Chandra X-ray Center

AHELP for CIAO 3.4 **rmfimg** Context: tools

Jump to: Description Examples Parameters Bugs See Also

Synopsis

Create a simple image of an RMF or ARF file

Syntax

```
rmfimg infile outfile [arf] [arfout] [product] [verbose] [clobber]
[mode]
```

Description

A response matrix file (RMF) encapsulates the mapping between the physical properties of incoming photons (such as their energy) and their detected properties (such as detector pulse heights or PHA) for a given detector. For X–ray spectral analysis, the RMF encodes the probability $R(E,p)$ that a detected photon of energy E will be assigned to a given channel value (PHA or PI) of p. The standard Legacy format (see the GSFC HEASARC journal Legacy, issue no. 2) for the RMF uses a binary table in which each row contains $R(E,p)$ for a single value of E as a function of p. Non–zero sequences of elements of $R(E,p)$ are encoded using a set of variable length array columns. This format is compact but hard to manipulate and understand. The rmfimg program converts this format to a simple 2–dimensional image of $R(E,p)$ versus E and p, allowing one to visualize the response of the instrument.

An ancillary response file (ARF) contains the effective area $A(E)$ including the probability that a photon of energy E will be detected at all. The total `response' is $A(E)R(E,p)$. The Legacy format for ARF is a binary table of $A(E)$ versus E. To make it easy to visualize $A(E)$ together with $R(E,p)$, rmfimg optionally allows you to create a two–dimensional `ARF image' $A(E,p)$, which just replicates $A(E)$ for each value of p. (To simply see the 1–dimensional function, you can use prism directly on the ARF, or use dmlist to dump it to an ascii file and use ChIPS to view it). Optionally, the output RMF image will be multiplied by the ARF to give an image of the total response.

Example 1

```
(1) rmfimg rmf.fits rmf.img
```

Makes an image from a Legacy RMF file.

Example 2

```
(2) rmfimg rmf.fits rmf.img arf=arf.fits arfout=arf.img product=no
```

As above, but also makes an ARF image from arf.fits.

Example 3

```
(3) rmfimg rmf.fits rsp.img arf=arf.fits arfout=arf.img product=yes
```

Makes response image rsp.img containing A(E)*R(E,p) (since product=yes) from rmf.fits and arf.fits. Also makes ARF image.

Example 4

```
(4) rmfimg "rmf.fits[bin energy=8,pi=2]" bin_rmf.img
```

Makes image from rmf.fits, binning axes by the given factor Only the names "ENERGY", "PHA", and "PI" are recognized, and only a single value representing the binning factor may be given; full DM rebinning syntax of the form start:stop:step is NOT supported. This functionality allows you to make a reasonable size image from a large RMF file.

Parameters

name	type	ftype	def	min	max	reqd
infile	file	input				yes
outfile	file	output				yes
arf	file	input				
arfout	file	output				
product	boolean		no			
verbose	integer		0	0	5	
clobber	boolean		no			

Detailed Parameter Descriptions

Parameter=infile `(file required filetype=input)`

Input Legacy RMF FITS file, possibly with binning expression.

Input RMF file.

Parameter=outfile `(file required filetype=output)`

Output RMF or response image.

Parameter=arf `(file filetype=input)`

Input ARF file, optional.

Parameter=arfout `(file filetype=output)`

Output ARF image, if input ARF file is supplied.

Parameter=product `(boolean default=no)`

If product is yes, the main output is a response image containing the product of RMF and ARF.

Parameter=verbose `(integer default=0 min=0 max=5)`

Controls screen output information, value is 0 (no info) to 5 (most info).

Parameter=clobber `(boolean default=no)`

Clobber output if it exists? (yes/no)

Bugs

See the bugs page for this tool on the CIAO website for an up−to−date listing of known bugs.

See Also

calibration
    ardlib
tools
    acis_bkgrnd_lookup, acis_fef_lookup, acis_set_ardlib, acisspec, add_grating_orders,
    add_grating_spectra, asphist, dither_region, dmarfadd, dmfilth, dmregrid, fullgarf, mkacisrmf, mkarf,
    mkexpmap, mkgarf, mkgrmf, mkinstmap, mkpsf, mkrmf, mkwarf, psextract, psf_project_ray, specextract

---

Parameter=outfile (file required filetype=output)