

ctools - Change request #2976

Improve procedure to deal with exclusion regions in reflected background generation in csphagen

07/25/2019 04:51 PM - Tbaldo Luigi

| | | | |
|---|--------------|------------------------|------------|
| Status: | Closed | Start date: | 07/25/2019 |
| Priority: | Normal | Due date: | |
| Assigned To: | Tbaldo Luigi | % Done: | 100% |
| Category: | | Estimated time: | 0.00 hour |
| Target version: | 1.7.0 | | |
| Description | | | |
| Currently the reflected regions are placed (independently of the exclusion regions), and afterward Off regions overlapping with exclusion regions are rejected. This may end up with cases in which many (all) Off regions are discarded because they overlap partially with exclusion regions, while placing the Off regions with prior knowledge of where the exclusion regions are would lead to higher background statistics. | | | |

History

#1 - 09/19/2019 03:49 PM - Tbaldo Luigi

- Status changed from New to In Progress
- Assigned To set to Tbaldo Luigi

Duplicates #2886

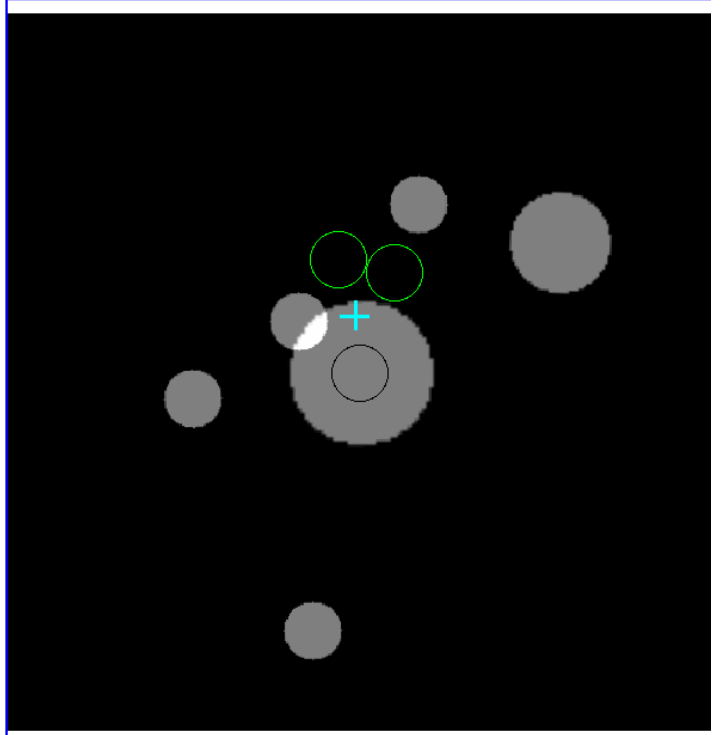
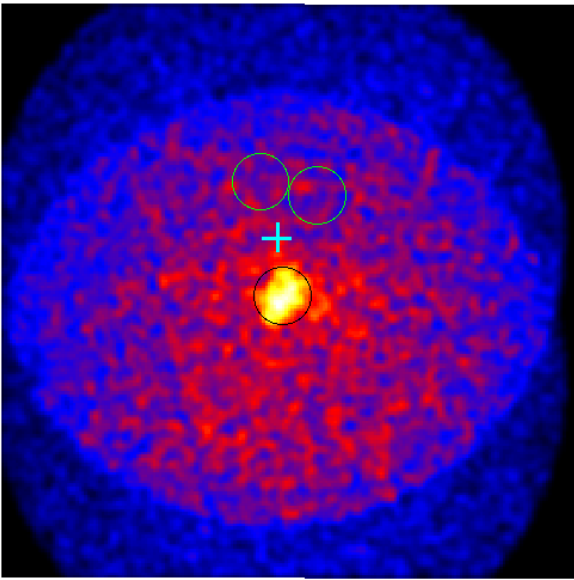
#2 - 01/16/2020 05:25 PM - Tbaldo Luigi

- File `off_new1.png` added
- % Done changed from 0 to 20

I am testing a new procedure in which rather than 1) pre-define the Off regions, 2) reject those overlapping with exclusion regions I place the Off regions one by one and check immediately whether the latest one overlaps with the exclusion region, and if not a new tentative exclusion region is placed with a shift in position angle of 10% of the size of an Off region.

For the test I'm using the HESS public observations of MSH 15-52 and an exclusion region provided by Andreas Specovius. I'm using a circular On region centered at RA = 228.547, Dec = -59.174, and with a radius of 0.2 deg. The current reflected-region method finds at least 2 Off regions only for 9 observations out of the 20 available. The remaining 11 observations (pointed North of the source) are therefore rejected (they have one Off region not overlapping with the exclusion region).

For the new algorithm I'm testing I find ≥ 2 Off regions for all 20 observations. The image below shows how the Off regions are distributed in this case. Left panel: count map. Right panel: exclusion region. In both panels overlaid: cyan cross - the pointing direction to the North of the source, black circle - On region, green circles - Off regions found by the new algorithm.



#3 - 01/17/2020 09:58 AM - Tibaldo Luigi

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

- % Done changed from 20 to 40

I verified that for the simpler test cases in test_csphagen.py the modifications leave the number of Off regions unchanged:

- the Off regions are exactly the same for the cases with no exclusion regions;

- the number of Off regions is the same with a slight shift in position for the case with a simple exclusion region.

Awaiting for feedback from Andreas to see if the improvement can be considered sufficient or there are other test cases to consider.

#4 - 01/20/2020 12:27 PM - Specovius Andreas

Nice to hear that your approach works so well!

I am not sure how to understand "one by one", do you scan the position angle parameter starting from the ON-region and place the corresponding regions as soon as they are free of overlap?

#5 - 01/20/2020 12:29 PM - Tibaldo Luigi

Exactly.

Are there any other test cases you would recommend to check?

#6 - 02/03/2020 11:53 AM - Tibaldo Luigi

- Status changed from *Feedback* to *Pull request*

- % Done changed from 40 to 100

I think the current changes ready to be merged into devel.

Branch 2976-reflreg

#7 - 02/10/2020 09:54 AM - Knödseder Jürgen

- Status changed from *Pull request* to *Closed*

Merged into devel.

Files

| | | | |
|--------------|--------|------------|---------------|
| off_new1.png | 118 KB | 01/16/2020 | Tibaldo Luigi |
|--------------|--------|------------|---------------|