lightcurve Documentation

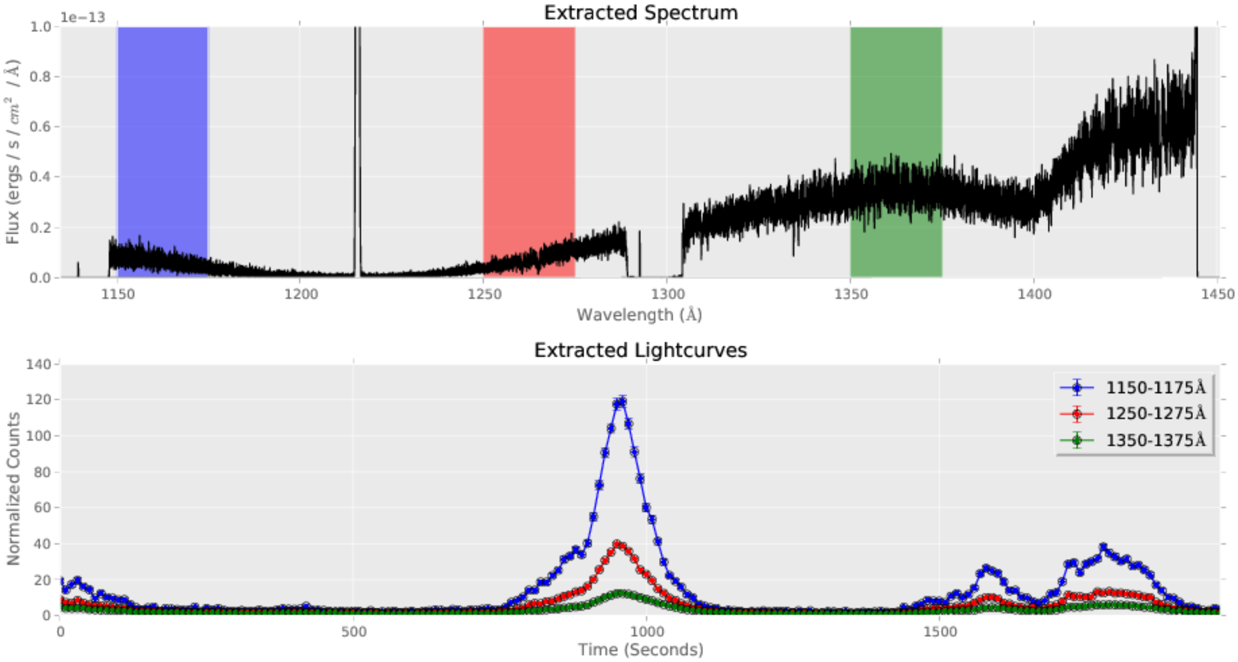
Release 0.6.0

Justin Ely

Aug 27, 2017

Contents

Hello, and welcome to the lightcurve documentation. So far it's pretty useless, but hopefully it makes you feel better that it at least exists.



CHAPTER 1

Contents:

1.1 Installation instructions

1.1.1 Install via Anaconda

Note: If you do not have Anaconda, please follow the instructions here to install it, or scroll down for manual installation of lightcurve.

After you have anaconda setup, then you can install lightcurve by specifying the channel in your install command:

```
$ conda install --channel justincely lightcurve
```

1.1.2 Install vi pip

If you have pip installed:

```
$ pip install lightcurve
```

1.1.3 Install from source

Refstis can also be installed manually using the source code:

```
$ git clone https://github.com/spacetelescope/lightcurve.git
$ cd lightcurve
$ python setup.py install
```

1.2 Simple extraction

Performing a basic lightcurve extraction from either COS or STIS individual datasets is achieved by a simple call to `read()` as shown below.

```
>>> import lightcurve
>>> table = lightcurve.read('lbova4b2q_corrtag_a.fits')
```

This returns the extracted data as an Astropy Table, which can be used for further analysis and plotting.

```
>>> type(lc)
astropy.table.table.Table
```

```
>>> lc
<Table length=750>
dataset background      mjd      error    bins  ...      flux      signal_to_noise ␣
 ↪counts  gross        net
float64  float32       float64    float32 float64 ...    float64         float32     ␣
 ↪float32 float32    float64
------- ---------- ------------- ------- ------- ... -------------- --------------- --
 ↪----- ------- -------------
    1.0    12.7314 55899.6041542 46.6337     1.0 ... 0.154727564001         46.3607␣
 ↪2149.24 2161.97 2149.23583984
    1.0    15.3585 55899.6041658 45.9691     1.0 ... 0.149929073704          45.635␣
 ↪2082.44  2097.8 2082.44238281
    1.0     11.119 55899.6041774  45.537     1.0 ... 0.147685074003         45.2928␣
 ↪2051.38  2062.5  2051.3815918
    1.0    16.2978 55899.6041889 45.5294     1.0 ... 0.146895813966         45.1715␣
 ↪2040.33 2056.63 2040.33349609
...
```

1.2.1 Instrument-specific extraction

More specific tailored extractions for each instrument can be seen on their respective pages below. These detail specific options for extracion, as well as how to setup parameters and reference files.

Extracting COS data

Reference API

Utility functions for extracting COS spectral data into lightcurves

Functions

`extract`(filename, **kwargs)	Extract lightcurve from COS dataset
`collect_inputs`(filename)	Populate HDU dictionary from available corrtag files
`get_both_filenames`(filename)	Get a list of both filenames for FUV data
`get_extraction_region`(hdu, segment[, mode])	Get y_start,y_end for given extraction

extract

`lightcurve.cos.`**`extract`**(*filename*, ***kwargs*)

Extract lightcurve from COS dataset

This is the main driver of the lightcuve extracion, and definitely needs some better documentation.

Parameters

- **`filename`** (*`str`*) – name of FITS file to extract from
- **`**kwargs`** (*`dict`*) – arbitrary keyword arguements for tailored extraction

Returns **data, meta** – Table with extracted data and dictionary of metadata pairs

Return type Astropy table, dict

collect_inputs

`lightcurve.cos.`**`collect_inputs`**(*filename*)

Populate HDU dictionary from available corrtag files

get_both_filenames

`lightcurve.cos.`**`get_both_filenames`**(*filename*)

Get a list of both filenames for FUV data

Regardless if rootname_corrtag_a.fits or rootname_corrtag_b.fits is passed in, both will be returned in a list.

Parameters **`filename`** (*`str`*) – full path to COS file

Returns **files** – rootname_corrtag_a.fits, rotname_corrtag_b.fits

Return type tuple

get_extraction_region

`lightcurve.cos.`**`get_extraction_region`**(*hdu*, *segment*, *mode='spectrum'*)

Get y_start,y_end for given extraction

Extracting STIS data

Reference API

Utility functions for extracting STIS spectral data into lightcurves

Functions

`extract`(filename, **kwargs)	Extract lightcurve from STIS dataset
`stis_corrtag`(tagfile[, clean])	Create a COS-like corrtag file for STIS data
`map_image`	
`epsilon`(tagfile)	Compute the total epsilon factor for each event
`dqinit`(tagfile)	Compute the data quality information for each pixel from the BPIXTAB.

extract

`lightcurve.stis.`**`extract`** (*filename*, ***kwargs*)

Extract lightcurve from STIS dataset

This is the main driver of the lightcuve extracion, and definitely needs some better documentation.

Parameters

- **filename** (*str*) – name of FITS file to extract from
- ****kwargs** (*dict*) – arbitrary keyword arguements for tailored extraction
- **parameters** (*Kwarg*) –
- **----------------** –

- **`verbosity`** (*`int, default=0`*) – Verbosity level for print output
- **`step`** (*`int, default=1`*) – timestep in seconds for output Lightcurve
- **`wlim`** (*`tuple`*) –

Returns **data, meta** – Table with extracted data and dictionary of metadata pairs

Return type Astropy table, dict

stis_corrtag

`lightcurve.stis.`**`stis_corrtag`**(*tagfile*, *clean=True*)

Create a COS-like corrtag file for STIS data

Parameters **`str`** (*`tagfile,`*) – input STIS time-tag data file

stis.map_image

`stis.`**`map_image`**

epsilon

`lightcurve.stis.`**`epsilon`**(*tagfile*)

Compute the total epsilon factor for each event

Compute the flatfield correction from the P-flat and L-flat reference files (PFLTFILE and LFLTFILE respectively).

Parameters **`str`** (*`tagfile,`*) – input STIS time-tag data file

Returns array of epsilons

Return type *epsilon*, np.ndarray

dqinit

`lightcurve.stis.`**`dqinit`**(*tagfile*)

Compute the data quality information for each pixel from the BPIXTAB.

Parameters **`str`** (*`tagfile,`*) – input STIS time-tag data file

Returns array of bitwise dq flags

Return type dq, np.ndarray

1.3 Complete API

1.3.1 lightcurve.io

Library of I/O routines to get data into a LightCurve object.

Functions

check_filetype(filename)	Determine the type of data being input.
read([source])	
composite(filelist, output[, trim])	Creates a composite lightcurve from files in filelist and saves it to the save_loc.
prepare_header(filename, filelist[, override])	Prepare headers with MAST requirements

check_filetype

lightcurve.io.**check_filetype**(*filename*)

Determine the type of data being input.

File type is determined by the culumns in the first data extension.

Parameters **filename** (*str*) – name of the input file

Returns **filetype** – determined type of file

Return type str

read

lightcurve.io.**read**(*source=None*, ***kwargs*)

composite

lightcurve.io.**composite**(*filelist*, *output*, *trim=True*, ***kwargs*)

Creates a composite lightcurve from files in filelist and saves it to the save_loc.

Parameters

- **filelist** (*list*) – A list of full paths to the input files.
- **output** (*string*) – The path to the location in which the composite lightcurve is saved.
- **trim** (*bool, opt*) – Trim wavelengths to common ranges for all files

prepare_header

lightcurve.io.**prepare_header**(*filename*, *filelist*, *override={}*)

Prepare headers with MAST requirements

1.3.2 lightcurve.analysis

Functions

lomb(time, counts, frequencies)	Compute the lombscargle periodigram

lomb

`lightcurve.analysis.`**`lomb`**(*time*, *counts*, *frequencies*)

Compute the lombscargle periodigram

Necessary wrapper around the set lomscargle algorithm https://github.com/scipy/scipy/issues/2643

Parameters

- **`time`** (*np.ndarray*) – array of data times
- **`counts`** (*np.ndarray*) – array of counts
- **`frequencies`** (*np.ndarray*) – What frequencies

Returns **freqs** – calculated freqencies

Return type np.ndarray

1.3.3 lightcurve.cos

Utility functions for extracting COS spectral data into lightcurves

Functions

extract(filename, **kwargs)	Extract lightcurve from COS dataset
collect_inputs(filename)	Populate HDU dictionary from available corrtag files
get_both_filenames(filename)	Get a list of both filenames for FUV data
get_extraction_region(hdu, segment[, mode])	Get y_start,y_end for given extraction

1.3.4 lightcurve.stis

Utility functions for extracting STIS spectral data into lightcurves

Functions

extract(filename, **kwargs)	Extract lightcurve from STIS dataset
stis_corrtag(tagfile[, clean])	Create a COS-like corrtag file for STIS data
map_image	
epsilon(tagfile)	Compute the total epsilon factor for each event
dqinit(tagfile)	Compute the data quality information for each pixel from the BPIXTAB.

1.3.5 lightcurve.utils

General purpose utility functions

Functions

expand_refname(refname)	Expand header reference file name to full path if $ is present.
enlarge(a[, x, y])	Enlarges 2D image array a using simple pixel repetition in both dimensions.
is_uniq(values)	Check if input items are unique

expand_refname

lightcurve.utils.**expand_refname**(*refname*)

Expand header reference file name to full path if $ is present.

Parameters **str** (*refname,*) – reference file name

Returns expanded full path to reference file

Return type reffile, str

enlarge

lightcurve.utils.**enlarge**(*a*, *x=2*, *y=None*)

Enlarges 2D image array a using simple pixel repetition in both dimensions. Enlarges by factor x horizontally and factor y vertically. If y is left as None, uses factor x for both dimensions.

is_uniq

lightcurve.utils.**is_uniq**(*values*)

Check if input items are unique

Parameters **values** (*set*) – set of all values

Returns

Return type True/False, MULTI/unique value

CHAPTER 2

Coding API

- genindex
- modindex
- search

CHAPTER 3

Issue Reporting

If you find bugs, problems, or even new features that you'd like to see, please report it on the github issue tracker.

CHAPTER 4

Citing

TODO

Python Module Index

Index