Gammapy Modeling and Fitting

Axel Donath

Gammapy Coding Sprint in Madrid
Monday, October 1st

Introduction

- For a long time we have discussed which modeling and fitting library to use for Gammapy: astropy.modeling, sherpa or maybe another?
- Now we have decided to implement the modeling part and likelihood evaluation completely in Gammapy and create an interface to the most common optimizer libraries, such as sherpa, iminuit, scipy.optimize or emcee
- This gives us full flexibility in future (e.g. to add further fitting backends) but we still have complete control over the modeling part, which is specific to Gamma-ray astronomy.

Past work and Status

- Currently we have three fit classes in Gammapy: `MapFit`, `SpectrumFit` and `FluxPointFit`, others might be added in future.
- In #1785 and #1789 I introduced a `Fit` base class, which defines a common user interface to various fitting backends such as sherpa and iminuit.
- I also introduced a `FitResult` object, which stores the best fit model, as well as additional info such as success, number of function evaluations, fit statistics value at minimum, likelihood profiles, etc.

Suggestions for this week

- Fully support sherpa backend:
  - Add `LevMar` optimizer
  - Add handling of freezed parameters
  - Add support for error estimators `Conf` and `Covar`
- Add further fit utility methods:
  - Likelihood contours
  - Improve interactive interface (e.g. add a nice `FitResult` representation for notebooks)
- Add support for further backends such as emcee and scipy.optimize