

GammaLib

une bibliothèque ouverte pour l'analyse de données en astronomie gamma

J. Knödlseder¹, M. Mayer², C. Deil³, F. Forest¹, T. Louge⁴, P. Martin¹, S. Brau-Nogué⁴

¹IRAP, 9 avenue du Colonel-Roche, 31028 Toulouse, France
²Université Humboldt, Under den Linden 6, 10099 Berlin, Allemagne
³Max-Planck-Institut für Kernphysik, Saupfercheckweg 1, 69117 Heidelberg, Allemagne
⁴IRAP, 57 avenue d'Azereix, 65008 Tarbes, France



Grâce au succès des télescopes de rayon gamma actuels (Fermi, H.E.S.S., MAGIC, VERITAS) et en vue des perspectives d'observatoires futurs comme le Cherenkov Telescope Array (CTA), l'astronomie des rayons gamma devient aujourd'hui une partie intégrante de l'astrophysique moderne. Si les données des télescopes actuels ne sont que partiellement publiques aujourd'hui, les observatoires futurs se tournent de plus en plus vers un accès ouvert, nécessitant le développement d'outils publics pour leur analyse.



<http://cta.irap.omp.eu/gammalib>

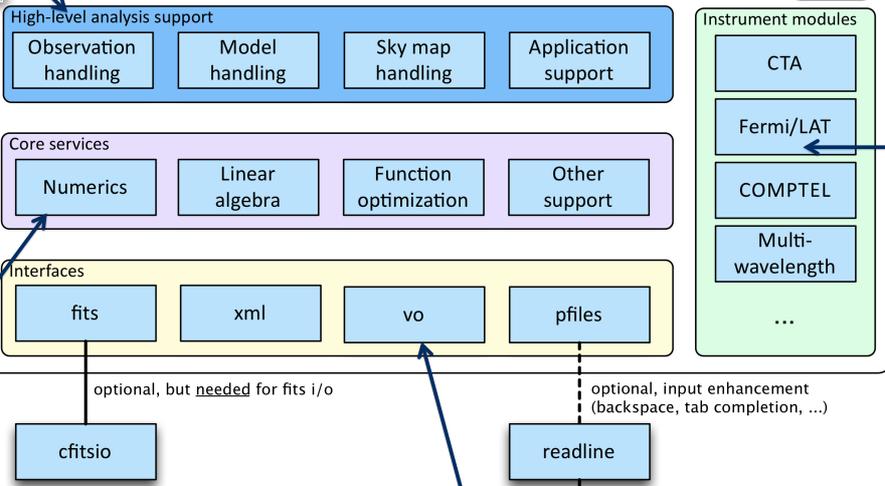


<http://cta.irap.omp.eu/ctools>

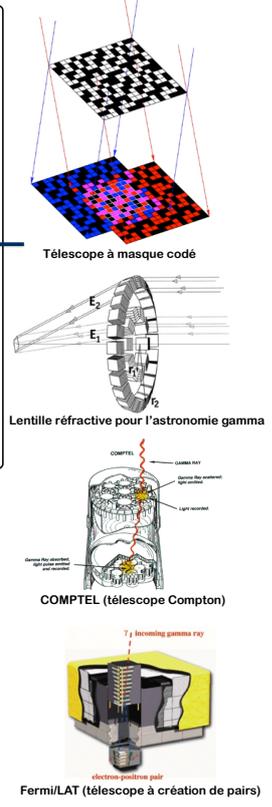
Nous présentons ici la bibliothèque GammaLib que nous avons développée pour répondre à ce besoin. GammaLib est une bibliothèque écrite en C++ qui implémente un cadre générique et toutes les fonctionnalités nécessaires pour effectuer une analyse de données en astronomie gamma. GammaLib est utilisable comme une interface de programmation (API) ou comme un module Python pour réaliser des workflow d'analyse. Un exemple d'utilisation sont les ctools proposés comme outils d'analyse scientifique pour l'observatoire CTA. GammaLib et ctools sont des logiciels communautaires dont les codes sources sont disponibles en libre accès.

Des classes abstraites pour définir des concepts

```
class Observation : public Base {
public:
    Observation() {}
    Observation(const Observation& obs):
        Observation(obs) {}
    virtual ~Observation() {}
    virtual Observation* clone() const { return new Observation(*this); }
    virtual Observation* copy() const { return new Observation(*this); }
    virtual Observation* copy(const Observation& obs) const { return new Observation(obs); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2) const { return new Observation(obs, obs2); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3) const { return new Observation(obs, obs2, obs3); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4) const { return new Observation(obs, obs2, obs3, obs4); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4, const Observation& obs5) const { return new Observation(obs, obs2, obs3, obs4, obs5); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4, const Observation& obs5, const Observation& obs6) const { return new Observation(obs, obs2, obs3, obs4, obs5, obs6); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4, const Observation& obs5, const Observation& obs6, const Observation& obs7) const { return new Observation(obs, obs2, obs3, obs4, obs5, obs6, obs7); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4, const Observation& obs5, const Observation& obs6, const Observation& obs7, const Observation& obs8) const { return new Observation(obs, obs2, obs3, obs4, obs5, obs6, obs7, obs8); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4, const Observation& obs5, const Observation& obs6, const Observation& obs7, const Observation& obs8, const Observation& obs9) const { return new Observation(obs, obs2, obs3, obs4, obs5, obs6, obs7, obs8, obs9); }
    virtual Observation* copy(const Observation& obs, const Observation& obs2, const Observation& obs3, const Observation& obs4, const Observation& obs5, const Observation& obs6, const Observation& obs7, const Observation& obs8, const Observation& obs9, const Observation& obs10) const { return new Observation(obs, obs2, obs3, obs4, obs5, obs6, obs7, obs8, obs9, obs10); }
};
```



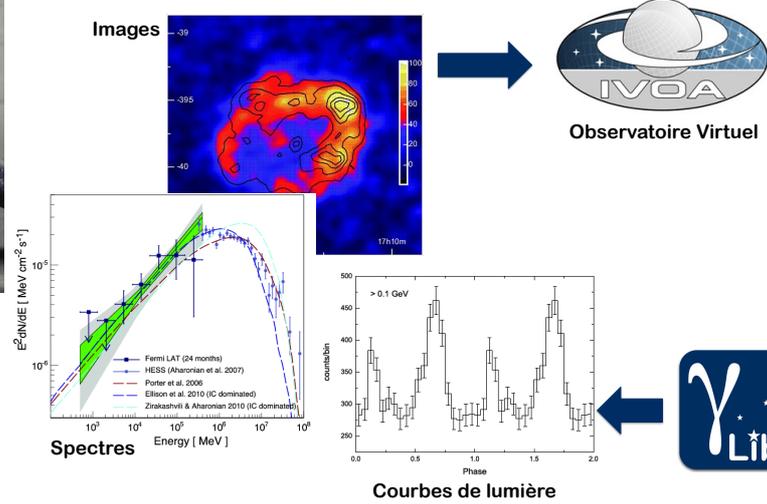
Une bibliothèque extensible pour prendre en compte de nouveaux instruments



Une implémentation native pour éviter des dépendances



Des interfaces compatibles avec les standards dans l'astronomie pour assurer l'interopérabilité



Observatoire Virtuel



COMPTEL
Fermi/LAT
H.E.S.S. (télescope Tcherevok)