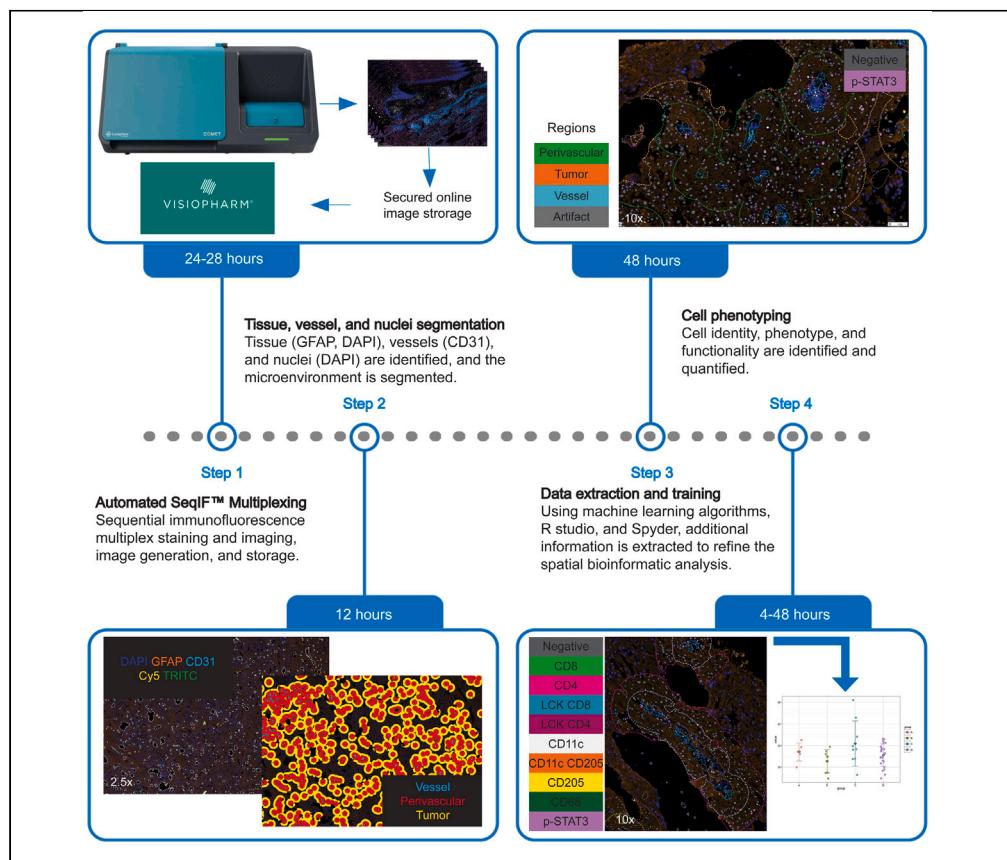


## Protocol

# Protocol to quantify immune cell distribution from the vasculature to the glioma microenvironment on sequential immunofluorescence multiplex images



Although myeloid-derived immune cells can be dispersed throughout the tumor microenvironment (TME), anti-tumor effector cells are confined to the perivascular space. Here, we present a protocol to quantify immune cell distribution from tumor vasculature to its glioma microenvironment on sequential immunofluorescence multiplex images. We describe steps for sequential immunofluorescence multiplex staining, image generation, and storage. We then detail the procedures for tissue, vessel, and nuclei segmentation; cell phenotyping; data extraction; and training using RStudio and Spyder.

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

Hinda Najem,  
Sebastian Pacheco,  
Joanna Kowal, Dan  
Winkowski, Jared K.  
Burks, Amy B.  
Heimberger

amy.heimberger@northwestern.edu

**Highlights**  
Steps for sequential multiplex immunofluorescence staining on the COMET system

Deep learning algorithms for tissue segmentation, vessel, and nuclei detection

Instructions to quantify and phenotype immune cells relative to the vasculature

Steps for data extraction and manipulation through RStudio and Spyder

Najem et al., STAR Protocols 5, 103079  
June 21, 2024 © 2024 The Author(s). Published by Elsevier Inc.  
<https://doi.org/10.1016/j.xpro.2024.103079>



## Protocol

# Protocol to quantify immune cell distribution from the vasculature to the glioma microenvironment on sequential immunofluorescence multiplex images

Hinda Najem,<sup>1,2,6</sup> Sebastian Pacheco,<sup>1,2</sup> Joanna Kowal,<sup>3</sup> Dan Winkowski,<sup>4</sup> Jared K. Burks,<sup>5</sup> and Amy B. Heimberger<sup>1,2,7,\*</sup>

<sup>1</sup>Department of Neurological Surgery, Northwestern University, Chicago, IL 60611, USA

<sup>2</sup>Malnati Brain Tumor Institute of the Robert H. Lurie Comprehensive Cancer Center, Feinberg School of Medicine, Northwestern University, Chicago, IL 60611, USA

<sup>3</sup>Lunaphore Technologies SA, 1131 Tolochenaz, Switzerland

<sup>4</sup>Visiopharm A/S, 2970 Horsholm, Denmark

<sup>5</sup>MD Anderson Cancer Center, Houston, TX 77030, USA

<sup>6</sup>Technical contact

<sup>7</sup>Lead contact

\*Correspondence: [amy.heimberger@northwestern.edu](mailto:amy.heimberger@northwestern.edu)  
<https://doi.org/10.1016/j.xpro.2024.103079>

## SUMMARY

Although myeloid-derived immune cells can be dispersed throughout the tumor microenvironment (TME), anti-tumor effector cells are confined to the perivascular space. Here, we present a protocol to quantify immune cell distribution from tumor vasculature to its glioma microenvironment on sequential immunofluorescence multiplex images. We describe steps for sequential immunofluorescence multiplex staining, image generation, and storage. We then detail the procedures for tissue, vessel, and nuclei segmentation; cell phenotyping; data extraction; and training using RStudio and Spyder.

## BEFORE YOU BEGIN

In solid tumors, there is not uniform distribution of immune cells throughout the TME because of chemokines gradients, osmotic pressures, and physical stromal barriers, to name a few. This lack of immune cell distribution is further confounded in the case of central nervous system tumors because of the variable presence of the blood-brain barrier (BBB). Innate immune cells such as macrophages constitute a frequent peripheral originating immune cell population that can distribute widely throughout the glioma TME<sup>1,2</sup>; however, T cell responses are typically confined to the perivascular niche.<sup>3</sup> With the development of strategies that now open the BBB in clinical trials<sup>4</sup> and a variety of immune therapeutics that may enhance distribution,<sup>5</sup> quantification of immune cells in various TME regions will be increasingly needed for endpoint evaluations during window-of-opportunity clinical trials.<sup>6,7</sup> The protocol below describes the specific steps for spatial bioinformatic analysis of immune cell distribution from the tumor vasculature on sequential multiplex immunofluorescence images. Baseline immune cell distribution including markers of activation and immune suppression in gliomas, before initiating an immunotherapy clinical trial, were quantified based on the distribution in the TME as a function of the tumor vasculature.

## Institutional permissions

Human glioma specimens were collected from consented patients who received surgery at Northwestern Memorial Hospital and Ann & Robert H. Lurie Children's Hospital of Chicago, according



to the approved Institutional Review Board protocols STU00214485 and STU00217628, respectively. All experiments conformed to the relevant regulatory standards. Similar permissions are required for other investigators at their respective institutions.

### Sequential multiplex immunofluorescence (SeqIF) staining on the COMET platform

⌚ Timing: 24–28 h for 4x 20plex SeqIF

This technique generates immunofluorescence (IF) multiplex images using the COMET platform from Lunaphore Technologies on which the subsequent spatial bioinformatic analysis is performed ([Figure 1A](#)).

1. Formalin fixed paraffin embedded (FFPE) slides are prepped first.
  - a. Mount 4 microns thick tissue sections on positively charged slides
  - b. Store the unstained slides at room temperature (20°C–22°C) until use.

**Note:** Antigen expression can be lost on cut sections over time, so it is advisable to section tissues within several weeks of the antigen retrieval step.

- c. Perform Antigen retrieval and dewaxing/deparaffinization of the FFPE slides with commercially available reagents.
  - i. Dewax and Hier retrieval buffer pH9 (Lunaphore Technologies) with the use of the PT Module (Epredia) following a 1-h protocol at 102°C.

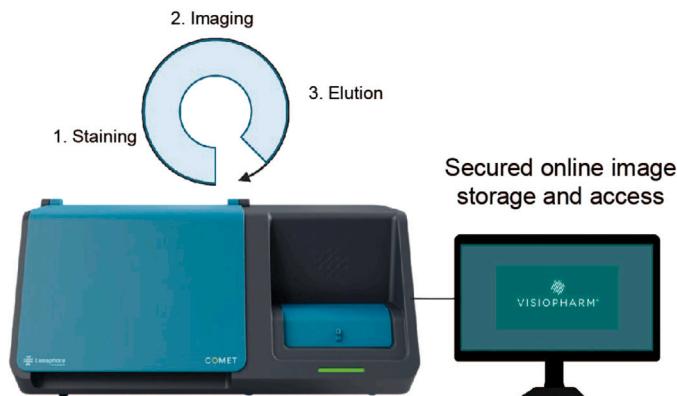
**Note:** The reagents are based on patented technology and are common practice reagents in immunohistochemistry experiments.<sup>8–11</sup>

2. Validate all antibodies using conventional immunohistochemistry and/or IF staining on positive control tissue in conjunction with the corresponding fluorophore and the spectral 4',6-diamidino-2-phenylindole (DAPI; Thermo Fisher Scientific) counterstain.
  - a. For optimal concentration and the best signal/noise ratio, test the antibodies at three different dilutions, starting with the manufacturer-recommended dilution (MRD), then MRD/2 and MRD/4.
  - b. Use the secondary Alexa fluorophore 555 (Thermo Fisher Scientific) and Alexa fluorophore 647 (Thermo Fisher Scientific) at 1/200 and 1/400 dilutions, respectively.
3. Execute the optimizations (characterization 2 and 3) and full protocol runs (Sequential IF) of the multiplex panel using the sequential IF (seqIF) methodology and protocols integrated into the Lunaphore COMET.<sup>12</sup>
4. Performed the staining on up to 4 tissue slides simultaneously following automated cycles of staining with 2 antibodies at a time, imaging, and then elution.

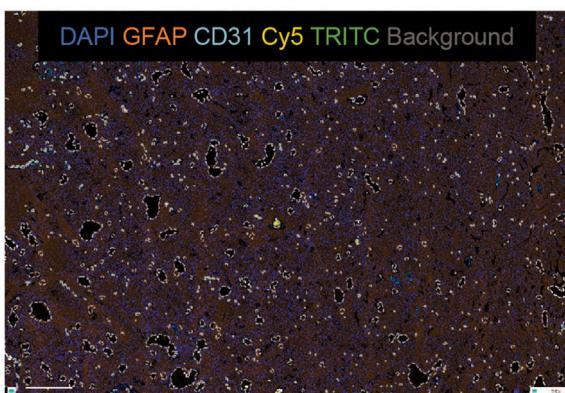
**Note:** No human intervention is required during this time.

- a. Dilute all reagents in Multistaining Buffer (BU06, Lunaphore Technologies).
- b. Select the software settings as below:
  - i. 2 min for the elution step in each cycle using the Elution Buffer (BU07-L, Lunaphore Technologies) at 37°C.
  - ii. 30 s for quenching using Quenching Buffer (BU08-L, Lunaphore Technologies).
  - iii. 4 min incubation time for all primary antibodies and 2 min for secondary antibodies during the staining step.
  - iv. Use Imaging Buffer (BU09, Lunaphore Technologies) for the imaging step.

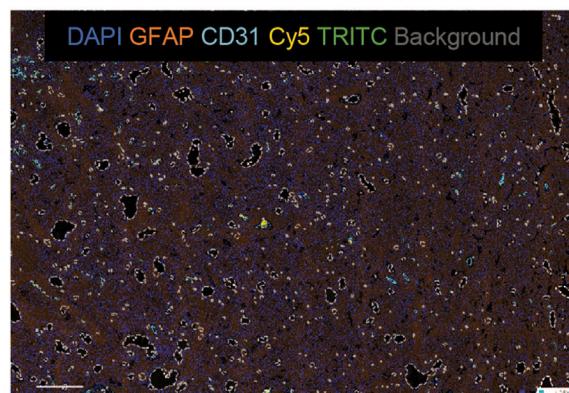
**A Automated SeqIF™ multiplexing**



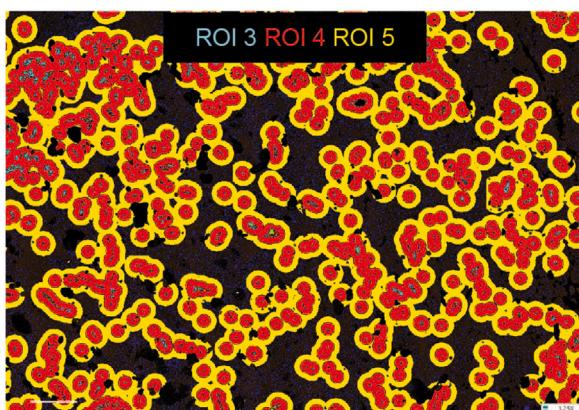
**B Tissue Segmentation**



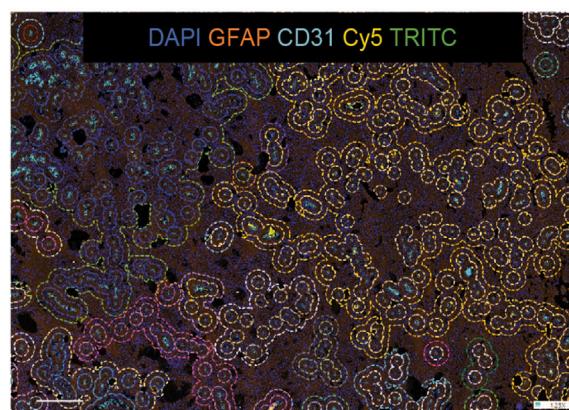
**C Vessel Identification**



**D Vessel Segmentation**



**E Consolidate Vessel Segmentation**



**Figure 1. Representative images of a pilocytic astrocytoma showing the first key steps in the bioinformatic analysis**

(A) The COMET platform from Lunaphore Technologies performs a fully automated protocol of seqIF, where at each cycle 2 antibodies' staining, imaging, and elution are performed, with the ability to detect 40 biomarkers in a single experiment. The images ( $n = 4$ ; one image per case) are then transferred into Visiopharm software for data analysis.

(B) Tissue Segmentation representative image showing nuclear DAPI (dark blue), glioma GFAP (orange), and vessel CD31 (cyan blue). Signal capture in the Cy5 (yellow) and TRITC (green) channels is initially turned on to remove the non-specific signals present within the image from subsequent analyses. Regions of the tissue that do not have either GFAP, CD31, or DAPI signal are labeled as background and removed from the analysis.

(C) During Vessel Identification, the CD31 signal is optimized through the selection of the threshold.

(D) During Vessel Segmentation, the distance from the vessel is then set at 55  $\mu\text{m}$  (red) and then a further 55  $\mu\text{m}$  (yellow) to define the tumor.

(E) Areas of confluence are merged into one ROI. All images are 1.25x magnification with 500  $\mu\text{m}$  scale bars.

**Note:** Imaging is performed through an integrated epifluorescent microscope at 20x magnification.

**Note:** Image registration is performed immediately after concluding the staining and imaging procedures by COMET Control Software.

**Note:** Due to the staining of 2 antibodies at a time, several precautions can be taken to avoid non-specific binding:

- Specificity of the primary antibodies is ensured during the multiplex optimization process on the COMET platform<sup>12</sup> (<https://lunaphore.com/resource-center/technical-notes/comet-assay-validation/>; <https://lunaphore.com/resource-center/technical-notes/assay-development-of-immuno-oncology-panels-with-comet/>).

- When a primary-secondary approach is being used, non-specific crosstalk is avoided by using two distinct species of primary antibodies with highly cross-adsorbed species-specific secondary antibodies (e.g., mouse and rabbit primary antibodies with goat anti-mouse secondary antibody detected in the TRITC channel and the goat anti-rabbit secondary antibody detected in the Cy5 channel).

- The sequential immunofluorescence protocol on the COMET platform takes advantage of a patented microfluidic technology and executes staining with a short tissue incubation time (range between 2 and 8 min) and dynamic incubation in which the mixing of the reagents is optimized to increase staining efficiency. Together, these two characteristics of the device act in synergy and decrease non-specific antibody binding.<sup>12</sup>

- For cases in which there is persistent non-specific background, the primary and/or secondary antibodies can be diluted in a blocking solution and a blocking step can be enabled during the automated protocol in COMET. This step can be implemented before the primary and/or secondary antibody incubation step (<https://lunaphore.com/resource-center/technical-notes/comet-assay-validation/>; <https://lunaphore.com/resource-center/technical-notes/assay-development-of-immuno-oncology-panels-with-comet/>).

If there is still a persistent non-specific signal, this is likely attributed to the secondary antibodies. A feature of the COMET platform includes the ability to eliminate and subtract this signal from downstream analysis. A standard cycle of only secondary antibody staining without primary antibodies can be integrated into all the protocols.

⚠ CRITICAL: Each seqIF protocol results in a single multi-layer OME-TIFF file where the imaging outputs from each cycle are stitched and aligned. COMET OME-TIFF files contain a DAPI image, intrinsic tissue autofluorescence in TRITC and Cy5 channels, and a single fluorescent layer per marker.

**Note:** Images generated are automatically saved in an online shared drive within the institution that can be accessed by the users from any computer.

**Note:** Markers can be analyzed and pseudo-colored for visualization in the Viewer from Lunaphore at any time after the completion of image registration.

### Data transfer to Visiopharm software

⌚ Timing: 5 min

This step allows the upload of the OME.TIFF images generated into the Visiopharm software.

5. Within the Visiopharm software 'Database' tab, create a new folder according to each project and import the respective raw generated multiplex images.
  - a. By selecting the 'Add' icon from within the Database toolbar, import the images by choosing the 'New Images to Database (Import)' setting.

**Note:** For this protocol, all raw OME.TIFF files are imported and analyzed without any manipulation or background subtraction added. These latter features can be enabled and exported into the images using the Viewer from Lunaphore.

**⚠ CRITICAL:** For the appropriate transfer of OME.TIFF file formats into the software: After the folder that includes the images is selected for transfer, the file type option to choose should be 'All (\*)'.

### KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
<b>Antibodies</b>		
DAPI	Thermo Fisher Scientific	Cat# 62248
Alexa Fluorophore 647 secondary antibody	Thermo Fisher Scientific	Cat# A32733
Alexa Fluorophore 555 secondary antibody	Thermo Fisher Scientific	Cat# A32727
GFAP	Abcam	Cat# Ab68428 Clone EPR1034Y Dilution 1/2,000
CD31	Abcam	Cat# Ab225883 Clone EPR17259 Dilution 1/1,500
CD4	Abcam	Cat# Ab133616 Clone EPR6855 Dilution 1/500
CD8	Leica	Cat# PA0183 Clone 4B11 No dilution required
CD68	Dako Agilent	Cat# M0876 Clone PG-M1 No dilution required
CD11c	Abcam	Cat# Ab52632 Clone EP1347Y Dilution 1/300
CD163	Abcam	Cat# Ab182422 Clone EPR19518 Dilution 1/600
CD205	Abcam	Cat# Ab124897 Clone EPR5233 Dilution 1/700
TMEM119	Abcam	Cat# Ab185333 Polyclonal Dilution 1/100
P2RY12	Atlas Antibodies	Cat# HPA014518 Polyclonal Dilution 1/1,000
FOXP3	Cell Signaling Technology	Cat# 98377S Clone D2W8E Dilution 1/100
p-STAT3	Cell Signaling Technology	Cat# 9145S Clone D3A7 (Tyr705) XP Dilution 1/250
PD-1	Abcam	Cat# Ab137132 Clone EPR4877(2) Dilution 1/400

(Continued on next page)

**Continued**

REAGENT or RESOURCE	SOURCE	IDENTIFIER
PD-L1	GenomeMe	Cat# IHC411-1 Clone IHC411 Dilution 1/100
TIM-3	Abcam	Cat# Ab241332 Clone EPR22241 Dilution 1/1,000
LAG-3	Novus Biologicals	Cat# NBP1-97657 Clone L4-PL33 (17B4) Dilution 1/500
Granzyme B	Santa Cruz	Cat# Sc-8022 Clone 2C5 Dilution 1/500
LCK	Cell Signaling Technology	Cat# 2984S Clone D88 XP Dilution 1/100
HLA-DR	Abcam	Cat# Ab20181 Clone TAL1B5 Dilution 1/1,100
CD45-RO	US Biological	Cat# 172054 Clone SPM125 Dilution 1/200
<b>Biological samples</b>		
Human pilocytic astrocytoma	Northwestern University Nervous System Tumor Bank	
Human glioblastoma	Northwestern University Nervous System Tumor Bank	
<b>Chemicals, peptides, and recombinant proteins</b>		
Dewax and Hier buffer H	Lunaphore Technologies SA/Epredia	Cat# AR03/TA-999-DHBH
Multistaining buffer	Lunaphore Technologies SA	Cat# BU06
Elution buffer	Lunaphore Technologies SA	Cat# BU07-L
Quenching buffer	Lunaphore Technologies SA	Cat# BU08-L
Imaging buffer	Lunaphore Technologies SA	Cat# BU09
<b>Software and algorithms</b>		
Visiopharm version 12.2022	Visiopharm A/S	<a href="https://visiopharm.com/">https://visiopharm.com/</a>
R version 4.3.2	Online	<a href="https://cran.rstudio.com/">https://cran.rstudio.com/</a>
RStudio version 2023.09.1–494	Online	RStudio Desktop - Posit
Spyder (Python 3.11)	Online	<a href="https://Home — Spyder IDE (spyder-ide.org)">Home — Spyder IDE (spyder-ide.org)</a>
<b>Other</b>		
COMET	Lunaphore Technologies SA	<a href="https://lunaphore.com/">https://lunaphore.com/</a>
PT Model	Epredia	<a href="https://epredia.com/solutions/immunohistochemistry-solutions/pt-module/">https://epredia.com/solutions/immunohistochemistry-solutions/pt-module/</a>

## STEP-BY-STEP METHOD DETAILS

**Note:** Select default settings if not otherwise specified.

All intensity ranges for each marker can be empirically determined from the positive control staining (for the higher limit) and the signal to noise ratio (for the lower limit).

### Part 1: Tissue segmentation

⌚ Timing: 6 h

This part identifies the tissue areas and eliminates background or non-tissue from subsequent analysis ([Methods video S1](#)). It also compensates for missing tissue in certain regions of the image resulting from sample degradation during post-resection processing. It ultimately refines the blood vessel shape and improves the accuracy of their identification (Cf. Part 2).

1. Create a new APP for training the “deep learning” algorithm and use two image classes: Class 1 for ‘Background’ and Class 2 for ‘Tissue.’
  - a. For ‘Input’ settings, set up a magnification of 5X and a field of view of 1024 × 1024 pixels.
  - b. In the ‘Regions to Analyze’ settings, select the ROI for outlining training areas. In this instance, ROI 2.
  - c. For ‘Classification’ settings, set the Deep Learning method with the DeepLabv3+ architecture.
    - i. In the ‘AI Architect’ menu, set the input size to 512 × 512 pixels, the learning rate adjusted to 1.0E-5, and the minibatch size to 4.

⚠ CRITICAL: Training should be conducted in this step using only the DAPI channel, restricting pixel intensities to a range between 3000 and 65000 to reduce noise because data is in 16-bit format.

**Note:** The DAPI channel signal might vary depending on cellular density between tissues. As such, the intensity interval should be adjusted accordingly for the best visualization of the nuclear morphology across tissues.

2. For the Classifier, outline training areas using ROI 2.

**Note:** In each image, at least one training area should be generated to enhance the robustness of the classifier and the view of GFAP, DAPI, and CD31 channels should be enabled to help in labeling training areas.

- a. Label areas with high GFAP, DAPI, and CD31 signal as ‘Tissue’ under Class 2 (Figure 1B).
- b. Assign the areas without tissue to the ‘Background’ label under Class 1. The software will then automatically label unselected areas within the ROI as either Class 1 or 2 accordingly.

**Note:** Only a small set of examples are needed in the Class 1 category for the software to identify regions accurately.

3. Load all images into the slide tray.

- a. Initiate the training within the ‘Classification’ section of the App and monitor the “loss function” graph.

**Note:** Training the algorithm of the “tissue segmentation App” on all the images simultaneously increases its accuracy.

- b. Maintain the training until the loss function converges and plateaus to a final minimal value of <0.5.

⚠ CRITICAL: The closer to zero the final minimal value is, the better, indicating a low final loss value. For instance, our APP is trained for approximately 22000 iterations until a plateau at the lowest value of 0.3 is achieved.

**Note:** At this point the “tissue segmentation App” is finalized.

4. To add subsequent steps and move to the next APP, remove all training areas by clearing ROIs and labels.

**Note:** Editing the ‘Post Processing’ settings is therefore necessary.

- a. The first step is to convert areas labeled as ‘Background’ to the ‘Clear’ label.

- b. Next, it is necessary to surround areas labeled as 'Tissue' with a specific ROI. In this instance, ROI 2 is used for this example.
- c. Finally, include a step to change the 'Tissue' label to the 'Clear' label.

## Part 2: Vessel identification and segmentation

⌚ Timing: 3 h

This part identifies all vessels within the tissue area and segments the tumor areas according to the distance away from the vessels ([Methods video S2](#)). A threshold algorithm is employed to accurately label regions exhibiting the presence of CD31, an endothelial marker of choice in this project, thereby establishing the vessel area.

**Note:** Alternative markers for vessel labeling include collagen 4A, podoplanin, or smooth muscle actin alpha 2.

5. Create a new App for this thresholding algorithm and use one image class: Class 5 for 'Vessel'.
  - a. For 'Input' settings, set up a magnification of 5X and a field of view of  $1024 \times 1024$  pixels in 'Advanced Sampling' settings.
  - b. In the 'Regions to Analyze' settings, chose the ROI that surround areas identified as tissue and was used in the previous App 'Tissue Segmentation'. For this instance, this would be ROI 2.
  - c. For 'Classification' settings, select the 'Threshold' method.
    - i. The 'Vessel' label is assigned to areas displaying the CD31 marker, with pixel intensities ranging from 6000 to 30000.

**Note:** This range was visually chosen across multiple vessels within the tissue to reduce noise and most effectively identify all the vessels within the tissue but may be different for others ([Figure 1C](#)).

**Note:** Depending on the intensity of the CD31 signal between projects and staining techniques, the pixel intensity range assigned may change. By visual assessments, testing multiple ranges throughout the tissue and between images is necessary for each project to allow the software to find the most appropriate range to identify potentially all vessels present.

6. Refinement of vessel identification and segmentation:
  - a. Add a new step within the 'Post-Processing' section to exclude vessel-labeled areas smaller than  $4 \mu\text{m}^2$ .

**Note:** This filters out isolated pixels expressing CD31, ensuring the inclusion of regions with vessel-like morphology.

- b. Add another step to expand ('dilate') vessel-labeled areas by 15 pixels.

**Note:** This mirrors the visual signal given by the CD31 marker after thresholding, taking into consideration the vessel architecture and thickness from the lumen to the outer membrane.

**⚠ CRITICAL: Images can display areas of nonspecific binding indicated visually by high levels of secondary antibodies signal (Cy5 and TRITC in this setting). These areas may inadvertently receive a 'Vessel' label. To correct this and refine your labeling, the eraser tool for both the ROIs and labels can be used repetitively to remove these areas and refine vessel segmentation. The refinements can be done manually on one of the images, but the algorithm is being trained on multiple images at the same time loaded within the slide tray.**

7. To refine the labeling of the perivascular areas within the tissue, add a new step within the 'Post-Processing' section.
  - a. Assign a new ROI for each area of tissue chosen at a certain distance surrounding a vessel. These ROIs would be part of the ROI assigned previously for all the tissue areas (i.e., ROI 2).
    - i. In this instance, ROI 3 was assigned to the proper vessel area consisting of the 15 pixels around the CD31 marker (Figure 1D).
    - ii. Multiple ROIs can then be generated at a certain assigned distance to create segmented areas of the TME (e.g., ROI 4 and ROI 5, at 50 and 100 microns from the vessel (ROI 3), respectively).
    - iii. Perivascular regions can be consolidated into a designated ROI to account for heterogeneity in vascular density (Figure 1E).

### Part 3: Nuclei detection

⌚ Timing: 3 h

Following vessel labeling, this part focuses on detecting nuclei (Figures 2A and 3A; Methods video S3) via the detection of the DAPI staining to facilitate subsequent cell phenotyping (Figures 2A and 3A, B) (Cf. Steps 5 and 6). This process uses a pre-trained deep-learning algorithm commercially available within the Visiopharm Software.

8. Within the new App button, select the 'Nuclei Detection, AI (Fluorescence)' App.
  - a. Use three image classes to identify a cell: Class 1 for 'Background', Class 2 for 'Nucleus', and Class 3 for 'Border.'

**Note:** The 'Border' label is precisely drawn around objects in the image labeled as 'Nucleus', representing areas with strong circular DAPI signals, ensuring differentiation between cells. Everything else is assigned a 'Background' label.

**Note:** As the App is pre-trained, additional training areas like those mentioned in "Step 1: tissue segmentation" are not necessary. If desired, more training data can be added.

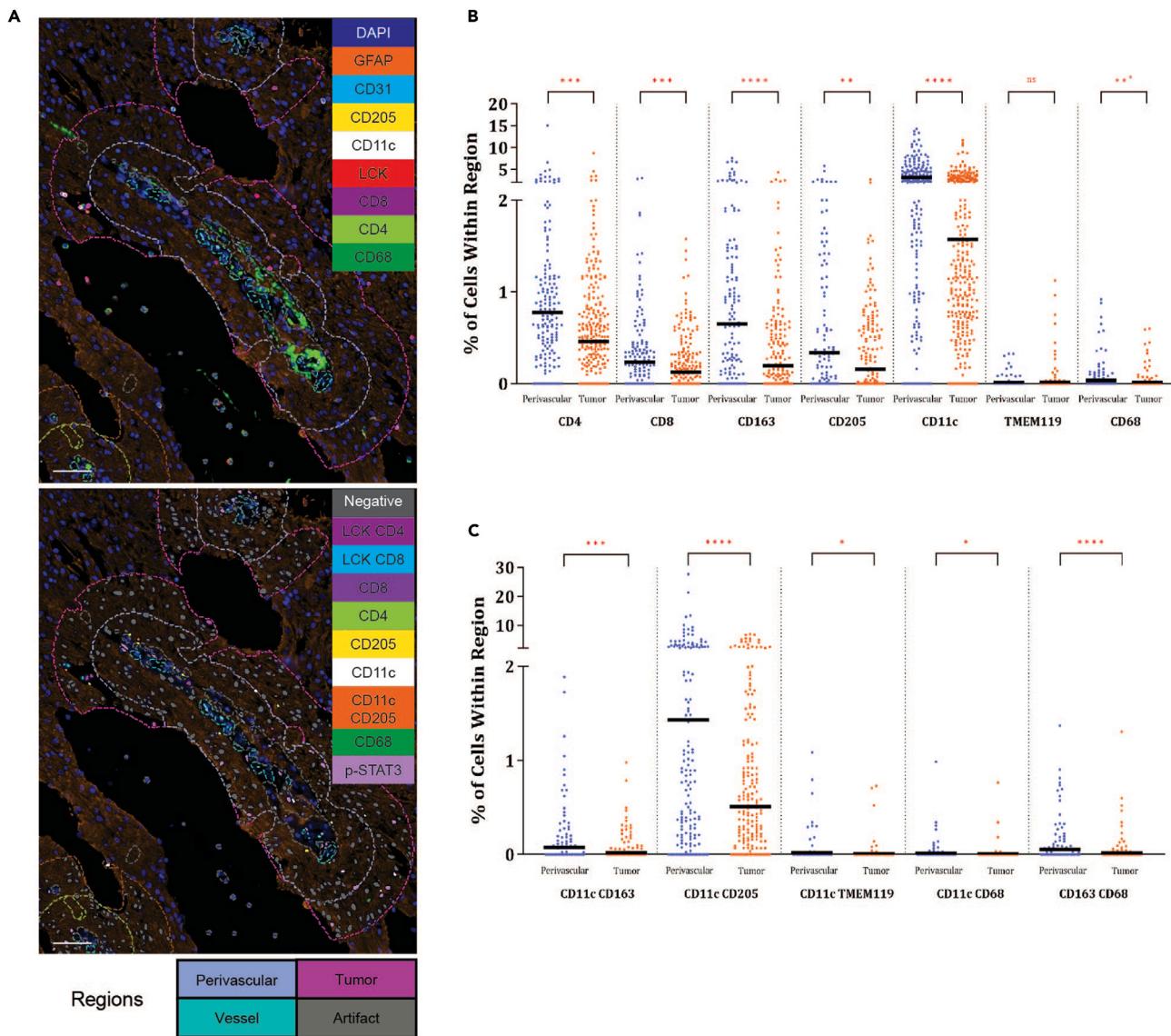
- b. For 'Input' settings, set up a magnification of 20X and a field of view of  $512 \times 512$  pixels.
- c. In 'Advanced Sampling' settings, apply a  $25 \mu\text{m}$  counting frame and deselect the option to discard labels.
- d. For 'Regions to Analyze', select ROI 2 including tissue areas with labeled vessels identified in the previous App (Cf. Part 2: Vessel Segmentation).
- e. For 'Classification' settings, used the Deep Learning method with the U-Net architecture.
- f. In the 'AI Architect' menu, adjust the input size to  $512 \times 512$  pixels, the learning rate to 1.0E-5, and the minibatch size to 2.

**Note:** The algorithm is then trained using only the DAPI channel and limited pixel intensities to the empirically determined range of 6500–65000 to minimize noise.

9. Refining steps within the 'Post-Processing' section of the App:

**Note:** For better visualization of the subsequent steps the 'Border' and 'Background' labels can be shifted to the 'Clear' label.

- a. Add a 'Separate Objects' step to refine each nucleus/cell identification separately from its surroundings.
  - i. The object label should be the 'Nucleus'.



**Figure 2. Representative images and plots of nuclei detection and cell composition**

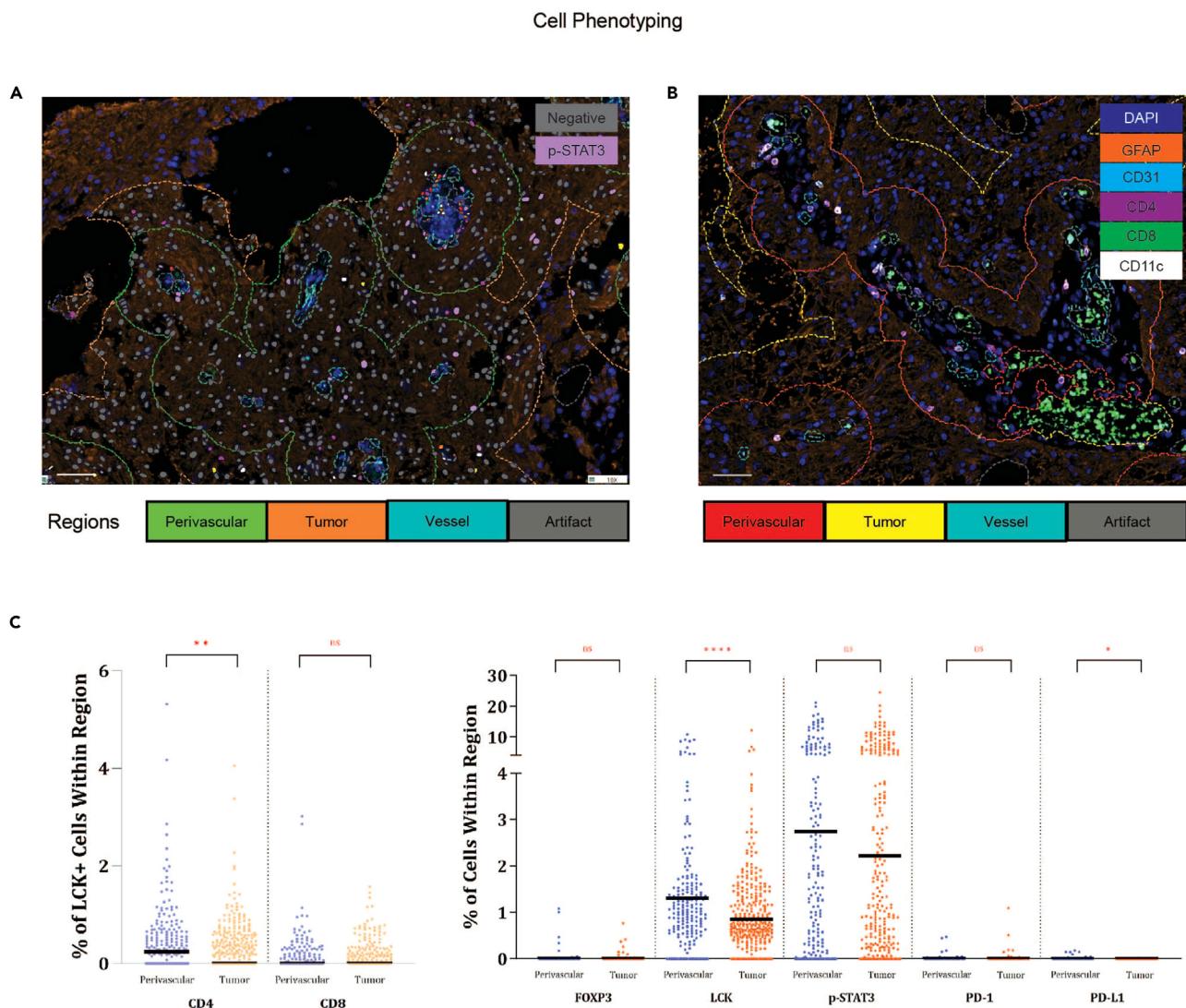
(A) Representative images of pilocytic astrocytoma showing the tumor segmentation of the perivascular region (dashed blue line) and the tumor (dashed purple line). Cell populations are then identified based on the presence of DAPI+ nuclei and are assigned a phenotype based on the marker defined in the figure (10x magnification, 200  $\mu\text{m}$  scale bars).

(B) Quantification of immune cell populations based on the expression of a single marker as a function of regions (perivascular versus tumor). Percentage of cells is calculated from the total number of DAPI+ cells.

(C) Quantification of immune cell populations based on the expression of two markers as a function of regions (perivascular versus tumor). % of cells is calculated from the total number of DAPI+ cells. ns p-value >0.05; \* p-value <0.05, \*\* p-value <0.01, \*\*\* p-value <0.001, \*\*\*\* p-value <0.0001.

- ii. The separator label should be the ‘Separator image class’ created within the App in the ‘Image Classes for Output’ section.
- iii. The background label should be set to ‘All except object’.
- iv. The object’s diameter should be 15  $\mu\text{m}$  and the option to separate the object’s surroundings should be a surrounding width of 3  $\mu\text{m}$ .

**Note:** It is preferable not to select the option that assumes that objects are ‘elliptic’. Rather, it is better to select the ‘F(3): Boundary’ heatmap option with a value of 0.7.



**Figure 3. Representative images and plots of immune cell phenotyping**

(A) Representative multiplex image showing the nuclear staining of immune suppressive p-STAT3+ cells (purple) as a function of tumor region and relative to total DAPI+ cell count (gray) (10x magnification, 200  $\mu$ m scale bar).

(B) Representative multiplex image analyzing LCK expression, as a marker for T cell activation, as a function of immune cell lineage, and as a function of regions (10x magnification, 200  $\mu$ m scale bar).

(C) Quantification of immune effector functions based on regional location of perivascular versus tumor. % of cells is calculated from the total number of total DAPI+ cells. ns p-value >0.05; \* p-value <0.05; \*\* p-value <0.01; \*\*\* p-value <0.001; \*\*\*\* p-value <0.0001.

- To take into consideration the cytoplasm and membrane of the cell, a step to dilate the 'Nucleus' label for 3 pixels is standardized and can be implemented.
  - Disable the option to avoid merging objects.
  - Chose a 'Separator' as the exclusion label.
- In 'Change Surrounded' step, convert the 'Clear' label surrounded by 'Nucleus' to the 'Nucleus' label, with 0.1 coverage. The image border should be included as well.
- Next, change the 'Separator' label to the 'Clear' label.
- Adding a 'Change by Shape' step can be beneficial, taking into consideration all nuclei and cell shapes, especially within central nervous system tissue.
  - Use the 'Area' label in this setting.

- ii. Replace the 'Nucleus' label by the 'Clear' label with a maximum chosen value of  $10 \mu\text{m}^2$ .
  - f. Lastly, apply the counting frame to all labels.
10. To collect relevant data from each cell identified, edit the 'Output Variable' section of the App to avoid the use of ROIs.
- a. Create a new variable named 'Total Nuclei' using the 'Classified image' as the input source.

**Note:** It is configured to calculate the 'Count' of the occurrences of the 'Nucleus' label per image. The output level is set to the image avoiding the use of ROI.

- b. Next, create the variable named 'Object' using the 'Classified image' as input.

**Note:** This should calculate the 'Object Info' for all labels on a per-object basis. The output level is set per object this time, without using the ROI.

- i. Create another variable 'Multiplex', using the 'Original image' as input.

**Note:** This variable calculates the 'Multiplexing' for all labels on a per-object basis. The output level is set per object, without using the ROI.

- ii. Create an 'Area' variable, again using the 'Classified image' as input.

**Note:** This variable calculates the area of the 'Nucleus' label on a per-object basis for the output level.

- iii. Finally, create a variable named 'Distance' using the 'Classified image' as input.

**Note:** The 'Object Distance' from the 'Nucleus' label to the 'Vessel' label (also referred to as 'Label 005') can be calculated on a per-object basis.

**Note:** A maximum distance limit of 1 mm can be set and it can be increased according to the computer's processing power and any other time constraints.

#### Part 4: Data extraction

⌚ Timing: 12 h

In this part, the data already generated from the previous steps can be extracted and used in R Studio to normalize the data.

**Note:** This part is optional.

11. Within the 'Database', select all images for batch analysis.
  - a. The Apps should be selected to run sequentially beginning with tissue segmentation followed by vessel identification and segmentation, and finally nuclei detection.
  - b. Initiate batch analysis and wait until completion.
12. After completion, for each image, save the data generated.
  - a. Within the 'Database', select an image and click on 'Export selected results (.tsv)' to extract its data. This action will generate a TSV file for each image.

**Note:** All TSV files should be saved in the same folder before moving to the next step.

13. Download and install the latest version of R Studio.

- a. Create a new R script and save it in the folder containing the TSV files.
- b. In R Studio, set up the working directory to this folder.
- c. Load the tidyverse library and convert each TSV file into CSV files (cf. Code 1).
- d. Load all the newly created CSV files into a list.

Code 1

```
library(tidyverse)

tsv_files <- list.files(pattern = "\\.tsv$")
```

- e. Create a function to normalize columns with multiplex data by dividing the signal intensity of the marker of interest by the signal intensity of its associated fluorophore (in this instance Cy5 or TRITC) (cf. Code 2).

**Note:** The resultant columns show the relative intensity of the marker against autofluorescence, controlling for artifacts like nonspecific antibody binding due to red blood cells.

- i. Remove rows with 'NA' values indicating cells outside the distance limit.
- ii. Preserve only the columns showing the normalized signal intensity, area, and distance.
- iii. Apply this function to each CSV file, generating a new processed CSV file.

Code 2

```
process_data_list <- lapply(tsv_files, function(file) {

  # Process each TSV file

  df <- read_tsv(file) %>%
    mutate(
      GFAP = 'Multiplex - GFAP_D' / 'Multiplex - Cy5',
      CD31 = 'Multiplex - CD31' / 'Multiplex - TRITC',
      CD4 = 'Multiplex - CD4 EPR6854' / 'Multiplex - Cy5',
      CD8 = 'Multiplex - CD8' / 'Multiplex - TRITC',
      CD68 = 'Multiplex - CD68 PG-M1' / 'Multiplex - TRITC',
      CD11c = 'Multiplex - CD11c' / 'Multiplex - Cy5',
      CD163 = 'Multiplex - CD163' / 'Multiplex - Cy5',
      CD205 = 'Multiplex - CD205_D' / 'Multiplex - Cy5',
      TMEM119 = 'Multiplex - TMEM119_C' / 'Multiplex - Cy5',
      P2RY12 = 'Multiplex - P2RY12' / 'Multiplex - Cy5',
      FOXP3 = 'Multiplex - FOXP3 236A/E6' / 'Multiplex - TRITC',
      pSTAT3 = 'Multiplex - p-STAT3_C' / 'Multiplex - Cy5',
      PD1 = 'Multiplex - PD-1 EPR4877(2)' / 'Multiplex - TRITC',
      PDL1 = 'Multiplex - PD-L1' / 'Multiplex - Cy5',
      TIM3 = 'Multiplex - TIM-3' / 'Multiplex - TRITC',
      LAG3 = 'Multiplex - LAG-3' / 'Multiplex - Cy5',
      GRZMB = 'Multiplex - GRZMB' / 'Multiplex - TRITC',
      LCK = 'Multiplex - LCK' / 'Multiplex - Cy5',
```

```

HLADR = 'Multiplex - HLA-DR_B' / 'Multiplex - TRITC',
) %>%
select(Area, Distance, GFAP, CD31, CD4, CD8, CD68, CD11c, CD163, CD205, TMEM119, P2RY12,
FOXP3, pSTAT3, PD1, PDL1, TIM3, LAG3, GRZMB, LCK, HLADR) %>%
filter(!is.na(Distance))
processed_file_name <- sub("\\.tsv$", "_Processed.tsv", file)
write_tsv(df, processed_file_name)

```

**Note:** The folder corresponding to the working directory should now contain all generated CSV files. Each file contains data for all cells identified by the nuclei detection App contained within an image. When exported into Excel, each row in a file corresponds to a specific cell.

**Note:** The column names in the TSV files, used to calculate normalized multiplex values, can vary significantly. The code provided is just an example of how those column names may look. These names depend on the markers used in a specific project and how the markers are associated with the fluorophore channels. In this instance, the column name for each marker intensity value is divided by the column name for TRITC or Cy5 intensity values accordingly.

⚠ CRITICAL: Selecting the appropriate fluorophore, TRITC, or Cy5, for association with your marker of interest is important for accurate normalization. This choice varies depending on the specifics of the staining protocol.

## Part 5: Machine learning labeling

⌚ Timing: 24 h

This part allows the user to train the Visiopharm machine learning algorithm while integrating the data into other modalities of analysis like R Studio and subsequently Spyder for more accurate results (cf. part 6).

**Note:** This part is optional.

14. Within the 'Database' in the Visiopharm software, remove all columns containing data from all images, and delete ROIs and labels.

**Note:** This data can be preserved by creating a copy of the folder containing these images and performing machine-learning labeling in the duplicated folder.

15. To start, select a marker of interest to generate datasets for training the machine learning models.
  - a. In the Visiopharm viewer, enable the channel corresponding to this marker along with other channels that could help identify cells expressing this marker of interest.

**Note:** These markers include DAPI and known cell lineage markers. The 'Min/Max %' intensity settings are chosen accordingly for an effective visualization of these markers.

- b. Next, identify all areas where cells are positive for the chosen marker in all images and use ROI 2 to outline these areas.

16. Run the nuclei detection App (created in part 3) on these areas and save the data generated.
17. Create a new App for labeling and use four image classes in this setting: Class 1 for 'Background', Class 2 for 'Cell', Class 3 for '1', and Class 4 for '0'.
  - a. In the 'Input' settings, under 'Advanced Sampling', set up a magnification of 10X and a field of view of 512 × 512 pixels and select the default settings for all other options.
  - b. In the 'Regions to Analyze' settings chose ROI 2.
  - c. For 'Classification' settings, select 'No Classification'.
18. Edit the 'Output Variable' section of the App to collect relevant data from each cell.
  - a. Create the variable 'Multiplex' using the 'Original image' as input.

**Note:** This variable calculates the 'Multiplexing' inputs for all labels on a per-object basis. The output level is set per object, without using the ROI.

- b. Create an 'Area' variable next, again using the 'Classified image' as input.

**Note:** This variable calculates the area of all labels on a per-object basis for the output level.

- c. Create the variable named ObjectInfo also using the 'Classified image' as input.

**Note:** This should calculate the 'Object Info' for all labels on a per-object basis.

19. By accessing the Wheel for labels, tag the objects classified as 'Cell' and generated by the nuclei detection App (Step 16).
  - a. In 'more settings', by accessing the 'Draw' section, select 'Edit drawn objects(s)'.
  - b. Hover over cells and label them '1' if you assess them to be positive for the marker of interest, or '0' if negative.
  - c. After labeling all cells within the ROIs, run the App.

**Note:** It is advisable to label all positive cells first, then change all other objects marked as 'Cell' to '0' to streamline the process. In addition, post-processing steps can be used to change 'Cell' objects based on intensity, assigning them the label '1' or '0' depending on their marker average intensity. Once the App is run, you can use the wheel to adjust the labels as necessary. When all cells have been labeled, disable these post-processing steps before running the App to extract the output variables.

20. Save the data generated by the Apps for each image.
  - a. Within the 'Database', select the folder holding all the images and select 'Batch Export.'
  - b. Choose to export the results as 'Numerical variables (.tsv)'.
  - c. Select 'Export a single file for all images' to combine all labeled data from every image into one file; this is important for subsequent training.
21. After clearing all labels and ROIs from every image, repeat steps 15–20 to phenotype each additional marker of interest.

**⚠ CRITICAL:** All these TSV files generated should be stored in a separate folder, distinct from the group of TSVs generated in the previous part 4.

**Note:** The percent composition of cells labeled '1' to '0' should be relatively similar for each marker. It's important to avoid a disproportionate representation of cells labeled '0' while training the algorithm to make sure you increase the chances of it identifying '1' labeled cells. Under-sampling the '0' class might be necessary so that cases labeled '1' make up at least 10% of all labeled cells.

22. Conduct the next steps in R Studio to process each TSV file.

- a. Create a new R script and save it in the folder containing these TSV files.
  - b. Set your working directory to this folder.
- Load the tidyverse library and put these TSV files in a list to be processed (cf. Code 3).

Code 3

```
library(tidyverse)

tsv_files <- list.files(pattern = "\\.tsv$")
```

- c. Create a function to first rename the 'ObjectInfo - LabelName' columns to 'LabelName' only for better clarity (cf. Code 4).

**Note:** Each of these columns, containing 0s or 1s, indicates whether a cell (represented by a single row) is positive or negative for a specific marker. It will then normalize columns similarly.

- i. Remove the rows with the 'Cell' value located in the 'LabelName' column as these represent unlabeled cells.
- ii. Only keep columns with the rest of labels and normalized signal intensities (cf. Code 4).

Code 4

```
process_data_list <- lapply(tsv_files, function(file) {

  # Process each TSV file
  df <- read_tsv(file) %>%
    rename(LabelName = 'ObjectInfo - LabelName') %>%
    mutate(
      GFAP = 'Multiplex - GFAP_D' / 'Multiplex - Cy5',
      CD31 = 'Multiplex - CD31' / 'Multiplex - TRITC',
      CD4 = 'Multiplex - CD4 EPR6854' / 'Multiplex - Cy5',
      CD8 = 'Multiplex - CD8' / 'Multiplex - TRITC',
      CD68 = 'Multiplex - CD68 PG-M1' / 'Multiplex - TRITC',
      CD11c = 'Multiplex - CD11c' / 'Multiplex - Cy5',
      CD163 = 'Multiplex - CD163' / 'Multiplex - Cy5',
      CD205 = 'Multiplex - CD205_D' / 'Multiplex - Cy5',
      TMEM119 = 'Multiplex - TMEM119_C' / 'Multiplex - Cy5',
      P2RY12 = 'Multiplex - P2RY12' / 'Multiplex - Cy5',
      FOXP3 = 'Multiplex - FOXP3 236A/E6' / 'Multiplex - TRITC',
      pSTAT3 = 'Multiplex - p-STAT3_C' / 'Multiplex - Cy5',
      PD1 = 'Multiplex - PD-1 EPR4877(2)' / 'Multiplex - TRITC',
      PDL1 = 'Multiplex - PD-L1' / 'Multiplex - Cy5',
      TIM3 = 'Multiplex - TIM-3' / 'Multiplex - TRITC',
      LAG3 = 'Multiplex - LAG-3' / 'Multiplex - Cy5',
      GRZMB = 'Multiplex - GRZMB' / 'Multiplex - TRITC',
      LCK = 'Multiplex - LCK' / 'Multiplex - Cy5',
    )
})
```

```

HLADR = 'Multiplex - HLA-DR_B' / 'Multiplex - TRITC',
) %>%
select(LabelName, GFAP, CD31, CD4, CD8, CD68, CD11c, CD163, CD205, TMEM119, P2RY12,
FOXP3, pSTAT3, PD1, PDL1, TIM3, LAG3, GRZMB, LCK, HLADR) %>%
filter(LabelName == "Cell")
processed_file_name <- sub("\\.tsv$", "_Processed.tsv", file)
write_tsv(df, processed_file_name)
})

```

- iii. Apply this function to each TSV file within the folder, generating new processed TSV files respectively.

**Note:** The working directory folder should now contain these new TSV files. Each file contains data for all labeled cells in a sample identified by the nuclei detection App. When exported into Excel, each row in a file corresponds to a specific cell.

### Part 6: Random forest classifier creation using spyder

⌚ Timing: 12 h

This part allows the use of the generated data files in steps 4 and 5 and integrates them into Spyder to generate scores indicating the accuracy of the results and cell phenotypes.

**Note:** This part is optional.

23. Download and install the latest version of the Spyder IDE. Once installed, launch Spyder.

- Create a new script file.
  - On your computer, create a new folder, where the new script file is saved.
  - Add the processed TSV files generated in parts 4 and 5 also to this folder.
  - Within Spyder settings, set the working directory to this newly created folder.
- Load necessary libraries for machine learning functions (cf. Code 5).

```

Code 5

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import precision_score, recall_score, f1_score, accuracy_score,
roc_auc_score

```

24. First, load into the script the processed TSV file, generated in part 5, of the marker of interest.
  - Divide the dataset based on values in the 'LabelName' column, separating groups with labels '1' and '0' (cf. Code 6).

```

Code 6

data = pd.read_csv("ExampleMarker1_Processed.tsv", sep='\t')

```

```
X = data.drop("LabelName", axis=1)
y = data["LabelName"]
```

- b. Divide this labeled dataset into training and testing sets (cf. Code 7).

**Note:** The standard split is 80% for training and 20% for testing. However, this ratio can be adjusted according to the users' specific needs.

- c. Create a random forest classifier next, with a max depth of 4 to prevent overfitting (cf. Code 8). You may add other settings to enhance the classifier's robustness.

Code 7

```
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
state=42)
```

Code 8

```
clf = RandomForestClassifier(max_depth=4, random_state=42)
clf.fit(x_train, y_train)
```

- d. Predict the positivity of the marker on the test set and calculate key evaluation metrics: precision, recall, F1 score, accuracy, and area under the curve (AUC) (cf. Code 9).

Code 9

```
y_pred = clf.predict(x_test)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

accuracy = accuracy_score(y_test, y_pred)

auc = roc_auc_score(y_test, clf.predict_proba(x_test)[:, 1])
```

**△ CRITICAL:** The metrics help determine the effectiveness of the random forest classifier and the accuracy of the data generated. If these scores are not satisfactory, the protocol in step 5 for that specific marker can be repeated. This may involve adding new examples of positive cells '1' or under-sampling negatively labeled cells '0' due to potential class imbalances in the dataset. For low expressed markers, it is best to try and collect as many positive examples as possible to make sure the composition of positive and negative cells is between 40% and 60%, with all scores (cf. note below) above 0.7 ([Table 1](#)). Then codes 5–9 should be run again all at the same time.

**Note:** 'Precision' quantifies the accuracy of positive prediction. 'Recall' measures the accuracy of how many of the actual positive cases the model has been able to predict. The 'F1 score' is the harmonic mean value of the 'Precision' and 'Recall'. 'Accuracy' measures the proportion of true positives and true negatives identified by the model. Precision, recall, and F1 scores are more reliable of an effective model than accuracy alone. 'AUC' (area under the curve) evaluates the model's ability to differentiate between classes. For AUC, any score above 0.5 indicates performance better than random guessing. Scores ranging from

0.9–1 are excellent, 0.8–0.89 are good, 0.7–0.79 are fair, and anything below 0.7 is poor (Table 1).

25. Load one of the processed TSV files from part 4, containing the unlabeled dataset of all cells (cf. Code 10).

**Note:** Ensure that the features in this unlabeled set match the features used in the training set for part 5 for that particular marker.

Code 10

```
Sample1_Labeled = pd.read_csv("Sample1_ProCESSED.tsv", sep='\t')  
unlabeled_features = Sample1_Labeled[X.columns]
```

- a. Use the classifier to add a new column to the unlabeled dataset, while choosing its positivity for the marker of interest (cf. Code 11).

Code 11

```
unlabeled_predictions = clf.predict(unlabeled_features)  
Sample1_Labeled['ExampleMarker1_Positive'] = unlabeled_predictions
```

26. Repeat steps 24 and 25 for every marker analyzed in each sample.

**Note:** Once all the markers have been processed, save the sample with the newly added columns as a new TSV file (Cf. Code 12).

Code 12

```
Sample1_Labeled.to_csv("Sample1_Labeled.tsv", sep='\t', index=False)
```

**Note:** Remember to modify column names in codes 10–11 to add a new positivity column specific to each marker.

27. Implement step 26 for each sample in your study and update the sample names in codes 10–11 accordingly. This ensures the creation of a separate TSV file for each sample.

### Part 7: Cell phenotype counting using R studio

⌚ Timing: 2 h

This part allows the counting and graphical representation of all cell phenotypes and data generated in the previous steps (Figures 2B, 2C, and 3C).

**Note:** This step can follow either parts 6 or 8.

28. Using R Studio, process the TSV files generated in step 6, and the Excel datasheets generated in step 8. Within R Studio:
  - a. Create a new R script and place it in a newly created folder.
  - b. Add either the TSV files or Excel datasheets generated in previous steps into this folder.
  - c. Set your working directory to this folder.

d. Load the tidy verse library and put these files in a list to be processed (cf. Code 13).

Code 13

```
library(tidyverse)
tsv_files <- list.files(pattern = "\\.tsv$")
```

29. To hold all phenotypes, create different vectors for different analyses, and initialize count matrices to count these phenotypes in each sample (cf. Code 14).

Code 14

```
all_phenotypes1 <- c("CD4+", "CD8+", "CD68+", "CD11c+", "CD163+", "CD205+",
                      "TMEM119+", "P2RY12+", "FOXP3+", "p-STAT3+", "PD-1+",
                      "TIM-3+", "LAG-3+", "GRZMB+", "LCK+", "HLADR+", "Other")
all_phenotypes2 <- c("CD11c+CD163+", "CD11c+CD205+", "CD11c+TMEM119+",
                      "CD11c+CD68+", "CD163+CD68+", "CD11c+CD163-CD205-TMEM119-CD68-",
                      "CD4+LCK+", "CD8+LCK+", "CD4+p-STAT3+", "CD8+p-STAT3+",
                      "CD4+p-STAT3-", "CD8+p-STAT3-", "CD8+FOXP3+", "Other")
Phenotype1_Counts <- matrix(0, nrow = length(tsv_files), ncol = length(all_phenotypes1),
                           dimnames = list(NULL, all_phenotypes1))
Phenotype2_Counts <- matrix(0, nrow = length(tsv_files), ncol = length(all_phenotypes2),
                           dimnames = list(NULL, all_phenotypes2))
```

30. Create a function to process each data file (cf. Code 15).

Code 15

```
for (idx in seq_along(tsv_files)) {
  file <- tsv_files[idx]
  df <- read_tsv(file)
  df <- df %>%
    mutate(Phenotype1 = case_when(
      CD4_Positive == 1 ~ "CD4+",
      CD8_Positive == 1 ~ "CD8+",
      CD68_Positive == 1 ~ "CD68+",
      CD11c_Positive == 1 ~ "CD11c+",
      CD163_Positive == 1 ~ "CD163+",
      CD205_Positive == 1 ~ "CD205+",
      TMEM119_Positive == 1 ~ "TMEM119+",
      P2RY12_Positive == 1 ~ "P2RY12+",
      FOXP3_Positive == 1 ~ "FOXP3+",
      pSTAT3_Positive == 1 ~ "p-STAT3+"
```

```

PD1_Positive == 1 ~ "PD-1+",
TIM3_Positive == 1 ~ "TIM-3+",
LAG3_Positive == 1 ~ "LAG-3+",
GRZMB_Positive == 1 ~ "GRZMB+",
LCK_Positive == 1 ~ "LCK+",
HLADR_Positive == 1 ~ "HLADR+",
TRUE ~ "Other"), # Default case if no other conditions are met
Phenotype2 = case_when(
  CD11c_Positive == 1 & CD163_Positive == 1 ~ "CD11c+CD163+",
  CD11c_Positive == 1 & CD205_Positive == 1 ~ "CD11c+CD205+",
  CD11c_Positive == 1 & TMEM119_Positive == 1 ~ "CD11c+TMEM119+",
  CD11c_Positive == 1 & CD68_Positive == 1 ~ "CD11c+CD68+",
  CD163_Positive == 1 & CD68_Positive == 1 ~ "CD163+CD68+",
  CD11c_Positive == 1 & CD163_Positive == 0 & CD205_Positive == 0 &
    TMEM119_Positive == 0 & CD68_Positive == 0 ~ "CD11c+CD163-CD205-TMEM119-CD68-",
  CD4_Positive == 1 & LCK_Positive == 1 ~ "CD4+LCK+",
  CD8_Positive == 1 & LCK_Positive == 1 ~ "CD8+LCK+",
  CD4_Positive == 1 & pSTAT3_Positive == 1 ~ "CD4+p-STAT3+",
)

```

**Note:** Especially for the TSV file generated in part 6, this function allows the addition of phenotype columns based on the values in the positivity columns, and thus, multiple phenotype columns can be generated. It is essential to ensure that all phenotypes listed in the vectors are accounted for with conditional statements to be added to a phenotype column.

⚠ **CRITICAL:** Make sure 'Other' is a phenotype in both the vectors and conditional statements.

- Include a phenotype name and conditional statement for 'Other' to categorize all other unlabeled cells.
- Update the count matrices to reflect the number of each unique phenotype in its respective column for each sample.

**Note:** This function will create a new processed TSV file containing the distance, phenotype columns, and count matrices.

Code 15 continued

```

CD8_Positive == 1 & pSTAT3_Positive == 1 ~ "CD8+p-STAT3+",
CD4_Positive == 1 & pSTAT3_Positive == 0 ~ "CD4+p-STAT3-",
CD8_Positive == 1 & pSTAT3_Positive == 0 ~ "CD8+p-STAT3-",
CD8_Positive == 1 & FOXP3_Positive == 1 ~ "CD8+FOXP3+",
# Add more conditions here for other phenotypes

```

```

TRUE ~ "Other")
) %>%
select(Distance, Phenotype1, Phenotype2)

phenotype1_counts <- table(factor(df$Phenotype1, levels = all_phenotypes1))
phenotype2_counts <- table(factor(df$Phenotype2, levels = all_phenotypes2))

Phenotype1_Counts[idx, names(phenotype1_counts)] <- as.numeric(phenotype1_counts)
Phenotype2_Counts[idx, names(phenotype2_counts)] <- as.numeric(phenotype2_counts)
processed_file_name <- sub("\\.tsv$", "_Processed.tsv", file)
write_tsv(df, processed_file_name)
}

```

**Note:** To assign a phenotype as a string value in its respective phenotype column based on the corresponding positivity values in the same row, use multiple conditionals. These conditionals should be based on whether the value from the random forest classifiers in Step 6, in the positivity column, is either 1 or 0. Different phenotype columns can be generated by different ways of looking at the data. As an example, one phenotype column may count the number of cells positive for one marker and another phenotype column may highlight selected phenotypes of interest based on multiple conditionals.

31. With the processed TSV files and count matrices, perform different analyses on your data.
  - a. For easier subsequent analysis, bind all these files together for data analysis (cf. Code 16).

```

Code 16

processed_files <- list.files(pattern = "_Processed\\*.tsv$")
Combined_Processed_Data <- processed_files %>%
map_dfr(~read_tsv(.x))

write_tsv(Combined_Processed_Data, "Combined_Processed_Data.tsv")

```

- b. As an example, generate and bin the counts of CD11c+ cells away from a vessel in the Phenotype1 column every 50 µm using the combined processed data (cf. Code 17).

```

Code 17

write_tsv(Combined_Processed_Data, "Combined_Processed_Data.tsv")

df_visual <- read_tsv("Combined_Processed_Data.tsv")
cd11c_positive <- df_visual %>%
filter(Phenotype1 == "CD11c+")
max_distance <- max(cd11c_positive$Distance, na.rm = TRUE)
bin_width <- 50
bins <- seq(0, max_distance + bin_width, by = bin_width)
ggplot(cd11c_positive, aes(x = Distance)) +
geom_histogram(breaks = bins, color = "black", fill = "blue") +
labs(title = "Histogram of CD11c+ Cells Count by Distance",

```

```
x = "Distance",
y = "Count of CD11c+ Cells") +
theme_minimal()
```

- c. Another example, compare how the composition of CD11c+ versus CD11c- cells changes based on distance away from a vessel (cf. Code 18).
  - i. For each phenotype, create a bar with two segments stacked on each other.

**Note:** Each segment's height represents the proportion of that phenotype in the corresponding distance bin, relative to the total count of cells in that bin.

```
Code 18

df <- read_tsv("Combined_Processed_Data.tsv")

max_distance <- max(df$Distance, na.rm = TRUE)

bin_width <- 50

bins <- seq(0, max_distance + bin_width, by = bin_width)

df_binned <- df %>%
  mutate(Bin = cut(Distance, breaks = bins, include.lowest = TRUE, right = FALSE)) %>%
  group_by(Bin, Phenotype1) %>%
  summarise(Count = n(), .groups = 'drop') %>%
  # Calculate the total count per bin
  mutate(Total = sum(Count)) %>%
  # Calculate the percentage
  mutate(Percent = (Count / Total) * 100)

ggplot(df_binned, aes(x = Bin, y = Percent, fill = Phenotype1)) +
  geom_bar(stat = "identity") +
  labs(title = "Percentage Composition of CD11c+ vs CD11c- Cells by Distance",
       x = "Distance Bin",
       y = "Percentage (%)") +
  scale_fill_brewer(palette = "Set1") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

#### Part 8: Visiopharm phenoplex feature

⌚ Timing: 4 h

In this step, the 'Phenoplex' feature within the Visiopharm software is implemented. The phenotypes of interest are predetermined and encompass various categories including those positive for a single biomarker (e.g., TMEM119+, CD4+, or CD11c+), double positive (e.g., CD163+CD11c+, CD68CD163+, or CD205+CD11c+), and functional markers (e.g., CD8+LCK+, CD11c+pSTAT+, or CD205+CD11c+PDL1+).

**Table 1.** The parameters and scores generated for each of the markers under analysis per image

Markers	Number of cells under training	Percentage of positive cells	Percentage of negative cells	Accuracy	Precision	Recall	F1 score	AUC
CD4	1082	51.20%	48.80%	0.86347	0.75546	0.99088	0.8573	0.9904
CD8	1580	41.58%	58.42%	0.93176	0.98185	0.96845	0.9751	0.98691
CD68	13617	42.51%	57.49%	0.87877	0.97656	0.72655	0.8332	0.95556
CD11c	44093	46.00%	54.00%	0.7586	0.71357	0.7976	0.75325	0.75217
CD163	13078	51.87%	48.13%	0.81621	0.7814	0.94862	0.85693	0.96392
CD205	13266	49.33%	50.67%	0.80667	0.78428	0.86281	0.82167	0.84274
TMEM119	6504	53.18%	46.82%	0.74214	0.94066	0.72237	0.81719	0.9723
P2RY12	5006	44.71%	55.29%	0.99508	0.93167	0.75961	0.83689	0.99773
FOXP3	2045	48.80%	51.20%	0.70165	0.94464	0.91206	0.92806	0.94255
pSTAT3	3056	47.09%	52.91%	0.91797	0.93138	0.72221	0.81357	0.99148
PD1	1020	50.59%	49.41%	0.80718	0.73476	0.95893	0.83201	0.8685
TIM3	4032	51.54%	48.46%	0.88637	0.79927	0.71907	0.75705	0.84187
LAG3	2339	47.11%	52.89%	0.79298	0.79755	0.91888	0.85393	0.84017
GRZMB	3560	44.04%	55.96%	0.89063	0.96616	0.84166	0.89962	0.93734
LCK	3045	49.56%	50.44%	0.73576	0.91397	0.92824	0.92105	0.95191
HLADR	850	49.53%	50.47%	0.86782	0.93129	0.84814	0.88777	0.98159

**Note:** This step can replace parts 4–6 and be implemented directly after part 3 for faster data generation.

**Note:** New updates and refinements are regularly added to this feature software that should be implemented and trained for using the tools offered by the Visiopharm company. The analysis for this protocol was conducted using version 12.2022 of the software.

32. Enable phenotyping through the step involving the detection of nuclei (cf. part 3).
  - a. Use the pre-trained deep-learning algorithm in the Visiopharm software.
  - b. For training purposes, use three image classes: Background, Nuclei, and Border corresponding to classes 1, 2, and 3, respectively.
    - i. Precisely draw the Border label around objects labeled as Nuclei, representing areas with strong DAPI signals, thereby ensuring differentiation between cells, and assign all other regions as Background label.
    - ii. To apply the classifier effectively, choose optimal inputs of 20X magnification to capture intricate details, a field of view measuring 512 × 512 pixels, and a counting frame of 10 µm within ROIs 10–255.
    - iii. Train the classifier (U-Net architecture; input size = 512 × 512 pixels; learning rate = 1.0E-5; mini-batch size = 2; 100000 iterations) exclusively on the DAPI channel with pixel intensities ranging from 5000 to 65000.
    - iv. During post-processing, clear both the Border and Background labels.
    - v. Subsequently, separate Nuclei objects using a Separator label corresponding to class 4 with a surroundings width of 2 µm assuming the smallest objects to be separated are 15 µm in diameter.
    - vi. Dilate the resulting Nuclei label by 3 pixels to accurately resemble cell boundaries.
    - vii. Finally, clear the Separator label and Nuclei objects with an area of 5 µm<sup>2</sup> or less.
33. Generate and name image classes for output based on the respective biomarkers they represent (in this instance 45 image classes for output were highlighted).
  - a. Classify Nuclei objects into the appropriate phenotype class based on the mean pixel intensity of various biomarker signals.
  - b. To ensure accuracy, select specific pixel threshold values to minimize false positive and false negative results, and apply them consistently across all images.

**Note:** Consider including 100% of pixels while calculating the mean intensities, as previous applications had already removed areas that would introduce noise.

- c. Following the application of the algorithm, generate an object info output variable associating each cell with an image class and the corresponding ROI.
34. For quantification, download the Excel table and employ R programming to generate counts and percentages of all cell phenotypes per perivascular and tumor area similar to part 7 (Cf. Codes 13–18).

**Note:** Pause Points and breaks can be taken at any step of this protocol for any length of time. Only make sure to save data at the end of each step.

## EXPECTED OUTCOMES

Immune cell populations that originate from the periphery including CD4+ T cells, CD8+ T cells, CD163+ macrophages, CD205+ myeloid blood dendritic cells, CD11c+ dendritic cells, and CD68+ monocytes are all preferentially enriched in the perivascular space relative to the tumor (Figures 2A and 2B). Only TMEM119+ microglia, which are intrinsic to the CNS, demonstrate no difference in the frequency in the perivascular region relative to the tumor. Similar findings were also observed when various immune cell populations were characterized based on more than one marker (Figure 2C). When immune activation markers such as Lck or immune suppression such as p-STAT3 were assessed, these were also found to be significantly enriched in the perivascular region (Figures 2A, 2B, 3A, and 3B). The bioinformatics analysis extraction was sufficiently robust that it could discern that the Lck expression in the CD4+ T cells was localized to the perivascular region (Figures 3B and 3C). Notably if a cell population reaches at or below 1% within a given region, it is difficult to ascertain statistical differences. Cumulatively, these data indicate that most immune reactivity is localized to the perivascular regions of pilocytic astrocytomas.

This spatial analysis provides a strategy that successfully identifies regions that quantify cells based on phenotyping as a function of location relative to the tumor vasculature. The subsequent detection of nuclei, performed through a pre-trained deep-learning algorithm, allows for the classification of cells into specific image classes. Phenotypes of interest, including single positive, double positive, and functional marker positive cells are determined based on the mean pixel intensity of relevant biomarker signals. Pixel threshold values are selected to minimize false positive and false negative outcomes. Subsequent data analysis involved the generation of object-level information and the utilization of R programming to extract information such as phenotype counts, counts within specific regions, and the percentage composition of phenotypes.

## QUANTIFICATION AND STATISTICAL ANALYSIS

Data derived from the images are analyzed in R to create tables detailing the counts of each of the 45 phenotypes in each region of the tissue. Before any comparisons are made, regions with less than 100 cells are removed from the dataset. 214 perivascular regions and 351 tumor regions are left after filtering across pilocytic astrocytoma samples ( $n = 4$ ). To compare immune cell counts, relative immune cell composition, and immune cell phenotype counts, a Mann-Whitney test is performed between perivascular and tumor regions, and  $p$ -values are calculated. The same test was done to compare positive and negative phenotypes within each region.

## LIMITATIONS

CD31, an endothelial cell marker, may not show a homogeneous consistent vessel border because it depends on the cell orientation. In addition, Pixels representing a long vessel are sometimes smaller

in count than the number of pixels representing a DAPI-positive cell. This may not appear to be the case in visualized images depending on the thresholding. This made it difficult to apply the deep-learning algorithm to identify these vessels. Ultimately, other vessel markers may be more useful for training on vessel structures.

Rather than looking at positive or negative counts of a phenotype, pixel intensities could be used instead. This makes it possible to create a gradient representation of a marker to create a line graph showing the pixel intensity of a marker as a function of distance. This removes subjective decisions in deciding what is a “perivascular area” or “tumor area” or what threshold defines a particular cell lineage.

Non-specific binding for some targets such as PD-1 and PD-L1 will require a higher threshold. Adding another algorithm to remove these regions may increase accuracy.

## TROUBLESHOOTING

### Problem 1

Based on the processing power of the computer available and the number of images within a project, the accuracy of your training may vary (cf. Part 5).

#### Potential solutions

- The more images that are loaded into the slide tray and the App is trained on, the better the outcome.
- If the computer capacity is limited, load different amounts of images onto the slide tray. Train on one set of images for a portion of time and then train on the other sets of images for equal amounts of time.

### Problem 2

Sometimes the ‘loss function’ won’t converge based on the nature of the parameters (cf. Part 1).

#### Potential solution

- Lower the learning rate and increase the mini-batch size. However, training time will be longer.

### Problem 3

Different staining techniques and different tissue types can generate different signal intensities for each of the markers (cf. Part 5).

#### Potential solution

- For each tissue type, visually select the best threshold range appropriate to the marker’s architecture.
  - To determine the best low and high intensity, start with a wider range interval and narrow it down until you have the best visualization of your tissue architecture (e.g., our ultimate range for CD31 was selected as 6000–30000, but we first started with the empirically determined range of 2000–50000 and narrowed it down accordingly).

### Problem 4

The DAPI channel signal may vary based on cellular density (cf. Part 3).

### Potential solution

- The input channel for DAPI can be adjusted based on the strength and range of your DAPI signal for the best detection and segmentation of the cells.

### RESOURCE AVAILABILITY

#### Lead contact

amy.heimberger@northwestern.edu.

#### Technical contact

hindaj.najem@northwestern.edu.

#### Materials availability

This study did not generate new unique reagents.

#### Data and code availability

The published article includes all datasets/code generated or analyzed during this study. Original data have been deposited to GitHub data: <https://doi.org/10.5281/zenodo.10845756>

### SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.xpro.2024.103079>.

### ACKNOWLEDGMENTS

This study was funded by the NIH grants CA120813, NS120547, P30 CA016672, R50 CA243707, and P50CA221747. The Lunaphore COMET multiplex immunofluorescence system was enabled by a gracious gift from the Stephen M. Coffman Trust to the Northwestern Medicine Malnati Brain Tumor Institute.

### AUTHOR CONTRIBUTIONS

H.N. performed the multiplex analysis and H.N. and S.P. devised the spatial bioinformatic analysis. A.B.H. oversaw all aspects of the protocol and provided financial support. All authors contributed to writing the protocol.

### DECLARATION OF INTERESTS

A.B.H. serves on the advisory board of Caris Life Sciences.

### REFERENCES

1. Najem, H., Ott, M., Kassab, C., Rao, A., Rao, G., Marisetty, A., Sonabend, A.M., Horbinski, C., Verhaak, R., Shankar, A., et al. (2022). Central nervous system immune interactome is a function of cancer lineage, tumor microenvironment, and STAT3 expression. *JCI Insight* 7, e157612. <https://doi.org/10.1172/jci.insight.157612>.
2. Gabrusiewicz, K., Rodriguez, B., Wei, J., Hashimoto, Y., Healy, L.M., Maiti, S.N., Thomas, G., Zhou, S., Wang, Q., Elakkad, A., et al. (2016). Glioblastoma-infiltrated innate immune cells resemble M0 macrophage phenotype. *JCI Insight* 1, e85841. <https://doi.org/10.1172/jci.insight.85841>.
3. Weathers, S.P., Penas-Prado, M., Pei, B.L., Ling, X., Kassab, C., Banerjee, P., Bdiwi, M., Shaim, H., Alsuliman, A., Shanley, M., et al. (2020). Glioblastoma-mediated Immune Dysfunction Limits CMV-specific T Cells and Therapeutic Responses: Results from a Phase I/II Trial. *Clin. Cancer Res.* 26, 3565–3577. <https://doi.org/10.1158/1078-0432.Ccr-20-0176>.
4. Sonabend, A.M., Gould, A., Amidei, C., Ward, R., Schmidt, K.A., Zhang, D.Y., Gomez, C., Bebawy, J.F., Liu, B.P., Bouchoux, G., et al. (2023). Repeated blood-brain barrier opening with an implantable ultrasound device for delivery of albumin-bound paclitaxel in patients with recurrent glioblastoma: a phase 1 trial. *Lancet Oncol.* 24, 509–522. [https://doi.org/10.1016/s1470-2045\(23\)00112-2](https://doi.org/10.1016/s1470-2045(23)00112-2).
5. Sabbagh, A., Beccaria, K., Ling, X., Marisetty, A., Ott, M., Caruso, H., Barton, E., Kong, L.Y., Fang, D., Latha, K., et al. (2021). Opening of the Blood-Brain Barrier Using Low-Intensity Pulsed Ultrasound Enhances Responses to Immunotherapy in Preclinical Glioma Models. *Clin. Cancer Res.* 27, 4325–4337. <https://doi.org/10.1158/1078-0432.Ccr-20-3760>.
6. Vogelbaum, M.A., Li, G., Heimberger, A.B., Lang, F.F., Fueyo, J., Gomez-Manzano, C., and Sanai, N. (2022). A Window of Opportunity to Overcome Therapeutic Failure in Neuro-Oncology. *Am. Soc. Clin. Oncol. Educ. Book.* 42, 1–8. [https://doi.org/10.1200/edbk\\_349175](https://doi.org/10.1200/edbk_349175).
7. de Groot, J., Penas-Prado, M., Alfaro-Munoz, K., Hunter, K., Pei, B.L., O'Brien, B., Weathers, S.P., Loghin, M., Kamiya Matsouka, C., Yung, W.K.A., et al. (2020). Window-of-opportunity clinical trial of pembrolizumab in patients with recurrent glioblastoma reveals predominance of immune-suppressive macrophages. *Neuro Oncol.* 22, 539–549. <https://doi.org/10.1093/neuonc/noz185>.

8. Freeland, J., Bueren, E.v., Muralitharan, S., Meng, X.-Y., Harvey, M., Lienhart, W., King, C., and Milliman, K. (2013). Water-soluble organic solvent improves paraffin displacement in all-in-one epitope retrieval buffers. *J. Histotechnol.* 36, 51–58. <https://doi.org/10.1179/2046023613Y.0000000023>.
9. Freeland, J.H., and Harvey, M.A. (2014). Dewaxing buffer containing a water-soluble organic solvent and methods of use thereof. US Patent. US8753836B2.
10. Freeland, J.H., and Harvey, M.A. (2015). Dewaxing buffer containing a water-soluble organic solvent and methods of use thereof. Patent, People's Republic of China. CN104334717A.
11. Freeland, J.H., and Harvey, M.A. (2018). Dewaxing buffer containing a water-soluble organic solvent and methods of use thereof. European Patent. EP2807249B1.
12. Rivest, F., Eroglu, D., Pelz, B., Kowal, J., Kehren, A., Navikas, V., Procopio, M.G., Bordignon, P., Pérès, E., Ammann, M., et al. (2023). Fully automated sequential immunofluorescence (seqIF) for hyperplex spatial proteomics. *Sci. Rep.* 13, 16994. <https://doi.org/10.1038/s41598-023-43435-w>.