

FISH-quant v3

Generate FQ outlines from automated cell segmentation

MUELLER Florian¹ muellerf.research@gmail.com

SAMACOITS Aubin¹ aubin.samacoits@gmail.com

WALTER Thomas² thomas.walter@curie.fr

¹Institut Pasteur Paris, Computational Imaging & Modeling Unit, France

² Institut Curie, Paris, France

1. Introduction.....	3
Overview.....	3
Workflow.....	3
Requirements - Software	3
Requirements - Data.....	4
2. Focus-based z-projections to obtain 2D images.....	5
Introduction.....	5
Performing focus-based projections.....	5
3. Segmentation of 2D images with <i>CellProfiler</i>.....	8
Segmentation in CellProfiler.....	8
Test mode.....	9
Analyze Images.....	10
Save pipeline or project.....	10
Generate FQ outlines files from <i>CellProfiler</i> results.....	11
Specify experimental parameters.....	11
Define naming scheme of original images and segmentation results.....	11
[Optional] Generating outline files for a second color.....	12
Specify folder where the results will be saved	12
Specify images that will be analyzed	12
Create outlines.....	12
[SCRIPT] Generating outlines files for many colors	12
4. Segmentation of 2D images with <i>CellCognition</i>	13
Segmentation.....	13
Naming convention of CellCognition: position, well, subwell.....	13
Importing of images.....	14
Cell segmentation.....	15
Tuning the parameters for global and local thresholds.....	16
Saving the results	17
Analysis and organization of results files.....	17
Generate FQ outlines files from CellCognition results.....	18
Specify experimental parameters.....	18

Define channel and segmentation method identifier.....	18
Specify folder where the results will be saved	18
Define folder	18
Create outlines.....	18
5. Inspect outline files.....	19
6. Annex.....	20
Example for naming convention and regular expression.....	20
7. References.....	21

1. Introduction

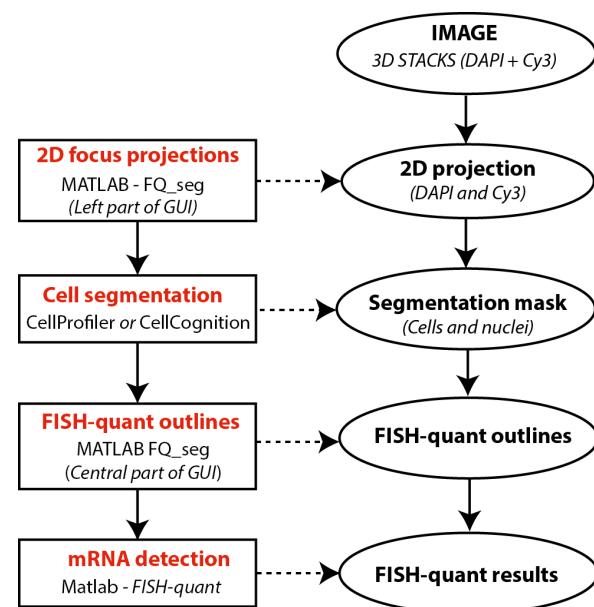
Overview

This document describes the workflow for how to automatically segment nuclei and cells in the context of single molecule FISH (smFISH) and *FISH-quant* (FQ)¹. Image analysis in FISH-quant is performed on 3D images, while segmentation is performed in 2D. We developed a focus-based projection method (available in FQ v3) to obtain these 2D images from the acquired 3D stacks. Cell segmentation is then performed on 2D images with dedicated analysis packages (*CellProfiler* or *CellCognition*, more details below). The obtained segmentation results can then be used in FISH-quant¹ for automated mRNA detection. To segment cells, a nuclear stain (such as DAPI) is used in a first step to automatically segment nuclei. These nuclei are then used as seeds to find the outlines of cells. Cellular segmentation can be performed either directly in the smFISH images – using the non-specific background, or on a dedicated cellular stain (such as HCS CellMaskTM).

Workflow

Workflow (left) requires *CellProfiler* (CP) or *CellCognition* (CG) for cell segmentation, *FISH-quant* (MATLAB) with a dedicated user-interface (FQ_seg) for focus projections and generating outline files that can be read by FISH-quant.

1. Image acquisition in 3D (see section below for more detailed requirements)
2. Focus-based z-projections to obtain 2D images **FQ_seg**
3. Segmentation of 2D images in *CellProfiler* or *CellCognition* **CP or CG**
4. Conversion of segmentation results in FQ outline files. **FQ_seg**
Check quality of segmentation results, e.g. in Fiji or directly in FISH-quant.



Requirements - Software

Cell segmentation – the automatic outlining of nuclei and cells - can be performed with many specialized software packages. Two widely used open-source packages are

- *CellProfiler*² described as a “free open-source software designed to enable biologists without training in computer vision or programming to quantitatively measure phenotypes from thousands of images automatically”.
<http://www.cellprofiler.org/>
- *CellCognition*³ described as a “computational framework dedicated to the automatic analysis of live cell imaging data in the context of High-Content Screening (HCS)”.
<http://www.cellcognition.org>

Either package allows the automated detection (segmentation) of cells and nuclei. These are independent, stand-alone software packages not requiring MATLAB. The most recent versions can be downloaded from their respective websites, where you can also find detailed installation instructions.

Requirements - Data

- Images have to be 3D stacks and saved as separate TIF images for each channel.
- Images have to follow a strict naming convention and contain a unique identifier, e.g. *img_001* where **001** is a running index. Subsequent identifiers specify channel, e.g. **_DAPI** for DAPI images and **_CY3** for FISH. The complete file-name for different FISH image would be *img_001_CY3.tif*, *img_002_CY3.tif*, ... and similar for DAPI *img_001_DAPI.tif*, *img_002_DAPI.tif*,

<u>Required 3D imaging data</u>	→	<u>Output/outlines</u>
• Nuclear stain (e.g. DAPI)	→	Segmentation of nuclei
• smFISH or cellular stain	→	Segmentation of cells

2. Focus-based z-projections to obtain 2D images

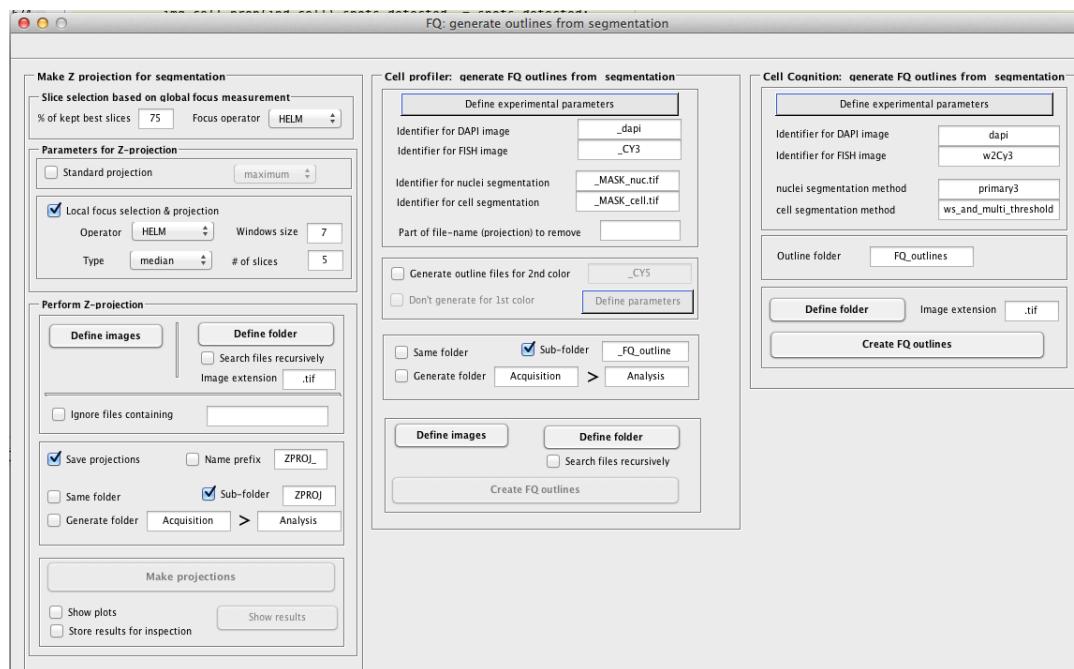
Introduction

CellProfiler and *CellCognition* were developed for segmentation of **2D** images. Frequently, for the segmentation a **maximum intensity projection** of each channel is used. However, we found that better segmentation results can be obtained by performing a focus-based projection. This is achieved by a 2-step process:

1. Calculation of the **total focus** of each z-slice. With this measurement, blurry slices, which have mainly out-of-focus signal, can be removed.
2. Local focus projection. Here the focus is measured **locally** for a user defined window size. Then the z-projection is performed for each pixel over the best-identified slices.

Performing focus-based projections

These projections can be obtained with a MATLAB tool which can be opened by typing **FQ_seg** in the command window (or from the Tools menu in FQ). This will open the user interface shown below. The projection is performed with the panels on the left side with the title '**Make Z projections for segmentation**'.



Slice selection

In the first panel, you can select the focus projection method (See⁴ for details, default 'HELM'). Reducing the number of slices can improve the quality/sharpness of z-projection by removing out-of-focus slices. You can then select how many slices will be kept (default 20). This number depends on the z-spacing you use and also the dimension of the specimen you are imaging. If you select a large number corresponding to more than the size of the stack, then all slices will be used.

Z-projection

You can either perform either a **standard global projection** based on a mean, median, or maximum projection or a projection based on local focus measurements.

For the local focus, you can again select the focus metric. In addition, you can define the window size that will be used to calculate the local focus. You can specify how many slices with the respective best focus

values will be projected. Finally, the same projection methods as for the global approach are available if more than one slice will be used.

We **found that different channels might require different settings for optimal results**. The easiest way to do different type of projection for the two channels is to use the “Ignore file containing” and to give the identifier of other channel. When you want to project the Cy3, write identifier of other channel that might be present in the images folder (DAPI, GFP) separated by **semicolon**, and only the Cy3 will be used to make the projections.

Commonly used parameters for the projections

Cells (smFISH or CellMask):

- **Global projection:** 20 slices selected with the **HELM** operator;
- **Local projection:** **HELM** operator on a 7 pixels window and a median on 5 slices.

Nuclei (DAPI):

- **Global projection:** 20 slices selected with **TENG** operator,
- **Local projection:** **HELM** operator on a 7 pixels window and a median on 5 slices.

Define images / Define folder

In this panel, you can specify the images that will be analyzed. Several features allow a general selection of images

- You can define manually images in a given folder with the button ‘*Define images*’.
- Alternatively, you can specify a folder in which all images will be analyzed. In this case, you have to define additionally the image extension, e.g. *.tif*. Note, that the image extension is case-sensitive.
- You can also define folders recursively, i.e. images will be searched in the defined folder and all subfolders.
- You can also specify that certain files will not be processed. Here you have to specify parts of the file-name that uniquely specifies them, e.g. ‘CY3’.

Save projected images

The generated images can then be saved. By default, they are saved under the same file-name as the original image in a **subfolder** of location of the original image called ‘ZPROJ’.

There are different options to change this behavior.

- You can add a pre-fix to the file-name. Important: if you add a prefix, you have to specify during step where the FQ outline files are created, that this prefix has to be removed. Otherwise, the outline files will point to an image that doesn’t exist (more details below).
- You can save the files in the same folder as the original files.
- You can save the files in a different folder. This option allows you to replace parts of the folder-name by another string. For instance, files are saved in */Users/FM/img/raw*. You can now say that ‘*raw*’ should be replaced by ‘*ZPROJ*’. The images will then be saved in the folder */Users/FM/img/ZPROJ*

Note 1: If a file with the same name is present, the image will not be saved. You have to manually delete this image first.

Note 2: it is possible to combine saving in a subfolder and replace parts of a name of the folder.

In addition to the images, the program will also save a small text file with the settings used to create the z-projections.

Show plots

If you enable this option, you will obtain a plot for each image that summarizes the results for each analyzed image. We recommend doing this only for a small number of images, to avoid having too many open images.

Show results

With this option enable, the program will save the standard maximum intensity projection and the advanced projection for each image. When the processing is done, you can then inspect these in a separate user interface by opening 'Show results'. Enable this option when processing only for a few images, otherwise the resulting data might be so large that MATLAB crashes.

Perform z-projection

Pressing the button 'Make Projections' will then apply the Z-projection to all specified images. Note that images with fewer than 3 Z slices will not be skipped in the analysis.

With good quality z-projections made via FQ_seg, we can now use CellProfiler or CellCognition for the **actual segmentation** of nuclei and cells. These steps will only require the obtained the 2D projections.

3. Segmentation of 2D images with *CellProfiler*

CellProfiler uses the 2D projections of the nuclear stain (DAPI) and smFISH images to automatically detected nuclei and cells, respectivtely.

Segmentation in CellProfiler

Below we list a typical workflow in *CellProfiler* for the segmentation of nuclei and cells. The results are saved in masks that can be used to generate outline files for FISH-quant. Such workflows are saved in so-called pipelines, which are provided with this toolbox (files ending with *.cppipe*).

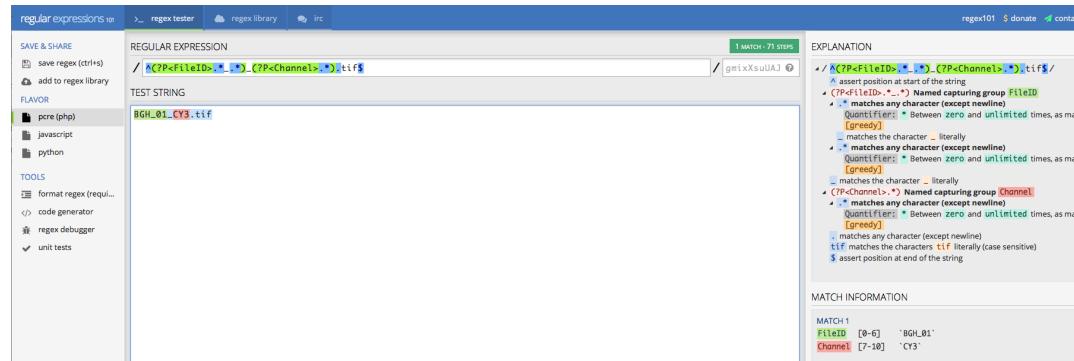
The described workflow is for nuclear AND cellular segmentation. It therefore requires the 2D projections of DAPI for the nuclear segmentation and projection for cellular segmentation (either from the actual smFISH image or a dedicated cell marker).

1. Open *CellProfiler*
2. Import analysis pipeline from Menu file > Import Pipeline > from file. Such a pipeline is provided in FISH-quant in the folder Segmentation > *CellProfiler*.
3. Then you have to **specify the images** that you would like to analyze. These are the 2D projection obtained above. You can drag and drop them into the big window in the **Images** module (part of the Input modules).
4. The next step is very important. Here the **channels** belonging to the same image (e.g. DAPI for the nuclei and smFISH for the cellular outlines) are assigned to each other. This is done with two other modules. First, the **Metadata** module where the channel and position information is extracted. Second, the **NamesAndTypes** module where images obtain names for the further analysis steps.
 - The extraction method for the meta-data depends on how your images are named and organized. We usually use a method that extracts the information from file-names. To guarantee largest flexibility so-called regular expression are used. This is a relatively simple syntax to automatically extract different parts of a text. For more details, you can consult any website about regular expression such as <http://www.regular-expressions.info/quickstart.html>

For the example data, the image name follows the convention *gene_pos_channel.tif*, where *gene* is the analyzed gene, *pos* the position of the image (i.e. a value that will be increased for each acquired multi-color stack), and *channel* the color channel (e.g. Cy3 or Dapi). To extract the relevant information, we use the following regular expression

`^(?P<FileID>.*_*)(?P<Channel>.*).tif$`

This might look a little bit confusing at the beginning. A good website to test your regular expression is <https://regex101.com/>. It not only allows you to test your expressions, but provides information about the different terms. See also Annex for an additional example of a different naming convention. When we test this expression on an example file-name (*BGH_01_CY3.tif*), we see the following



With this regular expression, we simple extract the first part of the string and a name it *FileID* (e.g. BGH_01) and the second par of the string and name it *Channel* (e.g. CY3). For the corresponding DAPI image BGH_01_DAPI.tif, we obtain the same *FileID* but a different channel (DAPI). So with these two extracted strings, we have unique identifiers to properly assign for each image the different channels. If your image names don't follow this convention, you have to adapt the regular expression. See *CellProfiler* for more help.

To apply the settings to extract the Metadata press the button '**Update metadata**'. This might then take a little while. If everything works fine, a table in the lower part of the panel will populate containing information about the channel and position. If that's the case you can continue, if not please contact the *CellProfiler* manual for help.

- To now assign the different channels to a position, go to the **NamesAndType** module. Here you tell *CellProfiler* how to refer to the different images (e.g. DAPI vs CY3). This will be important for the subsequent analysis steps, where for instance the nuclear segmentation is done based on the DAPI image. In the example data, we simply use the channel information in the file-name to decide if a image is DAPI or CY. Simply click update. This will populate a table in the lower part, where the different images of the different channels are correctly assigned to each other and names are attributed. For other images, you will likely have to update these naming rules. Please consult the *CellProfiler* help if this does not work.
5. You can then either simply apply the segmentation settings to all images (Button '*Analyze Images*') or start a test mode (Button '*Start test mode*').
 6. We recommend saving the *CellProfiler* project in the folder with the z-projections. This allows later inspecting the settings used for the segmentation.

Test mode

This allows you to change settings and look at their impact at the individual steps of the pipeline. Next to each option in the different processing modules, you can find a button with a questions mark, which opens the *CellProfiler* help. Below typical settings used for nuclear and cellular segmentation are shown. *CellProfiler* provides an excellent help function that explains the different parameters.

<h3>Nuclei</h3> <p>Select the input image: DAPI (from NamesAndTypes)</p> <p>Name the primary objects to be identified: Nuclei</p> <p>Typical diameter of objects, in pixel units (Min n,Max): 50 - 200</p> <p>Discard objects outside the diameter range? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Discard objects touching the border of the image? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Threshold strategy: Global</p> <p>Thresholding method: Otsu</p> <p>Two-class or three-class thresholding? Three classes</p> <p>Minimize the weighted variance or the entropy? Weighted variance</p> <p>Assign pixels in the middle intensity class to the foreground or the background? Background</p> <p>Select the smoothing method for thresholding: Manual</p> <p>Threshold smoothing scale: 4</p> <p>Threshold correction factor: 1.1</p> <p>Lower and upper bounds on threshold: 0 - 1</p> <p>Method to distinguish clumped objects: Shape</p> <p>Method to draw dividing lines between clumped objects: Intensity</p> <p>Automatically calculate size of smoothing filter for declumping? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Automatically calculate minimum allowed distance between local maxima? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Speed up by using lower-resolution image to find local maxima? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Retain outlines of the identified objects? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Name the outline image: NucOutlines</p> <p>Fill holes in identified objects? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Handling of objects if excessive number of objects identified: Continue</p>	<h3>Cells</h3> <p>Select the input image: CY3 (from NamesAndTypes)</p> <p>Select the input objects: Nuclei (from IdentifyPrimaryObjects #05)</p> <p>Name the objects to be identified: Cells</p> <p>Select the method to identify the secondary objects: Watershed - Image</p> <p>Threshold strategy: Global</p> <p>Thresholding method: Otsu</p> <p>Two-class or three-class thresholding? Three classes</p> <p>Minimize the weighted variance or the entropy? Weighted variance</p> <p>Assign pixels in the middle intensity class to the foreground or the background? Foreground</p> <p>Select the smoothing method for thresholding: Manual</p> <p>Threshold smoothing scale: 5</p> <p>Threshold correction factor: 1.1</p> <p>Lower and upper bounds on threshold: 0 - 1</p> <p>Discard holes in identified objects? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Discard secondary objects touching the border of the image? <input type="radio"/> Yes <input checked="" type="radio"/> No</p> <p>Retain outlines of the identified secondary objects? <input checked="" type="radio"/> Yes <input type="radio"/> No</p> <p>Name the outline image: CellOutlines</p>
---	---

Analyze Images

This will perform the automated segmentation of all images and save the results automatically.

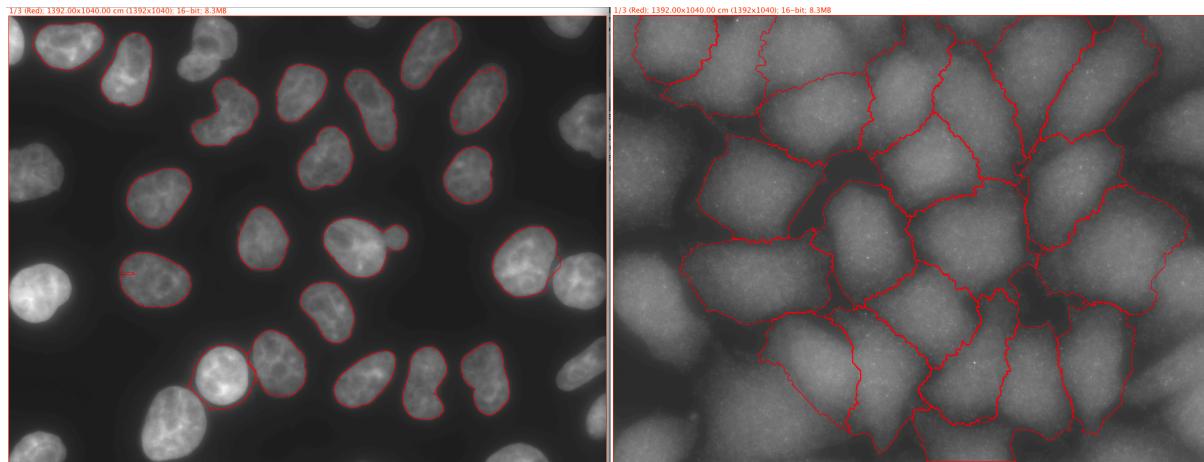
FOR FASTER PROCESSING DISABLE THE DISPLAY OF THE SEGMENTATION.

This can be done by clicking on the icon with the eye next to the different processing steps.

The **segmentation results** are saved in the folder of the 2D projections

- Masks for cells and nuclei. These are images with the same name as the MIP, followed by '_MASK_cell' or '_MASK_nuc'. In these images each identified cell or nucleus has a different gray-scale value. They will then serve as an input to automatically generate FQ outline files.
- Images that allow to check how well the segmentation worked, they also have the same file-name as the MIP and end by '_CONTOUR_cell' or '_CONTOUR_nuc'

Below are two examples for segmentation nuclei (left) and cells (right).



Save pipeline or project

You can then also save the analysis pipeline of *CellProfiler*. This allows you to open this pipeline later to either re-analyze the data again or apply the pipeline to another data set.

Generate FQ outlines files from *CellProfiler* results

The segmentation masks generated by *CellProfiler* can be used to automatically create FQ outline files with the middle part of the user interface.

Specify experimental parameters

First, you have to define the '**Experimental parameters**'. Use the same parameters as for the main FISH-quant interface when pressing the 'Modify' button in the panel 'Experimental parameters'. Simply adjust them to your data by changing the corresponding values.

Note 1: These parameters have to be defined – even if the default parameters are good. Only then the button to generate the outlines will be enabled.

Note 2: You can change the default parameters in a text-file called 'FISH-QUANT_default_par.txt' which is located in the same directory as the *FQ_seg* MATLAB file.

Note 3: If you added a prefix to the file-name of the 2D projections, then you have to specify in the dedicated box that you want to remove this prefix when generating the outline files. This is important, since the outline files point to the original image files (whose file-name does not contain this prefix). For instance, if you add *ZPROJ_*' to any z-projected file, you need to exactly remove this prefix again.

Define naming scheme of original images and segmentation results

Second, you have to define a few parameters regarding the **naming convention of your files**. You have to define the unique identifier for the FISH and DAPI images. These identifiers have to be defined in a way that when you take the full file-name of the FISH image and you replace the identifier for FISH, e.g. CY3, by the identifier of the DAPI, e.g. 'DAPI', you get the DAPI file-name. Then, you have to define the identifier of your segmentation results using the suffix that *CellProfiler* added in the name of your masks (MASK_nuc and MASK_cell for example). See also example below.

You also have to specify the **file-extension of the original (3D) images**. By default, the extension *.tif* is used. If your files have another extension (*.tiff*, *.stk*, ...), you have to accordingly change the text. This is important to guarantee that these files are correctly referenced in the FISH-quant outline files.

You also have to specify the **file-extension of the masks**. By default, the extension *.tif* is used. If your files have another extension (*.tiff*), you have to accordingly change the text. This is important to guarantee that these files are found.

Example

Original images and 2D focus-projections (saved in a different folder)

BGH_01_CY3.tif	→ smFISH image
BGH_01_DAPI.tif	→ DAPI image

Segmentation results

BGH_01_CY3_CONTOUR_cell.tif	→ Segmentation of cells (contours)
BGH_01_CY3_MASK_cell.tif	→ Segmentation of cells (segmentation mask)
BGH_01_DAPI_CONTOUR_nuc.tif	→ Segmentation of cells (contours)
BGH_01_DAPI_MASK_nuc.tif	→ Segmentation of cells (segmentation mask)

The channel identifier are highlighted in **blue**, and the identifier for the segmentation in **red**.

[Optional] Generating outline files for a second color

This option allows to generate outline files for a second color, e.g. for a dual-color FISH experiment. The outlines for this color will be based on the segmentation results of the first color and the exact same cells will be used. This allows a simple comparison between the detection results.

As above, the identifier for the second color has to be specified, e.g. 'CY5'. You also have to redefine the experimental parameters (most often to adjust the excitation and emission wavelength).

As an **option** (*Don't generate for 1st color*), you can only generate the outlines for the second color but not for the first color. This option is useful if the first color does not contain actual smFISH data but results of a dedicated cell segmentation stain.

Specify folder where the results will be saved

The same options as for the projections are available. More details can be found there.

Specify images that will be analyzed

You can either choose different images that you want to analyze (Define images), or select an entire folder (Select folder). For the latter, you can also specify a recursive search; this means that all subfolders will be searched as well. The script **will only consider images** that follow the above explained naming convention – other images will be ignored.

Create outlines

Lastly, press the button 'Create FQ outlines'. The script will then automatically search for the files describing the segmentation of cells and nuclei. For each image an outline file with the reference to the ORIGINAL 3D image will be generated and nuclei assigned to their respective cells.

[SCRIPT] Generating outlines files for many colors

If you want to generate outlines for even more colors, then you can use the script **SCR_outlines_create_v1**. This script requires a file specifying the different channels, and outline files from a first channel, which will be used to create the other outline files.

You have to first specify in a **tab**-delimited text file all the colors that you would like to generate. This file has a very strict structure. The first row contains the unique string of the first color, e.g. CY3. The following rows specify one color each. First element is the unique string, e.g. CY5; 2nd element is the excitation wavelength in nm, 3rd element the emission wavelength.

In the script you first define all the outline files that you would like to process, then you can load the text file defining the different colors. The script will then generate the outlines for the different colors, and save them in a separate folder indicating the unique string of this color. The script **requires only the outline files** (and not the underlying segmentation results).

4. Segmentation of 2D images with *CellCognition*

Segmentation

CellCognition recursively scans user provided folders for images, performs the segmentation, and saves the segmentation results as masks. Such workflows are saved as **settings files**, which are provided with this toolbox (files ending with `.conf`).

You can load the settings file from within *CellCognition* using *File>Open Settings*. Once the settings file is loaded, you just have to set the input and output folder and the segmentation is ready to start.

Four different settings files are provided. We recommend the first two ones for segmentation of smFISH images.

- “`cell_mask_setting_seg`” which can be used to segment cells based on a CellMask stain. It combines a combination of local and global thresholds (see section ‘Tuning parameters’ below for more details).
- “`smFISH_setting_seg`” which uses the same segmentation strategy as above for CellMask but with parameters better suited to segment the non-specific background signal of an smFISH image as cellular marker.
- “`Global method settings`” which combines a global threshold with a watershed segmentation.
- “`3-Otsu method settings`” where a Otsu thresholding with 3 classes is performed. The middle class can be assigned to the background or the foreground. We assign the middle class to the foreground, i.e. the cells. The thresholded image is again analyzed with a watershed segmentation.

Naming convention of *CellCognition*: position, well, subwell

CellCognition was developed specifically for high-content screen. This affects the data organization and a few terms are important to understand and how they translate to a simple imaging experiment. In high-content screens, images are often acquired on multi-well **plates**. For a given plate, a **well** corresponds to a specific experiment (a specific treatment, a specific gene, etc). For each well, images at different **positions** can be acquired. These positions are referred to as sub-wells. The position of an image is a unique value for each plate and is defined by combination of the well and the subwell. In a given plate, the position will be for example « 001_04 », where 001 is the well and 04 is the subwell.

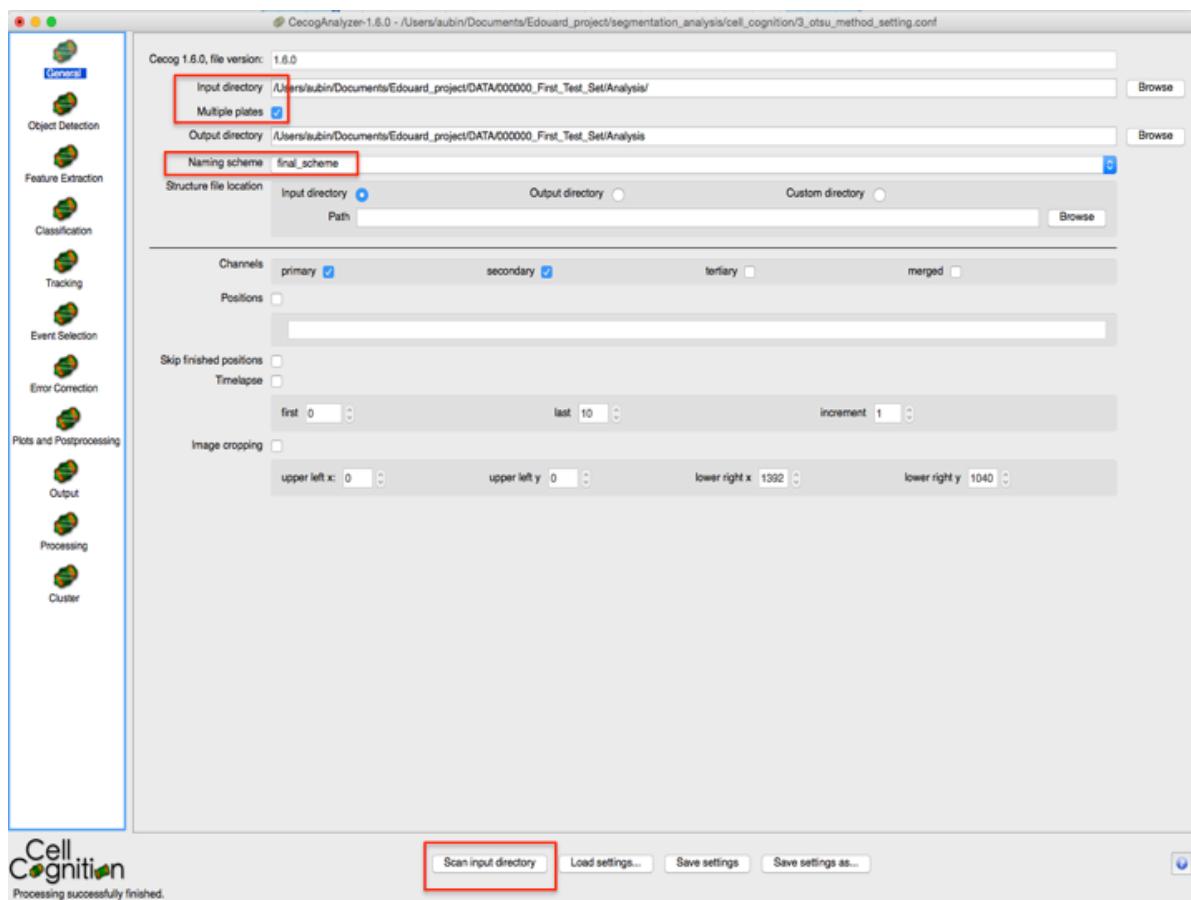
Important: Plates are separate folders (as explained below in more detail). Well and subwell are defined by the file-name (explained below). However, the well data can be (but does not have to be) in separate folders.

Example. smFISH experiments are often not performed in a multi-well format but this notion can be adapted and allows for a flexible organization. We perform FISH experiments against five different genes on two different days. We can consider each day as a “plate”. And each of the FISH experiments as a “well”. We have therefore two folders corresponding to the different dates. Each can contain separate folders for each gene with the different acquired positions (corresponding to sub-wells). What is important is how the smFISH images are named – they have to contain a part that can be interpreted as a well (the same for a given gene) and a part that is the sub-well (the actual different acquired positions).

Importing of images

You have to specify an input folder, which is scanned by *CellCognition* at the beginning of the analysis to detect all images. It performs a recursive search of the folder, looking into every subfolder of the input folder.

An important option is ‘multiple plates’ (see below). When enabled this indicates that the input folder contains subfolders corresponding to different plates. If the option is enabled, the organization of the results will be in a per-plate basis (more details below).



The import of images is based on a regular expression that allows extracting basic information about the images from their file names (well, time, position etc....). These regular expressions are stored in the “naming_schemas.ini” file, which can be found with *CellCognition* code package (On a Mac, right-click on the *CellCognition* icon and chose “Show package contents”) at “content\Resources\resource\naming_schemas.ini”. We provide an updated file, with a naming scheme called *FISH_HCS* adapted to the file-names with the naming convention:

well_cell_gene_pos_channel.tif.

For instance, *CellCognition* will interpret w3_HeLa_DYNC1H1_P4_DAPI.tif as

Plate : Determined by the folder with the images

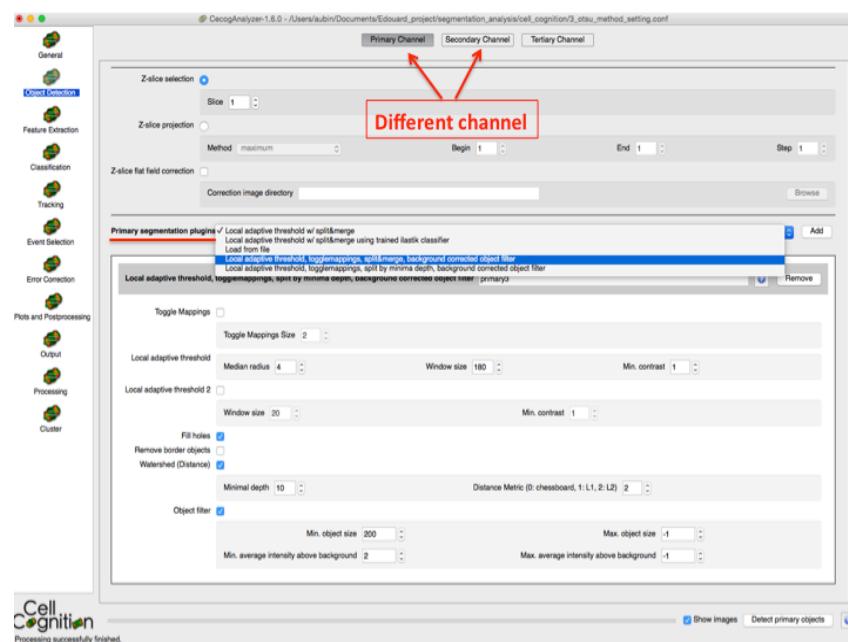
Well : 3
Subwell : 4
Channel : DAPI

→ Position (as defined by *CellCognition*): 3_4

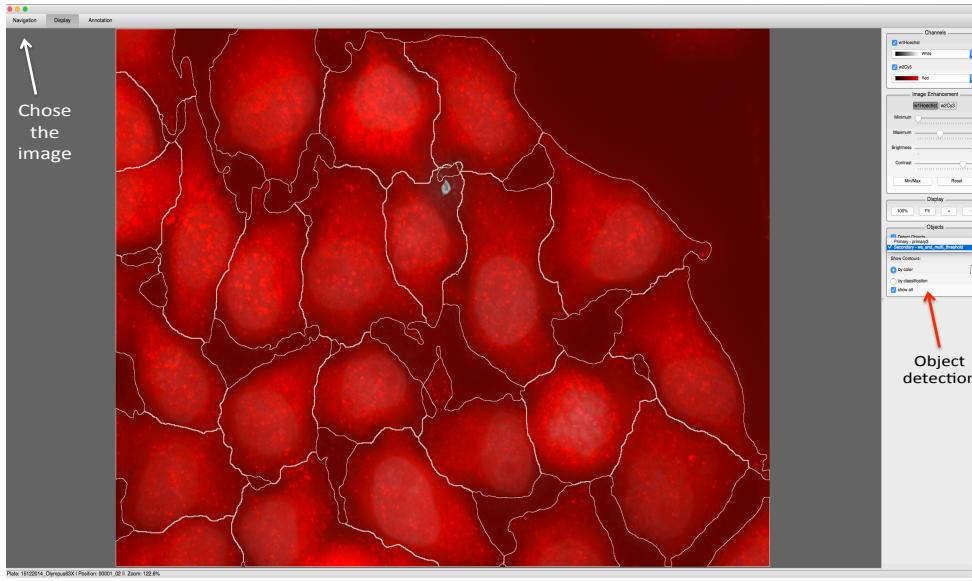
Once the input folder is defined and the correct naming schema is selected, you can click on “**scan input directory**” and *CellCognition* displays how many plates/images have been found.

Cell segmentation

The segmentation of cells is a two steps process. First, nuclei will be segmented with a global thresholding strategy. Second, each nucleus will serve as a ‘seed’ for the segmentation of the cells. These two detection steps have to be defined by the user. Once in the “*object detection*” tab, it is possible to navigate between the different channels, and, for each one, to select a primary/secondary plugin in the list box and add it. The set of parameters are now visible and can be adjusted.



The easiest way to adjust the parameter is to use the browser (**CTRL + B**). It is possible with the browser to select an image in the list, to adjust the contrast of the several channel, and to perform and visualized the segmentation with the settings specified in the main interface of *CellCognition*.



Tuning the parameters for global and local thresholds

Global threshold

Gauss filter size: a gaussian filter is applied as prefiltering step. Here, we can choose the bandwidth of this filter. Large values correspond to important smoothing. With increasing parameter, the contour detection will be more robust and more regular, but will also follow the contours less exactly.

Otsu method is a way to determine a threshold in order to separate classes of pixels according to their grey level. In most cases, there are only two classes (object pixels and background pixels), but in principle it is also possible to calculate several thresholds resulting in more classes. As the objective is segmentation, we would –in this case- still need a rule how to combine several classes into 2 classes.

The threshold is calculated in an automatic way, but it is possible to adjust it. If the threshold calculated by the algorithm is t , we will finally apply the threshold $a*t + b$.

Otsu factor (a): the threshold given by the Otsu method is multiplied with this factor. For factors smaller than 1, this results in a lower threshold (increased sensitivity, but also potentially leading to false positives) and consequently in larger objects.

Threshold offset (b): the product of Otsu factor and automatically determined threshold is added to a user-defined offset. With b increasing, the threshold applied will become more selective (smaller objects, reduced sensitivity, fewer false positives).

Local threshold

Median radius: a median filter is applied as prefiltering step. We can choose the size of this filter. Large values correspond to important smoothing. With increasing parameter, the contour detection will be more robust and more regular, but will also be less exact w.r.t the true contours.

Window size: Parameter of the local thresholding. The local threshold calculates first a background approximation by calculating the average value inside a sliding window. This value is subtracted from each pixel value. If this value is very large, this value becomes less local and adapts less well to the local

features of the image. But if a too small value is chosen, the interior of a cell might not be detected. Normally, one would fix this to be a bit larger than the diameter of the objects to be segmented.

Local Threshold: is actually a global threshold applied to the residue image. The “local character” of this threshold comes only from the fact that the background signal has been subtracted. This is an absolute value, in most cases relatively low.

Results from local and global thresholds are added (union of the two sets is taken). A watershed algorithm is calculated to separate connected cells. The seeds of the algorithm are the nuclei from the DAPI channel

Saving the results

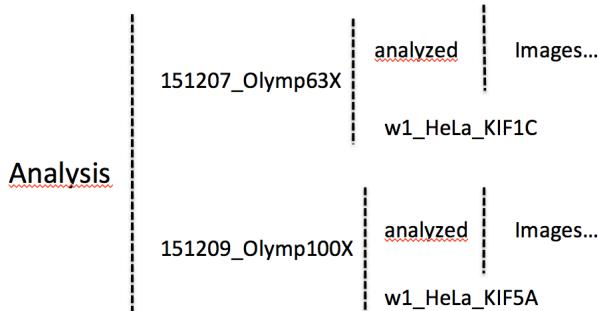
In the **output panel**, we can specify what segmentation result we want to export and save

- “**Contour images**” are the original images with the outline of cells segmented; it allows to visualize the segmentation results compared to the real images.
- “**Label images**” are simple images, with pixel value of 0 in the background, and a different value for each cells; these are the results we use to create FQ outlines.

Analysis and organization of results files

To launch the analysis on the whole image data set, use the tab “processing”. Press the “start processing” button to start the analysis.

The results will be saved in the output folder defined at the beginning of the analysis. *CellCognition* automatically creates four folders called ‘*hdf5*’, ‘*log*’, ‘*plots*’ and ‘*analyzed*’. If the “*multiple plates*” options is not checked, the set of four folders is created directly in the output folder defined at the beginning of the analysis. Otherwise, the four folders will be created in each subfolder corresponding to a plate.



The folders 151207_Olymp63X corresponds to a plate (an acquisition performed on a given day on a certain microscope), containing different subfolders such as w1_HeLa_KIF1C which corresponding to wells.

The segmentation results can be found in:

`analyzed/images/labels/channel/segmentation_method/`

The label images are named based on the information extracted with the regular expression:

P01_04_T00.tif

Where *P* is the position as explained above and *T* the time (always 1 in our case since we don not have time-resolved data).

Generate FQ outlines files from CellCognition results

The segmentation masks generated by *CellProfiler* can be used to automatically create FQ outline files with the middle part of the user interface.

Specify experimental parameters

First, you have to define the '**Experimental parameters**'. These are the same parameters that have to be defined for the images in the main FISH-quant interface when pressing the 'Modify' button in the panel 'Experimental parameters'. Simply adjust them to your data by changing the corresponding values.

Note 1: You have to ALWAYS define the parameters – even if the default parameters are good. Only then the button to generate the outlines will be enabled.

Note 2: you can change the default parameters in a text-file called 'FISH-QUANT_default_par.txt' which is located in the same directory as the *FQ_seg* MATLAB file.

Define channel and segmentation method identifier

Second, you have to define the names of the two channels that have been used for segmentation and the segmentation method for the first and for the second channel. Then, you need to define the name of the channel of interest (the one on which the spot detection will be performed. It can be the same than the segmentation channel). When saving the results, Cell Cognition creates a folder per channel, and a subfolder per segmentation method used to segment the channel. You have to indicate to the *FQ_Seg* which channel result you want to use, and from which segmentation method. The names have to correspond to these folders names.

Specify folder where the results will be saved

You only have the possibility to save the outlines file in a subfolder of the well folder of the projection. You can specify the name of the subfolder from the "Outline folder" of the GUI. Your outline will always be saved in "analysis→w1_HeLa_KIF1C→"Outline_folder".

Define folder

You have to define one or multiple folder that **corresponds to a plate**.

Create outlines

Lastly, press the button '**Create FQ outlines**'. The script will then automatically search for the files describing the segmentation of cells and nuclei. For each image an outline file with the reference to the original 3D image will be generated and nuclei assigned to their respective cells.

5. Inspect outline files

The next few steps summarize a simple workflow for a manual inspection of the outlined cells

1. Open FQ and define the folders with images and the new outlines.
2. Open '*List directory content*' from the tools menu. Here navigate to the folder with the outlines. Double-click on an outline will display it on the outline designer.
3. Click on the list with the different cells. You can use the arrows to navigate through this list (see shortcuts below).
4. Delete each cell that's not good with CTRL-X
5. Save outline with CTRL-S

The following short cuts help to speed up the analysis

CTRL-S	Save outline
CTRL-X	Delete currently selected cell <i>Note: works only when you clicked on the list with the cells before</i>
Arrow UP	Previous cell <i>Note: works only when you clicked on the list with the cells before</i>
Arrow DOWN	Next cell <i>Note: works only when you clicked on the list with the cells before</i>

6. Annex

Example for naming convention and regular expression

Below we provide another example for naming images and the corresponding regular expressions.

File-names:

ZPROJ_O03_mock_ed3_1_w4sdc405_s10_cor.tif

ZPROJ_O03_PUUV_ed3_1_w1sdc640_s1_cor.tif

Corresponding regular expression

.*_w(?P<Channel>).*_s(?P<FileID>.*)_

or even more complex to extract more information:

.*_(_(?P<Virus>.*)_ed3.*_w(?P<Channel>).*_s(?P<FileID>.*)_

7. References

1. Mueller, F. *et al.* FISH-quant: automatic counting of transcripts in 3D FISH images. *Nat. Methods* **10**, 277–278 (2013).
2. Kamentsky, L. *et al.* Improved structure, function and compatibility for CellProfiler: modular high-throughput image analysis software. *Bioinformatics* **27**, 1179–1180 (2011).
3. Held, M. *et al.* CellCognition: time-resolved phenotype annotation in high-throughput live cell imaging. *Nat. Methods* **7**, 747–754 (2010).
4. Pertuz, S., Puig, D. & Garcia, M. A. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition* **46**, 1415–1432 (2013).