Maps and Datasets

Atreyee Sinha

Datasets - What are they?

- New API for modelling and fitting; in progress after release of v0.10

- To enable joint likelihood fitting (of different types of `datasets` - eg: counts maps at TeV, flux points at Radio and counts spectra in X-rays)

- Separate *MapDataset, FluxPointDataset, SpectrumDataset*

- Each *dataset*

  - *bundles reduced data with a parameteric model and fit statistics function*

  - *evaluates the model and log-likelihood*

  - *passes it on to the fit object*

- See PIG 8 (#1986) for details

```
from gammapy.cube import MapDataset

model = SourceModel.read("model.yaml")
background = BackgroundModel.read("stacked/background.fits")

counts = Map.read("stacked/counts.fits")
exposure = Map.read("stacked/exposure.fits")
edisp = Map.read("stacked/edisp.fits")
psf = Map.read("stacked/psf.fits")
```

```
dataset = MapDataset(
    counts=counts,
    exposure=exposure,
    edisp=edisp,
    psf=psf,
    mask=mask,
    model=model,
    background=background_model,
    likelihood="cash"
    )

fit = Fit(dataset)
fit.optimize()
```

Joint fitting

- Through the *Datasets* class
  - add up the log-likelihood values of the individual datasets
  - join the parameter lists from each dataset
  - Main user interface to the fit class

```
dataset_1 = MapDataset(
    model=sky_model,
    counts=counts_map_1,
    exposure=exposure_map,
    mask=mask_map,
    psf=psf_map,
    edisp=edisp_map,
    background_model=background_model_1,
)

dataset_2 = MapDataset(
    model=sky_model,
    counts=counts_map_2,
    exposure=exposure_map,
    mask=mask_map,
    psf=psf_map,
    edisp=edisp_map,
    background_model=background_model_2,
)

background_model_1.parameters["norm"].value = 0.4
background_model_2.parameters["norm"].value = 0.9

fit = Fit([dataset_1, dataset_2])
result = fit.run()
```

Already Implemented

- FluxPointDataset class (#2023)
- MapDataset class (#2026)
- Datasets class (#2030)
- SpectrumDataset (#2047)
- EDispMap class (#2031)

What we need

- IrfMapMaker (#1970)
  - Create PSF and Edisp Maps
  - (thus, also exposure)
  - Can have a different *geom* than the counts
  - Have max field of view offset cut?
- Add support for energy dispersion maps to MapDataset
- Implement MapDataset.setup() method
- Add tutorial for joint-likelihood fitting of multiple observations

Implementation of ring background

- Clean up implementation with gammapy
  - *ring_background_estimate()* in *gammapy.background*
    - chooses events between *CircleSkyRegion* and *events.select_sky_ring*, divide by ring_area. No aaceptance correction
    - No acceptance correction, no exclusion region
    - Propose to rename it ?
    - *RingBackgroundEstimator*
      - *3 functions with one line computations*
      - *No exclusion region in on_region*
      - *No debug plots*
      - *No smoothing radius*
  - *AdaptiveRingBackgroundEstimator* - proper implementation
- Add an *ImageMaker* specially for 2D images with ring background estimation (#1850)