

# COOKBOOK FOR ANALYSIS PROCEDURES FOR *XMM-NEWTON* EPIC OBSERVATIONS OF EXTENDED OBJECTS AND THE DIFFUSE BACKGROUND

Updated for SAS Version 21.0

Questions and Problems Should Be Sent to [xmmhelp@athena.gsfc.nasa.gov](mailto:xmmhelp@athena.gsfc.nasa.gov)

S. L. SNOWDEN

*Formerly:* Code 662, NASA/Goddard Space Flight Center, Greenbelt, MD 20771

*Currently:* Retired and Enjoying Life to its Fullest

AND

K. D. KUNTZ

*Currently:* Department of Physics & Astronomy, Johns Hopkins University, Baltimore, MD 21218

*Futurely:* Beginning to Think about Retirement

[kkuntz1@jhu.edu](mailto:kkuntz1@jhu.edu)

## ABSTRACT

The following manual/cookbook is for the use of the *XMM-Newton* Extended Source Analysis Software (XMM-ESAS, or just ESAS) suite of packages for the analysis of imaging mode EPIC MOS and pn observations. ESAS creates both model quiescent particle background spectra for use in spectral fitting and exposure-corrected, background-subtracted (particle, soft proton, and solar wind charge exchange) images. The spectra and images are produced for user-defined regions within an observation field of view. The output files are in standard FITS format. Software for mosaicking multiple observations of not necessarily co-aligned pointings is included in this package.

**This version of ESAS is significantly different from previous versions of ESAS.** The original ESAS relied very heavily on scripts, in part to manage the large number of files required to produce the background images and spectra. The goal was to allow the scripts to do everything that did not require direct human intervention. As a result, ESAS required a very rigid directory/file name structure. As part of an effort to implement ESAS more closely into SAS, the ESAS routines have been modified 1.) to be less dependent upon the rigid directory/file name structure, 2.) to make the routines more modular, 3.) reduce the number of files produced, and 4.) clean up a number of loose ends, unclear usage, and the like. As a result, the bulk of the scripting has been moved out of ESAS. This is unlikely to be a traumatic change; users have become more accustomed to scripting in a variety of languages.

A new method of constructing the quiescent particle background spectra should arrive with SAS V22.0.

We would like to thank members of the MOS and pn hardware and software teams, the *XMM-Newton* SOC at the European Space Astronomy Centre (ESAC), and other members of the EPIC Background Working Group for their contributions that ranged from helping us to understand instrument and software issues to the identification of filter-wheel closed observations in the archive. Carlos Gabriel and Aitor Ibarra of the SOC and Brendan Perry of the NASA GOF deserve special thanks for their efforts in converting XMM-ESAS routines into SAS tasks. Ignacio de la Calle Pérez also deserves our thanks for being the point person at the SOC for responding to users' questions about ESAS. We would also like to thank the many users that have provided comments that have been used to improve this software. Suggestions and comments are always welcome.

Users of this package should be aware of the informational web pages covering the background issues of EPIC observations at: <http://www.cosmos.esa.int/web/xmm-newton/background> and the EPIC Calibration Status document at: <http://www.cosmos.esa.int/web/xmm-newton/calibration>.

## Contents

<b>1 INTRODUCTION</b>	<b>4</b>	5.13 Creation of Model Particle Background Spectra and Images . . . . .	34
1.1 Overview . . . . .	4	5.13.1 Extraction OR mos/pnspectra . . . . .	35
1.2 Methodology . . . . .	6	5.13.2 Combination OR mos/pnback . . . . .	39
1.2.1 The Useable Modes . . . . .	6	5.13.3 Diagnostics . . . . .	39
1.2.2 The Useable Exposure Times . . . . .	6	5.13.4 Clobbering . . . . .	40
1.2.3 The Spacecraft Position . . . . .	6	5.13.5 Region Descriptors . . . . .	40
1.3 MOS Reduction <i>versus</i> pn Reduction . . . . .	7	5.13.6 Creating Identical Regions Across Observations: . . . . .	41
1.3.1 The Corner Problem . . . . .	7	5.13.7 Point Source Masking . . . . .	42
1.3.2 The Pattern Problem . . . . .	7	5.14 Preparing for Further Analysis . . . . .	42
1.4 More on Background Components . . . . .	7	<b>6 SPECTRAL FITS</b>	<b>45</b>
1.4.1 Quiescent Particle Background . . . . .	7	6.1 Quiescent Particle Background . . . . .	45
1.4.2 Residual Soft Proton Contamination . . . . .	7	6.2 Cosmic & Galactic Backgrounds . . . . .	46
1.4.3 Solar Wind Charge Exchange Contamination . . . . .	7	6.3 Residual Soft Proton Emission . . . . .	47
1.5 The Problem of Sparseness . . . . .	9	6.4 Solar Wind Charge Exchange Background . . . . .	48
<b>2 CHANGE LOG</b>	<b>10</b>	6.5 Outside Constraints . . . . .	48
2.1 Summary . . . . .	10	6.6 Models and Xspec . . . . .	49
2.2 Concordance . . . . .	14	6.6.1 Importing the Data . . . . .	49
2.3 Future . . . . .	18	6.6.2 Declaring Models . . . . .	49
<b>3 CAVEAT EMPTOR</b>	<b>19</b>	6.7 Xspec Fits of Multiple Regions: Clusters . . . . .	51
<b>4 ESAS FLOW CHART</b>	<b>20</b>	6.7.1 Constraining the Cosmic Background . . . . .	52
<b>5 XMM-ESAS COOKBOOK</b>	<b>21</b>	6.7.2 Crosstalk . . . . .	53
5.1 The XMM-ESAS Package . . . . .	21	<b>7 IMAGE PRODUCTION</b>	<b>55</b>
5.2 Calibration/CCF Files . . . . .	21	7.1 Particle Background . . . . .	55
5.3 XMM-ESAS Produced Files - Some Warnings . . . . .	21	7.2 Soft Proton Background . . . . .	55
5.4 How This Cookbook Works . . . . .	21	7.2.1 Transfer from One Region to Another . . . . .	56
5.5 SAS Reminders . . . . .	22	7.3 Solar Wind Charge Exchange Background . . . . .	56
5.6 Setup . . . . .	22	7.4 Stray Light . . . . .	57
5.7 Initial ESAS Processing . . . . .	23	7.5 Masking . . . . .	58
5.7.1 The Chains . . . . .	23	7.6 Image Creation . . . . .	59
5.7.2 The Procs . . . . .	23	7.7 Image Analysis . . . . .	59
5.8 Examining Some Diagnostics . . . . .	23	7.8 Image Massaging . . . . .	60
5.9 Examining CCDs for Anomalous States . . . . .	24	<b>8 MOSAICS</b>	<b>62</b>
5.10 Soft Proton Flare Filtering . . . . .	25	8.1 The Overall Mechanics . . . . .	62
5.11 Merging . . . . .	29	8.2 Source Removal Considerations . . . . .	64
5.12 Detecting and Excising of Point Sources . . . . .	29	<b>9 MOSAIC MODE OBSERVATIONS</b>	<b>66</b>
5.12.1 Missed Sources . . . . .	31	9.1 Soft Proton Filtering . . . . .	66
5.12.2 External Catalogues . . . . .	32	<b>10 OTHER USEFUL THINGS</b>	<b>67</b>
5.12.3 Rolling Your Own . . . . .	32	10.1 PPS File Names . . . . .	67
5.12.4 A Middle Way . . . . .	33	10.2 merge . . . . .	67
5.12.5 And if the Source is Really Big... . . . .	34		

10.3	Maximal Identical Regions for All Detectors Simultaneously . . . . .	67	12	A1795 Cheese Mask Image . . . . .	32
<b>11</b>	<b>REFERENCES</b>	<b>69</b>	13	A1795 MOS1 (Source and Background) and FWC Spectra . . . . .	35
<b>A</b>	<b>TASK DESCRIPTIONS</b>	<b>70</b>	14	A1795 PN (Source and Background) and FWC Spectra . . . . .	36
A.1	BINADAPT . . . . .	72	15	<b>A1795 Count and Exposure Images</b> . . . . .	37
A.2	BINADAPTMERGE . . . . .	74	16	Corner Data Augmentation Plots . . . . .	40
A.3	CHEESE . . . . .	76	17	Fitted A1795 Spectra . . . . .	43
A.4	COMBIMAGE . . . . .	78	18	A1795 Spectral Model Components . . . . .	44
A.5	CONVREGION . . . . .	80	19	Spectral Fits to QPB Spectra . . . . .	47
A.6	EMANOM . . . . .	81	20	A1795 Radial Profile Regions . . . . .	51
A.7	EMLFILL . . . . .	82	21	Fitted MOS and PN A1795 Radial Profile Spectra . . . . .	52
A.8	EMLMERGE . . . . .	83	22	A1795 Fitted Temperature Radial Profile	53
A.9	ESPFILT . . . . .	84	23	A2204 Fitted Temperature Radial Profile	53
A.10	MOSAICMERGE . . . . .	86	24	Comparison of A1795 Radial Profiles . . . . .	54
A.11	MOSBACK . . . . .	88	25	Cross-Talk ARF Comparison . . . . .	54
A.12	MOSSPECTRA . . . . .	91	26	Particle Background Images . . . . .	55
A.13	PNBACK . . . . .	94	27	Soft Proton Background Images . . . . .	57
A.14	PNSPECTRA . . . . .	97	28	SWCX Background Images . . . . .	58
A.15	POINTSRC . . . . .	100	29	Stray Light Image . . . . .	58
A.16	PROTON . . . . .	101	30	LMC Stray Light Image . . . . .	59
A.17	PROTONSCALE . . . . .	102	31	Smoothed A1795 Image . . . . .	60
A.18	ROTDET2SKY . . . . .	103	32	A1795 Smoothing FWHM Image . . . . .	61
A.19	SPPARTIAL . . . . .	104	33	M101 Observation Light Curve Filtering	62
A.20	SWCX . . . . .	105	34	Hard and Soft M101 Mosaic Images . . . . .	63
<b>B</b>	<b>EXAMPLES</b>	<b>106</b>	35	M101 Merged Exposure Map . . . . .	64
B.1	A Simple Single Observation, Spectroscopy and Imaging . . . . .	107	36	Coma Cluster Mosaic . . . . .	65
B.2	A Complex Single Observation, Spectroscopy and Imaging . . . . .	122	37	Mosaic Mode Pointing Demonstration . . . . .	66
B.3	Multiple Overlapping Obsids, Imaging . . . . .	123	38	The plots produced by <code>mosback</code> . . . . .	90
B.4	Cheese with Multiple Overlapping Observations . . . . .	125	39	The plots produced by <code>pnback</code> . . . . .	96
<b>List of Figures</b>			<b>List of Tables</b>		
1	Introduction to the FOV . . . . .	5	1	Change Log . . . . .	11
2	A1795 Temperature Profile Comparison . . . . .	6	2	Routine Concordance between Old ESAS and Current ESAS . . . . .	14
3	PN QPB Patterning . . . . .	8	3	File Name Concordance . . . . .	15
4	SWCX Contamination Spectrum . . . . .	8	4	Future Modifications . . . . .	18
5	ESAS Flow Chart . . . . .	20	5	Event File Names . . . . .	24
6	Anomalous CCD Spectra . . . . .	25	6	Anomalous Criteria . . . . .	25
7	Event Images Including Anomalous State . . . . .	26	7	Instrumental Lines. . . . .	45
8	A1795 Temporal Filtering . . . . .	27	8	Common SWCX Emission Lines. . . . .	56
9	A1795 Temporal Filtering . . . . .	28			
10	Temporal Filtering of Heavily Contaminated Light Curve . . . . .	29			
11	<i>Cheese</i> intermediate files . . . . .	31			

## 1. INTRODUCTION

The *XMM-Newton* (Ehle et al. 2005) Science Analysis System (SAS) Extended Source Analysis Software (XMM-ESAS) package described here follows the methods outlined in Snowden et al. (2008, based on the background calibrations of Kuntz & Snowden 2008) for the analysis European Photon Imaging Camera (EPIC) MOS (Turner et al. 2001) observations of extended objects and the diffuse X-ray background. XMM-ESAS was subsequently extended to include processing of pn (Strüder et al. 2001) observations. The analysis of pn data is similar to that of MOS data but has a number of added complications: needing to account for out-of-time (OOT<sup>1</sup>) events, soft proton flare contamination of the corner data, and some as yet unclear interaction between those two issues.

XMM-ESAS began life as a collection of routines to allow the spatial and spectral analysis of diffuse emission filling the field of view. Some of the routines provided capabilities missing from the original SAS, such as creating the quiescent particle background spectrum for a given observation, or creating an image of the soft proton contamination. The original XMM-ESAS package consisted of a group of IDL routines (developed by Kuntz) and a number of perl scripts used to call both SAS tasks and the IDL routines in the correct order. In the second version (the initial public release), the IDL routines were replaced with FORTRAN, by Snowden, for portability (and to avoid the need for IDL licenses), and the perl scripts were cleaned, updated, and extended. Other routines were intended to provide tried and tested out-of-the-box scripts for repetitive or complex tasks, such as point source removal. As a result, XMM-ESAS had become a mixed collection of FORTRAN or C routines for new functions, and PERL routines for scripts. The third version of the package saw the FORTRAN routines and perl scripts incorporated into SAS. The fourth version (SAS V13.0) saw the conversion of the FORTRAN routines from FORTRAN 77 to FORTRAN 90/95. With the release of SAS V21.0, all the FORTRAN code is now integrated into SAS, PERL routines have been re-implemented in SAS-format FORTRAN, and most (but not all) calibration files have been incorporated into the SAS CCF. As will become apparent below, XMM-ESAS is still a work in progress. Please bear with us!

<sup>1</sup>OOT events occur when a pn data frame is being clocked out. This takes a small but finite amount of time during which new events can also be recorded by the detector, however, they will not be associated with the correct sky location.

### 1.1. Overview

XMM-ESAS is intended to produce both images and spectra of the background for regions that cover a significant fraction of the FOV<sup>3</sup>. The production of *images* depends critically on the production of *spectra*, and the order of analysis can sometimes become confusing. This overview is intended to blunt some of that confusion.

*Imaging:* The purpose of XMM-ESAS, for imaging, is to produce images of all of the known backgrounds and foregrounds so that one can isolate the true cosmic X-ray emission. The background/foreground components that XMM-ESAS handles are:

- the quiescent particle background (QPB)
- soft proton flares (SPF) and, to some extent
- the solar wind charge exchange (SWCX).

In each case one must know that distribution of that emission across the FOV and the absolute strength of that emission.

For the QPB, the distribution across the FOV is given by the filter-wheel closed (FWC) data, while the normalization is provided by the “corner” data: the parts of the detectors that are shielded from the X-rays focussed by the mirrors<sup>4</sup> as can be seen in Figure 1.

For the SPF, the distribution across the FOV was laboriously determined by Kuntz & Snowden (2008)<sup>5</sup> and the normalization must be obtained by fitting the *spectrum* from the entire FOV. Thus, imaging relies heavily on spectroscopic analysis. Although the bulk of the SPF emission can be removed by excising the time periods containing the flares, there is nearly always some residual emission from the remainder of the observation. The amount of that residual can be determined from the *spectrum* of the entire FOV as the SPF spectrum is (usually) distinctly different from that of any other likely emission component.

For the SWCX, the distribution across the FOV is given by the standard exposure map and the normalization must be obtained by fitting the *spectrum* of the

<sup>3</sup>That is, XMM-ESAS was built to handle observations of objects that are sufficiently large that there is essentially no “background region”. XMM-ESAS can be used for smaller regions, but then one should carefully check the statistics.

<sup>4</sup>The “corners” are essentially regions behind a field stop, an aluminum mask. Thus, although this region is protected from X-rays being focussed by the mirrors, it is not protected from X-rays and secondary electrons created in the space around the detector. However, the field stop would appear not to be close to the surface of the pn, and there appears to be a contribution from the FOV to the corners.

<sup>5</sup>This determination needs to be updated, but is very time consuming, and thus is a low priority for ESAS development.

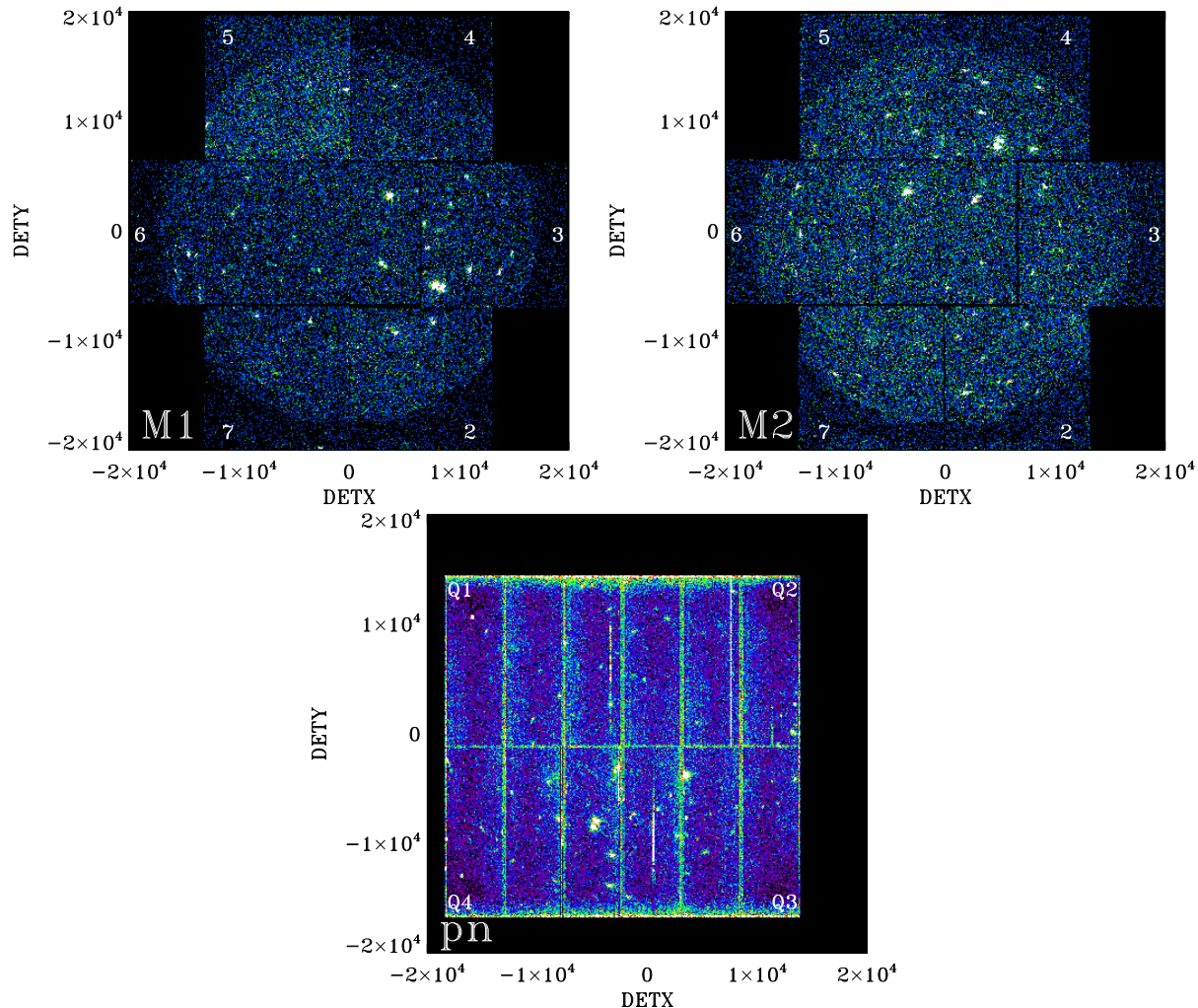


Fig. 1.— The full field of view of each of the three instruments, all in the detector coordinate frame<sup>2</sup>. For MOS1 (upper left panel) and MOS2 (upper right panel) the individual CCDs (except for the center one, which is CCD 1) are labeled. For the pn (lower panel) the quadrants used by ESAS are labeled. This observation was chosen because it shows moderate soft proton flare emission within the FOV, which highlight the corner regions. For both MOS1 and MOS2, one can see scattered light within the bottom corner regions. For both MOS1 and MOS2, CCD 5 is in an anomalous state, though not strongly so. Note the relative difficulty of seeing the corner data for the pn.

entire FOV. Thus, one can see once again the need to do spectroscopic analysis in order to obtain a background/foreground subtracted image.

*Spectroscopy:* The purpose of XMM-ESAS, for spectroscopy, is to produce the spectrum of the quiescent particle background for the observation. All other spectral components must be characterized by fitting. In order to get the best spectrum of the diffuse emission, however, one will want to remove the point sources, which will require at least some imaging.

*Order of Operations:* Whether one is interested in imaging or spectroscopy, the initial steps are the same. We assume the existence of freshly recalibrated event files. We first determine which chips (if any) are in anomalous states and need to be removed. We then

determine which time segments should be removed due to soft proton flares. We then identify point sources, build region files describing the location of the point sources, and build masks to exclude those regions. Those masks are required both to exclude the sources in making images, as well as to exclude them from the spectrum of the diffuse emission.

The next step can be rather time consuming and requires some skill. We extract the spectrum from the region of interest excluding, of course, the point sources. We build a QPB spectrum. Then, if we are only interested in the spectrum, the analysis may begin.

If we are interested in imaging, we fit the spectrum from the FOV (or the region of interest) in order to de-

termine the strength of the remaining soft proton flare contamination; that fitting need not be done precisely, nor in a completely astrophysically motivated manner. Similarly, the spectral fitting may be required in order to determine the amount of solar wind charge-exchange (SWCX) emission. Then, we build the QPB image, the soft proton image, and the SWCX image so that we may subtract them from the count image and divide by the exposure image to get a properly calibrated image of the flux from the source of interest.

## 1.2. Methodology

Our methodology for producing model QPB spectra (Kuntz & Snowden 2008) and other background components is based as much as possible on “first principles”. We attempt to model as many of the aspects as possible using a wide range of inputs, e.g., FWC data, data from the unexposed corners of archived observations, and *ROSAT* All-Sky Survey data.

We avoid the use of blank-sky data for a number of reasons. First, the blank-sky data include unknown levels of residual SP contamination and SWCX contamination. Second, the cosmic X-ray background varies strongly across the sky, so what might be a good background in one region may be quite inappropriate in another. A significant fraction of the authors’ scientific interest lies in the study of the diffuse X-ray background so we need a method that will not throw out our “signal”. Other methods, such as that of Arnaud et al. (2001), Read & Ponman (2003), and Nevalainen, Markevitch, & Lumb (2005), have relied more on blank-field data for their subtractions. A comparison between the methods used in the study of clusters of galaxies using Nevalainen et al. (2005) results is shown in Figure 2. As expected, the results of the two methods are in reasonably good agreement with the size of the uncertainties at large annuli being smaller using the XMM-ESAS method. At smaller annuli the background is small compared to the signal so any differences in the fit results are expected to be minor. Also included in Figure 2 are the results from the *Chandra* data analyzed by Vikhlinin et al. (2005). The discrepancy between *XMM-Newton* and *Chandra* was determined to arise from the *Chandra* calibration, which has been subsequently corrected, although inconsistencies still exist.

### 1.2.1. The Useable Modes

The ESAS method for creating model background spectra and images relies on data from the unexposed (to the sky) corners of the detectors. This means that, for the PN, one can only process data from the **Full Frame** or **Full Frame Extended** modes. For the other modes, the corner data are not recorded.

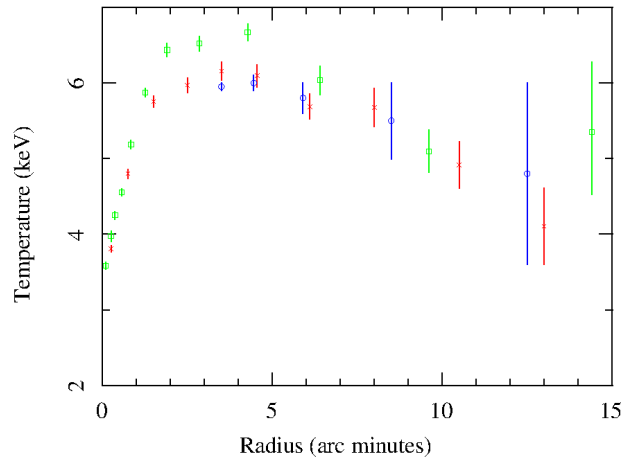


Fig. 2.— Comparison of the fitted temperature profile for the Abell 1795 cluster of galaxies as determined by the (red) XMM-ESAS method, the (blue) double background subtraction method of Nevalainen, Markevitch, & Lumb (2005), and the (green) *Chandra* analysis of Vikhlinin et al. (2005).

For the MOS detectors, CCDs 2 through 7 always operate in Full Frame mode, providing ESAS-compatible imaging for all observations, at least for the outer CCDs. ESAS can also process the Large and Small Window modes for MOS CCD 1 but for typical observations with bright sources the results may not be particularly useful due to the broad wings of the PSF. It is generally a good idea to deselect the central CCD if it is in a window mode. If the central CCDs are operated in *timing* mode they *must* be deselected in *mosspectra* (and other tasks where relevant). We should also note that in window mode there may be interference noise that appears in other chips. We have not yet studied its effect, so be aware of it and be careful.

### 1.2.2. The Useable Exposure Times

Observations with small filtered exposures ( $\sim 5$  ks or less) are likely to be problematic due to the limited number of events in the unexposed corners of the detectors (decent statistics are needed for accurate background modeling). Exposures in the  $\sim 5 - 10$  ks range can also be problematic, particularly for the pn, with the QPB being over- or under-subtracted. When the QPB is under-subtracted the fitting of the SP component can correct for this effect, but in an uncertain manner with unknown systematics. We expect significant improvements with the next iteration of ESAS.

### 1.2.3. The Spacecraft Position

We note that the background does depend upon the location of the spacecraft with respect to the particle belts and with respect to the magnetopause (Kuntz & Snowden in prep). Although these results will not be

fully exploited in this version of ESAS, if there are issues with the MOS QPB seeming to be over- or under-subtracted at lower energies, it is worthwhile to check to see how close to the particle belts the observation was made.

### 1.3. MOS Reduction *versus* pn Reduction

#### 1.3.1. The Corner Problem

There may be an issue concerning an under- or over-estimation of the pn particle background, particularly for relatively short observations (less than  $\sim 10$  ks of good time). This is primarily due to the poor statistics of the pn corner data as they are both limited in physical extent and they are affected by OOT events.

However, there is a secondary problem; the pn corner data are much more sensitive to SP flares than previously understood. The current pn QPB spectra are thus not as free of SPF emission as they should be. (This is being corrected, but probably will not be available until V22.0.) Thus, over-subtraction of the pn background *may* occur. Differences between the background subtracted MOS spectra and the background subtracted pn spectrum should be treated with caution, and the MOS spectra should be trusted somewhat more than the pn spectrum.

#### 1.3.2. The Pattern Problem

Note that the pn has a very strong low energy tail that can be exacerbated when using two-pixel (`PATTERN <= 4`) events, affecting data at energies as high as 600 eV. Starting with SAS V17, ESAS has allowed the selection of either one-pixel (`PATTERN == 0`) events or one- and two-pixel (`PATTERN <= 4`) events. There are several trade-offs that the user will need to consider. If low-energy data ( $E < 1$  keV) data are of primary interest then only one-pixel events should be used and the analysis can be extended down to 0.3 keV. If high-energy data are of primary interest then both one- and two-pixel events should be used as excluding two-pixel events will seriously limit the number of counts accumulated at higher energies. In this case the analysis can be extended down to only 0.4 keV (or perhaps 0.45 keV) with the caveat that the spectrum should be examined for an anomalously strong low-energy tail.

An alternative is to process the observation extracting both a one-pixel event spectrum and a one- and two-pixel event spectrum. If hard and soft band images are going to be created then this doesn't require any additional effort beyond bookkeeping. Process the bands with energies less than the Al-K $\alpha$  line (1.486 keV) with `pattern=0` in the call to *pn spectra* (`PATTERN == 0` in the selection expression) and the bands including and above the Al-K $\alpha$  line with

`pattern=4` in the call to *pn spectra* (`PATTERN <= 4`). Fit the `pattern=0` and `pattern=4` spectra simultaneously using an energy selection of, e.g., 0.3 – 2.0 keV for the `pattern=0` data and an energy selection of, e.g., 1.0 – 11.0 keV for the `pattern=4` data to examine the overlap region for confidence that the processing worked correctly. The image generation requires no additional effort beyond modifying the *pn spectra pattern* parameter.

### 1.4. More on Background Components

#### 1.4.1. Quiescent Particle Background

The shape of the QPB spectrum in the continuum is *relatively* uniform over a given chip; the spectral shape and the normalizations vary from chip to chip. (Stand by for the publication demonstrating that!) The strength of the lines does vary across chips, and different species vary in different ways (see Kuntz & Snowden 2008). Both the continuum and the lines of the QPB spectrum for the pn vary across the detector, sometimes quite dramatically (see Figure 3 and Lumb et al. 2002).

For spectroscopic investigations, the background is summed over different regions, so the variation is not a major issue for regions that cover a large fraction of the FOV. Imaging studies are often in relatively narrow energy bands. Thus the number of counts available from which to determine the background may be limited, and if the band contains a component that varies strongly with location, artifacts may occur. Imaging with the pn around 8.0 keV may be particularly problematic.

#### 1.4.2. Residual Soft Proton Contamination

Although a temporal filtering of the data will remove considerable contamination from SP flaring, and perhaps all of it for some observations, there is no guarantee that there will not be a significant residual flux. It has been our experience that this is the case for the majority of observations. We suggest an approach to account for residual SP contamination in the spectral fitting section below. Kuntz & Snowden (2008) characterizes the reasonable ranges of the variations in both the spectrum and spatial distribution over the detectors. Building on their results, tasks are included in this package which will spatially model fitted SP contamination in order to subtract it from images.

#### 1.4.3. Solar Wind Charge Exchange Contamination

Solar wind charge exchange (SWCX) emission can contribute significantly to the observed flux below 1.5 keV (see Figure 4), as shown in Snowden, Collier, & Kuntz (2004). SWCX emission may occur both near the Earth, in the Earth's magnetosheath, where the

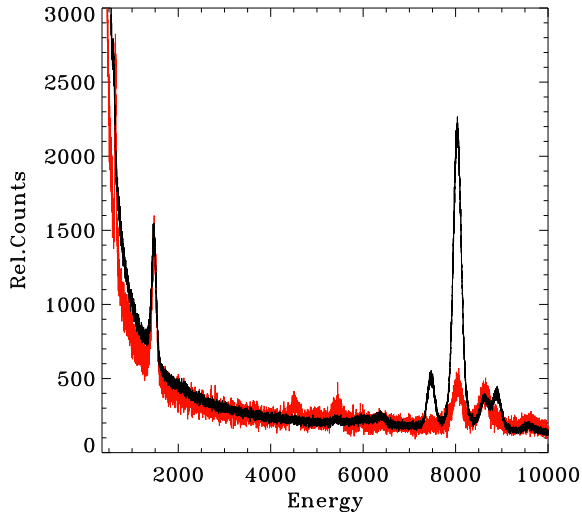
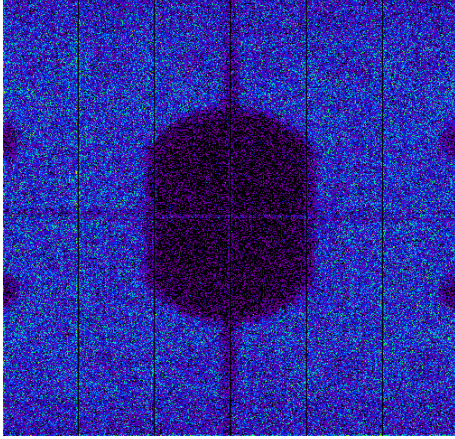


Fig. 3.— Image showing the QPB pattern in the 7.9-8.2 keV band. The *lower panel* shows the FWC spectrum in the central hole (*red*), and outside the central hole (*black*).

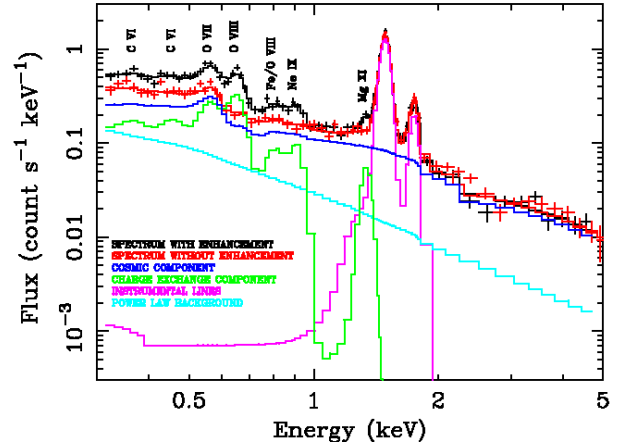


Fig. 4.— *XMM-Newton* MOS SWCX contaminated spectrum compared to an uncontaminated spectrum; both ObsIDs are from the *Hubble* Deep Field North direction that were analyzed in Snowden, Collier, & Kuntz (2004). Note the very strong excess O VIII emission, as well as other line emission, in the SWCX spectrum. These are also lines of interest as diagnostics of astrophysical plasmas.

solar wind interacts with exospheric material (e.g., Snowden et al. 2009), or distributed throughout the solar system where the solar wind interacts with interstellar neutrals (hydrogen and helium) flowing through the solar system (e.g., Koutroumpa et al. 2009). Further information can be found in Kuntz (2019).

The X-ray spectrum of the SWCX in the *XMM-Newton* band is dominated by line emission from ions that are also of astrophysical interest, e.g., O VII and O VIII. The magnitude of the emission is strongly variable, as are the line ratios that follow the flux, abundances, and ionization states of the solar wind. Thus, the use of standard line ratio plasma diagnostics, at least for diffuse emission filling the field of view, is extraordinarily sensitive to the amount of SWCX emission.

Carter & Sembay (2008) have published a method for identifying observations affected by SWCX emission based on the variation of the full FOV light curve for  $E < 1.35$  keV. This method is sensitive to the magnetospheric SWCX, but *only* if the observation is sufficiently long. The time scales for variation of the heliospheric SWCX are too long to allow a similar identification.

The *XMM-Newton* GOF has created a “Trend” data base<sup>6</sup> and associated software which can identify observations which have a greater likelihood of being

<sup>6</sup>Trend files contain *XMM-Newton* geometric (e.g., position and pointing) information, EPIC detector (e.g., operation mode and count rate) information, as well as relevant solar wind parameters. Trend data can be found at: [http://heasarc.gsfc.nasa.gov/docs/xmm/xmmhp\\_trend.html](http://heasarc.gsfc.nasa.gov/docs/xmm/xmmhp_trend.html).



affected by magnetospheric SWCX emission based on both observation geometry and solar wind flux.

We note that reliance upon the solar wind flux, as measured by upstream monitors such as ACE, Wind, or DSCOVR has significant uncertainties for reasons discussed at length by Kuntz (2019). Further, abundance data has become increasingly sparse over the last several years. Thus, it is now more difficult to assess the likelihood of significant SWCX contamination.

### 1.5. The Problem of Sparseness

Except for the centers of bright clusters of galaxies and particularly bright supernova remnants, most observations of extended sources will be photon limited, i.e., there will be insufficient detected events to make use of the full EPIC angular resolution ( $\sim 6''$  FWHM on-axis) beyond the excision of point sources. Even bright sources will be photon limited at high and low energies where the instruments' effective areas are reduced. Spatial binning of the data will therefore be needed to pursue any significant scientific studies. The pixel size of the images produced by XMM-ESAS is  $2.5''$ , which is a fine enough sampling to provide sufficient flexibility for creative binning (e.g., binning by high-resolution optical data) and the removal of contaminating point sources. For comparison, the physical size of the MOS pixels are  $1.1''$ , while the individual events are stored in  $0.05''$  pixels. The physical size of the pn pixels are  $4.1''$ , and are also stored in  $0.05''$  pixels.

## A Request

If the ESAS package has been useful for the publication of your results, we would appreciate an acknowledgment in your paper. It will help with our statistics and the justification for the continued maintenance of ESAS, and the existence of the XMM GOF.

## 2. CHANGE LOG

### 2.1. Summary

This new version of ESAS is (almost) totally integrated into SAS and obeys the SAS standards. For example, the standard SAS command line flags such as `-c`, `-h`, and `-V` are now fully implemented for the ESAS tasks. In an effort to become more “SAS-like”, many of the routines that formerly relied on fixed file names for input/output files have been modified to allow user-specified names. Of course this requires more non-default input parameters, but will allow greater flexibility in the application of these routines. In order to help users familiar with the old ESAS convert to the new ESAS, we include a table showing the equivalencies between the old and new file names.

The mixture of base routines and scripts has engendered some confusion about what is required for XMM-ESAS to work, and what is strictly a matter of convenience. To relieve some of this confusion, some of the scripts, such as *mos-filter*, have been broken into more task specific pieces. The *cheese* task remains, but we now document how to use its components. Conversely, the *mos-spectra* and *pn-spectra* perl scripts have been completely converted to FORTRAN, *mosspectra* and *pnspectra*, and have grown in functionality. We note while these routines have, historically, been the slow choke-point for ESAS, we now have used options in the underlying *backscal* routines for speeding the processing. The *mos\_back* and *pn\_back* routines have been rewritten to become *mosback* and *pnback*, with some changes in calling and capability.

This version of ESAS introduces a new task, *emanom* which should, for the purposes of ESAS, replace *emtaglenoise* for determining whether a MOS chip is in an anomalous state.

Finally, an almost invisible change is that there is no longer a separate ESAS CCF; (almost) all ESAS calibration files are now within the standard CCF, and are called by standard SAS routines rather than fitsio routines. The remaining ESAS calibration files are now contained within the SAS distribution; we expect to eliminate/replace them in future versions.

Table 1: Change Log

Change Log		
SAS	Task	Change
V10.5	<i>pn_back</i>	Additional diagnostic output
–	<i>espfilt</i>	Improve the output plotting
–	CalDB files	Extensive reformatting of files requiring changes in multiple tasks
–	<i>pn_back</i>	New release – still under development
–	<i>cheese-bands</i>	New release
–	<i>proton_scale</i>	Parameter and input file changes for <code>mode=2</code>
–	<i>mos_back</i>	Bug fix affecting Mac users
–	<i>proton</i>	Combined CCF flare files
–	<i>comb</i>	Updated scale factors for merging data from different filters – assumed hardness now selectable
–	<i>merge_comp_xmm</i>	Same as <i>comb</i>
–	<i>swcx</i>	Cast SWCX background images
V11.0	CalDB	Updated QPB and FWC files – NEW DOWNLOAD REQUIRED
–	<i>mos_back</i>	Check for MOS1 CCD#6 status
–	<i>mos-spectra</i>	Check for MOS1 CCD#6 status
–	<i>proton</i>	Check for MOS1 CCD#6 status
–	<i>swcx</i>	Check for MOS1 CCD#6 status
–	<i>bin_image</i>	Implement SWCX processing
–	<i>bin_image_merge</i>	Implement SWCX processing
–	<i>adapt_2000</i>	Implement SWCX processing
–	<i>merge_comp_xmm</i>	Implement SWCX processing
–	<i>conv-region</i>	Simplify region description for multiple observations
V12.0	CalDB	Updated QPB and FWC files – NEW DOWNLOAD REQUIRED
–	–	Most critical is the conversion of MOS analysis to 5 eV channels
–	<i>comb</i>	Bug fix and correct treatment of multiband exposure maps
–	<i>bin_image</i>	Bug fixed
–	<i>merge_comp_xmm</i>	Added units and comment line to output images
–	<i>mos-filter</i>	Diagnostic output added to check for anomalous states
–	<i>mos-spectra</i>	Convert MOS processing to use 5 eV channels
–	<i>mos_back</i>	Fix ObsID misidentification in <i>*-spec.qdp</i>
–	–	Fixes required by changed file format for <i>emask</i> output
–	–	Changes required to implement 5 eV energy bins
–	–	Add additional diagnostic information in <i>*-aug.qdp</i>
–	–	Fixes required for updated QPB files
–	–	Fixes required for revised version of <code>read_rmf</code>
–	<i>pn_back</i>	Fixes required for revised version of <code>read_rmf</code>
–	<i>adapt_900</i>	Add smoothing scale image and qdp plot output
–	<i>adapt_2000</i>	Add smoothing scale image and qdp plot output
–	<i>swcx</i>	Add pulse-height smearing
–	–	Requires a new command parameter
–	<i>read_rmf</i>	Subroutine in <code>subs.f</code> – actually return the response matrix
–	<i>cheese</i>	Fixes required by changed file format for <i>emask</i> output
–	<i>make_mask_merge</i>	Fixes required by changed file format for <i>emask</i> output
–	–	Modify so that it will run with just the output from <i>cheese</i>
–	<i>esmosaic_prep</i>	New task, companion task to <i>prep_mosaic</i> which adds
–	–	the correct pointing keywords to the fits headers of the separated
–	–	observation segments. Not a final solution but works with the
–	–	correct pixie dust sprinkled liberally.

Change Log - *continued*

SAS	Task	Change
V13.0	General	Conversion of all FORTRAN 77 routines to FORTRAN 90/95
–	<i>esprep_mosaic</i>	New task to allow processing of mosaic-mode observations
–	<i>esprep_mosaic_time</i>	New task to extract the time intervals for mosaic-mode observations
–	<i>esas_mosaic</i>	Driver task to overwrite header keywords for mosaic-mode observations
–	<i>comb</i>	Bug fix for scalings using alpha between 1.7 and 2.4
–	<i>merge_comp_xmm</i>	Bug fix for scalings using alpha between 1.7 and 2.4
–	<i>conv_reg</i>	New task to convert region information from sky coordinates to detector coordinates
–	–	
–	<i>adapt_merge</i>	Renaming of <i>adapt_2000</i> to <i>adapt_merge</i>
–	<i>adapt</i>	Renaming of <i>adapt_900</i> to <i>adapt</i>
–	Various	Inclusion of <i>clobber</i> parameter
V13.0.1	<i>esprep_mosaic_time</i>	Bug fix
V13.0.3	<i>mos-filter</i>	Fix for processing MOS1 CCD#4 data after loss of CCD#3
–	<i>espfilt</i>	Fix for processing MOS1 CCD#4 data after loss of CCD#3
–	<i>mos-spectra</i>	Fix for processing MOS1 CCD#4 data after loss of CCD#3
–	<i>mos_back</i>	Include corner data from CCD#2 and CCD#7 for the calculation of CCD#1 background
–	–	
–	<i>mos_back</i>	Bug fix – MOS background spectra uncertainties were too large
–	<i>make_mask_merge</i>	Code improvement
–	<i>rot-im-det-sky</i>	Bug fix and addition of modes to allow merging of masks
–	<i>rot_det_sky</i>	Addition of modes to allow merging of masks
V15.0	Various	Removal of MOS1 CCD#4 fixes data after loss of CCD#3
–	<i>merge_comp_xmm</i>	Bug fix to allow <i>ehigh&gt;9999</i>
–	<i>rot_im_det_sky</i>	Removal of MOS1 CCD#4 fixes data after loss of CCD#3
–	<i>espfilt</i>	Removal of MOS1 CCD#4 fixes data after loss of CCD#3
–	<i>mos-spectra</i>	Removal of MOS1 CCD#4 fixes data after loss of CCD#3
–	<i>mos-back</i>	Removal of MOS1 CCD#4 fixes data after loss of CCD#3
–	<i>conv_reg</i>	Bug fix
V16.0	<i>adapt</i>	Cosmetic changes to radial plot output
–	<i>cheese-bands</i>	Eliminated, capabilities included in <i>cheese</i>
–	<i>cheese</i>	Expanded to process one or two bands
–	<i>cheese</i>	Converted to be a wrapper around <i>edetect_chain</i>
–	<i>cheese</i>	Modified to bring better agreement between DETXY and XY masking
–	<i>conv-reg</i>	Bug fixes
–	<i>conv-region</i>	Eliminated as no longer needed
–	<i>esas_mosaic</i>	Eliminated as no longer needed
–	<i>esprep_mosaic</i>	Eliminated as no longer needed
–	<i>esprep_mosaic_times</i>	Eliminated as no longer needed
V17.0	Various	Inclusion of CalDB files in the CCF structure
–	CalDB	Add MOS canned corner spectra
–	–	Add pn single-pixel QDP and canned corner spectra
–	<i>cheese</i>	Uniform usage of the <i>radiusstyle=contour</i> parameter for source excision
–	<i>mos_back</i>	<del>Inclusion of capability to make count background spectra (CSTAT)</del>
–	–	<del>Inclusion of capability to use canned corner spectra</del>
–	<i>pn_back</i>	<del>Inclusion of capability to make count background spectra (CSTAT)</del>
–	–	<del>Inclusion of capability to use canned corner spectra</del>
–	–	Add the ability to select either single- or double-pixel events
–	<i>pn-spectra</i>	Add the ability to select either single- or double-pixel events

Change Log - *continued*

SAS	Task	Change
V 21.0	Various and Sundry	
–	<i>general</i>	change of input/output file name convention, convention for passing CCD information.
–	<i>binadaptmerge</i>	Renamed from <i>adapt_merge</i> and combined with <i>bin_merge</i> .
–	<i>binadapt</i>	Created to contain the functionality of <i>bin_image</i> and <i>adapt</i> .
–	<i>cheese</i>	Converted from perl to FORTRAN, HEASOFT useage eliminated, filenaming convention changed, parameter names changed.
–	<i>cheese-bands</i>	Now part of <i>cheese</i> .
–	<i>comb</i>	Changed name to <i>combimage</i> . Parameters changed. Functionality unchanged.
–	<i>combimage</i>	Changed name from <i>comb</i> . Parameters changed. Functionality unchanged.
–	<i>emanom</i>	Introduction of a new task, equivalent functionality removed from <i>mos-filter</i>
–	<i>emlfill</i>	Changed name from <i>file_list</i> .
–	<i>emlmerge</i>	New version of <i>merge_source_list</i> .
–	<i>espfilt</i>	Flags rectified, histogram fit parameters output to header, root name of output files determined from subid in header, more user tuneable parameters added, ability to turn off extra output added, changed some output file names
–	<i>file_list</i>	Changed name to <i>emlfill</i> . Parameters changed.
–	<i>merge_comp</i>	Changed name to <i>mosaicmerge</i> . Parameters changed.
–	<i>merge_source_list</i>	Changed name to <i>emlmerge</i> . Parameters changed.
–	<i>mosback</i>	Changed name from <i>mos_back</i> , changed some parameter names and input/output file names.
–	<i>mos-filter</i>	Retired task as just a wrapper for <i>espfilt</i> .
–	<i>mosaicmerge</i>	Changed name from <i>merge_comp</i> . Parameters changed. Functionality unchanged.
–	<i>mosspectra</i>	Converted from the perl script <i>mos-spectra</i> , change of input parameters and input/output file names, accelerated by increasing <b>badpixelresolution</b> parameter passed to <i>backscale</i> , now using SOC provided FWC files
–	<i>pnback</i>	Changed name from <i>pn_back</i> , changed some parameter names and input/output file names, now QPB files separated by mode
–	<i>pn-filter</i>	Retired task as just a wrapper for <i>espfilt</i> .
–	<i>pnspectra</i>	Converted from the perl script <i>pn-spectra</i> , change of input parameters and input/output file names, accelerated by increasing <b>badpixelresolution</b> parameter passed to <i>backscale</i> , now using SOC provided FWC files
–	<i>pointsrc</i>	Changed name from <i>point_source</i> . Parameters changed. Functionality unchanged.
–	<i>proton</i>	Complete re-write with a change in parameters. Functionality unchanged.
–	<i>protonscale</i>	Changed name from <i>proton_scale</i> . Parameters changed.
–	<i>rotdet2sky</i>	Converted from the perl task <i>rot-im-det-sky</i> , made more general (no longer needs to know what type of file is input (QPB, SPF, SWCX, etc.), and input/output files explicitly passed as parameters
–	<i>rot-im-det-sky</i>	Changed name to <i>rotdet2skyl</i> .
–	<i>sppartial</i>	Changed name from <i>sp_partial</i> . Parameters changed. Functionality unchanged.
–	<i>swcx</i>	Parameters changed. Functionality not significantly changed.

## 2.2. Concordance

Because the changes in ESAS have been so drastic, the following tables will help users understand the transformation of file names between the old ESAS and the current ESAS.

Table 2: Routine Concordance between Old ESAS and Current ESAS

Routine Concordance		
Old Routine	Old Type	New Routine <sup>a</sup>
adapt	FORTRAN	binadapt
adapt_merge	FORTRAN	binadaptmerge
bin_image	FORTRAN	binadapt
bin_image_merge	FORTRAN	binadaptmerge
cheese	PERL	cheese
cheese-bands	PERL	cheese
comb	FORTRAN	combimage
conv_reg	FORTRAN	convregion
fill_list	FORTRAN	emlfill
	FORTRAN	emanom
merge_comp	FORTRAN	mosaicmerge
merge_source_list	FORTRAN	emlmerge
mos_back	FORTRAN	mosback
mos-filter	PERL	espfilt
mos-spectra	PERL	mospectra
pn_back	FORTRAN	pnback
pn-filter	PERL	espfilt
pn-spectra	PERL	pnspectra
point_source	FORTRAN	pointsrc
proton	FORTRAN	proton
proton_scale	FORTRAN	protonscale
rot-im-det-sky	PERL	rotdet2sky
sp_partial	FORTRAN	sppartial
swcx	FORTRAN	swcx

<sup>a</sup> All current routines are in SAS-consistent FORTRAN.

Table 3: File Name Concordance

1/O <sup>a</sup>		Name <sup>b</sup>		Diag. <sup>c</sup>	Inter. <sup>d</sup>	Coord. <sup>e</sup>	Mask <sup>f</sup>	Used	By <sup>g</sup>	Old	Comment	New
←	→	Old	New									
←	→	mosl/2	mosl/2 -anom.log	✓						nonexistent		
→	→	P-corn.fits	P-corev.fits	✓	✓	espfilt (was mos-filter or pn-filter)		nothing further		FLAG & 0x766a0f63	FLAG & 0x766a000	FLAG & 0x766a000
→	→	P-corn-image.fits	P-cornim.fits	✓			D	-		780,-19499:19500	"600" × "600", 1:640	"600" × "600", 1:640
→	→	P-ratec.fits	P-cornimc.fits	✓			D	-		FLAG & 0x766a0f63	FLAG & 0x766a000	FLAG & 0x766a000
→	→	P-clean.fits	P-allevc.fits	✓				-	cheese, mos/pnspectra	FLAG & 0x766a0f63	FLAG & 0x766a000	FLAG & 0x766a000
→	→	P-obj-image-det.fits	P-allim.fits	✓			D	-		780 <sup>2</sup> , -19499:19500, Int32		
→	→	P-obj-image-det-soft.fits	P-obj-image-det-unfilt.fits	✓			D	-		780 <sup>2</sup> , -19499:19500, Int32		
→	→	P-obj-image-det-obj-8.fits	P-obj-image-sky.fits	✓			D	-		780 <sup>2</sup> , -19499:19500, Int32		
→	→	P-ori.fits	P-allimc.fits	✓			S	-	good question	900 <sup>2</sup> , 3401:48400, Int32, FOV		"600" × "600", 1:640
→	→	P-rate.fits	P-fovlc.fits	✓				-	nothing further	renamed input file		not renamed
→	→	P-gti.txt	P-gti.fits	✓				-	not used (as per SS)	FLAG & 0x766a0f63		FLAG & 0x766a000
→	→	P-gti.fits	P-gti.fits	✓				-	nothing further			not produced
→	→	P-hist.qdp	P-hist.qdp	✓				-				no change
←	←	P-clean.fits	P-allevc.fits			cheese						minor changes
→	→	atthk.fits	atthk.fits	✓								no change
→	→	eboxlist_1.fits	eboxlist_1.fits	✓								no change (from atthkgen)
→	→	eboxlist_2m.fits	eboxlist_2m.fits	✓								no change (from edetect_chain)
→	→	emllist.fits	emllist.fits	✓								no change (from edetect_chain)
→	→	P-bkg-region-det-8.fits	P-bkgregion-det.fits	✓								no change (from edetect_chain)
→	→	P-bkg-region-sky-8.fits	P-bkgregion-sky.fits	✓								no change (from region)
→	→	P-cheese-8.fits	P-cheese8.fits	✓								no change (from region)
→	→	P-obj-im.fits	P-fovinB.fits	✓			S			900 <sup>2</sup> , Int32		no change (from make_mask)
→	→	P-obj-im-8.fits	P-fovinB8.fits	✓			S			900 <sup>2</sup> , 3401, 48400, Int32, FOV & C		
→	→	P-mask-im.fits	P-fovinBexp.fits	✓								
→	→		P-fovinBmask.fits	✓								
→	→		P-fovinBsen.fits	✓								
→	→		P-fovinBsmmap.fits	✓								
←	←	P-allevc.fits	P-allevc.fits			mos-spectra/mosspectra <sup>h</sup>						
←	←	atthk.fits	atthk.fits	✓								no change (from atthkgen)
→	→	P-loc.pi	P-corevcdm.pi	✓								
→	→	P-ffc.pi	P-corfwccdm.pi	✓								
→	→	P-obj.pi	P-fovecdm.pi	✓								
→	→	P-ff.pi	P-fovwccdm.pi	✓								
→	→	P-obj2.pi	P-fovt.pi	✓								
→	→	P-arf	P-arf	✓								
→	→	P-rmf	P-rmf	✓								
→	→	P-exp-im-elo-ehi-ccd1.fits	P-expimccd1-elo-ehi.fits	✓			D	A	-	780 <sup>2</sup> , FIt32, contains sel. chips		same
→	→	P-exp-im-elo-ehi.fits	P-expimsky-elo-ehi.fits	✓			S	A	comb	900 <sup>2</sup> , FIt32		same
→	→	P-exp-im.fits	P-expimt.fits	✓			S	A		900 <sup>2</sup> , FIt32		same
→	→	P-im-elo-ehi.fits	P-fwimccd1det-elo-ehi.fits	✓			D	ASR	-	780 <sup>2</sup> , Int32		same
→	→	P-mask-im-elo-ehi-ccd1.fits	P-maskimccd1-elo-ehi.fits	✓			D	A	-	780 <sup>2</sup> , Int32, contains sel. chips		same
→	→	P-mask-im-elo-ehi.fits	P-maskimsky-elo-ehi.fits	✓			S	A	comb	900 <sup>2</sup> , Int32 in 2 <sup>nd</sup> extension		same
→	→	P-mask-im.fits <sup>j</sup>	P-maskimt.fits	✓			S	-		900 <sup>2</sup> , Int32 in 2 <sup>nd</sup> extension		same
→	→	P-obj-im-elo-ehi-ccd1.fits	P-fovinccd1-elo-ehi.fits	✓			D	-		780 <sup>2</sup> , Int32, only ccd1		same
→	→		P-fovinint.fits	✓			S	-		900 <sup>2</sup> , Int32		same
→	→	P-obj-im-elo-ehi.fits	P-fovinmsky-elo-ehi.fits	✓			S	ASR	comb	900 <sup>2</sup> , Int32		same
→	→	P-obj-im-det-elo-ehi.fits	P-fovinmdet-elo-ehi.fits	✓			D	AR	-	780 <sup>2</sup> , Int32		same
→	→	P-obj-im-sp-det.fits	P-fovinmspdet.fits	✓			D	ASR	sppartial,proton_scale	780 <sup>2</sup> , Int32		same

File Name Concordance - *continued*

$1/O^a$	Old	Name <sup>b</sup> New	Diag. c	Inter. d	Coor. e	Mask f	Used By g	Old	Comment New
←									
→	P-back.pi	P-bkg.pi					USER		
→	P-back-im-det-elo-ehi.fits	P-bkgimdet-elo-ehi.fits			D	ASR	rot-im-det-sky → comb	780 <sup>2</sup> _Fit32	same
→	P-aug.qdp	P-ratehard.qdp	✓				-		
→	P-aug-indiv.qdp	P-augindiv.qdp	✓				-		
→	P-back.qdp	P-bkgspec.qdp	✓				-		
→	P-back-accum.qdp	P-bkgaccum.qdp	✓				-		
→	P-back-indiv.qdp	P-bkgindiv.qdp	✓				-		
→	P-bridge-fit.qdp	P-bridgefit.qdp	✓				-		
→	P-ratio-indiv.qdp	P-ratioindiv.qdp	✓				-		
→		P-augspec.qdp	✓				-		
←									
←		P-allevc.fits							
←		P-allevcoot.fits							
←		atthk.fits							
→							this routine		no change (from atthkgen)
→	P-corn.fits	P-corevc.fits							
→	P-corn-oot.fits	P-corevcoot.fits							
→		P-corcq.pi	✓						pnback
→	P-qoc-oot.pi	P-coroottq.pi	✓						pnback
→	P-qfc.pi	P-corfwcq.pi	✓						pnback
→	P-qfc-oot.pi	P-corfwoottq.pi	✓						pnback
→	P-qobj.pi	P-fovq.pi	✓						pnback
→	P-qobj-oot.pi	P-fovoottq.pi	✓						pnback
→	P-qff.pi	P-fovfwcqq.pi	✓						pnback
→	P-qff-oot.pi	P-fovfwcoottq.pi	✓						pnback
→	P-obj.pi	P-fovtoot.pi	✓						USER
→	P-obj-oot.pi	P-fovtooot.pi	✓						USER
→		P-arf							USER
→		P-rmf							USER
→									
→	P-exp-im-elo-ehi.fits	P-expimsky-elo-ehi.fits			S	-	comb	900 <sup>2</sup> _Fit32	same
→	P-exp-im-det-elo-ehi.fits	P-expimdet-elo-ehi.fits	✓		D	-	-	780 <sup>2</sup> _Fit32	not produced
→	P-exp-im.fits	P-expim.fits	✓?		S	-	-	900 <sup>2</sup> _Fit32	same
→	P-imq-elo-ehi.fits	P-fwcimqdet-elo-ehi.fits	✓	✓	D	SR	-	780 <sup>2</sup> _Int32	same
→	P-imq-elo-ehi-oot.fits	P-fwcimqcootdet-elo-ehi.fits	✓	✓	D	SR	-	780 <sup>2</sup> _Int32	same
→	P-mask-im-det-elo-ehi.fits	P-maskimsky-elo-ehi.fits			S	-	comb	900 <sup>2</sup> _Int32	same
→	P-mask-im.fits	P-maskim.fits	✓		D	-	-	780 <sup>2</sup> _Int32	not produced
→	P-obj-im-elo-ehi-oot.fits	P-fovimsky-elo-ehi.fits	✓		S	-	-	900 <sup>2</sup> _Int32	extension
→	P-obj-im-elo-ehi.fits	P-fovimskyoot-elo-ehi.fits			S	SR	comb	900 <sup>2</sup> _Int32	same
→	P-obj-im-det-elo-ehi-oot.fits	P-fovimdetoct-elo-ehi.fits	✓?		D	SR	-	780 <sup>2</sup> _Int32	same
→	P-obj-im-det-elo-ehi.fits	P-fovimdetoct-elo-ehi.fits	✓?		D	SR	-	780 <sup>2</sup> _Int32	same
→	P-obj-im-oot.fits	P-fovimtcoot.fits	✓		S	-	-	900 <sup>2</sup> _Int32	same
→		P-fovimt.fits	✓		S	-	-	900 <sup>2</sup> _Int32	same
→	P-obj-im-sp-det.fits	P-fovimsdet.fits	✓		D	SR	sppartial,proton,scale	780 <sup>2</sup> _Int32	same
←									
←									
→	P-back.pi	P-bkg.pi							
→	P-obj-op.pi	P-fovtootsub.pi							
→									
→	P-back-im-det-elo-ehi.fits	P-bkgimdet-elo-ehi.fits			D	ASR	rot-im-det-sky → comb	780 <sup>2</sup> _Fit32	same
→	P-aug.qdp	P-ratehard.qdp	✓						
→	P-aug-rev-hard.qdp	P-augratehard.qdp	✓						
→	P-aug-spec.qdp	P-augraterev.qdp	✓						
→	P-back-spec.qdp	P-bkgspec.qdp	✓						
→	P-bridge-fit.qdp	P-bridgefit.qdp	✓						
→	P-quad-spec.qdp	P-quadspec.qdp	✓						
→	P-spec.qdp	P-augspec.qdp	✓						



# File Name Concordance - continued

$1/O^a$	Old	Name <sup>b</sup> New	Diag. <sup>c</sup>	Inter. <sup>d</sup>	Coord. <sup>e</sup>	Mask <sup>f</sup>	Used By <sup>g</sup>	Old	New	Comment
←	P-obj-im-det-elo-ehi.fits	P-fovindet-elo-ehi.fits			proton					
←	P-obj.pl	P-fov.pl								
←	P-prot-im-det-elo-ehi.fits	P-protimdet-elo-ehi.fits					rot-im-det-sky→comb			
←	P-obj-im-det-elo-ehi.fits	P-fovindet-elo-ehi.fits								
←	P-obj.pl	P-fov.pl								
←	P-swxc-im-det-elo-ehi.fits	P-swxcimdet-elo-ehi.fits			swcx		rot-im-det-sky→comb			
←	P-back-im-sky-elo-ehi.fits	P-bkgimsky-elo-ehi.fits			comb/combimage			900 <sup>2</sup>		
←	P-cheese	<b>P-cheese-elo-ehi.fits</b>						900 <sup>2</sup>		option c but not currently made!
←	P-exp-im-elo-ehi.fits	P-expimsky-elo-ehi.fits						900 <sup>2</sup>		
←	P-mask-im-elo-ehi.fits	P-maskimsky-elo-ehi.fits						900 <sup>2</sup>		option e
←	P-obj-im-elo-ehi.fits	P-fovimsky-elo-ehi.fits						900 <sup>2</sup>		
←	P-obj-im-elo-ehi-oot.fits	P-fovimotksy-elo-ehi.fits						900 <sup>2</sup>		
←	P-prot-im-sky-elo-ehi.fits	P-protimsky-elo-ehi.fits						900 <sup>2</sup>		
←	P-swxc-im-sky-elo-ehi.fits	<b>P-nsi-elo-ehi.fits</b>						900 <sup>2</sup>		option m but what is it?
→	comb-obj-im-elo-ehi.fits	comb-fovim.fits					USER or mosaicmerge	900 <sup>2</sup>		
→	comb-back-im-sky-elo-ehi.fits	comb-bkgim.fits					USER or mosaicmerge	900 <sup>2</sup>		
→	comb-exp-im-elo-ehi.fits	comb-expim.fits					USER or mosaicmerge	900 <sup>2</sup>		
→	comb-prot-im-sky-elo-ehi.fits	comb-protim.fits					USER or mosaicmerge	900 <sup>2</sup>		
→	comb-swxc-im-sky-elo-ehi.fits	comb-swxcim.fits					USER or mosaicmerge	900 <sup>2</sup>		
←	(various)	(various)			adapt/binadapt				Variable size	
→	adapt-elo-ehi.fits	comb-adaptimsky-elo-ehi.fits							Variable size	
→	size-elo-ehi.fits	comb-size-elo-ehi.fits								
→	size-elo-ehi.qdp	comb-size-elo-ehi.qdp								
→	radial-fill-elo-ehi.qdp	comb-radfill-elo-ehi.qdp								
←	(various)	(various)			bin_image/binadapt					
→	rate-elo-ehi.fits	comb-rateimsky-elo-ehi.fits							Variable size	
→	rate-elo-ehi.fits	comb-simimsky-elo-ehi.fits							Variable size	
→	radial-fill-elo-ehi.qdp	comb-radfill-elo-ehi.qdp								
←	(various)	(various)			merge-comp_xmm/mergemosaic					
→	obj-im-elo-ehi.fits	mosaic-fovimsky-elo-ehi.fits						2000 <sup>2</sup>		
→	exp-im-elo-ehi.fits	mosaic-expimsky-elo-ehi.fits						2000 <sup>2</sup>		
→	back-im-elo-ehi.fits	mosaic-bkgimsky-elo-ehi.fits						2000 <sup>2</sup>		
→	prot-im-elo-ehi.fits	mosaic-protimsky-elo-ehi.fits						2000 <sup>2</sup>		
→	swxc-im-elo-ehi.fits	mosaic-swxcimsky-elo-ehi.fits						2000 <sup>2</sup>		
←	(various)	(various)			adapt_merge/binadaptmerge					
→	adapt-elo-ehi.fits	merged-adaptimsky-elo-ehi.fits							Variable size	
→	size-elo-ehi.fits	merged-size-elo-ehi.fits							Variable size	
→	size-elo-ehi.qdp	merged-size-elo-ehi.qdp								
→	radial-fill-elo-ehi.qdp	merged-radfill-elo-ehi.qdp								
←	(various)	(various)			bin_image-merge					
→	rate-elo-ehi.fits	merged-rateimsky-elo-ehi.fits							Variable size	
→	rate-elo-ehi.fits	merged-simimsky-elo-ehi.fits							Variable size	
→	radial-fill-elo-ehi.qdp	merged-radfill-elo-ehi.qdp								

<sup>a</sup> A "→" indicates output. A "←" indicates input. I've been sloppy and not put some inputs.  
<sup>b</sup> A P indicates a prefix, such as mos1S001. A B indicates a letter denoting the band, hard (h), soft (s), or total (t). A M indicates a CCD number for the MOS. A Q indicates a quadrant for the pn.  
<sup>c</sup> A check means that the file is diagnostic and not used in further analysis.  
<sup>d</sup> A check means that the file is an intermediate product in this program and not used in further analysis.  
<sup>e</sup> "D" for detector coordinates, "S" for sky coordinates.  
<sup>f</sup> "R" if the user supplied region file is applied, "S" if the source mask is applied, and "A" if the anomalous chip mask is applied.  
<sup>g</sup> A "USER" means that the output will be used by the user for analysis. That is, this file is one of the ultimate ESAS products.  
<sup>h</sup> The actual size varies.  
<sup>j</sup> This is not the same file produced by cheese, however the differences are minor.

### 2.3. Future

A large number of nearly invisible changes are being made in order to remain “SAS-compliant”. For example, reliance on `cfitsio`, `ftools`, and other HEASoft based tools is being phased out. SAS is also phasing out Perl routines in favour of Python, so ESAS no longer has any Perl routines. The arrival of Python *will* mean the final retirement of QDP from ESAS.

Improvements to the creation of the quiescent particle background spectra for the MOS are being developed (Kuntz & Snowden in prep) and substantial improvements are being implemented for the pn. Those improvements will appear in the next release of SAS. They will incorporate information about space weather to determine whether the calculated QPB is sufficient.

Table 4: Future Modifications

SAS	Task	Change
Future	<i>espfilt</i>	Tweak the histogram fitting algorithm to avoid some exceptional cases that have time intervals without counts
–	<i>espfilt</i>	More careful implementation of MOS small window mode selections
–	<i>espfilt</i>	Auto-remove time intervals with zero counts (due to data gap) to allow merged event files to be used
–	<i>emlfill</i>	Fix!
–	<i>mosspectra</i> & <i>pnspectra</i>	Better implementation of keepinterfiles
–	<i>mosspectra</i> & <i>pnspectra</i>	ensure the ability to use <code>MASK(filename)</code>
–	<i>mosback</i>	Automatic call to <i>rotdet2sky</i> for background image
–	<i>mosback</i>	rationalization of diagnostic plots, elimination of QDP
–	<i>pnback</i>	Automatic call to <i>rotdet2sky</i> for background image
–	<i>pnback</i>	rationalization of diagnostic plots, elimination of QDP
–	<i>proton</i>	Automatic call to <i>rotdet2sky</i> for background image
–	<i>swcx</i>	Automatic call to <i>rotdet2sky</i> for background image
–	<i>swcx</i>	Implementation of more SWCX lines
–	<i>combimage</i>	User specified scale factors
–	<i>adapt</i>	Produce uncertainty image, better diagnostic plots
–	<i>overlap</i>	New task to determine the offsets between overlapping images
–	<i>mosaicmerge</i>	Capability for larger mosaics?
–	<i>check_orbit</i>	This is a routine to determine which detectors are active for a given observation. It needs to be fixed for future use with MOS2 or pn; it currently only operates for MOS1.
–	<i>convregion</i>	Implement boxes
–	<i>protonscale</i>	Change name to <i>getbackscale</i>
–	<i>mosspectra</i> & <i>pnspectra</i>	New QPB implementation
–	<i>weather</i>	Build space weather data for a given observation
–	<i>gtiedit</i>	GUI editor for editing gti files
–	<i>TBD</i>	A routine to calculate the maximal region that allows coverage by all three detectors. This would be useful if one needed exactly the same sky region from each detector.
–	<i>general</i>	Determine action due to MOS2-1 edge issue
–	<i>TBD</i>	A routine to apply pn GTI to MOS exposures properly

### 3. CAVEAT EMPTOR

- 1) Be very careful about running more than one SAS task at a time in the same directory. This can have unforeseen consequences which may not be obvious, and will often produce erroneous results. **It is our strong recommendation that this not be done, ever.**
- 2) In this document when examples of task calls are provided they are broken into multiple lines for clarity. When actually invoking a task all parameters should be on one line (or the lines should be ended with the proper continuation character).
- 3) Observations with small filtered exposures ( $\sim 5$  ks or less) are likely to be problematic due to the limited number of events in the unexposed corners of the detectors; decent statistics are needed for accurate background modeling. Exposures in the  $\sim 5 - 10$  ks range can also be problematic, particularly for the pn, with the QPB being over- or under-subtracted. When the QPB is under-subtracted the fitting of the SP component can correct for this effect, but in an uncertain manner with unknown systematics.
- 4) Fitting the SP component when the residual contamination is relatively small can produce unreasonable values for the spectral index, especially for the pn. The fitted indices can run away to large values, either positive or negative. In general, reasonable indices lie between 0.15 and 1.5 when fitting a single power law or 0.1 and 2.5 for a broken power law. Note that when the fitted value for the SP normalization drops below a few times  $10^{-3}$  (for the full FOV) the values are effectively zero. There is no practical difference between normalizations of  $10^{-4}$  and  $10^{-30}$ .
- 5) Double-pixel events for the pn contribute an unacceptably high background below 0.4 keV and occasionally can affect data up to  $\sim 0.6$  keV. Because of this we recommend using only single-pixel (PATTERN == 0) events when data less than 0.7 keV are of interest. On the other hand, excluding double-pixel events (PATTERN <= 4) from the study of higher energies forgoes a significant part of the signal. We therefore recommend the following complication: process the data twice, creating two pn spectra with single- and double-pixel events. The spectra can then be simultaneously fit including 0.3 – 2.0 keV data from the single-pixel spectrum and 1.0–11.0 keV data from the double-pixel spectrum.
- 6) Multiple pn segments may exist for your observation. While *emchain* will process all MOS segments within an obsid, *epchain* will only process the first segment, unless you specifically tell it

which segment to process. You will have to determine the existence of the other segments and process them separately. This is discussed in §5.7.

- 7) If you update the CCF directory contents, *cifbuild* must be rerun.
- 8a) If *cifbuild* or *odfingest* have any problems, you should delete the *ccf.cif* file and/or the \*.SAS file in the *odf* directory and rerun *cifbuild* and *odfingest* after correcting the issue. A “corrupt” *ccf.cif* or \*.SAS file will allow SAS to *try* to do things if it’s there, and fail with often difficult to decipher error messages. (Our helpdesk guru notes that this issue is the most common source of problems.)
- 8b) The environment variable SAS\_CCF points to the *ccf.cif* that all tasks will use. Similarly, the \*.SAS file in the directory pointed to by SAS\_ODF. If you move on to another observation, SAS\_CCF and SAS\_ODF must be changed, or else you will not get the right results. We suggest doing only one obsid per window.
- 9) ESAS is undergoing major changes as it is being made more SAS compliant. If you have used ESAS in the past, please check the change logs for a first look at where your analysis methods/scripts may need to be changed.

#### 4. ESAS FLOW CHART

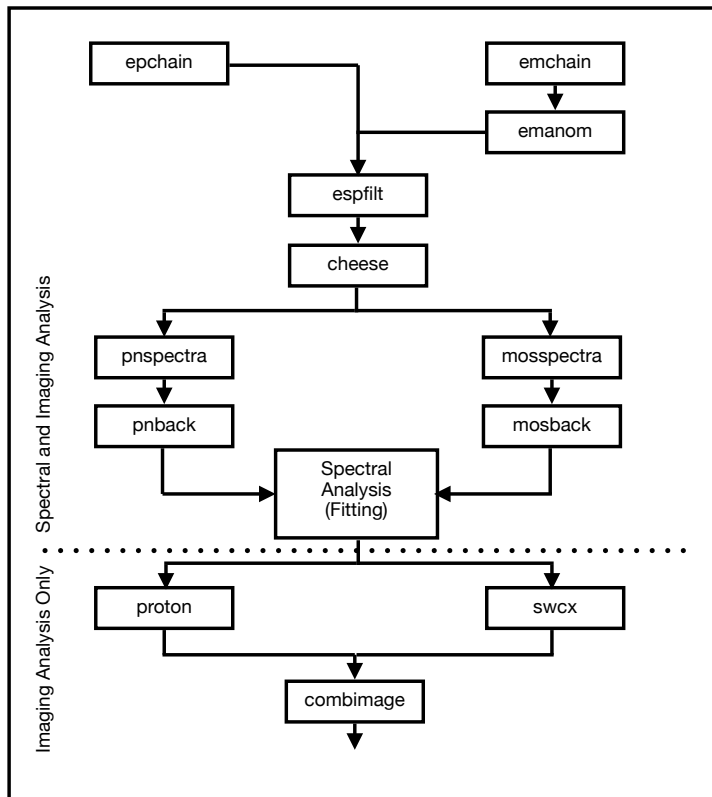


Fig. 5.— General flow chart for ESAS processing of *XMM-Newton* EPIC data for extended emission. The upper part must be done for both spectral analysis or imaging analysis. The lower part need be done only for imaging analysis. This flow chart is a bit simplified; if the region from which the soft proton rate was determined does not match the area for which one wishes to create a soft proton image, some other steps will be necessary.

## 5. XMM-ESAS COOKBOOK

### 5.1. The XMM-ESAS Package

The purpose of the XMM-ESAS package is to provide the tools required for the analysis of spectra and images of extended (relative to the instrumental PSF) diffuse emission. Its primary functions are to 1) measure, construct, and remove several different non-cosmic background components, 2) provide a relatively automated method for reducing extended diffuse data, and 3) provide a means for mosaicking images of extended emission (including background subtraction, exposure correction, and inter-instrument normalization).

The XMM-ESAS package before you consists of a number of SAS tasks with various functionalities. Most of these SAS tasks are direct inclusions of FORTRAN routines, while others are SAS FORTRAN versions of what had been perl scripts. If you should have some special circumstance not covered by the existing routines, you can probably accomplish what you need using basic SAS functions. On the other hand, if there is something nifty that you would like to accomplish that can't currently be done, please contact us with suggestions, as we are always open to adding new functionalities. A complete alphabetical list of ESAS routines appears at the beginning of Appendix A. The next sections contain the step-by-step instructions for using each of the routines in the order they need to be applied.

### 5.2. Calibration/CCF Files

There are calibration files associated with the XMM-ESAS tasks. The bulk of these files have been ported into the standard CCF format and are available via the standard CCF download. These files are required for the processing both spectra and images. For earlier SAS V18 and earlier versions they had to be downloaded separately. For those earlier versions of SAS, the local directory containing the CalDB data was used as an input parameter in many of the tasks, this specification is no longer necessary.

The CCF files should be downloaded to a dedicated CCF directory on your computer. To do so, move to your CCF directory, whatever name you have given it, and execute

```
wget -m -nH --cut-dirs=4 -e robots=off -l 2 -np
https://heasarc.gsfc.nasa.gov/FTP/xmm/data/CCF
```

(all on one line). Note that you can repeat this command in the future and the *wget* command will retrieve *only* the files that you are missing, so you won't need to download the

entire  $\sim 7$  GB to update. Better yet, go to [https://heasarc.gsfc.nasa.gov/docs/xmm/xmmgof\\_mirror.html](https://heasarc.gsfc.nasa.gov/docs/xmm/xmmgof_mirror.html) and follow the instructions given there (they tend to change!)

One issue to note about the CCF files: if you download a new batch of CCF files while you are in the middle of your analysis, you will need to repeat the *cifbuild* command in order to use those new files in that analysis.

### 5.3. XMM-ESAS Produced Files - Some Warnings

The XMM-ESAS package produces a fairly large number of intermediate files during the processing of an observation data set. Many shouldn't be touched while the processing is in progress. In the following, we will indicate whether output files are diagnostic (thus disposable), an intermediate step, or final output, in order to make it more clear what needs to be protected and what can be overwritten. However, as discussed below, if spectra are required from multiple regions of the same exposure, the final products (e.g., spectral files, RMFs, and ARFs) must be renamed or (better yet) moved before processing another region. The example scripts in Appendix B demonstrate a simple way of keeping track of everything. Similarly, some files will not change, so they should be retained in order to speed processing.

Among the developers there has been a great deal of discussion about *file names* and trying to get them to fit the SAS model. Unlike most SAS tasks, some ESAS tasks require multiple (sometimes an indefinite number) of input files and produce a prodigious number of output files. In previous versions of ESAS, input file names were partially hard-wired. The new ESAS tasks have been restructured so that, with some exceptions<sup>7</sup>, the input and output filenames can be either user determined or constructed by the routine itself.

### 5.4. How This Cookbook Works

Each of the following subsections describes a single step of processing. Each section begins with an explanation of what the step actually does, and then describes the commands that must be run to accomplish that step. This description is followed by a discussion of the output products. As many of the outputs are diagnostic, we attempt to explain how to use them. Many of the diagnostic files are not needed for further processing and can be deleted or saved elsewhere.

There are a number of different conventions for naming the individual CCDs of the MOS instruments.

---

<sup>7</sup>The exceptions are the plethora of files that are produced by *mosspectra* or *pnspectra* and that are *immediately* used by *mosback* or *pnback*.

We will refer to the  $n^{\text{th}}$  chip of MOSM as *MOSM-n*. There are twelve pn CCDs but ESAS does not treat them separately, only as quadrants, which are sets of three CCDs. These are labeled in Figure 1.

In this document when examples of task calls are provided they are broken into multiple lines for clarity. When actually invoking a task all parameters should be on one line (or the lines should be ended with the proper continuation character). In the appendices, long single line commands are broken, and the continuation lines are indented. This is to make it difficult to cut and paste from the cookbook to the command line. A version of the main reduction script where the command lines are unbroken can be found at <https://pages.jh.edu/kkuntz1/run.kdk.txt>.

Many of the commands have input parameters with sensible defaults. For the most part, when giving example commands, we will not list parameters with sensible defaults.

We note, and apologize for, having said the same thing multiple times in different places. We understand that users may dip in and out of the cookbook as they run into problems. We can't count on the reader having read just the right part of the preceding material that is necessary to understand the current material. Thus the repetition!

## 5.5. SAS Reminders

One of the advantages to the user of our work to fully incorporate the ESAS routines into SAS is that standard SAS command options now work. The following text is brazenly stolen from the SAS User's Guide<sup>8</sup> which is well worth browsing.

All SAS tasks support the following command line options:

- h provides information about the command line options
- d launches the GUI for this specific task
- p prints all parameters with their default and current value
- t turns on tracing information from the libraries
- m displays the HTML documentation of the task using a browser
- c no clobber prevents files from being overwritten
- v prints the version number of the task
- V sets the verbosity level of the task to opt. The opt value ranges from 0 to 10 and the larger the verbosity value, the more log messages will be produced. The default value is 1, a value of 4 is recommended for a good verbose control. Larger

values are only recommended for debugging purposes.

## 5.6. Setup

Far be it from us to define a user's directory structure for them, but over the years we have found that it is convenient to have a specific setup. First a main directory is created using the ObsID number of the observation to be processed, e.g., `/path/0123456789` (where `/path` is wherever you want to place the data on your computer). Under the main directory two subdirectories should be created, `/path/0123456789/ODF` for the ODF data files and `/path/0123456789/analysis` for all of the processing output. The ODF data for the observation should be copied to the `/path/0123456789/ODF` directory and uncompressed. The `*.SAS` file should be removed if it exists. SAS should be set up to run in the `/path/0123456789/analysis` directory making sure that the paths are correctly set, e.g.:

```
setenv SAS_CCF /path/0123456789/analysis/ccf.cif
setenv SAS_ODF /path/0123456789/odf
setenv SAS_CCFPATH /ccfpath/CCF
```

It is necessary to point to an existing CIF file or to explicitly run `cifbuild` to create it. An ODF summary file (`*.SAS` file) is also needed. The version of the `*.SAS` file that comes with the ODF will not point to the correct files, and must be replaced. This file is created by running the task `odfingest`. Thus, after removing the `*.SAS` file from the ODF directory, one uses the following two commands:

```
cifbuild withccfpath=no analysisdate=now
category=XMMCCF calindexset=$SAS_CCF fullpath=yes
odfingest odfdir=$SAS_ODF outdir=$SAS_ODF
```

These commands will produce the necessary `ccf.cif` file in the `analysis` directory and the `*SUM.SAS` file in the `ODF` directory. Note that while often it is not necessary to create a new `ccf.cif` file each time ODF data are processed (i.e., it is possible to have a single, up-to-date file for general use), it takes only a little time to run `cifbuild` and you will be assured that you are pointing to the most recent versions of the CCF files (providing you keep the CCF directory properly updated). In addition, you will have a local version of the `ccf.cif` file listing precisely which versions of the CCF files were used for the specific processing in an easily accessible form. This is particularly useful if the same analysis is done on the same data at two different times and the results are different!

<sup>8</sup>[https://xmm-tools.cosmos.esa.int/external/xmm\\_user\\_support/documentation/sas.usg/USG/SASUSG.html](https://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/sas.usg/USG/SASUSG.html)

## 5.7. Initial ESAS Processing

While pipeline processed versions of the event files can be used (after unzipping), it is not overly time consuming (at least compared with the rest of the ESAS processing), and so it is not unreasonable to rerun the chains to ensure that the most recent versions of the CCF and SAS software are used.

There are two ways of recreating the event files: one can use “the chains” *or* one can use “the procs”. The ultimate output of the “procs” will be identical to that of the “chains”, but the “chains” produce many more intermediate files. The “procs” are more memory intensive than the “chains”, but the “chains” require more disk space. The names of the output event files from the “procs” are different from those of the “chains”. The “chains” utilize Perl, so if you have problems with your Perl installation, the “procs” may be more useful.

### 5.7.1. The Chains

The commands are :

```
emchain
epchain
epchain withoutoftime=true
```

The *emchain* task will produce event files for all CCDs for MOS1 and MOS2 for imaging exposures, as well as the outer CCDs, 2 through 7, for timing observations. However, *epchain*, by default, will create event files for *only* the first imaging exposure and must be run explicitly calling out the exposure number if there are multiple exposures (e.g., *epchain exposure=2*) How would you know that there is more than one pn segment? There is, as usual, more than one way.

In the ODF directory, look for files of the form \*\_PN\*IME\*.FIT. There will be one file for each of the 12 CCDs for each segment. In the file name, the segment number (like S001) follows directly after the ‘PN’. An even easier way to determine how many pn segments there are is to type “*epchain exposure=99*”; you will get a list of pn segments:

```
-- EXPOSURE = 99
-- 2 IMAGING exposures in ODF:
-- 1: 0100_0097820101_PNS005 IM
-- 2: 0100_0097820101_PNS013 IM
```

followed by an error message. Here there are two exposure segments, 1 and 2.

The output of *emchain* and *epchain* is prodigious. Of that output you will want to retain the four (or

more) files containing the \*EVLI\*.FIT string<sup>9</sup>. Example file names are given Table 5 in the *em/epchain* column.

We suggest deleting (or saving in their own sub-directory) all of the other .FIT files. For simplicity, for the rest of this cookbook, we will assume that these \*EVLI\*.FIT files have been renamed to *mos1S001.fits*, *mos2S002.fits*, *pnS003.fits*, and *pnS003-oot.fits*. These names are not arbitrary, as several ESAS tasks are still a bit picky about the possible input file names because they use the *EXPIDSTR* parameter in the header to construct file names. See Appendix B for an example of renaming.

One of the common panic-inducing discoveries that inexperienced users make, is that sequential runs of *emchain* and *epchain* do not produce the same results. This is due to both chains introducing a randomization in position (one CCD pixel), time (one frame), and energy (one ADU=5 eV). This randomization converts a distribution of integers into a pseudo-continuous distribution in order to avoid aliasing. It is *sometimes* useful to turn off randomization when debugging.

### 5.7.2. The Procs

The “chains” aren’t the only way to do the initial reduction, one can also use the “procs”:

```
emproc
eprocc
move the output product(s)
eprocc withoutoftime=true
```

Note that *eprocc withoutoftime=true* will overwrite the output from *eprocc*, so be careful to rename files between the two commands. The output event files from the “procs” have different names than those produced by the “chains”. Table 5 shows the correspondence between the two systems.

## 5.8. Examining Some Diagnostics

The first step after *emchain/epchain* is seeing what is in the data. For this purpose we suggest forming some images with which to diagnose potential problems:

```
evselect table=mos1S001.fits withimageset=yes
        imageset=mos1S001-diag-det-unfilt.fits
        filtertype=expression
        expression="(PI in [300:1000])&&
```

<sup>9</sup>For a complete description of the file name structure, see §10.1. For a complete description of the contents of the files see the data files handbook at [http://xmm-tools.cosmos.esa.int/external/xmm\\_user\\_support/documentation/dfhb/](http://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/dfhb/)

Table 5: Event File Names

Detector	em/epchain	em/epproc
MOS1	P0204870101M1S001MIEVLI0000.FIT	0749_0204870101_EMOS1_S001_ImagingEvts.ds
MOS2	P0204870101M2S002MIEVLI0000.FIT	0749_0204870101_EMOS2_S002_ImagingEvts.ds
pn	P0204870101PNS003PIEVLI0000.FIT	0749_0204870101_EPN_S003_ImagingEvts.ds
pn OOT	P0204870101PNS003OOEVLI0000.FIT	0749_0204870101_EPN_S003_ImagingEvts.ds

```

(PATTERN<=12)&&((FLAG & 0x766aa000)==0) "
ignorelegallimits=yes imagebinning=imageSize
xcolumn=DETX ximagesize=780
ximagemax=19500 ximagemin=-19499
ycolumn=DETY yimagesize=780
yimagemax=19500 yimagemin=-19499
evselect table=mos2S002.fits withimageset=yes
imageset=mos2S002-diag-det-unfilt.fits
filtertype=expression
expression="(PI in [300:1000])&&
(PATTERN<=12)&&((FLAG & 0x766aa000)==0) "
ignorelegallimits=yes imagebinning=imageSize
xcolumn=DETX ximagesize=780
ximagemax=19500 ximagemin=-19499
ycolumn=DETY yimagesize=780
yimagemax=19500 yimagemin=-19499
evselect table=pnS003.fits withimageset=yes
imageset=pnS003-diag-det-unfilt.fits
filtertype=expression
expression="(PI in [300:1000])&&
(PATTERN <= 4)&&(#XMMEA_EP)"
ignorelegallimits=yes imagebinning=imageSize
xcolumn=DETX ximagesize=780
ximagemax=19500 ximagemin=-19499
ycolumn=DETY yimagesize=780
yimagemax=19500 yimagemin=-19499
evselect table=pnS003-oot.fits withimageset=yes
imageset=pnS003-oot-diag-det-unfilt.fits
filtertype=expression
expression="(PI in [300:1000])&&
(PATTERN <= 4)&&(#XMMEA_EP)"
ignorelegallimits=yes imagebinning=imageSize
xcolumn=DETX ximagesize=780
ximagemax=19500 ximagemin=-19499
ycolumn=DETY yimagesize=780
yimagemax=19500 yimagemin=-19499

```

This is a large amount of typing, which is why it is a good idea to script much of the reduction.

For the MOS we have used `((FLAG & 0x766aa000)==0)` which is not equivalent to `#XMMEA_EM` but is the equivalent to `#XMMEA_EM` with the addition of the out-of-FOV events. For the pn we use `((FLAG & 0xcfa0000)==0)` which *is* the equivalent of `#XMMEA_EP`, which does include the out-of-FOV events. (The flag for the pn which excludes the corner data would be `((FLAG & 0xfb0000)==0)`.) We have

extracted the standard patterns. We have restricted the images to the soft band, as that is where most of the problems can be detected.

If the circular FOV is easily seen in all the instruments, then there is a significant soft proton flare problem. Lower amounts of soft proton flares may not be readily visible to the eye in these images. If, for the MOS images, some individual CCDs appear brighter than others (typically MOS1-4, MOS1-5, MOS2-2, or MOS2-5) then those CCDs are in their “anomalous states” and may have to be removed from the analysis, *but only if* the analysis includes data from  $E < 1.5$  keV. Comparison of the pn image with the pn-oot image will show whether there are any bright objects that are likely to cause OOT problems in the later analyses.

If the analysis is to include data at higher energies, it would be wise to repeat these diagnostic images at those higher energies. Scattering arcs due to bright sources just outside the field of view tend to be stronger at higher energies, so a quick check at those energies is encouraged.

Given the large number of files created by ESAS, we suggest limiting the number of files in your `analysis` directory; it is a good idea to either delete these diagnostic files after they have been created, or to move them to their own subdirectory.

## 5.9. Examining CCDs for Anomalous States

Some of the individual CCDs in the MOS detectors occasionally operate in anomalous states where the background at  $E < 1$  keV is strongly enhanced (see Kuntz & Snowden 2008 and Figure 1). XMM-ESAS at this time<sup>10</sup> does not adequately handle this situation and so the data must be screened and any affected CCDs excluded from further processing when soft data are of interest. Data above 2 keV are unaffected.

The effects of the anomalous state can usually be seen in the diagnostic plots created in §5.8. As the mission has progressed, those CCDs affected by anomalous states have in general been more frequently af-

<sup>10</sup>Given the time variability of the spectral shape in the anomalous state, it will be difficult to produce a *reliable* background even for the anomalous states that have become quite common. However, data above 2 keV are unaffected so images and spectra of higher energy sources can be useful.



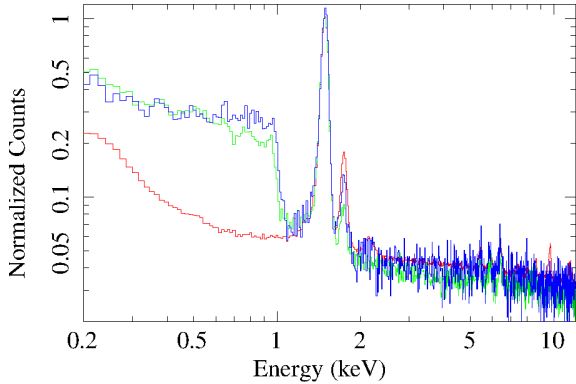


Fig. 6.— MOS1 spectra from FWC data with CCDs in their nominal state (red), CCDs #4 (green) and #5 (blue) in their anomalous states. Note the excess at energies less than 1 keV for the anomalous states. The data have been normalized so that the peaks of the Al  $K\alpha$  lines ( $E \sim 1.5$  keV) are equal to 1.0. The variations in the strength of the Si  $K\alpha$  is due to the relative variations of the line strengths over the area of the detector.

ected. Anomalous state CCDs can be excluded in further processing by an explicit CCD selection input in several of the tasks.

However, if the observation is short, visual screening may not be sufficient. In that case, examination of the diagnostic output from *emanom* will reveal the existence of anomalous states. The *emanom* task calculates the (2.5-5.0 keV)/(0.4-0.8 keV) hardness ratio from the corner data to determine whether a chip is in an anomalous state. However, it should be noted that the “anonymous” anomalous state of MOS1 CCD#4 is not always detectable from the unexposed corner data. Thus, comparing spectra from different CCDs is important for detecting (and removing) this anomalous state.

The *emanom* routine is invoked simply:

```
emanom eventfile=mos1S001.fits keepcorner=no
emanom eventfile=mos2S002.fits keepcorner=no
```

where the inputs are MOS event files. These need not have been filtered for the soft proton flares as the corner data required to determine whether a chip is in an anomalous state are shielded from the soft proton flares. In the default mode, the results are written to the header of the event file. For each chip  $n$  with corner data (chips 2 through 7), the header keywords ANOMHR $n$ , ANOMHEN, and ANOMFL $n$  contain the hardness ratio, uncertainty in the hardness ratio, and the anomalous state flag. The anomalous state flags are as follows:

**G** The hardness ratio indicates that the chip is not in an anomalous state. The QPB spectra con-

structed by ESAS will be an adequate description of the actual QPB spectrum for this chip.

- I** The hardness ratio indicates that the chip is not in its normal state, but the chip is not too deep in an anomalous state. The QPB spectra constructed by ESAS *may* be adequate to describe the actual QPB spectrum for this chip.
- B** The hardness ratio indicates that the chip is in an anomalous state. The QPB spectra constructed by ESAS *will not be* adequate to describe the actual QPB spectrum for this chip.
- O** The chip is not active in this observation. (Thus far, this will only apply to MOS1-3 and MOS1-6 which have become inactive due to micrometeorite strikes.)
- U** The count rate is sufficiently low that there are no counts in the soft band. It would be impossible to determine if the chip were in an anomalous state or not.

The anomalous state information is also printed to an ASCII text file. For the above example the output text files would be `mos1S001-anom.log` and `mos2S002-anom.log`. One has the option to save the corner data used for the anomalous state check, but this is not particularly useful. The lower allowable limits of the (2.5-5.0 keV)/(0.4-0.8 keV) for the “indeterminate” and “good” states are given in Table 6.

### 5.10. Soft Proton Flare Filtering

Soft proton (SP) filtering is accomplished with the *espfilt* task. This task creates two light curves (one from the FOV data, and one from the corner data) in the 2.5-8.5 keV band, and creates an X-ray count rate histogram from the FOV data. For a typical observation the histogram will have a roughly Gaussian peak at some nominal count rate (the count rate during time intervals unaffected, or at least minimally affected, by SP contamination) with a higher count-rate tail. Depending on how contaminated the observation is, the Gaussian peak can be very well defined or just a small bump in the distribution. In the latter case, *espfilt* is unlikely to provide a reasonable result (it may even fail completely) and the observation is probably unusable for the study of extended sources. In the former case, *espfilt* will fit a reasonable Gaussian to the peak and determine thresholds at plus or minus  $1.5\sigma$ . The

Table 6: Anomalous Criteria

Chip	I Lim	G Lim
1-4	2.75	3.00
1-5	2.25	3.60
2-2	2.10	2.50
2-5	2.00	4.00

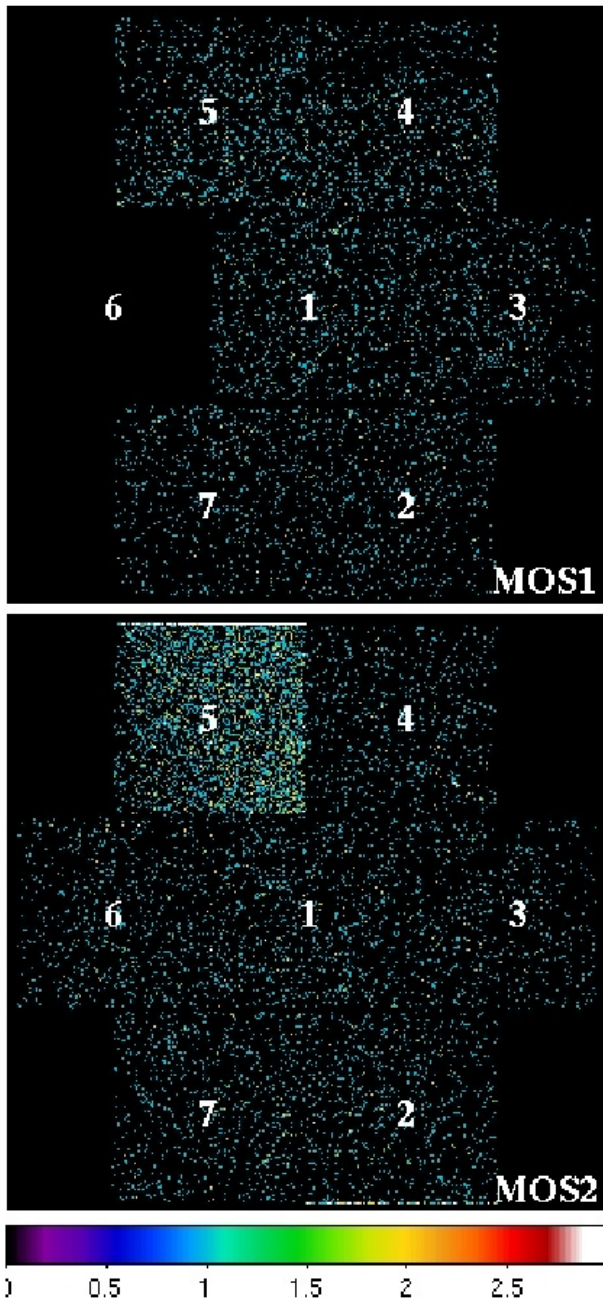


Fig. 7.— MOS1 and MOS2 event images in the 0.2 – 0.9 keV band showing the CCD IDs and two CCDs in anomalous states. The observation (ObsID 0402530201) was performed in 2006 June after the loss of MOS1 CCD#6 due to a meteorite hit (2005 March 9, ~01:30 UT). MOS2 CCD#5 is clearly in an anomalous state, however MOS1 CCD#5 is in a fainter anomalous state as well.

*espfilt* task then creates a GTI file for those time intervals with count rates within the thresholds and uses the task *evselect* to filter the data to create “cleaned” photon event files.

We stress that this process does not necessarily remove *all* of the SP contamination, it only removes time

intervals with obvious contamination, i.e., where the count rate is significantly enhanced over a nominal level. In general this provides a reasonable method for minimizing the SP contamination while still leaving the best, and most of the time sufficient, data to be analyzed. One should check the diagnostic files to see how well the data have been cleaned.

The task input is an event file created with *emchain* or *epchain*. For the histogram method the execution can be simple:

```
espfilt eventfile=mos1S001.fits elow=2500 ehigh=8500
  withsmoothing=yes smooth=51 rangescale=6.0
  allowsigma=3.0 method=histogram
  keepinterfiles=false
espfilt eventfile=mos2S002.fits elow=2500 ehigh=8500
  withsmoothing=yes smooth=51 rangescale=6.0
  allowsigma=3.0 method=histogram
  keepinterfiles=false
espfilt eventfile=pnS003.fits elow=2500 ehigh=8500
  withsmoothing=yes smooth=51 rangescale=15.0
  allowsigma=3.0 method=histogram
  withoot=Y ootfile=pnS003-oot.fits
  keepinterfiles=false
```

The first two of these commands runs *espfilt* on the event files from MOS1 and MOS2. The last runs *espfilt* on the event file from the pn, and then applies the same time selection to the pn-oot event file as well.

Because the pn is more sensitive to the soft proton flares, the pn and MOS filtering may not produce consistent time intervals, that is, the MOS data may include intervals excluded from the pn data. It is up to the user to determine whether this is acceptable. (Yes, the pn may mark some times as being flared, but if the MOS data is not flagged, then the effect on the MOS data will likely be minimal.) Editing GTI files by hand to make the instruments have consistent time selection is currently painful.

There are a number of possible parameters that one can set in order to tweak the way the filtering is done. However, the defaults are the result of a substantial amount of testing, and thus are likely to work well. If there is a bright extended source in the FOV, increasing the *rangescale* to 10 for the MOS and 25 for the pn may be necessary to get a good fit.

The output of each run of *espfilt* is the following list of files:

- P-allevc.fits - the filtered event list for all of the data, both FOV and corner. **This file is used for all following steps.**
- P-allimc.fits - diagnostic image of FOV and corners after filtering
- P-corevc.fits - the filtered event list for the cor-

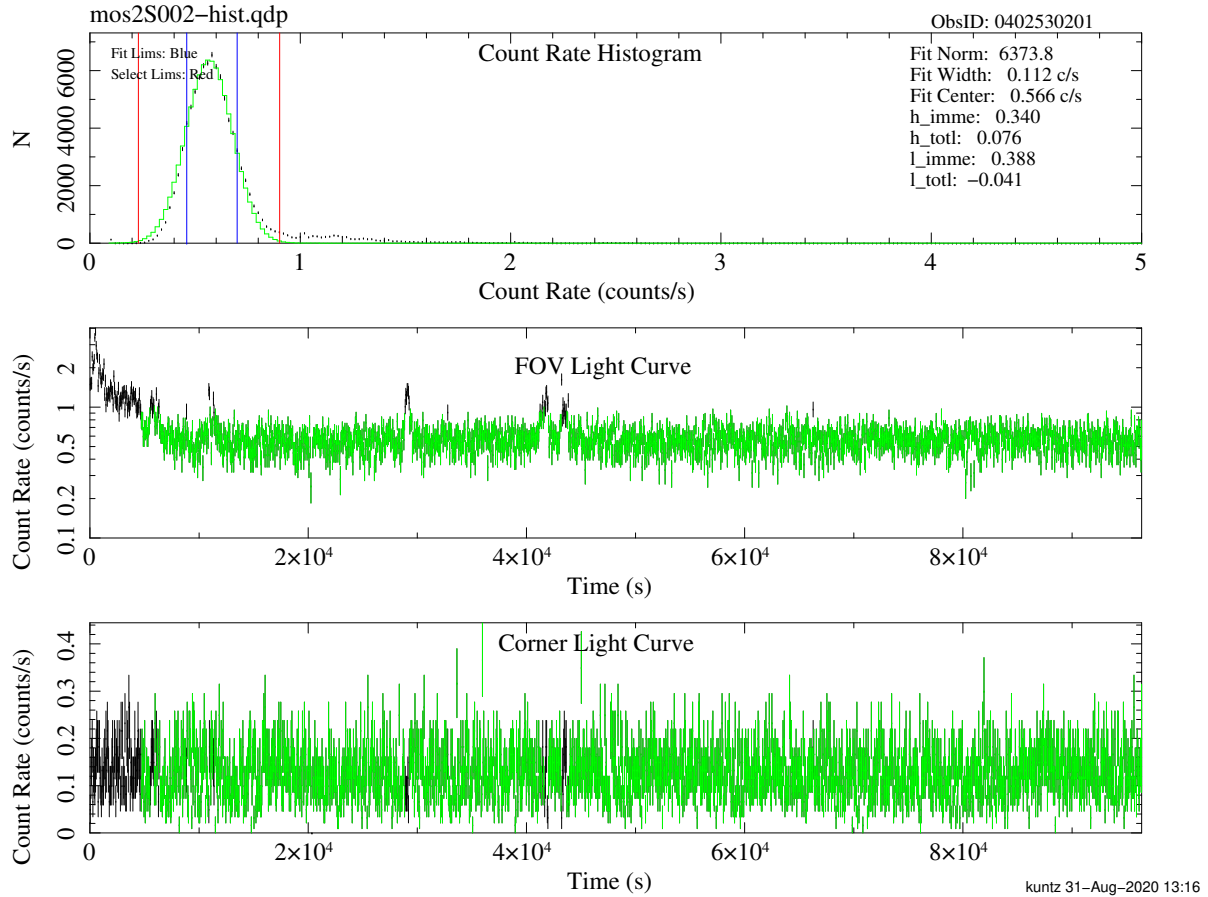


Fig. 8.— Temporal filtering results for the MOS2S002 SWCX-1 exposure with ObsID 0402530201. The upper panel plots the light curve histogram for the 2.5 – 8.5 keV band from the FOV, the middle panel displays the 2.5 – 8.5 keV band FOV light curve, and the lower panel displays 2.5 – 8.5 keV band light curve from the unexposed corners of the instrument. The histogram is derived from the smoothed light curve. In the upper panel, the blue vertical lines show the range for the Gaussian fit, the green curve shows the Gaussian fit, while the red vertical lines show the upper and lower bounds for filtering the data. In the bottom two panels green points indicate accepted data while black points indicate data excluded by the filtering algorithm. The high count rate excursions are produced by soft protons rather than a higher-energy particle background flare as the latter case would produce a mirror increase in the MOS corner data light curve. The pn corner data do show an increase with SP flares because of OOT events.

ner data. This file can be used by *mosspectra/pnspectra*, but we recommend allowing *mosspectra/pnspectra* to create their own corner files, just in case the definition of the corner or the required flags changes in either *espfilt* or *mosspectra/pnspectra*.

P-corimc.fits - diagnostic image of corner data after filtering

P-corlc.fits - the light curve for the corner data

P-fovlc.fits - the light curve for th FOV data

P-gti.fits - the GTI file containing the intervals without soft proton flares

P-hist.qdp

where P is a prefix such as “mos1S001”. If `keepinterfiles=true` then the following files are kept

as well:

P-corev - the unfiltered event list for the corner data

P-corim.fits - diagnostic image of corner data before filtering

P-allim.fits - diagnostic image of FOV before filtering

The QDP files should be plotted and examined to determine whether the exposure is actually useful (see Figures 8 and 10 for examples). Figure 9 shows the plot for the mos1S003 exposure from the Abell 1795 observation. This is a clear example where there is likely to be residual SP contamination, as evidenced by the slight ripple in the nominally constant level of the light curve. While the selection criteria could be tweaked to remove more of the contaminated time in-

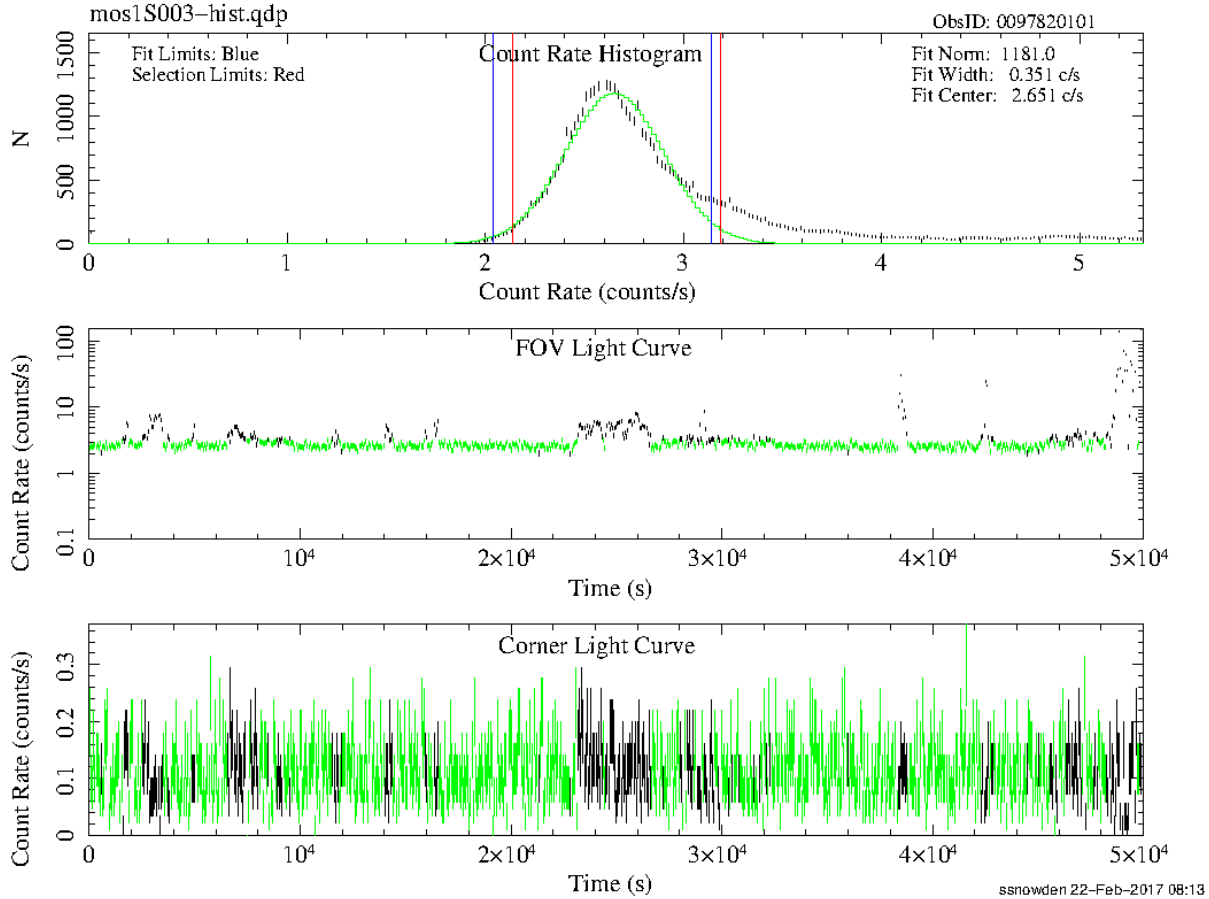


Fig. 9.— Temporal filtering results for the MOS2S002 exposure of Abell 1795. Although the SP flaring is relatively minor, one can still see that some accepted time intervals are likely to contain some residual SP flare contamination. This plot is from an older version of ESAS.

tervals, there would still be no guarantee that there isn't some finite minimum contamination at all times. Figure 10 shows the plot for an observation much more strongly affected by SP flaring where the data are effectively useless for studies of diffuse emission. Unfortunately for the authors, this was one of their observations.

To double check that the soft proton filtering worked well, compare the `P-allimc.fits` and the `P-corimc.fits` images. In the filtered file, the FOV should be indistinguishable from the corners if the SP filtering succeeded.

Of the files created by `espfilt` only `P-allevc.fits` and `P-gti.fits` need be saved for the future steps. The remaining files can be moved to wherever the diagnostic files are kept.

It should be noted that `espfilt` will not run for certain filters or submodes. Observations taken with the “CalClosed” filter are not processed because there is (by definition) no soft proton flaring seen when the filter is closed. `espfilt` will not run for pn observations

taken in the timing or small window modes; there simply isn't enough area from which to determine the light curve of the background. `espfilt` does work for pn observations take in large window mode, even though those observations can't be used for further ESAS processing. `espfilt` will work on all MOS modes.

It should also be noted that `espfilt` silently applies a filter to the `PATTERN` ( $\leq 12$  for the MOS and  $\leq 4$  for the pn) and the `FLAG` (`(FLAG & 0x766ab000)==0`) for the MOS and `#XMMEA_EP` for the pn). If the user wishes to apply a different selection, the `P-gti.fits` file can be applied to whatever event file they desire, as below.

```

evselect table=input_file.fits
withfilteredset=yes filtertype=expression
expression='GTI(P-gti.fits,TIME)'
filteredset=output_file.fits
keepfilteroutput=yes updateexposure=yes
filterexposure=yes

```

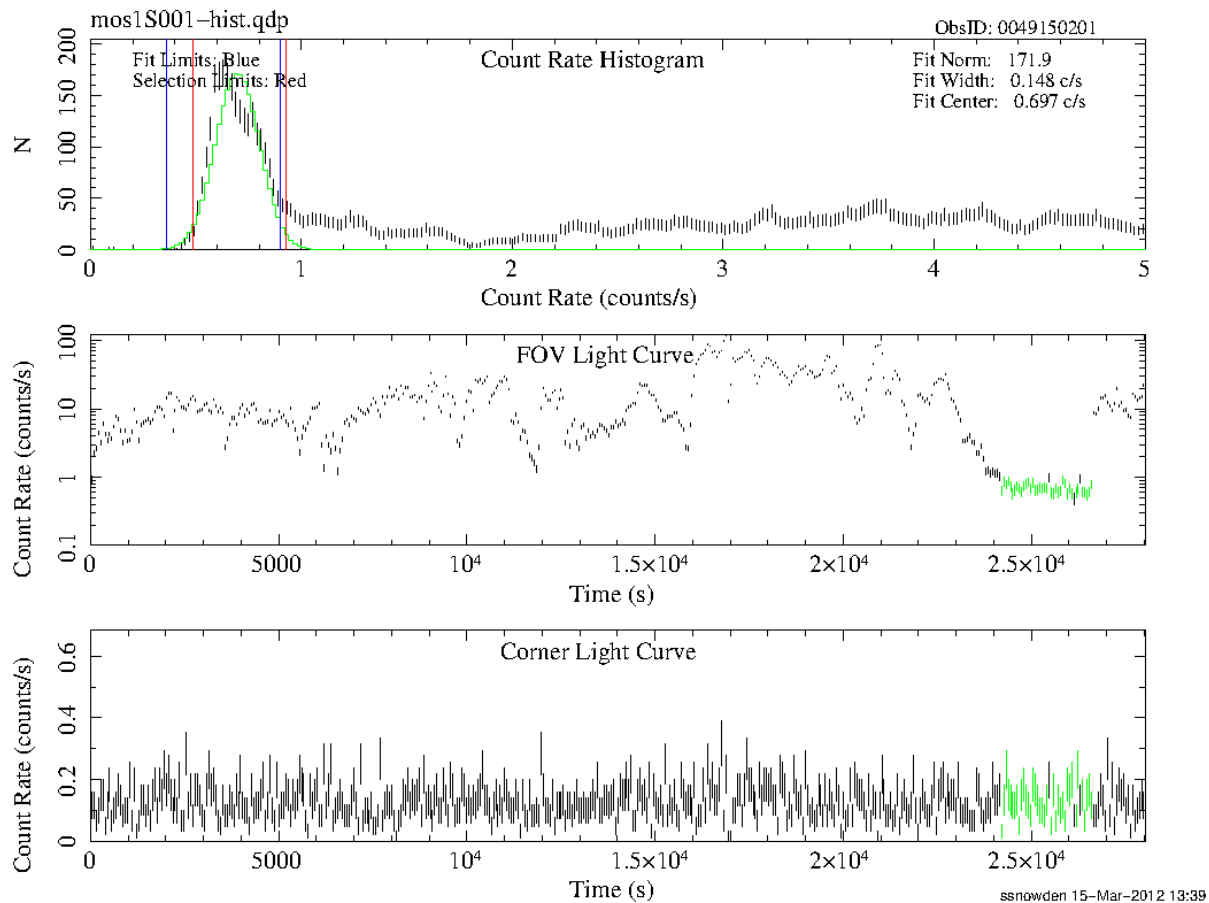


Fig. 10.— Temporal filtering results for the MOS1 Magellanic Bridge observation (0049150201). The SP flaring is so strong and affects so much of the observation that the data were not useful for the study of the diffuse emission in the field. Even the roughly 2.5 ks of low count rate exposure are likely to be still contaminated with residual soft protons. This plot is from an older version of ESAS.

The numbers listed in the histogram plot can be used for a reasonable (but not spectacularly good) automated determination of goodness of soft proton filtering.

### 5.11. Merging

Each instrument may have multiple observation segments. It is generally a good idea to merge the segments for a given instrument. However, merging should not be done until after a first examination of each segment with *emanom* and *espfilt*.

Anomalous states *can* change between segments, though that is probably not likely. Thus, one should check the anomalous states for each observation segment. Of course, some segments may be too short to produce a good measure of the anomalous states, so it should *usually* be assumed that the anomalous states have not changed between segments.

Since observations are often segmented due to particle background rates exceeding the threshold for a

short period of time, soft proton flaring will often be different between segments, while the cosmic X-ray background should not be changing between segments. It may be clear from *espfilt* output for individual segment that some segments are flared beyond use. There may be segments that are too short to allow one to determine a reliable base level. In this case it is often better to merge the segments, and apply *espfilt* to the result.

In any case, the segments that are to be used should be merged before the following step.

### 5.12. Detecting and Excising of Point Sources

If you are using ESAS, then you are probably interested in the diffuse emission. In that case, it makes sense to identify and remove the point sources. Granted, most point sources have power law spectra that are easily distinguished from the spectrum of the diffuse emission. However, the point sources are contributing counts that are not of interest and increase

the noise level. Thus, it is useful to remove them entirely at an early stage of the analysis.

The XMM-ESAS *cheese* task runs *edetect\_chain* to create source lists and masks (see Figure 12) for use in excising sources from spectra and images. In general for the analysis of diffuse emission it is desirable to remove the contribution of point sources in the field to a uniform threshold. The *cheese* task will combine both MOS and pn data for the source detection creating images and exposure maps in a selected single band or selected soft, hard, and total (combined) bands. It will create source lists that then can be used to create source-excluded spectra and make “Swiss-cheese” masks for image processing.

The *cheese* routine was formerly only a script that was intended to call standard SAS tasks that create the images required to run *edetect\_chain*, to call *edetect\_chain* itself, to call the SAS tasks that create region files, and then to call the ESAS program *make\_mask*. The current *cheese* routine is a SAS task that does much the same thing. The advantage of using *cheese* is that it produces a reasonable point source mask without much parameter fiddling, and without requiring much interaction with the source detection routine itself. The disadvantage of *cheese* is that it is not that flexible. As soon as you have, for example, wholly overlapping observations, you will want follow what *cheese* does, but without using *cheese*, as discussed below. The functionality of *cheese* can be reproduced with your own script, with whatever modifications that you need. The following will both demonstrate *cheese* and what you need to do to create *cheese*-like outputs for use by other ESAS routines.

The call for *cheese* is extensive, e.g.:

```
cheese mos1file='mos1S001-allevc.fits'
      mos2file='mos2S002-allevc.fits'
      pnfile='pnS003-allevc.fits'
      pnootfile='pnS003-allevcoot.fits'
      elowlist=350 ehighlist=1100 scale=0.4
      mlmin=15 ratetotal=0.2 dist=50.
      keepinterfiles=no
```

for a single band, or

```
cheese mos1file='mos1S001-allevc.fits'
      mos2file='mos2S002-allevc.fits'
      pnfile='pnS003-allevc.fits'
      pnootfile='pnS003-allevcoot.fits'
      elowlist='350 2500' ehighlist='1100 8000'
      scale=0.4 mlmin=15 dist=50.
      ratesoft=0.2 ratehard=0.2 ratetotal=0.2
      keepinterfiles=no
```

for multiple bands. *The cheese routine can be*

*run on a maximum of one event file per instrument, and a maximum of two energy bands. Note well that cheese will accept only one MOS1 observation, one MOS2 observation, and one pn observation.* This constraint is placed by the underlying *edetect\_chain* routine. Thus, if you have multiple observation segments for each instrument, merge all the segments for each instrument before submitting to *cheese*. See §10.2. The limitation in bands per instrument is placed by *cheese*. It could be increased in future if there were a pressing need.

If data from one of the instruments is missing, or needs to be excluded, just don't set the parameter for that instrument's file name. The `scale=0.5` is the PSF threshold to which the point sources are masked, and is in terms of the fraction of the local background (i.e., a value of 0.5 means that the point source is removed down to a level where the surface brightness of the point source is one half of the surrounding background). The `ratesoft=0.2 ratehard=0.2 ratetotal=0.2` parameters are the point-source flux thresholds in units of  $10^{-14}$  ergs  $\text{cm}^{-2}$   $\text{s}^{-1}$ . The `dist=50.0` is the minimum separation for point sources in arc seconds (this avoids the excision of large areas of bright extended sources like the inner regions of clusters of galaxies). However, the `dist` parameter may need to be “tuned” to get the desired result. The `elowlist=400` and `ehighlist=1100` (for one band) and `elowlist='400 2000'` and `ehighlist='1100 7200'` (for two bands) are the energy band limits in eV for the source detection. The energy conversion factors used by *edetect\_chain* are only approximate, so the flux limits specified are only approximate as well.

The output files from *cheese* are also rather extensive:

`atthk.fits` - The SAS attitude file (if not already constructed).

`eboxlist_l.fits` - The output from the first pass of *ebordetect*.

`eboxlist_m.fits` - The output from the second pass of *ebordetect*. As with the previous file, this file is no longer of use once *emllist.fits* is produced.

`emllist.fits` - The final source list produced by *emldetect*. This file will be used by *region* within *cheese*. Once satisfactory region files have been produced, this file is no longer useful, and can be stored with the diagnostic files.

`P-bkregBdet.fits` - The .fits format region file in detector coordinates produced by *region* within *cheese* in band B. This file will be used by *mosspectra* and *pnspectra* to remove the point sources from the extracted spectra.

`P-bkregBsky.fits` - The .fits format region file in sky coordinates produced by *region* within *cheese* in

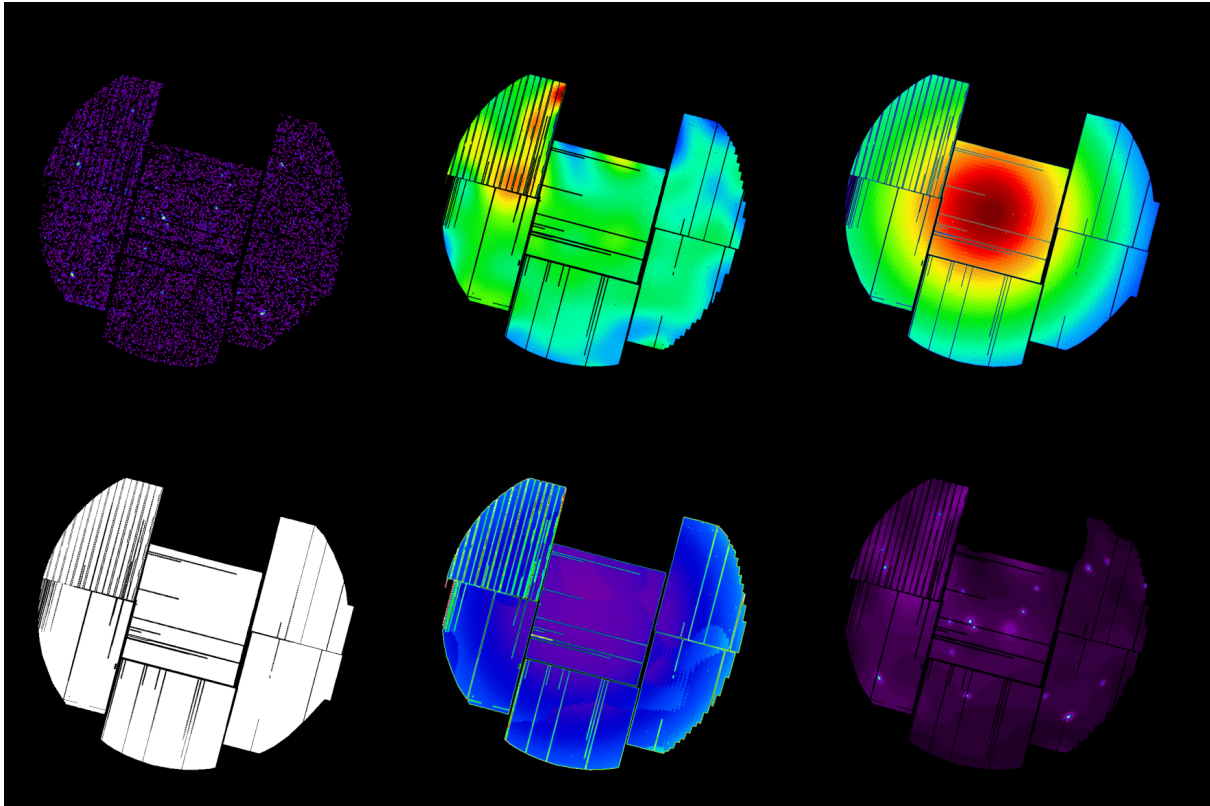


Fig. 11.— Intermediate files produced by *cheese*. *Top Left*: the band image. *Top Middle*: the background image create by *esplinemap*. *Top Right*: the exposure map. *Bottom Left*: the mask showing regions to be searched for sources. *Bottom Middle*: the sensitivity map. *Bottom Right*: the source map produced by *eboxdetect*. Since there is one CCD missing, this is clearly MOS1 from the middle part of the mission.

band B. This file will be used by *mosspectra* and *pnspectra*.

**P-cheeseB.fits** - The point source mask in sky coordinates produced by *make\_mask* for band B.

**P-fovimb.fits** - An image of the FOV in band B. This image is diagnostic, to be compared to **P-cheeseB.fits** to ensure that all of the appropriate sources are removed.

where P is a prefix such as “mos1S001”. If *cheese* was invoked for a single band then B will be *t*, else there will be bands *s* for the softer and *h* for the harder. There are a number of diagnostic files which will be silently deleted if `keepinterfiles=no`. These are

**P-fovimbkg.fits** - A map of the background in band B made by *esplinemap* in *edetect\_chain*.

**P-fovimbexp.fits** - An exposure map (effective area map) for the FOV in band B made by *eexpmap* in *edetect\_chain*.

**P-fovimbmask.fits** - An exposure mask for the FOV in band B made by *emask* in *edetect\_chain*. This mask sets the region that will be searched for sources.

**P-fovimbSen.fits** - A map of the sensitivity for the FOV in band B produced by *esensmap* in *emldetect*.

**P-fovimbSrc.fits** - The source image maps in band B created by *eboxdetect*.

Now, it will be noted by even the least observant that we build regions in both detector and sky coordinates. This might seem duplicative. The problem is that the *arngen* and *rmfgen* routines require selection criteria in detector coordinates. However, for comparison with images or source catalogues, one generally needs sky coordinates. Thus, we create region lists in both coordinate systems. The final mask to be applied to images (the *cheese* mask) is created solely in sky coordinates.

#### 5.12.1. Missed Sources

Occasionally the source detection algorithms will miss an obvious source. This can happen when the source lies, for instance, in the gap between CCDs on one of the detectors. Because of this effect we recommended the simple test of using the *blink* function in *ds9* to flip between the *cheese* image and the event sky

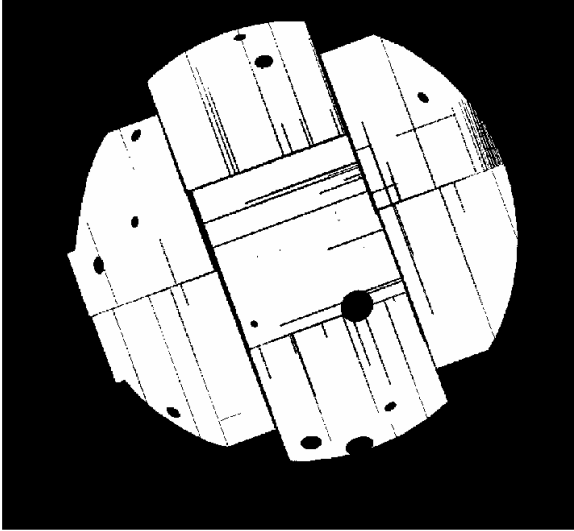


Fig. 12.— MOS1 cheese image from the Abell 1795 observation.

image for each detector.

Missed sources may often be “recaptured” by modifying the `mlmin` or `dist` parameters. However, it is usually a good idea to examine the `emllist.fits` file closely to determine why a source has been missed before rerunning *cheese*. As seen below, one can modify the `emllist.fits` file and run a subset of the *cheese* routine to iterate quickly to a better solution.

### 5.12.2. External Catalogues

There are a number of circumstances in which one would like to use a different or external source list to generate the masks. This can be done in a relatively straightforward manner by modifying the external source list so that it has the same format as the `emllist.fits` file. Most of the columns do not matter for the purposes of making a mask, but the **RA**, **Dec**, **RADEC\_ERR**, should all be populated with the correct (or at least reasonable) values. The **DIST\_NN** column should be set to a large number (like 1000) so that the *make\_mask* routine will not ignore sources. Then, for each observation:

```
region eventset=mos1S001-allevc.fits
  operationstyle=global
  srclisttab=extern_emllist.fits:SRCLIST
  expression="(ID_INST == 2)&&(DET_ML >= 100)"
  bkgregionset=mos1S001-bkregtdet.fits
  energyfraction=0.4 radiusstyle=contour
  outunit=detxy verbosity=1
region eventset=mos1S001-allevc.fits
  operationstyle=global
  srclisttab=extern_emllist.fits:SRCLIST
  expression="(ID_INST == 2)&&(DET_ML >= 100)"
```

```
bkgregionset=mos1S001-bkregtsky.fits
energyfraction=0.4 radiusstyle=contour
outunit=xy verbosity=1
makemask imagefile=mos1S001-fovimt.fits
maskfile=mos1S001-fovimtmask.fits
regionfile=mos1S001-bkregtsky.fits
cheesefile=mos1S001-stinky-cheeset.fits .
```

This sequence reproduces the actions of *cheese* for the user-modified source list `extern_emllist.fits`. The output *cheese* mask, `mos1S001-stinky-cheeset.fits`, can replace the *cheese* output `mos1S001-cheeset.fits` and used in standard ESAS processing or used as an explicitly added mask during image creation<sup>11</sup>.

### 5.12.3. Rolling Your Own

As mentioned above, *cheese* was originally a Perl script that did the repetitive setup for *edetect\_chain*, and then take the output from that and uses *region* and *make\_mask* to produce the source masks. Now, *cheese* is a FORTRAN program that does the same thing. This section steps through the commands that one would need to use if, for some reason, *cheese* cannot be run. Similarly, one may have a different method for finding point sources. In that case, so long as one can create a correctly formatted file like the `emllist.fits` file, one can use it with SAS and ESAS.

*edetect\_chain* can handle up to six bands per instrument, but the bands must be the same among the different instruments. For each band of each instrument one needs to extract the image. In the following example, we extract a soft image from MOS1.

```
evselect table=mos1S001-allevc.fits
  withfilteredset=yes filtertype=expression
  keepfilteroutput=false
  expression='(PATTERN<=12)&&(FLAG == 0)&&
  (PI in [350:1100])'
  imagebinning=imageSize imagedatatype=Int32
  squarepixels=yes ignorelegallimits=yes
  withxranges=yes withyranges=yes
  xcolumn=X ximagesize=900
  ximagemax=48400 ximagemin=3401
  ycolumn=Y yimagesize=900
  yimagemax=48400 yimagemin=3401
  updateexposure=yes filterexposure=yes
  verbosity=5 imageset=mos1S001-fovims.fits
```

Many of these parameters are required so that

<sup>11</sup>The instrument ID set by *edetect\_chain* and used in the `emllist.fits` file is 0 for the summary, 1 for the PN, 2 for MOS1, and 3 for MOS2.



the images from different bands and different instruments will align correctly for use in *edetect\_chain*. Executing this for multiple bands of multiple instruments is tedious and best put into a script, like *cheese*.

Since *edetect\_chain* requires the attitude history file, it must be created if it does not already exist. This can be done simply:

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

The setup for *edetect\_chain* is quite long and, in recognition of fat fingers and tiny typos, should not be attempted directly on the command line. Here is a relatively simple example:

```
edetect_chain imagesets='mos1S001-fovims.fits
mos1S001-fovimh.fits mos2S002-fovims.fits
mos2S002-fovimh.fits pnS003-fovims.fits
pnS003-fovimh.fits'
eventsets='mos1S001-allevc.fits
mos2S002-allevc.fits pnS003-allevc.fits'
esp_ooteventset='pnS003-allevc-oot.fits'
attitudeset=atthk.fits
pimin='350 2500 350 2500 350 2500'
pimax='1100 5000 1100 5000 1100 5000'
ecflist='1.2 1.2 1.2 1.2 3.2 3.2'
witheexpmap=yes likemin=10
eboxl.likemin=15eboxm.likemin=15
ebox.withdetmask=yes
ebox.withexpimage=yes esp_scut=0.01
esp_withootset=yes esp_nsplinenodes=20
eml_ecut=15 eml_scut=15
eml_withtwostage=yes
eml_withthreshold=yes eml_threshold=15
esen_mlmin=10 eml_list=emllist.fits
```

There are other parameters that may need to be tweaked as well. Note also that there are a large number of files created by *edetect\_chain*.

Users will generally want to examine the *emllist.fits* file to determine what cuts best remove point sources for the scientific program at hand. This process can be quite subjective! For each source there are multiple entries in the *emllist.fits* table, all of which have the same *ID\_CLUSTER* value. There will be an entry for each band/instrument combination, as well as one entry that is a summary (*ID\_INST=0*). The *region* routine allows one to filter the *emllist.fits* by the values in its columns to determine which sources are excluded or not. In general, one would want to set a limiting flux (*FLUX*) and a limiting probability (*DET\_ML*). One might also want to set the condition that the same sources be excluded from all instruments. This latter

condition suggests that the summary entry be used. However, if a source does not fall into the FOV of every instrument, the flux for the summary entry may be listed as NULL, meaning that it would not be considered a real source for any filtering that relies on the *FLUX* column. Thus, it is important to look at the data in the *emllist.fits* file carefully before running *region*.

The next step is to create region files to exclude or excise the point sources. This is also done on an instrument by instrument basis. Again using *mos1S001* as our example:

```
region eventset=mos1S001-allevc.fits
operationstyle=global
srclisttab=emllist.fits:SRCLIST
expression='(DIST_MN >=25)
&&(FLUX>=1e8)&&(ID_INST == 1)
&&(DET_ML >= 15)&&(ID_BAND == 1)'
bkgregionset=mos1S001-bkregtsky.fits
bkgfraction=0.4 radiusstyle=contour
nosrcellipse=no outunit=detxy
```

Here the filtering uses a plethora of conditions, mostly to demonstrate the variables most likely to be used. Remember that *region* should also be run in detector coordinates: *outunit=detxy* *bkgregionset=mos1S001-bkregsdet.fits*.

The final step is to run *makemask* as was shown in the example in the previous section.

Often, the users objection to the output from *cheese* lies in the way the *emllist.fits* file was filtered by *region*. It is generally more efficient to run *cheese*, modify the *emllist.fits* file, and then rerun the *region* and *makemask* steps than it is to run all the constituent steps of *cheese*.

#### 5.12.4. A Middle Way

The default *cheese* is unlikely to produce *exactly* what you want. For many applications one would like the same sources removed from each detector. Due to the way the *emllist.fits* file is constructed, and the way *region* accesses it, this is not a trivial task to program in an automated way. However, it can be done with a small amount of hand editing of the *emllist.fits* file, if you are willing to learn the details of the *emllist.fits* file and how it is used by *region*.

From inspection of the images, determine the *FLUX* or *DET\_ML* limit you wish to impose. Then remove all of the sources that do not meet that criteria for the combination of instruments. Keep in mind that a source may have different *DET\_ML* in different instruments, and so may get excluded from some instru-

ments but not others. (This can be fixed by editing the `emllist.fits` file or by using `emlfill`.) Further, if there is a bright source in a chip gap, you will want to excise it, but it may not have an entry for that detector. In that case, you may need to add a line to the `emllist.fits` file, making sure that the `SCTS` column is appropriate. And it should be noted that the fluxes for the summary lines will be NAN if the source falls out of the FOV (or on a CCD gap) for any of the detectors.

#### 5.12.5. *And if the Source is Really Big...*

Let us assume, for the moment, that you are interested in the diffuse cosmic X-ray background from an observation that has an extended source in the middle of it that you want to remove. For example, a galaxy at a certain  $(\alpha, \delta)$  with semi-major axes  $(a, b)$  and a position angle  $P$  (as usual, measured counter clockwise from North). You will need to convert those quantities into detector coordinates. The  $(\alpha, \delta)$  can be converted to  $(detx, dety)$  with `esky2det`. The  $(a, b)$  can be converted to detector coordinates by scaling to  $0''.05$  pixels. The position angle is a bit trickier as that will depend upon the detector. The spacecraft “up” for which the roll is measured is the +Y axis (detector coordinates) for MOS1, the +X axis for MOS2, and the -X axis for the pn. This means that the rotation for the region needs to be:

**MOS1:** roll angle -  $P + 90.0$  degrees

**MOS2:** roll angle -  $P$  degrees

**pn:** roll angle -  $P + 180.0$  degrees

Converting a region from sky to detector coordinates can also be done with `convregion`. However, `convregion` does not yet have the full functionality that may be required for more complex regions (i.e., regions other than circles and ellipses) and `convregion` does not calculate the rotation.

### 5.13. Creation of Model Particle Background Spectra and Images

The creation of the quiescent particle background (QPB) spectra is a two step process requiring `mosspectra` and `mosback` or `pnspectra` and `pnback`. The first step can be rather time consuming (though we have implemented a new way of making the process faster) while the second step runs quite quickly.

The `mosspectra` and `pnspectra` tasks start the process of implementing the Kuntz & Snowden (2008) particle background calibration. They extract QPB spectral information from three sources: the corner spectra from each chip of the observation being analyzed, the corner spectra from each chip of the FWC data, and the FOV spectra from each chip of the FWC data. The

FWC data are derived from observations where the filter wheel is in the closed position, blocking the detection of any X-rays from outside the instrument and the penetration of any soft protons<sup>12</sup>. Cosmic rays, however, still penetrate to the detectors so a clean measure of the QPB is allowed. The FWC data provide the spatial distribution of the QPB events. The corner data provide a measure of the both normalization of the QPB rate and the shape of the QPB spectrum, which was found by Kuntz & Snowden (2008) to be temporally variable<sup>13</sup>.

While `mosspectra` and `pnspectra` extract these data, `mosback` and `pnback` combine all of these different pieces into the QPB spectrum for a specified region in the FOV for the observation and an image of the QPB component in a given band. Because the count rate in the corner data is quite low, the corner spectrum is *augmented* from a database of corner data extracted from the public archive. (See Kuntz & Snowden, 2008 for a full explanation.) This augmentation is done by `mosback` and `pnback`.

*Spectra:* Figure 13 (upper panel) displays the observed spectrum and model QPB spectrum from the MOS1 observation of Abell 1795 from the full field of view. The observed spectrum clearly dominates over the background over the entire energy range where the MOS has significant response (i.e.,  $< 8.0$  keV). In spite of the fact that the Abell 1795 cluster is relatively hot and bright, the excess of the observed spectrum over the model QPB spectrum at energies above  $\sim 10$  keV is not cosmic in origin. The excess is most likely due to residual SP contamination, as suggested by the low-level variation in the accepted time intervals of the light curve (the green part of the curves in Figure 9).

Figure 13 (lower panel) displays the FWC spectrum from the MOS1 instrument due to the QPB. The most striking features are the low-energy tail, the strong Al  $K\alpha$  and Si  $K\alpha$  fluorescent lines near 1.49 keV and 1.75 keV, and several other fluorescent lines at higher energies. The Al and Si lines are the only problematic parts of the spectrum above 0.3 keV. Because these lines are so strong, any errors or slight changes in the intrinsic line spread function will show up as significant residuals. Since the FWC data is collected over the entire mission, and the line spread function has changed over that time, the FWC data do not provide a usable template for the background at the Al  $K\alpha$  and Si  $K\alpha$  energies. Small gain variations can also produce

<sup>12</sup>This is a more complicated issue for the pn, but that will be addressed later or elsewhere.

<sup>13</sup>Our understanding has changed since 2008, and while we would maintain that the QPB *is* time-variable, our understanding of what we were seeing in 2008, and our understanding of the source of the variation, has changed substantially. These changes will be reflected in the next ESAS release.

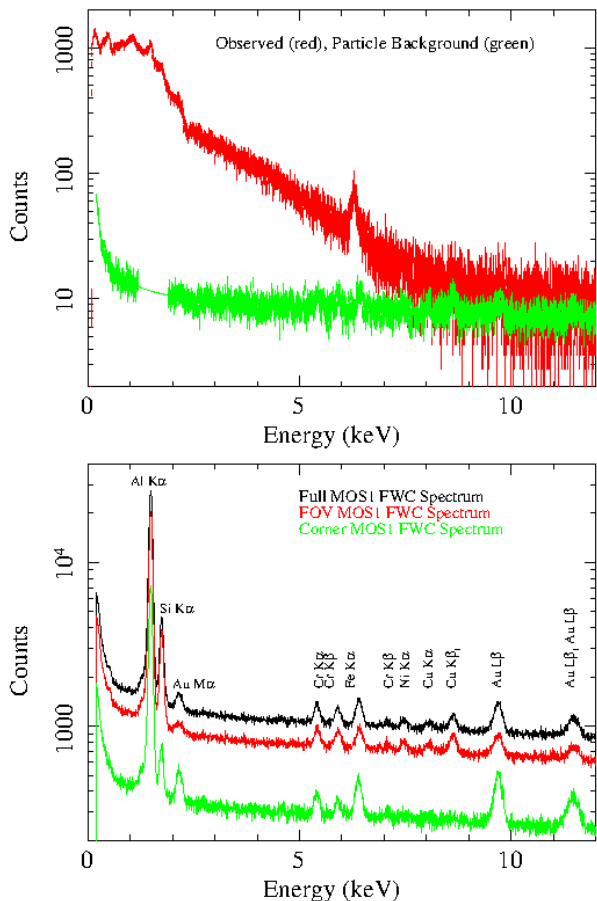


Fig. 13.— *Upper Panel:* Observed spectrum (red) and model QPB spectrum (green) for the full field of view for the MOS1 observation of the Abell 1795 cluster observation from ObsID 0097820101. Note the discrepancy at high energies where the model QPB is still low compared to the observed spectrum even though there is essentially no effective area for detecting cosmic X-rays. This is another indication of residual SP contamination as first suggested by the low-level variations in the light curve for the nominally low intensities. Note the emission lines including the strong Fe line at  $\sim 6.5$  keV in the observed spectrum of Abel 1795. The smooth part of the model QPB at  $\sim 1.5$  keV is the “bridge” where the Al  $K\alpha$  and Si  $K\alpha$  fluorescent instrumental lines affect the data. *Lower Panel:* The complete FWC spectrum from the MOS1 instrument showing the very strong Al  $K\alpha$  and Si  $K\alpha$  fluorescent instrumental lines ( $\sim 1.49$  keV and  $\sim 1.75$  keV, respectively) on top of the continuum. Other fluorescent lines are visible at higher energies as well as a strong low-energy tail due to detector noise.

very striking residuals in spectral fits where the surface brightness of the object of interest is typically smaller than the strength of the fluorescent lines. This has not been observed to be a problem for the higher energy lines. To circumvent this problem, a smooth “bridge” is fit to the data on either side of the Al  $K\alpha$  and Si  $K\alpha$  lines and the continuum component interpolated. This “bridge” is the smooth part of the model QPB shown in Figure 13. As discussed below in § 6, the Al  $K\alpha$  and Si  $K\alpha$  lines must then modeled as separate Gaussians in spectral fits.

Below 0.3 keV, the QPB rises rapidly. Thus we do not recommend attempting spectral fitting below 0.3 keV.

Figure 14 (upper panel) displays the observed spectrum and model QPB spectrum from the pn observation of the cluster Abell 1795. Similar to Figure 13, the cluster spectrum dominates at low energies. However, unlike the MOS spectrum, the pn spectrum has several very strong instrumental Cu, Cr, and Ni lines in the 7.2–9.0 keV band and is missing the Si  $K\alpha$  line at 1.75 keV. With such strong Cu lines it may be reasonable to simply mask those data from the spectral analysis. Similar to MOS spectral fits, the Al  $K\alpha$  and Cu, Cr, and Ni lines in the 7.2–10.0 keV range must be explicitly fit. Doing so, spectral data for hard sources can be used up to  $\sim 11$  keV.

As noted previously, the lower energy limit depends upon the pattern selection. Selecting only PATTERN==0 allows one to reduce the QPB contribution at low energies and one can extract a spectrum down to 0.3 keV to reasonable confidence. Selecting PATTERN==0 has the consequence of significantly reducing the count rate at higher energies. Selecting PATTERN<=4 provides a higher count rate at higher energies, but produces a strong QPB signal at low energies. With PATTERN<=4 it would be unwise to extract the spectrum below  $\sim 0.55$  keV.

*Imaging:* Figure 15 shows typical output images for a MOS exposure: the total counts in the FOV (P-fovmsky-elo-ehi.fits, made by *mosspectra*), the exposure map (P-expmsky-elo-ehi.fits, made by *mosspectra*), the quiescent particle background image (P-bkgimdet-elo-ehi.fits, made by *mosback*) and the source mask (P-cheeset.fits, made by *cheese*). The residual soft proton image, made by *proton*, and the solar wind charge exchange image, made by *swcx*, will be shown in §7.

### 5.13.1. Extraction OR mos/pnspectra

For each individual chip (MOS) or quadrant (pn) the *mosspectra* or *pnspectra* routine extracts the spectra from *both* the user specified region of interest (which may be the full FOV) *and* the corner, and does

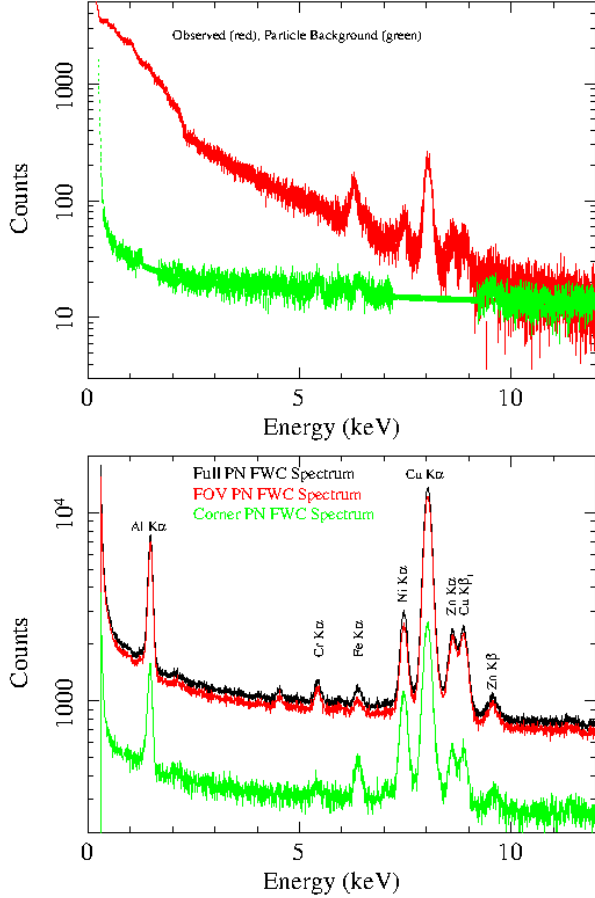


Fig. 14.— *Upper Panel:* Observed spectrum (red) and model QPB spectrum (green) for the full field of view for the pn observation of the Abell 1795 cluster. Note that besides the strong Fe emission line at  $\sim 6.5$  keV there are strong instrumental Cu, Cr, and Ni lines in the observed spectrum. The smooth parts of the model QPB at  $\sim 1.5$  keV and 8 keV are the bridges where the Al  $K\alpha$  and higher energy (note that the pn is not affected by a Si  $K\alpha$  line) fluorescent instrumental lines strongly affect the data. *Lower Panel:* The complete filter wheel closed spectrum from the pn instrument showing the very strong Al  $K\alpha$  and Cu, Cr, and Ni fluorescent instrumental lines ( $\sim 1.49$  keV and  $\sim 8$  keV, respectively) on top of the continuum. Other fluorescent lines are visible at higher energies as well as a strong low-energy tail due to detector noise.

so for *both* the observation *and* the FWC data. These four spectra, extracted from each chip, are required to construct the QPB spectrum of the region of interest; that construction is done by *mosback* and *pnback*.

It should be noted that *mosspectra* and *pnspectra* extracts spectra and, if the *withimages* parameter set, images as well. The energy band parameters, *elow* and *ehigh*, specify the energy band of the *images*; in all cases the spectra are extracted for the entire *XMM-Newton* band.

There are several different layers of event filtering done within *mosspectra* and *pnspectra*. First, the `PATTERN<=12` and `((FLAG & 0x766aa000) == 0)` filters are applied for the MOS, and the `PATTERN` filter (user selected to be either 0 or  $\leq 4$ ) and `((FLAG & 0xcfa0000) == 0)` filter is applied for the pn. Second, the user specified energy selection is applied for images. Third, if the `P-bkgregBdet.fits` file exists, events in the point source regions are excluded. Finally, if the file indicated by the `region` parameter exists, the events are further filtered by region. Of course, all of these selections are done simultaneously, with a rather complex selection expression. If the file indicated by the `region` parameter does not exist, or is empty, then the spectrum and images will be extracted from the entire FOV.

The calls to *mosspectra* and *pnspectra* can be simple, relying on default file names, for example:

```
mosspectra eventfile=mos1S001-allevc.fits
withsrcrem=yes maskdet=mos1S001-bkregtdet.fits
masksky=mos1S001-bkregtsky.fits
withregion=yes regionfile=regmos1.txt
keepinterfiles=yes pattern=12
withimages=yes
elow=350 ehight=1100 ccds="T T T F F F T"
```

or the input/output file names can be more expressly specified:

```
mosspectra eventfile=mos1S001-allevc.fits
withsrcrem=yes maskdet=mos1S001-bkregtdet.fits
masksky=mos1S001-bkregtsky.fits
cornerfile=mos1S001-corevc.fits
imagefile=mos1S001-fovimt.fits
expmap=mos1S001-expimt.fits
spmask=mos1S001-fovspdet.fits
specfile=mos1S001-fovpt.pi
rmfile=mos1S001.rmf arffile=mos1S001.arf
withregion=yes regionfile=regmos1.txt
pattern=12 keepinterfiles=yes
withimages=yes
elow=350 ehight=1100 ccds="T T T F T F T"
```

For the pn

```
pnspectra eventfile=pnS003-allevc.fits
ootvtf=pnS003-allevc-oot.fits
withsrcrem=yes maskdet=pnS003-bkregtdet.fits
masksky=pnS003-bkregtsky.fits
withregion=yes regionfile=regpn.txt
keepinterfiles=yes pattern=0
withimages=yes
elow=350 ehight=1100 quads="T T T T"
```

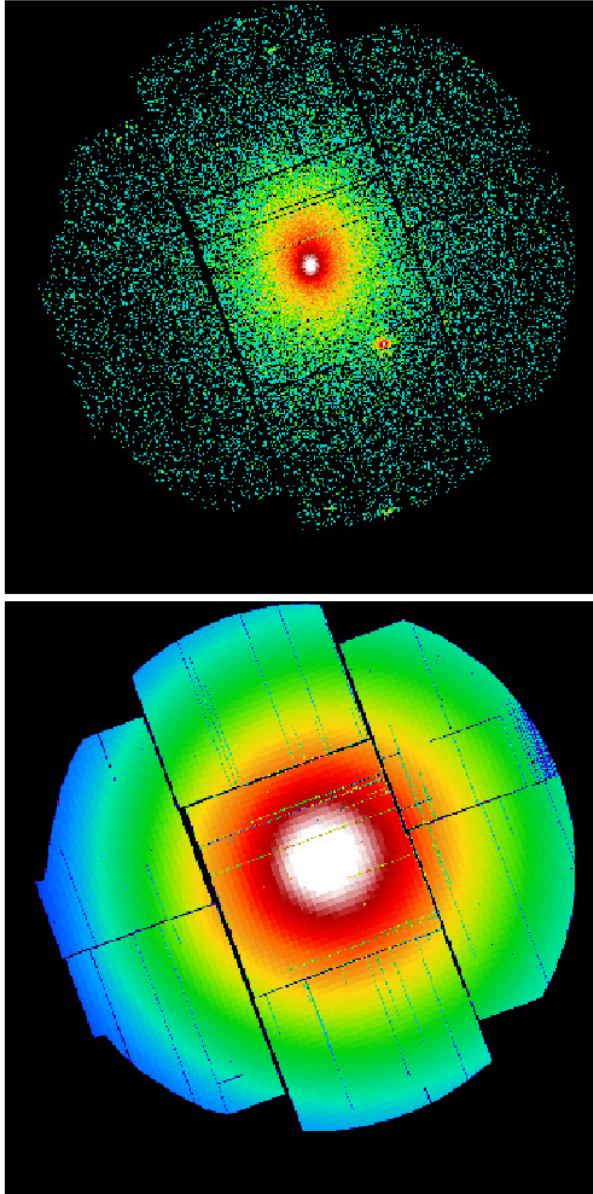


Fig. 15.— Output MOS1 images from *mosspectra* of the observed counts (upper panel) and model exposure (lower panel) in sky coordinates. The images are for the 0.35 – 1.25 keV band.

In these rather lengthy calls, quite a few input parameters can be ignored if the user is willing to accept the default names for the output files. As a result, we can divide the parameters into required parameters and more cosmetic ones. The essential input parameters are the event file name (in the case of the pn, the OOT event file name as well), the ccids to be included, and the flags for source removal, *withsrcrem*, region specification, *withregion*, and image production *withimages*. For source removal one needs to specify the .fits format source region file *maskdet=P-bkregtdet.fits* and *masksky=P-bkregtsky.fits*. For region specification

one needs to specify the *regionfile* parameter. For image construction one needs to specify the energy limits for those images, *elow* and *ehigh*.

In these calls, *region=regm1.txt* is an ascii file containing region descriptors in detector coordinates<sup>14</sup> for the region of interest. The format of the region descriptor files is discussed below. If the selection is

<sup>14</sup>The use of detector coordinates, DETX and DETY, is required for the selection expression. While inconvenient, it is forced by the usage of certain SAS tasks. In practice the use of detector coordinates rather than sky coordinates is not a problem because of the very good pointing stability and the lack of a programmed wobble.

to exclude point sources then *cheese* or an equivalent must have previously been run.

One must specify which CCDs (for MOS) or quadrants (for pn) are to be included, as a string of booleans; “T” for including a CCD or quadrant, and “F” to excluding a CCD or quadrant. (Note that this is a change from previous versions of ESAS.) Be sure to mark as excluded chips that are in an anomalous state or chips that are not operating, such as MOS1 CCD#6 after it was lost to a micrometeorite strike. It is necessary to exclude CCD#1 if it is in small window or timing modes. It is also necessary to exclude chips with no valid events<sup>15</sup>.

For the pn the pixel pattern can be set, `pattern=0` will select single pixel events while `pattern<=4` will select both single and double pixel events. As discussed above, both of these choices have their advantages, and one may want to extract both spectra for some types of analysis. One *could* set the pixel pattern for the MOS, but it must be set to 12 for the current version of ESAS.

One can also specify the names of many of the output files, `cornerfile`, `imagefile`, `expmap`, `spmask`, `specfile`, `rmffile`, and `arffile`, but that is not necessary.

It should be noted that both *espfilt* and *mosspectra/pnspectra* extract the event list for the corner data. We strongly suggest that the `P-corevc.fits` file produce by *espfilt* be removed from the working directory before running *mosspectra/pnspectra* because we need the filtering of the corner data to be consistent between the current observation and an existing database of corner data. We can’t guarantee that the *espfilt* filtering in future versions will remain consistent as that routine has different needs.

The output of each run of *mosspectra* is the following lists of files. The first list contains files created with `withimage=no`:

`P-corccdN.pi` - observation data in the corner region for chip N (which does not exist for chip 1). This file is an intermediate step needed by *mosback*. An oot file is also produced by *pnspectra*.

`P-corevc.fits` - event file for the corner regions.

`P-corfwccdN.pi` - filter-wheel closed data in the corner region for chip N (which does not exist for chip 1). This file is an intermediate step needed by *mosback*. An oot file is also produced by *pnspectra*.

`P-expimt.fits` - exposure map in sky coordinates for

300-8000 keV

`P-fovccdN.pi` - observation data in the region of interest for chip N. This file is an intermediate step needed by *mosback*. An oot file is also produced by *pnspectra*.

`P-fovfwccdN.pi` - filter-wheel closed data in the region of interest for chip N. This file is an intermediate step needed by *mosback*. An oot file is also produced by *pnspectra*.

`P-fovimpdet.fits` - image of the region of interest over all energies in detector coordinates. This will be used by *proton* to generate the soft proton image

`P-fovimt.fits` - image of the region of interest in the specified energy band in sky coordinates. Even if one is not interested in producing images, this is useful diagnostic image to check to make sure that you have included only the correct region for your spectrum. An oot file is also produced by *pnspectra*.

`P-maskimt.fits` - mask for the region of interest.

`P-fovt.pi` - spectral data for the entire region of interest (i.e., the FOV excluding the excised point sources and certainly excluding the corner data)

`P.arf` - the auxiliary response file for the region covered by `P-fovt.pi`

`P.rmf` - the redistribution matrix for the region covered by `P-fovt.pi`

where P is a prefix such as “mos1S001” and n is the chip number. The `P-fovfwccdN.pi`, `P-corfwccdN.pi`, `P-fovccdN.pi`, and `P-corccdN.pi` files will be used by *mosback* or *pnback* to create the QPB spectrum/image for the region of interest. For the pn, each of the spectrum (`*.pi` files) will have an analogous counterpart with N running from 1 to 4 over the quadrant numbers. There will also be an oot counterpart for each spectral file.

The output files listed above are for the entire energy range (spectroscopic files) or for the 0.3-8.0 keV range (imaging files). These files are always produced for the same energy ranges no matter what the specified `elow` and `ehigh`. This second list contains output images that are created when *mosspectra* or *pnspectra* are run in imaging mode (`withimages=yes`, `elow!=0` and/or `ehigh!=0`):

`expimccd1-elo-ehi.fits` - The exposure map for CCD1 in detector coordinates.

`expimsky-elo-ehi.fits` - The exposure map for all selected CCDs in sky coordinates.

`fovimccd1-elo-ehi.fits` - The object image in detector coordinates for CCD1.

`fovimd1-elo-ehi.fits` - The object image for all selected CCDs in band elo-ehi in detector coordinates.

<sup>15</sup>Instances of chips that are active but producing no valid events are rare, but do occur on occasion. We cite as an example 0690751601 where all MOS1-4 and MOS1-7 events are marked as being derived from bad frames.

dinates. This file will be needed by the *proton* task.

*fovinsky-elo-ehi.fits* - The object image for all selected CCDs in band *elo-ehi* in sky coordinates.

*fwcimccdNdet-elo-ehi.fits* - The FWC data within the FOV for CCD N.

*maskimccd1-elo-ehi.fits* - Mask file for CCD1 in detector coordinates.

*maskimsky-elo-ehi.fits* - Mask file for all selected CCDs in sky coordinates. An oot file is also produced by *pnspectra*.

where *elo-ehi* is the band selected by *elow* and *ehigh*.

### 5.13.2. Combination OR *mos/pnback*

The tasks *mosback* and *pnback* take the intermediate files produced by *mosspectra* and *pnspectra*, as well as information from the calibration database, and turn them into model QPB spectra and images in detector coordinates. They must be run separately for each call of *mosspectra* and *pnspectra* with different region selections (spectral analysis) or different band selections (image analysis) The following are examples of their calls:

```
mosback inspecfile=mos1S001-fovt.pi
      elow=350 ehight=1100 ccids="T T T F F F T"
mosback inspecfile=mos2S002-fovt.pi
      elow=350 ehight=1100 ccids="T T T T F T T"
pnback inspecfile=pnS003-fovt.pi
      inspecoot=pnS003-fovtoot.pi
      elow=350 ehight=1100 quads="T T T T"
```

where *inspecfile* is the source region spectrum produced by *mosspectra* or *pnspectra*, *elow* and *ehigh* are the limits of the energy band for the creation of the images and must be the same as used in the *mosspectra* or *pnspectra* calls. The seven *ccd#*'s for *mosback* and four *quad#*'s for *pnback* must be the same as used in the *mosspectra* and *pnspectra* calls. For *pnspectra* one also needs the spectrum extracted over the same source region from the OOT events file, *inspecoot*. Note that multiple bands can be run without overwriting vital files. However, as mentioned above, if multiple regions from the same exposures are extracted (e.g., for spectral analysis purposes such as investigating the radial profile of a cluster of galaxies) renaming of certain files is required.

The output of *mosback* is primarily diagnostic:

*P-augindiv.qdp* - The augmented unexposed corner spectra for each chip used.

*P-augspec.qdp* - The source spectrum and the background spectrum.

*P-bkgaccum.qdp* - The accumulated unexposed corner spectra for each chip used.

*P-bkgindiv.qdp* - The corner data for each chip used(?).

*P-bkgspec.qdp* - The QPB background spectrum for the region of interest.

*P-bridgefit.qdp* - A plot showing the function fitted to the QPB background spectrum in order to bridge the region occupied by the Al K lines.

*P-ratioindiv.qdp* - The ratio of observation unexposed corner data to FWC unexposed corner data for each chip.

All of these are QDP plot files. Similar files are produced by *pnback*.

The essential output files for *mosback* is:

*P-bkg.pi* - The QPB spectrum for the area of interest.

*P-coraugN.pi* - The augmented corner spectra for chip N.

*P-bkgimdet-elo-ehi.fits* - The QPB background image in detector coordinates for the region of interest, in the *elo* to *ehi* band, if the *withimages* flag is set to true.

The files produced by *pnback* are similar and the essential files are:

*P-bkg.pi* - The QPB spectrum. Note that this is the essentially the modified FWC data after the subtraction of the OOT events from the FWC data.

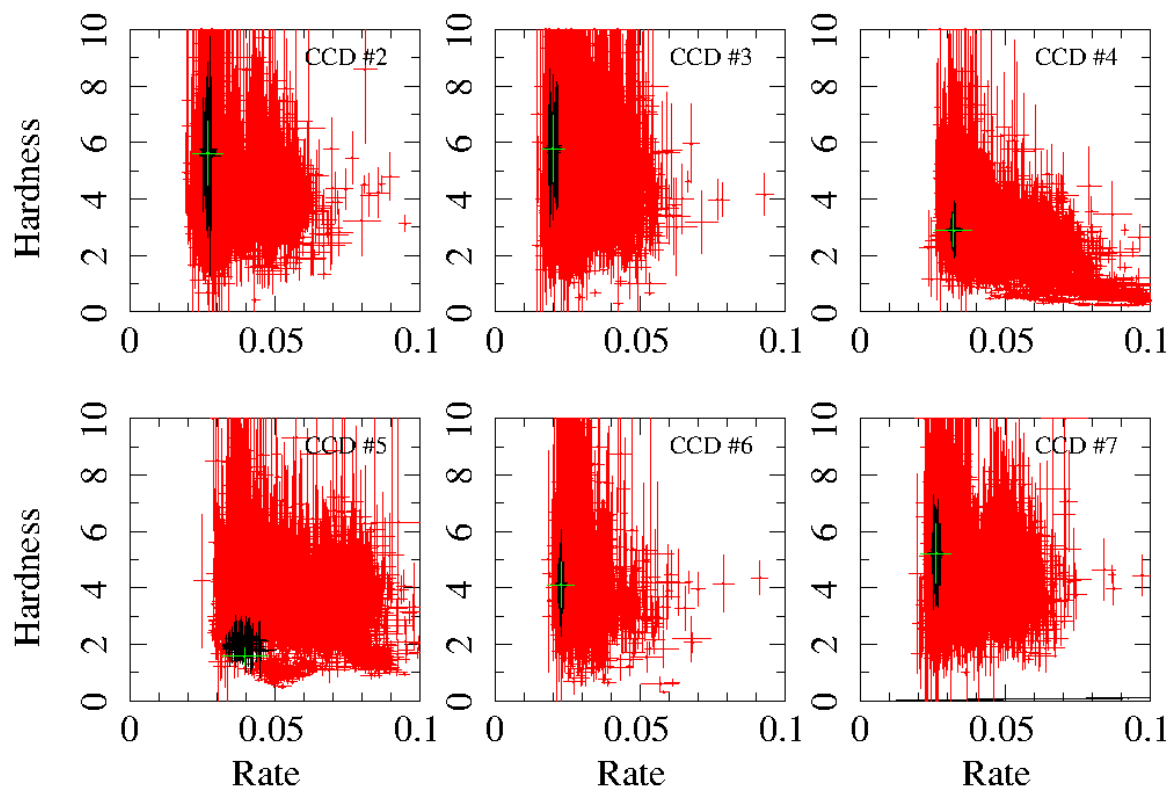
*P-fovtootsub.pi* - The object spectrum after the subtraction of the OOT events from the observation.

*P-bkgimdet-elo-ehi.fits* - The QPB background image in detector coordinates for the region of interest, in the *elo* to *ehi* band, if *withimages* is set to true.

### 5.13.3. Diagnostics

There are times that the output from *mos/pn spectra/back* is not optimal. This is usually due to one chip (or one quadrant) being ill-behaved. The multitudinous *P-\*.qdp* files will help you track down which chip or quadrant is causing problems, and will help diagnose the problem with that chip or quadrant. It is also a good idea to look at the *P-fovimt.fits* files to make ensure that the correct region has been selected and that the point sources have been properly excised. Another important diagnostic is the *P-augindiv.qdp* which shows the hardness and rate of the observation compared to other observations in the archive, as seen in Figure 16. Observations falling on the edges of the distributions are cause for concern.

## 97820101 -- MOS1S003



ssnowden 26-Mar-2015 11:07

Fig. 16.— Plots showing the selection of data used for the augmentation of the corner spectra (from `mos1s003-aug.qdp`). Red points indicate the full set of data while black points indicate which data were selected. Note that CCD#5 is close to the region indicating anomalous data (grouping of points just slightly down and to the right of the green point), and so suggests further diagnostic monitoring, e.g., a smoothed soft-band image can be made and examined for enhanced emission from the CCD area.

### 5.13.4. Clobbering

Given the large number of files produced by `mosspectra`, `pnspectra`, `mosback`, and `pnback`, it is important to understand what can be discarded, what is required for spectral analysis, what is required for imaging analysis, and what must be removed if you change the region (imaging or spectroscopy) or band (imaging only).

There are only a few simple things that need to be remembered.

1. Changing the band to be used does not change any of the spectral files (`*.pi`, `*.arf`, `*.rmf`, etc.). Those files will be overwritten, but with identical copies of themselves. It will produce a new set of all those files with `e1o-ehi` in their names.
2. Changing the region to be used changes *all* the output files except the corner spectra files: (`P-fovccdN.pi` and `PfovfwccdN.pi`). The existing files will all be overwritten. Rename or

(better yet) move the files to a subdirectory to protect them.

From these rules you can determine what might need to be deleted (or saved) when you change the band or region of interest. Since the most time consuming part is doing the `backscal` for each spectrum file, minimizing the re-creation of spectrum files is important!

### 5.13.5. Region Descriptors

As noted above, there are multiple levels of filtering and masking employed by `mosspectra` and `pnspectra`. The first filter is for pattern and flag and the largest possible FOV. The second filter defines the region of interest and is taken from an ascii file specified by the user in the `regionfile` parameter. This filter expression is, quite literally, concatenated onto the end of the first filter. The third filter uses the `P-bkgregBdet.fits` file to exclude the locations of point sources, and it is concatenated onto the end of



the previous expression. Thus, the ascii file defining the region must be constructed carefully, and in detector coordinates.

The region descriptors in the ascii files, which we have named `regm1.txt`, `regm2.txt`, and `regpn.txt` for convenience, can be created easily using the *xmmselect* task in SAS. Using the *xmmselect* GUI, produce an image of the sky in detector coordinates (i.e., click the square boxes for DETX and DETY and then click Image, and then Run in the pop-up *evselect* GUI). In the image displayed in the *ds9* window, define the desired region, and then click the 2D region button in the *xmmselect* GUI which will transfer the region descriptor to the *Selection expression* window. The region descriptor can then be copied to the `reg*.txt` file where the symbols “&&” must be added to the start of the string, for example:

```
&&((DETX,DETY) IN circle(300,500,1200))
```

where the numbers 300 and 500 are the DETX and DETY coordinates of the center of the extraction region and 1200 is the radius of the circular region. All numbers are in units of  $0.05''$  so the specified region is  $60''$  in radius. This process needs to be done separately for all instruments as the detector coordinates for a given sky region are different in each detector. To create an annulus extraction region, for instance for a  $30'' - 60''$  degree annulus, the region expression is:

```
&&((DETX,DETY) IN circle(300,500,1200))
&&!((DETX,DETY) IN circle(300,500,600))
```

where the “!” sign is for the Boolean “not”.

A number of users have noted that *evselect* has an option

```
MASK(FILENAME)
```

that allows one to use an image mask to select the events to be extracted, which allows one to build very complex extraction regions. However,

1. because the extraction regions used by *mosspectra* and *pnspectra* use detector coordinates, the mask must also be in detector coordinates. We do not yet have a tool to rotate/recast a sky coordinate image into a detector coordinate image.

2. through some subtlety that we have not yet identified, detector coordinate masks *seem* to work with *pnspectra* but not for *mosspectra*.

We do hope to have this working by the next release.

### 5.13.6. Creating Identical Regions Across Observations:

If you want to extract precisely the same region on the sky for all three instruments, using the above method to define the region for the first instrument is easy. However, transferring that region to the other instruments requires a different approach.

The task *convregion* provides an alternative method for converting region information from right ascension and declination to detector coordinates for circular and elliptical regions. Run with `mode=2`, *convregion* will input an ASCII file with region information and output an ASCII file with the necessary region information.

```
convregion mode=2 inputfile=convreg.txt
outputfile=tempout.txt
imagefile=mos1S001-fovimt.fits
```

where in this case the input file, `convreg.txt`, which will extract circular and elliptical annuli, contains:

```
circle 207.12 26.59 4.0
!circle 207.12 26.59 2.0
ellipse 207.32 26.59 4.0 2.0 30.0
!ellipse 207.32 26.59 2.0 1.0 30.0
```

and the output file contains:

```
circle -2063.3 5786.5 4800.0
!circle -2063.3 5786.5 2400.0
ellipse 2372.8 -6303.1 4800.0 2400.0 260.153
!ellipse 2372.8 -6303.1 2400.0 1200.0 260.153
```

The output data must then be edited into a region expression:

```
&&((DETX,DETY) IN circle(-2063,5786,4800))
&&!((DETX,DETY) IN circle(-2063,5786,2400))
||((DETX,DETY) IN ellipse(2373,-6303,4800,2400,260))
&&!((DETX,DETY) IN ellipse(2373,-6303,2400,1200,260))
```

*convregion* can also both be run with direct input:

```
convregion imagefile=mos1S003-obj-image-sky.fits
shape=CIRCLE ra=207.12 dec=26.59 radius=3
semimajor=2 semiminor=3 rotangle=30 mode=3
```

or using a region file as input.

It should be noted that *convregion* does not yet convert boxes. Specifying a region in sky coordinates and converting it to detector coordinates for the three in-

struments produces approximately the correct result. However, it should be remembered that the detectors do cover different regions of the sky; different chips may be off between the two MOS detectors, and the pn is somewhat offset from the MOS detectors. Thus, *if* it is really important to have *exactly* the same region for all three detectors, one has to use the chip information as well. Thus, making sure that you have *exactly* the same region for every detector is a messy and painful business<sup>16</sup>.

### 5.13.7. Point Source Masking

The use of point source masking is accomplished in two ways, by using mask images or source region lists produced by running the perl script *cheese* (see § 5.12). If images are of interest then a simple masking of the data can be done by using the “Swiss Cheese” images. If the desire is to remove the source contributions to spectra then the source region lists must be used in the *mosspectra* or *pnspectra* calls, which is done using a calling parameter of the tasks. If point sources are removed from the spectra in the calls to *mosspectra* and *pnspectra* they are also removed from the images, and during the image creation process (see below) the images must also be masked using the Swiss-cheese images. However, as discussed below, it is possible to exclude point sources in the spectral fits used to determine the SP contribution and then scale the results to the whole image.

## 5.14. Preparing for Further Analysis

If spectral analysis is the goal for the processing, this is the end of ESAS, and you can use your favorite spectral analysis program with the P-*fov*t.pi, P-*bkg*.pi, P.*rmf*, and P.*arf* files. (For the pn, you can use the P-*fov*t<sub>oot</sub>sub.pi file with the P-*bkg*.pi file.)

If imaging analysis is the goal of the processing, you will now need to do a spectral analysis to determine 1.) the strength and spectral shape of the residual SP

emission, and 2.) the strength of the SWCX emission. Using the results from the spectral fitting described below, one then proceeds with the ESAS tasks *proton*, *swcx*, and *combimage*, which are described in §7.

If only spectral analysis is to be done, RMFs and ARFs, and background spectra are desired (and if so the “*withimages=no*” should be set for the tasks above), *mosback* and *pnback* are as far through the processing as one needs to go for a single exposure and region selection. The processing must be repeated for all instruments individually, as well as for any selections for different regions on the sky and for different exposures of the same instrument. Before rerunning *mosspectra* and *pnspectra* for a different region on a given instrument (MOS1, MOS2, or pn), *mosback* or *pnback* should also be run and the important output files should be either renamed or moved so they are not overwritten. This includes the source and background spectra as well as the RMF and ARF files. For example, the process of completing one spectral set is as follows:

```
mosspectra eventfile=mos1S001-allevc.fits
           withsrcrem=yes maskdet=mos1S001-bkregtdet.fits
           withregion=yes regionfile=regmos1.txt
           keepinterfiles=yes pattern=12
           withimages=yes
           elow=350 ehigh=1100 ccds="T T T F F F T"
mosback inspecfile=mos1S001-fov
```

```
           t.pi
           withimages=yes elow=350 ehigh=1100
           ccds="T T T F F F T" withqdp=yes
mv mos1S001-fov
```

```
           t.pi mos1S001-fov-reg1.pi
mv mos1S001-bkg.pi mos1S001-bkg-reg1.pi
mv mos1S001.rmf mos1S001-reg1.rmf
mv mos1S001.arf mos1S001-reg1.arf
mv mos1S001-fov
```

```
           imspdet.fits
           mos1S001-fov
```

```
           imspdet-reg1.fits

mosspectra eventfile=mos2S002-allevc.fits
           withsrcrem=yes maskdet=mos2S002-bkregtdet.fits
           withregion=yes regionfile=regmos1.txt
           keepinterfiles=yes pattern=12
           withimages=yes
           elow=350 ehigh=1100 ccds="T T T F F F T"
mosback inspecfile=mos2S002-fov
```

```
           t.pi
           withimages=yes elow=350 ehigh=1100
           ccds="T T T F F F T" withqdp=yes
mv mos2S002-fov
```

```
           t.pi mos2S002-fov-reg1.pi
mv mos2S002-bkg.pi mos2S002-bkg-reg1.pi
mv mos2S002.rmf mos2S002-reg1.rmf
mv mos2S002.arf mos2S002-reg1.arf
mv mos2S002-fov
```

```
           imspdet.fits
           mos2S002-fov
```

```
           imspdet-reg1.fits

pnspectra eventfile=pnS003-allevc.fits
```

<sup>16</sup>To do this one has to work in detector coordinates. For the MOS detectors, you need to know the location and size of each chip that is on, as well as the location of the mask. For the pn you need to know the center and size of the detector, as well as the location of the mask. For each detector, that information has to be converted to sky coordinates for some arbitrary observation. Then, the sky coordinates can be converted into the detector coordinate system for each of the detectors in turn; that is, you need to determine how the MOS1 detector coordinates map to the MOS2 and pn detector coordinates. Then, for each detector, you have to specify the intersection of the active regions of all the active CCDs. Thus the selection expression is very long! Note that there are rotations involved as well. However, once this is done for all chips, you never have to do it again, you just discard whatever set of chips are not on. I’ve done this, but am not convinced that the rotations are all correct. This information may get posted on a website.

```

ootevtfile=pnS003-allevc-oot.fits
withsrcem=yes maskdet=pnS003-bkregtdet.fits
withregion=yes regionfile=regpn.txt
keepinterfiles=yes pattern=0
withimages=yes
elow=350 ehigh=1100 quads="T T T T"
pnback inspecfile=pnS003-fovft.pi
inspecoot=pnS003-fovftoot.pi
withimages=yes elow=350 ehigh=1100
quads="T T T T" withqdp=yes
mv pnS003-fovft.pi pnS003-fovft-reg1.pi
mv pnS003-fovft-oot.pi pnS003-fovft-oot-reg1.pi
mv pnS003-bkg.pi pnS003-bkg-reg1.pi
mv pnS003.rmf pnS003-reg1.rmf
mv pnS003.arf pnS003-reg1.arf
mv pnS003-fovimpdet.fits
pnS003-fovimpdet-reg1.fits
mv pnS003-fovft-os.pi pnS003-fovft-os-reg1.pi

```

One author likes to do the above "mv-ing". The other author of this cookbook prefers to create directories for each instrument region combination. This procedure is demonstrated in the appendix.

Note that the task *pnback* produces an OOT subtracted spectrum with the file name *prefix-fovft-os.pi* which should be used for spectral analysis. The spectra, however, are therefore count rates rather than counts which makes the use of *cstat* statistics problematic.

In most cases for chi-squared statistics the data are going to be too sparse, particularly at higher energies, to be usefully fit without binning. Binning can be done conveniently by using the Ftool *grppha*. In the binning process with *grppha* the background spectrum, RMF, and ARF files can also be associated with the source spectrum simplifying data entry into the spectral fitting program. The following commands will group the data with a minimum of 50 counts per channel and place the output spectrum in a file with the *-grp* qualifier added to the file name.

```

grppha
mos1S001-fovft.pi
mos1S001-fovft-grp.pi
chkey BACKFILE mos1S001-bkg.pi
chkey RESPFILE mos1S001.rmf
chkey ANCRFILE mos1S001.arf
group min 50
exit

```

```

grppha
pnS003-fovftootsub.pi
pnS003-fovft-grp.pi
chkey BACKFILE pnS005-bkg.pi
chkey RESPFILE pnS005.rmf

```

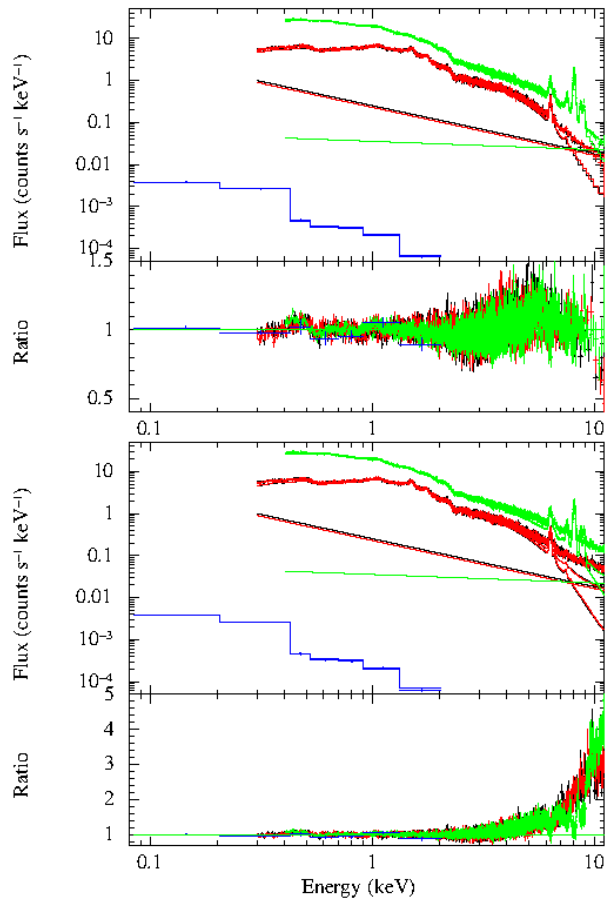


Fig. 17.— Fitted MOS1 (black), MOS2 (red), pn (green), and RASS (blue, see Section 6.6 below) spectra from the entire field of view of the Abell 1795 observation. The upper panel shows the fit with the model background subtracted (the poorness of the fit is due to fitting one emission temperature to a multithermal spectrum). The lower panel shows the same fit but the data do not have the particle background subtracted, thus the deviation at higher energies. The diagonal lines in both plots show the contribution of the fitted residual SP background.

```

chkey ANCRFILE pnS005.arf
group min 50
exit

```

Of course, you will probably want to add more qualifiers to some of these file names, particularly if you are going to be fitting more than one observation of the same source. However, this is just a minimal example.

Figure 17 shows the fitted MOS1, MOS2, and pn spectra from the entire field of view of the Abell 1795 observation. The difference between the two panels shows the effect of subtracting the model QPB. At lower energies the source clearly dominates over the background but at  $E \sim 4$  keV the falling flux and falling instrumental effective area coupled with the relatively flat QPB allows the latter to dominate. Note,

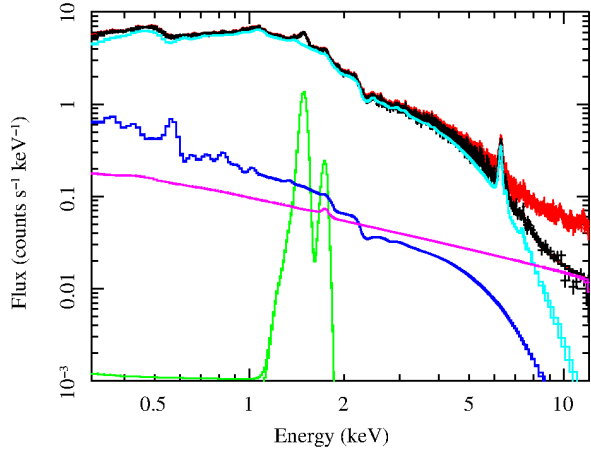


Fig. 18.— Various components of the MOS1 spectral fit shown in Figure 17. The black points and curve show the background subtracted and fitted data. The red points show the observed spectrum without the particle background subtracted. The light blue curve shows the fitted cluster spectrum, the dark blue curve shows the fitted cosmic background spectrum. The green curve shows the fitted Al  $K\alpha$  and Si  $K\alpha$  instrumental lines. The violet (it may appear red depending on your printer) curve shows the fitted SP component.

however, that Abell 1795 is a relatively bright object and thus the QPB will be relatively more significant at lower energies for many other observations. The particle background even in this observation will also be much more important when the outer annuli of the cluster are considered in § 6.

Figure 18 shows the contributions of the various spectral components to the observed spectrum of Abell 1795 (demonstrated for the MOS1 detector). For a source as bright and extended as Abell 1795, the object spectrum dominates all other components except at the highest energies. However, it is clear that observations of fainter sources such as the cosmic background will be very significantly affected by the particle background and possible residual SP contamination over the entire energy band.

## 6. SPECTRAL FITS

Spectral fitting can get very complicated very quickly (e.g., see Figure 18). There are typically six or seven spectral components that should be fit to the data, even before the spectrum of the object of interest (if that is not the cosmic background) is added. These components are split between instrumental and cosmic origins. Some of the components can vary with time, observation direction on the sky, position on the detector, location of the satellite in its orbit, and location of the Earth in its orbit.

As alluded to above, even if your primary interest is imaging, you will still need to do *some* spectral fitting to determine the contributions from the residual soft proton flares and solar wind charge exchange (SWCX). If you have only a single observation, disentangling the solar wind charge exchange emission from your source spectrum is likely to be impossible. If you have multiple overlapping observations, determining the SWCX and residual soft proton flare (SPF) emission becomes more certain. **In the case of imaging analysis, the fitting of the source spectrum may not need to be as physically meaningful as it would be for a spectral analysis,** it just needs to reproduce the underlying source spectrum well enough to get a good measure of the residual soft proton flare and SWCX emission

The following sections describe the peculiarities of spectral fitting for ESAS extracted spectra. Those concentrating on imaging analysis may not need to read more than the first several parts. For imaging analysis, fitting in the “traditional” manner with subtracted background will probably be sufficient. More rigorous spectral analysis will probably require simultaneous fitting of the source *and* the background. ESAS is not yet creating the background spectra in a way that such fitting can be done with the C-statistic, but that is being developed.

### 6.1. Quiescent Particle Background

The quiescent particle background (QPB) spectra created in the previous section is primarily a continuum spectrum. There are a number of lines due to emission by materials in the vicinity of the detector, as can be seen in Figures 13 and 14. Many of these lines are included in the particle background spectra created in the previous section. However, there are a number of emission lines that are too strong to be well modeled in this manner, and these lines were excised from the quiescent particle background spectra. It should also be remembered that the background spectra are created from collections of corner data and filter-wheel-closed data that have accumulated over the entire mission. The line spread function has varied significantly

over the mission and thus the background spectra derived from either corner data or the filter wheel closed (FWC) data have a different line width than the actual observation data. This is not a problem for weak lines, but it is a substantial problem for the strong lines. Further, there are slight variations in the gain which are often not noticeable except around the very strongest lines.

Since the Al  $K\alpha$  and Si  $K\alpha$  lines for the MOS and the Al  $K\alpha$  and Cu  $K\alpha$  lines for the pn are excised from the QPB spectra, these lines must be fit explicitly to the data. For the pn it may be more reasonable to simply exclude the Cu line region from the analysis, although this may shorten the lever arm for fitting the SPF component.

For the MOS, fitting the instrumental lines is done easily by the addition of two Gaussians in the spectral model with energies of  $E \sim 1.49$  keV and  $E \sim 1.75$  keV with zero width. The intensities of these two lines vary over the individual detectors but they seem to be in relatively good agreement between the two MOS instruments. We note, however, that since the lines are strong, minor errors in the line redistribution function can produce significant residuals, particularly at energies just below that of the line.

The pn, unfortunately, requires six lines at  $E \sim 1.49$  keV,  $E \sim 7.47$  keV,  $E \sim 7.06$  keV,  $E \sim 8.03$  keV,  $E \sim 8.62$  keV, and  $E \sim 8.90$  keV. The normalization of the pn Si line should be fixed at zero.

We note that for the typical exposure, only the brightest of the lines in Table 7 need be fit. However, for the QPB spectra available from the corner data demonstrate the existence of several more lines. Fitting the QPB spectra by themselves is difficult given

Table 7: Instrumental Lines.

Line	Energy (keV)	Instrument
Mg $K\alpha$	1.253	MOS & pn
Al $K\alpha$	1.486	MOS & pn
Al $K\beta$	1.557	MOS only?
Si $K\alpha$	1.739	MOS
Au $M\alpha$	2.123	MOS & pn
???	4.535	MOS
Cr $K\alpha$	5.405+5.415	MOS & pn
Cr $K\beta$	5.947	MOS & pn
Fe $K\alpha$	6.391+6.404	MOS & pn
Fe $K\beta$	7.058	MOS & pn
Ni $K\alpha$	7.460+7.478	MOS & pn
Cu $K\alpha$	8.028+8.048	pn
Zn $K\alpha$	8.615+8.638	MOS & pn
Cu $K\beta$	8.905	pn
Au $L\alpha$	9.628	pn
Au $L\alpha$	9.713	MOS only?
Au $L\beta$	11.442+11.584	MOS & pn

that the intrinsic line spread function has increased with time, but it is possible, as shown in Figure 19. As we move to more statistically correct ways of fitting the data, such fitting may become important; currently this is shown merely as a demonstration of the number and location of instrumental lines in the QPB spectra.

## 6.2. Cosmic & Galactic Backgrounds

The cosmic background, even if not the object of the observation, must be modeled explicitly as it is a significant background in all directions and at all energies, and it varies significantly both in intensity and spectral shape over the sky. The cosmic background can usually be modeled by four or fewer components.

First, a cool ( $E \sim 0.1$  keV), unabsorbed thermal component representing emission from the Local Hot Bubble and heliosphere. It may be useful to look at the work of Snowden et al (1998) or Liu et al. (2017) to get an understanding of the strength of the LHB in the direction of your field. The exact temperature of this component will depend on the model (Raymond & Smith, APEC, MeKaL, etc.) and abundances (Anders & Grevesse, Wilms, etc.) used. (Note that the temperature quoted by Liu et al. is not appropriate for the current version of APEC!)

Second, a cool ( $E \sim 0.1$  keV), absorbed thermal component representing emission from the cooler halo. This component may have an intrinsically zero intensity or may be effectively absorbed if the Galactic column is greater than a few times  $10^{20}$  H I  $\text{cm}^{-2}$ . In either case the component can be deleted from the fitted model.

Third, a higher temperature ( $E \sim 0.25 - 0.7$  keV), absorbed thermal component representing emission from the hotter halo and/or intergalactic medium. At high Galactic latitudes, the 0.25 keV component tends to dominate. In the plane of the Galaxy, the 0.7 keV component can be quite strong. For an observation near the Galactic plane, this component may also end up representing distributed thermal emission through the Galactic disk.

Fourth, an absorbed power law with  $\alpha \sim 1.45$  representing the unresolved background of cosmological sources.

Since getting some of these normalizations correct is important, consider a fit of the model

`apec+tbabs(apec+apec+pow)`

to the RASS data for  $b > 85^\circ$  and  $45^\circ < \ell < 270^\circ$ . We found

$$N_{LHB} = 8.63 \times 10^{-7} \text{ for } kT = 0.0901 \text{ keV}$$

$$\begin{aligned} N(H) &= 0.0136 \times 10^{22} \text{ cm}^{-2} \\ N_{soft \text{ halo}} &= 2.57 \times 10^{-6} \text{ for } kT = 0.1074 \text{ keV} \\ N_{hard \text{ halo}} &= 1.06 \times 10^{-6} \text{ for } kT = 0.2373 \text{ keV} \\ N_\Gamma &= 1.11 \times 10^{-6} \text{ for } \Gamma = 1.45 \end{aligned}$$

Note that since this is RASS data, the normalizations will be per square arcminute. In this fit the absorbing column was held fixed since the absorption is due mainly to atomic gas which is well traced by HI 21 cm emission. At high Galactic latitude there is no perceptible 0.75 keV component. These values are representative of those found at high Galactic latitudes. We can now consider what values might be expected for other observations.

The normalization for the Local Hot Bubble emission (which also contains some of the heliospheric solar wind charge exchange emission) is somewhat higher than usual. Reference to Snowden et al (1998) shows why; the path length through the Local Hot Bubble is, in general, greater towards the poles than it is in the Galactic Plane.

The normalization of the Galactic halo components is typical for high Galactic latitudes; at lower Galactic latitudes the emission may be so strongly absorbed that the normalization is very uncertain. Kaaret et al (2020) have shown that the  $\frac{3}{4}$  keV Galactic halo emission is patchy, so the normalization of the harder component can be quite variable.

The normalization of the extragalactic power law has been studied by a number of authors and is  $\sim 10.5$  keV  $\text{cm}^{-2} \text{ s}^{-1} \text{ sr}^{-1} \text{ keV}^{-1}$ . This normalization translates to  $8.88 \times 10^{-7}$  photon  $\text{cm}^{-2} \text{ s}^{-1} \text{ keV}^{-1} \text{ arcmin}^{-2}$  at 1 keV, the standard xspec units for a single square arcminute. Note that the normalization for your particular observation is likely to be lower as you have removed point sources. In principle, given the cosmic normalization, the point source detection limit, and the luminosity function for extragalactic sources (suitably corrected to the absorbing column over the field of view), one can calculate the expected normalization. The XMM-ESAS package provides the task *point-source* which will calculate the normalization using several different models and user-selected parameters. In practice one finds significant variation. However, it is important to get this normalization correct because errors in this normalization are (usually) highly correlated with errors in the normalization of the SPF component; fixing the value of  $N_\Gamma$  to its expected value may help to better determine the normalization of the soft proton component.

The absorbing column density can be found using the HEASARC NH Tool:

<https://heasarc.gsfc.nasa.gov/cgi-bin/Tools/w3nh/w3nh.pl>,

the HEASARC X-ray Background Tool:

<https://heasarc.gsfc.nasa.gov/cgi-bin/Tools/xraybg/xraybg.pl>,

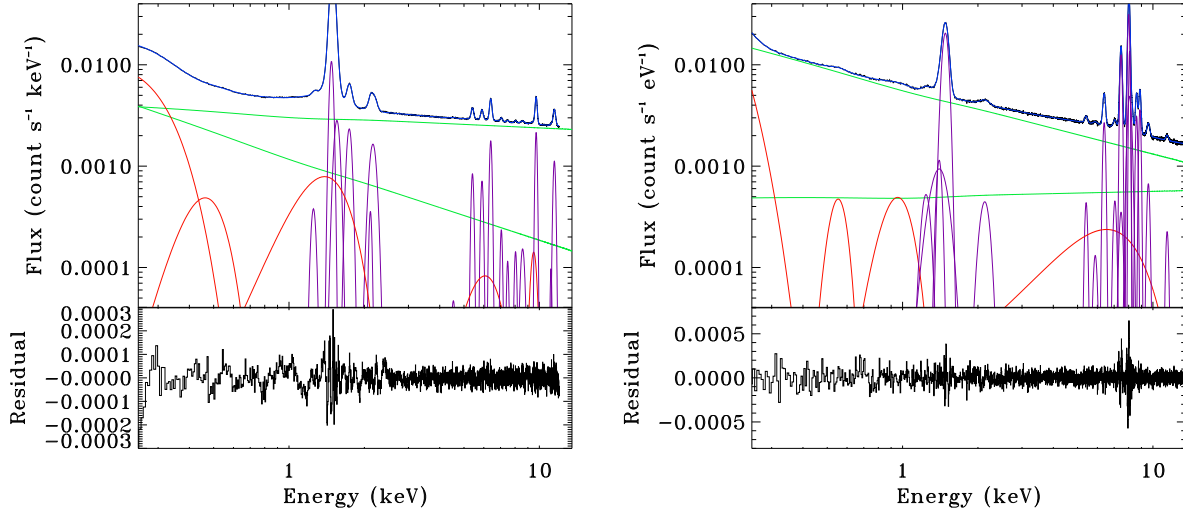


Fig. 19.— Fits and residuals to the mean MOS2 (left) and pn (right) spectra. The spectra have not been binned or smoothed. We show the complete decomposition of the background spectrum (black) into its components. *Green*: power law components for the bulk of the continuum. *Red*: low amplitude features in the continuum fitted with Gaussians. It is not clear what some of these features are. *Solid Purple*: emission lines fitted with zero-width Gaussians and no instrumental line spread function.

or the Planck maps. What one desires is the total column density of hydrogen, in all of its forms: H I, H II, and H<sub>2</sub>. One assumes that the ratio between the hydrogen column density and the column density of the actual absorbing atoms is reasonably set by the chosen abundances. There is a certain amount of complexity here, as the H I 21 cm measurements do not capture the molecular or ionized components of the absorbing ISM, and the abundances may vary significantly from one line of sight to another. Conversely, the translation from Planck dust emission to the actual column density of absorbing atoms is not as straight-forward as it might seem.

### 6.3. Residual Soft Proton Emission

The spectral shape of the SPF emission is highly variable. (See Kuntz & Snowden (2008) for a more complete discussion.) There is some evidence that, on average, the stronger flares have harder spectra, but such information is of little use when determining the spectral shape for a single observation. The mean spectrum at a given flare strength can be reasonably well fit with a broken power law. A typical flared observation will contain flares at various strengths with, presumably, different spectral shapes. Thus, rather than attempting to create a precise characterization of the spectral shape, we attempt to reduce the strength of the soft proton flare sufficiently so that error in the spectral shape will be insignificant. This is the reason why one filters with *espfilt* to remove the bulk of the

soft proton component rather than attempting to fit that component.

Since protons are not photons, the effective area curve should not be used. It would, for instance, produce edges where there are none in the soft proton flare spectrum. The case for or against using the redistribution function is more complex. Presumably, the deposition of energy with depth for soft protons will be different from photons. Conversely, there will still be losses both near the surface and at depth. Thus, one does expect the response to a monoenergetic beam of soft protons to have a finite energy width. Although I am not aware of a measured redistribution function for soft protons for the XMM CCDs, it is not strictly necessary. Since the energy spectrum for soft protons is not expected to have sharp features, the difference in shape between the proper redistributed spectrum and initial spectrum should not be strong. (We note that there has been work recently published on the redistribution function for soft proton flares. It has not yet been incorporated into ESAS.)

Thus, one can model the soft proton component with a broken power law which has not been folded through the instrument response, that is, without using the effective area file (the *.arf*) and using a special redistribution file (the *.rmf*) for an infinitely narrow line spread function. For Xspec V12 and higher this fitting is accomplished by adding a separate model with a *diagonal unitary matrix*. This is demonstrated in the XMM-ESAS examples in the Xspec \*.xcm

files. The diagonal matrices were previously supplied with the ESAS CalDB files. This is no longer the case. The diagonal response files are now accessible from the ESAS web site<sup>17</sup> as `mos1-diag.rsp.gz`, `mos2-diag.rsp.gz`, and `pn-diag.rsp.gz`.

For moderately SPF affected data, the SPF spectral parameters are usually robustly fit. For a simple power law the fitted index is typically in the range 0.5-1.0 but can vary between 0.1 and 1.4. However, in certain circumstances the index can blow up to relatively large or small (even negative) numbers. In these circumstances it is probably reasonable to freeze the index at a reasonable value (e.g., 0.1 for small or negative indices and 1.4 for large indices). The SPF index for the PN is likely to be different from that of the MOS detectors so in general they should not be linked, nor should any of the normalizations between different instruments be linked. This is a bit of an art form and it is not possible at this time to make any definitive rules. In addition, use of a broken power law spectrum with a break at 3.0 keV can be warranted. For broken power law fits the indices share the lower reasonable limit but may range to 2.5 or higher for the upper range. When the fitted value for the SP normalization drops below a few times  $10^{-6}$  arcmin<sup>-2</sup> the values are effectively zero. There is no practical difference between normalizations of  $10^{-7}$  and  $10^{-30}$ .

#### 6.4. Solar Wind Charge Exchange Background

A component that can be very problematic originates in the solar system and in the near-Earth environment. SWCX occurs when highly ionized ions in the solar wind collide with either exospheric hydrogen near the Earth or the neutral interstellar medium passing through the solar system and pick up an electron. The acquired electron is typically in a highly excited state which then radiatively decays.

Prominent SWCX emission lines include those from C VI, O VII, O VIII, Ne IX, and Mg XI that are some of the same lines that are important diagnostics of astrophysical plasmas. If an observation is long enough, some evidence for SWCX contamination can possibly be found by comparing the 0.5 – 0.7 keV and 2.0 – 8.0 keV light curves (see Carter & Sembay 2008). If the lower-energy light curve varies while the high-energy light curve doesn't, the data may be affected. Another SWCX diagnostic is the presence of very strong O VIII and Mg XI lines in the cosmic diffuse background component. Additional observations of a region may be the only way to determine the presence and minimum extent of SWCX contamination. Further discussion of solar wind charge exchange can be

found in Kuntz (2019).

To model the SWCX component, Gaussian lines at the characteristic energies (see Table 8 for most common lines in the EPIC energy range) with zero width are added to the spectral fit. In the fit, the line energies should be fixed until a reasonable fit is obtained and then allowed to float.

Generally, for imaging analysis, SWCX is not an issue. The SWCX lines typically have strengths that are small enough that they can be accommodated by the LHB/Galactic halo components, thus producing incorrect values for the temperatures and normalizations of those spectral components.

#### 6.5. Outside Constraints

One of the problems for the study of diffuse emission with *XMM-Newton* is that several of the important components have a significant fraction of their emission just outside the *XMM-Newton* band pass. The Local Hot Bubble has the bulk of its emission in the  $\frac{1}{4}$  keV band, and the same is true of the softer components of the Galactic halo. Thus, *XMM-Newton*-only spectral fits do not constrain these components well, despite the fact that there is still a significant amount of emission from them in the *XMM-Newton* band pass. These considerations are not usually important for the spectral fitting needed for imaging analysis!

One way of helping to constrain the fits for these components is to use the spectral energy distribution (i.e. a very low resolution spectrum) from the *ROSAT* All-Sky Survey (RASS). The RASS data has an effective angular resolution of 12' but has a sufficiently low count rate that one needs about a square degree of data to get a good signal-to-noise ratio. The RASS data is in seven bands over 0.11 to 2.X keV, so it overlaps *XMM-Newton* band pass nicely. This overlap allows one to see whether the *ROSAT* data, being taken from a somewhat larger region than an *XMM-Newton* FOV, is consistent with the *XMM-Newton* data. Use of the *ROSAT* SED with *XMM-Newton* spectra is further demonstrated in §6.7.1.

The RASS SED can be obtained using the HEASARC X-ray background tool: <https://heasarc.gsfc.nasa.gov/cgi-bin/Tools/xraybg/xraybg.pl>

One issue with using the RASS SED to fit the *XMM-Newton* spectra is that of the SWCX. The bulk of the SWCX emission in the RASS was due to magnetospheric/exospheric emission, which was manifested as the long-term enhancements (LTEs), which were removed before the release of the RASS. With the RASS there remains not only residual LTE emission, but also a certain amount of heliospheric SWCX emission. The work of Liu et al (2017) shows that the

<sup>17</sup>[https://heasarc.gsfc.nasa.gov/docs/xmm/xmmhp\\_xmmesas.html](https://heasarc.gsfc.nasa.gov/docs/xmm/xmmhp_xmmesas.html)



heliospheric contribution is relatively small.

There are a number of reasons not to attempt to fit a SWCX component in the RASS data. First, although we know that the SWCX at the time of the RASS observations will be different from that during the *XMM-Newton* observations, we do not have the information required to calculate the SWCX emission during the RASS. Further, we don't yet have a robust model for the SWCX emission within the  $\frac{1}{4}$  keV band. Second, the SWCX emission remaining in the RASS SED is small, and is lost in the only moderate signal-to-noise ratio of the RASS SED. However, for the most precise work, one would want to correct the RASS SED for the heliospheric emission found by Lie et al (2017).

## 6.6. Models and Xspec

One of the hurdles to using ESAS effectively is fitting the spectrum for the FOV; even in the most simple case, we need to fit multiple spectra with multiple models and multiple different response files, with a large number of linked parameters. Even for those with Xspec (sherpa, etc.) experience, there are many ways to make mistakes. This section steps through the essentials of such a fit for a simple case where the "source" emission is smoothly distributed over the FOV. We have found it useful to fit the spectra in terms of emission per square arcminute, as this allows quicker comparison between instruments, missions, and fields.

### 6.6.1. Importing the Data

For a typical XMM observation, there is data from three instruments. We will assume that the spectra have, as per §5.14, been grouped, and the names of the redistribution file (.rmf), the ancillary response file (.arf), and background file (-bkg.pi) have already been added to the header. Then we can import the data:

```
data 1:1 mos1S001-fovt-grp.pi
ig 1:0.0-0.3,11.0-**
data 2:2 mos2S002-fovt-grp.pi
ig 2:0.0-0.3,11.0-**
data 3:3 pnS003_0-fovt-grp.pi
ig 3:0.0-0.3,2.0-**
data 4:4 pnS003_4-fovt-grp.pi
ig 4:0.0-1.0,11.0-**
ig 4:7.2-9.2
data 5:5 rosat.pi
ig bad
```

Both of the MOS spectra are used for energies in  $0.3 < E < 11.0$ . Note that we are importing two

different versions of the pn data, one filtered to contain only the PATTERN==0 events, and the other filtered to include PATTERN<=4. The former is used only for  $0.3 < E < 2.0$  while the latter is used for  $1.0 < E < 11.0$ . For the pn data we exclude 7.2 to 9.2 keV in order to avoid fitting the several instrumental lines in that range. We have also included a spectrum from *ROSAT* in order to better constrain the lower end of the fit.

### 6.6.2. Declaring Models

There are a couple of different philosophies about setting up the models, and the authors often disagree. However, since this section is being written while the other author skives off on his farm, the bias will be clear.

There are four different things that are to be fit to the data: the cosmic X-ray background (§6.2), the instrumental lines (§6.1), the solar wind charge exchange (§6.4), and the residual soft proton flares (§6.3). Of these, the first three all require the full photon redistribution/effective area functions while the last requires its own redistribution function and no effective area function. Since there are two different redistribution functions involved, there have to be at least two different model declarations. At the minimum:

```
model const*const*(apec+tbabs(apec+pow)
+gaus+gaus+gaus+gaus+gaus+gaus)
parameter 1, etc.
```

```
resp 2:1 mos1-diag.rsp
resp 2:2 mos2-diag.rsp
resp 2:3 pn-diag.rsp
resp 2:4 pn-diag.rsp
model 4:spf const(bknpow)
parameter spf:1, etc.
```

However, I find that having a large multi-component model, especially where some components do not apply to some of the spectra (i.e., the XMM instrumental lines certainly do not apply to the *ROSAT* spectrum), rapidly becomes confusing. Thus, I find it more useful to declare the models as follows:

```
model const*const*(apec+tbabs(apec+pow))
parameter 1, etc.
```

```
resp 2:1 mos1S001.rmf
arf 2:1 mos1S001.arf
resp 2:2 mos2S002.rmf
arf 2:2 mos2S002.arf
resp 2:3 pnS003_0.rmf
arf 2:3 pnS003_0.arf
```

```

resp 2:4 pnS003_4.rmf
arf 2:4 pnS003_4.arf
model 2:alsi const*const*(gaus+gaus)
parameter alsi:1, etc.

resp 3:1 mos1-diag.rsp
resp 3:2 mos2-diag.rsp
resp 3:3 pn-diag.rsp
resp 3:4 pn-diag.rsp
model 3:spf const*const*(bknpow)
parameter spf:1, etc.

resp 4:1 mos1S001.rmf
arf 4:1 mos1S001.arf
resp 4:2 mos2S002.rmf
arf 4:2 mos2S002.arf
resp 4:3 pnS003_0.rmf
arf 4:3 pnS003_0.arf
resp 4:4 pnS003_4.rmf
arf 4:4 pnS003_4.arf
model 4:swcx const*const*(gaus+gaus+gaus+gaus)
parameter swcx:1, etc.

```

This second way of doing things may take longer to set up, but it requires fewer statements for linking parameters and helps avoid errors when linking the parameters. In this formulation, the first model contains the cosmic X-ray background components, the second model contains the instrumental lines (the big Al  $K\alpha$  and Si  $K\alpha$  lines), and the third contains the residual SPF emission. Finally, there is the SWCX, which may be completely unnecessary for your work. Either way you set up the models there are at least 173 parameters. However, many of these parameters are linked, minimally used, or simply there in case they might be needed. We will now consider the models in order.

*The Cosmic Background:*

```
const*const*(apec+tbabs(apec+pow))
```

There are two constants. The first is the number of square arcminutes covered by the field of view. This value will be different for each instrument, and should be frozen once set. The solid angle of the region from which the spectrum was extracted can be determined from the BACKSCAL keyword of the SPECTRUM extension of the spectrum file. The BACKSCAL keyword is in units of  $0.05'' \times 0.05''$  pixels.

The second constant is a fudge factor to account for the slight mismatches in the calibrations of the instruments. It should initially be set to unity and frozen. Once the fit is nearly correct, it can be thawed; it should change by less than 10%.

The `apec+tbabs(apec+pow)` is the actual cosmic background; an unabsorbed Local Hot Bubble com-

ponent, an absorbed Galactic halo component, and an absorbed contribution from the unresolved extragalactic background. Purists will note that the Galactic halo has multiple components; whether you include that complexity will depend upon the signal to noise ratio of the data, the location on the sky, and the objectives in fitting the data. If you are only interested in determining the residual SPF emission in order to do imaging, then this simple model may be appropriate. The absorbing column density can usually be frozen at the measured value. The index of the unresolved extragalactic background should be frozen. If you've been careful in correcting for the point source detection limit, the normalization of the extragalactic background can be frozen.

Note that all of these components are linked across the different instruments (and missions). Thus, there are *only* 18 active parameters<sup>18</sup> from this part of the model.

*Instrumental Lines:*

```
const*const*(gaus+gaus)
```

Here the two constants are the same as for the previous model, and should be linked to the constants in the cosmic background model. The lines are described by their central energy, width, and normalization. Unfortunately the effective central energy is not constant across the different EPIC instruments due to small shifts in the gain. The normalizations vary over the different EPIC instruments. We find it a good idea, for the initial fit, to freeze the central energies at their expected values (1.486 keV for Al  $K\alpha$  and 1.739 keV for Si  $K\alpha$ ) and to freeze their widths to 0.0. Once a reasonable value has been found for the normalizations, the line centers and widths should be thawed. Note that there is no Si  $K\alpha$  line for the pn.

*Solar Wind Charge Exchange Lines:*

```
const*const*(gaus+gaus+gaus+gaus)
```

Here the two constants are the same as for the previous model, and should be linked to the constants in the cosmic background model. Here all the parameters are linked between instruments. However, you will note that I have not included a SWCX component for *ROSAT*; the bulk of the charge exchange in the *ROSAT* All Sky Survey was magnetospheric, and has been removed and the remainder is heliospheric and likely included in the LHB emission. Most users will probably not want to bother with the SWCX component when worrying about imaging analyses; it is usually small. However, there are examples where there

<sup>18</sup>You thought I miscounted? The extra one comes from the fact that the *XMM-Newton* data and the *ROSAT* data have different point source detection limits, and thus may need a different normalization for the extragalactic power law.

are strong mismatches in the soft surface brightness between two overlapping observations. These are the cases where the possibility of SWCX should be checked in the spectra.

We’ve only fitted four charge exchange lines here. In principle, one could fit several more. In all cases, one should freeze the line centers to the expected values and the line widths to zero. Once a good fit has been obtained, then one can see if the energies need to be modified slightly, and in a self-consistent way.

*Residual Soft Proton Flare Emission:*

```
const*const*(bknpow)
```

The constants here are not actually needed since we are not using a real response matrix. However, I like using them in order to get a good feel for the relative amount of emission per square arcminute for the different instruments.

We do see differences between the two MOS detectors, both in normalization and spectral shape. Thus, the parameters for the two MOS observations should not be linked. However, we admit that we often do link them for the initial fit, and then separate the two instruments once we are in the vicinity of a good fit.

Similarly, the pn “sees” a very different spectral shape than the MOS detectors, so it can’t be linked to the MOS. However, we are entering a bit of *terra incognita* here: we do not have extensive experience (yet) as to the relative behaviors of pn with PATTERN==0 and pn with PATTERN<=4. We expect that both the normalizations and the spectral shapes will be similar, but we have not fitted enough cases to know to what extent this isn’t true.

### 6.7. Xspec Fits of Multiple Regions: Clusters

Quite often analysis will require the simultaneous fits of spectra from multiple regions, such as cluster radial annuli (e.g., Figure 20). In the Abell 1795 radial profile there are 10 regions for a total of 30 EPIC spectra plus the one RASS spectrum. With 58 parameters per spectrum and 31 spectra, that makes a total of 1798 parameters. With so many parameters, the aim again is to link or freeze as many as possible in order to reduce the number of free parameters. As with the full-field spectra, while the normalizations for the MOS Al K $\alpha$  and Si K $\alpha$  lines and the pn Al K $\alpha$  and Cu lines will vary with annulus, the energies and widths will not and therefore can be linked. The cosmic background parameters can also be linked for all spectra providing the region solid angle scale factors have all been set correctly. The cluster emission component and absorption should be the same for the same annuli from the three EPIC instruments, and so can be linked. While the residual SPF contamination may

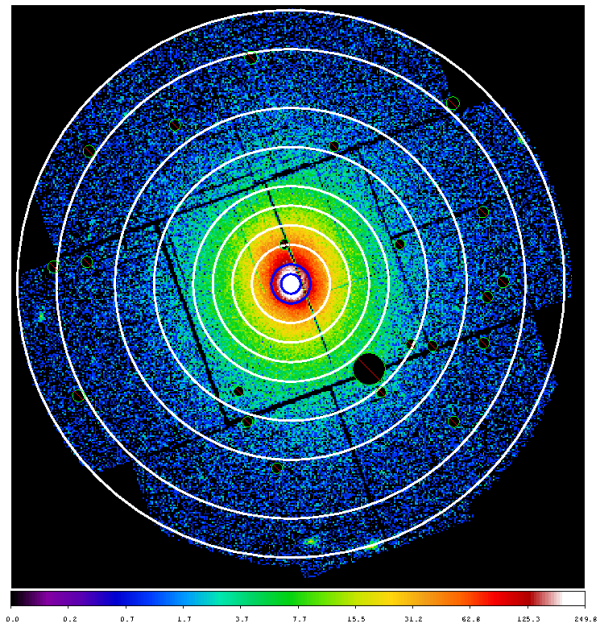


Fig. 20.— Extraction annuli (in white) overlaid on the MOS2 sky coordinate image of Abell 1795. Also shown are the regions (in green with a red slash) excluded because of point sources.

not have a constant spectrum over the field of view, as a first approximation it can be assumed to be so. Thus the spectral parameters (the spectral indices and break energy for the broken power laws) can be linked (the MOS detectors together). However, the distribution over the CCDs is not uniform and the tool *proton\_scale* can be used to generate appropriate scale factors. These scale factors should all be fixed to their appropriate values. When running *proton\_scale*, three numbers are printed out. The last of these (“Scaled Ave Flux”) is the normalization for the soft proton events for that region and instrument. Pick one of your regions and then scale all the others to it. The scaling is just the ratio of that region’s normalization to the normalization of the chosen region. The scale factors for the SPF contribution should be used to link all of the regions to one active parameter (in Xspec, `newpar B = A * F` where the B is the parameter number to be linked, the A is the parameter number of the active parameter, and F is the scaling between the two parameters). Normalizations for the SPF contamination, Gaussian lines, and cluster emission should be frozen at 0 for the RASS data and the solid angle normalization should be frozen to 1 (the units of the RASS spectrum provided by the X-ray background tool is in units of arcmin<sup>-2</sup>). *proton\_scale* can be run in two modes, either on individual spectra with command line input or on groups where the spectra are identified in an ascii file.

With these linkages the number of parameters that

are fit are already reduced from  $\sim 1000$  to about 80, but we are not done yet. The Galactic absorption in the cosmic background component can be frozen to the Galactic column (also provided by the HEASARC X-ray Background Tool) and the extragalactic power law index can be frozen to the canonical value ( $\alpha \sim 1.46$ ). If the solid angle scale factor has been set in units of square arc minutes, the extragalactic power law normalization can be set to its canonical value as well ( $8.88 \times 10^{-7}$  in Xspec units, equivalent to a normalization of  $10.5 \text{ photons keV cm}^{-2} \text{ s}^{-1} \text{ sr}^{-1}$ ). However, when point sources have been removed the normalization should be adjusted. The XMM-ESAS package provides the task *point-source* which will calculate the scale factor using several different models and user-selected parameters. The redshift for the cluster in the various annuli should also be the same so it can be linked. The number of free parameters for the initial fit attempt is still too large for an easy convergence so a few more parameters can be temporarily frozen. The temperatures of the cosmic components, the energies and widths of the Gaussian instrumental components, and the redshift of the cluster if it is known. This can reduce the number of free parameters to about 55.

After an initial fit some of the frozen parameters should be thawed. These include the energies and widths of the instrumental Gaussians (note, however, that if the source emission is bright relative to the Gaussians the width should remain frozen). It may also become clear that other parameters can be linked, but this is where scientific intuition comes into play. In this case, there are insufficient statistics to significantly constrain the fitted abundances in the outer annuli of the cluster so they can be linked to those of inner annuli successively until significance is achieved. The upper panel of Figure 21 shows the best fit for the ten extracted annuli from the Abell 1795 observation. Even after Xspec has converged in fitting the data, it may have only found a local minimum. This will often become apparent when the parameter confidence interval is being determined (e.g., using the Xspec tool *steppar*). So the fitting process will need to be redone to reflect the better fit.

### 6.7.1. Constraining the Cosmic Background

The RASS spectrum constraint can significantly improve the reliability of the fits. The middle panel of Figure 21 shows how the initial fit (upper panel) with only the *XMM-Newton* data is a poor representation as it significantly under predicts the cosmic background at lower energies for the RASS data. To do so, and to produce the apparently good fit of the upper panel, power has been shuffled between model components. Using the RASS data to constrain the fit forces the power to be shifted to (hopefully) the proper compo-

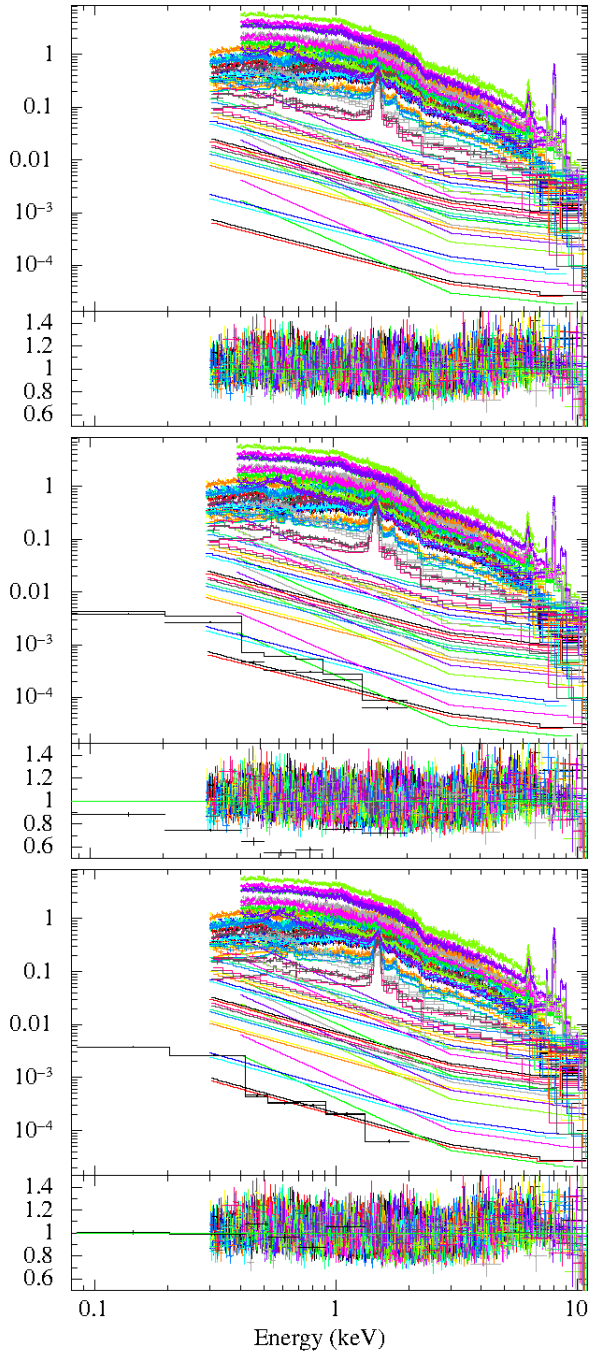


Fig. 21.— Fitted EPIC spectra (MOS1, MOS2, and PN) spectra from ten annuli covering the field of view of the Abell 1795 observation. Upper Panel: The fit with the model background subtracted and the ratio between the data and the fit. Middle Panel: The same as the upper panel with the addition of the RASS spectrum. The data have not been refit and the plot shows how the *XMM-Newton* data by themselves may not constrain the lower temperature components of the model cosmic background. Lower Panel: The plot shows the result when the RASS data are included in the spectral.

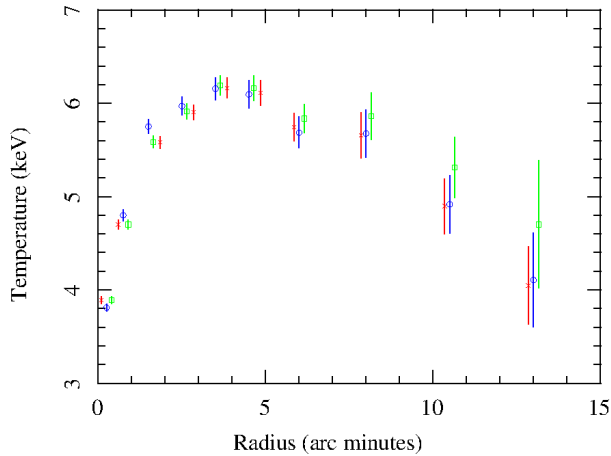


Fig. 22.— Radial profiles of the fitted temperatures for the annuli of the Abell 1795 cluster. Green points are from the fits to the data without adding the RASS data as a constraint. Red points show the fitted values when the RASS data are simultaneously fit. The annuli used are the same but were given shifted radii for clarity. The blue points are from refitting the data applying the crosstalk ARFs to account for flux from one region of the sky which due to the finite PSF appear in a different region of the detector.

nents, as is shown in the lower panel of Figure 21.

Figure 22 shows how the application of the RASS constraint can affect the fitted parameters. It shows the fitted values for the temperature radial profile of Abell 1795. As expected near the center there is very little difference since the cluster emission so dominates the background. However, at large radii where the cluster emission isn’t so bright, the fitted values for the temperature start to vary significantly.

The final background component that can play a significant role in spectral analysis is the solar wind charge exchange flux. As noted above, this is particularly problematic as in the *XMM-Newton* energy range it is primarily comprised of emission lines that are typically of astrophysical interest, e.g., O VII, O VIII, Ne IX, and Mg XI. Also as noted above, occasionally the presence of SWCX emission can be seen by comparing the 0.5 – 0.7 keV and 2.0 – 8.0 keV light curves and searching for variations in the former that are not seen in the latter. However, intervals as long as 30 ks have been observed where the SWCX contamination is significant and yet the light curve has remained relatively constant. Another symptom of the contamination is the detection of very strong O VIII and Mg XI lines. However, as there are astrophysical objects that also show enhanced emission in these lines some care must be taken with the interpretation of the results. In some cases it may be advantageous to include additional Gaussian lines for the O VII, O VIII, and perhaps other lines.

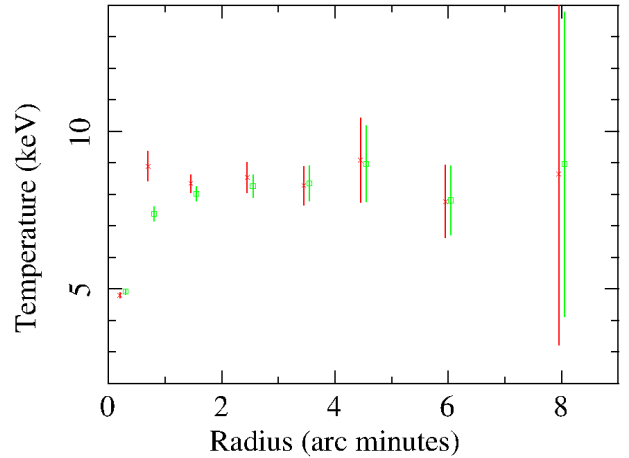


Fig. 23.— Radial profiles of the fitted temperatures for the annuli of the Abell 2205 cluster. Green points are from the fits to the data without accounting for the finite PSF. The red points are from refitting the data applying the crosstalk ARFs to account for flux from one region of the sky which due to the finite PSF appear in a different region of the detector.

### 6.7.2. Crosstalk

Because of the finite PSF of the *XMM-Newton* mirrors and their broad wings a significant flux that originates from one area of the sky is detected in a different area of the detector. If the source is uniform over the field then this effect isn’t a major consideration, however if there are strong gradients in the emission and spectral parameters over the field then the effect can be significant. This is the case for many cooling flow clusters. SAS now includes a modification of the task *arfgen* to account for this “crosstalk”. The blue points in Figure 22 show the effect of the application of this modification to the determination of the temperature radial profile of Abell 1795. The effect is as should be expected with the central temperature being colder and the other inner annuli being hotter when crosstalk is included (the effect of the PSF spreading of the on-average cooler photons has been removed).

Figure 23 shows similar results for the cluster of galaxies Abell 478 where the effect is much stronger.

Like with the model SPF component, the crosstalk contribution must be input as a separate model but its components will be linked to existing parameters. In the example below, the crosstalk model represents the contribution of the spectrum which originates in the 0.5’ – 1.0’ annulus on the sky to the central 0.5’ region of the cluster on the detector. The lines below associate the model parameters of the cluster emission from the annulus with the spectrum and RMF of the central region and the cross talk ARF. Specifically they link the spectral parameters of the cluster annulus thermal emission crosstalk.

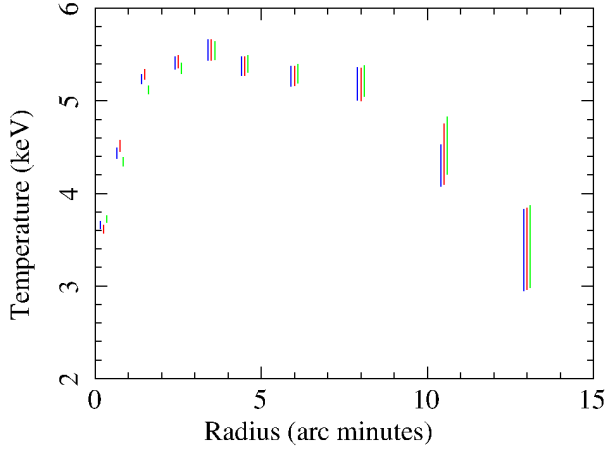


Fig. 24.— Radial profiles of the fitted temperatures for the annuli of the Abell 1795 cluster. Note that these results are from the CCF of 11 September 2008 and SAS V8 that are responsible for the lower values for the fitted temperatures than seen elsewhere. The green data are from the fits performed without accounting for the finite PSF. The blue data are from applying the cross-talk ARFs to account for flux from one region of the sky which due to the finite PSF appear in a different region of the detector. The red points are from applying the cross-talk arf calculation to the individual annuli.

```

resp 4:1 mos1S001-0-30.rmf
arf 4:1 mos1S001-30-60-0-30.arf
model 4:mycross1 con*apec*wabs
newpar mycross1:1 = 64
newpar mycross1:2 = 80
newpar mycross1:3 = 81
newpar mycross1:4 = 82
newpar mycross1:5 = 83
newpar mycross1:6 = 84

```

The *arfgen* crosstalk calculations can also be used to improve the ARF calculations for individual regions. Run in its nominal mode using the extended source parameter turns off the encircled-energy calculation that for small regions can generate a significant over estimate of the effective area. Figure 24 shows the effect of using the *arfgen* cross-talk mode to calculate the ARFs for the individual annuli. (When the annuli are larger in area the effect is minor so for this test only the first two annuli were recalculated.) The effect is to enhance the cross-talk correction, which for a cooling-flow cluster is to lower the fitted temperature of the inner region and increase the fitted temperatures of the inner annuli. Care must be taken in choosing the bin sizes of the detector maps. If the binning is too coarse the ARF will be underestimated, which is a known feature of the method for calculating the cross-talk ARFs and not a bug. Figure 25 shows ARFs calculated for the innermost region using different bin sizes. A bin size of 1.5'' produces close to the limiting value for the

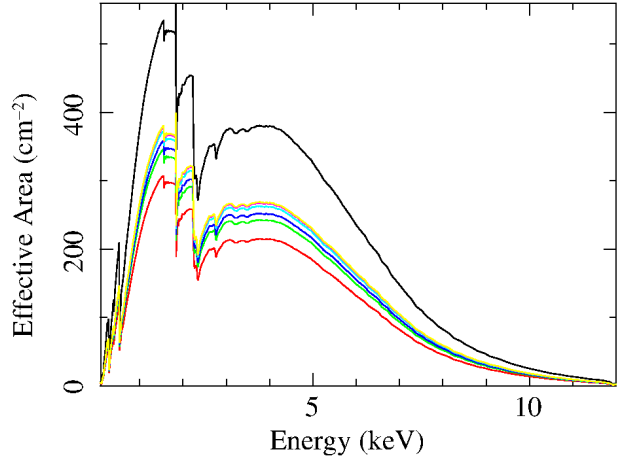


Fig. 25.— Calculated ARFs for the Abell 1795 cluster. The top curve is the ARF calculated using the standard *arfgen* method while the rest of the ARFs were calculated using the cross-talk mode of *arfgen* using different binnings for the detector maps. From the bottom the bin sizes were 7.5'', 5.0'', 3.75'', 2.5'', 1.5'', and 1.0''.

ARF, as does a bin size of 2.5''. In general the bin size should be chosen so that the receiving region should have at least 300 pixels.

## 7. IMAGE PRODUCTION

At this point, we have produced all of the components required to construct a background subtracted image. All of the difficult work has been done. Now we need to 1) combine all of components across the different instruments (the work of *combimage*) and 2) combine the different components (the work of *binadapt*).

### 7.1. Particle Background

If selected by setting `withimages=yes` and entering non-zero values for `elow` and `ehigh`, *mosback* and *pback* also produce particle background images in detector coordinates for the specified energy band. Figure 26 (upper panel) shows the MOS1 particle background image for the Abell-1795 observation for the 0.4 – 1.25 keV band in detector coordinates. The distribution is relatively flat although there is one noticeable feature: the CCDs tend to have different apparent brightnesses. This is primarily due to the differences in total FWC exposure for each CCD and therefore the total number of counts for a given CCD (the counts are normalized by exposure). MOS1 CCD 6 is brightest because of its loss earlier in the mission. MOS1 CCD 4 and MOS1 CCD 5 are also slightly brighter than the rest due to having lost exposure due to time spent in the anomalous state.

In general, images in detector coordinates are not particularly useful; one needs them in sky coordinates. Transformation of the detector coordinate images into sky coordinate images is done simply with *rotdet2sky*. Here is an example where we transform a QPB background image:

```
rotdet2sky intemplate=mos1S001-fovimsky-350-1100.fits
          inimage=mos1S001-bkgimdet-350-1100.fits
          outimage=mos1S001-bkgimsky-350-1100.fits
          withdetxy=false withskyxy=false
```

The output images, `x-bkgimsky`, are in sky coordinates with the same projection and pixel size as the source images `x-fovimsky-350-1100.fits` produced by *mosspectra* and *pnspectra*.

The parameters for *rotdet2sky* are the name of the image whose coordinate system is to be matched, `intemplate`, the name of the input image, `inimage`, the name of the output image `outimage`. The `withdetxy=false` and `withskyxy=false` are probably not necessary. Figure 26 (lower panel) shows the MOS1 image after casting into sky coordinates which requires a reflection, an offset, and a rotation. As will be seen below, the *proton* and *swcx* routines produce output images in detector coordinates; *rotdet2sky* can be used to rotate these images as well.

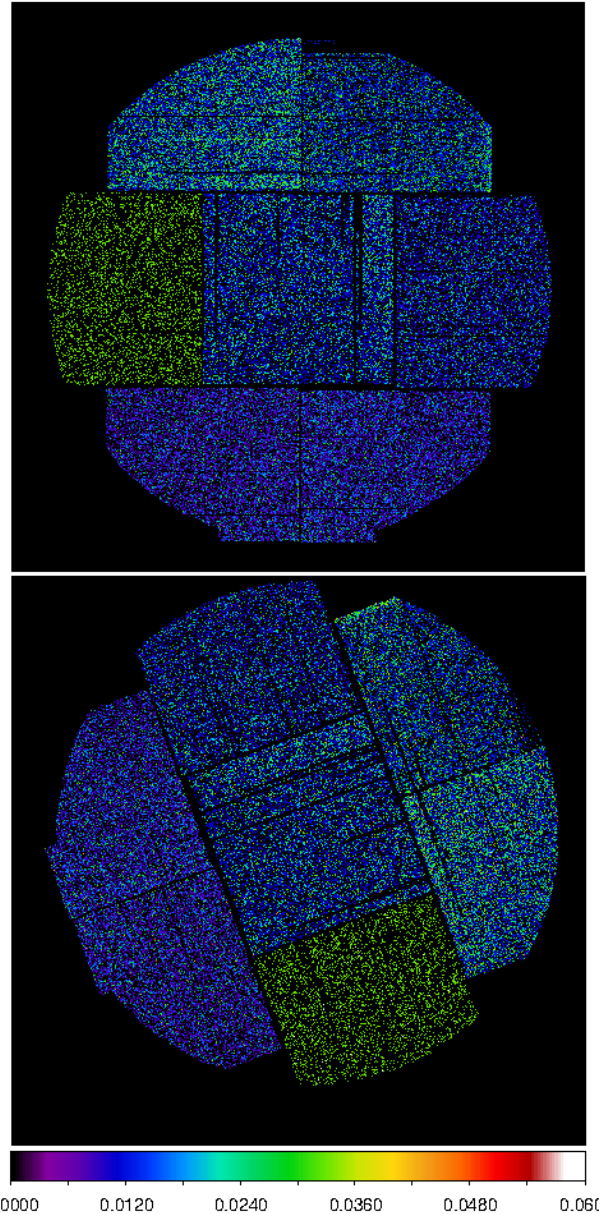


Fig. 26.— Images of the model particle background in detector coordinates (upper panel, the output of *mos\_back*) and in sky coordinates (lower panel, output of *rotdet2sky*). The differences in the general colors of the events in the different CCDs are due to differences in the FWC data exposure times. For a CCD with less FWC exposure there are fewer sample events, and therefore each event must have a greater weighting.

### 7.2. Soft Proton Background

Many, if not most, observations have some residual SPF contamination after light curve screening. To determine the level of the residual contamination, if any, and to remove it first requires spectral fitting of the data. In the spectral fitting process, a power law (or

broken power law) which is not folded through the instrumental effective areas is added to the model. See §6.3 and §6.6.

After the spectral parameters for the SPF contamination have been derived, the task *proton* can be run. If there is no additional normalization applied to the Xspec model, the SPF spectral parameters as displayed by Xspec for the full FOV (normalization: counts  $\text{keV}^{-1} \text{s}^{-1}$ , power law index: the negative photon index) can be input into *proton*. The *proton* routine produces an image in detector coordinates. The following is an example of a call of *proton* and *rotdet2sky* for the MOS:

```
proton imagefile=mos2S002-fovmdet-350-1100.fits
      specfile=mos2S002-fovft.pi elow=350 ehigh=1100
      speccontrol=1 pindex=0.1 pnorm=2.208e-2
```

or

```
proton imagefile=mos2S002-fovmdet-350-1100.fits
      specfile=mos2S002-fovft.pi elow=350 ehigh=1100
      speccontrol=2 bindl=0.1 bbreak=1.0
      bindh=1.0 bnorm=2.208e-2
```

where `imagefile` is the image for the appropriate detector in detector coordinates. `specfile=mos2S002-fovft.pi` provides a file for the extraction of the EXPOSURE keyword, and `must` be the spectrum used in the spectral fits where the magnitude of the residual SPF contamination was determined. The values of `elow`, and `ehigh` are as defined above and must be the same values as used in *mosspectra*, *pnspectra*, *mosback*, and *pnback*. The `spectrumcontrol` flag controls the spectrum mode: `spectrumcontrol=1` for a power-law spectrum and `spectrumcontrol=2` for a broken power-law spectrum. `pindex` is the power-law index and `pnorm` is the power-law normalization taken directly from the Xspec fit for entire FOV for which the image is to be created. `bindl`, `bindh`, `bbreak`, and `bnorm` are the low and high energy spectral indices, the break energy (in keV), and the broken power law normalization taken from the spectral fit.

Figure 27 (upper panel) shows the MOS1 SPF background image for the Abell-1795 observation for the 0.4–1.25 keV band in detector coordinates. *rotdet2sky* is run to convert the image from detector to sky coordinates (shown in Figure 27, lower panel).

As for the particle background maps, after *proton* has been run, *rotdet2sky* must be run to recast the SPF image from detector coordinates to sky coordinates.

```
rotdet2sky intemplate=mos2S002-fovimsky-350-1100.fits
```

```
inimage=mos2S002-protimdets-350-1100.fits
outimage=mos2S002-protimsky-350-1100.fits
```

### 7.2.1. Transfer from One Region to Another

In cases where there is strong emission from the extended source, as there can be for clusters of galaxies, the fitted parameters for the SPF component can be significantly over or under estimated. In such cases it can be helpful to fit a spectrum extracted from the lower surface brightness regions in the field. For clusters of galaxies an outer annulus in the FOV can serve this purpose. However, in this case, the fitted normalization must be scaled from the limited region to the full FOV, which can be done by the routine *sppartial*:

```
sppartial fullimage=ffov/mos1S001-fovimsdpdet.fits
      fullspec=ffov/mos1S001-fovft.pi
      regimage=annu/mos1S001-fovimsdpdet.fits
      regspec=annu/mos1S001-fovft.pi
      regnorm=0.03
```

where `fullimage=ffov/mos1S003-sp-full.fits` is the SPF image template for the full FOV, `fullspec=ffov/mos1S003-obj-all.pi` is the spectrum for the full FOV, `regionimage=annu/mos1S003-sp-ann.fits` is the SPF template image for the restricted region, `regionspec=annu/mos1S003-obj-ann.pi` is the spectrum for the restricted region, and `rnorm=0.03` is the fitted SPF normalization for the restricted region. (Here I've assumed that the extraction from the full FOV has ended up in a subdirectory named `ffov` while the extraction from the partial FOV has ended up in a subdirectory named `annu`.) In the *proton* call use the fitted spectral index from the restricted region and the scaled value for the normalization.

### 7.3. Solar Wind Charge Exchange Background

Table 8: Common SWCX Emission Lines.

Line	Energy (keV)	In <i>swcx</i> Call
C VI	0.37	not yet implemented
C VI	0.46	not yet implemented
O VII	0.57	OVII
O VIII	0.65	OVIII
O VIII	0.81	OVIIB
Ne IX	0.92	NEIX
Ne IX	1.02	not yet implemented
Mg XI	1.35	MGXI



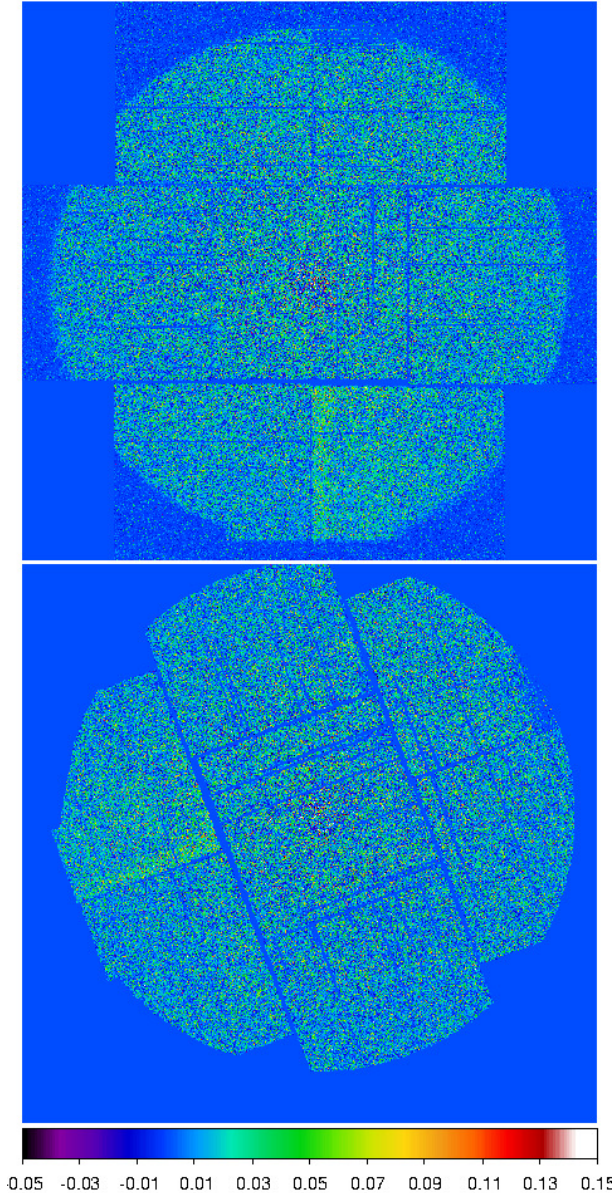


Fig. 27.— Images of the model SPF background in detector coordinates (upper panel, the output of *proton*) and in sky coordinates (lower panel, output of *rotdet2sky*). Note that values are both negative and positive due to the method for creating the instrument maps. On average the values are positive and the count images are sparse requiring averaging on angular scales where the SPF values provide good information.

Solar wind charge exchange (SWCX, see §6.4) emission, more so than the soft proton background, probably affects all observations. Also like the SPF background, the level of the background is not known *a priori* and must be modeled from the data. For sources like the soft X-ray background which cover the entire instrumental FOV, this can be very problematic as

there is no method to distinguish a SXR X-ray from a SWCX X-ray. Again, like the SPF background, an estimate of the level of SWCX background can be derived from spectral fitting and then subtracted from images. In cases of diffuse emission (e.g., most clusters of galaxies) which does not cover the entire FOV a more robust estimate can be made by explicitly fitting SWCX lines in the on- and off-emission spectra.

The solar wind charge exchange (SWCX) component is treated in a similar manner as the soft proton background. Scale factors are determined during the spectral fitting process for a user-determined number of emission lines that are then used to create model count images. **Unlike proton, where the normalization is for the full FOV, here the normalizations are per square arcminute.** The *swcx* routine is called:

```
swcx imagefile=mos2S002-fovmdet-350-1100.fits
specfile=mos2S002-fov.t.pi elow=350 ehigh=1100
rmffile=mos2S002.rmf arffile=mos2S002.arf
lines="OVII" gnorms='1.83305e-7'
```

where *imagefile* is the image for the appropriate detector in detector coordinates. *specfile=mos2S002-fov.t.pi* provides a file for the extraction of the EXPOSURE keyword, and *must* be the spectrum used in the spectral fits where the magnitude of the SWCX contamination was determined. *objrmf=mos2S002.rmf* and *objarf=mos2S002.arf* are the response and redistribution matrix files. The values of *elow*, and *ehigh* are as defined above and must be the same values as used in *mosspectra*, *pnspectra*, *mosback*, and *pnback*. *linelist='OVII OVIII'* sets the energies for the SWCX lines which are being modeled and subtracted. *gnormlist='4.38E-07 3.63E-07'* are the fitted Xspec normalizations for the SWCX lines *where the line normalizations are fitted per arcmin<sup>-2</sup>*. Figure 28 (upper panel) shows the MOS1 SWCX background image for the Abell 1795 observation for the 0.4 – 1.25 keV band in detector coordinates. As for the particle background and SP maps, *rotdet2sky* is run to convert the image from detector to sky coordinates (Figure 28, lower panel).

#### 7.4. Stray Light

The *XMM-Newton* mirrors have a small sensitivity to stray light originating just out of the field of view (0.4–1.4 degrees) due to un baffled single reflections (two reflections are necessary for X-ray focusing). In most cases the effect has little significance, however, if a bright source is located just outside of the FOV then the effect can be spectacular. This can be seen in the LMC data where LMC X-1 produces spectacular arcs in neighboring pointings. There is nothing that can be done to model and subtract this

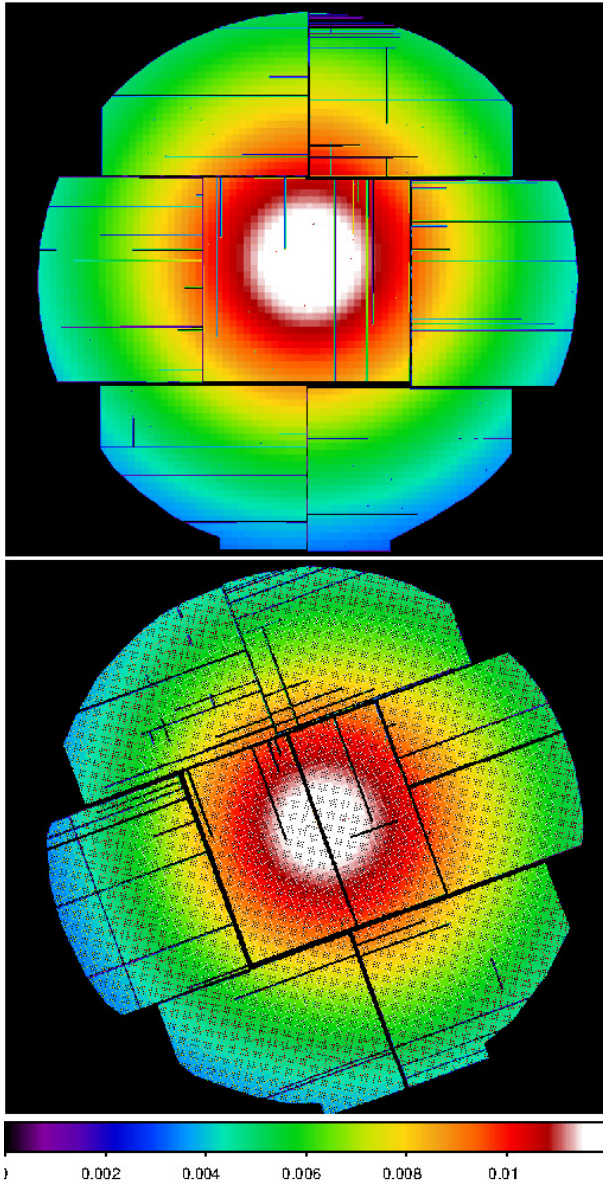


Fig. 28.— Images of the model SWCX background in detector coordinates (upper panel, the output of *swcx*) and in sky coordinates (lower panel, output of *rotidet2sky*). The images are very similar morphologically to the exposure maps as they are produced by simply scaling exposure maps calculated for the SWCX energies. The fine structure in the sky image is due to the transforming where occasionally two original pixels get cast into a single transformed pixel, and sometimes none do. However, for any reasonable pixel binning or smoothing this effect is washed out.

additional background component, and excision of the affected region is the only option. Extended sources, like the SNR N132D in the LMC can complicate the identification of stray light as the clear arcs due to a point source become blurred and merge into a bright emission region. Figure 29 shows the effect from a point source for a single exposure and instrument. As

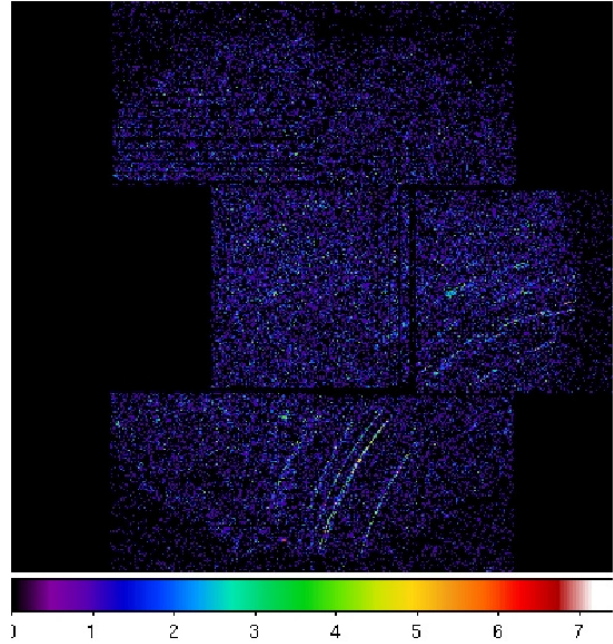


Fig. 29.— MOS1 count image showing the stray-light arcs produced by a bright point source outside of the FOV.

demonstrated in Figure 30, the effect of stray light can be considerably stronger in the hard band when the source is hard but the surrounding diffuse emission is relatively soft.

### 7.5. Masking

One of the more vexed issues when reconstructing ESAS completely within the SAS structure, was the handling of mask files. In practice one usually wants to detect sources over a rather broad band, in order to capture the fainter sources, but usually wants to examine the diffuse emission in some other, narrower band. Thus, the “cheese” masks created early in the process probably are not in the same energy band as the images that you are creating at this step. Thus, the names of the “cheese” masks are not going to fit easily in to the `P-root-elo-ehi.fits` structure, and so we didn’t try to do that. What we have done is make sure that the “cheese” masks have the same coordinate frame as the other components.

One other thing we wanted avoid was requiring long lists of input file names. Thus, at this point, one will have to choose the appropriate mask files, and rename them to follow the `P-root-elo-ehi.fits` structure. We don’t care what you use for “root”, but the prefix and the `-elo-ehi.fits` need to match the other component names you’ve created already. This is a bit of a pain, but it saves greater pain elsewhere.

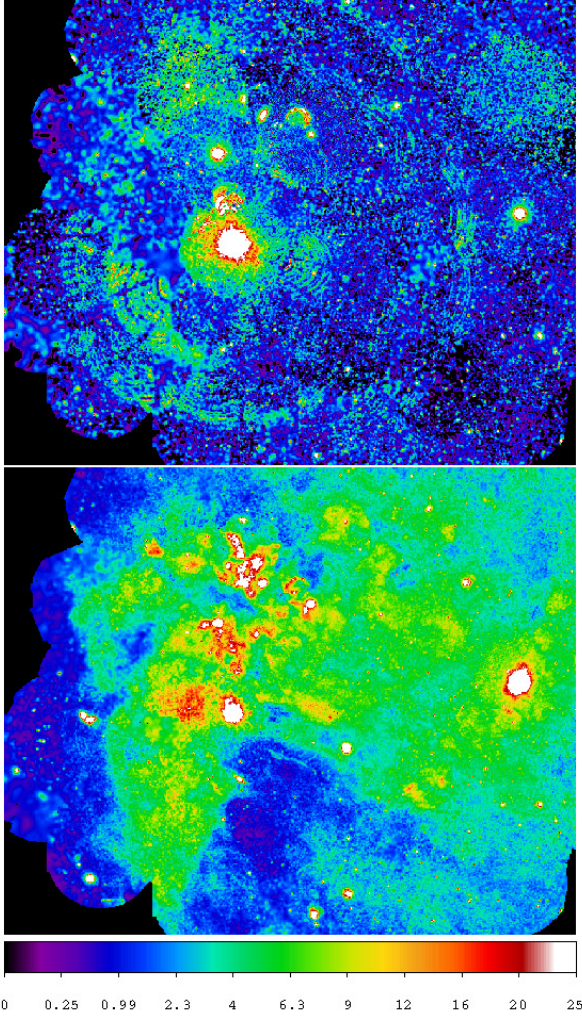


Fig. 30.— Images of the LMC X-1 region in the 2.0 – 7.2 keV band (upper) and 0.4 – 1.25 keV band (lower). The stray light from LMC X-1 is relatively much stronger in the hard band.

## 7.6. Image Creation

After *rotidet2sky* has been run for the QPB, and either or both of the SPF and SWCX backgrounds, all of the primary components for a background-subtracted and exposure-corrected image are ready.

Before a final image is produced, it is reasonable to combine the MOS1, MOS2, and pn data if more than one data set exists. This can be done using the task *combimage*, which simply adds the individual components from the three instruments with proper scalings for different filters and different instruments, and even from multiple exposures from the same ObsID. For example:

```
combimage prefixlist="1S001 2S002 S003"
withpartbkg=true withspbkg=true withswcxbkg=true
```

```
withcheese=true cheesetype=t
elowlist=350 ehighlist=1100 alpha=1.7
```

`prefixlist="1S003 2S004 S005"` is a list of the prefixes (exposures) to be combined, `withpartbkg=true` controls whether QPB images are processed, `withspbkg=true` controls whether SPF images are processed, `withswcxbkg=true` controls whether SWCX images are processed, `withcheese=true` and `cheesetype=t` controls the source masking, `elowlist=400` is a list of the energy band lower limits, and `ehighlist=1250` is a list of the energy band upper limits. If multiple bands are combined the resultant output files are labeled with the lowest low energy and the highest high energy. The specified energy bands should be ordered by increasing energy.

The task *combimage* scales the data sets to the response of the MOS2 medium filter by using a look-up table to scale the exposure. The look-up table was created using PIMMS assuming an  $E^{-\alpha}$  power law spectrum with  $2 \times 10^{20}$  H I  $\text{cm}^{-2}$  absorption. The assumed value for  $\alpha$  is set by the parameter `alpha=1.7`. Combining MOS and pn data for objects with radically different spectral parameters (different from the assumed spectrum) may produce artifacts in the final image. The artifacts are most easily seen in the CCD gaps where the surface brightness in the gap, which will be covered by a different instrument (pn or MOS), will be brighter or fainter than the adjacent regions. Varying the input parameter `alpha` may alleviate this effect (which in our experience has been rare).

## 7.7. Image Analysis

At this point the user is left nearly to their own devices. For scientific analysis the data should not be smoothed and, depending upon the analysis needs, binned. A very simple task, *binadapt*, is provided to combine all the components properly to create an image, perhaps with binning. (In the next section we'll deal with the same routine in terms of smoothing.) The output file is in count rate:

$$R = \left( \frac{\sum_{inst} c - [\sum_{inst} b + \sum_{inst} p + \sum_{inst} s]}{\sum_{inst} f(\alpha)e} \right) \quad (1)$$

and count rate uncertainties:

$$\sigma_R = \left( \frac{\sqrt{\sum_{inst} c}}{\sum_{inst} f(\alpha)e} \right) \quad (2)$$

where  $f(\alpha)$  is a scale factor (as a function of the power law index) used to convert the exposure maps from a particular instrument, filter, bandpass combination to the

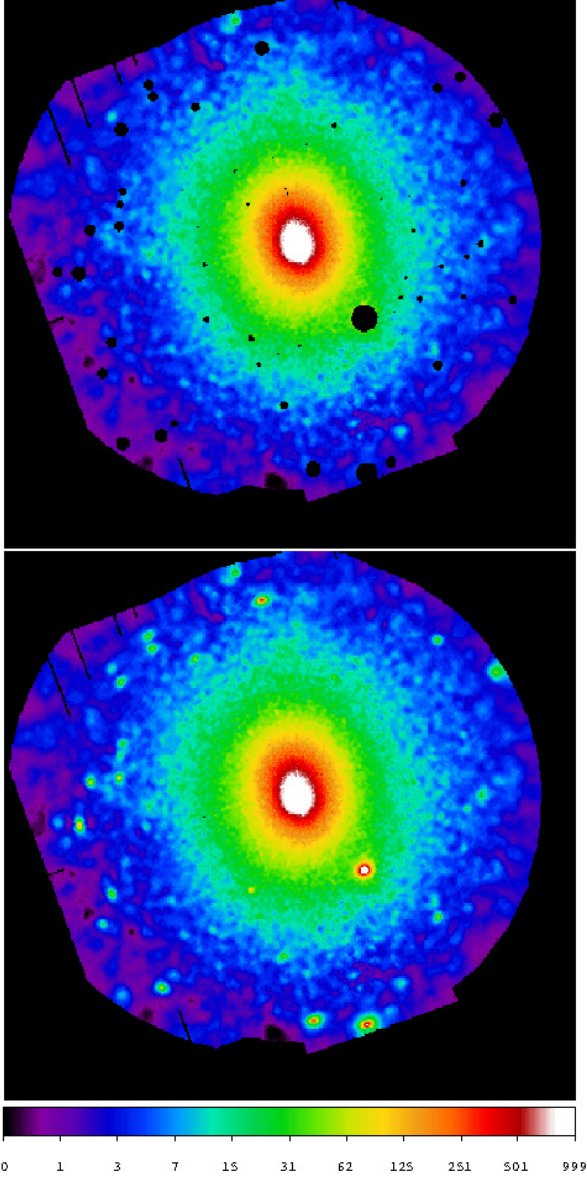


Fig. 31.— EPIC (MOS plus pn) background-subtracted and exposure-corrected image of Abell 1795 in the 0.4 – 1.25 keV band. The image shows the results with a logarithmic scaling if point sources are removed. The size of the removed regions depends both on the PSF at the location of the source as well as the strength of the source.

same bandpass for the MOS2, Medium filter combination.

to units of counts  $s^{-1} \text{ deg}^{-2}$ , and is equal to the reciprocal of the fundamental analysis pixel size of  $2.5'' \times 2.5'' (= 2.0736 \times 10^6)$ .  $c$  is the number of counts,  $b$  is the number of model particle background counts,  $p$  is the number of model SPF counts, and  $s$  is the number of model SWCX counts). The assumption is that the counting statistics of the observation

dominates the uncertainty in the model background counts. In practice this probably underestimates the uncertainty which would make any analysis results more conservative. The combination of total counts, backgrounds, exposure, and mask is done with *binadapt*.

```
binadapt prefix=comb elow=400 ehigh=1250
withpartbkg=true withspbkg=true
withswxcbkg=true withbinning=false
withsmoothing=false
```

As might be divined from the above, *binadapt* can be used for a single detector (for which the *prefix* would be something like *mos1S001*), or for the combination of any of the three EPIC detectors (in which case the *prefix* is *comb*). The *elow* and *ehigh* are used to construct the input file names. The *withpartbkg*, *withspbkg*, and *withswxcbkg* flags can be set to include or exclude the various backgrounds as required. Even if a mask is not specified, one can set a minimum exposure threshold, *maskthresh* to exclude strongly vignettted regions. Of course, an extra mask can be used as well *withmask=true maskfile=* and the name of the mask file.

## 7.8. Image Massaging

Typically there are too few counts in an image, even after combining multiple exposures, to produce a pretty image without smoothing and/or binning. The XMM-ESAS package provides an adaptive filter task, *binadapt*, which will produce a reasonable background-subtracted and exposure-corrected image (a pretty picture).

```
binadapt prefix=comb elow=350 ehigh=1100
withpartbkg=true withswxcbkg=true withspbkg=true
withmask=false withbinning=true binfactor=2
withsmoothing=true smoothingcounts=50
```

The call is mostly as in the previous example, but with a few exceptions. The *smoothingcounts=50* defines the smoothing kernel for the adaptive smoothing, while *binning=2* sets the number of (linear) pixels for binning the image before the smoothing. *maskthresh=0.02* sets the threshold for excluding low exposure regions of the image, the

Figure 31 displays the resultant background-subtracted and exposure-corrected image without (upper panel) and with (lower panel) point-source exclusion. If individual exposures are being smoothed the *prefix* (e.g., *prefix=1S003*) as defined above must be entered. The algorithm uses a conical weighting for the smoothing, and a image is created providing the conical FWHM in arc minutes (Figure 32).

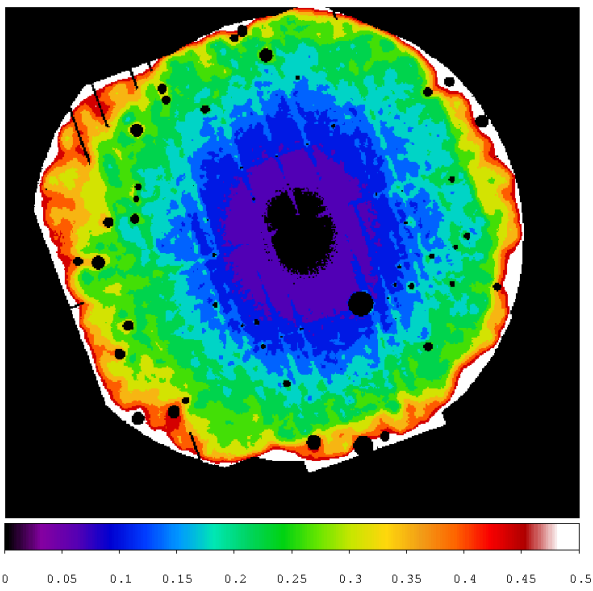


Fig. 32.— Smoothing FWHM for the Abell 1795 image (Figure 31) in the 0.4 – 1.25 keV band. The color table is in units of arc minutes.

## 8. MOSAICS

The XMM-ESAS software package offers users an opportunity to mosaic multiple observations of a region regardless of their pointing directions and orientations. By *mosaics* we mean multiple observations that are relatively close or overlapping on the sky. Mosaics may include mosaic-mode observations, but need not do so. The peculiarities of mosaic-mode observations are addressed in the next section.

The programs described in the previous sections produce components (counts, exposure time, quiescent particle background, soft proton flare background, and SWCX background) for a instrument of an observation. The mosaic merging programs can be run only after all contributing observations have been processed to the point that all the individual images can be produced (i.e., you can't merge what you haven't created). Of course, for some components (e.g., SPF and SWCX) that component may exist for some observations and not for others; this is not a problem.

### 8.1. The Overall Mechanics

For the example below of the mosaicked image of the galaxy M101, the most simple directory structure is as follows:

```
/M101/merge
/M101/0104260101/odf
/M101/0104260101/proc
/M101/0164560701/odf
/M101/0164560701/proc
/M101/0212480201/odf
/M101/0212480201/proc
```

All three observations of M101 are significantly contaminated by SP flaring, 0164560701, however, is a truly pathological case where even after light-curve filtering there remains a considerable amount of residual SP contamination. We use all three observations as an illustrative example. Figure 33 displays the light curves with their accepted time intervals (an output of *espfilt*) for the three observations.

The merging processing will take place in the `/M101/merge` directory. Within the merge directory is an ASCII file listing the paths *and prefixes* of the individual exposures to be merged which is named `mosaic.list`:

```
../0104260101/proc/mos1S001
../0104260101/proc/mos2S006
../0104260101/proc/pnS003
../0164560701/proc/mos1S001
../0164560701/proc/mos2S002
../0164560701/proc/pnS003
```

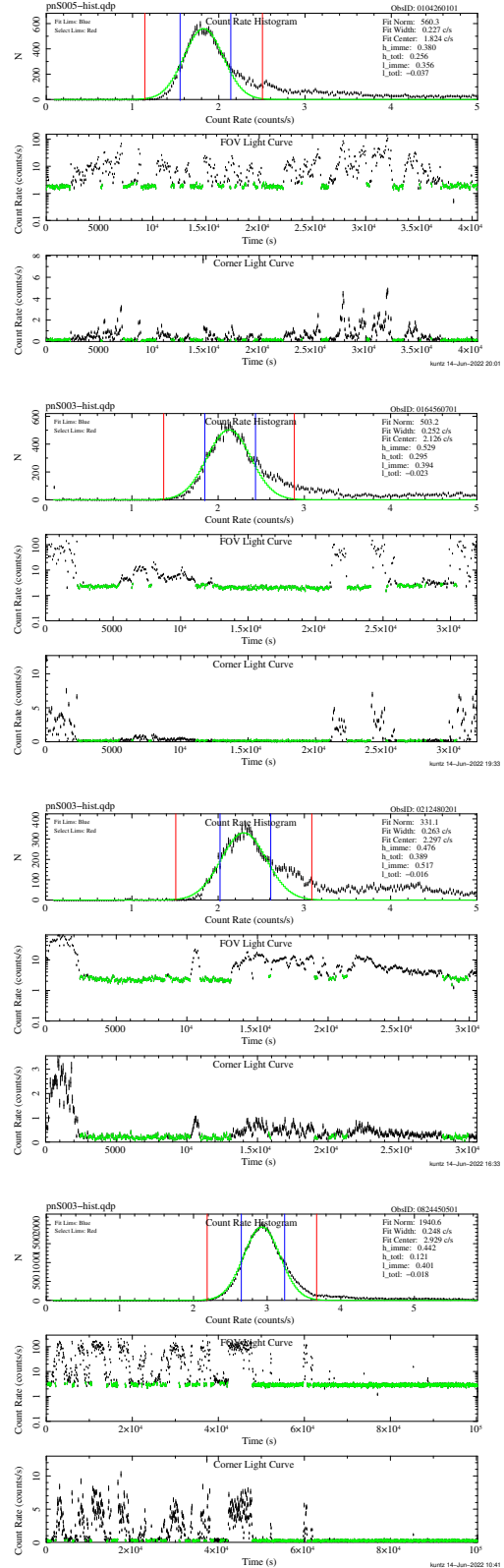


Fig. 33.— Light curves of the M101 observations.

```

../0212480201/proc/mos1S001
../0212480201/proc/mos2S002
../0212480201/proc/pnS003

```

(The appendices use a slightly different structure that is more consistent with the other reduction scripts.) The following shows the simple set of commands to mosaic the exposures processed for the 0.4 – 1.3 keV band to create the image in the top panel of Figure 34:

```

setenv RAPNT 210.8
setenv DECPNT 54.32

```

```

mosaicmerge dirfile=mosaic.list type=1
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'
mosaicmerge dirfile=mosaic.list type=2
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'
mosaicmerge dirfile=mosaic.list type=3
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'
mosaicmerge dirfile=mosaic.list type=4
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'
mosaicmerge dirfile=mosaic.list type=5
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'
mosaicmerge dirfile=mosaic.list type=6
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'
mosaicmerge dirfile=mosaic.list type=7
  coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350
  ehigh=1100 withcheese=yes cheesemasktype='t'

```

In the above example, the individual components for an image of the 0.35 – 1.31 keV (elow=350 to ehigh=1100) band are cast into a mosaic using the task *mosaicmerge*. The coord=2 selects the coordinate system to be used (1 would be ecliptic coordinates, 2 would be equatorial coordinates, and 3 would be Galactic coordinates), and crval1=210.8 and crval2=54.32 are the central coordinates of the mosaic projection. Note that in order to reduce problems with typos, I put the coordinates into environment variables, and then reference the environment variables in each call to *mosaicmerge*.

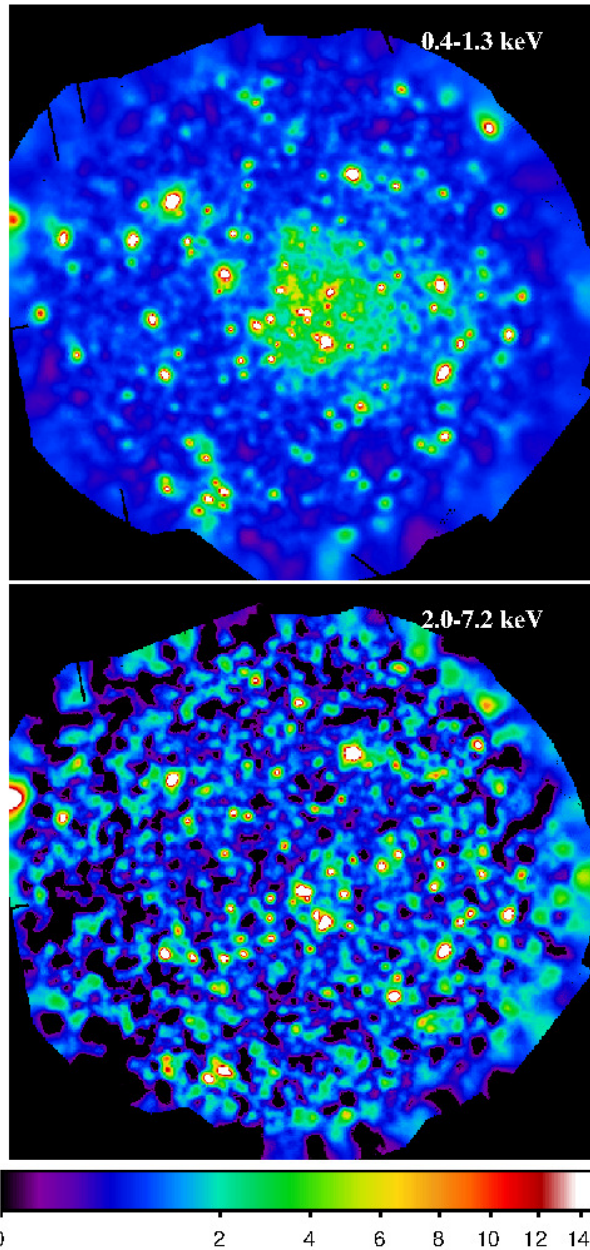


Fig. 34.— Merged images of M101 in the 0.4 – 1.3 keV band (top) and 2.0 – 7.2 keV band (bottom). While the low-energy band looks quite reasonable, the high-energy band shows the effect of uneven residual SP contamination. These images were made with an earlier version of ESAS with 0'03 pixels.

The pixelsize=0.0416 gives the scale size of the projection in units of arc minutes. A few notes on pixel sizes are appropriate. The “native” pixel size for the ESAS routines *cheese*, *mosspectra/back*, and *pnspectra/back* is 2'5=0'041666668. In the above example, I stuck with that pixel size. The arrays into which the mosaic is cast are 2000 × 2000 pixels in size (currently hardcoded), making an image 1°38 × 1°38 which is plenty for M101. Compared to some projects

to which ESAS has been applied, M101 is relatively bright, so this pixel size is not ridiculous. Much fainter objects would not have sufficient counts per pixel to have reasonable signal to noise, and so larger pixels would make sense. Thus the limit on the size of the output image is not (generally) an issue. We can, however, imagine projects for which it could be a problem, and are considering what to do about them.

To build a rate image from the components one uses *binadaptmerge*, which works much like *binadapt*.

```
binadaptmerge elowlist=350 ehighlist=1100
  withpartbkg=yes withspbkg=true withswcxbkg=true
  withmask=true maskfile=mosaic-cheeset.fits
  withbinning=false withsmoothing=false fill=0
```

In this case, the components are combined to create a rate image without binning or smoothing. To bin the output image one sets `withbinning=true` and `binfactor` equal to the linear number of pixels to be binned. To adaptively smooth one sets `withsmoothing=true` and `smoothcounts` to the number of counts to accumulate for smoothing. One can also control the inclusion of partially exposed region by setting `maskthresh` to remove regions where the mask is below some value.

Finally, `fill` controls infilling. If `fill` is not equal to zero the program will loop through the data, find pixels with no data (a blank pixel), and if the blank pixel has at least three adjacent non-blank pixels the blank pixel is set to the average of all of the adjacent non-blank pixels. The parameter controls how many of these passes will be made. This is purely a cosmetic process to make prettier pictures. However the smoothed images shouldn't be used for scientific analysis anyway.

If the processing of the individual observations had been done in two bands which span the energy range, 0.4 – 0.75 keV and 0.75 – 1.3 keV, the data can be combined with `elowlist='400 750'` and `ehighlist='750 1300'`.

The mosaics show the effects of both uneven residual SP contamination and significantly different exposure times over the field. The different residual SP contamination is most easily seen at the right-hand edge of the hard band image. Because there is typically many fewer cosmic background events at higher energies, the particle background is relatively much brighter, and the typical hardness of the SP background, the hard band is much more sensitive to residual uncertainties in the SP background determination. Note also that the higher-intensity crescent in the hard band also has a smoother appearance in both bands. This is an artifact of the adaptive filtering algorithm when the exposure time (see Figure 35), and therefore number of

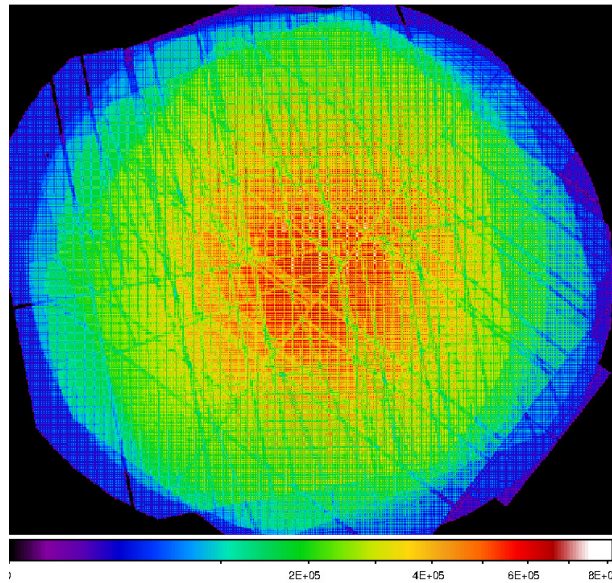


Fig. 35.— Merged exposure map of the M101 observations in the 0.35 – 1.25 keV band.

counts, is relatively non-uniform.

Figure 36 shows the spectacular MOS mosaic of the Coma Cluster in the 0.8 – 1.25 keV band. 28 observations from both MOS instruments are included in the mosaic.

## 8.2. Source Removal Considerations

There are a number of considerations when it comes to point source removal from a mosaic! If you are interested in producing the cleanest possible image of the diffuse emission, then combining the source lists from the individual observations will probably not be satisfactory; even fainter sources below the threshold of a single observation will suddenly become very noticeable once you have stacked several observations together. On the other hand, pushing the point source detection limit as deep as possible will produce a very non-uniform point source removal limit, which may or may not be an issue. So what are our choices?

First, you have the point source lists (and masks) from each individual observation. One can apply to each observation the mask from that observation, and then merge. This has the difficulty that a source detected in one observation, but not detected in the other observations of that location, may be just below the detection thresholds in those observations. After mosaicking, it can be statistically significant and, due to the fact that you've removed that region in the one observation where it was detected, accentuated. On the other hand, if the source is *well* below the threshold in all the observations but one, when the source was flaring, for example, it does not make much sense to remove that region from the observations when it was



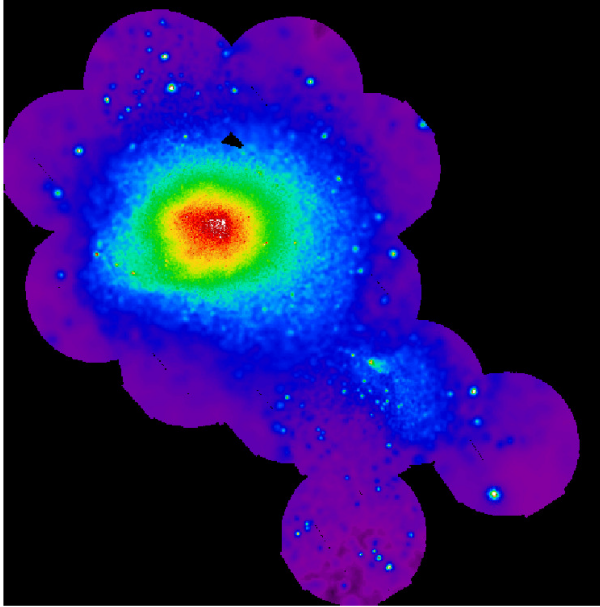


Fig. 36.— Merged image of the Coma Cluster in the 0.8–1.25 keV band.

not detected.

A second option is to produce a list of the unique sources from those lists using *emlfill* and *emlmerge*, and then recreate the source masks for the individual observations from that list of unique sources. **The *emlmerge* routine is undergoing revision and is not currently available.**

A third approach can be taken if the mosaic is not much bigger in area than a single observation. The observations of M101 are a good example. In this case you can *merge* the event files for each instrument over all of the observations, and then use that merged event list for source detection with *cheese*. (Remember that you will also want to merge the *atthk.fits* files as well.) There are a number of caveats!

First, *edetect\_chain*, which is called by *cheese* does not handle strong changes in effective exposure very well, so you may want to fiddle with *edetect\_chain* directly to get the parameters correct. (The *emask\_threshold1* parameter may need to be reduced.)

Second, although this process will produce an *emlhist.fits* file with the deepest possible detection limit in each location, it may not pick up the sources in regions that have low relative exposure. You may need to augment this list with sources from the individual exposures.

Third, this list is not without its perils as well; since it has been constructed from overlapping images with (possibly) very different PSFs at the same point, the probabilities determined by *emldetect* may not be reasonable. And the output list will require a great deal of hand selection! This can be done if you are looking for

a really smooth image, but does not allow one to set a uniform source elimination criterion. In this third approach, you can use the *emlhist.fits* from the merged observations to create masks for the individual obsids (i.e., §5.12.2).

Finally, the SAS routine *emosaicproc* should be considered as well. (We have not tried using it yet.)

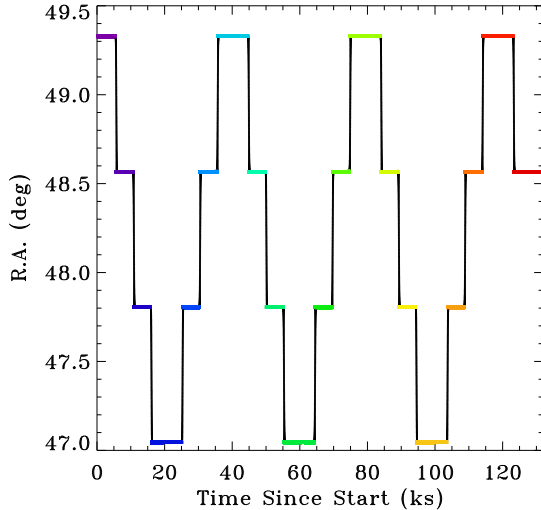


Fig. 37.— The R.A. as a function of time for a mosaic mode observation. The black points are each entry in the attitude information file. The colored points are show the time intervals of the individual pointings as identified by *emosaic\_prep*

## 9. MOSAIC MODE OBSERVATIONS

Mosaic mode observations have been handled by SAS and by ESAS in different ways at different times. This section describes the current situation, but the user should be aware that mosaic mode data processed in earlier epochs may require different handling.

The root of the issue is that, for a given instrument, all of the events from all of the pointings of a given observation are in a single event file with a single projection. Some SAS tasks, such as *edetect\_chain*<sup>19</sup> can handle multiple pointings within an event file. Most ESAS tasks, however, were built with the assumption of a single pointing, e.g., *task\_names*. Thus, to run most ESAS tasks, the multi-pointing event files will need to be split into event files containing only a single pointing, and then changing the world coordinate system appropriately.

In the current regime, a mosaic mode observation will be stored in the archive in two ways. In the first form there is a single ODF and the obsid number for that ODF will, typically, end in 01, as in xxxxxxxx01. After processing with *emchain/epchain* (or, alternately *emproc/epproc*) the event file for each instrument will contain all of the events for all of the  $n$  pointings. We refer to this as a “multi-pointing ODF”. In the second form, there is one ODF for each of the  $n$  pointings, and the obsid numbers will have

<sup>19</sup>Although *edetect\_chain* can handle multiple pointings within a single event file, unless there is overlap between pointings, it is far more efficient to run *edetect\_chain* on each separate pointing.

the same xxxxxxxx root, but will end in 31, 32, ... 31+n. We will refer to these as “individual pointing ODFs”. Both forms are needed for ESAS analysis. The xxxxxxxx01 version is required for the soft proton filtering, while the xxxxxxxx31 version is required for nearly everything else.

Note that processing each individual ODF through *emchain/epchain* requires as much time as processing the multi-pointing ODF.

### 9.1. Soft Proton Filtering

Unless the different pointings have strongly different mean count-rates, it is best to run the soft proton filtering on the multi-pointing ODF. The GTI file created in that process for a given instrument can then be applied to the individual pointings for that instrument, e.g.:

```

evselect table=mos2S003.fits:EVENTS
  withfilteredset=yes
  expression="(PATTERN<=12)
  &&GTI(mosaic_mos2S003-gti.fits,TIME)
  &&((FLAG & 0x766aa000) == 0)"
  filteredset=mos2S003-clean.fits
  filtertype=expression
  keepfilteroutput=yes updateexposure=yes
  filterexposure=yes

```

N'est-ce pas? If there is a significant number of pointings, then executing this for each instrument of each pointing would become onerous, but that can be relieved by relatively simple scripting.

## 10. OTHER USEFUL THINGS

Over the course of this cookbook, there have been a number of references to standard SAS tasks whose use is not always straightforward. It might be appropriate to remind the user to read the manual for those tasks. However, it is probably more efficient to mention here some common issues.

### 10.1. PPS File Names

The names of the files produced by *emchain* and *epchain* are of the form<sup>20</sup> PnnnnnnnnnXXsMMMccccDDDD\*.FIT where

nnnnnnnnnn - the observation number

XX is the name of the instrument (M1 for MOS1, M2 for MOS2, or PN for the pn)<sup>21</sup>.

s a single character, either S or U denoting either a “scheduled” or an “unscheduled” observation segment.

MMM - the observation segment number

cccccc - a character string denoting what type of file it is

DDDD - four digit encoding the data subset number and other information.

The “S” or “U” together with the three digits of MMM form the subexposure number. The “S” segments were the scheduled subexposures, while the “U” segments were the subexposures that may have occurred due to an unexpected interruption to the scheduled exposure. Note that while the subexposure numbers for scheduled observations are unique, the subexposure numbers for unscheduled observations may not be; the segments mos1U001 and mos2U001 would likely be simultaneous.

### 10.2. merge

If, for example, one observation has multiple segments from the same instrument, (i.e., mos1S001 and mos1U003) then one would want to merge the event files so all of the data can be used by, for example, *edetect*. In this case, the use of *merge* is trivial:

```
merge set1=mos1S001.fits
      set2=mos1U003.fits outset=mos1T001.fits
```

The only possible issue here is to ensure that the observation segment in the *set1* file occurs before the observation segment in the *set2* file. Also bear in mind that *merge* only merges two files at a time; more

files requires calls to *merge*. And, although it would make sense in many programming languages, you are not allowed to have the output file name being the same as one of the input file names.

One may also merge different observations if the pointing directions are very close. For example, observations 0820840301, 0820840401, and 0820840501 were all taken with the same pointing direction, but in different orbits. One would like to merge these observations to get the deepest possible source detection. In this case, one must merge more than just the event files for individual instruments, one must also merge the attitude files. The following was used to merge files for use with *edetect*:

```
merge set1=0820840301/analysis/atthk.fits
      set2=0820840401/analysis/atthk.fits
      outset=atthk.temp.fits
merge set1=atthk.temp.fits
      set2=0820840501/analysis/atthk.fits
      outset=atthk.fits
rm atthk.temp.fits

merge set1=0820840301/analysis/mos1S001.fits
      set2=0820840401/analysis/mos1S001.fits
      outset=mos1.temp.fits
      mergedifferentobs="yes"
merge set1=mos1.temp.fits
      set2=0820840501/analysis/mos1S001-clean.fits
      outset=mos1.event.fits
      mergedifferentobs="yes"
rm mos1.temp.fits
```

Note well in this case, the required use of the *mergedifferentobs="yes"* parameter.

If you merge individual segments of a single obsid, *espfilt* will work well. If you merge different obsids, *espfilt* will currently fail. (However, it is not clear that it should be able to work.)

### 10.3. Maximal Identical Regions for All Detectors Simultaneously

I had a project with a FOV whose spectrum varied strongly with position. Thus, in order to extract spectra from all detectors that could be used for fitting the soft proton contamination, it was necessary to extract exactly the same region on the sky for each detector. Of course, I wanted as large a region as possible. This is not a trivial selection expression to construct. It's not so much difficult as just painful.

For a MOS detector the expression for the FOV can be expressed as

```
circlemos && (chip1 || chip2 || chip3 || chip4 ||
```

<sup>20</sup>The complete description can be found in [http://xmm-tools.cosmos.esa.int/external/xmm\\_user\\_support/documentation/dfhb/pps.html](http://xmm-tools.cosmos.esa.int/external/xmm_user_support/documentation/dfhb/pps.html).

<sup>21</sup>There are other possibilities, but they are not pertinent here.

chip5 || chip6 || chip7)

where the circle is an expression describing the FOV, and chipn is a description of the area covered by chip n. Since we don't generally lose parts of the pn, we can use an expression of the form

circle<sub>pn</sub> && box<sub>pn</sub>.

If we are extracting events from the MOS1 detector, we need to extract in MOS1 detector coordinates, which means that we must translate the expression for the MOS2 FOV out of MOS2 detector coordinates into sky coordinates and then into MOS1 detector coordinates. Similarly, the expression for the pn must be similarly transformed from one set of detector coordinates to another. To extract events from the MOS2 detector, the FOV expressions for all of the other detectors must be converted to MOS2 detector coordinates. Similarly for the pn.

Assuming that we have the appropriate expressions in the appropriate detector coordinates, the selection expression for a single detector can be expressed as:

```
circleMOS1 && (chip1-1 || chip1-2 || chip1-3 ||
chip1-4 || chip1-5 || chip1-6 || chip1-7) &&
circleMOS2 && (chip2-1 || chip2-2 || chip2-3 ||
chip2-4 || chip2-5 || chip2-6 || chip2-7) &&
circlepn && boxpn
```

which is a rather cumbersome expression. Indeed, you'll get an error message that the expression is too large to store. For MOS1 this corresponds to:

```
((DETX,DETY) IN circle(109.0,-227.0,17158.0))&&
(((DETX,DETY) IN box(-11.0,-0.5,6593.0,6604.5,0.00))||
(DETX,DETY) IN box(6498.0,-13580.0,6594.0,6583.0,0.00))||
(DETX,DETY) IN box(13255.0,-327.5,6595.0,6583.5,0.00))||
(DETX,DETY) IN box(6562.0,13035.5,6593.0,6549.5,0.00))||
(DETX,DETY) IN box(-6405.0,13112.5,6591.0,6591.5,0.00))||
(DETX,DETY) IN box(-13167.0,-76.5,6593.0,6581.5,0.00))||
(DETX,DETY) IN box(-6521.0,-13438.5,6600.0,6582.5,0.00))&&
(DETX,DETY) IN circle(-63.8,-159.8,17088.0))&&
(((DETX,DETY) IN box(219.4,-158.5,6593.0,6604.5,-90.00))||
(DETX,DETY) IN box(-13181.0,-6861.9,6594.5,6584.5,-90.00))||
(DETX,DETY) IN box(29.4,-13533.4,6595.5,6582.5,-90.00))||
(DETX,DETY) IN box(13357.0,-6701.2,6596.0,6573.0,-90.00))||
(DETX,DETY) IN box(13367.8,6480.6,6606.0,6595.0,-90.00))||
(DETX,DETY) IN box(96.0,13059.5,6594.0,6593.5,-90.00))||
(DETX,DETY) IN box(-13213.4,6456.1,6594.5,6573.0,-90.00))&&
(DETX,DETY) IN circle(1163.6,-2521.3,18000.0))&&
(DETX,DETY) IN box(1163.6,-2521.3,16122.5,15642.0,-90.00))
```

For MOS2 this corresponds to:

```
((DETX,DETY) IN circle(57.2,-110.8,17158.0))&&
(((DETX,DETY) IN box(-169.5,-223.4,6593.0,6604.5,90.00))||
(DETX,DETY) IN box(13439.7,6213.8,6594.0,6583.0,90.00))||
(DETX,DETY) IN box(224.5,13033.8,6595.0,6583.5,90.00))||
(DETX,DETY) IN box(-13168.3,6410.8,6593.0,6549.5,90.00))||
(DETX,DETY) IN box(-13314.2,-6556.6,6591.0,6591.5,90.00))||
(DETX,DETY) IN box(-160.5,-13385.5,6593.0,6581.5,90.00))||
(DETX,DETY) IN box(13236.1,-6804.5,6600.0,6582.5,90.00))&&
(DETX,DETY) IN circle(-12.5,-284.5,17088.0))&&
(((DETX,DETY) IN box(-10.0,-0.5,6593.0,6604.5,0.00))||
(DETX,DETY) IN box(6626.0,-13434.5,6594.5,6584.5,0.00))||
(DETX,DETY) IN box(13362.0,-258.5,6595.5,6582.5,0.00))||
(DETX,DETY) IN box(6600.0,13105.0,6596.0,6573.0,0.00))||
(DETX,DETY) IN box(-6583.0,13182.0,6606.0,6595.0,0.00))||
(DETX,DETY) IN box(-13229.0,-56.5,6594.0,6593.5,0.00))||
(DETX,DETY) IN box(-6690.0,-13397.0,6594.5,6573.0,0.00))&&
(DETX,DETY) IN circle(2356.8,934.4,18000.0))&&
(DETX,DETY) IN box(2356.8,934.4,16122.5,15642.0,0.00))
```

For the pn this corresponds to:

```
((DETX,DETY) IN circle(88.3,-34.3,17158.0))&&
(((DETX,DETY) IN box(314.1,80.2,6593.0,6604.5,90.00))||
(DETX,DETY) IN box(-13237.6,-6477.4,6594.0,6583.0,90.00))||
(DETX,DETY) IN box(37.5,-13179.9,6595.0,6583.5,90.00))||
(DETX,DETY) IN box(13371.1,-6438.5,6593.0,6549.5,90.00))||
(DETX,DETY) IN box(13402.1,6529.7,6591.0,6591.5,90.00))||
(DETX,DETY) IN box(188.3,13241.8,6593.0,6581.5,90.00))||
(DETX,DETY) IN box(-13149.4,6542.2,6600.0,6582.5,90.00))&&
(DETX,DETY) IN circle(154.9,136.3,17088.0))&&
(((DETX,DETY) IN box(157.3,-146.8,6593.0,6604.5,0.00))||
(DETX,DETY) IN box(-6596.5,13228.2,6594.5,6584.5,0.00))||
(DETX,DETY) IN box(-13218.3,-7.2,6595.5,6582.5,0.00))||
(DETX,DETY) IN box(-6335.9,-13309.0,6596.0,6573.0,0.00))||
(DETX,DETY) IN box(6845.8,-13270.3,6606.0,6595.0,0.00))||
(DETX,DETY) IN box(13374.7,26.2,6594.0,6593.5,0.00))||
(DETX,DETY) IN box(6721.3,13310.7,6594.5,6573.0,0.00))&&
(DETX,DETY) IN circle(-2201.5,-1101.0,18000.0))&&
(DETX,DETY) IN box(-2201.5,-1101.0,16122.5,15642.0,0.00))
```

If a chip has been turned off, then one can remove it by zeroing out its box size.

## 11. REFERENCES

- Arnaud, M. Neumann, D. M., Aghanim, N., Gastaud, R., Majerowicz, S., & Hughes, J. P. 2001, *A&A*, 365, L80
- Carter, J. A., & Sembay, S. 2008, *A&A*, 489, 837
- Ehle, M., et al. 2005, *XMM-Newton Users' Handbook* (Madrid: ESA)
- Kaaret, P, Koutroumpa, D., Kuntz, K. D., Jahoda, K., Bluem, J., Gulick, H., Hodges-Kluck, E., LaRocca, D. M., Ringuette, R. and Zajczyk, A. 2020 *NatAs*, 4, 1072
- Koutroumpa, D., Collier, M. R., Kuntz, K. D., Lallement, R.; & Snowden, S. L. 2009, *ApJ*, 697, 1214
- Kuntz, K.D., & Snowden, S. L. 2023, in preparation (<https://pages.jh.edu/~kkuntz1/>)
- Kuntz, K. D. 2019, *A&ARv*, 27, 1
- Kuntz, K. D., & Snowden, S. L. 2008, *A&A*, 478, 575
- Liu, W., Chiao, M., Collier, M. R., Cravens, T., Galeazzi, M., Koutroumpa, D., Kuntz, K. D., Lallement, R., Lepri, S. T., McCammon, D., Morgan, K., Porter, F. S., Snowden, S. L., Thomas, N. E., and Uprety, Y., Ursino, E., & Walsh, B. M. 2017, *ApJ*, 834, 33
- Lumb, D. H., Warwick, R. S., Page, M., & De Luca, A. 2002, *A&A*, 389, 93
- Nevalainen, J., Markevitch, M., & Lumb, D. 2005, *ApJ*, 629, 172
- Read, A., & Ponman, T. 2003, *A&A*, 409, 395
- Snowden, S. L., Collier, M. R., Cravens, T., Kuntz, K. D.; Lepri, S. T., Robertson, I., & Tomas, L. 2009, *ApJ*, 691, 372
- Snowden, S. L., Collier, M. R., & Kuntz, K. D. 2004, *ApJ*, 610, 1182
- Snowden, S. L., Mushotzky, R. F., Kuntz, K. D., and Davis, D. S. 2008, *A&A*, 478,615
- Snowden, S. L., Egger, R., Finkbeiner, D. P., Freyberg, M. J., Plucinsky, P. P. 1998, *ApJ*, 493, 715
- Strüder, L., et al. 2001, *A&A* 365, L18
- Turner, M. J. L., et al. 2001, *A&A* 365, L27
- Vikhlinin, A., Markevitch, M., Murray, S. S., Jones, C., Forman, W., & Van Speybroeck, L. 2005, *ApJ*, 628, 655

## A. TASK DESCRIPTIONS

Appendix A provides detailed information on how to run the individual tasks of the XMM-ESAS package. *prefix* is used to indicate a general identifier of output files. It is the unique exposure identifier for the data set, e.g., 1S001 or 2U002 for the MOS instruments, where it is preceded in file names by the letters *mos*, e.g., *mos1S001* or *mos2U002*. For the pn the *prefix* is of the form, e.g., S003 or U003 preceded by the letters *pn*, e.g., *pnS003* or *pnU003*.

Since the ESAS tasks are now part of SAS, SAS must be initialized, with `$SAS_ODF` and `$SAS_CCF` pointing to the appropriate locations. The ODF data must be on disk and unzipped and the current location should be the working directory.

What follows is an alphabetical list of the tasks that will be described in the remainder of this appendix. If a bug is found in the original version, please let the authors know about it. Be aware that the FORTRAN routines can't be updated without recompiling, which is beyond the scope of what most users will want to do.

- *binadapt*

*binadapt* bins/adaptively smooths individual exposures or exposures from single observation combined by the task *combimage* into count rate and count rate uncertainty images.

- *binadaptmerge*

*binadaptmerge* bins/adaptively smooths mosaicked images combined by the task *mosaicmerge* into count rate and count rate uncertainty images.

- *cheese*

*cheese* does source detection and creates “cheese” masks for point-source masking during image processing. *cheese* allows the user to set a flux threshold over the field for the removal of the source contributions to the spectra as well as the images.

- *combimage*

*combimage* combines co-aligned event, exposure, QPB, SPF, and SWCX background images from different exposures and different instruments from the same ObsID.

- *convregion*

*convregion* converts region information in celestial coordinates into region information in detector coordinates. It will do so for region fits files, ascii lists, or individual regions.

- *emanom*

*emanom* reads an event file for a MOS detector and determines whether any of the CCDs are in anomalous states.

- *emlfill*

*emlfill* corrects an `emllist.fits` file for sources whose mean flux is marked as NULL by recalculating that flux from just those observations where the source was detected.

- *emlmerge*

*emlmerge* merges a number of `emllist.fits` files from different observations to create a single list with no duplicate sources. **May not yet be fully implemented.**

- *espfill*

*espfill* filters and cleans event files of soft proton flare contamination. Produces both a cleaned event file and a number of diagnostics.

- *makemaskmerge*

*makemaskmerge* creates a “cheese” mask from a merged source list created by *emlmerge*. **May not yet be fully implemented.**

- *mosaicmerge*

*mosaicmerge* creates mosaicked count, exposure, QPB, SPF, and SWCX background images for multiple observations and exposures.

- *mosback*

*mosback* takes the output of the routine *mosspectra* and creates QPB spectra and images in detector coordinates, as selected, for EPIC MOS data.

- *mosspectra*

*mosspectra* processes the filtered event files from the task *espfilt* to produce a set of intermediate files for the production of QPB background spectra and images. *mosspectra* also produces source spectra and the appropriate Redistribution Matrix Files, RMFs, and Ancillary Region Files, ARFs, for spectral analysis.

- *pnback*

*pnback* takes the output of *pnspectra* and creates QPB spectra and images in detector coordinates, as selected, for EPIC pn data.

- *pnspectra*

*pnspectra* provides the same functions for pn data as *mosspectra* provides for the MOS. However, it also creates spectra and images from the out-of-time processing.

- *pointsrc*

*pointsrc* calculates the appropriate Xspec normalization for the extragalactic background given the user-selected point-source exclusion threshold.

- *proton*

*proton* produces images in detector coordinates of the model residual SP contamination. To do so, it uses the fitted values of the SP component from Xspec and standard detector maps.

- *protonscale*

*protonscale* extracts the appropriate scale factors for the solid angle and relative SP contributions to include in spectral fitting.

- *rotdet2sky*

*rotdet2sky* does the rotation of the QPB images output from *mosback* and *pnback*, the SP images output from *proton*, and the SWCX output from *swcx* which are in detector coordinates, into images in sky coordinates.

- *sppartial*

*sppartial* scales the SP spectral fit results from a limited region of the detector to the full field of view. This allows the user to fit for the SP contamination in regions of lower surface brightness (e.g., an outer annulus for a cluster of galaxies) which can significantly improve the accuracy of the fit.

- *swcx*

*swcx* produces images in detector coordinates of the model residual SWCX contamination. To do so, it uses the fitted values of the SWCX component from Xspec and standard detector maps.

## A.1. BINADAPT

---

### Description:

*binadapt* combines the different components of an image to produce background subtracted and exposure corrected image for either a single exposure or all exposures of a single ObsID. The components must have been previously merged using the task *combimage*. Once combined, the resulting image can be binned and/or adaptively smoothed. In adaptive smoothing, for each unmasked pixel, the program will average neighboring pixels within a circle of increasing radius until a selected number of counts (weighted by a triangular weighting function) (*smoothcounts*) is reached. The original pixel is then given the weighted average surface brightness for the pixels within the circle. Images can also be binned by *binfactor* pixels before smoothing. In the cases where the components are just combined, or combined and binned, both the rate and the uncertainty in the rate are produced. It is assumed that adaptively smoothed images are not suitable for quantitative analysis, so no uncertainty image is produced for them. Some other diagnostic files are also produced.

### Requirements:

The current location should be the working directory.

All of the files to be used for all of the instruments must be previously created. These include event images, exposure images, background images (QPB, SP, and SWCX), and mask images. The task *combimage* must have been successfully run for the exposures for the data from multiple exposures to be used.

### Calling Parameters:

*prefix* – Either the input instrument and exposure ID (1S001, etc.) for a single instrument or “comb” for the combination of instruments of a given obsid.  
*elow* – The lower energy limit (used for constructing the input and output file names).  
*ehigh* – The upper energy limit (used for constructing the input and output file names).  
*withpartbkg* – If true, subtract the quiescent particle background.  
*withspbkg* – If true, subtract the soft proton flare background.  
*withswcxbkg* – If true, subtract the SWCX background.  
*withmask* – If true, use an extra mask to remove regions with low exposure, bright sources, etc..  
*maskfile* – The file name for an image to provide additional masking as controlled by *withmask*. The mask images must be the same size and projection as the other images.  
*withbinning* – If true output is binned by *binfactor* pixels.  
*binfactor* – Number of pixels by which to bin.  
*withsmoothing* – If true output is adaptively smoothed.  
*smoothcounts* – The weighted number of counts to accumulate for the smoothing (e.g., 50).  
*maskthresh* – The scale factor for excluding regions from the smoothing (e.g., 0.02) based on a mask image. In the default mode the average exposure is calculated and then any pixel with exposure less than *fraction\*average* value is excluded.

### Output:

*comb-adaptimsky-350-1100.fits* – adaptively smoothed rate image.  
*comb-radfilt-350-1100.qdp* – QDP file containing the radial profile. Created for all calls to *binadapt*.  
*comb-rateimsky-350-1100.fits* – combined rate image. Not produced if *withsmoothing* is true.  
*comb-sigimsky-350-1100.fits* – uncertainty in the rate image. Not produced if *withsmoothing* is true.  
*comb-size-350-1100.qdp* – QDP file containing a histogram of the radii needed to accumulate *smoothcount* counts. Produced only if *withsmoothing* is true.  
*comb-sizeimsky-350-1100.fits* – image showing the radii required to accumulate *smoothcount* counts. Produced only if *withsmoothing* is true.



## Examples:

Combining but doing nothing else:

```
binadapt prefix=comb elow=350 ehigh=1100 withpartbkg=true withswcxbkg=true  
  withspbkg=true withmask=false withbinning=false withsmoothing=false smoothcounts=50
```

Combining and binning only:

```
binadapt prefix=comb elow=350 ehigh=1100 withpartbkg=true withswcxbkg=true  
  withspbkg=true withmask=false withbinning=true binfactor=4  
  withsmoothing=false
```

Combining and smoothing only:

```
binadapt prefix=comb elow=350 ehigh=1100 withpartbkg=true withswcxbkg=true  
  withspbkg=true withmask=false withbinning=false  
  withsmoothing=true smoothcounts=50
```

Combining, binning, and then smoothing:

```
binadapt prefix=mos1S001 elow=350 ehigh=1100 withpartbkg=true withswcxbkg=true  
  withspbkg=true withmask=false withbinning=true binfactor=4  
  withsmoothing=true smoothcounts=50
```

## A.2. BINADAPTMERGE

---

### Description:

*binadaptmerge* combines the different components of an image to produce background subtracted and exposure corrected image for either a mosaic of multiple ObsIDs. The components must have been previously merged using the task *mosaicmerge*. Once combined, the resulting image can be binned and/or adaptively smoothed. In adaptive smoothing, for each unmasked pixel, the program will average neighboring pixels within a circle of increasing radius until a selected number of counts (weighted by a triangular weighting function) (`smoothcounts`) is reached. The original pixel is then given the weighted average surface brightness for the pixels within the circle. Images can also be binned by `binfactor` pixels before smoothing. In the cases where the components are just combined, or combined and binned, both the rate and the uncertainty in the rate are produced. It is assumed that adaptively smoothed images are not suitable for quantitative analysis, so no uncertainty image is produced for them. Some other diagnostic files are also produced.

If the image is being adaptively smoothed, then there may be a desire to “fill in” the small gaps left due to chips gaps and the like. For each “fill” pass, any pixel that is zero, but has at least three adjacent pixels that are non-zero, will be replaced with the average of the surrounding non-zero pixels. *The fill procedure is meant for cosmetic use only.*

### Requirements:

The current location should be the working directory with the mosaicked images. *mosaicmerge* must have been successfully run for the components to be used.

### Calling Parameters:

`elowlist` – The lower energy limit (used for constructing the input and output file names).  
`ehighlist` – The upper energy limit (used for constructing the input and output file names).  
`withpartbkg` – If true, subtract the quiescent particle background.  
`withspbkg` – If true, subtract the soft proton flare background.  
`withswcxbkg` – If true, subtract the SWCX background.  
`withmask` – If true, use an extra mask to remove regions with low exposure, bright sources, etc..  
`maskfile` – The file name for an image to provide additional masking as controlled by `withmask`. The mask images must be the same size and projection as the other images.  
`withbinning` – If true output is binned by `binfactor` pixels.  
`binfactor` – Number of pixels by which to bin.  
`withsmoothing` – If true output is adaptively smoothed.  
`smoothcounts` – The weighted number of counts to accumulate for the smoothing (e.g., 50).  
`maskthresh` – The scale factor for excluding regions from the smoothing (e.g., 0.02) based on a mask image. In the default mode the average exposure is calculated and then any pixel with exposure less than `fraction*average` value is excluded.  
`fill` – Number of “filling” passes in the case of adaptively smoothing.

### Output:

`merged-adapt-350-1100.fits` – an adaptively smoothed rate image. Produced only if `withsmoothing` is true.  
`merged-radfilt-350-1100.qdp` – a QDP file containing the radial profile. Produced for all call to this program.  
`merged-rate-350-1100.fits` – the rate (image). Produced only if `withsmoothing` is false.  
`merged-sigma-350-1100.fits` – the uncertainty in the rate (image). Produced only if `withsmoothing` is false.  
Here the philosophy is that an adaptively smoothed image is not suitable for quantitative analysis, so an uncertainty image is otiose.  
`merged-size-350-1100.fits` – an image showing the radius required to accumulate `smoothcount` counts.

Produced only if `withsmoothing` is true.

`merged-size-350-1100.qdp` – a QDP file containing a histogram of the radii required to accumulate `smoothcount` counts. Produced only if `withsmoothing` is true.

## Example:

Combining but doing nothing else:

```
binadaptmerge elowlist='350' ehighlist='1100' withpartbkg=yes withspbkg=true  
  withswcxbkg=true withmask=false maskfile=mosaic-cheeset.fits  
  withbinning=false withsmoothing=false fill=0
```

Combining, binning, and then smoothing:

```
binadaptmerge elowlist='350' ehighlist='1100' withpartbkg=yes withspbkg=true  
withswcxbkg=true withmask=false maskfile=mosaic-cheeset.fits  
  withbinning=true withsmoothing=true smoothcounts=50 fill=0
```

### A.3. CHEESE

---

#### Description:

*cheese* runs source detection on full-field images and creates cheese masks (source removal masks) from the output. *cheese* produces the event, exposure, and mask images that are required in a user-selected energy band. Running *cheese* is not required if only the spectral files with all counts including point sources are required, or if excluding point sources is not of interest.

*cheese* now includes the capabilities of *cheese-bands* to create all of the above in three bands – soft, hard, and total, for instance using bands of 0.4 – 1.3 keV, 2.0 – 7.2 keV, and 0.4 – 7.2 keV (the last excluding the 1.3 – 2.0 keV band).

In essence, *cheese* provides an ESAS consistent wrapper around the SAS task *edetect-chain*.

#### Requirements:

SAS must be initialized, with \$SAS.ODF and \$SAS.CCF pointing to the appropriate locations.

The current location should be the working directory.

*espfilt* must have been successfully run.

#### Calling Parameters:

**mos1file** – The MOS1 event file.

**mos2file** – The MOS2 event file.

**pnfile** – The pn event file.

**pnootfile** – The pn OOT event file.

**scale** – This is the energy fraction (e.g., 0.5) which sets the exclusion radius of point sources.

**ratetotal** – This is the flux threshold (in units of  $1.0 \times 10^{-14}$  cgs for the exclusion of point sources in the total (combined) band (e.g., 1.0).

**ratesoft** – This is the flux threshold (in units of  $1.0 \times 10^{-14}$  cgs for the exclusion of point sources in the soft band (e.g., 1.0).

**ratehard** – This is the flux threshold (in units of  $1.0 \times 10^{-14}$  cgs for the exclusion of point sources in the hard band (e.g., 1.0).

**dist** – Minimum separation in arc seconds between masked sources.

**elowlist** – Low-energy limits for the bands in eV (e.g., 400 or 400 2000).

**ehighlist** – High-energy limits for the bands in eV (e.g., 7200 or 1300 7200).

**mlmin** – The minimum likelihood to accept for a source.

**keepinterfiles** – Boolean for keeping (or not) the intermediate files.

#### Output:

**atthk.fits** – SAS attitude file (created only if necessary).

**boxlist\_l.fits** – The output from the first pass of *eboardetect*.

**boxlist\_m.fits** – The output from the second pass of *eboardetect*.

**emllist.fits** – The output from the *emldetect*.

Instrument specific files are created only when the instruments are included

For each band

**P-bkgregBdet.fits** – The band B background region file made from the filtered source list. Note that this list excludes the sources and is in detector coordinates.

**P-bkgregBsky.fits** – The band B background region file made from the filtered source list. Note that this list excludes the sources and is in sky coordinates.

P-cheeseB.fits – The band B cheese mask image for the P exposure.

#### Additional Diagnostic Files

P-fovimB.fits – The FOV image in band B.

P-fovimBbkg.fits – The background surface brightness in band B determined by *edetect\_chain*.

P-fovimBexp.fits – The exposure map in band B determined by *edetect\_chain*.

P-fovimBmask.fits – The mask in band B determined by *edetect\_chain*.

P-fovimBsen.fits – The sensitivity map in band B determined by *edetect\_chain*.

P-fovimBsmmap.fits – The source map in band B determined by *edetect\_chain*.

## Examples:

```
cheese mos1file='mos1S001-allevc.fits' mos2file='mos2S002-allevc.fits'
  pnfile='pnS003-allevc.fits' pnootfile='pnS003-allevcoot.fits' elowlist=350
  ehighlist=1100 scale=0.5 mlmin=10 ratetotal=0.2 dist=50. keepinterfiles=no -V 7
cheese mos1file='mos1S001-allevc.fits' mos2file='mos2S002-allevc.fits'
  pnfile='pnS003-allevc.fits' pnootfile='pnS003-allevcoot.fits' elowlist='350 2000'
  ehighlist='1100 7200' scale=0.5 mlmin=10 ratetotal=0.2 ratesoft=0.2 ratehard=0.2
  dist=50. keepinterfiles=no -V 7
```

## A.4. COMBIMAGE

---

### Description:

*combimage* combines the various MOS1, MOS2, and pn images from a single observation, as well as images from multiple exposures into single count, exposure, model QPB, and SP background images.

*combimage* compensates for the inclusion of observations with different filters in the mosaic. It uses the results of PIMMS with the assumption of a power-law spectrum with photon indices (alpha) of 2.4, 1.7, and 1.0, and absorption of  $N_H = 2 \times 10^{20}$  H I  $\text{cm}^{-2}$ . The user enters a value for alpha between 1.0 and 2.4 where 1.0 will select the hard spectrum, 1.7 selects the medium spectrum, and 2.4 selects the soft spectrum. Intermediate values will produce a linear scaling between the two nearest spectra. The exposure image is then scaled by the ratio of the model count rates for the MOS2 medium filter versus the thin or thick, making the resultant image appropriate for the MOS2 medium filter.

### Requirements:

The current location should be the working directory.

To merge event and exposure images, *mosspectra* and/or *pnspectra* must have been successfully run for all exposures.

If particle background images are to be merged *mosback* and/or *pnback* and *rotdet2sky* must have been successfully run for all exposures.

To include the soft proton flare images the soft proton spectrum must have been fit and the tasks *proton* and *rotdet2sky* successfully run.

To include the SWCX images the SWCX Gaussian spectra must have been fit and the tasks *swcx* and *rotdet2sky* successfully run.

Inclusion of the QPB, SP, and SWCX images are selected in the command.

### Calling Parameters:

**prefixlist** – Detector and exposure identifiers (e.g., **prefixlist**="1S001 2S002 S003")

**withpartbkg** – If true, includes the quiescent particle background.

**withspbkg** – If true, includes the soft proton flare background.

**withswcxbkg** – If true, includes the swcx background.

**withmask** – If true, includes a mask - mutually exclusive with **withcheese**.

**withcheese** – If true, uses a cheese mask whose type is specified in **cheesetype**.

**cheesetype** – If **withcheese** is true, applies a cheese mask, s, h, or t (soft, hard, or total), or a merged cheese mask **smsl**, **hmsl**, or **tmsl** (soft, hard, or total).

**elowlist** – List of the low energy limits for the energy bands to be merged in eV (e.g., **elowlist**=400 for a single band, **elowlist**="350 2500" for two bands).

**ehighlist** – List of the high energy limits for the energy bands to be merged in eV (e.g., **ehighlist**="750 1250").

**alpha** – Assumed power law spectral index for scaling exposure maps to account for different filters.

### Output:

**comb-fovim-elow-ehigh.fits** – The combined count image for the *prefix* exposure, selected energy band (**elow** and **ehigh**), and the selected region in sky coordinates.

**comb-expimelow-ehigh.fits** – The combined exposure image for the *prefix* exposure, selected energy band (**elow** and **ehigh**), and the selected region in sky coordinates.

**comb-bkgim-elow-ehigh.fits** – The combined model particle background image for the *prefix* exposure, selected energy band (**elow** and **ehigh**), and the selected region in sky coordinates.

**comb-protim-elow-ehigh.fits** – The combined model soft proton background image for the *prefix* exposure,

selected energy band (**e**low and **e**high), and the selected region in sky coordinates.

`comb-swxcim-elow-high.fits` – The combined model swcx background image for the *prefix* exposure, selected energy band (**e**low and **e**high), and the selected region in sky coordinates.

### Example:

```
combimage prefixlist="1S001 2S002 S003" withpartbkg=true withspbkg=true withswcxbkg=true  
withcheese=true cheesetype=t elowlist=350 ehighlist=1100 alpha=1.7
```

## A.5. CONVREGION

---

### Description:

*convregion* converts region information in sky coordinates (RA,Dec) into region information in detector coordinates (DETX,DETY). It operates in three modes where 1) region input and output files are used, 2) ascii file lists of region descriptors are both input and output, and 3) region descriptors for individual regions are input on the command line and output on the screen. Currently only circular and elliptical regions are supported.

### Requirements:

The current location should be the working directory.

A region information file with entries in celestial coordinates with RA and Dec in decimal degrees, the radius in circular regions in arc minutes or the semi-major and semi-minor axes in arc minutes for elliptical regions, and the rotation angle in decimal degrees for elliptical regions.

### Calling Parameters:

**mode** – **mode=1** uses fits region files for both input and output. **mode=2** uses ASCII files for both input and output. Each line is an ordered groups of **SHAPE**, **RA**, **DEC**, **R** (for elliptical regions the semi-major and semi-minor axes are stored in **R**), and **ROTANG** (**ROTANG** is not included for circular regions. **mode=3** uses direct input of parameters, with output to the screen.

**inregion** – Input region file (either .fits format or ASCII). An example ASCII file:

```
ELLIPSE 84.7 -68.35 4.0 2.0 25.0
!ELLIPSE 84.5 -68.45 5.0 3.0 25.0
ellipse 84.3 -68.55 6.0 4.0 25.0
!ellipse 84.1 -68.65 7.0 5.0 25.0
CIRCLE 84.7 -68.65 4.0
!CIRCLE 84.5 -68.55 5.0
circle 84.3 -68.45 6.0
!circle 84.1 -68.35 7.0
```

**outregion** – Output region file (either .fits format or ASCII, will be the same format as the input file).

**imagefile** – Sky coordinate image used to extract the observation coordinates and roll angle.

In addition, only for **mode=3**: **shape**, **ra**, **dec**, **radius** or **semimajor** and **semiminor**, and **rotangle** (elliptical regions only) can be input parameters.

### Output:

**mode=1** – User named region fits file

**mode=2** – User named file containing region data

**mode=3** – Screen output with region data

### Example:

```
convregion mode=1 inputfile=input-radec.fits outputfile=output-det-mos1.fits
  imagefile=mos1S001-fovimt.fits
convregion mode=2 inputfile=input.txt outputfile=output-det-mos1.txt
  imagefile=mos1S001-fovimt.fits
convregion mode=3 imagefile=mos1S001-fovimt.fits shape=CIRCLE ra=84.7 dec=-68.35 radius=5.0
```



## A.6. EMANOM

---

### Description:

*emanom* reads a MOS event file, calculates the 0.4-0.8/2.5-5.0 keV hardness ratio for the corner data for each CCD, and determines whether the CCD is in an anomalous state or not. The hardness ratios (anomhrN), the uncertainty in the hardness ratios (anomheN), and the anomalous state flag (anomflN) for each chip (N) is written to the primary header of the event file.

The anomalous state flags are

- G good
- B bad
- I intermediate
- O chip is off
- U undetermined as there were too few counts to form a hardness ratio

### Requirements:

The current location should be the working directory.

The soft proton filtering need not be done before this routine is run as the energy bands used by *emanom* and *espfilt* are disjoint.

### Calling Parameters:

**eventfile** – The name of the input event file containing a MOS exposure.

**writekeys** – If yes (the default) writes the header keywords, otherwise it does not.

**writelog** – If yes (the default) writes a log file, otherwise it does not.

**keepcorner** – If yes (the default) the corner event list is saved (not recommended)

**cornerfile** – The name of the out event file containing only the corner data. If **keepcorner** set to “yes” and the **cornerfile** parameter is not set, the output will be in **temp\_corner.fits**.

### Output:

**M1S001-anom.log** or **M2S002-anom.log** – A file containing the hardness ratios, hardness ratio errors, and anomalous state flags for the instrument.

**temp\_corner.fits** or the user specified file – An event file containing the events from the corners only.

### Examples:

```
emanom eventfile=mos1S0001.fits writekeys=yes keepcorner=no
```

## A.7. EMLFILL

---

### Description:

*emlfill* takes an `emllist.fits` format source list, and hunts for sources where the mean flux (calculated over all instruments) is NULL. This generally means that the source does not exist in one of the instruments is missing. The NULL is replaced with the mean flux calculated over all instruments that detected the source.

### Requirements:

The current location should be the working directory.

The `emllist.fits` file should exist.

### Calling Parameters:

`emlin` – Name of the input `emllist.fits` format file.

`emlout` – Name of the output `emllist.fits` format file.

### Output:

A file containing an `emllist.fits` format file where the mean flux of each source is properly calculated.

### Example:

```
template emllin=emllist.fits emlout=emllist_out.fits
```

## A.8. EMLMERGE

---

### Description:

*emlmerge* creates a master list of sources detected from a list of observations which have been processed to the point of creating an *emldetect* source list. Sources are compared and those within 2" of each other have the one with the lower flux thrown out. *emlmerge* will process source lists with only a total band as created by it cheese or with two bands, hard and soft, along with a total band as created by it cheese.

The output fits file can then be used on the individual observation (obsID) level to make self-consistent cheese masks for the mosaicking of observations.

### Requirements:

The individual observations which will have their *emldetect* source lists merged must be processed through *cheese*.

### Calling Parameters:

**dirfile** – File containing the list of observations for data to be merged. For instance, if the merging processing is being done in the directory `/DATA/merge` and the individual observations are located in the parallel directories `/DATA/obs1/proc` and `/DATA/obs2/proc`, the file `dirlist.dat` should have the entries:

```
/DATA/obs1/proc/  
/DATA/obs2/proc/
```

**maxlikelim** – Minimum maximum-likelihood detection significance for sources in the merged source list.

**runemlfill** – If true, will run *emlfill* on each `emllist.fits` file before merging.

### Output:

`emlmerge.fits` – The merged source list fits file in the same format as the `emllist.fits` file.

### Example:

```
emlmerge dirfile=dir.dat maxlikelim=15.0
```

## A.9. ESPFILT

---

### Description:

*espfilt* reads an event file from either MOS or pn, calculates the light curve in both the FOV and the corner data and, from those, determines which time periods are contaminated with soft proton flares. Two different methods can be applied.

The *histogram* method is an implementation of the method proposed in Kuntz & Snowden (2008). This method builds a histogram of the count rate from a binned and/or smoothed light curve. In the presence of short proton flares, the peak of that histogram represents the quiescent rate, while the flared intervals form a tail to higher values. The algorithm fits the peak of the histogram with a gaussian and sets as “flared” all intervals further than a user defined number of  $\sigma$  from the peak.

The *ratio* method is an implementation of the method proposed in De Luca & Molendi (2004). This method looks at the ratio of the count rate in an annulus just inside the FOV to that in the corners. Intervals for which that ratio exceeds some user determined value are considered “flared”.

For both methods *espfilt* forms a GTI file which is then used to remove the flared intervals.

It should be noted that for a MOS exposure, the histogram method will run in  $\sim 10$  seconds, while the ratio method may require  $\sim 5$  minutes.

### Requirements:

The current location should be the working directory.

### Calling Parameters:

**eventfile** – The name of the input event file.

**withoot** – If true then, for pn apply GTI to OOT event file as well.

**ootfile** – Input OOT event file.

**elow** – The minimum spectral channel for the light curve. For the histogram method this should be 2500; for the ratio method it should be 8000.

**ehigh** – The maximum spectral channel for the light curve. For the histogram method this should be 8500; for the ratio method it should be 12000.

**keepinterfiles** – If true then intermediate files are kept.

**method** – Either *histogram*, the default, or *ratio*.

For the histogram method:

**withsmoothing** – If yes the light curve is smoothed by **smooth** seconds. The default is no.

**smooth** – The number of seconds by which to smooth the light curve if **withsmoothing** is true. The default is 60.

**rangescale** – The width of the region over which the Gaussian is fit to the histogram. The default is 6, which is good for the MOS. We recommend 15 for the pn.

**allowsigma** – The  $\pm$  range of  $\sigma$  to be accepted as being “flare-free”.

For the ratio method:

**withbinning** – If yes the light curve is binned by **binning** seconds. The default is no.

**binsize** – The number of seconds by which to bin the light curve if **withbinning** is true. The default is 60.

For the ratio method:

**ratio** – The threshold ratio of FOV/corner over which the interval is considered to be flared.

**limits** – A two element parameter setting the values over which to form the histogram. This should probably not be tampered with unless you really know what you are doing.

### Output:

**x-corevc.fits** The filtered event list for the corner data.

`x-corimc.fits` The diagnostic image of corner data after filtering.  
`x-corlc.fits` The light curve for the corner data.  
`x-allevc.fits` The filtered event list for all the data.  
`x-allevc-oot.fits` The filtered event list for all the data if `withoot=Y`  
`x-fovimc.fits` The diagnostic image of FOV data after filtering.  
`x-fovlc.fits` The light curve for the FOV data.  
`x-gti.fits` The GTI file containing the intervals without soft proton flares.  
`x-hist.qdp` The graphical output of the light curve.

If `keepinterfiles` is true, the following are also produced:

`x-corev.fits` The unfiltered event list for the corner data.  
`x-corim.fits` The diagnostic image of corner data before filtering.  
`x-fovim.fits` The diagnostic image of FOV data before filtering.  
`x-gti.txt` The ASCII version of the GTI file.

## Examples:

```
espfilt eventfile=mos2S002.fits method=histogram elow=2500 ehigh=8500 withsmoothing=yes
smooth=51 rangescale=6.0 allowsigma=3.0 keepinterfiles=false
espfilt eventfile=pnS003.fits method=histogram elow=2500 ehigh=8500 withsmoothing=yes
smooth=51 rangescale=15.0 allowsigma=3 withoot=Y ootfile=pnS003-oot.fits
keepinterfiles=false
```

## A.10. MOSAICMERGE

---

### Description:

*mosaicmerge* combines the images produced for individual observations (ObsIDs) into larger field mosaics. This includes the event and exposure images (output from *mosspectra/pnspectra*), QPB background images (output from *mosback/pnback* processed by *rotdet2sky*), the SP images (output from *proton* also processed by *rotdet2sky*) and the SWCX images (output from *swcx* also processed by *rotdet2sky*). Data type to be mosaicked, image size, pixel size, coordinate system, and central coordinates are all user selected.

*mosaicmerge*, when mosaicking the exposure images compensates for the inclusion of observations with different instruments and filters. It uses the results of PIMMS with the assumption of a power-law spectrum with photon indices (alpha) of 2.4, 1.7, and 1.0, and absorption of  $N_H = 2 \times 10^{20}$  H I  $\text{cm}^{-2}$ . The user enters a value for alpha between 1.0 and 2.4 where 1.0 will select the hard spectrum, 1.7 selects the medium spectrum, and 2.4 selects the soft spectrum. Intermediate values will produce a linear scaling between the two nearest spectra. The exposure image is then scaled by the ratio of the model count rates for the MOS2 medium filter versus the thin or thick, making the resultant image appropriate for the MOS2 medium filter.

### Requirements:

With no surprise, the component to be mosaicked must already have been produced for all ObsIDs to be included.

### Calling Parameters:

**dirfile** – File containing the list of exposures, including the directory string and the prefixes (i.e. mos1S001), for the data to be merged.

**type** – Type of file/component to mosaic: 1: count image, 2: exposure, 3: QPB counts, 4: SP counts, 5: SWCX counts, 6: cheese image, or 7: msl cheese image.

**cheesemosaictype** – If one is mosaicking the cheese maps themselves, what type of cheese map to use (s (soft), h (hard), t (total), smsl, hmsl, tmsl (merged)).

**coordsys** – Selects which coordinate system should be used, 1: ecliptic, 2: equatorial, 3: galactic.

**crvaln1** – Central longitude of the projection.

**crvaln2** – Central latitude of the projection.

**pixelsize** – Pixel size of the projection in arc minutes.

**alpha** – Assumed power law spectral index ( $2.4 > \alpha > 1.0$ ) for scaling the exposure maps to account for different filters.

**xdim** – Out image size in X dimension.

**ydim** – Out image size in Y dimension.

**withmask** – If true apply a non-cheese mask. (**withmask** and **withcheese** are mutually exclusive.)

**maskfile** – Name of mask file if **withmask** is true.

**withcheese** – If true apply a cheese mask to the mosaic. (**withmask** and **withcheese** are mutually exclusive.)

**cheesemasktype** – Type of cheese mask to be used if **withcheese** is true: **s**: cheeses, **h**: cheeseh, **t**: cheeset, **smsl**: soft merged mask, **hmsl**: hard merged mask, **tmsl**: total merged mask created by *makemaskmerge*.

### Output:

comb-bkgimsky-350-1100.fits comb-expimsky-350-1100.fits comb-fovimsky-350-1100.fits comb-protimsky-350-1100.fits comb-radfilt-350-1100.qdp comb-rateimsky-350-1100.fits comb-sigimsky-350-1100.fits comb-swxcimsky-350-1100.fits

For the different values of **comp**, the output files are:

**component=1: mosaic-fovim-ehigh-ehigh.fits** – The count image

component=2: mosaic-expim-*ehigh-ehigh.fits* – The exposure image  
component=3: mosaic-bkgims-*ehigh-ehigh.fits* – The QPB count image  
component=4: mosaic-protim-*ehigh-ehigh.fits* – The SP count image  
component=5: mosaic-swcxims-*ehigh-ehigh.fits* – The SWCX count image  
component=6: mosaic-cheeseN-*ehigh-ehigh.fits* – The source mask image.

### Example:

```
mosaicmerge dirfile=mosaic.list type=1 coordsys=2 crval1=$RAPNT crval2=$DECPNT  
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'
```

## A.11. MOSBACK

---

### Description:

*mosback* creates model particle background spectra and images (if selected with a non-zero energy range) for the selected region from the intermediate files produced from *mosspectra*. The resultant image is in detector coordinates. The resultant image is in detector coordinates which can be transformed into sky coordinates by with *rotdet2sky*. *mosback* creates a QDP plot file which shows the source and model background spectra for the observation. Any enhancement of the data over the particle background model at higher energies probably indicates residual soft proton contamination, unless there are really hard and bright sources in the field.

### Requirements:

The current location should be the working directory.

*mosspectra* must have been successfully run at least once for the exposures to be used.

### Calling Parameters:

**inspecfile** – Name of the input spectrum file from *mosspectra* (e.g., `mos1S001-fov.t.pi`).

**outspecfile** – Name of the output spectrum file. If it is unspecified, the program will construct the file `P-bkg.pi`.

**rmffile** – Name of the input redistribution matrix from *mosspectra* (e.g., `mos1S001.rmf`)

**withplotfiles** – If true, creates diagnostic QDP plot files.

**iningfile** – Template image for background image production. If it is unspecified, the program uses `P-fovimgdet-elow-ehigh.fits`.

**outingfile** – Name of the output quiescent particle background image. If it is unspecified, the program will construct the file `P-bkgimgdet-elow-ehigh.fits`.

**elow** – The low energy limit for the energy band in eV (e.g., 350).

**ehigh** – The high energy limit for the energy band in eV (e.g., 1100). If **elow** and **ehigh** are set to 0, the image processing will be eliminated and only spectral files will be produced.

**ccds** – The CCDs to be included, (e.g., "T T T F T T T").

### Output:

**P-bkg.pi** – The model particle background spectrum for the P exposure and the selected region.

**P-bkgimgdet-elow-ehigh.fits** – The model particle background image for the P exposure, selected energy band (**elow** and **ehigh**), and the selected region. The image is in detector coordinates.

**P-augindiv.qdp** – Plots of the augmented QPB spectra for each chip. Produced if **withplotfiles** is true.

**P-augspec.qdp** – A QDP plot file showing the observed spectrum and the model background spectrum. Produced if **withplotfiles** is true.

**P-bkgaccum.qdp** – A QDP plot file showing the accumulating background spectrum. Chip 1 at the bottom increasing upwards. Produced if **withplotfiles** is true

**P-bkgspec.qdp** – A QDP plot file showing the normalized model background spectrum. Produced if **withplotfiles** is true.

**P-bkgindiv.qdp** – Plots of the individual corner spectra. Produced if **withplotfiles** is true.

**P-bridgefit.qdp** – A QDP plot file showing the fit for the Al-Si bridge. Produced if **withplotfiles** is true.

**P-ratehard.qdp** – A QDP plot file showing the selected region of hardness/count rate distributions for the various CCDs. Produced if **withplotfiles** is true.

**P-ratioindiv.qdp** – Plots of the FOV/corner ratio spectra for each chip. Produced if **withplotfiles** is true

### Examples:



```
mosback inspecfile=mos1S001-fovvt.pi elow=350 ehigh=1100 ccds="T T T F F F T" (to create both the
spectrum and the band image)
mosback inspecfile=mos1S001-fovvt.pi elow=0 ehigh=0 ccds="T T T F F F T" (to create the spectrum
but not the band image)
```

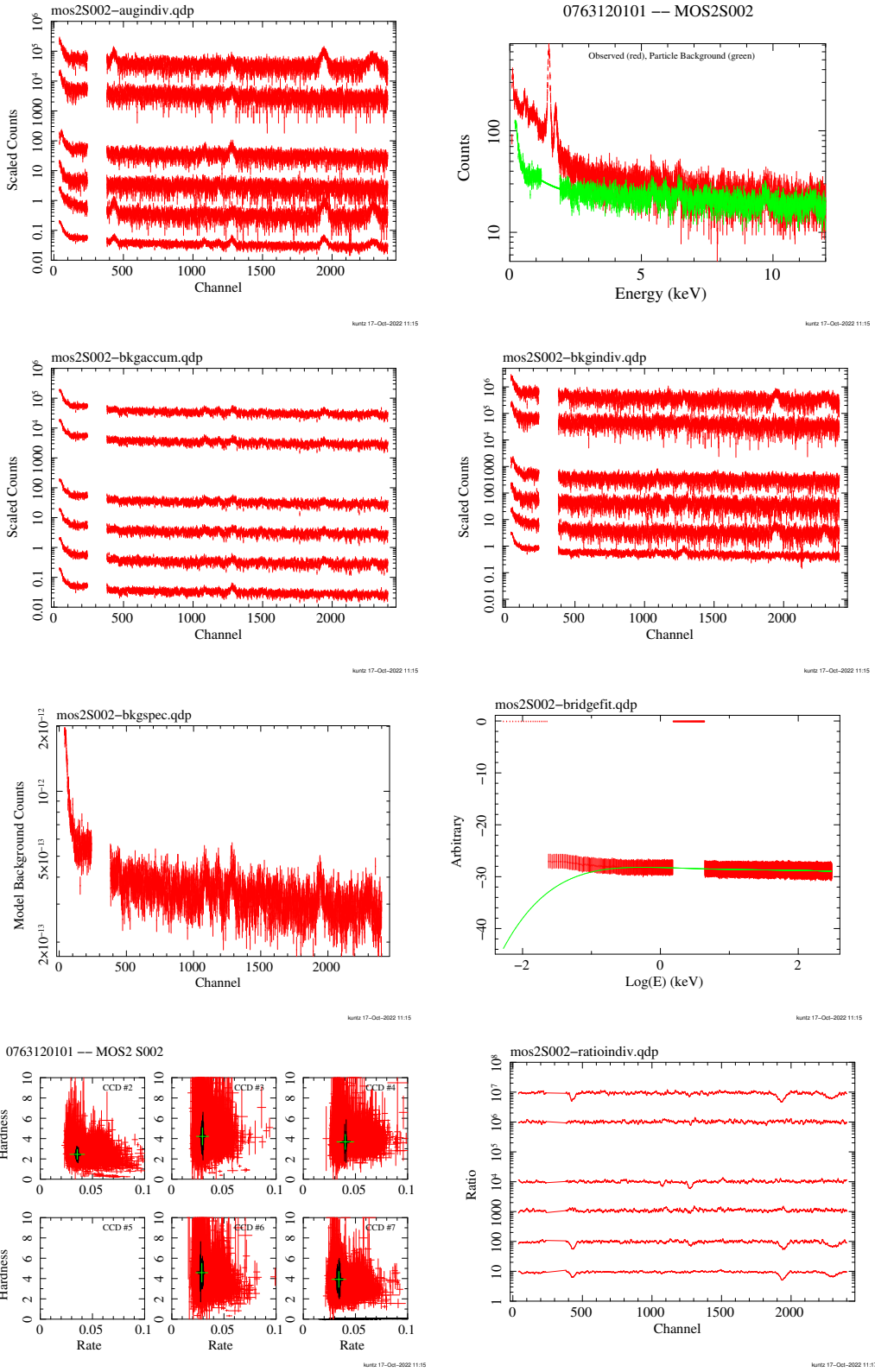


Fig. 38.— These are the figures produced by `mosback`. From upper left to lower right: `-augindiv.ps`, `-augspec.ps`, `-bkgaccum.ps`, `-bkgindiv.ps`, `-bkgspec.ps`, `-bridgefit.ps`, `ratehard.ps`, and `-ratioindiv.ps`.

## A.12. MOSSPECTRA

---

### Description:

*mosspectra* processes the cleaned event file output to produce intermediate files for the creation of model particle background spectra and images by the FORTRAN program *mosback*. The user can select specific regions of the detector and specific energy bands to create spectra and images. With no additional region specification, *mosspectra* will create spectra and images for the entire FOV.

### Requirements:

The current location should be the working directory.

*espfilt* must have been successfully run. The light curve plot file should be examined to check whether there are any useful data for an individual exposure.

*cheese* must have been successfully run for point source exclusion.

The observation soft images should be checked to determine both if all CCDs were operating in imaging mode or if they are in an anomalous mode. If they are not operating or if they are operating but in an anomalous mode they should be deselected using the `ccds` parameter.

### Calling Parameters:

`eventfile` – Name of the input cleaned event file (e.g., `mos1S001-allevc.fits`). This file should contain both the FOV and the corners.

`cornerfile` – Name of the input cleaned corner event file. If this is unspecified, the program will look for `P-corevc.fits` file. If that file does not exist, it will create it de novo. **We strongly recommend allowing the program to form the corner event file de novo.**

`imagefile` – Name of the input image file. If this is unspecified, the program will use the file `P-fovimtfits`.

`expmap` – Name of the input exposure map file. If this is unspecified, the program will use the file `P-expimt.fits` created by *cheese*

`spmask` – Name of the output soft proton flare map file. If this is unspecified, the program will construct the file `P-fovimpsdet.fits`.

`mask` – If true, uses the contents of `maskdet` and `masksky` to apply masks.

`specfile` – Name of the output spectrum file. If this is unspecified, the program will construct the file `P-fovt.pi`

`rmffile` – Name of the output redistribution matrix file. If this is unspecified, the program will construct the file `P.rmf`.

`arffile` – Name of the output redistribution matrix file. If this is unspecified, the program will construct the file `P.arf`.

`keepinterfiles` – If true, several intermediate results files are kept. (Not currently active!)

`withregion` – If true, expects a user specified region file to be applied in addition to any other regions (such as the source removal mask, or the chip mask created from the `ccds` parameter.) The name of the file is specified in `regionfile`

`regionfile` – The name of the user specified region file to be applied if `withregion` is true.

`pattern` – The greatest value of `PATTERN` to be accepted. For MOS, this should be 12. For the pn this should be either 4 or 0.

`withsrcrem` – If true, expects a source region file to be applied to remove regions containing sources. The name of the file is specified in `maskdet`.

`maskdet` – Name of the input source removal region file (e.g., `mos1S001-bkgregtdet.fits`).

`masksky` – Name of the input source removal region file (e.g., `mos1S001-bkgregtsky.fits`).

`elow` – The low energy limit for the energy band in eV (e.g., 350).

`ehigh` – The high energy limit for the energy band in eV (e.g., 1100). If `elow` and `ehigh` are set to 0, the image processing will be eliminated and only spectral files will be produced.

`ccds` – The CCDs to be included, (e.g., "T T T F T T T").

`badpixelresolution` – The bad pixel resolution parameter to pass to `backscale`. The default is 16. The smaller the number, the greater the accuracy, but the slower the execution.

## Output:

`atthk.fits` – Attitude history file (if not already created by *cheese*).

`P-corccdN.pi` – The corner spectrum from the observation data from the individual CCDs. The ccd number, "N" in the file name, runs from 2 to 7.

`P-corfwccdN.pi` – The corner spectrum from the filter-wheel-closed data from the individual CCDs. The CCD number, "N" in the file name, runs from 2 to 7.

`P-fovccdN.pi` – The observation data spectrum from the selected region from the individual CCDs. The CCD number, "N" in the file name, runs from 1 to 7 including only the selected CCDs.

`P-fovfwccdN.pi` – The filter-wheel-closed data spectrum from the selected region from the individual CCDs. The CCD number, "N" in the file name, runs from 1 to 7.

`-fovt.pi` – The observation data spectrum from the selected region.

`P.arf` – The ARF file for the `P-fovt.pi` spectrum.

`P.rmf` – The RMF file for the `P-fovt.pi` spectrum.

`P-expimccd1-elow-ehigh.fits` – The exposure image for the observation data from the selected region for ccd #1 for the selected band. `elow` and `ehigh` indicate the band limits. This is a diagnostic image.

`P-expimsky-elow-ehigh.fits` – The exposure image for the observation data from the selected region for all selected CCDs for the selected band. `elow` and `ehigh` indicate the band limits. This file will be used by *combimage*.

`P-expimt.fits` – The exposure image for the observation data in sky coordinates from the field-of-view for all selected CCDs for the full energy band. This is primarily a diagnostic file.

`P-fwcimccdNdet-elow-ehigh.fits` – The image of the filter-wheel-closed data from the selected region from the individual CCDs for the selected band. The CCD number, "N" in the file name, runs from 1 to 7 and the band limits, `elow` and `ehigh` indicate the energy band. This is in detector coordinates.

`P-maskimccd1-elow-ehigh.fits` – The mask image for the observation data from the selected region for CCD #1 for the selected band. `elow` and `ehigh` indicate the band limits. This is a diagnostic image.

`P-maskimsky-elow-ehigh.fits` – The mask image for the observation data from the selected region for all selected CCDs for the selected band. `elow` and `ehigh` indicate the band limits. This image will be used by *combimage*.

`P-maskimt.fits` – The mask image for the observation data from the field-of-view for all selected CCDs for the full energy band. This is a diagnostic image.

`P-fovimccd1-elow-ehigh.fits` – The image of the observation data from the selected region for CCD #1 for the selected band. `elow` and `ehigh` indicate the band limits. This is a diagnostic file.

`P-fovimd-elow-ehigh.fits` – The image of the observation data in detector coordinates from the selected region for all selected CCDs for the selected band. `elow` and `ehigh` indicate the band limits.

`P-fovimsky-elow-ehigh.fits` – The image of the observation data in sky coordinates from the selected region for all selected CCDs for the selected band. `elow` and `ehigh` indicate the band limits. This file will be used by *combimage*.

`P-fovimpdet.fits` – Image of the selected region in detector coordinates. This image is used in the task *proton\_scale*.

`P-fovimt.fits` – Image of the selected region in sky coordinates in the total band. This image is diagnostic.

`P-corevc.fits` – Event file for the corner regions.

## Examples:

```
mosspectra eventfile=mos1S001-allevc.fits withregion=yes regionfile=regmos1.txt pattern=12
  withsrcrem=yes maskdet=mos1S001-bkgregtdet.fits elow=350 ehight=1100 ccds="T T T F F F T"
  (to create both the spectrum and the band image)
```

```
mosspectra eventfile=mos1S001-allevc.fits withregion=yes regionfile=regmos1.txt pattern=12  
withsrcrem=yes maskdet=mos1S001-bkgregtdet.fits elow=0 ehigh=0 ccds="T T T F F F T" (to  
create the spectrum but not the band image)
```

## A.13. PNBACK

---

### Description:

*pnback* creates model particle background spectra and images (if selected with a non-zero energy range) for the selected region from the intermediate files produced from *pnspectra*. The resultant image is in detector coordinates which can be transformed into sky coordinates by the task *rotdet2sky*. *pnback* creates several QDP plot files which show the source and model background spectra for the observation and other diagnostic information. Any enhancement of the data over the particle background model at higher energies probably indicates residual soft proton contamination, unless there are really hard and bright sources in the field that haven't been masked.

### Requirements:

The current location should be the working directory.

*pnspectra* must have been successfully run at least once for the exposures to be used.

### Calling Parameters:

**inspecfile** – Name of the input spectrum file from *pnspectra*, (e.g., `pnS003-fov.t.pi`).

**inspecoot** – Name of the input OOT spectrum file from *pnspectra*, (e.g., `pnS003-fovtoot.pi`).

**outspecfile** – Name of the output spectrum file. If it is unspecified, the program will construct the file `P-bkg.pi`.

**outspecoot** – Name of the output OOT and background subtracted spectrum file. If it is unspecified, the program will construct the file `P-fovtootsub.pi`.

**rmffile** – Name of the input redistribution matrix file. If it is unspecified, the program will use the file `P.rmf`.

**withplotfiles** – If true, creates diagnostic QDP plot files.

**inimagfile** – Template image for background image production. If it is unspecified, the program uses `P-fovimgdet-elow-ehigh.fits`.

**inimagoot** – Template image for background image production. If it is unspecified, the program uses `P-fovimgdetoot-elow-ehigh.fits`.

**outingfile** – Name of the output quiescent particle background image. If it is unspecified, the program will construct the file `P-bkgimgdet-elow-ehigh.fits`.

**elow** – The low energy limit for the energy band in eV (e.g., 350).

**ehigh** – The high energy limit for the energy band in eV (e.g., 1100). If **elow** and **ehigh** are set to 0, the image processing will be eliminated and only spectral files will be produced.

**quads** – The quadrants to be included, (e.g., "T T T T").

### Output:

**P-bkg.pi** – The model particle background spectrum for the P exposure and the selected region.

**P-fovtootsub.pi** – Out of time subtracted spectrum for the selected region.

**P-bkgimgdet-elow-ehigh.fits** – The model particle background image for the P exposure, selected energy band (**elow** and **ehigh**), and the selected region. The image is in detector coordinates.

**P-augratehrev.qdp** – A QDP plot file showing the hardness ratios from the calibration data base with the selected ObsIDs highlighted for the four quadrants. Produced if **withplotfiles** is true.

**P-augraterev.qdp** – A QDP plot file showing the count rates from the calibration data base with the selected ObsIDs highlighted for the four quadrants. Produced if **withplotfiles** is true.

**P-augspec.qdp** – A QDP plot file showing the augmented spectra for the four quadrants. Produced if **withplotfiles** is true.

**P-bkgspec.qdp** – A QDP plot file showing the background spectrum. Produced if **withplotfiles** is true.

**P-bridgefit.qdp** – A QDP plot file showing the bridge fitting across the strong instrumental lines. Produced

if `withplotfiles` is true.

`P-quadspec.qdp` – A QDP plot file showing the background spectra for each quadrant. Produced if `withplotfiles` is true.

`P-ratehard.qdp` – A QDP plot file showing the selected region of hardness/count rate distributions for the four quadrants. Produced if `withplotfiles` is true.

`P-spec.qdp` – A QDP plot file showing the source spectrum and the background spectrum. Produced if `withplotfiles` is true.

## Examples:

```
pnback inspecfile=pnS003-fovvt.pi inspecoot=pnS003-fovvtoot.pi elow=350 ehigh=1100 quads="T T  
T T" (to create both the spectrum and band image)
```

```
pnback inspecfile=pnS003-fovvt.pi inspecoot=pnS003-fovvtoot.pi elow=0 ehigh=0 quads="T T T T"  
(to create the spectrum but not the band image)
```

```
pnS003-augratehrev.eps pnS003-augspec.eps pnS003-bridgefit.eps pnS003-ratehard.eps pnS003-  
augraterrev.eps pnS003-bkgspec.eps pnS003-quadspec.eps pnS003-spec.eps
```

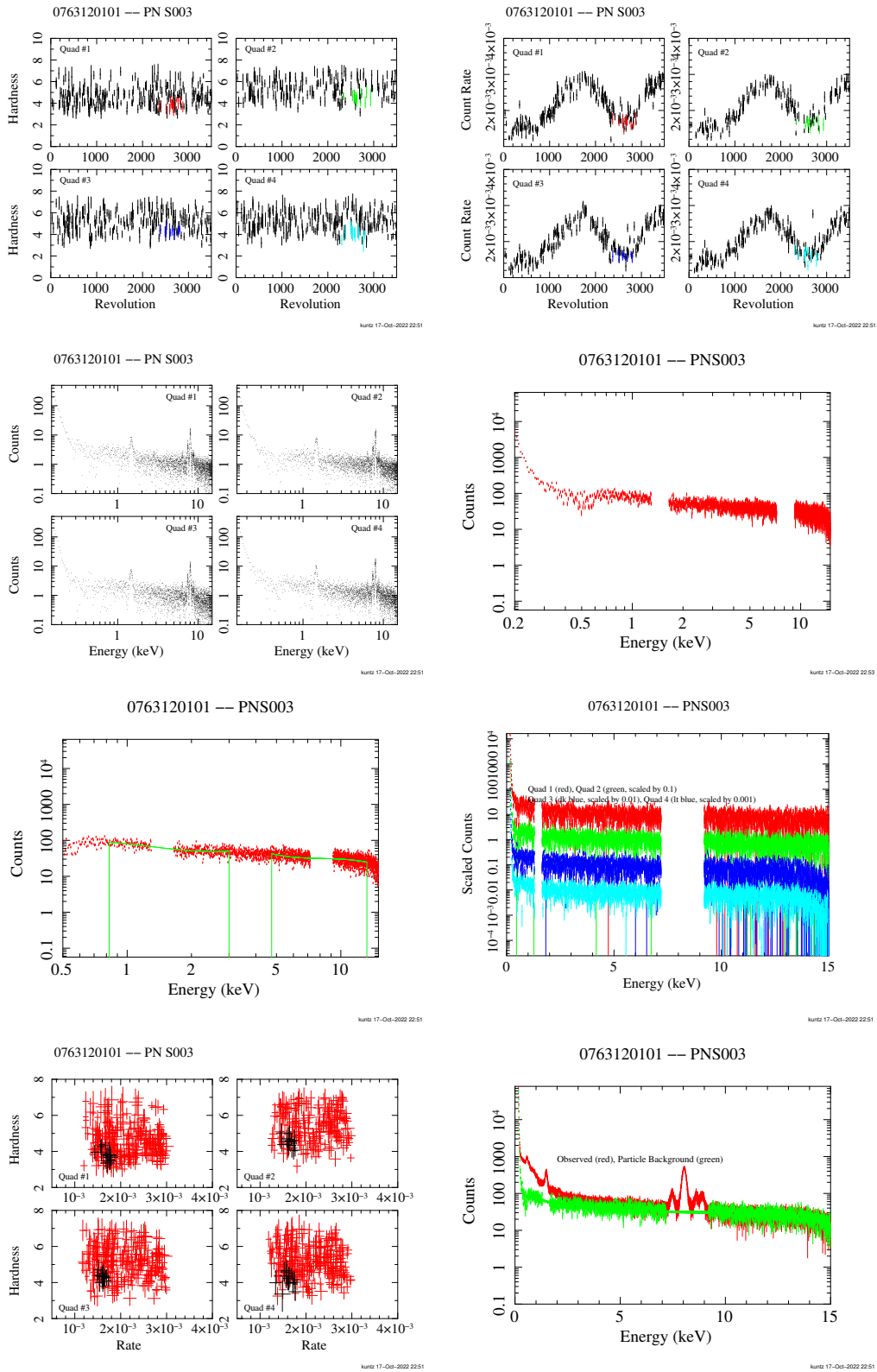


Fig. 39.— These are the figures produced by pnbac. From upper left to lower right: -augratehrev.eps, -augraterev.eps, -augspec.eps, -bkgspec.eps, -bridgefit.eps, -quadspec.eps, -ratehard.eps, and -spec.eps.



## A.14. PNSPECTRA

---

### Description:

*pnspectra* processes the cleaned event file output from *espfilt* to produce intermediate files for the creation of model particle background spectra and images by *pnback*. The user can select specific regions of the detector and specific energy bands to create spectra and images. With no additional region specification, *pnspectra* will create spectra and images for the entire FOV.

### Requirements:

*espfilt* must have been successfully run. The light curve plot file should be examined to check whether there are any useful data for an individual exposure.

*cheese* must have been successfully run for point source exclusion.

The observation image should be checked to determine if all quadrants were operating in imaging mode. If not, they should be selected against using the `quads` parameter.

### Calling Parameters:

`eventfile` – Name of the input cleaned event file (e.g., `pnS003-allevc.fits`). This file should contain both the FOV and the corners.

`ootevtfile` – Name of the input cleaned OOT event file (e.g., `pnS003-allevcoot.fits`). This file should contain both the FOV and the corners.

`cornerfile` – Name of the input cleaned corner event file. If this is unspecified, the program will look for `P-corevc.fits` file. If that file does not exist, it will create it de novo. **We strongly recommend allowing the program to form the corner event file de novo.**

`ootcornfile` – Name of the input cleaned OOT corner event file. If this is unspecified, the program will look for `P-corevc.fits` file. If that file does not exist, it will create it de novo. **We strongly recommend allowing the program to form the corner event file de novo.**

`imagefile` – Name of the output image file. If this is unspecified, the program will use the file `P-fovimt.fits`.

`ootimagefile` – Name of the output OOT image file. If this is unspecified, the program will use the file `P-fovimtoot.fits`.

`expmap` – Name of the input exposure map file. If this is unspecified, the program will use the file `P-expimt.fits` created by *cheese*.

`spmask` – Name of the output soft proton flare map file. If this is unspecified, the program will construct the file `P-fovimspdets.fits`.

`mask` – If true, uses the contents of `maskdet` and `masksky` to apply masks.

`specfile` – Name of the output spectrum file. If this is unspecified, the program will construct the file `P-fovtpi`.

`ootspecfile` – Name of the output spectrum file. If this is unspecified, the program will construct the file `P-fovtootpi`.

`rmffile` – Name of the output redistribution matrix file. If this is unspecified, the program will construct the file `P.rmf`.

`arffile` – Name of the output redistribution matrix file. If this is unspecified, the program will construct the file `P.arf`.

`keepinterfiles` – If true, several intermediate results files are kept. (Not currently active!)

`withregion` – If true, expects a user specified region file to be applied in addition to any other regions (such as the source removal mask, or the chip mask created from the `ccds` parameter.) The name of the file is specified in `regionfile`.

`regionfile` – The name of the user specified region file to be applied if `withregion` is true.

`pattern` – The greatest value of `PATTERN` to be accepted. For MOS, this should be 12. For the pn this should be either 4 or 0.

`withsrcrem` – If true, expects a source region file to be applied to remove regions containing sources. The name

of the file is specified in `maskdet`.

`maskdet` – Name of the input source removal region file (e.g., `mos1S001-bkgregtdet.fits`).

`masksky` – Name of the input source removal region file (e.g., `mos1S001-bkgregtdet.fits`).

`elow` – The low energy limit for the energy band in eV (e.g., 350).

`ehigh` – The high energy limit for the energy band in eV (e.g., 1100). If `elow` and `ehigh` are set to 0, the image processing will be eliminated and only spectral files will be produced.

`pattern` – The maximum pattern to be accepted. Should be 0 or 4.

`quads` – The quadrants to be included, (e.g., "T T T T").

`badpixelresolution` – The bad pixel resolution parameter to pass to `backscale`. The default is 16. The smaller the number, the greater the accuracy, but the slower the execution.

## Output:

`atthk.fits` – Attitude history file (if not already created by *cheese*).

`P-corqQ.pi` and `P-corootqQ.pi` – The corner spectrum from the observation data from the individual CCDs. The quadrant number, "Q" in the file name, runs from 1 to 4.

`P-corfwcqQ.pi` and `P-corfwcootqQ.pi` – The corner spectrum from the filter-wheel-closed data from the individual quadrants. The quadrant number, "Q" in the file name, runs from 1 to 4.

`P-fovqQ.pi` and `P-fovootqQ.pi` – The observation data spectrum from the selected region from the individual quadrants. The quadrant number, "Q" in the file name, runs from 1 to 4 including only the selected quadrants.

`P-fovfwcqQ.pi` and `P-fovfwcootqQ.pi` – The filter-wheel-closed data spectrum from the selected region from the individual quadrants. The quadrant number, "Q" in the file name, runs from 1 to 4.

`-fovt.pi` and `-fovtoot.pi` – The observation data spectrum from the selected region.

`P.arf` – The ARF file for the `P-fovt.pi` spectrum.

`P.rmf` – The RMF file for the `P-fovt.pi` spectrum.

`P-expimsky-elow-ehigh.fits` – The exposure image for the observation data from the selected region for all selected CCDs for the selected band. `elow` and `ehigh` indicate the band limits. This file will be used by *combimage*.

`P-exp-imt.fits` – The exposure image for the observation data in sky coordinates from the field-of-view for all selected CCDs for the full energy band. This is primarily a diagnostic file.

`P-fwcimqQdet-elow-ehigh.fits` and `P-fwcimootqQdet-elow-ehigh.fits` – The image of the filter-wheel-closed data from the selected region from the individual quadrants for the selected band. The quadrant number, "Q" in the file name, runs from 1 to 4 and the band limits, `elow` and `ehigh` indicate the energy band. This is in detector coordinates.

`P-maskimsky-elow-ehigh.fits` – The mask image for the observation data from the selected region for all selected quadrants for the selected band. `elow` and `ehigh` indicate the band limits. This image will be used by *combimage*.

`P-maskimt.fits` – The mask image for the observation data from the field-of-view for all selected quadrants for the full energy band. This is a diagnostic image.

`P-fovimd-elow-ehigh.fits` and `P-fovimdoot-elow-ehigh.fits` – The image of the observation data in detector coordinates from the selected region for all selected quadrants for the selected band. `elow` and `ehigh` indicate the band limits.

`P-fovimsky-elow-ehigh.fits` and `P-fovimskyoot-elow-ehigh.fits` – The image of the observation data in sky coordinates from the selected region for all selected quadrants for the selected band. `elow` and `ehigh` indicate the band limits. This file will be used by *combimage*.

`P-fovimpdet.fits` – Image of the selected region in detector coordinates. This image is used in the task *proton\_scale*.

`P-fovimt.fits` – Image of the selected region in sky coordinates in the total band. This image is diagnostic.

`P-corevc.fits` and `P-corevcoot.fits` – Event file for the corner regions.

## Examples:

```
pnspectra eventfile=pnS003-allevc.fits otevtfile=pnS003-allevcoot.fits
  keepinterfiles=yes withregion=yes regionfile="regpn.txt" pattern=4 withsrcrem=yes
  maskdet=pnS003-bkgregtdet.fits withinimages=yes elow=350 ehigh=1100 quads="T T T T" (to
  create both the spectrum and the band image)
```

```
pnspectra eventfile=pnS003-allevc.fits otevtfile=pnS003-allevcoot.fits
  keepinterfiles=yes withregion=yes regionfile="regpn.txt" pattern=4 withsrcrem=yes
  maskdet=pnS003-bkgregtdet.fits withinimages=no elow=0 ehigh=0 quads="T T T T" (to create
  the spectrum but not the band image)
```

## A.15. POINTSRC

---

### Description:

*pointsrc* calculates the appropriate extragalactic background scale factor for Xspec for several different models and user-selected parameters including the excision of point sources to a specified source flux.

### Requirements:

No specific requirements.

### Calling Parameters:

**func** – Model of the extragalactic X-ray luminosity function to be used for the calculation:

hms (Hasinger, Miyaji, & Schmidt 2005)

mushotzky (Mushotzky et al. 2000)

cappelluti (Cappelluti et al. 2008)

mateos (Mateos et al. 2008)

**minsrcflux** – Minimum source flux in units of  $\text{ergs cm}^{-2} \text{s}^{-1}$

**cxbnorm** – Cosmic X-ray background normalization in units of  $\text{photons cm}^{-2} \text{s}^{-1} \text{sr}^{-1} \text{keV}^{-1}$  at 1 keV.

**index** – Optional parameter for index of unresolved spectrum defaults to 1.46.

### Output:

Xspec normalization for the extragalactic power law in units of  $\text{photons cm}^{-2} \text{s}^{-1} \text{arcmin}^{-2} \text{keV}^{-1}$ .

### Example:

```
point-source func=mateos minsrcflx=1.e-14 cxbnorm=10.6 index=1.46
```

## A.16. PROTON

---

### Description:

*proton* uses the spectral fitting results from Xspec and model soft proton detector maps to create model soft proton contamination maps for a given observation.

### Requirements:

The current location should be the working directory.

*mosspectra* and *mosback* or *pnspectra* and *pnback* must have been successfully run for the exposure to be used.

The spectra produced by these tasks must be fit to determine the spectral parameters of the soft proton contamination. Currently the spectrum must be modeled as a power law or a broken power law not folded through the instrumental effective areas. In Xspec V11 this is accomplished by including *pow/b* or *bknpow/b* in the fitted model. For Xspec V12 this requires adding an additional model and response where the response is a diagonal matrix which is included in the XMM-ESAS calibration files.

The input spectral parameters are simply the fitted values from Xspec (assuming that no additional normalization has been applied to the component).

### Calling Parameters:

*imagefile* – Input object image file in detector coordinates.

*specfile* – File name of spectrum file used in the spectral fit to determine the residual SP contamination.

*withsky* – Parameter that does not currently work. If set to ‘yes’, is supposed to rotate the image from detector to sky coordinates.

*spmapdet* - File name for the output SPF map in detector coordinates.

*ccds* – These are flags to include or exclude individual CCDs for the MOS detectors (e.g., “T T T T T T”), or to include or exclude the quadrants of the pn (e.g., “T T T T”).

*elow* – The low energy limit for the energy band in eV (e.g., *elow*=400).

*ehigh* – The high energy limit for the energy band in eV (e.g., *ehigh*=1250).

*speccontrol* – 1 for a power law model, 2 for a broken power law.

*spectrumcontrol*=1

*pindex* – The fitted power law index.

*pnorm* – The fitted scale factor for the power law.

*spectrumcontrol*=2

*bindl* – The fitted lower power law index.

*bbreak* – The break energy, typically fixed to the default value of 3.0.

*bindh* – The fitted upper power law index.

*bnorm* – The fitted scale factor for the power law.

### Output:

*prefix-protimdet-elow-ehigh.fits* – The model soft proton background image count image in detector coordinates.

### Example:

```
proton imagefile=mos1S001-fovmdet-350-1100.fits specfile=mos1S001-fovvt.pi elow=350
      ehigh=1100 ccds="T T T F F F T" speccontrol=1 pindex=0.1 pnorm=2.208e-2
```

```
proton imagefile=mos1S001-fovmdet-350-1100.fits specfile=mos1S001-fovvt.pi elow=350
      ehigh=1100 ccds="T T T F F F T" speccontrol=2 bindl=0.85 bbreak=2.0 bindh=1.42
      bnorm=0.025
```

## A.17. PROTONSCALE

---

### Description:

*protonscale* extracts the BACKSCAL keyword and average total soft proton intensity from the detector map for a specified region. These values are necessary in the spectral fitting process of multiple regions which are linked.

### Requirements:

The current location should be the working directory.

*mos-spectra* must have been successfully run for the exposure to be used.

### Calling Parameters:

**mode** – Set mode for single or list input.

1: do a single region

2: do multiple regions with the required input provided in a text file.

For **mode=1**

**maskfile** – File name for the mask file. This is the, e.g., *mosprefix-fovimspdets.fits* file produced for the region by *mosspectra*.

**specfile** – File name for the spectral file for the region.

For **mode=2**

**spfile** – ASCII text file with the input for multiple regions. The file should contain, on separate lines, the mask file name (**maskfile**) and spectrum file name (**specfile**) for each region. For example:

```
pnS005-fovimspdets-0-15.fits
pnS005-fov-0-15-grp.pi
pnS005-fovimspdets-15-60.fits
pnS005-fov-15-60-grp.pi
pnS005-fovimspdets-120-150.fits
pnS005-fov-120-150-grp.pi
```

### Output:

Screen output only –

**mode=1**: BACKSCAL keyword value converted to units of arcmin<sup>-2</sup> and the average soft proton flux.

**mode=2**: Ordered pairs of the BACKSCAL keyword value converted to units of arcmin<sup>-2</sup> and the relative soft proton value normalized to that of the first region.

### Example:

```
protonscale maskfile=mos1S001-fovimspdets.fits specfile=mos1S001-fov.pi
protonscale spfile=spec-ann.txt
```

## A.18. ROTDET2SKY

---

### Description:

*rotdet2sky* uses information from the previously created count image in sky coordinates to rotate the detector coordinate model QPB images produced by *mosback* and *pnback*, model SP background images produced by *proton*, and model SWCX background images produced by *swcx* into images in sky coordinates.

### Requirements:

The current location should be the working directory.

*mosspectra* or *pnspectra* and *mosback* or *pnback* must have been successfully run for the recasting of the particle background.

*proton* must have been successfully run for the recasting of the soft proton background.

*swcx* must have been successfully run for the recasting of the swcx component of the background.

### Calling Parameters:

**intemplate** – The template sky coordinate image to be matched.

**inimage** – The detector coordinate image to be rotated.

**outimage** – The output detector coordinate image rotated into the sky coordinate frame set by **intemplate**.

**withmask** – True if a mask file is to be applied.

**inmaskimage** – The name of the mask file to be applied.

**withdety** – To set the detector reference pixel location (i.e., CRPIX1, CRPIX2) manually in the detector coordinates. The use of this parameter is not recommended unless you really know what you are doing.

**dety** – The DETY reference pixel location (obtained by running *esky2det*).

**dety** – The DETY reference pixel location (obtained by running *esky2det*).

**withskyxy** – To set the sky reference pixel location (i.e., CRPIX1, CRPIX2) manually in the sky coordinates. The use of this parameter is not recommended unless you really know what you are doing.

**skyx** – The X location of the reference pixel (CRPIX1)

**skyy** – The Y location of the reference pixel.

### Output:

*prefix-bkgimsky-elow-ehigh.fits* – The QPB count image in sky coordinates.

*prefix-protimsky-elow-ehigh.fits* – The SP background count image in sky coordinates.

*prefix-swcximsky-elow-ehigh.fits* – The SWCX background count image in sky coordinates.

### Example:

```
rotdet2sky intemplate=pnS003-fovimsky-350-1100.fits inimage=pnS003-protimdet-350-1100.fits
outimage=pnS003-protimsky-350-1100.fits dety=-1103.3 dety=1473.3 withskyxy=no
```

## A.19. SPPARTIAL

---

### Description:

*sppartial* uses information from the spectral fit of a limited region of the detector and from the full field-of-view to scale the fitted SP normalization of the limited region to be appropriate for the full FOV. This is useful for the case where bright diffuse emission or the effects of very bright point sources in part of the FOV may be affecting the SP spectral fit, for example many clusters of galaxies or LMC X-1.

### Requirements:

The current location should be the working directory.

*mosspectra* or *pnspectra* will need to have been run twice, once for the full field of view, once for a limited source-free region. This requires a number of files to be renamed or else they will be overwritten (the spectra and soft proton template files, see the image example).

### Calling Parameters:

**fullimage** – File name for the mask file of the full FOV. This is the *mosprefix-fovimpdet.fits* file produced for the region by *mosspectra*.

**fullspec** – File name for the spectral file for the full FOV.

**regimage** – File name for the mask file of the limited region. This is the *mosprefix-fovimpdet.fits* file produced for the limited region by *mosspectra*.

**regspec** – File name for the spectral file for the limited region.

**regnorm** – The fitted SP normalization from the limited region.

### Output:

Scaled value for the SP normalization.

### Example:

```
sppartial fullimage=mos1S001-fovimpdet-full.fits fullspec=mos1S001-full.pi  
regimage=mos1S001-fovimpdet-ann.fits regspec=mos1S001-ann.pi rnorm=0.029
```



## A.20. SWCX

---

### Description:

*swcx* uses the spectral fitting results from Xspec and model solar wind charge exchange (SWCX) detector maps (simply narrow band exposure maps created on a grid of energies) to create model SWCX contamination maps for a given observation. *swcx* includes the line-broadening effect of the response matrix (RMF).

Note that the SWCX lines are typically those found in the thermal spectra such as O VII and O VIII. Because of this, great care must be used in the determination of the SWCX contribution so that power isn't shifted to or from the cosmic plasma. This can best be accomplished where the object of interest does not fill the entire field of view or where there are multiple observations of the field. In the latter case the SWCX contribution can at least be reduced to the minimum level of contamination among the multiple observations by fitting their spectra simultaneously.

### Requirements:

The current location should be the working directory.

*mos-spectra* and *mos-back* and/or *pn-spectra* and *pn-back* must have been successfully run for the exposure to be used. The spectra produced by these tasks must be fit to determine the spectral parameters of the SWCX contamination. The spectrum should be modeled as a Gaussian of narrow or zero width (depending on whether the line is actually a complex of lines, e.g., O VII).

### Calling Parameters:

**imagefile** – Image created by *mosback* or *pnback*.

**specfile** – Spectrum created by *mosback* or *pnback*.

**swcxmapdet** – Output SWCX map in detector coordinates.

**ccds** – These are flags to include or exclude individual CCDs for the MOS detectors (e.g., “T T T T T T”), or to include or exclude the quadrants of the pn (e.g., “T T T T”).

**elow** – The low energy limit for the energy band in eV (e.g., **elow=400**).

**ehigh** – The high energy limit for the energy band in eV (e.g., **ehigh=1250**).

**lines** – list of SWCX lines to cast, for example  $E \sim 0.56$  keV for O VII,  $E \sim 0.65$  keV for O VIII,  $E \sim 0.81$  keV for O VIII,  $E \sim 0.91$  keV for Ne IX, and  $E \sim 1.34$  keV for Mg XI.

**gnorms** – list of fitted Gaussian scale factors from Xspec in the same order as **linelist**.

**arffile** – ARF for the exposure.

**rmffile** – RMF for the exposure.

### Output:

*prefix-swcximdet-elow-ehigh.fits* – The model SWCX background image count image in detector coordinates.

### Example:

```
swcx imagefile=pn-fovmdet-300-1100.fits specfile=pn-fovvt.pi ccds="T T T T" elow=300
      ehigh=1100 rmffile=pn.rmf arffile=pn.arf lines="OVII OVIII" gnorms="1.69938e-7
      1.75262e-8"
```

## B. EXAMPLES

The following pages contain an example to demonstrating how to do simple reduction to obtain both a spectrum and an image of an observation. As a useful guide, we include here a list of routines used for the reduction of several observations, and approximately how long it took to execute each.

Routine Timing		
Routine	Time	Notes
cifbuild	0:20	
odfingest	0:56	
enchain	3:08	
epchain	12:39	Run twice, once for the OOT
emanom	0.01	
espfilt	0:05	mos in histogram mode
espfilt	0:30	pn in histogram mode
espfilt	~5:15	mos in ratio mode
cheese	10:05	
mosspectra	19:04	
mosback	0:04	
pnspectra	47:28	
pnback	0:07	
rotim2det	~0:01	
proton	0:04	
swcx	0:01	
combimage	~0:02	
adapt	0:26	actually doing adaptive smoothing

<sup>a</sup> All current routines are in SAS-consistent FORTRAN.

## B.1. A Simple Single Observation, Spectroscopy and Imaging

By a simple observation, we mean an observation *like* a “blank-sky” observation, where the residual soft proton contamination can be fit from a spectrum extracted from the entire FOV. This section provides a recipe for extracting the spectrum and constructing an image of the entire FOV. If you have a big cluster in the middle of your FOV, you’ll want to follow the recipe in this section to extract the raw image, the quiescent particle background image, and the spectrum of the entire FOV, but you will want to follow the recipe in the following section to extract the spectrum from the region least dominated by the cluster and, after fitting that, to construct the soft proton image.

One of the things that you will notice is that this script does a large amount of logging, so you can retain a record of what actually happened, and listing, so you know what files were produced by which routine. Given the number of files that SAS and ESAS produces, it’s often a good thing to keep track of. This script also uses a number of environment variables to reduce the amount of typing (and errors).

```
#####
###          THIS IS THE XMM-ESAS SCRIPT KDK-RUN          ###
#####
# SAS V21
#
# This script is designed to run under MACOSX. It assumes that SAS has been set up, but sets several
# of the SAS parameters for the specific observation. The required data files are in directories which
# are inputs to scripts and programs which use them.
#
# All ODF data have been placed in the directory
#   /path/ODF
# but nothing further has been done to them
#
# All processing takes place in the directory
#   /path/analysis
#
# Date: 11/09/22 but with constant modifications
# Purp: this script is for <insert your identifier> to be run in the "analysis" directory
# Note: this script is meant to be modified by the user for their own purpose
#       There are many things in here that may not be useful for you particular analysis,
#       Just delete what you don't need!
#       This script lists the files repeatedly. That is so you can compare the lists to
#       determine what was made in a particular step...in case you need to delete them.

# Modify the following lines before you do anything else to make sure that you are pointing to
# the correct directory

# Set a few SAS environment variables

setenv OBSID 0000000001
setenv SAS_CCFPATH /Users/kuntz/Soft/sas_ccf
setenv SAS_CCF /Users/kuntz/Data/xmm_proj/${OBSID}/analysis/ccf.cif
setenv SAS_ODF /Users/kuntz/Data/xmm_proj/${OBSID}/ODF

# Cleaning up the ODF directory - this only has to be done once
# IF
# the data downloaded from the HEASARC:

cd $SAS_ODF
gzip -d *.gz

# ENDIF ELSE downloaded from the SOC
```

```

tar -xvf *.tar # for an ODF extracted from XSA this may have to be run several times
rm *.SAS      # this is a "just in case" step which would be needed
              # if you have moved the directory since the last odffingest

# ENDELSE
# End of cleaning up the ODF

cd ..
mkdir analysis
cd analysis

# Run the initial tasks to set up SAS for this ObsID - that is - set up the CCF
# Here is where you want to make sure that there is no *.SAS file in the ODF directory

cifbuild withccfpath=no analysisdate=now category=XMMCCF calindexset=$SAS_CCF fullpath=yes verbosity=1

odffingest odffdir=$SAS_ODF outdir=$SAS_ODF verbosity=1

# At this point there should be only the ccf.cif in the analysis directory

# Run epchain and emchain to create calibrated photon event files using the latest calibrations.
# emchain will process all sub-observations for both MOS detectors
# epchain will process only the first sub-observation
# and will need to be run for each pn sub-observation
# epchain will need to be run a second time for each sub-observation
# to produce the OOT event files

emchain verbosity=1 >& emchain.log

# While emchain processes all segments automatically, epchain only processes the first pn segment
# automatically. Further segments require specifying the segment number.
# To determine the number of pn segments to process:

epchain exposure=90

# This will end with an error since there is no exposure with index 90. However,
# the output will show you what exposures exist for the pn.
# For each pn segment, where the argument to "exposure" is the index of the exposure

epchain exposure=1 verbosity=1 >& epchain_1.log
epchain exposure=1 withoutoftime=true verbosity=1 >& epchain_oot_1.log

# Clean up after the chains since the chains produce many files that you won't need

ls -l *.* > post_chain.list
mkdir chains
mkdir logs
mv *.FIT chains
cp chains/*IEVLI*.FIT .
cp chains/*OOEVLI*.FIT .
mv *.log logs
mv *.list logs

# At this point you should determine the filter and exposure time for each segment, that is, for each
# IEVLI file determine whether it should be retained or deleted.
# You may find it useful to create a routine to extract information from all of the
# *IEVLI* files the way I have:

#P0000000001M1S001MIEVLI0000.FIT 336.86800 61.02600 Thin1 PrimeFullWindow 35.51
#R0000000001M1U002MIEVLI0000.FIT 336.86800 61.02600 CalClosed PrimeFullWindow 0.46

```

```

#P0000000001M1U003MIEVLI0000.FIT 336.86800 61.02600 Thin1 PrimeFullWindow 10.26
#P0000000001M1U004MIEVLI0000.FIT 336.86800 61.02600 CalClosed PrimeFullWindow 0.64
#P0000000001M1U005MIEVLI0000.FIT 336.86800 61.02600 Thin1 PrimeFullWindow 1.26
#P0000000001M2S002MIEVLI0000.FIT 336.86800 61.02600 Thin1 PrimeFullWindow 47.32
#P0000000001M2U002MIEVLI0000.FIT 336.86800 61.02600 CalClosed PrimeFullWindow 0.11
#P0000000001M2U005MIEVLI0000.FIT 336.86800 61.02600 Thin1 PrimeFullWindow 1.21
#P0000000001PNS003PIEVLI0000.FIT 336.86800 61.02600 Thin1 PrimeFullWindowExtended 33.36

# The event files for the "CalClosed" segments are not useful and can be deleted.

rm P${OBSID}M1U002MIEVLI0000.FIT
rm P${OBSID}M1U004MIEVLI0000.FIT
rm P${OBSID}M2U002MIEVLI0000.FIT

# It is often a good idea to remove really short observations, usually <5 ks

rm P${OBSID}M1S002MIEVLI0000.FIT
rm P${OBSID}M2S003MIEVLI0000.FIT

# For this observations the useful segments are mos1S001, (possibly) mos1U003,
# mos2S002, and pnS003.

# If there are multiple segments for a given detector, you may want to merge them here, but it is
# probably better to proceed with the emanom and espfilt steps, and only merge if both segments
# should be retained

merge set1=P${OBSID}M1S001MIEVLI0000.FIT set2=P${OBSID}M1U003MIEVLI0000.FIT outset=P${OBSID}M1T000MIEVLI0000.FIT

# Rename the files that you need for ease and clarity
# These lines work if you used the "chains", but won't if you used the "procs"
# Please don't try to give segments new numbers...many of the ESAS programs
# look for the segment number in the header and use it to create output file names.

mv P${OBSID}M1T000MIEVLI0000.FIT mos1S001.fits
mv P${OBSID}M2S002MIEVLI0000.FIT mos2S002.fits
mv P${OBSID}PNS00300EVLIO0000.FIT pnS003-oot.fits
mv P${OBSID}PNS003PIEVLI0000.FIT pnS003.fits

# Set up some system variables so you don't have to go through this script,
# changing every "mos1S001" to "mos1U017", etc.

setenv M1 "mos1S001"
setenv M2 "mos2S002"
setenv PN "pnS003"

ls -l *.* > post_chain_clean.list

# Take a look at the data to see what you've got

evselect table=${M1}.fits withimageset=yes imageset=${M1}-diag-det-unfilt.fits
expression="(PI in [300:1000])&&(PATTERN<=12)&&((FLAG & 0x766aa000)==0)" filtertype=expression
ignorelegallimits=yes imagebinning=imageSize xcolumn=DETX ximagesize=780 ximagemax=19500
ximagegin=-19499 ycolumn=DETY yimagesize=780 yimagemax=19500 yimagegin=-19499

evselect table=${M2}.fits withimageset=yes imageset=${M2}-diag-det-unfilt.fits
expression="(PI in [300:1000])&&(PATTERN<=12)&&((FLAG & 0x766aa000)==0)" filtertype=expression
ignorelegallimits=yes imagebinning=imageSize xcolumn=DETX ximagesize=780 ximagemax=19500
ximagegin=-19499 ycolumn=DETY yimagesize=780 yimagemax=19500 yimagegin=-19499

evselect table=${PN}.fits withimageset=yes imageset=${PN}-diag-det-unfilt.fits

```

```

expression="(PI in [300:1000])&&(PATTERN<=4)&&(#XMMEA_EP)" ignorelegallimits=yes imagebinning=imageSize
xcolumn=DETX ximagesize=780 ximagemax=19500 ximagemin=-19499 ycolumn=DETY yimagesize=780
yimagemax=19500 yimagemin=-19499

ds9 *diag-det-unfilt.fits &

# Some clean-up of the diagnostic files (of course you can also just delete them)

mkdir diags
mv *diag-det-unfilt.fits diags

# Get the anomalous chip information - you should look at the screen output or
# the anom*.log files carefully so you will know which CCDs to deselect in some of the
# future steps (mosspectra and mosback)

emanom eventfile=${M1}.fits keepcorner=FALSE

emanom eventfile=${M2}.fits keepcorner=FALSE

# Determine which chips should be on, and build environment variables for them
# you'll be glad that you set these as environment variables later!
# These are examples only! (These get rid of chips that were problems in my obsid.)

setenv M1ON "T T F F T F T"
setenv M2ON "T T T T T T T"
setenv PNON "T T T T"

ls -l *.* > post_emanom.list
mv *-anom.log logs
mv *.list logs

# Soft proton filtering - this will remove periods of obvious soft proton flares from
# the event files. It will also produce QDP plot files showing the light curves and
# the accepted time intervals. Note that while much of the SP contamination is removed,
# there is likely to be some residual contamination.
# Note that a different rangescale is used for the pn than for the MOS!
# The most important output is the cleaned event file: x-allevc.fits
# For the pn, both a cleaned event file and a cleaned OOT event file are produced.
# Note that the energy range is chosen to maximize the soft proton/cosmic ratio.

espfilt eventfile=${M1}.fits method=histogram elow=2500 ehigh=8500 withsmoothing=yes smooth=51
rangescale=6.0 allowsigma=3.0 keepinterfiles=true > espfilt_mos1.log

espfilt eventfile=${M2}.fits method=histogram elow=2500 ehigh=8500 withsmoothing=yes smooth=51
rangescale=6.0 allowsigma=3.0 keepinterfiles=true > espfilt_mos2.log

espfilt eventfile=${PN}.fits method=histogram elow=2500 ehigh=8500 withsmoothing=yes smooth=51
rangescale=15.0 allowsigma=3 withoot=Y ootfile=${PN}-oot.fits keepinterfiles=true > espfilt_pn.log

# Look at the QDP output from espfilt. The plots should be examined to get an idea
# of the extent of any residual contamination. Asymmetry of the peaks in the count
# rate histograms or strong high count-rate tails often imply that there is going
# to be residual soft proton contamination.

qdp ${M1}-hist.qdp
/cps
exit
mv pgplot.ps ${M1}-hist.ps
qdp ${M2}-hist.qdp
/cps

```

```

exit
mv pgplot.ps ${M2}-hist.ps
qdp ${PN}-hist.qdp
/cps
exit
mv pgplot.ps ${PN}-hist.ps

# At this point you might want to check the diagnostic images.
# See whether the edge of the FOV is still visible in the cleaned images.
# It shouldn't be unless your extended source is brighter than the cosmic X-ray background.

ds9 *-allim.fits *-allimc.fits &

# IF
# the espfilt cleaning is not sufficient, you may edit the -gti.fits files by hand to produce a
# -gti-edit.fits file that does the cleaning that you want.
# You will need to apply the new -gti-edit.fits files to the original event files as follows:

evselect table=${M1}.fits withfilteredset=yes expression="(GTI(${M1}-gti-edit.fits,TIME))&&
((FLAG & 0x766aa000)==0)&&(PATTERN<=12)" filtertype=expression keepfilteroutput=yes updateexposure=yes
filterexposure=yes filteredset=${M1}-allevc.fits

evselect table=${M2}.fits withfilteredset=yes expression="(GTI(${M2}-gti-edit.fits,TIME))&&
((FLAG & 0x766aa000)==0)&&(PATTERN<=12)" filtertype=expression keepfilteroutput=yes updateexposure=yes
filterexposure=yes filteredset=${M2}-allevc.fits

evselect table=${PN}-oot.fits withfilteredset=yes expression="(GTI(${PN}-gti-edit.fits,TIME))
&&(#XMMEA_EP)&&(PATTERN<=4)" filtertype=expression keepfilteroutput=yes updateexposure=yes
filterexposure=yes filteredset=${PN}-allevcoot.fits

evselect table=${PN}.fits withfilteredset=yes expression="(GTI(${PN}-gti-edit.fits,TIME))&&
(#XMMEA_EP)&&(PATTERN<=4)" filtertype=expression keepfilteroutput=yes updateexposure=yes
filterexposure=yes filteredset=${PN}-allevc.fits

# ENDIF

# Note that the gti selection must be applied to both the pn and the pn-oot event files.
# Note that by default espfilt applies filters to both the flags and the patterns, which you must
# reproduce to create consistent files.

# Take a look at the data to see whether it's really cleaned.

evselect table=${M1}-allevc.fits withimageset=yes imageset=${M1}-diag-det-filt.fits
expression="((PI in [300:1000])&&(PATTERN<=12)&&((FLAG & 0x766aa000)==0))" filtertype=expression
ignorelegallimits=yes imagebinning=imageSize xcolumn=DETX ximagesize=780 ximagemax=19500
ximagemin=-19499 ycolumn=DETY yimagesize=780 yimagemax=19500 yimagemin=-19499

evselect table=${M2}-allevc.fits withimageset=yes imageset=${M2}-diag-det-filt.fits
expression="(PI in [300:1000])&&(PATTERN<=12)&&((FLAG & 0x766aa000)==0)" filtertype=expression
ignorelegallimits=yes imagebinning=imageSize xcolumn=DETX ximagesize=780 ximagemax=19500
ximagemin=-19499 ycolumn=DETY yimagesize=780 yimagemax=19500 yimagemin=-19499

evselect table=${PN}-allevc.fits withimageset=yes imageset=${PN}-diag-det-filt.fits
expression="(PI in [300:1000])&&(PATTERN<=4)&&(#XMMEA_EP)" ignorelegallimits=yes imagebinning=imageSize
xcolumn=DETX ximagesize=780 ximagemax=19500 ximagemin=-19499 ycolumn=DETY yimagesize=780
yimagemax=19500 yimagemin=-19499

ds9 *-diag-det-filt.fits
ls -l *.* > post_espfilt.list

```

```

# Clean up after espfilt

mv *-diag-* diags
mkdir espfilt
mv *.qdp espfilt
mv *-corev.fits espfilt
mv *-corimc.fits espfilt
mv *-corim.fits espfilt
mv *-corlc.fits espfilt
mv *-allim.fits espfilt
mv *-allimc.fits espfilt
mv *-fovlc.fits espfilt
mv *-hist.* espfilt
mv *-gti.fits espfilt
mv *-gti-edit.fits espfilt
mv *-gti.txt espfilt
mv *-corevc.fits espfilt
mv ${M1}.fits espfilt
mv ${M2}.fits espfilt
mv ${PN}.fits espfilt
mv ${PN}-oot.fits espfilt

ls -l *.* > post_espfilt_clean.list
mv *.log logs
mv *.list logs

# At this point all you need in your analysis directory are
# ${M1}-allevc.fits, ${M2}-allevc.fits, ${PN}-allevc.fits, & ${PN}-allevcoot.fits

# Source detection - cheese is named such because it creates masks that look much like Swiss cheese.
# Here we are running it on two bands, but it can run for only one band as well.
# If run for two bands, the program produces three bands: soft (s), hard (h), and total (t = s+h)
# (though no total band cheese is produced, it can be produced as the product of the soft and hard)
# If run for a single band, it is considered the total (t) band.

cheese mos1file=${M1}-allevc.fits mos2file=${M2}-allevc.fits pnfile=${PN}-allevc.fits
  pnootfile=${PN}-allevcoot.fits elowlist=350 ehighest=1100 scale=0.5 mlmin=10 ratetotal=0.2 dist=50.
  keepinterfiles=no -V 7 >& cheese.log

# Inspect the cheese images for any anomalies: sources that didn't get removed, or spurious sources
# that did. See the cook book for advice on what to do to get rid of the right sources. One may need
# to edit a copy of the emllist.fits file and rerun a few steps of cheese in order to produce better
# source exclusion regions, x-bkregBdet.fits and x-bkregBsky.fits and the source mask x-cheeseB.fits

ds9 *fovim?.fits *cheese?.fits &

ls -l *.* > post-cheese.list

# Clean up after running cheese but before the ultimate background region files are produced

mkdir cheese
mv cheese.log cheese
mv eboxlist* cheese
cp mos?????-cheese?.fits cheese
cp pn?????-cheese?.fits cheese

# IF
# You really need to reconstruct with a better source list then here is how to do it:
# If all "bad" sources are removed from emllist_fixd_filt.fits then you can set the selected
# DET_ML arbitrarily low and reconstruct the region files (-bkregtdet.fits and -bkregtbsky.fits) and

```



```

# cheese file. You will need the fovimN.fits and fovimNmask.fits files

# MOS1:

region eventset=${M1}-allevc.fits operationstyle=global srclisttab=emllist_fixd_filt.fits:SRCLIST
  expression="(ID_INST == 2)&&(DET_ML >= 10)" bkgregionset=${M1}-bkgregtdet.fits energyfraction=0.4
  radiusstyle=contour outunit=detxy verbosity=4
region eventset=${M1}-allevc.fits operationstyle=global srclisttab=emllist_fixd_filt.fits:SRCLIST
  expression="(ID_INST == 2)&&(DET_ML >= 10)" bkgregionset=${M1}-bkgregtsky.fits energyfraction=0.4
  radiusstyle=contour outunit=xy verbosity=4
makemask imagefile=${M1}-fovimt.fits maskfile=${M1}-fovimtmask.fits regionfile=${M1}-bkgregtsky.fits
cheesefile=${M1}-cheeset.fits

# MOS2:

region eventset=${M2}-allevc.fits operationstyle=global srclisttab=emllist_fixd_filt.fits:SRCLIST
  expression="(ID_INST == 3)&&(DET_ML >= 10)" bkgregionset=${M2}-bkgregtdet.fits energyfraction=0.4
  radiusstyle=contour outunit=detxy verbosity=4
region eventset=${M2}-allevc.fits operationstyle=global srclisttab=emllist_fixd_filt.fits:SRCLIST
  expression="(ID_INST == 3)&&(DET_ML >= 10)" bkgregionset=${M2}-bkgregtsky.fits energyfraction=0.4
  radiusstyle=contour outunit=xy verbosity=4
makemask imagefile=${M2}-fovimt.fits maskfile=${M2}-fovimtmask.fits regionfile=${M2}-bkgregtsky.fits
cheesefile=${M2}-cheeset.fits

# PN:

region eventset=${PN}-allevc.fits operationstyle=global srclisttab=emllist_fixd_filt.fits:SRCLIST
  expression="(ID_INST == 1)&&(DET_ML >= 10)" bkgregionset=${PN}-bkgregtdet.fits energyfraction=0.4
  radiusstyle=contour outunit=detxy verbosity=4
region eventset=${PN}-allevc.fits operationstyle=global srclisttab=emllist_fixd_filt.fits:SRCLIST
  expression="(ID_INST == 1)&&(DET_ML >= 10)" bkgregionset=${PN}-bkgregtsky.fits energyfraction=0.4
  radiusstyle=contour outunit=xy verbosity=4
makemask imagefile=${PN}-fovimt.fits maskfile=${PN}-fovimtmask.fits regionfile=${PN}-bkgregtsky.fits
cheesefile=${PN}-cheeset.fits

ds9 *fovim?.fits *cheese?.fits &

# ENDIF

# Second post cheese cleanup

mv emllist*.fits cheese
mv *-fovim?.fits cheese
ls -l *.fits > post_cheese_clean.list
mv *.log logs
mv *.list logs

# At this point in the main directory you should have only the files *-allevc, *-bkgregdet,
# *-bkgregtsky, and *-cheeseB for each instrument, as well as the atthk.fit and ccf.cif files.
# At this point I find it useful to save the analysis directory so if something goes wrong with
# mos/pn spectra and mos/pn back, I can quickly resurrect the directory with which I started!

cd ..
tar -cvf ffov_analysis.tar analysis
cd analysis

# Extracting and constructing the background spectrum - this is the most time consuming step.
# For this step you will need to have constructed your region selection files (ffov_mos1.txt, etc.)
# I've called the region 'ffov' because it is the full FOV but without the point sources.
# Use whatever descriptive name you like.

```

```
# This is the part of ESAS that produces the most files and can cause the most trouble if you are
# extracting more than one region. To reduce problems, I stick all of the region specific products
# into separate subdirectories. There is more to be done with these files, but I've set it up to be easy
# to follow rather than efficient.
```

```
# BREAK POINT 1 (see text in the following subsection of the appendix)
```

```
# MAKE SURE THAT THE REGION FILES ARE IN THE ANALYSIS DIRECTORY AT THIS POINT
```

```
# You need files containing expressions that describe the region that you want to extract. We assume
# that that region it is either the full FOV, or some large fraction of it. These files do not contain
# information about the sources to be excised (that information is already in the *-bkgregtdet.fits
# files).
```

```
# Model region files can be found [webpage to be determined].
```

```
# In the following steps the region files are assumed to be called
```

```
# regmos1-ffov.txt, regmos2-ffov.txt, and regpn-ffov.txt
```

```
# Set the elow and ehigh with environment variables to make doing multiple bands easier
```

```
setenv ELO 350
```

```
setenv EHI 1100
```

```
# MOS1 & MOS2
```

```
mosspectra eventfile=${M1}-allevc.fits keepinterfiles=yes withregion=yes regionfile=regmos1.txt
pattern=12 withsrcrem=yes maskdet=${M1}-bkgregtdet.fits masksky=${M1}-bkgregtsky.fits elow=${ELO}
ehigh=${EHI} ccds="${M1ON}" -V=7 >& mosspectra_1.log
```

```
mosspectra eventfile=${M2}-allevc.fits keepinterfiles=yes withregion=yes regionfile=regmos2.txt
pattern=12 withsrcrem=yes maskdet=${M2}-bkgregtdet.fits masksky=${M2}-bkgregtsky.fits elow=${ELO}
ehigh=${EHI} ccds="${M2ON}" -V=7 >& mosspectra_2.log
```

```
mosback inspecfile=${M1}-fov.t.pi elow=${ELO} ehigh=${EHI} ccds="${M1ON}" >& mosback_1.log
```

```
mosback inspecfile=${M2}-fov.t.pi elow=${ELO} ehigh=${EHI} ccds="${M2ON}" >& mosback_2.log
```

```
# Look at the diagnostics! There are many diagnostic files. I've gotten out of the habit.
```

```
# (Shame on me!) You can catch some problems early by looking at them.
```

```
qdp ${M1}-augindiv.qdp
```

```
/cps
```

```
exit
```

```
mv pgplot.ps ${M1}-augindiv.ps
```

```
qdp ${M1}-augspec.qdp
```

```
/cps
```

```
exit
```

```
mv pgplot.ps ${M1}-augspec.ps
```

```
qdp ${M1}-bkgaccum.qdp
```

```
/cps
```

```
exit
```

```
mv pgplot.ps ${M1}-bkgaccum.ps
```

```
qdp ${M1}-bkgindiv.qdp
```

```
/cps
```

```
exit
```

```
mv pgplot.ps ${M1}-bkgindiv.ps
```

```
qdp ${M1}-bkgspec.qdp
```

```
/cps
```

```
exit
```

```
mv pgplot.ps ${M1}-bkgspec.ps
```

```
qdp ${M1}-bridgefit.qdp
```

```
/cps
```

```

exit
mv pgplot.ps ${M1}-bridgefit.ps
qdp ${M1}-ratehard.qdp
/cps
exit
mv pgplot.ps ${M1}-ratehard.ps
qdp ${M1}-ratioindiv.qdp
/cps
exit
mv pgplot.ps ${M1}-ratioindiv.ps

```

```

qdp ${M2}-augindiv.qdp
/cps
exit
mv pgplot.ps ${M2}-augindiv.ps
qdp ${M2}-augspec.qdp
/cps
exit
mv pgplot.ps ${M2}-augspec.ps
qdp ${M2}-bkgaccum.qdp
/cps
exit
mv pgplot.ps ${M2}-bkgaccum.ps
qdp ${M2}-bkgindiv.qdp
/cps
exit
mv pgplot.ps ${M2}-bkgindiv.ps
qdp ${M2}-bkgspec.qdp
/cps
exit
mv pgplot.ps ${M2}-bkgspec.ps
qdp ${M2}-bridgefit.qdp
/cps
exit
mv pgplot.ps ${M2}-bridgefit.ps
qdp ${M2}-ratehard.qdp
/cps
exit
mv pgplot.ps ${M2}-ratehard.ps
qdp ${M2}-ratioindiv.qdp
/cps
exit
mv pgplot.ps ${M2}-ratioindiv.ps

```

# Now do some clean-up in order to keep your working directory small enough to find stuff!

```
ls -l *.* > post_mos.list
```

```

mkdir ffov_${M1}
mv mos1*-${ELO}-${EHI}* ffov_${M1}
mv mos1*.pi ffov_${M1}
mv mos1*.qdp ffov_${M1}
mv mos1*.psf ffov_${M1}
mv mos1*imt* ffov_${M1}
mv mos1*.arf ffov_${M1}
mv mos1*.rmf ffov_${M1}
mv mos1*imspdet* ffov_${M1}

```

```

mkdir ffov_${M2}
mv mos2*-${ELO}-${EHI}* ffov_${M2}

```

```

mv mos2*.pi ffov_${M2}
mv mos2*.qdp ffov_${M2}
mv mos2*.psf fov_${M2}
mv mos2*imt* ffov_${M2}
mv mos2*.arf ffov_${M2}
mv mos2*.rmf ffov_${M2}
mv mos2*imspdet* ffov_${M2}
ls -l *.* > post_mos_clean.list

# Now we want to repeat the QPB extraction for the pn. However we need to do it once for PATTERN==0
# (for the soft emission) and once for PATTERN<=4 (for the hard emission).

# PN PATTERN==0

pnspectra eventfile=${PN}-allevc.fits ootevtfile=${PN}-allevcoot.fits keepinterfiles=yes withregion=yes
  regionfile=regpn.txt pattern=0 withsrcrem=yes maskdet=${PN}-bkgregtdet.fits
  masksky=${PN}-bkgregtsky.fits elow=${ELO} ehigh=${EHI} quads="${PNON}" -V=7 >& pnspectra_0.log

pnback inspecfile=${PN}-fovt.pi inspeccoot=${PN}-fovtoot.pi elow=${ELO} ehigh=${EHI}
  quads="${PNON}" >& pnback_0.log

# Look at the diagnostics

qdp ${PN}-augratehrev.qdp
/cps
exit
mv pgplot.ps ${PN}-augratehrev.ps
qdp ${PN}-augspec.qdp
/cps
exit
mv pgplot.ps ${PN}-augspec.ps
qdp ${PN}-bridgefit.qdp
/cps
exit
mv pgplot.ps ${PN}-bridgefit.ps
qdp ${PN}-ratehard.qdp
/cps
exit
mv pgplot.ps ${PN}-ratehard.ps
qdp ${PN}-augraterev.qdp
/cps
exit
mv pgplot.ps ${PN}-augraterev.ps
qdp ${PN}-bkgspec.qdp
/cps
exit
mv pgplot.ps ${PN}-bkgspec.ps
qdp ${PN}-quadspec.qdp
/cps
exit
mv pgplot.ps ${PN}-quadspec.ps
qdp ${PN}-spec.qdp
/cps
exit
mv pgplot.ps ${PN}-spec.ps

# and do some clean-up

ls -l *.* > post_pn0.list

```

```

mkdir ffov_${PN}_0
mv pn*-${ELO}-${EHI}* ffov_${PN}_0
mv pn*.pi ffov_${PN}_0
mv pn*.qdp ffov_${PN}_0
mv pn*.ps ffov_${PN}_0
mv pn*.psf ffov_${PN}_0
mv pn*.imt* ffov_${PN}_0
mv pn*.arf ffov_${PN}_0
mv pn*.rmf ffov_${PN}_0
mv pn*imspdets* ffov_${PN}_0

# PN PATTERN<=4

pnspectra eventfile=${PN}-allevc.fits ootevtfile=${PN}-allevcoot.fits keepinterfiles=yes withregion=yes
  regionfile="regpn.txt" pattern=4 withsrcrem=yes maskdet=${PN}-bkgregtdet.fits withinimages=yes
  elow=${ELO} ehig=${EHI} quads="${PNON}" -V=7 >& pnspectra_4.log

pnback inspecfile=${PN}-fovt.pi inspecoot=${PN}-fovtoot.pi elow=${ELO} ehig=${EHI}
  quads="${PNON}" >& pnback_4.log

# It is always useful to look at the diagnostics, copy and past the commands from PN_0

# and do the clean-up

ls -l *.* > post_pn4.list

mkdir ffov_${PN}_4
mv pn*-${ELO}-${EHI}* ffov_${PN}_4
mv pn*.pi ffov_${PN}_4
mv pn*.qdp ffov_${PN}_4
mv pn*.psf ffov_${PN}_4
mv pn*.imt* ffov_${PN}_4
mv pn*.arf ffov_${PN}_4
mv pn*.rmf ffov_${PN}_4
mv pn*imspdets* ffov_${PN}_4
ls -l *.* > post_pn_clean.list

mv *.log logs
mv *.list logs

# We now need to get all of the spectral products in one place.
# IF you are working with something like a blank sky observation, these are the products that
# you will need for the spectral fitting to determine the residual soft proton flare emission
# (and maybe swcx).
# IF you have a large, bright, extended source in your FOV (such as a cluster of galaxies)
# you will need some of these products for the imaging reduction/analysis

# We create a directory just for the spectra. Then we visit each of the ffov_xxxx directories
# to copy the right stuff to the spectrum directory. Grouping the spectra is necessary for
# chi-squared fitting. Populating the header keywords is to save time and effort when we are
# actually fitting. While we are at it, we can use protonscale to determine the BACKSCAL parameter
# (in square arcminutes) which we will need to use in the fitting.

mkdir ffov_spectrum

cd ffov_${M1}

# An easy way to get the BACKSCAL header keyword which you will need for the spectral fitting

protonscale mode=1 maskfile=${M1}-fovimspdets.fits specfile=${M1}-fovt.pi

```

```

# Grouping the spectrum and adding the ANCRFILE, BACKFILE, and RESPFILE header keywords (important)

grppha ${M1}-fovt.pi mos1-grp.pi 'chkey BACKFILE mos1-bkg.pi & chkey RESPFILE mos1.rmf &
  chkey ANCRFILE mos1.arf & group min 50 & exit'
mv mos1-grp.pi ../ffov_spectrum/mos1-grp.pi
cp ${M1}-bkg.pi ../ffov_spectrum/mos1-bkg.pi
cp ${M1}.rmf ../ffov_spectrum/mos1.rmf
cp ${M1}.arf ../ffov_spectrum/mos1.arf
cd ..

cd ffov_${M2}

protonscale mode=1 maskfile=${M2}-fovimspdet.fits specfile=${M2}-fovt.pi

grppha ${M2}-fovt.pi mos2-grp.pi 'chkey BACKFILE mos2-bkg.pi & chkey RESPFILE mos2.rmf &
  chkey ANCRFILE mos2.arf & group min 50 & exit'
mv mos2-grp.pi ../ffov_spectrum/mos2-grp.pi
cp ${M2}-bkg.pi ../ffov_spectrum/mos2-bkg.pi
cp ${M2}.rmf ../ffov_spectrum/mos2.rmf
cp ${M2}.arf ../ffov_spectrum/mos2.arf
cd ..

cd ffov_${PN}_0

protonscale mode=1 maskfile=${PN}-fovimspdet.fits specfile=${PN}-fovt.pi

grppha ${PN}-fovtootsub.pi pn0-grp.pi 'chkey BACKFILE pn0-bkg.pi & chkey RESPFILE pn0.rmf &
  chkey ANCRFILE pn0.arf & group min 50 & exit'
mv pn0-grp.pi ../ffov_spectrum/pn0-grp.pi
cp ${PN}-bkg.pi ../ffov_spectrum/pn0-bkg.pi
cp ${PN}.rmf ../ffov_spectrum/pn0.rmf
cp ${PN}.arf ../ffov_spectrum/pn0.arf
cd ..

cd ffov_${PN}_4

protonscale mode=1 maskfile=${PN}-fovimspdet.fits specfile=${PN}-fovt.pi

grppha ${PN}-fovtootsub.pi pn4-grp.pi 'chkey BACKFILE pn4-bkg.pi & chkey RESPFILE pn4.rmf &
  chkey ANCRFILE pn4.arf & group min 50 & exit'
mv pn4-grp.pi ../ffov_spectrum/pn4-grp.pi
cp ${PN}-bkg.pi ../ffov_spectrum/pn4-bkg.pi
cp ${PN}.rmf ../ffov_spectrum/pn4.rmf
cp ${PN}.arf ../ffov_spectrum/pn4.arf
cd ..

# BREAK POINT 2 (see text in the following subsection of the appendix)

# IF you have a large, bright, extended source in your FOV (such as a cluster of galaxies)
# you now need to refer to the recipe in the following section. If you have a "blank-sky-like"
# observation, proceed to spectral fitting.

# Spectral fitting: this is up to you!

# We now assume that you have fit the soft proton flare spectrum and the swcx spectrum.
# We are now ready to create all of the components of our image from the individual detectors.
# As with the spectra, we create an imaging directory, and then visit all of the ffov_xxxx
# directories to copy the right stuff to the image directory. There are also a number of other
# pieces to be created for each instrument.

```

```

# Building the background images:

# At this point we have the raw images, the *-fovimsky-elo-ehi.fits files, and the QPB images, the
# *-bkgimdet-elo-ehi.fits files. However the QPB images must be rotated from detector coordinates
# into sky coordinates. In this section we will do that rotation, as well as create and rotate the
# SPF and SWCX images, one detector at a time.
# As with the spectral components, we sequester all the image components into one directory

mkdir ffov_image

# copy the cheese masks if you will be using them

cp *cheese?.fits ffov_image

# BREAK POINT 3 (see text in the following subsection of the appendix)

# MOS1

cd ffov_${M1}

# Rotate the QPB image from detector to sky coordinates

rotdet2sky intemplate=${M1}-fovimsky-${ELO}-${EHI}.fits inimage=${M1}-bkgimdet-${ELO}-${EHI}.fits
outimage=${M1}-bkgimsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Create and rotate the soft proton flare background image (if you need to). Note that the
# normalization (pnorm) is for the entire FOV, and will be different for each instrument

proton imagefile=${M1}-fovimdets-${ELO}-${EHI}.fits specfile=${M1}-fovft.pi ccds="${M1ON}" elow=${ELO}
ehigh=${EHI} speccontrol=1 pindex=1.26191 pnorm=2.208e-2 > proton_mos1.log
rotdet2sky intemplate=${M1}-fovimsky-${ELO}-${EHI}.fits inimage=${M1}-protimdets-${ELO}-${EHI}.fits
outimage=${M1}-protimsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Create and rotate the swcx background image (if you need to). Note that the normalization here is
# per square arcminute, and is the same for all instruments

swcx imagefile=${M1}-fovimdets-${ELO}-${EHI}.fits specfile=${M1}-fovft.pi ccds="${M1ON}" elow=${ELO}
ehigh=${EHI} rmffile=${M1}.rmf arffile=${M1}.arf lines="OVII OVIII"
gnorms="1.69938e-7 1.75262e-8" > swcx_mos1.log
rotdet2sky intemplate=${M1}-fovimsky-${ELO}-${EHI}.fits inimage=${M1}-swcximdets-${ELO}-${EHI}.fits
outimage=${M1}-swcximsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Copy all the useful stuff to the image directory

cp ${M1}-fovimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M1}-bkgimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M1}-maskimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M1}-expimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M1}-protimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M1}-swcximsky-${ELO}-${EHI}.fits ../ffov_image
mv *.log ../logs
cd ..

# MOS2

cd ffov_${M2}

# Rotate the QPB image made by mosback into sky coordinates

rotdet2sky intemplate=${M2}-fovimsky-${ELO}-${EHI}.fits inimage=${M2}-bkgimdet-${ELO}-${EHI}.fits

```

```

outimage=${M2}-bkgimsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Create and rotate the soft proton flare background image (if you need to).

proton imagefile=${M2}-fovmdet-${ELO}-${EHI}.fits specfile=${M2}-fovft.pi ccds="${M2ON}" elow=${ELO}
ehigh=${EHI} speccontrol=1 pindex=0.1 pnorm=2.546e-2 > proton_mos2.log
rotdet2sky intemplate=${M2}-fovimsky-${ELO}-${EHI}.fits inimage=${M2}-protimdet-${ELO}-${EHI}.fits
outimage=${M2}-protimsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Create and rotate the swcx background image (if you need to).

swcx imagefile=${M2}-fovmdet-${ELO}-${EHI}.fits specfile=${M2}-fovft.pi ccds="${M2ON}" elow=${ELO}
ehigh=${EHI} rmffile=${M2}.rmf arffile=${M2}.arf lines="OVII OVIII"
gnorms="1.69938e-7 1.75262e-8" > swcx_mos2.log
rotdet2sky intemplate=${M2}-fovimsky-${ELO}-${EHI}.fits inimage=${M2}-swcximdet-${ELO}-${EHI}.fits
outimage=${M2}-swcximsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Copy all the useful stuff to the image directory

cp ${M2}-fovimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M2}-bkgimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M2}-maskimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M2}-expimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M2}-protimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${M2}-swcximsky-${ELO}-${EHI}.fits ../ffov_image
mv *.log ../logs
cd ..

# PN_0 (assuming that you want the soft image. Use PN_4 if you want the hard image.)

cd ffov_${PN}_0

# Rotate the QPB image made by mosback into sky coordinates

rotdet2sky intemplate=${PN}-fovimsky-${ELO}-${EHI}.fits inimage=${PN}-bkgimdet-${ELO}-${EHI}.fits
outimage=${PN}-bkgimsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Create and rotate the soft proton flare background image (if you need to).

proton imagefile=${PN}-fovmdet-${ELO}-${EHI}.fits specfile=${PN}-fovft.pi ccds="${PNON}" elow=${ELO}
ehigh=${EHI} speccontrol=1 pindex=0.1 pnorm=5.479e-2 > proton_pn.log
rotdet2sky intemplate=${PN}-fovimsky-${ELO}-${EHI}.fits inimage=${PN}-protimdet-${ELO}-${EHI}.fits
outimage=${PN}-protimsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Create and rotate the swcx background image (if you need to).

swcx imagefile=${PN}-fovmdet-${ELO}-${EHI}.fits specfile=${PN}-fovft.pi ccds="${PNON}" elow=${ELO}
ehigh=${EHI} rmffile=${PN}.rmf arffile=${PN}.arf lines="OVII OVIII"
gnorms="1.69938e-7 1.75262e-8" > swcx_mos2.log
rotdet2sky intemplate=${PN}-fovimsky-${ELO}-${EHI}.fits inimage=${PN}-swcximdet-${ELO}-${EHI}.fits
outimage=${PN}-swcximsky-${ELO}-${EHI}.fits withdetxy=false withskyxy=false

# Copy all the useful stuff to the image directory

cp ${PN}-fovimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${PN}-fovimootsky-${ELO}-${EHI}.fits ../ffov_image
cp ${PN}-bkgimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${PN}-maskimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${PN}-expimsky-${ELO}-${EHI}.fits ../ffov_image
cp ${PN}-protimsky-${ELO}-${EHI}.fits ../ffov_image

```



```

cp ${PN}-swcximsky-${ELO}-${EHI}.fits ../ffov_image
mv *.log ../logs
cd ..

# Now, combine all of the image components

cd ffov_image
ls -l *.* > pre_comb.list

combimage prefixlist='1S001 2S006 S005' withpartbkg=true withspbkg=true withswcxbkg=true
  withcheese=true cheesetype=t elowlist=${ELO} ehighlist=${EHI} alpha=1.7 >& combimage.log

ls -l *.* > post_comb.list
mv *.list logs

#Adaptive smoothing or binning? You can do either or both

# Just adaptive smoothing...

binadapt prefix=comb elow=${ELO} ehigh=${EHI} withpartbkg=true withswcxbkg=false withspbkg=true
  withmask=false withbinning=false binfactor=1 withsmoothing=true smoothcounts=50

# ...just binning...

binadapt prefix=comb elow=${ELO} ehigh=${EHI} withpartbkg=true withswcxbkg=true withspbkg=true
  withmask=false withbinning=true binfactor=4 withsmoothing=false smoothcounts=50

# ...or both!

binadapt prefix=comb elow=${ELO} ehigh=${EHI} withpartbkg=true withswcxbkg=true withspbkg=true
  withmask=false withbinning=true binfactor=4 withsmoothing=true smoothcounts=50

ls -l *.* > post_adapt.list
mv *.list logs

# I will note that this leaves the main directory clean to extract another region if desired.
# You just need to name it something other than "ffov".
# Note the ESAS does not require this type of directory structure, but this directory structure
# has developed with the new ESAS as a way of keeping things neat, and keeping things from
# becoming too confusing.

```

## B.2. A Complex Single Observation, Spectroscopy and Imaging

This section contains a partial recipe for a single observation that has a bright extended source in the FOV. It is assumed that you have already applied the recipe from the previous section to extract the raw image of the full FOV, and to extract the spectrum of the entire FOV. Because that spectrum contains the emission from the bright source, the contribution due to the residual soft proton contamination or SWCX may be entirely overwhelmed by the source spectrum. Thus, we need to extract spectra for a region that minimizes the contribution from the bright source.

For the full FOV, one should have followed the recipe in the previous section up to “BREAK POINT 2”. In the `analysis` directory, you should have a number of subdirectories: `ffov_mos1S001`, `ffov_mos2S002`, `ffov_pnS003_0`, and `ffov_pnS003_4`.

It is assumed that you have constructed region files, similar to the `reg*.txt` files of the previous recipe that exclude the bright source, perhaps named `reg*-annu.txt`. Using these region files, re-execute from “BREAK POINT 1” to “BREAK POINT 2”, placing the output products into the subdirectories `annu_mos1S001`, `annu_mos2S002`, `annu_pnS003_0`, `annu_pnS003_4`, and `annu_spectra`. Then fit the spectra for all the Galactic components, the residual soft proton flare emission, and the SWCX emission. We can now use the fit parameters to construct the residual soft proton flare images and the SWCX images using the recipe in the previous section with the following exception:

At “BREAK POINT 3” we need to convert the normalization of the residual soft proton emission from the region that was extracted for the spectral fit (`annu`) to the region that was extracted for the full FOV (`ffov`). This is done with `sppartial`, where “xxx” is the normalization from the fit to the soft proton emission. (It does not make a difference whether it was a power law or a broken power law.)

```
sppartial fullimage=ffov_${M1}/${M1}-fovimpdet.fits fullspec=ffov_${M1}/${M1}-fovvt.pi
regimage=annu_${M1}/${M1}-fovimpdet.fits regspec=annu_${M1}/${M1}-fovvt.pi regnorm=xxx_mos1
```

```
sppartial fullimage=ffov_${M2}/${M2}-fovimpdet.fits fullspec=ffov_${M2}/${M2}-fovvt.pi
regimage=annu_${M2}/${M2}-fovimpdet.fits regspec=annu_${M2}/${M2}-fovvt.pi regnorm=xxx_mos2
```

```
sppartial fullimage=ffov_${PN}_0/${PN}-fovimpdet.fits fullspec=ffov_${PN}_0/${PN}-fovvt.pi
regimage=annu_${PN}_0/${PN}-fovimpdet.fits regspec=annu_${PN}_0/${PN}-fovvt.pi regnorm=xxx_pn
```

This routine will return the normalization that is required for the spectra in “ffov”. We can now return to the “ffov” directories and run the recipe after “BREAK POINT 3”. Note that no changes are required for creating the SWCX images.

### B.3. Multiple Overlapping Obsids, Imaging

For this example, we assume that each obsid to be included in our final image has been reduced, as in the previous example, up to but not including *combimage*. (Not that there is a problem if you have run *combimage*, it just isn't used for the mosaics.) We assume that in your main analysis directory, that you have a subdirectory for each obsid. The "mosaic.list" file contains a listing of all the directories containing the data.

For the following example, the file mosaic.list contains the following lines:

```
/Users/kuntz/Data/xmm_m101/0104260101/analysis/base_image/mos1S001
/Users/kuntz/Data/xmm_m101/0104260101/analysis/base_image/mos2S006
/Users/kuntz/Data/xmm_m101/0104260101/analysis/base_image/pnS005
/Users/kuntz/Data/xmm_m101/0164560701/analysis/base_image/mos1S001
/Users/kuntz/Data/xmm_m101/0164560701/analysis/base_image/mos2S002
/Users/kuntz/Data/xmm_m101/0164560701/analysis/base_image/pnS003
/Users/kuntz/Data/xmm_m101/0212480201/analysis/base_image/mos1S001
/Users/kuntz/Data/xmm_m101/0212480201/analysis/base_image/mos2S002
/Users/kuntz/Data/xmm_m101/0212480201/analysis/base_image/pnS003
```

Note that this is not a directory listing, per se, but a listing of directories and the observing segments within them that should be combined into a mosaic. The contents of an individual 0104260101/analysis/base\_image directory is (for example):

```
mos1S001-bkgimsky-350-1100.fits
mos1S001-cheeset.fits
mos1S001-expimsky-350-1100.fits
mos1S001-fovimsky-350-1100.fits
mos1S001-maskimsky-350-1100.fits
mos1S001-protimsky-350-1100.fits
mos1S001-swcximsky-350-1100.fits
mos2S006-bkgimsky-350-1100.fits
mos2S006-cheeset.fits
mos2S006-expimsky-350-1100.fits
mos2S006-fovimsky-350-1100.fits
mos2S006-maskimsky-350-1100.fits
mos2S006-protimsky-350-1100.fits
mos2S006-swcximsky-350-1100.fits
pnS005-bkgimsky-350-1100.fits
pnS005-cheeset.fits
pnS005-expimsky-350-1100.fits
pnS005-fovimootsky-350-1100.fits
pnS005-fovimsky-350-1100.fits
pnS005-maskimsky-350-1100.fits
pnS005-protimsky-350-1100.fits
pnS005-swcximsky-350-1100.fits
```

Now, it is quite possible to have other things within this directory, but these are the things that will get made into a mosaic. In particular, notice you could have the results of other observation segments in this directory, but they would not be included in the mosaic unless they are particularly listed in the "mosaic.list" file. Finally, here is a script for putting together a mosaic of M101.

```
setenv RAPNT 210.8
setenv DECPNT 54.32
```

```
#building the individual components of the mosaic
```

```
mosaicmerge dirfile=mosaic.list type=1 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'
```

```
mosaicmerge dirfile=mosaic.list type=2 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'

mosaicmerge dirfile=mosaic.list type=3 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'

mosaicmerge dirfile=mosaic.list type=4 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'

mosaicmerge dirfile=mosaic.list type=5 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'

mosaicmerge dirfile=mosaic.list type=6 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'

mosaicmerge dirfile=mosaic.list type=7 coordsys=2 crval1=$RAPNT crval2=$DECPNT
  pixelsize=4.163999e-2 alpha=1.7 elow=350 ehigh=1100 withcheese=yes cheesemasktype='t'

#putting together the image with no smoothing or binning

binadaptmerge elowlist=350 ehighlist=1100 withpartbkg=yes withspbkg=true withswcxbkg=true
  withmask=true maskfile=mosaic-cheeset.fits withbinning=false withsmoothing=false fill=0
```

#### B.4. Cheese with Multiple Overlapping Observations

This example shows how one can optimize point source detection for a set of observations that are almost entirely overlapping, such as three observations of the same target. It may not matter, but it is a good idea to run this from a window whose \$SAS\_CCF points to the CCF for the first observation.

```
# merge the attitude files for the three observations

merge set1=0820840301/analysis/atthk.fits set2=0820840401/analysis/atthk.fits outset=atthk_temp.fits
  mergedifferentobs="yes"
merge set1=atthk_temp.fits set2=0820840501/analysis/atthk.fits outset=atthk.fits
  mergedifferentobs="yes"
rm atthk_temp.fits

# Merge all of the mos1 event files

merge mergedifferentobs="yes" set1=0820840301/analysis/mos1S001-allevc.fits
  set2=0820840401/analysis/mos1S001-allevc.fits outset=mos1_t1.fits
merge set1=mos1_t1.fits set2=0820840501/analysis/mos1S001-allevc.fits outset=mos1_event.fits
  mergedifferentobs="yes"
rm mos1_t?.fits

# merge all of the mos2 event files

merge mergedifferentobs="yes" set1=0820840301/analysis/mos2S002-allevc.fits
  set2=0820840401/analysis/mos2S002-allevc.fits outset=mos2_t1.fits mergedifferentobs="yes"
merge set1=mos2_t1.fits set2=0820840501/analysis/mos2U002-allevc.fits outset=mos2_event.fits
  mergedifferentobs="yes"
rm mos2_t?.fits

# merge all of the pn event files

merge set1=0820840301/analysis/pnS003-allevc.fits set2=0820840401/analysis/pnS003-allevc.fits
  outset=pn_t1.fits mergedifferentobs="yes"
merge set1=pn_t1.fits set2=0820840501/analysis/pnS003-allevc.fits outset=pn_event.fits
  mergedifferentobs="yes"
rm pn_t1.fits

# merge all of the pn oot event files

merge mergedifferentobs="yes" set1=0820840301/analysis/pnS003-allevcoot.fits
  set2=0820840401/analysis/pnS003-allevcoot.fits outset=pn_t1.fits
merge set1=pn_t1.fits set2=0820840501/analysis/pnS003-allevcoot.fits outset=pn_oot_event.fits
  mergedifferentobs="yes"
rm pn_t1.fits

# now run cheese

cheese mos1file=mos1_event.fits mos2file=mos2_event.fits pnfile=pn_event.fits
  pnootfile=pn_oot_event.fits elowlist=350 ehighlist=1100 scale=0.5 mlmin=10 ratetotal=0.2
  dist=50. keepinterfiles=no -V 7 >& cheese.log

# apply the results to the individual observations
# (A demonstration just for the first obsid)

cp emllist_fixd_filt.fits 0820840301/analysis/emllist_joint.fits
```

```
cd 0820840301/analysis
```

```
# building the region files and mask for MOS1
```

```
region eventset=${M1}-allevc.fits operationstyle=global srclisttab=emllist_joint.fits:SRCLIST  
expression="(ID_INST == 2)&&(DET_ML >= 10)" bkgregionset=${M1}-bkgregtdet.fits  
energyfraction=0.4 radiusstyle=contour outunit=detxy verbosity=4  
region eventset=${M1}-allevc.fits operationstyle=global srclisttab=emllist_joint.fits:SRCLIST  
expression="(ID_INST == 2)&&(DET_ML >= 10)" bkgregionset=${M1}-bkgregtsky.fits  
energyfraction=0.4 radiusstyle=contour outunit=xy verbosity=4  
makemask imagefile=${M1}-fovimt.fits maskfile=${M1}-fovimtmask.fits regionfile=${M1}-bkgregtsky.fits  
cheesefile=${M1}-cheeset.fits
```

```
# building the region files and mask for MOS2
```

```
region eventset=${M2}-allevc.fits operationstyle=global srclisttab=emllist_joint.fits:SRCLIST  
expression="(ID_INST == 3)&&(DET_ML >= 10)" bkgregionset=${M2}-bkgregtdet.fits  
energyfraction=0.4 radiusstyle=contour outunit=detxy verbosity=4  
region eventset=${M2}-allevc.fits operationstyle=global srclisttab=emllist_joint.fits:SRCLIST  
expression="(ID_INST == 3)&&(DET_ML >= 10)" bkgregionset=${M2}-bkgregtsky.fits  
energyfraction=0.4 radiusstyle=contour outunit=xy verbosity=4  
makemask imagefile=${M2}-fovimt.fits maskfile=${M2}-fovimtmask.fits regionfile=${M2}-bkgregtsky.fits  
cheesefile=${M2}-cheeset.fits
```

```
# building the region files and mask for PN
```

```
region eventset=${PN}-allevc.fits operationstyle=global srclisttab=emllist_joint.fits:SRCLIST  
expression="(ID_INST == 1)&&(DET_ML >= 10)" bkgregionset=${PN}-bkgregtdet.fits  
energyfraction=0.4 radiusstyle=contour outunit=detxy verbosity=4  
region eventset=${PN}-allevc.fits operationstyle=global srclisttab=emllist_joint.fits:SRCLIST  
expression="(ID_INST == 1)&&(DET_ML >= 10)" bkgregionset=${PN}$-bkgregtsky.fits  
energyfraction=0.4 radiusstyle=contour outunit=xy verbosity=4  
makemask imagefile=${PN}-fovimt.fits maskfile=pnS003-fovimtmask.fits  
regionfile=${PN}-bkgregtsky.fits cheesefile=${PN}-cheeset.fits
```