# Clustering analysis in SMLM data using Diinamic

In contrast to the images obtained with optical microscopy, SMLM techniques generate datasets of molecular coordinates. Many characteristics arising from the preparation of the sample as well as the targeted molecule itself make some type of analysis algorithms more suitable than others. Some analysis strategies may be more appropriate than others depending on how the experiment was run.

Diinamic (*Density and Image INtensity based Analysis Method for Identification of Clusters*) is a MATLAB-based modulable sequence of cluster detection steps that couples a space fragmentation method with a local density calculation. The analysis can be performed in two (cluster detection) or three phases (cluster and intra-cluster subdomain detection).

The first phase preselects pixels or detections that are likely to belong to clusters, using density-based criteria. Next, these pixels are then used to create candidate clusters by coalescence.

- Diinamic-R algorithm selects candidate pixels by intensity-based segmentation of the rendered image. The pixel size of this image determined the grid used to calculate density. In addition to the intensity threshold, it is possible to use an additional density threshold applied to each pixel. These two thresholds are chosen to eliminate pixels with a value of intensity and/or a number of detections too low to be considered being part of a cluster.
- Diinamic-V selects candidate clusters by Voronoi tessellation: Voronoi polygons whose size is below the threshold are retained to create candidate clusters.

Phase two selects clusters among the candidates that fulfill user-defined parameters (minimum density, minimum and maximum size). This step is aimed to discard clusters which are likely to be false (i.e. due to multiple detections of one single-molecule) or badly defined (not enough labeling, out of focus, etc.). Density is calculated in a pixel-wise manner, using the pixel size of the rendered image.

The third phase, optional, looks for subdomains with different densities in the selected clusters using DBSCAN.

**This user manual explains:**



**Disclaimer**: the scripts and this help were written to be run on Windows-based systems. If you are using Mac OS, you have to replace the slashes in path names: replace all the "\" by "/". No other incompatibility was detected by us so far; but do not hesitate to contact us if you are having difficulties to run the programs.

## Licence, how to cite and contact details

Please refer to the article for more information about how the analysis is done and now parameters must be chosen:

**Paupiah et al (2023). "Introducing Diinamic, a flexible and robust method for clustering analysis in single molecule localization microscopy". Biological Imaging, in revision.**



If you have difficulties to implement the analysis, please contact marianne.renner@sorbonne-universite.fr

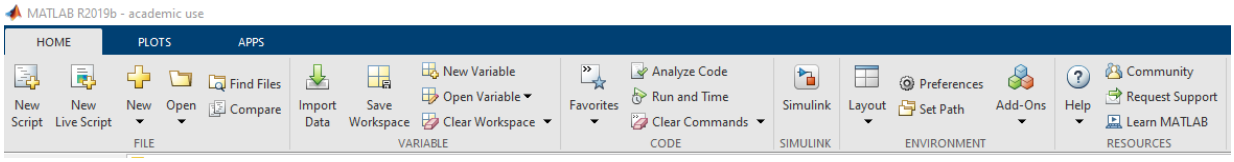
## Running scripts in MATLAB

If you are familiar with MATLAB, you can skip this part!.

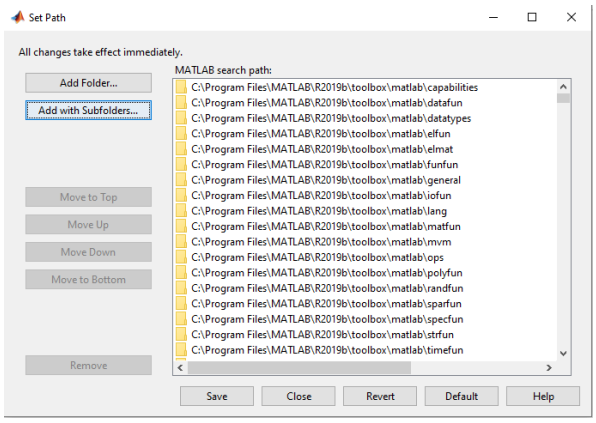
### Set the Path

The first thing to do is to set the path to the folder containing Diinamic scripts. To proceed, you can use the "Set Path" shortcut in the Home/Environment menu:

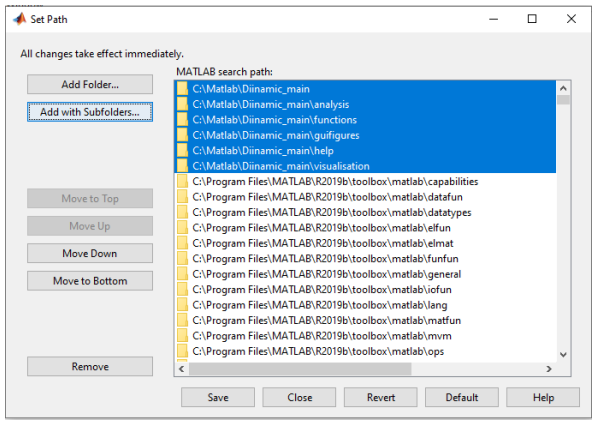
Example: Matlab 2019




A window pops up:




Choose **Add with Subfolders** and look for the Diinamic folder




If you **Save** the path (you need administrator rights), you will not need to do this operation again. Otherwise, the path will be lost when MATLAB is closed.

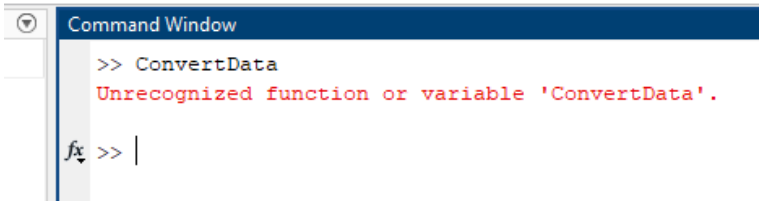
## Run the programs

To run the scripts, which are all based on graphic user interfaces (GUI), you only need to type the name of the GUI in the command window and press Enter.

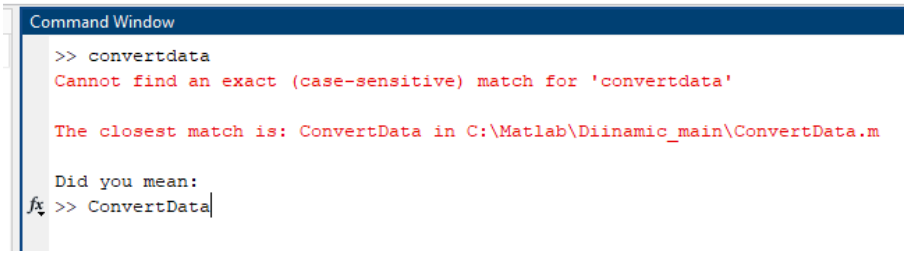
The available GUIs are:

- ConvertData.m: to convert .csv data into a format usable by Diinamic.
- Segment.m: to create ROIs.
- Cluster.m: to perform clustering analysis.

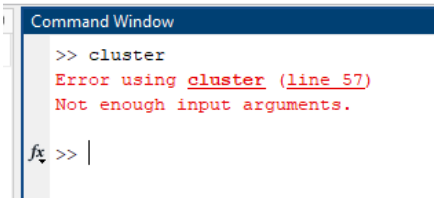
**Note that if you try to execute the programs without setting the path, you will get the following error:**



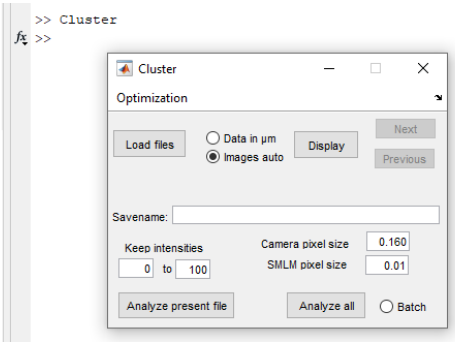

**Please note that MATLAB is case-sensitive**: please pay attention to capital letters. If you type "convertdata" instead of "ConvertData" you will obtain an error message:




In this case, MATLAB proposes the correct spelling, but this is not always the case:



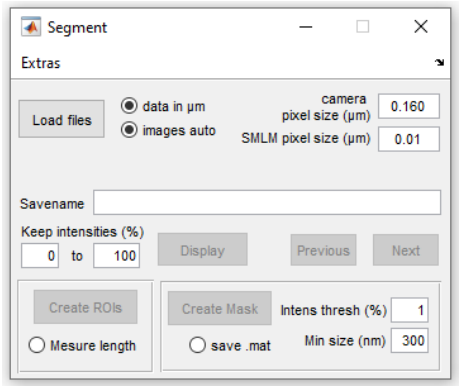

If you type "cluster" instead of "Cluster", MATLAB tries to execute the "cluster" function and not the "Cluster" GUI. Typing "Cluster" you obtain:



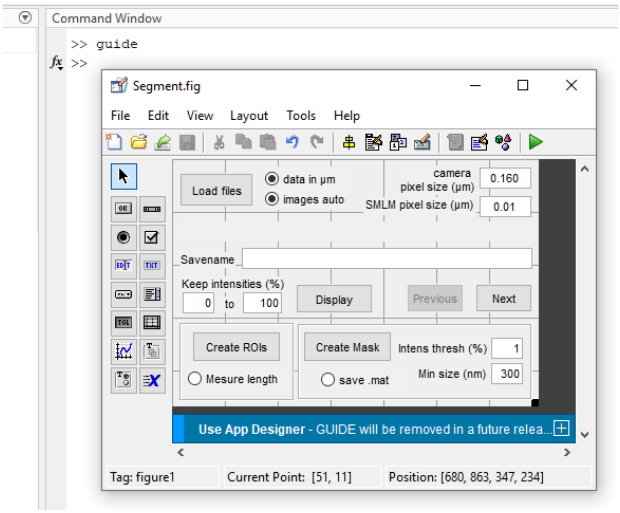

If you get other error messages, there could be a problem of MATLAB version (some functionalities may be absent or different). Please make a screen capture of the error and contact us!

## Changing the default values in the GUIs

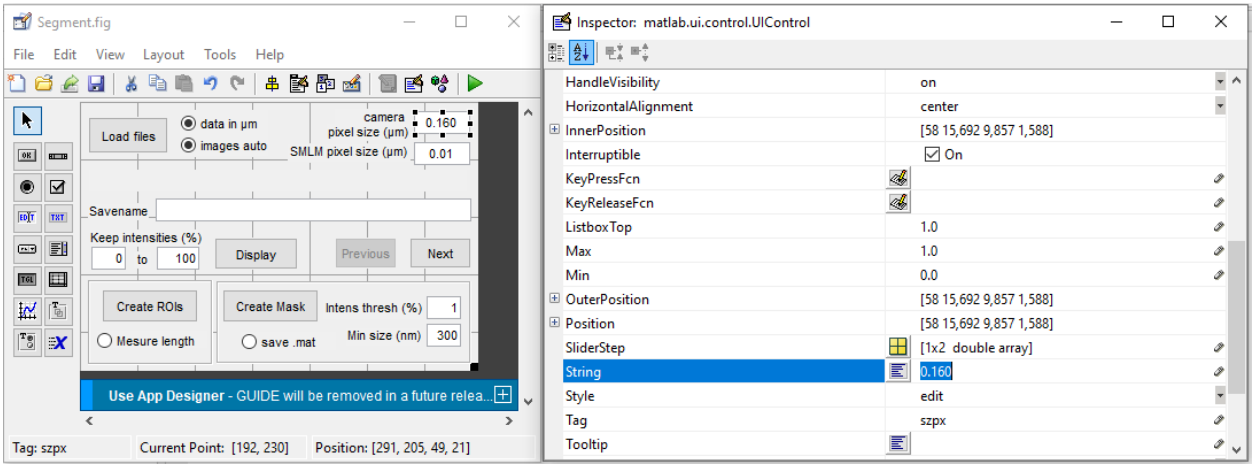
GUIs windows are MATLAB .fig files that can be easily modified. For example, for Segment.m, the GUI figure is Segment.fig:



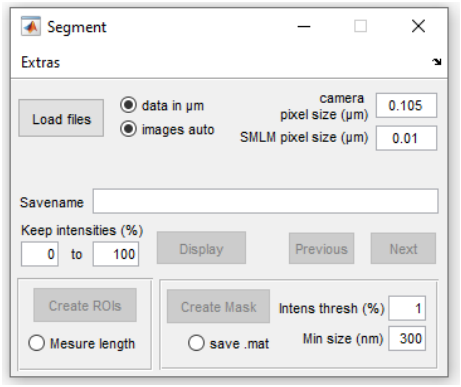

Type “guide” on the command window and browse to find the GUI window that you want to modify in Diinamic-main\guifigures.




Double-click on the feature that you want to change; this opens the Property Inspector. For example, to modify the camera pixel size from “0.160” to “0.105”, double-click on it and scroll down on the Inspector to find the “String” value:



Type the new value, save the changes. When you call again Segment.m, you will see:




## Problems with the font size in GUIs

GUIs were optimized for a 1920 x 1080 resolution, but the size of fonts should automatically be readjusted to your screen definition. If it is not the case, find the lines below in the programs with GUI (ConvertData.m, Segment.m, Cluster.m, DetectClusters.m, DetectClustersBatch.m and OptimDetectClusters.m) and change the active line that sets the font size. They are at the beginning of the code.

```
% to fit screen definition/changes in size
h3 = findobj('Type','figure');
txtHand = findall(h3, '-property', 'FontUnits');
%set(txtHand, 'FontUnits', 'normalized');            -> line not active
set(txtHand, 'FontUnits', 'centimeters');            -> line active
```

Note: the character "%" indicates a commentary, so a line with this character at the beginning is not executed.

You can try to :

1) Activate the option 'FontUnits'= 'normalized' instead of 'FontUnits'= 'centimeters'

```
% to fit screen definition/changes in size
h3 = findobj('Type','figure');
txtHand = findall(h3, '-property', 'FontUnits');
set(txtHand, 'FontUnits', normalized);               -> line active
%set(txtHand, 'FontUnits', centimeters);             -> line not active
```

2) Or leave the option 'FontUnits'= 'centimeters' activated and set the size that you want by adding the following line at the end : `set(txtHand, 'FontSize', the value that you want ) ;`

```
% to fit screen definition/changes in size
h3 = findobj('Type','figure');
txtHand = findall(h3, '-property', 'FontUnits');
%set(txtHand, 'FontUnits', 'normalized');            -> line not active
set(txtHand, 'FontUnits', 'centimeters');            -> line active
set(txtHand, 'FontSize', 0.5) ;
```

In this case, all the fonts will have a size of 0.5 centimeters (change this value at your convenience).

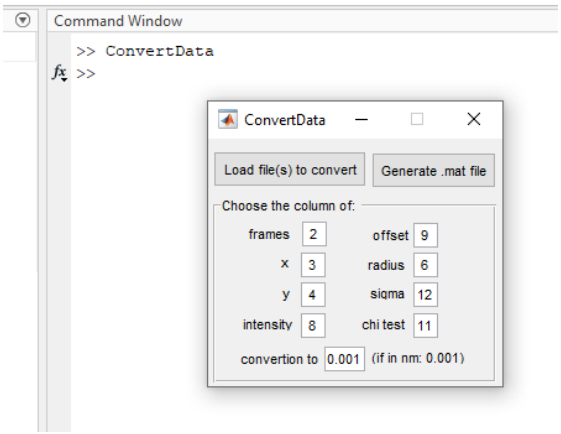
# Running Diinamic

To use Diinamic, you need to:

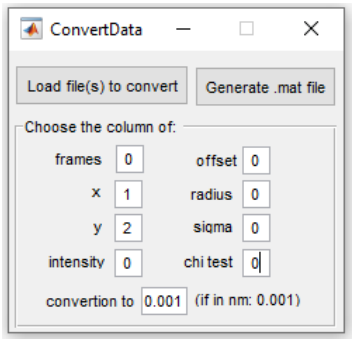
1) Perform single molecule detection on your recordings, correct for stage drift if needed and correct for multiple blinking if possible.
2) Convert your detection data. Diinamic uses its proper data organization stored in .mat type files (MATLAB-type format). You can convert your data files from .csv format into Diinamic format using ConvertData.m (more explanations below).
3) Select ROIs using Segment.m: see the dedicated chapter below.
4) Run Cluster.m to choose the type of clustering analysis: see the dedicated chapter below.

## Prepare your data: ConvertData.m

Call ConvertData GUI. On the window that pops up, choose the list of files to convert with "**Load file(s) to convert**" and indicate the column number that encodes the different type of data indicated on the window. By default, the column numbers correspond to .csv files produced by ThunderSTORM.




For example, to convert data that contains only three colums (x,y, cluster number):

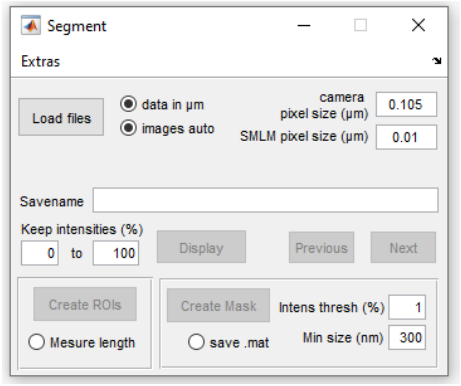



**Please pay attention to the units!** By default, the program considers that coordinates are in nm.

Please ask for help if your data is different and you do not know how to modify the code.

# Create ROIs: Segment.m

This GUI serves to select ROIs and to create segmented images from SMLM data.



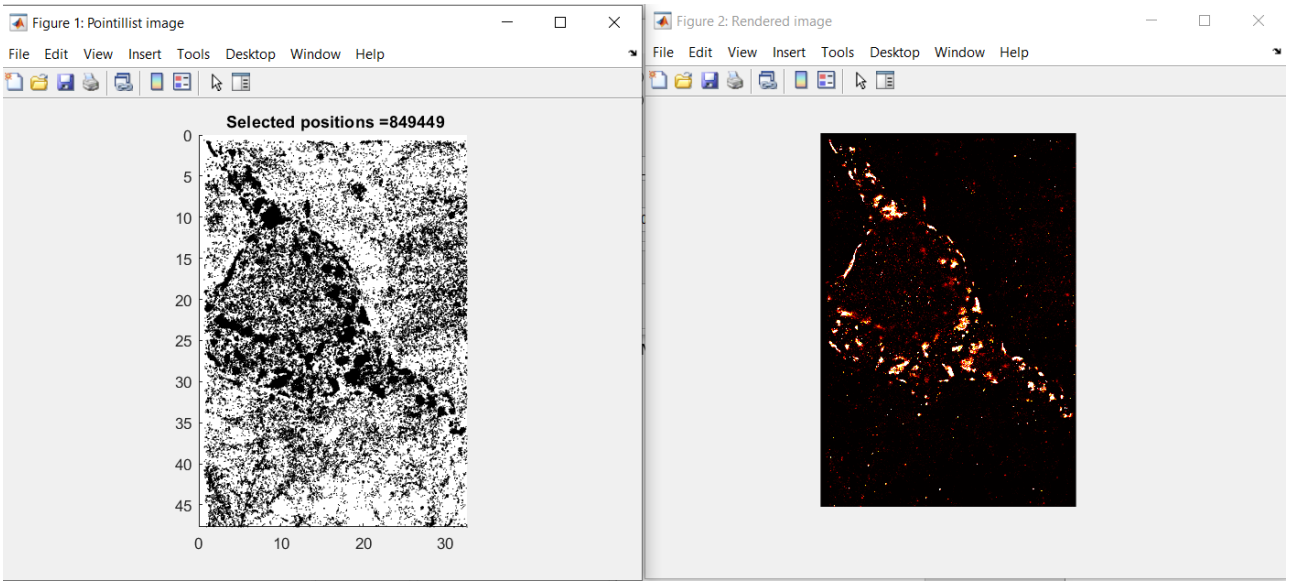

Enter the **camera pixel size** and the **SMLM pixel size** (half of the localization precision) of your set-up. The camera pixel size is needed only if your data is not calibrated in nm or µm.

Use **Load files** to select the files to be treated (.mat files). If your data is already calibrated in microns (i.e. you used ConvertData.m) , click on **data in µm** before loading the files.

The menu Extras gives the possibility to create ROIs of the full images in batch mode. If you want to analyze the whole image at once, use this option. As a piece of advice, if your image contains regions with different background noise, it is better to choose smaller ROIs in which the background noise is rather homogeneous.

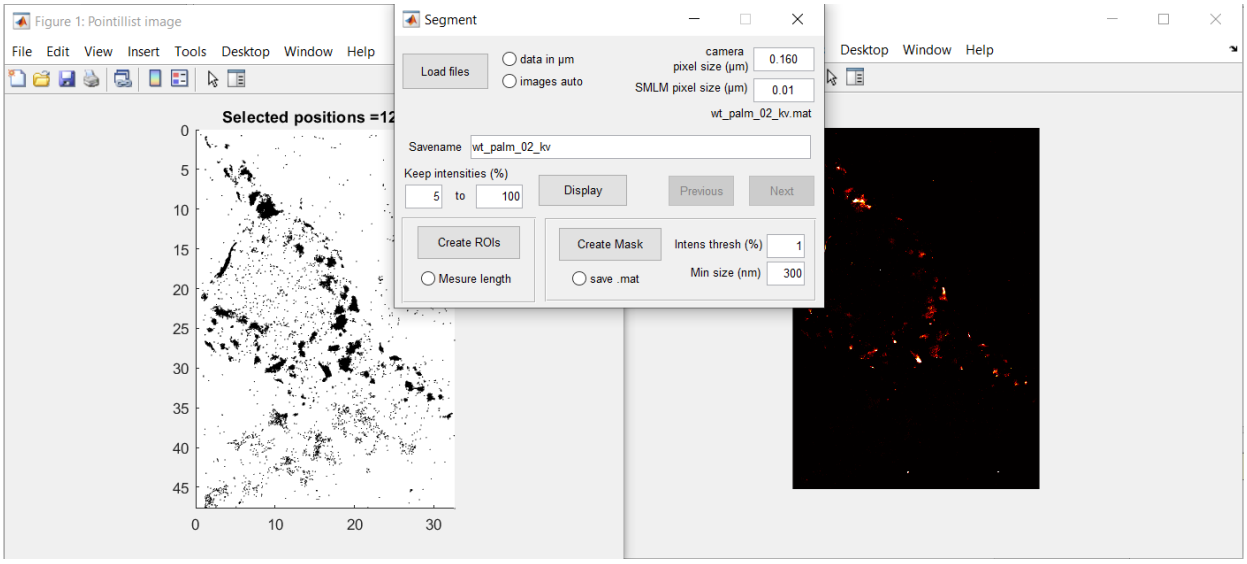
The resulting images will be saved with the **Savename** indicated (modifiable). Be careful if ROIs are going to be used for clustering analysis, as they have to be automatically recognized by the analysis program. The name has to be *name*.mat for the detection file, and *name*.rgn for the ROI datafile.

If **images auto** is selected, the pointillistic and rendered images will be displayed automatically for the active file. Otherwise, you can use **Display** to visualize them. If you loaded a list of files, use **Next** or **Previous** to move to the next or previous file, respectively.




**Keep intensities (%)** can be useful to eliminate low intensity or high intensity spurious signals: set the low and high thresholds (expressed as percentage of maximum intensity) and use Display to verify the selection.

For the previous example:




Please note: the ROI (.rnd) will be created from the *original* set of detections, whereas the mask file will be created using the detections whose intensity is within the low and high limits.

Use **Create ROIs** to select ROIs by hand and to manage previous selections. If you want to add a distance measurement, select **Mesure length**.

**Create Mask** will create a segmented image from the rendered image (no ROI in this case).

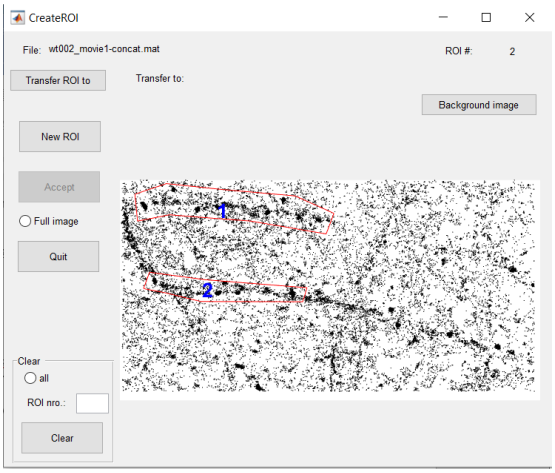
## Create ROIs

This will open a new window where you can select ROIs by hand on the pointillistic image. If there is already a selection for the active file (a .rgn file), this selection will be shown.

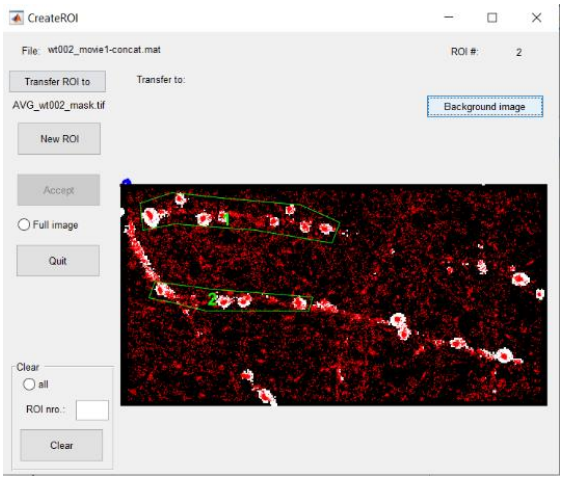
You can overlay a background image (.tif) loading it with **Background image**. **Transfer ROI to** will create the same ROIs also in the selected .mat file (please note that both images must have the same size). **Quit** to come back to the Segment window.

*Example with previous ROI data (2 ROIs):*

*No background image:*



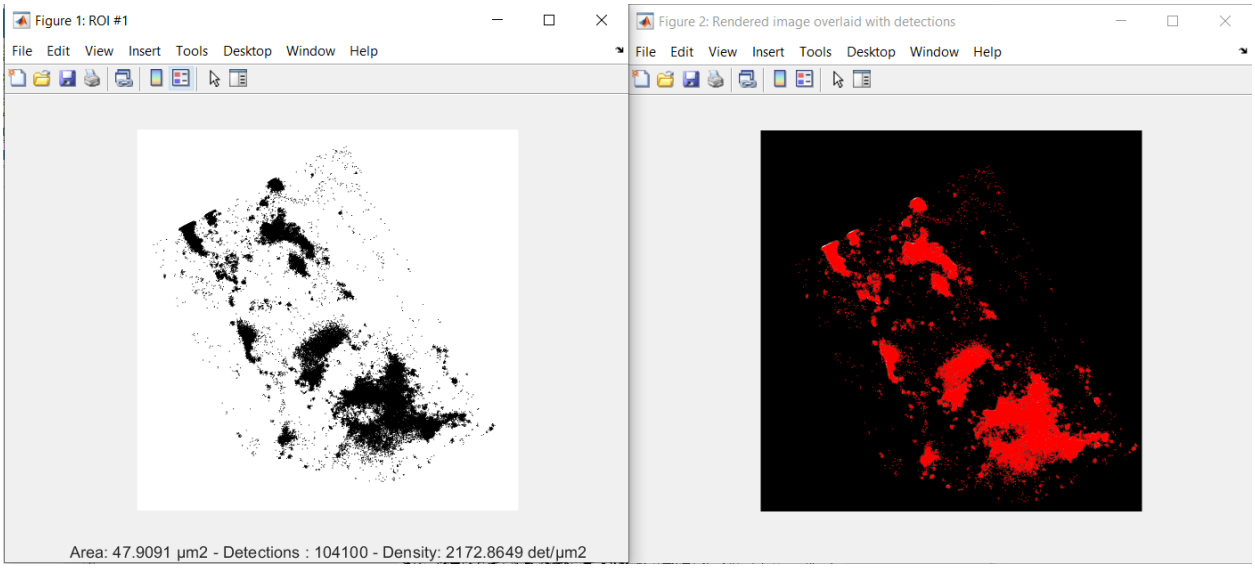

*With background image:*



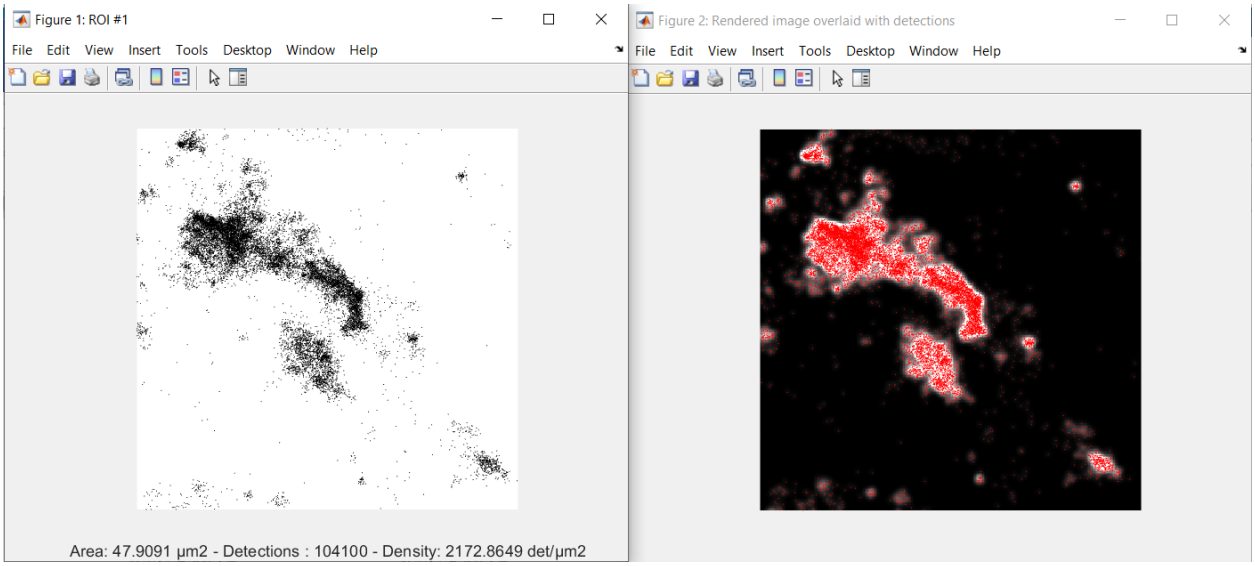
## Drawing ROIs

You can select a polygon on the image (ROI) by clicking on **New ROI**. If you want to keep the whole image as ROI, choose **Full image** before clicking on New ROI.

After drawing the polygon (double click to finish it), the pointillistic and the rendered image of the ROI (overlaid with the detections in red) will pop-up.



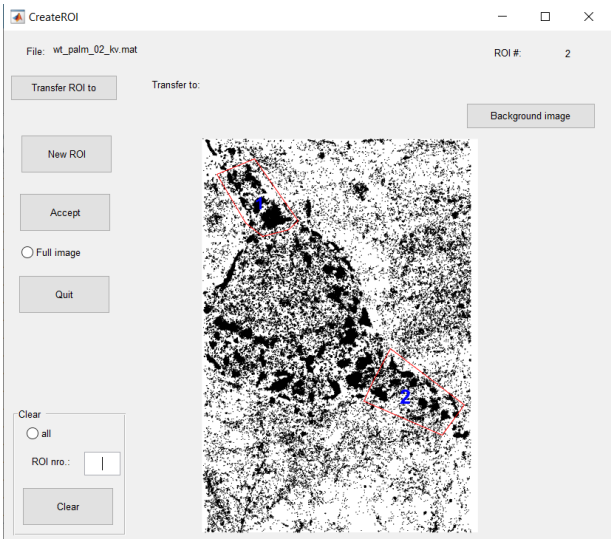

A zoom on the images:




Note that on the pointillistic image window you have the area of the ROI, the number of detections found inside and the density of detections in the ROI, assuming a homogeneous distribution.

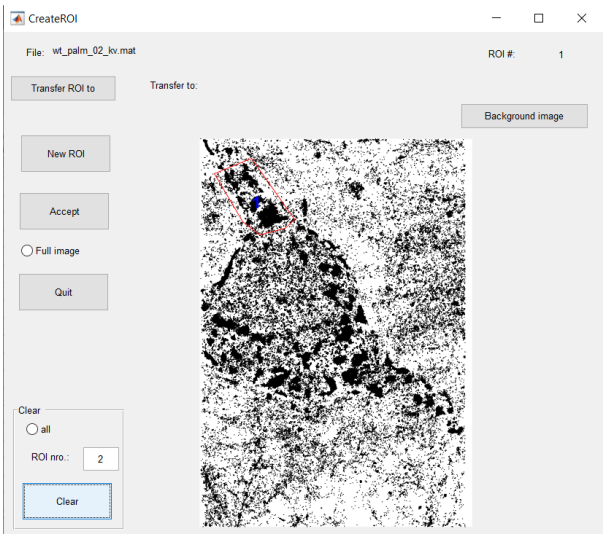
On the CreateROI window, click on **Accept** to keep the selection, or click on **New ROI** if you want to select another one. “Accept” will number the ROI and show this number on the window.

You can add as many ROIs as you want. As an advice to facilitate clustering analysis, select rather small regions with a homogeneous background level, if possible.




## Deleting ROIs

You can delete one ROI using **ROI nro** and **Clear**. After deleting one ROI, the remaining ones are re-numbered. Choose **all** to delete all the ROIs.




The results will be saved in:

1) *name*-ROIinfo.txt
   Text file with a table where the columns are

| ROI # | surface area ($\mu m^2$) | # detections in the ROI | density of the ROI (detections/$\mu m^2$) |
|---|---|---|---|

2) *name*.rgn

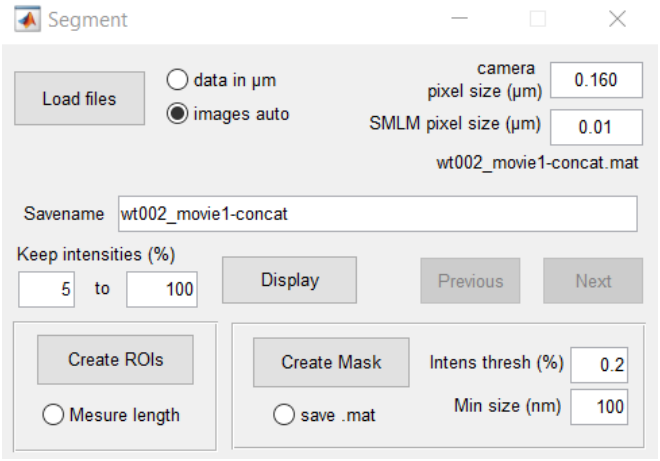
   .mat file with the structure ROI that contains the coordinates of the ROI and the detections that colocalize, among other data.

3) *name-roi#*-Irend.mat

   .mat file with the rendered image of the ROI, which is used by Cluster.m for clustering analysis.

## Create Mask

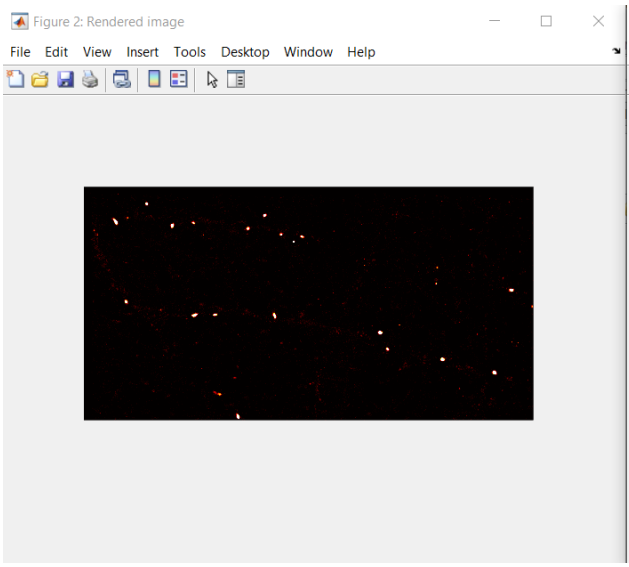
Create Mask makes a segmentation on the rendered image. You can set an intensity threshold with **Intens thresh** (in % of the maximum) and a minimum object size with **Min size**. This will select objects in the segmented image that have at least this size, assuming a circular shape. Play also with **Keep intensities** values to improve the segmentation.



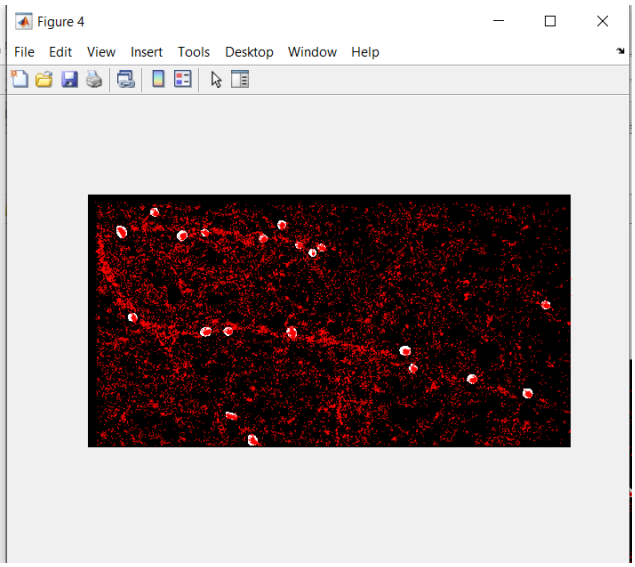

After clicking on **Create Mask**, a new window shows the result overlaid with detections plotted in red: zoom to see the objects in the mask that appear in white.

In comparison with the rendered image:

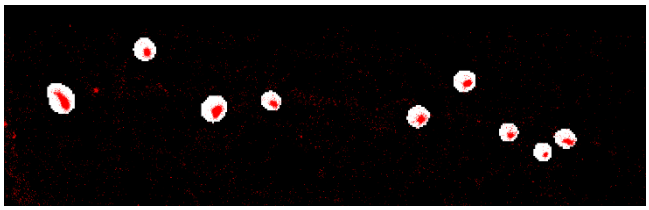
*Rendered image*



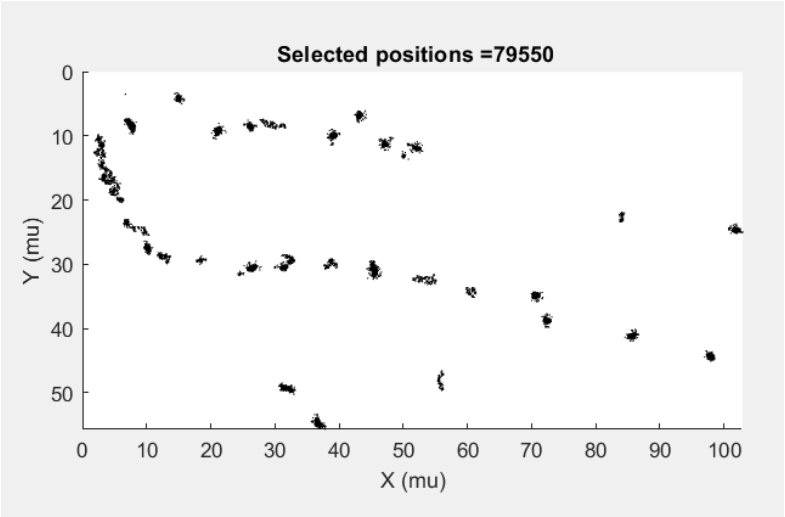

*Mask overlaid with detections:*




*-> a detail after zooming:*



Select **save .mat** to obtain a new .mat file that contains only detections in the selected objects (*name*-mask.mat). In the previous example, the pointillistic image of the resulting *name*-mask.mat file is:

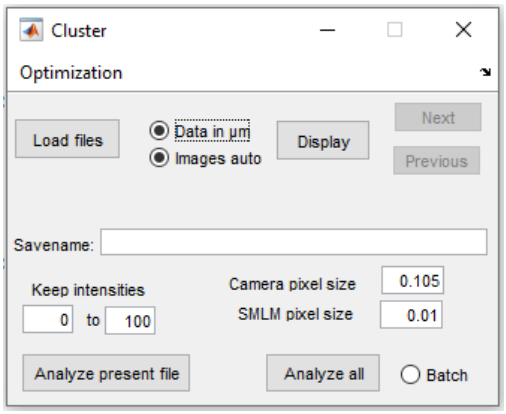


# Run the analysis: Cluster.m

This window is the main menu for clustering analysis. Use **Load files** to select the .mat files containing detections. The analysis is done only on ROIs, created with **Segment.m**. If you want to analyze the whole image at once, you must create a ROI covering the entire image. See the chapter dedicated to Segment for more details.

***Please note:***

- Click on **Data in µm** if your detection data is already converted to µm, before loading the files.
- Cluster.m requires files named *name*.mat (detections), *name*.rnd (ROIs) and *name-ROI#-*Irend.mat. The last two files are created by Segment.m



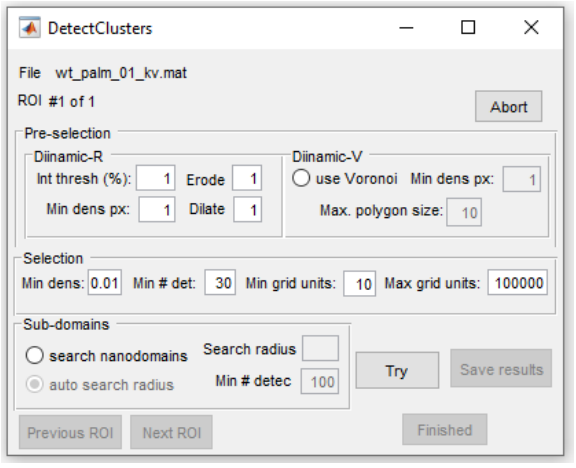

If **Images auto** is selected, the pointillistic and rendered images will be displayed automatically for the active file. Otherwise, you can use **Display** to visualize them. If you loaded a list of files, use **Next** or **Previous** to move to the next or previous file, respectively.

**Keep intensities** options may help the visualization of pointillistic ad rendered images by eliminating detections with intensities below or above the minimum and maximum values provided (in %). Please note that *all* the detections will be considered for the analysis.

The **Optimization** menu opens a new GUI dedicated to executing clustering analysis iteratively to choose the best parameters. See the dedicated chapter below “Choosing and optimizing Diinamic parameters”.

**Analyze present file** will, as it says, perform the analysis only on the present file. If you loaded a list of files, **Analyze all** will sequentially analyze all of them. In these two cases, the clustering detection will ask for confirmation at each step and you will be able to tests different parameters on the same file. When you select **Batch** instead, **Analyze all** will ask to set the parameters for the first file and then it will automatically analyze all the files with the same parameters.

Clicking on one of the Analysis buttons will open a new window, DetectClusters, where you can choose the strategy to select candidate clusters (Pre-selection), enter parameters for pre-selection and selection of clusters, and chose to perform the detection of subdomains.

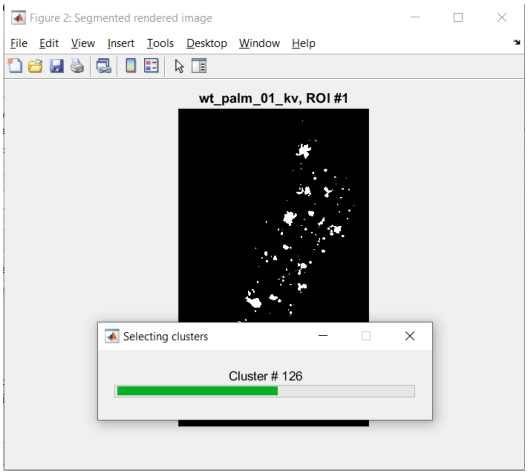


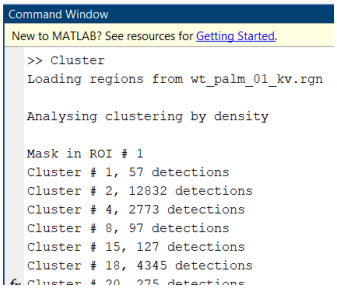
**Diinamic-R** is the option by default. It uses the rendered image to create a mask with the **Intensity threshold**. You may improve the mask using the morphological operators **Dilate** and **Erode** (enter the number of pixels to add or remove). If the density of detections is high enough (at least one detection per SMLM pixel) you may set the threshold for a minimum density per pixel (**Min dens pixel**). This option helps the pre-selection in presence of multiple detections per fluorophore. However, if your detections density is low, set it to zero.

The selection of clusters is done considering:

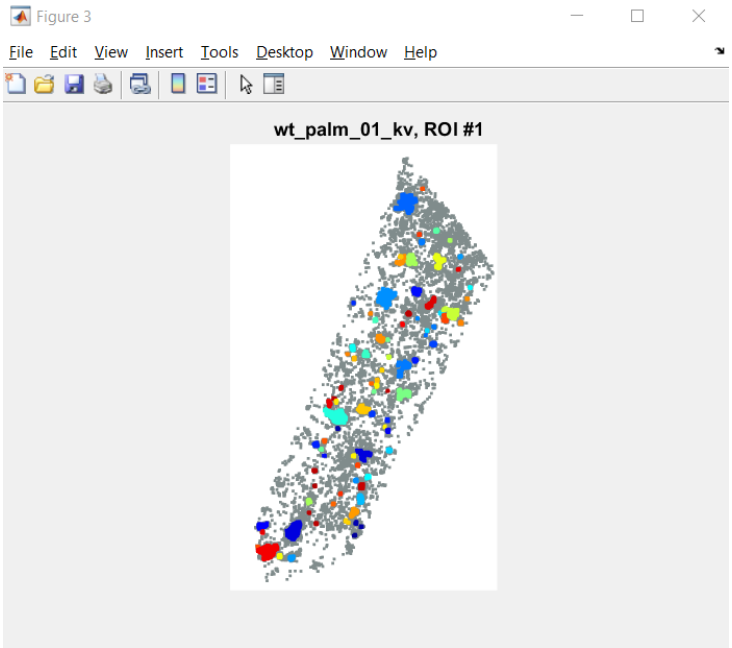
- Minimum density (**Min density**) per pixel (SMLM pixel),
- Minimum number of detections in the cluster (**Min # detec**),
- A minimum (**Min size** or **Min grid units**) and a maximum (**Max size** or **Max grid units**) cluster size (number of pixels).

**Try** will launch the analysis on each ROI of the file. The rendered mask is shown on an independent figure, and you can follow the progression of the analysis on the Command window. The wait bar also shows the progression. Close it to abort the analysis. If you want to stop the loop over a list of files, close the window or use the **Abort** button. Please note that partial results are not saved.

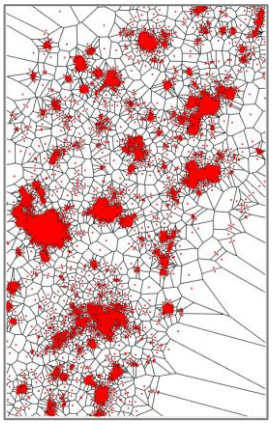



A new figure shows the selection results for the ROI, with the detections belonging to clusters appearing in color (each cluster has a different color) and those non-clustered detections appearing in gray.



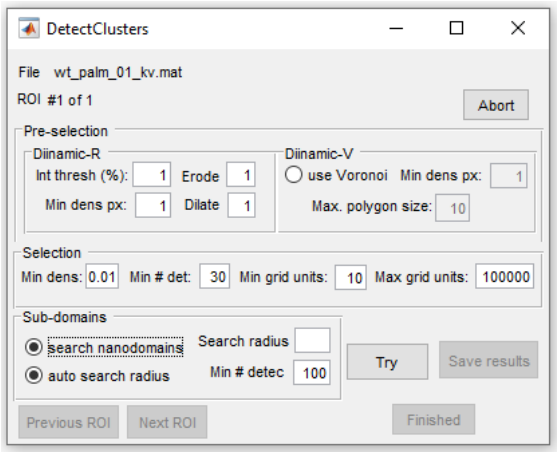
**DiinamicV** uses Voronoi tessellation to create the mask. Select **use Voronoi** if you want to use this strategy, and enter the **Max. polygon size** (arbitrary units) to create the mask. The result will include detections whose polygons are smaller than this threshold. An image with the Voronoi tessellation is displayed. The selection of clusters is done as for DiinamicR.



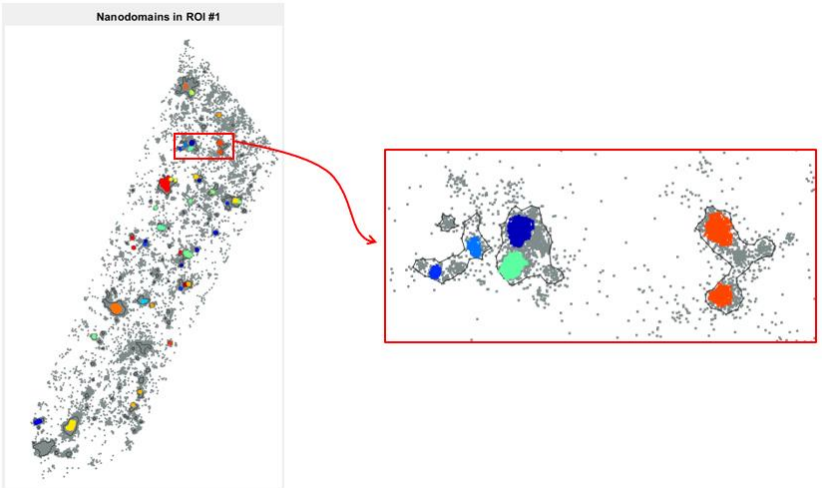
Example of Voronoi tessellation on SMLM data

## Subdomains inside clusters

The detection of sub-domains on the selected clusters is done if you click on **search nanodomains**.




This part uses a DBSCAN algorithm: it needs a **search radius** and the minimum number of detections in the subdomain (**Min # detect**). The calculation of the search radius can be done automatically (**auto search radius**), based on the mean distance between detections in the sample.



Clusters that were detected previously are shown by a grey contour line that depicts their border. Nanoclusters appear in color. Note that detections themselves are not represented in scale so their size change if you zoom in or zoom out. The number of nanoclusters (and clusters) found are shown on DetectClusters window.

## Results

If you agree with the results, click on **Save results** to keep them. Two files will be saved: *name*-roidata_*number of the ROI*.mat and *name*-selection.tif.

If there are more than one ROI on the data file, use **Next ROI** to move to the next ROI and click on **Try** to launch the detection on this new ROI.

When all the ROIs were analyzed, click on **Finished**. A new results file will be generated: *name*-allclust.txt. If you also made the detection of subdomains, these results will be saved in another file called *name*-nanoclust.txt. In addition to the results, there will be a text file that recapitulates the analysis that was done (Parameters_clustering_*date of analysis*.txt):

Example after analysis with subdomain detection:

```
Parameters for clustering analysis

Analysis of density after segmentation with rendered image. Intensity threshold: 1%. Density threshold: 1
Segmented image improved with dilation: 1 px and erosion: 1 px.

Cluster selection criteria:
Minimum density: 0.01. Minimum number of detections: 30
Minimum cluster size (pixels): 10. Maximum cluster size (pixels): 100000

Including nanodomain detection by DBSCAN. Minimum number of detections in the nanodomain : 100
Search radius calculated automatically : 2.5
              Files:
    wt_01_647_SR.mat
```

### Format of results

Files *name*-allclust.txt: groups the results of all the ROI of a data file.

| ROI # | cluster # | number of detections | detections in ROI | cluster area | cluster density |
|---|---|---|---|---|---|
| 1.0000000e+00 | 5.0000000e+00 | 6.8000000e+01 | 3.7999000e+04 | 6.2571744e-03 | 1.0867525e+04 |
| 1.0000000e+00 | 6.0000000e+00 | 8.3000000e+01 | 3.7999000e+04 | 8.1745284e-03 | 1.0153491e+04 |
| 1.0000000e+00 | 1.5000000e+01 | 3.2000000e+01 | 3.7999000e+04 | 2.0911326e-03 | 1.5302712e+04 |

Files *name*-nanoclust.txt: groups the results of all the ROI of a data file.

| ROI # | cluster # | nanodomain # | number of detections in the nanodomain | number of detections in the cluster | nanodomain area | nanodomain density |
|---|---|---|---|---|---|---|

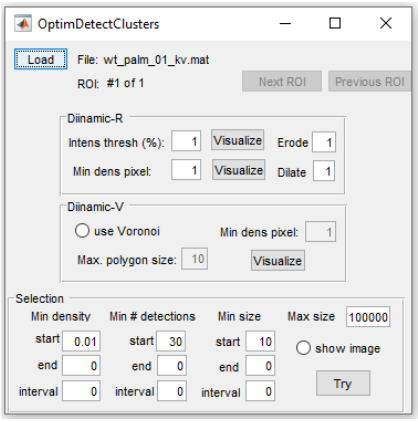
Files *name*-selection.tif: Image of the selected clusters (each cluster appears in a different color).

## Choosing and optimizing Diinamic parameters

This GUI helps choosing the parameters by executing iteratively the selection of clusters and by allowing the visualization of the pre-selection and selection steps. See the chapter below (Tips for choosing parameters) for a guide about the effect of each parameter.

**Load** the data file (.mat). If you had selected more than one ROIs, you can navigate between them with **Next ROI** and **Previous ROI**. The parameters are the same than those described above.

Use the buttons **Visualize** to show the result of the intensity thresholding (Intens thresh (%)) or the thresholding by the density of detections per pixel (Min dens pixel, for Diinamic-R or Diinamic-V).




In case of the selection parameters, you can try different combinations of minimum cluster density (**Min density**), minimum number of detections per cluster (**min # detections**) and minimum size of the cluster (**Min size**). As a start point, consider what you expect to be the minimum size or density of a cluster, depending on how you define it (number of molecules or size). The algorithm will retain all the clusters that are larger and or denser than these parameters, up to the maximum size (**Max size**).

For these parameters, if you specify a **start** value and **end** value, the algorithm will step up the start value up to the end one, given the **interval** that you enter. If the end value is zero, the algorithm will use only one value (the start one).

Click on **Try** to start the series. If you want to see the result, click on **show image**. **Please note that the images will not disappear automatically,** so if you choose to loop over hundreds of possible combinations of parameters, MATLAB will end up with hundreds of figures open on the screen!

You can follow the progression on the command window of MATALB. At the end, a result file summarizes the three selection parameters and the number of clusters obtained.

*name*-optimDiinamic.txt:

| Min density | Min # detections | Min cluster size | # clusters found |
|---|---|---|---|
| 1.0000000e+00 | 5.0000000e+01 | 7.0685835e+00 | 2.6000000e+01 |
| 1.0000000e+00 | 5.0000000e+01 | 1.9634954e+01 | 2.6000000e+01 |
| 1.0000000e+00 | 1.0000000e+02 | 3.8484510e+01 | 1.6000000e+01 |
| 1.0000000e+00 | 1.5000000e+02 | 7.0685835e+00 | 1.4000000e+01 |
| 1.0000000e+00 | 1.5000000e+02 | 1.9634954e+01 | 1.4000000e+01 |
| 1.0000000e+00 | 1.5000000e+02 | 3.8484510e+01 | 1.4000000e+01 |
| 2.0000000e+00 | 5.0000000e+01 | 7.0685835e+00 | 2.6000000e+01 |
| 2.0000000e+00 | 5.0000000e+01 | 1.9634954e+01 | 2.6000000e+01 |
| 2.0000000e+00 | 5.0000000e+01 | 3.8484510e+01 | 1.9000000e+01 |
| 2.0000000e+00 | 1.0000000e+02 | 7.0685835e+00 | 1.6000000e+01 |

# Tips for choosing parameters

Here we provide some cues about the preselection parameters of Diinamic-R and the selection parameters. Please refer to the article describing the algorithms to learn more about Diinamic-R and Diinamic-V and their applications (Paupiah et al. 2023). The following examples were extracted from the testing Scenarios proposed by Nieves and coll. (2023) and examples of data obtained in our laboratory.

As a rule of thumb:

- If the density of detections is low, rely on the pre-selection by intensity segmentation (Diinamic-R).
- However, if the density of non-clustered detections is variable, Diinamic-V could be a better choice (see the examples below). Alternatively, draw ROIs in a way that the non-clustered detections of the ROI are distributed more homogeneously.
- If multiple blinking is present, try the minimum pixel density threshold to reduce the size of clusters and use the minimum cluster size threshold to eliminate false clusters arising from single fluorophores.
- If you know the expected characteristics of the clusters, use them to calculate the selection parameters (minimum allowed values).
- If the ratio of detections densities in and out clusters is low, try the detection of subdomains within clusters to improve the results.

## Pre-selection of clusters with Diinamic-R

Besides eliminating non-relevant data, the use of a segmentation mask sharpens the borders of the clusters. This sharpening effect is enhanced by the pixel minimum density threshold, that can be useful in case of high density of detections.

**Intensity threshold**: The intensity threshold for segmentation is calculated as a percentage of the maximum value observed in the image. If you want to keep all the detections for the selection step, just set this threshold to zero. This parameter is particularly useful in case of low density of detections, situation in which density thresholds could fail.

**Minimum density in pixel**: pre-selection parameter that helps reducing the bias induced by multiple blinking of fluorophores. If a false cluster comes from the cloud of detections generated by a single fluorophore, this threshold will reduce the size of this false cluster so it will be more easily discarded during the selection step (by the cluster size threshold). Note that it must be zero in case of low density of detections.

## Pre-selection of clusters with Diinamic-V

The only parameter related to Voronoi tessellation is the size of Voronoi polygons: the detections that are kept are those associated to polygons that are smaller than the threshold. As there is only one parameter, this pre-selection method is less flexible. Please note that the computational burden is more important than for Diinamic-R. We propose this alternative in case of non-homogeneous distribution of non-clustered detections (particularly in presence of multiple blinking) because Diinamic-R is less performant in this case. Therefore, we privilege Diinamic-R (faster) unless it fails.

**Minimum density in pixel**: same principle than for Diinamic-R (see above).

## Selection of clusters

On the basis of the rendered image (SMLM pixel size), pixels bearing at least one detection (or at least the required minimum density by pixel) are retained to create candidate clusters by coalescence. These candidate clusters than go through three possible tests to decide to retain them or not:

- The size of the clusters corresponds to their number of pixels (grid units),
- The number of detections is the number of detections colocalized with the pixels in the cluster
- The detection density in the cluster is the number of detections per area unit (surface of the pixel).

These three criteria are correlated. However, depending on the experimental design and on the knowledge about the molecules that are investigated, it may be easier to set one of these parameters among the others. If you cannot or do not want to use one or more of these parameters, just put a very low value to skip it. Indeed, if the candidate clusters are already those that you want to keep, you can skip all of them!

**Minimum size of clusters**: probably the more straightforward selection threshold. In many cases, it is enough to get adequate results. However, it may not be satisfactory if clusters have very variable sizes. Nevertheless, it is useful to reduce the impact of multiple blinking: set its value (at least) to discard false clusters generated by the blinking of one fluorophore (these clusters should have a size of 1-2 times the localization precision).

**Minimum density in clusters**: prefer this threshold to use when clusters have different sizes.

**Minimum number of detections**: this is a threshold to use when you can guess how many fluorophores you should have in a cluster: typically, in PALM acquisitions and when the clustering behavior of the molecule under study is well known. If the expected number of molecules in a cluster is unknown, set this threshold to a low value. Nevertheless, it can be useful to reduce the impact of multiple blinking: increase its value to discard false clusters generated by the blinking of one fluorophore.

## Detection of non-homogeneous regions within clusters

This step, optional, is executed after selecting the clusters. DBSCAN is applied separately on each cluster, to look for sub-domains with higher density inside the cluster. The parameters are those of DBSCAN: a search radius and a minimum number of detections per cluster. If the density of detections is variable among clusters, use the option “auto epsilon”. The search radius is then calculated automatically based on the density inside the cluster.

## Examples of application

The following possibilities are shown:

1: Comparison of simulated data (low density) with and without multiple detections.

2: Simulated data with strong heterogeneity of non-clustered detections.

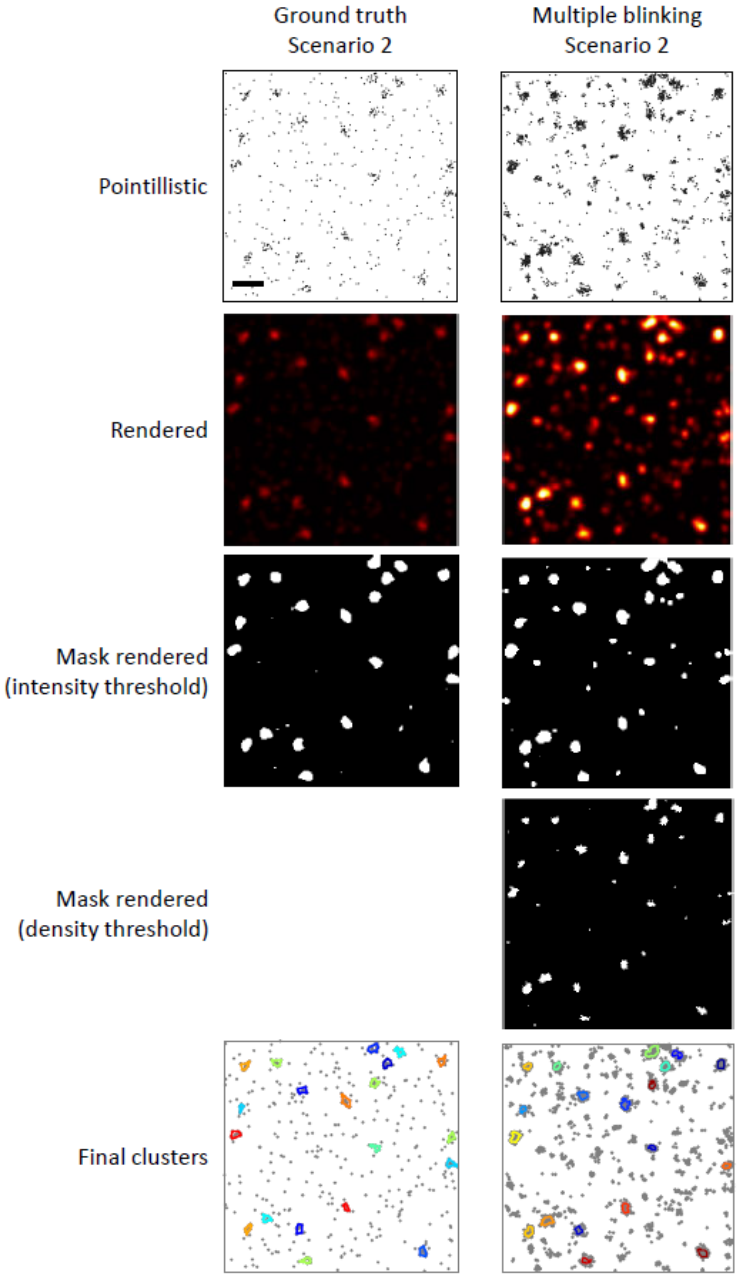
3: PALM data with clusters of regular size.

4: PALM data with clusters with variable density (inside clusters and within clusters) and size

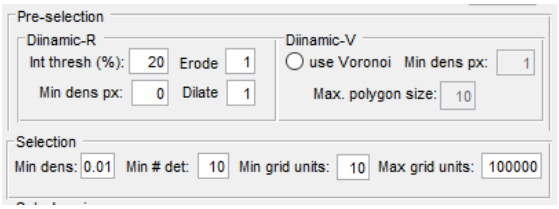
5: Comparison of PALM and STORM data of the same molecule.

**Example 1:**

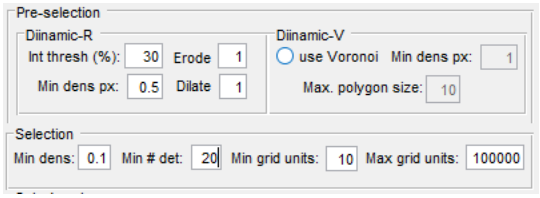
Images taken from the Scenario 2 proposed by Nieves and coll (2023). Simulations of clusters of 15 molecules, surrounded by non-clustered detections at low density (in average, less than one detection per SMLM pixel), with (Multiple Blinking) or without (Ground truth) simulation of multiple detections. Scale bar: 250nm).




No multiple detections ("ground truth"): The best result here was obtained by relaying mainly on the intensity threshold for the pre-selection step (Diinamic-R).

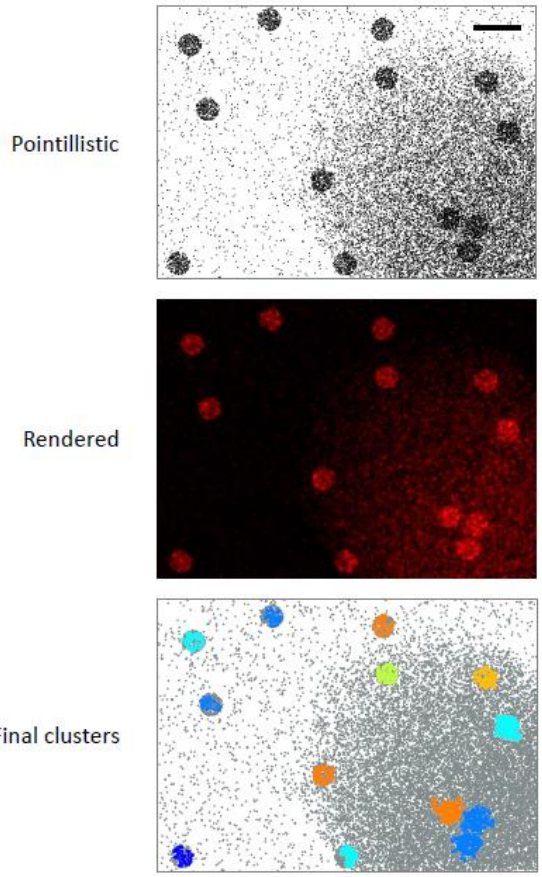


Multiple blinking : the best result (no false cluster detected, only one real cluster missed) was found by including the Minimum density pixel threshold (Diinamic-R). Note the small white regions in the mask: they correspond to false clusters arising from multiple detections of one molecule. The selection parameters were chosen to exclude these false clusters, based on their expected size and density. Note that the minimum number of detections allowed per cluster is higher than the real one (to account for multiple detections).

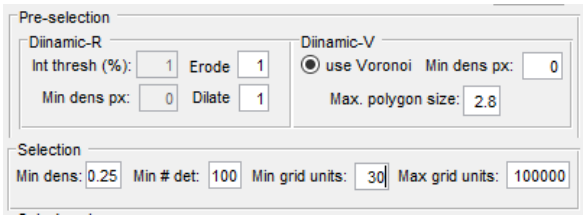



**Example 2:**

Simulated medium density clusters of regular size (500 nm in diameter), surrounded by non-clustered detections that are distributed non-homogeneously (scale bar: 1µm).

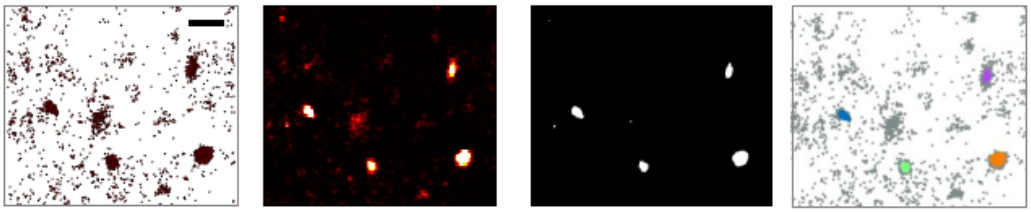



In this case, the intensity threshold was impossible to find to correctly pre-select all the clusters. Indeed, the best result was obtained using Diinamic-V for the preselection (the intensity threshold here is inoperant).



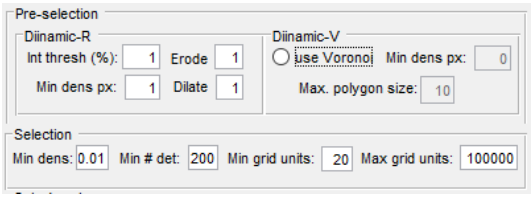
**Example 3:**

PALM image of Gephyrin in cultured neurons. This protein, present at inhibitory synapses in neurons, is localized in round or ovoid clusters (typically the longest axis length is 200-300nm). In this image, clusters are well distinguishable from the non-clustered detections. Scale bar: 1 µm).



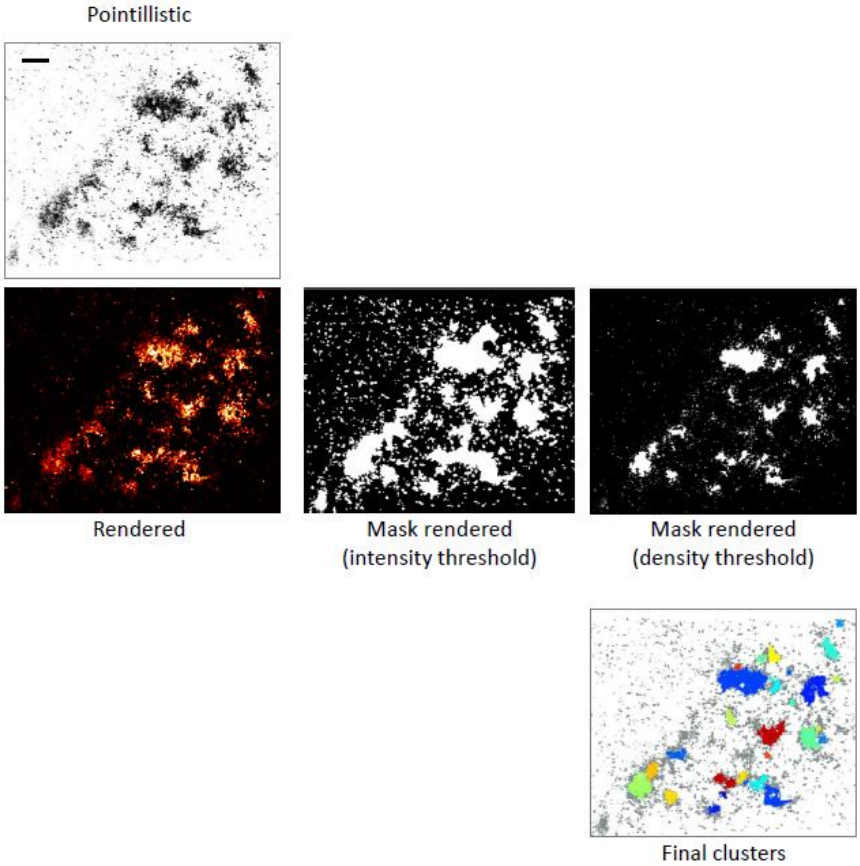
Pointillistic | Rendered | Mask rendered (intensity threshold) | Final clusters

Increasing the intensity threshold in this example caused a significant reduction of the size of clusters (the density of detections is not very high inside clusters). Then the retained combination was a low intensity threshold together with a density threshold in the preselection. The values used in the selection were chosen on the basis of the characteristics already described of Gephyrin clusters.

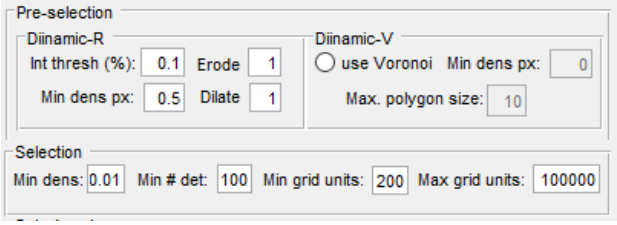



**Example 4:**

PALM imaging of the potassium channel Kv2.1 in neurons. It forms clusters of different size and density. Multiple blinking was not corrected. Scale bar: 2 µm.



To avoid splitting large clusters in many small ones (the distribution of detection in large cluster is not homogeneous), we chose a low intensity threshold but we included a minimum density per pixel threshold. Then we set the selection parameters (minimum number of detections and minimum size of clusters) to reject false clusters due to multiple blinking.



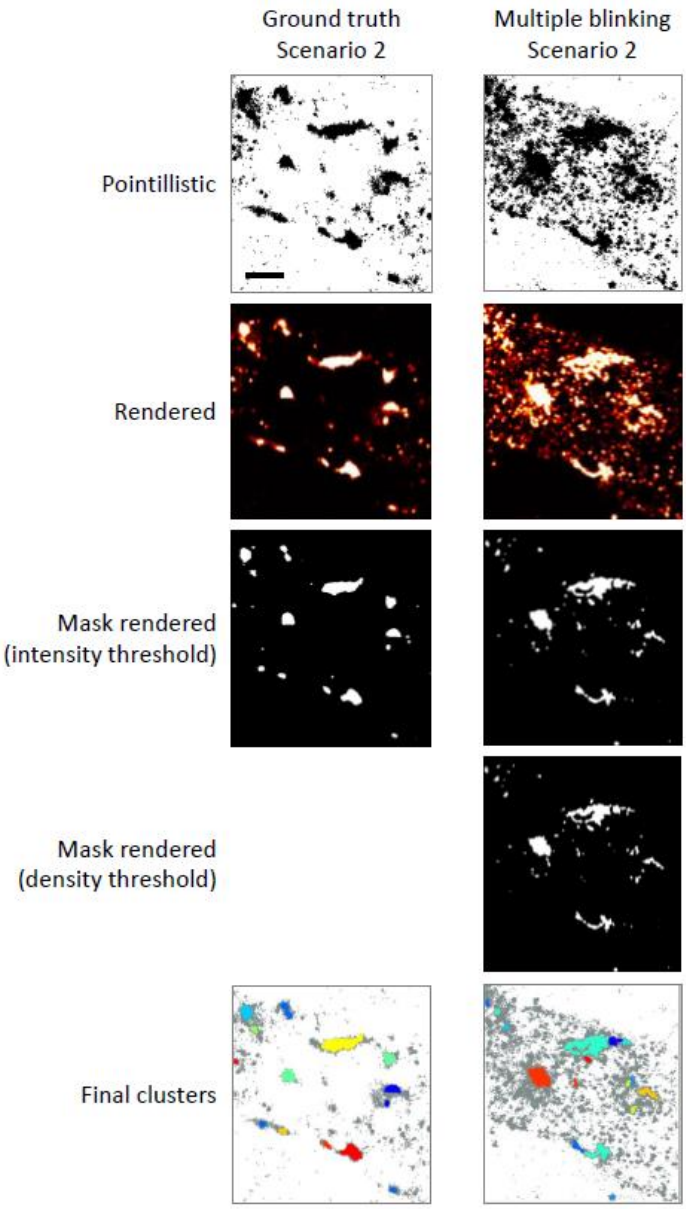

**Example 5:**

PALM and STORM acquisitions of the same molecule, made on the same ROI (Dendra2-tagged Kv2.1, observable by PALM; and immunodetection of Dendra2 with an Alexa647-tagged antibody, observable by STORM). Both techniques introduce a bias in the detection of molecules but in opposite directions:

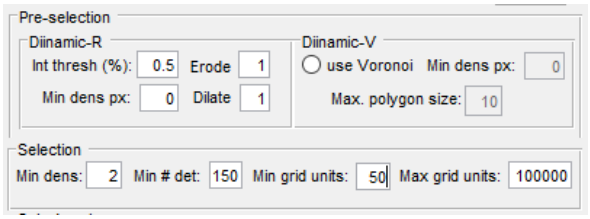
PALM may under-sample due to the maturation fluorescent proteins (~40% of Dendra are not fluorescent, and not all the subunits of the channel are tagged), whereas STORM over-samples (multiple blinking, unknown number of fluorophores per channel). In addition to this, the distance between the fluorophore and the molecule of interest is longer in STORM due to the size of antibodies). Therefore, their results are intrinsically different (see the figure below, scale bar: 2 µm). Visually, some clusters of PALM detections overlap as expected with those of STORM detections, some others were present only in one set of data.

In the example, we considered the following facts to decide which parameter to use:

- Due to the different density of detections between both datasets, the most important difference between the parameters for PALM and STORM data was the intensity threshold to segment the rendered image (PALM: 0.5; STORM: 15) and the use of a minimum density per pixel (2) in case of STORM data.
- Kv2.1 is a tetrameric potassium channel known to form large clusters in neurons (up to 4 µm). The expected size of the molecule is ~7nm. We could not know how many chimeric subunits were present per cluster (tetramers could contain both endogenous and chimeric subunits), so we use a low threshold for the number of detections (150).
- Data were not corrected for multiple blinking, and STORM data showed a multitude of small clusters that were not observable in PALM data. We considered them as being mainly false clusters arising from multiple detections of the same molecule (i.e. multiple fluorophores/antibodies on the same molecule). Moreover, some of the clusters observed with STORM seemed to be larger than in PALM data. We chose the minimum size of cluster threshold just to eliminate false clusters due to multiple detections, and we set different thresholds for PALM and STORM to consider the possibility of enlargement of clusters due to the use of antibodies. We calculated a minimum size for clusters of 50 grid units for PALM and 70 grid units for STORM.




Analysis of PALM data:




Analysis of STORM data: