From light curves to power spectra: unveiling time-domain behavior with gammapy

C. Galelli[1], C.Boisson[1], B.Khelifi[2] and R.Terrier[2]

[1]Laboratoire Univers et Théories, Observatoire de Paris, Université PSL, Université Paris Cité, CNRS, F92190 Meudon, France

[2]Université Paris Cité, CNRS, Astroparticule et Cosmologie, F75013 Paris, France

Core dependencies: matplotlib, astropy, NumPy, SciPy

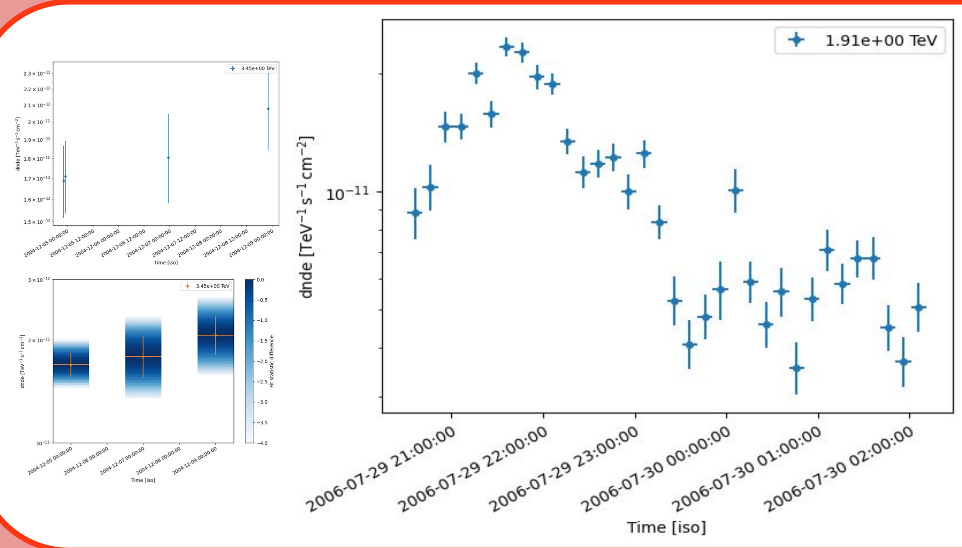
A **Python** package for **gamma-ray** astronomy

The core library for the analysis tools of CTAO

[1][2][3]

- Event lists and IRFs from experiments in a common format
- Structured in two steps: **Data reduction** and **modeling/fitting**
- Offers spectral, spatial, **time-domain analysis** and a library of associated models

The time domain in Gammapy

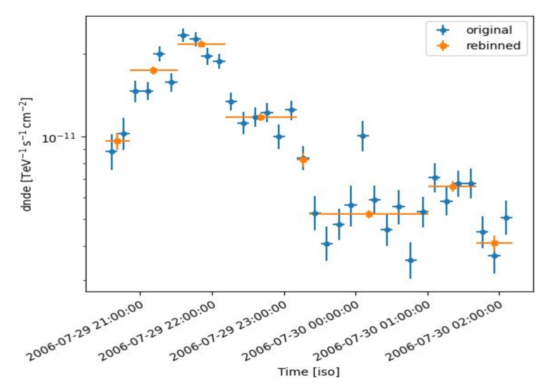


The base block: Building light curves

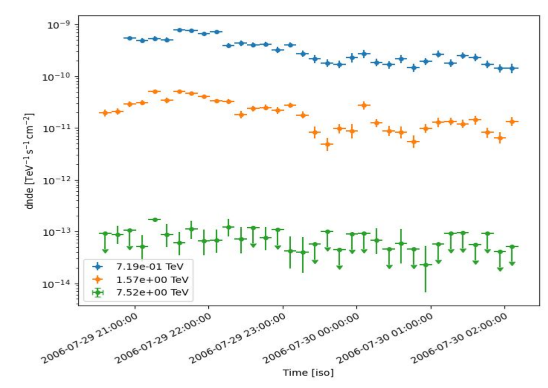
In gammapy lightcurves are `FluxPoints` objects built with the specialized `LightCurveEstimator`

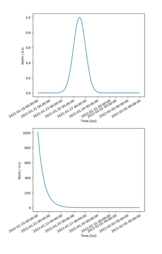
Tutorials: creating lightcurves + lightcurves for flares

Lightcurves can be rebinned according to requirements e.g. minimum TS or flux, or using algorithm such as Bayesian blocks [4]



Multiple energy bins are supported to highlight difference in behavior in different energy bands





Gammapy provides a library of temporal models to be used for fitting and simulation

Tutorial: simulating and fitting a lightcurve

Utility functions implement estimators of time variability
- Fractional [5] and point-to-point variability [6]
- Doubling/halving time
- Structure Functions [8]

Tutorial: estimating variability in a lightcurve

From time to frequencies

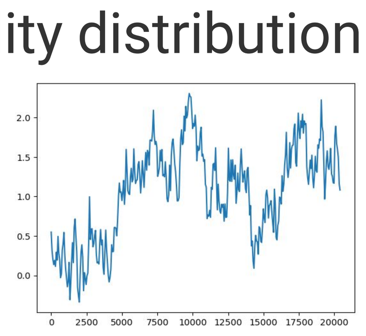
To better model time-variable sources it is useful to study the frequency domain. An important passage is the simulation of synthetic lightcurves according to a power spectral density model. The simulated curves can be used to predict behavior of data, test algorithms or fit observations

A new function in mainline Gammapy

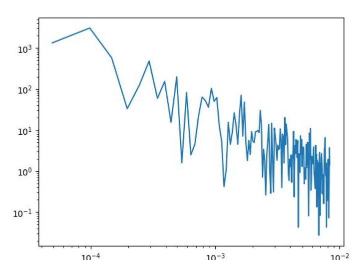
A new Gammapy recipe: Fitting the PSD of behavior of a Lightcurve

Simulation: the TK95 algorithm

Simulation algorithm proposed by Timmer & Koenig, 1995 [8] Based on FFT and a gaussian probability distribution of data.



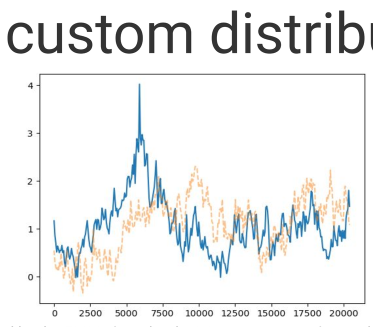
Synthetic lightcurve created by the TK95 algorithm by injecting a power-law with index -1.4



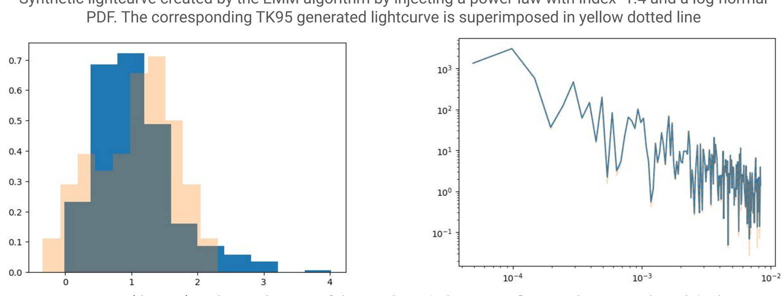
Periodogram of the synthetic lightcurve reflecting the injected model

Simulation: the Emmanoulopoulos algorithm

Extension of the TK95 proposed by Emmanoulopoulos, 2012 [9] Introduces the possibility of using custom distributions.



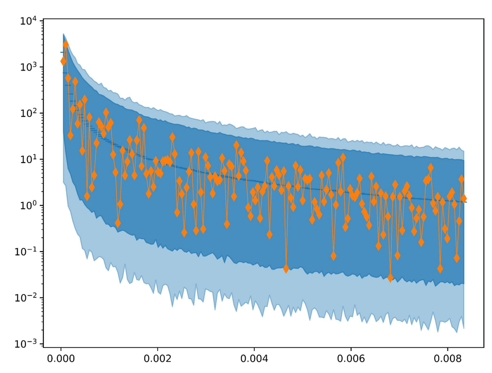
Synthetic lightcurve created by the EMM algorithm by injecting a power-law with index -1.4 and a log-normal PDF. The corresponding TK95 generated lightcurve is superimposed in yellow dotted line



Histogram (density) and periodogram of the synthetic lightcurve reflecting the injected model. The corresponding quantities from the TK95 curves are in yellow. It is easy to notice that the PDFs are different, showing a gaussian distribution in the TK95 and the lognormal for the EMM; the periodograms are identical, as both come from a -1.4 index power-law.

PSD envelopes

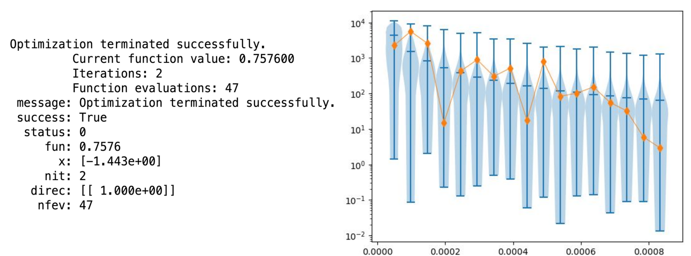
Envelopes can be built by simulating a large number of LCs according to a PSD model and extracting the periodogram statistical distribution.



Envelope of the EMM synthetic lightcurve obtained by simulating 10000 analogous curves and storing the statistical distributions of values at each frequency. The 95% and 99% quantiles, and the median are shown.

A PSD fitting algorithm for observed lightcurves

The observed LC periodogram is compared to the envelopes through a $\chi 2$ estimator. The best-fit PSD model is obtained by minimizing with `scipy` the number of realizations in the envelope that have a lower $\chi 2$ score than data.



Result of the fit for the EMM lightcurve after convolution with IRFs using Gammapy. The algorithm reconstructs well the spectral index especially using oversampling in the envelope to account for red noise leakage. The right is a plot of the reconstructed periodogram superimposed over the best-fit envelope

Future and conclusions

New functionalities and utilities will be continually added to Gammapy according to interest from users and suggestions.

A restructuring or splintering of the `FluxPoints` class is in the works to create a specialized `LightCurve` class for Gammapy

Gammapy is continually adding to its capabilities in the time domain by adding functionalities and utilities, improving and specializing its classes. Strong focus on the time domain is one of the key points of the CTAO project.

Resources

[1]A. Donath et al., 2023, Gammapy: A Python package for gamma-ray astronomy, A&A, Forthcoming article arXiv:2308.13584 [2]https://github.com/gammapy/ [3]https://gammapy.org/ [4] Scargle, J et al., 2013, Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations https://ui.adsabs.harvard.edu/abs/2013ApJ...764..167S [5] Vaughan, S et al., 2003, On characterizing the variability properties of X-ray lightcurves from active galaxies, https://ui.adsabs.harvard.edu/abs/2003MNRAS.345.1271V [6] Edelson, R. et al., 2002, X-Ray Spectral Variability and Rapid Variability of the Soft X-Ray Spectrum Seyfert 1 Galaxies Arakelian 564 and Ton S180, https://iopscience.iop.org/article/10.1086/323779 [7] Emmanoulopoulos, D. et al., 2010, On the use of structure functions to study blazar variability: caveats and problems, https://academic.oup.com/mnras/article/404/2/931/968488 [8] Timmer, J and Koenig, M, 1995, On generating power law noise, https://ui.adsabs.harvard.edu/abs/1995A%26A...300..707T/abstract [9] Emmanoulopoulos, D. et al., 2012, Generating artificial light curves: revisited and updated, https://academic.oup.com/mnras/article/433/2/907/1746942?login=true