Cell Reports Methods, Volume 3

Supplemental information

Label-free cell segmentation

of diverse lymphoid tissues in 2D and 3D

John W. Wills, Jack Robertson, Pani Tourlomousis, Clare M.C. Gillis, Claire M. Barnes, Michelle Miniter, Rachel E. Hewitt, Clare E. Bryant, Huw D. Summers, Jonathan J. Powell, and Paul Rees



Figure S1 related to Figure 1 – Comparing label-free probability maps using reflectance data obtained using 488 nm or 638 nm laser excitation. The choice of excitation wavelength for generating the reflectance signal has minimal influence on the probability maps obtained from the network (left versus middle; difference shown right). The user can therefore reasonably use whatever is available on their individual microscope. For some applications, choosing a longer excitation wavelength may reduce fluorophore photobleaching / improve tissue penetrance and reflectance recovery - for instance, during 3-D Z-stack imaging in thicker tissue specimens.



Figure S2 related to Figure 1 – Number of first-encoder filters and encoder depth optimisation to define the best performing 2-D Unet model. Bars represent normalised training time whilst circles indicate label-free cell segmentation accuracies (assessed by Jaccard index). The best performing model used an encoder depth of 4 with 64 filters at the level of the first encoder (indicated, **). Of note, using 32 filters instead of 64 can achieve a training speed up of ~50% for a negligible (~1%) decrease in segmentation accuracy (indicated, *). Increasing the encoder depth to 5 did not further improve cell segmentation accuracies (bottom row).



Figure S3 related to Figure 1 – Spectral bandwidth saving achieved by the label-free cell segmentation strategy. a, Emission spectra for Hoechst 33342 and AlexaFluor 647 as might typically be used to delineate cell nuclei and cell cytoskeletons when carrying out fluorescence-based cell segmentation. b, Harnessing reflectance information, the label-free cell segmentation method described here removes the need for these fluorescence stains leaving the spectrum entirely open for sensitive experimental measurements with single-cell quantification.



Figure S4 related to Figure 2 – Label-free cell segmentation of confocal microscopy image-data collected from formalin-fixed, paraffin embedded tissue sections. a, In frozen cryostat sections, f-actin staining using phalloidin conjugates clearly delineates cell outlines providing ground truth to enable the presented label-free cell segmentation approach. b, In contrast, in formalin-fixed paraffin embedded (FFPE) tissue sections, phalloidin staining fails because solvent exposure during the fixation and paraffin embedding process degrades the actin cytoskeleton. c, Demonstrated here using murine Peyer's patch tissue sections, successful ground truth labelling can be restored in the FFPE section-type by switching to cell membrane (i.e., phospholipid) staining using wheat germ agglutinin (WGA) fluorescence conjugates. d/e Comparison of the reflectance signal from the frozen and FFPE section-types. Cytoskeletal degradation appears to change the reflectance images observed from the FFPE tissue: the faint trace of the cell outlines visible in the frozen sections is no longer apparent and instead the intracellular regions appear to exhibit the highest reflectance signal. f-h Despite this, a relationship between the reflectance signal and a WGA-delineated ground truth is still determinable by the neural network allowing (g/h) successful label-free cell segmentation direct from the reflectance signal. h, Intersection over union (IOU) score distribution comparing a (f) hand-drawn segmentation and the (g) automated, label-free cell segmentation outcome. An IOU score of 1 represents a perfect, per-pixel overlap between the hand-drawn and automated cell segmentations. Within the comparison presented here, scores ≥ 0.6 are seen to represent a good match, approaching the limits of hand-drawing accuracy. By harnessing ground truth from other fluorescence labels, the label-free strategy can operate in both FFPE and frozen tissue-types. Given that tissue archiving in FFPE format is commonplace worldwide, this finding dramatically increases the application domain of the presented label-free cell segmentation strategy. Scale bars: a/b = 20 microns; c/d = 100 microns; e/f = 75 microns; g/h = 50 microns.



Figure S5 related to Figure 4 – Assessing 3-D label-free cell segmentation accuracies using mouse Peyer's patch tissue. a, Hand-drawn cell segmentations performed using the nuclei/actin fluorescence information for Z-planes (a) 57 in the XY dimension (d) 512 in ZY dimension and (g) 512 in the XZ dimension (unseen test image-data is 512x512x114 (X,Y,Z)). b/e/h, Automated cell segmentations for the same image-regions as (a/d/g) but achieved label-free direct from the reflectance signal. c/f/i, Cell-object intersection-over-union score distributions comparing – cell-object by cell-object – the (a/d/g) hand-drawn segmentations against the (b/e/h) automated, label-free cell segmentations. An IOU score of 1 represents perfect, per-pixel overlap between the hand-drawn and automated cell segmentations. Within the comparison presented here, scores \geq 0.6 are seen to represent a good match, approaching the limits of hand-drawing accuracy. Encouragingly, the 3-D approach outperformed the segmentation accuracies achieved in 2-D (shown, Figure 2). Scale bars equal 20 microns.



Figure S6 related to Figure 4 – Fluorescence versus label-free nuclei predictions at Z-depths of ~10 and ~90 microns. Using reflectance information from a 638 nm excitation laser, the 3-D network is able to consistently recover nuclear information long after the blue nuclear stain (Hoechst 33342) has decayed from multiple scattering effects (bottom right versus middle right). The resultant pixel intensity histograms from the probability map images are extremely stable (bottom right). This is advantageous for achieving consistent, depth-invariant 3-D cell segmentation in thick tissue specimens.

Table S1 related to	Star Methods -	Antibody	and Image	Information	Table

PRIMARY ANTIBODIES	Product no	Supplier	Dilution primary	Stock concentration	Host	Secondary (detailed below)	Detection	Figure
Anti-mouse CD3-EF450	48-0032-82	Thermo Fisher	1:25	0.2 mg/mL	Rat	N/A	eFluor 450	Figure 1
Anti-mouse CD4-PE	12-0041-83	Thermo Fisher	1:25	0.2 mg/mL	Rat	N/A	R-phycoerythrin	Figure 1
Anti-mouse CD11c-EF660	50-0114-82	Thermo Fisher	1:25	0.2 mg/mL	Armenian Hamster	N/A	eFluor 660	Figure 1
Anti-mouse FOXP3-EF660	50-5773-82	Thermo Fisher	1:25	0.2 mg/mL	Rat	N/A	eFluor 660	Figure 2
Anti-mouse CD3	AB5690	Abcam	1:150	0.2 mg/mL	Rabbit	Anti-Rabbit AF568	AlexaFluor 568	Figure 3
Anti-mouse CD11c	AB33483	Abcam	1:400	0.5 mg/mL	Armenian Hamster	Anti-Hamster A488	AlexaFluor 488	Figure 3

SECONDARY ANTIBODIES	Product no	Supplier	Stock concentration	Host	Secondary dilution	Fluorophore
Goat anti-Rabbit IgG (H+L)	A-11011	Thermo Fisher	2 mg/mL	Goat	1:400	AlexaFluor568
Goat anti-Hamster IgG (H+L)	A-11008	Thermo Fisher	2 mg/mL	Goat	1:400	AlexaFluor488

Figure ID	Section type	Tissue type (Sectioning orientation)	Objective lens Magnification / Numerical aperture (Microscope)	Pixel density Pixels per micron	Voxel size x,y,z μm	Reflectance Excitation laser (Detector placement) Wavelength, nm	Train / Test image(s) (number) x/y/z dimensions Pixel dimensions	Patch size / (patches per epoch)	Approx. Training time (single NVIDIA 1080 Ti GPU)
Figure 1	Cryostat (frozen)	Mouse spleen (transverse)	40X/1.3 (Leica SP8 inverted)	5.2842	0.1892 x 0.1892	488 (485-491)	Train: (1) 7617 x 7661 Test: (1) 5766 x 5787	256x256 (1000)	268 min (50 epochs)
Figure 4	Cryostat (frozen)	Mouse mesenteric lymph node (transverse)	40X/1.3 (Zeiss LSM780 upright)	4.8177	0.2076 x 0.2076 x 0.1500	561 (558-564)	Train: (2) 1024 x 1024 x 107 Test: (1) 1024 x 1024 x 111	64x64x64 (1000)	750 min (150 epochs)
Figure S1 Figure S6	Cryostat (frozen)	Mouse ileal Peyer's patch	40X/1.3 (Leica SP8 inverted)	3.5200	0.2840 x 0.2840 x 0.346	488 (485-491) 638 (635-641)	Train: (3) 512 x 512 x 103 Test: (1) 512 x 512 x 103	64x64x64 (500)	320 min (150 epochs)
Figure 3 Video S1	Cryostat (frozen)	Mouse ileal Peyer's patch (transverse)	63X/1.4 (Leica SP8 inverted)	8.324	0.1201 x 0.1201	488 (485-491)	Train: (1) 7607 x 11253 Test: (1) 7610 x 11247	256x256 (1500)	360 min (50 epochs)
Figure S4	FFPE	Mouse ileal Peyer's patch (transverse)	40X/1.3 (Leica SP8 inverted)	7.0463	0.1419 x 0.1419	638 (635-641)	Train: (1) 11364 x 11421 Test: (1) 7641 x 7660	256x256 (2000)	502 min (50 epochs)

Methods S1 – Related to Star Methods

Table of contents -

Pages 2-5:	Setting up reflectance imaging on Leica and Zeiss confocal microscopes
Pages 6-12:	Label-free probability map generation using standalone Windows software
Pages 13-21:	Running the label-free cell segmentation deep learning scripts using MATLAB
Pages 22-29:	MATLAB code: 2-D pipeline
Pages 30-39:	MATLAB code: 3-D pipeline
Pages 40-58:	Running the label-free cell segmentation deep learning scripts using Python
Pages 59-68:	Python code: 2-D pipeline
Pages 69-83:	Python code: 3-D pipeline
Pages 84-89:	Extracting single-cell features using CellProfiler 4
Pages 90-119:	2-D CellProfiler pipeline
Pages 120-139:	3-D CellProfiler pipeline



Setting up sequential reflectance imaging using a standard Zeiss LSM780 confocal microscope.

a, Once the fluorescence imaging sequences are set up, a new track for reflectance is added to the sequential scan. **b**, The desired excitation laser for reflectance imaging is selected (here, 561 nm). The choice of laser is not particularly important, but use of longer wavelengths may reduce fluorophore photobleaching and improve penetrance / recovery in thicker specimens (shown, **Figures S1/S6**). **c**, A photomultiplier detector (here Ch1) is turned on and the range set to approximately +/- 3 nm either side of the excitation wavelength (here 558 – 564 nm was entered, but the software rounds to display ~ 557 – 566 nm). **d**, The tick-box allowing reflected light to pass to the detector is turned on. **e**, The T80/R20 beam splitter is chosen (this indicates a transmission/reflection ratio of 80:20). **f**, A low laser excitation power (here, 1%) is entered. *N.B.*, use of a reflectance light path with high laser excitation power may damage the camera, so care should be taken here. The pinhole is set to ~ 1 airy unit, yielding an optical section of around ~ 1 micron with a high numerical aperture 40X or 63X objective. **g**, Running in 'live mode', the gain is slowly increased until the reflectance signal occupies approximately 80% of the range histogram (indicated, green box). Compressing the histogram in the range indicated yields a typical 'view' of the reflectance signal from a lymphoid tissue specimen (on display in the main image window).



Setting up sequential reflectance imaging using a standard Leica SP8 confocal microscope.

a, Once the fluorescence imaging sequence(s) are set up, a new track for reflectance is added to the sequential scan (here, 'Seq 4'). **b**, The desired excitation laser for reflectance imaging is turned on (here, 488 nm). The choice of laser is not particularly important, but use of longer wavelengths may reduce fluorophore photobleaching and improve penetrance / recovery in thicker specimens (shown, **Figures S1/S6**). A low laser excitation power (*e.g.*, 1%) is also specified at this step. *N.B.*, Use of a reflectance light path with high laser excitation powers could damage the camera, so care should be taken at this step. **c**, An appropriate beam splitter is chosen for the excitation line, or, the 'Autoselect' checkbox can be ticked to set this automatically. **d**, A photomultiplier detector (here PMT1) is turned on and the range set to approximately +/- 3 nm either side of the excitation wavelength (*i.e.*, here, 485-491 nm). **e**, Running in 'live mode', the gain is slowly increased until the reflectance signal occupies approximately 80% of the available intensity range. The main window shows a typical 'view' of the reflectance signal from a lymphoid tissue specimen under this setup.



Setting up simultaneous reflectance and fluorescence imaging using a standard Leica SP8 confocal microscope.

Once fluorescence excitation and collection are configured, (a) any remaining laser line can be used for reflectance imaging (*e.g.*, here, the 488 nm line is used). The choice of laser is not particularly important, but use of longer wavelengths may reduce fluorophore photobleaching and improve penetrance / recovery in thicker specimens (shown, **Figures S1/S6**). A low laser excitation power (*e.g.*, 1%) should also be specified here. Reflectance imaging with high laser excitation powers could damage the camera, so care should be taken at this step. **b**, An appropriate beam splitter for the excitation lasers is chosen, or the 'Autoselect' checkbox ticked to enable automatic setting. **c**, A free detector (here PMT2) is turned on and the range set to approximately +/- 3 nm either side of the excitation wavelength (*i.e.*, 485-491 nm). **d**, Running in 'live mode', the gain is slowly increased until the reflectance signal occupies approximately 80% of the available intensity range. The top-right image in the main window shows a typical 'view' of the reflectance signal from a lymphoid tissue specimen under this setup. This approach allows reflectance data to be concomitantly collected alongside fluorescence without adding the additional run-time of further sequences. *N.B.*, It is worth noting that in a similar way, a laser that is already being used for fluorescence excitation may also be used to obtain reflectance data (exemplified **page below**). For example, here, PMT2 could be moved up to collect reflectance from the 638 nm laser in the range 635-641 nm. Doing this has the advantage of reducing the photon budget for the sample. However, it also necessitates that enough excitation power is being used to obtain a good reflectance signal, and that a free detector can be moved within the necessary detection range. This is not always compatible with optimal fluorescence imaging – hence the setup shown here.



Setting up 'free' reflectance imaging alongside fluorescence collection using a standard Leica SP8 confocal microscope.

Once fluorescence excitation and collection are configured, (a) any remaining detector can be used to simultaneously collect the reflectance signal from one of the excitation lasers being used to stimulate fluorescence (*e.g.*, here, 'PMT2' is used to collect reflectance from the 638 nm laser line (b) – which is also being used to excite AlexaFluor 647). This is achieved by placing the detector approximately +/- 3 nm either side of the excitation wavelength (*i.e.*, 635-641 nm). **c**, Running in 'live mode', the gain is slowly increased until the reflectance signal occupies approximately 80% of the available intensity range. The top-right image in the main window shows a typical 'view' of the reflectance signal from a lymphoid tissue specimen under this setup. Simultaneous reflectance imaging has the advantage of reducing the photon budget for the sample, as the reflectance information is effectively recovered for 'free' by harnessing scatter from a laser that is already in use. However, this setup also necessitates that enough excitation power is available to obtain a good reflectance signal (shown bottom-left in the main image window) the AlexaFluor647 detection range was narrowed (d) to ~ 700-750 nm. Where this setup cannot be accommodated one of the other options that instead use a dedicated laser for reflectance imaging should be utilised (shown, **three above pages)**.

Windows 10: Running the label-free cell segmentation pipeline using standalone software

Running this software requires:

- Windows 10 machine (NVIDIA GPU desirable for model training)
- Label-free stain prediction standalone software (BioStudies download)
- MATLAB runtime 9.11 (installs automatically alongside software see below).

Note: This software is free to install. No MATLAB license is required to run this standalone software.

Installation Steps:

1. **Install the Label Free Stain Prediction software** by running the Windows 10 installer from the BioStudies download



2. Follow through the setup procedure to install the software:



- Check 'add shortcut to the desktop':

🚪 Installation Options		_		\times
Choose installation folder:		3	53	Y
C:\Program Files\LabelFreeStainPrediction	Browse	\sim	1)	T
	Restore Default Folder	\checkmark	\bigcirc)
Add a shortcut to the desktop		Th.	A	5
		\sim	F	λ
		1	my	T
			Ż	~~
		\mathcal{H}	3	\searrow
		4 5		~
			hand	4
< Back Next >	Cancel			

- When prompted, install MATLAB runtime (free; no license is required):

🔏 Required Software	_		×
MATLAB Runtime is required.			
Choose installation folder:	MA	TLA	∆B°
C:\Program Files\MATLAB\MATLAB Runtime Browse	R	2021 b	
MATLAB and Simulink are registered trademarks of The MathWorks, Inc. Please see mathworks.com/trademarks for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders. WARNING: This program is protected by copyright law and international treaties. Copyright 1984-2021, The MathWorks, Inc. Protected by U.S. and other patents. See MathWorks.com/patents			
< Back Next > Cancel	📣 Ma	athWorl	CS*

- click 'install' to install the label free stain prediction software and MATLAB runtime 9.11

a Confirmation	_		×
LabelFreeStainPrediction will be installed in: C:\Program Files\LabelFreeStainPrediction LabelFreeStainPrediction requires MATLAB Runtime R2021b. MATLAB Runtime R2021b will be installed in: C:\Program Files\MATLAB\MATLAB Runtime\v911 Download Size: 2,730 MB		States and the states	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
< Back Install > Cancel			

3. Training 2-D or 3-D UNET model using the standalone software

-A screencast video is included with the BioStudies archive.

- -Launch the Label Free Stain Prediction software from the desktop icon
- -Example 2-D and 3-D image data and training labels are included in the BioStudies download.

MATLAB App	- 🗆 X	LabelFreeStainPrediction			- 0
Train a label-free segmentation model Load reflectance image Image file Image rstal Reger stal Reger stal Reger stal C Reg	Generate label-free predictions Load pretrained model Pretrained model life Pretrained model information: Number discolution 0 8000 000 0 0 0 0 0 0 0 0 0 0 0 0 0 0				
Load nuclei labels. Nuclear label file Load cell outline labels: C	Load reflectance image:	Select File to Open $\label{eq:entropy} \leftarrow \ \ \rightarrow \ \ \ \ \ \ \ \ \ \ \$	tandalone_software_biostudies >	ບ ບັ ,⊃ Search St	itandalone_software
Cell outline tabels file Load background/other tabels Background labels file Load ROL-mask (If required) ROI mask file	Image tile 0 Image y ster 0 Image y ster 0 Image z ster 0	Organize - Newfolder Organize - University of Cambridge University of Cambridge Eachel Hewit: - 11/8/Six consect	Name LExample_image_data ZWINDOWS10 INSTALL Date created: Scrample_pretrained_m_Size_995 MB Enders FXM	Date modified 04/01/2022 20:38 04/01/2022 20:37 04/01/2022 20:37	Type File folder
Number of epochs 0 e.g., 2.0 = 50, 3.0 = Batch Size 0 e.g., 2.0 = 12, 3.0 =	50 Load ROI-mask (if required) 8 ROI mask file	This PC JD Objects Osttop Documents ↓ Downloads	5,3D_CelProfiler_pipeline	04/01/2022 21:13	File folder
TRAIN NETWORK	PREDICT LABELS	Music Pictures Gives Windows (C) Data (P)	v k		
		File name:		 (*.tiff) Open 	Cancel

- To Train a 2-D or 3-D UNET model to predict the probability images needed to enable the label-free segmentation, load reflectance data and matching labels for 'Nuclei', 'Cell Outlines' (e.g. actin) and 'background/other' classes.

-The label images should be in .tiff format with the stains represented as foreground (i.e., white). -2-D or 3-D data will be automatically detected.



- If desired, a ROI-mask can also be loaded to mask the outputted probability maps to the desired tissue region.

- If this isn't needed – just leave this box blank.

- Set the desired batch-size and number of training epochs and click 'Train Network'.

- After a few seconds, the loaded reflectance data and labels will be indicated in "Figure 1"

- Progress can be tracked in the Command Prompt console

- MAT formatted data and the trained model will be saved to folders on the desktop once training completes:

- Training with the example data takes ~6h on an NVIDIA 1080 GTX GPU. (Previously trained models are provided).



4. Using a pretrained 2-D or 3-D UNET model to generate probability images for label-free segmentation using unseen reflectance data.

-On the right-side of the software, **load** a pretrained model.

-This can be from Step 3 (above), or, example pretrained networks are provided in 'Folder 3' of the BioStudies download. Example unseen 'test' image-data is also provided in 'Folder 1' in 2-D and 3-D.



- Load 2-D or 3-D reflectance data:

MATLAB App	- 🗆 X		
Train a label-free segmentation model Lod reflectance image Image file Image file Image file Image file Image file Load nuclei labels Cell outline labels file Load cell outline labels Background labels file Load Rol-mask (if regured) Rol mask file	Generate label-free predictions Load pretrained model MLN_Biostudies_27-May-2021-1355 mal Pretrained model information: Nume of spoors State name Load reflectance image: Image sizes 0 Image sizes 0 Image sizes 0	Select File to Open Ciganize V New folder Organize V New folder OneDrive - University of Cambridge Sachel Howit - ISX data for Rachel Rachel Howit - ISX data for Rachel Rachel Howit - ISX data for Rachel Rachel Howit - ISX data for Rachel Documents Documents Documents Videos Videos Videos	X V O P Search 30_Test V V II O SD_MUN, Test, RO Umask
Number of epochs 0 # g, 2.0 = 50; 3.0 = 150. Batch size 0 # g, 2.0 = 12; 3.0 = 8	Load ROI-mask (if required). ROI mask file		(*.tiff) ~
TRAIN NETWORK	PREDICT LABELS		

- If desired, load a ROI-mask to limit the probability maps to the desired tissue region:
 (Leave blank if not required).

MATLAB App	- 🗆 ×		
Train a label-free segmentation model	Clearate label-free predictions Lad pretrained model information Pretrained model information Pretrained model information CLEAR	Select File to Open (× ð P Search 30, Test R • III @ 30, M.N. Test, RO L.mask
Number of epochs 0 g. 2-0 Batch size 0 g. 2-0 TRAIN NETWORK	0 = 50, 50 = 150 Load ROI-mask (if required) 0 = 12, 3.0 + 8 ROI mask file 30, MLN_Test, ROI PREDICT LABELS PREDICT LABELS	Videos Videos Vindous (C.) DATA (D:) File name: 30_MLN_Test_ROI_mask	✓ [*:iff) ✓ Open Cancel

- Click 'Predict Labels' to generate the label-free probability images

- Progress can be tracked in the Command Prompt console

- Figure 1 demonstrates probability images for each class (central z-plane for 3-D data)

- **Outputs** are saved to the Desktop ready for loading into the CellProfiler pipelines enabling cell feature extraction (described below).

~					
1				Figure 2	- 🗆 X
Recycle Bin		1 🖸 📄 = Standalone_software_biostud	lies	File	
💽 МАПАВ Арр	- 🗆 ×	File Home Share View			
Train a label-free segmentation model	Generate label-free predictions	🖈 🗎 📋 💑 Cut	📕 투 🗙 📑 📕	Central zplane: example netw	ork predictions
Load reflectance image:	Load pretrained model:	access Paste Paste shortcut	to t	LF-nuclei LF-actin	Background/other
Image file	MLN_Biostudies_27-May-2021-1355.mat	Clipboard	Organise		
Image x size 0	9	\leftarrow \rightarrow \checkmark \uparrow \blacksquare \rightarrow This PC \rightarrow DATA (D:)	> John > 2021 > Standalone_software_t		
image y star 0	Pretrained model information:	Poruments	A Name		
Image z nize 0	Number of epochs 150				ê basa
Jabel_free	Datch size 6	h Music	2 WINDOWS10 INST011 Tabel free		
ediction		E Pictures	3 Example pretrained models		
Nuclear label file		Videos	4_2D_CellProfiler_pipeline	Uncertainty	
Load cell outline labels	Load reflectance image:	L Windows (C:)	5_3D_CellProfiler_pipeline	and the second se	邂
Cell outline labels file	Image file 3D_MLN_Test_reflectance.tiff	DATA (D:)	ve		<u>8</u>
I nari barkrimuntiniher labele:	Image x size 1024	5 items 1 item selected			<u>R</u>
Background labels file	image y size 1024	LabelFreeStainPrediction			
Land POL mark // maniparty	Image z size 107	Starting label-free stain predict	ian		
		starting 3-D block-wise processin finished.	g of the reflectance data		
NOT Mask the					
				·	
	Load ROI-mask (if required):				
Number of epochs 0 8 g, 2-0 = 50; 3-0 = 150	ROI mask file 3D MLN Test ROI				
Batch size 0 e.g., 2-D = 12; 3-D = 8.					
and the second se					
TRAIN NETWORK	PREDICT LABELS				
		L			

Windows 10: Running the label-free cell segmentation deep learning scripts on an NVIDIA GPU using MATLAB R2021b and the Deep Learning Toolbox

Running these deep learning scripts in MATLAB requires:

- MATLAB (version R2019a or later)
- Deep Learning Toolbox
- Image Processing Toolbox
- Computer Vision Toolbox

Installation Steps: 1. Install MATLAB

Download MATLAB Simulink Str x +

https://ch.mathworks.com/downloads/web_downloads

- Login to your MathWorks account to access the downloads page at the link above.
- Download and run the installer for MATLAB R2021b.

	*	widthworks.					
	Dov	wnloads					
	FAQ	Installation and Licen	nsing Help				C Cc
	Selec	t Release	R 2021h				About R2021b
	~ R	2021b	120210				Latest Features
	R	2021a	Get MATLAB an	d Simulink Products			Release Notes
	R	2020b					System Requirements
				Download for V (196 MB) Includes R2021 (12 Not	Vindows		
				(
			Installation Instru	uctions			
			1. Download and	d launch the installer.			
			2. Sign in as jw2	020@cam.ac.uk.			
			In addition to in download produ	stalling MATLAB, you can also uct files for installation on an of	add products to your exist ffline machine.	ing installation or	
			Download details:	1			
			 IIIdudu hzuziu 	THERE AND LITTLE VALUE. VALUE. VALUEAT VAL	Rec01o1cf74o10obb00501d)		
orks Product Installer					×		
orks Product Installer		PRODUCTS	OPTIONS	Advanced Options CONFIRMATION	• - Dur Com	ring insta puter Vis	llation, install ion, Deep Lea
orks Product Installer R20 LICEN SING		PRODUCTS	OPTIONS	Advanced Options CONFIRMATION	• - Dur Com and I	ring insta puter Vis mage Pro	llation, install ion, Deep Lea ocessing Tool
erks Product Installer	210 DESTINATION ot products (recommend	PRODUCTS O	OPTIONS O e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc 	ing insta puter Vis mage Pro lecting th lucts' ste	llation, install ion, Deep Lea ocessing Tool nem at the p of the
orks Product Installer	DESTINATION C t products (recommend Select All	PRODUCTS O led products are	OPTIONS O e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proci insta 	ing insta puter Vis mage Pro lecting th lucts' ste llation.	llation, install ion, Deep Lea ocessing Tool nem at the p of the
orks Product Installer	215 DESTINATION e t products (recommend Select All Bioinformatics Toolbox	PRODUCTS O	OPTIONS O e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc insta 	ing insta puter Vis mage Pro lecting th lucts' ste llation.	llation, install ion, Deep Lea ocessing Tool nem at the p of the
Corres Product Installer Control Contro Control Control Control Control Cont	215 DESTINATION e t products (recommend Select All Bioinformatics Toolbox Communications Toolbox	PRODUCTS O	OPTIONS O e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc insta 	ing insta puter Vis mage Pro lecting th lucts' stej llation.	llation, install ion, Deep Lea ocessing Tool nem at the p of the
CICENSING	210 DESTINATION e t products (recommend Select All Bioinformatics Toolbox Communications Toolbox	PRODUCTS O led products are	options o e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc insta - Onc 	ring insta puter Vis mage Pro electing th lucts' step llation.	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is
orks Product Installer	DESTINATION DESTINATION C t products (recommend Select All Bioinformatics Toolbox Communications Toolbox Computer Vision Toolbox Control System Toolbox	PRODUCTS O	OPTIONS e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proci insta - Onc 	ing insta puter Vis mage Pro electing th lucts' step llation.	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is
orks Product Installer	210 DESTINATION C t products (recommend Select All Bioinformatics Toolbox Computer Vision Toolbox Computer Vision Toolbox Control System Toolbox Control System Toolbox	PRODUCTS O	options o e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proci insta - Onc comp 	ing insta puter Vis mage Pro lecting th lucts' step llation. ce the inst plete, you	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is can proceed
orks Product Installer	215 DESTINATION C t products (recommend Select All Bioinformatics Toolbox Computer Vision Toolbox Computer Vision Toolbox Control System Toolbox Curve Fitting Toolbox Data Acquisition Toolbox Data Acquisition Toolbox	PRODUCTS O	OPTIONS O e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proci insta - Onc comp 	ring insta puter Visi mage Pro electing th lucts' step llation. ce the inst plete, you	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is a can proceed
Corres Product Installer	215 DESTINATION C t products (recommend Select All Bioinformatics Toolbox Computer Vision Toolbox Computer Vision Toolbox Control System Toolbox Control System Toolbox Data Acquisition Toolbox Data Acquisition Toolbox Data Acquisition Toolbox	PRODUCTS O	OPTIONS O e preselected)	Advanced Options CONFIRMATION	× • Dur Comj and I by se 'Proc insta - Onc comp runn	ring insta puter Vis mage Pro electing th lucts' step llation. the the inst plete, you ing the so	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is can proceed cripts.
	215 DESTINATION	PRODUCTS O led products are	options o e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc insta - Onc comp runn 	ring insta puter Vis mage Pro electing th lucts' step llation. the inst olete, you ing the so	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is can proceed cripts.
crics Product Installer	210 DESTINATION C t products (recommend Bioinformatics Toolbox Communications Toolbox Computer Vision Toolbox Computer Vision Toolbox Control System Toolbox Curve Fitting Toolbox Data Acquisition Toolbox Data Acquisition Toolbox Data Securision Toolbox Data Securision Toolbox Data Biockset DDS Blockset	PRODUCTS O led products are	options o e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc insta - Onc comp runn 	ring insta puter Vis mage Pro electing th lucts' step llation. the inst plete, you ing the so	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is a can proceed cripts.
orks Product Installer	DESTINATION DESTINATION C t products (recommend Select All Bioinformatics Toolbox Computer Vision Toolbox Computer Vision Toolbox Control System Toolbox Curve Fitting Toolbox Data Acquisition Toolbox DataAcquisition Toolbox DataAcquisition Toolbox DataAcquisition Toolbox DataAcquisition Toolbox Deep Learning HDL Toolbox Deep Learning Toolbox	PRODUCTS	options e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proc insta - Onc comp runn 	ing insta puter Vis mage Pro electing th lucts' step llation. the inst olete, you ing the so	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is can proceed cripts.
orks Product Installer	DESTINATION DESTINATION C t products (recommend Select All Bioinformatics Toolbox Computer Vision Toolbox Control System Toolbox Control System Toolbox Control System Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DataAcset Toolbox DoS Blockset Deep Learning HDL Toolbox	PRODUCTS	options e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proci insta - Onc comp runn 	ing insta puter Vision lecting the lucts' step llation. the inst plete, you ing the so	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is can proceed cripts.
Corres Product Installer	DESTINATION DESTINATION C t products (recommend Select All Bioinformatics Toolbox Communications Toolbox Comtrol System Toolbox Control System Toolbox Control System Toolbox Control System Toolbox DataAsed Toolbox DataAsed Toolbox DataSed Toolbox DataSed Toolbox DataSed Toolbox DataSed Toolbox DataSed Toolbox Dos Biockset Deep Learning HDL Toolbox	PRODUCTS	options e preselected)	Advanced Options CONFIRMATION	 - Dur Comj and I by se 'Proci insta - Onc comp runn 	ing insta puter Vis mage Pro lecting th lucts' step llation. the inst olete, you ing the so	llation, install ion, Deep Lea ocessing Tool nem at the p of the tallation is can proceed cripts.

2. Training a 2-D UNET Model

-A screencast video is included with the BioStudies archive.

- Download and unzip the MATLAB BioStudies project archive at a suitable location on your computer.



- Inside the MATLAB_2D_UNET folder, there is a sequentially-organised workflow starting with raw data and ending with a CellProfiler pipeline that obtains cell features from the label-free cell segmentation:



- To train a 2-D UNET model, open MATLAB and set the working directory to the "2_TRAIN_Unet" directory. Open the "A_TRAIN_UNET" script.

MATLAB R2021b - academic use														
HOME	PLOTS		APPS											
New New New Open Script Live Script		G Find Files	Import Data	Save Workspace VA	B New Variable Image: the state of the st	Favorites	Analyze Code	Layout	 Preferences Set Path Parallel ENVIRONMENT 	Add-Ons	? Help	Community Request Support Learn MATLAB RESOURCES		
💠 🌩 🛅 🖾 🎘	•	D: •	John 🕨 2021 🕨	MATLA	B_biostudies	MATLAB_2D_UNET +	2_TRAIN_U	net 🕨						
Command Window	N										\odot	Worksp	ace	
fx >>												Name	A	Value

- Update the path to the training data (located in the "1_ImageData" folder).

- Update the path to tissue ROI mask which isolates the lymphoid follicle from the surrounding tissue (located in the "1_ImageData" folder).

- Update the path to the pixel classification labels (located in the "1_ImageData" folder)

D:	Jonn	2021/MAILAB_biostudies/M	IAILAB_2D_UNET\2_TRAIN_U	Inet\A_IKAIN_UNEI.m									
E	DITOR	PUBLISH	VIEW										
New	Open	Gave FILE Compare ▼	Go To WAVIGATE	Refactor CODE	Profiler Profiler Analyze ANALYZE	Run Section SECTI	ction Break in and Advance in to End ION	Run R	Step Step	top			
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18	FILE NAVIGATE CODE ANALYZE SECTION RUN clear close all close close close close 2 Clear close all close close close 3 X Read in the reflectance data for training fraining_data = imread('D:\John\2021\MATLAB_biostudies\MATLAB_DUNET\1_ImageData\TRAIN_IMAGE.tif',3) ; % channel 3 is reflectance 3 X Read in the mask to isolated just the lymphoid tissue mask = imread('D:\John\2021\MATLAB_biostudies\MATLAB_DUNET\1_ImageData\TRAIN_IMAGE.tif',4) ; 3 X Read in the mask to isolated just the lymphoid tissue mask = imread('D:\John\2021\MATLAB_biostudies\MATLAB_DUNET\1_ImageData\TRAIN_IMAGE.tif',4) ; 4 X Rescale the training data rraining_data = Training_data .* mask ; 5 X Rescale the training data in the interval [0 1] 6 X Read in the pixel-class labels created from the nuclei and actin staining												
19 20 21 22 23 24 25 26 27 28 29		<pre>% Visually inspec figure(1) subplot(1,2,1) imshow(Training_d title('Reflectanc subplot(1,2,2) imshow(Training_l title('Pixel class)</pre>	<pre>imread('D:\John\28 t the data and labe ata, []) e data') abels, []) sification labels')</pre>	21 (MATLAB_blostud: ls	ies (MATLAB_2	D_UNET(1_1	mageData\IRAI	N_LABEI	LS.png);			

- On Line 46, update the path to the pixel classification labels (located in the "1_ImageData" folder) used by the "pixelLabelDatastore" function.

D:\John\2021\MATLAB_biostudies\MATLAB_2D_UNET\2_TRAIN_Unet\A_TRAIN_UNET.m EDITOR fx % % % Section Break ¢ 2 🕂 🛅 🔚 😥 Compare 🕶 D Profiler Refactor Run and Advance Q Find -Analyze Run Section Run to End New Open Save 🚔 Print 🕶 Go To Run Step Stop Bookmark 💌 -fi -NAVIGATE ANALYZE SECTION FILE CODE RUN %% Create 'Datastores' for the reflectance information and matching pixel-class labels 36 % Read saved mat file containing reflectance info into an image datastore |
% Directions for processing the mat file-type are in the accompanying 'matReader' function
reflectance_ds = imageDatastore([pwd, '/mat_training_data/'],'FileExtensions','.mat','ReadFcn',@matReader); 37 38 39 40 41 % Read pixel labels into a pixel label datastore 42 PixelClassNames = ["LF_actin","LF_nuclei","BackgroundOther"]; 43 PixelLabel_ds = pixelLabelDatastore('D:\John\2021\MATLAB_biostudies\MATLAB_2D_UNET\1_ImageData\TRAIN_LABELS.png', PixelClassNames,Label_Ids); 44 45 46

- At the bottom of the script on Line 195, update the path to the saved_network_directory. - This should be folder 3 of the workflow ("3_Saved_models").



- Run the script by clicking the green arrow at the top.

- Model training takes several hours (~ 4h on a NVIDIA 1080 Ti GPU)

- The newly-trained model will be saved in the "3_Saved_models" directory with a new time/date stamp.

D:\John\2021\MATLAB_biostudies\MATLAB_2D_UNET\2_TRAIN_Unet\A_TRAIN_UNET.m

ED	ITOR		PUBLISH	VIEW									
New	Open	Save	🔄 Compare 👻 🚍 Print 💌	Go To	♀ ↔ Q Find ▼ Bookmark ▼	Refactor	% ‰ %7 ▶ +2 ₽ •A ▼	😓 Profiler 🚽 Analyze	Run Section	E Section Break Run and Advance Run to End	Run	G Step	Stop
		FILE			NAVIGATE	C	ODE	ANALYZE		SECTION		RUN	

3. Testing a pretrained 2-D model using unseen data

-A screencast video is included with the BioStudies archive.

- Change the MATLAB working directory to the "4_Test_Unet" directory. Open the "A_TEST_UNET" script.



- Change the path to the unseen test image (located in the "1_ImageData" folder).

- Specify the ROI mask which identifies the lymphoid tissue (located in the "1_ImageData" folder).

- Specify which pretrained network to use (pretrained networks are located in the "3_Saved_models" folder.

D:\John\2021\MATLAB_biostudies\MATLAB_2D_UNET\4_TEST_Unet\A_TEST_UNET.m

E	DITOR		PUBLISH	VIEW					
New	Open	Save	E Compare ▼ ➡ Print ▼	Go To AVIGATE	Refactor	Profiler	Run Section Break	Run Step	Stop
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18		FILE clear close clc % Load Mask · Mask · % Resc TEST_I % Masl TEST_I % Masl	all <u>d the reflect</u> DATA = imread d the mask fo = imread('D:\ = double(imbi <u>cale the test</u> DATA_rescaled ATA_rescaled ATA_rescaled	<pre>naviGATE ance data from the ('D:\John\2021\MATLAB_bi narize(Mask)); data [0 1] data [0 1] data d data = TEST_DATA_rescal d natwork</pre>	unseen test image AB_biostudies\MAT ue region ostudies\MATLAB_2 y(TEST_DATA, [0 4 ed .* Mask ;	ANALYZE	SECTION T\1_ImageData\TEST_IMAG mageData\TEST_IMAGE.ti	RUN	3 is reflectance channel
19 20	l	load('D:\John\2021	\MATLAB_biostudies	MATLAB_2D_UNET\3_	Saved_mode	ls\TEST_e5023-May-2021	-23-28-02.mat');

- Click the "Run" button to process the unseen reflectance data with the selected pretrained network:



- Probability maps for the LF-nuclei, LF-actin and Background/Other classes will be saved in the working directory named ready for input into the CellProfiler pipeline (filenames Ch_5_lm_001, Ch_6_lm_001, Ch_7_lm_001, respectively).

\leftarrow \rightarrow \checkmark \uparrow MATLAB_biostudies \rightarrow	MATI	AB_2D_UNET > 4_TEST_Unet	م ت ۲	Search 4_TEST_Unet
Documents	^	Name	Date	Туре
🕹 Downloads		A_TEST_UNET	30/11/2021 14:08	MATLAB Code
b Music		Ch_5_lm_001	30/11/2021 14:33	TIFF File
Pictures		Ch_6_Im_001	30/11/2021 14:33	TIFF File
Videos		Ch_7_Im_001	30/11/2021 14:33	TIFF File

4. Training a 3-D UNET Model

-A screencast video is included with the BioStudies archive.

- Inside the MATLAB_3D_UNET folder, there is a sequentially-organised workflow starting with raw data and ending with a CellProfiler pipeline that obtains 3-D cell features from the label-free cell segmentation:

2021	> MATLAB_biostudies > MATLAB_3D_UNET	ٽ ~	
^	Name	Date modified	Туре
	🔒 1_ImageData	30/11/2021 14:08	File folder
	2_TRAIN_Unet	30/11/2021 14:12	File folder
	3_Saved_models	30/11/2021 14:10	File folder
	4_TEST_Unet	30/11/2021 14:12	File folder
	5_CellProfiler_cell_measurements	30/11/2021 14:10	File folder

- To train a 3-D UNET model, open MATLAB and set the working directory to the "2_TRAIN_Unet" directory. Open the "A_TRAIN_3D_UNET" script.



- Specify the path to the bioformats library. This is included inside the "2_TRAIN_Unet" folder ("bfmatlab").

- Update the path to the training data directory (located in the "1_ImageData/TRAIN/DATA" folder).

- Specify the channel number in the training data that contains the reflectance information (here, '3').

- Specify the location of the pixel classification training labels (located at "1_ImageData/TRAIN/LABELS" folder.



- Specify the location to save the model when training completes. This should be the "3_Saved_models" directory.

227	%% Train and save the network			
228	close all			
229				
230	% Specify location to save the network			
231	<pre>saved_network_directory = 'D:\John\2021\MATLAB_biostudies\MATLAB_3D_UNET\3_</pre>	Saved_m	odels\';	
232				
233	% Train the network			
234	<pre>modelDateTime = datestr(now,'dd-mmm-yyyy-HHMM');</pre>			
235	<pre>[net,info] = trainNetwork(Training_ds,lgraph,options);</pre>			
236				
237	% Timestamp and save the network after training			
238	<pre>save([saved_network_directory,'MLN_Biostudies_',modelDateTime,'.mat'],'net'</pre>	,'optio	ns','inf	o');
239				

- Click "Run" to commence model training.

- This takes considerable time (approximately 6h on a NVIDIA 1080Ti GPU card).



6. Testing a pretrained 3-D model using unseen data

-A screencast video is included with the BioStudies archive.

- Change the MATLAB working directory to the "4_Test_Unet" directory. Open the "A_TEST_3D_UNET" script.



- Specify the path to the Bioformats library. (This is included inside the "4_TEST_Unet" folder ("bfmatlab").

- Specify the location of the unseen test image-data. This is located inside the "1_ImageData" folder at "1_ImageData/TEST/".



- Specify a pretrained 3-D Unet model from the "3_Saved_models" directory.

40	%%
41	% load the trained 3D Unet network
42	load('D:\John\2021\MATLAB biostudies\MATLAB 3D UNET\3 Saved models\MLN Biostudies 27-May-2021-1355.mat);
43	

- Click "Run" to process the unseen reflectance data with the 3-D Unet model.

🗐 D:	D:\John\2021\MATLAB_biostudies\MATLAB_3D_UNET\4_TEST_Unet\A_TEST_3D_UNET.m												
E	DITOR		PUBLISH	VIEW	1							6 4	1
New	Open	Save	🔄 Compare 👻 🚍 Print 👻	Go To		Refactor	% ‰ %7 ∑ +2 ⊡ Fa ←	🕤 Profiler 📝 Analyze	Run Section	 Section Break Run and Advance Run to End 	Run	G Step	Stop
4.20		FILE			NAVIGATE	0	ODE	ANALYZE		SECTION		RUN	

- Probability maps for the LF-nuclei, LF-actin and Background/Other classes will be saved as multipage .TIFF files in the working directory named ready for input into the CellProfiler pipeline (filenames Ch_3_lm_001, Ch_4_lm_001, Ch_5_lm_001, respectively).



```
% MATLAB SCRIPT: TRAIN 2D UNET
% All code, image-data and screen-cast tutorial videos available for download at
% the Biostudies database under accession number S-BSST742
clear
close all
clc
% Read in the reflectance data for training
Training data =
imread('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\1 ImageData\TRAIN IMAGE.ti
f',3) ; % channel 3 is reflectance
% Read in the mask to isolated just the lymphoid tissue
mask =
imread('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\1 ImageData\TRAIN IMAGE.ti
f',4);
mask = uint16(imbinarize(mask));
% Apply the mask to the training data
Training data = Training data .* mask ;
% Rescale the training data in the interval [0 1]
Training data = double(1*mat2gray(Training data, [0 4095])) ;
% Read in the pixel-class labels created from the nuclei and actin staining
Training labels =
imread('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\1 ImageData\TRAIN LABELS.p
ng');
% Visually inspect the data and labels
figure(1)
subplot(1,2,1)
imshow(Training data, [])
title('Reflectance data')
subplot(1,2,2)
imshow(Training labels, [])
title('Pixel classification labels')
%% Once happy with labels and data, commit the reflectance information data to
sub-directory in .mat format
if ~exist('mat training data', 'dir')
   mkdir('mat training data');
end
save([pwd,'/mat training data/','Training data.mat'],'Training data');
%% Create 'Datastores' for the reflectance information and matching pixel-class
labels
% Read saved mat file containing reflectance info into an image datastore
% Directions for processing the mat file-type are in the accompanying
'matReader' function
reflectance ds = imageDatastore([pwd,
'/mat training data/'],'FileExtensions','.mat','ReadFcn',@matReader) ;
% Read pixel labels into a pixel label datastore
PixelClassNames = ["LF_actin","LF_nuclei","BackgroundOther"];
Label_Ids = 1:3; % these represent the pixel values in the labels png file
PixelLabel ds =
pixelLabelDatastore('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\1 ImageData\T
RAIN LABELS.png',...
```

PixelClassNames,Label_Ids);

```
%% Define augmentation options
augmenter = imageDataAugmenter(...
                                             %%% description and defaults %%%
    'FillValue',0,...
                                      % define out-of-bounds points when resampling
0
    'RandXReflection', true,...
                                      % Random reflection in the left-right
direction false
    'RandYReflection',true,...
                                      % Random reflection in the top-bottom
direction false
    'RandRotation',[0 360],...
                                      % Range of rotation, in degrees [0 0]
    'RandScale',[1 1],...
                                      % Range of uniform (isotropic) scaling [1 1]
    'RandXScale',[1 1],...
'RandYScale',[1 1],...
                                      % Range of horizontal scaling [1 1]
                                      % Range of vertical scaling [1 1]
                                      % Range of horizontal shear [0 0]
    'RandXShear',[0 0],...
    'RandYShear',[0 0],...
                                      % Range of vertical shear [0 0]
    'RandXTranslation',[0 0],...
                                    % Range of horizontal translation [0 0]
    'RandYTranslation',[0 0]...
                                     % Range of vertical translation [0 0]
    );
%% Create a training datastore comprising matching patches (256x256 pixels) of
image-data and training labels
training ds =
randomPatchExtractionDatastore(reflectance ds,PixelLabel ds,[256,256],...
     'PatchesPerImage',12000,'DataAugmentation',augmenter); % 743 draws == 1
epoch
%% CREATE THE UNET
% input layer 256x256x1
% encoder depth 4
% 64 filters at the level of the first encoder
lgraph = layerGraph();
tempLayers = [
    imageInputLayer([256 256 1], "Name", "ImageInputLayer")
    convolution2dLayer([3 3],64, "Name", "Encoder-Stage-1-Conv-
1","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Encoder-Stage-1-ReLU-1")
    convolution2dLayer([3 3],64,"Name","Encoder-Stage-1-Conv-
2","Padding","same","WeightsInitializer","he")
    reluLayer("Name","Encoder-Stage-1-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([2 2],"Name","Encoder-Stage-1-MaxPool","Stride",[2 2])
    convolution2dLayer([3 3],128,"Name","Encoder-Stage-2-Conv-
1", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name","Encoder-Stage-2-ReLU-1")
convolution2dLayer([3 3],128,"Name","Encoder-Stage-2-Conv-
2","Padding","same","WeightsInitializer","he")
reluLayer("Name","Encoder-Stage-2-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([2 2],"Name","Encoder-Stage-2-MaxPool","Stride",[2 2])
    convolution2dLayer([3 3],256,"Name","Encoder-Stage-3-Conv-
1","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Encoder-Stage-3-ReLU-1")
    convolution2dLayer([3 3],256,"Name","Encoder-Stage-3-Conv-
2","Padding","same","WeightsInitializer","he")
```

```
reluLayer("Name", "Encoder-Stage-3-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling2dLayer([2 2], "Name", "Encoder-Stage-3-MaxPool", "Stride", [2 2])
convolution2dLayer([3 3],512, "Name", "Encoder-Stage-4-Conv-
1", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name", "Encoder-Stage-4-ReLU-1")
    convolution2dLayer([3 3],512,"Name","Encoder-Stage-4-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name", "Encoder-Stage-4-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    dropoutLayer(0.5, "Name", "Encoder-Stage-4-DropOut")
    maxPooling2dLayer([2 2], "Name", "Encoder-Stage-4-MaxPool", "Stride", [2 2])
convolution2dLayer([3 3],1024,"Name","Bridge-Conv-
1","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Bridge-ReLU-1")
convolution2dLayer([3 3],1024,"Name","Bridge-Conv-
2","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Bridge-ReLU-2")
    dropoutLayer(0.5, "Name", "Bridge-DropOut")
transposedConv2dLayer([2 2],512,"Name","Decoder-Stage-1-
UpConv","BiasLearnRateFactor",2,"Stride",[2 2],"WeightsInitializer","he")
    reluLayer("Name", "Decoder-Stage-1-UpReLU")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(2, "Name", "Decoder-Stage-1-DepthConcatenation")
    convolution2dLayer([3 3],512,"Name","Decoder-Stage-1-Conv-
1","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Decoder-Stage-1-ReLU-1")
convolution2dLayer([3 3],512, "Name", "Decoder-Stage-1-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name", "Decoder-Stage-1-ReLU-2")
    transposedConv2dLayer([2 2],256,"Name","Decoder-Stage-2-
UpConv", "BiasLearnRateFactor", 2, "Stride", [2 2], "WeightsInitializer", "he")
    reluLayer("Name", "Decoder-Stage-2-UpReLU")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(2, "Name", "Decoder-Stage-2-DepthConcatenation")
    convolution2dLayer([3 3],256,"Name","Decoder-Stage-2-Conv-
1","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Decoder-Stage-2-ReLU-1")
convolution2dLayer([3 3],256, "Name", "Decoder-Stage-2-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name", "Decoder-Stage-2-ReLU-2")
transposedConv2dLayer([2 2],128,"Name","Decoder-Stage-3-
UpConv","BiasLearnRateFactor",2,"Stride",[2 2],"WeightsInitializer","he")
    reluLayer("Name", "Decoder-Stage-3-UpReLU")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(2, "Name", "Decoder-Stage-3-DepthConcatenation")
    convolution2dLayer([3 3],128,"Name","Decoder-Stage-3-Conv-
1", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name", "Decoder-Stage-3-ReLU-1")
```

```
convolution2dLayer([3 3],128,"Name","Decoder-Stage-3-Conv-
2","Padding","same","WeightsInitializer","he")
    reluLayer("Name", "Decoder-Stage-3-ReLU-2")
    transposedConv2dLayer([2 2],64,"Name","Decoder-Stage-4-
UpConv","BiasLearnRateFactor",2,"Stride",[2 2],"WeightsInitializer","he")
    reluLayer("Name", "Decoder-Stage-4-UpReLU")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    depthConcatenationLayer(2, "Name", "Decoder-Stage-4-DepthConcatenation")
    convolution2dLayer([3 3],64,"Name","Decoder-Stage-4-Conv-
1", "Padding", "same", "WeightsInitializer", "he")
    reluLayer("Name", "Decoder-Stage-4-ReLU-1")
    convolution2dLayer([3 3],64, "Name", "Decoder-Stage-4-Conv-
2","Padding","same","WeightsInitializer","he")
    reluLayer("Name","Decoder-Stage-4-ReLU-2")
    convolution2dLayer([1 1],3,"Name","Final-
ConvolutionLayer", "Padding", "same", "WeightsInitializer", "he")
    softmaxLayer("Name", "Softmax-Layer")
    pixelClassificationLayer("Name", "Segmentation-Layer")];
lgraph = addLayers(lgraph,tempLayers);
clear tempLayers;
% encoder / decoder connections
lgraph = connectLayers(lgraph,"Encoder-Stage-1-ReLU-2","Encoder-Stage-1-
MaxPool");
lgraph = connectLayers(lgraph,"Encoder-Stage-1-ReLU-2","Decoder-Stage-4-
DepthConcatenation/in2");
lgraph = connectLayers(lgraph, "Encoder-Stage-2-ReLU-2", "Encoder-Stage-2-
MaxPool");
lgraph = connectLayers(lgraph, "Encoder-Stage-2-ReLU-2", "Decoder-Stage-3-
DepthConcatenation/in2");
lgraph = connectLayers(lgraph, "Encoder-Stage-3-ReLU-2", "Encoder-Stage-3-
MaxPool");
lgraph = connectLayers(lgraph, "Encoder-Stage-3-ReLU-2", "Decoder-Stage-2-
DepthConcatenation/in2");
lgraph = connectLayers(lgraph,"Encoder-Stage-4-ReLU-2","Encoder-Stage-4-
DropOut");
lgraph = connectLayers(lgraph,"Encoder-Stage-4-ReLU-2","Decoder-Stage-1-
DepthConcatenation/in2");
lgraph = connectLayers(lgraph, "Decoder-Stage-1-UpReLU", "Decoder-Stage-1-
DepthConcatenation/in1");
lgraph = connectLayers(lgraph, "Decoder-Stage-2-UpReLU", "Decoder-Stage-2-
DepthConcatenation/in1");
lgraph = connectLayers(lgraph, "Decoder-Stage-3-UpReLU", "Decoder-Stage-3-
DepthConcatenation/in1");
lgraph = connectLayers(lgraph, "Decoder-Stage-4-UpReLU", "Decoder-Stage-4-
DepthConcatenation/in1");
% analyzeNetwork(lgraph) % comment in to check structure and errors
plot(lgraph) % comment in to show network structure
%% Specify training options
options = trainingOptions('sgdm',...
    'InitialLearnRate',0.05, ...
    'Momentum',0.9,...
    'L2Regularization',0.0001,...
    'MaxEpochs',50,...
    'MiniBatchSize',12,...
```

```
'LearnRateSchedule', 'piecewise',...
    'Shuffle', 'every-epoch',...
    'GradientThresholdMethod', 'l2norm',...
    'GradientThreshold',0.05, ...
    'Plots', 'training-progress', ...
    'VerboseFrequency',20,...
    'ExecutionEnvironment', 'auto');
%% Train and save the network
close all
% Specify location to save the network
saved network directory =
'D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\3 Saved models\';
% Train the network
modelDateTime = datestr(now, 'dd-mmm-yyyy-HH-MM-SS');
[net,info] = trainNetwork(training ds,lgraph,options);
% Timestamp and save the network after training
save([saved network directory,'PeyersPatchBiostudies ',modelDateTime,'.mat'],...
    'net','options','augmenter','info');
```



2-D Unet architecture schematic. The network uses an input layer for the reflectance data of 256x256x1 (x, y, channels). The best performing three-class Unet architecture uses an encoder depth of 4 with 64 filters at the level of the first encoder (shown, **Figure S2**). The network uses complete up-convolutional expansion to yield outputted probability maps that are identically sized to the input layer.

```
% MATLAB SCRIPT: TEST 2D UNET
% All code, image-data and screen-cast tutorial videos available for download at
% the Biostudies database under accession number S-BSST742
clear
close all
clc
% Load the reflectance data from the unseen test image
TEST DATA =
imread('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\1 ImageData\TEST IMAGE.tif
',3); % 3 is reflectance channel
% Load the mask for the lymphoid tissue region
Mask =
imread('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\1 ImageData\TEST IMAGE.tif
',4);
Mask = double(imbinarize(Mask)) ;
% Rescale the test data [0 1]
TEST DATA rescaled = double(1*mat2gray(TEST DATA, [0 4095])) ;
% Mask the rescaled data
TEST DATA rescaled = TEST DATA rescaled .* Mask ;
% Load a pretrained network
load('D:\John\2021\MATLAB biostudies\MATLAB 2D UNET\3 Saved models\Rescaled0-
1 2DUNET S7-RL-SA enc-4 filt-64 e50.mat');
%% Patch the reflectance data through the network
****
% Set the patch sizes to be passed to the net
patchSize = [2048 2048]; % decrease if insufficient GPU memory eg., [1024 1024]
% Segment blockwise then reassemble in full
% Define image dimensions
[height, width, nChannel] = size(TEST DATA rescaled);
patch = zeros([patchSize, nChannel], 'like', TEST_DATA_rescaled);
% Pad image to have dimensions as multiples of patchSize
padSize(1) = patchSize(1) - mod(height, patchSize(1));
padSize(2) = patchSize(2) - mod(width, patchSize(2));
im_pad = padarray (TEST_DATA_rescaled, padSize, 0, 'post');
[height_pad, width_pad, nChannel_pad] = size(im_pad);
% Preallocate some matrices to receive the network outputs
out Uncertainty Scores = zeros([size(im pad,1), size(im pad,2)], 'double');
out Pmap_cat1 = out Uncertainty Scores;
out Pmap cat2 = out Uncertainty Scores;
out_Pmap_cat3 = out_Uncertainty Scores;
% Loop through blocks of 'patchSize'
    for loop = 1:patchSize(1):height pad
        for j =1:patchSize(2):width pad
           for p = 1:nChannel
                patch(:,:,p) = squeeze( im_pad( loop:loop+patchSize(1)-1,...
                                               j:j+patchSize(2)-1,p));
           end
            % deploy net
```
```
[patch_seg, Scores, allScores] = semanticseg(patch, net,
'OutputType', 'double',...
                'ExecutionEnvironment', 'auto');
            % catch what comes out
            out Uncertainty Scores(loop:loop+patchSize(1)-1, j:j+patchSize(2)-1)
= Scores;
            out Pmap cat1(loop:loop+patchSize(1)-1, j:j+patchSize(2)-1) =
allScores(:,:,1);
            out_Pmap_cat2(loop:loop+patchSize(1)-1, j:j+patchSize(2)-1) =
allScores(:,:,2);
            out Pmap cat3(loop:loop+patchSize(1)-1, j:j+patchSize(2)-1) =
allScores(:,:,3);
        end
    end
% Remove padding from the network outputs
out Uncertainty Scores = out Uncertainty Scores(1:height, 1:width);
out_Pmap_cat1 = out_Pmap_cat1(1:height, 1:width);
out_Pmap_cat2 = out_Pmap_cat2(1:height, 1:width);
out_Pmap_cat3 = out_Pmap_cat3(1:height, 1:width);
%% visualise network outputs
figure(2)
clf(figure(2))
ax1 = subplot(2,3,1);
imshow(out Pmap_cat1,[])
title('LF-actin')
ax2 = subplot(2,3,2);
imshow(out_Pmap_cat2,[])
title('LF-nuclei')
ax3 = subplot(2,3,3);
imshow(out_Pmap_cat3,[])
title('Background/other')
ax4 = subplot(2,3,5);
imshow(out Uncertainty Scores,[])
title('Uncertainty')
linkaxes([ax1 ax2 ax3 ax4], 'xy')
%% Map to 16-bit and save for loading into CellProfiler
ui16_PMAP_cat1_xy = uint16(65535*mat2gray(out_Pmap_cat1, [0 1])) ;
imwrite(ui16_PMAP_cat1_xy, ['Ch_6_Im_001','.tiff']) ;
uil6 PMAP cat2 xy = uint16(65535*mat2gray(out_Pmap_cat2, [0 1])) ;
imwrite(ui16 PMAP cat2 xy, ['Ch 5 Im 001','.tiff']) ;
ui16 PMAP_cat3_xy = uint16(65535*mat2gray(out_Pmap_cat3, [0 1])) ;
imwrite(uil6 PMAP cat3_xy, ['Ch_7_Im_001','.tiff']) ;
```

```
% MATLAB SCRIPT: Train 3D UNET
% All code, image-data and screen-cast tutorial videos available for download at
% the Biostudies database under accession number S-BSST742
clear
close all
clc
% Specify path to bioformats library
addpath('D:\John\2021\MATLAB biostudies\MATLAB 3D UNET\2 TRAIN Unet\bfmatlab\')
% Specify directory containing training data
Training data directory =
'D:\John\2021\MATLAB_biostudies\MATLAB 3D UNET\1 ImageData\TRAIN\DATA\';
% Specify channel number in training data that contains reflectance information
RL training directory = 3;
% Specify directory containing training labels
Training labels path =
'D:\John\2021\MATLAB_biostudies\MATLAB_3D_UNET\1_ImageData\TRAIN\LABELS\';
8% COMMIT TRAINING DATA TO DIRECTORY rescaled [0 1] IN MAT FORMAT
if ~exist('mat training data', 'dir')
   mkdir('mat_training_data');
end
for loop = 1:2
   counter = sprintf('%03d',loop) ;
    % find metadata describing file
    reader = bfGetReader([Training data directory, 'TRAIN ',counter,'.tif']);
    omeMeta = reader.getMetadataStore();
    number of channels = omeMeta.getChannelCount(0);
    stackSizeZ = omeMeta.getPixelsSizeZ(0).getValue(); % number of Z slices
    % Load reflectance information
    for zplane = 1:stackSizeZ
        iPlane = reader.getIndex(zplane -1, RL_training_directory -1, 0) + 1; %
because zplanes and channels are numbered from zero
        channel zimage{zplane} = bfGetPlane(reader, iPlane);
    end
    IM DATA = cat(3,channel zimage{:}) ;
    IM DATA = double(1*mat2gray(IM DATA, [0 65535])) ;
    save([pwd,'/mat_training_data/','TRAIN_DATA_',counter,'.mat'], 'IM_DATA') ;
end
%% COMMIT TRAINING LABELS TO DIRECTORY IN MAT FORMAT
if ~exist('mat_training_labels', 'dir')
   mkdir('mat_training_labels');
end
for loop = 1:2
    counter = sprintf('%03d',loop) ;
        % load label information
        for zplane = 1:stackSizeZ
            iplane =
imread([Training_labels_path,'LABELS_',counter,'.tif'],zplane);
```

```
label zimage{zplane} = iplane ;
        end
        IM LABELS = cat(3,label zimage{:});
        IM LABELS = double(IM LABELS) ;
        save([pwd,'/mat_training_labels/','TRAIN_LABELS_',counter,'.mat'],
'IM LABELS') ;
end
% Once data is prepared for training, clear workspace
clear ;
%% Create 'Datastores' for the reflectance information and matching pixel-class
labels
% Read saved mat file containing reflectance info into an image datastore
% Directions for processing the mat file-type are in the accompanying 'matRead'
function
data location = [pwd, '\mat training data\'];
volReader = @(x) JWmatRead(x);
reflectance ds =
imageDatastore(data location, 'FileExtensions', '.mat', 'ReadFcn', volReader);
% Read pixel labels into a pixel label datastore
label location = [pwd, '\mat training labels\'];
labelReader = @(x) JWLabelRead(x);
classNames = ["LFNuclei", "LFActin", "BackgroundOther"];
pixelLabelID = 1:3; % these represent the pixel values in the labels file
PixelLabel_ds = pixelLabelDatastore(label_location,classNames,pixelLabelID,...
    'FileExtensions', '.mat', 'ReadFcn', labelReader);
%% Set up patch extraction from reflectance datastore
patchSize = [64 64 32];
patchPerImage = 375;
MiniBatchSize = 8; % set the batch size
reflectance patches ds =
randomPatchExtractionDatastore(reflectance ds, PixelLabel ds, patchSize, 'PatchesPe
rImage',patchPerImage);
reflectance patches ds.MiniBatchSize = MiniBatchSize;
% Augment the patches using 'augment3dPatch' function
Training ds = transform(reflectance patches ds,@JWaugment3dPatch);
%% CREATE THE 3D UNET
% input layer 64x64x32x1
% encoder depth 4
% 32 filters at the level of the first encoder
lgraph = layerGraph();
tempLayers = [
    image3dInputLayer([64 64 32 1], "Name", "ImageInputLayer")
    convolution3dLayer([3 3 3],32,"Name","Encoder-Stage-1-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Encoder-Stage-1-BN-1")
    reluLayer("Name", "Encoder-Stage-1-ReLU-1")
convolution3dLayer([3 3 3],64,"Name","Encoder-Stage-1-Conv-
2","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Encoder-Stage-1-BN-2")
    reluLayer("Name", "Encoder-Stage-1-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
```

```
tempLayers = [
    maxPooling3dLayer([2 2 2], "Name", "Encoder-Stage-1-MaxPool", "Stride", [2 2 2])
convolution3dLayer([3 3 3],64,"Name","Encoder-Stage-2-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Encoder-Stage-2-BN-1")
    reluLayer("Name", "Encoder-Stage-2-ReLU-1")
    convolution3dLayer([3 3 3],128,"Name","Encoder-Stage-2-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    batchNormalizationLayer("Name", "Encoder-Stage-2-BN-2")
    reluLayer("Name", "Encoder-Stage-2-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling3dLayer([2 2 2], "Name", "Encoder-Stage-2-MaxPool", "Stride", [2 2 2])
convolution3dLayer([3 3 3],128,"Name","Encoder-Stage-3-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Encoder-Stage-3-BN-1")
    reluLayer("Name", "Encoder-Stage-3-ReLU-1")
    convolution3dLayer([3 3 3],256,"Name","Encoder-Stage-3-Conv-
2","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Encoder-Stage-3-BN-2")
    reluLayer("Name", "Encoder-Stage-3-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling3dLayer([2 2 2], "Name", "Encoder-Stage-3-MaxPool", "Stride", [2 2 2])
    convolution3dLayer([3 3 3],256,"Name","Encoder-Stage-4-Conv-
1", "Padding", "same", "WeightsInitializer", "he")
    batchNormalizationLayer("Name", "Encoder-Stage-4-BN-1")
    reluLayer("Name", "Encoder-Stage-4-ReLU-1")
    convolution3dLayer([3 3 3],512,"Name","Encoder-Stage-4-Conv-
2","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Encoder-Stage-4-BN-2")
    reluLayer("Name", "Encoder-Stage-4-ReLU-2")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    maxPooling3dLayer([2 2 2], "Name", "Encoder-Stage-4-MaxPool", "Stride", [2 2 2])
convolution3dLayer([3 3 3],512,"Name","Bridge-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Bridge-BN-1")
    reluLayer("Name","Bridge-ReLU-1")
    convolution3dLayer([3 3 3],1024,"Name","Bridge-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    batchNormalizationLayer("Name", "Bridge-BN-2")
    reluLayer("Name", "Bridge-ReLU-2")
    transposedConv3dLayer([2 2 2],1024,"Name","Decoder-Stage-1-
UpConv", "BiasLearnRateFactor", 2, "Stride", [2 2 2], "WeightsInitializer", "he")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    concatenationLayer(4,2,"Name","Decoder-Stage-1-Concatenation")
    convolution3dLayer([3 3 3],512,"Name","Decoder-Stage-1-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Decoder-Stage-1-BN-1")
    reluLayer("Name", "Decoder-Stage-1-ReLU-1")
    convolution3dLayer([3 3 3],512,"Name","Decoder-Stage-1-Conv-
2","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Decoder-Stage-1-BN-2")
```

```
reluLayer("Name", "Decoder-Stage-1-ReLU-2")
    transposedConv3dLayer([2 2 2],512,"Name","Decoder-Stage-2-
UpConv", "BiasLearnRateFactor", 2, "Stride", [2 2 2], "WeightsInitializer", "he")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    concatenationLayer(4,2,"Name","Decoder-Stage-2-Concatenation")
convolution3dLayer([3 3 3],256,"Name","Decoder-Stage-2-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Decoder-Stage-2-BN-1")
    reluLayer("Name", "Decoder-Stage-2-ReLU-1")
    convolution3dLayer([3 3 3],256,"Name","Decoder-Stage-2-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    batchNormalizationLayer("Name", "Decoder-Stage-2-BN-2")
    reluLayer("Name", "Decoder-Stage-2-ReLU-2")
    transposedConv3dLayer([2 2 2],256,"Name","Decoder-Stage-3-
UpConv","BiasLearnRateFactor",2,"Stride",[2 2 2],"WeightsInitializer","he")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    concatenationLayer(4,2,"Name","Decoder-Stage-3-Concatenation")
    convolution3dLayer([3 3 3],128,"Name","Decoder-Stage-3-Conv-
1", "Padding", "same", "WeightsInitializer", "he")
    batchNormalizationLayer("Name", "Decoder-Stage-3-BN-1")
    reluLayer("Name", "Decoder-Stage-3-ReLU-1")
convolution3dLayer([3 3 3],128,"Name","Decoder-Stage-3-Conv-
2","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Decoder-Stage-3-BN-2")
    reluLayer("Name", "Decoder-Stage-3-ReLU-2")
    transposedConv3dLayer([2 2 2],128, "Name", "Decoder-Stage-4-
UpConv", "BiasLearnRateFactor", 2, "Stride", [2 2 2], "WeightsInitializer", "he")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
    concatenationLayer(4,2,"Name","Decoder-Stage-4-Concatenation")
convolution3dLayer([3 3 3],64,"Name","Decoder-Stage-4-Conv-
1","Padding","same","WeightsInitializer","he")
    batchNormalizationLayer("Name", "Decoder-Stage-4-BN-1")
    reluLayer("Name", "Decoder-Stage-4-ReLU-1")
convolution3dLayer([3 3 3],64, "Name", "Decoder-Stage-4-Conv-
2", "Padding", "same", "WeightsInitializer", "he")
    batchNormalizationLayer("Name", "Decoder-Stage-4-BN-2")
    reluLayer("Name", "Decoder-Stage-4-ReLU-2")
    convolution3dLayer([1 1 1],3,"Name","Final-
ConvolutionLayer", "Padding", "same", "WeightsInitializer", "he")
    softmaxLayer("Name", "Softmax-Layer")
    pixelClassificationLayer("Name", "Segmentation-Layer")];
lgraph = addLayers(lgraph,tempLayers);
clear tempLayers;
% encoder / decoder connections
lgraph = connectLayers(lgraph,"Encoder-Stage-1-ReLU-2","Encoder-Stage-1-
MaxPool");
lgraph = connectLayers(lgraph, "Encoder-Stage-1-ReLU-2", "Decoder-Stage-4-
Concatenation/in2");
lgraph = connectLayers(lgraph, "Encoder-Stage-2-ReLU-2", "Encoder-Stage-2-
MaxPool");
lgraph = connectLayers(lgraph,"Encoder-Stage-2-ReLU-2","Decoder-Stage-3-
Concatenation/in2");
```

```
lgraph = connectLayers(lgraph, "Encoder-Stage-3-ReLU-2", "Encoder-Stage-3-
MaxPool");
lqraph = connectLayers(lqraph,"Encoder-Stage-3-ReLU-2","Decoder-Stage-2-
Concatenation/in2");
lgraph = connectLayers(lgraph,"Encoder-Stage-4-ReLU-2","Encoder-Stage-4-
MaxPool");
lgraph = connectLayers(lgraph,"Encoder-Stage-4-ReLU-2","Decoder-Stage-1-
Concatenation/in2");
lgraph = connectLayers(lgraph, "Decoder-Stage-1-UpConv", "Decoder-Stage-1-
Concatenation/in1");
lgraph = connectLayers(lgraph, "Decoder-Stage-2-UpConv", "Decoder-Stage-2-
Concatenation/in1");
lgraph = connectLayers(lgraph, "Decoder-Stage-3-UpConv", "Decoder-Stage-3-
Concatenation/in1");
lgraph = connectLayers(lgraph, "Decoder-Stage-4-UpConv", "Decoder-Stage-4-
Concatenation/in1");
% analyzeNetwork(lgraph) % comment in to check structure and errors
plot(lgraph) % comment in to show network structure
%% Specify training options
options = trainingOptions('adam', ...
    'MaxEpochs',150, ...
    'InitialLearnRate',5e-4, ...
    'L2Regularization',1e-4,...
    'LearnRateSchedule', 'piecewise', ...
    'LearnRateDropPeriod',5,...
    'LearnRateDropFactor',0.95, ...
    'Plots', 'training-progress', ...
    'Verbose',true, ...
    'VerboseFrequency',20,...
    'Shuffle', 'every-epoch',...%
    'ExecutionEnvironment', 'auto',...
    'MiniBatchSize',MiniBatchSize);
%% Train and save the network
close all
% Specify location to save the network
saved network directory =
'D:\John\2021\MATLAB biostudies\MATLAB 3D UNET\3 Saved models\';
% Train the network
modelDateTime = datestr(now, 'dd-mmm-yyyy-HHMM');
[net,info] = trainNetwork(Training_ds,lgraph,options);
% Timestamp and save the network after training
save([saved network directory,'MLN Biostudies ',modelDateTime,'.mat'],'net','opt
```

ions','info');



3-D Unet architecture schematic. The network uses an input layer for the reflectance data of 64x64x64x1 (x, y, z, channels). The three-class Unet architecture uses an encoder depth of 4 with 64 filters at the level of the first encoder. The network uses complete up-convolutional expansion to yield outputted probability maps that are identically sized to the input layer.

```
% MATLAB SCRIPT: TEST 3D UNET
% All code, image-data and screen-cast tutorial videos available for download at
% the Biostudies database under accession number S-BSST742
clear
close all
clc
% Specify path to bioformats library
addpath('D:\John\2021\MATLAB biostudies\MATLAB 3D UNET\4 TEST Unet\bfmatlab\')
% Specify directory containing test data
Test data directory =
'D:\John\2021\MATLAB biostudies\MATLAB 3D UNET\1 ImageData\TEST\';
% Specify channel number in test data that contains reflectance information
RL channel number = 2;
% COMMIT REFLECTANCE TEST DATA TO DIRECTORY RESCALED [0 1] IN MAT FORMAT
if ~exist('mat_test_data', 'dir')
   mkdir('mat test data');
end
for loop = 1:1
   counter = sprintf('%03d',loop) ;
   % find metadata describing file
   reader = bfGetReader([Test data directory,'TEST ',counter,'.tif']);
   omeMeta = reader.getMetadataStore();
   number_of_channels = omeMeta.getChannelCount(0);
   stackSizeZ = omeMeta.getPixelsSizeZ(0).getValue(); % number of Z slices
   % Load reflectance information
   for zplane = 1:stackSizeZ
       iPlane = reader.getIndex(zplane -1, RL channel number -1, 0) + 1; %
because zplanes and channels are numbered from zero
       channel zimage{zplane} = bfGetPlane(reader, iPlane);
   end
   IM_DATA = cat(3,channel_zimage{:}) ;
   IM DATA = double(1*mat2gray(IM DATA, [0 65535])) ; % rescale zero-one
   save([pwd,'/mat test data/','TEST DATA ',counter,'.mat'], 'IM DATA');
end
clear
88
% load the trained 3D Unet network
load('D:\John\2021\MATLAB biostudies\MATLAB 3D UNET\3 Saved models\MLN Biostudie
s 27-May-2021-1355.mat') ;
% load the rescaled reflectance test data
load([pwd,'/mat_test_data/','TEST_DATA_001.mat']);
%% Patch the reflectance data through the network
****
% Set the patch sizes to be passed to the net
patchSize = [256 256 64];
% Segment blockwise then reassemble in full
% Define image dimensions
```

```
[height, width, depth, nChannel] = size(IM DATA);
patch = zeros([patchSize, nChannel], 'like', IM DATA);
number of height patches =ceil(height/patchSize(1));
number_of_width_patches =ceil(width/patchSize(2));
number of depth patches =ceil(depth/patchSize(3));
% Pad image to have dimensions as multiples of patchSize
height_pad = number_of_height_patches*patchSize(1);
width_pad = number_of_width_patches*patchSize(2);
depth pad = number of depth patches*patchSize(3);
% Amount to pad different dimensions of image
padSize(1) = height pad - height;
padSize(2) = width pad - width;
padSize(3) = depth pad - depth;
% Pad the image by correct amounts
im_pad = padarray (IM_DATA, padSize, 0, 'post');
% Preallocate some matrices to catch the probability maps
out Uncertainty Scores = zeros([size(im pad,1), size(im pad,2), size(im pad,3)],
'double'); % needs to match OutputType in semanticseq below
out scores from network all classes=zeros([size(im pad,1), size(im pad,2),
size(im_pad,3), 3], 'double');
% Loop through blocks of 'patchSize'
for loop height = 1:number of height patches
    for loop_width = 1:number_of_width_patches
        for loop_depth = 1:number_of_depth_patches
            start height position=(loop height-1)*patchSize(1)+1;
            end height position=loop height*patchSize(1);
            start width position=(loop width-1)*patchSize(2)+1;
            end_width_position=loop_width*patchSize(2);
            start_depth_position=(loop_depth-1)*patchSize(3)+1;
            end_depth_position=loop_depth*patchSize(3);
            patch to deploy=...
im pad(start height position:end height position,start width position:end width
position,start depth position:end_depth_position,:);
            % deploy net
            [patch_seg, Scores, allScores] = semanticseg(patch to deploy, net,
'OutputType', 'double',...
            'ExecutionEnvironment', 'auto');
out Uncertainty Scores(start height position:end height position,start width pos
ition:end width position,start depth position:end depth position)...
            =Scores;
out scores from network all classes(start height position:end height position,st
art width position:end width position, start depth position:end depth position,:)
. . .
            =allScores;
```

end

```
end
% Script is general for n classes to this point
out Pmap cat1 = out scores from network all classes(:,:,:,1);
out Pmap cat2 = out scores from network all classes(:,:,:,2);
out Pmap cat3 = out scores from network all classes(:,:,:,3);
% Remove padding from probability maps
out Uncertainty Scores = out Uncertainty Scores(1:height, 1:width, 1:depth);
out Pmap cat1 = out Pmap cat1(1:height, 1:width, 1:depth);
out Pmap cat2 = out Pmap cat2(1:height, 1:width, 1:depth);
out Pmap cat3 = out Pmap cat3(1:height, 1:width, 1:depth);
%% visualise network outputs
figure(2)
clf(figure(2))
ax1 = subplot(2,3,1);
imshow(out Pmap cat1(:,:,60),[])
title('LF-nuclei')
ax2 = subplot(2,3,2);
imshow(out Pmap cat2(:,:,60),[])
title('LF-actin')
ax3 = subplot(2,3,3);
imshow(out Pmap cat3(:,:,60),[])
title('Background/other')
ax4 = subplot(2,3,5);
imshow(out Uncertainty Scores(:,:,60),[])
title('Uncertainty')
linkaxes([ax1 ax2 ax3 ax4], 'xy')
%% Map to 16-bit and save for loading into CellProfiler
uil6 PMAP cat1 xy = uint16(65535*mat2gray(out Pmap cat1, [0 1]));
uil6 PMAP cat2 xy = uint16(65535*mat2gray(out Pmap cat2, [0 1])) ;
ui16_PMAP_cat3_xy = uint16(65535*mat2gray(out_Pmap_cat3, [0 1])) ;
data to save = cat(4,uil6 PMAP cat1 xy,uil6 PMAP cat2 xy,uil6 PMAP cat3 xy);
% Map to 16-bit and save for loading into CellProfiler
for loop = 1:3
   CP channel number = num2str(loop +2) ;
   MultiChImgTile = data to save(:,:,:,loop) ;
   % spec a fiji description
   fiji descr = ['ImageJ=1.52p' newline ...
               'images=' num2str(size(MultiChImgTile,3)*...
                                size(MultiChImgTile,4)*...
                                size(MultiChImgTile,5)) newline...
               'channels=' num2str(size(MultiChImgTile,4)) newline...
               'slices=' num2str(size(MultiChImgTile,3)) newline...
               'frames=' num2str(size(MultiChImgTile,5)) newline...
               'hyperstack=true' newline...
               'mode=grayscale' newline...
               'loop=false' newline...
               'min=0.0' newline...
               'max=65535.0']; % bitdepth spec
   t = Tiff(['Ch ',CP channel number,' Im ','001','.tif'],'w');
```

end

```
tagstruct.ImageLength = size(MultiChImgTile,1);
    tagstruct.ImageWidth = size(MultiChImgTile,2);
    tagstruct.Photometric = Tiff.Photometric.MinIsBlack;
    tagstruct.BitsPerSample = 16;
    tagstruct.SamplesPerPixel = 1;
    tagstruct.Compression = Tiff.Compression.None; %% lzw is not compatible w
CP4 out-of-box
    tagstruct.PlanarConfiguration = Tiff.PlanarConfiguration.Chunky;
    tagstruct.SampleFormat = Tiff.SampleFormat.UInt;
    tagstruct.ImageDescription = fiji_descr;
        for frame = 1:size(MultiChImgTile,5)
            for slice = 1:size(MultiChImgTile,3)
                for channel = 1:size(MultiChImgTile,4)
                    t.setTag(tagstruct)
                    t.write(im2uint16(MultiChImgTile(:,:,slice,channel,frame)));
                    t.writeDirectory(); % saves a new page in the tiff file
                end
            end
        end
        t.close()
end
```

Windows 10: Running the label-free cell segmentation deep learning scripts on an NVIDIA GPU using Python 3.6 and Tensorflow-gpu 1.9.0

In brief, it is recommended to use these deep learning files with:

- Python 3.6
- Tensorflow-gpu 1.9.0
- Keras 2.1.5
- Numpy 1.18.1
- Scipy 1.4.1
- Java SE Development Kit 11.0
- Python-bioformats 1.5.2
- CUDA Toolkit 9.0 / cuDNN v7.6.4

Installation Steps:

1. Install Visual Studio Express Community 2017

https://visualstudio.microsoft.com/dev-essentials/#software

- This is necessary to enable the install of the CUDA toolkit.
- At the link above, join Visual Studio Development Essentials (free sign-up).
- Use the search tool to find, download and install Visual Studio Express Community 2017.
- Using the recommended configurations at every step of the installation works fine.

Hicrosoft Visual Studio Subscr	iptions				jw2020 Sig
Your Downloads	• £03 ·	₽ ij	· /2 • 4	· 🗜 • 🎾 ·	
Benefits Downloads Product Keys	Subscriptions Get Help	Marketplace			
All Visual Studio Com	munity				×
FILTER BY PRODUCT FAMILY	DOWNLOADS (5)				Sort by: Release date \checkmark
Visual Studio 2013 Update 5	Visual Studio Community	2019 (version 16.11)			
Visual Studio 2015 Update 3	🔍 No key required	1 Info	Release date: 16/Nov/2021	x64 V Multiple Lang V exe V	Download 🛓
Visual Studio 2017 (version 15.9) Visual Studio 2019 (version 16.11)	Visual Studio Community	2022 (version 17.0)			
Visual Studio 2022 (version 17.0)	🔦 No key required	1 Info	Release date: 16/Nov/2021	x64 V Multiple Lang V exe V	Download 🛓
	Visual Studio Community	r 2017 (version 15.9)			
	🔦 No key required	1 Info	Release date: 09/Nov/2021	x64 V Multiple Lang V EXE V	Download 🛓
	Visual Studio Community	2015 with Update 3			
	A No key required	1 Info	Release date: 27/Jun/2016	x64 V English V DVD V	Download 🛓
	Visual Studio Community	2013 with Update 5			
	🔍 No key required	1 Info	Release date: 13/Nov/2015	x86 V English V DVD V	Download 🛓

- Restarting your PC after installing Visual Studio is a probably a good idea.

2. Install CUDA Toolkit 9.0 and accompanying patches for Windows 10.

- Tensorflow-GPU 1.9.0 requires CUDA 9.0 - not whatever the latest version of the toolkit is. - This is available at the Nvidia website "CUDA Toolkit Archive" --> select CUDA Toolkit 9.0 https://developer.nvidia.com/cuda-toolkit-archive

🔲 🞯 NVIDIA. DEVELOPER

CUDA Toolkit Archive

Home

Previous releases of the CUDA Toolkit, GPU Computing SDK, documentation and developer drivers can be found using the links below. Please select the release you want from the list below, and be sure to check www.nvidia.com/drivers for more recent production drivers appropriate for your hardware configuration.

Download Latest CUDA Toolkit Learn More about CUDA Toolkit 11

Latest Release

CUDA Toolkit 11.5.1 (November 2021), Versioned Online Documentation

Archived Releases

CUDA Toolkit 11.5.0 (October 2021), Versioned Online Documentation CUDA Toolkit 11.4.3 (November 2021), Versioned Online Documentation CUDA Toolkit 11.4.2 (September 2021), Versioned Online Documentation CUDA Toolkit 11.4.1 (August 2021), Versioned Online Documentation CUDA Toolkit 11.4.0 (June 2021), Versioned Online Documentation CUDA Toolkit 11.3.1 (May 2021), Versioned Online Documentation CUDA Toolkit 11.3.0 (April 2021), Versioned Online Documentation CUDA Toolkit 11.2.2 (March 2021), Versioned Online Documentation CUDA Toolkit 11.2.1 (Feb 2021), Versioned Online Documentation CUDA Toolkit 11.2.0 (Dec 2020), Versioned Online Documentation CUDA Toolkit 11.1.1 (Oct 2020), Versioned Online Documentation CUDA Toolkit 11.1.0 (Sept 2020), Versioned Online Documentation CUDA Toolkit 11.0 Update1 (Aug 2020), Versioned Online Documentation CUDA Toolkit 11.0 (May 2020), Versioned Online Documentation CUDA Toolkit 10.2 (Nov 2019), Versioned Online Documentation CUDA Toolkit 10.1 update2 (Aug 2019), Versioned Online Documentation CUDA Toolkit 10.1 update1 (May 2019), Versioned Online Documentation CUDA Toolkit 10.1 (Feb 2019), Online Documentation CUDA Toolkit 10.0 (Sept 2018), Online Documentation CUDA Toolkit 9.2 (May 2018), Online Documentation CUDA Toolkit 9.1 (Dec 2017). Online Documentation CUDA Toolkit 9.0 (Sept 2017), Online Documentation IDA Toolkit 8 0 GA2 IEeh 2017L Online Docum

- Select the target platform as Windows, X86_64, version 10 and the installer type as exe(local)

41



Download Installers for Windows 10 x86_64

The base installer is available for download below. There are 4 patches available. These patches require the base installer to be installed first.

> Base Installer	Download (1.4 GB) 📥
Installation Instructions:	
 Double click cuda_9.0.176_win10.exe Follow on-screen prompts 	
> Patch 1 (Released Jan 25, 2018)	Download (54.1 MB) 📥
cuBLAS Patch Update: This update to CUDA 9.0 includes new GEMM kernels optimized for the Volta architecture and improved heuristics to select GEMM kernels for given input sizes.	
> Patch 2 (Released Mar 5, 2018)	Download (54.7 MB) 📥
cuBLAS Patch Update: This update to CUDA 9 includes GEMM heuristics improvements to selects the most optimized algorithms for input sizes commonly used in Deep Learning RNNs. The update also includes other bug- fixes and performance enhancements.	
> Patch 3 (Released Jun 7, 2018)	Download (82.3 MB) 📩
cuBLAS Patch Update: This update to cuBLAS addresses issues with Convolutional Seq2Seq and RNN inference performance.	
> Patch 4 (Released Aug 6, 2018)	Download (56.2 MB) 📥
cuBLAS Patch Update: This update to cuBLAS includes optimized implementations of GEMV operations for mixed precision input and output types and important fixes to address performance issues.	
The checksums for the installer and patches can be found in Installer Checksums. For further information, see the Installation Guide for Microsoft Windows and the CUI	DA Quick Start Guide.

- First install the "base installer" following the standard, "express configurations" options.

- Then install Patches 1-4 sequentially in order by just following the on-screen prompts after each download.

3. Install cuDNN v7.6.4 for CUDA 9.0

- To download and install cuDNN, first join the NVIDIA Developer Program – which is free. https://developer.nvidia.com/cudnn-download-survey

- Once signed in, proceed the cuDNN download page and click archived cuDNN releases: https://developer.nvidia.com/rdp/cudnn-archive

cuDNN Download

Home

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

a) Agree To the Terms of the cuDNN Software License Agreement
Note: Please refer to the Installation Guide for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.
For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the Deep Learning SDK Documentation web page.

Download cuDNN v8.3.1 (November 22nd, 2021), for CUDA 11.5
Download cuDNN v8.3.1 (November 22nd, 2021), for CUDA 10.2
Archived cuDNN Releases

Ethical AI
NVIDIA's platforms and application frameworks enable developers to build a wide array of AI applications. Consider potential algorithmic bias when choosing or creating the models being deployed. Work with the model's developer to ensure that it meets the requirements for the relevant industry and use case; that the necessary instruction and documentation are provided to understand error rates, confidence intervals, and results; and that the model is being used under the conditions and in the manner intended.

- Scroll down to the option to download cuDNN v7.6.4 for CUDA 9.0 and download the library for Windows 10.

Download cuDNN v7.6.4 (September	27, 2019], for CUDA 9.0
Library for Windows,	Mac, Linux, Ubuntu(x86_64 architecture)
cuDNN Library for Windows 7	
cuDNN Library for Windows 10	
cuDNN Library for Linux	
cuDNN Runtime Library for Ubuntu1	5.04 (Deb)
cuDNN Developer Library for Ubuntu	16.04 (Deb)
cuDNN Code Samples and User Guid	e for Ubuntu16.04 (Deb)
cuDNN Runtime Library for Ubuntul	4.04 (Deb)
cuDNN Developer Library for Ubuntu	14.04 (Deb)
cuDNN Code Samples and User Guid	e for Ubuntu14.04 (Deb)
Library for Red Hat (x	(86_64)
cuDNN Runtime Library for RedHat/0	lentos 7.3 (RPM)
cuDNN Developer Library for RedHat	/Centos 7.3 (RPM)
cuDNN Code Samples and User Guid	e for RedHat/Centos 7.3 (RPM)

- Unzip the downloaded cuDNN .zip file.

- Inside are three files which need to be copied to the correct folder subdirectories of your Window 10 installation of the CUDA 9.0 toolkit:

- These files are cudnn64_7.dll, cudnn.h and cudnn.lib:

1. cudnn64_7.dll

Copy the file from the unzipped cuDNN download folder at e.g., <ur><unzipped_download>/cudnn-9.0-windows10-x64-v7.6.2.24\cuda\bin\cudnn64_7.dll

and paste it into the NVIDIA GPU computing toolkit v9.0 subdirectory 'bin' located at e.g., C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin\

2. cudnn.h

Copy the file from the unzipped cuDNN download folder at e.g., <ur>unzipped_download>/cudnn-9.0-windows10-x64-v7.6.2.24\cuda\ include\cudnn.h

and paste it into the NVIDIA GPU computing toolkit v9.0 subdirectory 'include' located at e.g., C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\include\

3. cudnn.lib

Copy the file from the unzipped cuDNN download folder at e.g., <unzipped_download>/cudnn-9.0-windows10-x64-v7.6.2.24\cuda\lib\x64\cudnn.lib

and paste it into the NVIDIA GPU computing toolkit v9.0 subdirectory 'include' located at e.g., C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\lib\x64\



4. Check that the CUDA environment variables are set in Windows 10

- On the right-side, of the dialogue that opens, click "Advanced system settings"

Settings		- B X
û Home	About	
Find a setting P	Your PC is being monitored and	This page has a few new settings
System	protected. See details in Windows Security	Some settings from Control Panel have moved here, and you can copy your PC info so it's easier to share.
C Display	Device specifications	Related attrices
40 Sound		Related settings
Notifications & actions	OMEN by HP Desktop PC 880-p1xx Device name DESKTOP-1998L7D	Device Manager
∂ Focus assist	Processor Intel/R) Core(TM) (7-8700K CPU (8) 3 70GHz - 3.70 GHz	Remote desktop
() Power & sleep	Installed RAM 16.0 GB Device ID 7DB16A01-6579-4FCF-AD9A-3446341A2189	
🖙 Storage	Product ID 00329-10180-00000-AA484 System type 64-bit operating system, z64-based processor	Advanced system settings
ानु Tablet	Pen and touch No pen or touch input is available for this display	
H Multi-tasking	Copy	Get help
部 Projecting to this PC	Rename this PC	
X Shared experiences	Windows specifications	
1 Clipboard	Edition Windows 10 Enterprise Version 20H2	
> Remote Desktop	Installed on 09/11/2021 OS build 15042.1348	
③ About	Experience Windows Feature Experience Pack 120.2212.39200	
	Сору	
	Change the product key or upgrade your edition of Windows	
	Read the Microsoft Services Agreement that applies to our services	
	Read the Microsoft Software Licence Terms	

•					
Computer Name	Hardware	Advanced	System Protection	Remote	
You must be log	gged on as a	an Administrat	or to make most of th	ese changes	3.
Visual effects	. processor s	schedulina, m	emory usage and virt	ual memory	
				,	
				Settings	
User Profiles					
Desktop setti	ngs related t	o your sign-in			
			_		
				Settings	
Start-up and F	Recovery				
System start-u	up, system fa	ilure and deb	ugging information		
			_		
				Settings	
			-		
			Environme	nt Variables.	

- In the "System Properties" dialogue box, click "Environment Variables" at the bottom

×

Environment Variables

ariable	Value
P_JAVA_HOME	C:\Program Files (x86)\CellProfiler\java
RE_HOME DneDrive DneDriveCommercial lath EMP MP	C:\PROGRA-TUava\RE18-1.0_2 C:\Users\Claire\OneDrive - University of Cambridge C:\Users\Claire\OneDrive - University of Cambridge C:\Users\Claire\AppData\Local\Programs\Python\Python36\Scripts\C:\Users\Claire\AppData\Local\Programs\Python\Py. C:\Users\Claire\AppData\Local\Temp C:\Users\Claire\AppData\Local\Temp
	New Edit Delete
ariables omSpec UDA_PATH	Value C\WINDOWS\sectem32\cmd.exe C\Program Files\WIDIA GPU Computing Toolkit\CUDA\v9.0
tem variables ariable COMSPEC CUDA_PATH CUDA_PATH_V9_0 INVELPATA	Value CNVINDOWSLowtem32/cmd.exe C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0 C:\Program Siles\Systemscomeeschiescomeeschiescome
tem variables ariable comSner CUDA_PATH CUDA_PATH CUDA_PATH CUDA_PATH CUDA_PATH CUDA_SAMPLES_ROOT IVCUDASAMPLES_ROOT IVCUDASAMPLES_0_ROOT IVCUDASAMPLES_0_ROOT INITIASERVICE S ath ATHEXT Latformcode ROCESSOR_ARCHITECTURE ROCESSOR_ARCHITECTURE ROCESSOR_ARCHITECTURE ROCESSOR_REVISION SModulePath	Value C\\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0 C\\Program Files\NVIDIA Corporation\CUDA Samples\v9.0 C\\Program Tiles\NVIDIA Corporation\CUDA Samples\v9.0 C\\Program Files\NVIDIA Corporation\CUDA Samples\v9.0 C\\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\binxC\\Program Files\NVIDIA GPU Computing Toolkit\CUD. C\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\binxC\\Program Files\NVIDIA GPU Computing Toolkit\CUD. .COM_EXE;BAT;.CMD;\VBS;\VBE;JS;JSE;WSF;WSH;MSC 1M AMD64 Intel64 Family 6 Model 158 Stepping 10, GenuineIntel 6 9e0a %ProgramFiles%\WindowsPowerShell\Modules;C\WINDOWS\system32\WindowsPowerShell\v1.0\Modules

- Make sure in the <u>bottom window</u> (labelled "System Variables") that variables named CUDA_PATH and CUDA PATH V9.0 exist and point to the correct locations e.g.,

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin

and

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\libnvvp

respectively.

<u>- These paths should be auto-installed</u>. If they are missing on your system, they can be added by clicking the "New" button below the system variables <u>bottom</u> panel and entering the name and path into the dialogue box for your system.

5. Install Python 3.6.8

- To install Python version 3.6.8 navigate to the previous release at: <u>https://www.python.org/downloads/release/python-368/</u>

	🥏 Py	thon Rele	ase Python 3.6.8 Py 🗙	+		-	D
\leftarrow	\rightarrow	С	https://www.pyt	hon.org/downloads/release/python-368/	₹=	Ē	

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		48f393a04c2e66c77bfc114e589ec630	23010188	SIG
XZ compressed source tarball	Source release		51aac91bdf8be95ec0a62d174890821a	17212420	SIG
macOS 64-bit/32-bit installer	macOS	for Mac OS X 10.6 and later	eb1a23d762946329c2aa3448d256d421	33258809	SIG
macOS 64-bit installer	macOS	for OS X 10.9 and later	786c4d9183c754f58751d52f509bc971	27073838	SIG
Windows help file	Windows		0b04278f5bdb8ee85ae5ae66af0430b2	7868305	SIG
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	73df7cb2f1500ff36d7dbeeac3968711	7276004	SIG
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	72f37686b7ab240ef70fdb931bdf3cb5	31830944	SIG
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	39dde5f535c16d642e84fc7a69f43e05	1331744	SIG
Windows x86 embeddable zip file	Windows		60470b4cceba52094121d43cd3f6ce3a	6560373	SIG
Windows x86 executable installer	Windows		9c7b1ebdd3a8df0eebfda2f107f1742c	30807656	SIG
Windows x86 web-based installer	Windows		80de96338691698e10a935ecd0bdaacb	1296064	SIG

- Scroll to the bottom of the page and download the "Windows x86-64 executable installer" with description "AMD64/EM64T/x64".

- Once downloaded, follow the on-screen prompts to install Python 3.6.8.



- Check the box to add Python to the Windows path.

6. Install Java Development Kit 11

- This is used by Python-bioformats to enable read/write of image-data.

- Download and install Java Development kit 11

https://www.oracle.com/java/technologies/downloads/#java11

Java SE Development Kit 11.0.5 You must accept the Oracle Technology Network License Agreement for Oracle Java SE to download this software. Thank you for accepting the Oracle Technology Network License Agreement for Oracle Java SE; you may now download this software.						
Product / File Description	File Size	Download				
Linux	147.82 MB	jdk-11.0.5_linux-x64_bin.deb				
Linux	154.47 MB	• jdk-11.0.5_linux-x64_bin.rpm				
Linux	171.62 MB	jdk-11.0.5_linux-x64_bin.tar.gz				
macOS	166.73 MB	jdk-11.0.5_osx-x64_bin.dmg				
macOS	167.06 MB	jdk-11.0.5_osx-x64_bin.tar.gz				
Solaris SPARC	188.32 MB	jdk-11.0.5_solaris-sparcv9_bin.tar.gz				
Windows	151.39 MB	jdk-11.0.5_windows-x64_bin.exe				
Windows	171.47 MB	jdk-11.0.5_windows-x64_bin.zip				

- Once Java SE Development Kit 11 is installed, set the **JAVA_HOME** environment variable and add the Java development kit to the Windows path. To do this, as above, open a File Explorer window, right-click on the '**This PC**' option and select '**Properties**' from the drop-down menu.



The control panel will pop up as a separate window. Select '**Advanced system settings'** from the list appearing at the right of the window.

In the 'system properties' dialogue that opens, select '**Environment Variables**'. This will cause the environment variables window to appear. To set the 'JAVA_HOME' variable, Click the '**New**' button option at the bottom of this window in the '<u>System variables'</u> section.

Name the new variable 'JAVA_HOME' and use the '**Browse Directory**' option to specify the path to JDK 11. Select '**OK**' to create this new variable:

mputer Name Hardware Advanced System Protection Remote	User variables for Claire			
You must be logged on as an Administrator to make most of these changes.	Variable	Value		^
Performance	JRE HOME	C:\PROGRA~1\Java\JRE18~1.0 2		
Visual effects, processor scheduling, memory usage and virtual memory	OneDrive	C:\Users\Claire\OneDrive - University of Ca	ambridge	
	OneDriveCommercial	C:\Users\Claire\OneDrive - University of Ca	ambridge	
Settings	Path	C:\Users\Claire\AppData\Local\Programs\	Python\Python36\Scripts	
leer Profiles	TEMP	C:\Users\Claire\AppData\Local\Temp		
Deskton settings related to your sign-in	TMP	C:\Users\Claire\AppData\Local\Temp		
beautop acturing related to your aign an				~
Settings		New	Edit Delete	
Startuin and Recovery				_
System start-up, system failure and debugging information	System variables			
System startup, system raildre and debugging information	W-11	14.1		•
Settings	Variable	value		<u> </u>
	ComSpec	C:\WINDOWS\system32\cmd.exe	T. II. W. CUDAN A.A.	
Environment Variables		C:\Program Files\NVIDIA GPU Computing		
Environment Valiables	DriverData	C:\Windows\System32\Drivers\DriverData	IOURIL(CODA(V).0	
	JRE HOME	C:\PROGRA~1\Java\JRE18~1.0_2		
OK Cancel Apply	NUMBER OF PROCESSORS	12		
	NVCUDASAMPLES ROOT	C:\ProgramData\NVIDIA Corporation\CUD	A Samples\v9.0	~
				_
		New	Edit Delete	
A DESCRIPTION OF A DESC			OK Cancel	
			Cancer	
	New System Variable			
	New System variable			
	Variable name: JAVA_HC	IME		
	Variable name: JAVA_HC)ME		
	Variable name: JAVA_HC Variable value: C:\Progr)ME am Files\Java\jdk-1.8.2		
	Variable name: JAVA_HC Variable value: C:\Progr)ME am Files\Java\jdk-1.8.2		

To add JDK11 to the Windows path, Click on the exisiting 'Path' system variable and click the '**Edit**' button. A new window will appear. Click new and type '**%JAVA_HOME%\bin**'.

System Properties X	Environment Variables		× Edit environment variable	×
Computer Name Hardware Advanced System Protection Remote You must be logged on as an Administrator to make most of these changes.	User variables for Claire		C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\bin A C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.0\libnv	New
Performance	Variable	Value	C:\Program Files\Common Files\Oracle\Java\javapath	Edit
Visual effects processor scheduling memory usage and vidual memory	CP_JAVA_HOME	C:\Program Files (x86)\CellProfiler\java	C:\Program Files (x86)\Common Files\Oracle\Java\javapath	
hada crocco, processor seriesaling, memory asage and made memory	JRE_HOME	C:\PROGRA~1\Java\JRE18~1.0_2	C:\windows\system32	Browse
Settings	OneDrive	C:\Users\Claire\OneDrive - University of Cambridge	C:\windows	
	OneDriveCommercial	C:\Users\Claire\OneDrive - University of Cambridge	C:\windows\System32\Wbem	Delete
User Profiles	Path	C:\Users\Claire\AppData\Local\Programs\Python\Python36\Scripts	C:\windows\System32\WindowsPowerShell\v1.0\	
Desktop settings related to your sign-in	TEMP	C:\Users\Claire\AppData\Local\Temp	C:\windows\System32\OpenSSH\	
	TMP	C:\Users\Claire\AppData\Local\Temp	C:\Program Files (x86)\Intel\Intel(R) Management Engine Compon	Move Up
Settings		New Edit Delate	C:\Program Files\Intel\Intel(R) Management Engine Components\	
		New Edit Delete	C:\Program Files (x86)\Intel\Intel(R) Management Engine Compon	Move Down
Start-up and Recovery			C:\Program Files\Intel\Intel(R) Management Engine Components\I	
System start-up, system failure and debugging information	System variables		C:\Program Files\MATLAB\R2021b\runtime\win64	
	Variable	Value	C:\Program Files\MATLAB\R2021b\bin	Edit text
Settings			C:\Program Files\MATLAB\R2020a\bin	
	NVCUDASAMPLES_ROUT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\V9.0	C:\Program Files\Git\cmd	
Environment Variables	NVCUDASAMPLES9_0_ROOT	C:\ProgramData\NVIDIA Corporation\CUDA Samples\v9.0	%JAVA HOME%\bin	
	Onlineservices	Unline services	C:\WINDOWS\system32	
	05	Windows_N1	C:\WINDOWS	
OK Cased Asph	Path	C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\V9.0\bin;C	C:\WINDOWS\System32\Wbem	
OK Candel Apply	PATHEXT	.COM; EXE; BAT; CMD; VBS; VBE; JS; JSE; WSF; WSF; MSH; MSC	C:\WINDOWS\System32\WindowsPowerShell\v1.0\	
	blattormcode	IM	C:\WINDOWS\System32\OpenSSH\	
		New Edit Delete	C:\Program Files\Microsoft SQL Server\120\Tools\Binn\	
			C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common	
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1		OK Cancel	ОК	Cancel

- Click '**OK**' to apply changes and close all opened windows.

7. Setup a virtual environment and install the dependencies necessary to run the code

- Download and unzip the "Python_biostudies" folder from the BioStudies project archive to a suitable location on your computer.

📙 🛃 🚽 = 20211118_Python_bic	ostudies				– 🗆 ×
File Home Share View					~ (?)
Pin to Quick Copy Paste access	ith iortcut	Move to Copy to Rename Organise	New folder	Properties	Select all Select none
\leftarrow \rightarrow \checkmark \uparrow \bigcirc « Users \rightarrow John	n > 20	211118_Python_biostudies		م ب 5	Search 20211118
Documents	* ^	Name		Date modified	а Туре
CELLPROFILER_OUT	*	Python		29/11/2021 12	1:56 File folder
TRAIN	*	requirements		29/11/2021 12	:58 Text Docum
👩 Recycle Bin	*				
TRAIN_DATA	*				
20200429_TiO2FiguresBackup	*				
2_TRAIN_Unet					
2_TRAIN_Unet					
4_TEST_Unet					
4_TEST_Unet					
	~	<			>
2 items 1 item selected					

- Run command prompt as administrator:

- To do this, Search "CMD", right-click on "Command Prompt" from the search results dialogue and click "Run as administrator":



- Using the 'cd' command, change the current directory to the unzipped python scripts folder downloaded from Biostudies e.g.,



- Make a new subdirectory 'venv' and initialise an instance of Python inside it e.g.,

C:\Users\John\20211118_Python_biostudies>python -m venv ./venv

- Activate the newly-created virtual environment e.g.,

C:\Users\John\20211118_Python_biostudies>venv\Scripts\activate.bat

- Update pip to the latest version (N.B., this is critical to the successful install of javabridge with Python 3.6)

(venv) C:\Users\John\20211118_Python_biostudies>python -m pip install --upgrade pip

- Using the "requirements.txt" file included in the BioStudies download, install the required dependencies and the Spyder 4.0 Integrated Development Environment (IDE) to the newly-created virtual environment e.g.,

(venv) C:\Users\John\20211118_Python_biostudies>pip install -r requirements.txt

8. Training a 2-D UNET model

- A screencast video is included with the BioStudies project archive.

- Download and unzip the Python BioStudies project archive at a suitable location on your computer.

📙 🛃 📮 Python					-		×
File Home Share View							~ 🕐
Pin to Quick Copy Paste	Move Copy to * Copy	New item ▼ ↑ Easy access ▼ folder	Properties	Select all	ion		
Clipboard	Organise	New	Open	Select			
\leftarrow \rightarrow \checkmark \uparrow \square \rightarrow This PC \rightarrow Windows	(C:) > Users > John > 20211118	Python_biostudies > Pyth	ion >	5 V	,⊂ Sea	rch Pytho	n
CELLPROFILER_OUT	A Name	Date	modified Ty	pe	Size		
TRAIN	2D_UNET_PYTHON	18/11	/2021 13:43 Fil	le folder			
A Recycle Bin	B 3D_UNET_PYTHON	18/11	/2021 13:44 Fi	le folder			
TRAIN_DATA	*						
20200429_TiO2FiguresBackup	*						
2_TRAIN_Unet							
2_TRAIN_Unet							
4_TEST_Unet							
4_TEST_Unet							
OneDrive - University of Cambridge							
University of Cambridge							
2 items	v						

- Inside the 2D_UNET_Python folder, there is a sequentially-organised workflow starting with raw data and ending with a CellProfiler pipeline that obtains cell features from the label-free cell segmentation:

I Image: Share Image: View					-	□ × ^ (?)
Ar Pin to Quick Copy Copy	Move Copy to Copy to Copanise	New item • Tasy access • New folder	Properties • Open • Edit • History Open	Select all	ion	
\leftarrow \rightarrow \checkmark \uparrow \square \ll Users \Rightarrow John \Rightarrow 202	11118_Python_biostudies > Python	> 2D_UNET_PYTHON >	2D_PeyersPatch >	5 v	, Search	2D_Peyers
4_TEST_Unet	^ Name	Date	modified Ty	pe	Size	
OneDrive - University of Cambridge	1_ImageData	18/11	/2021 13:43 Fil	e folder		
🙀 University of Cambridge	2_TRAIN_Unet 3_Saved_models	07/12 29/11	/2021 21:08 Fil /2021 15:50 Fil	e folder e folder		
💻 This PC	4_TEST_Unet	29/11	/2021 16:00 Fil	e folder		
3D Objects	5_CellProfiler_cell_measu	urements 18/11	/2021 13:44 Fil	e folder		
Desktop						
🗄 Documents						
👆 Downloads						
b Music						
Pictures						
📑 Videos						
L Windows (C:)						
5 items	¥					

- In command prompt with the virtual environment active, type 'Spyder' to start the Spyder4 IDE.



- To train a 2-D UNET model, set the Spyder working directory to the "2_TRAIN_Unet" directory. Open the "A_TRAIN_UNET" script.



Update the path saving the training data and pixel classification labels in numpy format.This should be inside the"2_TRAIN_Unet" directory.



- The script will save the model along with checkpoint values taken during training. You should specify where these files are to be saved. This should be folder 3 of the workflow: ("3_Saved_models").



- To start model training, click the green arrow button at the top of the Spyder dialogue or type the script name at the command line.



- Model training takes several hours (~ 4h on a NVIDIA GTX 1080 Ti GPU)

- The newly-trained model will be saved in the "3_Saved_models" directory.

9. Testing a pretrained 2-D model using unseen data

- A screencast video is included with the BioStudies project archive.

- Change the Spyder working directory to the "4_Test_Unet" directory. Open the

"A_TEST_UNET" script.

Spyder (Pyth	non 3.6)								
File Edit Search	n Source Run De	ebug Consoles P	rojects Tools View	v Help					
🗅 🗁 🖻		D 🕨 🚺	🔁 💵 😋		· 🔳 🗈 🛛	8 I 🔎	-	€ →	C:\Users\John\20211118_Python_biostudies\Python\20_UNET_PYTHON\20_PeyersPatch\4_TEST_Unet

- Change the path to the unseen test image (located in the "1_ImageData" folder).



- Specify which pretrained network to use (pretrained networks are located in the "3_Saved_models" folder

pretrained_model = unet()
pretrained_model.load_weight;('C:/Users/John/20211118_Python_biostudies/Python/2D_UNET_PYTHON/2D_PeyersPatch/3_Saved_models/my_h5_model_fully_trained.h5'

- Clicking the green arrow button or type the file name at the command line to process the unseen reflectance data with the selected pretrained network:



- Probability maps for the LF-nuclei, LF-actin and Background/Other classes will be saved in the working directory named ready for input into the CellProfiler pipeline (filenames Ch_5_lm_001, Ch_6_lm_001, Ch_7_lm_001, respectively).



10. Training a 3-D UNET Model

- A screencast video is included with the BioStudies project archive.

- Inside the 3D_UNET_Python folder, there is a sequentially-organised workflow starting with raw data and ending with a CellProfiler pipeline that obtains 3-D cell features from the label-free cell segmentation:

📙 🛃 📮 🗧 3D_UNET_PYTHON				- 🗆 ×
File Home Share View				^ (
★ Cut ★ Copy Paste Copy path Copy Paste shortcut Clipboard	Move Copy to ~ Copy to ~ Copy organise	New item • Easy access • New Open	n Select all Select none Invert selection Select	1
← → ~ ↑ 🛄 « John > 20211118_Pyt	hon_biostudies > Python > 3D_UNET_PYT	HON > 3D_UNET_PYTHON >	ٽ ~	
4_TEST_Unet	^ Name	Date modified	Туре	Size
OneDrive - University of Cambridge	1_ImageData	18/11/2021 13:44	File folder	
R. University of Cambridge	2_TRAIN_Unet	30/11/2021 13:46	File folder	
in Oniversity of Cambridge	3_Saved_models	07/12/2021 17:50	File folder	
This PC	4_TEST_Unet	07/12/2021 17:39	File folder	
🗊 3D Objects	5_CellProfiler_cell_measurements	18/11/2021 13:45	File folder	
E. Desktop				
Documents				
🖶 Downloads				
b Music				
Pictures				
Videos				
Windows (C:)				
5 items	v			

56

- To train a 3-D UNET model, open Spyder and set the working directory to the "2_TRAIN_Unet" directory. Open the "A_TRAIN_UNET_3D" script.



- Specify the checkpoint path during model training:



- Specify the location to save the model when training completes. This should be the "3_Saved_models" directory.



- Click "Run" to commence model training.
- This takes considerable time (approximately 6h on a NVIDIA GTX 1080 Ti GPU card).



11. Testing a pretrained 3-D UNET Model using unseen data

- A screencast video is included with the BioStudies project archive.

- Change the Python working directory to the "4_Test_Unet" directory. Open the

"A_TEST_UNET_3D" script.



- Specify the location of the unseen test image-data. This is located inside the "1_ImageData" folder at "1_ImageData/TEST/".

- Specify the channel containing the reflectance information (here, '2').

- Specify where the unseen data should be stored upon conversion to Numpy format. This should be inside the "4_Test_Unet" folder.

	#%% Specify directory containing test data			
58	Test_data_directory = 'C:/Users/John/20211118_Python_biostudies/Python/3D_UNET_	PYTHON/3D	UNET_PYTHON/1	_ImageData/TEST/'
	# Specify channel number in test data that contains reflectance information			
	RL channel number = 2;			
	#COMMIT REFLECTANCE TEST DATA TO DIRECTORY RESCALED [0 1] IN MAT FORMAT			
	MYDIR ='C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHON/3D UNET	PYTHON/4 1	TEST Unet/npy	test data/'
	CHECK_FOLDER = os.path.isdir(MYDIR)			
66				

- Specify a pretrained 3-D Unet model from the "3_Saved_models" directory.

285 206 #XX load the trained 3D Unet network 207 #model.load_weights("C:/Users/John/20211118_Python_biostudies/Python/3D_UNET_PYTHON/3D_UNET_PYTHON/3_Saved_models/my_h53D_model.h5") 208 pretrained_model.load_weights("C:/Users/John/20211118_Python_biostudies/Python/3D_UNET_PYTHON/3D_UNET_PYTHON/3_Saved_models/my_h53D_model_updated.h5") 209 pretrained_model.load_weights("C:/Users/John/20211118_Python_biostudies/Python/3D_UNET_PYTHON/3D_UNET_PYTHON/3_Saved_models/my_h53D_model_updated.h5")

- Click the green arrow to process the unseen reflectance data with the 3-D Unet model.

۲	Spyder	(Pytho	on 3.6)																
File	Edit S	earch	Sourc	e Run	Debug	g Cons	soles	Project	s Tool	s View	Help								
Ľ			8		@			2	₽	Ç	M	۲:	¶≡	c≡	₩		K 2	æ	è

- Probability maps for the LF-nuclei, LF-actin and Background/Other classes will be saved as multipage .TIFF files in the working directory named ready for input into the CellProfiler pipeline (filenames Ch_3_lm_001, Ch_4_lm_001, Ch_5_lm_001, respectively).

I Image: Imag	anage 4_TEST_Unet				-	□ × ^ (2)
Pin to Quick access Copy Paste Cipiboard	Move Copy to* Copy to* Copy to* Copy	New item •	Properties Open	Select all	ne ection	
← → ~ ↑ 📙 « 20211118_Python_biost	tudies > Python > 3D_UNET_PY	THON > 3D_UNET_PYTHO	N > 4_TEST_Unet >	ٽ v	, Sear	ch 4_TEST_Unet
4_TEST_Unet	^ Name ^	Date	Туре	Size	Tags	
OneDrive - University of Cambridge	npy_test_data	18/11/2021 13:45	File folder			
University of Cambridge	A_TEST_UNET_3D	07/12/2021 17:39	Python File TIFF File	16 KB 219 173 KB		
💻 This PC	Ch_4_Im_001	07/12/2021 15:26	TIFF File	219,173 KB		
🇊 3D Objects	Ch_5_Im_001	07/12/2021 15:26	TIFF File	219,173 KB		
Desktop	Predictions	07/12/2021 15:19	PNG File	290 KB		
Documents						
Downloads Music						
E Pictures						
Videos						
Kindows (C:)	~					
6 items 1 item selected 214 MB						

58

```
% PYTHON SCRIPT: TRAIN 2D UNET
% All code, image-data and screen-cast tutorial videos available for download at
% the Biostudies database under accession number S-BSST742
# -*- coding: utf-8 -*-
Created on Mon Sep 20 23:58:04 2021
@author: Paul
#Import all required modules
from PIL import Image
import cv2
import matplotlib.pyplot as plt
import matplotlib.pyplot as plt
import skimage
from skimage import data
from skimage.filters import threshold otsu
import cv2
import os
import numpy as np
import skimage.transform as trans
import tensorflow
from tensorflow.keras.models import *
from tensorflow.keras.layers import *
from tensorflow.keras.optimizers import *
from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler
from tensorflow.keras import backend as keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import sklearn.feature extraction
from sklearn.datasets import load sample image
from sklearn.feature extraction import image
from datetime import datetime
from keras.utils import to categorical
from tensorflow.python.keras.optimizers import *
import skimage.io
import h5py # this was missing
#88
# Read in the reflectance data for training
Training data holder =
skimage.io.imread('C:/Users/John/20211118 Python biostudies/Python/2D UNET PYTHO
N/2D PeyersPatch/1 ImageData/TRAIN IMAGE.tif')
Training data = Training data holder[0,0,:,:,2]
#Read in the mask to isolated just the lymphoid tissue
mask = Training data holder[0,0,:,:,3]
thresh = threshold otsu(Training data)
mask = mask > thresh
mask = mask*1
mask = np.uint16(mask)
# Apply the mask to the training data
```

```
Training data = np.multiply(Training data,mask);
Training data = np.double(Training data);
# Rescale the training data in the interval [0 1]
Training data norm holder = np.zeros((8551, 5701), np.double)
Training data norm = cv2.normalize(Training data, Training data norm holder ,
1.0, 0.0, cv2.NORM MINMAX)
#Read in the pixel-class labels created from the nuclei and actin staining
Training labels holder =
skimage.io.imread('C:/Users/John/20211118 Python biostudies/Python/2D UNET PYTHO
N/2D PeyersPatch/1_ImageData/TRAIN_LABELS.png') ;
Training labels = np.where(Training labels holder == 3, 0,
Training labels holder)
#% Visually inspect the data and labels
fig1, (ax1, ax2) = plt.subplots(1, 2) # figure1
ax1.imshow(Training data norm)
ax1.set title('Reflectance data')
ax2.imshow(Training labels)
ax2.set title('Pixel classification labels')
#%% Once happy with labels and data, commit the reflectance information data to
sub-directory in .npy format
MYDIR =
'C:/Users/John/20211118 Python biostudies/Python/2D UNET PYTHON/2D PeyersPatch/2
_TRAIN_Unet/npy_training_data/
CHECK FOLDER = os.path.isdir(MYDIR)
# If folder doesn't exist, then create it.
if not CHECK_FOLDER:
   os.makedirs(MYDIR)
np.save(os.path.join(MYDIR, 'Training_data'), Training_data_norm)
# Create matrices to store training data
Patching data = np.zeros((8551, 5701,2))
Patching data[:,:,0] = Training data norm
Patching_data[:,:,1] = Training_labels
#Create a training data comprising matching patches (256x256 pixels) of image-
data and training labels
Patched images =
sklearn.feature extraction.image.extract patches 2d(Patching data, patch size =
[256, 256], max patches=12000, random state=None)
#%%
# CREATE THE UNET
# input layer 256x256x1
# encoder depth 4
# 64 filters at the level of the first encoder
def unet(pretrained weights = None,input size=(256,256,1), n class=3):
```

```
inputs = tensorflow.keras.Input(shape=input size)
    conv1 = Conv2D(64, 3, activation = 'relu', dilation rate=2,padding = 'same',
kernel initializer = 'he normal')(inputs)
    conv1 = BatchNormalization()(conv1)
    conv1 = Conv2D(64, 3, activation = 'relu', dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(conv1)
    conv1 = BatchNormalization()(conv1)
    pool1 = MaxPooling2D(pool_size=(2, 2))(conv1)
    conv2 = Conv2D(128, 3, activation = 'relu', dilation rate=2,padding =
'same', kernel initializer = 'he normal')(pool1)
   conv2 = BatchNormalization()(conv2)
    conv2 = Conv2D(128, 3, activation = 'relu', dilation rate=2, padding =
'same', kernel_initializer = 'he_normal')(conv2)
    conv2 = BatchNormalization()(conv2)
    pool2 = MaxPooling2D(pool size=(2, 2))(conv2)
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(pool2)
   conv3 = BatchNormalization()(conv3)
    conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv3)
    conv3 = BatchNormalization()(conv3)
    pool3 = MaxPooling2D(pool size=(2, 2))(conv3)
    conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(pool3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv4)
    conv4 = BatchNormalization()(conv4)
    drop4 = Dropout(0.5)(conv4, training=True)
    pool4 = MaxPooling2D(pool size=(2, 2))(drop4)
    conv5 = Conv2D(1024, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(pool4)
    conv5 = BatchNormalization()(conv5)
    conv5 = Conv2D(1024, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv5)
    conv5 = BatchNormalization()(conv5)
    drop5 = Dropout(0.5)(conv5, training=True)
    up6 = Conv2D(512, 2, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(UpSampling2D(size = (2,2))(drop5))
   merge6 = concatenate([drop4,up6], axis = 3)
    conv6 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(merge6)
    conv6 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv6)
    up7 = Conv2D(256, 2, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(UpSampling2D(size = (2,2))(conv6))
   merge7 = concatenate([conv3,up7], axis = 3)
    conv7 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(merge7)
    conv7 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(conv7)
    up8 = Conv2D(128, 2, activation = 'relu', padding = 'same',
```

kernel_initializer = 'he_normal')(UpSampling2D(size = (2,2))(conv7))

```
merge8 = concatenate([conv2,up8], axis = 3)
   conv8 = Conv2D(128, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he_normal')(merge8)
    conv8 = Conv2D(128, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv8)
    up9 = Conv2D(64, 2, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(UpSampling2D(size = (2,2))(conv8))
   merge9 = concatenate([conv1,up9], axis = 3)
    conv9 = Conv2D(64, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(merge9)
    conv9 = Conv2D(64, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv9)
   conv10 = Conv2D(n class, 1, activation = 'softmax')(conv9)
   model = tensorflow.keras.Model(inputs = inputs, outputs = conv10)
   model.compile(optimizer = Adam(lr = 0.0001),loss =
'sparse categorical crossentropy', metrics = ['sparse categorical accuracy'])
    if(pretrained weights):
     model=keras.models.load model(pretrained weights)
    return model
#%% Split data into image and label data and reshape ready for training
trainX = Patched images[:,:,:,0]
trainX = np.asarray(trainX).reshape((12000, 256, 256,1))
trainY = Patched images[:,:,:,1]
trainY = np.asarray(trainY).reshape((12000, 256, 256, 1))
#Define model for data
model = unet(pretrained weights = None,input size = (256,256,1),n class=3)
# Define augmentation options
def get train augmented(trainX=trainX, trainY=trainY, BATCH SIZE=12):
    aug_X = ImageDataGenerator(rotation_range=360, zoom_range=[1,1],
width_shift_range=[0,0], height_shift_range=[0,0], horizontal_flip=True,
vertical_flip = True, shear_range = 0,fill_mode = "constant",cval=0.0)
    aug Y = ImageDataGenerator(rotation range=360, zoom range=[1,1],
width shift range=[0,0], height_shift_range=[0,0], horizontal_flip=True,
vertical_flip = True, shear_range = 0,fill_mode = "constant",cval=0.0)
    aug X.fit(trainX, augment=True, seed=1)
    aug_Y.fit(trainY, augment=True, seed=1)
    X train augmented = aug X.flow(trainX, batch size=BATCH SIZE, shuffle=True,
seed=1)
```

```
Y_train_augmented = aug_Y.flow(trainY, batch_size=BATCH_SIZE, shuffle=True,
seed=1)
    train generator = zip(X \text{ train augmented}, Y \text{ train augmented})
    for (X train augmented, Y train augmented) in train generator:
        yield (X train augmented, Y train augmented)
#%% Define path to safe network, time stamp network
checkpoint filepath =
'C:/Users/John/20211118 Python biostudies/Python/2D UNET PYTHON/2D PeyersPatch/3
Saved models/'
now = datetime.now()
dt string = now.strftime("%d/%m/%Y%H:%M:%S")
NAME =dt string
Name_file = 'PeyersPatchBiostudies '
checkpoint filepath name = checkpoint filepath + Name file+dt string
checkpoint dir = os.path.dirname(checkpoint filepath name)
#Specify that weights are to be saved
model checkpoint callback = tensorflow.keras.callbacks.ModelCheckpoint(
    filepath=checkpoint dir,
    save weights only=True,
    monitor='val accuracy',
    mode='max',
    save best only=True)
# Specify training options
BATCH SIZE = 12
NUM EPOCHS = 50
#Augment training data
train generator = get train augmented(trainX=trainX, trainY=trainY,
BATCH SIZE=BATCH SIZE)
#Train and save the network
history = model.fit generator(train generator,
steps per epoch=len(trainX)/(BATCH SIZE*2), epochs=NUM EPOCHS,
callbacks=[model checkpoint callback])
#Plot progress # this plots over the view of the data
plt.figure(2)
plt.plot(history.history['sparse categorical accuracy'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.savefig('Training plot.png')
plt.show()
#Save final fully trained model as a .h5 file
model.save("C:/Users/John/20211118 Python biostudies/Python/2D UNET PYTHON/2D Pe
yersPatch/3 Saved models/my h5 model.h5")
checkpoint filepath name h5 = checkpoint filepath + Name file+dt string+'.h5'
model.save(checkpoint filepath name h5)
```

```
% PYTHON SCRIPT: TEST 2D UNET
% All code, image-data and screen-cast tutorial videos available for download at
% the Biostudies database under accession number S-BSST742
# -*- coding: utf-8 -*-
Created on Tue Sep 21 02:46:52 2021
@author: Paul
#Import all required modules
import skimage
from skimage import data
from skimage.filters import threshold otsu
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import sklearn.feature extraction
from sklearn.datasets import load sample image
from sklearn.feature_extraction import image
import matplotlib.cm as cm
import matplotlib.pyplot as plt
import matplotlib.cbook as cbook
from matplotlib.path import Path
from matplotlib.patches import PathPatch
import PIL
import numpy as np
import cv2
import tensorflow
from tensorflow.keras.models import *
from tensorflow.keras.layers import *
from tensorflow.keras.optimizers import *
from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler
from tensorflow.keras import backend as keras
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import sklearn.feature extraction
from sklearn.datasets import load sample image
from sklearn.feature extraction import image
from datetime import datetime
from keras.utils import to categorical
from tensorflow.python.keras.optimizers import *
import skimage.io
#%% Load the reflectance data from the unseen test image
TEST DATA =
skimage.io.imread('C:/Users/John/20211118 Python biostudies/Python/2D UNET PYTHO
N/2D PeyersPatch/1 ImageData/TEST IMAGE.tif');
Test_data = TEST_DATA[0,0,:,:,2]
# Load the mask for the lymphoid tissue region
Mask = TEST DATA[0,0,:,:,3]
thresh = threshold otsu(Test data)
Mask = Mask > thresh
Mask = Mask*1
Mask = np.uint16(Mask)
# Rescale the test data [0 1]
# Mask the rescaled data
Test data = np.multiply(Test data, Mask);
Test data = np.double(Test data);
```
```
Test_data_norm_holder = np.zeros((11247, 7610),np.double)
Test data rescaled = cv2.normalize(Test data, Test data norm holder , 1.0, 0.0,
cv2.NORM MINMAX)
#88
# Define model for testing
def unet(pretrained weights = None, input size=(256,256,1), n class=3):
   inputs = tensorflow.keras.Input(shape=input size)
   conv1 = Conv2D(64, 3, activation = 'relu', dilation_rate=2,padding = 'same',
kernel initializer = 'he_normal')(inputs)
   conv1 = BatchNormalization()(conv1)
   conv1 = Conv2D(64, 3, activation = 'relu', dilation rate=2,padding = 'same',
kernel initializer = 'he normal')(conv1)
   conv1 = BatchNormalization()(conv1)
   pool1 = MaxPooling2D(pool size=(2, 2))(conv1)
   conv2 = Conv2D(128, 3, activation = 'relu', dilation rate=2, padding =
'same', kernel_initializer = 'he_normal')(pool1)
   conv2 = BatchNormalization()(conv2)
   conv2 = Conv2D(128, 3, activation = 'relu', dilation_rate=2, padding =
'same', kernel initializer = 'he normal')(conv2)
   conv2 = BatchNormalization()(conv2)
   pool2 = MaxPooling2D(pool_size=(2, 2))(conv2)
   conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(pool2)
   conv3 = BatchNormalization()(conv3)
   conv3 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv3)
   conv3 = BatchNormalization()(conv3)
   pool3 = MaxPooling2D(pool_size=(2, 2))(conv3)
   conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(pool3)
   conv4 = BatchNormalization()(conv4)
   conv4 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(conv4)
   conv4 = BatchNormalization()(conv4)
   drop4 = Dropout(0.5)(conv4, training=True)
   pool4 = MaxPooling2D(pool size=(2, 2))(drop4)
   conv5 = Conv2D(1024, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(pool4)
   conv5 = BatchNormalization()(conv5)
   conv5 = Conv2D(1024, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv5)
   conv5 = BatchNormalization()(conv5)
   drop5 = Dropout(0.5)(conv5, training=True)
   up6 = Conv2D(512, 2, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(UpSampling2D(size = (2,2))(drop5))
   merge6 = concatenate([drop4,up6], axis = 3)
   conv6 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(merge6)
   conv6 = Conv2D(512, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv6)
```

```
up7 = Conv2D(256, 2, activation = 'relu', padding = 'same',
kernel initializer = 'he_normal')(UpSampling2D(size = (2,2))(conv6))
   merge7 = concatenate([conv3,up7], axis = 3)
   conv7 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(merge7)
   conv7 = Conv2D(256, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(conv7)
   up8 = Conv2D(128, 2, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(UpSampling2D(size = (2,2))(conv7))
   merge8 = concatenate([conv2,up8], axis = 3)
   conv8 = Conv2D(128, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(merge8)
   conv8 = Conv2D(128, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(conv8)
   up9 = Conv2D(64, 2, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(UpSampling2D(size = (2,2))(conv8))
   merge9 = concatenate([conv1,up9], axis = 3)
   conv9 = Conv2D(64, 3, activation = 'relu', padding = 'same',
kernel initializer = 'he normal')(merge9)
   conv9 = Conv2D(64, 3, activation = 'relu', padding = 'same',
kernel_initializer = 'he_normal')(conv9)
   conv10 = Conv2D(n_class, 1, activation = 'softmax')(conv9)
   model = tensorflow.keras.Model(inputs = inputs, outputs = conv10)
   model.compile(optimizer = Adam(lr = 0.0001),loss =
'sparse_categorical_crossentropy', metrics = ['sparse_categorical_accuracy'])
   if(pretrained weights):
     model=keras.models.load model(pretrained weights)
   return model
#%%
# Patch the reflectance data through the network
# Set the patch sizes to be passed to the net
patchSize = [256, 256]
Test data rescaled =
np.asarray(Test data rescaled).reshape((Test data rescaled.shape[0],
Test data rescaled.shape[1],1))
# Segment blockwise then reassemble in full
# Define image dimensions
[height,width,nChannel] = np.shape(Test data rescaled)
```

```
patch = np.zeros([patchSize[0], patchSize[1],
nChannel],dtype=Test data rescaled.dtype);
# Pad image to have dimensions as multiples of patchSize
padSize = np.empty([1, 2])
padSize[0,0] = patchSize[0] - np.remainder(height, patchSize[0]);
padSize[0,1] = patchSize[1] - np.remainder(width, patchSize[1]);
Starting_shape = np.shape(Test_data_rescaled)
a1 = Starting_shape[0]+padSize[0,0]
a2 = Starting shape[1]+padSize[0,1]
im_pad = np.zeros([int(a1),int(a2)])
im pad[:Test data rescaled.shape[0],:Test data rescaled.shape[1]] =
Test_data_rescaled[:,:,0]
im pad = np.asarray(im pad).reshape((im pad.shape[0], im pad.shape[1],1))
[height_pad, width_pad, nChannel_pad] = np.shape(im_pad);
# Preallocate some matrices to receive the network outputs
out Uncertainty Scores = np.zeros([np.shape(im pad)[0], np.shape(im pad)[1]],
'double');
out_Pmap_cat1 = np.zeros([np.shape(im_pad)[0], np.shape(im_pad)[1]], 'double');
out Pmap cat2 = np.zeros([np.shape(im pad)[0], np.shape(im pad)[1]],
                                                                      'double');
out Pmap cat3 = np.zeros([np.shape(im_pad)[0], np.shape(im_pad)[1]], 'double');
# Load a pretrained network
pretrained model = unet()
pretrained model.load weights('C:/Users/John/20211118 Python biostudies/Python/2
D_UNET_PYTHON/2D_PeyersPatch/3_Saved_models/my_h5_model_fully_trained.h5')
# Loop through blocks of 'patchSize'
for loop in range(1, int(height_pad),int(patchSize[0])):
    print(loop)
    for j in range(1, int(width pad), int(patchSize[1])):
        for p in range(1,(nChannel+1)):
            PP3 = np.empty([1,256,256,1])
            im pad touse =im pad[(loop-1):((loop-1)+patchSize[0]),(j-1):((j-
1)+(patchSize[1]))]
            patch[:,:,0] = np.squeeze(im_pad_touse,axis=None)
            PP3[0,:,:,:]=patch
            # deploy net
            predictions = pretrained model.predict(PP3)
            scores = np.max(predictions,axis=3)
            predictions.argmax(axis=3)
            out Uncertainty Scores ((loop-1): (loop+patchSize[0]-1), (j-
1):(j+patchSize[1]-1)] = scores;
            out Pmap cat1[(loop-1):(loop+patchSize[0]-1), (j-1):(j+patchSize[1]-
1)] = predictions[0,:,:,0];
            out_Pmap_cat2[(loop-1):(loop+patchSize[0]-1), (j-1):(j+patchSize[1]-
1)] = predictions[0,:,:,1];
            out_Pmap_cat3[(loop-1):(loop+patchSize[0]-1), (j-1):(j+patchSize[1]-
1)] = predictions[0,:,:,2];
```

```
out Uncertainty Scores = out Uncertainty Scores[0:height, 0:width];
out Pmap cat1 = out_Pmap_cat1[0:height, 0:width];
out Pmap cat2 = out_Pmap_cat2[0:height, 0:width];
out Pmap cat3 = out Pmap cat3[0:height, 0:width];
image_to_plot = np.zeros((11247, 7610,3))
image to plot[:,:,0] = out Pmap cat1
image to plot[:,:,1] = out Pmap cat2
image_to_plot[:,:,2] = out_Pmap_cat3
#%% visualise network outputs
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, sharey=True)
fig.set size inches(14, 12)
im = ax1.imshow(out Pmap cat1)
ax1.title.set text('Background/Other')
im = ax2.imshow(out Pmap cat2)
ax2.title.set text('LF-Actin')
im = ax3.imshow(out Pmap cat3)
ax3.title.set text('LF-Nuclei')
im = ax4.imshow(out_Uncertainty_Scores)
ax4.title.set_text('Uncertainty')
plt.savefig('Predictions.png')
#%% Map to 16-bit and save for loading into CellProfiler
# Background/other
data_norm_holder = np.zeros((11247, 7610),np.uint16)
uil6 PMAP cat1 xy = cv2.normalize(out Pmap cat1, data norm holder , 65535.0,
0.0, cv2.NORM MINMAX)
img_data2 = (ui16_PMAP_cat1_xy).astype(dtype=np.uint16)
rawtiff=PIL.Image.fromarray(img_data2)
rawtiff.save('Ch_7_Im_001'+'.tiff')
# LF-Actin
data norm holder = np.zeros((11247, 7610), np.uint16)
uil6 PMAP cat2 xy = cv2.normalize(out Pmap cat2, data norm holder , 65535.0,
0.0, cv2.NORM MINMAX)
img_data2 = (ui16_PMAP_cat2_xy).astype(dtype=np.uint16)
rawtiff=PIL.Image.fromarray(img_data2)
rawtiff.save('Ch 6 Im 001'+'.tiff')
# LF-Nuclei
data norm holder = np.zeros((11247, 7610),np.uint16)
uil6_PMAP_cat3_xy = cv2.normalize(out_Pmap_cat3, data_norm_holder , 65535.0,
0.0, cv2.NORM_MINMAX)
img data2 = (ui16 PMAP cat3 xy).astype(dtype=np.uint16)
rawtiff=PIL.Image.fromarray(img data2)
rawtiff.save('Ch 5 Im 001'+'.tiff')
```

% PYTHON SCRIPT: TRAIN 3D UNET % All code, image-data and screen-cast tutorial videos available for download at % the Biostudies database under accession number S-BSST742 # -*- coding: utf-8 -*-Created on Mon Jun 14 15:21:44 2021 @author: Paul from PIL import Image import javabridge import bioformats javabridge.start_vm(class_path=bioformats.JARS) import cv2 import matplotlib.pyplot as plt import matplotlib.pyplot as plt import skimage from skimage import data from skimage.filters import threshold otsu import os import numpy as np import skimage.transform as trans import numpy as np import tensorflow from tensorflow.keras.models import * from tensorflow.keras.layers import * from tensorflow.keras.optimizers import * from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler from tensorflow.keras import backend as keras from tensorflow.keras.preprocessing.image import ImageDataGenerator import sklearn.feature extraction from sklearn.datasets import load sample image from sklearn.feature extraction import image from datetime import datetime from keras.utils import to categorical from tensorflow.python.keras.optimizers import * import random import matplotlib.pyplot as plt from mpl toolkits.mplot3d.art3d import Poly3DCollection import numpy as np from skimage import exposure, io, util #%% Specify directory containing training data Training data directory = 'C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHON/3D UNET PYTHON/1 ImageData/TRAIN/DATA/' # Specify channel number in training data that contains reflectance information RL training directory = 3; # Specify directory containing training labels Training_labels_path = 'C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHON/3D UNET PYTHON/1 ImageData/TRAIN/LABELS/' ;

```
# COMMIT TRAINING DATA TO DIRECTORY rescaled [0 1] IN MAT FORMAT
MYDIR
='C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHON/3D UNET PYTHON/
1 ImageData/TRAIN/LABELS/mat training data/'
CHECK FOLDER = os.path.isdir(MYDIR)
# If folder doesn't exist, then create it.
if not CHECK FOLDER:
    os.makedirs(MYDIR)
Training labels holder =
skimage.io.imread('C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHO
N/3D UNET PYTHON/1 ImageData/TRAIN/LABELS/LABELS 001.tif') ;
Training labels = np.where(Training labels holder == 3, 0,
Training labels holder)
#%%
for i in range(0,2):
    loop = i+1
    counter = '00'+str(loop)
    # find metadata describing file
    TOTAL path = Training data directory+'TRAIN '+counter+'.tif'
    reader = bioformats.ImageReader(TOTAL path)
    NUMBERIMAGES = reader.rdr.getSeriesCount(TOTAL path)
    number of channels = reader.rdr.getSizeC(TOTAL path)
    Xlength = reader.rdr.getSizeX(TOTAL path)
    Ylength = reader.rdr.getSizeY(TOTAL path)
    stackSizeZ = reader.rdr.getSizeZ(TOTAL path)# number of Z slices
    channel zimage = np.zeros((Xlength,Ylength,stackSizeZ))
    # Load reflectance information
    for zplane in range(1,stackSizeZ+1):
        channel zimage[:,:,(zplane-1)] = bioformats.load image(TOTAL path,z =
(zplane-1), c= (RL training directory-1))
    IM DATA = channel zimage
    IM DATA = np.double(IM DATA);
    DIM = IM DATA.shape
    IM DATA norm holder = np.zeros((DIM[0],DIM[1],DIM[2]),np.double)
    IM DATA norm = cv2.normalize(IM DATA, IM DATA norm holder, 1.0, 0.0,
cv2.NORM MINMAX)
    pw = os.getcwd()
    FILENAME = pw+'\\npy training data\\'+'TRAIN DATA '+counter
    FOLDER CHECK = pw+'\\npy training data\\'
    CHECK FOLDER = os.path.isdir(FOLDER CHECK)
    if not CHECK FOLDER:
        os.makedirs(FOLDER CHECK)
    np.save(FILENAME,IM DATA norm)
```

```
pw = os.qetcwd()
#%% COMMIT TRAINING LABELS TO DIRECTORY IN MAT FORMAT
FOLDER NAME = pw+'\\npy training labels\\'
CHECK FOLDER2 = os.path.isdir(FOLDER NAME)
if not CHECK FOLDER2:
    os.makedirs(FOLDER NAME)
for i in range(0,2):
    loop = i + 1
    counter = '00'+str(loop)
    label zimage = [];
    label zimage = np.zeros((Xlength,Ylength,stackSizeZ))
    # load label information
    for zplane in range(1,stackSizeZ+1):
        TOTAL path = Training labels path+'LABELS '+counter+'.tif'
        iplane = bioformats.load image(TOTAL path,z=(zplane-1), c=0,rescale =
False )
        label zimage[:,:,(zplane-1)] = iplane
    IM LABELS = np.double(label zimage);
    DIM = IM LABELS.shape
    IM LABELS norm holder = np.zeros((DIM[0],DIM[1],DIM[2]),np.double)
    IM LABELS norm holder[:,:,:] = label zimage[:,:,:]
    pw = os.getcwd()
    FILENAME = pw+'\\npy_training_labels\\'+'TRAIN_LABELS_'+counter
    FOLDER CHECK = pw+'\\npy training labels\\'
    CHECK FOLDER = os.path.isdir(FOLDER CHECK)
    if not CHECK FOLDER:
        os.makedirs(FOLDER_CHECK)
   np.save(FILENAME, IM_LABELS_norm_holder)
#%%
NUMBER batch = 1
del
     channel zimage, IM LABELS, IM DATA, label zimage
# Prepare to extract patches
patchPerImage = 375;
xrand = random.sample(range(0,959), patchPerImage)
yrand = random.sample(range(0,959), patchPerImage)
zrand = np.random.choice(78,patchPerImage, replace=True)
# Define random patch extraction and augmentarion of patches
def get_train_patched(xrand, yrand, zrand, BATCH_SIZE, batch_number =
NUMBER batch, trainXm=IM DATA norm, trainYm=IM LABELS norm holder):
    patched data image = np.zeros((BATCH SIZE,64,64,32))
    patched data labels = np.zeros((BATCH SIZE,64,64,32))
    for i in range(0,BATCH SIZE):
        val = (i + batch number) - 1;
        patched_data_image[i,:,:,:] =
trainXm[(xrand[val]):(xrand[val]+64),(yrand[val]):(yrand[val]+64),(zrand[val]):(
zrand[val]+32)];
```

```
patched_data_labels[i,:,:,:] =
trainYm[(xrand[val]):(xrand[val]+64),(yrand[val]):(yrand[val]+64),(zrand[val]):(
zrand[val]+32)];
   return patched data image, patched data labels
def get train augmented(trainXm,trainYm ,BATCH SIZE,batch number =
NUMBER batch):
   for i in range(0,BATCH SIZE):
       rand number = random.sample(range(BATCH SIZE), 1)
       my arr = [np.rot90, np.flipud, np.fliplr]
       if int(rand number[0])<3:
          trainXm[i,:,:,:] = my arr[int(rand number[0])](trainXm[i,:,:,:])
          trainYm[i,:,:,:] = my_arr[int(rand_number[0])](trainYm[i,:,:,:])
       elif int(rand_number[0]) ==3:
          trainXm[i,:,:,:] = np.fliplr(trainXm[i,:,:,:])
          trainYm[i,:,:,:] = np.fliplr(trainYm[i,:,:,:])
          trainXm[i,:,:,:] = np.rot90(trainXm[i,:,:,:])
          trainYm[i,:,:,:] = np.rot90(trainYm[i,:,:,:])
       else:
          trainXm[i,:,:,:] = trainXm[i,:,:,:]
          trainYm[i,:,:,:] = trainYm[i,:,:,:]
       return trainXm, trainYm
#88
# CREATE THE 3D UNET
# input layer 64x64x32x1
# encoder depth 4
# 32 filters at the level of the first encoder
def unet3(pretrained weights = None, input size= (64,64,32,1), n class=4):
   inputs = tensorflow.keras.Input(shape=input size)
   conv1 = Conv3D(32, kernel size=(3, 3, 3), dilation rate=2, padding = 'same',
kernel initializer = 'he normal')(inputs)
   conv1 = BatchNormalization()(conv1)
   conv1 = Activation('relu')(conv1)
   conv1 = Conv3D(64, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(conv1)
   conv1 = BatchNormalization()(conv1)
   up1 = Activation('relu')(conv1)
   pool1 = MaxPooling3D(pool size=(2,2,2), strides = (2,2,2))(up1)
   conv2 = Conv3D(64, kernel size=(3, 3, 3), dilation rate=2,padding = 'same',
kernel initializer = 'he normal')(pool1)
   conv2 = BatchNormalization()(conv2)
   conv2 = Activation('relu')(conv2)
   conv2 = Conv3D(128, kernel size=(3, 3, 3), dilation rate=2,padding = 'same',
kernel_initializer = 'he_normal')(up1)
   conv2 = BatchNormalization()(conv2)
   up2 = Activation('relu')(conv2)
   pool2 = MaxPooling3D(pool size=(2,2,2),strides = (2,2,2))(up2)
```

```
conv3 = Conv3D(128, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(pool2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Activation('relu')(conv3)
    conv3 = Conv3D(256, kernel size=(3, 3, 3), dilation rate=2, padding = 'same',
kernel initializer = 'he normal')(conv3)
   conv3 = BatchNormalization()(conv3)
    up3 = Activation('relu')(conv3)
    pool3 = MaxPooling3D(pool_size=(2,2,2),strides = (2,2,2))(up3)
    conv4 = Conv3D(256, kernel size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he normal')(pool3)
   conv4 = BatchNormalization()(conv4)
    conv4 = Activation('relu')(conv4)
    conv4 = Conv3D(512, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he normal')(conv4)
    conv4 = BatchNormalization()(conv4)
    up4 = Activation('relu')(conv4)
    pool4 = MaxPooling3D(pool size=(2,2,2),strides = (2,2,2))(up4)
    conv5 = Conv3D(512, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he normal')(pool4)
   conv5 = BatchNormalization()(conv5)
    conv5 = Activation('relu')(conv5)
    conv5 = Conv3D(1024, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he normal')(conv5)
    conv5 = BatchNormalization()(conv5)
    conv5 = Activation('relu')(conv5)
    drop5 = Conv3DTranspose(1024, kernel_size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he normal')(conv5)
   merge6 = concatenate([drop5,up4], axis = 4)
    conv6 = Conv3D(512, kernel size=(3, 3, 3), dilation rate=2, padding =
'same', kernel initializer = 'he normal')(merge6)
    conv6 = BatchNormalization()(conv6)
    conv6 = Activation('relu')(conv6)
    conv6 = Conv3D(512, kernel size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he normal')(conv6)
   conv6 = BatchNormalization()(conv6)
    conv6 = Activation('relu')(conv6)
    drop6 = Conv3DTranspose(512, kernel_size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he normal')(conv6)
   merge7 = concatenate([drop6,up3], axis = 4)
    conv7 = Conv3D(256, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(merge7)
    conv7 = BatchNormalization()(conv7)
    conv7 = Activation('relu')(conv7)
    conv7 = Conv3D(256, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(conv7)
   conv7 = BatchNormalization()(conv7)
    conv7 = Activation('relu')(conv7)
    drop7 = Conv3DTranspose(256, kernel size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he_normal')(conv7)
   merge8 = concatenate([drop7,up2], axis = 4)
    conv8 = Conv3D(128, kernel size=(3, 3, 3), dilation rate=2, padding =
'same', kernel initializer = 'he normal')(merge8)
```

```
conv8 = BatchNormalization()(conv8)
   conv8 = Activation('relu')(conv8)
    conv8 = Conv3D(128, kernel_size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he normal')(conv8)
   conv8 = BatchNormalization()(conv8)
    conv8 = Activation('relu')(conv8)
    drop8 = Conv3DTranspose(128, kernel size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation_rate=2,padding = 'same', kernel initializer =
'he_normal')(conv8)
    conv8 = BatchNormalization()(drop8)
    conv8 = Activation('relu')(conv8)
    drop8 = Conv3DTranspose(128, kernel_size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he_normal')(conv7)
   merge9 = concatenate([drop8,up1], axis = 4)
    conv9 = Conv3D(64, kernel size=(3, 3, 3), dilation rate=2,padding = 'same',
kernel_initializer = 'he_normal')(merge9)
   conv9 = BatchNormalization()(conv9)
    conv9 = Activation('relu')(conv9)
   conv9 = Conv3D(64, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(conv9)
   conv9 = BatchNormalization()(conv9)
   conv9 = Activation('relu')(conv9)
   conv9 = Conv3D(n class, kernel size=(1, 1, 1), activation = 'softmax')(conv9)
   model = tensorflow.keras.Model(inputs = inputs, outputs = conv9)
   model.compile(optimizer = Adam(lr = 0.0005),loss =
'sparse_categorical_crossentropy', metrics = ['sparse_categorical_accuracy'])
    if(pretrained weights):
     model=keras.models.load model(pretrained weights)
    return model
#%% Specify location to save the network
checkpoint filepath =
'C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHON/3D UNET PYTHON/3
Saved models/'
now = datetime.now()
dt string = now.strftime("%d/%m/%Y%H:%M:%S")
NAME =dt string
Name_file = 'PeyersPatchBiostudies '
checkpoint_filepath_name = checkpoint_filepath + Name_file+dt_string
checkpoint dir = os.path.dirname(checkpoint filepath name)
model checkpoint callback = tensorflow.keras.callbacks.ModelCheckpoint(
    filepath=checkpoint dir,
    save_weights_only=True,
   monitor='val accuracy',
   mode='max',
    save best only=True)
#Specify training options
```

```
BATCH SIZE = 8
NUM EPOCHS = 50
patchPerImage = 375;
steps = np.floor(patchPerImage/8);
NUMBER batch = 1;
start = 0;
finish = 8;
storeX = np.zeros((375, 64, 64, 32, 1))
storeY = np.zeros((375, 64, 64, 32, 1))
#Extract patches
for i in range(1, int(steps)):
    [patched data image, patched data labels]=get train patched(xrand, yrand,
zrand, BATCH SIZE, batch number = NUMBER batch, trainXm=IM DATA norm,
trainYm=IM LABELS norm holder)
    [trainXmgg, trainYmgg] =
get train augmented(BATCH SIZE=8, trainXm=patched data image,
trainYm=patched data labels, batch number = NUMBER batch)
    trainYmgg = trainYmgg.reshape((8,64,64,32,1))
    trainXmgg = trainXmgg.reshape((8,64,64,32,1))
    trainXmgg = trainXmgg.astype('float32')
    trainYmgg = trainYmgg.astype('float32')
    storeX[start:finish,:,:,:,:] = trainXmgg
    storeY[start:finish,:,:,:,:] = trainYmgg
    start = finish
    finish = start+8
    NUMBER batch = NUMBER batch+1
[patched data image, patched data labels]=get train patched(xrand, yrand, zrand,
BATCH SIZE=(patchPerImage-finish), batch number = NUMBER batch,
trainXm=IM DATA norm, trainYm=IM LABELS norm holder)
[trainXmgg, trainYmgg] = get train augmented(BATCH SIZE=(patchPerImage-
finish),trainXm=patched data image, trainYm=patched data labels, batch number =
NUMBER batch)
trainYmgg = trainYmgg.reshape((patchPerImage-finish,64,64,32,1))
trainXmgg = trainXmgg.reshape((patchPerImage-finish,64,64,32,1))
trainXmgg = trainXmgg.astype('float32')
trainYmgg = trainYmgg.astype('float32')
storeX[finish:patchPerImage,:,:,:,:] = trainXmgg
storeY[finish:patchPerImage,:,:,:,:] = trainYmgg
model = unet3(pretrained weights = None, input size =
(64,64,32,1),n class=4)#change to 3
```

```
#%% Train the model
history = model.fit(
    storeX,
    storeY,
    batch_size = BATCH_SIZE,
    epochs = NUM_EPOCHS,callbacks=[model_checkpoint_callback])
plt.plot(history.history['sparse_categorical_accuracy'])
plt.title('Model accuracy')
plt.ylabel('accuracy')
plt.ylabel('accuracy')
plt.legend(['train', 'test'], loc='upper left')
plt.savefig('Training_plot.png')
plt.show()
model_cave("C:(Users/John/20211119_Puthen_biestudies/Puthen/3D)
```

```
model.save("C:/Users/John/20211118_Python_biostudies/Python/3D_UNET_PYTHON/3D_UN
ET PYTHON/3 Saved models/my h53D model.h5")
```

% PYTHON SCRIPT: TEST 3D UNET % All code, image-data and screen-cast tutorial videos available for download at % the Biostudies database under accession number S-BSST742 # -*- coding: utf-8 -*-Created on Mon Sep 20 16:37:00 2021 @author: Paul from PIL import Image import javabridge import bioformats javabridge.start vm(class path=bioformats.JARS) import cv2 import matplotlib.pyplot as plt import matplotlib.pyplot as plt import skimage from skimage import data from skimage.filters import threshold_otsu import os import numpy as np import skimage.transform as trans import numpy as np import tensorflow from tensorflow.keras.models import * from tensorflow.keras.layers import * from tensorflow.keras.optimizers import * from tensorflow.keras.callbacks import ModelCheckpoint, LearningRateScheduler from tensorflow.keras import backend as keras from tensorflow.keras.preprocessing.image import ImageDataGenerator import sklearn.feature extraction from sklearn.datasets import load sample image from sklearn.feature extraction import image from datetime import datetime from keras.utils import to categorical from tensorflow.python.keras.optimizers import * import random import matplotlib.pyplot as plt from mpl toolkits.mplot3d.art3d import Poly3DCollection import numpy as np from skimage import exposure, io, util from skimage import data from tensorflow.keras.preprocessing.image import ImageDataGenerator import sklearn.feature extraction from sklearn.datasets import load sample image from sklearn.feature extraction import image import matplotlib.cm as cm import matplotlib.cbook as cbook from matplotlib.path import Path from matplotlib.patches import PathPatch import math import json import numpy as np import PIL import imageio

```
#%% Specify directory containing test data
Test data directory =
'C:/Users/John/20211118 Python biostudies/Python/3D UNET PYTHON/3D UNET PYTHON/1
ImageData/TEST/'
# Specify channel number in test data that contains reflectance information
RL channel number = 2;
#COMMIT REFLECTANCE TEST DATA TO DIRECTORY RESCALED [0 1] IN MAT FORMAT
MYDIR
='C:/Users/John/20211118_Python_biostudies/Python/3D UNET PYTHON/3D UNET PYTHON/
4 TEST Unet/npy test data/'
CHECK FOLDER = os.path.isdir(MYDIR)
# If folder doesn't exist, then create it.
if not CHECK FOLDER:
    os.makedirs(MYDIR)
for i in range(0,1):
    loop = i+1
   counter = '00'+str(loop)
   TOTAL path = Test data directory+'TEST '+counter+'.tif'
   # find metadata describing file
   reader = bioformats.ImageReader(TOTAL path)
   NUMBERIMAGES = reader.rdr.getSeriesCount(TOTAL path)
   number of channels = reader.rdr.getSizeC(TOTAL path)
   Xlength = reader.rdr.getSizeX(TOTAL path)
    Ylength = reader.rdr.getSizeY(TOTAL path)
    stackSizeZ = reader.rdr.getSizeZ(TOTAL path)
   channel zimage = np.zeros((Xlength,Ylength,stackSizeZ))
    # Load reflectance information
    for zplane in range(1,stackSizeZ+1):
        channel zimage[:,:,(zplane-1)] = bioformats.load image(TOTAL path,z =
(zplane-1), c= (RL channel number-1))
    IM_DATA = channel_zimage
    IM DATA = np.double(IM DATA);
    DIM = IM DATA.shape
    IM DATA norm holder = np.zeros((DIM[0],DIM[1],DIM[2]),np.double)
    IM DATA norm = cv2.normalize(IM DATA, IM DATA norm holder, 1.0, 0.0,
cv2.NORM MINMAX)
    pw = os.getcwd()
    FILENAME = pw+'\\npy_test_data\\'+'TEST_DATA '+counter
    FOLDER_CHECK = pw+'\\npy_test_data\\'
    CHECK FOLDER = os.path.isdir(FOLDER CHECK)
    if not CHECK FOLDER:
        os.makedirs(FOLDER CHECK)
   np.save(FILENAME,IM_DATA_norm)
```

```
#88
#Redefine network
def unet3(pretrained weights = None, input size= (64,64,32,1), n class=4):
    inputs = tensorflow.keras.Input(shape=input size)
    conv1 = Conv3D(32, kernel_size=(3, 3, 3), dilation_rate=2, padding = 'same',
kernel initializer = 'he normal')(inputs)
   conv1 = BatchNormalization()(conv1)
    conv1 = Activation('relu')(conv1)
    conv1 = Conv3D(64, kernel size=(3, 3, 3), dilation rate=2,padding = 'same',
kernel initializer = 'he normal')(conv1)
   conv1 = BatchNormalization()(conv1)
    up1 = Activation('relu')(conv1)
    pool1 = MaxPooling3D(pool_size=(2,2,2),strides = (2,2,2))(up1)
    conv2 = Conv3D(64, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(pool1)
   conv2 = BatchNormalization()(conv2)
   conv2 = Activation('relu')(conv2)
    conv2 = Conv3D(128, kernel_size=(3, 3, 3), dilation rate=2,padding = 'same',
kernel initializer = 'he normal')(up1)
   conv2 = BatchNormalization()(conv2)
    up2 = Activation('relu')(conv2)
   pool2 = MaxPooling3D(pool size=(2,2,2),strides = (2,2,2))(up2)
    conv3 = Conv3D(128, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel_initializer = 'he_normal')(pool2)
    conv3 = BatchNormalization()(conv3)
    conv3 = Activation('relu')(conv3)
    conv3 = Conv3D(256, kernel size=(3, 3, 3), dilation rate=2,padding = 'same',
kernel initializer = 'he normal')(conv3)
   conv3 = BatchNormalization()(conv3)
    up3 = Activation('relu')(conv3)
   pool3 = MaxPooling3D(pool size=(2,2,2), strides = (2,2,2))(up3)
   conv4 = Conv3D(256, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he normal')(pool3)
    conv4 = BatchNormalization()(conv4)
    conv4 = Activation('relu')(conv4)
    conv4 = Conv3D(512, kernel size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he normal')(conv4)
   conv4 = BatchNormalization()(conv4)
    up4 = Activation('relu')(conv4)
   pool4 = MaxPooling3D(pool_size=(2,2,2),strides = (2,2,2))(up4)
   conv5 = Conv3D(512, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he_normal')(pool4)
   conv5 = BatchNormalization()(conv5)
   conv5 = Activation('relu')(conv5)
    conv5 = Conv3D(1024, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he normal')(conv5)
    conv5 = BatchNormalization()(conv5)
    conv5 = Activation('relu')(conv5)
    drop5 = Conv3DTranspose(1024, kernel_size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation_rate=2,padding = 'same', kernel_initializer =
'he normal')(conv5)
   merge6 = concatenate([drop5,up4], axis = 4)
    conv6 = Conv3D(512, kernel size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he normal')(merge6)
   conv6 = BatchNormalization()(conv6)
    conv6 = Activation('relu')(conv6)
    conv6 = Conv3D(512, kernel size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he normal')(conv6)
```

```
conv6 = BatchNormalization()(conv6)
   conv6 = Activation('relu')(conv6)
   drop6 = Conv3DTranspose(512, kernel_size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation_rate=2, padding = 'same', kernel_initializer =
'he normal')(conv6)
   merge7 = concatenate([drop6,up3], axis = 4)
   conv7 = Conv3D(256, kernel size=(3, 3, 3), dilation rate=2, padding = 'same',
kernel_initializer = 'he_normal')(merge7)
   conv7 = BatchNormalization()(conv7)
   conv7 = Activation('relu')(conv7)
   conv7 = Conv3D(256, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(conv7)
   conv7 = BatchNormalization()(conv7)
   conv7 = Activation('relu')(conv7)
   drop7 = Conv3DTranspose(256, kernel size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he normal')(conv7)
   merge8 = concatenate([drop7,up2], axis = 4)
   conv8 = Conv3D(128, kernel_size=(3, 3, 3), dilation_rate=2,padding =
'same', kernel initializer = 'he normal')(merge8)
   conv8 = BatchNormalization()(conv8)
   conv8 = Activation('relu')(conv8)
   conv8 = Conv3D(128, kernel size=(3, 3, 3), dilation rate=2,padding =
'same', kernel initializer = 'he_normal')(conv8)
   conv8 = BatchNormalization()(conv8)
   conv8 = Activation('relu')(conv8)
   drop8 = Conv3DTranspose(128, kernel size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he normal')(conv8)
   conv8 = BatchNormalization()(drop8)
   conv8 = Activation('relu')(conv8)
    drop8 = Conv3DTranspose(128, kernel_size=(2, 2, 2), strides=(2, 2, 2),
activation = 'relu', dilation rate=2, padding = 'same', kernel initializer =
'he normal')(conv7)
   merge9 = concatenate([drop8,up1], axis = 4)
   conv9 = Conv3D(64, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(merge9)
   conv9 = BatchNormalization()(conv9)
   conv9 = Activation('relu')(conv9)
   conv9 = Conv3D(64, kernel_size=(3, 3, 3), dilation_rate=2,padding = 'same',
kernel initializer = 'he normal')(conv9)
   conv9 = BatchNormalization()(conv9)
   conv9 = Activation('relu')(conv9)
   conv9 = Conv3D(n class, kernel_size=(1, 1, 1), activation = 'softmax')(conv9)
```

```
model = tensorflow.keras.Model(inputs = inputs, outputs = conv9)
model.compile(optimizer = Adam(lr = 0.0005),loss =
'sparse categorical crossentropy', metrics = ['sparse categorical accuracy'])
```

```
if(pretrained_weights):
     model=keras.models.load model(pretrained weights)
   return model
#%% load the trained 3D Unet network
#model.load weights("C:/Users/John/20211118 Python biostudies/Python/3D UNET PYT
HON/3D_UNET_PYTHON/3_Saved_models/my_h53D_model.h5")
pretrained model = unet3()
pretrained model.load weights("C:/Users/John/20211118 Python biostudies/Python/3
D UNET PYTHON/3D UNET PYTHON/3 Saved models/my h53D model updated.h5")
#%% Patch the reflectance data through the network
# Set the patch sizes to be passed to the net
patchSize = [64, 64, 32]
# Segment blockwise then reassemble in full
# Define image dimensions
Shape overall = np.shape(IM DATA norm)
height = Shape overall[0]
width = Shape overall[1]
depth = Shape overall[2]
if len(Shape overall) == 3:
   nChannel = 1
else:
   nChannel = Shape overall[3]
patch = np.zeros([patchSize[0],patchSize[1],patchSize[2],
nChannel],dtype=IM DATA norm.dtype);
number_of_height_patches = math.ceil(height/patchSize[0]);
number of width patches = math.ceil(width/patchSize[1]);
number_of_depth_patches = math.ceil(depth/patchSize[2]);
#Pad image to have dimensions as multiples of patchSize
height_pad = number_of_height_patches*patchSize[0];
width pad = number of width patches*patchSize[1];
depth pad = number of depth patches*patchSize[2];
# Amount to pad different dimensions of image
padSize = np.empty([1, 3])
# Pad the image by correct amounts
padSize[0,0] = height pad;
padSize[0,1] = width_pad;
padSize[0,2] = depth_pad;
Starting_shape = np.shape(IM_DATA_norm)
a1 = padSize[0,0]
a2 = padSize[0,1]
a3 = padSize[0,2]
im_pad = np.zeros([int(a1),int(a2),int(a3)])
im_pad[:IM_DATA_norm.shape[0],:IM_DATA_norm.shape[1],:IM_DATA_norm.shape[2]] =
IM DATA norm[:,:,:]
```

```
im_pad = np.asarray(im_pad).reshape((im_pad.shape[0],
im pad.shape[1],im pad.shape[2],1))
[height pad, width pad, depth pad, nChannel pad] = np.shape(im pad);
#Preallocate some matrices to catch the probability maps
out_Uncertainty_Scores = np.zeros([height_pad, width_pad, depth_pad],'double');
out scores from network all classes = np.zeros([height pad, width pad,
depth_pad, 4], 'double');
#Loop through blocks of 'patchSize'
for loop height in range(1,number of height patches+1):
    for loop width in range(1, number of width patches+1):
        for loop depth in range(1, number of depth patches+1):
            PP3 = np.empty([1, 64, 64, 32, 1])
            start height position=(loop height-1)*patchSize[0];
            end height position=loop height*patchSize[0];
            start width position=(loop width-1)*patchSize[1];
            end_width_position=loop_width*patchSize[1];
            start depth position=(loop depth-1)*patchSize[2];
            end depth position=loop depth*patchSize[2];
patch to deploy=im pad[start height position:end height position,start width pos
ition:end width position,start depth position:end depth position,:];
            PP3[0,:,:,:]=patch to deploy
            # deploy net
            predictions = pretrained model.predict(PP3)
            scores = np.max(predictions,axis=4)
            predictions.argmax(axis=3)
```

out_Uncertainty_Scores[start_height_position:end_height_position,start_width_pos ition:end_width_position,start_depth_position:end_depth_position]=scores

out_scores_from_network_all_classes[start_height_position:end_height_position,st
art_width_position:end_width_position,start_depth_position:end_depth_position,0]
=predictions[0,:,:,:,0];

out_scores_from_network_all_classes[start_height_position:end_height_position,st
art_width_position:end_width_position,start_depth_position:end_depth_position,1]
=predictions[0,:,:,:,1];

```
out_scores_from_network_all_classes[start_height_position:end_height_position,st
art_width_position:end_width_position,start_depth_position:end_depth_position,2]
=predictions[0,:,:,:,2];
```

```
out_scores_from_network_all_classes[start_height_position:end_height_position,st
art_width_position:end_width_position,start_depth_position:end_depth_position,3]
=predictions[0,:,:,:,3];
```

```
#Script is general for n classes to this point
out Pmap cat1 = out scores from network all classes[:,:,:,0];
out Pmap cat2 = out scores from network all classes[:,:,1];
out Pmap cat3 = out scores from network all classes[:,:,2];
out Uncertainty Scores = out Uncertainty Scores[0:height, 0:width, 0:depth];
#Remove padding from probability maps
out Pmap cat1 = out Pmap cat1[0:height, 0:width, 0:depth];
out Pmap cat2 = out Pmap cat2[0:height, 0:width, 0:depth];
out_Pmap_cat3 = out_Pmap_cat3[0:height, 0:width, 0:depth];
#%%visualise network outputs
fig, (ax1, ax2, ax3, ax4) = plt.subplots(1, 4, sharey=True)
fig.set size inches(14, 12)
im = ax1.imshow(out Pmap cat1[:,:,59])
ax1.title.set text('Background/Other')
im = ax2.imshow(out_Pmap_cat2[:,:,59])
ax2.title.set_text('LF-nuclei')
im = ax3.imshow(out Pmap cat3[:,:,59])
ax3.title.set text('LF-actin')
im = ax4.imshow(out Uncertainty Scores[:,:,59])
ax4.title.set_text('Uncertainty')
plt.savefig('Predictions.png')
#%%Map to 16-bit and save for loading into CellProfiler
data_norm_holder = np.zeros((1024, 1024, 107), np.uint16)
uil6_PMAP_cat1_xy = cv2.normalize(out_Pmap_cat1, data_norm_holder , 65535.0,
0.0, cv2.NORM MINMAX)
data_norm_holder = np.zeros((1024, 1024, 107), np.uint16)
uil6 PMAP cat2 xy = cv2.normalize(out Pmap cat2, data norm holder , 65535.0,
0.0, cv2.NORM MINMAX)
data_norm_holder = np.zeros((1024, 1024, 107), np.uint16)
uil6 PMAP cat3 xy = cv2.normalize(out Pmap cat3, data norm holder , 65535.0,
0.0, cv2.NORM MINMAX)
data to save = np.zeros((1024,1024,107,3))
data to save[:,:,:,0] = ui16 PMAP cat1 xy
data to save[:,:,:,1] = ui16 PMAP cat2 xy
data_to_save[:,:,:,2] = ui16_PMAP_cat3_xy
#%% write files out as multipage tiff
for channel in range(0,3):
    image to save = (data to save[:,:,:,channel]).astype(dtype=np.uint16)
    rearranged image to save = np.transpose(image to save, axes=[2,0,1])
imageio.mimwrite('Ch '+str(channel+3)+' lm 001'+'.tiff',rearranged image to sav
e)
```

Extracting single-cell features using label-free cell segmentation and CellProfiler 4

- Download and install CellProfiler4 from the CellProfiler website. https://cellprofiler.org



1. 2-D CellProfiler Pipeline

-A screencast video is included with the BioStudies archive.

- Folder 5 "5_CellProfiler_cell_measurements" in the 2-D BioStudies project archive contains example data, a CellProfiler pipeline and the subsequent CellProfiler single-cell outputs:





- The image-data (inside 1_Structured_test_data) contains four immunofluorescence channels followed by the probability maps obtained for the label-free nuclei, actin and background/other classes from either the Python or MATLAB deep learning scripts (channels 5, 6 and 7, respectively). - Open CellProfiler 4 and load the image analysis pipeline from the "2_CellProfiler_pipeline" folder.

- Drag-and-drop the "1_structured_test_data" folder into the CellProfiler image-window to load the 2-D image-data:

Image: Signal Control Signal	ments		- 0	Cellerford File Edit 1 € € € € € € € € € € € € € € € € € € €	er 4.1.5: CM415_PeyensPatch lestData lest Windows Help \$ Images \$ Metadata \$ NamesAndTypes \$ Groups \$ Threshold \$ Threshold \$ Threshold \$ MaskImage	Celevanes open (Dr. Vannou C. 1944), an international Markan, an Unit 15, Celevanes, org	
Image: Second secon	Move Copy to* Copy Organise NatLAB_2D_UNET > 5_CellProfiler_co	New item * Easy access * Properties Call ew Open ell_measurements v C	n • ESelect all Select none ory Invert selection Select Select	0 G 0 G 0 G 0 G 0 G 0 G 0 G 0 G 0 G 0 G	5 Maskimage 5 RescaleIntensity 6 RescaleIntensity 6 GrayToColor 5 GrayToColor 5 SaveImages 5 SaveImages	Drop files and folders	hore
This PC 30 Objects Desktop Bocuments Documents	Name 1,Structured,test_data 2,CellProfiler_pipeline 3,Outputs	Date modified 30/11/2021 14:08 30/11/2021 14:08 30/11/2021 14:09	Type Sid File folder File folder File folder	(ze) Φ G G G Φ G G G Φ G G G Φ G G G Φ G G Φ G G G Φ G G G Φ G G G Φ G G G Φ G G G Φ G G G Φ G G G G Φ G	5 Savemages 5 Threshold 5 Maskimage 6 ImageMath 5 GaussianFilter 5 Threshold 5 ImageMath 6 Savemathelat	Drop mes and loiders	liere
Music Videos Videos Videos Videos Videos Videoss Video	• <			0000 0000 0000 0000 0000 0000 0000 0000 0000	Maskimage (Maskimage) (Maskimage) Sovermages Sovermages OverlayOutlines Savelmages (MeasureObjectSizeShape MeasureObjectIntensity	Filter imagen? [No filtering v	?

- Choose where to save the outputted cell features by clicking on the "Output Settings" tab at the bottom-left of the CellProfiler dialogue.



- To run the CellProfiler pipeline and save the cell features and other pipeline outputs at the specified location, click "Analyse Images" at the bottom-left of the CellProfiler screen.

CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellC	Dutlines.cpproj (D:\John\2021\MATLAB_biostudies\MATLAB_2D_UNET\5_CellProfiler_cell —	×
CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellC File Edit Test Windows Help C Images C Metadata C NamesAndTypes C Groups C Threshold C Threshold C MaskImage C RescaleIntensity C RescaleIntensity C GrayToColor C Gray	Dutlines.cpproj (D:Vohn\2021\MATLAB_biostudies\MATLAB_2D_UNET\5_CellProfiler_cell	× it_data\/ it_data\/ it_data\/ it_data\/ it_data\/ ? ?
Output Settings View Workspace		
? Adjust modules: + - ^ V I Start Test Mode Analyze Images	Found 7 rows	

- Previously-saved CellProfiler outputs are also available inside the "3_Outputs" folder:

MATLAB_biostudie	s > MATLAB_2D_UNET > 5_CellProfiler_ce	ell_measurements v	🖸 🔎 Se
^	Name	Date modified	Туре
	1_Structured_test_data	30/11/2021 14:08	File folder
	2_CellProfiler_pipeline	30/11/2021 14:08	File folder
	3_Outputs	30/11/2021 14:09	File folder

2. 3-D CellProfiler Pipeline

-A screencast video is included with the BioStudies archive.

- Folder 5 "5_CellProfiler_cell_measurements" in the 3-D BioStudies project archive contains example data, a 3-D CellProfiler pipeline and the subsequent CellProfiler single-cell outputs:

□ < DATA (D:) > John > 2021	> MATLAB_biostudies > MATLAB_3D_UNET	5_CellProfiler_cell_measu	rements ~	ē
^	Name	Date modified	Туре	Size
ts		30/11/2021 14:10	File folder	
	2_CellProfiler_pipeline	30/11/2021 14:10	File folder	
nts	3_Outputs	30/11/2021 14:10	File folder	



- The image-data (inside 1_Structured_test_data) contains two immunofluorescence channels followed by the 3-D probability maps obtained for the label-free nuclei, actin and background/other classes from either the Python or MATLAB deep learning scripts (channels 3, 4 and 5, respectively).

- Open CellProfiler 4 and load the image analysis pipeline from the "2_CellProfiler_pipeline" folder.

- Drag-and-drop the "1_structured_test_data" folder into the CellProfiler image-window to load the 2-D image-data:



- Choose where to save the outputted cell features by clicking on the "Output Settings" tab at the bottom-left of the CellProfiler dialogue.

Ce	ellProfile	r 4.1.3: CP413	3_3D_MLN_pipeline.cpproj (E	:\John\2021\MATLAB_biostudies\MATLAB_3D_UNET\5_CellProfiler_cell_measurements\2_CellPro	_		×
File	Edit Te	st Window	vs Help				
	C	Images		2 Default land Falder cillilluscretillelaire			
	S	Metadata					
	c	NamesAnd	Types	? Default Output Folder: D:\\CELLPROFILER_OUT		\sim	
	S	Groups					
	<⊅ ₪	Threshold					
	<⊅ ₪	Maskimage	2				
	<⊅ ₪	Savelmage	s				
	<⊅ ₪	Savelmage	s				
	<₽ €	Watershed					
	<₽ €	ConvertOb	jectsTolmage				
	<₽ €	GaussianFil	ter				
	Image: A state of the state	Threshold					
	Image: A start and a start	ImageMath	1				
	Image: Second	RemoveHo	les				
	(2)	Watershed					
		ConvertOb	Jects lolmage				
		Savelmage	S				
		MassureOb	ojectintensity				
		EvportToSp	ojectoizeonape				
	\$P 🕒	Exportiosp	readsheet				
	Outp	ut Settings	View Workspace				
	outh	- secongs	Ten ronspore				
?	Adj	ust modules	+ - ^ v				
Ð	Start Te	st Mode	Analyze Images				

- To run the CellProfiler pipeline and save the cell features and other pipeline outputs at the specified location, click "Analyse Images" at the bottom-left of the CellProfiler screen.

CellProfiler 4.1.3: CP413_	3D_MLN_pipeline.cpproj ([D:\John\2021\MATLAB_biostudies\MATLAB_3D_UNET\5_CellProfiler_cell_measurements\2_CellPro — 🛛 💫	<
File Edit Test Windows	Help		
Images File Edit Test Windows Images Images Images Metadata Images Images Images Images <tdimages< td=""> Images</tdimages<>	3D_MLN_pipeline.cpproj (I Help /pes ctsToImage sr ctsToImage ectIntensity ectSizeShape addsheet	D:\John\2021\MATLAB_biostudies\MATLAB_3D_UNET\5_CellProfiler_cell_measurements\2_CellPro — _ >	
≪ 🕼 ExportToSpre	eadsheet	 ✓ Show files excluded by filters 	>?
		Filter images? Images only ? Apply filters to the file list ?	
Output Settings	View Workspace		
2 Adiational di	nan nonspace		
Adjust modules: Start Tart Made	+ · · · V]	
Start Test Mode	Analyze Images	Found 5 rows	

- Previously-saved CellProfiler outputs are also available inside the "3_Outputs" folder:

A w DATA (D:) > Joł	hn → 2021 :	> MATLAB_biostudies > MATLAB_3D_UNET >	5_CellProfiler_cell_measure	ments v	5
	^	Name	Date modified	Туре	Size
ts			30/11/2021 14:10	File folder	
		2_CellProfiler_pipeline	30/11/2021 14:10	File folder	
nts		3_Outputs	30/11/2021 14:10	File folder	8

2-D Cell Profiler image analysis pipelines. This section presents screenshots of a CellProfiler image analysis pipeline used to achieve label-free cell segmentation in 2-D from the Unet network outputs, and to measure the intensity and size/shape features of identified cell-objects. To use the image analysis pipeline with new image data, the 'IdentifyPrimaryObjects' module simply needs adjusting so that the 'typical diameter of objects' size-range matches the pixel scaling of the new images. For newcomers to CellProfiler, we recommend downloading the image-data and pipeline from BioStudies database https://www.ebi.ac.uk/biostudies/ under accession number S-BSST742. This enables the pipeline to be run with the data described in the manuscript and allows the user to see how each module works.

Image: Bit The Workers Hege Image: Bit The Workers Hege <th>CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell</th> <th>Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FRE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch</th> <th>. 🗆 X</th>	CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FRE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch	. 🗆 X
Channel - CD12 Chane	File Edit Test Windows Help		
Marinda Montal (1 - CD1) Convert 3: 9 Wind Markel Convert	⊠ Images		
Vancakaflyse George Googe Goo	Metadata	Channel_1 = CD11c	^
Concept Section C	NamesAndTypes	Channel_2 = CD3	
	Groups	Channel 3 = UP	
	() C Threshold	Channel 5 = label-free nuclei probability man from Unet	
Constraining	() C Threshold	Channel 6 = label-free actin probability map from Unet	
Sustainage Sustaina	() C Maskimage		
 C. L. I., MO.1.8 C. Saccialization M. C. L. I., MO.1.8 C. L. I., MO.1.8 C. L. I., MO.1.8 C. L. I., MO.1.8 C. Saccialization M. C. L. I., MO.1.8 C. L. I., MO.1.8 C. Saccialization M. C. M. MO.1.20, LABL, FREL SCIANTIATION, PROJECTS 2021032, 2D, Sactude A.J., Fail J. Depar. Patch assign MATLAB.2D, PyrenPatch S, Cell Profile cell measurements 1, S. C. M. MO.1.8 C. Saccialization M. M. M. MO.1.20, LABL, FREL SCIANTIATION, PROJECTS 2021032, 2D, Sactude A.J., Fail J. Depar. Patch assign MATLAB.2D, PyrenPatch S, Cell Profile cell measurements 1, S. C. M. MO.1.80, D. PyrenPatch S, Cell Profile cell measurements 1, S. C. M. MO.1.80, D. PyrenPatch S, Cell Profile cell measurements 1, S. C. M. MO.1.80, D. PyrenPatch S, Cell Profile cell measurements 1, S. C. Saccialization M. M. MO.2000, MO.1.20, LABL, FREL SCIANTIATION, PROJECTS 2021032, 2D, Sactude A.J., Fail J. Depar. Patch assign MATLAB.2D, PyrenPatch S, Cell Profile cell measurements 1, S. C. Saccialization M. M. MO.2000, MO.1.20, LABL, FREL SCIANTIATION, PROJECTS 2021032, 2D, Sactude A.J., Fail J. Depar. Patch assign MATLAB.2D, PyrenPatch S, Cell Profile cell measurements 1, S. C. Saccialization M. C. Saccialization M. M. MO.2000, MO.1.20, LABL, FREL SCIANTIATION, PROJECTS 2021032, 2D, Sactude A.J., Fail J. Depar. Patch assign MATLAB.2D, PyrenPatch S, Cell Profile cell measurements 1, S. C. Saccialization M. C. M. MO. MO. MO. MO.2000, MO.1.20, LABL, FREL SCIANTIATION, PROJECTS 2021032, 2D, Sactude M. T. M. M. MATLAB.2D, PyrenPatch S, Cell Profile cell measurements 1, S. Saccialization M. C. M. M.	Maskimage	D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch\5_CellProfiler_cell	_measurements\1_St
		Ch_1_lm_001.tif	
 Constructions Construc		D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch\5_CellProfiler_cell	_measurements\1_St
 Operation of the comparison of the	() C GravInColor	Ch_2_Im_U01.tdt D_01.bbs/2020/2020/417_2D_1_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_MATLAD\2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APEL_EPEE_SEGMENTATION_DROLECTS\20210522_2D_biostudies\2_Eixel_2D_Dware_Databased_APE	massuraments\1 St
 S. Starmage S. Starm	() C GravToColor	Ch 3 im 00.titi	_measurements(1_5
 Church Cliff Churc	() Savelmager	D:/John/2020/20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS/20210522_2D_biostudies/2_Final 2D Peyers Patch example/MATLAB/2D_PeyersPatch/5_CellProfiler_cell	_measurements\1_St
 C Tueshal C Tueshal C Mukanoge Mukanoge <li< td=""><td>Savelmages</td><td>Ch_4_Im_001.tif</td><td></td></li<>	Savelmages	Ch_4_Im_001.tif	
 Control Mathemate Control Mathemate<	C Devenhoges C Threshold	D:\John\2020/20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch\5_CellProfiler_cell	_measurements\1_St
 C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. S. Jim, 2014/fi C. Jim, 2014/fi S. SaveInage: 	🔿 🗹 Maskimage	Cn_2_IIII_WILLIIII EI IN/JANN/2020/20200417 2D LABEL FREE SEGMENTATION DROJECTS/20210522 2D biostudies/2 Final 2D Devers Datch example/MATLAR/2D DeversDatch/S CellDrofiler cell	measurements\1 St
		Ch 6 Im 001.iff	_measurements/1_3
 Churchold Chur		D: John/2020/2020/417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch\5_CellProfiler_cell	_measurements\1_St
© E inspektati, © MasuroDigit: © Adjust module: • • • • • • • • • • • • • • • • • • •	♂ S Threshold	Ch_7_Im_001.tiff	
Oudput Setting: Vew Workspace 2 Adjust module: Post Trat Mode Found 1 image sets	♂ S ImageMath		,
Image: Stating: Sta		•	
IdentifyFirmsyObjects IdentifyFirmsyObjec	Maskimage		3
Interinger, Konnethig Interinger, Konnethig Sectors and Secto		Eilter imager? No fileging	2
© E Savetmags © Sovetmags © Sovetmags © MesureObjetciteShape © EbportToSpredsheet Output Settings Yee Workspace 2 Adjuar modules: + + Savet Test Mode Found 1 image sets		niteringes: No filtering V	f
© E Overlag Outlines © E MessureObjectIntensity © E ExportToSpreadsheet			
 ♥ G Savelinger ♥ G MesureObjectSizeShape ♥ G EponToSpreadsheet 			
 [®] G MasureObjectSizeShape [®] G ExportToSpreadtheet [®] G ExportToSpreadtheet [®] G ExportToSpreadtheet [®] G Adjust modules [®] G G G G G G G G G G G G G G G G G G G			
G KessureObjectintensity G ExportToSpreadsheet Uutput Settings View Workspace Adjust modules: + - ^ v Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules • Start Test Mode • Analyze Images			
Output Settings View Workspace 2 Adjust modules + - + - + Start Test Mode + Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - * Start Test Mode * Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - +> Start Test Mode -> +> Start Test Mode >> +> Start Test Mode >>			
Output Settings View Workspace ? Adjust modules: + - > Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - I b Start Test Mode I mage sets			
Output Settings View Workspace 2 Adjust modules: + - I > Start Test Mode > Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - > Start Test Mode Found 1 image sets	1		
Output Settings View Workspace ? Adjust modules: + - > Start Test Mode Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ v > Start Test Mode > Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ * * * * * * Start Test Mode * Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - > Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ v > Start Test Mode > Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ v Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ > Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - > Start Test Mode > Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ > Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ v I Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ V Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - > Start Test Mode Analyze Images Found 1 image sets			
Output Settings View Workspace ? Adjust modules: + - ^ v Start Test Mode Analyze Images Found 1 image sets			
? Adjust modules: + - ^ V > Start Test Mode > Analyze Images Found 1 image sets	Output Sattings View Washerson		
Start Test Mode Analyze Images Found 1 image sets	? Adjust modules: + - ^ v		
Found 1 image sets	Ib Stat Task Made		
	Analyze Images	Found 1 image sets	



CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellC	Outlines.cpproj* (D:	\John\2020\202	00417_2D_LABEL	FREE_SEGMENTA	TION_PROJECTS	\20210522_2D_bio	studies\2_Final 2D	Peyers Patch exar	mple\MATLAB\2D_PeyersPatch.	. –		×
File Edit Test Windows Help												
🕑 Images												_
🖾 Metadata	Loads images a	and associates t	hem together acc	cording the specif	ied rule criteria							\sim
☑ NamesAndTypes												
🖸 Groups												
International 🗇 🗹 Threshold												
< 🗹 Threshold												\sim
< 🗹 Maskimage												
🗇 ビ Maskimage	Assic	n a name to In	nages matching r	ules 🗸							?	^
🗇 🗹 RescaleIntensity	-											
I RescaleIntensity	Pr	ocess as 3D?									?	
I GrayToColor												
I GrayToColor		Ma	atch All 🗸 of	the following rule	5						?	
I Savelmages	Select the	e rule criteria —	Disastana		. A . L .	Chan	nel 1				i	
I Savelmages			Directory ~	Does V Col	ntain	~ Chan	nei_i			- +		
🗇 🗹 Threshold	Name to assign t	hese images c	1								?	
🔿 🗹 Maskimage												
🔿 🗹 ImageMath	Select the	e image type Gr	rayscale image	\sim							?	
🔿 🖸 GaussianFilter	Set intensity	range from Im	nage bit-depth	~							?	
G Inreshold												
C Imagewath		L	Juplicate this ima	ige								
Kentovenores												
IdentifyPrimaryObjects		Ma	atch All 🗸 of	the following rule	s						?	
ConvertObjectsTolmage	Select the	e rule criteria —	D : 1			C					i	
			Directory ~	Does V Coi	ntain	~ Chan	nei_2			- +		
I CverlayOutlines	Name to assign t	hese images 🥻	2								?	
🗇 🗹 Savelmages												
I MeasureObjectSizeShape	Select the	e image type Gi	rayscale image	~							3	
I MeasureObjectIntensity	Set intensity	range from Im	nage bit-depth 🕚	~							?	
I ExportToSpreadsheet			Qualicata this ima								2	
			Supreate this inte	ige							<u> </u>	
		F	Remove this imag	je							?	
		Ma	atch All 🗸 of	the following rule	<.						2	
	Select the	e rule criteria				Char					i	
		_	Directory ~	Does V	ntain	~ Chan	nei_5			- +		
	Name to assign t	hese images c	3								?	
								-			-	
	Update	cl	C2	C3	c4	co	CD	c/				
	1 (Ch_1_lm_001.tif	Ch_2_Im_001.tif	Ch_3_Im_001.tif	Ch_4_Im_001.tif	Ch_5_Im_001.tiff	Ch_6_Im_001.tiff	Ch_7_Im_001.tiff				
Output Settings View Workspace												
? Adjust modules: + - ^ v												
Start Test Mode Analyze Images	Found 7 rows											
	.,											_



CellProfile	er 4.1.3: CP413	PeyersPatchTestData_CellC	Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch	×
File Edit T	est Window	rs Help		
C	Images			
C	Metadata		I his module thresholds the ci i mages (CU i i c channe) The output is a binary image for civels above the threshold	
C	NamesAnd	Types		
C	Groups			
۵ 🖾	Threshold			
Image: Contract of the second seco	Threshold			× .
	Maskimage	1		
	Recoleinte	e situ	Select the input image c1 v (from NamesAndTypes)	?
0 0 0	RescaleInte	nsity	Name the systematic typesh CD11c binary	2
() () () ()	GravToCol	or		•
ا ا	GrayToCol	or	Threshold strategy Global 🗸	?
ي 🖒	Savelmage	5	Thresholding method Manual	2
ا ال	Savelmage	5		•
ا 🖾	Threshold		Manual threshold 0.01	?
<	Maskimage	:	Threshold smoothing scale 0	?
S	ImageMath	1		•
<!<!<!<!<!<!<!<!<!<!<!<!<!<!<!<</td <td>GaussianFil</td> <td>ter</td> <td></td> <td></td>	GaussianFil	ter		
	Ihreshold			
	Pamageiviatr	l los		
	Mackimage	ies .		
	ldentifvPrir	narvObiects		
() C	ConvertOb	iectsTolmage		
() C	Savelmage	5		
ا 🖉	0verlayOut	lines		
ا 🖉	Savelmage	5		
ى 🖒	MeasureOb	jectSizeShape		
ا 🖉	MeasureOb	jectIntensity		
< ♥ ☑	ExportToSp	readsheet		
0	aut Cattings	View Workspace		
2	liust modules	view workspace		
r Ad	just modules	+ - ~ V		
Start T	est Mode	Analyze Images	Found 7 rows	

Comparison The module thresholds the clamage (CD3 channe) Comparison The output is a binary image for pilots above the threshold Comparison Statustication Statustication Statustic	it Test Windows Help	
Output Interaction	C Images	This module thresholds the c2imaoes (CD3 channel)
Company product The endpoint is a binary image for picks above the threshold Company product The endpoint is a binary image for picks above the threshold Company product Security Security Security	Nietadata	
Comparison Second Second Second <	Groups	The output is a binary image for pixels above the threshold
Computational Status Image Status Computational Status Status Status		
Comput Stating View Workspace Appler Stating View Workspace Appler Stating View Workspace	C Threshold	
Composition Select the input image Image: Imag	⊕ 🖸 Maskimage	
Control Status Contr	🗇 🖸 Maskimage	Select the input impage 2 (from Namer And Turger)
Corpor Setting: Yeer Workspace Apport Setting: Yeer Workspace	🗇 🗹 RescaleIntensity	
Corput Stating: Yee Workspace: Apjust mediate: Image:	🗇 ビ RescaleIntensity	Name the output image thresh_CD3_binary
Corput Setting: Verw Workspace Agust module: • • • • • • • • • • • • • • • • • • •	🗇 🗹 GrayToColor	
Copper Setting: Verw Workspret Apjurt module: •	I GrayToColor 🖸 🗇	Inreshold strategy Global V
Corpert Setting: VerwWorkspace: Adjust models: • • • • • • • • • • • • • • • • • • •	I Savelmages	Thresholding method Manual 🗸
Output Stating: VerwWorkspace: Adjust mednes: • • • • • • • • • • • • • • • • • • •	🗇 🗹 Savelmages	
Output Stating: VerwWorkspace Agust models: • • • • • • • • • • • • • • • • • • •	🗇 🗹 Threshold	Manual threshold 0012
Output Stitling: Verw Vorkpace Adjut modules: <!--</td--><td>🗇 🗹 Maskimage</td><td>Threshold smoothing scale</td>	🗇 🗹 Maskimage	Threshold smoothing scale
Output Stetling: View Workspace: Adjut module: Image Nath	🗇 🗹 ImageMath	
Output Setting: View Workspace Adjust module: Notes	🗇 🕒 GaussianFilter	
Output Settings View Workspace Adjut modules • • • • • • • • • • • • • • • • • • •	Inresnoid	
Output Setting: View Workspace Adjust modules: • Adjust modules: • Adjust modules: • Versupport • Versupport • Versupport • Versupport • •<	🗇 🖸 imageiviath	
Output Settings View Workspace Adjust modules: • • • • • • • • • • • • • • • • • • •	🖤 🖸 Kemoverioles	
Output Setting: View Workspace Adjut module: • Adjut module: • Very Workspace • • •	🗇 🖸 Maskimage	
Output Settings View Workspace Adjust modules:	GenuryPrimaryObjects ConvertObjectsTolmage	
Output Setting: View Workspace Adjust module: •	Convertobjects ronnage Savelmager	
Output Settings View Workspace Adjust modules: • • •	OverlayOutlines	
 Vew Workspace Adjust modules A v 	C Savelmages	
© E kportToSpreadsheet Output Settings View Workspace Adjust modules + + ∧ ∨	MeasureObjectSizeShape	
© EpontToSpreadsheet		
Output Settings View Workspace Adjust modules: + A V	🗇 ビ ExportToSpreadsheet	
Output Settings View Workspace Adjust modules: + - ^ V		
Output Settings View Workspace Adjust modules: + + . N V		
Output Settings View Workspace Adjust modules:		
Output Settings View Workspace Adjust modules: + • v		
Output Settings View Workspace Adjust modules: <lu> </lu> <lu> </lu> 		
Output Settings View Workspace Adjust modules: •		
Output Settings View Workspace Adjust modules: +		
Output Settings View Workspace Adjust modules: +		
Output Settings View Workspace Adjust modules: • • • • •		
Output Settings View Workspace Adjust modules: + - ^ v		
Output Settings View Workspace Adjust modules: + - ^ ^ v		
Output Settings View Workspace Adjust modules: + - ^ ^ v		
Output Settings View Workspace Adjust modules: + - ^ v		
Output Settings View Workspace Adjust modules: + - ^ V		
Output Settings View Workspace Adjust modules: + - ^		
Output Settings View Workspace Adjust modules: + - ^ / v		
Output Settings View Workspace Adjust modules: + - ^ ^ v		
Output Settings View Workspace Adjust modules: + - ^		
Output Settings View Workspace Adjust modules: + - ^ v		
Adjust modules: + - ^ v	Output Settings View Works	ace
Adjust modules: + - ^ v		
	Adjust modules: + -	

CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellC	Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FRE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch – 📃	×
File Edit Test Windows Help		
Mages		
Metadata	This module creates a greyscale version of the c1 (CD11c) data - where the threshold calculated above is applied.	^
NamesAndTypes		
Groups	Ine outputted image from this module can be measured in each cell-object to calculate per-cell intensity values AF LEX the background has been thresholded out. This is particulary imported here as the imported in the cancel and the set of th	
	important nere, as integration of this image in centrologicus anows for background conection according to the dissue-matched secondary-only and sotype controls.	
C Threshold		~
↓ S Maskimage		2
	Select the input image c1 (from NamesAndTypes)	- ?
	Next the set of these h CD11c groups all	2
CravEcolor		
() C GrayToColor	Use objects or an image as a mask? Image V	?
() C Glayrocolor		
() C Savelmages	Select image for mask thresh_CD11c_binary v (from Threshold #05)	?
G Savennages		2
A C Maskimage	Invert the mask? O Yes No	- f
G ImageMath		
() C GauccianFilter		
() C ImageMath		
C RemoveHoles		
Kentovenoies		
IdentifyPriman(Objects		
ConvertObjects		
Savelmager		
() Saveninges		
Swelmager		
MeasureObjectSizeShape		
MeasureObjectIntensity		
W C Export rospreadsneet		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode Analyze Images	Found 7 rows	

😨 CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellOutlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch – 🛛 🗙					(
File Edit Test Windows Help					
	🕑 Images	This module creater a grouped everyon of the c2 (CD2) data , where the threshold calculated shows is applied			1
	🖸 Metadata	This module creates a greyscale version of the C2 (CD3) data - where the threshold calculated above is applied.			
	☑ NamesAndTypes	The outputted image from this module can be measured in each cell-object to calculate per-cell intensity values AFT	TER the background has been thresholded out. This is particulary		
	Groups	important here, as integration of this image in cell objects allows for background correction according to the tissue-	matched secondary-only and isotype controls.		
Ì	Threshold				
$\langle \rangle$	C Threshold			~	1
()	G Maskimage				
(b)	MaskImage	Select the input image c2 v (from NamesAndTypes)		?	
()	C RescaleIntensity				
(Þ)	KescaleIntensity	Name the output image thresh_CD3_greyscale		?	
(P)		Use objects or an image as a mask? Image		2	
<i>(</i>)	GrayloColor	and a second s			
(P)	Saveimages	Select image for mask thresh_CD3_binary v (from Threshold #06)		?	
	Saveimages			2	
	Mackimage	Invert the mask? O Yes No		- f	
	Maskinage	0.13 0.10			
()	GaussianFilter				
đ	Threshold				
đ	MageMath				
()	C RemoveHoles				
1	Maskimage				
Þ	☑ IdentifyPrimaryObjects				
Þ	ConvertObjectsTolmage				
(٢)	Savelmages				
(٢)	OverlayOutlines				
(٩)	Savelmages				
Þ	MeasureObjectSizeShape				
Þ	MeasureObjectIntensity				
Þ	ExportToSpreadsheet				
0	utput Settings View Workspace				
	A Part of the second seco				
?	Adjust modules: + - ^ v				
No. 10 Start	t Test Mode 🕨 Analyze Images	Found 7 rows			

CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellOutlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch				
File Edit T	Test Windows Help			
C C	3 Images 3 Metadata	This module rescales the 12-bit thresholded CD11c data for subsequent visulisation and saving.	1	
C	VamesAndTypes Groups	Saturated pixels (i.e., 4095) in the source 12-bit image are rescaled to the maximum		
() () ()	Threshold			
<u>ن</u>	Maskimage			-
ا	Maskimage	Select the input impage thread (D11- encoded and (from Mackimans #07)	2	
Image: Contract of the second seco	RescaleIntensity	Select the hippic image (thesh_cDTrc_greyscale > (from Maskinage + 07)		
\$\Phi\$	RescaleIntensity	Name the output image thresh_CD11c_greyscale_Rescaled	?	
ا 🗇	GrayToColor		2	
Image: Contract of the second seco	GrayToColor	Choose specific values to be reset to a custom range	f	
() (i	Savelmages	Method to calculate the minimum intensity Custom V	?	
00	Saveimages	Method to calculate the maximum intensity Custom	?	
00	Maskimage			
چ <u>د</u>	2 ImageMath	intensity range for the input image 0.0 0.0025	ſ	
\$ C	GaussianFilter	Intensity range for the output image 0.0 1.0	?	
ی ک	2 Threshold			
ی ک	3 ImageMath			
\$\phi\$	RemoveHoles			
(1)	Maskimage			
	GenuaryObjects			
0 C	Savelmages			
() C	VerlavOutlines			
ا 🗇	Savelmages			
\$\Phi\$	MeasureObjectSizeShape			
1	MeasureObjectIntensity			
ا	ExportToSpreadsheet			
Out	tput Settings View Workspace			
? Ac	djust modules: + - ^ v			
🕩 Start T	Fest Mode Analyze Images	Found 7 rows		-

CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	Outlines.cpproj* (D:\John\2020\20200417.2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522.2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —	
File Edit Test Windows Help		
C Images		
G Metadata	This module rescales the 12-bit thresholded CD3 data for subsequent visulisation and saving.	~
NamesAndTypes	Structed pixels (i.e. 4005) in the course 12-bit image are recorded to the maximum	
🖾 Groups	Saurated pixels (i.e., 405) in the source re-bit image are rescaled to the maximum	
Interstold		
Interstold		\sim
🗇 🗹 Maskimage		
🗇 🗹 Maskimage	Select the input image thresh_CD3_greyscale <pre>v (from MaskImage #08)</pre>	?
RescaleIntensity		
KescaleIntensity	Name the output image Thresh_CU3_greyscale_Kescaled	ſ
CrayToColor	Rescaling method Choose specific values to be reset to a custom range	?
	Mathe d to select the minimum intensity for the selection of the selection	2
I Savelmages		
🗇 🗹 Threshold	Method to calculate the maximum intensity Custom	?
🗇 🖸 Maskimage	Intensity range for the input image 0.0 0.0625	?
ImageMath		
🗇 🗹 GaussianFilter	intensity range for the output image 0.0 1.0	f
🔿 🕑 Threshold		
G Remarkalas		
Maskimage		
IdentifyPrimaryObjects		
I ConvertObjectsTolmage		
🗇 🗹 Savelmages		
International States (1997) 🖾 🗇		
I Savelmages		
MeasureObjectintensity EvenetTeSpeedsheet		
W C Export rospreadsneet		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode Analyze Images	Found 7 rows	

File Edit Tes	st Windows Help		
। । । । । ।	Images Metadata NamesAndTypes Groups	This module makes a colour image of the thresholded CD11c image data	<
@ U	I hreshold		
	Threshold		~
\$ 5	Maskimage		_
	Maskimage	Select a color scheme Composite \checkmark ?	
0 C	RescaleIntensity	Name the output image View thresh CD11c rescaled ?	
	GravTeColor		- 1
	GravToColor	Rescale intensity Over One	
0	Savelmages	include intensity O yes @ No	
\$ C	Savelmages	made name threeh CD11s exercise Recorded or (from PerceleIntentity #00)	
1	Threshold	image name tutest_CDTTC_greystate_rescaled v (non-rescalentensisy +09)	- 1
S	Maskimage	Color	
Image: A marked black of the second secon	ImageMath	w : u 10	- I
🗇 🕑	GaussianFilter	weight 10	
I	Threshold	Add another channel Add another channel ?	
₫ 🗇	ImageMath		-
S	RemoveHoles		
< ♥ ♥	Maskimage		
Image: Second	IdentifyPrimaryObjects		
	ConvertObjects I olmage		
0 C	Saveimages		
	Savelmages		
0	MeasureObjectSizeShane		
00	MeasureObjectIntensity		
0 5	ExportToSpreadsheet		
Outpu	ut Settings View Workspace		
? Adju	ust modules: + - ^ v		
Start Tes	st Mode 🕨 Analyze Images	Found 7 rows	

CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellOutlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_final 2D Peyers Patch example\MATLAB\2D_PeyersPatch... -
CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	IIOutlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch	– 🗆 ×
File Edit Test Windows Help		
び Images び Metadata び NamesAndTypes	This module makes a colour image of the thresholded CD3 image data	^
Groups		<u>_</u>
 	Select a color scheme Composite V	?
 	Name the output image View_thresh_CD3_rescaled Rescale intensity O Yes No	?
 	Image name thresh_CD3_greyscale_Rescaled v (from RescaleIntensity #10)	?
🗇 🗹 Maskimage	Color	?
	Weight 1.0	?
 	Add another channel Add another channel	?
 ⑦ E RemoveHoles ⑦ E Maskimage ⑦ E IdentifyPrimaryObjects ⑦ E ConvertObjectsTolmage ⑦ E SaveImages 		
Export I obpreadsheet Export I o		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode Analyze Images	Found 7 rows	



CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellOutlines.cpproj* (D:\John\2020/20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch... -



🔽 CellProfiler 4.1.3: CP413_PeyersPa	tchTestData_CellOutline	es.cpproj* (D:\John\2020\20200417_	2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —		×
File Edit Test Windows Help					
☑ Images ☑ Metadata ☑ NamesAndTypes ☑ Groups ④ ☑ ⑦ ☑ ⑦ ☑ ⑦ ☑ ⑦ ☑ ☑ Threshold	Cr	reates a binary mask of the image-r	egion containing lymphoid tissue		<
🚸 🗹 Maskimage					
🗇 🗹 Maskimage		Select the input image	c4 v (from NamesAndTypes)	1	?
🗇 🗹 RescaleIntensity					
KescaleIntensity KescaleIntensity		Name the output image	Iymphoid_tissue_KOI_mask		<i>(</i>
GrayToColor		Threshold strategy	Global 🗸	1	?
I Savelmages		Thresholding method	Oteu		2
I Savelmages		in caloung neurod			•
🗇 🗹 Threshold	Two	o-class or three-class thresholding?	Two classes 🗸	1	?
I Maskimage		Threshold smoothing scale	0	1	?
🗇 🕑 ImageMath		Theorem and a second second second	10		2
		Infeshold correction factor			:
	Low	ver and upper bounds on threshold	0 1.0	1	?
🗇 🕑 RemoveHoles					?
I Maskimage	L	.og transform before thresholding?	○ Yes ◉ No		
🚸 🗹 IdentifyPrimaryObject	ts				
ConvertObjects Folma	age				
OverlayOutlines					
🚸 🗹 MeasureObjectSizeSh	ape				
I MeasureObjectIntens	ity				
I ExportToSpreadsheet					
Output Settings View Wo	rkspace				
? Adjust modules: +	- ^ v				
Start Test Mode	nalyze Images Four	ind 7 rows			

File Edit Test Windows Help	
Images Applies the lymphoid ROI mask to the LF-actin probabil NamesAndTypes Groups Threshold Fraction probability	ility map used to enable cell-segmentation
🗇 🕑 Threshold	
🗇 🗹 Maskimage	
	(from NamesAndTypes) ?
	?
🗇 🗹 GrayToColor	
	<i>₹</i>
	Il mask v (from Threshold #15) ?
	?
Maskimage	
() Gaussianhilter	
() C Inteshold	
Maskimane	
IdentifyPrimaryObjects	
ConvertObjectsTolmage	
🛷 🗹 MeasureObjectIntensity	
🗇 🕑 ExportToSpreadsheet	
Output Settings View Workspace	
? Adjust modules: + - ^ v	
Start Test Mode Analyze Images Found 7 rows	

© Threshold © Threshold © Makalange © Makalange © Stackalantandy	び Metadata び NamesAndTypes び Groups	Inverts the LF-actin probability ma	o such that the 'cytoplasm' of each cell appears as segmentable foreground
Current Nationage Compositional State	Croups Croups Threshold	-	
Output Stating Operation wrett Operation wrett wrett Operation wre	⊕ G Threshold		
Operation Interdet	🗇 🖸 Maskimage		
Current Stream The state stream Current Stream Name the output single inverted_marked_d Current Stream Stream Current Stream Stream Current Stream Multiply the first image bill Current Stream Stream Stream Str	🗇 🖸 Maskimage	Operation	Invert V
Control Section	🗇 🗹 RescaleIntensity	operation	
Output Stating: Verw Workspace Output Stating: Verw Workspace	🗇 🗹 RescaleIntensity	Name the output image	inverted_masked_c6
Option Setting: Yee Workspeet Option Setting: Yee Workspeet	🗇 🗹 GrayToColor		
© Sevenages Setext the first image Makked_6 (from Maskimge #16) © Sevenages Multiply the first image Makked_6 (from Maskimge #16) © Maskimage Multiply the first image by 10 © Gustamilier Multiply the first image by 10 © Gustamilier Multiply the result by 10 © Gustamilier Multiply the result by 10 © Gustamilier Multiply the result by 10 © Sevenages Set values less than 0 equal to 07 © Sevenages Set values less than 0 equal to 17 © MessureObjectificEnhage Set values less than 0 equal to 17 © MessureObjectificEnhage No @ MessureObjectificEnhage No @ MessureObjectificEnhage No @ MessureObjectificEnhage No <tr< td=""><td>🗇 🗹 GrayToColor</td><td></td><td></td></tr<>	🗇 🗹 GrayToColor		
© Sevenages Multiply the first image by © Convertigies Multiply the result by © Convertigies Set values less than 0 equal to 0? © Soverlages Set values greater than 1 equal to 1? © MeasureDigetSintennagy Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages Set values greater than 1 equal to 1? © Soverlages No Set subjectimenagy Set values with 0? © Soverlages No Set subjectimenagy Set © No Ignore the image masks? Yes @ No	🗇 🗹 Savelmages	Select the first image	Masked_c6 v (from MaskImage #16)
Output Setting: View Workspace: Output Setting: View Workspace:	I Savelmages	Multiplushed Statistics and built	10
© MaskImage © Constrainting © Constrainting <td>Ihreshold</td> <td>Multiply the first image by</td> <td>10</td>	Ihreshold	Multiply the first image by	10
Cotput Setting Yeew Worksace Adjust modules +	🗇 🗹 Maskimage		
Output Setting: View Workspace: Adjust module: +	🗇 🗹 ImageMath	Deire the second discount is built	10
Output Settings View Workspace Output Settings View Workspace	🗇 ビ GaussianFilter	Raise the power of the result by	
Output Settings: View Workspace Adjust module: +	🗇 🕑 Threshold	Multiply the result by	1.0
Output Settings: View Workspace Adjust module: I	🗇 🗹 ImageMath		
Output Setting: View Workspace: Adjust module: +	🗇 🗹 RemoveHoles	Add to result	0.0
Output Setting: View Workspace Adjust modules: +	🗇 🗹 Maskimage		
Output Setting: View Workspace Adjust modules: +	GentifyPrimaryObjects	Set values less than 0 equal to 0?	● Yes ○ No
Output Setting: View Workspace Adjust modules: +	ConvertObjects Loimage		
Output Settings View Workspace Adjust modules: +	Saveimages	Catural and a sector three 1 and 12	
Output Settings View Workspace Adjust modules: +	C Swelmager	Set values greater than 1 equal to 1:	● Yes ○ No
Output Settings View Workspace Adjust modules: +	Saveimages MeasureObjectSizeShape		
Output Settings View Workspace Adjust modules: +		Replace invalid values with 0?	
Output Settings View Workspace Adjust modules: 	Kessureobjeenmensky ExportToSpreadsheet		
Output Settings View Workspace Adjust modules: +		Ignore the image masks?	
Output Settings View Workspace Adjust modules: + - ^ v		ignore the image masts.	V Yes (I) No
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: + - ^ v			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: + - ^ v			
Output Settings View Workspace Adjust modules: +			
Output Settings View Workspace Adjust modules: + - ^ v			
Output Settings View Workspace Adjust modules: + - ^			
Output Settings View Workspace Adjust modules: + - ^ v			
Output Settings View Workspace Adjust modules: + - ^			
Output Settings View Workspace Adjust modules: + - ^			
Output Settings View Workspace			
Adjust modules: + - ^ V v			
	Output Settings View Workspace	_	



File Edit Test Windows Help			
ⓒ Images ⓒ Metadata ⓒ NamesAndTypes ⓒ Groups	Thresholds the smoothed background probability map to yield a binary image		^
 ♥ E Threshold ♥ E Threshold 			~
 		Γ	
♦ ♥ RescaleIntensity	Select the input image Smoothed_background_pmap v (from	i GaussianFilter #18)	7
 	Name the output image binary_background		?
	Threshold strategy Global \sim		?
	Thresholding method Minimum Cross-Entropy 🗸		?
 ♦ Saveimages ♦ E Threshold 	Threshold smoothing scale 0.0		?
	Threshold correction factor 1.0		?
♦ C Imagenation ♦ C GaussianFilter	ower and upper bounds on threshold 0.0 1.0		?
			?
	Log transform before thresholding? O Yes No	L	-
IdentifyPrimaryObjects ConvertObjectsTolmage			
🚸 🖸 OverlayOutlines			
I Savelmages			
I MeasureObjectSizeShape			
(1) C Export i ospreadsheet			
Output Settings Way Walsones			
2 Adjust modules:			
ID Shed Test Made			
P Start Test Mode Analyze Images	ound 7 rows		

Ē

C mages			
☑ Metadata ☑ NamesAndTyp ☑ Groups	pes	Switches the foreground and back	ground i
🗇 🗹 Maskimage			
🗇 🖸 Maskimage		Operation	Invert 🗸
C RescaleIntensity RescaleIntensity	ty	Name the output image	inverted_binary_background
	y .	1 3	
I Savelmages		Select the first image	binary_background v (from Threshold #19)
Savelmages Savelmages		Multiply the first image by	1.0
		r	
I GaussianFilter		Raise the power of the result by	1.0
		Multiply the result by	1.0
TimageMath			00
		Add to result	
🚸 🗹 IdentifyPrimary	yObjects	Set values less than 0 equal to 0?	OV-CON-
	sTolmage		Tes O No
Savelmages Savelmages		Caturalization that 1 and to 12	
Savelmages	5	Set values greater than 1 equal to 1:	Yes O No
🗄 🗹 MeasureObject	tSizeShape	ſ	
🗇 🗹 MeasureObject	tIntensity	Replace invalid values with 0?	● Yes ○ No
I Export To Spread	dsheet		
		Ignore the image masks?	○ Yes ◉ No
Output Settings Vie	iew Workspace		
Output Settings Vie Adjust modules:	iew Workspace		

CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —	×
File Edit Test Windows Help		
🖸 Images		
🖸 Metadata	Removes small holes from the background other image	~
NamesAndTypes		
C Groups		
🔿 🖸 Threshold		~
🗇 ビ Maskimage		
🗇 🖸 Maskimage	Select the input impage invested binner bedramand and (from impageMath #20)	2
I RescaleIntensity	Select the input image invertee_binary_background (infinitiagewath #20)	
I RescaleIntensity	Name the output image deholed_inverted_binary_background	?
I GrayToColor		
I GrayToColor	Size of holes to fill 20.0	?
I Savelmages		
I Savelmages		
🗇 🗹 Threshold		
🗇 🗹 Maskimage		
💿 🗹 ImageMath		
🗇 🗹 GaussianFilter		
🔿 🖸 Threshold		
Imageiviatn G		
Kemoveholes		
IdentifyBriman/Objects		
ConvertObjects		
Savelmages		
OverlavOutlines		
Savelmages		
I MeasureObjectIntensity		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode		
Printiges	Found 7 rows	

CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellC	Outlines.cpproj* (D:\John\2020\202004	17_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —	X
File Edit Test Windows Help			
🕑 Images	[
🕑 Metadata	Subtracts the background from th	e LF probability map	^
NamesAndTypes			
🖾 Groups			
Inreshold			
Inreshold			\sim
🗇 🗹 Maskimage	-		
🗇 🗹 Maskimage	Select the input image	inverted masked c6 v (from ImageMath #17)	?
🗇 🗹 RescaleIntensity		interce_mosce_co	
I RescaleIntensity	Name the output image	background_masked_seeds	?
I GrayToColor	Use abiests as an image as a mark?		
	Use objects of an image as a mask:	image ~	f
	Select image for mask	deholed_inverted_binary_background v (from RemoveHoles #21)	?
🔿 🖸 Savelmages			
G Ihreshold	Invert the mask?		?
Maskimage Maskimage		U TES ©INO	
Gaussian Filter			
ImageMath			
C RemoveHoles			
🗇 🗹 Savelmages			
International State (1997) International State (
🗇 🗹 Savelmages			
I MeasureObjectSizeShape			
I MeasureObjectIntensity			
I ExportToSpreadsheet			
Output Settings View Workspace			
? Adjust modules: + - ^ v			
Start Test Mode Analyze Images	Found 7 rows		

CellProfiler 4.1.3: CP413_PeyersPatchTestData_CellC	Dutlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FRE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —	
File Edit Test Windows Help		
🖾 Images	I shall free call instance composition using the outputted probability many from the Unit model	
Metadata	Laber-nee cen instance segmentation using the outputted probability insps from the oner model	
☑ NamesAndTypes		
Groups		
< 🗹 Threshold		
💿 🗹 Threshold		×
🔿 🗹 Maskimage		
🔿 🖸 Maskimage	lise advanced settings?	?
Transity	O'Yes O'No	
The scale intensity		2
T Gray I o Color	Select the input image background_masked_seeds v (from Maskimage #22)	1
Gray I o Color	Name the primary objects to be identified Cells	?
Saveimages		
Saveimages	Typical diameter of objects, in pixel units (Min,Max) 26 59	?
() C Maskingang		2
() C Image Math	Discard objects outside the diameter range? 🔿 Yes 💿 No	•
() Constant		
		?
() C ImageMath	Discard objects touching the border of the image? O Yes No	
Mackimage	I hreshold strategy Global V	?
IdentifyPrimaryObjects	Thresholding method Manual 🗸	?
© ConvertObjectsTolmage		
Savelmages	Manual threshold 0.05	7
	Threshold smoothing scale 22	?
	Method to distinguish clumped objects Intensity 🗸	?
	Method to draw dividing lines between clumped objects listencity	2
	include to and arrangement champed objects internet v	•
	Automatically calculate size of smoothing filter for declumping?	?
	Adomatically calculate size of shootaling method accompany. In vest O No	
		2
	Automatically calculate minimum allowed distance between local maxima? 🔿 Yes 💿 No	•
	Suppress local maxima that are closer than this minimum allowed distance 13	?
	Speed up by using lower-resolution image to find local maxima?	?
		?
	Display accepted local maxima? 🔿 Yes 💿 No	
	Fill holes in identified objects? After declumping only	?
	Handling of objects if excessive number of objects identified Continue	?
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode		
p childres p ranayze images	Found 7 rows	

		~
CellProfiler 4.1.3: CP413_PeyersPatch TestData_Cell	Jutlines.cpproj* (D:\John\2020\2020\41/_2D_LABEL_FKE_SEGMENTATION_PKOJECTS\2021D522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —	×
File Edit Test Windows Help		
🖾 Images	This would as some the identified all a birth into a income (with d 'CallMarket) when a solution in the source and a source and the source at the intervent is the income field.	
🕑 Metadata	This module converts the identified cell-objects into an image (called Celliviasis) where each pixel is humbered according to the cell-object humber it belongs to in that image-field.	\sim
NamesAndTypes	This image is useful for colouring specific cell-objects for downstream visulisations of the data e.g., immunofluorescence / nearest-cell neighbour visulisations etc.	
Groups		
Inreshold		
Inreshold		~
🔿 🗹 Maskimage		
🔿 🗹 Maskimage	Select the input objects Cells v (from IdentifyPrimaryObjects #23)	?
KescaleIntensity		
	Name the output image	1
	Select the color format uint 16 v	?
Savelmages		
🗇 ビ Maskimage		
🗇 🗹 ImageMath		
🗇 🗹 GaussianFilter		
Interstold		
ImageMath 🗹		
I RemoveHoles		
🗇 🗹 Maskimage		
ConvertObjectsTolmage	4	
Saveimages		
G Savennages		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode		
Analyze Images	Found 7 rows	



CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers	Patch example\MATLAB\2D_PeyersPatch —	о x
File Edit Test Windows Help			
 Images Metadata 	This module overlays the label-free cell segmentation onto a chosen image - here the CD3 immunofluorescence data		^
 NamesAndTypes Groups 	In this way, it allows a visual check of how well the cell segmentation is performing.		
			~
🗇 🗹 Maskimage			
🚸 🕑 Maskimage	Display outlines on a blank image?		?
KescaleIntensity KescaleIntensity	Visplay outlines on a blank image. Ves No		
GravToColor	Select image on which to display outlines View thresh CD3 rescaled v (from GravToColor #12)		?
I GrayToColor			
I Savelmages	Name the output image Segmentation_checker		?
I Savelmages	Outline display mode Color 🗸 🗸		?
Inreshold Mackimage	How to outline Inner		2
			•
🗇 🕑 GaussianFilter			
Inreshold	Select objects to display Cells v (from IdentifyPrimaryObjects #23)		?
	Select outline color		?
KemoveHoles Maskimage	Add another outline		2
	Add another oddine		
I ConvertObjectsTolmage			
🚸 🕑 Savelmages			
OverlayOutlines			
Saveimages Saveimages MeasureObjectSizeShape			
I ExportToSpreadsheet			
Output Settings View Workspace			
? Adjust modules: + - ^ v			
Start Test Mode Analyze Images	Found 7 rows		



CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	Outlines.cpproj* (D:\John\2020\20200417_2D_LABEL_FREE_SEGMENTATION_PROJECT5\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —	
File Edit Test Windows Help		
C Images C Metadata C NamesAndTypes C Groups Ø C Threshold	This module measures shape and size properties for cell-objects	^
		~ ·
🗇 🗹 Maskimage		
🗇 🗹 Maskimage	Cells (from IdentifyPrimaryObjects #23)	?
I RescaleIntensity		
GrayToColor	Select object sets to measure	
🚸 🕑 Threshold		
🗇 ビ Maskimage		
🚸 ビ ImageMath	Calculate the Zernike features?	?
	Conclusion and the second se	
Inreshold ImageMath		?
C RemoveHoles	Calculate the advanced features? O Yes No	
🗇 🗹 IdentifyPrimaryObjects		
I ConvertObjectsTolmage		
Saveimages Saveimages MeasureObjectSizeShape		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode Analyze Images	Found 7 rows	

CellProfiler 4.1.3: CP413_PeyersPatchTestData_Cell	utlines.cpproj* (D:\John\2020\2020417.2D_LABEL_FREE_SEGMENTATION_PROJECTS\20210522_2D_biostudies\2_Final 2D Peyers Patch example\MATLAB\2D_PeyersPatch —		×
File Edit Test Windows Help			
G Metadata	This module measures intensity properties in cell-objects for each of the images specified		^
VamesAndTypes			
Groups	e.g., amount of CDTIc or CD3 expression in each cell object		
			~
🔿 🗹 Maskimage			
🔿 🖸 Maskimage			2
	CellMasks (trom ConvertObjects loimage #24)		:
	Sampatrian chacker (from Owerschuld Hollen #16)		
() C GravToColor	Select images to measure a segmentation_checker (inform OverlayOutlines #20)		
	Select mage to measure		
	View thresh CD3 rescaled (from GravToColor #12)		
	background masked seeds (from MaskImage #22)	~	
🔿 🗹 Maskimage			
	Cells (from IdentifyPrimaryObjects #23)		?
n GaussianFilter	(initial and initial and initiana and initial and initial and initial and initial and init		_
🚸 🖸 Threshold			
🚸 🖸 ImageMath	Select objects to measure		
🗇 ビ RemoveHoles			
🚸 🖸 Maskimage			
IdentifyPrimaryObjects			
I ConvertObjectsTolmage			
I Savelmages			
International States (Second Second S			
I Savelmages			
I MeasureObjectSizeShape			
🗇 🗹 MeasureObjectIntensity			
I ExportToSpreadsheet			
Output Settings View Westerness			
Adjust medules			
r Adjust modules: + - A v			
Start Test Mode Analyze Images	Found 7 rows		



3-D CellProfiler pipeline. This section presents screenshots of a CellProfiler image analysis pipeline used to achieve label-free cell segmentation in 3-D from the Unet network outputs and to measure the intensity and size/shape features of identified cell-objects. To use the image analysis pipeline with new image data, the 'IdentifyPrimaryObjects' module simply needs adjusting so that the 'typical diameter of objects' size-range matches the pixel scaling of the new images. For newcomers to CellProfiler, we recommend downloading the image-data and pipeline from BioStudies database https://www.ebi.ac.uk/biostudies/ under accession number S-BSST742. This enables the pipeline to be run with the data described in the manuscript and allows the user to see how each module works.

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D):\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\ – 🔲	×
File Edit Test Windows Help		
 CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D File Edit Test Windows Help Mages Metadata NamesAndTypes Groups Converbolight SaveImages SaveImages SaveImages ConvertObjectsToImage GaussianFilter ImageMath RemoveHoles SaveImages SaveImages SaveImages SaveImages ExportToSpreadsheet 	Auchin 2020/2020017, 3D_LABEL_FREE_SEGMENTATION_PROJECTS/20210325_biostudies/2_Final 3D MLN example/MATLAB/3D_MLN/S_CellProfiler_cell_measurements/1_Structured channel 1 = FOXP3 channel 3 = EX-Actin probability map channel 4 = LF-Actin probability map chan	X test_dat test_dat test_dat test_dat ? ? ?
Output Settings View Workspace ? Adjust modules: + - > > Analyze Images		
Adjust modules: + - A V Start Test Mode Analyze Images	Found 5 rows	



CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\... Х

_

File Edit Test Windows Help	:Vonn/2020/2020041/_3D_LABEL_FREE_SEGMENTATION_PROJECTS/20210525_biostudies/2_Final 3D MLN example/MATLAB/3D_MLN/55_CellProfiler_cell_measurements/ —	
☑ Images ☑ Metadata ☑ NamesAndTypes ☑ Groups ④ ☑ Threshold ④ ☑ MaskImage	The NamesAndTypes module allows you to assign a meaningful name to each image by which other modules will refer to it.	< >
 	Assign a name to Images matching rules ?	^
 	Relative pixel spacing in X 0.207	
 	Relative pixel spacing in Y 0.207 ? Putting rind participation in Z 0.150 2	
 ♥ ☑ ConvertObjects I olmage ♥ ☑ SaveImages ♥ ☑ MeasureObjectIntensity 	Match All v of the following rules ?	
 	Metadata v Does v Have channel matching v 1 - + Name to assign these images c1 ?	
	Select the image type Grayscale image V ? Duplicate this image ?	ľ
	Match All of the following rules ? Metadata V Does V Have channel matching V 2 - +	
	Name to assign these images c2 ? Select the image type Gravscale image	
	Duplicate this image ? Remove this image ?	
	Select the rule criteria Match All v of the following rules ? Metadata v Does v Have channel matching v 3 - +	~
	Update c1 c2 c3 c4 c5 1 Ch_1_m_001.tiff Ch_2_lm_001.tiff Ch_3_lm_001.tiff Ch_4_lm_001.tiff Ch_5_lm_001.tiff	
Output Settings View Workspace		
Start Test Mode Analyze Images	Found 5 rows	

r.



CellProfil	er 4.1.3: CP413	_3D_MLN_pipeline.cpproj (E	3:Vohn/2020/20200417_3D_LABEL_FRE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\ — 🛛	×
File Edit T	Fest Window	s Help		
	Images		This module creates a binary image of Channel 1 (c1) - the FOXP3 immunofluorescence data	~
6	Metadata	T		
G	Groups	rypes		
<u>م</u> و	Threshold			
0 0	Maskimage			
() C	Savelmage			
1	Savelmages		Select the input impediate (from Nomer And Tuner)	2
() C	Watershed		select the input image c1 v (from NamesAnd types)	
ی 🗇	ConvertObj	ectsTolmage	Name the output image mask_c1	?
() C	GaussianFil	ter		
Image: Contract of the second seco	5 Threshold		Infeshold strategy Global V	1
Image: Contract of the second seco	/ ImageMath		Thresholding method Manual V	?
	KemoveHo Watarahad	es	Manual threshold 0.3	?
0 C	ConvertObi	ectsTolmage		_
@ C	Savelmages	eetsronnage	Threshold smoothing scale 0.0	?
¢ ۲	MeasureOb	jectIntensity		
() C	MeasureOb	jectSizeShape		
ا 🗇	ExportToSp	readsheet		
0.	and Catting	16		
? Ac	diust modules	+ - ^ v		
🕪 Start T	est Mode	Analyze Images	Found 5 rows	_

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D:\John\2020\20200417.3D_LABEL_FREE_SEGMENTATION_PROJECT5\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\		×
File Edit Test Windows Help			
🕑 Images			_
🖾 Metadata	Applies the mask created in the first module to create a background subtracted greyscale image of the FOXP3 immunofluorescence signal	· · · · · · · · · · · · · · · · · · ·	\sim
NamesAndTypes			
🖸 Groups			
🗇 🕑 Threshold			
🗇 🖸 Maskimage			\sim
I Savelmages			
I Savelmages	Select the input image c1 v (from NamesAndTypes)	?	
I Watershed			
	Name the output image Thresh_c1	?	
GaussianFilter	Use objects or an image as a mask? Image	?	1
C Inreshold	integer and integer an		4
	Select image for mask mask_c1 v (from Threshold #05)	?	
Watersbed		2	1
ConvertObjectsTolmage	Invert the mask? O Yes No		1
I MeasureObjectSizeShape			
I ExportToSpreadsheet			
Output Settings View Workspace			
? Adjust modules: + - ^ v			
ID Start Task Mark			
Analyze Images	Found 5 rows		





🔽 Ce	llProfile	r 4.1.3: CP413	_3D_MLN_pipeline.cpproj (I	:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\	- 🗆 ×
File E	dit Te	est Window	s Help		
	8 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8 9 8	Images Metadata NamesAnd Groups	Гуреs	identifies nuclear 'seeds' from the label-free nuclei channel outputted from the UNET	^
	\$ 5	Maskimage			~
	\$ U \$	Savelmages Savelmages		Select the input image c3	?
	\$ U \$	ConvertObj	ectsTolmage	Name the output object Nuclei	?
	\$ 5	Threshold	er	Generate from Distance 🗸	?
	\$ 5	ImageMath RemoveHol	es	Footprint 8	?
	\$ 6 \$ 6	Watershed ConvertObj	ectsTolmage	Downsample 1	?
	\$ 6 \$ 6	Savelmages MeasureOb	jectIntensity		
	\$ 6 \$ 6	MeasureOb ExportToSp	jectSizeShape readsheet		
	Outp	ut Settings	View Workspace		
?	Adj	ust modules:	+ - ^ v		
₽	Start Te	est Mode	Analyze Images	Found 5 rows	

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cppr	oj (D:\John\2020\2020417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\202110525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\	- 0	×
File Edit Test Windows Help			
C Images C Metadata C NamesAndTypes	Converts the segmented nuclear seeds into an image		^
Groups Groups Groups			
🔿 🗹 Maskimage			\sim
	Select the input objects Nuclei v (from Watershed #09)		- 7
	Name the output image Nuclei_Image		?
	Select the color format uint 16 v		?
🗇 🗹 RemoveHoles			
I MeasureObjectIntensity			
W Export rospreadsneet			
Output Settings View Workspace			
? Adjust modules: + - ^	V I I I I I I I I I I I I I I I I I I I		
Start Test Mode Analyze Images	County Steering		
	Found 5 rows		

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\ 🛛 🗙				×
File Edit Test Windows Help				
🕑 Images				
🕑 Metadata	Smooths the background/	ther probability map created by the UNET		^
NamesAndTypes				
🖾 Groups				
🗇 🗹 Threshold				
🔿 🗹 Maskimage				~
Savelmages				
Saveimages Watershed	Select the input image	c5 v (from NamesAndTypes)		?
ConvertObjectsTolmage	Name the output image	Tissue Mask smoothed		2
	Nume the output image			
🗇 🗹 Threshold	Sigma	3		?
ImageMath	L			-
I RemoveHoles				
🗇 🗹 Watershed				
🔿 🗹 ConvertObjects Folmage				
Saveimages MassuraObjectIntensity				
MeasureObjectimensity MeasureObjectSizeShape				
Octored Settliness Minutes				
Output Settings View Workspace				
? Adjust modules: + - ^ v				
Start Test Mode Analyze Images	Found 5 rows			

Copyer Steins Very S	CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D File Edit Test Windows Help	D:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\ —	
Very Stering Very Stering <th>び Images び Metadata び NamesAndTypes び Groups</th> <th>creates a binary image of the smoothed 'background/other' probability map - identifying the region of the image volume occupied by tissue</th> <th>^</th>	び Images び Metadata び NamesAndTypes び Groups	creates a binary image of the smoothed 'background/other' probability map - identifying the region of the image volume occupied by tissue	^
 C Surchrages C Surchrages C Vateshed C Vateshed<td> </td><td></td><td>~</td>	 		~
ConvertSpetting ConvertSpetting Verter Magdath ConvertSpetting ConvertSpetting Verter Magdath ConvertSpetting ConvertSpe	の ビ Savelmages の ビ Savelmages の ビ Watershed	Select the input image Tissue_Mask_smoothed \checkmark (from GaussianFilter #11)	?
Count of line The choid stategy (acbit)		Name the output image TissueMask	?
0 ImageNulfi Threshold smoothing settion 0	 GaussianFilter GuissianFilter	Threshold strategy Global 🗸	?
Output Settings Year Workspace 2 Adjust module:		Thresholding method Minimum Cross-Entropy 🗸	?
© ConvertObject: Torinage Threshold correction factor 10 ? © G MasureObject: NetWorkspee © 10 ? ? © G EportToSpreadheet © 10 ? ? © Log zastings Yiew Workspee ? ? ? Adjuit module: • • • • • •	 ♥ ☑ RemoveHoles ♥ ☑ Watershed 	Threshold smoothing scale 0.0	?
Output Setting: View Workspace Image: Adjust module: Image: Adjust module: Adjust module: Image: Adjust module:		Threshold correction factor 1.0	?
 C S Massure Object Stellsuppe Centre du de la mais de	 	I ower and unner hounds on threshold 0.0 10	2
Confunct Setting: View Workspace Adjust module: Adjust module: Adjust module: A v			. 2
Output Settings View Workspace ? Adjust modules • ^		Log transform before thresholding? O Yes No	•
Output Settings View Workspace ? Adjust modules: + -			
	? Adjust modules: + - ^ v		
Start Test Mode Analyze Images Found 5 rows	Start Test Mode Analyze Images	Found 5 rows	

Г

 Images Metadata NamesAndTypes Groups 	Inverts the TissueMask such that it can be used to limit the cell segmentation to the space occupied with cells	
 		
 	Operation Invert V	?
 ♥ ConvertObjectsTolmage ♥ CaussianFilter ♥ Threshold 	Name the output image Inverted_TissueMask	?
	Select the first image TissueMask \checkmark (from Threshold #12)	?
 	Multiply the first image by 1.0	?
SaveImages SaveImages MeasureObjectIntensity	Raise the power of the result by 1.0	?
	Multiply the result by 1.0	?
Export i ospreadsneet	Add to result 0.0	?
	Set values less than 0 equal to 0? Yes O No	?
	Set values greater than 1 equal to 1?	?
	Replace invalid values with 0?	?
	Ignore the image masks? O Yes	?
Output Settings View Workspace		
Adjust modules: + - ^ ,	<i>i</i>	
Start Test Mode	From d From	

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj	(D:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\	×
File Edit Test Windows Help		
ビ Images ビ Metadata ビ NamesAndTypes ビ Groups	removes small holes from the TissueMask	^
 グ ビ Threshold グ ビ Maskimage ダ ビ Savelmages 		~
 	Select the input image Inverted_TissueMask v (from ImageMath #13)	?
	Name the output image Inverted_TissueMask_deholed	?
 	Size of holes to fill 200	<u> </u>
 		
ConvertObjectsTolmage Savelmages ManuaryObjectIntersity		
VieasureObjectIntensity VieasureObjectSizeShape VieasureObjectSizeShape VieasureObjectSizeShape		
Output Settings View Workspace		
? Adjust modules: + - ^ v		
Start Test Mode Analyze Images	Found 5 rows	

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D File Edit Test Windows Help	:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\	
 ☑ Images ☑ Metadata ☑ NamesAndTypes ☑ Groups 	Marker controlled watershed using the previously segmented label-free nuclei as seeds to now find the cell outlines from the label-free actin probability map	^
 Ketadata NamesAndTypes Groups Threshold MaskImage SaveImages SaveImages SaveImages GussianFitter GouvertObjectsToImage KernoveHoles ConvertObjectsToImage SaveImages MasureObjectsToImage MeasureObjectSizeShape KeyportToSpreadsheet 	Select the input image of the processing regimented user included as seen to non-inductive second regiments and included as seen included as seen included in processing improvements in the cut of the second regiments and included as a second of non-inductive second regiments and included as a second of non-inductive second regiments and included as a second of non-inductive second regiments and included as a second of non-inductive second regiments and included as a second of non-inductive second regiments and included as a second regiment of non-inductive second regiments and included as a second regiment of non-inductive second regiments and included as a second regiment of non-inductive second regiments and included as a second regiment of non-inductive second regiments and included as a second regiment of non-inductive second regiments and	
Output Settings View Workspace ? Adjust modules: + - ^ v Ib Start Test Mode > Analyze Images	Found 5 rows	

Г

CellProfiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (D:\John\2020\20200417_3D_LABEL_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_MLN\5_CellProfiler_cell_measurements\ —		×
File Edit Test Windows Help			
び Images び Metadata び NamesAndTypes	Converts segmentation results to 'CellMasks'. These are image volumes where each voxel contains the identity of the cell it belongs to		^
Groups			
 			~
 	Select the input objects Cells v (from Watershed #15)		?
 	Name the output image CellMasks		?
 ⑦ ☑ Threshold ⑦ ☑ ImageMath 			:
KemoveHoles Watershed			
Savelmages Konvertobjectstotmage			
Viewsdreobjectmensky ViewsdreobjectSizeShape ViewsdreobjectSizeShape ViewsdreobjectSizeShape			
() C capacitospicadance			
Output Settings View Workspace			
? Adjust modules: + - ^ v			
Start Test Mode Analyze Images	Found 5 rows		


🔽 Cell	Profiler 4.1.3: CP413_3D_MLN_pipeline.cpproj (l	D:\John\2020\20200417_3D_LA	L_FREE_SEGMENTATION_PROJECTS\20210525_biostudies\2_Final 3D MLN example\MATLAB\3D_N	/ILN\5_CellProfiler_cell_measurements\ —		×
File Ed	lit Test Windows Help					
	© Images © Metadata © NamesAndTypes © Groups ♥ © Threshold	measures cell intensities u	ng the selected images to measure		,	~
	Image Image	Select images to measure	CellMasks (from ConvertObjectsTolmage #16) Inverted_TissueMask (from ImageMath #13) Inverted_TissueMask_deholed (from RemoveHoles #14) Nuclei_Image (from ConvertObjectsTolmage #10) Z Thresh_c1 (from MaskImage #06) TissueMask (from Threshold #12) Tissue_Mask_smoothed (from GaussianFilter #11)		?]
		Select objects to measure	Cells (from Watershed #15) Nuclei (from Watershed #09)		?]
2	Output Settings View Workspace					
IÞ S	tart Test Mode Analyze Images	Found 5 rows				

 Images Metadata NamesAndTypes Groups 	measures cell-object size and sh	ape properties		
Savelinages Savelinages <t< th=""><th>Select object sets to measure</th><th>☑ Cells ☐ Nuclei</th><th>(from Watershed #15) (from Watershed #09)</th><th></th></t<>	Select object sets to measure	☑ Cells ☐ Nuclei	(from Watershed #15) (from Watershed #09)	
ConvertObjects Folmage Savelmages MeasureObjectIntensity	Calculate the Zernike features?	⊖Yes ◉No		
C ExportToSpreadsheet	Calculate the advanced features?	Ves No		
Output Settings View Workspace				

Г

