xmatraceback: Special Topic Guide STG003
Version 1.0

XRISM Science Data Center
Isabella Brewer and Tahir Yaqoob

Last updated: May 28, 2026

Contents

1 Introduction

The `xmatraceback` tool is a simulation tool designed to aid extended source analysis for XRISM Resolve or Hitomi SXS. Using observations of real objects, it is impossible to determine definitively where a photon originated. However, using *simulated* images of sources, it is possible to precisely know the origin of each simulated photon. Extracting the origin of simulated photons is particularly important in extended source analysis because it can inform the user about how many of the photons that impact a particular region on the detector originate from outside of that region at the source. Although the photons hitting the detector region are from a simulated source, the `xmatraceback` tool can help to quantify contributions from photons originating from inside and outside a particular region on an extended source. `xmatraceback` helps to assess which parts of the detector are dominated by contributions from which parts of the source, and may be a valuable tool when performing spatial-spectral mixing (SSM) analysis.

In a typical analysis, a user would take a real observation from another mission with higher spatial resolution, such as Chandra, and run this observed image through either `aharfgen` or `xaarfgen`. This will produce a raytracing event file and a heasim event file, which can in turn be used as input for `xmatraceback`. When using xmatraceback, the two input event files must be generated from the same instance of `aharfgen` or `xaarfgen`. Currently, xmatraceback is only available for the XRISM Resolve or Hitomi SXS instruments.

The `xmatraceback` tool outputs 37 images and 37 pairs of region files (corresponding to 37 regions in both RA/DEC and DET coordinates), as well as a "photon fraction" file entitled `<user_input>_photon_fractions.fits`. The latter holds various calculated fractional contributions to the full array and to individual pixels (see Section 4 for details). The 37 images correspond to 36 individual pixel images and one full array image; each image shows only simulated photons that hit the corresponding detector region. Individual pixel images can be summed together to create custom regions, which may be useful when conducting an analysis. Image masks, made using the RA/DEC or DET region files, may also be applied to give an idea of how many photons originate from inside or outside of a certain detector region.

It is important to note that xmatraceback is only designed to work for a single attitude bin (this information is in the exposure map file made by ahexpmap/xaexpmap). If your other analysis was done with more than one attitude bin, you must make a new exposure map file and re-run aharfgen or xaarfgen (see Section 5.2).

Note that all of the images shown in the figures in this guide have RA and DEC units for the X and Y axis respectively. Additional documentation that may be useful includes the help files for xmatraceback, xaarfgen, and aharfgen.

2 What You Will Need

To use `xmatraceback`, you will need access to the HEASOFT software (version 6.35 or later) and a working calibration database (CalDB) version 11 or later. The XRISM CalDB files can be downloaded from the HEASARC XRISM CalDB web page. Please follow directions on the HEASARC website when installing HEASOFT. Additional instructions for setting up your CalDB can also be found on the HEASARC website.

After your HEASOFT installation is complete and you have set up your CalDB, you can now proceed to your analysis. There are two different scenarios which would require different files to get started:

- You are starting with an image from a mission with higher spatial resolution, such as Chandra or XMM-Newton, or made from a theoretical simulation. If you are starting here, you will need to run xaarfgen if you would like to simulate XRISM Resolve or aharfgen if you would like to simulate Hitomi SXS. When starting with `xaarfgen/aharfgen`, you will need, at the minimum,
  - An RA/DEC image file from an observatory with higher spatial resolution or made from a theoretical simulation
  - An exposure map file for a single attitude bin (created using xaexpmap)

  You may need additional files for `xaarfgen/aharfgen`, depending on the requirements of your analysis. The XRISM Quick-Start Guide is a useful resource that provides examples of running, among other tasks, `xaexpmap` and `xaarfgen`. After running `xaarfgen` or `aharfgen`, you will need the raytracing and heasim event files that are output from these tools, in order to run `xmatraceback`. **Please note** that the heasim event file will automatically be named `heasim_events.fits`. It is important to rename this file to something different and unique so that it is not overwritten by a subsequent run, or confused with other heasim event files.
- You already have raytracing and heasim event files from a previous run of `xaarfgen` or `aharfgen`. In this case, you are all set to run `xmatraceback`. Please remember, however, that these two files must come from the same instance of `xaarfgen` or `aharfgen`. The `xmatraceback` task cannot check for this rigorously and only tests whether the DATE keywords in the heasim and raytracing event files are consistent, but this does not guarantee that the two files are self-consistent. `xmatraceback` provides a warning if the test fails but it continues to run.

3 Parameters

The usage for `xmatraceback` is:

```
xmatraceback xrtevtfile heasimevtfile xcen ycen telescop instrume teldeffile erange
outroot cleanup clobber
```

- `xrtevtfile`: The name of the raytracing event file output from either `xaarfgen` or `aharfgen`.
- `heasimevtfile`: The name of the heasim event file produced when running either `xaarfgen` or `aharfgen`; this should be produced at the same time as the `xrtevtfile`. After generating the

heasimevtfile, it should be renamed to something different and unique; the file will be overwritten if xaarfgen or aharfgen is rerun and the heasimevtfile has not been renamed.

- xcen, ycen: The X and Y coordinates respectively, of the detector position you would like to align with the origin of the raytracing coordinates. The position is specified in DET coordinates. The default is xcen=ycen=CENTER, which selects the center of the detector (3.5,3.5). Note that if only xcen or only ycen is set to 'CENTER', then both xcen and ycen are forced to take the value 'CENTER'. If xcen=ycen=CALDB, then the optical axis stored in the CalDB is used (i.e. xcen and ycen are set to the values of the keywords OPTAXISX and OPTAXISY respectively, as they appear in teldeffile). If only xcen or only ycen is set to 'CALDB', then both xcen and ycen are forced to take the value 'CALDB'.
  - **Note** that neither xcen nor ycen affect the positioning of the detector in RA/DEC space: they only affect the positioning of the XMA optical axis relative to the detector. Setting xcen=ycen=CALDB selects the ground-measured optical axis position, which may have changed inflight, but the optical axis was not measured inflight. Strictly speaking, only xcen=ycen=CALDB is self-consistent with the original run of xaarfgen or aharfgen. However the differences in contribution fractions are not significant compared to setting xcen=ycen=CENTER. If in doubt, try both settings and assess the differences.
- telescop: Name of mission (XRISM/HITOMI).
- instrume: Name of the instrument (RESOLVE/SXS).
- teldeffile: The name of the telescope definition file. The default is teldeffile=CALDB, which sets the teldef file to the one stored in the CalDB.
- erange: The energy range (inclusive) in keV of simulated photons to use in traceback images. The energy range must overlap with the energy range input to xaarfgen or aharfgen; if the minimum energy requested by the user is less than that originally input to xaarfgen/aharfgen, then the minimum energy used by xmatraceback is the minimum energy from the raytracing file. Likewise, if the maximum energy requested by the user is greater than the maximum energy input to xaarfgen/aharfgen, then the maximum energy is set to the maximum value stored in the raytracing event file.
- outroot: A string that forms the root name of all output files.
- cleanup: Delete temporary files (yes/no). If you want to later make traceback images for custom regions that are not single pixels or the full detector array, you must set 'cleanup=no'.
- clobber: Overwrite existing files (yes/no).

4 Outputs

xmatraceback outputs the following files (consult the help file for full details on naming conventions):

- <outroot>_<instrume>_allpix_srcimg.fits: A full detector array traceback image (sans pixel 12). This contains all simulated photons that hit the detector.
- <outroot>_<instrume>_allpix_det.reg, <outroot>_<instrume>_allpix_radec.reg: Two full detector array region files, the former in DET coordinates, the latter in RA/DEC.
- <outroot>_<instrume>_pix<>_x<>y<>_srcimg.fits: 36 RA/DEC traceback images, one for each pixel, where pix<> is the pixel ID, and x and y specify the pixel's integer DETX and DETY coordinates respectively. Each traceback pixel image contains simulated photons that hit that pixel detector region. Pixel 12 is included for completeness, even though it is not an active region of the detector. Its position corresponds to the missing pixel in the 6x6 Resolve array, not to the physical location of the calibration pixel.
- <outroot>_<instrume>_pix<>_x<>y<>_det.reg, <outroot>_<instrume>_pix<>_radec.reg: Two region files for each pixel, the former in DET coordinates, the latter in RA/DEC.

- `<outroot>_photon_fractions.fits` (HEASoft 6.36 and later): A "photon fraction" FITS file that contains:
  - Extension 0, keyword `FREXTERN`: The fraction of signal in the full array that comes from outside of the full array.
  - Extension 1, column `PixFracExtern`: The fraction of the signal in each pixel that comes from outside the pixel.
  - Extension 0, keyword `FRINPIMG`: The ratio of the signal in the full array image to the (entire) image that was input into `xaarfgen` or `aharfgen`.
  - Extension 1: column `PixFracInpIMG`: The ratio of the signal in each individual pixel traceback image to the (entire) image that was input into `xaarfgen` or `aharfgen`.
- `full_array_mask.fits, pixel_<>_mask.fits`: A mask for the full array and for each pixel. The mask files are needed for traceback image analysis for custom regions involving an arbitrary combination of pixels (see Section 5.2). These masks are used to calculate the photon fractions that are later stored in the photon fraction FITS file. If you want to use these mask files you must run xmatraceback with 'cleanup=no', otherwise these files will be deleted when xmatraceback finishes running. Additionally, these file names do not include the `outroot` parameter, meaning these files will be overwritten if `xmatraceback` is re-run and these files have not been renamed to something unique.
- `pixel_<>_photons_outside_reg.fits`: Individual traceback images for each pixel with the corresponding pixel mask applied; that is, pixel images showing only photons that originate from outside the pixel detector region. These image files are used to calculate the photon fractions in `<outroot>_photon_fractions.fits`. These are considered temporary files, so if 'cleanup=yes', these masked pixel images will be deleted as `xmatraceback` finishes running. It should be also noted that these files do not include the `outroot` parameter in their name, meaning these files will be overwritten if `xmatraceback` is re-run in the same directory.

Note: all of the output files taken together can occupy more than a GB of disk space. If disk space is a concern, using gzip results in a substantial compression factor due to many sparsely populated images. Using a specific example, all of the files related to one run-through of `xmatraceback` with cleanup=no (including output files and the raytracing and heasim event files) uses up about 5.6 GB of disk space. After using `tar -czvf example_zip.tar.gz *` to compress all files, the resulting file example_zip.tar.gz is about 160 MB in size, which corresponds to a compression factor of ∼35.

4.1 Bug in xmatraceback in HEASoft 6.36

In the HEASoft 6.36 version of `xmatraceback`, there is bug that populates the keyword `FRINPIMG` and the column `PixFracInpIMG` (in the photon fractions output file) with incorrect values. A workaround is as follows:

1. Create a new raytracing file (say rtfile2.fits), from the input raytracing file (say rtfile1.fits), by selecting only events that have en energy in the range emin to emax inclusive (these energy bounds are the same as the inputs to your particular `xmatraceback` run):

```
ftcopy "rtfile1.fits[1][(ENERGY >=emin)&&(ENERGY <=emax)]" rtfile2.fits
```

2. Extract the value of `NAXIS2` from rtfile2.fits (extension 1), call this value B.

3. Extract the value of `NAXIS2` from the heasim event file (extension 1) that was input to your particular run of `xmatraceback`, call this value T.

4. Calculate the correction factor C = (T/B).

5. Multiply the keyword `FRINPIMP` by C to correct its value. Multiply values in the column `PixFracInpIMG` by C to correct their values.

5 Use in Analysis

5.1 Walk-through 1

Objective:

Use existing raytracing and heasim event files to create Hitomi SXS traceback images of the Perseus Cluster.

In some cases, you may have pre-existing raytracing and heasim event files from previous runs of `aharfgen`. In such cases, you can proceed to running `xmatraceback`.

Files needed to get started:

- Raytracing and heasim event files made using the same instance of `aharfgen` using Hitomi SXS. In this case, our raytracing file is “raytrace_perseus_sxs.fits” and our heasim event file is “heasim_events_perseus.fits”

Next, we run `xmatraceback` with the center of the detector (xcen=CENTER, ycen=CENTER) aligned with the origin of the raytracing coordinates over the energy range 0.5 to 17.1 keV.

xmatraceback command:

```
xmatraceback xrtevtfile=raytrace_perseus_sxs.fits
heasimevtfile=heasim_events_perseus.fits
xcen=CENTER ycen=CENTER telescop=HITOMI instrume=SXS teldeffile=
CALDB erange="0.5 17.1" outroot=traceback cleanup=yes
clobber=yes
```

This will output 37 traceback images– one for the full Resolve detector array and one for each pixel– as well as region files in DET and RA/DEC coordinates. An example of the full array traceback image can be seen in Figure 1.

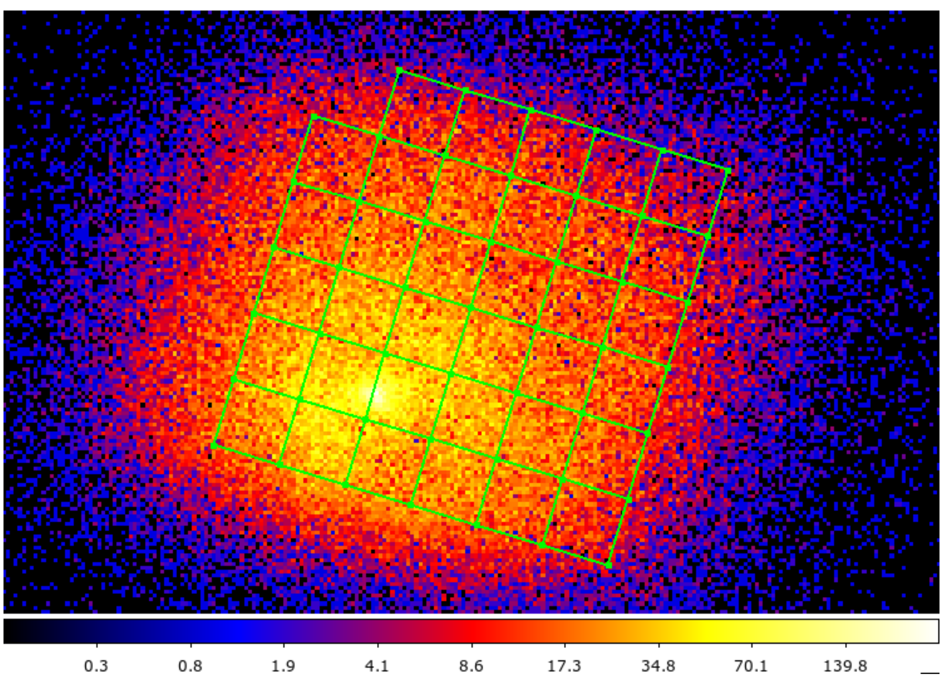


Figure 1: The traceback image of the Perseus Cluster made for the full Hitomi SXS detector array. The detector region in RA/DEC is overlaid in green.

A FITS file with various photon fractions will also be produced. One of the photon fractions, stored as the keyword `FREXTERN` in extension 0, relays the fraction of simulated photons that landed inside the full-array detector region which originated from outside the full array detector region. This fraction is calculated by applying a mask covering the full array detector region to the full array traceback image; the photons in this masked image are summed and then this number is divided by the total niumber of photons in the full array traceback image. In Figure 2, the full array image for the Perseus Cluster with the mask applied can be seen. Using this image, the full array photon fractional contribution from outside the detector is calculated to be 26.2%.

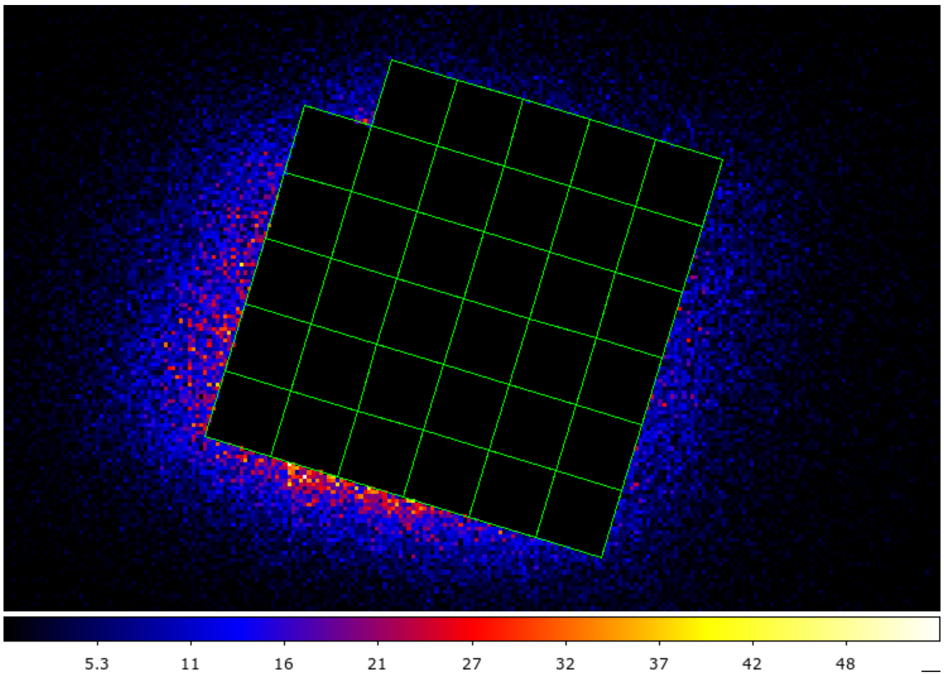
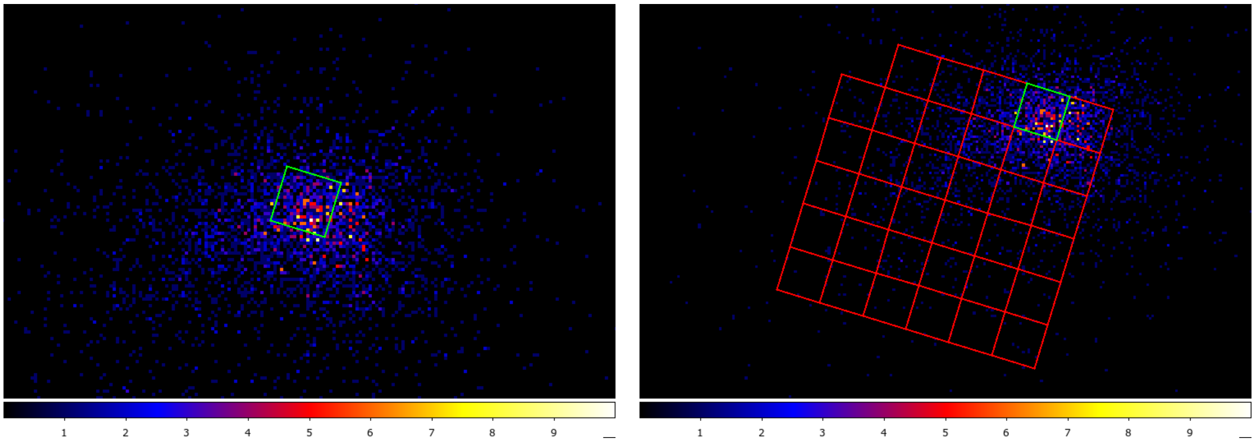


Figure 2: The full array traceback image of the Perseus Cluster with the mask applied for the entire detector region. This image is used to calculate the fraction of photons in the full-array that originates from outside the detector region. In this case, 26.2% of photons originate from outside the full array detector region.

The fraction of the total input image that contributes to the full array detector region (given by the `FRINPIMG` in extension 0) is 29.4% (Note that this is corrected for the xmatraceback bug in HEASoft 6.36). Fluxes derived from spectral fitting to spectra from the detector must be multiplied by this fraction. This is because the ARF made by `aharfgen` or `xaarfgen` in image mode corresponds to the whole input image..

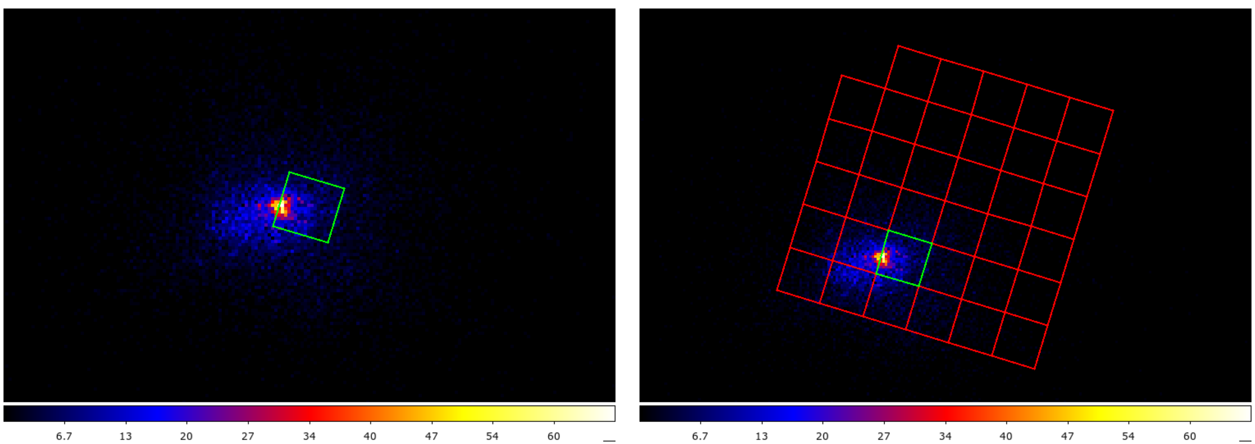
In Figures 3 and 4, examples of individual traceback images can be seen. Figure 3 depicts pixel 21, which in this case is from a dim region. Figure 3b shows the pixel 21 traceback image with the full array detector region overlaid in red for reference. From row 22 of the column `PixFracExtern` in the output file `traceback_photon_fractions.fits`, we find that for pixel 21, 85.15% of simulated photons originate from outside the pixel region for 21.

Figure 4 portrays pixel 2, which lies in a bright region of the source. Again, Figure 4b shows the pixel 2 traceback image with the full array detector region depicted in red and pixel 2 highlighted in green. Figures 4c and 4d are identical to images 4a and 4b respectively, but plotted on a log scale. From row 3 of the column `PixFracExtern` in the output file `traceback_photon_fractions.fits`, we find that for pixel 2, 79% of simulated photons come from outside the pixel 2 detector region.

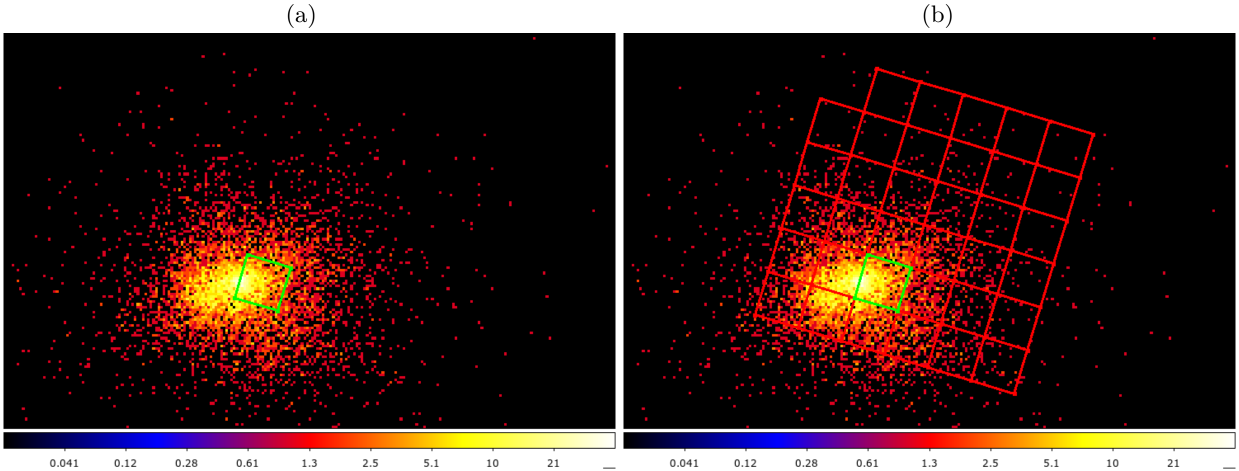


(a) (b)

Figure 3: a. Traceback image of pixel 21, plotted with a linear scale. b. Traceback image of pixel 21 with the detector region for the full detector region overlaid in red for reference, also plotted with a linear scale.



(a) (b)



(c) (d)

Figure 4: a. Traceback image of pixel 2, plotted with a linear scale. b. Traceback image of pixel 2 with the detector region for the full detector region overlaid in red for reference (plotted with a linear scale). c. Traceback image of pixel 2, but shown in log scale. d. Traceback image of pixel 2, plotted with a log scale and with the entire detector array overlaid for reference.

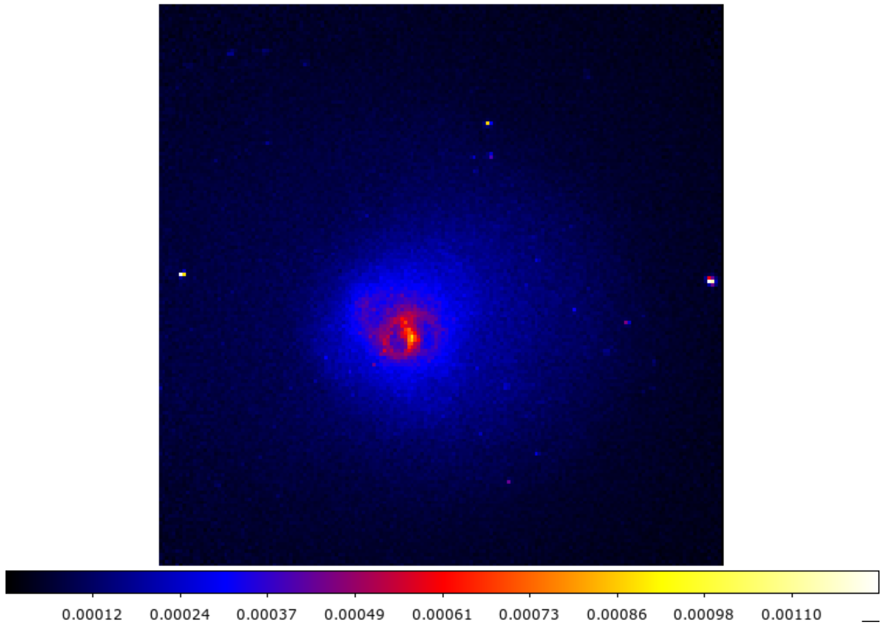


Figure 5: The original Chandra image of the Centaurus Cluster in the range 2-8 keV (file rateimage_rsl_2.0-8.0_bsub_expcorr_6bin_8minrect.img), used in "Walk-through 2."

5.2 Walk-through 2

Objective:

Run `xmatraceback` for a XRISM Resolve observation of the Centaurus Cluster, using a Chandra image as input to `xaarfgen`.

Files needed to get started:

- Image (real data) of the Centaurus Cluster using an observatory with higher spatial resolution, such as Chandra or XMM-Newton. In this example, we will use the 2-8 keV Chandra image file rateimage_rsl_2.0-8.0_bsub_expcorr_6bin_8minrect.img (see Figure 5).
- An RMF file for this source (see the Quick-Start Guide). In this example, the name of the RMF file that will be used is `centaurus_out2b_HP_S_det.rmf`
- An exposure map file. Here is an example of using `xaexpmap` to create an exposure map:
  - `xaexpmap ehkfile=xa000138000.ehk.gz gtifile="xa000138000rsl_p0px1000_cl.evt.gz[GTI]" instrume=Resolve badimgfile=NONE pixgtifile=xa000138000rsl_px1000_exp.gti.gz outfile=xa000138000rsl_p0px1000.expo outmaptype=EXPOSURE delta=20.0 numphi=1 maskcalsrc=yes`

After obtaining/producing all files necessary for the analysis, run `xaarfgen`. The file for the `pixgtifile` parameter can be found in the reprocessed data directory, or in **resolve/event_uf/**. `xaarfgen` will run the task `xrtraytrace` and the simulator task `heasim`, resulting in a raytracing event file and photon event list named `heasim_events.fits`. Here is an example of an `xaarfgen` command:

xaarfgen command:

```
xaarfgen xrtevtfile=raytracing_chandra_subimage_2to8_v1.fits
source_ra=192.205833 source_dec=-41.31125 telescop=XRISM instrume=RESOLVE
emapfile=xa000138000rsl_p0px1000.expo regmode=DET regionfile=region_RSL_det.reg
```

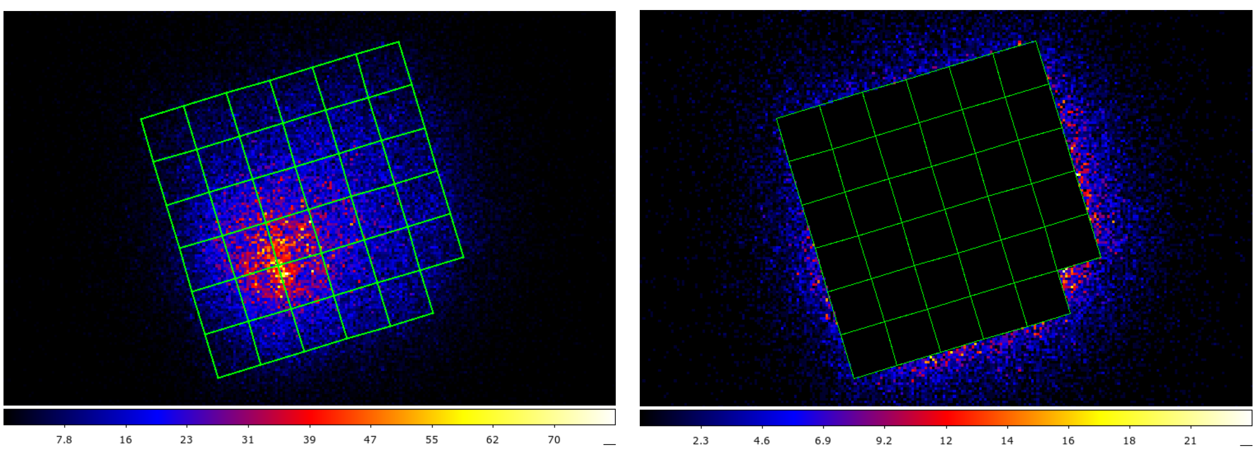
```
sourcetype=image imgfile=rateimage_rsl_2.0-8.0_bsub_expcorr_6bin_8minrect.img
rmffile=centaurus_out2b_HP_S_det.rmf.gz erange="0.5 17.3 2.0 8.0"
outfile=chandra_subimage_2to8.arf numphoton=300000 minphoton=1 mode=h
teldeffile=CALDB qefile=CALDB obffile=CALDB fwfile=CALDB
```

In this example, `source_ra` and `source_dec` correspond to the center of the cluster. Rename the output file `heasim_events.fits` to `centaurus_heasim_events.fits`. The raytracing file created in this run is `raytracing_chandra_subimage_2to8_v1.fits` and will be directly input into `xmatraceback`. Here we use the raytracing and heasim event files just generated, specifying xcen=3.5 and ycen=3.5 (the center of the detector– the default xcen=CENTER and ycen=CENTER would work just as well) and using the energy range 2-8 keV (inclusive) for the input image (but note that the ARF and raytracing file is made for a wider range, as shown).

xmatraceback command:

```
xmatraceback xrtevtfile=raytracing_chandra_subimage_2to8_v1.fits
heasimevtfile=centaurus_heasim_events.fits xcen=3.5 ycen=3.5
telescop=XRISM instrume=RESOLVE
teldeffile=CALDB erange="2.0 8.0"
outroot=traceback_centaurus cleanup=no clobber=yes
```

The `xmatraceback` tool will output traceback images for the full detector array and for each pixel. Each traceback image contains simulated photons that hit that detector region. Traceback images can be viewed with any tool that plots FITS images, e.g. ds9.

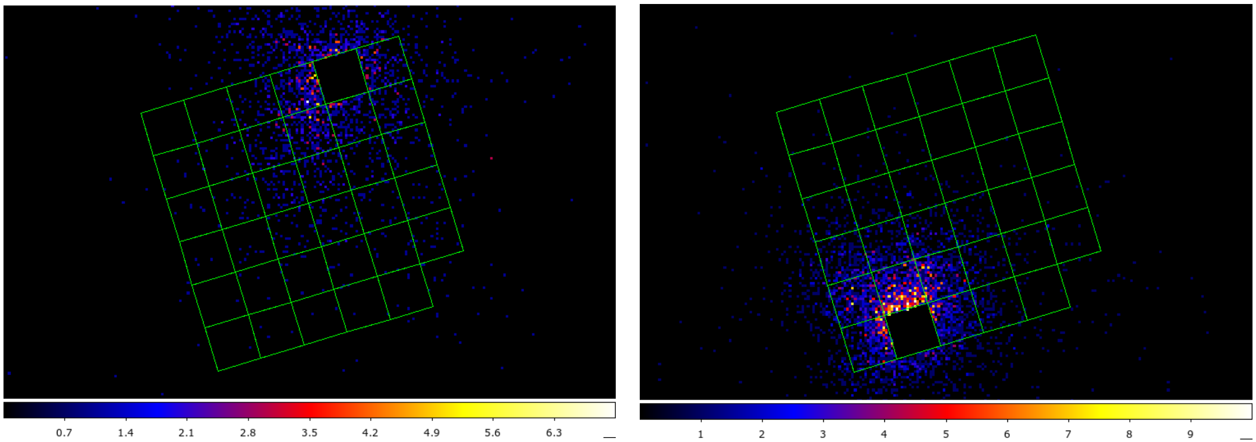


(a) (b)

Figure 6: a. Full array traceback image of the Centaurus cluster using XRISM Resolve and with the full region in RA/DEC overlaid in green. b. The same full array traceback image but with the mask for the full detector applied. Both images are shown on a linear scale.

In Figure 6, the full array traceback image (6a) and full array image with a mask applied (6b) can be seen. `xmatraceback` divides the number of photons in the masked image by the number of photons in the full-array unmasked `traceback` image and writes the result to the keyword `FREXTERN` in the extension 0 header of the output file centaurus_traceback_photon_fractions.fits. This number is the contribution to the full-array image that originates from outside the array and is 15.83% in this example.

Figure 7a shows the traceback image for pixel 3 with the mask applied. Pixel 3 is from a dim region and has a photon fraction– that is, fraction of photons originating from outside of pixel 3– of 80.9%. This fraction can be found in row 4 of the column `PixFracExtern` in the output file traceback_photon_fractions.fits. Figure 7b displays a traceback pixel image (with the pixel mask applied) from a brighter region. The pixel image is for pixel 21 and has a photon fraction of 75.5%, which can be found in row 22 of the column `PixFracExtern` in the output file traceback_photon_fractions.fits.



(a) (b)

Figure 7: a. Pixel traceback image for pixel 3, plotted with a linear scale, with the mask for pixel 3 applied. This shows a dimmer region of the source. b. The traceback image for pixel 21 with the pixel mask applied, also plotted with a linear scale. This pixel covers a brighter region of the source.

Custom region traceback images can be made by adding together individual pixel traceback images for an arbitrary combination of pixels. For custom region analysis, xmatraceback should have been run with 'cleanup=no'. For example, ftpixcalc can be used to sum pixel images for pixel 1 and 2 to create an image for this custom region:

```
ftpixcalc custom_region_1_2.fits A+B
a=centaurus_rsl_pix01_x6y3_srcimg.fits
b=centaurus_rsl_pix02_x5y3_srcimg.fits
```

The above can be extended to a larger number of input images, but `fimgmerge` may be more convenient (see help file for `fimgmerge` on usage).

The file `traceback_centaurus_photon_fractions.fits` (see Section 4) only contains contribution photon fractions for the full array and for single pixels. Photon fractions for custom regions can be calculated by applying masks to custom regions. Masks for custom images can be made from masks for the appropriate individual pixels. The latter are made by xmatraceback if 'cleanup=no'. Alternatively, you can make the individual pixel mask images using ftimgcalc. For example a mask for pixel 1 can be made by running:

```
ftimgcalc pix_1_mask.fits
'regfilter("caldb_check_rsl_pix01_radec.reg", a.P1, a.P2) ? (0):
(1)' a=centaurus_rsl_pix01_x6y3_srcimg.fits
```

This mask can be used to extract regions of a traceback image that are outside of pixel 1. To change the command to make a mask for extracting the image region inside pixel 1, switch (0):(1) to (1):(0).

Individual masks can be merged together using `fmrgmsk` to create a larger mask for the custom region. Here we make a custom mask combining the pixel masks of pixels 1 and 2:

```
fmrgmsk
pix_1_mask.fits pix_2_mask.fits
custom_region_mask_1_2.fits ops=AND
```

Masks can be applied to custom regions using ftimgcalc. The following example applies the mask image `custom_region_mask_1_2.fits` to the custom traceback image `custom_region_1_2.fits`:

```
ftimgcalc photons_outside_1_2_reg.fits 'a*b'
```

```
a="custom_region_1_2.fits" b="custom_region_mask_1_2.fits"
```

The resulting `photons_outside_reg_1_2.fits` file will contain simulated photons that hit your custom region (that is, they hit either pixel 1 or 2) and that originate from outside the mask. To find the photon fraction from this two-pixel combination, you would first sum the photons in `photons_outside_reg_1_2.fits`. You would then take this sum and divide it by the total number of photons in the custom image `custom_region_1_2.fits`. This is the image you made previously that combines pixels 1 and 2. In Figure 8, the image from `photons_outside_reg_1_2.fits` is shown. This image contains the combined custom traceback image for pixels 1 and 2 with the masks for pixels 1 and 2 overlaid.

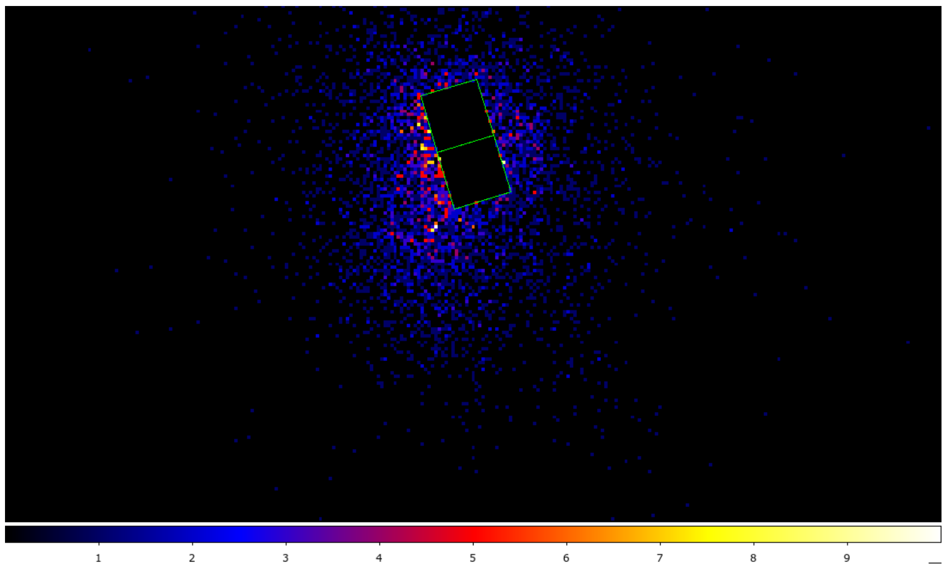


Figure 8: The custom traceback image combining the traceback images of pixels 1 and 2 (linear scale applied) with the custom mask for pixels 1 and 2 overlaid. This image would be used to calculate the fraction of photons in the 2-pixel custom region that originates from outside of the 2-pixel region.