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. The former is an image stack of a single fluorescent bead with 2.5µm diameter. The latter is the corresponding point spread function (PSF). Both are cropped versions of the dataset available at http://bigwww.epfl.ch/deconvolution/bead/.

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

    - ➢ First drag and drop bead.tif file to the *Image* tab.
    - ➢ Drag and drop beadPSF.tif 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 but 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.

### B. Challenge: C. Elegans Embryo

- Try to deconvolve a 3 channel C. Elegans dataset using DeconvolutionLab2. You will need to download the dataset which can be found at http://bigwww.epfl.ch/deconvolution/bio/.

    - Note the dataset is quite large and computation of the deconvolution may take some time. Your will also need a substantial amount of RAM.

# Section 2: Segmentation

## A. Simple Filtering Approach

- Open segmentation_example_1.tif in Fiji [3]. 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?

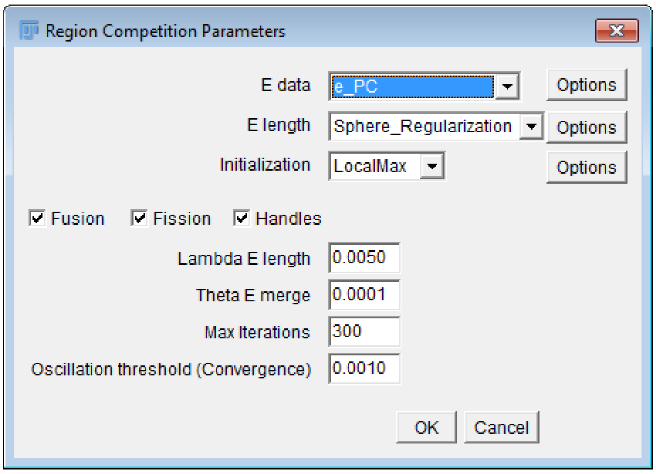
## B. Region Competition: A Model Based Approach

- 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 [4]. 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

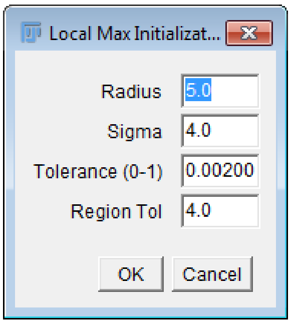
Theta E merge: 0.0001

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:



Local Max Initializat...

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 [5].

- 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.

# Section 3: Pixel Based Colocalization Analysis

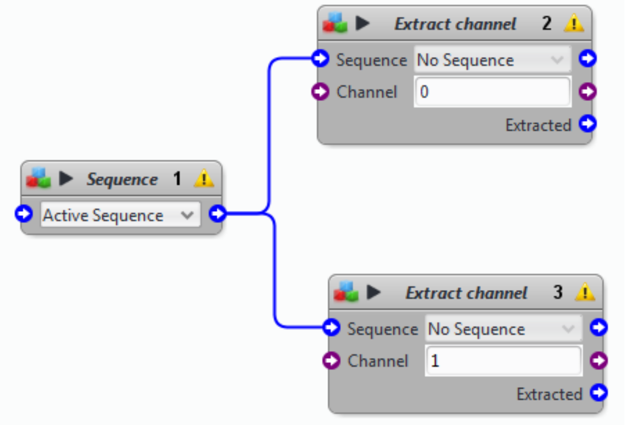
## A. Create an Icy Protocol for 3D Colocalization Analysis

- Open Icy and install the *Colocalization Studio* plugin [6]. Installing plugins in Icy is super easy, simply search for it in the search bar and select! For more information on this plugin see http://icy.bioimageanalysis.org/plugin/Colocalization_Studio.

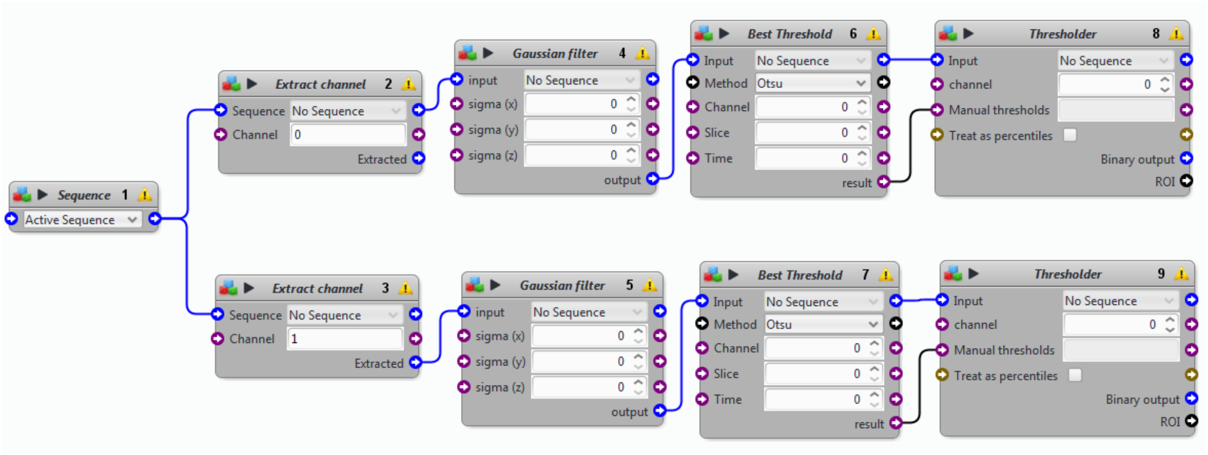
- Navigate to the SmulatedData folder and open Set1_ColocalizationSimulation_0.tif which contains a 3D two channel simulated image stack. All of the image stacks in this folder where created using the *Colocalization Simulator* plugin [7].

- Create an Icy protocol (*Tools > Protocols*) to quantify the Manders Coefficients (M1 and M2) for the image stack [8]. An in-depth guide to protocols can be found at http://icy.bioimageanalysis.org/doc/icy-protocols.pdf.

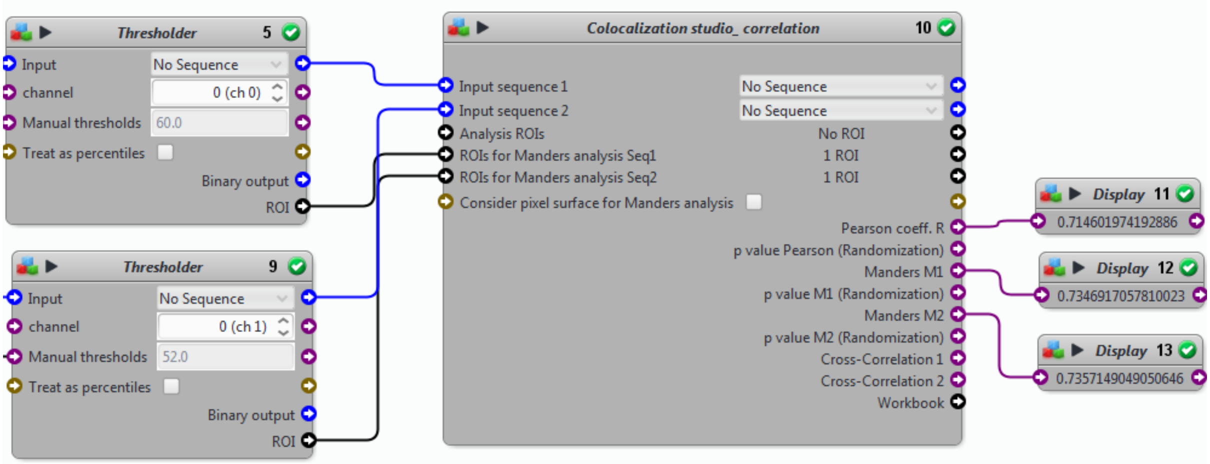
  - ➢ First add a *Sequence* block to read the active sequence (window). Blocks can be found by right clicking or through the search tool.
  - ➢ Add two *Extract Channel* blocks to extract the data for each channel. These blocks should be linked to the *Sequence* block as shown below.



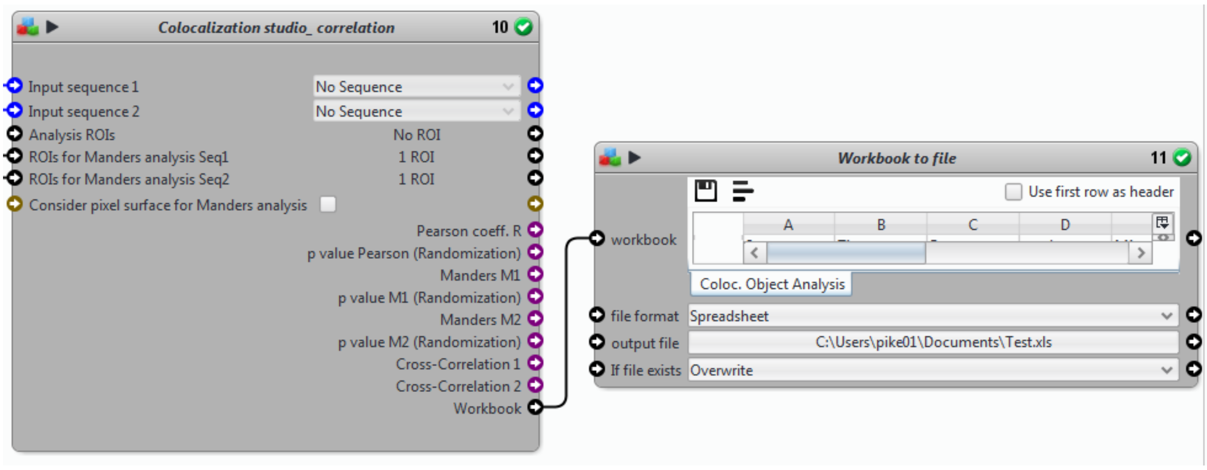
  - ➢ Add a pre-processing step (or steps) to each channel. For example use a *Gaussian filter* block to apply a 3D Gaussian blur.
  - ➢ Next apply an automated global threshold to isolate the signal in each channel. For example an Otsu threshold. Note the *Best Threshold* block finds a threshold value using the specified method, and the *Thresholder* block applies it to produce either a binary output or ROI.



- ➢ Having performed data pre-processing and signal isolation we are ready to calculate some colocalization statistics. This can be done with the *Colocalization studio_correlation* block. Note we do not need any analysis ROIs for the simulated data but for real data this will probably be needed. For example a nuclear or cellular segmentation
- ➢ The *Display* block is very useful and can be used to display the output of other blocks.



- ➢ The *Workbook* output of the *Colocalization studio_correlation* block contains a table with the calculated colocalization statistics. Save this as a spreadsheet using the *Workbook to file* block. You should specify an output folder and file name with the correct extension (e.g colocalizationOutput.xls).


Colocalization studio_ correlation 10
Input sequence 1 No Sequence
Input sequence 2 No Sequence
Analysis ROIs No ROI
ROIs for Manders analysis Seq1 1 ROI
ROIs for Manders analysis Seq2 1 ROI
Consider pixel surface for Manders analysis
Pearson coeff. R
p value Pearson (Randomization)
Manders M1
p value M1 (Randomization)
Manders M2
p value M2 (Randomization)
Cross-Correlation 1
Cross-Correlation 2
Workbook
Workbook to file 11
Use first row as header
A
B
C
D
Coloc. Object Analysis
workbook
file format Spreadsheet
output file C:\Users\pike01\Documents\Test.xls
If file exists Overwrite

- ➢ If you haven’t already, save your protocol and try running it. Check it works as expected and produces a spreadsheet. Try it on several of the stacks in the SimulatedData folder.

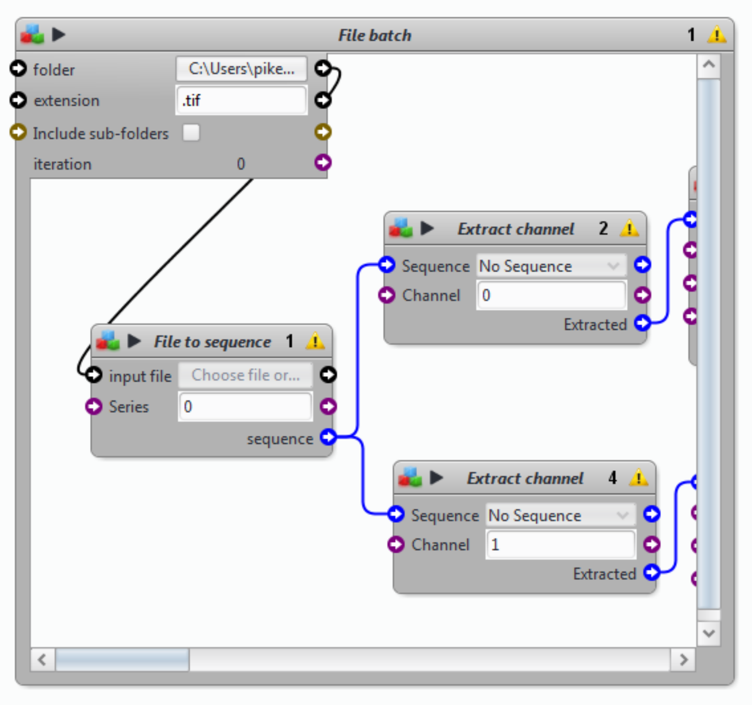
## B. Batch Analysis with Icy Protocols

- The SimulatedData folder contains two sets of two channel image stacks. In this exercise we hypothesise that there is a difference in the level of co-occurrence between the two sets. First, using Icy, open several image stacks from each set. Do you think you can you visually access if there is a difference in the level of co–occurrence between the two sets?

- Clearly we need a quantitative approach, and automated batch processing all the files in the SimulatedData folder will save time and reduce user error and bias. We will therefore modify the protocol from Section A for batch processing across all files in a specified folder. First, navigate to the protocol editor (*Tools > Protocols*) and load the protocol (if it isn’t already open).

- Embed the protocol in a *File Batch* block (*Embedded > File Batch*). This may take a while so please be patient. After embedding the protocol several small modifications need to be made:

  - ➢ Specify the SimulatedData folder containing the test image stacks. You should also specify a .tif extension. All files that are not tifs will now be ignored.
  - ➢ Remove and replace the *Sequence* block with a File to *Sequence Block*. This should be linked to folder output of the File Batch block. The protocol will now load each file in the specified directly.



folder C:\Users\pike...
extension .tif
Include sub-folders □
iteration 0

input file Choose file or...
Series 0

Sequence No Sequence
Channel 0

Sequence No Sequence
Channel 1

  - ➢ Remove and replace the *Sequence* block with a *File to Sequence Block*. This should be linked to folder output of the *File Batch* block. The protocol will now load each file in the specified directly.
  - ➢ Modify the *If file exists* of the *Workbook to file* block to *Merge sheets, excluding first row*. This ensures that the spreadsheet is added to for each file, and not

overwritten. We don't merge the first row as this contains the column names. You should also check and change the output file name and directory.
    - The block number represents the order that they will be run. Make sure all the blocks are performed in a sensible order. If they are not you can click on a block number to re-order it.

- Run the batch protocol on the SimulatedData folder. You should see the iteration number increasing after each file is processed.

- Open the output spreadsheet file using Microsoft Excel or similar. Perform a two way t-test to test the null hypothesis that the two sets have equal mean Manders coefficients (M1 or M2). Can the null hypothesis be rejected such that we can say there is a difference in co-occurrence between the sets?

- Note the Colocalization Studio plugin calculates the Pearson coefficient across the entire analysis ROI. It would be more informative to calculate the Pearson coefficient using only the voxels containing isolated signal from both channels [9]. This would allow us to quantify the correlation in the co-occurring signal.

# Section 4: Tracking with TrackMate

## A. TrackMate Basics and Documentation

- TrackMate is as single particle tracking plugin for Fiji/ImageJ [10]. 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 [3].

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

# 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] 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.

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

[5] 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.

[6] Lagache T, Sauvonnet N, Danglot L, Olivo-Marin J-C. Statistical analysis of molecule colocalization in bioimaging. Cytometry Part A 2015.

[7] Chenouard N. Advances in probabilistic particle tracking for biological imaging. 2011.

[8] De Chaumont F, Dallongeville S, Chenouard N, Hervé N, Pop S, Provoost T, et al. Icy: an open bioimage informatics platform for extended reproducible research. Nature Methods 2012;9:690–6.

[9] Adler J, Parmryd I. Quantifying colocalization by correlation: the Pearson correlation coefficient is superior to the Mander’s overlap coefficient. Cytometry Part A 2010;77:733–42.

[10] 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.