eimsimreduce

November 4, 2014

Abstract

This is part of the package **eimsim**, which is a collection of routines used to make simulated XMM-Newton x-ray images, to perform source detection on them, and to assess the results.

1 Which documents to read

At last count, there are 25 tasks within the package **eimsim**, each with its own documentation (well, they will have, eventually). However, you don't need to read 25 sets of documentation! I recommend that you read the documentation for only 4 of the tasks, in the following order:

1. **eimsim**
2. **eimsimprep**
3. **eimsimbatch**
4. **eimsimreduce** (ie, the present document)

The documentation for **eimsim** contains all generic information relating to the package as a whole, such as the change history.

2 Description

As described in the **eimsim** documentation, the functionality of package **eimsim** is divided into things that need to be done only once and things which must be done N times for N simulation runs. The 'done once' things can be further divided into those things which must be done before the simulation runs, and those things which must be done afterward. The 'before' things are gathered into **eimsimprep**; the 'N simulation runs' things are to be found within **eimsimbatch**; and the 'afterward' things are performed within the present task **eimsimreduce**.

The task **eimsimreduce** does 4 things in sequence, which are described individually in the following four subsections.

2.1 Merge lists of detections

This function may be performed alone by calling the script with `entrystage` and `finalstage`='merge'. The task counts the number of files in subdirectory `simgensubdir` whose names match the detected-sources name template; all these FITS files are then merged into one. The number of files found is written to a keyword `N_FIELDS` in the first extension of the merged dataset.

This merged file alone allows some interesting comparisons to be made. For example, one could make a scatter plot of `FLUX` against `SIM_FLUX` (see eg figure 1), to see how accurately the detection procedure can measure source flux, and to check to see if any of the weaker, non-XMM-PSF sources have been detected; or a scatter plot of `MATCH_PNULL` against `SIM_FLUX` (see eg figure 2).

2.2 Look for biases

This function may be performed alone by calling the script with `entrystage` and `finalstage`='bias'.

Note that action is taken only if the merged source list contains a keyword `COMPARED` which has the value 'T'. This keyword is written to the lists of detected sources by the task **srccompare**, called as part of default **eimsim** functioning.

The function calls a task `eimsimbias`, which constructs some histograms from the merged list of detections, with the aim of comparing the detection procedure's estimates of the values of position and flux with the true values. The estimated uncertainties in these parameters are also compared with the scatter in the estimates around the true values. As you can see from the description in the **eimsim** documentation of the procedure which attempts to match detected with simulated sources, it is vital that the detection procedure calculates accurate estimates of not just the position and flux of each source, but also the uncertainties in these quantities. The histograms created in the present section can help one to determine if there is a bias in any of these quantities.

Six histograms are made and all are written to separate tables in the output dataset. These histograms are described in the following subsubsections:

2.2.1 For p = each of the input source-list columns `X`, `Y` and `FLUX`: makes a frequency histogram of relative p offset.

'Frequency histogram' means that the area under this histogram is equal to the number of rows N in the merged list. The x axis of the plot is the quantity

$$\delta_i = \frac{\Delta p_i - \frac{1}{N}\sum_1^N \Delta p_i}{\sigma_{p,i}}$$

where

$$\Delta p_i = p_{\rm det} - p_{\rm sim}.$$

The interpretation of these p quantities in column terms, taking for the sake of an example the Y spatial coordinate, is $p_{\rm det}$ = `Y`, $p_{\rm sim}$ = `SIM_Y` and $\sigma_{p,i}$ = `Y_ERR`.

These histograms are written to tables named p`REL`.

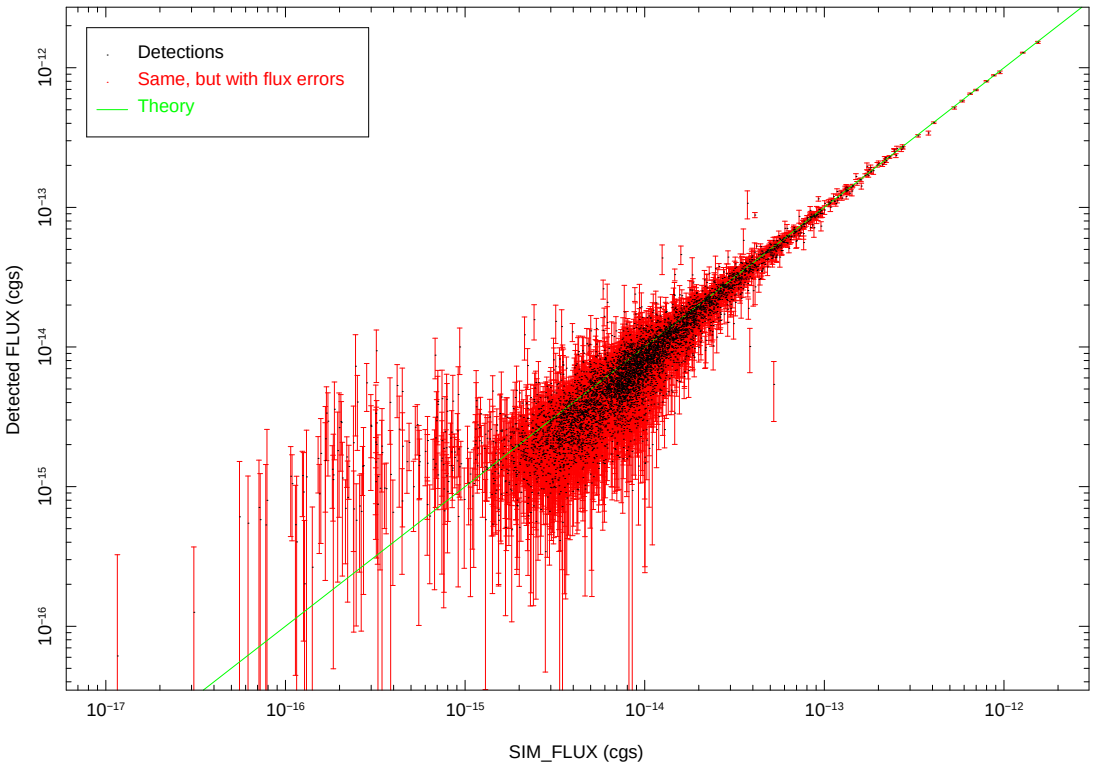

Figure 1: Fluxes of simulated vs detected sources (100 runs, 2xmm detection).

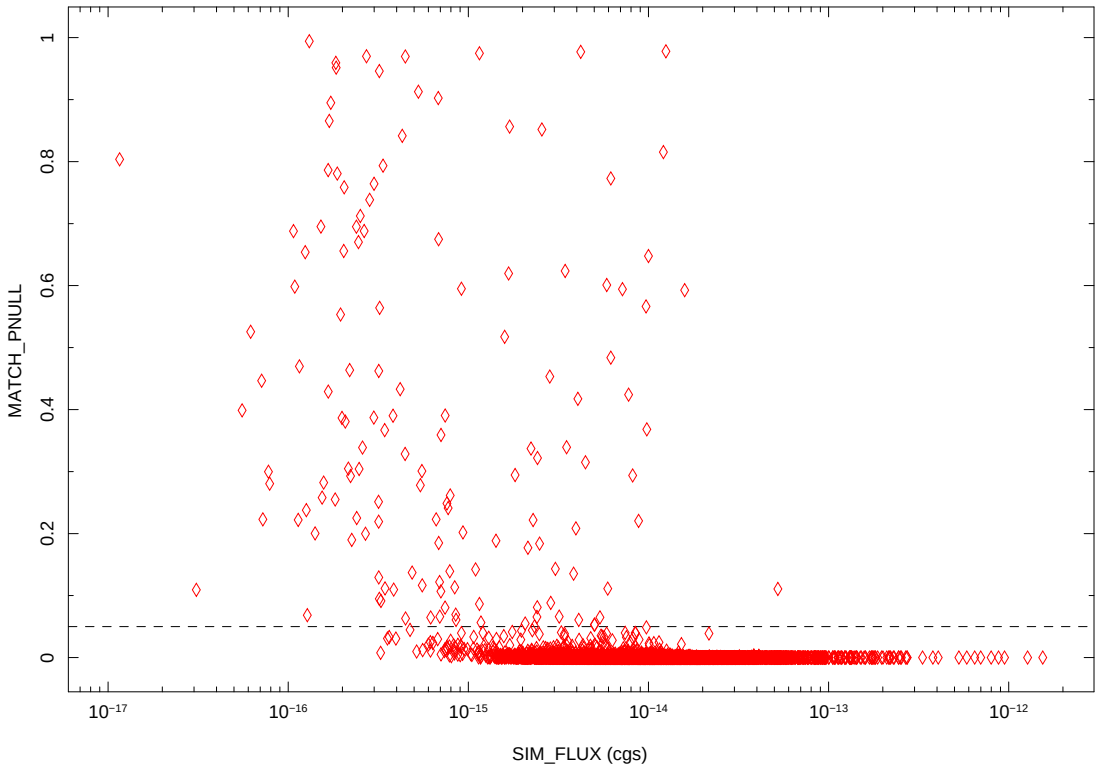

Figure 2: The variation of matching probability with source brightness (100 runs, 2xmm detection).

2.2.2 For each of the input source-list columns X and Y: a histogram of position against flux

These are not so much histograms as binned averages of various position-related quantities at a series of values of detected flux. This pair of histograms is written to tables X_VS_S and Y_VS_S in the output data set.

Log10 of the flux values are taken before the binning process is begun.

A description of the columns written (taking again the Y spatial coordinate as the example) is as follows:

- FLUX_MID: ith value given by 10 to the power of $(\log_{10}[S_{i+1}] + \log_{10}[S_i])/2$ where S is taken from column FLUX.
- Y_DIFF: the bin-averaged value of Y - SIM_Y.
- Y_DIFF_STDDEV: for each bin, the standard deviation σ of the values of Y - SIM_Y which fall within that bin, according to the formula

$$\sigma^2 = \frac{1}{N-1}\sum_{i=1}^{N}(\Delta p_i - < \Delta p >)^2$$

  where $\Delta p = p_{\rm det} - p_{\rm sim}$ and N is the number of values within the bin. σ is of course undefined for $N < 2$.
- Y_DS_ERR: a rough estimate of the uncertainty of Y_DIFF_STDDEV, derived by dividing Y_DIFF_STDDEV by the square root of the number of values in the bin.
- Y_AVE_ERR: this is the simple mean of values of the input source list column Y_ERR which fall within the histogram bin.

2.2.3 Same for column FLUX

...although the x axis here is given by SIM_FLUX of the input source list, not FLUX.

2.3 Generate some 'completeness' histograms

This function may be performed alone by calling the script with entrystage and finalstage='completeness'.

The main aim of this function is to produce a histogram which shows the fraction of the simulated sources which have been detected, as a function of the flux of the simulated source. One expects this to be close to 1 in the bright limit, but to fall to zero towards the faint end. The flux at which the detected fraction falls to about 1/2 can be considered the sensitivity of the detection technique which was employed. See figure 3 for an example of a plot of cumulative completeness.

Note that any sensitivity figure obtained in this fashion represents an average across the entire mosaiced field of view. Typically the exposure and detected background flux vary greatly over such a mosaic. If precise sensitivity figures are desired it would probably be better to use artificial exposure and background templates, in which the pixel values for each instrument and energy band were either constant>0 or 0. The non-zero area would also need to be the same shape and extent for each instrument.

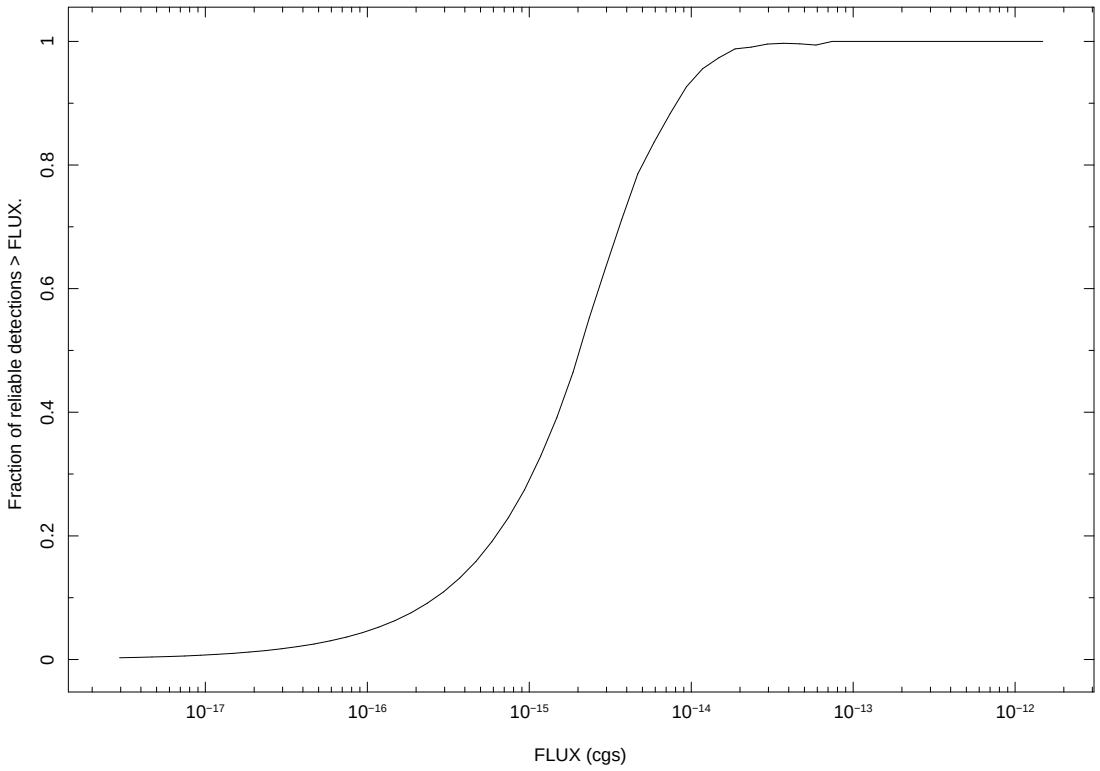

Figure 3: Cumulative completeness (100 runs, 2xmm detection).

The first step performed by the present function is to make a histogram of the occurrence of simulated sources as a function of $\log_{10}$(SIM_FLUX). All the available lists of simulated sources are harvested in this step. The columns created are

- LOG10_SF_LO, $\log_{10}$ of the lower edge of the bin.
- LOG10_SF_HI, $\log_{10}$ of the upper edge of the bin. The bins all have equal widths in $\log_{10}$ space.
- N_SIM, the total number of 'detectable' simulated sources (ie, those for which INV_SENSY > 0) which have fluxes that fall within the bin.
- NET_FLUX, sum of SIM_FLUX within the histogram bin.

Some possibly useful additional columns are next calculated from these:

- N_SIM_ERR: Square root of N_SIM.
- N_SIM_INT: Reverse-cumulative total of N_SIM (ie, sum of N_SIM in this plus all brighter bins).
- DENS_SIM: Average sky-density (in $\deg^{-2}$) of (detectable) sim sources. This is N_SIM divided by SKY_AREA divided by the number N of source lists which were merged. SKY_AREA, which is read from the keyword of that name in the lists of detected sources, is the area in square degrees of the non-zero parts of the mosaiced maps of reciprocal sensitivity, which were constructed as part of **eimsimprep**.
- DENS_SIM_ERR: Error in DENS_SIM.
- DENS_SIM_INT: Reverse-cumulative total of DENS_SIM.

- FLUX_DENS_INT: Forward-cumulative total of NET_FLUX, divided by the sky area.
- SIM_FLUX_LO: SIM_FLUX at the lower edge of the bin (10.0**LOG10_SF_LO). This should be used as the x-value when plotting any 'reverse-cumulative' quantity on the y-axis.
- SIM_FLUX_MID: SIM_FLUX at the middle of the bin. This should be used as the x-value when plotting any 'differential' quantity on the y-axis.
- SIM_FLUX_HI: SIM_FLUX at the upper edge of the bin (10.0**LOG10_SF_HI). This should be used as the x-value when plotting any 'forward-cumulative' quantity on the y-axis.

Now it is time to tally up the detections. However, we need now to make a distinction between detections which are likely to be 'genuine' and those which are not. 'Genuine' is a somewhat slippery concept in present application, but we do have a quantity which we can use to get a handle on it, namely the probability MATCH_PNULL that the match between a detection and its matching simulated source could have occurred by chance. We define a cutoff value of MATCH_PNULL and declare that all those detections for which MATCH_PNULL falls below the cutoff are genuine, and the others not. The cutoff is under user control via the parameter probcutoff of **eimsimreduce**. Detected sources for which SIM_INV_SENSY = 0 are also screened out at this stage.

The situation is actually even a little bit more complicated, due to the fact that, although we may be fairly confident that genuine detections have small values of MATCH_PNULL, spurious detections have values which are evenly spread between 0 and 1. This means that our initial tally of detections with MATCH_PNULL below the cutoff $P_{\rm cut}$ comprises not the total number R of reliable detections, out of a total A, but $R' = R + P_{\rm cut} \times (A - R)$ - ie there are some black sheep among the white. R is thus calculated from R' as

$$R = \frac{R' - P_{\rm cut} A}{1 - P_{\rm cut}}$$

and

$$\sigma_R^2 = \frac{R' + P_{\rm cut}^2 A}{(1 - P_{\rm cut})^2}.$$

The next batch of columns to be calculated are as follows:

- N_ALL_DET: All detections (same bin edges as for the simulated sources).
- N_ALL_DET_ERR: Square root of N_ALL_DET.
- N_ALL_DET_INT: Reverse-cumulative total of N_ALL_DET.
- N_TRUE_DET: 'Genuine' detections, as given by the formula above.
- N_TRUE_DET_ERR: Uncertainty in N_TRUE_DET, as given by the formula above.
- N_TRUE_DET_INT: Reverse-cumulative total of N_TRUE_DET.
- DENS_DET: Average sky-density (in $\deg^{-2}$) of reliable detections, derived in the same fashion as column DENS_SIM.
- DENS_DET_ERR: Uncertainty in DENS_DET.
- DENS_DET_INT: Reverse-cumulative total of DENS_DET.

The desired result is then calculated and expressed in the final three columns:

- COMP_RATIO: DENS_DET/DENS_SIM.

- COMP_RATIO_ERR: Error in COMP_RATIO.
- COMP_RATIO_INT: DENS_DET_INT/DENS_SIM_INT.

A last function of this task is to append to the output dataset a table named THEORY, which is a version of the SRCSPECS table of the sim source specification template designed to make it easy to compare the theoretical logN-logS of the simulated sources with the actual logN-logS. You can do this for example using the ftool fv. If you plot first DENS_SIM_INT against SIM_FLUX_INT; then overlay this with a second plot, of THEORY columns DENSITY against FLUX; then change the axes scales to log-log; you will see what I mean.

PLEASE NOTE if you do this that the real distribution will very often appear not to match the theoretical logN-logS very well at the bright end of the scale. Such deviations appear more significant than they really are, because the brain expects the values in adjacent flux bins to be statistically independent, which is not true of a cumulative plot. A comparison of differential plots is often much more satisfying.

2.3.1 What's the difference between DETEC_PNULL and MATCH_PNULL?

Both of these values are attempts to quantify the probability that a source detection arises not from a real point source but from a chance fluctuation of background. It would be nice to be able to say absolutely whether a given detection arose from source or background, instead of having to rely on fuzzy estimates of probability. But without knowing the origin of every x-ray photon, either within this simulation or in the real world, it is not possible to know whether even the brightest and seemingly most significant bunch of events on a CCD is really due to a bright source at that location or just due to a chance grouping in space and time of photons in the far PSF wings of distant, perhaps faint sources. All that one can do is to calculate a probability for it, based on the average level of background and the number of events comprising the detection. This number is DETEC_PNULL.

Within the context of the simulation we are a bit better off, since we have *a priori* knowledge of both the positions of the sources in any instance and the probability density of their distribution in general. This extra information enables us to make a more accurate estimate of the null probability, which is MATCH_PNULL. By comparing DETEC_PNULL (which is calculated by the detection script, thus via a method not under the direct control of **eimsim**) with MATCH_PNULL we can assess how effectively the detection task is calculating this probability.

2.4 Generate some 'reliability' histograms

This function may be performed alone by calling the script with entrystage and finalstage='reliability'.

It is desirable to discard detections for which the probability $P_{\rm null}$ is high that the detection arose by a chance fluctuation of the background. It is assumed here that any detection scheme called by **eimsim** will calculate this probability and store it in the DETEC_PNULL column of the output source list. Generally one would expect that the detection scheme to implement a cutoff in the value of $P_{\rm null}$, sources falling above this cutoff being discarded. The purpose of the function is to check that the calculation of $P_{\rm null}$ works as expected. This is done by comparing the column DETEC_PNULL values to the number $N_{\rm false}$ of false detections. In fact $N_{\rm false}$ is plotted as a function of DETEC_PNULL. How does this work? Well, the average number of false detections should be proportional to the probability of detection from background fluctuations - at least for the situation in which the 'real' sources are relatively sparse, and thus unable to muddy the waters. A histogram of $N_{\rm false}$ against $P_{\rm null}$ should therefore give a straight line of slope 1 on a log-log plot.

The following histogram columns are calculated:

- DPNUL_INT: DETEC_PNULL at the lower edge of the bin. This should be used as the x-value when plotting any 'cumulative' quantity on the y-axis.
- DPNUL_MID: DETEC_PNULL at the middle of the bin. This should be used as the x-value when plotting any 'differential' quantity on the y-axis.
- N_ALL: The total number of detected sources which fall within the DETEC_PNULL bin.
- N_ALL_ERR: Square root of N_ALL.
- N_ALL_INT: Cumulative value of N_ALL (ie, the sum of N_ALL in the present bin plus all those at lower values of DPNUL_MID).
- N_BAD: Number of unreliable detections. This comprises sources which have values of MATCH_PNULL which are larger than probcutoff. The numbers are divided by (1-probcutoff) to correct for an expected fraction of sources which are unreliable but nevertheless fell close enough to a simulated source to have MATCH_PNULL > probcutoff.
- N_BAD_ERR: Square root of N_BAD, divided by $\sqrt{}$(1-probcutoff).
- N_BAD_INT: Cumulative value of N_BAD.

The DETEC_PNULL bins are calculated such that they have equal widths in $\log_{10}$ space. DPNUL_MID is also the geometric mean of the bin boundaries, not the arithmetic mean.

3 Parameters

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

Parameter	Mand	Type	Default	Constraints
entrystage	no	string	merge	merge-bias-completeness-reliability

This allows the user to enter the **eimsimreduce** script at one of several places in its processing sequence.

finalstage	no	string	reliability	merge-bias-completeness-reliability

This allows the user to exit the **eimsimreduce** script at one of several places in its processing sequence.

simgensubdir	no	string	sim_generic	

The task writes non-observation-specific output to this directory.

srcspecset	no	dataset	srcspec.fits	

This is the name of a FITS dataset which contains specification of the source probability distributions and also band-related specifications. See the 'input files' sections of the **eimsimprep** and **eimsim** documents for a detailed description. Example files can be found in $SAS_DIR/lib/data/eimsimdata/ .

biashistobinsize	no	real	0.1	

The size (in log10(x)) of the X-axis of the histogram created by **eimsimbias**. This X axis is log10(S) where S is flux in CGS units.

comphistobinsize	no	real	0.1	

The size (in log10(x)) of the X-axis of the histogram created by **eimsimcompleteness**. This X axis is log10(S) where S is flux in CGS units.

relhistobinsize	no	real	0.1	

The size (in log10(x)) of the X-axis of the histogram created by **eimsimreliability**. This X axis is log10($P_{\rm null}$) where $P_{\rm null} = \ln(\texttt{DET\_ML})$.

probcutoff	no	real	0.05	

Detections which have values of `MATCH_PNULL` which are lower than this cutoff are (subject to some correction for statistical bias in their numbers) are considered for statistical purposes to be 'genuine' detections.

astest	no	bool	no	

If 'yes', no tasks are called.

4 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.

label *(error)*
explanation

label *(warning)*
explanation
corrective action: this is the corrective action

5 Input Files

1. A FITS template file, as described in item 1 of the 'input files' section of the documentation for tasks **eimsimprep** and **eimsim**.
2. For each of perhaps several streams and fields, FITS datasets, each of which contains a list of simulated sources. The structure of this file type is described in the 'output files' section of the **eimsim** documentation.
3. For each of perhaps several streams and fields, FITS datasets, each of which contains a list of detected sources The structure of this file type is described in the 'output files' section of the **eimsim** documentation.

6 Output Files

1. A FITS dataset which is a merger of all the lists of detected sources mentioned in item 3 of the 'input files' section.

2. A FITS dataset which contains histograms relating to biases in positions and fluxes (and their errors) of detected sources. The structure is described in section 2.2.

3. A FITS dataset which contains histograms relating to the completeness of the list of detected sources - ie, what fraction of the simulated sources were detected. The structure of this file is described in section 2.3.

4. A FITS dataset which contains histograms relating to reliability of the source detections - ie, what fraction of the detections can be ascribed to background fluctuations. The structure of this file is described in section 2.4.

References