



Protocol for multicolor three-dimensional dSTORM data analysis using MATLAB-based script package Grafeo

Kalina Tamara Haas, Alexis Peaucelle

► To cite this version:

Kalina Tamara Haas, Alexis Peaucelle. Protocol for multicolor three-dimensional dSTORM data analysis using MATLAB-based script package Grafeo. STAR Protocols, 2021, 2 (3), pp.100808. 10.1016/j.xpro.2021.100808 . hal-03767947

HAL Id: hal-03767947

<https://hal.inrae.fr/hal-03767947>

Submitted on 11 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

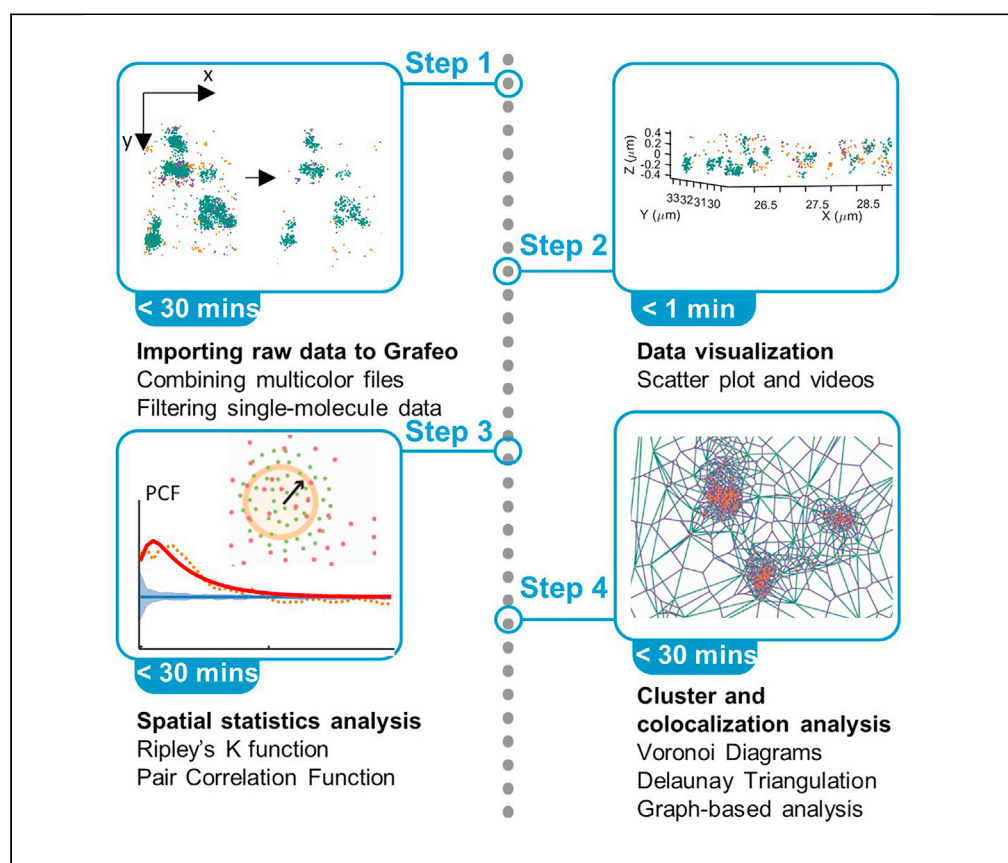
L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Protocol

Protocol for multicolor three-dimensional dSTORM data analysis using MATLAB-based script package Grafeo



This protocol describes the step-by-step analysis of the multicolor (one-, two-, or three-color) two- and three-dimensional dSTORM data using MATLAB-based script package Grafeo. Grafeo primarily uses pointillist data visualization and analysis frameworks, including the nearest neighbors, approach, Voronoi tessellation, Delaunay triangulation, Ripley's (K, L) and two-point correlation functions, and graph-based clustering.

Kalina Tamara Haas,
Alexis Peaucelle

kalina.haas@inrae.fr

Highlights

Multicolor 3D
dSTORM data
visualization using
scatter plots and
Voronoi Diagrams

Procedure for data
filtering

Spatial statistics,
cluster, and
colocalization
analysis

Graph-based cluster
analysis with
Delaunay
triangulation

Haas & Peaucelle, STAR
Protocols 2, 100808
September 17, 2021 © 2021
<https://doi.org/10.1016/j.xpro.2021.100808>



Protocol

Protocol for multicolor three-dimensional dSTORM data analysis using MATLAB-based script package Grafeo

Kalina Tamara Haas^{1,2,3,*} and Alexis Peaucelle¹

¹Institut Jean-Pierre Bourgin, INRAE, AgroParisTech, Université Paris-Saclay, 78000 Versailles, France

²Technical contact

³Lead contact

*Correspondence: kalina.haas@inrae.fr
<https://doi.org/10.1016/j.xpro.2021.100808>

SUMMARY

This protocol describes the step-by-step analysis of the multicolor (one-, two-, or three-color) two- and three-dimensional dSTORM (direct STochastic Optical Reconstruction Microscopy) data using MATLAB-based script package Grafeo. Grafeo primarily uses pointillist data visualization and analysis frameworks, including the nearest neighbors, approach, Voronoi tessellation, Delaunay triangulation, Ripley's (K, L) and two-point correlation functions, and graph-based clustering.

For complete details on the use and execution of this protocol, please refer to Peaucelle et al. (2020), Haas et al., 2018, Haas et al. (2020b).

BEFORE YOU BEGIN

This protocol is presented using the data obtained on the Nikon N-STORM microscope. You can download the example data and the analysis steps from Mendeley database (see the [key resources table](#)). However, we have also used this protocol for the data acquired with other dSTORM microscopes such as Vutara 350 (Bruker), Zeiss Elyra (Zeiss), and Abbelight, and the data analyzed with ThunderStorm Fiji plugin (Ovesný et al., 2014).

Computing setup

⌚ Timing: 5 min

1. Downloading and installation of Grafeo package:
 - a. Download the latest Grafeo script package (see [key resources table](#))
The current version of the Grafeo is 3 (v3).
 - b. Extract the Grafeo package and copy all files to the desired folder on your computer, e.g., "C:\Users\Documents\Programs\Matlab\Grafeo_v3".

Note: This protocol uses a Matlab desktop environment, which requires a license. You have to install Matlab before going to step c). To run Grafeo v3, you need to install the following Matlab toolboxes: Curve Fitting Toolbox, Image Processing Toolbox, Statistics and Machine Learning Toolbox, Signal Processing, and Optimization Toolbox.

Grafeo can also use a third-party function InPolygon that can be downloaded from Matlab file exchange (see the [key resources table](#)). For large polygons, InPolygon can perform faster than a build-in Matlab function inpolygon. However, the installation of InPolygon requires Matlab C



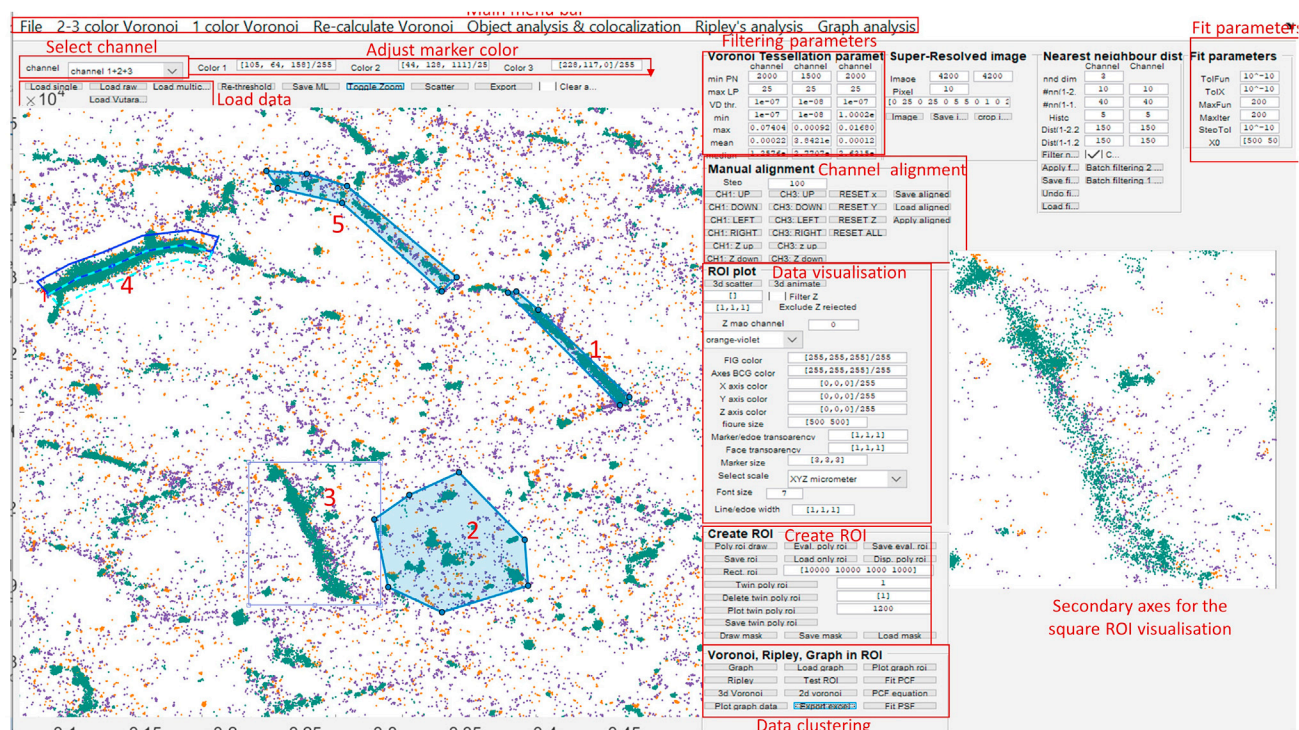


Figure 1. The Grafeo GUI window

Different sections discussed in the protocol are highlighted with red boxes. The main menu bar permits, among others, importing the raw single-molecule data in the different formats (here, only Nikon NSTORM 'txt' format is discussed). The raw dSTORM data is converted to Matlab ".mat" file format that can be loaded as a single color file and combined to a multicolor file (see "Load data" red box). The single-molecule data can be filtered by applying the threshold to photon number (PN), localization precision (LP), and the Voronoi diagram density (VD) (see "Filtering parameters" red box). Multicolor data can be aligned automatically (see the main menu bar, "2-3 color Voronoi") or manually (see "Channel alignment" red box). The data can be visualized using still, or animated scatter plots (see "Data visualization" box), Voronoi diagrams, and Delaunay triangulation (see "Data clustering" red box). Data visualization and analysis require prior creation of a region of interest (ROI) in the main axes. The different types of ROI can be drawn: polygonal or polygonal freehand ROI (No. 1, 2, 5), square ROI (No. 3), and twin ROI (Twin poly roi, No. 4, not used for the data visualization). The data can be analyzed using Ripley's K and L functions, point correlation function (PCF), and using Delaunay triangulation (graph-based segmentation) (see "Data clustering" red box).

compiler to compile mex files, and it does not work on the Windows 32 bits. In the absence of the InPolygon function, Grafeo uses Matlab built-in inpolygon function.

c. Open your Matlab desktop.

d. In the Matlab "Command Window" type: `addpath('fullpathToGrafeoFolder')`, using the example above, `addpath('C:\Users\Documents\Programs\Matlab\Grafeo_v3')`, and press "enter".

Opening Grafeo in Matlab.

e. In the Matlab "Command Window" type in `grafeo_v3`, and press "enter". This will open the Grafeo graphical user interface (GUI) (Figure 1). `grafeo_v3` is the name of the main Grafeo function. If you rename it or the version changes, use the update name, e.g., `grafeo_v4`, for the Grafeo version 4.

f. **Installing 'InPolygon' third party function.** First, check if you have a C language compiler compatible with Matlab installed on your machine. For this type mex -setup c in the Matlab command window. If you have, e.g., a free C compiler installed for Matlab (MinGW), the following message will appear: "MEX configured to use 'MinGW64 Compiler (C)' for C language compilation", where 64 stands for the Windows 64 bits. If no C compiler is installed, an error message will be displayed: "No supported compiler was found". Download MinGW compiler from the Matlab website (<https://fr.mathworks.com/support/requirements/supported-compilers.html>) and follow the installation instruction. Once the installation of C

compiler is completed, copy the downloaded InPolygon folder to the desired location on your computer and change the Matlab current directory (cd) by typing in the command window `cd("InPolygonDirectory")`, where "InPolygonDirectory" is a full path on your computer, e.g., "C:\Documents\matlab\InPolygon". Next, in the Matlab command window type `mex InPolygon.c`. Once the InPolygon is installed, the following message will appear: "Building with 'MinGW64 Compiler (C)'. MEX completed successfully."

Note: Step d has to be performed anytime a new Matlab session is started. Alternatively, in the Matlab "Home" tab, section "Environment", press "Set Path", select "Add folder," and browse to the Grafeo package folder. Press "Save". This way, Matlab will locate the Grafeo package automatically any time you start a new Matlab session. A simplified Grafeo manual is posted with each new package release (see [key resources table](#)). The extensive knowledge of MATLAB is not necessary to use Grafeo and to follow this protocol. However, we strongly recommend familiarizing oneself with the basic Matlab functionalities using online Matlab tutorials. Grafeo version 3 was developed and tested through Matlab 2019a to Matlab 2021a on both Mac and Windows 64-bit operating systems. The majority of Grafeo functionalities will work on Matlab 2017–2018. However, few functions used previously were replaced with new functions in the 2019 release (some examples are provided in the following sections). For better performance, Grafeo requires a computer with sufficient processor power and RAM. However, it was tested on a portable computer with an i5 1.70 GHz processor and 8 Gb of RAM. Higher specifications will increase the analysis time and speed up the visualization. Grafeo will run on any processor compatible with Matlab (<https://fr.mathworks.com/support/requirements/matlab-system-requirements.html>), including dual core (I3). However, its performance will depend on the generation of the processor. The minimum of 8 Gb of RAM is recommended.

Grafeo is under constant development; please consult the GitHub repository regularly to download the latest version.

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
Multicolor 3D dSTORM sample data sets	(Peaucelle et al., 2020)	https://data.mendeley.com/datasets/mj4hdpwyb8/1
Experimental models: organisms/strains		
<i>Arabidopsis thaliana</i>	Na	Na
Software and algorithms		
InPolygon third-party MATLAB function	Guillaume Jaquenot	https://uk.mathworks.com/matlabcentral/fileexchange/20754-fast-inpolygon-detection-mex
MATLAB	MathWorks	https://www.mathworks.com/products/matlab.html
Grafeo MATLAB script package	(Peaucelle et al., 2020) (Haas et al., 2018) (Haas et al., 2020b) (Haas et al., 2020a)	https://github.com/inatamara/Grafeo-dSTORM-analysis

MATERIALS AND EQUIPMENT

Spatial statistics

The dSTORM data can be considered as a spatial point pattern that consists of sets of coordinates of different types (colors/tags/epitopes). This allows the implementation of a multiscale spatial statistics functions to estimate the proximity, colocalization and spatial correlation for a multicolor dSTORM data sets, and domain or cluster size for each 'color' in the data set. The most commonly

used spatial statistics functions are Ripley's K function (and its derivatives) and Point Correlation Function (PCF), Figure 9. Both K and PCF are functions of the inter point distance d and can be calculated for the univariate (single color) or bivariate (two color) point patterns. The univariate Ripley's K function probes whether points are distributed randomly, are dispersed or cluster at a given scale in the studied area (ROI). The bivariate Ripley's K function determines whether two sets of points are distributed independently, segregate or aggregate at a given scale in the studied area (ROI).

For a distance d univariate (bivariate) Ripley's K function is calculated as a number of points of type 1 at the distance d or smaller from the point of type 1 (2), summed over all points (Figure 9A, top row). Thus, Ripley's K is a cumulative sum of the number of points in a circle with a radius d for 2D analysis, and in concentric spheres of a radius d for 3D analysis. Grafeo is using the following estimator for the univariate (single color) K function for a point pattern of type l (Dixon, 2014):

$$K_l(d) = (N\lambda_l)^{-1} \sum_i \sum_{j \neq i}^n w(l_i, l_j) I(d_{ij} < d) \quad (\text{Equation 1})$$

Where $I(d_{ij} < d)$ is unity if d_{ij} , a distance between i th and j th point, is smaller than d , otherwise it is 0. $\lambda_l = N/A$ for 2D and $\lambda_l = N/V$ for 3D is a point density, $n = N$ is a number of points in a point pattern l and A (V) is an area (volume) of the boundary B enclosing the data points. Here B corresponds to a ROI, which in 2D is a polygon and in 3D a polyhedron (a polygonal ROI with a thickness h), and w_{ij} is an edge correction factor. In 2D (3D), w_{ij} is calculated as an inverse of a fraction of a circle (sphere) with a radius d_{ij} , centered at a point i , which lies inside region B .

Grafeo is using the following estimator for the bivariate K function calculated for a point patterns of type l and k (Dixon, 2014):

$$K_{kl}(d) = (\lambda_k \lambda_l V)^{-1} \sum_k^m \sum_l^n w(k_i, l_j) I(d_{k,l_j} < d) \quad (\text{Equation 2a})$$

Where l and k denote different point patterns (colors or molecules tagged), λ_l and λ_k is a point density for pattern l and k respectively, d_{k,l_j} is a distance between i th point of type k to the j th point of the type l , n and m is a number of points in a point pattern l and k , respectively. The edge correction factor $w(k_i, l_j)$ is as defined above. Since $K_{kl}(d)$ is typically different from $K_{lk}(d)$, Grafeo is also calculating the following bivariate estimator:

$$K_{(kl, lk)}(d) = (\lambda_k K_{lk}(d) + \lambda_l K_{kl}(d)) / (\lambda_k + \lambda_l) \quad (\text{Equation 2b})$$

The observed Ripley's function is typically compared to what is expected if the points were distributed randomly (independently for bivariate distribution). Under the random distribution (univariate pattern) and independent distributions (bivariate pattern) the expected value of $K(d) = \pi d^2$ (area of a circle with a radius d) in 2D and $K(d) = 4\pi d^3/3$ (volume of a sphere with a radius d) in 3D. Thus, in practice, it is easier to interpret a transformed Ripley's K function, which for the univariate pattern in two dimensions (a) and the bivariate pattern in three dimensions (b, c) goes as follows:

$$L_l(d) = \sqrt{\frac{K_l(d)}{\pi}} - d \quad (\text{Equation 3a})$$

$$L_{kl}(d) = \sqrt[3]{\frac{3K_{kl}(d)}{4\pi}} - d \quad (\text{Equation 3b})$$

$$L_{(kl, lk)}(d) = \sqrt[3]{\frac{3K_{(kl, lk)}(d)}{4\pi}} - d \quad (\text{Equation 3c})$$

If at the distance d points are distributed randomly (independently), $L_l(d)$ ($L_{kl}(d)$, $L_{(kl, lk)}(d)$) is ~ 0 . If the points cluster (aggregate) $L_l(d)$ ($L_{kl}(d)$, $L_{(kl, lk)}(d)$) > 0 , and when points tend to disperse (segregate) $L_l(d)$ ($L_{kl}(d)$, $L_{(kl, lk)}(d)$) < 0 . If points cluster (aggregate) only at certain distances, $L_l(d)$ ($L_{kl}(d)$, $L_{(kl, lk)}(d)$) presents a peak corresponding to the clustering (aggregation) domain size. To test the departure

Table 1. Expected values for the univariate (bivariate) Ripley's K and L functions, and PCF for 2D and 3D data sets

Univariate Bivariate	Random Independent	Dispersed Segregated	Clustered Aggregated
K^{2D}	πd^2	$< \pi d^2$	$> \pi d^2$
K^{3D}	$4 \pi d^3/3$	$< 4 \pi d^3/3$	$> 4 \pi d^3/3$
L^{2D}	0	< 0	> 0
L^{3D}	0	< 0	> 0
PCF^{2D}	1	< 1	> 1
PCF^{3D}	1	< 1	> 1

from the random (independent) distribution Grafeo performs repetitive randomization steps of the experimental points to calculate a random (independent) distribution envelopes. If $L_i(d)$ ($L_{kl}(d)$, $L_{(kl,ik)}(d)$) lie within the randomization envelopes, the pattern is random (independent). If L_i (L_{kl} , $L_{(kl,ik)}(d)$) is greater/smaller than envelopes the patterns display clustering (aggregation)/dispersion (segregation) (Figure 9C, top row), see also Table 1.

The PCF function is a derivative of Ripley's K function. Grafeo estimates PCF as a difference between Ripley's K function at distances d and $d + dr$, normalized by the difference in an area (volume) of a ring (spherical shell) with radius d and the thickness dr for 2D (3D) analysis (Figure 9A, bottom row):

$$PCF(d) = \frac{K(d+dr) - K(d)}{\pi((d+dr)^2 - d^2)} \quad (\text{Equation 4a})$$

$$PCF(d) = \frac{K(d+dr) - K(d)}{4/3\pi((d+dr)^3 - d^3)} \quad (\text{Equation 4b})$$

Where Equation 2a (2b) corresponds to a 2D (3D) point pattern and can be applied to both univariate and bivariate point patterns. The univariate (autocorrelation function)/bivariate PCF estimate the increase in probability of finding molecule of the same/different type at a distance d to $d + dr$ as compared to the random/independent distribution ($PCF = 1$) (Figure 9C, bottom row). In contrast, K and L function are cumulative functions of the distance d , so the characteristics at the different spatial scales are combined.

Model fitting to ACF and PCF

Fitting the model to univariate PCF (ACF)

Grafeo permits fitting a model to the univariate PCF (autocorrelation function, ACF) that accounts for the over-counting of the same molecule and the final localization precision as described previously (Veatch et al., 2012, Sengupta et al., 2011, Sengupta et al., 2013):

$$ACF(r)_{tot} = \left(\frac{\delta(r)}{\rho} + \left(1 + Ae^{-\frac{r}{d}} \right) \right) \otimes ACF(r)_{PSF} \quad (\text{Equation 5})$$

The first term quantifies the correlation due to the multiple detection of the same fluorescent molecule, where $\delta(r)$ is a Dirac delta, and ρ the average density of molecules. The second term models the spatial organization of the molecule of interest (e.g., a protein) where d is a correlation length that estimates the cluster size and A is correlation amplitude that is proportional to molecules' density within a cluster. The $ACF(r)_{PSF}$ is an autocorrelation function of the point spread function (PSF) describing the uncertainty in localizing the fluorescence molecule, and \otimes is a convolution operator. The convolution of $\delta(r)$ with $ACF(r)_{PSF}$ quantifies the cluster-like appearance of a single fluorescence molecule, i.e., the spread from a single point ($r = 0$) to an area defined by the localization precision. Grafeo models PSF as a Gaussian function $PSF(r) = (\sigma\sqrt{2\pi})^{-1} e^{-\frac{r^2}{2\sigma^2}}$, so its autocorrelation function is also a Gaussian function with a sigma of $\sqrt{2}\sigma$, $ACF(r)_{PSF} = \frac{1}{4\pi\sigma^2} e^{-r^2/(4\sigma^2)}$. Thus, the first term in the (Equation 5) is given by:

$$ACF(r)_{single} = \frac{1}{4\pi\rho\sigma^2}e^{-r^2/(4\sigma^2)} \quad (\text{Equation 6})$$

Parameter estimation. The fit to (Equation 5) recovers the correlation length (domain size) d , and amplitude A . The two variable input parameters: (1) the average density of molecules ρ and (2) the average uncertainty in a single molecule localization σ should be estimated beforehand. The σ can be estimated as a mean of the localization precision. Alternatively, you can image the isolated secondary antibodies on the glass coverslip in the same imaging condition as your experiment. Calculate the univariate PCF (ACF) and fit (Equation 6) to recover σ , given the ρ . To estimate ρ , divide the mean (or median) Voronoi diagram density by the average number of detection within a cluster formed by the isolated secondary antibody. To do so, segment clusters as described in the section “graph-based cluster analysis” and calculate the mean density of clusters. For detailed estimation of σ and ρ for PALM (Photo Activation Localization Microscopy), refer to Sengupta et al. 2013 (Sengupta et al., 2013). The average number of molecules in the domain of a size d is calculated as follows: $N = 2\pi A d^2 \rho$. Note that most of the secondary antibodies have more than one fluorescence molecule attached. To improve the model fitting, use the secondary antibodies tagged with only one fluorescence molecule.

Fitting the model to bivariate PCF

The bivariate PCF is not influenced by the single molecule over-counting. Grafeo permits fitting the following model of the bivariate PSF (Sengupta, Jovanovic-Talisman and Lippincott-Schwartz, 2013):

$$PCF(r) = \left(1 + Ae^{-\frac{r}{d}}\right) \otimes C(r) \quad (\text{Equation 7})$$

Where $C(r)$ is a cross-correlation of the two point spread functions corresponding to two imaging channels and is given by the Gaussian function with the sigma $\sigma_c = \sqrt{\sigma_1^2 + \sigma_2^2}$, where 1, 2 correspond to imaging channels. The fit to (Equation 7) recovers the correlation domain d and amplitude A .

STEP-BY-STEP METHOD DETAILS

Loading “.txt” dSTORM molecular lists to grafeo

⌚ Timing: variable time, several to tens of minutes

This step describes how to upload a molecular lists (ML) file containing the single-molecule localization data. This is demonstrated on the ML in “.txt” format obtained from the Nikon NSTORM microscope. Nikon NSTORM outputs a “.txt” localization data file containing coordinates (x, y, z) for each localized molecule, along with other attributes used in the later steps, notably the localization precision (LP), the number of photons (PN), and frame index at which given molecule was detected. In this step Grafeo converts the data to Matlab “.mat” data file format that contains single-molecule features used for the subsequent data analysis and visualization. Next, we show how to combine individual channel “.mat” files to reconstruct a multicolor data file.

⚠ CRITICAL: Step 1

1. Importing the molecular list to Grafeo.

- In the main Grafeo menu bar press “File” -> “3D” or “2D” -> “single file” or “batch” -> “Import single Nikon txt molecular list”, Figure 2A.
- A pop-up window will be displayed where you should provide input parameters, such as a column index for the listed ML parameters (e.g., LP and PN), and the minimum PN and the LP, Figure 2B.

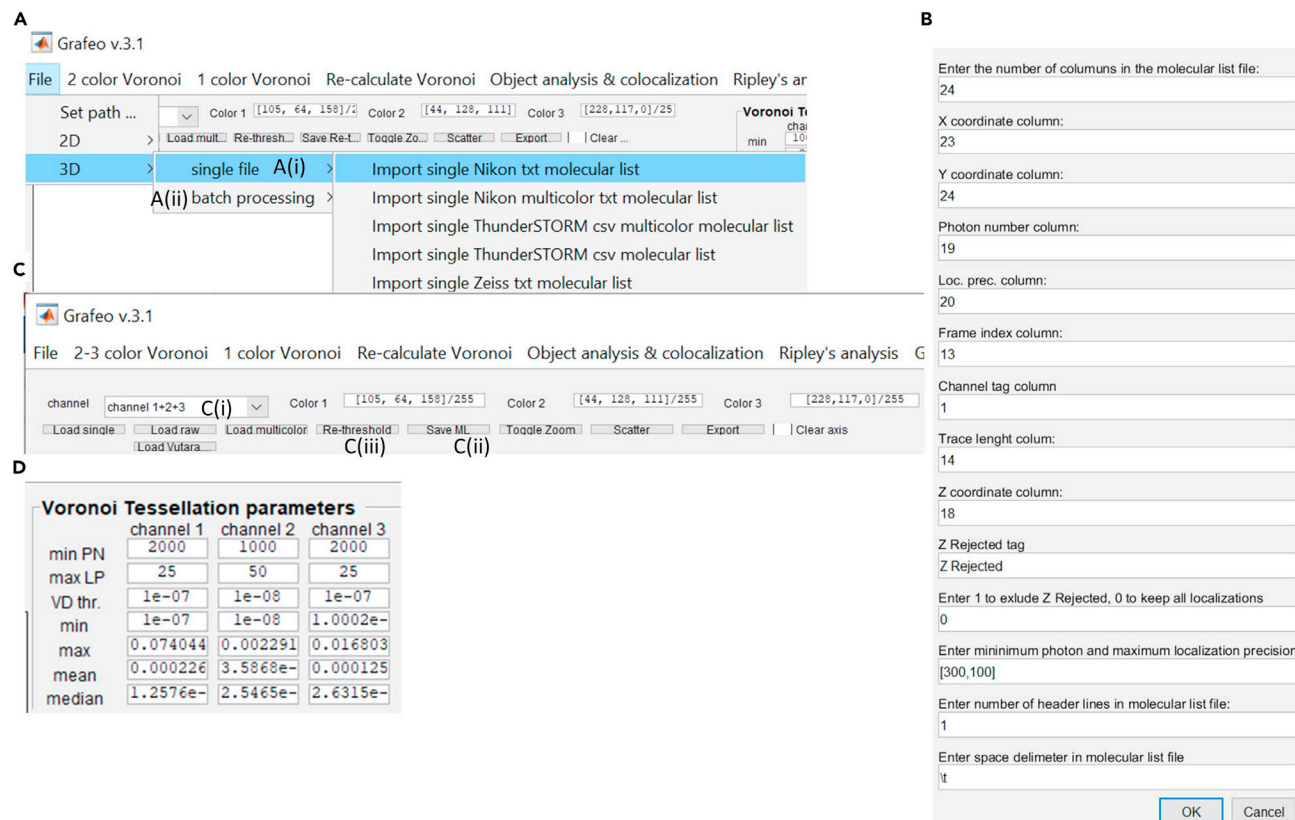


Figure 2. Data importation and filtering

(A) Importing a single file (A(i)) or multiple files in a batch processing mode (A(ii)).

(B) The data importation input parameters. From the top to bottom: (1) the total number of columns in a molecular list file, (2–3) the column numbers for the X (2) and Y (3) coordinate, (4) photon count, (5) localization precision, (6) frame index at which the molecule was detected, (7) column index with the channel tag (in the Nikon NIS elements software, the name for each channel can be set, e.g., 488, then the same name will be used in the molecular list file), (8) trace length (the number of subsequent image frames the single molecule appeared), (9) Z coordinate column, (10) a flag for the molecules for which Z position fit failed (in the Nikon file it is 'Z Rejected', and this tag replaces the channel tag), (11) a binary tag specifying whether to import all the data (set to 0) or only the molecules with a successful Z position fit (set to 1), (12) the two element vector specifying the minimum number of photons and the maximum localization precision (the molecules with fewer number of photons or greater localization precision will be discarded from the subsequent analyses), (13) the number of header lines preceding the column data, and finally (14) the file space delimiter (for Tab use '\t').

(C) Once the data is imported to the Matlab format, it can be loaded to the Grafeo memory. C(i) Channel selection dropdown menu, "Re-threshold" push button applies a new filtering parameters, "Save ML" saves the updated molecular list file.

(D) Data filtering panel using a minimum photon number (PN), a maximum localization precision (LP), and a minimum Voronoi diagram density (VD). The last four rows correspond to a minimum, maximum, mean, and median of the Voronoi diagram density for each channel and are populated automatically whenever the filtering parameters are changed or a new file is loaded.

Note: Once the data import and processing are finished, the output ".mat" file is automatically saved; its name is suffixed with "_Xwc_Ywc_pn_f_lp_Zc_l_zr.mat" that indicates variables stored in the file: y, x coordinates after wrap (w) transformation and chromatic corrections (c), photon number (pn), frame index (f), z coordinate after chromatic shift correction (Zc), trace length, that is the number of successive frames the molecule has appeared (l), and Z rejected binary tag (zr). See [troubleshooting 2](#).

Note: You can upload all ".txt" files in a given folder at once, including first-level subfolders, by choosing the "batch processing" option, [Figure 2A\(i\)](#). After the data is uploaded to Matlab, the corresponding Voronoi Diagram is calculated to link each localized molecule to a Voronoi polygon (VP). The local molecule density is estimated as an inverse of VP size and can be used to efficiently filter the data (see step 3). The zr value is 1 if the Z coordinate fit (refer to Nikon

manual) was successful given the calibration and the constraints on fit parameters. Usually, molecules with $z_r = 0$ have a fixed value (e.g., -400 for each of them) and should be rejected from further analysis. However, for a too strict parameter selection or incorrect calibration, a molecule may be tagged 0, while the Z position is appropriate. This could result in suppressing too many valid points.

△ CRITICAL: Step 2

2. **Combining multicolor molecular list.** This step shows how to combine single-channel ".mat" files generated in the step 1 into two/three-channel multicolor ".mat" files.
 - a. In the channel dropdown menu, select the option "Chanel 1", then press the "Load raw" pushbutton below, [Figure 2C\(i\)](#). You will be prompted to select a first (that you chose to be your channel 1, here A647 prefixed file) ".mat" file that was generated in the step 1.
 - b. Repeat the same steps by selecting the "Chanel 2/3" option in the selection box and loading channel 2 and 3 ".mat" files (CF568 and A488).
 - c. Once all three single-channel files are loaded to the Grafeo memory, select "Chanel 1+2+3" in the dropdown menu and then press "Save ML" to save a combined multicolor molecular list to a file with the desired name (here Corralign_647_568_488_2.mat), [Figure 2C\(ii\)](#). The single-channel files can be loaded in any order; here, we follow the acquisition sequence (i.e., 647, 568, 488).

Note: At this step no data filtering is applied, and filtering parameters (see step 3) that are saved to the file correspond to those chosen in the step 1 (PN, LP), and the minimum VD. Optionally, save a file only after the step 3 that is once the data is appropriately filtered. When using "Load raw" pushbutton, the data is added to the previous plot. If you want to clear the main axes, check the "Clear axes" checkbox located above the main axes in the right corner. Uncheck it once you load the first data file.

3. **Single-molecule data filtering with Voronoi tessellation, photon counts, and localization precision.** In this step, we show how to filter the data by adjusting localization precision LP, photon number PN, and the Voronoi diagram density (VD, an inverse of VP size) in a "Voronoi Tessellation Panel", [Figures 1](#) and [2D](#). The lower the LP, the more precisely single-molecule is localized. By decreasing LP and increasing PN and VD, more molecules are filtered out. A large VPs (small VD) usually corresponds to a noise. For the 3D dSTORM data filtering, start with the VD $\sim 10^{-8}$ (default), increase (decrease) the threshold to discard (include) more molecules. A good starting value for PN is ~ 1000 – 1500 , and for LP is ~ 25 – 50 nm. Once the parameters are set up, press "re-threshold" to filter the data, [Figure 2C\(iii\)](#). Save the filtered data using the "Save ML" pushbutton, [Figure 2C\(ii\)](#).

Note: The LP and PN values displayed in the "Voronoi Tessellation panel" are automatically changed whenever a new multicolor file is loaded ("Load multicolor" pushbutton). The LP and PN correspond to the minimum PN and the maximum LP found in the data after any filtering step, [Figure 2D](#).

4. **Multicolor channel registration.** In this step, we show how to perform the automatic channel alignment.
 - a. Before the alignment, it is recommended to apply a stringent threshold for the filtering parameters in the "Voronoi tessellation panel" (see step 3). The exact values depend on the experiment; we recommend a minimum photon number $PN > 3000$, maximum localization precision $LP < 20$ nm, and the Voronoi density $VD > 10^{-7}$.
 - b. After filtering, save the file using "Save ML" pushbuttons with the appropriate name, e.g., "Corralign_647_568_488_2_filtered.mat". You will need to load this file in the following steps.

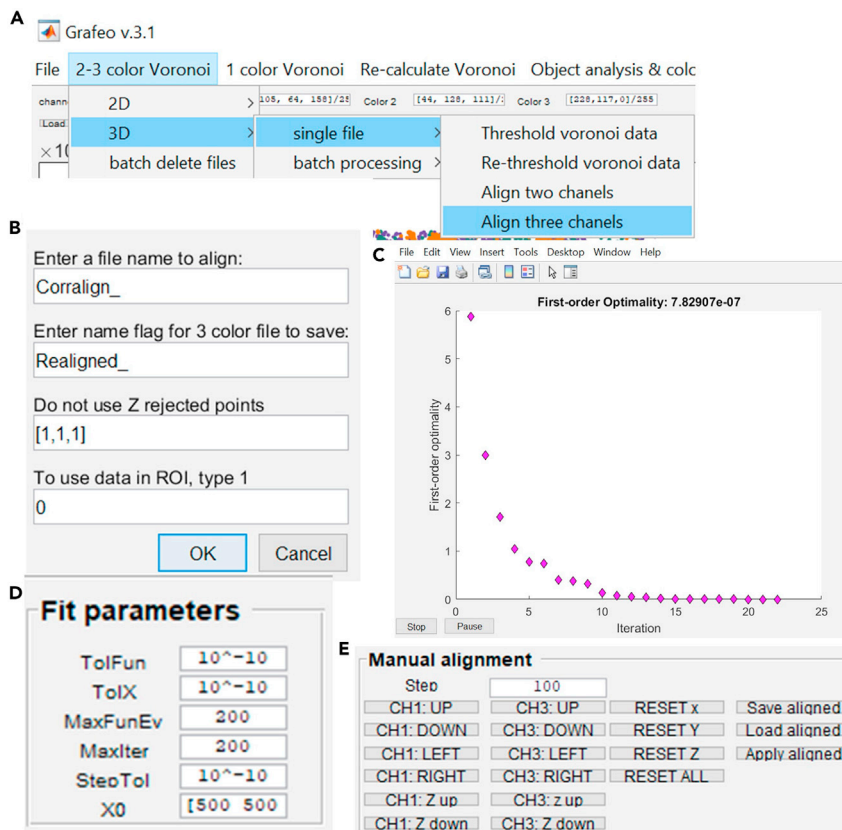


Figure 3. Automatic multicolor data alignment

(A) Selecting the option to align the three-channel data.
(B) The input parameter window for the data alignment, from top to bottom: (1) a prefix (or the first few letters) of the file that will be loaded and used for the alignment, (2) A prefix for the output file containing the results of alignment. The prefix from (2) will replace prefix from (1), and the remaining file name will be the same.
(C) The optimality plot displaying the alignment optimization steps.
(D) The Grafeo panel permitting setting up the optimization parameters.
(E) The Grafeo manual alignment panel. It also permits loading the alignment results ('Load aligned' pushbutton) and applying it to the currently opened data ('Apply aligned' pushbutton).

- c. To perform automatic alignment of the three-channel data, select "2-3 color Voronoi" -> 3D/2D -> "Single file" -> "Align three channels" in the main Grafeo menu bar, [Figure 3A](#). The pop-up window will appear where you have to provide the input parameters: a prefix of a filtered file name that you will be asked to select (here "Corralign_"), and the replacement prefix for the output file (here "Realigned_"), [Figure 3B](#). You should also specify whether to exclude "Z-rejected" points from the alignment analysis (recommended) for each channel separately (1 – do not use "Z rejected"). Press "ok".
- d. Next, you will be prompted to select the file that you want to align (here "Corralign_647_568_488_2_filtered.mat"). The automatic alignment will start.

The optimization progress is displayed both in the Matlab "Command Window" and as the first order optimality plot. The alignment procedure for a three-color dSTORM data consists of two steps: first, channel 1 is aligned to channel 2; in the second step, channel 3 is aligned to channel 2. For each step, a separate optimality plot will appear. The fit parameters can be adjusted in the "Fit Parameters" panel, [Figure 3D](#) (see also [Figure 1](#)). For better optimization results, it is recommended to use stringent values for TolFun, TolX, StepTol, i.e., a minimum of $\sim 10^{-10}$. At the end of each

optimization step, a message box will appear confirming the termination of the alignment procedure. The output aligned file is automatically saved (using a chosen prefix name, here "Realigned_").

Note: The alignment algorithm uses Matlab build-in function `fminunc` and minimizes the distance between two sets of points (channel 1 to channel 2, and then channel 3 to channel 2). To learn more, please refer to Matlab documentation on `fminunc` and the optimization parameters (`optimset`, `TolX`, `StepTol`, `FunTol`). The alignment can also be done on the fiducial markers or the subpart of the data (see the last parameter in [Figure 3B](#)), which substantially speeds up the alignment process. For this, either plot ROI (Region of Interest) around a fiducial marker (see next section to learn how to draw ROI in the main Grafeo axes), or filter the data by setting very stringent parameters, e.g., $VD \sim 0.01$, so that almost all the data except the fiducial markers are filtered out. Use the "Poly roi draw" pushbutton in the "Create ROI" panel (see below) to plot a polygonal ROI that would include mostly the fiducial markers or relevant data. In the current Grafeo version, only the latest drawn ROI is used. Thus, draw the ROI that contains several fiducial markers but avoiding the remaining data as much as possible. Future Grafeo releases should include multi ROI alignment. You can also place one rectangular ROI around one fiducial marker, but the fit will not consider the spatial inhomogeneity due to the optical aberrations. The alignment performed on an unfiltered data set can be lengthy. We recommend using ROI either around fiducials or around a subpart of the data. The ROI option is not available for the alignment in the batch mode.

5. **Apply alignment results to the currently opened data file.** At the end of step 4, the alignment results are saved automatically in a file with a prefix specified by the user (see above). This file contains a highly filtered data ("Realigned_647_568_488_2_filtered.mat").
 - a. Before applying the alignment to the currently loaded dataset, first adjust the filtering parameters, [Figure 2D](#).
 - b. To load and apply the alignment on the currently open file in Grafeo, go to the "Manual alignment panel", press "Load align" and selected the file with the results of alignment, and then press "Apply align" [Figure 3E](#). To refresh the plot, press "Scatter" in the panel above the main Grafeo axes.
6. **Manual alignment.** If the automatic alignment results are not satisfactory (or if you do not have Matlab optimization toolbox), you can perform the manual alignment. In the "Manual alignment" panel, select the step size (in nm), [Figure 3E](#). As in the automatic alignment, during the manual alignment, channel 2 stays unchanged. You can align channel 1 to channel 2 (column 1), and channel 3 to channel 2 (column 2), by pressing "Step UP", "Step DOWN", "Step LEFT", "Step RIGHT", "Z up", "Z down". Use the press button "Toggle Zoom" in the panel above the main axes to zoom on the data and evaluate XY's alignment. For the manual alignment in the Z dimension, you have first to draw a ROI (see below). When you press "Z up" or "Z down", a 3D scatter plot will appear, displaying the data in ROI.

Data visualization

⌚ Timing: variable, ~several minutes

This section describes how to draw different regions of interest and how to generate 2D/3D still and animated scatter plots using the data from the small ROI.

7. **Drawing polygonal ROI.** To draw a polygonal ROI, press "Poly ROI draw" in the "Create ROI" panel, [Figure 4A](#). You will be prompted to select one of two options: polygonal ROI (0) or free-hand ROI (1), [Figure 4B](#). Double click to draw the last vertex. For more details on how to draw polygonal and freehand ROI, consult Matlab documentation pages for the Matlab build-in functions "drawfreehand" and "drawpolygon". Only the latest drawn ROI is active. You can load a previously saved polygonal using "Load ROI". To display the loaded ROI press "Disp. Poly roi".

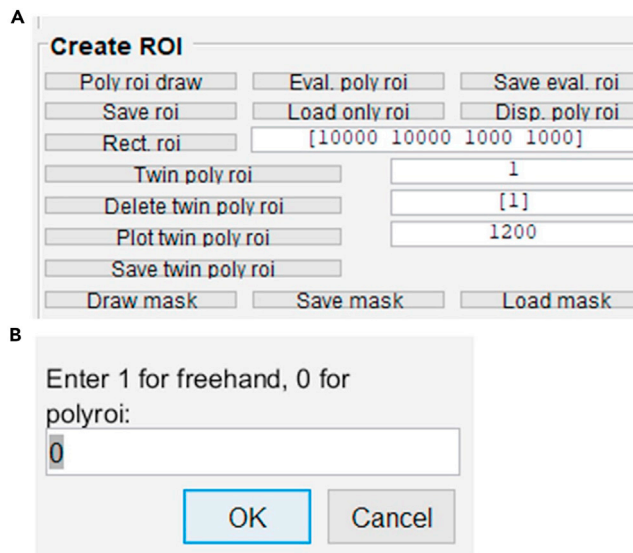


Figure 4. Creating the region of interest (ROIs)

(A) ROI panel permitting (1) to create ('Poly roi create' pushbutton), save ('Save ROI' pushbutton), load ('Load roi' pushbutton), and display ('Disp. Poly roi' pushbutton) polygonal ROI; (2) to create a rectangular ROI ('Rect. roi' pushbutton); (3) to create a twin polygonal ROI ('Twin poly roi', pushbutton), plot ('Plot twin poly roi', pushbutton), delete ('Delete twin poly roi', pushbutton), and save ('Save twin roi', pushbutton). For more details on the ROIs, see Figure 1 and the text.

(B) This window appears when you press 'Poly roi draw' and permits selecting between a freehand ROI and polygonal ROI (see text for more information).

Note: "Poly ROI draw" uses "drawpolygon" and "drawfreehand" that replaced the former Matlab functions (polygonal ROI, freehand ROI) and require at least Matlab 2019a version. If you have a previous Matlab version, use a rectangular ROI only.

8. **Drawing a rectangular ROI.** To draw rectangular ROI press "rect. ROI" located in the "Create ROI" panel. To adjust the initial rectangular ROI size and position, change the edit box parameters located next to the "rect. roi" button ([left position, bottom position, width, height]). Rectangular ROI is resizable and draggable.
9. **Drawing a twin polygonal ROI.** Twin polygonal ROI is composed of the two mirror-image polygonal ROIs. It permits analyzing the number of points in the two adjacent polygonal ROI, e.g., on the two sides of a cell wall or cell membrane.
 - a. First, zoom in to the plot area where you want to draw a ROI using "toggle zoom" pushbutton.
 - b. To plot twin polygonal ROI, press "Twin poly roi" and draw a line separating two regions. Double click to finish drawing a line. Mind the direction of the line drawing; it will define which ROI is tagged as the first. You can plot several twin poly roi. Each time you finish drawing a line, you will be asked if you want to draw a new line (press "Yes" to draw another line) or finish drawing the ROIs (press "No").
 - c. To visualize the drawn ROIs, press "Plot twin poly roi". To remove drawn ROIs, provide the ROI number (fill the edit box) and press "Delete twin poly roi". To check if a correct ROI was removed, refresh the main axes by pressing the "Scatter" pushbutton, and then press "Plot twin poly roi". Each time the twin polygonal ROI is displayed, the number of points in the two rectangular ROIs is calculated for each twin polygonal ROI. The width of the rectangular ROI can be adjusted (without need to draw a new roi) using the edit box next to the "Plot twin poly roi" push button (the number is provided in nanometers).

- d. To save a twin polygonal ROI and the analyzed data, press "Save twin poly roi". You can load a previously saved twin polygonal ROI using "Load twin poly roi".
10. **Visualizing the data in the polygonal and rectangular ROI.** To visualize the data in the polygonal or rectangular ROI using a scatter plot press the "3d scatter" pushbutton in the "ROI plot" panel [Figure 5A\(i\)](#). For the 3D data, you can rotate the plot manually or change the view by typing view(az, el) in the Matlab command window, where az is an azimuth angle ($0 < az < 180$) and el is elevation angle ($0 < el < 360$).
11. **Generating 3D animated scatter plot.** To animate your scatter plot, press "3D animate" in the "ROI plot" panel, [Figure 5A\(ii\)](#). You will be prompted to provide several input parameters, including initial and final azimuth and elevation angles, the number of rotation steps, and the output file format (".gif" or ".avi" are recommended), [Figure 5B](#).
12. **Adjusting figure properties.** To limit the data Z-axis range display, check the "Z filter box" checkbox and populate the "Z range" edit box that is located at the top of the "Plot ROI" panel, e.g., [-450,450], [Figure 5A\(iii\)](#). You can adjust the marker color by changing the edit boxes located above the main Grafeo axes (Color 1, Color 2, Color 3). Use the numbers between 0 and 255 for the red, blue, green (RGB), e.g., for green color type [0, 255, 0]/255, [Figure 2C](#). Additionally, you can use a colormap to represent the position of a fluorophore along the Z-axis. For this, in the "ROI plot" panel, first select the channel you want to map by typing 1, 2, or 3 in the "Z map channel" edit box (set to 0 for no mapping), and chose a predefined color map from a dropdown menu, or select "custom", to create your color map, [Figure 5A\(iv\)](#). You can also adjust the different figure display parameters: figure and axes background color (the default is white [255,255,255]/255), the axes color (the default is black [0,0,0]/255), marker size and transparency for each channel, line width (for Voronoi diagrams and Delaunay graphs plot), face transparency (for Voronoi diagram plots), and font size, [Figure 5A\(v\)](#).

Spatial statistics on the multicolor 2D/3D dSTORM data

⌚ Timing: variable time, few to tens of minutes

This section describes the spatial statistics functions, such as Ripley's (K, L) and Pair Correlation Functions (PCF) that permit evaluating spatial distribution and spatial correlation between biomolecules. These functions are commonly used to analyses spatial point patterns, such as 2D or 3D dSTORM datasets, composed of the point types (colors) and their coordinates ([Sengupta et al., 2011](#); [Veatch et al., 2012](#); [Sengupta, Jovanovic-Talisman and Lippincott-Schwartz, 2013](#); [Shivanandan et al., 2014](#); [Shivanandan, Unnikrishnan and Radenovic, 2016](#); [Nicovich, Owen and Gaus, 2017](#); [Peters et al., 2017](#), [Peters et al., 2018](#)). For the three-color data set, these functions are computed pairwise.

13. Ripley/PCF cluster analyses and colocalization.
 - a. First, draw a polygonal ROI as described in the "[data visualization](#)" section". The ROI should be a convex polygon. To test your ROI, press the "Test roi" pushbutton, [Figure 6A](#). If the blue points lie outside the red polygon, then the polygon is not convex (compare [Figure 6C](#), a concave polygon, randomized data fall outside of the ROI) with [Figure 6D](#) (a convex polygon, the randomized data fall inside the ROI). The polygonal ROI can be adjusted by dragging and repositioning its vertices. This can be applied only to the latest drawn polygonal ROI.
 - b. Press "Ripleys" in the "Voronoi, Ripley, Graph in ROI" panel. You will be prompted to provide several input parameters, as described in [Figure 6B](#).
 - c. When the operation is finished, you will be prompted to save the data to a file. See [troubleshooting 3](#).

Note: Ripley's analysis can be lengthy for a large dataset and many randomization steps. To reduce the computation time, you can: (1) select a small ROI that contain only the meaningful

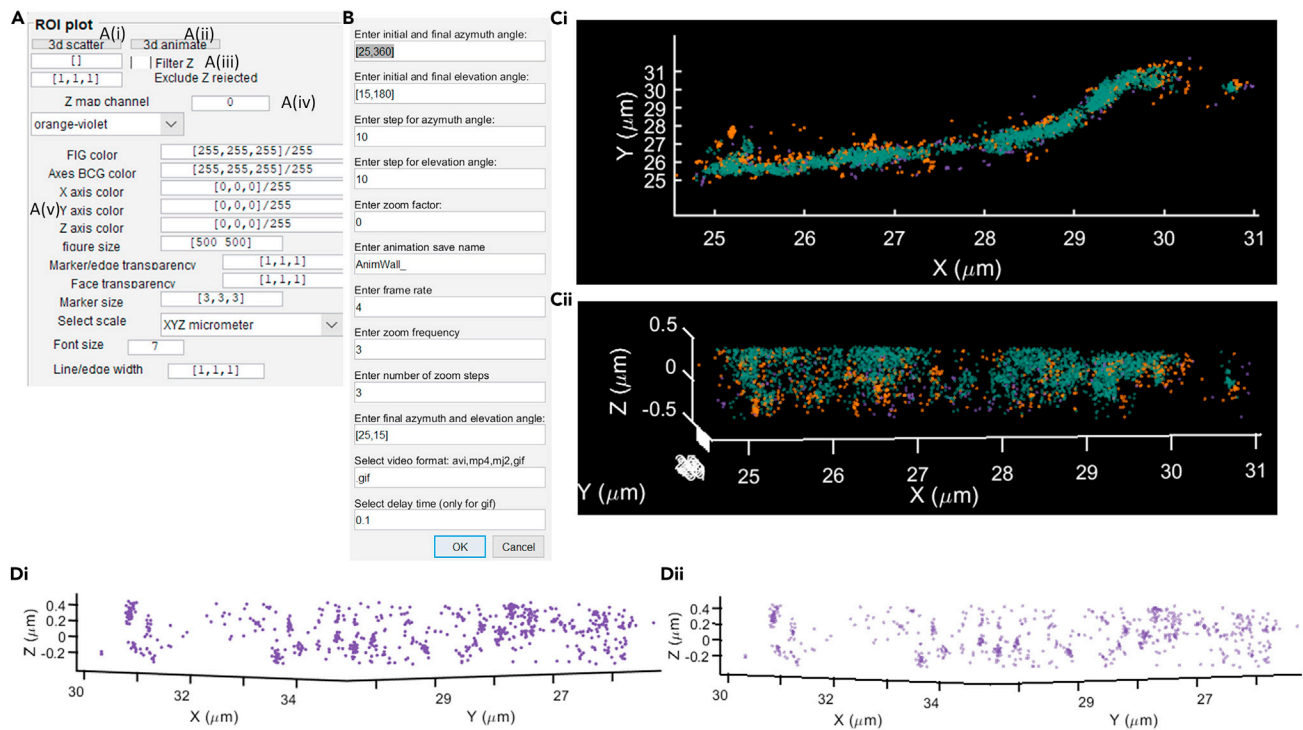


Figure 5. Displaying the data inside the region of interest (ROIs)

(A) The data inside ROI can be visualized as a still (A(i), "3d scatter") or animated (A(ii), "3d animate") scatter plot. A(iii) The Z-axis range of the displayed data can be adjusted. A(iv) For one of the channels (colors), a colormap to encode the Z position of a molecule can be applied. Select channel (1, 2, or 3) and the colormap. A(v) Setting up the figure properties, such as marker color, transparency, and size, axes color, the figure, and axes background color. (B) The input parameters window for the animated scatter plot. You have to provide the following parameters, from top to bottom: (1) the azimuth and (2) the elevation angle range (initial and final angles), the number of rotation steps for the (3) azimuth, and (4) elevation angle, (5) the zoom factor, (6) the final video file name, (7) the video frame rate (for '.avi', and '.mp4' formats), (8) the zoom frequency, and (9) the number of steps to achieve the zoom factor, (10) The final azimuth and elevation angle, (11) the output file format, (12) the delay time ('.gif' format only, equivalent to the frame rate). (C) A still scatter plot of a three-color data; C(i) a top (X, Y) 2D view, and C(ii) side (X,Y,Z) 3D view (see also [Methods video S1](#)). (D) The marker transparency can be used to enhance the clustering. D(i) Scatter plot with transparency 1 (completely opaque points), and D(ii) with the transparency set to 0.6.

data (i.e., the data you want to analyze), (2) filter the data by adjusting photon counts (PN), localization precision (LP), and Voronoi density (VD) thresholds in the "Voronoi Tessellation parameters" panel. Before running longer randomization, first, simulate with ~5 randomization steps, and verify the K, L, and PCF plots. By default, Grafeo performs randomization, since the spatial statistics functions are meaningless without appropriate comparison to a null hypothesis (here, the randomized data that represent independent (bivariate functions) or random (univariate functions) distributions).

14. **Visualizing the spatial statistics functions.** To visualize the output Ripley's K, L, or PCF functions, go to the main Grafeo menu bar and select "Ripley's analysis" -> "Plot randomized", and choose the variable that you wish to plot (univariate/bivariate Ripley's K, L, or PCF), press "ok" and then select the figure display parameters, [Figures 6E](#) and [6F](#). The plot shows experimental (observed) function (e.g., L, K, PCF) as a solid violet line overlaid on the 5% and 95% randomization envelopes (blue shaded area) and the average function overall randomization steps (blue line), [Figures 8B–8E](#), [8H](#), and [8J](#).

See [troubleshooting 4](#)

15. **Fitting the model to the univariate and bivariate PCF functions.** The current version of Grafeo (v3.1) permits fitting the model functions to univariate (PCF_x or an autocorrelation function ACF) and bivariate (PCF_{xy}) point correlation function ([Sengupta et al., 2011](#); [Veatch et al., 2012](#);

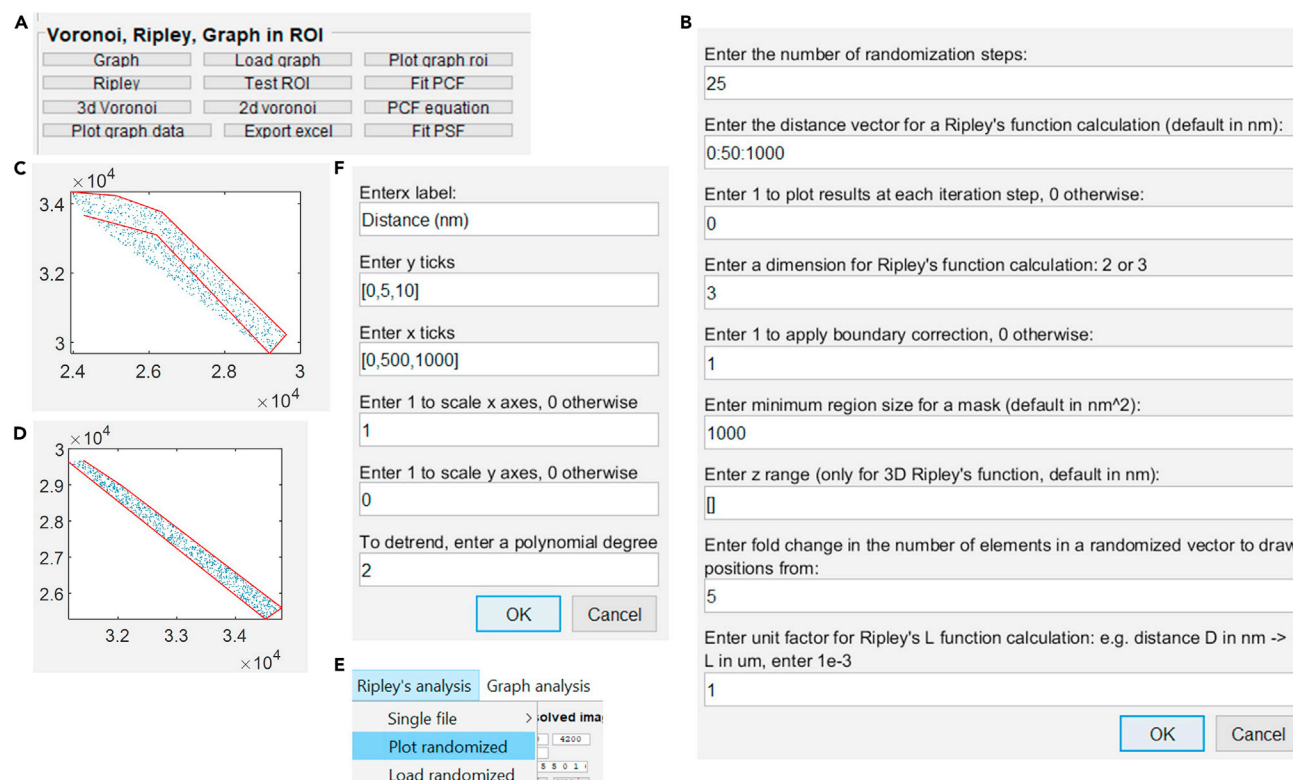


Figure 6. Data clustering, colocalization, and spatial point pattern analysis

(A) The data in the cluster analysis selection panel.

(B) The input parameters window for Ripley's analysis ('Ripley' pushbutton in A), from top to bottom: (1) the number of randomization steps, (2) a distance-vector to evaluate Ripley's and PCF functions (see Matlab manual for the vector notation), (3) a flag to plot (set to 1) or not (set to 0) the data for each randomization step, (4) A dimension for the Ripley's and PCF functions calculations (2 or 3), (5) a flag to apply (set to 1) or not (set to 0) boundary conditions to Ripley's and PCF functions calculation, (6) this parameter will be discontinued in the subsequent releases, (7) the Z range of the data to calculate the Ripley's and PCF functions, (8) The fold increase in the number of elements in a position vector from which randomized positions will be drawn (with respect to the number of data points).

(C and D) The randomized data plot for concave (C) and a convex (D) polygonal ROI.

(E) A zoom in on the Grafeo main menu bar and the selection to plot Ripley's and PCF functions.

(F) The input parameters for the Ripley's and PCF function plots, from top to bottom: (1) The X-axis label, (2) The X-axis tick position. (3, 4) A flag to constrained the plot (set to 1) or not (set to 0) to the minimum and maximum data value for the X-axis (3) and Y-axis (4), (5) A polynomial degree to remove the polynomial trend from the data (detrending) Ripley's and PCF function (typically 2 and 3), this affects only the data display.

Sengupta, Jovanovic-Talisman and Lippincott-Schwartz, 2013). To perform the fit, press "Fit PCF" push button located in the "Voronoi, Ripley, Graph in ROI" panel.

- a. **Selecting the model.** You will be prompted to choose a variable that you want to fit a model to, Figure 7A. To fit ACF chose "1 Gauss + 1 exGauss" option and to fit PCF_{xy} chose "1 ex-Gauss", Figure 7B.
- b. **Estimating the variable input parameters.** The localization uncertainty (σ) and the average molecule density (ρ) should be estimated prior to fitting (see "model fitting to ACF and PCF" in the "materials and equipment" section). The uncertainty σ can be estimated from the localization precision. However we recommend to perform the fit (see step 15b,i below).
 - i. **Fitting the ACF model to Point Spread Function (PSF).** Grafeo permits fitting the model to recover σ . This requires dataset where the molecule of interest is distributed randomly and does not form any clusters with more than one molecule. We recommend imaging the secondary antibodies on the cover glass in the same imaging conditions as your experiment. First, calculate the PCF as described in the step 13 or load the previously obtained data ("Ripley's analysis" -> 'Load randomized'). To fit the model press "Fit PSF" pushbutton

Figure 7. Fitting a model to PCF functions

(A) A selection panel for the PCF model fitting.

(B) A selection panel for the fit type, select '1 Gauss + 1 exGauss' and '1 exGauss' for univariate and bivariate PCF, respectively.

(C) The input parameters for PCF model fitting, from top to bottom: (1) the X range of fit in the form of element vector (1 – the first, 50 – the fifties element of the distance vector at which the PCF was calculated). (2–4) The lower (2) and upper (3) bonds and (4) initial fit parameters. To see the fit equations, press the 'PCF equation' pushbutton. (5) The estimated localization uncertainty. For the univariate PCF, only the first element is used. (6) The X-axis label. (7–8) The Y (7) and X-axis (8) ticks. The first and the last element are also applied as the axes limits. (9) Polynomial degree for the polynomial detrending of PCF/K/L functions, typically 2 or 3.

located in the "Voronoi, Ripley, Graph in ROI" panel. Note, if you fit the model on the isolated secondary antibodies, the fit parameter ρ will correspond to the average density of your antibodies and not of your molecule of interest. To estimate ρ for your molecule of interest, divide the mean VD for your molecule of interest by the average number of detection within a cluster formed by the isolated secondary antibody. To recover the mean detections per antibody, segment clusters generated by isolated antibodies as described in the section "[graph-based cluster analysis](#)". Use the mean VD over all image or in the small ROI (see "[graph-based cluster analysis](#)" step 16 for more details).

- c. **Providing the initial values.** A pop-up window will be displayed, and you will have to provide the input parameters, such as the initial value for the fit parameters (x_0), and their upper (UB), and lower bonds (LB), [Figure 7C](#). Grafeo fits three parameters: the average molecule density – ρ , correlation amplitude – A , and the correlation length – d . However, it is strongly recommended to fit only second (A) and third (d) parameter (second and third element of the UP, LB, and x_0 parameters) and to constrain the first parameter (ρ) by setting UP, LB, and x_0 to the estimated density of molecules (see step above).

Note: The quality of the fit can be susceptible to x_0 , UB, and LB. Test different initial parameters. After the fit is completed, you will be prompted to save the data. The figure showing fit results will be displayed automatically, [Figures 8F, 8G, 8I, and 8K](#), where the fitted function is displayed as a solid red line.

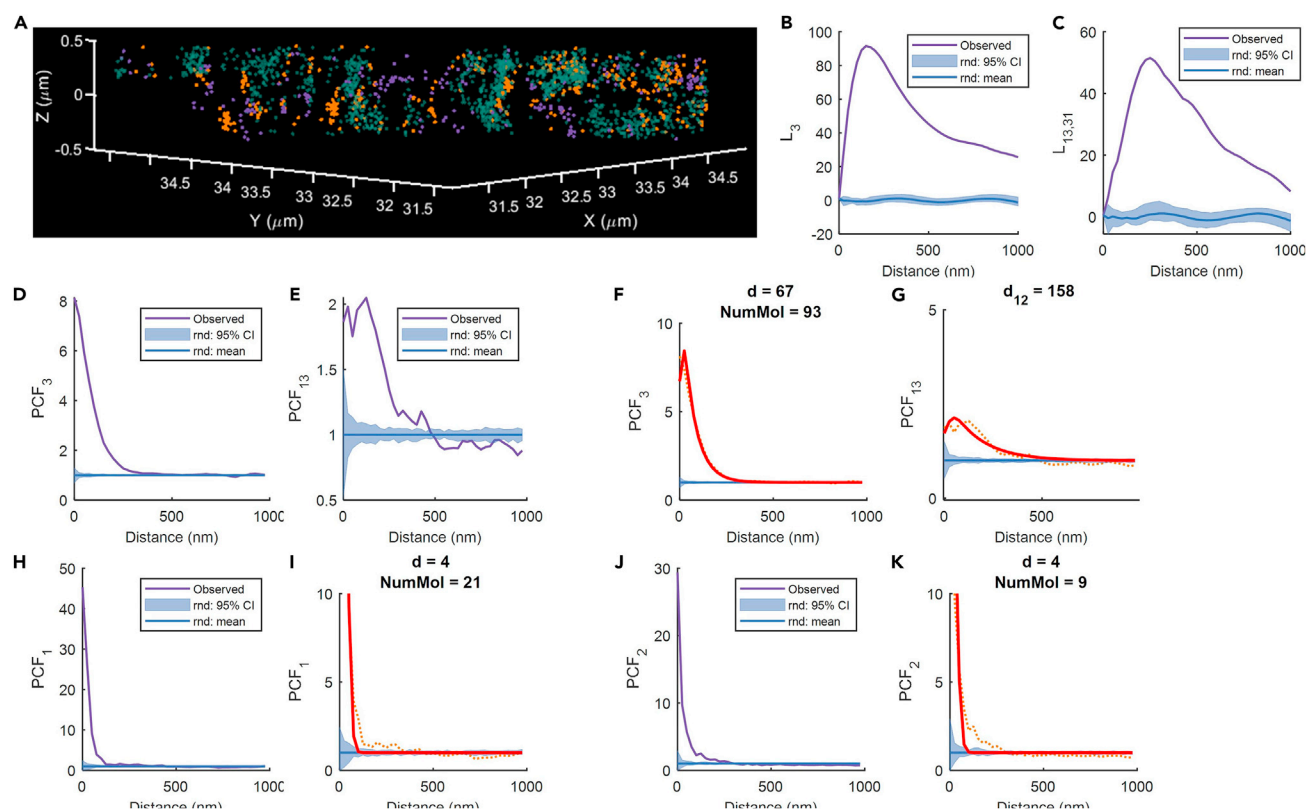


Figure 8. Plotting the Ripley's and PCF function

(A) A 3D scatter plot of the data for which the subsequent plots were generated. Channel 1 – violet (CBM3), channel 2 – green (Callose), channel 3 – orange (Hetero-mannans) (see also [Methods video S2](#)).

(B) The univariate transformed Ripley's function (L) for the third channel.

(C) The bivariate transformed Ripley's function (L) calculated between the first and the third channel.

(D) The univariate PCF function for the third channel.

(E) The bivariate PCF function is calculated between the first and the third channel.

(F and G) The same as (D-E) but displaying the fit to PCF (red line).

(H) The univariate PCF function for the first channel.

(I) The same as (H) but displaying the fit to PCF (red line).

(J) The univariate PCF function for the second channel.

(K) The same as (I) but displaying the fit to PCF (red line).

See [troubleshooting 5](#).

Note: For detailed information about model fitting and the description of parameters, please refer to ([Peaucelle, Wightman and Haas, 2020](#)). To check the equations for "1 Gauss + 1 ex-Gauss" and "1 exGauss", press "PCF equation" pushbutton.

Graph-based cluster analysis

⌚ Timing: variable time, few seconds to tens of minutes

This section describes undirected graph-based cluster analysis on the data in the active ROI, [Figure 10](#) ([Shivanandan et al., 2014](#); [Levet et al., 2015](#); [Khater, Nabi and Hamarneh, 2020](#)). The graph is computed using 2D or 3D Delaunay triangulation converted to an undirected graph consisting of nodes (data points) connected by undirected edges (Delaunay triangulation DT edges) ([Yang and Cui, 2010](#)). The graph-based analysis permits segmenting objects (clusters) by thresholding the maximum inter point distance (ipd, or graph edge length), the minimum number

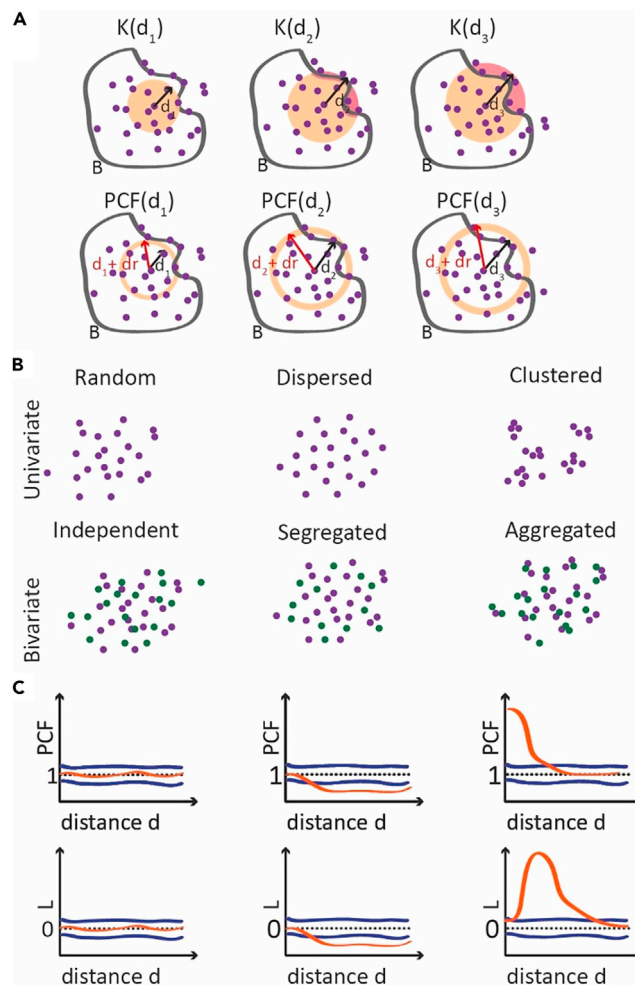


Figure 9. Ripley's and PCF functions

(A) A cartoon representation of the Ripley's K (top) and PCF (bottom) functions calculation. Ripley's K is calculated as an estimated number of points in a neighborhood of a randomly selected point located at a distance not greater than d . PCF is a derivative of K, and is estimated as a number of points in a neighborhood of a randomly selected point located at a distance smaller than $d+dr$ and larger than d . B is a boundary (ROI) enclosing the data set; for large distances d , the circle/sphere overhangs the boundary B introducing a bias in the calculation of K. This can be corrected by the appropriate boundary correction factor (see the text).

(B) A schematic representation of the three classes of bivariate (univariate) point patterns that can be detected with K and PCF analysis.

(C) Schematic representation of PCF and transformed Ripley's L function for the three classes of point patterns showed in B). Schematic plots are the same for univariate and bivariate L/PCF functions.

of points in a cluster (area, or the number of detections/points), and a node degree (the number of connections with other points). These functions are located in the panel "Voronoi, Ripley, Graph in ROI", and can be applied to a 2D or 3D dataset. For three colors, the functions are computed pairwise.

16. Segmenting the clusters.

- First, draw an ROI as described in the "data visualization" section.
- Then press "Graph". You will be prompted to provide several input parameters, Figure 11, notably, a minimum number of points in a final segmented graph, a maximum inter point distance, ipd (a maximum DT edge length), and the nodes degree (a minimum number of

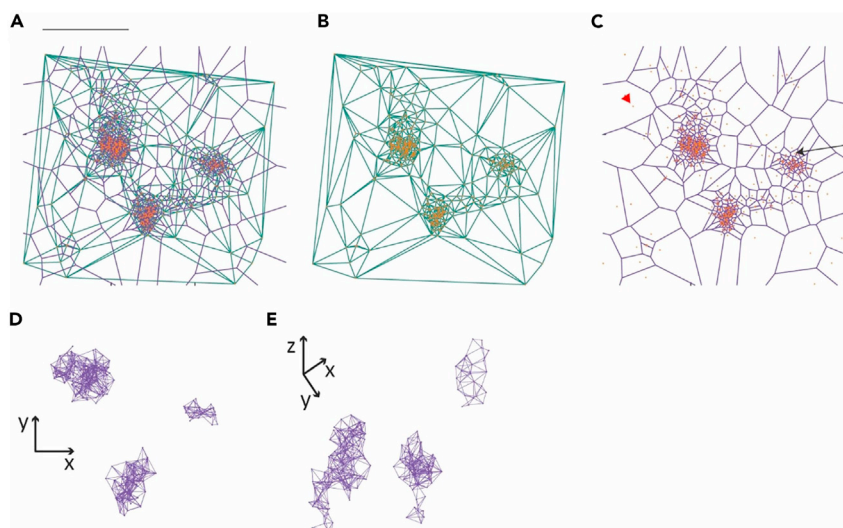


Figure 10. Voronoi diagrams and Delaunay triangulation

(A–C) (A and C) Voronoi diagram (violet) and (A and B) Delaunay triangulation (green) calculated for a set of points (orange). In (C) a red arrowhead indicates a point associated with a large Voronoi polygon (dispersed data, noise) and black arrow shows a point associated with a small Voronoi polygon (dense, clustered data). This propriety is implemented to efficiently filter the data based on the Voronoi polygon size and permits separating dense clustered data from the noise or dispersed points.

(D and E) The results of graph segmentation in lateral xy view – (D), and in 3D view – (E).

connections with other points). For a dense cluster (high VD), the node's degree is typically high. Good starting values are minimum 3–5 detections, $nd \sim 2-3$, $ipd \sim 50$ nm. Graph-based analysis can be calculated using 2D or 3D Delaunay triangulation. The maximum colocalization distance (valid only for multicolor data) is a maximum distance separating segmented clusters from two channels. If a segmented graph (cluster) has no neighbor at this distance or smaller, it is removed from the analysis. Thus, choose a high colocalization distance to retain all the segmented clusters. Graph-based data analysis is performed on two channels at a time. Thus, the analysis has to be performed pairwise. For three-color data, select the channels you want to run the analysis for (the last parameter in [Figure 11](#)).

- c. Once the segmentation is completed, you will be prompted to save the results to a file. If no clusters were segmented in at least one channel, the file would not be saved, and you will have to adjust the segmentation parameters.

See [troubleshooting 6](#) and [7](#).

Note: Cluster analysis can be lengthy for large data sets. Before running the analysis, try first testing small ROI to adjust the segmentation parameters.

17. **Graph visualization.** To visualize the segmented graphs, press “Plot graph roi” pushbutton located in the “Voronoi, Ripley, Graph in ROI”, [Figures 6](#) and [12](#). If the data is 3D and the graph segmentation was performed in 3D, you can rotate the Matlab plot. You can control the figure display by adjusting parameters in the “ROI plot”, e.g., marker color and transparency, edge/line width and transparency.

See [troubleshooting 8](#).

Note: You can load previously analyzed graph data using the “Load graph” pushbutton. It will also load ROI that was used for the graph segmentation.

18. **Voronoi Tessellation visualization.** To plot 2D or 3D Voronoi diagrams from the data in a small ROI, press “3d Voronoi”. This function is not performing a full cluster segmentation. Thus, the

Enter minimum # of points for color 1, 2 and 3:
[2,2,2]

Enter maximum Delaunay edge length for color 1, 2 and 3:
[50,50,50]

Enter graph node degree for color 1, 2 and 3:
[1,1,1]

Enter dimension for Delaunay triangulation: 2 or 3
3

Enter maximum colocalization distance:
200

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

Enter number of nearest neighbours for nearest neighbour distance estimation:
1

Select channels for two color analysis:
[1,2]

OK Cancel

Figure 11. Performing the graph-based segmentation

The input parameters for the graph-based object segmentation. From top to bottom: (1) The minimum number of points for the final segmented graph. (2) The maximum edge length (the same as interpoint distance, ipd). (3) A node degree, i.e., a minimum number of connections with other points (nodes) in a final segmented graph. (4) The dimension for Delaunay triangulation calculation, 2 or 3 (only for the 3D data). (5) The maximum distance between a segmented graph of the first color and the nearest segmented graph of the second color. (6) Select the mean or median to calculate the cluster density (by default it is median). (7) The number of the nearest neighbors for the colocalization analysis. (8) The channel selection for the analysis; type [1,2], [1,3], or [2,3].

Voronoi diagram may be continues. However, it lets you adjust the minimum/maximum displayed VD to suppress the large/small VPs and to disconnect the clusters, [Figure 13](#). When using "3d Voronoi" function, the average VD in a ROI will be displayed in the Matlab command window both before and after suppressing the large/small VPs. The last input parameter permits the control of the individual VP plot, as either calculated VP or the convex hull of a VP (the smallest convex polygon/polyhedron in 2D/3D that enclose the vertices defining a Voronoi Polygon). The pushbutton "2d Voronoi" displays the 2D VP (works only for 2D data) using the colormap and transparency scaled by the VD.

19. **Object properties and colocalization visualization.** To visualize the graph and colocalization data, press the "Plot graph data" pushbutton. You will be prompted to select a variable you want to visualize (see below for the variable description), [Figure 14A](#). Next, you will be prompted to provide the input parameters for a histogram plot, [Figure 14B](#). Insert '[]' (empty vector) in the two first rows if you do not know the parameter range (i.e., a minimum and a maximum value). Press "ok"; a figure will display showing mean and median values of the chosen variable. The x-axis specifies the chosen variable name.
20. **Exporting the data to excel file.**
 - a. **Exporting the cluster (graph) properties.** To export the data to excel file press "Export excel" pushbutton located in the "Voronoi, Ripley, Graph in ROI" panel, [Figure 6A](#). You will be prompted to select whether you want to save only the cluster properties data, the colocalization analysis or both. The segmented clusters properties that are saved to an excel file are listed in [Table 2](#). The data for each channel is saved to a separate excel sheet. The name of output file has the following structure: 'filename_dim_nd_ROIgraph_ij.xls', where the filename is the name of the '.mat' file currently loaded in the Grafeo memory, nd is the

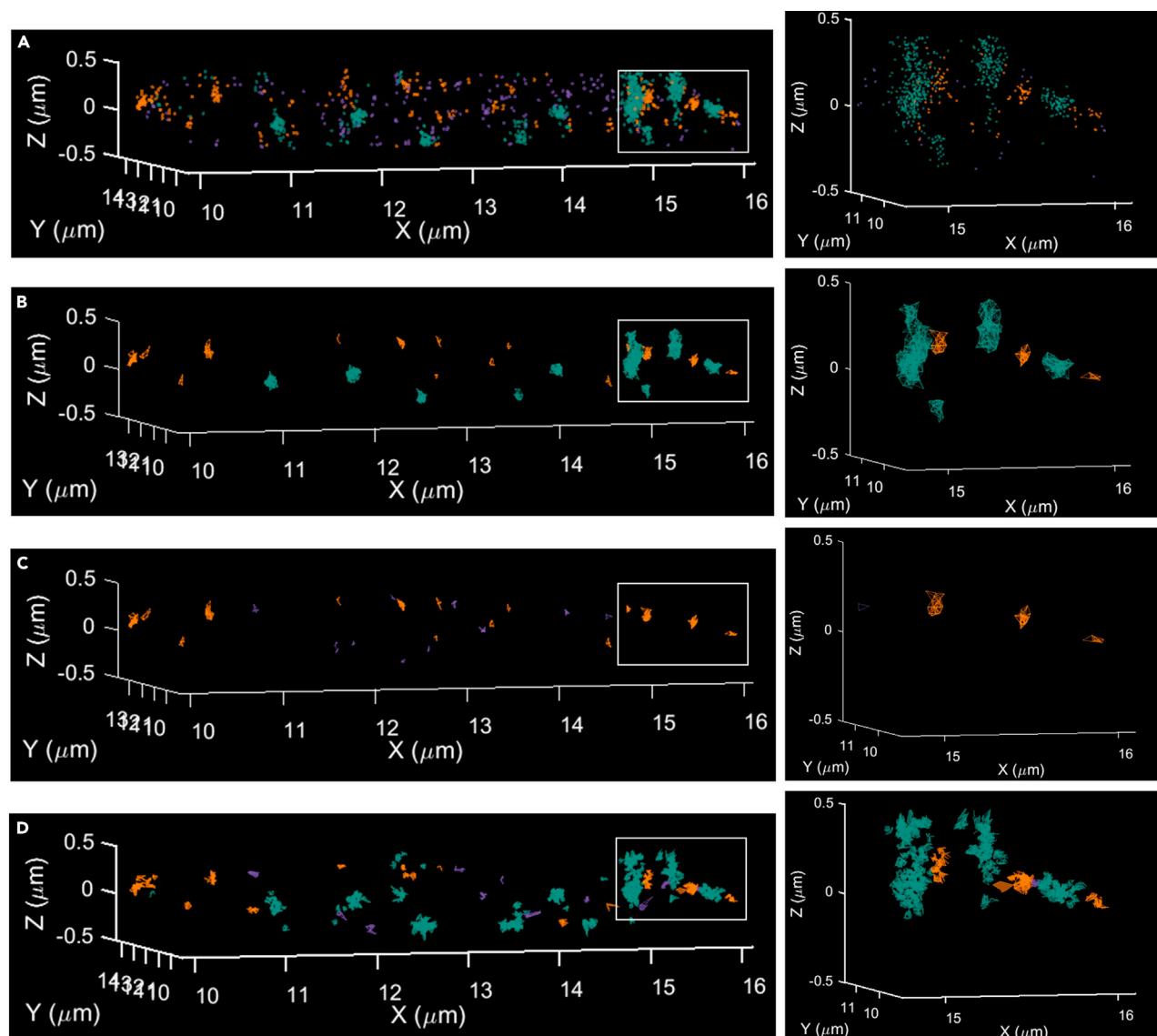


Figure 12. Visualizing the segmented clusters

(A) (left) A 3D scatter plot of the data subject to data clustering. Channel 1 – violet (CBM3), channel 2 – green (Callose), channel 3 – orange (Heteromannans). (right) The close up on the data enclosed by the white rectangle.

(B and C) The result of the graph-based cluster segmentation for B) channels 2 and 3, and C) channels 1 and 3. The data is displayed as the 3D graphs (3D disconnected Delaunay triangulation). (right) The close up on the data enclosed by the white rectangle.

(D) The data displayed as the 3D Voronoi polygons. The Voronoi diagram was disconnected by applying the upper and lower bounds for the VD Voronoi diagram density (see also [Figure 13](#)).

Delaunay triangulation dimension (2 or 3), and i, j are two channel tags ($i, j = 1, 2$, or 3). For one channel analysis only, $i = 1, 2$, or 3 .

- b. **Exporting the colocalization analysis.** To export the colocalization analysis data to excel file press "Export excel" push button and chose second or third option. The colocalization analysis data saved to an excel file is listed in [Table 3](#). Each data structure/matrix is saved to a separate excel file. The nearest neighbor point to point distance matrix and centroid to centroid distance matrix are save to a file with the following structure: 'filename_dim_nd_ROIgraph_P2P_ij.xls', and 'filename_dim_nd_ROIgraph_C2C_ij.xls' respectively, where i, j is as above.

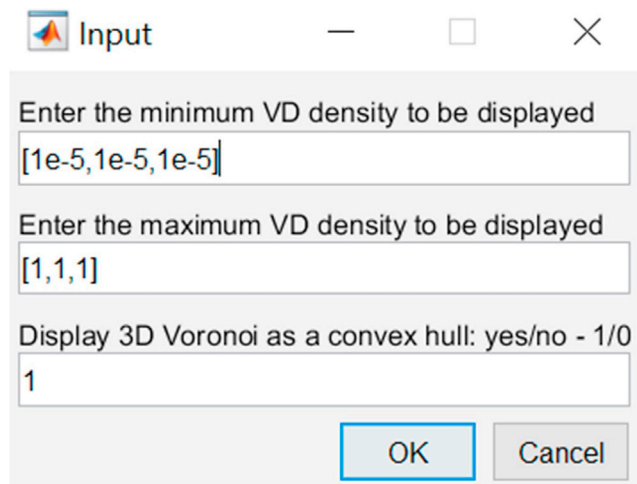


Figure 13. The input parameters for the Voronoi polygon (VP) visualization

From top to bottom: the minimum and the maximum Voronoi diagram density. The last option permits displaying the individual 3D Voronoi polygons as a convex hull.

Note: This function lets displaying only selected input variables, see [Tables 2 and 3](#). Not all the calculated variables can be displayed; for a more comprehensive variable description, please refer to the Grafeo manual that can be downloaded from the GitHub page (see [key resources table](#)).

EXPECTED OUTCOMES

The expected outcomes of this protocol are combined multicolor data files, visualization using still or animated plots, cluster and colocalization analysis including Ripley's function, Pair correlation function, and Delaunay triangulation. The analysis results can be exported to an excel file and be subjected to further analysis.

LIMITATIONS

This protocol uses Matlab based script package. Matlab programing environment is not free. Matlab-based programs can be slower than programs written in other languages e.g., C. Grafeo does not performed single molecule localization, chromatic aberration correction, and drift correction.

TROUBLESHOOTING

Problem 1

The analysis and display of the large data sets can be lengthy. Matlab can crush or suspend (freeze) its activity for a too large data set (Step 1).

Potential solution

We recommend filtering the data and using a relatively small ROI for the visualization and analysis.

Problem 2

Data importation takes too much time. Cause: Large data set (step 1).

Potential solution

Increase the minimum PN and the maximum LP to reject noise and badly localized molecules. However, no molecules with greater LP and smaller PN will be imported. The PN and LP can be adjusted

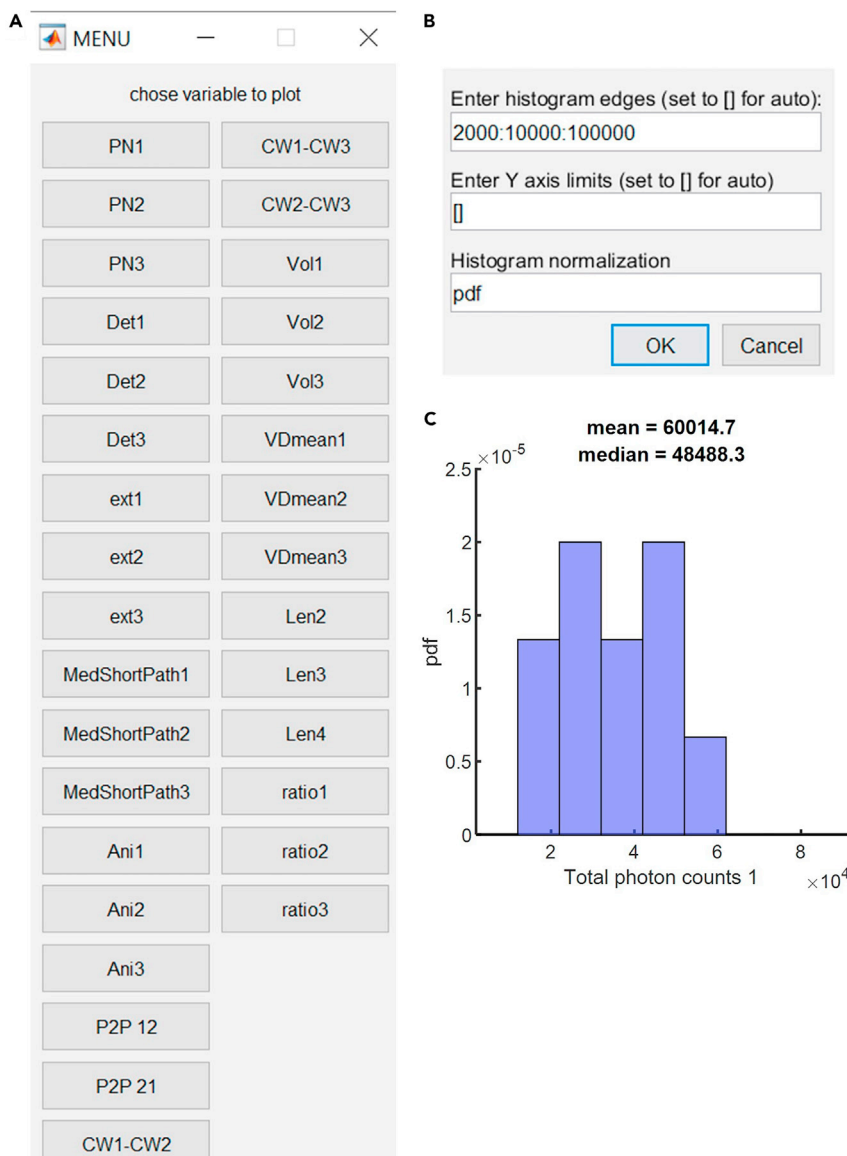


Figure 14. Displaying the results of the graph-based segmentation and the colocalization analysis

(A) A variable selection panel. PN – photon number, Det – the number of detection (points or nodes) in a graph, ext – the graph extension, MedShortPath – the median shortest path spanning the graph, Ani – the graph anisotropy, P2P – point-to-point distance (12, 21 corresponds to the first and second channel selected for the graph-based analysis; channel 1 can be 1 or 2, and the channel 2 can be 2 or 3, see also [Figure 11](#)), CW1toCW2 – the centroid to centroid distance between each segmented graph of the first color to its nearest neighboring segmented graph in the second color (12, 21 corresponds to the first and second channel selected for the graph-based analysis; channel 1 can be 1 or 2, and the channel 2 can be 2 or 3, see also [Tables 2 and 3](#)).

(B) The input parameters for the histogram plot of a variable selected in (A). From top to bottom: (1) a vector of the histogram bins (for more details, see the Matlab vector notation and the help page for the Matlab histogram function). (2) Y-axis limits, (3) the histogram normalization method, pdf – probability density function (for more details, see the help page for the Matlab histogram function).

later (see step 3), so it is recommended not to use too stringent values at this step. The good starting values are PN ~ 500 and LP ~ 50 nm.

Problem 3

Randomization takes too much time (Step 13).

Table 2. Segmented graphs data description

Variable	Internal grafeo variable name	Variable description
AniMed1	AniMed1	The median distance between the centroid of a graph and all the points/nodes belonging to the same graph.
AniMed2	AniMed2	
AniMed3	AniMed3	
Det1	Det1	The number of points (detection) in a segmented graph.
Det2	Det2	
Det3	Det3	
Ext1	Extention1	A vector of graphs' extension is defined as the maximum shortest path spanning the graph calculated for all the inter point distances (ipd). The shortest path is calculated using a Matlab function "distances" applied to a graph object.
Ext2	Extention2	
Ext3	Extention3	
MedShortPath1	MedShortPath1	A vector of the median shortest paths spanning the graph.
MedShortPath2	MedShortPath2	
MedShortPath3	MedShortPath3	
Vol1	Vol1	A vector of the total area (volume) of graphs given in $\text{nm}^2/(\text{nm}^3)$ for 2D (3D) data.
Vol2	Vol2	
Vol3	Vol3	
VDmean1	DpGsubMed1	Mean or median (option selected by the user) Voronoi density (VD) within a segmented cluster.
VDmean2	DpGsubMed2	
VDmean3	DpGsubMed3	
PN1	Pnum1	The total number of photons detected per cluster.
PN2	Pnum2	
PN3	Pnum3	
Len1	Len1	The number of points (detection) in a segmented graph taking into account multiple single molecule counting.
Len2	Len2	
Len3	Len3	
ratio1	ratio1	The ratio between the maximum to the minimum percentage of the total variance explained by each principal component.
ratio2	ratio2	
ratio3	ratio3	

Potential solution

Decrease ROI size, decrease the number of randomization steps, filter the data.

Problem 4

The randomization curve for L and PCF deviates from the straight line (Step 14).

Potential solution

The detrend polynomial degree may be incorrect. The polynomial degree is usually 2 or 3, try both values, and if necessary the higher degree polynomials.

Problem 5

The PCF fit is incorrect (Step 15).

Potential solution

Cause 1: Wrong initial values (x_0) and/or upper (UB) and lower bonds (LB) for the fit parameters. Potential solution: First, try different values of x_0 , then try to restrict UB and LB. Cause 2. The variable input parameters (σ and ρ) are wrongly estimated. Potential solution: Try first changing parameters slightly to see if the fit is improved; in parallel, change x_0 , UB, and LB. Re-estimate the parameters. Cause 3. The fitting function cannot accurately model the data.

Problem 6

No clusters were segmented (Step 16).

Table 3. Colocalization data description

Variable	Internal grafeo variable name	Variable description
CW1-CW2	CW1toCW2sort	A sorted distance matrix between weighted centroids of a graph of the first type/color to the weighted centroid of a graph of the second type/color. This matrix contains all the nearest neighbor distances from graph set 1 to graph set 2 sorted in ascending order. The "Plot graph data" function plots the histogram of distances to the first nearest neighbor.
CW1-CW3	CW1toCW3sort	
CW2-CW3	CW2toCW3sort	
Median P2P ij	dist1nn(:,1)	A median of the point to point (P2P) distances for every point (node) in a graph from a set i (j) to the n th nearest neighbor point/node in a nearest graph from a set j (i), where (i,j) can take the following values (1,2) (1,3), or (2,3). The nearest graph is evaluated on centroid to centroid distance from the set i to the set j. n, the number of nearest neighbors taken into consideration is an input parameter defined by the user, see Figure 6.
Median P2P ji	dist1nn(:,2))	
		The Median is calculated over all the calculated nearest neighbors n.
Mean P2P ij	dist1nn(:,3)	As above, but here the mean is calculated
Mean P2P ji	dist1nn(:,4))	
Std P2P ij	dist1nn(:,5)	As above, but here the standard deviation is calculated
Std P2P ji	dist1nn(:,6))	
SE P2P ij	dist1nn(:,7)	As above, but here the standard error of the mean is calculated
SE P2P ji	dist1nn(:,8))	
Minimum P2P ij	dist1nn(:,9)	As above, but here the minimum P2P is calculated
Minimum P2P ji	dist1nn(:,10))	

Potential solution

One or more input parameters are too stringent. Increase the minimum number of detections, nd, and ipd parameters.

Problem 7

Clusters are connected or 'hairy' (Step 16).

Potential solution

One or more input parameters are too low. Decrease the minimum number of detection, nd, and ipd parameters.

Problem 8

No clusters are displayed in the plot (Step 17).

Potential solution

Cause 1: No clusters were segmented. Solution: change the segmentation parameters. Cause 2: No cluster data exist in the Grafeo memory. Solution: Perform cluster segmentation or load the previously segmented data.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, [Kalina Tamara Haas] (kalina.haas@inrae.fr).

Materials availability

This protocol did not generate/use any new materials.

Data and code availability

The Grafeo script package is freely accessible through a GitHub page (<https://github.com/inatamara/Grafeo-dSTORM-analysis->). Original data containing example raw and analyzed

multicolor 3D dSTORM data have been deposited to Mendeley Data: <https://doi.org/10.17632/mj4hdpwyb8.1>.

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at <https://doi.org/10.1016/j.xpro.2021.100808>.

ACKNOWLEDGMENTS

A.P. has received the support of the French National Research Agency (ANR) GoodVibration ANR-17-CE13-0007 and of the EU in the framework of the Marie-Curie FP7 COFUND People Program, through the award of an AgreeSkills+ fellowship (under grant agreement n° 201310). The IJPB benefits from the support of Saclay Plant Sciences-SPS (ANR-17-EUR-0007).

AUTHOR CONTRIBUTIONS

A.P. prepared the samples, K.T.H. and A.P. performed the experiments, K.T.H. analyzed the data, and K.T.H. and A.P. wrote the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

- Dixon, P.M. (2014). Ripley's K function. In Wiley StatsRef: Statistics Reference Online, 3, N. Balakrishnan, T. Colton, B. Everitt, W. Piegorsch, F. Ruggeri, and J.L. Teugels, eds (Wiley). <https://doi.org/10.1002/9781118445112.stat07751>.
- Haas, K.T., Lee, M.Y., Esposito, A., and Venkitaraman, A.R. (2018). Single-molecule localization microscopy reveals molecular transactions during RAD51 filament assembly at cellular DNA damage sites. *Nucleic Acids Res.* 46, 2398–2416. <https://doi.org/10.1093/nar/gkx1303>.
- Haas, K.T., Rivière, M., Wightman, R., and Peaucelle, A. (2020a). multitarget immunohistochemistry for confocal and super-resolution imaging of plant cell wall polysaccharides. *Bio Protoc.* 10, e3783. <https://doi.org/10.21769/BioProtoc.3783>.
- Haas, K.T., Wightman, R., Meyerowitz, E.M., and Peaucelle, A. (2020b). Pectin homogalacturonan nanofilament expansion drives morphogenesis in plant epidermal cells. *Science* 367, 1003–1007. <https://doi.org/10.1126/science.aaz5103>.
- Khater, I.M., Nabi, I.R., and Hamarneh, G. (2020). A Review of Super-Resolution Single-Molecule Localization Microscopy Cluster Analysis and Quantification Methods. *Patterns* 1, 100038. <https://doi.org/10.1016/j.patter.2020.100038>.
- Levet, F., Hosy, E., Kechkar, A., Butler, C., Beghin, A., Choquet, D., and Sibarita, J.B. (2015). SR-Tesseler: A method to segment and quantify localization-based super-resolution microscopy data. *Nat. Methods* 12, 1065–1071. <https://doi.org/10.1038/nmeth.3579>.
- Nicovich, P.R., Owen, D.M., and Gaus, K. (2017). Turning single-molecule localization microscopy into a quantitative bioanalytical tool. *Nat. Protoc.* 12, 453–461. <https://doi.org/10.1038/nprot.2016.166>.
- Ovesný, M., Krížek, P., Borkovec, J., Švindrych, Z., and Hagen, G.M. (2014). ThunderSTORM: A comprehensive ImageJ plug-in for PALM and STORM data analysis and super-resolution imaging. *Bioinformatics* 30, 2389–2390. <https://doi.org/10.1093/bioinformatics/btu202>.
- Peaucelle, A., Wightman, R., and Haas, K.T. (2020). 'Multicolor 3D-dSTORM reveals native-state ultrastructure of polysaccharides' network during plant cell wall assembly'. *iScience* 23, 101862. <https://doi.org/10.1016/j.isci.2020.101862>.
- Peters, R., Benthem Muñoz, M., Griffié, J., Williamson, D.J., Ashdown, G.W., Lorenz, C.D., and Owen, D.M. (2017). Quantification of fibrous spatial point patterns from single-molecule localization microscopy (SMLM) data. *Bioinformatics* 33, 1703–1711. <https://doi.org/10.1093/bioinformatics/btx026>.
- Peters, R., Griffié, J., Burn, G.L., Williamson, D.J., and Owen, D.M. (2018). Quantitative fibre analysis of single-molecule localization microscopy data. *Sci. Rep.* 8, 1–8. <https://doi.org/10.1038/s41598-018-28691-5>.
- Sengupta, P., Jovanovic-Talman, T., and Lippincott-Schwartz, J. (2013). Quantifying spatial organization in point-localization superresolution images using pair correlation analysis. *Nat. Protoc.* 8, 345–354. <https://doi.org/10.1038/nprot.2013.005>.
- Sengupta, P., Jovanovic-Talman, T., Skoko, D., Renz, M., Veatch, S.L., and Lippincott-Schwartz, J. (2011). Probing protein heterogeneity in the plasma membrane using PALM and pair correlation analysis. *Nat. Methods* 8, 969–975. <https://doi.org/10.1038/nmeth.1704>.
- Shivanandan, A., Deschout, H., Scarselli, M., and Radenovic, A. (2014). Challenges in quantitative single molecule localization microscopy. *FEBS Lett.* 588, 3595–3602. <https://doi.org/10.1016/j.febslet.2014.06.014>.
- Shivanandan, A., Unnikrishnan, J., and Radenovic, A. (2016). On characterizing protein spatial clusters with correlation approaches. *Sci. Rep.* 6, 1–12. <https://doi.org/10.1038/srep31164>.
- Veatch, S.L., Machta, B.B., Shelby, S.A., Chiang, E.N., Holowka, D.A., and Baird, B.A. (2012). Correlation functions quantify super-resolution images and estimate apparent clustering due to over-counting. *PLoS One* 7. <https://doi.org/10.1371/journal.pone.0031457>.
- Yang, X., and Cui, W. (2010). A novel spatial clustering algorithm based on delaunay triangulation. *J. Softw. Eng. Appl.* 03, 141–149. <https://doi.org/10.4236/jsea.2010.32018>.