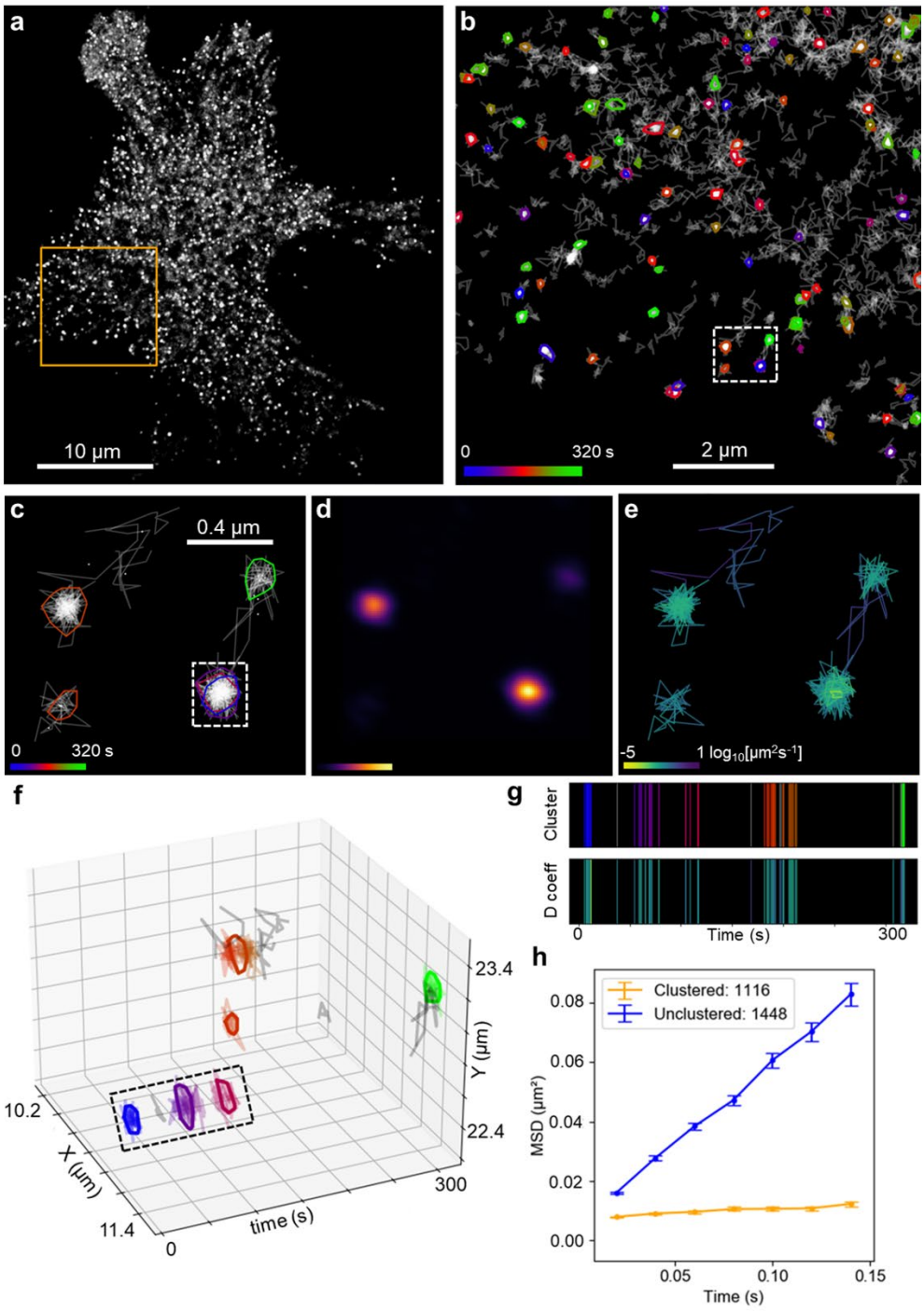
NASTIC USER MANUAL



Tristan Wallis t.wallis@uq.edu.au

Alex McCann a.mccann2@uq.edu.au

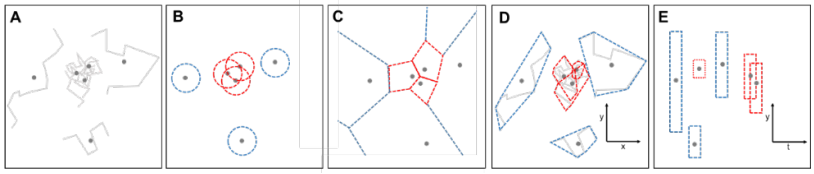
Single Molecule Neuroscience Laboratory (Fred Meunier)

Queensland Brain Institute

The University of Queensland, Australia

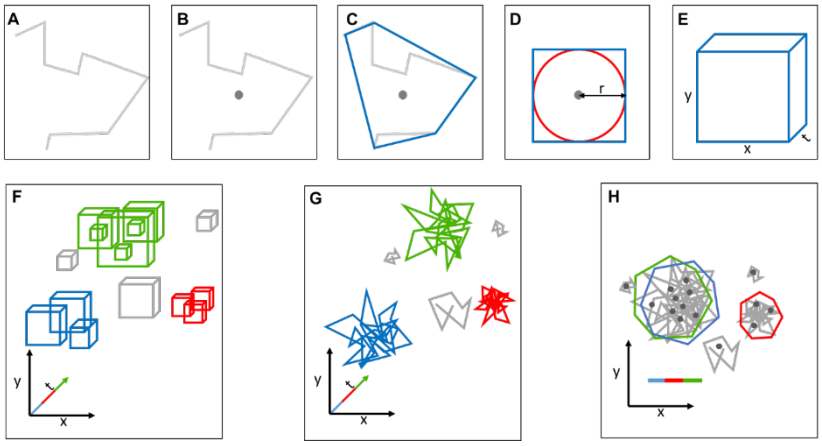
Introduction

NASTIC = nanoscale spatiotemporal indexing clustering, which represents the QBI Single Molecule Neuroscience lab's take on molecular clustering in live cells. NASTIC works by establishing whether entire molecular trajectories overlap in both space and time, and then building up clusters based on this potential overlap. NASTIC takes a different approach than widely used clustering algorithms which rely on DBSCAN or Voronoï tessellation to establish thresholds for molecular detections. NASTIC also differs markedly from these algorithms in that it also determines whether molecules are interacting in time.



Schematic representation of clustering algorithms. (A) Molecular trajectory data, with each trajectory spatial centroid indicated with a dot. **(B)** DBSCAN. Multiple molecule centroids present within a defined radius are considered clustered (red circles). The most effective radius (**ε**) and minimum number of centroids within it (MinPts) are determined empirically. **(C)** Voronoï tessellation. Tiles are drawn around each centroid such that the distance from any point within the tile is closer to its centroid than to any other centroid. Molecular centroids with tile areas less than an empirically determined threshold (red) are considered clustered. **(D)** Spatial indexing. Clustered molecules are determined by overlapping 2D bounding regions (red) defining the spatial extent of each molecular trajectory. **(E)** Spatiotemporal indexing. Each trajectory bounding region is assigned an arbitrary "thickness" in the time dimension. Overlapping 3D bounding regions represent spatiotemporally clustered molecules.

NASTIC relies on the R-tree spatial indexing algorithm, which is widely used in databases, mapping and especially videogames, where it is used to establish whether in-game objects, such as bullets and bad guys, are interacting. R-tree and other -tree algorithms such as Quad-tree, Oct-tree etc are highly optimised, which allows large numbers of interactions to be calculated quickly in order to not slow the game down unduly. In NASTIC we leverage this ability to rapidly establish whether trajectory bounding boxes are overlapping.



Schematic representation of spatiotemporal indexing clustering workflow. (A) Molecular trajectory composed of individual detections. (**B)** Spatiotemporal centroid representing the trajectory's average position in space and time. **(C)** Convex Hull (blue) defining the approximate spatial extent of the trajectory. **(D)** Simplified 2D spatial bounding box (blue square) based on the approximate radius (r) of the Convex Hull (red circle). **(E)** 3D spatiotemporal bounding box of user defined "thickness" in the time dimension. **(F)** R-tree spatiotemporal index of all trajectory bounding boxes. Discrete clusters of overlapping bounding boxes are indicated in colour, unclustered boxes in grey. **(G)** 3D clusters of trajectories associated with overlapping bounding boxes. **(H)** 2D representation of clustered trajectories. Coloured polygons represent the spatial Convex Hull of all detections comprising each of the clustered trajectories. Clusters are coloured according to the averaged detection time of their component trajectories, allowing assignment of overlapping clusters (green and blue) occupying the same spatial extent at different times.

NASTIC allows us to determine not only whether a trajectory overlaps with another trajectory in space, but also whether the overlap occurs in time. The idea being that trajectories which overlap spatiotemporally may represent clusters of molecules interacting at a certain space and time on the plasma membrane. Additionally, spatiotemporal metrics also allow us to

determine whether clusters repeatedly form and reform on the same region of a plasma membrane – hotspots. The temporal component of NASTIC literally gives us another dimension of molecular dynamics data to play with.

NASTIC was conceived by Fred Meunier and Tristan Wallis during the first great Coronavirus lockdown of 2020, and subsequently prototyped and implemented as a Python GUI by Tristan Wallis 2020-2021. Sophie Hou assisted with aspects of GUI debugging, and Kye Kudo provided additional bug fixing and parameter optimisation. Kyle Young contributed vector autoregression code, and Alex McCann contributed data format translation code and extensive debugging, code optimisation and GUI refinements. Rachel Gormal and Merja Joensuu provided valuable testing-to-destruction feedback and feature requests, and Nela Durisic provided critical MSD input. None of us are trained computer science graduates! So, while very effort has been made to create a functional, useful and stable program, bugs and missing features are bound to crop up. Your input with bug reports and feature requests for both the program and this documentation is very important to help NASTIC mature.

A considerable amount of effort has gone into wrapping the complexity of spatiotemporal analysis into a reasonably intuitive and functional interface. This manual should further help you to become a world leader in spatiotemporal cluster analysis. In addition, NASTIC has popup help for most of its buttons and fields which should help you further. If in doubt, just contact us at f.meunier@uq.edu.au and t.wallis@uq.edu.au

Computer requirements

NASTIC is a Python script and requires Python 3.8 or later to run. Python is available for most computer platforms so you can run NASTIC on Windows, Linux and Mac, or even on a Raspberry Pi if you were so inclined. NASTIC will not run on the older version of Python 2.7 which is still lingering on a lot of computer systems. You are strongly encouraged to either visit https://www.python.org and download and install the latest version* or hassle someone in IT to do it for you. You/they will also need to install a number of Python modules, which NASTIC uses to do a lot of the heavy lifting. This is simple to do from a command line.

The Python module versions used in the most recent version of NASTIC are:

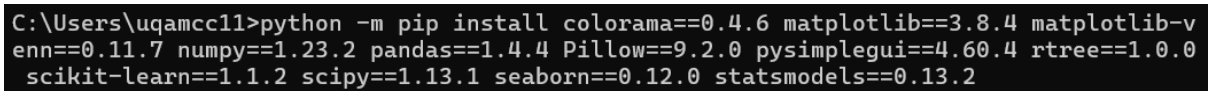
scipy	v1.13.1
numpy	v1.23.2
matplotlib	v3.8.4
matplotlib-venn	v0.11.7
pysimplegui	v4.60.4
rtree	v1.0.0
scikit-learn	v1.1.2
statsmodels	v0.13.2
colorama	v0.4.6

In addition to these Python modules, other NASTIC family members require the following Python modules with the most recent versions shown:

pandas	v1.4.4
pillow	v9.2.0
seaborn	v0.12.0

To install these Python module versions, copy paste the following into the command line:

```
python -m pip install colorama==0.4.6 matplotlib==3.8.4 matplotlib-venn==0.11.7 numpy==1.23.2 pandas==1.4.4 Pillow==9.2.0 pysimplegui==4.60.4 rtree==1.0.0 scikit-learn==1.1.2 scipy==1.13.1 seaborn==0.12.0 statsmodels==0.13.2
```



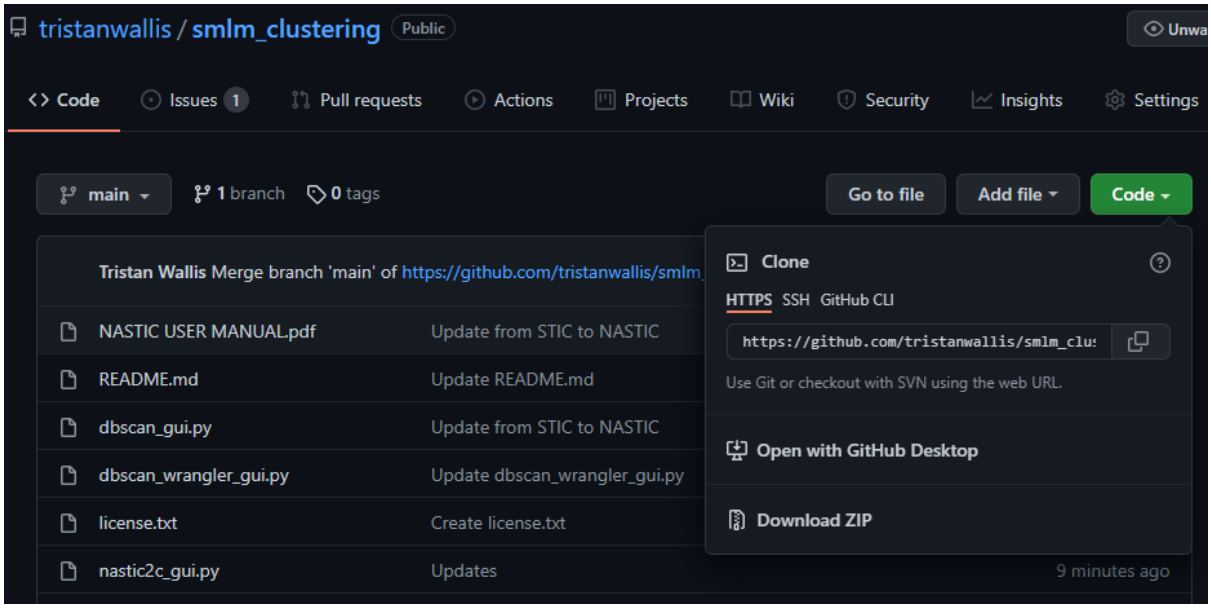
NASTIC is a multithreaded application which means that the intensive number crunching is farmed out to however many cores your computer has. You'll therefore theoretically get better performance on a 4Ghz computer with 8 cores than you will on a 5Ghz computer with 4 cores. But even on modest i5 laptop such as mine most analyses should take less than a minute.

IMPORTANT: Because of the multithreading, NASTIC does not play properly on virtual computers which dynamically assign the user virtual cores depending on usage etc. You will need to run it on a physical computer. Single threaded versions of NASTIC (NASTIC_ST) and its derivatives are also available, which trade slower trajectory preprocessing for the ability to run on virtual architecture. NASTIC_ST can also be compiled using PyInstaller, which will generate a standalone executable that can be run on any computer regardless of having Python installed or not.

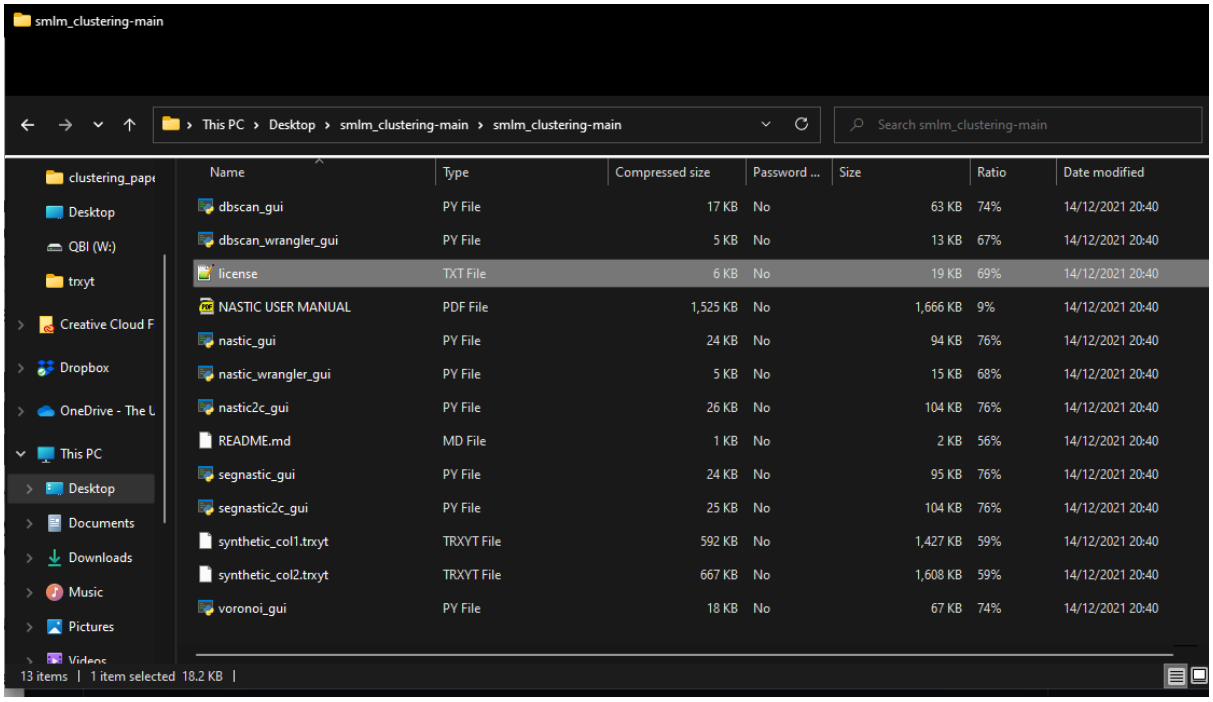
***ALSO IMPORTANT:** Be aware that if you want to use the latest bleeding edge version of Python, there's a chance that not all modules will have been compiled for it. The pip commands as detailed above may fail because your computer will try to unsuccessfully build the modules.

Obtaining NASTIC

The latest greatest versions of NASTIC and related clustering software are always available on Github: https://github.com/tristanwallis/smlm_clustering. The total download is just a few Mb so it should only take 10 seconds to download and ensure that you're not running a superseded version. Just click on the green "Code" button and then "Download zip".



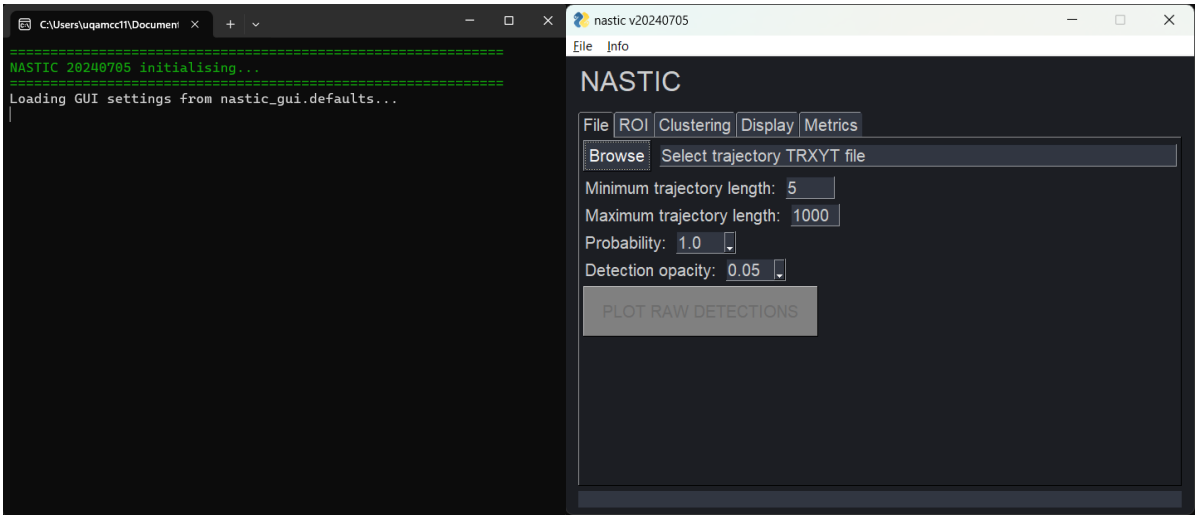
This will then download smlm_clustering_main.zip



Extract these files into the directory of your choice and you should be good to go!

Running NASTIC

Provided that you've installed Python and the required modules correctly, double clicking on **NASTIC_gui.py** should open a command shell, load a brief splash screen and then the main program itself. The command shell is where NASTIC will print various useful information as it runs. If you close the shell, you'll close NASTIC too. NASTIC will display its version number (just the datestamp of its last changes) in the window title, so you'll always know what version you are running.



Alternatively, NASTIC can be opened using the command line by navigating to the location of the **NASTIC_GUI.py** script and typing the below followed by the return key:

```
python nastic_gui.py
```

In order to simplify the potentially confusing clustering workflow, NASTIC is divided into a number of functional tabs, which it will automatically switch to as appropriate:

- **File:** select trajectory data, screen it based on trajectory length, and display the raw detections
- **ROI:** draw one or more rectangular or free hand regions of interest on the raw detection display (or load in previously saved ROIs generated by NASTIC (.tsv files) or PalmTracer (.rgn files)) and select encompassed trajectories with optional density screening
- **Clustering:** enter the appropriate parameters for spatiotemporal clustering
- **Display:** plot the results of the clustering with a large range of options for trajectory, centroid, cluster and hotspot display, and optionally export high resolution images for publication
- **Metrics:** visualise and save a wide range of spatiotemporal clustering metrics

Each of these tabs will be detailed below.

File Tab

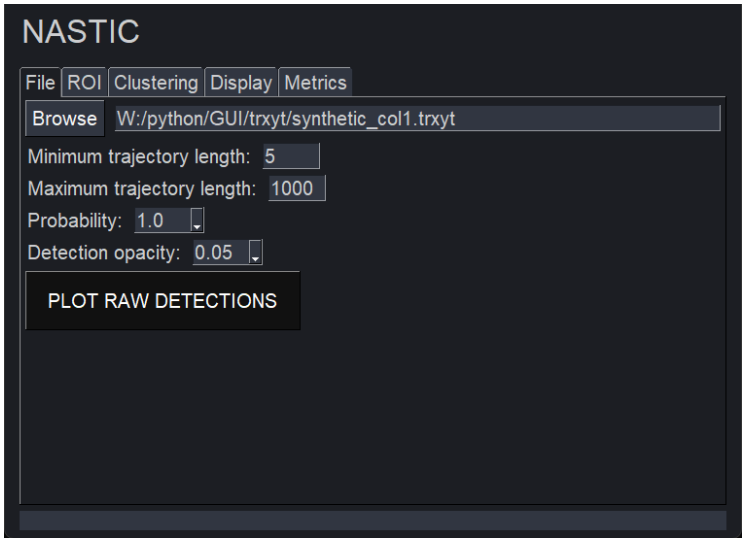
This tab allows you to browse for and select an input trajectory file and display its raw detections.

NASTIC works on simple text files where each line contains the info for a single detection: TRajectory number, X co-ordinate (µm), Y co-ordinate (µm) and Time (sec), separated by spaces (not commas or tabs). These files should have a **trxyt** suffix.

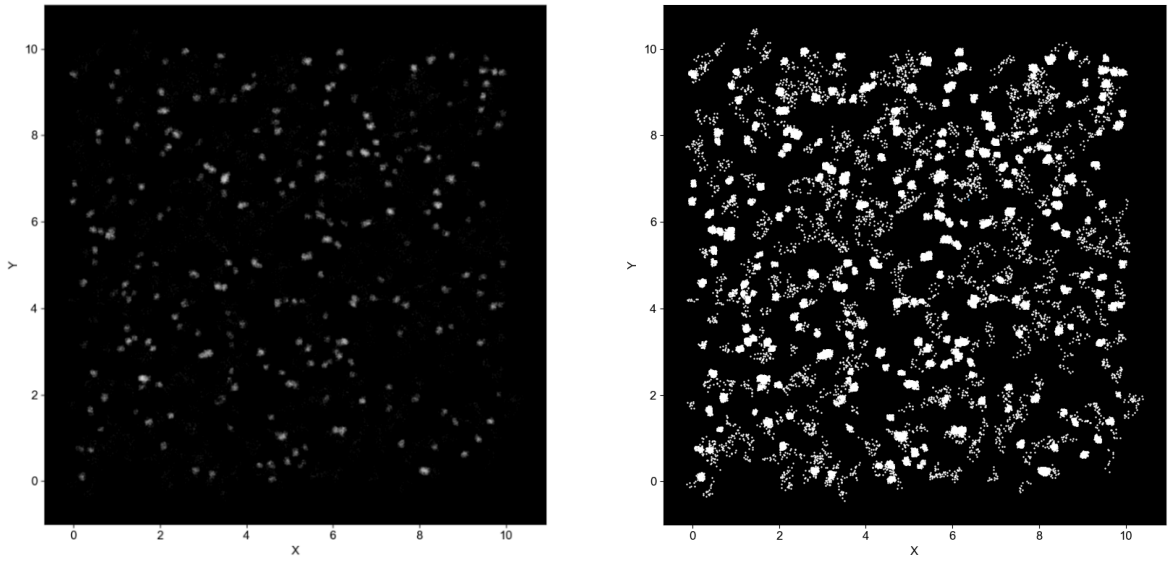
TRXYT files should contain no headers or comments, just the data:

```
1 9.0117 39.86 0.02
1 8.9603 39.837 0.04
1 9.093 39.958 0.06
1 9.0645 39.975 0.08
2 9.1191 39.932 0.1
2 18.9266 39.915 0.12
Etc
```

Clicking the **"Browse"** button will open a file selector which will allow you to choose TRXYT files (only). Once you've selected a file its path will be displayed, and you can then opt to change the length filters which by default screen out really short (largely uninformative) trajectories and really long ones (which are usually background noise).



In order to plot the raw data contained within the selected TRXYT file, NASTIC needs to know what % of the raw detections to use (default Probability = 1.0 → 100%) and how opaque each detection should be on the plot. If you have high density data then probability and opacity values that are too high might result in a solid white plot which could make subsequent ROI assignment tricky. The below images were generated using 0.05 and 1.0 opacity respectively:



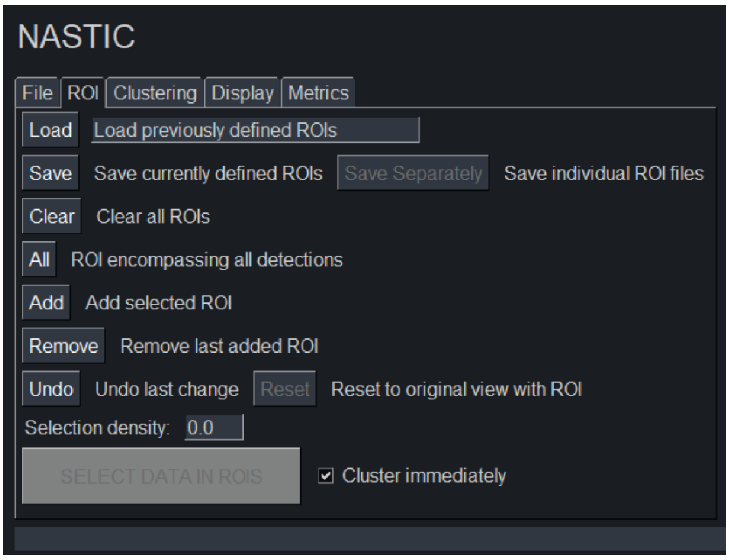
Clicking **"PLOT RAW DETECTIONS"** will plot the raw detections, and the program will automatically switch to the **ROI** tab. You won't be able to click **"PLOT RAW DETECTIONS"** until you have specified an input file. In fact, all of the buttons in NASTIC will only be active at the appropriate stages of the analysis.

IMPORTANT NOTE: It appears that some MATLAB processing of trajectory data converts trajectory numbers > 99999 into scientific notation with insufficient decimal points (e.g. 102103 to 1.0210e+05, 102104 to 1.0210e+05). This can cause multiple trajectories to be incorrectly merged into a single trajectory. For trajectories > 99999 NASTIC empirically determines whether detections are within 0.32u of each other, and assigns them into a single trajectory accordingly. For trajectories <99999 it honours the existing trajectory number.

IMPORTANT NOTE 2: Selecting a probability of < 1 will result in fewer detections being displayed, but internally NASTIC is still working with the entire set of detections and will select them according to your subsequent ROI. So, if you load a dataset of 10000 detections comprising 1000 trajectories and select a probability of 0.5, only 5000 detections will be displayed. If you subsequently use an ROI encompassing all detections, NASTIC will then select the original 10000 trajectories.

IMPORTANT NOTE 3: A range of other trajectory filetypes can be converted to the TRXYT format using the **Super Res Data Wrangler** (see Super Res Data Wrangler User Manual). Additionally, synthetic TRXYT files can be generated using **Synthetic** (see Synthetic User Manual). Two example synthetic TRXYT files have been included with NASTIC for reference.

ROI TAB



This tab allows you to select single or multiple regions of interest on the raw detections plot.

Clicking **"Load"** will allow you to load in saved ROIs from previous NASTIC analyses (.tsv files), or saved ROIs generated by PalmTracer analyses (.rgn files).

The units for the X and Y co-ordinates in NASTIC .tsv files are in µm, whereas PalmTracer .rgn files are in pixels and will therefore need to be converted to µm using the appropriate conversion factor.

The conversion factor used to convert PalmTracer .rgn X and Y co-ordinates from pixels to µm for use in NASTIC will depend upon the pixel size (in µm) of the image. For example, if one pixel is equal to 0.106 µm in a particular image, all X and Y pixel co-ordinates would need to be mulitpled by a conversion factor of 0.106 to obtain their values in µm.

If the selected ROI file is a PalmTracer .rgn file, the option to convert pixels to µm using a conversion factor will appear **"Pixel(um):"**.

The **"Replot ROIs"** button will also appear, which enables the user to redraw the .rgn ROI using a different conversion factor.

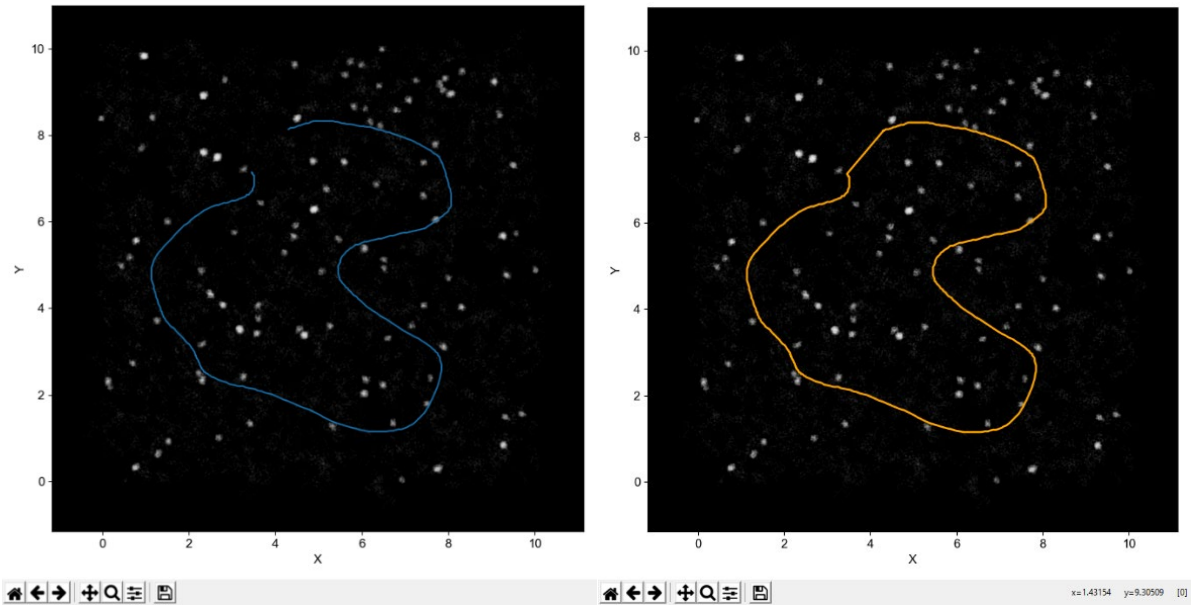
These options will dispear once another ROI type is selected (e.g., a .tsv file is loaded, or an ROI is drawn in NASTIC).

Clicking **"Save"** will allow you to save the current ROI(s) to a single datestamped YYYYMMDD_roi_coordinate.tsv file, within a folder that is generated in the same directory as the selected trxyt file with following format: trxytfilename_NASTIC_ROIs_YYYYMMDD-HHMMSS. If you are doing a complex series of ROIs, I recommend saving them as you go to prevent potential aggravation. ROI files will also be saved automatically in a folder called "saved_ROIs" when you save the analysis later on.

Alternatively, the **"Save Separately"** button, which becomes availble after at least two ROIs have been selected, can be used to save ROIs as individual files. These files are saved using the format: YYYYMMDD_roi_coordinate#.tsv, with # corresponding to ROI number (starting at 0 and increasing with each ROI), in the trxytfilename_NASTIC_ROIs_YYYYMMDD-HHMMSS folder. Individual ROI files will also be saved automatically in the "saved_ROIs" folder, together with an ROI file that contains all ROIs, when the analysis is saved later on.

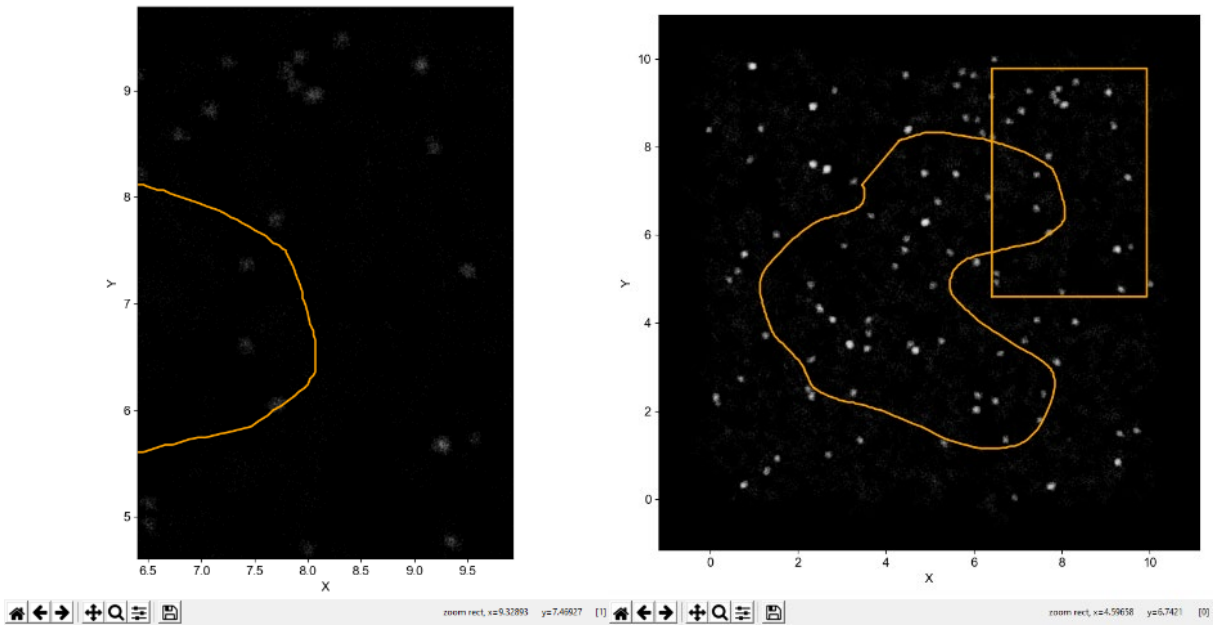
Defining your own ROIs can be done either by freehand drawing, or drawing a rectangular shape. You can add as many ROIs as you'd like, and it doesn't matter if they overlap as the trajectories will only be selected once in overlapping areas.

NASTIC defaults to using the freehand selector. On the raw detections plot, just hold down the mouse and draw. You don't have to close the shape. Clicking the **"Add"** button in the GUI will close the shape and add the ROI in yellow to the raw detections plot.



To switch to a rectangle selector, just click on the little magnifying glass icon on the bottom of the display window.

The mouse cursor will change to a crosshair. On the raw detections plot, hold and drag the mouse to select the region you want, and when you release it the display will zoom in on the selected area. Click the **"Add"** button to add this zoomed area and return to the full sized image.



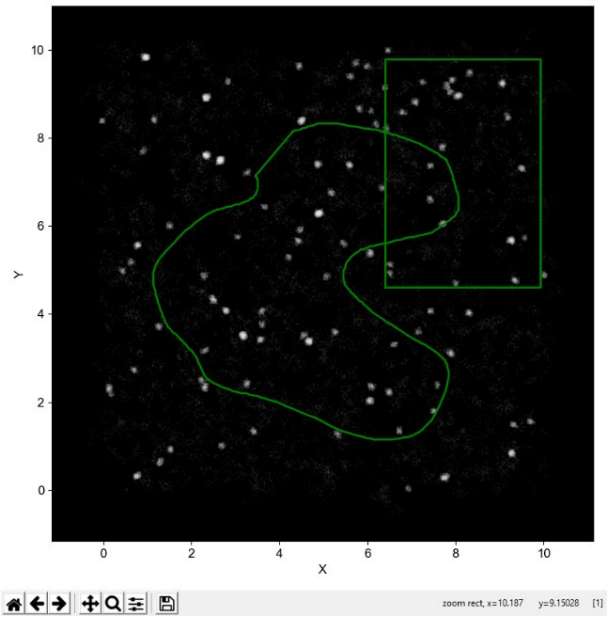
Clicking the **"Remove"** button will allow you to sequentially remove the last added ROI. You can click it more than once to remove more ROIs. This also applies to ROIs loaded in using the "Load" button.

"Clear" will completely clear the ROIs.

"All" will create a single ROI which encompasses the full spatial extent of all the detections.

"Undo" will undo the last change you made to the ROIs. If you have 5 laboriously selected ROIs and accidentally press "Clear" or "All", you can click this button and get back to those 5 ROIs.

You can load a previous set of ROIs, and add more ROIs, etc etc etc. Basically there is a lot of flexibility to allow you to just analyse the areas you think are important. Once you've decided on the perfect set of ROIs, click the **"SELECT DATA IN ROIS"** button, which will now be active. If the "Cluster immediately" box is ticked, trajectories will be selected in ROIs and immediately clustered using pre-defined parameters in the "Clustering" tab. If the box is unticked, then the trajectories will just be selected. The ROIs will turn green to indicate this.



You'll also see a figure appear next to the **"Selection density"** box in the GUI, this is the number of selected trajectories per μm^2. You can decrease this value by entering the value you want in the **"Selection density"** box. Clicking **"SELECT DATA IN ROIS"** will reselect the trajectories and randomly drop some to bring the number down close to your entered value. E.g. for the above analysis you could enter 15, and the system will return 14.89 trajectories/ μm^2. This feature can be useful if analysing data from multiple acquisitions with varying trajectory density, as it can allow you to adjust all acquisitions to a common density. Be aware that randomly dropping trajectories means that you will get slightly different clustering results using the same data and the same selection density override. If you enter 0 in the box and then click on **"SELECT DATA IN ROIS"**, it will just go back to selecting everything.

NASTIC will automatically switch to the **Clustering** tab once trajectories have been selected.

If no trajectories are found in the selected ROI(s), then pressing **"SELECT DATA IN ROIS"** will not progress to the next tab.

Following clustering, you can return to the original view of the plotted raw data with the ROIs that were selected for clustering shown, by returning to the **ROI** tab and pressing the "**Reset**" button.

Clustering Tab

The values on this tab are at the heart of R-tree based spatiotemporal clustering.

By default, NASTIC uses sensible values, which generally give good results with synthetic datasets which represent the kind of density and clustering typically seen in PC12 sptPALM experiments.

Acquisition time: The length of the acquisition in seconds. The program works internally using seconds, not frames.

Frame time: The time between each frame in sec. In our lab the data is typically acquired at 50 Hz, so each frame is 0.02 s (20 ms). A 320 s acquisition is thus 16000 of these frames.

Time threshold: This is the time “thickness” in seconds that defines whether trajectories are considered to be interacting in time. A time threshold of 20 sec means that two clusters are considered as clustered if they overlap in space, and are within 20 seconds of each other. Each trajectory will be examined to determine if there are other trajectories within the same spatial extent up to 10 seconds earlier, and up to 10 seconds later. If you set a time threshold of 640 seconds, then NASTIC will essentially only report spatial clustering (NASIC?) and will return clusters more or less the same as DBSCAN or Voronoï.

Radius factor: Because of the need for the R-tree database to use regular rectangular regions, NASTIC does not use the convex hull of the trajectory, but rather calculates the ideal radius of the convex hull and builds a box based on this. Unless the trajectory moved within a perfectly circular area, this box does not fully represent the extent of the molecule. So we can expand the box to better account for this. A value of 1.2 works well for starters. You can take it up to 1.5, possibly even 2 if you have a lot of directed trajectories or low-density trajectories. But this will of course potentially start showing more spurious overlap. Alternatively, if you are using very high density data, you can use a radius factor of <1. If your data is really high density, you should consider either changing the selection density in the ROI tab, using MSD filtering as described below, or perhaps using **segNASTIC**, which clusters based on trajectory segment overlap rather than whole trajectory bounding box overlap. Of note, the **2-in-1 GUI (NASTIC/segNASTIC)** can be used to easily switch between the two analyses.

Cluster threshold: How many trajectories have to overlap in order to be considered a cluster. 3 is a good start. In high density data you may want to consider increasing this. Before you start complaining, be aware that a cluster threshold like this is at the core of DBSCAN and Voronoï too.

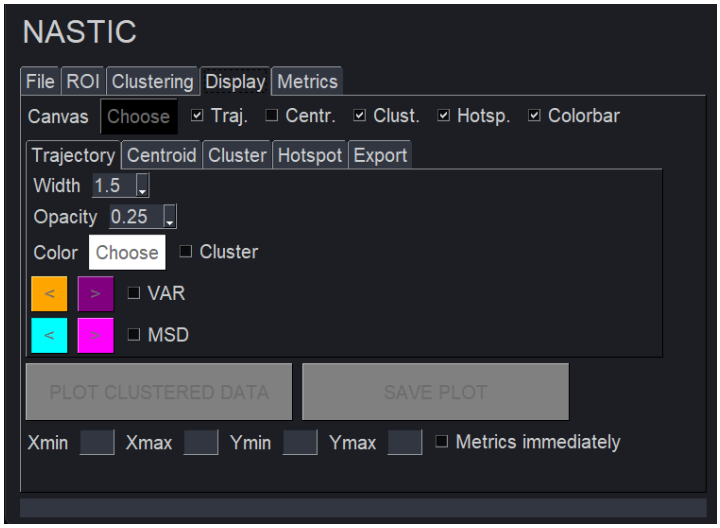
Cluster size screen: The general consensus is that protein nanoclusters are usually less than 0.1 µm in radius. NASTIC clusters much larger than this are more likely to be artifacts of random overlap of higher mobility trajectories. These can be screened out by setting this value

appropriately. If you set it much below 0.15 μm you might risk throwing out valid clusters, so use this wisely.

MSD screen: This setting may be useful to screen out high mobility background trajectories which could otherwise contribute to large diffuse clusters. Ticking this box will cause NASTIC to measure the mean square displacement at timepoint zero (MSD[0]) for all trajectories, and establish the average MSD. It will then reject trajectories whose MSD is greater than the average. Use this carefully – for datasets with lots of low mobility trajectories the average MSD will be lower, and you might end up discarding trajectories with low mobility but still higher than the average MSD.

"CLUSTER SELECTED DATA" will apply your values and use spatiotemporal indexing clustering on the selected trajectories. If **"Plot immediately"** is ticked then the clusters will be automatically displayed using pre-determined parameters in the "Display" tab after the analysis is complete.

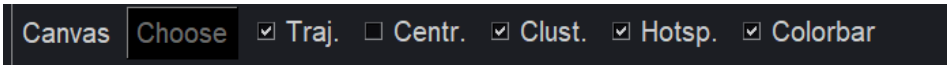
Display Tab



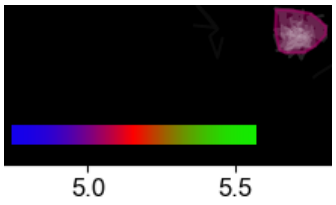
NASTIC gives you a very large amount of flexibility to display your clustered data, and therefore has a few extra subtabs that you can access. At any time, you can change various parameters and hit the **"PLOT CLUSTERED DATA"** button to update the display.

You can use the controls on the bottom of the displayed cluster plot to zoom in (make sure you've selected the magnifying glass and changed the cursor to a crosshair). The arrows can be used to move backwards or forwards through the various zooms (e.g., switching between a zoomed in rectangle and the original full view). The home icon can be used to go back to full view (this will reset the plot to the full view and will therefore remove any previously drawn rectangle zooms previously accessed by using the arrows).

In the GUI, choose the background colour of the plot, and whether you want trajectories, centroids, clusters and hotspots displayed.

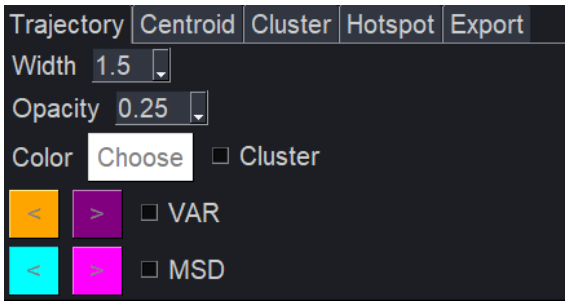


The **"Colorbar"** checkbox will plot a linear colorbar onto the bottom left-hand corner of the display. For a 320 sec acquisition: 0 sec = blue, 160 sec = red and 320 sec = green. You will need to click the **"PLOT CLUSTERED DATA"** button to make the colorbar appear after changing zoom.



Trajectory subtab:

The appearance of the plotted trajectories can be altered in the "Trajectory" subtab.



Width: thickness of plotted trajectory lines

Opacity: 0 = completely transparent, 1 = solid

Color: all trajectories (clustered and unclustered) will be plotted in this color.

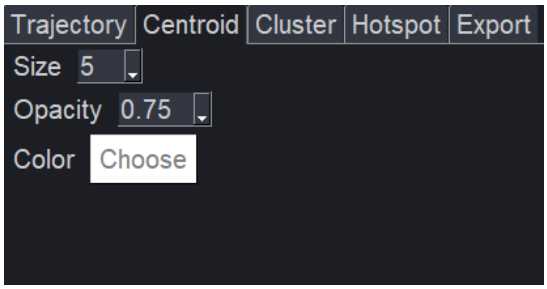
Cluster: ticking this checkbox will color clustered trajectories according to their time (i.e., earlier clustered trajectories will be blue, later clustered trajectories green). Only unclustered trajectories will remain colored according to the specified color.

VAR: ticking this box will color trajectories based on their "confinement" as established by vector autoregression. Click on the "<" arrow to change the color of "confined" trajectories, and on the ">" arrow to change the color of "unconfined" trajectories.

MSD: ticking this box will color trajectories based on whether their mean square displacement (MSD) is below the average MSD of all trajectories or above the average. Click on the "<" arrow to change the color of trajectories below the average MSD, and on the ">" arrow to change the color of trajectories above the average MSD.

VAR and MSD should colour trajectories in a broadly similar way, and can be useful in establishing whether clusters really do correspond to less mobile/more confined trajectories. VAR and MSD options will override the trajectory color/cluster settings.

Centroid subtab:



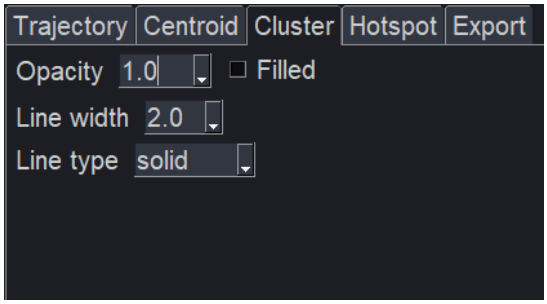
Size: how large the trajectory centroid will appear

Opacity: 0 = completely transparent, 1 = solid

Color: Use this option to change centroid color. All centroids will have the same color, regardless of whether they are in clusters or not.

Cluster subtab:

Each cluster represents a convex hull boundary around the greatest extent of the component detections. The color of the cluster represents its approximate time in the acquisition, as detailed above.



Opacity: 0 = completely transparent, 1 = solid

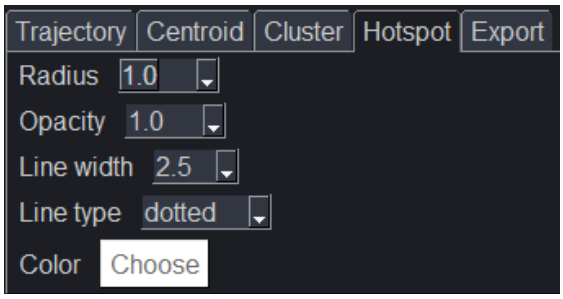
Filled: The cluster will be filled with solid color at half the opacity.

Line width: how thick the trajectory border lines are

Line type: Style of the trajectory border line

Hotspot subtab:

Hotspots represent regions where clusters occupy the same spatial region, but at different times. This tab will allow plotting a convex hull encompassing the detections of all overlapping clusters.



Radius: NASTIC calculates the average radius of all clusters. Setting radius to 1 means that two clusters whose centroids are closer than the average radius will be considered as a hotspot. <1 = clusters have to be closer than the average radius, >1 = clusters can be further apart than the average radius

Opacity: 0 = completely transparent, 1 = solid

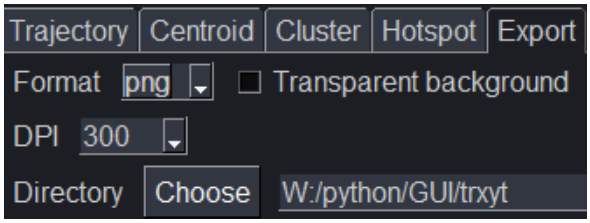
Line width: thickness of the hotspot border line

Line type: Style of the hotspot border line

Color: all hotspots will be plotted in this color.

Export subtab:

At any time, you can export a high-quality file representing the contents of the main plot window, by clicking the **"SAVE PLOT"** button. These images may be more suitable for publication etc.



Format: eps, pdf, png, ps, svg. Depending on what software you want to use for figure wrangling, png is a common lossless bitmap format which will open in anything.

Transparent background: The image will ignore the background canvas color and will be output with a transparent background. This could come in handy if you want to make a fancy image with a gradient background for instance.

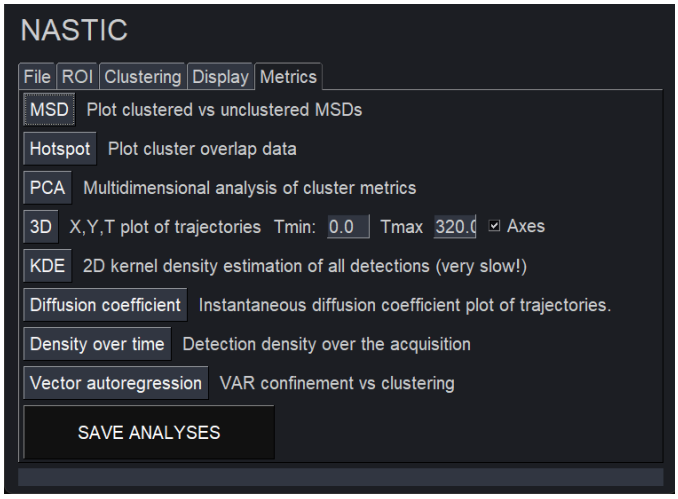
DPI: Dots per inch. 50 = lower resolution, 1200 = very high resolution but very large file size

Directory: Where to save the image.

Rather than use the mouse to zoom in on a region, you can specify it exactly by entering the corner co-ordinates for the zoom box. This can come in handy if you are e.g. comparing NASTIC, DBSCAN and VORONOÏ (the GUIs of these programs all allow you to zoom like this) of a particular dataset and want to visualise how the clusters differ in the same area. Hit **"PLOT CLUSTERED DATA"** to update the image once you've entered the co-ordinates. You can enter 0 in all boxes to revert to the full plot of all selected trajectories.

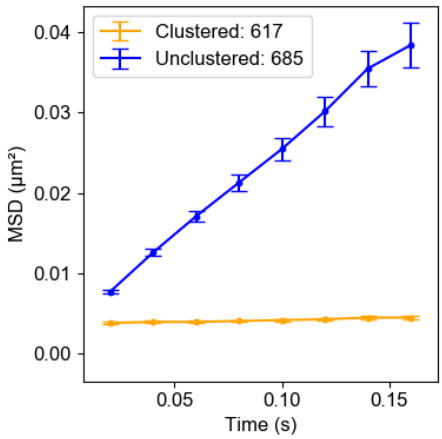
"Metrics immediately" will automatically switch to the Metrics tab once the plot has completed. This can be useful if you are analysing large numbers of TRXYTs all with the same parameters, and don't care much about the plotted cluster image. Ticking **"Cluster immediately" + "Plot immediately" + "Metrics immediately"** and clicking **"SELECT DATA IN ROIS"** in the ROI tab will select trajectories, cluster them, plot the output, and immediately switch to the Metrics tab so that you can save the analyses or examine them further.

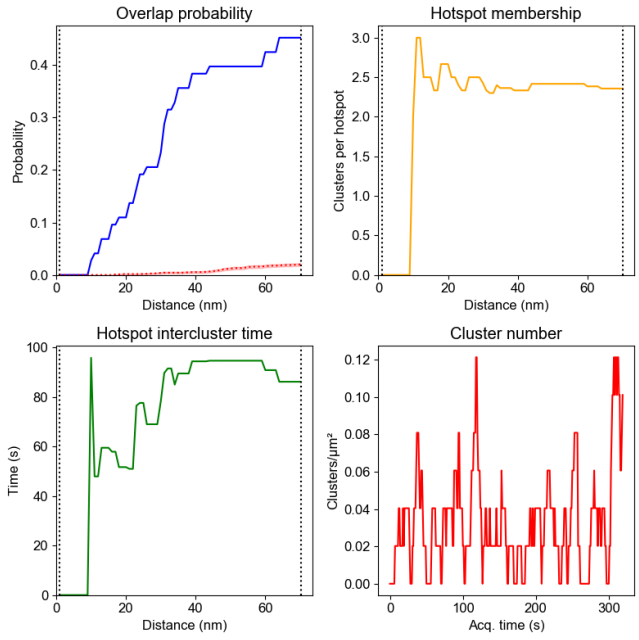
Metrics Tab



This tab will allow you to examine various cluster metrics to allow further assessment of the data. Because NASTIC does temporal clustering, many of these metrics are not available when using DBSCAN or Voronoï.

MSD: Will plot Mean square displacement vs time for clustered trajectories and unclustered trajectories. Ideally we want to see clustered detections with lower mobility as per the picture on the right.

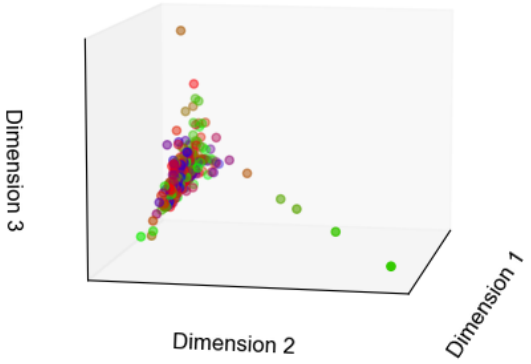




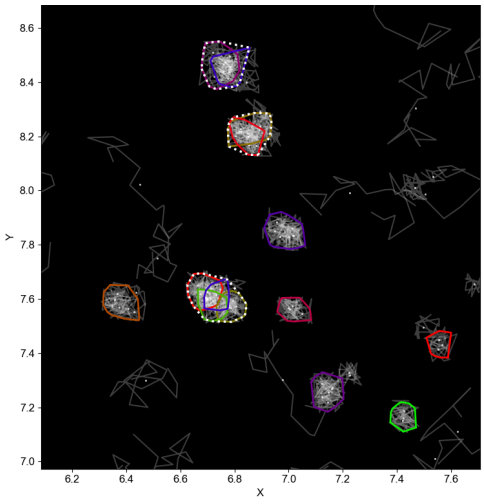
Hotspot: Because we can get spatial and temporal clustering information using NASTIC, we are able to determine the presence of “hotspots” where clusters repeatedly form and dissociate on the same region. Overlap probability indicates the likelihood of this. Firstly, the average radius of all clusters is determined. Next, the spatial centroid of each cluster is determined. DBSCAN is then performed multiple times using an epsilon between 0 and the average cluster radius. At each epsilon the number of centroids with another centroid within epsilon is calculated, and converted to a probability such that 1 = all centroids have a neighbor within epsilon. In the figure on the left approximately 45% of all clusters have another cluster within 70nm (the average cluster radius), and ~ 20% have a cluster within ½ the average radius. Remember, they may overlap in space, but occupy a different point in time. Additional information can be extracted from the DBSCAN clusters, such as the number of clusters in each hotspot, and the time between these clusters. The

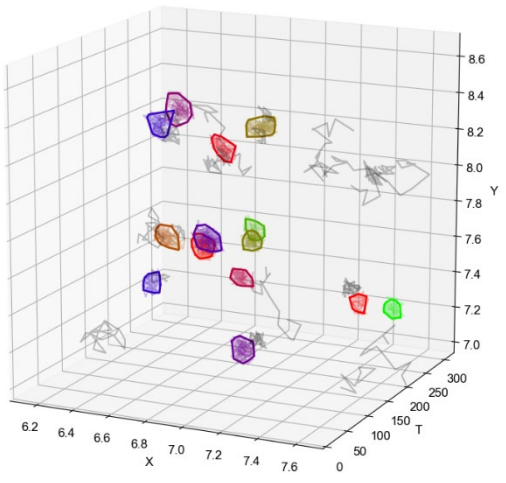
cluster number plot allows you to determine whether the overall number of clusters at a given point in time changed over the course of the acquisition. This could be useful in a system where cells were stimulated ½ way through an acquisition for instance.

PCA: Each cluster is assigned multiple metrics such as size, number of trajectories, average MSD, lifetime etc etc. Dimensionality reduction using principal component analysis (PCA) allows you to determine whether there are subpopulations of clusters with distinctly different metrics. Usually this is not the case, but one could imagine a situation where an analysis of a cell with two distinct membrane regions resulted in a differently behaving clusters. These might show up as distinct groups of dots on the PCA plot. The PCA plot is colored according to the average acquisition time of the cluster, so that the user can potentially establish whether cluster metrics change over time.



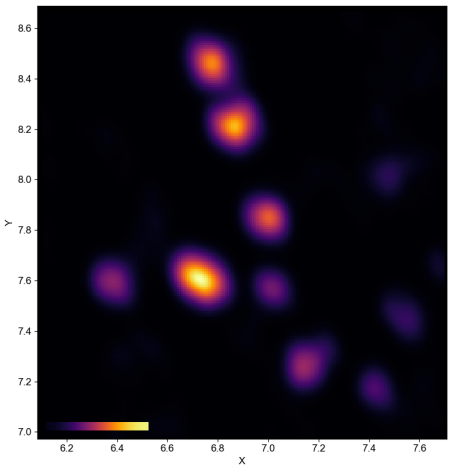
3D: NASTIC works by treating super-res data as 3 dimensional [x, y, time]. In the final 2D plots, the time dimension is represented as color. If you click on the “3D” the trajectories on the main display plot will be rendered into 3D space, which will not only allow you to get a better feel for how the clustering picked up things in space and time, but might also make for some pretty pictures for publication. By default the 3D plot displays trajectories over the full acquisition time window. **Tmin** and **Tmax** can be used to adjust the time window to display, allowing you to zoom in on a particular time period. Unticking the **Axes** box will display the 3D plot on a black background with no grid or axis labels.

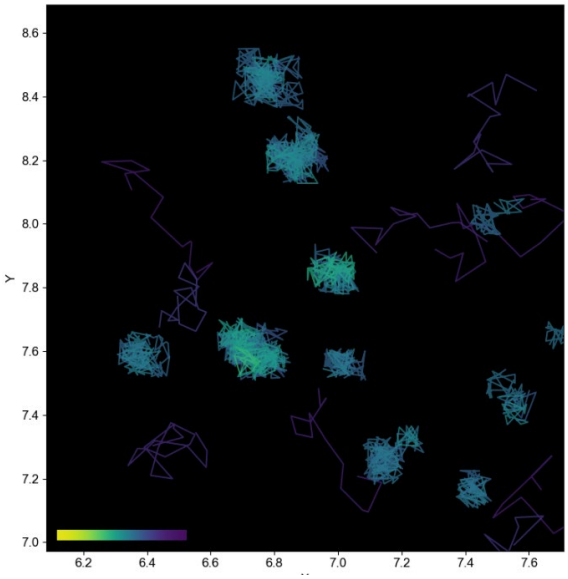




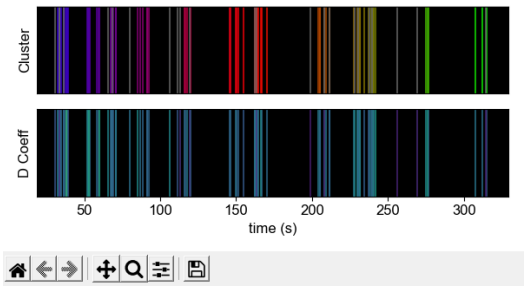
The 3D plot inherits its zoom, line, centroid and cluster display properties from the Display tab. It does not plot cluster hotspots. You can rotate the 3D plot by holding the left mouse button and moving the mouse appropriately. You can zoom in or out by holding the right mouse button and moving the mouse up or down.

KDE: Kernel density estimation is basically a 2D histogram of the density of the detections, which shows regions of higher density as a brighter color. Not only does it look pretty but it can be very useful for determining whether your clusters correspond to regions of higher detection density. This is not always the case however, since it's possible for a single trajectory with many detections to show up on KDE, but not be selected for NASTIC. KDE analysis can be very slow with lots of detections so please don't use it to analyse your entire image - start with a 2 µm x 2 µm region and see how long that takes before increasing things.



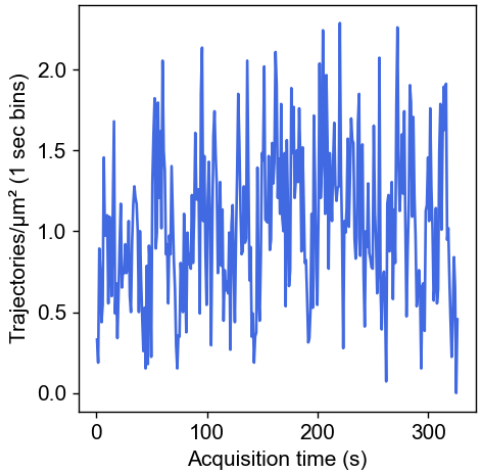


Diffusion coefficient: NASTIC calculates the instantaneous diffusion coefficient as the gradient of the first 4 steps of the MSD for each trajectory. Plotting this can help to visualise whether clusters consist of lower mobility (low diffusion coefficient) trajectories against a background of higher mobility unclustered detections.



The data is also plotted as 1D, so that you can determine whether trajectories with low diffusion coefficients are present in clusters.

Density / time:

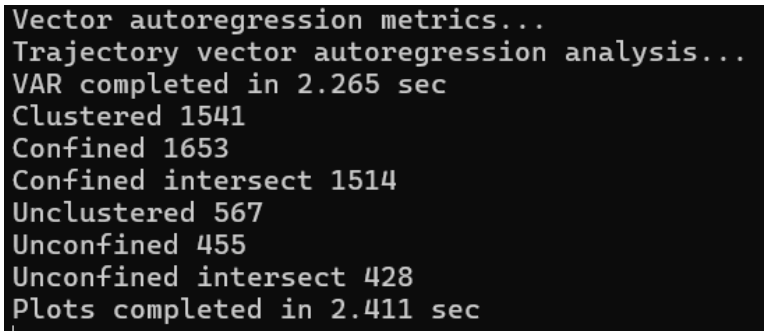


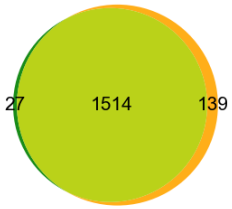
Divides the acquisition into 1 second bins (e.g., a 320 sec aqcquisition creates 320 bins) and counts the trajectories in each bin. This allows rapid determination of whether the number of molecular detections remains constant over the analysis.

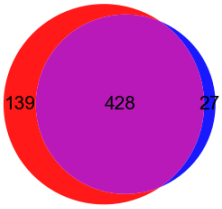
For acquisitions where the conditions change at some point (e.g., stimulation of PC12 cells), this might result in recruitment of additonal molecules to the plasma membrane, which can be visualised with this metric.

Vector autoregression:

This will run vector autoregression analysis to determine whether trajectories are “confined” or “unconfined”, and will then plot two Venn diagrams, showing the overlap of clustered and confined trajectories (top plot) and unclustered and confined trajectories (bottom plot). The details are printed into the shell.







“SAVE ANALYSES” will save any open plot windows, as well as the raw detections plot with the ROI shown, the ROI coordinates themselves (both as a single file containing all ROIs, and as individual files for each ROI, in the “save_ROIs” subfolder), and a TSV file (**metrics.tsv**), containing the various analysis metrics to a datestamped folder in the same directory as the TRXYT you analysed. The file can be opened in any text editor or Excel.

The top lines of the metrics.tsv file contain an overview of the analysis:

NASTIC: NANOSCALE SPATIO TEMPORAL INDEXING CLUSTERING - Tristan Wallis t.wallis@uq.edu.au	
TRAJECTORY FILE:	W:/python/GUI/trxyt/synthetic_col1.trxyt
ANALYSED:	20240725-142637
TRAJECTORY LENGTH CUTOFFS (steps):	5 - 1000
SELECTION DENSITY:	0
ACQUISITION TIME (s):	320
FRAME TIME (s):	0.02
TIME THRESHOLD (s):	20
RADIUS FACTOR:	1.2
CLUSTER THRESHOLD:	3
CLUSTER MAX RADIUS (um):	0.2
MSD FILTER THRESHOLD (um^2):	None
SELECTION AREA (um^2):	111.8869
SELECTED TRAJECTORIES:	2108
CLUSTERED TRAJECTORIES:	1541
UNCLUSTERED TRAJECTORIES:	567
VAR CONFINED TRAJECTORIES:	1653
VAR UNCONFINED TRAJECTORIES:	455
TOTAL CLUSTERS:	269
CLUSTERED TRAJECTORIES AVERAGE INSTANTANEOUS DIFFUSION COEFFICIENT (um^2/s):	0.00566
UNCLUSTERED TRAJECTORIES AVERAGE INSTANTANEOUS DIFFUSION COEFFICIENT (um^2/s):	0.26903
HOTSPOTS (CLUSTER SPATIAL OVERLAP AT 1/2 AVERAGE RADIUS):	14
TOTAL CLUSTERS IN HOTSPOTS:	28
AVERAGE CLUSTERS PER HOTSPOT:	2
PERCENTAGE OF CLUSTERS IN HOTSPOTS:	10.409

The MSD curve data is also tabulated should the user wish to plot in Prism etc:

MSD CURVE DATA:							
TIME (S):	0.02	0.04	0.06	0.08	0.1	0.12	0.14
UNCLUST MSD (um^2):	0.004916261	0.008227591	0.011649468	0.014824236	0.018202581	0.021589937	0.024744567
UNCLUST SEM:	6.92E-05	0.000200777	0.000353356	0.00052163	0.000711869	0.000912965	0.001127653
CLUST MSD (um^2):	0.003385254	0.00337221	0.003404058	0.003440583	0.003448916	0.00351042	0.003503516
CLUST SEM:	2.34E-05	2.71E-05	3.59E-05	4.54E-05	6.01E-05	7.29E-05	8.62E-05
ALL MSD (um^2):	0.003989181	0.005287481	0.006656573	0.00793102	0.009268701	0.010642133	0.011882339
ALL SEM:	3.50E-05	9.66E-05	0.000167116	0.000241649	0.00032539	0.000412598	0.000503848

The remainder of the file contains the individual and averaged metrics for the clusters identified during the analysis:

INDIVIDUAL CLUSTER METRICS:								
CLUSTER	MEMBERSHIP	LIFETIME (s)	AVG MSD (um^2	AREA (um^2)	RADIUS (um)	DENSITY (traj/ur	RATE (traj/sec)	AVG TIME (s)
1	4	4.34	0.003045833	0.014944422	0.068970699	267.658392	0.921658986	273.22
2	4	13.39	0.002938636	0.010071418	0.056620067	397.1635301	0.298730396	118.3475
3	3	3.36	0.002403054	0.004695381	0.038659878	638.925795	0.892857143	62.22666667
4	5	3.72	0.003565336	0.013906241	0.066531901	359.5508055	1.344086022	54.134
5	7	5.05	0.003329512	0.016727002	0.072968281	418.4850396	1.386138614	253.2128571
6	3	6.58	0.003079342	0.007711196	0.049543415	389.044693	0.455927052	65.48333333
7	3	7.96	0.002638875	0.007332106	0.048310265	409.1594154	0.376884422	80.73333333
8	8	9.19	0.003176084	0.018912975	0.077589863	422.9900427	0.870511425	162.43625
9	4	9.16	0.003954768	0.033445685	0.103179901	119.5968943	0.436681223	63.015
10	5	9.61	0.003301422	0.014065784	0.066912467	355.4725319	0.520291363	54.892
11	3	1.61	0.00347634	0.007758419	0.049694882	386.676738	1.863354037	221.1766667
etc	etc	etc	etc	etc	etc	etc	etc	etc
AVG	5.349344978	6.674366812	0.00337696	0.013613329	0.064836745	401.8836942	0.985935471	175.1764285
SEM	0.12320699	0.155182697	2.44E-05	0.000322412	0.000751864	5.646237537	0.082770292	5.97882413

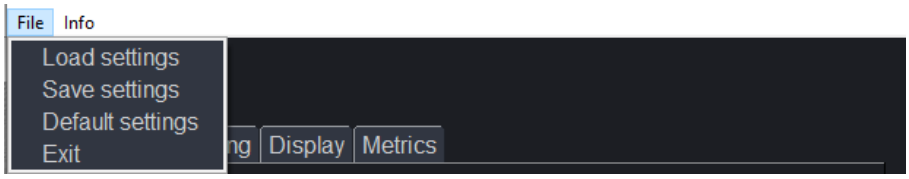
This data can be plotted as is, but the real power comes from being able to combine these metrics files from larger numbers of analyses across multiple conditions. This is handled by **NASTIC WRANGLER**, which is detailed in a separate document (See Nastic Wrangler User Manual).

Once you have finished analysing a given TRXYT, you can flick back to the **File** tab and load another one. This will automatically close all other open windows except the main display window. You should never need to close this window, but if you do the program will let you know all about it and will open a fresh window and reset the analysis.

Menus

In addition to the tab and button functionality of NASTIC, there are some menu items you should get acquainted with. These are located at the top left of the GUI.

File:



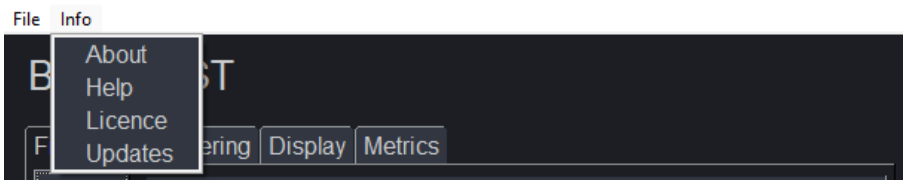
Trajectory probability	1
Raw trajectory detection plot opacity	0.05
Selection density	0
Trajectory minimum length	5
Trajectory maximum length	1000
Acquisition time (s)	320
Frame time (s)	0.02
Time threshold (s)	20
Radius factor	1.2
Cluster threshold	3
Canvas color	black
Plot trajectories	TRUE
Plot centroids	FALSE
Plot clusters	TRUE
Plot hotspots	TRUE
Plot colorbar	TRUE
Trajectory line width	1.5
Trajectory line color	white
Trajectory line opacity	0.25
Centroid size	5
Centroid color	white
Centroid opacity	0.75
Cluster fill	FALSE
Cluster line width	2
Cluster line opacity	1
Cluster line type	solid
Hotspot line width	2.5
Hotspot line opacity	1
Hotspot line type	dotted
Hotspot radius	1
Hotspot color	white
Plot save format	png
Plot save dpi	300
Plot background transparent	FALSE
Auto cluster	TRUE
Auto plot	TRUE
Cluster size screen	0.2
Auto metric	FALSE
MSD filter	FALSE
VAR color	FALSE
MSD color	FALSE
Cluster color	FALSE
3D axes	TRUE
VAR color confined	orange
VAR color unconfined	purple
MSD color < AVG	cyan
MSD color > AVG	magenta
Pixel size (um)	0.106

The first time you run NASTIC it will create a file called **NASTIC_gui.defaults** which contains the default values for the various steps of the analysis. You can load it into a text editor or spreadsheet to view it if you'd like.

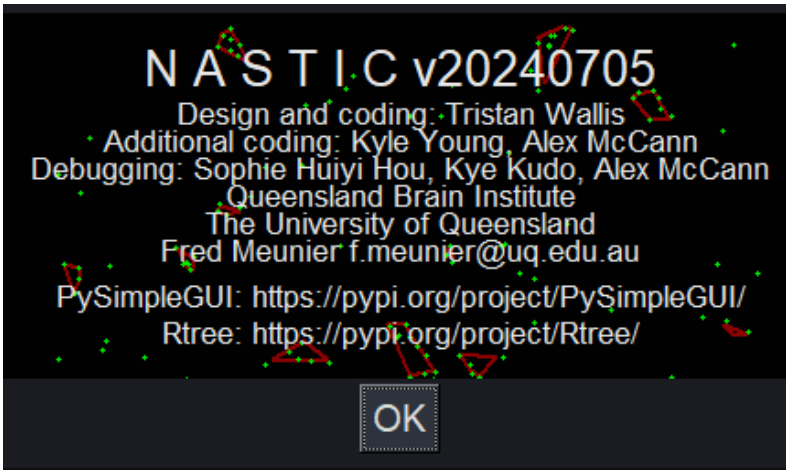
Nearly every value that you can change in NASTIC can be saved to this file. So if you arrive at a set of values that work well for a given set of data, **"File >> Save settings"** will save them to NASTIC_gui.defaults. The next time you start NASTIC, they'll be loaded in as the defaults. If you get too far out of whack, **"File >> Default settings"** will restore the defaults. But remember if you don't save them, the next time you load NASTIC you'll get your messed up values! Similarly, if you started with your preferred values, and changed multiple settings, **"File >> Load settings"** will restore your last saved settings.

IMPORTANT NOTE: Periodically NASTIC is updated with new functionality that requires a new **nastic_gui.defaults** file. Newer versions of NASTIC may crash if loading an older defaults file. You are encouraged to delete your existing defaults file and run the new version of NASTIC to recreate the correct file with default settings.

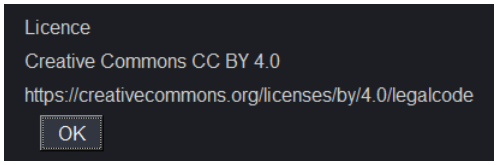
Info:



Clicking **"Info >> About"** will pop up this glorious little window with a real time molecular clustering simulation. Warning this is mesmerising to look at and could waste several minutes of your life. Why did I do this? Because I could.

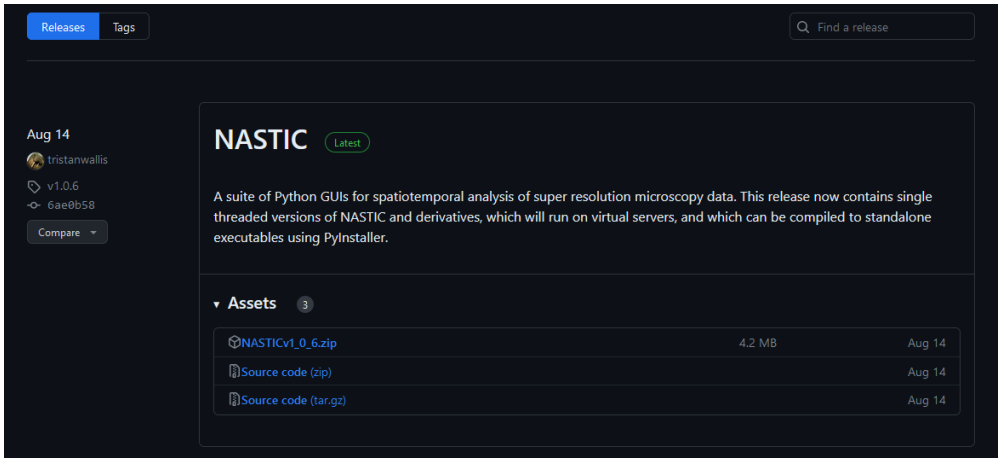


"Info >> Help" does nothing much at this stage, eventually it will link to a version of this document.



"Info >> Licence" reiterates the CC BY 4.0 concept embraced by NASTIC and allows the user to find out more about this powerful open source licence.

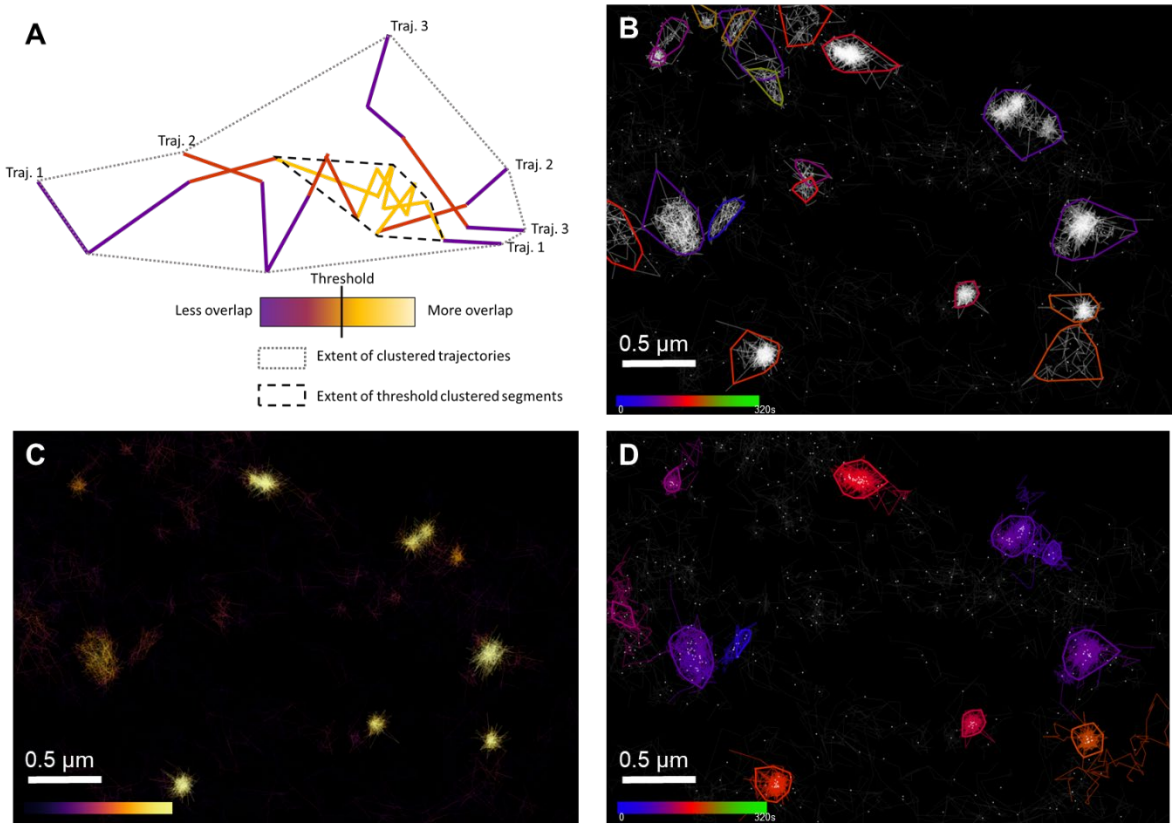
"Info >> Updates" will open the NASTIC GitHub release page in your default browser:



The latest release of NASTIC is always at the top of the page. Compare the date in your NASTIC window with the release date, and update as necessary. NASTIC development is ongoing, and you are strongly encouraged to keep up to date for bug fixes and feature improvements.

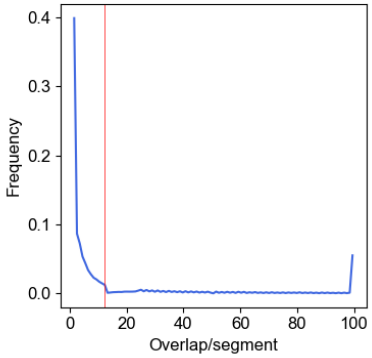
Appendix 1: Segment NASTIC (segNASTIC)

NASTIC works by establishing whether the bounding boxes of trajectories overlap in space and time. If trajectory density is very high it's possible that high mobility trajectories might still overlap, leading to spurious clustering.



NASTIC of trajectory segments (segNASTIC). (A) Schematic representation of trajectory segment thresholding, based on overlap with segments from other trajectories. **(B)** Syntaxin1a-EGFP imaged by uPAINT using Atto-647-labelled anti-GFP nanobodies in PC12 cells. Spatiotemporal clusters identified using spatiotemporal indexing of trajectory bounding boxes using ***r*** = 1.2 and ***t*** = 20 s. Each coloured cluster boundary represents the convex hull of the detections belonging to all trajectories in the cluster **(C)** Pseudo-density map of trajectory segment overlap, with each trajectory coloured according to the number of overlaps with other trajectory segments, as determined by spatiotemporal indexing of segment bounding boxes. **(D)** Spatiotemporal clusters identified using thresholded segments ***t*** = 20 s. Each coloured cluster represents the convex hull of detections belonging to the clustered segments. All trajectories containing clustered segments are shown in the same colour as the cluster.

Segment NASTIC attempts to get around this by clustering based on trajectory segment overlap. Trajectory segments which overlap with many other segments are more likely to represent areas of true molecular interaction. As the above figure shows, it more accurately returns areas of interaction in high density uPAINT data. The segment version of NASTIC is found as segNASTIC_gui.py, and double clicking it loads an interface similar to NASTIC.



The clustering parameters of segNASTIC are slightly different to allow for the fact that it is clustering based on segment overlap. Basically, segNASTIC analyses the overlap of all trajectory segments, and then works out what the average overlap is. Segments with overlap greater than this are considered as potentially clustered. It turns out that the overage overlap also corresponds to the inflection point on the distribution of overlap (red line)

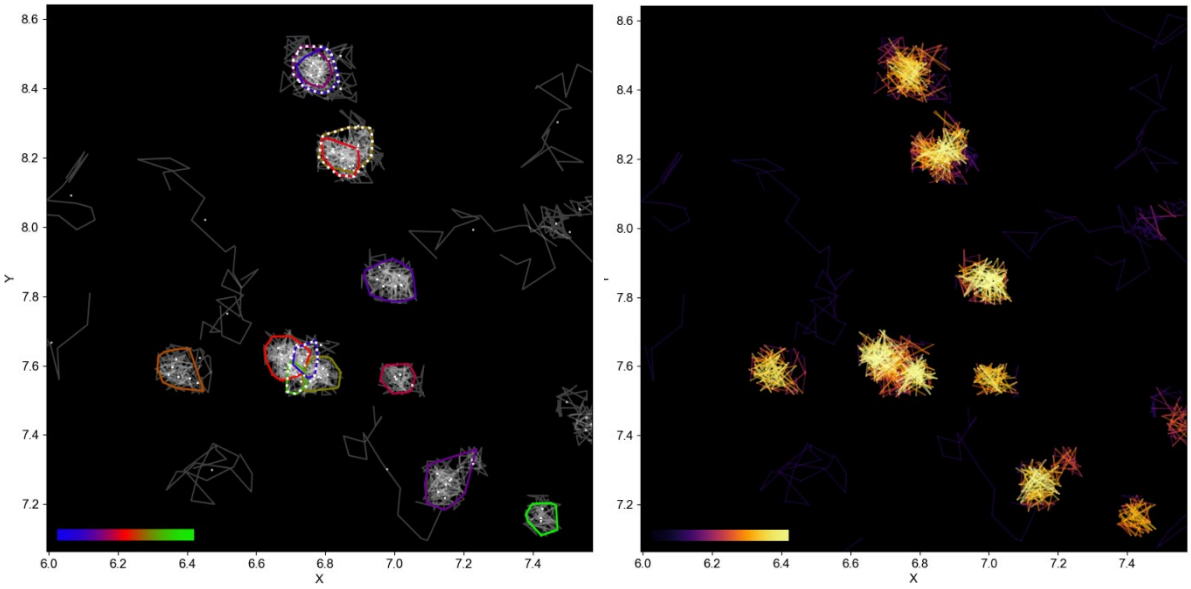
Segment threshold: This determines how many segments each trajectory must have with greater than the average overlap, in order for the trajectory to be considered as part of a cluster. By default it is 1, but can be set higher for greater stringency. Internally, segNASTIC requires three or more trajectories to contribute to a cluster.

Overlap threshold override: Yes, too many occurrences of "over". Basically, if this is set to 0, then the system will use the average overlap as a threshold. Otherwise, you can make it more or less stringent by entering your own value here. Obviously you can't know what the average overlap is until you've already performed an analysis using *overlap threshold override = 0*, in which case the threshold will be displayed in the shell:

```
Generating bounding boxes of segments in selected trajectories...
13759 segment bounding boxes generated in 0.498 sec
Total segment overlap...
13759 segments analysed in 1.919 sec. Average segment overlap (threshold): 13.91
1
Clustering thresholded segments...
41 clusters of 5043 thresholded segments analysed in 0.642 sec. Average segment
overlap: 59.813
Generating metrics of clustered trajectories...
41 unique spatiotemporal clusters identified in 0.088 sec
Plotting clustered trajectories...
Plotting unclustered trajectories...
Highlighting clustered trajectories...
Plot complete in 0.418 sec, please wait for display...
Plotting segment overlap pseudo density...
Segment overlap plot completed in 7.177 sec
Plotting histogram of segment overlap...
Segment histogram completed in 0.082 sec
```

In this case, you might want more stringent clustering and you could increase the threshold from 13.91 up to e.g. 20.

Once you've performed the analysis, the metrics are largely the same except for the inclusion of the "**Segment**" option, which allows you to plot the segment overlap density and the segment overlap distribution (as shown above).


8.6
8.4
8.2
8.0
7.8
7.6
7.4
7.2
Y
6.0
6.2
6.4
6.6
6.8
7.0
7.2
7.4
X

Appendix 2: 2-in-1 GUI (NASTIC/segNASTIC)

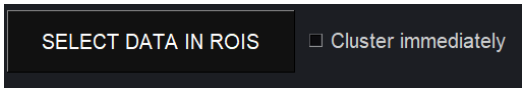
The 2-in-1 GUI combines NASTIC and segNASTIC into a single interface that enables the user to easily switch between NASTIC and segNASTIC analyses. It looks and functions in the same way as the other two GUIs, with a few minor differences.

To open, either double click on the **2in1_nastic_segnastic_gui.py** script, or navigate to the location of the script using the command line and type the below followed by the return key:

```
python 2in1_nastic_segnastic_gui.py
```

ROI tab:

The "**Cluster immediately**" checkbox is unticked by default in order to give the user a chance to swap analyses in the Clustering tab prior to clustering.



Clustering tab:

The default analysis is set to NASTIC. To change to segNASTIC, click on the dropdown box next to "**Select analysis type:**" and select SEGNASTIC.

The heading of the GUI and the relevant clustering parameters will change accordingly:

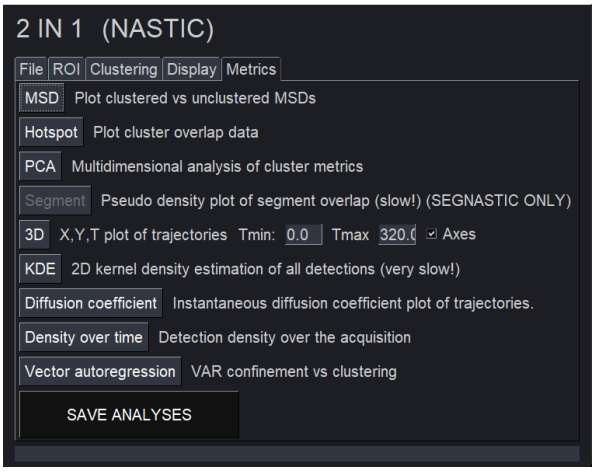
Upon pressing the "**CLUSTER SELECTED DATA**" button, the name of the selected analysis will appear as a header in the console, followed by the usual print statements.

```
============================================================
2 IN 1 (NASTIC)
============================================================
Generating bounding boxes of selected trajectories...
2108 bounding boxes generated in 1.485 sec
```

```
============================================================
2 IN 1 (SEGNASTIC)
============================================================
Generating bounding boxes of segments in selected trajectories...
35153 segment bounding boxes generated in 1.544 sec
```

Metrics tab:

The “Segment” metric will only become enabled if SEGNASTIC is selected as the analysis type, and will become disabled if NASTIC is selected:



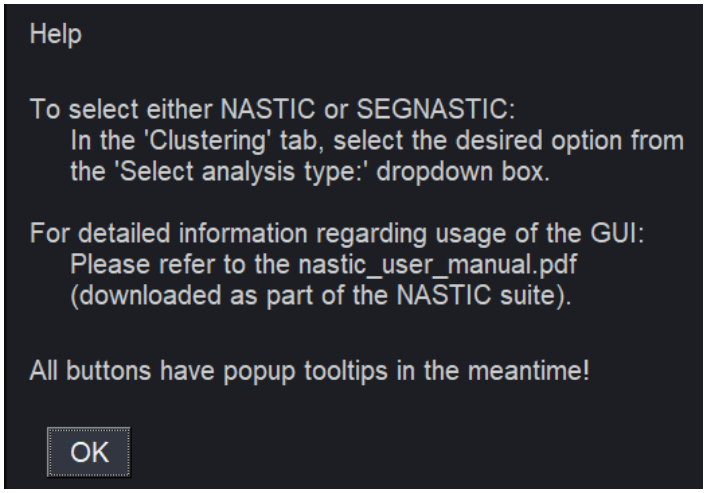
The “**SAVE ANALYSES**” button functions the same as for NASTIC and segNASTIC. The datestamped folder containing all of the saved files will have either “NASTIC” or “SEGNASTIC’ in the name depending on which analysis was used.

Defaults file:

The defaults file differs slightly from that of NASTIC and segNASTIC as it contains an additional parameter “Analysis type” which stores whether NASTIC or segNASTIC analysis has been selected, contains both NASTIC and segNASTIC clustering parameters, and has the “Auto cluster” parameter set to FALSE.

Info:

The 2-in-1 GUI has a basic Help popup that can be accessed by clicking “Info >> Help” in the Menu.



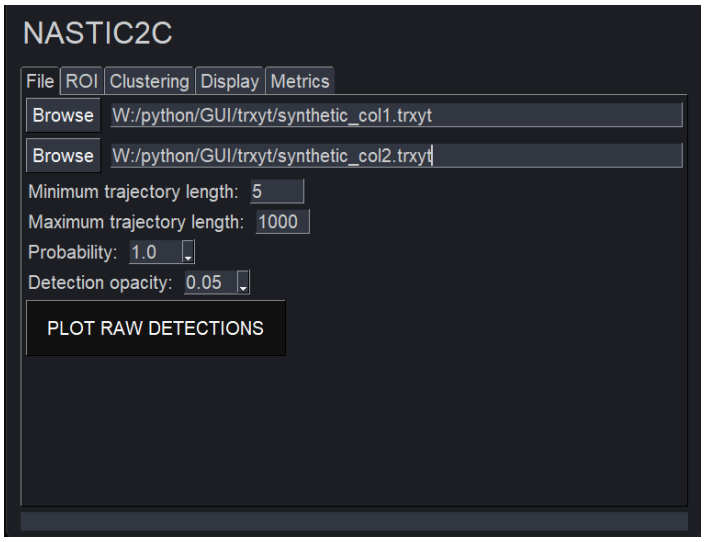
Appendix 3: Two-color NASTIC (NASTIC 2C)

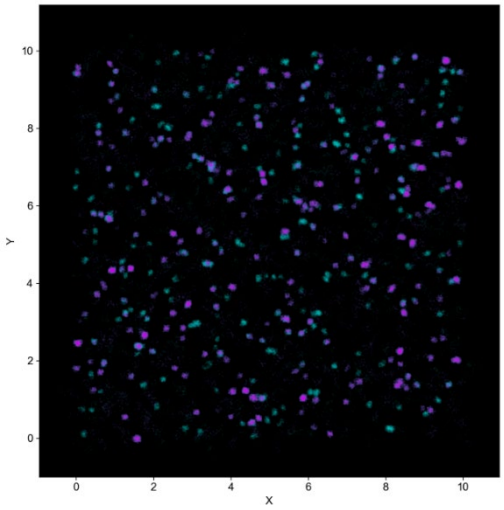
The spatiotemporal indexing approach used in NASTIC can readily be extended to two-color analyses, in which two target molecules are independently labelled with different fluorophores and SMLM data simultaneously acquired at two different wavelengths to generate two datasets. Two-color NASTIC (NASTIC2C) is performed by combining the datasets and establishing spatiotemporal overlap of all trajectories. The type (color) information for each trajectory in the combined datasets is retained, which allows calculation of the relative contribution of each molecular type to the resulting clusters, and is used to inform the graphic output. In addition to identifying clusters of each molecular type, NASTIC2C can identify mixed clusters with varying ratios of molecule 1 and 2, and resolve areas of spatiotemporal overlap. A segment version of two-color NASTIC (segNASTIC2C) is also available, which may be useful for high-density two-color data.

The two-color versions of NASTIC and segNASTIC have similar interfaces to the single-color versions, with allowances for the fact that two datasets are used.

File tab:

Raw detections from the TRXYT files are plotted in two colors (default = cyan and magenta).





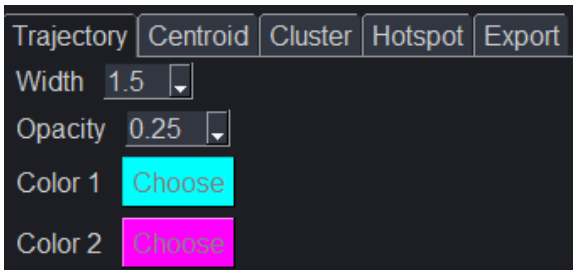
Roi tab:

ROIs are selected using the same mechanism as NASTIC.

"Balance colors" will randomly remove trajectories from the higher density dataset so that both datasets contain the same number of trajectories in the ROI(s).

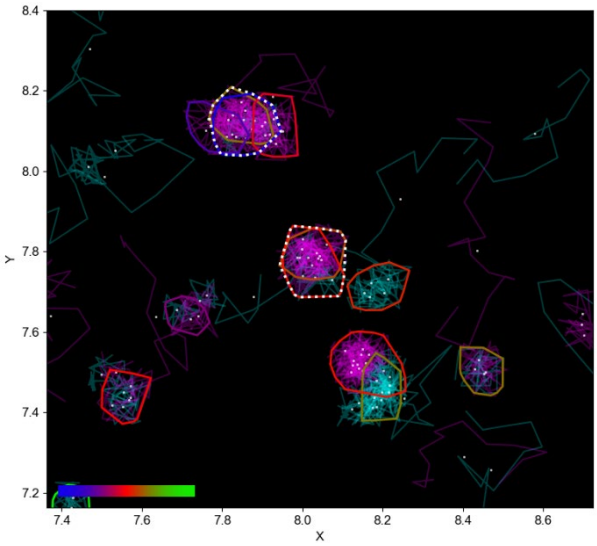
Display tab > Trajectory subtab:

The display tab allows you to select the two colors for the trajectory display. These colours will be used for the raw detection display also.

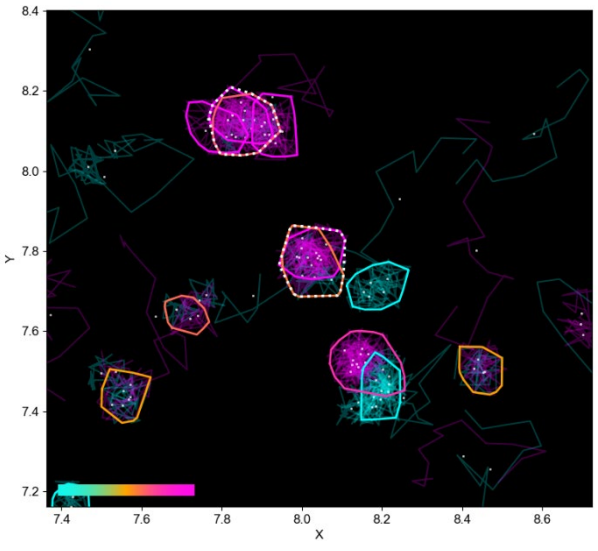


Display tab > Cluster subtab:

Clusters may be displayed according to their average time in the acquisition (as for NASTIC and segNASTIC) by clicking on the "**Color by**" dropdown box and selecting "time":

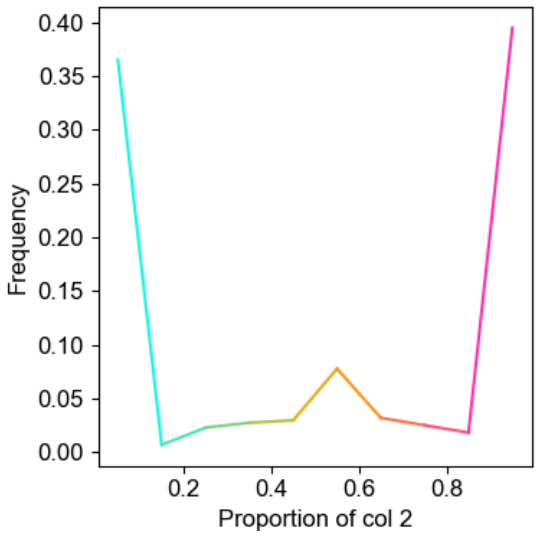


Alternatively, clusters may be displayed according to the relative ratios of molecule 1 and molecule 2 (belonging to TRXYT files 1 and 2 respectively):



Metrics tab:

The metrics returned are largely the same, with the addition of a **"2 color metrics"** button which will plot the distribution of color ratio across all detected clusters.



Appendix 4: BOOSH

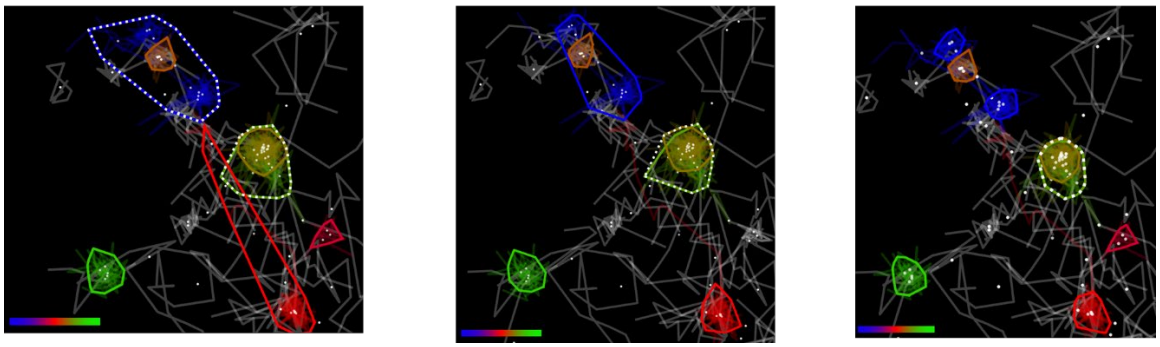
BOOSH is an **experimental** derivative of the NASTIC pipeline which uses the same GUI framework as NASTIC/segNASTIC, but which leverages the DBSCAN algorithm instead of the R-tree.

BOOSH establishes whether individual trajectory detections are in spatiotemporal proximity to detections from other trajectories. This is achieved by converting the temporal dimension of each detection into a spatial dimension. Ordinarily, DBSCAN will spatially cluster points based on a user defined radius (ε). If a given point has a minimum number (MinPts) of other points within ε, then the points are considered clustered. In BOOSH, the user defined time window (*tw*) in which points may be considered temporally clustered, is considered as one ε. Each spatiotemporal point is converted to a 3D spatial point [x, y, t] → [x, y, ε **tw*/t].

Clustering tab:

The BOOSH interface is identical to other NASTIC GUIs, except for the DBCAN specific clustering parameters tab. BOOSH exposes the usual DBSCAN epsilon (ε) and MinPts to the user, along with a time window. By default, a detection (and by extension its parent trajectory) are considered potentially clustered if detections from at least 2 other trajectories lie within 50nm and 10sec. BOOSH outputs the same metrics as the other GUIs, which can also be meta analysed using NASTIC WRANGLER.

NASTIC segNASTIC BOOSH



NASTIC, segNASTIC and BOOSH rely on different algorithmic approaches to largely produce similar output. Using 3D DBSCAN, BOOSH is quicker than the R-tree based NASTICs, and can potentially discern regions of spatiotemporal clustering with higher resolution. BOOSH is however critically reliant on ε correctly reflecting the scale (and density) of the input data, in contrast to NASTIC which is scale independent. BOOSH/DBSCAN's scale dependence can potentially cause issues when analysing multiple datasets in which image acquisition and/or tracking vagaries have resulted in a range of data densities. These issues must be considered before deploying BOOSH, which is clearly marked as **experimental** for a reason.

FAQ

Q: Why did you even bother when there are other clustering programs available?

A: Because our laboratory is particularly interested in the temporal aspects of nanomolecular clustering and these are difficult to extract with current solutions

Q: Why won't NASTIC even start when I double click it?

A: Have you or your IT support installed Python 3.8+, and assigned the correct PATH so the system can find it?

Q: Why does NASTIC crash with missing module errors?

A: Have you correctly installed the required modules using PIP?

Q: I updated NASTIC, why is it now crashing with weird variable errors?

A: The new version probably doesn't like your existing **nastic_gui.defaults** file. Delete it and try again.

Q: Why does NASTIC run slowly on my shiny new Mac?

A: Because Macs and Python don't always play together properly, particularly with GUI Python programs.

Q: Why won't NASTIC run on our server?

A: Because it needs access to a physical computer to do its multicore processing. Use NASTIC_ST instead.

Q: Why does NASTIC occasionally crash after I've analysed many files?

A: Because I'm a second rate programmer. There, I came out and said it. If you're a real programmer and want to help me to optimise NASTIC, please do.

Q: What's with the stupid open source licence?

A: Open source licences actually empower both the author and the user, unlike most commercial licences where the vendor holds all the cards. You can install NASTIC on as many computers as you want, give copies to your friends, sell it, modify it, collaborate with me and others to improve it, or even release your own forked version of the code with a better name. What you can't do is release a version without acknowledging the author of the original concept and code (me), and you can't release your version under a more restrictive licence. Sounds fair to me.

Q: How do I report bugs to you?

A: Firstly, thanks for wanting to do so. Please let me know:

- Your operating system
- Your Python version
- The NASTIC version
- The nature of the bug
- Steps taken to reproduce the bug (if possible)