How to work with XMM-Newton?

Dmytro Iakubovskyi*

I. HOW TO START?

To start working with *XMM-Newton*, please, open the terminal and go to your working direktory: `cd /home/XMM_DATA/guest/cenx3/`. You will further asked to work in the C shell, so please type `csh`. After that, you need to configure the parameters correctly. To do that, print `source /virgo/scripts/login` This will end your preparation.

II. DATA PREPATION

In your local directory you can find a subdirectory, called `0111010101` (you can see it when typing `ls`). This is the direktory with XMM-Newton data. It contains a direktory `ODF`, which mean Observation Data Files - raw data files, which are usually distributed by XMM-Newton Science Data Centre. Note that there are just instrumental files. To obtain the files of scientific interest (the so-called event lists - lists of events, with coordinate, time and energy of each events listed. Event lists are useful to produce the major scientifically interesting rezults - images, lightcurves, spectra), one needs to run several well-developed routines in his local computer. Below is one of the pipelines.

1. Firstly, you need to choose the data directory: `cd 0111010101`.

2. To adjust the necessary calibration files, you need to run the procedure `cifbuild`, which does it for you, just typing `cifbuild`.

3. The procedure `cifbuild` creates the file with the name `ccf.cif`. You need to mention this file for the next operations: `setenv SAS_CCF ccf.cif`

4. Now you need to adjust the header information for your ODF files. This can be done by typing `odfingest`.

5. The result of `odfingest` is the header file with the name `*SUM.SAS`, where "*" means something, which is not important for us. However, we need to teach our software the name

---

*Electronic address: yakubovskiy@bitp.kiev.ua

of this file: `setenv SAS_ODF *SUM.SAS`[1]

6. Now we can really procceed for obtaining the event lists. It usually takes several minutes, so type `emproc; epproc` and have a rest for a while :-) You've gained it!

III. PRODUCING RAW IMAGES AND LIGHTCURVES FROM AN IMAGING MODE.

After `em(p)proc` completes, you will get a number of event lists, which you can see by typing `ls *Evts.ds` The result of the execution should be the following list of 4 files:

```
0208_0111010101_EMOS1_S001_ImagingEvts.ds
0208_0111010101_EMOS1_S001_TimingEvts.ds
0208_0111010101_EMOS2_S002_ImagingEvts.ds
0208_0111010101_EPN_S003_ImagingEvts.ds
```

First 3-digit number is the number of *XMM-Newton* revolution. Each revolution lasts approximately three days, *XMM-Newton* was launched at the end of 1999, so you can roughly estimate the observation time, which is January 2001[2] Second number is the number of observation, followed by the name of the camera and the index of the pointing. There are three EPIC (European Photon Imaging Cameras) inside *XMM-Newton*. MOS1 and MOS2 cameras are similar to each other, PN is somewhat different, so we need quite different algorithms for MOS and PN to proceed the data reduction. Index "Imaging"/"Timing" represents a working mode, and the algorithms of reducing the data seem to be very different in these two modes. Below we discuss how to do the analysis in each of those cases.

---

[1] You do not need to type all the stuff which "*" means, just type "*", and Linux understands you :-)

[2] Of cource, you can derive it in a more robust way, typing, e.g., `fkeyprint 0208_0111010101_EMOS1_S001_ImagingEvts.ds+1 DATE-OBS`.

A. MOS/PN, Imaging mode.

1. Images.

In the Imaging mode the data are collected using two spatial coordinates, so it is possible to recover a 2D image. Let's do that by typing

```
cp 0208_0111010101_EMOS1_S001_ImagingEvts.ds mos1.fits
cp 0208_0111010101_EMOS2_S002_ImagingEvts.ds mos2.fits
cp 0208_0111010101_EPN_S003_ImagingEvts.ds pn.fits
evselect table=mos1.fits:EVENTS withimageset=yes imageset=mos1-sky.im
xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
evselect table=mos2.fits:EVENTS withimageset=yes imageset=mos2-sky.im
xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
evselect table=pn.fits:EVENTS withimageset=yes imageset=pn-sky.im xcolumn=X
ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
ds9 mos1-sky.im mos2-sky.im pn-sky.im &
```

The obtained files `mos1-sky.im, mos2-sky.im` are the *unfiltered XMM-Newton* images of Cen X-3 binary and nearby region[3]. However, they are good only for a first look. The more thorough look with, e.g., `ds9` will show that the outer parts of the pictures are fainter, mostly because of the fact that they do not correspond to open FoV. That's why we should remove the photons from such regions. There is a number of CCD gaps, bright pixels etc, which we also need to remove. The photons far outside from *XMM-Newton* energy range can distort the picture; that's why we also need to remove them. The corresponding filtering expressions are:

```
evselect table=mos1.fits:EVENTS withfilteredset=yes expression='(PATTERN
<= 12)&&#XMMEA_EM&&(PI in [200:12000])' filteredset=mos1-filt.fits
filtertype=expression keepfilteroutput=yes
evselect table=mos2.fits:EVENTS withfilteredset=yes expression='(PATTERN
<= 12)&&#XMMEA_EM&&(PI in [200:12000])' filteredset=mos2-filt.fits
filtertype=expression keepfilteroutput=yes
evselect table=pn.fits:EVENTS withfilteredset=yes expression='(PATTERN
```

---

[3] The field of view (FoV) of *XMM-Newton* EPIC cameras has $\simeq 15$ arcmin in radius

```
<= 0)&&#XMMEA_EP&&(PI in [150:15000])' filteredset=pn-filt.fits
filtertype=expression keepfilteroutput=yes
evselect table=mos1-filt.fits:EVENTS withimageset=yes
imageset=mos1-filt-im.fits xcolumn=X ycolumn=Y imagebinning=imageSize
ximagesize=600 yimagesize=600
evselect table=mos2-filt.fits:EVENTS withimageset=yes
imageset=mos2-filt-im.fits xcolumn=X ycolumn=Y imagebinning=imageSize
ximagesize=600 yimagesize=600
evselect table=pn-filt.fits:EVENTS withimageset=yes
imageset=pn-filt-im.fits xcolumn=X ycolumn=Y imagebinning=imageSize
ximagesize=600 yimagesize=600
```

You can clearly see the distinction between the filtered and unfiltered images. We should admit that the applied filtering is rather coarse. For instance, it does not take into account any background rejection, which worsen the image quality. In order to obtain the high-quality image of an extended object, which one should use for a quantitative research, the exposure correction should also be done. However, as our primary goal is to study the central object, Cen X-3, which is bright point source[4], we do not consider this issue here.

2. Lightcurves

Because the Cen X-3 is variable, it is interesting to produce and analyze its lightcurve. It can be done by using the raw-filtered event lists:

```
evselect table=mos1-filt.fits:EVENTS withrateset=yes rateset=mos1-filt.lc
maketimecolumn=yes makeratecolumn=yes timebinsize=50
evselect table=mos2-filt.fits:EVENTS withrateset=yes rateset=mos2-filt.lc
maketimecolumn=yes makeratecolumn=yes timebinsize=50
evselect table=pn-filt.fits:EVENTS withrateset=yes rateset=pn-filt.lc
maketimecolumn=yes makeratecolumn=yes timebinsize=50
fv mos1-filt.lc mos2-filt.lc pn-filt.lc &
```

---

[4] Actually, it doesn't look like a *point*. This is because of the finite extent of so-called point-spread function, PSF.

FIG. 1: MOS1 (black), MOS2 (red) and PN (blue) lightcurves, obtained from raw-filtered event lists. Note clear distinction between the behaviour of MOS2 and PN lightcurves, which is due to differences on their effective areas and quiscent background rates.

FIG. 2: The same as in the previous figure, but the lightcurves are produced from the central region, so the background count rate is highly suppressed.

Since we do not have Cen X-3 in MOS1 observation, its lightcurve approximately represents the temporal behaviour of the background. It is shown that the background has two components with a different temporal behaviour - the *quiscent* and *flaring* components. Let's have a closer look:

```
evselect table=mos1-filt.fits:EVENTS withrateset=yes
rateset=mos1-filt_200.lc maketimecolumn=yes makeratecolumn=yes
timebinsize=200
```

, where we decide to increase time binning interval to 200 sec, which helps to reduce the statistical errors. The flaring component is mostly due to soft protons from the Sun, which scatter through the mirrors and interact with the CCD camera. It usually has a very hard spectra, so it can be easily distinguished at high energies (10-12 keVfor MOS and 12-15 keVfor PN camera). After that one should filter out all intervals of high (flaring) background, using `(TIME in [T_min:T_max])` selection expression.

It is important to know the number of counts in the event list. You can calculate it by typing, e.g. `fstatistic mos1.fits TIME -`. The last column of `fstatistic` output represents the number of counts.

The raw MOS and PN lightcurves are shown in Fig. 1. You can see the different ratios of MOS2 and PN count rates during "low" and "high" states. to check whether it is the artifact of background rate, we prepare the lightcurves from the central region of Cen X-3, with minor background contamination. The difference on count rates remains, so we consider this issue below in details.

B. Production of MOS2 spectra

To produce a MOS2 spectrum, several procedures should be done. First of all, we need to select all events from the source, which can be done by `evselect`. Then we need to quantify what is the background spectrum is, and subtract it. To do that, we need to select background region, away from the source (however, to avoid a difference in electronic noise component, it is plausible to take background from the same CCD), and select the background spectrum with `evselect`.

To properly subtract the background, we need to rescale it using the solid angle of source region. More precisely, we need to calculate all "good" pixels in source and background regions, excluding bad pixels, hot pixels, CCD gaps etc. This is done by procedure `backscale`.

Because the value of "energy" (PI) is roughly corresponds to a particle energy (e.g., because of particle redistribution), we should calculate a matrix, which transforms PI value into "proper energy", or PHA (pulse height amplitude), with the help of rather time-consuming procedure, `rmfgen`.

To obtain physical fluxes, we need to adopt a correct value of effective area, using `arfgen`. In this procedure there several important corrections are also calculated. For instance, the finite size of our source region (in comparison with point-spread function extent) means that the part of photons from the source is not in the source region, so that `arfgen` models PSF and calculate the correction factor. Other important effect is vignetting, which means that the effective area decreases with off-axis angle. The important correction for PN camera is due to out-of-time events, which we will discuss further.

Finally, we will group the results - source and background spectra - with corresponding responce files and rebin them (to ensure gaussian statistics) with the help of FTOOL `grppha`.

The final set of commands to obtain a total MOS2 spectrum fror Cen X-3 is:

```
 evselect energycolumn=PI specchannelmax=11999 specchannelmin=0
spectralbinsize=15 spectrumset=mos2-source.pi table=mos2-filt.fits:EVENTS
expression='((X,Y) in circle(26361.5,23882,720))' updateexposure=yes
withspecranges=yes withspectrumset=yes
backscale spectrumset=mos2-source.pi badpixlocation='mos2-filt.fits'
withbadpixcorr=yes useodfatt=no
evselect energycolumn=PI specchannelmax=11999 specchannelmin=0
```

```
spectralbinsize=15 spectrumset=mos2-back.pi table=mos2-filt.fits:EVENTS
expression='!((X,Y) in circle(26361.5,23882,1500))&&((X,Y) in
circle(26361.5,23882,3000))' updateexposure=yes withspecranges=yes
withspectrumset=yes
backscale spectrumset=mos2-back.pi badpixlocation='mos2-filt.fits'
withbadpixcorr=yes useodfatt=no
rmfgen format='var' rmfset='mos2-source.rmf' spectrumset='mos2-source.pi'
threshold=1.e-6
arfgen arfset='mos2-source.arf' extendedsource=no modelee=yes psfenergy=5
rmfset='mos2-source.rmf' spectrumset='mos2-source.pi' withrmfset=yes
withbadpixcorr=yes badpixlocation=mos2-filt.fits modelootcorr=yes
grppha mos2-source.pi mos2-source.grp comm='chkey RESPFILE mos2-source.rmf
& chkey ANCRFILE mos2-source.arf & chkey BACKFILE mos2-back.pi & group 50 &
exit' clobber=yes
```

C. PN spectrum

Getting spectrum for PN camera is very similar to that in MOS camera. Only `evselect` parameters `specchannelmax` and `spectralbinsize` should be changed (for PN camera they are equal to 20479 and 5, respectively).

IV. ANALYSIS FOR TIMING MODE - THE SAMPLE FROM MOS1 CAMERA.

A. Pile-up

All previous images and spectra are produced for Imaging Mode. However, for bright sources and high-fluxes Imaging Mode can be unefficient. For instrance, there can be the situation, where two photons hit the CCD so close with each other, so that the secondary electrons appear in the same pixel during the CCD integration time. This means the photons will record as ONE photon, with bigger energy. This effect is calling by “pile-up”; it leads to spectral hardening and flux reduction for bright sources.

There are several possibilities to avoid pile-up. First is to throw away the central part of the source, which is mostly affected by pile-up, and reconstruct central flux by PSF

FIG. 3: The MOS1, MOS2 and PN lightcurves of Cen X-3. The difference between the shapes of MOS2 lightcurves, on the one hand, and MOS1/PN, on another, is clearly seen

modelling. The second way is to reduce the CCD integration time, which can be done by reducing the amount of information, which we need to tranfer to Earth.

The second approach is realized in EPIC (MOS and PN) Timing modes. In this mode we collect only one spatial coordinate about the source, making one-dimensional picture of the object. This approach enhances the amount of irreducible background (which is OK for bright sources), and does not allow to obtain the exact coordinates of the source (which, however, is possible in Imaging Mode). The gain is that we can obtain the spectrum, which is not affected with pile-up. Below we will describe the data analysis in Timing Mode.

B. Building "image histograms" and lightcurves in Timing Mode

Strictly speaking, there is no images in Timing Mode. However, to proceed background subtraction, create source lightcurve and spectrum one needs to know the 1-D spatial distribution of events. To do this, we create a histogram:

```
evselect
table=mos1-ti-filt.fits:EVENTS withhistogramset=yes histogramcolumn=RAWX
histrogramset=mos1-ti-filt.hist
```

From this histrogram we show that the source region has $300 \leq RAWX \leq 320$, and we can take a background region from $260 \leq RAWX \leq 280$. Then we build source lightcurve:

```
evselect table=mos1-ti-filt.fits:EVENTS withrateset=yes
rateset=mos1-ti-filt-cen.lc maketimecolumn=yes makeratecolumn=yes
timebinsize=200 expression='(RAWX in [300:320])'
```

C. Comparison between MOS1, MOS2 and PN lightcurves. Pile-up. Fixing pile-up

The comparison of MOS1, MOS2 and PN lightcurves clearly shows the subtantial difference of MOS2 lightcurve at high countrates.

Because MOS2 camera was observing Cen X-3 in Full Window Mode[5], there is a concern

---

[5] we can check it by typing `fkeyprint mos2-filt.fits+1 SUBMOD | tail -1`

FIG. 4: The MOS1, MOS2 (within 20-50 arcsec) and PN lightcurves of Cen X-3. The difference between the shapes of lightcurves, which was due to pile-up, now completely disappears

about pile-up. Let us check the pile-up level via `epatplot`:

```
epatplot set=mos2-filt.fits
```

The single and double events do not match the simulated curves, which indicates a presence of pile-up. The possible way of fixing pile-up is to throw away bright central pixels, which are mostly affected by pile-up:

```
evselect table=mos2-filt.fits:EVENTS withfilteredset=yes
filteredset=mos2-filt-400_1000.fits expression='((X,Y) in
circle(26361.5,23882,1000))&&!((X,Y) in circle(26361.5,23882,400))'
epatplot set=mos2-filt-400_1000.fits -V 5
```

The important quantities to check are the observed-to-model fractions of single and double particles, which SHOULD be 1 within errors. The figure ?? already shows the effective pile-up correction.

Therefore, pile-up can be easily corrected, and the correct shape of lightcurve (as well as of spectrum) can be reconstructed.

D. Building spectra in Timing Mode

Now we are ready to build the MOS1 spectrum:

```
arfgen arfset=’mos1-source.arf’ extendedsource=no modelee=yes psfenergy=5
rmfset=’mos1-source.rmf’ spectrumset=’mos1-source.pi’ withrmfset=yes withbadpixcorr=yes badpixlocation=mos1-pi-filt.fits modelootcorr=yes
grppha mos1-source.pi mos1-source.grp comm=’chkey RESPFILE mos1-source.rmf
& chkey ANCRFILE mos1-source.arf & chkey BACKFILE mos1-back.pi & group
50 & exit’ clobber=yes
evselect energycolumn=PI specchannelmax=11999
specchannelmin=0 spectralbinsize=15 spectrumset=mos1-source.pi
table=mos1-ti-filt.fits:EVENTS expression='(RAWX in [300:320])&&(PATTERN
<=12)' updateexposure=yes withspecranges=yes withspectrumset=yes
backscale spectrumset=mos1-source.pi badpixlocation='mos1-ti-filt.fits'
```

```
withbadpixcorr=yes useodfatt=no
evselect energycolumn=PI specchannelmax=11999 specchannelmin=0
spectralbinsize=15 spectrumset=mos1-back.pi table=mos1-pi-filt.fits:EVENTS
expression=’(RAWX in [260:280])&&(PATTERN <=12)’ updateexposure=yes
withspecranges=yes withspectrumset=yes
backscale spectrumset=mos1-back.pi badpixlocation=’mos1-ti-filt.fits’
withbadpixcorr=yes useodfatt=no
```

```
  arfgen arfset=’mos1-source.arf’ extendedsource=no modelee=yes psfenergy=5
rmfset=’mos1-source.rmf’ spectrumset=’mos1-source.pi’ withrmfset=yes
withbadpixcorr=yes badpixlocation=mos1-pi-filt.fits modelootcorr=yes
grppha mos1-source.pi mos1-source.grp comm=’chkey RESPFILE mos1-source.rmf
& chkey ANCRFILE mos1-source.arf & chkey BACKFILE mos1-back.pi & group 50 &
exit’ clobber=yes
```

To obtain model parameters, you need to fit these spectra in Xspec.