ctools - Change request #1351

ctselect should respect previously applied cuts

10/30/2014 03:03 PM - Mayer Michael

Status:	Closed		**Start date:**	10/30/2014
Priority:	Normal		**Due date:**	
Assigned To:	Knödlseder Jürgen		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:	1.0.0			

Description

Currently, it is possible to use ctselect to cut on a certain energy range (say 1-10 TeV). If the user makes this cut and forgets about it, it is possible that she/he applies another cut (say 0.5-50 TeV) later. The header keywords are updated accordingly and, more important, the GEbounds-definition. The ebounds definition is then used in the likelihood fit to apply the energy range.
The problem now is that the user has already thrown away every event outside [1,10] TeV. Accordingly, the likelihood fit does not find events outside this range. However, the IRFs might indicate high sensitivity in that region (the fitter thinks we are analysing from 0.5-50 TeV). Accordingly, the fit might get mixed up between zero source flux in the low energy range and signal between 1-10TeV. The problem is the same for the RoI and GTI cuts.
Accordingly, ctselect should be able to read previously applied cuts and only cut further if the new cuts are contained inside the old ones.

History

#1 - 10/30/2014 05:44 PM - Knödlseder Jürgen

- Target version set to 1.0.0

Fully agree that this needs to be corrected. I think we need that for release 1.0 to make the code save.

#2 - 10/30/2014 11:28 PM - Knödlseder Jürgen

- Status changed from New to Feedback

- Assigned To set to Knödlseder Jürgen

- % Done changed from 0 to 100

Implemented and merged into devel branch. I also implemented a check on the ROI so that the radius is automatically reduce if the new ROI does not fully overlap with an existing one. I did some quick checks, but I leave it on Feedback to get confirmation from your side on the proper functioning.

#3 - 10/31/2014 02:54 PM - Mayer Michael

I also made some quick checks. They all passed my tests, where I was cutting several times on the same eventlist. All results look reasonable.

The only minor thing is that ctselect::select_events() now becomes quite longish. But I have no idea where and how we could separate functionality there. This shouldn't be important towards 1.0

#4 - 10/31/2014 03:41 PM - Knödlseder Jürgen

Mayer Michael wrote:

> I also made some quick checks. They all passed my tests, where I was cutting several times on the same eventlist. All results look reasonable.
>
> The only minor thing is that ctselect::select_events() now becomes quite longish. But I have no idea where and how we could separate functionality there. This shouldn't be important towards 1.0

I started splitting things off, but then I recognized that I needed the event selection parameters later for setting the event list, hence I put the code back together ...

#5 - 02/06/2015 05:20 PM - Knödlseder Jürgen

I guess we can close this one now?

#6 - 02/06/2015 05:23 PM - Mayer Michael

Yes. This works quite well now.

#7 - 02/06/2015 05:38 PM - Knödlseder Jürgen

- Status changed from Feedback to Closed