
boulliau Documentation

Release 0.0.dev028

Brett M. Morris

Nov 26, 2018

Contents

I	boulliau Documentation	3
1	Reference/API	7
	Python Module Index	15

This is the documentation for boulliau. Open source python toolkit for differential photometry with the ARCSAT 0.5 m Telescope at Apache Point Observatory.

Part I

boulliau Documentation

This is the documentation for boulliau.

1.1 boulliau Package

1.1.1 Functions

`force_photometry(image_paths, archive_path)` Perform forced photometry on a series of images.

`generate_master_dark(dark_paths, ...)` Create a master flat from night-sky flats, and a master dark.

`generate_master_flat_and_dark(flat_paths, ...)` Create a master flat from night-sky flats, and a master dark.

`init_centroids(first_image, target_centroid)`

`photometry(image_paths, master_dark_path, ...)`

Parameters

`regression_coeffs(X, Y, errors)` Compute linear regression coefficients for basis vectors in the columns of matrix X.

`regression_model(C, X)` Compute linear regression model given the regression coefficients.

`test(**kwargs)` Run the tests for the package.

PCA_light_curve

```
bouliaou.PCA_light_curve(pr, transit_parameters, buffer_time=<Quantity 5.
min>, outlier_mad_std_factor=3.0, plots=False, validation_duration_fraction=0.16666666666666666, flux_threshold=0.89,
validation_time=-0.65, plot_validation=False, outlier_rejection=True)
```

Parameters

pr
[PhotometryResults]

transit_parameters
[TransitParams]

buffer_time
[Quantity]

outlier_mad_std_factor
[float]

plots
[bool]

validation_duration_fraction
[float]

Returns

best_lc
[ndarray]

force_photometry

```
bouliaou.force_photometry(image_paths, archive_path)
```

Perform forced photometry on a series of images.

Will shift all images so that the brightest star is in the center, then do photometry on the next N brightest stars

Parameters

image_paths
[str] String fed to `glob` to pick up FITS image paths.

archive_path
[str] Path to an HDF5 archive of the images that will be created by this method

Returns

pr
[PhotometryResults] Results of the forced photometry

generate_master_dark

```
bouliaou.generate_master_dark(dark_paths, master_dark_path)
```

Create a master flat from night-sky flats, and a master dark.

Parameters

dark_paths

[list] List of paths to dark frames

master_dark_path

[str] Path to master dark frame that will be created

generate_master_flat_and_dark

`boulliau.generate_master_flat_and_dark(flat_paths, dark_paths, master_flat_path, master_dark_path)`

Create a master flat from night-sky flats, and a master dark.

Parameters

flat_paths

[list] List of paths to flat fields

dark_paths

[list] List of paths to dark frames

master_flat_path

[str] Path to master flat that will be created

master_dark_path

[str] Path to master dark frame that will be created

init_centroids

`boulliau.init_centroids(first_image, target_centroid, min_flux=0.2, plots=False)`

photometry

`boulliau.photometry(image_paths, master_dark_path, master_flat_path, star_positions, aperture_radii, centroid_stamp_half_width, psf_stddev_init, aperture_annulus_radius)`

Parameters

master_dark_path

[str] Path to master dark frame

master_flat_path :str

Path to master flat field

target_centroid

[ndarray] position of centroid, with shape (2, 1)

comparison_flux_threshold

[float] Minimum fraction of the target star flux required to accept for a comparison star to be included

aperture_radii

[ndarray] Range of aperture radii to use

centroid_stamp_half_width

[int] Centroiding is done within image stamps centered on the stars. This parameter sets the half-width of the image stamps.

psf_stddev_init

[float] Initial guess for the width of the PSF stddev parameter, used for fitting 2D Gaussian kernels to the target star's PSF.

aperture_annulus_radius

[int] For each aperture in `aperture_radii`, measure the background in an annulus `aperture_annulus_radius` pixels bigger than the aperture radius

regression_coeffs

`boulliau.regression_coeffs(X, Y, errors)`

Compute linear regression coefficients for basis vectors in the columns of matrix X.

Parameters

X

[ndarray] Design matrix of regressors

Y

[ndarray] Data to fit X to

errors

[ndarray] Uncertainties on each element of Y

Returns

C

[ndarray] Regression coefficients for each column of X

regression_model

`boulliau.regression_model(C, X)`

Compute linear regression model given the regression coefficients.

Parameters

C

[ndarray] Regression coefficients for each column of X

X

[ndarray] Design matrix of regressors

Returns

M

[ndarray]

test

`boulliau.test(**kwargs)`

Run the tests for the package.

This method builds arguments for and then calls `pytest.main`.

Parameters

package

[str, optional] The name of a specific package to test, e.g. 'io.fits' or 'utils'. Accepts comma separated string to specify multiple packages. If nothing is specified all default tests are run.

args

[str, optional] Additional arguments to be passed to `pytest.main` in the `args` keyword argument.

docs_path

[str, optional] The path to the documentation .rst files.

open_files

[bool, optional] Fail when any tests leave files open. Off by default, because this adds extra run time to the test suite. Requires the `psutil` package.

parallel

[int or 'auto', optional] When provided, run the tests in parallel on the specified number of CPUs. If parallel is 'auto', it will use the all the cores on the machine. Requires the `pytest-xdist` plugin.

pastebin

[('failed', 'all', None), optional] Convenience option for turning on `py.test` pastebin output. Set to 'failed' to upload info for failed tests, or 'all' to upload info for all tests.

pdb

[bool, optional] Turn on PDB post-mortem analysis for failing tests. Same as specifying `--pdb` in `args`.

pep8

[bool, optional] Turn on PEP8 checking via the `pytest-pep8` plugin and disable normal tests. Same as specifying `--pep8 -k pep8` in `args`.

plugins

[list, optional] Plugins to be passed to `pytest.main` in the `plugins` keyword argument.

remote_data

[{'none', 'astropy', 'any'}, optional] Controls whether to run tests marked with `@pytest.mark.remote_data`. This can be set to run no tests with remote data (none), only ones that use data from <http://data.astropy.org> (astropy), or all tests that use remote data (any). The default is none.

repeat

[int, optional] If set, specifies how many times each test should be run. This is useful for diagnosing sporadic failures.

skip_docs

[bool, optional] When `True`, skips running the doctests in the .rst files.

test_path

[str, optional] Specify location to test by path. May be a single file or directory. Must be specified absolutely or relative to the calling directory.

verbose

[bool, optional] Convenience option to turn on verbose output from py.test. Passing True is the same as specifying -v in args.

1.1.2 Classes

`PhotometryResults(times, fluxes[, errors, ...])`

`UnsupportedPythonError`

PhotometryResults

class `boulIAU.PhotometryResults`(*times, fluxes, errors=None, xcentroids=None, ycentroids=None, air-mass=None, background_median=None, psf_stddev=None, aperture_radii=None*)

Bases: `object`

Methods Summary

<code>load(path)</code>	Load photometry results from an output file.
<code>save(path)</code>	Save photometry results.

Methods Documentation

classmethod `load(path)`

Load photometry results from an output file.

Parameters

path

[str] Path to results to load.

save(path)

Save photometry results.

Parameters

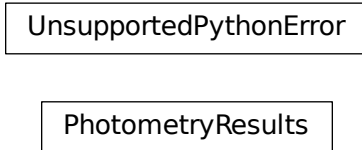
path

[str] Path to results to save.

UnsupportedPythonError

exception `boulIAU.UnsupportedPythonError`

1.1.3 Class Inheritance Diagram



Note: The layout of this directory is simply a suggestion. To follow traditional practice, do *not* edit this page, but instead place all documentation for the package inside `boulliau/`. You can follow this practice or choose your own layout.

b

boulliau, [7](#)

B

[boulliau \(module\)](#), [7](#)

F

[force_photometry\(\)](#) (in module [boulliau](#)), [8](#)

G

[generate_master_dark\(\)](#) (in module [boulliau](#)), [8](#)

[generate_master_flat_and_dark\(\)](#) (in module [boulliau](#)), [9](#)

I

[init_centroids\(\)](#) (in module [boulliau](#)), [9](#)

L

[load\(\)](#) ([boulliau.PhotometryResults](#) class method), [12](#)

P

[PCA_light_curve\(\)](#) (in module [boulliau](#)), [8](#)

[photometry\(\)](#) (in module [boulliau](#)), [9](#)

[PhotometryResults](#) (class in [boulliau](#)), [12](#)

R

[regression_coeffs\(\)](#) (in module [boulliau](#)), [10](#)

[regression_model\(\)](#) (in module [boulliau](#)), [10](#)

S

[save\(\)](#) ([boulliau.PhotometryResults](#) method), [12](#)

T

[test\(\)](#) (in module [boulliau](#)), [11](#)

U

[UnsupportedPythonError](#), [12](#)