SpatTrack  
User Guide

# graphical abstract

# Table of Contents

[Table of Contents 2](#_Toc411703410)

[Introduction 4](#_Toc411703411)

[Installation 4](#_Toc411703412)

[Starting SpatTrack 4](#_Toc411703413)

[Image Denoising in SpatTrack 5](#_Toc411703414)

[Particle Tracking in SpatTrack 8](#_Toc411703415)

[Introduction to the graphical user interface 8](#_Toc411703416)

[Tracking particles using SpatTrack 11](#_Toc411703417)

[Trajectory Analysis 12](#_Toc411703418)

[Isolating the Cell from the Background 16](#_Toc411703419)

[Spatial Pattern Analysis in SpatTrack 17](#_Toc411703420)

[Distance to Plasma Membrane 17](#_Toc411703421)

[Nearest Neighbor Analysis 19](#_Toc411703422)

[Analysis by the Radial Distribution Function 21](#_Toc411703423)

[Two‐Color Nearest Neighbor Analysis 22](#_Toc411703424)

[Two‐Color RDF Analysis 24](#_Toc411703425)

[Particle Detection Based Co-localization Analysis (PDBCA) 25](#_Toc411703426)

[Spatial Pattern and Co-Localization Analysis by Batch Processing 27](#_Toc411703427)

[Simulation of Diffusing Particles 29](#_Toc411703428)

[References 32](#_Toc411703430)

# Introduction

SpatTrack is a multipurpose program for particle tracking and subsequent analysis of particle trajectories (i.e., the mobility of the tracked particles) or spatial pattern analysis in the cell geometry. For a detailed description of the program please see reference [1] where SpatTrack is introduced in detail. This document serves as a user guide for SpatTrack and describes the different components of the program using simulated or live cell images. In this document filenames are given by ”filename.filetype”.

# Installation

SpatTrack was developed in MatLab and is deployed as a 64-bit stand alone Windows application. Thus, to run the program you must first install the MatLab Compiler Runtime which can be found at <http://www.mathworks.com/products/compiler/mcr/>. Select the 64-bit Windows version and follow the download and installation instructions. Subsequently, run “SpatTrack.exe” from the SpatTrack folder to start the program.

# Starting SpatTrack

When SpatTrack starts, it displays an opening menu where you can choose which type of analysis to perform, see Figure 1. Together with a collection of example files, this document serves as an introduction to each of these functions.

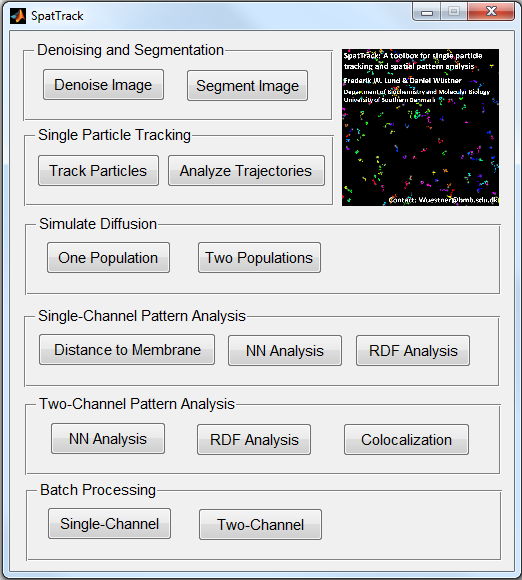


Figure : Opening screen for SpatTrack

# Image Denoising in SpatTrack

Image denoising in SpatTrack is based on a bandpass filtering method first described by Crocker and Grier [2] and also used by Lund and Wüstner [3]. First, two low-pass images are generated; 1) an image convolved by a Gaussian blur with a variance of one pixel to suppress image noise and 2) an image convolved with a boxcar filter with extent 2*w*+1, where *w* is an integer larger than the average radius of the particles but smaller than the smallest inter-sphere separation. Finally, the denoised image is generated by subtracting the boxcar filtered image from the Gaussian blurred image.

Figure 2 shows the GUI for image denoising. To try this, select ”Denoise Image” in the SpatTrack opening screen and open the ”noisy.tif” image from the test images folder. It should be noted that using an innappropriate particle size during image denoising will affect the particle sizes in the denoised image. Therefore, SpatTrack includes a GUI to optimize the denoising settings. In the ”Image Denoiser” GUI Click ”Optimize” and then click ”Load” in the GUI that opens, see Figure 3.

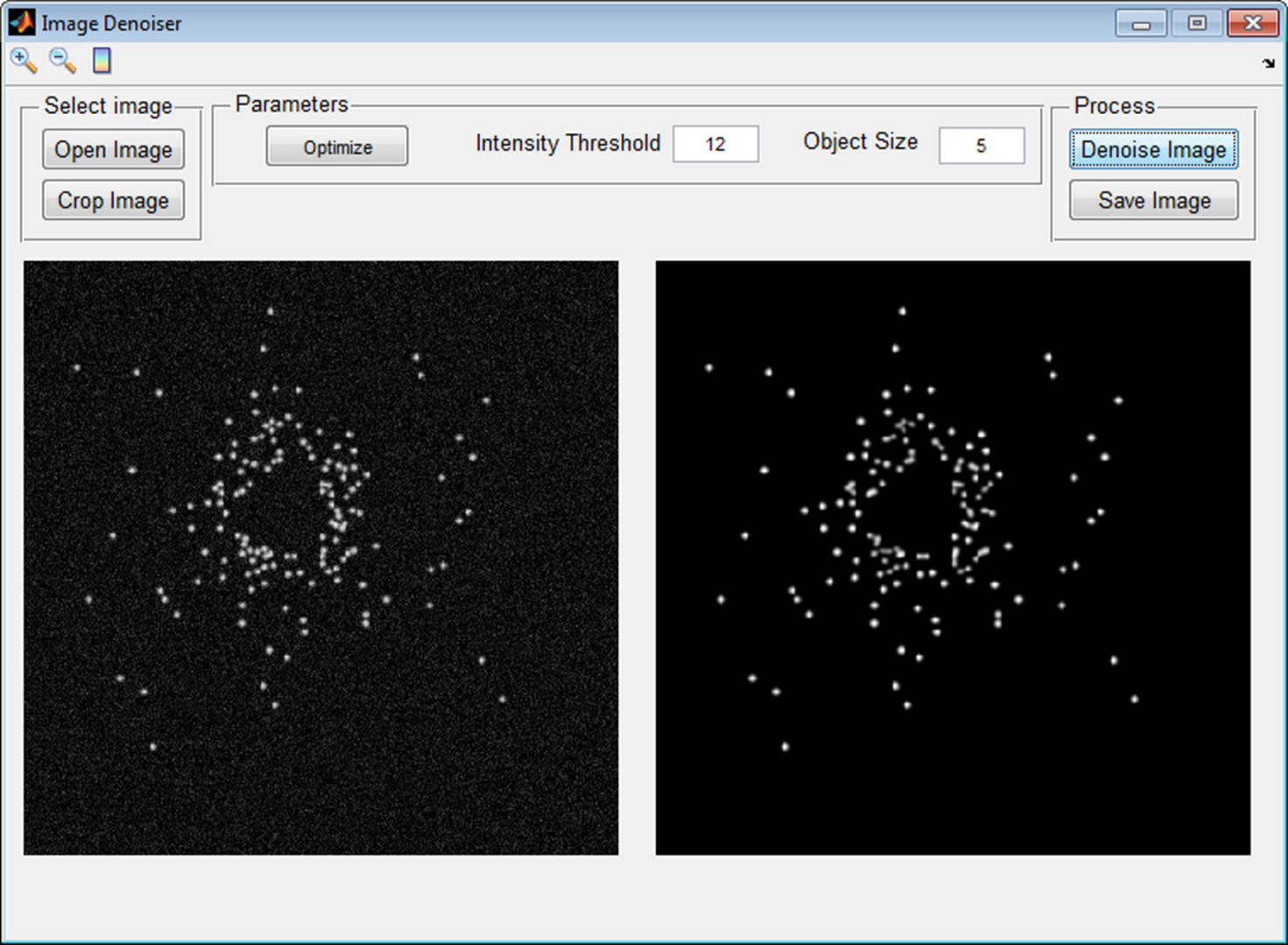


Figure : GUI for image denoising. The raw (noisy) image is shown in the left side panel while the right side panel shows the denoised image.

In the optimization GUI, the user must select an integer diameter of the particles and an intensity threshold. Then, the image is denoised with different particle diameters, and the particle sizes in the denoised image are compared to the size of the particles in the raw (noisy) image. This comparison of particle sizes is performed by fitting a 2D Gaussian function to a user defined number of particles in the raw image and in each of the denoised images. Then, the pixel-wise deviation of the variances of the 2D Gaussian functions fitted to particles in the denoised and the raw images are displayed in the right hand panel. For example, in Figure 3 the initial particle size is set to 6 pixels and the intensity threshold is set to 12.5. From the graph in the right hand panel, one can then see that the variance of the 2D Gaussian fitted to the particles in the image denoised with a particle diameter of 5 pixels is equal to the variance of the 2D Gaussian fitted to the particles in the raw image. Thus, denoising the image with a particle diameter of 5 pixels and an intensity threshold of 12.5 results in a denoised image with particles of the same size as in the raw image.

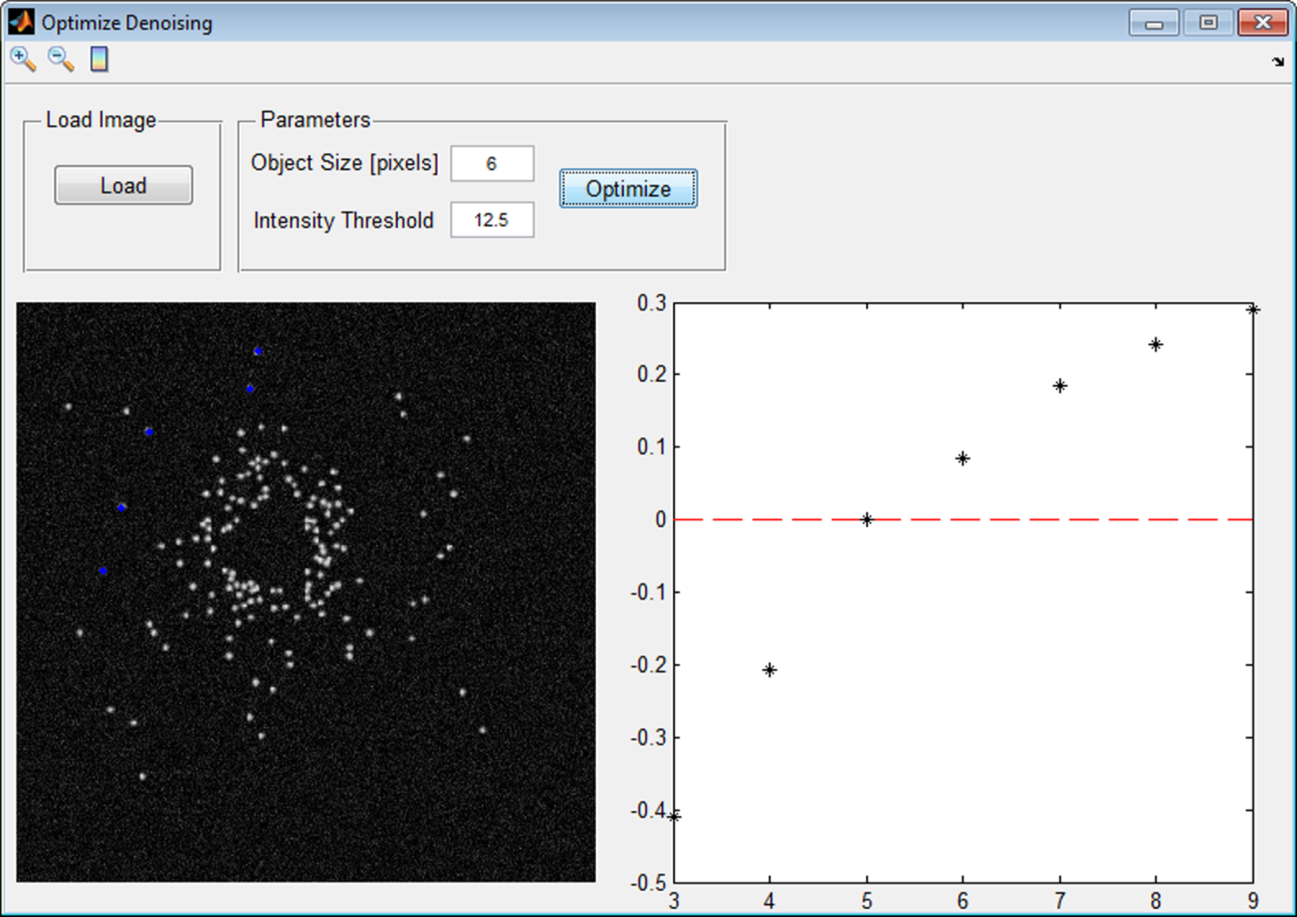


Figure : GUI for optimization of the denoising settings.

To try the denoising optimization GUI, click ”Optimize” in the image denoising GUI, and once the optimization GUI opens, click ”Load”. Set the ”Object size” and ”Intensity threshold” to appropriate values and click ”Optimize”. This opens a dialog which asks for a number of particles to include in the optimization. Usually 5 – 10 particles should do it. Click ”OK” and in the new window, select a number of easily separable particles by placing the cursor over the particle and clicking the left mouse button. For an example, see the blue dots in the left side panel of Figure 3. After the desired number of particles has been selected, the image window closes, and SpatTrack denoises the image with different particle sizes and plots the difference between the fitted variances of the 2D Gaussian in each denoised image compared to the raw image. Notice, that you can fit the image screen to fit the computer screen by double clicking the upper bar of the window. This may make it easier to select the particles. Furthermore, if the optimization process is performed additional times, the program asks, whether the same particle coordinates should be reused. Thus, it is not necessary to select the particles multiple times.

Once the appropriate denoising settings have been determined, return to the image denoising GUI to perform the actual denoising of the image. Insert an intensity threshold of 12.5 and an object size of 5 pixels and click “Denoise”. Subsequently, save the image to file for further use.

# Particle Tracking in SpatTrack

## Introduction to the graphical user interface

Single particle traking (SPT) in SpatTrack v.1.1 has been considerably upgraded compared to SPT in SpatTrack v.1.0. Most importantly the tracking algorithm is now approximately 10 times faster than the original algorithm while it allows for particle merging and/or particle splitting. Furthermore, visualization of the trajectories has been significantly improved and it is now possible to save the tracking settings for a later session. Figure 4 shows the new GUI for SPT. SPT in SpatTrack is performed in three steps. First the particles are located in each frame and the coordinates are refined to sub-pixel precision, second the coordinates are linked from frame-to-frame and finally the algorithm attempts to close the gaps in incomplete trajectories.

Detection of the particles require two parameters, which are set in the “Particle Detection” panel, 1) a fluorescence intensity threshold and 2) a particle location diameter. To enable tracking of bleaching particles the intensity threshold is adjusted for each frame in the image stack. Simply put, for each frame, the average intensity of the respective frame is calculated and multiplied by the user supplied threshold. For example, in Figure 4 the fluorescence intensity threshold is four times the average fluorescence intensity of each respective frame. The particle “Location Diameter” ensures that the algorithm does not assign more than one particle coordinate per particle. For example in a noisy image there may be several high intensity pixels in a single particle. In this case the “Location Diameter” should be set to a value slightly larger than the particles but smaller than the distance between two neighboring particles. On the other hand, in image sequences where two or more particles merge or split there may be more than one correct particle coordinate in what appears to be one particle. Hence, for merging/splitting particles the “Location Diameter” may be set to a value smaller than the particle radius. For example in Figure 4, the image was simulated with a particle radius of 4 pixels but due to merging/splitting particles the “Location Diameter” is set to 3 pixels.

Frame-to-frame linking is performed by the [simpletracker](http://www.mathworks.com/matlabcentral/fileexchange/34040-simple-tracker) MatLab function developed by Jean-Yves Tinevez and available on the MatLab File Exchange. Initially, the Euclidean distance is calculated for all particle coordinates in frame *i* with respect to all particle coordinates in frame *i*+1. Subsequently, the sum of particle-particle distances is minimized by the Hungarian algorithm such that each particle in frame *i* is connected to the nearest neighbor in frame *i*+1. To avoid unrealistically large link-distances the user must supply the “Maximum Displacement” allowed in the “Trajectory Linking” panel of the tracking GUI, see Figure 4.

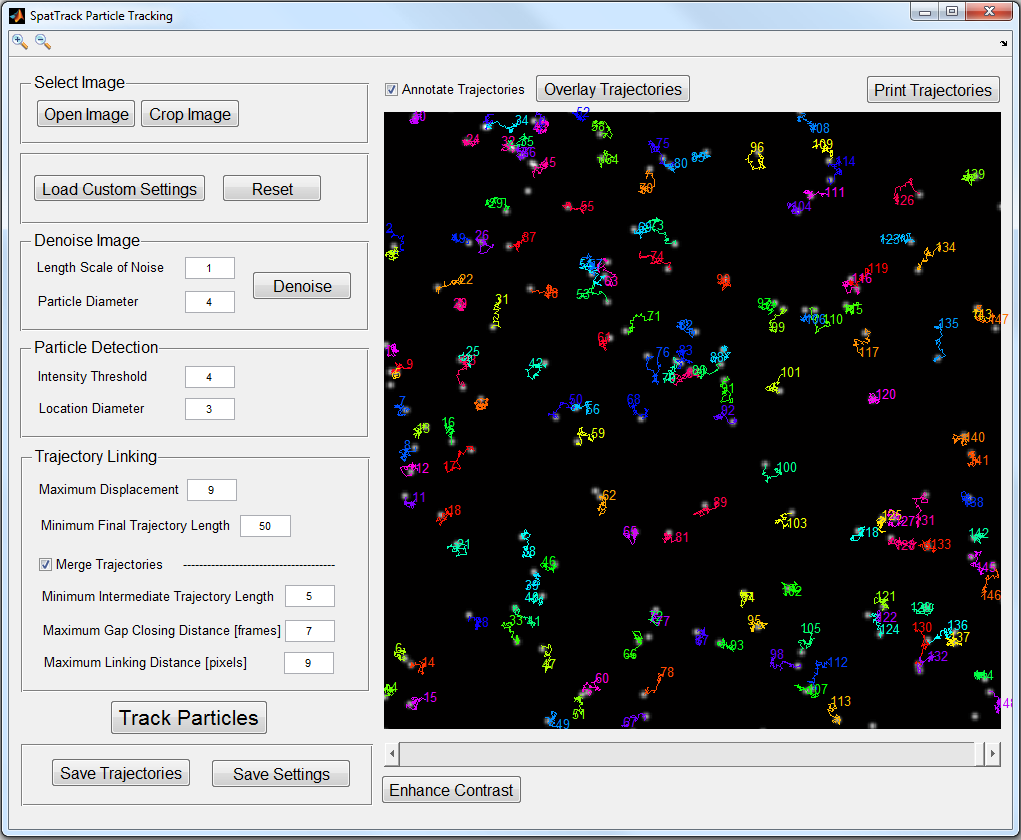


Figure : ) Graphical user interface of the SpatTrack particle tracking suite. Here the particles in simulatedImage1.tif were tracked and overlaid on top of the image.

At this stage there may be several incomplete trajectories (i.e., trajectories shorter than the number of image frames). These may arise from particles entering or leaving the imaged area during image acquisition, due to particles temporarily dissapearing from the focal plane or due to merging or splitting particles. SpatTrack now includes the possibility to re-link incomplete trajectories. To apply re-linking the “Merge Trajectories” box must be checked in the GUI. To account for image noise the user must set the “Minimum Intermediate Trajectory Length” which is the shortest length of a trajectory considered for linking. For example, a particle which is only detected in a single frame may very likely be due to noise while it is less likely that a trajectory of e.g. 5 consecutive coordinates arise from image noise. After these short trajectories have been discarded the remaining incomplete trajectories may be classified as one of three types, see Figure 5; 1) the trajectory starts at t = 0 but ends before the final frame, 2) the trajectory starts at t > 0 and ends before the final frame or 3) the trajectory starts at t > 0 and ends at the final frame. The incomplete trajectories may be re-linked in three steps as shown in Figure 5.

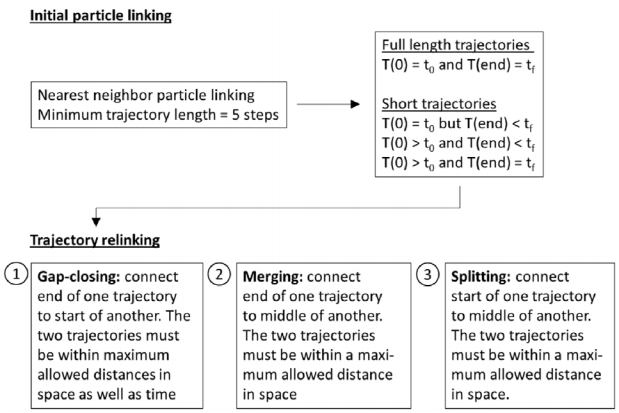


Figure : Schematic description of the particle tracking algorithm. Initially, the found particle coordinates are linked using the nearest neighbor linking method with a minimum trajectory length of 5 steps. Subsequently, any trajectories shorter than the number of frames in the video sequence are re‐linked in a three‐step process. First short gaps are closed by linking the end of one trajectory to the start of another. Then merging events are detected by letting the end of one trajectory link to the middle of the nearest other trajectory. Finally, particle splitting events are handled by letting the middle of one trajectory link to the start of another nearby trajectory in the following frame.

Gap closing is based on the temporal gap as well as the euclidean distance between the end of one trajectory and the start of another. The second trajectory must start in a later frame than the first one ends but no later than a maximum number of frames. Similarly the Euclidean distance between the end and the start of the two trajectories must not exceed a maximum distance. In practice these parameters are given by the “Maximum Gap Closing Distance [frames]” and the “Maximum Linking Distance [pixels]” in the GUI, see Figure 4. From all possible trajectories fulfilling these criteria, the link is made between the two trajectories which minimize the Euclidean end-to-start distance but such that no end is connected to more than one start and vice versa. After this step the trajectory is still incomplete. In order for the trajectory to be complete the missing coordinates must be filled with coordinates from another trajectory within the allowed spatial linking distance in the same frames. Otherwise the trajectory is eventually discarded. Particle merging is performed by linking the end of one trajectory (at time t‐1) to the middle of another trajectory (at time t) while splitting is performed by linking the start of one trajectory (at time t) to the middle of another trajectory (at time t‐1). Unlike gap‐closing, trajectory merging and splitting is only allowed between coordinates found in consecutive frames. Additionally, merging or splitting is only allowed if the Euclidean distance between the trajectories is smaller than the “Maximum Linking Distance [pixels]”.

## Tracking particles using SpatTrack

To try particle tracking in SpatTrack, open the image sequence “simulatedImage1.tif” from the “Test Images” folder. For convenience the tracking addon includes a denoising component. However, this image sequence was simulated without noise and you can go directly to particle tracking. Under “Particle Detection” set the “Intensity Threshold” to 4 and the “Location Diameter” to 3. As described above linking the trajectories requires a number of parameters. Additionally, you can determine whether you want SpatTrack to re-link the trajectories to close holes. Generally, we recommend relinking but it may produce a large number of false positive links if the particle density is high. For now we will leave the “Trajectory Linking” parameters as they are. Click “Track Particles” and wait for a few seconds. To visualize the trajectories click “Overlay Trajectories” above the image. Now it is possible to follow the progression of each trajectory over time by clicking the scroll bar below the image. Additionally it is possible to annotate the trajectories. To do this check the “Annotate Trajectories” check box and click on “Overlay Trajectories” to update the plot. Using the looking glass icons at the upper left corner of the GUI it is possible to zoom in on a region of interest. Now if you click the scroll bar you will be able to follow the progression of the trajectories and possible particle merging in a zoomed in view. To save a high resolution of the image click “Print Trajectories” which allows you to save a publication quality figure at 300 dpi. Finally it is possible to save the trajectories to three separate .txt files. One for the x-coordinates over time, one for the y-coordinates over time and one for the particles’ fluorescence intensity over time. These files are saved as comma separated text files and may be opened by any of the other GUIs in SpatTrack or alternatively, they may be opened in a spread sheet program (e.g. microsoft excel). Importantly, the files are formatted such that each column contains the coordinate (or intensity) of one particle as a function of time (i.e. information about particle number one is found in the first column, information about particle number two in the second column and so on).

# Trajectory Analysis

The first type of trajectory analysis in SpatTrack is based on the mean square displacement (MSD) of the trajectories. To perform trajectory analysis press “Analyze Trajectories” in the starting screen. To demonstrate trajectory analysis open the tracking GUI and track the particles in “simulatedImage1.tif” as described above. Then open the trajectory analysis GUI and press “Load” in the upper right corner of the GUI. This imports the particle coordinates directly from the tracking GUI. Set the “Pixel size” to 0.1 and the frame rate to 1; the pixel size is in microns while the frame rate is in seconds. Finally, set the “Range” under MSD analysis to 0.8, press “Plot MSD” and the “Fit MSD”. Now, you should get a diffusion constant of ~0.01 µm2/s in the results panel on the right, see Figure 6. As an alternative to loading the particle coordinates directly from the tracking GUI, you may load the coordinates from properly formatted .txt files as those discussed above. This also allows you to perform particle tracking with another program and then import the particle coordinates into SpatTrack for subsequent analysis. To do this, the files must be formatted as described above (i.e. one file for the x- and y-coordinates of the particles and one more file for the fluorescence intensities. The files should be formatted such that the data for a given trajectory progresses through the same column). Notice, however, that it is not required to import a file containing particle fluorescence intensities. One way to save the files in the required format is to save them as a .csv file from Microsoft Excel, where each column in the spread sheet contains the coordinates at each frame number. This generates a file where the coordinates are separated by a semicolon. Thus, the semicolons must be replaced by a comma. In Windows this can be done in notepad.

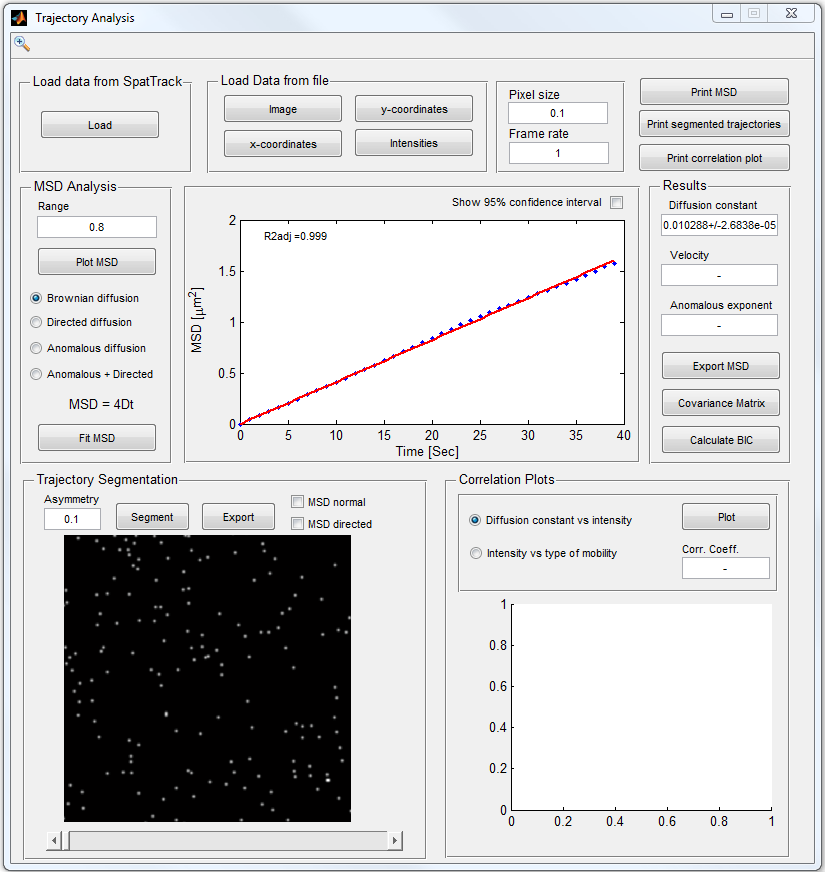


Figure : GUI for trajectory analysis in SpatTrack. Here the trajectories obtained from SimulatedImage1.tif were analyzed which revealed that the particles were moving by normal diffusion with a diffusion constant of ~0.01 µm2/s.

In addition to the Brownian or normal diffusion analyzed above, SpatTrack includes three other motion models 1) directed diffusion, 2) anomalous diffusion and 3) anomalous diffusion plus flow. The latter was developed to describe the motion of endosomes which showed anomalous subdiffusion on short time scales and directed diffusion on longer time scales [3]. However, it may also be useful where some particles are moving by anomalous subdiffusion while other move by directed diffusion.

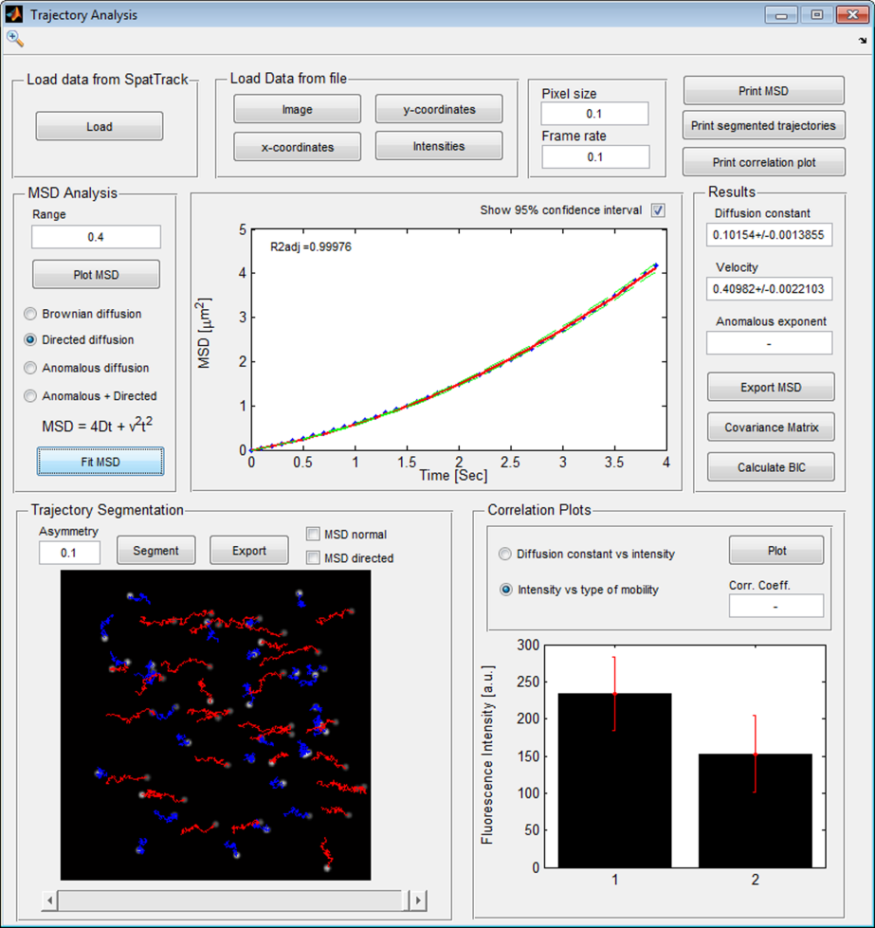


Figure : Trajectory analysis GUI of SpatTrack

SpatTrack allows the user to show the 95% confidence interval of the fitted curve, see the green dashed lines in Figure 7. Furthermore, the standard error is given for each fitted parameter value. To assist the user in determining the best model of fit a particular MSD, we suggest using the decision tree described in reference [1].

SpatTrack may also separate particles moving by normal/anomalous subdiffusion from particles moving by directed diffusion. Open “twoTypes.tif” in the particle tracking GUI and track the particles. You can use the same settings as for the “SimulatedImage1.tif” image. Now save the trajectories or import them directly into the “Trajectory Analysis” GUI by clicking the “Load” button. This image sequence contains two particle populations. One is moving by Brownian diffusion with a diffusion constant of 0.01 μm2/s while the other is moving by directed diffusion with a diffusion constant of 0.01 μm2/s and a drift velocity in the positive x-direction of 0.1 μm/s. Again, the image sequence was simulated with a pixel size of 0.1 μm/pixel and a frame rate of 1 frame per second. Thus, the MSD analysis yields a diffusion constant of 0.012 μm2/s and a velocity of 0.071 μm/s. In this image we can separate particles moving by directed diffusion from particles moving by normal diffusion. To do this, set the “Asymmetry” in the lower part of the GUI to 0.1 and press “Segment”. Now, trajectories of the particles moving by normal diffusion are shown in blue, while trajectories of the particles moving by directed diffusion are shown in red, see Figure 8B. The asymmetry of each trajectory is calculated from its radius of gyration as discussed in [1]. Clicking the “MSD normal” or “MSD directed” above the image lets you fit the MSD of the two particle populations. Click “MSD directed” then “Plot MSD” and finally “fit MSD”. This shows that the particles moving by directed diffusion have an average diffusion constant of 0.0153 µm2/s while the velocity is 0.0969 µm/s. After the trajectories have been segmented you may save the coordinates of the particles to comma separated text files by pressing “Export” in the trajectory segmentation panel.

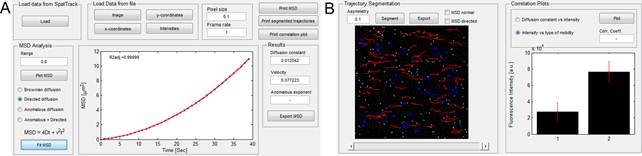


Figure : Tracking of particles moving by normal diffusion and diffusion plus flow. A) shows the MSD analysis while B) shows the segmented trajectories and a plot of the fluorescence intensity versus type of mobility.

The particles in the image sequence were simulated so the fluorescence intensity of the particles moving by directed diffusion was three times larger than the fluorescence intensity of the particles moving by Brownian diffusion. This is shown in the right panel of Figure 8B, in which 1 denotes Brownian diffusion while 2 denotes directed diffusion. Finally, if the image sequence contains different particles with different diffusion constants and fluorescence intensities, we may want to find out whether there is a correlation between both parameters. Open “fourTypes.tif” and track the particles. Then load the data into the trajectory analysis GUI, check the “Diffusion constant vs intensity” and press “Plot”. Now, we can see from the plot that there is a positive correlation between the diffusion constants and the fluorescence intensities, see Figure 9. In the plot, the fluorescence intensities are normalized so the largest intensity equals the largest diffusion constant.

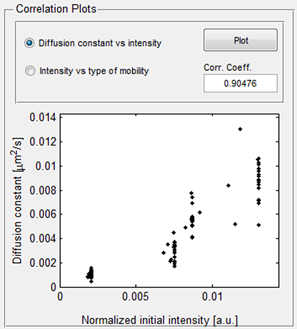


Figure : A plot of diffusion constant versus fluorescence intensity.

Several data may be saved from the trajectory analysis GUI. Clicking “Print MSD”, “Print segmented trajectories” or “print correlation plot” lets you save the plots as high resolution images. The “Export MSD” opens a spread sheet from which the calculated MSD may be copied to other programs e.g. Microsoft Excel.

# Isolating the Cell from the Background

In order to analyze the spatial pattern of endosomes with respect to the cell geometry SpatTrack requires a mask of the cell on a dark background. This may be obtained by the cell segmentation utility, see Figure 10. Initially, the image is divided into a number of tiles in each direction and the contrast of the image is enhanced locally in each of these tiles using the contrast-limited adaptive histogram equalization (CLAHE) algorithm [4, 5]. For example, in Figure 10 the image is divided into 6 tiles in the x- and y-direction giving a total of 36 tiles. Second, a median filter is applied to the image for an easier separation of the cell from the background. Finally, the cell may be isolated by adjusting the intensity threshold.

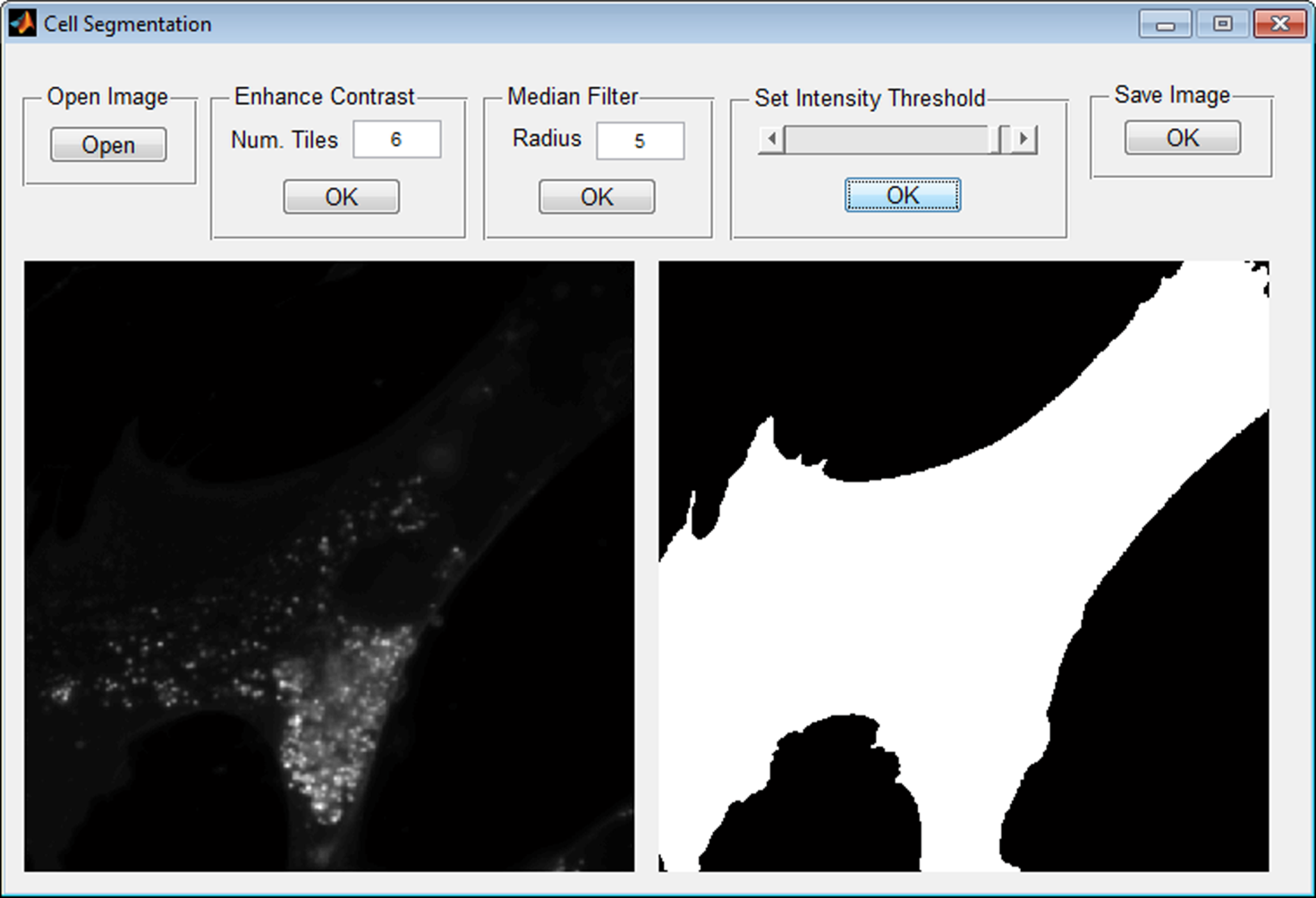


Figure : Cell segmentation GUI. Initially, the image is divided into a number of tiles in each direction. Then the image contrast is enhanced in each tile by the CLAHE algorithm. Second, the image is applied a median filter which eventually allows the segmentation of the cell from the background.

From the image folder in the user guide, open “FilipinImage.tif”. Then try changing the number of tiles and click “OK”. The result is shown in the right hand panel of Figure 10, while the left side panel always shows the raw image. Usually, using fewer tiles makes it easier to isolate the cell. Subsequently, try adjusting the size of the median filter to see the effect on the processed image. Finally, adjust the intensity threshold to isolate the cell from the background. At this stage several smaller spots outside the cell are included in the segmentation. However, once you click “OK” only the largest blob is preserved.

# Spatial Pattern Analysis in SpatTrack

## Distance to Plasma Membrane

Given the location of endosomes in a living cell and the shape of the cell we can determine the distance from the particles to the plasma membrane (PM). Figure 11 shows the GUI for this analysis. Here the minimum, maximum and average distances may be calculated. Furthermore, one can generate a random particle distribution and compare the experimental and random distributions. The details of this analysis are described in [1]. To perform this analysis, open “NPC2\_Image.tif” in the tracking GUI, set the particle size to 4 pixels and click “Show Bpass Image”. Once the image has been denoised set the threshold to 8 and the particle location diameter and maximum displacement to 5 and press “Track”. Next, click “Distance to PM” to open the distance to PM GUI. In the distance to PM GUI open image “Mask.tif” from the folder containing the example images. Click ”Load Coordinates” to import the x- and y-coordinates from the tracking GUI and click “Plot Data”. Now you should see the outline of the cell in yellow and the detected particles in blue. Notice that any particles outside of the cell geometry are automatically excluded from the calculation. Set the particle size to 0.222 (in μm/pixel) and click “Calculate Distances”. Now, you should see the results for the tracked particles. Next, set the particle size to 4 (this is the particle size used during image denoising) and the iterations to 20. Press “Calculate” to generate 20 randomly distributed particle distributions to compare to the detected particles. As described in [1], performing 20 calculations gives us a 95% confidence interval for the calculations. Now, you can see a curve of the number of particles as a function of the distance to the PM in blue and red for the detected and random particles, respectively. Finally, in the upper right corner you may show the particles within a given distance to the PM. Here particles within 1 μm of the PM are shown in red. This is 4% of the detected particles and 16.6% of the random particles. Finally, pressing “Export Results” opens a spread sheet from which the two curves may be copied.

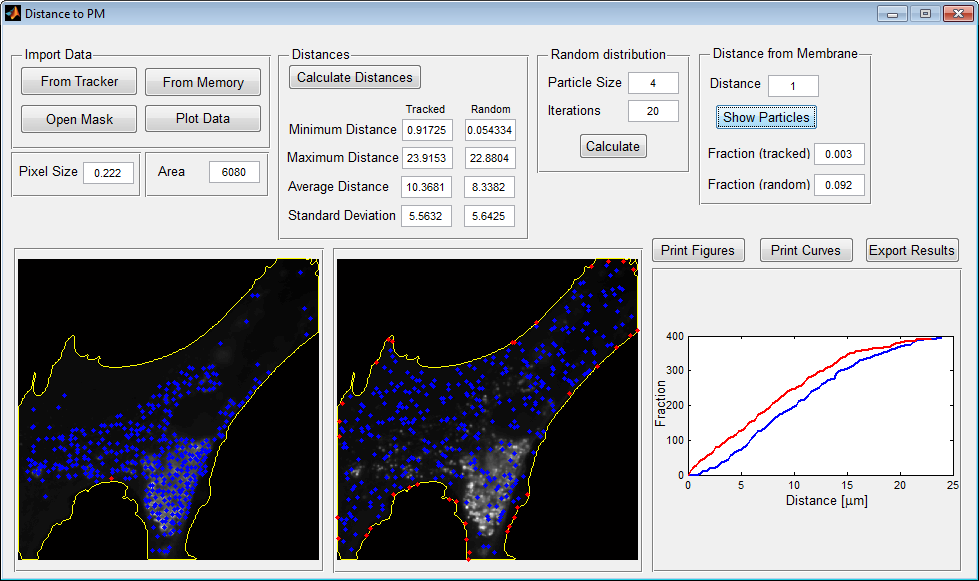


Figure : GUI for analysis of the particles’ distance to the plasma membrane (PM).

The process above described how to analyze data directly from the tracking GUI. Alternatively, the particle coordinates may be loaded from text files containing the x- and y-coordinates, respectively. To do this, click “From Memory”. This opens the SpatTrack file manager which is used as a local memory to ease multiple analyses of the same data. In the remainder of this document we will use coordinates from the file manager. In the ‘Distance to PM GUI’ click “From Memory”. Then open “NPC2\_xCoords.txt” and “NPC2\_yCoords.txt” as the x- and y-coordinates for population 1. The text files can be found in the “Tracked Coordinates” folder in the “Test Images” folder. Additionally, open “Mask.tif” from the “Test Images” folder. We will not load any coordinates for population 2, for now.

## Nearest Neighbor Analysis

Another type of spatial pattern analysis is the nearest neighbor (NN) analysis. Here the distance from each particle to is nearest neighbor is calculated. Subsequently, the average NN of the detected particles divided by the average NN of a random particle distribution in the same cell geometry may reveal whether the particles are randomly distributed, accumulating or repulsive to each other.

Figure 12 shows the GUI for NN analysis. In the GUI, click ”From Memory” to import the particle coordinates and the mask image from the file manager and click ”Plot Data”. Subsequently, set the pixel size to 0.222 and the particle size to 4. Notice that any particles located outside of the boundary are not included in the calculation. To calculate the minimum, maximum and mean NN distances for the detected and the random particles press “Calculate”. This also yields an ANN score of 0.84. This is the average NN distance of the detected particles divided by the average NN distance of the randomly generated particles. Thus, for ANN = 1 the particles are randomly distributed while ANN < 1 or ANN > 1 indicates accumulation or repulsion, respectively.

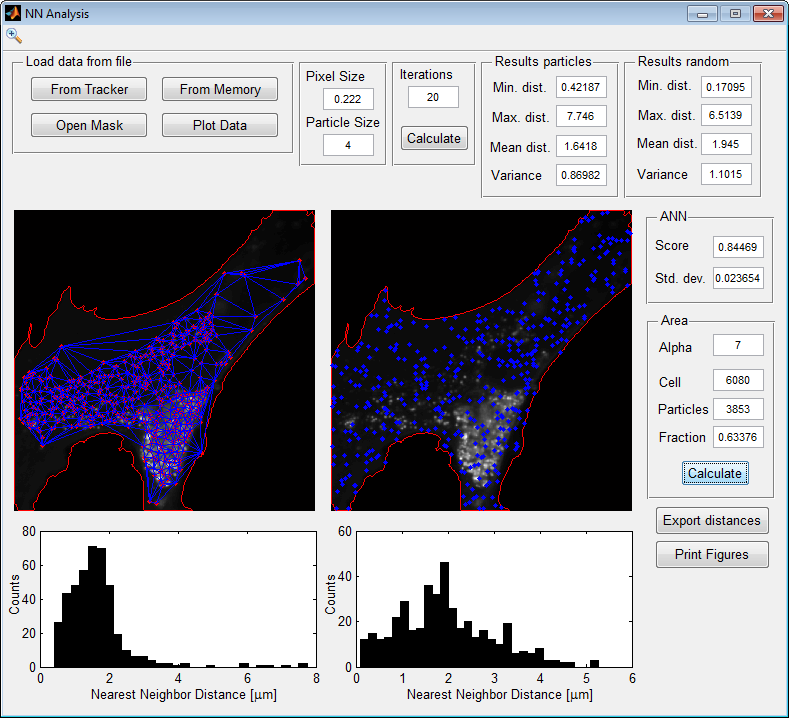


Figure : GUI for NN analysis. The detected particles are shown in the left panel while the random particles are shown in the right panel.

In addition to NN analysis this GUI lets you calculate the fractional area occupied by the particles in the cell. As described in [1] calculating the area occupied by the particles is based on an alpha shaping method. First the particles are connected by a Delaunay triangulation, as shown by the blue lines connecting the particles. However, the triangulation tends to overestimate the area. Therefore, we place a circle with a radius given by  where α is a user defined parameter, and *NNobs* is the average NN distance of the detected particles. This circle is placed at the center of each line segment in the triangulation, and circles which do not contain any of the tracked coordinates are deleted. Thus, for a sufficiently large α no line segments are deleted while for a sufficiently small α all line segments are deleted. For, further explanations of this method we refer to reference [1] and the supplementary material. In the open image, set α ~7 and press “Calculate”. Now we can see that the area of the cell is 6080 µm2 while the area of occupied by the detected particles is 3853 μm2 which yields a fractional area occupied by 0.63.

## Analysis by the Radial Distribution Function

While the NN analysis may reveal, if the particles are randomly distributed, accumulating or repulsive, this analysis is limited to the nearest neighbor length scale. On the other hand, analysis using the radial distribution function (RDF), as shown in Figure 13, may be used to determine spatial patterns on a large range of distances. For each particle, the number of neighboring particles is calculated as a function of distance. Next, the same calculation is performed for a random particle distribution. Then, the radial distribution function g(r) is given by the average number of particles for the detected population divided by the average number of particles at the same distance found in the random distribution. For more information see [1].

The GUI for RDF analysis, see Figure 13, may be opened from the start screen or from the tracking program. In the GUI click “From Memory” and then ”Plot Data” to load and display the cell geometry and particle coordinates from the file manager. Set the particle size to 0.222 and the particle size to 4. Keep the number of iterations to 20 and the “Max. Radius” to 10. When you press “Calculate”, two plots are generated. On the left, the average number of particles as a function of radius is shown for the detected and random particles in black and blue, respectively. The left plot shows g(r) as a function of the radius, where g(r) = 1 indicates a random distribution while g(r) < 1 or g(r) > 1 indicate repulsion or accumulation, respectively.

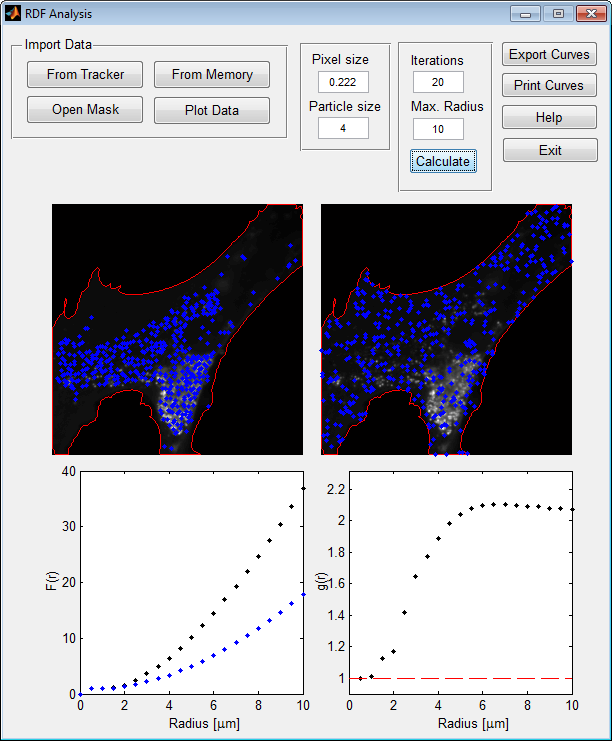


Figure : GUI for analysis by the radial distribution function (RDF). The detected particles are shown in the left side panel while the random particles are shown in the right side panel.

## Two‐Color Nearest Neighbor Analysis

The nearest neighbor (NN) analysis provides information about whether the particles in a population are randomly distributed, accumulating or repulsive, see above. The average NN distance for all the particles is determined and compared to the average NN distance for an equally large, randomly distributed population of particles in the same cell geometry. Two-color NN analysis reveals how the particles in one population relate to the particles in another population (e.g. how filipin containing endosomes relate to NPC2 containing endosomes). In this analysis, the NN distance for a particle in population 1 is the distance to the nearest particle of type 2. Subsequently, this is compared to the average NN distance for the particles of type 1 to a population of randomly distributed particles of the same size as population 2.

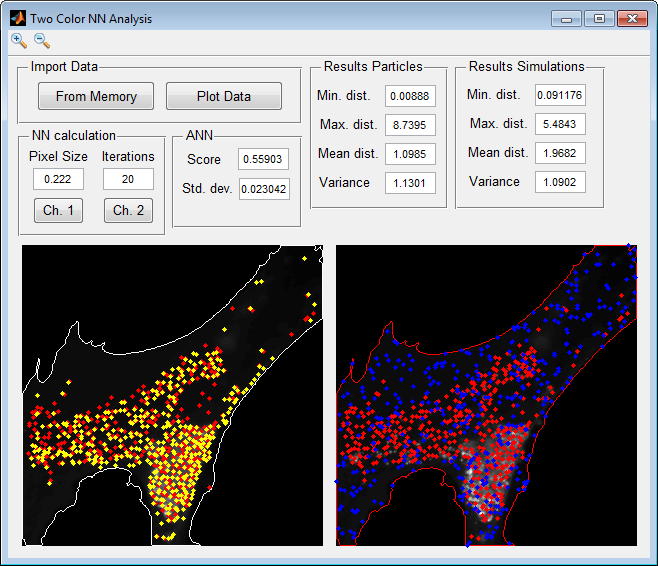


Figure : GUI for the two-color NN analysis. In Panel A, the particles from population 1 and 2 are shown in red and yellow respectively. Panel B shows the particles from channel 2 (yellow) compared to a random distribution of particles (blue). Data import and calculations are performed at the top of the GUI.

To perform the two-color NN analysis go to the file manager and open “Filipin\_xCoords.txt” and “Filipin\_yCoords.txt” as x- and y-coordinates for population 2. Then open the two channel NN analysis GUI from the SpatTrack main GUI and click “From Memory” followed by “Plot Data”. Once the image mask and the particle coordinates are loaded set the pixel size to 0.222 µm/pixel and the number of iterations to 20. Figure 14 shows the GUI for two-color NN analysis, where the fibroblast cell co-labeled with filipin and NPC2 described above is analyzed. In Panel A, the locations of NPC2 containing endosomes are shown in red while the locations of filipin containing endosomes are shown in yellow. The distribution of particles in population 1 (NPC2) with respect to particles in population 2 (filipin) is calculated by clicking the “Ch. 1” button. The opposite calculation is performed by clicking the “Ch. 2” button. In panel B of Figure 14 the distribution of NPC2 containing endosomes are shown compared to the distribution of a random particle population containing the same number of endosomes as the population of filipin containing endosomes. From this it can be seen that the average NN distance from NPC2 containing endosomes to filipin containing endosomes is 1.0985 µm while the average NN distance to the randomly distributed particles is 1.9682 µm. This yields an average NN distance ratio (ANN) of 1.0985 µm / 1.9682 = 0.56 which indicates that the bNPC2 and filipin containing endosomes are accumulating.

## Two‐Color RDF Analysis

As described in [1], the NN analysis is restricted to relatively short distances, while analysis by the radial distribution function provides information about the distribution of the particles over a range of distances. For one type of particles, the number of particles as a function of distance to a reference particle is calculated and averaged over all particles. Then the same measure is calculated for a randomly distributed particle population and the distribution state is given by the ratio of the RDF for the detected particles to the RDF for the random particles. Figure 15 shows the GUI for two-color RDF analysis. Load the mask image and particle coordinates by clicking “From Memory” and “Plot Data”. Again, population 1 and 2 contain the coordinates of NPC2 containing endosomes and filipin containing endosomes, respectively. Set the pixel size to 0.222 µm/pixel and the number of tests to 20. Finally, set the Max. Radius to a suitable distance, in Figure 15 it is set to 10 µm. Figure 15B shows at plot of the average RDF as a function of distance to a particle. As described above, the RDF is defined by the following properties; g(r) = 1 indicates a random distribution while g(r) < 1 or g(r) > 1 indicate repulsion or accumulation, respectively, where g(r) is the RDF. From Figure 15B it can be seen that at radii below 1.5 – 2 µm, the g(r) is much larger than 1 which shows that the two particle populations are accumulating on this length scale. However, at longer distances the RDF approaches a value of 2 suggesting a weaker accumulation over longer distances.

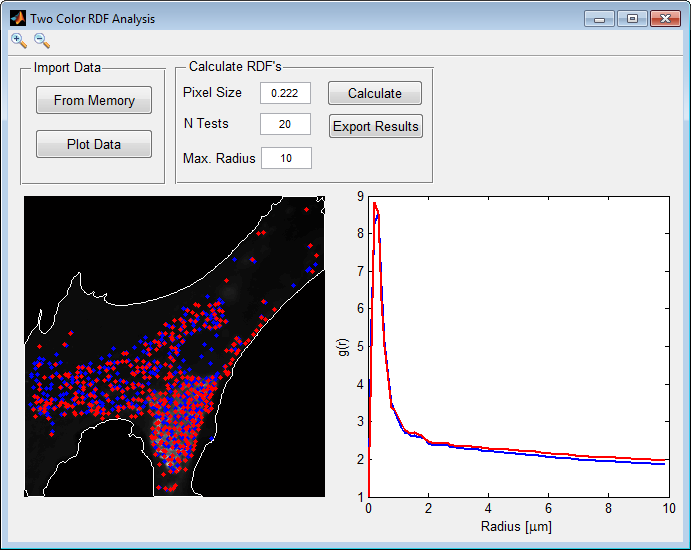


Figure : GUI for two-color RDF analysis. Data import and calculations are performed at the top of the GUI. In the left panel the particles are plotted on top of the cell where particles of type 1 and 2 are shown as blue and red dots, respectively. The plot in the right hand panel shows the RDF as a function of radius in the same colors.

# Particle Detection Based Co-localization Analysis (PDBCA)

Figure 16 shows the GUI for PDBCA. Here the x- and y-coordinates for the particles in each channel as well as the denoised images used for particle tracking, are loaded in the top of the GUI. Then the number and fraction of co-localized particles are determined on basis of the pixel size of the images and the maximum allowed displacement between the same particles in the two channels. The GUI operates in two modes. In the first mode, the co-localization between all particles in the images are calculated. This could for example be applied to images containing more than one cell. Alternatively, the user may choose to load a mask image, in which the cell is shown in an image with a background of zero intensity. From this mask, the cell boundary is determined and any particles located outside the boundary are excluded from the calculation. The top and bottom images to the left in Figure 16 show the denoised images from channel 1 and 2, respectively. In this example, the calculations were performed with a mask, and the larger image to the right shows the cell with the boundary (white line). The particles found in channel 1 and 2 are shown as blue and red points respectively. Finally, co-localized particles are shown as yellow dots.

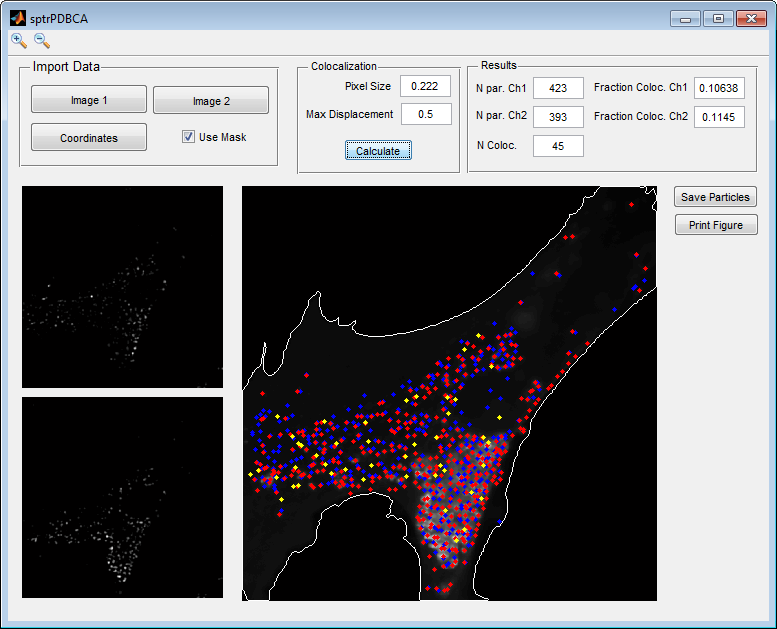


Figure : GUI for particle detection based co-localization analysis (PDBCA). Initially, the data for the two particle populations are loaded. Then the number of co-localized particles is determined based on the pixel size of the images and the maximum allowed displacement between co-localized particles in the two channels.

To perform co-localization analysis, open the mask image and the x-and y-coordinates for the NPC2 and filipin containing endosomes from the file manager by clicking “Coordinates”. subsequently, open “NPC2\_denoised.tif” and “Filipin\_denoised.tif” as images for channel 1 and 2, respectively. Check the “Use Mask” check box and set the pixel size to 0.222 µm/pixel and the max displacement to 0.5 and click “Calculate”. Now, you should see the result displayed in Figure 16 where blue and red dots represent particles found in channel 1 and 2, respectively, while yellow dots represent co-localized particles. In the top right of the GUI the number of particles found in each channel as well as the number of co-localized particles are shown. Additionally, the fraction of co-localized particles is calculated for each channel as the number of co-localized particles divided by the number of particles in the given channel.

# Spatial Pattern and Co-Localization Analysis by Batch Processing

For large datasets, SpatTrack provides the possibility to perform spatial pattern analysis and co-localization analysis in batch mode. Figure 17 shows the GUIs for single and two color batch processing in SpatTrack. Batch processing requires a stack of images (possibly denoised) which would normally be used for particle tracking as well as a corresponding stack of mask images. Before running the analysis the user must set the required paraemeters for particle tracking in the lower left of the GUI and select the type(s) of analysis.

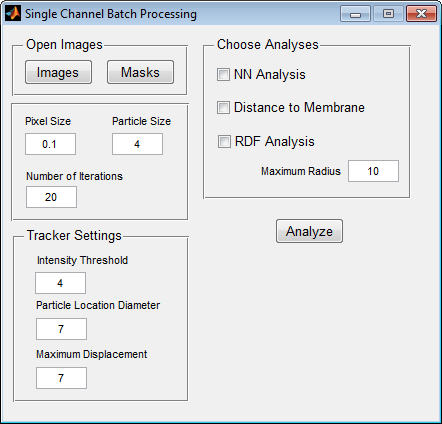
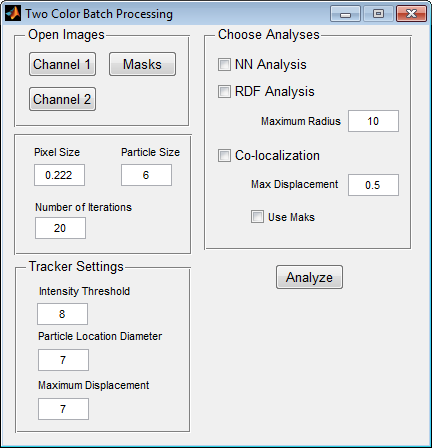
 

Figure : GUIs for single and two color batch processing.

Subsequently, the program runs the following for each image 1) detect coordinates for particles in the given image, 2) using the mask and the detected coordinates calculate the spatial statistics for the image, 3) save the results in memory until all images have been analyzed. Finally, the program outputs a number of comma separated txt files, as below.

**Output from single color analysis:**

1. particle detection: x- and y-coordinates and intensities for each image (in three separate files)
2. NN-analysis: A single file containing the following columns; col 1 – 4: min NN, max NN, mean NN and standard deviations for experimental particles, col 5 – 8: min NN, max NN, mean NN and standard deviations for random particles (these are the mean values obtained from “numTests” simulations) and cols 9 and 10: ANN and variance of ANN.
3. Number of particles as a function of the distance to the plasma for membrane for each cell and for random particle populations in each cell. These are reported in two separate files.
4. Radial distribution function g(r) for the particles in each cell.

**Output from two color analysis:**

1. x- and y-coordinates and intensities for particles detected in each cell in each channel in 6 idividual files.
2. results of NN-analysis in one file for each channel. e.g. for channel1 the following is returned

* min NN for channel 1 with respect to channel 2
* max NN for channel 1 with respect to channel 2
* mean NN for channel 1 with respect to channel 2
* Standard deviation of NN for channel 1 with respect to channel 2
* min NN for channel 1 with respect to random particles
* max NN for channel 1 with respect to random particles
* mean NN for channel 1 with respect to random particles
* Standard deviation of NN for channel 1 with respect to random particles

1. RDF analysis for each channel in the same format as for single-color analysis.
2. Results of co-localization in the following format  
   col1: number of particles detected in chanel 1  
   col2: number of particles detected in chanel 2  
   col3: number of co-localized particles  
   col4: fraction of co-localized particles in channel 1  
   col5: fraction of co-localized particles in channel 1

# Simulation of Diffusing Particles

SpatTrack includes add-ons for simulating one or two populations of diffusing particles moving by normal diffusion, diffusion plus flow or anomalous confined diffusion. These tools allow the user to get a feel for the different types of motion and to test different hypotheses from simulated images. The GUI for simulations of one population is shown in Figure 18, while Figure 19 shows the GUI for simulations of two populations.

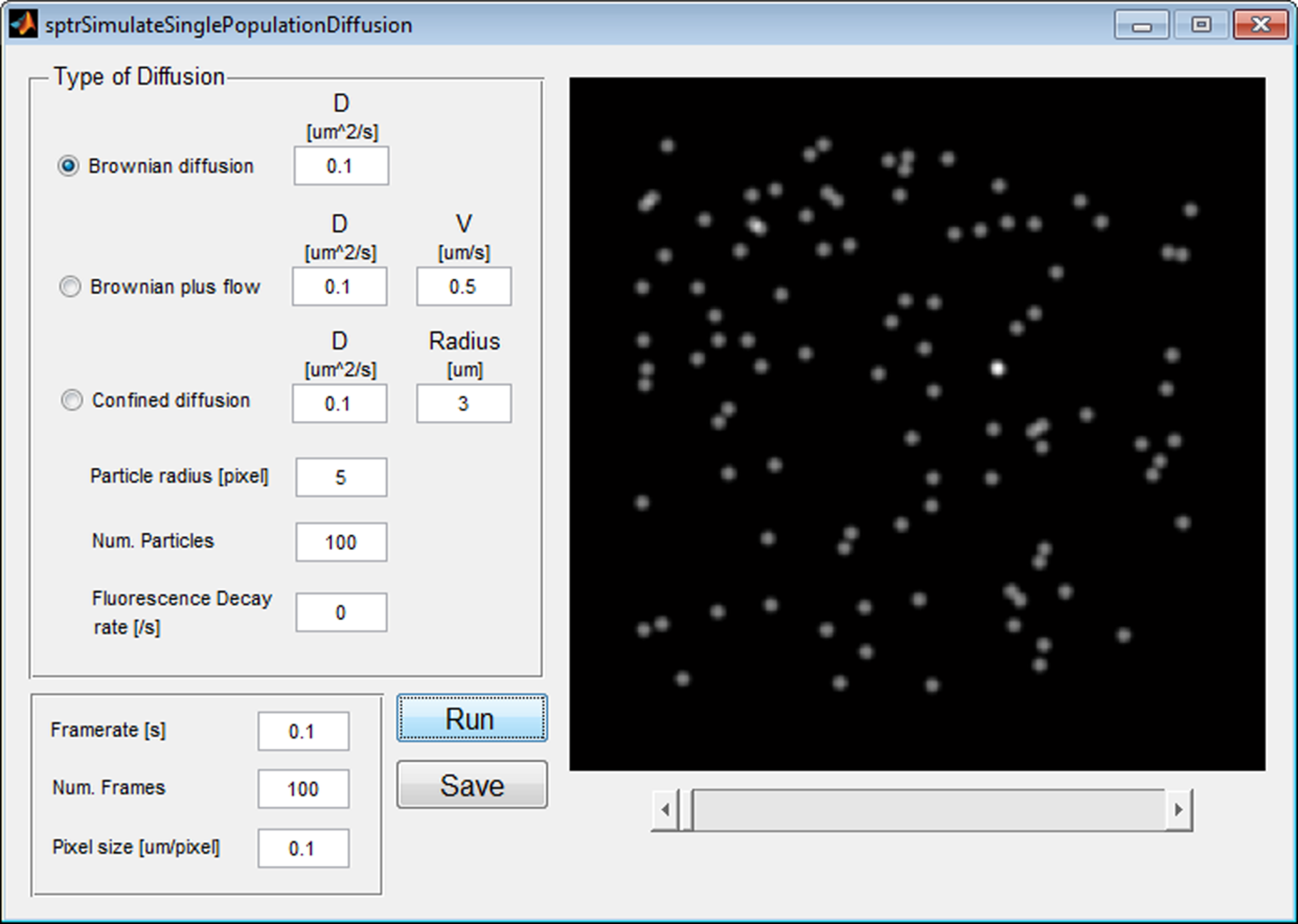


Figure : Add-on for simulation of one particle population moving by normal diffusion, diffusion plus flow or anomalous confined diffusion. Here 100 particles moving by normal diffusion with a diffusion constant of 0.1 µm2/s was simulated with a frame rate of 0.1 second (i.e., 10 frames/second) in an image stack with a pixel size of 0.1 µm and a total of 100 frames.

# 

Figure : Add-on for simulation of two different particle populations moving by diffusion. Here population 1 consisted of 50 particles moving by normal diffusion with a diffusion constant of 0.1 µm2/s while population 2 consisted of 50 particles moving by diffusion plus flow with a diffusion constant of 0.1 µm2/s and a velocity of 0.5 µm/s. The image stack consisted of 100 frames simulated with a frame rate of 0.1 s (i.e., 10 frames/second) and a pixel size of 0.1 µm/pixel. The particles were simulated such that the fluorescence intensity of population 2 was twice as high as the fluorescence intensity of population 1.

Once the simulation is completed, the generated image may be saved together with the particle intensities and x- and y-coordinates. Table 1 and Table 2 show the files generated for the simulation of or two particle populations, respectively.

Table : Overview of files generated for simulation of 1 particle population moving by diffusion where <image\_name> is the file name chosen by the user.

|  |  |
| --- | --- |
| Filename | File content |
| <image\_name>.tif  <image\_name>\_intensities.txt  <image\_name>\_xCoords.txt  <image\_name>\_yCoords.txt | Image stack of simulated particles  Fluorescence intensity of particles as a function of time  x-coordinates of particles as a function of time  y-coordinates of particles as a function of time |

Table : Overview of files generated for simulation of 2 particle populations moving by diffusion where <image\_name> is the file name chosen by the user.

|  |  |
| --- | --- |
| Filename | File content |
| <image\_name>.tif  <image\_name>\_ints\_combined.txt  <image\_name>\_ints\_pop1.txt  <image\_name>\_ints\_pop2.txt  <image\_name>\_xCoords\_combined.txt  <image\_name>\_xCoords\_pop1.txt  <image\_name>\_xCoords\_pop2.txt  <image\_name>\_yCoords\_combined.txt  <image\_name>\_xCoords\_pop1.txt  <image\_name>\_xCoords\_pop2.txt | Image stack of simulated particles  Fluorescence intensity of population 1 and 2 particles as a function of time  Fluorescence intensity of particles in population 1 as a function of time  Fluorescence intensity of particles in population 2 as a function of time  x-coordinates of population 1 and 2 particles as a function of time  x-coordinates of particles of population 1 as a function of time  x-coordinates of particles of population 2 as a function of time  y-coordinates of population 1 and 2 particles as a function of time  y-coordinates of particles of population 1 as a function of time  y-coordinates of particles of population 2 as a function of time |

# References

1. Lund FW, Larsen MLV, Christensen T, Nielsen GK, Heegaard CW and Wüstner D (2014) SpatTrack: an imaging toolbox for analysis of vesicle motility and distribution in living cells. Traffic. 12: 1406-1429.

2. Crocker JC, Grier DG (1996) Methods of Digital Video Microscopy for Colloidal Studies. Journal of Colloid and Interface Science 179: 298-311.

3. Lund FW, Wüstner D (2013) A comparison of single particle tracking and temporal image correlation spectroscopy for quantitative analysis of endosome motility. Journal of Microscopy doi: 10.1111/jmi.12080.

4. Zuiderveld, Karel. "Contrast Limited Adaptive Histograph Equalization." Graphic Gems IV. San Diego: Academic Press Professional, 1994. 474–485.

5. Pisano ED, Zong S, Hemminger BM, DeLuca M, Johnston RE, et al. (1998) Contrast limited adaptive histogram equalization image processing to improve the detection of simulated spiculations in dense mammograms. J Digital Imaging 11: 193-200.