

ctools - Change request #1513

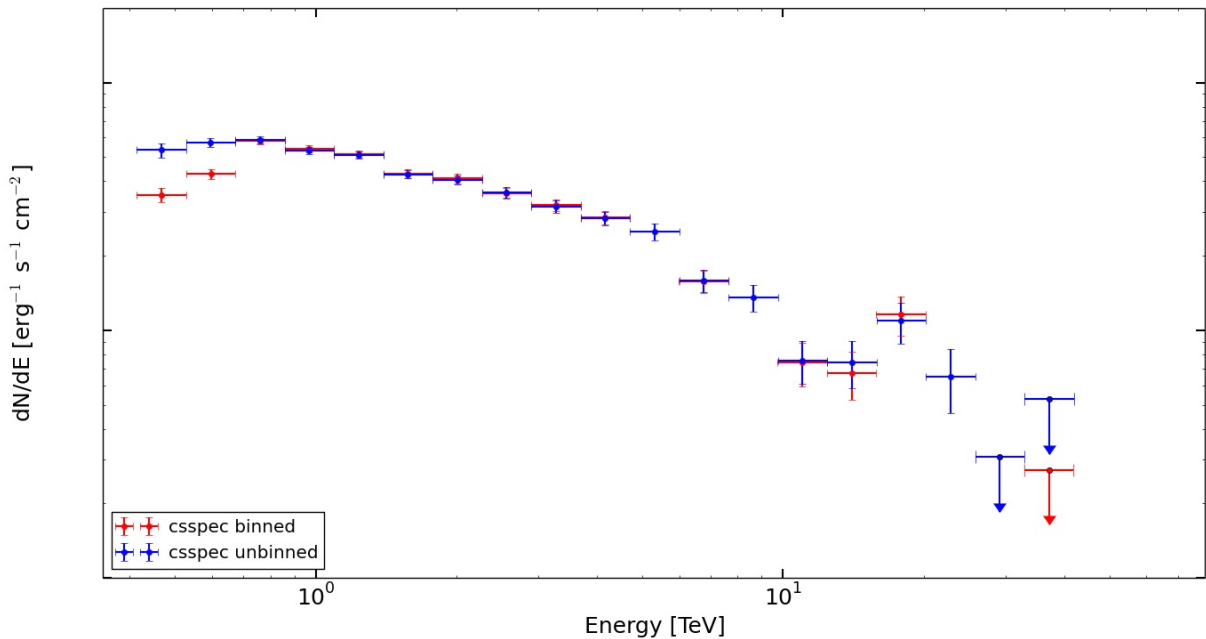
tools for binned analysis should consistently fill cubes

07/17/2015 02:22 PM - Mayer Michael

<b>Status:</b>	Closed	<b>Start date:</b>	07/17/2015
<b>Priority:</b>	Immediate	<b>Due date:</b>	
<b>Assigned To:</b>	Mayer Michael	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>	1.0.0		

Description

When comparing csspec for binned and unbinned analysis I stumbled upon an issue which needs to be fixed. Please see attached plot.



It turns out that ctexpcube, ctpsfcube and ctbkgcube don't check if the energy bin to be filled is actually inside the observation energy boundaries. When combining lots of observations with different energy thresholds, this leads to an overestimation in exposure at low energies (where only parts of the runs contribute). Accordingly, the flux in the low energy bins is underestimated quite significantly. Therefore a check should be implemented in these tools and only fill cubes if the bin is contained in the energy boundaries.

In addition, I realised that the procedure how ctbin fills the cube is a bit different how ctmodel does it. Therefore, one might expect residual maps which do not perfectly fluctuate around zero.

- ctbin fills an event into the cube if the event is inside the cube range (which makes sense)
- ctmodel only fills the cube if the GCTAEventBin is contained within the observation energy boundaries. It asks for obs\_ebounds.contains(bin->energy()).

So in the worst case, ctbin fills a bin with some events, while ctmodel doesn't since obs\_ebounds.contains(bin->energy()) evaluates to false. Accordingly, the residual map is slightly wrong.

To summarise:

In my view, and as we discussed during the coding sprint, bins that are not completely contained inside the observation energy boundaries should not be considered in the binning ctools: ctbin, ctmodel, ctexpcube, ctpsfcube, ctbkgcube.

History

#1 - 07/17/2015 02:31 PM - Mayer Michael

- Description updated

## #2 - 07/21/2015 10:35 AM - Mayer Michael

- File `spectrum_fixed.png` added
- Subject changed from tool for binned analysis should consistently fill cubes to tools for binned analysis should consistently fill cubes
- Status changed from New to Pull request
- Assigned To set to Mayer Michael
- % Done changed from 0 to 100

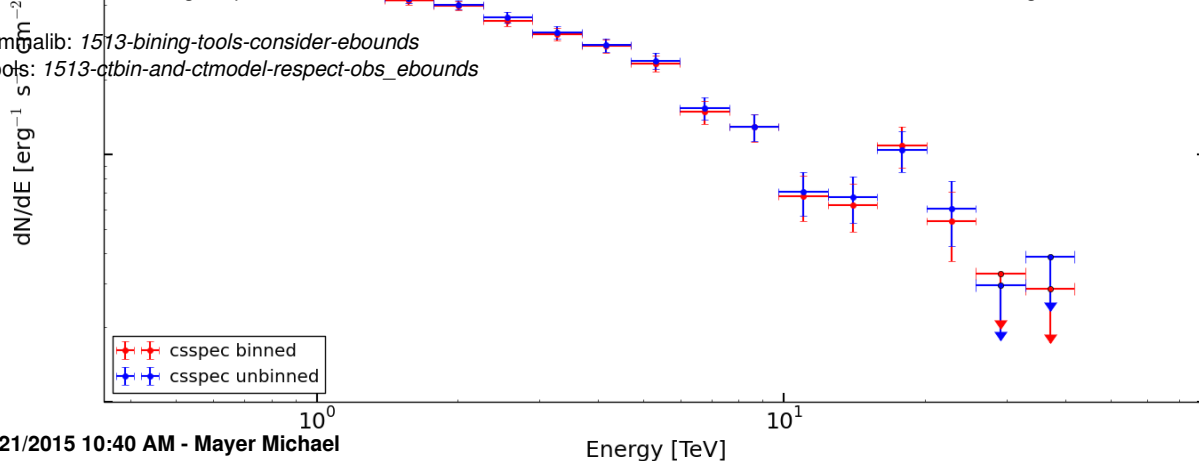
I think I have fixed the issue (see attached plot). I create a new function: `GEbounds::is_in_range(const GEnergy& emin, const GEnergy& emax)` - we can argue about the name :)

This function checks if an energy range (e.g. `emin` and `emax` of an energy bin) is fully contained inside the `GEbounds`-object.

I've modified the tools `ctbin`, `ctmodel`, `ctexpcube`, `ctpscube` and `ctbkcube` such that this function is used coherently when filling the cubes. Bins that are not fully contained inside the energy boundaries of the considered `GCTAObservation` are hence neglected in the same way. Therefore all cubes for stacked analysis are now filled consistently. Spectra and residual maps look satisfactorily.

I also found some minor bugs/improvements in `csspec` which are corrected now. Please find the code on the following branches:

- `gamlib: 1513-bining-tools-consider-ebounds`
- `ctools: 1513-ctbin-and-ctmodel-respect-obs_ebounds`



## #3 - 07/21/2015 10:40 AM - Mayer Michael

Note: I had some problems with fit convergence in some energy bins when using binned analysis. These problems disappeared when fixing the index of the background model in each energy bin. Maybe in the long run, we might want to change the output of `ctbkcube` to a `GModelSpectralConstant` or at least make it steerable by the user.

Anyhow, could you merge the change?

## #4 - 07/21/2015 10:42 AM - Knödlseider Jürgen

Thanks for pointing out that problem.

I'll merge in your changes soon (I'm currently distracted by some other duties but will soon come back to merge in the pull requests).

## #5 - 07/21/2015 10:46 AM - Mayer Michael

Sure, no worries. Thanks for the feedback.

## #6 - 07/23/2015 11:12 PM - Knödlseider Jürgen

I renamed `GEbounds::is_in_range()` into `GEbounds::contains()` which is semantically better. There are thus now two variants of the `GEbounds::contains()` method: one for a single energy and one for an energy bin. The latter does apply in principle only for contiguous energy bins. In all practical cases the bins are indeed contiguous, but in the future there could be exceptions. I created an issue for this to not forget about that (#1516).

I merged the GammaLib change into devel. Now work on ctools.

**#7 - 07/23/2015 11:36 PM - Knödlseeder Jürgen**

- Status changed from Pull request to Feedback

I now also merged the ctools change into devel.

Note that I also set the default anumbins in csspec to 200 (otherwise the PSF will get badly sampled), but I reduced the spatial sampling by a factor 20 (simply by dividing the number of spatial bins by 20 and increasing the binsize by 20).

You may cross-check if this works properly.

**#8 - 07/27/2015 10:43 AM - Mayer Michael**

Ok, I agree it makes sense to reduce the spatial binning instead of the anumbins parameter. I am not sure however, if hard-reducing this quantity by a factor 20 is reasonable. Unit tests now fail since we use only 10 spatial bins in csspec unit tests. Therefore  $10 / 20 = 0$ , which throws an exception. Would it make sense to introduce something like a PSF binning reduction factor as parameter? Could be set to 5 on default (wouldn't cause problems in the unit tests), but of course could be increased by the user.


**#9 - 07/28/2015 04:33 PM - Mayer Michael**

I don't know if I forgot to commit that but apparently ctmodel doesn't check if a bin is contained in the observation Rol anymore. I guess we should add support for this as well. Could you add support for this on devel?


**#10 - 07/29/2015 06:24 PM - Mayer Michael**

- Status changed from Feedback to Pull request

I realised that ctmodel should be fine in stacked mode since the model value always depends on the exposure (which is zero outside Rol boundaries). However, when passing an unbinned observation to ctmodel, the Rols should be considered. Individual bins should not be filled in case the bin is not inside the observation Rol.

I have made the change to support this on branch *1513-ctmodel-considers-Rol*. Please merge 

**#11 - 08/07/2015 08:59 AM - Mayer Michael**

just a quick reminder to merge the code on branch *1513-ctmodel-considers-Rol*  


**#12 - 08/12/2015 02:19 PM - Mayer Michael**

btw: I added some improvements for chessobs on this branch, too.

**#13 - 08/20/2015 10:31 PM - Knödlseeder Jürgen**

- Status changed from Pull request to Closed

Finally I'm back from vacation. Sorry for having to wait so long for the pull request. It's just great to disconnect for some week in summer :-)

Your change is now merged into devel.

**Files**

spectrum.jpg	80.3 KB	07/17/2015	Mayer Michael
spectrum_fixed.png	53.5 KB	07/21/2015	Mayer Michael