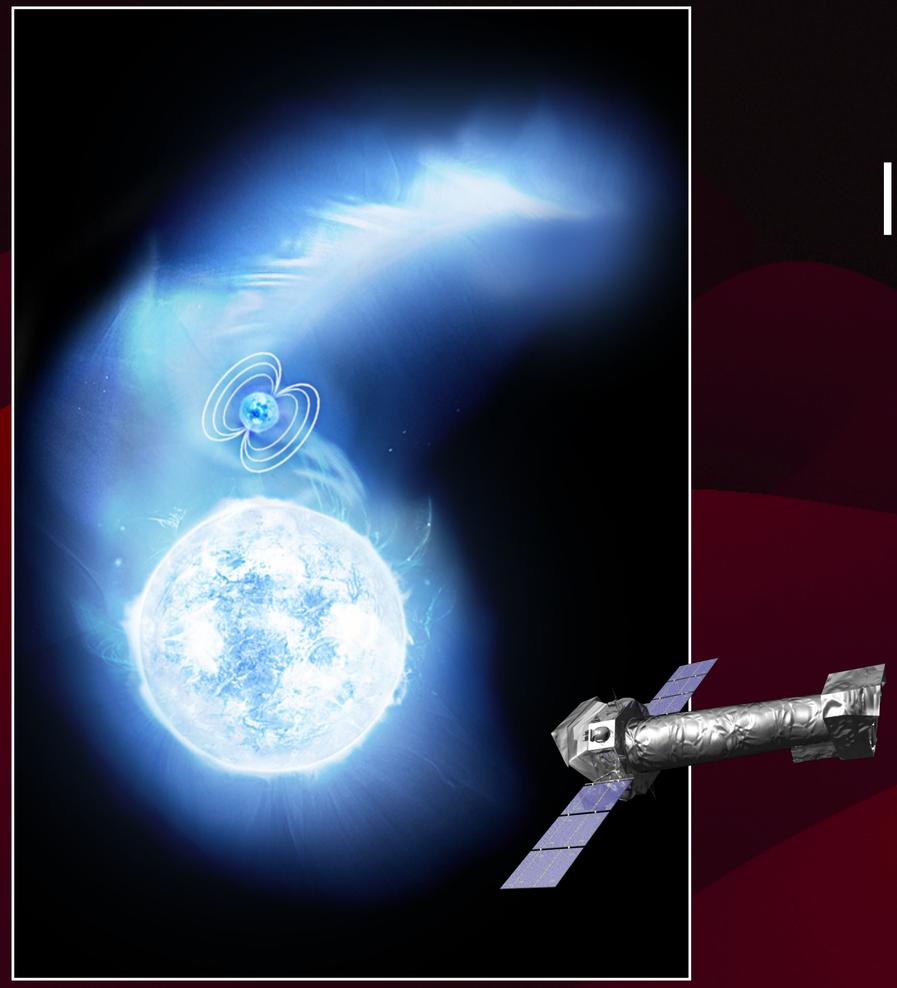


Introduction to SAS in Datalabs & EPIC+RGS data extraction

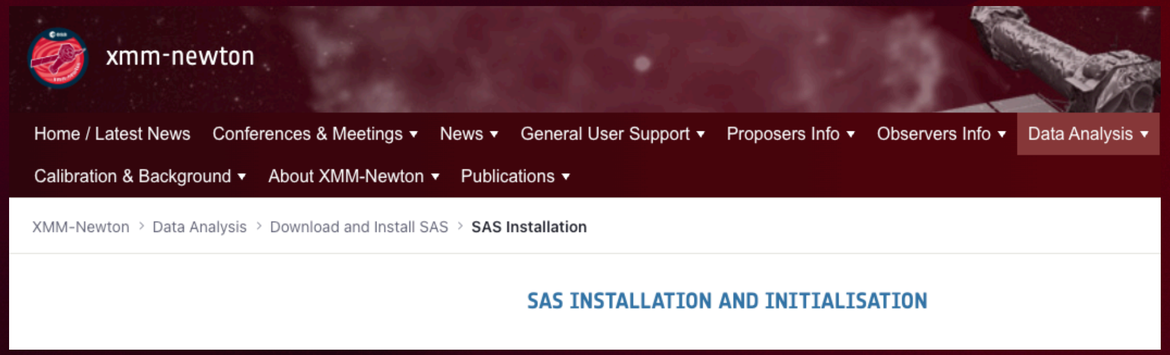
Camille Diez (ESAC, Spain), ACME Swift/XMM Workshop, 9 Oct. 2025

My journey as an XMM user: From bash to Datalabs

+ installation of other dependencies

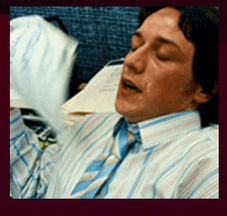
My journey as an XMM user: From bash to Datalabs

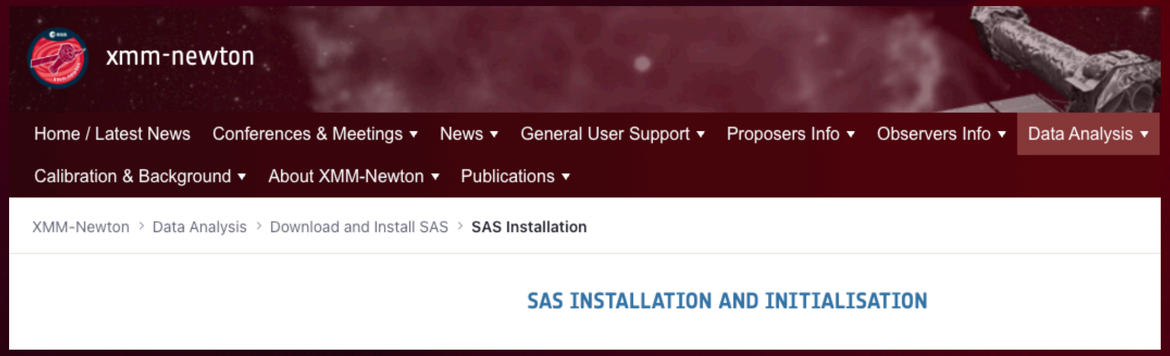
+ installation of other dependencies



My journey as an XMM user: From bash to Datalabs

+ installation of other dependencies





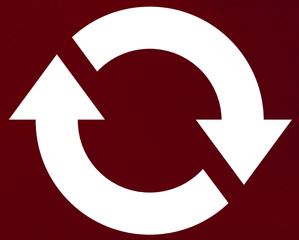
version	build identifier	distribution
22.1.0	xmmsas_22.1.0-a8f2c2afa-20250304	public
22.0.0	xmmsas_22.0.0-9173c7d25-20250128	public
21.0.0	xmmsas_20230412_1735	public
20.0.0	xmmsas_20211130_0941	public

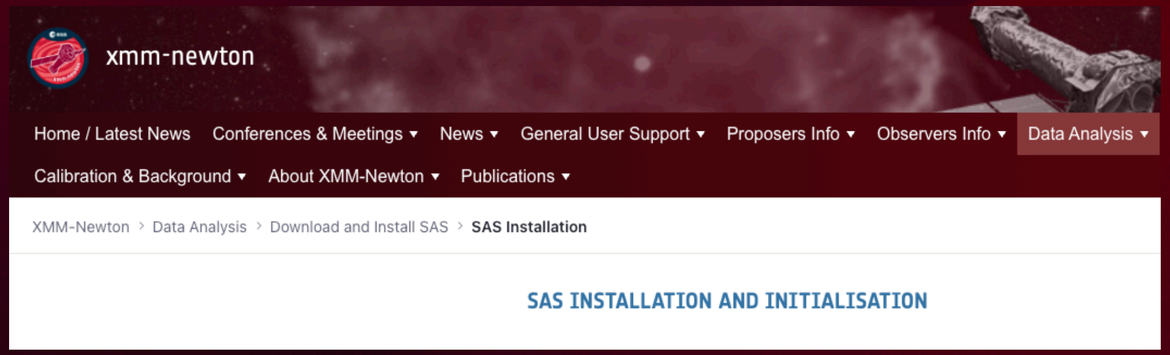
My journey as an XMM user: From bash to Datalabs

Installing HEASoft on a Linux

+ installation of other dependencies

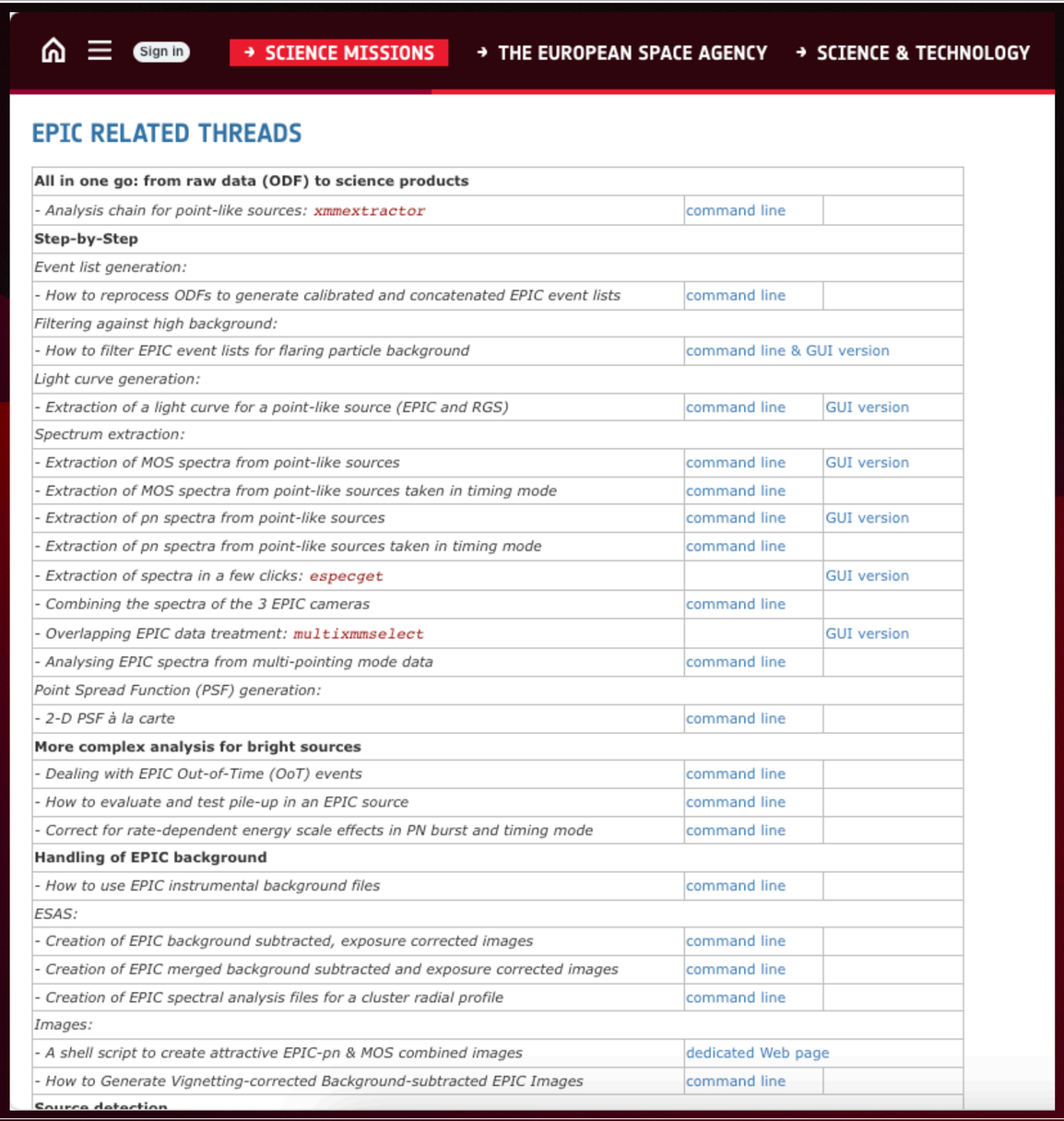
```
Installation successful !
```





version	build identifier	distribution
22.1.0	xmmsas_22.1.0-a8f2c2afa-20250304	public
22.0.0	xmmsas_22.0.0-9173c7d25-20250128	public
21.0.0	xmmsas_20230412_1735	public
20.0.0	xmmsas_20211130_0941	public

My journey as an XMM user: From bash to Datalabs



Sign in → SCIENCE MISSIONS → THE EUROPEAN SPACE AGENCY → SCIENCE & TECHNOLOGY

EPIC RELATED THREADS

All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
Event list generation:		
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists	command line	
Filtering against high background:		
- How to filter EPIC event lists for flaring particle background	command line & GUI version	
Light curve generation:		
- Extraction of a light curve for a point-like source (EPIC and RGS)	command line	GUI version
Spectrum extraction:		
- Extraction of MOS spectra from point-like sources	command line	GUI version
- Extraction of MOS spectra from point-like sources taken in timing mode	command line	
- Extraction of pn spectra from point-like sources	command line	GUI version
- Extraction of pn spectra from point-like sources taken in timing mode	command line	
- Extraction of spectra in a few clicks: `especget`		GUI version
- Combining the spectra of the 3 EPIC cameras	command line	
- Overlapping EPIC data treatment: `multixmmselect`		GUI version
- Analysing EPIC spectra from multi-pointing mode data	command line	
Point Spread Function (PSF) generation:		
- 2-D PSF à la carte	command line	
More complex analysis for bright sources		
- Dealing with EPIC Out-of-Time (OoT) events	command line	
- How to evaluate and test pile-up in an EPIC source	command line	
- Correct for rate-dependent energy scale effects in PN burst and timing mode	command line	
Handling of EPIC background		
- How to use EPIC instrumental background files	command line	
ESAS:		
- Creation of EPIC background subtracted, exposure corrected images	command line	
- Creation of EPIC merged background subtracted and exposure corrected images	command line	
- Creation of EPIC spectral analysis files for a cluster radial profile	command line	
Images:		
- A shell script to create attractive EPIC-pn & MOS combined images	dedicated Web page	
- How to Generate Vignetting-corrected Background-subtracted EPIC Images	command line	
Source detection		

My journey as an XMM user: From bash to Datalabs

Sign in → SCIENCE MISSIONS → THE EUROPEAN SPACE AGENCY → SCIENCE & TECHNOLOGY

EPIC RELATED THREADS

All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
Event list generation:		
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists	command line	
Filtering against high background:		
- How to filter EPIC event lists for flaring particle background	command line & GUI version	
Light curve generation:		
- Extraction of a light curve for a point-like source (EPIC and RGS)	command line	GUI version
Spectrum extraction:		
- Extraction of MOS spectra from point-like sources	command line	GUI version
- Extraction of MOS spectra from point-like sources taken in timing mode	command line	
- Extraction of pn spectra from point-like sources	command line	GUI version
- Extraction of pn spectra from point-like sources taken in timing mode	command line	
- Extraction of spectra in a few clicks: `especget`		GUI version
- Combining the spectra of the 3 EPIC cameras	command line	
- Overlapping EPIC data treatment: `multixmmselect`		GUI version
- Analysing EPIC spectra from multi-pointing mode data	command line	
Point Spread Function (PSF) generation:		
- 2-D PSF à la carte	command line	
More complex analysis for bright sources		
- Dealing with EPIC Out-of-Time (OoT) events	command line	
- How to evaluate and test pile-up in an EPIC source	command line	
- Correct for rate-dependent energy scale effects in PN burst and timing mode	command line	
Handling of EPIC background		
- How to use EPIC instrumental background files	command line	
ESAS:		
- Creation of EPIC background subtracted, exposure corrected images	command line	
- Creation of EPIC merged background subtracted and exposure corrected images	command line	
- Creation of EPIC spectral analysis files for a cluster radial profile	command line	
Images:		
- A shell script to create attractive EPIC-pn & MOS combined images	dedicated Web page	
- How to Generate Vignetting-corrected Background-subtracted EPIC Images	command line	
Source detection		

My journey as an XMM user: From bash to Datalabs

Sign in → SCIENCE MISSIONS → THE EUROPEAN SPACE AGENCY → SCIENCE & TECHNOLOGY

EPIC RELATED THREADS

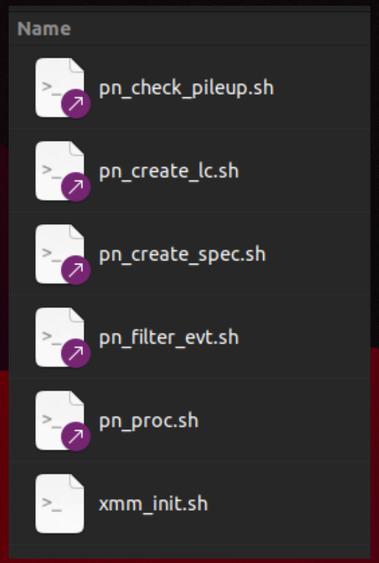
All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
Event list generation:		
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists	command line	
Filtering against high background:		
- How to filter EPIC event lists for flaring particle background	command line & GUI version	
Light curve generation:		
- Extraction of a light curve for a point-like source (EPIC and RGS)	command line	GUI version
Spectrum extraction:		
- Extraction of MOS spectra from point-like sources	command line	GUI version
- Extraction of MOS spectra from point-like sources taken in timing mode	command line	
- Extraction of pn spectra from point-like sources	command line	GUI version
- Extraction of pn spectra from point-like sources taken in timing mode	command line	
- Extraction of spectra in a few clicks: `especget`		GUI version
- Combining the spectra of the 3 EPIC cameras	command line	
- Overlapping EPIC data treatment: `multixmmselect`		GUI version
- Analysing EPIC spectra from multi-pointing mode data	command line	
Point Spread Function (PSF) generation:		
- 2-D PSF à la carte	command line	
More complex analysis for bright sources		
- Dealing with EPIC Out-of-Time (OoT) events	command line	
- How to evaluate and test pile-up in an EPIC source	command line	
- Correct for rate-dependent energy scale effects in PN burst and timing mode	command line	
Handling of EPIC background		
- How to use EPIC instrumental background files	command line	
ESAS:		
- Creation of EPIC background subtracted, exposure corrected images	command line	
- Creation of EPIC merged background subtracted and exposure corrected images	command line	
- Creation of EPIC spectral analysis files for a cluster radial profile	command line	
Images:		
- A shell script to create attractive EPIC-pn & MOS combined images	dedicated Web page	
- How to Generate Vignetting-corrected Background-subtracted EPIC Images	command line	
Source detection		

My journey as an XMM user: From bash to Datalabs

Sign in → SCIENCE MISSIONS → THE EUROPEAN SPACE AGENCY → SCIENCE & TECHNOLOGY

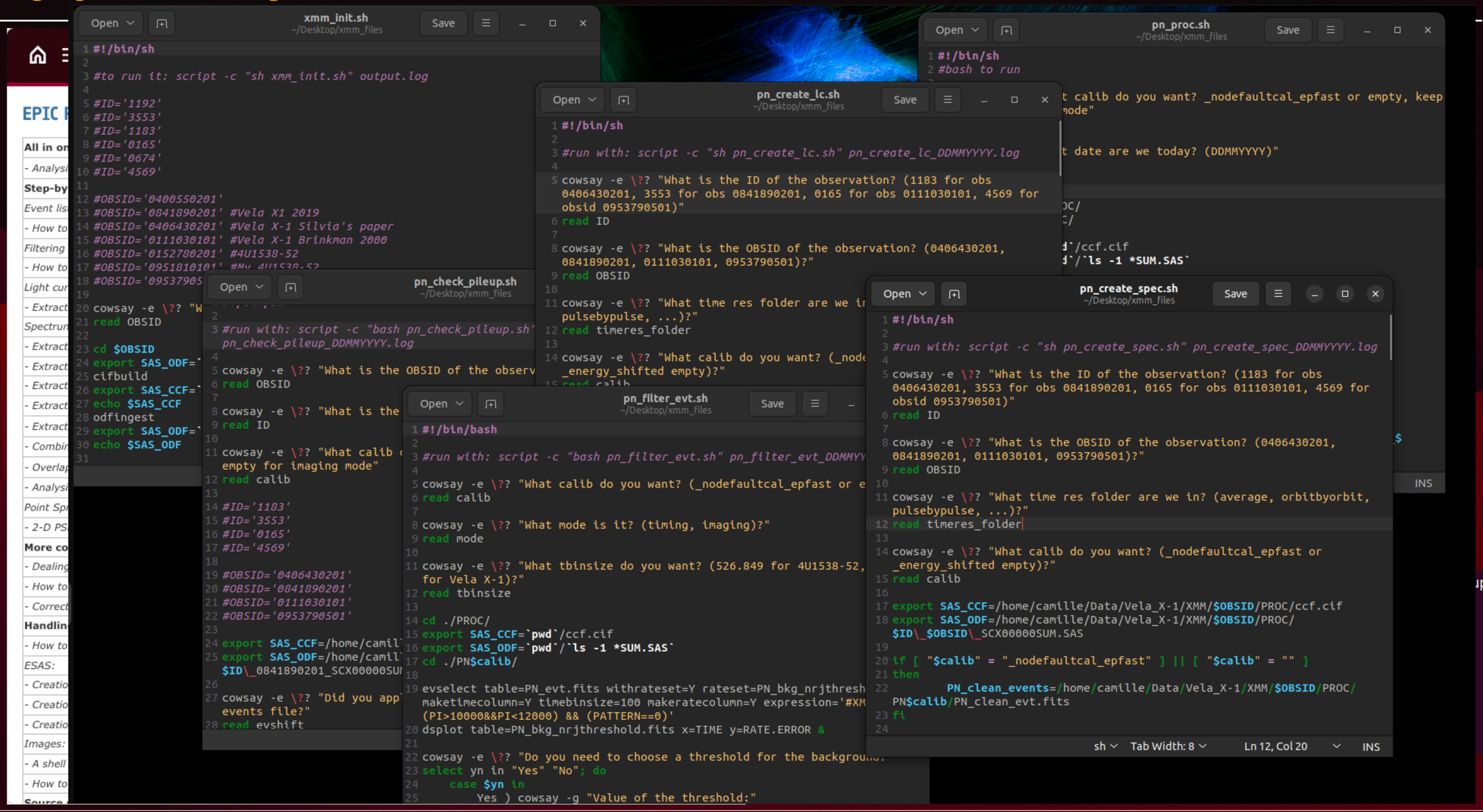
EPIC RELATED THREADS

All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
Event list generation:		
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists	command line	
Filtering against high background:		
- How to filter EPIC event lists for flaring particle background	command line & GUI version	
Light curve generation:		
- Extraction of a light curve for a point-like source (EPIC and RGS)	command line	GUI version
Spectrum extraction:		
- Extraction of MOS spectra from point-like sources	command line	GUI version
- Extraction of MOS spectra from point-like sources taken in timing mode	command line	
- Extraction of pn spectra from point-like sources	command line	GUI version
- Extraction of pn spectra from point-like sources taken in timing mode	command line	
- Extraction of spectra in a few clicks: `especget`		GUI version
- Combining the spectra of the 3 EPIC cameras	command line	
- Overlapping EPIC data treatment: `multixmmselect`		GUI version
- Analysing EPIC spectra from multi-pointing mode data	command line	
Point Spread Function (PSF) generation:		
- 2-D PSF à la carte	command line	
More complex analysis for bright sources		
- Dealing with EPIC Out-of-Time (OoT) events	command line	
- How to evaluate and test pile-up in an EPIC source	command line	
- Correct for rate-dependent energy scale effects in PN burst and timing mode	command line	
Handling of EPIC background		
- How to use EPIC instrumental background files	command line	
ESAS:		
- Creation of EPIC background subtracted, exposure corrected images	command line	
- Creation of EPIC merged background subtracted and exposure corrected images	command line	
- Creation of EPIC spectral analysis files for a cluster radial profile	command line	
Images:		
- A shell script to create attractive EPIC-pn & MOS combined images	dedicated Web page	
- How to Generate Vignetting-corrected Background-subtracted EPIC Images	command line	
Source detection		




```
(base) camille@camille-G5-5590:~/Data/Vela_X-1/XMM/0841890201$ sh pn_check_pileup.sh
 _____________________________________
< What is the OBSID of the observation? >
 -------------------------------------
        \   ^__^
         \  (??)\_______
            (__)\       )\/\
                ||----w |
                ||     ||
0841890201
```

My journey as an XMM user: From bash to Datalabs



My journey as an XMM user: From bash to Datalabs

xmm_init.sh
~/Desktop/xmm_files

```
#!/bin/sh

#to run it: script -c "sh xmm_init.sh" output.log

#ID='1192'
#ID='3553'
#ID='1183'
#ID='0165'
#ID='0674'
#ID='4569'

#OBSID='0400550201'
#OBSID='0841890201' #Vela X1 2019
#OBSID='0406430201' #Vela X-1 Silvia's paper
#OBSID='0111030101' #Vela X-1 Brinkman 2000
#OBSID='0152780201' #4U1538-52
#OBSID='0951810101' #My 4U1538-52
#OBSID='09537905
```

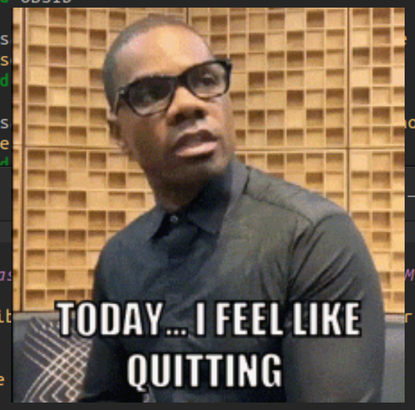
```
cowsay -e \?? "W
read OBSID

cd $OBSID
export SAS_ODF=`
cifbuild
export SAS_CCF=`
echo $SAS_CCF
odfingest
export SAS_ODF=`
echo $SAS_ODF
```

pn_check_pileup.sh
~/Desktop/xmm_files

```
#run with: script -c "bash pn_check_pileup.sh"
pn_check_pileup_DDMMYYYY.log

cowsay -e \?? "What is the OBSID of the observ
read OBSID

cowsay -e \?? "What is the
read ID

cowsay -e \?? "What calib
empty for imaging mode"
read calib

#ID='1183'
#ID='3553'
#ID='0165'
#ID='4569'

#OBSID='0406430201'
#OBSID='0841890201'
#OBSID='0111030101'
#OBSID='0953790501'

export SAS_CCF=/home/camil
export SAS_ODF=/home/camil
$ID\_0841890201_SCX00000SU

cowsay -e \?? "Did you app
events file?"
read evshift
```

pn_create_lc.sh
~/Desktop/xmm_files

```
#!/bin/sh

#run with: script -c "sh pn_create_lc.sh" pn_create_lc_DDMMYYYY.log

cowsay -e \?? "What is the ID of the observation? (1183 for obs
0406430201, 3553 for obs 0841890201, 0165 for obs 0111030101, 4569 for
obsid 0953790501)"
read ID

cowsay -e \?? "What is the OBSID of the observation? (0406430201,
0841890201, 0111030101, 0953790501)?"
read OBSID
```




```
#!/bin/bash

#run with: script -c "bas

cowsay -e \?? "What calib
read calib

cowsay -e \?? "What mode
read mode

cowsay -e \?? "What tbinsize do you want? (526.849 for 4U1538-52,
for Vela X-1)?"
read tbinsize

cd ./PROC/
export SAS_CCF=`pwd`/ccf.cif
export SAS_ODF=`pwd`/`ls -1 *SUM.SAS`
cd ./PN$calib/

evselect table=PN_evt.fits withrateset=Y rateset=PN_bkg_nrjthresh
maketimecolumn=Y timebinsize=100 makeratecolumn=Y expression='#XM
(PI>10000&&PI<12000) && (PATTERN==0)'
dsplot table=PN_bkg_nrjthreshold.fits x=TIME y=RATE.ERROR &

cowsay -e \?? "Do you need to choose a threshold for the backgrou
select yn in "Yes" "No"; do
    case $yn in
        Yes ) cowsay -g "Value of the threshold:"
```

pn_proc.sh
~/Desktop/xmm_files

```
#!/bin/sh
#bash to run

t calib do you want? _nodefaultcal_epfast or empty, keep
node"

t date are we today? (DDMMYYYY)"

OC/
C/

d`/ccf.cif
d`/`ls -1 *SUM.SAS`
```

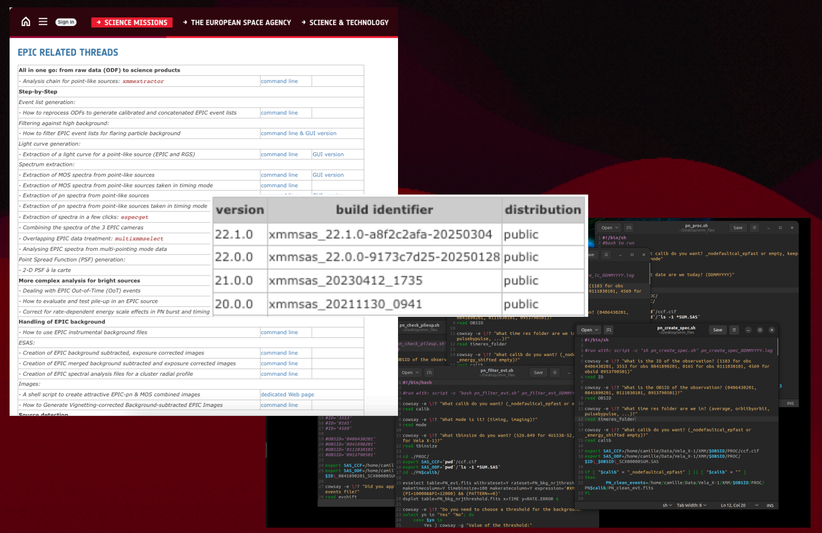
pn_create_spec.sh
~/Desktop/xmm_files

```
#!/bin/sh

#run with: script -c "sh pn_create_spec.sh" pn_create_spec_DDMMYYYY.log

cowsay -e \?? "What is the ID of the observation? (1183 for obs
0406430201, 3553 for obs 0841890201, 0165 for obs 0111030101, 4569 for
obsid 0953790501)"
read ID

cowsay -e \?? "What is the OBSID of the observation? (0406430201,
0841890201, 0111030101, 0953790501)?"
read OBSID

cowsay -e \?? "What time res folder are we in? (average, orbitbyorbit,
pulsebypulse, ...)?"
read timeres_folder

cowsay -e \?? "What calib do you want? (_nodefaultcal_epfast or
_energy_shifted empty)?"
read calib

export SAS_CCF=/home/camille/Data/Vela_X-1/XMM/$OBSID/PROC/ccf.cif
export SAS_ODF=/home/camille/Data/Vela_X-1/XMM/$OBSID/PROC/
$ID\_$OBSID\_SCX00000SUM.SAS

if [ "$calib" = "_nodefaultcal_epfast" ] || [ "$calib" = "" ]
then
        PN_clean_events=/home/camille/Data/Vela_X-1/XMM/$OBSID/PROC/
PN$calib/PN_clean_evt.fits
fi
```

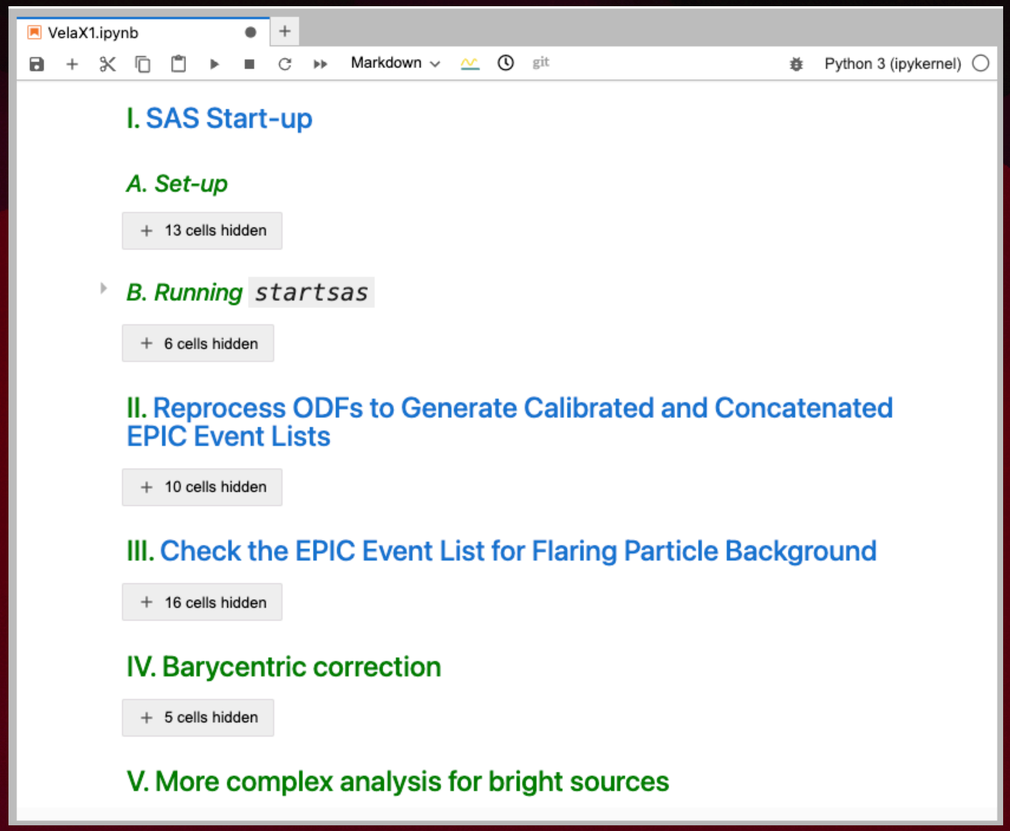
sh Tab Width: 8 Ln 12, Col 20 INS

My journey as an XMM user: From bash to Datalabs

My journey as an XMM user: From bash to Datalabs



My journey as an XMM user: From bash to Datalabs



ESA Datalabs

- Run on a JupyterLab interface
- Python to run SAS commands: PySAS
- No additional installations needed
- SAS 21.0 and 22.1.0
- All your favourite Python packages for interactive X-ray data analysis (PyXspec, BXA, LCviz, JS9, jaxspec)

ESA Datalabs

- Run on a JupyterLab interface
- Python to run SAS commands: PySAS
- No additional installations needed
- SAS 21.0 and 22.1.0
- All your favourite Python packages for interactive X-ray data analysis

(PyXspec, BXA, LCviz, JS9, jaxspec)

Pure Python library using JAX for X-ray spectral fitting (led at IRAP, still under development)
see Dupourqué, Barret, Diez et al. (2024) + science case with XRISM in Diez et al. (2025)

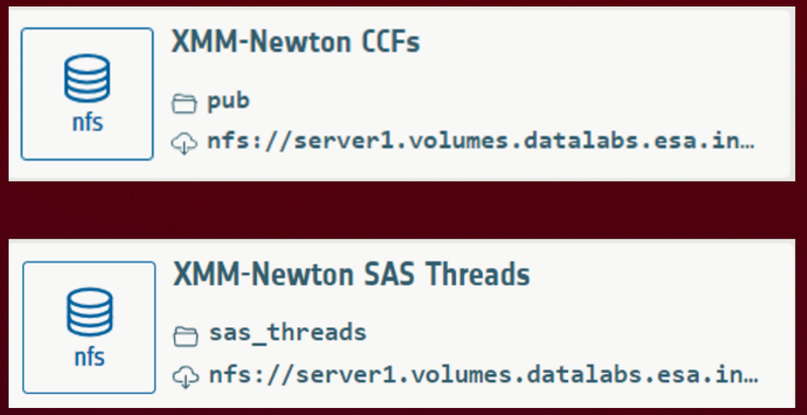
ESA Datalabs

- Run on a JupyterLab interface
- Python to run SAS commands: PySAS
- No additional installations needed
- SAS 21.0 and 22.1.0
- All your favourite Python packages for interactive X-ray data analysis

(PyXspec, BXA, LCviz, JS9, jaxspec)

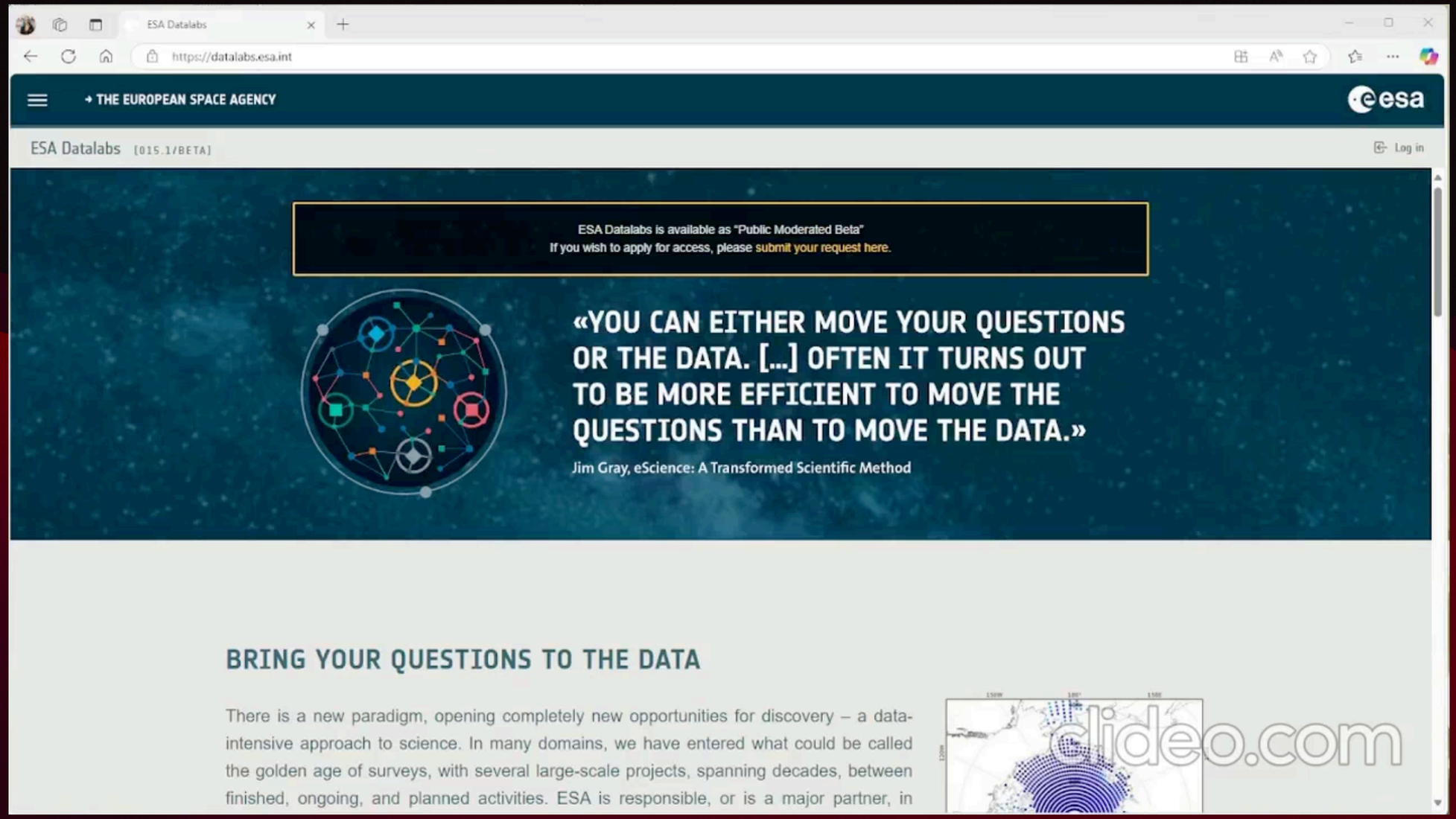
Pure Python library using JAX for X-ray spectral fitting (led at IRAP, still under development)
see Dupourqué, Barret, Diez et al. (2024) + science case with XRISM in Diez et al. (2025)

Datalab instance



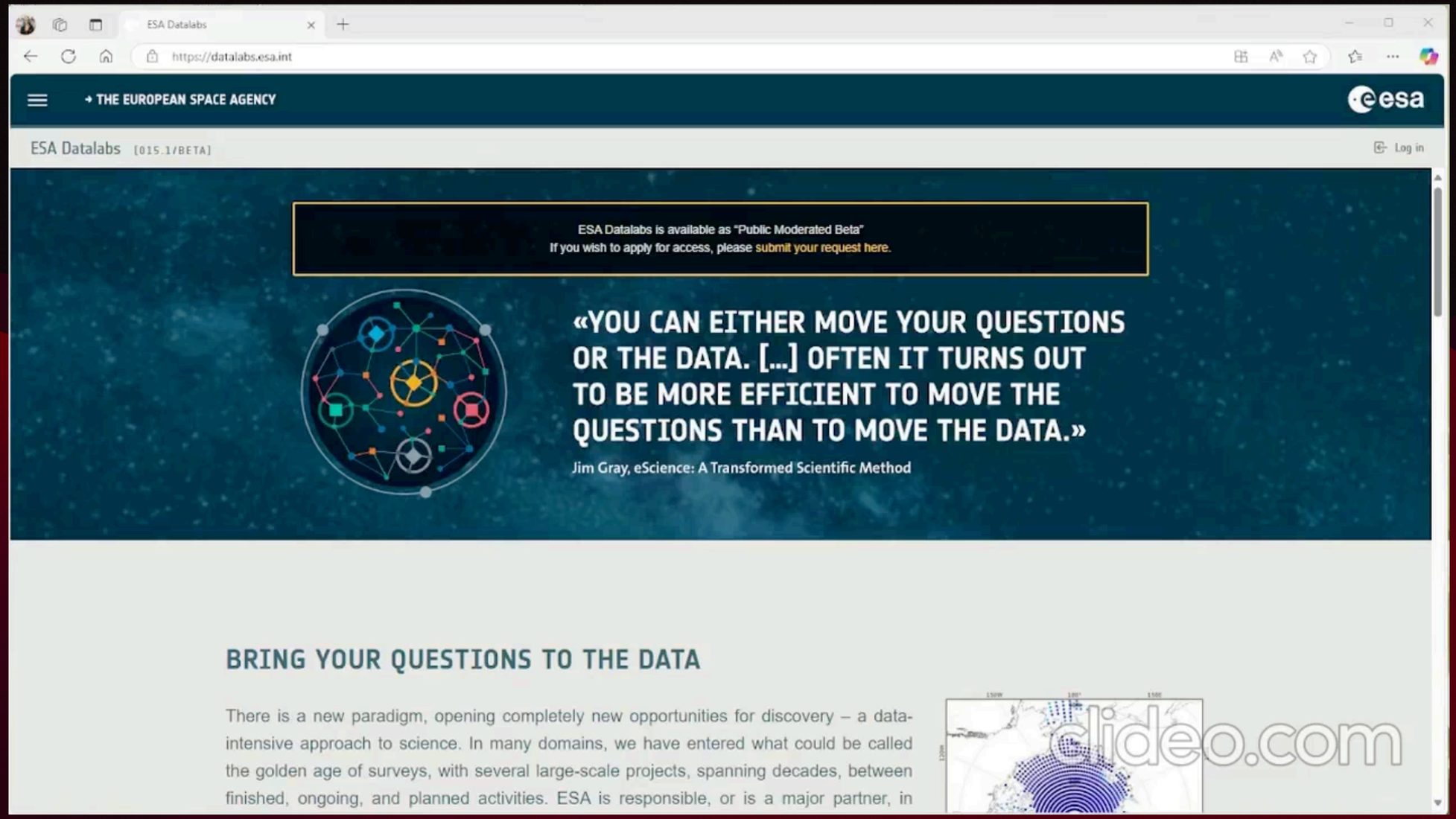
Data volumes

ESA Datalabs



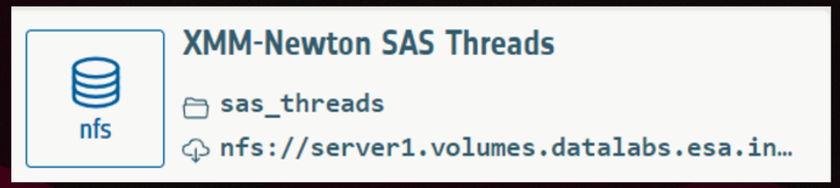
Courtesy of: Esin Gülbahar (former ESA YGT, now PhD in Bamberg, Germany)

ESA Datalabs

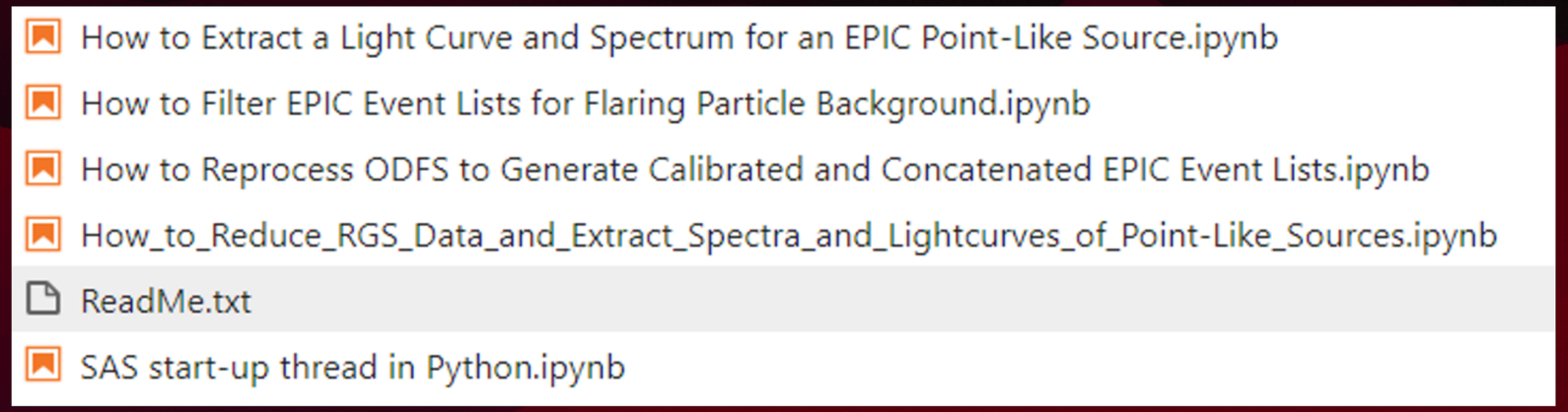


Courtesy of: Esin Gülbahar (former ESA YGT, now PhD in Bamberg, Germany)

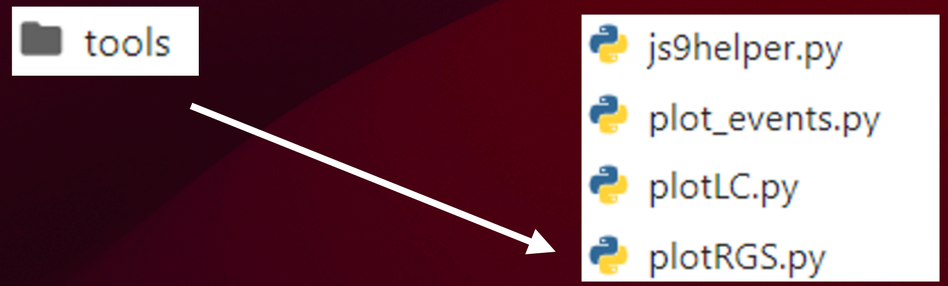
ESA Datalabs



Currently mountable as Data Volume, working on automatic mount.



Converting the public threads to Python format, accessible for Datalabs environment

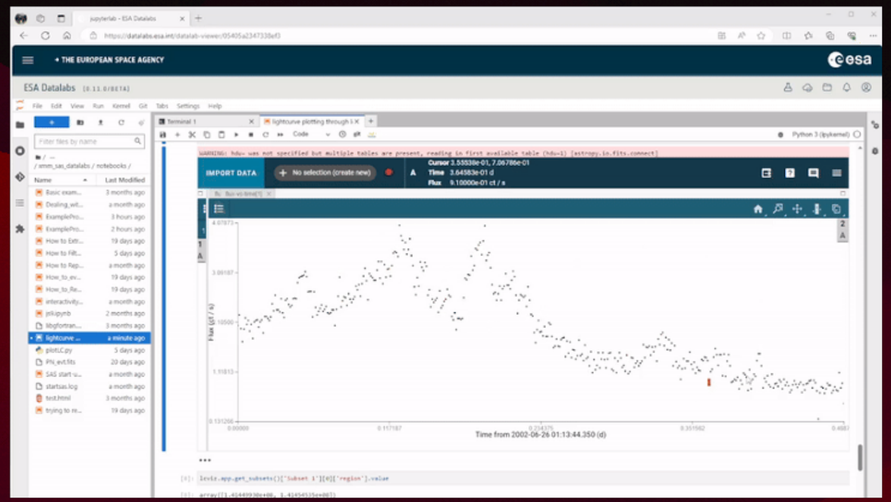


Functions for interactive and easy visualisation

Courtesy of: Esin Gülbahar

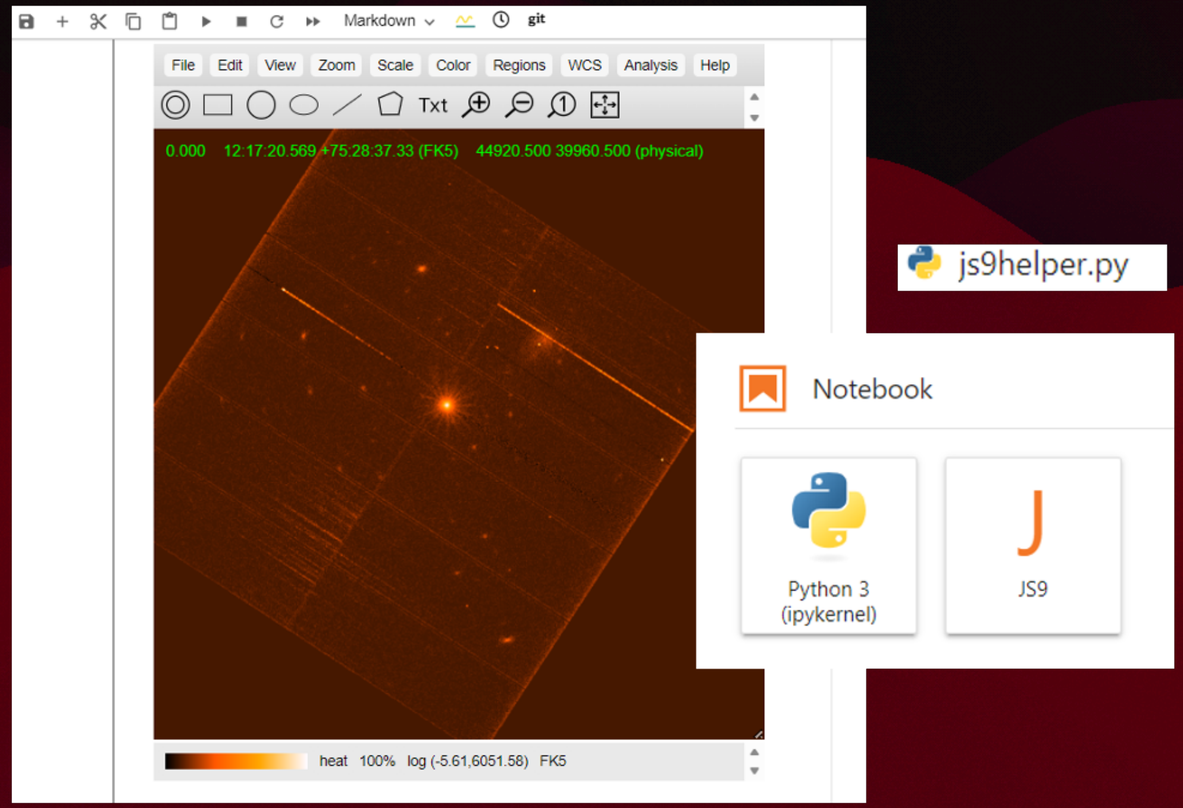
ESA Datalabs

- Interactive light curve plotting with LCviz



plotLC.py

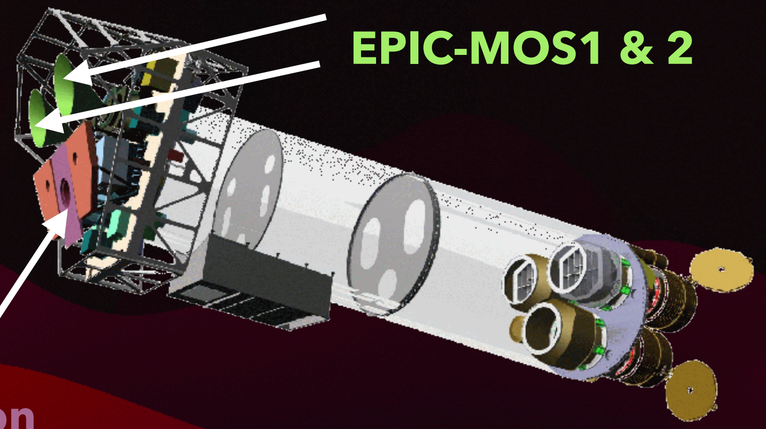
- Image visualisation directly in JupyterLab with JS9



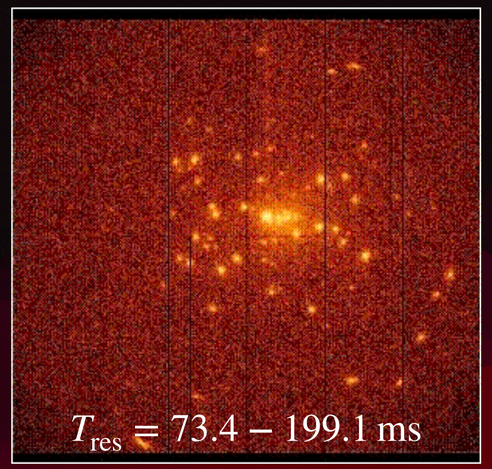
Courtesy of: Esin Gülbahar

A few things to watch out for bright sources with EPIC/RGS

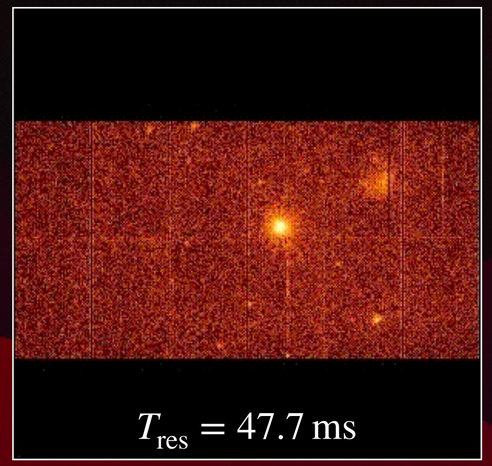
EPIC-pn -MOS: timing mode



EPIC-pn Full Frame and Extended Full Frame mode

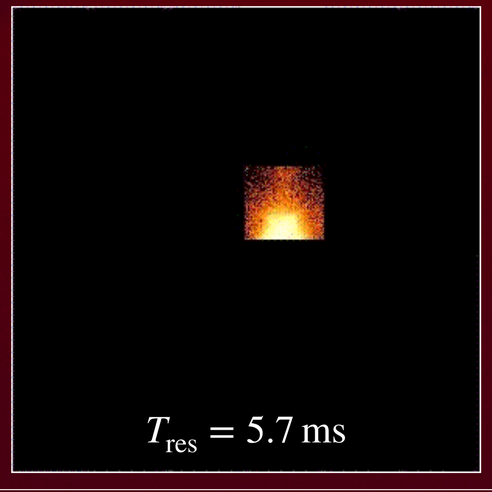


EPIC-pn Large Window mode

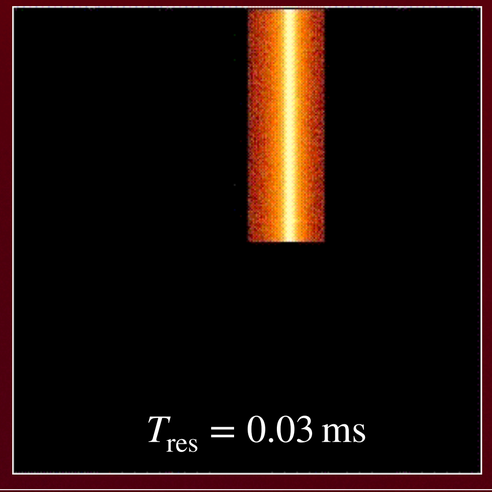


Full energy range	0.3-12 keV
Orbit	Elliptic (e ~ 0.816585)
Launch	Dec. 10th 1999

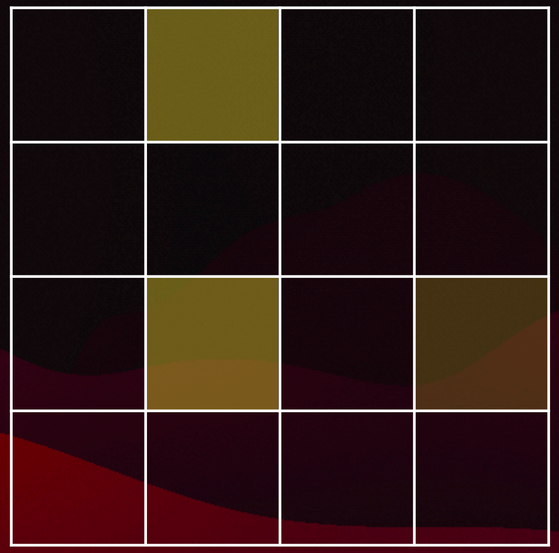
EPIC-pn Small Window mode



EPIC-pn Timing mode

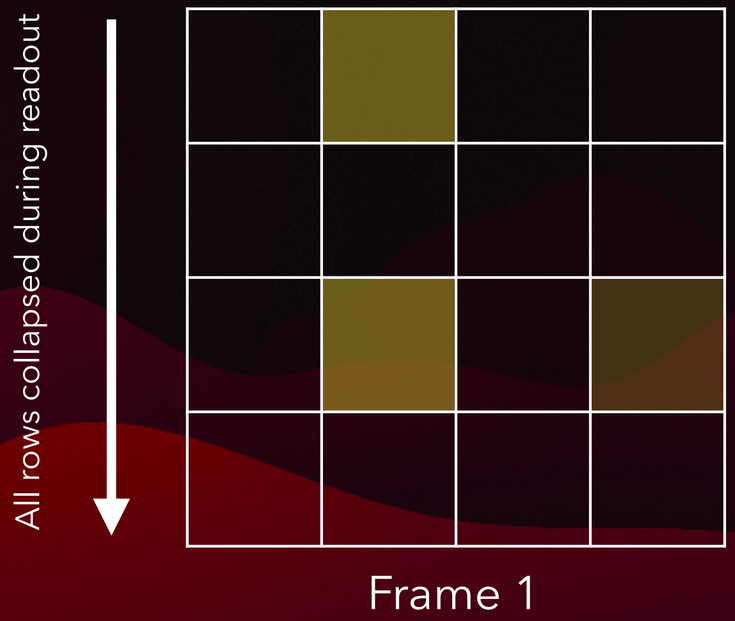


Timing mode

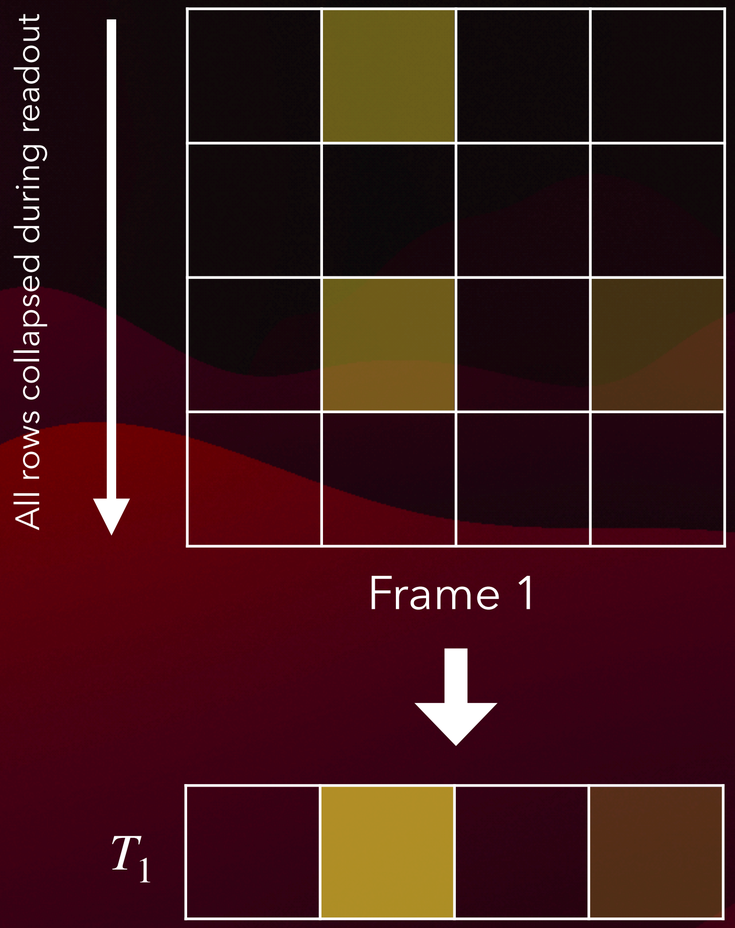


Frame 1

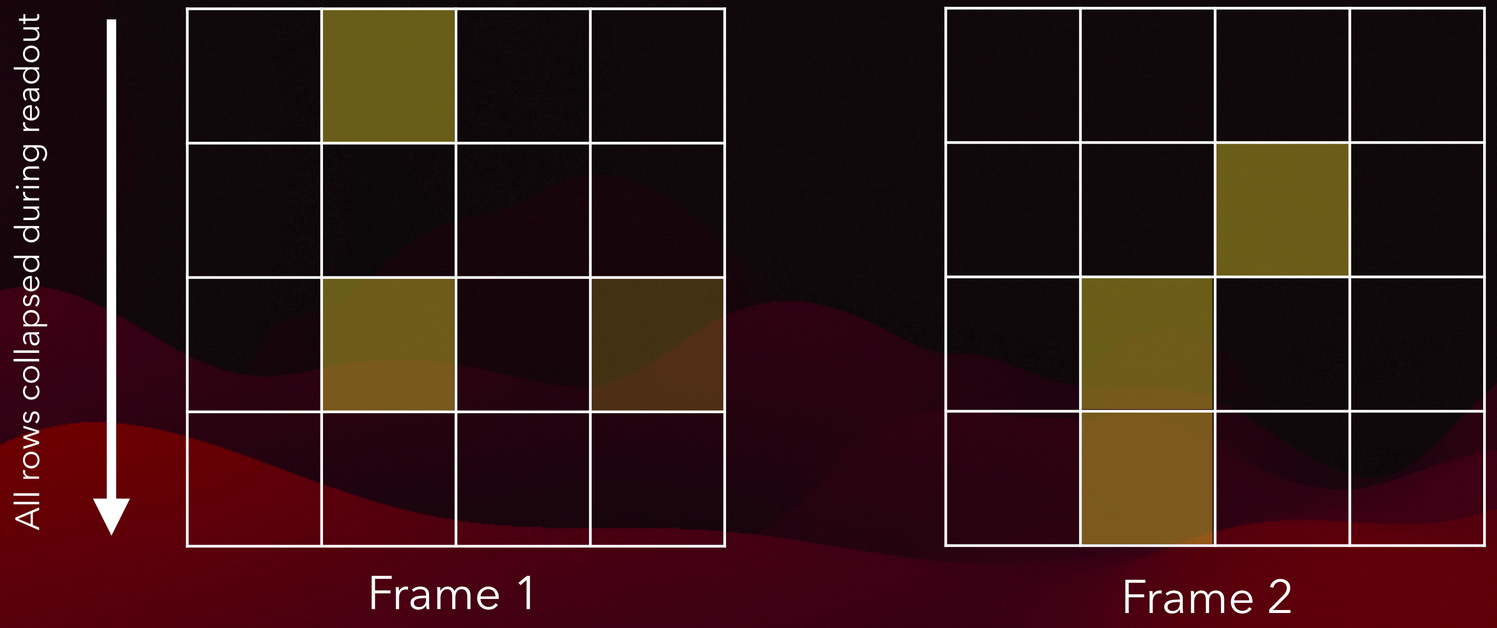
Timing mode



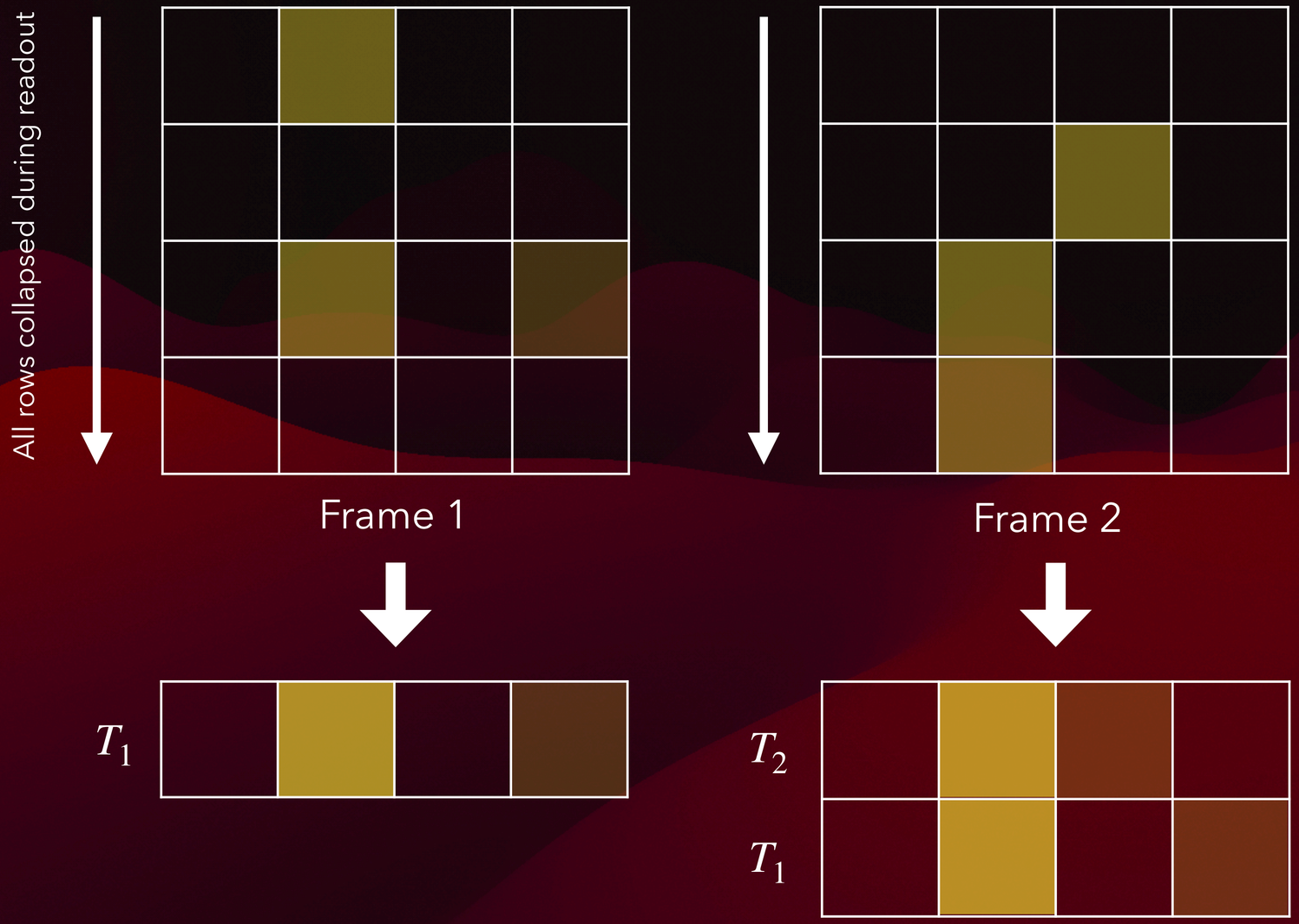
Timing mode



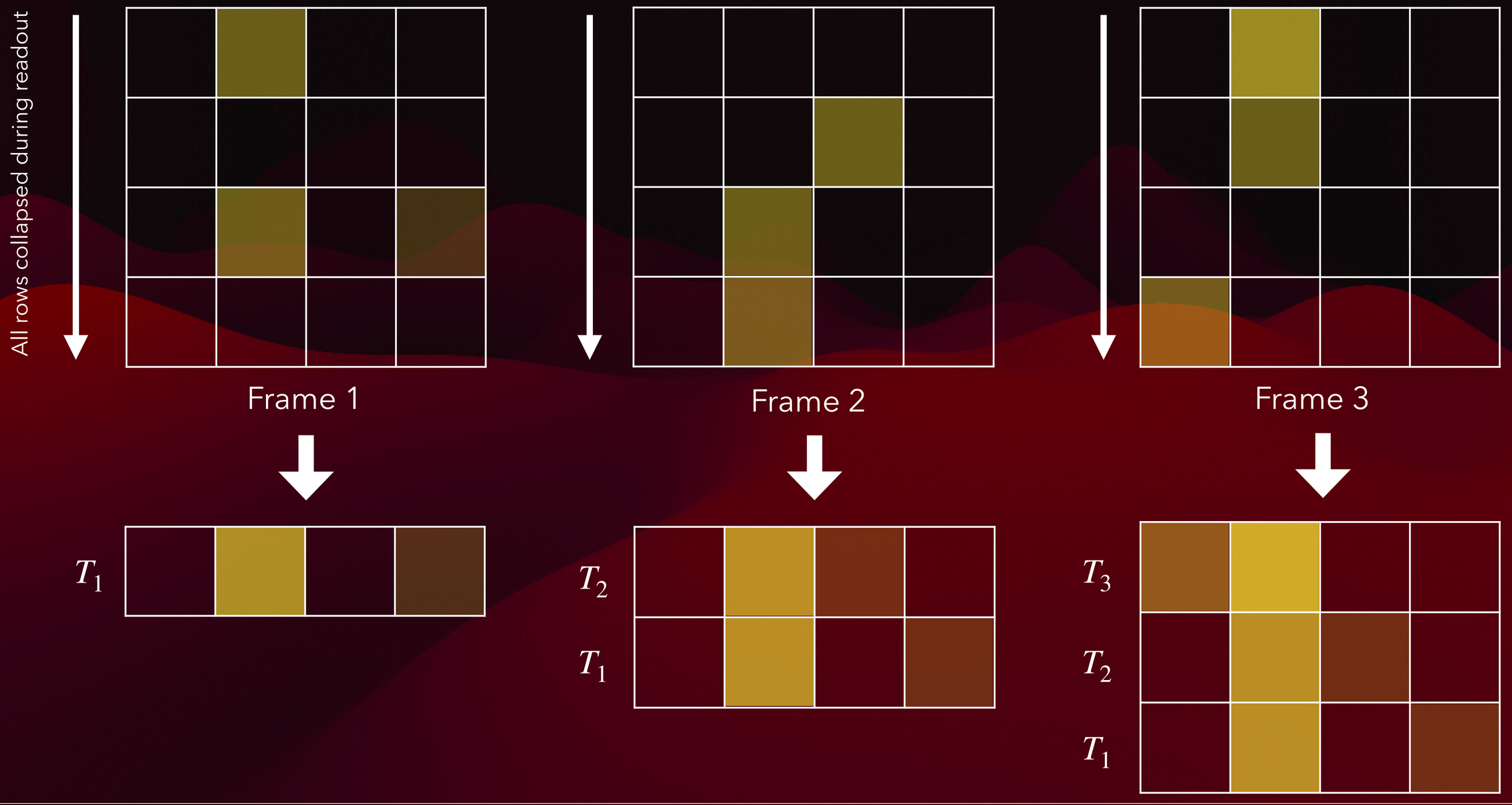
Timing mode



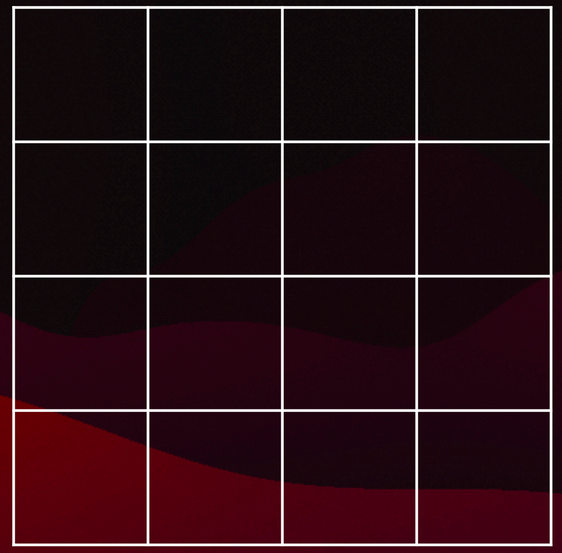
Timing mode



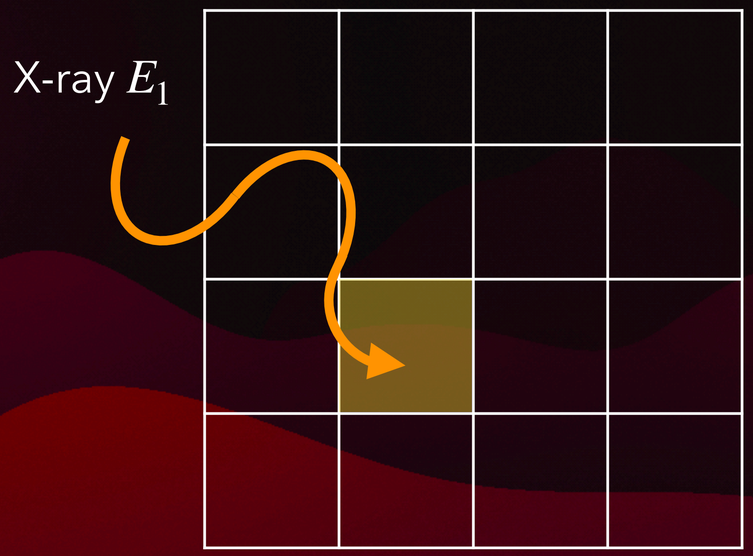
Timing mode


All rows collapsed during readout
Frame 1
Frame 2
Frame 3
T_1
T_2
T_1
T_3
T_2
T_1

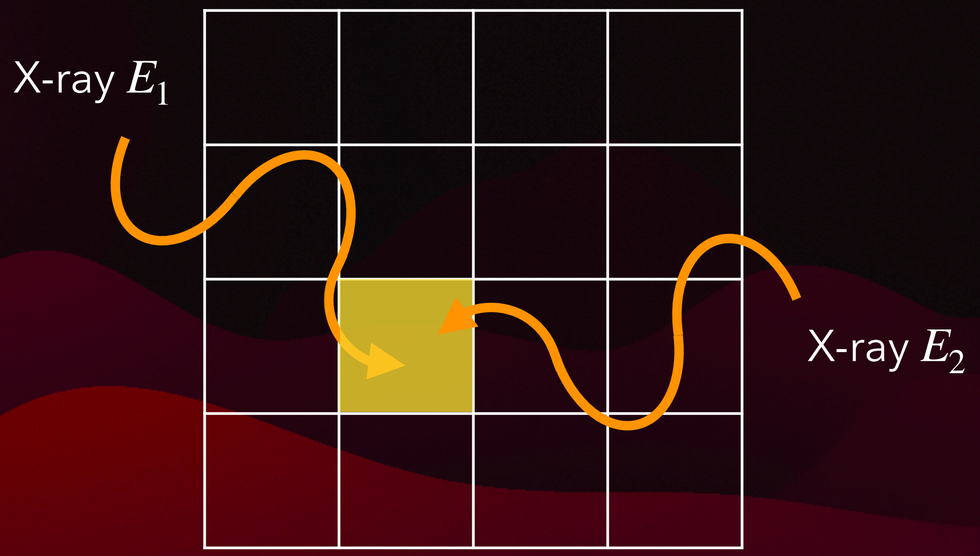
Pile-up



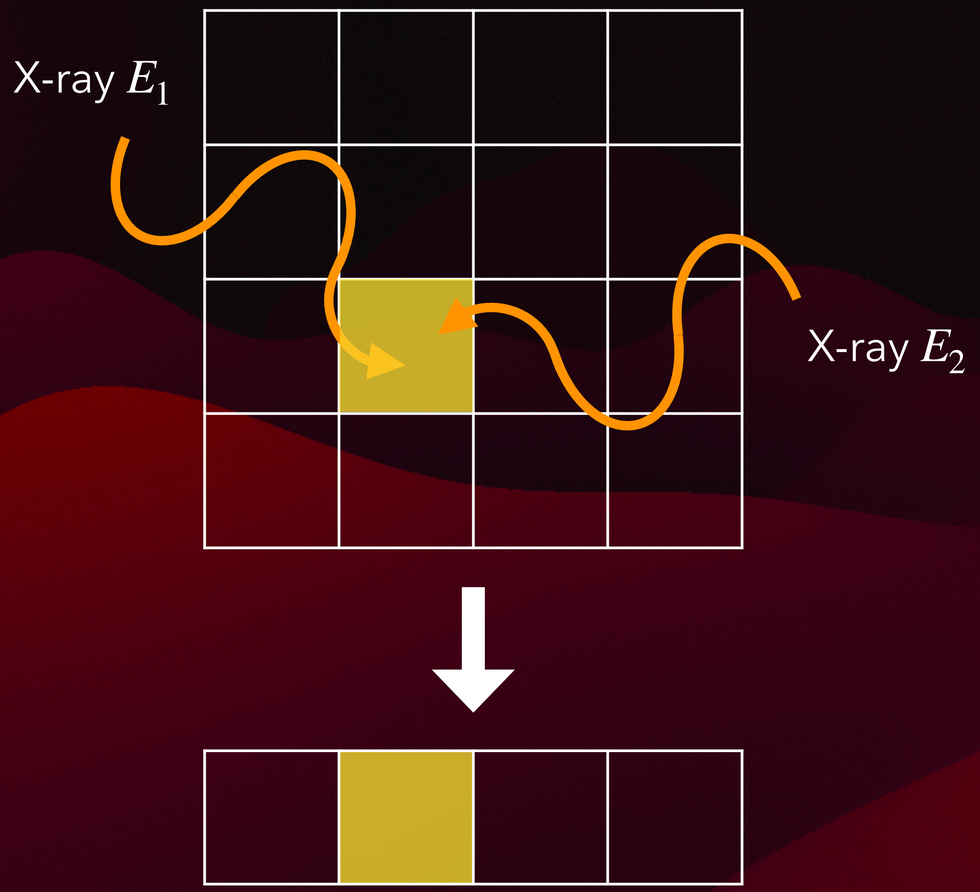
Pile-up



Pile-up

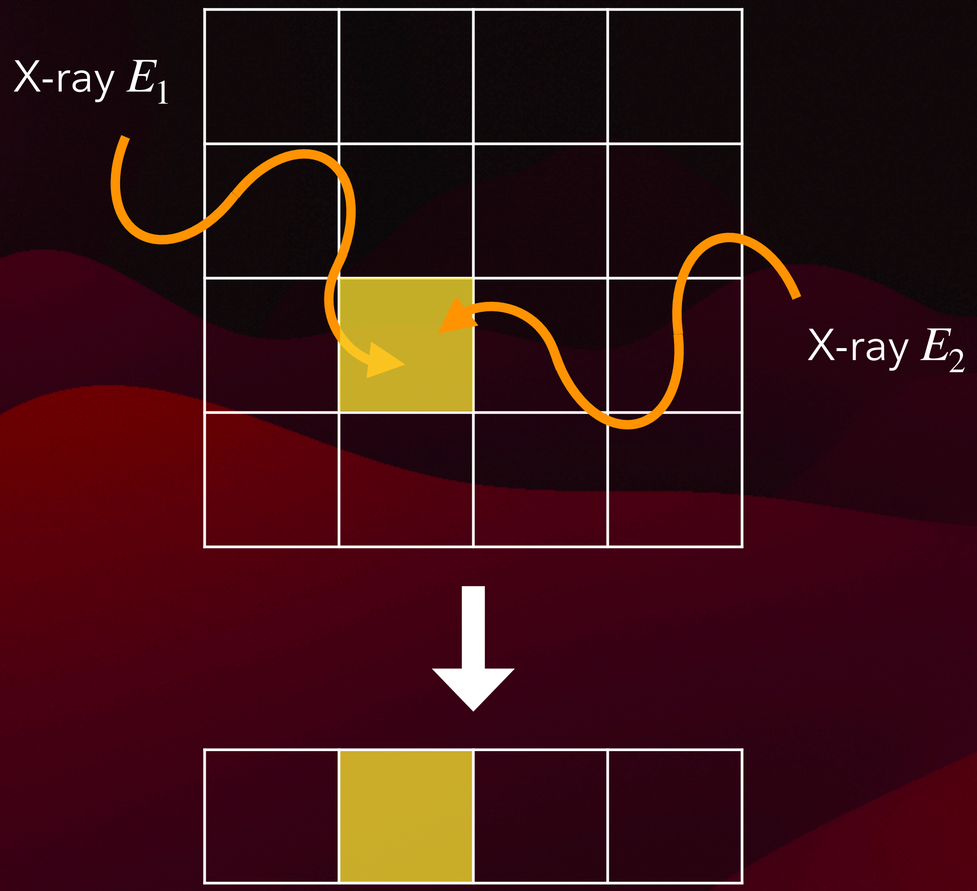


Pile-up

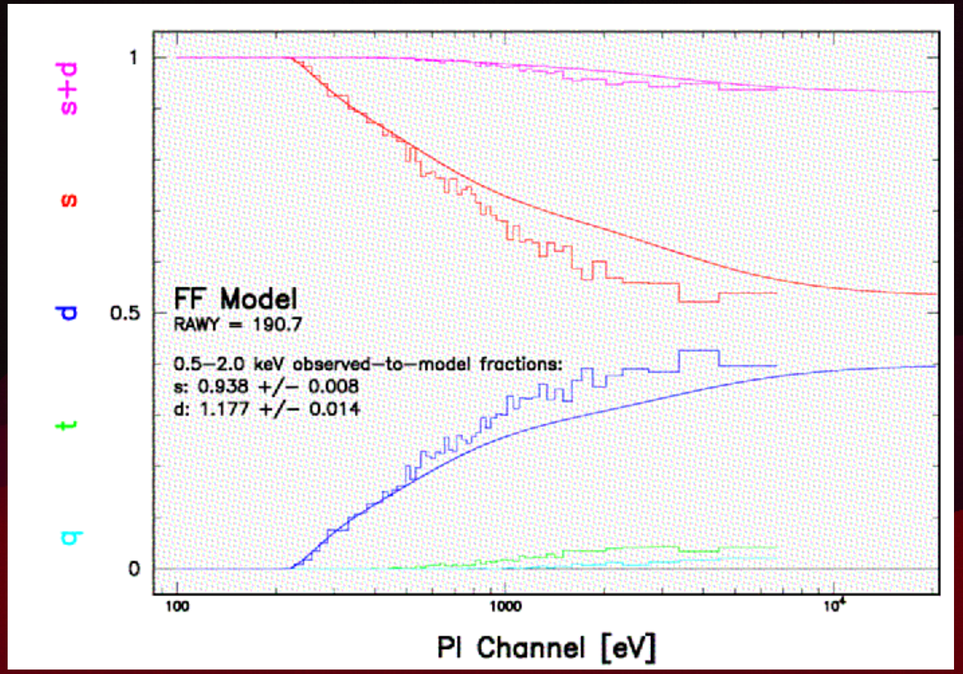


Measured with $E_1 + E_2$
if $T_{\text{arrival},2} - T_{\text{arrival},1} < T_{\text{readout}}$

Pile-up

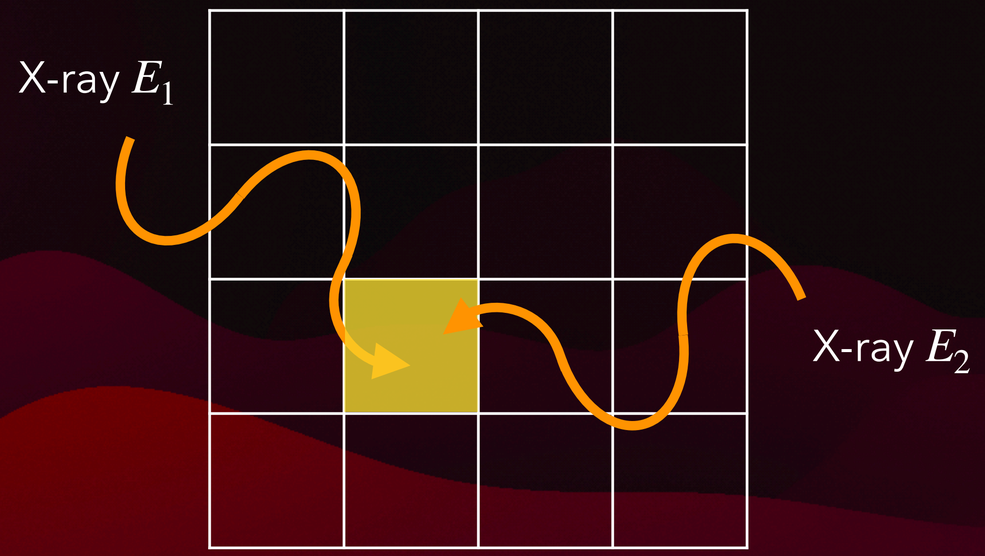


Measured with $E_1 + E_2$
if $T_{\text{arrival},2} - T_{\text{arrival},1} < T_{\text{readout}}$



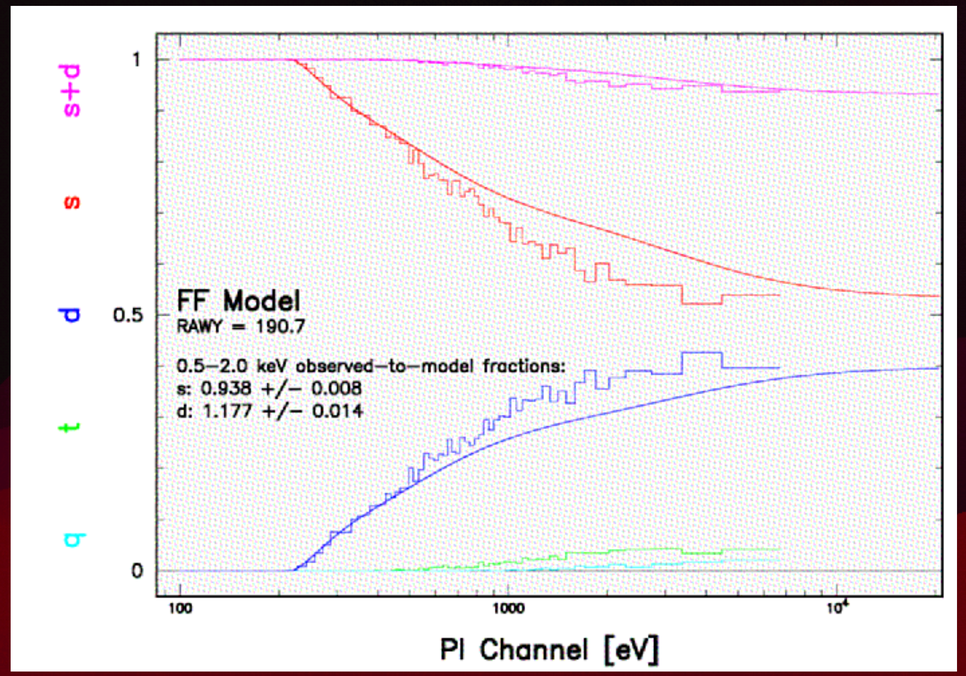
Over estimation of high-energy photons

Pile-up



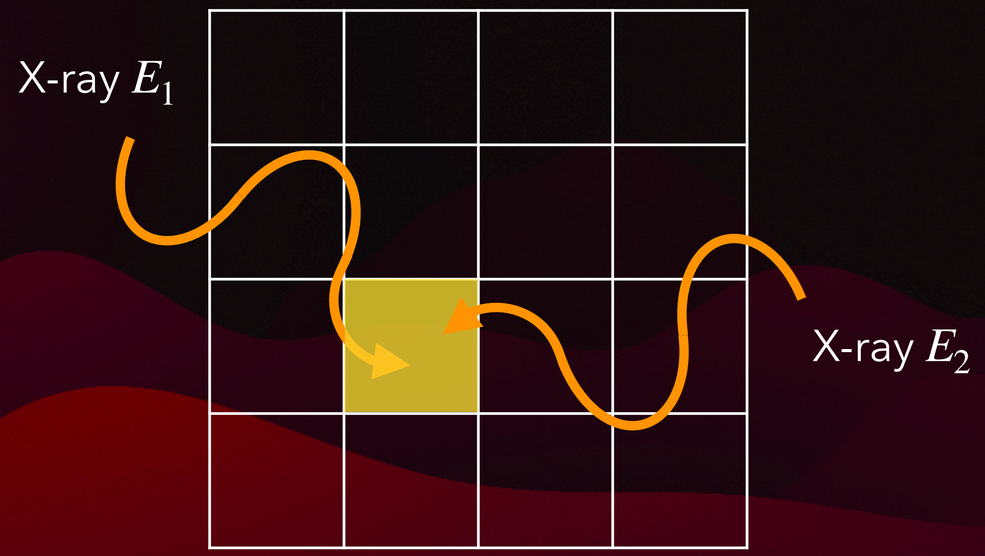
Measured with $E_1 + E_2$
if $T_{\text{arrival},2} - T_{\text{arrival},1} < T_{\text{readout}}$

?



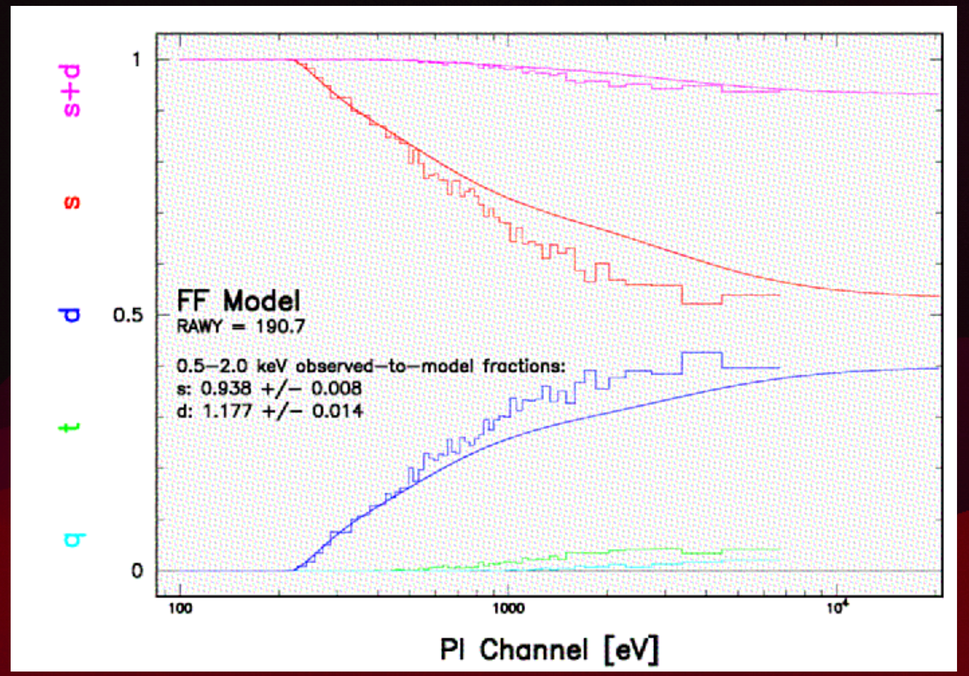
Over estimation of high-energy photons

Pile-up



Measured with $E_1 + E_2$
if $T_{\text{arrival},2} - T_{\text{arrival},1} < T_{\text{readout}}$

?



Over estimation of high-energy photons

→ Remove piled-up columns (see tutorial later)


PRETTY CONVINCING, HUH?


PRETTY CONVINCING, HUH?

Let’s start the data extraction

Example with Vela X-1

Let’s start the data extraction

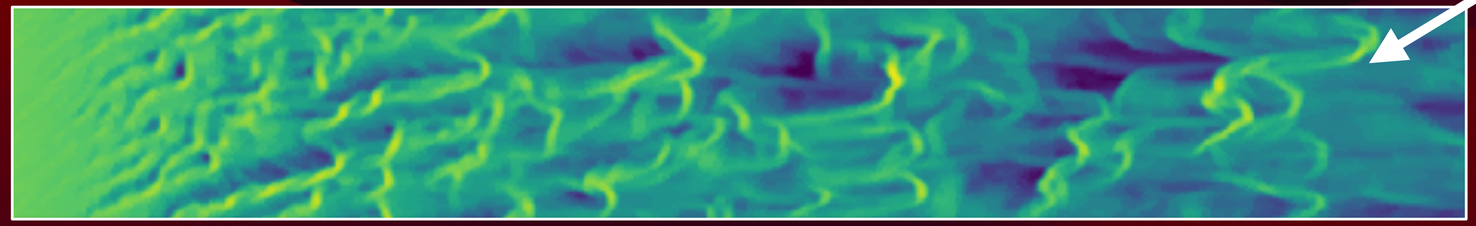
Example with Vela X-1

(All steps are explained in the Notebooks)

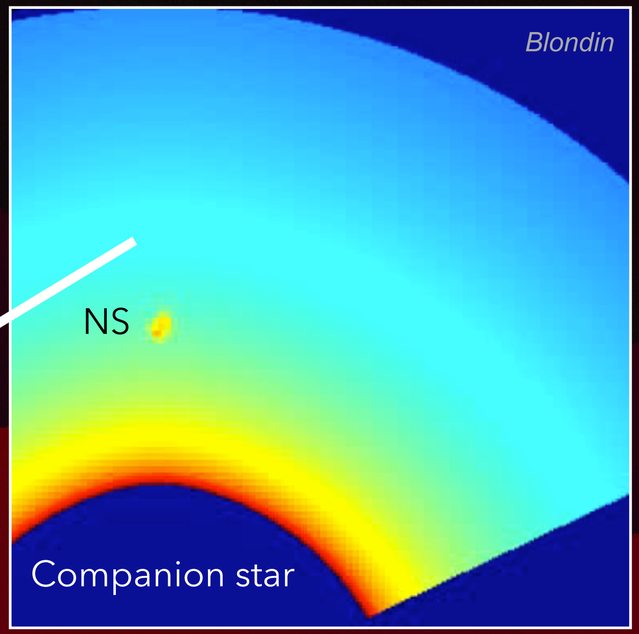
Vela X-1: analysis of the stellar wind variability

Complex structure

- Clumpy wind + ionisation and accretion wakes
- Highly disturbed wind + velocity perturbations
- Affects accretion rate and X-ray emission



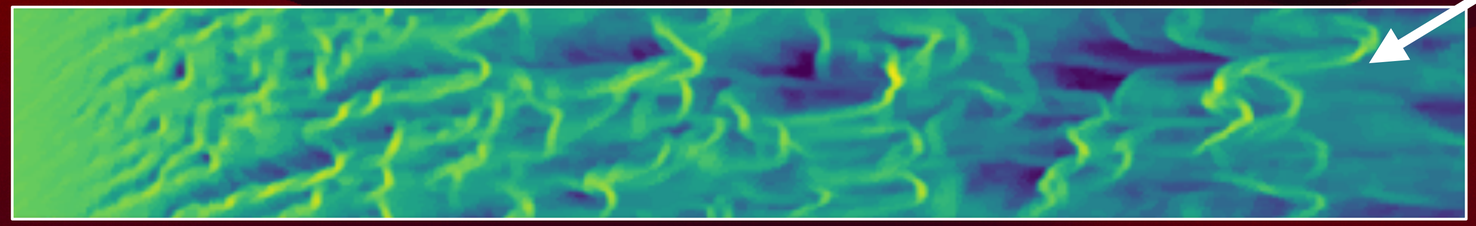
El Mellah et al. (2018)



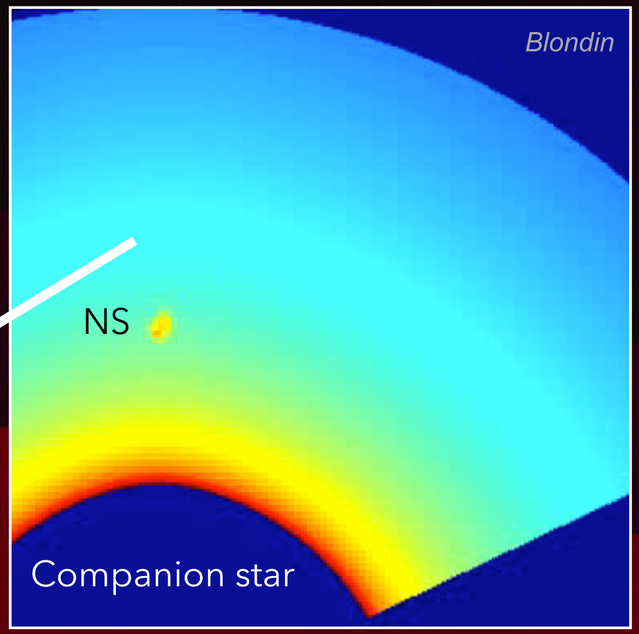
Vela X-1: analysis of the stellar wind variability

Complex structure

- Clumpy wind + ionisation and accretion wakes
- Highly disturbed wind + velocity perturbations
- Affects accretion rate and X-ray emission



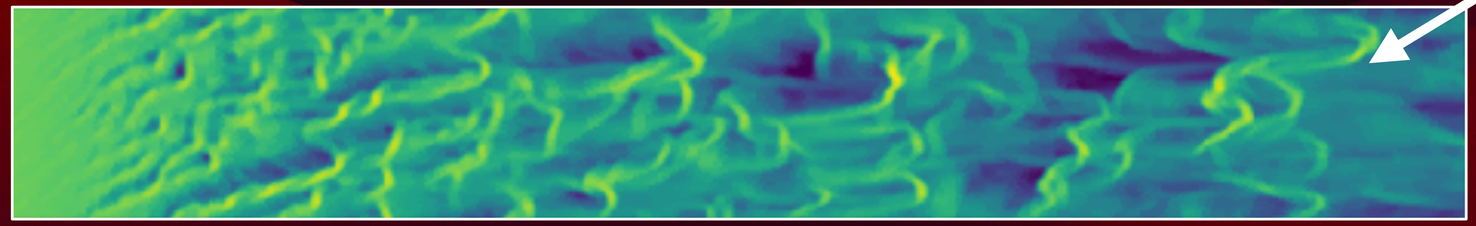
El Mellah et al. (2018)



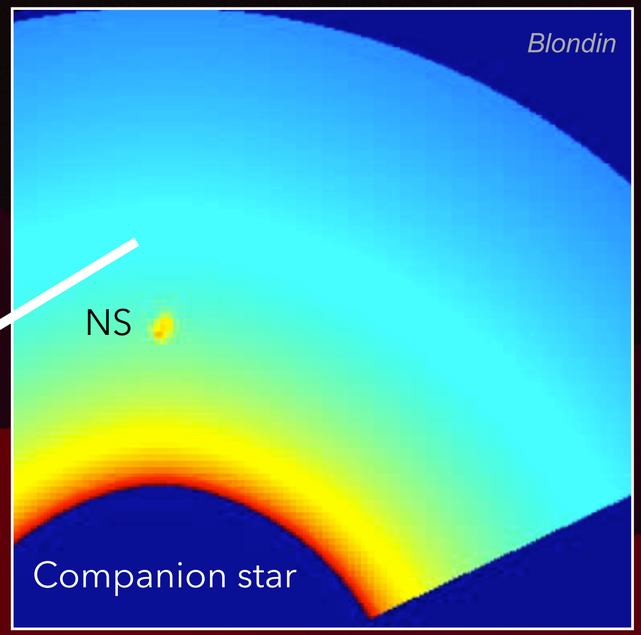
Vela X-1: analysis of the stellar wind variability

Complex structure

- Clumpy wind + ionisation and accretion wakes
- Highly disturbed wind + velocity perturbations
- Affects accretion rate and X-ray emission



El Mellah et al. (2018)



See Diez et al. (2022, 2023, 2025) for previous analyses with NuSTAR, XMM-Newton and XRISM

I. SAS Start-up

A. Set-up

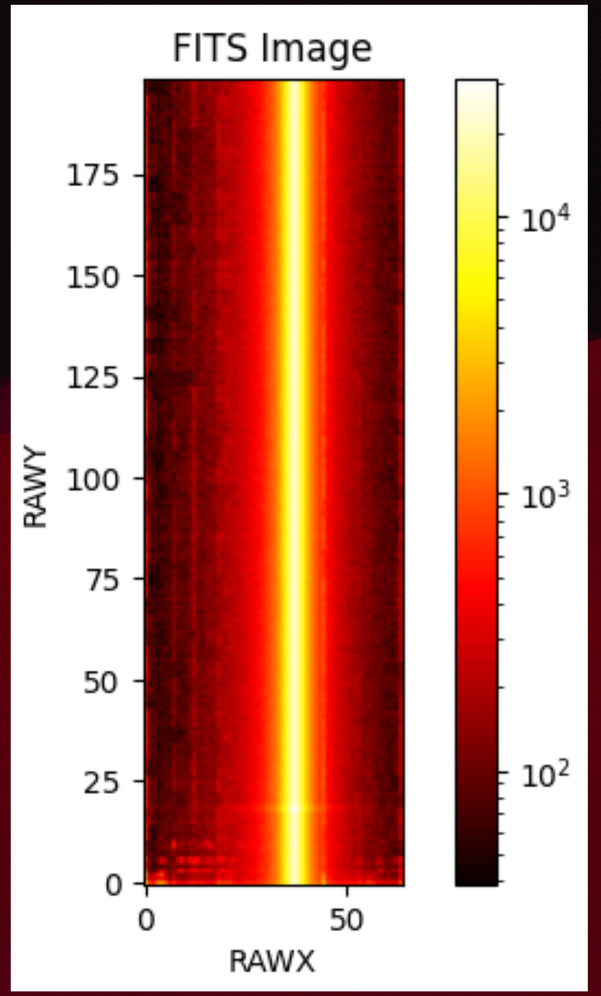
- Import Python libraries and other tools of interest
- Define path for Current Calibration Files (CCFs)

B. Running `startsas`

- Choose the Observation Data File (ODF) you are interested in
- Generate a Calibration Index File for such ODF
- Create the Observation Summary File for such ODF

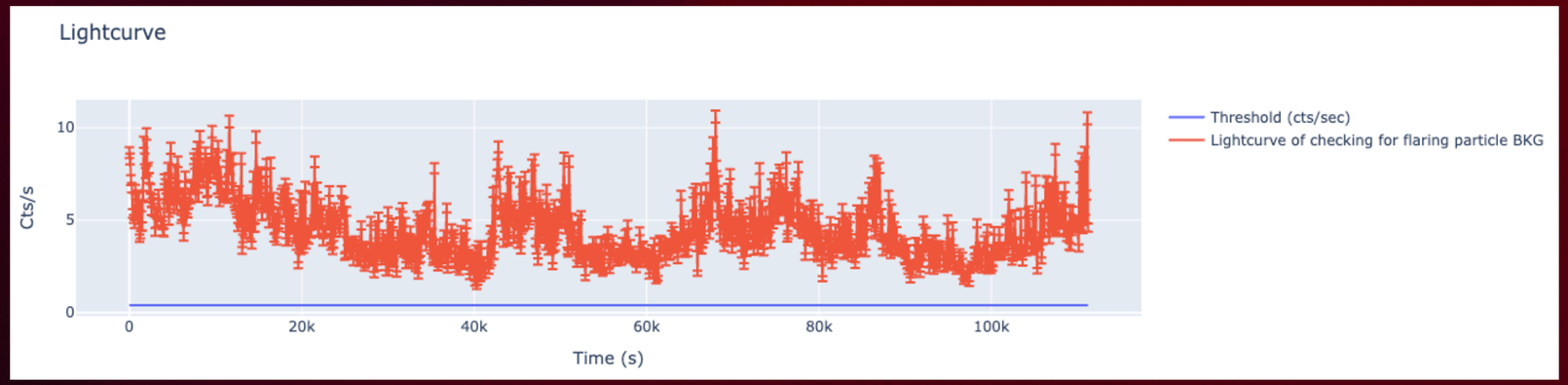
II. Reprocess ODFs to Generate Calibrated and Concatenated EPIC Event Lists

- `epproc`
- If you don’t want to use the default calibration, you can change the arguments of `epproc`
- Generate an image



III. Check the EPIC Event List for Flaring Particle Background

- Extract lightcurve for all events
- Extract lightcurve at high energies for single events
- Compare lightcurves and see if flaring particle background is observed (i.e. higher than a user-defined threshold)
- For this observation, Vela X-1 is so bright that the overall countrate strongly dominates any possible flaring particle background



IV. Barycentric correction

- `barycen` to convert from the local satellite frame to Barycentric Dynamical Time (TDB)
- Important for timing analyses

IV. Barycentric correction

The `barycen` task is designed to convert times from the local satellite frame to Barycentric Dynamical Time (TDB) using a specified table in a dataset containing times in XMM-Newton MET (Mission Elapsed Time) format. The task checks if barycentric conversion has been performed, exits if true, and proceeds to correct time tags in the specified column. Additionally, it converts interval start and stop times to TDB if a Good Time Interval extension exists and corrects time tags in EXPOSU tables if the relevant parameter is enabled. The task updates TIMEREF, TSTART, TSTOP, and TELAPSE attributes where necessary.

```
nobarycen_clean_evt_file = proc_dir+'/PN_clean_evt_nobarycen.fit'
```

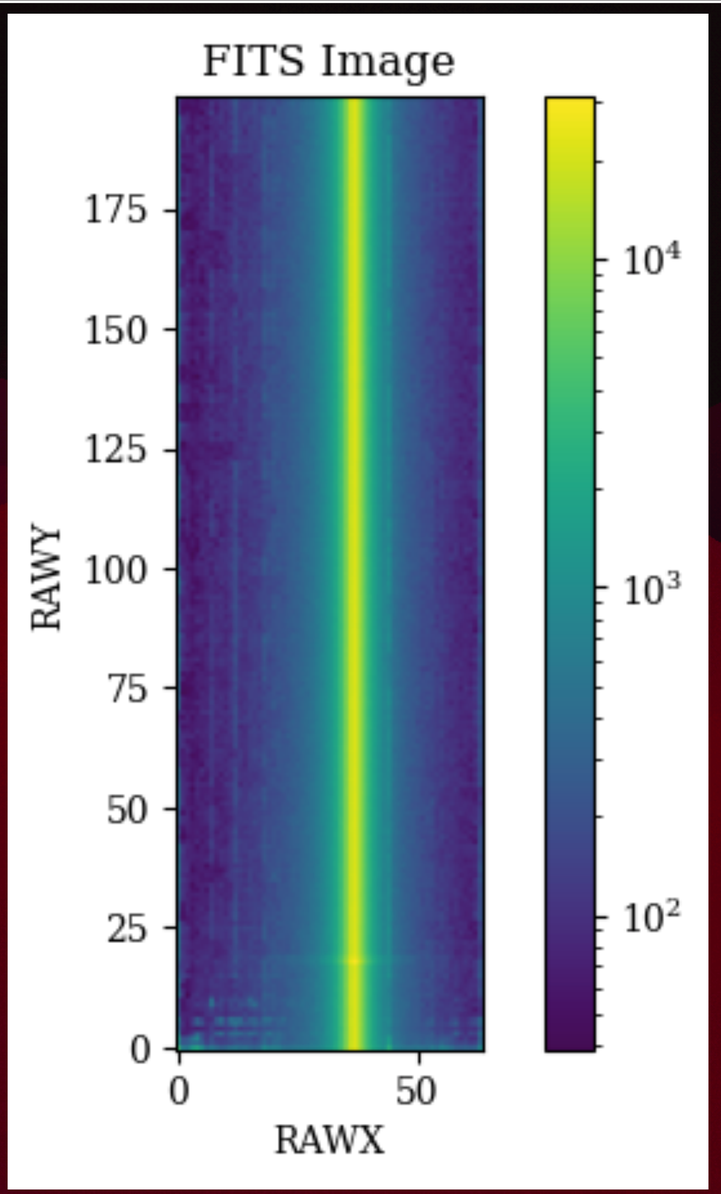
```
# Use the 'copy' function from the 'shutil' module to copy the file 'PN_clean_evt.fits'
# to a new file named 'PN_clean_evt_nobarycen_cor.fits'.
shutil.copy(out_cleanEVFile, nobarycen_clean_evt_file)
```

```
# SAS Command
cmd        = "barycen" # SAS task to be executed

# Arguments of SAS Command
inargs     = [f'table={out_cleanEVFile}:EVENTS']

print("   SAS command to be executed: "+cmd+", with arguments; \n")
inargs
```

```
w(cmd, inargs).run()
```

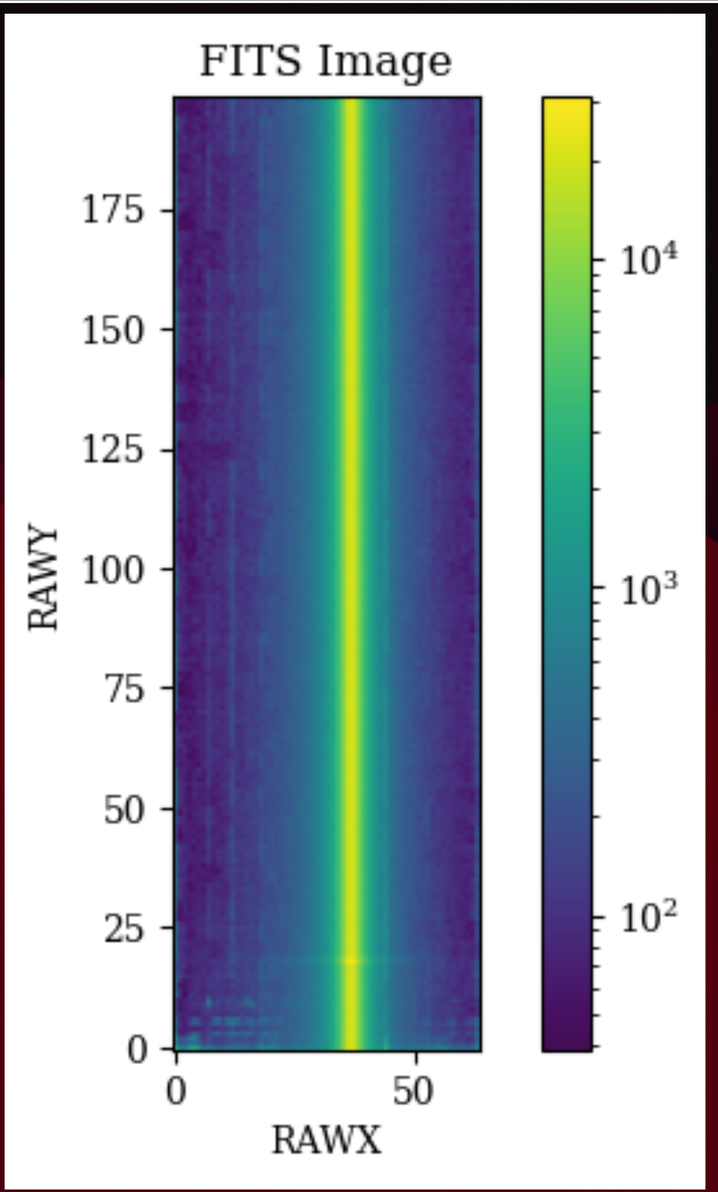
V. More complex analysis for bright sources

V. More complex analysis for bright sources



V. More complex analysis for bright sources

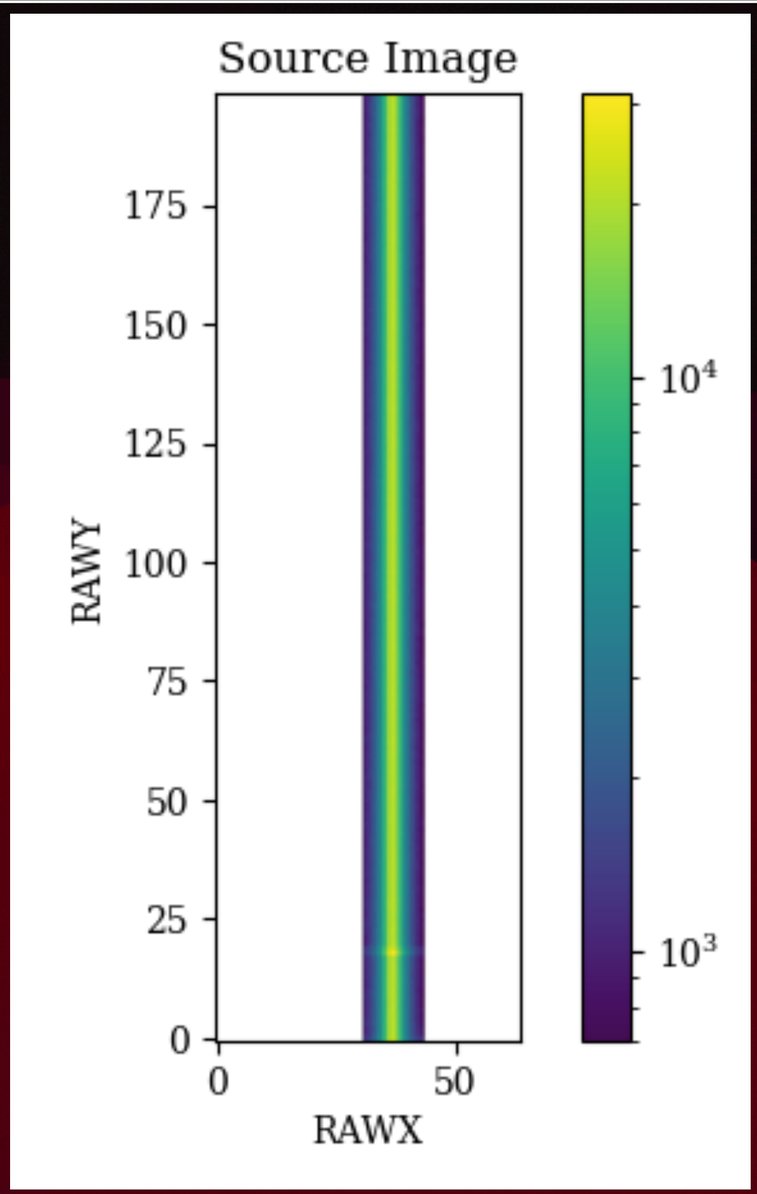
A. Pile-up correction



V. More complex analysis for bright sources

A. Pile-up correction

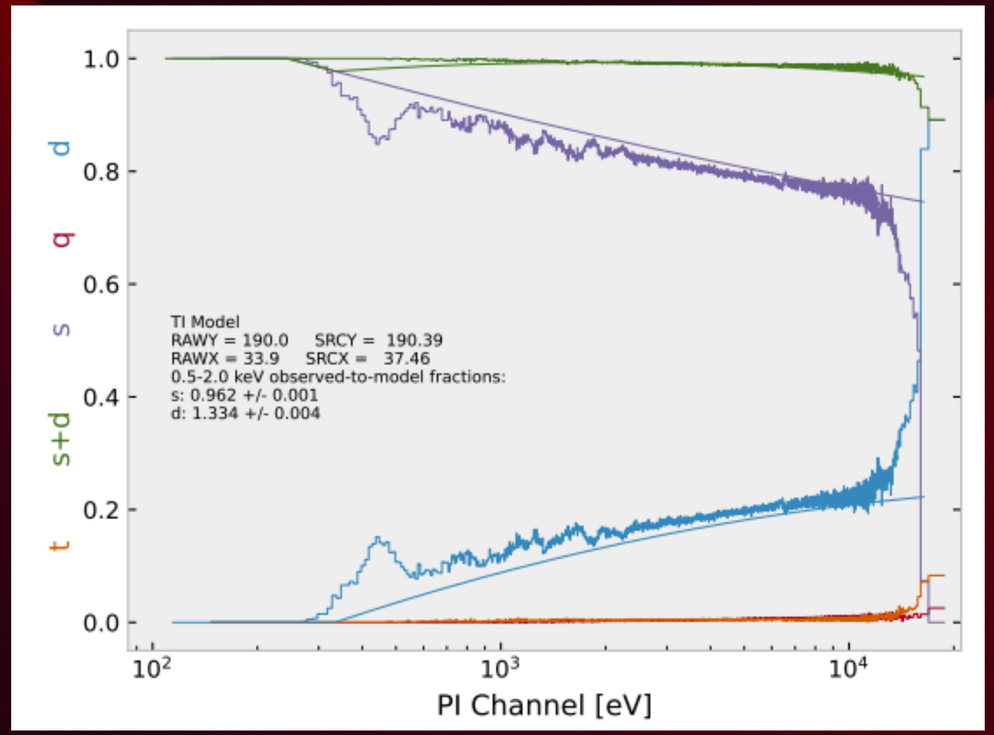
- Extract source region (background region not needed for such bright sources)

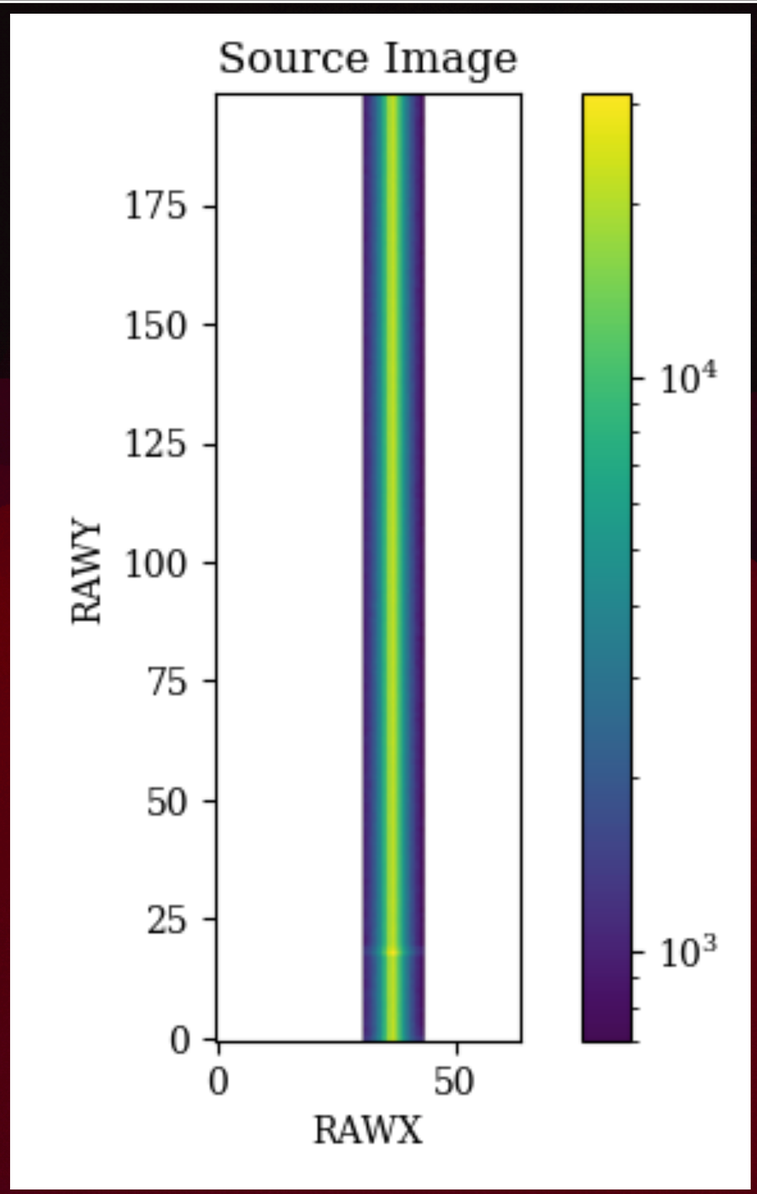


V. More complex analysis for bright sources

A. Pile-up correction

- Extract source region (background region not needed for such bright sources)
- Checks with `epatplot`

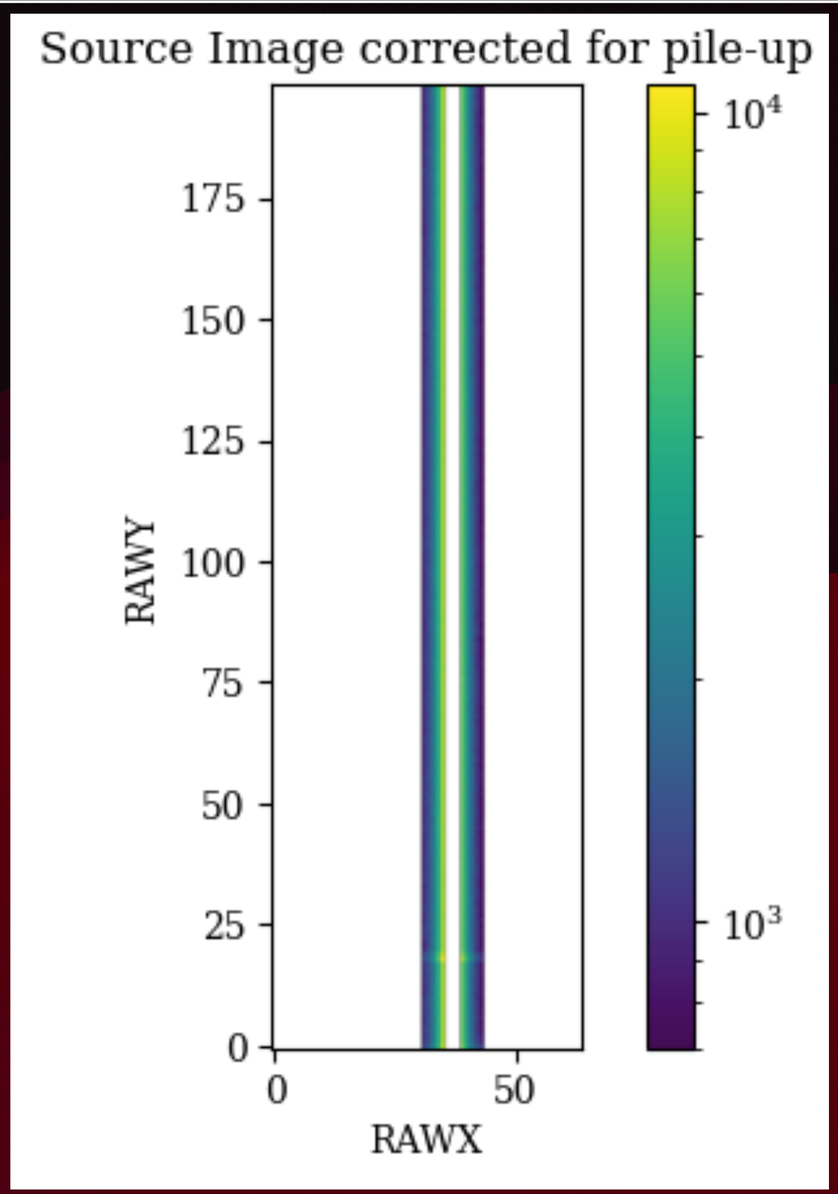




V. More complex analysis for bright sources

A. Pile-up correction

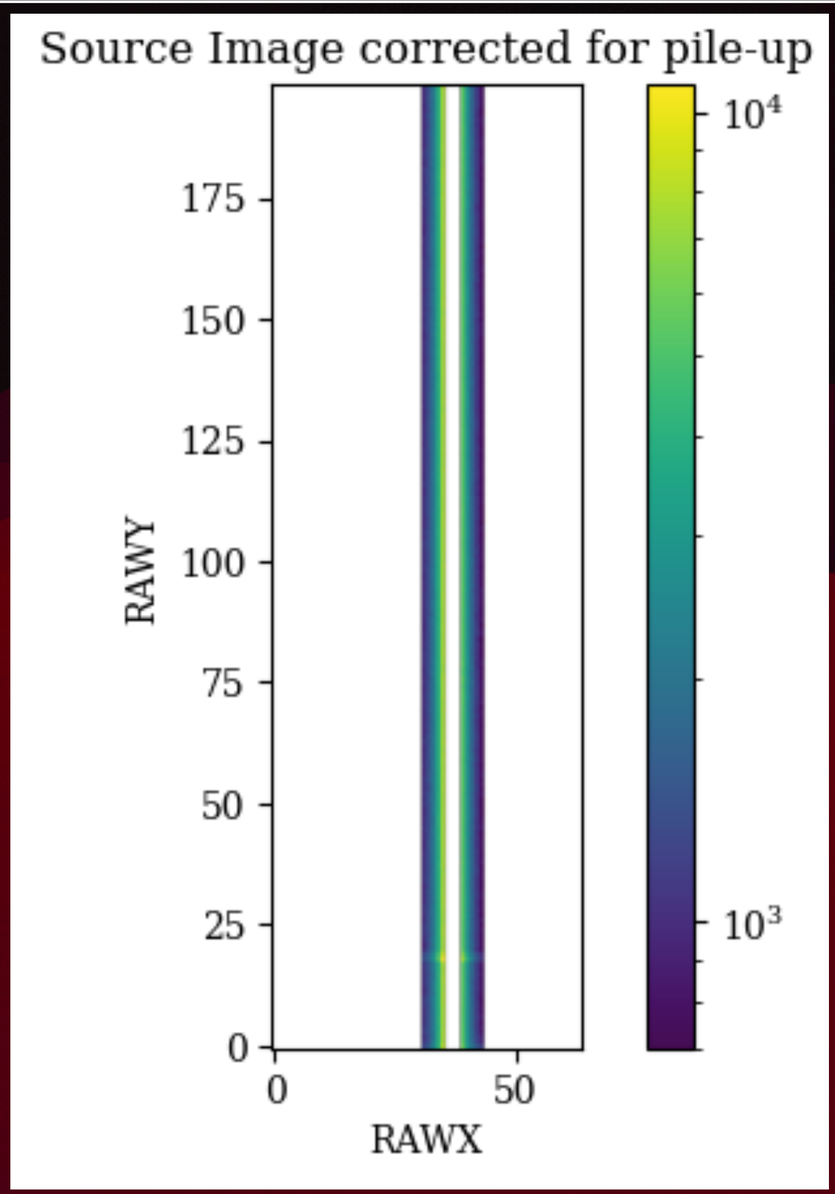
- Extract source region (background region not needed for such bright sources)
- Checks with `epatplot`
- If observed to model fractions $\neq 1$, excise centremost columns of the PSF



V. More complex analysis for bright sources

A. Pile-up correction

- Extract source region (background region not needed for such bright sources)
- Checks with `epatplot`
- If observed to model fractions $\neq 1$, excise centremost columns of the PSF
- Find the right balance between S/N and piled-up pixels removal



V. More complex analysis for bright sources

V. More complex analysis for bright sources

B. Correct for energy scale effects

V. More complex analysis for bright sources

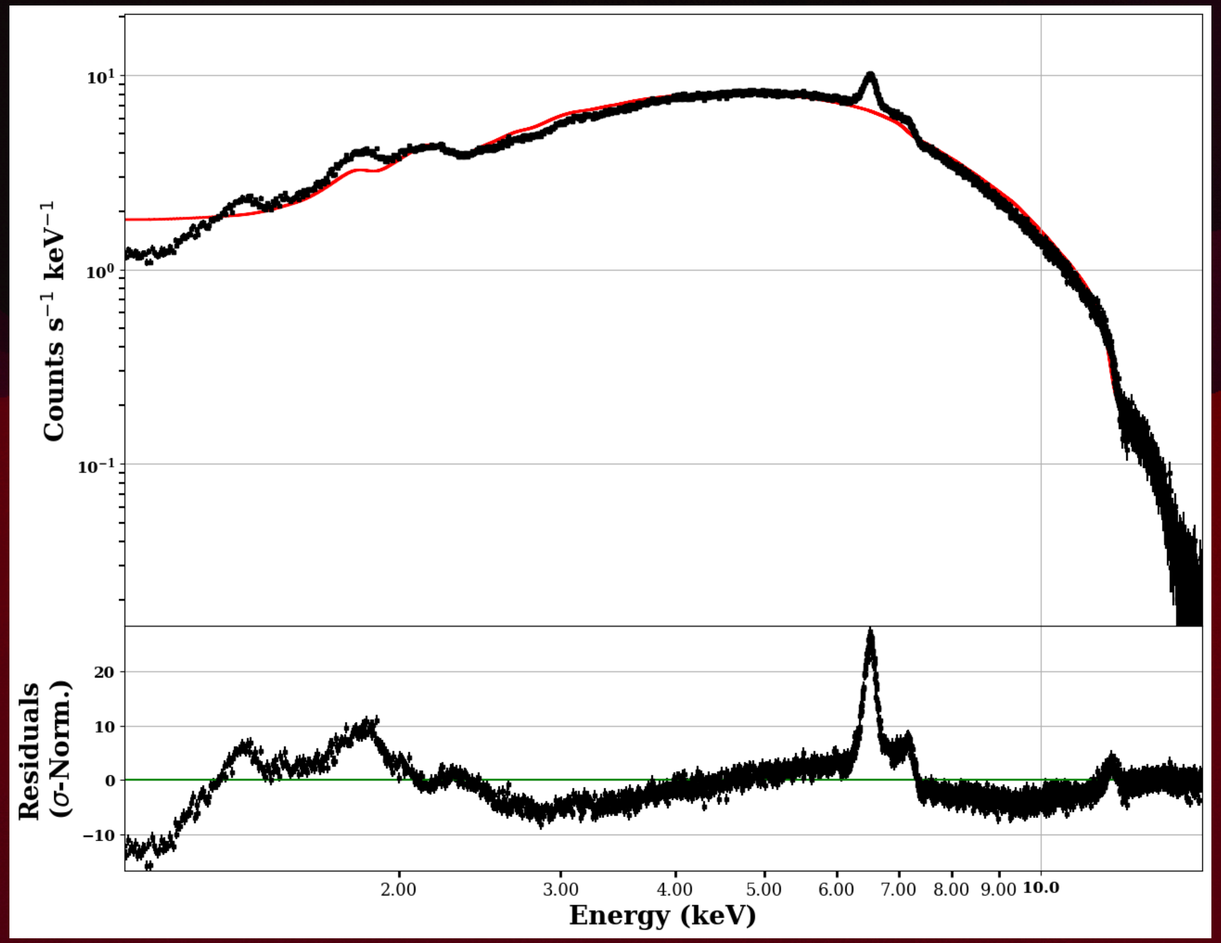
B. Correct for energy scale effects

- `evselect`, `arfgen` and `rmfgen` to extract a spectrum with corresponding ARF/RMF

V. More complex analysis for bright sources

B. Correct for energy scale effects

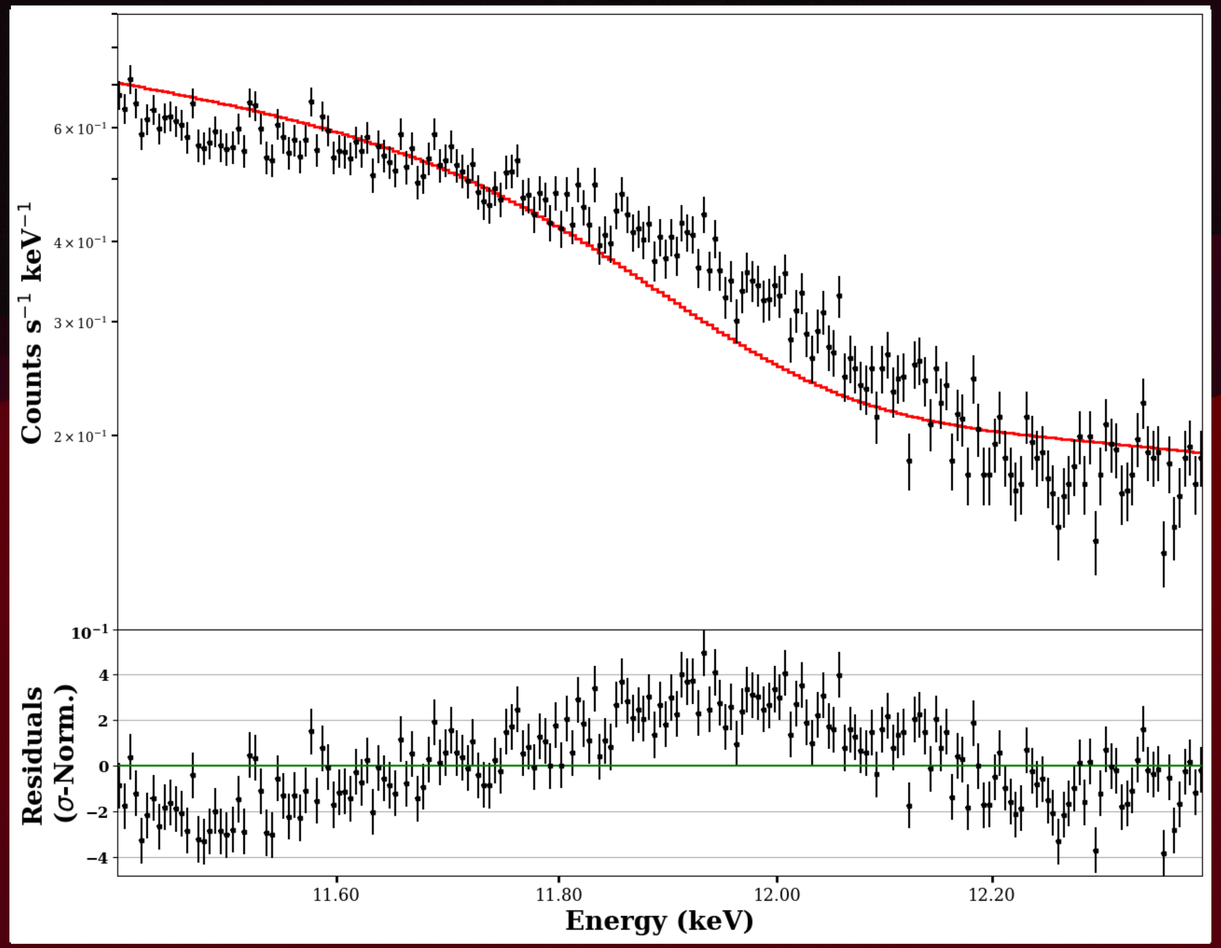
- `evselect`, `arfgen` and `rmfgen` to extract a spectrum with corresponding ARF/RMF
- Fit a simple model



V. More complex analysis for bright sources

B. Correct for energy scale effects

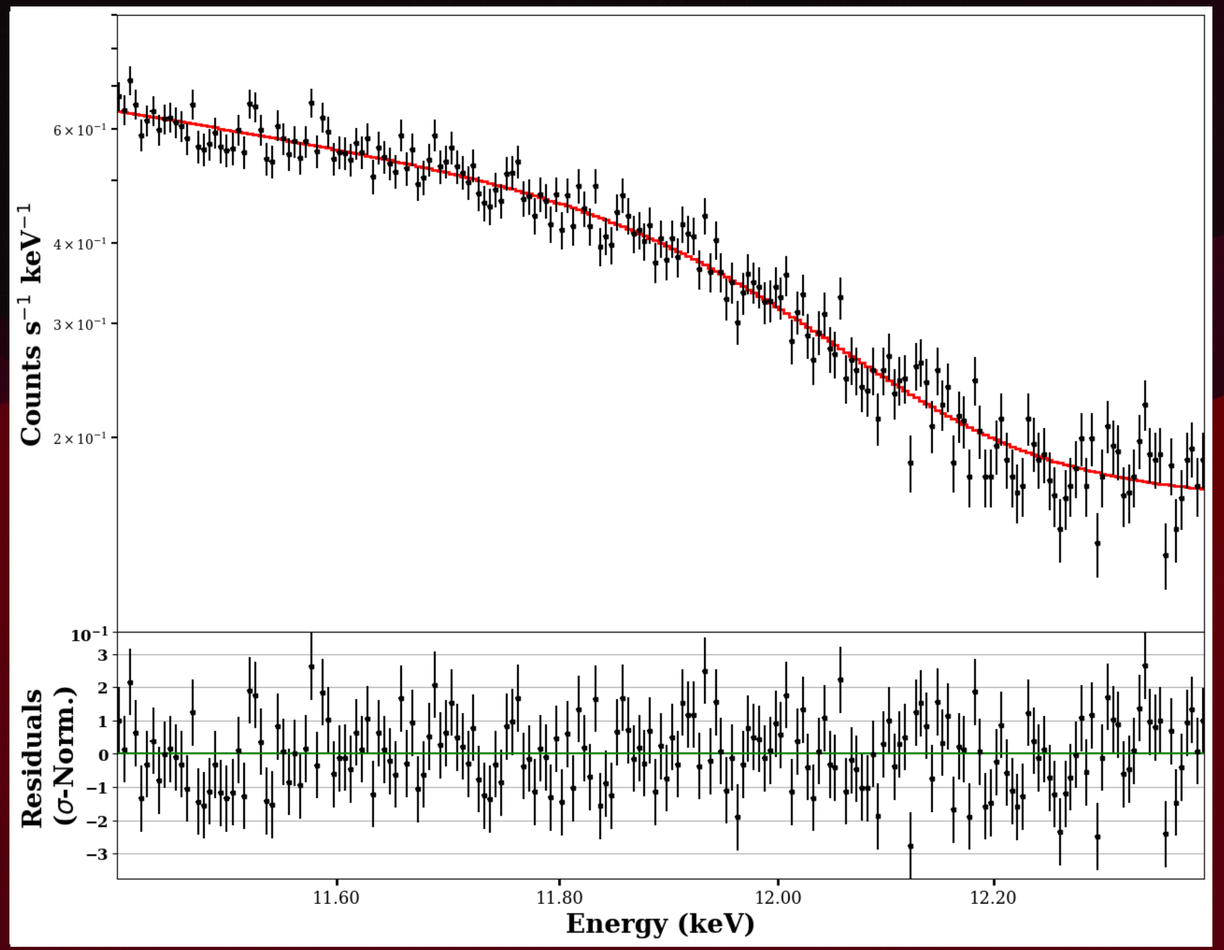
- `evselect`, `arfgen` and `rmfgen` to extract a spectrum with corresponding ARF/RMF
- Fit a simple model
- Check residuals at instrumental edges at 1.8 keV (Si K), 2.2 keV (Au M) and 11.9 keV (AuL)



V. More complex analysis for bright sources

B. Correct for energy scale effects

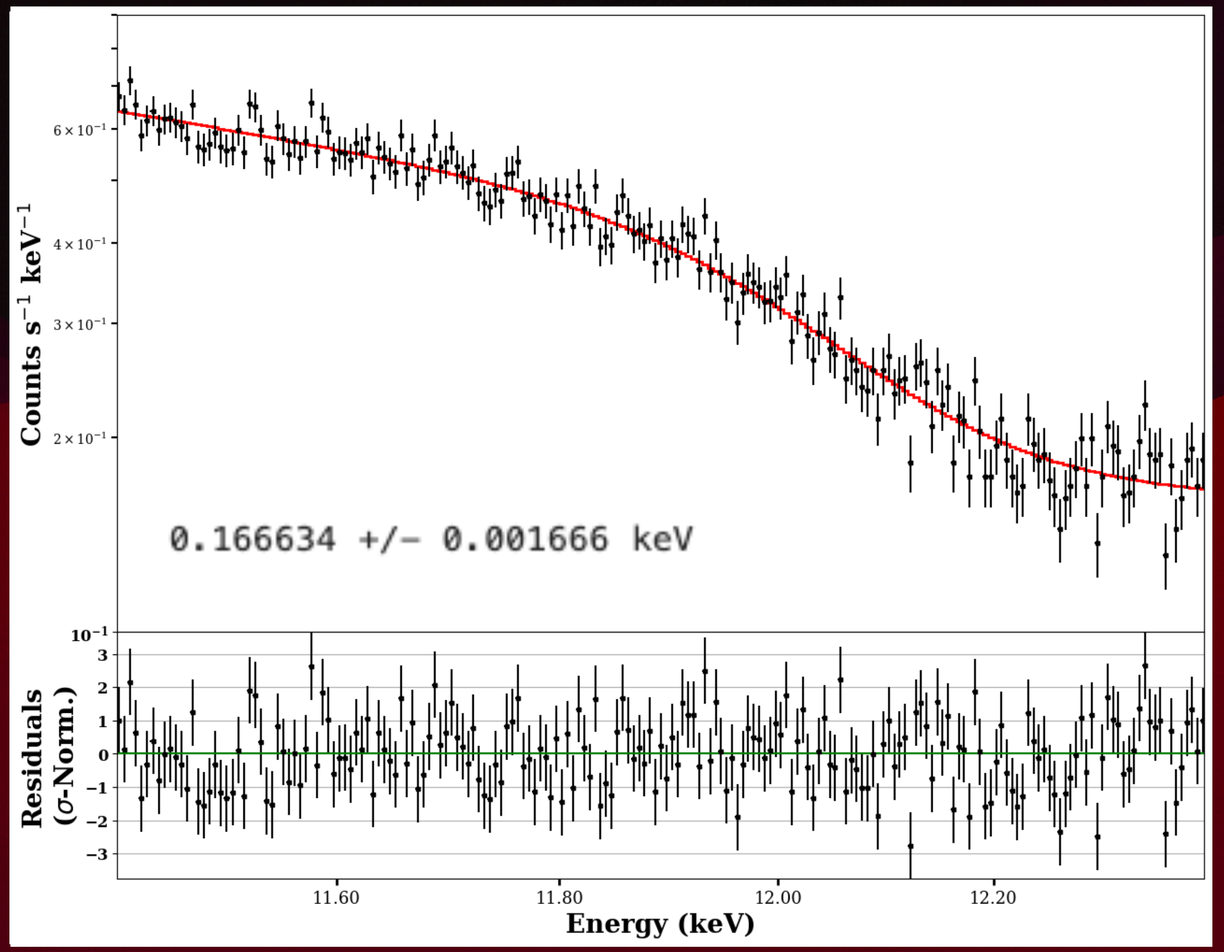
- `evselect`, `arfgen` and `rmfgen` to extract a spectrum with corresponding ARF/RMF
- Fit a simple model
- Check residuals at instrumental edges at 1.8 keV (Si K), 2.2 keV (Au M) and 11.9 keV (AuL)
- If energy shift, add a gain with your favourite data analysis software



V. More complex analysis for bright sources

B. Correct for energy scale effects

- `evselect`, `arfgen` and `rmfgen` to extract a spectrum with corresponding ARF/RMF
- Fit a simple model
- Check residuals at instrumental edges at 1.8 keV (Si K), 2.2 keV (Au M) and 11.9 keV (AuL)
- If energy shift, add a gain with your favourite data analysis software
- `evenergyshift` to shift all events by the energy offsets you found



VI. Extract time-averaged lightcurve

VI. Extract time-averaged lightcurve

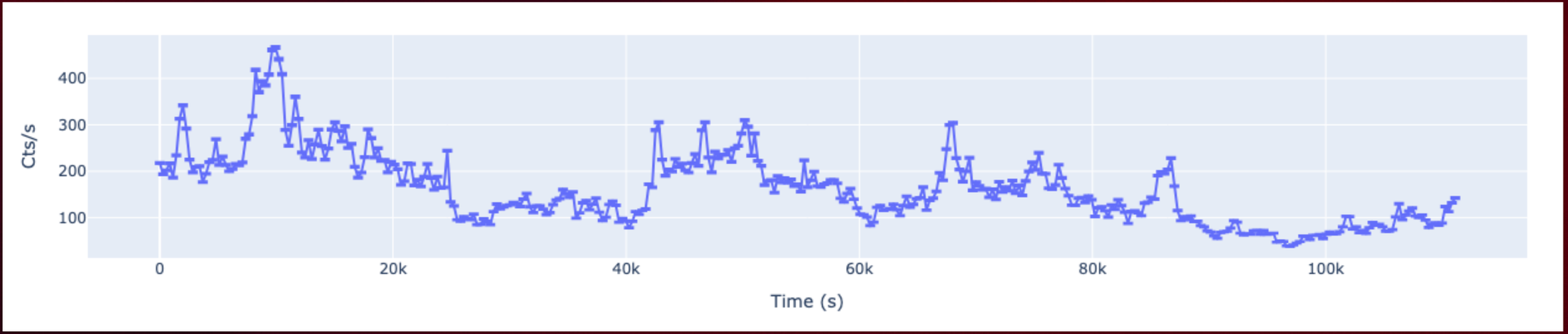
- `evselect` on the final cleaned event file with chosen filters to extract the lightcurve

```
# Define a SAS filter expression to derive the lightcurve

pn_pattern    = 4         # pattern selection
pn_pi_min     = 500.     # Low energy range eV
pn_pi_max     = 10000.    # High energy range eV
pn_time_binsize = 283.44     # time binsize

in_LCSRCFile    = lc_dir+f'/PN_raw_srclc_bin{round(pn_time_binsize)}sec.fit'
```

VI. Extract time-averaged lightcurve

- `evselect` on the final cleaned event file with chosen filters to extract the lightcurve
- `epiclccorr` for futher corrections

```
# Define a SAS filter expression to derive the lightcurve

pn_pattern    = 4         # pattern selection
pn_pi_min     = 500.     # Low energy range eV
pn_pi_max     = 10000.    # High energy range eV
pn_time_binsize = 283.44     # time binsize

in_LCSRCFile    = lc_dir+f'/PN_raw_srclc_bin{round(pn_time_binsize)}sec.fit'
```



VII. Extract time-averaged spectrum

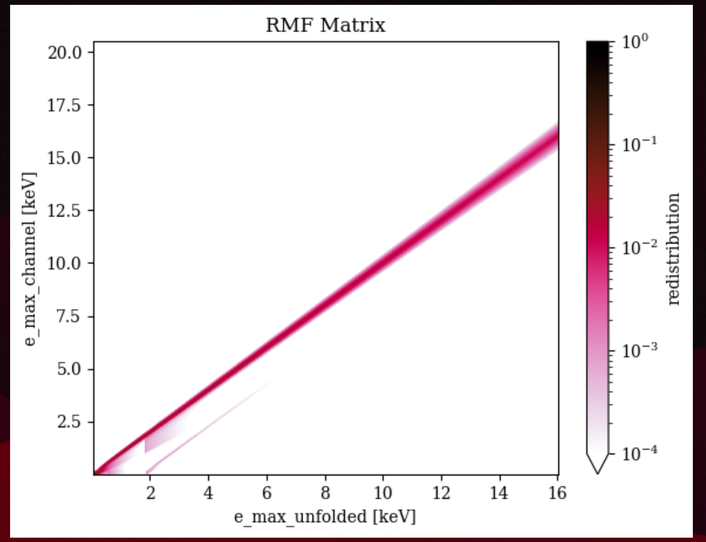
VII. Extract time-averaged spectrum

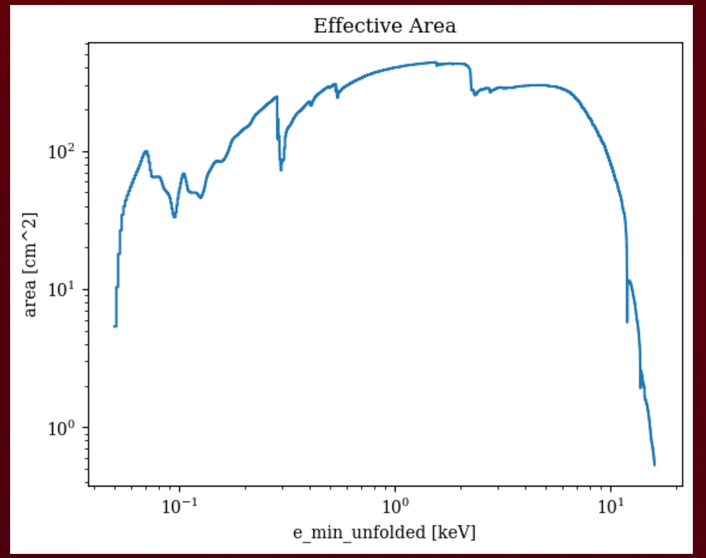
- `evselect` on the final cleaned event file with chosen filters to extract the spectrum

```
# SAS Command
cmd        = "evselect" # SAS task to be executed

# Arguments of SAS Command
expression = f'(FLAG==0) && (PATTERN<=4) && (RAWX in [{rawX1src}:{rawX3src}] || RAWX in [{rawX4src}:{rawX2src}])'  # event filter expression
inargs     = [f'table={final_clean_evfile}','withspectrumset=yes',f'spectrumset={spectrumset}',
              'energycolumn=PI', 'spectralbinsize=5', 'withspecranges=yes', 'specchannelmin=0', 'specchannelmax=20479',f'expression={expression}']
```

VII. Extract time-averaged spectrum

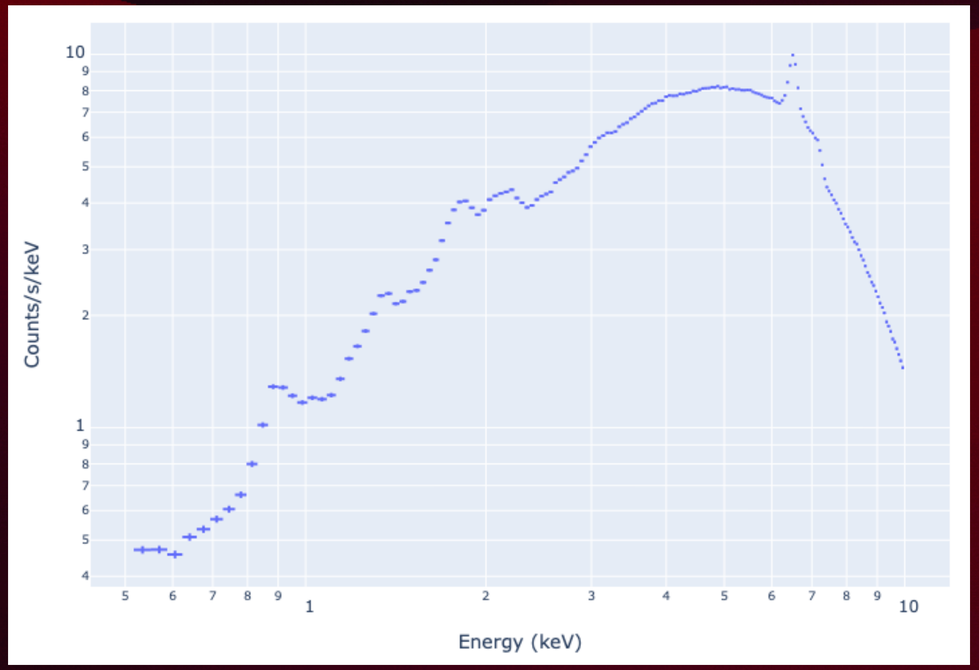
- `evselect` on the final cleaned event file with chosen filters to extract the spectrum
- `arfgen` and `rmfgen` for ARF and RMF extraction

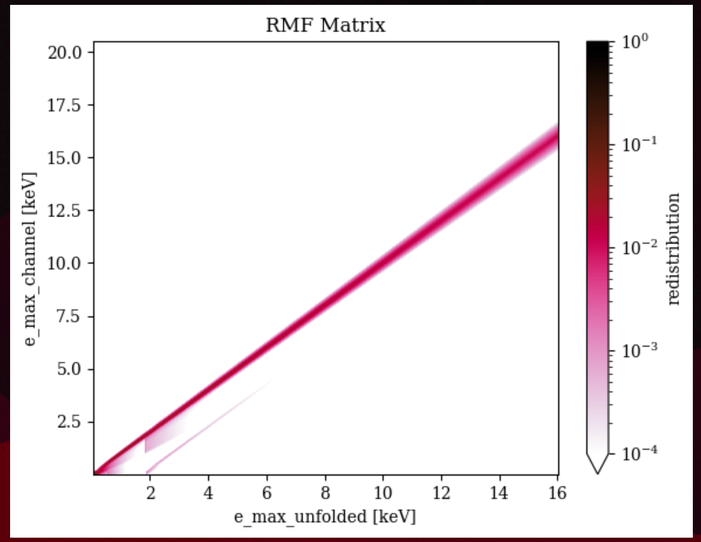


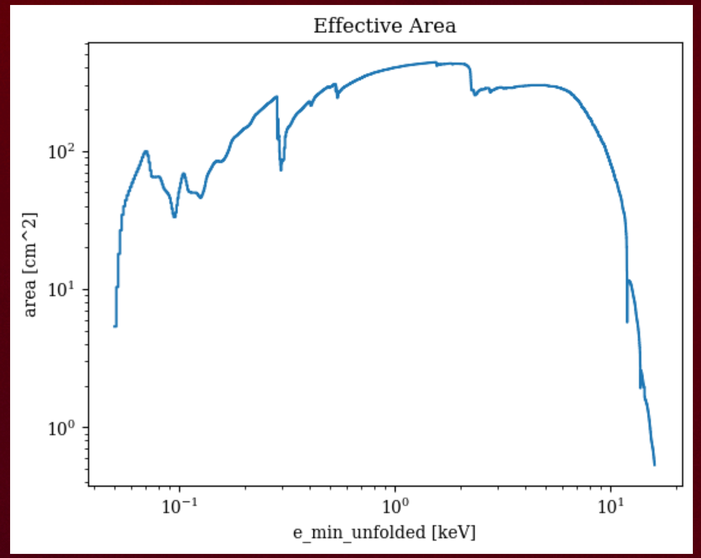


VII. Extract time-averaged spectrum

- `evselect` on the final cleaned event file with chosen filters to extract the spectrum
- `arfgen` and `rmfgen` for ARF and RMF extraction
- Use your favourite data analysis software for plotting spectrum and perform spectral analysis







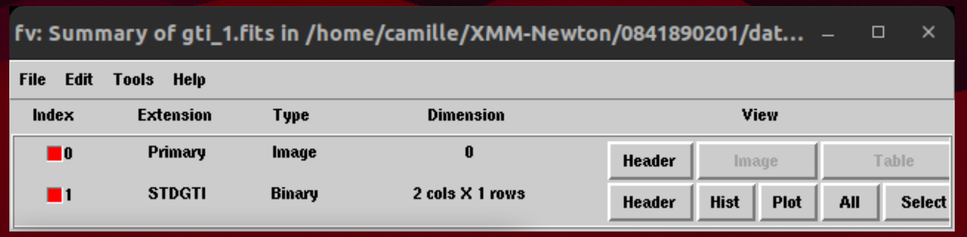
VIII. Use GTIs

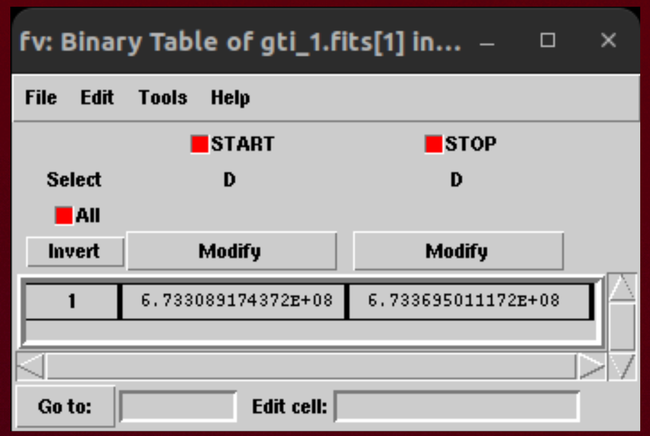
VIII. Use GTIs

- Create GTI fits files (e.g. with `tabgtigen`)

```
time_min = tstart + 20000.0
time_max = tstart + 80000.0

# SAS tabgtigen Command
cmd        = "tabgtigen" # SAS task to be executed

# Arguments of SAS tabgtigen Command
expression = f'(TIME >= {time_min}) && (TIME <= {time_max})'  # event filter expression
inargs     = [f'table={final_clean_evfile}',f'gtiset={gti_file}',f'expression={expression}']

print("   Filter expression to use: "+expression+" \n")
print("   SAS command to be executed: "+cmd+", with arguments; \n")
inargs

w(cmd, inargs).run()
```

```
   Filter expression to use: (TIME >= 673328775.717167) && (TIME <= 673388775.717167)
```





VIII. Use GTIs

- Create GTI fits files (e.g. with `tabgtigen`)
- Add GTI files/keywords in `evselect`, `arfgen` and `rmfgen`

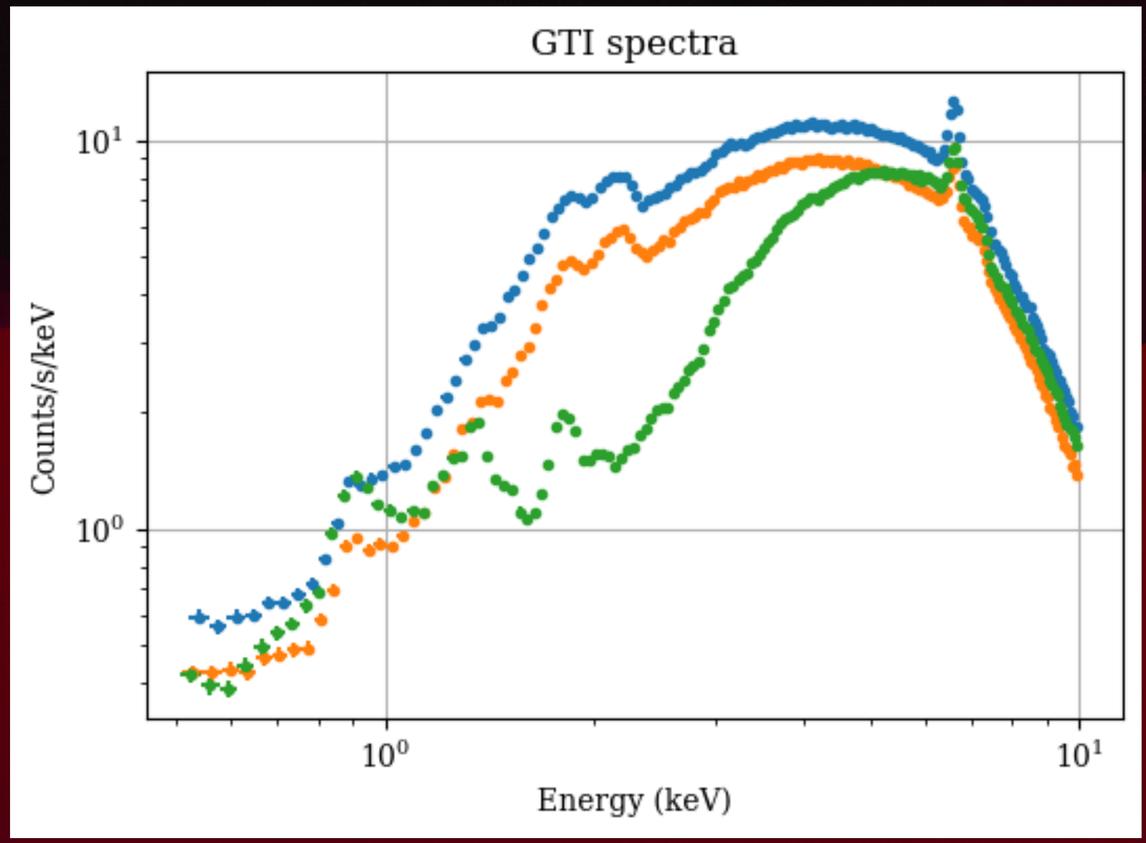
```
# SAS Command
cmd        = "evselect" # SAS task to be executed

# Arguments of SAS Command
expression = f'(FLAG==0) && (PATTERN<=4) && (RAWX in [{rawX1src}:{rawX3src}] || RAWX in [{rawX4src}:{rawX2src}]) &&(gti({gti_file},TIME))' # event filter expression
inargs     = [f'table={final_clean_evfile}','withspectrumset=yes',f'spectrumset={spectrumset_gti}',
              'energycolumn=PI', 'spectralbinsize=5', 'withspecranges=yes', 'specchannelmin=0', 'specchannelmax=20479',f'expression={expression}']

print("   Filter expression to use: "+expression+" \n")
print("   SAS command to be executed: "+cmd+", with arguments; \n")
inargs

w(cmd, inargs).run()
```

VIII. Use GTIs

- Create GTI fits files (e.g. with `tabgtigen`)
- Add GTI files/keywords in `evselect`, `arfgen` and `rmfgen`
- Same procedure as time-averaged spectrum extraction

VIII. Use GTIs

- Create GTI fits files (e.g. with `tabgtigen`)
- Add GTI files/keywords in `evselect`, `arfgen` and `rmfgen`
- Same procedure as time-averaged spectrum extraction
- You can do a loop and extract several gti spectra all at once



Conclusion

With this example, you learned how to:

- Work with sources taken with EPIC-pn in timing mode
- How to remove piled-up pixels (can be applied to imaging mode)
- Extract lightcurves and spectra
- Use Datalabs with a known case study
- Do all your science from data extraction to spectral analysis with Python

Conclusion

With this example, you learned how to:

- Work with sources taken with EPIC-pn in timing mode
- How to remove piled-up pixels (can be applied to imaging mode)
- Extract lightcurves and spectra
- Use Datalabs with a known case study
- Do all your science from data extraction to spectral analysis with Python

Special thanks to Esin Gülbahar for her Jupyter Notebook adaptation of my bash scripts which was crucial for writing this Notebook and to Simon Dupourqué for his help with Python, statistical methods for X-ray spectral fitting and jaxspec