GammaLib - Action #1096

BackgroundModel from IRF

01/22/2014 02:37 PM - Gerard Lucie

Status:	Closed	**Start date:**	02/07/2014
Priority:	Immediate	**Due date:**	
Assigned To:	Gerard Lucie	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

Implementing the possibility to construct a GCTAModelBackground from the IRF file and the GObservation.
The background description would be given in camera coordinate.
The background map would be automatically centered on the observation pointing direction.
This also include working on the format of the IRF fits file and the reading of this file.

Subtasks:

Action # 1137: Bug Fix for the new GCTAModelBackground constructor	**Closed**

Related issues:

Related to GammaLib - Action # 1144: Use Prod2 FITS file for CTA response mod...	**Closed**	**02/12/2014**
Related to GammaLib - Feature # 1146: Create GCTAInstBackground class for han...	**Closed**	**02/13/2014**

History

#1 - 01/22/2014 09:47 PM - Knödlseder Jürgen

Can you explain more what you mean to construct a background model from the IRF and the observation? I though that the background model would eventually be constructed from the data, using empty fields, off regions or templates.

Maybe you can elaborate a little bit on this.

#2 - 01/23/2014 09:37 AM - Gerard Lucie

Currently to construct a GCTAModelBackground where the m_spatial is a GModelDiffuseCube one needs to provide a background event map in sky coordinate. We thought it would be more natural to provide it in camera coordinate. Since this background event map (created from empty fields, or simulation for CTA currently) represents the instrument response we also thought its natural place could be the IRF file. As for creating it also from the GObservation it would be to ensure that the center of the background map is centered on the pointing direction of the observation. And the projection from camera coordinate to sky coordinate, to produce the background GSkymap, could be done in the GCTAModelBackground constructor.
We were also wondering how much of the background construction should be left in the hand of the user. Maybe the user could choose the background type, but the background creation could be done automatically to make sure it is correctly done. It's also with this in mind we thought about creating it from the provided IRF file and the GObservation.

Does this sound reasonable?

#3 - 01/31/2014 03:59 PM - Gerard Lucie

- Status changed from New to Pull request

- % Done changed from 0 to 70

Constructor available. It is now possible to construct a GCTAModelBackground from
- an observation
- a fits file with the background information in instrument coordinate (see inst/cta/test/data/bg_test.fits for the format and expected units)
- and a spectral model.

- optionally the user can give the binning for the sky map spatial cube to be created.

Some thinking is still required in GCTAModelBackground::write() since it writes out the spatial model. In the case of the new constructor, however, we have a GModelSpatialDiffuseCub without an associated file name. An xml file coming out from this new constructor, would therefore not allow to read in the skymap again.

GCTAPointing and GCTAInstDir have been modified
They now contained the functionality to go from instrument coordinates to sky coordinates and vise versa. Tests unit for the coordinates rotation in classes need to be implemented. A dedicated issues will be created for that.

Test unit are in place for GCTAModelBackground. For the moment it only tests if the constructor works. Further testing needs to be implemented. In particular the normalization of the sky maps needs to be tested.

#4 - 02/03/2014 11:50 AM - Knödlseder Jürgen

- Status changed from Pull request to Feedback

- Remaining (hours) set to 0.0

I merged the branch 1096-CTABack_from_inst_coord into devel. I did some adjustments of the code to respect the coding conventions. I made some changes to the code, so if you continue, best start with a fresh checkout from devel.

#5 - 02/06/2014 12:04 PM - Gerard Lucie

- Estimated time set to 0.00

Bug related to the new constructor of GCTAModelBackground.

The two following lines:
GModelTemporalConst temporal;
m_temporal = temporal.clone();

have to be added to the
GCTAModelBackground::GCTAModelBackground(const GCTAObservation& obs,
const std::string& filename,
const GModelSpectral& spectral,
const int& nx_sky,
const int& ny_sky,
const int& n_energy) :
GModelData()

I'm still in the process a testing this new background.
Either I create a branch to fix this bug now, or I wait till a full analysis can be done with this model (created with the new constructor) so the new branch can be used to fix all the bugs.

Unless I'm recomended to do other wise, I'll choose the second option.

#6 - 02/07/2014 01:29 PM - Gerard Lucie

- Status changed from Feedback to In Progress

Issue 1137 have been created to fix bugs related to the new constructor.

#7 - 02/12/2014 06:20 PM - Knödlseder Jürgen

Maybe #1144 is just a double of this action. I just wanted to make sure that we can read the new Prod2 files. Check the extension BACKGROUND in inst/cta/caldb/data/cta/e/bcf/IFAE20120510_50h/irf_file.fits for the actual format.

#8 - 02/12/2014 06:26 PM - Gerard Lucie

The BACKGROUND format in the inst/cta/caldb/data/cta/e/bcf/IFAE20120510_50h/irf_file.fits are not, to my knowledge, in the format needed for GCTAModelBackground. I'm actually pretty sure about this because the information is not in the root file provided by the MC people.

Do you think having the same extension name for two different things is a problem?

#9 - 02/12/2014 08:01 PM - Knödlseder Jürgen

Gerard Lucie wrote:

> The BACKGROUND format in the inst/cta/caldb/data/cta/e/bcf/IFAE20120510_50h/irf_file.fits are not, to my knowledge, in the format needed for GCTAModelBackground. I'm actually pretty sure about this because the information is not in the root file provided by the MC people.
>
> Do you think having the same extension name for two different things is a problem?

Maybe I missed this: what is the format you're expecting to use for GCTAModelBackground?

Wouldn't it be possible to create a background map from the information in the root file?

Given the fact that today we have information in the root file we should see how we can use it already. We can of course change the extension name if needed. Otherwise, we may work with extension version numbers.

#10 - 02/13/2014 01:31 AM - Knödlseder Jürgen

After looking into the code I better understood what you meant.

So I changed the format of the response files that I generate from Prod2 to match the format you have implemented. Now we should be compliant.

I also added some code to GCTABackgroundModel to support defining the instrumental background model through an XML file. Here the actual syntax I implemented:

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="CTABackgroundModel" type="CTABackground" instrument="CTA" id="00001">
    <spatialModel type="Instrumental" instrument="CTA">
      <parameter name="EventList"          file="$(PACKAGE_SOURCE)/inst/cta/test/data/crab_events.fits"/>
      <parameter name="EffectiveArea"      file=""/>
```

```
    <parameter name="PointSpreadFunction" file=""/>
    <parameter name="EnergyDispersion"    file=""/>
    <parameter name="Background"          file="$(PACKAGE_SOURCE)/inst/cta/test/data/bg_test.fits"/>
   </spatialModel>
   <spectrum type="PowerLaw">
    <parameter name="Prefactor" scale="1.0"  value="1.0"  min="1e-6" max="1e6"    free="1"/>
    <parameter name="Index"     scale="1.0"  value="0.0"  min="-5.0" max="+5.0"   free="1"/>
    <parameter name="Scale"     scale="1e6"  value="1.0"  min="0.01" max="1000.0" free="0"/>
   </spectrum>
  </source>
</source_library>
```

This looks not very nice, but works for the moment. What is not so nice is that you have to replicate the entire structure that you find in the observation definition file. I did this to be able to simply use the GCTAObservation::read method to create the observation.

I recognized that with the actual code we lost the possibility to have the same background model for multiple observations. As each observation has it's own pointing, each GCTABackgroundModel that is created from an observation can formally apply only to that observation. The reason is that we tried to reuse the GModelSpatialDiffuseCube class. This adds the need for coordinate transformations. In principle, these are not needed in the GCTABackgroundModel::eval method that now should have directly DETX and DETY available through the GCTAInstDir member of the CTA event.

If we decide that we store the instrumental background information within GCTAResponse (which is somehow natural as the information is actually in the same file as the other instrument response informations), one could access the background model in instrument coordinates directly in GCTABackgroundModel::eval by corresponding methods (note that the observation is provided as argument to GCTABackgroundModel::eval, we thus can access the background model on a run-by-run basis while having only a single GCTABackgroundModel instance.

A possibility to signal that the instrumental background should be used in GCTABackgroundModel could be that the spatial component is simply omitted, i.e. one would write for the model

```
<?xml version="1.0" standalone="no"?>
<source_library title="source library">
  <source name="CTABackgroundModel" type="CTABackground" instrument="CTA" id="00001">
    <spectrum type="PowerLaw">
      <parameter name="Prefactor" scale="1.0"  value="1.0"  min="1e-6" max="1e6"    free="1"/>
      <parameter name="Index"     scale="1.0"  value="0.0"  min="-5.0" max="+5.0"   free="1"/>
      <parameter name="Scale"     scale="1e6"  value="1.0"  min="0.01" max="1000.0" free="0"/>
    </spectrum>
  </source>
</source_library>
```

One would still have the possibility to apply this model to only one (or a set of) observation(s) using the id attribute.

The eval and eval_gradients methods would then switch to instrumental background if the m_spatial pointer is NULL. The mc method would do the same thing.

#11 - 02/13/2014 09:45 AM - Gerard Lucie

I don't quite understand how the current Prod2 instrument root file can be used to produce the BACKGROUND fit file in the format needed by the new constructor. I think in this root file the background information is in offset bin. Having something more detailed was one of the reason for this GCTAModelBackground.

We also had some discussion about where the background file should be. In the instrument fits file or in an independent file. Maybe we can review the different argument about where this information should be.

As for the coordinate transformation and the use of GModelSpatialDiffuseCube, I thought this was a good think if we wanted the possibility to fit the background?

#12 - 02/13/2014 10:24 AM - Knödlseder Jürgen

Gerard Lucie wrote:

> I don't quite understand how the current Prod2 instrument root file can be used to produce the BACKGROUND fit file in the format needed by the new constructor. I think in this root file the background information is in offset bin. Having something more detailed was one of the reason for this GCTAModelBackground.

Well, you just assume symmetry around the camera centre and compute DETX and DETY from offset.

Once more information is available, more can be used.

> We also had some discussion about where the background file should be. In the instrument fits file or in an independent file. Maybe we can review the different argument about where this information should be.

I think we should always keep the possibility to have it in a separate file. For the moment I just put it as an extension in the IRF file for convenience, but the code does not assume that it is there. You can always specify a specific filename if the file is somewhere else, that's also the purpose for having added the Background parameter to the XML file.

> As for the coordinate transformation and the use of GModelSpatialDiffuseCube, I thought this was a good think if we wanted the possibility to fit the background?

It was just a convenience as the GModelSpatialDiffuseCube is already there. But in GModelSpatialDiffuseCube nothing is fitted, the fitting is done over the spectral component, and this one is still there in the scheme I propose. So functionality will be identical (or even improved as the same background model can be used for different observations).

#13 - 02/13/2014 02:53 PM - Gerard Lucie

So you are suggesting to transfer the background information from GModelSpatialDiffuseCube to the GCTAResponse.
We thought about this at the beginning of the coding sprint. I cannot quite remember why this option was finally not chosen.
In order to do this I guess one would have to implement some of the GModelSpatialDiffuseCube in GCTARepsonse (what I have in mind is the function used in GCTAModelBackground::mc(const GObservation& obs, GRan& ran)). I can look into this if a two week time scale is not too long.

Before you said:
So I changed the format of the response files that I generate from Prod2 to match the format you have implemented.
Does this mean you have updated the ctools script cta_root2caldb.py?

And for the information in the Prod2 instrument response root file, since with the current tools we don't have to assume radial symmetry we could simply ask for more information to be added.

#14 - 02/13/2014 03:05 PM - Knödlseder Jürgen

Gerard Lucie wrote:

> So you are suggesting to transfer the background information from GModelSpatialDiffuseCube to the GCTAResponse.
> We thought about this at the beginning of the coding sprint. I cannot quite remember why this option was finally not chosen.
> In order to do this I guess one would have to implement some of the GModelSpatialDiffuseCube in GCTARepsonse (what I have in mind is the function used in GCTAModelBackground::mc(const GObservation& obs, GRan& ran)). I can look into this if a two week time scale is not too long.

If you'd like I can certainly also start looking into that. The functionality is in fact needed for people who want to use Prod2 for their studies. This is of course relevant in the framework of the key science projects.

> Before you said:
> So I changed the format of the response files that I generate from Prod2 to match the format you have implemented.
> Does this mean you have updated the ctools script cta_root2caldb.py?

Exactly. I also uploaded a response database on Sharepoint at https://portal.cta-observatory.org/WG/DM/DM_wiki/DATA_Access/Pages/Science%20Tools.aspx. I prefer to have the response file CTA consortium internal to avoid problems with using the calibrations outside the consortium.

> And for the information in the Prod2 instrument response root file, since with the current tools we don't have to assume radial symmetry we could simply ask for more information to be added.

Right. We can also ask the MC people to extend their format (you may eventually just go next door and have a chat with Gernot on this). I guess that the statistics is just too limited for the moment to correctly fill the table.

#15 - 02/13/2014 03:10 PM - Mayer Michael

Hi Lucie, Jürgen,
I think the main idea to create a GModelSpatialDiffuseCube as m_spatial was to use the functionality of GSkymap. Furthermore, GModelSpatialDiffuseCube supported 3D -interpolation, which was not available before in GCTAResponseTable (but is now).
I fully agree that it would make sense at some point to just stay in instrument coordinates, using the GEvent properties to calculate the background. However, in my opinion, we should not infinitely expand GCTAModelBackground. I would propose to maybe move the newly created constructor to a new class, e.g. GCTAInstBackground or GCTAInstrumentBackground. This class could have its own read - function and could be disconnected from GCTAModelBackground. With this, the user would have both opportunities to model the background - either in sky, or instrument coordinates. What do you think?

#16 - 02/13/2014 03:14 PM - Knödlseder Jürgen

Agree with this strategy. One vote for GCTAInstrumentBackground.

#17 - 02/13/2014 03:17 PM - Mayer Michael

> One vote for GCTAInstrumentBackground.

smile.png I would rather go with GCTAInstBackground. It clarifies the connection to GCTAInstDir doesn't it?

#18 - 02/13/2014 03:20 PM - Knödlseder Jürgen

Good point. Buy in.

#19 - 02/13/2014 04:23 PM - Gerard Lucie

Knödlseder Jürgen wrote:

> Right. We can also ask the MC people to extend their format (you may eventually just go next door and have a chat with Gernot on this). I guess that the statistics is just too limited for the moment to correctly fill the table.

About the statistic, I would not think so. For the divergent pointing I'm working on, I extracted the necessary information and the background is correctly filled. I asked Gernot about adding this information to the Prod2 irf files. It is perfectly doable.

#20 - 02/13/2014 04:29 PM - Gerard Lucie

> I fully agree that it would make sense at some point to just stay in instrument coordinates, using the GEvent properties to calculate the background. However, in my opinion, we should not infinitely expand GCTAModelBackground. I would propose to maybe move the newly created constructor to a new class, e.g. GCTAInstBackground or GCTAInstrumentBackground. This class could have its own read - function and

could be disconnected from GCTAModelBackground. With this, the user would have both opportunities to model the background - either in sky, or instrument coordinates. What do you think?

Which new constructor are you referring to? If this is the one written during the coding sprint, it is going from instrument to sky coordinate so why would we want to put it in GCTAInstBackground, where things would remain in instrument coordinate?
Also, this new constructor is working (and I'm using it, if that's a valid argument), so it would be nice to leave it in GCTAModelBackground for a little while still.

Otherwise I also vote for GCTAInstBackground

#21 - 02/13/2014 05:58 PM - Knödlseder Jürgen

I think that's exactly the goal: leave GCTAModelBackground as is and implement a new class / model named GCTAInstBackground for handling the response tables directly without passing through the GModelSpatialDiffuseCube.

By the way, I was just trying to use GCTAModelBackground to validate the new response files and I encountered some unit problems. As I understand, you assumed that DETX and DETY are in radians, while I assumed that they were in degrees. This created some trouble for me, so I adjusted the code a little bit to cope with both possibilities.

Ultimately, we would need to define precisely the units for all quantities we're dealing with, but independently, it is always good practice to have the code analyzing the unit field for the response.

In the same direction, the unit of the diffuse model map is events/s/MeV/sr while you assumed I think in your test file events/s/TeV/sr. This will certainly need to a normalization problem in some place. So better switch to events/s/MeV/sr, as MeV is the internal energy unit of GammaLib. But best practice would be to analyse the unit field and perform conversion as needed on the fly.

#22 - 02/17/2014 10:14 PM - Knödlseder Jürgen

- Status changed from In Progress to Feedback

Changed to Feedback as GCTAModelInstBackground has been implemented and should do the job.

#23 - 02/19/2014 04:24 PM - Gerard Lucie

Juergen, you were very right about the background being in events/s/MeV/sr.

In the irf fits file from the above link (very useful) the BACKGROUND extension has the expected format for GCTAModelBackground (and I guess also for GCTAModelInstBackground, I'm not familiar with this class yet) but when I try to create those files myself from a fresh version of ctools it does not work.

Juergen, did you put the new version of cta_root2caldb.py in the devel?

#24 - 02/19/2014 04:37 PM - Knödlseder Jürgen

Gerard Lucie wrote:

> Juergen, you were very right about the background being in events/s/MeV/sr.
>
> In the irf fits file from the above link (very useful) the BACKGROUND extension has the expected format for GCTAModelBackground (and I guess also for GCTAModelInstBackground, I'm not familiar with this class yet) but when I try to create those files myself from a fresh version of ctools it does not work.
>
> Juergen, did you put the new version of cta_root2caldb.py in the devel?

Not in devel but in integration (forgot to merge into devel).

#25 - 07/23/2014 09:00 AM - Knödlseder Jürgen

- Status changed from Feedback to Closed