Full unbinned analysis

Giacomo D’Amico, Tim Unbehaun, Julia Djuvsland

Unbinned likelihood

$$N_{pred} \equiv \int dE \int d\mathbf{r} \;\; \phi(E, \mathbf{r})$$

$$-2 \log \mathcal{L} = 2 \cdot N_{pred} - 2 \sum_i \phi(E_i, \mathbf{r}_i)$$

Observed flux

Is there a way to get this quantity for all events (i=1,…, N) without having to define a binning in reconstructed energy and direction and without having to perform an interpolation?

For the i-th event with observed energy E_i and reconstructed direction $\mathbf{r_i}$

Observed flux — Energy Dispersion — PSF — Collection Area — True flux

$$\phi(E_i, \mathbf{r}_i) = \int dE' \int d\mathbf{r}' \; \mathrm{D}(E_i|E', \mathbf{r}') \times \mathrm{P}(\mathbf{r}_i|E', \mathbf{r}') \times \mathrm{A}(E', \mathbf{r}') \times \phi'(E', \mathbf{r}')$$

i = 1, …, total # of events

Numerical integration

$$\phi(E_i, \mathbf{r}_i) = \sum_k \Delta E'_k \sum_{l,b} \Delta \mathbf{r}'_{l,b} \; \mathrm{D}(E_i|E'_k, \mathbf{r}'_{l,b}) \times \mathrm{P}(\mathbf{r}_i|E'_k, \mathbf{r}'_{l,b}) \times \mathrm{A}(E'_k, \mathbf{r}'_{l,b}) \times \phi'(E'_k, \mathbf{r}'_{l,b})$$

i : index from 1 to **total number of events**
k : index of binning in **true energy**
l : index of binning in **right ascension**
b : index of binning in **declination**

Unlike the “binned” approach, we only need a **“true” geometry** which is used for performing the **numerical integration**

For the i-th event with observed energy E_i and reconstructed direction $\mathbf{r_i}$

$$\phi(E_i, \mathbf{r}_i) = \sum_k \Delta E'_k \sum_{l,b} \Delta \mathbf{r}'_{l,b} \;\; \mathrm{D}(E_i | E'_k, \mathbf{r}'_{l,b}) \times \mathrm{P}(\mathbf{r}_i | E'_k, \mathbf{r}'_{l,b}) \times \mathrm{A}(E'_k, \mathbf{r}'_{l,b}) \times \phi'(E'_k, \mathbf{r}'_{l,b})$$

taking into account the circular symmetry of the IRF

$$\phi(E_i, \mathbf{r}_i) = \sum_k \Delta E'_k \sum_{l,b} \Delta \mathbf{r}'_{l,b} \;\; \mathrm{D}(E_i | E'_{i,k}, \mathrm{O}_{l,b}) \times \mathrm{P}(d_{i,l,b} | E'_k, \mathrm{O}_{l,b}) \times \mathrm{A}(E'_k, \mathrm{O}_{l,b}) \times \phi'(E'_k, \mathbf{r}'_{l,b})$$

$$\mathrm{O}_{l,b} = |r'_{l,b} - \mathrm{pointing}|$$

"Offset" = distance between the telescope pointing and the true direction

$$d_{i,l,b} = |r'_{l,b} - r_i|$$

"rad" = distance between the true and the reconstructed direction

Dataset used for the tests

DataStore.from_dir("$GAMMAPY_DATA/hess-dl3-dr1")

obs_id = 23523

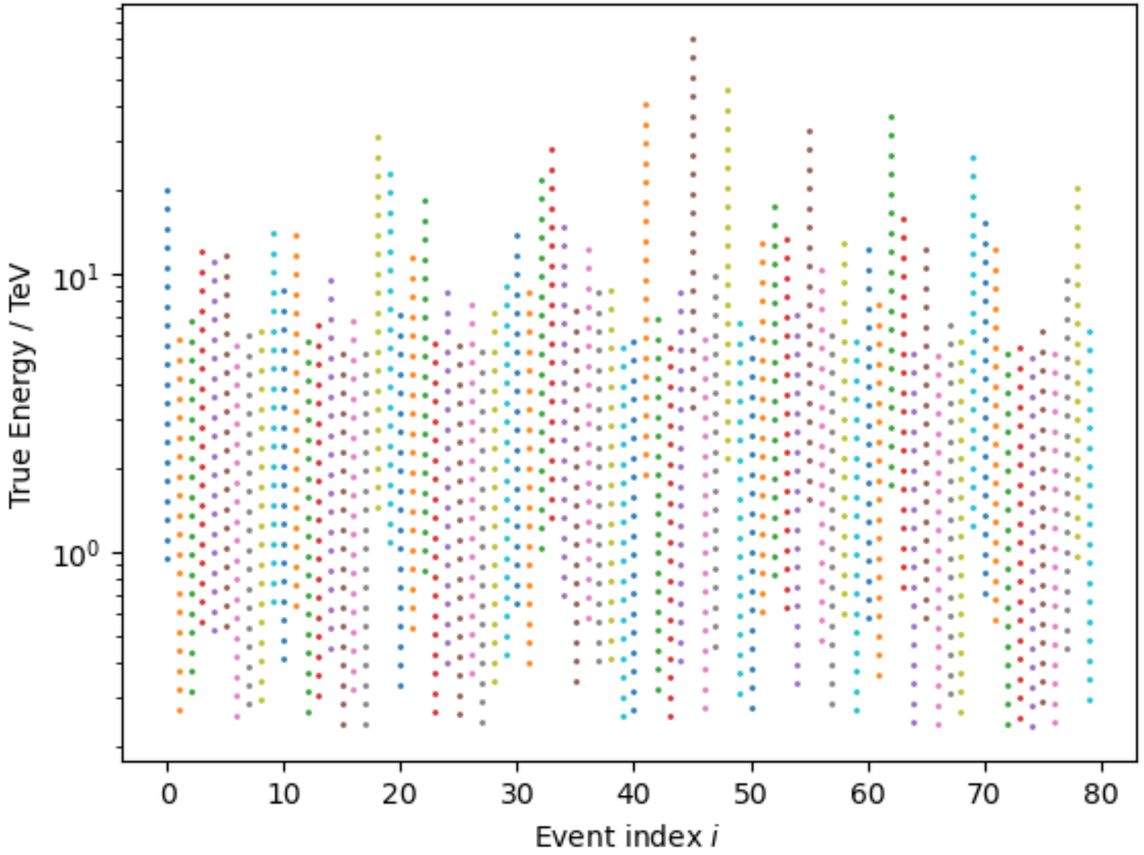
```
events = obs1.events.select_energy( (0.8*u.TeV, 20*u.TeV))
events = events.select_offset(    (0,1.5)*u.deg)
len(events.table)
```

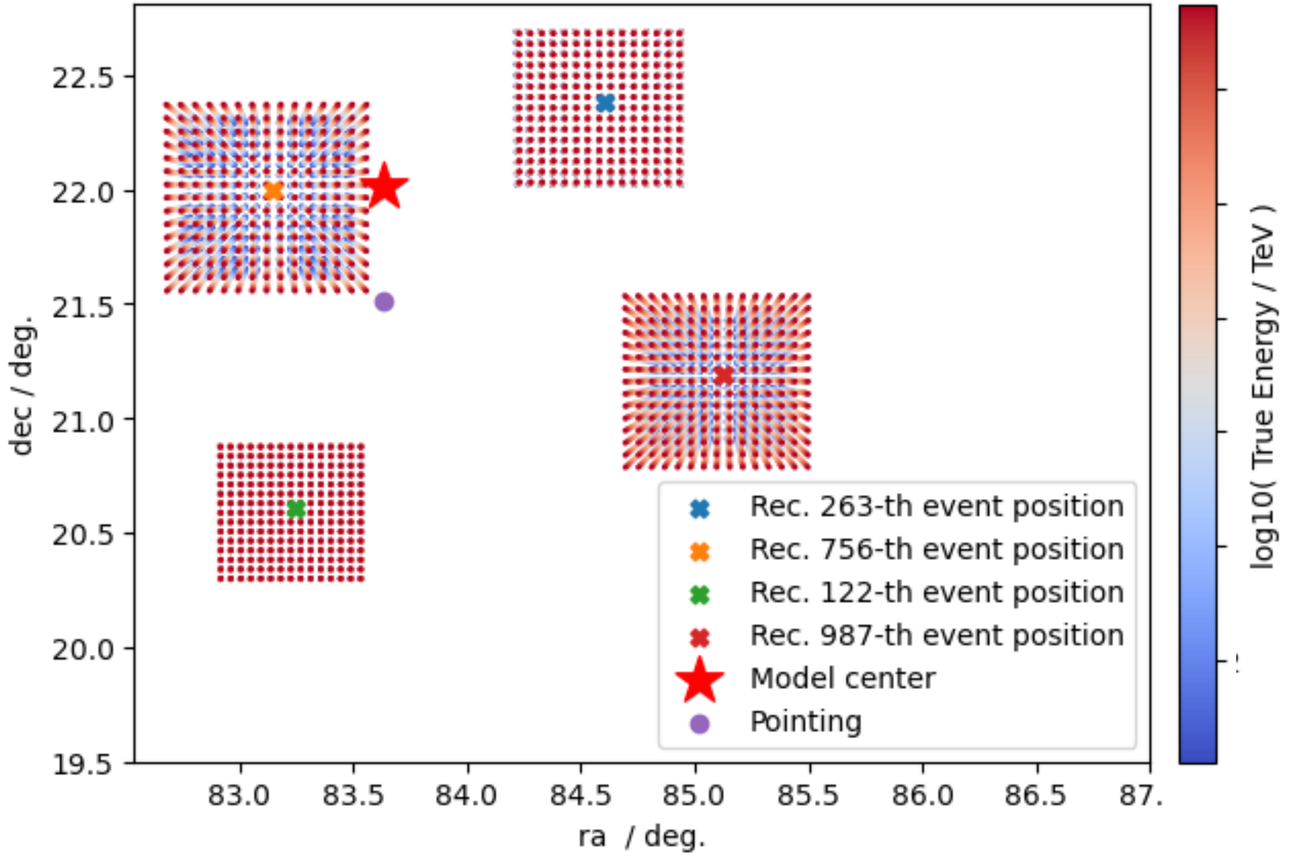
```
1665
```

Table length=7613

EVENT_ID	TIME	RA	DEC	ENERGY
	s	deg	deg	TeV
int64	float64	float32	float32	float32
5407363825684	123890826.66805482	84.97964	23.89347	10.352011
5407363825695	123890826.69749284	84.54751	21.004095	4.0246882
5407363825831	123890827.23673964	85.39696	19.41868	2.2048872
5407363825970	123890827.79615426	81.93147	20.79867	0.69548655
5407363826067	123890828.26131463	85.98302	21.053099	0.86911184
5407363826095	123890828.41393518	86.97305	21.837437	4.1240892
5407363826128	123890828.52555823	83.40073	19.771587	1.6680022
5407363826168	123890828.6829524	82.25036	19.22003	4.7649446
5407363826383	123890829.53362775	83.18322	22.008213	0.7920148
...	...	...	...	...
7198365188529	123892511.59526515	85.58907	21.491413	0.6518216
7198365188578	123892511.85330749	85.239136	23.94445	6.0601764
7198365188605	123892511.93477488	82.4973	22.135513	0.5850589
7198365188619	123892512.01627827	81.832565	20.889784	0.79988396
7198365188742	123892512.49603844	83.06237	22.34547	0.5771867
7198365188792	123892512.74561787	83.14342	23.21574	9.994803
7198365188797	123892512.78117561	81.45052	22.960012	5.6078057
7198365188816	123892512.85406923	84.41192	21.330505	0.5524084
7198365188843	123892513.0062654	84.47432	21.634737	1.1091197

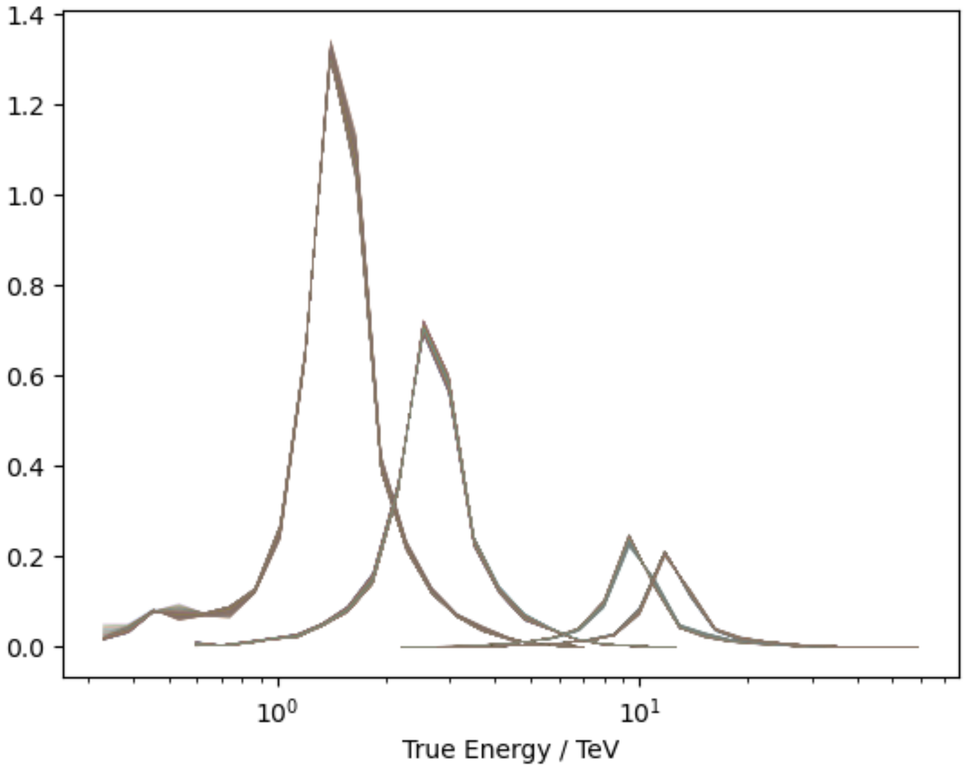
Approach #1: True geometry tailored to the event reconstructed energy and direction





Energy Dispersion

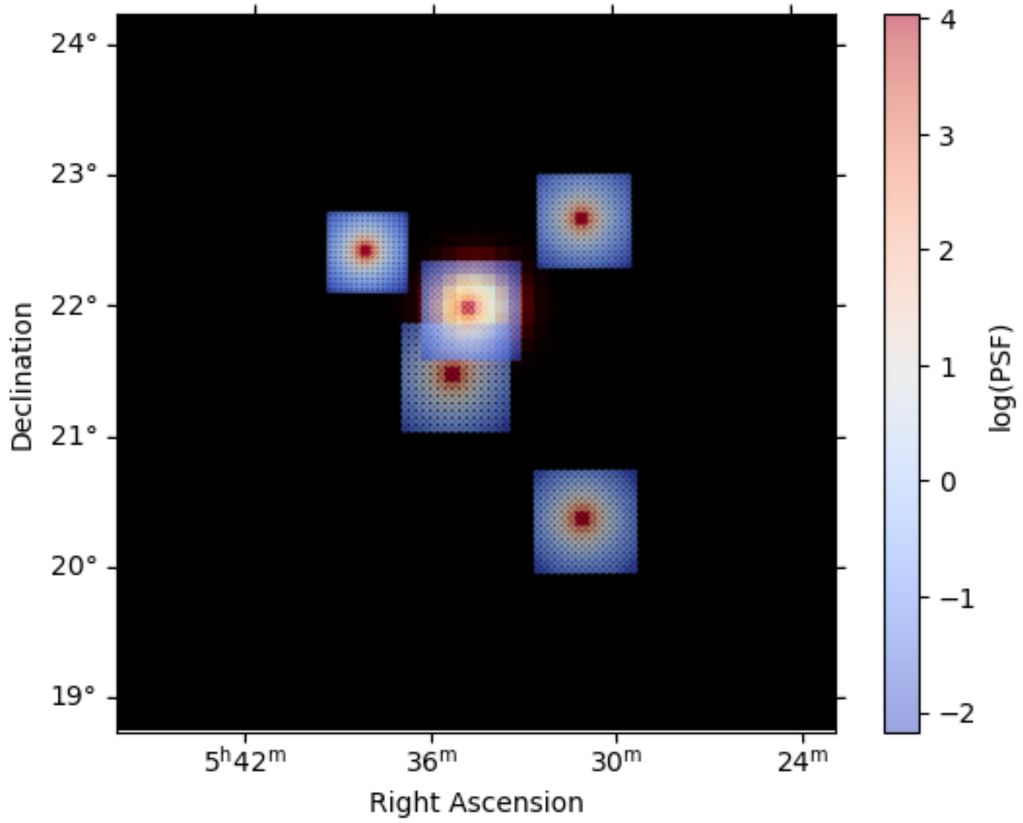
Unit = 1 / TeV




```
ereco_events       =  events.energy[:,None,None,None]

edisp2D        = observation.edisp
edisp2D.normalize()
edisp        =  edisp2D.evaluate(
                 offset =true_offset,
                 energy_true=etrue,
                 migra=ereco_events/etrue
                ) / etrue
```

PSF

Unit = 1 / deg^2

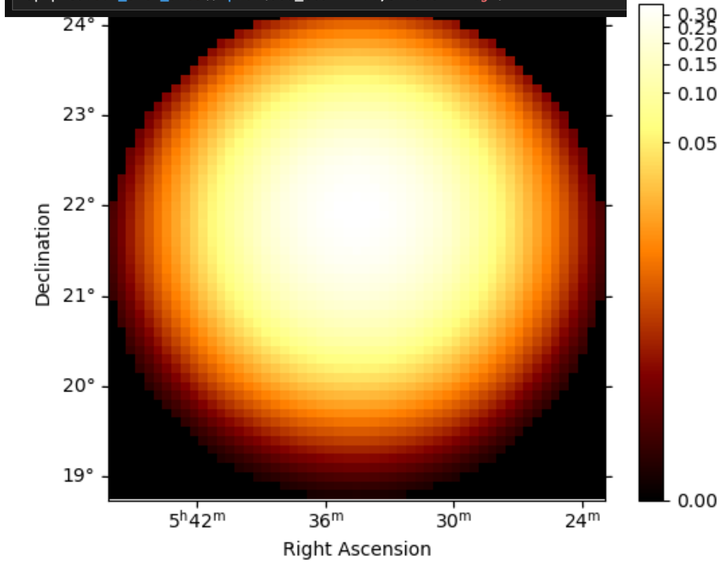


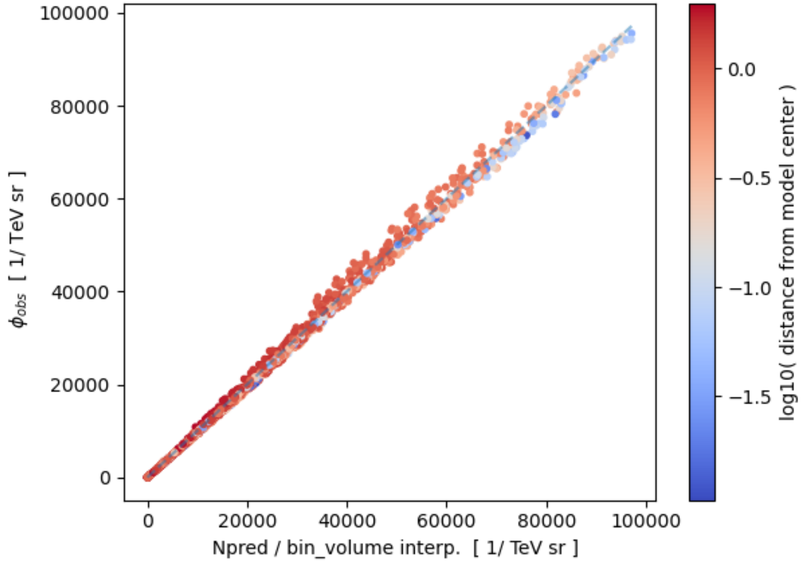

```
psf3d        = observation.psf
psf3d.normalize()

psf_factors = psf3d.evaluate(
                energy_true=etrue,
                rad=rad,
                offset=true_offset
              )
```

```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",           lat_0     = "-5.784 deg", sigma = '1 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

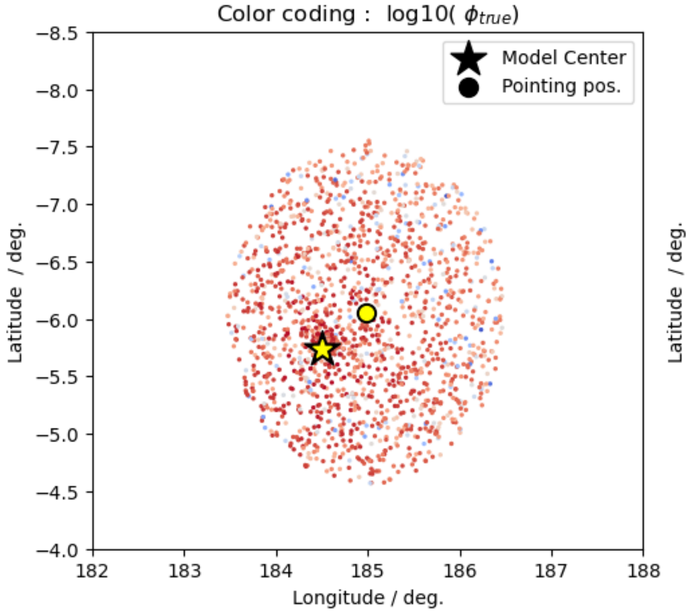
Sigma = 1 deg.

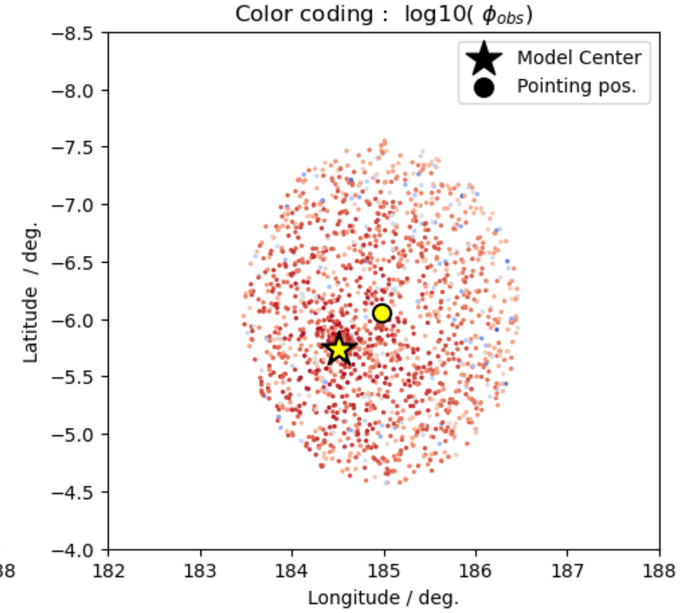
```
mapnpred      = dataset.npred()
mapnpred      = dataset.evaluators['crab_model_gauss'].compute_npred()

mapnpred.sum_over_axes().plot(add_cbar=True,stretch='log')
```

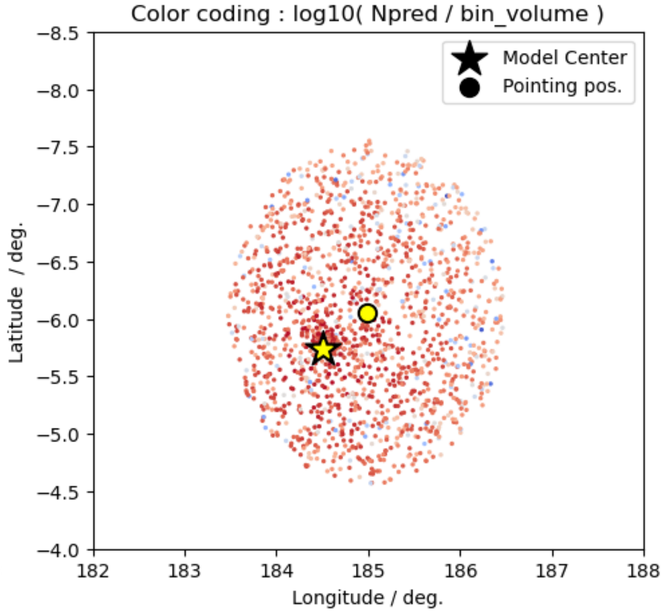





```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```



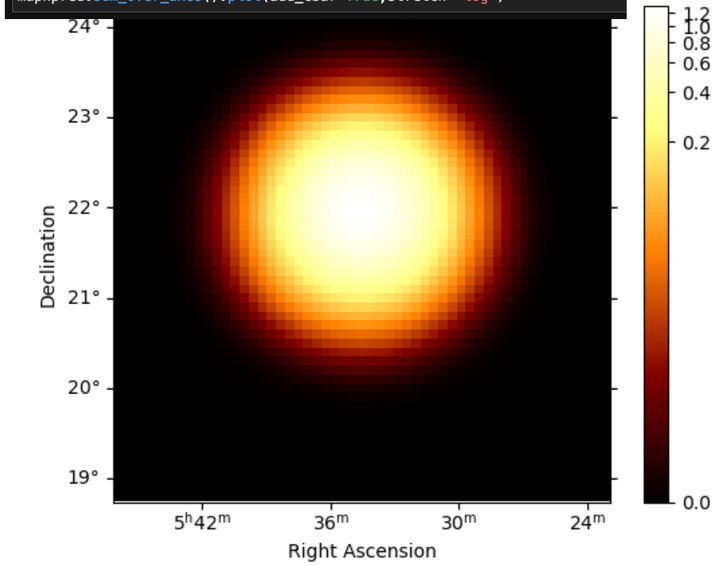


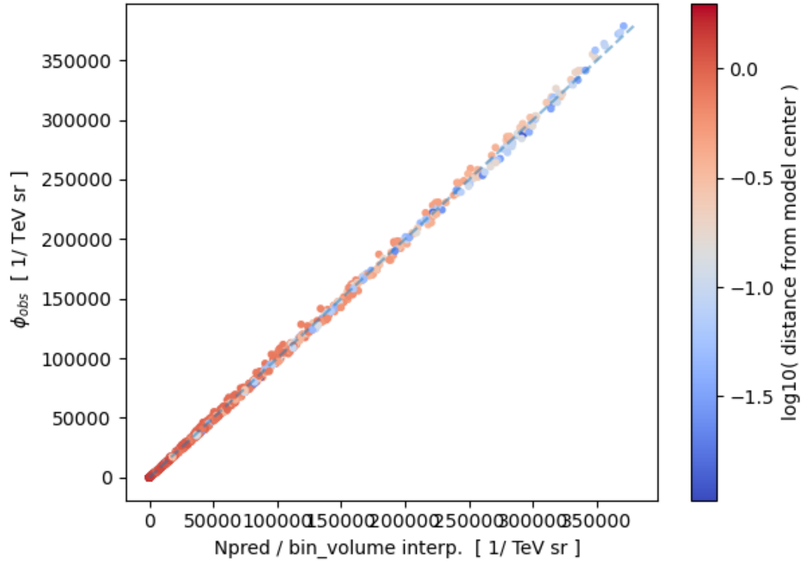


Sigma = 0.5 deg.

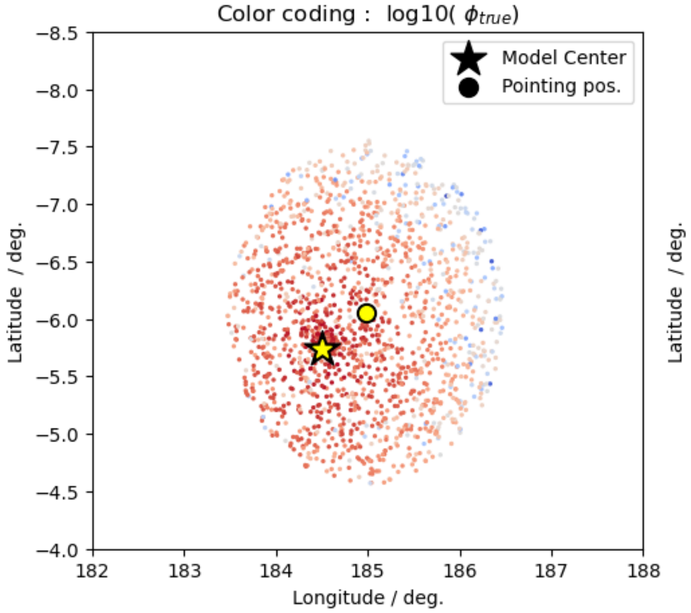
```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",          lat_0     = "-5.784 deg", sigma = '0.5 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

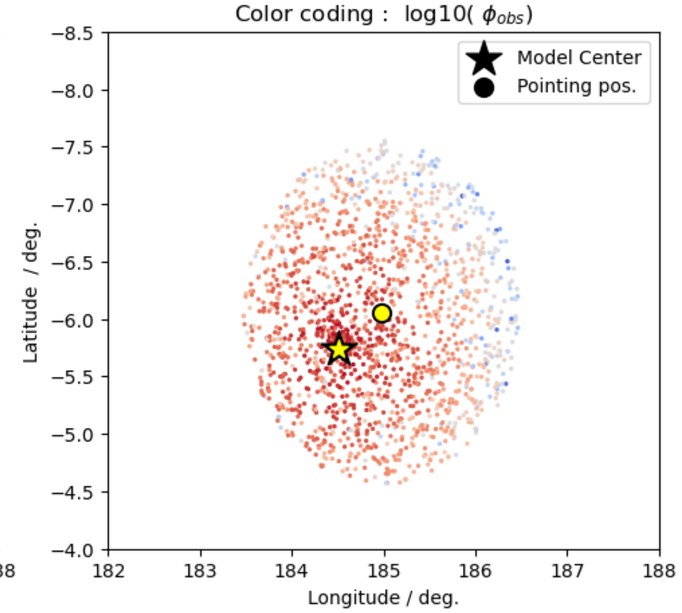
```
mapnpred      = dataset.npred()
mapnpred      = dataset.evaluators['crab_model_gauss'].compute_npred()

mapnpred.sum_over_axes().plot(add_cbar=True,stretch='log')
```

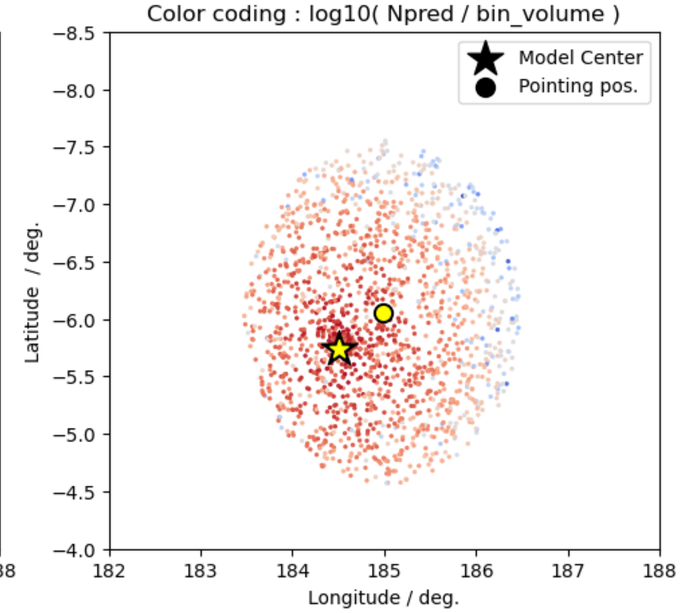





```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```



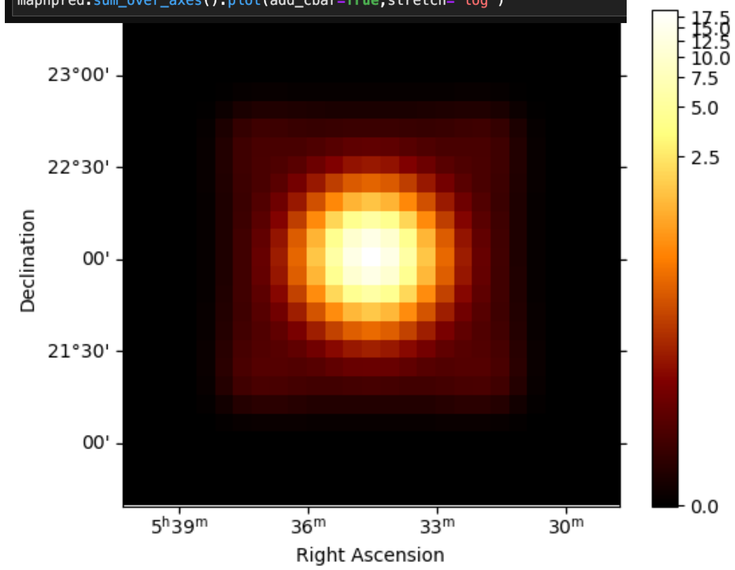


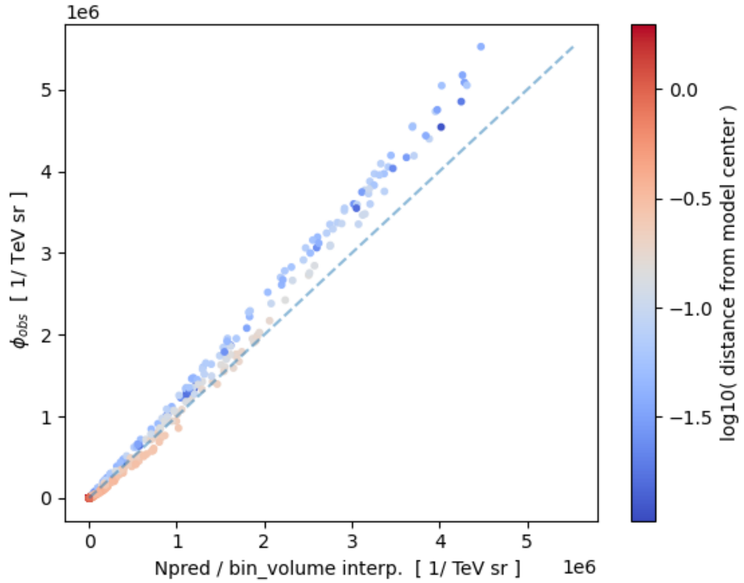


```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",           lat_0     = "-5.784 deg", sigma = '0.1 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

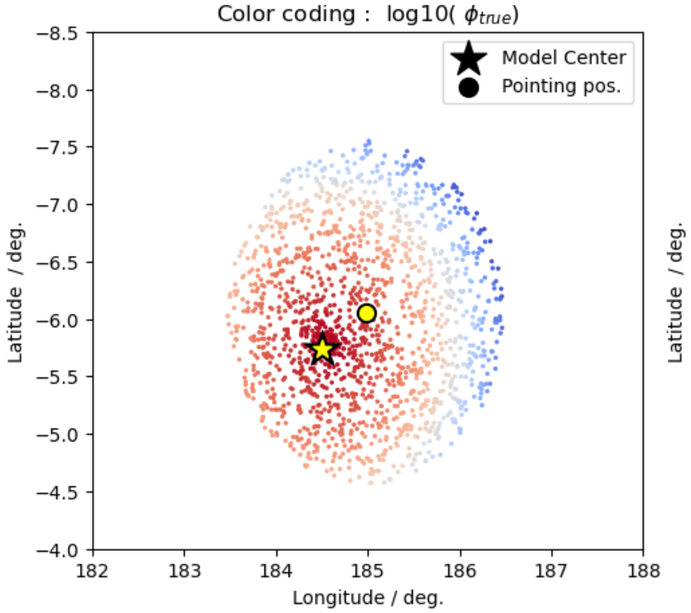
Sigma = 0.1 deg.

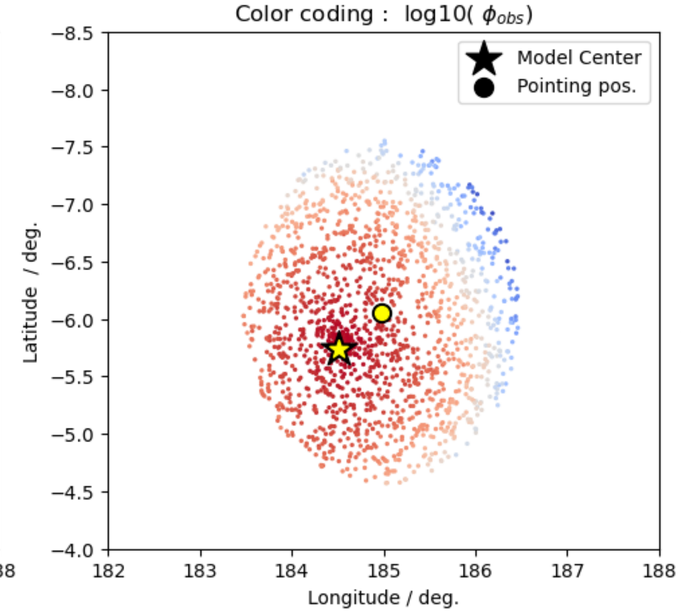
```
mapnpred      = dataset.npred()
mapnpred      = dataset.evaluators['crab_model_gauss'].compute_npred()

mapnpred.sum_over_axes().plot(add_cbar=True,stretch='log')
```

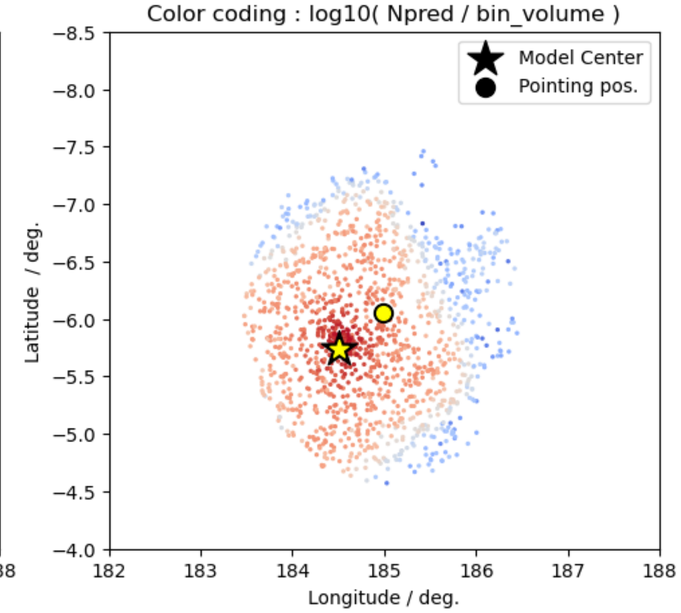





```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```



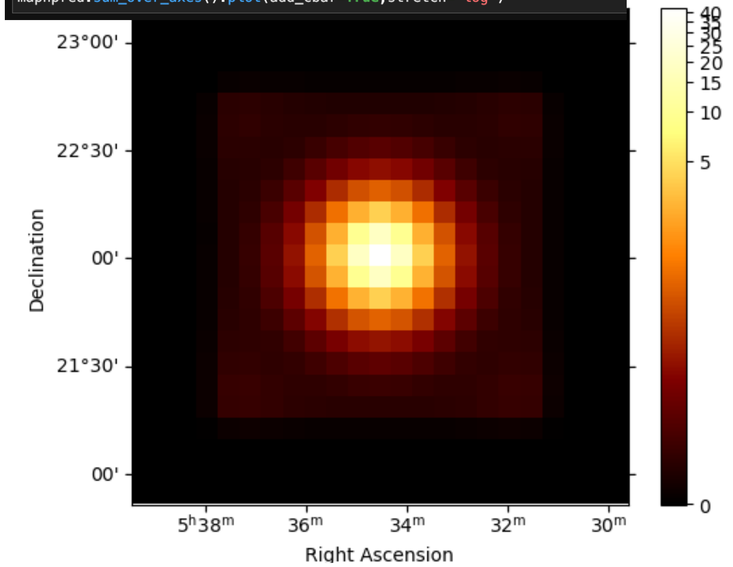


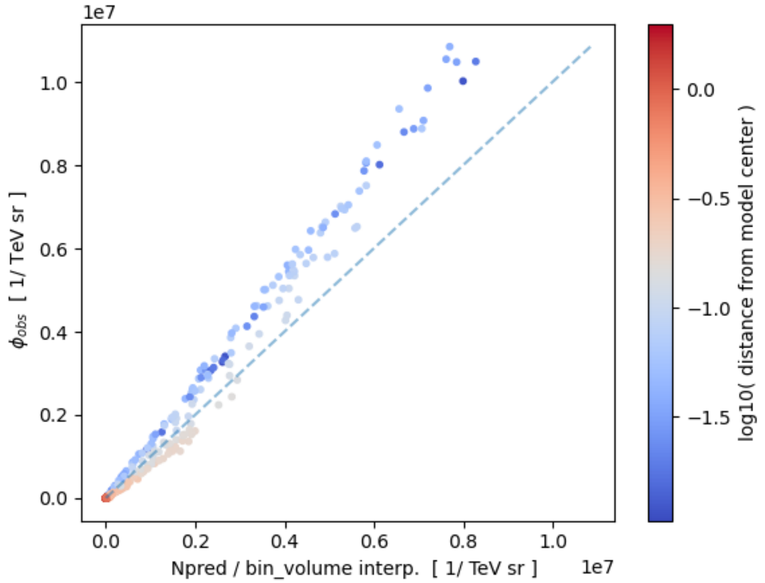


```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",           lat_0     = "-5.784 deg", sigma = '0.05 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

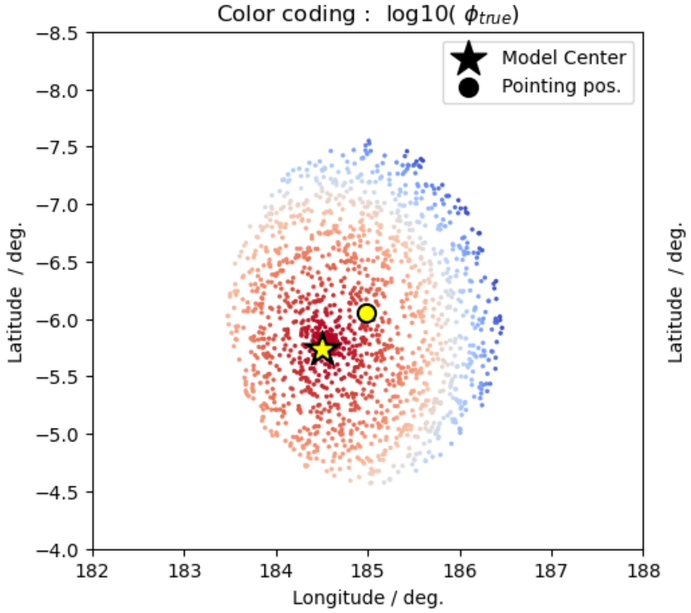
Sigma = 0.05 deg.

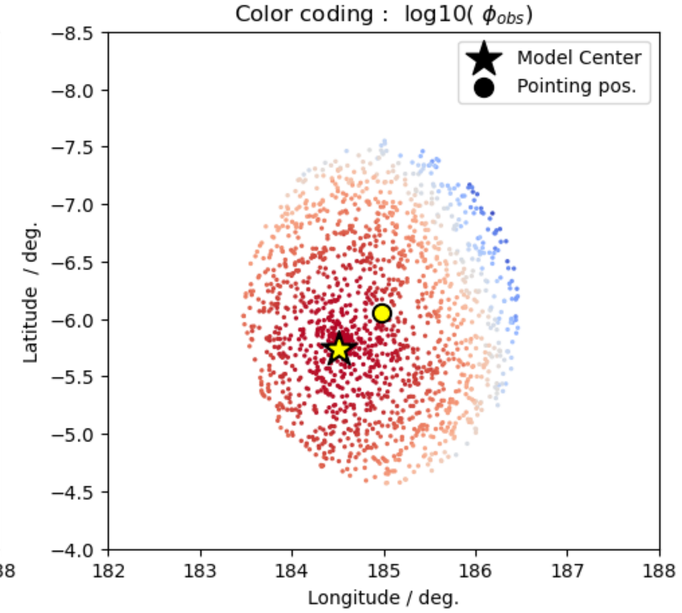
```
mapnpred      = dataset.npred()
mapnpred      = dataset.evaluators['crab_model_gauss'].compute_npred()

mapnpred.sum_over_axes().plot(add_cbar=True,stretch='log')
```

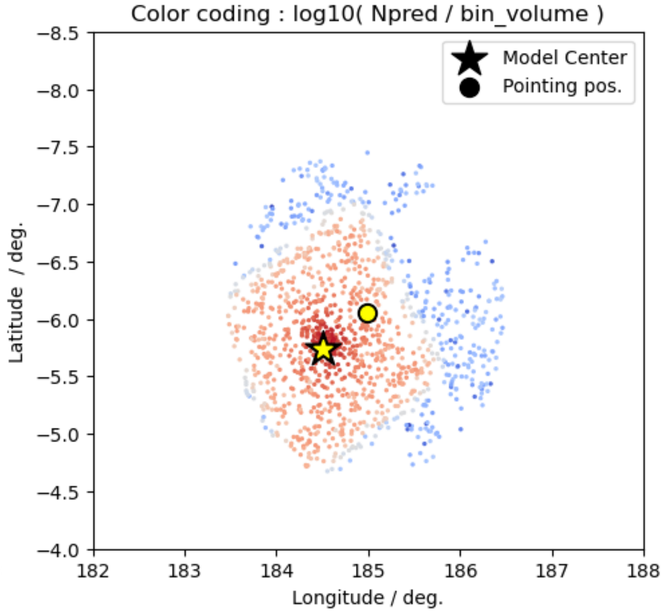


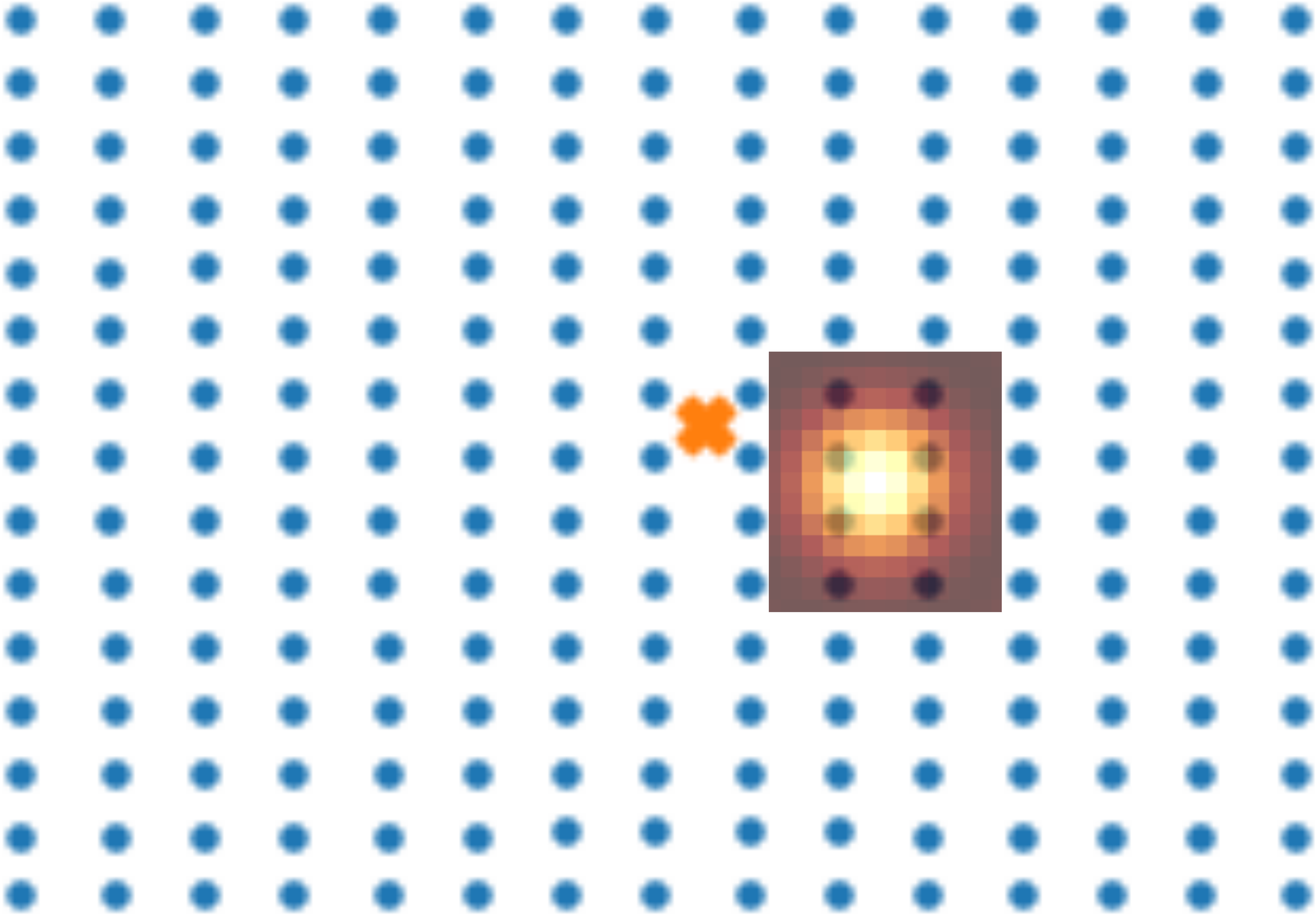



```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```

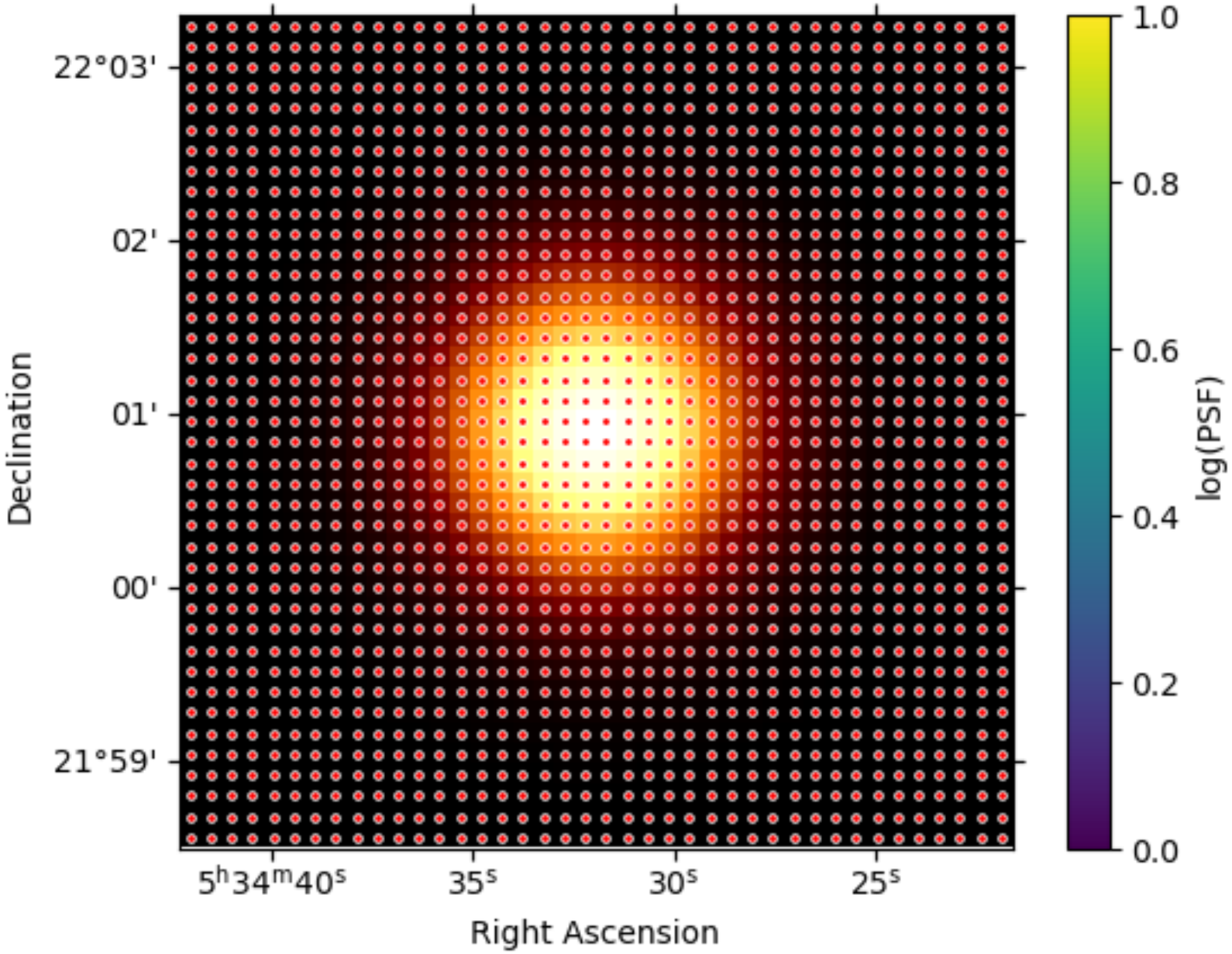






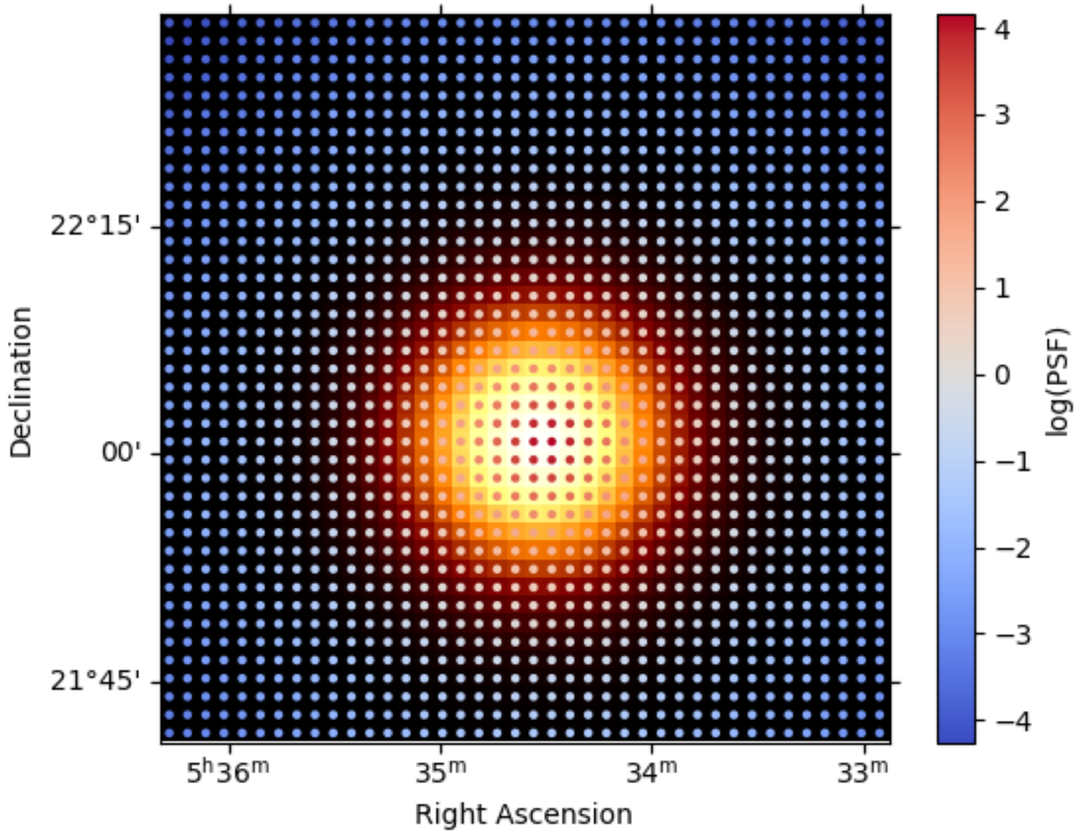


Approach #2: True geometry tailored to the source model

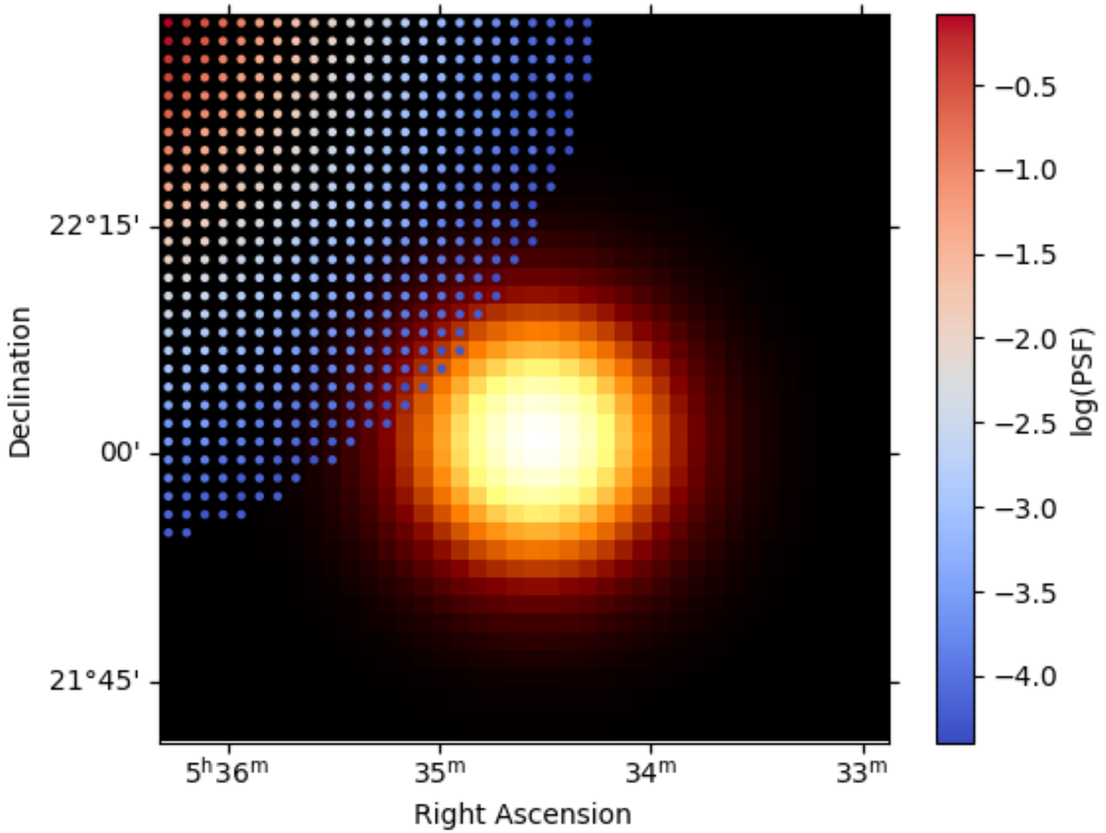


PSF example for 2 different events and for a small source

Event very close to the source

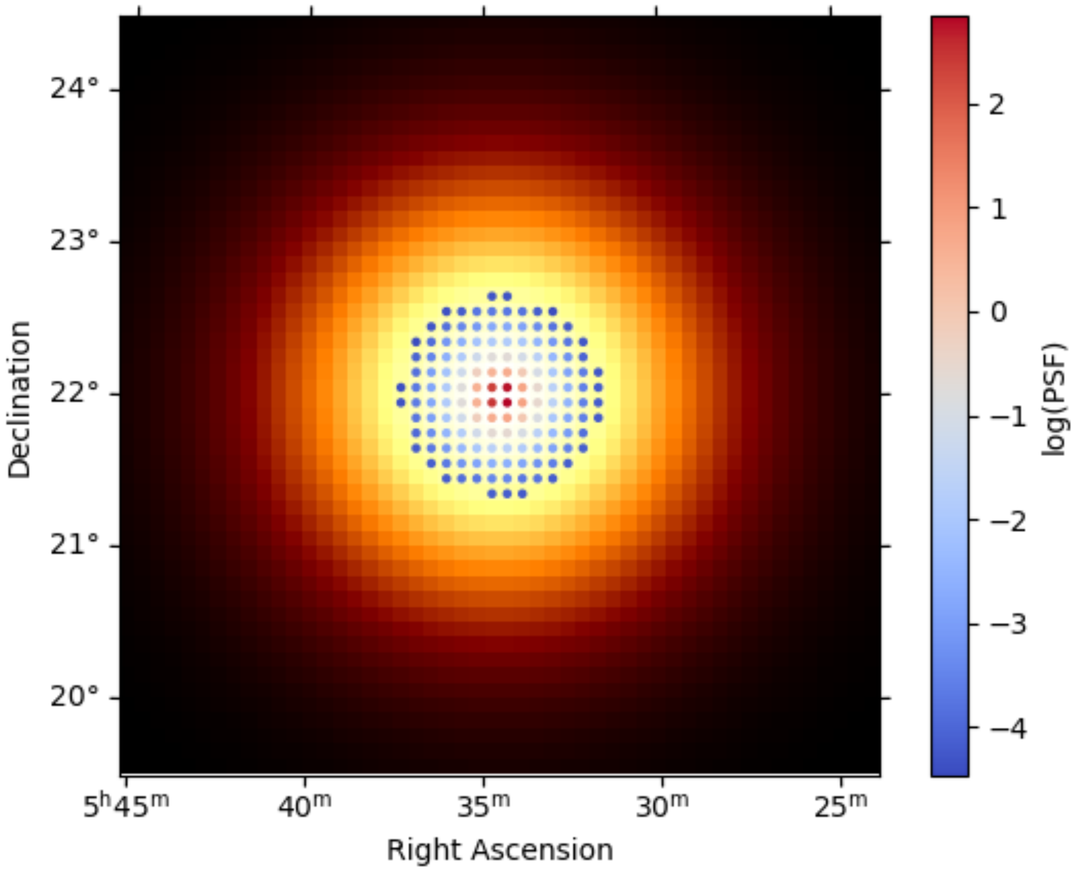


Event far away from the source

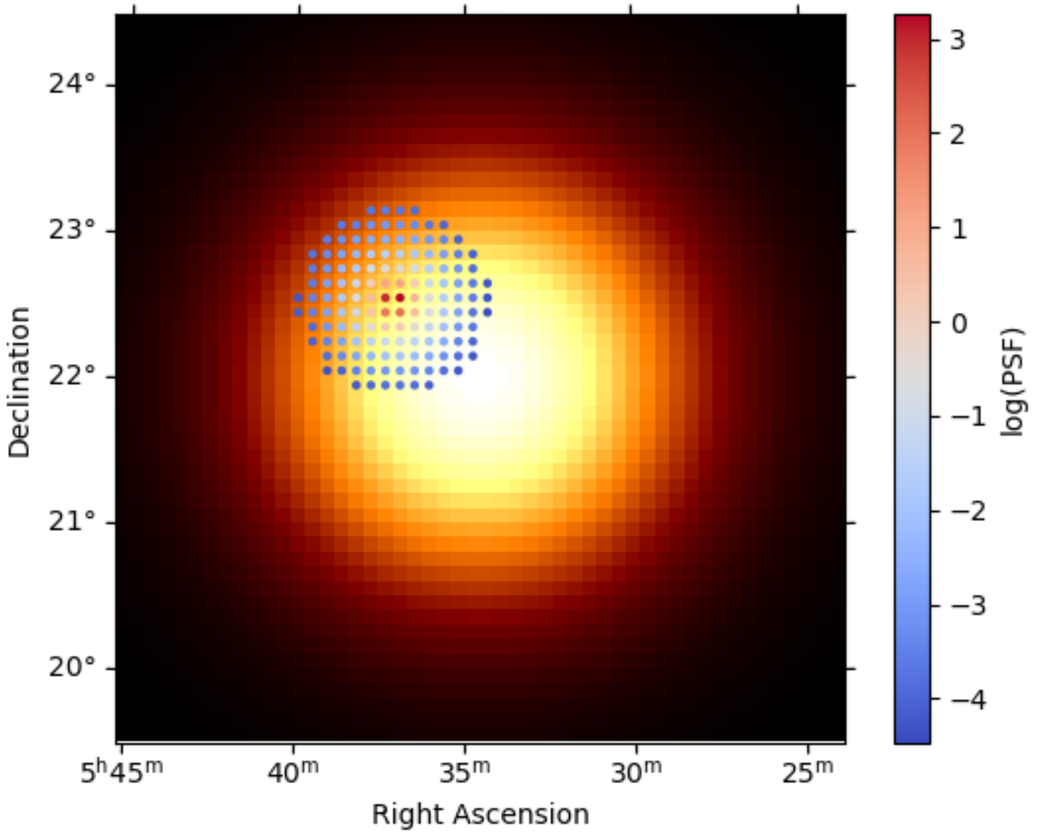


PSF example for 2 different events and for a big source

Event very close to the source



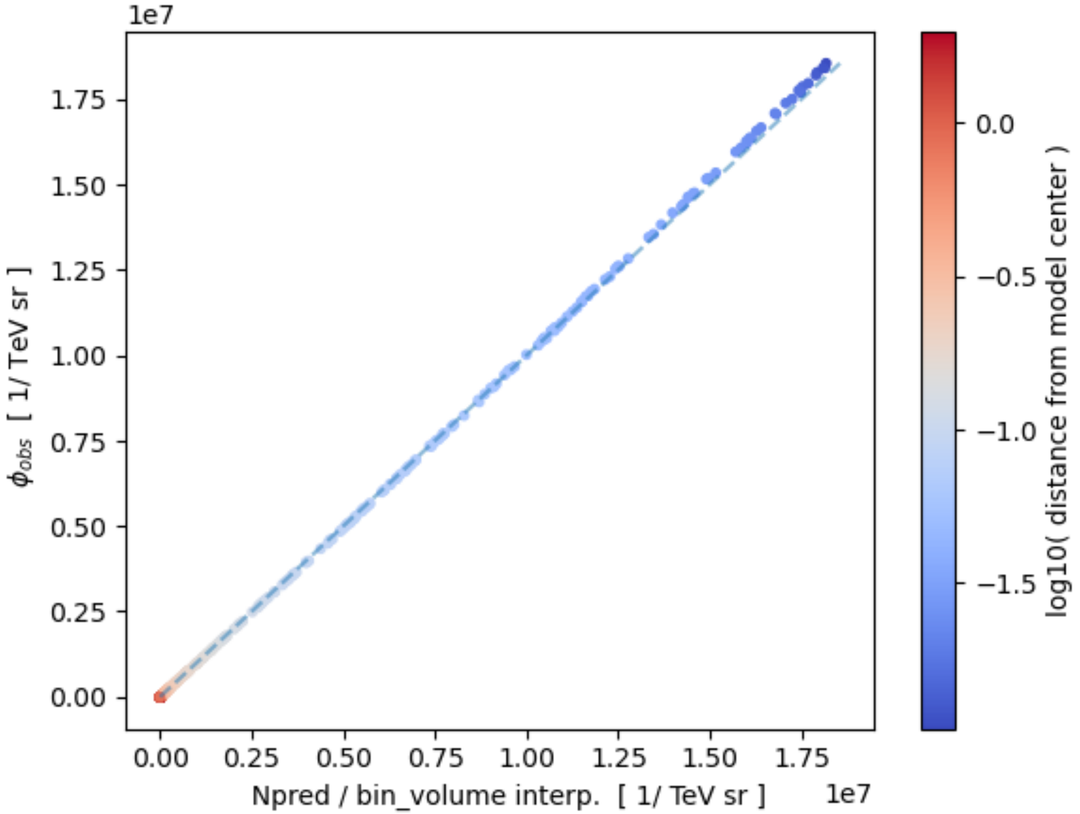
Event far away from the source



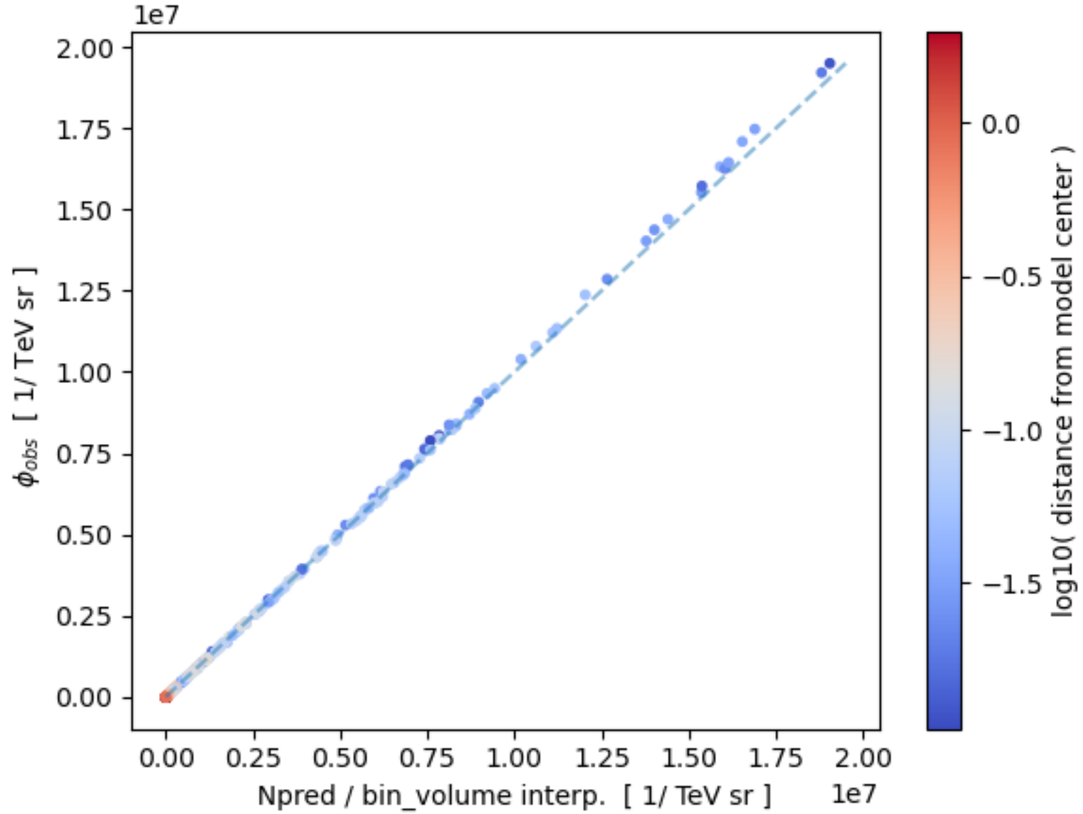
```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",          lat_0     = "-5.784 deg", sigma = '0.01 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

Sigma = 0.01 deg.

Ignoring the event energy (fixed to 1 TeV for all events)




```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```

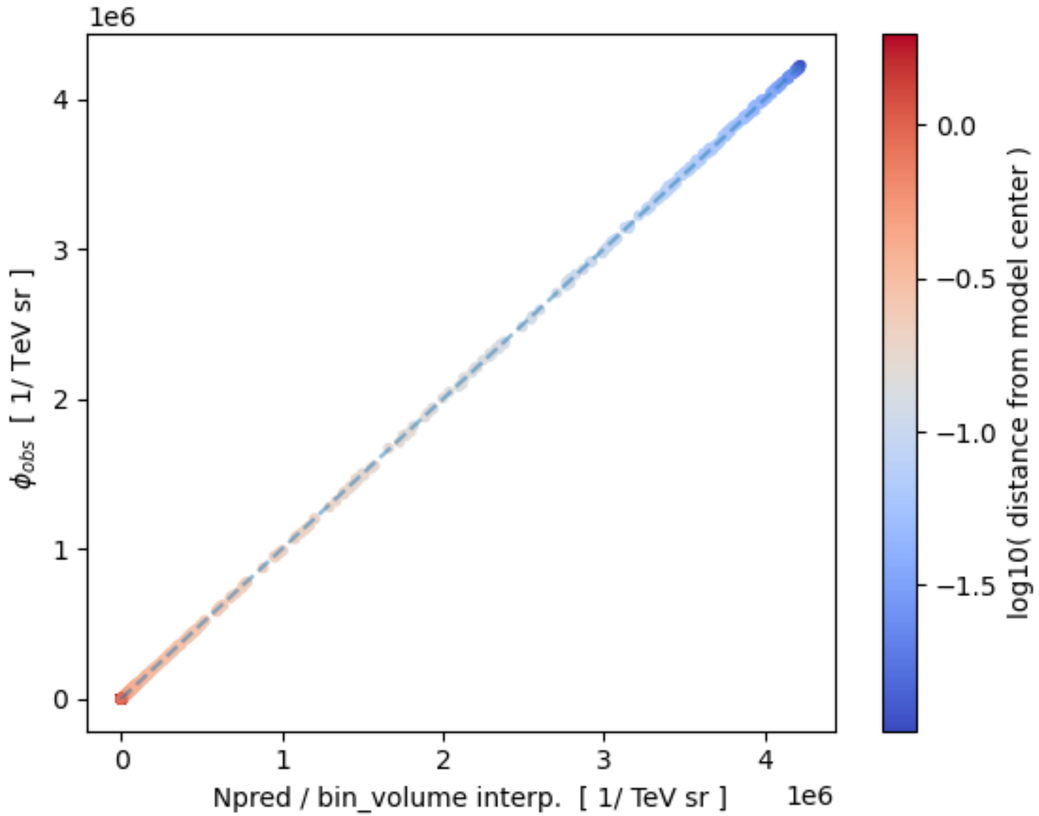


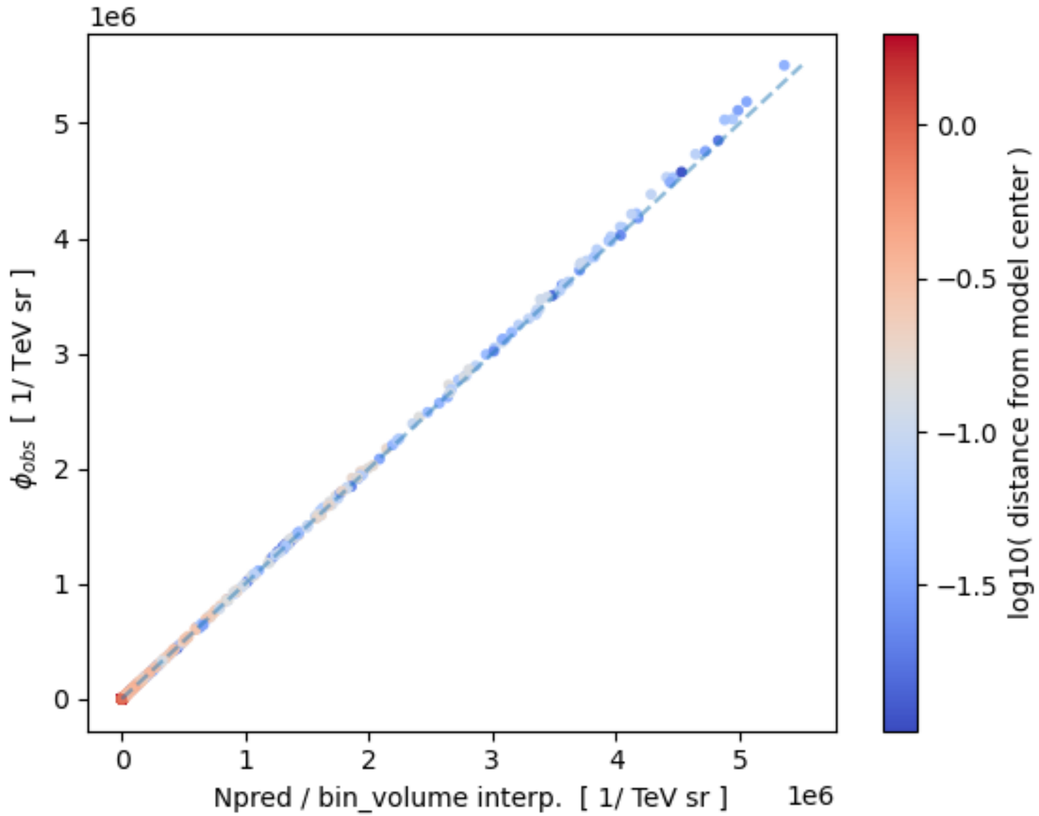

```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```

```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",           lat_0     = "-5.784 deg", sigma = '0.1 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

Sigma = 0.1 deg.

Ignoring the event energy (fixed to 1 TeV for all events)





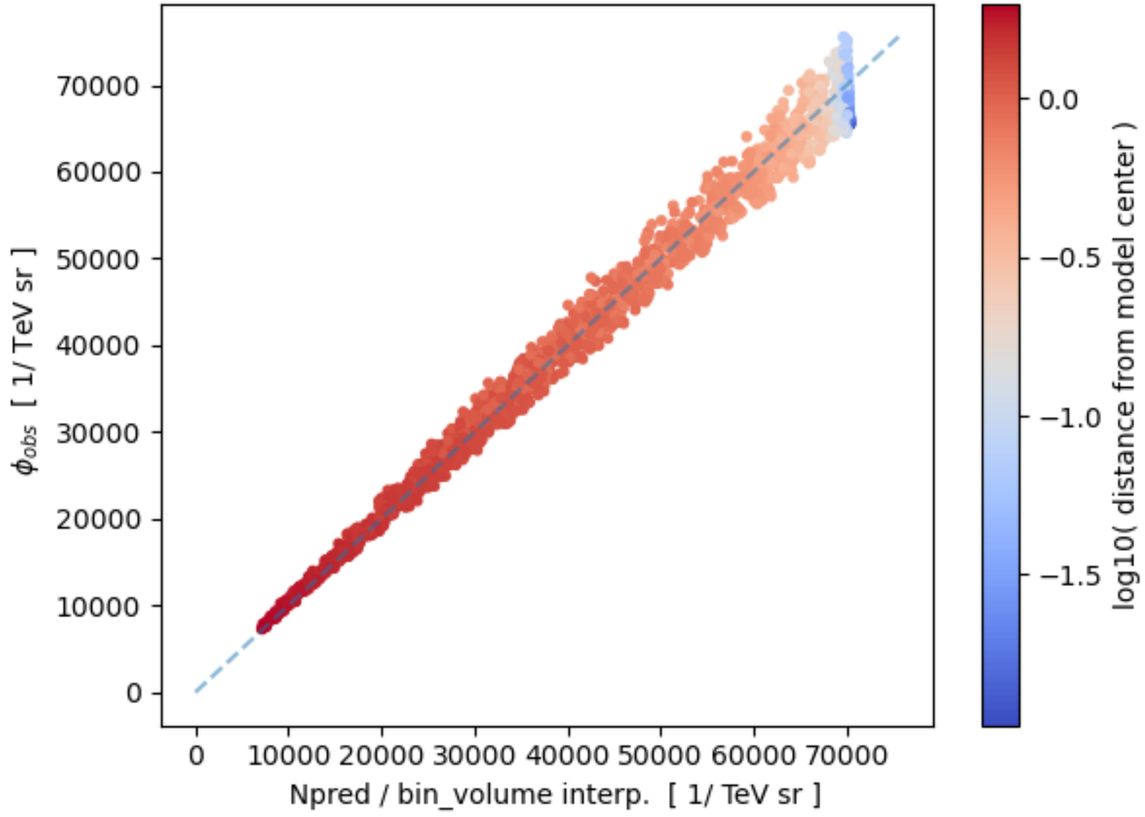

```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```

```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```

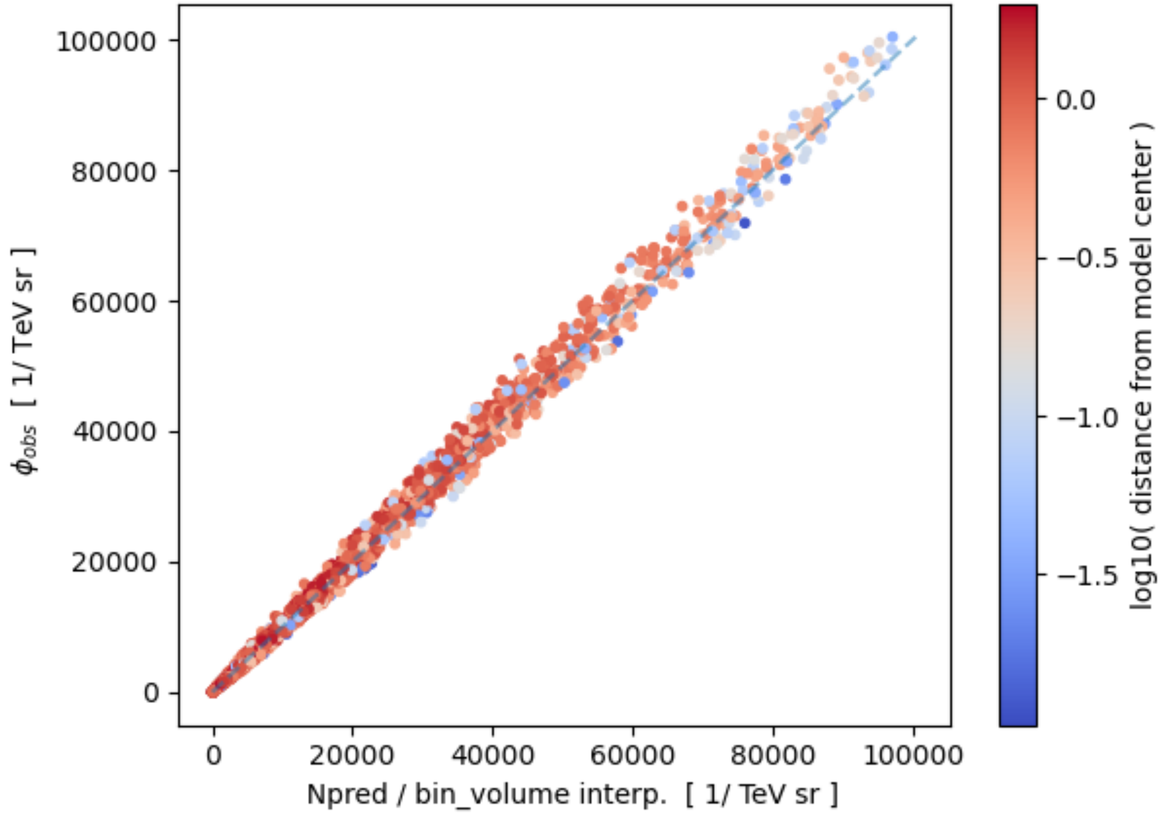
```
model_gauss = SkyModel(
    spatial_model  = GaussianSpatialModel(     lon_0     = "184.557 deg",           lat_0     = "-5.784 deg", sigma = '1 deg', frame = 'galactic'),
    spectral_model = LogParabolaSpectralModel( amplitude ='3.5e-11 cm-2 s-1 TeV-1', reference = '1 TeV',      alpha = 1.8,       beta  = 0.4),
    name           = 'crab_model_gauss'
)
```

Sigma = 1 deg.

Ignoring the event energy (fixed to 1 TeV for all events)




```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```




```
mapnpred        = dataset.npred()
mapnpred2       = mapnpred.copy()
mapnpred2.data  = mapnpred.data/ mapnpred.geom.bin_volume().to( "TeV sr" ).value
flux_interp     = mapnpred2.interp_by_coord( events.map_coord( mapnpred2.geom ))/ u.Unit( "TeV sr" )
```