

GammaLib - Change request #1789

Add weighting array to CTA counts cube

06/11/2016 07:50 AM - Knödseder Jürgen

<b>Status:</b>	Closed	<b>Start date:</b>	06/11/2016
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assigned To:</b>	Knödseder Jürgen	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.1.0		
<b>Description</b> Add a weighting array to the CTA counts cube so that the event selection for stacked analysis is properly taken into account. The weights of a bin correspond to the relative livetime for a given bin. Partial bin coverage should also be considered, both spatially and spectrally. The weight should then be taken into account when computing the model values. This can be done over the size() method.			
<b>Related issues:</b>			
Related to ctools - Bug # 1791: ctbin results for in-memory / cmd-line workflow		<b>Closed</b>	
Related to GammaLib - Action # 1362: Implement run-wise energy thresholds for...		<b>Closed</b>	<b>11/11/2014</b>

History

#1 - 06/21/2016 11:09 AM - Knödseder Jürgen

- Status changed from New to In Progress
- Assigned To set to Knödseder Jürgen
- Target version set to 1.1.0
- % Done changed from 0 to 80

ctbin now computes the counts cube weights according to the overlaps between the Region of Interests and energy boundaries with the counts cube bins. This allows in particular for a proper handling of different energy thresholds between observations that will be stacked into a single cube. Also different Region of Interests are now properly handled, and there is no longer an issue with counts cubes that are larger than the Region of Interest of the event list. Also the energy margin has been removed, solving potentially #1791 (needs to be checked).

The code is not yet merged in, I want first to do some checking.

#2 - 06/21/2016 11:09 AM - Knödseder Jürgen

- Related to Bug #1791: ctbin results for in-memory / cmd-line workflow added

#3 - 06/21/2016 12:53 PM - Knödseder Jürgen

Here now a summary of the rational of this change, which is a quite major change for the stacked analysis.

Formerly, Region of Interest selections and energy boundary selections have been applied in the cube response calculation to allow the combination of data with different spatial or spectral event selections. Doing this in the response computation was however not correct, since the response is computed for true quantities while the data space selections (RoI, energy boundaries) are measured quantities. Hence problems occurred at the edge of the field of view, and the stacked response was also not fully correct for energy dispersion.

To implement properly the event selection, a cube of weights has now been added to the counts cube FITS file (and the GCTAEventCube class). If a counts cube bin is fully selected by all observations in a stacked analysis, the weight for this cube bin is unity. If however the cube bin is only filled by a subset of observations, or if the cube bin is only partially filled because the minimum energy of the event list falls in the middle of an counts cube energy bin, the weight will be smaller than unity.

ctbin now correctly fills the weight cube according to the data space selection, including RoI selection and energy selection. For RoI selection the tool now checks what fraction of the bin lies within the RoI, and sets the weight accordingly. For energy selection, the tool assumes that the event distribution within a bin follow a power law with index of -2. This is definitely an assumption, but as long as sufficient energy bins are used, the assumption should not be too bad.

**#4 - 06/21/2016 02:05 PM - Knödlseider Jürgen**

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

- % Done changed from 80 to 100

Merged into devel.

**#5 - 06/21/2016 02:39 PM - Mayer Michael**

Great! I guess this is somewhat related to what we discussed already in #1362?

How does that play together with the stacked responses? Do they need to be weighted then as well?

**#6 - 06/21/2016 03:30 PM - Knödlseider Jürgen**

The stacked responses are not changed (except that ROI and energy selections are no longer applied).

The way how the weights are implemented is through the `GCTAEventBin::size()` method. This method returns the size of a counts cube bin in units of  $\text{MeV s sr}$ , and now is in addition multiplied by the weight. In that way, all models that are computed in the binned likelihood get correctly weighted (in the binned likelihood, the model is multiplied by `GCTAEventBin::size()`).

I agree that this fixes also #1362.

**#7 - 06/21/2016 03:30 PM - Knödlseider Jürgen**

- Related to Action #1362: Implement run-wise energy thresholds for stacked analysis added