

GammaLib - Feature #1001

Implement general scheme to handle event classes

11/21/2013 07:11 AM - Deil Christoph

Status:	New	Start date:	11/21/2013
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>The basic reconstructed event parameters are TIME, X, Y, ENERGY and these are the basis of the current instrument response function, model, data and likelihood analysis.</p> <p>But some events are better than others.</p> <p>Discrete event property example: events with higher `telescope multiplicity`, i.e. number of observed showers, have a better direction reconstruction and PSF and higher signal / background ratio.</p> <p>Continuous event property example: events with lower shower goodness have a better signal / background ratio (see Fig. 33. Left in http://adsabs.harvard.edu/abs/2009APh....32..231D).</p> <p>There's several other examples of event properties where splitting the dataset into event classes and then doing a joint likelihood fit should result in much-improved sensitivity for HESS 1, HESS 2 (~50%) and CTA (maybe even more). That is as long as the relative signal and background between the event classes is well behaved ... especially too many free background amplitudes for event classes kill the sensitivity improvement.</p> <p>It's clear these event parameters can simply be stored in the event list.</p> <p>For the IRFs and the Gammalib / ctools software it's not clear to me how this should be implemented, but I think a general scheme to handle extra event properties / classes is needed. (Actually this is what Werner Hofmann said yesterday after a discussion with Chia-Chun and me.)</p> <p>The crucial part is that it's flexible in general, i.e. it let's us explore different properties, and wrt. background modeling in particular, i.e. it let's us introduce more and less free parameters into the background model.</p>			

History

#1 - 11/21/2013 12:57 PM - Knödlseider Jürgen

Note sure what this feature implies on code and interface design.

I had foreseen to move the likelihood computation into the GXXXObservation classes, so that more fancy schemes taken the co-variance of individual events can be taken into account.

But I think a scheme where you group events of different quality also makes sense, somewhat similar to the Fermi/LAT front and back event classification. Such kinds of schemes can already be supported with the present software (at least in principle; some code is probably needed for implementation, but interfaces should be okay).

#2 - 11/21/2013 01:47 PM - Deil Christoph

Should we try to make a simple case work first?

Let's take one Crab run and split it by event class (event multiplicity 2, 3, 4).
Each event class will have their own effective area and PSF (better at higher multiplicity).

Is there a way to feed this into Gammalib and fit this at the moment?
What are the possibilities for background modeling?

At the moment Chia-Chun and I don't know how to get started with Gammalib / ctools with this, so we'll definitely need some hand-holding.

#3 - 11/21/2013 02:12 PM - Kosack Karl

I would argue that this should not be a feature of the fitting, but rather the IRF. I don't think there will be fixed "event-classes" as in Fermi with CTA, but rather each event has some extra metadata that can be used to make fancier IRFs. The IRFs already use the event's azimuth and zenith angle as input parameters. One could easily foresee in the future also adding multiplicity as a dimension to the IRFs (or something similar, as depends on which telescope type as well). Again, that is not an event class per se, it's just using different responses for different events. There is also a strong possibility that events in CTA will (unlike HESS) have different reconstruction algorithms chosen per-event, depending on these sorts of parameters. In that case, you could also have a "reconstruction-id" or something, that would also be a dimension in your lookups if you want it to be.

Therefore, I suggest the following: don't make event classes a special case. Instead, ensure that the IRF tables are sufficiently flexible that you can later add dimensions as needed. It will also be necessary to define the interpolation-type per dimension such that some dimensions have interpolation-type=none (e.g. for multiplicity, or reconstruction-id, you wouldn't want to interpolate, but just use the fixed number).

For initial IRFs we produce (similar to the ones in HESS), we keep the IRFs simple, and simply assume the multiplicities are averaged over all events, so there is only 1 bin in multiplicity that contains all possibilities. Later, you may find that differentiating between multiplicity 1-5 and multiplicity 6-100 events gives better results, and all you have to do is make new tables with that dimension included.

There you need in your IRF reading routine the ability to read some metadata that describes all the dimensions that exist in the file, how each should be interpolated, and to which column in the event-list the dimension corresponds.

```
dimension="multiplicity"  
interpolation="none"  
eventlist_column="MULTIP"  
bins=...
```

#4 - 11/21/2013 02:15 PM - Knödlseher Jürgen

Deil Christoph wrote:

Should we try to make a simple case work first?

Let's take one Crab run and split it by event class (event multiplicity 2, 3, 4).
Each event class will have their own effective area and PSF (better at higher multiplicity).

Is there a way to feed this into Gammalib and fit this at the moment?
What are the possibilities for background modeling?

What you would do now is:

- create separate event lists for each event class
- create separate IRFs for each event class
- declare each event class / IRF as one observation in the XML file
- use the id attribute to tie a particular observation to a particular background model

At the moment Chia-Chun and I don't know how to get started with Gammalib / ctools with this, so we'll definitely need some hand-holding.

#5 - 11/21/2013 04:27 PM - Deil Christoph

Knödseder Jürgen wrote:

- use the id attribute to tie a particular observation to a particular background model

This is the part I don't know how to do. Do you have a working example to get us started?

#6 - 11/21/2013 04:34 PM - Knödseder Jürgen

Putting something like id="001" in the attributes of an observation and of a background model will tie them together.

Here an example of an observation XML file:

```
<observation_list title="observation library">
  <observation name="Crab" id="vp0001_0_075_1" instrument="COM">
    <parameter name="DRE" file="/project-data/comptel/phase01/vp0001_0/m50438_dre.fits"/>
    <parameter name="DRB" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRG" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRX" file="/project-data/comptel/phase01/vp0001_0/m32171_drx.fits"/>
    <parameter name="IAQ" file="com/u47512_iaq.fits"/>
  </observation>
  <observation name="Crab" id="vp0001_0_1_3" instrument="COM">
    <parameter name="DRE" file="/project-data/comptel/phase01/vp0001_0/m50439_dre.fits"/>
    <parameter name="DRB" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRG" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRX" file="/project-data/comptel/phase01/vp0001_0/m32171_drx.fits"/>
    <parameter name="IAQ" file="com/u47569_iaq.fits"/>
  </observation>
  <observation name="Crab" id="vp0001_0_3_10" instrument="COM">
    <parameter name="DRE" file="/project-data/comptel/phase01/vp0001_0/m50440_dre.fits"/>
    <parameter name="DRB" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRG" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRX" file="/project-data/comptel/phase01/vp0001_0/m32171_drx.fits"/>
    <parameter name="IAQ" file="com/u47741_iaq.fits"/>
  </observation>
  <observation name="Crab" id="vp0001_0_10_30" instrument="COM">
    <parameter name="DRE" file="/project-data/comptel/phase01/vp0001_0/m50441_dre.fits"/>
    <parameter name="DRB" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRG" file="/project-data/comptel/phase01/vp0001_0/m34997_drg.fits"/>
    <parameter name="DRX" file="/project-data/comptel/phase01/vp0001_0/m32171_drx.fits"/>
    <parameter name="IAQ" file="com/u48091_iaq.fits"/>
  </observation>
  <observation name="Crab" id="00001" instrument="LAT">
    <parameter name="CountsMap" file="/project-data/cta/data/fermi/crab/srcmap.fits"/>
    <parameter name="ExposureMap" file="/project-data/cta/data/fermi/crab/binned_expmap.fits"/>
    <parameter name="LiveTimeCube" file="/project-data/cta/data/fermi/crab/lcube.fits"/>
    <parameter name="IRF" value="P7SOURCE_V6"/>
  </observation>
</observation_list>
```

and part of the source model (link here is between energy ranges for COMPTEL):

```
<source name="Bgd. 0.75-1 MeV" type="DRBFitting" instrument="COM" id="vp0001_0_075_1">
  <node>
    <parameter name="Phibar" scale="1.0" value="1.0" min="0.0" max="50.0" free="0"/>
    <parameter name="Normalization" scale="1.0" value="0.0" min="0.0" max="1000.0" free="0"/>
  </node>
  <node>
    <parameter name="Phibar" scale="1.0" value="3.0" min="0.0" max="50.0" free="0"/>
    <parameter name="Normalization" scale="1.0" value="0.0" min="0.0" max="1000.0" free="0"/>
  </node>
```

```
<node>
  <parameter name="Phibar" scale="1.0" value="5.0" min="0.0" max="50.0" free="0"/>
  <parameter name="Normalization" scale="1.0" value="0.0" min="0.0" max="1000.0" free="0"/>
</node>
...
<node>
  <parameter name="Phibar" scale="1.0" value="49.0" min="0.0" max="50.0" free="0"/>
  <parameter name="Normalization" scale="1.0" value="0.0" min="0.0" max="1000.0" free="0"/>
</node>
</source>
<source name="Bgd. 1-3 MeV" type="DRBFitting" instrument="COM" id="vp0001_0_1_3">
  <node>
    <parameter name="Phibar" scale="1.0" value="1.0" min="0.0" max="50.0" free="0"/>
    <parameter name="Normalization" scale="1.0" value="0.0" min="0.0" max="1000.0" free="0"/>
  </node>
  <node>
    <parameter name="Phibar" scale="1.0" value="3.0" min="0.0" max="50.0" free="0"/>
    <parameter name="Normalization" scale="1.0" value="0.0" min="0.0" max="1000.0" free="0"/>
  </node>
  ...
  <node>
    <parameter name="Phibar" scale="1.0" value="49.0" min="0.0" max="50.0" free="0"/>
    <parameter name="Normalization" scale="1.0" value="1.0" min="0.0" max="1000.0" free="1"/>
  </node>
</source>
</source_library>
```

#7 - 12/11/2013 08:26 PM - Knödlseher Jürgen

- Target version set to 2nd coding sprint

#8 - 07/19/2014 02:09 AM - Knödlseher Jürgen

- Target version deleted (2nd coding sprint)