

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	<i>https</i> : <i>//github.com/sebastianffx/stainlib</i>
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	<i>juan.otalora@etu.unige.ch</i>

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

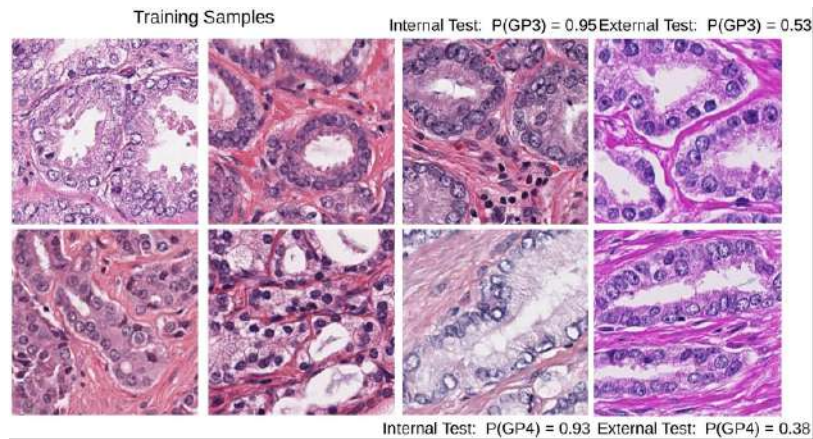


Figure 1: Training images with H&E concentrations that are noticeable different from external test sets can lead to model misclassifications; image taken from [14]. P(GP3) and P(GP4) stands for probability of the image region to contain Gleason patterns 3 and 4 respectively.

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

27 One of the most important factors preventing the application of machine
28 learning methods in clinical practice is related to the heterogeneity of H&E
29 images due to the many parameters involved in the tissue preparation and
30 the digital scanning process (temperature of the tissue, the thickness of the
31 cuts, the image sensor of the digital camera, the stitching algorithm among
32 others) [11]. Figure 1 shows an example of stain variation in training and
33 test sets and its impact on the performance of a CNN model trained only
34 with partial variations. Two approaches are most commonly applied to take
35 into account such variations when training CNN models. First, methods that
36 transform an input H&E image given a target image template are known as
37 "stain normalization". Their aim is to match the input color distributions
38 (or H&E concentrations) with the one given in a target image. The second
39 approach refers to "stain" or "color augmentation methods", which create
40 new synthetic samples to increase the training dataset size, creating more
41 robust models regarding color variations. There are novel image processing

42 and machine learning techniques reported in the literature to deal with color
43 heterogeneity, improving classification, and segmentation performance for
44 various tissue types [19, 6, 14, 4, 11]. While the specific normalization tech-
45 nique depends on the task to solve [19, 4], recent work has reported consistent
46 improvements in performance and robustness to external datasets employing
47 color augmentation techniques [19, 4] or a combination of normalization and
48 augmentation [14].

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

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

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

70 *1.1. Related work*

71 QuPath, Staintools, and HistomicksTK, are likely among the most popu-
72 lar existing software tools to deal with color heterogeneity. QuPath (<https://qupath.github.io/>) is an open and extensible software platform for
73 Whole Slide Image (WSI) analysis. It includes methods for estimating and
74 setting stain vectors. Scripts created for running specific color normaliza-
75 tion methods can also be used within QuPath. Due to the big codebase of
76 QuPath, it is challenging to run a classification or segmentation model with-
77 out having to write a considerable amount of scripts to have a full pipeline,
78 taking into account datasets with considerable color heterogeneity.

79 Staintools is a set of tools for tissue stain normalization and augmentation
80 in python 3. It contains implementations that follow the same coding style
81

82 of scikit-learn, where the methods are made to fit or train a model. It is
83 open-source and can be downloaded from the Github repository: <https://github.com/Peter554/StainTools>. The library contains two extractive
84 normalization methods (Macenko, Vahadane), and the only augmentation
85 techniques included in staintools are based on the same extractive methods
86 by modifying the estimated stain concentrations.

87
88 HistomicksTK is a python toolkit for histopathology image analysis. It
89 contains several methods for stain normalization and color augmentation
90 based on the stain perturbation methods from Tellez et al. [18]. It can
91 be downloaded from <https://pypi.org/project/histomicstk/>. The His-
92 tomicksTK toolkit contains many overlapping methods with `stainlib` but it
93 lacks modularity to use the color tools as standalone modules, which creates
94 difficulties for its usage in different research scenarios. Despite the existence
95 of few libraries, the domain still lacks a modular library including both stan-
96 dard and more modern methods that can be easily evaluated on varying
97 datasets.

98 2. Software description

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

108 2.1. Software Architecture

109 `stainlib` is composed of three main modules `stainlib.augmentation`,
110 `stainlib.normalization`, `stainlib.dlmodels`. The methods and the un-
111 derlying theory of each of the modules are explained in the following subsec-
112 tions.

113 2.1.1. *stainlib.normalization*

114 In digital pathology, the thin slice tissue cuts that are counter-stained in
115 the H&E tissue slides are digitized using digital tissue scanners. The stained
116 tissue slide's light absorbance is quantified and represented in a computer
117 as a two-dimensional digital image (despite coming from a three-dimensional
118 biological structure). In general, if the digital representation of the image

119 is in the RGB color space, each pixel should contain a composition of the
120 color representation of Hematoxylin (*purple*), Eosin(*pink*), and background
121 (*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 :

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

122 Where c ranges in the RGB channels, $S \in [0, +\infty]^{3 \times 2}$ is the matrix of ab-
123 sorbance coefficients, $\mathbf{C} \in [0, +\infty]^2$ is the vector of the two staining concen-
124 tration coefficients and I_0 is the background value.

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

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

137 In `stainlib` we include the Vahadane and Macenko methods using the
138 code base from the implementations in the `staintools` library¹. In the Rein-
139 hard method [16], the color histogram of the source image (in the LAB color

¹Staintools github repository: <https://github.com/Peter554/StainTools/tree/master/staintools>

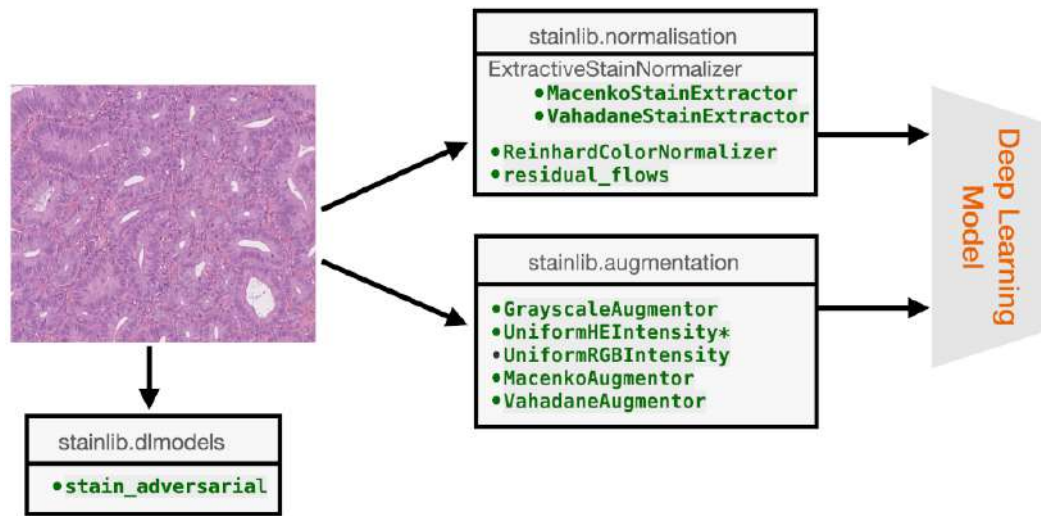


Figure 2: Implemented methods in stainlib for stain normalization and augmentation. The first version of the library includes all the methods in green and linked submodules of the residual flows and stain adversarial methods. For next versions of the library, further state of the art methods will be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

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

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

146 2.1.2. *stainlib.augmentation*

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

²Invertible flows github repository: <https://github.com/sara-nl/color-information>

158 **stainlib** we have included five stain augmentation methods: 1) grayscale
159 transformation, 2) shifts of the stain concentrations in H&E space, 3) shifts
160 of the stain concentrations in RGB space, 4) shifts of the stain concentration
161 matrix using the Macenko method and 5) shifts of the stain concentration
162 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:

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

163 Where the values a_c and b_c are drawn from a uniform distribution in the $[1 -$
164 $\alpha_c, 1 + \alpha_c]$ and $[1 - \beta_c, 1 + \beta_c]$ range. Values of $\alpha = 0.2$ and $\beta_c = 0.2$ were set as
165 default values as they usually yield good qualitative and quantitative results
166 in experiments. In the case of the augmentations based on the shifts of the
167 stain concentrations in H&E space, we followed the implementation of Tellez
168 et al. [18]. Section 3 presents a qualitative evaluation of the normalization
169 and augmentation methods included in **stainlib**. Quantitative evaluation
170 of these methods has also been performed in previous research from the
171 authors [14, 6, 19].

172 2.1.3. *stainlib.dlmodels*

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

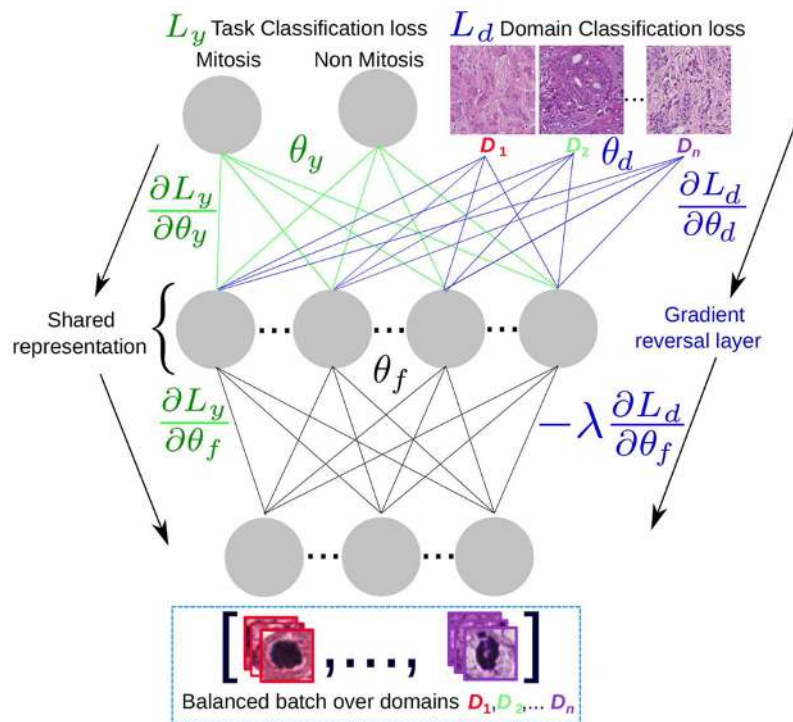


Figure 3: Domain adversarial scheme: A domain-balanced batch of images is passed as input to the network that has two types of outputs: the task classification output and the domain classification output. The shared representation θ_f is optimal for the task classification and unable to discriminate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

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

196 *2.2. Software Functionalities*

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

- 201 • **H&E image normalization:** The second main functionality is to nor-
202 malize the color of image regions extracted from a H&E WSIs, given
203 a template image. Currently, four normalization techniques are imple-
204 mented, as described in section 2.1.1.

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

212 **3. Visual Examples**

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

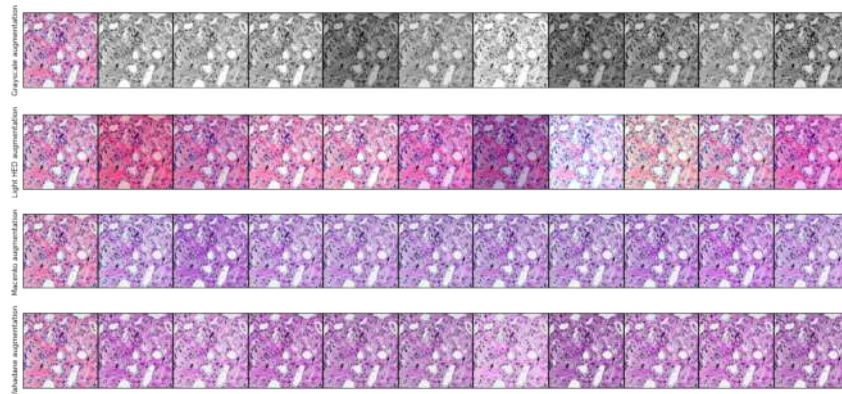


Figure 4: Illustrative examples for the augmentation methods of stainlib.

217 3.0.1. Image Augmentation

218 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
219 second row shows ten augmented images using the light-HED augmentation from Tellez et al. [18]. Finally, the third and fourth rows correspond to
220 the augmentation methods based on Macenko and Vahadane techniques, respectively. The augmented versions are generated perturbing the estimated
221 stains, as in the case for stain normalization. Here the images for both methods look similar.
222
223
224
225

226 3.1. Image Normalization

227 Figure 5 and 6 show the results for the Vahadane and Macenko normalizations methods. Both methods estimate the stain concentration matrix,
228 Vahadane with a non-negative matrix factorization approach and Macenko with a singular value decomposition. Thus, results appear similar with the
229 images normalized with Macenko being slightly darker or with more stain concentration than with the Vahadane method. In Figure 7, examples for
230 the Reinhard normalization method are presented. Because the method aims to match the color histogram of the target image directly, the background
231 matches the lightest color in the target image, which produces unrealistically looking images. We included a tissue detector that masks the tissue content
232 from the background to alleviate this. The tissue detector is included as a parameter in the call for transforming new images using the Reinhard
233 method in `stainlib`³.
234
235
236
237
238
239

³<https://github.com/sebastianffx/stainlib/blob/main/normalization/normalizer.py>

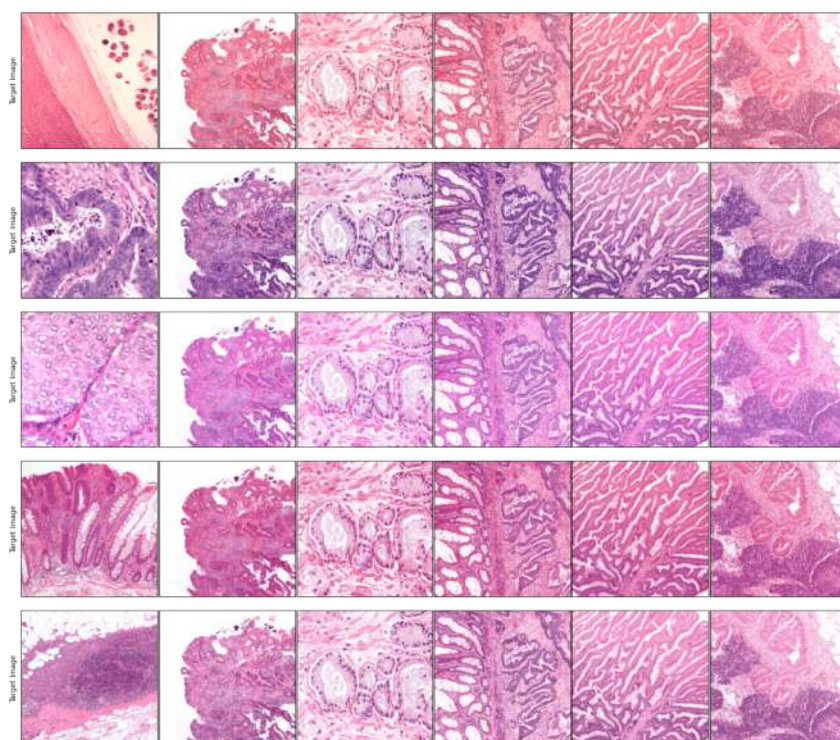


Figure 5: Illustrative examples for the implemented Vahadane normalization method in stainlib.

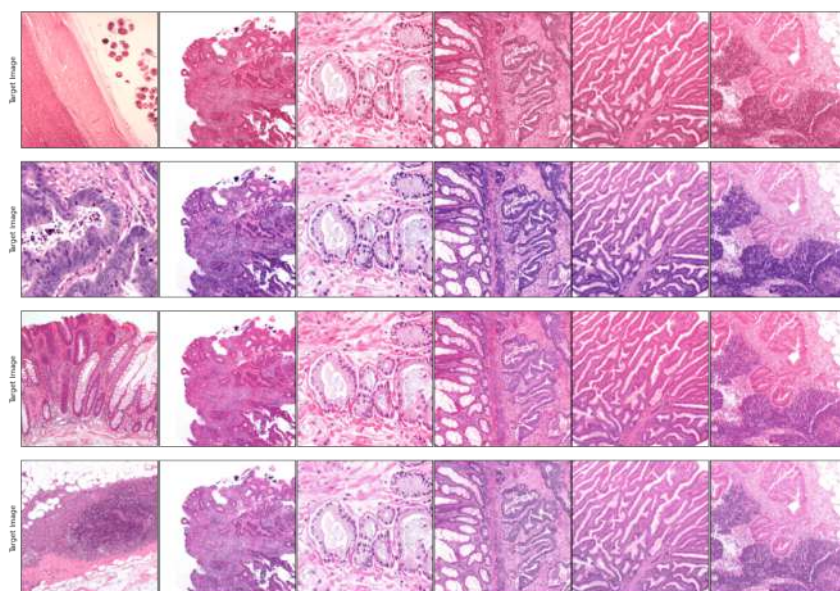


Figure 6: Illustrative examples for the implemented Macenko normalization method in stainlib.

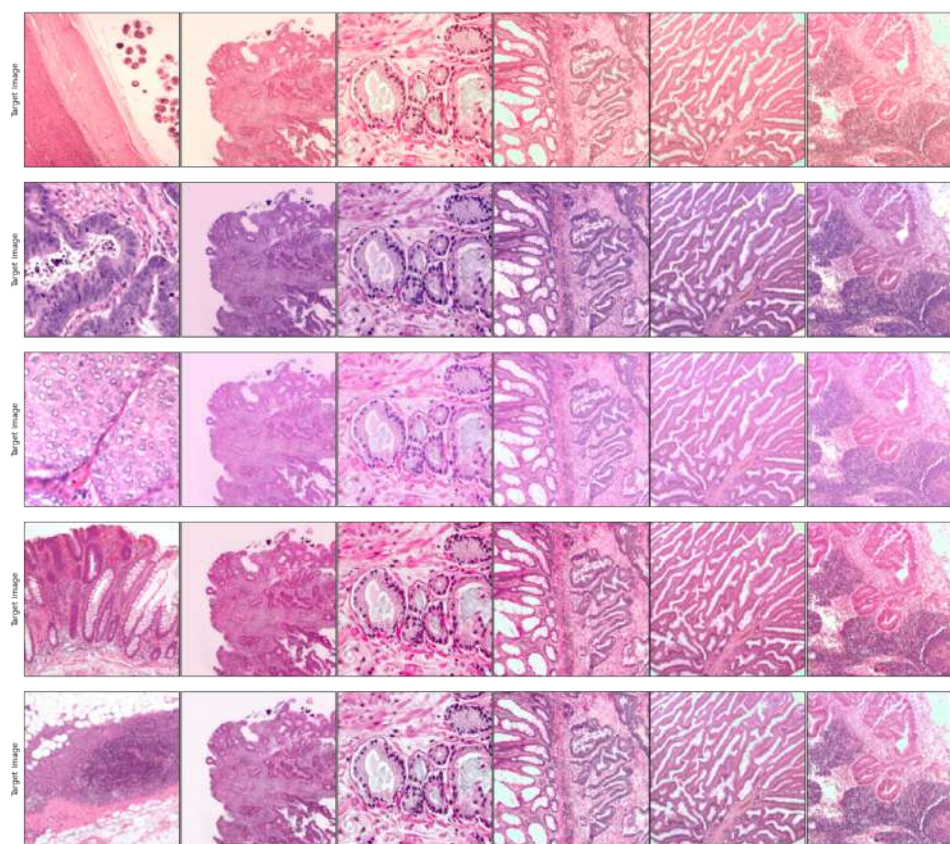


Figure 7: Illustrative examples for the implemented Reinhard normalization method in stainlib.

240 4. Impact

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

260 5. Conclusions

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

273 6. Conflict of Interest

274 No conflict of interest exists: We wish to confirm that there are no known
275 conflicts of interest associated with this publication, and there has been no

276 significant financial support for this work that could have influenced its out-
277 come.

278 Acknowledgements

279 We developed `stainlib` as part of the ExaMode project⁴. This project
280 has received funding from the European Union’s Horizon 2020 research and
281 innovation programme under grant agreement No 825292 (ExaMode, <http://www.examode.eu/>). Sebastian Otálora thanks Minciencias through the call
282 //www.examode.eu/. Sebastian Otálora thanks Minciencias through the call
283 756 for PhD studies.

284 References

- 285 [1] Bandi, P., Geessink, O., Manson, Q., Van Dijk, M., Balkenhol, M.,
286 Hermsen, M., Bejnordi, B.E., Lee, B., Paeng, K., Zhong, A., et al.:
287 From detection of individual metastases to classification of lymph node
288 status at the patient level: the camelyon17 challenge. *IEEE transactions*
289 *on medical imaging* **38**(2), 550–560 (2018)
- 290 [2] Bulten, W., Balkenhol, M., Belinga, J.J.A., Brilhante, A., Çakır, A.,
291 Egevad, L., Eklund, M., Farré, X., Geronatsiou, K., Molinié, V., et al.:
292 Artificial intelligence assistance significantly improves gleason grading
293 of prostate biopsies by pathologists. *Modern Pathology* pp. 1–12 (2020)
- 294 [3] Chen, R.T.Q., Behrmann, J., Duvenaud, D., Jacobsen, J.: Residual
295 flows for invertible generative modeling. In: *Advances in Neural Infor-*
296 *mation Processing Systems* (2019)
- 297 [4] Ciompi, F., Geessink, O., Bejnordi, B.E., De Souza, G.S., Baidoshvili,
298 A., Litjens, G., Van Ginneken, B., Nagtegaal, I., Van Der Laak, J.: The
299 importance of stain normalization in colorectal tissue classification with
300 convolutional networks. In: *2017 IEEE 14th International Symposium*
301 *on Biomedical Imaging (ISBI 2017)*. pp. 160–163. IEEE (2017)
- 302 [5] Echle, A., Rindtorff, N.T., Brinker, T.J., Luedde, T., Pearson, A.T.,
303 Kather, J.N.: Deep learning in cancer pathology: a new generation of
304 clinical biomarkers. *British Journal of Cancer* pp. 1–11 (2020)
- 305 [6] Khan, A., Atzori, M., Otálora, S., Andrearczyk, V., Müller, H.: Gener-
306 alizing convolution neural networks on stain color heterogeneous data for

⁴<https://www.examode.eu/>

- 307 computational pathology. In: *Medical Imaging 2020: Digital Pathology*.
308 vol. 11320, p. 113200R. International Society for Optics and Photonics
309 (2020)
- 310 [7] Komura, D., Ishikawa, S.: Machine learning methods for histopathologi-
311 cal image analysis. *Computational and structural biotechnology journal*
312 **16**, 34–42 (2018)
- 313 [8] Lafarge, M., Pluim, J., Eppenhof, K., Veta, M.: Learning domain-
314 invariant representations of histological images. *Frontiers in medicine*
315 **6**, 162 (2019)
- 316 [9] Litjens, G., Sánchez, C.I., Timofeeva, N., Hermsen, M., Nagtegaal, I.,
317 Kovacs, I., Hulsbergen-Van De Kaa, C., Bult, P., Van Ginneken, B., Van
318 Der Laak, J.: Deep learning as a tool for increased accuracy and effi-
319 ciency of histopathological diagnosis. *Scientific reports* **6**(1), 1–11 (2016)
- 320 [10] Macenko, M., Niethammer, M., Marron, J.S., Borland, D., Woosley,
321 J.T., Guan, X., Schmitt, C., Thomas, N.E.: A method for normalizing
322 histology slides for quantitative analysis. In: *2009 IEEE International
323 Symposium on Biomedical Imaging: From Nano to Macro*. pp. 1107–
324 1110. IEEE (2009)
- 325 [11] Marini, N., Atzori, M., Otálora, S., Marchand-Maillet, S., Müller, H.:
326 H&E-adversarial network: a convolutional neural network to learn stain-
327 invariant features through hematoxylin & eosin regression. In: *Proceed-
328 ings of the IEEE/CVF International Conference on Computer Vision*.
329 pp. 601–610 (2021)
- 330 [12] Marini, N., Otálora, S., Müller, H., Atzori, M.: Semi-supervised train-
331 ing of deep convolutional neural networks with heterogeneous data and
332 few local annotations: An experiment on prostate histopathology image
333 classification. *Medical image analysis* **73**, 102165 (2021)
- 334 [13] Niazi, M.K.K., Parwani, A.V., Gurcan, M.N.: Digital pathology and
335 artificial intelligence. *The lancet oncology* **20**(5), e253–e261 (2019)
- 336 [14] Otálora, S., Atzori, M., Andrearczyk, V., Khan, A., Müller, H.: Stain-
337 ing invariant features for improving generalization of deep convolutional
338 neural networks in computational pathology. *Frontiers in Bioengineering
339 and Biotechnology* **7**, 198 (2019)
- 340 [15] Otálora, S., Marini, N., Müller, H., Atzori, M.: Semi-weakly supervised
341 learning for prostate cancer image classification with teacher-student

- 342 deep convolutional networks. In: *Interpretable and Annotation-Efficient*
343 *Learning for Medical Image Computing*, pp. 193–203. Springer (2020)
- 344 [16] Reinhard, E., Adhikhmin, M., Gooch, B., Shirley, P.: Color transfer
345 between images. *IEEE Computer graphics and applications* **21**(5), 34–
346 41 (2001)
- 347 [17] Schmitt, M., Maron, R.C., Hekler, A., Stenzinger, A., Hauschild,
348 A., Weichenthal, M., Tiemann, M., Krahl, D., Kutzner, H., Utikal,
349 J.S., Haferkamp, S., Kather, J.N., Klauschen, F., Krieghoff-Henning,
350 E., Fröhling, S., von Kalle, C., Brinker, T.J.: Hidden variables in
351 deep learning digital pathology and their potential to cause batch ef-
352 fects: Prediction model study. *J Med Internet Res* **23**(2), e23436 (Feb
353 2021). <https://doi.org/10.2196/23436>, [https://www.jmir.org/2021/](https://www.jmir.org/2021/2/e23436)
354 [2/e23436](https://www.jmir.org/2021/2/e23436)
- 355 [18] Tellez, D., Balkenhol, M., Otte-Höller, I., van de Loo, R., Vogels, R.,
356 Bult, P., Wauters, C., Vreuls, W., Mol, S., Karssemeijer, N., et al.:
357 Whole-slide mitosis detection in h&e breast histology using phh3 as a
358 reference to train distilled stain-invariant convolutional networks. *IEEE*
359 *transactions on medical imaging* **37**(9), 2126–2136 (2018)
- 360 [19] Tellez, D., Litjens, G., Bándi, P., Bulten, W., Bokhorst, J.M., Ciompi,
361 F., van der Laak, J.: Quantifying the effects of data augmentation and
362 stain color normalization in convolutional neural networks for computa-
363 tional pathology. *Medical image analysis* **58**, 101544 (2019)
- 364 [20] Vahadane, A., Peng, T., Sethi, A., Albarqouni, S., Wang, L., Baust,
365 M., Steiger, K., Schlitter, A.M., Esposito, I., Navab, N.: Structure-
366 preserving color normalization and sparse stain separation for histolog-
367 ical images. *IEEE transactions on medical imaging* **35**(8), 1962–1971
368 (2016)

369 **Current executable software version**

Nr.	(Executable) software meta-data 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



slide_preproc

- obtain_active_frame

slide_normalization

Extract features from normalized slide

- ResNet50 for slide feature
- ResNet50 for slide feature

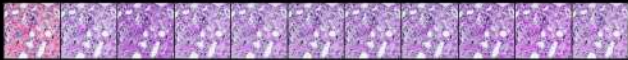
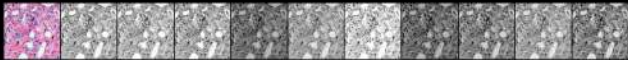
• ResNet50 for Normal Slide

• residual_flow

slide_augmentation

- Geometric Augmentation
- Color Augmentation
- Color Augmentation
- Geometric Augmentation
- Geometric Augmentation

Deep Learning



\hat{L}_1 Node Classification Loss

EMNIST

SVN dataset

\hat{L}_2 Sparse Classification Loss

