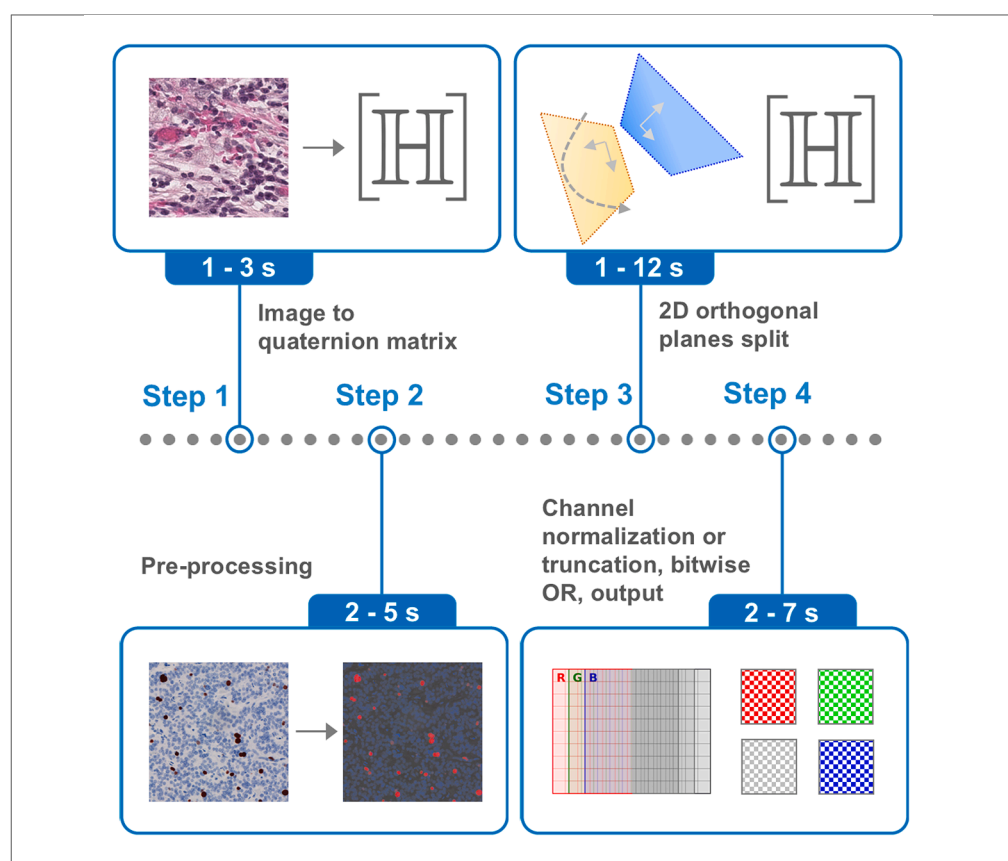


Protocol

Protocol for natural and biomedical image processing in the hypercomplex domain using the 2D orthogonal planes split



We present a protocol for processing natural and biomedical images in the hypercomplex domain. First, the protocol converts images into quaternion matrices. It then applies the 2D orthogonal planes split to perform re-colorization, de-colorization, contrast enhancement, re-staining, and stain separation. Finally, it uses post-processing operations to generate the output. The resulting output supports qualitative assessment and downstream processing in computer vision and computational pathology pipelines.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Nektarios A. Valous,
Eckhard Hitzer,
Dragoş Duşe, ...,
Niels Halama, Inka
Zörnig, Dirk Jäger

nek.valous@
nct-heidelberg.de

Highlights

Encode image as a pure quaternion matrix for workflow processing

Use orthogonal planes split to decompose pixels into steerable components

Generate transformed color or grayscale images

Apply the protocol for five different tasks on natural and biomedical images

Valous et al., STAR Protocols 7, 104548

June 19, 2026 © 2026 The Author(s). Published by Elsevier Inc.

<https://doi.org/10.1016/j.xpro.2026.104548>



Protocol

Protocol for natural and biomedical image processing in the hypercomplex domain using the 2D orthogonal planes split

Nektarios A. Valous,^{1,2,3,20,21,*} Eckhard Hitzler,⁴ Dragoş Duşe,^{5,6} Rodrigo Rojas Moraleda,^{1,3} Ferdinand Popp,^{1,7} Meggy Suarez-Carmona,^{8,9,10,11} Anna Berthel,^{8,9} Ismini Papageorgiou,^{12,13} Carlo Fremd,^{2,5,14,15} Alexander Rölle,^{1,2,5} Christina C. Westhoff,¹⁶ Bénédicte Lenoir,^{1,17} Niels Halama,^{8,9,18,19} Inka Zörnig,^{1,2,5} and Dirk Jäger^{1,2,3,5}

¹Applied Tumor Immunity Clinical Cooperation Unit, National Center for Tumor Diseases (NCT) Heidelberg, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 460, Heidelberg 69120, Germany

²Medical Faculty Heidelberg, Heidelberg University, Department of Medical Oncology, Heidelberg University Hospital (UKHD), Im Neuenheimer Feld 672, Heidelberg 69120, Germany

³Center for Quantitative Analysis of Molecular and Cellular Biosystems (Bioquant), Heidelberg University, Im Neuenheimer Feld 267, Heidelberg 69120, Germany

⁴College of Liberal Arts, International Christian University, Osawa 3-10-2, Mitaka, Tokyo 181-8585, Japan

⁵Department of Medical Oncology, National Center for Tumor Diseases (NCT) Heidelberg, Heidelberg University Hospital (UKHD), Im Neuenheimer Feld 460, Heidelberg 69120, Germany

⁶Synaptiq, Str. Clinicilor 23, Cluj-Napoca 400006, Romania

⁷Division of Applied Bioinformatics German Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, Heidelberg 69120, Germany

⁸Department of Cancer Immunology and Cancer Immunotherapy, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, Heidelberg 69120, Germany

⁹Tumor Immunology & Tumor Immunotherapy Group, Helmholtz Institute for Translational Oncology (HI-TRON), Obere Zahlbacher Straße 63, Building 911, Mainz 55131, Germany

¹⁰Molecular Immunology Laboratory, Institute Jules Bordet, Université Libre de Bruxelles (ULB), Rue Meylenmeersch 90, 1070 Brussels, Belgium

¹¹Institute of Medical Immunology (IMI) and Immunobiology Laboratory, ULB Center for Research in Immunology (U-CRI), Rue Adrienne Bolland 10 Bte CP 300, 6041 Gosselies, Belgium

¹²Institute of Radiology, Südharz Hospital Nordhausen, Dr.-Robert-Koch-Straße 39, 99734 Nordhausen, Germany

¹³Institute of Diagnostic and Interventional Radiology, Jena University Hospital, Am Klinikum 1, 07747 Jena, Germany

¹⁴Division of Gynecological Oncology, National Center for Tumor Diseases (NCT) Heidelberg, Heidelberg University Hospital (UKHD), Im Neuenheimer Feld 460, 69120 Heidelberg, Germany

¹⁵Division of Molecular Genetics, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany

¹⁶Institute of Pathology, Philipps University of Marburg and University Hospital Giessen and Marburg GmbH (UKGM), Baldingerstrasse, 35033 Marburg, Germany

¹⁷Systems Immunology and Single-Cell Biology Group, German Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany

¹⁸University Center for Tumor Diseases Mainz (UCT Mainz), University Medical Center of Johannes Gutenberg University Mainz, Langenbeckstraße 1, 55131 Mainz, Germany

¹⁹Department of Hematology and Medical Oncology, III. Medical Clinic and Polyclinic, University Medical Center of Johannes Gutenberg University Mainz, Langenbeckstraße 1, 55131 Mainz, Germany

²⁰Technical contact

²¹Lead contact

*Correspondence: nek.valous@nct-heidelberg.de
<https://doi.org/10.1016/j.xpro.2026.104548>

SUMMARY

We present a protocol for processing natural and biomedical images in the hypercomplex domain. First, the protocol converts images into quaternion matrices. It then applies the 2D orthogonal planes split to perform re-colorization,



de-colorization, contrast enhancement, re-staining, and stain separation. Finally, it uses post-processing operations to generate the output. The resulting output supports qualitative assessment and downstream processing in computer vision and computational pathology pipelines.

For details and literature comparisons, please refer to Valous et al.¹

BEFORE YOU BEGIN

Working with images in a hypercomplex domain makes it possible to construct richer, more intuitive representations whose algebraic structure and properties can yield new insights and optimizations.^{2,3} Quaternions are hypercomplex numbers and form one of the four existing normed division algebras: real, complex, quaternions, and octonions.⁴ Real and complex numbers are used for one-dimensional and two-dimensional data, whereas quaternions can perform algebraic operations on ternary quantities and can directly encode color data.⁵ For reference, a pure quaternion is a quaternion with a zero real part (e.g., $q = 0 + bi + cj + dk$) and a pure unit quaternion is a pure quaternion that also satisfies the unit norm condition (e.g., $\|q\| = \sqrt{0^2 + b^2 + c^2 + d^2} = 1$). Based on a hypercomplex framework called the two-dimensional (2D) orthogonal planes split (OPS), which allows the splitting of a quaternion (representing a pixel) into pairs of orthogonal two-dimensional planes,^{6,7} we outline an image processing protocol for natural and biomedical image re-colorization, natural image de-colorization, computational re-staining for biomedical images, natural and biomedical image contrast enhancement, and stain separation for biomedical images (Figure 1), in easy to follow step-by-step descriptions. To carry out the 2D OPS, we are utilizing either: a) pure unit quaternions μ_1 to μ_{13} (Figure 2) that derive from pure quaternions, respectively: $(i), (j), (k), (i+j), (i+k), (j+k), (i+j+k), (-i+j), (-i+k), (-j+k), (-i+j+k), (i-j+k), (i+j-k)$, or b) values represented as pure quaternions or pure unit quaternions that are determined by the user (e.g., pixel or area sampling) or computed by an automated technique. As an example, quaternions can have geometric meaning in the RGB (red, green, blue) color space, e.g., μ_1, μ_2 , and μ_3 refer to the red, green, and blue color axes respectively, while μ_7 is the direction corresponding to the luminance or gray-line axis.^{8,9} All essential definitions, equations, and mathematical descriptions to facilitate a clear understanding of subject matter are provided in the methods section (quaternion algebra, 2D orthogonal planes split, quaternion matrix representation) of Valous et al.¹

In order to expound the concept of the 2D OPS, a quaternion can be viewed as a vector in four-dimensional space.^{6,7} The OPS writes this quaternion as $q = q_+ + q_-$ where q_+ and q_- are not arbitrary algebraic parts, but components of q that lie in two different and mutually orthogonal 2D planes.^{6,7} In the general form: $q_{\pm} = 1/2(q \pm f g q)$ where f and g are pure unit quaternions.^{6,7} Therefore, instead of treating a quaternion as one indivisible 4D object, 2D OPS reorganizes it into two simpler pieces that live in two separate 2D planes inside the same 4D space. These two planes are orthogonal, so they do not overlap or mix with each other. That is why the split is meaningful geometrically: it is not just breaking a formula into two terms, but separating the quaternion into two independent geometric components whose “energies” add back to the original whole. The key geometric property is that the map $f()g$ leaves q_+ unchanged but changes the sign of q_- .^{6,7} This means that the transform acts like a 180° half-turn of one plane around the other, which is why the decomposition has a geometric interpretation rather than being only a formal algebraic split.^{6,7} Therefore, the transformation defined by the chosen quaternions acts differently on these two components in a very structured way. One of the two planes is left unchanged, while the component in the other plane is reversed in sign. Geometrically, this behaves like a 180° half-turn; one plane plays the role of a fixed reference plane, and the component in the other plane is flipped relative to it. So the 2D OPS is useful because it reveals which part of the quaternion stays aligned with the chosen geometric setting and which part lies in the complementary plane. For a pixel encoded as a pure quaternion, the split separates it into two complementary geometric components defined by the chosen steering quaternions

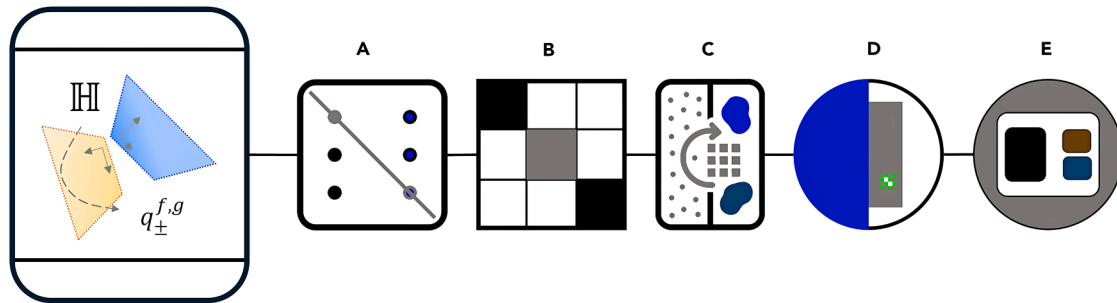


Figure 1. A single hypercomplex algebraic tool, the 2-D orthogonal-planes split (OPS), drives five image operations utilized in computer vision and computational pathology

(A) alternative chromatic representations (re-colorization), B) shifts to texture, contrast and light (de-colorization), C) targeted chromatic alterations (re-staining), D) visual content improvements (contrast enhancement), and E) partitions of contextual stain components (stain separation).

f and g .^{6,7} These components are orthogonal and together contain all information of the original pixel.^{6,7} In the special case where the steering direction is the gray-line, the split has the familiar interpretation of separating luminance-related content from chrominance-related content.^{6,7} In the general case, it provides a steerable decomposition into two analysis planes chosen to emphasize specific image structures or color mixtures.^{6,7} Therefore, once a color pixel is encoded as a quaternion, the 2D OPS does not split it into arbitrary algebraic pieces, but into two complementary geometric parts. These two parts together contain the full pixel information, but they are now organized into two orthogonal components. That makes it easier to analyze or manipulate specific aspects of the pixel content, depending on how the analysis planes are chosen. As an additional point, OPS is steerable in both directions.⁶ On the one hand, any chosen pair of f and g defines a split into two orthogonal 2D planes.^{6,7} On the other hand, if one starts from a pre-determined pair of orthogonal analysis planes, then a corresponding pair of f and g can be determined that reproduces exactly this split.⁶

In principle, there is no limit on how many images this approach can analyze; the practical bottleneck is the size of each image. Linear-time algorithms are fast but still face scalability limits. Because the 2D OPS runs in time that grows at most linearly with the number of input pixels, and since system memory is finite, larger images and near real-time use cases call for further optimizations, e.g., image tiling, parallel pixel processing. Table 1 summarizes the symbols, variables, and operators used throughout the protocol.

Innovation

This protocol consolidates five image processing tasks (re-colorization, de-colorization, contrast enhancement, computational re-staining, and stain separation) utilized in computer vision and computational pathology into one rigorously defined concept. We represent each color pixel as a pure quaternion and apply the orthogonal planes split to decompose it into two 2D planes. A steerable mapping (parameterized by quaternions) executes all five tasks by mainly varying the steering parameters followed by post-processing steps. Using basic arithmetic and matrix operations, this

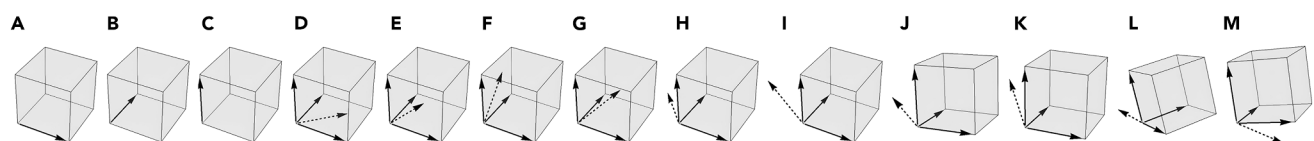


Figure 2. Pure unit quaternions μ_1 to μ_{13}

(A) $\mu_1 = i$, B) $\mu_2 = j$, C) $\mu_3 = k$, D) $\mu_4 = (i + j) / \sqrt{2}$, E) $\mu_5 = (i + k) / \sqrt{2}$, F) $\mu_6 = (j + k) / \sqrt{2}$, G) $\mu_7 = (i + j + k) / \sqrt{3}$, H) $\mu_8 = (-i + j) / \sqrt{2}$, I) $\mu_9 = (-i + k) / \sqrt{2}$, J) $\mu_{10} = (-j + k) / \sqrt{2}$, K) $\mu_{11} = (-i + j + k) / \sqrt{3}$, L) $\mu_{12} = (i - j + k) / \sqrt{3}$, and M) $\mu_{13} = (i + j - k) / \sqrt{3}$.

Table 1. Protocol notation: main symbols, variables, and operators used throughout the protocol

Symbol/variable/operator	Definitions and comments
I_C	Input color image.
I_q	Pure quaternion matrix representation of the input image.
I_T	Transformed color image output.
I_G	Transformed greyscale image output.
$q(n,m)$	Quaternion-valued pixel at image coordinates (n,m) .
$r(n,m), g(n,m), b(n,m)$	Red, green, and blue components of a pixel.
i, j, k	Quaternion imaginary basis units used to encode RGB channels.
TI_q	Transformed quaternion matrix after the pre-processing step.
TI_{nq}	Normalized transformed quaternion matrix used in later OPS steps.
$\mu_1 \sim \mu_{13}$	Predefined range of pure unit quaternions.
s_1, s_2	Pure quaternions derived from the stain-separation matrix.
f_1, f_2	User-defined pure quaternions for the two re-staining branches.
f, g	Pure quaternions or pure unit quaternions used to steer the OPS.
$q_+^{(f,f)}$	Positive OPS component with respect to f .
$q_-^{(f,f)}$	Negative OPS component with respect to f .
$q_+^{(f,g)}$	Positive OPS component with respect to f and g .
$q_-^{(f,g)}$	Negative OPS component with respect to f and g .
ccA, ccB, ccC	Extracted color channel matrices from a quaternion matrix.
I_{ccA}, I_{ccB}	Quaternion matrix representations of selected color channels.
ccA_n, ccB_n	Linearly scaled or normalized channel matrices.
$ccAB_n$	Combined output channel after bitwise OR.
$\alpha, \beta, \gamma, \delta$	Parameters controlling the contrast-enhancement expression.
X	Upper intensity bound used in linear histogram scaling.
$\otimes$	Quaternion multiplication operator as written in the manuscript.
$\vee$	Bitwise OR operator.
$[0, 1]$	Range used for normalization or truncation.

hypercomplex framework becomes the basis of several computational workflows that share a common backbone, thus replacing a patchwork of task-specific approaches. The approach is interpretable and data-light; it requires no model training, provides explicit control of hue-luminance interactions, and yields deterministic, reproducible transformations with well-defined parameters. Because, virtually, the same framework governs all five tasks, color handling is precise and consistent across the workflows, improving comparability and simplifying validation. The protocol scales linearly with image size, and supports tile-wise processing as well as parallel execution on CPU and GPU backends. In summary, a mathematically coherent framework in the hypercomplex domain, replaces a patchwork of techniques and delivers unified color/contrast control and efficient execution across natural and biomedical imagery.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Software and algorithms		
MATLAB and Python code	This paper	https://doi.org/10.6084/m9.figshare.29683382
Workflow #1 Re-colorization	This paper	Files: <i>recolorization.m</i> and <i>recolorization.py</i>
Workflow #2 De-colorization	This paper	Files: <i>decolorization.m</i> and <i>decolorization.py</i>
Workflow #3 Contrast enhancement	This paper	Files: <i>contrast_enhancement.m</i> and <i>contrast_enhancement.py</i>
Workflow #4 Stain separation	This paper	Files: <i>stain_separation.m</i> and <i>stain_separation.py</i>
Workflow #5 Re-staining	This paper	Files: <i>restaining.m</i> and <i>restaining.py</i>

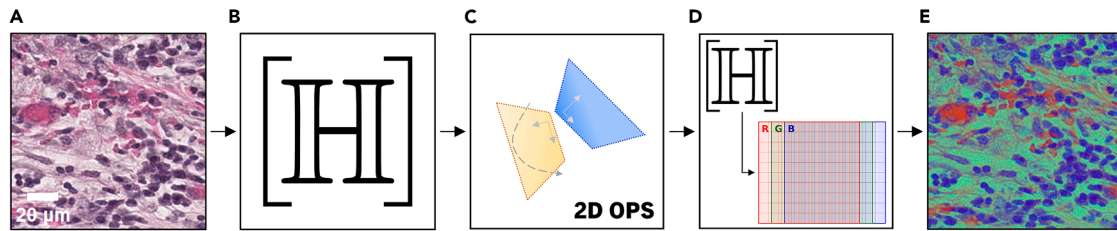


Figure 3. Computational re-colorization workflow (#1)

(A) input a histological image, (B) represent it as a quaternion matrix, (C) carry out the 2D orthogonal planes split (Equation 2, $f=g=\mu_5$), (D) extract and normalize (range [0, 1]) color channel matrices from the resulting quaternion matrix, and (E) concatenate the color channels to obtain the transformed color image.

STEP-BY-STEP METHOD DETAILS

Computational workflows #1–5 are visualized in Figures 3, 4, 5, 6, and 7 (respectively, re-colorization, de-colorization, contrast enhancement, stain separation, and re-staining).

Image to quaternion matrix (workflows #1–5)

⌚ Timing: 1–3 s per image (depending on size)

Here, we describe the step pertinent to converting an input color image to a quaternion matrix which applies to all workflows.

1. Convert a color (RGB) image I_C to a pure quaternion matrix I_q by placing the R, G, B components into the quaternionic imaginary parts (i, j, k).⁸

⚠ CRITICAL: Represent a pixel at image coordinates (n, m) as $q(n, m) = r(n, m)i + g(n, m)j + b(n, m)k$ where $r(n, m)$, $g(n, m)$ and $b(n, m)$ are the red, green, and blue components of the pixel, respectively.⁹

Pre-processing step (workflows #4–5)

⌚ Timing: 2–5 s per image (depending on size)

Here, we describe the step pertinent to transforming an input color image to a different visual representation which applies to stain separation (#4) and re-staining (#5).

2. Transform the input image I_q to a different representation TI_q with the following equation:

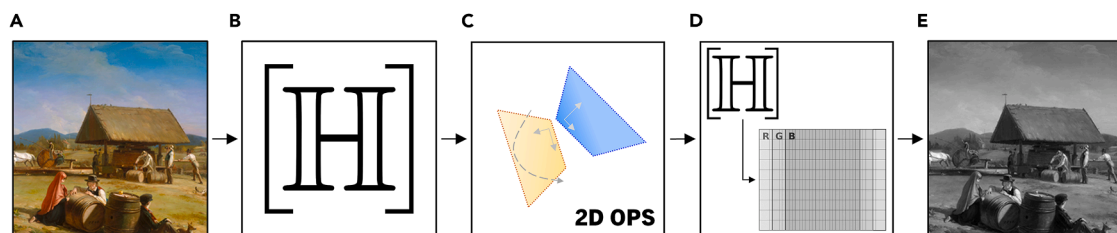


Figure 4. Computational de-colorization workflow (#2)

(A) input a natural image, (B) represent it as a quaternion matrix, (C) carry out the 2D orthogonal planes split (Equation 3, $f=g=\mu_7$), (D) extract and normalize (range [0, 1]) color channel matrices from the resulting quaternion matrix, and (E) compute the arithmetic mean of the color channels to obtain the transformed color image.

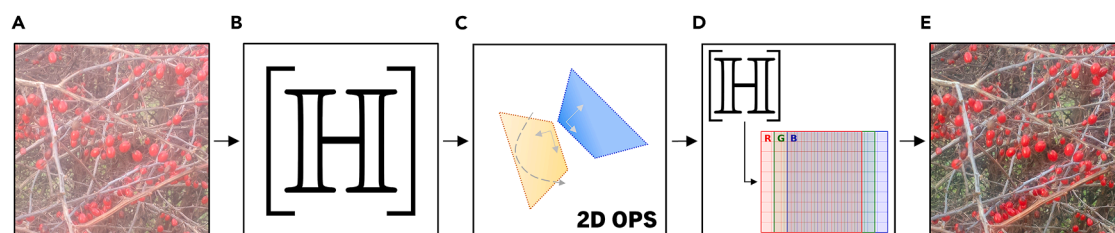


Figure 5. Computational contrast enhancement workflow (#3)

(A) input a histological image, (B) represent it as a quaternion matrix, (C) carry out the 2D orthogonal planes split (Equation 6, $f=1i+1j+1k$ and $g=0.01i+0.01j+0.01k$, parameters: $\alpha=10, \beta=1, \gamma=-1, \delta=1$), (D) extract and normalize (range [0, 1]) color channel matrices from the resulting quaternion matrix, and (E) concatenate the color channels to obtain the transformed color image.

$$Tl_q = \ln(I_q) e^{I_q} \quad (\text{Equation 1})$$

3. Extract the color channel matrices from the resulting quaternion matrix of Equation 1.
4. Retain the scalar parts during computations but discard them prior to image encoding.
5. Carry out data normalization for all color channels jointly in the range [0, 1].
6. Concatenate the color channels to obtain the transformed color image: Tl_{nq} (24-bit).

2D orthogonal planes split (workflows #1–5)

⌚ Timing: 1–12 s per image (depending on size)

Here, we describe the step pertinent to performing the 2D orthogonal planes split which applies to all workflows.

7. Carry out the split with respect to one or two pure quaternions or pure unit quaternions f, g (where $\otimes$ is the quaternion multiplication operator):

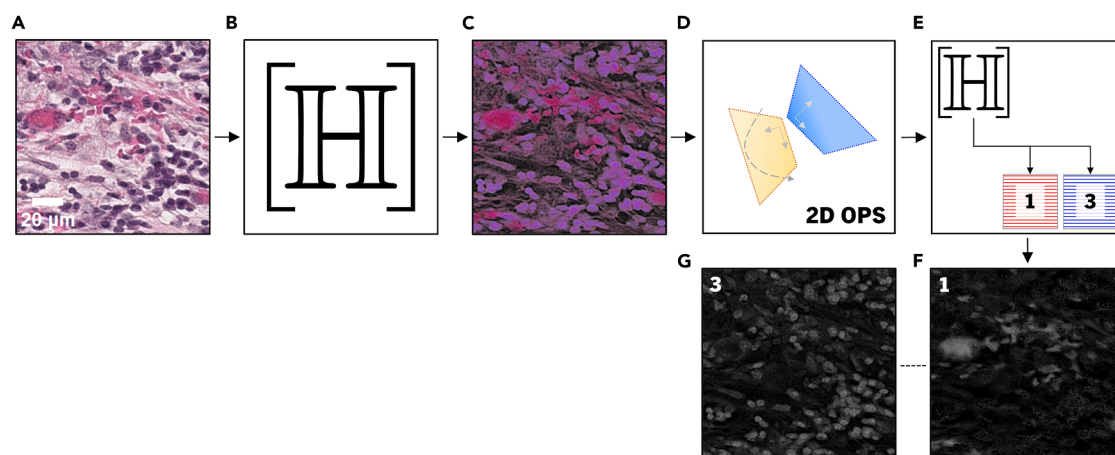


Figure 6. Computational stain separation workflow (#4)

(A) input a histological image, (B) represent it as a quaternion matrix, (C) transform it to a different representation, (D) carry out the 2D orthogonal planes split (Equation 8, $f=s_1$ and $g=s_2$, values derive from a computational method to estimate the stain separation matrix^{10,11}), (E) extract and truncate (range [0, 1]) color channel matrices from the resulting quaternion matrix, (F) retain the red color channel as the first output image, and (G) retain the blue color channel as the second output image.

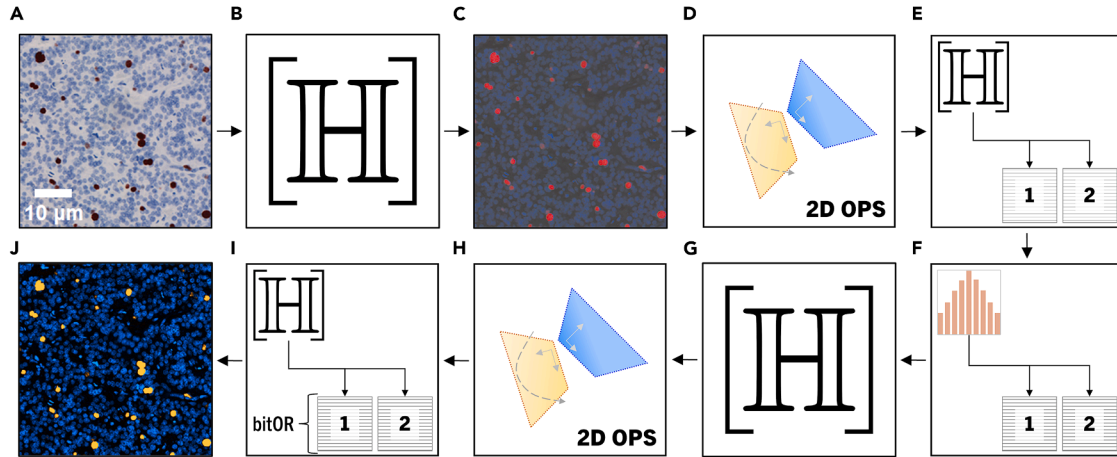


Figure 7. Computational re-staining workflow (#5)

(A) input a histological image (square section for presentation purposes), (B) represent it as a quaternion matrix, (C) transform it to a different representation, (D) carry out the 2D orthogonal planes split (Equation 7, $f=g=\mu_7$), (E) extract and truncate (range [0, 1]) color channel matrices from the resulting quaternion matrix, (F) linearly scale the histograms of the color channel matrices, (G) represent them as a quaternion matrices, (H) carry out the 2D orthogonal planes split (Equations 9 and 10, $f_1 = \frac{30}{255}i + \frac{144}{255}j + \frac{255}{255}k$ and $f_2 = \frac{210}{255}i + \frac{125}{255}j + \frac{45}{255}k$ for the shades of blue and orange, respectively), (I) extract and truncate (range [0, 1]) color channel matrices from the resulting quaternion matrix, perform a bitwise OR operation, and (J) obtain the transformed color image.

$$q_+^{(f,f)} = \frac{1}{2}(l_q + f \otimes l_q \otimes f) \quad (\text{Equation 2})$$

$$q_-^{(f,f)} = \frac{1}{2}(l_q - f \otimes l_q \otimes f) \quad (\text{Equation 3})$$

$$q_+^{(f,g)} = \frac{1}{2}(l_q + f \otimes l_q \otimes g) \quad (\text{Equation 4})$$

$$q_-^{(f,g)} = \frac{1}{2}(l_q - f \otimes l_q \otimes g) \quad (\text{Equation 5})$$

$$q_+^{(f,g)} = \frac{1}{2} \left(\left(\alpha \left[\frac{l_q}{2} \right] - \beta \left[\frac{1}{e^{l_q}} \right] + \gamma \left[\frac{e^{l_q}}{l_q + e^{l_q}} \right] + \delta \left[\frac{l_q^{1/2}}{l_q + e^{l_q}} \right] \right) + f \otimes l_q \otimes g \right) \quad (\text{Equation 6})$$

$$q_+^{(f,f)} = \frac{1}{2}(Tl_{nq} + f \otimes Tl_{nq} \otimes f) \quad (\text{Equation 7})$$

$$q_+^{(f,g)} = \frac{1}{2}(Tl_{nq} + f \otimes Tl_{nq} \otimes g) \quad (\text{Equation 8})$$

8. For Workflow #1 (re-colorization), perform the split according to any of the Equations 2, 3, 4, and 5.
9. For Workflow #2 (de-colorization), perform the split according to Equation 3.
10. For Workflow #3 (contrast enhancement), perform the split according to Equation 6; parameters have the following values (natural and histology images): $\alpha=10, \beta=1, \gamma=-1, \delta=1$.

Note: For medical CT grayscale images, which exhibit different dynamic range and other characteristics, parameters have the following values: $\alpha=1, \beta=10^5, \gamma=10^4, \delta=10^4$.

11. For Workflow #4 (stain separation), perform the split according to any of the [Equations 7](#) and [8](#).
12. For Workflow #5 (re-staining), perform the split according to [Equation 7](#).

Channel normalization/truncation, histogram operation, and output (workflows #1–5)

⌚ Timing: 2–7 s per image (depending on size)

Here, we describe the step pertinent to the required post-processing operations for obtaining the output (#1–4) or intermediary image (#5) which applies to all workflows.

13. For Workflow #1 (re-colorization): a. Extract color channel matrices from resulting quaternion matrix of any of the [Equations 2, 3, 4, and 5](#). b. Retain the scalar parts during computations but discard them prior to image encoding. c. Carry out data normalization for each color channel in the range [0, 1]. d. Concatenate the color channels to obtain the transformed color image: I_T (24-bit).
14. For Workflow #2 (de-colorization): a. Extract color channel matrices from resulting quaternion matrix of [Equation 3](#). b. Retain the scalar parts during computations but discard them prior to image encoding. c. Carry out data normalization for each color channel in the range [0, 1]. d. Compute the arithmetic mean of the color channels to obtain the transformed greyscale image: I_G (8-bit).
15. For Workflow #3 (contrast enhancement): a. Extract color channel matrices from resulting quaternion matrix of [Equation 6](#). b. Retain the scalar parts during computations but discard them prior to image encoding. c. Carry out data normalization for each color channel in the range [0, 1]. d. Concatenate the color channels to obtain the transformed color image: I_T (24-bit).
16. For Workflow #4 (stain separation): a. Extract color channel matrices (ccA, ccB, ccC) from resulting quaternion matrix of [Equations 7](#) and [8](#). b. Retain the scalar parts during computations but discard them prior to image encoding. c. Carry out data truncation for each color channel in the range [0, 1]. d. Depending on the stains present in the histological image, retain different color channels as output images.
17. For Workflow #5 (re-staining): a. Extract color channel matrices ccA and ccB from resulting quaternion matrix of [Equation 7](#). b. Retain the scalar parts during computations but discard them prior to image encoding. c. Carry out data truncation for each color channel in the range [0, 1]. d. Linearly scale the histograms of ccA and ccB map the intensity values between 0 and some value X (that depends on the image, e.g., 0.15–0.50) to values between 0 and 1.

2D orthogonal planes split (workflow #5)

⌚ Timing: 1–12 s per image (depending on size)

Here, we describe the step pertinent to performing the 2D orthogonal planes split which applies to Workflow #5.

18. Convert color channels ccA and ccB to quaternion matrices: $I_{ccA} = ccA(n, m)i + ccA(n, m)j + ccA(n, m)k$ and $I_{ccB} = ccB(n, m)i + ccB(n, m)j + ccB(n, m)k$.
19. Carry out the split for both color channels with respect to user-defined pure quaternions f_1, f_2 (where $\otimes$ is the quaternion multiplication operator):

$$q_-^{(f, f)} = \frac{1}{2}(I_{ccA} - f_1 \otimes I_{ccA} \otimes f_1) \quad (\text{Equation 9})$$

$$q_-^{(f, f)} = \frac{1}{2}(I_{ccB} - f_2 \otimes I_{ccB} \otimes f_2) \quad (\text{Equation 10})$$

Channel truncation, bitwise OR operation, and output (workflow #5)

⌚ Timing: 2–7 s per image (depending on size)

Here, we describe the step pertinent to the required post-processing operations for obtaining the output image which applies to Workflow #5.

20. Extract the color channel matrices ccA_n and ccB_n from the resulting quaternion matrix of both [Equations 9 and 10](#).
21. Retain the scalar parts during computations but discard them prior to image encoding.
22. Carry out data truncation for each color channel in the range $[0, 1]$.
23. Perform a bitwise OR operation such that $ccAB_n = ccA_n \vee ccB_n$ in order to obtain the transformed color image: $I_T = ccAB_n$ (24-bit).

EXPECTED OUTCOMES

Using this protocol, you generate a set of image outputs across natural and biomedical image data. From any color image (RGB) input, you obtain: a) re-colored renditions that modulate color appearance, b) grayscale (de-colored) images suitable for downstream analysis, c) contrast-enhanced images that improve visibility of fine details without introducing artifacts, d) re-stained images that selectively amplify or suppress stains on a uniform dark background assisting interpretation, and e) stain-separated images that isolate histological stains into distinct images for further processing. For natural images, you obtain distinct alternative color-based or grayscale representations, and improved visual quality and interpretability. For histological images, you obtain colorblind-friendly renditions, crisper separation of tissue compartments, clearer visualization of immune infiltrates, and improved perception of subcellular signals, for faster manual review and more reliable region-of-interest selection. All transformations rely only on basic operations, so outputs are reproducible and computationally accessible; runtime scales linearly with image size and very large image applications are feasible via tiling and parallelization. Practically, you obtain 24-bit color images from the re-colorization, re-staining, and contrast enhancement workflows, 8-bit grayscale images from the de-colorization workflow, and single-channel stain maps from the stain separation workflow. Altogether, you obtain a collection of varied renderings for natural images, and interpretable or analysis-ready outputs for biomedical images.

As a guide, the choice of quaternions f and g should be viewed as a steering choice, not as the search for a universally optimal setting. Any valid pair f, g defines a legitimate split into two orthogonal analysis planes, and conversely a desired pair of orthogonal planes can be associated with a corresponding pair f, g . In practice, the usefulness of a given choice depends on the application and on whether the selected directions reflect meaningful structures in the image, such as RGB axes, the gray-line direction, or stain vectors. Thus, non-optimal choices do not invalidate the method, but they may produce decompositions that are less informative or less useful for a specific task, e.g., by emphasizing color mixtures that do not correspond well to the target structures or by reducing separation quality when dominant hues are very similar.

Regarding parameters $\alpha, \beta, \gamma, \delta$ that control contrast enhancement behavior by weighting the four sub-terms of the modular expression in [Equation 6](#), in this sense, they do not act as universal constants, but as application-dependent tuning parameters that regulate the balance between stronger contrast enhancement and preservation of a visually acceptable output. Because the underlying term is constructed from sequential contrast adjustment/refinement components, changing these weights changes the final enhancement profile. In practice, the method is therefore sensitive to parameter choice in a practical rather than mathematical sense; the transformation remains valid for different values, but non-optimal choices may lead to weak

enhancement, excessive enhancement, or less natural-looking results. Thus, these parameters should be viewed as empirically useful values for the given image types.

LIMITATIONS

In general, the protocol depends on how the quaternions f, g are chosen. While the study demonstrates mathematically-driven values as good choices that depend on the problem at hand, user preferences or computationally derived choices are valid alternatives. Selection is more intuitive when the transformed image's color gamut is visualized (e.g., 3D RGB/HSV point clouds) since the shape, density, and topology of these point clouds track visible differences such as smoothness and vividness. In particular, stain separation can be sensitive to colors being adjacent to one another. When stains are close in hue (e.g., reds vs. pinks), direct separation can be unreliable; in this case additional re-colorization steps are required to highlight the remaining colors. Furthermore, clinical use is not established; non-standard renderings and re-stained visualizations may disrupt routine interpretation because clinicians are trained on established visual patterns. Adoption would require retraining and rigorous validation to show that diagnostic accuracy is preserved or improved in larger clinical studies; until then, use should be limited to research and education. Even with promising ideas like dynamic overlay visualizations, clinical reliability remains unproven. Another practical constraint is scalability; although the protocol scales linearly with pixels, gigapixel whole-slide images and memory limits can prevent near-real-time use without tiling and parallelization. Finally, tooling is at an early-stage; the current codebase is intended for understanding rather than production, and there is no integrated whole slide viewer or plugin yet, factors that can hinder robust deployment across labs.

TROUBLESHOOTING

The following notes provide practical troubleshooting guidance for issues that may arise when applying the protocol. They address four main problems: i) re-colorization when some colors are underrepresented, ii) selection of suitable f, g , iii) tuning of contrast enhancement parameters for different image types, and iv) sub-optimal stain separation. For each case, we outline targeted adjustments that can be applied while keeping the overall protocol unchanged.

Problem 1

Re-colorization: colors underrepresented in an image (step 13).

Potential solution

When some colors are underrepresented in an image (e.g., sparse representation, faint appearance) then use an exemplar image; crop a part of an existing image that represents preferred color distributions or choose a completely new image to optimally capture color variability. Carry out the split on the exemplar image with respect to one or two pure quaternions or pure unit quaternions f, g , and then carry out data normalization for each color channel in the range $[0, 1]$. The pairs comprising $[min, max]$ values are computed rapidly for the exemplar image and any desired combination of pure unit quaternions. Utilize these pre-computed normalization ranges in step 13, when doing re-colorization, before concatenating the color channels to obtain the transformed color image (Figure 8).

Problem 2

Re-colorization: choosing suitable f, g (step 8).

Potential solution

To improve how you identify color patterns and select appropriate f, g , visualize the color gamut of the transformed images. Then, analyze (either qualitatively or quantitatively) the resulting 3D point clouds to assess how different renditions relate to each other. In particular, examine the shape (spatial distribution), density (point concentration), and topology (point connectivity) of each color gamut point cloud, as these properties can reflect differences in visual appearance, such as distinct

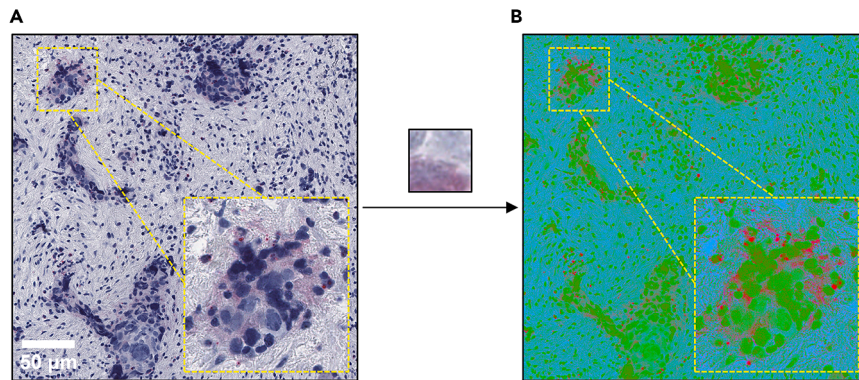


Figure 8. Troubleshooting for underrepresented colors

(A) crop a small exemplar image (50×50 pixels) shown in the middle from the single immunostain (in situ hybridization) showing human epithelial ovarian carcinoma, and B) compute the pair comprising minimum and maximum values for the exemplar image using $f = \mu_d$, and then utilize this normalization pair in Step 4 to obtain the transformed image. The exemplar image is selectively cropped to include shades of blue and red present in the original image. The embedded images show magnified areas.

color patterns, color smoothness, and color vividness. Once suitable f, g have been identified, utilize them in step 8 for batch processing. As an example, [Figure 9](#) shows sample 3D color gamut visualizations of a histological image (for different re-colorizations) in the RGB and HSV color spaces.

Problem 3

Contrast enhancement: parameter tuning (step 10).

Potential solution

The parameters α, β, γ , and δ weight the four sub-terms of [Equation 6](#), and therefore control the contrast enhancement applied to the image. These parameters can be tuned for different types of images while keeping the rest of the workflow unchanged. For datasets deriving from biological and medical imaging modalities (e.g., electron microscopy), fine-tune on a dataset that encapsulates the intended use, i.e., a small development set that is representative of the intended application and acquisition protocol. Alternatively, adjust these parameters using a low-dimensional search by varying one parameter at a time or exploring a coarse grid of candidate values, and then select combinations that preserve a natural visual appearance. [Equation 6](#) involves only basic arithmetic operations, quaternion additions/subtractions, and component-wise multiplications/divisions, thus this parameter search is computationally inexpensive. Once a suitable parameter set has been identified, reuse it in step 10 for batch processing without further tuning.

Problem 4

Stain separation: stains are not optimally separated (step 11).

Potential solution

Stain separation can be sub-optimal due to the limited effectiveness of the approach or the similarity of the values in the stain separation matrix for the two stains. As a first step, obtain a better stain separation matrix with another computational method. If this is not possible, then perform an additional step before stain separation (step 11); this comprises computational re-staining as described in the protocol. [Figure 10](#) showcases this scenario.

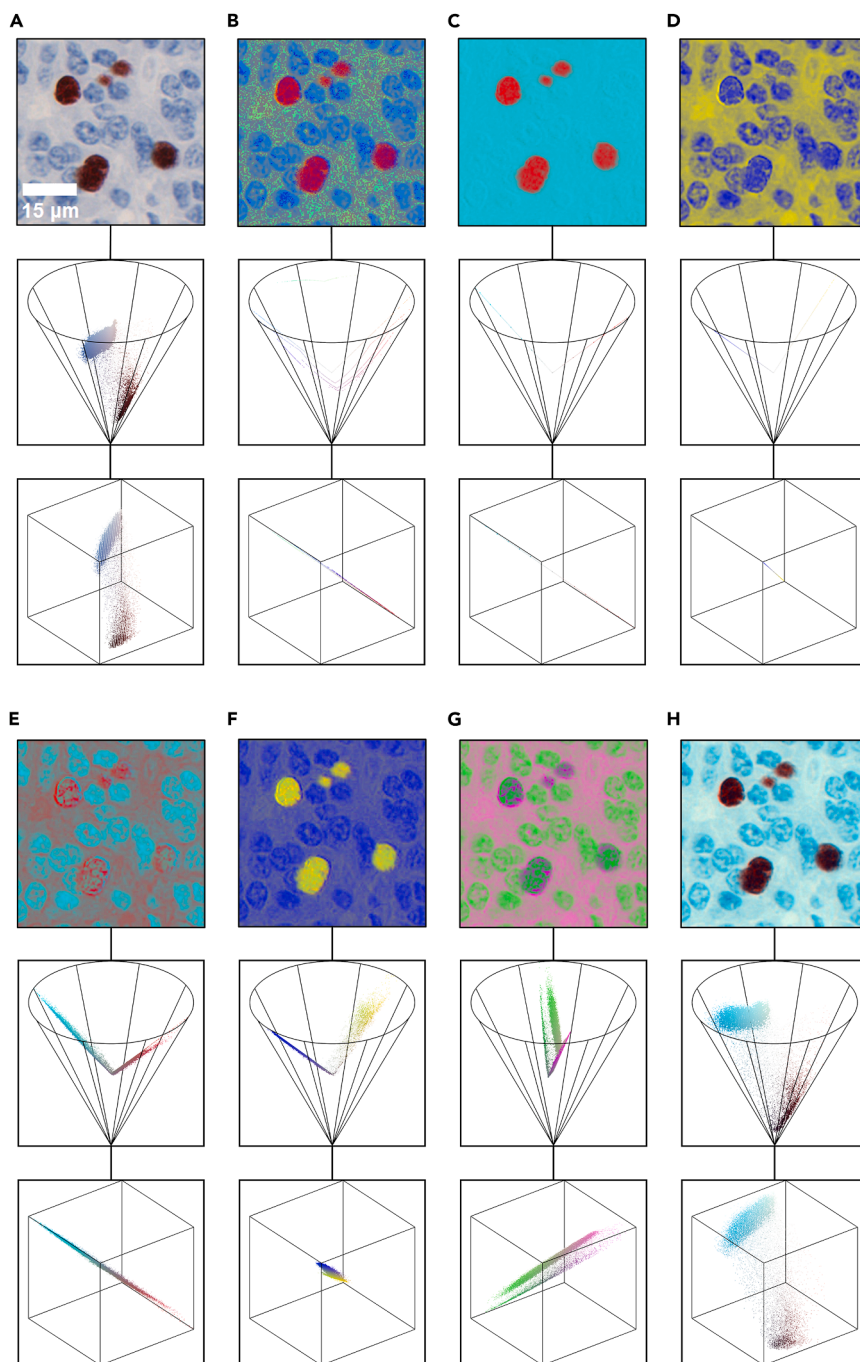


Figure 9. Choosing suitable f, g : 3D color gamut visualizations as point clouds in the RGB and HSV color spaces of (A) a histological image section with point cloud visualizations corresponding to the following renditions: B) $q_-, f = \mu_9$, C) $q_-, f = \mu_{11}$, D) $q_-, f = \mu_{13}$, E) $q_+, f = \mu_1, g = \mu_{11}$, F) $q_+, f = \mu_3, g = \mu_{13}$, G) $q_+, f = \mu_8, g = \mu_{12}$, and H) $q_+, f = \mu_8, g = \mu_{13}$.

RESOURCE AVAILABILITY

Lead contact

Requests for further information and resources should be directed to and will be fulfilled by the lead contact and corresponding author, Nektarios A. Valous (nek.valous@nct-heidelberg.de).

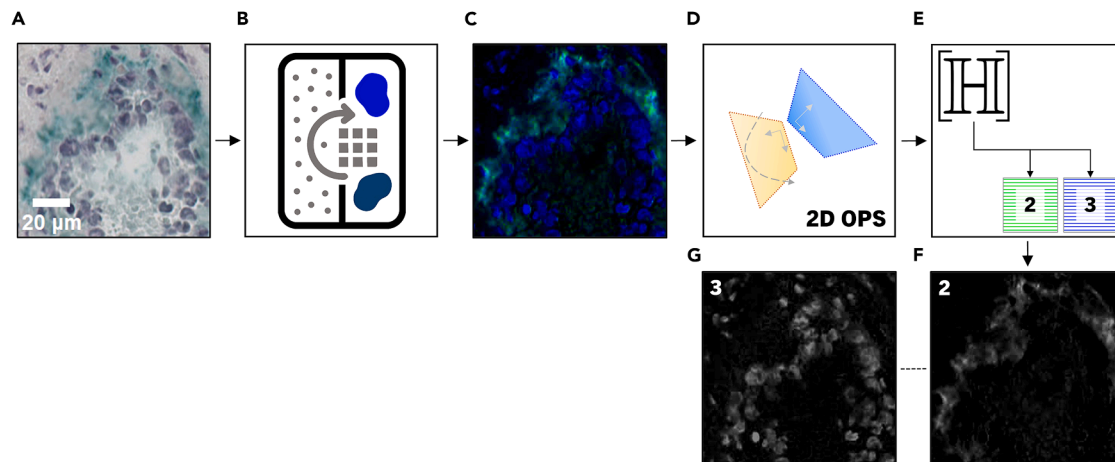


Figure 10. Troubleshooting for sub-optimal stain separation

(A) input a histological image, (B) perform re-staining as guided by the protocol, (C) obtain the re-stained image, (D) carry out the 2D orthogonal planes split (Equation 8, $f=s_1$ and $g=s_2$, values derive from a computational method to estimate the stain separation matrix^{10,12}), (E) extract and truncate (range [0, 1]) color channel matrices from the resulting quaternion matrix, (F) retain the green color channel as the first output image, and (G) retain the blue color channel as the second output image.

Technical contact

Technical questions on executing this protocol should be directed to and will be answered by the technical contact, Nektarios A. Valous (nek.valous@nct-heidelberg.de).

Materials availability

This study did not generate new unique reagents.

Data and code availability

Python scripts, Jupyter notebooks, MATLAB code, and test images are publicly available in the following Figshare repository: <https://doi.org/10.6084/m9.figshare.29683382>.¹¹ The code implements functionalities including re-colorization, de-colorization, contrast enhancement, re-staining, and stain separation. It provides insights into the workflows and their outputs, showcases the methodologies introduced in this work, and clarifies key implementation details. Although the code is fully operational, it is intended mainly for learning and exploration rather than direct use in production settings. The output files generated by the code include results from both Python and MATLAB implementations; these images serve as validation, confirming that both implementations produce consistent results.

ACKNOWLEDGMENTS

We wish to extend our gratitude to the scientists and mathematicians who provided feedback and comments during oral and poster presentations of this work.

AUTHOR CONTRIBUTIONS

N.A.V. designed and performed research; E.H., D.D., R.R.M., F.P., M.S.-C., A.B., I.P., C.F., A.R., C.C.W., B.L., N.H., I.Z., and D.J. contributed resources and/or expertise in their respective domains, including the provision of materials, wet-laboratory methods, clinical insights, and mathematical/computational reasoning; D.D. developed Python-based computational libraries; and N.A.V. analyzed the data and wrote the manuscript. All authors read, reviewed, and approved the manuscript.

DECLARATION OF INTERESTS

N.A.V., N.H., I.Z., and D.J. declare that segments of the work presented in the manuscript are protected by intellectual property rights and are covered under US patent no. 11,501,444, titled "Method, software module, and integrated circuit for color visualization and separation" (United States Patent and Trademark Office, 2022). The patented components pertain to image re-colorization and histological stain separation.

REFERENCES

1. Valous, N.A., Hitzer, E., Duşe, D., Moraleda, R.R., Popp, F., Suarez-Carmona, M., Berthel, A., Papageorgiou, I., Fremd, C., Rölle, A., et al. (2025). Computational workflows for natural and biomedical image processing based on hypercomplex algebras. *Patterns* 6, 101388. <https://doi.org/10.1016/j.patter.2025.101388>.
2. Valous, N.A., Hitzer, E., Vitabile, S., Bernstein, S., Lavor, C., Abbott, D., Luna-Elizarrarás, M.E., and Lopes, W. (2024). Hypercomplex Signal and Image Processing: Part 1. *IEEE Signal Process. Mag.* 41, 11–13. <https://doi.org/10.1109/MSP.2024.3378129>.
3. Valous, N.A., Hitzer, E., Vitabile, S., Bernstein, S., Lavor, C., Abbott, D., Luna-Elizarrarás, M.E., and Lopes, W. (2024). Hypercomplex Signal and Image Processing: Part 2. *IEEE Signal Process. Mag.* 41, 18–20. <https://doi.org/10.1109/MSP.2024.3415248>.
4. Le Bihan, N., and Mars, J. (2004). Singular value decomposition of quaternion matrices: a new tool for vector-sensor signal processing. *Signal Process.* 84, 1177–1199. <https://doi.org/10.1016/j.sigpro.2004.04.001>.
5. Pei, S.-C., and Cheng, C.-M. (1997). A novel block truncation coding of color images using a quaternion-moment-preserving principle. *IEEE Trans. Commun.* 45, 583–595. <https://doi.org/10.1109/26.592558>.
6. E. Hitzer, and S.J. Sangwine, eds. (2013). *Quaternion and Clifford Fourier Transforms and Wavelets* (Springer Basel). <https://doi.org/10.1007/978-3-0348-0603-9>.
7. Hitzer, E. (2015). The orthogonal planes split of quaternions and its relation to quaternion geometry of rotations. *J. Phys. Conf. Ser.* 597, 012042. <https://doi.org/10.1088/1742-6596/597/1/012042>.
8. Moxey, C.E., Sangwine, S.J., and Ell, T.A. (2003). Hypercomplex correlation techniques for vector images. *IEEE Trans. Signal Process.* 51, 1941–1953. <https://doi.org/10.1109/TSP.2003.812734>.
9. Ell, T.A., and Sangwine, S.J. (2007). Hypercomplex Fourier Transforms of Color Images. *IEEE Trans. Image Process.* 16, 22–35. <https://doi.org/10.1109/TIP.2006.884955>.
10. Pocock, J., Graham, S., Vu, Q.D., Jahanifar, M., Deshpande, S., Hadjigeorgiou, G., Shephard, A., Bashir, R.M.S., Bilal, M., Lu, W., et al. (2022). TIAToolbox as an end-to-end library for advanced tissue image analytics. *Commun. Med.* 2, 120. <https://doi.org/10.1038/s43856-022-00186-5>.
11. Valous, N.A. (2025). 2D Orthogonal Planes Split: Python and MATLAB Code (Figshare). <https://doi.org/10.6084/m9.figshare.29683382.v1>.
12. Macenko, M., Niethammer, M., Marron, J.S., Borland, D., Woosley, J.T., Xiaojun, G., Schmitt, C., and Thomas, N.E. (2009). A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, C. Karl, D. Brooks, and B. Rosen, eds. (IEEE), pp. 1107–1110. <https://doi.org/10.1109/ISBI.2009.5193250>.