Implementation of joint X-ray and γ-ray fitting with Gammapy

L. Giunti★, R.Terrier

giunti@apc.in2p3.fr

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

SciPy astropy NumPy GitHub

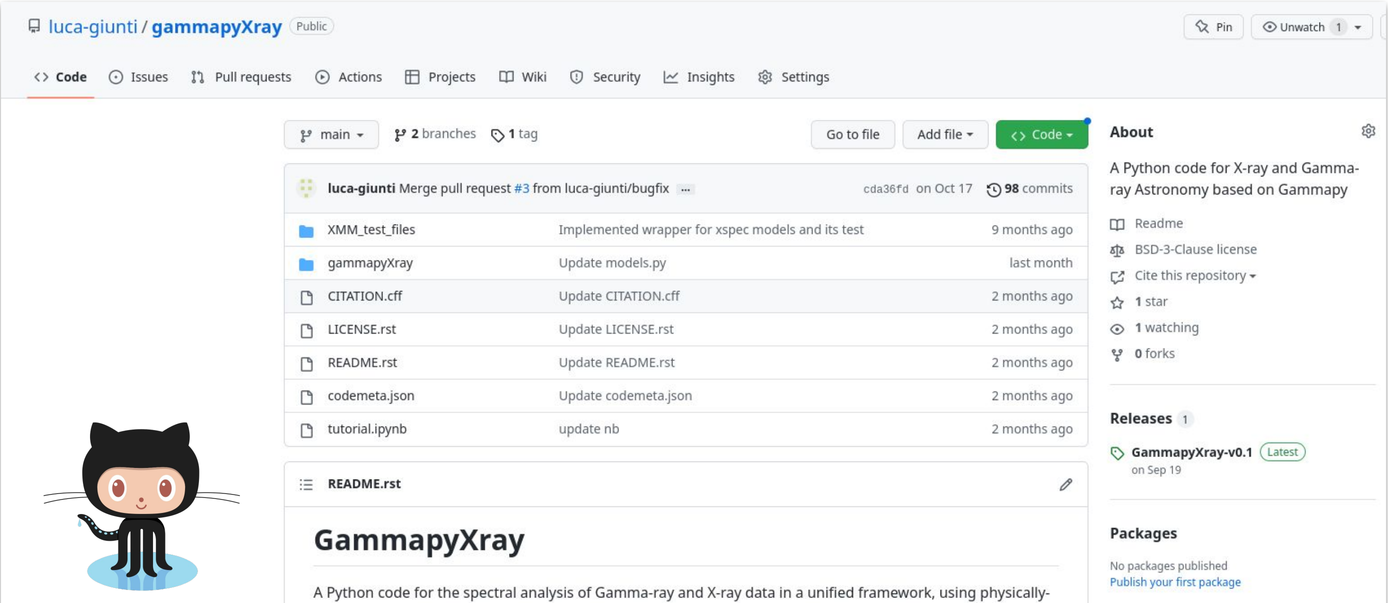
https://docs.gammapy.org/1.0/

GammapyXray

- A prototype implementation of a MWL modeling framework for X-ray and γ-ray data
- Based on:
  - **Gammapy** (data handling + fitting)
  - **Sherpa** (Xspec models library)
  - **Naima** (Radiative models library)

https://github.com/luca-giunti/gammapyXray

https://zenodo.org/record/7092736



API

X-ray Data handling : `StandardOgipDataset`

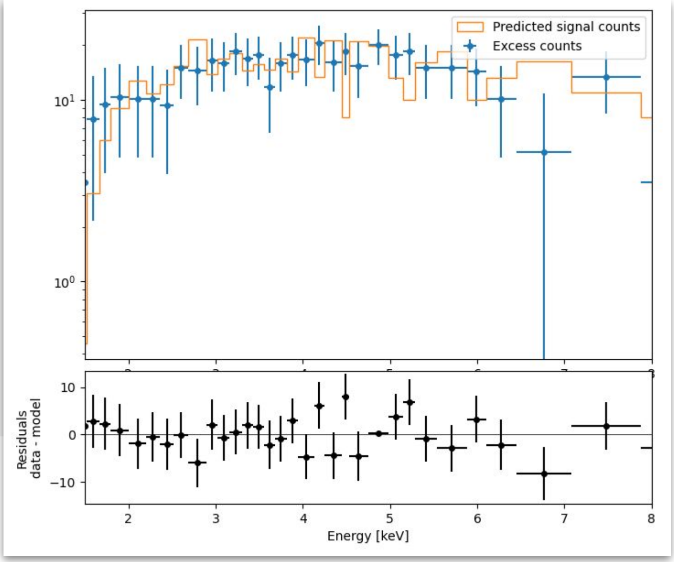
- A special type of `gammapy.datasets.SpectrumDataset`
- `I/O`: reads standard OGIP files (pha)
- Handles X-ray specific features
  E.g. for **grouping** uses: `Dataset.resample_energy_axis(...)`

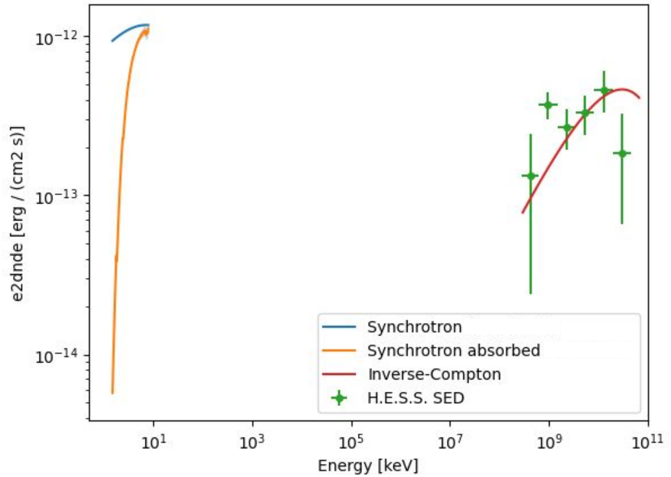
X-ray models interface: `SherpaSpectralModel`

- A wrapper for Sherpa (also **Xspec**) models
- Adapts the model parameters for fitting within the Gammapy framework

Workflow

```
# read data
dataset = StandardOGIPDataset.read(pha)

# define model
pl = SherpaSpectralModel(sherpa.models.PowLaw1D())
absorption = SherpaSpectralModel(sherpa.astro.xspec.XSwabs())
model = pl * absorption

# assign model to the dataset
dataset.models = [model]

# fit
fit = Fit()
fit_result = fit.run(dataset)

# inspect residuals
dataset.plot_fit()
```





Allows joint fit of **X-ray** and **γ-ray** data

```
datasets = Datasets([swift_ds, hess_ds])
fit.run(dataset)
```

Can combine **Xspec** and **Naima** models

```
electrons = naima.models.ExponentialCutoffPowerLaw()
synchrotron = naima.radiative.Synchrotron(electrons)
naima_model = NaimaSpectralModel(synchrotron)

absorption = SherpaSpectralModel(sherpa.astro.xspec.XSwabs())

absorbed_synchrotron_model = absorption * naima_model
```

First usage example with code description

Astronomy & Astrophysics   All volumes   For authors   Search   Menu

Home ▸ All issues ▸ Volume 667 (November 2022) ▸ A&A, 667 (2022) A130 ▸ Full HTML

Open Access

Issue	A&A **Volume** 667, November 2022
Article Number	A130
Number of page(s)	13
Section	Astrophysical processes
DOI	https://doi.org/10.1051/0004-6361/202244696
Published online	17 November 2022

A&A 667, A130 (2022)

Constraining leptonic emission scenarios for the PeVatron candidate HESS J1702−420 with deep *XMM-Newton* observations*

L. Giunti[1], F. Acero[2], B. Khélifi[1], K. Kosack[2], A. Lemière[1] and R. Terrier[1]

+

Received: 5 August 2022 | **Accepted:** 15 September 2022

Abstract

Aims. We aim to search for a hidden leptonic accelerator, such as a high-$\dot{E}$ pulsar, associated with the

All volumes
Special issues
Forthcoming articles
Press releases

Table of Contents

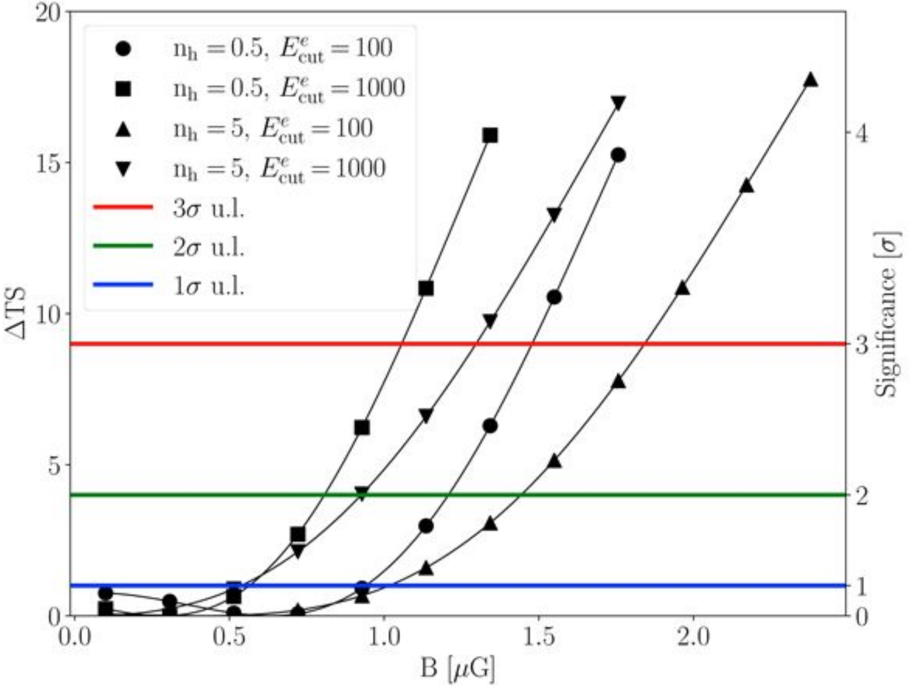
Article

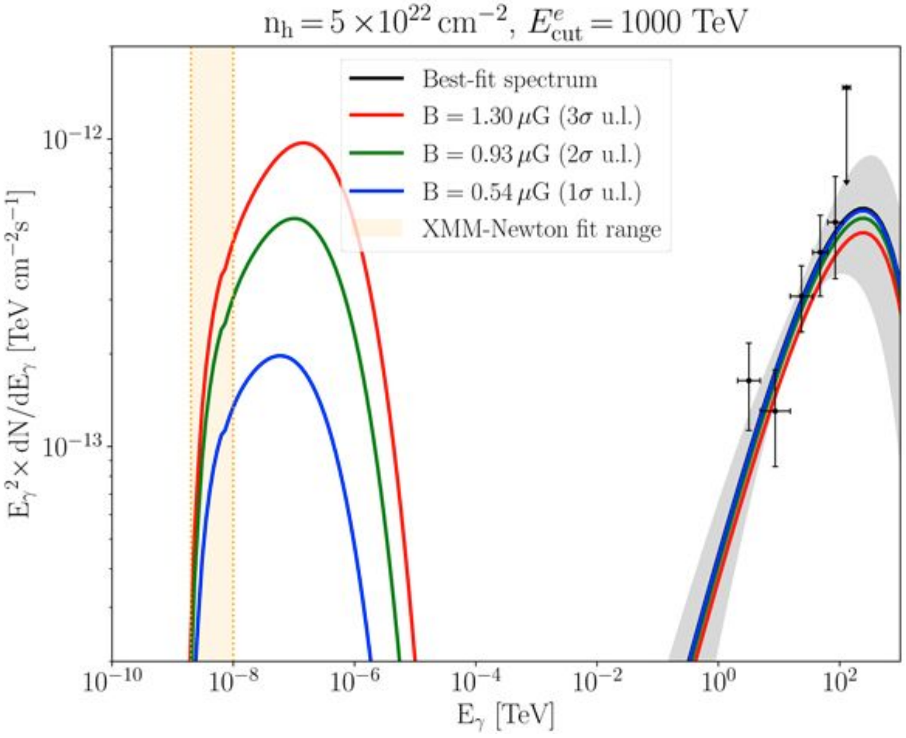
Abstract | Full HTML | PDF (2.230 MB) | ePUB (6.901 MB) | References | Tables at CDS | NASA ADS Abstract Service

Metrics

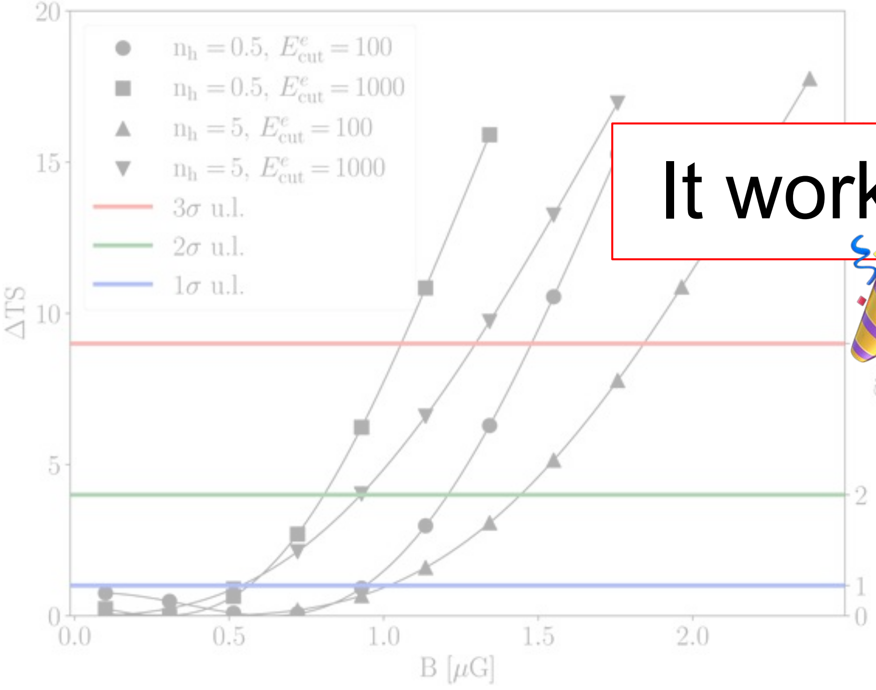
Show article metrics

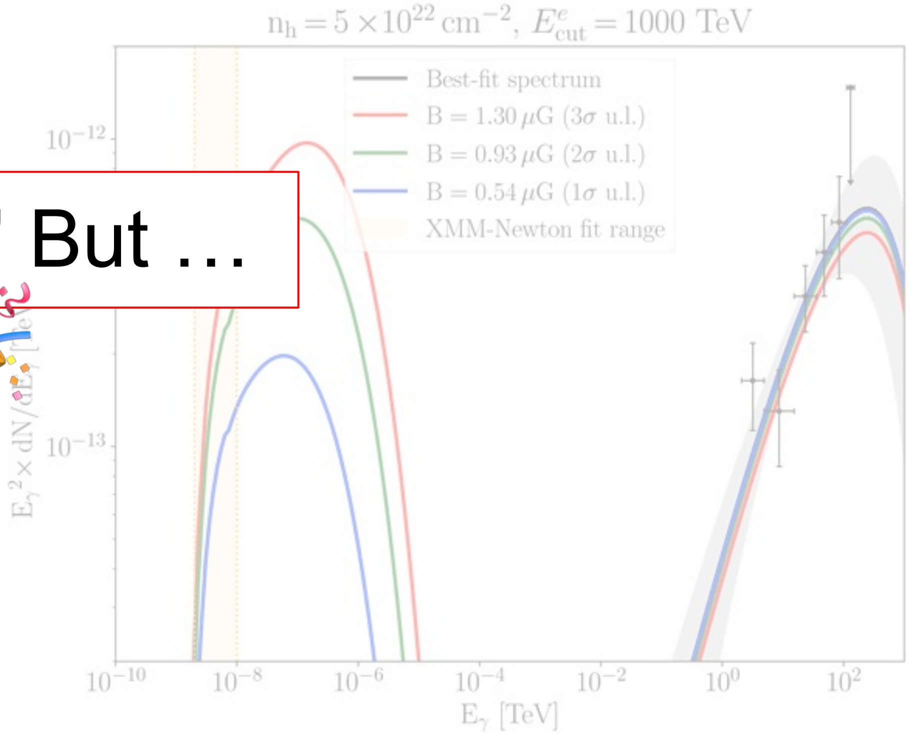
Joint fit of XMM-Newton and H.E.S.S. data to constrain the magnetic field value (one-zone leptonic hypothesis)





Joint fit of XMM-Newton and H.E.S.S. data to constrain the magnetic field value (one-zone leptonic hypothesis)





It works! But …

Limitations

- The current grouping logic creates various issues, e.g.:
  - For flux points estimation
  - For models assignment to specific datasets

- Complex implementation scheme
  - `I/O`: 280 code lines
  - `StandardOgipDataset`: 240 code lines
  - `SherpaSpectralModel`: 70 code lines

- Requires workarounds to adapt the Sherpa models evaluation scheme (w/wo integration) for Gammapy fitting

- Re-invents the wheel: most I/O functionalities are already existing in Sherpa

- It limits the fit to Cash or Wstat statistics, while others might be needed (e.g. chi2, leastsq, …)

Possibilities for a cleaner implementation

- Re-implement the `StandardOgipDataset` based on the Sherpa low-level API for:
    - `I/O`
    - Statistic

```
data = sherpa.astro.ui.unpack_pha(pha)
stat = sherpa.stat.Chi2()
dataset = StandardOgipDataset(data, stat)
```

✔ Avoids code duplication with Sherpa and allows more freedom in the fit statistic choice

- Model evaluation and likelihood calculation handled directly by Sherpa

`StandardOgipDataset.stat_sum()` calls `sherpa.stat.calc_stat(data, model)`

✔ Avoids issues due to the difference between the Gammapy and Sherpa model evaluation schemes

- A working prototype already exists. Only 200 lines of code → Easier to eventually merge into Gammapy

Sherpa

```python
filename = "XMM_test_files/MOS1_PWN.grp"
ui.load_pha(1, filename)
```

```python
ui.notice(2, 8)
```

```python
ui.set_stat("wstat")
```

```python
ui.set_source(ui.xstbabs.absorption * ui.powlaw1d.pl)

absorption.nh = 5
pl.gamma = 1.8
pl.ampl = 4e-4
```

```python
ui.guess(pl)
```

```python
ui.fit()
result_sherpa = ui.get_fit_results()
```

Gammapy

```python
data = unpack_pha("XMM_test_files/MOS1_PWN.grp")
```

```python
stat = WStat()
```

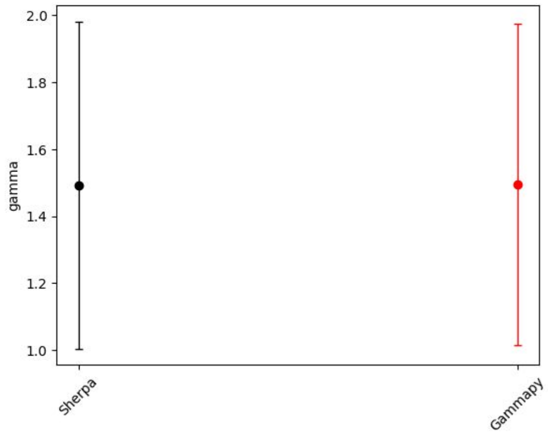
```python
dataset = SherpaSpectrumDataset(data, stat)
```

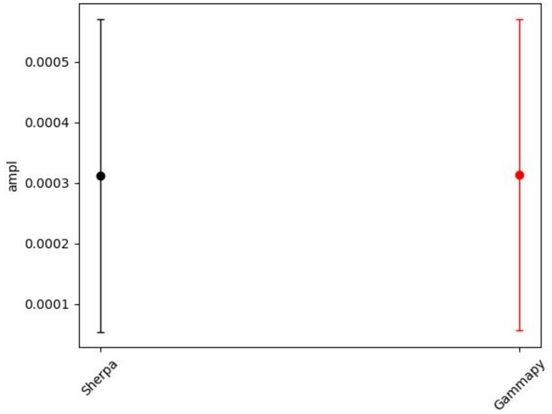
```python
dataset.notice(2, 8, ignore=False)
```

```python
pl = PowLaw1D()
pl.gamma.val = 1.8
pl.ampl.val = 4e-4
absorption = XSTBabs()
absorption.nH = 5
model = pl*absorption

wrapped_model = SherpaSpectralModel(model)
```

```python
dataset.models = [wrapped_model]
```

```python
fit = Fit()
result_gammapy = fit.run([dataset])
```





Summary and perspectives

- GammapyXray is a prototype code that allows to jointly model X-ray and γ-ray data based on Gammapy, Sherpa and Naima
- It is completely open source: https://github.com/luca-giunti/gammapyXray
- Despite some limitations, it works and has been already used to publish scientific results on real data: Giunti et al. 2022
- Alternative (cleaner) implementations are already under study
- Open questions:
  - Should this be a standalone tool, or be merged into Gammapy?
  - Can we extend this logic to include a full 3D analysis of X-ray data? (Fabio is looking into that)
- Stay tuned!