URL: http://cxc.harvard.edu/ciao3.4/set_groups.html
Last modified: December 2006

AHELP for CIAO 3.4

set_groups

Context: sherpa

Jump to: Description Examples Bugs See Also

Synopsis

Module functions for grouping and setting quality to source and background files from an array.

Syntax

```
Integer_Type set_groups([Integer_Type,]Array_Type)
Integer_Type set_bgroups([Integer_Type,]Array_Type)
Integer_Type set_quality([Integer_Type,]Array_Type)
Integer_Type set_bquality([Integer_Type,]Array_Type)

Success/Error Return Values: 1/0

Arguments:

(1) Dataset number (default 1).

(2) An array defining the grouping or quality and it should be of the
same length as the input data set. This argument is required.

Note that if only one argument is provided, it is assumed to be an
array, and the dataset is assumed to be dataset 1.
```

Description

The set_(b)groups and set_(b)quality functions allow the user to set new groups or quality for source and background datasets.

The input array is a Integer_Type array of the same length as the input (ungrouped) dataset for which a new grouping or quality scheme is to be defined. An input grouping array element set to −1 marks the beginning of a group, while an input array element set to 1 marks members of that group, so the corresponding bins are treated as one during fitting.

A quality array contains the quality flags for each group: 0 for good (grouped) data; 5 for data labeled as bad by the user (within a tab), and 2 for data labeled as questionable by dmgroup (incomplete groups, etc.).

The grouping and quality definitions are based on OGIP standard.

See the related Sherpa commands GROUP and QUALITY for more information.

Example 1

Apply a grouping definition from the other PHA file to the data set.

```
sherpa> DATA spec.pha
sherpa> show
....
-----------------
Input data files:
-----------------

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

.......

sherpa> g=readfile("grouped.pha")
sherpa> print(g)
_filename        =  grouped.pha
_path            =  /data
_filter          =  NULL
_filetype        =  7
_header          =  String_Type[361]
_exptime         =  13804
_ncols           =  13
_nrows           =  1024
channels         =  Float_Type[596]
counts           =  Float_Type[596]
grouping         =  Integer_Type[1024]
qualityflags     =  Integer_Type[596]
phachans         =  Integer_Type[596]
areascal         =  1
backscal         =  4.2132e-07
errors           =  Float_Type[596]
background       =  /data/bg.pha
arf              =  /data/spec.arf
response         =  /data/spec.rmf
numgroups        =  596
numchans         =  1024

sherpa> set_groups(1,g.grouping)
sherpa> show
...
-----------------
Input data files:
-----------------

Data 1: spec.pha pha.
Total Size: 596 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231
.....
```

In this example, ungrouped and group data are read into Sherpa, and then group information is retrieved from the other file, by reading the file first into a variable g using readfile. A new array g.grouping can be apply to the ungrouped data set. g.grouping is defined by the grouping in the dataset 2. This grouping scheme is then apply to the ungrouped data set with set_groups.

Example 2

Apply a grouping scheme from the other data set to ungrouped data.

```
sherpa> DATA spec.pha
sherpa> show
....
−−−−−−−−−−−−−−−−−
Input data files:
−−−−−−−−−−−−−−−−−

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

.......
sherpa> DATA 2 spec_grp.pha
sherpa> show
....
−−−−−−−−−−−−−−−−−
Input data files:
−−−−−−−−−−−−−−−−−

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231


......

Data 2: spec_grp.pha pha.
Total Size: 131 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231
......
sherpa> g=get_groups(2)
sherpa> print(g)
1
−1
−1
−1
−1
−1
−1
−1
−1
−1
−1
−1
−1
−1
−1
−1
1
−1
.....
sherpa> set_groups(1,g)
WARNING: any applied filters are being deleted!
1
sherpa> show

Data 1: spec.pi pha.
Total Size: 131 bins (or pixels)
```

```
Dimensions: 1
Total counts (or values): 2231
.....
```

In this example, ungrouped and group data are read into Sherpa, and then group information is retrieved using get_groups from the grouped dataset. A new array g is defined whose elements are defined by the grouping of the dataset 2. This grouping scheme is then apply to the ungrouped data set with set_groups.

Example 3

Apply a grouping scheme from the source data set to ungrouped background data.

```
sherpa> DATA spec.pha
sherpa> show
....
-----------------
Input data files:
-----------------

Data 1: spec.pha pha.
Total Size: 1024 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

 Background 1: bg.pha pha.
  Total Size: 1024 bins (or pixels)
  Dimensions: 1
  Total counts (or values): 662

.......

sherpa> g=get_groups(1)
sherpa> print(g)
1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
-1
1
-1
.....
sherpa> set_bgroups(1,g)
WARNING: any applied filters are being deleted!
1
sherpa> show

Data 1: spec.pi pha.
Total Size: 131 bins (or pixels)
Dimensions: 1
Total counts (or values): 2231

 Background 1: bg.pha pha.
 Total Size: 131 bins (or pixels)
  Dimensions: 1
  Total counts (or values): 662
......
```

In this example, group data and ungrouped background data are read into Sherpa, and then group information is retrieved using get_groups from the grouped dataset. A new array g is defined whose elements are defined by the grouping of the source data set. This grouping scheme is then apply to the ungrouped background data set with set_bgroups.

Bugs

See the Sherpa bug pages online for an up−to−date listing of known bugs.

See Also

sherpa

analysis, get_groups, ignore, notice, set_filter, set_ignore, set_ignore2d, set_ignore_all, set_ignore_bad, set_notice, set_notice2d, set_notice_all

The Chandra X−Ray Center (CXC) is operated for NASA by the Smithsonian Astrophysical Observatory.
60 Garden Street, Cambridge, MA 02138 USA.

URL: http://cxc.harvard.edu/ciao3.4/set_groups.html
Last modified: December 2006