

ctools - Change request #1757

cssens should respect previously applied energy cuts (and thresholds)

04/08/2016 11:31 AM - Mayer Michael

Status:	In Progress	Start date:	04/08/2016
Priority:	Normal	Due date:	
Assigned To:		% Done:	0%
Category:		Estimated time:	0.00 hour
Target version:			
Description			
<p>I want to compute a sensitivity curve for a large observation container. The observation container was preselected according to given energy thresholds.</p> <p>When running cssens in differential mode, the function <code>set_obs_ebounds</code> simply overwrites the existing energy boundaries of each observation. It should however, respect previous energy cuts and omit observations when their energy range does not contribute in the differential sensitivity bin.</p>			

History

#1 - 05/12/2016 01:57 PM - Mayer Michael

- Status changed from New to Pull request

I have implemented the change on branch `1757-cssens-respect-ebounds`.

#2 - 06/03/2016 12:38 AM - Knödseder Jürgen

- Tracker changed from Bug to Change request

- Status changed from Pull request to In Progress

Sorry for not looking into this earlier.

I had a look into your code and I was wondering whether it actually does what you want. The point is that `_set_obs_ebounds()` is called for every energy bin `[emin,emax]`, and after the first call to the method the original observation energy boundary information is lost. I think for the code to work properly one would need to store all observation energy ranges somewhere, and use this stored information in every call.

#3 - 06/14/2016 04:19 PM - Mayer Michael

Indeed, I haven't considered that.

I slightly updated the code on the respective branch. The class `cssens` now contains a new member `self._obs_ebounds = []` that stores a list of the individual `ebounds` objects from the observation container. The list is filled right after the `_get_parameters` call. At the end of the function `_get_sensitivity()`, I restore now all energy boundaries by the following loop:

```
for i, obs in enumerate(self._obs):
    obs.events().ebounds(self._obs_ebounds[i])
```

I am still wondering if the enumeration is safe enough to ensure the `ebounds` and `observations` objects are not mixed up. As an alternative, we could use a Python dictionary that stores energy boundaries by `obs_id` or `name` keys. What do you think?