Manual for the Grafeo Software

Version v1beta

Kalina Tamara Haas

UNIVERSITY OF CAMBRIDGE
School of Clinical Medicine

MRC Cancer Unit

Table of content

1. Glossary

VD – Voronoi diagram

VP – Voronoi polygon

DT – Delaunay Triangulation

SMLM – Single Molecule Localization Microscopy

STORM – STchastic Optical Reconstruction Microscopy

Graph – mathematical representation of a network of points (nodes) connected by lines (edges)

1C – one color

2C – two color

2D – two-dimensional

3D – three-dimensional

2. Table of Figures

3. Introduction

Grafeo is Matlab based program for the analysis of a single molecule localization microscopy (SMLM) data e.g. dSTORM or PALM. It works on a single molecule coordinates rather than pixelated images. Grafeo uses various pointillist approaches previously described in the literature, e.g. Voronoi Tessellation, Delaunay Triangulation, Ripley's function. Grafeo does not pretend to be the first to apply these functions for the single molecule data. It was developed to cope with huge amount of data to work in the batch processing mode (Haas et al., 2018). Grafeo v1beta and its manual are far from being ideal. All the errors and inconsistencies will be corrected in the future release. More functionalities will be added in the future release. If you have any questions please contact the author via email: inakuflers@wp.pl or Twitter @KalinaHaas.

To run Grafeo you need Matlab with the following toolboxes:

`Curve Fitting Toolbox, Image Processing Toolbox, Statistics and Machine Learning Toolbox, Signal Processing Toolbox`

Grafeo uses following third party function: InPolygon, created by Guillaume Jaquenot. You can download it from here:

https://uk.mathworks.com/matlabcentral/fileexchange/20754-fast-inpolygon-detection-mex. If you cannot download it, Grafeo will automatically switch to the Matlab `inpolygon` function, however InPolygon is few time faster than its Matlab counterpart.

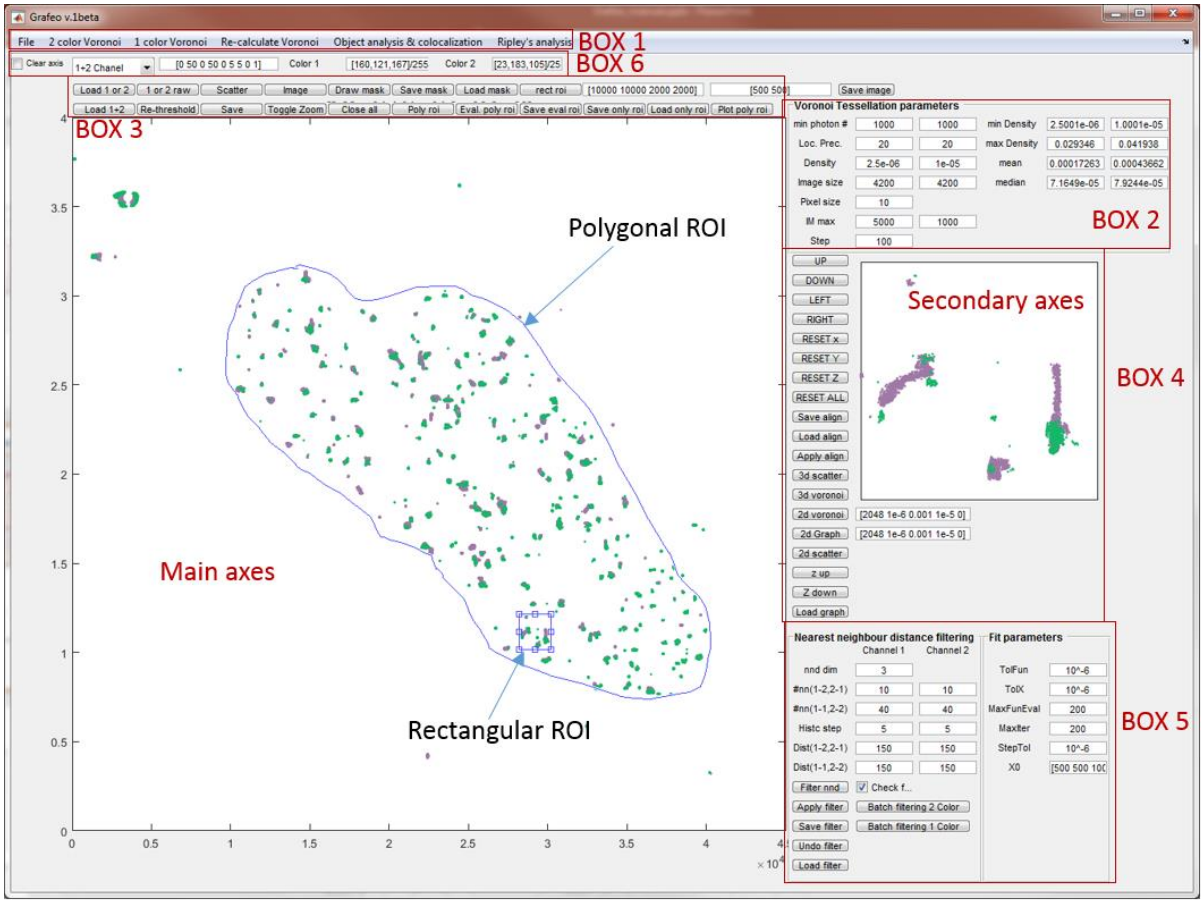
First, download Grafeo from https://github.com/inatamara/Grafeo-dSTORM-analysis- . Add all the functions and scripts from GrafeoSubfunctions folder to your Matlab working path (`addpath`(FullPathToGrafeoFolder)), then type in the Matlab command line run('Grafeo'). This brief manual does not describe the details of Voronoi Tessellation, Delaunay Triangulation and spatial statistics functions. To learn more about these functions, please refer to the following publications

and citations therein (Levet et al., 2015, Andronov et al., 2016, Haas et al., 2018, Lee and Schachter, 1980, Dixon, 2002).

You can download the example data here: from https://github.com/inatamara/Grafeo-dSTORM-analysis-. These data was used for Haas et al., (Haas et al., 2018) but here are analysed with different parameters. The folder named zeiss contains the Test data from: https://github.com/PRNicovich/ClusDoC, which works as the Zeiss STORM file format sample.

To download the sample data folder, please go to:

https://drive.google.com/drive/folders/12x9_x4LKqZ-RZuEZ0ufcO9ayR8mtWcL-?usp=sharing



BOX 1 – Single and batch mode processing, importing molecular list, Voronoi Diagram thresholding, two channel alignment, object and co-localization analyses

BOX 2 – Display and setting Voronoi diagram properties

BOX 3 – Loading the data, creating ROI, displaying scatter plot and an image

BOX 4 – Manual alignment of two channels, plotting rectangular ROI as Voronoi Polygons DP, Scatter plots and Delaunay Triangulation (DT) based graphs

BOX 5 – Stettin the parameters for nearest neighbour distance function data filtering and optimization parameters

BOX 6 – Slecting the channel to display, setting colors for a plot, controlling super-resolved image disply

Main axes – axes where all the data is displayed

Secondary axes – axes for rectangular ROI display

Figure 1. Grafeo GUI window

4. Brief description of main sections

The Grafeo GUI is displayed in **Figure 1**. It contains following main functionalities as described below.

BOX 1 - This section allows the user for single and batch mode file processing of two dimensional (2D) or three dimensional (3D), one color (1C) or two colour (2C) STORM single molecule localization data or similar (e.g. PALM). Grafeo was developed for Nikon 2D/3D N-STORM txt data format, but should work with other txt molecular list files. User starts with importing a molecular list in txt format. Relevant data are copied to Matlab variables and Voronoi Diagram (VD) is computed on 2D or 3D coordinate list for each channel separately (1C or 2C mode possible). Grafeo allows thresholding the data based on the localization precision, photon count and VD density (an inverse of Voronoi Polygon VP size). For 2C data, Grafeo attempts to align two channels (Yet, it is not very robust). Additional filtering step based on univariate and bivariate nearest neighbour distance function is described in BOX 5 (`Nearest neighbour distance filtering panel`). After VD based thresholding, Delaunay Triangulation (DT) is computed and converted into Matlab graph data type (see Matlab `graph` function documentation). DT diagram is segmented into discreet subgraphs/clusters by thresholding the DT edge length. Co-localization of 2C data can be evaluated using spatial statistics functions like Ripley's or nearest neighbour function calculated on the centroids of clusters or molecular coordinates.

BOX 2 - After loading Matlab file containing VD data edit text boxes are populated (BOX 2, `Voronoi Tesselation parameters panel`), first column corresponds to channel one and second column to channel two. These values are used at different steps of processing. `Min photon #` – lower threshold for the number of photons gathered for a localization/molecule, `Loc. Prec.` – upper threshold for localization precision, `Density` – upper threshold for VD density (an inverse of the maximum VP size), `min` (`max`, `mean`, `median`) Density – minimum (maximum, mean, median) VD density. `Pixel size` is used only for image binning, default is in nm. `Image size` – the number of pixels in an image. `Step` is used for manual channel alignment (see BOX 4).

BOX 3 - This part allows to load and display individual 1C or 2C VD data files. The VD threshold can be readjusted using values (only `Density`, `min photon #` and `Loc. Prec.`) from the BOX 2. Rectangular or polygonal regions of interests ROI can be created and saved.

BOX 4 - This section allows for manual alignment of two channels and visualization of the rectangular ROI as 1C/2C and 2D/3D scatter plot, VD and DT graphs.

BOX 5 - Filtering of the data based on univariate and bivariate nearest neighbour distance thresholding.

BOX 6 - Setting the colour of the plot for channels 1 and 2 and setting the properties for an image display.

5. Naming of the molecular list file

If you work in a single file processing mode, the name of the file is not relevant, since you will be always ask to select a file manually. Working with a batch processing mode, especially with 2C data,

requires specific file name format as follows: channel tag + delimiter + number + .txt, e.g. `647_1.txt`, `GFP_1.txt`, `first_2.txt`, `second_2.txt`. Best practise is to keep always '_' as a delimiter and to avoid using a number in a channel tag e.g. `second_2.txt` instead `2_2.txt`. Use the same channel tags for all your data to avoid multiple batch processing instances.

Set the working folder go to `File` -> `Set path` and select the working folder.

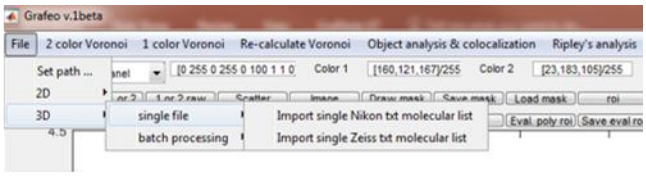


Figure 2. Importing molecular list

6. Importing molecular list

Grafeo program was developed for Nikon N STORM molecular lists in txt format. Additional functionality was added for Zeiss txt format based on sample file downloaded from the internet. Other txt files may be compatible with Grafeo by setting appropriate parameters described below.

6.1. Single file importation

To import single txt molecular list file go to `File` ->, `select 2D or 3D option` -> `single file` -> `import Nikon/Zeiss molecular list`, **Figure 2.**

You will be prompted to select a txt file. Then a dialog box will appear and you will need to enter the following parameters from top to bottom (use default parameters for Nikon and Zeiss), **Figure 3**:

Number of columns in a molecular list files, X coordinate, Y coordinate, photon count, localization precision, frame number in which the molecule appears. The column with channel tag; in Nikon it is a string e.g. 647, 568 in Zeiss it is a number e.g. 1, 2. The trace length column, the trace length it is a number of frames the molecule was detected. The Z coordinate column. The Z rejected tag, in Nikon

it is a string: e.g. Z rejected. If you want to keep Z rejected localization in further analyses, enter 0 in the next line. To reduce the size of a file, already at the stage of importation the molecular list is filtered based on photon count and localization precision. In the following two lines enter the minimum number of detected photons and the maximum localization precision. Next, enter the number of header lines in a molecular list file, for Nikon and Zeiss enter 1. Lastly, space delimiter in a molecular list file, '\t' for Nikon and Zeiss. Then press ok. The header line will display in the Matlab command line, for Nikon it reads as follows:

Channel Name X Y Xc Yc Height Area Width Phi Ax BG I
Frame Length Link Valid Z Zc Photons Lateral Localization
Accuracy Xw Yw Xwc Ywc.

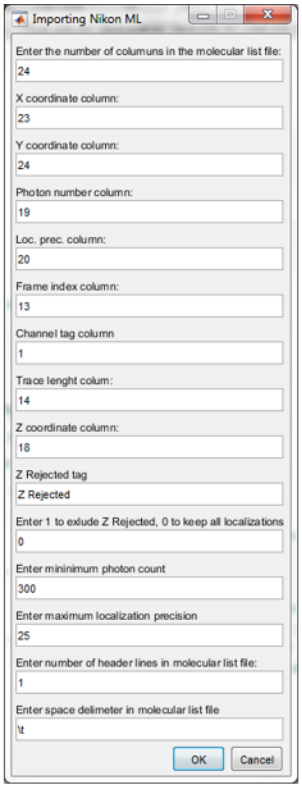


Figure 3. Setting the parameters for the importation of the Nikon molecular list

Molecular list is imported, data is filtered based on photon count and localization precision after which Voronoi diagram is calculated. After processing is completed, file is automatically saved and a full path is displayed in Matlab command line followed by the time it takes to process a give file. `Example data` folder contains the following Nikon 3D N STORM molecular list txt files: `647_2.txt` and `568_2.txt`, which will be saved as:

`647_2_Xwc_Ywc_I_F_Sig_Zc.mat` and `568_2_Xwc_Ywc_I_F_Sig_Zc.mat.` Each file will contain following variable:

`dim` – dimension of the VD

`photonTH` – photon number threshold

`locprec` – localization precission threshold

`numfr` – number of image frames in a data,

`v` - Voronoi vertices

`c` - Voronoi cells of the Voronoi diagram of variable `X`, as returned by Matlab function voronoin: `[v,c] = voronoin(X)`

`Vp` = v(c) – vertices of the VP (Voronoi Polygons)

`VOLp` – volume of each VP (in nm)

`datatesselerfilt` – 8 column data with following variables: X coordinate, X coordinate, photon count, frame number, localization precision, Z position, trace length, 0/1 Z rejected/or not. Attention, column order will be changed in the consecutive steps of processing.

Number of rows in `datatesselerfilt`, `Vp`, `VOLp` and `c` must be the same and equal to filtered number of localized molecules *Nf*. `VOLp` is a column vector with length *Nf*; `c` and `Vp` are a cell arrays with size Nfx1, `datatesselerfilt` has a size Nfx8.

6.2. How to import a custom file

Only X, Y (and Z for 3D STORM) data columns are strictly necessary for Grafeo to work. If your file does not have eight different variables described above, you can pass indices to any other columns and set photon and/or localization precision threshold to 0.

6.3. Batch mode file importation

`File` -> `2D/ 3D` -> `batch processing` -> `import Nikon/Zeiss txt molecular lists`. First, you will be prompted to select a root folder. All sub-folders and sub-sub-folders within

the root folder will be analysed. Only two level of folder processing is available. Example root folder structure looks as follows: root folder – mySTORMdata; subfolders – day1, day2, day3, … day100; sub-sub-folders – treatment1, treatment2, … treatment10.

`Example data` folder contains root folder – `GrafeoSampleData` with 2 subfolders, `EUFA423BRCA2_5h` and `EUFA423_5H` and no sub-sub-folders (no different treatment).

After processing `example data` folder in batch processing mode the following information appears in the Matlab command line:

Sub folder #1 = EUFA423BRCA2_5h

Sub folder #2 = EUFA423_5h

No Sub folder #1 = EUFA423BRCA2_5h

Channel Name X Y Xc Yc Height Area Width Phi Ax BG I
Frame Length Link Valid Z Zc Photons Lateral Localization
Accuracy Xw Yw Xwc Ywc

G: \grafeo sample data\EUFA423BRCA2_5h\568_2_Xwc_Ywc_I_F_Sig_Zc.mat

Channel Name X Y Xc Yc Height Area Width Phi Ax BG I
Frame Length Link Valid Z Zc Photons Lateral Localization
Accuracy Xw Yw Xwc Ywc

G: \grafeo sample data\EUFA423BRCA2_5h\647_2_Xwc_Ywc_I_F_Sig_Zc.mat

Elapsed time is 137.806193 seconds.

No Sub folder #2 = EUFA423_5h

Channel Name X Y Xc Yc Height Area Width Phi Ax BG I
Frame Length Link Valid Z Zc Photons Lateral Localization
Accuracy Xw Yw Xwc Ywc

G: \grafeo sample data\EUFA423_5h\568_1_Xwc_Ywc_I_F_Sig_Zc.mat

Channel Name X Y Xc Yc Height Area Width Phi Ax BG I
Frame Length Link Valid Z Zc Photons Lateral Localization
Accuracy Xw Yw Xwc Ywc

G: \grafeo sample data\EUFA423_5h\647_1_Xwc_Ywc_I_F_Sig_Zc.mat

Elapsed time is 251.106677 seconds.

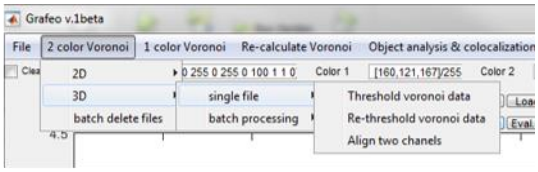


Figure 4. Thresholding VD

7. Two colour voronoi data processing

7.1. Single file two color voronoi data processing

After data importation is completed, two files corresponding to two channels are combined within one file. At this stage VD is thresholded using one sided threshold value for the VD density (an inverse of VP size). All VP with density (size) smaller (larger) than a threshold values will be removed.

Go to `2 color Voronoi` -> `2D/3D` -> `Single file` -> `Threshold voronoi data` and press ok, **Figure 4.** You will be asked to select a file for channel one and then for channel two. Next, dialog box will appear and you will have to populate following lines, **Figure 5**:

`delimiter` – a delimiter used in previous step (preferably '_')

`flag for channel 1` – '647' in above example

`flag for channel 1` – '568' in above example

`Enter 1 for mean, 0 for median density` – VD can be automatically thresholded using mean or median value of VP density. This variable is a two element vector, e.g. [1,1] mean will be used for channels 1 and 2, [0,1] median will be used for channel 1 and mean for channel 2 etc.

`Enter multiplication factor for mean/median density` – this additional parameter allows you to offset mean/median density threshold by a multiplication factor e.g. [0.01,0.001], means that a threshold for a VP density will be mean/median (depending on previous choice) x 0.01 for channel 1 and 0.001 for channel 2. You can also set and arbitrary density threshold for each channel by inserting 1 under `Enter 1 to use arbitrary density threshold.` The `Density` threshold values will be taken from main GUI window (highlighted in BOX 2, `Voronoi Tessellation parameters` panel).

If you want to try to align two channels, insert 1 under `Enter 1 if you want to align two channels`.

`The last` line, `Enter name flag for 2 color file saving`
allows you to enter name tag for two colour combined file, the default is `Aligned_`.

The following variables will be saved (here 1 and 2 corresponds to channel tag):

`dataout1/2` – is a matrix with single molecule data. The size is N1/N2 x 8, whre N1/2 is number of points in respective channel. The column content is as follows: x coordinate, y coordinate, photon count, localization precision, z coordinate, frame number, a trace length – the number of frames given molecule was localized and z rejection tag (0 was marked as z rejected).

`dataoutshift1` – is a the same as dataout1 but align to the chanel 2.

`dim` – dimension of the data, 2 or 3

`photonTH1/2` – photon number lower threshold

`locprec1/2` – localization precision upper threshold

`densitythr1/2` – VD density upper threshold

`exitflag` - describe exit condition of `fminunc` solver (e.g. equal 1 if global minimum is found), empty if no alignment

`output` - structure with the information about optimization process, empty if no alignment

`zcount` – z direction step for the alignment of channel 1 with respect to channel 2, empty if not align or for 2D data

`updowncount` – x direction step for the alignment of channel 1 with respect to channel 2, empty if not align

`leftrightcount` – y direction step for the alignment of channel 1 with respect to channel 2, empty if not align

`Vp1/2` – vertices of the VP (Voronoi Polygons)

`Dp1/2` – vector of voronoi density

`namech1/2` – name of the file analysed, used for the re-thresholding of VD

If you chose to align two channels, a figure displaying each optimisation step will appear. To learn more about the optimisation procedure, please refer to Single File automatic alignment after voronoi data thresholding. At the end of the optimisation step Matlab command line will be populated with the summary of the optimisation step:

Iteration	Func-count	f(x)	First-order Step-size	optimality
0	4	82059.6		43.3
1	16	78135.1	2.10352	24.6
2	20	76081.7	1	1.98
3	24	76065.1	1	0.268
4	28	76064.8	1	0.0454
5	32	76064.8	1	0.00935
6	36	76064.8	1	0.00144
7	40	76064.8	1	0.00112
8	44	76064.8	1	5.91e-05
9	48	76064.8	1	9.85e-06

Optimization completed: The first-order optimality measure, 2.225870e-07, is less than options.OptimalityTolerance = 1.000000e-06.

Optimization Metric Options

relative norm(gradient) = 2.23e-07 OptimalityTolerance = 1e-06 (selected)

Finished ...
Elapsed time is 141.533911 seconds.

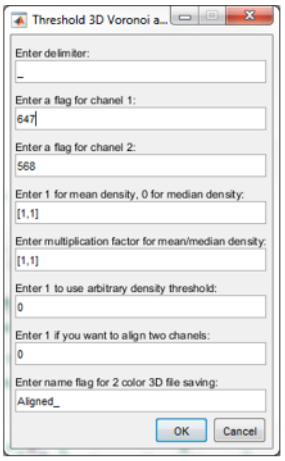


Figure 5. Setting the parameters for 2C VD thresholding

The alignment algorithm tries to find a translation vector, which minimizes 2D/3D Euclidean distance between localization in two channels using Matlab `fminunc` nonlinear programing solver. It works only if two channels display some degree of co-localization and the level of noise is low. You do not need a fiducial markers for algorithm to work, but they improve the results. If the data in two channels is not co-localizing, it is recommended to skip automatic alignment and use manual alignment described later (BOX 4). For better performance of the algorithm, it is advantageous to apply stringent VP density threshold accompanied with alignment. In the next step you can re-threshold VD using aligned file as described later in Re-thresholding voronoi diagrams_section. You can set an optimisation parameters passed to `fminunc` in the panel `Fit parameters` (for more details, see documentation for Matlab function `fminunc`)

7.2. Batch mode two colors voronoi data processing

Go to `2 color Voronoi` -> `2D/3D -> batch processing -> Threshold voronoi data` and press ok. You will be asked to select a root folder. All sub-folders and sub-sub-folders within a root folder will be analysed. Next, dialog box will appear and you will have to populate lines as described in Single file two color voronoi data processing subsection.

7.3. Single file one color voronoi data processing

Go to `1 color Voronoi` -> `2D/3D -> Single file -> Threshold voronoi data` and press ok. You will be asked to select a file for channel one or two. Next, dialog box will appear and you will have to populate following lines, **Figure 6**:

`Enter a channel flag` – enter a tag used to label selected channel

`Enter 1 for mean, 0 for median density` – VD can be automatically thresholded using mean or median value of VP density. This variable is a two element vector, e.g. [1,1] mean will be used for channels 1 and 2, [0,1] median will be used for channel 1 and mean for channel 2 etc.

`Enter multiplication factor for mean/median density` – this additional parameter allows you to offset mean/median density threshold by a multiplication factor e.g. [0.01,0.001], means that a threshold for a VP density will be mean/median (depending on previous choice) x 0.01 for

channel 1 and 0.001 for channel 2. You can also set and arbitrary density threshold for each channel by inserting 1 under

`Enter 1 to use an arbitrary density threshold.` The `Density` threshold values will be taken from main GUI window (highlighted in BOX 2, `Voronoi Tessellation parameters` panel).

`Enter channel number` – 1 or 2

`Enter dimensions for voronoi data processing`
-Enter 0 to determine the dimension based on the data file

The following variables will be saved (1 or 2, depending on the selected channel):

`dataout1/2` – is a matrix with single molecule data. The size is N1/N2 x 8, where N1/2 is number of points in respective channel. The column content is as follows: x coordinate, y coordinate, photon count, localization precision, z coordinate, frame number, a trace length – the number of frames given molecule was localized and z rejection tag (0 was marked as z rejected).

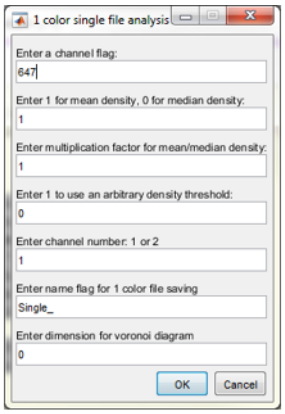


Figure 6. Setting the parameters for 1C VD thresholding

`dim` – dimension of the data, 2 or 3

`photonTH1/2` – photon number lower threshold

`locprec1/2` – localization precision upper threshold

`densitythr1/2` – VD density upper threshold

`Vp1/2` – vertices of the VP (Voronoi Polygons)

`Dp1/2` – vector of voronoi density

`namech1/2` – name of the file analysed, used for the re-thresholding of VD

7.4. Batch mode one color voronoi data processing

Go to `1 color Voronoi` -> `batch processing` -> `Threshold voronoi data` and press ok. You will be asked to select a root folder. All sub-folders and sub-sub-folders within a root folder will be analysed. Next, the dialog box will appear and you will have to populate lines as described in Single file one color voronoi data processing subsection.
When the analysis is finished, similar information will appear in the Matlab command line:
Sub folder #1 = EUFA423BRCA2_5h
Sub folder #2 = EUFA423_5h
No Sub folder #1 = EUFA423BRCA2_5h

Saved: G:\ grafeo sample data\EUFA423BRCA2_5h\Single_568_2.mat
Elapsed time is 11.089838 seconds.
No Sub folder #2 = EUFA423_5h
Saved: G:\ grafeo sample data\EUFA423_5h\Single_568_1.mat
Elapsed time is 31.098614 seconds.

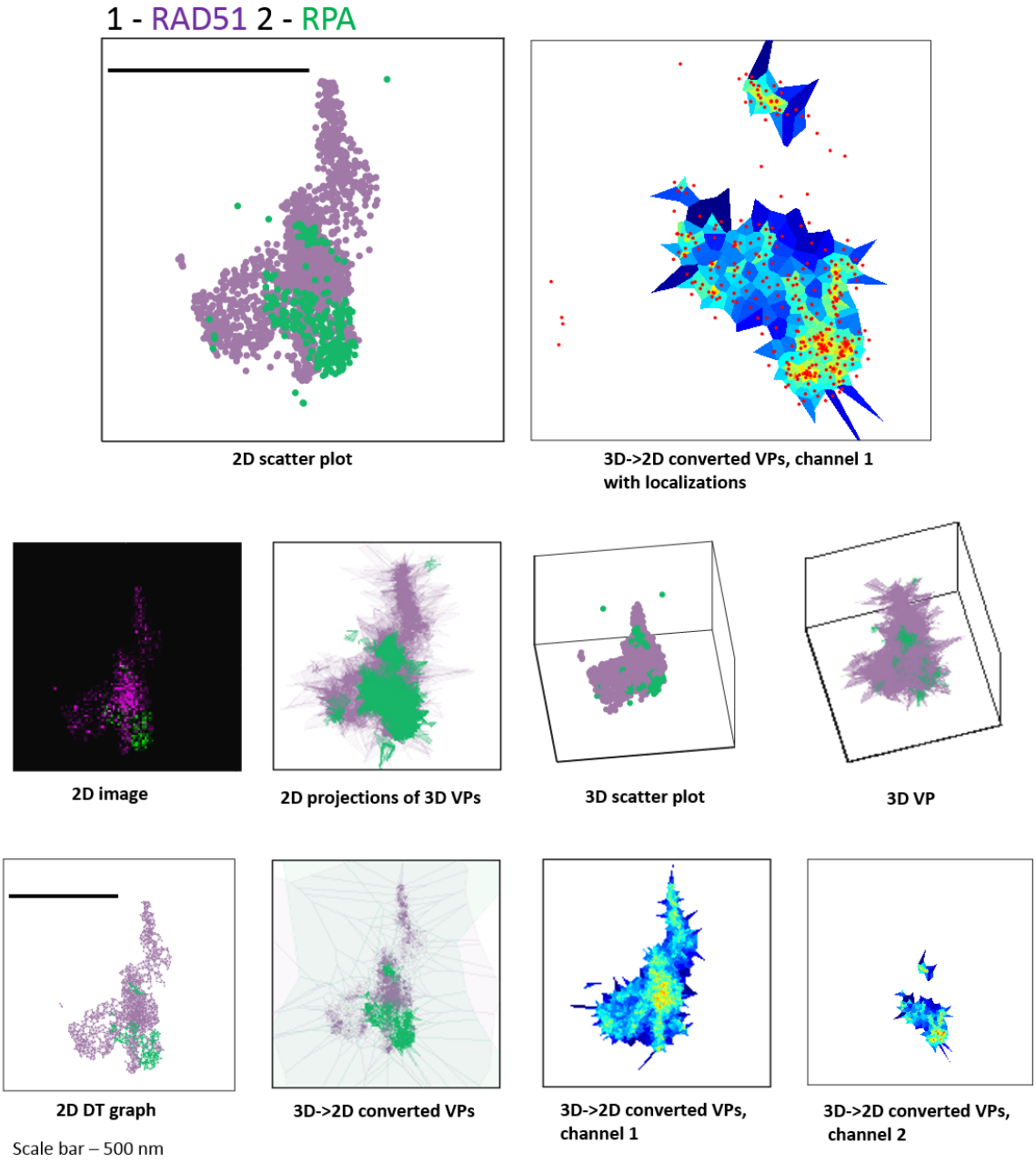


Figure 7. Visualization of the data.

8. Loading and visualizing the data

8.1. Loading two color voronoi data

Select `1+2 Chanel` from the drop down menu (BOX 6). Press `Load 1+2` (BOX 3) and select two color voronoi thresholded data file (e.g. select a file with the prefix `Aligned_` or `ReThr_` from the

Grafeo sample data folder). After loading is finished, two color scatter plot appears in the main Grafeo axes. The color of individual channels can be change with Color 1/2 edit boxes (BOX 6). To zoom in the data by pressing Toggle Zoom. To re-threshold voronoi data, populate two first columns in the Voronoi Tessellation parameters panel and then press Re-threshold (BOX 3). When you load a voronoi data, edit boxes in Voronoi Tessellation parameters panel will be automatically populated displaying following parameters: min photon # - the lower threshold for the photon count, Loc. Prec. – the upper threshold for the localization precision, Density – lower threshold for the VD density, min (max, min, median), minimum, maximum, mean and median of the data VD density.

8.2. Loading one color voronoi data

To load either channel 1 or channel 2 data select 1 Chanel /2 Chanel from the drop down menu (BOX 6). To clear main Grafeo axes check Clear axis check box. If you want to load and plot one channel and then load and overly it with second channel uncheck Clear axis check box (BOX 6). Press Load 1 or 2 (BOX 3) and select one color (you can also load individual channels from two color data file) voronoi data file (e.g. select a file with the prefix Single_ or ReThr_ from the Grafeo sample data folder). After loading is finished, two color scatter plot appears in the main Grafeo axes. To zoom in the data, press Toggle Zoom. To re-threshold voronoi data, populate two first columns in the Voronoi Tessellation parameters panel and then press Re-threshold (BOX 3).

8.3. Loading one color raw tessellated voronoi data

To load either channel 1 or channel 2 data select 1 Chanel /2 Chanel 1 from the drop down menu (BOX 6). To clear main Grafeo axes check Clear axis check box. If you want to load and plot one channel and then load and overly it with second channel uncheck Clear axis check box (BOX 6). Press Load raw (BOX 3) and select one color (you can also load individual channels from two color data file) voronoi data file (e.g. select an example file 568_1_Xwc_Ywc_I_F_Sig_Zc.mat_ or 568_1_Xwc_Ywc_I_F_Sig_Zc.mat from the Grafeo example data folder). Note, when the molecular list is imported, the first step of filtering based on photon count and localization precision is performed with the lower and upper threshold applied respectively. All the points with less photon than the threshold or higher localization precision (in terms of absolute value) are eliminated from the data. If you want e.g. lower the photon threshold, you will need to re-import the molecular list and change the threshold. Note, if a molecular list with given name exist, the file will not be re-imported, you will need to delete it first.

8.4. Displaying a super-resolved image

Grafeo uses various pointillist approaches to analyse SMLM data rather than working on a pixelated image. However, you can display and save a super-resolved image with a custom image and pixel size.

To display an image select from the drop down menu 1+2 Chanel for two color image, 1 (2) Chanel for chanel 1 (2) image. The parameters in the edit box next to the Chanel selection drop down menu, control an image display, the default values are: `[0 50 0 50 0 5 5 0 1].`
First two parameters are used as the minimum and maximum output image pixel values for contrast stretching of image 1, next two for contrast stretching of image 2. An output image is calculated as follows:
`Iout = (Iin - min(Iin)) * (T21 - T11)) / (max(Iin) - min(Iin))`, where `max(Iin)` and `min(Iin)` are the maximum and minimum pixel values of an input image `Iin`, `T21` and `T11` are the maximum and minimum values of an output image `Iout`, here `T21` = 50 and `T11` = 0 for both channels. Next three parameters control an image display. The parameter five is a minimum value for an image 1 and 2 display, sixth and seventh parameters are the maximum values for the display of an image 1 and 2 respectively. The eight parameter controls a scatterplot overly over an image, to display the scatterplot on the image, set it to 1. The last parameter determines if photon count (set to 1) or number of molecules (set to 0) evaluates the image intensity values. An image and a pixel size is controlled by the `Image size` and `Pixel size` edit boxes in the `Voronoi Tessellation parameters` panel.

8.5. Saving a super-resolved image

After displaying an image, you can save it using `Save image` push button. The following variables will be saved for two color image: `imRGB` – a scale output RGB image, where `imRGB(:,:,1)` and `imRGB(:,:,2)` are equal to scaled image 1, `imRGB(:,:,2)` is equal to scaled image 2. `Im1pw` and `im2pw` are unscaled image 1 and image 2. `Imsc` stores the parameters used for image scaling. If only channel 1 or 2 was displayed, only the `im1pw` or `im2pw` respectively will be saved and `imRGB` will be equal to one color scaled image.

8.6. Displaying a rectangular ROI as 2D/3D VD, DT or 2D/3D scatter plot

You can plot subset of a data as 2D/3D VD, DT or 2D/3D scatter plot. To create a rectangular ROI in the main axes press `rect roi` push button. The size of a rectangular ROI is control by the edit box next to `rect roi` push button. First two parameters control the position of the ROI, third and fourth the size of a ROI (all in nm). You can drag and resize the ROI. When ROI is created a cropped data set is displayed in a small secondary axes. In order to generate 3D scatter plot, 2D scatter plot, 3D voronoi diagrams, 2D voronoi diagrams, or 2D graph press the following bush buttons (highlighted by BOX 4) respectively, **Figure 7**: `3d scatter`, `2d scatter`, `3d voronoi`, `2d voronoi`, `2d Graph`. The 2D/3D voronoi plots are memory consuming; therefore, it is recommended to use small rectangular ROI. 3d plots are only supported for 3D SMLM data. When 3D data is analysed, 2d voronoi plot will display the 2d projections. If you want to recalculate 3D data using 2D voronoi polygons refer to Convert 3D VD to 2D VD section. To plot 2D/3D voronoi polygons or scatter plot a voronoi file has to be loaded (use Load 1+2 or Load 1 or 2). To plot 2d graph, graph based data has to be loaded. To load graph-based data press `Load graph` pushbutton (BOX 4). To display individual channels only, select `1 Chanel` or `2 Chanel` in a drop down menu (BOX 6). Two color 2D/3D voronoi polygons plot use a transparency to scale the size of a voronoi polygon. When only one channel is selected 2D/3D voronoi polygons is displayed using a heat map scaled using a voronoi density. To adjust the

display and scaling of channel 1 or 2 respectively, populate the parameters in two edit boxes located next to `2d voronoi` and `2d Graph` push buttons. There are five parameters in each box (channel 1 – upper, channel 2 -lower) to be determined (from the first to the last): the number of the density levels, minimum density, maximum density, density step and the last parameters controls the scatter plot overlying VPs.

8.7. Cropping an image

In order to crop an image, the image has to be created in Grafeo memory. Load the data and display a super resolved image as described in Displaying a super-resolved image sub-section. Then press scatter to display a scatter plot, create a rectangular ROI and then press `crop image` push button (BOX 4), **Figure 7**.

9. Alignment of two color data

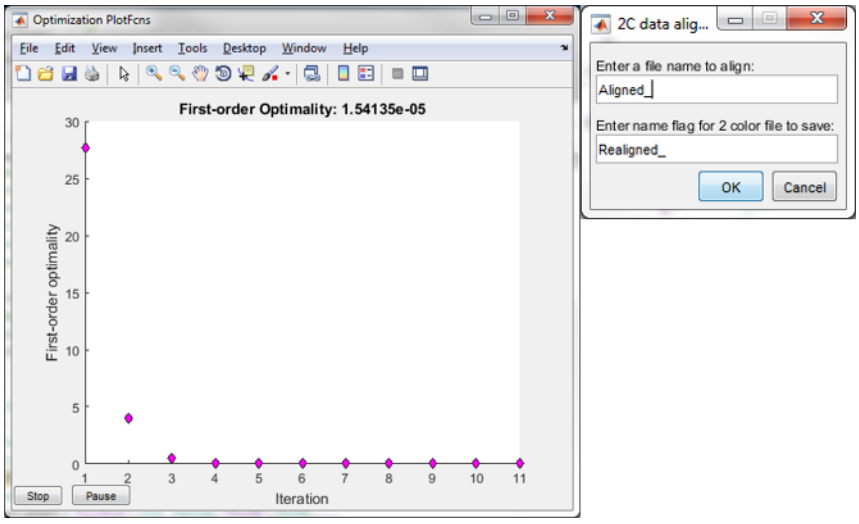


Figure 8. Alignment of 2C VD data

9.1. Manual alignment

First, load two color voronoi data file using `Load 1+2` push button. Channel 1 data is aligned with respect to channel 2 using following push buttons (BOX 4): `UP`, `DOWN`, `LEFT`, `RIGHT`, `Z down`, `Z up`. You can reset alignment in any dimension separately using RESET x, RESET y, RESET z or all the dimensions using `RESET ALL`. You can save an alignment vector using `Save align`, load an alignment vector using `Load align` and apply an alignment vector to an open data using `Apply align`. A step of an alignment can be changed using `Step` edit box (see BOX 2, default 100 nm).

9.2. Single file automatic alignment after voronoi data thresholding

Go to `2 color Voronoi` -> `2D/3D` -> `Single file` -> `Align two channels` and press ok. First, the dialog box will appear and you will need to enter a flag name for the files to be aligned and a flag name for the file saving, **Figure 8**. An optimization process is based on Matlab nonlinear solver `fminunc`. The input parameters of the solver can be changed in the `Fit parameters` panel (BOX 5), where `TolFun` is a termination tolerance on the function value, `TolX` is a termination tolerance on the optimization coefficient, current step, `MaxFunEval` is a maximum number of function evaluations allowed and `MaxIter` is a maximum number of iterations allowed, `X0` range vector for the coefficient initialization, e.g. the initial values of the optimization coefficient 1 will be:
`randi([-X0(1),X0(1)],1)`. To learn more about stopping criteria, please refer to Matlab documentation page `Tolerances and stopping criteria`. At the start of the optimization process Matlab Optimization `PlotFcns` will pop up displaying the first-order optimality measure, **Figure 8**. Following optimization variables will be saved to an output file:

`exitflag` - describe exit condition of `fminunc` solver (e.g. equal 1 if global minimum is found), empty if no alignment

`output` - structure with the information about optimization process, empty if no alignment

`zcount` – z direction step for the alignment of channel 1 with respect to channel 2, empty if not align or for 2D data

`updowncount` – x direction step for the alignment of channel 1 with respect to channel 2, empty if not align

`leftrightcount` – y direction step for the alignment of channel 1 with respect to channel 2, empty if not align

9.3. Batch mode automatic alignment after voronoi data thresholding

Go to `2 color Voronoi` -> `2D/3D` -> `batch processing` -> `Align two channels` and press ok. First, the dialog box will appear and you will need to enter a flag name for the files to be aligned and a flag name for the file saving. Then, select a root folder for voronoi data re-thresholding. All sub-folders and sub-sub-folders within a root folder will be analysed.
For more details on the optimization process, please refer to Single File automatic alignment after voronoi data thresholding sub-section. After alignment of each file, similar information (Matlab automatic iterative display) show up in the Matlab command line:

Iteration	Func-count	f(x)	Step-size	First-order optimality
0	4	90138.3		48.7
1	16	85347.6	1.87038	27.6

2	20	83225.8	1	3.95
3	24	83172.3	1	0.495
4	28	83171.7	1	0.0451
5	32	83171.7	1	0.00835
6	36	83171.7	1	0.0017
7	40	83171.7	1	0.000906
8	44	83171.7	1	0.000444
9	48	83171.7	1	0.000148
10	52	83171.7	1	7.69e-05
11	56	83171.7	1	1.54e-05

Optimization completed: The first-order optimality measure, 3.104217e-07, is less than options.OptimalityTolerance = 1.000000e-06.

Optimization Metric Options

relative norm(gradient) = 3.10e-07 OptimalityTolerance = 1e-06 (selected)

10. Re-thresholding voronoi diagrams

10.1. Single file two color voronoi data re-thresholding

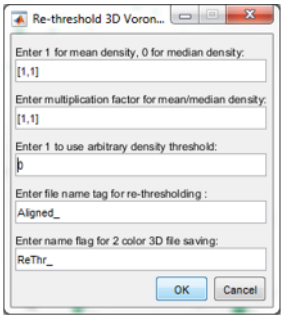


Figure 9. Setting the parameters for 2C VD re-thresholding

Go to `2 color Voronoi` -> `2D/3D` -> `Single file` -> `Re-threshold voronoi data` and press ok. First, the dialog box will appear and you will need to populate following lines as described in Single file two color voronoi data processing subsection, **Figure 9**: `Enter 1 for mean, 0 for median density`, `Enter multiplication factor for mean/median`, `Enter 1 to use arbitrary density threshold.`

In the next line, `Enter file name flag for re-thresholding` enter name tag for the files to be re-processed. The default is `Aligned_` and only files containing this string will be analysed. Last, `Enter name flag for 2 color 2D/3D file saving`, enter a name tag, default is `ReThr_`, which will replace `Aligned_` in the processed file e.g. if the input file has a name `Aligned_647_568_1.mat`, the re-thresholded output file will have the name `ReThr_647_568_1.mat`. Lastly, you will be prompted to select a file for two color voronoi data re-thresholding.

10.2. Batch mode re-thresholding of two color voronoi data

Go to `2 color Voronoi -> 2D/3D -> batch processing -> Re-threshold voronoi data` and press ok. First, you will be prompted to select a root folder for voronoi data re-thresholding. All sub-folders and sub-sub-folders within a root folder will be analysed.

Next, the dialog box will appear and you will need to populate lines as explained in Single file two color voronoi data processing subsection.

After batch re-thresholding is finished, similar information appears in the Matlab command window:

```
Sub folder #1 = EUFA423BRCA2_5h
Sub folder #2 = EUFA423_5h
No Sub folder #1 = EUFA423BRCA2_5h
Completed: G:\ grafeo sample data\EUFA423BRCA2_5h\ReThr_ 647_568_2.mat
Folder analysed ...
Elapsed time is 15.952255 seconds.
No Sub folder #2 = EUFA423_5h
Completed: G: \ grafeo sample data\EUFA423_5h\ReThr_ 647_568_1.mat
Folder analysed ...
Elapsed time is 28.675930 seconds.
```

10.3. Batch mode re-thresholding of one color voronoi data

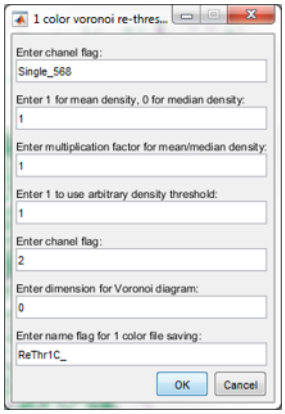


Figure 10. Setting the parameters for 1C VD re-thresholding

Go to `1 color Voronoi -> batch processing -> Re-threshold voronoi data` and press ok. First, you will be prompted to select a root folder for voronoi data re-thresholding. All sub-folders and sub-sub-folders within a root folder will be analysed.

Next, the dialog box will appear and you will need to populate lines as explained in Single file two color voronoi data processing subsection, **Figure 10**.

After batch re-thresholding is finished, similar information appears in the Matlab command window:

```
Sub folder #1 = EUFA423BRCA2_5h
Sub folder #2 = EUFA423_5h
No Sub folder #1 = EUFA423BRCA2_5h
saved: G:\ grafeo sample data\EUFA423BRCA2_5h\ReThr1C__2.mat
Elapsed time is 12.219575 seconds.
No Sub folder #2 = EUFA423_5h
saved: G:\ grafeo sample data\EUFA423_5h\ReThr1C__1.mat
Elapsed time is 31.613545 seconds.
```

11. Deleting files in batch mode

To remove all the files containing a given string go to `2 color Voronoi` -> `2D/3D` -> `batch delete files` and press ok. First, you will be prompted to select a root folder. All the files containing given string will be removed from all sub-folders and sub-sub-folders within a root folder. Next, the dialog box will appear and you will have to provide the search string for files to be delated and search string for files to be retained; then press ok. Search string to retain is useful if certain files contains both string to be deleted and retained.

Let's say you want to remove all the files containing '`ReThr_`' string. After batch deletion is finished, following information appears in the Matlab command window:

```
Sub folder #1 = EUFA423BRCA2_5h
Sub folder #2 = EUFA423_5h
No Sub folder #1 = EUFA423BRCA2_5h
deleted:  G:\ grafeo sample data\EUFA423BRCA2_5h\ReThr_ 647_568_2.mat
No Sub folder #2 = EUFA423_5h
deleted:  G:\ grafeo sample data\EUFA423_5h\ReThr_ 647_568_1.mat
```

If you want to reanalyse the data, but a file with the same already exist, `Already exist` message appears in the Matlab command line and the analysis will not proceed. You will need to first remove all the file with given name type. You can do it manually or in batch mode.

12. Convert 3D VD to 2D VD

Once the 3D STORM data was processed using 3D VD, you can convert given file to 2D VD. This option is available only in single file mode and two color files. First, load 2C 3D STORM voronoi processed file using `Load 1+2` push button. Go to Re-calculate Voronoi -> single file -> 2 Color – 3D->2D Voronoi and press ok. Loaded file will be converted to 2D VD and you will be prompted to save a new 2D VD file.

13. Data filtering based on univariate and bivariate nearest neighbour distance function

13.1. Two color data, single file mode filtering

This function allows deleting a localization if it has less neighbouring localization of same/different 'color' than present number at a chosen distance. First, load two channel file using `Load 1+2` push button. Enter desired parameters values in the `Nearest neighbour distance filtering`

panel (BOX 5). nnd dim – the nearest neighbour distance dimension, can be 2 or 3 (only for 3D STORM). #nn(1-2,2-1) – the number of nearest neighbours for bivariate filtering required to keep given localization; 1-2 – measured from color 1 to color 2 (Channel 1 column) and 2-1 from color 2 to color 1 (Channel 2 column). #nn(1-1,2-2) – the number of nearest neighbours for univariate filtering required to keep given localization; 1-1 – measured from color 1 to color 1 (Channel 1 column) and 2-2 from color 2 to color 2 (Channel 2 column). Dist(1-2,2-1) (Dist(1-1,2-2)) – bivariate (univariate) distance threshold at which each localization has to have chosen number of nearest neighbours – #nn(1-2,2-1) (#nn(1-1,2-2)) – to be retained in the data set. Example data was analysed with default settings (BOX 5). I.e. gtiven point in channel 1 (violet) was retained if the mean distance to 10 nearest neighbours from chanel 2 (green) is equal or smaller than 150 nm **OR** if the mean distance to 40 nearest neighbour points of the same kind (violet) is equal or smaller than 150 nm. Likewise for channel 2. Bivariate and univariate thresholding are operating as logical OR operation with respect to each other. This reduces likelihood of 'forced co-localization' and functions to remove a 'small' isolated structures, often corresponding to noise or isolated proteins. Histc step is a histogram bin siz. After setting the parameters press Filter nnd. The figure appears showing four histogram plots, **Figure 12**: mean nearest neighbour distance from the point set 2 to the point set 1, from the point set 1 to the point set 2, from the point set 1 to the point set 1 and from the point set 2 to the point set 2. This figure helps to estimate quickly the proximity and structure size in two point data sets. At the same time, filtered scatter plots appears in the main axis of the Grafeo GUI. You can use Toggle Zoom push button to zoom in the data and estimate the quality of the filtering. You can change the parameters in the filter settings multiple times and press Filter nnd without filtering the data. In order to display the unfiltered data again press Scatter push button (highlighted in BOX 3). Once you are happy with the chosen filter parameters, press Apply filter, which filters out the data. To save the filtered data, press Save filter and to load previously filtered data press Load filter. It is possible to filter the data in multiple steps, by repetitively pressing Apply filter. You can undo one filtering step by pressing Undo filter.

10.2. Two color data, batch mode filtering

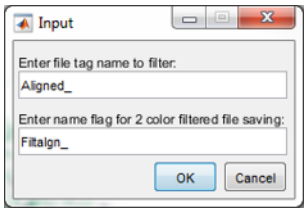


Figure 11. Setting the parameters for nearest neighbour based data filtering

Set filtering parameters as described in Two color data, single file mode filtering. Press Batch filtering 2 Color. First, you will be prompted to select a root folder. All the files from all sub-folders and sub-sub-folders within a root folder will be filtered. Next, you will be ask to provide file name tag and saved file prefix tag, **Figure 11**. All the files containing file name tag will be filtered and automatically saved with chosen prefix. After filtering is finished, following information will display in the Matlab command line:

Sub folder #1 = EUFA423BRCA2_5h
Sub folder #2 = EUFA423_5h
No Sub folder #1 = EUFA423BRCA2_5h
Saving ...G:\ grafeo sample data\EUFA423BRCA2_5h\FiltaIgn_647_568_2.mat
Finished ...

No Sub folder #2 = EUFA423_5h
Saving ...G:\ grafeo sample data\EUFA423_5h\Filtalgn_647_568_1.mat
Finished ...

10.3. One color data, single file mode filtering

Load voronoi data using `Load 1+2` or `Load 1 or 2` push buttons. Select a channel you want to filer in a drop down menu (BOX 6), `1 Chanel` or `2 Chanel`. The loaded file can be two channel data, but only one channel will be filtered. Then proceed as described in Two color data, single file mode filtering section, populating the edit boxes corresponding to selected channel. Filtering is based on the univariate nearest distance function only.

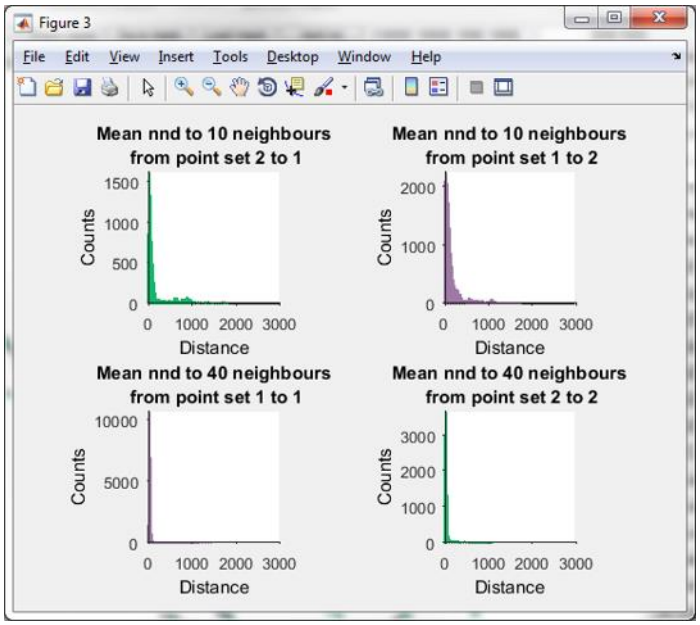


Figure 12. Displaying the summary of nearest neighbour 2C data filtering.

10.4. One color data, batch mode filtering

Set filtering parameters as described in One color data, single file mode filtering. Press `Batch filtering 1 Color`, and push button. First, you will be prompted to select a root folder. All the files from all sub-folders and sub-sub-folders within a root folder will be filtered. Then you will be prompted to enter name tag for files to be filtered and a name tag for the file saving.

11. Creating a region of interest ROI

11.3. Polygonal region of interest

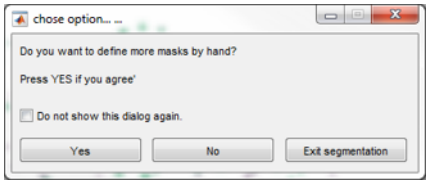


Figure 13. Creating a polygonal ROI.

Load a file using `Load 1+2` or `Load 1 or 2` (for single color/channel data or if you want to visualize just one channel). Press `Poly roi` push button. A small message box will appear, press `ok` to draw a freehand ROI (region of interest), **Figure 13**. You can draw multiple ROI by pressing `ok` each time. When you are finished, press `no`. By pressing `exit segmentation`, you will interrupt the operation and no ROI will be created. Press `Eval. poly roi`, then Save only roi. You can load previously saved poly roi by pressing `Load poly roi` and then `Plot poly roi` to visualize it. The name of poly roi file should be `mask_number.mat` e.g. `mask_1.mat`, where the number is automatically taken from the loaded voronoi data file. Keep this name structure for batch processing mode, where mask file will be automatically associated with voronoi data file.

11.4. Create an image mask/ROI

Load a file using `Load 1+2` or `Load 1 or 2` (for single color/channel data or if you want to visualize just one channel). To control an image size and pixel size enter an `Image size` (in nm) and `Pixel size` (in nm) parameters in the `Voronoi Tessellation parameters` panel. Press `Image` push button, a super resolved image appears in main Grafeo axes.Then press `Draw mask`. A small message box will appear, press `ok` to draw a freehand ROI (region of interest). You can draw multiple ROI by pressing `ok` each time. You can create holes in a mask by selecting `remove 1s`. When you are finished, press `no`. By pressing `exit segmentation`, you will interrupt the operation and no ROI will be created. To save an image mask press `Save mask`, to load previously saved mask press `Load mask`.

12. Object analysis and co-localization

12.3. Two color single file co-localization analysis

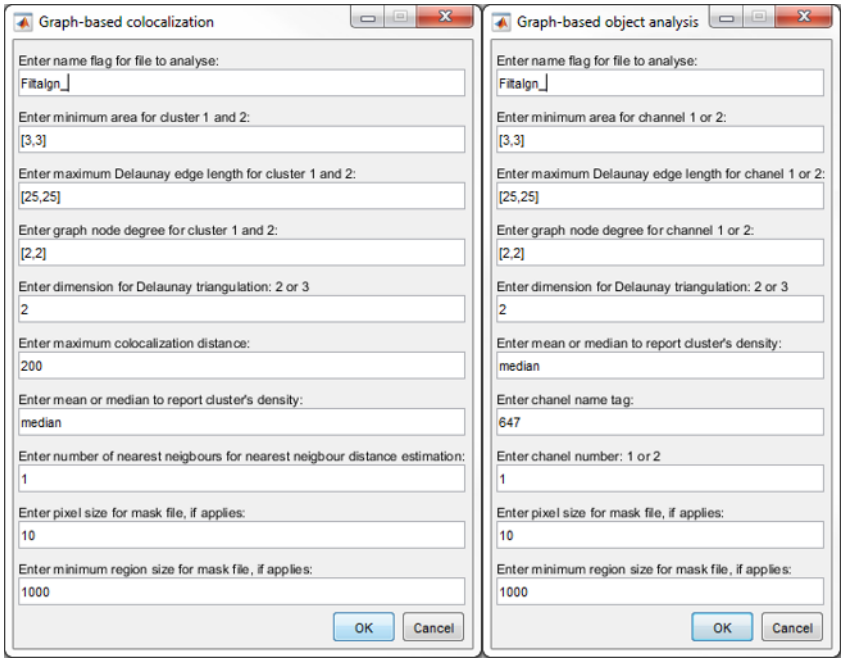


Figure 14. Setting the parameters for graph-based co-localization and object based analysis.

Object analysis and co-localization can be performed on all the data or ROI. To create a ROI for each file read Creating a region of interests section.

After data tessellation, thresholding of VD and optional filtering based on nearest neighbour distance data can be segmented into discreet clusters by thresholding edges of Delaunay Triangulation DT. In this section cluster, object and graph are exchangeable.

Go to `Object analysis & colocalization` -> `single file` -> `2 Color -> Graph Colocalization`. The dialog box will appear and you will need to populate following line as explained below, **Figure 14**.

`'Enter a file name flag:'` – only files containing this string will be analysed.
`'Enter minimum area for cluster 1 and 2:'` – after segmentation, all clusters smaller then this value will be supressed. Here area is given by the number of localization contained within a graph/cluster. It is a two-element vector, one element per each channel.

`'Enter maximum Delaunay edge length for cluster 1 and 2 segmentation:'` – All edges larger than this value will be removed, creating disconnected/segmented discreet clusters represented as graphs. It is a two-element vector, one element per each channel.

`'Enter graph node degree for cluster 1 and 2:'`- a degree is a number of connection given point has within a graph. All points with degree smaller than that will be suppressed. This operation removes 'hairs' of the graph. It is a two-element vector, one element per each channel.

`'Enter dimension for Delaunay triangulation: 2 or 3'` – Dimension of DT, typically 2.

'Enter minimum C2C distance for colocalization fraction calculation:' – if two structures of different color will be find at this distance or smaller, they will be considered as co-localized.

'Enter mean or median to report cluster's density:' –

'Enter number of nearest neighbours for nearest neighbour distance estimation:' – Point to point (P2P) distance will be estimated between each point of one color and chosen number of neighbours in other color. The default is 1 i.e., for each point within a graph the nearest neighbour distance will be estimated to 1 closest point in the nearest graph of other color/kind.

'Enter a pixel size for mask file, if applies:' – If you chose to use image mask (described in Image region of interest) you will need to provide the pixel size used, default is 10 nm (see BOX 2, Voronoi Tessellation parameter panel).
'Enter minimum region size for mask file, if applies:' – The smallest ROI to be analysed.

Output file contains different variables describing the properties of Ng1/Ng2.
(number of clusters/graphs in channel 1 (2)) and their co-localization.
Following variables will be saved (1 and 2 corresponds to channel tag, unless stated otherwise):

Ani1/2 – the distance between centroid of a segmented graph to all the points/nodes belonging to the same graph. It is a vector cell array with the number of cells equal to the number of graphs Ng1/Ng2.
Each cell is a vector of distances created by the Matlab function pdist2 of the size Npx1, where Np is the number of points/nodes within a graph.

AniMed1/2 – is a vector with the number of elements equal to number of graphs Ng1/Ng2.
Each element is a median of the corresponding number of Ani1/2 cell.

CW1toCW2sort – is a sorted distance matrix between weighted centroids of a graph of the first kind/color to the weighted centroid of a graph of second kind/color (C2C distance). It size is equal to Ng1xNg2. This matrix contains all the nearest neighbour C2C distances from graph set 1 to graph set 2 sorted in ascending order.

CellArea – is a six element vector with following variables: area of the ROI in nm, number of co-localized graphs as measured by C2C distance, number of co-localized graphs in channel 1 as measured by P2P distance, number of graphs in channel 1, fraction of co-localized graphs in chanel 1 (number of co-localized graphs in channel 1 as measured by C2C/ number of graphs in channel 1), fraction of co-localized graphs in chanel 1 (number of co-localized graphs in channel 1 as measured by P2P/ number of graphs in channel 1)

Cen1/2 – vector of the centroid coordinates of all segmented graphs in channel 1/2. Its size is Ng1/Ng2.

DE1/2 – is a distance matrix between two ends of a graph from the set 1 to the closest point from the graph from set 2. Here 1\2 correspond to two ends of a graph from a set 1. Ends of a graph are defined as two nodes connecting maximum shortest path spanning the graph (see below).

DGsub1/2 – is a cell array of distance matrices between every point in a graph to every other point in the same graph. The number of elements (cells) is Ng1/Ng2.

Det1/2 – is a vector with the number of molecules/nodes within all segmented graphs, with the size Ng1/Ng2

Dp1/2 – is a vector of the densities (an inverse of the voronoi polygon VP size), with the size N1/N2, where N1/N2 is a total number of points in channel 1/2 (number of rows in the variables dataout1 (dataoutshift1) and dataout2).

DpGsubMed1/2 – is a vector with the median (mean, defined by `flag` input parameter) densities of a graph, which size is Ng1/Ng2

G1/2 – is a total, non-segmented weighted graph formed of DT edges containing all points from channel 1/2. G1/2 is generated using Matlab command `G = graph(edgex,edgey,w)` where edgex/y are x/y coordinates of DT edges and w is a weight, here a length of an edge.

Gbin1/2 – is a cell array with segmented graphs, containing row indices of the points in a variable dataoutshift1 and dataout2, the size is Ng1/Ng2

Gsub1/2 – is a cell array with segmented graphs containing edges and points, which size is Ng1/Ng2. Each cell contains a segmented graph (subgraph, see `H = subgraph(G,nodeIDs)`) object.

Len1/2 – is a vector with the number of localization within a graph taking into account multiple single molecule counting, with the size Ng1/Ng2. See Nikon Trace Length parameter.

Vol1/2 – is a vector of the total area/volume of all segmented graphs, the size is Ng1/Ng2

VorDpSum1/2 – is a vector of the summed voronoi density of all segmented graphs, the size is Ng1/Ng2

VorDpAll1/2 – is a cell array of the voronoi densities of each graph, the size is Ng1/Ng2 and the size of each cell is equal to the number of the nodes in the graph/cluster.

Vp1/2 – is a cell array of the voronoi polygons. The size is N1/N2.

WC1/2 – is a vector of the weighted centroids of all segmented graphs, the size is Ng1/Ng2

XYZnodes1/2 – is a cell array containing nodes coordinates of each graph. The size is Ng1/Ng2 and the size of each cell is equal to the number of the nodes in the graph/cluster x dimension (2 or 3).

dataout1/2 – is a matrix with single molecule data. The size is N1/N2 x 8. The column content is as follows: x coordinate, y coordinate, photon count, localization precision, z coordinate, frame number, a trace length – the number of frames given molecule was localized and z rejection tag (0 was marked as z rejected).

dataoutshift1 – is a the same as dataout1 but align to the chanel 2.

Dim – dimension of the data, 2 or 3

dimdeluney– dimension of a DT, 2 or 3

dist1nn – is a 12 x Ng1 matrix containing following variables:

-the median of the numnn-th nearest neighbour point to point (P2P) distances for every point/node in a graph from a set 1 to the closest point/node in a graph from a set 2

- the median of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 2 to the closest point/node in a graph from a set 1

-the mean of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 1 to the closest point/node in a graph from a set 2

-the median of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 2 to the closest point/node in a graph from a set 1

-the standard deviation of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 1 to the closest point/node in a graph from a set 2

-the standard deviation of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 2 to the closest point/node in a graph from a set 1

-the standard error of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 1 to the closest point/node in a graph from a set 2

-the standard error of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 2 to the closest point/node in a graph from a set 1

-the minimum of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 1 to the closest point/node in a graph from a set 2

-the minimum of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 2 to the closest point/node in a graph from a set 1

-the maximum of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 1 to the closest point/node in a graph from a set 2

-the maximum of the numnn-th nearest neighbour P2P distances for every point/node in a graph from a set 2 to the closest point/node in a graph from a set 1

`distcoloc` – is an input parameter provided by the user. The maximum C2C distance at which two graphs/clusters are considered co-localized.

`Fac1/2` – is an input parameter provided by the user. The edge length threshold. All edges longer than fac1/2 will be suppressed

`flag` – is an input parameters provided by the user, can be 'mean' or 'median'. Parameter used to calculate DpGmed.

`Frame1/2` – number of frame in channel 1/2.

`Ind12C2C` – matrix of indices of sorted weighted centroid-to-centroid (C2C) distances between two sets/color of graphs (chanel 1/2). Indices are pointing to the graph number (can be used to access an element in the variable like Cen1/2, Det1/2 Gbin1/2 etc…). Its size is Ng1 x Ng2.

`extension1/2` – is a vector of graph extension – the maximum path spanning the graph, the size is Ng1/Ng2

`extensionIND1/2` – is a two column vector of the indices to the beginning and end nodes forming `extension1/2`, the size is Ng1/Ng2

`AllShortPath1/2` – is a cell array containing the lengths of a shortest path from node *i* to node *j* for each graph/cluster, the size of a cell array is Ng1/Ng2 and the size of each cell is equal to the number of the nodes in the graph/cluster.

`MedShortPath1/2` – is a vector the median of shortest paths `maxDGsub1`, the size is Ng1/Ng2.

`Fac1/2` – is an input parameter provided by the user. The minimum number of points/nodes in a graph.

`minDE1ind` – is a vector of indices to the start-node connecting the maximum shortest path spanning the graph, the size is Ng1.

`minDE2ind` – is a vector of indices to the end-node connecting the maximum shortest path spanning the graph, the size is Ng1.

`minregmask` – is an input parameter provided by the user. The minimum size of a region in the image mask to be analysed.

noddeg1/2 – is an input parameter provided by the user. The minimum degree of a node (number of edges it forms) of the graph.

numnn – is an input parameter provided by the user. The number of the nearest neighbours to be considered in calculation of dist1nn variable.

pcacoeff1/2 – is a cell array of the principle component coefficients calculated on the graph nodes coordinates, the size is Ng1/Ng2.

pcascore1/2 – is a cell array of the principle component scores calculated on the graph nodes coordinates, the size is Ng1/Ng2.

pcalatent1/2 – is a cell array of the principle component variances calculated on the graph nodes coordinates, the size is Ng1/Ng2.

pcaexplained1/2 – is a cell array of the percentages of the total variance explained by each principal component, the size is Ng1/Ng2.

ratio1/2 – is a vector of the ratio of the maximum to minimum pcaexplained1/2, the size is Ng1/Ng2.

totalMask1 – is an input ROI/mask save in the same folder as analysed data.

var1/2 – is a matrix containing different parameters of graphs/clusters in channel 1/2. Its size is Ng1/Ng2 x 8. The eight columns contain following variables:

var1/2(:,1)= VorD1 - the summed density of a graph

var1/2(:,2)= Vol1/2 - the area (volume) of graphs

-var1/2(:,3)= Det1/2 - number of nodes within a graph

-var1/2(:,4)= DpGmed1/2 - the median (mean) density of a graph

-var1/2(:,5)= mSHpath1/2 - extension, the maximum shortest path spanning a graph

-var1/2(:,6)= ratio1/2

-var1/2(:,7)= Pnum1/2 - total number of photons in a graph

-var1/2(:,8)= Len1/2 - total number of nodes/points in a graph including multiple localizations.

`varSort2` – is a matrix containing different parameters of the 1st to the -nth = Ng1-th nearest neighbour graphs/clusters in channel 2 co-localizing with graph in channel 1. Its size is Ng1xNg2 x 8. Column variables are the same as in `var1/2`, but calculated for all nearest neighbours; e.g. the graph number 1 in channel 1 has parameters described by `var1(1,:)` and the parameters of its first nearest neighbour (measured by the weighted by the photon number C2C distance) graph from chanel 2 are contained in `varSort2(1,1,:)` and various distance co-localization measures in `dist1nn(1,:)`.

The object analysis uses following Matlab functions: `G = graph(A)`, `DT = delaunayTriangulation(X,Y,Z)`, `Gsub = subgraph(G,nodeIDs)`, `[coeff,score,latent,tsquared,explained,mu] = pca(A).` If you want to learn more about these functions, please refer to the Matlab documentation.

12.4. Two color batch mode co-localization analysis

Go to `Object analysis & colocalization` -> `batch processing` -> 2 `Color -> Graph-based colocalization`. You will be asked to select a root folder. All sub-folders and sub-sub-folders within root folder will be analysed. The dialog box will appear and you will need to populate it as explained in Two color single file co-localization analysis section, **Figure 14**. When batch processing is completed, following information will display in the Matlab command window:

Sub folder #1 = EUFA423BRCA2_5h

Sub folder #2 = EUFA423_5h

No Sub folder #1 = EUFA423BRCA2_5h

Saved: G:\ grafeo sample data\EUFA423BRCA2_5h\GraphDelune2d3D_2.mat

No Sub folder #2 = EUFA423_5h

Saved: G:\ grafeo sample data\EUFA423_5h\GraphDelune2d3D_1.mat

If following files already exist, following information will appear in the Matlab commands window:

Sub folder #1 = EUFA423BRCA2_5h

Sub folder #2 = EUFA423_5h

No Sub folder #1 = EUFA423BRCA2_5h

Already Analysed

No Sub folder #2 = EUFA423_5h

Already Analysed

If you want to reanalyse these files, first delete old files (manually or in batch mode). The default file name is GraphDelaunay2d_number.mat for 2D DT and GraphDelaunay3d_number.mat for 3D DT.

12.5. One color single file graph-based object analysis

Object analysis of one channel can be performed on all the data or masked data using ROI. To create a ROI for each file read Creating a region of interests section.

After data tessellation, thresholding of VD and optional filtering based on nearest neighbour distance data can be segmented into discreet clusters using edge length thresholding of Delaunay Triangulation DT.

Go to `Object analysis & colocalization` -> `single file` -> `1 Color -> Graph-based object analysis`

First, you will be prompted to select a file for the analysis. Then, the dialog box will appear and you will need to populate it as explained in Two color single file co-localization analysis section, **Figure 14**. Two additional lines include `Enter channel name tag:` – name tag of a channel you want to analyse and `Enter chanel number: 1 or 2` – 1 or 2. For vector input variable, only the one corresponding to chosen channel will be used. The default file name is SingleGraph2d_number.mat for 2D DT and SingleGraph3d_number.mat for 3D DT.

12.6. One color batch mode graph-based object analysis

Go to `Object analysis & colocalization` -> `batch processing` -> `1 Color` -> `Graph-based object analysis`. You will be asked to select a root folder. All sub-folders and sub-sub-folders within a root folder will be analysed. The dialog box will appear and you will need to populate it as explained in One color single file graph-based object analysis section, **Figure 14**. Once batch processing is completed, following information will display in the Matlab command window:

Sub folder #1 = EUFA423BRCA2_5h

Sub folder #2 = EUFA423_5h

No Sub folder #1 = EUFA423BRCA2_5h

Saved: G:\Current Users\kalina\grafeo sample data\EUFA423BRCA2_5h\SingleGraph2d3D_568_2.mat

No Sub folder #2 = EUFA423_5h

Saved: G:\Current Users\kalina\grafeo sample data\EUFA423_5h\SingleGraph2d3D_568_1.mat

13. Ripley' function analysis

13.3. Two color single file object-based Ripley's function analysis

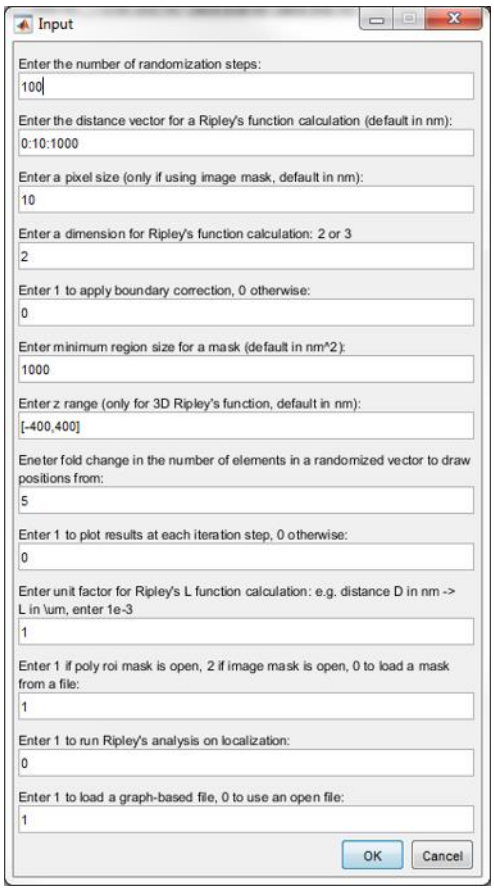


Figure 15. Setting the parameters for Ripley's analysis.

Ripley's function analysis can be calculated on the single molecule localization data or on the centroid of clusters/objects/graphs. This section discusses the later. Grafeo v1.beta supports only single file Ripley's function analysis, batch mode will be added in future releases.

First, load a voronoi tessellated two channel data file using `Load 1+2` or `Load filter` and draw a region of interest ROI (Ripley's analysis requires a mask/ROI). If you have previously created ROI/mask, you can chose to load it later, then you do not need to Load any voronoi tessellated data file. Ripley's function and akin spatial statistics functions should be always compared to a null model. A null model for univariate (bivariate) Ripley's function is a random (independent) distribution of clusters in a set (two sets) of points. The simplest way to test a null hypothesis of a randomness (an independence) is a repetitive randomization of all the points in a set and calculation of a spatial statistics function (Ripley's function here) at each randomization step. This way one can obtain a confidence interval of Ripley's function corresponding to a random (an independent) distribution of a set (two sets) of points. In this release of Grafeo only two different functions are calculated: univariate and bivariate Ripley's K-functions and transformed Ripley's L-functions (note here the L function corresponds to L(d) - d). The Ripley's functions are calculated as described in *Dixon et al.,* (Dixon, 2002) http://lib.dr.iastate.edu/stat_las_preprints/52/ .

Go to `Ripley's analysis` -> `Single file` -> `2 Color` -> `Run Randomization`. You will be asked to select a root folder. All sub-folders and sub-sub-folders within a root folder will be analysed. The dialog box will appear and you will need to populate it as explained below, **Figure 15**:

`Enter the number of randomization steps:` -an input variable name- `Nsimul` – number of times clusters/object/graphs will be randomized to get a confidence intervals for a null hypothesis.
`Enter the distance vector for a Ripley's function calculation (default in nm):` -an input variable name- `D` -the distances at which Ripley's function will be evaluated

`Enter a pixel size (only if using image mask, default in nm):` -an input variable name-`pix` -the size of a pixel (in nm) of an image mask, only if you use image mask and not a poly roi mask.

`Enter a dimension for Ripley's function calculation: 2 or 3:` -an input variable name- `dim` -Ripley's functions can be calculated in 2 (2D and 3D STORM) or 3 (only 3D STORM) dimensions

`Enter 1 to apply boundary correction, 0 otherwise:` -an input variable name- `Boundarycorrection` -edge correction can be very time consuming. If the distances used to compute Ripley's functions are smaller than the diameter of a studied region/ROI, edge corrections can be avoided. For 3D Ripley's function, edge corrections are applied only in lateral direction

`'Enter minimum region size for a mask (default in nm^2):` -an input variable name- `minregmask` -valid only for image masks/ROI when multiple ROI are created

`Enter z range (only for 3D Ripley's function, default in nm:` -an input variable name- `z range` -valid only for 3D data, z-range for 3D Ripley's function evaluation

`Enter fold change in the number of elements in a randomized vector to draw: positions from:` -an input variable name- `multfac` -if the data has Np points, for each randomization step temporary variable with `multfac` x Np will be created storing random points located within a study region ROI.

`Enter 1 to plot results at each iteration step, 0 otherwise :` -an input variable name- `plotflag` -if you want to plot Ripley's function at every step of randomization, enter 1. This may slow down the processing and use up the memory.

`Enter unit factor for Ripley's L function calculation: e.g. distance D in nm -> L in um, enter 1e-3:` -an input variable name- `unitfactor` - if the input distances D are given in nm, to get e.g. evaluation of Ripley's functions in μm^{-2} enter 10^{-3}.

`Enter 1 if poly roi mask is open, 2 if image mask is open, 0 to load a mask from a file:` -if current instance of Grafeo has polygonal ROI open, enter 2, if image mask is open, enter 1, if ROI masked, either polygonal or image mask/ROI, is stored in a file, enter 0. If no ROI exist, please draw a ROI first.

`Enter 1 to run Ripley's analysis on localization:` -for object-based Ripley's analysis enter 0.

`Enter 1 to load a graph-based file, 0 to use an open file:` -If there is no file open in Grafeo, enter 1. For object-based Ripley's analysis, select and load graph-based process file as explained in Object analysis and co-localization section.

Ripley's function is evaluated as follows:

$$K(d) = VOL \sum_{i=1}^{n1} \sum_{j=1}^{n2} I_{d(i,j)} / (n1 * n2) \quad (1)$$

Where *VOL* is an area (volume) of studied region for 2D (3D) case. $I_{d(j,i)}$ is a number of points in a circular (spherical) neighbourhood with radius *d* from the i^{th} point in a point set 1 to the j^{th} point in a point set 2 for 2D (3D) case. The sum runs over all points in studied region. *N1* and *n2* is the number of points in set 1 and 2. Equation (1) can be used for univariate (point sets 1 and 2 are equal) and bivariate case.

When analysis is finished, you will be prompted to save a file manually. Input variables and the following output variables will be saved:

`Cr1/2` – a cell array containing randomized positions of the centroids/points in channel 1/2. The number of cells is equal to the number of the regions of interest ROI analysed and the size of each cell is `inPoint1/2` x 2 x `Nsimul`, where `inPoint1/2` is the number of points/centroids and `Nsimul` is the number of randomization iterations.

`cen1/2` – an array with the centroids in channel 1/2. The size is Nc1/2 x 2.

D – The vector of distances at which Ripley's functions is evaluated. The size is Nd, number of elements in `D` vector.

`inPoint1r/2r` – a vector with the number of points analysed at each iteration. It should be equal to `inPoint1/2`, however if the same position was randomly selected more then one time, only one instance of it will be used at given iteration, reducing the number of randomly sampled points.

`K12` – the bivariate Ripley's K function evaluated from the point set 1 to the point set 2. The size is Nd.

`K21` – the bivariate Ripley's K function evaluated from the point set 2 to the point set 1. For stationary point patterns, K12 = K21. The size is Nd.

`KdM` – is an estimator of the bivariate Ripley's K function, calculated as the linear combination of `K12` and `K21`: *`KdM` = (`lam2` * `K12` + `lam1` * `K21`)/(`lam1` + `lam2`)*, (see below for lam1/2). The size is Nd.

`KdMs` – is a matrix of KdM function evaluated at every randomization step (only point set 1 is randomized). The size is Nd x `Nsimul`.

`Khat1/2` – the univariate Ripley's K function evaluated for the point set 1 /2. The size is Nd.

`Ksim1/2` – is a matrix of `Khat1/2` function evaluated at every randomization step (the point set 1/2 is randomized). The size is Nd x `Nsimul`.

Ksim12/21– is a matrix of K12/21 function evaluated at every randomization step (only point set 1 is randomized). The size is Nd x Nsimul.

L12/21 – is a transformed bivariate Ripley's K12/21 function calculated as $L(d) = [K(d) /fact]^{1/n}-d$, where *d* is a distance and *fac* is π (3* π/4) and *n* is 2 (3) for 2D (3D) case respectively. The size is Nd.

Lhat1/2 – is a transformed univariate Ripley's K function (see L12/21). The size is Nd.

LdM– is a transformed bivariate Ripley's KdM function (see L12/21). The size is Nd.

LdMs – is a matrix of LdM function evaluated at every randomization step (only point set 1 is randomized). The size is Nd x Nsimul.

Lsim12/21 – is a matrix of L12/21 function evaluated at every randomization step (only point set 1 is randomized). The size is Nd x Nsimul.

Lsim1/2 – is a matrix of L1/2 function evaluated at every randomization step (point set 1/2 is randomized). The size is Nd x Nsimul.

vol – is an area (volume) of ROI for 2D (3D) case.

lam11/22 – is a density of points in a set 1/2 used to evaluate univariate Ripley's function K1/2. It is calculated as $(inPoint1/2)^2/vol$.

lam12/21 – is a density of points used to evaluate bivariate Ripley's function K12/21. It is calculated as $(inPoint1 * inPoint1)/vol$.

13.4. Two color single file localization-based Ripley's function analysis

Proceed as in Two color single file object-based Ripley's function analysis section. When prompted to populate dialog box, make sure you enter 1 in the previous to last line, Figure 15:

Enter 1 to run Ripley's analysis on localization: -for localizations-based Ripley's analysis enter 1.

In the last line: Enter 1 to load a graph-based file, 0 to use an open file: -select/ load any voronoi processed file which contains dataout1/dataoutshift1 and dataout2 variables.

Localization-based Ripley's analysis can be very time consuming. If you see similar message displayed in the Matlabs command line:

```
Error using pdist2mex
Requested 129300x129300 (124.6GB) array exceeds maximum array
size preference. Creation of arrays greater than this limit may
take a long time and cause MATLAB to become unresponsive. See
array size limit or preference panel for more information.
```

Try to reduce size of a ROI.

13.5. Loading Ripley's analysis data

Go to `Ripley's analysis` -> `Load randomized`. You will be prompted to select a file with saved Ripley's analysis data.

13.6. Plotting Ripley's analysis data

Go to `Ripley's analysis` -> `Plot randomized`. You will be prompted to select a variable you want to plot and chose the plot parameters, **Figure 16**:

`Enter x label` – the label for the x axis

`Enter y label` – the label for the y axis

`Enter x ticks` – tick on x axis to be displayed

`Enter y ticks` – tick on y axis to be displayed

`Enter 1 to scale x axis` – if you want to use x axis limits (see `Enter x ticks`)

`Enter 1 to scale y axis` – if you want to use x axis limits (see `Enter y ticks`)

Plots for all the ROIs appears.

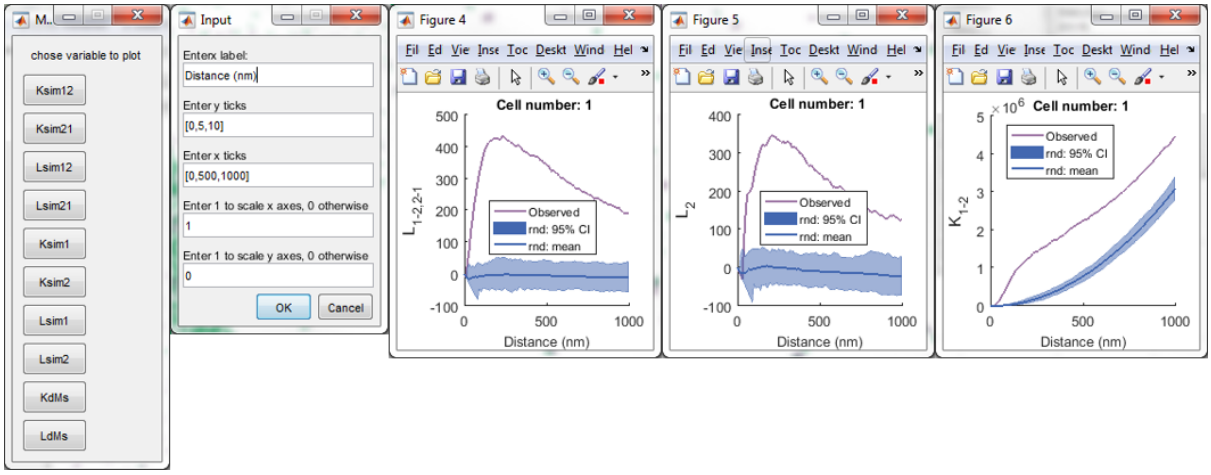


Figure 16. Plotting the Ripley's function

14. Bibliography

ANDRONOV, L., ORLOV, I., LUTZ, Y., VONESCH, J. L. & KLAHOLZ, B. P. 2016. ClusterViSu, a method for clustering of protein complexes by Voronoi tessellation in super-resolution microscopy. *Sci Rep,* 6**,** 24084.

DIXON, P. M. 2002. Ripley's K Function. *Encyclopedia of Environmetrics.*

HAAS, K. T., LEE, M., ESPOSITO, A. & VENKITARAMAN, A. R. 2018. Single-molecule localization microscopy reveals molecular transactions during RAD51 filament assembly at cellular DNA damage sites. *Nucleic Acids Res*.

LEE, D. T. & SCHACHTER, B. J. 1980. Two algorithms for constructing a Delaunay triangulation. *International Journal of Computer & Information Sciences,* 9**,** 219-242.

LEVET, F., HOSY, E., KECHKAR, A., BUTLER, C., BEGHIN, A., CHOQUET, D. & SIBARITA, J. B. 2015. SR-Tesseler: a method to segment and quantify localization-based super-resolution microscopy data. *Nat Methods,* 12**,** 1065-71.