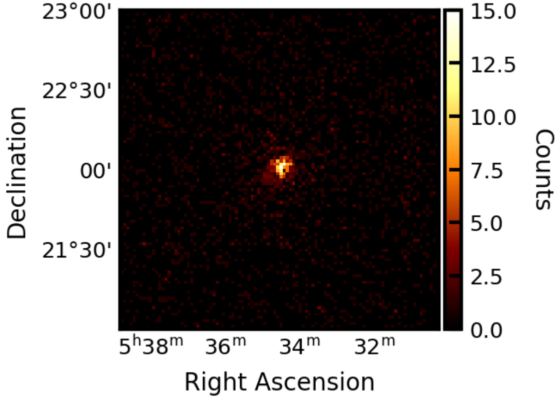
```
from gammapy.maps import WcsGeom, MapAxis
from gammapy.datasets import MapDataset
from regions import CircleSkyRegion

# Define energy axes
energy_axis = MapAxis.from_energy_bounds(1.0, 10.0, 10, unit="TeV")
energy_axis_true = MapAxis.from_energy_bounds(0.5, 20, 40, unit="TeV", name="energy_true") # always more range and more bins than reco!

# Define the sky geometry
geom = WcsGeom.create(
    skydir=(83.633, 22.014),
    binsz=0.02,
    width=(2, 2),
    frame="icrs",
    proj="CAR",
    axes=[energy_axis],
)

# create an empty dataset
stacked = MapDataset.create(geom=geom, energy_axis_true=energy_axis_true, name="crab-stacked")
```

Reduce data

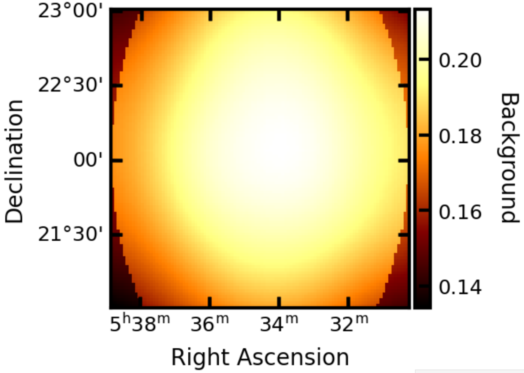
GO FROM DETECTOR TO SKY COORDINATES AND BUNDLE

```python
# Define the makers
maker = MapDatasetMaker()
maker_safe_mask = SafeMaskMaker(methods=["offset-max", "aeff-max"], offset_max="2.5 deg")

circle = CircleSkyRegion(center=SkyCoord("83.63 deg", "22.14 deg"), radius=0.2 * u.deg)
exclusion_mask = geom.region_mask(regions=[circle], inside=False)
maker_fov_bkg = FoVBackgroundMaker(method="fit", exclusion_mask=exclusion_mask)

for obs in observations:
    # First a cutout of the target map is produced
    cutout = stacked.cutout(
        obs.get_pointing_icrs(obs.tmid), width=2 * offset_max, name=f"obs-{obs.obs_id}"
    )
    # A MapDataset is filled in this cutout geometry
    dataset = maker.run(cutout, obs)
    # The data quality cut is applied
    dataset = maker_safe_mask.run(dataset, obs)
    # fit background model
    dataset = maker_fov.run(dataset)
    print(
        f"Background norm obs {obs.obs_id}: {dataset.background_model.spectral_model.norm.value:.2f}"
    )
    # The resulting dataset cutout is stacked onto the final one
    stacked.stack(dataset)
```

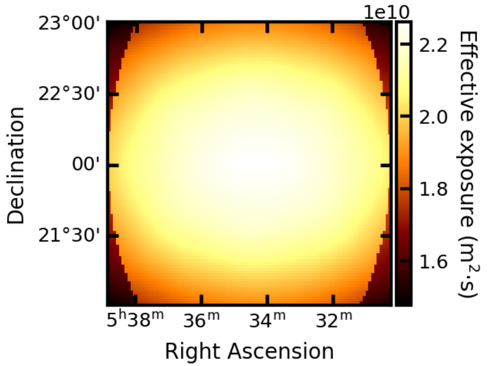
```
ax = stacked.counts.reduce_over_axes().plot(add_cbar=True)
cbar = ax.images[-1].colorbar
cbar.set_label('Counts', rotation=270, labelpad=30)
```



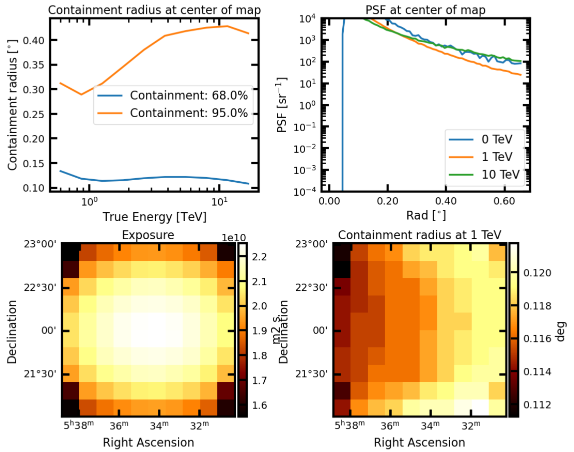

```
ax = stacked.background.reduce_over_axes().plot(add_cbar=True)
cbar = ax.images[-1].colorbar
cbar.set_label('Background', rotation=270, labelpad=30)
```



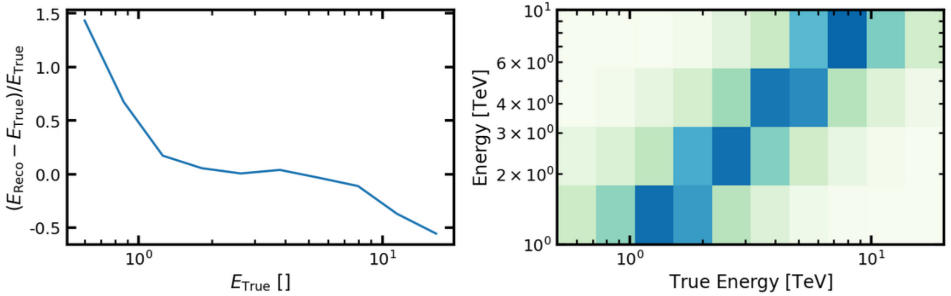

```
ax = stacked.exposure.reduce_over_axes().plot(add_cbar=True)
cbar = ax.images[-1].colorbar
cbar.set_label('Effective exposure (m$^2$·s)', rotation=270, labelpad=30)
```




```
stacked.psf.peek()
```




```
stacked.edisp.peek()
```



DL3 → DL4 → DL5

DL3
γ-like events

DL4
Binned data

DL5
Science products

Data reduction

CHOOSE A SKY GEOMETRY AND BIN/PROJECT THE EVENTS AND IRFs

.data
- DataStore
- Observations
- Observation
- GTI

.makers
- MapDatasetMaker
- SafeMaskMaker
- FoVBackgroundMaker
- RingBackgroundMaker
- etc.

.datasets
- Datasets
- MapDataset
- MapDatasetOnOff
- etc.

Likelihood fitting

.modeling
- Fit, Models, SkyModel
- FoVBackgroundModel
- etc.

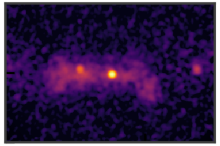
.estimators
- FluxPointsEstimator
- FluxMapEstimator
- etc.

YAML → **.analysis**

Source Catalogs

Name	Flux	Size
SNR	1e-12	1 deg
PWN	1e-11	0.2 deg
GRB	1e-10	0 deg

Flux & TS Maps



SEDs & Lightcurves