
flimage Documentation

Release 0.3.2

Paul Müller

Jul 01, 2021

CONTENTS

1	Introduction	3
1.1	The Problem	3
1.2	Why flimage?	3
1.3	Citing flimage	3
2	Getting started	5
2.1	Installing flimage	5
2.2	User API	5
2.2.1	Storing FLImage data on disk	5
2.2.2	Dealing with measurement series	6
2.2.3	Notes	6
3	Examples	7
3.1	Loading and plotting	7
4	HDF5 file format	9
4.1	FLImage	9
4.1.1	Attributes	9
4.1.2	Groups	10
4.2	FLSeries	10
5	Code reference	11
5.1	module level aliases	11
5.2	core (FLImage)	11
5.2.1	Classes	11
5.3	image_data (basic image management)	12
5.3.1	Classes	12
5.4	meta (definitions for FLImage meta data)	13
5.4.1	Constants	13
5.5	series (FLSeries)	13
5.5.1	Classes	13
6	Changelog	15
6.1	version 0.3.2	15
6.2	version 0.3.1	15
6.3	version 0.3.0	15
6.4	version 0.2.1	15
6.5	version 0.2.0	16
6.6	version 0.1.0	16
7	Bibliography	17

8 Indices and tables	19
Bibliography	21
Python Module Index	23
Index	25

Flimage is a Python library for handling fluorescence data. Its API is (as much as it makes sense) is identical to that of [qpimage](#), making it a convenient tool for phase- and fluorescence colocalization. This is the documentation of flimage version 0.3.2.

INTRODUCTION

1.1 The Problem

In correlative fluorescence and refractive index tomography, it is sometimes necessary to record phase and fluorescence images with different cameras, different frame rates, and different resolutions [SCG+17]. Handling these images and frame times easily gets out of hand (especially when they are stored in different files) and it would be nice to have a unified, data-driven interface to single fluorescence images and fluorescence series.

1.2 Why flimage?

There already exist many file formats for fluorescence data. I know that. However, I have made good experiences with the `qpimage` file format approach and wanted to have a similar API for the fluorescence data. This makes it easier to write transparent code for tomographic image analysis. Flimage also comes with a convenient bleach correction function.

1.3 Citing flimage

If you are using flimage in a scientific publication, please cite it with:

```
(...) using flimage version X.X.X (available at  
https://pypi.python.org/pypi/flimage).
```

or in a bibliography

```
Paul Müller (2017), flimage version X.X.X: Phase image analysis  
[Software]. Available at https://pypi.python.org/pypi/flimage.
```

and replace X.X.X with the version of flimage that you used.

GETTING STARTED

2.1 Installing flimage

Flimage is written in pure Python and supports Python version 3.6 and higher.

To install flimage, use one of the following methods (package dependencies will be installed automatically):

- **from PyPI:** `pip install flimage`
- **from sources:** `pip install .` or `python setup.py install`

2.2 User API

The flimage API is built upon the hdf5 file format using `h5py`. It is almost identical to the [user API of qpimage](#). Each instance of `flimage.FLImage` generates an hdf5 file, either on disk or in memory, depending on the preferences of the user.

2.2.1 Storing FLImage data on disk

To cache the FLImage data on disk, use the `with` statement in combination with the `h5file` keyword argument

```
with flimage.FLImage(data=fluorescence_ndarray, h5file="/path/to/file.h5"):
    pass
```

where all data is stored in `/path/to/file.h5`. This will create an hdf5 file on disk that, at a later time point, can be used to create an instance of `FLImage`:

```
# open previously cached data for reading
fli = flimage.FLImage(h5file="/path/to/file.h5", h5mode="r")

# or open cached data for writing (e.g. for changing the background)
with flimage.FLImage(h5file="/path/to/file.h5", h5mode="a") as fli:
    # do something here
```

The default value of `h5mode` is “a”, which means that data will be overridden. In the hdf5 file, the following data is stored:

- all data for reproducing the background-corrected fluorescence (`fli.fluorescence`) including
 - the experimental fluorescence data
 - the experimental background data

- all measurement specific meta data, given by the keyword argument `meta_data`

2.2.2 Dealing with measurement series

Flimage also comes with a *FLSeries* class for handling multiple instances of *FLImage* in one hdf5 file. For instance, to combine two *FLImages* in one series file, one could use:

```
paths = ["file_a.h5", "file_b.h5", "file_c.h5"]

with flimage.FLSeries(h5file="/path/to/series_file.h5", h5mode="w") as fls:
    for ii, pp in enumerate(paths):
        fli = flimage.FLImage(h5file="/path/to/file.h5", h5mode="r")
        fls.add_flimage(fli=fli, identifier="my_name_{}".format(ii))
```

Note that the function *add_flimage* accepts the optional keyword argument “identifier” (overriding the identifier of the *FLImage*) which can also be used for indexing later:

```
with flimage.FLSeries(h5file="/path/to/series_file.h5", h5mode="r") as fls:
    # these two are equivalent
    fli = fls[0]
    fli = fls["my_name_0"]
```

2.2.3 Notes

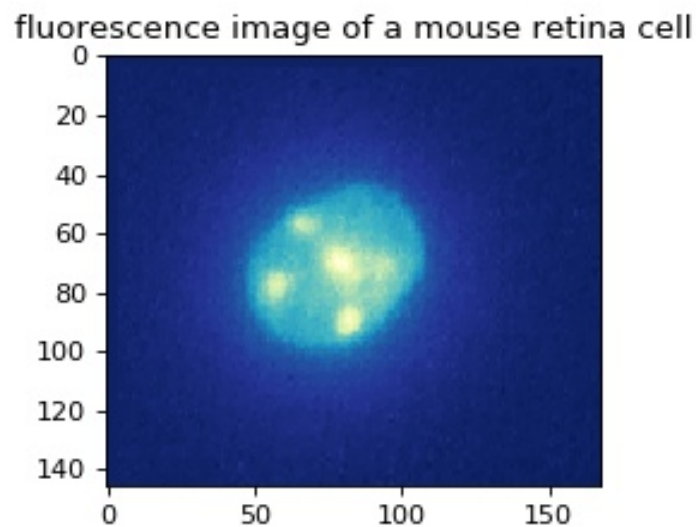
- Even though the hdf5 data is stored as gzip-compressed single precision floating point values, using flimage hdf5 files may result in file sizes that are considerably larger compared to when only the output of e.g. `qpi.fluorescence` is stored using e.g. `numpy.save()`.
- Units in flimage follow the international system of units (SI).
- *flimage.FLSeries* provides convenience functions for bleach correction.

EXAMPLES

3.1 Loading and plotting

This example illustrates the simple usage of the *flimage.FLImage* class for reading and managing quantitative phase data. The attribute *FLImage.fluorescence* yields the image data.

The image shows a young retina cell. The original dataset, available on [figshare](#) (*retina-young_sinogram_fli.h5*) [SCG+17], can be opened in the same manner using the *flimage.FLSeries* class.



simple_fluorescence.py

```
1 import matplotlib.pyplot as plt
2 import flimage
3
4 # load the experimental data
5 fli = flimage.FLImage(h5file="./data/retina_p10.h5", h5mode="r")
6
7 # plot
8 plt.figure(figsize=(4, 3))
9
10 plt.subplot(title="fluorescence image of a mouse retina cell")
11 plt.imshow(fli.fl, cmap="YlGnBu_r")
```

(continues on next page)

(continued from previous page)

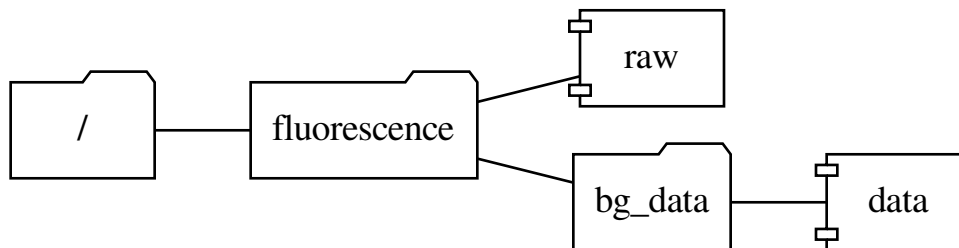
```
12  
13 plt.tight_layout()  
14 plt.show()
```

HDF5 FILE FORMAT

The data of a *flimage.FLImage* or *flimage.FLSeries* can be stored on disk, using the `h5file` parameter upon class instantiation. This section describes the scheme used to store the data using the [HDF5 file format](#).

4.1 FLImage

The following graph visualized the HDF5 file structure of an *FLImage* instance:



4.1.1 Attributes

These attributes of the root group (*/*) describe physical parameters of the data:

key	description
pixel size	detector pixel size [m]
time	acquisition time of the image [s]

These other attributes may be used by e.g. data simulators such as [cellsino](#):

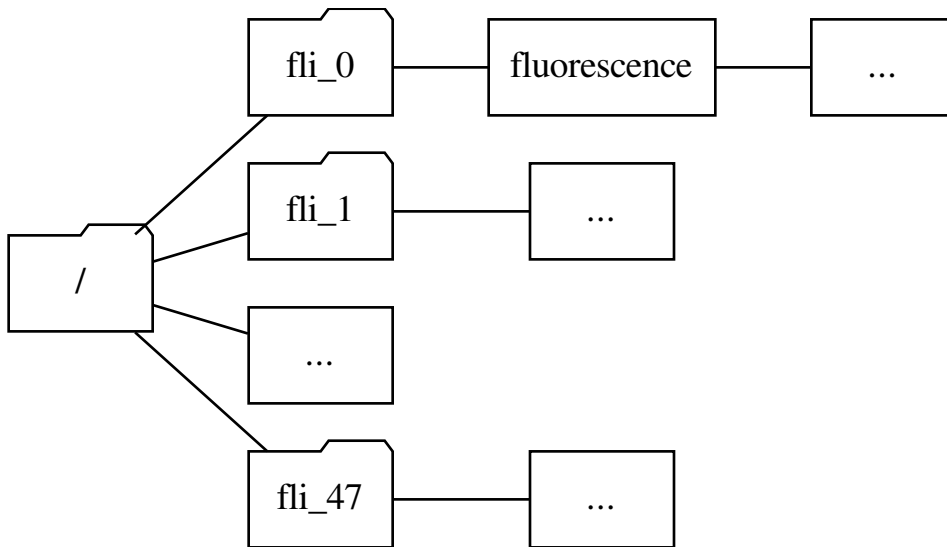
key	description
flimage version	flimage software version used
identifier	image identifier
qpimage version	qpimage software version used
sim center	Simulation: center of object [px]
sim model	Simulation: model used
sim radius	Simulation: object radius [m]

4.1.2 Groups

The group *fluorescence* does not hold attributes. It contains a dataset called *raw* (the raw fluorescence image) and a group called *bg_data* which may contain a dataset *data*, a simple background image.

4.2 FLSeries

The following graph visualized the HDF5 file structure of an FLSeries instance (with a total of 48 FLImages):



Note that the name of each FLImage group always starts with “fli_” and that the enumeration does not contain leading zeros. The root node (*/*) of an FLSeries may have the *identifier* attribute.

CODE REFERENCE

This section does not cover the base classes, methods, and constants of `qpimage`, which can be found [here](#).

5.1 module level aliases

For user convenience, the following objects are available at the module level.

```
class flimage.FLImage
    alias of flimage.core.FLImage

class flimage.FLSeries
    alias of flimage.series.FLSeries

flimage.META_KEYS_FL
    alias of flimage.meta.META_KEYS_FL
```

5.2 core (FLImage)

5.2.1 Classes

```
class flimage.core.FLImage(data=None, meta_data={}, h5file=None, h5mode='a', h5dtype='float32')
    Fluorescence image manipulation
```

This class makes available fluorescence microscopy data in a manner similar to `qpimage.QPImage`.

Parameters

- **data** (*2d ndarray (float or complex) or list*) – The experimental fluorescence image.
- **meta_data** (*dict*) – Meta data associated with the input data. see `flimage.meta.META_KEYS`
- **h5file** (*str, pathlib.Path, h5py.Group, h5py.File, or None*) – A path to an hdf5 data file where all data is cached. If set to *None* (default), all data will be handled in memory using the “core” driver of the `h5py`’s `File` class. If the file does not exist, it is created. If the file already exists, it is opened with the file mode defined by *hdf5_mode*. If this is an instance of `h5py.Group` or `h5py.File`, then this will be used to internally store all data.
- **h5mode** (*str*) – Valid file modes are (only applies if *h5file* is a path)
 - “r”: Readonly, file must exist

- "r+": Read/write, file must exist
- "w": Create file, truncate if exists
- "w-" or "x": Create file, fail if exists
- "a": Read/write if exists, create otherwise (default)
- **h5dtype** (*str*) – The datatype in which to store the image data. The default is "float32" which is sufficient for 2D image analysis and consumes only half the disk space of the numpy default "float64".

property bg_fl

background fluorescence image

property fl

background-corrected fluorescence image

property dtype

dtype of the fluorescence data array

property info

list of tuples with FLImage meta data

property meta

dictionary with imaging meta data

property raw_fl

raw fluorescence image

property shape

size of image dimensions

copy(*h5file=None*)

Create a copy of the current instance

This is done by recursively copying the underlying hdf5 data.

Parameters **h5file** (*str*, *h5py.File*, *h5py.Group*, or *None*) – see *FLImage.__init__*

set_bg_data(*bg_data*)

Set background fluorescence data

Parameters **bg_data** (2d ndarray, FLImage, or *None*) – The background data (must be same type as *data*). If set to *None*, the background data is reset.

5.3 image_data (basic image management)

5.3.1 Classes

class flimage.image_data.**Fluorescence**(*h5*, *h5dtype='float32'*)

Bases: *qpimage.image_data.Phase*

Dedicated class for fluorescence data

For fluorescence data, background correction is defined by subtracting the background image from the raw image.

Parameters

- **h5** (*h5py.Group*) – HDF5 group where all data is kept

- **h5dtype** (*str*) – The datatype in which to store the image data. The default is “float32” which is sufficient for 2D image analysis and consumes only half the disk space of the numpy default “float64”.

5.4 meta (definitions for FLImage meta data)

5.4.1 Constants

```
flimage.meta.META_KEYS_FL = ['pixel size', 'time', 'flimage version', 'identifier',
                             'qpimage version', 'sim center', 'sim model', 'sim radius']
    valid flimage.core.FLImage meta data keys
```

5.5 series (FLSeries)

5.5.1 Classes

```
class flimage.series.FLSeries(flimage_list=[], meta_data={}, h5file=None, h5mode='a', identifier=None)
    Fluorescence microscopy series data
```

Parameters

- **flimage_list** (*list*) – A list of instances of *flimage.FLImage*.
- **meta_data** (*dict*) – Meta data associated with the input data (see `flimage.META_KEYS`). This overrides the meta data of the FLImages in *flimage_list* and, if *h5file* is given and *h5mode* is not “r”, overrides the meta data in *h5file*.
- **h5file** (*str*, *h5py.Group*, *h5py.File*, or *None*) – A path to an hdf5 data file where all data is cached. If set to *None* (default), all data will be handled in memory using the “core” driver of the *h5py*’s *File* class. If the file does not exist, it is created. If the file already exists, it is opened with the file mode defined by *hdf5_mode*. If this is an instance of *h5py.Group* or *h5py.File*, then this will be used to internally store all data. If *h5file* is given and *flimage_list* is not empty, all FLImages in *flimage_list* are appended to *h5file* in the given order.
- **h5mode** (*str*) – Valid file modes are (only applies if *h5file* is a path):
 - “r”: Readonly, file must exist
 - “r+”: Read/write, file must exist
 - “w”: Create file, truncate if exists
 - “w-” or “x”: Create file, fail if exists
 - “a”: Read/write if exists, create otherwise (default)

property **identifier**

unique identifier of the series

add_flimage(*fli*, *identifier=None*)

Add a FLImage instance to the FLSeries

Parameters

- **fli** (*flimage.FLImage*) – The FLImage that is added to the series

- **identifier** (*str*) – Identifier key for *fli*

bleach_correction(*h5out=None, border_px=20, flscorr=None, count=None, max_count=None*)

Perform a correction for photobleaching

Bleaching is modeled with an exponential and an offset.

Parameters

- **h5out** (*path* or *h5py.Group*) – A new FLSeries will be written to this HDF5 file or h5py group.
- **border_px** (*int*) – Number of border pixels to include for background estimation.
- **flscorr** (*flimage.series.FLSeries*) – Apply the background correction to this FLSeries instead of the current instance. Use this in combination with filtered versions of the same series.

Returns

- **bg** (*float*) – Background value subtracted from each image
- **flint** (*1d ndarray*) – Fluorescence intensity trace extracted from the series
- **decay** (*1d ndarray*) – Exponential fit to *flint*
- **times** (*1d ndarray*) – Recording times corresponding to the indices in *flint* and *decay*

Notes

It is recommended to first denoise the fluorescence data with *FLSeries.denoise* to avoid an amplification of the background noise.

denoise(*h5file, count=None, max_count=None*)

Denoise fluorescence data

h5file specifies a path or HDF5 group for the FLSeries to which the denoised fluorescence data is written.

get_flimage(*index*)

Return a single FLImage of the series

Parameters *index* (*int* or *str*) – Index or identifier of the FLImage

Notes

Instead of `fls.get_flimage(index)`, it is possible to use the short-hand `fls[index]`.

CHANGELOG

List of changes in-between flimage releases.

6.1 version 0.3.2

- fix: easy fix for regression in h5py with backing_store=False

6.2 version 0.3.1

- tests: fix HDF5-related issue
- build: setup.py test is deprecated
- build: migrate from traxisCI to GH Actions
- docs: refurbish docs

6.3 version 0.3.0

- ref: renamed *meta.DATA_KEYS* to *meta.DATA_KEYS_FL* and removed keys that are not relevant for fluorescence imaging
- docs: initial documentation with code reference and example

6.4 version 0.2.1

- maintenance release

6.5 version 0.2.0

- feat: bleach correction and denoising of FLSeries (#2)
- tests: add them

6.6 version 0.1.0

- initial working state

BILBLIOGRAPHY

INDICES AND TABLES

- genindex
- modindex
- search

BIBLIOGRAPHY

- [SCG+17] M. Schürmann, G. Cojoc, S. Girardo, E. Ulbricht, J. Guck, and P. Müller. 3d correlative single-cell imaging utilizing fluorescence and refractive index tomography. *Journal of Biophotonics*, pages n/a, aug 2017. doi:10.1002/jbio.201700145.

PYTHON MODULE INDEX

f

`flimage.image_data`, 12

A

add_flimage() (*flimage.series.FLSeries* method), 13

B

bg_fl (*flimage.core.FLImage* property), 12

bleach_correction() (*flimage.series.FLSeries* method), 14

C

copy() (*flimage.core.FLImage* method), 12

D

denoise() (*flimage.series.FLSeries* method), 14

dtype (*flimage.core.FLImage* property), 12

F

fl (*flimage.core.FLImage* property), 12

FLImage (*class in flimage.core*), 11

flimage.FLImage (*built-in class*), 11

flimage.FLSeries (*built-in class*), 11

flimage.image_data
module, 12

flimage.META_KEYS_FL (*built-in variable*), 11

FLSeries (*class in flimage.series*), 13

Fluorescence (*class in flimage.image_data*), 12

G

get_flimage() (*flimage.series.FLSeries* method), 14

I

identifiant (*flimage.series.FLSeries* property), 13

info (*flimage.core.FLImage* property), 12

M

meta (*flimage.core.FLImage* property), 12

META_KEYS_FL (*in module flimage.meta*), 13

module

flimage.image_data, 12

R

raw_fl (*flimage.core.FLImage* property), 12

S

set_bg_data() (*flimage.core.FLImage* method), 12

shape (*flimage.core.FLImage* property), 12