Event Sampling & Simulation

Axel Donath & Fabio Pintore

Coding Sprint, Granada, Nov. 18th 2019

Status

- Work plan described in PIG 9, implementation in progress (Fabio Pintore)
- We have `MapDataset.fake()`, `SpectrumDataset.fake()` and `SpectrumDatasetOnOff.fake()` methods
- The following building blocks are implemented:
  - `InverseCDFSampler` in `gammapy.utils.random`
  - `Map.sample_coord()`
  - `EdispMap.sample_coord()`
  - `PSFMap.sample_coord()`
  - `ConstantTemporalModel.sample_time()`
  - `PhaseCurveTemplateTemporalModel.sample_time()`
  - `LightCurveTemplateTemporalModel.sample_time()`

Missing Work

- An in memory `Observation` object to represent simulated CTA IRFs (see also GH 2527)

```
obs = Observation(id=, pointing=, livetime=, bkg=, edisp=, aeff=,psf= )
```

```
maker = MapDatasetMaker()

dataset = maker.run(obs, steps=["bkg", "edisp", "exposure", "psf"])

dataset.model = ...
dataset.fake()

# or later once we have event sampling
events = dataset.fake_events()
events.write(obs.location(hdu_type="events", hdu_class="events"))
```

Missing Work

- Higher level API, that brings all components together and creates an `EventList` object. Implemented as `MapDatasetEventSampler()` and / or `MapDataset.fake_events()`,
- Improve implementation of `PhaseCurveTemplateTemporalModel.sample_time()`
- Testing, testing, testing, …
- Add simulation validation to gammapy-benchmarks
- Add tutorials / documentation (generally included in PIG 18)