

GammaLib - Feature #1706

Support composite spatial models

02/19/2016 06:39 PM - Mayer Michael

Status:	Closed	Start date:	
Priority:	Normal	Due date:	
Assigned To:		% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.2.0		

Description

I got an interesting feature request from Philipp Willmann.

He asked me if it is possible to fit one spectrum for a composite spatial model, e.g. a superposition of two Gaussians.

This might be a scenario that will come up more often towards CTA. Source morphologies are not always easy to model. Having a spatial model that is a superposition of individual models could help modelling complex morphologies.

We could have a new `GModelSpatialComposite` model that contains a `std::vector<GModelSpatial>` objects. I guess the only worry is to get the normalisation of the resulting PDF correct, right?

Of course we would also have to think about the XML interface. What first popped into my head are two options:

1. Have nested spatial models:

```
<spatialModel type="Composite">
  <spatialModel type="GaussFunction">
    ...
  </spatialModel>
  <spatialModel type="GaussFunction">
    ...
  </spatialModel>
</spatialModel>
```

2. Allow several spatial, spectral (and temporal) models per `GModelSky` element:

```
<spatialModel type="GaussFunction">
  ...
</spatialModel>
<spatialModel type="GaussFunction">
  ...
</spatialModel>
```

This could, in principle also allow several spectral components for one spatial model. An important use case for this is e.g. the Crab in Fermi where we have a pulsed spectrum, the synchrotron and IC component from one single spot in the sky.

I haven't thought about how this could be done technically yet, I just wanted to mention it here that we don't forget about it.

History

#1 - 03/04/2016 12:28 PM - Knödseder Jürgen

I think we can solve this issue by introducing the concept of connected or linked parameters (or maybe joint parameters).

The idea I had in mind to add something like the following to the model definition XML file:

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="Crab pulsar" type="PointSource">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" scale="1e-16" value="5.7" min="1e-07" max="1000.0" free="1"/>
    </spectrum>
  </source>
</source_library>
```

```

<parameter name="Index" scale="-1" value="2.48" min="0.0" max="+5.0" free="1"/>
<parameter name="Scale" scale="1e6" value="0.3" min="0.01" max="1000.0" free="0"/>
</spectrum>
<spatialModel type="SkyDirFunction">
  <parameter max="360" min="-360" name="RA" scale="1" value="83.6331" free="1" />
  <parameter max="90" min="-90" name="DEC" scale="1" value="22.0145" free="1" />
</spatialModel>
</source>
<source name="Crab nebula" type="PointSource">
  <spectrum type="PowerLaw">
    <parameter name="Prefactor" scale="1e-16" value="5.7" min="1e-07" max="1000.0" free="1"/>
    <parameter name="Index" scale="-1" value="2.48" min="0.0" max="+5.0" free="1"/>
    <parameter name="Scale" scale="1e6" value="0.3" min="0.01" max="1000.0" free="0"/>
  </spectrum>
  <spatialModel type="SkyDirFunction">
    <parameter max="360" min="-360" name="RA" scale="1" value="83.6331" free="1" />
    <parameter max="90" min="-90" name="DEC" scale="1" value="22.0145" free="1" />
  </spatialModel>
</source>
<link>
  <source name="Crab pulsar" parameter="RA">
  <source name="Crab nebula" parameter="RA">
  <parameter max="360" min="-360" name="RA" scale="1" value="83.6331" free="1" />
</link>
</source_library>

```

The <link> information should be saved together with the parameters in a GModels container. The method

```

void GObservations::optimize(GOptimizer& opt)
{
  // Extract optimizer parameter container from model container
  GOptimizerPars pars = m_models.pars();

  // Optimize model parameters
  opt.optimize(m_fct, pars);

  // Return
  return;
}

```

should scan the linked parameters and make modifications to the parameters. All linked parameters should be fixed, and an additional parameter should be added that reflects the common linked parameter. We then have to think how to resolve the parameters in the model evaluation.

#2 - 10/03/2016 09:36 PM - Knödseder Jürgen

We discussed this issue today during the coding sprint and the following format seems to be most easy to implement:

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="Crab" type="PointSource">
    <spatialModel type="Composite">
      <spatialModel type="PointSource">
        <parameter name="RA" scale="1.0" value="83.6331" min="-360" max="360" free="1"/>
        <parameter name="DEC" scale="1.0" value="22.0145" min="-90" max="90" free="1"/>
      </spatialModel>
      <spatialModel type="RadialGaussian">
        <parameter name="RA" scale="1.0" value="83.6331" min="-360" max="360" free="1"/>
        <parameter name="DEC" scale="1.0" value="22.0145" min="-90" max="90" free="1"/>
        <parameter name="Sigma" scale="1.0" value="0.20" min="0.01" max="10" free="1"/>
      </spatialModel>
    </spatialModel>
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" scale="1e-16" value="5.7" min="1e-07" max="1000.0" free="1"/>
      <parameter name="Index" scale="-1" value="2.48" min="0.0" max="+5.0" free="1"/>
      <parameter name="PivotEnergy" scale="1e6" value="0.3" min="0.01" max="1000.0" free="0"/>
    </spectrum>
  </source>
</source_library>
```

for the spatial composite model and

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="Crab" type="PointSource">
    <spatialModel type="PointSource">
      <parameter name="RA" scale="1.0" value="83.6331" min="-360" max="360" free="1"/>
      <parameter name="DEC" scale="1.0" value="22.0145" min="-90" max="90" free="1"/>
    </spatialModel>
    <spectrum type="Composite">
      <spectrum type="PowerLaw">
        <parameter name="Prefactor" scale="1e-16" value="5.7" min="1e-07" max="1000.0" free="1"/>
        <parameter name="Index" scale="-1" value="2.48" min="0.0" max="+5.0" free="1"/>
        <parameter name="PivotEnergy" scale="1e6" value="0.3" min="0.01" max="1000.0" free="0"/>
      </spectrum>
      <spectrum type="Gaussian">
        <parameter name="Normalization" scale="1e-10" value="1.0" min="1e-07" max="1000.0" free="1"/>
        <parameter name="Mean" scale="1e6" value="5.0" min="0.01" max="100.0" free="1"/>
        <parameter name="Sigma" scale="1e6" value="1.0" min="0.01" max="100.0" free="1"/>
      </spectrum>
    </spectrum>
  </source>
</source_library>
```

for the spectral composite model. The spatial composite model should be implemented by the `GModelSpatialComposite` class, the spectral composite model by the `GModelSpectralComposite` class. Both classes would be container classes for spatial or spectral model components, and would simply loop over its components in the `eval()`, `flux()` and `eflux()` methods.

In the examples above there are two components shown, but the composite model may eventually contain an arbitrary number of components.

Note that there may be an issue with accessing model parameters by name of a composite model since the parameter names within a model component (for example a spectral model component) need to be unique. This could be solved for example by prepending a component index internally, e.g. "RA" could become "1:RA" for the first component.

The implementation of the `mc()` method needs also some thinking. The composite classes probably need to randomly dispatch between the model components, and then call the `mc()` method of the component.

#3 - 10/05/2016 03:26 PM - Mayer Michael

- File *HardComponent_Index.png* added
- File *HardComponent_Prefactor.png* added
- File *SoftComponent_Index.png* added
- File *SoftComponent_Prefactor.png* added
- Status changed from *New* to *In Progress*
- Target version set to *1.2.0*
- % Done changed from *0* to *50*

I have implemented the new class *GModelSpectralComposite* including unit tests and example XML files. The pull distributions also look reasonable. The distributions below were made using two power law components, one with an index of *-3.5* and the other using an index of *-2.0*. More pull distributions are underway.

#4 - 10/06/2016 12:54 PM - Mayer Michael

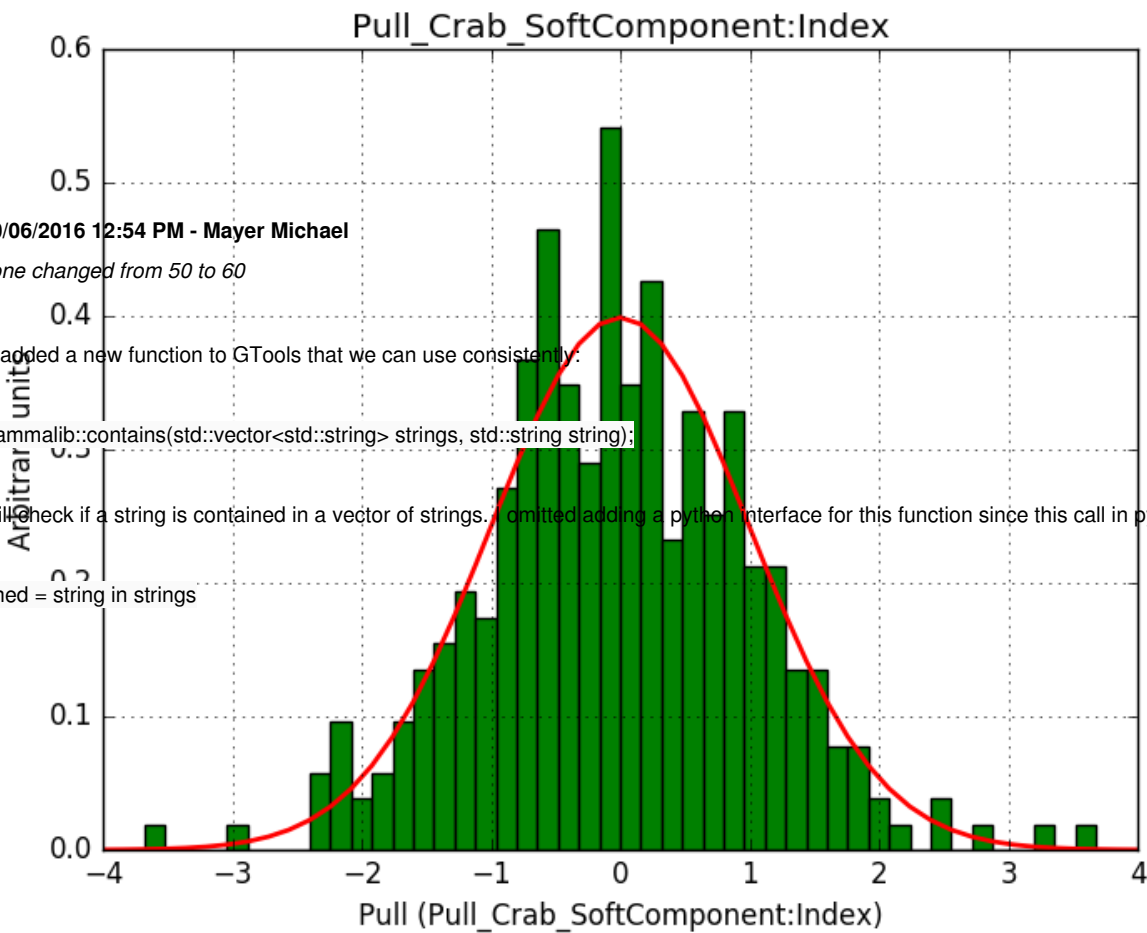
- % Done changed from *50* to *60*

I have added a new function to *GTools* that we can use consistently:

```
bool gammalib::contains(std::vector<std::string> strings, std::string string);
```

This will check if a string is contained in a vector of strings. I omitted adding a python interface for this function since this call in python is simply:

```
contained = string in strings
```



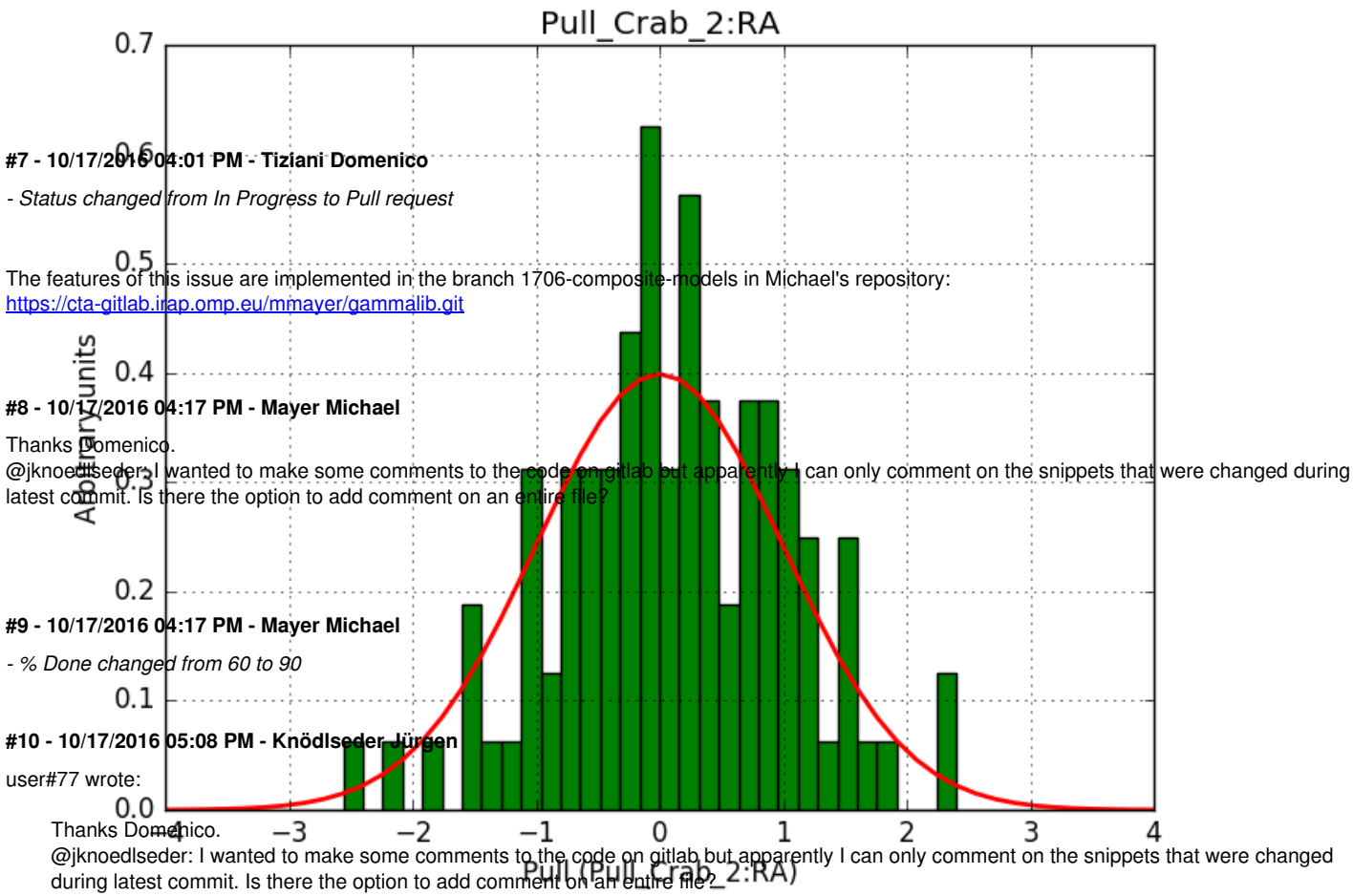
#5 - 10/06/2016 02:15 PM - Mayer Michael

Added related feature #1865 to not forget about documentation *smile.png*

#6 - 10/17/2016 03:49 PM - Tiziani Domenico

- File *1_DEC.png* added
- File *1_RA.png* added
- File *2_DEC.png* added
- File *2_RA.png* added

I have implemented the new GModelSpatialComposite class with corresponding unit tests.
 The pull distributions that I obtained for a composite model of two point sources with a distance of one degree between them look quite reasonable.



I think you can only comment commits, but not files.

#11 - 10/18/2016 12:21 AM - Knödseder Jürgen
 - Status changed from Pull request to Feedback
 - % Done changed from 90 to 100

Thanks for the code. These new models are really great. Looking forward to use them smile.png

Also congrats Domenico, these are your first commits to GammaLib wink.png

I integrated the code into the devel branch.

It would be good if you could double check with pull distributions that everything is okay. Once you confirm I will close the issue.

#12 - 10/18/2016 12:28 PM - Mayer Michael

We could still think of adding a "weight" attribute to each component of the GModelSpatialComposite to be read from the XML file. This way, we could have spatial components with different amplitudes as a combined spatial model. This could also be an add-on as a separate feature, what do you think?

#13 - 10/18/2016 12:48 PM - Knödlseeder Jürgen

user#77 wrote:

We could still think of adding a "weight" attribute to each component of the GModelSpatialComposite to be read from the XML file. This way, we could have spatial components with different amplitudes as a combined spatial model. This could also be an add-on as a separate feature, what do you think?

I support this idea. Maybe make it a separate feature.

#14 - 10/18/2016 01:01 PM - Tiziani Domenico

user#77 wrote:

We could still think of adding a "weight" attribute to each component of the GModelSpatialComposite to be read from the XML file. This way, we could have spatial components with different amplitudes as a combined spatial model. This could also be an add-on as a separate feature, what do you think?

I also agree. I think that I can implement this since I already created the class GModelSpatialComposite.

#15 - 10/18/2016 01:43 PM - Mayer Michael

Great, thanks Domenico. You can just go ahead, create the issue and make a new branch from devel to implement the changes.

#16 - 10/18/2016 02:54 PM - Tiziani Domenico

Added feature [\[\[redmine>#1872\]\]](#) for the weight attribute.

#17 - 10/18/2016 03:32 PM - Tiziani Domenico

I just re-calculated pull distributions for a spatial composite model and they still look OK.

#18 - 10/18/2016 04:10 PM - Knödlseeder Jürgen

user#195 wrote:

I just re-calculated pull distributions for a spatial composite model and they still look OK.

Thanks for double checking.

#19 - 10/18/2016 05:57 PM - Mayer Michael

user#195 wrote:

Added feature [[redmine>#1872]] for the weight attribute.

This link goes somewhere else.

#20 - 10/18/2016 06:30 PM - Knödlseider Jürgen

user#77 wrote:

user#195 wrote:

Added feature [[redmine>#1872]] for the weight attribute.

This link goes somewhere else.

Just remove the Redmine thing, #1872.

#21 - 03/03/2017 10:17 AM - Knödlseider Jürgen

- Status changed from Feedback to Closed

Files

HardComponent_Index.png	44.6 KB	10/05/2016	Mayer Michael
HardComponent_Prefactor.png	53.4 KB	10/05/2016	Mayer Michael
SoftComponent_Index.png	44.9 KB	10/05/2016	Mayer Michael
SoftComponent_Prefactor.png	45.9 KB	10/05/2016	Mayer Michael
1_RA.png	38.9 KB	10/17/2016	Tiziani Domenico
1_DEC.png	38.8 KB	10/17/2016	Tiziani Domenico
2_DEC.png	39.3 KB	10/17/2016	Tiziani Domenico
2_RA.png	39 KB	10/17/2016	Tiziani Domenico