GammaLib & ctools: A framework for the analysis of high energy gamma ray datasets

PyGamma - March 2019
Josh Cardenzana (on behalf of the ctools team)
Jürgen Knödlseder
Luigi Tibaldo

@ctoolssoftware
What can ctools do?

Analysis software for **scientific analysis of IACT data**, specifically targeting **CTA**:

- Fit spatial, spectral, & temporal properties of gamma-ray sources:
  - Unbinned, binned, and stacked 3D/4D maximum likelihood analysis
  - Classical On/Off analysis (**ring background skymaps, reflected region spectra**)
- Jointly fit data between instruments:
  - *Fermi*-LAT, COMPTEL, CTA, current IACTs (H.E.S.S., VERITAS, MAGIC)
- Simulate observations
- Analyze fit results (residual maps/plots)
- Generate light curves from data
- **And much more...**
What can ctools do?

Overview
- **49 separate tools**, each with a dedicated purpose
- **20 example scripts** capable of visualizing the results

Uses a **“Modular tools”** methodology similar to other high-energy astrophysical data analysis software (e.g. fermitools, ftools)

Makes it possible for users to build flexible workflows & analysis pipelines
What can ctools do?

**Example workflow** *(unbinned likelihood analysis)*:

- Inspect data runs with `csobsinfo` and `csobsselect`
- Apply energy/time/FoV cuts with `ctselect`
- Prepare model components with `csbkgmodel` and `csmodelmerge` (search for sources with `cssrcdetect`)
- Fit models to data with `ctlike`
- Fit spectral SED with `csspec`
- Generate residual maps and spectra with `csresmap` and `csresspec`
- Generate butterfly plot with `ctbutterfly`

Entire workflows & pipelines can be defined and run through `csworkflow`. 

March 2019

ctools - PyGamma - Heidelberg
Background models (run specific):

- **Spatial**: Gaussian multiplied by gradient in x and y (no energy dependence)
- **Spectral**: Piecewise broken power law (across 7 bins of energy from 0.1 - 12.8 TeV)

* Unbinned likelihood fit results
H.E.S.S. DR1 - Crab

- **1.75 hours** of Crab Nebula data
- Unbinned analysis tested with different spectra

<table>
<thead>
<tr>
<th>Model</th>
<th>TS</th>
<th>$N_{\text{src}}$</th>
<th>$k_0$</th>
<th>$\Gamma$</th>
<th>$E_c$ or $\beta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>PL</td>
<td>2023.2</td>
<td>691</td>
<td>4.6 ± 0.2</td>
<td>2.69 ± 0.07</td>
<td>-</td>
</tr>
<tr>
<td>EPL</td>
<td>2031.6</td>
<td>693</td>
<td>5.2 ± 0.4</td>
<td>2.20 ± 0.18</td>
<td>6.5 ± 2.7</td>
</tr>
<tr>
<td>CPL</td>
<td>2033.6</td>
<td>694</td>
<td>4.6 ± 0.2</td>
<td>2.28 ± 0.14</td>
<td>-0.22 ± 0.07</td>
</tr>
</tbody>
</table>

**Units:** $N_{\text{src}}$ (counts); $k_0$ ($10^{-11}$ ph cm$^{-2}$ s$^{-1}$ TeV$^{-1}$); $E_c$ (TeV)

**Power law with exponential cut-off (EPL) shown**

Background subtracted counts ($0.02^\circ \times 0.02^\circ$ bins)
H.E.S.S. DR1 - Crab

Can compare results from a range of different analyses

- All spectra use a **power law with exponential cut-off**:

<table>
<thead>
<tr>
<th>Analysis method</th>
<th>TS</th>
<th>$N_{src}$</th>
<th>$k_0$</th>
<th>$\Gamma$</th>
<th>$E_c$</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.E.S.S. published</td>
<td>n.c.</td>
<td>4283</td>
<td>3.84 ± 0.09</td>
<td>2.41 ± 0.04</td>
<td>15.1 ± 2.8</td>
</tr>
<tr>
<td>Unbinned</td>
<td>2031.6</td>
<td>693</td>
<td>5.2 ± 0.4</td>
<td>2.2 ± 0.2</td>
<td>6.5 ± 2.7</td>
</tr>
<tr>
<td>Joint binned</td>
<td>1949.2</td>
<td>666</td>
<td>5.2 ± 0.4</td>
<td>2.2 ± 0.2</td>
<td>6.0 ± 2.4</td>
</tr>
<tr>
<td>Stacked binned</td>
<td>1881.5</td>
<td>645</td>
<td>4.9 ± 0.4</td>
<td>2.1 ± 0.2</td>
<td>5.7 ± 2.2</td>
</tr>
<tr>
<td>Joint On/Off (wstat)</td>
<td>1134.3</td>
<td>562</td>
<td>5.0 ± 0.4</td>
<td>2.2 ± 0.2</td>
<td>6.8 ± 3.4</td>
</tr>
<tr>
<td>Joint On/Off (cstat)</td>
<td>995.5</td>
<td>549</td>
<td>4.8 ± 0.4</td>
<td>2.2 ± 0.2</td>
<td>8.0 ± 4.6</td>
</tr>
<tr>
<td>Stacked On/Off (wstat)</td>
<td>1359.5</td>
<td>575</td>
<td>5.0 ± 0.4</td>
<td>2.2 ± 0.2</td>
<td>7.4 ± 3.6</td>
</tr>
<tr>
<td>Stacked On/Off (cstat)</td>
<td>1114.5</td>
<td>554</td>
<td>4.9 ± 0.4</td>
<td>2.2 ± 0.2</td>
<td>6.6 ± 3.2</td>
</tr>
</tbody>
</table>

**Units:** $N_{src}$ (counts); $k_0$ (10^{-11} ph cm^{-2} s^{-1} TeV^{-1}); $E_c$ (TeV)

- Published data covers much longer time (10.6 vs 1.75 hours),
- **In general, all ctools results are in agreement**

March 2019

Aharonian et al. (2006)
H.E.S.S. DR1 - Crab nebula extension

**ctools** can also fit extended source models:

- Tested Gaussian model fit to H.E.S.S. Crab nebula data:
  - Sigma: $51'' (\pm 18)$
  - Robust against spectral model

- Holler et al. (2017):
  - Sigma: $52'' (\pm 3)$

- Difference in position possibly due to systematics in the position reconstruction or IRFs

Chandra X-ray image overlayed with extension models
H.E.S.S. DR1 - Crab nebula extension

Assessment of extension fit:

- Spectral residuals are flat
- Source subtracted significance is Gaussian
- Small spatial residuals could be result of systematic uncert. in the PSF model

Background subtracted residual counts map smoothed by 0.2° disk
**H.E.S.S. DR1 - MSH 15-52 extension**

*ctools* fits asymmetric extended models

Tested elliptical Gaussian:
- **ctools** Unbinned likelihood:
  - $\sigma_{\text{maj}} = 6.9' \pm 0.5$
  - $\sigma_{\text{min}} = 3.0' \pm 0.3$
  - Position angle = $152^\circ \pm 4$

- Aharonian et al. (2005):
  - $\sigma_{\text{maj}} = 6.4' \pm 0.7$
  - $\sigma_{\text{min}} = 2.3' \pm 0.5$
  - Position angle = $139^\circ \pm 13$

- Results are in good agreement

*Background subtracted counts map smoothed by 0.02° Gaussian*
H.E.S.S. DR1 - PKS 2155-304 lightcurve

c tools generates light curves (cslightcrv):

• Aharonian et al. (2009) covers same time period

• General shape and amplitude of flaring data appears consistent

Unbinned likelihood analysis

On/Off analysis

March 2019

ctools - PyGamma - Heidelberg
H.E.S.S. DR1 with ctools - summary

Comprehensive analysis of observations from the H.E.S.S. data release:

- 4 sources studied: Crab nebula, MSH 15-52, RX J1713.7-3946, and PKS 2155-304
  - Spectral, morphology, temporal analyses
- Different ctools analyses yield consistent results
  - e.g. 3D likelihood & classical ON/Off analyses agree
- Detailed results on full H.E.S.S. public data release in prep.

Comparison to published results:

- ctools results in reasonably good agreement with published work, despite notable differences in data samples:
  - Published work typically have larger data samples
  - Different processing software versions

Assessment of the tools:

- Tools seem to be in a stable and ready to use state
- Joint analysis of Fermi-LAT and H.E.S.S. data demonstrate ability to conduct broad-band, multi-instrument analyses.
What do people use *ctools* for?

1. 2019MNRAS.493.1802T
   Tavecchio, F.; Romano, P.; Landoni, M.; Vercellone, S.
   Putting the hadron beam scenario for extreme blazars to the test with the Cherenkov Telescope Array

2. 2019A&A...525..336H
   Hutton, M.; Merten, C.; Combi, C.; Martin, D.
   Mostly generating simulations of CTA observations for "perspectives for CTA" type studies (population studies, Fermi pulsar detection, dark matter detection potential, etc…)

3. 2019arXiv190305605S
   Sudoh, I.
   Deep learning detection of transients

4. 2018A&A...619A...7V
   Vink, J.; Struyf, M.; Frack, C.
   Spatial likelihood analysis for MAGIC telescope data. From instrument response modelling to spectral extraction

5. 2019cta.book..C
   CTA Consortium
   Science with the Cherenkov Telescope Array

6. 2018MNRAS.481.5046R
   Romano, P.; Vercellone, S.; Pesce, C.; Tavecchio, F.; Landoni, M.; Knödlseder, J.
   Prospects for gamma-ray observations of narrow-line Seyfert 1 galaxies with the Cherenkov Telescope Array

7. 2018arXiv18110970Y
   Yang, L.; Razzaque, S.; Soebur
   Constraints on very high energy gamma-ray emission from the Fermi Bubbles with future ground-based experiments

8. 2018A&A...619A...7V
   Vink, J.; Struyf, M.; Frack, C.
   Spatial likelihood analysis for MAGIC telescope data. From instrument response modelling to spectral extraction

9. 2018MNRAS.471.431B
   Sánchez-Conde, Miguel A.; Balbi, A.; Bamba, A.; Bruzzi, R.; Combi, C.; Martin, D.; Merten, C.
   Sensitivity of the Cherenkov Telescope Array to the detection of a dark matter signal in comparison to direct detection and collider experiments

10. 2018MNRAS.473.1485R
    Paradias, I.; Stumpe, M.; Auvergne, M.; Cefa, G.; Mau, C.
    Searching for gamma-ray counterparts to gravitational waves from merging binary neutron stars with the Cherenkov Telescope Array

11. 2018MNRAS.478..434R
    Bernet, A.; Saito, Y.; Zampieri, L.; Hassan, T.
    Prospects for the detection of high-energy (E > 25 GeV) Fermi pulsars with the Cherenkov Telescope Array

12. 2018MNRAS.478..434R
    Bernet, A.; Saito, Y.; Zampieri, L.; Hassan, T.
    Prospects for the detection of high-energy (E > 25 GeV) Fermi pulsars with the Cherenkov Telescope Array

13. 2018MNRAS.481.5046R
    Romano, P.; Vercellone, S.; Pesce, C.; Tavecchio, F.; Landoni, M.; Knödlseder, J.
    Prospects for gamma-ray observations of narrow-line Seyfert 1 galaxies with the Cherenkov Telescope Array

14. 2018MNRAS.471.431B
    Sánchez-Conde, Miguel A.; Balbi, A.; Bamba, A.; Bruzzi, R.; Combi, C.; Martin, D.; Merten, C.
    Sensitivity of the Cherenkov Telescope Array to the detection of a dark matter signal in comparison to direct detection and collider experiments

Publications citing *ctools*:
http://cdsads.u-strasbg.fr/cgi-bin/nph-ref_query?bibcode=2016A%26A...593A...1K&amp;refs=CITATIONS&amp;db_key=AST
Software usage and development practices
The goals of scientific software

Successful community-driven scientific software typically have the following attributes:

- **User-focused**
  - Trustworthy results
  - Installation should be straight-forward
  - Intuitive interface (or at least well documented)
  - Tutorials & examples demonstrating typical use cases
  - User support

- **Easily maintainable by developers**
  - Source code well documented & follows a common standard
  - Specifics of implementation well documented
  - Resources for assuring quality (tests, continuous integration, etc...)
  - Dependencies kept to a minimum (also nice for the user)
Installation procedure

Installation is done typically in 2 ways:

• **Anaconda** *(preferred for end-user):*

  ```
  $ conda config --append channels conda-forge
  $ conda config --append channels cta-observatory
  $ conda create -n myenv python=2.7  # or e.g. python=3.7
  $ source activate myenv
  (myenv) $ conda install ctools
  (myenv) $ source activate myenv
  (myenv) $ conda update ctools  # for updating software
  ```

• **From source** *(git or tar-ball):*
  – Git clone is typical for development
  – ‘tar-ball’ useful for disconnected systems
  – Uses standard `configure` -> `make` -> `make install` procedure

• **Mac OS `.dmg` binaries also available**
All tools can be executed from **Python** or directly from the **command-line**

### Python

*Scripts or Jupyter notebooks*

```python
skymap = ctools.ctskymap()
skymap['inobs'] = evfile
skymap['emin'] = emin
skymap['emax'] = emax
skymap['nxpix'] = 40
skymap['nypix'] = 40
skymap['binsz'] = 0.02
skymap['proj'] = 'TAN'
skymap['coordsys'] = 'CEL'
skymap['xref'] = 83.63
skymap['yref'] = 22.01
skymap['bkgsubtract'] = 'IRF'
skymap['caldb'] = caldb
skymap['irf'] = irf
skymap.run()
```

### Command-line

*Queries for parameters not passed initially*

```bash
$ ctskymap
```

- Input event list or observation definition XML file [events.fits] selected_events.fits
- First coordinate of image center in degrees (RA or galactic l) (0-360) [83.63]
- Second coordinate of image center in degrees (DEC or galactic b) (-90-90) [22.01]
- Projection method (AIT|AZP|CAR|GLS|MER|MOL|SFL|SIN|STG|TAN) [CAR]
- Coordinate system (CEL - celestial, GAL - galactic) (CEL|GAL) [CEL]
- Image scale (in degrees/pixel) [0.02]
- Size of the X axis in pixels [200]
- Size of the Y axis in pixels [200]
- Lower energy limit (TeV) [0.1]
- Upper energy limit (TeV) [100.0]
- Background subtraction method (NONE|IRF|RING) [NONE]
- Output skymap file [skymap.fits]
Scripted analyses improves reproducibility
- Can run within a Jupyter notebook (see the tutorials)
- Can pass outputs directly into other tools without needing intermediary files
  - Call `run()` method to prevent writing output to disk
  - Example demonstrates setting up a binned likelihood analysis

```python
import gammalib
import ctools

bindata = ctools.ctbin()
# ... configure ctbin ...
bindata.run()

expcube = ctools.ctexpcube()
# ... configure ctexpcube ...
expcube.run()

edispcube = ctools.ctedispcube()
# ... configure ctedispcube ...
edispcube.run()

psfcube = ctools.ctpsfcube()
# ... configure ctpsfcube ...
psfcube.run()

bkgcube = ctools.ctbkgcube()
# ... configure ctbkgcube ...
bkgcube.run()

obs = bindata.obs()[0]
obs.response(expcube.expcube(), psfcube.psfcube(),
edispcube.edispcube(), bkgcube.bkgcube())
obs_list = gammalib.GObservations()
obs_list.append(obs)
obs_list.models(bkgcube.models())

fitter = ctools.ctlike(obs)
# ... configure ctlike ...
fitter.execute()
```
Learning resources

An array of tutorials, walkthroughs, and downloadable Jupyter notebooks exist on the website: cta.irap.omp.eu/ctools/

- Installation instructions
- Individual tools tutorials
- Descriptions of different model components
- How to analyze H.E.S.S. DR1 data (new in 1.6.0)
- How to analyze Fermi-LAT data

Radial disk

```py
<source name="Crab" type="ExtendedSource" >
<spatialModel type="RadialDisk">
 <parameter name="RA"  scale="1.0" >
 <parameter name="DEC" scale="1.0" >
 <parameter name="Radius" scale="1.0" >
 <spectral type="..." >
 ...</source>
</source>
```

Radial Gaussian

```py
<source name="Crab" type="ExtendedSource" >
<spatialModel type="RadialGaussian">
 <parameter name="RA"  scale="1.0" >
 <parameter name="DEC" scale="1.0" >
 <parameter name="CenterValue" scale="1.0" >
 <parameter name="Sigma" scale="1.0" >
 <spectral type="..." >
 ...</source>
</source>
```

Combined likelihood analysis

Now you are ready to do a joint maximum likelihood analysis:

```py
In [5]:
like = ctools.ctlike()
like['nobs'] = 'obs.xml'
like['caldb'] = 'prod2'
like['irf'] = 'South_0.5h'
like['inmodel'] = '$CTOOLS/share/models/cta15model'
like['outmodel'] = 'crab_results.xml'
like.run()
```

Instead of providing an event list or a counts cube, you now call ctools.ctlike recognises this format and automatically performs a combined likelihood analysis.

How to combine observations?

Generally, the CTA data you may want to analyse will not only be composed of a single observation (a.k.a. run) but of a list of observations that should be combined in a joint analysis. ctools has the capability to collect individual observations in a list and to perform for example a joint maximum likelihood fit of all observations in a single shot. Here is an example that illustrates how to do that.

We will start with the usual Python imports:

```py
In [1]:
import gammapy
import ctools
import scripts
```

```py
In [2]:
MJD = np.array([56950, 56951, 56952])
like = ctools.ctlike()
like['nobs'] = 'obs.xml'
like['caldb'] = 'prod2'
like['irf'] = 'South_0.5h'
like['inmodel'] = '$CTOOLS/share/models/cta15model'
like['outmodel'] = 'crab_results.xml'
like.run()
```
User support

Users can request help in several ways:

- Simple questions or general discussion can be posted in the ctools Slack channel
  - E.g. “I’m trying to do X, what tool should I use?”
- Directly from the developers and users through email list
  - ctools@irap.omp.eu
- Participating in monthly ctools Users/Devs calls
- Submitting issues & feature requests to the ctools & GammaLib repositories
  - Created issues typically end in pull requests
Many developers:

- 9 coding sprints since 2013 (typically 1-2 per year)
- Most contribute a single feature

<table>
<thead>
<tr>
<th>Country</th>
<th>Institute</th>
<th>Contributors</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>IRAP</td>
<td>7</td>
<td>91.48%</td>
</tr>
<tr>
<td></td>
<td>IRFU</td>
<td>1</td>
<td>0.10%</td>
</tr>
<tr>
<td></td>
<td>LAPP</td>
<td>1</td>
<td>0.02%</td>
</tr>
<tr>
<td></td>
<td>LUPM</td>
<td>1</td>
<td>0.02%</td>
</tr>
<tr>
<td></td>
<td>CENBG</td>
<td>1</td>
<td>0.02%</td>
</tr>
<tr>
<td>Germany</td>
<td>DESY</td>
<td>8</td>
<td>5.02%</td>
</tr>
<tr>
<td></td>
<td>MPIK</td>
<td>4</td>
<td>1.65%</td>
</tr>
<tr>
<td></td>
<td>Erlangen</td>
<td>5</td>
<td>1.54%</td>
</tr>
<tr>
<td>Poland</td>
<td>Cyfronet</td>
<td>1</td>
<td>0.13%</td>
</tr>
<tr>
<td>Spain</td>
<td>IFAE</td>
<td>1</td>
<td>0.02%</td>
</tr>
</tbody>
</table>
Code breakdown

Code split:

- **C++** (for the parts that need to be fast and parallelization)
- **Python** (for easy user-interfacing)
- Nearly all C++ classes & methods accessible in Python via SWIG wrappers.

**GammaLib**
( > 166k lines of code )

- C++: 93.0%
- Python: 6.0%
- Other: 1.0%

**ctools**
( > 34k lines of code )

- C++: 54.7%
- Python: 37.2%
- Other: 8.1%
Development practices

Code managed via Git (GitLab flavor)

Coding standards:

- Python 2.7 & 3.5+ compatible
- C++98 compatible
- Enforced code formatting, e.g.:
  - Tool names begin with ‘ct...’ if written in C++ and ‘cs...’ if in Python
  - Class and variable names follow a specific format
- Doxygen documentation for all classes, methods, and variables
- All pull requests approved by a ‘central authority’ (a.k.a. Jürgen K.) to ensure standards are enforced
Development workflow and Release methodology

When a new release is ready:

- Freeze new feature integration
- Resolve critical bugs/issues
- Ensure code and build tests pass
- Tag new release
Quality assurance

**Jenkins:** Continuous integration & release management (tests many platforms)

**SonarQube:** Monitors technical debt, code duplication, test coverage

---

**Project ctools: integration**

Project name: ctools-integrate-os

Build, compile and test ctools integration branch on various operating systems

**Configurations**

- centos6_64
- centos7_64
- debian6_64
- fedora17_64
- macosx10
- macosx11
- macosx12
- macosx13
- macosx14
- macosx7
- macosx8
- macosx9
- mandriva2011_64
- opensolaris_11_32
- opensuse12_64
- sles_64
- ubuntu12_64
- ubuntu16_64

---

**Coverage**

- Line coverage: 89.7%
- Condition coverage: 72.1%
- Uncovered conditions: 6,030
- Coverage on new code: N/A

---

**Technical Debt**

- Technical Debt: 31d
- Technical Debt Ratio: 1.5%
- Issues: 364
Summary

Results on H.E.S.S. public DR1 in prep (plan to submit in the next few months):

- Includes RXJ 1713.7-3946
- Includes *Fermi*-LAT joint analysis of the Crab nebula

*ctools* & *GammaLib* are mature analysis software for analysis of gamma ray data:

- > 14 refereed papers cite *ctools* since 2016
- Used internally for CTA first data challenge
- Proposed as a candidate for the “CTA Science Tools”

Find out more:

- *ctools* site: [ctools](http://cta.irap.omp.eu/ctools/)
- *GammaLib* site: [gammalib](http://cta.irap.omp.eu/gammalib/)
- Email: [ctools@irap.omp.eu](mailto:ctools@irap.omp.eu)
- Twitter & Facebook: [@ctoolssoftware](https://twitter.com/ctoolssoftware)
References

First *ctools* & GammaLib paper:

Papers referenced in presentation:
Backup
Background models (run specific):

- **Spatial**: Gaussian multiplied by gradient in x and y (no energy dependence)
- **Spectral**: Piecewise broken power law (across 7 bins of energy from 0.1 - 12.8 TeV)

Spatial background parameters as a function of run number and energy