

GammaLib - Action #1250

Create GCTAMeanPsf class

07/08/2014 03:37 PM - Knödlseeder Jürgen

Status:	Closed	Start date:	07/08/2014
Priority:	Normal	Due date:	
Assigned To:	Lu Chia-Chun	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	3rd coding sprint		
Description <p>The GCTAMeanPsf class should implement the mean PSF in form of a histogram as function of Right Ascension or Galactic longitude, Declination or Galactic latitude and log10 energy (4D cube). The class may use the GSkymap class as storage container for the information. Log10 energy and offset angle can be implemented as nested array of the 3rd GSkymap dimension.</p> <p>The mean PSF is an exposure weighted average of the PSF. The typical interface should look like this:</p> <pre>// Constructors and destructors GCTAMeanPsf(void); GCTAMeanPsf(const GCTAMeanPsf& psf); GCTAMeanPsf(const GObservations& obs, const GSkymap& map, const double& min, const double& max, const int& nbins); virtual ~GCTAMeanPsf(void); // Operators GCTAMeanPsf& operator=(const GCTAMeanPsf& psf); // Methods void clear(void); GCTAMeanPsf* clone(void) const; void load(const std::string& filename); void save(const std::string& filename); std::string print(const GChatter& chatter = NORMAL) const; protected: // Methods void init_members(void); void copy_members(const GCTAMeanPsf& psf); void free_members(void); // Data GSkymap m_cube; //!< Average PSF cube</pre> <p>The number of maps in a GSkymap object is nebin*nbins, idelta + ie * nbins can then be the access formula.</p>			

History

#1 - 07/08/2014 03:37 PM - Knödlseeder Jürgen

- Assigned To set to Lu Chia-Chun
- Target version set to 3rd coding sprint

#2 - 07/09/2014 05:50 PM - Lu Chia-Chun

- % Done changed from 0 to 50

GCTAMeanPsf class is built and tested.
The 68% containment radius looks reasonable and the comparison of 68% containment with rsp.psf() shows the same result.

#3 - 07/10/2014 02:20 PM - Knödlseider Jürgen

- *Status changed from New to In Progress*
- *% Done changed from 50 to 60*

I merged the actual version of your GCTAMeanPsf class into trunk after adapting the class interface and making the code compliant with the coding conventions. PSF is now computed as response times exposure, divided by average exposure.

A `set()` method exists to set a PSF cube from a single GCTAObservation, a `fill()` method exists to fill the PSF cube from all CTA observations in an observation container. Writing and saving of the PSF cube works, reading and loading is not yet supported.

I also added unit tests to `test_CTA`.

Still to be done:

- implement `read()` and `load()` methods
- implement PSF cube access operator(s)

#4 - 07/11/2014 12:36 PM - Lu Chia-Chun

- *Status changed from In Progress to Pull request*
- *% Done changed from 60 to 70*

access operator and interpolation implemented.

To do:
`read()`
`load()`

#5 - 07/11/2014 02:16 PM - Knödlseider Jürgen

- *Status changed from Pull request to Resolved*
- *% Done changed from 70 to 100*

Added `load()` and `read()` methods and merged into trunk.

#6 - 07/20/2014 11:21 PM - Knödlseider Jürgen

- *Status changed from Resolved to Closed*
- *Remaining (hours) set to 0.0*