ctools - Feature #544

Create ctpoisson tool

10/10/2012 08:52 PM - Knödlseder Jürgen

Status:	New	**Start date:**	10/10/2012
Priority:	Normal	**Due date:**	
Assigned To:	Deil Christoph	**% Done:**	10%
Category:		**Estimated time:**	0.00 hour
Target version:			

Description

[Issue subject was: Add ctmodel and ctpoisson tools to efficiently simulate binned data]

One can simulate binned data by using ctobssim to simulate event lists and then bin them with ctbin or ctskymap.

But it should be much faster (and yield identical results) to directly compute expected counts images / cubes and then Poisson-fluctuate each bin.

Concretely I want to simulate a Galactic plane survey with ~ 10,000 runs and ~ 1000 sources and I also want to fill background according to different background models and study their residuals.

I guess a ctmodel tool could compute the expected count images / cubes such that the counts are preserved even for sources / PSF sizes that are small compared to the pixel size and using bounding boxes to not evaluate the model for the whole field of view.
And then there would be an extremely simple ctpoisson tool that simply Poisson fluctuated each bin.

Jürgen, would it be worth adding ctmodel and ctpoisson?
Sorry, I didn't check in detail what is already available in gammalib / ctools, maybe what I want is already possible?

Related issues:

Related to ctools - Feature # 352: Create ctmodel tool	**Closed**

History

#1 - 10/10/2012 08:54 PM - Knödlseder Jürgen

- Target version set to HESS sprint 1

#2 - 10/10/2012 09:02 PM - Knödlseder Jürgen

Both tools are indeed useful. In feature #352 you'll see that I have planned this already.

I'll thus move the feature #352 in the sprint and restrict this one the Poisson sampling. Maybe we should find a better name? ctpoisson is quite general. What about ctmodelsim or ctmodsim, in analogy to ctobssim?

#3 - 10/10/2012 09:21 PM - Deil Christoph

I think I'd prefer ctpoisson, because there I can guess from the name what it does, whereas for ctmodelsim and ctmodsim I'd not have guessed what the tool does.

Btw., I can't change the title or description of this issue. This is a bit annoying, since it is now incorrect. Can redmine be configured to allow title and description edits?

#4 - 10/10/2012 09:52 PM - Deil Christoph

- Subject changed from Add ctmodel and ctpoisson tools to efficiently simulate binned data to Create ctpoisson tool

- Description updated

#5 - 10/10/2012 10:03 PM - Deil Christoph

OK, I can write a ctpoisson prototype tomorrow.

- Is it ok if I do a prototype with pyfits and numpy.random.poisson(lam=1.0, size=None)? Or should I use gammalib FITS I/O?
- Is there a Poisson random number generator in gammalib?

#6 - 10/10/2012 10:59 PM - Knödlseder Jürgen

Christoph Deil wrote:

> OK, I can write a ctpoisson prototype tomorrow.

My only point with the name is that it does contain no hint upon what get's "poissoned" ... but maybe it's just a question of taste :-)

> - Is it ok if I do a prototype with pyfits and numpy.random.poisson(lam=1.0, size=None)? Or should I use gammalib FITS I/O?

I would use GammaLib as far as possible, which reduces dependencies, and also makes you more familiar with the GammaLib way.

> - Is there a Poisson random number generator in gammalib?

Yes, it's called GRan. It's part of the support module. Yet I recognized, it don't has a poisson method so far. Here you see my way of "efficient programming": just code what's needed.

It requires a little coding to get the poisson method implemented, but I have it already in some code that I used for INTEGRAL. I'll try to do this quickly.

#7 - 10/10/2012 11:56 PM - Knödlseder Jürgen

- % Done changed from 0 to 10

I just implement a poisson() method for the GRan class in GammaLib (see commit 3cf8ccc8 in HESS branch of GammaLib).

The method is not deeply tested, but since I copied over the code from a piece of software that I widely used for INTEGRAL, there is a good chance that it's okay.

However, some tests should be done to make sure that the method works as expected.

#8 - 10/11/2012 07:51 AM - Knödlseder Jürgen

There is another important aspect that needs discussion. For the moment, observations (or runs) are never added. The main reason for this is that this allows different and independent IRFs for each observation. But of course, it will you give some speed penalty. I'm in fact not sure that simulating events for a single run is slower than simulating a counts map.

Maybe it would be good to start putting together a script that is doing the job using ctobssim and ctbin. Just to see how long it takes, so that we have a reference. This may also help to improve the speed of the existing tools. We can then compare this to the ctmodel and ctpoisson tools later.

We could certainly add the option to add observations, but then we need a mechanism for time averaging IRFs, so that they are handled correctly.

#9 - 10/12/2012 11:26 AM - Knödlseder Jürgen

Development notes for ctpoisson can be found at the page [[ctpoisson]].

#10 - 10/15/2012 06:15 PM - Deil Christoph

I have this very preliminary ctpoisson tool implementation:

```
#!/usr/bin/env python
"""Draw a Poisson random sample from an expected counts map"""

import sys
import gammalib

def poisson_randomize_skymap(model):
    ran = gammalib.GRan()
    poisson = model.copy()

    for ii in xrange(model.npix()):
        poisson[ii] = ran.poisson(model[ii])

    return poisson

if __name__ == '__main__':
    infile = sys.argv[1]
    outfile = sys.argv[2]
    clobber = False

    print('Reading %s' % infile)
    model = gammalib.GSkymap(infile)

    print('Processing ...')
    poisson = poisson_randomize_skymap(model)

    print('Writing %s' % outfile)
    poisson.save(outfile)
```

I know I have to make this a GApplication and copy the FITS header and EBOUNDS and GTI info to the outfile.

But I wanted to ask if it would be OK to implement the utility function poisson_random_from_skymap in gammalib (if so where?) so that I can make ctpoisson fast (no looping in Python) but still write it in Python to avoid the C++ ctool boilerplate code?

#11 - 10/15/2012 06:40 PM - Knödlseder Jürgen

Christoph Deil wrote:

> But I wanted to ask if it would be OK to implement the utility function poisson_random_from_skymap in gammalib (if so where?) so that I can make ctpoisson fast (no looping in Python) but still write it in Python to avoid the C++ ctool boilerplate code?

I guess this should be done at the GCTAEventCube level. GCTAEventCube is the class that holds an event cube. There are load() and save() methods that you may use, which normally should take care already for GGti and GEbounds copying (to be confirmed). I think it would be best to add the method in that class, calling it for example mc() (this is how randomization methods are called throughout GammaLib). You want to pass a GRan object to the method, so that you can fully control the process (allowing for example reproducible random number generation; see for example GModelSpectralPlaw for the usage of the mc() method in a spectral model).

I may even think about adding mc() to the abstract GEventCube interface, making sure that this method is implement for all possible instruments.

#12 - 01/19/2014 02:05 AM - Knödlseder Jürgen

- Target version deleted (HESS sprint 1)