AN INTRODUCTION TO XMM-NEWTON DATA ANALYSIS

NASA/GSFC XMM-Newton Guest Observer Facility

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

Version 1.0

13 March 2002

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 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 is still under development. 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 Simon Rosen, Alice Breeveld, Jamie Kennea and Rudy Much for all the information they provided before and during the writing of this document.

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/news/xmmhelp_top.html`

  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/user/mplan/`

  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/news/advance_plan1.html`

2.4 PUBLIC DATA ARCHIVES

- SOC SV Public Data Archive:

    `http://xmm.vilspa.esa.es/public/xmm_sas_sv_top.html`
    Cal/PV data can be accessed under the "News" items at the left of the SOC main pages.

- GSFC Archive Mirror Site (SAS Validation and US GO data, when public):

    `http://heasarc.gsfc.nasa.gov/docs/xmm/xmmhp_archive.html`

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/user/calib_top.html`

- 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/sas/`

- Current status of SAS:

    `http://xmm.vilspa.esa.es/sas/docupdates/`

- 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/sas/`

    (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/user/uhb_top.html`

- 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: SV, PV, and GO CDs

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/sas/`). These documents, as well as this *ABC Guide* are included on the data CDs provided to observation PIs (see § 3.5).

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`.

3.2 SV AND PV DATA

The first use of *XMM-Newton* data for many people will be with Science Validation (SV, two groups of observations denoted by SV1 and SV2) and Calibration/Performance Validation (Cal/PV) data from the archive. The pool for these data sets are the observations conducted primarily early in the mission for the calibration of the scientific instruments and the testing of their performance. These data are becoming public through the main archive at the SOC and are also mirrored by the NASA GOF at GSFC. The fundamental files are the same as listed below for the GO CDs, however they are grouped differently and collected in TAR files. (TAR files can be ungrouped by the command `tar xvf` *filename*.) The pipeline products (see § 3.5.3) TAR files will have names like:

- `PPiiiiiijjkkAAAAAA000_0.TAR`

  `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

  `AAAAAA` – group identifier (see Table 3.1)

Not all groups listed in Table 3.1 will always be available for all observations. For example, if the OM was not operational for the observation, no OM files will be included.

The ODF data (see § 3.5.2) TAR file will have a name like:

- `mmmm_iiiiiijjkk.tar.gz`

  `mmmm` – orbit number, NOTE: SV1 data only `iiiiijjkk` – observation number

In addition, the SV data will be accompanied by compressed postscript and PDF files containing a science validation report.

3.3 THE GO CD

This section will describe the hopefully easy process of retrieving the *XMM-Newton* data from the CD-ROM provided to the PIs of GO and GT observations. In some cases a system administrator or someone with root access to the machine with the CD drive may be necessary.

1) If you insert the CD and a window with the contents opens on your desktop, or you can see the files upon doing a directory listing of the CD drive (e.g., `ls /mnt/cdrom`, `ls /cdrom`, go to Item 5).

2) If you cannot immediately view the contents of the CD, you may be able to mount the CD and then view it's contents. To mount the CD, some variation of the command `mount /mnt/cdrom` may work.
   Warning: These commands could cause serious problems with your operating system if used incorrectly. If after executing the mount command you can list the files in `/mnt/cdrom`, then go to Item 5).
   At the end of the process unmount the CD before ejecting it with the command `umount /mnt/cdrom`

3) If you cannot view the contents of `/mnt/cdrom`, and you cannot use the mount command due to permission problems, please contact your computer support personnel for the proper procedure for using a CD-ROM on your system.

4) If all attempts to mount the CD-ROM fail on your machine, please contact the NASA/GSFC *XMM-Newton* GOF (`xmmhelp@athena.gsfc.nasa.gov`).

5) Some Pipeline Processing products can be used directly from the CD, for others (most notably the ODF data) the data may need to be copied to disk (see § 3.4 for what data is where). Currently the CD-ROMs are formatted in such a way that when the files are transferred to disk the file names are in lower case. Many SAS tasks require upper case file names, however. **NOTE:** The tasks which reprocess the data require upper case file names. Therefore, the last step in preparing the data for use is to insure that all the file names are capitalized. To do so the user can retrieve a program from the *XMM-Newton* GOF anonymous ftp area that will prepare the files: `ftp://legacy.gsfc.nasa.gov/xmm/software/upcase.pl`. This is a *perl* script which simply renames (`perl upcase.pl`) the files and folders to upper case versions. A second *perl* script named *locase.pl* is also available in the same directory if for some reason the user wants to return the files to their original names.
   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 will 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 has written 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`.

3.4 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 can arrive on separate CDs. 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 $\sim$ 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 of 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). (For the PV and SV data these summary files can be found in the `PPSGRA` TAR file.) Second, to quickly access images from the various instruments examine the PPSGRA Pipeline Products page (§ 3.5.3).

3.5 CD Contents

XMM-Newton data come on one or more CDs depending on the size of the data set. This is, in turn, dependent not only on the length of the exposure and intensity of the sources in the observation but also on whether the observation suffered from background flaring events. Typical EPIC count rates for observations are in the low tens of counts per second. Background flare events can raise that count rate to over five hundred counts per second.

The data products on your CD(s) can be considered to come in three groups, although the division on the CD(s) may be different. The first contains information on the CD while the others contain the observation data files. The data file groups are in separate directories, and are Observation Data Files (ODF)files and Pipeline Processing (PP) 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.

Also included on the CD is a *xmmdoc/* directory which contains:

- `readme.html` – README
- `ssc-lux-sp-0004-1_1.ps.gz` – Data product specifications
- `xmm-sas_users_guide.ps.gz` – SAS Users Guide
- `xmm-soc-dfhb.ps.gz` – Data Files Handbook
- `xmm-soc-hw-tn-0014.pdf` – How to use *XMM-Newton* CDs
- `xmm_abc_guide.pdf` – This guide
- `xmm_abc_guide.ps.gz` – This guide
- `xmm_banner.jpg` – *XMM-Newton* banner
- `xmm_uhb.pdf` – *XMM-Newton* Users Handbook
- `xmm_uhb.ps.gz` – *XMM-Newton* Users Handbook

NOTE: For observation data sets going to US PIs, the GSFC GOF makes the data available on line after PGP encryption. For these data there are three directories containing the following groups of files.

- MISC – Miscellaneous files describing the contents of the CD
- ODF – The ODF (raw) data files
- PIPEPROD – The pipeline processed data products

3.5.1 CD Information

The first CD of the set will have a file in the main directory with a name like:

- `X_nnnnnnnn_iiiiiijjkk___ll.SET`

  `nnnnnnnn` – date tag for the processing

  `iiiiiijjkk` – observation number

  `ll` – number of CDs in the set

which will list the ODF and Pipeline Processed files which are included on the CD(s) and the CD where they can be found.

For each CD there will be a main directory which will have a file with a name like:

`X_nnnnnnnn_iiiiiijjkk_mm_ll.DSK`

`mm` – CD identifier, e.g., `01_02` means the first CD of a two CD set

(where the other strings have the same meaning as above) which will list the files on the CD. There is also an information file on all CDs, with the name `VOLDESC.SFD`, which lists processing and other information.

3.5.2 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 is providing a script to rename the files (see Item 5) of § 3.3).

3.5.3 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, as in the TAR files for the SV and PV data. (The number of groups can vary depending on the number of operational instruments, e.g., if the OM is turned off there will be 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 7.2.
[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/sas/`), developed by the Survey Science Centre (SSC) and Science Operations Centre (SOC), is a suite of about 270 programs and scripts that perform data reduction, extraction, and some analysis of *XMM-Newton* data. The Pipeline Processing System (PPS), comprised of a subset 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/sas/` (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. 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/install.html`.

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/ccf/constituents/extras/`.

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 command `env | grep SAS`.

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

```
rgsproc --addusersource='no'
rgsproc --addusersource="no"
```

However,

```
rgsproc -addusersource=F
rgsproc -addusersource=no
rgsproc -addusersource='no'
rgsproc -addusersource="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 addusersource=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 (see Item 5) of § 3.3).
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 § 5.6.1). 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 up 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 – EPIC Data

So, you've received an XMM-Newton EPIC data set. What are you going to do with it? 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. In addition, SAS will be required at any rate for the production of detector response files (rmfs and arfs) and other observatory-specific requirements.

5.1 USING PIPELINE PROCESSED DATA PRODUCTS

The Pipeline Processing produces quite a number of useful products which allow a first look at the data, but can overwhelm the user by their sheer numbers. With 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. 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` – observation ID - target number in proposal

  `kk` – observation ID - observation 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

  `iiiiii` – proposal number

  `jj` – observation ID - target number in proposal

  `kk` – observation ID - observation number for target

  `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

  `lll` – exposure number

  `CCCCCC` – file identification (Table 5.1)

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

  `mmm` – source number

  `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

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

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*. 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

  `iiiiii` – proposal number

  `jj` – observation ID - target number in proposal

  `kk` – observation ID - observation number for target

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

  `lll` – exposure number within the observation

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

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
```

In the following sections note that the SAS *xmmselect* GUI provides a very simple method for producing and displaying images and light curves.

5.2.1 Initialize SAS and Prepare the Data

For the following demonstration the PP data are assumed to be in the directory /PPDATA, the ODF data (with upper case file names) are in the directory /ODF, the analysis is taking place in the directory /PROC, and the CCF data are in the directory /CCF.

1) gunzip the PP event list to be examined, and for practical purposes shorten the file name as well, e.g.:
   `mv PjjjjjjkkkkM1S001MIEVLI0000.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) Create a CIF file using the SAS task *cifbuild* (§ 4.5.1).

   – `cifbuild withccfpath=no analysisdate=now category=XMMCCF fullpath=yes`

2c) Prepare the ODF data by using the SAS task *odfingest* (necessary for many SAS tasks) (§ 4.5.2).

   – `odfingest odfdir=$SAS_ODF outdir=$SAS_ODF`

5.2.2 Create and Display an Image

3) Create an image in sky coordinates by using the task *evselect*.

   – `evselect table=/PPDATA/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 &`.

5.2.3 Create and Display a Light Curve

4) Create a light curve of the observation using the task *evselect* then display with *dsplot*.

   – `evselect table=/PPDATA/mos1.fits:EVENTS withrateset=yes rateset=rate.fits maketimecolumn=yes timecolumn=TIME timebinsize=10`

     > `table` – input event table.
     > `withrateset` – make an light curve.
     > `rateset` – name of output light curve file.
     > `maketimecolumn` – control create of time column
     > `timecolumn` – time information.
     > `timebinsize` – time binning (seconds).
     > `maketimecolumn` – control create of time

   – `dsplot table=rate.fits x=TIME y=COUNTS &`

     > `table` – input event table.
     > `x` – column for plotting on X axis.
     > `y` – column for plotting on Y axis.

5.2.4 Filter the Data and Create a New Event File

5) Next apply some filtering to the data using the task *evselect*. The following "expression" will select only good events (flag value equal to zero) with CCD patterns less than or equal to 12 (MOS data, less than or equal to 4 for PN data), and with PI channels in the range 100 eV to 15 keV.
   `(FLAG == 0)&&(PATTERN <= 12)&&(PI in [100:15000])`
   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 to 300 eV will eliminate much if not all of the rest. An output file will be created for further processing.

   – `evselect table=/PPDATA/mos1.fits:EVENTS withfilteredset=yes expression='(FLAG == 0)&&(PATTERN <= 12)&&(PI in [100:15000])' filteredset=mos1-filt.fits filtertype=expression keepfilteroutput=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

6) If necessary, add a temporal filtering clause to the *evselect* selection "expression". To exclude a contaminated time interval (e.g., a soft proton flare observed in the light curve plot of § 5.2.3) use the syntax, e.g., `&&!(TIME in [95027600:95030540])` to exclude events in the time interval 95027600 to 95030540. (The "!" symbol stands for the logical "not".) The full "expression" would then be:
   `(FLAG == 0)&&(PATTERN <= 12)&&(PI in [100:15000])&&!(TIME in [95027600:95030540])`

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. The extraction of region-specific data (e.g., source spectra and light curves) is simplified by using the GUI because of the treatment of selection regions. To do so invoke the SAS GUI (i.e., enter `SAS &` from the command line) and then invoke the *xmmselect* task. **NOTE:** To run *xmmselect* from the GUI in SAS v5.2, the full path for the CIF file and the ODF directory must be set (e.g., as in § 5.2.1).

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 "Image" button, and then clicking on the "Run" button on the pop-up *evselect* GUI. (Note: to select the file using the pop-up GUI when *xmmselect* is first invoked, first click on the ".." button on the right which will bring up a second GUI for the file selection. Click on the desired event file in the left-hand column, click on "EVENTS" in the right-hand column with the right-hand button when it appears, then click on "select" when the open/select GUI pops up, and then click OK. Click "Run" and then Click "No" in the pop-up GUI that appears.) 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(24900,24300,300))`
The `circle` parameters are the X and Y positions and the radius of the circle in units of $0''.05$.

3) To extract a spectrum, first click the circular button (circular buttons are for 1-d analysis, e.g., TIME for a light curve and and PI channel for a spectrum, while square buttons are for 2-d analysis, e.g., X,Y for a sky image) next to the "PI" column on the *xmmselect* GUI. 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 `filteredset` to `source.pi`. If the canned responses are going to be used, the `spectralbinsize` must be set to 15 for the MOS and 5 for the PN. `withspecranges` must be checked, `specchannelmin` set to 0, and `specchannelmax` set to 11999 for the MOS and 20479 for the PN.

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 and then click the "2D region" button. This will transfer the region description of both circles to the "Selection expression". This will need to be edited. For example, for an annulus it should be as follows:
`((X,Y) IN circle(24900,24300,1200))&&!((X,Y) IN circle(24900,24300,600))` This will include data within a circle of radius $60''$ but not within a concentric circle of $30''$. Finally, repeat step 3) except set the `filteredset` parameter to a different file name, e.g., `back.pi`.

5) To extract the source light curve (for use in § 5.5), 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. Next click the "OGIP Rate curve" button. Select the "Lightcurve" page 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 100 for a reasonable binning for a source which is not exceptionally bright. (NOTE: set `timebinsize=1` for the example in § 5.5.)

5.3.2 Create RMFs and ARFs

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

- Create the photon redistribution matrix, the RMF, using the task *rmfgen*.
  - rmfgen format=fixed rmfset=source.rmf spectrumset=source.pi threshold=1.0e-5 pnline=190

> `format` – "fixed" for now
> `rmfset` – output rmf file name
> `spectrumset` – input spectrum file name
> `threshold` – matrix values below this level will be set to 0
> `pnline` – **PN only**, source position along the CCD, `pnline = 199 - ABS(DETY + 1000)/80`

- Create the ancillary region file, the ARF, using the task *arfgen*.
  - arfgen arfset=source.arf spectrumset=source.pi rmfset=source.rmf extendedsource=no modelee=yes withpatterns=yes patternmax=12 patternmin=0 withrmfset=yes badpixlocation=mos1-filt.fits

    > `arfset` – output rmf file name
    > `spectrumset` – input spectrum file name
    > `rmfset` – rmf file created by *rmfgen*
    > `extendedsource` – extended source flag
    > `modelee` – flag to model the encircled energy
    > `withpatterns` – flag to specify event patterns
    > `patternmax` – maximum event pattern (e.g., 12 for MOS data and 1 for PN data), the range must be consistent with that set for the filtering
    Note: for PN data, patterns from 0-4 include all single and double events, however, currently patternmin=0, patternmax=1 for PN data will model all single and double events (patternmax=4 will not work)
    > `patternmin` – minimum event pattern
    > `withrmfset` – flag to use the rmf
    > `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.

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

5.3.4 Fit the Spectra

- Next use Xspec to fit the spectrum.

  > `Xspec`

```
XSPEC> data source-grp.pi        ! input data
XSPEC> ignore 0.0-0.3,8.0-20.0   ! ignore unusable energy ranges, in keV
                                 ! set a range appropriate for the data
XSPEC> model pow wabs            ! set spectral model to absorbed power law
powerlaw:PhoIndex> 2.0           ! set model power law index to -2.0
```

```
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 delchi       ! plot two panels with the log of the data and
                               !     the delta chi^2 values along the Y axes
XSPEC> exit                    ! exit Xspec
Do you really want to exit? (y) y
```

5.4 SOURCE DETECTION

The list of commands in this section follows the *edetect_chain* task and will apply the source detection tasks to data from the MOS1 detector in two bands. It uses a filtered data file produced as in § 5.2.4 with the assumption that it is 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.

   – `atthkgen atthkset=atthk.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*. It will use the filtered event list `mos1-filt.fits` produced above. *evselect* is run two times to create the images in two bands (300 - 2000 eV, and 2000 - 10000 eV).

   – `evselect table=mos1-filt.fits withimageset=yes imageset=image1.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(PI in [300:2000])' withimagedatatype=yes imagedatatype=Int32`

     > `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
     > `withimagedatatype` – flag to set image data type
     > `imagedatatype` – set image data type

   – `evselect table=mos1-filt.fits withimageset=yes imageset=image2.fits imagebinning=binSize xcolumn=X ximagebinsize=50 ycolumn=Y yimagebinsize=50 filtertype=expression expression='(PI in [2000:10000])' withimagedatatype=yes imagedatatype=Int32`

     > `expression` – filtering expression, select events in the PI channel range 2000-10000 eV
     > `imageset` – fits image name to be created, `image2.fits` for band 2

3) Create an exposure map for each energy band using the *eexpmap* task. It will also need to be run two times.

   – `eexpmap attitudeset=atthk.fits eventset=mos1-filt.fits expimageset=expimage1.fits imageset=image1.fits pimax=2000 pimin=300`

> `attitudeset` – attitude table
> `eventset` – event table
> `expimageset` – output file name, `expimage1.fits` for band 1
> `imageset` – input event image for band 1 (provides binning, etc., information)
> `pimax` – maximum PI channel for energy range, 2000 ev for band 1
> `pimin` – minimum PI channel for energy range, 300 ev for band 1

– `eexpmap attitudeset=atthk.fits eventset=mos1-filt.fits expimageset=expimage2.fits imageset=image2.fits pimax=10000 pimin=2000`

> `expimageset` – output file name, `expimage2.fits` for band 2
> `imageset` – input event image for band 2
> `pimax` – maximum PI channel for energy range, 10000 ev for band 2
> `pimin` – minimum PI channel for energy range, 2000 ev for band 2

4) Create a detector map using the task *emask*. Only one mask is necessary for all bands of a single detector.

– `emask detmaskset=detmask.fits expimageset=expimage1.fits threshold1=0.05 threshold2=0.5`

> `detmaskset` – output file name
> `expimageset` – input exposure image
> `threshold1` – fraction of maximum exposure
> `threshold2` – threshold for gradient of exposure

5) Do a sliding box source detection with using the task *eboxdetect* in local mode.

– `eboxdetect boxlistset=boxlist-l.fits boxsize=5 detmasksets=detmask.fits withdetmask=yes withexpimage=yes expimagesets='expimage1.fits expimage2.fits' usemap=no likemin=10 imagesets='image1.fits image2.fits' nruns=4 ecf='1.19 0.29' pimin='300 2000' pimax='2000 10000'`

> `boxlistset` – output file name for detected source list
> `boxsize` – size of the detection box
> `detmasksets` – names of the detector masks
> `withdetmask` – flag for using detector mask
> `withexpimage` – flag for using exposure map
> `expimagesets` – names of the exposure maps
> `usemap` – set local (no) or map mode
> `likemin` – minimum detection likelihood
> `imagesets` – names of the event images
> `nruns` – number of detection runs, the detection box is doubled for each successive run
> `ecf` – energy conversion factors for the energy bands, in units of 10^{-11} counts cm^2 erg^{-1} (MOS thin filter with an E^{-2} power law with 5×10^{20} HI cm^{-2} absorption)
> `pimin` – list of minimum PI channels for the bands
> `pimax` – list of maximum PI channels for the bands

6) Create a source excluded background map using the task *esplinemap* and the source list from Step 5).

– `esplinemap bkgimageset=bkgimage1.fits boxlistset=boxlist-l.fits withdetmask=yes detmaskset=detmask.fits withexpimage=yes expimageset=expimage1.fits imageset=image1.fits withcheese=yes`

> `bkgimageset` – output background map file name for band 1
> `boxlistset` – input list of detected sources
> `withdetmask` – flag to use a detector map
> `detmaskset` – input detector mask file name for band 1
> `withexpimage` – flag to use an exposure map

> `expimageset` – input exposure map file name for band 1
> `imageset` – input event image file name for band 1
> `withcheese` – flag to create a “Swiss Cheese” event image

– `esplinemap bkgimageset=bkgimage2.fits boxlistset=boxlist-l.fits withdetmask=yes detmaskset=detmask.fits withexpimage=yes expimageset=expimage2.fits imageset=image2.fits withcheese=yes`

> `bkgimageset` – output background map file name for band 2
> `detmaskset` – input detector mask file name for band 2
> `expimageset` – input exposure map file name for band 2
> `imageset` – input event image file name for band 2

7) Run the sliding box detection task *eboxdetect* a second time using background map mode.

– `eboxdetect bkgimagesets='bkgimage1.fits bkgimage2.fits' boxlistset=boxlist-m.fits imagesets='image1.fits image2.fits' withdetmask=yes detmasksets=detmask.fits withexpimage=yes expimagesets='expimage1.fits expimage2.fits' boxsize=5 likemin=10 nruns=4 usemap=yes ecf='1.19 0.29' pimin='300 2000' pimax='2000 10000'`

> Same as for the first run of eboxdetect except for:
> `bkgimagesets` – names of the background images
> `usemap` – `yes` to run in map mode

8) Run a maximum likelihood detection method using the task *emldetect*.

– `emldetect bkgimagesets='bkgimage1.fits bkgimage2.fits' mllistset=emllist.fits boxlistset=boxlist-m.fits:SRCLIST expimagesets='expimage1.fits expimage2.fits' imagesets='image1.fits image2.fits' mlmin=12 withexpimage=yes ecf='1.19 0.29' pimin='300 2000' pimax='2000 10000'`

> `bkgimagesets` – names of the input background images
> `boxlistset` – input file name for map-mode box source detected source list
> `expimagesets` – names of the exposure maps
> `imagesets` – names of the event images
> `mllistset` – output file name for source list
> `mlmin` – minimum detection likelihood
> `withexpimage` – flag for using exposure map
> `ecf` – energy conversion factors for the energy bands
> `pimin` – list of minimum PI channels for the bands
> `pimax` – list of maximum PI channels for the bands

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

– `srcdisplay boxlistset=boxlist-m.fits imageset=image2.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

10) Display the results of *emldetect* using the task *implot*.

– `implot set=image1.fits device=/GIF plotfile=image.gif withsrclistset=yes srclistset=emllist.fits colour=7 grid=2 itf=2 maxsrc=30`

> `set` – input image for the plot

> `device` – type of output (/GIF, /PS, /XW)
> `plotfile` – output file name
> `withsrclistset` – flag to include sources
> `srclistset` – source list file name
> `color` – color lookup table for image
> `grid` – control for coordinate grid overlay
> `itf` – image transfer function
> `maxsrc` – maximum number of sources to display

5.5 TIMING ANALYSIS

5.5.1 Extract Source Light Curve

Refer to § 5.3.1 for details of how to extract a light curve for a specific source, but set the `timebinsize` parameter on the *evselect* GUI "Lightcurve" page to 1.

5.5.2 Basic 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.rate`, has been created as in § 5.3 with `timebinsize` set to 1.

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.rate window=- dtnb=100 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

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.rate window=- dtnb=6.0 nbint=8192 nintfm=INDEF rebin=0 plot=yes plotdev=/xw outfile=power.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 in each power spectrum
     > rebin – rebin factor for power spectrum (0 for no rebinning)
     > plot – plot flag
     > plotdev – device for plotting
     > 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.rate 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.rate 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.rate 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 ODF DATA

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

- `mmmm_iiiiiijjkk_aabeeeccfff.zzz`

  `mmmm` – revolution orbit number
  `iiiiii` – proposal identifier number
  `jj` – observation ID - target number in proposal
  `kk` – observation ID - observation number for target
  `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.

5.6.1 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*.)

1) 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 (see Item 5) of § 3.3).

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.

5) Run the SAS task *emchain*. From the command line of a window where SAS has been initialized, simply enter:
   `emchain withbadpixfind=yes`
   Go get lunch as this task can take a while. *emchain* processes the data from both MOS instruments producing calibrated photon event files. If the data set has more than one exposure, then subsequent exposures can be accessed using the `exposure` parameter, e.g.:
   `emchain exposure=`*n* `withbadpixfind=yes`
   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 withbadpixfind=yes keepintermediate=none`
   Start working on the MOS data or go home for the day as this task can take even longer than *emchain*. The parameter `keepintermediate=none` causes *epchain* to discard a number of fairly large intermediate files.

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

5.7 A More-or-Less Complete Example

The Lockman Hole SV1 ODF data have been used for a reasonably complete example of the EPIC data reduction. The data can be found at:
`http://heasarc.gsfc.nasa.gov/docs/xmm/sv1.html`
while the script (`run.com`)and output data files can be found at:
`ftp://legacy.gsfc.nasa.gov/xmm/data/examples/epic/`
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. This holds for the CCF and ODF directories as well. Included in the distribution are the important output files produced by the script. The data were processed using SAS V5.2.

The entire process took a little over 12 hours on a Sun Solaris Ultra 2. The result is also about half 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) Initializes SAS. There are thirteen setup commands which have been discussed above, some of which will need to be modified for the specifics of one's home system.

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

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

8) Repeats items 4 - 6 for the PN data

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

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

11) Creates an RMF and ARF for the source

12) Included in the script, but commented out, are the commands to group and fit the source spectrum using grppha and Xspec

13) Creates a light curve for the source

14) 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 itemizes current SAS bugs and analysis issues. At the time of writing, there are important items concerning RGS pipeline products created with SAS v5.0, and a minor bug in SAS v5.2 that arises when using the optional parameter 'betabinning' in rgsproc.

6.1 A PRELIMINARY FIT

6.1.1 Pipeline Products

You will find a variety of RGS-specific files on the CD delivered by the XMM-Newton SOC. Generally there are two of each because there are two RGS cameras. 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:

- `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 data file names are of the form (note that we will use the generic form of the name in the following examples):

- `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

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*

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

- `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 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 GOs 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 v5.3) and calibration.

6.1.4 Fit Preparation

Use the FTOOL grppha so that:

1) XSPEC will automatically load the correct response matrix for the data.
2) The energy channels will be grouped to a minimum of 20 counts per bin. This ensures that the χ^2 minimization method will fit the data accurately. If the counts per channel are too low, the χ^2 method will systematically under-fit the data.

```
grppha
    Please enter PHA filename[] PiiiiiijjkkaablllSRSPEC1mmm.FTZ
    Please enter output filename[] !PiiiiiijjkkaablllSRSPEC1mmm.FTZ
    GRPPHA[] chkey RESPFILE RGS1_ORDER1.RSP
    GRPPHA[] group min 20
    GRPPHA[] exit
```

The “!” before the output filename tells the task to overwrite the previous file.

6.1.5 Fitting a Spectral Model to the Data

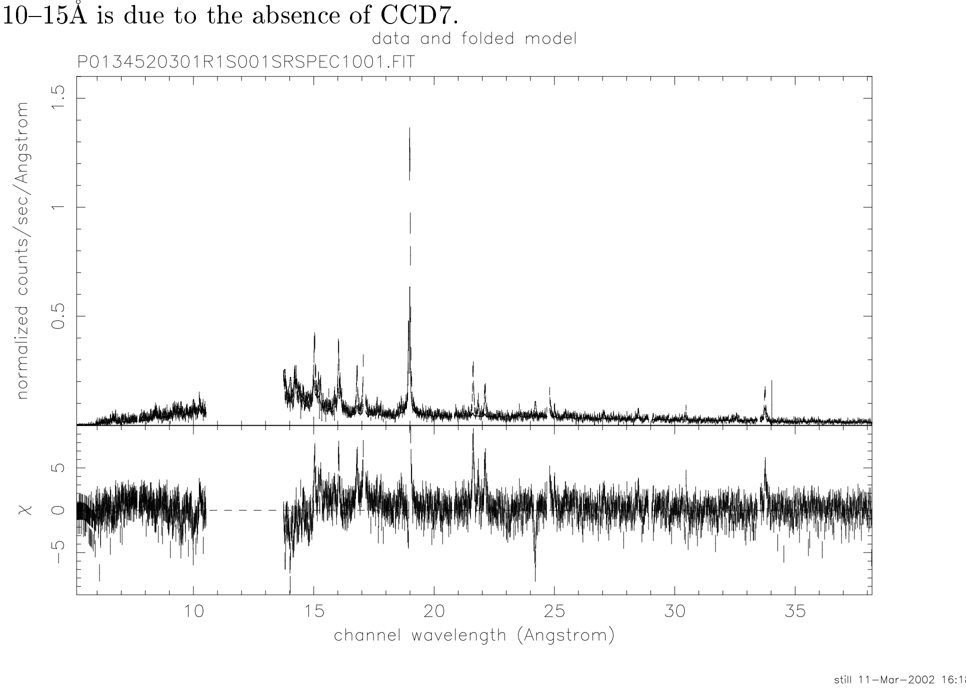
Now use XSPEC to fit an appropriate model to your spectrum:

```
xspec
    XSPEC>data PiiiiiijjkkaablllSRSPEC1mmm.FTZ
    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.

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.



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 nuber 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.

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 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 your 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 we want to avoid confusing solar flares for source variability, therefore we will use a region filter that that removes the source from the final event list. The region filters are kept in the source file product `Piiiiiijjkkaablll SRCLI_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))'`

  > `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.

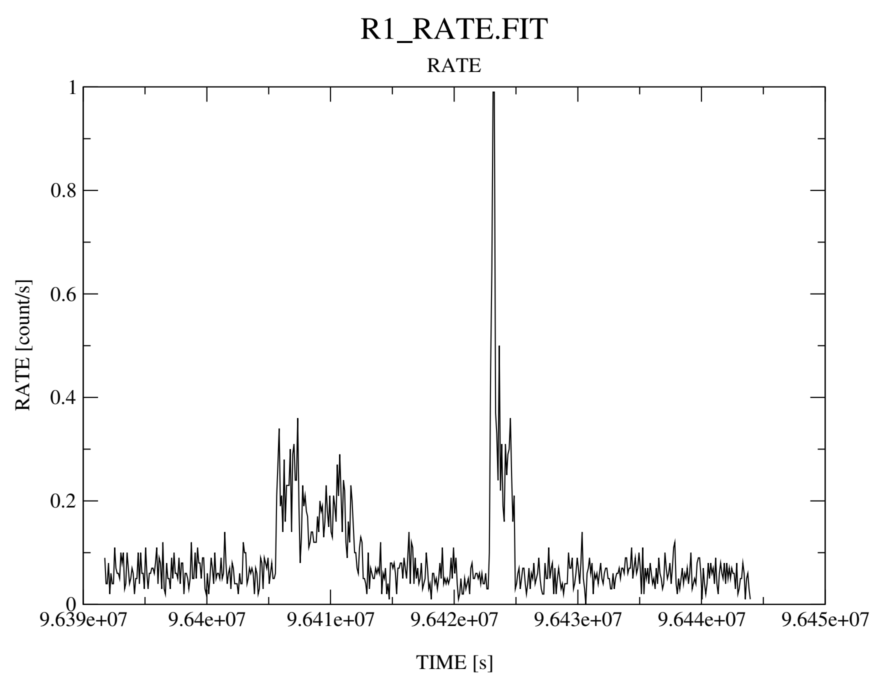
  > `timebinsize` – bin the time column to this size (in units of seconds).

  > `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 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 per second. In this example we have two intervals of significant background flaring. Determine which intervals you would like to reject and write these time intervals to an ASCII file, `gti.asc`, as follows:

```
9.6405e7 9.6413e7 –
9.6422e7 9.6425e7 –
```

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.

The first two columns provide the start and stop times (in seconds since the start of the mission) of intervals you would like to filter. The third column can be a '+' (good time interval) or '−' (bad time interval). In this case we will be excluding two intervals of high background activity.

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 we can create a GTI table 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<0.1)'`
  - > `table` – Input data file.
  - > `gtiset` – Output GTI table.
  - > `expression` – Boolean expression.

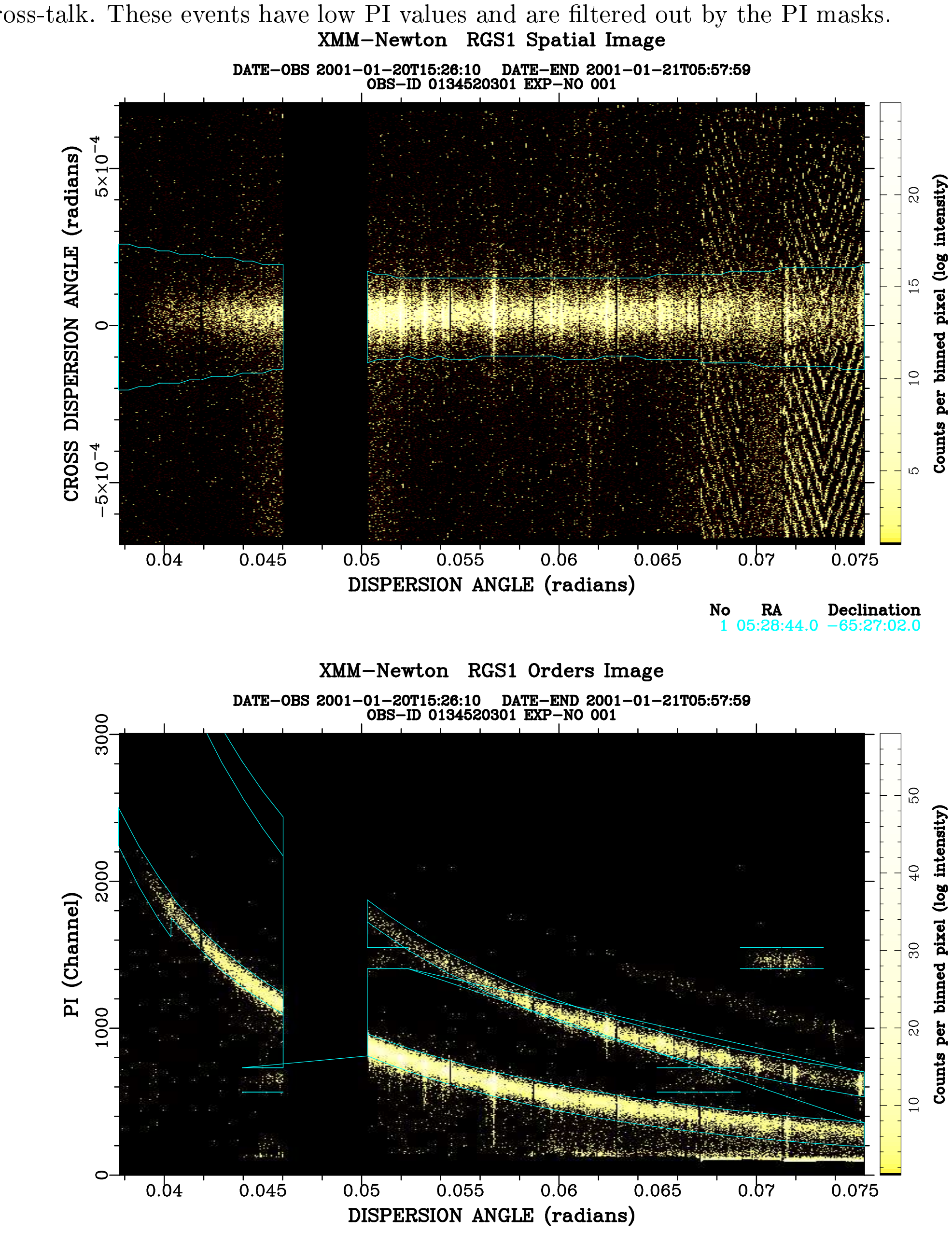
6.2.3 Running the RGS Pipeline

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

- `rgsproc orders='1 2' auxgtitables=GTI.FIT withmlambdacolumn=yes`
  - > `orders` – the spectral orders to extract.
  - > `auxgtitables` – a list of GTI files
  - > `withmlambdacolumn` – include a wavelength column in the event file product (we will use this to generate a "dirty" spectrum plot later).

New files will be 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 your observation, it is recommended you re-run the pipeline in order to take advantage of the most up-to-date calibration and ensure that region filters more appropriate for your source are created.

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.



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:

- 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.

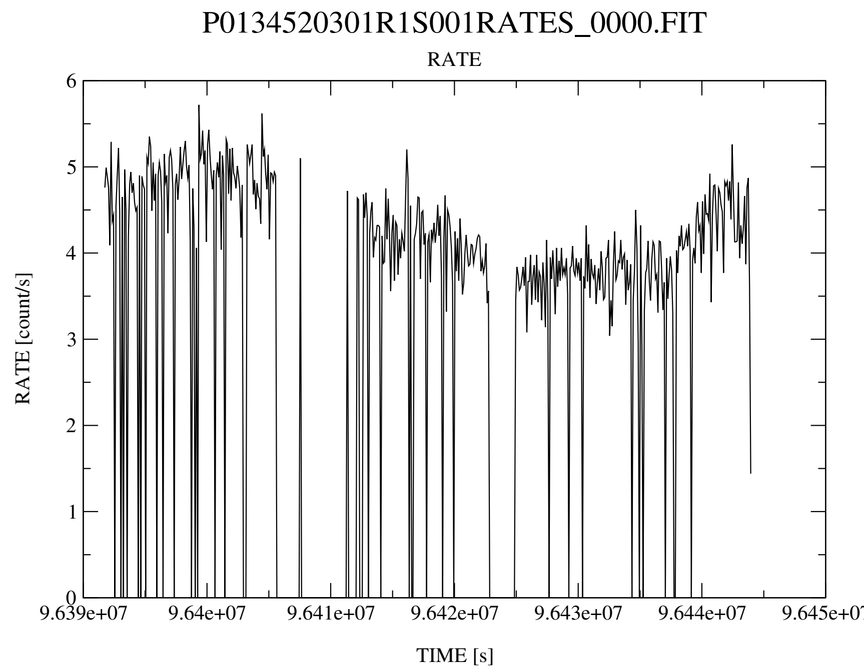
  > `maketimecolumn` – include an absolute time column.

  > `makeratecolumn` – create a rate column.

  > `timebinsize` – bin size (in seconds).

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

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



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 we can use the mlambdacolumn table column rather than a response matrix. One important caveat here is that all orders are superimposed on this table:

- `evselect table=PiiiiiijjkkaablllEVENLInmmm.FIT:EVENTS: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.

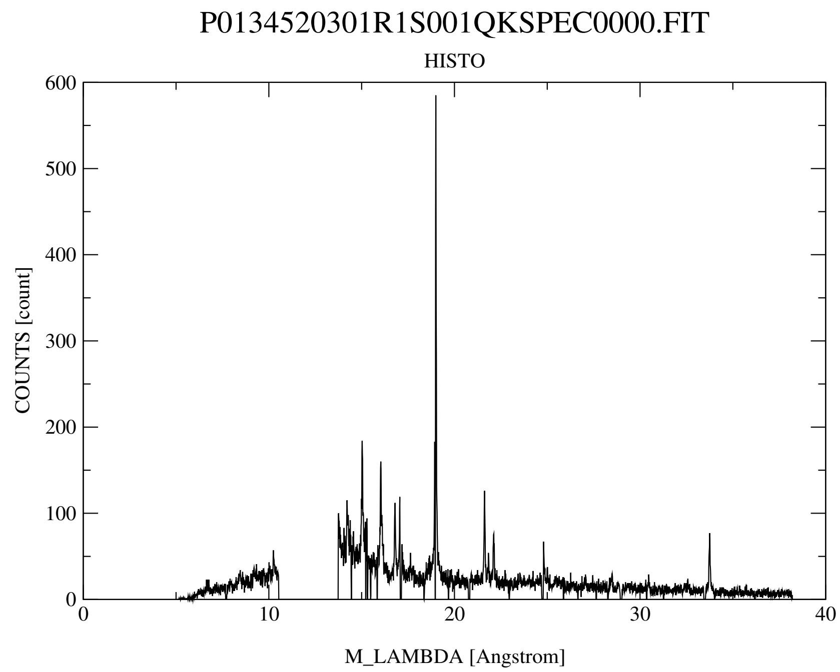
- `dsplot table=PiiiiiijjkkaablllHISTOGnmmm.FIT x=M_LAMBDA y=COUNTS`

The resulting histogram is provided in Figure 6.5.

6.3 PIPELINE EXAMPLES

Below we provide several examples of the flexibility of the RGS pipeline 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 your extraction regions, 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 your observation. This test, and the offset correction is not performed on your data before you receive it. To check for stray light and apply the appropriate offsets use:

- `rgsproc orders='1 2' 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, we have 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 it's coordinates have been taken from the EPIC source list which is included among the pipeline products. We have identified the bright neighboring object to be the third source listed in the sources file. The first source is the target:

- `rgsproc orders='1 2' 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 would save time if we didn't re-make the event table or re-calculate the exposure map. We can enter the pipeline at five different points. In this case we only need start from the spectral extraction stage:

- `rgsproc orders='1 2' entrystage=spectra finalstage=fluxing 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 you have retained your 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' 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 will be written to the RGS source list PiiiiiijjkkaablllEMSRLInmmm.FTZ 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 your 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 you wish to ignore during background extraction, then the majority of pipeline processing (drift correction, filtering etc) will remain identical to the previous examples. To save processing time we 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 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.

6.3.4 An Extended Source

The optics of the RGS allow spectroscopy of reasonably extended sources up to 1 arcmin. The location of your spatial extraction mask is defined by the detector coordinates of the source. The width of this mask is defined by the fraction of total events you wish to extract. This last quantity will be a trade off between improving your final signal-to-noise by maximizing the number of source counts with reducing background shot noise by minimizing the source extraction region. The further out the spatial point spread function of the source is sampled, the more background noise will dominate over source signal. The fraction of source counts within the default pipeline mask is 90%, however this is determined assuming the source is point-like.
Optimizing the mask width for a spatially-extended source may take some trial and error and, depending on the ionization distribution in your source, may depend on which lines you are 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 background mask is reduced so that the two do not overlap:

- `rgsproc orders='1 2' entrystage=spectra finalstage=fluxing xpsfincl=99 xpsfexcl=99 pdistincl=97`

  > `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.
Note that there are currently no SAS tasks to calculate correctly the response function for an extended source. The GOF are currently preparing a tool developed by the RGS calibration team for public delivery.
For very bright point sources the user may find that an unacceptable fraction of the source wings are being filtered out. In this case, the same procedure as above can be used to include more of the source.

6.4 PIPELINE ENTRY STAGES

There are five stages at which the user can enter 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 your 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 units 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 e.g., different exposure times, background subtractions and error propagation. 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.FTZ
> Please enter output filename[] !PiiiiiijjkkaablllSRSPEC1mmm.FTZ
```

> `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 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`

  > `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 you 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`

  > `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 you are 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 you have an up-to-date installation at your site.
One caveat of using the cstat option is that the scheme requires a “total” and “background” spectrum to be loaded into XSPEC. This is in order to calculate parameter errors correctly. Consequently, be sure not to use the “net” spectra that were created as part of product packages by SAS v5.2 or earlier. To change schemes in XSPEC before fitting your data, type:

- `XSPEC>statistic cstat`

6.7 A MORE-OR-LESS COMPLETE EXAMPLE

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

- `http://heasarc.gsfc.nasa.gov/docs/xmm/pv/public_list.html`

The script (`RGS_ABC.SC`) and output data files can be found at:

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

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. This holds for the CCF and ODF directories as well. Included in the distribution are the important output files produced by the script. The data were processed using SAS V5.3 and the entire process took on the order of 1 hour running on a linux box with an 800mHz Pentium III processor. 266 MB of processed data was created. 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 a bit different from the other instruments on board XMM-Newton and not only because it's not an X-ray instrument. The OM pipeline products can be used directly for science analysis and one does not necessarily need to reprocess the data. So in **principle** you could ignore the files in the ODF directory and go directly to the next section which explain what the files in the PIPEPROD directory are.

You will find a variety of OM-specific files provided on the CD delivered by the XMM-Newton SOC. The pipeline products differ slightly with different versions of the SAS software. We will give a brief description of what the files are in newest version of the SAS (in this text SAS 5.3.0) and we will mention the important difference with older pipeline products.
For a complete description of the files, you may want to check the Bible of pipeline products, a 131-page document found at:
`http://xmmssc-www.star.le.ac.uk/pubdocs/SSC-SP-004.ps.gz`
which contains the definite description of all pipeline products. It is a little outdated now, but a new version (valid for SAS 5.3.0) should be coming out soon (anytime now).

Users interested in either reprocessing their data or learning a bit more about the data processing can keep reading, the rest skip to the section on Pipeline products.

The OM can operate in three modes: IMAGING, FAST and GRISM. In general, the ODF names for the data will look something like:

- `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) observations.

    `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)

We describe below the IMAGING analysis chain called *omichain* step by step. We have also written an equivalent to this *omichain* which will allow you to vary the parameters input of each separate tasks if you want to do so or to run the pipeline on only a small part of the data.
In all what follows, we have used the public data on the Lockman Hole (Revolution number 70) to illustrate our text. If you want retrace step by step what follows, we suggest you to download the set on your machine.

7.1 IMAGING

7.1.1 The Stray-light Problem

All the OM images are affected by the so-called "stray-light" problem. This problem does **NOT** affect source detection and magnitude determination but contributes to a higher background (and an ugly picture). The problem exists but it is less noticeable at UV wavelengths. A program to produce clean images exists but the results are strictly for display purposes as the routine does conserve flux. The program is based on proprietary software. Because the problem is mainly aesthetic, there are no plan to develop a publicly available routine.

7.1.2 ODF PRODUCTS

In the `IMAGING` mode, a list of all the OM files under your ODF directory looks something 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 any 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.

```
jerusalem-24-Mydata: 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/tmp_demo/Mydata
//
```

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 and the third one (0070_0123700101_SCX00000ATS.FIT) contains the spacecraft attitude file.

`omichain` runs on filters specified by the users. If no argument are given, the chain runs on all the files present in the $SAS_ODF directory. If you were to rerun the `omichain` tasks one by one again, you may find some small differences between the files obtained this way and the pipeline products in your PIPEPROD directory. The main reasons for these differences are improvement made to the software and the type of products produced by the pipeline (for example, only the most recent products have a final image per filter)and some changes in the calibration products.

The following explains the "step-by-step" processing of your OM files. At the end of this section, we provide a script which goes automatically through all of the steps described below. The script is in fact a commented version of the `omichain` package to let you see what the processing does. We suggest you to go through all the steps at least once manually before going looking for the script.

Preparation for Processing the Data

If you want to group your ODF files by filter values, you'll have to extract the `FILTER` keyword from their headers – You can do that using FTOOLS task `fkeyprint` as in the following example:
`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 7.1.

Table 7.1: OM filter and file name correspondence.

File Name 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 file is available at the XMM-GOF site:
`ftp://legacy.gsfc.nasa.gov/xmm/software/om_tools/file_examine.shell`.

Running this script will give you immediately the list of files and their associated filter. It is then very easy to see which files are associated with each others.
The details of the association are less complicated that it may appear at first. In the standard configuration (the so-called Rudi-5 mode) one gets exposures in group 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 exposure covers the 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 the previous example, the high resolution window will be called `0070_0123700101_OMS00400IMI.FIT.gz` while the low resolution window will be `0070_0123700101_OMS00401IMI.FIT.gz`. The first three digits are paired such as in general the smallest 00x number goes with the 4 first smallest 4yy numbers.
For example, the file `0070_0123700101_OMS00400IMI.FIT.gz` is a high resolution file associated with the 4 files starting with `0070_0123700101_OMS41500IMI.FIT.gz`, `0070_0123700101_OMS41600IMI.FIT.gz`, `0070_0123700101_OMS41700IMI.FIT.gz` and `0070_0123700101_OMS41800IMI.FIT.gz`. If this is not clear, here is an example of what the script (file_examine.shell) will give:

```
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 you already know that the 00 and 01 numbers are the mark of high resolution and low resolution, one case immediately tell that the file 0070_0123700101_OMS00400IMI.FIT is associated with files 0070_0123700101_OMS41500IMI.FIT, 0070_0123700101_OMS41600IMI.FIT, 0070_0123700101_OMS41700IMI.FIT and 0070_0123700101_OMS41800IMI.FIT. Please be aware that one should **NOT** add low resolution and high resolution images together even when they cover the same part of the FOV (so one cannot add 0070_0123700101_OMS00401IMI.FIT and 0070_0123700101_OMS00400IMI.FIT for example).

Once you've decided which data to process (for example one exposure of one filter taken in a certain resolution), you should make sure that

1) You have created a Calibration Index File, using *cifbuild* (§ 4.5.1).
2) You have created a summary file of your ODF constituents and copy it in your ODF directory using the SAS task *odfingest* (§ 4.5.2).
3) Pick one set of exposure and run *omprep* on it.

As an example, we have chosen the first high resolution exposure for the Lockman Hole data. Our working directory looks like:

```
jerusalem-134: ls Mydata
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 with `cifbuild`.

Examine the Guide Star Record

- Both the ODF and the THX files should be processed by the *omprep* task before being able to run any other SAS task.
  - `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. You can examine the OM tracking history using the task *omdrifhist*. The output is a postscript file containing plots and statistics on tracking history. To run *omdrifhist* do:
  - `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)
- You can have a look at the output PS file. The other check to be done on the tracking is to look at the count rates of the guide stars. To do this, use the task *omthconv* to produce a FIT 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 § 7.1.2 except for input file which is not the IMI file.

Once the *omprep* task 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` – Sampling factor.
  - > `timefactor` – leave it to 1 (see note below)
  - > `set` – Corrected IMI file
  - > `thxset` – Corrected THX file

Note: For the moment, the bad pixel information **is not** used by the detection algorithm.

Note: The output file is a *modified* IMI file. You *can not* run this task twice as it fails if it detects an existing `QUALITY` extension. To prevent any problem, you may want to 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 details, it appears that the tracking is generally so good that sub-sampling does not seem necessary. This parameter should be left to one.

Flat Field Generation

There is and there will be no flat field generation in the OM pipeline and the task *omflatgen* produces a unit flatfield

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

  > `outset` – name of the output file.

  Note: `omflatgen` can produce the following warning:
  `** omflatgen:  warning (ArgumentIgnored), Warning:  countRateFrameRatio argument in small-scale flatfield retrieval currently ignored`
  This warning is generated from within a `CAL` routine called within *omflatgen*. As *omflatgen* only generate a FITS file with only 1s, you can safely ignore this warning.

Once you have your `OUT_FLATGEN.FIT` file, you 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 electronic (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 unit of 8 pixels

Perform Source Detection

The task *omdetect* employs a simple two-stage process to locate sources in an OSW image. The first stage is to determine the background. The second stage is a 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 fool by them. One way to separate them from real detection is to look at the FWHM max and min parameter in the source list. Spurious source detection associated with stray-light features will have large value associated to these parameters.

Convert Source Counts to Magnitudes

The task *ommag* converts the list of given counts to magnitudes in the appropriate instrumental band passes. The accuracy is estimated to a few 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, with source positions in pixels, and converts 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 catdir=''`

  > `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 catalog?

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

  > `catdir` – Path to the catalog directory

  Note: It was decided a while ago that because of its large size, the catalog would not be distributed but it is possible for people to provide their own catalog if they wish. The format is that used for the USNO cross-correlation FITS products. In general though, 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 you to run the pipeline only on file you want to. The script is available at:
`ftp://legacy.gsfc.nasa.gov/xmm/software/om_tools/omproc_gof`.
Please contact Ilana Harrus (imh@milkyway.gsfc.nasa.gov) if you have any problem with it.

7.1.3 PIPELINE PRODUCTS

When you receive your data, the OM data under the PIPEPROD directory will look like:

- `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 contents (Table 7.2)

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

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

New in this release:
The new pipeline produces a summed sky image for each of the filter in low resolution. The results of the sum is put in a file which looks like

- `PjjjjjjkkkkOMX000RSIMAGbb000.QQQ`

  `jjjjjj` – proposal number

  `kkkk` – observation ID

  `bb` – filter keyword- L is for UVW1 and S for UVW2

  `zzz` – file type (e.g., PNG, FTZ)

Table 7.2: 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 will be the low resolution final image in the **B** filter of the observation 56789 in sky coordinates (indicated by the S before the IMAG). The letter **L** is used for the UVW1 filter and **S** for UVW2. To know which of the files have been added to create the final image P0123456789OMX000RSIMAGB000.FTZ, one has to look at the keyword **XPROC0** in the FITS header. The keyword will tell you something like:

```
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 7.3: 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 your data product does not contain any mosaic file for all the exposures but about 10 files per filter, it means that it was processed with an older version of the pipeline. Because the newest version (SAS 5.3.0) of the pipeline is so much better, we recommend you to reprocess any older OM data that you have. Once `omichain` is finished, you will have to run the task `ommosaic` on the result to get only one single final file per filter. You have to specify which files to be added (the program does not do it independently) so you have to know which

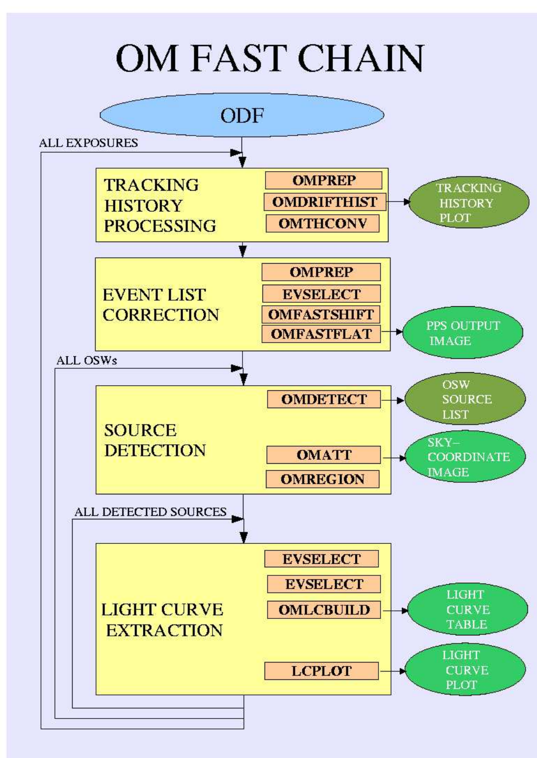
files are taken with which filters and at which resolution.
You can also use a software written at the XMM-Newton US GOF. The task is meant to be used on files in the PIPEPROD directory (which contains the outputs of the OM pipeline). It produces a final image and an exposure image in sky coordinates for each of the filter used in your observation. Low and high resolution images are treated separately It requires you to have FTOOLS and Perl installed on your machine and running the script from a writable directory in which you put the OM files. The tar file with the script is available at the XMM-GOF site:
`ftp://legacy.gsfc.nasa.gov/xmm/software/om_tools/om_prod_all.tar.gz`.
The program is fairly easy to use and modify. Please contact Ilana Harrus (imh@milkyway.gsfc.nasa.gov) if you have any problem with it.

7.2 FAST MODE

The new version of the `SAS` (version 5.3) has a working fast mode pipeline. If your data have not been processed by the newest version, you should run the task `omfchain`.

The chain works on a similar principle than the imaging chain and consists of a Perl script which calls all the necessary tasks one by one. It produces images of the sources found, extracts events related to the source and the background and extract the corresponding light curves. A more detailed description of the chain can be found in the SAS on-line help. The task is not yet included into the OM tasks but can be found in the list of all the tasks. 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 on Figure 7.1.

Figure 7.1: OM fast chain–diagram of the different tasks run.



7.3 GRISM ANALYSIS

At the time of this writing, there is no `SAS` task available to analyze the GRISM data. Please contact the XMM-Newton GOF help desk at `xmmhelp@athena.gsfc.nasa.gov` if you have GRISM data to analyze and don't know how to proceed.