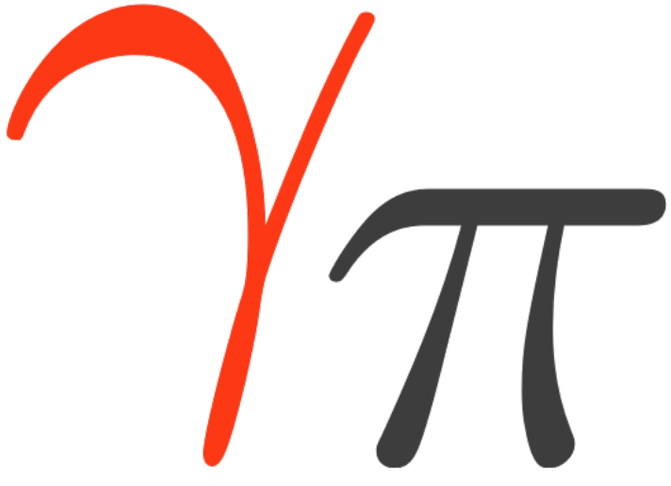


A **Python** package for **gamma-ray** astronomy

INTRODUCTION – OVERVIEW – STATUS – PLANS

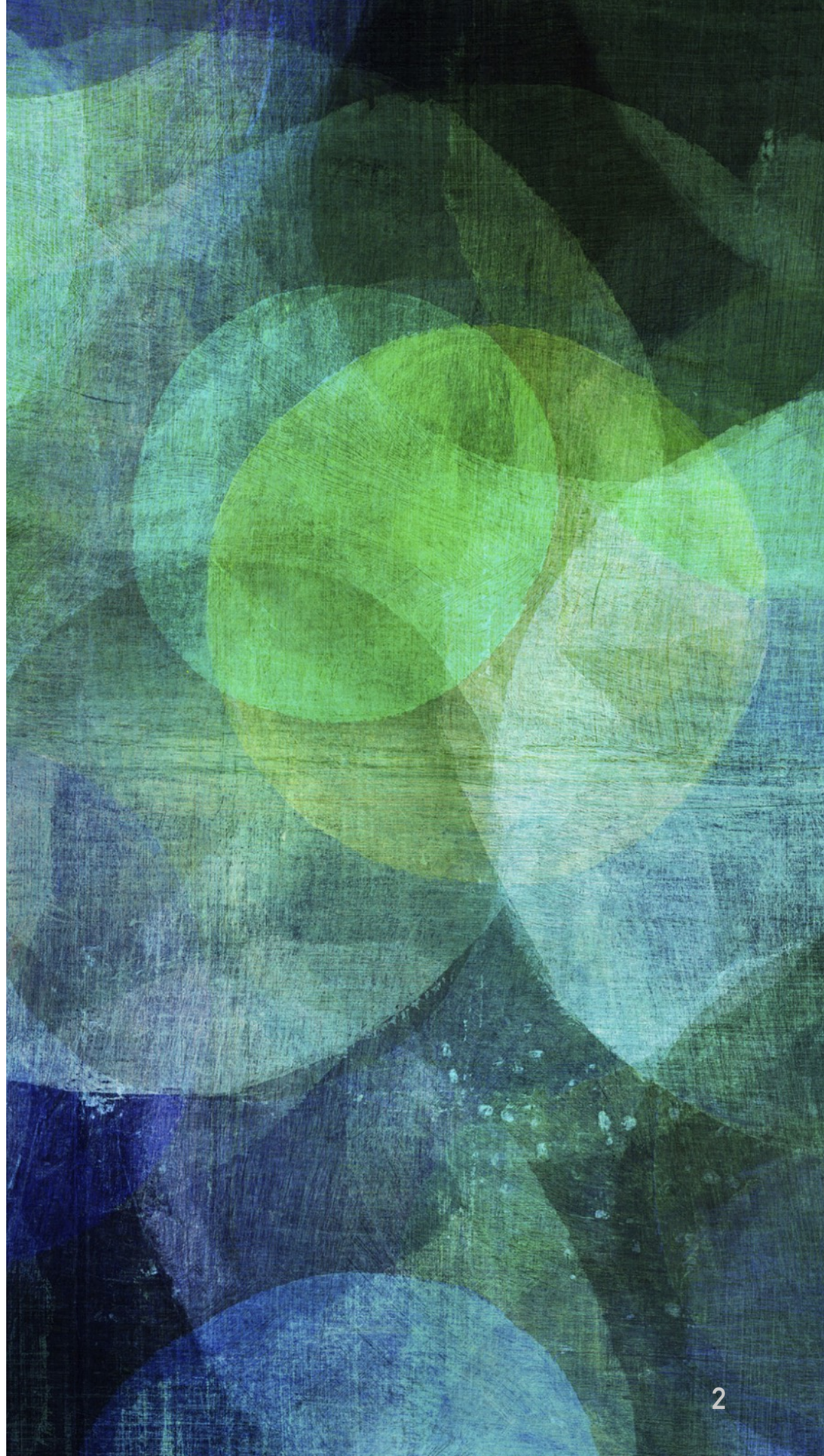
Christoph Deil (MPIK Heidelberg)

Gammapy workshop, Paris

Feb 20, 2017

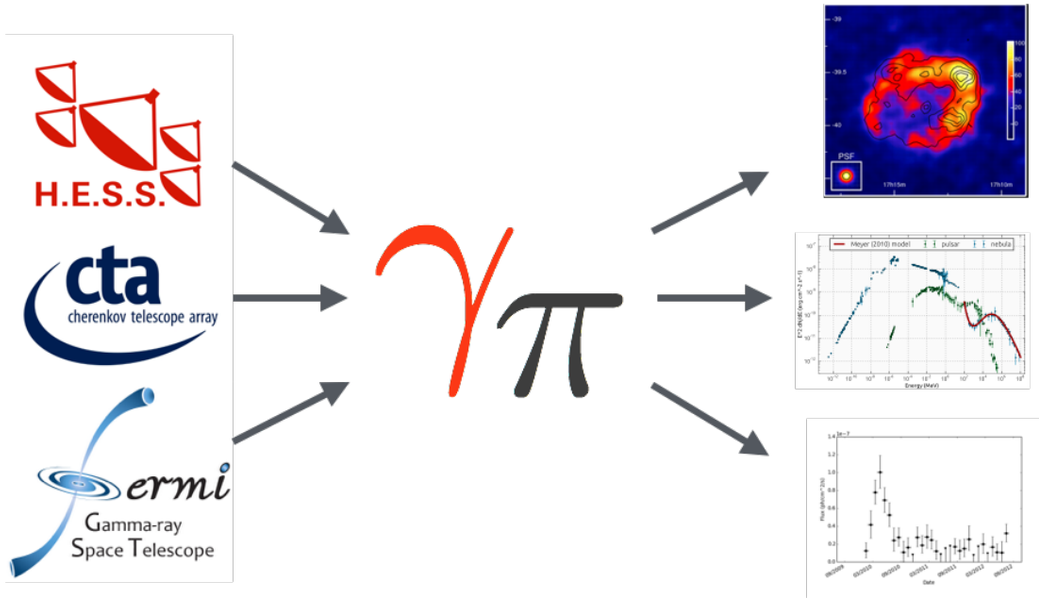
INTRODUCTION

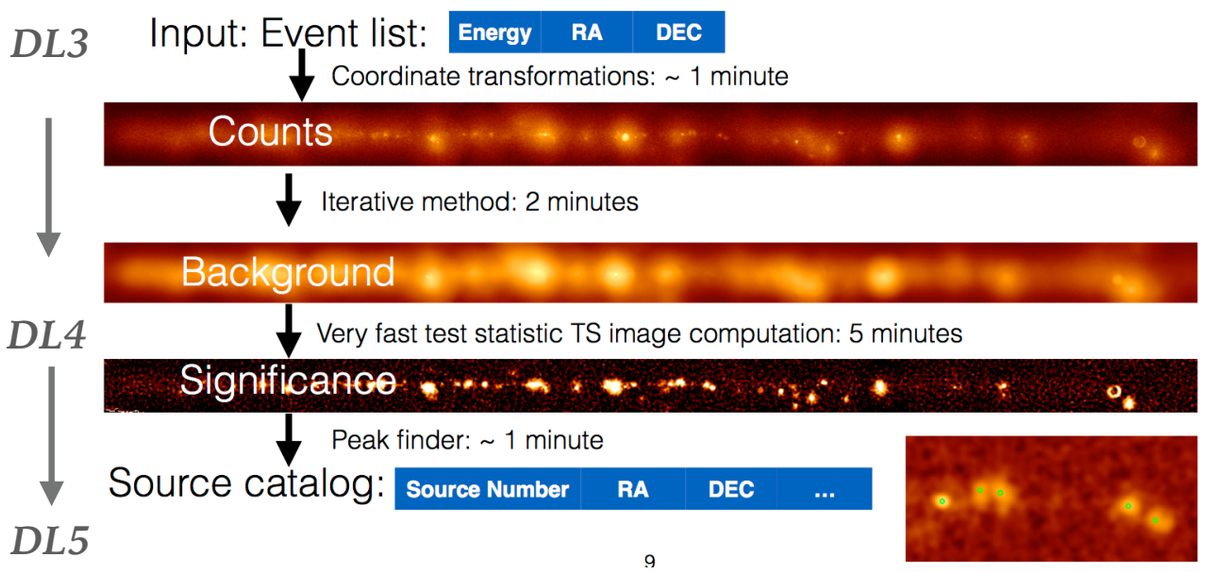
I didn't want to spend much time on making slides — this is mostly a copy of the slides from the presentation at the CTA Bologna meeting last fall



What is Gammapy?

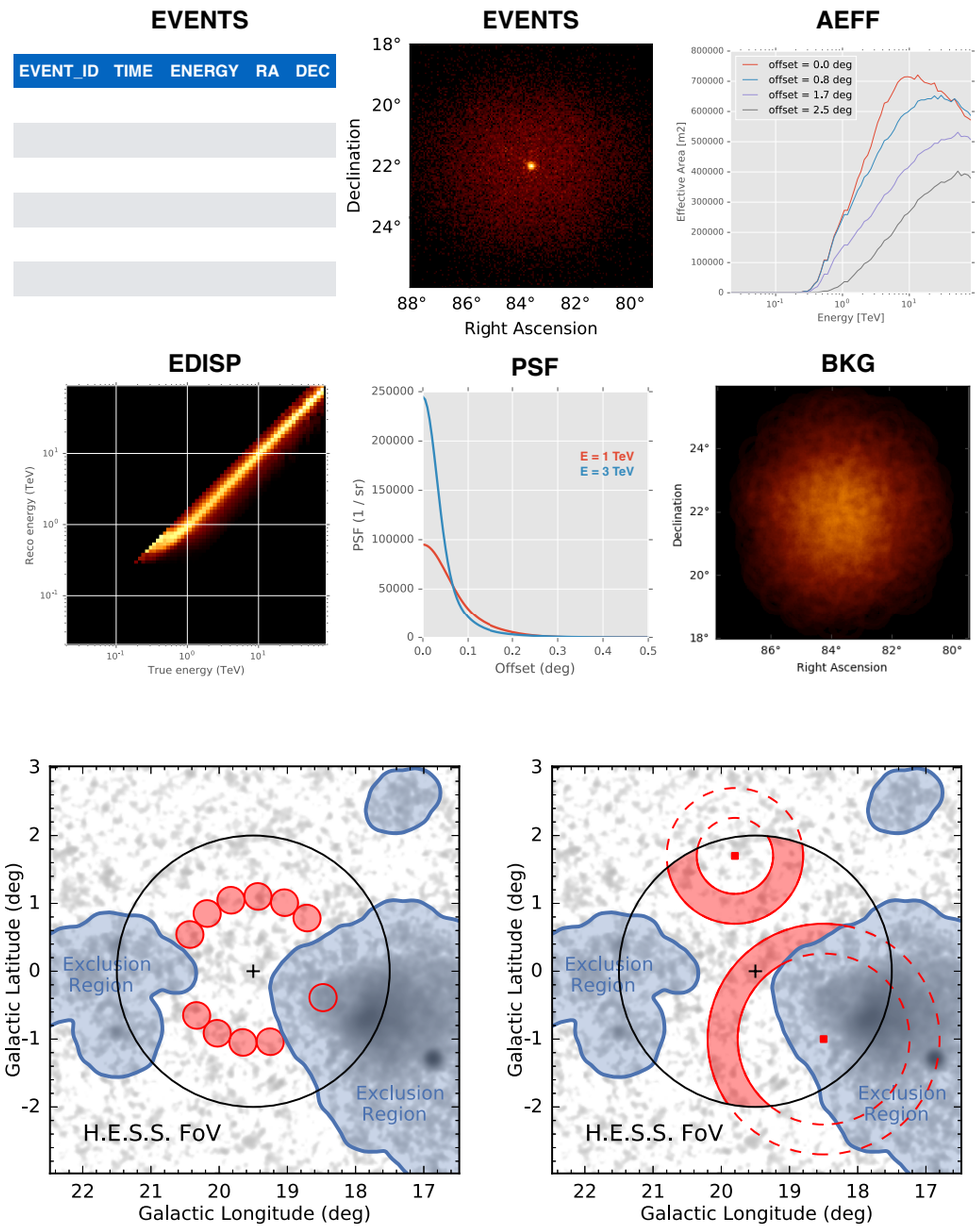
- Gammapy is a Python package for gamma-ray astronomy
- Open source, open development (on Github)
- Officially proposed as CTA science tool prototype at the Kashiwa meeting in May 2016
- Currently used for H.E.S.S., CTA prototyping and Fermi-LAT
- Used for upcoming H.E.S.S. Galactic plane survey paper



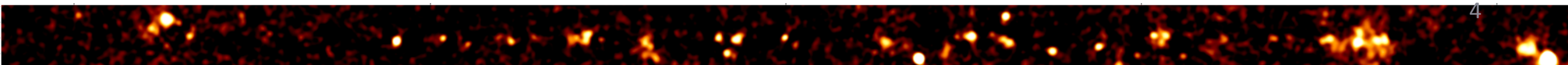


Gammapy features

- **gammapy.data & gammapy.irf**
  IACT DL3 data handling
- **gammapy.image**
  2-dim image analysis
- **gammapy.spectrum**
  1-dim region spectral analysis
- **gammapy.background**
  Background modeling methods
  (might merge in image, spectrum cube)
- **gammapy.cube**
  3-dim cube analysis (work in progress)
- **gammapy.detect**
  Source detection (image-based for now)

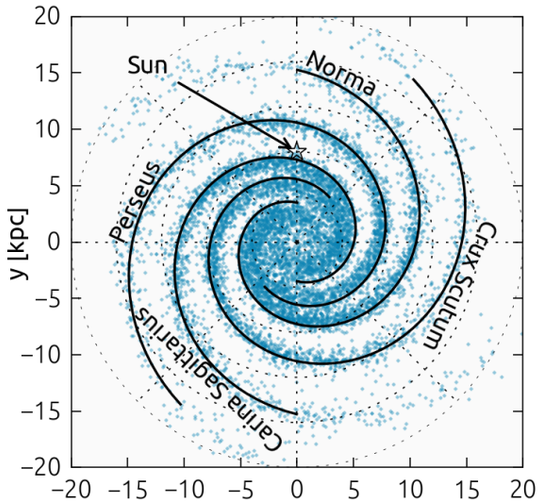


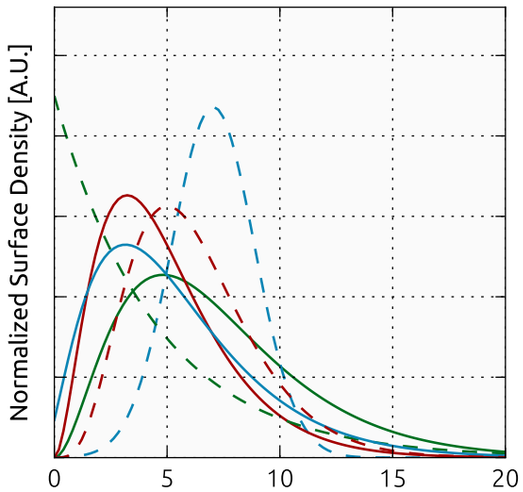
Fermi-LAT Galactic plane survey TS image ([tutorial](#))

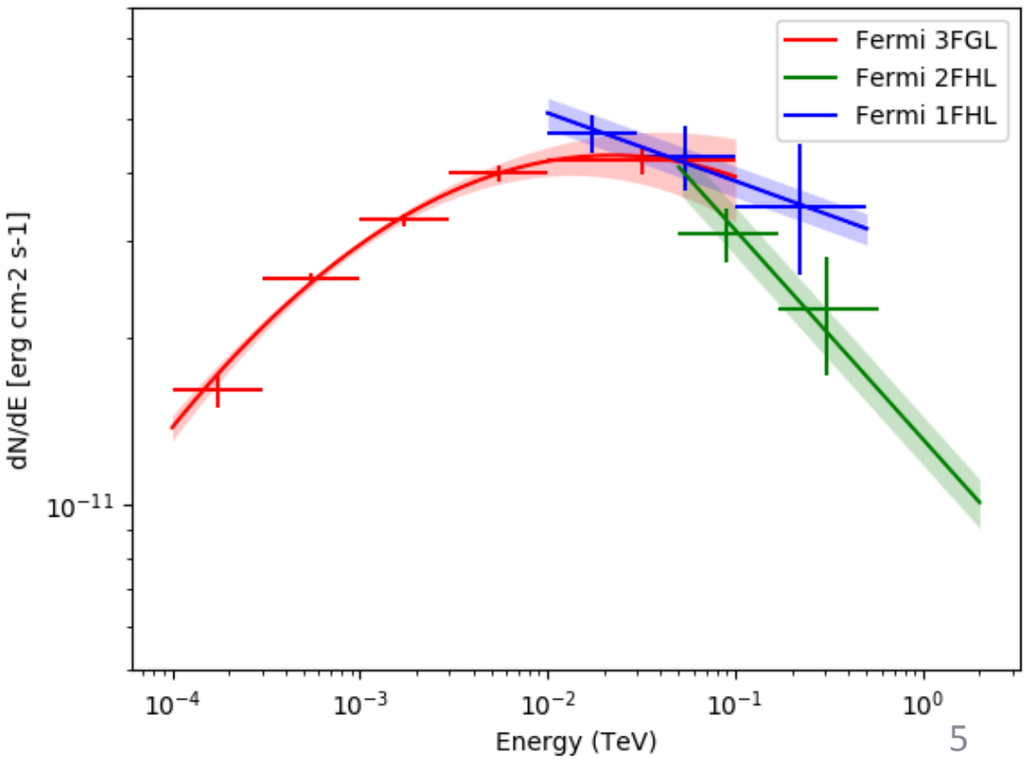


Gammapy features

- **gammapy.stats**
  Statistics methods
- **gammapy.time**
  Time analysis (not much available yet)
- **gammapy.catalog**
  - Fermi-LAT spectra, lightcurves
  - Next: TeV data (gamma-cat)
- **gammapy.astro**
  Some simple models for Galactic sources and source populations
  (could go in separate higher-level science package)
- **gammapy.scripts**
  Command line interface (CLI) tools for common operations
  (not much available yet, see comments on science too user interface in backup slides)

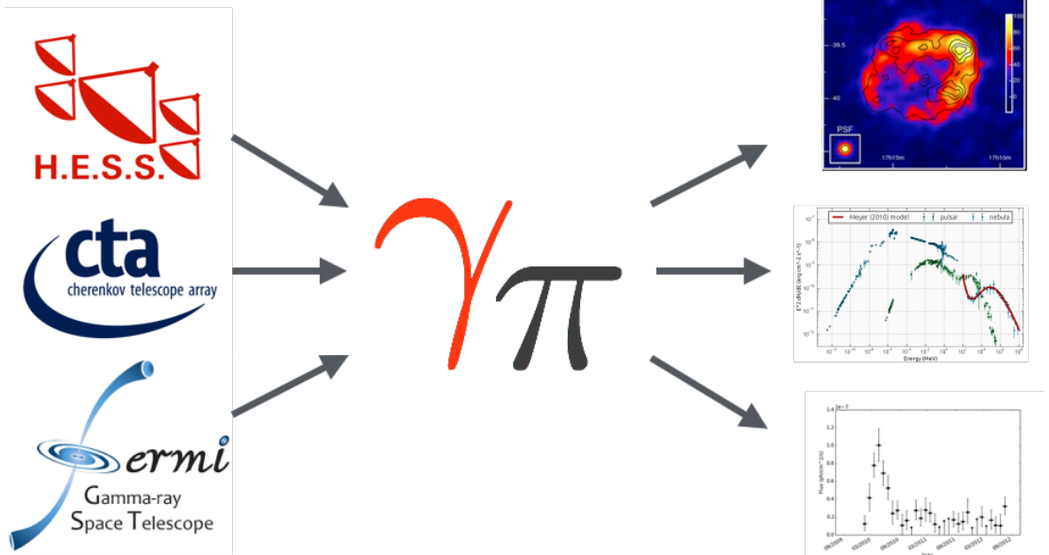


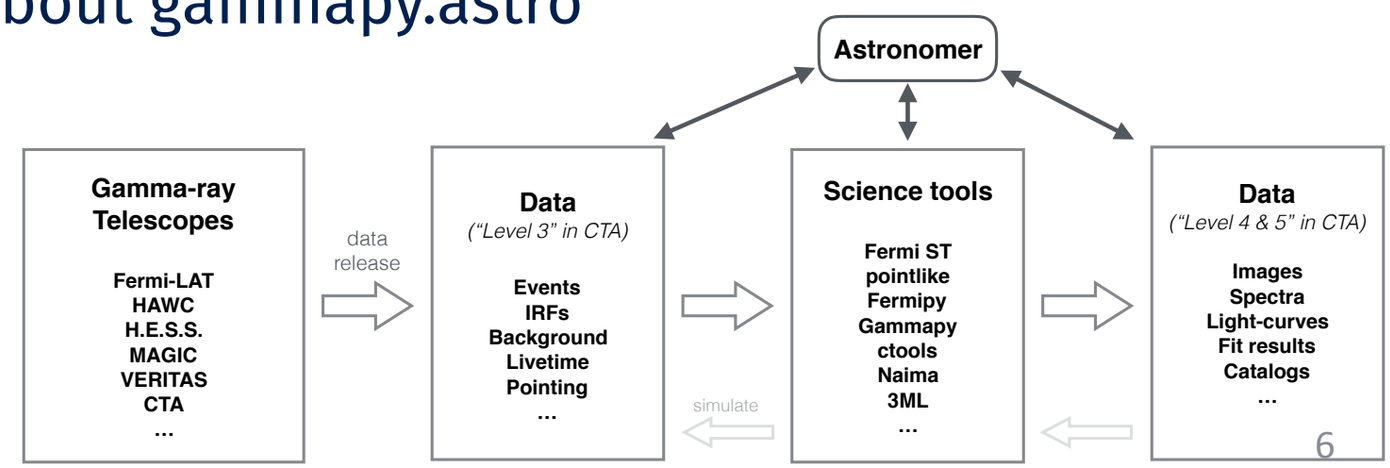




Gammapy scope & other projects

- Gammapy scope not set in stone
  Driven by users and contributions
- So far mostly prototyping of IACT DL3 FITS data formats and “classical TeV analysis” applied to H.E.S.S.
- Now project is expanding
  CTA use cases and data challenge
- Started collaborating with Fermipy
  Gammapy is the base package
  Move HEALPix, SED to Gammapy
- Astro model efforts are currently scattered (Naima, Gamera, …)
  Will have to see what to do about gammapy.astro





Gammapy approach

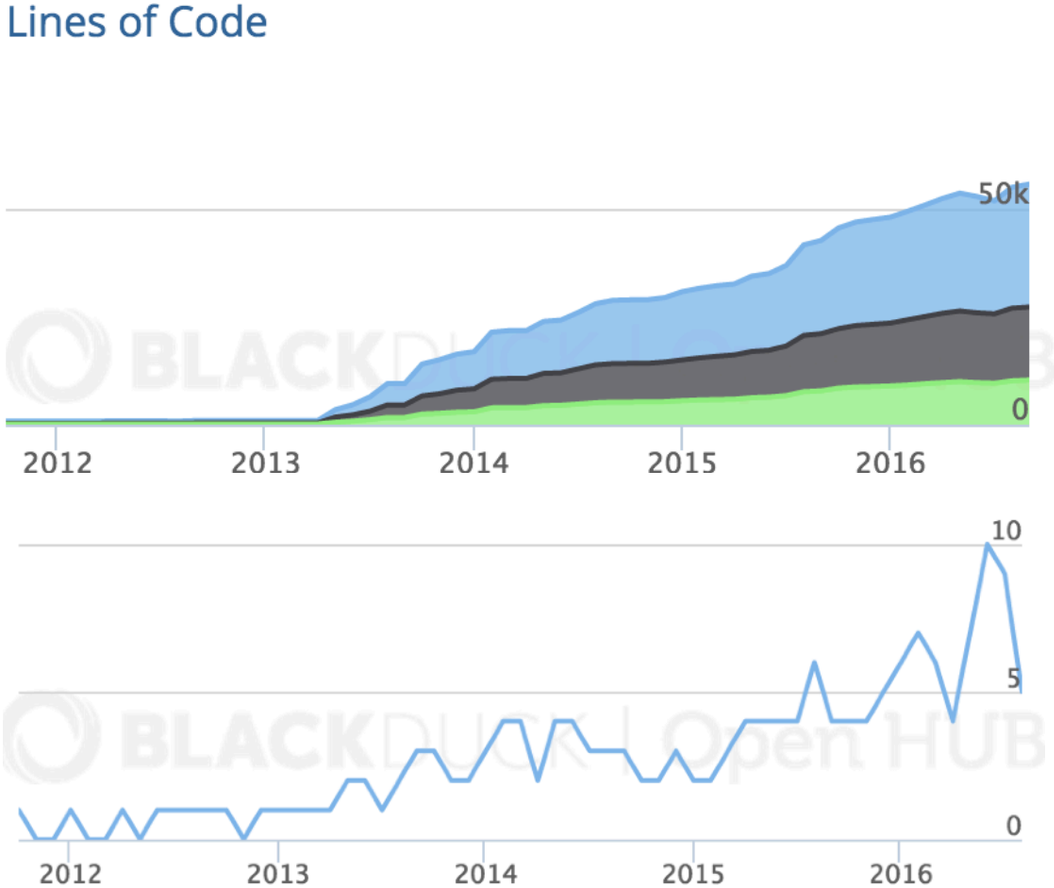
- Gammapy is written in Python, using Numpy, Scipy, Astropy
- For modeling / fitting, we currently use Sherpa
- A few stable, well-maintained, widely used dependencies
- Data in Gammapy objects *(EventList, SkyImage, …)* or Astropy objects *(Time, SkyCoord, WCS, …)* is stored as Numpy arrays

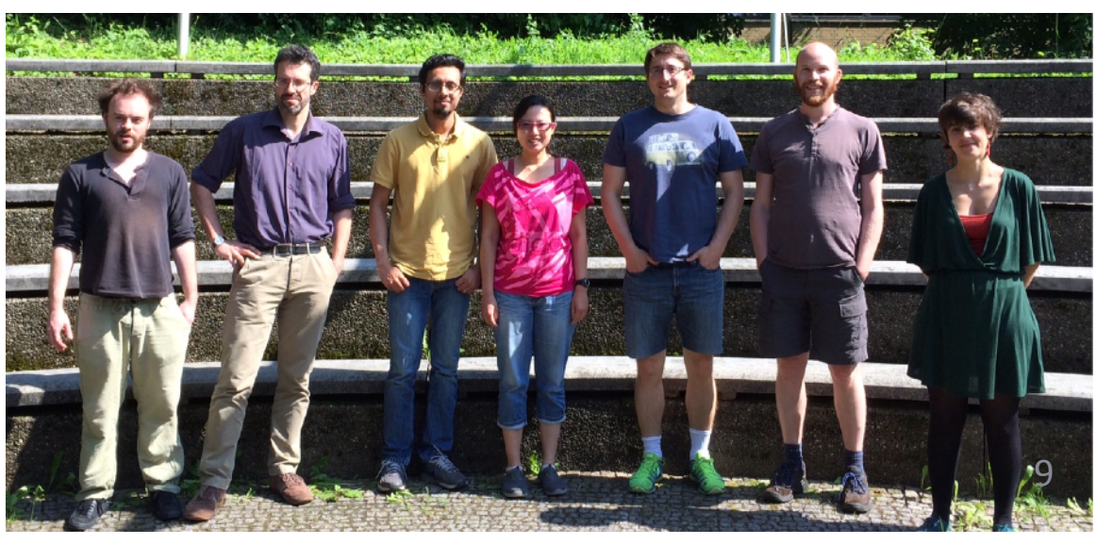
Gammapy approach benefits

- *"Standing on the shoulders of giants …"*
- Gammapy codebase is small and focused on gamma-ray astronomy
- Single high-level language codebase that's easy to use, read and extend
- Interoperable (Numpy arrays) with larger scientific Python ecosystem: iminuit, emcee, Fermipy, naima, pint, …
- Connected to the large scientific and specifically astronomy Python community

Gammapy development activity

- Gammapy development started in 2013 and is now accelerating
- 23 contributors (1+ commit)
- 27k lines of code (includes tests, but not docs/comments)
- First Gammapy coding sprint in June 2016 at MPIK, Heidelberg
- Releases ~ every 6 months, see changelog for details





Gammapy development activity

Contributors

The following people have contributed to Gammapy (fir

- Arpit Gogia (@arpitgogia)
- Axel Donath (@adonath)
- Brigitta Sipocz (@bsipocz)
- Christoph Deil (@cdeil)
- Dirk Lennarz (@dlennarz)
- Domenico Tiziani (@dltiziani)
- Ellis Owen (@ellisowen)
- Helen Poon (@helen-poon)
- Ignasi Reichardt (@ignasi-reichardt)
- Johannes King (@joleroi)
- Jonathan Harris (@JonathanDHarris)
- Julien Lefaucheur (@jjlk)
- Lars Mohrmann (@lmohrmann)
- Léa Jouvin (@JouvinLea)
- Luigi Tibaldo (@tibaldo)
- Manuel Paz Arribas (@mapazarr)
- Matthew Wood (@woodmd)
- Nachiketa Chakraborty (@cnachi)
- Olga Vorokh (@OlgaVorokh)
- Régis Terrier (@registerrier)
- Rolf Bühler (@rbuehler)
- Stefan Klepser (@klepser)
- Victor Zabalza (@zblz)

A detailed listing of contributions is here: Changelog.

Pull requests

This list is incomplete. Small improvements and bug fixes are not listed here.

See the complete Gammapy 0.6 merged pull requests list on Github.

- [#893] Add Fermi-LAT 3FGL catalog object lightcurve property (Arpit Gogia)
- [#888] Improve CTA IRF and simulation classes (point-like analysis) (Julien Lefaucheur)
- [#885] Improve spectral model uncertainty handling (Axel Donath)
- [#884] Improve BinnedDataAxis handling of lo / hi binning (Johannes King)
- [#883] Improve spectrum docs page (Johannes King)
- [#881] Add support for observations with different energy binning in SpectrumFit (Lars Mohrmann)
- [#875] Add CTA spectrum simulation example (Julien Lefaucheur)
- [#872] Add SED type e2dnde to FluxPoints (Johannes King)
- [#871] Add Parameter class to SpectralModel (Johannes King)
- [#870] Clean up docstrings in background sub-package (Arpit Gogia)
- [#868] Add Fermi-LAT 3FHL catalogue (Julien Lefaucheur)
- [#865] Add Fermi basic image estimator (Axel Donath)
- [#864] Improve edisp.apply to support different true energy axes (Johannes King)
- [#859] Remove old image_profile function (Axel Donath)
- [#858] Fix Fermi catalog flux point upper limits (Axel Donath)
- [#855] Add Fermi-LAT 1FHL catalogue (Julien Lefaucheur)
- [#854] Add Fermi-LAT dataset class (Axel Donath)
- [#851] Write Macports install docs (Christoph Deil)
- [#847] Fix Sherpa spectrum OGIP file issue (Régis Terrier and Johannes King)
- [#842] Add AbsorbedSpectralModel and improve CTA IRF class (Julien Lefaucheur)
- [#840] Fix energy binning issue in cube pipe (Léa Jouvin)
- [#837] Fix containment fraction issue for table PSF (Léa Jouvin)
- [#836] Fix spectrum observation write issue (Léa Jouvin)
- [#835] Add image profile estimator class (Axel Donath)
- [#834] Bump to require Astropy v1.3 (Christoph Deil)
- [#833] Add image profile class (Axel Donath)
- [#832] Improve NDDataArray (use composition, not inheritance) (Johannes King)
- [#831] Add CTA Sensitivity class and plot improvements (Julien Lefaucheur)
- [#830] Add gammapy.utils.modeling and GammaCat to XML (Christoph Deil)
- [#827] Add energy dispersion for 3D spectral analysis (Léa Jouvin)
- [#826] Add sky cube computation for IACT data (Léa Jouvin)

Gammapy development tools

- Gammapy is set up like Astropy, ctapipe and most other open-source packages.
- Github for version control, issue tracker, pull requests
- pytest for automated unit, integration and science tests
- Sphinx for documentation
- Jupyter notebooks for tutorials

Until now: use free cloud services for continuous integration (travis-ci, appveyor) and documentation build / hosting (readthedocs) and deployment (anaconda cloud). We might move some of that over to a Jenkins CI server at MPIK.

Gammapy installation

- Gammapy works with Python 2.7 and 3.4+ on Linux and Mac
- Stable version:
  - *pip install gammapy*
- Binary packages via conda:
  - *conda install -c openastronomy gammapy*
- Development version:
  - *git clone https://github.com/gammapy/gammapy.git*
    cd gammapy
    pip install .
- Example datasets and Jupyter notebooks are here: *https://github.com/gammapy/gammapy-extra*
- More info: http://docs.gammapy.org/

Gammapy Jupyter notebook tutorials

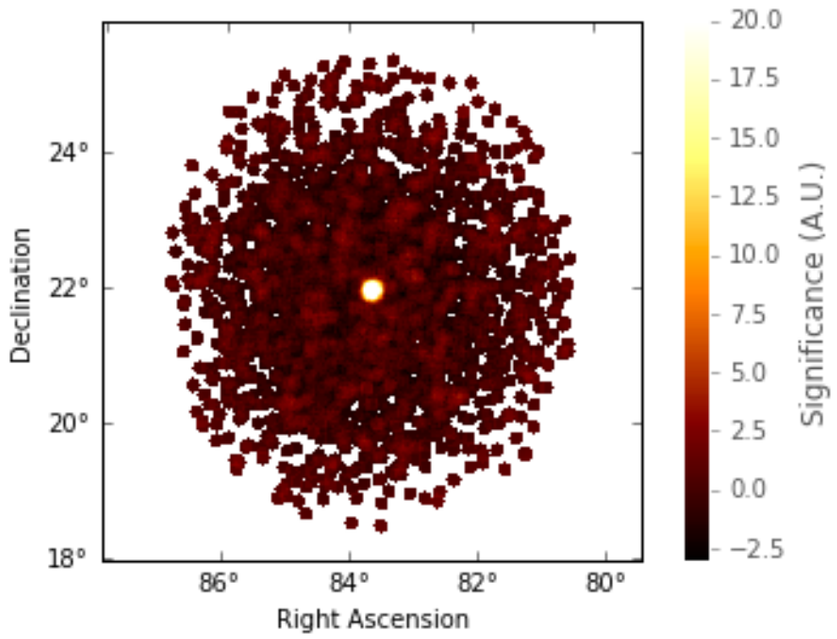
- Started to move tutorial-style documentation from RST / Sphinx to Jupyter notebooks
- *Pro: users get executable documentation, no need to start learning with copy & paste*
- *Con: notebook drawbacks:*
  - Markdown less powerful than RST / Sphinx
  - No nice diff in code reviews
- *A little better:*
  - Sphinx build is faster
  - testing notebooks in CI is a bit simpler

Notebooks

Getting started guides

To get started with gamma-ray data analysis:

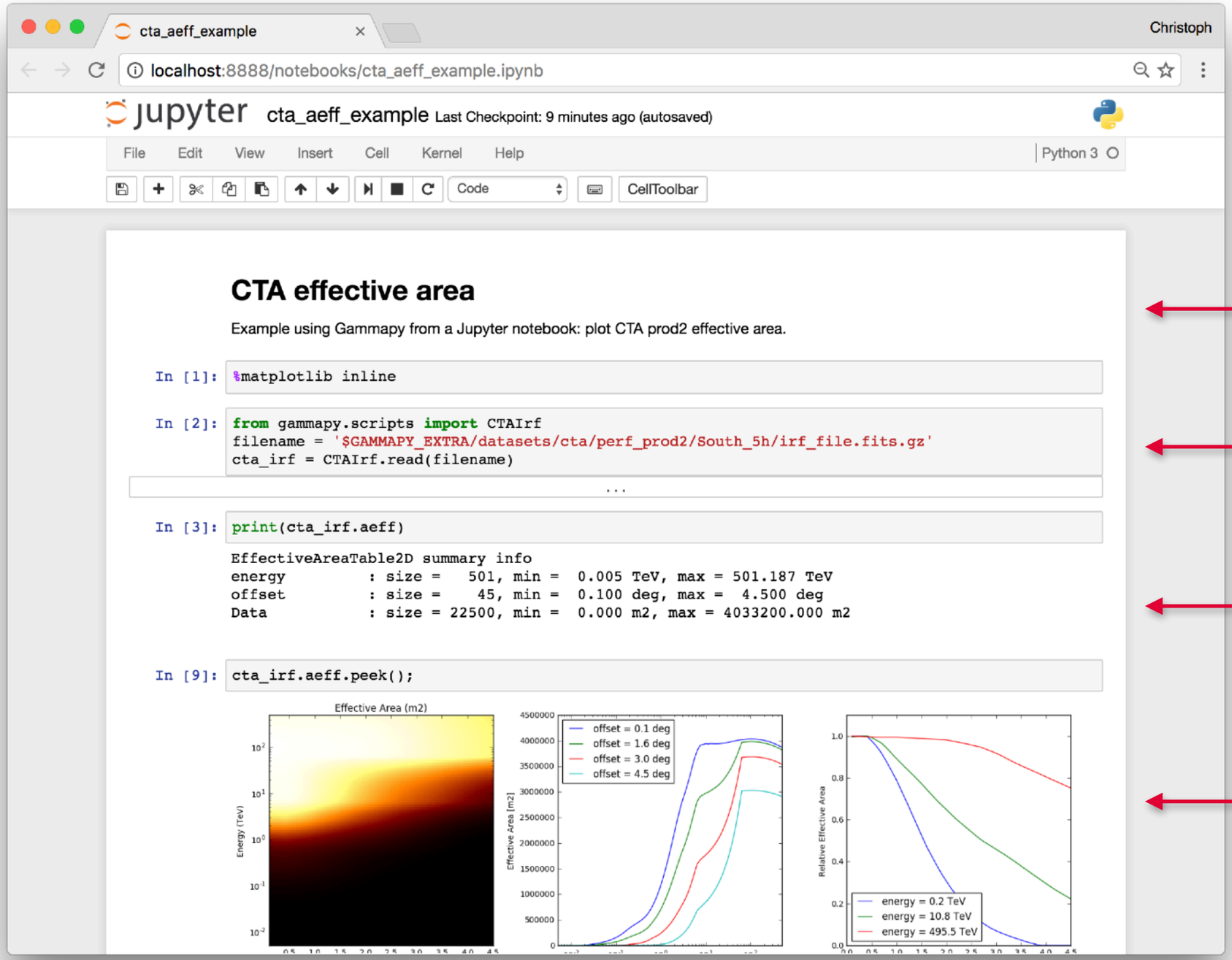
- IACT DL3 data with Gammapy (H.E.S.S. data example)
- Image analysis with Gammapy (H.E.S.S. data example)
- Spectral analysis with Gammapy (H.E.S.S. data example)
- Source detection with Gammapy (Fermi-LAT data example)
- Spectrum simulation and fitting (CTA example)



Gammapy from Jupyter notebook

Interactive data exploration and analysis from Jupyter notebook.

Notebook is a JSON file that can be shared and executed locally or on server.



Documentation

Code

Output text

Output images

Gammapy references

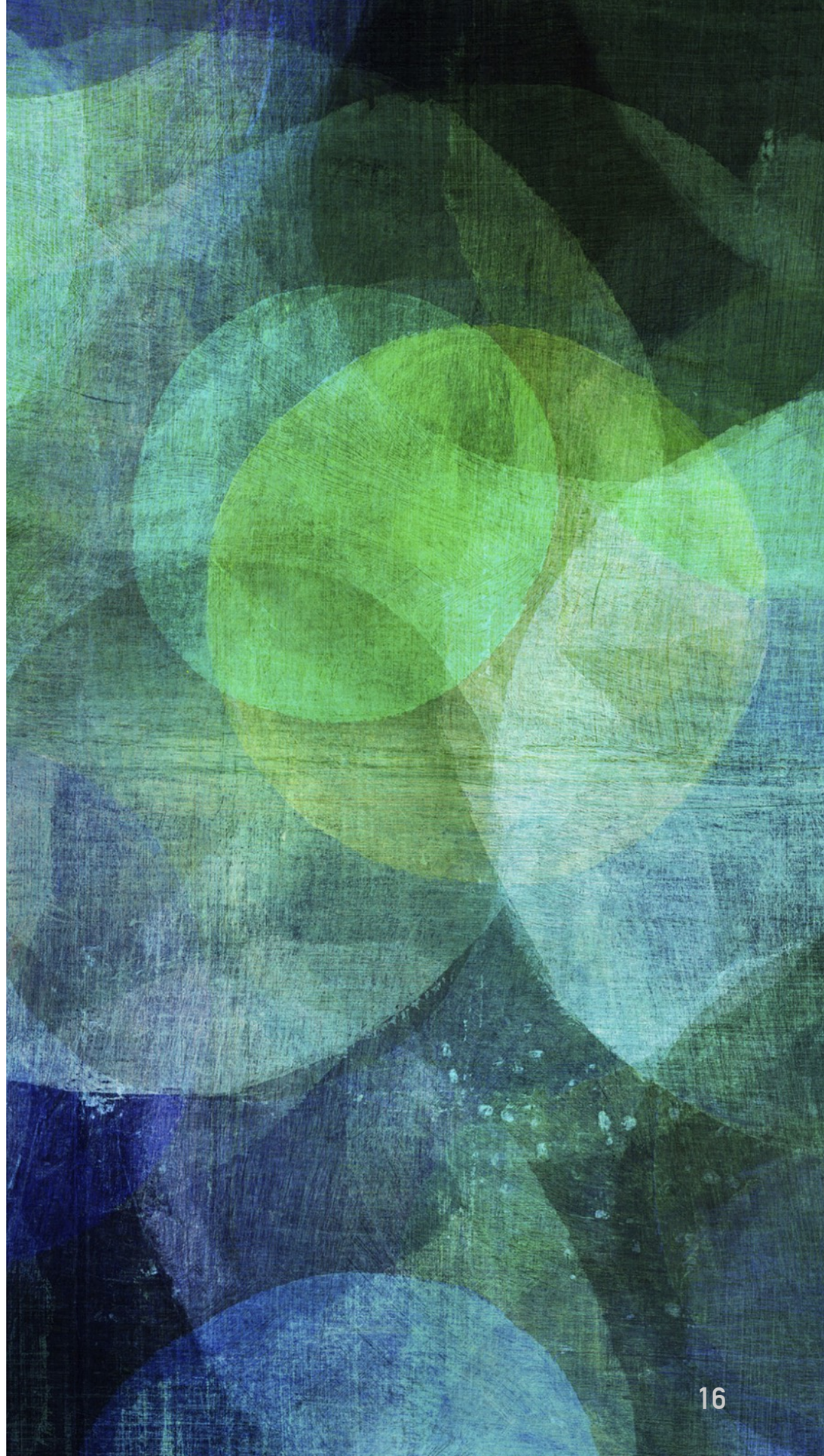
- ICRC 2015 paper: 2015arXiv150907408D
- Code: https://github.com/gammapy/gammapy
- Docs: http://docs.gammapy.org
- Mailing list: http://groups.google.com/group/gammapy
- License: BSD-3 (same as Numpy, Scipy, Astropy, ...)

In the CTA context:

- Gammapy slides IACT data meeting (Meudon, April 2016)
- Gammapy slides CTA consortium meeting (Kashiwa, May 2016)
- Gammapy CTA spectrum simulation example notebook

CTA SCIENCE TOOLS

Some more infos on CTA context for Gammapy



CTA SCIENCE TOOLS

- Job of the CTA science tools (ST):
  - Input: DL3 data (events and IRFs and aux files)
  - Output: images, spectra, lightcurves, fitted models, …, science results!
- CTA is a much larger project than current IACTs and an open observatory
- CTA observatory (CTAO) organisation is responsible to distribute data and science tools to 1000+ users for 20+ years.
- Like for many other things (hardware, other software),
  the CTA observatory doesn't produce the CTA ST.
- CTA ST developed by CTA consortium (CTAC) members / institutes
- CTA ST will be selected / accepted using an in-kind contribution review

CTA SCIENCE TOOLS

- Two competing prototypes for the CTA science tools
  - **Gammalib / ctools**
    - Gammalib started by Jürgen Knödlseder at IRAP 10 years ago
    - C++ library with SWIG Python wrapper (~150k SLOC)
      (ctools are mostly written in Python now)
    - No dependencies (except CFITSIO)
    - Proposed as CTA ST prototype since the start of CTA
  - **Gammapy**
    - Gammapy started by Axel Donath and me at MPIK 3 years ago
    - Python package (~30k SLOC), no command line interface yet.
    - Proposed as CTA ST prototype at the two CTA consortium meetings in 2016.
    - The concept is not new, e.g. PyFACT participated equally with Gammalib / ctools in the 2011 / 2012 CTA data challenge - see Raue & Deil (2012)

WHY DID I START GAMMAPY?

- Axel and I had a lot of Python scripts using Python / Sherpa for the H.E.S.S. Galactic plane survey analysis and wanted to share the useful parts (morphology fitting, catalog convenience functions) with our colleagues …
- As a developer: I don't want to re-invent all the wheels (Numpy, Scipy, MINUIT, Sherpa, XML parsing, FFT, random number generator, ….). I just want to implement gamma-ray methods in a nice high-level language and build on existing powerful libraries.
- As a user: I like the flexibility and tools available with Python (being able to implement models & algorithms in Python)
- As a gamma-ray astronomer: I think we should avoid isolating the gamma-ray science tools developers and users, and collaborate with the larger astronomy and data science community.
  - Remember: most people will move from gamma-ray astronomy to other parts of astronomy, science, industry after a few years, and it's good to have a few tools you can use there in your tool-belt.
  - I think CTA ST development is just starting now. Method and code development will be a large ongoing task by dozens of astronomers for the next decade, not a few hired software developers in the next year or two.

CTA ACTIVITIES

- CTA science tool selection timescale and procedure unclear.
- Short-term: “CTA first data challenge” (1DC).[1] The 1DC is organised by Stefan Funk, focused on PHYS studies and ST.
- https://forge.in2p3.fr/projects/data-challenge-1-dc-1/wiki
- Now: Sky models, observation pattern, prod 3 IRFs
- Soon: simulated DL3 data
- Milano meeting March 6–8

[1] If you were involved in the first data challenge (1DC) in 2011/2012, you must forget about it immediately, or call it 0DC from not on!

ACTIVITIES IN CTA USING GAMMAPY

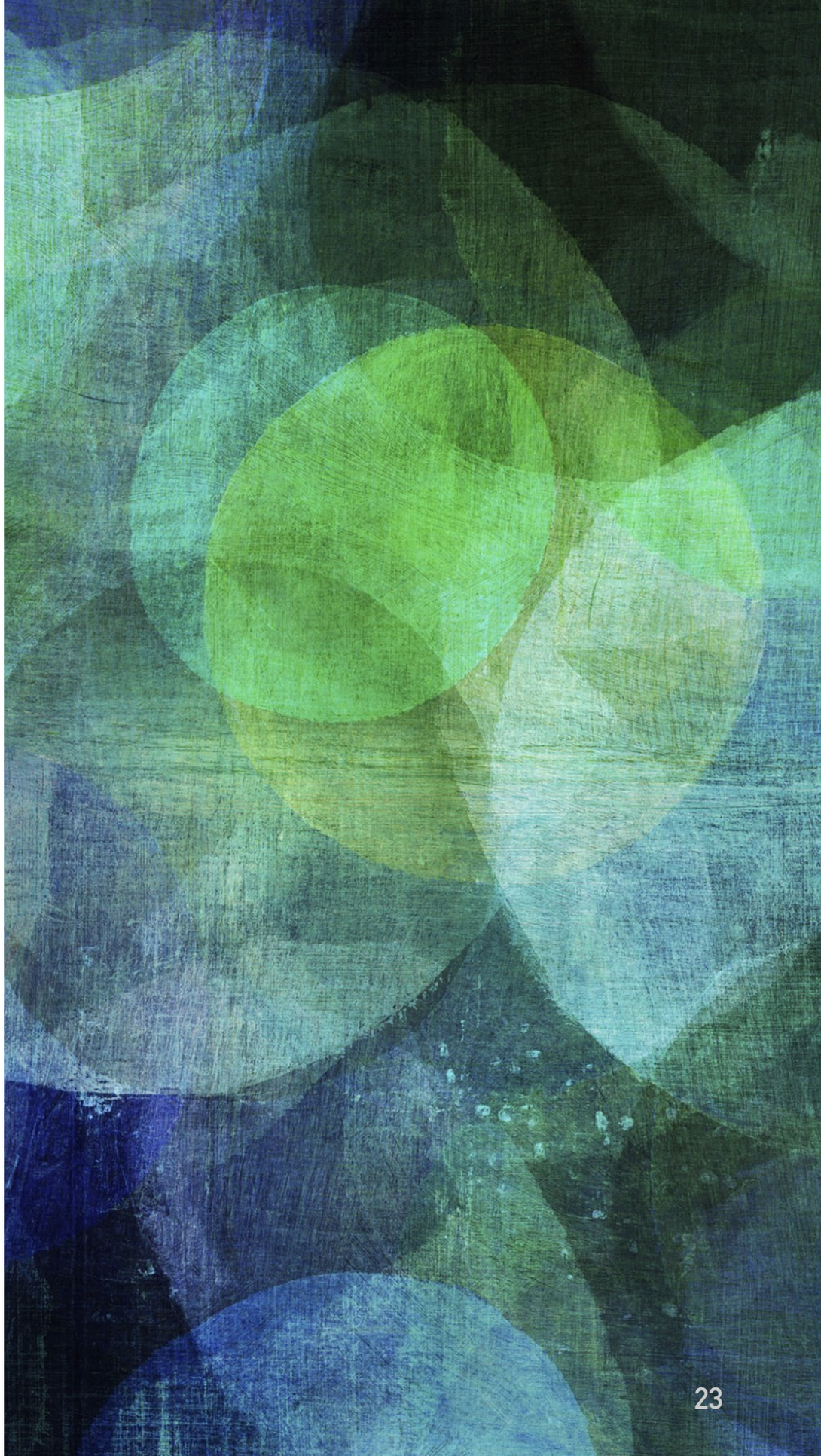
- Julien Lefaucheur — AGN spectra, EBL obsorption (see tutorial tomorrow)
  Roberta Zanin — GPS sky model (built with gamma-cat & Gammapy)
- Not so much CTA / Gammapy activity so far, but probably will increase a lot in the coming months due to CTA data challenge.
  - XML model serialisation — adding support in Gammapy is in progress
  - Event sampling and gammapy-obssim tool — unfortunately not started yet

DO WE NEED GAMMAPY.CTA?

- So far, Gammapy is an open-source project on Github, with a large contributor base and being applied for H.E.S.S., CTA, Fermi-LAT.
- This doesn't work for the CTA ST, they have to be coupled and targeted more closely at CTA (repository, tests, documentation, releases, deployment, issue tracker, code review, feature prioritisation, user support).
- Going the CTA ST route would mean to a certain degree giving up the current workflow (Github -> CTA Gitlab) and community aspect of the project.
- So this is a big question. For now a concrete question is:
  - Do we need a gammapy.cta sub-package?
    Or even a separate gammapy-cta repository?
    (I think no, second repo doesn't give any advantages, repos would be version-coupled anyways, so it's just an unnecessary complication.)
  - Or just a separate "CTA simulation & analysis" docs
    landing page and more examples for CTA?
    (I think yes, this is very much needed for CTA people.)

DISCUSSION

I think we'll mostly discuss in detail on Wednesday, but we could start today.



DISCUSSION — GAMMAPY PROJECT / MANAGEMENT

- How can we improve project management?
- How should we organise / distribute the work?
  - *Will be discussed later in the week.*

DISCUSSION — GAMMAPY PACKAGE (TECHNICAL)

- Modeling — what role does Sherpa play?
  - One extreme: Everything done in Gammapy (models & data & likelihood). Sherpa / iminuit / emcee as fitting backends.
  - Other extreme: Everything done with Sherpa as a framework. Gammapy defines model & data & IRF classes as Sherpa sub-classes.
- Instrument response functions (IRFs)
  - How to support point-like and full-enclosure in formats and Gammapy classes (mostly about book-keeping)?
  - Is the current assumption of one EDISP and PSF for a whole field of view good enough?
  - Gammapy's role in prototyping better IRF production? What data structures / formats will we need to handle for IRFs?

DISCUSSION

What do you want to do with Gammapy?

What should we focus on this week?