
jicbioimage.illustrate Documentation

Release 0.6.1

Tjelvar Olsson and Matthew Hartley

November 01, 2016

1	Content	1
1.1	The <code>jicbioimage.illustrate</code> Python package	1
1.2	Attributions	2
1.3	API documentation	2
	Python Module Index	5

1.1 The `jicbioimage.illustrate` Python package

The `jicbioimage.illustrate` Python package provides a set of tools for creating annotated images and illustrations.

- Documentation: <http://jicbioimageillustrate.readthedocs.io>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.illustrate>
- PyPI: <https://pypi.python.org/pypi/jicbioimage.illustrate>
- Free software: MIT License

1.1.1 Features

- Functionality for creating illustrations and annotated images
- Cross-platform: Linux, Mac and Windows are all supported
- Works with Python 2.7, 3.3 and 3.4

1.1.2 Related packages

`jicbioimage`

- Documentation: <http://jicbioimage.readthedocs.io>
- GitHub: <https://github.com/JIC-CSB/jicbioimage>

`jicbioimage.core`

- Documentation: <http://jicbioimagecore.readthedocs.io>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.core>

`jicbioimage.transform`

- Documentation: <http://jicbioimagetransform.readthedocs.io>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.transform>

jicbioimage.segment

- Documentation: <http://jicbioimagesegment.readthedocs.io>
- GitHub: <https://github.com/JIC-CSB/jicbioimage.segment>

1.2 Attributions

The package includes the UbuntuMono font (UBUNTU FONT LICENCE).

1.3 API documentation

1.3.1 jicbioimage.illustrate

Module for creating illustrations.

To create an annotated image we need an instance of the `jicbioimage.illustrate.AnnotatedImage` class.

```
>>> from jicbioimage.illustrate import AnnotatedImage
```

Suppose that we have an existing image.

```
>>> from jicbioimage.core.image import Image
>>> im = Image((50, 50))
```

We can use this image to create an canvas instance populated with the data as a RGB gray scale image.

```
>>> canvas = AnnotatedImage.from_grayscale(im)
```

The `jicbioimage.illustrate.Canvas` instance has built in annotation functionality.

One can use it to draw crosses.

```
>>> canvas.draw_cross(10, 20)
```

One can use it to mask out bitmaps (in the example below with the color cyan).

```
>>> bitmap = np.zeros((50, 50), dtype=bool)
>>> bitmap[30:40, 30:40] = True
>>> canvas.mask_region(bitmap, color=(0, 255, 255))
```

One can use it to add text at particular locations on the canvas.

```
>>> canvas.text_at("Hello", 30, 60)
```

class jicbioimage.illustrate.AnnotatedImage

Class for building up annotated images.

static from_grayscale (*im*, *channels_on*=(*True*, *True*, *True*))
Return a canvas from a grayscale image.

Parameters *im* – single channel image

Channels_on channels to populate with input image

Returns `jicbioimage.illustrate.Canvas`

class `jicbioimage.illustrate.Canvas`

Class for building up annotated images.

static `blank_canvas` (*width*, *height*)

Return a blank canvas to annotate.

Parameters

- **width** – xdim (int)
- **height** – ydim (int)

Returns `jicbioimage.illustrate.Canvas`

draw_cross (*position*, *color*=(255, 0, 0), *radius*=4)

Draw a cross on the canvas.

Parameters

- **position** – (row, col) tuple
- **color** – RGB tuple
- **radius** – radius of the cross (int)

draw_line (*pos1*, *pos2*, *color*=(255, 0, 0))

Draw a line between pos1 and pos2 on the canvas.

Parameters

- **pos1** – position 1 (row, col) tuple
- **pos2** – position 2 (row, col) tuple
- **color** – RGB tuple

mask_region (*region*, *color*=(0, 255, 0))

Mask a region with a color.

Parameters

- **region** – `jicbioimage.core.region.Region`
- **color** – RGB tuple

text_at (*text*, *position*, *color*=(255, 255, 255), *size*=12, *antialias*=False, *center*=False)

Write text at x, y top left corner position.

By default the x and y coordinates represent the top left hand corner of the text. The text can be centered vertically and horizontally by using setting the `center` option to `True`.

Parameters

- **text** – text to write
- **position** – (row, col) tuple
- **color** – RGB tuple
- **size** – font size
- **antialias** – whether or not the text should be antialiased
- **center** – whether or not the text should be centered on the input coordinate

j

`jicbioimage.illustrate`, [2](#)

A

`AnnotatedImage` (class in `jicbioimage.illustrate`), 2

B

`blank_canvas()` (`jicbioimage.illustrate.Canvas` static method), 3

C

`Canvas` (class in `jicbioimage.illustrate`), 2

D

`draw_cross()` (`jicbioimage.illustrate.Canvas` method), 3

`draw_line()` (`jicbioimage.illustrate.Canvas` method), 3

F

`from_grayscale()` (`jicbioimage.illustrate.AnnotatedImage` static method), 2

J

`jicbioimage.illustrate` (module), 2

M

`mask_region()` (`jicbioimage.illustrate.Canvas` method), 3

T

`text_at()` (`jicbioimage.illustrate.Canvas` method), 3