**Introduction**

The pipeline CellP-2D\_v1wNUaut\_032022 is deputed for the automated subcellular identification of 7 different organelles.  
The pipeline is divided in three parts, in order to facilitate the use of it, especially whenever is required edit for different batches. The first part () is appointed at the identification the whole soma (SO) and the cytoplasmic area (CY) based on the nuclear staining (NU). The forementioned objects will be subsequently used in PART-2 and PART-3 of the pipeline.  
  
The second part of the pipeline (), is assigned to identify 6 different organelles (lysosomes, LS; mitochondria, MT; Golgi, GL; peroxisomes, PO; Endoplasmic Reticulum, ER; lipid droplets, LD) previously masked from backgrounds. These objects will be used in PART-3 of the pipeline.  
  
The Third and last part of the pipeline it uses the objects identify in PART-2, to generate the contacts in between organelles. It measures area, size, shape and number of the organelles and contacts. Moreover, it calculates the distribution of the organelles and contacts, using the previously identified soma and nuclei in PART-1.   
All the measurements are exported directly from the PART-3. The excel sheet generated it will contain the following measurements.  
  
Each Part of the pipeline, and Modules, contains specific instructions on the use of it.  
  
**Description settings**

Importing and exporting data

**Name files**

The name of your pictures can be extract from the Metadata Section and it can be used to further group your samples. You can change the name of your samples in the Metadata section but please be careful to change the corresponding Name&Types, and the linked Modules along the Pipeline.  
Name the pictures as followed:   
03202022(*date*)\_SampleName\_BiologicalReplicate\_NumberImage

**PART-1**  
Input Files:  
- single images multichannel in TIFF format or czi.  
Output Files:  
 - Tile of the identified soma, cytoplasm and nuclei.  
 - Images in uint16 format of the identified soma (SO\_objects), cytoplasm (CY\_objects) and nuclei (NU\_objects)

**PART-2**  
Input Files:  
- single images multichannel in TIFF format or czi (That was being used in PART-1).  
- Identified objects from PART-1:  
 SO\_objects: whole soma area  
 CY\_objects: cytoplasmic area  
 NU\_objects: nuclei  
Output Files:  
- Tile of single organelle segmented per Cell ()  
- Tile summary of all the organelle segmented per Cell ()  
- Images in uint16 format of the identified organelle:  
 LS\_objects: lysosomes  
 MT\_objects: mitochondria  
 GL\_objects: Golgi  
 PO\_objects: peroxisomes  
 ER\_objects: Endoplasmic Reticulum   
 LD\_objects: lipid droplet

**PART-3**Input Files:  
- single images multichannel in TIFF format or czi (That was being used in PART-1).  
- Identified objects from PART-1:  
 SO\_objects: whole soma area  
 CY\_objects: cytoplasmic area  
 NU\_objects: nuclei  
- Identified objects from PART-2:  
 LS\_objects: lysosomes  
 MT\_objects: mitochondria  
 GL\_objects: Golgi  
 PO\_objects: peroxisomes  
 ER\_objects: Endoplasmic Reticulum   
 LD\_objects: lipid droplet  
  
Output Files:  
- Tile summary all contacts per Cell (generated from MaskObjects)  
- Tile distribution organelles per Cell  
- Tile distribution contacts per Cell  
- Csv files:

Rescale Intensities

All the images need to rescale the pixel intensities, before being processed.

Whenever the pixel intensity, of a certain channel, it remains steady across different pictures/cells, we suggest rescaling each individual picture, to the full intensity range.

However, when a specific channel presents strong differences in pixel intensities, across different pictures/cells from the same batch, (mainly due to the presence or absence of the targeted organelle), is it instead beneficial to rescale the pixel intensities along all the pictures from the same channel, including an additional image as positive, and/or negative control if required.  
  
Filter Modules  
Depending on the quality of your imaging or the purpose of the segmentation, before proceeding to the identification of the object of interest (soma or organelles), sometimes it can be advantageous to use specific modules for editing the picture, in order to denoise, smooth or enhance specific morpho structures.   
We here use different filter modules specific for each channel, dictated by the properties of each fluorophore/organelle and by the purpose of the segmentation.  
Please feel free to adjust your filter setting according to your purposes.  
  
Threshold Modules  
CellProfiler give the possibility to identify objects manually, by drawing the area of interest or by choosing a pixel intensity (manual thresholding), or with an assisted thresholding, referred as IdentifyPrimaryObjects module which it use an automated thresholding. The module “IdentifyPrimaryObjects“ contains three different sections as shown in Fig.x.The first section, **Size Object,** allows you to specify the size of your target objects (in pixel unit). Depending on the quality of your images, and the purpose of the segmentation you can decide either to discard the identified objects outside your diameter range or not.   
! Choosing a size range, is highly suggested to eliminate possible background and erroneous thresholding. Moreover, the size of your objects is required to assist the declumping of the objects.  
How to choose the size of your objects:  
- Measure the length (in pixel unit, with the measuring tool) of your smallest and largest object and use them as Min and Max in “Typical diameter of objects”.

Graphical user interface, text, application

Description automatically generated

The second section, **Threshold Object** , allows you to decide which thresholding strategy (Global or Adaptive), which method (Miniminum Cross-Entropy, Otsu, Robust-background, Sauvula) and bonds to adopt. Moreover, it gives you the possibility to set the area size whenever you select Adaptive Thresholding.  
We here selected specific thresholding strategy and methods, based on the organelle/fluorophore specificity (see section Threshold strategy), while instead the bonds are calculated empirically.  
We suggest to check the choice of the thresholding strategies as described in the next chapter.  
Change only the following settings, in this specific order:  
1-Size adaptive window.  
2-Lower and upper bonds .  
3-Threshold correction factor.

The third section, **Declumping Objects,** is appointed to declump the identified objects whenever the objects are correctly identified based on the intensity, however the threshold alone is not able to declump the observed object into individual ones.

Change only the following settings, in this specific order:  
 1-Unclick “Never” to have the option of Declumping.  
 2-Change size of adaptive window based on the average of your objects.  
 3-if necessary, change the smoothing size filter.

**!** If you changed the smoothing filters in the Filter modules, you must modify the threshold smoothing scale and the declumping size correspondingly.

**Description threshold strategies**

The thresholding strategy determines the type of input (area of the image) that is used to calculate the threshold. These options allow you to calculate a threshold based on the whole image or based on image sub-regions.

The choices for the threshold strategy are:

**Global:** Calculates a single threshold value based on the unmasked pixels of the input image and use that value to classify pixels above the threshold as foreground and below as background. A higher threshold value will result in only the brightest regions being identified, whereas a lower threshold value will include dim regions.   
This strategy is fast and robust, especially if the background is relatively uniform (for example, after illumination correction).

**Adaptive:** Calculates a different threshold for each pixel, thus adapting to changes in foreground/background intensities across the image. For each pixel, the threshold is calculated based on the pixels within a given neighborhood (or window) surrounding that pixel. The intensity threshold affects the decision of whether each pixel will be considered foreground (region(s) of interest) or background. The source image is broken into 'blocks' equal to the size of the "Adaptive Window" which you can set. A separate threshold can then be calculated for each block and blended to create a gradient of different thresholds for each pixel in the image, determined by local intensity. A higher threshold value will result in only the brightest regions being identified, whereas a lower threshold value will include dim regions.  
This method is slower but can produce better results for non-uniform backgrounds and in detecting small objects (like nuclear speckles).

To view pixel intensities in an open image, use the pixel intensity tool which is available in any open display window. When you move your mouse over the image, the pixel intensities will appear in the bottom bar of the display window.

You can have the threshold automatically calculated from a choice of several methods, or you can enter a number manually between 0 and 1 for the threshold.

The threshold that is used for each image is recorded as a per-image measurement, so if you are surprised by unusual measurements from one of your images, you might check whether the automatically calculated threshold was unusually high or low compared to the other images. See the FlagImage module if you would like to flag an image based on the threshold value.

There are a number of methods for finding thresholds automatically:

**Minimum Cross-Entropy**: The distributions of intensities that define foreground and background are used as estimates for probability distributions that produce the intensities of foreground and background pixels. For each possible threshold the cross-entropy between the foreground and background distributions is calculated and the lowest cross-entropy value is chosen as the final threshold. The lowest cross-entropy can be interpreted as the value where the information shared between the two probability distributions is the highest. On average, given a pixel of an arbitrary intensity, the likelihood it came from the foreground or background would be at its highest.

**Otsu:** This approach calculates the threshold separating the two classes of pixels (foreground and background) by minimizing the variance within each class.

This method is a good initial approach if you do not know much about the image characteristics of all the images in your experiment, especially if the percentage of the image covered by foreground varies substantially from image to image.

Our implementation of Otsu’s method allows for assigning the threshold value based on splitting the image into either two classes (foreground and background) or three classes (foreground, mid-level, and background). See the help below for more details.

NOTE that CellProfiler 2 used a non-standard implementation of two-class Otsu thresholding; CellProfiler 3.0.0 and onward use the standard implementation. While in most cases the calculated threshold is very similar, pipelines that are adapted from CellProfiler 2 and use two-class Otsu thresholding should be checked when converting to CellProfiler 3 and beyond to make sure that method is still the most appropriate.

NOTE that from CellProfiler 4.0.0 and onwards the standard implementation will be used for three-class Otsu thresholding as well. Results with three-class Otsu thresholding are likely to be slight different from older versions, so imported pipelines which use these methods should be checked when converting to the latest version to ensure that settings are still appropriate.

**Robust Background:** This method assumes that the background distribution approximates a Gaussian by trimming the brightest and dimmest X% of pixel intensities, where you choose a suitable percentage. It then calculates the mean and standard deviation of the remaining pixels and calculates the threshold as the mean + N times the standard deviation, where again you choose the number of standard deviations to suit your images.

This thresholding method can be helpful if the majority of the image is background. It can also be helpful if your images vary in overall brightness, but the objects of interest are consistently N times brighter than the background level of the image.

**Only when choosing Global:**

**Measurement:** Use a prior image measurement as the threshold. The measurement should have values between zero and one. This strategy can also be used to apply a pre-calculated threshold imported as per-image metadata.

**Manual**: Enter a single value between zero and one that applies to all images and is thus independent of the input image.

This approach is useful if the input image has a stable or negligible background, or if the input image is the probability map output of a pixel-based classifier (in which case, a value of 0.5 should be chosen). If the input image is already binary (i.e., where the foreground is 1 and the background is 0), a manual value of 0.5 will identify the objects.

**Only when choosing Adaptive:**

**Sauvola:** This method is a modified variant of Niblack's per-pixel thresholding strategy, originally developed for text recognition. A threshold is determined for every individual pixel, based on the mean and standard deviation of the surrounding pixels within a square window. The size of this window is set using the adaptive window parameter.

This thresholding method can be helpful when you want to use a very small adaptive window size, which may be useful when trying to detect puncti or fine details.

To improve speed and efficiency, most of these adaptive thresholding methods divide the image into blocks, calculate a single threshold for each block and interpolate the values between them. In contrast, the simplicity of the Sauvola formula allows our implementation to calculate every individual pixel seperately (no interpolation) without needing excessive computation time.

As regions are likely to contain no cells, adaptive thresholds are constrained to ensure all pixel thresholds are between 0.7x and 1.5x a global threshold, termed the "Guide Threshold". This guide is calculated using the global strategy using the same method as selected for adaptive mode. The one exception to this is Sauvola thresholding, which uses a Minimum Cross-Entropy global threshold as a guide (since Sauvola is only available as a local threshold).

**References** Sezgin M, Sankur B (2004) “Survey over image thresholding techniques and quantitative performance evaluation.” Journal of Electronic Imaging, 13(1), 146-165. (link)

Two classes or three classes?  
To understand how many pixel intensities classes have your pictures, you can check the histogram of the pixel intensities (in Test Mode, Right click) and make sure if you have or not multiple peaks.  
However, for the zoom we used and the approach (having only one cell stained), the pixel intensities per image are very “diluted” so that the background shows the majority of the pixels (is like a Global thresholding). If you want to see the pixel intensities of your objects you can zoom in the histogram or using a cropped image.  
Of course, you can also just simply look by eyes to verify if your objects shows different intensities within the same cell and to use the cursor along your object to determine the pixel intensities.

**Based on the previous description:**

Please Inspect your image: Size, staining (clean staining or not, good unmixing or not), area covered in the picture.

* Is your target object distinguishable clearly from the background? (**Strategy**) Yes (Global) No (Adaptive). How many Backgrounds? two or three.
* How much area of the image is covered by your staining/object? (**Strategy**) Little (Adaptive) or Big (Global)
* How many pixel intensities do have your objects? (**Method**) Can they be classified in Two or Three classes\*, and Is the foreground or the background that change along the images? Foreground can vary along images: Otsu, Two or Three classes. If images vary in overall brightness, but the objects of interest are consistently N times brighter than the background level of the image: Robust background.   
  The exception:  
  How big is your object? (**Method**) Small (very small: Adaptive Sauvula) or big (Any decision of the above).