
StainTools Documentation

Release 0.1.0

Peter Byfield

May 18, 2018

Contents

1	Install	3
2	Example usage	5
3	Docs	7
4	Contents	9
4.1	Normalization	9
4.2	Utils	12
5	Indices and tables	15
	Python Module Index	17

Tools for stain normalization and augmentation in Python (tested on 3.5). GitHub repository [here](#).

Latest build:

CHAPTER 1

Install

```
pip install staintools
```

Note: StainTools requires the SPAMS (SPArse Modeling Software) package. Please find out about this [here](#). This may be easily installed via conda. For example, see [here](#).

CHAPTER 2

Example usage

Please see `demo.ipynb` [here](#).

CHAPTER 3

Docs

Made with Sphinx and Read the Docs [here](#).

4.1 Normalization

4.1.1 Abstract base classes

Normalizer abstract base classes

```
class staintools.normalization.normalizer_abc.FancyNormalizer (**kwargs)
```

Abstract class for a ‘fancy’ normalizer (inherits from Normalizer). Adds methods for stain matrix and source concentration estimation.

```
    fetch_target_stains ()
```

Fetch the target stain matrix and convert from OD to RGB. Must call fit first (this builds the stain matrix).

Returns

```
    fit (target)
```

Fit to a target image.

Parameters **target** – Target image RGB uint8.

Returns

```
    static get_concentrations (I, stain_matrix, lamda=0.01)
```

Get the concentration matrix. Suppose the input image is $H \times W \times 3$ (uint8). Define $N_{\text{pix}} = H \times W$. Then the concentration matrix is $N_{\text{pix}} \times 2$ (or we could reshape to $H \times W \times 2$). The first element of each row is the Hematoxylin concentration. The second element of each row is the Eosin concentration.

We do this by ‘solving’ $OD = C \cdot S$ (Matrix product) where OD is optical density ($N_{\text{pix}} \times 3$), C is concentration ($N_{\text{pix}} \times 2$) and S is stain matrix (2×3). See docs for spams.lasso.

We restrict the concentrations to be positive and penalise very large concentration values, so that background pixels (which can not easily be expressed in the Hematoxylin-Eosin basis) have low concentration and thus appear white.

Parameters

- **I** – Image. A np array HxWx3 of type uint8.
- **stain_matrix** – a 2x3 stain matrix. First row is Hematoxylin stain vector, second row is Eosin stain vector.

Returns The Nx2 concentration matrix, where $N=H*W$ is number of pixels.

get_stain_matrix (*I*, *args)

Estimate stain matrix given an image and relevant method parameters

hematoxylin (*I*)

Hematoxylin channel extraction.

Parameters **I** – Image RGB uint8.

Returns

transform (*I*)

Transform an image.

Parameters **I** – Image RGB uint8.

Returns

class `staintools.normalization.normalizer_abc.Normaliser` (**kwargs)

Abstract base class for normalizers. Defines some necessary methods to be considered a normalizer.

fit (*target*)

Fit the normalizer to an target image

transform (*I*)

Transform an image to the target stain

4.1.2 Reinhard method

class `staintools.normalization.reinhard.ReinhardNormalizer` (**kwargs)

Normalize a patch stain to the target image using the method of: E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley, ‘Color transfer between images’, IEEE Computer Graphics and Applications, vol. 21, no. 5, pp. 34–41, Sep. 2001.

fit (*target*)

Fit to a target image

Parameters **target** – Image RGB uint8.

Returns

get_mean_std (*I*)

Get mean and standard deviation of each channel.

Parameters **I** – Image RGB uint8.

Returns

static lab_split (*I*)

Convert from RGB uint8 to LAB and split into channels.

Parameters **I** – Image RGB uint8.

Returns

static merge_back (*I1, I2, I3*)

Take separate LAB channels and merge back to give RGB uint8.

Parameters

- **I1** – L
- **I2** – A
- **I3** – B

Returns Image RGB uint8.

transform(*I*)

Transform an image.

Parameters **I** – Image RGB uint8.

Returns

4.1.3 Macenko method

class `staintools.normalization.macenko.MacenkoNormalizer` (**kwargs)

Stain normalization based on the method of: M. Macenko et al., ‘A method for normalizing histology slides for quantitative analysis’, in 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro, 2009, pp. 1107–1110.

fit(*target*)

Fit to a target image.

Parameters **target** – Image RGB uint8.

Returns

static **get_stain_matrix**(*I*, *beta*=0.15, *alpha*=1)

Get the stain matrix (2x3). First row H and second row E. See the original paper for details.

Parameters

- **I** – Image RGB uint8.
- **beta** –
- **alpha** –

Returns

transform(*I*)

Transform an image.

Parameters **I** – Image RGB uint8.

Returns

4.1.4 Vahadane method

class `staintools.normalization.vahadane.VahadaneNormalizer` (**kwargs)

Stain normalization inspired by method of: A. Vahadane et al., ‘Structure-Preserving Color Normalization and Sparse Stain Separation for Histological Images’, IEEE Transactions on Medical Imaging, vol. 35, no. 8, pp. 1962–1971, Aug. 2016.

static **get_stain_matrix**(*I*, *threshold*=0.8, *lamda*=0.1)

Get 2x3 stain matrix. First row H and second row E. See the original paper for details. Also see spams docs.

Parameters

- **I** – Image RGB uint8.

- **threshold** –
- **lamda** –

Returns

4.2 Utils

4.2.1 Visualization Utils

Visualization utilities.

`staintools.utils.visual.build_stack(images)`

Build a stack of images from a tuple/list of images.

Parameters **images** – A tuple/list of images.

Returns

`staintools.utils.visual.patch_grid(ims, width=5, sub_sample=False, rand=False, save_name=None)`

Display a grid of patches.

Parameters

- **ims** – A patch ‘stack’
- **width** – Images per row.
- **sub_sample** – Should we take a subsample?
- **rand** – Randomize subsample?

Returns

`staintools.utils.visual.read_image(path)`

Read an image to RGB uint8. Read with opencv (cv) and covert from BGR colorspace to RGB.

Parameters **path** – The path to the image.

Returns RGB uint8 image.

`staintools.utils.visual.show(image, now=True, fig_size=(10, 10))`

Show an image (np.array). Caution! Rescales image to be in range [0,1].

Parameters

- **image** –
- **now** – plt.show() now?
- **fig_size** – Figure size.

Returns

`staintools.utils.visual.show_colors(C)`

Visualize rows of C as colors (RGB)

Parameters **C** – An array N x 3 where the rows are considered as RGB colors.

Returns

4.2.2 Misc Utils

Other utilities.

`staintools.utils.misc.OD_to_RGB(OD)`

Convert from optical density (OD_RGB) to RGB $RGB = 255 * \exp(-1 * OD_RGB)$

Parameters `OD` – Optical denisty RGB image.

Returns Image RGB uint8.

`staintools.utils.misc.RGB_to_OD(I)`

Convert from RGB to optical density (OD_RGB) space. $RGB = 255 * \exp(-1 * OD_RGB)$.

Parameters `I` – Image RGB uint8.

Returns Optical denisty RGB image.

`staintools.utils.misc.array_equal(A, B, eps=1e-09)`

Are arrays A and B equal?

Parameters

- `A` – Array.
- `B` – Array.
- `eps` – Tolerance.

Returns True/False.

`staintools.utils.misc.check_image(x)`

Check if is an image. If gray make sure it is 'squeezed' correctly.

Parameters `x` – Input.

Returns True/False.

`staintools.utils.misc.is_gray_image(x)`

Is x a gray image?

Parameters `x` – Input.

Returns True/False.

`staintools.utils.misc.is_image(x)`

Is x an image? i.e. numpy array of 2 or 3 dimensions.

Parameters `x` – Input.

Returns True/False.

`staintools.utils.misc.is_uint8_image(x)`

Is x a uint8 image?

Parameters `x` – Input.

Returns True/False.

`staintools.utils.misc.normalize_rows(A)`

Normalize the rows of an array.

Parameters `A` – An array.

Returns Array with rows normalized.

`staintools.utils.misc.notwhite_mask(I, thresh=0.8)`

Get a binary mask where true denotes ‘not white’. Specifically, a pixel is not white if its luminance (in LAB color space) is less than the specified threshold.

Parameters

- **I** – RGB uint 8 image.
- **thresh** – Luminosity threshold.

Returns Binary mask where true denotes ‘not white’.

`staintools.utils.misc.remove_zeros(I)`

Remove zeros in an image, replace with 1’s.

Parameters **I** – An Array.

Returns New array where 0’s have been replaced with 1’s.

`staintools.utils.misc.sign(x)`

Returns the sign of x.

Parameters **x** – A scalar x.

Returns The sign of x in (+1, -1, 0).

`staintools.utils.misc.standardize_brightness(I, percentile=95)`

Standardize brightness.

Parameters **I** – Image uint8 RGB.

Returns Image uint8 RGB with standardized brightness.

CHAPTER 5

Indices and tables

- `genindex`
- `modindex`
- `search`

S

`staintools.normalization.macenko`, [11](#)
`staintools.normalization.normalizer_abc`,
 [9](#)
`staintools.normalization.reinhard`, [10](#)
`staintools.normalization.vahadane`, [11](#)
`staintools.utils.misc`, [13](#)
`staintools.utils.visual`, [12](#)

A

array_equal() (in module staintools.utils.misc), 13

B

build_stack() (in module staintools.utils.visual), 12

C

check_image() (in module staintools.utils.misc), 13

F

FancyNormalizer (class in staintools.normalization.normalizer_abc), 9

fetch_target_stains() (staintools.normalization.normalizer_abc.FancyNormalizer method), 9

fit() (staintools.normalization.macenko.MacenkoNormalizer method), 11

fit() (staintools.normalization.normalizer_abc.FancyNormalizer method), 9

fit() (staintools.normalization.normalizer_abc.Normaliser method), 10

fit() (staintools.normalization.reinhard.ReinhardNormalizer method), 10

G

get_concentrations() (staintools.normalization.normalizer_abc.FancyNormalizer static method), 9

get_mean_std() (staintools.normalization.reinhard.ReinhardNormalizer method), 10

get_stain_matrix() (staintools.normalization.macenko.MacenkoNormalizer static method), 11

get_stain_matrix() (staintools.normalization.normalizer_abc.FancyNormalizer method), 10

get_stain_matrix() (staintools.normalization.vahadane.VahadaneNormalizer static method), 11

H

hematoxylin() (staintools.normalization.normalizer_abc.FancyNormalizer method), 10

I

is_gray_image() (in module staintools.utils.misc), 13

is_image() (in module staintools.utils.misc), 13

is_uint8_image() (in module staintools.utils.misc), 13

L

lab_split() (staintools.normalization.reinhard.ReinhardNormalizer static method), 10

M

MacenkoNormalizer (class in staintools.normalization.macenko), 11

merge_back() (staintools.normalization.reinhard.ReinhardNormalizer static method), 10

N

Normaliser (class in staintools.normalization.normalizer_abc), 10

normalize_rows() (in module staintools.utils.misc), 13

notwhite_mask() (in module staintools.utils.misc), 13

O

OD_to_RGB() (in module staintools.utils.misc), 13

P

patch_grid() (in module staintools.utils.visual), 12

R

read_image() (in module staintools.utils.visual), 12

ReinhardNormalizer (class in staintools.normalization.reinhard), 10

remove_zeros() (in module staintools.utils.misc), 14

RGB_to_OD() (in module staintools.utils.misc), 13

S

`show()` (in module `staintools.utils.visual`), [12](#)
`show_colors()` (in module `staintools.utils.visual`), [12](#)
`sign()` (in module `staintools.utils.misc`), [14](#)
`staintools.normalization.macenko` (module), [11](#)
`staintools.normalization.normalizer_abc` (module), [9](#)
`staintools.normalization.reinhard` (module), [10](#)
`staintools.normalization.vahadane` (module), [11](#)
`staintools.utils.misc` (module), [13](#)
`staintools.utils.visual` (module), [12](#)
`standardize_brightness()` (in module `staintools.utils.misc`), [14](#)

T

`transform()` (`staintools.normalization.macenko.MacenkoNormalizer` method), [11](#)
`transform()` (`staintools.normalization.normalizer_abc.FancyNormalizer` method), [10](#)
`transform()` (`staintools.normalization.normalizer_abc.Normaliser` method), [10](#)
`transform()` (`staintools.normalization.reinhard.ReinhardNormalizer` method), [11](#)

V

`VahadaneNormalizer` (class in `staintools.normalization.vahadane`), [11](#)