THE XMM-NEWTON ABC GUIDE

AN INTRODUCTION TO XMM-NEWTON DATA ANALYSIS

NASA/GSFC XMM-Newton Guest Observer Facility

Steve Snowden, Stefan Immler, Michael Arida,
Brendan Perry, Martin Still, and Ilana Harrus

Version 2.01

23 July 2004

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.vilspa.esa.es/`

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

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

- PPARC Information Center, source of UK specific information and mirror site for some software and data access:

  `http://www.xmm.ac.uk/`

- 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.vilspa.esa.es/external/xmm_user_support/helpdesk.shtml`

  or via e-mail:

  `xmmhelp@xmm.vilspa.esa.es`

  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@athena.gsfc.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://xmm.vilspa.esa.es/external/xmm_obs_info/obs_stat_log.shtml`

  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.vilspa.esa.es/external/xmm_sched/advance_plan.shtml`

2.4 PUBLIC DATA ARCHIVES

- SOC Public Data Archive via the XSA:

  `http://xmm.vilspa.esa.es/external/xmm_data_acc/xsa/index.shtml`

- 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.vilspa.esa.es/external/xmm_sw_cal/calib/index.shtml`

- 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.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`

- 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.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`

  (click "Documentation", then click "its own documentation", then click "The SAS User's Guide"

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

  `http://xmm.vilspa.esa.es/sas/documentation/watchout/`

- *XMM-Newton* Users Handbook:

  `http://xmm.vilspa.esa.es/external/xmm_user_support/documentation/uhb_frame.shtml`

- This Guide:

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

- The MPE Analysis Guide:

`http://wave.xray.mpe.mpg.de/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.vilspa.esa.es/external/xmm_user_support/documentation/index.shtml`), and the *SAS Users Guide* (`http://xmm.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`).

Additional information concerning *XMM-Newton* data files can be found in the *Interface Control Document: Observation and Slew Data Files (XSCS to SSC) (SciSIM to SOCSIM)* (XMM-SOC-ICD-0004-SSD). This is an impressive tome which goes into great detail about the file nomenclature and structure. This document can be found in the documents area of the SOC web pages:
`http://xmm.vilspa.esa.es/cgi-bin/docs/DOC_list?Type=ICD`.

NOTE: For observation data sets going to US PIs, the GOF makes the data available on line 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

NOTE: 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 (§ 3.3.2).

3.3 PI Data

Proprietary *XMM-Newton* data is available for download via your XSA account. Email instructions from 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 PP data contain, among other things, calibrated photon event files and source lists.

NOTE: 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
- PIPEPROD – The pipeline processed data products

3.3.1 ODF Data

ODF data come with file names in the following format:

- `mmmm_iiiiiijjkk_aabcccddeee.FIT`

  `mmmm` – orbit number

  `iiiiiijjkk` – observation number

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

  `b` – S for scheduled data (U for unscheduled data, X for general purpose files)

  `ccc` – exposure number

  `dd` – CCD number or OM window number

  `eee` – type of data

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

PP data (listed in Table 3.1) contain some more 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. (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.) 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 3.1)

Table 3.1: Pipeline Processing data files.

Group ID	Contents
CRSCOR[1]	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[1]	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[1]	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[1]	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[1]	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.
OIMAGE[2]	Contains OM sky images in gzipped FITS format.
OMSLIS[2]	Contains OM observation source lists in gzipped FITS format.
OMSRTS[2]	Contains OM star tracking time series in gzipped FITS format.
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
REVLIS[3]	Contains the RGS source and event lists in gzipped FITS format
REXPIM[3]	Contains the RGS exposure maps in gzipped FITS format
RIMAGE[3]	Contains the RGS images (both energy dispersion and cross dispersion) in gzipped FITS and PNG formats
RSPECT[3]	Contains the RGS source and background spectra in gzipped FITS and PDF formats

[1]Further information on the files can be found in Table 5.1.
[2]Further information on the files can be found in Table 7.1.
[3]Further information on the files can be found in Table 6.1.

Chapter 4

Setting Up and Running SAS

The Science Analysis Software (SAS, `http://xmm.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml`), 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://xmm.vilspa.esa.es/external/xmm_sw_cal/sas_frame.shtml` (note that the final "/" is often required for SOC pages). 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://xmm.vilspa.esa.es/sas/installation/requirements.shtml`.

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://xmm.vilspa.esa.es/ccf/`). The calibration page also has links to the CCF release notes. In addition, background event files and canned spectral response files can be found under `http://xmm.vilspa.esa.es/external/xmm_sw_cal/calib/epic_files.shtml`.

4.3 SAS INVOCATION

There are a few parameters which need to be set for the proper operation of SAS. Many are taken care of by the initialization script, but it doesn't hurt to repeat them. The commands should, of course, be modified to be appropriate for your specific setup.

```
setenv SAS_DIR /path/to/xmmsas_yyyymmdd_hhmmsource
```

```
                                     Sets the SAS directory path
source $SAS_DIR/sas-setup.csh        Initializes SAS
       $SAS_DIR/sas-setup.sh         Alternate SAS initialization
setenv SAS_ODF /path/to/odf_data     Sets the directory path to the ODF data, it is
                                        probably a good idea to have this be the
                                        full path.
setenv SAS_CCFPATH /path/to/CCF      Sets the directory path to the CCF data
setenv SAS_CCF $SAS_ODF/ccf.cif      Sets the Calibration Index File (CIF) path and
                                        file name (note that the CIF file is normally
                                        part of an event list, so SAS_CCF can also
                                        be pointed at the list, this should probably
                                        be the full path as well
setenv SAS_VERBOSITY 3               Sets the verbosity, 1 => little, 10 => lot
setenv SAS_SUPPRESS_WARNING 3        Sets the warning level, 1 => little, 10 => lot
sas &                                Invokes the SAS GUI, SAS tasks can also be run
                                        on the command line
```

NOTE: To verify the SAS-specific settings, use the task `sasversion` (alternatively, the command `env | grep SAS` can be used).

SAS need not be run in the directory where the data are stored (for example, it will be possible to run off of the data CDs when the file names are changed to be upper case). To do so only requires that the `setenv SAS_CCF $SAS_ODF/ccf.cif` be reset to the directory string for the working directory, see § 4.5.1. This can also be done using the SAS *Preferences* GUI (found under the "File" menu). From the command line invocation of tasks the input and output directories, when relevant, can be set as parameters (e.g., see command line input for *odfingest*, § 4.5.2).

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 most tasks from the command line. In many cases parameters where the default values are acceptable are not included in the command list, which can be done in practice as well. If the GUI interface is being used then simply set the parameters there.

The MPE Analysis Guide, `http://wave.xray.mpe.mpg.de/xmm/data_analysis` 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 *Netscape* window. 3) In addition, the command `sashelp doc=`*sas_task* will pop up a *Netscape* window with the documentation for the task *sas_task* as well.

NOTE: 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 GENERAL SAS TASKS FOR DATA SET PREPARATION

WARNING: Before running the following tasks make sure that the ODF file names are all upper case.
NOTE: Run these tasks.

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 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 § 5.7.1). To help determine whether it is reasonable to reprocess the data, the CCF release notes (`http://xmm.vilspa.esa.es/user/calib_top.html`) should be examined.

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

WARNING: 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.
To run *cifbuild* and *cifdiff* on the command line use:

- `cifbuild withccfpath=no analysisdate=now category=XMMCCF fullpath=yes`
  - > `withccfpath` – flag to look for the CCF constituents in a specific directory (the parameter SAS_CCFPATH should be set, see § 4.3)

  - `analysisdate` – date when analysis was performed.
  - `category` – XMMCCF (SCISIMCCF if data were constructed by the SciSim simulation package).
  - `fullpath` – include the full path to each constituent within the CIF.

- `cifdiff calindex1set=ccf.cif calindex2set=CALIND.FIT`

  - `calindex1set` – name of the first file to be compared, in this case the output from the current run of *cifbuild*
  - `calindex2set` – name of the second file to be compared, in this case the (renamed) PP file

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.
To run *odfingest* on the command line use:

- `odfingest odfdir=$SAS_ODF outdir=$SAS_ODF`

  - `odfdir` – ODF directory
  - `outdir` – directory to deposit summary file (the ODF directory in this case although the user might want to deposit the summary file in the working directory)

Chapter 5

First Look – An EPIC Data 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 can start with the Pipeline Processed data. 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 $\sim 6.3\%$ 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`

5.1 USING PIPELINE PROCESSED DATA PRODUCTS

The Pipeline Processing (PP) 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`, where

  `iiiiii` – proposal number

  `jj` – target ID - target number in proposal

  `kk` – exposure ID - exposure number for target

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

  `AA` – ID (EP – EPIC Summary, OM – Optical Monitor Summary, RG – RGS Summary OB – Observation Summary, OB with SUMMAR − > PPSSUM – Pipeline Processing Summary, CA with SUMMAR − > XCORRE – Source Correlation Summary)

Each grouping of the pipeline products (Tables 3.1 and 5.1) there is an HTML (.HTM extension) file which lists the associated files and gives a few-word description of those files. It is useful to set up your web browser to automatically display a number of file types, e.g., PDF files. The HTML file names are of the following format:

- `PPiiiiiijjkkAAAAAA000_0.HTM`, where

  `iiiiii` – proposal number

`jj` – target ID - target number in proposal

`kk` – exposure ID - exposure number for target

`AAAAAA` – Group ID (Table 5.1)

The data file names are of the form (see Table 41 in the *XMM Data Files Handbook*, `ftp://xmm.vilspa.esa.es/pub/odf/data/sv/docs/datafileshb_2_0.pdf.gz`, or *.ps.gz):

- `PiiiiiijjkkaablllCCCCCCnmmm.zzz`, where

  `iiiiiijjkk` – observation number

  `aa` – detector, `M1` – `MOS1`, `M2` – `MOS2`, `PN` – `PN`, `CA` – for files from the CRSCOR group

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

  `lll` – exposure number

  `CCCCCC` – file identification (Table 5.1)

  `n` – exposure map band number, unimportant otherwise for EPIC data

  `mmm` – source number in hexidecimal

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

  `ASC` – ASCII file, use *Netscape*, other web browser, or the "more" command
  `ASZ` – gzipped ASCII file
  `FTZ` – gzipped FITS format, use *ds9, Ximage, Xselect, fv*
  `HTM` – HTML file, use *Netscape* or other web browser
  `PDF` – Portable Data Format, use *Acrobat Reader*
  `PNG` – Portable Networks Graphics file, use *Netscape* or other web browser
  `TAR` – TAR file

5.1.1 A Quick Look at What You Have

The ESKYIM files contain EPIC sky images in different energy bands whose ranges are listed in Table 5.1. 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 *Netscape*.

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

5.2 EXAMINE AND FILTER THE DATA - PIPELINE PRODUCTS

The EPIC event lists in the EEVLIS group of the Pipeline Processing will have names of the form:

- `PiiiiiijjkkaaSlllcIEVLI0000.FTZ`, where

  `iiiiiijjkk` – observation number

  `aa` – detector (`M1` – MOS1, `M2` – MOS2, `PN` – PN)

  `lll` – exposure number within the observation

  `c` – detector (`M` – MOS1 or MOS2, `P` – PN, `T` – Timing Mode)

These are OGIP standard calibrated photon event FITS files in gzipped format. Some tasks and software will require that these files be gunzipped, which usually means renaming as well, e.g.:

```
mv PiiiiiijjkkaaSlllcIEVLI0000.FTZ PiiiiiijjkkaaSlllcIEVLI0000.FIT.gz
gunzip PiiiiiijjkkaaSlllcIEVLI0000.FIT.gz
```

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. 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.

Table 5.1: EPIC Pipeline Processing data files.

Group ID	File ID	Contents	File Type	View With
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, Netscape*
	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*
	EXSNMP	Exposure sensitivity map	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, Netscape*

[1] NNNNN – Alphanumeric ID
[2] Files for only those modes which were active will be included

5.2.1 Initialize SAS and Prepare the Data

For the following demonstration, the PP data are assumed to be in the /PIPE 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. The data used are from the Lockman Hole SV1 observation.

1) gunzip the PP event list to be examined (not really necessary), and for practical purposes shorten the file name as well, e.g.:
```
mv P0123700101M1S001MIEVLI0000.FTZ mos1.fits.gz
gunzip mos1.fits.gz
```

2a) In preparation set a few SAS parameters (directory pointers):
```
setenv SAS_ODF /ODF
setenv SAS_CCFPATH /CCF
setenv SAS_CCF /PROC/ccf.cif
```

To verify the SAS-specific settings, use the command `env | grep SAS`, and remember that for SAS_ODF and SAS_CCF it is best to use the full path

2b) If it doesn't already exist, create a CIF file using the SAS task *cifbuild* (§ 4.5.1).

- `cifbuild fullpath=yes`

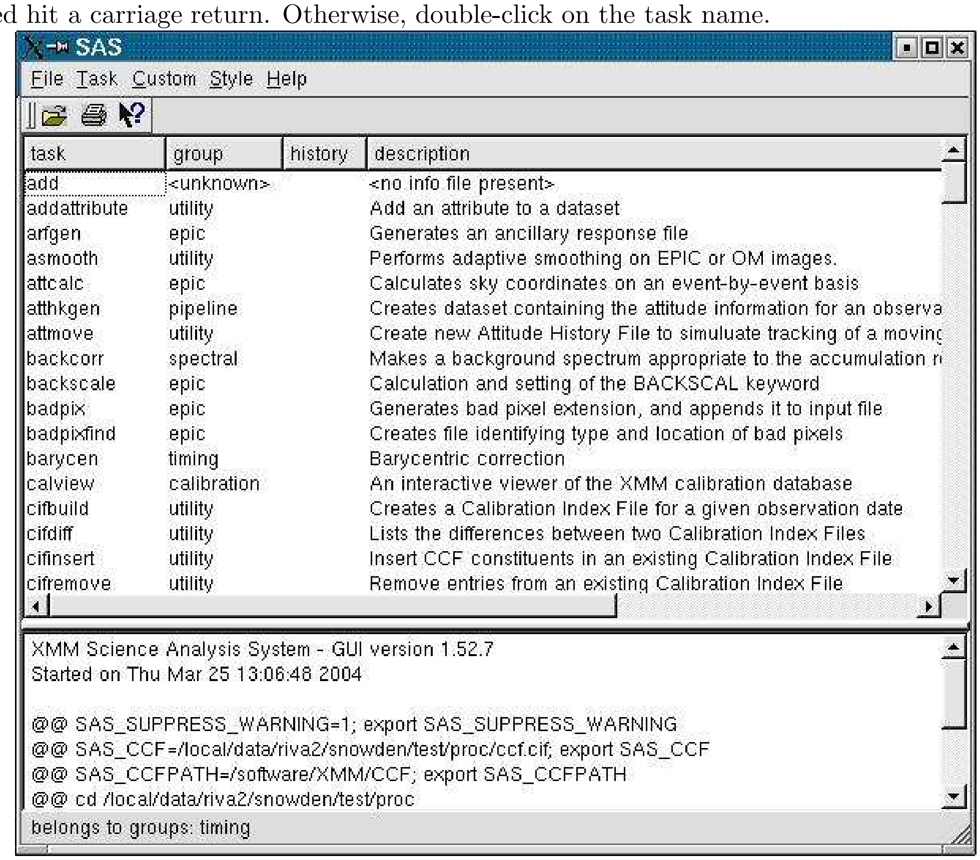
2c) If it hasn't already been done (don't do it twice), prepare the ODF data by using the SAS task *odfingest* (necessary for many SAS tasks) (see § 4.5.2).

- `odfingest odfdir=$SAS_ODF outdir=$SAS_ODF`

3) Invoke the *SAS* GUI (Figure 5.1).

- `sas &`

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

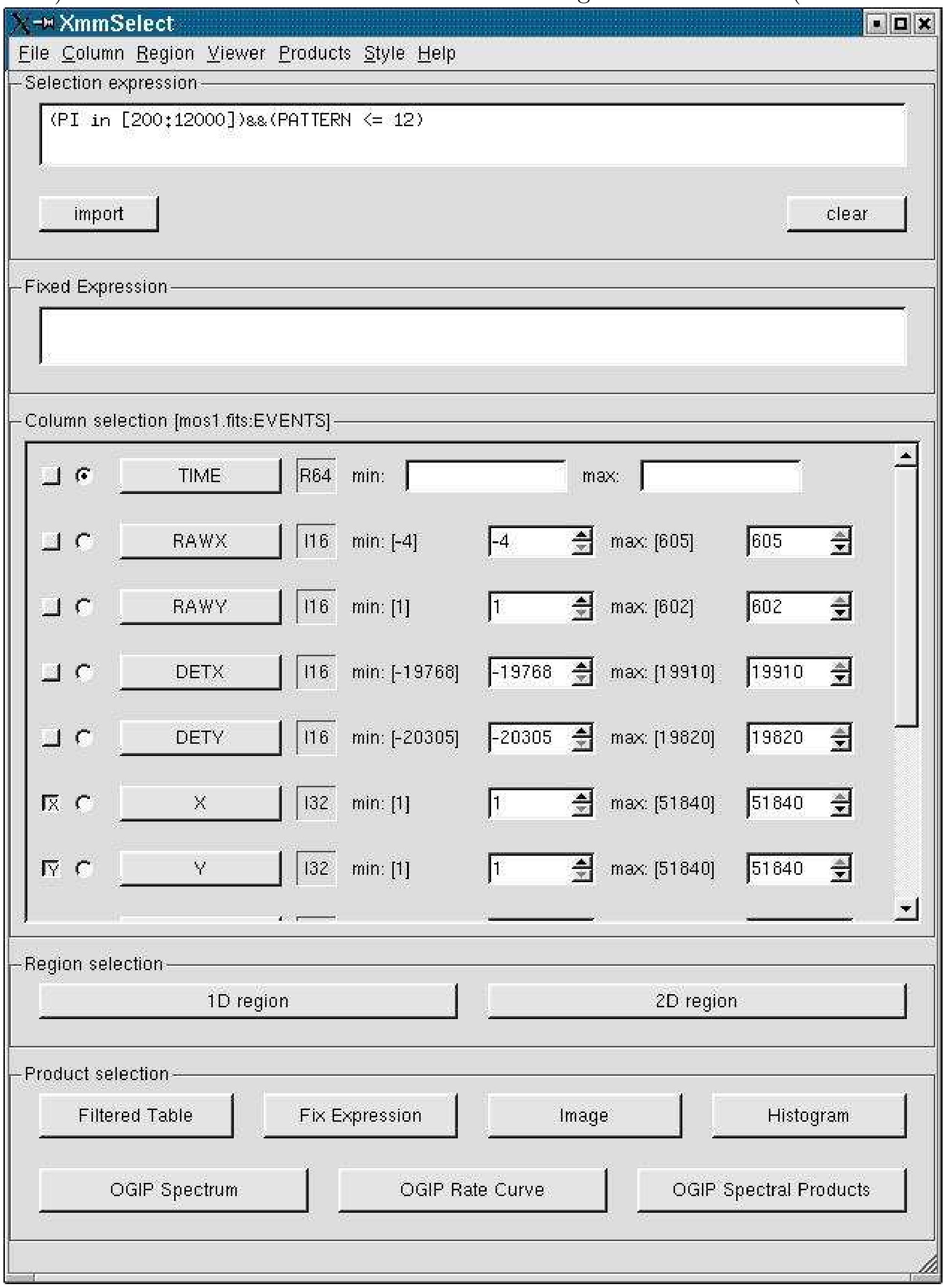


4) Invoke the *xmmselect* GUI (Figure 5.2) from the *SAS* GUI. To invoke a task one need only start typing the task name, and when it is high-lighted hit a carriage return.

- 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, "mos1.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.

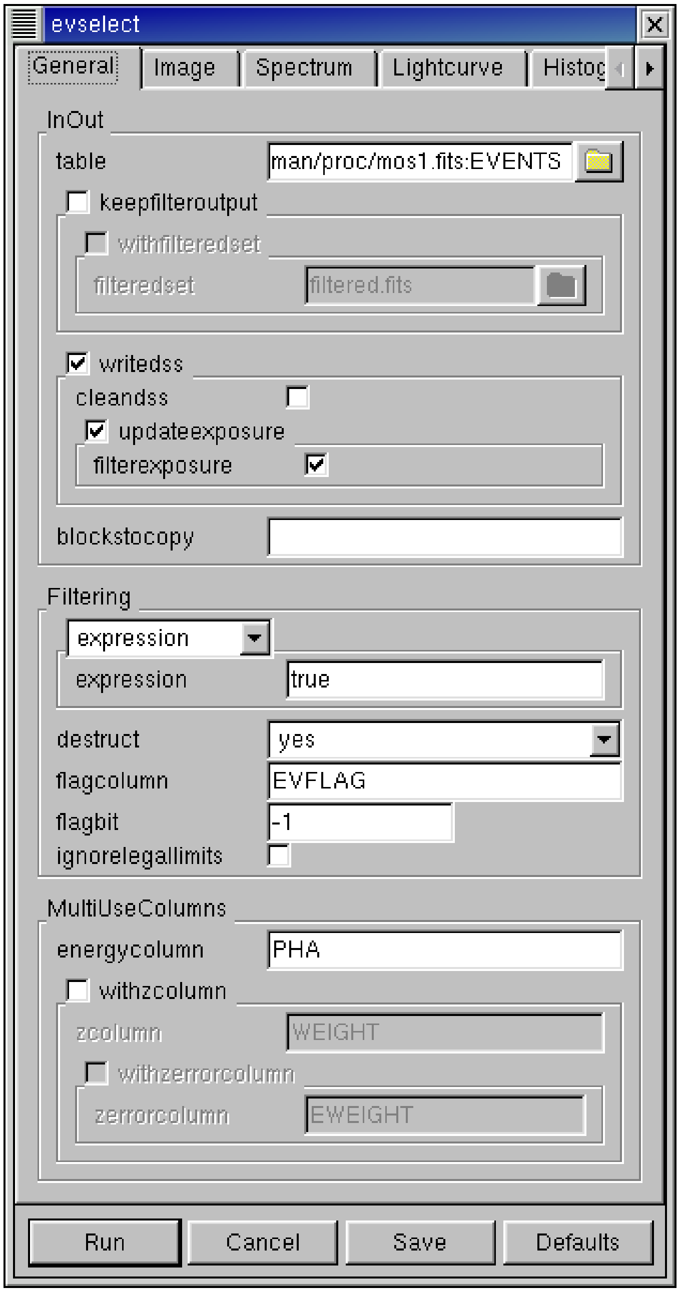
Figure 5.2: The *xmmselect* GUI. The top dialog area is for the selection expression. The central part of the GUI provides a list of the parameters available in the table (note the scroll bar on the right hand side). Two-dimensional data are selected using the square boxes on the left hand side (in this case X,Y, sky coordinates, have been selected) while one-dimensional data are selected using the round boxes (Time in this example).



5.2.2 Create and Display an Image

5a) Create an image in sky coordinates by using the *xmmselect* GUI.

Figure 5.3: The *evselect* GUI. Additional parameters for the selected process can be accessed through the tabs at the top of the GUI.



– To create an image of the data in sky coordinates check the square boxes to the left of the “X” and “Y” entries.

– Click on the “Image” button near the bottom of the page. This brings up the *evselect* GUI (Figure 5.3).

– The default settings are reasonable for a basic image so click on the “Run” button at 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.

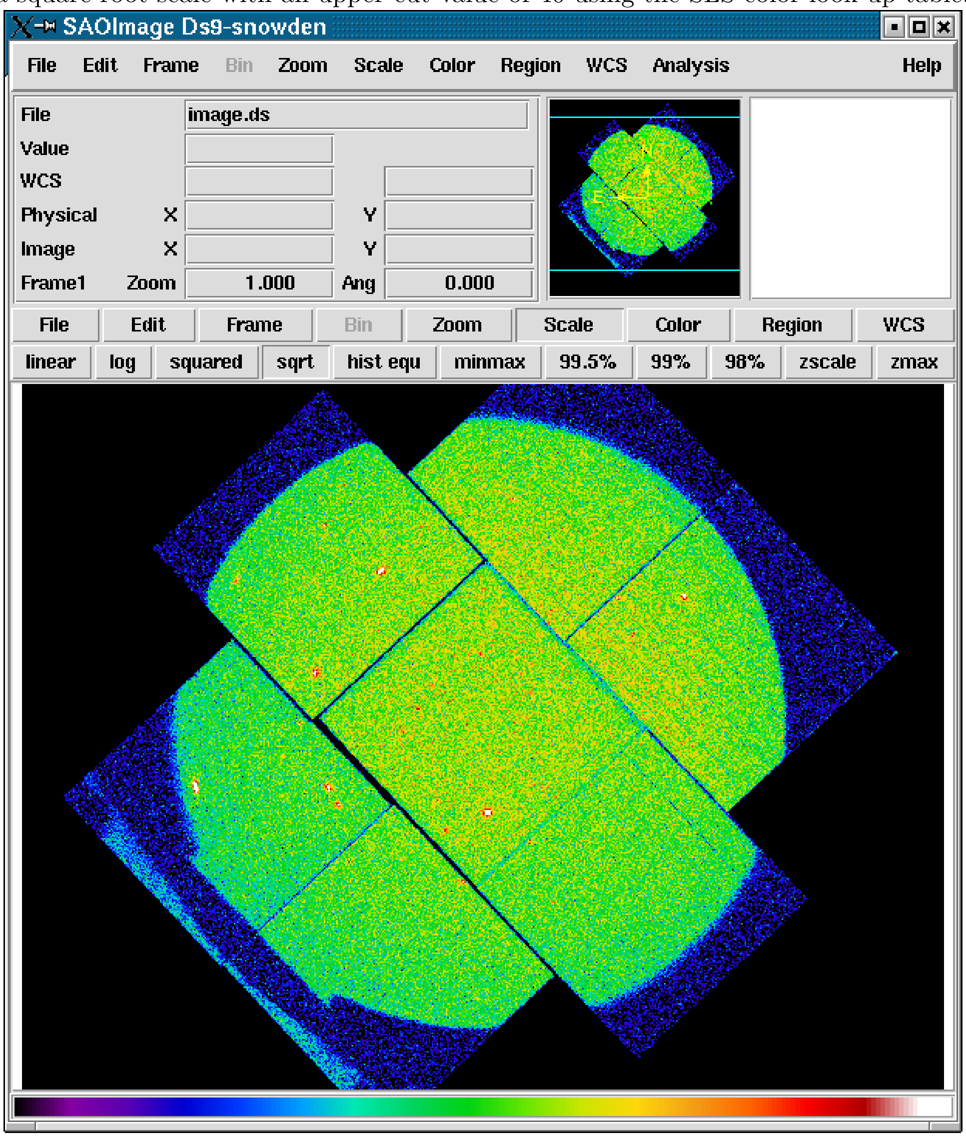
– The resultant image is written to the file `image.ds`, and the image is automatically displayed using *ds9*, and is shown in Figure 5.4.

5b) Using the command line interface, create an image in sky coordinates by using the task *evselect*. The same image produced in 5a) can be created using the following command.

– `evselect table=/PIPE/mos1.fits:EVENTS withimageset=yes imageset=image.fits xcolumn=X ycolumn=Y imagebinning=imageSize ximagesize=600 yimagesize=600`
  - > `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.
  - > `ximagesize` – output image pixels in Y.

Display the output file `image.fits` using, e.g., `ds9 image.fits &`.

Figure 5.4: *ds9* window showing the unfiltered image of the MOS1 data from the Lockman Hole SV1 observation, displayed on a square root scale with an upper cut value of 40 using the SLS color look-up table.



5.2.3 Create and Display a Light Curve

6a) Create a light curve of the observation by using the *xmmselect* GUI (Figure 5.2).

- To create a light curve check the round box to the left of the "Time" entry.
- Click on the "OGIP Rate Curve" button near the bottom of the page. This brings up the *evselect* GUI (Figure 5.3).
- The default setting is for a one-second bin which is a bit fine, so access the "Lightcurve" tab and change the "timebinsize" to, e.g., 100 (100 s). 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 (Figure 5.5).

6b) Using the command line interface, create a light curve of the observation using the task *evselect* then display with *dsplot*.

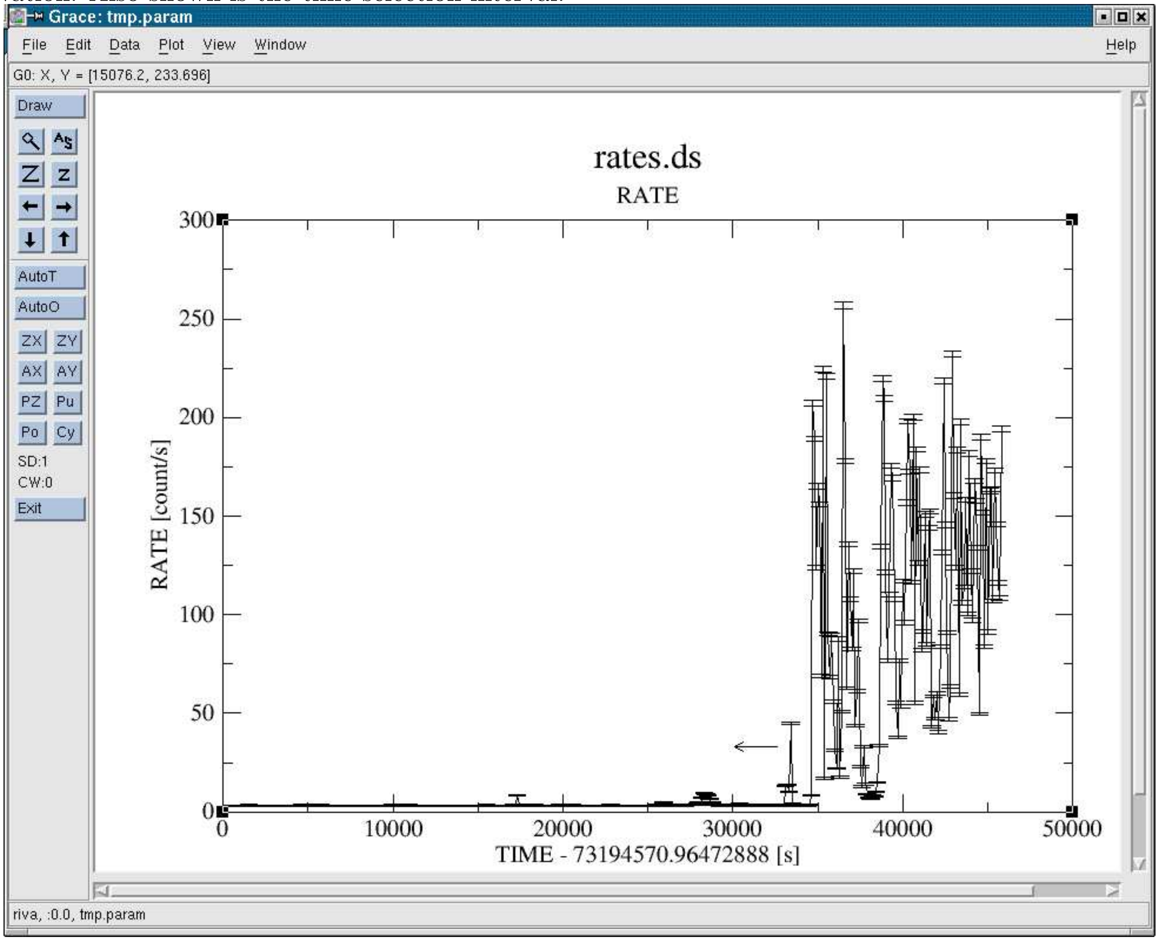
- `evselect table=/PIPE/mos1.fits:EVENTS withrateset=yes rateset=rate.fits maketimecolumn=yes timecolumn=TIME timebinsize=100 makeratecolumn=yes`
  > `table` – input event table.
  > `withrateset` – make an light curve.
  > `rateset` – name of output light curve file.
  > `maketimecolumn` – control to 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
- `dsplot table=rate.fits x=TIME y=RATE.ERROR withoffsetx=yes &`
  > `table` – input event table
  > `x` – column for plotting on X axis
  > `y` – column for plotting on Y axis, the nomenclature `RATE.ERROR` plots the count rate column (`RATE`) with the count-rate error column (`ERROR`) as uncertainties
  > `withoffsetx` – creates an offset to the X axis (-73194570.96472888 s in Figure 5.5)

5.2.4 Filter the Data and Create a New Event File

7) Next apply some filtering to the data. The "expressions" for the MOS and PN,
`(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM` and
`(PATTERN <= 4)&&(PI in [200:15000])&&#XMMEA_EP`,
will select good events with `PATTERN` in the 0 to 12 range (single, double, triple, and quadruple pixel events) and pulse height in the range of 200 to 12000 eV for the MOS and good events with `PATTERN` in the 0 to 4 range (single and double pixel events) and pulse height in the range of 200 to 15000 eV for the PN. 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. The selection on 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 `#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. The definitions of the `FLAG` values can be found in the FITS headers of the EVENTS extensions of the event files. FITS headers can easily be examined using *fv*.) An output file will be created for further processing.

7a) Filter the data using the *xmmselect* GUI.

Figure 5.5: Grace window showing the unfiltered light curve of the MOS1 data from the Lockman Hole SV1 observation. Also shown is the time selection interval.



- Since MOS data are being used, in the selection expression area at the top of the *xmmselect* GUI enter:
  `(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM`.
- Click on the "Filtered Table" box at the lower left of the *xmmselect* GUI.
- Change the *evselect* `filteredset` parameter, the output file name, to something useful, e.g., `mos1-filt.fits`. Click "Run".

7b) Filter the data using *evselect* on the command 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
  ```
  > `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
  ```

8) If necessary (and for the Lockman Hole SV1 data it most definitely is), add a temporal filtering clause to the *evselect* selection "expression". This is most often required because of soft proton flaring which can be painfully obvious with count rates of 50–100 counts a second, or more. Note that how much flaring needs to be excluded depends on the science goals of the analysis, a whopping bright point source will clearly be less affected than a faint extended object. A temporal filter can be easily created from the Grace light curve plot window.

   – Create a light-curve plot through the *xmmselect* GUI
   – In the Grace window, pull down the "Edit" menu, select "Regions", and select "Define"
   – For this case, select "Left of Line" for the "Region type"
   – 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.)
   – 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. Use `&&!(TIME in [73221920:73223800])` to exclude events in the time interval 73221920 to 73223800, the "!" symbol stands for the logical "not". The full "expression" would then be:
   `(PATTERN <= 12)&&(PI in [200:12000])&&#XMMEA_EM &&(TIME <= 73227600)`
   `&&!(TIME in [73221920:73223800])`
   Again, give the new file a useful name (`mos1-filt-time.fits`) and make sure that the `updateexposure` and `filterexposure` boxes are checked on the *evselect* GUI. Time filtering can also be done directly using the light curve by the creation of a secondary GTI file using the routine *tabgtigen* task.

   – `tabgtigen table=rate.fits:RATE expression='RATE<5'`
     `gtiset=gtisel.fits timecolumn=TIME`
     > `table` – input count rate table and extension (§ 5.2.3).
     > `expression` – filtering expression, in this case include those intervals where the count rate is < 5 counts s^{-1} in the individual 100 s intervals.
     > `gtiset` – output file name for selected GTI intervals.
     > `timecolumn` – time column.

   The output GTI table can then be used in the filtering expression in *evselect* with the syntax `&&GTI(gtisel.fits,TIME)`. The full "expression" would then be:
   `(PATTERN <= 12)&&(PI in [200:15000])&&#XMMEA_EM&&GTI(gtisel.fits,TIME)`.

   Figures 5.6 and 5.7 show the image and light curve generated from the filtered data.

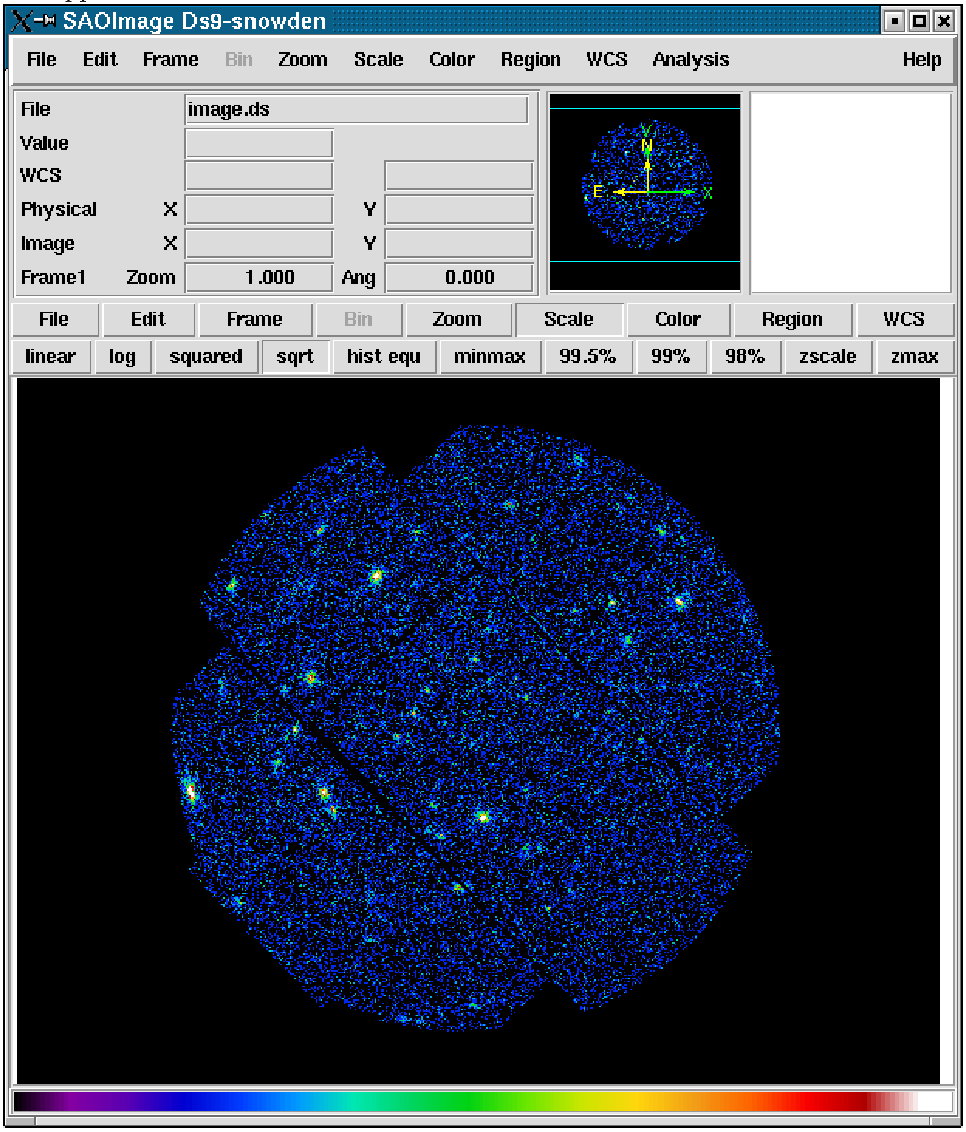
5.3 EXTRACT AND FIT A SOURCE SPECTRUM

5.3.1 Extract the Spectrum

While all of the data extraction can be done on the original file keeping the final selection "expression", it can save significant time and memory to operate on the filtered event file. For instance, in the case of the Lockman Hole data, the original MOS1 event file is 48.4 Mb while the filtered (spatial, temporal, and spectral) list is only 4.0 Mb. To change the event file, pull down the "file" menu on the *xmmselect* GUI and select "New Table". This will bring up the file selection browser and just follow the instructions in Item 4 of Section 5.2.1. The extraction of region-specific data (e.g., source spectra and light curves) is simplified by using the GUI again because of the treatment of selection regions.

1) With *xmmselect* running on the filtered file, create an image by selecting the small boxes to the left of the X and Y columns, clicking on the "Image" button, and then clicking on the "Run" button on the

Figure 5.6: Filtered image of the MOS1 data from the Lockman Hole SV1 observation. Displayed with a square root scale and an upper cut value of 20.



pop-up *evselect* GUI (for these purposes the default parameters are fine). To select a file name for the image rather than using the default `image.ds`, select the "Image" page on the *evselect* GUI and change the `imageset` entry.

2) On the *ds9* window, create a region for a source of interest. Click once on the *ds9* image and a region circle will appear. Click on the region circle and the region will be activated, allowing the region to be moved and its size to be changed. Having created, placed, and sized the region appropriate for the source, click the "2D region" button on the *xmmselect* GUI. This transfers the region information into the "Selection expression" text area, e.g., `((X,Y) IN circle(26144,22838,600))` for the bright source at the lower center of the Lockman Hole observation. The `circle` parameters are the X and Y positions and the radius of the circle in units of $0.''05$, so the above region description is for a circle of $30''$ radius.
Note: For serious spectral analysis the phrase `&&(FLAG == 0)` should be added to the selection expression. This provides the most stringent screening of the data and will exclude events such as those next to the edges of the CCDs and next to bad pixels which may have incorrect energies.

3) To extract the spectrum, first click the circular button next to the `PI` column on the *xmmselect* GUI.

Figure 5.7: Filtered light curve of the MOS1 data from the Lockman Hole SV1 observation.

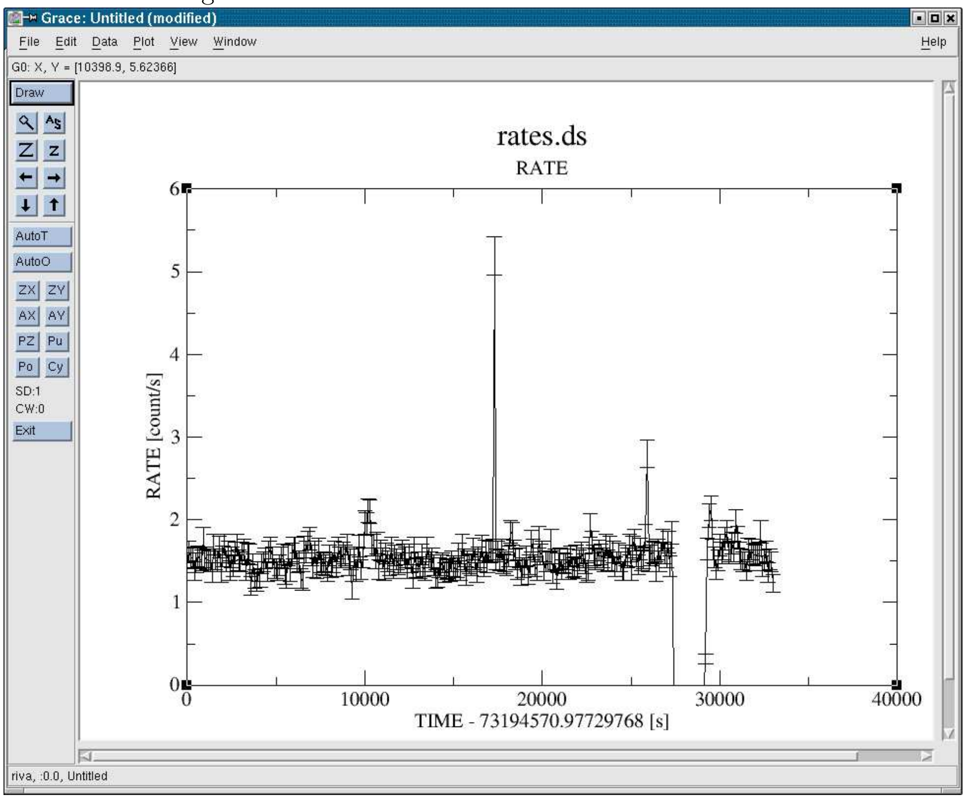
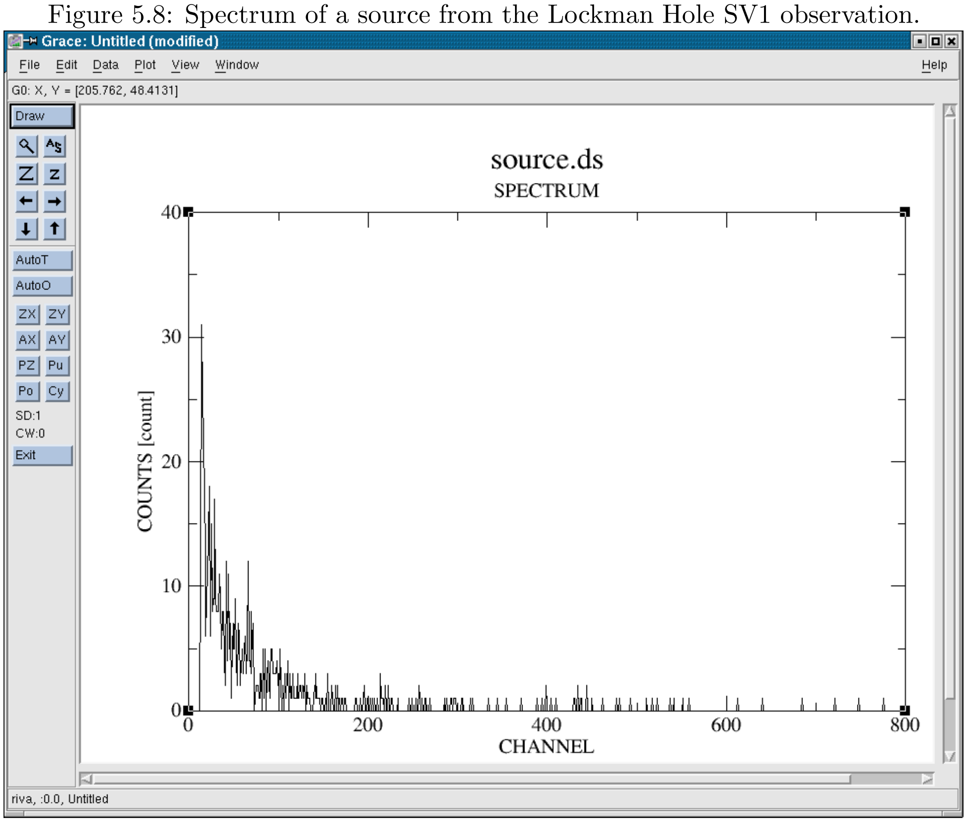


Figure 5.8: Spectrum of a source from the Lockman Hole SV1 observation.



Next click the "OGIP Spectrum" button. Select the "Spectrum" page of the *evselect* GUI to set the file name and binning parameters for the spectrum. For example, set `spectrumset` to `source.ds`. The `spectralbinsize` must be set to 15 for the MOS or 5 for the PN. `withspecranges` must be checked, `specchannelmin` set to 0, and `specchannelmax` set to 11999 for the MOS or 20479 for the PN. Figure 5.8 shows the spectrum.

4) To extract a background spectrum from an annulus surrounding the source, first clear the "Selection expression". Next repeat step 2) except create two circles defining the inner and outer edges of the background annulus. Use the "Properties" menu under the *ds9* "Region" menu to set the inner circle to "exclude". Then click the "2D region" button on the *xmmselect* GUI to transfer the region description of both circles to the "Selection expression". This may need to be edited. For example, for an annulus it should be as follows:
`((X,Y) IN circle(26144,22838,1500))&&!((X,Y) IN circle(26144,22838,900)).`
This will include data within a circle of radius 75″ but not within a concentric circle of 45″ (the values are in units of 0.″05). Finally, repeat step 3) except set the `filteredset` parameter to a different file name, e.g., `back.ds`.

5) To extract the source light curve, put the source "Selection expression" (the region descriptor used in Step 3) in place and click the circular button next to the "TIME" column on the *xmmselect* GUI. (Note: if you forgot to record it, the region selection criteria can be found in the FITS header of the spectrum extension of the spectrum file, e.g., `source.ds`.) Next click the "OGIP Rate curve" button. Select the "Lightcurve" tab of the *evselect* GUI to set the file name and binning parameters for the light curve. For example, set `filteredset` to `source.rate` and `timebinsize` to 1000 for a reasonable binning for the source examined in the spectral analysis section. (NOTE: set `timebinsize=1` and deselect `makeratecolumn` to create the light curve for the temporal analysis example in § 5.5. The first forces the time interval to be 1 s and the second creates a count rather than a count rate column.)

6) Depending on how bright the source is and what modes the EPIC detectors are in, event pile up can possibly be a problem. Pile up occurs when a source is so bright that there is the non-negligible possibility that X-rays will 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*. To run *epatplot* create source and background event files by extracting data from the original event file using the time and region selection expressions combined with the `FLAG == 0` filtering (all `PATTERN` values are required). On the *xmmselect* GUI click the "Filtered Table" button and check the `updateexposure` on the *evselect* "General" page and provide a `filteredset` name, e.g., `mos1-source.fits` and `mos1-back.fits`, for the resultant files. Invoke *epatplot* from the SAS GUI, enter the source event file name (e.g., mos1-source.fits) for the `set` parameter on Tab 0 and set `withbackgroundset` to yes and provide the background event file name (e.g., mos1-back.fits) for the `backgroundset` parameter on Tab 1, and click on "Run". If the plot shows the model distributions for single and double events diverging significantly from the measured distributions then pileup must be considered. Figure 5.9 shows an example of a bright source (from a different observation) which is not strongly affected by pileup. The source used in this example is too faint to provide reasonable statistics for *epatplot* and is far from being affected by pile up.

5.3.2 Create RMFs and ARFs

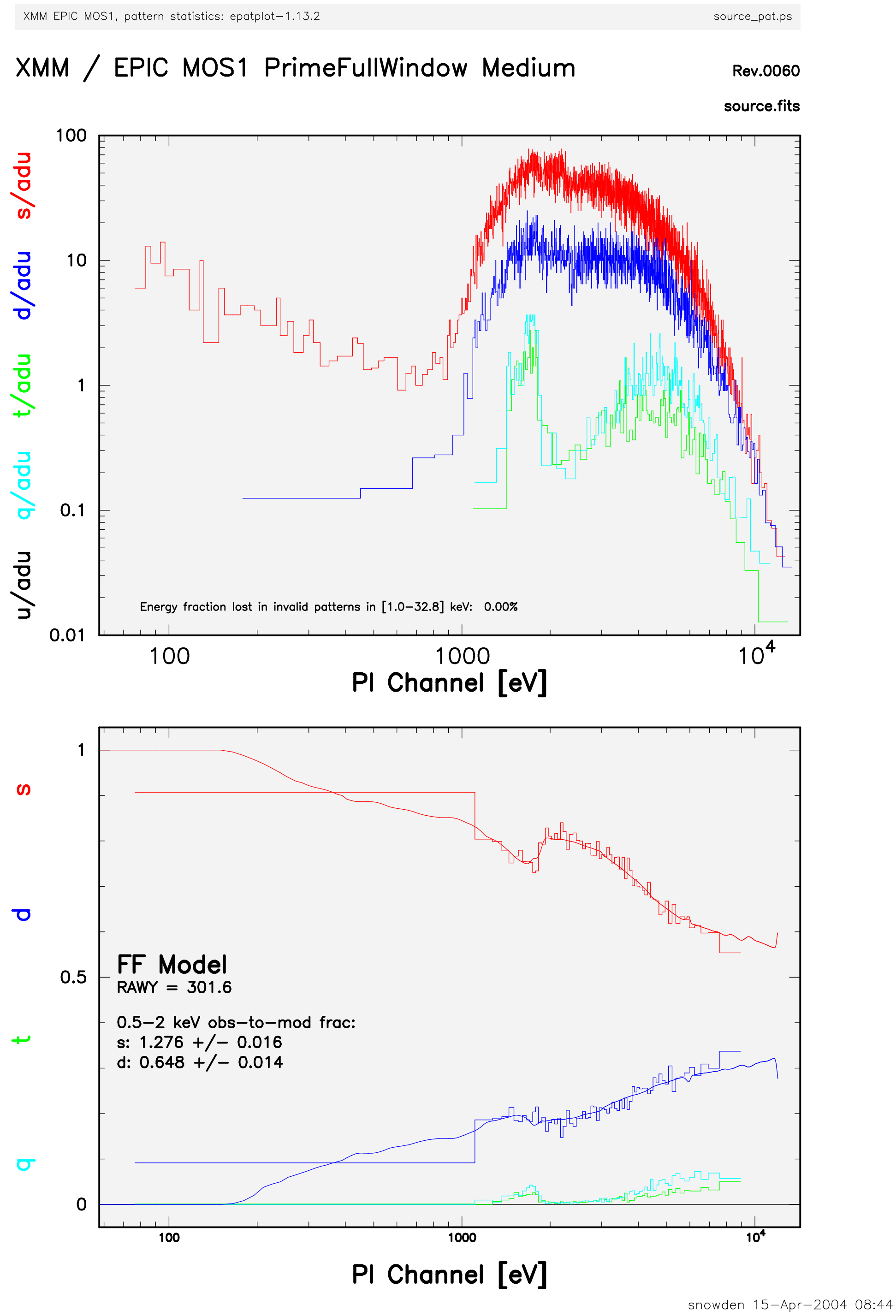
The following assumes that an appropriate source spectrum, named `source.ds`, has been extracted as in § 5.3.1.

7a) Create the photon redistribution matrix, the RMF, using the task *rmfgen* GUI.

- From the SAS GUI, invoke the *rmfgen* GUI (Figure 5.10)
- Set the `spectrumset` keyword to the spectrum file name, e.g., `source.ds`
- Set the `rmfset` keyword to the RMF file name, e.g., `rmf.ds`
- Click "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)

7b Create the photon RMF from the command line.

Figure 5.9: A MOS1 *epatplot* plot for a moderately bright source which does not show evidence for pileup. The central source from the observation of G21.5-09 (0122700101) is used.



– `rmfgen rmfset=response.ds spectrumset=source.ds`

  > `rmfset` – output RMF file name

  > `spectrumset` – input spectrum file name

8a) Create the ancillary region file, the ARF, using the task *arfgen* GUI.

  – From the SAS GUI, invoke the *arfgen* GUI (Figure 5.11)

  – On the "main" tab set the `spectrumset` keyword to the spectrum file name, e.g., `source.ds`

Figure 5.10: The *rmfgen* GUI.

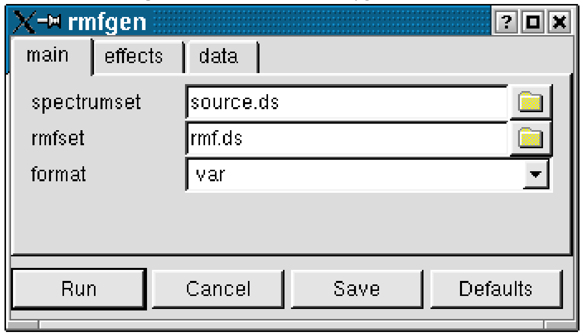
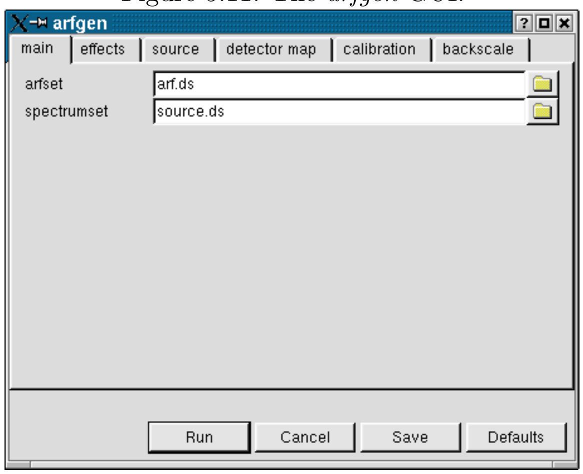


Figure 5.11: The *arfgen* GUI.



- On the “main” tab set the **arfset** keyword to the ARF file name, e.g., **arf.ds**
- On the “effects” tab set the **badpixlocation** keyword to the event file name from which the spectrum was extracted, e.g., **mos1-filt-time.fits**
- On the “calibration” tab check the “withrmfset” box and set the **rmfset** keyword to the RMF file name, e.g., **rmf.ds**
- Click “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)

8b) Create the ARF from the command line.

- arfgen arfset=arf.ds spectrumset=source.ds withrmfset=yes rmfset=rmf.ds badpixlocation=mos1-filt-time.fits

  > **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** – point to the file containing the bad pixel information, which should be the event file from which the spectrum was extracted

5.3.3 Prepare the Spectrum

Assuming that source and background spectra have been extracted as in § 5.3 and the RMF and ARF created as in § 5.3.2, spectral fitting will be demonstrated using HEASoft software.

9) Nearly all spectra will need to be binned for statistical purposes. The *FTOOL grppha* 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.

```
> grppha

Please enter PHA filename[] source.ds          ! input spectrum file name
Please enter output filename[] source-grp.ds   ! output grouped spectrum
GRPPHA[] chkey BACKFILE back.ds                ! include the background spectrum
GRPPHA[] chkey RESPFILE rmf.ds                 ! include the RMF
GRPPHA[] chkey ANCRFILE arf.ds                 ! include the ARF
GRPPHA[] group min 25                          ! group the data by 25 counts/bin
GRPPHA[] exit
```

5.3.4 Fit the Spectra

10) Next use Xspec to fit the spectrum.

```
> xspec

XSPEC> data source-grp.ds      ! 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 5.12 shows the fit to the spectrum.

5.4 SOURCE DETECTION

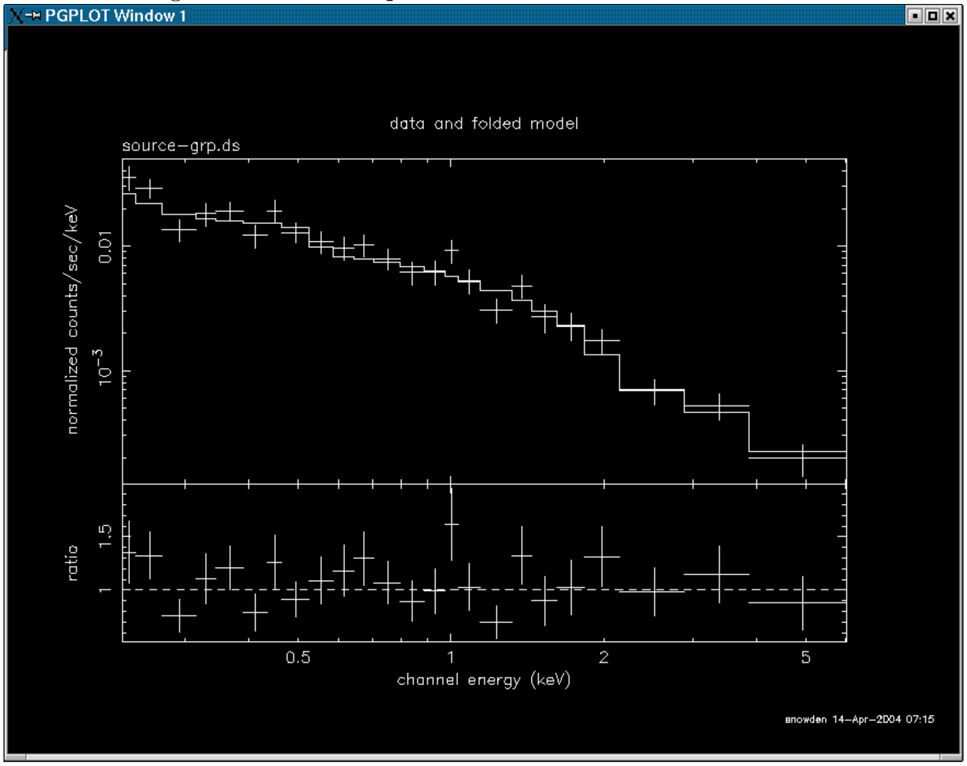
The *edetect_chain* does nearly all the work involved with EPIC source detection. In the example below source detection is done on images in two bands for all three detectors. The example uses the filtered event files produced as in § 5.2.4 with the assumption that they are located in the current directory.

1) Create an attitude file using *atthkgen*, this is required for the creation of the exposure maps. Note that the file *ATTTSR* is the attitude file created by the pipeline processing and can also be used. Both the *atthkgen* GUI and command line are easy to use.

   – `atthkgen atthkset=attitude.fits timestep=1`
     > `atthkset` – output file name
     > `timestep` – time step in seconds for attitude file

2) Create images in sky coordinates over the PI channel ranges of interest using the task *evselect* (the GUI can be used as well). It will use the filtered event list `mos1-filt.fits` produced above. In this example *evselect* is run six times to create the images in two bands (300 - 2000 eV, and 2000 - 10000 eV) for each of the three detectors.

Figure 5.12: Fitted spectrum of the Lockman Hole source.



- 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])'
  - table – event list
  - withimageset – flag to create an image
  - imageset – fits image name to be created, image1.fits for band 1
  - imagebinning – how to bin the image
  - xcolumn – table column to use for the X axis
  - ximagebinsize – binning in X axis (original pixels are 0.05″)
  - ycolumn – table column to use for the Y axis
  - yimagebinsize – binning in Y axis (original pixels are 0.05″)
  - filtertype – type of filtering
  - expression – filtering expression, select events in the PI channel range 300-2000 eV
- evselect table=mos1-filt-time.fits withimageset=yes imageset=mos1-h.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(FLAG == 0)&&(PI in [2000:10000])'
- evselect table=mos2-filt-time.fits withimageset=yes imageset=mos2-s.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(FLAG == 0)&&(PI in [300:2000])'
- evselect table=mos2-filt-time.fits withimageset=yes imageset=mos2-h.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(FLAG == 0)&&(PI in [2000:10000])'
- evselect table=pn-filt-time.fits withimageset=yes imageset=pn-s.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(FLAG == 0)&&(PI in [300:2000])'
- evselect table=pn-filt-time.fits withimageset=yes imageset=pn-h.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(FLAG == 0)&&(PI in [2000:10000])'

3) Create a merged count image for later display purposes.

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

     > `imagesets` – list of count images
     > `mosaicedset` – output file name

4) 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`

     > `imagesets` – list of count images
     > `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 10^{11} counts cm^2 erg^{-1}. 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} cm^{-2}, and the thin filters.

5) 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`

     > `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

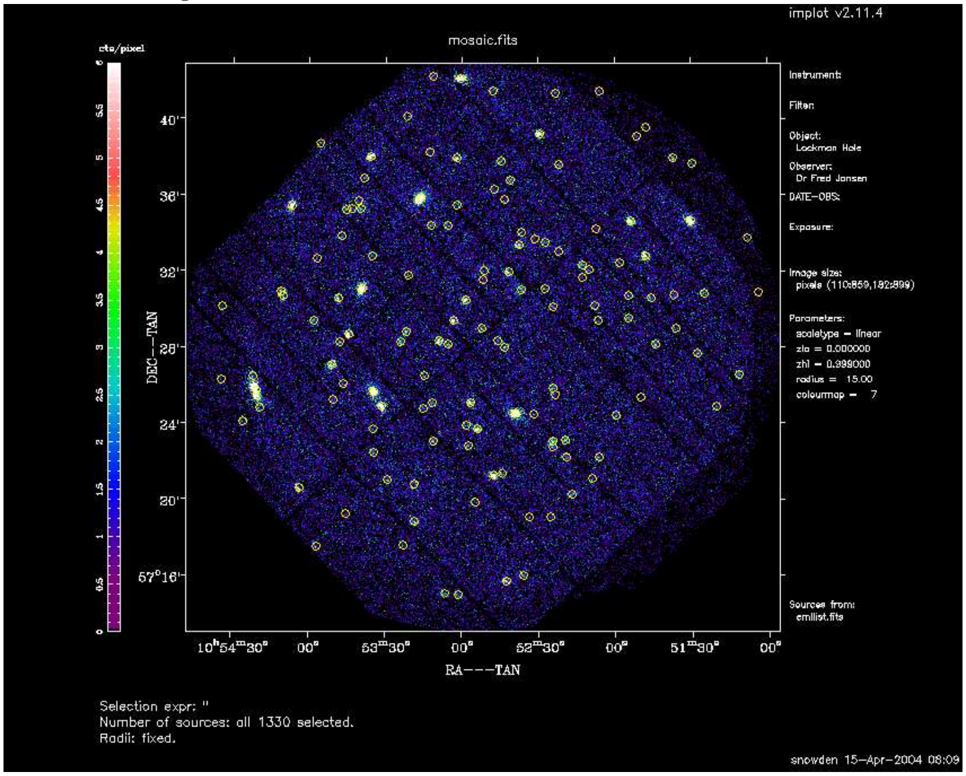
6) Display the results of *emldetect* using the task *implot*, in this case as a GIF file (pgplot.gif).

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

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

Figure 5.13 shows the output of *implot* for the maximum likelihood source detection (*emldetect*).

Figure 5.13: EPIC count image with the detected sources from the maximum likelihood task created by *implot*.



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 § 5.3 with `timebinsize` set to 1 and `makeratecolumn` set to "no". For this exercise the central source from the observation of G21.5-09 (0122700101) is used. For the aficionado, the task *barycen* can be used for the barycentric correction of the source event arrival times.

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`

     > 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 5.14

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`

     > cfile1 – filename first series

Figure 5.14: Light curve for the source analyzed in § 5.3.

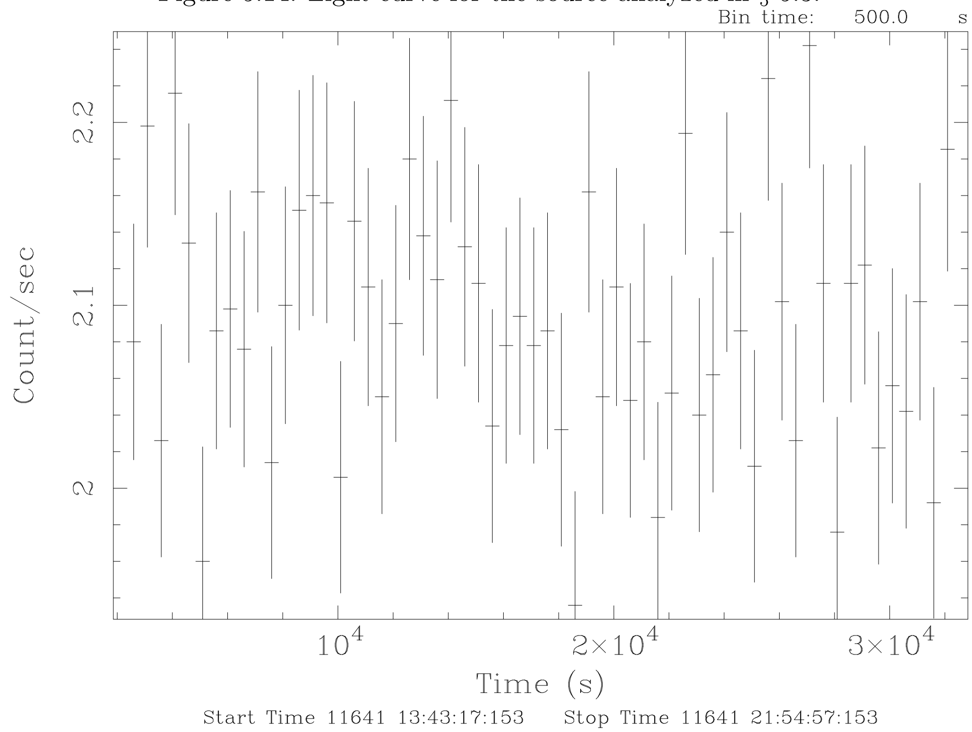
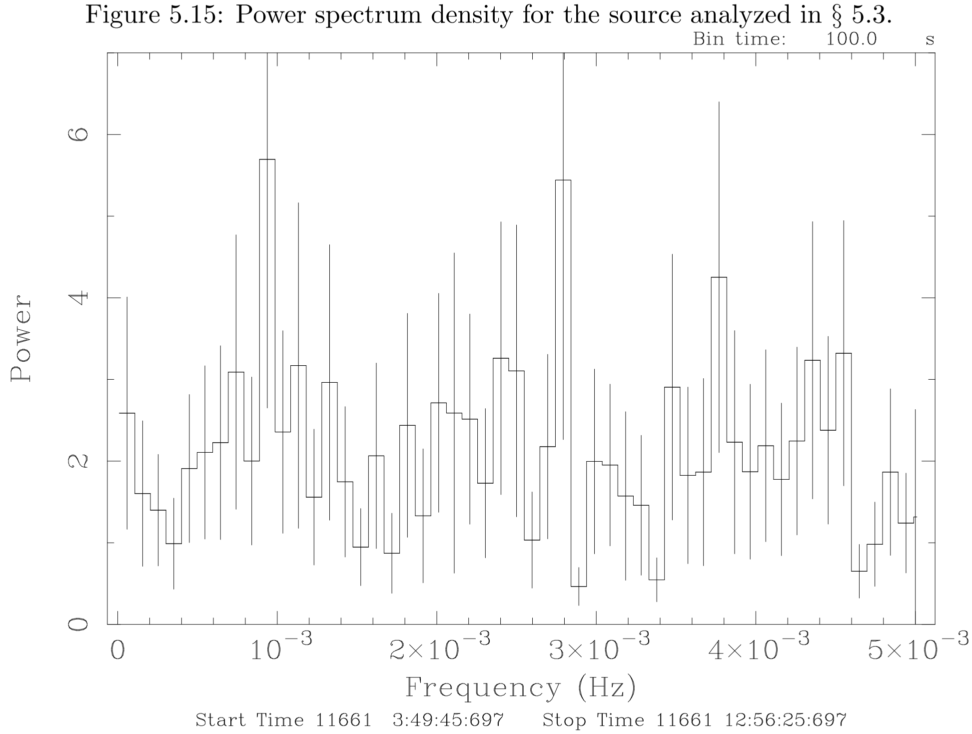


Figure 5.15: Power spectrum density for the source analyzed in § 5.3.



> 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 5.15
> 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`

   > 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`

   > 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. (Leave off the `>` *fname* to write the results to the screen.)

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

   > 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.

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

     > `FLAG < 65536` is the equivalent of the *xmmselect* expression `#XMMEA_EM`, for PN data also use `"FLAG < 65536"`

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).

   – Invoke the command `extract curve`

   – Invoke the command `plot curve`

6a) Filter on time using the cursor and light curve plot.

   – `filter time cursor`, then 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

6b) Filter on time using a threshold intensity.

   – `filter intensity`, then give the filter range, e.g., "0.01-3.0"

7a) Create the extraction region for the source.

   – Extract and plot a new image with the temporal filter

   – 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`

   – Adjust the region to be appropriate for the source of interest

   – Under the region menu select the `Save Regions` option

   – Save the region as a file (e.g., `ds9-source.reg`)

7b) Create an annulus extraction region for the background.

   – If necessary, resize the existing region to be appropriate for the inner annulus radius

   – Pull down the `Region` menu and select `Exclude` under `Properties`

   – Create a second region on the *ds9* window

   – Adjust the region to be appropriate for the outer boundary of the annulus

- Pull down the `Region` menu and select `Include` under `Properties`
- Make sure that the outer annulus is "in front" by selecting the `Move to Front` option under the `Region` menu.
- Under the region menu select the `Save Regions` option
- Save the region (e.g., `ds9-back.reg`)

8) Filter the data using the source region.
   - `filter region ds9-source.reg`

9) 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)

10) Filter the data using the background region.
   - First remove the source filter expression: `clear region all`
   - `filter region ds9-back.reg`

11) 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)

12) Extract, plot, and save the light curve from the region.
   - First remove the source filter expression: `clear region all`
   - `filter region ds9-source.reg`
   - `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 § 5.3.3 and § 5.3.4 for spectral analysis and § 5.5 for temporal analysis.

5.7 ODF DATA

The ODF names for the EPIC data will look something like:

- `mmmm_iiiiiijjkk_aabeeeccfff.zzz`

  `mmmm` – revolution orbit number

  `iiiiiijjkk` – observation number

  `aa` – detector ID (M1 - MOS1, M2 - MOS2, PN - PN).

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

  `eee` – exposure number within the observation

  `cc` – CCD identifier.

  `fff` – data identifier (see Table 5.2)

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

Table 5.2: EPIC ODF data files[1].

Data ID	Contents
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

[1]From the document GEN-ICD-0004-2-8.

5.7.1 Rerunning the EPIC Chains

When the CCF is updated it may be necessary to rerun the basic pipeline processing (see § 4.5.1), and luckily the process is reasonably simple. This next set of tasks will reproduce the calibrated photon event files found in the pipeline products. (Note: for reference, an executable log file of the entire pipeline processing can be found in the pipeline product *SCRLOG*.) Since data in the public archive are typically at least a year old (although with the coming reprocessing this will change, for a while) older versions of both the CCF and SAS were used, it is therefore useful to rerun the pipeline processing to reproduce the event files.

1) 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.

2) Initialize SAS (see § 4).

3) Create a CIF file 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.

4) Run the SAS task *odfingest* (§ 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, first delete the earlier *.SAS (the file produced by *odfingest*).

5) Run the SAS task *emchain*. From the command line of a window where SAS has been initialized, simply enter:
`emchain`
emchain processes the data from both MOS instruments producing calibrated photon event files. If the data set 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.

6) Run the SAS task *epchain*, which processes the data from PN instrument producing a calibrated photon event file. From the command line of a window where SAS has been initialized, simply enter:
`epchain`
To create an out-of-time event file, use the command:
epchain withoutoftime=yes
Adding the parameter `keepintermediate=none` causes *epchain* to discard a number of intermediate files.

Once the chains have completed with new event files the same analysis techniques described in the previous sections can used.

5.8 A More-or-Less Complete Example

The Lockman Hole ODF data have been used for a reasonably complete example of the EPIC data reduction. The data can be found via the XSA at:
`http://xmm.vilspa.esa.es/xsa/`
or via Browse at:
`http://heasarc.gsfc.nasa.gov/db-perl/W3Browse/w3browse.pl`
while the script (`run.com`) and output data files (except for the unfiltered event lists which are huge) can be found at:
`ftp://legacy.gsfc.nasa.gov/xmm/data/examples/epic/`
The script assumes that SAS V6.0 has been set up to run. The commands to set the CCF and ODF directories as well as the ccf.cif file are included but will need to be changed for the specific setup. The data were processed using SAS V6.0.

The entire process took two hours on a relatively new linux RH7.3 machine (1.67 GHz processor, 2 GB RAM). The result is about a gigabyte of new files. The script uses the SAS command-line interface, however in its creation the GUI interface to *xmmselect* was used to find the time filtering and source extraction parameters. The script goes through the following steps.

1) Sets a few SAS parameters.

2) Runs *cifbuild* and *odfingest* to prepare for SAS analysis.

3) Runs *emchain* to produce calibrated photon event files for the MOS1 and MOS2 detectors

4) Creates images and light curves of the MOS data

5) Filters the MOS event files to exclude bad events and times of background flares

6) Creates images and light curves of the filtered MOS data

7a) Runs *epchain* to produce a calibrated photon event file for the PN detector

7b) Runs *epchain* a second time to produce a calibrated photon event file for the PN detector of out-of-time events

8) Repeats items 4 - 6 for the PN data

9) Does source detection for two bands for each of the three detectors

10) Extracts source and background spectra for a brighter field source

11) Creates an RMFs and ARFs for the source

12) Groups the spectra using grppha

13) Included in the script, but commented out, are the commands to fit the source spectra using Xspec

14) Creates a light curve for the source

15) Included in the script, but commented out, are the commands to analyze the source light curve using Xronos

Chapter 6

First Look – RGS Data

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

- `http://xmm.vilspa.esa.es/sas/documentation/watchout`

This web site discusses current and past SAS bugs and analysis issues, e.g., regarding missing libraries when using *rgsproc* with SAS V6.

6.1 A PRELIMINARY FIT

6.1.1 Pipeline Products

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 6.1 lists typical file names, their purpose, the file format, and a list of tools that will enable the user to inspect their data. As usual, there are some HTML products to help you inspect the data with file names of the form (note that we will use the generic form of the name in the following examples):

- `PPiiiiiijjkkAAAAAA000_0.HTM`, where

  `iiiiii` – proposal number

  `jj` – observation ID - target number in proposal

  `kk` – observation ID - observation number for target

  `AAAAAA` – Group ID (Table 6.1)

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

The INDEX.HTML file will help you navigate. The data file names are of the form:

- `PiiiiiijjkkaablllCCCCCCnmmm.zzz`, where

  `iiiiii` – proposal number

  `jj` – observation ID - target number in proposal

  `kk` – observation ID - observation number for target

  `aa` – detector, `R1` – `RGS1`, `R2` – `RGS2`

  `b` – `S` for scheduled observation, `U` for unscheduled

  `lll` – exposure number

  `CCCCCC` – file identification (Table 6.1)

  `n` – spectral order number, unimportant otherwise

  `mmm` – source number

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

Table 6.1: RGS Pipeline Processing data files.

Group ID	File ID	Contents	File Type	View With
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	2nd Order Source Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC1	1st Order Back. Spectra	Zipped FITS	*Xspec, fv*
	BGSPEC2	2nd Order Back. 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, Netscape*
	IMAGE_	Images, disp. vs. PI	Zipped FITS, PNG	*ds9, Ximage, fv, Netscape*

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

`PNG` – use *Netscape* or other web browser

`HTM` – use *Netscape* or other web browser

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

6.1.2 Preparation for Running SAS Tasks

1) Ensure that you have created a Calibration Index File, using *cifbuild* (§ 4.5.1).
2) Ensure that you have created a summary file of your ODF constituents (just once) and deposited it in your ODF directory using the SAS task *odfingest* (§ 4.5.2).

6.1.3 Creating Response Matrices

Response matrices and ancillary response files are not provided as part of the pipeline product package, so a user must create their own before analyzing data. The SAS package *rgsrmfgen* generates an RMF and ARF and combines them within a single RSP file. The following command demonstrates this using the pipeline products above:

- `rgsrmfgen file=RGS1_ORDER1.RSP evlist=PiiiiiijjkkaablllEVENLInmmm.FTZ withspectrum=yes spectrumset=PiiiiiijjkkaablllSRSPEC1mmm.FTZ emin=0.3 emax=2.8 ebins=4000`

  > `file` – the name of the output response matrix.

  > `evlist` – the event list from which the spectrum was extracted.

  > `withspectrum` – Use the spectrum file product to calculate the RMF.

  > `spectrumset` – name of the spectrum file from the pipeline products. Source, order, background, and response channel binning will be taken from this file.

  > `emin` – the lower energy limit of the RSP file.

  > `emax` – the upper energy limit of the RSP file.

  > `ebins` – The number of bins calculated between emin and emax. The task documentation suggests this number be > 3000.

The response files take many factors into account, such as pointing, pile-up, telemetry saturation, hot pixels, instrument temperatures, etc. Therefore it is imperative to create new response files after any filtering of the data and the same response should never be used for fitting two different observations or pointings.
Note that if the pipeline products were created with SAS v5.0 this procedure will fail because of a more recent code alteration. To construct a response matrix, the pipeline should be re-run first using the latest software (SAS V6) and calibration.

6.1.4 Fitting a Spectral Model to the Data

Now use XSPEC to fit an appropriate model to your spectrum:

xspec

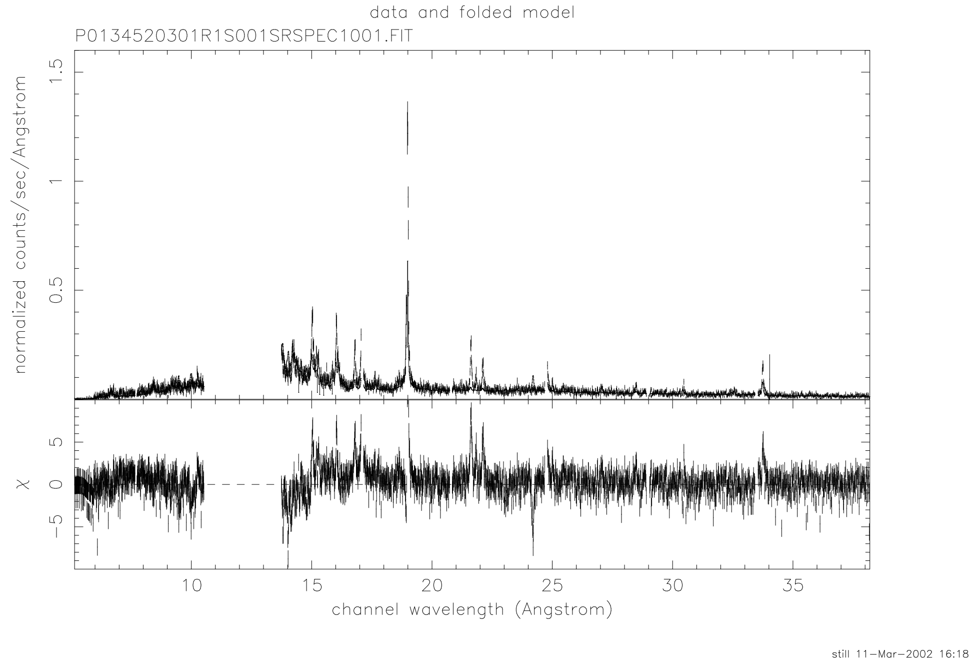
```
XSPEC>data PiiiiiijjkkaablllSRSPEC1mmm.FTZ
XSPEC>back PiiiiiijjkkaablllBGSPEC1mmm.FTZ
XSPEC>resp RGS1_ORDER1.RSP
XSPEC>ignore bad
XSPEC>model wabs*mekal
wabs:nH>1
mekal:kT>1
mekal:nH>
mekal:Abundanc>0.4
mekal:Redshift>
mekal:Switch>0
mekal:norm>1
XSPEC>renorm
XSPEC>fit
XSPEC>setplot device /xs
XSPEC>setplot wave
XSPEC>setplot command window all
XSPEC>setplot command log x off
XSPEC>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
Do you really want to exit?  (y)y
```

The plot is provided in Figure 6.1.
Please note:

- PiiiiiijjkkaablllSRSPECnmmm.FTZ is a "net" spectrum. This is the source+background events minus the background events which were extracted from a different detector region. A number of xspec functions will yield erroneous results using net spectra – see Sec. 6.6

- PiiiiiijjkkaaSlllBGSPECnmmm.FTZ is a background spectrum. Consequently, when analyzing the net file with XSPEC or CIAO, DO NOT employ this file as a background for your data. It has already been subtracted.

Figure 6.1: 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.



6.2 FILTERING EVENTS

Solar flares result in periods of high background. Observers may find an appreciable increase in signal-to-noise if they remove flare events from their data. The general SAS task *evselect* does not correct the RGS exposure maps during filtering which is vital in order to fit data accurately. Consequently the RGS-specific task *rgsfilter* must be run in order to perform any filtering of the data. As with the majority of RGS tasks, *rgsfilter* can be called from the meta-task *rgsproc* which provides a convenient interface between the user and the entire RGS pipeline. This section provides an example of how to produce a time-filtered spectrum.

6.2.1 Creating and plotting a light curve

Create a FITS light curve with 100 second binning from the pipeline product event file using the SAS task *evselect* (alternatively use the *xmmselect* GUI). Being closer to the optical axis, CCD9 is most susceptible to proton events and generally records the least source events, therefore we will extract events over this CCD only. 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 `PiiiiiijjkkaablllSRCLI_nmmm.FTZ`

- `evselect table=PiiiiiijjkkaablllEVENLInmmm.FTZ withrateset=yes rateset=RGS1_RATE.FIT makeratecolumn=yes maketimecolumn=yes timebinsize=100 expression='(CCDNR==9)&&(REGION(PiiiiiijjkkaablllSRCLI_nmmm.FTZ:RGS1_BACKGROUND, BETA_CORR,XDSP_CORR))'`

  > `table` – event list from the pipeline products.

  > `withrateset` – create a light curve.

  > `rateset` – name of the resulting FITS file.

  > `makeratecolumn` – create a rate column.

  > `maketimecolumn` – create a time column.

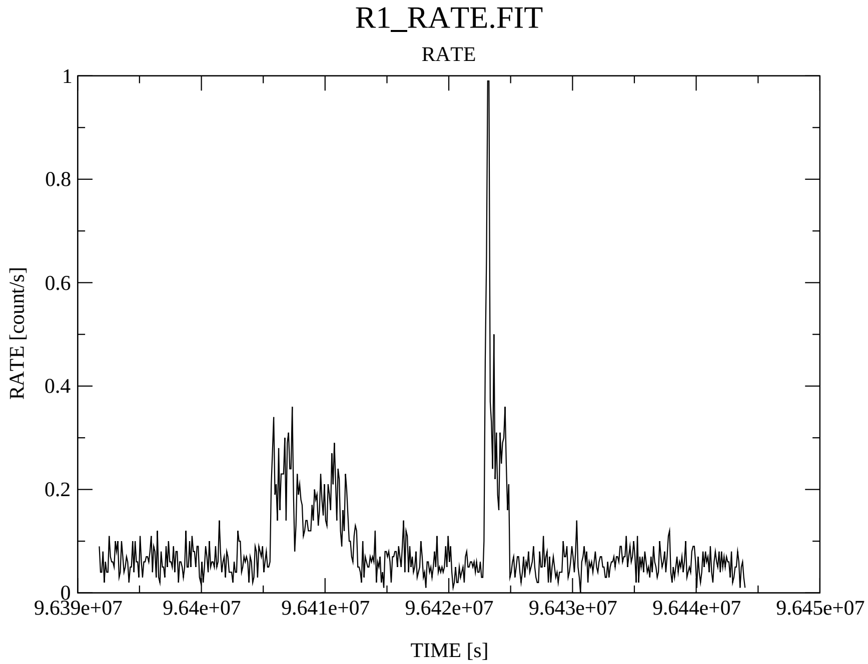
  > `timebinsize` – bin the time column to this size (in units of seconds).

  > `expression` – filter expression.

Plot the light curve using the SAS tool *dsplot* (see Figure 6.2):

- `dsplot table=RGS1_RATE.FIT x=TIME y=RATE`

Figure 6.2: Background event rate from the RGS1 CCD9 chip. The flares are solar events. The time units are elapsed mission time.



6.2.2 Creating a GTI file

The CCD9 quiescent count rate within the background region mask is ∼ 0.05 counts per second. In this example there are two intervals of significant background flaring. Determine which intervals should be rejected and write these time intervals to an ASCII file, `gti.asc`, as follows:

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

The first two columns provide the start and stop times (in seconds since the start of the mission) of intervals to be filtered. The third column can be a '+' (good time interval) or '−' (bad time interval). In this case two intervals of high background activity are excluded.

Execute `gtibuild` to convert the above into a FITS format GTI file:

- `gtibuild file=gti.asc table=GTI.FIT`

  > `file` – ASCII file of time intervals

  > `table` – GTI FITS file

Alternatively, a GTI table can be created using a Boolean expression to record times of acceptably low count rate with the task `tabgtigen`:

- `tabgtigen table=R1_RATE.FIT gtiset=GTI.FIT expression='(RATE<0.2)'`

  > `table` – Input data file.

  > `gtiset` – Output GTI table.

  > `expression` – Boolean expression.

6.2.3 Running the RGS Pipeline

One can now re-run the complete RGS pipeline using the SAS meta-task *rgsproc*.

Table 6.2: *rgsproc* output data files.

Data Type	Extension	File Type	Contents
`ATTTSR`	FIT	FITS table	attitude information for the complete observation.
`attgti`	FIT	FITS table	good time intervals from the attitude history.
`hkgti_`	FIT	FITS table	good time intervals from the housekeeping files.
`SRCLI_`	FIT	FITS table	list of sources and extraction masks.
`merged`	FIT	FITS table	event list merged from all CCDs.
`EVENLI`	FIT	FITS table	merged and filtered event list.
`EXPMAP`	FIT	FITS image	exposure map.
`SRSPEC`	FIT	FITS table	source spectrum.
`BGSPEC`	FIT	FITS table	background spectrum.
`matrix`	FIT	FITS table	response matrix.
`fluxed`	FIT	FITS table	fluxed spectrum. For quick and dirty inspection only.

- `rgsproc orders='1 2' auxgtitables=GTI.FIT bkgcorrect=no withmlambdacolumn=yes`

  > `orders` – the spectral orders to extract.

  > `auxgtitables` – a list of GTI files

  > `bkgcorrect` – Subtract background from source spectra?

  > `withmlambdacolumn` – include a wavelength column in the event file product (we will use this to generate a "dirty" spectrum plot later).

Note: If an error message that a library is missing occurs, follow these steps:

- go to the XMM-Newton SAS v6.0 ftp site `ftp://xmm.vilspa.esa.es/pub/sas/6.0.0/`
- go to the directory which contains the SAS V6 version for your platform
- download the package: *platform*-htrframes.tar.gz
- go to your $SAS_DIR
- gunzip *platform*-htrframes.tar.gz
- tar -xvf *platfor*m-htrframes.tar

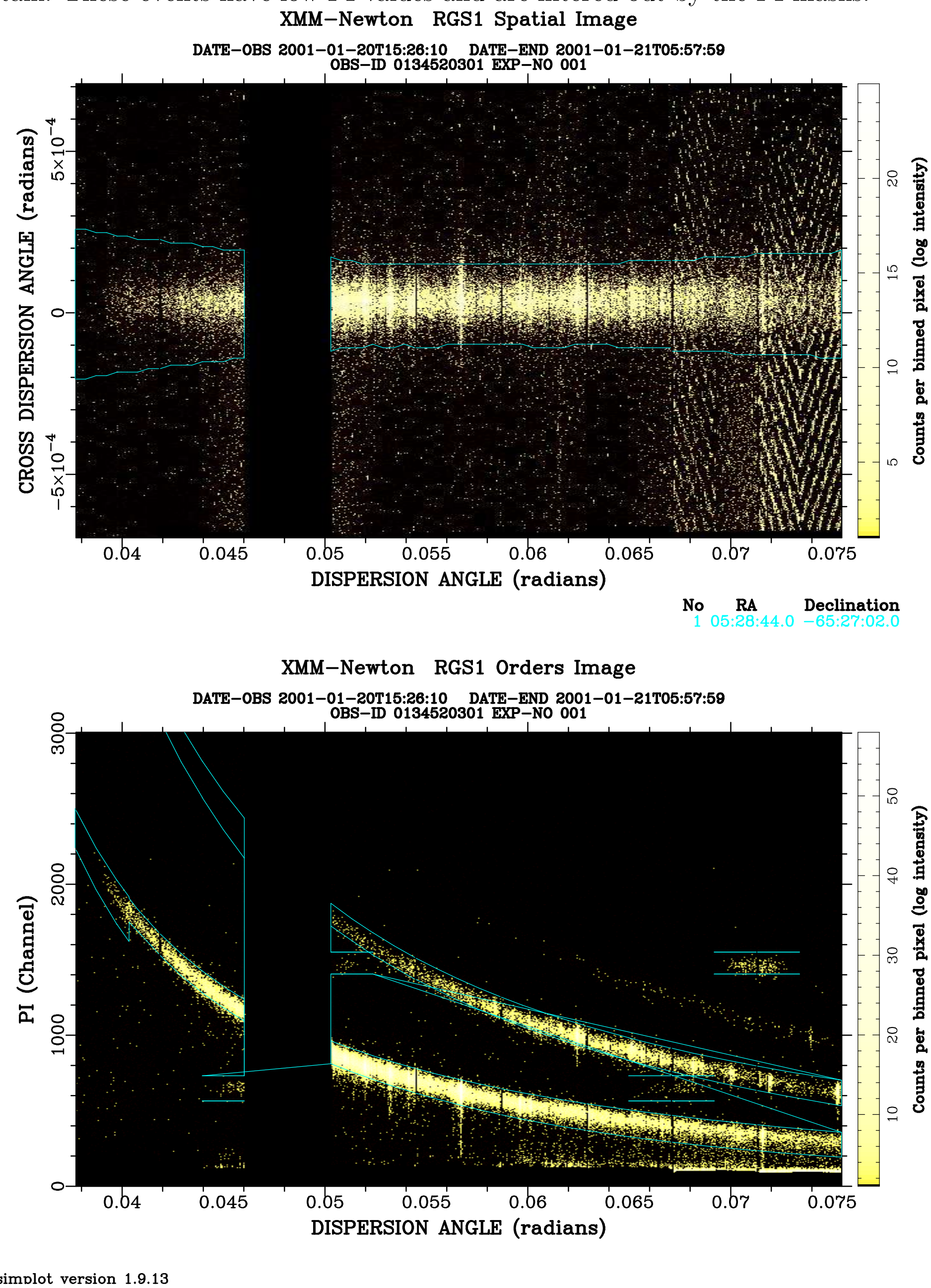
`bkgcorrect=no` will yield a source spectrum with background events included. The background level will be automatically subtracted if `bkgcorrect=yes`. Unless the spectra are of high signal-to-noise, it is recommended that scientific analysis only be carried out on those spectra where `bkgcorrect=no`. However, note that the fluxed spectrum (which is only suitable for initial data inspection) is best examined at after declaring `bkgcorrect=yes`. New files are written to the working directory. Table 6.2 lists these, and all are uncompressed FITS files. The filenames are of the same form given in Section 6.1.1:
Even if no solar flares occurred during the observation, it is recommended that the pipeline is re-run in order to take advantage of the most up-to-date calibration and ensure that region filters more appropriate for the source are created.

6.2.4 Inspecting New Products

To take a first look at these new products try the following recipes.

1) To examine images of dispersion versus PI and cross-dispersion directions:

Figure 6.3: Images over the dispersion–cross-dispersion plane (top) and the dispersion–pulse height plane (bottom). The lower and upper bananas are 1st and 2nd order events respectively. The blue lines define the source extraction regions, one spatial and the other over PI. Horizontal blue lines delineate the internal calibration sources. The regular chevron background pattern in the right hand CCDs (1 and 2) are a manifestation of electronic cross-talk. These events have low PI values and are filtered out by the PI masks.



- set srclst = 'PiiiiiijjkkaablllSRCLI_nmmm.FIT'
- evselect table=PiiiiiijjkkaablllEVENLInmmm.FIT:EVENTS withimageset=yes imageset=spatialimage.fit xcolumn=BETA_CORR ycolumn=XDSP_CORR

  > table – input events table.

  > withimageset – create an image.

  > imageset – output image file.

  > xcolumn – column in events file to extract

> `ycolumn` – column in events file to extract

- `evselect table=PiiiiiijjkkaablllEVENLInmmm.FIT:EVENTS withimageset=yes imageset=orderimage.fit xcolumn=BETA_CORR ycolumn=PI withyranges=yes yimagemin=0 yimagemax=3000 expression='region($srclst:RGS1_SRC1_SPATIAL,BETA_CORR,XDSP_CORR)'`

> `withyranges` – set the y range of the image.
> `yimagemin` – minimum y limit.
> `yimagemax` – maximum y limit.
> `expression` – filter expression. This example is filtering events found inside the spatial mask for the source.

- `rgsimplot withspatialset=yes withendispset=yes spatialset=spatialimage.fit endispset=orderimage.fit withspatialregionsets=yes withendispregionsets=yes srclistset=$srclst srcidlist=1 orderlist='1 2' colourmap=LOG colour=3 device=/XS`

> `withspatialset` – include spatial image.
> `withendispset` – include PI image.
> `spatialset` – name of spatial image.
> `endispset` – name of PI image.
> `withspatialregionsets` – include spatial mask in plot.
> `withendispregionsets` – include PI mask in plot.
> `srclistset` – name of source list.
> `srcidlist` – source number of the target within the source list. Source 1 will correspond to the target coordinates provided in the original proposal. Source 2 will be the camera boresight.
> `orderlist` – order of masks to plot
> `colourmap` – colour scale.
> `colour` – colour scheme for plot.
> `device` – plotting device (upper case is mandatory; e.g., /XS, /XSERVE, /PS, /CPS)

The output from rgsimplot is provided in Figure 6.3.

2) To plot a light curve from all events:

- `evselect table=PiiiiiijjkkaablllEVENLInmmm.FIT:EVENTS withrateset=yes rateset=PiiiiiijjkkaablllRATES_nmmm.FIT makeratecolumn=yes maketimecolumn=yes timebinsize=100`

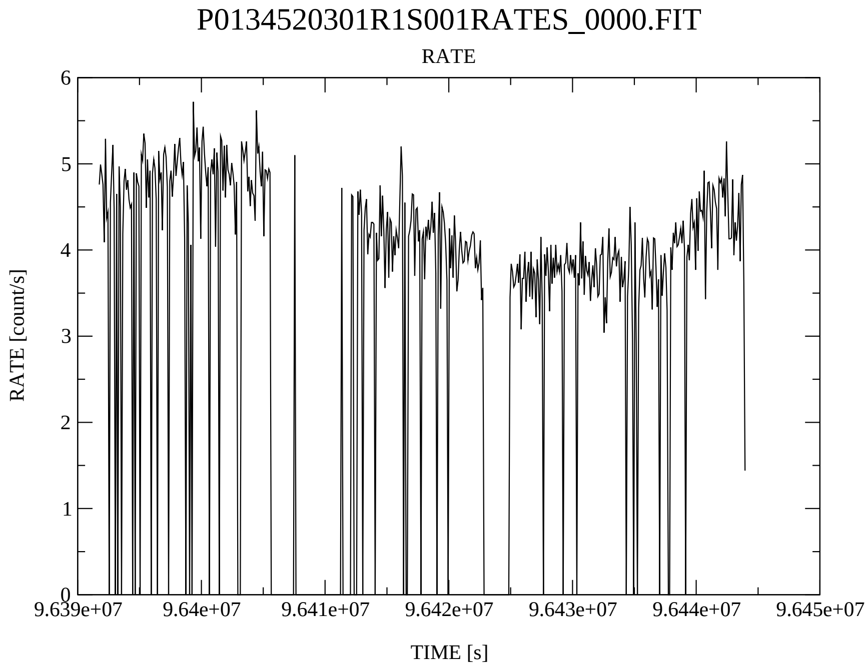
> `table` – input event list.
> `withrateset` – create a light curve.
> `rateset` – name of output light curve.
> `maketimecolumn` – include an absolute time column.
> `makeratecolumn` – create a rate column.
> `timebinsize` – bin size (in seconds).

- `dsplot table=PiiiiiijjkkaablllRATES_nmmm.FIT x=TIME y=RATE`

The resulting curve is provided in Figure 6.4. Note that unlike Figure 6.2 these events have been extracted across the whole detector and that our Good Time constraint has been adhered to.

3) To plot a spectrum with an approximate wavelength scale, use the mlambda table column rather than a response matrix. One important caveat here is that all orders are superimposed on this table:

Figure 6.4: Total event rate from RGS1 after Good Time filtering.



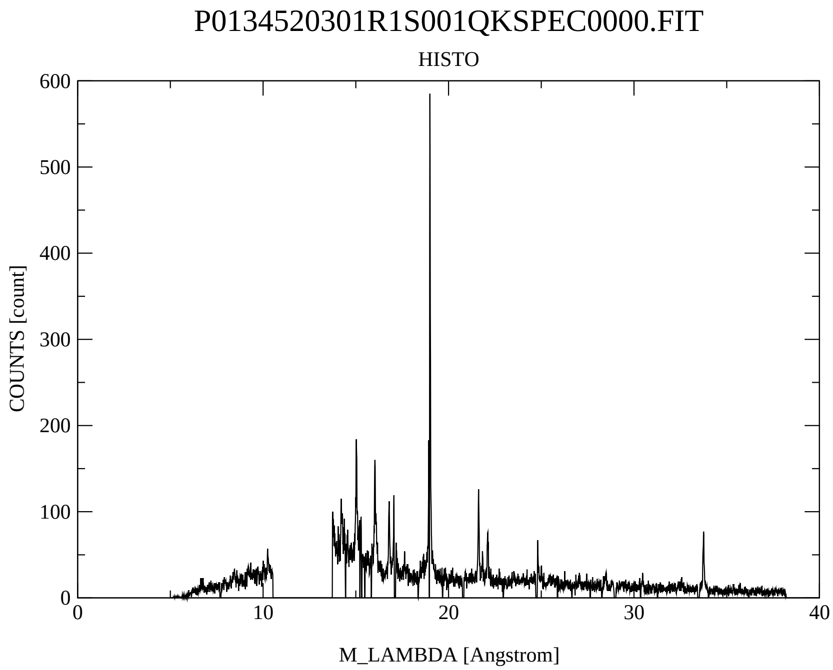
- `evselect table=PiiiiiijjkkaablllEVENLInmmm.FIT:EVENTS withhistogramset=yes histogramset=PiiiiiijjkkaablllQKSPECnmmm.FIT histogramcolumn=M_LAMBDA withhistoranges=yes histogrammin=5 histogrammax=40 histogrambinsize=0.0116667 expression='region($srclst:RGS1_SRC1_SPATIAL,BETA_CORR,XDSP_CORR)&& region($srclst:RGS1_SRC1_ORDER_1,BETA_CORR,PI)'`
  - > `table` – input event table.
  - > `withhistogramset` – make a histogram table.
  - > `histogramset` – name of output histogram table.
  - > `histogramcolumn` – event column to histogram.
  - > `withhistoranges` – include only certain ranges.
  - > `histogrammin` – lower limit.
  - > `histogrammax` – upper limit.
  - > `histogrambinsize` – size of histogram bins.
  - > `expression` – filter expression. This one takes only events from inside both the spatial and 1st order PI masks defined within the source list.
- `dsplot table=PiiiiiijjkkaablllHISTOGnmmm.FIT x=M_LAMBDA y=COUNTS`

The resulting histogram is provided in Figure 6.5.

6.3 PIPELINE EXAMPLES

Several examples of the flexibility of the RGS pipeline are provided below, and these address some of the potential pitfalls for RGS users.

Figure 6.5: RGS1 spectrum binned on the approximate wavelength scale provided in the M_LAMBDA column. the gap between 10 and 15Å is the missing chip CCD7. CCD4 is similarly missing in the RGS2 camera. Both failed after space operations began.



6.3.1 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:

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

  > `orders` – dispersion orders to extract

  > `calcoffsets` – calculate pha offsets from diagnostic images

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

6.3.2 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'`

  > `orders` – dispersion orders to extract.

  > `withepicset` – calculate extraction regions for the sources contained in an EPIC source list.

  > `epicset` – name of the EPIC source list.

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

Since this operation will alter only the size of the regions in the sources file, it saves time to not re-make the event table or re-calculate the exposure map. The pipeline can be entered at five different points. In this case one only need start from the spectral extraction stage:

- `rgsproc orders='1 2' entrystage=spectra finalstage=fluxing bkgcorrect=no withepicset=yes epicset=PiiiiiijjkkaablllEMSRLInmmm.FTZ exclsrcsexpr='INDEX==1&&INDEX==3'`

  > `orders` – dispersion orders to extract.

  > `entrystage` – entry stage to the pipeline (see Sec. 6.4).

  > `finalstage` – exit stage for the pipeline (see Sec. 6.4).

  > `withepicset` – calculate extraction regions for the sources contained in an EPIC source list.

  > `epicset` – name of the EPIC source list.

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

Note that this last example will only work if one has retained the event file from a previous re-running of the pipeline.

6.3.3 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:

- `rgsproc orders='1 2' bkgcorrect=no withsrc=yes srcra=81.823317 srcdec=-6.532072`

  > `orders` – dispersion orders to extract.

  > `withsrc` – with a user-defined source.

  > `srcra` – decimal RA of source.

  > `srcdec` – decimal Dec of source.

These coordinates are written to the RGS source list PiiiiiijjkkaablllEMSRLInmmm.FIT with a source ID which, in this example, will be '3'. Creating the source file is one of the first tasks of the pipeline. If these new coordinates correspond to the prime source then the entire pipeline must be run again in order to calculate the correct aspect drift corrections in the dispersion direction. However, if these new coordinates refer to a background source that should be ignored during background extraction, then the majority of pipeline processing (drift correction, filtering etc) will remain identical to the previous examples. To save processing time one can create a new source list by hand and then enter the pipeline at a later stage.

- `rgssources filemode=create srclist=PiiiiiijjkkaablllEMSRLInmmm.FIT atthkset='P0iiiiiijjkkaablllATTTSRnmmm.FIT' writeobskwds=yes writeexpkwds=yes instexpid='R1S001' addusersource=yes label='BACK_SOURCE' ra=81.823317 dec=-6.532072`

  > `filemode` – create or modify an existing source list.

  > `srclist` – name of resulting source list.

  > `atthkset` – attitude history file.

  > `writeobskwds` – write observation keywords to the source list.

  > `writeexpkwds` – write exposure keywords to the source list.

  > `instexpid` – instrument/exposure ID.

  > `addusersource` – add a source to the list.

  > `label` – label for the new source.

  > `changeprime` – change the prime source from the proposal coordinates.

> `userasprime` – change the prime source to the user added coordinates.
> `ra` – RA of user's source.
> `dec` – Dec of user's source.

- `rgsproc orders='1 2' entrystage=spectra finalstage=fluxing bkgcorrect=no exclsrcsexpr='INDEX==1&&INDEX==3'`

  > `orders` – dispersion orders to extract.
  > `entrystage` – entry stage to the pipeline (see Sec. 6.4).
  > `finalstage` – exit stage for the pipeline (see Sec. 6.4).
  > `exclsrcsexpr` – expression to identify which source(s) should be excluded from the background extraction region.

6.4 PIPELINE ENTRY STAGES

There are five stages at which the user can enter or leave the pipeline:

- `events` – Creates attitude time series, attitude-drift and housekeeping GTI tables, pulse height offsets, the source list, and unfiltered, combined event lists.
- `angles` – Corrects event coordinates for aspect drift and establishes the dispersion and cross-dispersion coordinates.
- `filter` – produces filtered event lists, creates exposure maps.
- `spectra` – Constructs extraction regions and source and background spectra.
- `fluxing` – creates "fluxed" spectra for quick data inspection and response matrices.

Provided the filtered event list is retained, users can apply their own filtering by entering the pipeline at the `filter` stage.
Changes in the extraction region sizes can be handled by entering at the `spectra` stage.
If the coordinates of the source differ from those in the original proposal, the pipeline must be run from `events`.
Extraction of spectra with different binning can be achieved at the `spectra` stage.
Recalculation of the response matrices can be done in the final `fluxing` stage.

6.5 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:

```
xspec

    XSPEC>data 1:1 PiiiiiijjkkaablllSRSPEC1mmm.FIT 1:2 PiiiiiijjkkaablllSRSPEC2mmm.FIT
    XSPEC>ignore bad
    XSPEC>model phabs*mekal
```

etc...

6.6 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 – 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 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, and the correct approach would normally be to use an optimization of the two.

6.6.1 Spectral Rebinning

By grouping channels in appropriately large numbers, the combined signal-to-noise of groups will jump into the Gaussian regime. The FTOOL *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.

- `grppha`

  > `Please enter PHA filename[] PiiiiiijjkkaablllSRSPEC1mmm.FIT`

  > `Please enter output filename[] !PiiiiiijjkkaablllSRSPEC1mmm.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 over-estimated in the resulting spectra.
An alternative approach is to bin the data during spectral extraction. The easiest way to do this is call the RGS pipeline after the pipeline is complete. The following rebins the pipeline spectrum by a factor 3:

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

  > `orders` – dispersion orders to extract.

  > `rebin` – wavelength rebinning factor.

  > `rmfbins` – number of bins in the response file (>3000 is recommended by the SAS documentation).

  > `entrystage` – entry stage to the pipeline (see Sec. 6.4).

  > `finalstage` – exit stage for the pipeline (see Sec. 6.4).

One disadvantage of this approach is that one can only choose integer binning of the original channel size. To change the sampling of the events the pipeline must be run from `angles` or earlier:

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

  > `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; e.g., the accuracy of redshift determinations will be degraded, transition edges will be smoothed and neighboring lines will become blended.

6.6.2 Maximum-Likelihood Statistics

The second alternative is to replace the χ^2-minimization scheme with the Cash maximum-likelihood scheme 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 ensure that one has 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`

6.7 ANALYSIS OF EXTENDED SOURCES

6.7.1 Region masks for extended sources

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:

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

  > `orders` – dispersion orders to extract.

  > `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. Consequently it is prudent to also increase the width of the PI masks using the `pdistincl` parameter in order to prevent event losses.

6.7.2 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 called, e.g. xsource.mod, containing three lines of input. 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

> 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, e.g. −1, −2...):

xspec

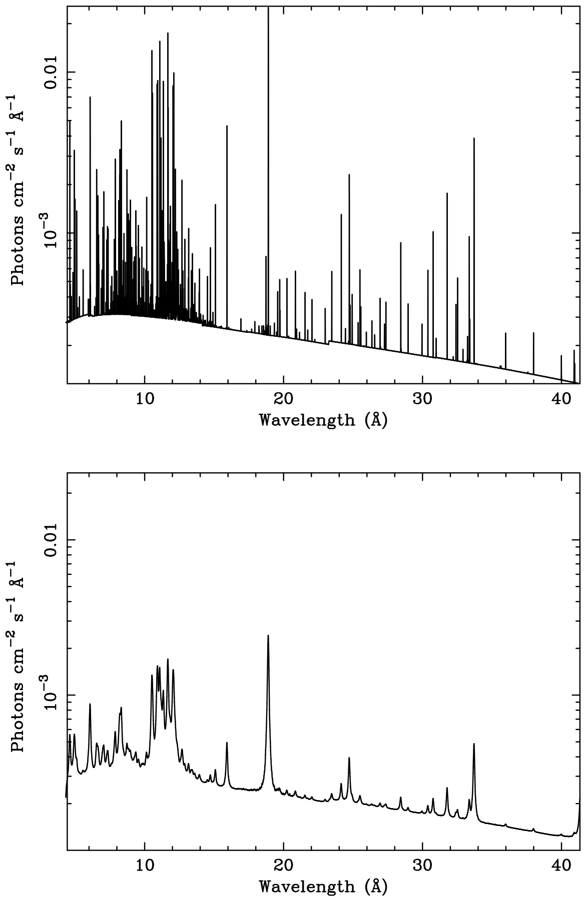
```
XSPEC>data P0108460201R1S004SRSPEC1003.FIT
XSPEC>ignore bad
XSPEC>xset rgs_xsource_file xsource.mod
XSPEC>model rgsxsrc*wabs*mekal
rgsxsrc:order>−1
wabs:nH>1
mekal:kT>2
mekal:nH>1
mekal:Abundanc>1
mekal:Redshift>
mekal:Switch>0
mekal:norm>1
XSPEC>renorm
XSPEC>fit
XSPEC>setplot device /xs
XSPEC>setplot 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
```

Fig. 6.6 compares a point source model with an extended source counterpart.

6.7.3 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. Contact John Peterson <*jrpeters@astro.columbia.edu*>.

Figure 6.6: 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.



6.8 A MORE-OR-LESS COMPLETE EXAMPLE

The AB Doradus PV ODF data (ObsID: 0134520301 from orbit 0205) have been used for a reasonably complete example of RGS data reduction. The script can be found at:

- `ftp://heasarc.gsfc.nasa.gov/xmm/data/examples/rgs/RGS_ABC.SC`

The lines of the script for setting up and running SAS are specific to installation at GSFC and so will need to be modified as appropriate. The script uses the SAS command-line interface and goes through the following steps:

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

2) Initializes SAS.

3) Creates a Current Calibration file.

4) Builds an ODF summary file.

5) Constructs a GTI file based on background activity.

6) Runs the RGS pipeline.

7) Makes a few useful data inspection products.

8) Fits a model to one of the resulting spectra.

Chapter 7

First Look – OM Data

The OM is somewhat different from the other instruments on-board XMM-Newton, and not only because it is not an X-ray instrument. Since the OM pipeline products can be used directly for most science analysis tasks, a re-processing of the data is not needed in most cases. So in **principle** one can ignore the files in the ODF directory and go directly to §7.1, which describes the files in the PPS (or PIPEPROD) directory. Users interested in re-processing of the OM data can go directly to §7.2 which explains the pipeline processing. For the analysis of OM data obtained in FAST or GRISM mode, however, a re-processing of data is needed, which is explained in more detail in §7.2.2 and §7.2.3.

7.1 PIPELINE PRODUCTS

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 V6, 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 the following nomenclature:

- `PjjjjjjkkkkOMlmmmNNNoooo.zzz`

  `jjjjjj` – Proposal number

  `kkkk` – Observation ID

  `l` – S (scheduled), U (unscheduled), or X (general)

  `mmm` – A number either of the form of 005/6 or 401/2

  `NNN` – File ID (see Table 7.1)

  `oooo` – Either 0000 (high res) or 1000 (low res)

  `zzz` – File type (FTZ, PNG, PDF, HTM,..)

The pipeline produces a summed sky image for each of the filters in low resolution. The results are in files with 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)

Table 7.1: OM Pipeline Processing data files.

Group ID	File ID	Contents	File Type	View With
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*

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 identification/coupling of the files (product/P0123456789OMS410SIMAGE1000.FIT) are identical to the ones described at the beginning of the previous section.

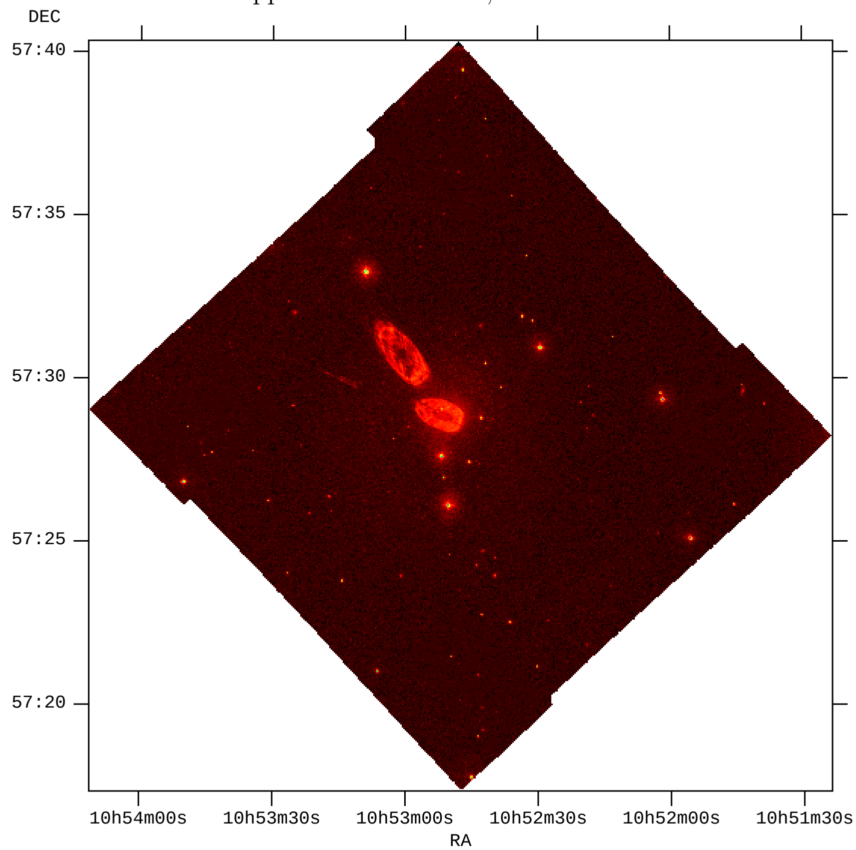
Table 7.2: Some of the important columns in the SWSRLI FITS file.

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

Creating images with the OM products

If the observation data products do not contain any mosaic files for all the exposures but about 10 files per filter, it means that they were processed with an older version of the pipeline. In this case we recommend a re-processing of the OM data with SAS V6 using *omichain*. The *omichain* task automatically produces one single final file per filter. If there are multiple OM exposures of the same field, *ommosaic* can be used to create one single image covering the full field of view. One must specify which files are to be added (the program does not do this automatically) so one must know which files were produced for which filters, and at which

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.



resolution. The task *ommosaic* be also be used to combine images observed with different filters. Note that the final image is not corrected for coincidence losses or for deadtime.

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 to retrace the following procedures. Figure 7.1 shows the merged V-band image from the Lockman Hole SV1 observation using the *ommosaic* task.

You can also use a program written at the NASA/GSFC *XMM-Newton* GOF. The task is meant to be used on files in the PPS directory (which contains the outputs of the OM pipeline). It produces a final event and exposure images in sky coordinates for each of the filters used in the observation. Low and high-resolution images are treated separately. The task requires that FTOOLS and Perl are installed on your machine and the script must be run from a writable directory which contains the OM files. The tar file with the script is available at the GOF site: `ftp://legacy.gsfc.nasa.gov/xmm/software/om_tools/om_prod_all.tar.gz`. The program is fairly easy to use, and to modify. If any problems arise with the task please contact the GOF.

7.1.2 Fast Mode

Most OM data are obtained in Imaging mode. If the default included the Fast mode, there will be an additional event list file corresponding to the Fast window (*FAE.FIT). We suggest a re-processing of data obtained in FAST mode, which is explained in detail in §7.2.2.

7.1.3 Grism Mode

OM Grism data require a re-processing, which is explained in detail in §7.2.3.

7.2 RE-PROSESSING OF ODF FILES

The OM can operate in IMAGING, FAST, and GRISM mode. Each of these modes has dedicated *chain* commands, *omichain*, *omfchain*, and *omgchain*. If you run these chains, it is helpful to inspect the `sas_log` file to get a detailed list of the performed tasks. In general, the ODF file names for the data will look like this:

- `mmmm_iiiiiijjjj_OMbeeeccfff.zzz`

  `mmmm` – Revolution orbit number

  `iiiiii` – Proposal identifier number

  `jjjj` – Observation ID (target and exposure)

  `b` – Flag for scheduled (S) or unscheduled (U) exposures

  `eee` – Exposure number within the observation

  `cc` – CCD or OM window Identifier

  `fff` – Data identifier (imaging, timing, reduced imaging...)

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

The IMAGING, FAST, and GRISM chains (*omichain*, *omfchain*, and *omgchain*) are described below. We have also written an equivalent to the *omichain* which allows one to vary the input parameters of each task, or to run the pipeline on only a subset of the data.

7.2.1 Imaging Mode

The Stray-light Problem

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.

ODF Products

In `IMAGING` mode, OM files in the ODF directory looks like:

```
0070_0123700101_OMS00400IMI.FIT  0070_0123700101_OMS42200RFX.FIT
0070_0123700101_OMS00400RFX.FIT  0070_0123700101_OMS42200THX.FIT
0070_0123700101_OMS00400THX.FIT  0070_0123700101_OMS42200WDX.FIT
0070_0123700101_OMS00400WDX.FIT  0070_0123700101_OMS42201IMI.FIT
0070_0123700101_OMS00401IMI.FIT  0070_0123700101_OMS42300IMI.FIT
0070_0123700101_OMS00500IMI.FIT  0070_0123700101_OMS42300RFX.FIT
0070_0123700101_OMS00500RFX.FIT  0070_0123700101_OMS42300THX.FIT
...
```

For each exposure there are: an image file (IMI), a tracking history file (THX), and a window data auxiliary file (WDX). There is one non-periodic (NPH) and periodic (PEH) housekeeping file per observation. In order to run a task, you will also need three files that are not specific to the OM. The first one (`0070_0123700101_SCX00000SUM.SAS`) is an ASCII file containing a summary of the observation, which is created by the *odfingest* task (§ 4.5.2):

```
/XMM/Mydata/ODF:  more 0070_0123700101_SCX00000SUM.SAS
// ----------------------------------
// XMM-Newton Science Analysis System
// ----------------------------------
//
```

```
// ODF Summary File
// By: odfingest(odfingest-3.4) [xmmsas_20020109_1903-no-aka] on 2002-01-25T20:42:44.000
//
//
// Directory where the ODF constituents were found.
// This may have to be edited to match the local file system structure.
//
PATH /XMM/Mydata/ODF
//
```

Please note that the keyword `PATH` **can be edited** to match your current location of the data.

The second general file (`0070_0123700101_SCX00000TCS.FIT`) is the spacecraft time correlation file, while the third (`0070_0123700101_SCX00000ATS.FIT`) contains the spacecraft attitude file.

Re-processing of Imaging data can be done automatically by using *omichain*. The task *omichain* runs on filters specified by the user. If no arguments are given, the chain runs on all the files present in the $SAS_ODF directory. If the *omichain* tasks are re-run one by one, there may be small differences between the files obtained in this manner and the pipeline products in the PPS directory. The main reasons for the differences are improvements made to the *SAS* software, the type of products produced by the pipeline (for example, only the most recent products have a final image for each filter), and some changes in the calibration products.

The following explains the step-by-step processing of OM files. At the end of this section, a script is provided which goes automatically through all of the steps described below. The script is essentially an annotated version of *omichain* and shows what the processing does. We suggest that the user goes through all of the steps at least once manually before using the script.

Preparation for Data Processing

If one wants to group the ODF files by filter values, one must extract the `FILTER` keyword from their headers. This can be done by using the FTOOLS task *fkeyprint*, e.g.:
`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 7.3.

Table 7.3: 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

We have written a script which goes through the complete list of files and gives back the filter used for each exposure. The script is available at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/om_tools/file_examine.shell`

Running this script provides a list of files and their associated filters. The details of the association are less complicated than it may appear at first. In the standard configuration (the so-called Rudi-5 mode) one gets exposures in groups of 5, in high- and low-resolution mode, for a total of 10 files per filter. The high-resolution mode covers the same small central window in all five exposures while the low-resolution mode covers different parts of the detector in each of the 5 exposures. The sum of the low-resolution exposures covers the entire FOV.

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. Here is an example of what running the script, *file_examine.shell*, produces:

```
/XMM/Mydata/ODF:  ./file_examine.shell
0070_0123700101_OMS00400IMI.FIT FILTER   V
0070_0123700101_OMS00401IMI.FIT FILTER   V
0070_0123700101_OMS00500IMI.FIT FILTER   U
0070_0123700101_OMS00501IMI.FIT FILTER   U
0070_0123700101_OMS00600IMI.FIT FILTER   WHITE
0070_0123700101_OMS00601IMI.FIT FILTER   WHITE
0070_0123700101_OMS41500IMI.FIT FILTER   V
0070_0123700101_OMS41501IMI.FIT FILTER   V
0070_0123700101_OMS41600IMI.FIT FILTER   V
0070_0123700101_OMS41601IMI.FIT FILTER   V
0070_0123700101_OMS41700IMI.FIT FILTER   V
0070_0123700101_OMS41701IMI.FIT FILTER   V
0070_0123700101_OMS41800IMI.FIT FILTER   V
0070_0123700101_OMS41801IMI.FIT FILTER   V
0070_0123700101_OMS41900IMI.FIT FILTER   U
0070_0123700101_OMS41901IMI.FIT FILTER   U
```

As noted above, the last three digits are paired so that in general one gets a low- and a high-resolution image. In this example the following images for the V filter are produced: 00401/00400 (inner part of the low-resolution image plus high-resolution frame of the inner part) 41501/41500 (left-hand frame of the low-resolution image plus high-resolution frame of the inner part), 41601/41600 (bottom frame of the low-resolution image plus high-resolution frame of the inner part), 41701/41700 (right-hand frame of the low-resolution image plus high-resolution frame of the inner part), and 41801/41800 (top frame of the low-resolution image plus high-resolution frame of the inner part). This means that usually five high-resolution images are produced which can be co-added to achieve deeper exposures. Please be aware that one should **NOT** add low-resolution and high resolution images, even if they cover the same part of the FOV (e.g., one can not add 0070_0123700101_OMS00401IMI.FIT and 0070_0123700101_OMS00400IMI.FIT).

Once one has decided which data to process (for example one exposure of one filter taken with a certain resolution), one should make sure that

1) A Calibration Index File has been created using *cifbuild* (§ 4.5.1).

2) A summary file of the ODF constituents has been created using *odfingest* (§ 4.5.2).

3) One set of exposures has been chosen on which to run *omprep*.

As an example, we use the first high-resolution exposure for the Lockman Hole SV1 data and have copied them into the `/XMM/Mydata` directory. The files associated with the exposure are:

```
/XMM/Mydata:  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).

Examine the Guide Star Record

- Both the ODF and the THX files should be processed by the *omprep* task before running any other SAS task (except *cifbuild* and *odfingest*).
  - `omprep set='Mydata/0070_0123700101_OMS00400THX.FIT'`
    `pehset='Mydata/0070_0123700101_OMX00000PEH.FIT'`
    `nphset='Mydata/0070_0123700101_OMX00000NPH.FIT'`
    `wdxset='Mydata/0070_0123700101_OMS00400WDX.FIT'`
    `outset='Mydata/0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'`
    `modeset=0`
    - > `set` – Tracking history file
    - > `pehset` – Periodic Housekeeping file
    - > `nphset` – Non-Periodic Housekeeping file
    - > `wdxset` – Window Data Auxiliary file
    - > `outset` – Output file
    - > `modeset` – Are these slew data (0=no)?
- Now the output THX file is ready to be used by the rest of the SAS tasks. One can examine the OM tracking history using the task *omdrifhist*. The output is a postscript file containing plots and statistics on the tracking history.
  - `omdrifthist set='Mydata/0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'`
    `plotfile='Mydata/0070_0123700101_OMS00400THX_drift.ps'`
    `trackradius=0.5 hardcopy=yes pages='1 2'`
    - > `set` – THX file output of the *omprep* task
    - > `plotfile` – Output name
    - > `trackradius` – Radius of pointing accuracy.
    - > `hardcopy` – Yes/no ?
    - > `page` – Pages to plot (maximum pages produced is 2)
- Now One can inspect the output PS file. The other check on the tracking is to look at the count rates of the guide stars. To do this, use the task *omthconv* to produce a FITS file containing up to 10 columns with the guide stars' count rates.
  - `omthconv thxset='Mydata/0070_0123700101_OMS00400THX_OUT_OMPREP.FIT'`
    `nphset='Mydata/0070_0123700101_OMX00000NPH.FIT'`
    `outset='Mydata/THX_trackingStar.FIT'`
    - > `thxset` – Corrected THX file (output from *omprep*)
    - > `nphset` – Non Periodic Housekeeping data
    - > `outset` – Output file

Examine the Images (IMI) files

Bad Pixels

IMI ODF files containing the OM images should also be corrected before running any other SAS task. The arguments of the *omprep* task are identical to the ones given for the THX file except for the input file which is now the IMI file.

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

  > for parameter definitions see above, except for the input file which is now the IMI file.

Once *omprep* has been run, the *omcosflag* task looks at the (corrected) 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.

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

  > `samplefactor` – Spatial oversampling factor (default 1)

  > `timefactor` – Temporal sampling factor (default 1)

  > `set` – Corrected IMI file

  > `thxset` – Corrected THX file

  Note: The output file is a *modified* IMI file. One *can not* run this task twice as it fails if it detects an existing `QUALITY` extension. To avoid problems, keep a copy of the original file (output from `omprep`) before running `omcosflag`.

  Note: The `timefactor` allows sub-sampling of the spacecraft jitter for tracking shifting of the bad pixel map. Although this has not yet been studied in detail, it appears that the tracking is generally so good that sub-sampling does not seem necessary. This parameter should be set to one (the default).

Flat Field Generation

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.

- `omflatgen outset= 'Mydata/OUT_FLATGEN.FIT'`

  > `outset` – Name of the output file

Once the `OUT_FLATGEN.FIT` file has been created, one should run the *omflatfield* task 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.

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

  > `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

  Note: Not (too) surprisingly *omflatfield* produces the following warning:
  `** omflatfield:  warning (Uniform flatfield- no correction to image will be applied)`

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.

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

  > `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

Perform Source Detection

The task *omdetect* employs a simple two-stage process to locate sources in an OSW image. OM source positions are corrected for a 0.5 pixel position error in both FAST and IMAGE Mode exposures in SAS V6. The first stage in source detection is to determine the background. The second stage is an island search, in which sets of pixels above the sigma significance cut-off are identified and grouped into individual objects. The task has a lot of parameters (see below) but only `set` and `outset` are mandatory.

- `omdetect set='Mydata/0070_0123700101_OMS00400OUT_OMMODMAP.FIT' outset='Mydata/0070_0123700101_OMS00400IMI_OUT_OMDETECT.FIT' nsigma=6 contrast=0.001 background='Mydata/0070_0123700101_OMS00400BACKGROUND.FIT' levelimage='Mydata/0070_0123700101_OMS00400LEVELIMAGE.FIT' signifimage='Mydata/0070_0123700101_OMS00400SIGNIFIMAGE.FIT' smoothsize=64 boxscale=3 maxscale=1 boximage='Mydata/0070_0123700101_OMS00400BOXIMAGE.FIT' pixelconnect=1 flatset='Mydata/OUT_FLATGEN.FIT' mod8set='Mydata/0070_0123700101_OMS00400PPSMODE8SET_OUT.FIT' outputregionfile=yes regionfile='Mydata/0070_0123700101_OMS00400oswList.reg'`

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

  > `outset` – Name of the output source list file

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

  > `contrast` – Blended source OK if source flux larger than contrast X total flux

  > `background` – Name of output background image

  > `levelimage` – Name of output image before deblending

  > `signifimage` – Name of output significance (σ) image

  > `smoothsize` – Size of smoothing box for background determination

  > `boxscale` – Minimum sliding box size for source detection

  > `maxscale` – Maximum binning to search

  > `boximage` – Name of output sliding box image

  > `pixelconnect` – Not used (keep to 1)

  > `flatset` – Name of input flat field image

> `mod8set` – Name of the modulo-8 noise map (mod8set output parameter from *ommodmap*)

> `outputregionfile` – Do you want to produce an *saoimage* region file?

> `regionfile` – Name of saoimage region file

Note: *omdetect* does a variable job with the stray-light features and it may sometimes be fooled by them. One way to separate them from real detections is to look at the FWHM max and min parameter in the source list. Spurious source detections associated with stray-light features will have large value associated with these parameters.

Convert Source Counts to Magnitudes

The task *ommag* converts the list of given source counts to magnitudes in the appropriate instrumental band passes. The accuracy is estimated to be a few tenths of a magnitude. SAS V6 takes a new PSF for the UVW1, UVM2, and UVW2 filters into account, leading to an improved OM photometry on the order of tenths of a magnitude.

- `ommag set='Mydata/0070_0123700101_OMS00400IMI_OUT_OMDETECT.FIT' wdxset='Mydata/0070_0123700101_OMS00400WDX.FIT'`

> `set` – Input list (output of *omdetect*)

> `wdxset` – Window Data Auxiliary file

Note: There is a "recipe" to convert the UV count rates to flux. The recipe was provided by Alice Breeveld (MSSL) and can be accessed at:
`http://xmm.vilspa.esa.es/sas/documentation/watchout/uvflux.shtml`

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.

- `omatt set='Mydata/0070_0123700101_OMS00400OUT_OMMODMAP.FIT' sourcelistset='Mydata/0070_0123700101_OMS00400IMI_OUT_OMDETECT.FIT' ppsoswset='Mydata/0070_0123700101_OMS00400FINAL_IMAGE.FIT' device='/NULL' usecat=no tolerance=3 catfile=''`

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

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

> `ppsoswset` – Output name for the corrected sky image

> `device` – Output device

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

> `tolerance` – Tolerance for catalog search in arc seconds

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

Note: 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`.

Note: 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.

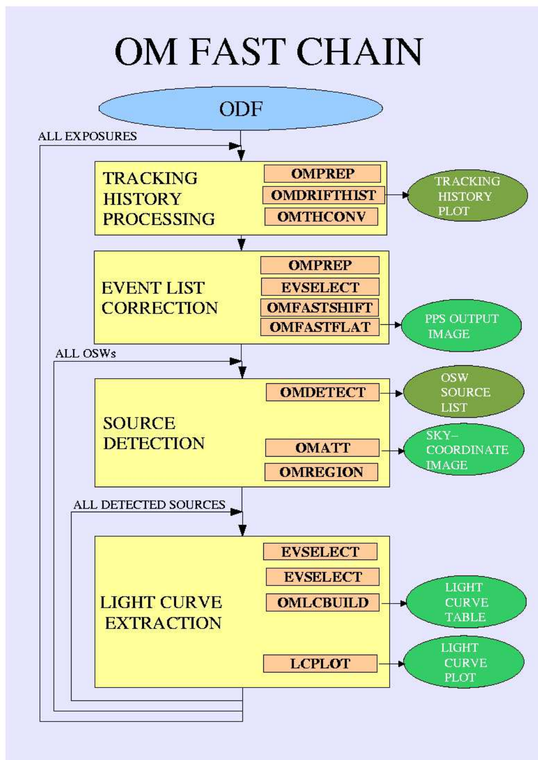
There is a script which does all this step by step and allows one to run the pipeline only on the desired file. The script is available at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/om_tools/omproc_gof`.
Please contact the GOF if you have any problems with it.

7.2.2 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.vilspa.esa.es/sas/current/doc/index.html`. You can also access the general description of the task at: `ftp://legacy.gsfc.nasa.gov/xmm/doc/fastmode.ps.gz`. A summary of the task is shown in Figure 7.2.

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



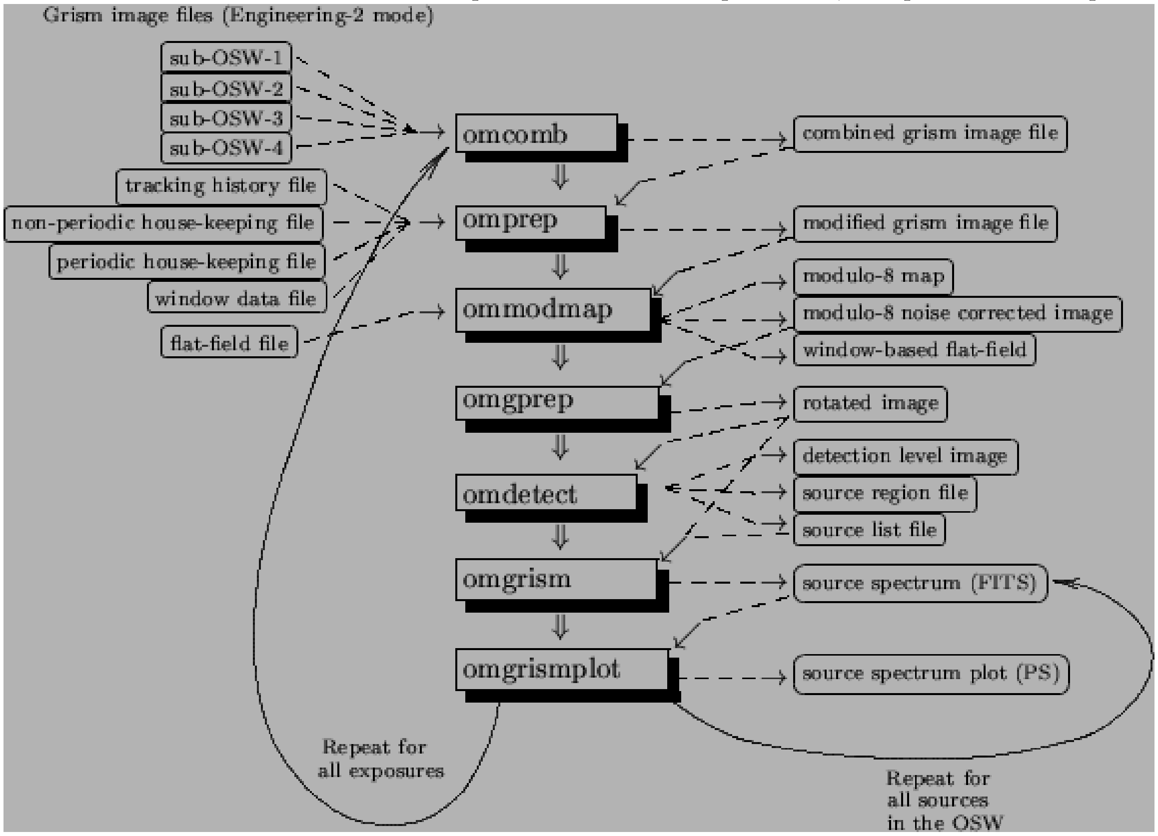
7.2.3 Grism Analysis

SAS V6 allows, for the first time, the analysis of data obtained with the OM grisms. A new 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 new 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*.

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 V6 also includes a new interactive task, *omgsource*, which allows the user to select with the cursor the spectrum to be extracted.

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.



The task *omgchain* has many parameters, but none of them are mandatory. Below is a description of the calling sequence and the individual parameters.

- `omgchain inpdirectory=MyData/ODF outdirectory=MyData 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`

  > `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)

  > `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 Gaussian fit

> `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

Note: if a source is not detected by *omdetect*, or does not fall within the grism window, *omgchain* will run without warning, but will not produce output files.

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