GammaLib - Feature #2919

Load GCTAObservation from self-contained fits file

06/27/2019 03:56 PM - Kelley-Hoskins Nathan

Status:	Rejected	**Start date:**	06/27/2019
Priority:	Normal	**Due date:**	
Assigned To:	Kelley-Hoskins Nathan	**% Done:**	10%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

I've been converting VERITAS data to fits files for the past two years, and I came up with a slightly modified scheme for storing our event lists and IRFs. **I've written a python function that can loaded fits files from this modified format into a GObservations object, and I'd like to add it to gammalib.**

The current schema: Event lists (EVENTS, TELARRAY, and GTI) go in one set of fits files, while IRF tables (POINT SPREAD FUNCTION, EFFECTIVE AREA, ENERGY DISPERSION, BACKGROUND) go in another set (the calibration database). The event list fits file also contains a reference that point to the IRFs in the database.
This feature's additional schema: Each fits file containing an event list also contains whole copies of relevant IRFs and a background model, rather than storing them by reference to an outside calibration database. Essentially each fits file has its own EVENTS, TELARRAY, GTI, POINT SPREAD FUNCTION, EFFECTIVE AREA, ENERGY DISPERSION, and BACKGROUND tables.

This additional schema has a few benefits:

- **Each fits file becomes completely atomic.** Since theres no outside references, it becomes much more intuitive for new users to grasp, and much easier for experienced users to go back to after they've been sitting in the archives for few years.
- There isn't a need for users to set up and keep track of a calibration database, which simplifies starting an analysis.
- A basic analysis can be easily packed and sent over email (for debugging or science): a python notebook/script plus its fits files, without worrying about outside references. Tutorials for new users could also be packaged this way.

The downside of this schema is that, since many data runs use IRFs from the same part of the parameter space (elevation, azimuth, atmosphere, NSB, etc), a group of runs can contain duplicates of some IRFs. This in itself is not bad, but they tend to take up more space. From my conversion of VERITAS data, 100 hours of data in this schema takes up ~5GB, or about **5MB per fits file**. In the regular schema, I think the calibration database takes up about 1GB, and event lists are ~10's of kB, so it's a modest increase.

A long-term problem may be the amount of RAM used by a large analysis. In an extreme example: 10 years of CTA data = 10,000 observing hours (assume its all on one source) = 500GB of fits files loaded into RAM. But this can be fixed by either:

- still having the fits files keep their calibration database reference as a default ("load IRFs from database if available, else load IRFs from fits file"), or
- as each IRF is loaded into RAM: hash the IRF for uniqueness, discard duplicate IRFs and re-associate event lists to point to the matching unique IRFs. This way only the unique IRFs are kept in memory.

In the end: I believe making the fits files more intuitive and reducing the setup complexity outweighs the extra few gigabytes of hard drive space that would be used up, so **I think its at least worth adding a loading function that supports this schema.**

New users who would want to use this feature would do something like:

```
import gammalib
fits  = []
fits += [ '/path/to/1.fits' ]
fits += [ '/path/to/2.fits' ]
fits += [ '/path/to/3.fits' ]
obs = gammalib.GObservations()
obs.load_atomic( fits, bkg='GModelSpectralPowerLaw' )
```

And now obs is now ready to have sky models added and/or be fed to different ctools.

Some practical notes:

- This is just an extra schema that I'd like gammalib to support, no person or observatory would have to convert their data to this

format if they didn't want to
- This feature shouldn't interfere with regular GObservations::load() behavior. "If fits files has 7 tables with the right names, load atomically, otherwise load like normal."
- The function needs to load background models from the fits files. Right now my VERITAS fits files don't specify the spectral component (only the spatial), so theres an argument ('bkg') for specifying the spectral part.

I have already developed and tested a python function to do this loading. I would add this to GObservations.i for now, and maybe later as C++ code in GObservations::load(string).

History

#1 - 06/27/2019 06:28 PM - Kelley-Hoskins Nathan

- % Done changed from 0 to 10

The basic python loader works fine, it seems. Though, the swig file doesnt seem to handle python # comments at all, so theres """comments""" everywhere instead.

#2 - 06/28/2019 04:46 PM - Kelley-Hoskins Nathan

- Status changed from New to Feedback

It seems this sort of function isn't compatible with the current gammalib/ctools architecture, so it'll stay with the veritas code. I'm gonna close this feature for the time being.

#3 - 07/01/2019 09:35 AM - Knödlseder Jürgen

- Status changed from Feedback to Rejected