Introduction to XMM-Newton & SAS

Aitor Ibarra on behalf of XMM-Newton team

Telespazio UK for ESA

XMM-Newton SOC

SCI-SOX

09.10.2025

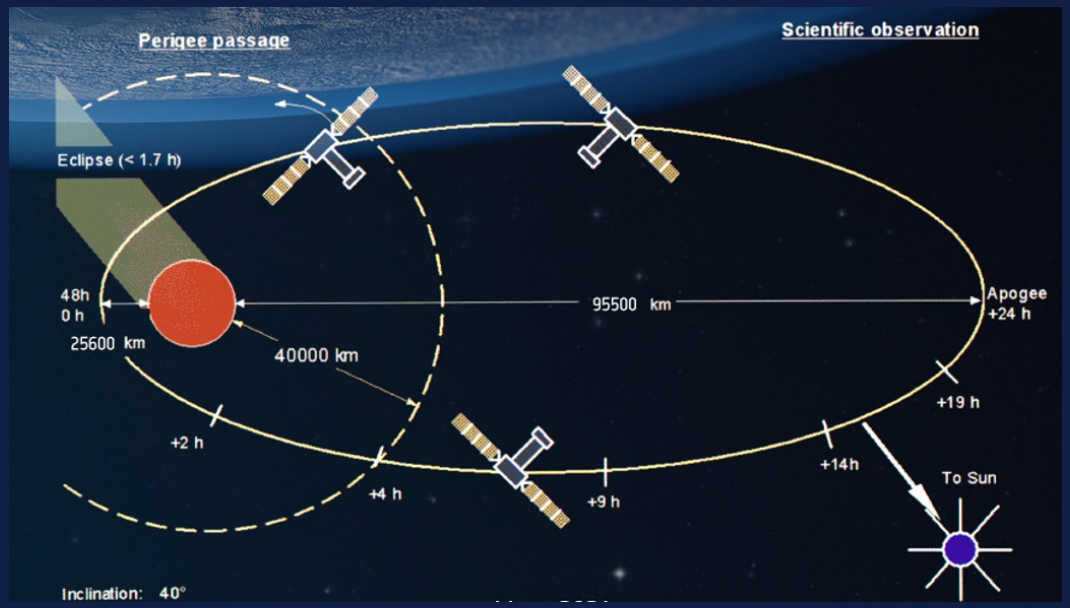
Outline

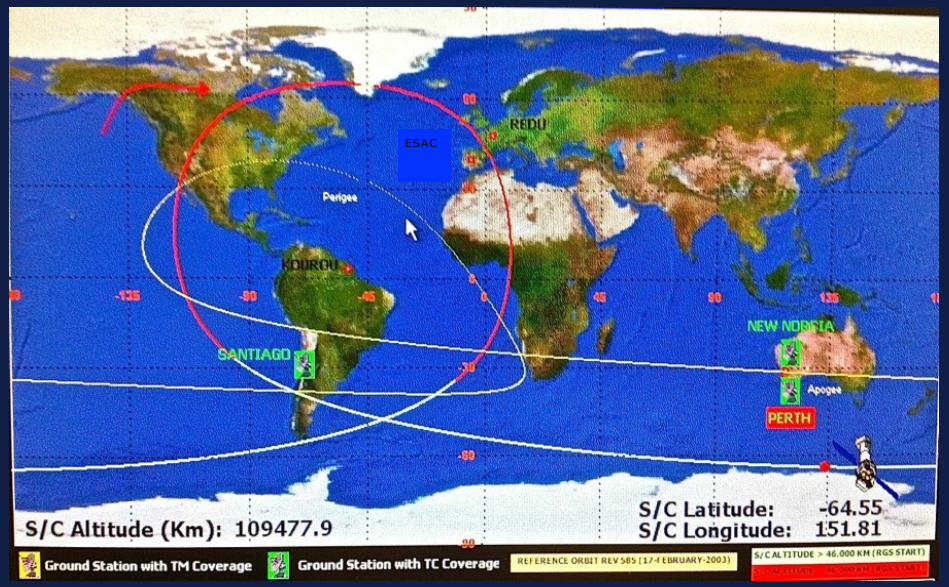
- XMM-Newton: Fast facts
- Instruments Specifications and Observing modes
- Science Analysis System: SAS
- Data reduction & Grand Scheme
- Standard procs → [pn/em/rgs]proc metatasks
- Bad pixels, Good Time Intervals (GTIs) and Data Filtering (Flaring particle background)
- Source Detection
- SAS Installation and set-up
- Getting started I: the Observation Data File (ODF) + odfbrowser
- Getting started II: the Calibration Current File (CCF) and the Calibration Index File (CIF)
- Retrieving XMM-Newton data →PPS products → XMM-Newton Science Archive (XSA)

XMM-Newton: Fast Facts

Launch:	December 10th 1999
Launch vehicle:	Ariane 5
Dimensions:	Mass: 3764 kg, 10 m long 16 m wide
Orbit:	Elliptical, 48 hour
Payload:	Service module carrying three 'mirrors modules' at its forward broader end Focal plane assembly housing the X-ray cameras and detectors at its other extremity 3 Wolter telescopes with 58 mirrors each
Mission phase:	**Operational**
Mission extension:	Approved: 2023-2026 Recommended: 2027-2029

XMM-Newton: The orbit





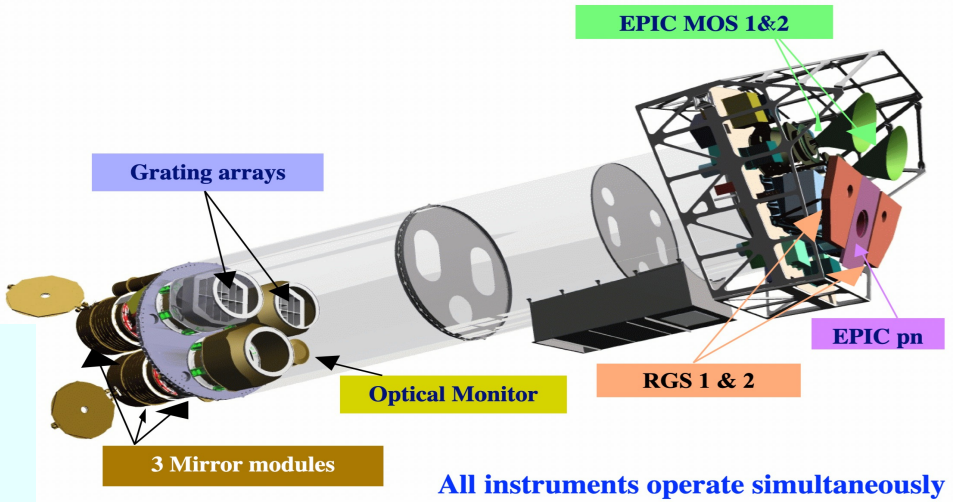
~ 48 hour orbit

Perigee alt. ~ 25000 km
Apogee ~ 95000 km

~8 hours in the Van Allen Radiation Belt,
~40 uninterrupted hours science per revolution

XMM-Newton: The spacecraft

European Photon Imaging Cameras (EPIC)	Two MOS CCD cameras (**EPIC-MOS**) One pn CCD camera (**EPIC-pn**) FOV: 30 arcmin Energy range: 0.15 – 15 keV
Reflection Grating Spectrometers (RGS)	Two Focal Plane Cameras & associated Grating Arrays Energy range: 2.5 to 0.35 keV FOV ± 2.4 arcmin in cross-dispersion
Optical Monitor (OM)	Ritchey-Chretien Optical/UV telescope. Range: 170-650 nm FoV: 17x17 arcmin



XMM-Newton: Ground Segment

MOC, MISSION OPERATIONS CENTER

SOC, SCIENCE OPERATIONS CENTER

Scientific community

SOC main tasks

- Science operations (Mission Planning)
- Run yearly call for proposals
- Instrument monitorng & callibration
- Data processing
- Data archive & distribution
- Community support

XMM-Newton: AO cycle

Run Call for proposals

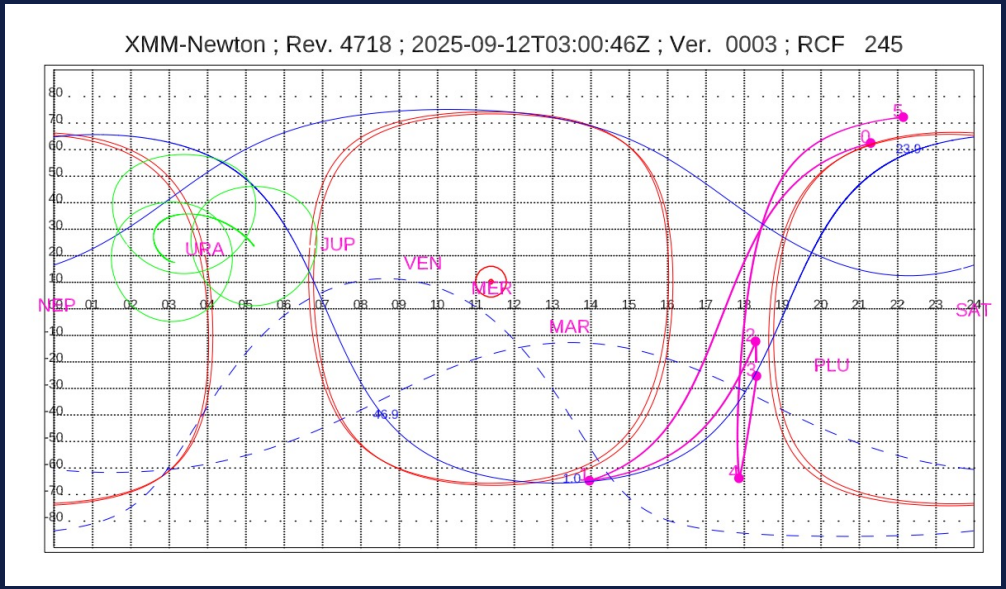
- Open call for proposals (yearly)
- Provide submission interface & tools
- Receive observing requests, (attend last minute questions)
- Organise the TAC, get priority & strategy for each proposal
- Implement TAC resolutions
- Provide feedback to Pis

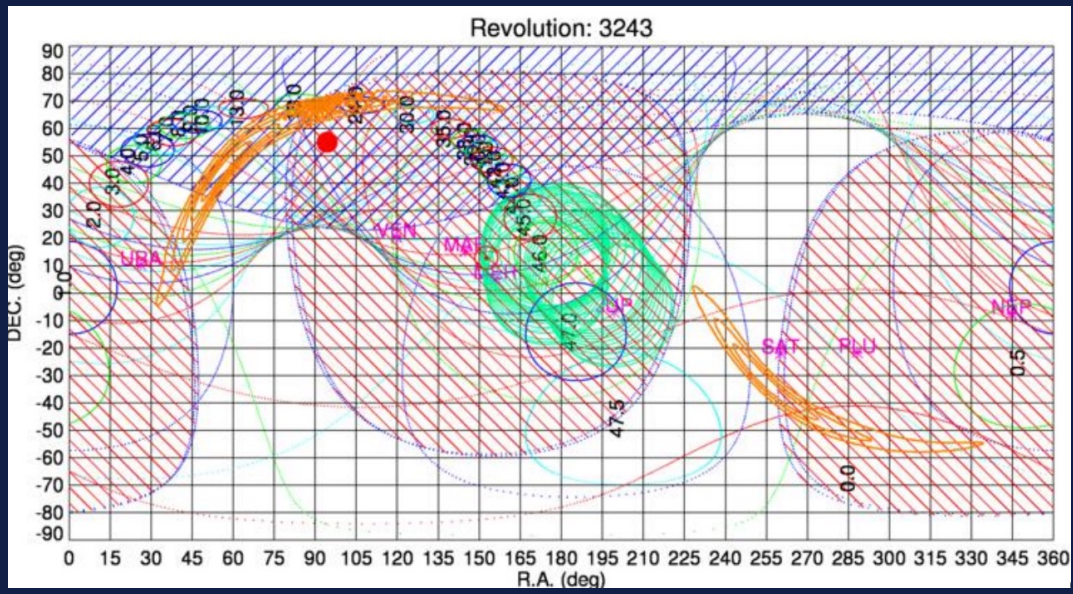
Implement observing plan

- Account for target priorities, visibilities & strateg.
  - > Allow for disruptions due to TOO observations
  - > Arrange coordinated & calibration campaigns
- Build Short Term plan: timeline (POS) generation
- Track progress: Replan failed observations
- Carry-over to next AO incompleted ones

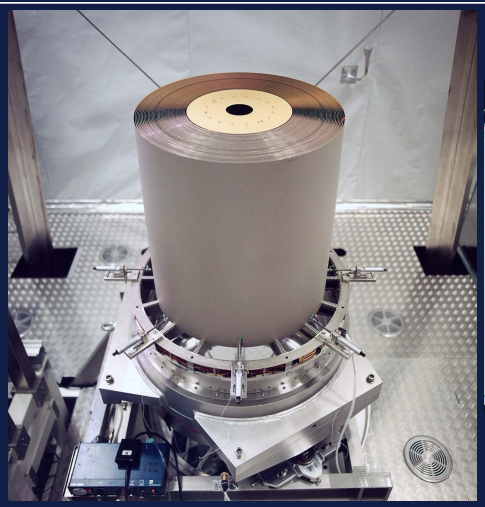
XMM-Newton: Scheduling & Mission Planning

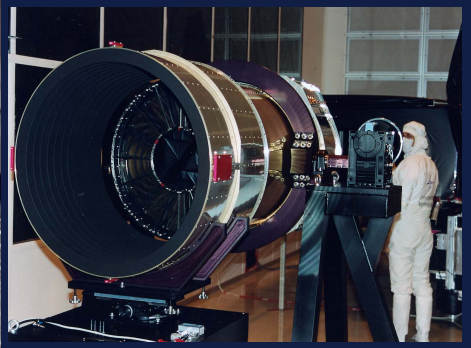
- The observations (targets) for each XMM-Newton revolution **are planned about three weeks in advance** (depending on the Targets of Opportunity received!).
- The instruments are **set up** to optimize the scientific return.

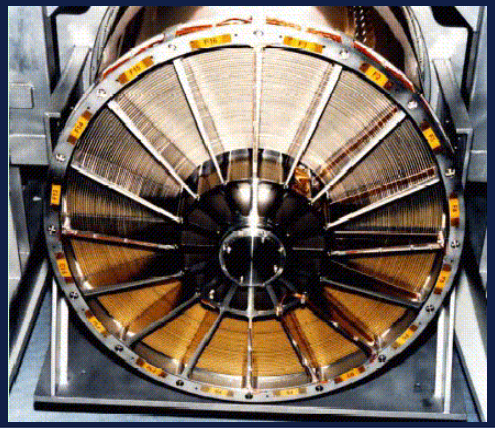


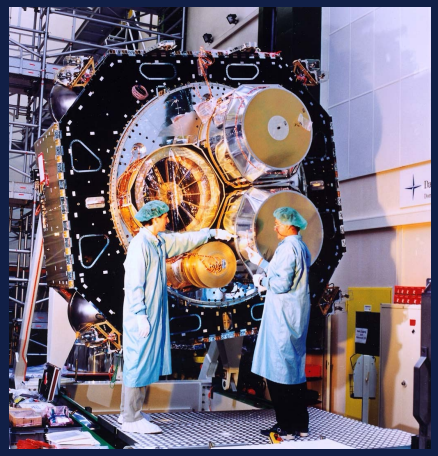


X-Ray Mirrors







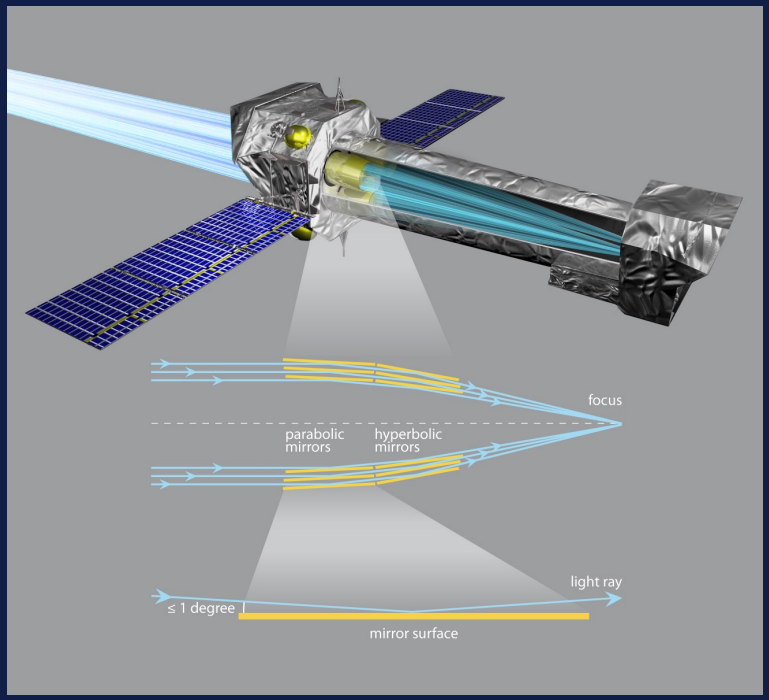


- 3 Wolter telescopes
- 58 nested mirror shells
- Au-coated electroformed nickel shells
- Au-M edge (~ 2.3-3.4 keV) with fine structure

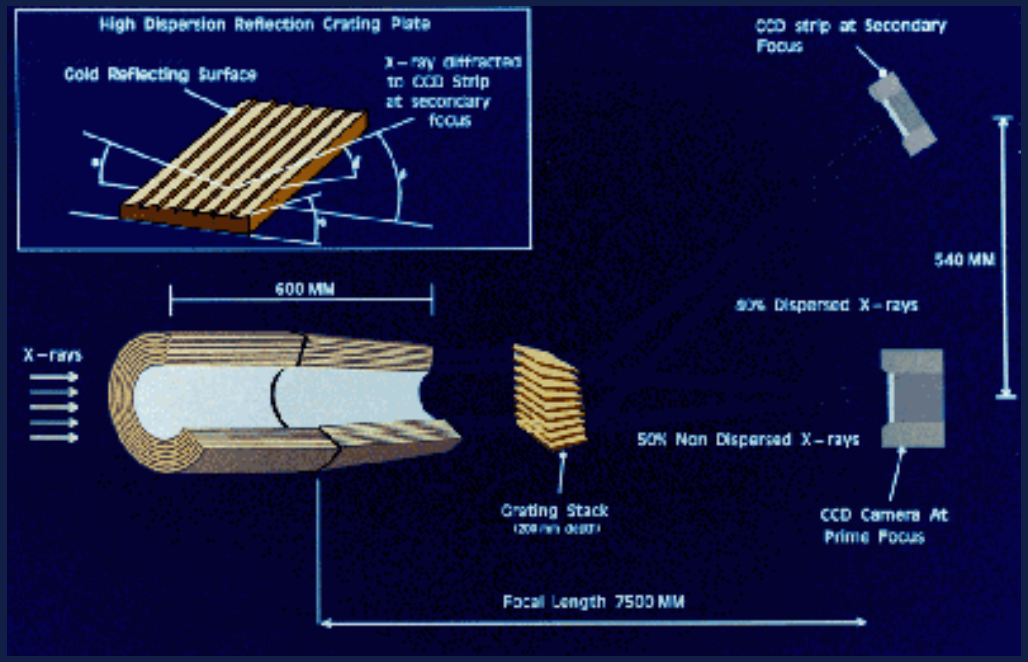
- Held by spider with 16 arms
- 16 scatter wings in PSF
- Triangular PSF (dependent on XRT)

X-ray telescopes

3 Wolter telescopes with 58 nested shells each



light path of X-rays through XMM-Newton

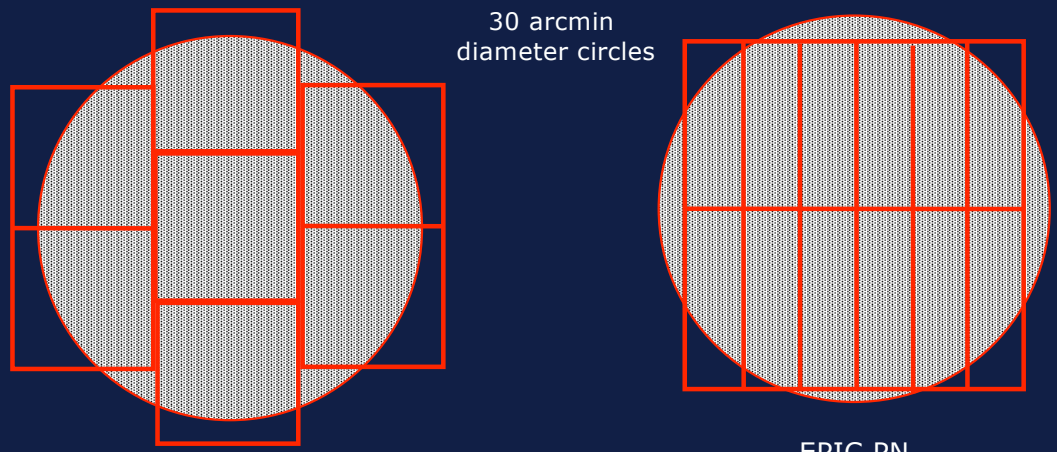


The imaging cameras: EPIC

- **2 MOS cameras (7 CCDs each)**
- Stacked to fit focal curvature
- cameras with different orientation (by 90°)
- Front illuminated: gatting structure in X-ray light path
- 1 CCD: 600 x 600 pixels;
  40 mm/pix, 1.1 arcsec/pix

- **1 pn camera (12 CCDs on one wafer)**
- Minimizing gaps between CCDs
- Back illuminated
- Larger pixel size ⇒ faster readout
- 1 CCD: 200 x 64 pixels; 150 mm/pix, 4.1 arcsec/pix
- Reading all 64 columns in parallel

Comparison of focal plane organisation of EPIC MOS and pn cameras

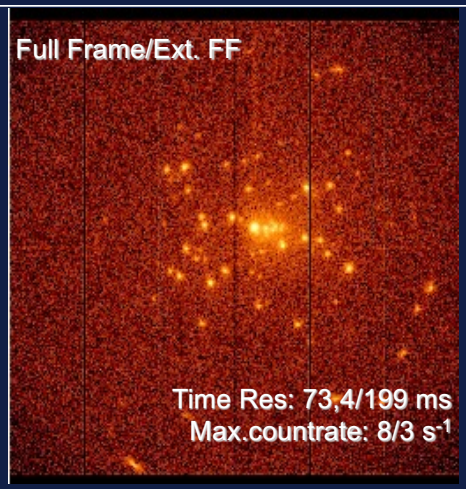


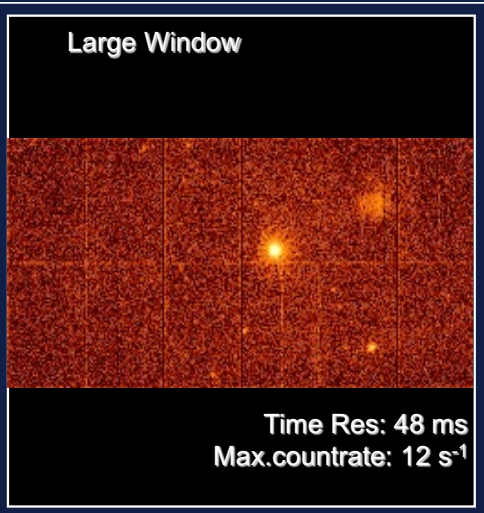
EPIC MOS
7 CCDs each 10.9 x 10.9 arcminutes

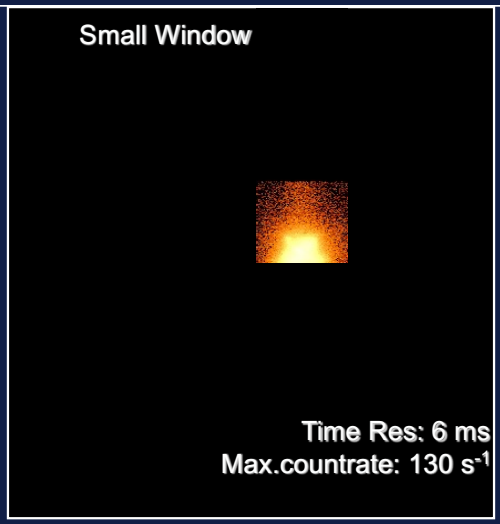
EPIC PN
12 CCDs each 13.6 x 4.4 arcminutes

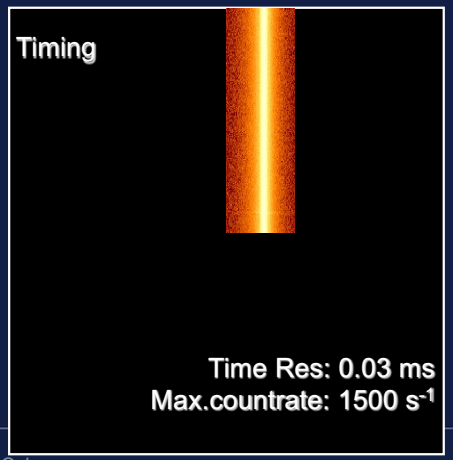
Filters required to reject light from longer (visible) wavelengths

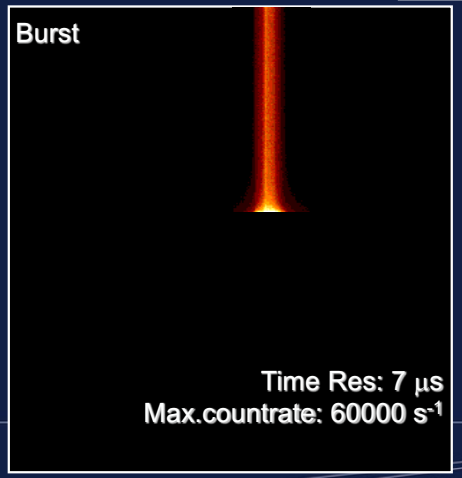
EPIC pn observing modes



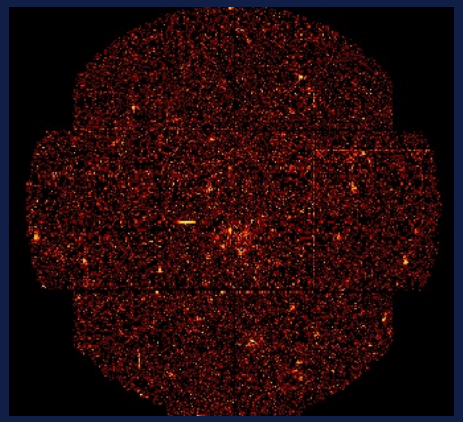








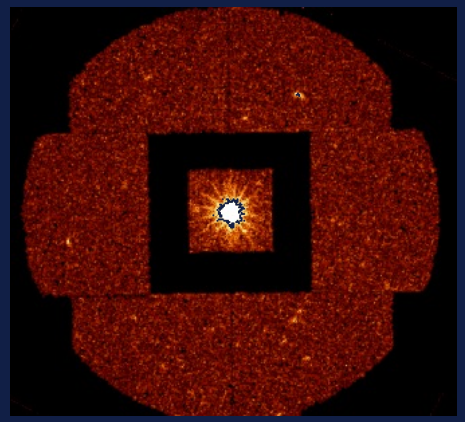
EPIC MOS observing modes



Full Frame

Time Resol: 2.6 s

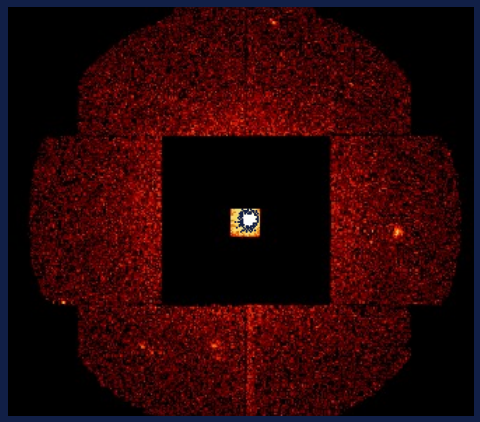
Max. countrate:
0.7 cts/s



Large Window

Time Resol.:
0.9 s central CCD
2.7 s outer CCDs

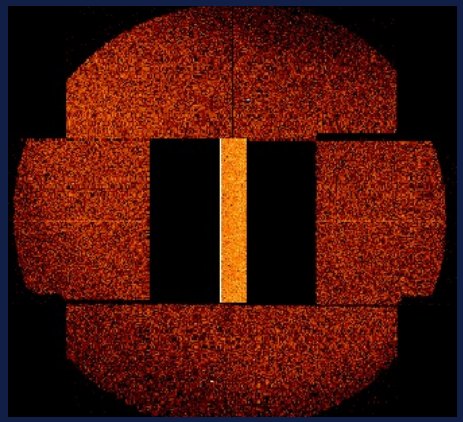
Max.countrate:
1.3/s



Small Window

Time Res.:
0.3 s central CCD
2.7 s outer CCDs

Max. countrate:
5/s



Timing

Time res.:
1.8 ms central CCD
2.6 s outer CCDs

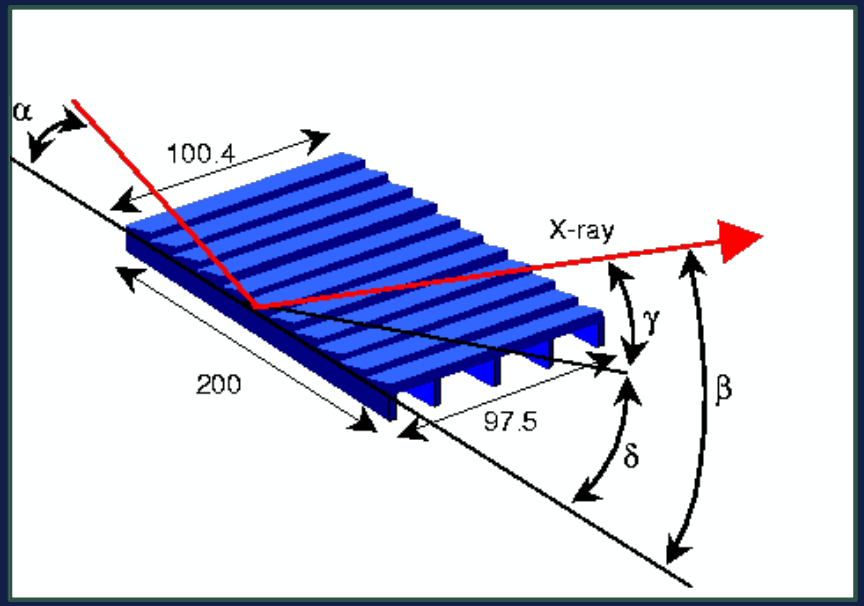
Max. countrate:
100/s

RGS Characteristics

- 2 Reflection Grating Spectrometers behind two of the XRTs
- each **RGS**:
  - a Reflection Grating Array (**RGA**),
  - a Focal plane Camera (**RFC**),
- RGAs continuously in the light-path, non switchable
- High sensitivity and resolution in [0.35-2.5]keV ([5-35]Å)
- Line rich region containing K-shell transitions of low-Z abundant elements (C, N, O, Ne, Si) and the diagnostically important L-shell Fe transitions
- FOV is ± 2.4 arcmin in cross-dispersion
- **The RGS operates simultaneously with EPIC.**

RGS spectra of sources in the FOV are obtained for every XMM observation.

Dispersion Geometry

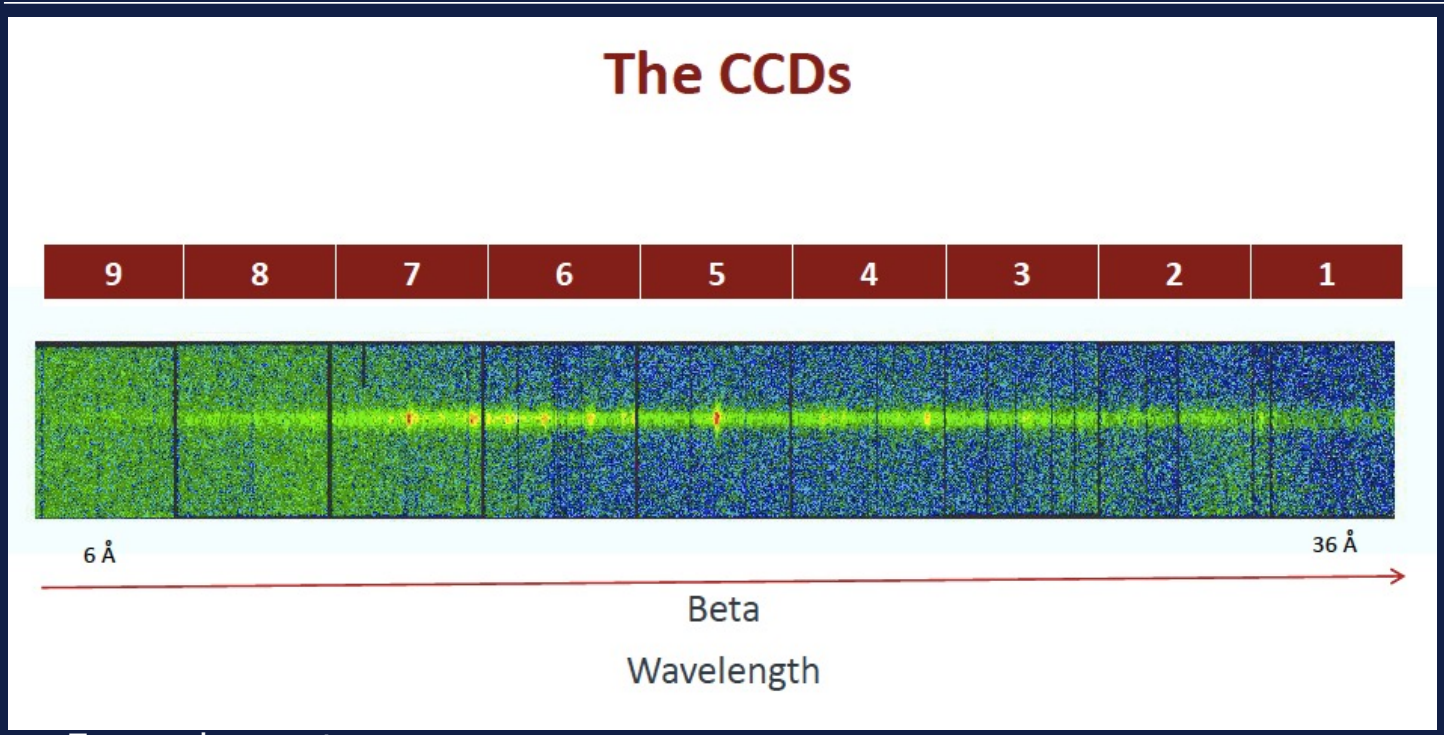


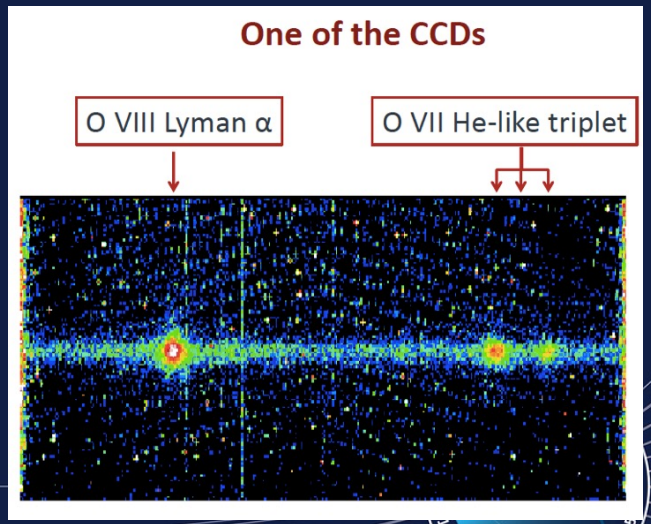
Dispersion equation:

$$\cos\beta = \cos\alpha + m\lambda / d$$

$$\lambda = (\cos\beta - \cos\alpha)\, d / m$$

The RGS CCDs





For each event:

- Time
- Position on the detector
- Energy

The Optical Monitor (OM)

OM is a 30 cm telescope, f/12.7 modified Ritchey Chrétien optics.

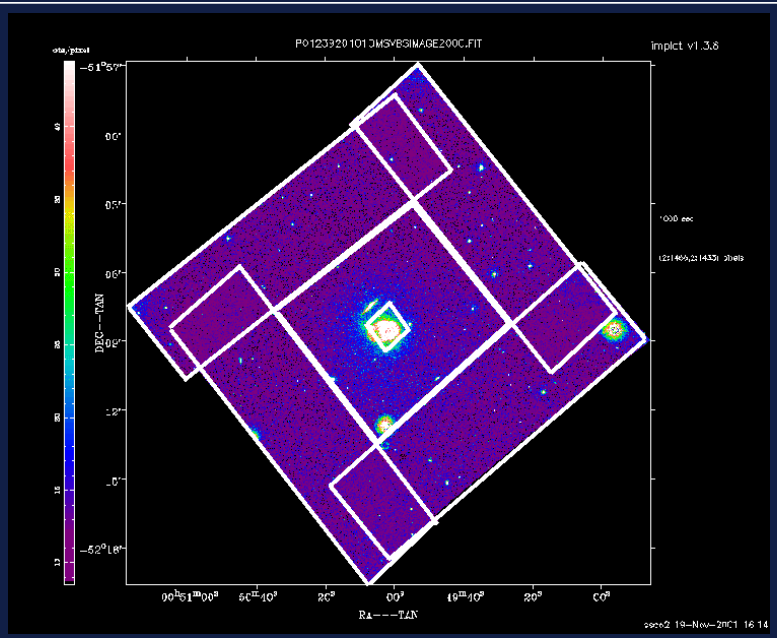
Simultaneous & co-aligned UV/optical observations of X-ray sources:

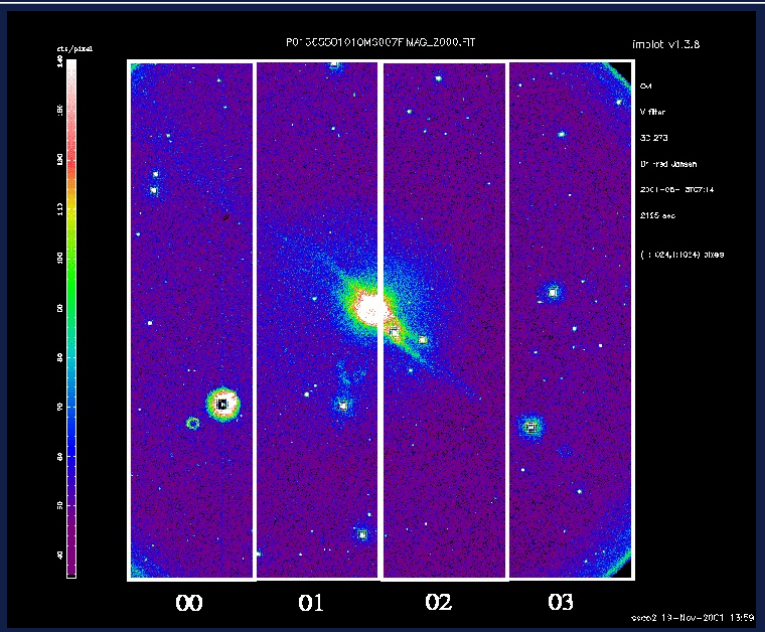
- Broad Band photometry
- Optical/UV spectra
- Fast timing photometry

Parameter	Typical Value / Range
Total bandwidth	160– 600 nm
Spectral bandwidth	160 – 550 nm
Sensitivity limit	23.5 mag
FOV	17’
PSF (FWHM)	1.6” – 2.3”

Parameter	Value
Timing resolution	0.5s
Spectral resolution	0.5/1.0 nm
Spatial resolution	0.5/2.0”
Brightness limit	m_v=7.4 mag

OM imaging and more…





Default: 5 consecutive windows cover entire FOV + continuous central high res. window

+ window in fast mode: time resolution of source
+ grism window: full spectral resolved source

Or: user defined windows (up to 5, 2 in fast mode)

Full-frame imaging: homogeneous sampling of whole FOV ⇒ surveys
> in low resolution (1024x1024 1” pixels)
> in high resolution (2048 x 2048 0.5” pixels)

+ grism: low spectral resolution of all sources in the FOV

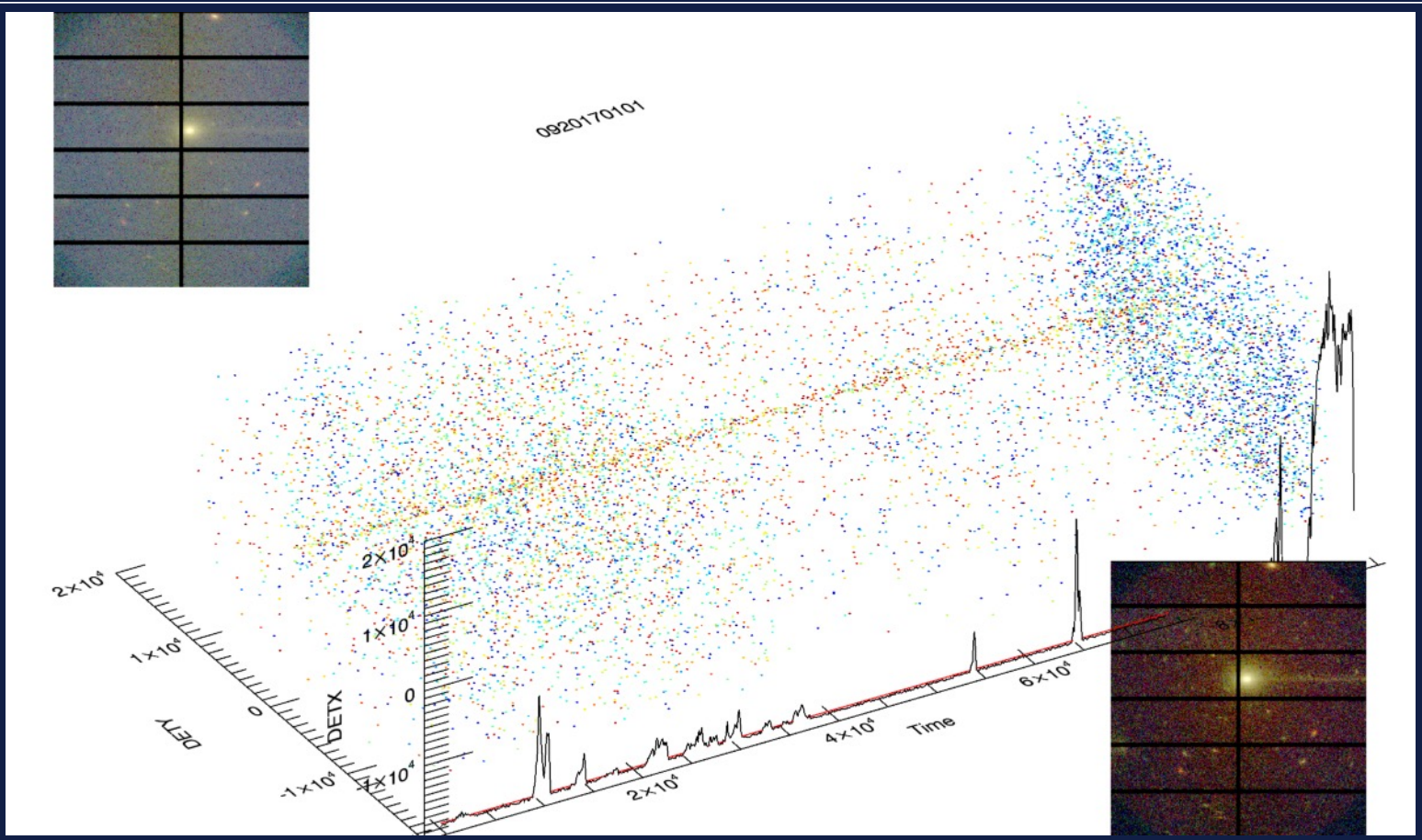
Science as projection I

The X-ray scientific products can be seen as **projections** onto the sub-spaces defined by the event physical quantities

- By collapsing *time* and *space*, one obtains an energy distribution function (***spectrum***) expressed in ***counts per energy bin***
- By collapsing *time* and *energy*, one obtains a 2-D ***image,*** *expressed* in ***counts per pixel.***
- By collapsing *space* and *energy*, one obtains an intensity ***time series*** *(light curve) expressed* in ***counts per time bin***

These scientific products are expressed in units that are ***in*directly related** to the intrinsic properties of celestial sources

Science as projection II



Courtesy: P. Rodriguez

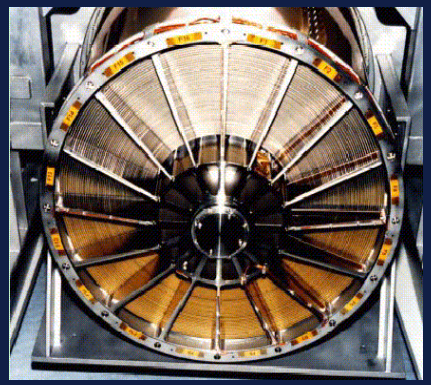
Basic of X-ray data

X-ray detectors are photon-counting → two main consequences:

- X-ray astronomy is an intrinsic Poissonian science

  Scientific products can have a few or even zero events in large ranges of their parameter spaces

- The “king” in the X-ray realm is the event, characterised by:
  - imaging — Where?
  - effective area — how many?
  - energy redistribution — Effects?
  - Gain/CTI — Which?
  - timing — When?
  - background — What?



Mirrors:
eff area
PSF
Astrometry
Vignetting

Filter: eff area

CCD:
QE, CTI, Gain, Redistribution, Astrometry

electronics:
Gain, Timing, Modes

What is SAS?

- The XMM-Newton Scientific Analysis System is a suite of programs (“tasks”) for dealing with data from all XMM-Newton Instruments

- It is written basically in C++ and Fortran 90/95. Perl and shell scripts constitute “metatasks”. It makes use of public libraries / programs like cfitsio, xmgrace, ds9

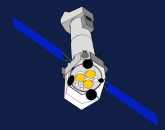
- It has been developed by ~ 30 programmers, working in 6 different countries along many, many years….

- A subset of the SAS is used as the core of the official Pipeline Processing System (PPS) for reducing the data to calibrated event lists, images, spectra, source lists (and much more)

Actual SAS version: SAS v22.1 – released on April 2025

New SAS v2**X**.0 to be released in 2026

Data reduction = calibration



Time calibration

Optics calibration

Image analysis:
• Exposure maps
• *Point Spread Function*

Timing analysis:
• Exposure maps
• *Point Spread Function*

ODF

ODF reduction

Scientific product extraction

Scientific product analysis

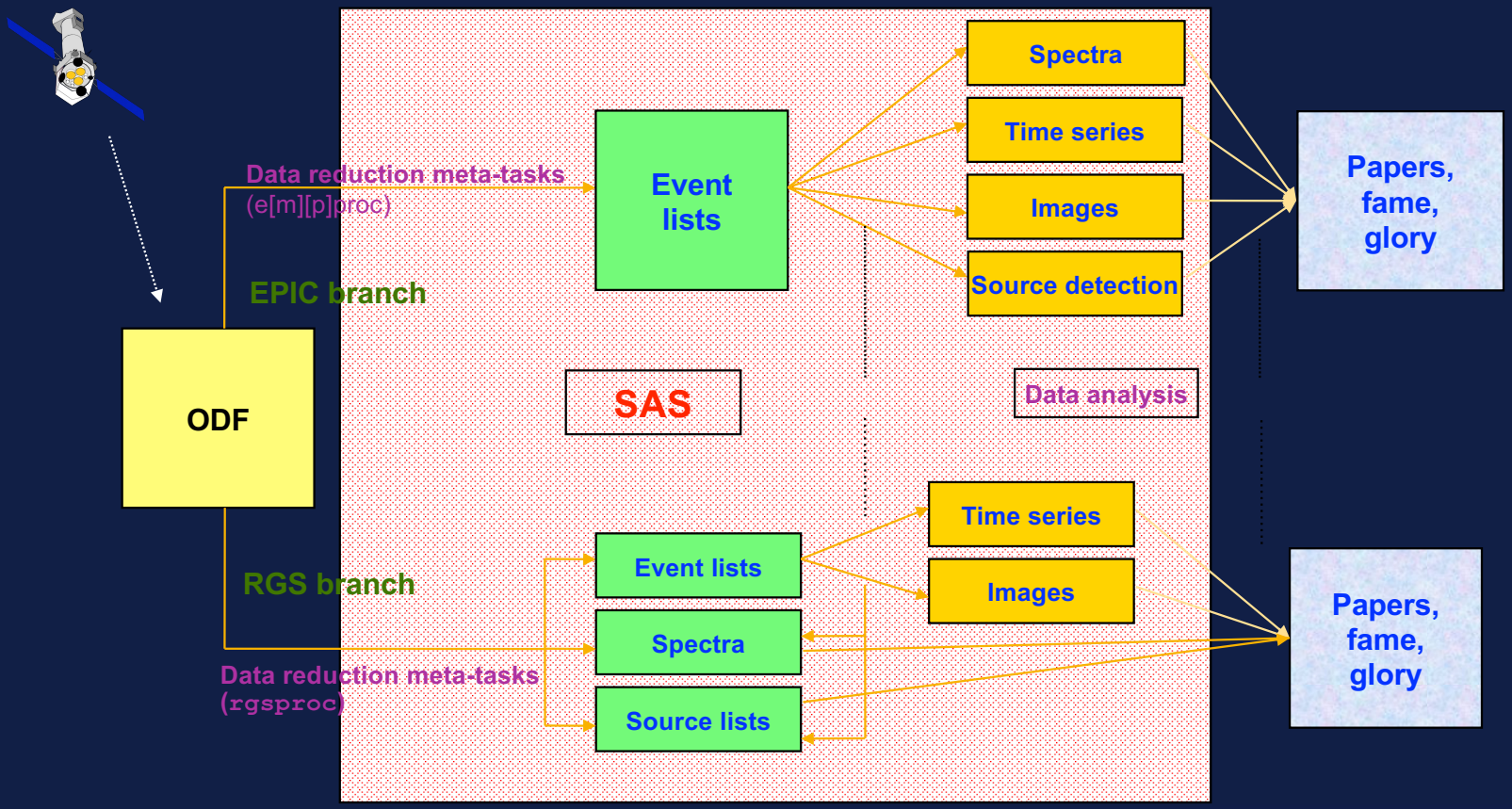
Science

Attitude calibration

Detector calibration

Energy spectra analysis:
• Effective area
• Redistribution matrix
• *Line Spread Function*

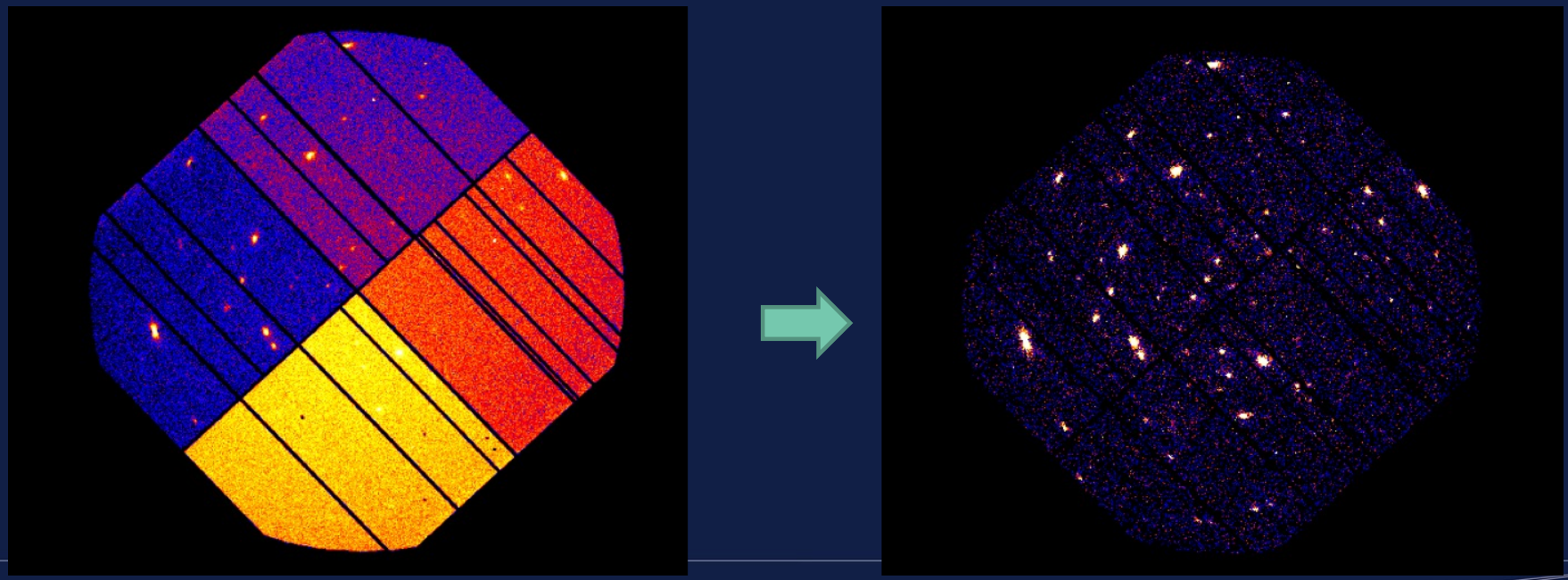
SAS Grand-Scheme



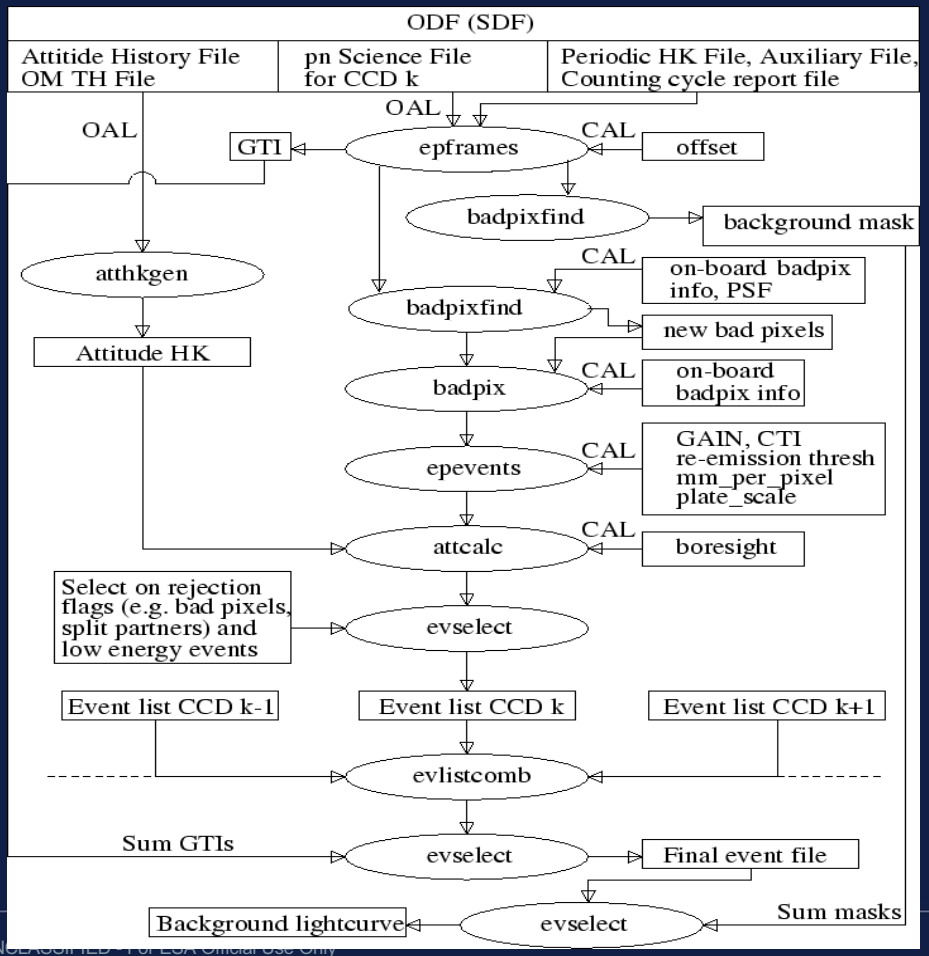
Reduction Scope

SAS does two things (to XMM data), that no other tool does:

- Applies calibrations to raw data
- Optimally screen/filter the data

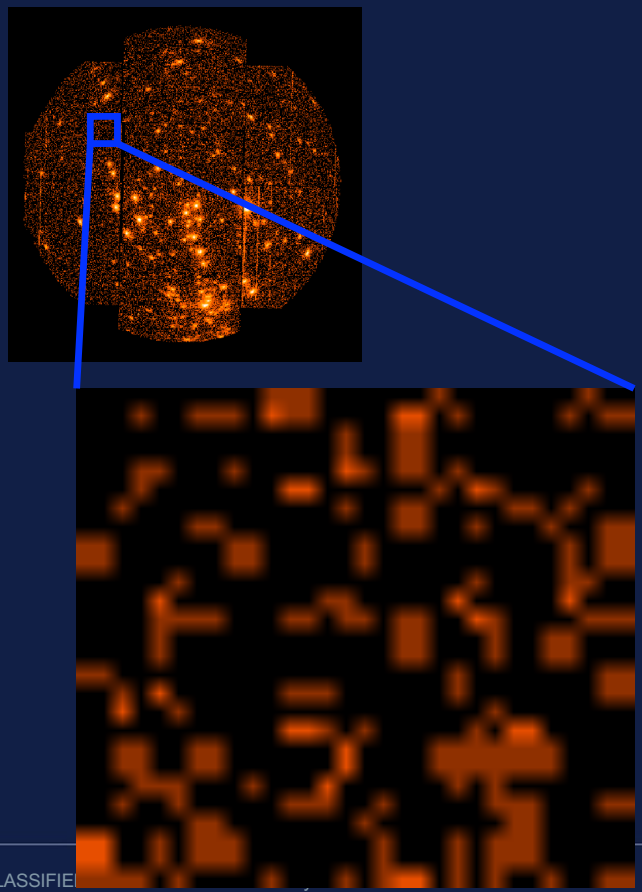


epproc reduction scheme



- **epframes** to process a CCD, exposure and datamode specific ODF file, creating the output raw event list and GTI data set
- **badpixfind** to find new bad pixels
- **badpix** to process the raw event list, adding the BADPIX extension
- **epevents** to process the event list file, flagging trailing events, performing split events pattern recognition, CTI and gain correction to create the calibrated event list
- **attcalc** to calculate the X and Y sky coordinates.
- **evlistcomb**, the CCD specific data sets are merged into a single event list.

Bad pixels



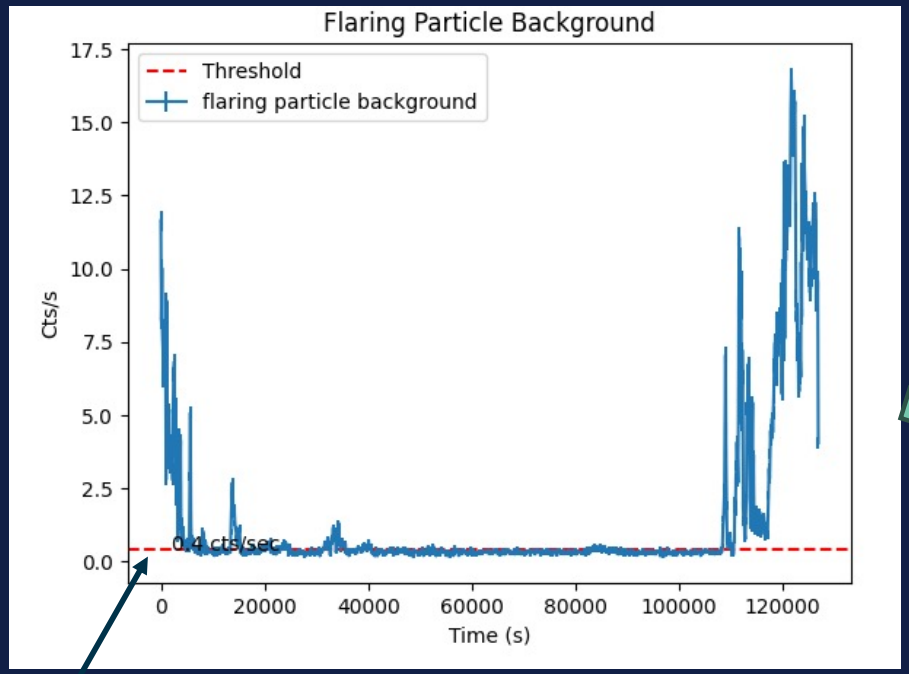
- **dead pixel:** no events are detected
- **hot pixel:** pixel "produces" ghost events very often
- **by default epicproc** will try to detect bad pixels for any imaging exposure.
- the **new bad pixels** are then used in the data reduction together with any other known (via the calibration files) bad pixels

Data Filtering

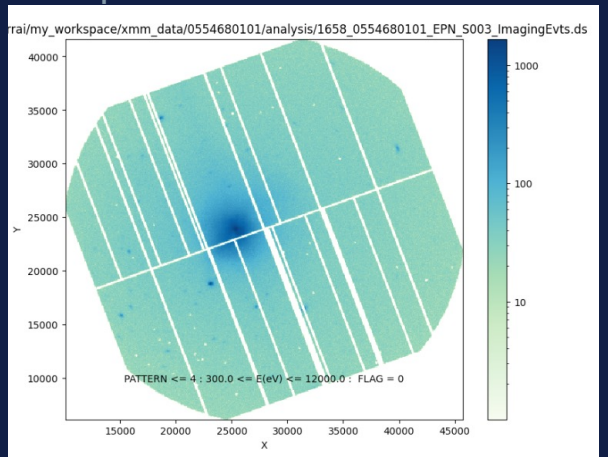
- by **default**, the event lists are filtered, and the filtered events are removed.
- the filter expression can be controlled by the user via parameter
  - **flagfilteredevents** == true:
    In this case all events will be retained, and a flag column will be set to indicate what events would have been removed.

Filtering [flaring particle background]

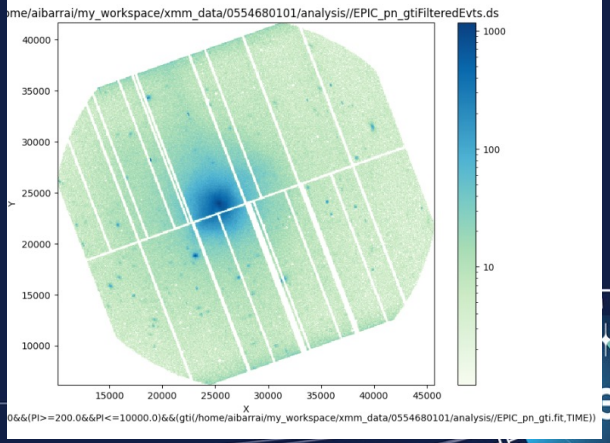
- How to filter EPIC event lists for flaring particle background → SAS Threads/Jupiter notebook



Single Events && PI [10-12keV]



Clean event file



0.4 cts/sec → Standard EPIC-pn thershold

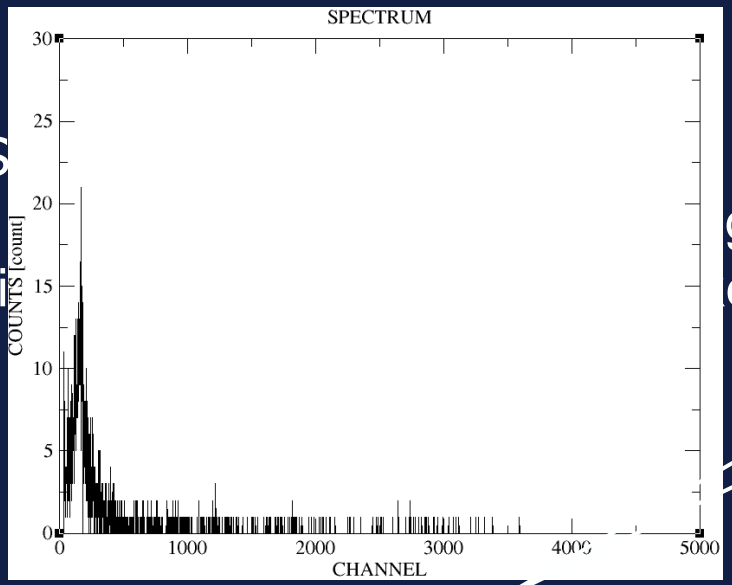
RGS → light curve → rgsfilter or eveselect to filtering event file

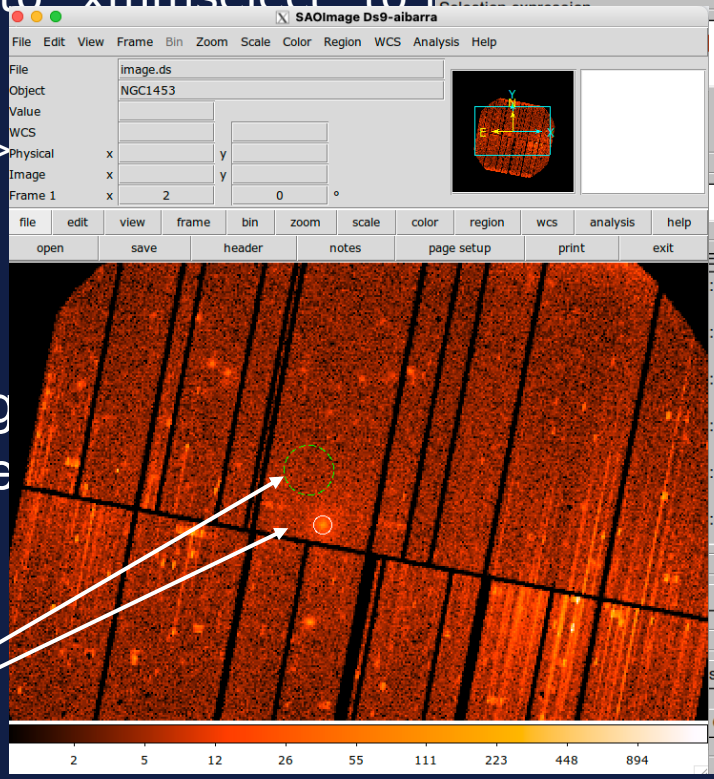
EPIC (pn) spectra

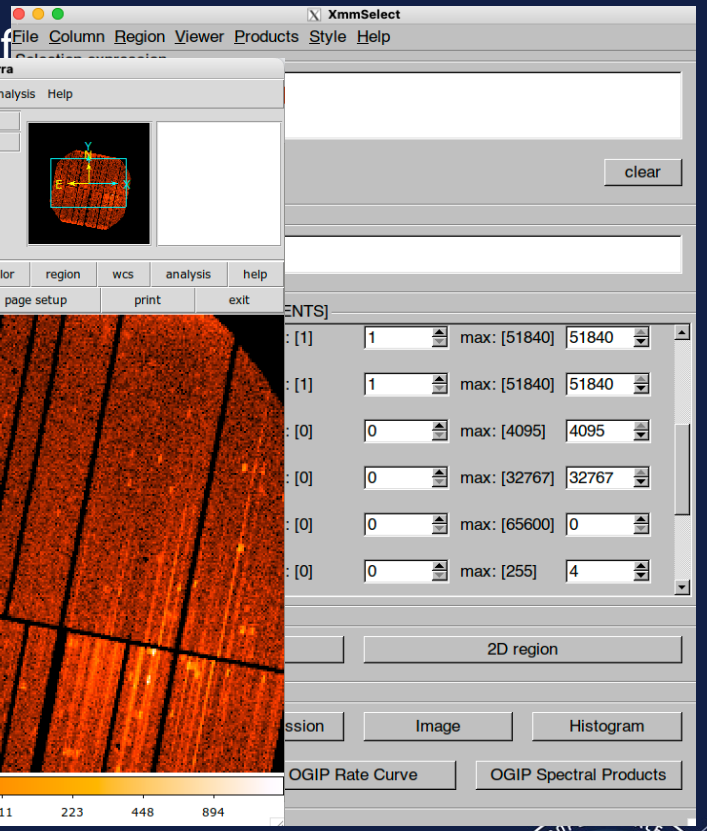
Once we have the event files…. It is time to "xmmselect" to f
produce images, spectra or light curves….

#> xmmselect table=<PN or MOS_event_file>

Follow S
- How to
- Extracti





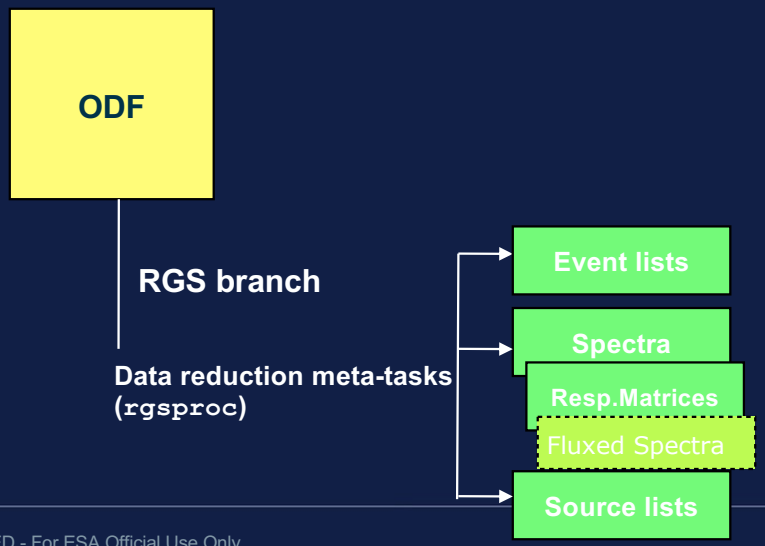


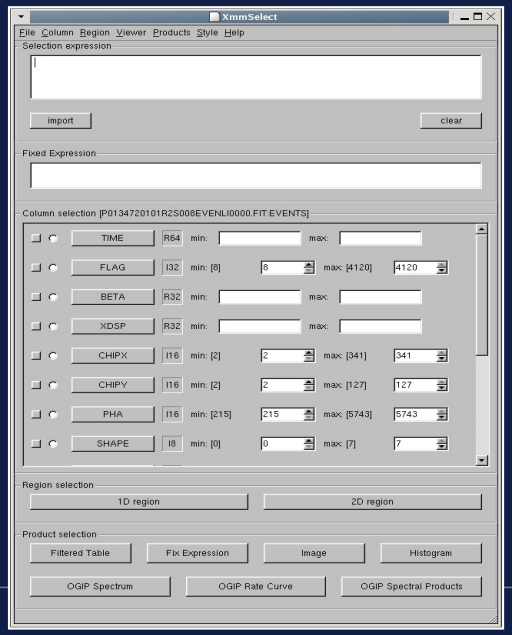
#> eregionanalyse → source centroid and optimum extraction radius

rgsproc

- meta-task: interface to 17 SAS tasks (that can also be run separately)
- controlled by ≈ 80 parameter switches
- five entry and final points (“processing stages”) >>
- produces filtered event lists, spectra and matrices
- the quality of the results depends critically on the source coordinates

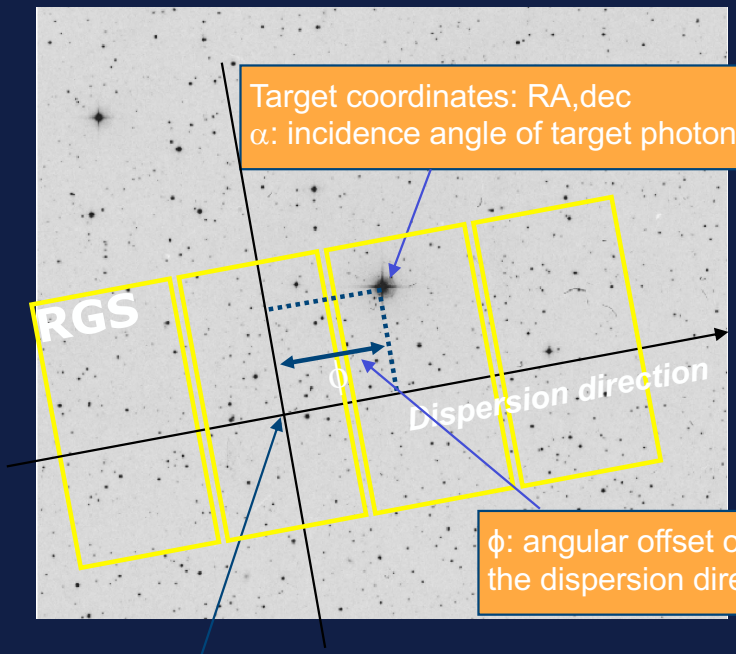
events
angles
filter
spectra
fluxing





Simplified scheme of the RGS FOV

why the coordinates are so important!:



Target coordinates: RA,dec
α: incidence angle of target photons

ϕ: angular offset of the source along the dispersion direction

S/C pointing: RA_o, dec_o
α_o: incidence angle at centre of FOV

According to the grating equation

$$\lambda = (\cos \beta - \cos \alpha)\, d / m$$

being $\alpha = \alpha_o + \phi\, F/L$

$$\phi = f(RA - RA_o, dec - dec_o, P.A.)$$

⇩

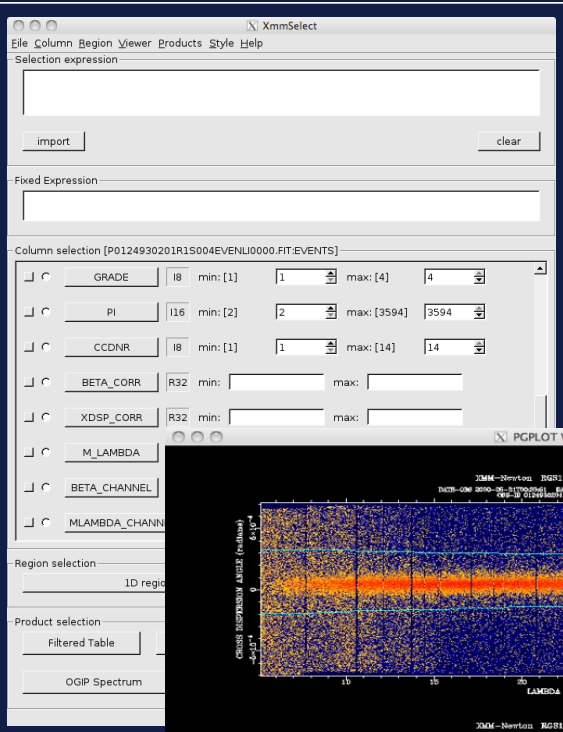
the wavelength scale and the effective area depend on the position of the source in the FOV

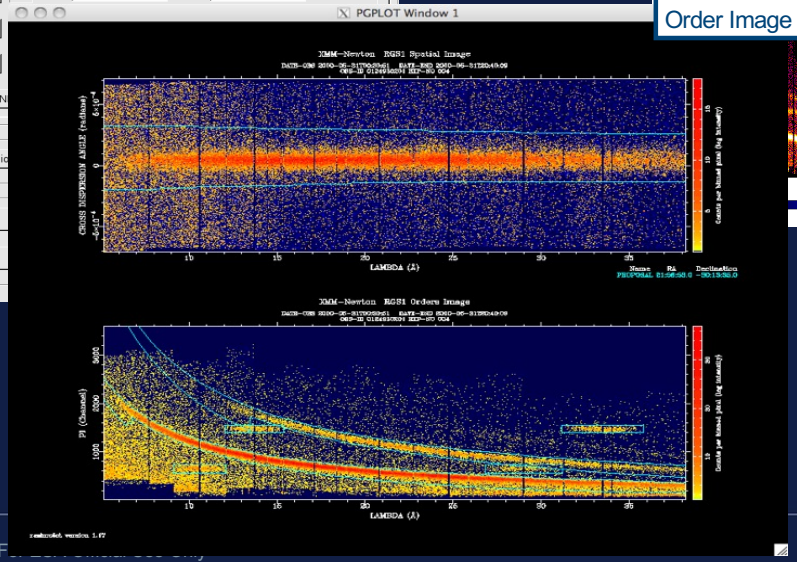
1 arcsec ≈ 2.3 mÅ (45 km/s at 15 Å)

rgsproc: what does it do?

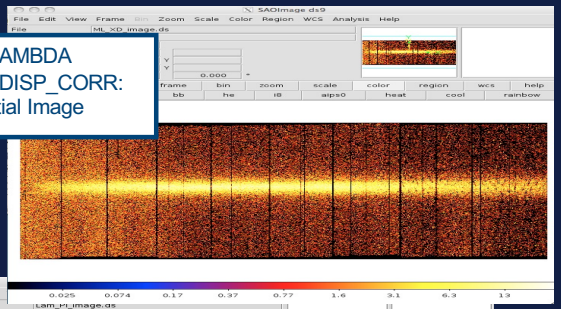
Stage	Task	Purpose	Output	
Events	atthkgen	generates attitude file	Source list + intermediate combined event list	Source independent
	attfilter	filters the attitude file		
	hkgtigen	generates housekeeping GTIs		
	rgsoffsetcalc	uses the diagnostic mode data for offset calculation		
	rgssources	creates the list of sources to processed		
	rgsframes	flags bad frames, convert RAW[XY] to readout node reference system ([XY]CORR), creates GTI for telemetry drops, calculates dead time		
	rgsenergy	performs energy calibrations, i.e. creates the PI column		
	rgsbadpix	flags bad pixels (CCF known + own analysis)		
	rgsevents	event reconstruction: total energy (ENERGY), "pattern" (GRADE/SHAPE), coordinates (CHIP[XY],BETA,XDSP)		
	evlistcomb	event list concatenation		
Angles	rgsangles	aspect correction (M_LAMBDA, XDSP_CORR)	Aspect correction	Source dependent
Filter	rgsfilter	creates filtered event list, removing unwanted frames and events and adding exposure maps	Final event list	
Spectra	rgsregions	computes background and source extraction regions for each source	Source and background spectra	
	rgsspectrum	extracts source and background spectra		
	rgsbkgmodel	generates model background (optional)		
Fluxing	rgsrmfgen	creates a response matrix	Response matrices and combined spectrum in physical units	
	rgsfluxer	combines a collection of RGS spectra into one "fluxed" spectrum		

Rgsproc does a lot

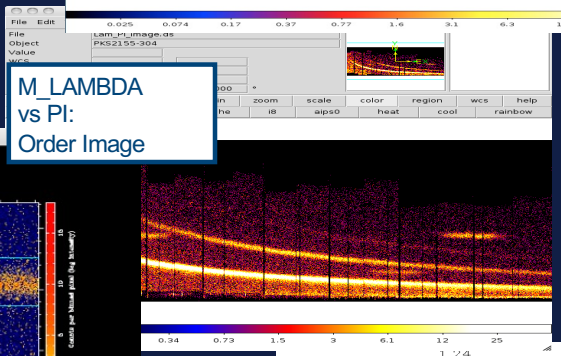


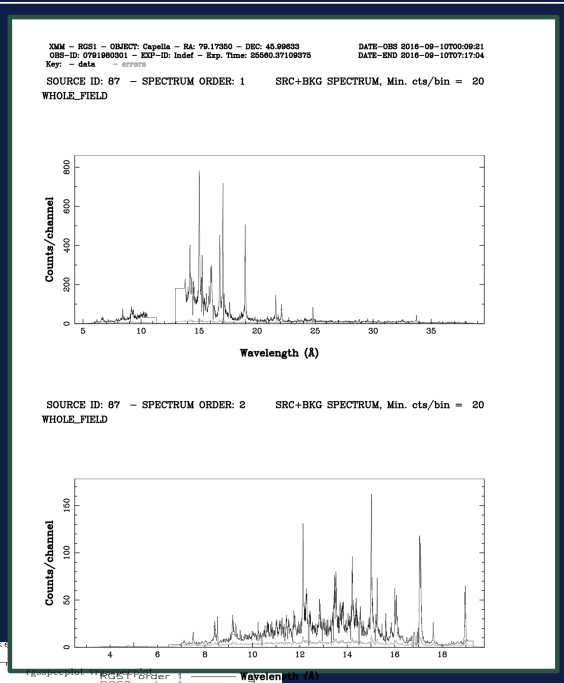


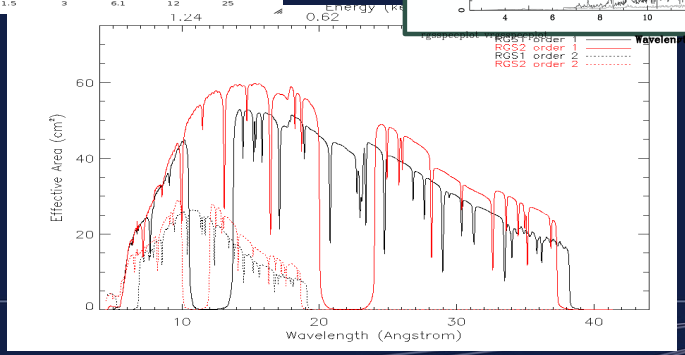
M_LAMBDA vs XDISP_CORR: Spatial Image



M_LAMBDA vs PI: Order Image

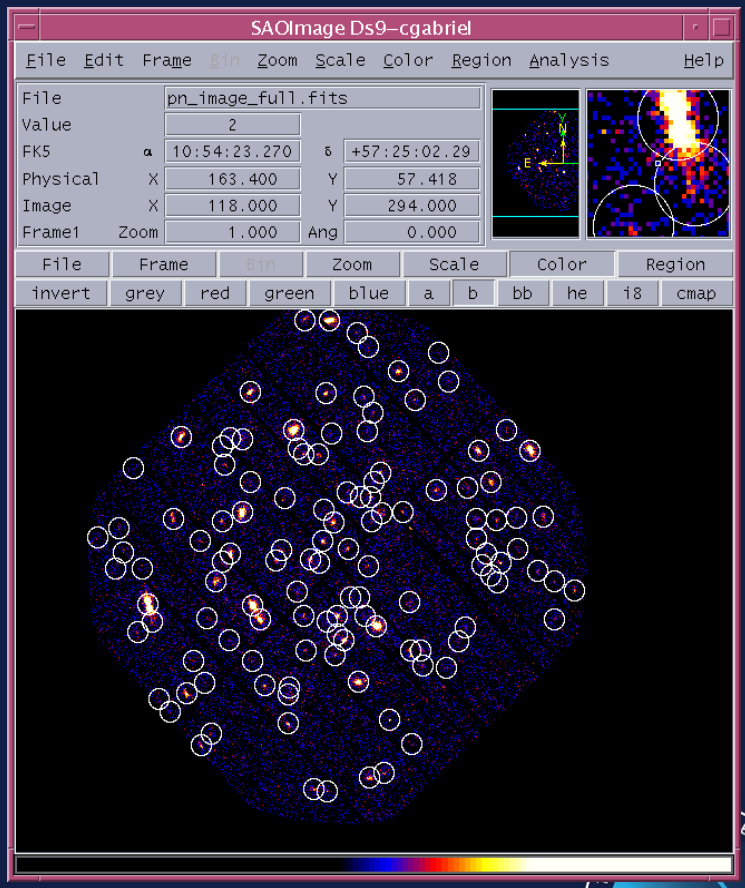






Source Searching

>> source searching means basically looking for:
* **significant fluctuations**, which are
* **compatible with sky sources**,
* lying on top of *more* or *less* **smooth distributions**,
* avoiding to get *fooled* by **detection defects**



>>goal:

to **maximize** source detection sensitivity
minimizing number of fake detections

SAS source searching: preparatory steps

Looking for (small) fluctuations on top of distributions

>> maximization of S/N ratio for sources to be found

... cleaning calibrated event lists against flaring periods

>> produce high energy background lightcurves

+ define a threshold

+ produce GTIs

... taking into account the different source spectral characteristics

>> apply band-passes for deriving corresponding images

SAS source detection tasks

Two methods of performing source detection on EPIC datasets:

1) edetect_chain

perl script running all these tasks consecutively

Task	Purpose	input data sets	output data sets
eexpmap	creation of exposure maps	images, attitude files	exposure maps
emask	creation of detection masks	exposure map	detection mask
eboxdetect (local mode)	sliding box detection	images, exposure maps, detection mask	local box list
esplinemap	creation of background maps	images, exposure maps, detection mask, local box list	background map
eboxdetect (map mode)	box detection using bkg map	images, exposure maps, detection mask, background maps	map detect source
emldetect	maximum likelihood fitting	images, exposure maps, detection mask, background maps	final source list
esensmap	creation of sensitivity maps	exposure map, detection mask, background map	sensitivity map

2) ewavelet mexican hat wavelet algorithm for detecting both point and **extended** sources.
Easy to use and efficient, but less reliable source parameters than those from edetect_chain

SAS Installation and set-up

- Binary distributions of SAS are available **only** for 64bit:
  - Ubuntu 22.04, Ubuntu 42.04 and RHEL8.10
  - Mac OS X (**x86_64**) - 64bit
    + 13.6 – *Ventura*
    + 14.6 - *Sonoma*
  - *Docker version since SAS v21.0*

- Objectives:
  - make it easy to install: untar and go
  - provide all libraries required, also external ones (like cfitsio) - (however, need to be installed: ds9 / FTOOLS / GRACE / Perl)

Mac Note:

- Apple M***X*** is based on ARM architecture
- Intel binaries can be executed thanks to Rosetta emulator
- Currently working on a SAS native Silicon processor build

SAS in could environments

SAS Installation and set-up

SAS Installation and set-up II

All SAS installations are binary (no support for building from source code)

#> tar zxf sas_22.1.0-[git_commit]-[date]-[OS]-[compiler]-[architecture].tgz

sas_22.1.0-a8f2c2afa-20250304-ubuntu22.04-gcc13.3.0-x86_64.tgz

./install.sh (will check everything is in place)

recommendation: download and install Miniconda (python)

>> `xmmsas_22.1.0-a8f2c2afa-20250304` directory with all contents

Every time you want to run SAS:
1.- Set HEASOFT
2.- Define SAS_DIR enviroment variable
#> export SAS_DIR=<top-dir>/xmmsas_22.1.0-a8f2c2afa-20250304
Or
#> source SAS_DIR <top-dir>/xmmsas_22.1.0-a8f2c2afa-20250304
3.- Execute **setsas** script
#> . $SAS_DIR/setsas.sh (bash)
#> source ${SAS_DIR}/setsas.csh (csh)

Getting started with SAS II: the CCF

XMM-Newton calibration data is contained in Current Calibration File (CCF)

CCF = the collection of all the XMM-Newton calibration files ever made public (= hundreds!)
Note: the calibration files are updated continuously >>> NO CCF version number

Calibration Index File (CIF) necessary for data analysis, pointing to the relevant files, according to:

- observation date
- analysis date

cifbuild operates on the calibration directory $SAS_CCFPATH
#> export SAS_CCFPATH=<ccf_dir>
#> setenv SAS_CCFPATH <ccf_dir>

Command: #> **cifbuild**
produces a FITS file ccf.cif with extension CALINDEX >

WHERE is the data???
Set **SAS_ODF** env variable

After the Calibration Index file has been produced:
#> export **SAS_CCF**=*absolute_path/*ccf.cif
#> setenv **SAS_CCF** *absolute_path*/ccf.cif

NOTICE: any file with extension CALINDEX is valid as SAS_CCF

File Edit Tools

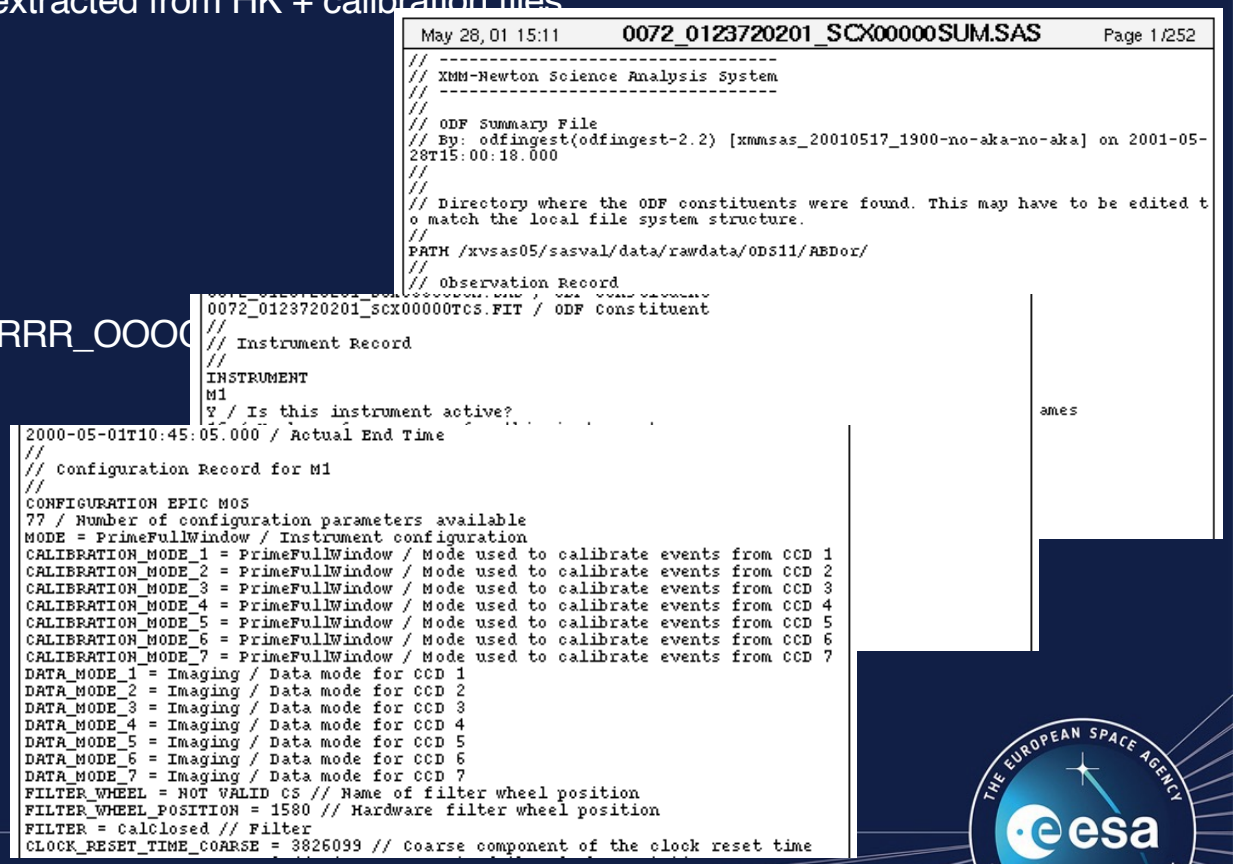
	TELESCOP 4A	SCOPE 6A	TYPEID 32A	ISSUE I	VALDATE 19A yyyy:dd:mmZhh:mm:ss
40	XMM	EPN	LINCOORD	9	1998-01-01T00:00:00
41	XMM	EPN	MODEPARAM	3	1999-01-01T00:00:00
42	XMM	EPN	PATTERNLIB	1	1998-01-01T00:00:00
43	XMM	EPN	QUANTUMEF	8	2000-01-01T00:00:00
44	XMM	EPN	REDIST	5	1998-01-01T00:00:00
45	XMM	EPN	TIMECORR	4	1998-01-01T00:00:00
46	XMM	OM	ASTROMET	8	1998-01-01T00:00:00
47	XMM	OM	BADPIX	2	1998-01-01T00:00:00
48	XMM	OM	COLORTRANS	5	1998-01-01T00:00:00
49	XMM	OM	DARKFRAME	3	1998-01-01T00:00:00
50	XMM	OM	DIFFUSEGALA	1	1998-01-01T00:00:00
51	XMM	OM	HKPARMINT	3	1999-01-01T00:00:00
52	XMM	OM	LARGESCALESENS	2	1998-01-01T00:00:00
53	XMM	OM	LINCOORD	1	1998-01-01T00:00:00
54	XMM	OM	PHOTTONAT	3	1998-01-01T00:00:00
55	XMM	OM	PIXTOPIXSENS	3	1998-01-01T00:00:00
56	XMM	OM	PSF1DRB	4	1998-01-01T00:00:00
57	XMM	OM	QUICKMAG	2	1998-01-01T00:00:00
58	XMM	OM	ZODIACAL	1	1998-01-01T00:00:00
59	XMM	RGS1	ADUCONV	5	2000-02-06T16:49:60
60	XMM	RGS1	BACKGROUND	1	1998-01-01T00:00:00
61	XMM	RGS1	BADPIX	5	2000-02-06T16:49:60
62	XMM	RGS1	CALSOURCEDATA	1	1998-01-01T00:00:00
63	XMM	RGS1	CLOCKPATTERNS	1	1998-01-01T00:00:00
64	XMM	RGS1	CROSSPSF	2	2000-01-01T00:00:00
65	XMM	RGS1	CTI	2	2000-02-06T16:49:60
66	XMM	RGS1	DARKFRAME	4	1998-01-01T00:00:00
67	XMM	RGS1	HKPARMINT	6	1999-01-01T00:00:00
68	XMM	RGS1	LINCOORD	7	1998-01-01T00:00:00
69	XMM	RGS1	LINESPREADFUNC	3	1999-01-01T00:00:00

Getting started with SAS I: the ODF

SAS needs for processing the ODF an Advanced Summary File (SUM.SAS), produced by odfingest, extending an original summary file with data extracted from HK + calibration files

odfingest operates on the ODF directory $SAS_ODF
#> setenv **SAS_ODF** <odf_dir>
#> export **SAS_ODF** <odf_dir>

Command: *#> odfdir=$SAS_ODF*
will produce a Summary **odfingest** file <SUM> = RRRR_OOOO
in the working directory



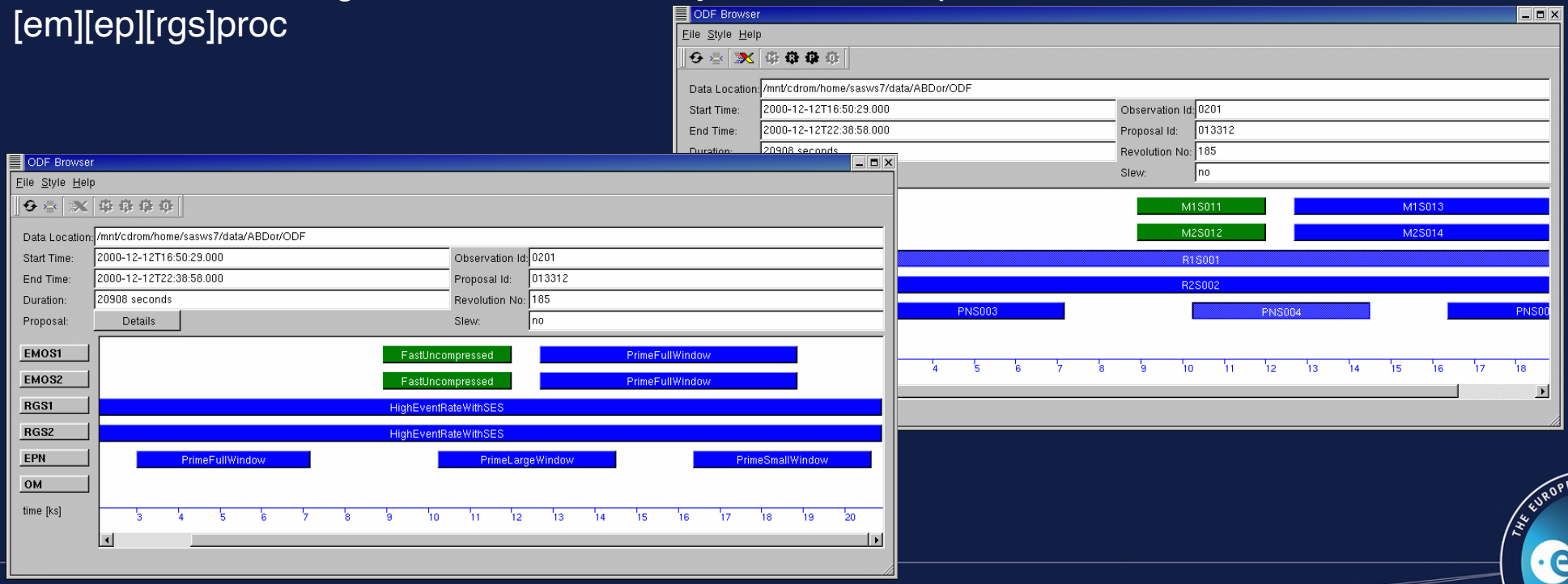
After the Summary file has been produced:

#> setenv **SAS_ODF** <SUM>
#> export **SAS_ODF**=<SUM>

odfbrowser

A task to view the contents of the ODF and more …

An ODF is constituted in the rule by hundreds of files. odfbrowser displays graphical summaries of an observation, allowing the user to select any number of exposures and launch the metatasks [em][ep][rgs]proc



cifbuild

cifbuild uses single CCF keywords:

- VALDATE as start of calibration validity period
- EVALDATE as end of validity period
- DATE as analysis validity period

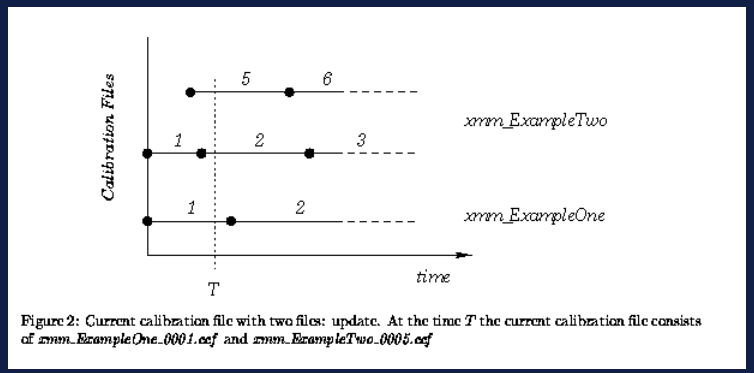


Figure 2: Current calibration file with two files: update. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0005.ccf*

Rule: out of all the CCF calibration files take the highest issue with VALDATE lower AND EVALDATE higher than observation date AND DATE lower than analysis date.

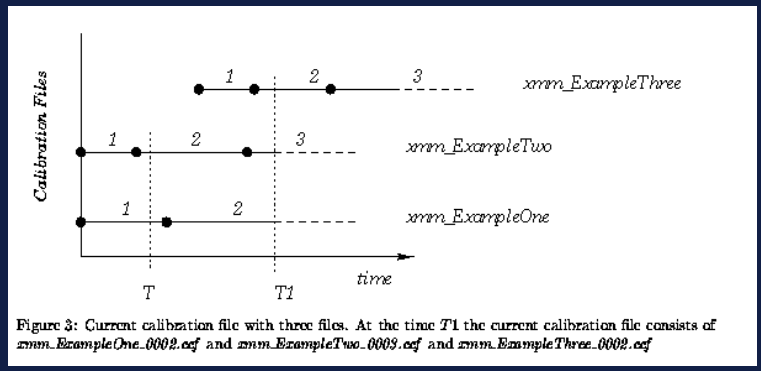


Figure 3: Current calibration file with three files. At the time $T1$ the current calibration file consists of *xmm_ExampleOne_0002.ccf* and *xmm_ExampleTwo_0003.ccf* and *xmm_ExampleThree_0002.ccf*

Getting started with SAS III: the basic steps

Summary of basic steps to start using SAS:
(csh / tcsh version)

#> source <SAS-DIR>**/setsas.csh**

#> setenv SAS_ODF <ODF-DIR>
#> setenv SAS_CCFPATH <Calibration Files-DIR>
#> cifbuild
#> setenv SAS_CCF ccf.cif

#> odfingest
#> set sumfile=`ls -1 *SUM.SAS`
#> setenv SAS_ODF $sumfile

>> ready to start working on the ODF data
located in <ODF-DIR>

```
xmml110:~% source $SAS_DIR/setsas.csh

sasversion:- Executing (routine): sasversion  -w 1 -V 4
sasversion:- sasversion (sasversion-1.3)  [22.1.0-a8f2c2afa-20250304] started:  2025-10-08T11:58:14.000
sasversion:- XMM-Newton SAS release and build information:

SAS release: 22.1.0-a8f2c2afa-20250304
Compiled on: Tue Mar  4 07:29:35 UTC 2025
Compiled by: sasbuild@8b74f8fb7fa2
Platform   : Ubuntu22.04

SAS-related environment variables that are set:

SAS_DIR = /sas/Linux/Ubuntu22.04/64/xmmsas_22.1.0-a8f2c2afa-20250304
SAS_PATH = /sas/Linux/Ubuntu22.04/64/xmmsas_22.1.0-a8f2c2afa-20250304
SAS_CCFPATH = /ccf/valid

sasversion:- sasversion (sasversion-1.3)  [22.1.0-a8f2c2afa-20250304] ended:    2025-10-08T11:58:14.000

Do not forget to define SAS_CCFPATH, SAS_CCF and SAS_ODF
```

Getting started with SAS III: the basic steps

OR, if you use the bash / sh / ksh:

#> . <SAS-DIR>/setsas.sh

#> export SAS_ODF =<ODF-DIR>
#> export SAS_CCFPATH =<Calibration Files-DIR>
#> cifbuild
#> export SAS_CCF=ccf.cif

#> odfingest
#> export sumfile=`ls -1 *SUM.SAS`
#> export SAS_ODF=$sumfile

>> ready to start working on the ODF data located in <ODF-DIR>

```
(pysas) aibarra@xmml110:~$ . $SAS_DIR/setsas.sh

sasversion:- Executing (routine): sasversion  -w 1 -V 4
sasversion:- sasversion (sasversion-1.3)  [22.1.0-a8f2c2afa-20250304] started:  2025-10-08T12:48:46.000
sasversion:- XMM-Newton SAS release and build information:

SAS release: 22.1.0-a8f2c2afa-20250304
Compiled on: Tue Mar  4 07:29:35 UTC 2025
Compiled by: sasbuild@8b74f8fb7fa2
Platform   : Ubuntu22.04

SAS-related environment variables that are set:

SAS_DIR = /xdata/sas/Linux/Ubuntu22.04/64/xmmsas_22.1.0-a8f2c2afa-20250304
SAS_PATH = /xdata/sas/Linux/Ubuntu22.04/64/xmmsas_22.1.0-a8f2c2afa-20250304
SAS_CCFPATH = /ccf/valid

sasversion:- sasversion (sasversion-1.3)  [22.1.0-a8f2c2afa-20250304] ended:    2025-10-08T12:48:46.000

Do not forget to define SAS_CCFPATH, SAS_CCF and SAS_ODF
```

Getting started with SAS IV: all the information

SAS public web page: https://www.cosmos.esa.int/web/xmm-newton/sas (download, installation, docs, etc)

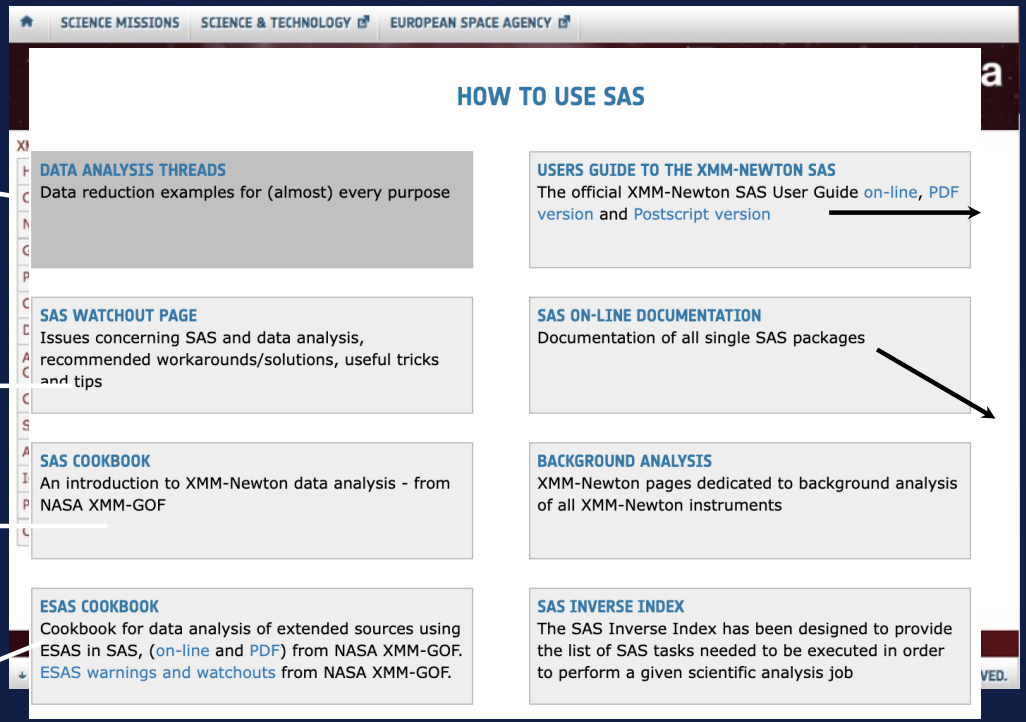
...How to use SAS

Analysis recipes

Problems??

ABC guide

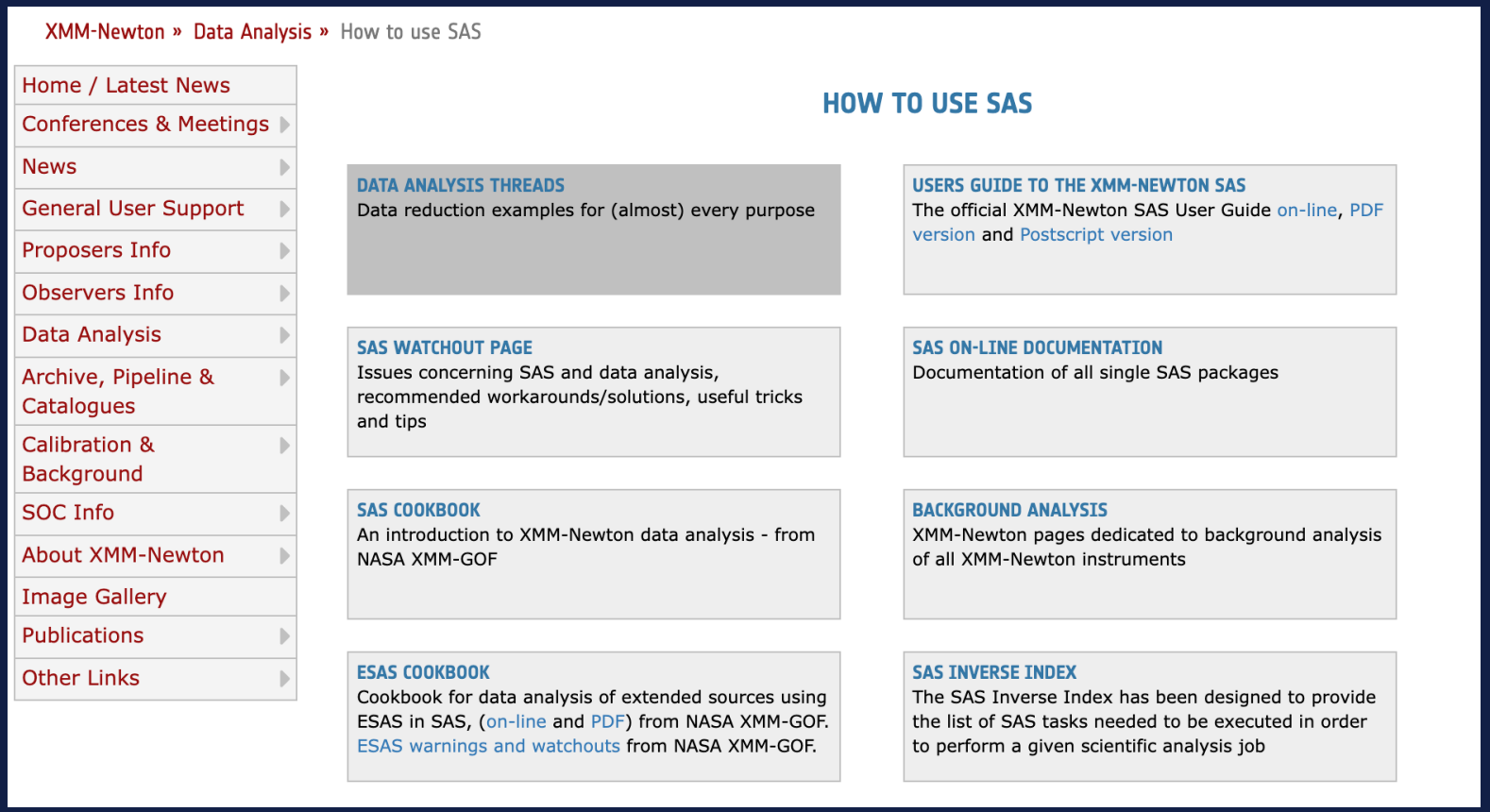
Extended sources



General guide

Task by task info

SAS Documentation



The SAS threads

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
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	
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		
- EPIC source finding thread in one go: `edetect_chain`	command line	
- EPIC source finding thread: step-by-step	command line	
- EPIC source finding in overlapping exposures	command line	
Slew data processing		
- How to process EPIC slew data	command line	

SAS THREADS

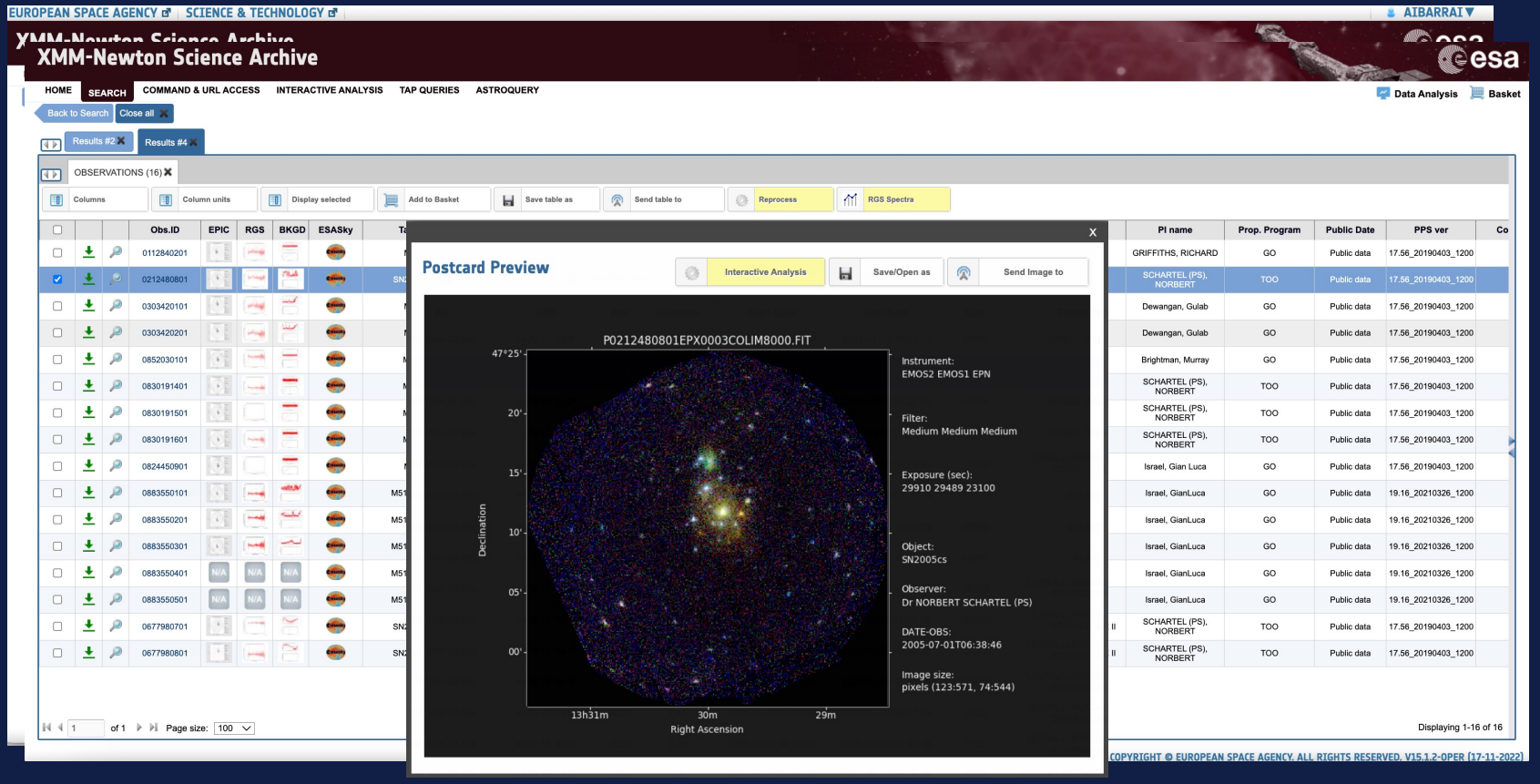
RGS RELATED THREADS

All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
- How to reduce RGS data and extract spectra of point-like sources	command line	
- `rgsproc`, coordinates and masks	command line	
Light curve generation:		
- Extraction of a light curve for a point-like source (EPIC and RGS)	command line	GUI version
More complex analysis for the very bright sources		
- Pile-up in the RGS: how to prevent it, evaluate its existence and make corrections	command line	

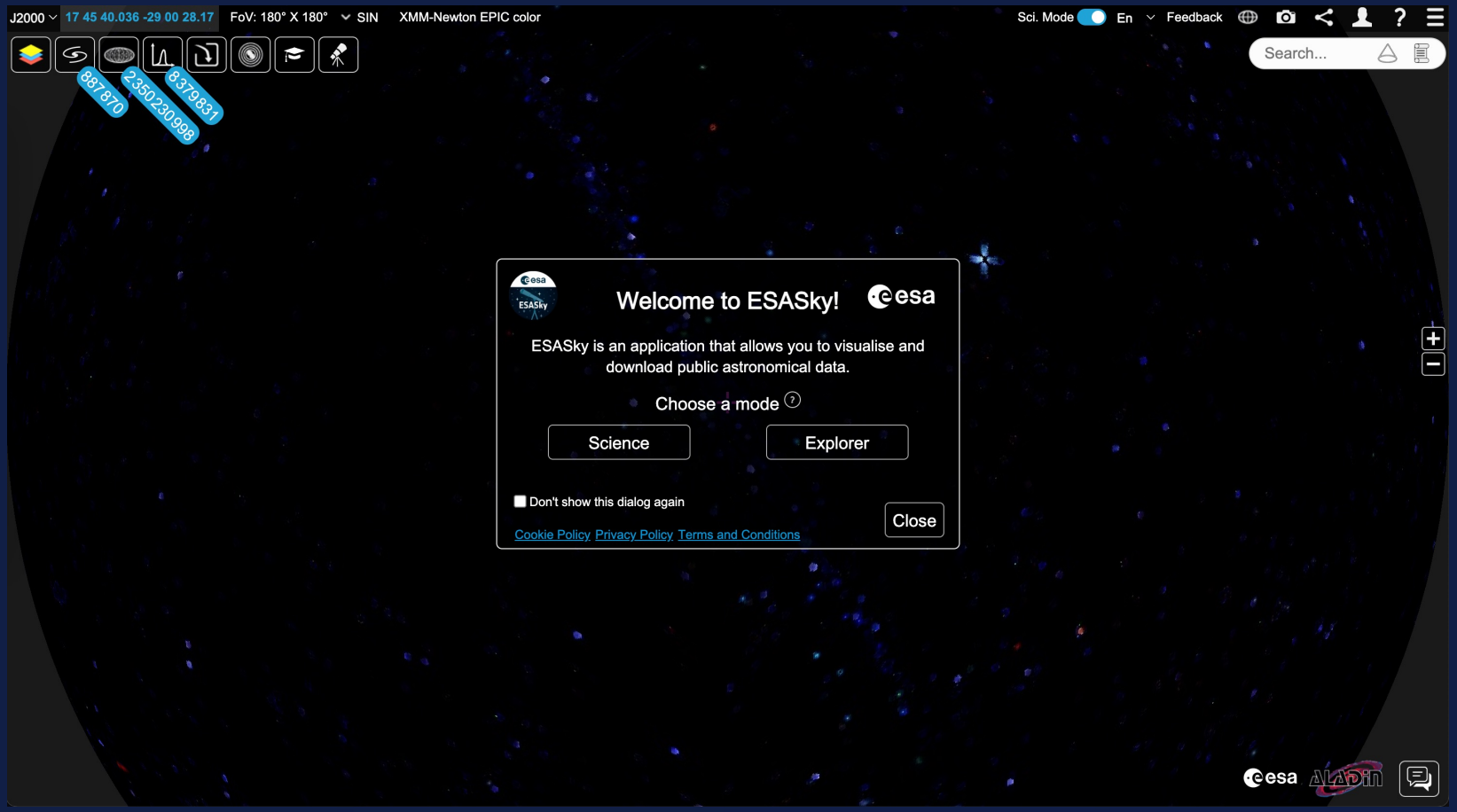
OM RELATED THREADS

All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
- OM image mode data processing chain	processing chain	command line
- OM fast mode data processing chain	processing chain	command line
- OM Grism processing chain	processing chain	command line
- Interactive OM photometry	command line	
- Converting OM data to OGIP II format (use in `xspec`)	command line	

How to get XMM-Newton data? The XSA



http://sky.esa.int



What I need to do ...

… before starting to analyse data of an XMM-Newton observation:

1. Verify the quality of the pre-processed scientific products (PPS), produced by the automatic Pipeline processing

2. Check the expected accuracy of the XMM-Newton calibrations, through:
    1. Instrument calibration status reports
    2. SAS Science Validation Reports
    3. Current Calibration File (CCF) Release Notes

3. Compare your own set of calibration files with the latest available
    1. Reduce the data again if a calibration file has changed, which may affect your scientific conclusions. **Always stay on the safe side!**

4. Once you have installed SAS, your job is not finished ...
    1. Check the SAS “watchout and evergreen” SAS pages, which contain known caveats or bugs
    2. Install an automatic mirror of the calibration files
    3. Make use of the threads, would you like to learn something new
    4. Contact the HelpDesk, if everything else fails: `xmmhelp@cosmos.esa.int`

THANK YOU FOR THE ATTENTION!

Any questions ?


Produced by
Aitor Ibarra on behalf of XMM-Newton SOC
THE EUROPEAN SPACE AGENCY
esa
1975
2025
50