GammaLib - Bug #2179

ctools science verification fails

08/22/2017 04:35 PM - Knödlseder Jürgen

Status:	Closed	**Start date:**	08/22/2017
Priority:	Urgent	**Due date:**	
Assigned To:	Knödlseder Jürgen	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.4.1		

Description

The science verification produces since April the following errors:

    ctools science verification.Test nodes model: Mean -19088.02665 of Pull_Crab_Intensity0 should be within [-0.40,0.40] range
    ctools science verification.Test nodes model: Standard deviation 17571.98646 of Pull_Crab_Intensity0 should be within
[0.80,1.20] range
    ctools science verification.Test nodes model: Mean -4.83094 of Pull_Crab_Intensity1 should be within [-0.40,0.40] range
    ctools science verification.Test nodes model: Standard deviation 1.98642 of Pull_Crab_Intensity1 should be within [0.80,1.20]
 range
    ctools science verification.Test nodes model: Mean -37.78501 of Pull_Crab_Intensity2 should be within [-0.40,0.40] range
    ctools science verification.Test nodes model: Standard deviation 10.75212 of Pull_Crab_Intensity2 should be within [0.80,1.20]
 range
    ctools science verification.Test nodes model: Mean -660.27252 of Pull_Crab_Intensity3 should be within [-0.40,0.40] range
    ctools science verification.Test nodes model: Standard deviation 4405.75126 of Pull_Crab_Intensity3 should be within
[0.80,1.20] range
    ctools science verification.Test diffuse cube model: Mean -8.34991 of Pull_Crab_Prefactor should be within [-0.40,0.40] range
    ctools science verification.Test diffuse cube model: Mean 13.47494 of Pull_Crab_Index should be within [-0.40,0.40] range

The following changes were done in GammaLib and ctools:

ctools
1   Add smoothly broken power law to science verification (#1948) (detail)
2   Optionally add LO_THRES and HI_THRES keywords to IRF (detail)
3   Enhance show_irf.py script (detail)
4   Introduce site-dependent energy range in make_pointings.py (detail)
5   Support of map cubes lying outside RoI in ctobssim (detail)
6   Do not fit smoothness parameter in smoothly broken power law for (detail)
7   Add user documentation for smoothly broken power law (detail)
8   Add unit test for show_irf.py example script (detail)
9   Correct debug message (detail)
10   Change show_irf.py so that it is also compliant with older matplotlib (detail)

GammaLib
1   Fixed issues with the GModelSpectralSmoothBrokenPlaw MC generation (detail)
2   Removed old code from GModelSpectralSmoothBrokenPlaw (#1948) (detail)
3   Enhance precision of GCTAEdisp2D::compute_ebounds_src (detail)
4   Set table boundaries in GCTAEdisp2D before normalizing the table (detail)
5   Revert to parallel tests (detail)
6   Correct model normalisation in test_model_spectral.py (detail)
7   Integrate smoothly broken power law spectrum (detail)
8   Correct comments (detail)
9   Add user documentation for smoothly broken power law (detail)
10   Make GModelSky::mc() method save against invalid energy ranges or model (detail)
11   Invert loop in GModelSpatialDiffuseCube::set_mc_cone to speed up (detail)
12   Assure that GCTAEdisp2D::mc() method does not block for empty energy (detail)
13   Add energy margin in response cube computation (detail)

History

#1 - 08/22/2017 04:36 PM - Knödlseder Jürgen

- Target version changed from 1.5.0 to 1.4.1

#2 - 08/22/2017 11:14 PM - Knödlseder Jürgen

- Project changed from ctools to GammaLib

- Status changed from New to In Progress

- Target version changed from 1.4.1 to 1.4.1

- % Done changed from 0 to 90

It turned out that a bug introduced in GModelSpectralNodes::mc() produced that problem. In fact, the case that there is a single node in the MC cache led to an exception instead of using this single node, which is needed for extrapolation. Changing the code as follows solved the problem:

```
if (m_mc_cum.size() > 1) {
    double u = ran.uniform();
    for (inx = m_mc_cum.size()-1; inx > 0; --inx) {
        if (m_mc_cum[inx-1] <= u) {
            break;
        }
    }
}
else if (m_mc_cum.size() == 0) {
    std::string msg = "No valid nodes found for energy interval ["+
                emin.print()+","+emax.print()+"]. Either restrict "
                "the energy range to the one covered by the "
                "spectral nodes or extend the spectral nodes "
                "in energy.";
    throw GException::invalid_return_value(G_MC, msg);
}
```

#3 - 08/23/2017 12:33 AM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 90 to 100

Fixed and merged into devel. A bugfix release 1.4.1 of GammaLib is in preparation.