SCIP

Single Cell Image Process Toolbox
V1.0 Manual

Authors: Leonardo Martins, Ramakanth Neeli-Venkata, Samuel M.D. Oliveira, Antti Häkkinen, Andre S. Ribeiro and José M. Fonseca
## Contents

1. Input Images .......................................................................................................................... 3  
2. How to run the SCIP program ............................................................................................ 4  
   2.1. Segmentation and Images Alignment ............................................................................. 5  
      2.1.1. Automatic Segmentation ....................................................................................... 5  
      2.1.2. Manual Adjustment of the Segmentation .............................................................. 7  
      2.1.3. Image Alignment ..................................................................................................... 11  
   2.2. Loading Segmented Images ............................................................................................ 13  
   2.3. Loading Handles ............................................................................................................. 15  
   2.4. Structure Detection ........................................................................................................ 15  
      2.4.1. Gaussian Segmentation .......................................................................................... 16  
      2.4.2. TreshMorph’ Segmentation ................................................................................. 17  
      2.4.3. Spot Detection Algorithms .................................................................................... 20  
      2.4.4. Nucleoid Detection ................................................................................................. 21  
      2.4.5. FtsZ Ring Detection ................................................................................................. 22  
      2.4.6. Min System Detection ............................................................................................ 23  
      2.4.7. Inclusion Bodies Detection ..................................................................................... 24  
      2.4.8. Undefined Structures Detection ........................................................................... 25  
      2.4.9. Multiple Structures Detection ............................................................................... 25  
   2.5. Visualization Panel and Frame Visualization ................................................................. 26  
      2.5.1. Seeds Manual Correction ....................................................................................... 30  
   2.6. Exporting Results ............................................................................................................ 31  
   3. References .......................................................................................................................... 35

This Manual informs on the usage of SCIP, via its Graphic User Interface (Figure 1). The Supplementary Material provides additional information on the Algorithms implemented in SCIP.

1. Input Images

Loaded images are required to be in formats supported by MATLAB (e.g. TIFF). Microscopy images should be in grayscale or RGB. Segmented images (masks) should be in RGB (with a different colour for each cell).

Images from a given microscopy modality need to be of the same size, while the size of images of different modalities can differ.

If the user loads a set of images (e.g. from a time series), the filenames must follow the format: ‘Setname’t001.tif, ‘Setname’t002.tif, etc.
It is possible to have multiple ‘functional images’ associated to one ‘morphological image’ (e.g. 5 multiple functional images taken every minute, for 1 morphological image taken every five minutes).

Given multiple allocations to one morphological image (Figure 2), SCIP compares the minimums between the absolute difference of the Morphological Indices and the Functional Indices. In case of a tie, SCIP selects the first allocated image.

![Figure 2 – Allocation of Morphological Images. (A) 1-to-1 allocation. (B) 1-to-2 allocation. (C) 1-to-5 allocation.](image)

The ‘Use Timestamps’ option requires a metadata file ‘meta.txt’ in the same folder as the images, consisting of lines with the following format (UTC timestamps have a 1 second precision):

`<filename><tab>modified<tab><YYYY>-<mm>-<dd>T<HH>:<MM>:<SS>Z`, where:

- `<filename>`: base name of the file (for example 1.tif)
- `<tab>`: a tabulation character (ASCII character 9, also known as HT or ^I)
- `<YYYY>`, `<mm>`, `<dd>`: year, month, day
- `<HH>`, `<MM>`, `<SS>`: hour, minute, second

2. How to run the SCIP program

The user can start SCIP by running the Executable file, which will launch the initial GUI (Figure 1). The toolbox workflow is divided in three major steps (see Figure 3), which can be selected by clicking on the options to load images for segmentation, segmented images, or handles (Figure 4A). Note that all steps in this workflow are automatic, but can be manually adjusted.
The user can also create a video from the images or export the handles file in a .mat format (with the original options and analysis results) using the “Export Handles option” (Figure 4B).

2.1. Segmentation and Images Alignment

2.1.1. Automatic Segmentation

If the user selected the first option in Figure 4 (‘Load Images for Segmentation’), it will activate the Cell Segmentation Interface options (Figure 5), where two methods for automatic segmentation can be selected. Afterwards, the automatic segmentation can be performed by pressing the Button ‘Automatic Segmentation’, similarly to the Software ‘CellAging’ (Häkkinen et al., 2013) and ‘iCellFusion’ (Santinha et al., 2015).
Figure 5 – Activation of the Cell Segmentation Interface options: (A) ‘GPL+CART’ (B) ‘Otsu + Median’.

The first option in the dropdown box of the GUI - ‘GPL+CART’ - uses the Gradient Path Labelling Algorithm (Mora et al., 2011) to create the segmentation seeds. It creates over-segmented sections, and then uses the Classification and Regression Trees Algorithm (Breiman et al., 1984) to merge and discard inappropriate sections (Queimadelas et al., 2012). The Inter-Frame Correction (Checkbox in the GUI interface) further enhances the segmentation’s quality using information from subsequent frames to merge or discard sections, ending the segmentation with morphological operations similar to the methods in iCellFusion (Santinha et al., 2015).

The second option (‘Otsu + Median’ in the dropdown box of the GUI) is based on Multilevel Otsu’s thresholding (Otsu, 1979), with a median filter. Clusters are separated by the Watershed and Distance transform. A Convex Hull of the resulting cluster is generated and subtracted from the cluster. The resulting objects are connected by the smallest line possible, followed by the morphological operations of thickening, opening and flood-filling.

Following this, a progress bar appears, which disappears when the segmentation is complete. After completing the automatic segmentation, the options ‘Manual Corrections of Segmentation Images’ (Section 2.1.2) and ‘Alignment’ (Section 2.1.3) are enabled.

The workflow for the automatic and manual segmentation is presented in Figure 6.
2.1.2. Manual Adjustment of the Segmentation

Manual corrections can be done prior to the alignment with the Functional images. Pressing the button “Manual Corrections of Segmentation images” (Figure 7) opens a new window (Figure 8).

Figure 6 – Segmentation workflow of the two cell segmentation algorithms, respectively Paths 1 and 2.

Figure 7 – Button for activation of the Manual Corrections tool.

The window (Figure 8) can be maximized in the Screen. It has zoom options for better visualization of each cell. A ‘Help Menu’ can be accessed by pressing ‘F1’ (Figure 9).
Figure 8 – Manual Adjustment Window. Blue outlines result from the automatic segmentation, while red outlines are manual adjustments.
The primary mouse button can be used to select cells (which become highlighted in green). Pressing again unselects a selected cell. The secondary mouse button can be used to toggle zoom on and off. Zooming can also be done by hitting ‘z’, which zooms to the mouse pointer position. The zoom ratio can be progressively changed by pressing 'shift+m' (increase) or 'shift+l' (decrease).
When segmenting a cell, the user can add a new segment using the ‘impoly’ function, which creates an interactive polygon. Clicking on the edge points of the polygon allows the tuning the shape (Figure 10A).

The user can segment a non-segmented cell, or modify the segmentation results using the ‘imfreehand’ function (by clicking ‘a’ and then drawing the cell borders) (Figure 10B). Cells to which a segmentation line is added or modified change their highlight to red (Figure 10C).

![Figure 10 - Process of manual segmentation and/or segmentation correction.](image)

If the polygon defined by the user intersects any existing segment, the action is queried with the possible actions being listed (popup menu in Figure 11). Otherwise, the new polygon is created.

If the new polygon does not intersect any existing segment, a new segment is created (Figure 10C). Otherwise, the action is queried (popup menu in Figure 11A), with the possible actions being listed.

Before closing the manual correction window, the user must save the changes by pressing ‘u’ and confirm by selecting ‘Yes’ from a new popup menu (Figure 11B).

![Figure 11 – Manual Corrections Popups. (A) Options allowed if the new manual segmentation overlaps with an existing cell segment; (B) Popup menu for applying and saving manual corrections](image)

If the new polygon (white borders in Figure 12) intersects with an existing polygon (blue borders in Figure 12A-1 or with two (or more) polygons (blue borders in Figure 12B-1), the popup menu in Figure 11A appears and requires the user to click different buttons. The result of clicking each button are shown in Figure 12 (red borders represent the resulting segments). There are cases where clicking on different buttons will give similar effects, especially when only one cell is intersected.
Figure 12 – Manual corrections when the segmentation overlaps with existing objects. (A-1) overlap with a single cell. Results of clicking on: (A-2) 'i' button; (A-3) 'u' or 'e' buttons; (A-4) 'a' button; (A-5) 's' or 't' buttons; (A-6) 'a' or 'r' buttons. (B-1) overlap with two cells. Results of clicking on: (B-2) 'u' button; (C-3) 'e' button; (B-4) 's' button; (B-5) 't' button; (B-6) 'r' button.

If the user clicks ‘i’, the resulting segment will be the intersection between the drawing segment and the existing segment (Figure 12-A-2). If clicking ‘u’, the resulting object is the union of the drawing segment and the existing segment (Figure 12A-3) or the union of all existing segments that touch the drawing segment (Figure 12B-2). By clicking on ‘x’ or ‘d’, the resulting segment is the subtraction of the drawing segment with the existing segment (Figure 12-A-4), even when two objects exist. If the user clicks on ‘e’, the resulting object is the extension of the largest segment (Figure 12A-3). For two existing segments, only the largest one is extended (see difference between red and blue borders in Figure 12B-3).

By clicking ‘s’, the result is the split between all touching segments. With 1 cell, the resulting object is two separated objects (Figure 12A-5). With two objects, the result is four new objects (Figure 12B-4). Clicking ‘t’ or ‘t’ is the same when only one object exists (Figure 12A-5). When two objects exist, if the user clicks on ‘t’ only the largest segment is splitted. Finally, the user can click ‘a’ or ‘r’, which splits and joins the drawing segment with the existing segment. If multiple segments touch the drawing segment, only the ‘r’ button will create a new segment based on the drawing and split all those touching the drawn segment.

2.1.3. Image Alignment
After segmentation, functional images can be loaded using the Microscopy Image Loading Interface (Figure 11).

Figure 13 - Activation of the Microscopy Image Loading Interface with the Load Images for Segmentation Pipeline

The alignment of morphological and functional images is automatic, as in CellAging (Häkkinen et al., 2013).

Figure 14 – Image alignment interface. (A) Activation of the Cell Alignment Interface options (B) Popup for the Execution of Local Adjustments during the Alignment Process.

When aligning single-time-point images, use the ‘No timestamps’ option (Figure 14). For timeseries, use the ‘Use t(\d+) pattern’ (Section 1). If the images lack timestamps or the required pattern, a warning message is displayed and the images are not loaded.

During alignment, a progress bar appears, which is interrupted with the Local Adjustments popup (Figure 12B). This popup is required because Functional Morphological images may not be recorded exactly at the same time, producing local image distortions. To execute the Local Alignment Adjustments, click ‘Yes’ when the popup menu appears (Figure 12B). Local adjustments are calculated by a local translation using the cross-correlation computation for each cell cluster. Two cells will belong to same cluster if the smallest distance between them is less than \( \frac{1}{2} \) of the mean cell width (obtained from all the segmented cells).

If the image registration problems persist, the user can perform manual alignment (Figure 15) developed in ‘iCellFusion’ (Santinha et al., 2015), which is based on applying a set of control points (blue dots, highlighted with white arrows in Figure 15, which can be added by clicking ‘n’) and moving them to
alter the original segmentation. A control point can be deleted by clicking ‘d’ and selecting the point to be deleted.

Figure 15 – Manual Alignment Strategy with Control Point (blue dots) Mapping. Dots are highlighted with arrows.

2.2. Loading Segmented Images

If the user has a mask prepared and aligned, the mask can be loaded into the tool (Figure 4) by clicking on ‘Load Segmented Images’ of the Save and Load User Interface. A load of the respective masks will activate the structure detection panels (Figure 16). Functional or morphological images can be loaded and merged with the corresponding masks.
The resulting logical mask and the activation of the specific panels can be observed if an RGB mask is loaded (Figure 17).
An example of an RGB mask is shown in Figure 18. The file should be in tiff format. Each cell is considered to be a different object (Section 2.5). Thus, each cell has a distinct colour.

![Figure 18 – Example of a RGB image with several segmented cells.](image)

2.3. Loading Handles

If the user wants to export (Section 2.6) the current state of the analysis (all buttons pressed and loaded images), the user can save a handles file (with .m type file), allowing to continue the analysis at a future point in time.

2.4. Structure Detection

Once having segmented masks, the user can load functional images. The loading of these images automatically launches the GPL algorithm for detection of seeds in the cells (except in ‘spot detection’ images). The execution of the GPL algorithm can be made remotely by activating the ‘Force GPL remote’ option in the Visualization Panel (Figure 19). The user can run the GPL algorithm on the UNINOVA servers (from http://griduni.uninova.pt/GPL_WEB/) or locally, by not checking the remote implementation.
2.4.1. Gaussian Segmentation

Nucleoids in live cells can be detected and segmented (Oliveira et al., 2016) using the GPL algorithm (Mora et al., 2011). This method labels each pixel, based on its gradient azimuth, creating a gradient path. Then, labels are reduced by tagging them as equivalents, which happens when two labels belong to the same maximum. After this step, the position and number of seeds can be obtained. The seeds are used for the Gaussian Segmentation algorithm tailored for DAPI-stained nucleoids segmentation (Oliveira et al., 2016).

To apply the Gaussian Segmentation Algorithm, the user needs to select the option ‘Gaussian Fit’ on the dropdown menu of the specific Structure (Figure 20) and click “Structure Detection” (which is specific for each structure).

When the Gaussian Segmentation Method is chosen, a new window appears (Figure 21). One of the parameters that can be changed is the amplitude profile adjustment (“d” parameter) between square shape, bell shape and thin shape (Oliveira et al., 2016). When two seeds are used but the fitting of the Nucleoids overlaps (Figure 21) on can reject the overlapped Nucleoids or reject nucleoids if the overlap is bigger than X% of the total area of the Nucleoids (X can be changed in the edit box). The third option is to not reject any overlapping Nucleoids, and the last option is to consider the overlapping Nucleoids as a single Nucleoid. The user can see the Gaussian Fitting of each cell, by clicking on the checkbox ‘Show Gaussian Fittings’. Figure 22 shows examples of 1 and 2 Nucleoids fitting using the Gaussian Segmentation Algorithm.
2.4.2. TreshMorph’ Segmentation

The ‘TreshMorph’ Algorithm allows automatic analysis of cellular structures without requiring seeds selection. Only the thresholding method needs to be selected. It is activated by
selecting the option ‘Morphological Fit’ on the dropdown box of the specific Structure (Figure 23) and clicking on the button “Structure Detection” (specific for each structure).

![Image](image.png)

**Figure 23 Activation of the ‘TreshMorph’ Algorithm**

When selecting the ‘TreshMorph’ Algorithm, a new window appears on the screen of the Toolbox (Figure 25) with the Parameters that can be changed.

The first step is to select a threshold level to separate the structures of interest from the background. Three threshold methods can be selected: a) the Global Otsu's Global image threshold method (Otsu, 1979), which minimizes the intraclass variance of the black and white pixels (using the ‘graythresh’ function); b) the Multilevel image threshold (using the ‘multithresh’ function) that allows defining the number of levels (second edit box in Figure 25) and which levels are used to threshold the image (first edit box in Figure 25); c) a threshold based on the mean and standard deviation of the intensity in the cell, where the amount of the standard deviation to be added to the mean intensity can be chosen.

Next, the ‘im2bw’ function can obtain the binary images (mask) of the structures based on the threshold. This is followed by the application of the ‘bwmorph’ function with the ‘majority’ operation, which sets a pixel to 0 or 1 if 5 or more pixels in its 3-by-3 neighborhood are 0s or 1s, respectively.

The next step is a morphological ‘closing’ operation (by selecting the radio box ‘Morphological Close’ in Figure 25) which performs a dilation followed by an erosion, to enlarge and smooth the boundaries of the structures, while also removing small holes in the mask. The user can use the ‘opening’ operation instead, by selecting the radio box ‘Morphological Open’.

Finally, all objects with a size smaller than X pixels can be removed from the analysis (X can be changed in the edit box ‘Delete objects smaller than X pixels” in Figure 25), using the ‘bwareaopen’ function, if this step does not remove all the objects inside the cell (if it does, this
step is skipped), which finalizes the automatic object segmentation. The workflow process of the ‘TreshMorph’ Segmentation Algorithm is provided in Figure 24.

Figure 24 – Workflow of the ‘TreshMorph’ Segmentation Algorithm.

Figure 25 - Activation of the Morphological Fitting parameters window for a Nucleoid Detection Example.
2.4.3. **Spot Detection Algorithms**

An example of spot detection is shown in Figure 26. Three spot detection algorithms can be used. The options for each method (Median, Kernel and Gaussian) are shown in Figure 27. For more information on each parameter see (Häkkinen et al., 2013).

![Image of Spot Detection with options](image.png)

**Figure 26** – Example of Spots Segmentation with the options: Zoomed Normal View, Show Bacterial Borders, Show Structure Borders and with Seed Removal Option also activated, using the median Algorithm.
2.4.4. Nucleoid Detection

Nucleoids can be detected using the Gaussian Segmentation Algorithm or the 'TreshMorph' Algorithm. Parameter ‘d’ can be used to tune the results depending on the images. Normally, objects smaller than 20 pixels should be removed and the morphological operation “closing” should be used. An example of Nucleoid Detection with the ‘TreshMorph’ Algorithm is shown in Figure 28.
2.4.5. FtsZ Ring Detection

FtsZ Rings can be detected with the Gaussian Segmentation Algorithm or with the ‘TreshMorph’ Algorithm. Several parameters might be tuned to best segment this structure. Parameter ‘d’ can be used to tune the results depending on the images. Normally, objects smaller than 20 pixels should be removed and the morphological operation “closing” should be used. An example of FtsZ Detection with the ‘TreshMorph’ Algorithm is shown in Figure 29.

Figure 28 – Example of Nucleoids Segmentation with the options: Zoomed Normal View, Show Bacterial Borders, Show Structure Borders and with Seed Removal Option, using the Morphological Algorithm.
2.4.6. Min System Detection

Fluorescent proteins of the Min System can be detected with the Gaussian Segmentation Algorithm or the ‘TreshMorph’ Algorithm. Parameter ‘d’ can be used to tune the results depending on the images. Normally, objects smaller than 20 pixels should be removed and the morphological operation “opening” should be used. An example of the detection of MinD proteins using the ‘TreshMorph’ Algorithm is shown in Figure 30.
2.4.7. **Inclusion Bodies Detection**

The panel for the Detection of Inclusion Bodies will be activated when a Phase-Contrast image is loaded using the ‘Load Images for Segmentation’ button (Section 2.1.1), after the cell segmentation algorithm is executed. This panel can also be activated if a mask is loaded and a Phase-Contrast image is merged with it.

Initial seeds are calculated using the GPL algorithm (Section 2.4 and Figure 19), followed by a rejection algorithm that rejects inappropriate seeds (Figure 31). Seeds approved have yellow colour, while rejected seeds are in red. Seeds manually added or confirmed (Section 2.5.1) are in green (Figure 31).
Figure 31 - Example of Inclusion Bodies Detection. Yellow seeds are accepted seeds for detection. Green seeds are seeds manually added (by clicking on the area inside the bacteria) or seeds that were previously red or yellow and were clicked. Red seeds are seeds that have been rejected by the rejection algorithm.

2.4.8. Undefined Structures Detection

Undefined structures can be segmented, using one of the algorithms that implemented in the tool. The dropdown box provides access (Gaussian Segmentation, ‘TreshMorph’ Segmentation and Spot Detection).

2.4.9. Multiple Structures Detection

Having all images aligned to the Masks, load each structure, which will activate each Analysis Panel, allowing the user to perform various structures’ analysis at the same time (Figure 32).

For this, use the dropdown menu in the ‘View:’ option (Figure 33) to select the desired structure, and the select the specific analysis method in the Analysis Panel dropdown box.
Figure 32 - Visualization Panel of the analysis of multiple channels (Red – Nucleoid and Green – FtsZ Rings).

Figure 33 – Option Selection for multiple structure Detection

2.5. Visualization Panel and Frame Visualization

Zoom is enabled by clicking the ‘z’ key or right-click the mouse to assist the manual seed correction. The zoom ratio (defaulted at 1/8) can be changed by clicking the ‘shift’+’m’ key (to increase the zoom ratio by 1/32) or ‘shift’+’l’ (to decrease the zoom ratio by 1/32). Pressing ‘shift’+’z’ resets the zoom ratio back to 1/8. The 1/8 zoom visualization is shown in Figure 34.
The Visualization Panel GUI (see Figure 35) allows calculating the GPL seeds by a remote server, or by the local computer.

The Visualization Panel also allows removing the GPL seeds visualization (Figure 36), show the internal ID number of each bacteria (Figure 37), show the segmented bacterial borders (Figure 38), compare the original image and the masked image (Figure 39) and observe the structure borders (Figure 40).
Figure 36 - Visualization panel effects - Zoomed Normal View with Seed Removal Option activated

Figure 37 - Visualization panel effects - Zoomed Normal View with Bacteria ID visualization Option and with Seed Removal Option activated
Figure 38 - Visualization panel effects - Zoomed Normal View with Bacteria Segmentation Borders Option and with Seed Removal Option activated.

Figure 39 - Visualization panel effects - Zoomed Normal View with Show Original Image Option, show Bacterial Borders and with Seed Removal Option activated.
The Frame Visualization Panel GUI (Figure 41) allows the user to change the type of the image and structure to visualize (Section 2.4) in the drop-down box menu (‘View’) of the Panel. It is possible to navigate from the first to the last frame of a time-series using the ‘Previous’ and ‘Next’ Button. Current Frame/Total Frames are shown in the ‘Frame’ textbox.

**2.5.1. Seeds Manual Correction**

Seeds can be automatically removed as explained in the Detection of Inclusion Bodies Section, turning them into red colour. Seeds automatically placed and not manually removed are shown in yellow. New seeds can be manually added and yellow seeds can be removed (in all image modalities) by clicking on the pixel where the seed is located. When creating a new seed, if a neighbour pixel is a seed it is deleted. Added seeds appear in green (Figure 42).

Seeds removal can also be done by drawing an interactive a freehand region of interest (ROI), by clicking the ‘d’ key. Once complete, any seed inside it is deleted. All seeds are deleted by pressing ‘alt’+’d’.
In addition to the existing seeds (red, yellow and green), the user can add markers inside a cell, which can be used to mark interesting features inside the cell. Two distinct markers (Figure 42) can be added by clicking ‘o’ (to add orange marks) and ‘p’ (to add pink marks) and clicking inside the cell. SCIP saves the coordinates of each marker.

Figure 42 – Manual Seed Correction. Seeds automatically selected for deletion are shown in red. Yellow seeds were not automatically selected for deletion. Green seeds were manually added. Some orange and pink marks are also shown.

2.6. Exporting Results

One option of the “Load and Save” menu (Figure 4) is to save a Handles Files, which must be saved with MATLAB versions higher than R16. It contains the fields: ‘output’, ‘my_handles’, ‘currFrame’, and ‘self’ (Figure 43).
The ‘output’ field is a 1-by-K array of structures, where each element has the information on frame K. All the fields in ‘output’ are shown in Figure 44.

The ‘my_handles’ and ‘self’ structures save the state of the buttons that have been pressed, the state of the dropdown, radio and check boxes. The ‘currFrame’ saves the frame that was currently being visualized.

The output structure has the following fields for each frame:

- FileName: the name of the file of the specific frame
- conDICIndices: the indices used to align different frames
- mask: the masked image for the specific frame
- Ncells: the number of cells
- Cells: a structure with the information regarding each cell (Figure 45)
- ImgBactBorders: the borders of the masked image
- *1*Fitted*2*: *1* represents the structures (Ring, Nucleoid, Min, etc) while *2* represents the method of analysis (Morphological, Gaussian). If the value is 0, the method was not used, while if it is 1, it has the analysis results.
- LabelPixListFLAG: flag to show if the Bacteria ID visualization Option has been used
- Aligned: flag that the images have been aligned with the mask
- FileName*1*: file name of the images containing structure *1*
- Image*1*: actual image containing structure *1*
- imageMasked*1*: multiplication of the original mask with the actual image containing structure *1*
- GPLLabels*1*: GPL seeds data, including the seed position and the maximum and minimum intensity of the surrounding pixels.
The structure ‘cells’ (Figure 45) for each cell contains:

- id: cell identification number.
- PixelList: Coordinates of pixels in the region, returned as a p-by-2 matrix.
- PixelIdxList: Linear indices of the pixels in the region, returned as a p-element vector.
- Pos: top left coordinates (x, y)
- posMax: bottom right coordinates (x, y)
- DimX: cell size in the x axis
- DimY: cell size in the y axis
- Boundaries: Locations of the borders pixels in the region, returned as a p-by-2 matrix.
- PinkMarks: position of pink marks
- OrangeMarks: position of orange marks
- meanIntensity*1*: mean intensity for each cell of the image with structure *1*.
- Fitted: flag indicating whether the cell has been fitted or not
- *1*Fitted: flag indicating if a specific structure *1* has been fitted
• GPL*1*Maxs: the identification of the GPL seed inside the cell
• GPL*1*Status: -3 for red seeds, 1 for yellow seeds and 2 for green seeds
• Mask: Cell mask
• Subimage*1*: pixel values inside the image with a specific structure *1*, background pixel are represented as NaN
• Data*1*: pixel values inside the image with a specific structure *1*, background pixel are represented as 0’s
• Parent: in a timeseries, this is populated by the ID of the parent
• *1*Morph: mask obtained by the Morphological Segmentation for structure *1*
• *1*MorphFitted: flag to check if structure *1* is fitted with the Morphological Segmentation
• *1*s: position, mask and extremities of structure *1*, as detected by the Gaussian Algorithm (one for each green or yellow seed inside the cell).
• *1*MaskBorders: borders of the mask detected by the Gaussian Algorithm

With these fields, one can calculate various parameters, such as size and center of the object, etc.
1 x 16 struct array with fields:

id
pixelList
PixelIdxList
pos
posMax
DimX
DimY
boundaries
PinkMarks
OrangeMarks
meanIntensity
meanIntensityNuc
NucFitted
Fitted
GPLNucleoidsMaxs
GPLNucleoidsStatus
mask
subimageImageNucleoid
dataNucleoid
parent
meanIntensityRing
RingFitted
GPLRingsMaxs
GPLRingsStatus
subImageRing
dataRing
RingMorph
RingMorphFitted
RingMorphMaskBorders
Nucleoids
NucleoidMaskBorders

Figure 45 – Fields in Structure cells

3. References


