



**HAL**  
open science

# GammaLib: A New Framework for the Analysis of Astronomical Gamma-Ray Data

Jürgen Knödlseeder

► **To cite this version:**

Jürgen Knödlseeder. GammaLib: A New Framework for the Analysis of Astronomical Gamma-Ray Data. Astronomical Data Analysis Software and Systems XXI, Nov 2011, Paris, France. hal-01381970

**HAL Id: hal-01381970**

**<https://hal.science/hal-01381970v1>**

Submitted on 28 Apr 2020

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## **GammaLib - A new framework for the analysis of Astronomical Gamma-Ray Data**

Jürgen Knödlseeder<sup>1,2</sup>

<sup>1</sup>*Université de Toulouse; UPS-OMP; IRAP; Toulouse, France*

<sup>2</sup>*CNRS; IRAP; 9 Av. colonel Roche, BP 44346, F-31028 Toulouse cedex 4, France*

**Abstract.** With the advent of a new generation of telescopes (INTEGRAL, Fermi, H.E.S.S., MAGIC, VERITAS, MILAGRO) and the prospects of planned observatories such as CTA or HAWC, gamma-ray astronomy is becoming an integral part of modern astrophysical research. Analysing gamma-ray data is still a major challenge, and today relies on a large diversity of tools and software frameworks that were specifically developed for each instrument. With the goal of facilitating and unifying the analysis of gamma-ray data, we are currently developing an innovative data analysis toolbox, called the GammaLib, that enables gamma-ray data analysis in an instrument independent way. We will present the basic ideas that are behind the GammaLib, and describe its architecture and usage.

### **1. Introduction**

The last decade has seen important progress in the field of gamma-ray astronomy, thanks to significant improvements in the performance of ground-based and space-based gamma-ray telescopes (Vandenbroucke 2010). Gamma-ray photons are nowadays studied over more than 8 decades in energy, from a few 100 keV up to more than 10 TeV. The technologies used for observing gamma-rays are very diverse, and cover indirect imaging devices, such as coded mask or Compton telescopes, and direct imaging devices, such as pair creation telescopes, and air or water Cherenkov telescopes.

Despite this technical diversity, the high-level data produced by the instruments for scientific analysis have great similarities. Generally, the data are comprised of individual events that are characterised by an arrival time, a direction (either given in celestial coordinates for direct imaging devices or in detector coordinates for indirect imaging devices), and an energy estimate. Standards, such as the FITS data format (Pence et al. 2010) and the OGIP conventions (Corcoran et al. 1995), have been implemented for many space-based instruments (CGRO, INTEGRAL, Fermi), although ground-based telescopes (H.E.S.S., MAGIC, VERITAS, MILAGRO) still use their own collaboration internal data formats. An increasing number of projects use analysis tools inspired by HEASARC's FTOOLS (Pence et al. 1993), and implement the IRAF parameter interface (Valdes 1992), allowing for command line control and scripting. Most (if not all) analysis methods imply at some point the fitting of parametric models, employing maximum likelihood techniques to infer physical parameters and their uncertainties (Cash 1979).

Despite these common features, there exist no common tools yet for the scientific analysis of gamma-ray data. So far, each instrument comes with its proper suite of software tools, which often requires costly development cycles and maintenance efforts, and which puts the burden on the astronomer to learn how to use each of them for his studies. For X-ray astronomy, HEASARC has developed standards (such as XSELECT or XSPEC) that unify the data analysis tasks, making X-ray data more accessible to the astronomical community at large. We propose here to follow a similar approach for gamma-ray astronomy.

## 2. What is GammaLib?

GammaLib has the ambition to provide a unified framework for the high-level analysis of astronomical gamma-ray data. GammaLib, which we currently develop at IRAP (Toulouse, France), is a self-contained, instrument independent, open source, multi-platform C++ library that implements all code required for high-level science analysis of astronomical gamma-ray data.

Self-contained means that GammaLib does not rely on any third-party software, with the only exception of the `cfitsio` library from HEASARC that is used to implement the FITS interface. This makes GammaLib basically independent of any other software package, increasing the maintainability and enhancing the portability of the library. The drawback of this approach is that a lot of code had to be written in a first place to implement basic functionalities in GammaLib, such as model fitting, IRAF and XML parameter interfaces, or supporting WCS and HEALPix coordinate projections.

Instrument independent means that GammaLib potentially supports any gamma-ray astronomy instrument. Large parts of the code treat gamma-ray observations in an abstract representation, and do neither depend on the characteristics of the employed instrument, nor on the particular formats in which data and instrument response functions are delivered. Instrument specific aspects are implemented as isolated and well defined modules that interact with the rest of the library through a common interface. Adding a new instrument to GammaLib consists of implementing a new instrument specific module, which considerably reduces the development costs compared to the full-fledged development of an instrument-specific scientific analysis package. This philosophy also enables the joint analysis of data from different instruments, providing a framework that allows for consistent broad-band spectral fitting or imaging.

Open source means that the GammaLib source code is freely available under the GNU General Public license version 3. The latest source code can be downloaded from <https://sourceforge.net/projects/gammlib/> which also provides bug trackers and mailing lists. Further information and documentation on GammaLib can be found on <http://gammlib.sourceforge.net/>.

Multi-platform means that GammaLib is designed to compile on any POSIX compliant platform. So far, GammaLib has been successfully compiled and tested on Mac OS X, OpenBSD, OpenSolaris (using the `gcc` compiler) and many Linux flavours. We are currently setting up an automated continuous build system based on virtual machine technology to guarantee multi-platform compliance throughout the continuing development process. Pre-packed binary versions of the code will become available soon.

C++ library means that GammaLib makes heavily use of C++ classes. Instrument independency is achieved by using abstract virtual base classes, which are implemented as derived classes in the instrument specific modules.

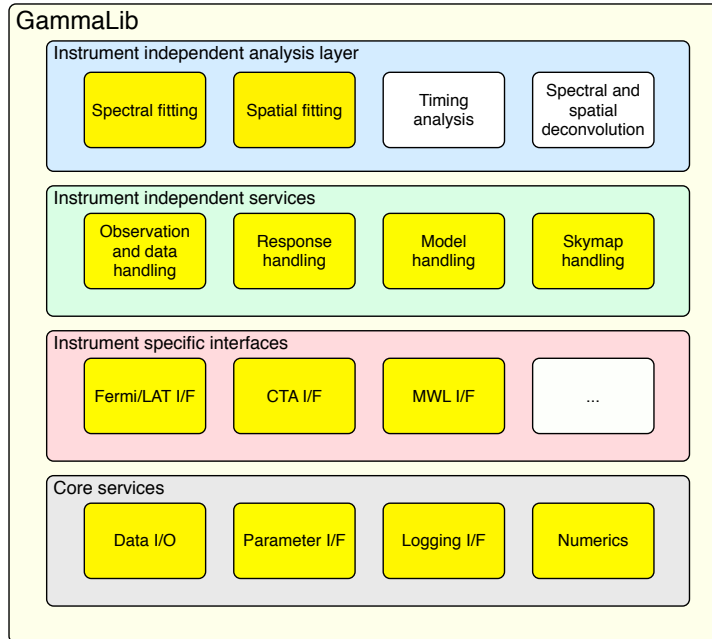


Figure 1. Organisation of GammaLib into software layers and modules. Existing modules are shown in yellow, while planned modules are shown in white.

### 3. How is GammaLib organised?

GammaLib is organised into four software layers, each of which comprises a number of modules (see Figure 1). The top layer provides support for instrument independent high-level data analysis, enabling spectral and spatial fitting by forwards folding of parametric source and background models. Additional modules that are foreseen (but not yet implemented) will deal with timing analysis, spectral and spatial deconvolutions. The second layer provides an instrument independent interface to data products, comprising information related to observations (event data, pointing and exposure information), instrument response functions, parametric source and background models used for model fitting, and sky maps. The latter may be defined using either WCS or HEALPix projections. The modules of the third layer implement interfaces for specific gamma-ray instruments. So far, the analysis of Fermi/LAT data is supported, and a prototype interface for CTA data has been implemented. Furthermore, a general multi-wavelength interface allows adding non-gamma-ray flux points to spectral fits, enabling joint broad-band multi-wavelength fitting of spectral energy distributions. The fourth layer provides core services for GammaLib. They comprise input and output of data (either in FITS format through the cfitsio library (Pence 1999) or in CSV format using a native interface), handling of parameters (either in the IRAF parameter format or in XML format), information logging (either to the console and/or into a dedicated log file), and numerical algorithms and methods needed for data analysis (e.g. optimization of functions, (sparse) matrix and vector handling, solving of linear equations, special mathematical functions, numerical integration and derivatives, interpolation of data, random number generator).

#### 4. How can GammaLib be used?

GammaLib implements an Application Programming Interface (API) that can be used to bind functionalities into specific analysis executables. In particular, GammaLib supports the creation of new FTOOLS, which are implemented by deriving new classes from the `GApplication` base class. A very simple C++ executable that reads source model definitions from a file named "source.xml" and prints them to the console would look like this:

```
#include <iostream>
#include "GammaLib.hpp"
int main(void) {
    GModels models("source.xml");
    std::cout << models << std::endl;
    return 0;
}
```

GammaLib C++ classes are also wrapped into Python using SWIG (see <http://www.swig.org/>), providing a Python module that can be used to create analysis scripts or to perform interactive analysis. The same result as above in Python would be obtained by typing:

```
$ python
>>> from gammalib import *
>>> models = GModels("source.xml")
>>> print models
```

We currently use GammaLib to implement a set of prototype FTOOLS (<http://cta.irap.omp.eu/ctools/>) for the scientific analysis of data from the next generation ground-based Cherenkov Telescope Array (CTA). CTA is in the preparatory phase with an envisioned first light in 2015.

#### References

- Cash, W. 1979, *ApJ*, 228, 939
- Corcoran, M. F., Angelini, L., George, I., McGlynn, T., Mukai, K., Pence, W., & Rots, A. 1995, in *Astronomical Data Analysis Software and Systems IV*, edited by R. A. Shaw, H. E. Payne, & J. J. E. Hayes, vol. 77 of *Astronomical Society of the Pacific Conference Series*, 219
- Pence, W. 1999, in *Astronomical Data Analysis Software and Systems VIII*, edited by D. M. Mehringer, R. L. Plante, & D. A. Roberts, vol. 172 of *Astronomical Society of the Pacific Conference Series*, 487
- Pence, W., Blackburn, J. K., & Greene, E. 1993, in *Astronomical Data Analysis Software and Systems II*, edited by R. J. Hanisch, R. J. V. Brissenden, & J. Barnes, vol. 52 of *Astronomical Society of the Pacific Conference Series*, 541
- Pence, W. D., Chiappetti, L., Page, C. G., Shaw, R. A., & Stobie, E. 2010, *A&A*, 524, A42+
- Valdes, F. 1992, in *Astronomical Data Analysis Software and Systems I*, edited by D. M. Worrall, C. Biemesderfer, & J. Barnes, vol. 25 of *Astronomical Society of the Pacific Conference Series*, 417
- Vandenbroucke, J. 2010, *ArXiv e-prints*. 1012.0849