

ctools - Action #3436

Analysis with alternating ON & OFF runs

11/05/2020 02:16 PM - Lemoine Marianne

Status:	Closed	Start date:	11/05/2020
Priority:	Normal	Due date:	
Assigned To:	Tibaldo Luigi	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	2.0.0		
Description			
For very extended and diffuse sources such as halos, the background models have to be very precise. To suppress some of these systematics related to background modeling, a classical approach was the ON/OFF observing mode. In this mode observations runs centred on the target are interspersed with equal-length observations of an empty field at equal zenith angle. The background is assumed to be equal in the two runs, the difference between them provides a measurement of the gamma-ray signal. Such approach is currently missing in the ctools and would be very welcome.			

History

#1 - 02/01/2021 10:53 AM - Knödseder Jürgen

It should be checked whether such an analysis can actually be emulated using the existing tools, running for example csphagen on On and Off runs and combining the results for the analysis. If possible, one could simple write a cscript that performs the necessary steps.

What I have in mind is running csphagen twice, once on the ON observations and once on the OFF observations, and then using the ON PHA file of the OFF observations as OFF PHA file for the analysis. Maybe some fiddling with the files is needed, and the XML file has probably to be created "by hand", and these steps, as well as the csphagen runs, could be integegrated into a new cscript. Probably also some code is needed to determine the ON region for the OFF observations.

#2 - 02/02/2021 11:03 AM - Tibaldo Luigi

After discussion with Marianne it looks like the alpha factors need to take into account optical efficiency, that is only possible to do via the (IRF) background model.

Therefore, it seems that this requires some modification at the level of GCTAOnOffObservation.

Below I outline my proposal.

In GCTAOnOffObservation:

- modify the private methods set and compute_alpha so that they always take in input two observations and two model containers (one for On and one for Off)
- add one more constructor with two observations and two model containers; in the two constructors the observations and model containers will either be set to be the same (same observation for On and Off) or to two different ones.

In csphagen:

- add the option OFF to the parameter bkgmethod; when this is invoked, query for an Off observation container (must be same size as On observation container)
- internally the script will have a new member variable self.obs_off to store the dedicated Off observations, set to None unless the method OFF is used
- add dedicated method to retrieve the sky region in the Off observation that correspond to the same camera region as the On region in the On observation, call this method in set_background_regions (implemented in csphagen to keep the GCTAOnOffObservation methods as general as possible)
- implement in process_observation the case bkgmethod=OFF

#3 - 02/02/2021 11:13 AM - Tibaldo Luigi

The new constructor for GCTAOnOffObservation needs to be added also in the Python extension GCTAOnOffObservation.i

#4 - 02/02/2021 03:28 PM - Knödlseher Jürgen

Looks good. I'm however wondering why you need two model containers? The source model in the model container is only used for ARF computation. For the background models we can use the observation ID to assign a given background model to a given observation.

#5 - 02/02/2021 03:40 PM - Tibaldo Luigi

- Status changed from New to In Progress

thanks Jürgen, this being clarified I'll start with the implementation

#6 - 02/02/2021 06:42 PM - Tibaldo Luigi

- % Done changed from 0 to 10

I modified the compute_alpha and set methods so that they accept in input different observations for On and Off. All the test in GammaLib and ctools are passed. Next I should implement the GCTAOnOffObservation constructor that takes different observations for On and Off.

Things to check or do to be added to my list.

- Understanding what the member variables m_ontime, m_livetime, and m_deadc are used for. For the moment I set them based on the On observations. They do not seem to be used anywhere. The On and Off spectra are assigned exposures based on the respective livetimes and they seem the only information used.
- The N_bgd method uses the On spectrum exposure, double check, but I think it should be using the Off one;
- Check the computation of stacked background spectra and alpha factors, which exposures are used etc.

#7 - 02/02/2021 10:06 PM - Knödlseher Jürgen

I think it's correct that the Off spectrum exposure should be used in N_bgd. I probably used on the On spectrum exposure since in the current code On and Off spectra have the same exposure.

#8 - 02/03/2021 10:59 AM - Tibaldo Luigi

- % Done changed from 10 to 40

Work on GCTAOnOffObservation completed

- methods set and compute_alpha and stacked observations constructor generalized so that On and Off observations can be different
- fixed N_bgd so that it uses the Off exposure (now it can be different)
- added new constructor that builds On/Off observation from different On and Off observations (including unit test)
- member variables m_ontime, m_livetime, and m_deadc are set based on the On observation (specified in header file), currently they are not used for the computations

Now I will move on to cspshagen to create a user interface.

#9 - 02/04/2021 11:46 AM - Tibaldo Luigi

- Assigned To set to Tibaldo Luigi

- % Done changed from 40 to 70

I finished the implementation in csphagen + reference manual and unit test. In the process I proceeded also to three modifications

- the bkgregmin parameter is now reserved for the reflected background estimation
- the maxoffset parameter is applied to all background estimation methods
- the stack parameter is queried only when there are multiple observations

I will now proceed to more in-depth testing.

#10 - 02/04/2021 12:33 PM - Tibaldo Luigi

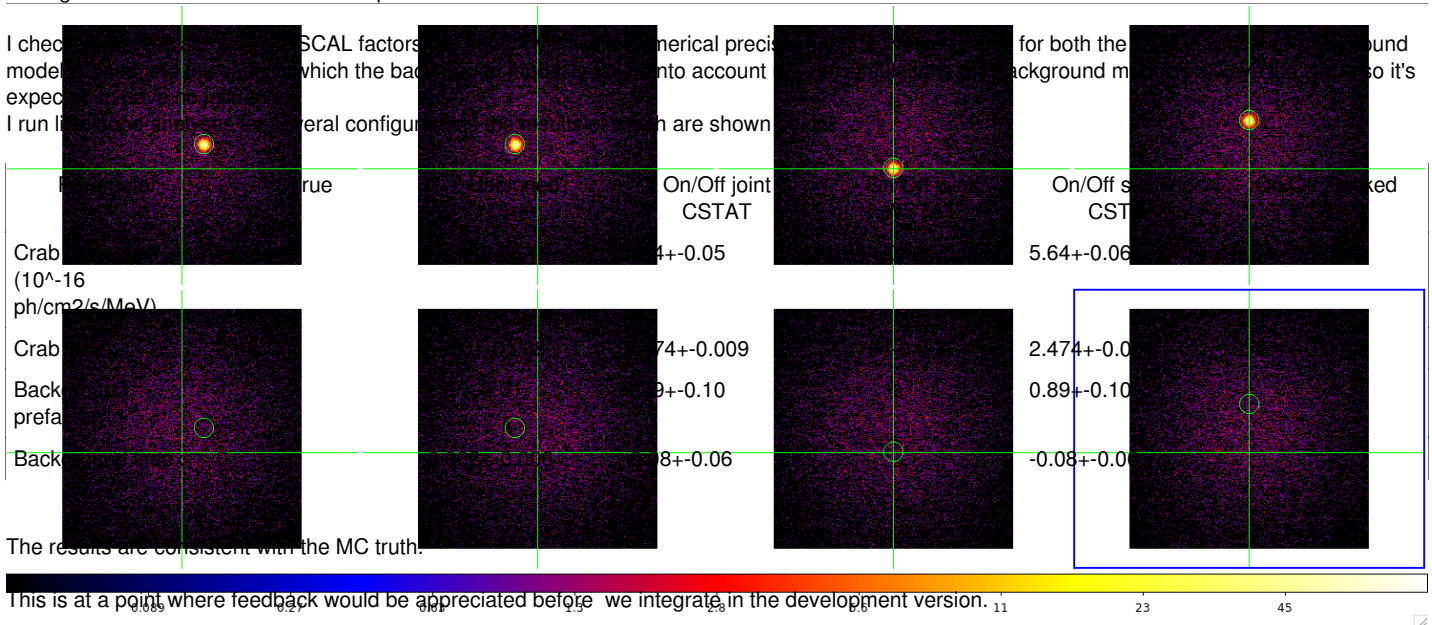
- File *crab_off.png* added

- Status changed from *In Progress* to *Feedback*

- % Done changed from 70 to 100

I performed a simple test on simulated data. I generated 4 simulated observations of a Crab-like source plus 4 simulated empty field observations. All observations last 30 minutes. Pointing directions and deadtime corrections are varied for each observation individually.

I proceeded with the analysis using the OFF method. Below you can see count maps in camera coordinates for the 8 observations. Top row is the source observations plus the On region, bottom row the empty-field observations plus Off regions determined by csphagen which are matching the On region in camera coordinates as required.



#11 - 02/04/2021 04:43 PM - Tivaldo Luigi

I had forgotten that energy dispersion is disabled by default in simulations and 3D analyses, but is always enabled in On/Off analyses. I repeated my test enabling it everywhere.

Parameter	True	Unbinned	On/Off joint CSTAT	On/Off joint WSTAT	On/Off stacked CSTAT	On/Off stacked WSTAT
Crab prefactor (10 ⁻¹⁶ ph/cm2/s/MeV)	5.7	5.76+-0.06	5.76+-0.06	5.76+-0.06	5.76+-0.06	5.76+-0.06
Crab index	2.48	2.471+-0.008	2.470+-0.008	2.472+-0.009	2.470+-0.008	2.471+-0.009
Background prefactor	1	1.00+-0.01	0.86+-0.10	-	0.86+-0.10	-
Background index	0	-0.001+-0.006	-0.11+-0.06	-	-0.11+-0.06	-

Now the results from unbinned analysis and all the flavours of On-Off analyses are almost identical.

#12 - 02/05/2021 02:16 PM - Tivaldo Luigi

- Status changed from Feedback to Pull request

#13 - 02/05/2021 03:14 PM - Knödlseher Jürgen

I looked into the GammaLib code. In GCTAOnOffObservation::compute_alpha I see:

```
// Get CTA observation pointing direction, zenith, and azimuth
GCTAPointing obs_onpnt = obs_on.pointing();
GCTAPointing obs_offpnt = obs_on.pointing();
```

I guess this code should be

```
// Get CTA observation pointing direction, zenith, and azimuth
GCTAPointing obs_onpnt = obs_on.pointing();
GCTAPointing obs_offpnt = obs_off.pointing();
```

Could you please check?

#14 - 02/05/2021 03:19 PM - Tbaldo Luigi

Yes, absolutely, thank you for catching this! I'm actually puzzled that this did not show up in the test I have described above :O

#15 - 02/05/2021 03:39 PM - Tbaldo Luigi

While checking why the error you found did not show up in my test (still not understood), I found another problem (missing statement, would have ignored exclusion regions for the OFF method). I pushed a fix to my branch

#16 - 02/05/2021 04:29 PM - Knödseder Jürgen

user#266 wrote:

While checking why the error you found did not show up in my test (still not understood), I found another problem (missing statement, would have ignored exclusion regions for the OFF method). I pushed a fix to my branch

On the GammaLib side or the ctools side?

#17 - 02/05/2021 05:07 PM - Tbaldo Luigi

ctools (in csphagen)

#18 - 02/05/2021 05:11 PM - Knödseder Jürgen

user#266 wrote:

ctools (in csphagen)

The if statement that you added at 15:38?

```
# ... otherwise store region in container
if is_valid:
    regions.append(region)
```

This was already in the code that I checked out for integration.

#19 - 02/05/2021 05:12 PM - Tbaldo Luigi

Yes, thanks

#20 - 02/05/2021 05:30 PM - Knödseder Jürgen

- Status changed from Pull request to Closed

- Target version set to 2.0.0

Good. I merged the code into devel (now both GammaLib and ctools).

Files

crab_off.png	200 KB	02/04/2021	Tibaldo Luigi
--------------	--------	------------	---------------