XRISM Quick-Start Guide
Version 2.3
(For data reprocessed in April 2024)

XRISM Science Data Center

Last updated: September 18, 2024

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). If you get stuck, e-mail the SDC help desk (`XRISM-SDC-help@lists.nasa.gov`).

By following this *Quick-Start Guide*, you will be able to quickly extract images, lightcurves, spectra, and response files. At the end of the guide, there will be pointers for things to do next. This version of the guide only applies to data that have been reprocessed in April 2024 or later. Earlier versions of this guide only apply to data that were not reprocessed in April 2024 or later.

It is highly recommended to read the *Hitomi Data Analysis Guide* because a lot of the information in there is still relevant, with appropriate changes to instrument and software task names. It can be downloaded from `https://heasarc.gsfc.nasa.gov/docs/hitomi/analysis/hitomi_analysis_guide_20220816.pdf`. In general, *XRISM* tools that have a *Hitomi* inheritance have the strings "ah", "sxs", "sxi" replaced by "xa", "rsl", and "xtd" respectively.

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 6.34 or later.
2. Installation of version 8 or 9 of the *XRISM* CalDB (the XRISM CalDB can 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.

You will also need to set the environment variable XSELECT_MDB to $HEADAS/bin/xselect.mdb.

Before beginning data analysis, familiarize yourself with the web page *Things to Watch Out For* (in XRISM data, analysis tools, and calibration). Hereafter, this will simply be referred to as the *TTWOF*. This *Quick-Start Guide* also refers to the *TTWOF* for more details and/or caveats on some of the procedures, as well as for variations on some of the procedures. Also check the *QSG* page for errata for this guide, and for later versions of this guide.

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 the top-level directory). However, until further notice, you should not need to reprocess the April 2024 data.

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*. 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 Initial Set-Up and Checks

Examine the quick-look preview products under:

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

Do not use these products for science or calibration analysis. Note that there is no ARF for either instrument in the preview products.

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/resolve/event_cl/ (Resolve)
000125000/xtend/event_cl/ (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/auxil/xa000125000.ehk.gz
000125000/resolve/event_uf/xa000125000rsl_px1000_exp.gti.gz
000125000/xtend/event_uf/xa000125000xtd_p031100010.bimg.gz
```

5 Assess Xtend Flickering Pixels and Run searchflickpix if Necessary

Follow instructions in the Appendix to determine whether you need to run `searchflickpix` on the Xtend data. If you don't need to run `searchflickpix`, you can skip this section and go straight to the next section. If you do need to run `searchflickpix`, follow the instructions in the Appendix to obtain appropriate values of the following input parameters for `searchflickpix`:

```
logprob2
bthresh (will be set equal to bgdlevel from the Appendix)
cellsize
n_division
grade
```

You will need to run `searchflickpix` twice, because an updated version of `xtdflagpix` is needed, and this latter tool has not yet been released. Execute the following commands in the `analysis/` directory:

```
punlearn searchflickpix
```

```
searchflickpix infile=xa000125000xtd_p031100010_cl.evt outfile=xa000125000xtd_a031100010.fpix
cellsize=cellsize logprob1=10 logprob2=logprob2 iterate=no n_division=n_division
bthresh=bgdlevel xcol=DETX ycol=DETX grade=grade cleanimg=no
```

```
searchflickpix infile=xa000125000xtd_p031100010_cl.evt outfile=xa000125000xtd_p031100010_cl2.evt
cellsize=cellsize logprob1=10 logprob2=logprob2 iterate=no n_division=n_division
bthresh=bgdlevel xcol=DETX ycol=DETX grade=grade cleanimg=yes
```

```
ftappend xa000125000xtd_p031100010_cl.evt+2 xa000125000xtd_p031100010_cl2.evt
```

6 Resolve Rise-Time Screening

Execute the following additional screening for Resolve data:

```
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
```

Note that the STATUS[4] part of the above may screen out false-positive frame events, and users may wish to omit that part of the expression for bright sources.

The rise-time screening expression is different to that given in previous versions of this guide. If you have already started performing spectral analysis using the older expression, compare spectra made using the new expression, to determine whether you need to repeat the spectral analysis with spectra made using the new expression.

7 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 the SDC to check if something went wrong with the reprocessing.

For the sake of convenience for this *Quick-Start Guide*, rename any cleaned event files that end in "cl.evt" to "cl2.evt" (for example, if you did not need to run `searchflickpix`). You will then have, in this example,

```
xa000125000rsl_p0px1000_cl2.evt (Resolve) and
xa000125000xtd_p031100010_cl2.evt (Xtend)
```

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 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 the 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_cl2.evt .
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.

8 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 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
```

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, high-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 pixel 27. You cannot create Resolve background spectra from the observation itself. You have to generate it using the XRISM tool `rslnxbgen`, which requires sufficient time to have passed since the XRISM launch in order to have a database that it can use. 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 §7):

```
xselect> clear all
xselect> read eve xa000125000xtd_p031100010_cl2.evt .
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`, check that the BACKSCAL keyword value will correctly normalize the background spectrum to the spectrum from the source region.

9 Make RMFs

Make a large Resolve RMF:

```
punlearn rslmkrmf
```

```
rslmkrmf infile=xa000125000rsl_p0px1000_cl2.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
```

Make a small Resolve RMF:

```
punlearn rslmkrmf
```

```
rslmkrmf infile=xa000125000rsl_p0px1000_cl2.evt outfileroot=targetname_obs1_60k_60k_Hp_allpix_S
regmode=DET whichrmf=S resolist=0 regionfile=ALLPIX
```

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. An extra-large (`whichrmf=X`) Resolve RMF that includes all of the physical components of the line-spread-function may be made by using 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 files for `rslrmf` and `rslmkrmf` for details, and to understand other possible variations. Without the split-matrix option, an extra-large single matrix can be of the order of ~ 7 GB in size and can cause problems. The "L" and "X" matrices can also slow down spectral-fitting, so consider hybrid approaches in which an "S" matrix is used for exploratory spectral fitting, followed by finer fitting, possibly in restricted energy bands.

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.

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
```

Read the help files for xtdrmf to understand the details and possible variations.

10 Make Exposure Map/Attitude Histogram Files

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_cl2.evt.gz
instrume=XTEND badimgfile=xa000125000xtd_p031100010.bimg.gz
pixgtifile=xa000125000xtd_a031100010.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
```

11 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. It is highly recommended to read the help file for `xaarfgen` to understand all of the functionality and options.

Check Coordinates

The ARF generator 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 ARFs, 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.

Before making the ARFs, 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
```

When running `xaarfgen`, 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. These latter items should be checked first, before increasing the number of input photons.

Make a Point-Source ARF for Resolve:

```
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 0 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.

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 0 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
```

These runs are appropriate for sources that are close to on-axis. The number of photons in runs of `xaarfgen` (the parameter `numphoton`) needs to be larger for larger off-axis angles, but note that the statistics are not

Poissonian.

If you try to make ARFs for off-axis angles of ~ 6 arcmin or greater, you will run into computational hurdles (not just in run time), and special techniques are required that are beyond the scope of this guide.

Note that the optical axes in CalDB 9 and earlier are based on ground calibration, so the true off-axis angles in each instrument are uncertain. Until inflight calibration activities are completed (and the CalDB is either validated or updated), the effective areas in the ARFs should be treated as tentative. It is possible to quantify the uncertainties, but that is beyond the scope of this guide.

12 Things to Do Next

Now that you have had a quick look at your data and perhaps even done some spectral fitting, following is a list of things to do that will help to raise your data quality to a publishable level. Some tasks go back to examine things more carefully, whilst others are more advanced tasks that are not covered in this guide. Not everything will apply to you and your particular data set.

1. Examine the Xtend background lightcurve. Are there anomalous features (e.g., flares)? If so, create GTI (Good Time Intervals) to exclude those intervals, and re-extract both source and background data, using those GTI.

2. For Resolve, you cannot examine the above kinds of anomalies in the same way. However, examine the lightcurve to see if there is any anomalous behaviour compared to corresponding Xtend lightcurves, and/or from your knowledge of the celestial target. Note that there is a known issue of skipped telemetry that results in data losses that can be as high as tens of ks in some cases. The issue is under investigation, and in the meantime, fluxes derived from Resolve data should be regarded as tentative. The systematic error on fluxes is different for different observations, and unfortunately, it is not possible at the moment to quantify this error. Updates will be posted on the *TTWOF*.

3. Examine the attitude stability during the observation (e.g., by plotting the 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,
   `ftselect infile=xa000125000.ehk outfile=xa000125000_gtifiltered.ehk`
   `expr=gtifilter("xa000125000rsl_p0px1000_cl2.evt[GTI]")`
   The file `xa000125000.ehk` is in `xa000125000/auxil/`,
   and the GTI-filtered output file is `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 files for `xaexpmap` and `xaarfgen` for guidance on how to do that.

4. Check that the images in the headers of the Xtend source and background spectral (.pi) files are consistent with the corresponding regions that were used to extract the spectra. The images should show no counts outside of the regions used to extract the spectra.

5. If any of the above items give cause to re-extract products, go back to filter the cleaned event file with the additional criteria and with any new GTI files, and/or region files that were made.

6. Do the Xtend data show signs of pile-up (e.g. a depression or absence of counts in regions that should be the brightest parts of the image)? Sometimes the effects of pile-up may not be visible by eye: you may have to construct radial profiles. If there is pile-up you may be able to construct a new extraction region that avoids the pile-up area (and then re-extract lightcurves and spectra from the new region). More advanced mitigation techniques are beyond the scope of this guide.

7. Are there contaminating sources in the source or background regions (or both)? If so, you can construct exclusion regions and re-extract lightcurves and spectra.

8. Do you need to extract lightcurves and/or images from different or additional energy bands? If so, go back to the extraction stage, using different values of `pha_cutoff` in XSELECT. The PI bin sizes are 0.5 eV and 6 eV per channel for Resolve and Xtend respectively.

9. Do you need Resolve spectra that are not from the full array, but only from selected pixels? If so, go back to the spectral extraction stage (and consult the help for the extraction tool that you will use). You will also need to make a new RMF and ARF.

10. The Resolve spectrum extracted for the example in this guide is for high-res events only (ITYPE==0). Run `rslbranch` to investigate branching ratios for the different event types. If you want to analyze mid-res and/or low-res spectra, go back to the spectral extraction stage, change the grade selection, re-extract spectra, and make new RMFs. However, note that the accuracy of the calibration of non-Hp events is likely to lag behind that of Hp events.

11. It is currently not recommended to use data beyond the upper end of the official energy bandpass (12 keV). At higher energies the statistical and systematic errors in the effective area (ARF) increase substantially. Moreover, the XMA for both instruments are not calibrated above 17.5 keV. In addition, for Resolve, at high energies, there are effects of pulse clipping and energy-dependent rise-time effects that introduce additional systematics.

12. The physical detector effects that are included in the Resolve RMF, as well as other properties of the RMF, are controllable by input parameters. Read the help files for `rslrmf` and `rslmkrmf` to determine whether you need to make a more accurate Resolve RMF, or for other different properties.

13. Is your celestial target not a point source? If the source is extended, you can run `xaarfgen` for two types of azimuthally symmetric extended source. However, be sure to understand how the effective areas are normalized, in order to derive correct fluxes from spectral fitting.

14. More complex scenarios for extended source analysis that involve accounting for contributions of source photons from outside the spectral extraction region or regions require more complex procedures and analyses, which are beyond the scope of this guide.

15. The ARFs that were made in the examples do not have good statistics below $\sim$ 0.6 keV and above 10 keV. If you need more accurate effective areas outside this energy range, re-run `xaarfgen` with input parameters that will give better statistics. Estimate both the systematic *and* the statistical uncertainties in the effective areas over all of the energy band that you will be utilizing (to work out how to do this, read the help files for `xaexpmap` and `xaarfgen`, and the paper Spectral response and effective area functions of the Hitomi imaging instruments).

16. Assess the Resolve gain and spectral resolution outcomes for your observation by fitting calibration pixel spectra. You can also examine the Fe 55 Filter Wheel lightcurves and spectra.

17. If you need to, you can analyze the Xtend data in the other cleaned event files.

13 Appendix: Mitigation Strategies for Searchflickpix

The task `searchflickpix` creates a .fpix output file that contains two extensions; one is a list of events that it deems originate in flickering pixels, and the other is a list of those pixels. Alternatively, if cleanimg=yes, the output file contains an event list for pixels that are *not* deemed to be from flickering pixels (the name of the cleanimg parameter is unfortunate - the output is not an image).

It has been found that the algorithms of `searchflickpix` are vulnerable to false positives when applied to XRISM data, removing large numbers of source photons. Extended sources are affected as well as point sources, but the latter are most severely affected. It has been found that there is no "one-size-fits-all" optimal choice of input parameters to `searchflickpix` that will mitigate the problem. Mitigation depends on the count rate, exposure time, source spatial extent, and whether there is significant pile-up or not. The best parameters for re-running `searchflickpix` have to be established on a case-by-case basis.

The main problem with the `searchflickpix` methodology is that it assumes pixel-to-pixel variations in a sampling region of $\sim$ 20–100 pixels are entirely statistical. However, the PSF has structure on the scale of Xtend pixels, and inflight data show that in bright point sources, adjacent pixels have significant differences in counts that imply essentially zero statistical probabilities.

There are 3 modes of operation of `searchflickpix`:

Mode 1: Apply a simple counts cut threshold: all pixels with counts above a specified threshold are deemed to be flickering pixels.

Mode 2: Calculate a mean value for the counts over a specified region (the default is the entire chip) and then flag flickering pixels as those whose counts have Poisson probabilities that are smaller than a specified user-set critical input probability. This mode assumes the entire source is a statistical fluctuation, which is not true.

Mode 3: Scan the image with a multi-pixel square cell that has a user-specified size, and calculate the Poisson probability that the central pixel in the cell has the observed counts, given the mean of the counts in the other pixels in the cell. If this probability is less than a specified user-set critical input probability, the central pixel in the cell is deemed a flickering pixel.

The tool `searchflickpix` can be operated with:

Mode 1 only;
Mode 2 followed by Mode 3 in the same run;
Mode 3 only.

Ultimately, the `searchflickpix` tool needs modification (for example, to be able to run mode 1 and 3 in a single run, as well as other algorithmic and functional improvements). However, until then, workarounds will have to suffice. Below is a general procedure.

Note that Mode 2 is highly unpredictable because it uses a reference mean count rate that has contributions from unpredictable events, so the behaviour of `searchflickpix` is not sufficiently controllable. Therefore, we will not use Mode 2 in the following procedure.

1. Start with an Xtend event file that has not been screened by `searchflickpix` flagging (e.g., the "cl" event files in the April 2024 or later reprocessed data). An event file that has not been screened by `searchflickpix` flagging will be referred to as the SFP-free file, sfp_cl.evt.

2. Make an image from the SFP-free file (e.g., with XSELECT or ds9, with no energy selection), using the calibration sources exclusion region file described in §7. Does this SFP-free image appear to have any unusually bright or anomalous pixels in the regions that you are going to use for extracting source and background spectra and/or lightcurves? If the answer is "NO", then you can stop here, and do not need to run `searchflickpix`. Note that in ds9, under the "Scale" menu, you can see a histogram of number of pixels versus counts. From this histogram, is there a clear division (obvious gap) in the number of counts per pixel that indicates some pixels have anomalously high counts that are not due to the source? If not, *any* algorithm will have a hard time finding flickering pixels, so you can proceed without running

`searchflickpix`. However, if you do have anomalous counts, determine whether the corresponding pixels are located in regions of interest (i.e., regions that you will use for source and/or background extraction). If the pixels are not in regions of interest, you can again proceed without running `searchflickpix`.

3. From the previous step, in the distribution of number of pixels versus the counts per pixel, if there was an obvious gap in the counts per pixel, decide whether a simple lower theshold cut on the number of counts per pixel will be adequate for your purpose. In this scenario you would reject all pixels that have a total number of counts that is greater than some critical value, and this critical value would be chosen to lie in the gap in the distribution of counts. If you select this option, you can stop here and take away the following parameter values, and follow the procedure in §5:

```
cellsize=0
logprob2 = -30
n_division = 1
grade = ALL
bgdlevel = a value of counts per pixel that lies in the gap in the distribution of
number of pixels versus counts per pixel
```

4. If you reach this step, determine the largest number of counts in any pixel in the source region of interest, and call this value topctval. In an image from the SFP-free file (with no energy filtering), what is the typical number of counts in a pixel that is far from the source? Call this value bgdlevel. If there is no source-free region, you will just set bgdlevel=3, which is equal to the default value of the bthresh parameter of `searchflickpix`.

5. Next, you will run `searchflickpix` in Mode 3 only, but several times, examining the output image from the .fpix events file, after each run. First determine the number of time divisions that the tool will split the data into: this will be set to the n_division parameter. As a rough guide, start with n_division = min(1, [topctval/10)]), where topctval is from step 2. In the runs of `searchflickpix` you will vary the parameter logprob2 and/or n_division. For the first run, use

```
punlearn searchflickpix
searchflickpix infile=sfp_cl.evt outfile=evt_cl_trial1.fpix
cellsize=5 logprob1=10 logprob2=-4.0 iterate=no n_division=n_division bthresh=bgdlevel
xcol=DETX ycol=DETY grade=0
```

where bgdlevel is from step 4. Note that logprob1=10 turns off Mode 2 (as does any value of logprob1 greater than 0).

(Parameters not specified are at their default values, after "punlearn".)

Examine images made from the .fpix file in DET coordinates (e.g., using ds9), noting if any pixels in the brightest parts of the source were flagged, and how many pixels in total were flagged.

Repeat runs of `searchflickpix` 3 to 4 times, varying logprob2 down to -30.0. Then repeat these for one or two more values of n_division that are larger than the initial value. If n_division is too large, the number counts per pixel in each time interval will be too small to give meaningful statistical results. The maximum value of n_division will be roughly (topctval/10). Find the regime of the two parameters logprob2 and n_division that does not produce false positives in the brightest parts of the source, and that produces a robust and consistent total number of flickering pixels found. Also check that the segment and chip boundaries are not being "eaten away". If there is pile-up, check that the pile-up "hole" does not become larger.

If a few pixels are persistently found in the brightest part of the source, assume they are genuine anomalous pixels: even if they are not, removing them is the safest thing to do, and the ARF generator (`xaarfgen`) will later correct the effective area for the missing pixels.

If you are having trouble converging to a viable pair of parameter values for logprob2 & n_division, try increasing the cellsize to 7. For a point source, it should not be increased to a higher value than 7 because

the spatial variation in the PSF is too great on a larger scale. When you have settled on the parameters, run the final searchflickpix commands in §5 using these values (and grade=0).