Chandra X-ray Center

AHELP for CIAO 3.4   **get_flux2d**   Context: sherpa

Jump to: Description Examples Bugs See Also

Synopsis

Module functions for computing fluxes/summing counts in 2–D images.

Syntax

```
Struct_Type get_pflux2d([Integer_Type[,String_Type[,String_Type]]])
Struct_Type get_eflux2d([Integer_Type[,String_Type[,String_Type]]])
Struct_Type
get_mcounts_sum2d([Integer_Type[,String_Type[,String_Type]]])
Struct_Type get_dcounts_sum2d([Integer_Type[,String_Type]])

Error Return Values: NULL

Arguments:

(1) Dataset number (default 1)

(2) Evaluation point, or 2-D region descriptor (default use all data)

(3) Model component or stack name (default use all appropriate models)
```

Description

These functions retrieve the photon flux, energy flux, summation of convolved predicted model counts, and summation of observed counts respectively. The flux units depend upon the units of the exposure map but may not be correct in the current version of Sherpa, so use caution.

The structure output by these functions contains the following fields:

2–D Flux Structure Fields

Field	Description
dataset	the dataset for which the flux is evaluated/counts summed
range	the single point at which the flux is computed/counts summed, or the range over which the flux is integrated/counts summed; if NULL, the integral is done over the entire dataset range
comp	the model stack or component for which the flux is computed/counts summed; if NULL, the whole source/bg stack is used
value	the computed flux value/sum of counts
units	the flux units (NULL for counts)

In the current version of the Sherpa/S–Lang module, there are no functions for computing fluxes/summations of counts for 2–D background datasets.

Also, in the current version of the Sherpa/S–Lang module, there is no way to specify a coordinate system when specifying a 2–D region descriptor; logical coordinates are assumed.

See the related Sherpa commands FLUX, EFLUX, MCOUNTS, and DCOUNTS for more information.

Example 1

Fit a 2–D Gaussian to data without an exposure map; compute photon fluxes and summed model counts (which will be equal in this simplistic example):

```
sherpa> DATA example_img.fits
sherpa> NOTICE FILTER circle(247,257,20)"
sherpa> PARAMPR OFF
sherpa> SOURCE = GAUSS2D[g]
sherpa> FIT
...
sherpa> print(get_pflux2d().value)
3523.46
sherpa> print(get_mcounts_sum2d().value)
3523.46
```

Example 2

Include the exposure map in fitting the data, filter the data, define the source model and fit; obtain the photon flux.

```
sherpa> data img_bin.fits
sherpa> farf2d[em1]
em1.file parameter value ["none"] expmap.fits
em1.norm parameter value [1]

sherpa> instrument=em1
               # include the exposure map in the instrument stack

sherpa> notice filter "rotbox(4022.825,4119.3,15.2,14.95,0)"

sherpa> foo=get_eflux2d
sherpa> print(foo)
NULL               # source model has not been defined

sherpa> source=gauss2d[g2]
g2.fwhm parameter value [10.8243]
g2.xpos parameter value [294.5]
g2.ypos parameter value [306.5]
g2.ellip parameter value [0]
g2.theta parameter value [0]
g2.ampl parameter value [30]

sherpa> freeze g2.xpos g2.ypos
sherpa> fit
 LVMQT: V2.0
 LVMQT: initial statistic value = 4.14528e+12
 LVMQT: final statistic value = 2625.34 at iteration 5
          g2.fwhm  1.03876
          g2.ampl  8.5054e-07

sherpa> foo1=get_flux2d
```

                                                      Example 1

```
sherpa> print(foo1)
dataset         =  1
range           =  NULL
comp            =  NULL
value           =  1.03989e-06
units           =  photons/cm**2/s
```

Bugs

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

See Also

chandra
        guide
sherpa
        bye, calc_kcorr, dataspace, dcounts, dollarsign, echo, eflux, eqwidth, erase, flux, get,
        get_dcounts_sum, get_dir, get_eflux, get_eqwidth, get_filename, get_flux_str, get_lfactorial,
        get_mcounts_sum, get_pflux, get_source_components, get_verbose, groupbycounts, guess, is,
        journal, list, list_par, mcounts, numbersign, paramest, plot_eprof, plot_rprof, prompt, reset, run, set,
        set_analysis, set_axes, set_coord, set_dataspace, set_dir, set_verbose, setplot, sherpa−module,
        sherpa_plotfns, sherpa_utils, show, simspec, use, version

Bugs