XRISM Quick-Start Guide
Version 3.1

XRISM Science Data Center

Last updated: August 29, 2025

Contents

1 Introduction

This guide is designed to get you started quickly when you receive your *XRISM* data and is deliberately terse, with minimal details and explanations. You should read the help files for each tool for more information (type `fhelp` followed by the tool name, or type the tool name followed by "xrism help" into Google search). If you get stuck, e-mail the SDC help desk (`XRISM-SDC-help@lists.nasa.gov`) or one of the other *XRISM* help desks.

By following this *Quick-Start Guide* (hereafter, QSG), you will be able to quickly extract publishing-quality images, lightcurves, spectra, and response files. Note that earlier versions of the QSG were based on earlier versions of the software, calibration database (hereafter, CalDB), and data reprocessing.

It is recommended to read the *XRISM Data Analysis Guide*, in which more information can be found on some of the topics. However, the current version of it is 1.0, older than this QSG, and does not include some new tools that have been released since then. Also, the *XRISM Data Analysis Guide* does not include a method for removing anomalous pixels in Xtend data. In addition, this QSG describes a more refined mitigation strategy for the adverse effects of anomalous Resolve Ls events and branching ratios on the net Resolve effective area.

Note that in the commands in the following sections, even though many default input parameter values are used, sometimes the parameters are given explicitly, because this shows key parameters that can be changed to give desired variations in the results.

2 What You Will Need

Here, we describe the set-up & files that you will need. Documentation on how to set up the items below can be found elsewhere.

1. Installation of the HEASoft software, version 6.36 or later. However, this QSG straddles software releases. If you only have HEASoft 6.35.2, you can still use this QSG to make publishing-quality products, but some of the "convenience" functionality will not be available. Steps that require HEASoft 6.36 or later will be indicated in the text by "`H6.36`". The tool needed for §8 will not be available until HEASoft 6.37 or later, but the steps in §8 are not mandatory.
2. Installation of version 11 or later of the XRISM Calibration Database (CalDB). The latest XRISM CalDB files be downloaded from the HEASARC XRISM CalDB web page.

Follow instructions on the software and CalDB web pages to correctly set up the software and CalDB. Initialize the *XRISM* software and CalDB. The environment variable $HEADAS should point to your *XRISM* software tree, and the environment variable $CALDB should point to the top of your CalDB.

Before beginning data analysis, familiarize yourself with the web page *Things to Watch Out For* (which covers XRISM data anomalies, analysis tools, and calibration). Hereafter, this will simply be referred to as the *TTWOF*. Also check the *QSG* page for errata for this guide, and for later versions of this guide.

Resolve Energy-Scale Reports Check the Resolve energy-scale report for your observation. The hyperlink takes you to directories named by the first digit of the OBSID of your observation. Go to the appropriate directory and find the report with your OBSID in the file name. The report shows details about the gain determination for the observation, as well as other useful metrics. There are instructions on what kind of anomalies to look for. If you find that something is not what would be expected, contact one of the help desks.

3 The Data

The sequence number of a *XRISM* observation data set is a key item in the naming convention of all the files associated with an observation. In this guide, for the sake of convenience, we will use a specific sequence number (000125000). Substitute the sequence number of your observation in the commands and file names.

The top of the directory tree is

`000125000/`

(further details on the rest of the directory structure can be found elsewhere).

Note that the pipeline scripts (`xapipeline`, `rslpipeline`, `xtdpipeline`) should be run in a directory that contains this top-level directory (*not* under this top-level directory). *It is recommended to reprocess the data with the latest software and CalDB, because there may have been updates to the software and/or CalDB since the data were processed by the official pipeline.* Instructions for reprocessing are given in §4.

There may be more than one cleaned event file for each instrument ("cl" will be in the file name, and the extension will be .evt). In this guide, we will only treat the principal file for each instrument. For Resolve, the following sub-strings in the file name indicate the Filter Wheel position:

Sub-string	Filter Wheel Position
p0px0000	Undefined
p0px1000	OPEN
p0px2000	Al/Polyimide
p0px3000	Neutral Density (ND)
p0px4000	Be
p0px5000	Fe 55 calibration source

The Xtend file names carry a lot of information, and details of the naming conventions can be found on the *TTWOF* and in the *XRISM Data Analysis Guide*. Here, we only note the meaning of the following principal sub-strings in the file names:

Sub-string	Meaning
p0300	The event list contains all four CCDs in full window mode
p0311	The event list contains CCD1 and CCD2 in 1/8 window mode
p0312	The event list contains CCD1 and CCD2 in full window + 0.1 sec burst mode
p0313	The event list contains CCD1 and CCD2 in 1/8 window + 0.1 sec burst mode
p0320	The event list contains CCD3 and CCD4 in full window mode

4 Reprocessing

To reprocess the data go to the directory above `000125000/` and run `xapipeline`. Although both instruments can be done at once, the following commands do the two instruments separately and place the results for each instrument into separate directories.

For Resolve data, using the Fe 55 filter-wheel source for gain calibration, with the results being placed in `000125000_rsl_reproc/`, execute

```
punlearn xapipeline

xapipeline indir=000125000 outdir=000125000_rsl_reproc steminputs=xa000125000
stemoutputs=DEFAULT entry_stage=1 exit_stage=3 instrument=resolve verify_input=no
create_ehkmkf=no calc_pointing=yes calc_optaxis=yes calc_gtilost=no calc_adrgti=no
calc_mxsgti=no rsl_gainfile=CALDB calmethod=FE55 linetocorrect=MnKa seed=650504
numevent=1000 minevent=200 extraspread=40 spangti=no
```

For Xtend data, execute

```
punlearn xapipeline
```

```
xapipeline indir=000125000 outdir=000125000_xtd_reproc steminputs=xa000125000 stemoutputs=DEFAULT
entry_stage=1 exit_stage=3 instrument=xtend verify_input=no create_ehkmkf=no calc_pointing=yes
calc_optaxis=yes seed=650504
```

This will place the results in `000125000_xtd_reproc/`. Note that there is now no task called by `xapipeline` or `xtdpipeline` that removes anomalous pixels (see §6).

Note that you can substantially reduce the size of the directories containing reprocessed data by deleting the *.hk files, which are simply duplicates of the corresponding files still present in the original data tree. All the remaining files can be gzipped.

5 Initial Set-Up and Checks

You can examine quick-look preview products before reprocessing under:

```
000125000/resolve/products/ (Resolve)
000125000/xtend/products/ (Xtend)
```

and the same products after reprocessing under

```
000125000_rsl_reproc/ (Resolve)
000125000_xtd_reproc/ (Xtend)
```

(file names will contain the strings "lc", ".pi", and "img" for lightcurves, spectra, and images respectively; there are no spectral response quick-look files). However, none of these products should be used for science or calibration analysis (instructions to make analysis-ready products will be given in the appropriate sections below).

Create a directory where you will do the data reduction and analysis. In this guide, this directory is called `analysis/`.

Copy the cleaned event files (i.e., files with names ending in cl.evt.gz) that you will be using, to this directory, from

```
000125000_rsl_reproc/ (Resolve)
000125000_xtd_reproc/ (Xtend)
```

From the Resolve event file, note the values of the following keywords from the extension 1 header (e.g., using `fkeyprint, ftlist`, etc.):

```
RA_NOM
DEC_NOM
PA_NOM
```

These will be needed later.

Also copy over the following files into the `analysis/` directory:

```
000125000_rsl_reproc/xa000125000.ehk.gz
000125000_rsl_reproc/xa000125000rsl_px1000_exp.gti.gz
000125000_xtd_reproc/xa000125000xtd_p031100010.bimg.gz
```

5.1 Run rslbratios on Resolve Event Files

PSP Limit Check If the *unscreened* count rate in a Resolve quadrant exceeds 50 cts s^{-1}, the PSP limit is exceeded and data from that quadrant will not be usable. In `000125000_rsl_reproc/`, run `rslbratios` on

the unfiltered ("uf") Resolve event file to check how close the count rates per quadrant are to the PSP limit (the information is written to the terminal screen):

```
rslbratios infile=xa000125000rsl_p0px1000_uf.evt.gz filetype=uf outroot=rsl000125000_ufbr
```

If the PSP limit is exceeded in a quadrant, that quadrant can be excluded from contributing to lightcurves and spectra (§11) by appropriate specifications of pixel IDs or selection regions. (Note that the run of `rslbratios` will create a large number of output files (beginning with the string `rsl000125000_ufbr`), but these outputs for unscreened data will not be needed in this QSG.)

Cleaned Event File Run

Run `rslbratios` on the cleaned Resolve event file:

```
rslbratios infile=xa000125000rsl_p0px1000_cl.evt.gz filetype=cl outroot=rsl000125000_clbr lcbin=128.0
```

This will produce many output files, some of which will be used in later steps in this QSG (consult the help file for `rslbratios` for detailed descriptions of the outputs). Here, you can examine the time variability and spatial distribution of grade fractions:

`rsl000125000_clbr_2keVto12keV_hist.fits:` Observed vs. predicted branching ratios in the energy range specified by the "eband" parameter
`rsl000125000_clbr_full_lc.fits:` Full array light curves by grade
`rsl000125000_clbr_pix_lc.fits:` Per-pixel light curves by grade
`rsl000125000_clbr_gradeFrac_lc.fits:` Grade fractions versus time
`rsl000125000_clbr_2keVto12keV_pixMap.img:` 20 images of counts by grade, and grade fractions for two energy bands

Consult the `rslbratios` help for more details. Temporal and/or spatial variation of the grade fractions require a more complicated treatment for generating spectral response files (see §14.4).

6 Remove Anomalous Pixels from Xtend Data

The cleaned Xtend data from the pipeline may still be contaminated by pixels that have anomalous count rates, due to either flickering pixels, cosmic-ray echo events (CRE), or other causes whose origin may not yet have been established. (Note that the use of the tool `searchflickpix` has been discontinued because it often removes a significant fraction of source events.) Currently, the only method available to remove anomalous pixels is to use the tool `xtdpixclip`, which offers a manual approach to identifying anomalous pixels in an arbitrary number of user-selected regions on the detector. Histograms produced by `xtdpixclip` are then examined to determine counts-per-pixel cut-off thresholds for the different regions. The `xtdpixclip` task is run again in "apply" mode, using the selected thresholds to produce a cleaned event file and an anomalous pixel list. Details on using `xtdpixclip` can be found in the help and in the manual `xtdpixclip_guide.pdf` (available from the SDC web pages).

Using the example sequence in this QSG, with `outroot=xa000125000xtd_p031100010_cl`, the output files from two runs of `xtdpixclip` (i.e., with `pmode=histo`, followed by `pmode=apply`) will be:

`xa000125000xtd_p031100010_cl_xpc_rmvpix.fpix:` (Anomalous pixels list; this file is used later in ARF generation)
`xa000125000xtd_p031100010_cl_xpc_clnevt.fits:` (Cleaned event file, made if `mkclean=yes`, with events in anomalous pixels removed)
`xa000125000xtd_p031100010_cl_xpc_hist.fits:` (Histograms of counts per pixel for each region)
`xa000125000xtd_p031100010_cl_xpc_regNN_hist.gif:` (Plots of the histograms in the above file, where NN= region ID = 01, 02,..etc.)

On the first-pass run of `xtdpixclip`, you may be fortunate to find that there are no anomalous pixels in your regions of interest, in which case you can skip the second pass run of `xtdpixclip` and continue to use the file `xa000125000xtd_p031100010_cl.evt` in downstream analysis. However, the commands in the rest of this QSG will assume that the Xtend event file output from `xtdpixclip` will be used. In the commands, substitute the file name `xa000125000xtd_p031100010_cl_xpc_clnevt.fits` with `xa000125000xtd_p031100010_cl.evt` if it was found that there were no anomalous pixels to be removed.

7 Resolve Proximity and Rise-Time Screening

Proximity screening removes frame events on the basis of the unlikeihood of source events occurring too close together in time given a pre-defined mean time interval, say Δt, and the count rate of the source, say C_0. However, a simple flagging criterion such as this will inevitably produce some false positives, thus removing valuable source counts. The frame count rate is ~ 0.02 cts s^{-1}, and the Poisson probability of getting one pair of (real) source counts in an interval $\pm\Delta t$ is $\exp(-C_0 2\Delta t)$. The value of Δt for the proximity flag, STATUS[4], is 0.72 ms, so the argument of the exponential is $\ll 1$. The false positive rate is then equal to the frame rate when $C_0(2C_0\Delta t) = 0.02$, or $C_0 = 3.73$ cts s^{-1}. This corresponds to a false positive percentage of $\sim 0.53\%$. You have to consider for your particular data and science goals, the trade-off between losing a percentage of (energy-independent) counts ($\sim 200 C_0\Delta t$), versus the contamination percentage ($2/C_0$). To facilitate this, you can examine the file `rsl000125000_clbr_prox_Hp_spectra.fits` that was created by `rslbratios` in §5.1, which gives a preview of the effect of the proximity screening on your particular data. The second extension of the file gives the Hp difference spectrum with and without the proximity screening, and shows the actual number of counts lost per PI channel, versus PI channel.

You can also preview the effect of rise-time screening from the `rslbratios` output. Extension 1 in `rsl000125000_clbr_rise_Hp_spectra.fits` shows the Hp counts versus PI channel after the rise-time screening, and extension 2 shows the Hp counts that are removed by the screening, versus PI channel. In the difference spectrum, watch out for significant counts removed in the energy band over which you will be doing spectral fitting. Also note that we know from data with the gate valve closed that it is clear that rise-time screening can often still leave a significant non-source signal below ~ 2 keV. In some cases, it may cause the spectrum to steepen at low energies, so could be misinterpreted as a "soft excess", or miscalibration of the effective area.

To perform proximity *and* rise-time screening on the cleaned event file, execute the following command:

```
ftcopy infile="xa000125000rsl_p0px1000_cl.evt[EVENTS]
[(PI>=600) && (((((RISE_TIME+0.00075*DERIV_MAX)>46)&&((RISE_TIME+0.00075*DERIV_MAX)<58))
&&ITYPE<4)||(ITYPE==4))&&STATUS[4]==b0]" outfile=xa000125000rsl_p0px1000_cl2.evt
copyall=yes clobber=yes history=yes
```

If you decide to skip proximity screening, omit "`&&STATUS[4]==b0`" in the above expression.

8 Run xaspecratios to Check for Spectral Distortion

Spectra extracted from regions that are smaller than the PSF, such as selected Resolve pixels or pixel groups, often show differences that cannot be attributed to known causes. You can run the tool `xaspecratios` (available only in a future software release, likely HEASoft 6.37) on Resolve or Xtend data to see if your data suffers from this. The tool has a lot of different options, but the following command makes spectral ratios for Resolve pixels groups corresponding to the central 4 pixels, the inner ring, and the outer ring:

```
punlearn xaspecratios
xaspecratios evtinfile=xa000125000rsl_p0px1000_cl2.evt.gz bgdspec=NONE ebinfac=8 refpixgrp=cntr4
pixgroups=rings outroot=xa000125000rsl_p0px1000_cl2bin8 resolist=0
```

The output file will be `xa000125000rsl_p0px1000_cl2bin8_specratios.fits`. This example does not include accounting for background. Consult the help for `xaspecratios` to learn about all the different options, including application to Xtend, and how to plot and interpret the output results. If the results show

that the spectral ratios versus energy are not flat, it does not necessarily mean that there is a problem because some or all of the spectral variation may be accounted for later when you apply the spectral responses during spectral-fitting analysis. However, if the spectral fitting results for spectra made from different spatial groups are inconsistent with each other, then it means that there are residual unexplained spectral differences. Until there is a better understanding of the origin of the problem, two mitigation strategies are:

- Try fitting spectra individually from the 4 central pixels to see if they are more consistent with each other, and use those.
- Restrict spectral fitting to an energy band that plots made by `xaspecratios` show flat spectral ratios.

9 Extract Broadband Images and Establish Source Center Coordinates

Check the value of the EXPOSURE keyword in the cleaned event files. If the exposure time for either instrument is unexpectedly low (compared to that planned), contact one of the help desks to check if something went wrong with the observation and/or reprocessing.

You should now have, by the example in this QSG, the following cleaned event files:

```
xa000125000rsl_p0px1000_cl2.evt (Resolve) and
xa000125000xtd_p031100010_cl.evt or xa000125000xtd_p031100010_xpc_clnevt.fits (Xtend)
```

The Resolve data analysis in this QSG primarily concerns the "hi-res" events, or equivalently Hp grade events. Only some of the brightest sources will be amenable for analysis using "mid-res primary" events (grade Mp), and some details on how to do this are described in Appendix A.

For the Xtend commands in the rest of this QSG, substitute xa000125000xtd_p031100010_xpc_clnevt.fits with xa000125000xtd_p031100010_cl.fits if you did not run `xtdpixclip`. In all of the XSELECT commands in the remainder of this guide, a generic prompt (`xselect>`) will be shown, instead of the often lengthy, and rather complex prompts that you will actually see.

Start XSELECT and make a 2–10 keV Resolve image in DET coordinates:

```
xselect> read eve xa000125000rsl_p0px1000_cl2.evt .
xselect> set image DET
xselect> filter pha_cutoff 4000 20000
xselect> extr image
xselect> save image xa000125000rsl_p0px1000_detimg.fits
```

For a standard, "on-axis" point source observation, the position of the source on the Resolve detector should be close to (3.5, 3.5) in DET coordinates, which is the center of the 6x6 Resolve pixel array. However, it is important to note that, due to asymmetry even in the core of the PSF, the brightest pixel is *not* the center or centroid of the source. Here, we only need to estimate the center of the source approximately, because it will only be used as a "sanity check" against the source RA & DEC coordinates. The ARF generator will use the actual RA & DEC coordinates of the source, and this "sanity check" is insurance against something going wrong with the treatment of coordinates throughout the pipeline processing. Using `ds9` (or other tool), estimate the source center. For example, in `ds9` you can use the "Horizontal Graph" and "Vertical Graph" options under the "View" menu. Depending on your purpose (desired accuracy), you could, alternatively, mathematically calculate the centroid, or the mean DETX & DETY, using the pixel counts values (for the entire array or for the inner 16 pixels), or you could even estimate it by eye. We will refer to the DETX and DETY coordinates of the Resolve source center as

```
RDETX0, RDETY0
```

respectively. For most standard point-source observations, it may be sufficient to simply assume `RDETX0=3.5` and `RDETY0=3.5`. If the source is extended, and there is no obvious source center, again, simply set `RDETX0=3.5` and `RDETY0=3.5`.

Copy over a Resolve region file that will also be used to make a Resolve ARF later:

```
cp $HEADAS/refdata/region_RSL_det.reg .
```

Now make a 0.5–10 keV Xtend image in DET coordinates:

```
xselect> clear all
xselect> read eve xa000125000xtd_p031100010_xpc_clnevt.fits.gz .
xselect> set image DET
xselect> filter region exclude_calsources.reg
xselect> filter pha_cutoff 83 1667
xselect> extr image
xselect> save image xa000125000xtd_p031100010_detimg.fits
```

In the above, the optional `filter` command with the region file `exclude_calsource.reg` excludes the Xtend calibration source regions from the image. You can easily make the region file because it is simply a plain text file. The contents of the region file are:

```
physical
-circle(920.0,1530.0,92.0)
-circle(919.0,271.0,91.0)
```

Next, estimate the centroid of the source in DET coordinates. We will refer to the DETX and DETY coordinates of the centroid as

```
XDETX0, XDETY0
```

respectively, and use these values later to check against the source coordinates, and to make a source extraction region file. If the source is extended and has no obvious center, these coordinates should correspond to the center of the region that you will be extracting a source spectrum from.

Using `ds9`, make region files for extracting Xtend source and background light curves and spectra (from the same chip). In Full Window (WINDOW1) mode, the source region you want will likely be a circle (for a point source a radius of 2.5 arcmin, or 85 pixels, is recommended). In 1/8 Window (WINDOW2) mode, the source and background regions will be rectangles. For a point source, 5 arcmin is recommended for the length of the source rectangle. The width should be as wide as possible, *but no part of the region should extend beyond the chip boundary, because this will result in an incorrect ARF*. The source and background regions will be referred to as

```
region_000125000xtd_src.reg
region_000125000xtd_bgd.reg
```

for source and background respectively. The Xtend regions `can` include chip gaps and bad columns because they are correctly treated by the ARF generator.

Xtend Pile-up The core of the PSF may be piled-up in bright sources. A mitigation strategy is to exclude a circular region of the PSF core. The size of the region will vary case-by-case. You have to examine the radial distribution of the counts, but the details are beyond the scope of this QSG.

Xtend Baffle Shadowing An outer corner of one of the Xtend CCDs in full window mode is in the shadow of a baffle in the optical path. For very large extended source analysis, data in the shadow should be exlcuded. The following large circle region excludes the shadowed corner:

```
physical
+circle(730.1,727.5,1083)
```

This region must be AND'ed with the region for data extraction (e.g., a square region covering the entire detector with boundaries that lie inside the detector).

10 Resolve Region Files, Pixel Selections, & Pixel Gaps

Resolve regions for making spectra and spectral responses must always be in DET coordinates. To make a Resolve region file for an arbitrary combination of pixels, paste the following full-array per-pixel region list into a region file and delete the rows that you don't need, and then delete the second column (which contains the pixel ID):

```
physical
+box(4.0,3.0,1.0,1.00000000) 0
+box(6.0,3.0,1.0,1.00000000) 1
+box(5.0,3.0,1.0,1.00000000) 2
+box(6.0,2.0,1.0,1.00000000) 3
+box(5.0,2.0,1.0,1.00000000) 4
+box(6.0,1.0,1.0,1.00000000) 5
+box(5.0,1.0,1.0,1.00000000) 6
+box(4.0,2.0,1.0,1.00000000) 7
+box(4.0,1.0,1.0,1.00000000) 8
+box(1.0,3.0,1.0,1.00000000) 9
+box(2.0,3.0,1.0,1.00000000) 10
+box(1.0,2.0,1.0,1.00000000) 11
+box(2.0,2.0,1.0,1.00000000) 13
+box(2.0,1.0,1.0,1.00000000) 14
+box(3.0,2.0,1.0,1.00000000) 15
+box(3.0,1.0,1.0,1.00000000) 16
+box(3.0,3.0,1.0,1.00000000) 17
+box(3.0,4.0,1.0,1.00000000) 18
+box(1.0,4.0,1.0,1.00000000) 19
+box(2.0,4.0,1.0,1.00000000) 20
+box(1.0,5.0,1.0,1.00000000) 21
+box(2.0,5.0,1.0,1.00000000) 22
+box(1.0,6.0,1.0,1.00000000) 23
+box(2.0,6.0,1.0,1.00000000) 24
+box(3.0,5.0,1.0,1.00000000) 25
+box(3.0,6.0,1.0,1.00000000) 26
+box(6.0,4.0,1.0,1.00000000) 27
+box(5.0,4.0,1.0,1.00000000) 28
+box(6.0,5.0,1.0,1.00000000) 29
+box(6.0,6.0,1.0,1.00000000) 30
+box(5.0,5.0,1.0,1.00000000) 31
+box(5.0,6.0,1.0,1.00000000) 32
+box(4.0,5.0,1.0,1.00000000) 33
+box(4.0,6.0,1.0,1.00000000) 34
+box(4.0,4.0,1.0,1.00000000) 35
```

Go to next page ...

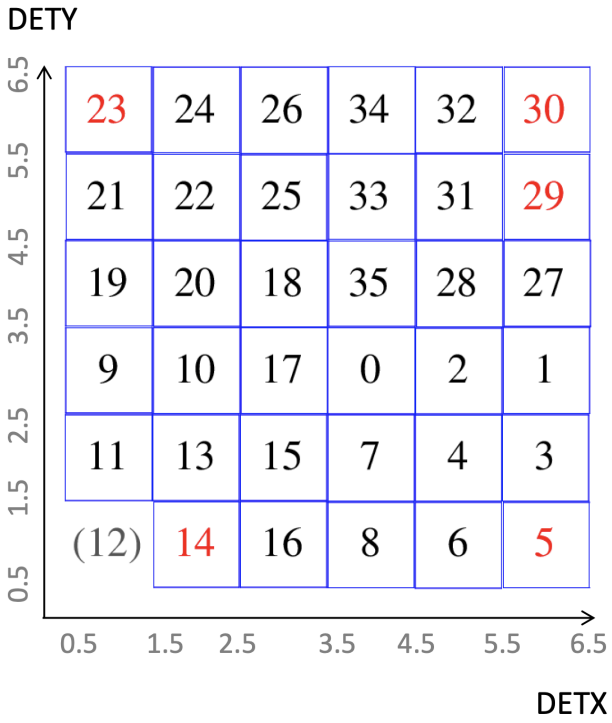

Figure 1: Resolve pixel ID map, also showing the correspondence with DET coordinates.

Use the pixel map in Figure 1 to construct desired regions for sub-array data analysis.

Resolve Pixel Gaps The Resolve pixel DET coordinate system is a sanitized representation of the actual physical pixels. In this system, the fractional area of the pixel gaps is 4.28% of the total area (including gaps). The ARF generator corrects the effective area for the pixel gaps by applying a constant energy-independent factor, using keywords stored in CalDB for the sanitized pixel geometry. That is the default behavior. However, `xaarfgen` is capable of a more sophisticated treatment of the pixel geometry and the pixel gaps, but this is beyond the scope of this QSG, suffice to say that more accurate treatment of the gaps can increase the gap fraction by up to $\sim 0.3\%$ (there is more than one way of treating the physical geometry). Note that `rslmkrsp` does not yet support the more sophisticated treatment of pixel gaps and requires that if an input region file is specified, it must be of the type that does not explicitly include pixel gaps, and the simple empirical correction is applied, which is the default for `xaarfgen`.

11 Extract Lightcurves and Spectra

Go to the `analysis/` directory.

Start a new XSELECT session (or issue a `clear all` command), and read in a cleaned Resolve event file:

```
xselect> read eve xa000125000rsl_p0px1000_cl2.evt .
```

Extract a Resolve full-array lightcurve in the 2–10 keV band (choose the time bin size in seconds).

```
xselect> set image det
xselect> filter pha_cutoff 4000 20000
xselect> set binsize 128.0
xselect> extr curve exposure=0.8
xselect> save curve xa000125000rsl_allpix_b128_lc.fits
```

Note: if a time bin has less than a fraction "*exposure*" of its width inside a Good Time Interval (GTI), it is rejected. If you get a warning or error about too many bins being cut, try a smaller value of exposure.

Extract a Resolve spectrum from the full array, for hi-res primary events only (Hp):

```
xselect> filter column "pixel=0:11,13:35"
xselect> filter GRADE "0:0"
xselect> extr spectrum
xselect> save spectrum xa000125000rsl_allpix_Hp_src.pi
```

Modify the first command if you want to exclude a pixel or group of pixels. (Note that certain prior versions of XSELECT had a bug if the string "pixel" in the command was capitalized.) You cannot create Resolve background spectra from the observation itself. You have to generate it using the XRISM tool `rslnxbgen` (§15). If you do produce a background spectrum, ensure that the keywords BACKSCAL in the source and background spectrum have identical values (both spectra should be drawn from the same Resolve pixels).

Clear everything and read in the cleaned Xtend event file, and then extract 0.5–10 keV lightcurves and spectra, from the background and source regions (see §9):

```
xselect> clear all
xselect> read eve xa000125000xtd_p031100010_xpc_clnevt.fits.gz .
xselect> set image det
xselect> filter region region_000125000_bgd.reg
xselect> filter pha_cutoff 83 1667
xselect> set binsize 128.0
xselect> extr curve exposure=0.6
xselect> save curve xa000125000xtd_0p5to10keV_b128_bgd_lc.fits
xselect> clear pha_cutoff
xselect> extr spectrum
xselect> save spectrum xa000125000xtd_bgd.pi
xselect> clear region
xselect> filter region region_000125000_src.reg
xselect> filter pha_cutoff 83 1667
xselect> extr curve exposure=0.6
xselect> save curve xa000125000xtd_0p5to10keV_b128_src_lc.fits
xselect> clear pha_cutoff
xselect> extr spectrum
xselect> save spectrum xa000125000xtd_src.pi
xselect> exit
```

Check that the BACKSCAL keywords in the source and background spectral files have a ratio that corresponds to the ratio of the areas of the regions used to derive the source and background spectra. If you generate a background using the XRISM tool `xtdnxbgen` (§15), check that the BACKSCAL keyword value will correctly normalize the background spectrum to the spectrum from the source region.

12 Check Attitude Stability

Examine the attitude stability during the observation (e.g., by plotting ANG_DIST, which is the angular deviation from nominal pointing, against time, using the "ehk" extended housekeeping file). You can plot ANG_DIST versus time, filtered for only the time intervals in your cleaned event file GTI, by, for example, for Resolve

```
ftselect infile=xa000125000.ehk outfile=xa000125000_gtifiltered.ehk
expr=gtifilter("xa000125000rsl_p0px1000_cl2.evt[GTI]")
```

(For Xtend, replace the event file with the cleaned Xtend file.)

Using, e.g., `fplot`, you can plot the `ANG_DIST` column versus the `TIME` column in `xa000125000_gtifiltered.ehk`. If you need to extract products with stricter attitude stability, make an appropriate GTI file. You can also make ARFs that are based on more than one attitude bin. Read the help for `xaexpmap` and `xaarfgen` for guidance on how to do that.

13 Make Exposure Map/Attitude Histogram Files

An attitude histogram is needed to generate ARFs. The following examples make histograms with one attitude bin, which should be good enough for the majority of cases. Consult the help for `xaexpmap` for directions on making histograms with more than one attitude bin. Note that the exposure map itself in the file is not required by the ARF generator, and is not used by it. The map is there for other applications.

Partial Pixel Exposures With the current software, if a Resolve pixel has undesirable behavior for only part of an observation, the pixel must either be removed for the whole observation, or *all* of the other pixels must be sacrificed during the bad time interval for that one pixel. However, Appendix B describes a work-around. If your data have any pixels with a partial "good" exposure, use the work-around to make a new `pixgtitfile` before running `xaexpmap` to make the exposure map/attitude histogram file.

Go to the `analysis/` directory. Execute:

```
punlearn xaexpmap
```

Make an exposure map & attitude histogram for Resolve:

Run `xaexpmap` (note that all input files are gzipped in this command, but edit as appropriate):

```
xaexpmap ehkfile=xa000125000.ehk.gz gtifile=xa000125000rsl_p0px1000_cl2.evt.gz
instrume=RESOLVE badimgfile=NONE pixgtifile=xa000125000rsl_px1000_exp.gti.gz
outfile=xa000125000rsl_p0px1000.expo outmaptype=EXPOSURE delta=20.0 numphi=1
stopsys=SKY instmap=CALDB qefile=CALDB contamifile=CALDB vigfile=CALDB obffile=CALDB
fwfile=CALDB gvfile=CALDB maskcalsrc=yes fwtype=FILE specmode=MONO specfile=spec.fits
specform=FITS evperchan=DEFAULT abund=1 cols=0 covfac=1 clobber=yes chatter=1
logfile=make_expo_xa000125000rsl_p0px1000.log
```

Make an exposure map & attitude histogram for Xtend:

Run `xaexpmap` (note that all input files are gzipped in this command, but edit as appropriate):

```
xaexpmap ehkfile=xa000125000.ehk.gz gtifile=xa000125000xtd_p031100010__clnevt.fits.gz
instrume=XTEND badimgfile=xa000125000xtd_p031100010.bimg.gz
pixgtifile=xa000125000xtd_a031100010_xpc_rmvpix.fpix.gz outfile=xa000125000xtd_a031100010.expo
outmaptype=EXPOSURE delta=20.0 numphi=1 stopsys=SKY instmap=CALDB qefile=CALDB
contamifile=CALDB vigfile=CALDB obffile=CALDB fwfile=CALDB gvfile=CALDB maskcalsrc=yes
fwtype=FILE specmode=MONO specfile=spec.fits specform=FITS evperchan=DEFAULT abund=1
cols=0 covfac=1 clobber=yes chatter=1 logfile=make_expo_xa000125000xtd_p031100010.log
```

In the Xtend command, the file assigned to `pixgtifile` is appropriate if you ran `xtdpixclip` or other anomalous pixel tool, but if you did not, that parameter should be set to `NONE`. *Note that you should not use any* `.fpix` *file that was produced by the XRISM pipeline because in later steps it may erroneously cause an unphysical and significant loss of effective area in the ARF* (versions of the pipeline later than August 2024 do not produce an `.fpix` file).

14 Make Spectral Response Files

Two methods for making Resolve spectral responses are presented here. One is the usual method that makes separate RMFs and ARFs using `rslmkrmf` (§14.2) and `xaarfgen` (§14.3 respectively. This can be used for Xtend in all cases, and for Resolve when the Hp fraction (not including anomalous Ls events) is approximately uniform across the array (and nearly unity). As the source brightness increases, the Hp

fraction varies from pixel-to-pixel, and due to the large dynamic range in structure of the PSF, the variations become important in distorting the relative contributions of the per-pixel XMA effective area. It is then more accurate to use the second method (§14.4), which results in a combined RMF and ARF file (an RSP file). You can examine the Hp fraction spatial distribution in the image in extension 3 of the file rsl000125000_clbr_2keVto12keV_pixMapNoLs.img that was produced by `rslbratios` in §5.1 (H6.36).

Note that for time-resolved or phase-resolved spectroscopy of bright sources, you will have to make unique RSP files for each state.

There are four sizes of the RMF (and RSP) matrix (S, M, L, and X), determined by the parameter `whichrmf`; progressively larger matrices include more and more components of the line-spread functions (LSF). In practice, the largest and most accurate matrix (size X) causes problems, so `splitrmf = yes` should be used with `rslmkrmf` or `rslmkrsp`, which produces two matrices (§14.6), one with the electron-loss continuum (ELC) at a coarser energy resolution than the other LSF components.

The task `rslmkrsp` calls `rslmkrmf` and `xaarfgen`, so whichever method you use to make the spectral responses, you have to set up the conditions for running `xaarfgen`.

14.1 Check Coordinates

ARF creation (with `xaarfgen` or `rslmkrsp`) requires the input parameters `source_ra` and `source_dec`. For a point source, these are the RA & DEC of the source respectively. These should be obtained from a reliable catalog or reference. Note that the `RA_OBJ` and `DEC_OBJ` keyword values in the data files are generally *not* sufficiently reliable, so should not be used as inputs to `xaarfgen`. For an extended source model option, `source_ra` and `source_dec` are the RA & DEC respectively of the center of the model. For image mode, `source_ra` and `source_dec` are not used for raytracing but have a small effect on the normalization of the ARF because of the $\cos\theta$ factor for incident source flux. Therefore, for image mode, `source_ra` and `source_dec` should be set to the position for which you want the derived flux to be most accurate. Alternatively, for image mode, `source_ra` and `source_dec` could simply be set to `RA_NOM` and `DEC_NOM` respectively, or to the center of the input image.

Before making the ARF or RSP files, check that the centroids established earlier for each instrument are consistent with the RA & DEC coordinates of (i) the source if it is a point source, or (ii) the intended position for which you want to make the ARFs for an extended source. Run coordpnt for each instrument, using the DET-coordinate source centers found earlier:

```
punlearn coordpnt

coordpnt input="RDETX0,RDETY0" outfile=NONE telescop=XRISM instrume=RESOLVE
teldeffile=CALDB startsys=DET stopsys=RADEC ra=RA_NOM dec=DEC_NOM roll=PA_NOM
ranom=RA_NOM decnom=DEC_NOM clobber=yes

punlearn coordpnt

coordpnt input="XDETX0,XDETY0" outfile=NONE telescop=XRISM instrume=XTEND
teldeffile=CALDB startsys=DET stopsys=RADEC ra=RA_NOM dec=DEC_NOM roll=PA_NOM
ranom=RA_NOM decnom=DEC_NOM clobber=yes
```

The runs of `coordpnt` will print the output RA & DEC (as OUTX & OUTY respectively), to the screen. Check that the values are sufficiently close (within 10" or so) to what you expect them to be. If they are not, then use the outputs from coordpnt for the `source_ra` and `source_dec` input parameters to `xaarfgen`, instead of the true source coordinates. Whatever you decide to use for `source_ra` and `source_dec`, in the commands in this QSG, their numerical values will be denoted by:

`rslsrc_ra, rslsrc_dec`: Resolve RA & DEC respectively

`xtdsrc_ra, xtdsrc_dec`: Xtend RA & DEC respectively

Before running `xaarfgen` or `rslmkrsp`, in order to avoid version conflicts, delete the parameter files for `xaarfgen` and `xaxmaarfgen` in your local `pfiles/` directory. For example:

```
rm /Home/yaqoob/pfiles/xaarfgen.par
rm /Home/yaqoob/pfiles/xaxmaarfgen.par
```

14.2 Make RMFs

If the Resolve data have Hp fractions that vary significantly from pixel to pixel, go to §14.4 to make spectral responses (RSP files, which combine the RMF and ARF). Note that the "L" (large) and "X" (extra-large) matrices can slow down spectral-fitting, so consider hybrid approaches in which an "S" matrix is used for exploratory spectral fitting, followed by finer fitting with a larger matrix, possibly in restricted energy bands.

Prepare to Make a Resolve RMF

Part of the effective area is incorporated into the RMF normalization because, to model an Hp spectrum, we must account for the Hp fraction accurately. However, Resolve data are known to have branching ratio anomalies, and an energy-dependent fraction of Ls events that are not due to the source, and this compromises calculation of the true Hp fraction. Currently it is not known how to calculate the true number of false Ls events, so we can only calculate lower or upper limits on the effective area, corresponding to the assumptions that the Ls events are all real or all false. For sources that have true source photons that are predominantly Hp (i.e., not bright sources), we can assume that all Ls events are false. For bright sources, or for cases that are ambiguous, we must calculate two response files that correspond to lower and upper limits on the effective area, using `rslmkrsp` (§14.4).

For weak sources, we first make a *special event file that will only be used for input to* `rslmkrmf`. This event file has all Ls events removed (corresponding to the assumption that all Ls events are false). Since the Hp spectrum at the lowest and highest energies may still have non-source Hp events, the most accurate result will be obtained if we only allow `rslmkrmf` (or `rslmkrsp`) to calculate the Hp fraction over a restricted, "clean", energy range. Here we choose 3 to 10 keV. Run the following command to make the special event file:

```
ftcopy infile="xa300036010rsl_p0px1000_cl2.evt[EVENTS][(PI>=6000)&&(PI<=20000)&&(ITYPE<4)]"
outfile=xa300036010rsl_p0px1000_UPR.evt copyall=yes
```

The file `xa300036010rsl_p0px1000_UPR.evt` *should never be used for regular analysis.*

Make a large full-array Resolve RMF:

In the commands below, the substring "targetname" is a string that you choose to help associate the celestial target with the response file.

```
punlearn rslmkrmf
```

```
rslmkrmf infile=xa000125000rsl_p0px1000_UPR.evt
outfileroot=targetname_obs1_60k_60k_Hp_allpix_L
regmode=DET whichrmf=L resolist=0 regionfile=ALLPIX eminin=0.0 dein=0.5 nchanin=60000
useingrd=no eminout=0.0 deout=0.5 nchanout=60000 pixeltest=CENTER
```

Make a small full-array Resolve RMF:

```
punlearn rslmkrmf
```

```
rslmkrmf infile=xa000125000rsl_p0px1000_UPR.evt outfileroot=targetname_obs1_60k_60k_Hp_allpix_S
regmode=DET whichrmf=S resolist=0 regionfile=ALLPIX pixeltest=CENTER
```

Neither of the above matrices includes the "ELC" (electron-loss continuum). If the ELC is not included in the response matrix, the source may appear to have a soft excess. See §14.6 for instructions on how to make an extra-large (`whichrmf=X`) Resolve response matrix.

If the Resolve spectrum is extracted from a subset of the full-array pixels, the same subset must be specified in the `rslmkrmf` command. This is best realized by using the `pixlist` parameter with `regionfile` set to "NONE". Read the help files for `rslrmf` and `rslmkrmf` for details. Note that in versions of the software before HEASoft 6.36, the default value of the parameter `pixeltest` gave an incorrect pixel list for a given

region file, and this results in the overall RMF and effective area for a given sub-array region potentially being incorrect.

Make an Xtend RMF:

Now make the Xtend RMF.

```
punlearn xtdrmf
```

```
xtdrmf infile=xa000125000xtd_src.pi outfile=xa000125000xtd_p031100010_src.rmf
rmfparam=CALDB eminin=200 dein="2,24" nchanin="5900,500" eminout=0 deout=6 nchanout=4096
```

14.3 Make ARFs

The ARF generator, `xaarfgen`, has complex functionality, and the commands given here will make ARFs for the simplest case of a point source, using a single, mean, attitude. See §14.7 and §14.8 for extended sources. It is important that you read the help for `xaarfgen` to understand all of the functionality and options.

Raytracing event file The raytracing event file parameter (`xrtevtfile`) can be an input or output: `xaarfgen` checks whether a file with the given name exists in the current directory. If it does not, `xaarfgen` launches fresh raytracing, otherwise it re-uses the existing file. This is useful (and much quicker) for making ARFs for different region files if nothing else is different.

numphoton Due to the way `xaarfgen` works, indefinitely increasing the number of raytracing photons (`numphoton`) *does not increase the statistics indefinitely*. To achieve greater statistical accuracy for the effective area you have to run `xaarfgen` or `rslmkrsp` many times with different random number seeds and combine the results. This topic, along with how to estimate systematic errors on the effective area, is beyond the scope of this QSG.

The erange Parameter The ARF generator is not designed to work below 0.3 keV, and data below 0.3 keV are outside of the official XRISM bandpass. At the upper end, the XMA is not calibrated above 17.5 keV. In `xaarfgen` and `rslmkrsp` you can request the energy range of the effective area function. However, you will not usually get the exact requested range because of the way the ARF generator works. This is a feature not a bug. On the upper end of the energy range, choose a value that is ~ 1 keV higher than desired. Note that narrower energy ranges will result in faster run times. However, if the energy band is narrower than ~ 4 kev wide, the accuracy of the ARF will be compromised.

minphoton The meaning of this parameter is commonly misinterpreted. It is *not* necessarily the minimum number of raytracing photons per energy per attitude bin needed to make an ARF with sufficient statistical quality. Rather, it is the minimum number of photons per energy per attitude bin needed for the ARF generator to apply its alogorithms and methodology. The default value of 100 ensures that the code will not crash, yet allows you to examine which energy bins that need the most attention (under the heading "PHOTONS PER ENERGY" in `xaxmaarfgen.log`). Generally, you need at least a few thousand photons per bin, but it varies case-by-case and is not predictable in advance. After one run you can examine `xaxmaarfgen.log` and scale up `numphoton` for another run. You could do this incrementally until changes in the effective area are acceptably small (with increasing values of `numphoton`). It is not recommended to set `minphoton` to a high number to start with, because you may waste a lot of time with repeated failures after long run times. You may also run into memory problems, and/or create raytracing files that are too large to work with.

Region Files For Resolve you can only use regions in DET coordinates. For Xtend you can use region files in DET or RA/DEC coordinates.

Much more information about how the ARF generator works can be found in the paper Spectral response and effective area functions of the Hitomi imaging instruments (the *XRISM* functionality is the same as that for *Hitomi*, with some improvements since the paper was published).

Make a Point-Source ARF for Resolve

If the Resolve data have Hp fractions that vary significantly from pixel to pixel, use §14.4 to make spectral responses (RSP files, which combine the RMF and ARF). The output files from `rslbratios` with "pixMap" in the names show the spatial distribution of grade fractions (see §5.1), but beware of these being skewed by false Ls events.

To make a full-array point-source ARF for Resolve, run the following:

```
punlearn xaarfgen
```

```
xaarfgen xrtevtfile=raytrace_xa000125000rsl_p0px1000_ptsrc.fits source_ra=rslsrc_ra
source_dec=rslsrc_dec telescop=XRISM instrume=RESOLVE
emapfile=xa000125000rsl_p0px1000.expo regmode=DET regionfile=region_RSL_det.reg
sourcetype=POINT rmffile=targetname_obs1_60k_60k_Hp_allpix_S.rmf erange="0.3 18.0 2.0 8.0"
outfile=xa000125000rsl_p0px1000_ptsrc.arf numphoton=300000 minphoton=100 teldeffile=CALDB
qefile=CALDB contamifile=CALDB obffile=CALDB fwfile=CALDB gatevalvefile=CALDB
onaxisffile=CALDB onaxiscfile=CALDB mirrorfile=CALDB obstructfile=CALDB
frontreffile=CALDB backreffile=CALDB pcolreffile=CALDB scatterfile=CALDB
mode=h clobber=yes seed=7 imgfile=NONE
```

Note: In the above command, substitute the name of the RMF that you chose to make. The output ARF energy grid will be made to match the input energy grid of the RMF file that you supply in the above command. The ARF will not work with an RMF that has a mismatched energy grid.

If the spectrum that you will be fitting is not from all 35 pixels in the array (e.g., if you are excluding pixel 27), you should substitute the region file `region_RSL_det.reg` with a region file that is appropriate for the pixels that you selected to make the spectrum (see §10).

Sub-array spectroscopy and off-axis sources The `xaarfgen` commands in this QSG are not optimized for sub-array spectroscopy, nor are they optimized for the rarer cases of off-axis sources, nor for contributions from large off-axis regions of extended sources. Smaller regions capture fewer raytracing photons, so require larger values of `numphoton` and longer run times. Off-axis angles greater than $\sim$ 5 arcmin may require unreasonable photon numbers. It is not possible to predict in advance what value of `numphoton` you need. Try increasingly larger values until you get two ARFs with an acceptably small difference. If `numphoton` exceeds 10^7 you will run into memory problems and unreasonable sizes of the output events file. In such cases you have to split the problem into multiple runs and then compactify and combine the raytracing files made by `xaarfgen`. However, details on how to do this are beyond the scope of this QSG.

Make a Point-Source ARF for Xtend

```
punlearn xaarfgen
```

```
xaarfgen xrtevtfile=raytrace_xa000125000xtd_p031100010_boxreg_ptsrc.fits
source_ra=xtdsrc_ra source_dec=xtdsrc_dec telescop=XRISM instrume=XTEND
emapfile=xa000125000xtd_a031100010.expo regmode=DET
regionfile=region_000125000_src.reg sourcetype=POINT
rmffile=xa000125000xtd_p031100010_src.rmf erange="0.3 18.0 2.0 8.0"
outfile=xa000125000xtd_p031100010_ptsrc.arf numphoton=300000 minphoton=100
teldeffile=CALDB qefile=CALDB contamifile=CALDB obffile=CALDB fwfile=CALDB
onaxisffile=CALDB onaxiscfile=CALDB mirrorfile=CALDB obstructfile=CALDB
frontreffile=CALDB backreffile=CALDB pcolreffile=CALDB
scatterfile=CALDB mode=h clobber=yes seed=7 imgfile=NONE
```

If you want to use an RA/DEC region, set `regmode=RADEC`.

14.4 Spectral Responses for Bright Sources

For Resolve spectra made from pixels that have different Hp fractions, the regular method of making an RMF and ARF is not sufficiently accurate, so we must use `rslmkrsp` (H6.36). This tool runs `rslmkrmf` and

`xaarfgen` to make a combined RSP response file that incorporates the combined RMF and ARF. The tool has almost all of the same input parameters as `rslmkrmf` and `xaarfgen`, so it has the same functionalities as the individual tools, but the effective areas for each pixel correctly account for the pixel-dependent grade fractions.

The same caveats on off-axis angle and sub-array spectroscopy described for `xaarfgen` in §14.3 apply to `rslmkrsp`. In addition, since the Hp distribution can vary with time in variable sources, you must make unique RSP files for each state for time-resolved or phase-resolved spectroscopy. There are other effects that should be considered for bright-source analysis that are beyond the scope of this QSG. More information can be found in the *XRISM Data Analysis Guide* and references therein.

As explained in §14.2, it is currently not possible to distinguish between real and false Ls events, giving rise to an uncertain Hp fraction, so we have to make two RSP files that bracket lower and upper limits on the effective area corresponding to the uncertainty in the Hp fraction. The tool `rslmkrsp` can use a regular event file as input because it treats the Ls events internally, with the Boolean parameter `includels`. The tool also allows the energy range for clean grade fraction calculations to be specified by the parameters `gfelo` and `gfehi`. Run the following commands to make the two RSP files:

Lower Limit Effective Area

```
punlearn rslmkrsp

rslmkrsp inevtfile=xa000125000rsl_p0px1000_cl2.evt
xrtevtfile=raytrace_xa000125000rsl_p0px1000_ptsrc.fits source_ra=rslsrc_ra source_dec=rslsrc_dec
emapfile=xa000125000rsl_p0px1000.expo regmode=DET regionfile=region_RSL_det.reg sourcetype=POINT
erange="0.3 18.0 2.0 8.0" numphoton=600000 minphoton=100 teldeffile=CALDB qefile=CALDB
contamifile=CALDB obffile=CALDB fwfile=CALDB gatevalvefile=CALDB onaxisffile=CALDB onaxiscfile=CALDB
mirrorfile=CALDB obstructfile=CALDB frontreffile=CALDB backreffile=CALDB pcolreffile=CALDB
scatterfile=CALDB seed=10439 outfileroot=targetname_obs1_60k_60k_Hp_allpix_L regmode=DET whichrmf=L
resolist="0" splitrmf=no splitcomb=no includels=yes gfelo=3.0 gfehi=10.0
```

Upper Limit Effective Area

```
punlearn rslmkrsp

rslmkrsp inevtfile=xa000125000rsl_p0px1000_UPR.evt
xrtevtfile=raytrace_xa000125000rsl_p0px1000_ptsrc.fits source_ra=rslsrc_ra source_dec=rslsrc_dec
emapfile=xa000125000rsl_p0px1000.expo regmode=DET regionfile=region_RSL_det.reg sourcetype=POINT
erange="0.3 18.0 2.0 8.0" numphoton=600000 minphoton=100 teldeffile=CALDB qefile=CALDB
contamifile=CALDB obffile=CALDB fwfile=CALDB gatevalvefile=CALDB onaxisffile=CALDB onaxiscfile=CALDB
mirrorfile=CALDB obstructfile=CALDB frontreffile=CALDB backreffile=CALDB pcolreffile=CALDB
scatterfile=CALDB seed=10439 outfileroot=targetname_obs1_60k_60k_Hp_allpix_L regmode=DET whichrmf=L
resolist="0" splitrmf=no splitcomb=no includels=no gfelo=3.0 gfehi=10.0
```

Note that the second command uses the raytracing file made by the first command (so runs faster), because the two runs do not change anything relating to the XMA effective area.

14.5 Sources with Dust Scattering Halos

X-ray binaries with dust-scattering halos are extended sources from the point-of-view of creating an ARF (they do not change the PSF because that is a property of the telescope). The tool `dustyarfmod`, given a table model of dust scattering, performs raytracing to create a file that can be used by `xaarfgen` to modify a point-source ARF. However, `dustyarfmod` currently does not give the correct results for Resolve data that have a variable Hp fraction across the array. The tool can be used for Xtend, and for Resolve data that have Hp $\sim$ 1 in all pixels. Refer to the help for details on how to use `dustyarfmod`. Unfortunately, the sources that have dust halos generally are very bright sources, which are just the kind of sources that do have widely varying Hp fractions across the array. Suitable tool updates or innovations may become available in the future.

14.6 Split Response Matrices

Extra-large (`whichrmf=X`) Resolve response matrices (RMF or RSP files) include all of the physical components of the line-spread-function, but have a very large size, of the order of ~ 7 GB, and this can cause problems. A solution is to use the option to split the response function into two matrices, a coarse-grid one for the ELC, and a fine-grid one for the other components. Read the help for `rslrmf` and `rslmkrmf`/`rslmkrsp` for details.

To make split response matrices run `rslmkrmf` (§14.2) or `rslmkrsp` (§14.4) with `splitrmf=yes` and `splitcomb=yes`. This will produce two matrices in a single file, with a name that ends in `_comb.rmf` or `_comb.rsp`. The single (dual) response matrix file can be loaded into XSPEC in the usual way (along with a single ARF if the matrices are RMFs), and XSPEC will automatically know what to do with the dual matrix file.

14.7 Extended Sources

There are three options for making ARFs for extended sources (with `xaarfgen` or `rslmkrsp`). Here we address only "image mode" (the most common), invoked by setting `sourcetype=image` (consult the help for the other two methods, which are flat, uniform emission, and the β-model).

Image mode requires an X-ray image in RA/DEC coordinates from an X-ray mission that has a higher spatial resolution than *XRISM* (e.g. *Chandra* or *XMM-Newton*). The third and fourth numbers in the parameter `erange` specify the lower and upper bounds respectively, of the energy range (in keV) from which this image was constructed.

For Resolve, after making the ARF, you can run `xmatraceback` (§14.8) to create RA/DEC images for every pixel (and the full array) that show the spatial distribution of photons that contribute to a given region on the detector. This is based on the raytracing performed by `xaarfgen`, and is useful for quantifying PSF mixing. If you are going to run `xmatraceback` later, after the Resolve `xaarfgen` or `rslmkrsp` run, rename the output file `heasim_events.fits` to something different, otherwise it will get over-written by the next `xaarfgen` or `rslmkrsp` run. `xmatraceback` is also useful for relating the ARF normalization to flux in the input image.

Here is the command for making a Resolve full-array ARF for extended source, using the 2–8 keV input image `chandra_image.fits`

```
punlearn xaarfgen
```

```
xaarfgen xrtevtfile=raytrace_xa000125000rsl_p0px1000_imgmode.fits source_ra=rslsrc_ra
source_dec=rslsrc_dec telescop=XRISM instrume=RESOLVE
emapfile=xa000125000rsl_p0px1000.expo regmode=DET regionfile=region_RSL_det.reg
sourcetype=IMAGE rmffile=targetname_obs1_60k_60k_Hp_allpix_S.rmf erange="0.3 18.0 2.0 8.0"
outfile=xa000125000rsl_p0px1000_imgmode.arf numphoton=300000 minphoton=100 teldeffile=CALDB
qefile=CALDB contamifile=CALDB obffile=CALDB fwfile=CALDB gatevalvefile=CALDB
onaxisffile=CALDB onaxiscfile=CALDB mirrorfile=CALDB obstructfile=CALDB
frontreffile=CALDB backreffile=CALDB pcolreffile=CALDB scatterfile=CALDB
mode=h clobber=yes seed=7 imgfile=chandra_image.fits
```

Rename the `heasim_events.fits` file for later use:

```
mv heasim_events.fits heasim_events_rsl000125000.fits
```

Spatially Variable Hp Fraction If the source is bright enough and has complex morphology, the Hp fraction may vary significantly across the Resolve array. In that case, you must use `rslmkrsp` (§14.4), otherwise the per-pixel weights of the LSFs and effective area versus energy functions will not be correct.

The Xtend command, for a spectrum extracted from the detector coordinates region `regionfile=region_000125000_src.reg`, is:

```
punlearn xaarfgen
```

```
xaarfgen xrtevtfile=raytrace_xa000125000xtd_p031100010_imgmode.fits
source_ra=xtdsrc_ra source_dec=xtdsrc_dec telescop=XRISM instrume=XTEND
emapfile=xa000125000xtd_a031100010.expo regmode=DET
regionfile=region_000125000_src.reg sourcetype=IMAGE
rmffile=xa000125000xtd_p031100010_src.rmf erange="0.3 18.0 2.0 8.0"
outfile=xa000125000xtd_p031100010_imgmode.arf numphoton=300000 minphoton=100
teldeffile=CALDB qefile=CALDB contamifile=CALDB obffile=CALDB fwfile=CALDB
onaxisffile=CALDB onaxiscfile=CALDB mirrorfile=CALDB obstructfile=CALDB
frontreffile=CALDB backreffile=CALDB pcolreffile=CALDB
scatterfile=CALDB mode=h clobber=yes seed=7 imgfile=chandra_image.fits
```

Should you use a soft or hard image if they have different spatial distributions? The effective area does not depend on the spectrum of the source, but on the spatial distribution of the source. However, the *detected* spatial distribution may clearly depend on the spectrum. Nevertheless, the key factor is the *matter distribution of the emitter*. There are two distinct cases:

1. The difference in soft and hard images is an artifact of detector sensitivity. If the detector had perfect efficiency and no limitation on sensitivity, at all energies, even a steep spectrum will produce the same images in the soft and hard bands if the *emission distribution is intrinsically uniform*. The practical question is, which image should you use for a real detector? It should be the one that corresponds to the energy range encompassing the largest effective area out of the different images that you have. However, you can always try all of the images that you have, and quantitatively assess the differences.

2. The source has intrinsic variation in the spatial distribution of the emission with energy. An example would be emission from a dust-scattering halo. The ARF generator is not designed for energy-dependent spatial distributions. For dust scattering halos you can use the tool `dustyarfmod` to correct the ARF (see §14.5). However, in general, the only option is make ARFs for 2 to 5 images made in different energy bands and assess the differences in the ARFs.

Flux Normalization for Extended Sources For extended sources, the ARF made by any of the choices for `sourcetype` corresponds to the flux of the *entire spatial extent of the input source* (as it should). For image mode, this corresponds to the flux of the entire input image. However, spectral fitting a model using an extended-source ARF to a spectrum that is made from a smaller region than the size of the input source (which will almost always be the case), will *not* give you the correct flux from that smaller region. You have to correct the flux obtained from spectral fitting in order to account for the size of the input source. The correction factor will not be trivial to calculate for a complex input image, but for Resolve, you can use `xmatraceback` to help (see §14.8). In the most naïve sense, the correction for an input source with uniform emission can be estimated to be the ratio of the area of the small region to the area of the whole input source.

Units of the Input Image The units of the values populating the input image pixels do not matter. The image is re-sampled by `xaarfgen` and converted into a normalized probability distribution.

Quality of the Input Image and Zero-valued Pixels Zero-valued pixels in the image and zero-valued pixels in the non-active image area (i.e., where there is no detector) are treated the same in the sense that they will make no contribution to the ARF. This is not desirable behavior (zero-valued pixels in the active and non-active areas of the image should not be treated in the same way). Therefore it is your responsibility to ensure that the quality of the input image is sufficiently high that there are a negligible number of zero-valued pixels in the active image area. If this cannot be avoided because the source is genuinely weak, then using `xaarfgen` in image mode is not appropriate anyway. You should instead use `sourcetype=flatcircle` or `sourcetype=betamodel`.

Point Sources and High Contrast Regions in the Input Image Bright point sources should be removed from the input image and separate point-source ARFs made for them. If there are regions of the image that are much brighter than most of the rest of the image, they will hog all of the raytracing photons, resulting in an ARF that does not adequately represent the larger dimmer regions. You should make several input images that separate out high-contrast regions and make ARFs separately for the different regions. Whether

you combine these separate ARFs for spectral fitting or not depends on the spatial-spectral properties of the source.

14.8 Run xmatraceback to Investigate Resolve PSF Mixing

For a given Resolve observation, `xmatraceback` uses raytracing to create 37 images showing the spatial distribution of photons contributing to each Resolve pixel and to the full array. The individual contribution image to pixel 12 is created for reference but not included in the full-array traceback image. `xmatraceback` also creates a file (H6.36) that shows numerical values for the fractional contribution to each pixel from inside the pixel and from outside the pixel, as well as the corresponding fractions for the full array. This output file also facilitates normalizing model fluxes obtained from spectral fitting.

`xmatraceback` operates on two output files created by a run of `xaarfgen` or `rslmkrsp` in image mode: the heasim events file and the raytracing file. Here is an example run following the `xaarfgen` run in §14.3:

```
punlearn xmatraceback

xmatraceback xrtevtfile=raytrace_xa000125000rsl_p0px1000_imgmode.fits
heasimevtfile=heasim_events_rsl000125000.fits erange="2.0 8.0" outroot=rsl000125000trcbck
```

As well as the traceback images, `xmatraceback` also outputs many useful region files. For more details on non-default choices of other input parameters, the naming conventions of the many output files, and their contents, consult the `xmatraceback` help and the mini-guide `xmatraceback_guide.pdf` (the latter also has detailed examples).

Important Caveat It is important to remember that `xmatraceback` results are *simulations* of the XMA and do not include detector effects. In particular, the results pertain to events *summed over all grades*, not just Hp events. The totality of events refers to true source events, not including false Ls events (or any other non-source events). In practice, this is only a problem if the source has significant variation in grade fractions per pixel across the array, *and* you try to combine individual pixel trace-back images for a pixel group in order to assess PSF mixing for only the grade used in the spectrum you are fitting. The trace-back image for the full array corresponds to the sum of grades for true source events.

14.9 Spatial-Spectral Mixing (SSM)

Spatial-Spectral Mixing (SSM) refers to the fact the spectrum from a particular detector region has source contributions from outside that region due to broad structure in the PSF, and if that outside contribution has a different spectrum to the contribution from inside the region, the observed spectrum will be a mix of the different spectra. Therefore, an ARF (or RSP file) is required for every outside region that contributes, as well as for the self-contribution of the region the spectrum to be fitted comes from. For sub-array analysis, if the source has significant variation across the array, many ARF (or RSP) pairs may be required, making the analysis very complicated.

First you need to select a *Chandra* or *XMM-Newton* input image, and from that image make FITS sub-image files that correspond to the region from which the spectrum is extracted, and to each of the regions outside of the spectral-extraction region. Then make the ARFs or RSPs with `xaarfgen` or `rslmkrsp` respectively, in image mode, using the derived *Chandra* or *XMM-Newton* sub-images as input. (If RMFs are being employed, `run rslmkrmf` to make them.) After each run, rename the `heasim_events.fits` file to something informative and unique to the run. Before setting up the spectral-fitting, run `xmatraceback` on the renamed `heasim_events.fits` file from each run. The outputs will indicate graphically and numerically (§14.8), whether you can neglect (considering your science goals), any of the contributing regions, thereby reducing the number of terms you have to include in the spectral fitting.

SSM is a vast subject, and examples and more detailed instructions are beyond the scope of this guide.

14.10 Common Error Messages & Their Remedies

- **Insuffcient number of photons** When running `xaarfgen` or `rslmkrsp`, if there is a warning or error

that there is an insufficient number of raytracing photons, it does not necessarily mean that you should increase the number of input photons. It may mean that the raytracing photons are missing the region on the detector, which could be due to an error in coordinates and/or the region file. It could also mean that the requested energy range extends into a region with negligible effective area. All of these items should be checked first, before increasing the number of input photons.

- **The ARF has negative values** Same possible causes as above.

15 Non-Xray Background (NXB)

At the time of writing, the methodology and databases for treating the NXB for both Resolve and Xtend are undergoing continuous improvement and development. For the latest information and guidance, go to the *XRISM* data analysis web pages, and consult the help for the tools `rslnxbgen` and `xtdnxbgen`.

16 Appendix A: Data Analysis with Resolve Mp Data

Is this an option for your data? From the output results files obtained by running `rslbratios` (§5.1), determine whether you have a sufficiently high (Mp/Hp) ratio and number of Mp events in your data to warrant considering trying to extract science out of them. If not, you can skip this section.

Difference between hi-res and mid-res sevents Triggered events that have no other events 70.72 ms before or after that event are processed using the full optimal filter template and classified as hi-res. Events that have another event in the interval 18.32 ms and 70.72 ms before or after that event are processed using a truncated template and classified as mid-res. The calibrated pixel-dependent spectral resolution and gain (conversion of pulse height to energy) are distinct for these two event grades. In addition, mid-res events are categorized as primary (Mp) or secondary (Ms) based on whether the time separation from the *preceding* event is greater than or less than 70.72 ms, respectively. All hi-res (Hp) events are primary events. Because the detector may not fully re-equilibrate following the energy deposited by the preceding event, an additional correction is applied as part of secondary event energy assignment.

Mp Correction There is a differential offset between mid-res and hi-res events that depends on the temperature of the Resolve XBOX that conducts the initial signal processing. Because the XBOX temperature in orbit is different to what it was during ground energy scale calibration, a relative in-flight offset between mid-res and hi-res events emerges and needs to be corrected on the ground. This pixel-dependent adjustment as a function of XBOX temperature, varies depending on spacecraft orientation with respect to the Sun and Earth, is now calibrated based on in-flight measurements, and the relevant data is included in XRISM CALDB 10 and later. The *XRISM* pipeline scripts in HEASoft 6.35.2 or later apply the correction to Mp events. Therefore, if you perform the mandatory reprocessing in §4, the Mp correction will have been applied. It is not recommended to apply the correction in a standalone mode (by running the tool `rslmpcor`) because the interdependence of energy and the STATUS flag may result in incorrect screening.

The OBSIDs processed with versions of the processing pipeline that applied the mid-res correction will have energy-scale reports that include an additional assessment of calibration-line, energy scale and resolution for Mp events. For data that have not had the Mp correction applied, the offset between the Hp and Mp energy scales will be evident. Note that the Mp energy-scale assessments are only for the full array (as opposed to per pixel), due to more limited statistics of the Mp events compared to Hp events.

Ms Events Note that further calibration work on Ms events needs to be done in order to bring the energy scale accuracy on par with that for Mp events, so it is currently not recommended to use Ms events.

Accuracy of the Mp Correction, and the Mp LSF The Mp correction reduces the relative offset of Mp events from ~ 3 eV to less than 0.5 eV at 5.9 keV in Resolve spectra extracted from the full Resolve array. The absolute offsets for Hp events are, by comparison, generally < 0.1 eV. Moreover, because the residual Mp offsets are different for different pixels, a slight broadening of the full-array energy resolution may result. Additional in-flight calibration may further reduce the Mp offset in the future. An in-flight update of the mid-res line-spread-functions (LSFs) is included in *XRISM* CalDB 11 onwards.

Using Mp Data For the most accurate work it is recommended to analyze Hp and Mp data separately and not try to combine them. Follow the same procedures in this QSG as for Hp data, with the following modifications to some of the commands:

- When extracting images, lightcurves and spectra from event files, specify `ITYPE=1`
- When running `xaspecratios`, set `resolist=1`
- Use `rslmkrsp`, not `rslmkrmf` and `xaarfgen`, and set `resolist=1`
- Remember that `xmatraceback` results are *simulations* and the tool does not use real data - the results refer to the *sum of all grades for true source events* (i.e. not including false Ls events).

You can use the same exposure map/attitude histogram files that you made for the Hp analysis.

17 Appendix B: Partial Exposure Resolve Pixels

This section describes a work-around to account for Resolve pixels that have undesirable behavior for only part of an observation. The following steps may be followed to exclude data from a single pixel over a specified time interval:

[**1**] Identify the problematic pixel (Pixel 'N'), as well as initial and final times (in terms of XRISM mission elapsed time as in the event file extension TIME, and GTI extension START and STOP, columns) over which events from that pixel are to be excluded. In this example, the initial and final times of the exclusion interval are 'TI' and 'TF' respectively.

[**2**] Apply the necessary post-pipeline screening based on RISE_TIME and STATUS[4] (§7). For event files that are going to be used to make RMF files with `rslmkrmf`, apply the anomalous Ls event treatment to the cleaned event file (as described in §14.2). Generally, screening for non-astrophysical events should always be applied first, and the resulting event file utilized as input for `rslmkrmf` , with screening of astrophysical events based on quality (e.g., pertaining to certain pixels and/or grades) applied afterwards. For example, for OBSID 201005010, TI= 202176000 s, TF= 202451340.78125 s, Pixel=7 the command is as follows:

```
ftselect infile="xa201005010rsl_p0px1000_cl2_forrmf.evt[EVENTS]"
outfile=xa201005010rsl_p0px1000_cl2_forrmf2.evt
expr="PIXEL!=7.OR.(PIXEL==7.AND.((TIME.GE.201717360.078125.AND.TIME.LT.202176000).OR.
(TIME.GT.202451340.78125.AND.TIME.LE.202451340.78125))"
```

where 201717360.078125 represents the beginning, and 202451340.78125 the end, of the observation (here, the same as TF).

[**3**] Trim the archived "pixgti" file to create a stripped-down version suitable for exposure map generation. While not strictly necessary, it is cleaner to proceed with a file that is internally consistent.

```
ftcopy "xa201005010rsl_px1000_exp.gti[GTIPIXELOFF][col START,STOP,PIXEL,DETX,DETY]"
xa201005010rsl_px1000_exp_off_small.gti copyall=no
```

[**4**] Filter the reduced "pixgti" file to delete the Pixel 7 (in this example) GTI in the interval between TI and TF. This is to be replaced by a single GTI covering that interval.

```
ftselect infile="xa201005010rsl_px1000_exp_off_small.gti[GTIPIXELOFF]"
outfile=xa201005010rsl_px1000_exp_tmp1.gti expr="PIXEL!=7.OR.(PIXEL==7.AND.
((START.GE.201717360.078125.AND.STOP.LT.202176000).OR.
(START.GT.202451340.78125.AND.STOP.LE.202451340.78125)))"
```

[**5**] Append a copy of the first Pixel 7 (N=7 in this example) table entry in the file created in step (4) to itself (note this is done by specifying the (N+1)th row of the file).

```
ftmerge infile="xa201005010rsl_px1000_exp_tmp1.gti[GTIPIXELOFF],
xa201005010rsl_px1000_exp_tmp1.gti[GTIPIXELOFF][#row==8]"
outfile=xa201005010rsl_px1000_exp_tmp2.gti
```

[**6**] Set the START and STOP values of the new row to the correct values (TI and TF).

```
ftcalc infile="xa201005010rsl_px1000_exp_tmp2.gti[GTIPIXELOFF]"
outfile=xa201005010rsl_px1000_exp_tmp2.gti
column=START expression=202176000.0 rows=4502 clobber=yes
```

```
ftcalc infile=xa201005010rsl_px1000_exp_tmp2.gti[GTIPIXELOFF]
outfile=xa201005010rsl_px1000_exp_tmp2.gti
column=STOP expression=202451340.78125 rows=4502 clobber=yes
```

In the above commands, 4502 is the row number of the last row in the file, and can be obtained by examining the NAXIS2 keyword using `ftlist` or `fkeyprint`.

[**7**] Sort the file created in step (6) to re-locate the new GTI into their proper place.

```
ftsort infile="xa201005010rsl_px1000_exp_tmp2.gti[GTIPIXELOFF]"
=xa201005010rsl_px1000_exp_forExpoMap.gti columns=START,STOP,PIXEL
```

[**8**] Generate the exposure map in the usual way from the files created above. The output file will have a reduced value in the FRACTION column of the PARTIALEXP001 extension for the pixel under consideration.

```
xaexpmap ehkfile=xa201005010.ehk gtifile=xa201005010rsl_p0px1000_cln2_forrmf2.evt
instrume=RESOLVE badimgfile=NONE pixgtifile=xa201005010rsl_px1000_exp_forExpoMap.gti
outfile=xa201005010rsl_px1000.expo outmaptype=EXPOSURE delta=20.0 numphi=1
```

End of XRISM Quick-Start Guide