

ctools - Bug #1450

ctselect doesn't allow observations with zero events

03/25/2015 05:46 PM - Mayer Michael

Status:	Closed	Start date:	03/25/2015
Priority:	Normal	Due date:	
Assigned To:	Mayer Michael	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:	1.0.0		

Description

Using ctselect with the ethres=DEFAULT parameter, results in an observation list with various energy ranges. This works fine. If one however subsequently runs ctselect with a small energy range, where only few runs contribute with events, an exception is thrown:

```
*** ERROR in GEbounds::insert_eng(int&, GEnergy&, GEnergy&): Invalid argument. Invalid energy interval specified. Minimum energy 607.552 GeV can not be larger than maximum energy 200 GeV.
*** ERROR encountered in the execution of ctselect. Run aborted ...
```

This can be reproduced by running csspec on with a larger energy range than the actual preselected observations. The problem occurs in ctselect::set_ebounds(). At the end of this function, we create GEbounds with emin and emax. If e.g. emin was adjusted to match previous cuts it may happen that emin>emax leading to the above exception while creating the ebounds. My suggestion is to add a check when creating the ebounds:

```
// Set selection energy boundaries
GEbounds result;

if (emin > emax) {
    result.append(GEnergy(emin,"TeV"), GEnergy(emin,"TeV"));
}
else {
    result.append(GEnergy(emin, "TeV"), GEnergy(emax, "TeV"));
}

// Return result
return result;
```

Note that in case emin>emax, we could use a GEbounds with [emin, emin]. This circumvents the exception and assures the selection of zero events. This zero energy range also allows to predict zero events, e.g. for ctmodel, etc.

However, on the gammalib-level, an exception is then thrown:

```
*** ERROR encountered in the execution of ctlike. Run aborted ...
*** ERROR in GObservation::npred_spec(GModel&, GTime&): Invalid energy range (Emin=0 MeV, Emax=0 MeV) specified.
```

This can easily be fixed by allowing emin=emax in GObservation::npred_spec(). Accordingly, we could change if(emin to if(emin<=emax) in GObservation::npred_spec(). I guess allowing npred=0 in case the energy width is 0.0 makes more sense anyway

History

#1 - 03/25/2015 05:53 PM - Mayer Michael

- Status changed from New to Pull request
- Assigned To set to Mayer Michael
- Target version set to 1.0.0
- % Done changed from 0 to 100

Fix is available on branch *adjust-ctselect-to-allow-runs-without-events* in ctools (forgot to add the issue number). On the gammalib-level, in line 1394 of GObservation.cpp "" just has to be changed to "<".

#2 - 03/25/2015 07:06 PM - Knödseder Jürgen

I'm wondering whether all this is very clean, or whether it would be better in `ctselect::set_ebounds()` to just return an empty energy boundaries object, hence

```
// Set selection energy boundaries
GEbounds result;
if (emax > emin) {
    result.append(GEnergy(emin, "TeV"), GEnergy(emax, "TeV"));
}

// Return result
return result;
```

In `GObservation::npred_spec` would could then change the code to simply return 0 if no energy boundaries exist. But maybe there are further implications.

#3 - 03/26/2015 07:07 PM - Mayer Michael

This should do it, too. I agree the other method was a bit dirty. I quickly ran make check which also did not throw an error.

#4 - 03/30/2015 02:31 PM - Mayer Michael

I was wondering if it was even cleaner to remove the observation from the container? But nevertheless, I also think that returning an empty GEbounds-object should be sufficient. Could you make the changes?

#5 - 04/10/2015 10:50 PM - Knödseder Jürgen

- *Status changed from Pull request to Closed*

I made the changes and merged the code into devel.