

GammaLib - Feature #1721

Implement energy dispersion for stacked analysis

03/01/2016 12:10 PM - Mayer Michael

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
What needs to be done is the following:			
<ul style="list-style-type: none">• Create GCTACubeEdisp• Extend GCTAResponseCube to use edisp if available and instructed• Implement GCTAResponseCube::ebounds() method• Write ctedispcube tool• Add "edispcube" parameter to all ctools and cscripts that support stacked analysis			

History

#1 - 03/01/2016 01:25 PM - Mayer Michael

We should also think about the interface for e.g. ctlike:

The user should be able to specify "NONE" for the edispcube without running into problems.

There are four combinations possible e.g. in ctlike:

- edisp=no and edispcube = "NONE" (do we query for edispcube if edisp=no?)
- edisp=yes and edispcube = "NONE" (do we throw an exception or do we silently continue without energy dispersion?)
- edisp=yes and edispcube = "edispcube.fits" (Clear: run ctlike including energy dispersion)
- edisp=no and edispcube="edispcube.fits" (Edisp shouldn't be used but cube should be loaded in order to have the GCTAResponseCube complete.)

#2 - 03/03/2016 09:51 PM - Mayer Michael

- Status changed from New to Feedback

- % Done changed from 0 to 70

I have added the gammalib support for energy dispersion in stacked mode on branch *1721-edisp-stacked*.

- The class GCTACubeEdisp has been added.
- The function GCTACubeEdisp::ebounds_src can still be improved as it does not take into account the spatial position inside the map (due to the interface of GCTACubeResponse::ebounds()).
- GCTACubeResponse now handles the energy dispersion as optional IRF.
- GCTACubeSourceDiffuse@ now holds a GNodeArray that stores the energy bins. This is necessary for interpolation when using energy dispersion in GCTACubeResponse::irf_diffuse().
- A rudimentary unit test has been added and make check runs successful.

We will need to focus on the ctools branch now to use the new functionality in gammalib (#1724).

#3 - 03/04/2016 03:27 PM - Mayer Michael

- *Status changed from Feedback to Pull request*

- *% Done changed from 70 to 100*

Let's try to merge it in. We probably have to make some in-depth tests on real data. I didnt have the time to do so yet.

#4 - 03/06/2016 10:36 PM - Knödseder Jürgen

- *Status changed from Pull request to Closed*

I merged the code into devel.

I changed a bit the interface to GCTACubeEdisp, and in particular I replaced the GEbounds definition of the cube by a GEnergies definition. In fact, for the stacked cube responses we do not really use energy boundaries but evaluate the response at given node energies. To reflect this it is thus better to use GEnergies and not the boundaries for which we compute the mean energies for evaluation.

We should also change this for the GCTACubeExposure and GCTACubePsf classes.

And also the folding of 2 dimensions into a single sky map dimension is also not very elegant. It would be better that have a real 4D skymap (the same is true for GCTACubePsf).

I propose thus to implement a GTuple class (see #1733) and use this to define additional sky map dimensions.