

Instructions for the GALFIT-SExtractor Modeling Pipeline

Authors: Ian Tibbetts '15
Jianing Yang '18
Ryan Cole '15
Elizabeth McGrath

Version: 1.0 as of 4/14/2015

Acknowledgments: Colby College Astrophysics
Joel Primack (UCSC)
Greg Snyder (STScI)

This is a procedure for analyzing real and simulated galaxies using Source Extractor and GALFIT automatically. The command line examples are for 'real' images, i.e. appear as though taken by a telescope, which is indicated by the presence of the `-r` flag. Remove this option if working with high resolution simulation images. The pipeline involves [Preprocessing](#), [Modeling](#), [Summarizing](#), then [Plotting](#). There is also a list of [TODO](#) at the end, as well as some links on how to edit this document using [reStructuredText](#). Finally I have included examples of how to generate the python documentation for the .py files using [pydoc](#).

Preprocessing

This is only necessary to do when starting to analyze new data, for example a set of images in a folder called `/diskarrays/astro_array1/hydroart/`. Be sure to modify the code of the files referenced here to correctly work for the file naming scheme of your data, or modify the filenames to conform to the expectations of this code.

Duplicate data before modifying

- If you have enough disk space it is a good idea to first make a new directory and duplicate your data
- `mkdir /diskarrays/astro_array1/hydroartMod/`
- `cp -r /diskarrays/astro_array1/hydroart/* /diskarrays/astro_array1/hydroartMod/`

Update the fits headers

- Check the `sum_galfit` method of [sumSimUtility](#) to determine what info is expected in the fits header
 - GALAXYID, HALOID, AVAL, FILTER, and CAMERA will be parsed from the filename if not present in header
 - If the filename is not in the expected format, these will be set to -99.0 in the summary
 - SCALESIM is only for converting distances in high resolution images to kpc
 - REDSHIFT, MASS, SFR, and SSFR will be set to -99.0 in the summary if not present in header
- Modify [updateHeader](#) to populate any missing fields (clobber=True to overwrite)
- eg.
`python McGrathResearch/src/updateHeader.py run/images/160withMRPsimImages.txt sum`
to update mass, sfr, ssfr

Generate sigma images

- Use [genSigmaImages](#) to create sigma images for each of the images in your data

- If the GALFIT generated data is good enough, skip this and GALFIT will output the generated sigma images and include each of them in the results of [simUtility](#)

Modeling

Make lists of images

- Using `echo` and piping into `xargs ls` (below) prevents the overflow that can occur when using `ls` alone.
- `echo /diskarrays/astro_array1/hydroartMod/VELA[01][2345]*/*.[^01]*/images/*F160W*_`
- If regular expressions become too much work, the `>>` (below) can be used to repeatedly append convenient chunks of images.
- `echo /diskarrays/astro_array1/hydroartMod/VELA2[678]*/*.[^01]*/images/*F160W*_nois`

Run [simUtility](#)

- Create a directory `requiredFiles` and put in the files `default.nnw`, `sex.conv`, and `sex.param` inside
- Navigate to the directory containing the `requiredFiles` directory, eg. `cd /diskarrays/astro_array1/imtibbet/run/`
- Use the `screen` command to make a new terminal session that can later be sent to the background
- Run the modeling program

- **Candelized**

```
python /diskarrays/astro_array1/imtibbet/McGrathResearch/src/simUtility.py
```

- **Simulation**

```
python /diskarrays/astro_array1/imtibbet/McGrathResearch/src/simUtility.py
```

- Send the command line output to `stdout.txt` by adding `> stdout.txt` to the end of either so it can be searched later (e.g. `grep -i warning stdout.txt`)
 - For high resolution and/or many images, the `-p` flag is recommended to reduce runtime but causes multithreading that is hard to cancel/abort and requires manual cleanup using the `kill` command on the top level python process, after which all child processes should die as GALFIT finishes
 - Ctrl a, d to detach from screen, `screen -r` to resume
 - To see progress, either use `top` or `ps -ef` (running processes) or `ls _/*single*param* | wc -l` (number of images processed so far)
- | | |
|---------------|---|
| -h | show this help message and exit |
| -b | include to run a GALFIT bulge fit after the initial galaxy fit |
| -g | include to suppress running galfit after Source-Extractor |
| -p | include to run images in parallel, otherwise series |
| -r | include to run Source-Extractor for real images, otherwise sim images assumed |
| --mpz=MPZ | set the magnitude photometric zeropoint for GALFIT to use [default: 26.23] |
| --plate=Plate | set the plate scale for GALFIT to use [default: 0.06] |
| --psf=PSF | set the file for GALFIT to use as a PSF |

Summarizing

This step aggregates all of GALFIT's multiple extension cube outputs into a csv summary file by parsing the `pyfits` header. This is also the step where any features not outputted by GALFIT (e.g. RFF, radius(kpc), type, etc.) are calculated. RFF is computed over an elliptical region, which can be shown in a picture outputted (`ellipseImage`).

Make lists of results

- `echo /diskarrays/astro_array1/imtibbet/results/resultsNoiseMRP/results/*single*mul`

Run **sumSimUtility**

- Note that this takes a while depending on the number of images and if you want to see the progress include `-v` on the command line

- `python McGrathResearch/src/sumSimUtility.py results/resultsNoiseMRP/singleMultiFit`

-h show this help message and exit

-b if running on results of GALFIT bulge (two component) fit

-d DELIM set the delimiter to separate the fields of the summary file [default:]

-r if running on candelized images

-o OUTPUT set the filename to write the output summary file [default: summary_04-14-2015.csv]

-v to enable command line printouts of state

Plotting

Run `plotSimUtility`

- `python McGrathResearch/src/plotSimUtility.py summaries/summaryNoise.csv -r -d , -p`

-h show this help message and exit

-m to plot MRP counterparts adjacent

-r to indicate candelized results are being plotted

-n the space separated list of galaxy names to be plotted (must exist in summary file). The default is all unique galaxy ids plotted separately

-y YFIELDNAMES the space separated list of y field names to be plotted, available options are: ['age', 'red', 'px', 'epx', 'py', 'epy', 'mag', 'emag', 'rpix', 'erpix', 'rad', 'erad', 'ser', 'eser', 'ba', 'eba', 'pa', 'epa', 'rff'], default: ser

-x XFIELDNAME the field name of the x values, available options are: ['age', 'red', 'px', 'epx', 'py', 'epy', 'mag', 'emag', 'rpix', 'erpix', 'rad', 'erad', 'ser', 'eser', 'ba', 'eba', 'pa', 'epa', 'rff'], default: red

-t COMPTYPE the type of component to be plotted (central, bulge, disk), default: central

-p PLOTTYPE the type of plot, available options are: ['default', 'allGalaxies', 'allFields', 'bulgeToTotal', 'mozena', 'barro', 'special', 'vivian'], default: default

-c CAMERAS specify specific cameras (e.g. 0 or 1 or... or all)

-d DELIMITER specify delimiter of data summary file, default whitespace

reStructuredText

- <https://github.com/ralsina/rst-cheatsheet/blob/master/rst-cheatsheet.rst>
- <https://gist.github.com/dupuy/1855764>

TODO

- Test RFF in `sumSimUtility`

- Summing over GALFIT, what about Source Extractor radius? What if the sex.param doesn't have FLUX_RADIUS? Should we make that required like X_IMAGE and Y_IMAGE?
- Sigma Images for [simUtility](#)
 - High resolution, using sqrt now, should be same as GALFIT generated, could be ones.fits?
- Required Files Directory for [simUtility](#)
 - Do we need to have a user populated folder at all?
 - sex.conv and default.nnw can be generated, where to save them?
 - sex.param can be generated or specified on the command line
 - config.sex can be generated or specified on the command line
 - one or the other, not both because the config.sex points to the sex.param (PARAMETERS_NAME)
- Source Extractor in [simUtility](#)
 - values for candelized and high resolution: min area, threshold (detect, deblend, and analysis), min contrast, clean bool and param
 - number of galaxies, identifying clumps as separate
 - dynamically changing values to work for different time periods/clumpyness/brightness?
- Barro plot in [plotSimUtility](#)
- High Res Vivian plots in [plotSimUtility](#)
- Bulge/Total in [plotSimUtility](#)
 - Can we make this a column of the csv in [sumSimUtility](#) (-99)
- Pyfits in [simUtility](#)
 - currently using IRAF to get image size because Pyfits freezes the processes in parallel
- Fixing [dashboard](#)
 - GUI for running the pipeline, finicky/need to be able to close easily
 - How could screen work with this?
- Interactive Plotting using [display](#)
 - Tune the logarithmic scaling of cube images (right double click)
 - Button on image display dialog for displaying the fits header

Pydoc

To view documentation

- `pydoc /diskarrays/astro_array1/mtibbet/McGrathResearch/src/*.py`
- `pydoc /diskarrays/astro_array1/mtibbet/McGrathResearch/src/sumSimUtility.sum_galf`

To generate .html files

- `pydoc -h <same as above>`