

## GammaLib - Feature #1044

### Implement ON-OFF fitting

12/16/2013 04:59 PM - Martin Pierrick

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Martin Pierrick	<b>% Done:</b>	80%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.1.0		
<b>Description</b>			
<p>After the work done during the first coding sprint, classes now exist to hold most data required for an ON-OFF classical analysis (regions, ON and OFF spectra, ARF and RMF arrays,...). What remains to be implemented now is the optimization part. This should allow both the fitting of a given spectral shape and the determination of flux points.</p> <p>As a starting point, it was decided to add to the gammalib only the minimum that is required, and to leave higher-level functionalities to specific ctools. These functionalities can then be moved to gammalib if it seems more appropriate.</p> <p>Anticipating on a possible evolution of the GObservations/GObservation, the optimization function will be attached to the observation and not to the container of observations (contrary to the initial design of the gammalib).</p>			
<b>Related issues:</b>			
Precedes GammaLib - Action # 1781: Finish implementation of on-off fitting			<b>Closed</b> <b>05/30/2016</b>

#### History

##### #1 - 01/21/2014 11:55 AM - Martin Pierrick

- % Done changed from 0 to 50

Created a dedicated branch and pushed to it a version of the code that compiles (on my machine). Still remains to be tested through python scripts. Also, the branch forks a bit behind the current state of the code. Will do the merging later.

The likelihood function is at the observation level, and the call of the optimizer is done at the container level (now the standard also for the rest of the gammalib). The likelihood function slightly differs in its arguments from the other likelihood functions. Can homogeneize that later if deemed necessary.

No unit test written so far. I will first try to play with this new functionality.

##### #2 - 01/23/2014 05:00 PM - Martin Pierrick

- File `test_OnOff.py` added

Here is a script to test the ON-OFF fitting (and associated functionalities by the way). The script simulates two wobble observations with one ON region and any number of OFF regions distributed around the centre of each pointing (rough method). It allows to compare ON-OFF analysis and unbinned analysis over the entire field of view. So far, it runs and tells that no fit is performed for the ON-OFF analysis. Problem of singular matrix. Inspection under way...

##### #3 - 01/24/2014 06:06 PM - Martin Pierrick

- File deleted (`test_OnOff.py`)

##### #4 - 01/24/2014 06:09 PM - Martin Pierrick

- File `test_OnOff.py` added

Attached a new version of the test script for the ON-OFF analysis. Corrected a few bugs and added many prints for checking. The `gammalib` code was also modified but currently yields a segmentation fault. Hopefully this will be solved during the second coding sprint...

**#5 - 01/30/2014 12:19 PM - Martin Pierrick**

- File deleted (`test_OnOff.py`)

**#6 - 01/30/2014 12:22 PM - Martin Pierrick**

- File `test_OnOff.py` added

Latest version of the script.

A few notes:

- OFF region creation is using a rough method that works only close to equator
- Python interface at the moment does not allow using derived classes specific methods for `GSkyRegion` objects
- Hard-coded path to data base
- ON-OFF fit not performed properly (`gammalib` problem, not the script)

**#7 - 01/31/2014 03:07 PM - Martin Pierrick**

Summary of the second coding sprint at DESY:

- ON-OFF fitting debugged, now works and give reasonable results (on just one test case)
- Code layout was made more similar to that of the global fit case (the standard `gammalib/ctool` one)
- `GSkyRegion` derived classes were debugged (constructors, DS9 file saving)
- `GSkyRegionRing` class created and `GSkyRegionMap` almost done
- Some test cases for the `GSkyRegionRing` class
- Draft python scripts for ON-OFF analysis ctools: `ctonoffreg`, `ctonoffirf`, `ctonofffit`
- Modified `GCTAPointing`:
- Implemented a `GHorizDir` class
- Reads in a pointing fits file, describing the pointing position in alt/az as a function of time
- Added `dir_horiz(time)` that interpolates the pointing table and returns alt az

Remains to be done for the ON-OFF analysis:

- Add handling of exclusion map (a `GSkyRegion*` member in `GCTAOnOffObservation`)
- Add energy dependence (ON region size varying with energy; could be computed from a table to be loaded)
- Make effective area computation more general (to allow ring OFF regions...; the integration can be done in a ctool so we would just need storage)
- Update alpha parameter calculation (very rough at the moment)
- Precomputation of OFF region solid angle (done too many times in likelihood computation at the moment)
- Write test cases for the ON-OFF analysis
- Have cscripts or ctools modules for a complete analysis

**#8 - 02/03/2014 11:05 AM - Martin Pierrick**

- File deleted (`test_OnOff.py`)

**#9 - 02/03/2014 11:06 AM - Martin Pierrick**

- File `test_OnOff.py` added

Revised version of the script after modifications by Anneli. She has tested the save methods for the ON-OFF data (and fixed some bugs).

**#10 - 02/17/2014 10:18 PM - Knödlseeder Jürgen**

- Target version set to 2nd coding sprint

**#11 - 02/17/2014 10:23 PM - Knödlseeder Jürgen**

- Status changed from New to In Progress

Is there any stuff that is sufficiently tested so that can it be merged into devel?

**#12 - 02/18/2014 08:07 AM - Martin Pierrick**

Not many tests performed so far, just the comparison of fitting a power-law spectrum by unbinned and ON-OFF analyses, for a Crab-like source (so a strong one). What remains to be done is testing the code for a weaker source, and determining spectral points using a NodeFunction.

**#13 - 07/19/2014 02:10 AM - Knödlseeder Jürgen**

- Target version deleted (2nd coding sprint)

**#14 - 03/11/2016 05:53 PM - Martin Pierrick**

Summary of progress after 5th coding sprint at DESY:

- Caught up with the development of the rest code over the past two years
- Replaced the old background model by `GCTAModelIrfBackground` (currently no other choice)
- Implemented clean calculation of alpha (ratio of ON to OFF effective areas)
- Implemented energy-dependent alpha
- Implemented dependencies on theta, phi, zenith, azimuth (instrument coordinates)
- Added a precalculation of the background rate in OFF regions (method `compute_bgd()` in `GCTAOnOffObservation`)
- Revised/corrected calculation of expected ON and OFF counts
- No modification to the likelihood calculation part

Status:

- The code runs and returns meaningful results
- ... but pull distributions show that true values are not exactly recovered on average

In the only tested case so far, a simple point source with power-law spectrum and two wobble observations, the average prefactor over 100 runs is about 5% below the true value. The photon index is also slightly below true value but the deviation is not as large. As a comparison, an unbinned analysis over the same setup leads to perfect recovery of the true values (on average).

To do:

- Investigate the systematic offset in prefactor mentioned above
- Use new class `Gfilename` where needed
- Move includes from `hpp` to `cpp` (new practice)

Current limitations:

- There is currently no check on region overlap; it is left to the user to input correct parameters
- The effective area and background rates are currently taken at the center of circular regions and not integrated over their full extent; this is probably alright for small regions but it needs to be improved upon for the general case
- The code currently handles only collections of circular regions; at least a ring region should be added.

That's all folks !

**#15 - 03/11/2016 06:00 PM - Martin Pierrick**

- File `pull_OnOff.py` added

- File `test_OnOff.py` added

Attached two scripts:

1- `test_OnOff.py` for testing the ON-OFF analysis on a single run: observations are simulated for two wobble pointings on either side of a point source with power-law spectrum; ON-OFF analysis is setup and executed; unbinned analysis is performed for comparison

2- `pull_OnOff.py` to perform a series of such runs and plot pull distributions

Both scripts are self-contained.

**#16 - 03/11/2016 06:01 PM - Martin Pierrick**

- File deleted (`test_OnOff.py`)

**#17 - 03/11/2016 06:10 PM - Martin Pierrick**

- % Done changed from 100 to 70

...during 5th coding sprint

**#18 - 04/28/2016 12:33 AM - Knödseder Jürgen**

- Target version set to 1.1.0

- % Done changed from 70 to 80

I started with integrating the code.

First I went over the code and adapt it to the coding conventions. I also needed to adjust the so far existing unit test at it was broken due to the additional of the models parameter in the `compute_response()` method.

I also removed the `GCTAOnOffObservations` class as it is no longer needed. Now `GCTAOnOffObservation` objects can be simply appended to the `GObservations` container (see #1669). This means that `GCTAOnOffObservation` now derives from `GObservation`.

There are a number of issues that I have been identified so far:

- there is no complete unit test that checks the On/Off fitting functionality. Such a test should be added to the `test_CTA.py` script (there is already a unit test, but it does so far not perform On/Off fitting)
- the computation of the Hessian in `GCTAOnOffObservation::likelihood()` looks strange. Could you please write down the intended formula in the documentation (can use LaTeX style), and then check whether the formula is indeed implemented?
- the `GCTAOnOffObservation::compute_alpha` method actually computes the ratio between effective areas, not exposures. Am I correct that it should compute the ratio of exposures? What about the solid angle?
- in the `GCTAOnOffObservation::compute_rmf` method, if a weighting is done, should it not be done over effective area times livetime? Where is the time coming in? Could you write down the formula in the documentation what the method should be doing? Also, should the `aeff` method not use true instead of reconstructed energy?
- shouldn't the `model_on()` and `model_off()` methods be private (it's always simpler to make a private method public when really needed then vice versa)
- are the `alpha()` and `offtime()` methods needed?

Furthermore I see that the PHA data format requires an EXPOSURE keyword (the integration time (in seconds) for the PHA data, assumed to be corrected for deadtime, data drop-outs etc.). So this is a kind of livetime. I think we should add this attribute to the `GPha` class, and then the storage of `ontime` and `offtime` is not longer needed. By the way I see

```
m_ontime = obs.livetime();  
m_offtime = obs.livetime();
```

which means that you store in fact exposure as `ontime`, which is not very correct.

**#19 - 04/28/2016 12:34 AM - Knödlseeder Jürgen**

I pushed a branch with the current code that has been rebased on devel into the gammalib repo: 1044-implement-on-off-fitting

Before now merging in the code, could you add a unit test so that we are sure that we test the likelihood fitting when running make check?

**#20 - 05/30/2016 10:22 PM - Knödlseeder Jürgen**

- Precedes Action #1781: Finish implementation of on-off fitting added

**#21 - 05/30/2016 11:36 PM - Knödlseeder Jürgen**

- Status changed from In Progress to Closed

Merge the branch now in the trunk to prevent further divergence, but there are still pending issues that should be resolved as soon as possible (see #1781).

**#22 - 10/06/2016 05:15 PM - Brun Francois**

- File test\_OnOff\_latest.py added

- File pull\_OnOff\_latest.py added

Update of the test files to work with the latest version of the software.

**Files**

pull_OnOff.py	23.9 KB	03/11/2016	Martin Pierrick
test_OnOff.py	22.1 KB	03/11/2016	Martin Pierrick
test_OnOff_latest.py	22.7 KB	10/06/2016	Brun Francois
pull_OnOff_latest.py	24.8 KB	10/06/2016	Brun Francois