GammaLib - Action #2429

Correctly handling run-specific energy thresholds during model fit in OnOff analysis

04/05/2018 04:02 PM - Specovius Andreas

Status:	Closed	**Start date:**	04/05/2018
Priority:	High	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.6.0		

Description

Performing a fit of a spectral model to data means to compare the counts that are predicted by the model to the counts that have been measured.
Typically the measured events are selected before applying the fit and run-specific energy thresholds are applied.

During the fit of a model to the data the run-specific energy thresholds applied during event selection should also be applied to the counts predicted by the model. This should be done so that both, the predicted and the measured (and selected) counts, can be compared, right?

I don't see whether or where this is implemented for the on/off analysis and Jürgen did also not know. (Jürgen see email 5.apr.2018) Jürgen assumed that the ARF and the RMF should take care of that.

History

#1 - 04/20/2018 05:37 PM - Tibaldo Luigi

Hi Andreas,
can we assume as per our email conversation that for the moment you can survive using the joint-likelihood option, and later on we will fully address this also for the stacked case in version 1.6?

#2 - 04/24/2018 03:18 PM - Specovius Andreas

user#266 wrote:

> Hi Andreas,
> can we assume as per our email conversation that for the moment you can survive using the joint-likelihood option, and later on we will fully address this also for the stacked case in version 1.6?

Currently I'm indeed using the joint-likelihood option and at least the generated on/off counts spectra look fine. It would be nice to have this functionality also for the stacked case but of course I understand if you have some other urgent things to do!

#3 - 06/06/2018 02:12 PM - Knödlseder Jürgen

- Status changed from New to In Progress

- Assigned To set to Knödlseder Jürgen

- Target version set to 1.6.0

- % Done changed from 0 to 10

I started to look into the problem with a simple Crab simulation, two 30 min runs, the first covering 0.1-10 TeV and the second covering 0.5-100 TeV.

Runs were pointed +1 and -1 deg off the Crab position (wobbling). Unbinned and stacked 3D analysis produce the correct results. For stacked, a sufficient number of energy bins was needed to reproduce the prefactor:

Parameter	True	Unbinned	Stacked (20 bins)	Stacked (30 bins)	Stacked (40 bins)	Stacked (50 bins)
Prefactor (1e-16)	5.7	5.728	5.401	5.525	5.597	5.635
Index	2.48	2.481	2.477	2.478	2.482	2.482

I then ran csphagen to prepare a On/Off observation. Data were not stacked. Using 50 bins from 0.1-100 TeV I got the following result, which is obviously wrong:

```
2018-06-06T12:09:16: === GModelSky ===
2018-06-06T12:09:16:  Name .....: Crab
2018-06-06T12:09:16:  Instruments .....: all
2018-06-06T12:09:16:  Instrument scale factors ..: unity
2018-06-06T12:09:16:  Observation identifiers ...: all
2018-06-06T12:09:16:  Model type .....: PointSource
2018-06-06T12:09:16:  Model components .....: "PointSource" * "PowerLaw" * "Constant"
2018-06-06T12:09:16:  Number of parameters .....: 6
2018-06-06T12:09:16:  Number of spatial par's ...: 2
2018-06-06T12:09:16:   RA .....: 83.6331 [-360,360] deg (fixed,scale=1)
2018-06-06T12:09:16:   DEC .....: 22.0145 [-90,90] deg (fixed,scale=1)
2018-06-06T12:09:16:  Number of spectral par's ..: 3
2018-06-06T12:09:16:   Prefactor .....: 3.55468416156847e-16 +/- 6.70495230828592e-18 [1e-23,1e-13] ph/cm2/s/MeV
 (free,scale=1e-16,gradient)
2018-06-06T12:09:16:   Index .....: -2.28628261868409 +/- 0.0143818034081698 [-0,-5]  (free,scale=-1,gradient)
2018-06-06T12:09:16:   PivotEnergy .....: 300000 [10000,1000000000] MeV (fixed,scale=1000000,gradient)
2018-06-06T12:09:16:  Number of temporal par's ..: 1
2018-06-06T12:09:16:   Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
2018-06-06T12:09:16: === GCTAModelIrfBackground ===
2018-06-06T12:09:16:  Name .....: CTABackgroundModel
2018-06-06T12:09:16:  Instruments .....: CTAOnOff
2018-06-06T12:09:16:  Instrument scale factors ..: unity
2018-06-06T12:09:16:  Observation identifiers ...: all
2018-06-06T12:09:16:  Model type .....: "PowerLaw" * "Constant"
2018-06-06T12:09:16:  Number of parameters .....: 4
2018-06-06T12:09:16:  Number of spectral par's ..: 3
2018-06-06T12:09:16:   Prefactor .....: 0.790646869274759 +/- 0.0291891270672604 [0.001,1000] ph/cm2/s/MeV (free,scale=1,gradient)
2018-06-06T12:09:16:   Index .....: 0.239563350871814 +/- 0.0231485547658986 [-5,5]  (free,scale=1,gradient)
2018-06-06T12:09:16:   PivotEnergy .....: 1000000 [10000,1000000000] MeV (fixed,scale=1000000,gradient)
2018-06-06T12:09:16:  Number of temporal par's ..: 1
2018-06-06T12:09:16:   Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
```

Restricting the energy range to 0.5-10 TeV gave a correct result (see below). Hence there is an issue with the energy range, even for joint analysis.

```
2018-06-06T12:11:10: === GModelSky ===
2018-06-06T12:11:10:  Name .....: Crab
2018-06-06T12:11:10:  Instruments .....: all
2018-06-06T12:11:10:  Instrument scale factors ..: unity
2018-06-06T12:11:10:  Observation identifiers ...: all
2018-06-06T12:11:10:  Model type .....: PointSource
2018-06-06T12:11:10:  Model components .....: "PointSource" * "PowerLaw" * "Constant"
2018-06-06T12:11:10:  Number of parameters .....: 6
2018-06-06T12:11:10:  Number of spatial par's ...: 2
2018-06-06T12:11:10:   RA .....: 83.6331 [-360,360] deg (fixed,scale=1)
2018-06-06T12:11:10:   DEC .....: 22.0145 [-90,90] deg (fixed,scale=1)
2018-06-06T12:11:10:  Number of spectral par's ..: 3
2018-06-06T12:11:10:   Prefactor .....: 5.81925161790969e-16 +/- 2.95299451300941e-17 [1e-23,1e-13] ph/cm2/s/MeV
 (free,scale=1e-16,gradient)
2018-06-06T12:11:10:   Index .....: -2.4871091060634 +/- 0.0306437503619763 [-0,-5]  (free,scale=-1,gradient)
2018-06-06T12:11:10:   PivotEnergy .....: 300000 [10000,1000000000] MeV (fixed,scale=1000000,gradient)
2018-06-06T12:11:10:  Number of temporal par's ..: 1
2018-06-06T12:11:10:   Normalization .....: 1 (relative value) (fixed,scale=1,gradient)
2018-06-06T12:11:10: === GCTAModelIrfBackground ===
2018-06-06T12:11:10:  Name .....: CTABackgroundModel
2018-06-06T12:11:10:  Instruments .....: CTAOnOff
2018-06-06T12:11:10:  Instrument scale factors ..: unity
2018-06-06T12:11:10:  Observation identifiers ...: all
2018-06-06T12:11:10:  Model type .....: "PowerLaw" * "Constant"
2018-06-06T12:11:10:  Number of parameters .....: 4
```

```
2018-06-06T12:11:10:  Number of spectral par's ..: 3
2018-06-06T12:11:10:   Prefactor ................: 0.950294898816245 +/- 0.042133644979734 [0.001,1000] ph/cm2/s/MeV (free,scale=1,gradient)
2018-06-06T12:11:10:   Index ....................: -0.0454010832632682 +/- 0.0668873099688798 [-5,5]  (free,scale=1,gradient)
2018-06-06T12:11:10:   PivotEnergy ..............: 1000000 [10000,1000000000] MeV (fixed,scale=1000000,gradient)
2018-06-06T12:11:10:  Number of temporal par's ..: 1
2018-06-06T12:11:10:   Normalization ............: 1 (relative value) (fixed,scale=1,gradient)
```

#4 - 06/06/2018 03:53 PM - Knödlseder Jürgen

I added a GCTAOnOffObservation::apply_ebounds protected method that is called after setting up the histograms and that set the PHA spectra, RMF reconstructed energy pixels and background response pixels to zero outside the energy range of the CTA observation. Partially overlapping bins are set to zero. **Note that the ARF and RMF true energy pixels are not reset to zero, which is different to the 3D stacked analysis where they are zero.** Neither way is correct, since the spill over is difficult to handle, but at least for the test case, this gave the better results.

Results are summarized in the table below. Events were simulated using energy dispersion, since energy dispersion is always used for a On/Off analysis. The last two columns are for the case that the ARF and RMF vector and matrix are also set to zero for true energies outside the relevant energy range. In this case the results depart more significantly from the true values.

Parameter	**True**	**Joint**	**Stacked**	**Joint (ARF=RMF=0)**	**Stacked (ARF=RMF=0)**
Prefactor (1e-16)	5.7	5.629	5.635	5.768	5.842
Index	2.48	2.478	2.479	2.492	2.493

#5 - 06/06/2018 04:08 PM - Knödlseder Jürgen

I tested a more extreme case, where run1 covered the low-energy band 26 GeV - 1 TeV for 30 minutes and run2 the high-energy band 0.5 - 100 TeV for 3 hours. Here not setting the etrue pixels in the ARF and RMF to zero produces a worse result. **Therefore, and for consistency with the 3D analysis, I decided to also set the etrue pixels to zero to avoid the spill over from one observation into the other.**

Parameter	**True**	**Joint**	**Stacked**	**Joint (ARF=RMF=0)**	**Stacked (ARF=RMF=0)**
Prefactor (1e-16)	5.7	5.584	5.591	5.732	5.811
Index	2.48	2.470	2.470	2.481	2.481

#6 - 06/06/2018 04:24 PM - Knödlseder Jürgen

- File gammalib_tn0002_roi_stacked.pdf added

Just as a reminder, here the document that explains why no exact solution exists for a stacked analysis: attachment:"gammalib_tn0002_roi_stacked.pdf".

#7 - 06/06/2018 04:53 PM - Knödlseder Jürgen

- Project changed from ctools to GammaLib

- Status changed from In Progress to Closed

- Target version changed from 1.6.0 to 1.6.0

- % Done changed from 10 to 100

Merged code into devel.

#8 - 06/11/2018 11:22 AM - Knödlseder Jürgen

I revised the handling of the run-specific energy threshold.

When an On/Off observation is constructed, only reconstructed energy bins are reset to zero outside the valid energy range, so that a joint-analysis correctly takes the energy dispersion at the thresholds into account.

Only when On/Off observations are stacked using the GCTAOnOffObservation stacking operator, the true energy range is also restricted to the valid energy range to avoid spill over from one observation into another. For that purpose, the OBS-EMIN and OBS-EMAX keywords were added to the header of the PHA files to store the valid energy range for each observation.

Files

gammalib_tn0002_roi_stacked.pdf	109 KB	06/06/2018	Knödlseder Jürgen