ObsAstro

Using Jupyter Notebook for a project feels a bit odd; it often resembles more like a playful learning tool, with a relaxed arrangement of code and variables scattering throughout the notebook, which complicates the task of gathering information effectively. Moreover, completing an academic task merely by modifying input filenames and clicking through the steps offers little educational value, serving only to secure three credit points. Therefore, I used the `demo.ipynb` as a reference to extensively refactor the code snippets into a typical Python project. which has undoubtedly been a more educational procedure.

TL; DR

Execute `pipe.sh` to validate the output file set below.

```
chmod +x pipe.sh && ./pipe.sh
```

```
fits
 ├──── figures
 │    ├──── 10-image-griz-bound.png
 │    ├──── 10-image-griz-color.png
 │    ├──── 10-image-hist.png
 │    ├──── 10-image-irg-bound.png
 │    ├──── 10-image-irg-color.png
 │    └──── 10-image-plot.png
 ├──── isophote
 │    ├──── 10-grad-r.png
 │    ├──── 10-gradiff-r.png
 │    ├──── 10-isophote-r.png
 │    ├──── 10-masked-r.png
 │    └──── 10-profile-r.png
 └──── source
      ├──── 10-aperture-r.png
      ├──── 10-background-r.png
      ├──── 10-deblend-r.png
      ├──── aperture-r.png
      ├──── background-r.png
      └──── deblend-r.png
```

Alternatively, you can inspect cherry-picked results in the `figures` directory.

```
figures
 ├──── 10-background-r.png
 ├──── 10-deblend-r.png
 ├──── 10-grad-r.png
 ├──── 10-gradiff-r.png
 ├──── 10-image-griz-color.png
 ├──── 10-image-irg-bound.png
 ├──── 10-image-plot.png
 ├──── 10-isophote-r.png
 ├──── 10-profile-r.png
 └──── aperture-r.png
```

Structure

The project consists of five main components named `sync.py`, `plot.py`, `utils.py`, `source.py`, and `isophote.py`. Each module's name reflects its respective functionality. Parallel execution is implemented where it is straightforward.

- `requirements.txt`

Provides a list of dependencies required to run the project.

It's recommended to utilize `venv` for environment management.

```
python3 -m venv obsastro
source obsastro/bin/activate
pip3 install -r requirements.txt
```

- `sync.py`

This module includes utilities to synchronize files from the legacy survey by downloading all `coadd` data files along with the `tractor` table files. Perform a sha256sum check to ensure the downloaded files are intact.

- Retrieving a list of image data. (`sync`)
- Downloading the file and verifying its checksum. (`download`)

- `plot.py`

This module generates basic images for understanding and estimation of images.

- Plotting the image and its corresponding data histogram. (`plot`, `hist`)
- Generating RGB image by stacking data from multi-wavelength image sources. (`color`)
- Mapping g, r, i, z bands into r, g, b color space using mapping function that mimics Legacy Survey Viewer. (`color`, `dr_rgb`)

- `utils.py`

This module comprises utilities that serve various common purposes and help with the processing of data files.

- Mathematical tools. (`magnitude`, `to_pix`)
- Merging adjacent `.fits.fz` image files as well as `.fits` tractor tables. (`merge`)
- Reprojecting the image around a specified central point and scaling it, while also filtering the associated tractor table to retain entries located within the cropped image. (`crop`)
- Visualizing images with `**kwargs`, using a projection from the world coordinate system, and opting to display on the front or export as image files. (`axis`, `plot`, `finalize`)

Q: *What should you do when there are problematic data (e.g.,* `NaN`*,* `Inf`*) on your image?*

A: *Maintain them as they signify a specific purpose. Utilize functions equipped to handle this data, such as* `np.nanpercentile` *rather than* `np.percentile`*.*

Q: *See if you can save the cutout image as a new* `FITS` *file.*

A: *Implemented methods (*`merge`*,* `crop`*) that not only save but also merge and crop, covering image data as well as* `tractor` *tables.*

- `source.py`

The module presents fundamental techniques for aperture computation.

- Background detection and elimination. (`background`)
- Aperture computing using sky circular approach, comparing results with the tractor table. (`aperture`)

- Seperating objects using source detection algorithms, followed by a deblending process. (`deblend`)
- Visualizing aperture curve and other findings. (`main`)

***Q**: Now that we know the background estimation is affected by bright sources, can you think of a way to improve it?*

***A**: An approach (implemented within `try` blocks) is implemented to mask these sources before estimation. This involves identifying bright sources, creating a mask, and applying it during the background estimation process.*

***Q**: Why do you think some of the point sources have fainter PSF magnitude than the aperture one? Shouldn't the PSF model recover the total flux better than the aperture ones?*

***A**: Several factors may influence this, including limitations of the PSF model accuracy, background calculation methods, dense stellar fields, and technical constraints of instruments. While theoretically the PSF model should prevail, in practice, it depends on the tractor's black box code.*

***Q**: Can you perform this in two of the g,r,z bands, and make a similar comparisons but for color?*

***A**: Simply execute the properly configured script twice, providing the appropriate command line arguments.*

`- isophote.py`

This component explores advanced methods for aperture calculation.

- Identifying and masking sources adjacent to the main object. (`mask`)
- Computing isophotes through elliptical geometries. (`isophote`)
- Visualizing cumulative aperture curve and other findings. (`main`)

***Q**: Why the surface brightness profile is "bumpy"? Can you try to fix it?*

***A**: Bright sources other than the selected object exists. Mask out these sources before fitting the image.*

***Q**: (Moreover) How to fit non-ellipse objects?*

***A**: Generate variation curves in different directions, determine suitable thresholds and isophotes, and interpolate the shape curves. (`10-grad-r.png`, `10-gradiff-r.png`)*