The XMM-Newton ABC Guide:

An Introduction to XMM-Newton Data Analysis

NASA/GSFC XMM-Newton Guest Observer Facility

Steve Snowden, Rick Shafer, Randall Smith, Lynne Valencic
Brendan Perry, Michael Arida

With contributions by: Ilana Harrus, Stefan Immler, Martin Still

Version 4.1
for XMM-SAS v8.0

August 2008

Copies of this guide are available in `html`, `postscript` and `pdf` formats.

Contents

List of Tables

Table 1: List of Acronyms

ARF	Ancillary Region File
CAL	Calibration Access Layer
CCD	Charge Coupled Device
CCF	Current Calibration File
CIF	Calibration Index File
EPIC	European Photon Imaging Camera
FITS	Flexible Image Transport System
GO	Guest Observer
GOF	NASA/GSFC Guest Observer Facility
GSFC	Goddard Space Flight Center
GUI	Graphical User Interface
HEASARC	High Energy Astrophysics Science Archive Research Center
HTML	Hyper Text Markup Language
OAL	ODF Access Layer
ODF	Observation Data File
OM	Optical Monitor
PDF	Portable Data Format
PP	Pipeline Processing System
PPS	Pipeline Processing
PV	Performance Validation
RGS	Reflection Grating Spectrometer
RMF	Redistribution Matrix File
SAS	Science Analysis System
SOC	Science Operations Center
SSC	Survey Science Centre
SV	Science Validation
XMM	X-ray Multi-Mirror Mission

Chapter 1

Introduction

The purpose of this *ABC Guide* to *XMM-Newton* data analysis is to provide a simple walk-through of basic data extraction and analysis tasks. Also included is a guide to references and help available to aid in the analysis of the data. We have tried to balance providing enough information to give the user a useful introduction to a variety of analysis tasks with not providing too much information, which would make a guide like this too ponderous to use. As such, there is no intention to replace the SAS Handbook, which should be considered the highest authority for the use of SAS. Therefore this document will not display the full versatility of the SAS tasks, and of SAS itself, but it will hopefully show a path through the forest.

Chapter 2 provides lists of web-based references for the *XMM-Newton* project, help desks, analysis guides, and science and calibration data. Chapter 3 provides a description of the data files provided for observation data sets. Chapter 4 discusses the installation and use of SAS. Chapters 5, 6, and 7 discuss the analysis of EPIC, RGS, and OM data respectively.

This document will continue to evolve. Updated versions will be made available on our web site at: `http://heasarc.gsfc.nasa.gov/docs/xmm/abc/`

1.1 ACKNOWLEDGMENTS

This guide would not have been possible without the help and comments from all people involved in the *XMM-Newton* project. In particular, we would like to thank Giuseppe Vacanti and Julian Osborne whose comments made this a more complete and accurate document.

IMH wishes to thank all the OM calibration team and in particular Antonio Talavera, Matteo Guainazzi and Bing Chen for their help in the preparation of this and other documents related to the OM.

SLS wishes to thank Dave Lumb, Richard Saxton, and Steve Sembay for their helpful insights into EPIC data analysis.

Chapter 2

Useful Information and References

2.1 MAIN WEB SITES

- *XMM-Newton* SOC, fount of all *XMM-Newton* project information:

  http://xmm.esac.esa.int/

- NASA/GSFC GOF, source of US specific information and a mirror site for software and public data access:

  http://xmm.gsfc.nasa.gov/

- Survey Science Centre

  http://xmmssc-www.star.le.ac.uk/

2.2 *XMM-NEWTON* HELP DESKS

- The main project helpdesk is located at Vilspa and can be accessed through the WWW:

  http://xmm.esac.esa.int/external/xmm_user_support/helpdesk.shtml

  or via e-mail:

  xmmhelp@sciops.esa.int

  The helpdesk also provides an archive of previously asked questions.

- The NASA/GSFC GOF offers an e-mail helpdesk for both general support and for US-specific issues:

  xmmhelp@lists.nasa.gov

  Some questions addressed to the NASA/GSFC GOF may be redirected to the Vilspa helpdesk.

2.3 MISSION PLANNING AND SPACECRAFT STATUS

- Observation Log:

  http://xmm2.esac.esa.int/external/xmm_mission_plan/index.php

  The scheduling information from this data base has been extracted and incorporated into a Browse data base at GSFC:

  http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl

- Long-Term Timeline:

  http://xmm.esac.esa.int/external/xmm_sched/advance_plan.shtml

2.4 PUBLIC DATA ARCHIVES

- SOC Public Data Archive via the XSA:

  http://xmm.esac.esa.int/xsa/

- GSFC Archive Mirror Site via Browse:

  http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl

2.5 CALIBRATION DATA

- *XMM-Newton* Calibration Page. Under this page can be found the Current Calibration File (CCF) archive, release notes for CCF updates, EPIC response and background files (top menu), and calibration information.

  http://xmm.esac.esa.int/external/xmm_calibration/

- Caldb, NASA/GSFC GOF mirror site for canned response files:

  ftp://legacy.gsfc.nasa.gov/caldb/data/xmm/

2.6 SOFTWARE

- *XMM-Newton* Standard Analysis System (SAS):

  http://xmm.esac.esa.int/external/xmm_data_analysis/

- HEASARC HEASoft Package:

  http://heasarc.gsfc.nasa.gov/docs/corp/software.html

- CXC CIAO Package:

  http://asc.harvard.edu/ciao/

2.7 ANALYSIS, DOCUMENTATION AND HELPFUL HINTS

- On-Line SAS Handbook:

  http://xmm.esac.esa.int/external/xmm_data_analysis/

  click "SAS User's Guide"

- There is a "watchout" page for current SAS bugs at:

  http://xmm.esac.esa.int/sas/8.0.0/watchout/

- *XMM-Newton* Users Handbook:

  http://xmm.esac.esa.int/external/xmm_user_support/documentation/uhb_frame.shtml

- This Guide:

  http://heasarc.gsfc.nasa.gov/docs/xmm/abc/

- The MPE Analysis Guide:

  http://www.mpe.mpg.de/xray/wave/xmm/cookbook/

- The Birmingham Analysis Guide (scripts etc. for EPIC extended source analysis):

  http://www.sr.bham.ac.uk/xmm2/

Chapter 3

Data

3.1 USEFUL DOCUMENTATION

There are a number of documents which the users of *XMM-Newton* data should be aware of. These documents include the *SSC Products Specification*, *Data Files Handbook*, *Reading Data Products CD's* (the most recent versions of these documents can be found in the SOC Document section under `http://xmm.esac.esa.int/external/xmm_user_support/documentation/uhb_frame.shtml`), and the *SAS Users Guide* (`http://xmm.esac.esa.int/external/xmm_user_support/documentation/sas_usg/USG/USG.html`).

For observation data sets going to US PIs, the GOF makes the data available online after PGP encryption and after converting the file names to upper case. When the proprietary period for the observation expires the data are decrypted leaving the file names unchanged. A simple decryption script, minus the relevant keys of course, can be found at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/decrypt.pl`.
NOTE: Laura Brenneman wrote a script and accompanying help file that gives explicit directions on how to most quickly pull over all the files in a data set from the archive, as well as decrypting, and uncompressing the files in preparation for data analysis. This package can be found at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/prepare_xmm_data.tar.gz`. and contains the following files: `README`, `decrpyt.pl`, and `prepare_xmm_data.pl`.

3.2 THE DATA

One of the first steps that should be taken when examining your data is to check to see what you actually have. *XMM-Newton* observations can be broken into several exposures which are each assigned separate observation numbers. These separate exposures can be radically different in length and can also have the different instruments in different modes. For example, in one case the full observation was 60 ks with EPIC and RGS active but there was one delivered exposure which was ∼ 3 ks and had only RGS active. (This can happen because the RGS can operate farther into regions of higher radiation than the EPIC detector. The additional observation time can be considered an additional exposure with only the RGS active.) Two files are useful for this examination. First, the primary HTML page is `INDEX.HTM` which is included in the Pipeline Products. This page lists basic information for the observation plus the operational modes, filters, and exposure start and stop times for the individual instruments. It also has links to various summary pages, including those for the instruments. (In the case above, the EPIC summary page simply stated that "EPIC exposures processed by PPS None.") Specifically, **LOOK** at the `P*SUMMAR0000.HTM` files in the pipeline products (easily available through the links). Second, to quickly access images from the various instruments, examine the PPSGRA Pipeline Products page, viewable with a web browser. These files have the nomenclature P*PPSGRA000_0.HTM. (see § 3.3.3).

3.3 PI Data

Proprietary *XMM-Newton* data is available for download via your XSA account. Email instructions from the XMM-SOC at Vilspa are sent to the address on record with detailed directions on how to retrieve your data via the XSA.

The data files can be considered to come in two groups in separate subdirectories when retrieved, the Observation Data Files (ODF) files and Pipeline Processing (PPS) files. The ODF data contain all of the observation-specific data necessary for reprocessing the observation. The PPS data contain, among other things, calibrated photon event files and source lists.

For observation data sets going to US PIs, the GSFC GOF makes the data available in two directories containing the following groups of files.

- ODF – The ODF (raw) data files
- PPS – The pipeline processed data products

3.3.1 ODF Data

ODF data come with file names in the following format:

- `mmmm_iiiiiijjkk_aabeeeccfff.zzz`

`mmmm` – revolution orbit number

`iiiiii` – proposal number

`jj` – target ID number in proposal

`kk` – exposure number for target

(**NOTE:** The ten-digit combination of `iiiiiijjkk` is the **observation number** and is used repetitively throughout the file nomenclature.)

`aa` – detector (`M1` – MOS1, `M2` – MOS2, `PN` – PN, `OM` – OM, `R1` – RGS1, `R2` – RGS2, `SC` – spacecraft)

`b` – flag for scheduled (`S`) or unscheduled (`U`) observations, or general purpose `X` files

`eee` – exposure number

`cc` – CCD number or OM window number

`fff` – data identifier for the three detectors or spacecraft itself; see Table 3.1.

`zzz` – format (`FIT` - FITS, `ASC` - ASCII)

NOTE: For SAS processing, the file names should contain all upper case characters. However, at least with early CDs, the file names used lower case characters. The GSFC *XMM-Newton* GOF provides a script to rename the files.

3.3.2 Pipeline Product Data – Summary Files and Groupings

The pipeline processing produces quite a number of useful products which allow a first look at the data, but can overwhelm the user by their sheer numbers. The first place to look is the `INDEX.HTM` page which organizes the presentation of the data and provides links to other PP pages. The `INDEX.HTM` page also lists general observation information (target, date, time, etc.) and instrument modes.

The `INDEX.HTM` page provides links to various observation summary pages, which have names with the following nomenclature:

- `PPiiiiiijjkkAAX000SUMMAR0000.HTM`

`iiiiiijjkk` – observation number

`AA` – detector ID (`EP` - EPIC, `OM` - Optical Monitor, `RG` - RGS, `OB` - Observation)

PP data contain some immediately useful data products such as calibrated photon event lists, source lists, and images. While there are a large number of products which come in a single directory, they can be associated in up to 15 groupings; see Table 3.2. (The number of groups can vary depending on the number of operational instruments, e.g., if the OM is turned off there are no OM products.) Further information on each of these groupings and associated files, such as file contents, file types, and how they may be viewed, can be found in Table 3.3. Each group has an associated HTML file which organizes access to the files and provides a limited description of them. The names of the HTML files are of the following form:

- `PPiiiiiijjkkAAAAAA000_0.HTM`

`iiiiiijjkk` – observation number

`AAAAAA` – group identifier (see Table 3.2)

3.3.3 Pipeline Product Data – Data Files

The data file names are of the form (see Table 41 in the *XMM Data Files Handbook*):

- `PiiiiiijjkkaablllCCCCCCnmmm.zzz`

  `iiiiiijjkk` – observation number

  `aa` – detector, `M1` - MOS1, `M2` - MOS2, `PN` - PN, `CA` - for files from the CRSCOR group, `R1` - RGS1, `R2` - RGS2, `OM` - OM.

  `b` – flag for scheduled (`S`) or unscheduled (`U`) observations, or `X` for files from the CRSCOR group (and any product that is not due to a single exposure)

  `lll` – exposure number

  `CCCCCC` – file identification for data from each detector; see Table 3.3

  `n` – For EPIC data, this is the exposure map band number; for RGS data, this is the spectral order number; for the OM, this is the OM window within the exposure.

  `mmm` – source number in hexidecimal

  `zzz` – file type (e.g., PDF, PNG, FTZ, HTM)

  `ASC` - ASCII file, use a web browser, or the “more” command

  `ASZ` - gzipped ASCII file

  `FTZ` - gzipped FITS format, use *ds9, Ximage, Xselect, fv*

  `HTM` - HTML file, use Firefox or other web browser

  `PDF` - Portable Data Format, use *Acrobat Reader*

  `PNG` - Portable Networks Graphics file, use a web browser

  `TAR` - TAR file

Table 3.1: ODF data file identifiers.

Data ID	Contents
	EPIC files
IME	Event list for individual CCDs, imaging mode
RIE	Event list for individual CCDs, reduced imaging mode
CTE	Event list for individual CCDs, compressed timing mode
TIE	Event list for individual CCDs, timing mode
BUE	Event list for individual CCDs, burst mode
AUX	Auxiliary file
CCX	Counting cycle report (auxiliary file)
HBH	HBR buffer size, non-periodic housekeeping
HCH	HBR configuration, non-periodic housekeeping
HTH	HBR threshold values, non-periodic housekeeping
PEH	Periodic housekeeping
PTH	Bright pixel table, non-periodic housekeeping
DLI	Discarded lines data
PAH	Additional periodic housekeeping
PMH	Main periodic housekeeping
	RGS files
AUX	on-board processing statistics
SPE	raw event list for one CCD
DII	diagnostic images
D1H	CCD readout settings
D2H	CCD readout settings
PFH	housekeeping data
ODX	pixel offset data
	XMM files
ATS	spacecraft attitude history
	OM files
IMI	imaging file
THX	tracking history file
WDX	window data auxiliary file
NPH	non-periodic housekeeping file
PEH	periodic housekeeping file
PAX	field acquisition data
RFX	priority reference frame data
PFX	priority fast mode data
FAE	event list (if fast mode was used)

Table 3.2: Pipeline Processing groupings.

Group ID	Contents
	PP files
PPSDAT	Contains the Calibration Index File (CIF) used in the pipeline processing (*CALIND*), PPS information, and the attitude history time series (*ATTTSR*) in gzipped FITS or ASCII format.
PPSGRA	Contains the OM tracking history plots, PPS, EPIC, OM, RGS observation, and PPS run summaries. **NOTE: CHECK THESE OUT**
PPSMSG	ASCII file containing pipeline processing report
	EPIC files
CRSCOR	Contains PDF files of POSS II finding charts, HTML files of cross correlations with the SIMBAD data base, FITS tables for the detected sources
EANCIL	Contains the exposure maps in a variety of energy bands and the source-detection sensitivity maps for the EPIC instruments. The sensitivities are in units of counts s^{-1} corrected for vignetting and corresponding to a likelihood specified in the FITS header. The files are gzipped with a .FTZ extension.
EEVLIS	Contains calibrated photon event files for the EPIC detectors. If the files are sufficiently large they may be separated into two tar files. The files are gzipped fits files with a .FTZ extension.
ESKYIM	This group contains the event images in a variety of energy bands. The fits files are gzipped with a .FTZ extension, the full images also come as PNG images.
ESRLIS	Contains EPIC observation source lists. There is an HTML page of the merged source list and gzipped fits tables of source lists from the different instruments and source detection tasks.
	OM files
OIMAGE	Contains OM sky images in gzipped FITS format.
OMSLIS	Contains OM observation source lists in gzipped FITS format.
OMSRTS	Contains OM star tracking time series in gzipped FITS format.
	RGS files
REVLIS	Contains the RGS source and event lists in gzipped FITS format
REXPIM	Contains the RGS exposure maps in gzipped FITS format
RIMAGE	Contains the RGS images (both energy dispersion and cross dispersion) in gzipped FITS and PNG formats
RSPECT	Contains the RGS source and background spectra in gzipped FITS and PDF formats

Table 3.3: Pipeline Processing data files

Group ID	File ID	Contents	File Type	View With
		EPIC files		
CRSCOR	FCHART	Finding chart	PDF	*Acrobat Reader*
	ROSIMG	ROSAT image of region	PDF	*Acrobat Reader*
	SNNNNN[1]	Source cross-correlation results	Zipped FITS	*fv*
	DNNNNN[1]	Catalog descriptions	PDF	*Acrobat Reader*
	FNNNNN[1]	FOV cross-correlation result	Zipped FITS	*fv*
ESKYIM	IMAGE_8	Sky image 0.2 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_1	Sky image 0.2 - 0.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_2	Sky image 0.5 - 2.0 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_3	Sky image 2.0 - 4.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_4	Sky image 4.5 - 7.5 keV	Zipped FITS	*ds9, Ximage, fv*
	IMAGE_5	Sky image 7.5 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
EANCIL	EXPMAP_8	Exposure map 0.2 - 12.0 keV	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
	EXPMAP_1	Exposure map 0.2 - 0.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_2	Exposure map 0.5 - 2.0 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_3	Exposure map 2.0 - 4.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_4	Exposure map 4.5 - 7.5 keV	Zipped FITS	*ds9, Ximage, fv*
	EXPMAP_5	Exposure map 7.5 - 12.0 keV	Zipped FITS	*ds9, Ximage, fv*
EEVLIS[2]	MIEVLI	MOS imaging mode event list	Zipped FITS	*xmmselect, fv, Xselect*
	PIEVLI	PN imaging mode event list	Zipped FITS	*xmmselect, fv, Xselect*
	TIEVLI	PN, MOS timing mode event list	Zipped FITS	*xmmselect, fv, Xselect*
ESRLIS	EBLSLI	Box-local detect source list	Zipped FITS	*fv*
	EBMSLI	Box-map detect source list	Zipped FITS	*fv*
	EMSRLI	Max-like detect source list	Zipped FITS	*fv*
	OBSMLI	Summary source list	Zipped FITS, HTML	*fv*, web browser
		RGS files		
REVLIS	SRCLI_	RGS Source Lists	Zipped FITS	*fv*
	EVENLI	RGS Event lists	Zipped FITS	*xmmselect, fv*
REXPIM	EXPMAP	RGS Exposure Maps	Zipped FITS	*ds9, Ximage, fv*
RSPECT	SRSPEC1	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	SRSPEC2	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC1	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC2	1st Order Source Spectra	Zipped FITS	*Xspec, fv*
	SRSPEC	Spectra Plots	PDF format	*Acrobat Reader*
RIMAGE	ORDIMG	Images, disp. vs. X-disp	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
	IMAGE_	Images, disp. vs. PI	Zipped FITS, PNG	*ds9, Ximage, fv*, web browser
		OM files		
OIMAGE	SIMAGE	OM Sky Image	Gzipped FITS	*ds9, Ximage, fv*
OMSLIS	SWSRLI	OM Source Lists	Zipped FITS	*fv*
OMSRTS	TSTRTS	Tracing Star Time Series	Zipped FITS	*fv*

[1] NNNNN - Alphanumeric ID

[2] Files for only those modes which were active will be included

Chapter 4

Setting Up and Running SAS

The Science Analysis Software (SAS, `http://xmm.esac.esa.int/sas/`), developed by the Survey Science Centre (SSC) and Science Operations Centre (SOC), is a suite of about 125 programs and scripts that perform data reduction, extraction, and some analysis of *XMM-Newton* data. The Pipeline Processing System (PPS), comprised of a superset of the SAS suite and Perl scripts, is run at Leicester University (`http://xmmssc-www.star.le.ac.uk/`) to create the basic data products provided to the Guest Observer from the satellite ancillary and science data. SAS is not designed for higher level scientific analysis such as spectral fitting and temporal analysis, but does provide for the creation of detector response files and barycentric corrected event timing information. SAS includes extensive EPIC and OM source-detection software. The SAS product files conform to OGIP FITS standards so any high-level analysis package used in high-energy astrophysics should theoretically be capable of processing *XMM-Newton* data. For example, the HEASoft package, `http://heasarc.gsfc.nasa.gov/docs/corp/software.html`, of the High Energy Astrophysics Science Archive Research Center (HEASARC, `http://heasarc.gsfc.nasa.gov/`) at NASA/GSFC and the CIAO package (`http://asc.harvard.edu/ciao/`) of the Chandra X-ray Observatory Center (`http://chandra.harvard.edu/`) can both be used with *XMM-Newton* data files.

4.1 Installation

The primary guide for the installation of SAS can be found through the SOC at `http://xmm2.esac.esa.int/sas/8.0.0/installation/`. Because of the complexity of the SAS installation, it is strongly recommended that users download and install the binary executables rather than compiling SAS from source code (which also necessitates the purchase of commercial software). It should also be noted that "optional" components, while not needed for running SAS tasks from the command-line, are critical to running SAS from the GUI. These optional components are listed at the SOC page `http://xmm2.esac.esa.int/sas/8.0.0/requirements/`.

4.2 Calibration Data

XMM-Newton data reduction and analysis requires extensive calibration data which must be available under a Current Calibration File (CCF) directory. Information on the CCF and instructions for downloading/mirroring the files can be found under the SOC *XMM-Newton* Calibration page (`http://xmm2.esac.esa.int/external/xmm_sw_cal/cal` links to the CCF release notes. In addition, background event files and canned spectral response files can be found under `http://xmm2.esac.esa.int/external/xmm_sw_cal/background/index.shtml`.

4.3 SAS Environmental Parameters and Invocation

There are a few parameters which need to be set for the proper operation of SAS. Two are taken care of by the initialization script, `SAS_DIR` and `SAS_PATH`. These both set the SAS directory path. The remaining parameters, listed below, still need to be set. (The commands should, of course, be modified to be appropriate for your specific setup.)

```
setenv SAS_CCFPATH /full/path/to/CCF       Sets the directory path to the CCF data
setenv SAS_ODF /full/path/to/ODF           Sets the directory path to the ODF data
setenv SAS_CCF /full/path/to/ODF/ccf.cif   Sets the Calibration Index File (CIF) path
                                             and file name
```

Please note that SAS_CCF can also be set after the creation of the ccf.cif file with *cifbuild* (see §4.5.1). Also, while not necessary to run SAS, the following parameters are useful to know about and should be set.

```
setenv SAS_VERBOSITY 3                     Sets the verbosity, 1 => little, 10 => lot
setenv SAS_SUPPRESS_WARNING 3              Sets the warning level, 1 => little, 10 => lot
setenv SAS_IMAGEVIEWER ds9                 Sets the default image viewer; in this case,
                                             it is ds9, but should be set to whatever
                                             the user prefers.)
```

Finally, SAS is invoked by sourcing the script that came with the SAS package:

```
source /full/path/to/xmmsas_20080701_1801/setsas.csh      Initializes SAS
```

or

```
source /full/path/to/xmmsas_20080701_1801/setsas.sh       Alternate SAS initialization
```

SAS can now be used on the command line. To invoke the GUI, type:

```
sas &
```

To verify the SAS-specific settings, type `env | grep SAS`.

It is strongly recommended that users include these environmental settings and make an alias to the initialization script in their login shell file (.cshrc, .bashrc, etc.)! It will save a lot of typing and lower the potential for frustration.

SAS tasks can be run equally well from the command line and from the SAS GUI. In this document we will demonstrate the use of some of the more commonly used tasks from both the GUI and command line, although in some instances, we only give command line examples. In these cases, the GUI can still be used – the user need only set the parameters there.

The MPE Analysis Guide, `http://www.mpe.mpg.de/xray/wave/xmm/cookbook/preparation/change.php` demonstrates many of the common tasks using GUIs.

4.3.1 SAS Helpful Hints

Command lines can often be quite long with a variety of parameters. To avoid considerable typing when creating command scripts a feature of the GUI interface can be of assistance. When invoking a task through the GUI a copy of the full command appears in the dialog box, from where it can then be cut and pasted.

There are several useful features of the command-line interface that users should be aware of. 1) If the `dialog` parameter is included in the command line, the task GUI will pop up with all parameters in the command line preset. This allows the use of the GUI interfaces at the task level without having to go through the main SAS GUI. 2) If the `manpage` parameter is included in the command line, the task documentation will pop up in a web browser window. 3) In addition, the command `sashelp doc=`*sas_task* will pop up a web browser window with the documentation for the task *sas_task* as well.

The command documentation (i.e., the pages brought up by `sashelp doc=`*sas_task* or *sas_task* `manpage`) has an Errors section. Common warning messages produced by the tasks and their meanings are listed here. This feature is *very* useful.

4.4 SAS Syntax and Logic

4.4.1 Command Line Syntax

There is some flexibility in command line syntax in SAS. The following are all valid task calls on the command line that result in identical operations:

```
    rgsproc withsrc=F
    rgsproc withsrc=no
    rgsproc withsrc='no'
    rgsproc withsrc="no"
    rgsproc --withsrc=no
    rgsproc --withsrc='no'
    rgsproc --withsrc="no"
```

However,

```
    rgsproc -withsrc=F
    rgsproc -withsrc=no
    rgsproc -withsrc='no'
    rgsproc -withsrc="no"
```

are not correct syntax.

One format is not "more correct" than another, and the choice of which to use is left to user preference. In this ABC guide we adopt the simplest format, and use no dashes and only single quotation marks only when required, e.g.,

```
    rgsproc withsrc=no orders='1 2 3'
```

where, in this case, the quotes provide the task with a list.

4.4.2 Table Syntax

When a task requires the use of a table within a file there are also several valid syntaxes, e.g.,

```
    xmmselect table=filtered.fits:EVENTS
    xmmselect table="filtered.fits:EVENTS”
    xmmselect table=filtered.fits%EVENTS
```

do an identical operation in opening the `EVENTS` table inside the file `filtered.fits`.

4.4.3 Filtering Logic

Filtering event files requires some command of the SAS logical language which consists of familiar arithmetic and Boolean operators and functions. These, and their syntax, are described within the on-line documentation supplied with the software. Pull up the help document using:

```
    sashelp doc=selectlib
```

4.5 Final Words Before the Meat and Taters

The rest of this guide consists of step-by-step examples of how to reprocess and analyze data for the EPIC (§5), RGS (§6), and OM (§7). At the start of each chapter, it is assumed that the user has just downloaded their data, and has not yet initialized SAS; thus, all chapters begin with the invocation of SAS and the setting of some environment parameters, as listed in 4.3. **Before running any of the tasks in the following chapters, make sure that the ODF file names are all upper case!**

Two tasks, *cifbuild* and *odfingest*, are necessary if the user wants to re-pipeline the data, regardless of which instrument they came from. Given their centrality to SAS reprocessing, it is worth knowing a little more about them and their output files; this is discussed further below. Examples of how to use *cifbuild* and *odfingest* are given in §5.1, §6.1, and §7.2, for the EPIC, RGS, and OM instruments, respectively.

4.5.1 cifbuild

Many SAS tasks require calibration information from the Calibration Access Layer (CAL). Relevant files are accessed from the set of Current Calibration File (CCF) data using a CCF Index File (CIF). A CIF is included in the pipeline products but if the CCF has been updated it can be recreated by the user. In practice, it is perhaps easiest to determine whether the CCF has been updated by recreating the CIF using the SAS task *cifbuild* (default output name `ccf.cif`) and then using the SAS task *cifdiff* to compare the new CIF with the old. If the CAL has changed the user may want to reprocess the data using the new CIF (e.g., see § 5.1). To help determine whether it is reasonable to reprocess the data, the CCF release notes (`http://xmm.esac.esa.int/external/xmm_calibration/`) should be examined.

CCF files can be downloaded directly from the SOC web site (see § 4.2)

The CIF file contains a list of files to be used in the calibration/processing of your data. The task *cifbuild* looks at the CCF directory and builds the CIF file accordingly. If the data are processed with two different CIF files (e.g., because they were generated at different times, with different files under the CCF directory) you can end up with *different* results (although most often not significantly different). Note that the pipeline product *CALIND* is the CIF file used for the pipeline processing.

4.5.2 odfingest

The task *odfingest* extends the Observation Data File (ODF) summary file with data extracted from the instrument housekeeping data files and the calibration database. It is required for reprocessing the ODF data with the pipeline tasks as well as for many other tasks.

Chapter 5

An EPIC Data Processing and Analysis Primer

So, you've received an *XMM-Newton* EPIC data set. What are you going to do with it? After checking what the observation consists of (see § 3.2), you should note when the observation was taken. If it is a recent observation, it was likely processed with the most recent calibrations and SAS, and you can immediately start to analyze the Pipeline Processed data. However, if it is more than a year old, it was probably processed with older versions of CCF and SAS prior to archiving, and the pipeline should be rerun to generate event files with the latest calibrations.

As noted in Chapter 4, a variety of analysis packages can be used for the following steps. However, as the SAS was designed for the basic reduction and analysis of *XMM-Newton* data (extraction of spatial, spectral, and temporal data), it will be used here for demonstration purposes (although see § 5.6 for a short tutorial on the use of *Xselect* for data extraction). SAS will be required at any rate for the production of detector response files (RMFs and ARFs) and other observatory-specific requirements. (Although for the simple case of on-axis point sources the canned response files provided by the SOC can be used.)

NOTE: For PN observations with very bright sources, out-of-time events can provide a serious contamination of the image. Out-of-time events occur because the read-out period for the CCDs can be up to ∼ 6.3% of the frame time. Since events that occur during the read-out period can't be distinguished from others events, they are included in the event files but have invalid locations. For observations with bright sources, this can cause bright stripes in the image along the CCD read-out direction. For a more detailed description of this issue, check: `http://wave.xray.mpe.mpg.de/xmm/cookbook/EPIC_PN/ootevents.html`

It is **strongly** recommended that you keep all reprocessed data in its own directory! SAS places output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipleline Processed data are assumed to be in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, the analysis is taking place in the PROC directory, and the CCF data are in the directory CCF.

If your data are recent and do not require reprocessing, you need only to gunzip the files and rename the event files; for simplicity, it is also recommended that you change the name of the unzipped event file to something easy to type. For example, an MOS1 event list:

```
gunzip ODF/*.gz
gunzip PPS/*.gz
mv PPS/PiiiiiijjkkM1SlllMIEVLI0000.FTZ PPS/PiiiiiijjkkM1SlllMIEVLI0000.FIT.gz
gunzip PPS/PiiiiiijjkkM1SlllMIEVLI0000.FIT.gz PPS/mos1.fits
```

where

`iiiiiijjkk` – observation number
`lll` – exposure number within the observation

At this point, feel free to skip §5.1 and proceed to §5.2, where various analysis procedures are demonstrated using the Lockman Hole SV1 dataset (Obs ID 0123700101). Also, the following procedures are applicable to all XMM-Newton datasets, so it is not required that you use this particular dataset; any observation in the archive should be sufficient.

For detailed descriptions of PP data nomenclature, file contents, and which tasks can be used to view them, see Tables 3.2 and 3.3. For detailed descriptions of ODF data nomenclature and file contents, see Table 3.1.

If you are viewing this online, please note that throughout this primer, images are presented as thumbnails. Clicking on them will open a new tab with a larger image.

5.0.1 A Quick Look at What You Have

The ESKYIM files contain EPIC sky images in different energy bands whose ranges are listed in Table 3.3. While the zipped FITS files may need to be unzipped before display in *ds9* (depending on the version of *ds9*), they can be displayed when zipped using *fv* (*fv* is FITS file viewer available in the HEASoft package). In addition, the image of the total band pass for all three EPIC detectors is also provided in PNG format which can be displayed with a web browser.

The PP source list is provided in both zipped FITS format (readable by *fv*) and as an HTML file.

5.1 Rerunning the Pipeline

The Lockman Hole observation (Obs ID 0123700101) is used throughout this primer to demonstrate various SAS tasks. Unless otherwise stated, it is assumed that your working directory is PROC.

If a dataset is more than about a year old (and our sample set most certainly is), you should generate up-to-date event lists. This is easily done following these steps:

1) Gunzip the all gzipped files in the ODF and PPS directories. If necessary, rename all files in the ODF directory to upper case. This can be done using the script provided by the NASA/GSFC *XMM-Newton* GOF.

```
gunzip ODF/*.gz
gunzip PPS/*.gz
```

2) Initialize SAS by calling one of the two setup scripts, depending on which shell you use.

```
source /full/path/to/xmmsas_20080701_1801/setsas.csh
```

or

```
. /full/path/to/xmmsas_20080701_1801/setsas.sh
```

It is **strongly** recommended that you add a line to your login shell file to set up an alias to these scripts! Calling the script will deal with most of the details needed to run SAS, except for three environment variables, which we will set next.

3) Set the SAS directory pointers. To verify the SAS-specific settings, use the command `env | grep SAS`. (For a detailed discussion of SAS initialization, see Chapter 4.)

```
setenv SAS_ODF /full/path/to/ODF
setenv SAS_CCFPATH /full/path/to/CCF
```

At this point, the SAS GUI can be run by typing `sas &` in the window where the pointers were set. However, since the next few procedures are very simple, it is faster to use the command line.

4) If it doesn't already exist, create a CIF file in the ODF directory using the SAS task *cifbuild* (§4.5.1). If a CIF file has previously been produced, it is only necessary to rerun *cifbuild* if the CCF has changed. Be sure to set the environment parameter again.

```
cd ODF
cifbuild
setenv SAS_CCF /full/path/to/ODF/ccf.cif
```

For `SAS_CCF`, be sure to include the filename `ccf.cif`.

5) If it hasn't already been done (don't do it twice), while still in the ODF directory, prepare the data by using the SAS task *odfingest* (see §4.5.2) and setting the environment parameter. It is only necessary to run it once on any data set, and will cause problems if it is run a second time. If for some reason *odfingest* must be rerun, you must first delete the earlier file produced by odfingest (`*SUM.SAS`).

```
odfingest
setenv SAS_ODF /full/path/to/file/full_name_of_*SUM.SAS
```

6) In your "processing directory" PROC, run the SAS tasks *emchain* or *emproc* (to produce calibrated photon event files for the MOS cameras), and *epchain* or *epproc* (to do the same for the PN camera). From the command line of a window where SAS has been initialized, simply enter:

   `emchain` or `emproc`

   and

   `epchain` or `epproc`

   If the dataset has more than one exposure, a specific exposure can be accessed using the `exposure` parameter, e.g.:

   `emchain exposure=`*n*

   where *n* is the exposure number. To create an out-of-time event file for your PN data, add the parameter *withoutoftime* to your *epchain* invocation:

```
epchain withoutoftime=yes
```

By default, none of these tasks keep any intermediate files they generate. *Emchain* and *epchain* maintain the naming convention described in §3.3.3. *Emproc* and *epproc* designate their output event files with "Evts.ds", with "*ImagingEvts.ds", "*TimingEvts.ds", and "*BurstEvts.ds" denoting the imaging mode, timing, and burst mode event lists, respectively. In either case, you may want to name the new files something easy to type. For example, to rename one of the new MOS1 event files output from *emchain*, type

```
mv P0123700101M1S001MIEVLI0000.FIT mos1.fits
```

To rename the one of the new MOS1 imaging event files output from *emproc*, type

```
mv 0070_0123700101_EMOS1_S001_ImagingEvts mos1.fits
```

Once the new event files have been obtained, the analysis techniques described in §5.2.1 and later can be used.

5.2 Examine and Filter the Data

Since the event files are current, we can proceed with some simple analysis demonstrations. As always, data from the Lockman Hole (Obs ID 0123700101) are used, though any source in the archive will do.

The following sections describe the use of SAS tasks using the both the command line and GUI interfaces, except in cases where one of the methods is particularly easy. People new to SAS will likely prefer the GUI, at least at first; however, as they become more familiar with the software and the keywords, they will probably migrate to the command line, which is faster. Assuming that the parameter values for any given task are the same, it does not matter if a task is invoked on the command line or in the GUI; the output files will be identical. The SAS *xmmselect* GUI provides a very simple method for producing and displaying images, spectra, and light curves, and is the recommended method for extracting data unless large numbers of sources are being analyzed.

5.2.1 An Introduction to the SAS GUI and *xmmselect*

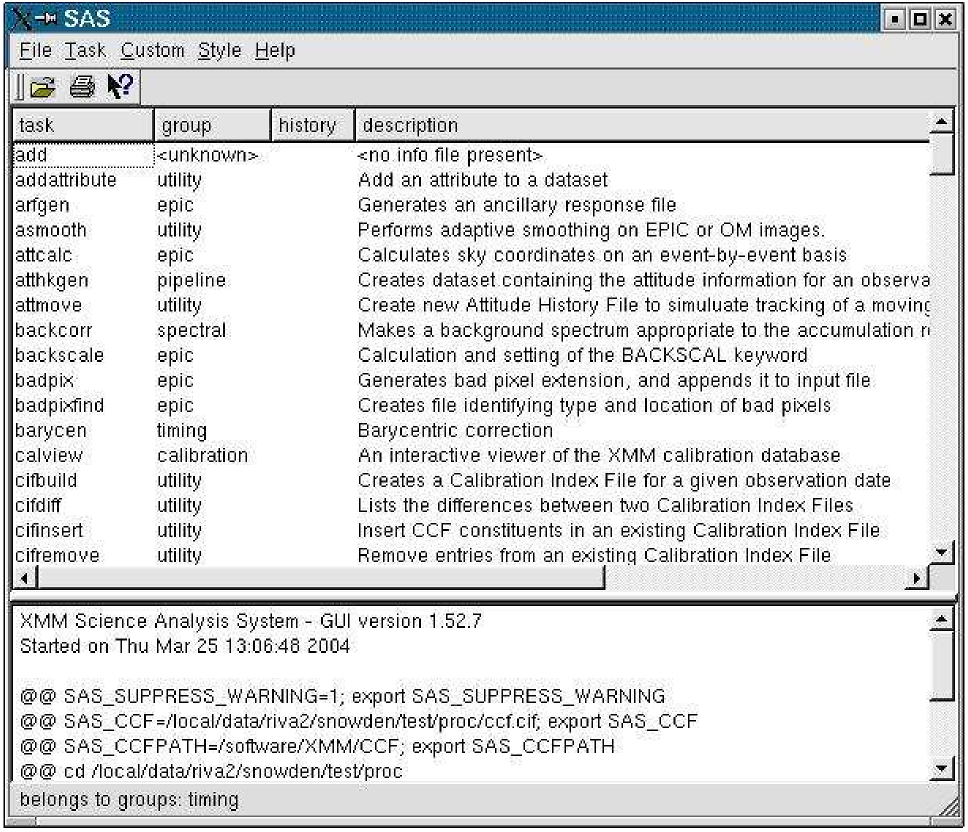
We are now ready to invoke the SAS GUI if we have not already done so (see Figure 5.1). Make sure that you are in the directory where you want the output to go before invoking the SAS GUI or any of the SAS tasks on the command line!

```
sas &
```

1) Invoke the *xmmselect* GUI from the SAS GUI. To invoke a task one need only start typing the task name, and when it is highlighted hit a carriage return.

   - When *xmmselect* is invoked a dialog box will first appear requesting a file name. You can either use the browser button or just type the file name in the entry area, "mos1.fits:EVENTS" in this case. To use the browser, first click on the file folder icon button on the right which will bring up a second GUI for the file selection. Double click on the desired event file in the right-hand column (you may have to open the appropriate directory first), click on the "EVENTS" extension in the right-hand column (which selects the extension), and then click "Ok". The directory GUI will then disappear and then click "Run" on the selection GUI.
   - When the file name has been submitted the *xmmselect* GUI (see Figure 5.2) will appear, along with a dialog box offering to display the selection expression. The selection expression will include the filtering done to this point on the event file, which for the pipeline processing includes for the most part CCD and GTI selections.

Figure 5.1: The SAS GUI. To locate and invoke a task one need only start typing the task name, and when it is highlighted, press carriage return. Otherwise, double-click on the task name.



task	group	history	description
add	<unknown>		<no info file present>
addattribute	utility		Add an attribute to a dataset
arfgen	epic		Generates an ancillary response file
asmooth	utility		Performs adaptive smoothing on EPIC or OM images.
attcalc	epic		Calculates sky coordinates on an event-by-event basis
atthkgen	pipeline		Creates dataset containing the attitude information for an observa
attmove	utility		Create new Attitude History File to simuluate tracking of a moving
backcorr	spectral		Makes a background spectrum appropriate to the accumulation r
backscale	epic		Calculation and setting of the BACKSCAL keyword
badpix	epic		Generates bad pixel extension, and appends it to input file
badpixfind	epic		Creates file identifying type and location of bad pixels
barycen	timing		Barycentric correction
calview	calibration		An interactive viewer of the XMM calibration database
cifbuild	utility		Creates a Calibration Index File for a given observation date
cifdiff	utility		Lists the differences between two Calibration Index Files
cifinsert	utility		Insert CCF constituents in an existing Calibration Index File
cifremove	utility		Remove entries from an existing Calibration Index File

```
XMM Science Analysis System - GUI version 1.52.7
Started on Thu Mar 25 13:06:48 2004

@@ SAS_SUPPRESS_WARNING=1; export SAS_SUPPRESS_WARNING
@@ SAS_CCF=/local/data/riva2/snowden/test/proc/ccf.cif; export SAS_CCF
@@ SAS_CCFPATH=/software/XMM/CCF; export SAS_CCFPATH
@@ cd /local/data/riva2/snowden/test/proc
```

belongs to groups: timing

5.2.2 Create and Display an Image

To create an image in sky coordinates by using the *xmmselect* GUI:

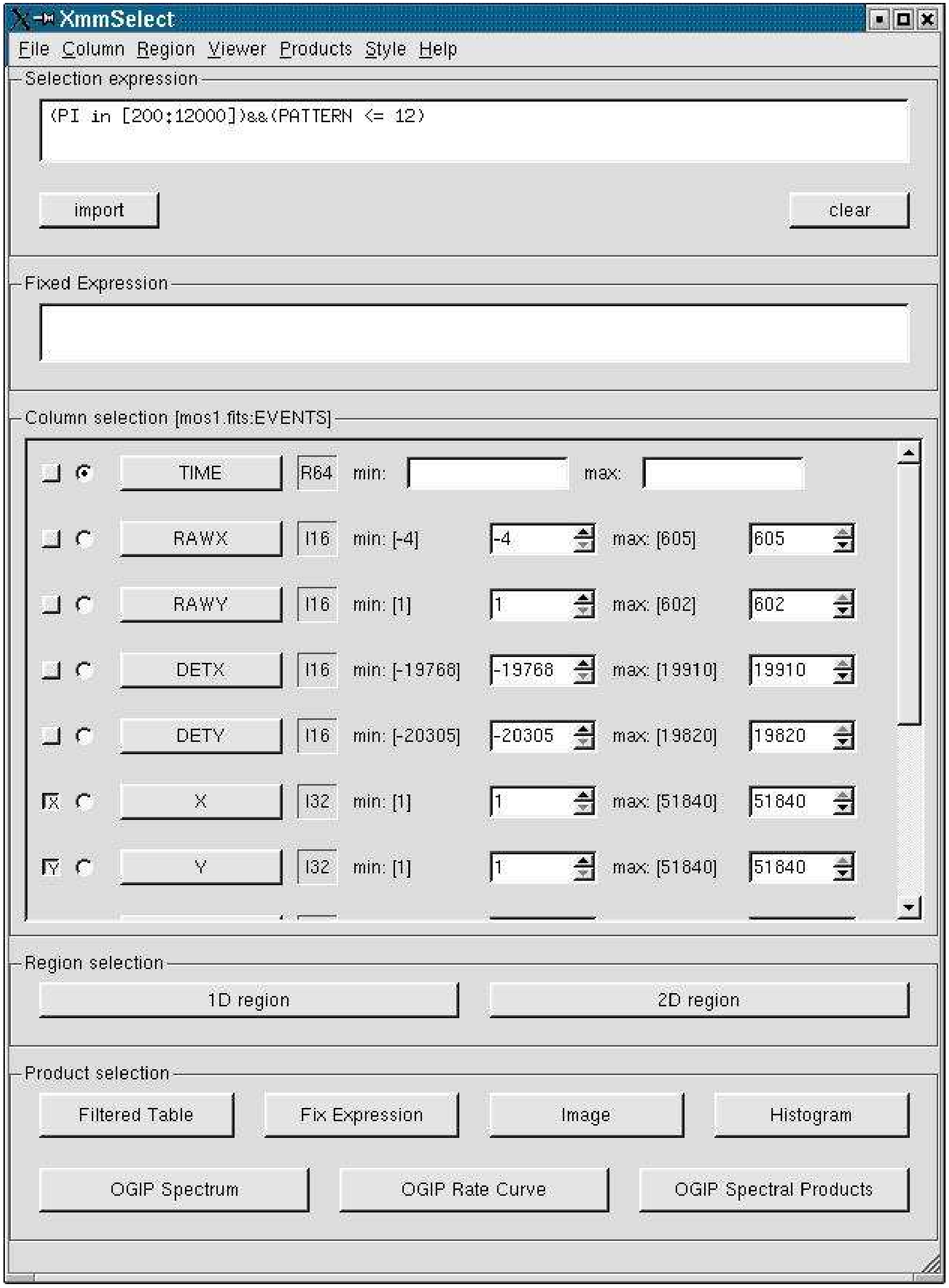
1) Check the square boxes to the left of the "X" and "Y" entries.

2) Click on the "Image" button near the bottom of the page. This brings up the *evselect* GUI (see Figure 5.3).

3) In the `imageset` box, enter the name of the output file, in this case, `image.fits`.

4) Click on the "Run" button on the lower left corner of the *evselect* GUI.

Figure 5.2: The *xmmselect* GUI.



Selection expression: (PI in [200:12000])&&(PATTERN <= 12)

Fixed Expression:

Column selection [mos1.fits:EVENTS]

□	●	TIME	R64	min:		max:	
□	○	RAWX	I16	min: [-4]	-4	max: [605]	605
□	○	RAWY	I16	min: [1]	1	max: [602]	602
□	○	DETX	I16	min: [-19768]	-19768	max: [19910]	19910
□	○	DETY	I16	min: [-20305]	-20305	max: [19820]	19820
☒	○	X	I32	min: [1]	1	max: [51840]	51840
☒	○	Y	I32	min: [1]	1	max: [51840]	51840

- Different binnings and other selections can be invoked by accessing the “Image” tab at the top of the GUI. The default settings are reasonable, however, for a basic image.
- The resultant image is written to the file `image.fits`, and the image is automatically displayed using *ds9*, and is shown in Figure 5.4.

To create an image in sky coordinates by using the task *evselect* on command line:

1) In the window in which SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line).

```
evselect table=mos1.fits:EVENTS withimageset=yes imageset=image.fits
   xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600
```

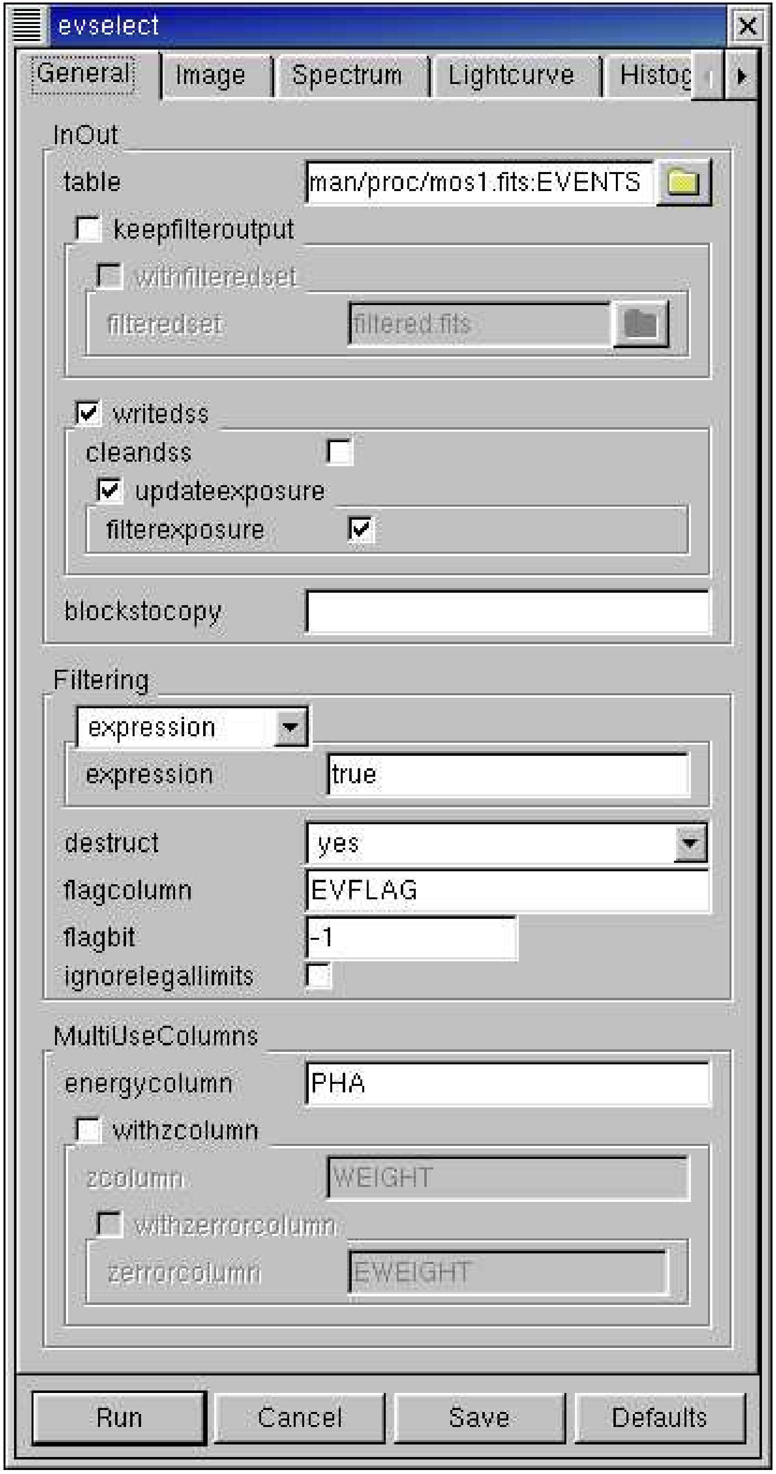
where

`table` – input event table
`withimageset` – make an image
`imageset` – name of output image
`xcolumn` – event column for X axis
`ycolumn` – event column for Y axis
`imagebinning` – form of binning, force entire image into a given size or bin by a specified number of pixels
`ximagesize` – output image pixels in X
`yimagesize` – output image pixels in Y

- The output file `image.fits` can be viewed by using a standard FITS display, such as *ds9* (see Figure 5.4) :

```
ds9 image.fits &
```

Figure 5.3: The *evselect* GUI.



InOut

table: man/proc/mos1.fits:EVENTS

□ keepfilteroutput

□ withfilteredset

filteredset: filtered.fits

☑ writedss

cleandss □

☑ updateexposure

filterexposure ☑

blockstocopy:

Filtering

expression

expression: true

destruct: yes

flagcolumn: EVFLAG

flagbit: -1

ignorelegallimits □

MultiUseColumns

energycolumn: PHA

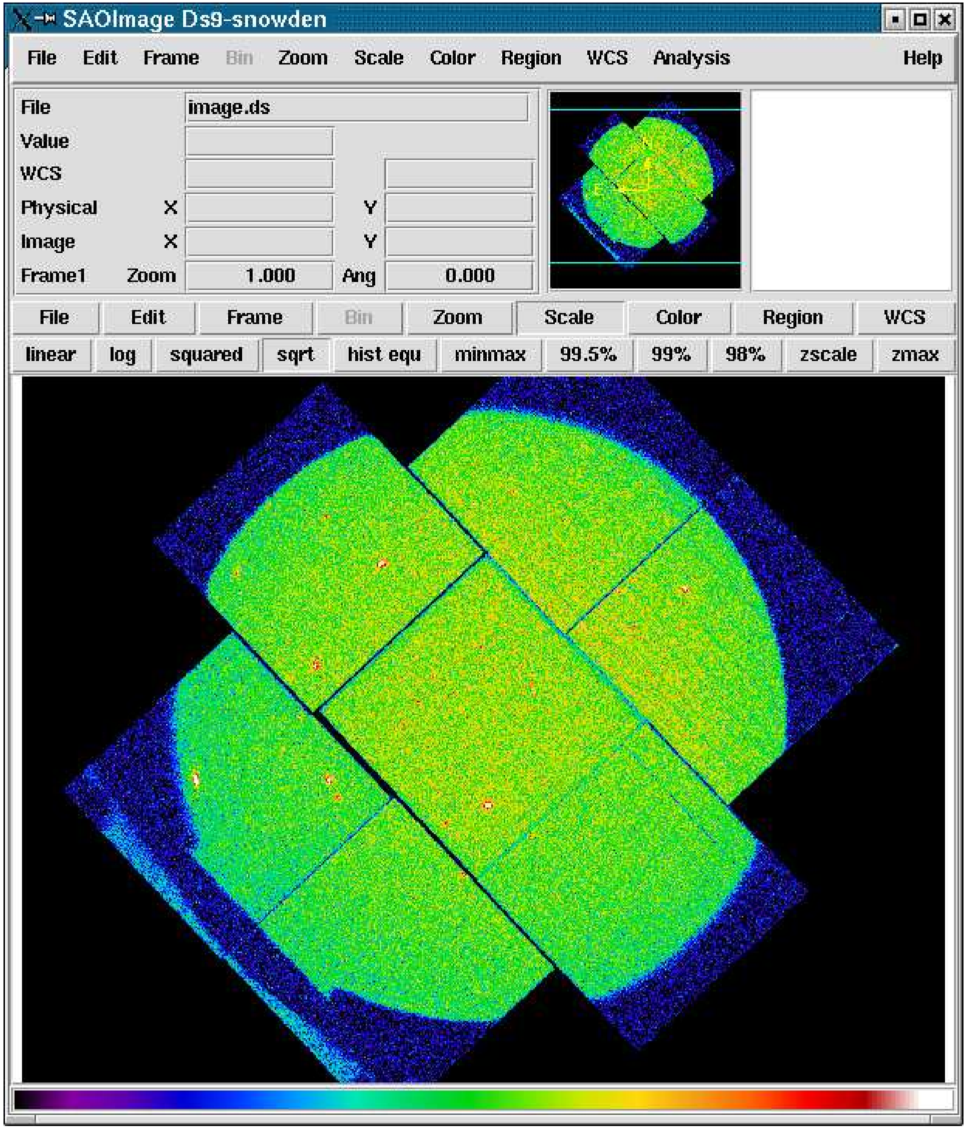
□ withzcolumn

zcolumn: WEIGHT

□ withzerrorcolumn

zerrorcolumn: EWEIGHT

Figure 5.4: The MOS1 image, displayed in *ds9*.



File image.ds

Frame1 Zoom 1.000 Ang 0.000

5.2.3 Create and Display a Light Curve

To create a light curve of the observation by using the *xmmselect* GUI:

1) Check the round box to the left of the "Time" entry.
2) Click on the "OGIP Rate Curve" button near the bottom of the page. This brings up the *evselect* GUI (see Figure 5.3).
3) Click on the "Lightcurve" tab and change the "timebinsize" to a reasonable amount, e.g. 10 or 100 s.
4) Click on the "Run" button at the lower left corner of the *evselect* GUI.
   - The resultant light curve is written to the file *rates.ds*, and is displayed automatically using Grace (see Figure 5.5).

To create a light curve of the observation by using the task *evselect* on the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line).

```
evselect table=mos1.fits:EVENTS withrateset=yes rateset=mos1_ltcrv.fits
   maketimecolumn=yes timecolumn=TIME timebinsize=100 makeratecolumn=yes
```

where

`table` – input event table
`withrateset` – make a light curve
`rateset` – name of output light curve file
`maketimecolumn` – control to create a time column
`timecolumn` – time column label
`timebinsize` – time binning (seconds)
`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

- The output file `mos1_ltcrv.fits` can be viewed by using *dsplot*:

```
dsplot table=mos1_ltcrv.fits x=TIME y=RATE.ERROR withoffsetx=yes &
```

where

`table` – input event table
`x` – column for plotting on X axis
`y` – column for plotting on Y axis
`withoffsetx` – creates an offset to the X axis; -73194570.96472888s in Figure 5.5

5.2.4 Applying Standard Filters the Data

Whether using the GUI or the command line, the needed filtering "expressions" for the MOS and PN are:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM)
```

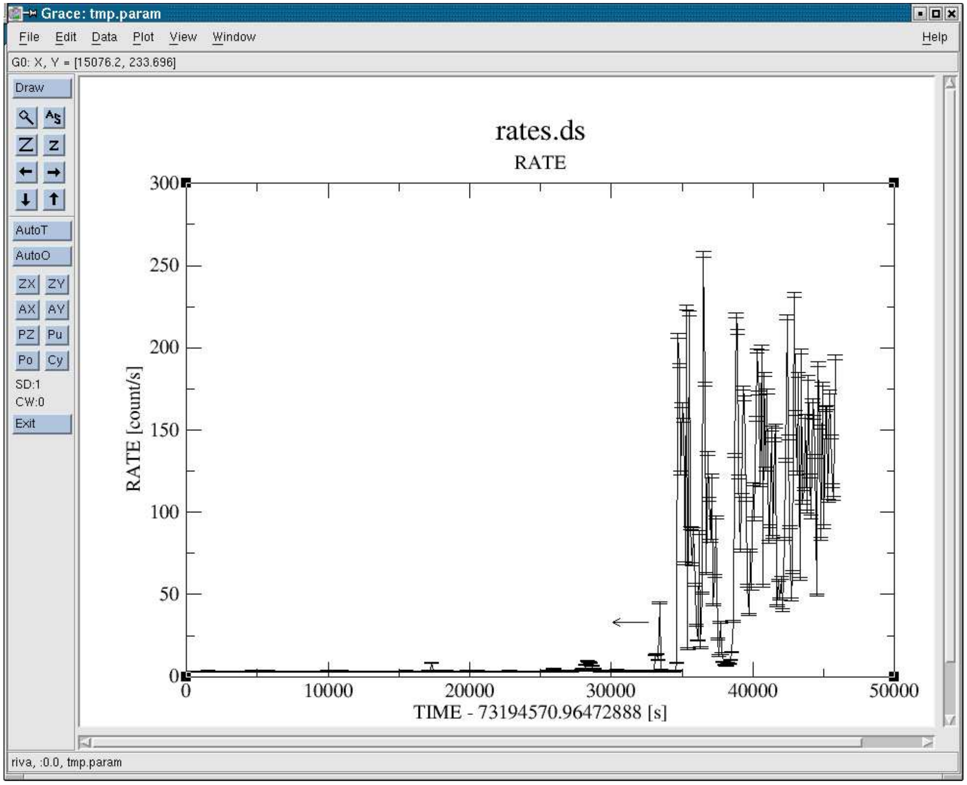
and

```
(PATTERN <= 12)&&(PI in [200:15000])&&#XMMEA_EP)
```

If the PN data is timed, then the `PATTERN` parameter should be set to 4:

```
(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP).
```

Figure 5.5: The light curve, displayed in Grace.



The first two expressions will select good events with `PATTERN` in the 0 to 12 range, and the last will select events with `PATTERN` between 0 and 4. The `PATTERN` value is similar the `GRADE` selection for *ASCA* data, and is related to the number and pattern of the CCD pixels triggered for a given event.The `PATTERN` assignments are: single pixel events: `PATTERN == 0`, double pixel events: `PATTERN in [1:4]`, triple and quadruple events: `PATTERN in [5:12]`.

The second keyword in the expressions, `PI`, selects the preferred pulse height of the event; for the MOS, this should be between 200 and 12000 eV. For the PN, this should be between 200 and 15000 eV. This should clean up the image significantly with most of the rest of the obvious contamination due to low pulse height events. Setting the lower PI channel limit somewhat higher (e.g., to 300 eV) will eliminate much of the rest.

Finally, the `#XMMEA_EM` (`#XMMEA_EP` for the PN) filter provides a canned screening set of FLAG values for the event. (The FLAG value provides a bit encoding of various event conditions, e.g., near hot pixels or outside of the field of view.) Setting `FLAG == 0` in the selection expression provides the most conservative screening criteria and should always be used when serious spectral analysis is to be done.

It is a good idea to keep the output filtered event files and use them in your analyses, as opposed to re-filtering the original file with every task. This will save much time and computer memory. As an example, the Lockman Hole data's original event file is 48.4 Mb; the fully filtered list (that is, filtered spatially, temporally, and spectrally) is only 4.0Mb!

To filter the data using the *xmmselect* GUI:

1) Enter the filtering criteria in the "Selection Expression" area at the top of the *xmmselect* GUI:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM)
```

2) Click on the "Filtered Table" box at the lower left of the *xmmselect* GUI.

3) Change the *evselect* `filteredset` parameter, the output file name, to something useful, e.g., `mos1_filt.fits`

4) Click "Run".

To filter the data using *evselect* on the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line).

```
evselect table=mos1.fits:EVENTS withfilteredset=yes
   expression='(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM'
   filteredset=mos1_filt.fits filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where

`table` – input event table

`filtertype` – method of filtering

`expression` – filtering expression

`withfilteredset` – create a filtered set

`filteredset` – output file name

`keepfilteroutput` – save the filtered output

`updateexposure` – for use with temporal filtering

`filterexposure` – for use with temporal filtering

5.2.5 Applying Temporal Filters the Data

Sometimes, it is necessary to use a temporal filter in addition to those mentioned above. This is because of soft proton background flaring, which can haee count rates of 100 counts/sec or higher.

It should be noted that the amount of flaring that needs to be removed depends in part on the object observed; a faint, extended object will be more affected than a very bright X-ray source.

There are two ways to filter temporally: with an explicit reference to the `TIME` or `RATE` parameters in the filtering expression, or by creating a secondary Good Time Interval (GTI) file with the tast *tabgtigen*. Both procedures are described below. For the example data, we will filter by time, though you can just as easily filter by rate.

To explicitly define the `TIME` or `RATE` parameters:

1) Make a light curve and display it in Grace (see §5.2.3).

2) In the Grace window, pull down the "Edit" menu, select "Regions", and select "Define"; for the Lockman case, select "Left of Line" for the "Region type".

3) Click the "Define" button and then click at two points to create a vertical line at the upper end of the desired range on the Grace plot. (It is possible to define up to five regions at one time by changing the "Define region" counter.)

4) Back on the *xmmselect* GUI, click on the "1D region" button. This will transfer the selection criteria to the "Selection expression" location.

The syntax for the time selection is `(TIME <= 73227600)`. A more complicated expression which would remove a small flare within an otherwise good interval (e.g., the soft proton flares observed in the light curve plot of Figure 5.5) could be: `(TIME <= 73227600)&&!(TIME IN [73221920:73223800])`. The syntax `&&(TIME < 73227600)` includes only events with times less than 73227600, and the "!" symbol stands for the logical "not". So, use `&&!(TIME in [73221920:73223800])` to exclude events in the time interval 73221920 to 73223800.

If combined with the standard filtering expression (see §5.2.3), the full filtering expression would then be:

```
(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM
   &&(TIME <= 73227600) &&!(TIME in [73221920:73223800])
```

This expression can then be used in either the *evselect* GUI or command line to filter the event file, as shown above. If using the GUI, be certain that the `updateexposure` and `filterexposure` boxes are checked. Similarly, if using the command line, make sure to set those parameters to "yes", as in the example above. As ever, give the new file a useful name; here, we will use `mos1_filt_time.fits`.

To make a secondary GTI file using *tabgtigen* and apply it in the GUI:

1) Double-click on *tabgtigen* in the SAS window and select the "quality-filtered" event file, `mos1_filt.fits`, and the extension, `EVENTS`.

2) Edit the output file name, the `gtiset` parameter; here, we will use `gtiset.fits`.

3) Enter the filtering expression. In this case, it is `(TIME <= 73227600) &&! (TIME in [73221920:73223800])`

4) Click "Run" in the lower left corner.

5) Click on *xmmselect* and load the quality-filtered event file, `mos1_filt.fits`.

6) In the "Selection Expression" box, type `GTI(gtiset.fits,TIME)`,

7) Click on the "Filtered Table" box at the lower left of the *xmmselect* GUI.

8) Change the *evselect* `filteredset` parameter, the output file name, to something useful; here, we will use `mos1_filt_time.fits`.

To make a secondary GTI file using *tabgtigen* and apply it on the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following commands (all on one line).

```
tabgtigen table=mos1_ltcrv.fits:RATE gtiset=gtiset.fits timecolumn=TIME
   expression='(TIME <= 73227600)&&!(TIME in [73221920:73223800])'
```

where

`table` – input count rate table and extension (see §5.2.3)
`expression` – filtering expression
`gtiset` – output file name for selected GTI intervals
`timecolumn` – time column

2) Filter the mostly-filtered event file with *evselect*.

```
evselect table=mos1_filt.fits:EVENTS withfilteredset=yes
   expression='GTI(gtiset.fits,TIME)' filteredset=mos1_filt_time.fits
   filtertype=expression keepfilteroutput=yes
   updateexposure=yes filterexposure=yes
```

where

`table` – input count rate table and extension (see §5.2.3)
`expression` – filtering expression
`withfilteredset` – create a filterered set
`filteredset` – output file name
`filtertype` – method of filtering
`keepfilteroutput` – save the filtered set
`updateexposure` – update exposure information in event list and in spectrum files
`filterexposure` – filter exposure extensions of event list with same time filters as for corresponding CCD

5.3 Extract and Fit the Spectrum

5.3.1 Extract the Source Spectrum

Throughout the following, some parameters are instrument-dependent! The parameter `spectralbinsize` should be set to 15 if you are using the MOS; if you are using the PN, it should be 5. The parameter `specchannelmax` should be set to 11999 for the MOS, or 20479 for the PN. Also, remember that the most stringent filter, `FLAG==0`, must be applied to get a high-quality spectrum.

To extract the source spectrum using the GUI:

1) Click on *xmmselect* and run the filtered file, `mos1_filt_time.fits`.
2) In the "Selection Expression" box, enter `FLAG==0`.
3) Make an image (see Section 5.2.2). It will be displayed automatically in a *ds9* window.
4) Click on the object whose spectrum you wish to extract.
   - This will produce a circle (extraction region), centered on the object. The circle's radius can be changed by clicking on it.
5) Adjust the size and position of the circle until you are satisfied with the extraction region.
6) Click on "2D Region" in the *xmmselect* GUI.
   - This transfers the region information into the "Selection Expression" text area, for example, `((X,Y) IN circle(26144,22838,600))`.
   - The `circle` parameters are the X, Y coordinates of the center, and the radius in units of 0.05 arcsec; so, the region used here has a radius of 30 arcsec.
7) Click the round button next the `PI` column on the *xmmselect* GUI.
8) Click on "OGIP Spectrum".
9) In the "General" page, check `keepfilteroutput` and `withfilteredset`.
10) In the `filteredset` box, enter the name of the event file output, in this case, `mos1_filtered.fits`.
11) Select the "Spectrum" page of the *evslect* GUI to set the file name and binning parameters for the spectrum.
12) Confirm that `withspectrumset` is checked.
13) Set `spectrumset` to the desired output name, in this case, `mos1_pi.fits`.
14) Confirm that `spectralbinsize` is set to the correct value: 15 for the MOS, 5 for the PN.
15) Confirm that `withspecranges` is checked.
16) Set `specchannelmin` to 0.
17) Set `specchannelmax` to 11999 for the MOS, or 20479 for the PN.
18) Click "Run".

To extract the source spectrum using the command line:

1) Bring up *ds9* and view the filtered event file:

```
ds9 mos1_filt_time.fits &
```

2) Click on the object whose spectrum you wish to extract.

- This will produce a circle (extraction region), centered on the object. The circle's radius can be changed by clicking on it.

3) Adjust the size and position of the circle until you are satisfied with the extraction region.

4) Double-click on the chosen region.

   - This will bring up a window showing the center coordinates and radius of the circle. For example, lets say the center is at (26144,22838) and the radius is 600.

5) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line). Remember that for PN data, `spectralbinsize` must be set to 5.

```
evselect table='mos1_filt_time.fits:EVENTS' energycolumn='PI' withfilteredset=yes
   filteredset='mos1_filtered.fits' keepfilteroutput=yes filtertype='expression'
   expression='((X,Y) in CIRCLE(26144,22838,600))&&(FLAG==0)'
   withspectrumset=yes spectrumset='mos1_pi.fits' spectralbinsize=15
   withspecranges=yes specchannelmin=0 specchannelmax=11999
```

where

`table` – the event file
`energycolumn` – energy column
`withfilteredset` – make a filtered event file
`keepfilteroutput` – keep the filtered file
`filteredset` – name of output file
`filtertype` – type of filter
`expression` – expression to filter by
`withspectrumset` – make a spectrum
`spectrumset` – name of output spectrum
`spectralbinsize` – size of bin, in eV
`withspecranges` – covering a certain spectral range
`specchannelmin` – minimum of spectral range
`specchannelmax` – maximum of spectral range

5.3.2 Extract the Background Spectrum

To extract the background spectrum using the GUI:

1) Follow steps 1 - 3 of §5.3.1 (GUI instructions).

2) Make an annulus around the object whose background spectrum you wish to extract. This can be done using two circles, each defining the inner and outer edges of the annulus.

3) Select the inner circle. From "Regions" on the *ds9* pulldown menu, select "Properties", then "Exclude".

4) Adjust the sizes and positions of the circles until you are satisfied with the extraction region.

5) In the *xmmselect* GUI, make sure the "Selection Expression" box is empty.

6) Click on "2D Region" in the *xmmselect* GUI.

7) Confirm that the correct values were transferred to the "Selection Expression" box; in this case, it should show

```
((X,Y) in CIRCLE(26144,22838,1500))&&!((X,Y) in CIRCLE(26144,22838,900))
```

8) Follow steps 7 - 18 of §5.3.1 (GUI instructions), giving the output files appropriate names. In this case, let the spectrum be `back_pi.fits` and the filtered file be `bkg_filtered.fits`.

To extract the background spectrum using the command line:

1) View the filtered event file with *ds9*.

```
ds9 mos1_filt_time.fits &
```

2) Make an annulus around the object whose background spectrum you wish to extract. This can be done using two circles, each defining the inner and outer edges of the annulus.

3) Select the inner circle. From "Regions" on the *ds9* pulldown menu, select "Properties", then "Exclude".

4) Adjust the sizes and positions of the circles until you are satisfied with the extraction region.

5) Double-click on the regions.

   - This will bring up windows listing the center coordinates and radius of each circle. For example, lets say the circles are centered on (26144,22838), and have radii of 900 and 1500.

6) Follow step 5 in §5.3.1 (command line instructions), remembering to change the output file names and the extraction expression:

```
spectrumset='bkg_pi.fits'
filteredset='bkg_filtered.fits'
expression='((X,Y) in CIRCLE(26144,22838,1500))&&!((X,Y) in CIRCLE(26144,22838,900))'
```

5.3.3 Check for Pile Up

Depending on how bright the source is and what modes the EPIC detectors are in, event pile up may be a problem. Pile up occurs when a source is so bright that incoming X-rays strike two neighboring pixels or the same pixel in the CCD more than once in a read-out cycle. In such cases the energies of the two events are in effect added together to form one event. If this happens sufficiently often it will skew the spectrum to higher energies. To check whether pile up may be a problem, use the SAS task *epatplot*. Note that this procedure requires as input the event files created when the spectrum was made.

The deafult output of *epatplot* is a postscript file named "`my event file`"`_pat.ps`. As ever, the user should feel free to rename files. The plot, which may be viewed with *gv*, contains two graphs describing the distribution of counts as a function of PI channel; see Figure 5.6.

A few words about interpretting the plots are in order. The top is the distribution of counts versus PI channel for each pattern class (single, double, triple, quadruple), and the bottom is the expected pattern distribution (**smooth lines**) plotted over the observed distribution (**histogram**). If the lower plot shows the model distributions for single and double events diverging significantly from the observed distributions, then the source is piled up.

The source used in this example is too faint to provide reasonable statistics for *epatplot* and is far from being affected by pile up. In contrast, Figure 5.7 shows an example of a bright source (from a different observation) which is strongly affected by pileup. Note the severe divergence between the model and the observed pattern distribution.

To check for pile up with the GUI:

1) Double-click *epatplot* in the SAS GUI.

2) In the *epatplot* window, in the "set" text area, enter the name of the event file that was made when the spectrum was extracted, in this case, `mos1_filtered.fits`.

3) Click the "withbackgroundset" box.

4) In the "backgroundset" text area, enter the name of the background event file output when the background spectrum was extracted, in this case, `bkg_filtered.fits`.

5) If you wish to change the output file name to something other than the default, click the "useplotfile" box and enter the name in the "plotfile" text area; For this example, we will use `mos1_epat.ps`.

5) Click "Run".

To check for pile up from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line).

```
epatplot set=mos1_filtered.fits plotfile=mos1_epat.ps useplotfile=yes
   withbackgroundset=yes backgroundset=bkg_filtered.fits
```

Figure 5.6: The output of *epatplot* for a very faint source without pileup. Note that in the lower plot, for energies less than ∼ 1500 eV, there are too few X-rays for *epatplot* to model.

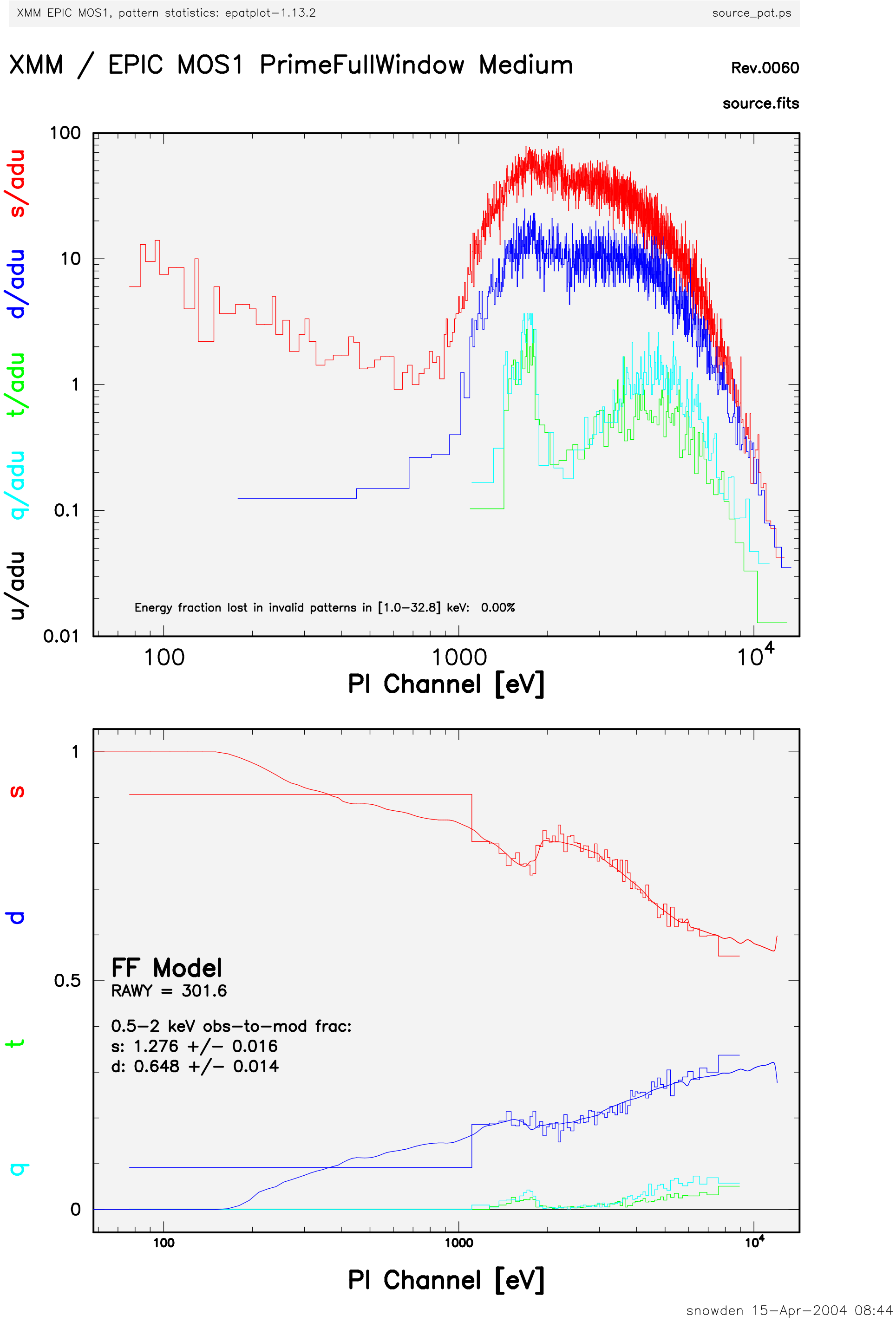
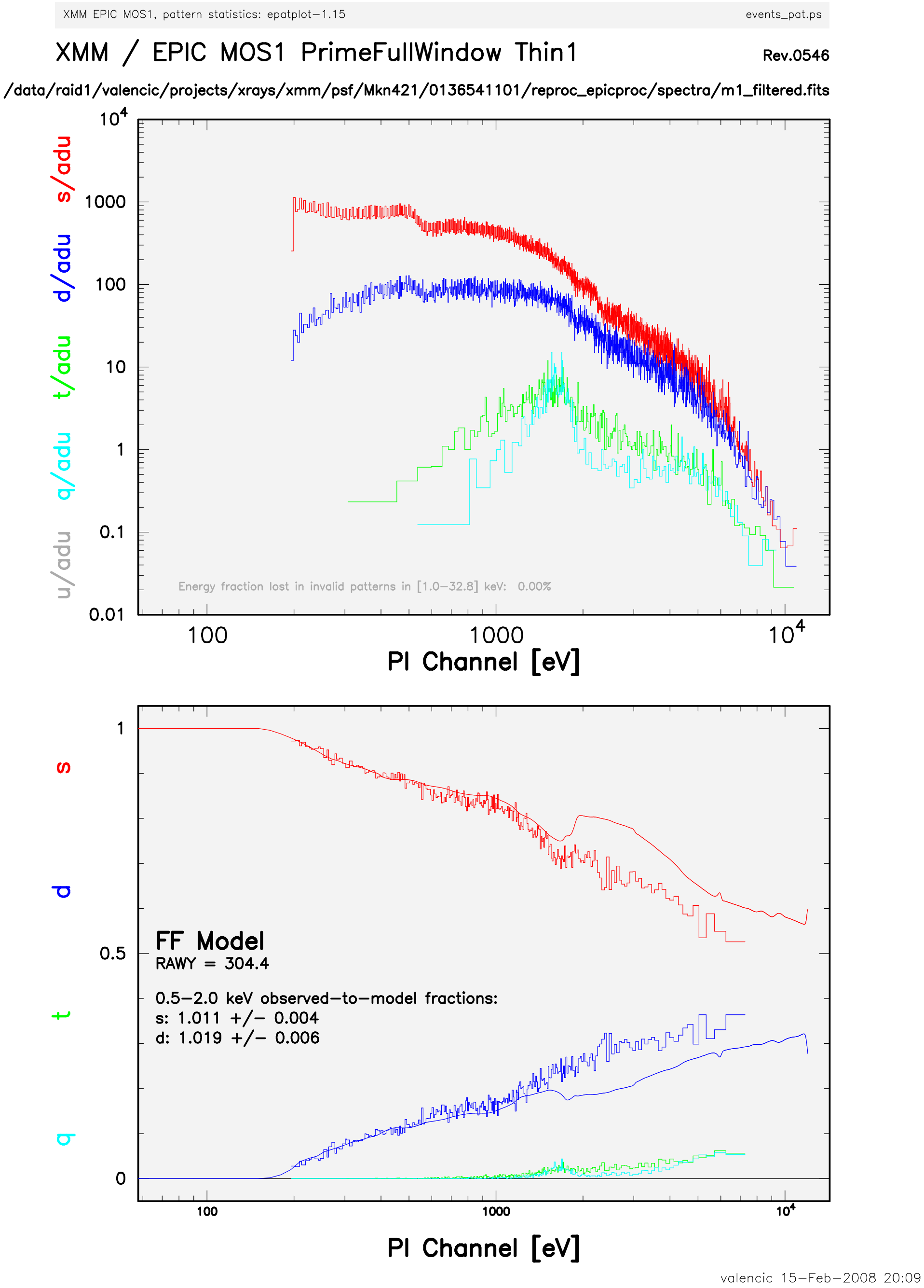


Figure 5.7: The output of *epatplot* for a heavily piled source. In the lower plot, there are large differences between the predicted and observed pattern distribution at energies above ~ 1000 eV.



5.3.4 My Observation is Piled Up! Now What?

There are two ways to remove the effects of pile up from an observation:

- Using the region selection and event file filtering procedures demonstrated in earlier sections, you can excise the inner-most regions of a source (as they are the most heavily piled up), re-extract the spectrum, and continue your analysis on the excised event file. For this procedure, it is recommended that you take an iterative approach: remove an inner region, extract a spectrum, check with *epatplot*, and repeat, each time removing a slightly larger region, until the model and observed distribution functions agree.

and/or

- Using the event file filtering procedures, consider only the pattern 0 events (`PATTERN==0`). Pattern 0 events are less sensitive to pile up than other patterns.

5.3.5 Create the Photon Redistribution Matrix (RMF)

The following assumes that an appropriate source spectrum, named `mos1_pi.fits`, has been extracted as in §5.3.1.

To make the RMF using the GUI:

1) Double-click the task *rmfgen* in the SAS GUI.

2) In the "Main" tab, set the `spectrumset` keyword to the spectrum file name, e.g., `mos1_pi.fits`.

3) Set the `rmfset` keyword to the RMF file name, e.g., `mos1_rmf.fits`.

4) Click "Run".

To make the RMF using the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type:

```
rmfgen rmfset=mos1_rmf.fits spectrumset=mos1_pi.fits
```

where

`rmfset` – output file
`spectrumset` – spectrum file

5.3.6 Create the Ancillary Region File (ARF)

The following assumes that an appropriate source spectrum, named `mos1_pi.fits`, has been extracted as in §5.3.1, and the RMF, named `mos1_rmf.fits`, has been made as in §5.3.5.

To make the ARF using the GUI:

1) Double-click on the task *arfgen* in the SAS GUI.

2) In the "Main" tab, set the `arfset` parameter to the ARF file name, for example, `mos1_arf.fits`.

3) In the "Main" tab, set the `spectrumset` parameter to the spectrum file name, in this case, `mos1_pi.fits`.

4) In the "Effects" tab, confirm that the `withbadpixcorr` box is checked. Set the `badpixlocation` keyword to the event file name from which the spectrum was extracted, in this case, `mos1_filt_time.fits`.

5) In the "Calibration" tab, check the `withrmfset` box and set the `rmfset` keyword to the RMF file name, in this case, `mos1_rmf.fits`.

6) Click on "Run". (If your *xmmselect* GUI is still running, a dialog box will occur asking whether rmfgen can be run. It can, as there is no conflict).

To make the ARF using the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type:

```
arfgen arfset=mos1_arf.fits spectrumset=mos1_pi.fits withrmfset=yes
   rmfset=mos1_rmf.fits badpixlocation=mos1_filt_time.fits
```

where

`arfset` – output ARF file name

`spectrumset` – input spectrum file name

`withrmfset` – flag to use the RMF

`rmfset` – RMF file created by *rmfgen*

`withbadpixcorr` – flag to include the bad pixel correction

`badpixlocation` – file containing the bad pixel information; should be set to the event file from which the spectrum was extracted.

5.3.7 Prepare the Spectrum

Assuming that source and background spectra have been extracted as in §5.3.1 and 5.3.2, and the RMF and ARF created as in §5.3.5 and 5.3.6, spectral fitting will be demonstrated using HEASoft software.

Nearly all spectra will need to be binned for statistical purposes. The procedure *grppha* from FTOOL provides an excellent mechanism to do just that. The following commands not only group the source spectrum for Xspec but also associate the appropriate background and response files for the source.

1) On the command line, type:

```
grppha
```

and edit the parameters and file names as appropriate:

```
Please enter PHA filename[] mos1_pi.fits      ! input spectrum file name
Please enter output filename[] mos1_grp.fits  ! output grouped spectrum
GRPPHA[] chkey BACKFILE back_pi.fits          ! include the background spectrum
GRPPHA[] chkey RESPFILE mos1_rmf.fits         ! include the RMF
GRPPHA[] chkey ANCRFILE mos1_arf.fits         ! include the ARF
GRPPHA[] group min 25                         ! group the data by 25 counts/bin
GRPPHA[] exit
```

5.3.8 Fit the Spectrum

Next, use Xspec to fit the spectrum.

1) On the command line, type:

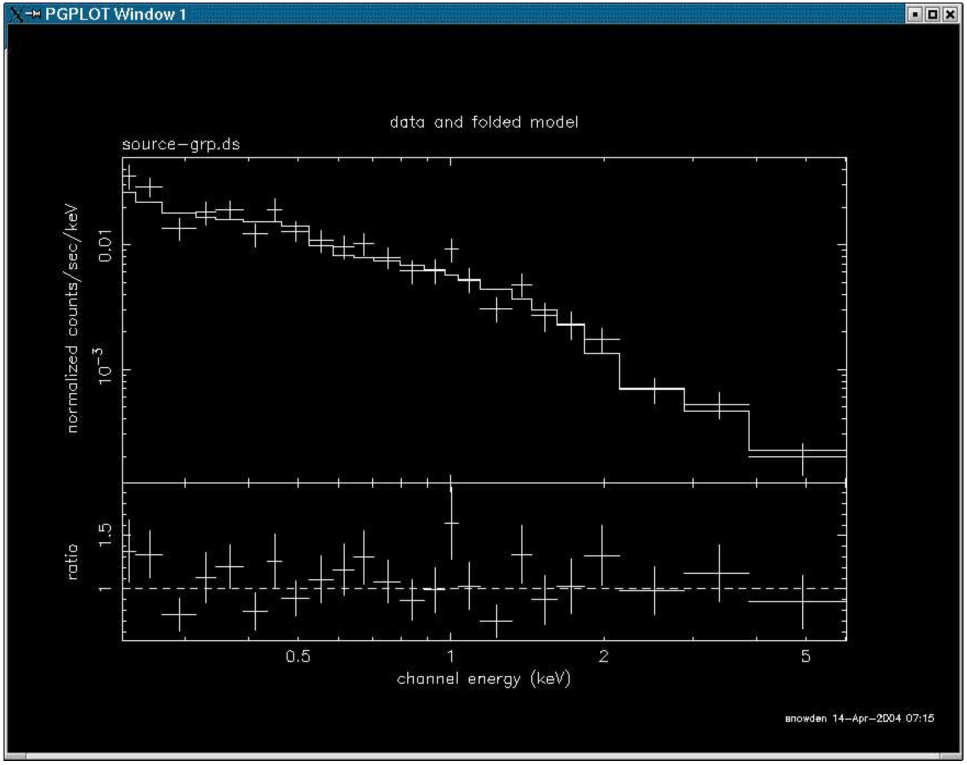
```
xspec
```

and edit the parameters and file names as appropriate:

```
XSPEC> data mos1_grp.fits      ! input data
XSPEC> ignore 0.0-0.2,6.6-**   ! ignore unusable energy ranges, in keV
                               ! set a range appropriate for the data
XSPEC> model wabs(pow+pow)     ! set spectral model to two absorbed power laws
1:wabs:nH> 0.01                ! set model absorption column density to 1.e20
2:powerlaw:PhoIndex> 2.0       ! set the first model power law index to -2.0
3:powerlaw:norm>               ! default model normalization
4:powerlaw:PhoIndex> 1.0       ! set the second model power law index to -1.0
5:powerlaw:norm>               ! default model normalization
wabs:nH> 0.01                  ! set model absorption column density to 1.e20
renorm                         ! renormalize the model spectrum
XSPEC> fit                     ! fit the model to the data
XSPEC> setplot device /xw      ! set the plot device
XSPEC> setplot energy          ! plot energy along the X axis
XSPEC> plot ldata ratio        ! plot two panels with the log of the data and
                               !     the data/model ratio values along the Y axes
XSPEC> exit                    ! exit Xspec
Do you really want to exit? (y) y
```

Figure 5.8 shows the fit to the spectrum.

Figure 5.8: The fitted spectrum of the Lockman Hole source.



5.4 Source Detection

The *edetect_chain* does nearly all the work involved with EPIC source detection, but its required input files must first be generated and prepared using the tasks *atthkgen*, *evselect*, and *emosaic*. Fortunately, these are all quick and straightforward.

In the example below, source detection is done on images in two bands (300 - 2000 eV, and 2000 - 10000 eV) for all three detectors. The example uses the filtered event file produced in §5.2.5, with the assumption that it is located in the current directory.

To detect sources with the GUI:

1) Double-click on the task *atthkgen*.

2) Set the `timestep` keyword to 1.

3) Set the `atthkset` keyword to the desired output file name, for example, `attitude.fits`.

4) Click “Run”.

5) Double-click on the task *evselect*.

6) In the “General” tab, set the “Table” paramter to the event file name (`mos1_filt_time.fits`). Confirm that “Filtertype” is set to “expression”, and in the “Expression” text area, type `(FLAG == 0)&&(PI in [300:2000])`.

7) In the “Image” tab, check the `withimageset` box, and enter the desired output image name, for example, `mos1-s.fits`. Set `xcolumn` to X and `ycolumn` to Y. Set `Binning` to `binSize`, `ximagebinsize` to 50, and `yimagebinsize` to 50.

8) Click “Run”.

9) Repeat steps 6 - 8 for each event file, changing the output image name to something appropriate, for example, `mos2-s.fits` and `pn-s.fits`.

10) Set the "Expression" text to `(FLAG == 0)&&(PI in [2000:10000])` and repeat steps 6 - 9. We will assume the output images are named `mos1-h.fits`, `mos2-h.fits`, and `pn-h.fits`, so that we now have six output images in all.

11) Double-click on the task *emosaic* (this will be used for display purposes later).

12) In the `imagesets` text area, type: `mos1-s.fits mos1-h.fits mos2-s.fits mos2-h.fits pn-s.fits pn.fits`.

13) Set `mosaicedset` to the desired output file name, for example, `mosaic.fits`.

14) Click "Run".

15) Double-click on the the task *edetect_chain*.

16) In the "0" tab, in the `imagesets` text area, type: `mos1-s.fits mos1-h.fits mos2-s.fits mos2-h.fits pn-s.fits pn-h.fit`. In the `eventsets` area, enter the names of the event files: `mos1_filt_time.fits mos2_filt_time.fits pn_filt_time.fits`. In the `attitudeset` area, enter the name of the attitude file made by the task *atthkgen* (`attitude.fits`). Set the `pimin` keyword to the minimum PI values (in eV) for the input images by typing: `300 2000 300 2000 300 2000`, and do similar for the maximum values for `pimax` (`2000 10000 2000 10000 2000 10000`). Set the `likemin` parameter to 10, `witheexpmap` to `yes`, `ecf` to `0.878 0.220 0.878 0.220 3.652 0.632`.

17) In the "1" tab, set `eboxl_list` to `eboxlist_l.fits` and `eboxm_list` to `eboxlist_m.fits`.

18) In the "2" tab, set `esp_withootset` to `yes`, `esp_ooteventset` to `pn-oot-filt-time.fits`, and `eml_list` to `emllist.fits`.

19) Click "Run".

To detect sources from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type:

```
atthkgen atthkset=attitude.fits timestep=1
```

where

`atthkset` – output file name

`timestep` – time step in seconds for attitude file

2) Now make the event files with *evselect*. On one line, type:

```
evselect table=mos1_filt_time.fits withimageset=yes imageset=mos1-s.fits
   imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50
   filtertype=expression expression='(FLAG == 0)&&(PI in [300:2000])'
```

where

`table` – event list
`withimageset` – flag to create an image
`imageset` – fits image name to be created
`imagebinning` – how to bin the image
`xcolumn` – table column to use for the X axis
`ximagebinsize` – binning in X axis
`ycolumn` – table column to use for the Y axis

`yimagebinsize` – binning in Y axis
`filtertype` – type of filtering
`expression` – filtering expression, select events in the PI channel range 300-2000 eV

3) Repeat step 2 for each detector, changing the output file name as needed. We will assume here that they are named `mos2-s.fits` and `pn-s.fits`.

4) Repeat steps 2 and 3, but change the filtering expression to `(FLAG == 0)&& (PI in [2000:10000])`. We will assume the output images are named `mos1-h.fits`, `mos2-h.fits`, and `pn-h.fits`, so that we now have six output images in all.

5) Create a merged count image to display later:

```
emosaic imagesets='mos1-s.fits mos1-h.fits mos2-s.fits mos2-h.fits pn-h.fits pn-s.fits'
   mosaicedset=mosaic.fits
```

where

`imagesets` – list of count images
`mosaicedset` – output image name

6) Run *edetect_chain*.

```
edetect_chain imagesets='mos1-s.fits mos1-h.fits mos2-s.fits
   mos2-h.fits pn-s.fits pn-h.fits' eventsets='mos1_filt_time.fits
   mos2_filt_time.fits pn-filt-time.fits' attitudeset=attitude.fits
   pimin='300 2000 300 2000 300 2000' pimax='2000 10000 2000 10000 2000 10000'
   likemin=10 witheexpmap=yes ecf='0.878 0.220 0.878 0.220 3.652 0.632'
   eboxl_list=eboxlist_l.fits eboxm_list=eboxlist_m.fits
   eml_list=emllist.fits esp_withootset=yes esp_ooteventset=pn-oot-filt-time.fits
```

where

`imagesets` – list of count images
`eventsets` – list of event files
`attitudeset` – attitude file name
`pimin` – list of minimum PI channels for the bands
`pimax` – list of maximum PI channels for the bands
`likemin` – maximum likelihood threshold
`witheexpmap` – create and use exposure maps
`ecf` – energy conversion factors for the bands
`eboxl_list` – output file name for the local sliding box source detection list
`eboxm_list` – output file name for the sliding box source detection in background map mode list
`eml_list` – output file name for maximum likelihood source detection list
`esp_withootset` – Flag to use an out-of-time processed PN event file, useful in cases where bright point sources have left streaks in the PN data
`esp_ooteventset` – The out-of-time processed PN event file

The ecfs are in units of Those used here are derived from PIMMS using the flux in the 0.1-10.0 keV band, a source power-law index of 1.9, an absorption of 0.5×10^{20}.

7) Display the results of *eboxdetect* using the task *srcdisplay*.

```
srcdisplay boxlistset=eboxlist_m.fits imageset=mosaic.fits
   regionfile=regionfile.txt sourceradius=0.01 withregionfile=yes
```

where

> `boxlistset` – *eboxdetect* source list
> `imageset` – image file name over which the source circles are to be plotted
> `includesources` – flag to include the source positions on the display
> `regionfile` – file name of output file containing source regions
> `sourceradius` – radius of circle plotted to locate sources
> `withregionfile` – flag to create a region file

8) Display the results of *emldetect* using the task *implot*, in this case as a GIF file (`pgplot.gif`).

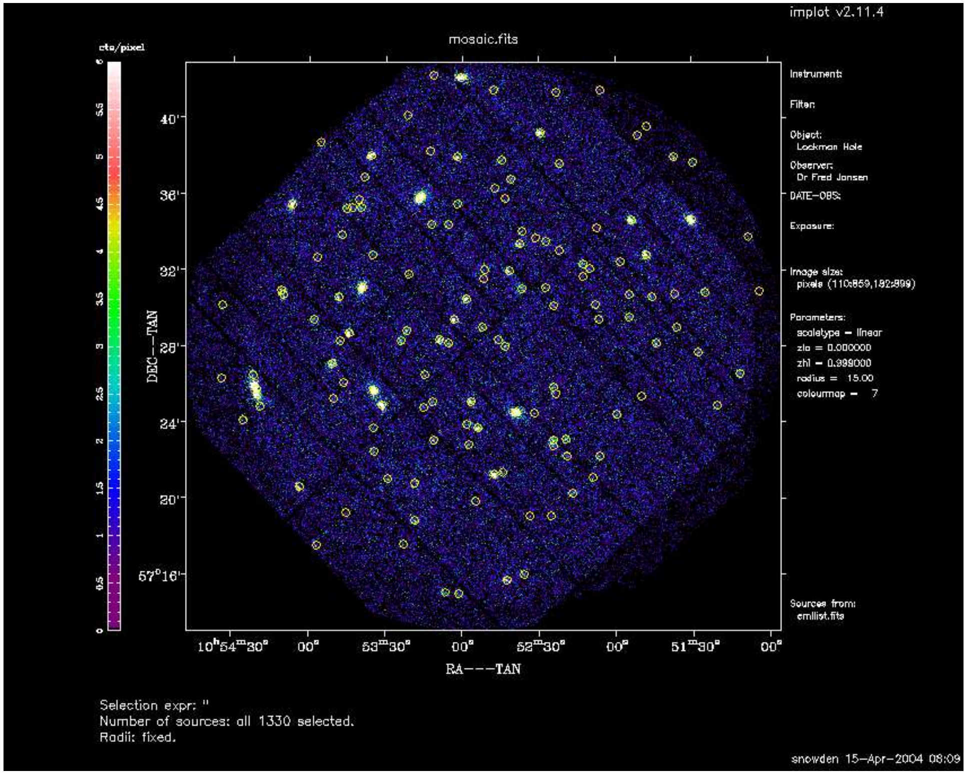
```
implot set=mosaic.fits device=/GIF srclisttab=emllist.fits
```

where

> `set` – input image for the plot
> `device` – type of output (/GIF, /PS, /XW)
> `srclisttab` – source list file name

Figure 5.9 shows the output of *implot* for the maximum likelihood source detection (*emldetect*).

Figure 5.9: EPIC count image with the detected sources from the maximum likelihood task created by *implot*.



```
Instrument:
Filter:
Object:
  Lockman Hole
Observer:
  Dr Fred Jansen
DATE-OBS:
Exposure:
Image size:
  pixels (110:859,192:999)
Parameters:
  scaletype = linear
  zlo = 0.000000
  zhi = 0.999000
  radius =  15.00
  colourmap =   7
Sources from:
  emllist.fits
```

```
Selection expr: ''
Number of sources: all 1330 selected.
Radii: fixed.
```

```
snowden 15-Apr-2004 08:09
```

5.5 Timing Analysis

This section will demonstrate some basic timing analysis of EPIC image-mode data using the Xronos analysis package. (Note: for PN timing and burst mode data, the task *epchain* must be run with `datamode=TIMING|BURST`.) These examples assume that an appropriate light curve, named `source.lc`, has been created as in § 5.2.3 with `timebinsize` set to `1` and `makeratecolumn` set to `no`. For this exercise the central source from the observation of G21.5-09 (Obs ID 0122700101) is used. For the aficionado, the task *barycen* can be used for the barycentric correction of the source event arrival times.

Figure 5.10: Light curve for the source analyzed as in § 5.2.3.

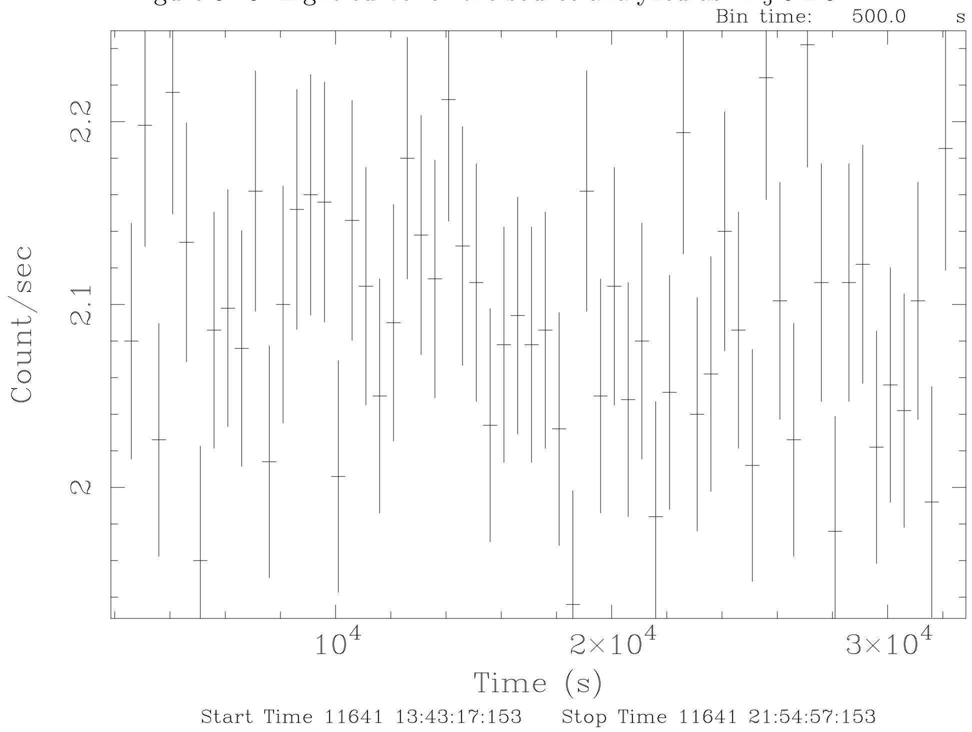
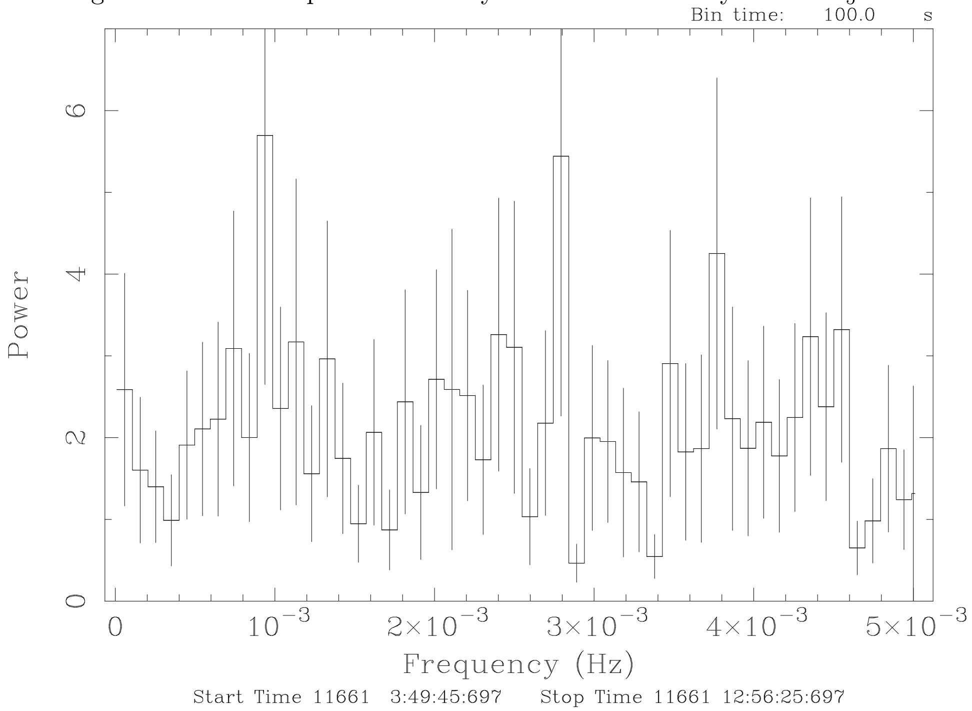


Figure 5.11: Power spectrum density for the source analyzed as in § 5.3.



1) Use the Xronos command *lcurve* to produce a binned lightcurve. The following command will also produce a screen plot using QDP ("quit" or "exit" will exit the QDP session).

```
lcurve nser=1 cfile1=source.lc window=- dtnb=500 nbint=450
   outfile=lightcurve.fits plot=yes plotdev=/xw
```

where

`nser` – number of time series

`cfile1` – filename first series

`window` – name of window file (if a subset of the time series is required)

`dtnb` – bin size (time)

`nbint` – number of bins per interval

`outfile` – output file name (FITS format light curve)

`plot` – plot flag

`plotdev` – device for plotting, output shown in Figure 5.10

2) Use the Xronos command *powspec* calculate power spectrum density. The following command will also produce a screen plot using QDP ("quit" or "exit" will exit the QDP session).

```
powspec cfile1=source.lc window=- dtnb=100.0 nbint=300
   nintfm=INDEF rebin=5 plot=yes plotdev=/xw outfile=power.fits
```

where

`cfile1` – filename first series

`window` – name of window file (if a subset of the time series is required)

`dtnb` – bin size (time)

`nbint` – number of bins per interval

`nintfm` – number of intervals in each power spectrum

`rebin` – rebin factor for power spectrum (0 for no rebinning)

`plot` – plot flag

`plotdev` – device for plotting, output shown in Figure 5.11

`outfile` – output file name (FITS format power spectrum)

3) Use the Xronos command *efsearch* to search for periodicities in the time series. The following command will also produce a screen plot using QDP ("quit" or "exit" will exit the QDP session).

```
efsearch cfile1=source.lc window=- sepoch=INDEF dper=20 nphase=10
   nbint=INDEF nper=100 dres=INDEF plot=yes plotdev=/xw outfile=efsearch.fits
```

where

`cfile1` – filename first series

`window` – name of window file (if a subset of the time series is required)

`sepoch` – value for epoch used for phase zero when folding the time series

`dper` – value for the period used in the folding

`nphase` – number of phases per period

`nbint` – number of bins per interval

`nper` – number of sampled periods during search

`dres` – sampling resolution of search

`plot` – plot flag

`plotdev` – device for plotting

`outfile` – output file name (FITS format)

4) Use the Xronos command *autocor* to calculate the auto correlation for an input time series. The following command will also produce a screen plot using QDP ("quit" or "exit" will exit the QDP session).

```
autocor cfile1=source.lc window=- dtnb=24.0 nbint=2048 nintfm=INDEF
   rebin=0 plot=yes plotdev=/xw outfile=auto.fits
```

where

`cfile1` – filename first series

`window` – name of window file (if a subset of the time series is required)

`dtnb` – bin size (time)

`nbint` – number of bins per interval

`nintfm` – number of intervals to be summed in each autocorrelation function

`rebin` – rebin factor for autocorrelation function (0 for no rebinning)

`plot` – plot flag

`plotdev` – device for plotting

`outfile` – output file name (FITS format autocorrelation spectrum)

5) Use the Xronos command *lcstats* to calculate statistical quantities for an input time series. The following command will write the output to an ASCII file `fname`. (Leave off the > `fname` to write the results to the screen.)

```
lcstats cfile1=source.lc window=- dtnb=6.0 nbint=8192 > fname
```

where

`cfile1` – filename first series

`window` – name of window file

`dtnb` – integration time (binning)

`nbint` – number of bins

`fname` – output file name

5.6 Once More, This Time with Feeling and Ftools

Most of the data extraction described in the previous sections can be done equally well in Ftools, and will be illustrated here using *fselect* and *Xselect*. Note that the HEASoft package is incorporated into SAS and so if SAS is operational, *fselect* and *Xselect* will be available. Keith Arnaud is responsible for the *XMM-Newton*-specific tools mentioned below, which he describes at:
`http://lheawww.gsfc.nasa.gov/users/kaa/xselect/xmm.html`

1) Filter the event file using the *xmmclean perl* script provided by Arnaud at the HTML page above by typing the following:

```
fselect mos1.fits mos1-filt.fits "FLAG ==0 && TIME <= 73227600)
   &&!(TIME in [73221920:73223800]) && PATTERN <= 12
   && PI <= 12000 && PI >= 200"
```

2) Invoke an *Xselect* session.

```
xselect
```

Enter a session name or default with a carriage return.

3) Read in the event list.

```
read events mos1-filt.fits
```

Enter the directory containing the event file; enter `yes` to reset the mission.

4) Create and plot an image (this will spawn a *ds9* window).

```
extract image
plot image
```

5) Create and plot a light curve (this will spawn a *Pgplot* window).

```
extract curve
plot curve
```

6) Filter the data. This can be done in one of two ways: using the cursor and light curve plot, or by using a threshold intensity.

   a) To filter on time using the cursor and light curve plot, type
      `filter time cursor`
      and follow the instructions.
      - Enter "quit" at the `PLT` prompt.
      - Right-click at the start and end points of the time intervals to keep.
      - When done entering intervals, enter "x" on the keyboard

   b) To filter on time using a threshold intensity, type
      `filter time cursor` ***`range`***
      where *range* is the filter range, e.g., `0.01-3.0`.

7) Create the extraction region for the source.

   a) Display the filtered image output from step 6 with *ds9*.
   b) Create a region on the *ds9* window:
      - In the *ds9* window pull down the `Region` menu and set 1) the `File Format` to `DS9/Funtools`, 2) the `File Coordinate System` to `Equatorial J2000`, and 3) the `Region Coordinate System` to `Degrees`
   c) Adjust the region to be appropriate for the source of interest
   d) Under the region menu select the `Save Regions` option
   e) Save the region as a file (e.g., `ds9-source.reg`)

8) Create an annulus extraction region for the background.

   a) If necessary, resize the existing region to be appropriate for the inner annulus radius
   b) Pull down the `Region` menu and select `Exclude` under `Properties`
   c) Create a second region on the *ds9* window
   d) Adjust the region to be appropriate for the outer boundary of the annulus
   e) Pull down the `Region` menu and select `Include` under `Properties`
   f) Make sure that the outer annulus is "in front" by selecting the `Move to Front` option under the `Region` menu.
   g) Under the region menu select the `Save Regions` option
   h) Save the region (e.g., `ds9-back.reg`)

9) Filter the data using the source region.

   – `filter region ds9-source.reg`

10) Extract, plot, save the spectrum from the source region and create RMF and ARF files.

```
extract spectrum
plot spectrum
```

`save spectrum resp=yes`

The `resp=yes` runs the *perl* script `xsl_xmm_epic_makeresp` which is available from Arnaud's web page above

- Enter a file name for the spectrum, e.g., `mos1-source.pi`
- Bin the data (i.e., enter `yes` at the query)

11) Filter the data using the background region.

    a) First remove the source filter expression: `clear region all`

    b) Now invoke the background filter: `filter region ds9-back.reg`

12) Extract, plot, and save the spectrum from the background region.

    – `extract spectrum`

    – `plot spectrum`

    – `save spectrum`

      - Enter a file name for the spectrum, e.g., `mos1-back.pi`
      - Bin the data (i.e., enter `yes` at the query)

13) Extract, plot, and save the light curve from the region.

    a) First remove the source filter expression: `clear region all`

    b) Now invoke the source filter: `filter region ds9-source.reg`

    c) And extract the light curve.

       `extract curve binsize=1000 phalcut_t=300 phahcut_t=10000`

       - use `binsize=1` to create a light curve for timing analysis
       - use `phalcut_t` to set the lower energy bound for the light curve
       - use `phahcut_t to set the upper energy bound for the light curve`

       `plot curve`

       `save curve`

       - Enter a file name for the light curve

From this point follow the procedures in § 5.3.7 and § 5.3.8 for spectral analysis and § 5.5 for temporal analysis.

5.7 In a Nutshell

To summarize, the basic steps taken in EPIC data reduction are as follows.

1) Obtain the raw and pipelined data.

2) Initialize SAS.

3) Make the CCF and ODF summary file (run the *cifbuild* and *odfingest* tasks).

4) Rerun the pipeline (run the *emchain/emproc* and *epchain/epproc* tasks).

5) Make the images and light curves.

6) Filter the event files.

7) Make filtered images.

8) Extract the source and background spectra.

9) Check for pile up (*eplatplot*).

10) Make the response files (*rmfgen* and *arfgen*).

11) Fit the spectrum.

Chapter 6

An RGS Data Processing and Analysis Primer

Before beginning this chapter please consult the “watchout” page at the VILSPA SOC:

```
http://xmm.esac.esa.int/sas/7.0.0/watchout/
```

This web site discusses current and past SAS bugs and analysis issues.

Many files are associated with an RGS dataset, and it is easy to be overwhelmed. The `INDEX.HTM` file, and links therein, are viewable with a web browser and will help you navigate the dataset. The different types of files are discussed in more detail in Chapter 3.

As ever, it is strongly recommended that you keep all reprocessed data in its own directory! SAS places output files in whichever directory it is in when a task is called. Throughout this primer, it is assumed that the Pipleline Processed data are in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, the reprocessing and analysis is taking place in the PROC directory, and the CCF data are in the directory CCF.

If you have just received your data from the SOC, it has been processed with the most recent version of SAS, and you should not need to reprocess it (though no harm is done if you do). However, it is very likely that you will want to filter your data; in this case, you will need to reprocess it in order to determine the appropriate filters. Therefore, we recommend that you rerun the pipeline regardless of the age of your dataset.

But if you decide that reprocessing is unnecessary, you need only to gunzip the files and rename event files for easier handling. For example, for the RGS1 event list,

```
gunzip ODF/*.gz
gunzip PPS/*.gz
mv PPS/PiiiiiijjkkR1lEVENLInmmm.FTZ PPS/PiiiiiijjkkR1lEVENLInmmm.FIT.gz
gunzip PPS/PiiiiiijjkkR1lEVENLInmmm.FIT.gz PPS/r1_evt.fits
```

where

`iiiiiijjkk` – observation number

`l` – scheduled (`S`) or unscheduled (`U`) obseravtion `n` – spectral order number

`mmm` – source number

6.0.1 A Quick Look at What You Have

As noted in Tables 3.2 and 3.3 you can view images of your data. While the zipped FITS files may need to be unzipped before display in *ds9* (depending on the version of *ds9*), they can be displayed when zipped using *fv* (*fv* is FITS file viewer available in the HEASoft package). As usual, there are some HTML products to help you inspect the data. These have file names of the form:

- `PPiiiiiijjkkAAAAAA000_0.HTM`

where

iiiiiijjkk – Observation number

jj – observation ID - target number in proposal

kk – observation ID - observation number for target

AAAAAA – Group ID (see Table 3.2)

You will find a variety of RGS-specific files in *XMM-Newton* data sets. Generally there are two of each because there are two RGS instruments. Table 3.3 lists typical file names, their purpose, the file format, and a list of tools that will enable the user to inspect their data. Remember that the INDEX.HTM file will help you navigate.

6.1 Rerunning the pipeline

You will need to reprocess the data before being able to determine the appropriate filters, but before doing that, you must prepare the data and initialize SAS. Throughout this primer, we will use the AB Dor dataset with ObsID 0134520301 available through links at the HEASARC archive.

1) If you have not already done so, gunzip all gzipped files in the ODF and PPS directories. If necessary, rename all files in the ODF directory to upper case. This can be done using the script provided by the NASA/GSFC *XMM-Newton* GOF.

```
gunzip ODF/*.gz
gunzip PPS/*.gz
```

2) Initialize SAS by calling one of the two setup scripts, depending on which shell you use.

```
source /full/path/to/xmmsas_20080701_1801/setsas.csh
```

or

```
. /full/path/to/xmmsas_20080701_1801/setsas.sh
```

It is **strongly** recommended that you add a line to your login shell file to set up an alias to these scripts! Calling the script will deal with most of the details needed to run SAS, except for three environment variables, which we will set next.

3) Set the SAS directory pointers. To verify the SAS-specific settings, use the command `env | grep SAS`. (For a detailed discussion of SAS initialization, see Chapter 4.)

```
setenv SAS_ODF full/path/to/ODF
setenv SAS_CCFPATH full/path/to/CCF
```

At this point, the SAS GUI can be run by typing `sas &` in the window where the pointers were set. However, since the next few procedures are very simple, it is faster to just use the command line.

4) If it doesn't already exist, create a CIF file in the ODF directory using the SAS task *cifbuild* (see §4.5.1). If a CIF file has previously been produced, it is only necessary to rerun *cifbuild* if the CCF has changed. Be sure to set the environment parameter afterwards.

```
cd ODF
cifbuild
setenv SAS_CCF /full/path/to/ODF/ccf.cif
```

5) If it hasn't already been done (don't do it twice), while still in the ODF directory, prepare the data by using the SAS task *odfingest* (see §4.5.2) and setting the environment parameter. It is only necessary to run it once on any data set and will cause problems if it is run a second time. If for some reason *odfingest* must be rerun, you must first delete the earlier file produced by *odfingest* (*SUM.SAS).

```
odfingest
setenv SAS_ODF /full/path/to/file/full_name_of_*SUM.SAS
```

6) In your “processing directory” PROC, run the SAS task *rgsproc*. From the command line of a window where SAS has been initialized, enter:

```
rgsproc orders='1 2' bkgcorrect=no withmlambdacolumn=yes
```

where

`orders` – dispersion orders to extract

`bkgcorrect` – subtract background from source spectra?

`withmlambdacolumn` – include a wavelength column in the event file product

This takes several minutes, and outputs 12 files per RGS, plus 3 general use FITS files. At this point, renaming files to something easy to type is a good idea.

```
ln -s *R1*EVENLI*FITS r1_evt1.fits
ln -s *R2*EVENLI*FITS r2_evt1.fits
```

Once the new event files have been obtained, the analysis techniques described in §6.3 and later can be used.

6.2 Potentially useful tips for using the pipeline

The pipeline task, *rgsproc*, is very flexible and can address potential pitfalls for RGS users. In §6.1, we used a simple set of parameters with the task, and if this is sufficient for your data, feel free to skip to §6.3. In the following sections, we will look at the cases of a nearby bright optical source, a nearby bright X-ray source, and a user-defined source.

6.2.1 A Nearby Bright Optical Source

With certain pointing angles, zeroth-order optical light may be reflected off the telescope optics and cast onto the RGS CCD detectors. If this falls on an extraction region, the current energy calibration will require a wavelength-dependent zero-offset. Stray light can be detected on RGS DIAGNOSTIC images taken before, during and after the observation. This test, and the offset correction, are not performed on the data before delivery. To check for stray light and apply the appropriate offsets use, type

```
rgsproc orders='1 2' bkgcorrect=no calcoffsets=yes withoffsethistogram=no
```

where the parameters are as described in §6.1 and

`calcoffsets` – calculate PHA offsets from diagnostic images

`withoffsethistogram` – produce a histogram of uncalibrated excess for the user

6.2.2 A Nearby Bright X-ray Source

In the example above, it is assumed that the field around the source contains sky only. Provided a bright background source is well-separated from the target in the cross-dispersion direction, a mask can be created that excludes it from the background region. Here the source has been identified in the EPIC images and its coordinates have been taken from the EPIC source list which is included among the pipeline products. The bright neighboring object is found to be the third source listed in the sources file. The first source is the target:

```
rgsproc orders='1 2' bkgcorrect=no withepicset=yes
   epicset=PiiiiiijjkkaablllEMSRLInmmm.FTZ exclsrcsexpr='INDEX==1&&INDEX==3'
```

where the parameters are as described in §6.1 and `withepicset` – calculate extraction regions for the sources contained in an EPIC source list

`epicset` – name of the EPIC source list, such as generated by *emldetect* or *eboxdetect* procedures

`exclsrcsexpr` – expression to identify which source(s) should be excluded from the background extraction region

6.2.3 User-defined Source Coordinates

If the true coordinates of an object are not included in the EPIC source list or the science proposal, the user can define the coordinates of a new source by typing:

```
rgsproc orders='1 2' bkgcorrect=no withsrc=yes srclabel=ABDor srcstyle=radec
   srcra=82.185493 srcdec=-65.449329
```

where the parameters are as described in §6.1 and

`withsrc` – make the source be user-defined

`srclabel` – source name

`srcstyle` – coordinate system in which the source position is defined

`srcra` – the source's right ascension in decimal degrees

`srcdec` – the source's declination in decimal degrees

6.3 Examine and Filter the Data

Since the event files are current, we can proceed with some simple analysis demonstrations, which will allow us to generate filters. The following sections describe the use of SAS tasks using the both the command line and GUI interfaces, except in cases where one of the methods is particularly easy. People new to SAS will likely prefer the GUI, at least at first; however, as they become more familiar with the software and the keywords, they will probably migrate to the command line, which is faster. Assuming that the parameter values for any given task are the same, it does not matter if a task is invoked on the command line or in the GUI; the output files will be identical. The SAS *xmmselect* GUI provides a very simple method for producing and displaying images, spectra, and light curves, and is the recommended method for extracting data unless large numbers of sources are being analyzed.

6.3.1 An Introduction to the SAS GUI and *xmmselect*

We are now ready to invoke the SAS GUI if we have not already done so. Make sure that you are in the directory where you want the output to go before invoking the SAS GUI or any of the SAS tasks on the command line!

```
sas &
```

1) Invoke the *xmmselect* GUI from the SAS GUI. To invoke a task, you need only start typing the task name, and when it is highlighted, hit a carriage return.

   - When *xmmselect* is invoked a dialog box will first appear requesting a file name. One can either use the browser button or just type the file name in the entry area, "r1_evt1.fits" in this case. To use the browser, first click on the file folder icon button on the right which will bring up a second GUI for the file selection. Double click on the desired event file in the right-hand column (you may have to open the appropriate directory first), click on the "EVENTS" extension in the right-hand column (which selects the extension), and then click "Ok". The directory GUI will then disappear and then click "Run" on the selection GUI.
   - When the file name has been submitted the *xmmselect* GUI (Figure 5.2) GUI will appear, along with a dialog box offering to display the selection expression. The selection expression will include the filtering done to this point on the event file, which for the pipeline processing includes for the most part CCD and GTI selections.

6.3.2 Create and Display an Image

Two commonly-made plots are those showing PI vs. BETA_CORR (also known as "banana plots") and XDSP_CORR vs. BETA_CORR.

To create images by using the xmmselect GUI:

1) Check the square boxes to the left of the "BETA_CORR" and "PI" entries.

2) Click on the "Image" button near the bottom of the page. This brings up the *evselect* GUI (Figure 6.3).

3) Click on the "Image" tab in the *evselect* GUI.

4) Confirm that the withimageset box is checked.

5) In the imageset box, change the output image name from image.ds to something descriptive, in this case, pi_bc.fits.

6) Click on the "Run" button on the lower left corner of the *evselect* GUI.

   - Different binnings and other selections can be invoked by accessing the "Image" tab at the top of the GUI. The default settings are reasonable, however, for a basic image.
   - The resultant image is automatically displayed using *ds9*. Similarly, plots can be made comparing BETA_CORR to XDSP_CORR. These two example plots can be seen in Figure 6.4.

To create images by using the task *evselect* on the command line:

1) In the window in which SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line). Make sure the correct path is given for the event file.

```
evselect table=r1_evt1.fits:EVENTS withimageset=yes
   imageset=pi_bc.fits xcolumn=BETA_CORR ycolumn=PI
   imagebinning=imageSize ximagesize=600 yimagesize=600
```

where

table – input event table

withimageset – make an image

imageset – name of output image

xcolumn – event column for X axis

ycolumn – event column for Y axis

imagebinning – form of binning, force entire image into a given size or bin by a specified number of pixels

ximagesize – output image pixels in X

yimagesize – output image pixels in Y

   - Plots comparing BETA_CORR to XDSP_CORR may be made in a similar way. The output files can be viewed by using a standard FITS display, such as *ds9* (see Figure 6.4) :

```
ds9 pi_bc.fits &
```

6.3.3 Create and Display a Light Curve

The background is assessed through examination of the light curve. We will extract a region, CCD9, that is most susceptible to proton events and generally records the least source events due to its location close to the optical axis. Also, to avoid confusing solar flares for source variability, a region filter that that removes the source from the final event list should be used. The region filters are kept in the source file product P*SRCLI_*.FIT. (For our example data, this would be P0134520301R1S001SRCLI_0000.FIT).

To create light curves of the observation by using the *xmmselect* GUI:

1) Enter the filtering criteria in the "Selection expression" box at the top of the *xmmselect* GUI: `(CCDNR==9)&&(REGION(P0134520301R1S001SRCLI_0000.FIT:RGS1_BACKGROUND,BETA_CORR,XDSP_CORR))`

2) Check the round box to the left of the `time` entry.

3) Click on the "OGIP Rate Curve" button near the bottom of the page. This brings up the *evselect* GUI (Figure 5.3).

4) Click on the "Lightcurve" tab and confirm that the `withrateset` box is checked.

5) Change the `timebinsize` to a reasonable amount, e.g. 10 or 100 s, and change the default output file name in the `rateset` box to something appropriate, in this case, `r1_ltcrv.fits`.

6) Click on the "Run" button at the lower left corner of the *evselect* GUI.

   - The resultant light curve is displayed automatically using Grace (see Figure 6.1).

To create a light curve of the observation by using the task *evselect* on the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line). Make sure the correct path is given for the event file.

```
evselect table=r1_evt1.fits:EVENTS withrateset=yes rateset=r1_ltcrv.fits
   maketimecolumn=yes timebinsize=100 makeratecolumn=yes
   expression=
   '(CCDNR==9)&&(REGION(P0134520301R1S001SRCLI_0000.FIT:RGS1_BACKGROUND,BETA_CORR,XDSP_CORR))'
```

where

`table` – input event table

`withrateset` – make a light curve

`rateset` – name of output light curve file

`maketimecolumn` – control to create a time column

`timebinsize` – time binning (seconds)

`makeratecolumn` – control to create a count rate column, otherwise a count column will be created

`expression` – filtering criteria

- The output file `r1_ltcrv.fits` can be viewed using *dsplot*:

```
dsplot table=r1_ltcrv.fits x=TIME y=RATE &
```

where

`table` – input event table

`x` – column for plotting on the X axis

`y` – column for plotting on the Y axis

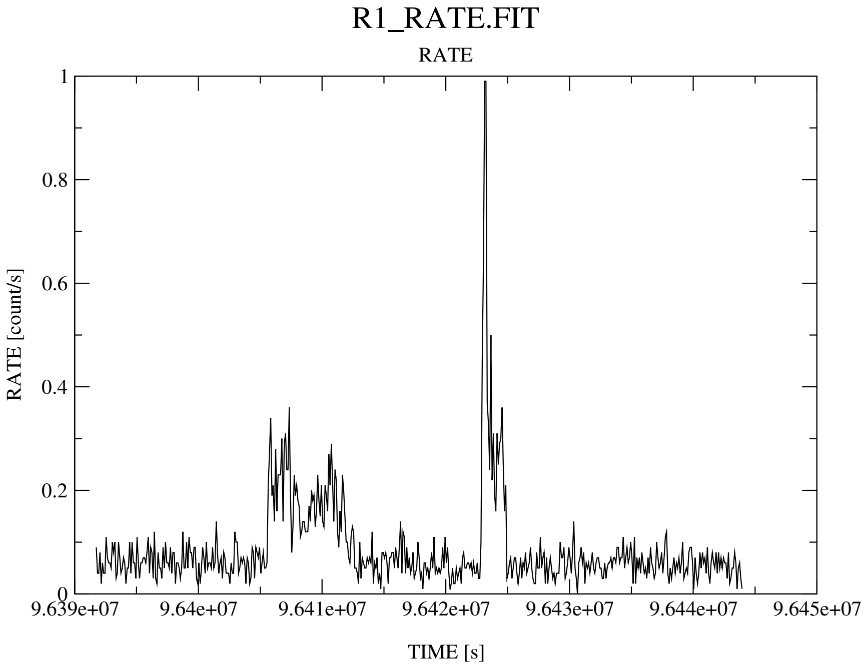
The light curve is shown in Figure 6.1.

6.3.4 Generating the Good Time Interval (GTI) File

Examination of the lightcurve shows that there are two noisy sections, one between 9.6405e7 and 9.6413e7 seconds, and another between 9.6422e7 and 9.6425e7 seconds. Both show rates well in excess of the normal background count rate of ∼ 0.05 count/second. There are two procedures that make the GTI file (*gtibuild* and *tabgtigen*) that, when applied to the event file in another run of *rgsproc*, will excise these sections.

The first method, using *gtibuild*, requires a text file as input. In the first two columns, refer to the start and end times (in seconds) that you are interested in, and in the third column, indicate with either a + or - sign whether that region should be kept or removed. In the example case, then, we would write in our ASCII file (named `gti.txt`):

Figure 6.1: Background event rate from the RGS1 CCD9 chip. The flares are solar events. The time units are elapsed mission time.



```
9.6405e7 9.6413e7 -
9.6422e7 9.6425e7 -
```

and proceed to the SAS task *gtibuild*.

To make the GTI with *gtibuild* in the SAS GUI:

1) Double-click on the *gtibuild* task in the SAS GUI.
2) Enter the name of the text file in the `file` box.
3) Enter the name of the output fits file in the `table` box.
4) Click “Run”.

To make the GTI with *gtibuild* on the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command.

   ```
   gtibuild file=gti.txt table=gti.fits
   ```

   where

   `file` – intput text file

   `table` – output gti table

To make the GTI with *tabgtigen* in the SAS GUI:

1) Double-click on the *tabgtigen* task.
2) Enter the name of the lightcurve file in the `table` box, in this case, `r1_ltcrv.fits`.

3) Enter the name of the output file in the gtiset box, in this case, gti.fits.

4) Enter the filtering expression in the expression box. Since the nominal count rate is about 0.05 count/sec, we will set the upper limit to 0.2 count/sec: RATE<0.2

5) Click “Run”.

To make the GTI with *tabgtigen* from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command.

```
tabgtigen table=r1_ltcrv.fits gtiset=gti.fits expression='(RATE<0.2)'
```

where

table – the lightcurve file

gtiset – output gti table

expression – the filtering criteria. Since the nominal count rate is 0.05 about count/sec, we have set the upper limit to 0.2 count/sec.

6.3.5 Applying the GTI

Now that we have GTI file, we can apply it to the event file by running *rgsproc* again. *rgsproc* is a complex task, running several steps, with five different entry and exit points. It is not necessary to rerun all the steps in the procudure, only the ones involving filtering.

To rerun the pipeline in the SAS GUI:

1) Double-click on *rgsproc* in the SAS GUI.

2) In the “global” tab, make sure that the orders box is set for both orders, 1 2 .

3) In the “global” tab, use the pulldown menus for entrystage and exitstage to select 3:filter and 5:fluxing, respectively.

4) In the “filter” tab, enter the name of the GTI file, in this case, gti.fits, in the auxgtitables box.

5) In the “spectra” tab, click on the “rgsspectrum” tab, and make sure that bkgcorrect is set to no.

6) In the “angles” tab, make sure the withmlambda column is set to yes.

7) Click on “Run”.

To rerun the pipeline from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command, all on one line:

```
rgsproc orders='1 2' auxgtitables=gti.fits bkgcorrect=no
   withmlambdacolumn=yes entrystage=3:filter finalstage=5:fluxing
```

where

orders – spectral orders to be processed

auxgtitables – gti file in FITS format

bkgcorrect – subtract background from source spectra?

withmlambdacolumn – include a wavelength column in the event file product

entrystage – stage at which to begin processing

finalstage – stage at which to end processing

6.3.6 Creating the Response Matrices (RMFs)

Response matrices (RMFs) are not provided as part of the pipeline product package, so you must create your own before analyzing data. This can be done with the package *rgsrmfgen*.

To make the RMFs using the GUI:

1) Double-click on the *rgsrmfgen* task in the GUI.

2) In the `spectrumset` box, enter the name of the spectrum file; it has the form `*SRSPEC*`, and in our case, is `P0134520301R1S001SRSPEC1001.FIT`.

3) In the `evlist` box, enter the name of the event list, r1_evt1.fits.

4) Set `emin` to `0.4`, `emax` to `2.5`, and `rows` to `5000`.

5) Set `rmfset` to the output file name, in this case, `r1_o1_rmf.fits`.

6) Click "Run".

To make the RMFs from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following command (all on one line).

```
rgsrmfgen spectrumset=P0134520301R1S001SRSPEC1001.FIT rmfset=r1_o1_rmf.fits
   evlist=r1_evt1.fits emin=0.4 emax=2.5 rows=5000
```

where

`spectrumset` – spectrum file

`evlist` – event file

`emin` – lower energy limit of the response file

`emax` – upper energy limit of the response file

`rows` – number of energy bins; this should be greater than 3000

`rmfset` – output FITS file

6.4 Fitting a Spectral Model

Now that we have a response file, we can fit the spectrum using Xspec.

1) On the command line, type:

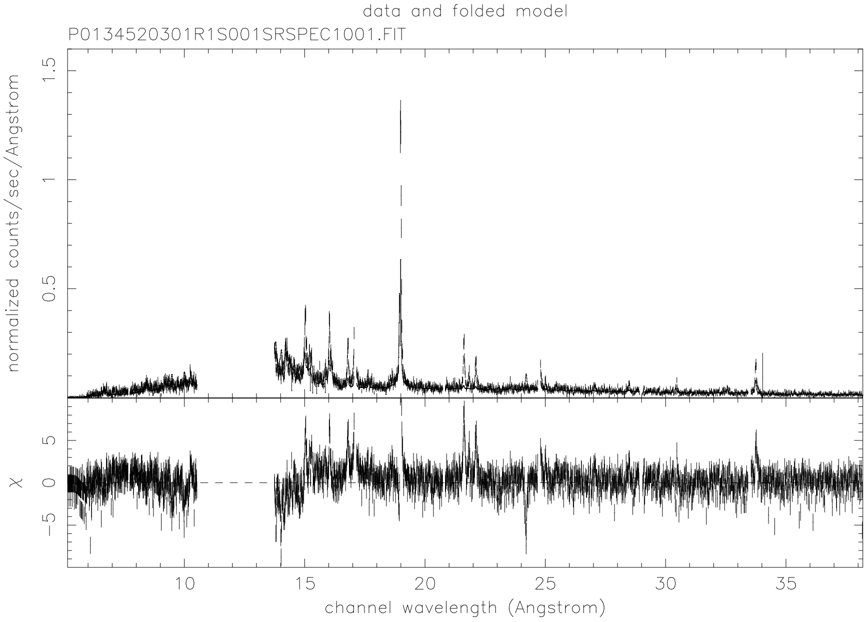
```
xspec
```

Enter the data, background, and response file at the prompts, and edit the fitting parameters as needed.

```
XSPEC> data P0136540101R1S001SRSPEC1003.FIT ! input data
XSPEC> back P0136540101R1S001BGSPEC1003.FIT ! input background
XSPEC> resp r1_o1_rmf.fits                  ! input response file
XSPEC> model wabs*mekal                     ! set spectral model to absorbed mekal
wabs:nH> 1
mekal:kT> 1
mekal:nH>
mekal:Anbundanc> .4
mekal:Redshift>
mekal:Switch> 0
mekal:norm> 1
XSPEC> renorm
XSPEC> fit
XSPEC> cpd /xw
XSPEC> setplot wave
XSPEC> setplot command window all
XSPEC> setplot command log x off
XSPEC> setplot command wind 1
XSPEC> setplot command r y 1e-5 1.6
XSPEC> setplot command wind 2
XSPEC> setplot command r y -9.99 9.99
XSPEC> plot data residuals
XSPEC> exit
```

Figure 6.2 shows the fit to the spectrum.

Figure 6.2: 1st order RGS1 spectrum of AB Dor. The fit is an absorbed single-temperature mekal model. The gap between 10–15Å is due to the absence of CCD7.



still 11–Mar–2002 16:18

6.4.1 Combining RGS1 and RGS2 Spectra

While it is tempting to merge the RGS1 and RGS2 data, or data from different pointings, to provide a single spectrum with a signal-to-noise improvement over either individual spectrum, this is strongly discouraged since it results in data degradation.

The pointings of the two instruments are not identical, resulting in different dispersion angles and wavelength scales. Separate response files are always required for each unit. While it is possible to merge spectra and response files, great care must be taken to account for different exposure times, background subtractions, error propagation, and so on. However, the resulting response will always have inferior resolution to the originals. It is therefore simpler and more accurate to keep data from the two RGS units separate and use both sets to fit one model in tandem.

1) On the command line, type:

```
xspec
```

```
XSPEC>data 1:1 P0136540101R1S001SRSPEC1003.FIT 1:2 P0136540101R1S001SRSPEC2003.FIT
XSPEC>ignore bad
XSPEC>model phabs*mekal
```

6.5 Approaches to Spectral Fitting

For data sets of high signal-to-noise and low background, where counting statistics are within the Gaussian regime, the data products above are suitable for analysis using the default fitting scheme in XSPEC, χ^2-minimization. However, for low count rates, in the Poisson regime, χ^2-minimization is no longer suitable. With low count rates in individual channels, the error per channel can dominate over the count rate. Since channels are weighted by the inverse-square of the errors during χ^2 model fitting, channels with the lowest count rates are given overly-large weights in the Poisson regime. Spectral continua are consequently often fit incorrectly, with the model lying underneath the true continuum level.

This will be a common problem with most RGS sources. Even if count rates are large, much of the flux from these sources can be contained within emission lines, rather than the continuum. Consequently, even obtaining correct equivalent widths for such sources is non-trivial. There are two approaches to fitting low signal-to-noise RGS data, spectral rebinning and maximum-likelihood statistics. The correct approach would normally be to use an optimization of the two.

6.5.1 Spectral Rebinning

By grouping channels in appropriately large numbers, the combined signal-to-noise of groups will jump into the Gaussian regime. There are two ways to do this: the FTOOL *grppha*, or the RGS pipeline. *grppha* can group channels using an algorithm which bins up consecutive channels until a count rate threshold is reached. This method conserves the resolution in emission lines above the threshold while improving statistics in the continuum.

1) On the command line, type the following and edit parameters as needed.

```
grppha

          > Please enter PHA filename[] P0136540101R1S001SRSPEC1003.FIT
          > Please enter output filename[] P0136540101R1S001SRSPEC1003.bin.FIT
          > GRPPHA[] group min 30
          > GRPPHA[] exit
```

The disadvantage of using *grppha* is that, although channel errors are propagated through the binning process correctly, the errors column in the original spectrum product is not strictly accurate. The problem arises because there is no good way to treat the errors within channels containing no counts. To allow statistical fitting, these channels are arbitrarily given an error value of unity, which is subsequently propagated through the binning. Consequently, the errors are overestimated in the resulting spectra.

The other approach, which involves calling the RGS pipeline after it is complete, bins the data during spectral extraction. The following rebins the pipeline spectrum by a factor 3.

1) On the command line, type

```
rgsproc orders='1 2' rebin=3 rmfbins=4000 entrystage=4:spectra
   finalstage=5:fluxing bkgcorrect=no
```

where

`orders` – dispersion orders to extract

`rebin` – wavelength rebinning factor

`rmfbins` – number of bins in the response file; this should be greater than 3000

`entrystage` – entry stage to the pipeline

`finalstage` – exit stage for the pipeline

One disadvantage of this approach is that you can only choose integer binning of the original channel size. To change the sampling of the events, the pipeline must be run from the second stage ("angles") or earlier.

1) On the command line, type

```
rgsproc orders='1 2' nbetabins=1133 rmfbins=4000 entrystage=2:angles
   finalstage=fluxing bkgcorrect=no
```

where the parameters are as defined previously, and

`nbetabins` – number of bins in the dispersion direction; the default is 3400

The disadvantage of using *rgsproc*, as opposed to *grppha*, is that the binning is linear across the dispersion direction. Velocity resolution is lost in the lines, so the accuracy of redshift determinations will be degraded, transition edges will be smoothed, and neighboring lines will become blended.

6.5.2 Maximum-Likelihood Statistics

The second method is to replace the χ^2-minimization scheme with the Cash maximum-likelihood scheme (`cstat` in Xspec) when fitting data. This method is much better suited to data with low count rates and is a suitable option only if one is running Xspec v11.1.0 or later. The reason for this is that RGS spectrum files have prompted a slight modification to the OGIP standard. Because the RGS spatial extraction mask has a spatial-width which is a varying function of wavelength, it has become necessary to characterize the `BACKSCL` and `AREASCL` parameters as vectors (i.e., one number for each wavelength channel), rather than scalar keywords as they are for data from the EPIC cameras and past missions. These quantities map the size of the source extraction region to the size of the background extraction region and are essential for accurate fits. Only Xspec v11.1.0, or later versions, are capable of reading these vectors, so be certain that you have an up-to-date installation at your site.

One caveat of using the `cstat` option is that the scheme requires a "total" and "background" spectrum to be loaded into Xspec. This is in order to calculate parameter errors correctly. Consequently, be sure not to use the "net" spectra that were created as part of product packages by SAS v5.2 or earlier. To change schemes in Xspec before fitting the data, type:

```
XSPEC> statistic cstat
```

6.6 Analysis of Extended Sources

6.6.1 Region masks

The optics of the RGS allow spectroscopy of reasonably extended sources, up to a few arc minutes. The width of the spatial extraction mask is defined by the fraction of total events one wishes to extract. With the default pipeline parameter values, 90% of events are extracted, assuming a point-like source.

Altering and optimizing the mask width for a spatially-extended source may take some trial and error, and, depending on the temperature distribution of the source, may depend on which lines one is currently interested in. While AB Dor is not an extended source, the following example increases the width of the extraction mask and ensures that the size of the background mask is reduced so that the two do not overlap.

To adjust the region mask with *rgsproc* in the SAS GUI:

1) Double-click on rgsproc in the SAS GUI.

2) In the "global" tab, make sure that the `orders` box is set for both orders, `1 2`.

3) In the "global" tab, use the pulldown menus for `entrystage` and `exitstage` to select `4:spectra` and `5:fluxing`, respectively.

4) In the "spectra" tab, in the "rgsregions" sub-tab, set both `xpsfincl` and `xpsfexcl` to `99`, and `pdistincl` to `95`.

5) Click "Run".

To adjust the region mask with *rgsproc* from the command line:

1) Type the following on the command line, in the directory where you want the output to go:

```
rgsproc orders='1 2' entrystage=4:spectra finalstage=5:fluxing bkgcorrect=no
   xpsfincl=99 xpsfexcl=99 pdistincl=95
```

where parameters are as they were decsribed previously, and

`xpsfincl` – include this fraction of point-source events inside the spatial source extraction mask

`xpsfexcl` – exclude this fraction of point-source events from the spatial background extraction mask

`pdistincl` – include this fraction of point-source events inside the pulse height extraction mask

Observing extended sources effectively broadens the psf of the spectrum in the dispersion direction. Therefore, it is prudent to also increase the width of the PI masks using the `pdistincl` parameter in order to prevent event losses.

6.6.2 Fitting spectral models to extended sources

RGS response matrices are consistent for point sources only. Since extended source spectra are broadened, the simplest way to deal with this problem during spectral fitting is to reproduce the broadening function, and convolve it across the spectral model. Xspec v11.2 contains the convolution model `rgsxsrc`. It requires two external files to perform the operation:

1) An OGIP FITS image of the source. The better the resolution of the image, the more accurate the convolution. For example, if a Chandra image of the source is available, this will provide a more accurate result than an EPIC image.

2) An ASCII file containing three lines of input. For this example case, we will name it `xsource.mod`. It defines three environment variables and should look like this example:

```
RGS_XSOURCE_IMAGE ./MOS1.fit

RGS_XSOURCE_BORESIGHT 23:25:19.8 -12:07:25 247.302646

RGS_XSOURCE_EXTRACTION 2.5
```

where

`RGS_XSOURCE_IMAGE` – path to the source image

`RGS_XSOURCE_BORESIGHT` – RA, Dec of the center of the source and PA of the telescope

`RGS_XSOURCE_EXTRACTION` – The extent (in arcmin), centered on the source, over which you want to construct the convolution function. You want this "aperture" to be larger than the source itself.

To set these environment variables within Xspec execute the command:

```
xset rgs_xsource_file xsource.mod
```

Here is an example. Note that the spectral order is always negative.

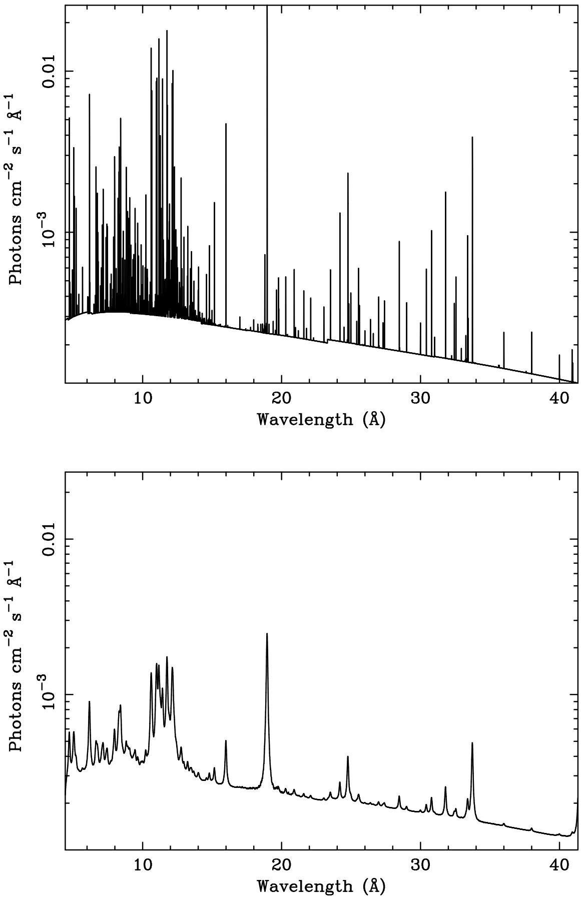
```
xspec

    XSPEC>data P0108460201R1S004SRSPEC1003.FIT
    XSPEC>ignore bad
    XSPEC>xset rgs_xsource_file xsource.mod
    XSPEC>model rgsxsrc*wabs*mekal
    rgsxsrc:order>-1
    wabs:nH>1
    mekal:kT>2
    mekal:nH>1
    mekal:Abundanc>1
```

```
mekal:Redshift>
mekal:Switch>0
mekal:norm>1
XSPEC>renorm
XSPEC>fit
XSPEC>setplot device /xs
XSPEC>setplot wave
XSPEC>setplot command window all
XSPEC>setplot command log x off
XSPEC>plot data residuals
XSPEC>exit
Do you really want to exit? (y)y
```

Figure 6.3 compares a point source model with an extended source counterpart.

Figure 6.3: The top figure is a thin, thermal plasma at 2 keV from a point source. The lower figure is the same spectral model, but convolved by the MOS1 0.3–2.0 keV spatial profile of a low-redshift cluster.



6.6.3 Model limitations

Users should be aware that this method assumes an isothermal source (or uniform emissivity from line to line in the case of a non-thermal spectrum) where the spatial distributions of all the lines are identical. In reality, however, the thermal structure of the source is likely to be more complicated. The broad-band convolution function may bear little resemblance to the correct function for particular line transitions.

One way around this problem would be to have a temperature map of the source to define line emissivity across the source and convolve the model spectrum accordingly. The RGS instrument team at the

Columbia Astrophysics Laboratory are developing a Monte Carlo code to perform an operation with this effect. While it is unlikely the code will be publicly available in the near future, the team welcomes investigators who would be interested in collaboration. Interested parties are encouraged to contact John Peterson (jrpeters@astro.columbia.edu).

6.7 In A Nutshell

To summarize, the steps you must take to prepare your data for analysis are:

1) Obtain the raw and pipelined data from the XMM archive.

2) Initialize SAS.

3) Make the CCF file and ODF summary file.

4) Rerun the pipeline.

5) Generate a light curve to determine the appropriate filter for your data.

6) Make the good time interval (GTI) file.

7) Apply the GTI file by rerunning the pipeline again.

8) Generate the response file (RMF).

Chapter 7

An OM Data Processing and Analysis Primer

As with EPIC and RGS datasets, many files are associated with an OM dataset. The `INDEX.HTM` file, and links therein, are viewable with a web browser and will help you navigate the dataset. The different types of files are discussed in Chapter 3; however, since the OM is somewhat different from the other instruments on-board XMM-Newton, we will discuss them in more detail in §7.1.

The OM can operate in IMAGING, FAST, and GRISM mode. Each of these modes has dedicated *chain* commands to reprocess the data: *omichain*, *omfchain*, and *omgchain*. These each call several procedures that are used to prepare the data for processing, make and apply flatfield images, and detect sources. The tasks *omichain* and *omfchain* also calculate the instrumental magnitudes of sources, find the position of the sources (in equatorial coordinates), and produce a sky image; *omgchain* produces a spectrum. If you run these chains, it is helpful to inspect the `sas_log` file to get a detailed list of the performed tasks. These chains rely on filters specified by the user; if no arguments are given, they run on all the files present in the ODF directory.

Most OM data are obtained in IMAGING mode. If they were obtained in the FAST mode, there will be an additional event list file corresponding to the Fast window (`*FAE.FIT`). §7.2 discusses in detail how the chains work. Reprocessing of data taken in FAST mode is discussed in §7.2.10. Reprocessing OM Grism data is discussed in §7.2.11.

All OM images are affected by the so-called "stray-light" problem (see Fig. 7.1). This problem does **not** affect source detection and magnitude determination but contributes to a higher background (and an ugly appearance of the images). The stray-light problem is less noticeable at UV wavelengths. A (proprietary) program to produce clean images exists but the results are strictly for display purposes only since the routine does not conserve flux. Because the stray-light problem is mainly aesthetic, there are no plans to develop publicly available routines to deal with it.

As usual, you are strongly encouraged to keep all reprocessed data in a separate directory! SAS places output files in whichever directory it is in when a task is called. We will assume that the Pipeline Processed data are in the PPS directory, the ODF data (with upper case file names, and uncompressed) are in the directory ODF, and the analysis is taking place in the PROC directory.

7.1 Pipeline Products

As with the EPIC and RGS, you will find a variety of OM-specific files in your data directories. The pipeline products differ slightly with different versions of the SAS software. We give a brief description of the files produced by SAS V7, and discuss the important differences with older pipeline products. For a complete description of all files check the pipeline products documentation, which can be found at:
`http://xmmssc-www.star.le.ac.uk/pubdocs/SSC-LUX-SP-0004.ps.gz`

7.1.1 Imaging Mode

The PPS directory for the OM products contains files with nomenclature as described in Tables 3.2 and 3.3. As can be seen in those tables, the OM produces, among other things, sky images (`*SIMAGE*.FTZ`) and source lists (*SWSRLI*.FTZ). There is a low resolution sky image for each filter; they follow the nomenclature:

- PjjjjjjkkkkOMX000RSIMAGbb000.QQQ

  jjjjjj – Proposal number

  kkkk – Observation ID

  b – Filter keyword: B, V, U, L (UVW1) and S (UVW2)

  zzz – File type (e.g., PNG, FTZ)

For example, P0123456789OMX000RSIMAG**B**000.FTZ is the low-resolution final image in the **B** filter of the observation 0123456789 in sky coordinates (indicated by the S before the IMAG). The letter **L** is used for the UVW1 filter and **S** for UVW2. The keyword **XPROC0** in the FITS header lists the files which have been added to create the final image P0123456789OMX000RSIMAGB000.FTZ. The keyword looks like this:

```
XPROC0  = ’ommosaic imagesets=’’"product/P0123456789OMS008SIMAGE1000.FIT"&’
CONTINUE  ’ "product/P0123456789OMS409SIMAGE1000.FIT" "product/P01234567&’
CONTINUE  ’89OMS410SIMAGE1000.FIT" "product/P0123456789OMS411SIMAGE1000.&’
CONTINUE  ’FIT" "product/P0123456789OMS412SIMAGE1000.FIT"’’ mosaicedset=’’&’
CONTINUE  ’product/P0123456789OMX000RSIMAGS000.FIT’’ sampling=’’point’’ # (&’
CONTINUE  ’ommosaic-1.2.1) [xmmsas_20011206_1713-no-aka]’
```

The source list file (*SWSRLI*.FTZ) also contains useful information for the user; the column names are listed in Table 7.1.

Table 7.1: Some of the important columns in the SWSRLI FITS file.

Column name	Contents
SRCNUM	Source number
RA	RA of the detected source
DEC	Dec of the detected source
POSERR	Positional uncertainty
RATE	extracted count rate
RATE_ERR	error estimate on the count rate
SIGNIFICANCE	Significance of the detection (in σ)
MAG	Brightness of the source in magnitude
MAGERR	uncertainty on the magnitude

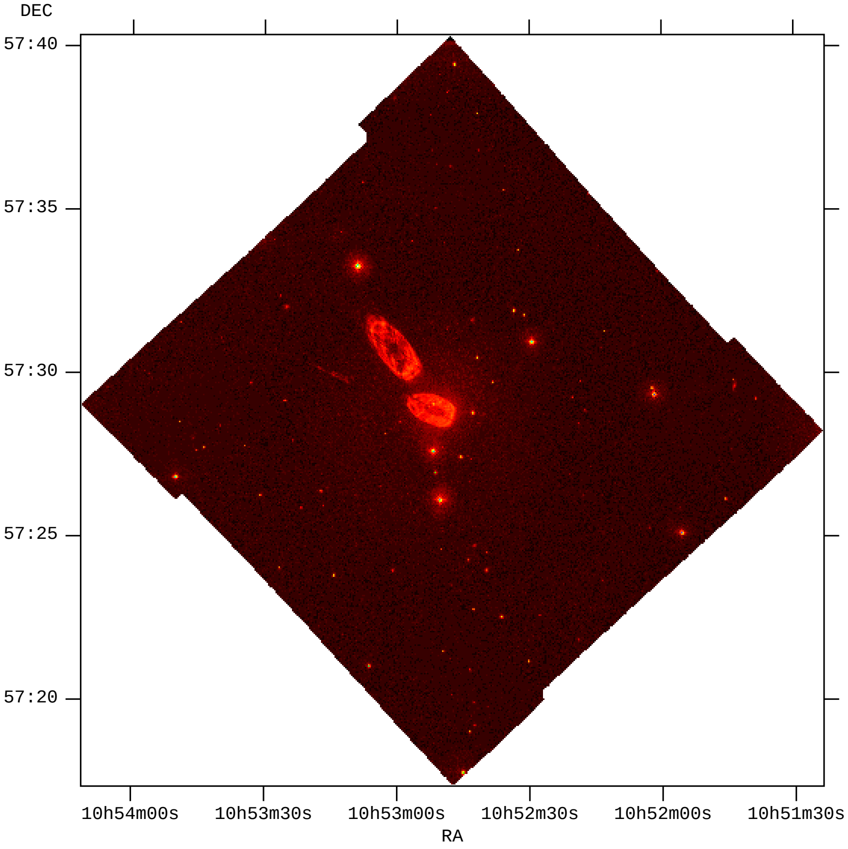
7.2 Rerunning the Pipeline

Throughout the OM section of this ABC Guide, public data from the Lockman Hole SV1 observation (Obs ID 0123700101) have been used to illustrate the SAS tasks. We suggest that the user download these data and retrace the following procedures. Figure 7.1 shows the merged V-band image from the Lockman Hole SV1 observation using the *ommosaic* task.

7.2.1 SAS Initialization

1) If you have not already done so, gunzip all gzipped files in the ODF and PPS directories. If necessary, rename all files in the ODF directory to upper case. This can be done using the script provided by the NASA/GSFC *XMM-Newton* GOF.

Figure 7.1: Merged OM image of the Lockman Hole SV1 observation obtained with the V filter. The image is displayed in logarithmic scale with an upper cut value of 20,000.



```
gunzip ODF/*.gz
gunzip PPS/*.gz
```

2) Set the SAS directory pointers. To verify the SAS-specific settings, use the command `env | grep SAS`. (For a detailed discussion of SAS initialization, see Chapter 4.)

```
setenv SAS_ODF full/path/to/ODF
setenv SAS_CCFPATH full/path/to/CCF
```

At this point, the SAS GUI can be run by typing `sas &` in the window where the pointers were set. However, since the next few procedures are very simple, it is faster to just use the command line.

3) If it doesn't already exist, create a CIF file in the ODF directory using the SAS task *cifbuild* (see §4.5.1). If a CIF file has previously been produced, it is only necessary to rerun *cifbuild* if the CCF has changed. Be sure to set the environment parameter afterwards.

```
cd ODF
cifbuild
setenv SAS_CCF /full/path/to/ODF/ccf.cif
```

4) If it hasn't already been done (don't do it twice), while still in the ODF directory, prepare the data by using the SAS task *odfingest* (see §4.5.2). It is only necessary to run it once on any data set (and will cause problems if it is run a second time). If for some reason *odfingest* must be rerun, you must first delete the earlier file produced by *odfingest* (`*SUM.SAS`). Again, remember to set the environment parameter afterwards.

```
odfingest
setenv SAS_ODF /full/path/to/file/full_name_of_*SUM.SAS
```

7.2.2 Prepare the Dataset

Now that the standard SAS initialization procedures *cifbuild* and *odfingest* are done, we must consider which files need to be processed. A good place to start is by grouping the ODF files by filter values. This can be done by using the Ftools task *fkeyprint* to print out the values for the header keyword `FILTER`:

```
fkeyprint odfile_name FILTER
```

The `FILTER` keyword in the initial ODF file is a number between 0 and 2100. The correspondence between number and filter value is given in Table 7.2.

Table 7.2: OM filter and file name correspondence.

File ID	Filter
1200	blocked
1400	V
1600	Magnifier
1800	U (no bar)
2000	B
0000	White (datum)
0200	Grism 2 (Optical)
0400	UVW1
0600	UVM2
0800	UVW2
1000	Grism 1 (UV)
2100	Bar

In general, the number following `_OMS` will either be of the form 00400, 00401, 00500... or 40100, 40101, 40200,.. The last two digits indicate the resolution. **00 is high-resolution and 01 is low-resolution**. In this example, the high-resolution window will be called 0070_0123700101_OMS004**00**IMI.FIT.gz while the low-resolution window will be 0070_0123700101_OMS004**01**IMI.FIT.gz. The low-resolution images for each of the five frames are taken consecutively to obtain the full FOV. For each low-resolution frame there is a high-resolution frame of the inner part of the detector.

Please be aware that you should **NOT** add low-resolution and high resolution images, even if they cover the same part of the FOV (that is, you cannot add `0070_0123700101_OMS00401IMI.FIT` and `0070_0123700101_OMS00400IMI.FIT`).

As this example, we will use the first high-resolution exposure for the Lockman Hole SV1 data. Since we will need other files associated with this exposure for processing, we will copy them over to our usual reprocessing directory PROC:

```
cp 0070_0123700101_OMS00*FIT ../PROC

cp 0070_0123700101_OMX00000*.FIT ../PROC

cp 0070_0123700101_SCX0* ../PROC
```

There should be eight files in all:

```
/XMM/PROC: ls

    0070_0123700101_OMS00400IMI.FIT 0070_0123700101_OMX00000PEH.FIT

    0070_0123700101_OMS00400THX.FIT 0070_0123700101_SCX00000ATS.FIT*

    0070_0123700101_OMS00400WDX.FIT 0070_0123700101_SCX00000SUM.ASC*
```

```
0070_0123700101_OMX00000NPH.FIT 0070_0123700101_SCX00000SUM.SAS
```

The file `0070_0123700101_SCX00000SUM.SAS` has been edited to point to that directory, SAS_ODF is also pointing to this directory, and SAS_CCF points to the file `ccf.cif` generated by *cifbuild* (§ 4.5.1).

We must now prepare the dataset we are interested in by processing it with the task *omprep.* This task will need to be run twice: once with the `THX` file as input, and again with the `IMI` file as input.

To prepare the data with *omprep* from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omprep set='0070_0123700101_OMS00400THX.FIT'
    pehset='0070_0123700101_OMX00000PEH.FIT'
    nphset='0070_0123700101_OMX00000NPH.FIT'
    wdxset='0070_0123700101_OMS00400WDX.FIT'
    outset='0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'
    modeset=0
```

where

`set` – Tracking History Data Auxiliary file

`pehset` – Periodic Housekeeping file

`nphset` – Non-Periodic Housekeeping file

`wdxset` – Window Data Auxiliary file

`outset` – Output file

`modeset` – specifies if run in imaging mode (0), fast mode (1), slew mode (2), or tracking mode (3)

2) And do the same for the `IMI` file:

```
omprep set='Mydata/0070_0123700101_OMS00400IMI.FIT'
    pehset='0070_0123700101_OMX00000PEH.FIT'
    nphset='0070_0123700101_OMX00000NPH.FIT'
    wdxset='0070_0123700101_OMS00400WDX.FIT'
    utset='0070_0123700101_OMS00400IMI_OUT_OMPREP.FIT'
    modeset=0
```

The output THX file is ready to be used by the rest of the SAS tasks.

7.2.3 Examine the Tracking History

There are two ways to examine the tracking history of the OM: by inspecting the postscript file output of the task *omdrifthist*, or by looking at the count rates of the guide stars with *omthconv.* This second task will produce a FITS file containing up to 10 columns with the guide stars' count rates.

To check the OM tracking using *omdrifthist* from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omdrifthist set='0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'
    plotfile='0070_0123700101_OMS00400THX_drift.ps'
    trackradius=0.5 hardcopy=yes pages='1 2'
```

where

`set` – THX file output from the *omprep* task

`plotfile` – name of output postscript file

`trackradius` – radius of pointing accuracy

`hardcopy` – produce hardcopy? yes/no

`pages` – Pages to plot (maximum pages produced is 2)

To check the OM tracking using *omthconv* from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omthconv thxset='0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'
    nphset='0070_0123700101_OMX00000NPH.FIT'
    outset='THX_trackingStar.FIT'
```

where

`thxset` – THX file output from the *omprep* task

`nphset` – name of Non Periodic Housekeeping file

`outset` – Output file

7.2.4 Detecting Bad Pixels

The task *omcosflag* looks at the (processed) OM tracking history and applies it to the map of bad pixels defined in the CCF. The resulting new bad pixel map is then used by the source detection algorithms. Bad pixels are set to 1, good pixels are set to 0.

To generate a bad pixel map on the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omcosflag samplefactor=1 timefactor=1
    set='0070_0123700101_OMS00400IMI_OUT_OMPREP.FIT'
    thxset='0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'
```

where

`samplefactor` – Spatial oversampling factor (default 1)

`timefactor` – Temporal sampling factor (default 1)

`thxset` – Corrected THX file (output from the first *omprep* task)

`set` – Corrected IMI file (output from the second *omprep* task)

7.2.5 Generate the Flat Field

OM flat field generation is implemented in the *omichain* command, but there is no flat field generation in the OM pipeline. Instead, users can run the task *omflatgen* to produce a unit flatfield, followed by *omflatfield*, which creates a tracking-shifted flatfield and applies it to an OM Science Window (OSW) Image.

The *omflatfield* task creates **two** output files: one is the actual image and the other (specified by the output parameter `ppsflatset`) contains the tracking-shifted version of the `omflatgen` file.

To generate and apply a flat field from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omflatgen outset='OUT_FLATGEN.FIT'
```

where

`outset` – Name of the output file

2) The output from *omflatgen* is then used as input for *omflatfield*.

```
omflatfield samplefactor ='1'
    set='0070_0123700101_OMS00400IMI_OUT_OMPREP.FIT'
    thxset= '0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'
    inorbitflatset='OUT_FLATGEN.FIT'
    tsflatset='0070_0123700101_OMS00400PPSFLATSET.FIT'
    outset='0070_0123700101_OMS00400IMI_OUT_FLATFIELD.FIT'
```

where

`samplefactor` Sampling factor (to be set to 1)

`set` – Corrected IMI file (output of the *omcosflag* task)

`thxset` – Corrected THX file (output of the *omprep* task)

`inorbitflatset` – Unit file (Output of the *omflatgen* task)

`tsflatset` – Output name for the tracking history flatfield

`outset` – Output name for the flat field image

7.2.6 Correct for Fixed-Pattern Noise

The task *ommodmap* corrects a given OM Science Window (OSW) image for "modulo-8" spatial fixed-pattern noise that results from the OM centroiding algorithm performed by the on-board electronics (see documentation at "$SAS_PATH/doc/ommodmap/ommodmap.html" for more details).
Note that the *ommodmap* task does not lose counts, it simply redistributes them.

To correct for fixed-pattern noise from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
ommodmap set='0070_0123700101_OMS00400IMI_OUT_FLATFIELD.FIT'
    mod8set='0070_0123700101_OMS00400PPSMODE8SET_OUT.FIT'
    outset='0070_0123700101_OMS00400OUT_OMMODMAP.FIT'
    nsig=3 nbox=16 mod8product=yes
```

where

`set` – Input file (output of *omflatfield*)

`mod8product` – Produce a Pipeline Processing System (PPS) file?

`mod8set` – Name of the output modulo-8 tile

`outset` – Name of the corrected image

`nsig` – Significance level for sigma clipping

`nbox` – Size of the sliding box in units of 8 pixels

7.2.7 Perform Source Detection

The task *omdetect* uses two different algorithms for detecting point and extended sources; they are discussed in detail at `http://xmm2.esac.esa.int/sas/7.1.0/doc/omdetect/node3.html` The task has a lot of parameters (see below) but only `set` and `outset` are mandatory.

To detect sources with *omdetect* from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omdetect set='0070_0123700101_OMS00400OUT_OMMODMAP.FIT'
    outset='0070_0123700101_OMS00400IMI_OUT_OMDETECT.FIT'
    levelimage='0070_0123700101_OMS00400LEVELIMAGE.FIT'
    regionfile='0070_0123700101_OMS00400oswList.reg'
    wdxset='0070_0123700101_OMS42200WDX.FIT'
    backgroundimage='0070_0123700101_OMS00400BKGIMAGE.FIT'
    minsignificance=0 detectextended=no nsigma=6
```

where

`set` – Input file (output of *ommodmap*)

`outset` – Name of the output source list file

`nsigma` – Number of σ above background for a detection

`wdxset` – name of WDX file (needed for images produced before SAS V5.4)

`backgroundimage` – Name of output background image file

`levelimage` – Name of image of island detections

`minsignificance` – minimum significance of sources to be included in OM OSW file. If 0, it is not used

`regionfile` – Name of saoimage region file

`detectextended` – use algorithm for detecting extended sources? yes/no

7.2.8 Convert Source Counts to Magnitudes

The task *ommag* converts the list of given source counts to magnitudes in the appropriate instrumental band passes.

To compute instrumental magnitudes with *ommag* from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
ommag set='0070_0123700101_OMS00400IMI_OUT_OMDETECT.FIT'
```

where

`set` – OM OSW source list file name (output of *omdetect*)

There is a “recipe” to convert the UV count rates to flux. It was provided by Alice Breeveld (MSSL) and can be accessed at:
`http://xmm.esac.esa.int/sas/7.0.0/watchout/Evergreen_tips_and_tricks/uvflux.shtml`

7.2.9 Convert Source OM Positions to Sky Coordinates

The task *omatt* converts an OM OSW source list from pixels to sky coordinates. These sky coordinates are then used to produce a sky coordinate image.

To compute sky coordinates with *omatt*from the command line:

1) In the window where SAS was initialized, and in the directory where you want the output to go, type the following:

```
omatt set=’0070_0123700101_OMS00400OUT_OMMODMAP.FIT’
    sourcelistset=’0070_0123700101_OMS00400IMI_OUT_OMDETECT.FIT’
    ppsoswset=’0070_0123700101_OMS00400FINAL_IMAGE.FIT’
    usecat=no catfile=’’ maxradecerr=1.0 maxrmsres=1.5 rotateimage=no
```

where

`set` – Input file (output of the *ommodmap* task)

`sourcelistset` – Source list (output of *omdetect* task)

`ppsoswset` – Output name for the corrected sky image

`usecat` – Do you want to use the USNO-SA 1 catalog?

`catfile` – Name of the USNO star catalog (default: ’usnocat.fit’)

`maxradecerr` – Maximum allowed RA/dec error in astrometry fit

`maxrmsres` – Maximum allowed rms residual in astrometry fit

`rotateimage` – create rotated sky image?

Due to the large size of the catalog, it is not distributed. Users, however, can provide their own catalog if they wish. The format is that used for the USNO cross-correlation FITS products. In general, the `usecat` keyword should be set to `no`.

The pointing stability about the spacecraft boresight position is better than 1″ (look at the tracking plots derived at the beginning). There is still a scatter of about 4″ between the planned and actual pointing position.

7.2.10 Fast Mode

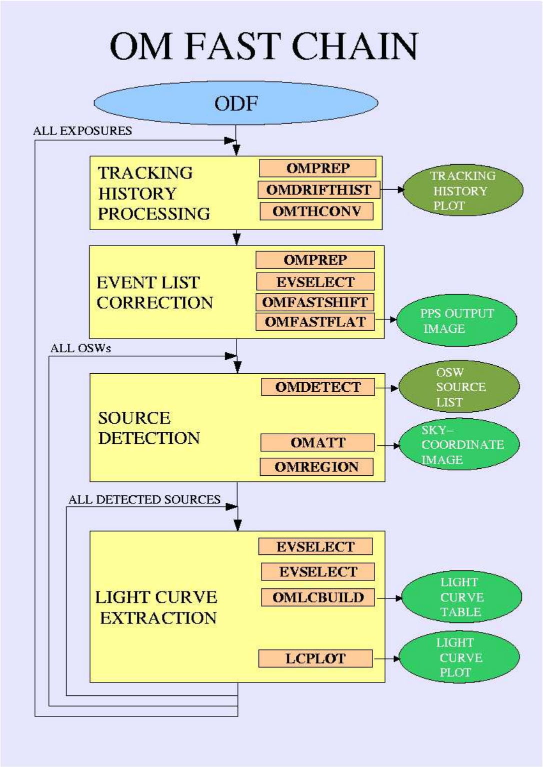
SAS has a working fast mode pipeline. If the data have not been processed by the latest version of SAS, the task *omfchain* should be run.

The chain works similarly to the imaging chain explained above, and consists of a Perl script which calls all the necessary tasks sequentially. It produces images of the detected sources, extracts events related to the sources and the background, and extracts the corresponding light curves. A more detailed description of the chain can be found in the SAS on-line help available at `http://xmm.esac.esa.int/sas/current/doc/index.html`. A summary of the task is shown in Figure 7.2.

7.2.11 Grism Analysis

The metatask *omgchain*, can be used to extract and automatically calibrate spectra produced by the OM grisms. OM grism data are taken in Image Mode. Hence *omgchain* uses already existing tasks, such as *omprep* and *ommodmap*, to handle housekeeping information and to perform some corrections (the “modulo-8” noise reduction for example). Also, *omdetect* is designed to find the spectra, zero and first orders, producing a source list. Other tasks are grism specific. *omgprep* is used to correct for geometric distortion of the detector and to rotate the image so as to have the dispersion direction aligned with the image Y axis. *omgprep* performs the spectral extraction and the wavelength and flux calibration. Finally, the extracted spectra are plotted using *omgplot*.

Figure 7.2: OM fast chain–diagram of the different tasks run.



The sequence of tasks used by *omgchain* is illustrated in Fig. 7.3. An output spectrum produced by *omgchain* is given in Fig. 7.4. Each of these tasks can be run individually. SAS V7 also includes a new interactive task, *omgsource*, which allows the user to select with the cursor the spectrum to be extracted.

The task *omgchain* has many parameters, but none of them are mandatory. *Omgchain* will search for grism images in the working directory; if at least one such image is found, it will process it. If none are found, the task will produce no output.

Below is a description of the calling sequence and the individual parameters.

```
omgchain inpdirectory=../ODF outdirectory=PROC comment=’’

   nsigma=3 combine=yes spectrumhalfwidth=-8 bkgoffsetleft=0 bkgwidthleft=-8

   bkgoffsetright=0 bkgwidthright=-8 spectrumsmoothlength=0 mod8correction=1

   extractionmode=0 plotbinsize=1 plotflux=2 scalebkgplot=no
```

where

`inpdirectory` – Input file directory

`outdirectory` – Output file directory

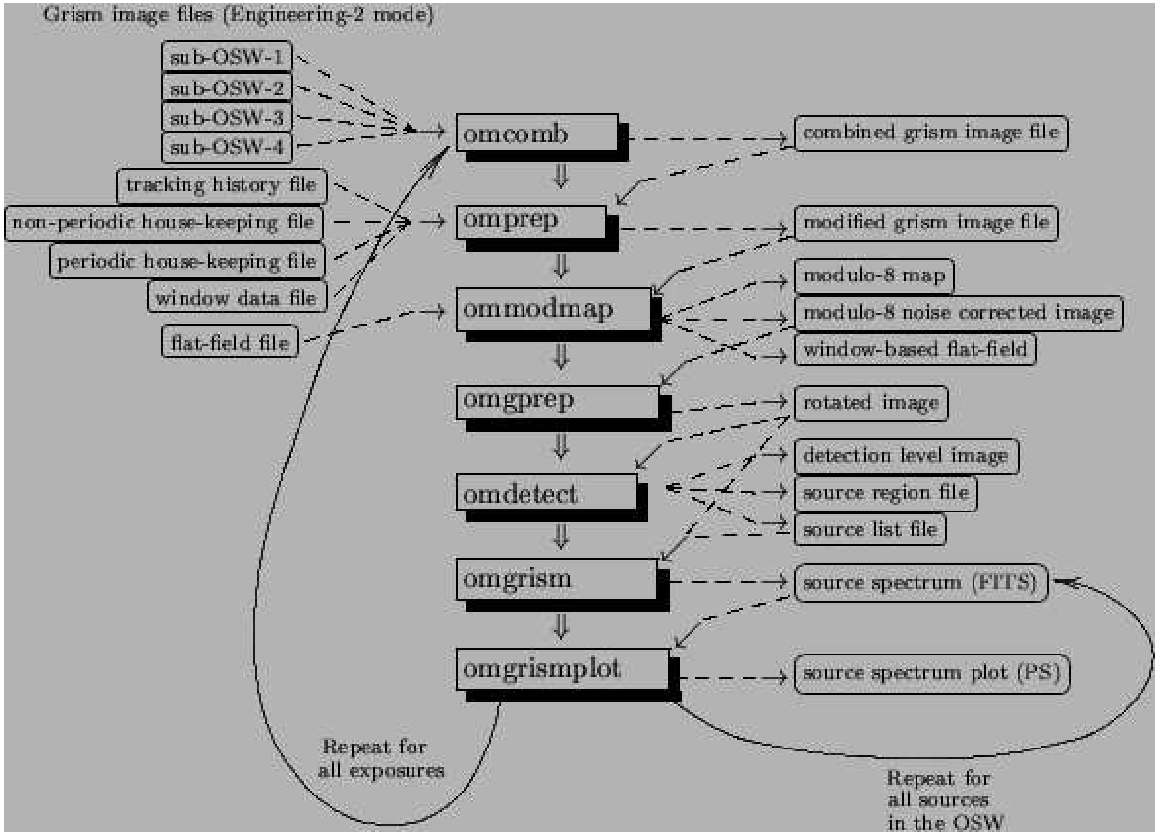
`comment` – User’s comments for output

`nsigma` – Number of σ above the background required for a detection (this parameter is passed to *omdetect*

`combine` – Condition for combining the Engineering-2 subwindows

`spectrumhalfwidth` – Half-width of the spectrum extraction region in pixels, if negative, and in FWHMs otherwise

Figure 7.3: Diagram of the different tasks used by *omgchain*. The first four tasks are preparatory, the other three tasks execute the source detection and spectral identification procedure, and produce the output files.



`bkgoffsetleft` – Offset of the left background extraction region from the edge of the spectrum extraction area; in pixels, if negative, or in FWHMs otherwise.

`bkgwidthleft` – Width of the left background extraction region; in pixels, if negative, or in FWHMs otherwise

`bkgoffsetright` – Offset for the right background extraction region; in pixels, if negative, and in FWHMs otherwise

`bkgwidthright` – Width of the right background extraction region; in pixels, if negative, or in FWHMs otherwise.

`spectrumsmoothlength` – Length of the smoothing window for smoothing the extracted spectra, if necessary. Values 0 or 1 of this parameter imply no smoothing

`mod8correction` – Condition for removing the modulo-8 noise: 0: correction not applied; 1: correction applied using the modulo-8 map extracted from the input image; 2: correction applied using the modulo-8 map extracted from the OM CCF flat field; 3: correction applied multiplying the input image by the OM CCF flat field

`extractionmode` – Switch between different extraction modes. The value 0 corresponds to the normal extraction (summation of counts in the cross-dispersion direction); 1 corresponds to the Optimal Extraction; 2 corresponds to the spline smoothing; 3 corresponds to the Gaussian fit

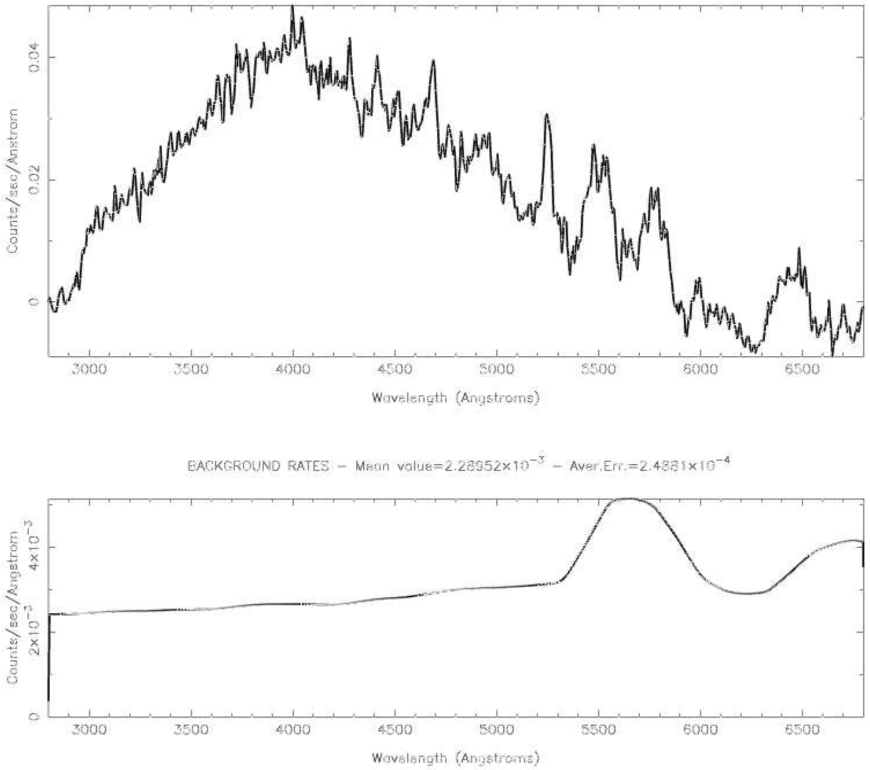
`extractfieldspectra` – Condition for extraction either only the target object spectrum or all available spectra of the sources in the field

`plotbinsize` – Size of spectrum wavelength bins for the output plot (in Å) `plotflux` – Flag for plotting the spectrum only (value 0), the background only (value 1), or both of them (value 2)

`scalebkgplot` – Condition for scaling the background plot differently from the spectrum plot

If if a source is not detected by *omdetect*, or does not fall within the grism window, *omgchain* will run nonetheless without giving any warnings, but will not produce output files.

Figure 7.4: OM optical grism spectrum obtained from a 4.7 ks observation of Mrk 478.



7.3 In a Nutshell

To summarize, the following steps are needed to process OM data:
1) Obtain the raw and pipeline data.
2) Initialize SAS.
3) Make the CCF and ODF summary file (run the *cifbuild* and *odfingest* tasks).
4) Rerun the appropriate pipline chain (*omichain*, *omgchain*, or *omfchain*). The basic steps of *omichain* have been described above.