GammaLib - Action #1285

Handle average response cubes

07/18/2014 02:10 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	07/18/2014
Priority:	High	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	8.00 hours
Target version:	00-09-00		

Description

Binned analysis should in the future always make use of the average response cubes, no pointing information should be used any longer. Response information will be stored using the GCTAExposure and GCTAMeanPsf classes.

This action is for implementing the use of average response information.

History

#1 - 07/18/2014 02:29 PM - Knödlseder Jürgen

The following methods need to be modified:

- GCTAResponse::irf
- GCTAResponse::irf_radial
- GCTAResponse::irf_elliptical
- GCTAResponse::irf_diffuse

The GCTAResponse::irf should use the following code for average response handling:

```
// Retrieve average response references
const GCTAExposure& exposure = ... (where ever we get the exposure from)
const GCTAMeanPsf&  psf = ... (where ever we get the PSF from)

// Get photon attributes
const GSkyDir& srcDir  = photon.dir();
const GEnergy& srcEng  = photon.energy();
const GTime&   srcTime = photon.time();

// Compute only if we're sufficiently close to PSF
if (delta <= delta_max) {

    // Get effective area component
    irf = exposure(srcDir, srcEng);

    // Multiply-in PSF
    if (irf > 0.0) {

        // Get PSF component
        irf *= psf(srcDir, delta, srcEng);

        // Apply deadtime correction
        irf *= obs.deadc(srcTime);

    } // endif: exposure was non-zero

} // endif: we were sufficiently close to PSF
```

The GCTAResponse::irf_radial, GCTAResponse::irf_elliptical and GCTAResponse::irf_diffuse methods need a more fundamental re-organisation as the integrations are performed in a system centred on the pointing, hence all coordinate transformations are pointing dependent.

#2 - 07/18/2014 02:34 PM - Knödlseder Jürgen

Maybe the best thing would be to introduce a new class, called for example GCTACubeResponse, that handles the cube-style response. It would then be upon reading (e.g. GCTAObservation::read) where the response type would be defined.

#3 - 07/18/2014 04:48 PM - Knödlseder Jürgen

Looks like to way forward is to create a GCTAResponse abstract base class and to derived classes:

- GCTAResponseIrf (which is the old GCTAResponse)
- GCTAResponseCube (which is the new cube-style IRF)

I'm working now on refactoring the code ...

#4 - 07/19/2014 12:42 AM - Knödlseder Jürgen

- Status changed from New to In Progress

- % Done changed from 0 to 90

I now refactored the CTA response function and implemented a new GCTAResponseCube response function. On reading an XML file, GCTAObservation now automatically sets the relevant response function, depending on the parameter names that are found in the XML file. For cube-style analysis, the XML definition is

```
<observation_list title="observation library">
  <observation name="Crab" id="00001" instrument="CTA">
    <parameter name="CountsMap"    file="crab_cntmap.fits"/>
    <parameter name="ExposureCube" file="expcube.fits"/>
    <parameter name="PsfCube"      file="psfcube.fits"/>
  </observation>
</observation_list>
```

Unit tests have been added to test_CTA that demonstrate that the cube-style analysis is working, at least for point sources. For extended or diffuse sources, the response methods are not yet implemented.

#5 - 07/19/2014 02:02 AM - Knödlseder Jürgen

- Status changed from In Progress to Resolved

- % Done changed from 90 to 100

Now also adapted ctools. This action can now been closed, although more unit tests should be added and more tests should be done whether all tools work as expected.

#6 - 07/20/2014 11:20 PM - Knödlseder Jürgen

- Status changed from Resolved to Closed

- Remaining (hours) changed from 8.0 to 0.0