Chandra X-ray Center

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

Synopsis

Generate an RMF for Chandra imaging data

Syntax

```
mkrmf  infile outfile axis1 axis2 [logfile] [weights] [thresh] [outfmt]
[clobber] [verbose] [axis3] [axis4] [axis5]
```

Description

The 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. This mapping is stored in the form of a 2–D matrix. Spectral fitting packages like Sherpa and XSPEC use this redistribution function (the matrix), plus an accompanying effective area function (the ARF), to predict the counts per channel produced by a given spectral model for the source.

mkrmf calculates the RMF on user–defined grids in energy (units of keV) and channel (units of PHA or PI) using detector response information tabulated in a FITS embedded function (FEF) file. The FEF file is a standard Chandra X–ray Science Center (CXC) calibration product and tabulates both the analytic form of the detector response function as well as the variation of the function's parameters with input photon energy and position on the detector. For the ACIS detector, the response function is also a function of focal plane temperature and so multiple FEFs are provided in the Chandra calibration database (CALDB) for each operating temperature. Since the response function also depends on the charge transfer inefficiency (CTI), there are also individual FEF files appropriate for CTI and non–CTI corrected data.

For point sources (the single–tile case discussed below) the acis_fef_lookup tool can be used to find the correct FEF file (with the corresponding spatial filter). For extended sources (i.e. weighted responses), the correct file can be found automatically from the CALDB, although users may explicitly specify a given FEF to use if they choose.

In general, the response of the detector is a function of position. This dependence is reflected in the FEFs which contain information for many distinct "tiles" of varying size where the detector response has been calibrated. mkrmf can create an RMF appropriate for a single position on the detector or, alternatively, an RMF which

represents the weighted sum over multiple FEF tiles. Such a "weighted" RMF (WRMF) may be more appropriate for analysis of extended sources where spectra may have been extracted from large regions which span many FEF detector tiles. The weighting used to construct the WRMF is specified using an additional file (the weights parameter) which can be generated using the mkwarf tool. Note that weighted responses can only be calculated when using a PI grid (ie axis2 must be binned on PI and not PHA).

Examples of the various options for creating RMFs can be found in the CIAO science threads:

http://cxc.harvard.edu/ciao/threads/ispec.html

Output format

The output RMF is a standard FITS file and contains two extensions. The MATRIX extension consists of the actual energy to detector channel mapping and is an MxN matrix where M and N correspond to the sizes of the grids selected by the user. The EBOUNDS extension is a set of vectors giving the upper and lower energy bounds for each detector channel and is determined from the gain information in the FEF. The EBOUNDS information is used by packages such as Chandra and XSPEC for plotting purposes but is not used directly when fitting. mkrmf outputs two formats for the RMF: "legacy" and "cxc". The legacy format retains the HEASARC OGIP 92–002 standard for 2–D RMFs, is fully compatible with Sherpa and XSPEC, and is the default. The "cxc" format is a new CXC standard designed with future higher dimensional RMFs in mind.

Example 1

```
mkrmf "fef.fits[ccd_id=7,chipx=(1:256),chipy=(1:32)]" rmf.fits
axis1="energy=0.3:10.0:0.005" axis2="pi=1:1024:1"
```

Creates the RMF "rmf.fits" using an input FEF file named "fef.fits". The filter syntax isolates the FEF tile corresponding to CCD 7 in the ACIS array and detector pixels 1–256 in CHIPX and 1–32 in CHIPY. This filter above will be ignored if a weights file is specified. The energy grid for the RMF is given in keV and in this example ranges from 0.3 to 10.0 keV in 5 eV increments. The channel grid in this case uses PI bins.

Example 2

```
mkrmf "fef.fits[regnum=1]" rmf.fits
axis1="energy=0.3:10.0:0.005" axis2="pi=1:1024:1"
```

Runs the same calculation as above but looks at FEF region number 1. Each individual calibration tile in the FEF has a unique number.

Example 3

```
mkrmf "fef.fits[ccd_id=7,chipx=(1:256),chipy=(1:32)]" rmf.fits
axis1="energy=0.3:10:0.05" axis2="pi=1:1024:1"
```

Creates an RMF as in Example 1 but with an energy grid which is binned linearly from 0.3 keV to 10 keV but with a bin size of 0.05 keV.

Example 4

```
mkrmf "fef.fits[ccd_id=7,chipx=(1:256),chipy=(1:32)]" rmf.fits
axis1="energy=0.3:10:0.05L" axis2="pha=1:4096:2"
```

Generates an RMF with an energy grid from 0.3 to 10.0 keV with a bin step of 0.05 keV in logarithmic space. For the PHA axis, the matrix values are calculated at full resolution (all 4096 channels) and then binned by a factor of 2 to produce 2048 output channels. The factor of 2 binning is also replicated automatically in the EBOUNDS array of the output RMF.

Example 5

```
mkrmf "fef.fits[ccd_id=7,chipx=(1:256),chipy=(1:32)]" rmf.fits
axis1="energy=grid(grid.txt)" axis2="pha=1:4096:2"
```

Indicates that mkrmf should use the first two columns in the ASCII file "grid.txt" to define the lower and upper bounds of the energy grid, respectively.

Example 6

```
mkrmf "fef.fits[ccd_id=7,chipx=(1:256),chipy=(1:32)]" rmf.fits
axis1="energy=grid(rmf_hires.fits[MATRIX][cols ENERG_LO,ENERG_HI])"
axis2="pha=1:4096:2"
```

The energy grid bounds are defined by "ENERG_LO" and "ENERG_HI" columns from the "MATRIX" extension of FITS file, "rmf_hires.fits". The first two columns from the first FITS extension will be extracted as default if no columns are specified here.

Example 7

```
mkrmf fef.fits rmf_wgt.fits weights=weights.fits
axis1="energy=0:1" axis2="pi=1:1024:1"
```

mkrmf will calculate a weighted RMF using the specified FEF file "fef.fits". The weights will be read from the file "weights.fits". The energy grid parameter will be ignored (hence the use of "0:1") as the tool will automatically get the grid bounds from "ENERG_LO" and "ENERG_HI" columns tabulated in the "EBOUNDS" extension of "weights.fits" file. mkrmf will evaluate the response in each FEF tile spanned by the indicated region and produce a weighted average using the indicated weights. Note that weighted RMFs can only be calculated for PI matrices.

Example 8

```
mkrmf CALDB rmf_wgt.fits weights=weights.fits
axis1="energy=0:1" axis2="pi=1:1024:1"
```

Another example of generating a WRMF by querying the CALDB for the correct FEF file (see "ahelp caldb" for more information on the syntax of the CALDB queries). mkrmf will find the appropriate FEF file and a full path to it by looking up values of DATE–OBS, CTI_CORR, and other header keywords found in the weight file (here

Example 4

3

weights.fits). Note that this syntax only works when a weights file is used.

Parameters

name	type	ftype	def	min	max	units	reqd
infile	file	input					yes
outfile	file	output					yes
axis1	string	input					yes
axis2	string	input					yes
logfile	file	output	STDOUT				
weights	file						
thresh	float		1e−5	0.0			
outfmt	string		legacy				
clobber	boolean		no				
verbose	integer		0	0	5		
axis3	string		none				
axis4	string		none				
axis5	string		none				

Detailed Parameter Descriptions

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

name of an input FEF file.

Includes full path and filter information (if a filter was used) or a CALDB query (e.g. infile="CALDB", see "ahelp caldb") if a weights file is being supplied to mkrmf. There are two valid file formats within CXC standards. One contains an analytic function only, for example, a single−chip, single−region embedding response function table file (or ECD) for MATRIX and gain interpolated in a linear expression of "scale" and "offset" header keywords for EBOUNDS. The second format contains multi−chips, multi−regions, and tabulated gain information (say, energy and pha) plus the analytic response function. An input file of this kind shall have an appropriate filter spec following a given file name as shown in the examples above. Prior to CIAO 2.3, the infile parameter had to be set to "CALDB(QUERY=CHANTYPE.EQ.PI)" when using a weight file. All that is needed now is "CALDB", since mkrmf can generate the PI RMF from a PHA FEF file ("PI on the fly"). Using the old value will result in an error from mkrmf saying:

```
   ERROR: couldn't find CALDB(QUERY=CHANTYPE.EQ.PI)
```

If infile=CALDB, the value of the CTI_CORR keyword in the header of the weights file is used to determine whether or not to use the FEF appropriate for CTI−corrected data.

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

name of an output FITS RMF file.

Displays in "legacy" or "cxc" format (see "outfmt" below) and contains a null primary binary table followed by two extensions: response MATRIX and EBOUNDS.

The MATRIX array in each gridded (say, energy) row are compressed so any element value below the cutoff value (specified by "thresh" parameter, see below) will be removed from the array and then normalized to one for that row over the effective channels (PHA/PI). The EBOUNDS arrays are interpolated from the embedded gain table over the full resolution of channels and output in the user−specified bin step.

Parameter=axis1 `(string required filetype=input)`

axis and binning attributes

Specify attributes containing name of axis and grid binning parameters. The description below is also valid for the other axis parameters ("axis2" up to "axis5").

The various ways to specify the grids are summarised in the following table, after which they are described in more detail.

Syntax	Bin Type	Example
<axis−name>=<min>:<max>:<bin−num>	linear bin in bin number	pi=1:1024:#1024
<axis−name>=<min>:<max>:<bin−step>	linear bin in bin step	energy=0.1:10.0:0.05
<axis−name>=<min>:<max>:<bin−step>L	logarithimic bin in bin step	energy=0.1:10.0:0.05L
<axis−name>=<min>:<max>:<bin−num>L	logarithimic bin in bin number	energy=0.1:10.0:#1500L
<axis−name>=grid(<file>)	grids tabulated in ASCII file	energy=grid(eng.txt)
<axis−name>=grid(<file>)	grids tabulated in FITS file	energy=grid(eng.fits)

Command line

The grid is specified by giving the lower bound, upper bound, and binning type, separated by the ":" character. The parameter of binning type can be specified as linear or logarithmic binning for either binning step or total binning number. Note that mkrmf will not accept a logarithmic spacing for the PHA/PI axis: the bin step along such an axis should be an integer (1, 2, or 3,...), otherwise it will exit with an error.

ASCII table file

The grid is tabulated in a text file and input using "axis−name=grid(<filename>)". The file should contain two columns, the first column listing the lower bound and the second column the upper bound for each bin.

FITS binary file

Here the grid is tabulated in a FITS file. The same syntax is used as the ASCII file, namely "acis−name=grid(filename)", although in this case the file name can include a DM filter to specify the block and columns to read (see 'ahelp dmsyntax' and 'ahelp dmcols' for more information). If no column specifier is given then the first two columns in the block are used; it is therefore safest to include the column names as in the following example:

```
axis1="energy=grid(spectrum.rmf[MATRIX][cols ENERG_LO,ENERG_HI])"
```

which uses the energy grid specified by the ENERG_LO and ENERG_HI columns of the MATRIX block in spectrum.rmf.

Parameter=axis2 `(string required filetype=input)`

axis and binning attributes

See the description of the axis1 parameter.

Parameter=logfile `(file filetype=output default=STDOUT)`

name of log file.

Records all STDERR messages. It also logs any STDOUT for verbose levels of 1 and higher.

Parameter=weights `(file)`

name of an input weights file in FITS, used for weighted RMF generation.

The weights file, created by mkwarf, consists of two FITS extensions: WSPECRESP and EBOUNDS. The first extension contains an array of weighted spectral response as function of energy for each response region, which is identified by columns of CHIP_ID, CHIP_X, CHIP_Y, and REGNUM. The second one stores energy bounds of the spectral response in the same way the EBOUNDS block does in RMF. If "weights=none" or "weights= ", the tool will generate a single–region RMF output based on the user's specifications of grids, otherwise, a weighted PI channel type RMF over multi–response regions will be created. For the multi–FEF case, the "energy" axis grid can be set to "energy=0:1" since the values are taken from the weights file.

Parameter=thresh `(float default=1e-5 min=0.0 units=)`

Minimum value included in the output matrix.

All MATRIX elements below this threshold value are excluded from the output file. This is used to reduce the size of the output file. It is suggested that the parameter value is left at its default value.

Parameter=outfmt `(string default=legacy)`

the format of RMF output

Accepts "legacy" and "cxc" formats. The former is to support the standard of the HEASARC OGIP 92–002 legacy: 2–D, valid for (energy, pha). The latter is a new alternative of CXC standard which supports 2–D or higher dimensionality matrix calculation with relative flexibility of axis "names".

Parameter=clobber `(boolean default=no)`

clobber option.

Overwrites existing files of the same output file name if "yes"

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

verbosity level

Displays verbosity level from 0 to 5, all messages are dumped in "logfile". Level 0: no display except error message. Level 1–3: all message of verbose=0 plus debugging message for moderate detail. And Level 4–5: all message of verbose=0 – 3 plus more detailed logs to be used for developer's debugging. User is suggested to set verbose up to 3 if needed.

Parameter=axis3 `(string default=none)`

axis and binning attributes

Has the same description as "axis1". It is suggested that the value is left as "none", because mkrmf is currently limited to creating 2–D RMFs.

Parameter=axis4 `(string default=none)`

axis and binning attributes

See the description of the axis3 parameter.

Parameter=axis5 `(string default=none)`

axis and binning attributes

See the description of the axis3 parameter.

DYMAMIC PHA REBINNING

Beginning with CIAO 2.3, mkrmf now has the ability to calculate PI matrices directly from PHA FEF files. Previously, separate PHA and PI FEF files were maintained in the CALDB. Now mkrmf can utilize the gain information contained in the FEF file to rescale the matrix to PI channels internally when the user selects PI channels for the second axis of the matrix. By default, mkrmf will use the "PI–on–the–fly" mapping to create RMFs. Users may still explicitly specify existing PI FEF files if they so desire.

As discussed above in the infile parameter description, the use of the "PI–on–the–fly" technique means that the use of "CALDB(QUERY=CHANTYPE.EQ.PI)" when creating a weighted response is no longer valid; the correct value to use is now "CALDB".

FEATURES AND LIMITATIONS

- Users may specify the energy grid of the RMF via the command line syntax discussed below, or using an external file. The file containing the desired custom energy grid can be either a FITS binary table or a simple ASCII file.
- All string inputs are case insensitive with this tool.
- Only 2–D response matrices are currently supported.

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

dm
> dmbinning

tools

> acis_bkgrnd_lookup, acis_fef_lookup, acis_set_ardlib, acisspec, add_grating_orders, add_grating_spectra, asphist, dither_region, dmarfadd, dmcopy, dmextract, dmfilth, dmgroup, dmimghist, dmregrid, fullgarf, get_sky_limits, mkacisrmf, mkarf, mkexpmap, mkgarf, mkgrmf, mkinstmap, mkpsf, mkwarf, psextract, psf_project_ray, rmfimg, specextract, tgextract

---