omvariability

November 4, 2014

Abstract

Program for computing variability parameters of sources that have multiple detections in one or more observations.

1 Instruments/Modes

Instrument	Mode
OM	FAST
OM	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This program computes variability statistics (**chi-squared** and the maximum deviation from the median corrected count-rate) and produces light-curves from an input list of **OM** product source-list (**SWS**) files. The source-lists can come from more than one observation, and for each set of source-lists in a particular observation **omvariability** will compute short-term (hours to days depending on how many exposures) variability parameters for sources that have two or more detections in a given OM filter. In the case where the input list of source-lists come from more than observation, **omvariability** will also compute long-term (years) variability parameters for unique sources that have been detected in two or more observations.

The given output file, which has to be an **OM** observation source-list file, is then modified by having variability columns added for each filter.

If desired,a plot file of the "variable" sources can be produced.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
directorylist	no	string	” ”	

List of directories containing OM product files. All the OM product source-list files in these directories will be used as input/

| **srclistsets** | no | string | ” ” | |

List of OM product source-list files. If none are specified the files will be either taken from the current directory or from the given list of input directories

| **obsset** | no | string | “” | |

Name of the observation source-list file to which the variability columns should be added. If no file is given, and if either you **do not** give a list of directories or do give a list of source-list files, the program will try to find the observation source-list file. If it does not find it a temporary one will be created by executing **omsrclistcomb** internally and then deleting it.

| **plotfile** | no | string | omvariability.ps | |

Name of the plot file for the source-variability graphs

| **minnumcounterparts** | no | integer | 3 | 2– |

Minimum number of source counterparts for variability statistics to be computed

| **minstd** | no | real | 2 | 0– |

Minimum deviation of any source corrected count-rate from the median corrected-count rate for a variability diagram to be produced.

| **nxsub** | no | integer | 2 | 1-10 |

Number of graphs in the horizontal direction

| **nysub** | no | integer | 2 | 1-10 |

Number of graphs in the vertical direction

| **eflag** | no | integer | -1 | -1-1 |

Plot variability data for a given unique source if eflag is -1 or all the extension flags of the counterparts are equal to eflag

cflag	no	integer	-1	-1-0

Plot variability data for a given unique source if cflag is -1 or all the confusion flags of the counterparts are equal to cflag

qflag	no	integer	-1	-1-0

Plot variability data for a given unique source if qflag is -1 or all the quality flags of the counterparts are equal to qflag

5   Usage

5.1   Command-line list of parameters

1. omvariability

   - Use all the OM product source-list files in the current directory as input.
   - Add variability parameters to the OM observation source-list file in the current directory.
   - Store the source-variability plots in omvariability.ps.

2. omvariability plotfile=plot.ps

   - Use all the OM product source-list files in the current directory as input.
   - Add variability parameters to the OM observation source-list file in the current directory.
   - Store the source-variability plots in plot.ps.

3. omvariability minnumcounterparts=5 minstd=3

   - Use all the OM product source-list files in the current directory as input.
   - Add variability parameters to the OM observation source-list file in the current directory.
   - Store the source-variability plots in omvariability.ps.
   - Only compute variability parameters when there are a minimum of 5 counterparts to a unique source and the computed standard-deviation of the corrected count-rate is at least 3..

4. omvariability directorylist="directory1 directory2 directory3"

   - Use all the OM product source-list files in the directories directory1, directory2 and directory3 as input.

- Store the source-variability plots in omvariability.ps.
- A temporary observation source-list file is created by running **omsrclistcomb** but deleted.

5. omvariability directorylist="directory1 directory2 directory3" obsset=observation.fits

- Use all the OM product source-list files in the directories directory1, directory2 and directory3 as input.
- Create an observation source-list file called observation.fits and add the variability columns to it.
- Store the source-variability plots in omvariability.ps.

6 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.

image file is not 2D *(fatal)*

different binning in x and y axes *(fatal)*

Could not allocate memory for: *name (fatal)*
    Memory allocation for the named allocatable array failed

Failed to release memory for: *name (warning)*

    corrective action: Memory deallocation for the named allocatable array failed

7 Input Files

1. Product exposure source-list files (output from OMMAT)

2. Product observation source-list files (output from OMSRCLISTCOMB)

8 Output Files

1. FITS product observation source-list file

2. Postscript plot file containing plots of the source magnitudes versus time.

9 Algorithm

omvariability is a **c++** program and works as follows:

- For each **SWS** file given in the list for the parameter **srclistsets**-
  - Read in the **Right-Ascension**, **Declination**, **corrected count-rate** and **corrected count-rate error**, **instrumental magnitude** and **instrumental magnitude error**, **extension flag**, **confusion flag** and **quality flag** and **Source ID**. Note that the last quantity identifies which unique source in an observation source-list file the source is a counterpart of. A **map** is used in which the **key** is a source **ID** and the **value** another map having its key as the **OM filter** and its value as a **c++ vector** that stores the parameters of the source from each **SWS** file.

- Iterate through the **source IDs** stored in the source map and for each one:
  - Iterate through the list of **OM** filters for that source and
    * If the list contains at least the number of sources specfied by the **parameter minnumsources**, compute variability parameters and store the parameters in another map **MAP 1** that the **OM filter** as its key and the variability parameters as its value (stored in a structure).
  - If the user has specified that a variability plot should be produced, iterate through the computed variability parameters for each **filter** for the source **ID** and if the minimum deviation of the corrected count-rate of any source from the median value is at least equal to the value set by the input **parameter minnstd**, plot the source details.

- Iterate through the filters in **MAP 1** and add the variability columns to the output file specified by the parameter **outset**.

9.1 Variability statistics

The following two variability parameters are computed

- The **reduced chi-squared** statistic-
  $\mathbf{chi} = (\sum_{k=1}^{n} w_k(c_k - mean))/n - 1$,
  where mean is the mean corrected count-rate$= (\sum_{k=1}^{n} w_k c_k)/\sum_{k=1}^{n} w_k$
  and
  $w_k = 1/\Delta c^2$

- **NSTD** The maximum deviation of any source away from the median corrected count-rate of all the sources.

10 Testing

The aim of the test is to demonstrate that **omvariability** correctly computes variability parameters for each common source in the list of input **SWS** source-list files and correctly adds the data in the correct positions in the given output file.

The test directory for omvariability contains a shell program that performs the following:

- Runs the program **createfiles** that produces-

  - A master source list file (**masterfile.fits** that contains celestial coordinates, corrected count-rates and errors, source extension, confusion and quality flags. This file has a similar structure to that of an observation source-list file produced by the **SAS** task **omsrclistcomb**.

  - Four source-list files each containing a list of sources taken randomly from the master source-list file. Each source has a random small error added to the corrected count-rate. These files would be the input source-lists to **omsrclistcomb** that would produce the output file **masterfile.fits**.

  - A reference source-list file (**checkfile.fits**, first copied from **masterfile.fits**, and then having variability columns **V_CHI2** and **V_NSTD** added. These columns are computed using the **corrected count-rates** for each source in a **SWS** file that corresponds to a given source in **masterfile.fits**.

- Runs **omvariability** with the parameter **srclistsets** set to the list of **SWS** files and the parameter **outset** set to the file **masterfile.fits**

- Runs the program **checkoutput** that compares the celestial-coordinates and variability parameters stored in the **SRCLIST** tables **masterfile.fits** with those stored in **checkfile.fits**. For any column (eg **RA** or **V_CHI2** the values in each row of the two tables are compared, and any difference in the quantities that exceeds a tolerance of 1e-6 is considered to be an error and the test-harness fails outputting details where the failure occurred.

  checkoutput also checks that the variability parameters are computed properly by computing the parameters for 10 unique sources, each having a hard-wired list of multiply-detected sources, and comparing the computed and hard-wired values. Any difference greater than 1e-6 is reported as an error.

For more thorough development testing, the number of **SWS** files can be set to any number by editing the program **createfiles**.

11 Future developments

References