StarBugII Manual

JWST PSF photometry in dusty crowded fields. Last updated: v0.2.15

Installation

```
$~ pip install starbug2
```

After the package is installed, there are a few steps required to initialise Starbug.

WEBBPSF Is a dependency of Starbug that has its own initialisation process. The full installation is documented on webbpsf homepage however it requires two main steps. Download the data file on the website, named something like webbpsf-data-X.X.X.tar.gz and expand it into a directory, then append to your .bashrc (or equivalent) `export "WEBBPSF_PATH=PATH/TO/DIRECTORY"`.

StarbugII has a command that should initialise everything else. It will create the folder `${HOME}/.local/share/starbug` and download/generate relevant files. It will take approx. 5 minutes to complete.

```
$~ starbug2 --init

//verify it starts up without issue
$~ starbug2 -vh
```

TAB Completion

StarbugII has a bash completion script `starbug2/extras/starbug2.completion`. This can be installed directly into `/etc/bash_completion.d/` or `"source /PATH/TO/COMPLETION/FILE"` can be place within your .bashrc. Unfortunately this completion script works only in bash shells.

Usage

```
Starbug II - JWST PSF photometry
usage: starbug2 [-ABCDfhMPv] [-b bgdfile] [-d apfile] [-o directory] [-p file.param] [-s opt=val] image.fits ...
   -A  --apphot          : run aperture photometry on a source list
   -B  --background      : run background estimation
   -b  --bgdfile         : load background (-bgd.fits) file
   -C  --clean           : run source cleaning before photometry
   -d  --apfile  ap.fits : load a source detection (-ap.fits) file to skip the source detection step
   -D  --detect          : run source detection
   -f  --find            : attempt to find associated -ap -bgd files
   -h  --help            : display uasage information
   -M  --match           : match outputs from all input image files
   -n  --ncores      num : number of CPU cores to split process between
   -o  --output      dir : output directory
   -p  --param   a.param : load parameter file
   -P  --photom          : run psf photometry
   -s  --set      option : set value in parameter file at runtime (-s SIGSKY=3)
   -S  --subbgd          : subtract background from image
   -v  --verbose         : display verbose outputs

   --> Single run commands
       --init                    : Initialise Starbug (post install)
       --generate-psf            : Generate ALL the PSF files to "PSFDIR"
       --local-param             : Make a local copy of the default parameter file
       --generate-region   a.fits : Make a ds9 region file with a detection file
       --clean-table       a.fits : Clean up an individual table
       --generate-run      *.fits : Generate a simple run script
       --version                 : Print starbug2 version

   --> typical runs
      $~ starbug2 -vD -p file.param image.fits      //Source detect on image with a parameter file
      $~ starbug2 -vDM -n4 images*.fits             //Source detect and match outputs of a list of images
      $~ starbug2 -vd image-ap.fits -BP image.fits  //PSF photometry on an image with a source file (image-ap.fi
```

Parameter File

The parameter file is where any dataset specific parameters can be tweaked. Ideally the default values should be sufficient however if the diffuse dust emissions cause a very complex background or the field is very crowded, certain parameters may need tuned. Additionally differing sensitivities between photometric bands may require different parameters to detect on. To generate a local parameter file, run:

```
$~ starbug2 --local-param
```

This will create `./starbug.param`, a file which will be loaded by default when starbug is ran from the folder that contains it. However it can be named anything and explicitly loaded at runtime with `-p file.param` in the command. It may be the case that you keep several parameter files on hand to conduct slightly different routines, this is how you would load them.

For quick testing or tweaking of certain parameters, starbug can override a file setting with the addition of `-s PARAM=VALUE` OR `--set PARAM=VALUE` in the command. This will not change the parameter file but will use this parameter value instead.

Settable Parameters

As of the current version. If your parameter file doesnt fit the template of the current version of the default file, starbug will warn you but may crash later if you dont update the local parameter file.

NAME	DTYPE	DESCRIPTION
VERBOSE	INT 0:1	Include verbose outputs.
NULLVAL	FLOAT	(depr.) Output table NULL value.
PSFDIR	STR	Folder where package data is stored. This will expand environment variables (${HOME} -> /home/dlister).
OUTDIR	STR	Folder to output to.
SIGSKY	FLOAT >0	Number of sigma below which pixels gets removed as sky. In images with bright diffuse emissions, drop this value slowly, in steps of ~0.1 and watch the number of sources detected increase, but be careful of false detections of bright spots.
SIGSRC	FLOAT >0	Minimum number of sigma above the median that a source must be to be detected. This is often 5sigma for a robust search or 3sigma for faint detections.
BOX_SIZE	INT >1 [pix]	Kernel size during background subtraction to estimate background within. For complex fields this is set low.
FILTER_SIZE	INT >1 [pix]	(depr.)
DOBGD2D	INT 0:1	Run complex background subtraction during source detection (This usually results in more sources detected, however it takes a long time).
SHARP_LO	FLOAT >0	Set bounds of source sharpness, outside which sources will be ignored.
SHARP_HI	FLOAT >0	Set bounds of source sharpness, outside which sources will be ignored.
ROUND_LO	FLOAT >0	Set bounds of source roundness, outside which sources will be ignored.
ROUND_HI	FLOAT >0	Set bounds of source roundness, outside which sources will be ignored.
FIT_APP_R	INT 0:1	Fit fraction encircled energy to aperture radius and use this as aperture radii (1), OR, use explicit aperture radius (0).
ENCENERGY	FLOAT >0	Fraction encircled energy to fit aperture radius to.
APPHOT_R	FLOAT >0	Aperture photometry radius.
SKY_RIN	FLOAT >0	Sky annulus inner radius.
SKY_ROUT	FLOAT >0	Sky annulus outer radius.
ERROR_CUT	-	DESCRIPTION
SHARP_HI_SIG	-	DESCRIPTION
SHARP_LO_SIG	-	DESCRIPTION
ROUND_HI_SIG	-	DESCRIPTION
ROUND_LO_SIG	-	DESCRIPTION
BGD_R	-	DESCRIPTION
AP_FILE	STR	Load source list file (-ap.fits) into starbug. This is equivalent to `-d file-ap.fits`.
BGD_FILE	STR	Load background estimation file (-bgd.fits) into starbug. This is equivalent to `-b file-bgd.fits`.
CRIT_SEP	INT >0	DESCRIPTION

NAME	DTYPE	DESCRIPTION
MATCH_THRESH	FLOAT >0 [arcsec]	Separation threshold between coordinate during astrometric matching. Set low to avoid mismatching.
MATCH_COLS	STR,STR,..	Comma separated list of columns to include in outputs during matching.
RM_MATCH	INT	(prep.)
NUMBER_ARTIFICIAL_STARS		DESCRIPTION
SUBIMAGE_SIZE	-	DESCRIPTION
MIN_FLUX	-	DESCRIPTION
MAX_FLUX	-	DESCRIPTION
SEPARATION_THRESH		DESCRIPTION
REGION_COL	STR	DS9 region colour.
REGION_SCAL	INT 0:1	Scale region radius with flux?
REGION_RAD	FLOAT >0	If REGION_SCAL=0, what is the region radius.
REGION_XCOL	STR	Table column name for X coordinate of region.
REGION_YCOL	STR	Table column name for Y coordinate of region.
REGION_WCS	INT 0:1	If the REGION_(X/Y)COL values are WCS. If not it will set them as pixel coordinates.

A Typical Run

A typical run begins with a MAST download, I recommend using jwst_mast_query, and it is worth splitting the data by photometric filter. For each filter, it is worth creating a unique parameter file with:

```
$~ starbug2 --local-param
```

This will place `starbug.param` in the current folder. If this file exists in the folder at runtime, starbug will automatically load it. However it can be renamed to anything as long as it is explicitly loaded with `-p file.param` during any subsequent commands.

Source detection

Begin honing the source detection parameters by running starbug on a single exposure. This will be an iterative process of detecting and tweaking parameters. If the background of the image is fairly flat or uniform, the parameter `DOBGD2D` can be set to 0 to turn off the slower background subtraction pass during the routine.

```
$~ starbug2 -vD [-p param.file] exposure1.fits

//To create a DS9 region file to look at the resulting source file
$~ starbug2 --generate-region exposure1-ap.fits
$~ ds9 exposure1.fits -regions exposure1-ap.fits
```

If the resulting source list seems to miss obvious sources, begin slowly lowering `SIG_SRC` and `SIG_SKY`. The former is usually left at 5 for a robust detections, below this we begin to get into the uncertainty limit of the data, however in certain cases this may be desirable, although be careful dropping it below 3. It situations where the background contains more complex dust emissions, or is has some non-uniformity, `SIG_SKY` can be a productive parameter to lower. Drop it slowly, by no more than 0.1 at a time, and watch the output source file carefully. This will drastically increase the detection of fainter sources however it will begin to make false detections of bright dusty peaks.

To remove (a significant number of) spurious sources and or resolved background galaxies, the geometric parameters `SHARP_HI/LO` and `ROUND_HI/LO` are available. Sharpness is a measure of how the peak of the source compares to the median within the FWHM. Cosmic rays are often very sharp and can be limited by lowering the upper limit. Whereas other artifacts often appear less sharp. It is worth opening the output source list `*-ap.fits` and plotting the distribution of `SHARPNESS`, usually there is a clear roughly normal distribution with wings - set `SHARP_HI` and `SHARP_LO` to cut off these wings. Roundness is a measure of eccentricity of a source. The distribution should be symmetric (`ROUND_HI` and `ROUND_LO` and measures of the same thing in orthogonal directions). Sources centred on 0 are round or pointlike, whereas resolved galaxies are often towards the edges of the distribution.

Introducing Dithers

You should have now arrived on appropriate parameters but may still be finding that certain extraneous bad sources are squeaking through the cuts. The next method we can deploy to remove these is rolling in the set of dithers. We can detect simultaneously on a set of dithers and match the outputs together, removing any sources that don't appear a threshold number of times.

If you give a starbug instance a list of files to work on, it will by default run them in series and produce a separate source list for each. However we can parallelise and match with the `-n NUMBER_CORES` and `-M` options:

```
$~ starbug2 -vD -M -n 4 exposure1.fits exposure2.fits exposure3.fits exposure4.fits

//General bash wildcarding etc. can also be used
$~ starbug2 -vDMn4 exposure{1..4}.fits
$~ starbug2 -vDMn4 exposure*.fits
```

The matched output will export into two catalogues: `exposure(1234)-apfull.fits` and `exposure(1234)-apmatch.fits`, the former being the complete catalogue containing every column, the latter being a condensed form that cuts sources without a threshold "NUM" value.

Aperture Photometry

PSF Photometry

Source Flags

Sources are given quality flags at various points in starbug routines. These flags are applied as bitwise masks, so that the source retains information from all stages during data reduction.

BIT	NAME	DESCRIPTION
0x00	SRC_GOOD	Source OK
0x01	SRC_BAD	Source aperture contains a pixel marked as saturate or bad
0x02	SRC_JMP	Source aperture contains a pixel marked with a jump during integration (possible cosmic ray)
0x04	SRC_VAR	Source has an asymmetric flux distribution between matches (mean and median more than 5% different)

FAQ