ctools - Feature #1996

ctselect - allow phase selection through parameters

04/09/2017 07:36 PM - Di Venere Leonardo

Status:	Closed	**Start date:**	04/09/2017
Priority:	Normal	**Due date:**	
Assigned To:	Di Venere Leonardo	**% Done:**	100%
Category:		**Estimated time:**	0.00 hour
Target version:	1.3.0		

Description

ctselect now doesn't allow phase selection unless specified in the expr parameter. Two new parameters could be added to handle this selection specifically.
Updates to gammalib from issue #1982 will be probably necessary.

History

#1 - 04/11/2017 10:14 AM - Di Venere Leonardo

- Status changed from New to In Progress

- % Done changed from 0 to 50

I've added the phasemin and phasemax parameters as optional parameters (default value = "NONE") and added the phase selection in select_events method using the column "PHASE" in the event list.
A check on the existence of the "PHASE" column in the event list is still missing.
A method to get phase information from event list is necessary in the GCTAEventList class, which has been implemented for issue #1982.

#2 - 04/11/2017 10:38 AM - Knödlseder Jürgen

Thanks for getting going on this issue.

The GCTAEventAtom class has a phase() method to get and set phase information.

GCTAEventList needs has_phase() setter and getter methods that set and return the m_has_phase member. This should then allow ctselect to check whether phase information is present in an event list.

We probably should also add data sub-space keywords to the FITS file so that we keep track of the phase selection. See

- https://confluence.slac.stanford.edu/display/ST/Data+SubSpace+keywords
- https://confluence.slac.stanford.edu/download/attachments/1208/SDS07.pdf?version=1&modificationDate=1380814716000&api=v2

A proposal would be:

```
DSTYPx  = 'PHASE   '
DSUNIx  = 'DIMENSIONLESS'
DSVALx  = '0.0:0.3'
```

or if there are multiple phase cuts

```
DSTYPx  = 'PHASE   '
DSUNIx  = 'DIMENSIONLESS'
DSVALx  = '0.0:0.3,0.5:0.6,0.9:1.0'
```

This means that the phase selection information has to be stored in GCTAEventList, since the data sub-space keywords are read in the GCTAEventList::read() method and written by the GCTAEventList::write_ds_keys() method. Probably two vectors

```
std::vector<std::double> m_phase_min;
std::vector<std::double> m_phase_max;
```

should be added for that to the GCTAEventList class.

This brings me to the point that the phasemin and phasemax parameters do not allow specifying multiple phase intervals. Multiple phase intervals are however often necessary due to the wrap around of the phase. I would hence suggest to add a single phase string parameter to ctselect so that multiple phases can be specified (in the format 0.0:0.3,0.5:0.6,0.9:1.0).

#3 - 04/12/2017 04:31 PM - Di Venere Leonardo

Thanks for the useful suggestions.
I have added the check on the existence of the PHASE column using the GCTAEventList has_phase() method. At the moment I implemented a selection on a single phase range using the 'phasemin' and 'phasemax' parameters. I've changed the default values of these parameters to 0.0 and 1.0 respectively. This allows a selection also when only one of the two parameters is specified.

Still to be implemented:

1. Add a new DSS keyword to keep track of the cuts. If I understood correctly, this needs the two additional values (or vectors to allow multiple selection) in the event list, which need to be updated when the selection is performed. I wonder if these fields could be generalized to any instrument EventList, adding them directly in the GEvents class, instead of GCTAEventList.
2. Allow multiple phase selection as you suggested.

These are the details of the branch on which I'm committing my changes.
CTools: git@cta-gitlab.irap.omp.eu:ldivenere/ctools.git
CTools branch: 1996-ctselect-add-phase-selection

#4 - 04/13/2017 09:12 AM - Di Venere Leonardo

- File selected_events_phase.png added

- File selected_events1_phase.png added

- File selected_events2_phase.png added

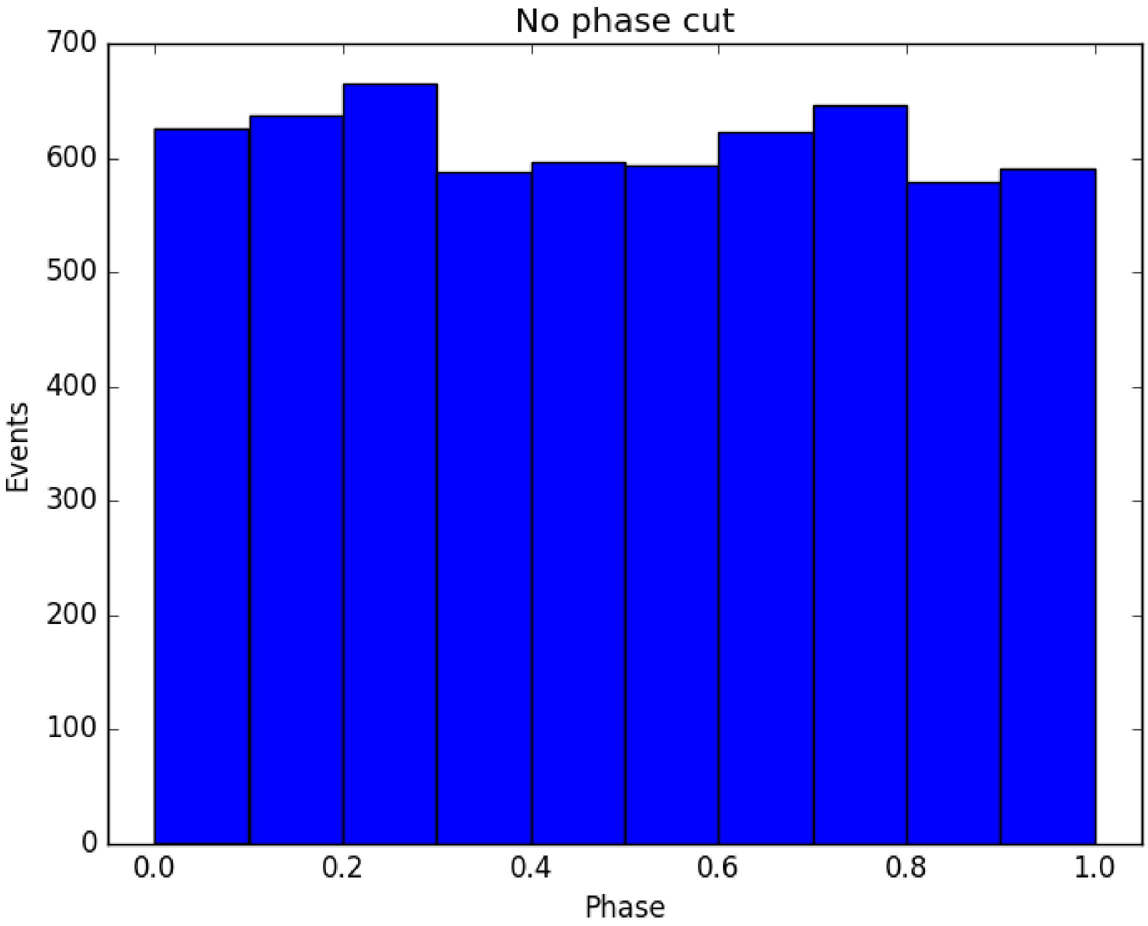
- % Done changed from 50 to 70

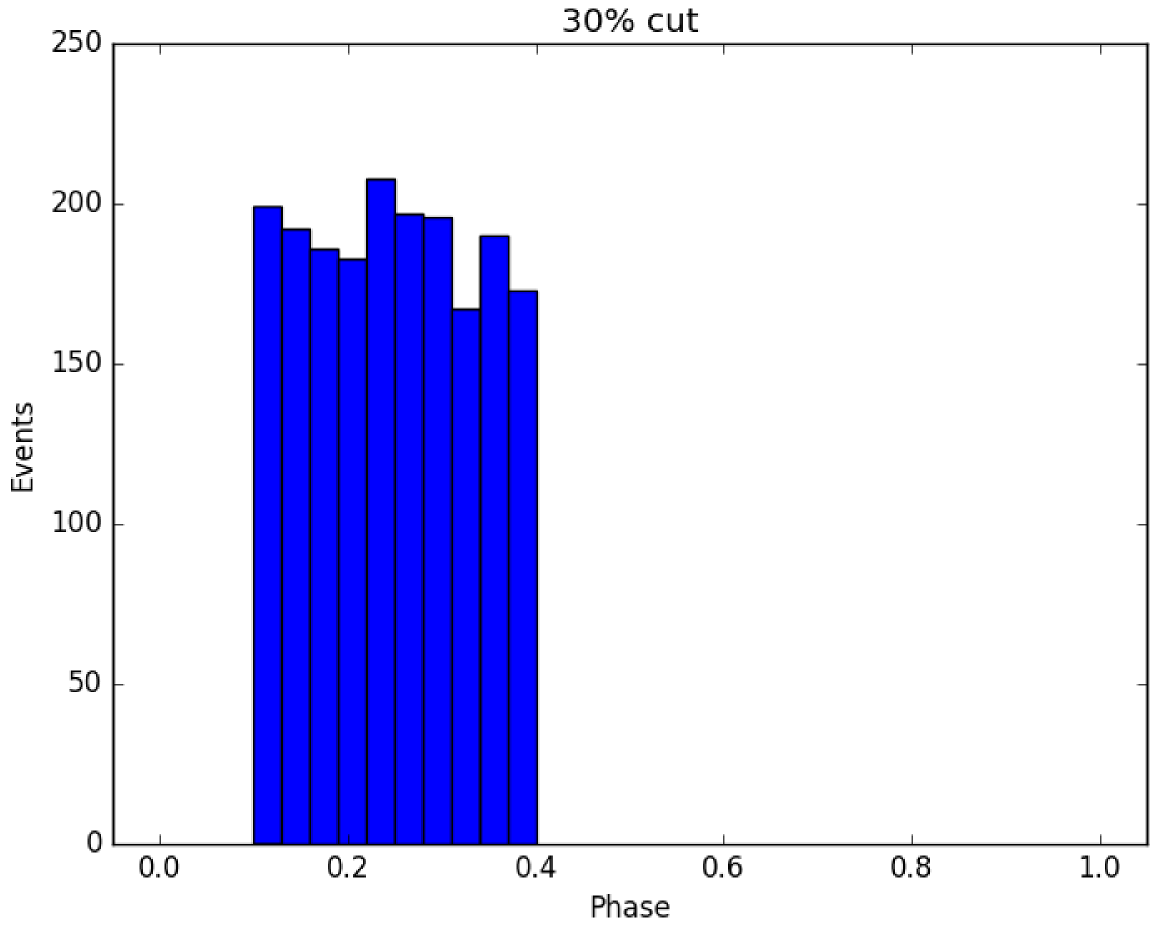
I have added the multiple phase selection. Now there is one parameter called "phase", which takes a string in the format: "phasemin0:phasemax0,phasemin1:phasemax1".
I have also added a python test in test/test_ctselect.py, in which a phase selection is performed. This test needs an input file with a PHASE column. So I added in the test/data folder an additional event list file (called "phase_crab_events.fits.gz"), generated applying ctphase to the file "crab_events.fits.gz".

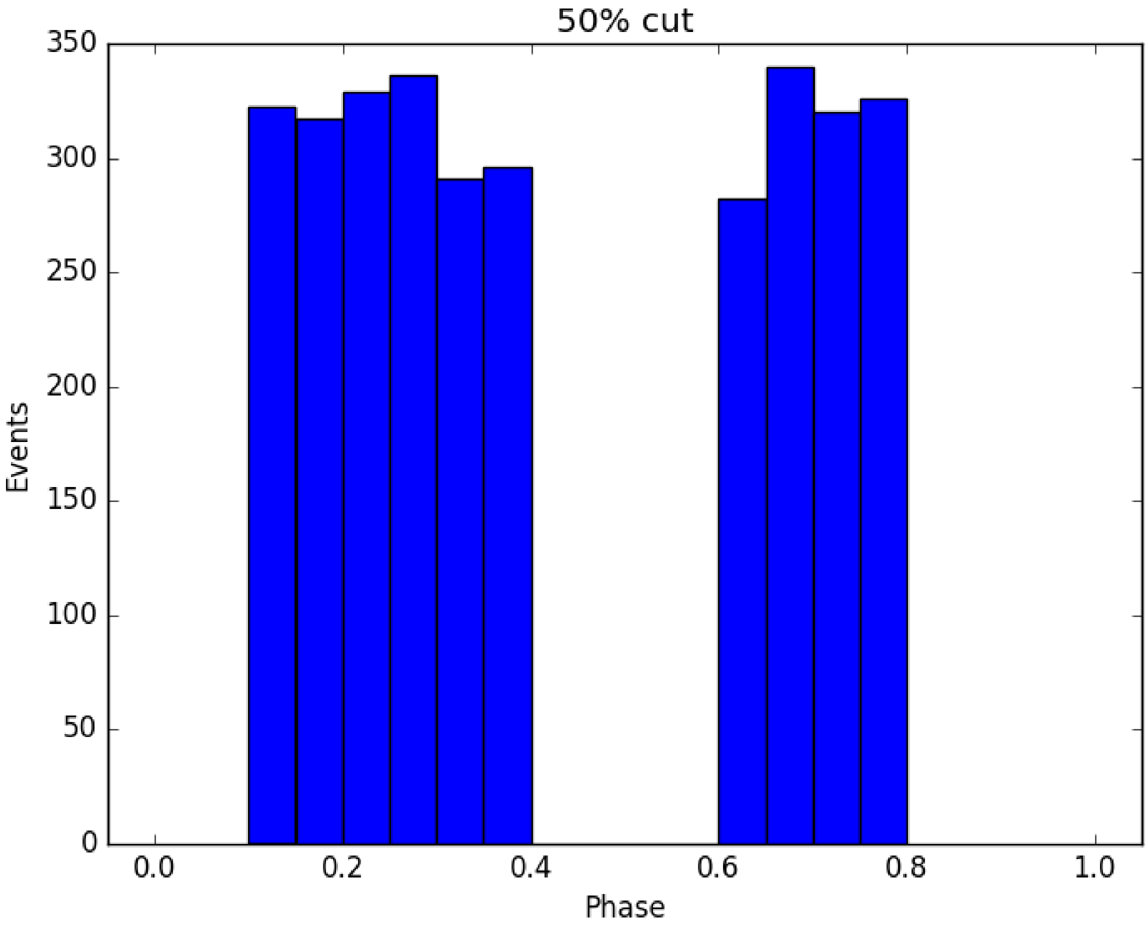
These are the results obtained applying ctselect to the file "crab_events.fits.gz" with different phase cuts:

- no phase selection -> nevts = 6141
- phase = "0.1:0.4" -> nevts = 1891 (approx. equal to 30% of 6141)
- phase = "0.1:0.4,0.6:0.8" -> nevts = 3159 (approx. equal to 50% of 6141)

The following plots show the phase distribution of events in the three cases described above.


No phase cut
Events
Phase
0
100
200
300
400
500
600
700
0.0
0.2
0.4
0.6
0.8
1.0


30% cut
Events
Phase
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
1.0


50% cut
Events
Phase
0
50
100
150
200
250
300
350
0.0
0.2
0.4
0.6
0.8
1.0

#5 - 04/13/2017 04:08 PM - Knödlseder Jürgen

- Status changed from In Progress to Closed

- % Done changed from 70 to 100

Thanks Leonardo, this looks great.

I integrated your code into devel.

I added some checks on the phase parameter in the ctselect::get_parameters() method. I also added the possibility to have a wrap around phase selection, i.e. "0.8:0.2" will select events from the intervals [0.8,1.0] and [0.0,0.2].

#6 - 04/13/2017 04:22 PM - Di Venere Leonardo

- File selected_events1_phase_DSSkey.png added

- File selected_events2_phase_DSSkey.png added

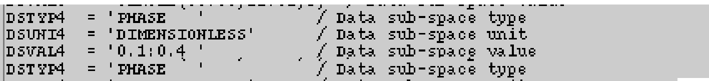
Thanks Jurgen. In the meanwhile, I have implemented new methods in gammalib to write an additional DSS keyword in the header of the fits file, corresponding to the PHASE selection. Here are the details of the classes I modified:

- in GCTASupport I added a read_ds_phase() method, to read the PHASE keyword from the header
- in GCTAEventList I added two new members (m_phasemin and m_phasemax) defined as double vectors to store boundaries of phase intervals; I also added new methods to access these members from outside the class
- in GCTAEventList I modified the read() and write_ds_keys() methods to read and write the new keywork respectively

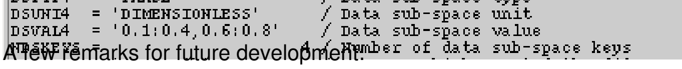
Finally, in ctools:

- in ctselect::select_events() I added a few lines to update the m_phasemin and m_phasemax members of the GCTAEventList before writing out the file
- (in get_parameters() I also added a few checks on the string passed as a phase parameter, this may be in conflict with your last changes!)

The following images show the header files of the selected events generated with the phase keyword as described in my previous comment.



```
DSTYP4  = 'PHASE   '           / Data sub-space type
DSUNI4  = 'DIMENSIONLESS'      / Data sub-space unit
DSVAL4  = '0.1:0.4 '           / Data sub-space value
```



```
DSTYP4  = 'PHASE   '           / Data sub-space type
DSUNI4  = 'DIMENSIONLESS'      / Data sub-space unit
DSVAL4  = '0.1:0.4,0.6:0.8'    / Data sub-space value
NDSKEYS =                    4 / Number of data sub-space keys
```

A few remarks for future development:

1. I implemented the phase selection methods in GCTAEventList, which means that this will not work for other instruments. A possible improvement to generalize this type of selection to other instruments could be done by moving all the phase-related members to the GEvents class. This would be the same approach already adopted for the energy boundaries.
2. At the moment, the boundaries of the phase intervals are stored in two arrays of double. This approach is slightly different from what was done for the energy boundaries. In this latter case, the boundaries are stored in another class object called GEbounds. In the case of phase boundaries, another class should be created (GPhaseBounds ?) and all the new methods I implemented should then be updated. This could be a future improvement, if we want to generalize this phase selection also to other instruments.

I actually don't know if these improvements are necessary. They will need some effort (especially the second one) and I'm not sure how useful they could be.

These are the details of the ctools and gammalib branches.
CTools: git@cta-gitlab.irap.omp.eu:ldivenere/ctools.git
CTools branch: 1996-ctselect-add-phase-selection
Gammalib: git@cta-gitlab.irap.omp.eu:ldivenere/gammalib.git
Gammalib brach: 1996-ctselect-add-phase-info-GCTAEventList

#7 - 04/13/2017 04:35 PM - Knödlseder Jürgen

- Status changed from Closed to In Progress

I was too quick closing the issue wink.png Thanks Leonardo for the new code ...

I now wait with any further merging until you issue a pull request.

#8 - 04/13/2017 04:43 PM - Di Venere Leonardo

- Status changed from In Progress to Pull request

#9 - 04/14/2017 10:20 PM - Knödlseder Jürgen

I'm about to merge in the code.

Currently, a std::vector< std::vector<double> > structure is used to transport the phase information which is not very flexible and makes the interface complicated. I therefore implemented a new GPhases class that holds the phase interval information and provides methods for appending, removing, and accessing phase intervals. In the future, the class can be expanded so that all event lists have phase intervals as boundaries, and the phase cut information can then even be used for a transparent handling of the model normalization. For the moment, however, the class is mainly used to simplify the interface.

#10 - 04/14/2017 11:19 PM - Knödlseder Jürgen

- Status changed from Pull request to Closed

Everything is now merged into devel

#11 - 04/15/2017 08:31 AM - Di Venere Leonardo

Great. I agree that having a new class to handle phase intervals is the best solution. Thanks for implementing it.

Files

selected_events1_phase.png	16.3 KB	04/12/2017	Di Venere Leonardo
selected_events_phase.png	19.5 KB	04/12/2017	Di Venere Leonardo
selected_events2_phase.png	18.2 KB	04/13/2017	Di Venere Leonardo
selected_events1_phase_DSSkey.png	2.19 KB	04/13/2017	Di Venere Leonardo
selected_events2_phase_DSSkey.png	2.17 KB	04/13/2017	Di Venere Leonardo