

## Quantifying spatial position in a branched structure in immunostained mouse tissue sections

Heilmann, Silja; Semb, Henrik ; Nyeng, Pia

*Published in:*  
STAR protocols

*DOI:*  
[10.1016/j.xpro.2021.100806](https://doi.org/10.1016/j.xpro.2021.100806)

*Publication date:*  
2021

*Document Version*  
Publisher's PDF, also known as Version of record

*Citation for published version (APA):*  
Heilmann, S., Semb, H., & Nyeng, P. (2021). Quantifying spatial position in a branched structure in immunostained mouse tissue sections. *STAR protocols*, 2(4), Article 100806.  
<https://doi.org/10.1016/j.xpro.2021.100806>

### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

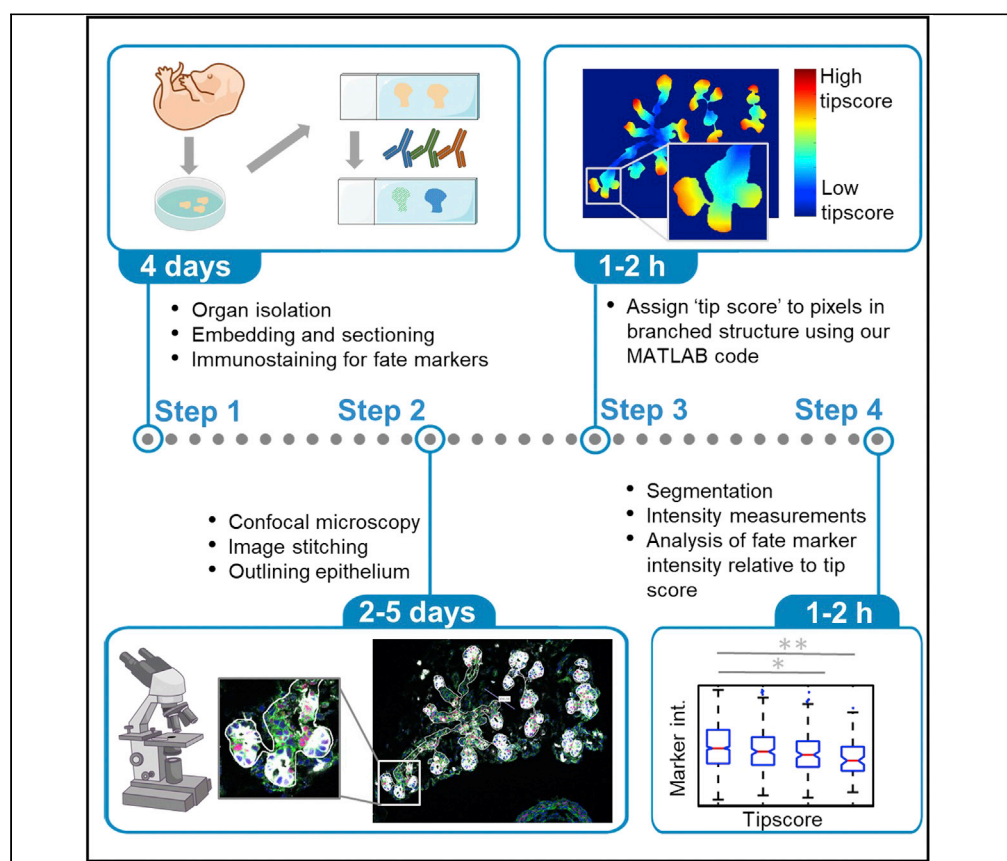
- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain.
- You may freely distribute the URL identifying the publication in the public portal.

### Take down policy

If you believe that this document breaches copyright please contact [rucforsk@kb.dk](mailto:rucforsk@kb.dk) providing details, and we will remove access to the work immediately and investigate your claim.

## Protocol

# Quantifying spatial position in a branched structure in immunostained mouse tissue sections



We have developed a protocol to quantify the position of a cell in a branched structure based on microscopy images of two-dimensional tissue sections. Biological branched structures include organs such as the lungs, kidneys, and pancreas. In these organs, cell fate has been correlated with position, based on a qualitative estimate. However, a quantitative means of evaluating the cell position has been lacking. With this protocol, the correlation between cell fate and cell position was measured in mouse embryonic pancreas.

Silja Heilmann,  
Henrik Semb, Pia  
Nyeng

pnnyeng@ruc.dk

### Highlights

A metric to quantify the position of a cell in a branched tissue structure

Low score/high score indicates proximity to the center/periphery, respectively

Score enables quantitative correlative studies of cell fate and cell position

Comprehensive pipeline from tissue processing instructions to image analysis code

Heilmann et al., STAR  
Protocols 2, 100806  
December 17, 2021 © 2021  
The Author(s).  
<https://doi.org/10.1016/j.xpro.2021.100806>





## Protocol

# Quantifying spatial position in a branched structure in immunostained mouse tissue sections

Silja Heilmann,<sup>1</sup> Henrik Semb,<sup>1,3</sup> and Pia Nyeng<sup>2,4,5,\*</sup>

<sup>1</sup>Novo Nordisk Foundation Center for Stem Cell Biology (Danstem), University of Copenhagen, Blegdamsvej 3B, 2200 Copenhagen N, Denmark

<sup>2</sup>Department of Science and Environment, Roskilde University, Universitetsvej 1, Postbox 260, 4000 Roskilde, Denmark

<sup>3</sup>Institute of Translational Stem Cell Research, Helmholtz Zentrum München, Deutsches Forschungszentrum für Gesundheit und Umwelt (GmbH), Ingolstädter Landstraße 1, 85764 Neuherberg, Germany

<sup>4</sup>Technical contact

<sup>5</sup>Lead contact

\*Correspondence: [pneng@ruc.dk](mailto:pneng@ruc.dk)

<https://doi.org/10.1016/j.xpro.2021.100806>

## SUMMARY

We have developed a protocol to quantify the position of a cell in a branched structure based on two-dimensional microscopy images of tissue sections. Biological branched structures include organs such as the lungs, kidneys, and pancreas. In these organs, cell fate has been correlated with position, based on a qualitative estimate. However, a quantitative means of evaluating the cell position has been lacking. With this protocol, the correlation between cell fate and cell position was measured in mouse embryonic pancreas. For complete details on the use and execution of this protocol, please refer to Nyeng et al. (2019).

## BEFORE YOU BEGIN

This protocol outlines a comprehensive pipeline for generating and staining biological samples from mice and quantitatively analyzing the correlation between cell position and cell fate in branched tissues. For this purpose, we have developed a metric for how distant each pixel inside the tissue is from the center of a branched structure (Nyeng et al., 2019). A low score indicates proximity to the center (in the pancreas called “trunk”), while a high score indicates a location close to the periphery (in the pancreas called “tip”). We named this metric the “tip score”.

If biological samples are already available in the form of tissue sections, the protocol can be followed from preparatory step 6 and protocol step 3. If stained tissue sections are already available, the protocol can be followed from preparatory step 8 and protocol step 9. For image analysis of existing images, the protocol can be followed from preparatory step 8 and protocol step 11.

**Note:** The computational part of this protocol relies on the proprietary software MATLAB

**Note:** While the protocol was optimized for embryonic mouse pancreas analysis and may require modification for use on other branched inner organs (lungs, liver, thyroid, etc) and will require optimization for use on other branched structures in general, the basic idea of the image analysis method should be widely applicable to any branched structure.

## Prepare for collecting tissue sections from organs

⌚ Timing: 2 days



1. Prepare 4% formaldehyde for tissue fixation from paraformaldehyde (PFA) while working in a chemical safety cabinet
  - a. Add 8 g of PFA to 192 mL sterile PBS in a 250 mL beaker
  - b. Heat at 70°C–80°C for ca. 1 h until all the powder has dissolved. Monitor temperature closely, and never bring the solution above 80°C
  - c. Cool down to 20°C–22°C
  - d. Aliquot into 15 mL tubes with 10 mL/tube and store for up to one year at -20°C, unless used immediately

⚠ **CRITICAL:** Paraformaldehyde is a toxic chemical which targets the respiratory system and should be used according to the safety instructions. Use gloves and work in a chemical safety cabinet or use eye/face shield and respirator cartridge type N100 (US), type P1 (EN143) respirator filter, type P3 (EN 143) respirator cartridges.

**Alternatives:** Commercially available ampules of premade aqueous solution of 4% formaldehyde without additives. Commercially available concentrated formaldehyde aqueous solutions not in ampules should be avoided, as they often include 10% methanol or butanol as stabilizing agents (Helander, 2000; Fox et al., 1985). Our staining protocol has been optimized for tissue fixed in pure 4% formaldehyde, however other fixatives such as glutaraldehyde and methanol may also be used according to your own protocol.

2. Prepare 30% sucrose solution in PBS
  - a. Add 300 g of sucrose to a 1 L glass bottle and fill with sterile PBS up to 1L
  - b. Heat at 50°C for 30 min with constant stirring
  - c. Store at 4°C for 4–8 weeks

### Setup mouse mating for embryonic tissue

⌚ **Timing:** 2–3 days (but needs to be set up several days in advance depending on embryonic stage needed)

If in need of embryonic tissue for analysis, setup mouse breeding as follows:

3. Setup mating of mice. These can be wildtype (C57bl/6 was used for exemplary data) or genetically modified
  - a. Move one male to each of 3–4 clean breeding cages (if time permits this can be done the previous evening)
  - b. Add two females to each breeding cage in the afternoon
4. Check for vaginal plug the next morning
  - a. Separate plugged females to a clean cage, and record number of plugged females
  - b. Separate males and unplugged females
  - c. Day of vaginal plug is recorded as embryonic day 0.5 (E0.5)
  - d. Carefully record mouse ID and genotype of both male and female along with day of setup and day of plug
5. Wait the required number of days according to which embryonic stage you aim to analyze. I.e., for E14.5 tissue you need to set up breeding 16 days prior to the day you plan to start the experiment

**Note:** Make sure you have the necessary ethical and breeding permits for laboratory animal use from your local authorities

### Prepare reagents for immunostaining of tissue sections

⌚ **Timing:** 2 h

6. Optional: Prepare Citrate buffer pH6 for antigen retrieval
  - a. Add 2.94 g Tri-sodium Citrate (dihydrate) to 900 mL dH<sub>2</sub>O in a 2 L glass bottle
  - b. Stir until dissolved
  - c. Set pH to 6.0 by adding citric acid powder while stirring
  - d. Add dH<sub>2</sub>O to a total of 1000 mL
  - e. Store at 20°C–22°C for up to 3 months

**Note:** Antigen retrieval is optional, as it depends on the antibodies used. For our test dataset we used a mild citrate buffer-based antigen retrieval method to optimize the SOX9 staining.

7. Prepare blocking buffer
  - a. Prepare 1 L of 0.1 M TRIS-HCL pH 7.5
    - i. Add 15.76 g TRIZMA-HCL to 800 mL dH<sub>2</sub>O
    - ii. Measure pH using a newly calibrated pH meter
    - iii. Adjust pH to 7.5 using HCL
    - iv. Add dH<sub>2</sub>O to a make up a total of 1000 mL
    - v. The Tris-HCL solution can be stored at 4°C for up to 1 year
  - b. To the 0.1 M TRIS-HCL solution, add NaCl to 0.15 M
  - c. Heat solution to 55°C with constant stirring
  - d. Add Blocking reagent to 0.5% (Akoya Biosciences) in small increments to buffer while stirring until the powder is completely dissolved (30–60 min). The solution will appear milky.
  - e. Cool down to 20°C–22°C
  - f. Aliquot into 50 mL tubes and freeze at -20°C, unless used immediately. The frozen stock can be stored for up to 1 year. A working solution can be stored at 4°C for 2–4 weeks.

Blocking Buffer		
Reagent	Final concentration	Amount
0.1 M TRIS-HCL pH 7.5	0.1 M	1000 mL
NaCl	0.15 M	8.77 g
Blocking reagent	0.5%	5.00 g

**Alternatives:** Other blocking buffers such as donkey serum may also be used according to your own protocol.

### Download, install, and run MATLAB and demo scripts

⌚ Timing: 3 h or up to 1–2 days depending on whether you have previously worked with MATLAB

⌚ Timing: approx. 30 min to 1 h depending on your computer for step 11 (optional)

We provide 4 custom written MATLAB functions:

```
1) tipScoreIm()
2) segmentDAPIimage()
3) returnTableWithCellInt()
4) returnTableWithCellPairInt()
```

These can be used for assigning a tip score to all regions of a branched organ (1), segmenting cell nuclei (2) and relating this tip score to the fluorescence intensity of cellular markers in either the nucleus/cytoplasm (3), or cell membrane (4). We also provide two demo scripts:

Demo 1) 'demoScript\_OneImage\_ManualInput.m'

Demo 2) 'demoScript\_Batch\_NoInput.m'

These scripts demonstrate how our custom functions can be called/used when analyzing sample images with hand drawn outlines, from the point of loading in images all the way to final statistical comparison of groups of cells with different tip scores.

8. Install MATLAB version 2018a or later on your computer.
  - a. Press here: [MATLAB](#) to go to the MathWorks homepage.

**Note:** MATLAB is a commercial product, which requires a license. Also, each toolbox requires its own license. Many universities have MATLAB licenses. For users without access to a MATLAB license, MATLAB offers a 30-day free trial, which will be sufficient time to run this analysis.

- b. Make sure to check the boxes for 'Signal Processing Toolbox', 'Image Processing Toolbox', and 'Statistics and Machine Learning Toolbox' during installation. Installation takes approx. one hour or less, depending on your internet connection.
9. Go to GitHub and Mendeley Data and download the functions and demo scripts or download the supplementary files [Data S1](#) and [S2](#).
  - a. Press here: [GitHub](#) to go to the GitHub homepage.
  - b. Search for 'TipScore' in the upper left search box and go to [SiljaHeilmann/TipScore](#)
  - c. Download all files by pressing the green 'code' button and select 'download zip'. The folder is called 'TipScore-main.zip'. Unzip and move the TipScore-main folder to your location of choice.
  - d. Download the folders 'raw\_images\_tif\_files' and 'outlines\_mat\_files' (approx. 574MB), containing a small test dataset (containing 7 images) from Mendeley Data: <https://doi.org/10.17632/nr9cyk265.1> and place them in the your TipScore-main folder.
10. Test that MATLAB works by running the script 'demoScript\_OneImage\_ManualInput.m'
  - a. First open MATLAB
  - b. Check that 'Current Folder' (see left side) is your 'TipScore-main' folder and that it contains the folders 'raw\_images\_tif\_files' and 'outlines\_mat\_files' and all .m files downloaded from GitHub (segmentDAPIimage.m, tipScoreIm.m, returnTableWithCellInt.m, returnTableWithCellPairInt.m)
  - c. Open 'demoScript\_OneImage\_ManualInput.m' in MATLAB
  - d. Run script either by typing 'demoScript\_OneImage\_ManualInput' in the MATLAB command window and pressing enter, or by putting the cursor in the editor window and pressing the green Run button in the Editor tab. (You can also run sections of the code one at a time). Note that this code runs the analysis on image number 6 in the test data set. If you wish to see analysis of another test image in the set edit 'ff = 6;' in line 11 of the code to a different number, (1, 2, 3, 4, 5 or 7).
  - e. A figure with the title 'Red: Ecad. Blue: DAPI. Green: P120CTN. White: CPA...' will open. A blue line tool, which measures length in pixels, is visible on the image. You can drag the ends of it and use it to measure the average length of the tip structures. If you wish to zoom in, then hover the mouse at the upper right corner of the image and controls for zoom and hand drag appear.
  - f. At the MATLAB command window, you are prompted: 'What is the average diameter/length of tip structures in pixels?... :'  
Input your response and press enter. For this test image an appropriate answer is approx. 300. (The final tip score of a location will depend on this input, a larger number will mean that a larger portion of the structure will get a higher tip score).

- g. A figure with the title 'Red: Ecad. Blue: DAPI. Green: P120CTN. White: CPA...' will open. A blue line tool, which measures length in pixels, is visible on the image. You can drag the ends of it and use it to measure the average length of the tip structures. At the MATLAB command window, you are prompted: 'What is the average diameter of nuclei in pixels? ...'. Input your response and press enter. For this test image set approx. 20 is an appropriate answer. (The final nuclei segmentation result will depend on this input; a larger number will mean a tendency to under segment and a smaller number will cause over segmentation).

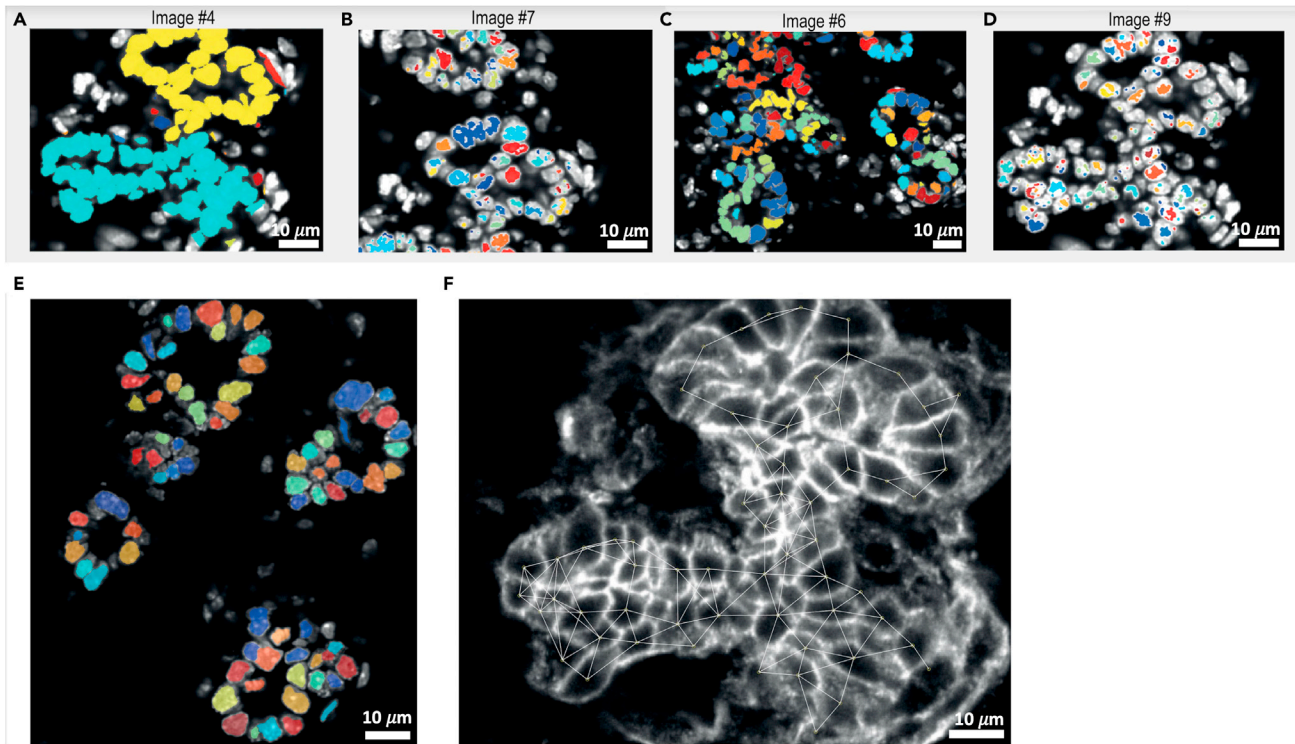
**Note:** Image segmentation is the process of partitioning an image into multiple segments (sets of pixels, also known as image objects). Image segmentation is typically used to locate objects and boundaries. Here it is used to detect cell nuclei stained with DAPI.

- h. You now should see 10 thresholded images. At the MATLAB command window, you are prompted: 'What thresholded image includes all nuclei and not more? ...:'. For this test image approx. '4' is an appropriate answer. See [Figure 1A](#). Note: It is easier to judge between the images if you zoom in a bit. (The final nuclei segmentation result will depend on this input, a larger number will mean a tendency to lose cells completely, a smaller number will cause cells to clump).
- i. You now should see 10 thresholded images. At the MATLAB command window, you are prompted: 'What thresholded image merges no nuclei? ... :'. For this image approx. '7' is an appropriate answer. See [Figure 1B](#). (The final nuclei segmentation result will depend on this input, a larger number will mean a tendency to lose cells completely, a smaller number could cause cells to clump).
- j. You now should see 10 thresholded images. At the MATLAB command window, you are prompted: 'What thresholded image looks best? I.e. separates most nuclei while losing the least?...'. For this image approx. '6' is an appropriate answer. See [Figure 1C](#). (The final nuclei segmentation result will depend on this input, a larger number will mean a tendency to lose cells completely, a smaller number could cause cells to clump).
- k. You now should see 10 thresholded images. At the MATLAB command window, you are prompted: 'What thresholded image merges no nuclei? ...:'. For this image approx. '9' is an appropriate answer. See [Figure 1D](#). (The final nuclei segmentation result, see [Figure 1E](#) will depend on this input, a larger number will mean a tendency to lose cells completely, a smaller number could cause cells to clump). A few nuclei merges and over segmentations are to be expected.
- l. From line 99 on onwards the script `demoScript_OneImage_ManualInput.m` uses the nucleus positions found, the tip score image and the images of nuclear/membrane marker to extract intensities from cells (nuclear) and between cell-pairs (membrane) via the `returnTableWithCellInt()` function and the `returnTableWithCellPairInt()` function, (see step 20 and step 21 below) and return them in table form. The output figure from the function `returnTableWithCellPairInt()` is shown in [Figure 1F](#). The returned tables contains all extracted information about each cell/cell-pair ('Cell/Cell-pair ID number', 'X coordinate', 'Y coordinate', 'SOX9 intensity/p120 intensity', 'tip score value', 'normalized SOX9 intensity/normalized p120 intensity', 'Assigned tip score group'). From line 165 and on boxplot figures are generated and non-parametric statistical tests performed to compare data points belonging to different tip score groups, see section '[quantification and statistical analysis](#)' for more details.

11. Optional: Run example batch script `demoScript_Batch_NoInput`.

**Note:** `demoScript_Batch_NoInput.m` does the same as `demoScript_OneImage_ManualInput.m` except it performs the operations/functions of all images of a batch of 7 (stored in 'raw\_images\_tif\_files' folder) and it does not require manual input (it already has





**Figure 1. Examples of output images**

(A) Example image number/answer for the 3rd question prompted during the run of `demoScript_OneImage_ManualInput.m`. Segmentation should cover all nuclei (but not more than that) and nuclei should not have holes in them.

(B) Example image number/answer for the 4th question prompted during the run of `demoScript_OneImage_ManualInput.m`. Segmentation should cover as much of nuclei as possible while not merging them. (Very few merges are ok).

(C) Example image number/answer for the 5th question prompted during the run of `demoScript_OneImage_ManualInput.m`. Segmentation should separate most nuclei while losing the least. Note: Smoother boundaries are better than nuclei separation.

(D) Example image number/answer for the 6th question prompted during the run of `demoScript_OneImage_ManualInput.m`. Segmentation should optimally merge no nuclei. (Very few merges are ok).

(E) Example of the final output of the `segmentDAPIImage()` image function. (Note: the output is a binary image, but different regions/connected components are here shown with different colors to emphasize the segmentation).

(F) Output figure during a call of the function `returnTableWithCellPairInt()`, where `plot_yes_no=1`. White lines show the cell neighbor pairs line scans performed in the image. Inspecting this plot can help you find the appropriate value of `nucleiD`, such that many linescans are performed but not too many that connect cells which are not actual neighbors. Note: you do not need to use the same value for `nucleiD` as you did for `segmentDAPIImage()`. The scale bars are 10  $\mu\text{m}$ .

the appropriate manual answers stored in the script, see line 21–29). This script can be used for inspiration when you want to write your own batch script.

- a. Open `demoScript_batch_NoInput.m` in MATLAB
- b. Run script either by typing '`demoScript_batch_NoInput`' in the MATLAB command window and pressing enter, or by putting the cursor in the editor window and pressing the green 'Run' button in the Editor tab. You should see folders appear in your current directory with output files for each of the 7 images in `raw_images_tif_files`. Finally, after all images have been tipscored and nuclei located (this takes a while) boxplots of SOX9 and P120CTN intensity versus tipscore for individual images and all images pooled, will be generated.

**Alternatives:** Other programming languages (such as Python or Julia) can be used, if the end-user adapts the script accordingly.



## KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
<b>Antibodies</b>		
Goat polyclonal anti-CPA1 (dilution 1:100)	R&D Systems	Cat #: AF2765 RRID: AB_2085841
Mouse polyclonal anti-p120ctn (pp120) (dilution 1:400)	BD Transduction Labs	Cat #: 610134 RRID: AB_397537
Rabbit polyclonal anti-Sox9 (dilution 1:1000)	Millipore	Cat #: AB5535 RRID: AB_2239761
Secondary antibody Rhodamine Red™-X (RRX) AffiniPure Goat Anti-Mouse IgG, Fcγ subclass 1 specific (min X Hu, Bov, Rb Sr Prot) (dilution 1:1000)	Jackson ImmunoResearch	cat# 115-295-205 RRID: AB_2338769
Secondary antibody Cy™5 AffiniPure Donkey Anti-Goat IgG (H+L) (dilution 1:1000)	Jackson ImmunoResearch	Cat #: 705-175-147 RRID: AB_2340415
Secondary antibody Alexa Fluor® 488 AffiniPure Fab Fragment Donkey Anti-Rabbit IgG (H+L) (dilution 1:1000)	Jackson ImmunoResearch	Cat #: 711-547-003 RRID: AB_2340620
<b>Chemicals, peptides, and recombinant proteins</b>		
PBS tablets pH 7.4 (for 500 ML)	Fisher Scientific	Cat #: 11510546 RRID: N/A
Sterile PBS w Mg/Ca	Gibco	Cat#: 14040083 RRID: N/A
Paraformaldehyde powder	VWR	Cat #: 28794.295 RRID: N/A
DAPI molecular grade	VWR	Cat #: APLIA4099.0010 RRID: N/A
NaCl	Merck	Cat #: 1.06404.0500 RRID: N/A
TRIZMA-HCl	Sigma-Aldrich	Cat #: T5941 RRID: N/A
Triton X-1000	Sigma-Aldrich	Cat #: X-100 RRID: N/A
Citric Acid NORMAPUR	Fisher Scientific	Cat #: 15672880 RRID: N/A
Tri-sodium citrate (dihydrate)	Merck	Cat #: 1064481000 CAS Number 6132-04-3
Sucrose (Saccharose NORMAPUR analytical reagent)	Fisher Scientific	Cat #: 10647421 RRID: N/A
<b>Critical commercial assays</b>		
Blocking Reagent	Akoya Biosciences	Cat #: SKU FP1012 RRID: N/A
<b>Deposited data</b>		
Original image data test set	This paper	Heilmann, Silja; Nyeng, Pia (2021), "Pancreas_sample_images", Mendeley Data, V1, <a href="https://doi.org/10.17632/nr9cyyk265.1">https://doi.org/10.17632/nr9cyyk265.1</a>
<b>Experimental models: organisms/strains</b>		
Mouse: C57BL/6BomTac	Taconic	Cat #: B6JBOM-F/M RRID: N/A
<b>Software and algorithms</b>		
Fiji 2.0/ImageJ version 1.51u or more recent	(Rueden, 2017) <a href="http://imagej.nih.gov/ij">http://imagej.nih.gov/ij</a>	RRID: SCR_003070
MATLAB version R2018a or later Make sure to include "Signal Processing Toolbox", "Image Processing Toolbox", and "Statistics and Machine Learning Toolbox" during installation.	MathWorks <a href="https://se.mathworks.com/campaigns/products/trials.html">https://se.mathworks.com/campaigns/products/trials.html</a>	RRID: SCR_001622

(Continued on next page)

<b>Continued</b>		
REAGENT or RESOURCE	SOURCE	IDENTIFIER
ZEN black edition including stitching module	Zeiss	RRID: SCR_018163
Original custom MATLAB functions: <code>tipScoreIm()</code> , <code>segmentDAPIimage()</code> , <code>returnTableWithCellInt()</code> , <code>returnTableWithCellPairInt()</code>	This paper	<a href="https://github.com/SiljaHeilmann/TipScore">https://github.com/SiljaHeilmann/TipScore</a>
<b>Other</b>		
Dumont #5 Forceps in Inox-Biology	Fine Science Tools	Cat#: #5SF RRID: N/A
Stereoscope for embryo dissection	Nikon	Cat# SMZ1270 RRID: N/A
Leica Cryostat	Leica	Cat #:CM1950 RRID: SCR_018061
Tissue-Tek® O.C.T.™ Compound Cryo Embedding Medium	Sakura	Cat #:4583 RRID: N/A
Cryomold® Intermediate 15 × 15 × 5 mm	Sakura	Cat #:4566 RRID: N/A
Thermo Scientific™ SuperFrost Plus™ Adhesion Slides 75×25×1 mm	Fisher Scientific <a href="https://www.fishersci.dk/shop/products/superfrost-plus-adhesion-slides-9/10149870#?keyword=10149870">https://www.fishersci.dk/shop/products/superfrost-plus-adhesion-slides-9/10149870#?keyword=10149870</a>	Cat #:10149870 RRID: N/A
PAP pen (liquid blocker)	Merck	Cat #: Z377821-1EA
High precision cover glasses 24×60 mm #1.5 (170 μm ± 5 μm)	Marienfeld distributed by VWR	Cat #: MARI0107242 RRID: N/A
DAKO Mounting Medium	Agilent Technologies	Cat #:CS70330-2 RRID: N/A
Humidity chamber: Simport Scientific Staintray System with black lid (20 slides). Simport™ M920-2	Fisher Scientific	Cat #:11610581 RRID: N/A
Tissue-Tek® 24-Slide Holder with Detachable Handle	Sakura	Cat #:4457 RRID: N/A
Tissue-Tek® Staining Dish White	Sakura	Cat #:4465 RRID: N/A
Micropipettes	N/A	N/A
Zeiss LSM 780 inverted confocal scanning microscope	Zeiss	N/A
Objective: Plan Apochromat 10×0.45 M27 – WD 2mm	Zeiss	N/A
Objective: Plan-Apochromat 40×1.30 Oil DIC M27 – WD 0.21mm	Zeiss	N/A

## STEP-BY-STEP METHOD DETAILS

### Step 1–2: Prepare tissue sections from mouse organs

⌚ Timing: 2 days

This step prepares tissue sections for the immunostaining protocol. For our exemplary data we prepared E14.5 embryonic pancreas from the mouse, but the protocol can be used for other tissues and ages depending on the biological question addressed, as long as a clear branching structure is evident, and cells/nuclei can be segmented.

**Note:** For the purpose of minimizing biological variation caused by developmental differences, it is important that dissection is always carried out at the same time of day (i.e., at noon), and that each embryo is examined with regard to normal development and embryonic age (see step 1e). For younger embryos somites can be counted to more accurately stage embryos. Consult (Kaufman, 1992) for more information on staging.

**Note:** For the purpose of statistical analysis of biological variation between littermates, it is advisable to dissect and analyze at least 3–4 embryos per pregnant female. In our experience there is some variation in cell differentiation between litters (this can however be minimized by careful staging of embryos as described above). If analyzing the effect of a treatment or a genetic perturbation, it is therefore important to test whether a difference is due to litter variation or the perturbation.

1. Isolate your organ of interest from embryos
  - a. Euthanize the pregnant female by cervical dislocation
  - b. Dissect the uterus containing the embryos from the pregnant female
  - c. Move embryos to ice-cold sterile 1 × PBS in a Petri dish stored on wet ice
  - d. Dissect the embryos one by one from the uterus and move to a clean Petri dish containing ice-cold sterile 1 × PBS
  - e. Check the developmental age of the embryo by careful examination of the front and rear paw characteristics consulting ([Kaufman, 1992](#))

**Note:** Some variation of embryo development (–0.5 days) may occur within a litter, and we routinely exclude embryos in which the developmental stage does not match the gestational stage

- f. Under a stereoscope, dissect your organ of interest from the embryos using #1 Dumont forceps or similar
- g. Transfer the tissue to 4% formaldehyde in 1 × PBS in a 12 well plate and fix at 20–22°C for 1–24 h depending on the size of the tissue.

**Note:** Although adequate fixation is required for good cytological preservation, over-fixation can block or prevent antibodies from binding to the protein epitope ([Wang and Matise, 2013](#)). We routinely fix E14.5 embryonic pancreas for 16–18 h. The reaction of formaldehyde depends on the temperature, and penetration of tissue by formaldehyde is a function of the square root of the time of exposure ([Fox et al., 1985](#)). We recommend consulting ([Fox et al., 1985](#)) to determine the optimal fixation time.

- h. Cryoprotect the tissue by transferring to 30% sucrose in 1 × PBS solution at 4°C until the tissue falls to the bottom of the well (usually within 3–16 h depending on the size).
- i. Infiltrate tissue with OCT by incubating in 100% OCT for 1 h at 4°C
- j. Embed tissue into blocks as follows:
  - i. Label plastic mold
  - ii. Place the plastic mold onto a piece of dry ice, adding enough OCT to cover bottom
  - iii. Wait for the bottom to freeze
  - iv. Place the tissue in the middle of the block and add OCT to cover. Be careful not to make any bubbles in the OCT.
  - v. When all of the OCT is white, store in a sealed plastic bag at –80°C immediately.

▮▮ **Pause point:** If needed, frozen tissue blocks can be stored at –80°C indefinitely

2. Section tissue on a cryostat
  - a. Section tissue on a cryostat following manufacturer's instructions
  - b. Section at 6 μm and place 2–3 sections per slide depending on the size

**Note:** Sections should stay within the boundaries of the + signs on the slide and should not overlap.

- c. Mount tissue sections on labeled and numbered Superfrost PLUS glass slides
- d. Store at –80°C

▮▮ **Pause point:** if needed, frozen sections can be stored at –80°C for 2–3 years

**Table 1. Appropriate controls for antibody staining and imaging (Lee and Kitaoka, 2018)**

Control name	Tissue	Antibody treatment	Purpose	Specific handling during staining
Negative control	Same as sample	-No primary and secondary antibodies	To determine autofluorescence of tissue	Step 6: Leave section in blocking buffer Step 8: Leave section in last PBS wash and do not add DAPI
Secondary antibody only control	Same as sample	-No primary antibody -All secondary antibodies	To control for unspecific secondary antibody binding and crossreactivity	Step 6: Leave section in blocking buffer
Optional: "Leave one out" controls (only when multiplexing antibodies)	Same as sample	-All primary antibodies except one -All corresponding secondary antibodies	This control allows you to take into account how the other antibodies in your panel affect the left out antibody: You can for instance detect bleed-through into the channel were the left out antibody is acquired	Step 6: Omit one primary antibody from your antibody panel Step 8: Omit the corresponding secondary antibody

### Step 3–8: Immunostain tissue sections

⌚ Timing: 2 days

This step stains the tissue in preparation for imaging and image quantification. It has been optimized for embryonic mouse pancreatic tissue fixed in 4% paraformaldehyde and prepared as frozen sections as described in the previous step but can also be used for other embryonic organs.

**⚠ CRITICAL:** After step 4c the tissue should always be kept moist or submerged. Any drying during antibody incubation will increase the antibody concentration and drying out at all other steps will increase nonspecific antibody binding and therefore background fluorescence. See [problem 1: Excessive background fluorescence](#)

**Alternatives:** Paraffin sections can be used instead of frozen sections. In that case, the sections should be deparaffinized and rehydrated: Incubate for 15 min in Xylene to deparaffinize and rehydrate in a descending alcohol series: 100% ethanol for 2 times 3 min, 95% ethanol for 3 min, 70% ethanol for 3 min, 50% ethanol for 3 min. After rehydration the slides are washed in running tap water and staining can proceed from step 4c below.

3. Select a number of slides containing tissue that adequately represents the whole 3-dimensional organ.

**Note:** Exclude any sections not showing the branching structure (typically, these are at the extreme ends of the tissue). For a single E14.5 pancreas we typically get 200 6  $\mu$ m sections, and stain every 30th section (excluding the extreme ends), giving a total of 4–6 sections/pancreas. Include sufficient individual biological replicates to be able to perform statistical analysis (3–5 biological replicates of each experimental condition) and have enough samples for controls (see [Table 1](#)).

4. Prepare slides for antibody incubation
  - a. Align 3–20 slides in a slide holder for staining. (See [Figure 2](#)). Always include at least one slide as a negative control and one as secondary antibody only control (see [Table 1](#) below for more information on controls).
  - b. Dry sections in an oven at 37°C for 20 min
  - c. Wash 3 times in 1  $\times$  PBS for 5 min each



**Figure 2. Methods and equipment**

Slide holder for washing steps, PAP pen and humidified chamber with numbered slides inside.

- d. Optional: Antigen retrieval
  - i. Remove 1× PBS and leave in Citric acid antigen retrieval buffer for 1 h at 37°C
  - ii. Wash 3 times in 1× PBS for 5 min each

**Note:** Antigen retrieval is necessary for unmasking of some but not all antigens prior to primary antibody incubation. Detection of some antigens may even suffer from antigen retrieval. For our antibody panel, antigen retrieval by microwave treatment in citrate buffer pH6 (Shi et al., 1993) is necessary for detection of SOX9, while P120CTN detection suffers. We therefore used a milder treatment with incubation in citrate buffer pH 6 without boiling in the optional steps above.

- e. Working one slide at a time to reduce drying:
  - i. Place slides flat in a humidified chamber
  - ii. Outline tissue section with a hydrophobic PAP pen (See Figure 2)
  - iii. Add enough blocking buffer to each slide to cover tissue
- f. Incubate slides in blocking buffer for 60–120 min at 20°C–22°C

5. In the meantime, prepare antibody dilutions (see Table 2 for an example):
  - a. Dilute primary antibodies for one or two antigens expressed in your cells of interest (here: SOX9 and Carboxypeptidase 1 (CPA1)), and for a marker visualizing the general shape of the organ (here: P120CTN) in blocking buffer.

**△ CRITICAL:** Primary antibody concentration has to be carefully titrated for specific tissues to avoid non-specific binding. See problem 1: Excessive background fluorescence

**Note:** Prepare enough primary antibody dilution to stain all your samples, allowing 100 µl per sample. Note that extra diluted antibody should be made to account for pipetting error (at least 10% extra volume).

6. Primary antibody incubation

**Table 2. Antibody panel used to generate the test image dataset**

Primary antibody solution	Final dilution	Amount for 9 samples (1 mL)
SOX9 antibody produced in rabbit	1:1000	1.0 $\mu$ L
CPA1 antibody produced in goat	1:100	10 $\mu$ L
P120CTN antibody produced in mouse	1:400	2.5 $\mu$ L
Blocking buffer	n/a	986.5 $\mu$ L
Total	n/a	1000 $\mu$ L

- a. Working one slide at a time to reduce drying:
  - i. Remove blocking buffer with a micropipette
  - ii. Add enough antibody solution (typically 100–200  $\mu$ L) to each slide to cover tissue completely.
  - iii. Leave at least two sections with only the blocking buffer on, to serve as negative and secondary antibody only controls (see [Table 1](#)).
  - iv. Leave for 16–20 h at 20°C–22°C in the humidified chamber (See [Figure 2](#))

7. Next morning: Prepare secondary antibody solution (See [Table 3](#) for an example)

- a. Use secondary antibodies against the species in which the primary antibodies are produced.

**△ CRITICAL:** Secondary antibodies produced in the same species (i.e., Donkey) are preferable to avoid cross-reaction. For our test data, we had to use a secondary antibody produced in goat, which would be bound by the anti-goat antibody used to detect CPA1, if not applied separately and in the correct order as outlined below. See also [Table 3](#).

**△ CRITICAL:** The secondary antibodies must be conjugated to stable fluorescent dyes matching the configuration of your fluorescent microscope. A good combination with minimal spectral overlap is Dylight-405/DAPI, Alexa Fluor-488, Rhodamine-RedX, Alexa-Fluor 647. See [problem 2: Unexpected staining patterns](#)

- b. Make secondary antibody solution 1 and 2 by diluting secondary antibodies in blocking buffer at a dilution of 1:1000 as outlined in [Table 3](#). Leave the tubes with diluted antibodies on wet ice and protected from light.

**Note:** Secondary antibodies dilutions should be optimized for specific tissues to avoid non-specific binding. See [problem 1: Excessive background fluorescence](#)

**Note:** Prepare enough secondary antibody dilution to stain all your samples and your secondary antibody only control, allowing 100  $\mu$ L per sample. Note that extra diluted antibody should be made to account for pipetting error (at least 10% extra volume).

8. Secondary antibody incubation

- a. Remove primary antibody solution with a micropipette
- b. Wash 3 times in 1  $\times$  PBS for 5 min each
- c. Working one slide at a time to reduce drying:
  - i. Add enough **secondary antibody solution 1** (typically 100–200  $\mu$ L) to each slide to cover tissue.
  - ii. Add secondary antibody to the secondary antibody only control slides. Leave the negative control slide in 1  $\times$  PBS.
- d. Incubate for 60–120 min.
- e. Wash 3 times in 1  $\times$  PBS for 5 min each
- f. Working one slide at a time to reduce drying:
  - i. Add enough secondary antibody solution 2 (typically 100–200  $\mu$ L) to each slide to cover tissue.

**Table 3. Secondary antibodies used to generate the test image dataset**

Secondary antibody solution 1	Final dilution	Amount for 9 samples (1 mL)
Anti-Rabbit antibody produced in donkey and conjugated to Alexa-Fluor 488	1:1000	1.0 $\mu$ L
Anti-goat antibody produced in donkey and conjugated to Cy-5	1:1000	1.0 $\mu$ L
Blocking buffer	n/a	998 $\mu$ L
Total	n/a	1000 $\mu$ L
Secondary antibody solution 2	Final dilution	Amount for 9 samples (1 mL)
Anti-Mouse IgG, Fc $\gamma$ subclass 1 specific, polyclonal antibody produced in goat and conjugated to Rhodamine Red-X	1:1000	1.0 $\mu$ L
Blocking buffer	n/a	999 $\mu$ L
Total	n/a	1000 $\mu$ L

- ii. Add secondary antibody solution 2 to the secondary antibody only control slides. Leave the negative control slide in 1  $\times$  PBS.
- g. Incubate for 60–120 min.
- h. Wash 3 times in 1  $\times$  PBS for 5 min each
- i. For blue staining of nuclei, add DAPI to the last wash at a dilution of 1:1000
- j. Mount with DAKO mounting medium or similar and #1.5 cover glass preferably with a thickness of 0.17–0.18 mm (usually marked #1.5H or “high performance”)
- k. Leave flat to dry for 10 min protected from light

**Pause point:** if needed, stained sections can be stored dark at 4°C for several weeks.

**CRITICAL:** Variation in staining intensity can occur from experiment to experiment, so it is critical to include all samples in the same experiment, if comparison of staining intensity is planned.

**Alternatives:** Different combinations of primary and secondary antibodies can be used as needed.

## Step 9–10: Confocal microscopy of stained tissue sections

**Timing:** 1–3 days depending on number of samples

Microscopy of stained tissue sections creates the images needed for subsequent image analysis steps.

Confocal imaging was performed on a Zeiss LSM780 inverted confocal microscope equipped with a 32PMT GaAsP spectral detector and two PMT's (one for blue and one for far-red). We recommend consulting (Shihan et al., 2021) for detailed instructions on how to acquire images in Zeiss Zen software.

9. Mount sample in confocal microscope
  - a. Turn confocal system on and wait for lasers to warm up
  - b. Place the slide with the cover slip facing down (if using an inverted microscope) in the sample holder and find your sample using a 10 $\times$  objective or similar
  - c. Add oil immersion to a Plan-Apochromat 40 $\times$ /1.30 oil objective or similar
  - d. Find the Z position that best represents the shape of your tissue. Note down Z position for later use
10. Acquire images:



- a. Adjust laser and detector gain settings to maximize the dynamic range of your system and minimize spectral overlap (Shihan et al., 2021). Save/note down the settings for later use. See also critical notes below and [problem 2: Unexpected staining patterns](#).
- b. Adjust the settings to 16-bit images (if possible) and at least 1024×1024 frame size at 1× zoom
- c. Set the pinhole as close to 1AU as possible but ensure that the same actual pinhole diameter is used for all channels.
- d. Acquire tiled images (Tile scan) of a single Z plane of the entire organ section with 10% overlap scanning each fluorescent dye separately (sequential acquisition), in order to avoid spectral overlap. See (Shihan et al., 2021)
- e. Save the file in the proprietary format (.czi or .lsm for a Zeiss confocal microscope)
- f. Stitch the tiled images using imaging software (such as Zen black, Imaris, Amira or the free software Image J/Fiji) and save in the proprietary format.
  - i. Stitching in Zen black is initiated in the stitch function under the processing tab
  - ii. Select the image to be stitched
  - iii. Set a threshold for stitching. We routinely use "strict"
  - iv. Select apply
  - v. Save the image in the proprietary format. For images acquired using Zen black this format is czi or lsm
- g. Export as TIFF files to perform image analysis. See [Figure 3](#) for an example of a stitched image of a pancreas section with four channels (P120CTN, SOX9, CPA1, DAPI).
  - i. Open the proprietary file in ImageJ: File; Open
  - ii. Save as TIFF in ImageJ: File; Save as; Tiff

**Note:** The images included in the test dataset are TIFF stacks with all channels included. These can be analyzed directly by our code without further modification. The channels include: Channel 1: CPA1, channel 2: P120CTN, channel 3: SOX9, channel 4: DAPI

**△ CRITICAL:** Identical microscopy settings and Z planes must be used for all samples, if comparison of staining intensity is planned.

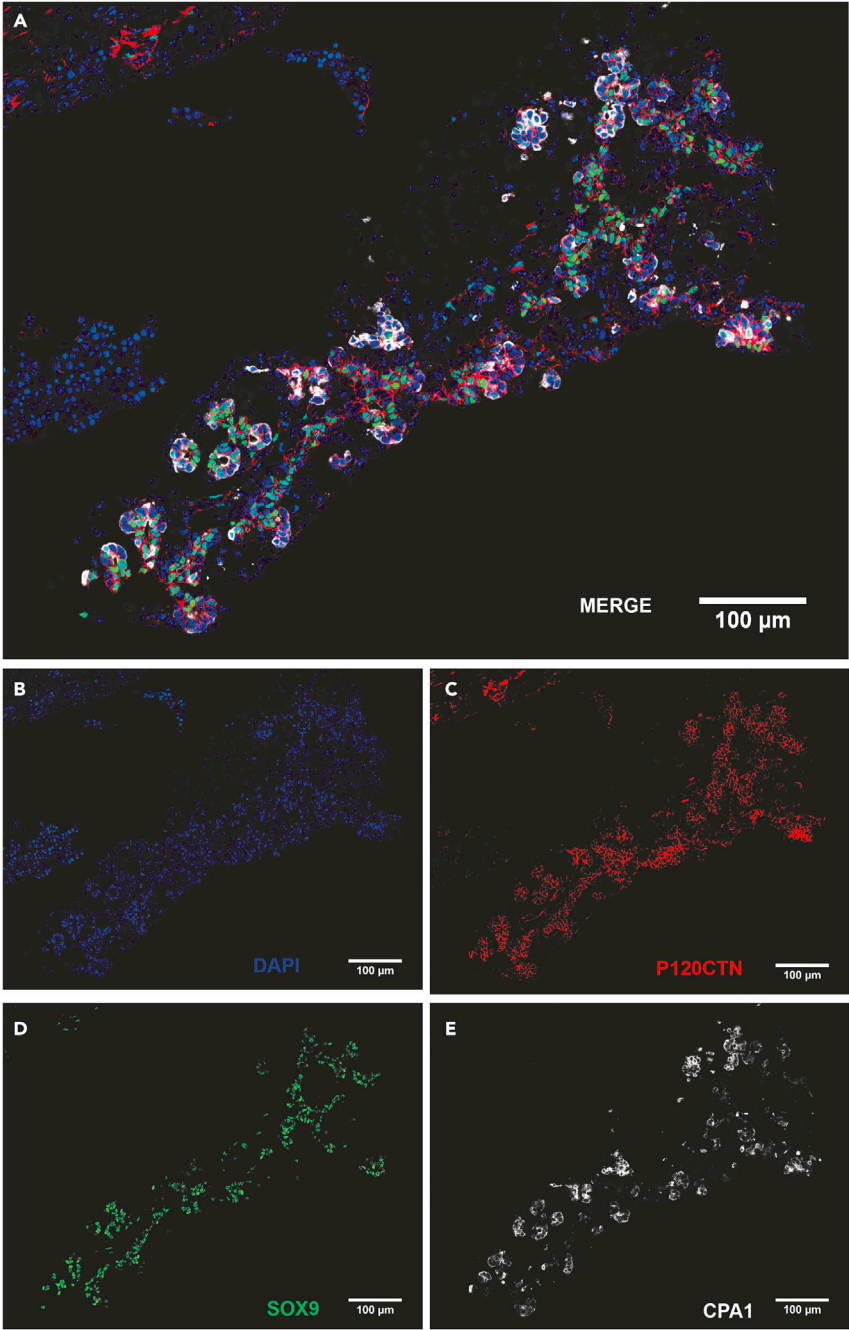
**△ CRITICAL:** Detector gain and laser intensity should be optimized to maximize the dynamic range. Most systems have the option of visualizing the image with a look up table (LUT) where pixels with an intensity of 0 are blue and pixels with the highest pixel intensity are red (in Zen this option is called "Range indicator"). Use this LUT to ensure that your image has only a few maximum intensity (red) pixels (Shihan et al., 2021). Whenever possible, data for intensity quantitation should be captured as 16-bit images, as this optimizes discrimination of pixel intensities as compared to 8-bit images (Shihan et al., 2021).

**Alternatives:** Different confocal microscope systems and non-confocal fluorescent microscopes can be used, as long as it is ensured that the images are compatible with nuclei segmentation. Please note that confocal microscopy is highly recommended, if images are to be used for other downstream applications such as intensity measurements or threshold-based area quantification.

### Step 11–13: Draw outline/mask of tissue shape

⌚ **Timing:** approx. 5 min per image

This is the only semi-manual step in the image analysis pipeline. It creates a mask of the tissue, which is used as an input in subsequent steps.



**Figure 3. Example input image**

Stitched image of an entire pancreatic section for embryonic day 14.5 mouse embryo. The pancreas was stained for DAPI (nuclear marker), P120CTN (membrane marker of the epithelium), SOX9 and CPA1.

(A) The merged image shows DAPI, P120CTN, SOX9 and CPA1.

(B–E)(B) DAPI, (C) P120CTN, (D) SOX9 and (E) CPA1.

The scale bars are 100 µm.

**△ CRITICAL:** The image chosen as input for this step has to be an image of an antibody staining for an antigen that marks either the whole epithelium or an outline of the epithelium. For this project we used images of P120CTN.

The outline of the organ can be drawn and saved as a binary image using the MATLAB app 'ImageSegmenter', as follows:

11. Open MATLAB
  - a. Go to the 'App' tap in MATLAB
  - b. Search for the 'Image Segmenter' app and open it
12. Load the image you want to annotate into the ImageSegmenter App (either from file or from the MATLAB workspace).
  - a. Go to the 'ADD TO MASK' menu and select 'Draw ROI's
  - b. Use the 'Freehand' or 'Assisted Freehand' tool to draw an organ ROI/outline
  - c. Press 'Apply' and 'Close ROI' when you are done drawing.
13. Export binary outline/ROI to MATLAB workspace
  - a. Press 'Export' and then 'Export Images'
  - b. Select 'Final Segmentation' (unselect 'Masked Images') and type in a name (for example 'myBranchedMask') for your new mask and press ok. Your new mask/ROI will now appear in the MATLAB workspace window).

**Note:** Hand-drawn masks for the test dataset are supplied in supplementary data

#### Step 14–16: Tip score locations inside your outline/mask using `tipScoreIm.m`

⌚ **Timing:** runtime for a script calling the `tipScoreIm()` function is approx. 1 min per image

This step calculates a metric for how distant each pixel inside the tissue-outline is from the center of a branched structure. A low score indicates proximity to the center (in the pancreas called "trunk"), while a high score indicates a location close to the periphery (in the pancreas called "tip"). We named this the "tip score".

`tipScoreIm()` is a custom MATLAB function (stored in `tipScoreIm.m`) that takes the binary mask (hereafter called `BranchedMask`) from step 10–12 above, and returns an image where each pixel contains a value that quantifies how 'tip like' this location is. Note that the tip score function was developed to quantify/correlate well with what a biologist judges by eye as "tip" and "trunk" in the branched structure of a pancreas, but it could potentially be used to quantify location in other branched organ structures too.

The `tipScoreIm(BM, approxPixelWidthOfTipStructure, plot_yes_no)` function needs 3 inputs:

- 1) **BM:** a binary mask/outline (class/type: logical, double, uint8, or uint16) of the branched structure. Has 1's everywhere inside the structure and 0's outside. This mask could be based on a hand drawn outline (see step 11–13) or based on a segmentation.
- 2) **approxPixelWidthOfTipStructure:** the approximate length scale/width/length measured in pixels of what is judged to be a 'tip structure' in your image. This was estimated based on the CPA1 staining in our example.
- 3) **plot\_yes\_no:** If set to 1, you will see a plot of the final tipscore image together with a plot of the components used to calculate it. (I.e., side branch, mid branch and convex hull images). Seeing this plot can help you fine tune the 'approxPixelWidthOfTipStructure' (see input 2) to an appropriate value for your image.

The `tipScoreIm(BM, approxPixelWidthOfTipStructure, plot_yes_no)` function has 1 output:

- 1) An image (of type double) where each pixel contains a value that quantifies how 'tip-like' its location is, (values >0 will be in very tip-like areas). Tip score values are usually in the range  $[-2, 1]$ ,

but the numbers depend on the input value 'approxPixelWidthOfTipStructure' and the specific structure given by 'BranchedMask'). Pixel values outside the BranchedMask are set to NaN.

14. Write a simple MATLAB script that calls the `tipScoreIm()` function:
  - a. Study the two demo scripts 'demoScript\_OneImage\_ManualInput.m' and 'demoScript\_Batch\_NoInput.m' for inspiration.
  - b. Make sure that the three function inputs 'BranchedMask', 'approxPixelWidthOfTipStructure', and 'plot\_yes\_no' are defined in the MATLAB Workspace.
  - c. Make sure that the file `tipScoreIm.m` is stored in your current MATLAB directory, or in the folder '/Documents/MATLAB/Add-Ons'.
  - d. Write your script in the MATLAB editor window. Here is an example code that calls `tipScoreIm()` and shows the output:

```
BranchedMask = MyBranchedMask;
approxPixelWidthOfTipStructure = 300,
plot_yes_no = 1;
MyTipScoreIm = tipScoreIm(BranchedMask,approxPixelWidthOfTipStructure,plot_yes_no);
figure
imshow(MyTipScoreIm,[]);
```

15. Run your script. (Press the green 'Run' button in the EDITOR tab).
16. Inspect the output figure from `tipScoreIm()` that shows the subdivision of the skeletonization (Haralick and Shapiro, 1992; Lam et al., 1992) into 'side branches' and 'mid branches' and the convex hull (see Figures 4B–4D). Test different values of `approxPixelWidthOfTipStructure`: run the script again and see how it impacts the tip score values. Repeat until you find a value which approximates the size of the actual tip domain (as based on tip marker). NB: Use the same value for images with the same resolution.

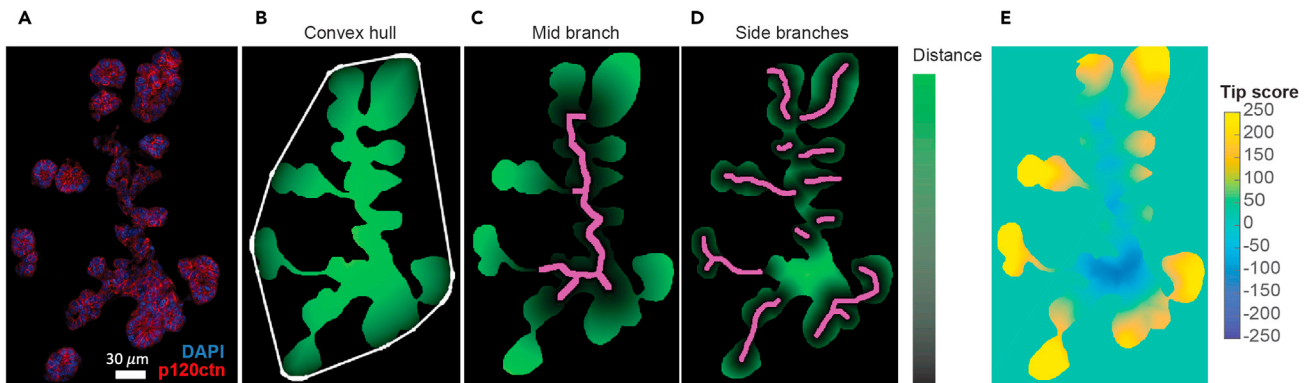
### Step 17–20: Define cell position using a DAPI channel image and `segmentDAPIImage.m`

⌚ **Timing:** runtime for a script calling the `segmentDAPIImage()` function is approx. 5 min per image

This step is crucial for getting a tip-score for cells, rather than pixels. It is based on analysis of images of nuclei from the DAPI channel. `segmentDAPIImage()` is a custom MATLAB function (stored in `segmentDAPIImage.m`) that takes a DAPI channel image and returns a binary mask of cell nuclei using, (Vincent, 1993)(Otsu 1979).

The `segmentDAPIImage(IM,ROI,nucleiD>manual_input_yes_no,answers)` function needs 5 inputs:

- 1) **IM:** a DAPI channel 2D image. Array of type double, uint8 or uint16.
- 2) **ROI:** A binary mask (same size as IM) with ones everywhere where you want to segment nuclei. Give an empty image (all zeros) if you want nuclei segmented everywhere. Array of type logical, double, uint8 or uint16. The mask from step 10–12 can be used here.
- 3) **nucleiD:** Integer number of type double, uint8 or uint16. The average diameter measured in number of pixels of the DAPI stained nuclei you want to segment (approx. 25 pixels for our test images).
- 4) **manual\_input\_yes\_no:** Integer of value 0 or 1. if set to one 1 you will be prompted 4 times during the run of the function and have to pick between images that show different threshold levels.



**Figure 4. Illustration of principle of method**

(A) Confocal image of E14.5 pancreas stained for P120CTN and DAPI.

(B–D) Black-green color map shows local distance to convex hull/mid branch/side branches respectively. (B) White line: convex hull of epithelial outline. (C) Magenta line: Mid branch of skeletonization of epithelial outline. (D) Magenta lines: Side branches of skeletonization of epithelial outline.

(E) Color map shows tip score value in different locations for the image shown in A. Figure reprinted with permission from (Nyeng et al., 2019).

Scale bar in A is 30  $\mu$ m.

(Eg. you will be shown a fig. and asked ‘What thresholded image includes all nuclei and not more?

Note: Nuclei should not have holes in them. (Enter image number from fig. here and press enter):’).

if ‘manual\_input\_yes\_no’ is set to 0 the function will run without prompting the user and use the values specified in the 1×4 array ‘answers’ input (see below).

5) **answers:** Integer array [opt\_index pess\_index regT\_index altPess\_index] with 4 integer index numbers in the range [1–20], type double, uint8 or uint16. The values correspond to the manual command window answers given during a run with ‘manual\_input\_yes\_no=1’. If ‘manual\_input\_yes\_no=0’ the values in answers will be used as input. Use this e.g., if you want to do a batch run but want to give different answers for different images, (see e.g., in demoScript\_Batch\_NoInput). Note if answers are set to [0 0 0 0] the default [3 9 7 10] will be used (optimal for some of our test images).

The `segmentDAPIimage(IM,ROI,nucleiD>manual_input_yes_no, answers)` function gives 1 output:

- 1) A binary image/mask with 1’s where there are nuclei, (see e.g., of a final nucleus segmentation result [Figure 1E](#), colors are added to highlight different nuclei).
17. Write a simple MATLAB script that calls the `segmentDAPIimage()` function:
  - a. Study the two demo scripts ‘demoScript\_OneImage\_ManualInput.m’ and ‘demoScript\_Batch\_NoInput.m’ for inspiration.
  - b. You may import a .tif image into the workspace using the MATLAB function ‘`imread()`’ (write e.g., `IM = imread('MyTiffFileName',xx)`, where xx is the number of the DAPI channel in your tif stack. If your .tif file contains only a single channel write: `IM = imread('MyTiffFileName')`).
  - c. Make sure that the 5 function inputs IM, ROI, nucleiD, manual\_input\_yes\_no and answers are defined in the MATLAB Workspace.
  - d. Make sure that the file `segmentDAPIimage.m` is stored in your current MATLAB directory or added to the folder ‘/Documents/MATLAB/Add-Ons’.

- e. Write your script in the MATLAB editor window. Here is a simple example code that calls `segmentDAPIimage()` and shows the output:

```
IM = MyDAPIimage;
ROI = MyBranchedMask;
nucleiD = 30;
manual_input_yes_no = 1;
answers = [0 0 0 0];
MyNucMask = segmentDAPIimage(IM,ROI,nucleiD>manual_input_yes_no,answers);
figure
imshow(MyNucMask,[]);
```

18. Run your script. (Press the green 'Run' button in the EDITOR tab).
19. Answer the 4 prompted questions asked in the command line. See the section 'MATLAB and demo scripts' for image examples of appropriate answers.
20. Inspect the output figure. If there is over or under segmentation see [problem 4: Over/under segmented nuclei](#).

### Step 21: Measure pixel intensity for nuclear cell marker using `returnTableWithCellInt()`

⌚ **Timing:** runtime for a script calling the `returnTableWithCellInt()` function is a few seconds per image

This is an optional step designed to measure intensity of a nuclear localized antigen. As stated in section 8, special precautions need to be taken if measuring and comparing staining intensity between different images. In our example, we measured the intensity of nuclear localized SOX9.

`returnTableWithCellInt()` is a custom MATLAB function (stored in `returnTableWithCellInt.m`) that takes a 2D image with a nuclear marker and binary mask of cell nuclei and returns a table (a [table](#) is a MATLAB object) that contains the image intensity found in each nucleus centroid pixel.

The `returnTableWithCellInt(NUCLEI_BW, IM, nucleiD)` function needs 3 inputs:

- 1) **NUCLEI\_BW:** 2D binary mask (1's where there are nuclei and 0's elsewhere) of type, logical, double, uint8 or uint16. Each connected component in NUCLEI\_BW defines the location of a cell nucleus. From step 19 above.
- 2) **IM:** An 2D Intensity image of type double, uint8 or uint16. Could e.g., be an immunohistochemical staining of a nuclear marker.
- 3) **nucleiD:** Length scale/diameter of an average nucleus. Used for setting the width of the gaussian filtering of the image. (We filter/blur the image to ensure that the extracted pixel intensities are representative of the local intensities and not extreme outliers. If you do not want any gaussian filtering, then set 'approxPixelWidthOfCellNuclei=1').

The `returnTableWithCellInt(NUCLEI_BW, IM, nucleiD)` function has 1 output:

a table object with 4 columns and one row for each cell/nucleus/connected comp in NUCLEI\_BW.

Column 1 have index number/ ID number in labeled image

Column 2 has x coordinate of nucleus centroid

Column 3 has y coordinate of nucleus centroid

Column 4 has intensity value in gaussian filtered image IM at nucleus centroid location

21. Write a simple MATLAB script that calls the `returnTableWithCellInt()` function:
  - a. Study the two demo scripts 'demoScript\_OneImage\_ManualInput.m' and 'demoScript\_Batch\_NoInput.m' for inspiration.
  - b. Make sure that the 3 function inputs NUCLEI\_BW, IM and nucleiD are defined in the MATLAB Workspace.
  - c. Make sure that the file `returnTableWithCellInt.m` is stored in your current MATLAB directory or added to the folder '/Documents/MATLAB/Add-Ons'.
  - d. Write your script in the MATLAB editor window. Here is an example code that calls `returnTableWithCellInt()` twice using first a nuclear marker image and second a tip-score image (output of `tipScoreIm()`), merges the returned tables, shows the first three rows of the merged table Tcells in the MATLAB command window and saves the table as a .csv file in the current directory (.txt, .dat, .csv, .xls, .xslm, or .xlsx are possible):

```
NUCLEI_BW = MyNucMask;
IM = MyNucMarkerImage;
TS = MyTipScoreIm,
nucleiD = 30;

T1 = returnTableWithCellInt(NUCLEI_BW, IM, nucleiD);
T2 = returnTableWithCellInt(NUCLEI_BW, TS, nucleiD);

% rename VariableName 'IMIntensity' in T2 so the two tables (T1 and T2) do not have
% columns with identical names (identical names will cause an error message when
% concatenating tables later)

T2.Properties.VariableNames('IMIntensity') = 'TipScore';

Tcells = [T1 , T2(:,4)]; % concatenate T1 and 4th column of T2

Tcells(1:3, :) % show first 3 rows in command window

writetable(Tcells, 'TableCells.csv'); % save as .csv file
```

## Step 22: Measure pixel intensity for membrane cell marker

`returnTableWithCellPairInt()`

⌚ **Timing:** runtime for a script calling the `returnTableWithCellPairInt()` function is a few seconds per image

In this optional step we have developed a method to estimate intensity of a membrane expressed protein based on a line scan between two cell centroids. If needed, this method can be implemented by running `returnTableWithCellPairInt()` which is a custom MATLAB function (stored in `returnTableWithCellPairInt.m`) that takes a 2D image with a membrane marker and a binary mask of cell nuclei and returns a table that contains the



max image intensity found along a line scan performed between the nucleus centroids of all neighbor cell pairs. See [Figure 6](#) for examples of the lines drawn between cell nuclei along which the max intensity is extracted at the cell membrane between the two cells.

The `returnTableWithCellPairInt(NUCLEI_BW, IM, nucleiD, plot_yes_no)` function needs 4 inputs:

- 1) **NUCLEI\_BW**: 2D binary mask (1's where there are nuclei and 0's elsewhere) of type, logical, double, uint8 or uint16. Each connected component in NUCLEI\_BW defines the location of a cell nucleus.
- 2) **IM**: An 2D Intensity image of type double, uint8 or uint16. Could e.g., be an immunohistochemical staining of a membrane marker.
- 3) **nucleiD**: Length scale/diameter of an average nucleus. Used for deciding which cells are close enough to be neighbors.
- 4) **plot\_yes\_no**: if set to 1 you will see a plot showing IM with white lines at the locations of all the cell neighbor pairs line scans performed in the image. Inspecting this plot can help you find the appropriate value of nucleiD. See example of output in [Figure 1F](#). Note: you do not need to use exactly the same value for nucleiD as you did for `segmentDAPIImage()`.

The `returnTableWithCellPairInt(NUCLEI_BW, IM, nucleiD, plot_yes_no)` function has 1 output: A table object with 9 columns and one row for each neighbor cell pair in NUCLEI\_BW.

Column 1 has x coordinate of midpoint of line scan

Column 3 has y coordinate of midpoint of line scan

Column 4 ID of first cell in pair

Column 5 ID of second cell in pair

Column 6 array with full linescan

Column 7 Length (in number of pixels) of line scan.

Column 8 Maximum intensity value of mean smoothed linescan (filter size = 3).

Column 9 Median intensity value of mean smoothed linescan (filter size = 3).

22. Write a simple MATLAB script that calls the `returnTableWithCellPairInt()` function:
  - a. Study the two demo scripts 'demoScript\_OneImage\_ManualInput.m' and 'demoScript\_Batch\_NoInput.m' for inspiration.
  - b. Make sure that the 3 function inputs NUCLEI\_BW, IM, nucleiD and plot\_yes\_no are defined in the MATLAB Workspace.
  - c. Make sure that the file `returnTableWithCellPairInt.m` is stored in your current MATLAB directory or added to the folder '/Documents/MATLAB/Add-Ons'.
  - d. Write your script in the MATLAB editor window. Here is an example code that calls `returnTableWithCellPairInt()` twice using first a nuclear marker image and second a tipscore image (output of `tipScoreIm()`), merges the returned tables, shows the first three rows of the merged table `TcellPairs` in the MATLAB command window and saves the table as a .csv file in the current directory (.txt, .dat, .csv, .xls, .xls, or .xlsx are possible):

```
NUCLEI_BW = MyNucMask;
IM = MyMembMarkerImage;
TS = MyTipScoreIm,
nucleiD = 30;
plot_yes_no = 1;
T1 = returnTableWithCellPairInt(NUCLEI_BW, IM, nucleiD, plot_yes_no);
T2 = returnTableWithCellPairInt(NUCLEI_BW, TS, nucleiD, 0);
% Change generic names to names specific for marker (xx) in table. (If this is not done
% you will get an error when you try to concatenate table T1 and T2)
T1.Properties.VariableNames('lineScanMax') = 'xxMaxInt';
T1.Properties.VariableNames('lineScanMed') = 'xxMedInt';
T1.Properties.VariableNames('lineScan') = 'xxLineScan';
T2.Properties.VariableNames('lineScanMed') = 'tipScore'; % we only use the median for
% the tip score of the cell pairs (column 9)
TcellPairs = [T1, T2(:,9)]; % concatenate T1 and 9th column of T2
TcellPairs(1:3,:) % show first 3 rows in command window
writetable(TcellPairs, 'TableCellPairs.csv'); % save as .csv file
```

## EXPECTED OUTCOMES

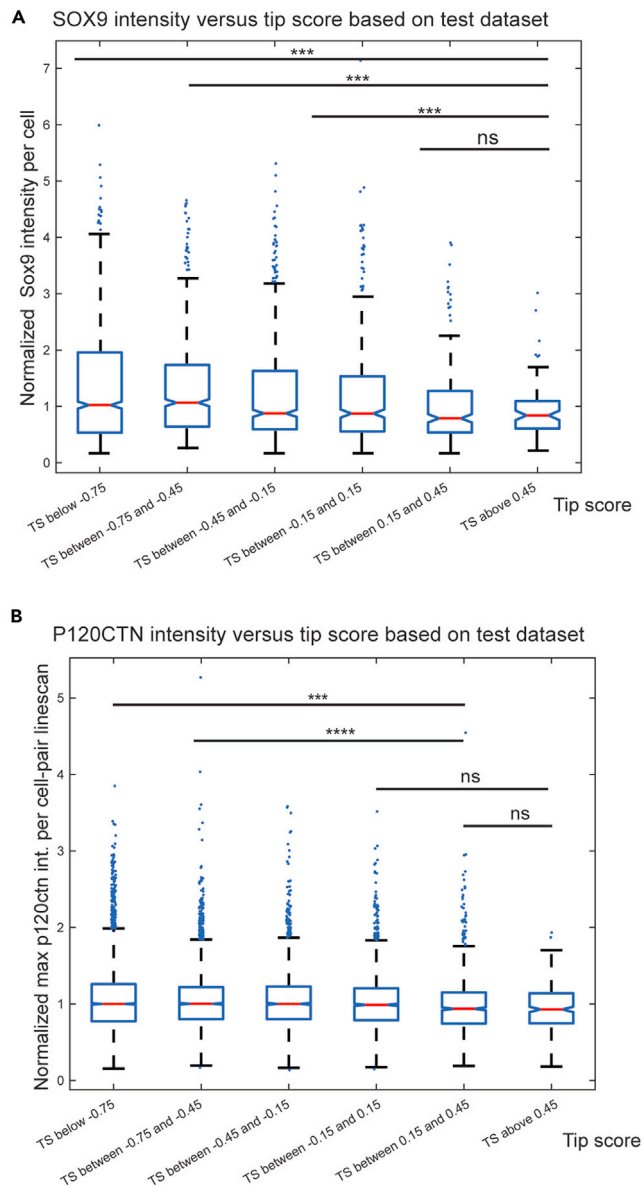
Our test images contain 4 channels: CPA1 (channel1), P120CTN (channel2), SOX9 (channel3) and DAPI (channel 4) (see [Figure 3](#)). Examples of tissue outline, convex hull, midbranch and sidebranches are shown in [Figures 4A–4D](#). Tip score can be visualized as shown in [Figure 4E](#).

To illustrate and validate the method we stained for SOX9, a transcription factor and trunk (central) marker, and CPA1, a tip (peripheral marker). The script output data for SOX9 and P120CTN using the small subset of images in the test dataset (7 images) can be seen in [Figure 5](#).

The trunk marker SOX9 is nuclear localized (see [Figure 3](#)) and we therefore quantified expression level per cell using the nuclear stain DAPI to create a mask for demarcating the area. In the test dataset statistical differences are detected between most bins and the bin with the highest tip score ([Figure 5A](#)). Cells with high tip scores had significantly lower SOX9 intensity compared to cells with low tip score, thus verifying the method for identifying trunk cells. With the full dataset (24 images) in ([Nyeng et al., 2019](#)) a similar difference was found (results not shown here).

To illustrate the method for measuring expression level of a membrane localized protein between two neighboring cells relative to tip score we stained for the protein P120-catenin (P120CTN), which is enriched on the cytoplasmic side of the cell membrane. For the test dataset statistical differences are detected between most bins and the bin with the lowest tip score. Cell pairs with a low tip score have a significantly higher expression of P120CTN than cell-pairs with a high tip score ([Figure 5B](#)).

An example of a cell pair stained for P120CTN and DAPI and analyzed with the line scan method is shown in [Figures 6A and 6B](#). [Figure 6C](#) shows an image with cell pairs indicated by lines from the larger dataset in ([Nyeng et al., 2019](#)). Quantification of this dataset showed a similar difference in p120CTN expression (see [Figure 6D](#) and note that the tip score is differently scaled in this dataset).

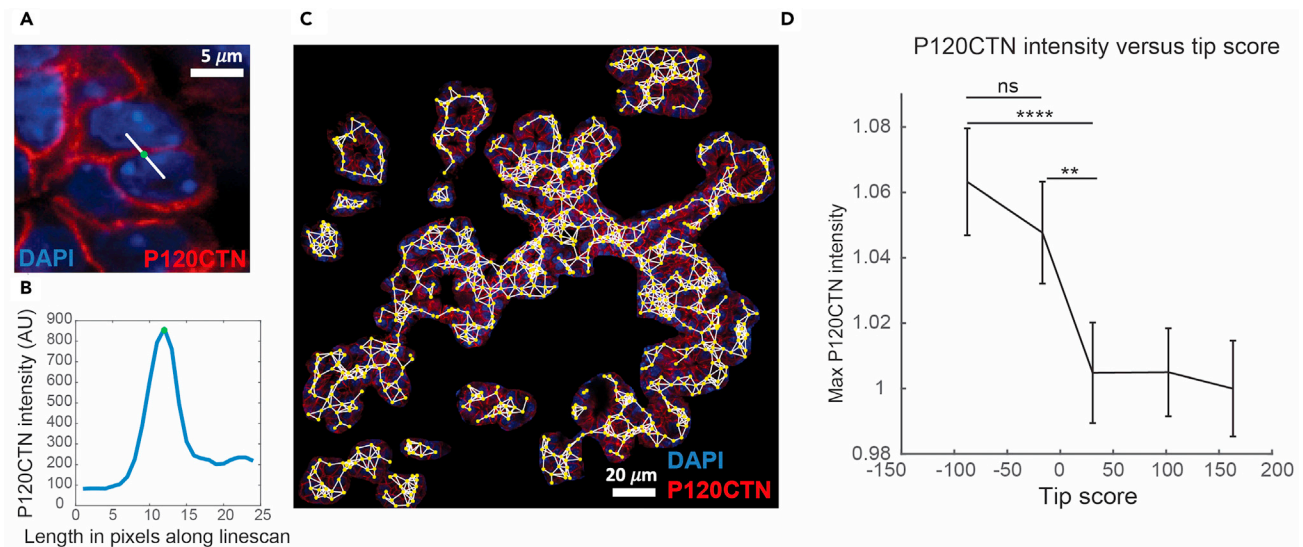


**Figure 5. Data plots from test dataset (tip score, intensity, and linescans)**

(A) SOX9 intensity versus tip score bins. Data comes from N=3 embryos (7 images).

(B) P120CTN max value between cell pairs versus tip score bins. Data comes from N=3 embryos (7 images). The graphs show median  $\pm$  confidence interval as notches. ns=not significant, \*\*\* $p$ <0.001, \*\*\*\* $p$ <0.0001 (Wilcoxon signed-rank test). The following P-values were additionally calculated (Wilcoxon signed-rank test): For plot A:  $p$ <0.0001 for group 1,2,3 (counting from the left side of the plot) compared to group 5,  $p$ <0.01 for group 4 compared to group 5. Groups 1,2,3,4 were not significantly different from each other. For plot B: Group 1 compared to group 6 not significantly different.  $P$ <0.05 for group 2 compared to group 6 (group 6 has very few observations as  $n$ =144 cells).

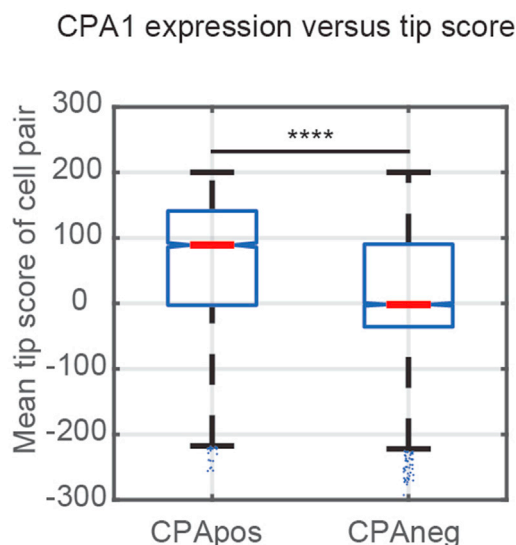
The tip marker CPA1 is expressed in the cytosol (Figure 3E), and we were thus only able to assess via thresholding whether cells were positive or negative (method not included in this protocol but can be found in (Nyeng et al., 2019)). Comparing tip scores for CPA1 positive versus negative cells, showed a significantly higher tip score for CPA1 positive cells, thus verifying the method for assigning high tip scores to tip localized cells (Figure 7).



## QUANTIFICATION AND STATISTICAL ANALYSIS

Statistically test differences in cell position for various cell types

After extracting intensity values from either nucleus or cell membranes and calculating tip score for the position of a cell, it can be tested if groups of cells with different tip scores have significantly different expressions.



**Figure 7. Mean tip score of CPA1 positive versus CPA1 negative cells**

Data is from the original dataset (Nyeng et al., 2019) and contains n=4 embryos (24 images). Note that the tip score scale is different in this dataset. n= 3305–3400 cell pairs per class. The graph shows median  $\pm$  confidence interval as notches. \*\*\*\* $p < 0.0001$  (Wilcoxon signed-rank test). Figure reprinted with permission from (Nyeng et al., 2019).

△ **CRITICAL:** If you merge data from several different IHC images it is crucial that you normalize intensities appropriately.

**Normalization:** Finding a reasonable way to normalize intensities is notoriously hard for IHC images and should be done with great care. Normalizing using intensities from channels with other markers is not necessarily a good idea since the relative intensity difference between different channels/markers can vary for different experiments. Further normalization with 'background' intensity can be problematic both since the relative difference between signal and background can vary between experiments and since this means normalizing with a value far below the dynamic range of the actual marker signal. Ideally one would have a 'reference tissue' included in each image to determine the median intensity value with which to normalize the signal from the ROI, however this is rarely possible. The concept of tipscore is useful for the purpose of normalization since it allows you to determine the median intensity of cells within a given region that has similar tip score, within the same image channel, and use this value for normalization. This way normalization is done with an intensity value that is within the dynamic range of the signal being normalized and is from the same image channel. **Example:** If for example expression of a certain marker is different in tip and trunk regions and individual images contain different proportions of tip and trunk (e.g., one image has more tip region than another) then normalization done with the median intensity of the entire image will be skewed depending on the ratio of tip/trunk in the image, whereas normalization using the median intensity of the same subregion (with similar tip score) in all images will give an acceptable result. In `demoScript_Batch_NoInput.m` we chose to normalize with intensities measured within regions that have tip score values within the range of  $[-0.75; -0.25]$ . This tip score range was chosen to be *i)* as narrow as possible while *ii)* containing as many cells as possible for all images in the test set.

**Statistical tests:** After intensity values had been normalized, we further chose to bin them in groups with similar tipscore values in order to do statistical testing between different tipscore bins. Appropriate binning depends on the size of the data set. The larger the data set the smaller the bin size you may use and still have sufficient statistical power. In `demoScript_Batch_NoInput.m` we have very few measurements with very high/low tip score values and therefore chose to have constant bin size for the middle bins while the first and the last bin contain a broader range of tip score values, (number of bins is adjusted by varying the 'factor = 3' in line 252 of `demoScript_Batch_NoInput.m`).

We compare groups of cells in bins with different tipscore ranges using the MATLAB function `multcompare()` (which corrects for multiple comparison) on the stats structure returned when you call the `kruskalwallis()` function. The Kruskal-Wallis test is a nonparametric version of classical one-way ANOVA, and an extension of the Wilcoxon rank sum test to more than two groups. It compares the *medians* of the groups of data to determine if the samples come from the same population (or, equivalently, from different populations with the same distribution). We chose Kruskal-Wallis since distributions within the bins failed the Anderson-Darling (`adtest()`) test for normality. If your binned data passes the AD-test you may use one analysis of variance `anova1()` instead, which compares the *means* of the groups of data and holds more statistical power.

## LIMITATIONS

Note that the tip score is calculated using lengths in pixels, the score will therefore depend on the resolution level, and the method could give misleading results if images of varying pixel resolution are compared, without specifying the different length scales of tip structures in different images.

This protocol measures an essentially three-dimensional characteristic (shape of a branching structure) from two-dimensional tissue sections. The protocol will therefore give misleading results if only few tissue sections or tissue sections only located in extreme ends are used for quantification.

## TROUBLESHOOTING

Optimizing all optical parameters as described in the protocol and the following trouble-shooting sections is crucial for image analysis. Downstream analysis such as intensity measurements and/or thresholding-based methods are very sensitive to problems such as high background, inconsistent imaging parameters, saturated pixels, non-confocal images etc. Below we list a number of potential problems. Some problems have several causes, which are described in list format. The solutions suggested are listed in a corresponding fashion, such that solution number a. addresses cause letter a. and so forth.

### Problem 1: Excessive background fluorescence

Minimizing excessive background fluorescence in immuno fluorescent stained tissue sections can be challenging, and the problem can have several causes:

- a. Tissue drying, improper blocking or inadequate washing in Step 3–8: Immunostain tissue sections.
- b. Antibody nonspecific binding to endogenous immunoglobulins in the tissue.
- c. Tissue autofluorescence.
- d. Dirty coverslips or objectives in Step 9–10: Confocal microscopy of stained tissue sections.

### Potential solution

- a. In Step 3–8: Immunostain tissue sections: Avoid drying of tissue by working quickly, and only processing one slide at a time when working in the humidified chamber. Ensure that tissue stays moist during longer incubation steps by using a humidified chamber. Follow blocking and washing instructions carefully.
- b. Before you begin, carefully test the correct dilution factor for your primary and secondary antibodies following the manufacturer's instructions. If these are lacking, test a dilution series of primary antibodies ranging from 1:50, 1:100, 1:200, 1:500, 1:1000, and a dilution series of secondary antibodies ranging from 1:500, 1:1000, 1:2000. Compare with secondary antibody only control and negative control to ensure that the staining is specific for the antigen of interest. Check that the staining pattern corresponds with prior knowledge of protein localization (i.e., nuclear or membrane localization etc.) If mouse tissue is probed with a primary antibody made in mouse, the endogenous immunoglobulins should first be blocked with a monovalent Fab fragment antibody of anti-mouse IgG, or IgG specific secondary antibodies used.
- c. Always include a control slide, which is not exposed to any antibodies (negative control) to check for autofluorescence. Tissue autofluorescence is generally much lower than fluorescence from fluorophores. Make sure to compare fluorescence level with no primary control and negative control, when setting up your confocal imaging settings in Step 9–10, keeping gain and laser output below the detection limit for autofluorescence. Note that background fluorescence should be similar in all samples and controls. If this is not the case, attention should be paid to sample preparation in order to eliminate this difference before acquiring images.
- d. In Step 3–8: Immunostain tissue sections: Check that coverslips are clean before, during and after mounting. Never touch the coverslip on the horizontal surface but handle it by holding the sides. In Step 9–10: Confocal microscopy of stained tissue sections: Clean the objectives carefully before and after use and use the appropriate immersion solution for your objective.

### Problem 2: Unexpected staining patterns

Unexpected staining patterns, not conforming to expected localization of protein can be due to:

- a. Unspecific binding of primary or secondary antibodies to endogenous immunoglobulins in the tissue during Step 3–8: Immunostain tissue sections.

- b. Cross-reactivity of primary and/or secondary antibodies to each other during Step 3–8: Immunostain tissue sections.
- c. Overlapping fluorescence spectra of fluorophores and/or inappropriate excitation/emission filters. Avoiding all these is essential for correct interpretation of results.

#### Potential solution

- a. For avoiding unspecific binding see (2) in Problem 1 above.
- b. For avoiding cross-reactivity between antibodies the primary and secondary antibodies must be carefully selected before you begin: Primary antibodies should be produced in different species. If overlap in species is unavoidable, the following modification of the protocol can be employed: For two primary antibodies produced in rabbit, an extra step (requiring one more day of staining) can be added to the protocol: After blocking, the first rabbit primary antibody is incubated with samples alone, followed by a monovalent Fab fragment fluorescence conjugated secondary antibody against rabbit. This secondary antibody will in most cases block any further binding to the first rabbit primary antibody, and the protocol can be followed as normal from hereon.

Ideally, secondary antibodies are all produced in the same species, excluding all species used for producing the primary antibodies. Our exemplary antibody panel is an example of a protocol where one secondary antibody is produced in the same species (goat), which another secondary antibody is directed against. However, most secondary antibodies available commercially are produced in donkey, and this is a good choice if none of the primary antibodies are produced in donkey. Additionally, the use of labeled secondary antibodies which are cross-adsorbed against the species of the experimental tissue, and the species of the other primary antibodies is recommended.

- c. In a multiple labeling protocol such as this, occurrence of identical staining patterns in two different channels during imaging should always be a red flag that fluorescence spectra of fluorophores may be overlapping, and/or that inappropriate excitation/emission filters are used. Always use secondary antibodies conjugated to stable fluorescent dyes that match the configuration of your fluorescent microscope. A good combination with minimal spectral overlap is Dylight-405I, Alexa Fluor-488, Rhodamine-RedX, Alexa-Fluor 647. Use of fluorophores with broader excitation/emission spectra (such as DAPI) can give signal in another channel (so-called bleed-through). Narrow band-pass emission filters, narrow detector detection ranges, and sequential acquisition can be used to minimize bleed-through.

#### Problem 3: No or weak fluorescent signal after staining

If fluorescence is undetectable or very low after staining it can be due to:

- a. The primary or secondary antibody is diluted too much in Step 3–8: Immunostain tissue sections.
- b. There is insufficient antigen present in tissue or the antigen is not accessible.
- c. The antigen is not compatible with the citrate buffer antigen retrieval method in Step 3–8: Immunostain tissue sections.
- d. Tissue has not been fixed appropriately in Step 1–2: Prepare tissue sections from mouse organs.

#### Potential solution

- a. Before you begin, carefully test the correct dilution factor for your primary and secondary antibodies following the manufacturer's instructions. If these are lacking, test a dilution series of primary antibody ranging from 1:50, 1:100, 1:200, 1:500, 1:1000, and a dilution series of secondary antibodies ranging from 1:500, 1:1000, 1:2000. There is insufficient antigen present in tissue.



- b. In Step 3–8: Immunostain tissue sections: The blocking buffer recommended in this protocol does include detergent. If further permeabilization is needed, add a 10 min washing step before blocking, adding Triton X-100 or Tween 20 to PBS. Detergent concentration should be determined empirically, but typically range between 0,01 and 0,1%.
- c. In Step 3–8: Immunostain tissue sections: Omit or include the optional citrate buffer antigen retrieval step.
- d. In Step 1–2: Prepare tissue sections from mouse organs: Use fresh 4% PFA and determine the correct fixation time empirically. Store tissue at  $-80^{\circ}\text{C}$ .

#### Problem 4: Over/under segmented and missed nuclei

- a. Over segmentation: Too many nuclei have been fragmented by `segmentDAPIimage()` in Step 17–20: Define cell position using a DAPI channel image and `segmentDAPIimage.m`.
- b. Under segmentation: Too many nuclei have been merged by `segmentDAPIimage()` in Step 17–20: Define cell position using a DAPI channel image and `segmentDAPIimage.m`.
- c. Missed nuclei: Only few nuclei have been detected () in Step 17–20: Define cell position using a DAPI channel image and `segmentDAPIimage.m`.

#### Potential solution

- a. Over segmentation: Test a larger value to the input 'nucleiD'.
- b. Under segmentation: Test a smaller value to the input 'nucleiD'.
- c. Missed nuclei: Test a smaller answer (lower number) to the third prompted question given by `segmentDAPIimage()`: 'What thresholded image looks best?. I.e., separates most nuclei while losing the least? Smoother boundaries are better than separation. (Enter image number from fig. here and press enter): '

If the nuclei segmentation results are still over or under segmenting the nuclei, after testing different sizes of nucleiD, slightly different answers to the 4 prompted questions can be tested. Double check that your images appear roughly like the images shown in the section '[download, install, and run MATLAB and demo scripts](#)', see [Figures 1A–1D](#).

`segmentDAPIimage()` gives satisfactory results when used on confocal images comparable in quality to those provided in the test dataset. However, it does not represent the current state of the art for nucleus detection/segmentation. Please see below for a detailed explanation of the code. If your nuclei segmentation result is still not satisfactory after testing the different parameters as suggested above, then we recommend making a nuclei mask outside MATLAB with the more sophisticated machine learning tools in [ilastik](#) ([Berg et al., 2019](#)) or [StarDist](#), ([Schmidt et al., 2018](#)).

Alternatively, a nuclei mask can be made manually with the MATLAB app 'ImageSegmenter' (see section '[draw outline/mask of tissue](#)' for instructions on how to use 'ImageSegmenter').

`segmentDAPIimage()` - point by point explanation of the code:

The most important parameter is the user input 'nucleiD'. This number is used to calculate size of gauss and median filter kernels as well as morphological structuring elements.

The steps of the `segmentDAPIimage()` function in pseudo-code:

- Gauss and median filter DAPI channel image (to reduce noise on a scale much smaller than a nucleus).
- Normalize intensities with respect to max inside region of interest (ROI)
- Calculate a range of Otsu threshold levels based on pixel values inside ROI.

Show the user the thresholded images to make the user pick a threshold for a mask that includes everything that could be a nucleus. (We term this the 'optimist mask').

Show the user the thresholded images to make the user pick a threshold for a mask which includes only the very brightest pixels globally which should definitely be included in the final mask. (We term this the 'pessimist mask').

Show the user thresholded images done with regional thresholding in order to pick good threshold levels for including all nucleus pixels ('regional threshold mask') and for getting good separation of different nucleus respectively ('regional pessimist mask').

Combine the 4 masks from above into one: First holes in 'regional threshold mask' are closed and objects smaller than 4 pixels are removed. Pixels in pessimist and regional pessimist are always included but pixels not in optimist are never included.

Iterate through each connected component in the combined mask and do morphological operations on it (first opening and then smoothing of the edge). The radius of the strel disk (the morphological structuring element) for `imopen()` depends on the size of the current component/object. (A larger object is more likely to be several nucleus touching and a more aggressive morphological opening is needed to separate them).

Take the distance transform of the output mask and gauss filter it and find regional max in it.

Uses regional max as seeds to separate touching nuclei by seed-based watershed on the distance transform image.

Remove very small objects (compared to expected nucleus size).

### Problem 5

Demo scripts will not run:

- MATLAB gives an error message: 'Undefined function or variable ...' when running demo scripts in before you begin.
- MATLAB gives an error message like: 'kruskallwallis requires Statistics and Machine Learning Toolbox' when running demo scripts in before you begin.
- MATLAB gives an error message: 'Undefined function or variable bwskel' when running demo scripts in in before you begin.

### Potential solution

- First, double check that 'Current Folder' (see left side of the MATLAB window) is your 'TipScore-main' folder downloaded from Github. Second, double check that your current folder contains the folders 'raw\_images\_tif\_files' and 'outlines\_mat\_files'. Then, double check that your current folder contains all.m files downloaded from Github: `segmentDAPIimage.m`, `tipScoreIm.m`, `returnTableWithCellInt.m`, and `returnTableWithCellPairInt.m`.
- Type 'ver' and enter in the MATLAB command window and check that you have the needed toolboxes installed which are: 'Signal Processing Toolbox', 'Image Processing Toolbox', and 'Statistics and Machine Learning Toolbox'. A free trial with all toolboxes can be ordered/used for 30 days.
- Type 'ver' and enter in the MATLAB command window and check that your MATLAB version is 2018a or later. The newest MATLAB version can be ordered as a free trial for 30 days.

## RESOURCE AVAILABILITY

### Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Pia Nyeng ([ponyeng@ruc.dk](mailto:ponyeng@ruc.dk)).

### Materials availability

All the materials used in this protocol are commercially available.

### Data and code availability

- This protocol includes a sample of data (Data S2) consisting of microscopy images obtained for a previously published study (Nyeng et al., 2019). Original data have been deposited to Mendeley Data: <https://doi.org/10.17632/nr9cyyk265.1> and are publicly available as of the date of publication. DOIs are listed in the [key resources table](#).
- Original code (Data S1) has been deposited to GitHub: <https://github.com/SiljaHeilmann/TipScore> and is publicly available as of the date of publication. The link is also listed in the [key resources table](#).
- Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

### SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.xpro.2021.100806>.

### ACKNOWLEDGMENTS

This work was supported by JDRF grant 3–2011-214 and DFF grant number 6110-00091B. The Novo Nordisk Foundation Center for Stem Cell Biology is supported by a Novo Nordisk Foundation grant number NNF17CC0027852. We acknowledge the DanStem Imaging Platform and the Core Facility for Integrated Microscopy, Faculty of Health and Medical Sciences, University of Copenhagen for use of microscopes.

### AUTHOR CONTRIBUTIONS

This work was completed by S.H., H.S., and P.N. S.H. developed and optimized the computational methods in collaboration with P.N. P.N. developed the staining and imaging methods and produced the image data for this protocol. S.H. and P.N. wrote the first draft of the manuscript. H.S. acquired funding and edited the manuscript.

### DECLARATION OF INTERESTS

The authors declare no competing interests.

### REFERENCES

- Berg, S., Kutra, D., Kroeger, T., Straehle, C.N., Kausler, B.X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., et al. (2019). Ilastik: Interactive machine learning for (bio)image analysis. *Nat. Methods* 16, 1226–1232.
- Fox, C.H., Johnson, F.B., Whiting, J., and Roller, P.P. (1985). Formaldehyde fixation. *J. Histochem. Cytochem.* <https://doi.org/10.1177/33.8.3894502>.
- Haralick, R.M., and Shapiro, L.G. (1992). *Computer and Robot Vision*, 1 (Addison Wesley Publishing Company).
- Helander, K.G. (2000). Formaldehyde prepared from paraformaldehyde is stable. *Biotech. Histochem.* 75, 19–22.
- Kaufman, M.H. (1992). *The Atlas of Mouse Development* (Academic Press).
- Lam, L., Lee, S.-W., and Suen, C.Y. (1992). Thinning methodologies—a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* <https://doi.org/10.1109/34.161346>.
- Lee, J.Y., and Kitaoka, M. (2018). A beginner's guide to rigor and reproducibility in fluorescence imaging experiments. *Mol. Biol. Cell.* <https://doi.org/10.1091/mbc.e17-05-0276>.
- Nyeng, P., Heilmann, S., Löf-Öhlin, Z.M., Pettersson, N.F., Hermann, F.M., Reynolds, A.B., and Semb, H. (2019). P120CTN-mediated organ patterning precedes and determines pancreatic progenitor fate. *Dev. Cell* 49, 31–47.e9.
- Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* 9, 62–66.
- Rueden. (2017). *BMC Bioinformatics*. <https://doi.org/10.1186/s12859-017-1934-z>.
- Schmidt, U., Weigert, M., Broaddus, C., and Myers, G. (2018). Cell detection with star-convex polygons. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pp. 265–273. [https://doi.org/10.1007/978-3-030-00934-2\\_30](https://doi.org/10.1007/978-3-030-00934-2_30).
- Shihan, M.H., Samuel, G.N., Le Marchand, S.J., Wang, Y., and Duncan, M.K. (2021). A simple method for quantitating confocal fluorescent images. *Biochem. Biophys. Rep.* 25, 100916.
- Shi, S.R., Chaiwun, B., Young, L., Cote, R.J., and Taylor, C.R. (1993). antigen retrieval technique utilizing citrate buffer or urea solution for immunohistochemical demonstration of androgen receptor in formalin-fixed paraffin sections. *J. Histochem. Cytochem.* 41, 1599–1604.
- Vincent, L. (1993). Morphological grayscale reconstruction in image analysis: applications and efficient algorithms. *IEEE Trans. Image Process.* <https://doi.org/10.1109/83.217222>.
- Wang, H., and Matisse, M.P. (2013). Immunofluorescence staining with frozen mouse or chick embryonic tissue sections. *Methods Mol. Biol.* [https://doi.org/10.1007/978-1-62703-444-9\\_17](https://doi.org/10.1007/978-1-62703-444-9_17).