cheese

January 27, 2025

Abstract

This task creates "Swiss cheese" masks after running source detection on full-field images. This is a f95 rewrite of the pre-SAS-21 *esas* perl subtask *cheese*. This task runs *edetect_chain*, *region*, and *makemask* in a self-consistent manner for up to two bands and all three EPIC instruments. Some of the output files from *cheese* can be further edited or modified by the user to allow very precise control of the source removal; doing so requires the reapplication of *region* and *makemask*.

1 Instruments/Modes

Instrument	Mode
EPIC	Imaging

2 Use

pipeline processing	no
interactive analysis	yes

3 Description

Cheese runs source detection on full-field images and creates "Swiss cheese" 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. Several issues in particular should be noted by the user.

First, the **dist** parameter set the minimum accepted distance between sources. If two sources are closer together than this limit, they will be combined. If this is problematic, decrease the value of **dist**.

Second, sources are detected on an instrument-by-instrument basis, and a detection in one instrument does not guarantee removal from another instrument. For example, if a source were detected in MOS1 and MOS2, but fell in a chip gap on the pn, even if the source is very bright and obviously impacts the surrounding pixels, it will not be removed. This is the default behavior. The output *emllist.fits* can be modified by the user, and *region* and *makemask* re-run in order to fix this issue.

Third, setting a single minimum maximum likelihood **mlmin** value may produce a mask that misses "obvious" sources, or may exclude regions where there is no "obvious" source. Since the **mlmin** value is determined over all the instruments, there may be complex reasons for a mismatch between the value and the user's expectation of the value. Careful examination (and perhaps by-hand modification of the *emllist.fits* file) is required to assure that the source identification/removal is appropriate for your science.

4 Parameters

This section documents the parameters recognized by this task (if any).

Parameter	Mand	Type	Default	Constraints
mos1file	no	string		
Cleaned (filtered) EMOS1 event list.				
mos2file	no	string		
Cleaned (filtered) EMOS2 event list.				
pnfile	no	string		
Cleaned (filtered) EPN event list.				
pnootfile	no	string		
Cleaned EPN Out Of Time event list (for edetect_chain).				
scale	no	real	0.5	$0 <= scale$
Energy fraction, which sets the exclusion radius of point sources.				
ratetotal	no	real	1.0	$0 <= ratetotal <= 1000$
Flux threshold full band (in units of $1.0E-14 cgs$ for the exclusion of point sources.				
ratesoft	no	real	1.0	$0 <= ratesoft <= 1000$
Flux threshold soft band (in units of $1.0E-14 cgs$ for the exclusion of point sources.				
ratehard	no	real	1.0	$0 <= ratehard <= 1000$
Flux threshold hard band (in units of $1.0E-14 cgs$ for the exclusion of point sources.				
dist	no	real	0.0	$0 <= dist <= 50$
Minimum allowed separation in arc seconds between sources in arcsec. If two sources are detected that are closer together than this limit, the sources will be combined. Reducing this value allows faint sources close to bright sources to be masked separately.				
mlmin	no	real	15.0	$0 <= mlmin <= 100$
Minimum Max Likelihood detection value.				
elowlist	no	int	350	$0 <= elowlist <= 11999$
The low energy for the band in eV.				
ehighlist	no	int	1100	$2 <= ehighlist <= 12000$

The high energy for the band in eV.

keepinterfiles	no	boolean	true	T/F

Keep (do not delete) intermediary files produced by cheese?

5 Errors

This section documents warnings and errors generated by this task (if any). Note that warnings and errors can also be generated in the SAS infrastructure libraries, in which case they would not be documented here. Refer to the index of all errors and warnings available in the HTML version of the SAS documentation.

NoInsts *(error)*
At least one of mos1/mos2/pn file must be entered

eLowsGT2 *(error)*
Number of low energy bands can't exceed 2

eHighGT2 *(error)*
Number of high energy bands can't exceed 2

eLowsNEeHighs *(error)*
Number of low and high energy bands must be equal

notM1M2PN *(error)*
Instrument is not M1, M2, or PN

6 Input Files

1. Filtered EMOS1 event list.
2. Filtered EMOS2 event list.
3. Filtered EPN event list.
4. Filtered PN OOT event list.

7 Output Files

Note: all output file names are derived from either the INST+EXPID+type, or are the standard output files of the constituent tasks that create them within cheese (e.g. edetect_chain). Files are only created for those instruments selected so not all files below will be created for every run of cheese.

1. MOS1 FOV image in sky coords.
2. MOS2 FOV image in sky coords.
3. PN FOV image in sky coords.
4. Max likelihood list (emllist.fits, from edetect_chain)
5. MOS1 Background region files in sky coords.
6. MOS2 Background region files in sky coords.
7. PN Background region files in sky coords.
8. MOS1 FOV masks.
9. MOS2 FOV masks.
10. PN FOV masks.
11. MOS1 FOV cheese masks.
12. MOS2 FOV cheese masks.
13. PN FOV cheese masks.
14. MOS1 FOV sensitivity maps.
15. MOS2 FOV sensitivity maps.
16. PN FOV sensitivity maps.
17. `atthk.fits` – SAS attitude file.
18. `eboxlist_l.fits` – The output from the first pass of *eboxdetect*.
19. `eboxlist_m.fits` – The output from the second pass of *eboxdetect*.
20. `emllist.fits` – The output from *emldetect*.

8 Algorithm

1. Read in parameters.
2. Check params for sanity.
3. Loop through each instrument (if exists)
    1. Create FOV image in total band.
    2. If two bands, also create FOV image in soft and hard bands
    3. Populate lists and strings for edetect_chain call
    4. Run atthkgen (if necessary) to make atthk.fits file
    5. Run edetect_chain to create emllist.fits, exp and sens maps.
    6. Run emask to make mask images.
    7. Run region to create background region files.
    8. Run make_mask to create final cheese maps.
    9. Clean up intermediary files if desired.

9 Comments

Note that cheese takes tens of minutes to run because of the call to edetect_chain (on a pre-M1 MacBook Pro).

The code for this task originally appeared as the perl *esas* subtask *cheese* 2009-2021. It was modularized as a single task in f95 for SAS-21. The *esas* task was removed in SAS-21.

References