COMPARE
CENTRE OF MEMBRANE PROTEINS AND RECEPTORS

Intermediate Image Analysis Exercises

Section 1: Deconvolution and PSF Generation

A. Deconvolution of fluorescent bead using DeconvolutionLab2

- Install DeconvolutionLab2 from http://bigwww.epfl.ch/deconvolution/deconvolutionlab2/ [1]. You need to download DeconvolutionLab_2.jar and place it in plugins folder of Fiji.

- Restart Fiji. Open bead.tif and beadPSF.tif[1]. The former is an image stack of a large single fluorescent bead with 2.5µm diameter. The latter is the corresponding point spread function (PSF).

- Use DeconvolutionLab2 (*Plugins > DeconvolutionLab2 > DeconvolutionLab2 Lab*) to deconvolve the bead:

  - First drag and drop the bead.tif file to the *Image* tab.
  - Drag and drop the beadPSF.tif file to the *PSF* tab.
  - Under the *algorithm* tab try using *Regularized Inverse Filtering* in the first instance.
  - Click *run* to perform the deconvolution.

- Also try the Naïve Inverse Filter and Richardson Lucy algorithms (10 iterations).

- Using *Image > Stacks > Orthogonal Views* compare the deconvolution results to the raw data. Do you think there is any improvement and which algorithm performs best?

- Deconvolve the data again using the iterative *Richardson-Lucy* algorithm but with 100 iterations (this may take a while!). Can you see any difference?

- Install the PSF Generator plugin from http://bigwww.epfl.ch/algorithms/psfgenerator/ [2].

- Use the PSF Generator (*Plugins > PSF Generator*) to generate a theoretical PSF for the bead data.

  - Try the *Born and Wolf* optical model but feel free to experiment with others.
  - Fill in as many parameters as possible using the acquisition information available at http://bigwww.epfl.ch/deconvolution/bead/.
  - Ensure the output PSF is the same size (*Size XYZ*) as the bead stack (70X70X41).

- Deconvolve the data using DeconvolutionLab2 and your theoretical PSF.

[1] Both are cropped versions of the dataset available at http://bigwww.epfl.ch/deconvolution/bead/.

B. Challenge: C. Elegans Embryo

- Deconvole the C. Elegans dataset (CElegans-CY3.tif) using DeconvolutionLab2 and the provided PSF (CElegans-CY3-PSF.tif)[2]. Try *Regularized Inverse Filtering* and iterative *Richardson-Lucy*. Which works best and which is fastest?

    - Note the dataset is quite large and computation of the Richardson-Lucy deconvolution may take some time. Your will also need a substantial amount of RAM. If this is the case you can open CElegans-CY3_RL_100.tif which contains the results of a Richardson-Lucy deconvolution with 100 iterations.

Section 2: Deep learning-based denoising

A. Denoising scanning electron microscopy (SEM) data using a self-supervised Noise2Void (N2V) model [3]

- Install the N2V Fiji plugin. To do this add the CSBDeep update site and restart Fiji, see https://imagej.net/plugins/n2v for detailed instructions.

- The n2v_2D directory contains a N2V model which was trained on train.tif[3] using a GPU within a python notebook[4]. Load this pretrained model via *File -> Import -> bioimage.io.zip* and select the n2v_2D/export.bioimage.io.zip file.

    - This will load a window called "model" which allows you to inspect the network metadata/training parameters and run prediction.
    - Alternatively you can train your own model on this data using the N2V Fiji plugin but without GPU integration this will take a long time so this is not recommend for this workshop. See https://www.youtube.com/watch?v=DJw2VMlDsME&ab_channel=DeborahSchmidt for a demo of this.

- Open both train.tif (the image used to train the model) and validation.tif (an image from the same dataset not seen during training). Pass these images through the N2V model using the *Predict -> Single Image or Stack* option. Axes of prediction input should by XY.

    - Visually inspect the results. Do you think they are superior to what you could achieve with classical denoising (e.g. a Gaussian blur)? Does the network produce reasonable results on both the training and validation data?

[2] This image volume is part of a three channel dataset available at http://bigwww.epfl.ch/deconvolution/bio/.

[3] Data by Reza Shahidi and Gaspar Jekely, Living Systems Institute, Exeter. Available at https://download.fht.org/jug/n2v/SEM.zip.

[4] Available at https://github.com/juglab/n2v/tree/main/examples/2D/denoising2D_SEM.

Section 3: Segmentation

A. Simple Filtering Approach

- Open segmentation_example_1.tif in Fiji [4]. Duplicate the data, apply a Gaussian filter (*Process > Filters > Gaussian Blur…*). Next apply an automated global threshold (*Image > Adjust > Threshold*) to produce a binary segmentation image.

- Add the segmentation contour to the ROI manager. *Edit > Selection > Create Selection* followed by *Edit > Selection > Add to Manager*. Rename the ROI as something sensible and display on the original data, how well has the segmentation worked?

- Create a macro using the command recorder (*Plugins > Macros > Record…*) to perform the above segmentation protocol. Run this macro on segmentation_example_2.tif. Does the segmentation protocol perform similarly for this image?

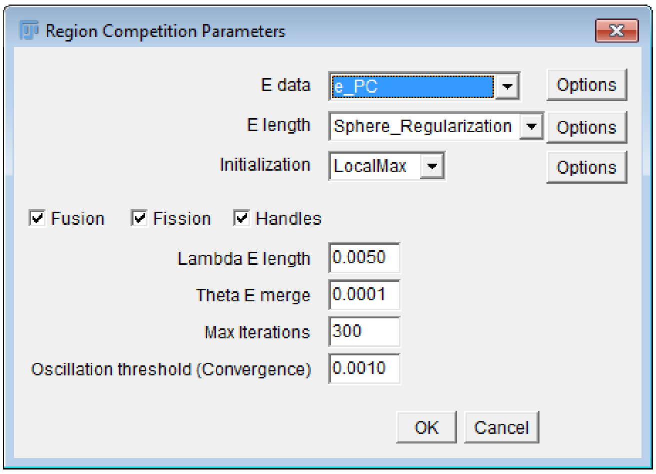
B. Region Competition: A Model Based Approach

- Install the MosaicSuite for Fiji. First select *Help > Update*. Then select *Manage Update Sites* and check *MOSAIC ToolSuite*. Click *Close* and then *Apply Changes*. Restart Fiji.

- More information about the MoasicSuite can be found at http://mosaic.mpi-cbg.de/?q=downloads/imageJ. A manual for the Region Competition plugin can be found at http://mosaic.mpi-cbg.de/MosaicToolboxSuite/Region_Competition.html [5]. Take a couple of minutes to look though some of this information.

- Open segmentation_example_1.tif and segment the nuclei using the Region Competition plugin (*Plugins > Mosaic > Segmentation > Region Competition*).

  ➢ Open the *Parameters* interface. Set the following parameters:



Region Competition Parameters

E data: e_PC

E length: Sphere_Regularization

Initialization: LocalMax

☑ Fusion ☑ Fission ☑ Handles

Lambda E length: 0.0050

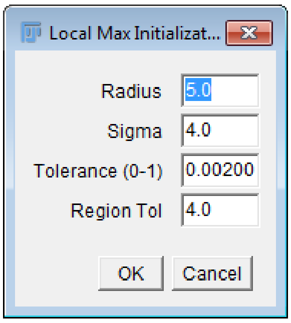
Theta E merge: 0.0001

Max Iterations: 300

Oscillation threshold (Convergence): 0.0010

Note e_PC uses a piecewise constant model. Therefore we model every nucleus as having a constant intensity but the intensity can vary between nuclei. For more information on each parameter refer to the online manual.

  ➢ Under *Initialization Options* set the following:



Radius 5.0
Sigma 4.0
Tolerance (0-1) 0.00200
Region Tol 4.0

A localMax initialization will place seeds at local maxima in the image. The options set here set how these local maxima are found.

- ➢ Finally, return to the main interface, choose the correct image from the dropdown menu and hit *Ok* to run.

- Covert the labelled output image to a binary segmentation. To do this set a manual threshold of 1 (*Image > Adjust > Threshold*). Add the segmentation contour to the ROI manager and rename the ROI as something sensible (*Edit > Selection > Create Selection* followed by *Edit > Selection > Add to Manager*). Overlay this ROI on the original data. How does this segmentation compare to the simple filtering approach from the part A?

- Run the Region Competition plugin on segmentation_example_2.tif using the same parameters. Does the plugin perform similarly for this image?

C. Pixel Classification with ilastik: A Machine Learning Approach

- Download and install ilastik (if not already installed) from http://ilastik.org/download.html [6].

- Open ilastik and create a new *Pixel Classification* project. Using the *Input Data* tab add the 5 training images from the *Training data for ilastik* folder. More information about this step can be found at http://ilastik.org/documentation/basics/dataselection.

- Create a classifier to segment the nuclei. To do this work through the pixel classification tutorial available at http://ilastik.org/documentation/pixelclassification/pixelclassification.

- After creating a classifier use the *Batch Processing* tab to produce segmentations for segmentation_example_1.tif and segmentation_example_2.tif.

  - ➢ To export a segmentation make sure you have selected *Simple Segmentation* from the *Source* dropdown menu in the *Prediction Export* tab.
  - ➢ Click *Choose Image Export Settings …*. Select *Format: .tif* and *Convert to Data Type: unsigned 8-bit.*

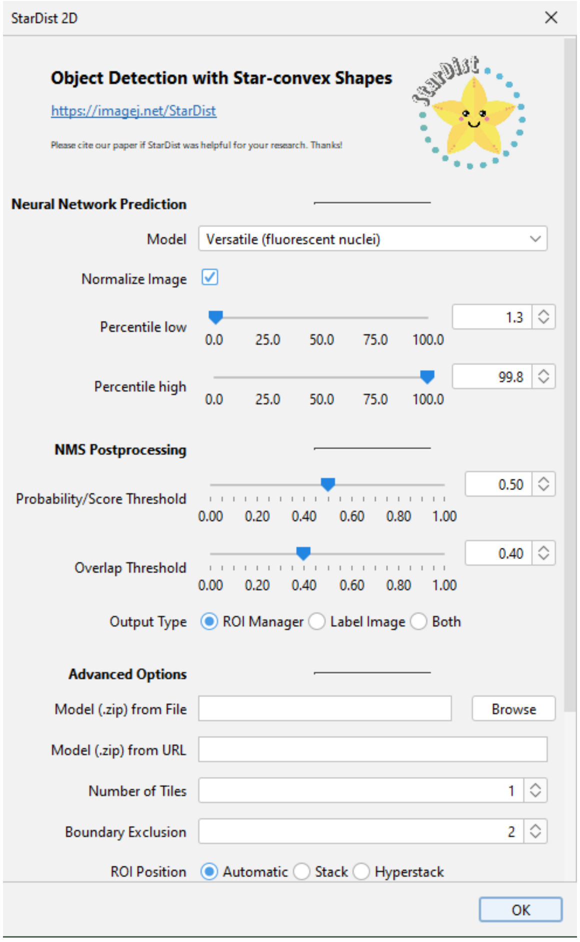
- Using Fiji open the segmentation results produced by ilastik. Convert to binary by setting a trivial threshold and add the outline to the ROI manager. Compare to the results of the model based and filtering approaches from Sections A and B.

D. Instance Segmentation of Nuclei using StarDist [7]

- Install the StarDist Fiji plugin. To do this add the CSBDeep and StarDist update sites before restarting Fiji, see https://imagej.net/plugins/stardist for detailed instructions.

- Open one of the images from the StarDist\BBBC008_v1_images[5] directory. These contain fluorescently labelled nuclei which in some cases are tightly clustered making instance segmentation difficult. Use a pretrained StarDist model (*Plugins -> StarDist -> StarDist 2D*) to segment the nuclei:

  - ➢ Use the "Versatile (fluorescent nuclei)" pretrained model.
  - ➢ You can output ROIs to the ROI manager and/or a labelled image.
  - ➢ Keep other parameters as default.



StarDist 2D

Object Detection with Star-convex Shapes

https://imagej.net/StarDist

Please cite our paper if StarDist was helpful for your research. Thanks!

Neural Network Prediction

Model: Versatile (fluorescent nuclei)

Normalize Image: ☑

Percentile low: 1.3

Percentile high: 99.8

NMS Postprocessing

Probability/Score Threshold: 0.50

Overlap Threshold: 0.40

Output Type: ● ROI Manager ○ Label Image ○ Both

Advanced Options

Model (.zip) from File:

Model (.zip) from URL:

Number of Tiles: 1

Boundary Exclusion: 2

ROI Position: ● Automatic ○ Stack ○ Hyperstack

  - ➢ Visually inspect the results. Has it done better than other methods you have tested in the course.

- Open one of the images from the StarDist\MoNuSegTestData[6] directory. These contain H&E histology data. Use StarDist and the "Versatile (H&E nuclei)" pretrained model to segment the nuclei.

[5] Images from the Broad Bioimage Benchmark Collection available at https://bbbc.broadinstitute.org/.

[6] Data from the MoNucSeg Grand Challenge, available at https://monuseg.grand-challenge.org/Data/.

E. Instance Segmentation of Cells using CellPose [8]

- Use the Cellpose webapp (https://www.cellpose.org/) to segment the cellpose/cells_with_nuclear_stain.png and cellpose/cells_without_nuclear_stain.png images[7].

- Use of the Cellpose GUI and command line interface will be shown in a demo.

F. Challenge: batch processing macro using StarDist

- Create a macro which can batch process the data in either StarDist\MoNuSegTestData and/or StarDist\BBBC008_v1_images. The macro should:

    - ➢ Use StarDist to segment the nuclei.
    - ➢ Create and save csv files containing measurements (e.g area and position) for each nuclei. There should be one output file for every input image.
    - ➢ Also create a summary table containing the number of nuclei in each image and the mean nuclei area.
    - ➢ Save a visual representation of each segmentation (labelled images or ROI outlines) as png files.

- This exercise may be difficult if you are new to macro programming. If this is the case a potential solution (batch_stardist.ijm) is available from the course website. You can download this and make sure you understand all the steps.

Section 4: Tracking with TrackMate

A. TrackMate Basics and Documentation

- TrackMate is as single particle tracking plugin for Fiji/ImageJ [9]. Take a few minutes to read through the information available at http://imagej.net/TrackMate.

- Complete the *Getting Started with TrackMate* tutorial available at http://imagej.net/Getting_started_with_TrackMate.

B. Tracking Nuclei in Live-Cell Time-lapse Data

- TimeLapse1.tif and TimeLapse2.tif contain time-lapse movies of HeLa cells expressing H2b-GFP. These datasets are publically available at http://www.codesolorzano.com/celltrackingchallenge/Cell_Tracking_Challenge/Datasets.html [4].

- Use what you have learnt from the tutorial to track the cells in TimeLapse1.tif and TimeLapse2.tif. Some hints:
    - ➢ The *Downsample LoG Detector* is good for larger objects (> 20 pixels).
    - ➢ What constraints do the biology of the system impose. Should track merging and/or splitting be allowed?

[7] Images from the Cellpose test data available at https://www.cellpose.org/.

- The time-step between frames is 30 minutes. This may help you to decide on reasonable distances for track-linking etc.

Section 5: Macros for 3D Colocalization Analysis

A. Calculating the Manders' coefficients with a macro

- Open and inspect coloc_simulated/Set1_ColocalizationSimulation_0.tif. This is a two channel 3D volume of spot like objects where roughly 70% of the objects are perfectly colocalizing. We know this because the data is simulated.

- Open the coloc_simulated/simple_coloc_macro.ijm macro within the script editor. This macro will ask the user to specify a two-channel file. The macro will then segment the signal in each channel (user a Gaussian filter and Otsu thresholding) and use these segmentations to calculate the Manders' coefficients. Have a look through the code and see if you can understand all the steps.

- Run the macro on Set1_ColocalizationSimulation_0.tif and inspect the results. Do the values calculated for the Manders' coefficient seem reasonable and match the data (roughly 70% colocalized).

B. Calculating the Manders' coefficients with the Coloc 2 plugin

- Open coloc_simulated/Set1_ColocalizationSimulation_0.tif and split the channels (*Image -> Color -> Split Channels*).

- Run the Coloc 2 plugin channels (*Analyze -> Colocalization -> Coloc 2*). Select the two channels and use the Costes' method for threshold regression. Select the Manders' correlation. For our purposes all other tick-boxes are optional and can be deselected.

- Inspect the Manders' coefficients calculated by Coloc 2. Are these consistent with those calculated by the macro, do they seem reasonable and match the data? If not why might this be the case?

C. Adapting the macro for batch processing

- Open the simple_coloc_macro.ijm macro within the script editor and adapt the macro for batch processing. A template for batch processing can be found at *Templates->Image J1.x -> Batch -> Process Folder (ImageJ macro)* which may help to speed this process up. You can also find a solution on the course website.

- Use this macro to batch process all images within the *coloc_simulated* directory.

References

[1] Sage D, Donati L, Soulez F, Fortun D, Schmit G, Seitz A, et al. DeconvolutionLab2 : An Open-Source Software for Deconvolution Microscopy. Methods 2017; 115: 28-41.

[2] Kirshner H, Aguet F, Sage D, Unser M. 3-D PSF fitting for fluorescence microscopy: implementation and localization application. Journal of Microscopy 2013;249:13–25.

[3] Krull A, Buchholz TO, Jug F. Noise2void-learning denoising from single noisy images. InProceedings of the IEEE/CVF conference on computer vision and pattern recognition 2019 (pp. 2129-2137).

[4] Ma\vska M, Ulman V, Svoboda D, Matula P, Matula P, Ederra C, et al. A benchmark for comparison of cell tracking algorithms. Bioinformatics 2014;30:1609–17.

[5] Cardinale J, Paul G, Sbalzarini IF. Discrete region competition for unknown numbers of connected regions. IEEE Transactions on Image Processing 2012;21:3531–45.

[6] Sommer C, Straehle C, Köthe U, Hamprecht FA. Ilastik: Interactive learning and segmentation toolkit. 2011 IEEE international symposium on biomedical imaging: From nano to macro, 2011, p. 230–3.

[7] Schmidt U, Weigert M, Broaddus C, Myers G. Cell detection with star-convex polygons. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11 2018 (pp. 265-273). Springer International Publishing.

[8] Stringer C, Wang T, Michaelos M, Pachitariu M. Cellpose: a generalist algorithm for cellular segmentation. Nature methods. 2021 Jan;18(1):100-6.

[9] Tinevez J-Y, Perry N, Schindelin J, Hoopes GM, Reynolds GD, Laplantine E, et al. TrackMate: An open and extensible platform for single-particle tracking. Methods 2016.