

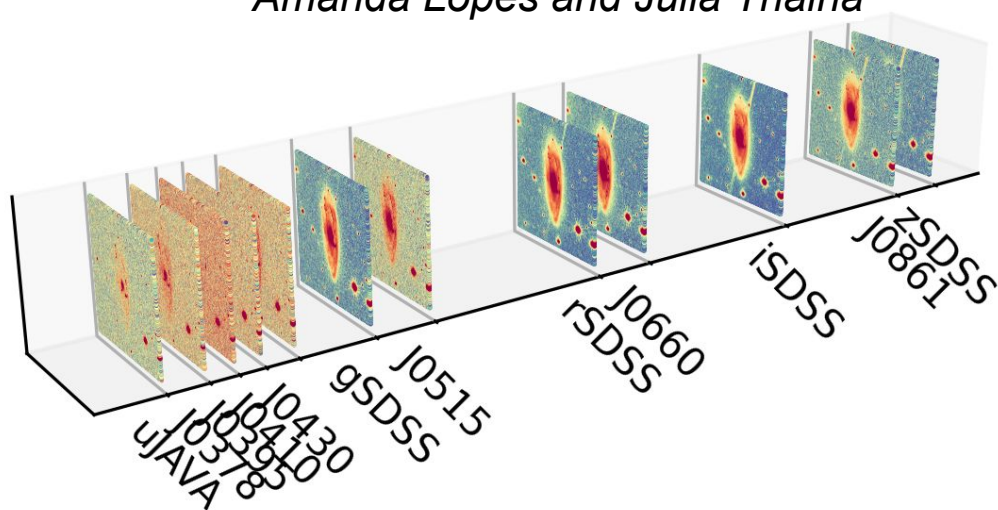
S-Cubes python package

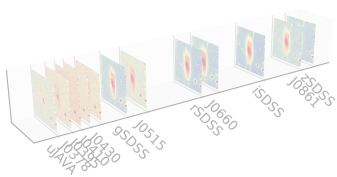
creating IFS-like cubes with S-PLUS data

Eduardo A. D. Lacerda

coding: Fábio Herpich, Gustavo Schwarz

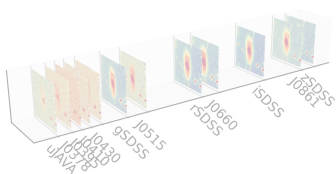
testing: Roberto Cid Fernandes, Maiara Sampaio Carvalho, Victor Hugo Sasse,
Amanda Lopes and Júlia Thainá





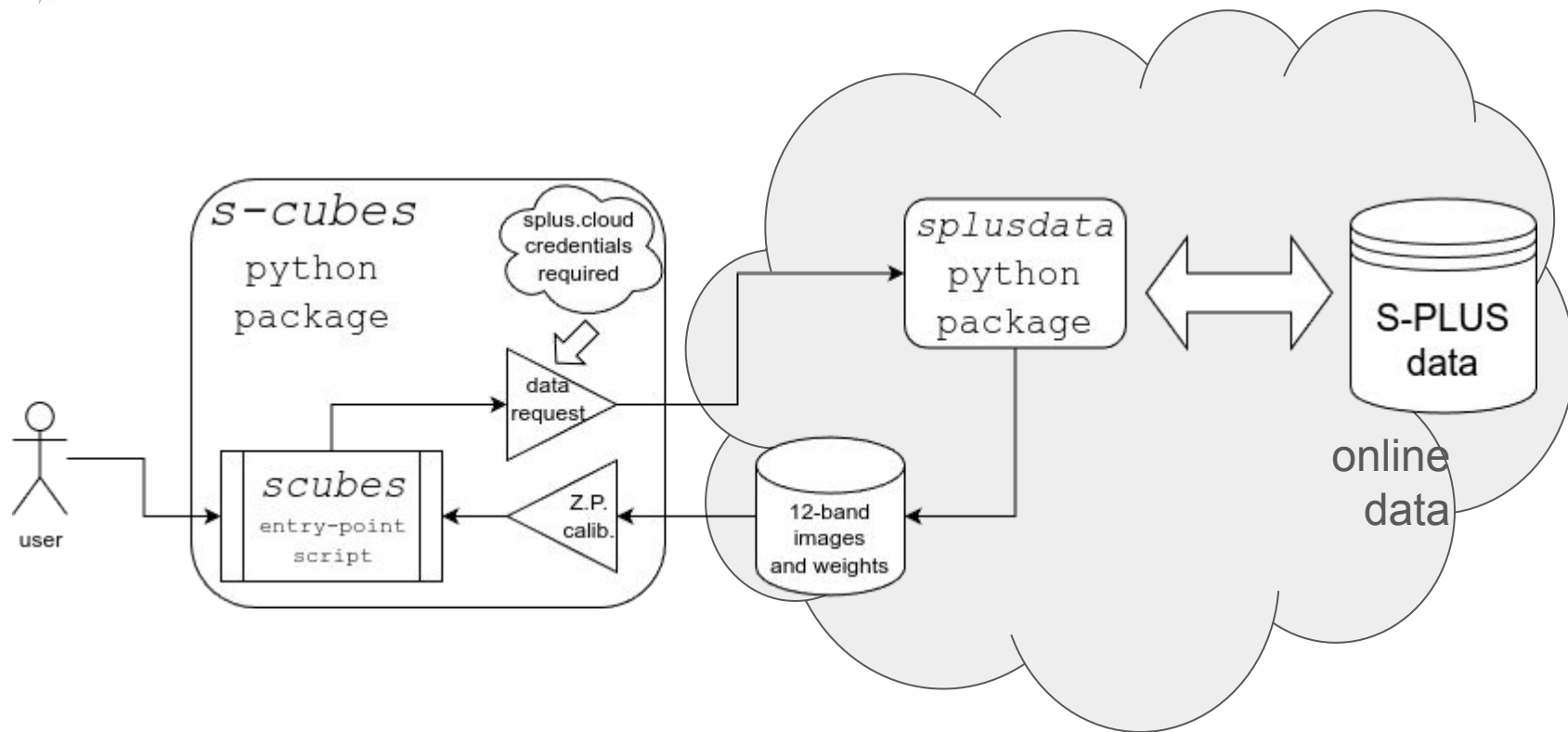
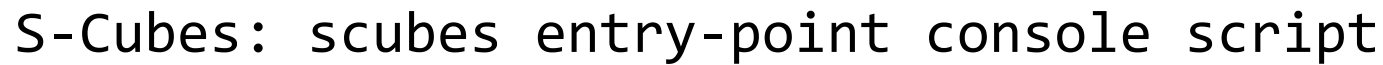
S-Cubes package presentation

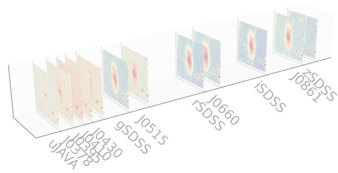
The software is a modern version of ***make_scubes_v02.py***, a script written by Fábio Herpich. **S-Cubes** has been completely remodeled and rewritten taking advantage of a new paradigm of coding, new python tools, and the recent release of the **sp1usdata** package.



S-Cubes package main console scripts

- *scubes* and *scubesml*: the main script of **S-Cubes**. Download 12-band images cropped from S-PLUS observed tiles. The images are zero-point calibrated and the fluxes and uncertainties are calculated. At the end, generates a FITS file with the cube and some metadata. The “*ml*” version operates with an input ***masterlist*** of objects.
- *sex_mask_stars*: uses **SExtractor** in order to create a spatial mask of stars, attempting to remove the areas enclosed by the brightest ones along the FOV.

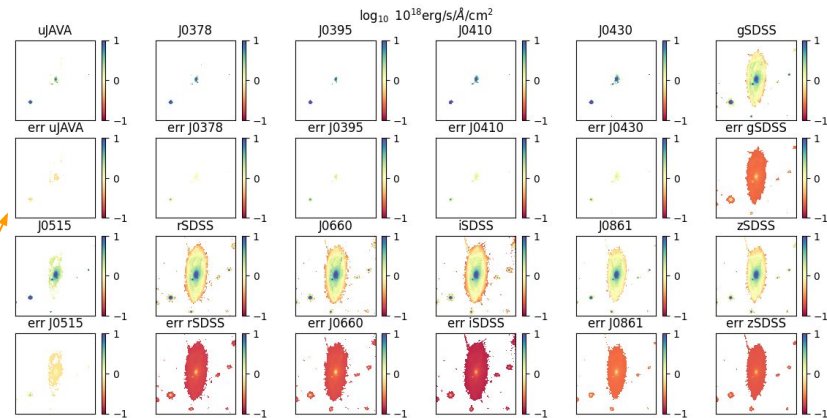




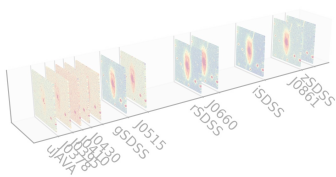
S-Cubes: SCUBE - the output cube

scubes
FITS file

scubes FITS	
0 - PRIMARY	<u>Object info @ HEADER</u>
1 - DATA	12-band 3D spectra (x, y, bands)
2 - ERRORS	12-band 3D error spectra (x, y, bands)
3 - WEIMASK	Weights mask (x, y)
4 - METADATA	filters, central wavelenghts, pivot wavelenghts, exposure times, gains, PSF FWHM, obs. date.



```
FITS_rec([('uJ0378', 3576.5900319, 3533.28150603, 1302.99580147, 2651.74575679, 1.4238565, '2017-02-19'),
('J0378', 3770.66765668, 3773.16495619, 1266.43687077, 2590.53003934, 1.24601953, '2017-02-19'),
('J0395', 3940.66900669, 3940.69812172, 680.64843148, 1401.35102734, 1.18031096, '2017-02-19'),
('J0410', 4094.0795908, 4094.92800733, 345.42806559, 706.83367717, 1.13351701, '2017-02-19'),
('J0430', 4292.0201202, 4292.10579006, 278.90845694, 570.57617371, 1.14671147, '2017-02-19'),
('gSDSS', 4774.02604026, 4758.4878587, 191.4341835, 370.91141421, 1.2285185, '2017-02-19'),
('J0515', 5132.82097321, 5133.13247975, 299.46581613, 610.78047227, 1.10798045, '2017-02-19'),
('rSDSS', 6274.74334743, 6251.83097429, 195.74024342, 396.72506048, 1.09843247, '2017-02-19'),
('J0660', 6613.99318993, 6613.87556039, 1430.33833785, 2904.99319306, 1.07404553, '2017-02-19'),
('iSDSS', 7702.49932499, 7670.61445983, 272.39806574, 560.98501139, 1.03689299, '2017-02-19'),
('J0861', 8611.48166482, 8607.25421702, 479.32598833, 984.36649508, 1.083632, '2017-02-19'),
('zSDSS', 8881.70071701, 8941.47606623, 275.62079939, 566.9703096, 1.03092001, '2017-02-19')],
dtype=numpy.record, [('FILTER', '<S5'), ('CENTWAVE', '>f8'), ('PIVOTWAVE', '>f8'), ('EXPTIME', '>f8')])
```

S-Cubes package source and online documentation

- Webpage: https://elacerda.github.io/s-cubes/nb_example.html

S-Cubes

Search docs

DOCUMENTATION

- Getting started
- How to create a cube
- Entry-point console scripts

RUNNING EXAMPLES

- ☒ Jupyter Notebook example
 - Creating a cube
 - How to read a cube
 - Headers
 - Filters information
 - Images plot
 - 3D image
 - RGB and Filters plot
 - Distance from center
- Mask stars with scubes package
- Masterlist run example

PACKAGE

- scubes

interactive
scubes package
usage with plots

/ Jupyter Notebook example

[View page source](#)

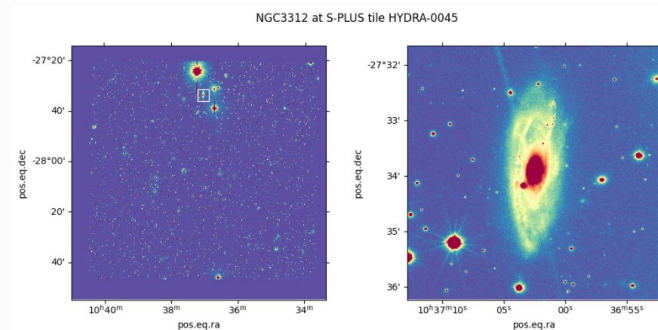
Jupyter Notebook example

[Download this notebook.](#)

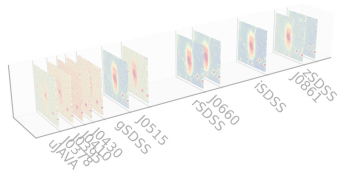
download .ipynb file

Creating a cube

This example will create a 500x500 pixels cube with the 12-bands images from S-PLUS TILE HYDRA-0045 for the NGC3312 galaxy. The stamps are made from a cropped 500x500 pixels area located at S-PLUS TILE mentioned before, centered at coordinates RA 10h37m02.5s and DEC -27d33'56".

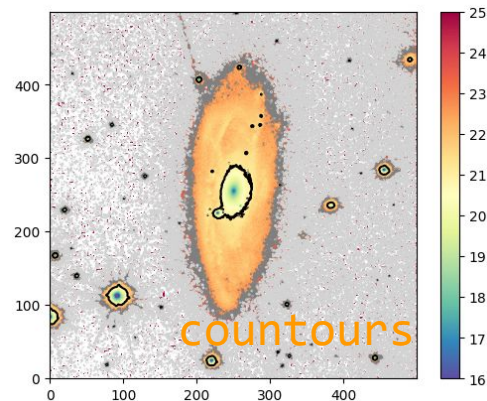
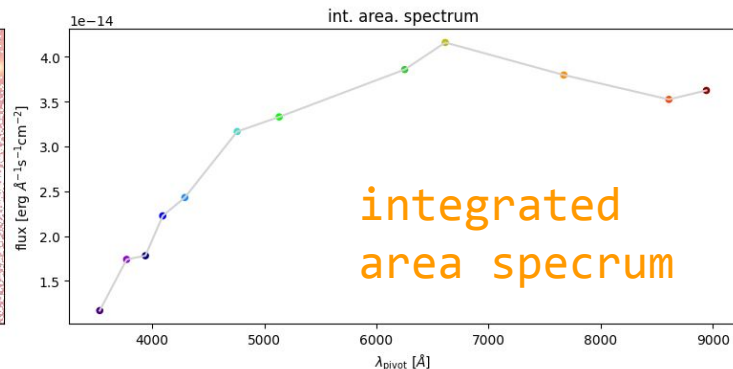
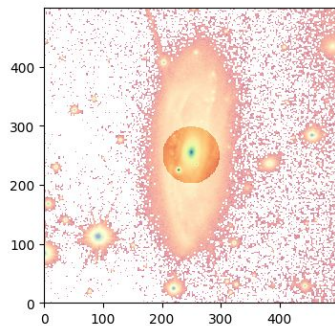
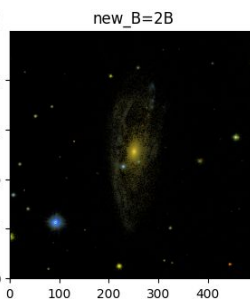
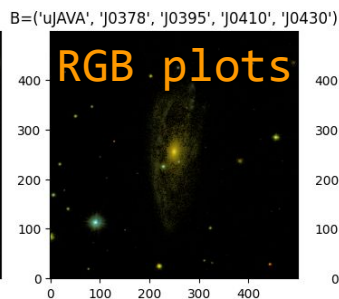
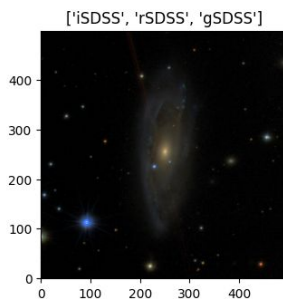
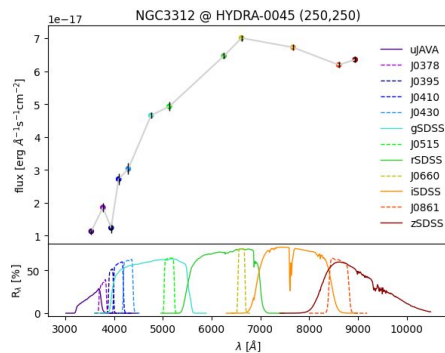
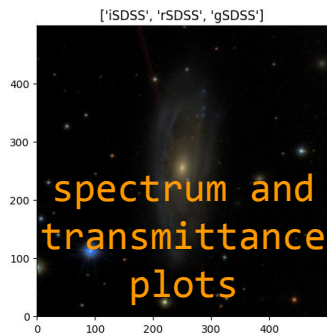


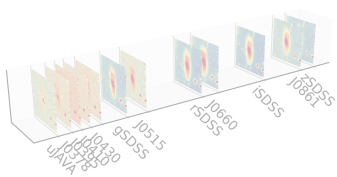
NGC3312 crop at HYDRA-0045 S-PLUS tile



S-Cubes package source and online documentation

- Webpage: https://elacerda.github.io/s-cubes/nb_example.html





S-Cubes package source and online documentation

- Webpage: https://elacerda.github.io/s-cubes/masterlist_example.html

How to run scubes
using a list of
objects

S-Cubes

Search docs

DOCUMENTATION

- Getting started
- How to create a cube
- Entry-point console scripts

RUNNING EXAMPLES

- Jupyter Notebook example
- Mask stars with `scubes` package
- Masterlist run example**
 - Size of the stamp
 - Header information
 - Running scubes for the entire list

PACKAGE

scubes

Masterlist run example

Download this notebook.

`scubes` package implements a script to run the program from a CSV file with a list of objects, a *Masterlist*. This script is called `scubesml`.

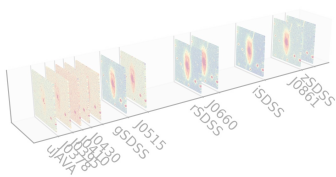
Masterlist is a csv text file in which one could gather information of a list of objects to create cubes. The file must contain at least 5 columns named with the following header and information:

1. `SNAME` : A nickname for the object
2. `FIELD` : S-PLUS Field (TILE) in which the program will search for the coordinates
3. `RA_deg` : Right-ascension in degrees
4. `DEC_deg` : Declination in degrees
5. `SIZE_pix` : SIZE of the object in pixels

Masterlist file content example:

```
SNAME,FIELD,RA_deg,DEC_deg,SIZE_pix S00001,SPLUS-
s24s34,52.08196,-31.06817,53.65902 S00002,SPLUS-
s24s35,52.87771,-30.21333,25.898617 (...)
```

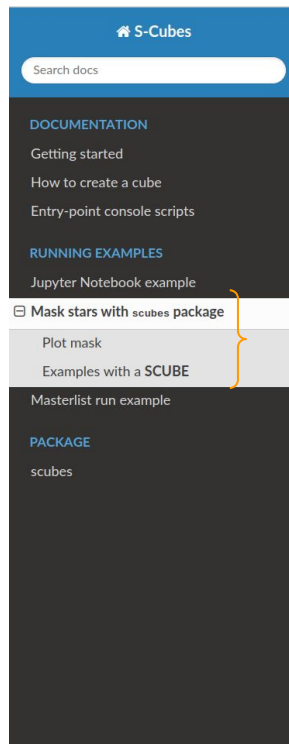
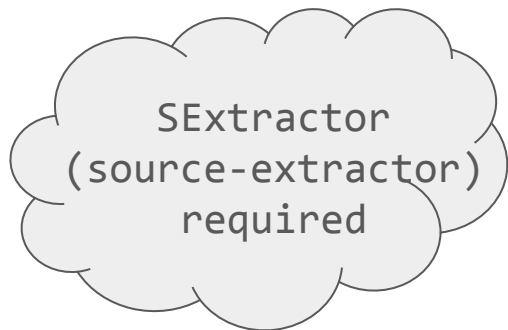
A *Masterlist* could contain more columns and, at the end of the run, the script will update the primary header of the output FITS file with all information inside the *Masterlist* for the chosen object.



S-Cubes package source and online documentation

- Webpage: https://elacerda.github.io/s-cubes/mask_stars_example.html

Using scubes package to
create a mask of stars in
the FOV



Mask stars with scubes package

[View page source](#)

Mask stars with `scubes` package

[Download this notebook.](#)

`scubes` package implements some ways to create a star mask for a selected stamp. The program uses `source-extractor` to create the mask and depends on a series of dedicated options.

Before the `SCUBE` created, one can choose two ways to create a stars mask:

1. `scubes` script has the option to include the source extractor (`-mask_stars` option);
2. `sex_mask_stars` script: Another script implemented by `scubes` package which creates a stars mask either using RA, DEC, TILE and SIZE options.

After the `SCUBE` generated, one can use two other options to create a stars mask:

1. `sex_mask_stars_cube` script, or;
2. reading a cube with `scubes.utilities.read_scube` class and using the method `source_extractor()`.

The console scripts (`scubes`, `sex_mask_stars` and `sex_mask_stars_cube`) allow the user to interact with the mask process. By using the option `--no_interact` the user can turn off the interaction with the mask process (in order to create non-assisted runs). At the end the script generates a 2D mask FITS file, together with the S-PLUS data detection image and some `source-extractor` metadata files (2d segmentation fits, catalogs and logs). Also, a image of the detection is created. Here we show an example using `sex_mask_stars` script without interaction:

