Introduction to the XMM-Newton Science Analysis System

Aitor IBARRA & Carlos GABRIEL (+ almost the whole... at least a lot of the)

XMM-Newton Science Operations Center - ESA

Outline

- What is the SAS?
- SAS installation and setup
- SAS data reduction scheme
- Getting started I: the Observation Data File (ODF) + odfbrowser
- Getting started II: the Calibration Current File (CCF) and the Calibration Index File (CIF)
- Getting started III: the common first steps
- Running SAS: GUI or command line
- PPS or “SAS has already reduced these data”
- Retrieving XMM-Newton data
- SAS as web services —> "RISA" —> XSA to reprocess data
- SAS & Jupyter notebooks & DataLabs Initiative: Shaping the future

What is the SAS?

What is the 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)

What is the 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

Python is coming….

- 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)

What is the 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

Python is coming….

- 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 v20.1 – released on November 11, 2021

What is the 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

Python is coming….

- 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 v20.1 – released on November 11, 2021

New SAS v21.0 to be released on April 2023

Installation and setup

- Binary distributions of SAS are available **only** for 64bit:
  - Linux Centos 7.3, Ubuntu 20.04 and RHEL8.6
  - Mac OS X - 64bit
    + 11.07 - *Big Sur* and 12.07 - *Monterrey*
  - *Virtual Machine running Ubuntu 20.04 64bit (Windows, Linux and MacOSX).*
  - *Docker version since SAS v20.0*

  >> "official" supported platforms which can be used by other OSs

- 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)

Installation and setup

- Binary distributions of SAS are available **only** for 64bit:
  - Linux Centos 7.3, Ubuntu 20.04 and RHEL8.6
  - Mac OS X - 64bit
    + 11.07 - *Big Sur* and 12.07 - *Monterrey*
  - *Virtual Machine running Ubuntu 20.04 64bit (Windows, Linux and MacOSX).*
  - *Docker version since SAS v20.0*

Mac Note:

- Apple M1 is based on ARM architecture
- Intel binaries can be executed thanks to Rosetta emulator
- Currently working on a SAS native M1 build

>> "official" supported platforms which can be used by other OSs

- 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)

Installation and setup II

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

tar zxf sas_20.0.0-[OS].tgz

./install.sh (will check everything is in place, download and install a Miniconda if necessary and finally install SAS)

>> xmmsas_20211130_0941
directory with all contents

Installation and setup II

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

GOAL TO RELEASE SAS SOURCE CODE IN 2023

tar zxf sas_20.0.0-[OS].tgz

./install.sh (will check everything is in place, download and install a Miniconda if necessary and finally install SAS)

>> xmmsas_20211130_0941
directory with all contents

Installation and setup II

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

GOAL TO RELEASE SAS SOURCE CODE IN 2023

tar zxf sas_20.0.0-[OS].tgz

./install.sh (will check everything is in place, download and install a Miniconda if necessary and finally install SAS)

>> xmmsas_20211130_0941
directory with all contents

Everytime you want to run SAS:

>. ./setsas.sh (bash) or > source ./setsas.csh (csh) in that directory

or

> . <top-dir>/xmmsas_20211130_0941/setsas.sh
> source <top-dir>/xmmsas_20211130_0941/setsas.csh

Basics 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:
  - **position (X-Y) on the detector**
  - **"pulse height", which is related to the X-ray energy (E) of the incoming photon in a complex and generally non-linear way**
  - **arrival time (t) at the spacecraft**
  - event "shape" (used to separate X-ray events from particles' signatures)
  - other secondary attributes (you don't generally have to worry about)

Event lists

When? **Where?** **Who?** **What?**

	TIME D s	X J 0.05 ARCSECONDS	Y J 0.05 ARCSECONDS	PHA I CHAN	PI I CHAN	PATTERN B	CCDNR B
1	9.506202266412E+07	23743	21330	423	1447	2	1
2	9.506202266412E+07	28728	21990	25	98	0	1
3	9.506202527717E+07	28176	31623	25	97	0	1
4	9.506202527717E+07	29829	30841	327	1131	0	1
5	9.506202527717E+07	23686	19319	541	1854	0	1
6	9.506203046611E+07	25510	32711	1810	6171	0	1
7	9.506203566620E+07	29814	28823	102	360	0	1
8	9.506203826626E+07	26635	30601	2062	7028	0	1
9	9.506204346625E+07	26429	20314	443	1519	4	1
10	9.506204606629E+07	20691	28728	1608	5471	3	1
11	9.506204606629E+07	27989	29777	202	700	0	1
12	9.506204606629E+07	21937	25667	117	402	2	1
13	9.506204866632E+07	28132	32491	462	1589	0	1
14	9.506204866632E+07	27204	29741	904	3095	0	1
15	9.506205126638E+07	22124	20257	290	994	0	1
16	9.506205906643E+07	23193	18795	1398	4771	0	1
17	9.506206166646E+07	23224	19326	276	950	0	1
18	9.506206946653E+07	27755	28979	183	637	0	1
19	9.506207206939E+07	22533	29563	33	118	0	1

Science as projection

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 gets an energy distribution function (***spectrum***) in units of ***counts per energy bin***
- By collapsing time and energy, one gets a 2-D ***image*** in units of ***counts per pixel***
- By collapsing space and energy, one gets an intensity ***time series*** in units of ***counts per time bin***

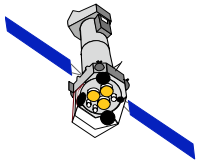
Science as projection

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 gets an energy distribution function (***spectrum***) in units of ***counts per energy bin***
- By collapsing time and energy, one gets a 2-D ***image*** in units of ***counts per pixel***
- By collapsing space and energy, one gets an intensity ***time series*** in units of ***counts per time bin***

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

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

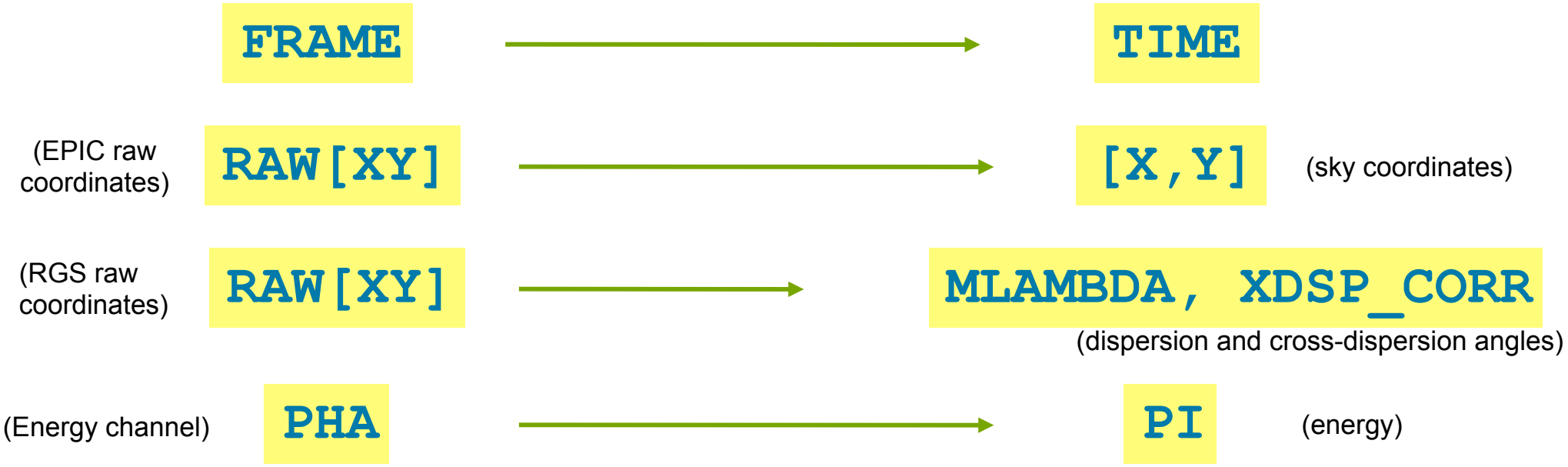
Detector calibration

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

Reduction scope

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

- applies calibrations to raw data

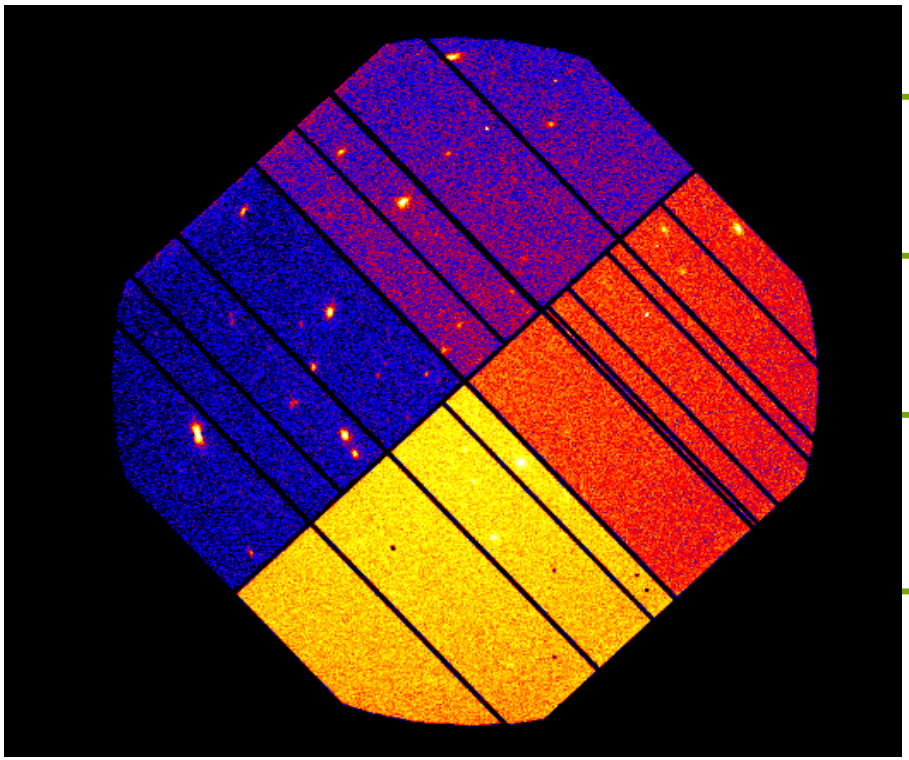


- optimally screen / filter your data

Reduction scope

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

- applies calibrations to raw data



`TIME`

`[X,Y]` (sky coordinates)

`MLAMBDA, XDSP_CORR`
(dispersion and cross-dispersion angles)

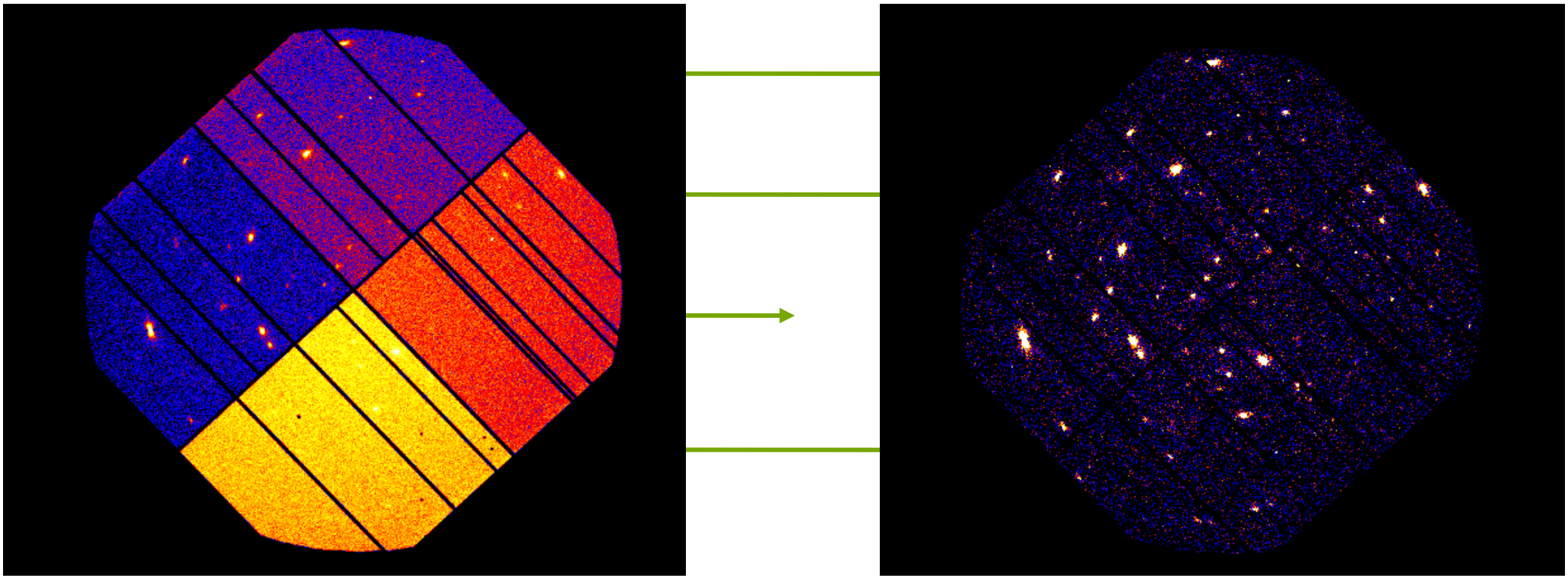
`PI` (energy)

- optimally screen / filter your data

Reduction scope

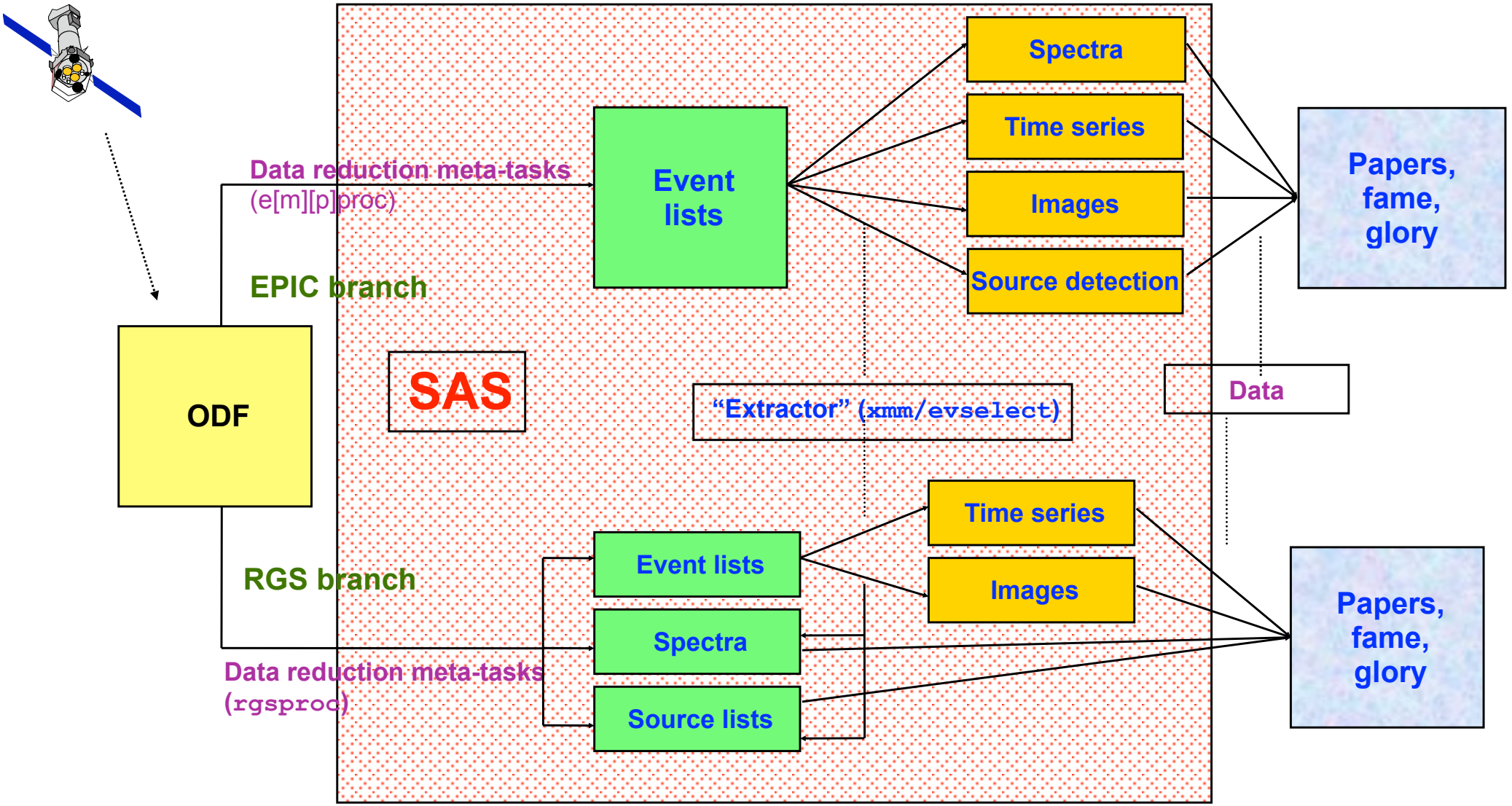
SAS does two things (to XMM data), that no other tool does:

- applies calibrations to raw data

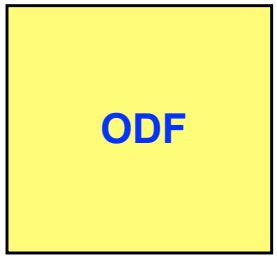


- optimally screen / filter your data

SAS Grand-Scheme



ODF contents



- CCD-based event lists, containing uncalibrated quantities
- Auxiliary and Housekeeping files, pn/RGS diagnostic images
- Spacecraft housekeeping
- Spacecraft attitude showing the satellite star tracker pointing
- Time correlation file (onboard time and frame counter versus UTC)
- ODF summary file

ODF contents

ODF

- CCD-based event lists, containing uncalibrated quantities
- Auxiliary and Housekeeping files, pn/RGS diagnostic images
- Spacecraft housekeeping

FITS

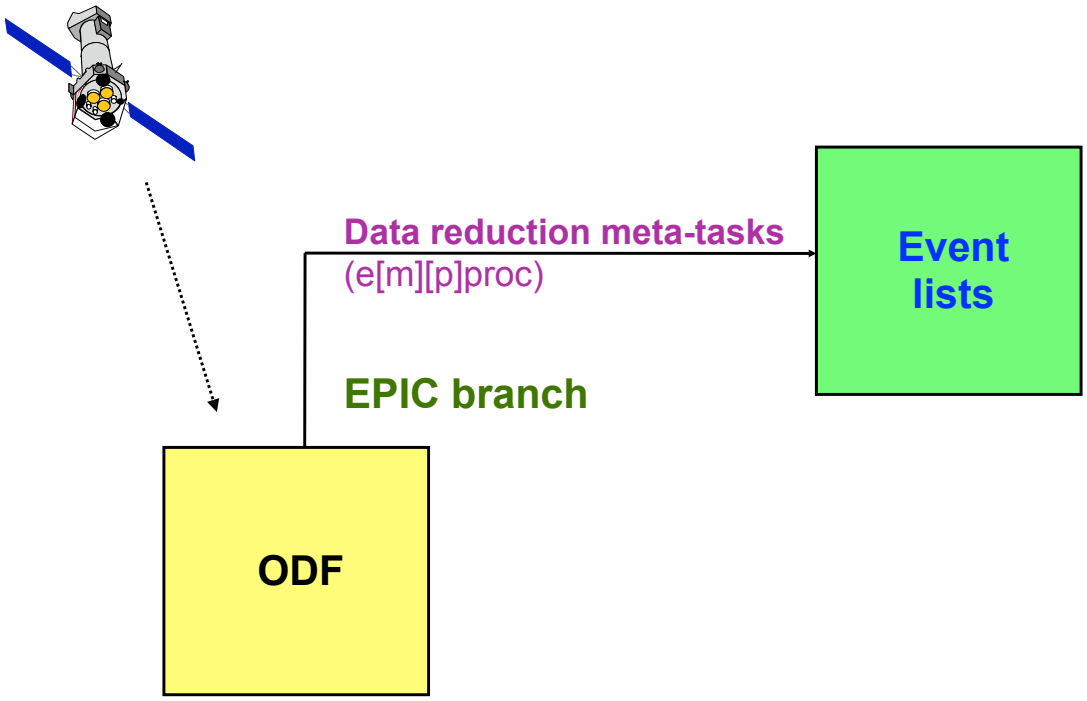
- Spacecraft attitude showing the satellite star tracker pointing
- Time correlation file (onboard time and frame counter versus UTC)
- ODF summary file

ODF contents

ODF

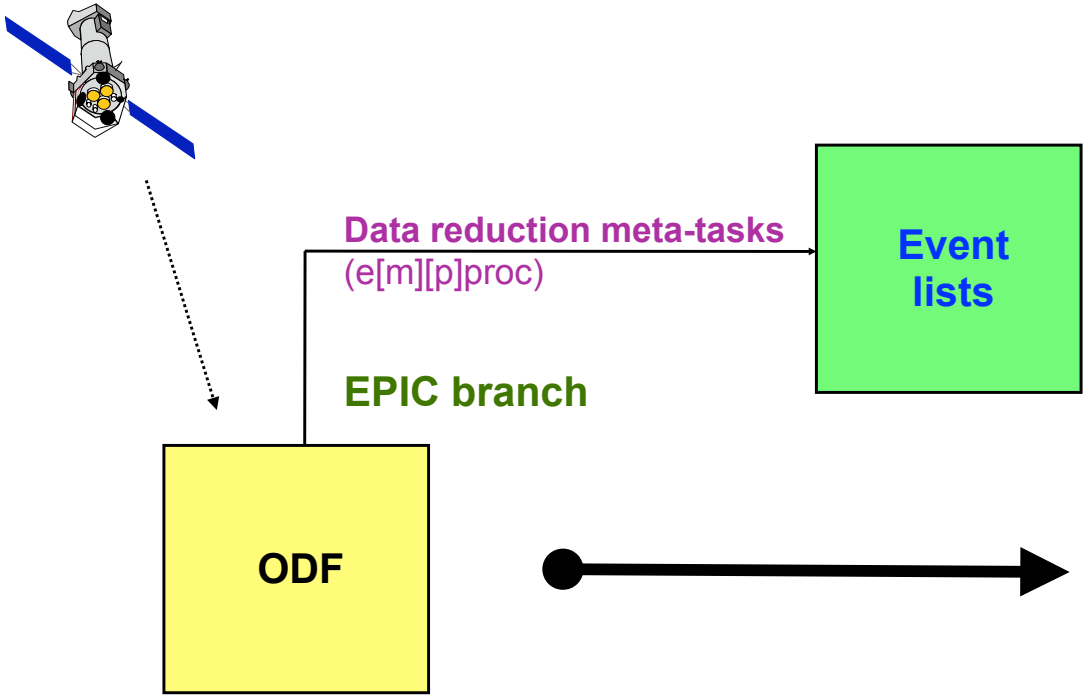
- CCD-based event lists, containing uncalibrated quantities
- Auxiliary and Housekeeping files, pn/RGS diagnostic images
- Spacecraft housekeeping — FITS
- Spacecraft attitude showing the satellite star tracker pointing
- Time correlation file (onboard time and frame counter versus UTC)
- ODF summary file — ASCII

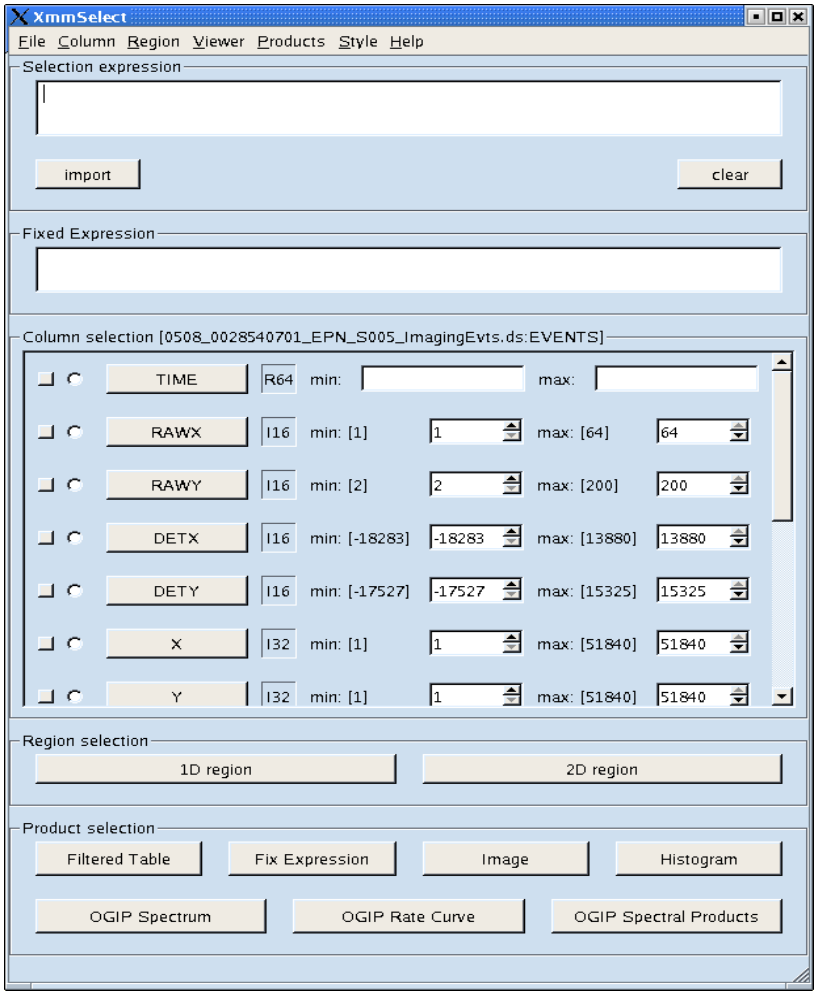
epicproc = e[m][p]proc



- metatasks to process MOS or pn data
- generate calibrated, filtered event lists
- leave user in control of GTI and filter expressions

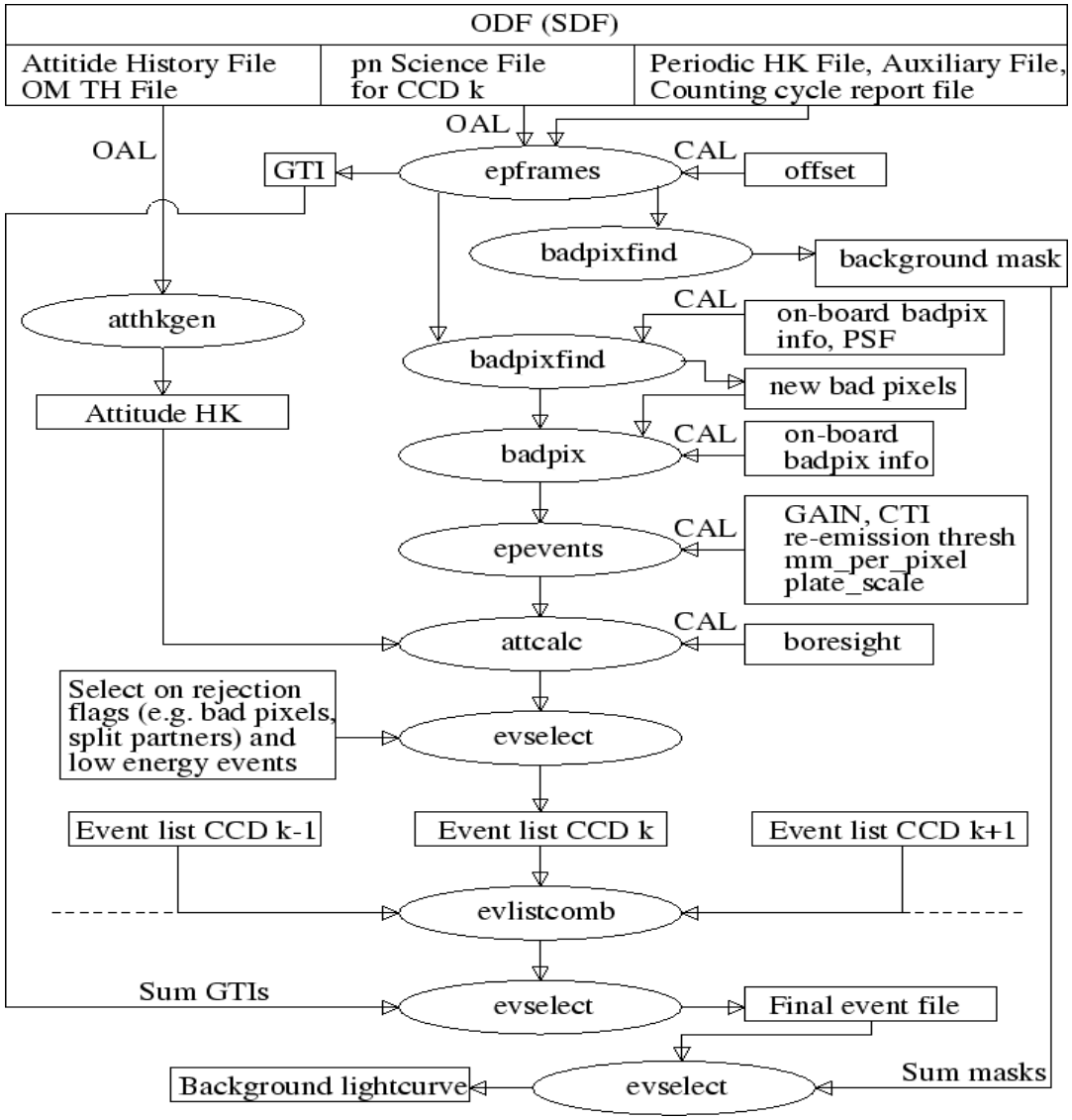
epicproc = e[m][p]proc





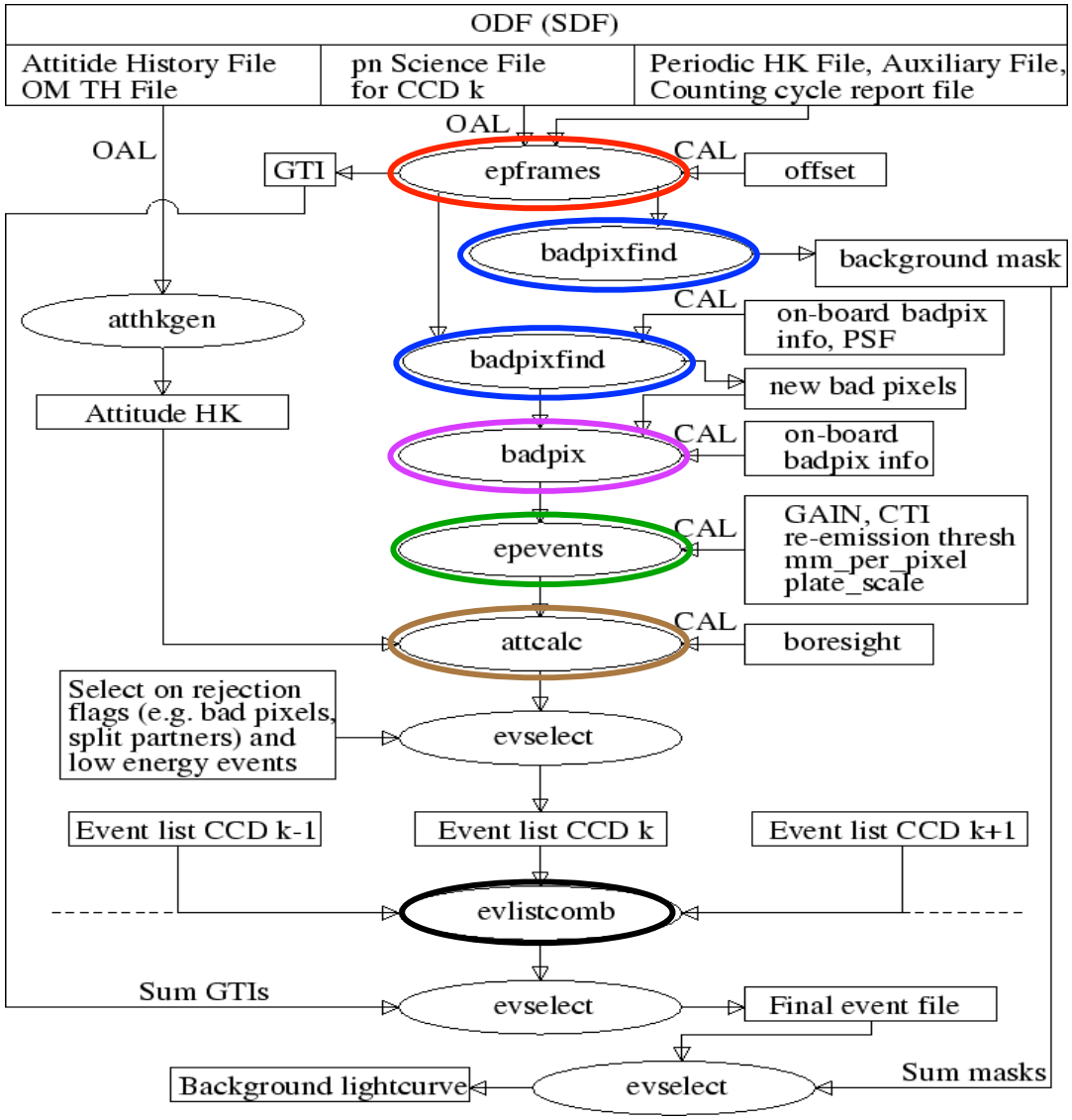
- metatasks to process MOS or pn data
- generate calibrated, filtered event lists
- leave user in control of GTI and filter expressions

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.

epproc reduction scheme

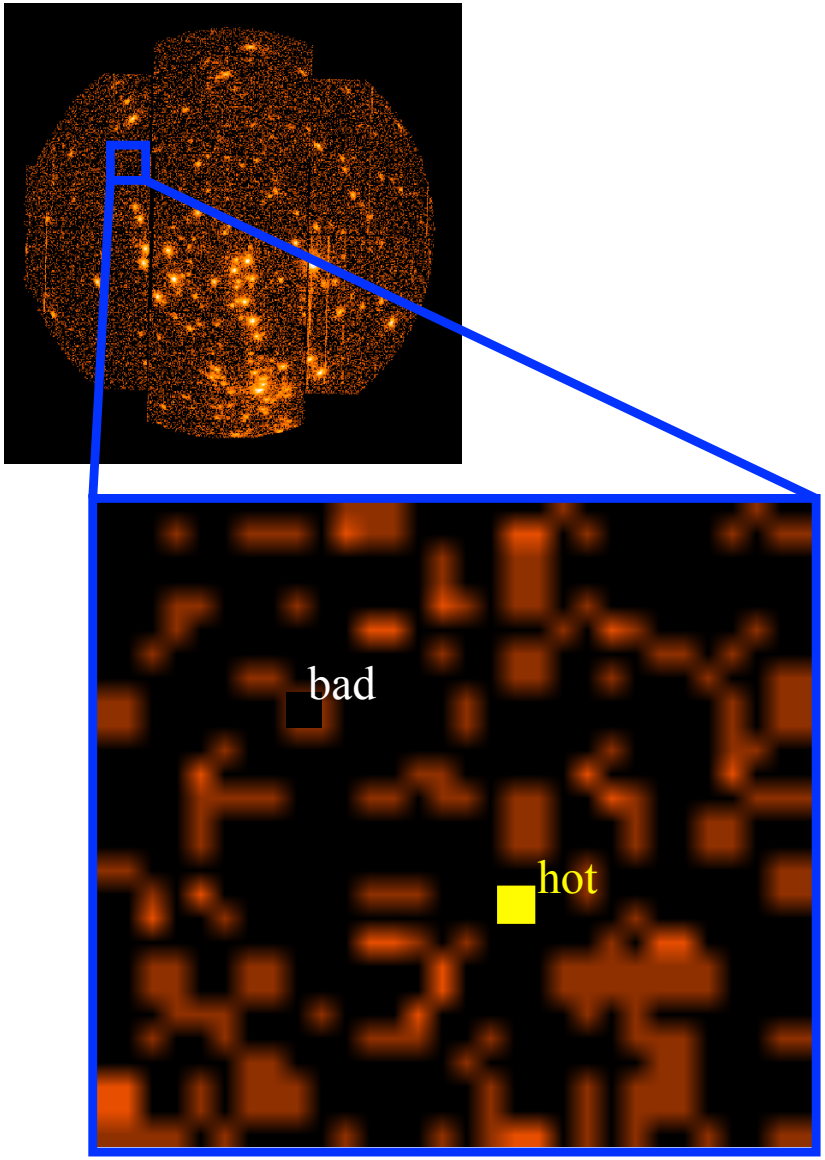


- **epframes** to process a CCD, exposure and datamode specific ODF file, creating the output raw event list and GTI data set
- **badpixfind** to find new bad pixels
- **badpix** to process the raw event list, adding the BADPIX extension
- **epevents** to process the event list file, flagging trailing 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.

Event lists

- In the EPIC pn imaging mode, the EVENTS table of the calibrated event list files contain 14 columns i.e :
  - TIME --> when did my photon arrive
  - RAWX RAWY --> where on the CCD
  - DETX DETY -->where on the detector
  - X Y --> where from the sky
  - PHA PI --> which energy did my photon have
  - FLAG --> did it hit the detector at a critical place
  - PATTERN --> was it a single/double.....
  - CCDNR --> on which CCD did it hit the detector

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
  - 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.

Fundamental final product of epicproc is the event list:

*[Imaging-Timing]Evts.ds (eg. 0193_0112570601_EMOS1_S001_ImagingEvts.ds)

+ BadPixel tables produced by (em)badpixfind

EPIC-PN spectra

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

#> xmmselect table=<PNorMOS_event_file>

EPIC-PN spectra

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

#> xmmselect table=<PNorMOS_event_file>

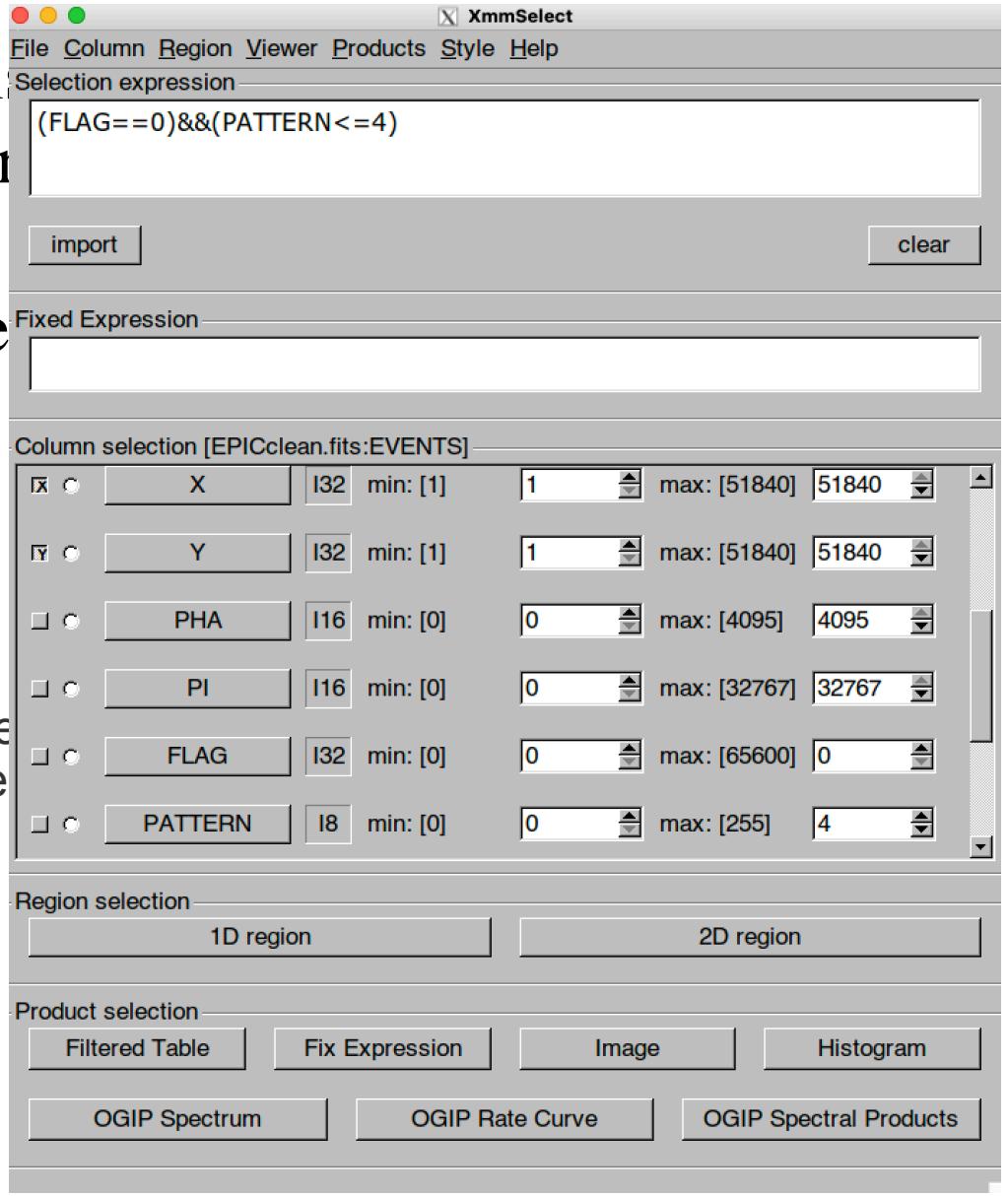
Follow SAS threads:

- How to filter EPIC event lists for flaring particle background
- Extraction of pn spectra from point-like sources

EPIC-PN spectra

Once we have the event files…. It i
data and produce images, spectra or

#> xmmselect table=<PNorMOS_e



Follow SAS threads:

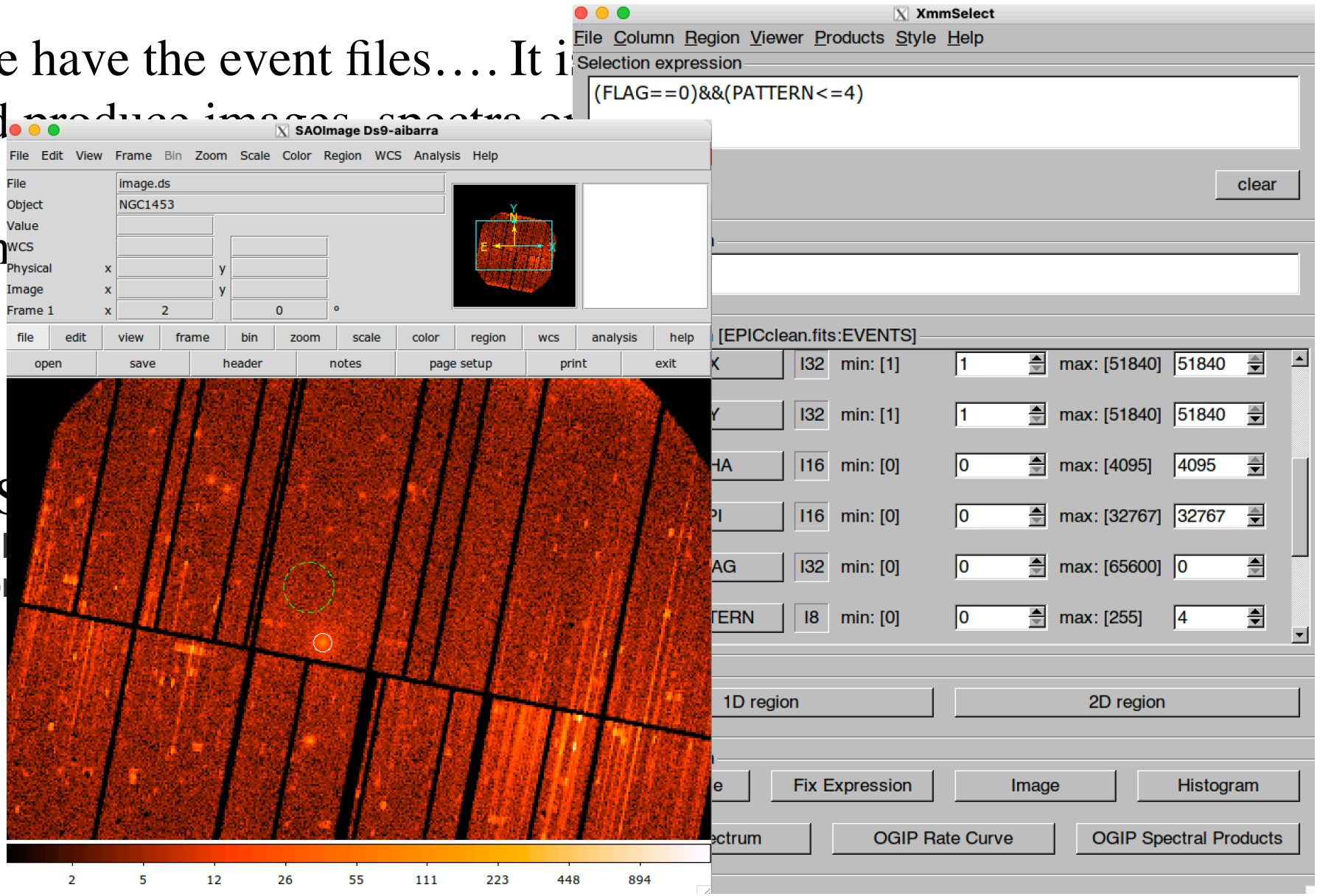
- How to filter EPIC event lists for flaring particle
- Extraction of pn spectra from point-like source

EPIC-PN spectra

Once we have the event files…. It i
data and produce images, spectra o

#> xmm

Follow S
- How to fi
- Extractio

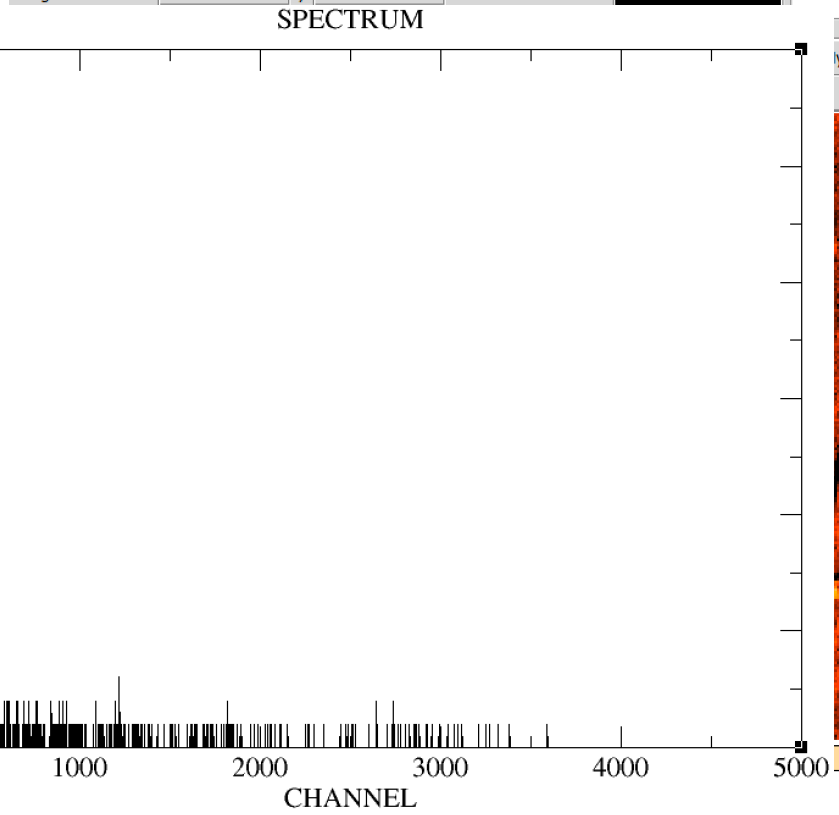


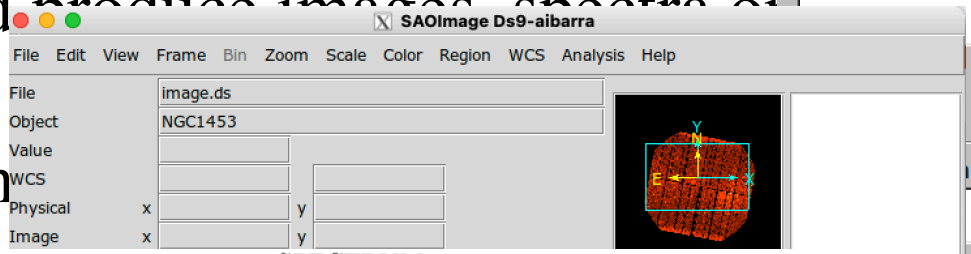
EPIC-PN spectra

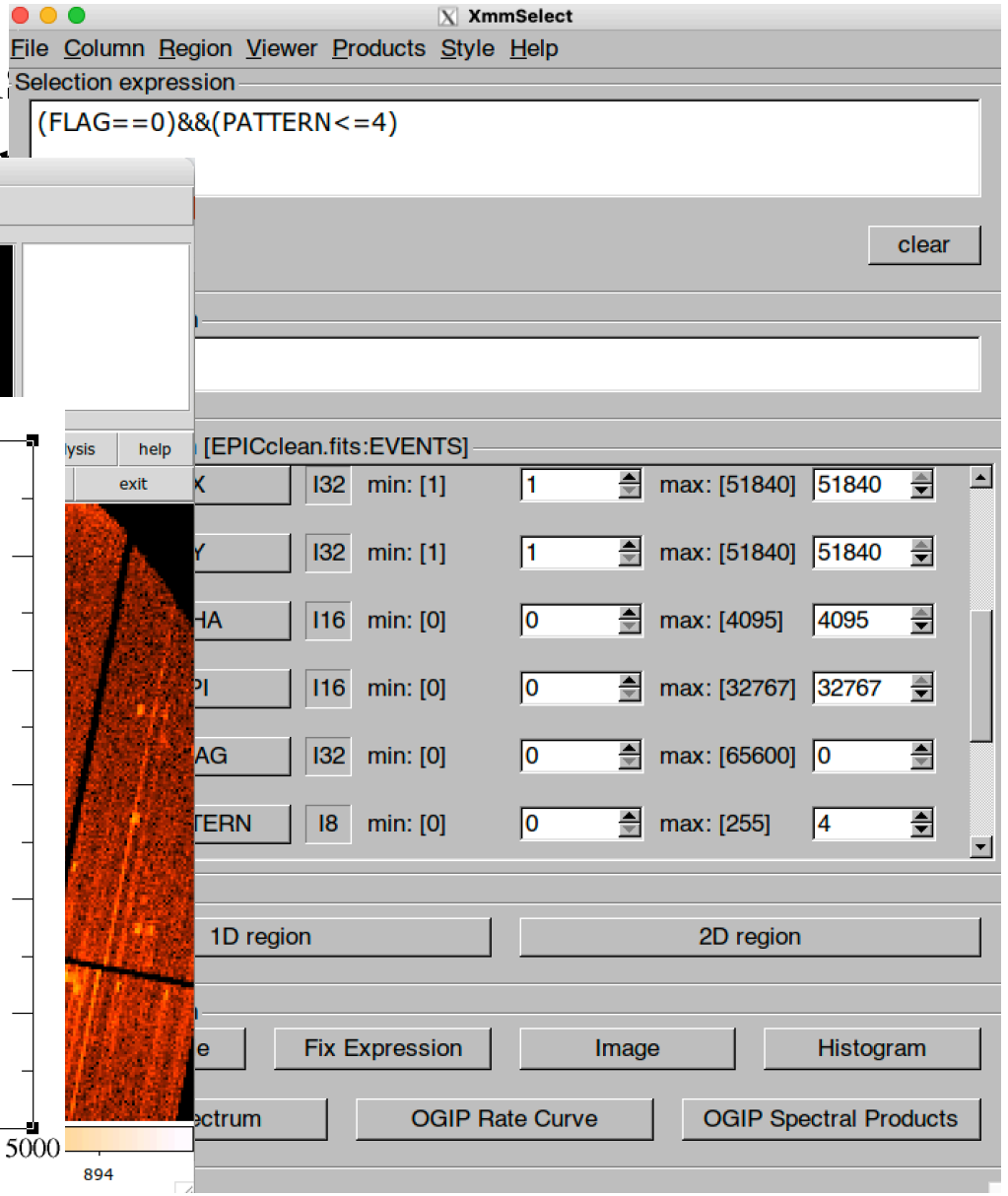
Once we have the event files…. It i
data and produce images, spectra o

#> xmm

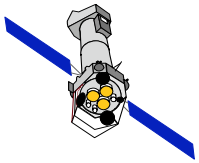
Fo
-
-

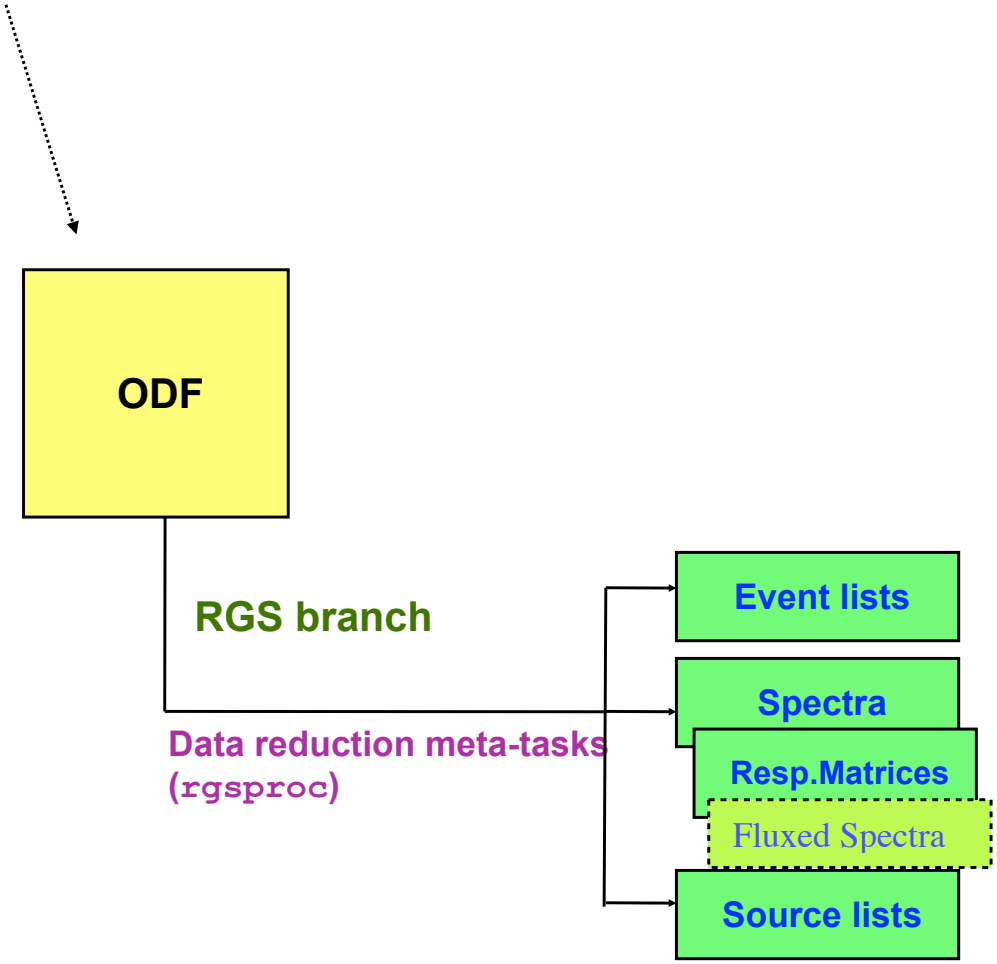




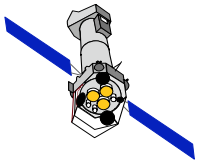


rgsproc

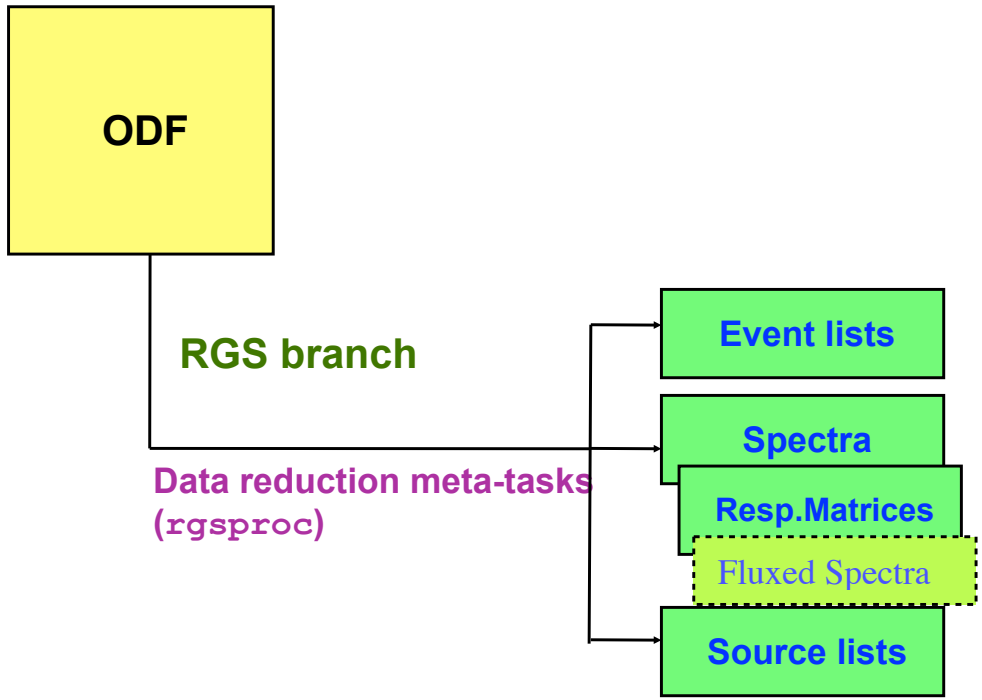



ODF
RGS branch
Data reduction meta-tasks
(rgsproc)
Event lists
Spectra
Resp.Matrices
Fluxed Spectra
Source lists

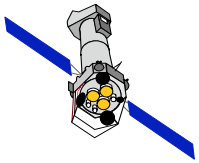
rgsproc



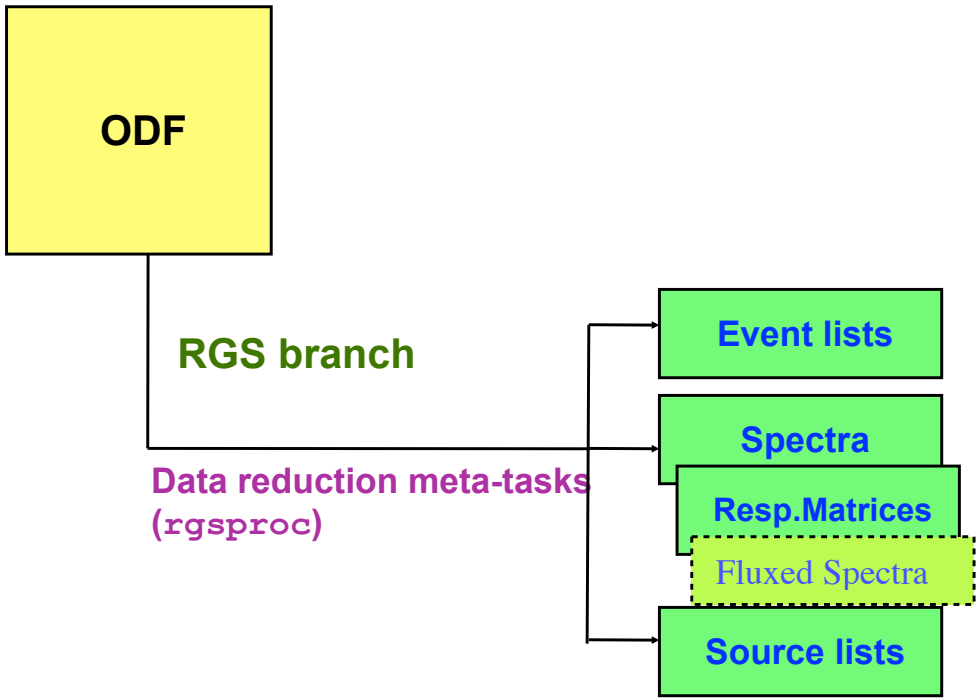
- meta-task: interface to 17 SAS tasks (that can also be run separately)
- controlled by ≈ 80 parameter switches
- five entry and final points (“processing stages”) >>
- produces filtered event lists, spectra and matrices
- the quality of the results depends critically on the source coordinates



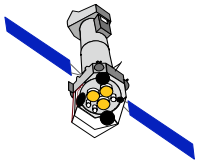
rgsproc



- 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") >>
  - *events*
  - *angles*
  - *filter*
  - *spectra*
  - *fluxing*
- produces filtered event lists, spectra and matrices
- the quality of the results depends critically on the source coordinates

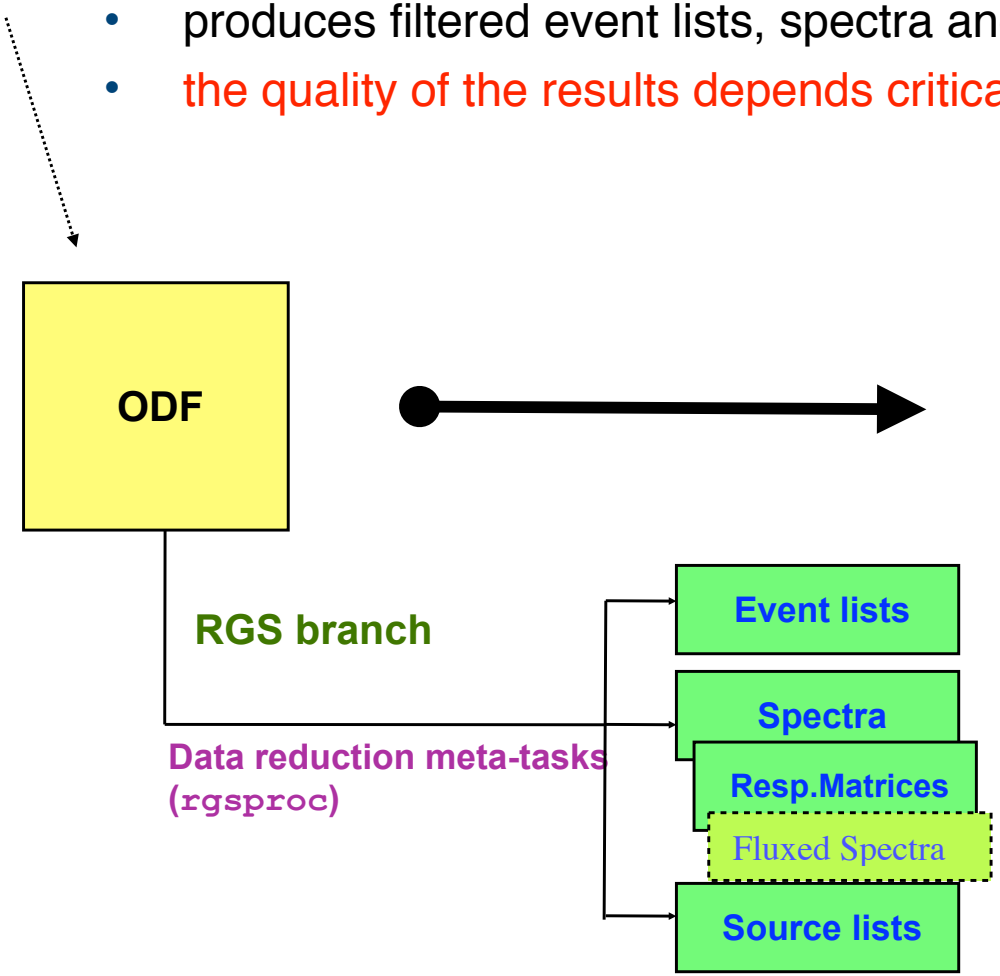


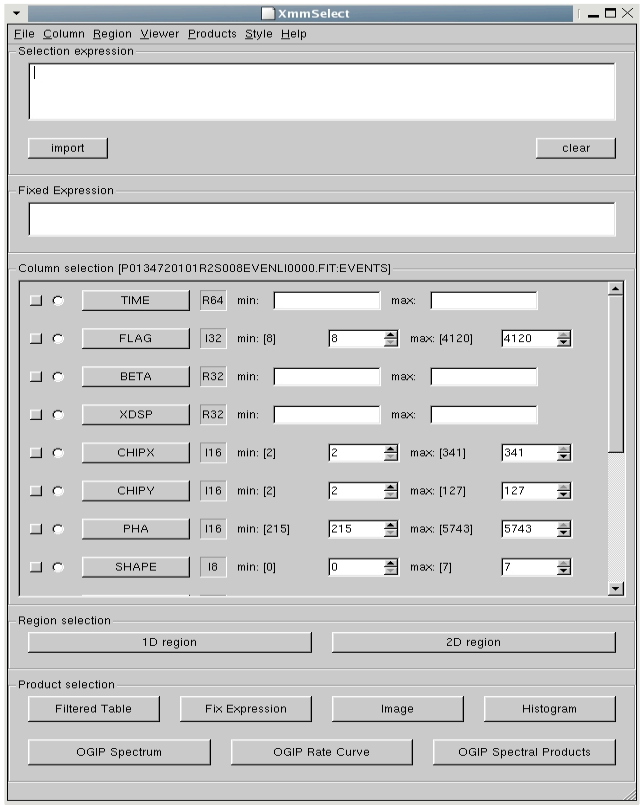
rgsproc



- 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



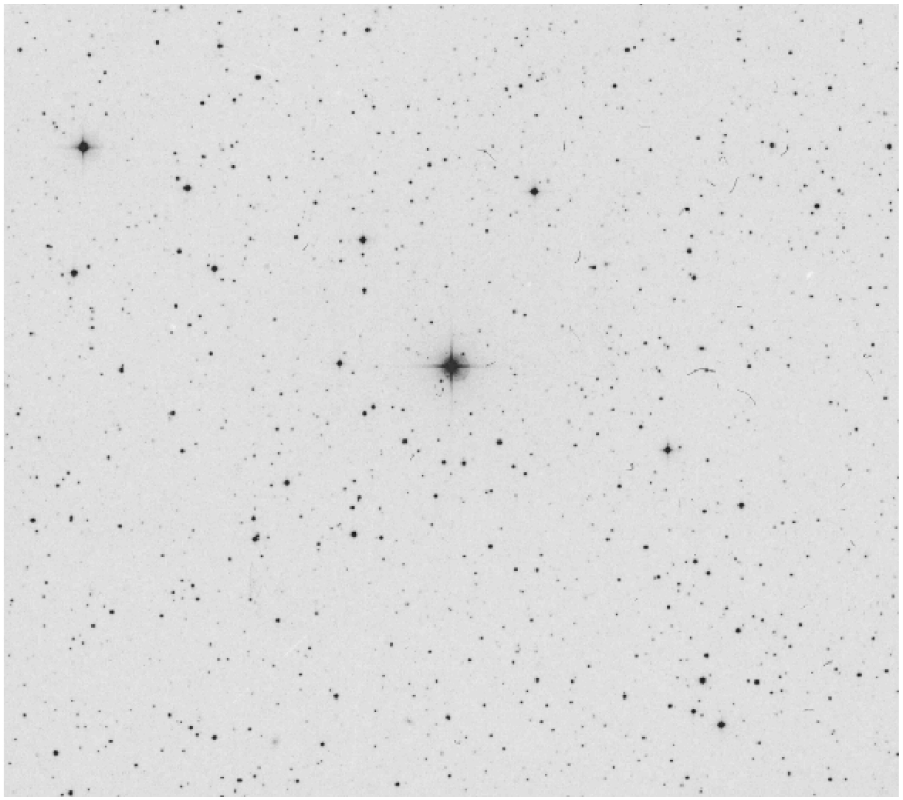


Simplified scheme of the RGS FOV

why the coordinates are so important!:

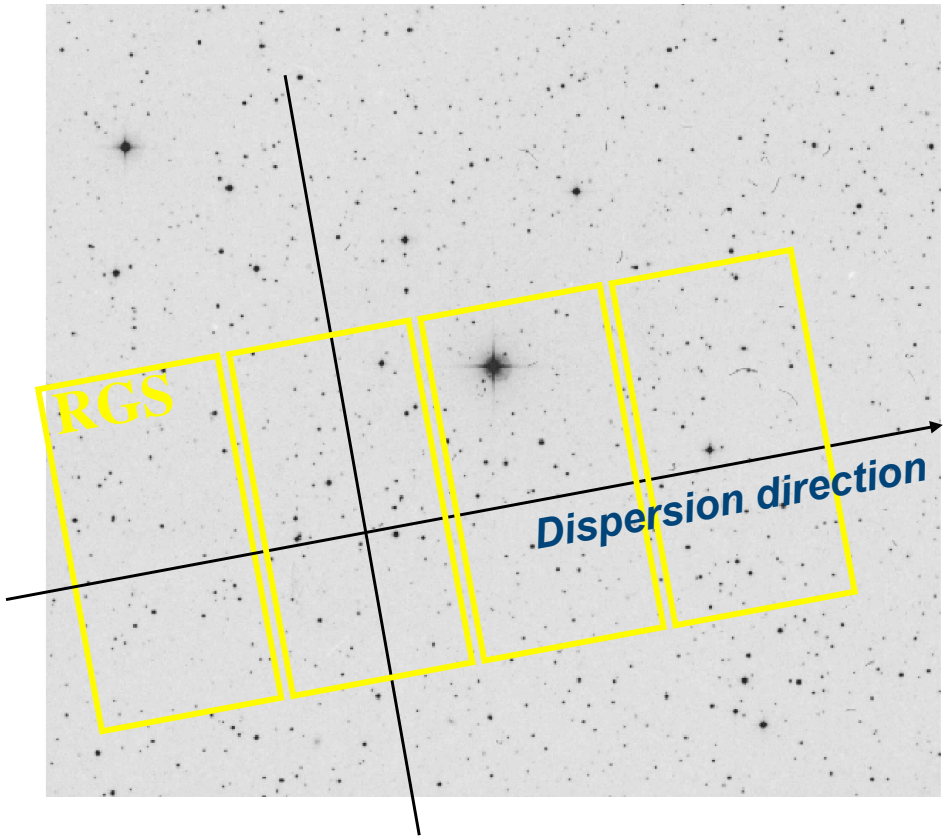
Simplified scheme of the RGS FOV

why the coordinates are so important!:



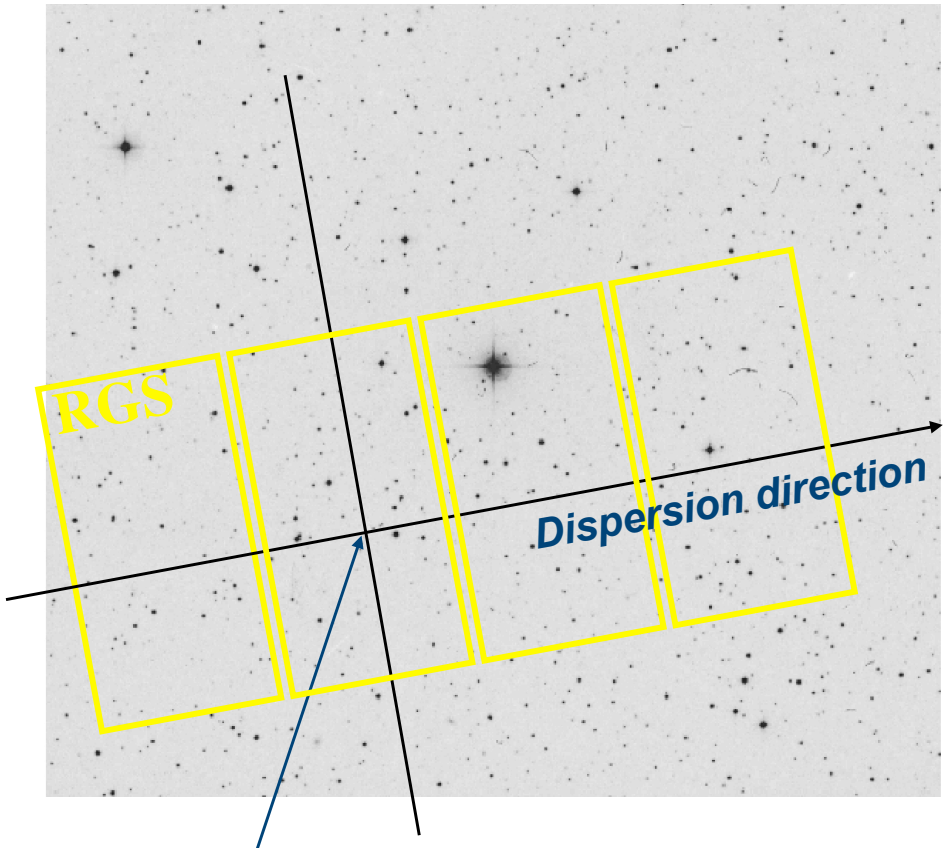
Simplified scheme of the RGS FOV

why the coordinates are so important!:



Simplified scheme of the RGS FOV

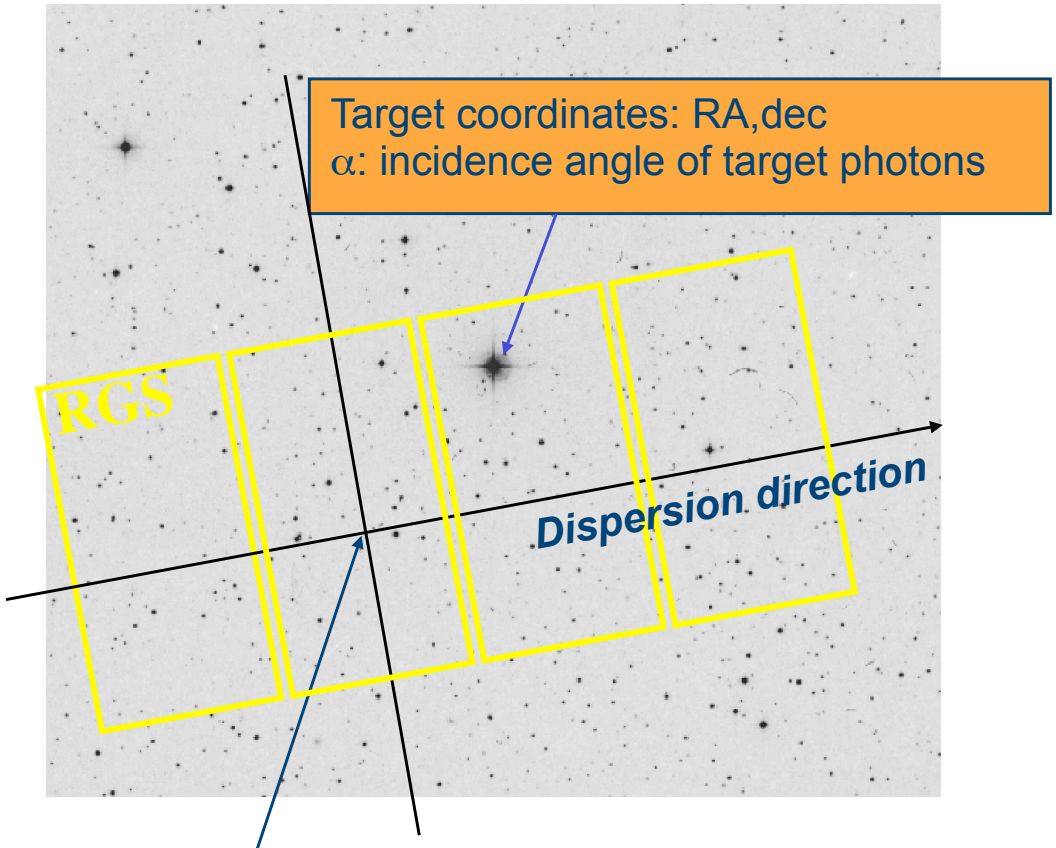
why the coordinates are so important!:



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

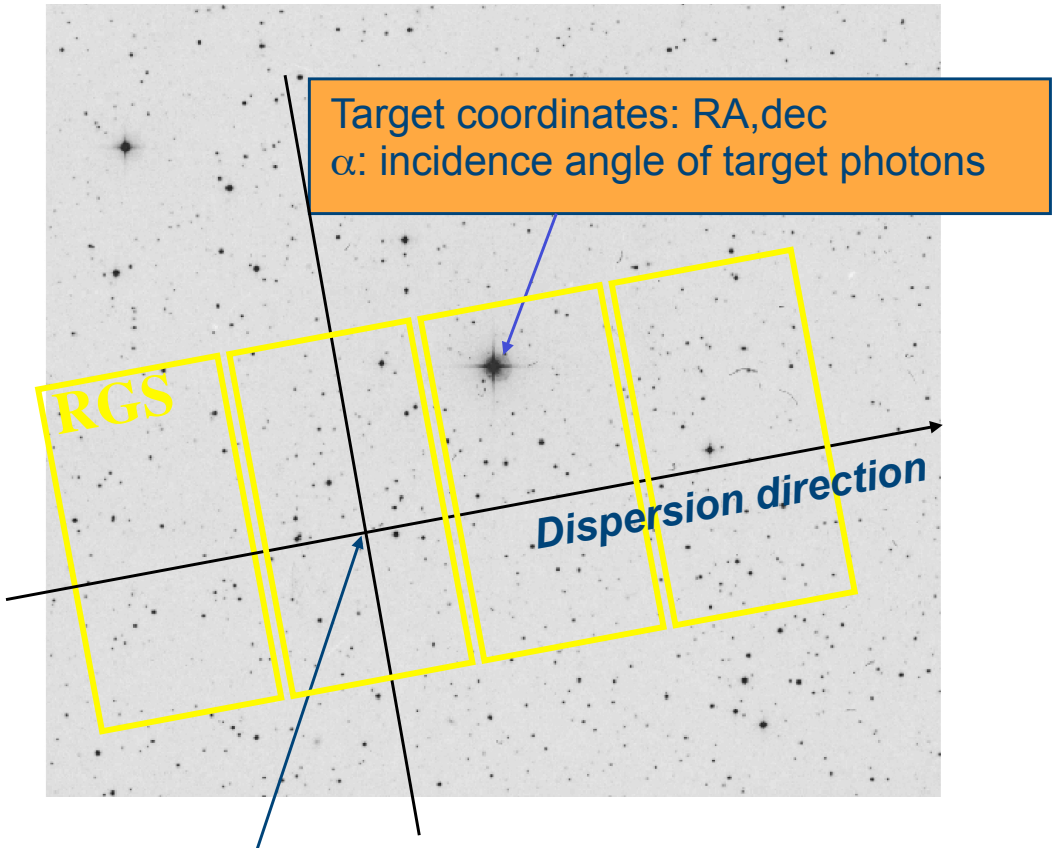
Simplified scheme of the RGS FOV

why the coordinates are so important!:



Simplified scheme of the RGS FOV

why the coordinates are so important!:



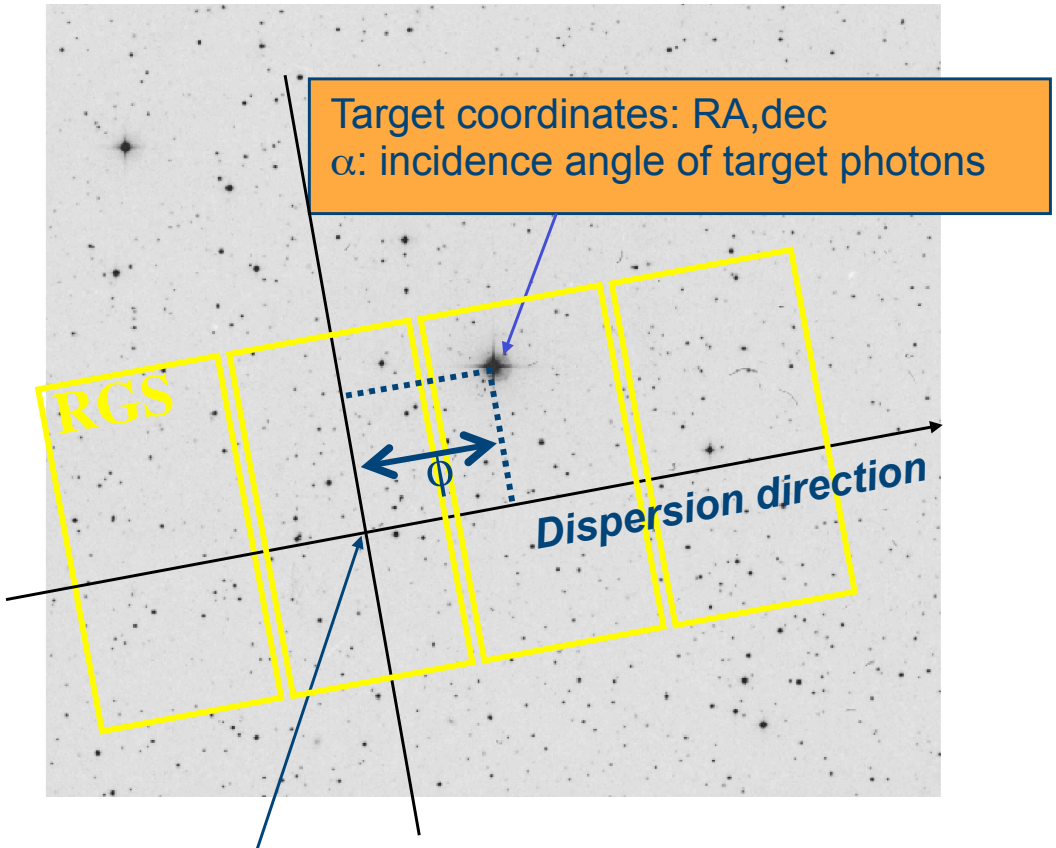
According to the grating equation

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

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

Simplified scheme of the RGS FOV

why the coordinates are so important!:



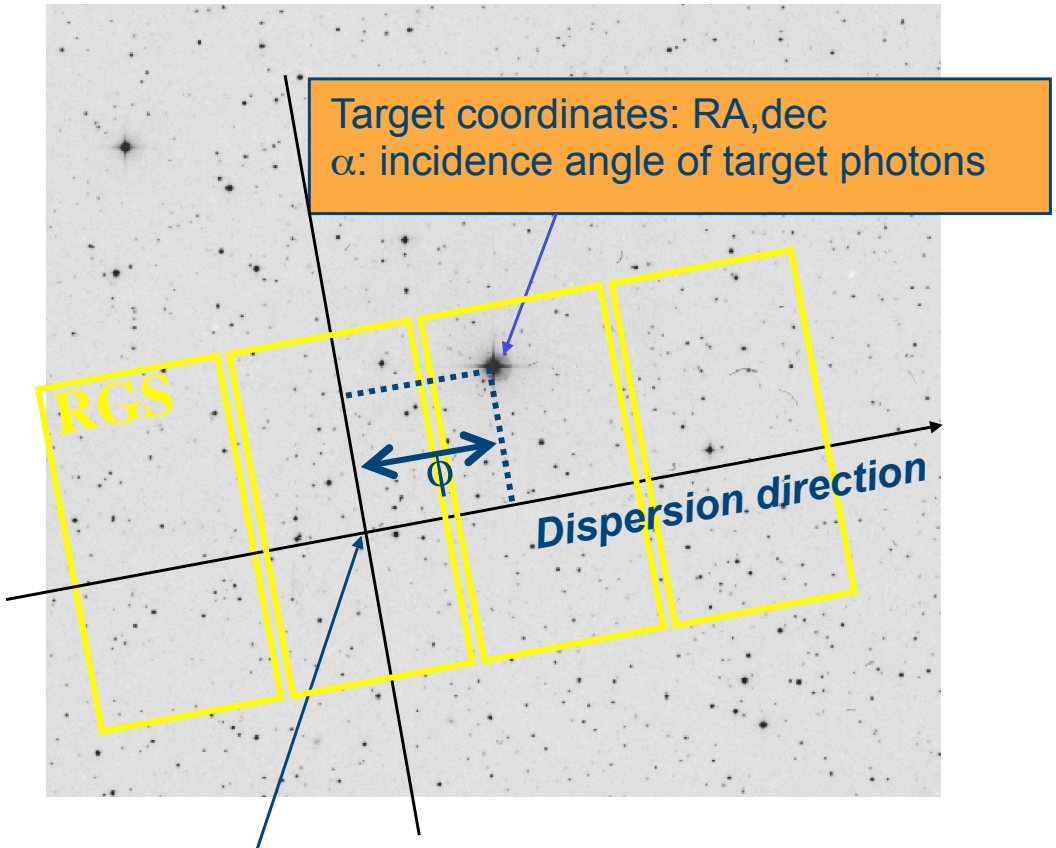
According to the grating equation

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

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

Simplified scheme of the RGS FOV

why the coordinates are so important!:



According to the grating equation

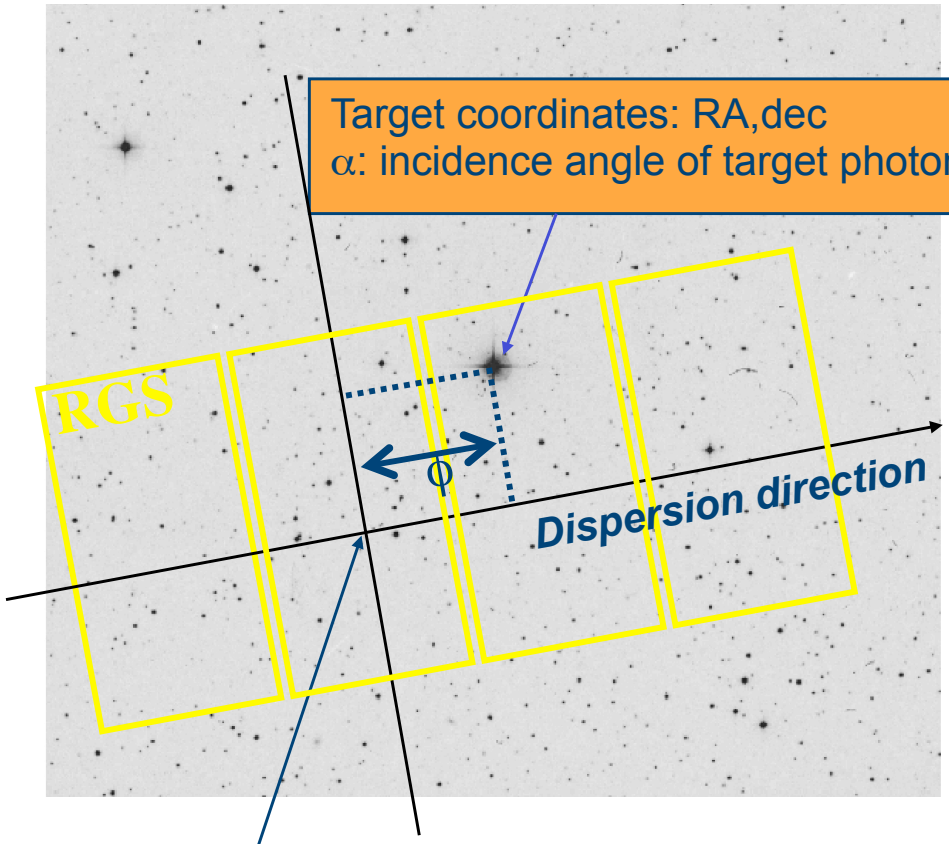
$$\lambda = (\cos \beta - \cos \alpha)\, d / m$$

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

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

Simplified scheme of the RGS FOV

why the coordinates are so important!:



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?

rgsproc: what does it do?

Stage	Task	Purpose	Output
Events	atthkgen	generates attitude file	Source list + intermediate combined event list
	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	

rgsproc: what does it do?

Stage	Task	Purpose	Output
Events	atthkgen	generates attitude file	Source list + intermediate combined event list
	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 (BETA_CORR, XDSP_CORR)	Aspect correction
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 foreach 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: what does it do?

Stage	Task	Purpose	Output
Events	atthkgen	generates attitude file	Source list + intermediate combined event list
	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 (BETA_CORR, XDSP_CORR)	Aspect correction
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 foreach 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: what does it do?

Stage	Task	Purpose	Output
Events	atthkgen	generates attitude file	Source list + intermediate combined event list
	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 (BETA_CORR, XDSP_CORR)	Aspect correction
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 foreach 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	

Source dependent

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 (BETA_CORR, 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 foreach 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: what does it do?

Stage	Task	Purpose	Output
Events	atthkgen	generates attitude file	Source list + intermediate combined event list
	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	

Source independent

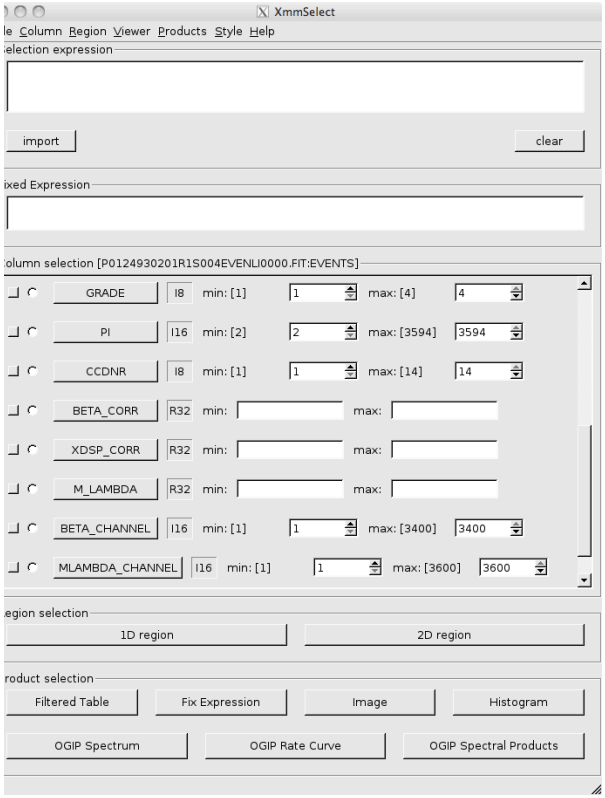
Angles	rgsangles	aspect correction (BETA_CORR, XDSP_CORR)	Aspect correction
Filter	rgsfilter	creates filtered event list, removing unwanted frames and events and adding exposure maps	Final event list
Spectra	rgsregions	computes background and source extraction regions foreach source	Source and background 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	

Source dependent

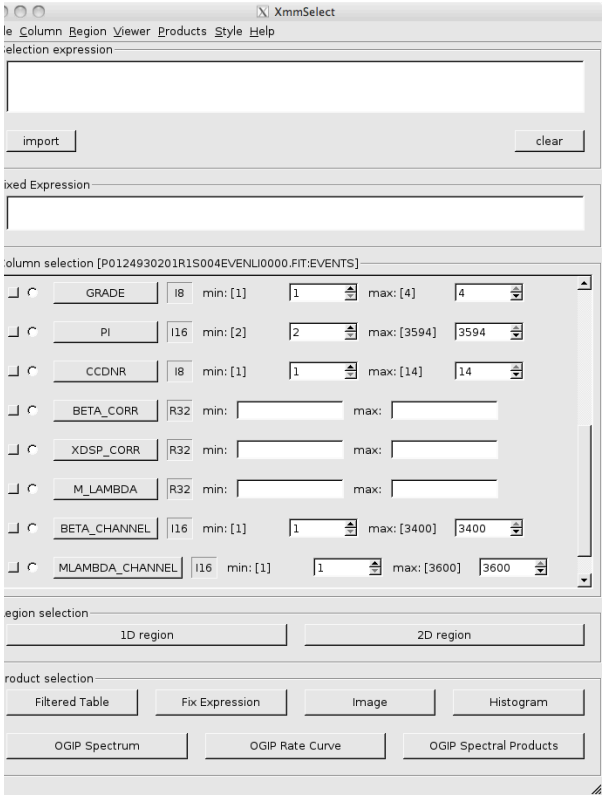
rgsproc: what does it do?

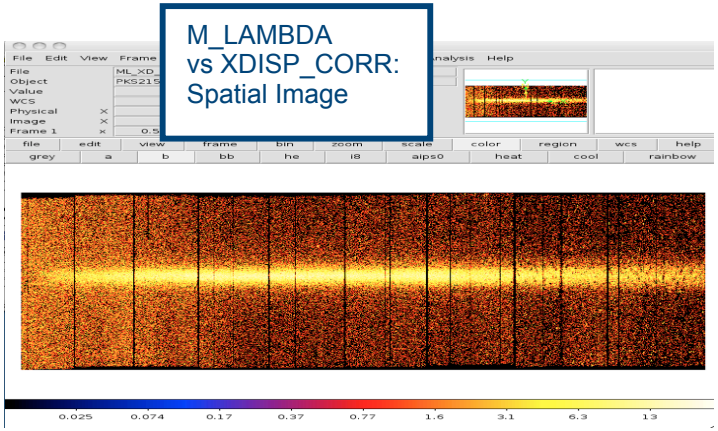
Stage	Task	Purpose	Output	
Events	atthkgen	generates attitude file	Source list + intermediate combined event list	Source independent
	attfilter	filters the attitude file		
	hkgtigen	generates housekeeping GTIs		
	rgsoffsetcalc	uses the diagnostic mode data for offset calculation		
	rgssources	creates 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

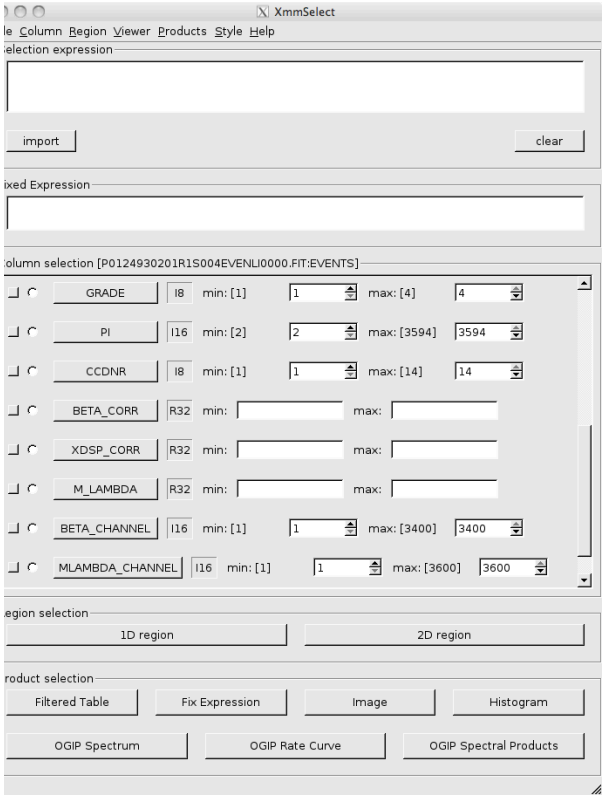


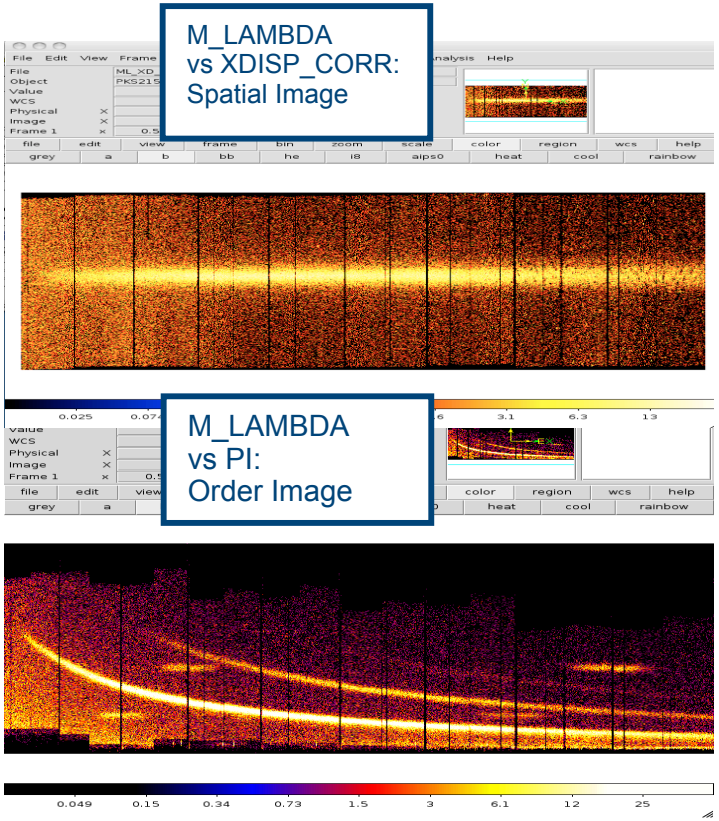
rgsproc does a lot



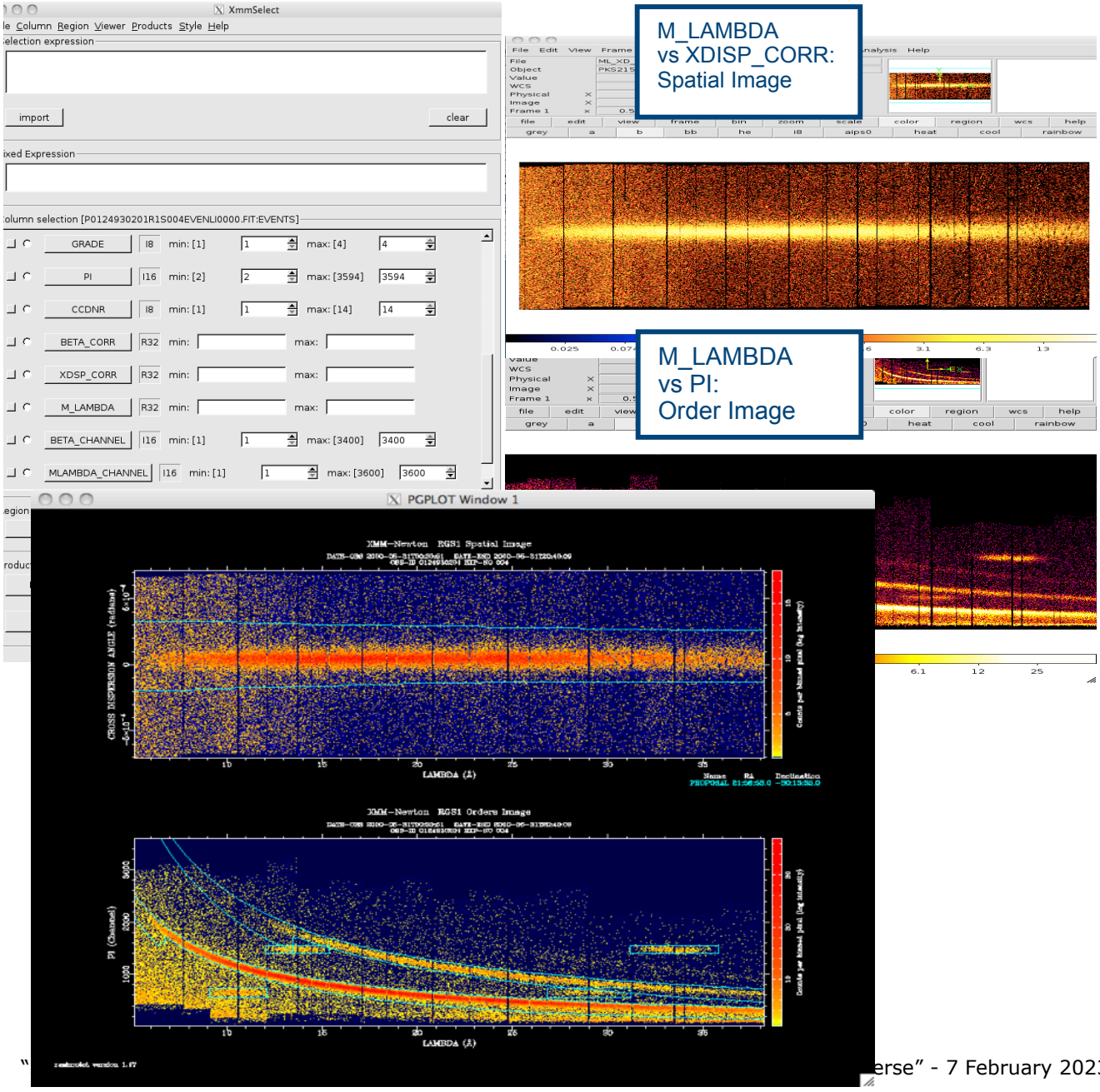


rgsproc does a lot





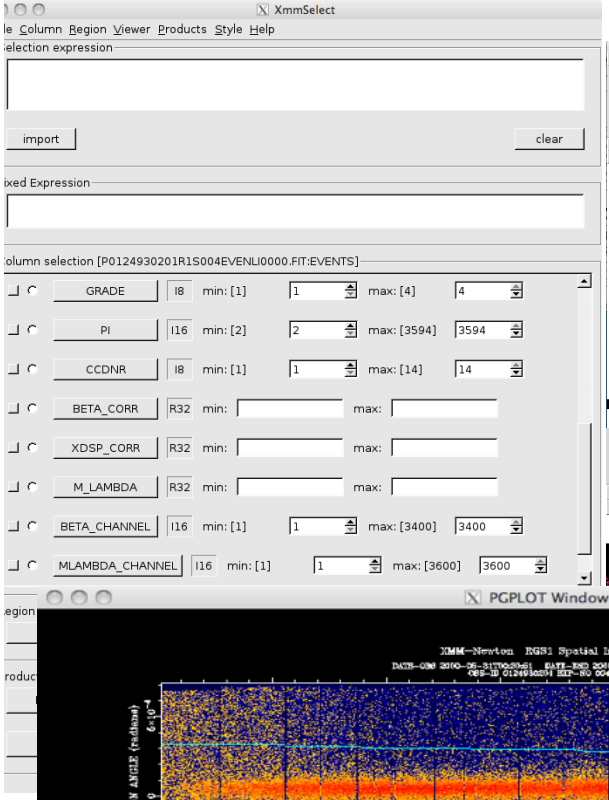
rgsproc does a lot

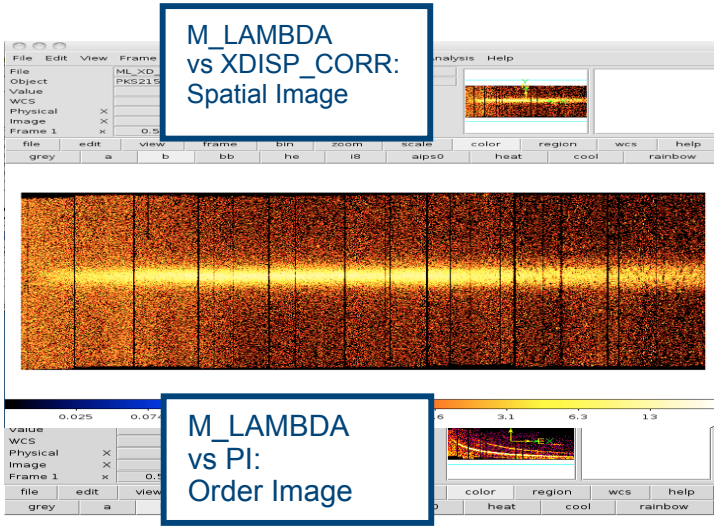


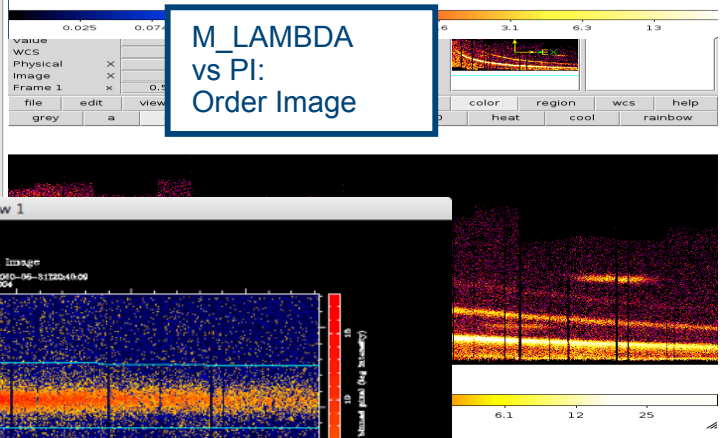
M_LAMBDA vs XDISP_CORR: Spatial Image

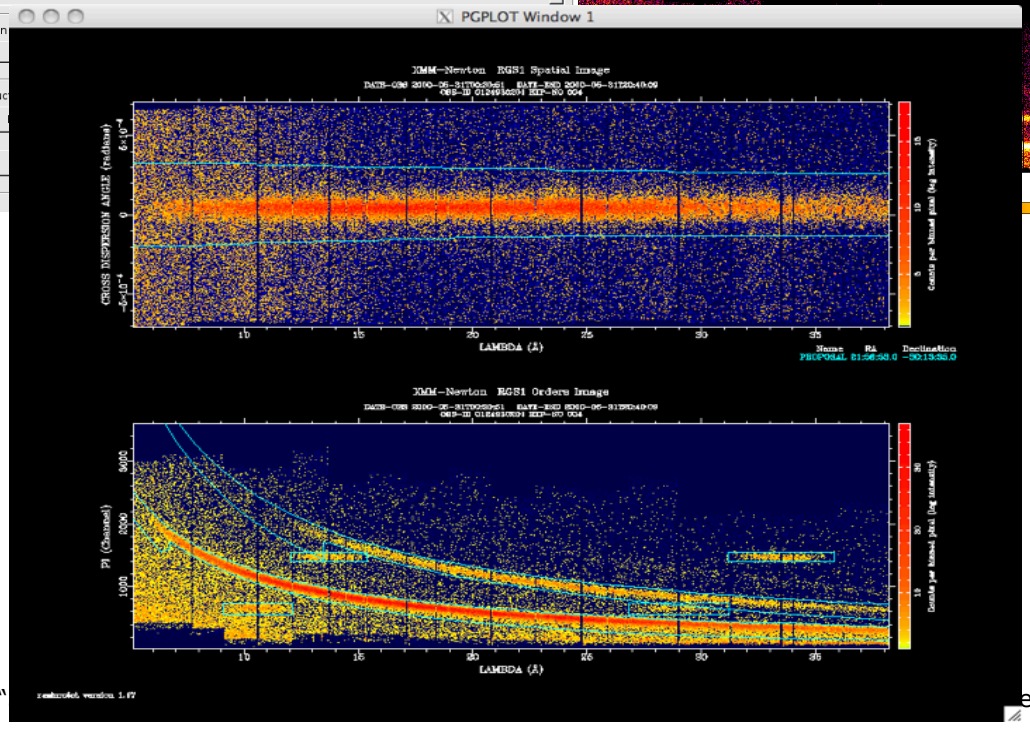
M_LAMBDA vs PI: Order Image

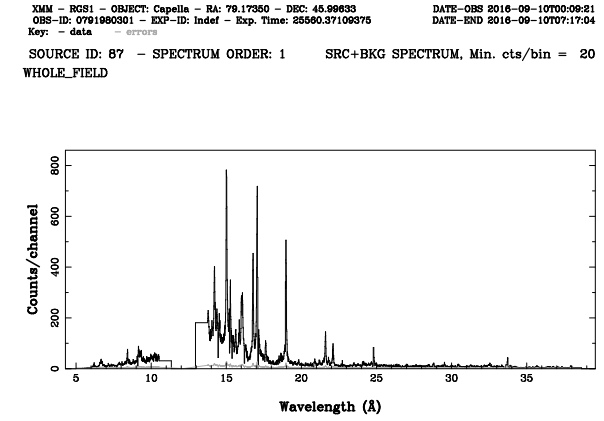
rgsproc does a lot

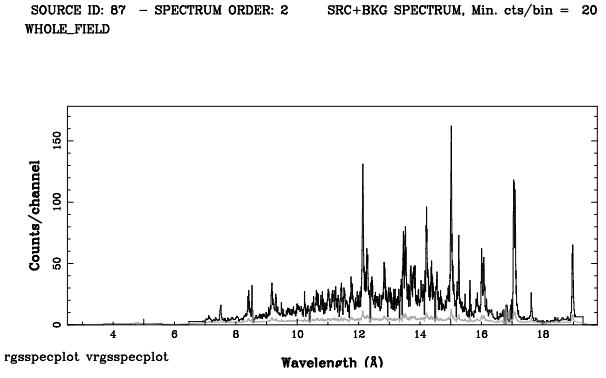






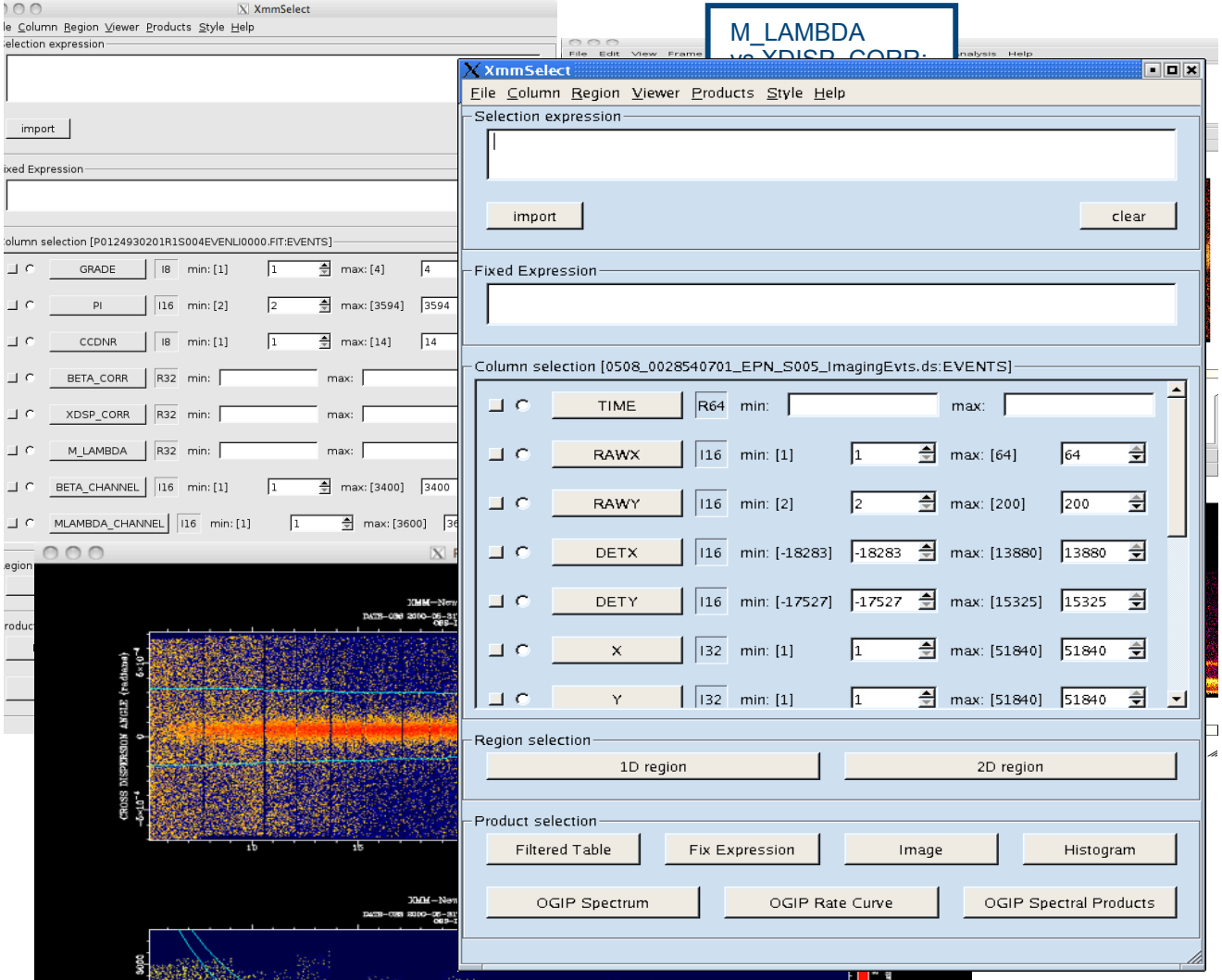


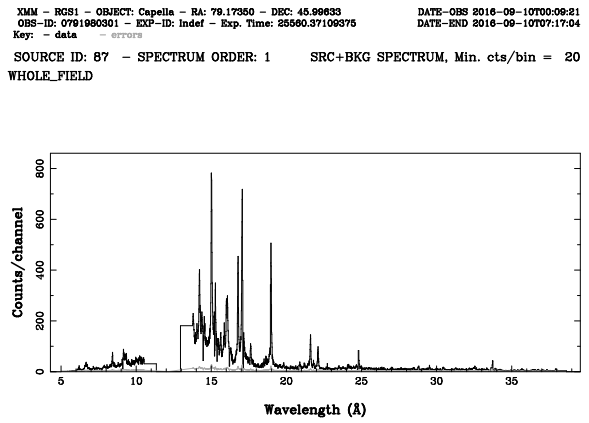


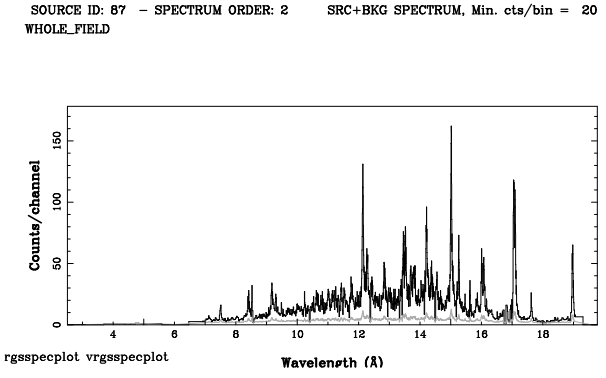


rgsspecplot vrgsspecplot

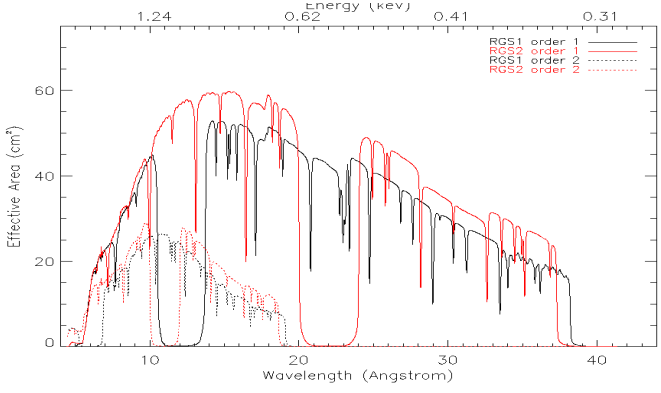
rgsproc does a lot







rgsspecplot vrgsspecplot



What do I get after processing?

For each RGS and exposure:

File	Content	rgsproc (default)	PPS
P0123456701R1S004EVENLI0000.FIT	Filtered Event List	Y	Y
P0123456701R1S004SRCLI_0000.FIT	Source List	Y (coord from proposal)	Y (coord from EPIC src list)
P0123456701R1S004BGSPEC1001.FIT P0123456701R1S004BGSPEC2001.FIT	Background Spectra (1st and 2nd order)	Y	Y
P0123456701R1S004SRSPEC1001.FIT P0123456701R1S004SRSPEC2001.FIT	Source Spectra (1st and 2nd order)	Y	Y
P0123456701R1S004SBSPEC1001.FIT P0123456701R1S004SBSPEC2001.FIT	Source+Bkg Spectra (1st and 2nd order)	Y	Y
P0123456701R1S004RSPMAT1001.FIT P0123456701R1S004RSPMAT2001.FIT	Response Matrices (1st and 2nd order)	Y	only for 1st order
P0123456701R1S004R[SR,BG]TSR1001.FIT P0123456701R1S004R[SR,BG]TSR2001.FIT	Time Series (1st and 2nd order) Source and Background	Y	Y
P0123456701R1S004IMAGE_0000.FIT P0123456701R1S004IMAGE_0000.PNG	Dispersion-CrossDispersion Image	N	Y
P0123456701R1S004ORDIMG0000.FIT P0123456701R1S004ORDIMG0000.PNG	Dispersion-Energy Image	N	Y
P0123456701R1S004EXPMAP0000.FIT	Exposure Map	Y	Y
P0123456701R1S004FBKTSR0000.FIT	Flare Background Timeseries	N	Y
P0123456701R1S004SRSPEC0001.PDF	Source Spectra (PDF)	N	Y

For each observation:

File	Content	rgsproc (default)	PPS
P0123456701OBX000fluxed1000.FIT P0123456701OBX000fluxed2000.FIT	Source Fluxed Spectra (1st and 2nd order)	Y	Y

Getting started with SAS I: the ODF

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

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: > odfingest odfdir=$SAS_ODF
will produce a Summary file <SUM> = RRRR_OOOOOOOOOO_SCX00000SUM.SAS
in the working directory

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: > odfingest odfdir=$SAS_ODF
will produce a Summary file <SUM> = RRRR_OOOOOOOOOO_SCX00000SUM.SAS
in the working directory

May 28, 01 15:11 **0072_0123720201_SCX00000SUM.SAS** Page 1/252

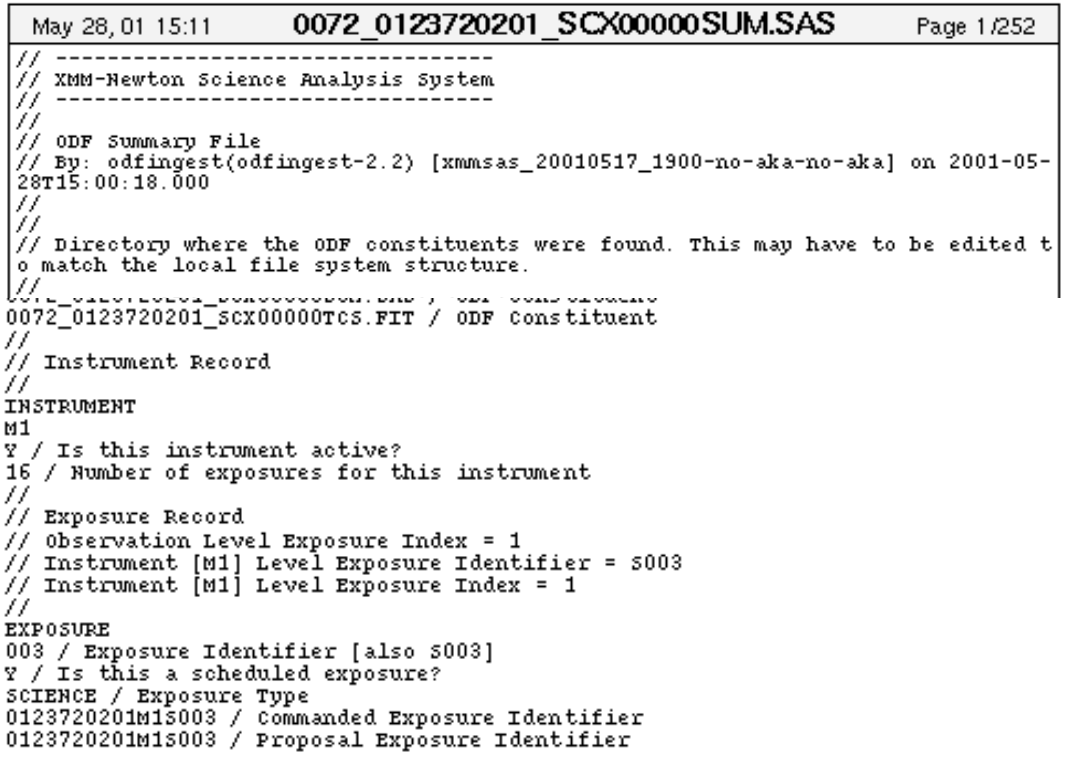
```
// ----------------------------------
// XMM-Newton Science Analysis System
// ----------------------------------
//
// ODF Summary File
// By: odfingest(odfingest-2.2) [xmmsas_20010517_1900-no-aka-no-aka] on 2001-05-
28T15:00:18.000
//
//
// Directory where the ODF constituents were found. This may have to be edited t
o match the local file system structure.
//
PATH /xvsas05/sasval/data/rawdata/0DS11/ABDor/
//
// Observation Record
//
OBSERVATION
0123720201 / Observation Identifier
0072 / Revolution Identifier
2000-05-01T02:30:21.000 / Observation Start Time
2000-05-01T19:46:33.000 / Observation End Time
//
// Number of files in the Observation Summary File, and their names
//
FILES
203 / Number of Files
0072_0123720201_M1S00300AUX.FIT / ODF Constituent
0072_0123720201_M1S00310IME.FIT / ODF Constituent
0072_0123720201_M1S00320IME.FIT / ODF Constituent
0072_0123720201_M1S00330IME.FIT / ODF Constituent
```

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: > odfingest odfdir=$SAS_ODF
will produce a Summary file <SUM> = RRRR_OOOOOOOOOO_SCX00000SUM.SAS
in the working directory


May 28, 01 15:11 0072_0123720201_SCX00000SUM.SAS Page 1/252

```
// ----------------------------------
// XMM-Newton Science Analysis System
// ----------------------------------
//
// ODF Summary File
// By: odfingest(odfingest-2.2) [xmmsas_20010517_1900-no-aka-no-aka] on 2001-05-
28T15:00:18.000
//
//
// Directory where the ODF constituents were found. This may have to be edited t
o match the local file system structure.
//
0072_0123720201_SCX00000TCS.FIT / ODF Constituent
//
// Instrument Record
//
INSTRUMENT
M1
Y / Is this instrument active?
16 / Number of exposures for this instrument
//
// Exposure Record
// Observation Level Exposure Index = 1
// Instrument [M1] Level Exposure Identifier = S003
// Instrument [M1] Level Exposure Index = 1
//
EXPOSURE
003 / Exposure Identifier [also S003]
Y / Is this a scheduled exposure?
SCIENCE / Exposure Type
0123720201M1S003 / Commanded Exposure Identifier
0123720201M1S003 / Proposal Exposure Identifier
```

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: > odfingest odfdir=$SAS_ODF
will produce a Summary file <SUM> = RRRR_OOOOOOOOOO_SCX00000SUM.SAS
in the working directory

May 28, 01 15:11 **0072_0123720201_SCX00000SUM.SAS** Page 1/252

```
// ---------------------------------
2000-05-01T10:45:05.000 / Actual End Time
//
// Configuration Record for M1
//
CONFIGURATION EPIC MOS
77 / Number of configuration parameters available
MODE = PrimeFullWindow / Instrument configuration
CALIBRATION_MODE_1 = PrimeFullWindow / Mode used to calibrate events from CCD 1
CALIBRATION_MODE_2 = PrimeFullWindow / Mode used to calibrate events from CCD 2
CALIBRATION_MODE_3 = PrimeFullWindow / Mode used to calibrate events from CCD 3
CALIBRATION_MODE_4 = PrimeFullWindow / Mode used to calibrate events from CCD 4
CALIBRATION_MODE_5 = PrimeFullWindow / Mode used to calibrate events from CCD 5
CALIBRATION_MODE_6 = PrimeFullWindow / Mode used to calibrate events from CCD 6
CALIBRATION_MODE_7 = PrimeFullWindow / Mode used to calibrate events from CCD 7
DATA_MODE_1 = Imaging / Data mode for CCD 1
DATA_MODE_2 = Imaging / Data mode for CCD 2
DATA_MODE_3 = Imaging / Data mode for CCD 3
DATA_MODE_4 = Imaging / Data mode for CCD 4
DATA_MODE_5 = Imaging / Data mode for CCD 5
DATA_MODE_6 = Imaging / Data mode for CCD 6
DATA_MODE_7 = Imaging / Data mode for CCD 7
FILTER_WHEEL = NOT VALID CS // Name of filter wheel position
FILTER_WHEEL_POSITION = 1580 // Hardware filter wheel position
FILTER = CalClosed // Filter
CLOCK_RESET_TIME_COARSE = 3826099 // Coarse component of the clock reset time
003 / Exposure Identifier [also S003]
Y / Is this a scheduled exposure?
SCIENCE / Exposure Type
0123720201M1S003 / Commanded Exposure Identifier
0123720201M1S003 / Proposal Exposure Identifier
```

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: > odfingest odfdir=$SAS_ODF
will produce a Summary file <SUM> = RRRR_OOOOOOOOOO_SCX00000SUM.SAS
in the working directory

After the Summary file has been produced:

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

May 28, 01 15:11 **0072_0123720201_SCX00000SUM.SAS** Page 1/252

```
// ---------------------------------
2000-05-01T10:45:05.000 / Actual End Time
//
// Configuration Record for M1
//
CONFIGURATION EPIC MOS
77 / Number of configuration parameters available
MODE = PrimeFullWindow / Instrument configuration
CALIBRATION_MODE_1 = PrimeFullWindow / Mode used to calibrate events from CCD 1
CALIBRATION_MODE_2 = PrimeFullWindow / Mode used to calibrate events from CCD 2
CALIBRATION_MODE_3 = PrimeFullWindow / Mode used to calibrate events from CCD 3
CALIBRATION_MODE_4 = PrimeFullWindow / Mode used to calibrate events from CCD 4
CALIBRATION_MODE_5 = PrimeFullWindow / Mode used to calibrate events from CCD 5
CALIBRATION_MODE_6 = PrimeFullWindow / Mode used to calibrate events from CCD 6
CALIBRATION_MODE_7 = PrimeFullWindow / Mode used to calibrate events from CCD 7
DATA_MODE_1 = Imaging / Data mode for CCD 1
DATA_MODE_2 = Imaging / Data mode for CCD 2
DATA_MODE_3 = Imaging / Data mode for CCD 3
DATA_MODE_4 = Imaging / Data mode for CCD 4
DATA_MODE_5 = Imaging / Data mode for CCD 5
DATA_MODE_6 = Imaging / Data mode for CCD 6
DATA_MODE_7 = Imaging / Data mode for CCD 7
FILTER_WHEEL = NOT VALID CS // Name of filter wheel position
FILTER_WHEEL_POSITION = 1580 // Hardware filter wheel position
FILTER = CalClosed // Filter
CLOCK_RESET_TIME_COARSE = 3826099 // Coarse component of the clock reset time
003 / Exposure Identifier [also S003]
Y / Is this a scheduled exposure?
SCIENCE / Exposure Type
0123720201M1S003 / Commanded Exposure Identifier
0123720201M1S003 / Proposal Exposure Identifier
```

Getting started with SAS II: the CCF

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

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 >

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 >

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 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 >

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

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

NOTICE: any file with extension CALINDEX is valid as SAS_CCF

odfbrowser

odfbrowser

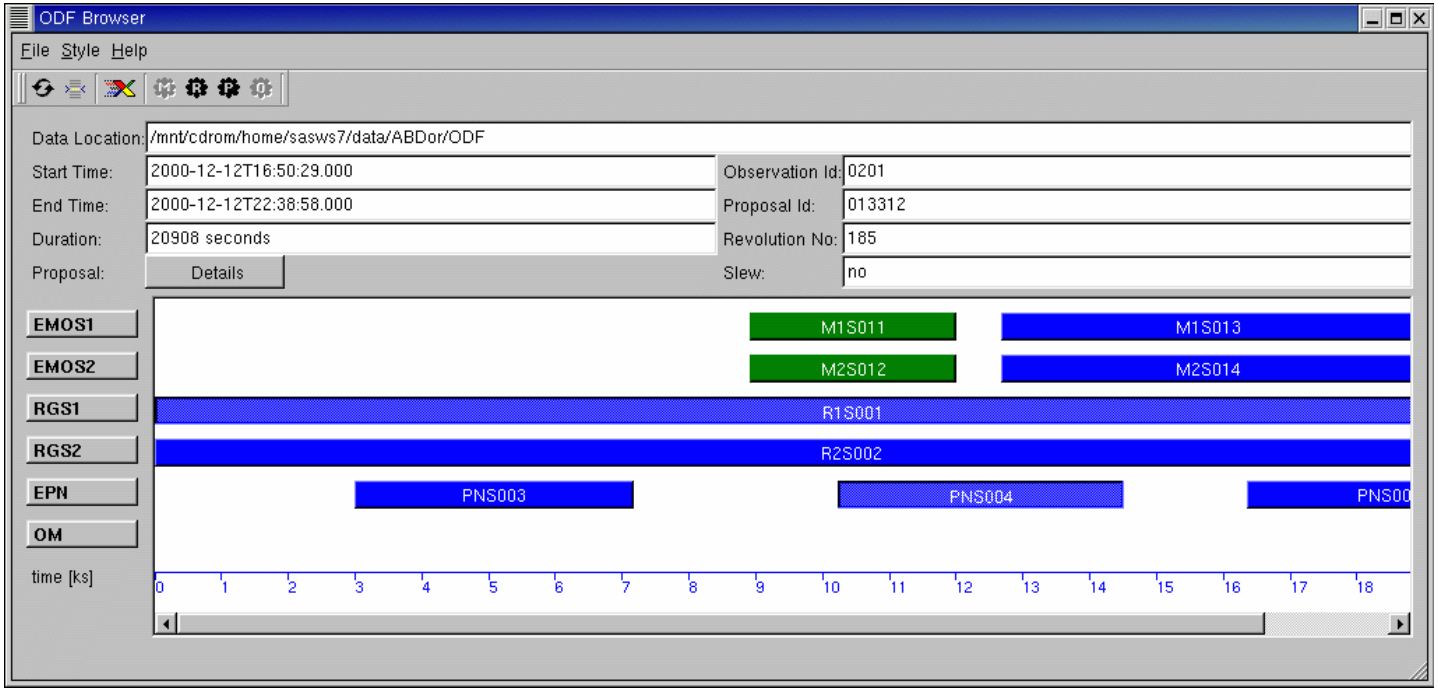
A task to view the contents of the ODF and more …

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

odfbrowser

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

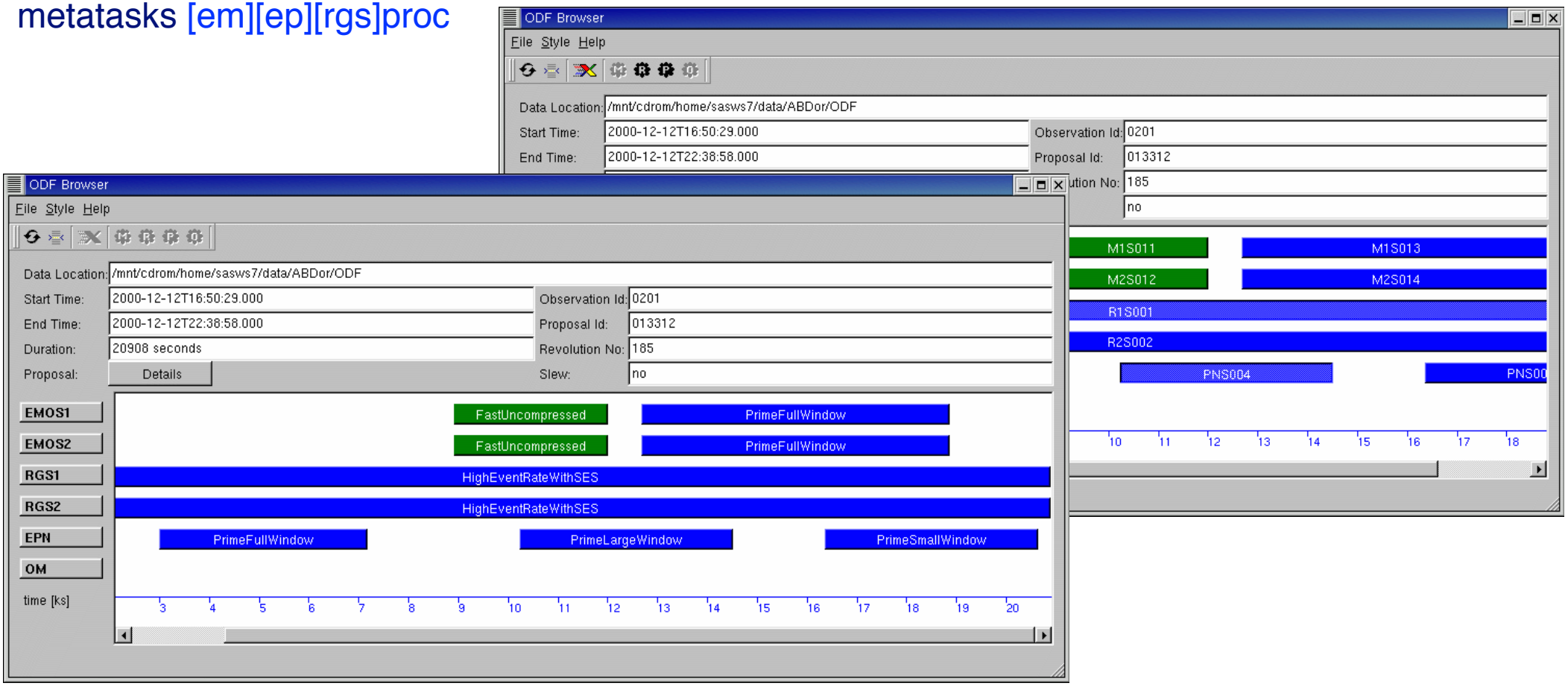
An ODF is constituted in the rule by hundreds of files. odfbrowser displays graphical summaries of an observation, allowing the user to select any number of exposures and launch the metatasks [em][ep][rgs]proc



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

cifbuild uses single CCF keywords:

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

cifbuild

cifbuild uses single CCF keywords:

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

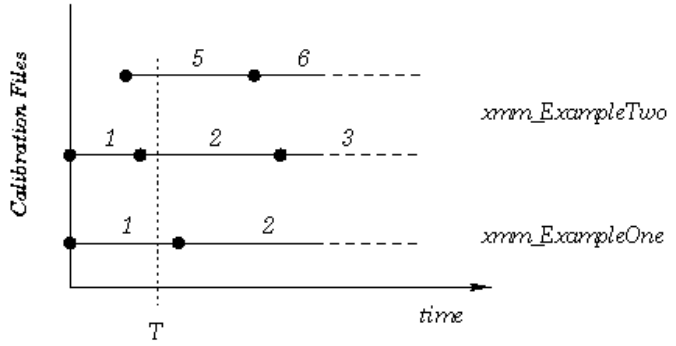


Figure 2: Current calibration file with two files: update. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0005.ccf*

cifbuild

cifbuild uses single CCF keywords:

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

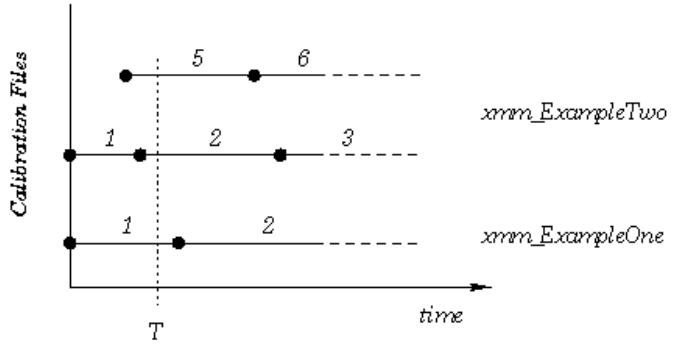


Figure 2: Current calibration file with two files: update. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0005.ccf*

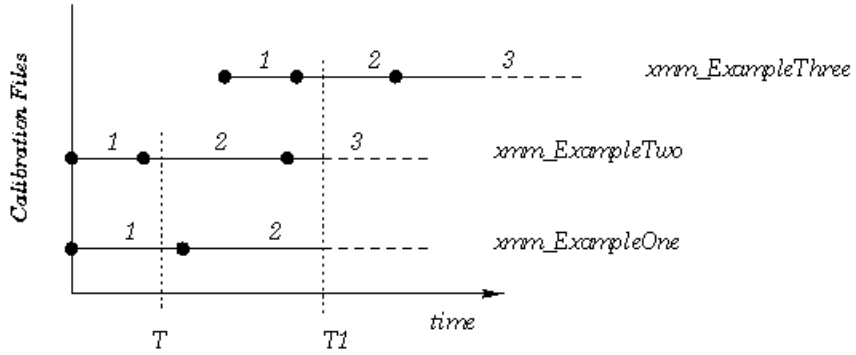


Figure 3: Current calibration file with three files. At the time $T1$ the current calibration file consists of *xmm_ExampleOne_0002.ccf* and *xmm_ExampleTwo_0003.ccf* and *xmm_ExampleThree_0002.ccf*

cifbuild

cifbuild uses single CCF keywords:

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

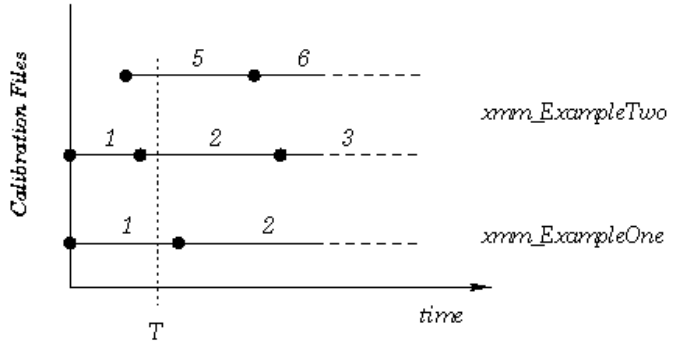


Figure 2: Current calibration file with two files: update. At the time T the current calibration file consists of *xmm_ExampleOne_0001.ccf* and *xmm_ExampleTwo_0005.ccf*

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.

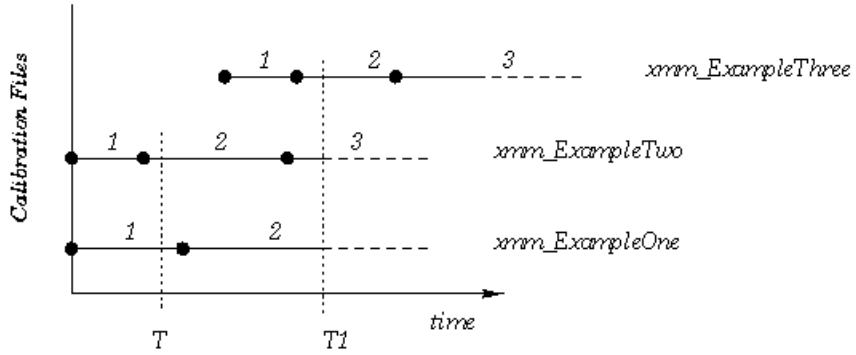


Figure 3: Current calibration file with three files. At the time $T1$ the current calibration file consists of *xmm_ExampleOne_0002.ccf* and *xmm_ExampleTwo_0003.ccf* and *xmm_ExampleThree_0002.ccf*

CIF / CCF on the web

- On the XMM-Newton calibration web pages:
  - updated cif can be generated on-line
    and compared to the one you generate
  - required (missing) CCF constituents can also be downloaded
  - local CCF library can be mirrored from XMM web site via the rsync or mirror commands (see doc web pages)
  - there is a “valid” CCF library (1.1 GB instead of > 6 GB)
    (if you don’t have it yet you can get it from me...)

- CCF release notes shall be consulted, at least periodically.
  - Subscribing to the CCF mailing list is also useful, to get the RNs only when there is something new.

Getting started with SAS III: the basic steps

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>

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>

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>

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>

```
sasbld03n:~> source setsas.csh

sasversion:- Executing (routine): sasversion  -w 1 -V 4
sasversion:- sasversion (sasversion-1.3)  [xmmsas_20211130_0941-20.0.0] started:  2023-02-06T22:30:32.000
sasversion:- XMM-Newton SAS release and build information:

SAS release: xmmsas_20211130_0941-20.0.0
Compiled on: Tue Nov 30 09:49:16 CET 2021
Compiled by: sasbuild@sasbld03n
Platform   : Ubuntu18.04

SAS-related environment variables that are set:

SAS_DIR = /sas/Linux/Ubuntu18.04/64/xmmsas_20211130_0941
SAS_PATH = /sas/Linux/Ubuntu18.04/64/xmmsas_20211130_0941
SAS_CCFPATH = /ccf/valid

sasversion:- sasversion (sasversion-1.3)  [xmmsas_20211130_0941-20.0.0] ended:    2023-02-06T22:30:32.000

Do not forget to define SAS_CCFPATH, SAS_CCF and SAS_ODF


SAS 20.0.0 ready !

sasbld03n:~>
```

Getting started with SAS III: the basic steps

Getting started with SAS III: the basic steps

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

```
. <SAS-DIR>/setsas.csh

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>

Getting started with SAS III: the basic steps

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

. <SAS-DIR>/setsas.csh 

```
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>

Getting started with SAS III: the basic steps

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

. <SAS-DIR>/setsas.csh

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>

```
aibarra@sasbld03n:~$ . ./setsas.sh

sasversion:- Executing (routine): sasversion  -w 1 -V 4
sasversion:- sasversion (sasversion-1.3)  [xmmsas_20211130_0941-20.0.0] started:  2023-02-06T22:26:33.000
sasversion:- XMM-Newton SAS release and build information:

SAS release: xmmsas_20211130_0941-20.0.0
Compiled on: Tue Nov 30 09:49:16 CET 2021
Compiled by: sasbuild@sasbld03n
Platform   : Ubuntu18.04

SAS-related environment variables that are set:

SAS_DIR = /sas/Linux/Ubuntu18.04/64/xmmsas_20211130_0941
SAS_PATH = /sas/Linux/Ubuntu18.04/64/xmmsas_20211130_0941
SAS_CCFPATH = /ccf/valid

sasversion:- sasversion (sasversion-1.3)  [xmmsas_20211130_0941-20.0.0] ended:    2023-02-06T22:26:33.000

Do not forget to define SAS_CCFPATH, SAS_CCF and SAS_ODF


SAS 20.0.0 ready !
```

Getting started with SAS III: the basic steps

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

. <SAS-DIR>/setsas.csh

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>

```
aibarra@sasbld03n:~$ . ./setsas.sh

sasversion:- Executing (routine): sasversion  -w 1 -V 4
sasversion:- sasversion (sasversion-1.3)  [xmmsas_20211130_0941-20.0.0] started:  2023-02-06T22:26:33.000
sasversion:- XMM-Newton SAS release and build information:

SAS release: xmmsas_20211130_0941-20.0.0
Compiled on: Tue Nov 30 09:49:16 CET 2021
Compiled by: sasbuild@sasbld03n
Platform   : Ubuntu18.04

SAS-related environment variables that are set:

SAS_DIR = /sas/Linux/Ubuntu18.04/64/xmmsas_20211130_0941
SAS_PATH = /sas/Linux/Ubuntu18.04/64/xmmsas_20211130_0941
SAS_CCFPATH = /ccf/valid

sasversion:- sasversion (sasversion-1.3)  [xmmsas_20211130_0941-20.0.0] ended:    2023-02-06T22:26:33.000

Do not forget to define SAS_CCFPATH, SAS_CCF and SAS_ODF


SAS 20.0.0 ready !
```

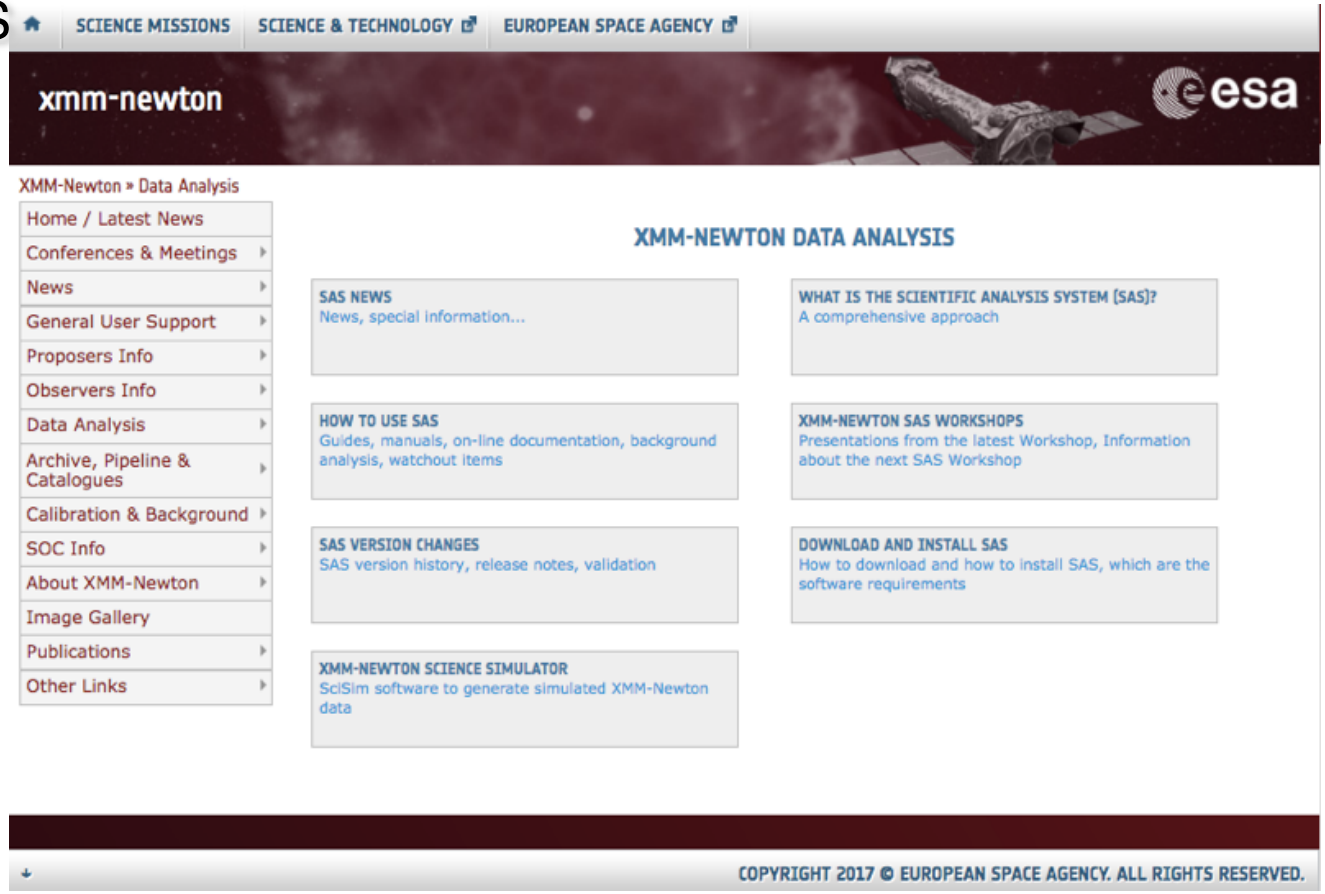
You can locate all these commands into (c)sh command files

Getting started with SAS IV: all the information

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

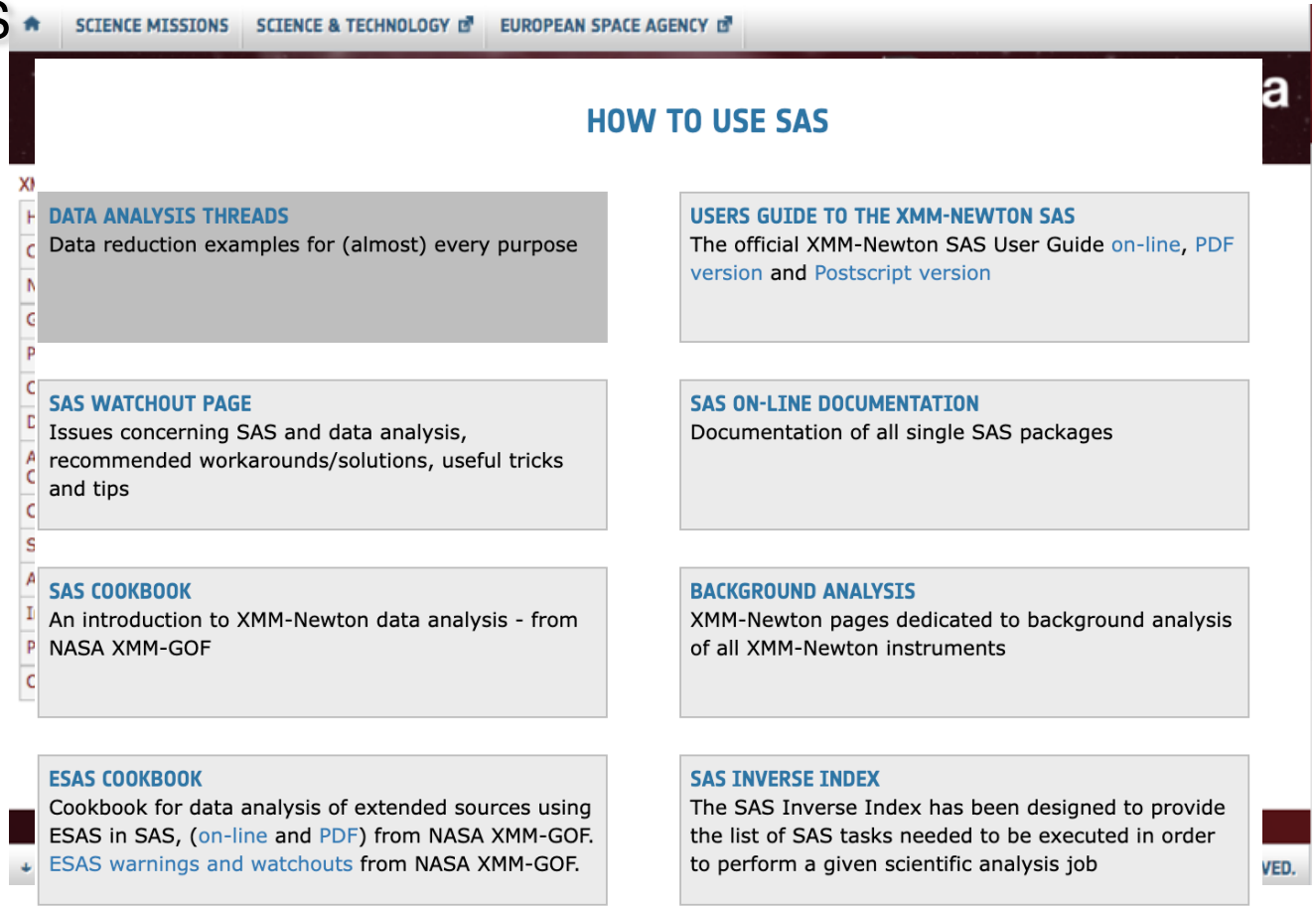
Getting started with SAS IV: all the information

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



Getting started with SAS IV: all the information

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

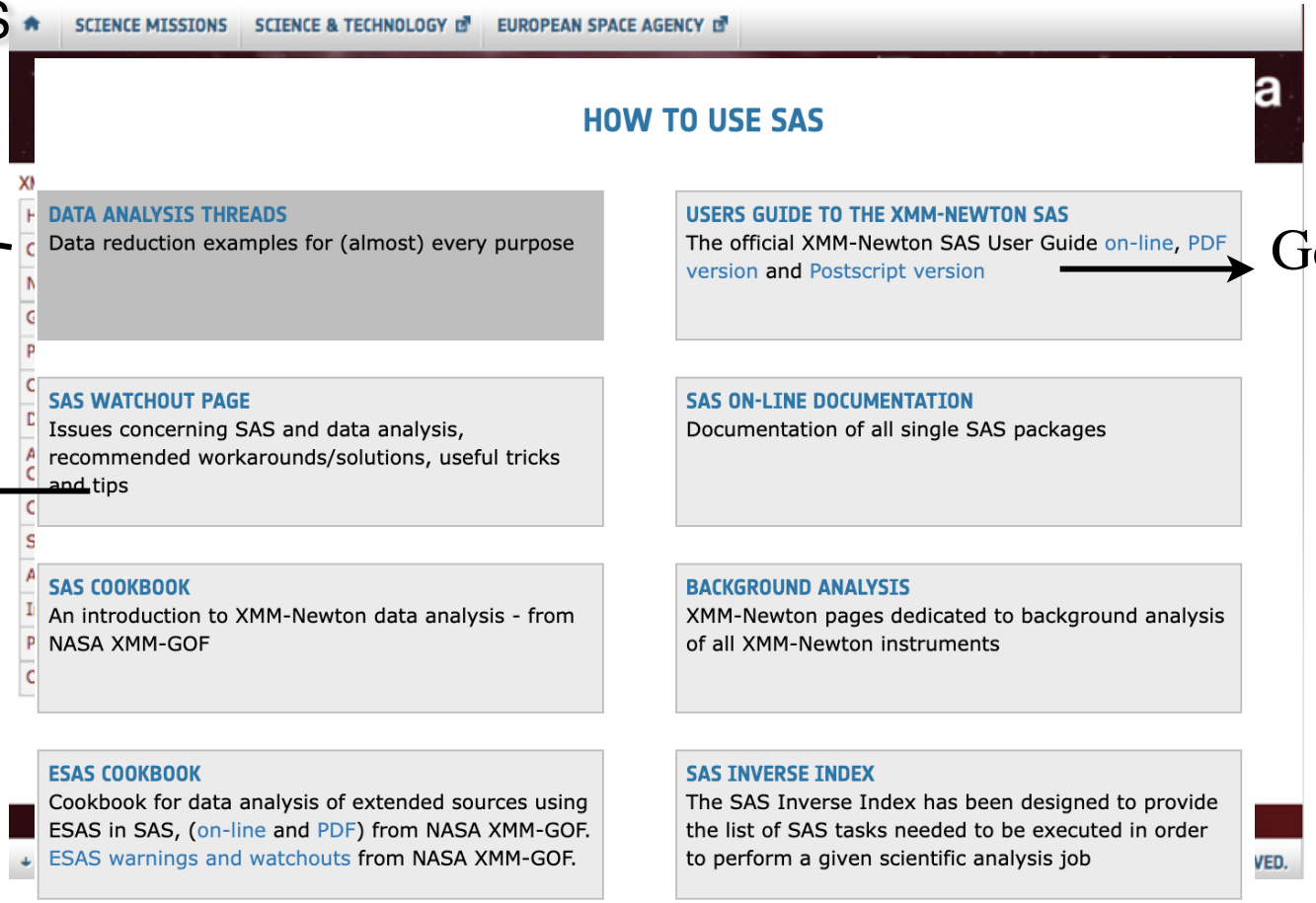


Getting started with SAS IV: all the information

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

Analysis recipes

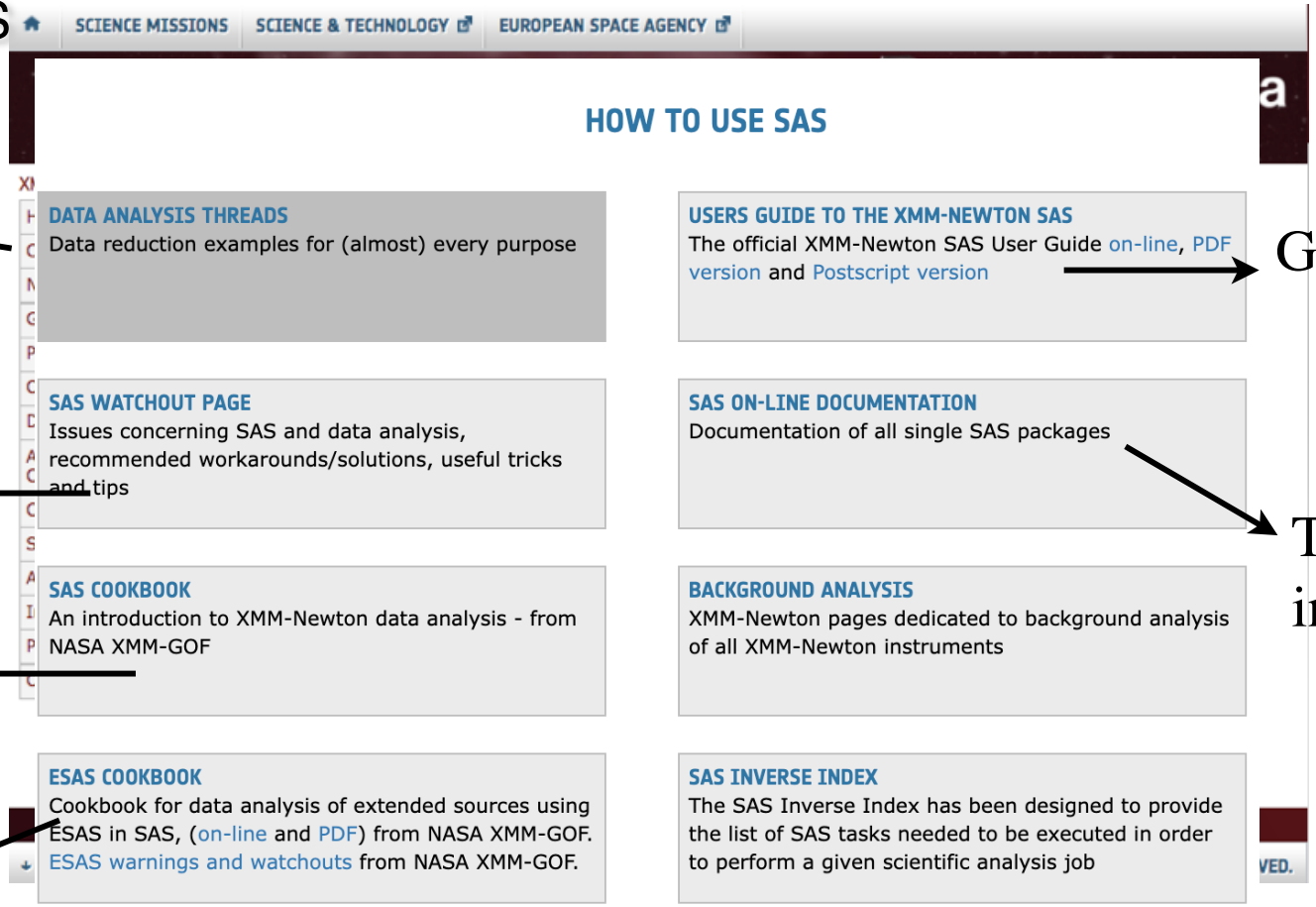
Problems??



General guide

Getting started with SAS IV: all the information

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



Analysis recipes

Problems??

ABC guide

Extended sources

General guide

Task by task info

The SAS threads

The SAS threads

XMM-Newton » Data Analysis » How to use SAS

- Home / Latest News
- Conferences & Meetings
- News
- General User Support
- Proposers Info
- Observers Info
- Data Analysis
- Archive, Pipeline & Catalogues
- Calibration & Background
- SOC Info
- About XMM-Newton
- Image Gallery
- Publications
- Other Links

HOW TO USE SAS

DATA ANALYSIS THREADS
Data reduction examples for (almost) every purpose

USERS GUIDE TO THE XMM-NEWTON SAS
The official XMM-Newton SAS User Guide on-line, PDF version and Postscript version

SAS WATCHOUT PAGE
Issues concerning SAS and data analysis, recommended workarounds/solutions, useful tricks and tips

SAS ON-LINE DOCUMENTATION
Documentation of all single SAS packages

SAS COOKBOOK
An introduction to XMM-Newton data analysis - from NASA XMM-GOF

BACKGROUND ANALYSIS
XMM-Newton pages dedicated to background analysis of all XMM-Newton instruments

ESAS COOKBOOK
Cookbook for data analysis of extended sources using ESAS in SAS, (on-line and PDF) from NASA XMM-GOF. ESAS warnings and watchouts from NASA XMM-GOF.

SAS INVERSE INDEX
The SAS Inverse Index has been designed to provide the list of SAS tasks needed to be executed in order to perform a given scientific analysis job

The SAS threads

SCIENCE MISSIONS | EUROPEAN SPACE AGENCY | SCIENCE &TECHNOLOGY | SIGN IN

xmm-newton

SAS THREADS

JUPYTER NOTEBOOK THREADS

With the infrastructure of Python introduced in SAS 19, three experimental threads have been released under Jupyter Notebooks. These threads are not intended to be complete but to serve the purpose of illustrating how to use the Python interface to run SAS from a Jupyter Notebook

SAS Start-up and event list manipulation		
- SAS start-up thread in Python	Jupyter Notebook	html
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists	Jupyter Notebook	html
- How to filter EPIC event lists for flaring particle background	Jupyter Notebook	html

The SAS threads

SCIENCE MISSIONS | EUROPEAN SPACE AGENCY | SCIENCE &TECHNOLOGY | SIGN IN

xmm-newton

XM

Home
Confe
News
Gene
Propo
Obse
Data
Archi
Catal
Calib

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

, three experimental threads have been released under Jupyter Notebooks. These
he purpose of illustrating how to use the Python interface to run SAS from a Jupyter

	Jupyter Notebook	html
ncatenated EPIC	Jupyter Notebook	html
ground	Jupyter Notebook	html

The SAS threads

SCIENCE MISSIONS | EUROPEAN SPACE AGENCY | SCIENCE &TECHNOLOGY | SIGN IN

xmm-newton

XM

Home
Confe
News
Gene
Propo
Obse
Data
Archi
Catal
Calib

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

, three experimental threads have been released under Jupyter Notebooks. These
he purpose of illustrating how to use the Python interface to run SAS from a Jupyter

RGS RELATED THREADS

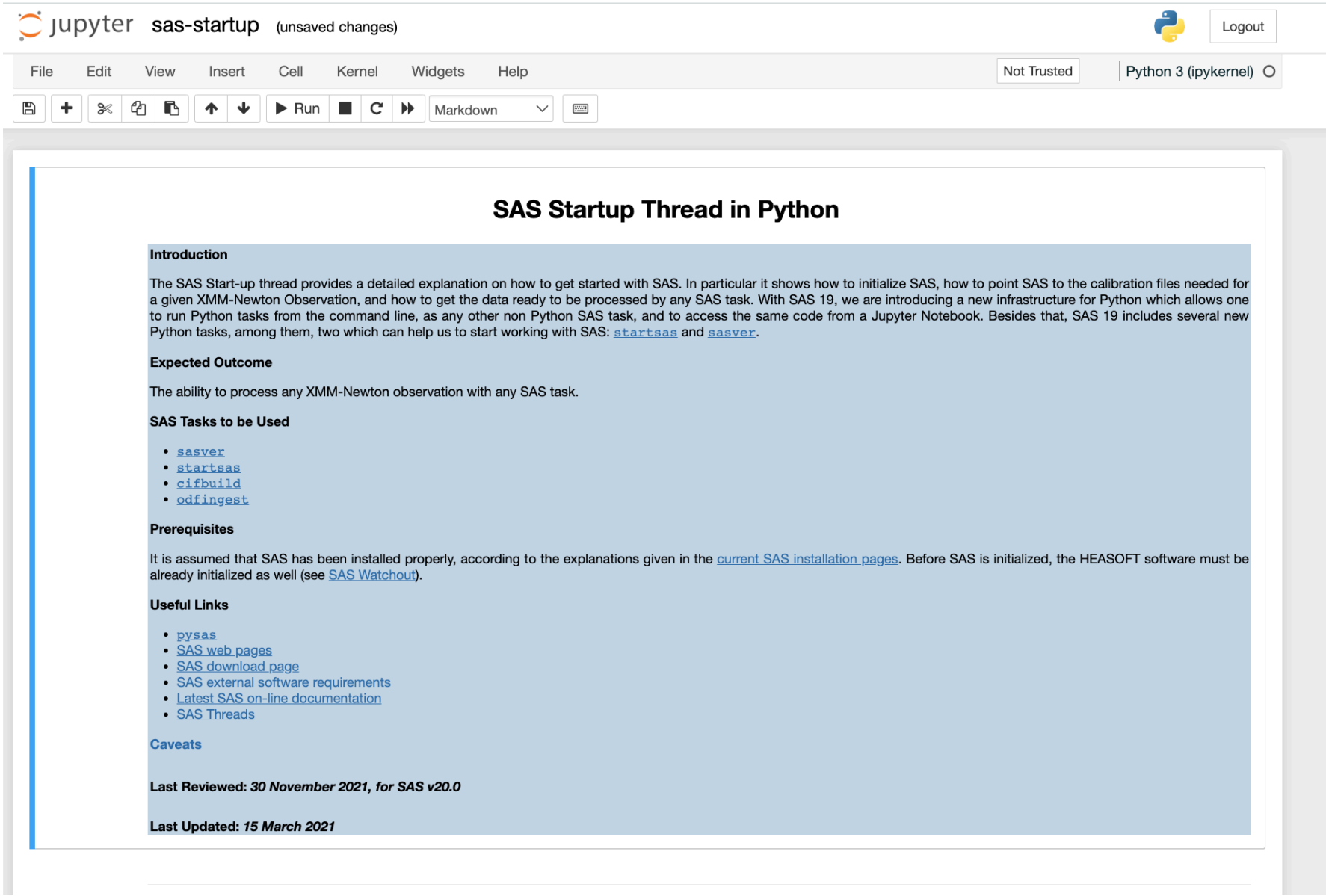
All in one go: from raw data (ODF) to science products		
- Analysis chain for point-like sources: `xmmextractor`	command line	
Step-by-Step		
- How to reduce RGS data and extract spectra of point-like sources	command line	
- `rgsproc`, coordinates and masks	command line	
Light curve generation:		
- Extraction 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	

A SAS thread (Jupyter Notebook)

A SAS thread (Jupyter Notebook)



A SAS thread (Jupyter Notebook)

jupyter sas-startup (unsaved changes) Logout

File Edit

Running `sasver`

You may try now to run your first SAS Python task: `sasver`. This task provides a sort of *about SAS* and also, a test of SAS *readiness*. If such task is able to run successfully, the whole SAS is ready to be used.

The purpose of `sasver` is to show the *identity* card of the SAS version you are running. Besides, it shows all SAS shell environment variables defined so far.

The task `sasver` can be run either from the command line or from a notebook. Most Python SAS tasks will behave this way. To run it from the command line, you simply have to invoke it as you would do any other SAS command,

```
sasver
```

which will produce in the terminal several output lines.

However, to run this task from a Jupyter Notebook, we need to employ a different method. Given that such method can be used to run any other SAS task, either Python or non Python, we are going to explain it by using the SAS task `sasver` as an example.

Invoking SAS Python tasks from notebooks

To work with any specific Python component included in SAS, we need to import the corresponding package from the Python core package for SAS. Such package is named `pysas`.

To execute any SAS task within a Notebook, we need to import from `pysas` a component known as `Wrapper`. The following cell shows how to do that,

In [ ]:
```
from pysas.wrapper import Wrapper as w
```

Any SAS task accepts arguments which can be either specific options, e.g. `--version`, which shows the task's version, or parameters with format `param=value`. When the task is invoked from the command line, these arguments follow the name of the task. However, in Notebooks we have to pass them to the task in a different way. This is done using a Python list, whose name you are free to choose. Let the name of such list be `inargs`.

To pass the option `--version` to the task to be executed, we must define `inargs` as,

A SAS thread (Not Jupyter Notebook)

This thread contains a step-by-step recipe to extract light curves of a point-like source for all the X-ray cameras, subtracting the background and correcting for exposure losses.

Expected Outcome

Corrected light curves of XMM-Newton EPIC and RGS instruments.

SAS Tasks to be Used

- `evselect`
- `epiclccorr`
- `rgslccorr`
- `barycen`

Prerequisites

- SAS Start-up Thread
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists Thread
- How to reduce RGS data and extract spectra of point-like sources Thread

Useful Links

- How to evaluate the pile-up fraction thread

Caveats

Last Reviewed: ***29 May 2013, for SAS v13.0***

Last Updated: ***29 May 2013***

Procedure

EPIC

As an example case, we will consider the extraction of a light curve from a pn event list (`PN_evt.fits`). The same recipe applies for MOS.

1. Set up your SAS environment (see **Prerequisites** for this thread at the top of the page).
2. Be aware: if you are interested in very short time periods, such as they appear in pulsars or cataclysmic variables, you have to perform a barycentric correction. This means that the arrival time of a photon is shifted as is it would have been detected at the barycentre of the solar system (the centre of mass) instead at the position of the satellite. In this way, the data are comparable. The SAS task `barycen` performs this correction. As `barycen` overwrites the `TIME` column entries, it is advisable first to copy the original event list.

```
cp PN_evt.fits PN_evt_barcen.fits
barycen table=PN_evt_barcen.fits:EVENTS
```

3. Extract an image (in sky coordinates in this example; extraction in detector - `DET[XY]` - coordinates is possible as well)

```
evselect table=PN_evt.fits imagebinning=binSize imageset=PNimage.img withimageset=yes \
   xcolumn=X ycolumn=Y ximagebinsize=80 yimagebinsize=80
```

4. Display the image

```
imgdisplay withimagefile=true imagefile=PNimage.img
```

5. Select the region, from which the light curve shall be accumulated, using the `Region/Circle` in `ds9` (see Fig.1)

A SAS thread (Not Jupyter Notebook)

This thread contains a step-by-step recipe to extract light curves of a point-like source for all the X-ray cameras, subtracting the background and correcting for exposure losses.

Expected Outcome

Corrected light curves of XMM-Newton EPIC and RGS instruments.

SAS Tasks to be Used

- `evselect`
- `epiclccorr`
- `rgslccorr`
- `barycen`

Prerequisites

- SAS Start-up Thread
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists Thread
- How to reduce RGS data and extract spectra of point-like sources Thread

Useful Links

- How to evaluate the pile-up fraction thread

Caveats

Last Reviewed: ***29 May 2013, for SAS v13.0***

Last Updated: ***29 May 2013***

Procedure

EPIC

As an example case, we will consider the extraction of a light curve from a pn event list (`PN_evt.fits`). The same recipe applies for MOS.

1. Set up your SAS environment (see **Prerequisites** for this thread at the top of the page).
2. Be aware: if you are interested in very short time periods, such as they appear in pulsars or cataclysmic variables, you have to perform a barycentric correction. This means that the arrival time of a photon is shifted as is it would have been detected at the barycentre of the solar system (the centre of mass) instead at the position of the satellite. In this way, the data are comparable. The SAS task `barycen` performs this correction. As `barycen` overwrites the `TIME` column entries, it is advisable first to copy the original event list.

```
cp PN_evt.fits PN_evt_barc
barycen table=PN_evt_barce
```

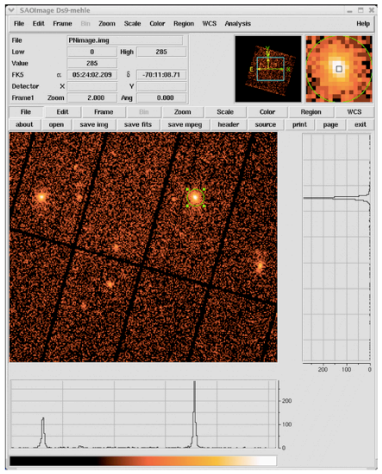
3. Extract an image (in sky coordi

```
evselect table=PN_evt.fits
  xcolumn=X ycolumn=Y xim
```

4. Display the image

```
imgdisplay withimagefile=t
```

5. Select the region, from which th



s well)

A SAS thread (Not Jupyter Notebook)

This thread contains a step-by-step recipe to extract light curves of a point-like source for all the X-ray cameras, subl

Expected Outcome

Corrected light curves of XMM-Newton EPIC and RGS instruments.

SAS Tasks to be Used

- evselect
- epiclccorr
- rgslccorr
- barycen

Prerequisites

- SAS Start-up Thread
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists Thread
- How to reduce RGS data and extract spectra of point-like sources Thread

Useful Links

- How to evaluate the pile-up fraction thread

Caveats

Last Reviewed: ***29 May 2013, for SAS v13.0***

Last Updated: ***29 May 2013***

Procedure

EPIC

As an example case, we will consider the extraction of a light curve from a pn event list (`PN_evt.fits`). The same re

1. Set up your SAS environment (see **Prerequisites** for this thread at the top of the page).
2. Be aware: if you are interested in very short time periods, such as they appear in pulsars or cataclysmic variabl arrival time of a photon is shifted as is it would have been detected at the barycentre of the solar system (the c the data are comparable. The SAS task `barycen` performs this correction. As `barycen` overwrites the `TIME` colum

```
cp PN_evt.fits PN_evt_barc
barycen table=PN_evt_barce
```

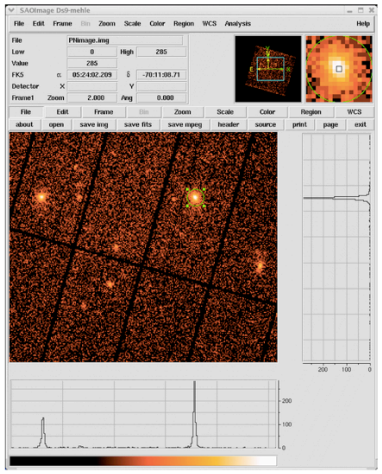
3. Extract an image (in sky coordir

```
evselect table=PN_evt.fits
  xcolumn=X ycolumn=Y xim
```

4. Display the image

```
imgdisplay withimagefile=t
```

5. Select the region, from which th



6. Double-click with the cursor on the defined region. A window pops up, showing the properties of the region (Fig.2). Write down the coordinates of the Centre (`25910.5, 25870.5`) and the `Radius(400)`.

Fig.2: Selection region properties window, popped-up by double-clicking on the region in the main ds9 window

Units of sky coordinates (X,Y) are 0.05 arcsec, hence the radius in our example is 20 arcsec.

7. Now you can extract a source+background light curve, using all the selection region and including a quality selection appropriate for a light curve extraction. For PN, taking good events, singles and doubles with an energy range between 200 and 10000 eV (`#XMMEA_EP && (PATTERN<=4) && (PI in [200:10000])`). For MOS, taking good events, singles, doubles, triples and quadruples with an energy range between 200 and 10000 eV (`#XMMEA_EM && (PATTERN<=12) && (PI in [200:10000])`). In the example, the bin size is 100 seconds.

```
evselect table=PN_evt.fits energycolumn=PI expression='#XMMEA_EP&&(PATTERN<=4)&& \
  ((X,Y) IN circle(25910.5,25870.5,400))&&(PI in [200:10000])' \
  withrateset=yes rateset="PN_source_lightcurve_raw.lc" timebinsize=100 \
  maketimecolumn=yes makeratecolumn=yes
```

The parameter `makeratecolumn=yes` produces a light curve in count rates (with errors). Otherwise the light curve is produced in counts (with errors).

8. Repeat steps #4 to #6 above to determine the region, from which the background light curve is to be extracted. We will assume in the following that the extraction region corresponds to an annulus, centered in `(25910.5,25870.5)` and with inner and outer radii `1000` and `2000` pixels, respectively.
9. Extract a background light curve, using all the selection expressions defined so far, and the same bin size (100 seconds) and energy range as for the source+background light curve

```
evselect table=PN_evt.fits energycolumn=PI expression='#XMMEA_EP&&(PATTERN<=4)&& \
  ((X,Y) IN annulus(25910.5,25870.5,1000,2000))&&(PI in [200:10000])' withrateset=yes \
  rateset="PN_light_curve_background_raw.lc" timebinsize=100 \
  maketimecolumn=yes makeratecolumn=yes
```

The light curves are OGIP-complaint, and therefore can be processed by standard XRONOS-like LHEASOFT packages.

10. However, light curves obtained in such a way should be corrected for various effects affecting the detection efficiency, such as vignetting, bad pixels, PSF variation and quantum efficiency, as well as for variations affecting the stability of the detection within the exposure, like dead time and GTIs. Since all these effects can affect in a different manner source and background light curves, the background subtraction has to be done accordingly. A SAS task, `epiclccorr`, performs all of these corrections at once. It requires as input both light curves (which are used to establish the binning of the final corrected background subtracted light curve) and the event file. A simple command line call:

```
epiclccorr srctslist=PN_source_lightcurve_raw.lc eventlist=PN_evt.fits outset=PN_lccorr.lc \
  bkgtslist=PN_light_curve_background_raw.lc withbkgset=yes applyabsolutecorrections=yes
```

11. Plot the resulting light curves, e.g.

```
dsplot table=PN_lccorr.lc withx=yes x=TIME withy=yes y=RATE
```

This command will launch the following `xmgrace` window

A SAS thread (Not Jupyter Notebook)

This thread contains a step-by-step recipe to extract light curves of a point-like source for all the X-ray cameras, subl

Expected Outcome

Corrected light curves of XMM-Newton EPIC and RGS instruments.

SAS Tasks to be Used

- evselect
- epiclccorr
- rgslccorr
- barycen

Prerequisites

- SAS Start-up Thread
- How to reprocess ODFs to generate calibrated and concatenated EPIC event lists Thread
- How to reduce RGS data and extract spectra of point-like sources Thread

Useful Links

- How to evaluate the pile-up fraction thread

Caveats

Last Reviewed: ***29 May 2013, for SAS v13.0***

Last Updated: ***29 May 2013***

Procedure

EPIC

As an example case, we will consider the extraction of a light curve from a pn event list (`PN_evt.fits`). The same re

1. Set up your SAS environment (see **Prerequisites** for this thread at the top of the page).
2. Be aware: if you are interested in very short time periods, such as they appear in pulsars or cataclysmic variabl arrival time of a photon is shifted as is it would have been detected at the barycentre of the solar system (the c the data are comparable. The SAS task `barycen` performs this correction. As `barycen` overwrites the `TIME` colum

```
cp PN_evt.fits PN_evt_barc
barycen table=PN_evt_barce
```

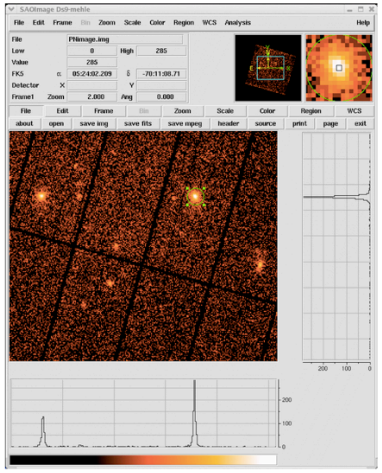
3. Extract an image (in sky coordir

```
evselect table=PN_evt.fits
  xcolumn=X ycolumn=Y xim
```

4. Display the image

```
imgdisplay withimagefile=t
```

5. Select the region, from which th



6. Double-click with the cursor on the defined region. A window pops up, showing the properties of the region (Fig.2). Write down the coordinates of the Centre (`25910.5, 25870.5`) and the `Radius(400)`.

Fig.2: Selection region properties window, popped-up by double-clicking on the region in the main `ds9` window

Units of sky coordinates (X,Y) are 0.05 arcsec, hence the radius in our example is 20 arcsec.

7. Now you can extract a source+background light curve, using all the selection region and including a quality selection appropriate for a light curve extraction. For PN, taking good events, singles and doubles with an energy range between 200 and 10000 eV (`#XMMEA_EP && (PATTERN<=4) && (PI in [200:10000])`). For MOS, taking good events, singles, doubles, triples and quadruples with an energy range between 200 and 10000 eV (`#XMMEA_EM && (PATTERN<=12) && (PI in [200:10000])`). In the example, the bin size is 100 seconds.

```
evselect table=PN_evt.fits energycolumn=PI expression='#XMMEA_EP&&(PATTERN<=4)&& \
  ((X,Y) IN circle(25910.5,25870.5,400))&&(PI in [200:10000])' \
  withrateset=yes rateset="PN_source_lightcurve_raw.lc" timebinsize=100 \
  maketimecolumn=yes makeratecolumn=yes
```

The parameter `makeratecolumn=yes` produces a light curve in count rates (with errors). Otherwise the light curve is produced in counts (with errors).

8. Repeat steps #4 to #6 above to determine the region, from which the background light curve is to be extracted. We will assume in the following that the extraction region corresponds to an annulus, centered in `(25910.5,25870.5)` and with inner and outer radii `1000` and `2000` pixels, respectively.
9. Extract a background light curve, using all the selection expressions defined so far, and the same bin size (100 seconds) and energy range as for the source+background light curve

```
evselect table=PN_evt.fits energycolumn=PI expression='#XMMEA_EP&&(PATTERN<=4)&& \
  ((X,Y) IN annulus(25910.5,25870.5,1000,2000))&&(PI in [200:10000])' withrateset=yes \
  rateset="PN_light_curve_background_raw.lc" timebinsize=100 \
  maketime
```

The light curv

10. However, ligh efficiency, as source and be both light cur

```
epiclccorr
  bkgtsli
```

11. Plot the resul

```
dsplot tab
```

This comman

11. Plot the resulting light curves, e.g.

```
dsplot table=PN_lccorr.lc withx=yes x=TIME withy=yes y=RATE
```

This command will launch the following `xmgrace` window

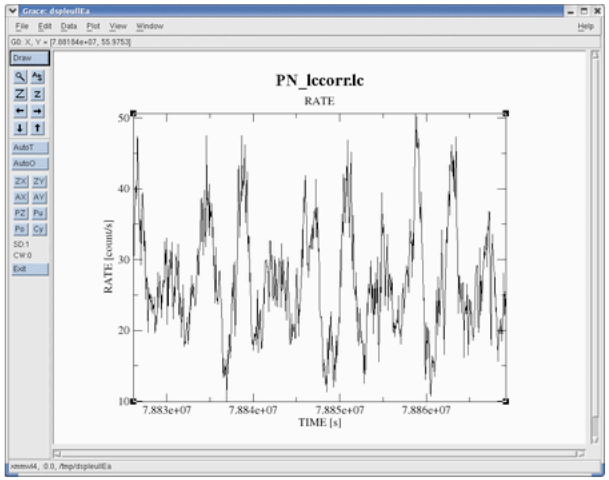


Fig.3: `xmgrace` window, containing the background-subtracted exposure-corrected light curve

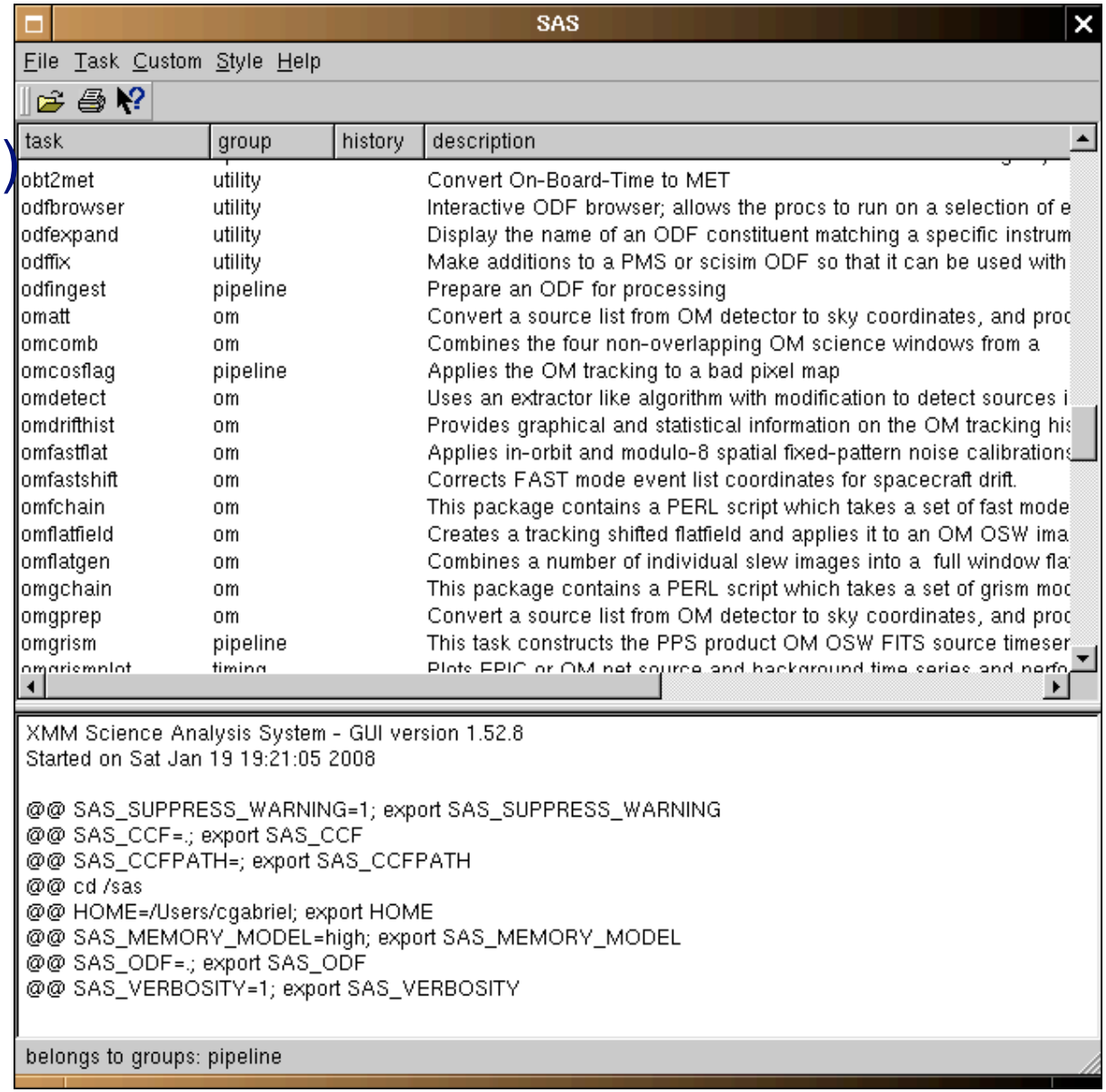
The main GUI

The main GUI

- Access to all tasks (GUI call) and descriptions
- Setting general defaults
- Access to help pages
- Writing log into window and file (sas_log)

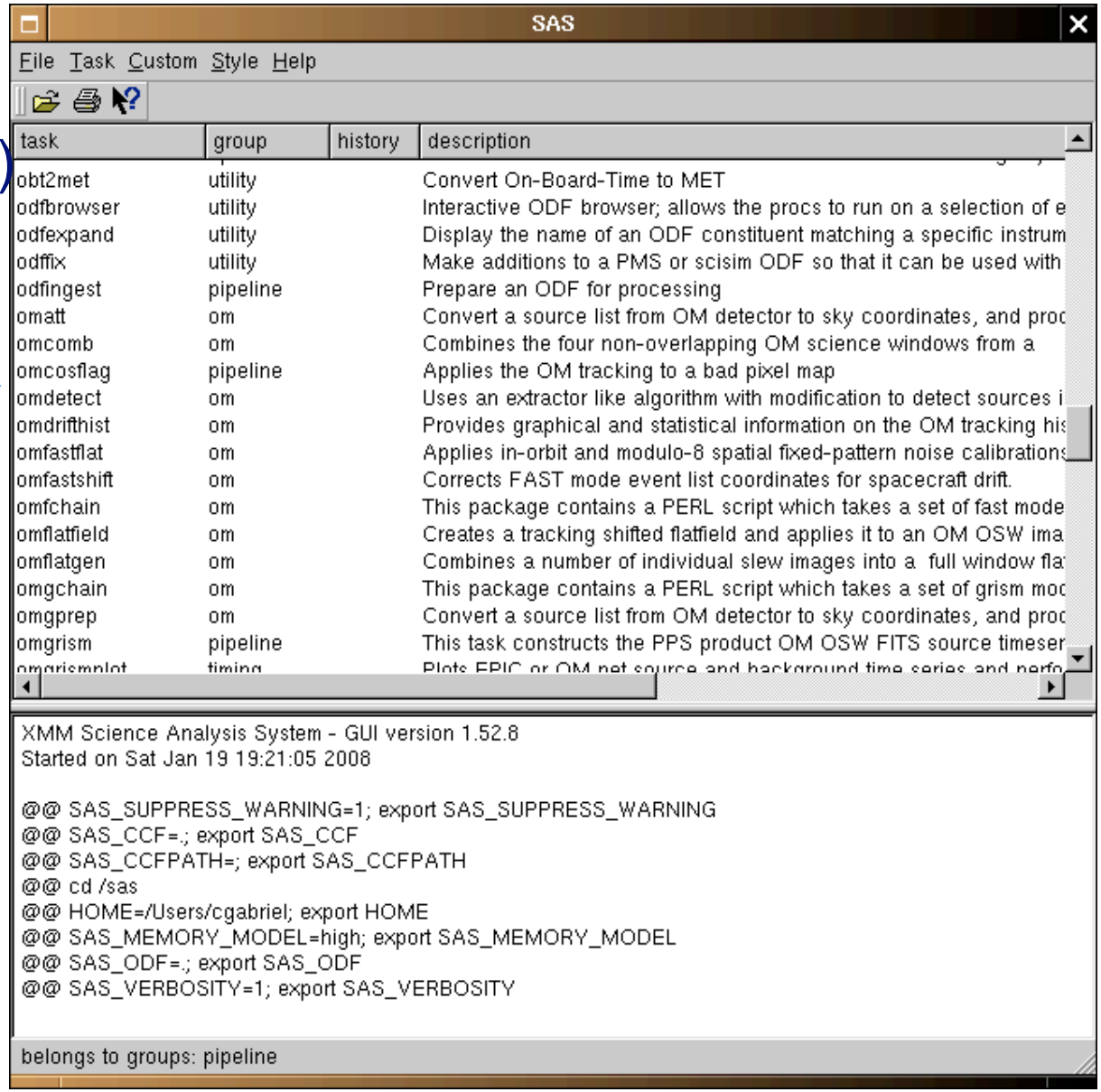
The main GUI

- Access to all tasks (GUI call) and descriptions
- Setting general defaults
- Access to help pages
- Writing log into window and file (sas_log)



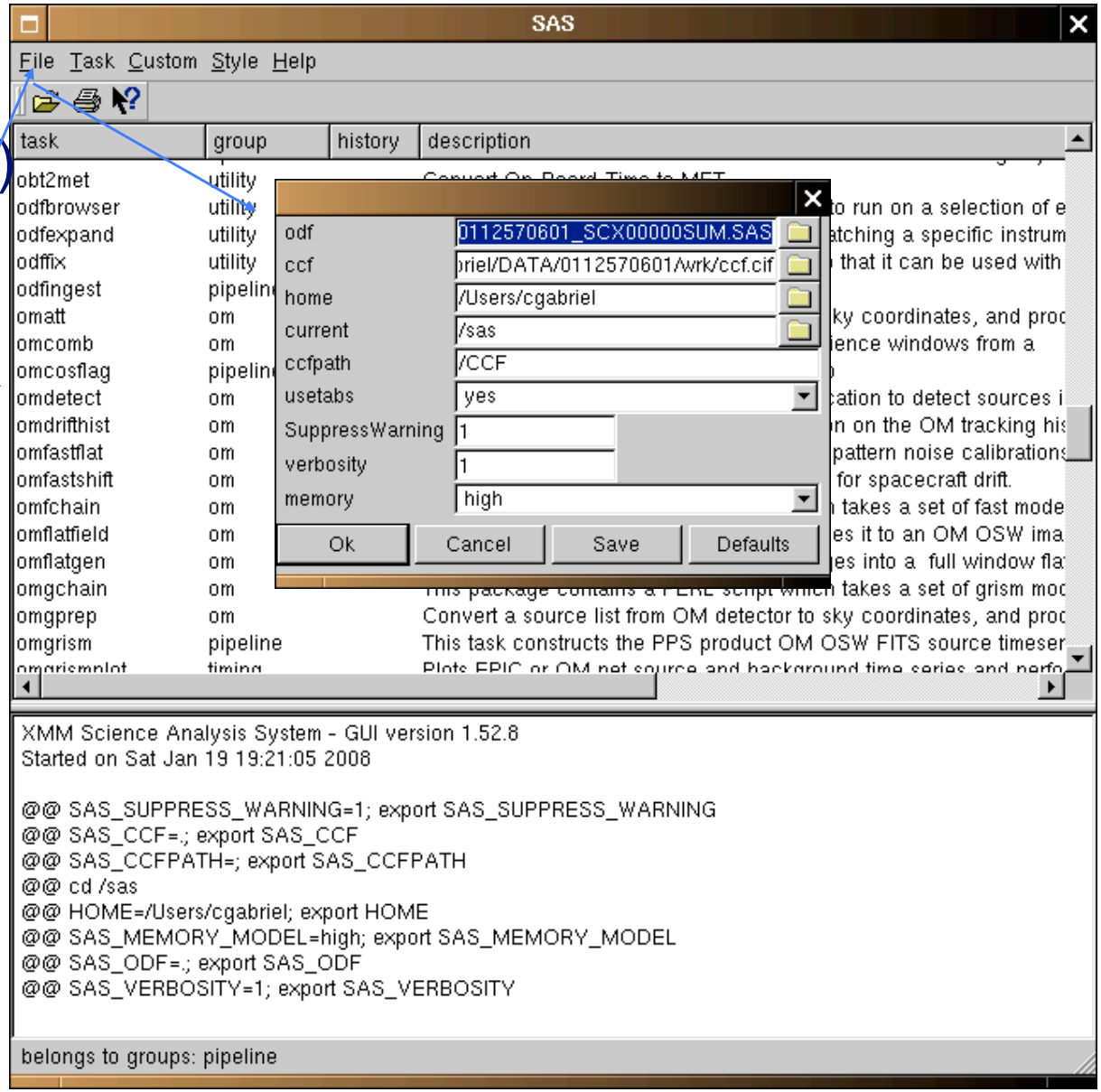
The main GUI

- Access to all tasks (GUI call) and descriptions
- Setting general defaults
- Access to help pages
- Writing log into window and file (sas_log)



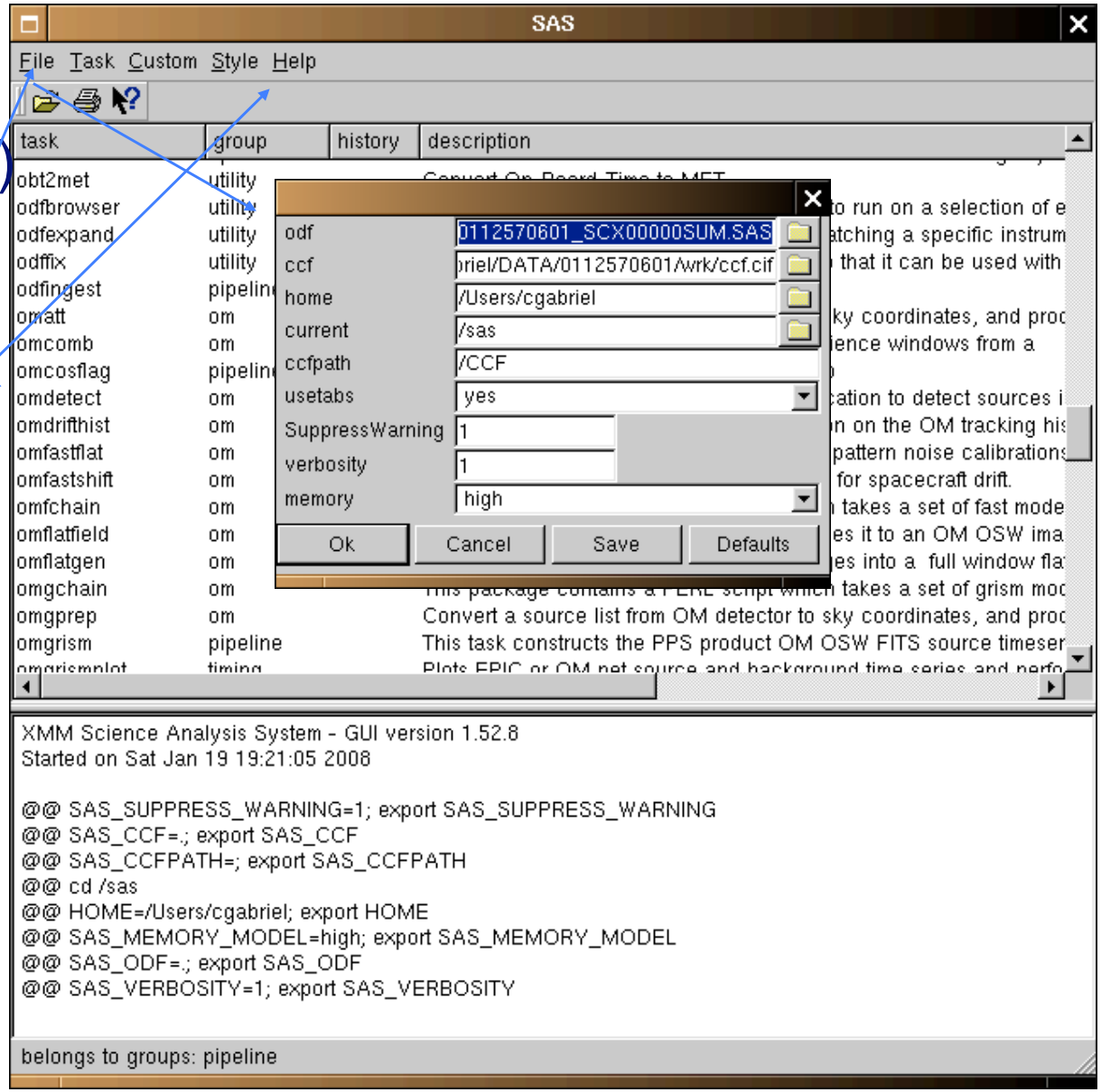
The main GUI

- Access to all tasks (GUI call) and descriptions
- Setting general defaults
- Access to help pages
- Writing log into window and file (sas_log)



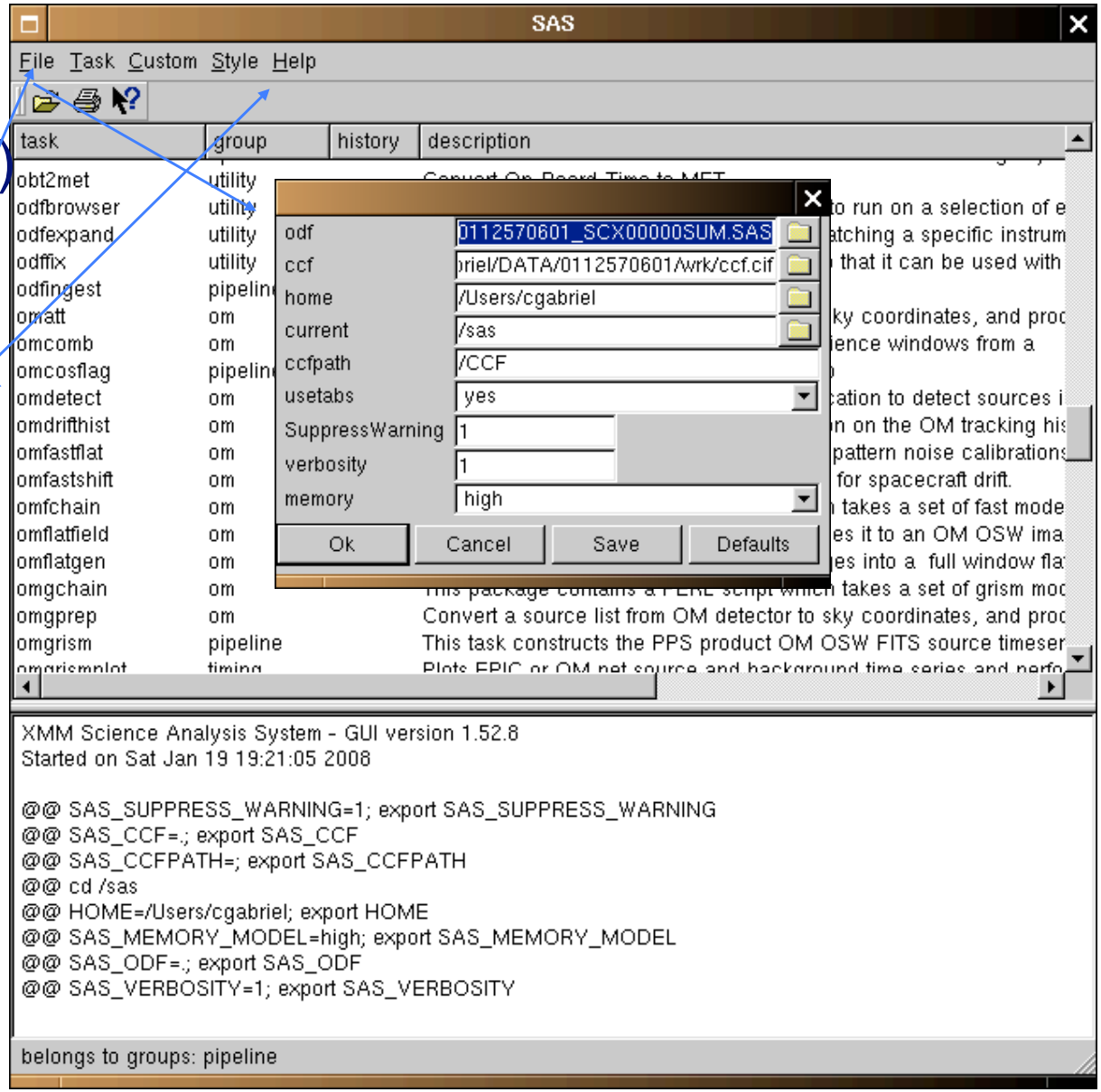
The main GUI

- Access to all tasks (GUI call) and descriptions
- Setting general defaults
- Access to help pages
- Writing log into window and file (sas_log)



The main GUI

- Access to all tasks (GUI call) and descriptions
- Setting general defaults
- Access to help pages
- Writing log into window and file (sas_log)



GUI or command line?

GUI or command line?

GUIs are very useful for beginners
Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

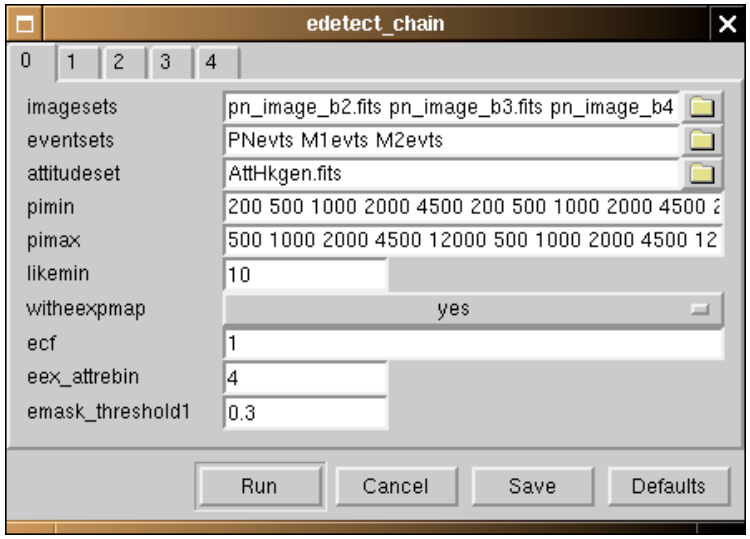
edetect_chain –d

GUI or command line?

GUIs are very useful for beginners

Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

edetect_chain –d

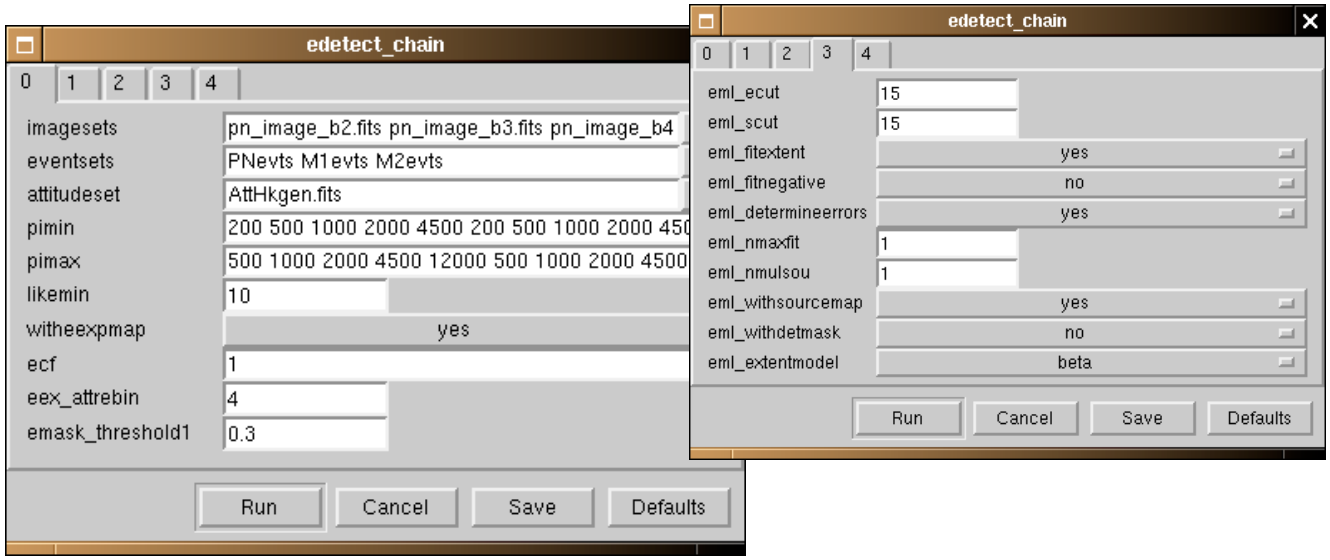


GUI or command line?

GUIs are very useful for beginners

Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

edetect_chain –d

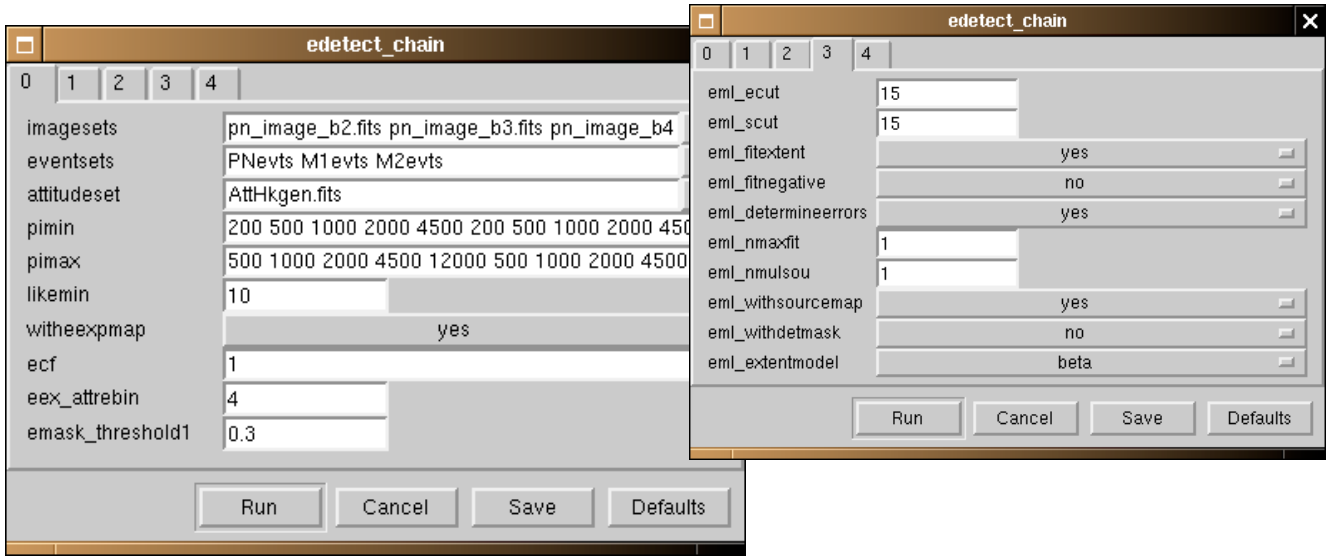


GUI or command line?

GUIs are very useful for beginners

Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

edetect_chain –d



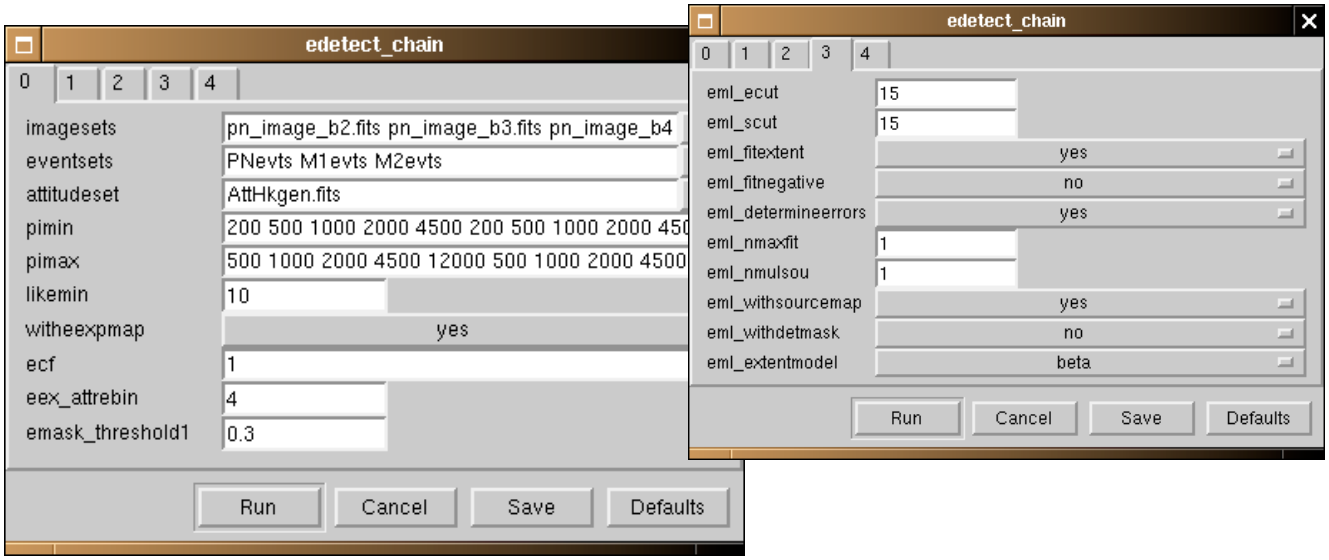
source list

GUI or command line?

GUIs are very useful for beginners

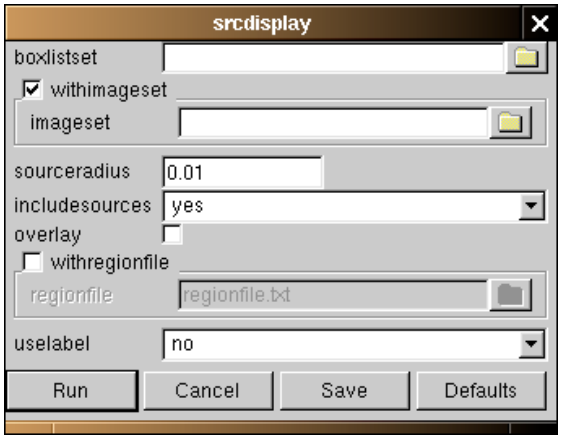
Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

edetect_chain –d



source list

srcdisplay -d

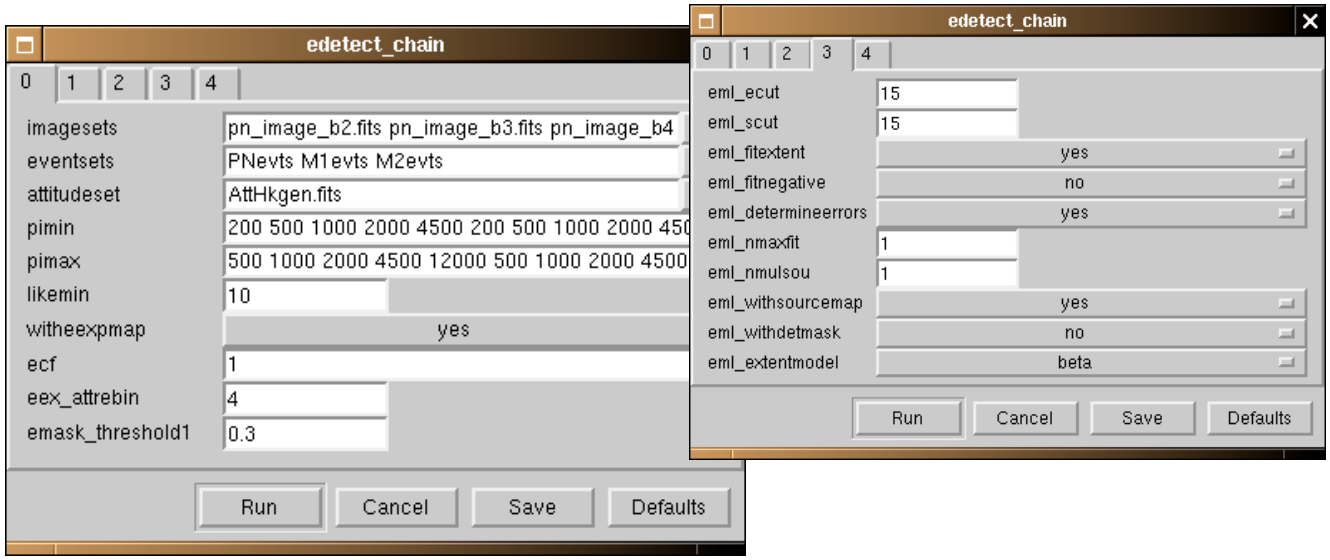


GUI or command line?

GUIs are very useful for beginners

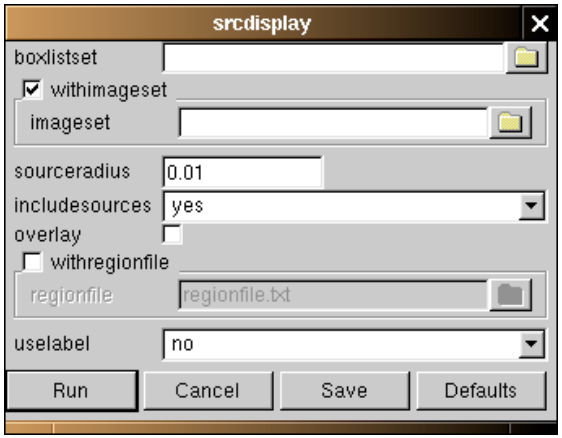
Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

edetect_chain –d



source list

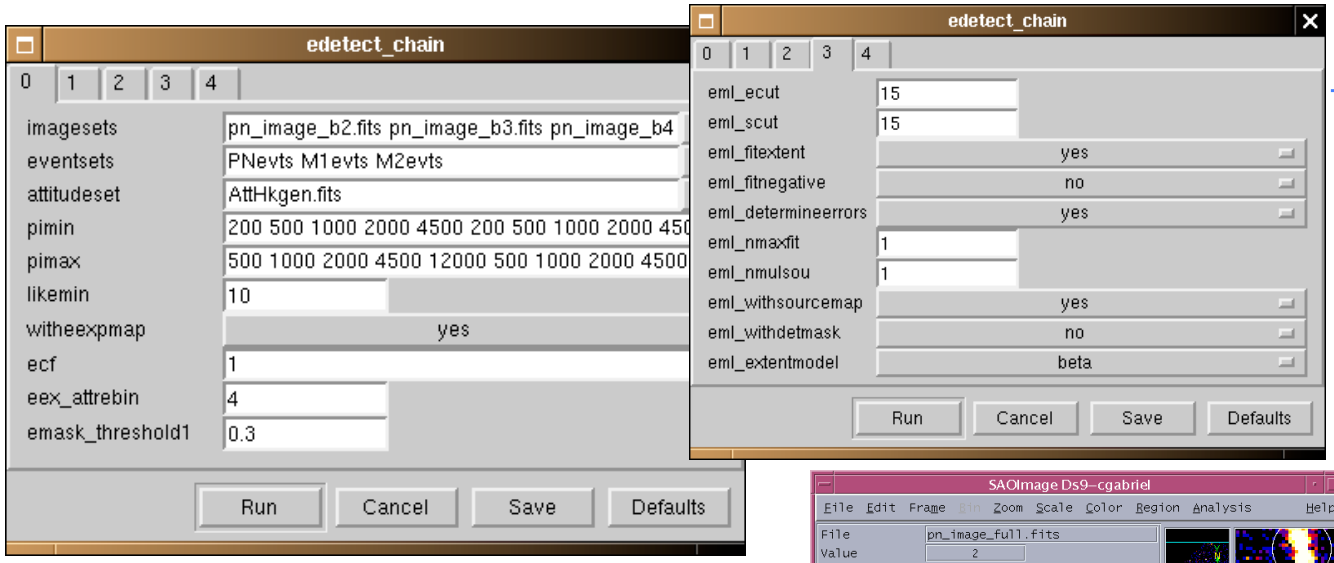
srcdisplay -d



GUI or command line?

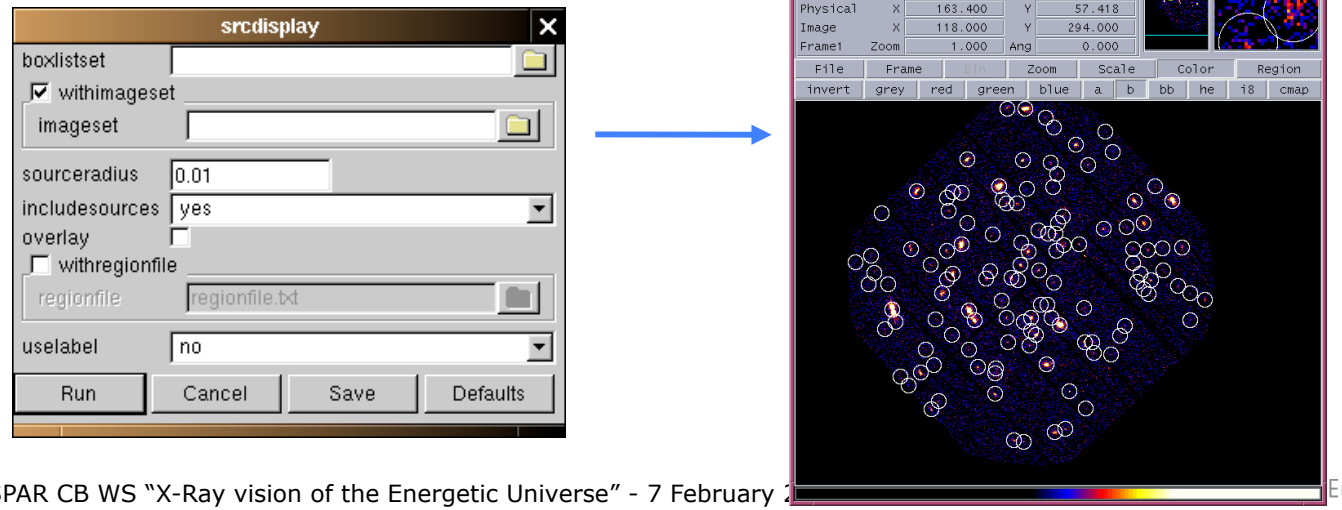
GUIs are very useful for beginners
Every SAS task has its own GUI - they can be called by other (main) GUI or directly from the command line by #> <task> -d:

edetect_chain –d



source list

srcdisplay -d



GUI or command line? BOTH

GUI or command line? BOTH

Line commands are, if you know them, faster in execution,
and you learn how to concatenate them, so that ...

GUI or command line? BOTH

Line commands are, if you know them, faster in execution, and you learn how to concatenate them, so that ...

eventually, you can produce your own scripts

det_script - /Users/cgabriel/DATA/0112570601/wrk/srcdet/

File Edit Search Preferences Shell Macro Windows Help

```
evselect table=MOS1evt:EVENTS expression='#XMMEA_EM&&(PI>10000)&&(PATTERN==0)' rateset="m1_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes
evselect table=MOS2evt:EVENTS expression='#XMMEA_EM&&(PI>10000)&&(PATTERN==0)' rateset="m2_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes
evselect table=PNevt:EVENTS expression='#XMMEA_EP&&(PI>10000)&&(PATTERN==0)' rateset="pn_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes

dsplot table=m1_back_lightc.fits x=TIME y=RATE &
dsplot table=m2_back_lightc.fits x=TIME y=RATE &
dsplot table=pn_back_lightc.fits x=TIME y=RATE &

tabgtigen table=m1_back_lightc.fits expression="RATE < 0.35" gtiset=m1_back_gti.fits
tabgtigen table=m2_back_lightc.fits expression="RATE < 0.35" gtiset=m2_back_gti.fits
tabgtigen table=pn_back_lightc.fits expression="RATE < 1.00" gtiset=pn_back_gti.fits

evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_full.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [200:12000])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b1.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [200:500])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b2.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [500:1000])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b3.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [1000:2000])&&(PATTERN in [0:12])&&(FLAG==0) &&
```

GUI or command line? BOTH

Line commands are, if you know them, faster in execution, and you learn how to concatenate them, so that ...

eventually, you can produce your own scripts

det_script - /Users/cgabriel/DATA/0112570601/wrk/srcdet/

File Edit Search Preferences Shell Macro Windows Help

```
evselect table=MOS1evt:EVENTS expression='#XMMEA_EM&&(PI>10000)&&(PATTERN==0)' rateset="m1_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes
evselect table=MOS2evt:EVENTS expression='#XMMEA_EM&&(PI>10000)&&(PATTERN==0)' rateset="m2_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes
evselect table=PNevt:EVENTS expression='#XMMEA_EP&&(PI>10000)&&(PATTERN==0)' rateset="pn_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes

dsplot table=m1_back_lightc.fits x=TIME y=RATE &
dsplot table=m2_back_lightc.fits x=TIME y=RATE &
dsplot table=pn_back_lightc.fits x=TIME y=RATE &

tabgtigen table=m1_back_lightc.fits expression="RATE < 0.35" gtiset=m1_back_gti.fits
tabgtigen table=m2_back_lightc.fits expression="RATE < 0.35" gtiset=m2_back_gti.fits
tabgtigen table=pn_back_lightc.fits expression="RATE < 1.00" gtiset=pn_back_gti.fits

evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_full.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [200:12000])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b1.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [200:500])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b2.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [500:1000])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b3.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [1000:2000])&&(PATTERN in [0:12])&&(FLAG==0) &&
```

My answer: GUI & command line

GUI or command line? BOTH

Line commands are, if you know them, faster in execution, and you learn how to concatenate them, so that ...

eventually, you can produce your own scripts

det_script - /Users/cgabriel/DATA/0112570601/wrk/srcdet/

File Edit Search Preferences Shell Macro Windows Help

```
evselect table=MOS1evt:EVENTS expression='#XMMEA_EM&&(PI>10000)&&(PATTERN==0)' rateset="m1_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes
evselect table=MOS2evt:EVENTS expression='#XMMEA_EM&&(PI>10000)&&(PATTERN==0)' rateset="m2_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes
evselect table=PNevt:EVENTS expression='#XMMEA_EP&&(PI>10000)&&(PATTERN==0)' rateset="pn_back_lightc.fits" timebinsize=10
withrateset=yes maketimecolumn=yes makeratecolumn=yes

dsplot table=m1_back_lightc.fits x=TIME y=RATE &
dsplot table=m2_back_lightc.fits x=TIME y=RATE &
dsplot table=pn_back_lightc.fits x=TIME y=RATE &

tabgtigen table=m1_back_lightc.fits expression="RATE < 0.35" gtiset=m1_back_gti.fits
tabgtigen table=m2_back_lightc.fits expression="RATE < 0.35" gtiset=m2_back_gti.fits
tabgtigen table=pn_back_lightc.fits expression="RATE < 1.00" gtiset=pn_back_gti.fits

evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_full.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [200:12000])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b1.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [200:500])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b2.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [500:1000])&&(PATTERN in [0:12])&&(FLAG==0) &&
gti(m1_back_gti.fits,TIME)'
evselect table=MOS1evt:EVENTS imagebinning='binSize' imageset='m1_image_b3.fits' withimageset=yes xcolumn='X' ycolumn='Y'
ximagebinsize=40 yimagebinsize=40 expression='#XMMEA_EM&&(PI in [1000:2000])&&(PATTERN in [0:12])&&(FLAG==0) &&
```

My answer: GUI & command line >> scripts

PPS or “proc” products? BOTH

All data → already reduced by PPS (SAS subset with default parameters)

PPS or “proc” products? BOTH

All data → already reduced by PPS (SAS subset with default parameters)

Why reprocessing then (epproc, emproc, rgsproc, om?chain) ?

- old SAS/PPS used for that data
- newer (probably more accurate) calibration than the one used by PPS
- special needs for using non-default parameters

PPS or “proc” products? BOTH

All data → already reduced by PPS (SAS subset with default parameters)

Why reprocessing then (epproc, emproc, rgsproc, om?chain) ?

- old SAS/PPS used for that data
- newer (probably more accurate) calibration than the one used by PPS
- special needs for using non-default parameters

During 2019 general reprocessing (third time in 19 years)
>> XMM Science Archive repopulated with s/w and calibration as 2019 knowledge
+ source specific products: spectra and light curves
+ 4XMM-DR9 - the largest X-ray sources catalogue

PPS or “proc” products? BOTH

All data → already reduced by PPS (SAS subset with default parameters)

Why reprocessing then (epproc, emproc, rgsproc, om?chain) ?

- old SAS/PPS used for that data
- newer (probably more accurate) calibration than the one used by PPS
- special needs for using non-default parameters

During 2019 general reprocessing (third time in 19 years)
>> XMM Science Archive repopulated with s/w and calibration as 2019 knowledge
+ source specific products: spectra and light curves
+ 4XMM-DR9 - the largest X-ray sources catalogue

Bulk reprocessing will take place in …

PPS or “proc” products? BOTH

All data → already reduced by PPS (SAS subset with default parameters)

Why reprocessing then (epproc, emproc, rgsproc, om?chain) ?

- old SAS/PPS used for that data
- newer (probably more accurate) calibration than the one used by PPS
- special needs for using non-default parameters

During 2019 general reprocessing (third time in 19 years)
>> XMM Science Archive repopulated with s/w and calibration as 2019 knowledge
+ source specific products: spectra and light curves
+ 4XMM-DR9 - the largest X-ray sources catalogue

Bulk reprocessing will take place in …

Even if you reduce yourself the data it is important to get a look into the PPS data beforehand, in ALL cases they will give a good impression about the contents

Where is the XMM-Newton data? In XSA

Where is the XMM-Newton data? In XSA

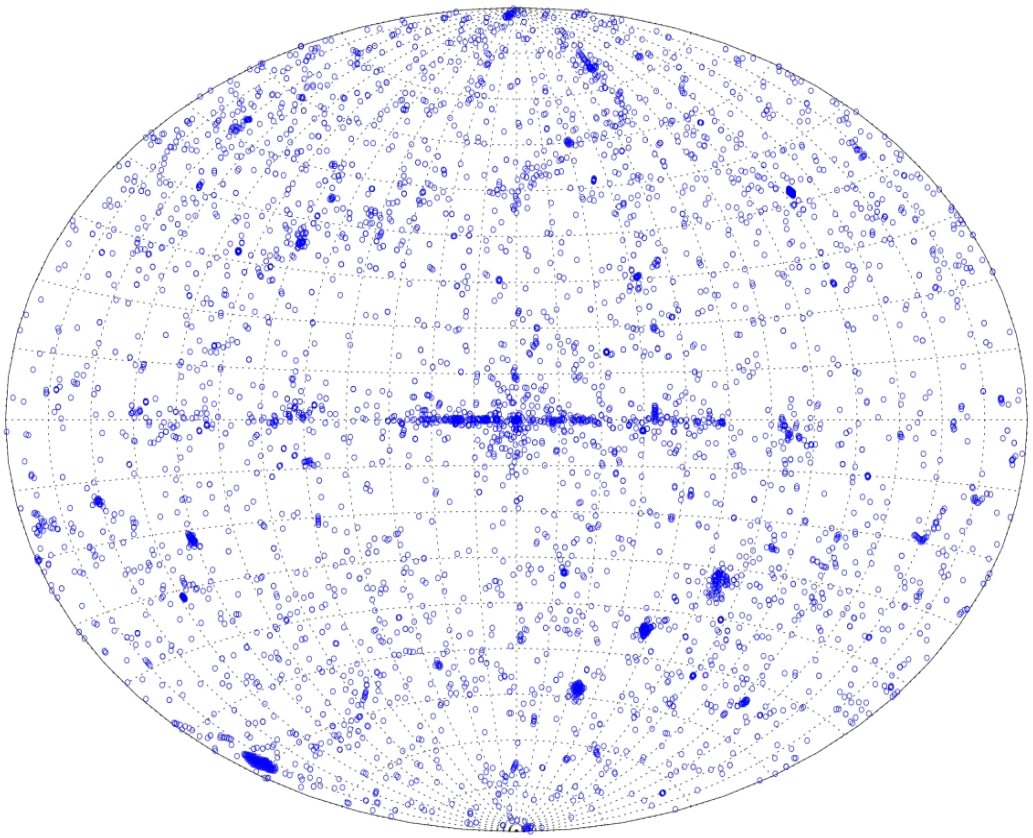
Contents of the XSA:

- ODF/PPS of ~ 17000 pointed observations
- SDF of ~ 5,000 slew observations
- SDF of ~ 200,000 Slew Survey sub-exposures
- 939.270 detections (4XMM-DR12 catalogue) / 630.347 unique sources
- 8,863,922 OM sources (XMMOM SUSS 5 catalogue)
- 72352 Slew Survey sources (XMMSL2)

Where is the XMM-Newton data? In XSA

Contents of the XSA:

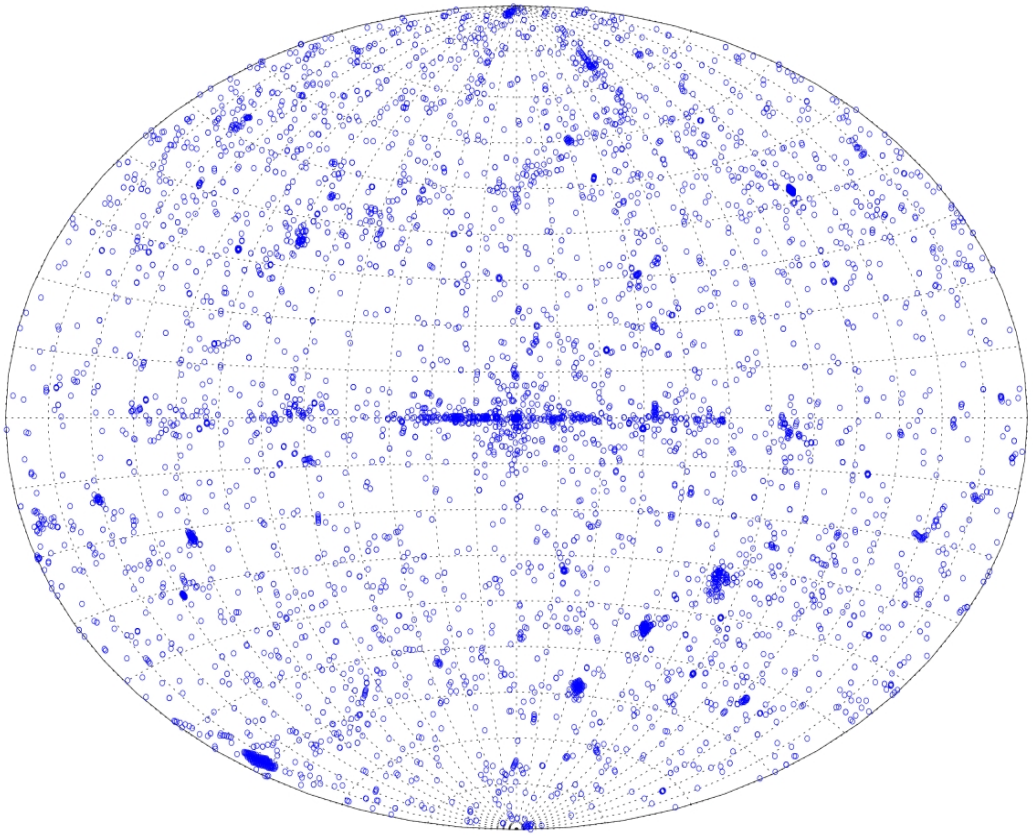
- ODF/PPS of ~ 17000 pointed observations
- SDF of ~ 5,000 slew observations
- SDF of ~ 200,000 Slew Survey sub-exposures
- 939.270 detections (4XMM-DR12 catalogue) / 630.347 unique sources
- 8,863,922 OM sources (XMMOM SUSS 5 catalogue)
- 72352 Slew Survey sources (XMMSL2)

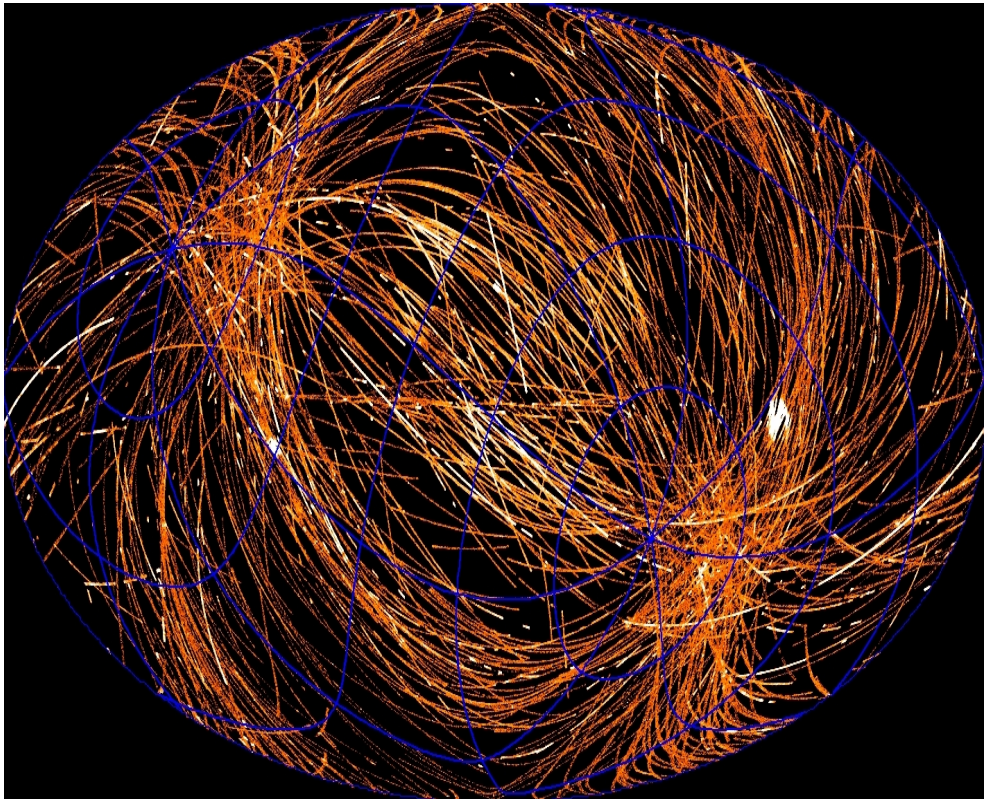


Where is the XMM-Newton data? In XSA

Contents of the XSA:

- ODF/PPS of ~ 17000 pointed observations
- SDF of ~ 5,000 slew observations
- SDF of ~ 200,000 Slew Survey sub-exposures
- 939.270 detections (4XMM-DR12 catalogue) / 630.347 unique sources
- 8,863,922 OM sources (XMMOM SUSS 5 catalogue)
- 72352 Slew Survey sources (XMMSL2)

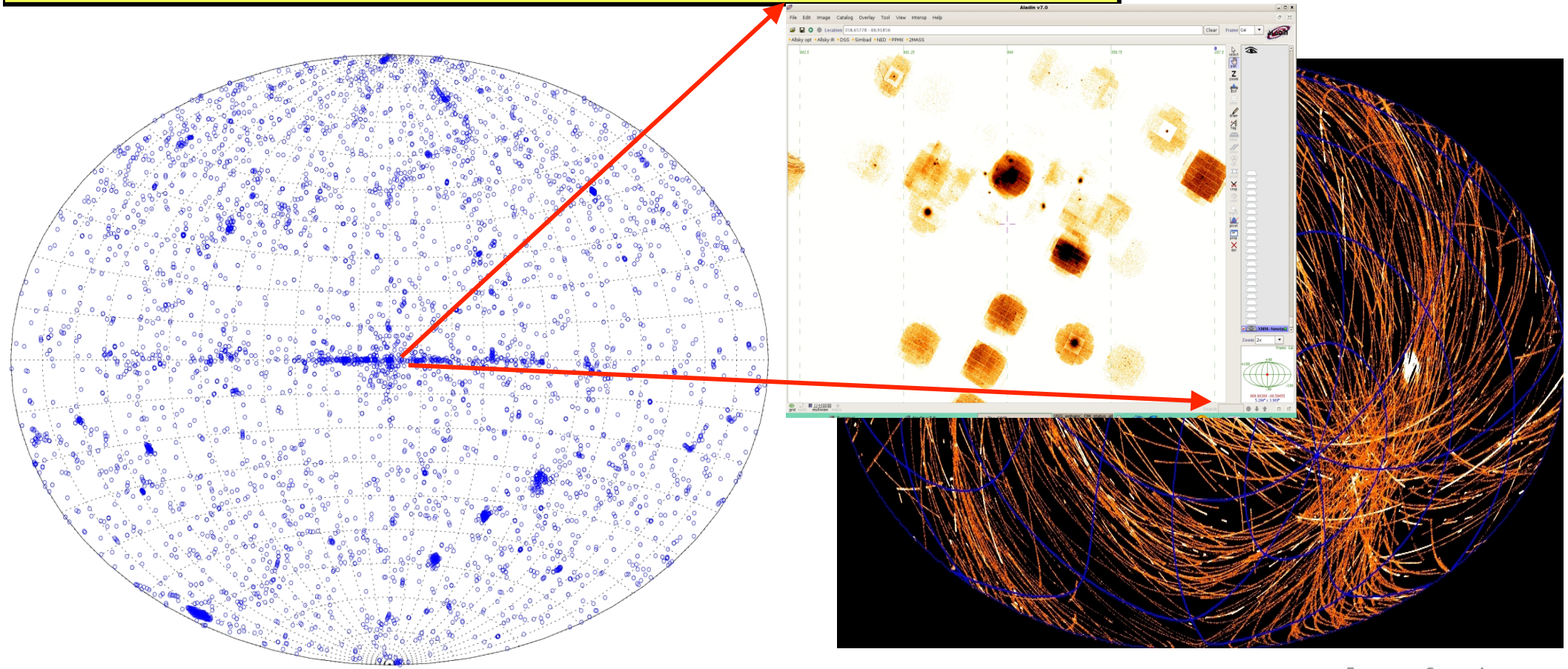




Where is the XMM-Newton data? In XSA

Contents of the XSA:

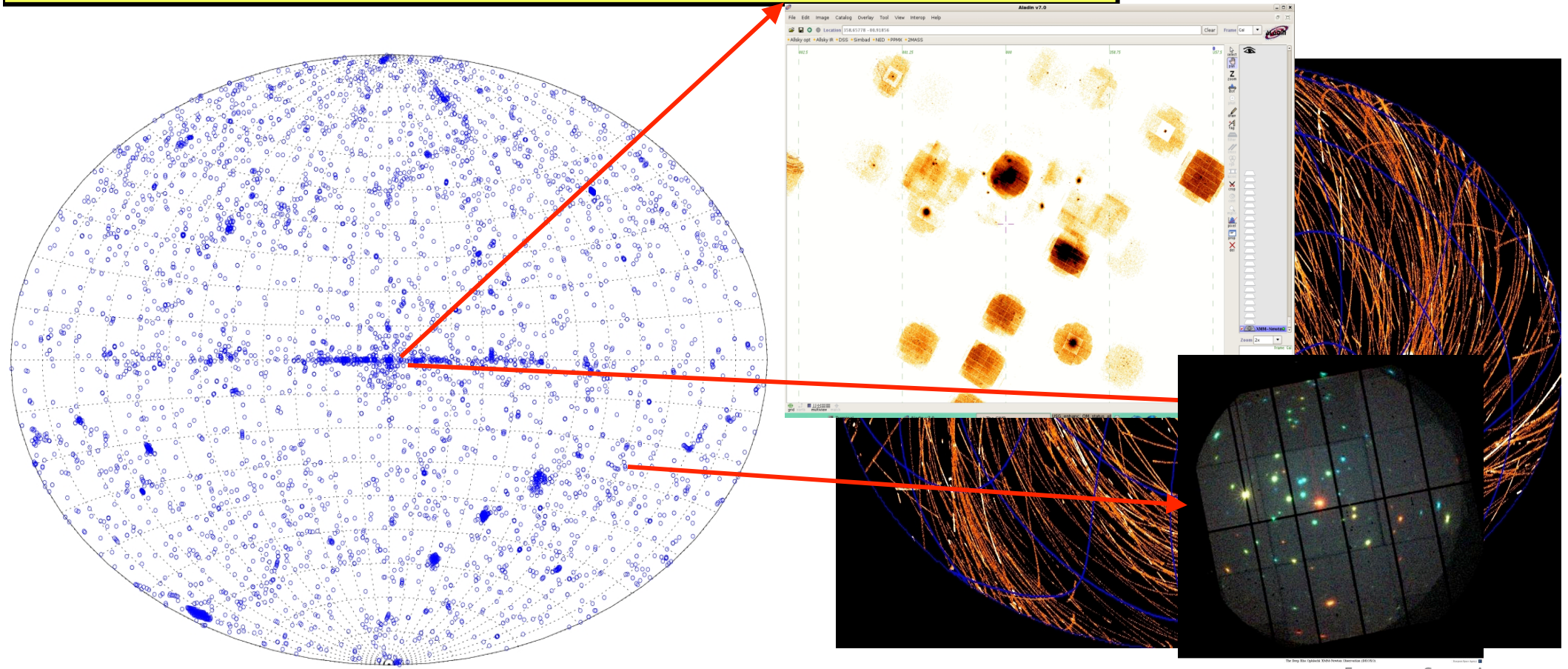
- ODF/PPS of ~ 17000 pointed observations
- SDF of ~ 5,000 slew observations
- SDF of ~ 200,000 Slew Survey sub-exposures
- 939.270 detections (4XMM-DR12 catalogue) / 630.347 unique sources
- 8,863,922 OM sources (XMMOM SUSS 5 catalogue)
- 72352 Slew Survey sources (XMMSL2)



Where is the XMM-Newton data? In XSA

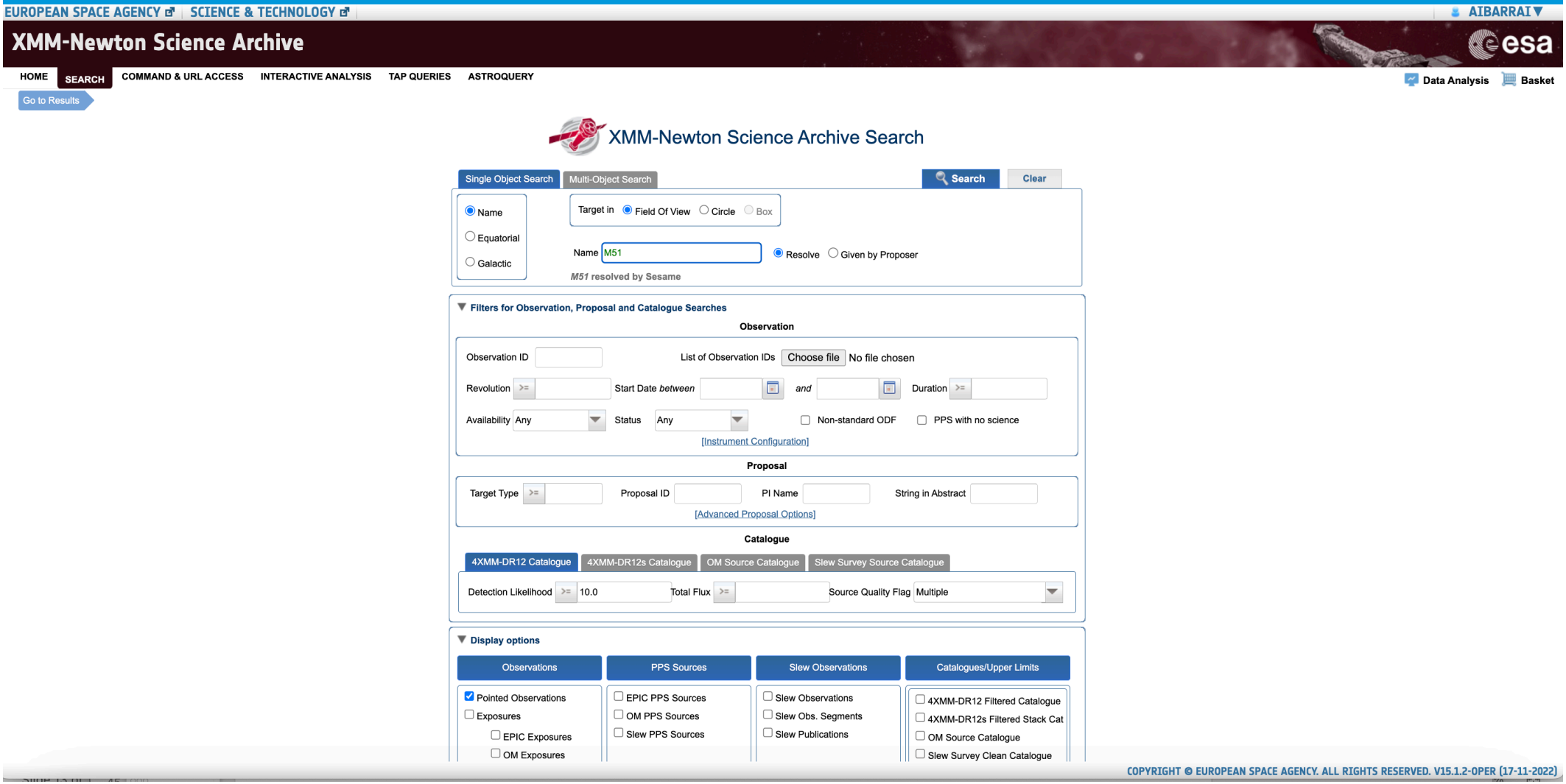
Contents of the XSA:

- ODF/PPS of ~ 17000 pointed observations
- SDF of ~ 5,000 slew observations
- SDF of ~ 200,000 Slew Survey sub-exposures
- 939.270 detections (4XMM-DR12 catalogue) / 630.347 unique sources
- 8,863,922 OM sources (XMMOM SUSS 5 catalogue)
- 72352 Slew Survey sources (XMMSL2)

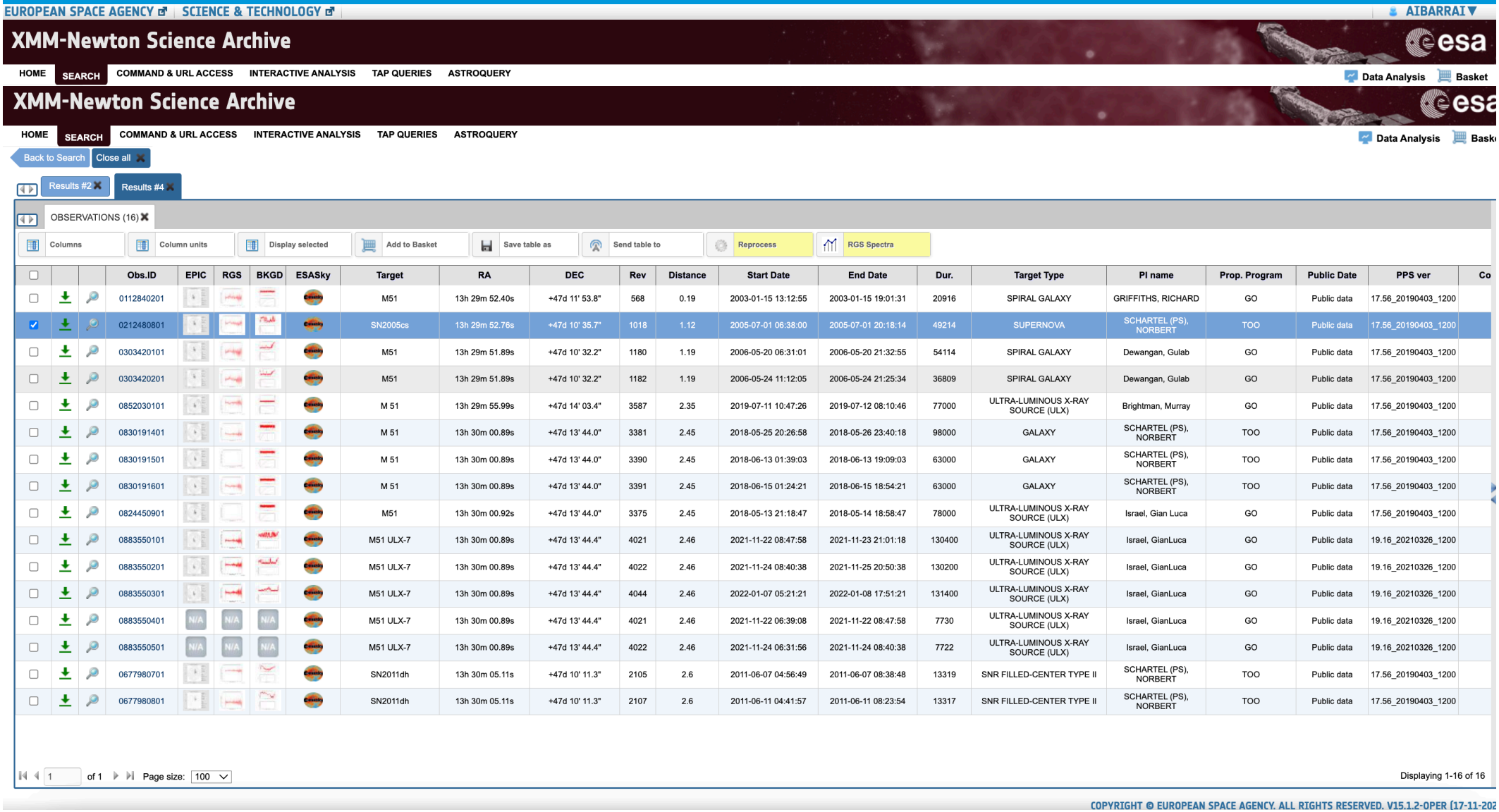


How to get XMM-Newton data? The XSA

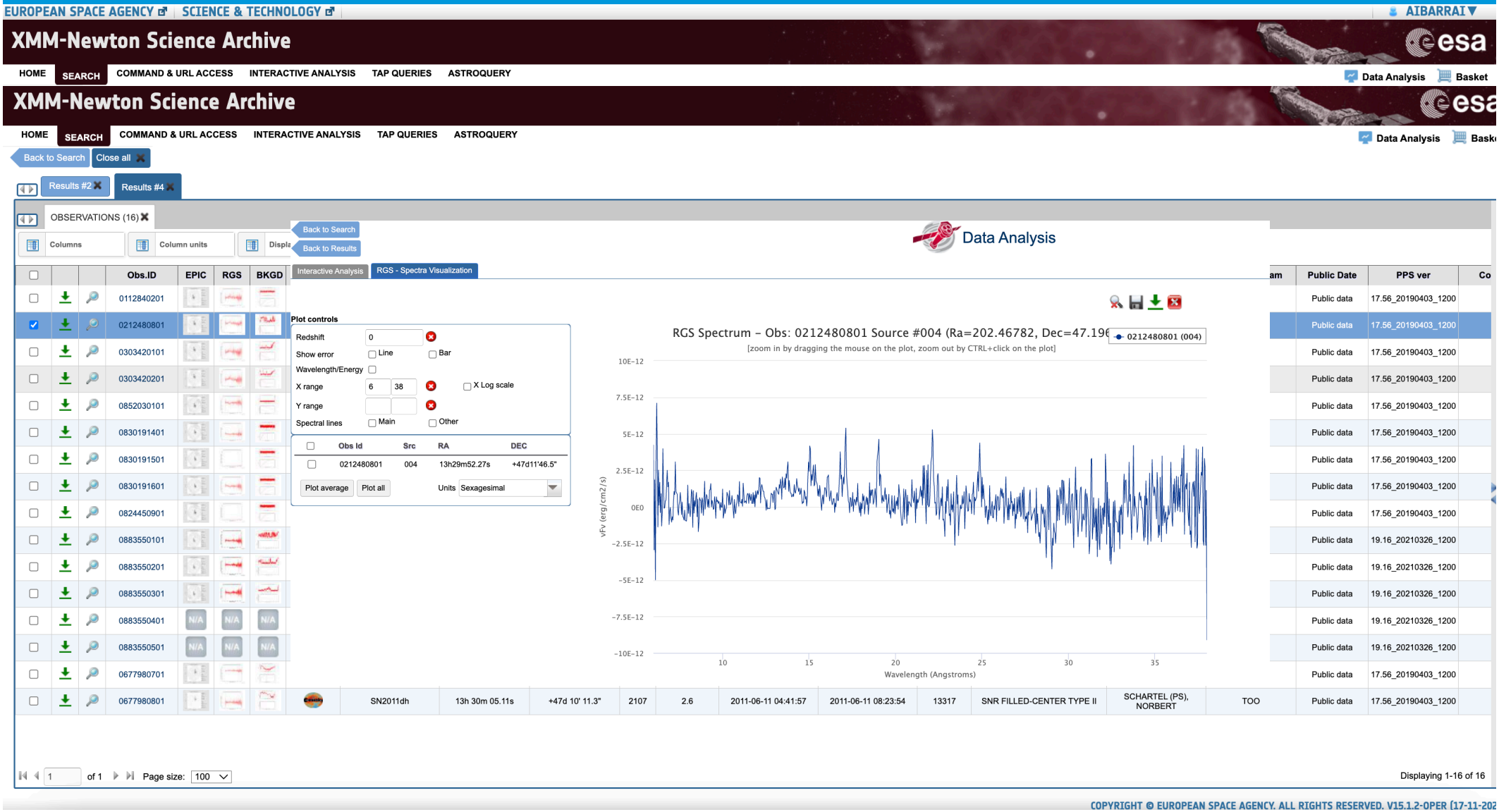
How to get XMM-Newton data? The XSA



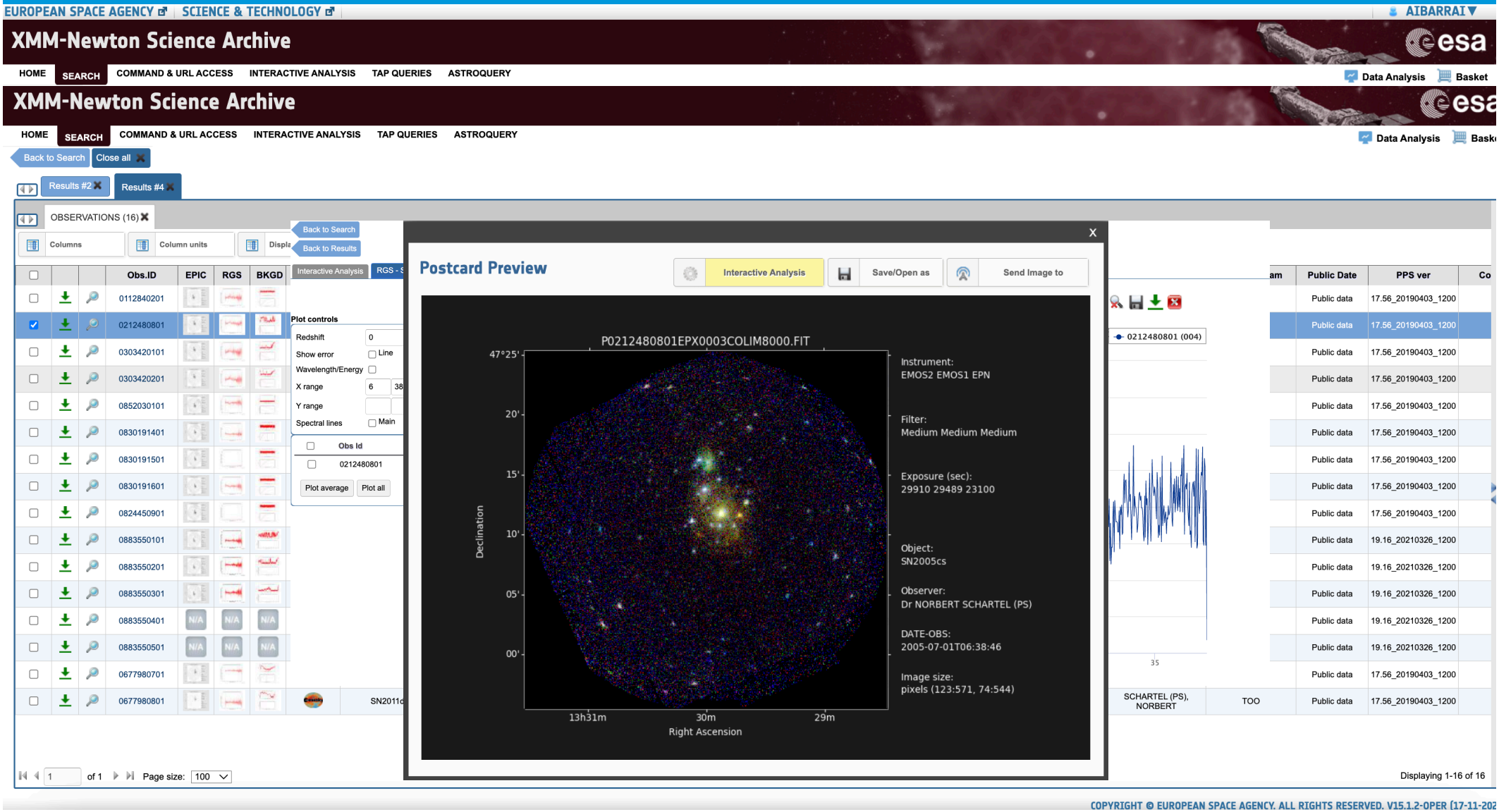
How to get XMM-Newton data? The XSA



How to get XMM-Newton data? The XSA



How to get XMM-Newton data? The XSA



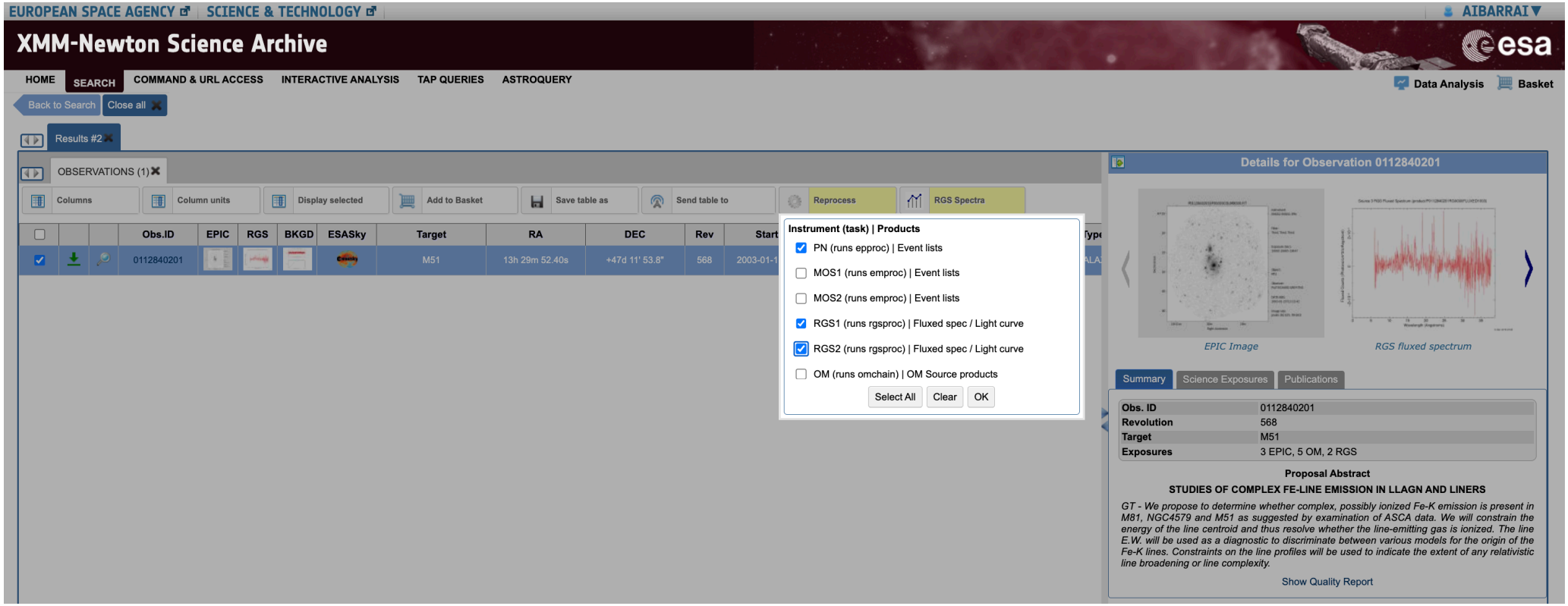
Processing from XSA (using RISA)

Processing from XSA (using RISA)

- Reprocessing

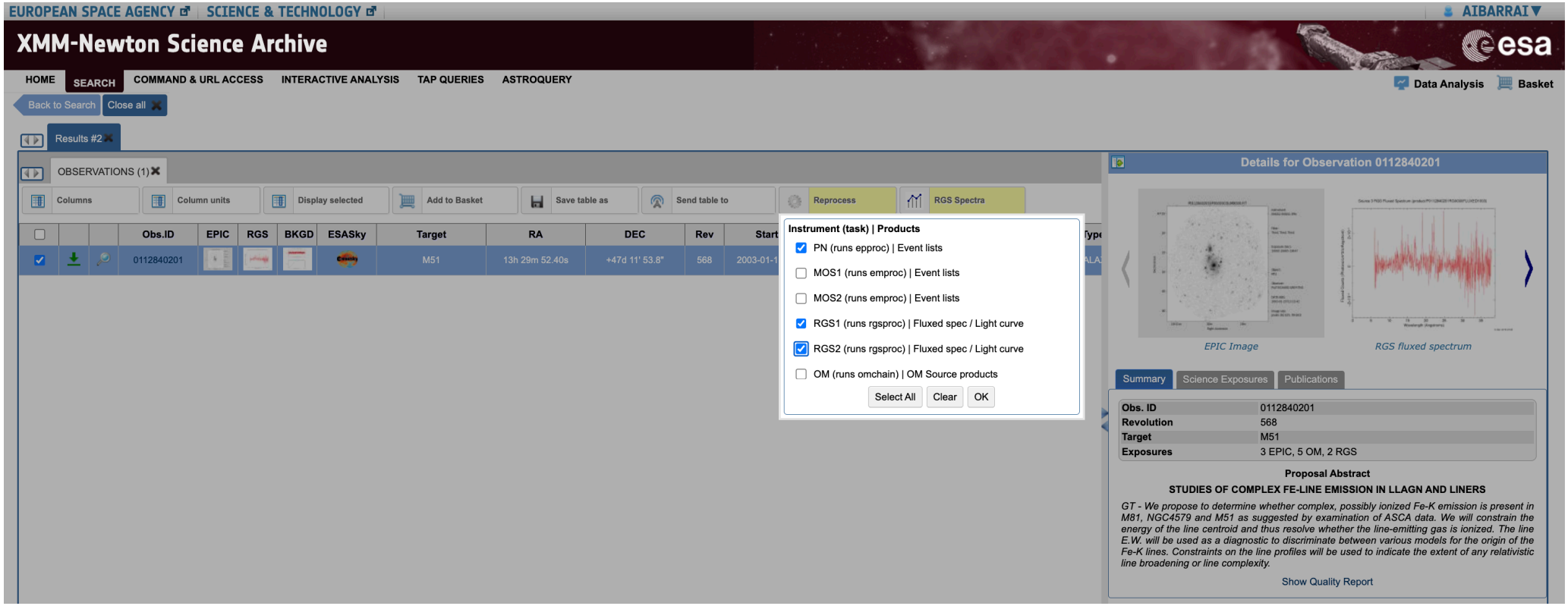
Processing from XSA (using RISA)

- Reprocessing

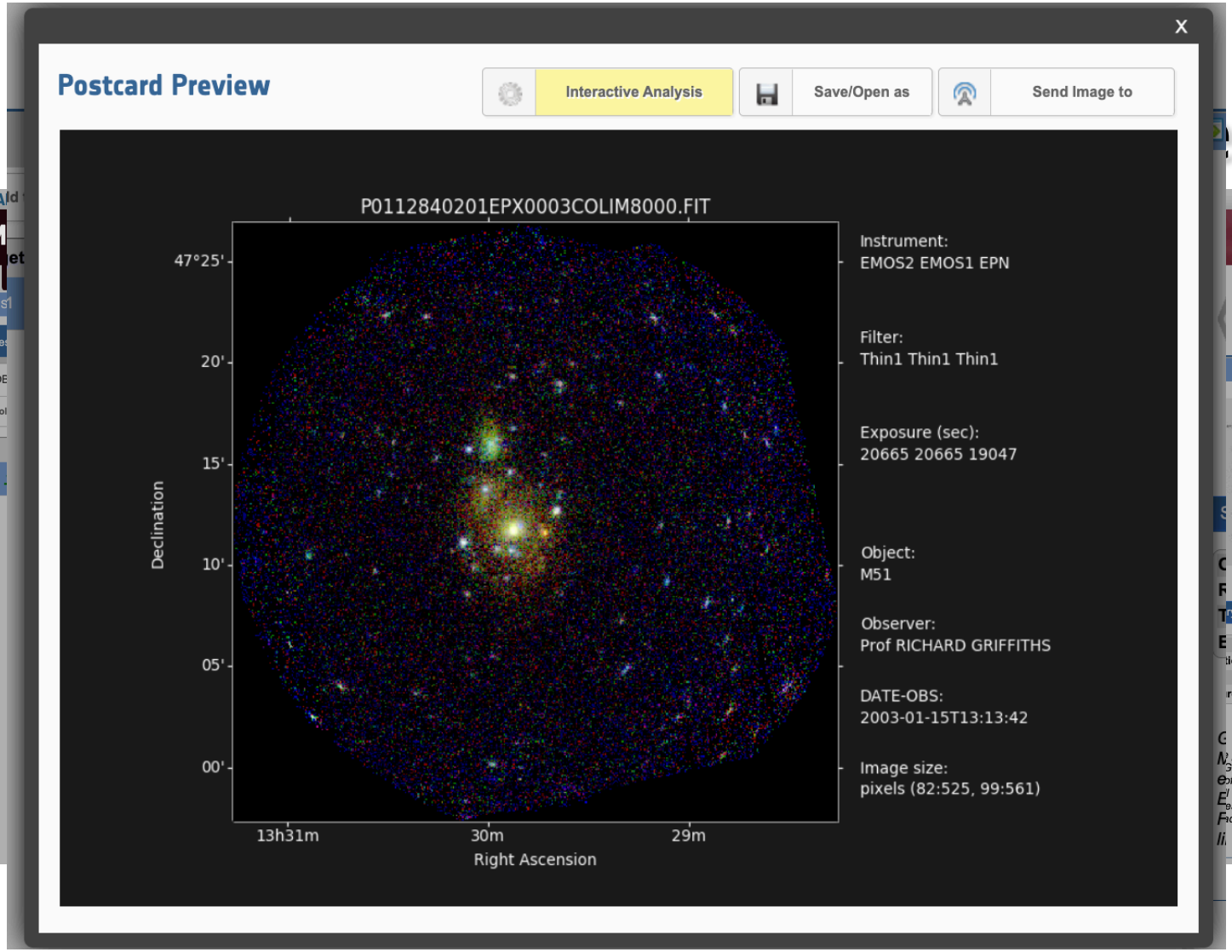


Processing from XSA (using RISA)

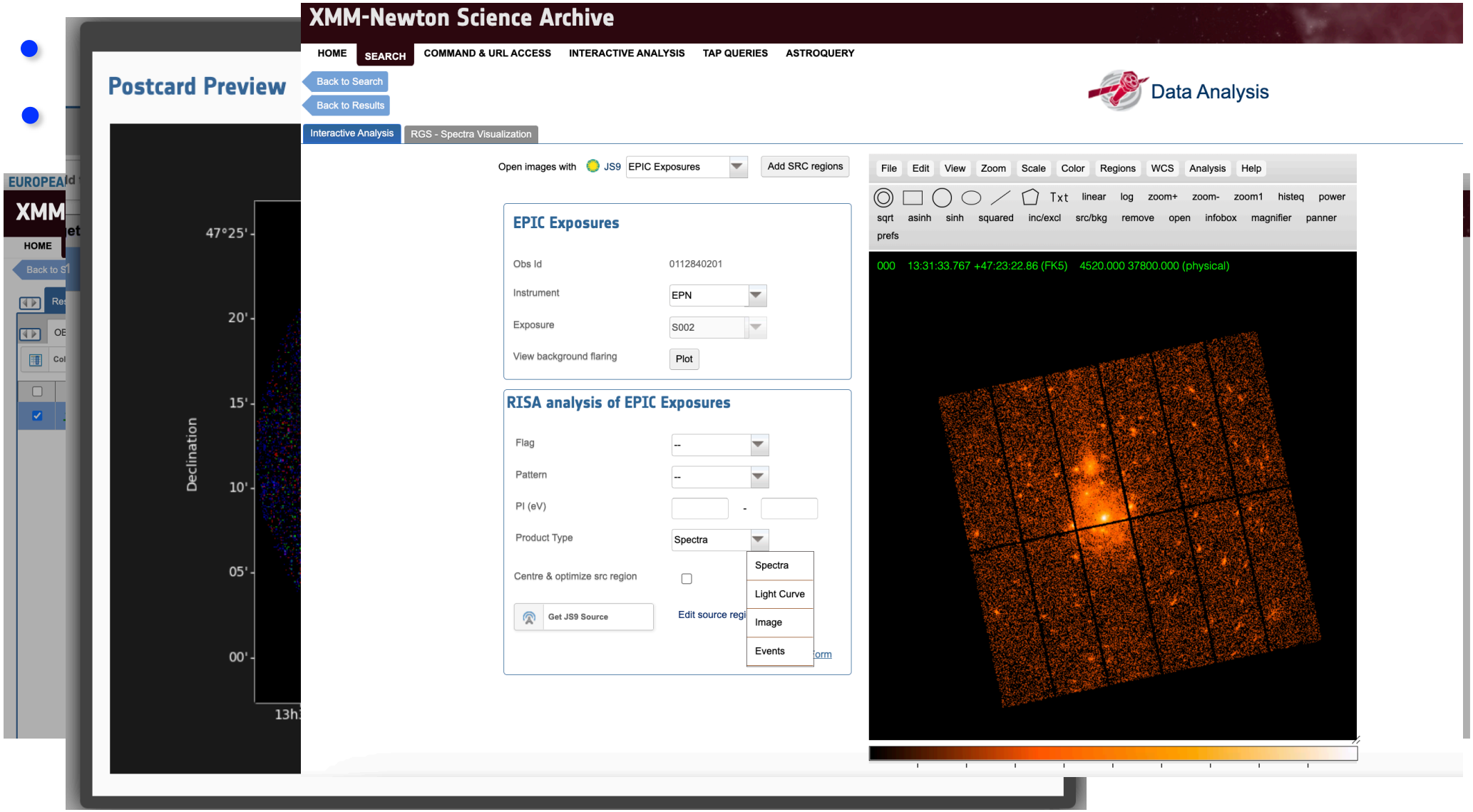
- Reprocessing
- EPIC Image, spectra or light curve



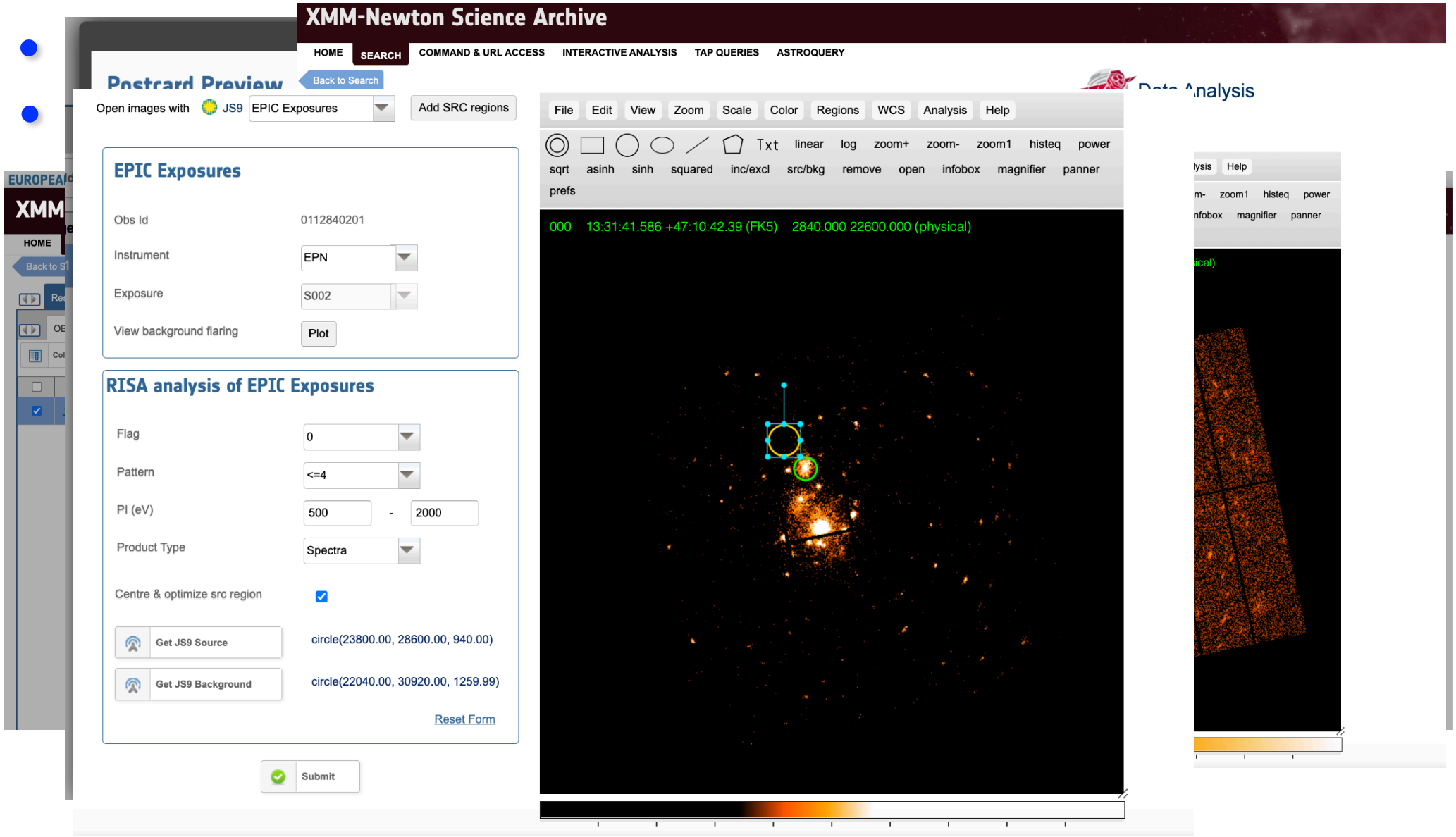
Processing from XSA (using RISA)



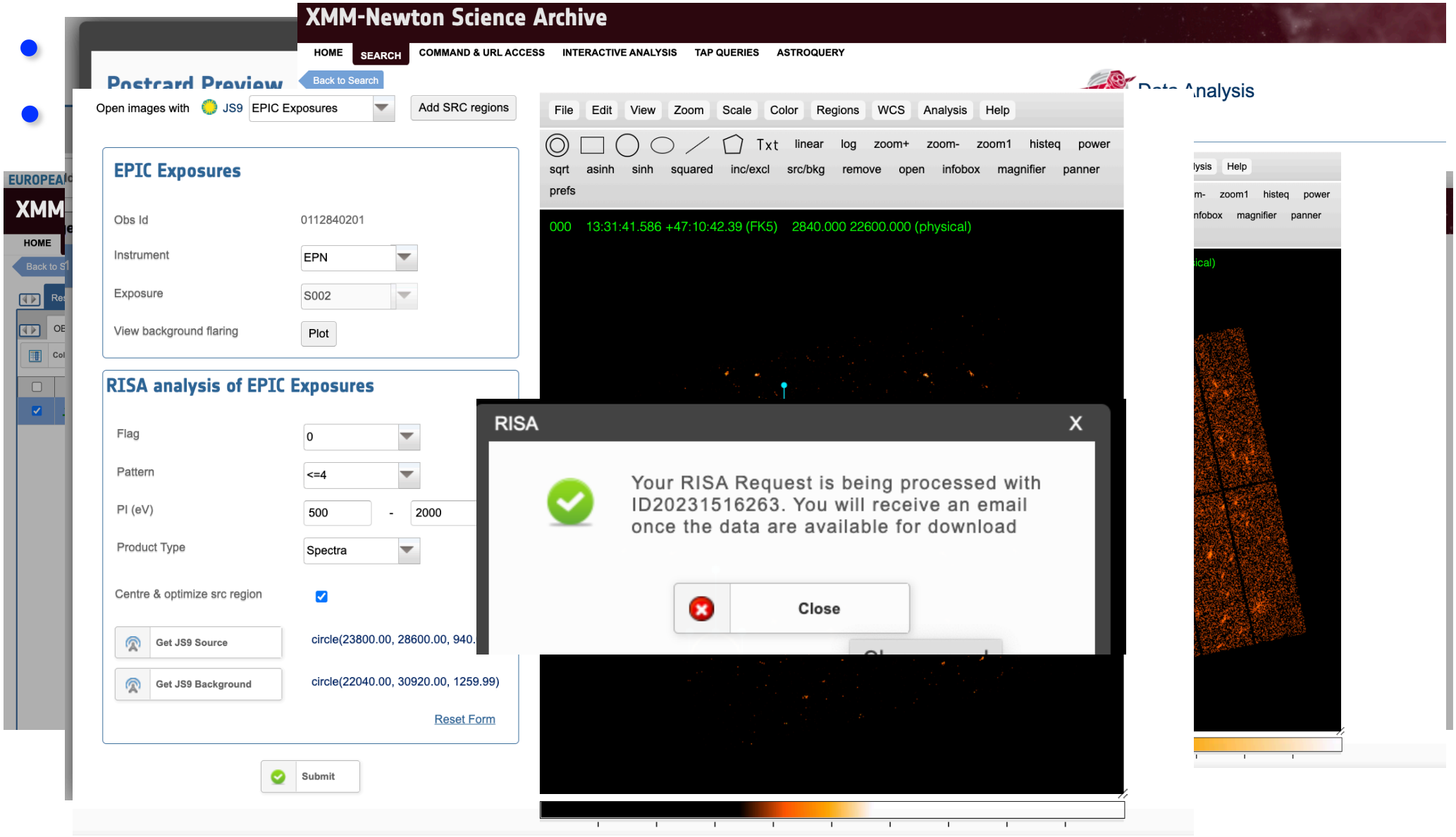
Processing from XSA (using RISA)



Processing from XSA (using RISA)

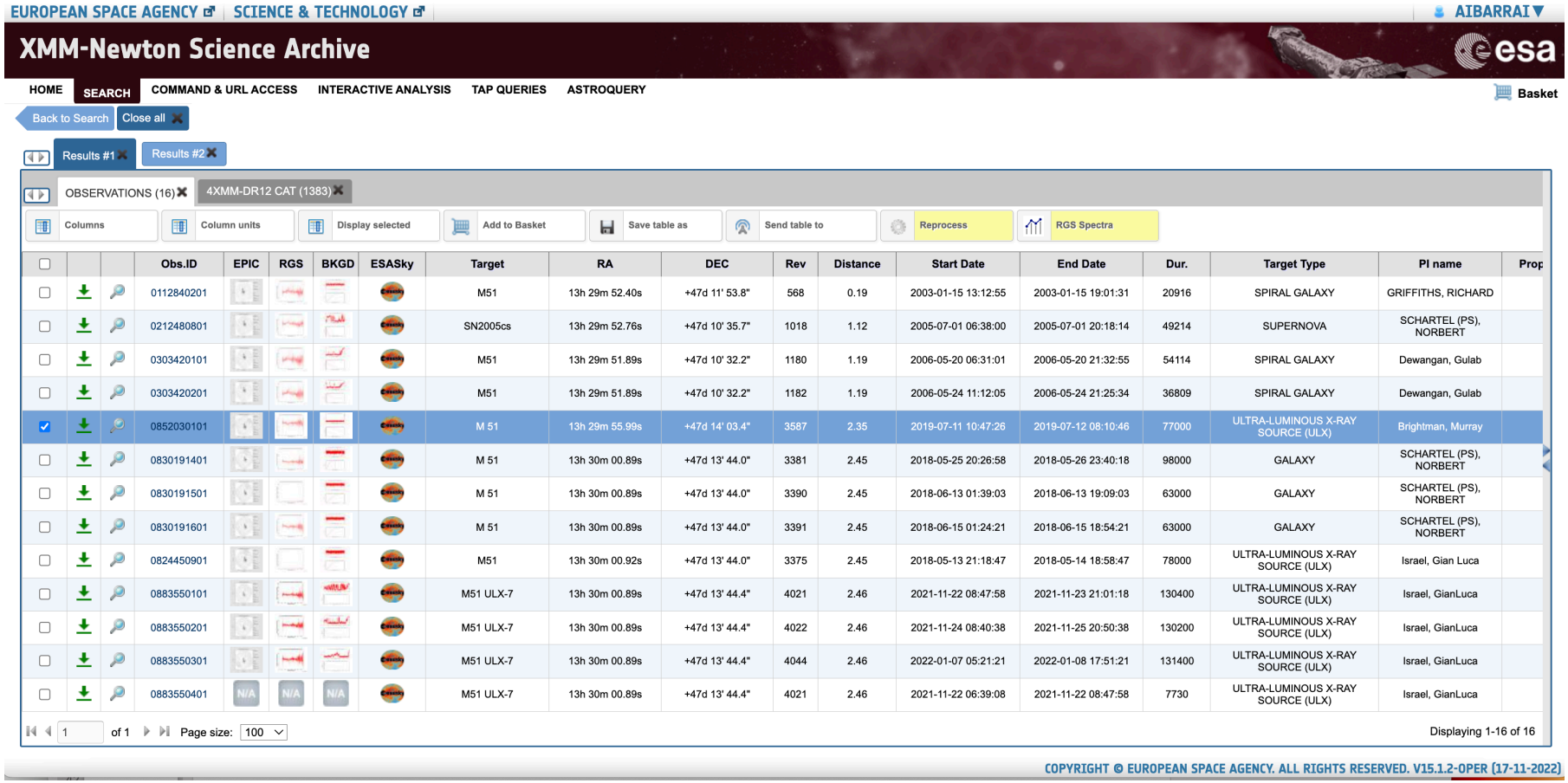


Processing from XSA (using RISA)



All the individual sources detected

All the individual sources detected



EUROPEAN SPACE AGENCY | SCIENCE & TECHNOLOGY

AIBARRAI

XMM-Newton Science Archive

HOME | SEARCH | COMMAND & URL ACCESS | INTERACTIVE ANALYSIS | TAP QUERIES | ASTROQUERY | Basket

Back to Search | Close all

Results #1 | Results #2

OBSERVATIONS (16) | 4XMM-DR12 CAT (1383)

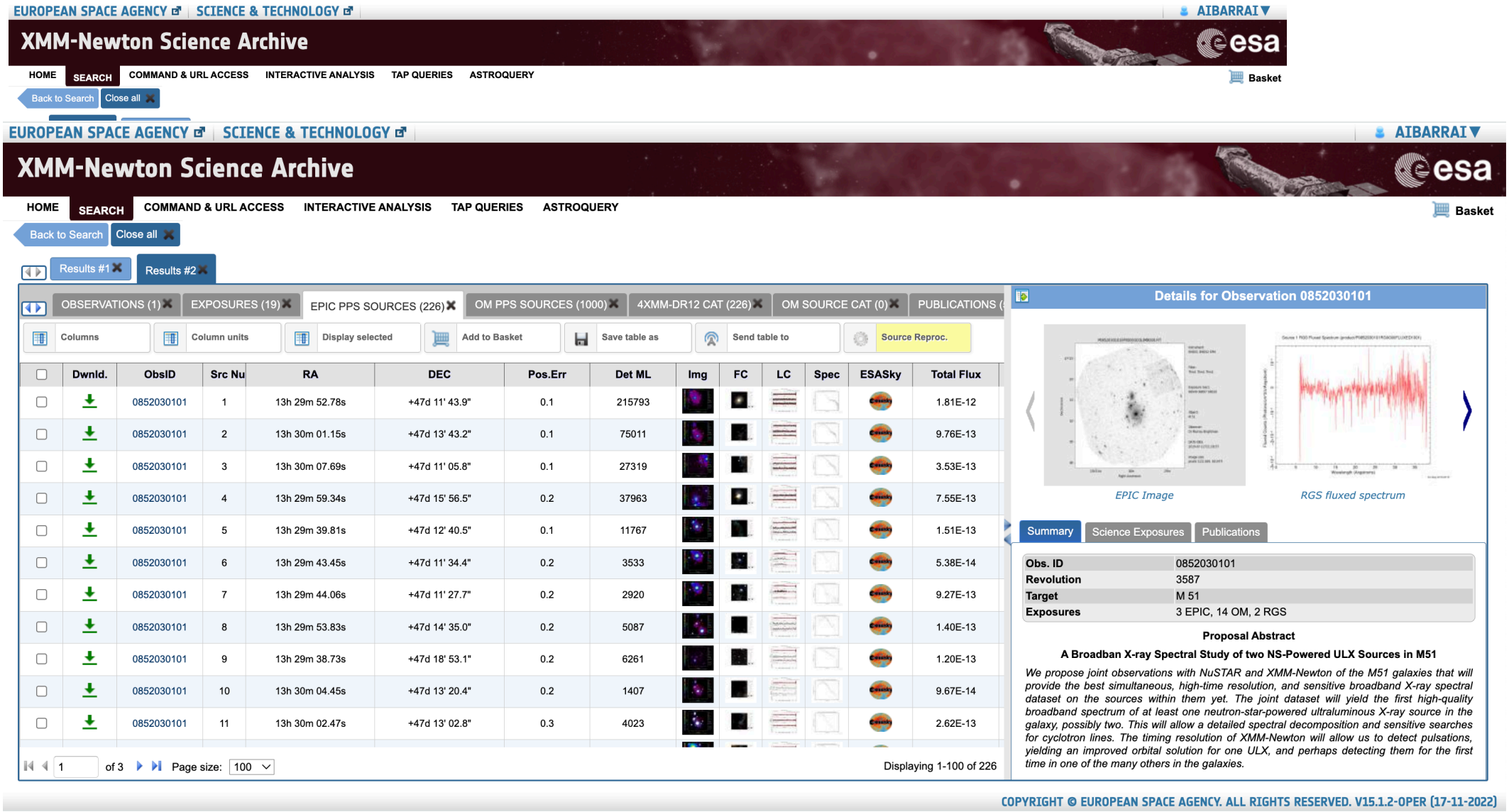
Columns | Column units | Display selected | Add to Basket | Save table as | Send table to | Reprocess | RGS Spectra

Obs.ID	EPIC	RGS	BKGD	ESASky	Target	RA	DEC	Rev	Distance	Start Date	End Date	Dur.	Target Type	PI name	Prop
0112840201					M51	13h 29m 52.40s	+47d 11' 53.8"	568	0.19	2003-01-15 13:12:55	2003-01-15 19:01:31	20916	SPIRAL GALAXY	GRIFFITHS, RICHARD	
0212480801					SN2005cs	13h 29m 52.76s	+47d 10' 35.7"	1018	1.12	2005-07-01 06:38:00	2005-07-01 20:18:14	49214	SUPERNOVA	SCHARTEL (PS), NORBERT	
0303420101					M51	13h 29m 51.89s	+47d 10' 32.2"	1180	1.19	2006-05-20 06:31:01	2006-05-20 21:32:55	54114	SPIRAL GALAXY	Dewangan, Gulab	
0303420201					M51	13h 29m 51.89s	+47d 10' 32.2"	1182	1.19	2006-05-24 11:12:05	2006-05-24 21:25:34	36809	SPIRAL GALAXY	Dewangan, Gulab	
0852030101					M 51	13h 29m 55.99s	+47d 14' 03.4"	3587	2.35	2019-07-11 10:47:26	2019-07-12 08:10:46	77000	ULTRA-LUMINOUS X-RAY SOURCE (ULX)	Brightman, Murray	
0830191401					M 51	13h 30m 00.89s	+47d 13' 44.0"	3381	2.45	2018-05-25 20:26:58	2018-05-26 23:40:18	98000	GALAXY	SCHARTEL (PS), NORBERT	
0830191501					M 51	13h 30m 00.89s	+47d 13' 44.0"	3390	2.45	2018-06-13 01:39:03	2018-06-13 19:09:03	63000	GALAXY	SCHARTEL (PS), NORBERT	
0830191601					M 51	13h 30m 00.89s	+47d 13' 44.0"	3391	2.45	2018-06-15 01:24:21	2018-06-15 18:54:21	63000	GALAXY	SCHARTEL (PS), NORBERT	
0824450901					M51	13h 30m 00.92s	+47d 13' 44.0"	3375	2.45	2018-05-13 21:18:47	2018-05-14 18:58:47	78000	ULTRA-LUMINOUS X-RAY SOURCE (ULX)	Israel, Gian Luca	
0883550101					M51 ULX-7	13h 30m 00.89s	+47d 13' 44.4"	4021	2.46	2021-11-22 08:47:58	2021-11-23 21:01:18	130400	ULTRA-LUMINOUS X-RAY SOURCE (ULX)	Israel, GianLuca	
0883550201					M51 ULX-7	13h 30m 00.89s	+47d 13' 44.4"	4022	2.46	2021-11-24 08:40:38	2021-11-25 20:50:38	130200	ULTRA-LUMINOUS X-RAY SOURCE (ULX)	Israel, GianLuca	
0883550301					M51 ULX-7	13h 30m 00.89s	+47d 13' 44.4"	4044	2.46	2022-01-07 05:21:21	2022-01-08 17:51:21	131400	ULTRA-LUMINOUS X-RAY SOURCE (ULX)	Israel, GianLuca	
0883550401	N/A	N/A	N/A		M51 ULX-7	13h 30m 00.89s	+47d 13' 44.4"	4021	2.46	2021-11-22 06:39:08	2021-11-22 08:47:58	7730	ULTRA-LUMINOUS X-RAY SOURCE (ULX)	Israel, GianLuca	

1 of 1 | Page size: 100 | Displaying 1-16 of 16

COPYRIGHT © EUROPEAN SPACE AGENCY. ALL RIGHTS RESERVED. V15.1.2-OPER (17-11-2022)

All the individual sources detected



All the individual sources detected

EUROPEAN SPACE AGENCY | SCIENCE & TECHNOLOGY

AIBARRAI

XMM-Newton Science Archive

HOME | SEARCH | COMMAND & URL ACCESS | INTERACTIVE ANALYSIS | TAP QUERIES | ASTROQUERY

Basket

EUROPEAN SPACE AGENCY | SCIENCE & TECHNOLOGY

AIBARRAI

XMM-Newton Science Archive

HOME | SEARCH | COMMAND & URL ACCESS | INTERACTIVE ANALYSIS | TAP QUERIES | ASTROQUERY

Basket

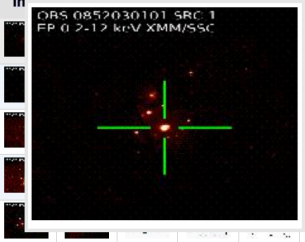
Back to Search | Close all

Results #1 | Results #2

OBSERVATIONS (1) | EXPOSURES (19) | EPIC PPS SOURCES (226) | OM PPS SOURCES (1000) | 4XMM-DR12 CAT (226) | OM SOURCE CAT (0) | PUBLICATIONS (5) | PROPOSALS (1)

Columns | Column units | Display selected | Save table as | Send table to

Dwnld.	Obs.ID	IAU name	RA	DEC	PosErr	OFFAXIS_PN	EP_8 Det ML	QUALITY FLAG	Im	Thmb	ESASky	SED 0	SED 1	SED 2	EP_8 Rate	EP_8 Rate Err
	0852030101	4XMM J132952.6+471144	13h 29m 52.63s	+47d 11' 44.3"	0.3	2.1	215793	Suspect parameters						N/A	1.48E00	5.80E-03
	0852030101	4XMM J133000.9+471343	13h 30m 01.00s	+47d 13' 43.6"	0.3	2.0	75011	Suspect parameters						N/A	5.11E-01	3.29E-03
	0852030101	4XMM J133007.5+471106	13h 30m 07.54s	+47d 11' 06.2"	0.3	4.1	27319	Suspect parameters				N/A	N/A	N/A	2.14E-01	2.24E-03
	0852030101	4XMM J132959.2+471557	13h 29m 59.19s	+47d 15' 56.9"	0.3	2.8	37963	Suspect parameters					N/A	N/A	5.43E-01	4.17E-03
	0852030101	4XMM J132939.7+471239	13h 29m 39.66s	+47d 12' 40.9"	0.3	2.0	11767	Suspect parameters				N/A	N/A	N/A	1.12E-01	1.64E-03
	0852030101	4XMM J132943.2+471134	13h 29m 43.30s	+47d 11' 34.8"	0.3	2.4	3533	Suspect parameters					N/A	N/A	6.49E-02	1.68E-03
	0852030101	4XMM J132943.9+471128	13h 29m 43.90s	+47d 11' 28.1"	0.3	2.5	2920	In suspect area				N/A	N/A	N/A	7.35E-01	2.55E-02
	0852030101	4XMM J132953.6+471435	13h 29m 53.68s	+47d 14' 35.4"	0.3	1.1	5087	Suspect parameters				N/A	N/A	N/A	6.93E-02	1.41E-03
	0852030101	4XMM J132938.5+471854	13h 29m 38.57s	+47d 18' 53.5"	0.4	5.4	6261	Good							7.66E-02	1.49E-03
	0852030101	4XMM J133004.2+471320	13h 30m 04.30s	+47d 13' 20.8"	0.4	2.6	1407	Suspect parameters					N/A	N/A	3.46E-02	1.13E-03
	0852030101	4XMM J133002.1+471304	13h 30m 02.32s	+47d 13' 03.2"	0.4	2.3	4023	Suspect parameters				N/A	N/A	N/A	2.83E-01	4.91E-03
	0852030101	4XMM J132950.8+471031	13h 29m 50.89s	+47d 10' 31.4"	0.4	3.3	1067	Suspect parameters				N/A	N/A	N/A	3.05E-02	1.12E-03
	0852030101	4XMM J132950.4+470956	13h 29m 50.40s	+47d 09' 56.1"	0.4	3.9	1222	In suspect area				N/A	N/A	N/A	5.45E-01	1.85E-02



1 of 3 | Page size: 100

Displaying 1-100 of 226

How to find files in the XMM-Newton data forest

EPIC:

MIEVLI / *PIEVLI* files are the event list files in PPS
*[Imaging-Timing]*Evts.ds files are the corresponding ones in PROC
IMAGE_8000.FTZ are the FITS compressed whole camera images in PPS
The only EPIC products from e[m-p]proc are the event list files and Bad Pixel tables

RGS:

*R1*EVENLI* / *R2*EVENLI* are the event list files in PPS and PROC
*R1*SRCLI* / *R2*SRCLI* are the source list files in PPS and PROC
SRSPEC are the source subtracted spectra in PPS and PROC
Response matrices are products in PROC (*Matrix*), as well as fluxed spectra (*fluxed*)
Intermediate RGS products are only kept in PROC (not in PPS), therefore RGS analysis starting on intermediate point only possible with PROC products

OM:

Whole analysis done by pipeline – Check source detections (using eg. implot, ds9, ftools)
If necessary, re-start analysis at intermediate steps (omdetect in om[i-f]thread.html)

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName

How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName
... but also to find everything produced by PPS (and that's a lot!)

How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName ... but also to find everything produced by PPS (and that's a lot!)

XMM SURVEY SCIENCE CENTRE esa

OBS Summary | PPS Summary | EPIC Summary | OM Summary | RGS Summary | Catalogue Summary

0303420201 Observation Data Summary File

Revolution	Target	Scheduled Length	Observer
1182	M51	36809	Dr Gulab Dewangan

Guest Observer Information | Proposal Target Information | Observation Record | Instrument Information | Exposure and Configuration Information

Guest Observer Information

Dr Gulab Dewangan
Carnegie Mellon University
Department of Physics
5000 Forbes Avenue
Pittsburgh
PA
UNITED STATES
15213
gulabd@cmu.edu

How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName ... but also to find everything produced by PPS (and that's a lot!)

XMM SURVEY SCIENCE CENTRE esa

OBS Summary | PPS Summary | EPIC Summary | OM Summary | RGS Summary | Catalogue Summary

0303420201 PPS Processing Summary

Revolution	Target	Scheduled Length	Observer
1182	M51	36809	Dr Gulab Dewangan

EPIC Exposures Processed | OM Exposures Processed | RGS Exposures Processed | Generic Products

EPIC Exposures processed by PPS

Inst.	Exp. Id	Sched	Mode	Datamode	Filter	Position	Duration	Exposure	SrcDet	SSP	Flare Scrn	Actual Start	Actual End
EMOS1	S001	Y	PrimeFullWindow	Imaging	THIN1	FILTER_A	36574	25015	Y	Y	Y	2006-05-24T11:12:42	2006-05-24T21:22:16
EMOS2	S002	Y	PrimeFullWindow	Imaging	THIN1	FILTER_A	36578	25412	Y	Y	Y	2006-05-24T11:12:43	2006-05-24T21:22:21
EPN	S003	Y	PrimeFullWindow	Imaging	THIN1		34997	22545	Y	Y	Y	2006-05-24T11:35:02	2006-05-24T21:18:19

OM Exposures processed by PPS

Inst	ExpID	Sched	Detector	Mode	Data Mode	Filter	Duration	Actual Start	Actual End
OM	S006	Y	REDUNDANT	Image	Engineering4	UVW1	3400	2006-05-24T11:35:40	2006-05-24T12:32:20
OM	S007	Y	REDUNDANT	Image	Engineering4	UVW2	3398	2006-05-24T15:20:22	2006-05-24T16:17:00
OM	S008	Y	REDUNDANT	Image	Engineering4	UVW2	3375	2006-05-24T18:35:00	2006-05-24T19:31:15

RGS Exposures processed by PPS

Inst.	Exp. Id	Sched	Mode	Datamode	Event Filtering	Duration	Actual Start	Actual End
RGS1	S004	Y	HighEventRateWithSES	Spectroscopy	rejflags,attGTI,hkGTI	36806	2006-05-24T11:12:05	2006-05-24T21:25:31
RGS2	S005	Y	HighEventRateWithSES	Spectroscopy	rejflags,attGTI,hkGTI	36809	2006-05-24T11:12:05	2006-05-24T21:25:34

How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName ... but also to find everything produced by PPS (and that's a lot!)

XMM SURVEY SCIENCE CENTRE esa

OBS Summary | PPS Summary | EPIC Summary | OM Summary | RGS Summary | Catalogue

0303420201 EPIC Processing Summary

Revolution	Target	Scheduled Length	Observer
1182	M51	36809	Dr Gulab Dewangan

Multi-Exposure Products | Exposures processed by PPS | Sources processed by PPS

Multi-Exposure Products

Filename	Content	Band
P0303420201EPX000OBKGMP8000.FTZ	EPIC OBSERVATION BACKGROUND MAP	0.2 - 12.0
P0303420201EPX000OBKGMP8000.PNG	EPIC OBSERVATION BACKGROUND MAP	0.2 - 12.0
P0303420201EPX000OBLSLI0000.FTZ	EPIC OBSERVATION BOX-LOCAL SOURCE LIST	
P0303420201EPX000OBMSLI0000.FTZ	EPIC OBSERVATION BOX-MAP SOURCE LIST	
P0303420201EPX000OBSMLI0000.FTZ	EPIC SUMMARY SOURCE LIST	
P0303420201EPX000OBSMLI0000.HTM	EPIC SUMMARY SOURCE LIST	
P0303420201EPX000OEXPMP8000.FTZ	EPIC OBSERVATION EXPOSURE MAP	0.2 - 12.0
P0303420201EPX000OEXPMP8000.PNG	EPIC OBSERVATION EXPOSURE MAP	0.2 - 12.0
P0303420201EPX000OIMAGE8000.FTZ	EPIC OBSERVATION IMAGE	0.2 - 12.0
P0303420201EPX000OIMAGE8000.PNG	EPIC OBSERVATION IMAGE	0.2 - 12.0
P0303420201EPX000OMSRLI0000.FTZ	EPIC OBSERVATION ML SOURCE LIST	
P0303420201EPX000OSNSMP8000.FTZ	EPIC OBSERVATION SENSITIVITY MAP	0.2 - 12.0
P0303420201EPX000REGION0000.ASC	EPIC SOURCE DS9 REGIONS	
P0303420201M1X000BKGMAP1000.FTZ	EPIC MERGED BACKGROUND MAP	0.2 - 0.5
P0303420201M1X000BKGMAP2000.FTZ	EPIC MERGED BACKGROUND MAP	0.5 - 1.0
P0303420201M1X000BKGMAP3000.FTZ	EPIC MERGED BACKGROUND MAP	1.0 - 2.0
P0303420201M1X000BKGMAP4000.FTZ	EPIC MERGED BACKGROUND MAP	2.0 - 4.5
P0303420201M1X000BKGMAP5000.FTZ	EPIC MERGED BACKGROUND MAP	4.5 - 12.0
P0303420201M1X000DETMSK1000.FTZ	EPIC DETECTION MASK	0.2 - 0.5
P0303420201M1X000DETMSK2000.FTZ	EPIC DETECTION MASK	0.5 - 1.0

How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName ... but also to find everything produced by PPS (and that's a lot!)

XMM SURVEY SCIENCE CENTRE esa

OBS Summary | PPS Summary | EPIC Summary | OM Summary | RGS Summary | Catalogue

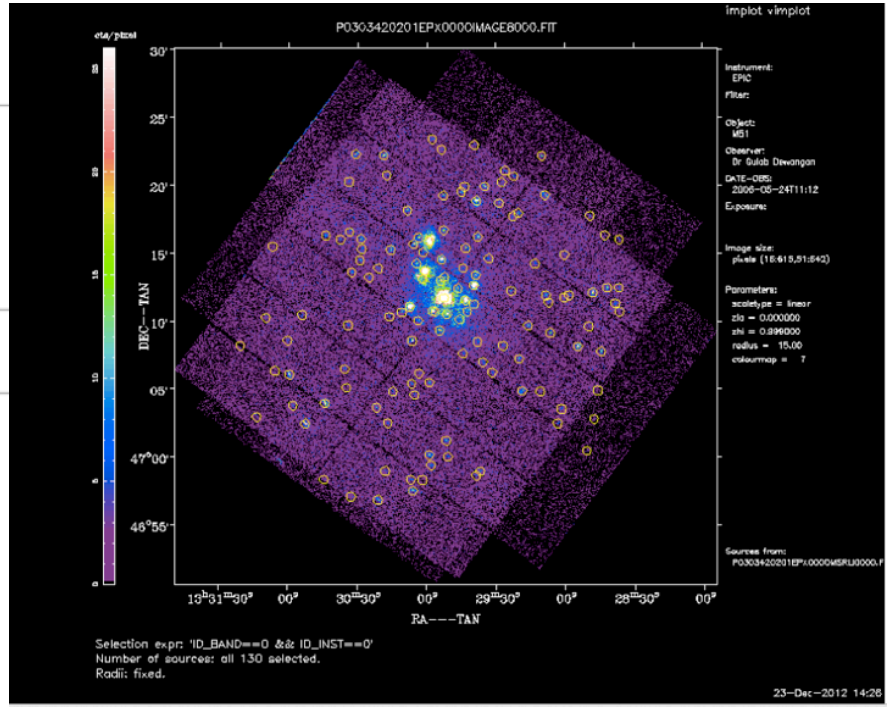
0303420201 EPIC Processing Summary

Revolution	Target	Scheduled Length	Observer
1182	M51	36809	Dr Gulab Dewangan

Multi-Exposure Products | Exposures processed by PPS | Sources processed by PPS

Multi-Exposure Products

Filename	Content	Band
P0303420201EPX000OBKGMP8000.FTZ	EPIC OBSERVATION BACKGROUND MAP	0.2 - 12.0
P0303420201EPX000OBKGMP8000.PNG	EPIC OBSERVATION BACKGROUND MAP	0.2 - 12.0
P0303420201EPX000OBLSLI0000.FTZ	EPIC OBSERVATION BOX-LOCAL SOURCE LIST	
P0303420201EPX000OBMSLI0000.FTZ	EPIC OBSERVATION BOX-MAP SOURCE LIST	
P0303420201EPX000OBSMLI0000.FTZ	EPIC SUMMARY SOURCE LIST	
P0303420201EPX000OBSMLI0000.HTM	EPIC SUMMARY SOURCE LIST	
P0303420201EPX000OEXPMP8000.FTZ	EPIC OBSERVATION EXPOSURE MAP	0.2 - 12.0
P0303420201EPX000OEXPMP8000.PNG	EPIC OBSERVATION EXPOSURE MAP	0.2 - 12.0
P0303420201EPX000OIMAGE8000.FTZ	EPIC OBSERVATION IMAGE	0.2 - 12.0
P0303420201EPX000OIMAGE8000.PNG	EPIC OBSERVATION IMAGE	0.2 - 12.0
P0303420201EPX000OMSRLI0000.FTZ	EPIC OBSERVATION ML SOURCE LIST	
P0303420201EPX000OSNSMP8000.FTZ	EPIC OBSERVATION SENSITIVITY MAP	0.2 - 12.0
P0303420201EPX000REGION0000.ASC	EPIC SOURCE DS9 REGIONS	
P0303420201M1X000BKGMAP1000.FTZ	EPIC MERGED BACKGROUND MAP	0.2 - 0.5
P0303420201M1X000BKGMAP2000.FTZ	EPIC MERGED BACKGROUND MAP	0.5 - 1.0
P0303420201M1X000BKGMAP3000.FTZ	EPIC MERGED BACKGROUND MAP	1.0 - 2.0
P0303420201M1X000BKGMAP4000.FTZ	EPIC MERGED BACKGROUND MAP	2.0 - 4.5
P0303420201M1X000BKGMAP5000.FTZ	EPIC MERGED BACKGROUND MAP	4.5 - 12.0
P0303420201M1X000DETMSK1000.FTZ	EPIC DETECTION MASK	0.2 - 0.5
P0303420201M1X000DETMSK2000.FTZ	EPIC DETECTION MASK	0.5 - 1.0



How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName ... but also to find everything produced by PPS (and that's a lot!)

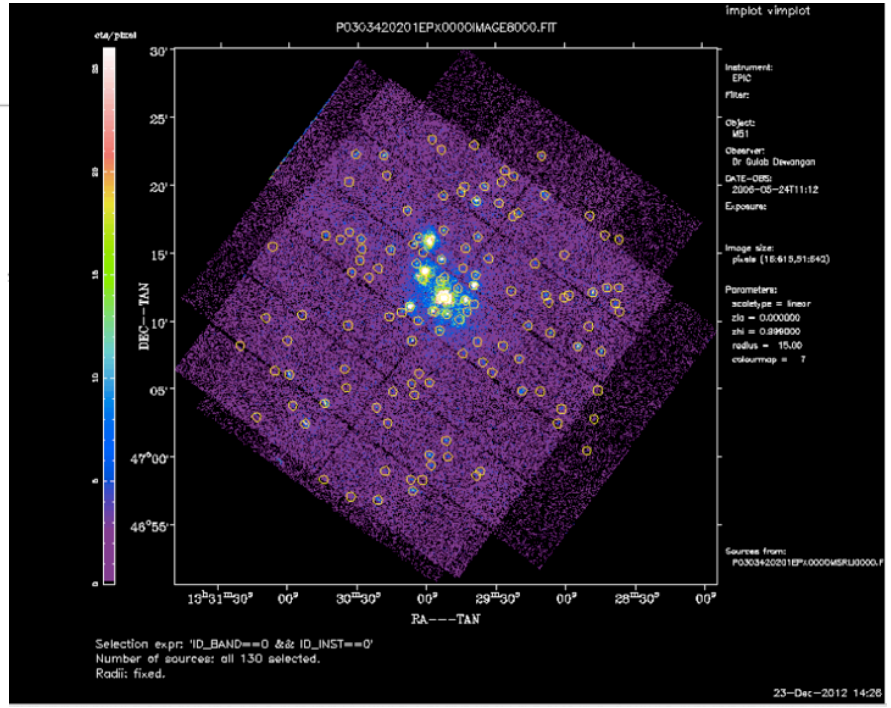
XMM SURVEY SCIENCE CENTRE esa

OBS Summary | PPS Summary | EPIC Summary | OM Summary | RGS Summary | Catalogue

0303420201 EPIC Processing Summary

Inst.	Exp. Id	Sched	Mode	Datamode	Filter	Position	Duration	Exposure	SrcDet	SSP	Flare Scrn
EPN	S003	Y	PrimeFullWindow	Imaging	THIN1		34997	22545	Y	Y	Y

Filename	Content	Band
P0303420201PNS003PIEVLI0000.FTZ	EPIC PN IMAGING MODE EVENT LIST	
P0303420201PNS003FBKTSR0000.FTZ	EPIC FLARE BACKGROUND TIMESERIES	
P0303420201PNS003FBKTSR0000.PDF	EPIC FLARE BACKGROUND TIMESERIES	
P0303420201PNS003EXPMAP1000.FTZ	EPIC EXPOSURE MAP	0.2 - 0.5
P0303420201PNS003IMAGE_1000.FTZ	EPIC IMAGE	0.2 - 0.5
P0303420201PNS003EXPMAP2000.FTZ	EPIC EXPOSURE MAP	0.5 - 1.0
P0303420201PNS003IMAGE_2000.FTZ	EPIC IMAGE	0.5 - 1.0
P0303420201PNS003EXPMAP3000.FTZ	EPIC EXPOSURE MAP	1.0 - 2.0
P0303420201PNS003IMAGE_3000.FTZ	EPIC IMAGE	1.0 - 2.0
P0303420201PNS003EXPMAP4000.FTZ	EPIC EXPOSURE MAP	2.0 - 4.5
P0303420201PNS003IMAGE_4000.FTZ	EPIC IMAGE	2.0 - 4.5
P0303420201PNS003EXPMAP5000.FTZ	EPIC EXPOSURE MAP	4.5 - 12.0
P0303420201PNS003IMAGE_5000.FTZ	EPIC IMAGE	4.5 - 12.0
P0303420201PNS003EXPMAP8000.FTZ	EPIC EXPOSURE MAP	0.2 - 12.0
P0303420201PNS003EXPMAP8000.PNG	EPIC EXPOSURE MAP	0.2 - 12.0
P0303420201PNS003IMAGE_8000.FTZ	EPIC IMAGE	0.2 - 12.0
P0303420201PNS003IMAGE_8000.PNG	EPIC IMAGE	0.2 - 12.0



Sources processed by PPS

Source	Inst.	Exp. Id	RA	Dec	RADec Err	Count	Band 8 Flux	EPIC Det ML	Inst. Det ML	Ontime	EPIC Extent	EPIC Flags	Srcdet
Dec (Hex)			[hms]	[° ′ ″]	[″]		[erg/s/cm^2]			[s]	[image pixels]		
1 (001)	EMOS1	S001	13 29 52.54	47 11 44.6	0.2	6105.9 ±90.4	1.87e-12 ±5.13e-14	50374	14744	25012	2 ±0	FFTFFFFFFFFF	Y
1 (001)	EMOS2	S002	13 29 52.54	47 11 44.6	0.2	5649.5 ±87.6	1.86e-12 ±5.39e-14	50374	13048	25389	2 ±0	FFTFFFFFFFFF	Y
1 (001)	EPN	S003	13 29 52.54	47 11 44.6	0.2	12549.8 ±138.9	1.5e-12 ±3.32e-14	50374	22575	22545	2 ±0	FFTFFFFFFFFF	Y
2 (002)	EMOS1	S001	13 30 0.93	47 13 43.6	0.1	1932.0 ±49.0	1.19e-12 ±5.36e-14	21364	5121	25012	0	FFTFFFFFFFFF	Y
2 (002)	EMOS2	S002	13 30 0.93	47 13 43.6	0.1	1823.2 ±47.3	1.16e-12 ±5.52e-14	21364	4692	25389	0	FFTFFFFFFFFF	Y
2 (002)	EPN	S003	13 30 0.93	47 13 43.6	0.1	5078.1 ±79.9	1e-12 ±2.8e-14	21364	11555	22545	0	FFTFFFFFFFFF	Y
3 (003)	EMOS1	S001	13 30 7.48	47 11 6.4	0.2	667.9 ±29.5	3.13e-13 ±2.58e-14	6822	1367	25012	0	FFTFFFFFFFFF	Y

How to find files in the XMM-Newton data forest

PPS Index Summary can help a lot to recognize correspondence FileType <-> FileName ... but also to find everything produced by PPS (and that's a lot!)

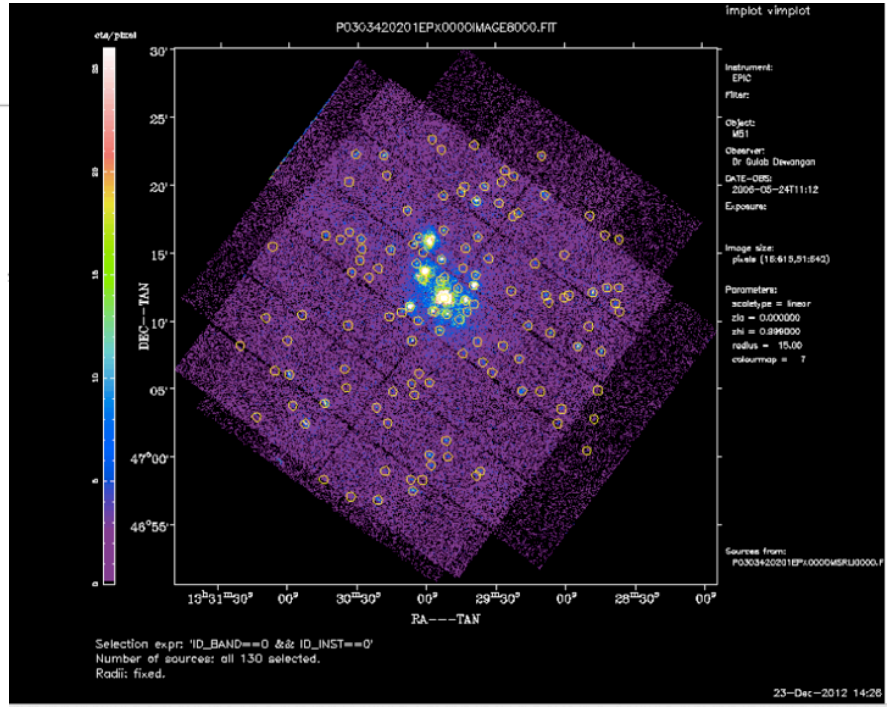
XMM SURVEY SCIENCE CENTRE esa

OBS Summary | PPS Summary | EPIC Summary | OM Summary | RGS Summary | Catalogue

0303420201 EPIC Processing Summary

Inst.	Exp. Id	Sched	Mode	Datamode	Filter	Position	Duration	Exposure	SrcDet	SSP	Flare Scrn
EPN	S003	Y	PrimeFullWindow	Imaging	THIN1		34997	22545	Y	Y	Y

Filename	Content	Band
P0303420201PNS003PIEVLI0000.FTZ	EPIC PN IMAGING MODE EVENT LIST	
P0303420201PNS003FBKTSR0000.FTZ	EPIC FLARE BACKGROUND TIMESERIES	
P0303420201PNS003FBKTSR0000.PDF	EPIC FLARE BACKGROUND TIMESERIES	
P0303420201PNS003EXPMAP1000.FTZ	EPIC EXPOSURE MAP	0.2 - 0.5
P0303420201PNS003IMAGE_1000.FTZ	EPIC IMAGE	0.2 - 0.5
P0303420201PNS003EXPMAP2000.FTZ	EPIC EXPOSURE MAP	0.5 - 1.0
P0303420201PNS003IMAGE_2000.FTZ	EPIC IMAGE	0.5 - 1.0
P0303420201PNS003EXPMAP3000.FTZ	EPIC EXPOSURE MAP	1.0 - 2.0
P0303420201PNS003IMAGE_3000.FTZ	EPIC IMAGE	1.0 - 2.0
P0303420201PNS003EXPMAP4000.FTZ	EPIC EXPOSURE MAP	2.0 - 4.5
P0303420201PNS003IMAGE_4000.FTZ	EPIC IMAGE	2.0 - 4.5
P0303420201PNS003EXPMAP5000.FTZ	EPIC EXPOSURE MAP	4.5 - 12.0
P0303420201PNS003IMAGE_5000.FTZ	EPIC IMAGE	4.5 - 12.0
P0303420201PNS003EXPMAP8000.FTZ	EPIC EXPOSURE MAP	0.2 - 12.0
P0303420201PNS003EXPMAP8000.PNG	EPIC EXPOSURE MAP	0.2 - 12.0
P0303420201PNS003IMAGE_8		
P0303420201PNS003IMAGE_8		



Sources processed by PPS

Source	Inst.	Exp. Id	RA	Dec	RADec Err	Count	Band 8 Flux	EPIC Det ML	Inst. Det ML	Ontime	EPIC Extent	EPIC Flags	Srcdet
Dec (Hex)			[hms]	[° ′ ″]	[″]		[erg/s/cm^2]			[s]	[image pixels]		
1 (001)	EPN	S003	13 29 52.54	47 11 44.6	0.2	12549.8 ±138.9	1.5e-12 ±3.32e-14	50374	22575	22545	2 ±0	FFTFFFFFFFFF	Y

Filename	Content	Band
P0303420201PNS003SRSPEC0001.FTZ	EPIC SOURCE SPECTRUM	
P0303420201PNS003BGSPEC0001.FTZ	EPIC SOURCE BACKGROUND SPECTRUM	
P0303420201PNS003SPCPLT0001.PDF	EPIC SOURCE SPECTRUM PLOT	
P0303420201PNS003SRCTSR8001.FTZ	EPIC SOURCE TIMESERIES	0.2 - 12.0
P0303420201PNS003STSPLT0001.PDF	EPIC SOURCE TIMESERIES PLOT	
P0303420201PNS003SRCREG0001.ASC	EPIC SOURCE DS9 REGION	
P0303420201PNS003SRCARF0001.FTZ	EPIC ANCILLARY RESPONSE FUNCTION	

Source	Inst.	Exp. Id											
Dec (Hex)													
1 (001)	EMOS1	S001											
1 (001)	EMOS2	S002											
1 (001)	EPN	S003											
2 (002)	EMOS1	S001											
2 (002)	EMOS2	S002											
2 (002)	EPN	S003											
3 (003)	EMOS1	S001	13 30 7.48	47 11 6.4	0.2	667.9 ±29.5	3.13e-13 ±2.58e-14	6822	1367	25012	0	FFTFFFFFFFFF	Y

SAS documentation

SAS documentation

XMM-Newton » Data Analysis

- Home / Latest News
- Conferences & Meetings
- News
- General User Support
- Proposers Info
- Observers Info
- Data Analysis
- Archive, Pipeline & Catalogues
- Calibration & Background
- SOC Info
- About XMM-Newton
- Image Gallery
- Publications
- Other Links

XMM-NEWTON DATA ANALYSIS

SAS NEWS
News, special information...

WHAT IS THE SCIENTIFIC ANALYSIS SYSTEM (SAS)?
A comprehensive approach

HOW TO USE SAS
Guides, manuals, on-line documentation, background analysis, watchout items

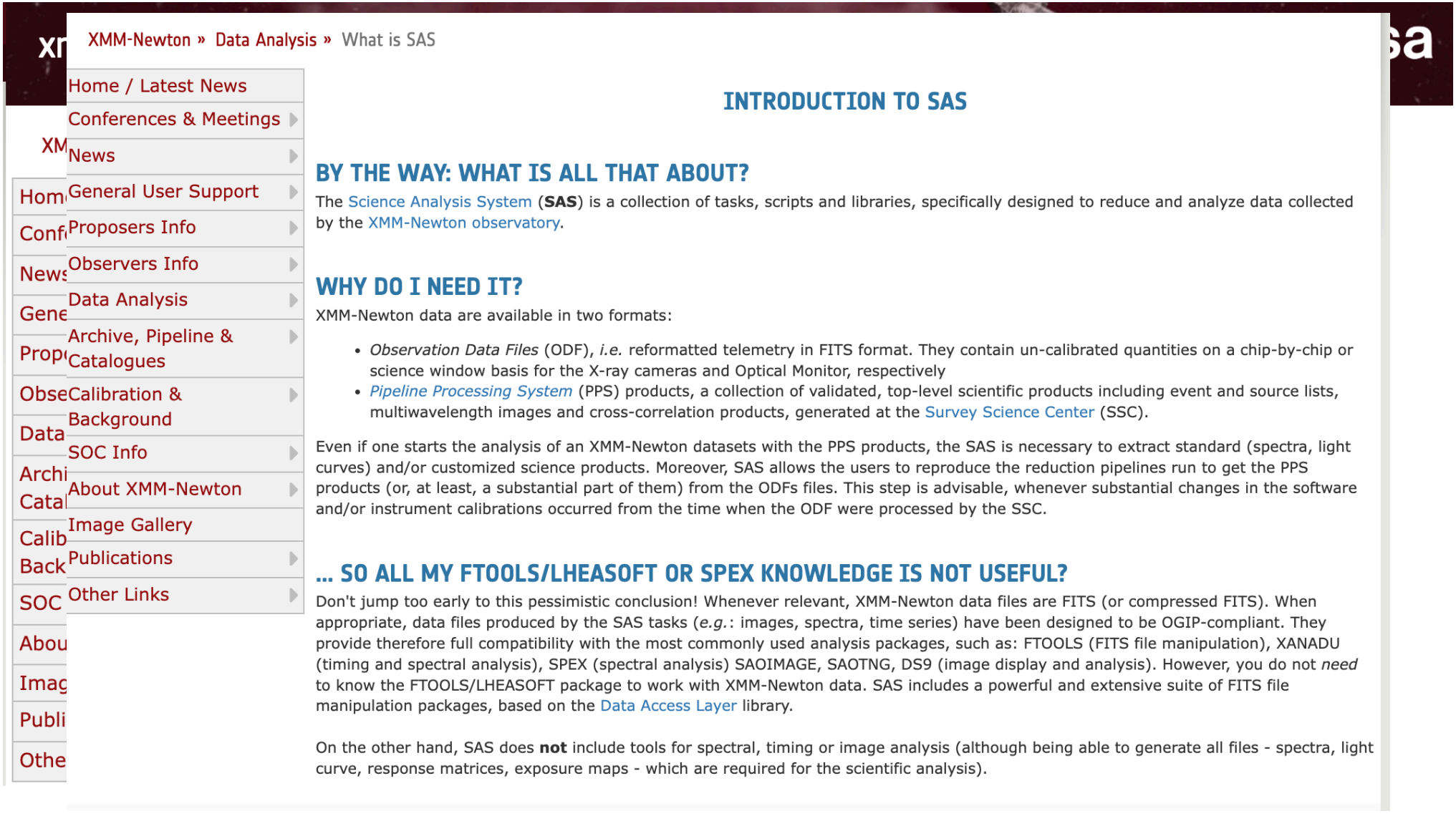
XMM-NEWTON SAS WORKSHOPS
Presentations from the latest Workshop, Information about the next SAS Workshop

SAS VERSION CHANGES
SAS version history, release notes, validation

DOWNLOAD AND INSTALL SAS
How to download and how to install SAS, which are the software requirements

XMM-NEWTON SCIENCE SIMULATOR
SciSim software to generate simulated XMM-Newton data

SAS documentation



XMM-Newton » Data Analysis » What is SAS

- Home / Latest News
- Conferences & Meetings
- News
- General User Support
- Proposers Info
- Observers Info
- Data Analysis
- Archive, Pipeline & Catalogues
- Calibration & Background
- SOC Info
- About XMM-Newton
- Image Gallery
- Publications
- Other Links

INTRODUCTION TO SAS

BY THE WAY: WHAT IS ALL THAT ABOUT?

The Science Analysis System (**SAS**) is a collection of tasks, scripts and libraries, specifically designed to reduce and analyze data collected by the XMM-Newton observatory.

WHY DO I NEED IT?

XMM-Newton data are available in two formats:

- *Observation Data Files* (ODF), *i.e.* reformatted telemetry in FITS format. They contain un-calibrated quantities on a chip-by-chip or science window basis for the X-ray cameras and Optical Monitor, respectively
- *Pipeline Processing System* (PPS) products, a collection of validated, top-level scientific products including event and source lists, multiwavelength images and cross-correlation products, generated at the Survey Science Center (SSC).

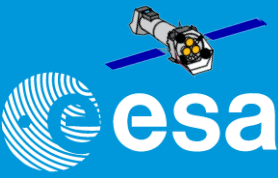
Even if one starts the analysis of an XMM-Newton datasets with the PPS products, the SAS is necessary to extract standard (spectra, light curves) and/or customized science products. Moreover, SAS allows the users to reproduce the reduction pipelines run to get the PPS products (or, at least, a substantial part of them) from the ODFs files. This step is advisable, whenever substantial changes in the software and/or instrument calibrations occurred from the time when the ODF were processed by the SSC.

... SO ALL MY FTOOLS/LHEASOFT OR SPEX KNOWLEDGE IS NOT USEFUL?

Don't jump too early to this pessimistic conclusion! Whenever relevant, XMM-Newton data files are FITS (or compressed FITS). When appropriate, data files produced by the SAS tasks (*e.g.*: images, spectra, time series) have been designed to be OGIP-compliant. They provide therefore full compatibility with the most commonly used analysis packages, such as: FTOOLS (FITS file manipulation), XANADU (timing and spectral analysis), SPEX (spectral analysis) SAOIMAGE, SAOTNG, DS9 (image display and analysis). However, you do not *need* to know the FTOOLS/LHEASOFT package to work with XMM-Newton data. SAS includes a powerful and extensive suite of FITS file manipulation packages, based on the Data Access Layer library.

On the other hand, SAS does **not** include tools for spectral, timing or image analysis (although being able to generate all files - spectra, light curve, response matrices, exposure maps - which are required for the scientific analysis).

SAS documentation



XMM-Newton » Data Analysis » How to use SAS

- Home / Latest News
- Conferences & Meetings
- News
- General User Support
- Proposers Info
- Observers Info
- Data Analysis
- Archive, Pipeline & Catalogues
- Calibration & Background
- SOC Info
- About XMM-Newton
- Image Gallery
- Publications
- Other Links

HOW TO USE SAS

DATA ANALYSIS THREADS
Data reduction examples for (almost) every purpose

USERS GUIDE TO THE XMM-NEWTON SAS
The official XMM-Newton SAS User Guide on-line, PDF version and Postscript version

SAS WATCHOUT PAGE
Issues concerning SAS and data analysis, recommended workarounds/solutions, useful tricks and tips

SAS ON-LINE DOCUMENTATION
Documentation of all single SAS packages

SAS COOKBOOK
An introduction to XMM-Newton data analysis - from NASA XMM-GOF

BACKGROUND ANALYSIS
XMM-Newton pages dedicated to background analysis of all XMM-Newton instruments

ESAS COOKBOOK
Cookbook for data analysis of extended sources using ESAS in SAS, (on-line and PDF) from NASA XMM-GOF. ESAS warnings and watchouts from NASA XMM-GOF.

SAS INVERSE INDEX
The SAS Inverse Index has been designed to provide the list of SAS tasks needed to be executed in order to perform a given scientific analysis job

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:
   - Instrument calibration status reports
   - SAS Science Validation Reports
   - Current Calibration File (CCF) Release Notes

3. Compare your own set of calibration files with the latest available
   - 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 ...
   - Check the SAS "watchout and evergreen" SAS pages, which contain known caveats or bugs
   - Subscribe to the calibration mailing list
   - Install an automatic mirror of the calibration files
   - Make use of the threads, would you like to learn something new

SAS as a web service: RISA

SAS as a web service: RISA

So far, SAS runs locally on user's machine:

- Integration on several different platforms + distribution
- Large maintenance due to need of compatibility with new libraries in new versions — SOC
- SAS download + installation + setup necessary
- Data + Calibration DB download — User

SAS as a web service: RISA

So far, SAS runs locally on user's machine:

- Integration on several different platforms + distribution
- Large maintenance due to need of compatibility with new libraries in new versions — SOC
- SAS download + installation + setup necessary
- Data + Calibration DB download — User

Running SAS through Web Services?

- reduction of maintenance due to the limitation to few platforms (1 ?)
- easy to be fully "frozen" from a certain point in time
- neither SAS installation nor data download needs by single user
- automatic access to large H/W and S/W resources (ESAC Grid + VO tools)
- full data access (processing close to XSA and central CCF repository)

SAS as a web service: RISA

So far, SAS runs locally on user's machine:

- Integration on several different platforms + distribution
- Large maintenance due to need of compatibility with new libraries in new versions — SOC
- SAS download + installation + setup necessary
- Data + Calibration DB download — User

Running SAS through Web Services?

- reduction of maintenance due to the limitation to few platforms (1 ?)
- easy to be fully "frozen" from a certain point in time
- neither SAS installation nor data download needs by single user
- automatic access to large H/W and S/W resources (ESAC Grid + VO tools)
- full data access (processing close to XSA and central CCF repository)

Further Advantages:

- processing in semi-batch mode large amounts of data
- data combination
- complement for archive >> on-the-fly reprocessing
- size-able according to needs - scalability

SAS as a web service: RISA

So far, SAS runs locally on user's machine:

- Integration on several different platforms + distribution
- Large maintenance due to need of compatibility with new libraries in new versions — SOC
- SAS download + installation + setup necessary
- Data + Calibration DB download — User

Running SAS through Web Services?

- reduction of maintenance due to the limitation to few platforms (1 ?)
- easy to be fully "frozen" from a certain point in time
- neither SAS installation nor data download needs by single user
- automatic access to large H/W and S/W resources (ESAC Grid + VO tools)
- full data access (processing close to XSA and central CCF repository)

Further Advantages:

- processing in semi-batch mode large amounts of data
- data combination
- complement for archive >> on-the-fly reprocessing
- size-able according to needs - scalability

Disadvantage:

- no own scripting possible ...
  "only" SAS workflows

SAS as a web service: RISA

So far, SAS runs locally on user's machine:

- Integration on several different platforms + distribution
- Large maintenance due to need of compatibility with new libraries in new versions — SOC
- SAS download + installation + setup necessary
- Data + Calibration DB download — User

Running SAS through Web Services?

- reduction of maintenance due to the limitation to few platforms (1 ?)
- easy to be fully "frozen" from a certain point in time
- neither SAS installation nor data download needs by single user
- automatic access to large H/W and S/W resources (ESAC Grid + VO tools)
- full data access (processing close to XSA and central CCF repository)

Further Advantages:

- processing in semi-batch mode large amounts of data
- data combination
- complement for archive >> on-the-fly reprocessing
- size-able according to needs - scalability

Disadvantage:

- no own scripting possible ...
  "only" SAS workflows

but

... FTOOLS could be added
... other "certified" S/W

SAS as a web service: RISA

So far, SAS runs locally on user's machine:

Integration on several different platforms + distribution — SOC
Large maintenance due to need of compatibility with new libraries in new versions — SOC
SAS download + installation + setup necessary — User
Data + Calibration DB download — User

Running SAS through Web Services?

reduction of maintenance due to the limitation to few platforms (1 ?)
easy to be fully "frozen" from a certain point in time
neither SAS installation nor data download needs by single user
automatic access to large H/W and S/W resources (ESAC Grid + VO tools)
full data access (processing close to XSA and central CCF repository)

Further Advantages:

- processing in semi-batch mode large amounts of data
- data combination
- complement for archive >> on-the-fly reprocessing
- size-able according to needs - scalability

Disadvantage:

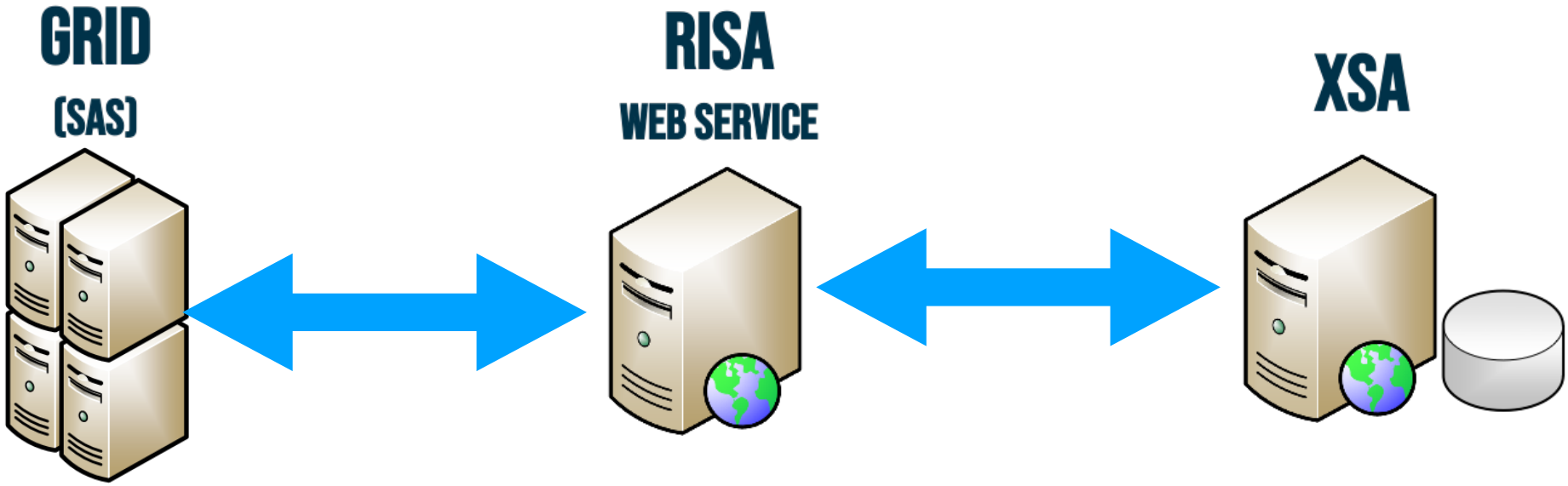
- no own scripting possible ...
  "only" SAS workflows

but

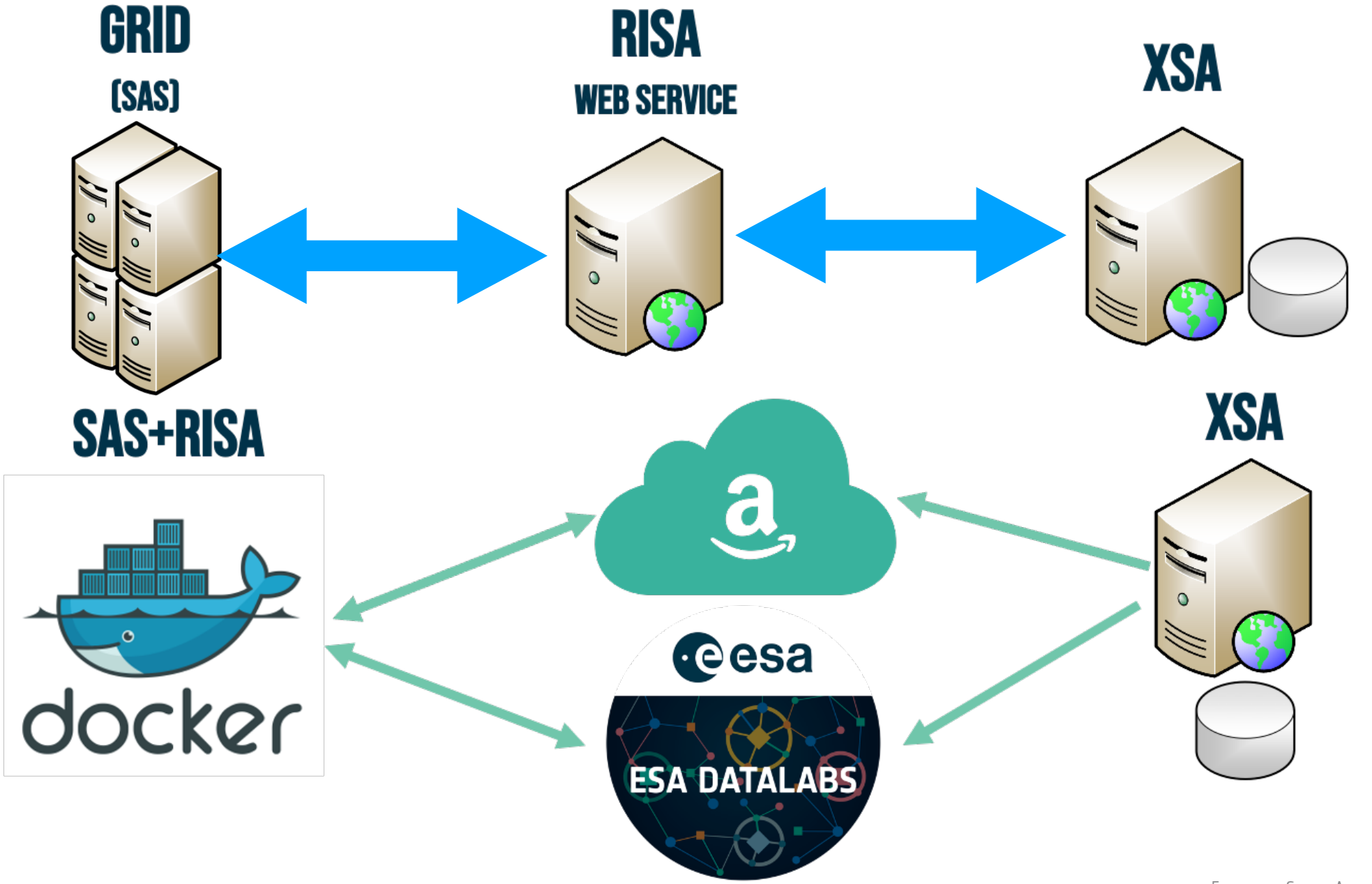
... FTOOLS could be added
... other "certified" S/W

>> complemented with VM (based on same OS)
+ longer cycles / possibility of freezing for long periods
+ allowing for larger control of analysis, scripting + mixing with other tools (IDL, etc)

RISA evolution to AWS

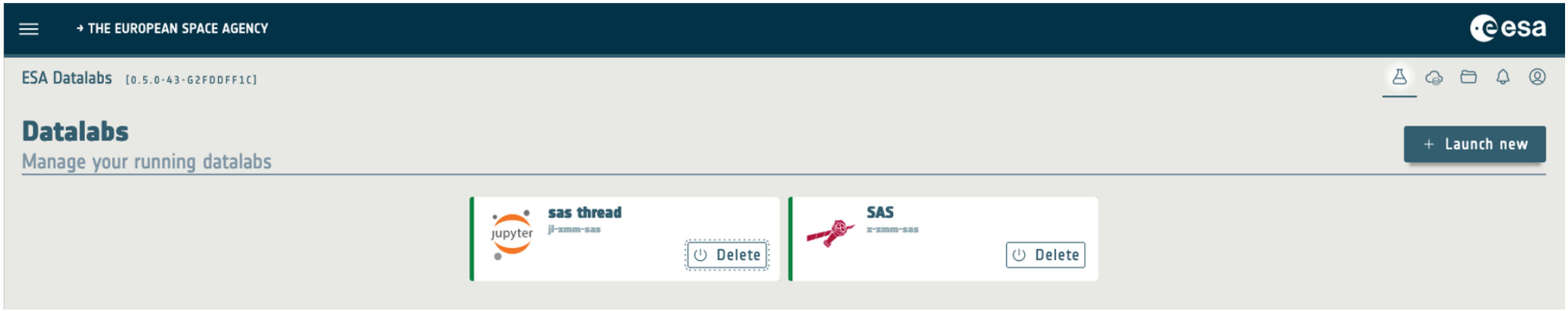


RISA evolution to AWS

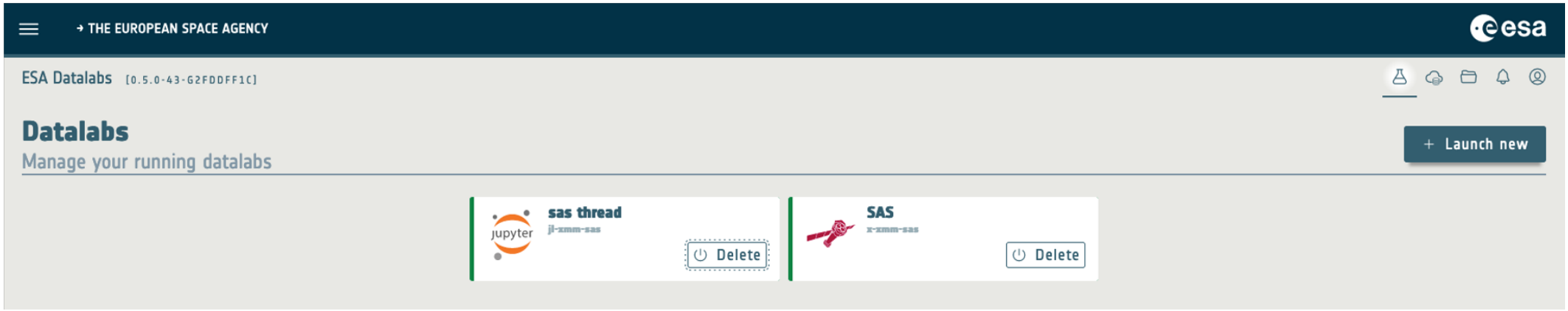


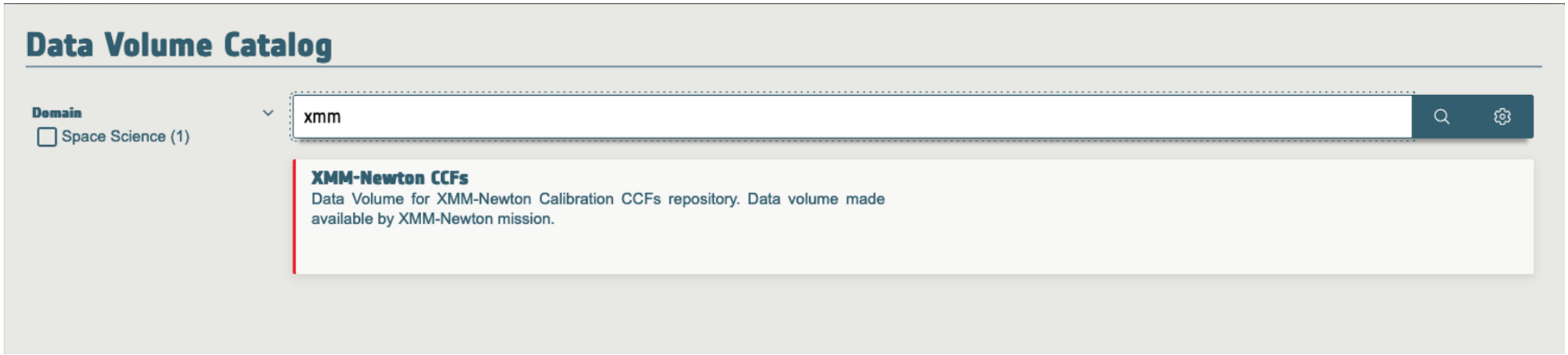
Shaping the SAS future: DataLabs

Shaping the SAS future: DataLabs

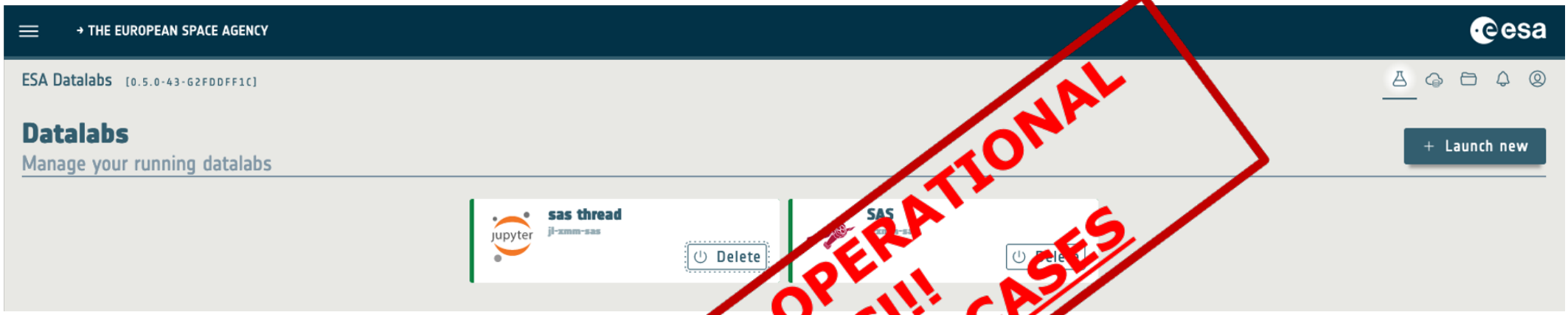


Shaping the SAS future: DataLabs





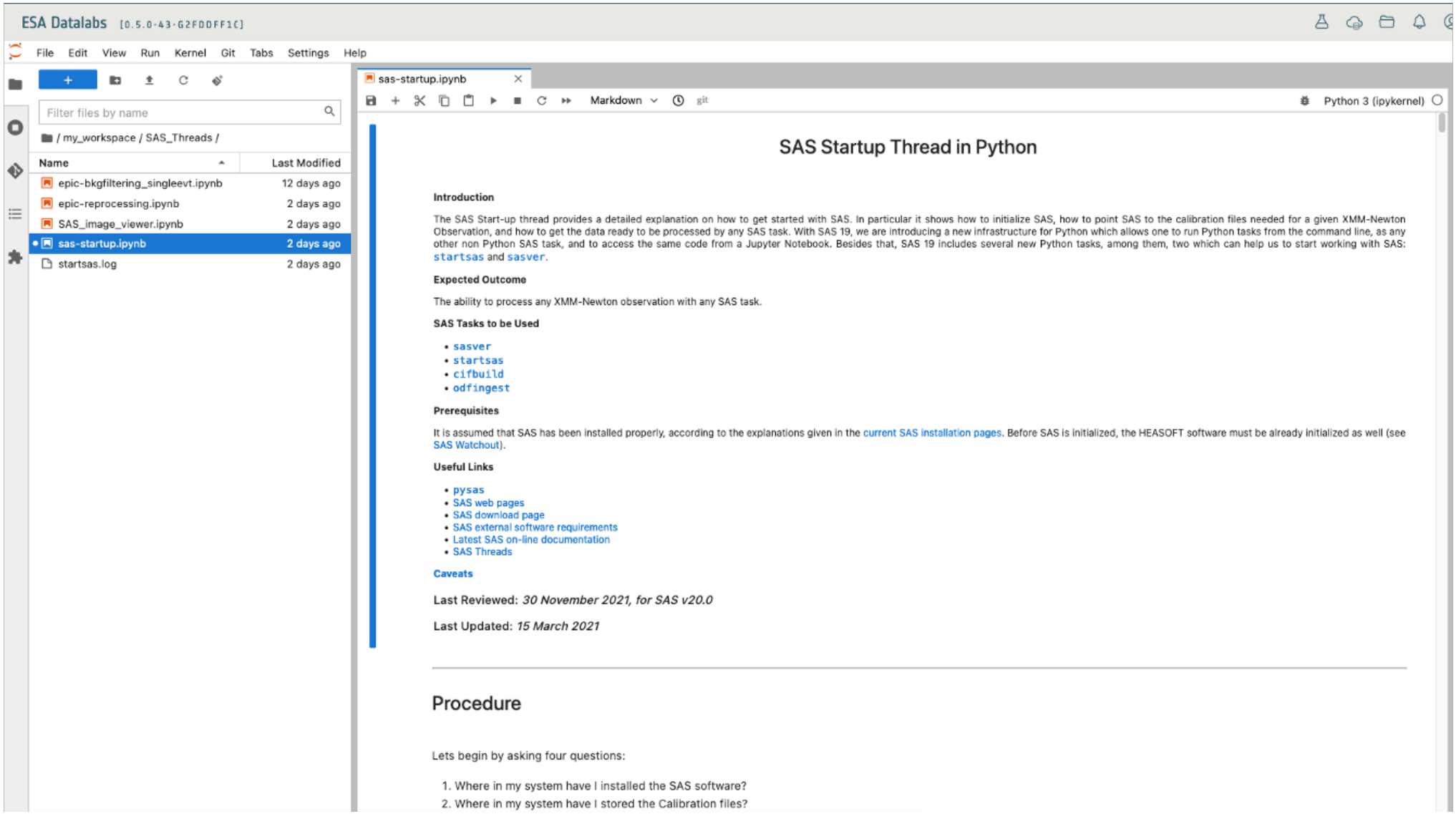
Shaping the SAS future: DataLabs



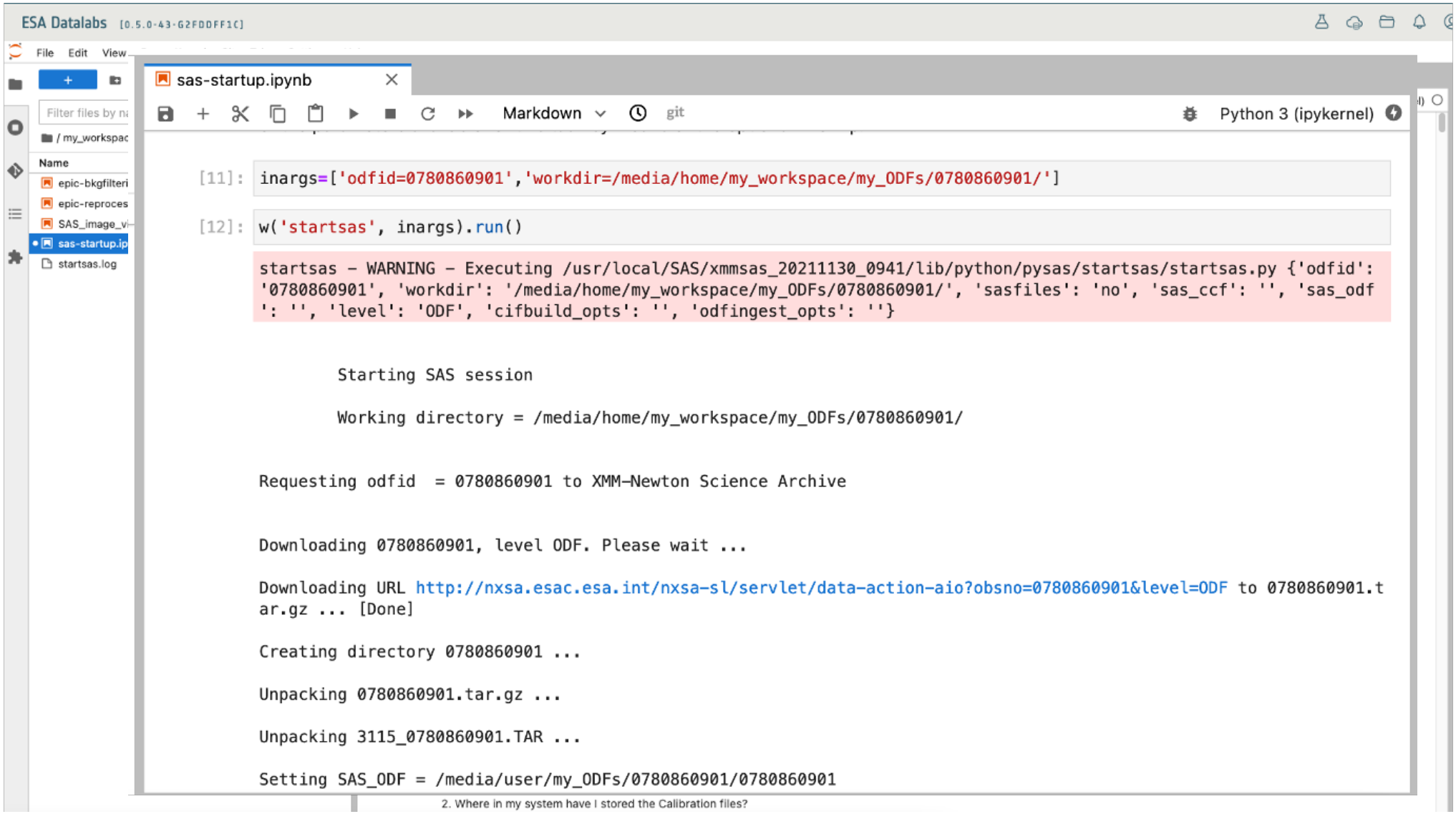
NOT YET IN THE OPERATIONAL DATALABS!!! WORKING ON USE CASES

Shaping the SAS future: DataLabs

Shaping the SAS future: DataLabs

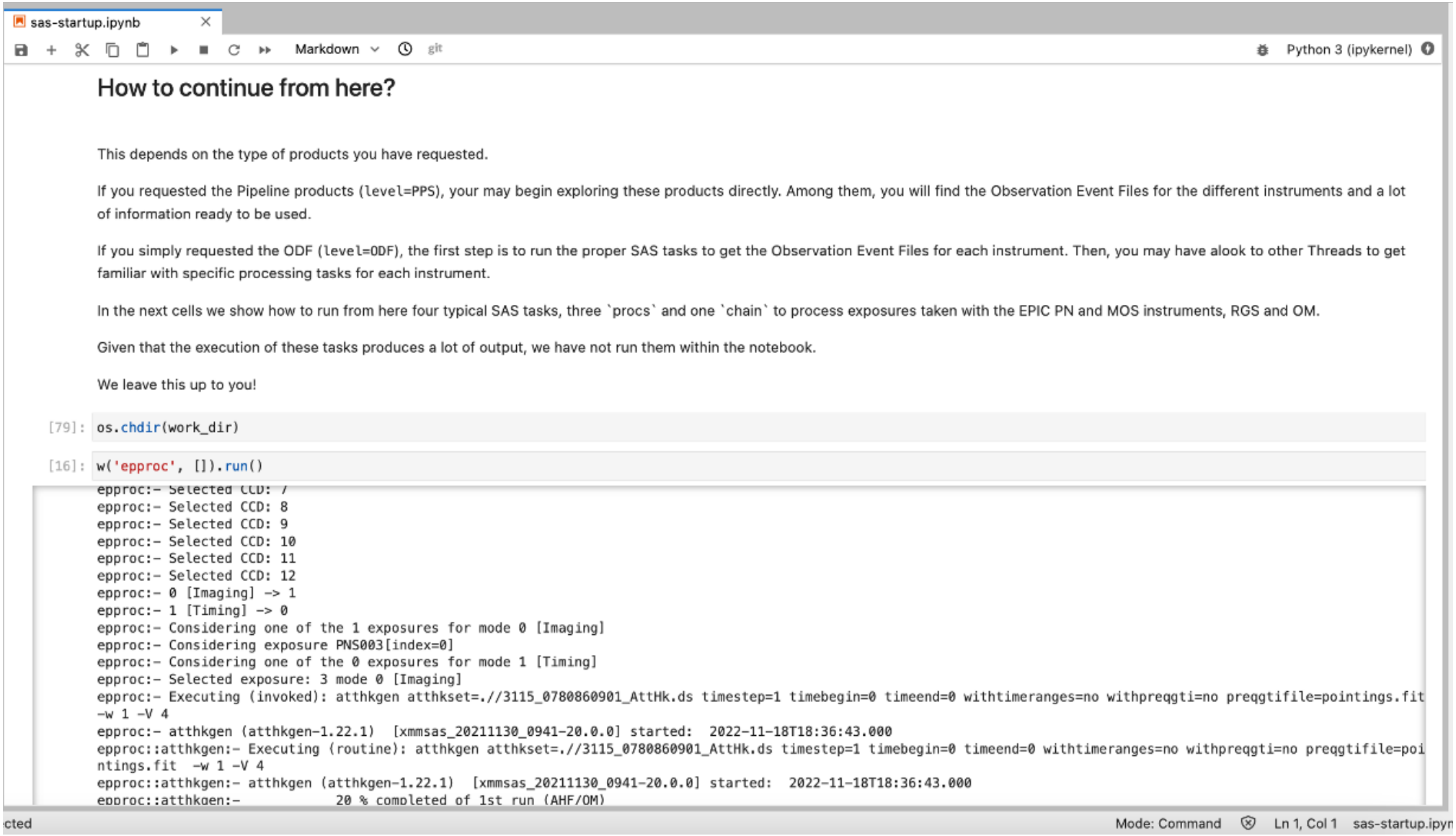


Shaping the SAS future: DataLabs



Shaping the SAS future: DataLabs

Shaping the SAS future: DataLabs



Shaping the SAS future: DataLabs