



Test Report for the Herschel Source Extraction Module

Isaac Roseboom, University of Sussex

SPIRE-SUX-REP-003190

12th February 2009

Applicable Documents

AD1 SPIRE Data Processing Pipeline Module Requirements, SPIRE-ICS-DOC-002998, Issue 2.1 January 2009

Reference Documents

RD1 PACS Point Source Extraction Requirements, PICC-ME-SR-001, Issue 1.0 July 2007

Introduction

This document recounts efforts to run the source Extractor task in the HIPE environment on a variety of simulated input maps. Tests were performed on DP-SPIRE development build number 252. For the most part the simulated input maps were constructed from the simulated catalogues of Fernandez-Conde et al. (2008). Objects from these catalogues were placed into our simulated maps taking into account the PSF (a 2d Gaussian with FWHM=18.6, equivalent to the 250 micron SPIRE PSF) and Gaussian noise. At all times the map size used was 450x450 pixels or $\frac{1}{4}$ sq. deg. The module was tested via both the GUI interface and Jython scripts. As this testing was done isolated from the rest of the pipeline, images had to be imported into HIPE in a slightly unnatural way, using the `importImage` task for non-Herschel compliant products, and the `simpleFitsReader` for PACS and SPIRE data products.

Verification

The operation of the point source extraction module was checked against the requirements specified in **[AD1]**.

SRCEXT-FUN-010 The module shall operate on data from a single map product
SRCEXT-FUN-020 The module shall be able to deal with both SPIRE and PACS mapping data products
SRCEXT-FUN-030 The module shall take as an input an Image Product. The capability to process a Timeline Product is desirable.

The module will accept SPIRE, PACS and imported data products as `simpleImage()` variables as an input via both GUI and command line interaction. There is a minor bug that the product must be `Double2d` (not `Float2d`) and simply reading the file using `importImage()` and then passing it to the `sourceExtractor` task results in no output (and no error message). An extra line of code setting the image values to `Double2d` is required [SPR-5736]. There is no way to pass the module a Timeline product.

SRCEXT-FUN-040 The module shall be capable of identifying , and extracting photometric parameters for, point sources from a given band in the input map, conforming to the source detection threshold and other input parameters provided.

The module will identify sources and extract their photometric parameters accurately for input maps that are well suited to the DAOPHOT algorithm which is currently the only algorithm available.

SRCEXT-FUN-050 The module shall be capable of extracting photometric parameters for sources in a list provided as input (in a Source List Product).

The module presently does not accept a list of sources as an input., nor does there appear to be any option for this type of input in the GUI or command line options to the module [SPR-6002].

SRCEXT-FUN-060 The module shall take as parameters the following:

- a source detection threshold (in terms of S/N or Bayesian evidence)
- the band in which the detection is performed
- the band for which photometric parameters will be extracted
- sub-region of the map to be considered for processing around each pixel
- sub-region of the map to be processed, specified in RA, Dec

The module will take detection threshold, sub-regions of the map, and pixel regions as an input. The module currently doesn't take the SPIRE band as an input [SPR-6003], but does allow the parameters that define a band (FWHM, PRF) as inputs.

One annoying feature is that in the GUI interface if the pixel regions are not defined to the size of the input map both corners are set to 0.0,0.0 by default, meaning that none of the map is used in the source extraction [SPR-6007].

SRCEXT-FUN-070 The module shall take the point spread function as an optional parameter. Where no PSF is specified, a default PSF shall be used. Each instrument will have a different default.

While there is a option in the GUI interface to pass the PSF it appears that this is not taken into account in the source extraction. The module assumes the PSF to be a 2d Gaussian with a FWHM as given. It doesn't appear that there is any way presently to change this via interactions with the GUI or command line [SPR-6008].

SRCEXT-FUN-080 The module will be designed such that the exact algorithm for source extraction is pluggable.

Verified by inspection of the Javadoc

SRCEXT-FUN-090 The module shall produce a Source List Product as output

This product shall contain at least the following information:

- *Source position, including errors*
- *Estimated source flux, including error for detected band*
- *Estimated Source size, including errors (TBC)*
- *Background parameters (TBD)*
- *Source quality*
- *TBD*

The output source list product contains the sources positions and fluxes. When run on test images with no noise or crowding the fluxes and positions extracted match up reasonably to those in the map, although there is an unacceptably high error. Even

for no noise cases the fluxes appear to be at least 5% off the true values. For more complicated scenarios it fails, however this is not a fault of the module but the DAOPHOT algorithm. It does not give positional or flux errors, nor does it provide any of the extra information detailed in the requirements (i.e. background level, size, quality) [SPR-6009]

Validation

From using the sourceExtractor module on range of pseudo-realistic SPIRE maps it is clear that the DAOPHOT algorithm presently implemented in the module will not be sufficient for all but the shallowest images. For testing purposes the module was run on a series of test maps produced by injecting sources from the simulations of Fernandez-Conde et al. (2008) into artificial maps. Only the 250 micron (PSW) band was considered as any issues with the module should not be band dependent. A number of test maps were produced:

- Shallow ($S_{250} > 40$ mJy), with clustering (i.e. positions as per Fernandez-Conde)
- Deep ($S_{250} > 10$ mJy), no clustering (positions on Grid)
- Deep ($S_{250} > 10$ mJy), with clustering

For the first two situations with no noise the sourceExtractor module can reliably find all the sources in the map and recover their fluxes. There is an issue in that the errors in the recovered fluxes are unacceptably high. For a no noise case the source extraction algorithm should be able to get the fluxes right to within 1%, however even for sources on a grid with no noise the sourceExtractor module cannot get the fluxes right to within better than 5%. This may be due to some issues with the default PSF assumed by the module.

For any case with clustering and/or noise the sourceExtractor breaks down completely. It cannot be relied upon to find any sources except for the very brightest (> 100 mJy).

This is (I assume) because the DAOPHOT algorithm does not work well in crowded fields where it cannot find enough “blank” pixels around sources to determine the background level.

Notes on Integration with HIPE

One minor complaint is that the module is still very fragile and will only work under the right and slightly esoteric conditions. For instance running HIPE on my own laptop the module would not work at all, always returning the null reference as the source list object. However running on a colleague’s machine (which was exactly the same Macbook pro) and with exactly the same setup it seemed to work fine. This is more likely to be an issue with HIPE rather than the source extraction module.

The GUI interface is presently not very easy to use and fails somewhat in hiding the underlying object orientated nature of the software. While I understand that in a

typical scenario the operator of the software would have all of his data downloaded from ESA and placed into Variables in the Variable Menu automatically I don't think its is a good idea for this to be the only way to get data into the various modules. Simple drop down menus in the task GUIs which only include data they can recognise would be of much benefit and ease the operation of the software. Alternatively improving the variables menu to include the type of the variable as well would greatly enhance the usability of the software. Also the way to get external data into the source extraction module is totally non-obvious. Having something called the simpleFitsReader not able to read any fits image but only Herschel data is not a good situation. However none of these complaints are about the source extraction module itself, rather its interaction with the HIPE environment.