

## GammaLib - Action #540

Feature # 912 (Rejected): Implement an ON-OFF fitter class

### Preliminary discussion

10/10/2012 01:19 PM - Deil Christoph

<b>Status:</b>	Closed	<b>Start date:</b>	06/23/2013
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Deil Christoph	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	HESS sprint #1		

#### Description

### Introduction

In TeV data analysis it is very common to use on-off analysis methods, i.e. to choose an on regions that contains signal and background, an off region that only contains background, and then to compute the excess as  $n_{\text{on}} - \alpha * n_{\text{off}}$ , where alpha is the ratio of background exposures of the on and off region. See [Berge et al. 2007](#) .

The major difference to the current analysis method in gammalib (which is the Fermi LAT method) is that the PSF shape is not used, instead a relatively large region is chosen that contains the most of the signal. On-off is less sensitive, but more robust in case the PSF shape is not well known, which is the case for HESS.

### Statistics for one bin

The significance of such an excess is usually computed using the [Li & Ma formula](#) and at least in the HESS software the excess errors are also computed using profile likelihood on the "Li & Ma likelihood" (the code is hard to understand and not commented well, I have to have a close look).

Upper (an lower) limits are computed using the [Rolke](#) profile likelihood method as implemented in the [ROOT TRolke](#) class.

The first part of this project would be to implement these methods in gammalib.

I see there is already gammalib/src/numerics and gammalib/src/support, but I think a new folder gammalib/src/stats would be the right place?

### Statistics for many bins

When fitting many bins (e.g. energy bins for spectra or spatial bins for maps) in the HESS software, two methods are implemented:

1. Compute the likelihood  $P(n_{\text{on}} | \text{background})$ , where  $\text{background} = \alpha * n_{\text{off}}$
2. Compute the likelihood  $P(n_{\text{on}}, n_{\text{off}} | \alpha, n_{\text{gamma}}, n_{\text{hadron}})$

The second method takes that fact that  $n_{\text{off}}$  has statistical fluctuations into account and has to be used when there are bins only few  $n_{\text{off}}$  counts. A very nice implementation and documentation can be found e.g. in [Mathieu de Naurois's OnOffFitter class](#) I'll ask him for permission to modify / include it in gammalib.

### On-off containers and fitter interface in the HESS software

In HAP the basic output of HAP are these objects:

- BgStats contains a few numbers:  $n_{\text{on}}$ ,  $n_{\text{off}}$ ,  $\text{acceptance}_{\text{on}}$ ,  $\text{acceptance}_{\text{off}}$ ,  $\text{gamma}_{\text{exposure}_{\text{on}}}$  (plus histograms for zenith angle distribution and other cruft)
- BgMaps contains a few 2-dimensional ROOT histograms:  $n_{\text{on}}$ ,  $n_{\text{off}}$ ,  $\text{acceptance}_{\text{on}}$ ,  $\text{acceptance}_{\text{off}}$ ,  $\text{gamma}_{\text{exposure}_{\text{on}}}$
- Spectrum contains a few 1-dimensional ROOT histograms:  $n_{\text{on}}$ ,  $n_{\text{off}}$ ,  $\text{acceptance}_{\text{on}}$ ,  $\text{acceptance}_{\text{off}}$ ,  $\text{gamma}_{\text{exposure}_{\text{on}}}$

So in HESS we have decided to have the on-off containers to be

Container  
- array  $n_{\text{on}}$   
- array  $n_{\text{off}}$

...

instead of

Container

- array<OnOffStats>

with

OnOffStats

- int n\_on

- int n\_off

...

All containers have then methods to get OnOffStats like `spectrum.GetStats(bin=42).GetSignificance()` or `bgmaps.GetSignificanceMap(oversample_radius=0.1 deg)`.

## Tools

I'll make a separate issue for end-user tools to actually perform on-off analyses, i.e. to fill and fit OnOffContainers.

What I have in mind is write tools to produce a map with the ring background method and a spectrum with the reflected background method in ctools. These would be the use / test cases while developing the on-off library code.

## On-off analysis interface in gammalib

It is not clear to me how to best integrate on-off analysis in gammalib.

Please comment!

Presumably new classes OnOffStats, OnOffSpectrum, OnOffMap, OnOffCube are added that contain floats / Spectrum / Map / Cube objects for n\_on, n\_off, ... similar to the HESS software?

Given some models they then know how to compute expected counts for each bin and they know how to pass their data to the OnOffFitter.

I guess the goal should be to avoid copies and boilerplate code, but i'm not sure it's possible.

## History

### #1 - 10/10/2012 08:48 PM - Knödlseder Jürgen

- Target version set to HESS sprint #1

### #2 - 10/10/2012 09:52 PM - Knödlseder Jürgen

Christoph Deil wrote:

The major difference to the current analysis method in gammalib (which is the Fermi LAT method) is that the PSF shape is not used, instead a relatively large region is chosen that contains the most of the signal. On-off is less sensitive, but more robust in case the PSF shape is not well known, which is the case for HESS.

Are you sure about this? It would be interesting to study how much the actual knowledge of the PSF does indeed affect a maximum likelihood fit. The main effect will probably come from an incorrect estimation of the PSF size, but you suffer from this effect also for region analyses, unless you chose an area that is substantially larger than the PSF. In that case, the region analysis would however be less sensitive than the maximum likelihood analysis.

In any case, once we have both methods in GammaLib, we may be able to decide upon this question :-)

The significance of such an excess is usually computed using the [Li & Ma formula](#) and at least in the HESS software the excess errors are also computed using profile likelihood on the "Li & Ma likelihood" (the code is hard to understand and not commented well, I have to have a close look).

Upper (an lower) limits are computed using the [Rolke](#) profile likelihood method as implemented in the [ROOT TRolke](#) class.

The first part of this project would be to implement these methods in gammalib.

I see there is already gammalib/src/numerics and gammalib/src/support, but I think a new folder gammalib/src/stats would be the right place?

Lets put them into the numerics module, otherwise we will end up having a model per function type. The numerics module is intended for any kind of numerical support classes, so I guess this is well suited.

By the way: you may recognize that there is GIntegrand and GFunction, which both provide a general description for a function. Ultimately, GIntegrand should be replaced by GFunction (see feature #40), so any new numerical method operating on a function should directly use GFunction.

It is not clear to me how to best integrate on-off analysis in gammalib.

Please comment!

Presumably new classes OnOffStats, OnOffSpectrum, OnOffMap, OnOffCube are added that contain floats / Spectrum / Map / Cube objects for `n_on`, `n_off`, ... similar to the HESS software?

Given some models they then know how to compute expected counts for each bin and they know how to pass their data to the OnOffFitter. I guess the goal should be to avoid copies and boilerplate code, but i'm not sure it's possible.

In principle one could store a HESS counts spectrum in GCTAEventCube, so the skymap part of the cube would here be a single pixel. But then we have to think how to deal with the acceptance corrections, etc.

Maybe the best would be to start writing down the relevant mathematics in a small document. This would help to understand what the essential operations are, and how they should best be factorized in the computation.

### #3 - 10/10/2012 10:21 PM - Deil Christoph

Jürgen Knödseder wrote:

Christoph Deil wrote:

The major difference to the current analysis method in gammalib (which is the Fermi LAT method) is that the PSF shape is not used, instead a relatively large region is chosen that contains the most of the signal. On-off is less sensitive, but more robust in case the PSF shape is not well known, which is the case for HESS.

Are you sure about this? It would be interesting to study how much the actual knowledge of the PSF does indeed affect a maximum likelihood fit. The main effect will probably come from an incorrect estimation of the PSF size, but you suffer from this effect also for region analyses, unless you chose an area that is substantially larger than the PSF. In that case, the region analysis would however be less sensitive than the maximum likelihood analysis.

In any case, once we have both methods in GammaLib, we may be able to decide upon this question :-)

No, I'm not sure how big the effects are, but I want to be able to easily simulate this.  
We don't have very nice tools in HAP to simulate data, no position-energy cubes, no unbinned likelihood fit.  
Having ctobssim and ctlike is one of the major motivations for me to learn / work on gammalib / ctools. :-)

Presumably new classes OnOffStats, OnOffSpectrum, OnOffMap, OnOffCube are added that contain floats / Spectrum / Map / Cube objects for n\_on, n\_off, ... similar to the HESS software?  
Given some models they then know how to compute expected counts for each bin and they know how to pass their data to the OnOffFitter.  
I guess the goal should be to avoid copies and boilerplate code, but i'm not sure it's possible.

In principle one could store a HESS counts spectrum in GCTAEventCube, so the skymap part of the cube would here be a single pixel. But then we have to think how to deal with the acceptance corrections, etc.

Maybe the best would be to start writing down the relevant mathematics in a small document. This would help to understand what the essential operations are, and how they should best be factorized in the computation.

I'll attach Mathieu's code (quite complicated C++) and doxygen documentation, which is a really nice writeup.  
If needed I can also attach his documentation for morphology and spectral fitting methods.  
Let me know if you want more info on some parts or still think a separate document is needed (trying to avoid writing it because that would take me a few days).

#### **#4 - 10/11/2012 02:12 PM - Deil Christoph**

- File *Mathieu\_OnOffFitter.pdf* added
- File *Mathieu\_Spectrum\_Principles.pdf* added
- File *Mathieu\_SpectrumFitter.pdf* added

Here's the documentation by Mathieu on his OnOffFitter and how it's used to fit e.g. spectra.

#### **#5 - 10/15/2012 03:40 PM - Knödseder Jürgen**

Here the place where the evaluation of the likelihood function is implemented in GammaLib:

source:src/obs/GObservations\_optimizer.cpp#L182

Look at the method poisson\_binned(): source:src/obs/GObservations\_optimizer.cpp#L619

This is the binned likelihood evaluation. A variant of the method is needed for on-off fitting.

I started a Wiki page for the on-off fitting implementation: [\[\[CTA on-off fitting\]\]](#).

**#6 - 05/02/2013 04:32 PM - Deil Christoph**

A while back I wanted to implement the Rolke and FeldmanCousins and Li&Ma methods in GammaLib. This is needed to compute significance, errors, sensitivity, limits for flux points (i.e. for one on-off measurement). I did start but then didn't have time to finish it.

For Rolke and FeldmanCousins there are standalone C++ versions extracted from the ROOT package here:

[https://github.com/cdeil/root\\_poisson\\_stats](https://github.com/cdeil/root_poisson_stats)

These are not easy to implement and I think could maybe be simply included in GammaLib, but not be part of the public API?

Probably for the public API there should be functions like here:

<http://nbviewer.ipython.org/5502591>

or an OnOffMeasurement class (similar to the attached SpectrumStats class from the HESS software).

Just as a reference for myself about the status, I attach some existing code to this issue from the HESS software.

**#7 - 05/02/2013 04:37 PM - Deil Christoph**

- File *SpectrumStats.hh* added

- File *SpectrumStats.C* added

- File *Spectrum.hh* added

- File *BgMaps.hh* added

Spectrum and BgMaps are basically OnOffMeasurement containers, where Spectrum is 1D in log(energy) for spectral analysis and BgMaps is for 2D maps and used for morphology analysis.

It's not nice that there is no abstract OnOffMeasurements container and the on-off fitting is implemented twice in HAP. In ParisAnalysis this is done better I think and I've put the ParisAnalysis OnOffFitter code in some issue already.

**#8 - 05/02/2013 05:34 PM - Deil Christoph**

- File *BgStats.hh* added

**#9 - 06/23/2013 04:16 PM - Martin Pierrick**

- Tracker changed from *Feature* to *Action*

- Subject changed from *Add on-off analysis methods* to *Preliminary discussion*

- Status changed from *New* to *Resolved*

- Start date set to *06/23/2013*

- % Done changed from *0* to *100*

- Estimated time set to *0.00*

- Parent task set to *#912*

- Position deleted (*48*)

**#10 - 06/23/2013 10:23 PM - Knödlseeder Jürgen**

- Status changed from *Resolved* to *Closed*

**Files**

---

Mathieu_OnOffFitter.pdf	535 KB	10/11/2012	Deil Christoph
-------------------------	--------	------------	----------------

Mathieu_Spectrum_Principles.pdf	165 KB	10/11/2012	Deil Christoph
Mathieu_SpectrumFitter.pdf	478 KB	10/11/2012	Deil Christoph
SpectrumStats.hh	2.98 KB	05/02/2013	Deil Christoph
SpectrumStats.C	10.8 KB	05/02/2013	Deil Christoph
Spectrum.hh	12.7 KB	05/02/2013	Deil Christoph
BgMaps.hh	8.43 KB	05/02/2013	Deil Christoph
BgStats.hh	6.62 KB	05/02/2013	Deil Christoph