rgsfluxer

February 1, 2016

Abstract

Combines and fluxes RGS spectra.

1 Instruments/Modes

Instrument	Mode
RGS	Spectroscopy

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

rgsfluxer combines any collection of spectrum files generated by **rgsspectrum** to produce a single fluxed spectrum. *The fluxed spectrum should not be used as the starting point for any serious analysis of the data because it neglects the redistribution of monochromatic response into the dispersion channels.* A spectral fitting package, such as XSPEC, should be used for quantitative analysis of the raw spectra. Nevertheless, **rgsfluxer** is an extremely effective tool for visualizing the data free from the peculiarities of the instrument. This is especially so because of its ability to combine the data from both cameras into a single spectrum, thus eliminating many of the gaps between functioning detectors.

Each raw spectrum is fluxed using the effective area calculated from its corresponding response matrix. Since version 1.9, **rgsfluxer** also accepts as input parameter an ARF file created by **rgsrmfgen** and a resdistributiuon matrix created by the same task. The user has to make sure that the **rgsfluxer** is not being used with and ARF file **AND** a resditribution matrix which contains the effective area.

Ideally, each spectrum should be paired with a response matrix computed specifically for it, but that is in no way required. It is only required that each spectrum be paired with a response matrix of the same reflection order. Using a response matrix computed with fewer than 500 incident energy bins is not recommended, and it is left entirely to the user to assess the advisability of combining dissimilar spectra.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints

pha	yes	list of files		

A list of one or more spectrum files, all produced by **rgsspectrum**. The task does not enforce any constraint on the combining of dissimilar spectra.

rmf	no	list of strings	@	

A list of response matrix files, all produced by **rgsrmfgen**. Each response matrix is paired with the corresponding spectrum and must be compatible with it. The only practical requirement for compatibility is that the spectrum and paired response matrix share the same reflection order. Because of this, it is possible, if not always advisable, to combine several spectra using only one response matrix. As a shorthand for this, the special filename "." may be used to indicate that the previous response matrix should be repeated. Any number of successive repeats are allowed, but the number of response matrix files must equal the number of spectrum files. A further shorthand notation, "@", may be used to indicate that the name of the response matrix file can be inferred from the name of the corresponding spectrum file by replacing the "SRSPEC" label with "RSPMAT" (assuming the PPS-style file naming convention supported by **rgsproc**).

bkg	no	list of files		

A list of one or more background spectrum files, all produced by **rgsspectrum**. The task does not enforce any constraint on the combining of dissimilar spectra. The number of background spectra and pha spectra must equal and they must be in the same order in the parameter lists.

witharf	no	boolean	no	

Switch parameter to enable the possibility os running **rgsfluxer** using response matrix and effective area as two different files. This is only possible if **rgsrmfgen** task was executed with the parameter `witharffile` enabled.

arf	no	list of files		

A list of one or more effective area files, all produced by **rgsrmfgen** using parameter `witharffile`. If using this parameter, the corresponding rmf file had to be generated by **rgsrmfgen** using the parameter `witharffile`. The number of effective area files and rmf files must be in the same order in the parameter lists.

format	no	string	dal	dal—ascii—qdp

The desired format of the output fluxed spectrum: either a FITS-encoded SAS dataset (dal), a basic ASCII-encoded text file, or a text file formatted as input to qdp.

file	no	file		

The file name for the output fluxed spectrum. The default name varies according to the `format`: flux.ds (dal), flux.txt, or flux.qdp.

mode	no	string	wavelength	wavelength—energy

The computational mode: wavelength units (Å) or energy units (keV). This sets the units for the fluxing bins as specified by the parameters `min` and `max`, below, and as represented in the output file.

tryspectrumgrid	no	boolean	yes	yes—no

Boolean parameter to use the input spectra grid if they are all consistent. This parameter overwrites `min`, `max` and `bins` parameters

min	no	real	4.0	non-negative

The lower edge of the first output bin in units determined by the `mode`. The default is not appropriate if energy units have been selected.

max	no	real	40.0	non-negative

The upper edge of the last output bin in units determined by the `mode`. The default is not appropriate if energy units have been selected.

bins	no	integer	3600	positive

The number of output bins.

quality	no	list of integers	0	0 ≤ quality ≤ 5

This parameter controls how the quality flags from the input spectrum files are propagated into the output fluxed spectrum. If the quality flag of an input spectral channel is not represented in this list, the associated count rate is simply discarded. Where multiple channels, either from the same or from different spectra, overlap the same output bin, the output quality is degraded to the worst input quality. This ranking is set by the order of the list itself: from highest quality to lowest. Thus the list "0 2 1" preserves all input channels and degrades the quality in the sequence, "good" to "dubious" to "bad". The default preserves only the "good" channels. When only one quality value is permitted, the quality column is omitted from the output file. Quality flags three, four and five are defined within XSPEC, but are not currently used by **rgsspectrum**.

withaltnan	no	boolean	no	

Undefined values (NaN) occur at spectral channels where there is no exposure, such as in the gaps between chips. To accommodate the many standard analysis packages that do not react well to NaN, assert this parameter to cause all instances of NaN in the output to be replaced with the finite alternative specified by the parameter `nan`, below.

nan	no	real	0	

If enabled, this value is used instead of the IEEE NaN (or "no", in the case of `format`=qdp) to indicate an undefined value in the output fluxed spectrum.

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

pairing *(error)*
The number of response matrix files does not match the number of spectrum files; there must be a one-to-one pairing.

pairingbkg *(error)*
The number of background spectra files does not match the number of spectrum files; there must be a one-to-one pairing.

badPhaName *(error)*
The name of a response matrix file can be inferred from the name of the corresponding spectrum file only if the final 14 characters of the spectrum file name begin with "SRSPEC".

illegalRmfName *(error)*
The first response matrix file name cannot be “.”; it must either be “@” or else be given explicitly.

wrongOrder *(error)*
The `RFLORDER` attribute (reflection order) of the response matrix must match the `RFLORDER` of the corresponding spectrum.

channelGaps *(error)*
The `EBOUNDS` table of the response matrix must specify the channels in increasing sequential order, starting with `CHANNEL`=1. Note that this is not required by the OGIP standard for a response matrix file, but is guaranteed by **rgsrmfgen**.

6 Input Files

- A spectrum file as produced by **rgsspectrum**.
- A response matrix file, as produced by **rgsrmfgen**.

7 Output Files

The output file is a table with three or four columns. The first three are bin center, flux, and error. The bin centers are given in units of Åor keV depending on the `mode`: wavelength or energy respectively. Similarly the flux and error are given in $(s\ cm^2 Å)^{-1}$ or $(s\ cm^2 keV)^{-1}$. The fourth column, the quality flag, is omitted if the `quality` parameter specifies uniform quality for all bins. The structure of the file depends upon the selected output `format`:

dal A FITS-compliant DAL dataset with the table placed in an extension named “FLUXED”. The four columns are:

`CHANNEL`	real32	bin center
`FLUX`	real32	observed flux
`ERROR`	real32	statistical uncertainty
`QUALITY`	int16	quality flag

ascii A plain-text file, beginning with two comment lines indicated by “!” as the first character. The first line lists the names of the columns and the second line describes their units. The table follows; its columns are space-delimited. The name of the first column is either “wavelength” or “energy”. The remaining columns are “flux”, “error”, and “quality”, in that order.

qdp A plain-text file, beginning with the qdp directive, “READ SERR 2”. The rest of the file is structured in the same way as `format`=ascii, except that NaN values are written as “no” by default.

8 Algorithm

```
FOR EACH pair of spectrum and response:

  FOR EACH spectrum channel:
    IF ( acceptable QUALITY )
      time = EXPOSURE * AREASCAL
      rate = COUNTS / time
      err2 = (HDUCLAS2 == NET) ? (STAT_ERR/time)^2 : rate/time

      Rebin:
        map spectrum channel boundaries onto the fluxed channel grid
        accumulate rate and err2 according to fluxed channel overlap
        degrade QUALITY in ranked order on all overlapping fluxed channels
    ELSE
      mark spectrum channel as discarded

  FOR EACH response bin (rows):
    FOR EACH response channel (columns):
      IF ( corresponding spectrum channel not discarded )
        area = MATRIX * (ENERG_HI - ENERG_LO)
        Rebin:
          map response channel boundaries onto the fluxed channel grid
          accumulate area according to fluxed channel overlap

FOR EACH fluxed channel:
  IF ( area != 0 )
    FLUX = rate / area
    ERROR = sqrt(err2) / area
  ELSE
    FLUX = ERROR = NaN
```

9 Comments

References