



ommergelists

November 4, 2014

Abstract

This program produces an **OM observation** source-list file from the merging of the two observation source-list files that are produced by the **omichain** when source-detection has been conducted on the mosaiced sky-images. The output observation source-list will contain all the sources from the source-list produced from the individual exposure images, plus the unique detections from the source-list produced from the mosaiced sky-images.

1 Instruments/Modes

Instrument	Mode
OM	IMAGING

2 Use

pipeline processing	yes
interactive analysis	yes

3 Description

This program has originated due to the new **OM** functionality of source-detection on mosaiced sky-images. It matches the sources from the first input observation source-list file (ie those detected from the individual exposure images) with those from the second input observation source-list file (ie those that have been detected on the mosaiced sky-image), and propogates these matches into an output source-list with the following caveats:

1. Any **unmatched** sources from the second-list having a raw count-rate (including the background) \leq **maxrawrate** value are added to the output observation source-list file.
2. Often fields are imaged through more than one filter. An object exists in a source-list if it is detected through a minimum of one filter. Non-detections of the known source through other filters are not flagged. However, if the second source-list is constructed using, for example, a deeper mosaicked image then it can potentially contain new detections of a known source which has already been found in the shallow images of individual exposures,



through different filters. Provided new detections in list 2 of known sources already found in other wavebands in list 1 have intrinsic + background count-rates ≤ **maxrawrate** then they are propogated into the new combined source-list.

- 3. Deeper exposures provide a more significant detection and better constrained source properties. If there is a source match between the two lists through a specific filter, the properties of the most significant detection are adopted within the output list. An exception to this rule occurs if a source with intrinsic + background count-rate > **maxrawrate**. In these bright-source cases the properties derived or combined from individual images are adopted within the output list. This exception prevents systematic errors related to combining different CCD dead-times. frame-rates and coincidence-losses from propogating into the bright-source sample.

The **maxrawrate** value is an optional input parameter, with a default value of 5 counts/sec; below this threshold a coincidence-loss correction to photon rates is not required within photometric conversions. The reason why there is a maximum raw-rate cut-off is due to the way **OM** photometry is conducted. A coincidence loss correction is applied to the count-rates and this correction depends on the frame-time and dead-time fraction. These CCD properties vary from exposure to exposure and so coincidence-losses cannot be corrected in mosaicked images with the current SAS algorithms. By propogating only the faint sources from the second list into the output table we avoid systematic bright-source photometry error, while retaining the full potential size of the source sample within the output.

Since this cut-off is determined by the input parameter **maxrawrate**, the user has complete control over what sources are retained. It should be noted that for the photometry of the sources detected on the mosaiced sky-images, no coincidence-loss correction will have been applied and the dead-fraction correction will have been incorporated into the exposure images.

Ommergelists will compare the celestial coordinates of the matched sources from both source-lists and compute mean **RA** and **Dec** offsets and add them to the coordinates of the second source-list to correct for any small differences in their origins (it will give a warning about any significant offsets indicating bad astrometry).

The program can also produce graphs comparing various properties of the common sources in the two lists, which can be a useful check for possible problems. It can also produce a **ds9** region file that allows the new sources detected on the mosaiced sky images to be easily identified.

4 Parameters

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

Parameter	Mand	Type	Default	Constraints
-----------	------	------	---------	-------------

srclist1	yes	string	none	
-----------------	-----	--------	------	--

Name of the OM observation source-list file that has been created by **omsrclistcomb** from the processing of the exposure images.

srclist2	yes	string	none	
-----------------	-----	--------	------	--

Name of the OM observation source-list file that has been created by **omsrclistcomb** from the processing of the mosaiced sky-images. **Please note that these two input files can be given in the wrong order.**



outset	yes	string	none	
---------------	-----	--------	------	--

Name of the output observation source-list file (**To prevent an existing file with this name from being over-written the SAS environment variable SAS_CLOBBER should be set to 0**)

maxrawrate	no	real	5.0	0.0-
-------------------	----	------	-----	------

The maximum allowed raw count-rate (including background) of the unique sources in the second source-list file to be added to the output list.

tolerance	no	real	2.0	1.0-
------------------	----	------	-----	------

Maximum allowed angular-distance (arcsecs) between two sources to be treated as identical.

regionfile	no	string	""	
-------------------	----	--------	----	--

Name of a file to store ds9 regions that display the common sources in both input source-list files (red regions), the ones unique to the first one (green regions) and the ones unique to the second one (blue regions).

plotfile	no	string	""	
-----------------	----	--------	----	--

Name of a postscript file to store various graphs comparing parameters of the matched sources in the two source-list files.

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.

Identical source-list files- stopping (*fatal*)

name of the file

Non-observation source-list file- (*fatal*)

Can only process observation source-list file!



File does not exist! (*fatal*)

Name of an input source-list file

Error loading column from SRCLIST table- (*fatal*)

Name of the column

Output file exists (*warning*)

corrective action: Ignore (if you don't want the file overwritten set the environment variable **SAS_CLOBBER** to 0)

6 Input Files

Two input **OM** observation source-list files are required (ie those produced by **omsrlistcomb**).

7 Output Files

1. An **OM** observation source-list file (having the same format as that produced by **omsrlistcomb**) The **RA_CORR** and **DEC_CORR** columns in the **SRCLIST** table will contain the source celestial coordinates after correction for any **RA** and **DEC** offsets between the two input source-lists. It will also contain the extra columns **FILT_SKY** (where **FILT** refers to an **OM** filter) that will identify if the source parameters for that particular filter came from the second input source-list file (**YES**) or not (**NO**).
2. (Optional) A ds9 region file showing which sources are common to both source-list files, and those unique to each file..
3. (Optional) A postscript plot file containing various parameters of the sources matched in the two input source-list files.

8 Algorithm

The program is written in **c++**. Within the following description of the algorithm, the term 'table' refers to an XMM DAL (Data Access Layer), API (Application Programmable Interface) structure that provides access (read, create or modify) to a **FITS** table.

The program works in the following way:

1. Carry out the following tests on the input and output files given on the command-line
 - Check that both input source-list files exist- if not stop the program with an error message.
 - Check that both source-lists contain at least one source- if not issue a warning and stop the program.
2. Read in the **RA** and **Dec** values from the **RA_CORR** and **DEC_CORR** columns, respectively, in the **SRCLIST** table (**srlist1**) of the product observation source-list file obtained from the non-sky exposure images.



3. Read in the **RA** and **Dec** values from the **RA_CORR** and **DEC_CORR** columns, respectively, in the **SRCLIST** table (**srclist2**) of the product observation source-list file obtained from the mosaiced sky-images.
4. Produce a list of the sources that are found in both source-lists, by matching both lists using their celestial coordinates and using a maximum matching angular separation of **tolerance**, where **tolerance** is an input parameter and defaults to 2 arcsecs. **matchedIndex1** will contain a list of the row numbers in **srclist1** that have a match in **srclist2**, **matchedIndex2** a corresponding list of the row numbers in **srclist2** that have a match in **srclist1** and **unmatchedIndex2** a list of the row numbers in **srclist2** that were not matched with any source in **srclist1**.
5. Compute the **RA \ Dec** offsets between the matched coordinates of the two lists.
6. Create the output file with a **SRCLIST** table (**srclist3**) by copying **srclist1** to **srclist3**.
7. Create a list of the **OM** filters that are found in both tables (maybe less in **srclist2**)
8. increase the size of the **srclist3** by the size of **unmatchedIndex2**. The column elements along each of these rows are initially set to null (indicating that the element has no value attached to it) and will be populated with the unmatched data in the next stage..
9. Loop through all the filters that are present in **srclist2** and
 - (a) For the current filter **FILT**
 - (b) Load the source raw-count rate from column **FILT_RAW_RATE** and significances from column **FILT_SIGNIFICANCE** in **srclist2**.
 - (c) Loop through from $i=1$ to $i=\text{the size of } \mathbf{matchedIndex2}$
 - i. Set j to **matchedIndex1**[i] and k to **matchedIndex2**[i].
 - ii. If source[k] in **srclist2** has a non-null raw count-rate and a value no greater than 5 then- **EITHER**
If the source parameters for source[j] in **srclist1** have null values (ie not detected) then set the values in **srclist3** to the corresponding values in **srclist2**. **OR**
 - iii. If the significance of source[k] in **srclist2** has a greater significance than the source significance of source[j] in **srclist1** then set the filter columns for filter **filt** for row **matchedIndex1**[i] in **srclist3** to the corresponding values in **srclist2**.
 - (d) Loop through from $i=1$ to $i=\text{the size of } \mathbf{unmatchedIndex2}$
 - i. Set j to **unmatchedIndex2**[i]
 - ii. If the total raw count-rate of source[j] for filter **FILT** of **srclist2** has a value no greater than the **maxrawrate** value then set the parameters for the **FILT** columns in **srclist3** to the corresponding values to the **FILT** columns in **srclist2**.
 - (e) When a row is appended to the final table from the unmatched sources in source-list 2, and that row is populated with all NULLS because the count-rates for all filters > **maxrawcountrate**, then the row is deleted before the output table is written.
 - (f) Delete the **RAW_RATE** columns. **Note that omsrclistcomb added these columns to srclist2 because the input source-lists originated from mosaiced sky-images.**
 - (g) For each filter present, add a string column **FILT_SKY** to **srclist3**. Each row of the new column will be set to **YES** if the data for that particular filter originated from a mosaiced-image, or **NO** otherwise.
10. Sort **srclist3** into increasing **RA** order.
11. If input parameter **plotfile** has a valid name, draw graphs that compare various properties of the matched sources in **srclist1** and **srclist2**.



12. If input parameter **regionfile** has a valid name, produce a **ds9** region file that contains the matched sources (**red regions**), those unique to **srclist2** (**blue regions**) and those unique to **srclist1** (**green regions**).

8.1 Source-matching algorithm

The matching algorithm takes into account that more than one source in source-list 1 may have a match with the same source in source-list 2. This can become important in crowded-fields. For example, sources 1 and 2 in source-list 1 may be matched with source 10 in source-list 2, with separations of 2 and 1 arcsecs, respectively. In a simple matching algorithm, source 1 could be matched with source 10, when in fact source 2 of source-list 1 is better matched with it..

In the following, a **list** is a c++ vector that can grow in size as necessary, and each element of the vector is also a vector which can grow in size.

The algorithm works as follows-

1. Create an empty list, **matchedindex1**, that will record the row-number of a source in source-list 1 that has a match with a source in source-list 2.
2. Create an empty list, **matchedindex2**, that will record the corresponding row-number of a source in source-list 2 that has been matched with a source in source-list 1.
3. Create an empty list, **matcheslist**, of size equal to the number of rows in source-list 1, that will store the row-numbers and distances of the sources in source-list 2 that are no greater than the distance **tolerance** from each source in source-list 1.
4. Loop through from $i=1$ to the number of rows in source-list 1 and for each one:
 - Find all the sources in source-list 2 that are no more than a distance **tolerance** arcsecs from it, and add the source-list 2 row-numbers and distances to **matchesList[i]**.
5. Loop through **matchesList** and for each element sort the list of sources in source-list 2 that have 2 or more matches with a source in source-list 1 into increasing distance order, preserving the correct source-list 2 row numbers.
6. Loop through from $i=1$ to $i=\text{size of source-list 1}$ and
 - If source i has no matches move on to the next i .
 - Loop through from $j=1$ to $j=\text{size of matchesList}[i]$ and
 - Set l to the index of **matchesList**[i][j]
 - Check to see if this source in source-list 2 has been matched with any other source in source-list 1
 - **IF NOT** then increase the size of **matchedIndex1** and **matchedIndex2** by 1, and set the former to i and the latter to l . Move on to the next i .
 - **IF YES** Check to see if the distance in **matchesList**[i][j] is no greater than any of the other matched distances for this source in source-list 2- if not then increase the size of **matchedIndex1** and **matchedIndex2** by 1, and set the former to i and the latter to l . Move on to the next i .
7. Using the refined matched-list of sources, compute the mean differences in **RAxcos(Dec)** and **Dec**.
8. If either of these offsets are more than 0.1 arcsecs then correct the coordinates of the second source-list and repeat the matching process (a maximum of 3 iterations).



8.2 SRCLIST table sorting algorithm

The algorithm for sorting a **SRCLIST** table into increasing **Right Ascension** order is designed to work with a table with any number of columns that contains either **RA_CORR**, **DEC_CORR** or **RA**, **DEC** columns, with the former taking precedence. The algorithm works as follows-

1. Produce an index list of the sorted **Right Ascensions** positions that gives the row number that each source in the table should be placed to be in the correct order.
2. Double the number of rows in the table.
3. Copy the original rows of the table to a position starting at the the last row in the original table plus one.
4. Loop through the number of rows in the original table and replace each one by the appropriate sorted row from the appended rows using the index list.
5. Delete the appended rows from the table.

8.3 Notes

- Astrometrically-corrected celestial coordinates (ie **RA_CORR** and **DEC_CORR** columns) will be used if present.
- All **FITS** file operations are performed using the **SAS Data Access Layer** interface.
- All graphics is done using the **C** binding library **cpplot**, which allows **PGPLOT** routines to be called from **c/c++**.
- All arrays are implemented as **c++ vectors**.

9 Example of a Region file produced by ommergelists

The following diagram was obtained from ommergelists by reducing the **OM** data for observation 0105260601 using the **omichain**.

Figure 1 shows the regions produced by **ommergelists** overlaid on the **V-band** mosaiced sky-image. The red regions are the sources that are common to both input source-list files. The blue regions are the ones only found in the second observation file (the sources from the mosaiced sky-images) and the green ones those unique to the first observation source-list file (the sources from the exposure images). As can be seen, there are a few spurious blue sources associated with the scattered light feature near the centre.

10 Example of comparison plots produced by ommergelists

The optional postscript plot file can be useful in identifying any problems in the processing of the mosaiced sky images. The following diagrams were obtained from ommergelists by reducing the **OM** data for observation 0105260601 using the **omichain**.

Figure 2 compares the **RA** and **Dec** differences in coordinates from the matched sources of the two input files (ie **srclist1 - srclist2** source). At the top of these two histograms, the mean, median and mode

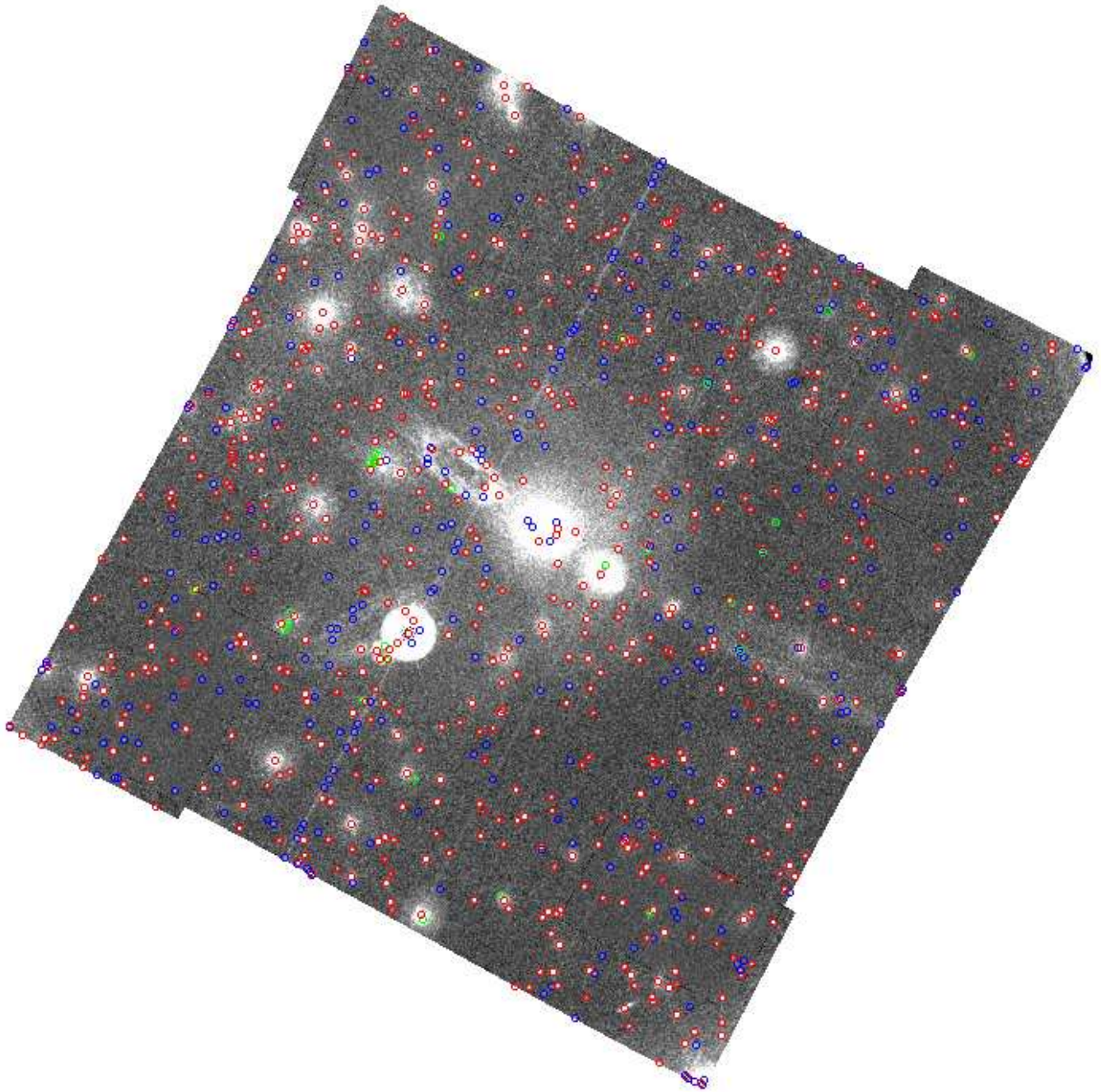


Figure 1: V-Band mosaiced sky-image from observation 0105260601, with the regions produced by **om-mergelists** overlaid

values of the distributions are given, the latter using the method of Bickel 2002 (Computational Statistics Volume 39, Issue 2, April 2002, Pages 153-163). It is a good idea to check these diagrams for skewness, which may indicate that the astrometry is not too good- the difference between the mean and mode should usually be no greater than about 0.2 arc seconds, and the median should be in between the two.

Figures 3 to 5 compare various parameters of the matched sources from the two input source-list files. In these diagrams, the **1** and **2** on the **x** and **y** axes, respectively, refer to the matched sources from **srclist1** and **srclist2**, respectively.

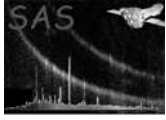


Figure 3 shows an example of plots comparing the significances, instrumental magnitudes, corrected count-rates and histograms of the differences in extension flags, from the matched sources of the two input files.

11 Testing

The aim of the test is to check that **ommerge**lists

1. Correctly constructs the new output source-list file by
 - Checking that the merged SRCLIST table contains all the columns in the SRCLIST table of the first source-list file plus the columns unique to the SRCLIST table of the second input source-list file.
 - Correctly finds the identical sources in the two input source-list files.
 - The data in each row of each column of the merged SRCLIST data is correct.
 - Correctly sorts the merged table into order of increasing Right Ascension.

A program **createtest** creates the test files and works as follows-

1. It creates an observation source-list file (**test1.fits** that contains the columns **RA_CORR**, **DEC_CORR**, **U_RAW_RATE**, **U_CORR_RATE** and **U_EXTENSION_FLAG**.
2. A random number of rows, between 10 and 1000, is added to the table.
3. The table is filled by generating random values for each row of each column.
4. A second observation source-list file (**test2.fits** is created that contains the columns **RA_CORR**, **DEC_CORR**, **V_RAW_RATE**, **V_CORR_RATE** and **V_EXTENSION_FLAG**.
5. A random number of rows, between 10 and 1000, is added to the table.
6. The table is filled by selecting random rows from the first SRCLIST table and by filling the columns of the unfilled rows unfilled with random values.
7. A third source-list file **test3.fits** is then formed from **test1.fits** and all the unique sources in **test2.fits** with raw count-rates no greater than 5.0 are added to it. This file will therefore contain columns for filters **U** and **V**.
8. The **SRCLIST** tables in the files **test1.fits**, **test2.fits** and **test3.fits** are sorted into increasing RA order.
9. **ommerge**lists is then executed with **test1.fits** and **test2.fits** as input source-list files and it produces a merged source-list file **test4.fits**.
10. A program **checkout** is then run to check the following
 - That the **SRCLIST** table in **test3.fits** contains the same number of rows as the **SRCLIST** table in **test4.fits**.
 - That the RA values in the **SRCLIST** table of **test3.fits** are in order of increasing RA.
 - That each value in each row of each of the columns **RA_CORR**, **DEC_CORR**, **U_RAW_RATE** and **B_RAW_RATE** in the **SRCLIST** table of **test3.fits** are identical to the corresponding values in **test4.fits** to within a tolerance (0.0001 arcsecs for **RA_CORR** and **DEC_CORR** and 1e-6 for the remainder)-between any bigger difference is reported as an error and the test is deemed to have failed.



- That the algorithm for computing the angular distance between 2 sources is correct. One thousand sources with random RAs and Decs are produced and a second list of sources is produced from this having small known angular separations. The distances between each source in the first list and its corresponding one in the second list is computed using the algorithm that **ommerge**lists and any deviation greater than 0.001 arcsecs from the known deviation is reported as an error and the test deemed to have failed.

The actual testing is controlled by a shell program in the **ommerge**lists test directory and is initiated by typing the command **make test**. To do the testing in **low memory** mode, you should set the environment variable **SAS_MEMORY_MODEL** equal to **low**. **Note** that at the daily builds at Villspa the testing of all SAS software packages is done in both **high** and **low** memory modes (referring to the memory model used by the **SAS Data Access Layer**).

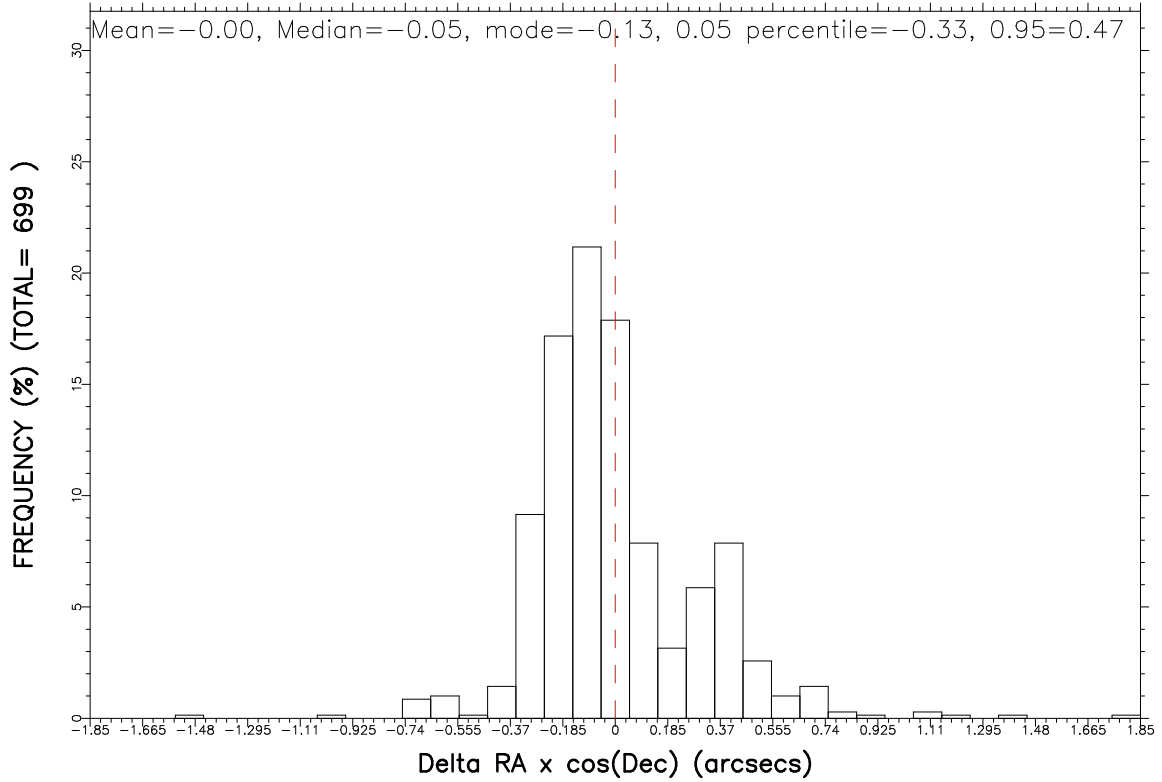
For more demanding development testing, the number of **SWS** files can be set to any number by editing the program **createfiles**- tests have been done by setting the number of rows in the **SRCLIST** table in **test1.fits** to 1,000,000.

12 Future developments

References



0105260601 Right Ascension comparison



0105260601 Declination comparison

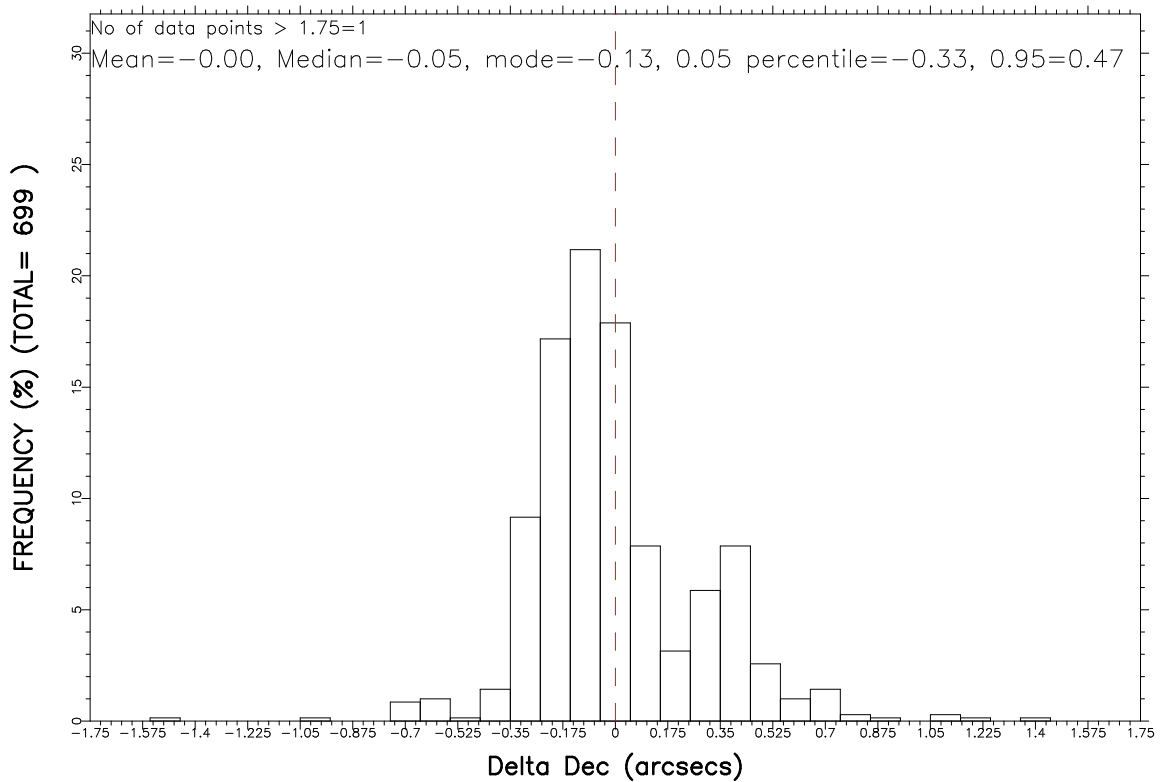


Figure 2: Comparison of matched celestial coordinates

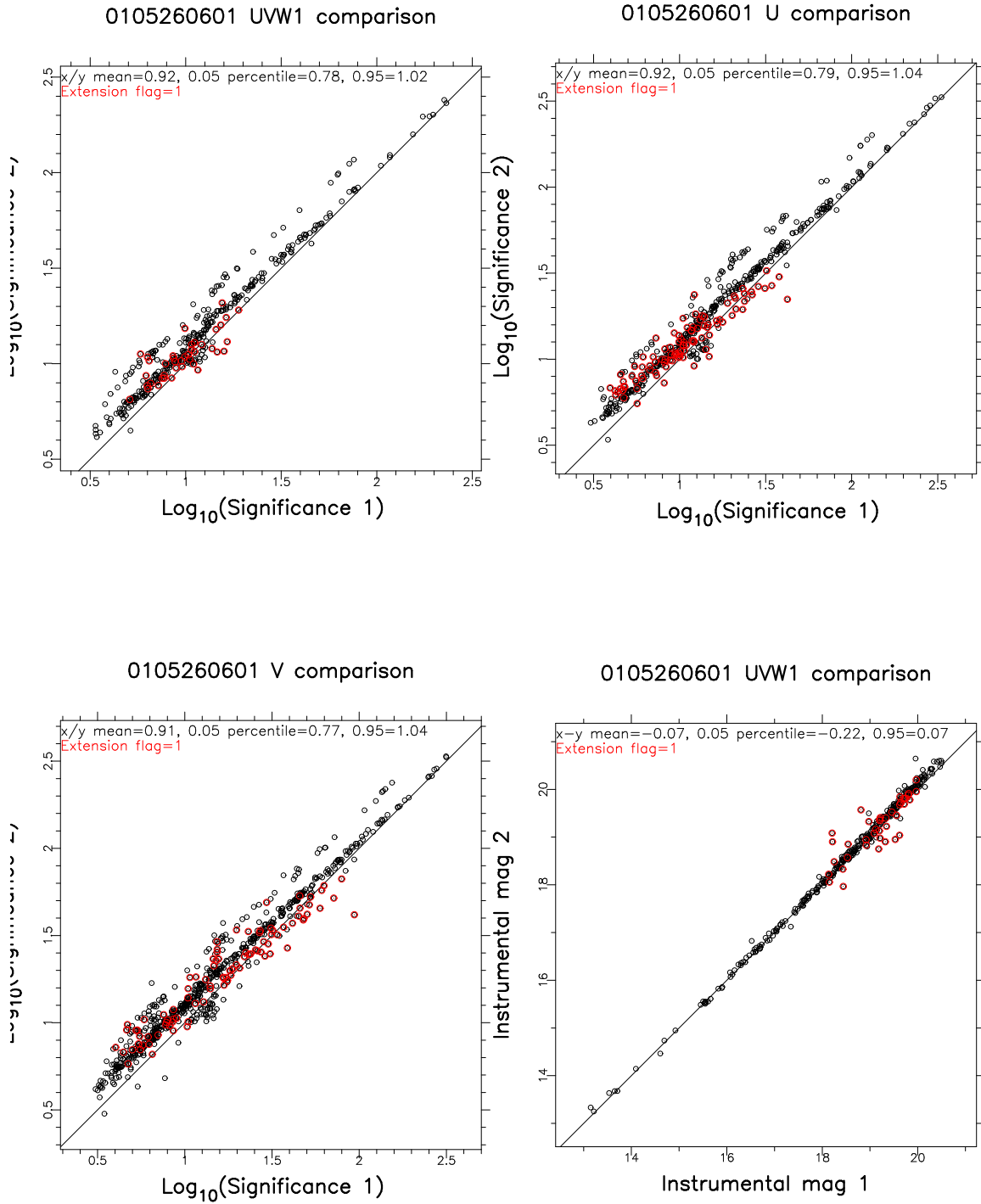


Figure 3: Comparison of various parameters of the matched sources from the two input source-lists

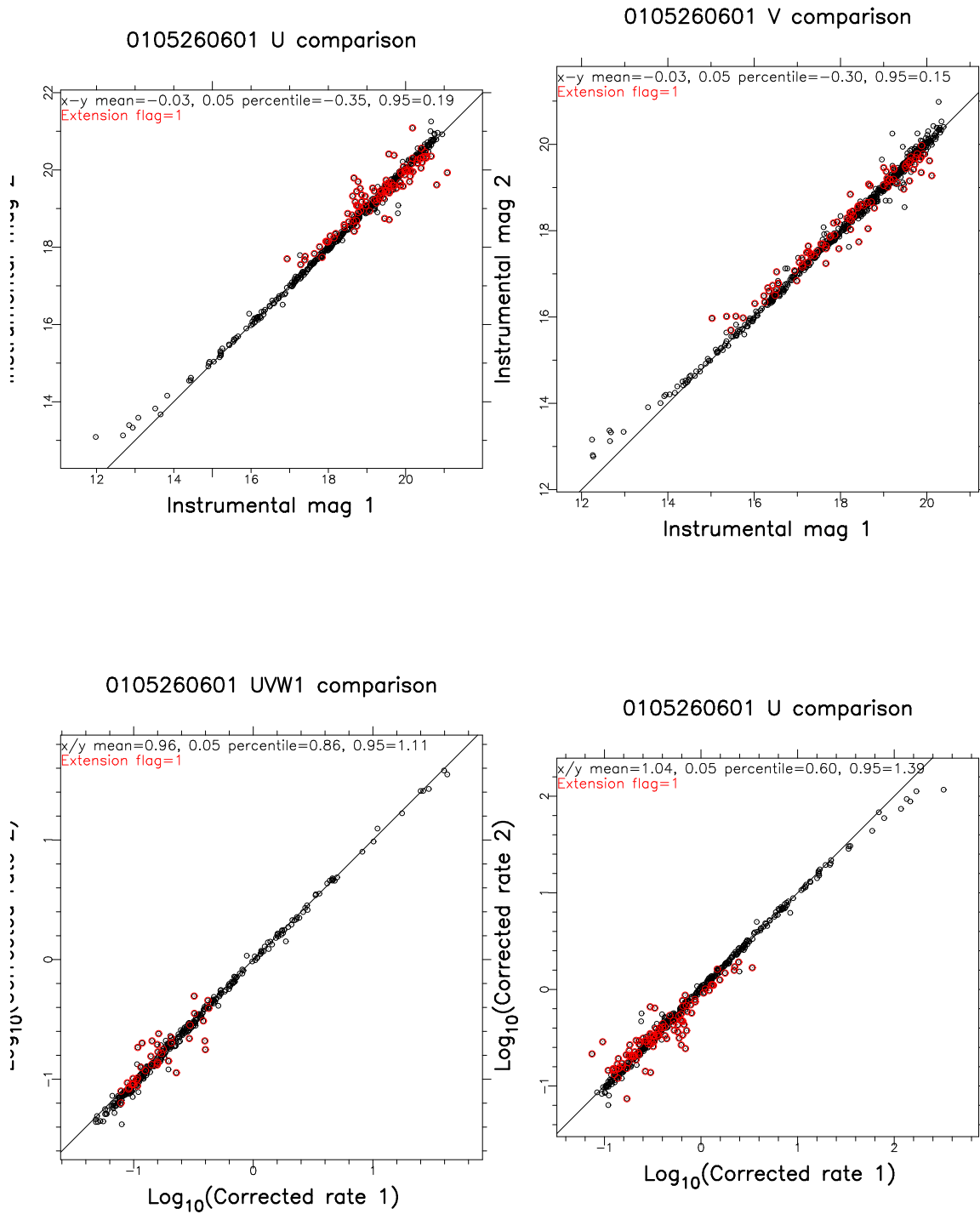


Figure 4: Comparison of various parameters of the matched sources from the two input source-lists-continued

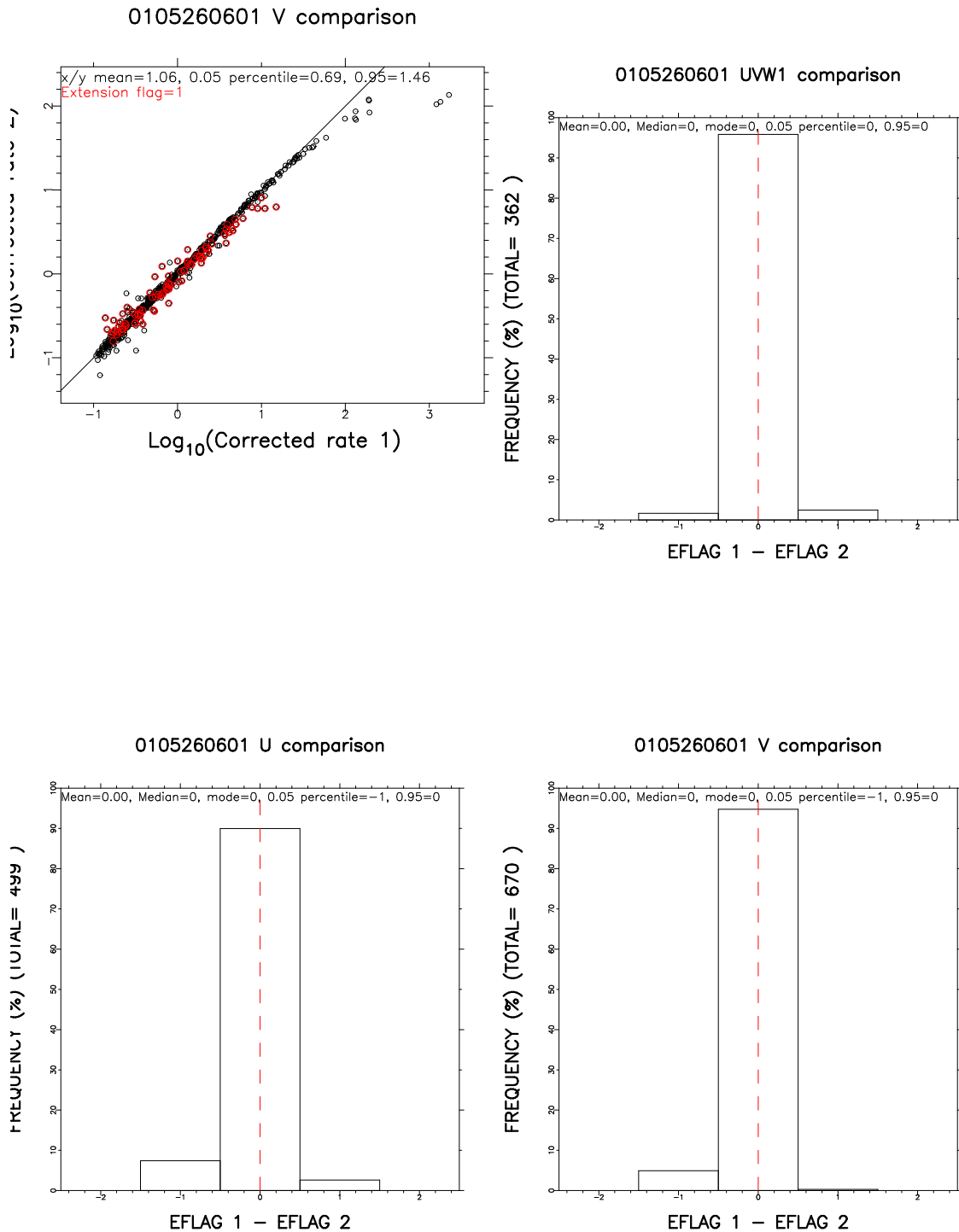


Figure 5: Comparison of various parameters of the matched sources from the two input source-lists-continued