stainlib: a python library for augmentation and normalization of histopathology H&E images.

Sebastian Otálora[a,e], Niccoló Marini[a,e], Damian Podareanu[b], Ruben Hekster[b], David Tellez[c], Jeroen Van Der Laak[c], Henning Müller[a], Manfredo Atzori[a,d]

[a] *Information Systems Institute, University of Applied Sciences Western Switzerland (HES-SO Valais), Technopôle 3, 3960 Sierre, Switzerland*
[b] *SURF BV, Science Park 140, 1098 XG Amsterdam, Netherlands*
[c] *Diagnostic Image Analysis Group, Department of Pathology, Radboud University Medical Center, Geert Grooteplein Zuid 10, 6525 GA Nijmegen, Netherlands*
[d] *Department of Neurosciences, University of Padua, Via Giustiniani 2, 35128, Padua, Italy*
[e] *Computer Science Centre (CUI), University of Geneva, Route de Drize 7, Battelle A, Carouge, Switzerland*
[f] *Division of Radiology, Geneva University Hospitals (HUG), Rue Gabrielle-Perret-Gentil 4, Geneva, Switzerland*

Abstract

Computational pathology is a domain of increasing scientific and social interest. The automatic analysis of histopathology images stained with Hematoxylin and Eosin (H&E) can help clinicians diagnose and quantify diseases. Computer vision methods based on deep learning can perform on par or better than pathologists in specific tasks [1, 2, 15]. Nevertheless, the visual heterogeneity in histopathology images due to batch effects, differences in preparation in different pathology laboratories, and the scanner can produce tissue appearance changes in the digitized whole-slide images. Such changes impede the application of the trained models in clinical scenarios where there is high variability in the images. We introduce `stainlib`, an easy-to-use and expandable python3 library that collects and unifies state-of-the-art methods for color augmentation and normalization of histopathology H&E images. `stainlib` also contains recent deep learning-based approaches that perform a robust stain-invariant training of CNN models. `stainlib` can help researchers build models robust to color domain shift by augmenting and harmonizing the training data, allowing the deployment of better models in the digital pathology practice.

Keywords: Computational Pathology, Stain Augmentation, Stain normalization, Data augmentation

Code Metadata

Current code version

Nr.	Code metadata description	1.0.0
C1	Current code version	1.0.0
C2	Permanent link to code/repository used for this code version	$https : //github.com/sebastianffx/stainlib$
C3	Code Ocean compute capsule	
C4	Legal Code License	MIT
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	python, jupyter notebook
C7	Compilation requirements, operating environments & dependencies	scikit-image, scipy, pillow, opencv-python, spams
C8	If available Link to developer documentation/manual	
C9	Support email for questions	juan.otalora@etu.unige.ch

Table 1: Code metadata (mandatory).

1. Motivation and significance

During the last decade, the automatic analysis of digital pathology images has increased until the point where commercial products are now available, and new ones are being cleared by health control and supervision organizations. In research, modern computational pathology techniques are based on the steady development of deep learning algorithms and deep convolutional neural networks (CNN) [13, 9]. Shifting from handcrafted features towards end–to–end training of deep learning models made it possible to automatically detect cancer in digitized histopathology images, both, in image regions and at the whole-slide-image level, with performances previously unseen [1, 2].

Methods have become more reliable, achieving in some cases a performance that is comparable to pathologists for specific segmentation and classification tasks [1, 5, 2, 15, 9, 12]. Despite the remarkable good performance of some methods, there are still technical barriers that prevent the translation of these advances into clinical applications [17]. A clinically applicable deep learning method needs to be able to cope with the heterogeneity in the color

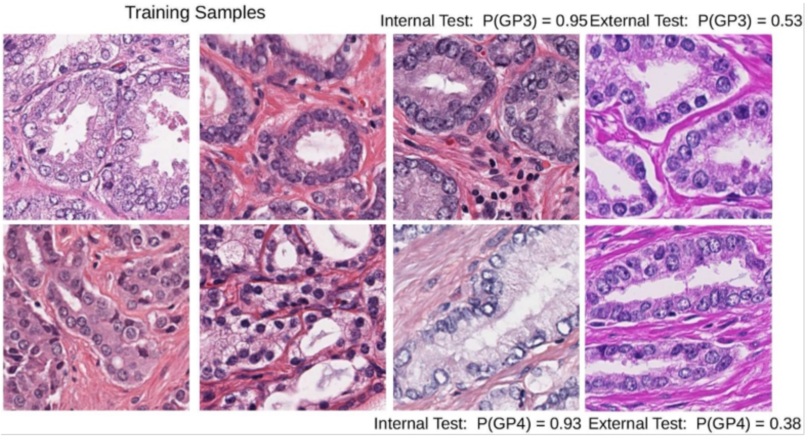

Training Samples
Internal Test: P(GP3) = 0.95
External Test: P(GP3) = 0.53
Internal Test: P(GP4) = 0.93
External Test: P(GP4) = 0.38

Figure 1: Training images with H&E concentrations that are noticeable different from external test sets can lead to model misclassifications; image taken from [14]. P(GP3) and P(GP4) stands for probability of the image region to contain Gleason patterns 3 and 4 respectively.

of the images that arises from preparing and staining the tissue samples in the pathology laboratory [14, 19]. Stains are chemical reagents that attach to specific proteins and that are used to enhance the contrast between different tissue structures for their examination under a microscope by a pathologist. The most commonly used stains are a combination of hematoxylin and eosin (H&E). These two reagents highlight the nuclei DNA content with a dark blue-purple color (Hematoxylin) and cytoplasm and stromal matrix contents with a light pink-red color (Eosin), see Figure 1 for exemplar H&E stained image regions of prostate tissue.

One of the most important factors preventing the application of machine learning methods in clinical practice is related to the heterogeneity of H&E images due to the many parameters involved in the tissue preparation and the digital scanning process (temperature of the tissue, the thickness of the cuts, the image sensor of the digital camera, the stitching algorithm among others) [11]. Figure 1 shows an example of stain variation in training and test sets and its impact on the performance of a CNN model trained only with partial variations. Two approaches are most commonly applied to take into account such variations when training CNN models. First, methods that transform an input H&E image given a target image template are known as "stain normalization". Their aim is to match the input color distributions (or H&E concentrations) with the one given in a target image. The second approach refers to "stain" or "color augmentation methods", which create new synthetic samples to increase the training dataset size, creating more robust models regarding color variations. There are novel image processing and machine learning techniques reported in the literature to deal with color heterogeneity, improving classification, and segmentation performance for various tissue types [19, 6, 14, 4, 11]. While the specific normalization technique depends on the task to solve [19, 4], recent work has reported consistent improvements in performance and robustness to external datasets employing color augmentation techniques [19, 4] or a combination of normalization and augmentation [14].

There are existing tools and methods to deal with H&E color heterogeneity, however not unified under a standard tool-set, many are also written in different programming languages, with various dependencies, and others are unknown by some researchers.

Few libraries comprise multiple methods for H&E image normalization and augmentation. Furthermore, only a handful of methods tackle color heterogeneity in H&E images using the modern machine and deep learning techniques. The codebase from articles in the literature is mostly in self-contained repositories, and its evaluation is usually performed in ad-hoc tasks using specific and often private datasets. The lack of libraries with multiple methods limits the possibilities to evaluate the best strategy to deal with color heterogeneity for new datasets or tasks.

This article aims to present and validate `stainlib`, an easy-to-use, extensible library to extract homogeneous representations of heterogeneous color information. With `stainlib` we make an effort to find, extract, collect, test, and unify most of the existing methods into a single library and make them easy to use. `stainlib` includes the most commonly used methods for color augmentation and normalization of histopathology images, having input local image regions (or patches). It contains classical machine learning and novel deep learning techniques to tackle the heterogeneity of color in H&E images.

1.1. Related work

QuPath, Staintools, and HistomicksTK, are likely among the most popular existing software tools to deal with color heterogeneity. QuPath (`https://qupath.github.io/`.) is an open and extensible software platform for Whole Slide Image (WSI) analysis. It includes methods for estimating and setting stain vectors. Scripts created for running specific color normalization methods can also be used within QuPath. Due to the big codebase of QuPath, it is challenging to run a classification or segmentation model without having to write a considerable amount of scripts to have a full pipeline, taking into account datasets with considerable color heterogeneity.

Staintools is a set of tools for tissue stain normalization and augmentation in python 3. It contains implementations that follow the same coding style of scikit-learn, where the methods are made to fit or train a model. It is open-source and can be downloaded from the Github repository: `https://github.com/Peter554/StainTools`. The library contains two extractive normalization methods (Macenko, Vahadane), and the only augmentation techniques included in staintools are based on the same extractive methods by modifying the estimated stain concentrations.

HistomicksTK is a python toolkit for histopathology image analysis. It contains several methods for stain normalization and color augmentation based on the stain perturbation methods from Tellez et al. [18]. It can be downloaded from `https://pypi.org/project/histomicstk/`. The HistomicksTK toolkit contains many overlapping methods with `stainlib` but it lacks modularity to use the color tools as standalone modules, which creates difficulties for its usage in different research scenarios. Despite the existence of few libraries, the domain still lacks a modular library including both standard and more modern methods that can be easily evaluated on varying datasets.

2. Software description

Stainlib is a python library containing methods for H&E image normalization and augmentation. The objective is to develop an easy-to-use python3 library that includes the most commonly used methods for color augmentation and normalization of histopathology images, having as input local image regions and to add more recent methods to tackle color heterogeneity based on deep learning approaches, too. In Figure 2, the structure and methods included in the library are displayed. The library can be downloaded from the following github repository: `https://github.com/sebastianffx/stainlib`.

2.1. Software Architecture

`stainlib` is composed of three main modules `stainlib.augmentation`, `stainlib.normalization`, `stainlib.dlmodels`. The methods and the underlying theory of each of the modules are explained in the following subsections.

2.1.1. stainlib.normalization

In digital pathology, the thin slice tissue cuts that are counter-stained in the H&E tissue slides are digitized using digital tissue scanners. The stained tissue slide's light absorbance is quantified and represented in a computer as a two-dimensional digital image (despite coming from a three-dimensional biological structure). In general, if the digital representation of the image is in the RGB color space, each pixel should contain a composition of the color representation of Hematoxylin (*purple*), Eosin(*pink*), and background (*white*).

Images acquired from the same center and using the same preparation methods share similar stain absorbance coefficients, which can be written as the linear transformation (omitting background that should be close to white for the three channels):

$$S = \begin{pmatrix} H_R & H_G & H_B \\ E_R & E_G & E_B \end{pmatrix}$$

Where the first-row vector corresponds to the RGB components of hematoxylin and the second one to the components of Eosin. In staining normalization methods, the aim is to estimate the individual staining absorbance coefficients of the images S and quantify the absorbed light $\mathbf{C}$ by the tissue when it is scanned, which is the value in the H&E space of each pixel. The Beer-Lambert law provides a way to estimate them in the optical density space, given the original pixel content for the c-channel I_c:

$$I_c = I_0 \exp(-S_c \cdot \mathbf{C})$$

Where c ranges in the RGB channels, $S \in [0, +\infty]^{3\times 2}$ is the matrix of absorbance coefficients, $\mathbf{C} \in [0, +\infty]^2$ is the vector of the two staining concentration coefficients and I_0 is the background value.

Several well-known stain extraction methods provide an estimation of S. In the widely used method of Macenko [10], this matrix is computed by calculating a plane using the two largest singular value decomposition vectors of the image and then projecting the data into this plane and clipping extreme values. In the method of Vahadane [20] this estimation is done by learning a sparse non-negative matrix factorization.

An alternative approach is to use residual flows for invertible generative modeling [3]. Flow-based generative models parameterize probability distributions through an invertible transformation and can be trained by maximum likelihood. Invertible residual networks provide a flexible family of transformations where only Lipschitz conditions rather than strict architectural constraints are needed for enforcing invertibility.

In `stainlib` we include the Vahadane and Macenko methods using the code base from the implementations in the staintools library[1]. In the Reinhard method [16], the color histogram of the source image (in the LAB color

[1] Staintools github repository: `https://github.com/Peter554/StainTools/tree/master/staintools`

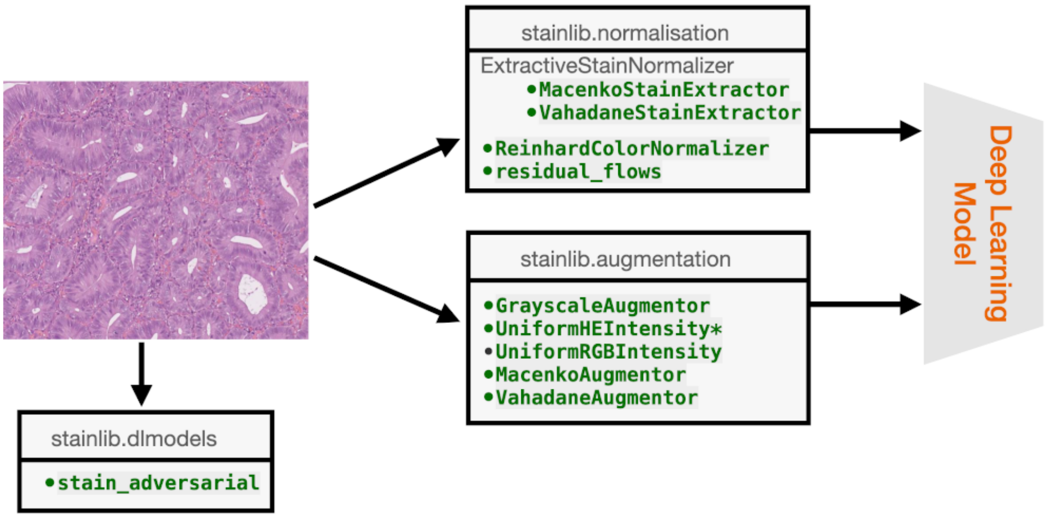


Figure 2: Implemented methods in stainlib for stain normalization and augmentation. The first version of the library includes all the methods in green and linked submodules of the residual flows and stain adversarial methods. For next versions of the library, further state of the art methods will be included.

space) is matched with the target image. Despite its simplicity and original domain of application of natural images, it yields good results in histopathology images. We have also included the Reinhard normalization method in `stainlib`.

The invertible flows method is fully compatible with `stainlib` and can be used from the base implementation [2].

2.1.2. stainlib.augmentation

It is now well known that deep learning classification and segmentation models for histopathology yield better results when data augmentation is used [19, 6, 14, 7]. The benefits of data augmentation might be intuitive in training deep learning models, where the larger the amount of data the model is fed with, the more variations the model is exposed to. Therefore, data augmentation usually can make the model more robust to changes in appearance in the test set. When there is a wide range of images with variations in color and preparation sources included in the training set, the models are more likely to output the correct prediction for new samples. Such a range of variations could be synthetically simulated, especially for the color variations in tissue appearance due to the stain concentrations. In

[2] Invertible flows github repository:`https://github.com/sara-nl/color-information`

`stainlib` we have included five stain augmentation methods: 1) grayscale transformation, 2) shifts of the stain concentrations in H&E space, 3) shifts of the stain concentrations in RGB space, 4) shifts of the stain concentration matrix using the Macenko method and 5) shifts of the stain concentration matrix using the Vahadane method.

For the RGB to grayscale transformation, the method `rgb2gray` from the `scikit-image` library is used to generate realistic samples using random uniform variations of the grayscale values within a fixed range as follows:

$$Grayscale_{aug} \leftarrow a_c Grayscale_{aug} + b_c$$

Where a_c and b_c are values drawn from a uniform distribution in the [0,1] range. Similarly, for the shifts of the RGB channels, we generate new samples as follows for each channel, c as follows:

$$I'_c \leftarrow a_c I_c + b_c$$

Again, a_c and b_c are values drawn from a uniform distribution in the [0,1] range. In the case of the Macenko and Vahadane augmentation, we use the estimated stain H&E concentration matrices of the methods and shift them as follows:

$$S'_c \leftarrow a_c S_c + b_c$$

Where the values a_c and b_c are drawn from a uniform distribution in the $[1-\alpha_c, 1+\alpha_c]$ and $[1-\beta_c, 1+\beta_c]$ range. Values of $\alpha = 0.2$ and $\beta_c = 0.2$ were set as default values as they usually yield good qualitative and quantitative results in experiments. In the case of the augmentations based on the shifts of the stain concentrations in H&E space, we followed the implementation of Tellez et al. [18]. Section 3 presents a qualitative evaluation of the normalization and augmentation methods included in `stainlib`. Quantitative evaluation of these methods has also been performed in previous research from the authors [14, 6, 19].

2.1.3. stainlib.dlmodels

Domain-invariant training of CNN's is a promising technique to address training a single model for different domains. It includes the source domain information to guide the training towards domain-invariant features, allowing to achieve state-of-the-art results in classification tasks. In the case of training classification models with histopathology images, the domain represents the center where the tissue preparation characteristics are similar, e.g., hospital A, hospitals B. This technique shows excellent generalization performance to external test sets, and further improvements have been reported, when combined with data augmentation techniques [14, 8].

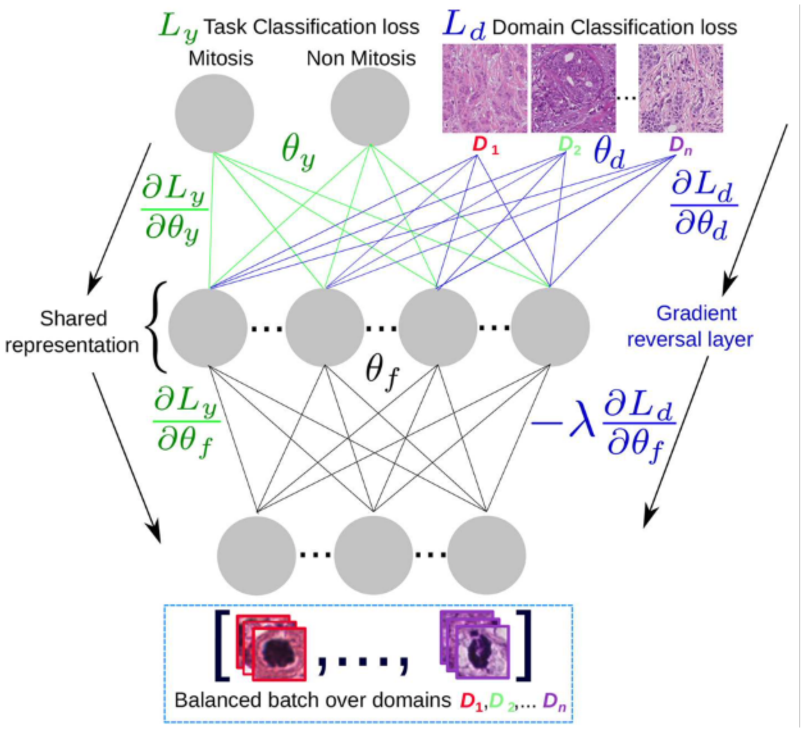

L_y Task Classification loss
Mitosis
Non Mitosis
L_d Domain Classification loss
D_1
D_2
D_n
θ_y
θ_d
$\frac{\partial L_y}{\partial \theta_y}$
$\frac{\partial L_d}{\partial \theta_d}$
Shared representation
Gradient reversal layer
θ_f
$\frac{\partial L_y}{\partial \theta_f}$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
Balanced batch over domains $D_1, D_2, \ldots D_n$

Figure 3: Domain adversarial scheme: A domain-balanced batch of images is passed as input to the network that has two types of outputs: the task classification output and the domain classification output. The shared representation θ_f is optimal for the task classification and unable to discriminate between the n domains.

To explicitly write all the possible variations that lead to changes in the appearance of H&E images is infeasible. Stain invariant training of CNN models aims at detaching the domain or center information, where the changes in appearance originate, from the features that the model learns. In Figure 3, the inner workings of a stain invariant model is demonstrated by showing the flow of gradients in a small neural network example. The CNN has shared features for both the mitosis/no-mitosis classifier and the domain classifier. The main difference with a multi-task CNN is that the gradients from the domain branch are reversed to allow the penalization of unwanted domain information in the features. The stain-invariant model penalizes when the learned features help classify the domain, guiding the model towards features that do not consider the domain information. Evaluation of stain invariant models is described in detail in a previously published journal article from the authors [14].

2.2. Software Functionalities

- **H&E image data augmentation**: This functionality allows generating one or more synthetic, yet realistic, copies of an image region extracted from a H&E WSI. Currently, five augmentation techniques are implemented, as described in section 2.1.2.

- **H&E image normalization**: The second main functionality is to normalize the color of image regions extracted from a H&E WSIs, given a template image. Currently, four normalization techniques are implemented, as described in section 2.1.1.

- **Stain invariant training of CNNs**: This functionality refers to the enabling of training a deep convolutional neural network with H&E images to tackle the stain heterogeneity when the center or source of the images is known. This source-code includes the python implementations of gradient reversal strategies [8, 14] and examples in a CNN model for the classification of H&E images of prostate and breast tissues. The method is explained and illustrated in Section 2.1.3.

3. Visual Examples

To demonstrate the proposed capabilities of the library, we show examples for augmentation and normalization using openly accessible images. The code snippets necessary to reproduce these examples are contained in a jupyter notebook in the source-code repository.

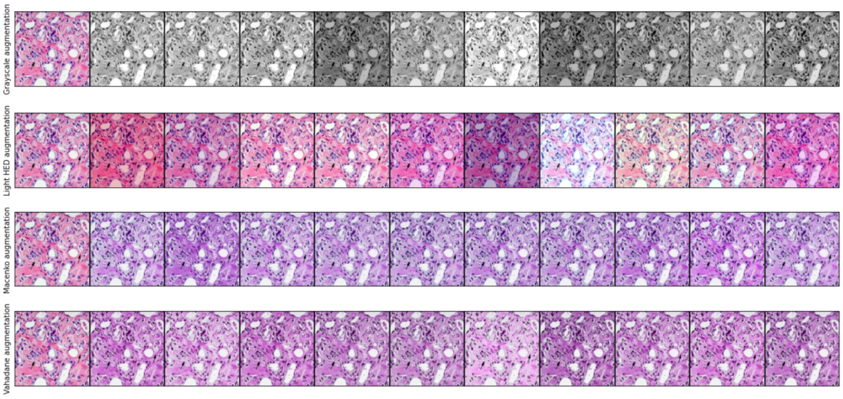


Figure 4: Illustrative examples for the augmentation methods of stainlib.

3.0.1. Image Augmentation

In Figure 4 the first row corresponds to ten grayscale augmented versions of the leftmost image using the method described in section 2.1.2. The second row shows ten augmented images using the light-HED augmentation from Tellez et al. [18]. Finally, the third and fourth rows correspond to the augmentation methods based on Macenko and Vahadane techniques, respectively. The augmented versions are generated perturbing the estimated stains, as in the case for stain normalization. Here the images for both methods look similar.

3.1. Image Normalization

Figure 5 and 6 show the results for the Vahadane and Macenko normalizations methods. Both methods estimate the stain concentration matrix, Vahadane with a non-negative matrix factorization approach and Macenko with a singular value decomposition. Thus, results appear similar with the images normalized with Macenko being slightly darker or with more stain concentration than with the Vahadane method. In Figure 7, examples for the Reinhard normalization method are presented. Because the method aims to match the color histogram of the target image directly, the background matches the lightest color in the target image, which produces unrealistically looking images. We included a tissue detector that masks the tissue content from the background to alleviate this. The tissue detector is included as a parameter in the call for transforming new images using the Reinhard method in `stainlib` [3].

[3] `https://github.com/sebastianffx/stainlib/blob/main/normalization/normalizer.py`

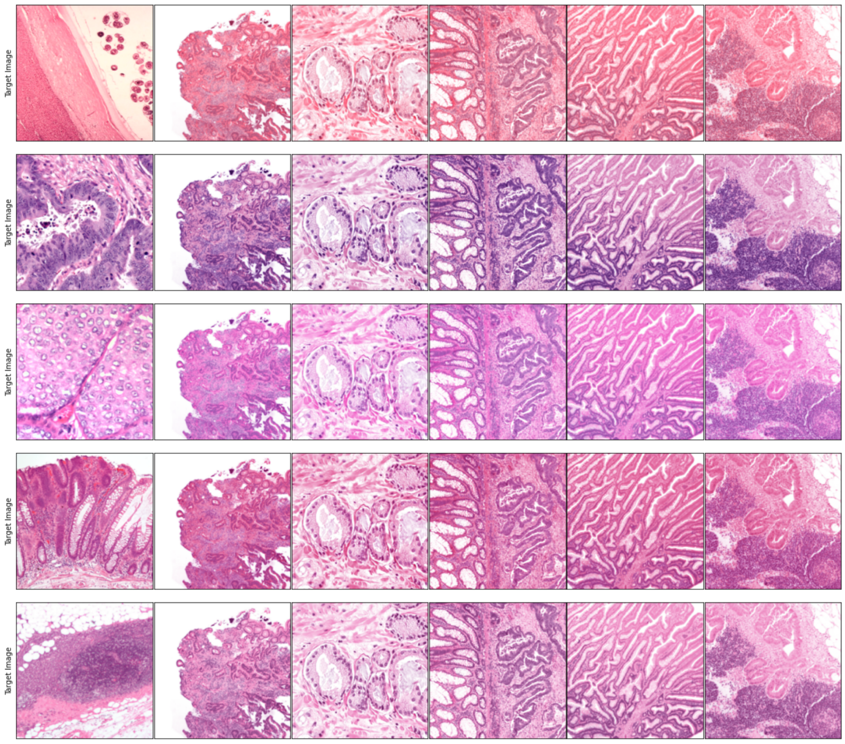


Figure 5: Illustrative examples for the implemented Vahadane normalization method in stainlib.

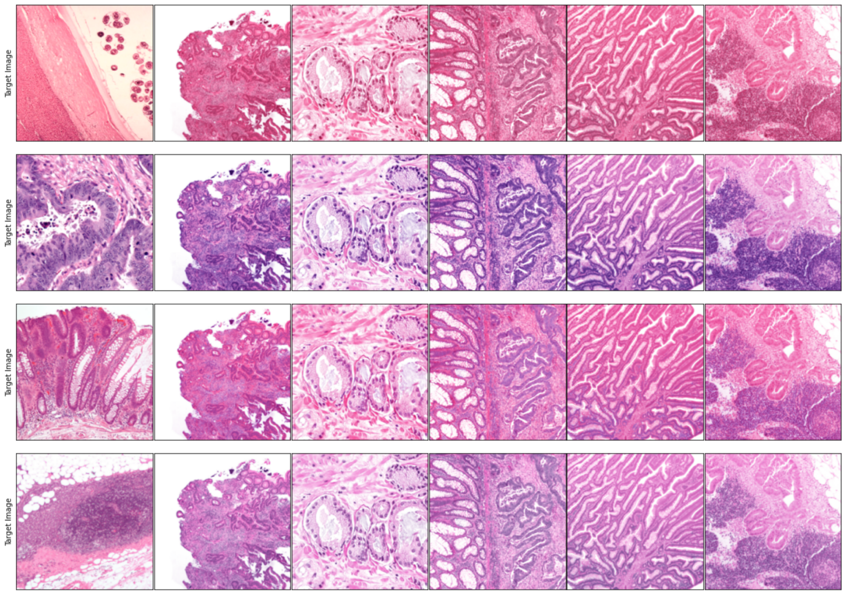


Figure 6: Illustrative examples for the implemented Macenko normalization method in stainlib.

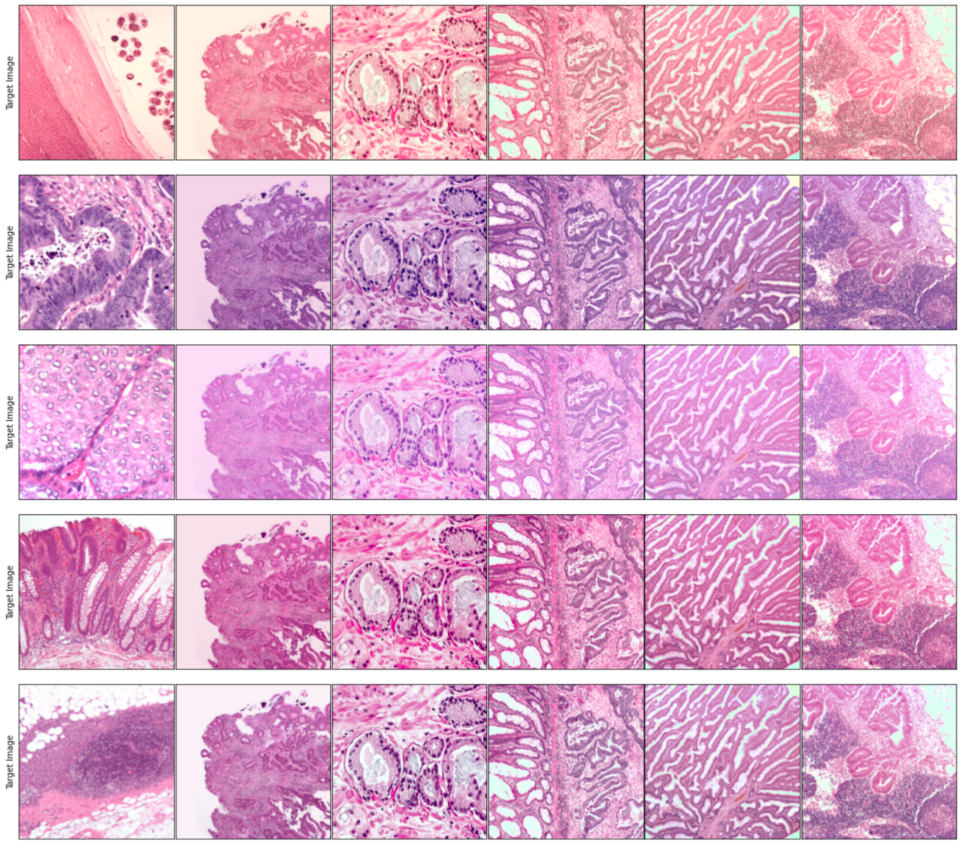


Figure 7: Illustrative examples for the implemented Reinhard normalization method in stainlib.

4. Impact

`stainlib` allows researchers worldwide to tackle scientific challenges in computational pathology more easily in several ways. First, `stainlib` allows testing new algorithms under multiple data-heterogeneity scenarios, given the characteristics of the augmented and normalized images that simulate uncontrolled variations in the real-world test sets. For example, it allows researchers to test an algorithm having specific performance on a given test set on realistic variations of the same dataset generated with `stainlib`, to test if the performance remains the same over the variations or the algorithms requires additional improvement (potentially including such variations of the dataset in the training data). Second, it empowers researchers to use small data sets for training deep learning models by augmenting significantly the data sets with color-shifted versions of the training data. The bigger data-augmented training sets make trained CNN models more robust and less prone to errors. Third, `stainlib` includes very recent image processing and machine learning techniques reported in the literature to deal with color heterogeneity. Finally, the `stainlib`'s modular structure is designed to be easy to expand and the authors encourage researchers to contribute to this library by suggesting changes and improvements or directly adding or improving methods into the code base.

5. Conclusions

This article presents and qualitatively tests the `stainlib` library, implemented to include novel and widely used tools that extract homogeneous representations of heterogeneous color visual information from H&E images. `stainlib` is easy to use and to expand and it includes the most commonly used methods not only for stain normalization but also for augmentation, together with recent deep learning based approaches. We anticipate continuous updates for `stainlib`, making it efficient and useful for normalizing image regions and WSIs. We also strongly encourage the contribution of additional tools by researchers worldwide. The source code for all the tools is now fully accessible in the repositories. We are confident that these resources allow researchers to build more easily and quickly robust models that generalize to unseen images from heterogeneous sources.

6. Conflict of Interest

No conflict of interest exists: We wish to confirm that there are no known conflicts of interest associated with this publication, and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgements

We developed `stainlib` as part of the ExaMode project[4]. This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825292 (ExaMode, `htttp://www.examode.eu/`). Sebastian Otálora thanks Minciencias through the call 756 for PhD studies.

References

[1] Bandi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M., Hermsen, M., Bejnordi, B.E., Lee, B., Paeng, K., Zhong, A., et al.: From detection of individual metastases to classification of lymph node status at the patient level: the camelyon17 challenge. IEEE transactions on medical imaging **38**(2), 550–560 (2018)

[2] Bulten, W., Balkenhol, M., Belinga, J.J.A., Brilhante, A., Çakır, A., Egevad, L., Eklund, M., Farré, X., Geronatsiou, K., Molinié, V., et al.: Artificial intelligence assistance significantly improves gleason grading of prostate biopsies by pathologists. Modern Pathology pp. 1–12 (2020)

[3] Chen, R.T.Q., Behrmann, J., Duvenaud, D., Jacobsen, J.: Residual flows for invertible generative modeling. In: Advances in Neural Information Processing Systems (2019)

[4] Ciompi, F., Geessink, O., Bejnordi, B.E., De Souza, G.S., Baidoshvili, A., Litjens, G., Van Ginneken, B., Nagtegaal, I., Van Der Laak, J.: The importance of stain normalization in colorectal tissue classification with convolutional networks. In: 2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017). pp. 160–163. IEEE (2017)

[5] Echle, A., Rindtorff, N.T., Brinker, T.J., Luedde, T., Pearson, A.T., Kather, J.N.: Deep learning in cancer pathology: a new generation of clinical biomarkers. British Journal of Cancer pp. 1–11 (2020)

[6] Khan, A., Atzori, M., Otálora, S., Andrearczyk, V., Müller, H.: Generalizing convolution neural networks on stain color heterogeneous data for

[4] `https://www.examode.eu/`

computational pathology. In: Medical Imaging 2020: Digital Pathology. vol. 11320, p. 113200R. International Society for Optics and Photonics (2020)

[7] Komura, D., Ishikawa, S.: Machine learning methods for histopathological image analysis. Computational and structural biotechnology journal **16**, 34–42 (2018)

[8] Lafarge, M., Pluim, J., Eppenhof, K., Veta, M.: Learning domain-invariant representations of histological images. Frontiers in medicine **6**, 162 (2019)

[9] Litjens, G., Sánchez, C.I., Timofeeva, N., Hermsen, M., Nagtegaal, I., Kovacs, I., Hulsbergen-Van De Kaa, C., Bult, P., Van Ginneken, B., Van Der Laak, J.: Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. Scientific reports **6**(1), 1–11 (2016)

[10] Macenko, M., Niethammer, M., Marron, J.S., Borland, D., Woosley, J.T., Guan, X., Schmitt, C., Thomas, N.E.: A method for normalizing histology slides for quantitative analysis. In: 2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro. pp. 1107–1110. IEEE (2009)

[11] Marini, N., Atzori, M., Otálora, S., Marchand-Maillet, S., Müller, H.: H&e-adversarial network: a convolutional neural network to learn stain-invariant features through hematoxylin & eosin regression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 601–610 (2021)

[12] Marini, N., Otálora, S., Müller, H., Atzori, M.: Semi-supervised training of deep convolutional neural networks with heterogeneous data and few local annotations: An experiment on prostate histopathology image classification. Medical image analysis **73**, 102165 (2021)

[13] Niazi, M.K.K., Parwani, A.V., Gurcan, M.N.: Digital pathology and artificial intelligence. The lancet oncology **20**(5), e253–e261 (2019)

[14] Otálora, S., Atzori, M., Andrearczyk, V., Khan, A., Müller, H.: Staining invariant features for improving generalization of deep convolutional neural networks in computational pathology. Frontiers in Bioengineering and Biotechnology **7**, 198 (2019)

[15] Otálora, S., Marini, N., Müller, H., Atzori, M.: Semi-weakly supervised learning for prostate cancer image classification with teacher-student

deep convolutional networks. In: Interpretable and Annotation-Efficient Learning for Medical Image Computing, pp. 193–203. Springer (2020)

[16] Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer between images. IEEE Computer graphics and applications **21**(5), 34–41 (2001)

[17] Schmitt, M., Maron, R.C., Hekler, A., Stenzinger, A., Hauschild, A., Weichenthal, M., Tiemann, M., Krahl, D., Kutzner, H., Utikal, J.S., Haferkamp, S., Kather, J.N., Klauschen, F., Krieghoff-Henning, E., Fröhling, S., von Kalle, C., Brinker, T.J.: Hidden variables in deep learning digital pathology and their potential to cause batch effects: Prediction model study. J Med Internet Res **23**(2), e23436 (Feb 2021). https://doi.org/10.2196/23436, `https://www.jmir.org/2021/2/e23436`

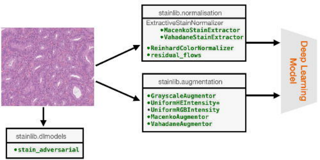
[18] Tellez, D., Balkenhol, M., Otte-Höller, I., van de Loo, R., Vogels, R., Bult, P., Wauters, C., Vreuls, W., Mol, S., Karssemeijer, N., et al.: Whole-slide mitosis detection in h&e breast histology using phh3 as a reference to train distilled stain-invariant convolutional networks. IEEE transactions on medical imaging **37**(9), 2126–2136 (2018)

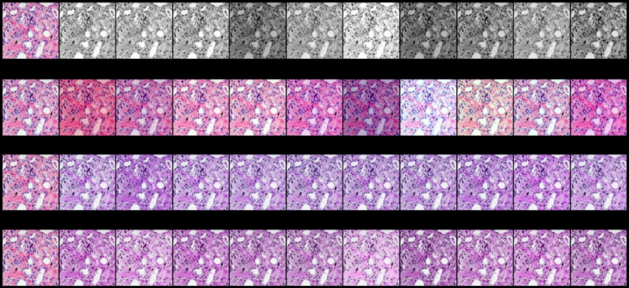
[19] Tellez, D., Litjens, G., Bándi, P., Bulten, W., Bokhorst, J.M., Ciompi, F., van der Laak, J.: Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. Medical image analysis **58**, 101544 (2019)

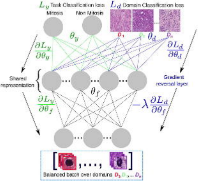
[20] Vahadane, A., Peng, T., Sethi, A., Albarqouni, S., Wang, L., Baust, M., Steiger, K., Schlitter, A.M., Esposito, I., Navab, N.: Structure-preserving color normalization and sparse stain separation for histological images. IEEE transactions on medical imaging **35**(8), 1962–1971 (2016)

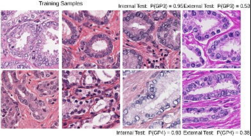
Current executable software version

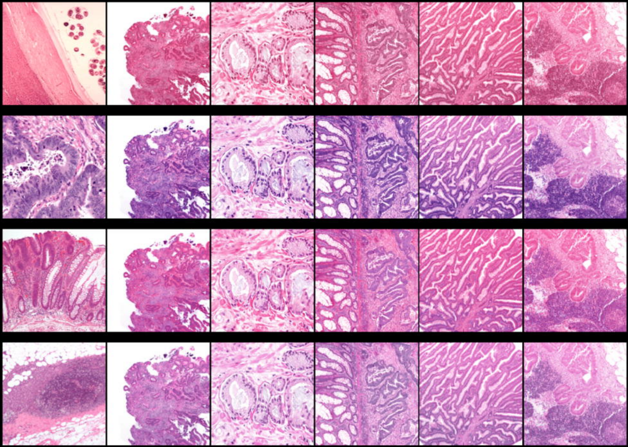
Nr.	**(Executable) software metadata description**	**Please fill in this column**
S1	Current software version	1.0
S2	Permanent link to executables of this version	`https://test.pypi.org/project/stainlib/`
S3	Legal Software License	MIT
S4	Computing platforms/Operating Systems	Linux, OS X, Microsoft Windows, Unix-like
S5	Installation requirements & dependencies	scikit-image, scipy, pillow, opencv-python, spams
S7	Support email for questions	juan.otaloramontenegro@hevs.ch

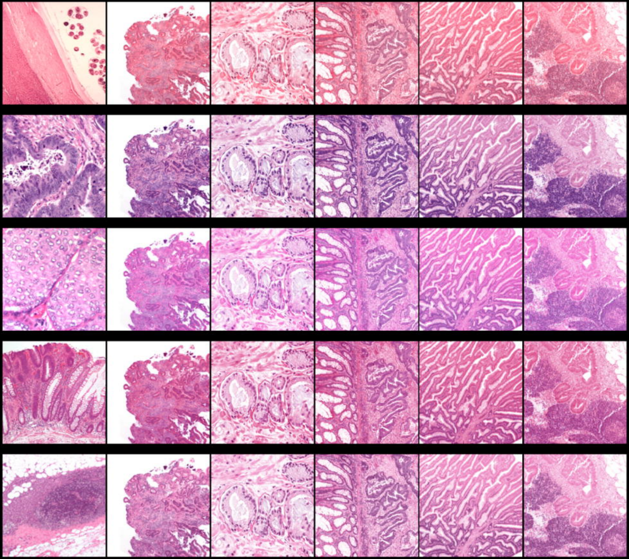

stainlib.normalisation
ExtractiveStainNormalizer
•MacenkoStainExtractor
•VahadaneStainExtractor
•ReinhardColorNormalizer
•residual_flows
stainlib.augmentation
•GrayscaleAugmentor
•UniformHEIntensity*
•UniformRGBIntensity
•MacenkoAugmentor
•VahadaneAugmentor
Deep Learning Model
stainlib.dlmodels
•stain_adversarial

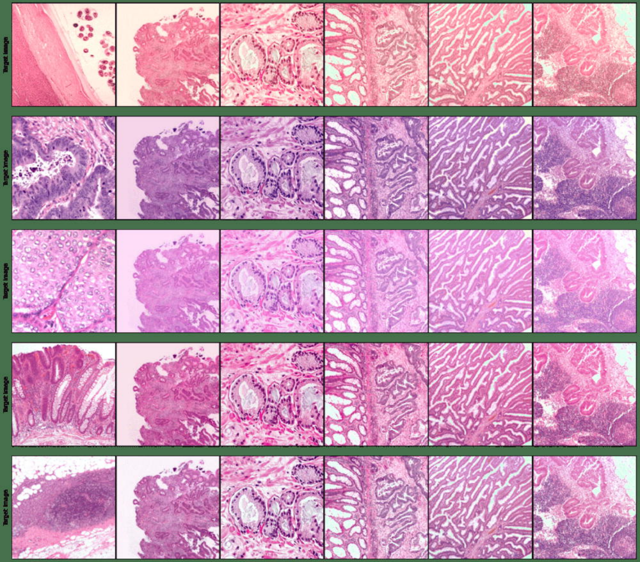



L_y Task Classification loss
L_d Domain Classification loss
Mitosis
Non Mitosis
D_1
D_2
D_n
θ_y
θ_d
$\frac{\partial L_y}{\partial \theta_y}$
$\frac{\partial L_d}{\partial \theta_d}$
Shared representation
Gradient reversal layer
θ_f
$\frac{\partial L_y}{\partial \theta_f}$
$-\lambda \frac{\partial L_d}{\partial \theta_f}$
Balanced batch over domains $D_1, D_2, \ldots D_n$


Training Samples
Internal Test: P(GP3) = 0.95
External Test: P(GP3) = 0.53
Internal Test: P(GP4) = 0.93
External Test: P(GP4) = 0.38






Target image
Target image
Target image
Target image
Target image