

Contents

1	Getting Started	2
1.1	Launching the program:	2
1.2	Loading and viewing your data.....	2
1.3	Setting parameters.....	5
1.4	Cropping points.....	6
1.5	Selecting regions of interest.....	7
2	Analysis.....	8
2.1	Pairwise distance distribution	8
2.2	Cumulative distance distribution	9
2.3	Getis-based Clustering Analysis	11
2.4	Spatial Statistics.....	13
2.5	DBSCAN	14
2.6	Pair Auto-Correlation	16
3	Saving your results.....	17
4	References	18

1 Getting Started

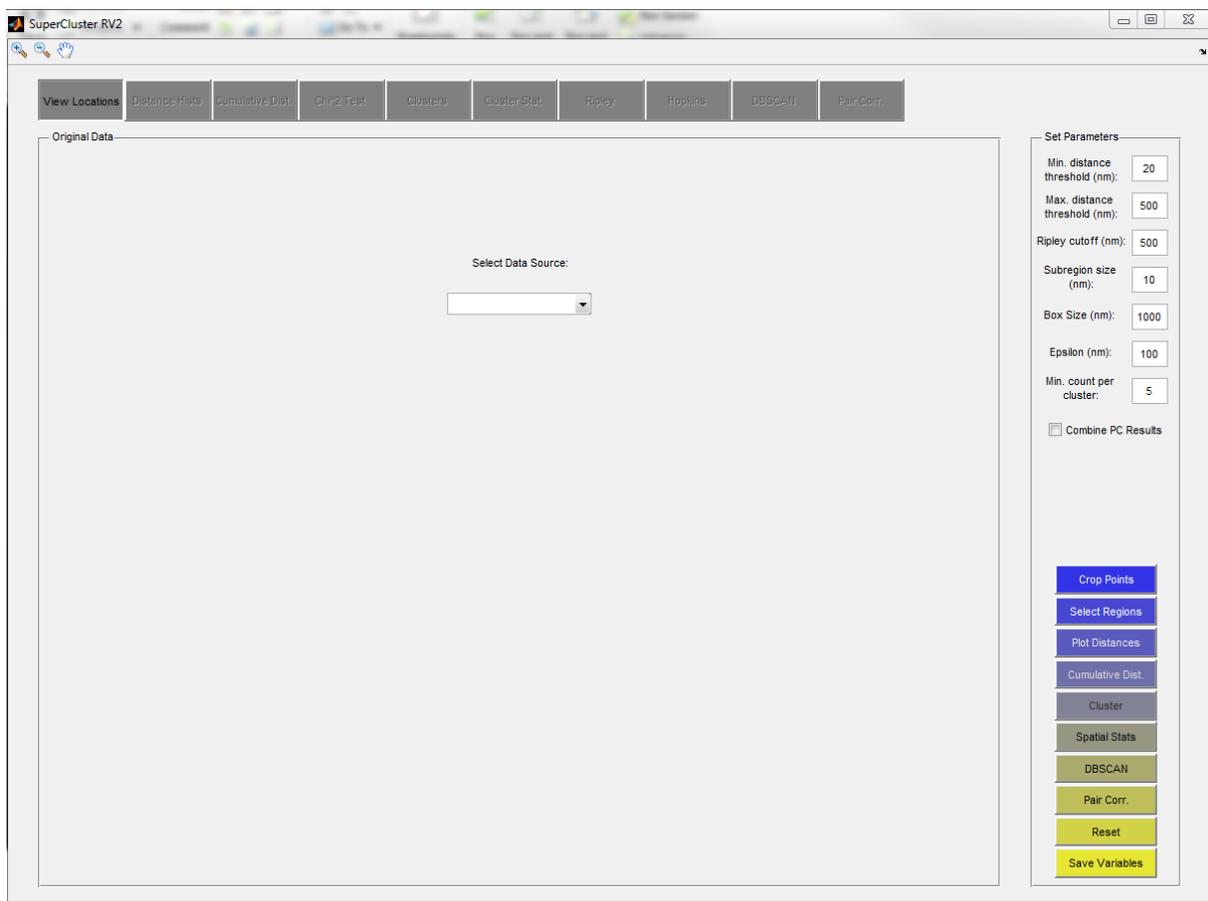
1.1 Launching the program:

Download the file SuperCluster.m into a directory on your MATLAB path. Launch the program either by typing “SuperCluster” at the command line or by opening the file in the MATLAB editor and hitting the run button.

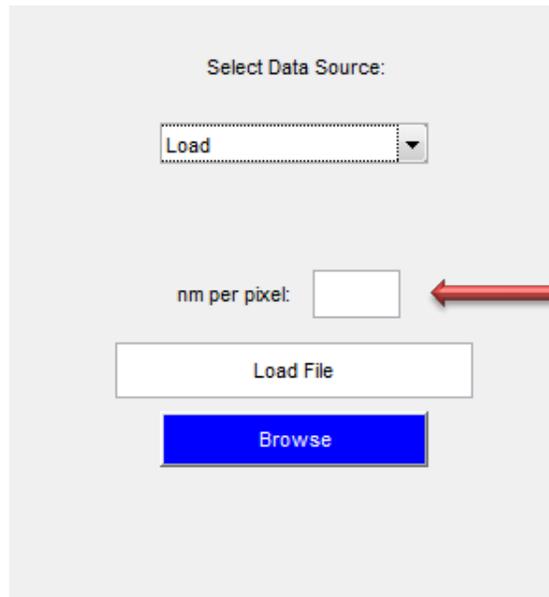
Required MATLAB toolboxes and classes: Image Processing Toolbox, Statistics Toolbox, [DERIVEST Suite](#) (used in autocorrelation curve fitting)

1.2 Loading and viewing your data

Upon launching the program, you will be presented with the main GUI window:



First, select the source of your data – either dynamically generated simulated data or data loaded from file. To load saved data from file, first enter the conversion factor between pixels (or whatever your data units might be) and nanometers. If your data is already in units of nanometers, leave this box empty:



The image shows a software interface titled "Select Data Source:". It contains a dropdown menu with "Load" selected. Below the dropdown is a text input field labeled "nm per pixel:" with a red arrow pointing to it. At the bottom are two buttons: "Load File" and "Browse".

Next, load your data using the “Browse” button. Acceptable file types and formats are:

- .mat files containing SR_demo objects
- .mat files containing x-y coordinates of super resolution data
- .mat files containing previous SuperCluster results files
- .csv files containing x-y coordinates of super resolution data
- .txt files containing comma-separated pairs of super resolution coordinates, with no headers, either in 2 rows or in 2 columns

If you choose to simulate data, enter the desired simulation parameters and hit the “Simulate” button:

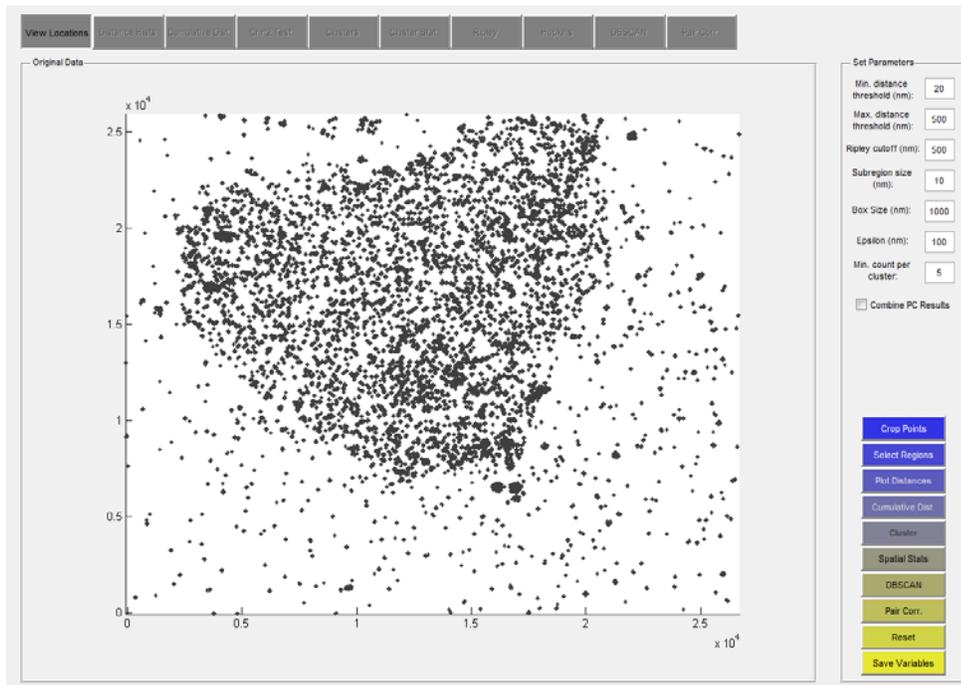
Select Data Source:

Simulate ▼

10	particles per domain
1	domain density (μm^{-2})
.1	2D Gaussian sigma for domain size
10	observations per molecule
.02	localization error (μm)
25	bounding box size (μm)
.01	pixel size in μm

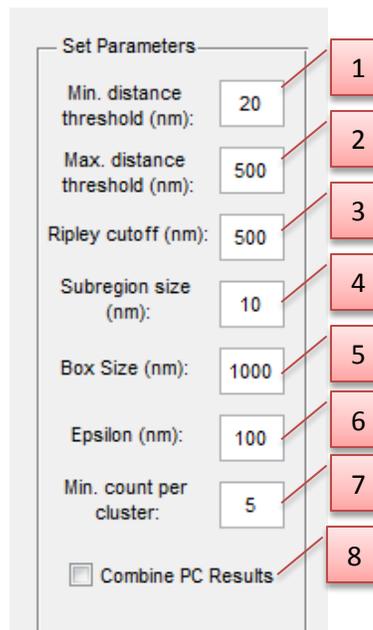
Simulate Data

When you've selected your file or simulated data, the localizations will be plotted in the main GUI figure window:



1.3 Setting parameters

At this point, you will want to set your parameters.



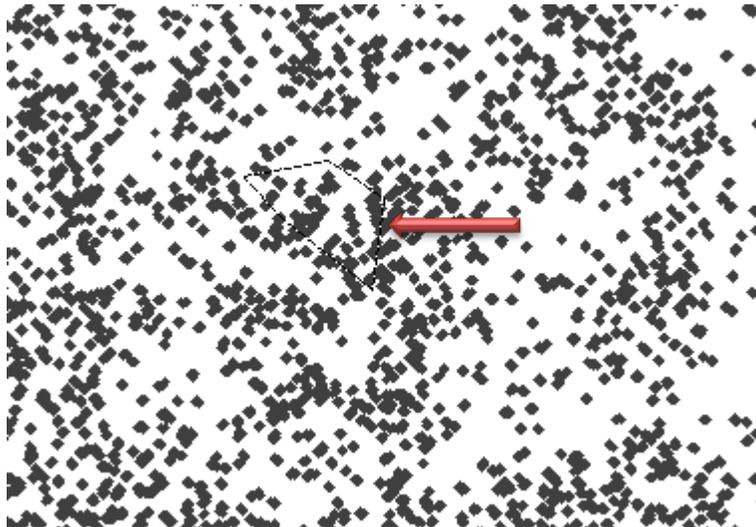
1. **Minimum distance threshold:** This is the minimum point-to-point distance that you wish to consider in your analyses. Point-to-point distances below this threshold will be considered noise and disregarded.

2. **Maximum distance threshold:** This is the maximum point-to-point distance that you wish to consider in your analyses.
3. **Ripley cutoff:** This is the largest radius r to be used in calculating the Ripley's K statistic (and related values).
4. **Subregion size:** For portions of the clustering analysis, including the Getis-based cluster analysis and the pair auto-correlation analysis, the initial ROIs are broken down into a number of equal-sized subregions (you can think of these subregions as bins for super-resolution localizations, or pixels) which are used to create a slightly lower resolution map of the ROI for area-based analyses (this helps to ameliorate the effects of multiple localizations).
5. **Box size:** This is the length of one side of a square ROI. Set this value if you wish to create an ROI of a specific size. If you plan to draw your ROI, you can leave this as the default.
6. **Epsilon:** This is the search radius used in the DBSCAN clustering algorithm.
7. **Min. Count per Cluster:** This is the minimum number of SR localizations to be considered a cluster. Used in both DBSCAN and the Getis-based clustering.
8. **Combine PC Results:** Combine the pair auto-correlation results for all ROIs. This will result in a single curve fit for the data set. Leaving this box unchecked will result in an individual curve fit for each ROI.

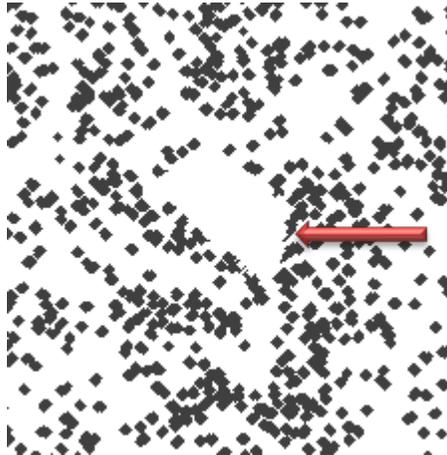
**** NOTE: All units are in nanometers unless otherwise specified ****

1.4 Cropping points

If your data set has an anomaly that you wish to remove to prevent artifacts in your analyses, you can do this using the “Crop Points” button. To remove a region from the data, first click the “Crop Points” button. You will see the cursor change into a plus sign. Next, (single left) click points around the area you wish to remove: (you will see a dotted line indicating the selected region)



When you reach your last point, double click the mouse to close the region and remove the points:



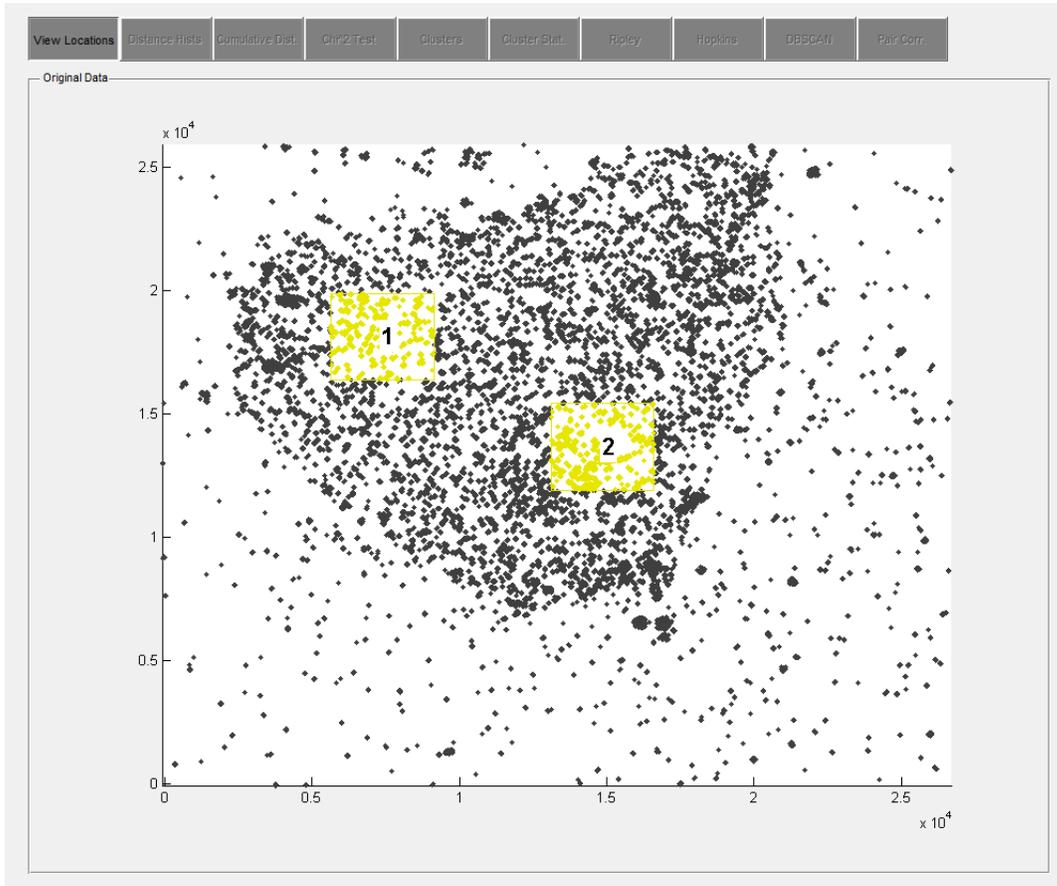
You may repeat this operation as many times as you wish by repeating these steps. Please note that you may not crop points that are already inside an ROI.

1.5 Selecting regions of interest

There are two ways to specify a region of interest (ROI): automatic and manual.

For automatic specification: enter the desired box size (see section 1.3), then click the “Select Regions” button. Left click to place an ROI box in the data viewing pane (the clicked point will be the center point of the ROI). To place additional ROIs, repeat the previous steps.

For manual specification: click the “Select Regions” button. Right click the mouse anywhere in the data viewing pane and then right click and drag to draw an ROI box. The size of this ROI will be retained and you can place additional ROIs of the same size by now following the steps for automatic ROI specification.



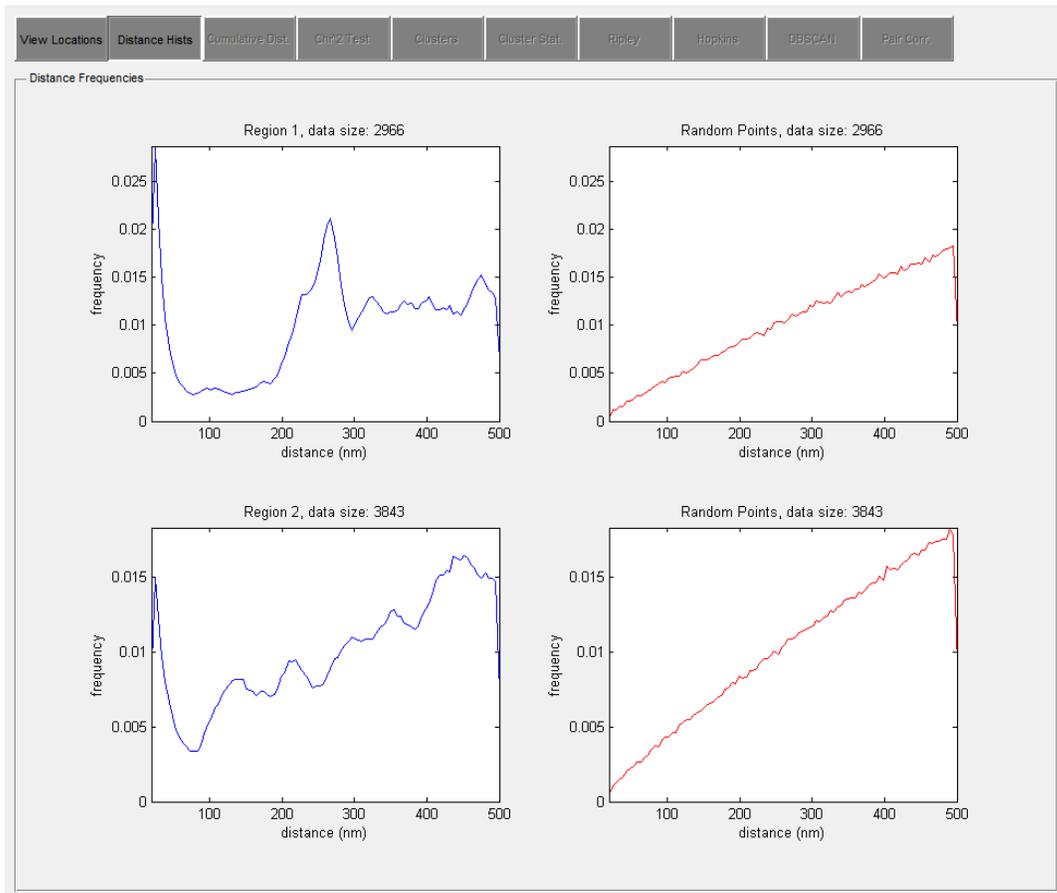
Note: You may use anywhere between 1 and 12 ROIs in a single data set, however using more than 4 ROIs will cause increased processing time, some garbled display text, and difficult to read plots. If you do not care about the display and only want the values, you may use up to 12 ROIs.

2 Analysis

Now that you have selected your regions, it is time to move on to analyzing your data. The following are descriptions of the currently available analyses, with appropriate references provided at the end of this document.

2.1 Pairwise distance distribution

“Plot Distances” calculates the pairwise distance between all localizations in the ROI, except those below or above the minimum and maximum distance thresholds (respectively) and displays the probability density function of the distances. The results are displayed side-by-side with the expected results of an equal number of randomly distributed points (drawn from random uniform distribution) in an ROI of the same size.



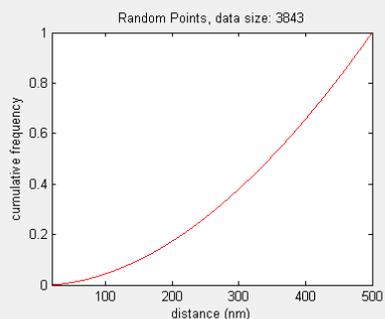
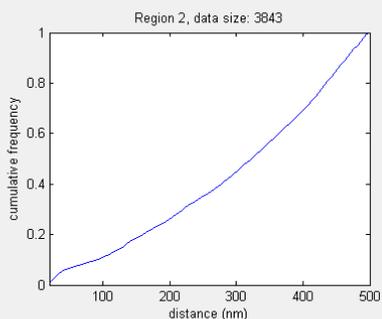
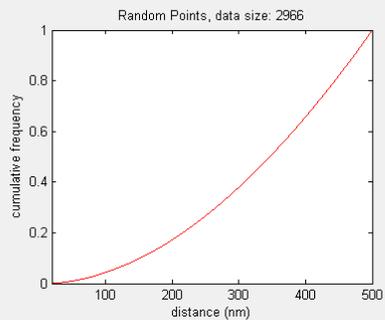
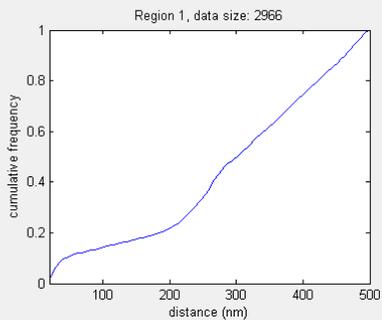
The numerical output (see section 3) of this function is a list of all the pairwise distances (thresholds are NOT applied), a list of the pairwise distances between random points for comparison, the probability histogram in the form of [bins, counts] (with thresholds applied).

2.2 Cumulative distance distribution

The “Cumulative Dist.” function calculates and displays the cumulative density function of the pairwise distances (thresholds applied), as well as the comparison to random data. It also performs a comparison between the data using the 2 sample Chi-Squared test (this only applies if you are interested in differences between ROIs in your data set). If more than 2 ROIs are being analyzed, this test is performed in a pairwise fashion. The output (see section 3) of this function is the cumulative histogram of the pairwise distance in the form of [bins, counts], the empirical CDF, and the results of the Chi-Squared test (names of the regions being compared, a value indicating the acceptance or rejection of the null hypothesis (1 = reject, 0 = accept) and the p-value.

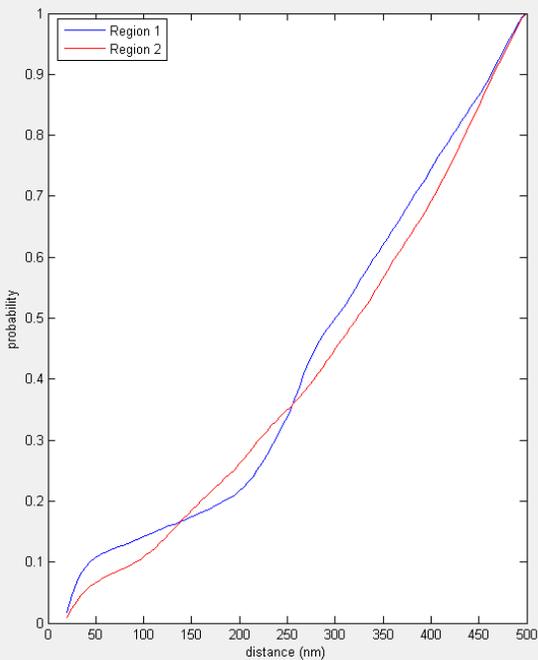
View Locations Distance Hists Cumulative Dist. **Ch² Test** Clusters Cluster Stat. Ripley Hopkins DBSCAN Pair Corr.

Cumulative Frequencies



View Locations Distance Hists Cumulative Dist. **Ch² Test** Clusters Cluster Stat. Ripley Hopkins DBSCAN Pair Corr.

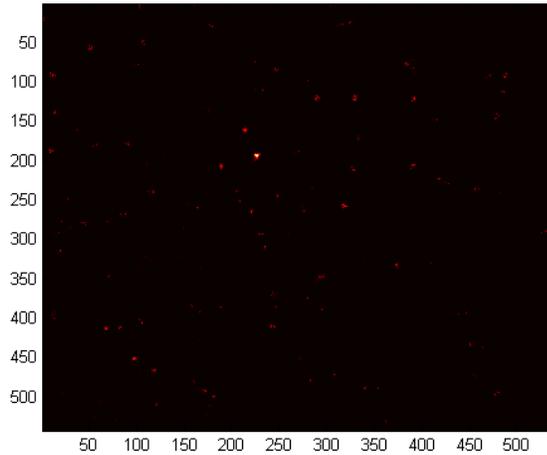
Ch² Test Results



Comparison: Region 1 Region 2
No Significant Difference
p = 0.096546

2.3 Getis-based Clustering Analysis

This portion of the analysis involves breaking the ROI into equal-area subregions (see section 1.3). These subregions are then assigned a value (x_i) based on the number of super-resolution localizations they each contain (a histogram image).



Histogram image of ROI 1 (from above).

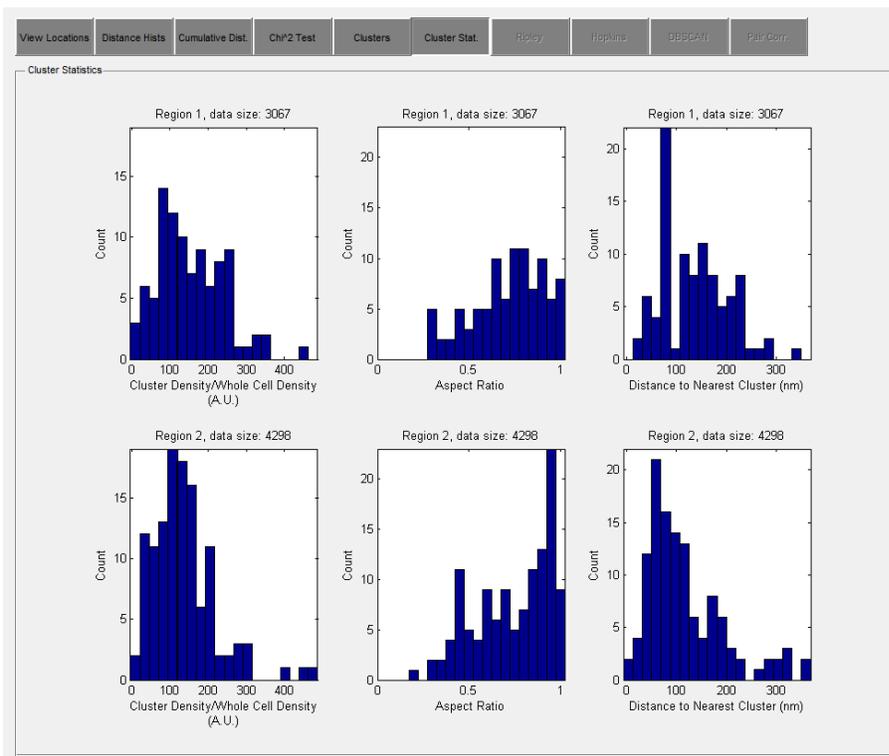
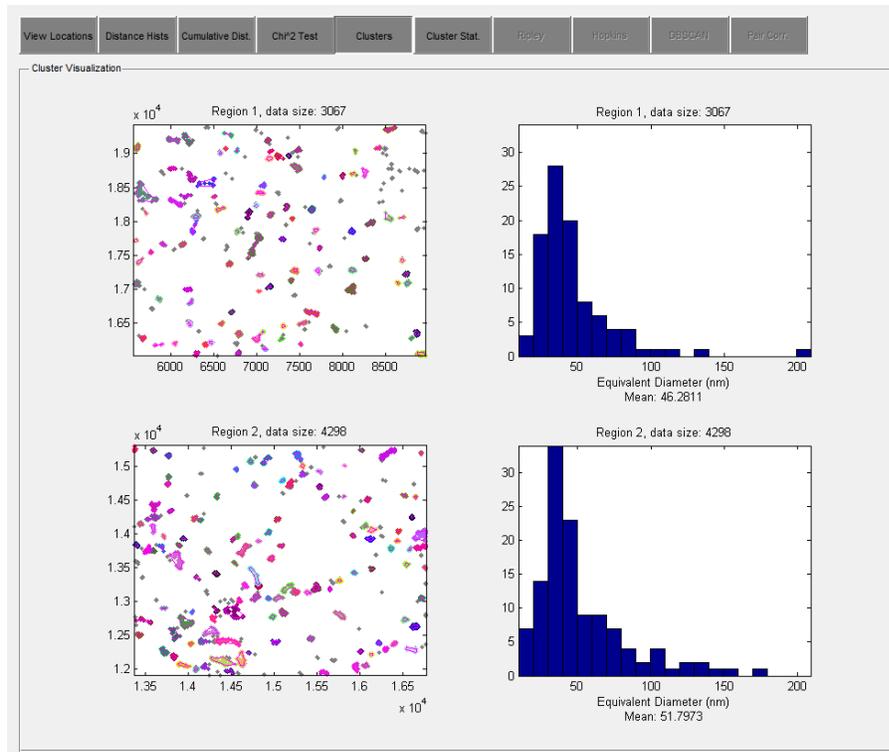
The Getis statistical analysis has been used to analyze geographical clustering data, and is a method that indicates the local amount of clustering between subregions (pixels) of a given size in an image^(1,2). The calculation is carried out based on the following equation:

$$G_i(d) = \frac{\sum_{j=1}^n w_{ij}(d)x_j}{\sum_{j=1}^n x_j}, \text{ for } j \neq i$$

where d is the pairwise distance between subregions i and j ; n is the number of subregions in the ROI; and w_{ij} is a binary weight matrix where $w_{ij}(d) = 1$ if the pairwise distance, d , between subregions i and j is less than the cutoff distance D_c and $w_{ij}(d) = 0$ if d is greater than D_c .

As you can see, this results in a ratio between the sum of the values of subregions within a distance d of the i th subregion and the sum of the values of all subregions, which is a measure of local clustering. By evaluating $\max(G_i(d))$ for $d = [0, D_c)$, we can identify length scales over which the degree of local clustering is increasing (indicating continuous domain structure) and where it is not (indicating discontinuities in domain structure).

This analysis method finds clusters based on local maxima of the Getis G statistic and provides domain visualization, as well as a histogram of the equivalent diameters, the convexity, compactness and distance to nearest neighboring cluster.



The output variables (see section 3) are: a cell array containing the values of $G_i(d)$ for $d = [0, cutoff)$ at each pixel in the ROI, a $1 \times n$ array of cluster aspect ratios, a $1 \times n$ array of cluster equivalent diameters, a $1 \times n$ array of the cluster density/whole cell density, a maximum projection

of pixel-by-pixel G values, a 1 x n array of nearest cluster distances, a cell array containing the alpha-hull of each cluster, a 1 x n array of the area of each cluster (nm²), a cell array containing all of the SR localizations contained in each cluster, and the mean equivalent diameter.

2.4 Spatial Statistics

The spatial statistics module calculates both the Hopkins statistic and Ripley's K function, with methods adapted from previous STMC work³ (<http://stmc.health.unm.edu/tools-and-data/index.html>).

The Hopkins statistic tests for spatial randomness by comparing nearest-neighbor distances from simulated randomly distributed points to their nearest real neighbors, and the nearest-neighbor distances within the set of real points. The Hopkins statistic is given by:

$$H = \frac{U}{U + W}$$

where W is the set of nearest-neighbor distances between real points and U is the set of distances between the simulated random points and their nearest real neighbors. If the real points are randomly distributed, the value of H will be approximately ½, while clustered points will result in an H value close to 1. Points that are uniformly distributed will result in an H that is approaching 0.

Ripley's K function (and related statistics) are based on the detection of deviation from spatial homogeneity. Ripley's K function is given by:

$$K(r) = \frac{1}{n} \sum_{i=1}^n N_{pi}(r) / \lambda$$

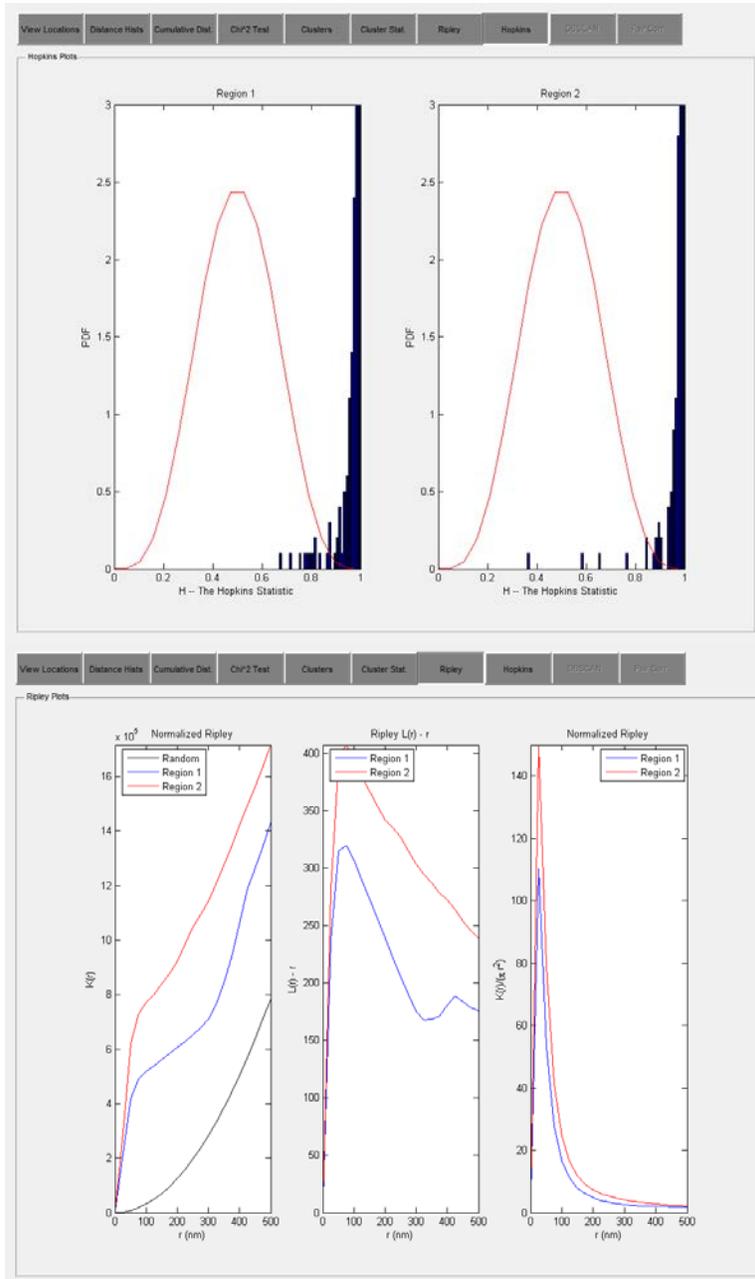
where $N_{pi}(r)$ is the expected number of points within a distance r of the i th point, λ is the point density (approximated by the number of points divided by the area of the region), and n is the total number of points. The expected value of this function is πr^2 . The function can be normalized to give a linear expected value:

$$L(r) = \sqrt{\frac{K(r)}{\pi}}$$

and further normalized so that the expected value is 0:

$$\hat{L}(r) = L(r) - r$$

The Spatial Statistics function calculates and plots H , $K(r)$, $L(r)$, and $\hat{L}(r)$. The output (see section 3) includes an array containing the values of H , and a 1x4 cell array containing the values of $K(r)$, the distance step-size dr , $L(r)$, and $\hat{L}(r)$.



2.5 DBSCAN

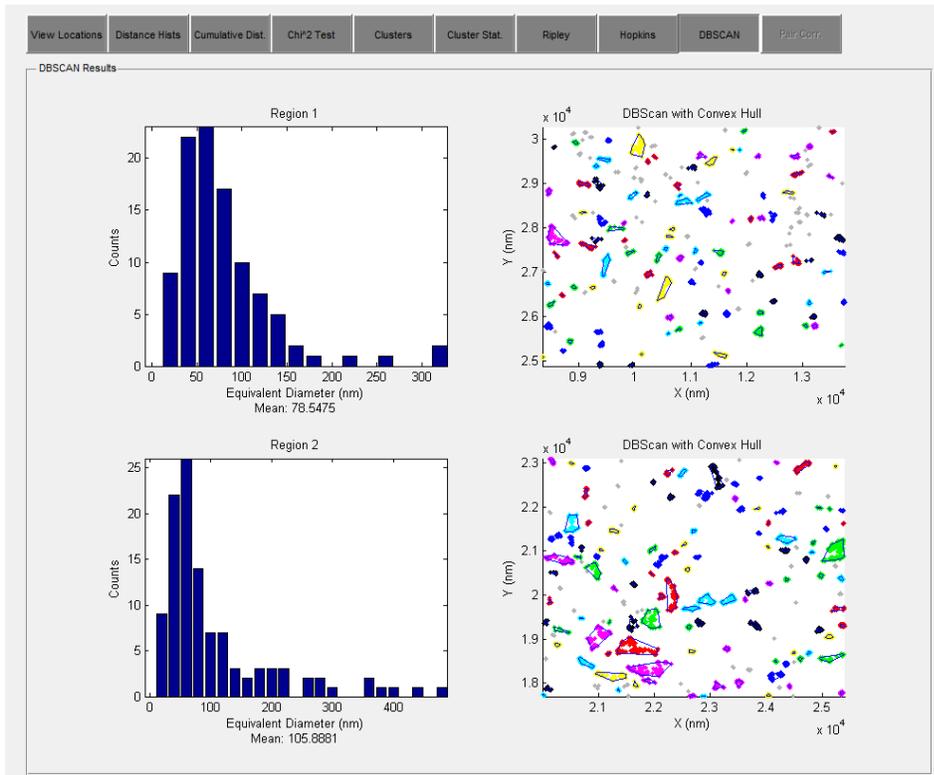
Density-based spatial clustering of applications with noise (DBSCAN) is a data clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu in 1996⁴.

DBSCAN's definition of a cluster is based on the notion of density reachability. A point q is directly density-reachable from a point p if it is not farther away than a given distance ϵ (i.e., is part of its ϵ -neighborhood) and if p is surrounded by sufficiently many points such that one may consider p and q to be part of a cluster. q is called density-reachable (note the distinction from "directly density-reachable") from p if there is a sequence p_1, \dots, p_n of points with $p_1 = p$ and $p_n = q$ where each p_{i+1} is directly density-reachable from p_i .

Note that the relation of density-reachable is not symmetric. q might lie on the edge of a cluster, having insufficiently many neighbors to count as dense itself. This would halt the process of finding a path that stops with the first non-dense point. By contrast, starting the process with p would lead to q (though the process would halt there, q being the first non-dense point). Due to this asymmetry, the notion of density-connected is introduced: two points p and q are density-connected if there is a point o such that both p and q are density-reachable from o . Density-connectedness is symmetric.

A cluster, which is a subset of the points of the database, satisfies two properties:

1. All points within the cluster are mutually density-connected.
2. If a point is density-connected to any point of the cluster, it is part of the cluster as well.

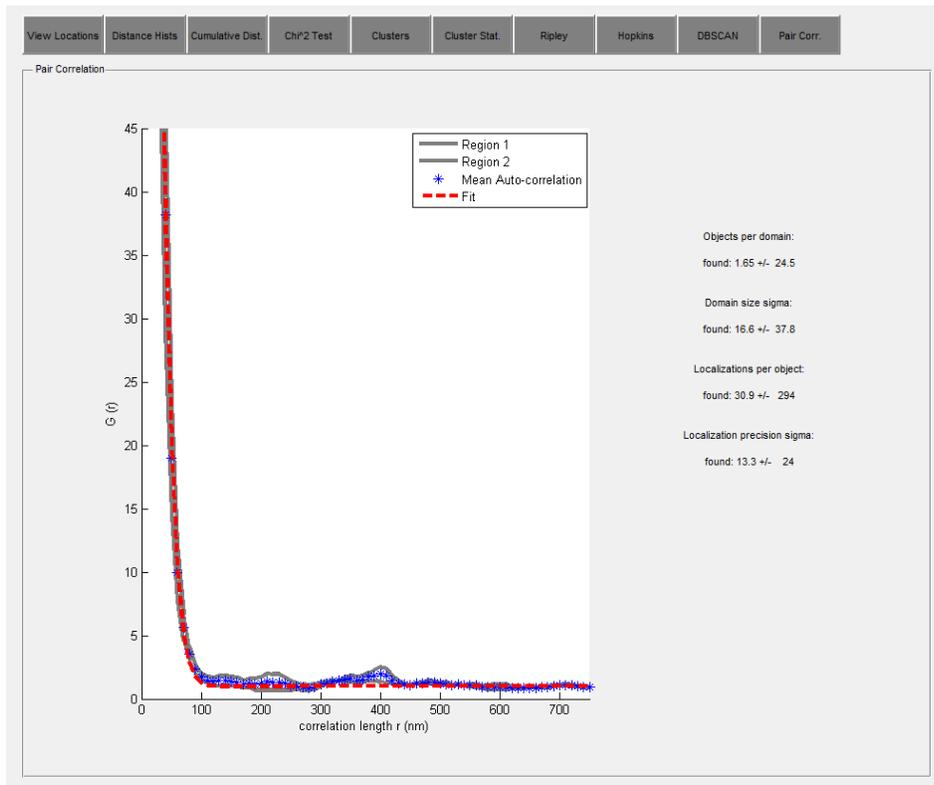


Output variables (see section 3): a cell array containing the number of SR localizations per DBSCAN cluster, a cell array containing the convex hull of each DBSCAN cluster, a $1 \times n$ array of cluster equivalent diameters, the mean cluster equivalent diameter, and a cell array containing the area of each DBSCAN cluster (based on convex hull).

2.6 Pair Auto-Correlation

This analysis is a method developed by Sarah Veatch and colleagues⁵. Pair correlation functions quantify organization in heterogeneous systems and are easily applied to super-resolution localization data. The pair auto-correlation function, $g(r)$, that reports the increased probability of finding a second localized signal a distance r away from a given localized signal, is efficiently calculated using Fast Fourier Transforms, and can account for complex boundary shapes without additional assumptions. The pair autocorrelation code was written by Sarah Veatch and the complimentary fitting code was written by Keith Lidke.

The pair auto-correlation analysis can be performed on each ROI individually, or on the combined ROIs (shown below).



Output variables (see section 3): a 1x4 cell array containing: the ROI image; a matrix with the radius values in column 1, the angularly averaged autocorrelation function in column 2, and the errors of the angularly averaged autocorrelation function in column 3; the average density ρ ; and the 2D autocorrelation function; a matrix containing the estimated fitting parameters (objects per domain, sigma for 2D Gaussian domain size, observations per object, sigma for Gaussian localization precision) with one row for each ROI or one row for the combined ROIs; and the fit model generated from these parameters.

3 Saving your results

The plots and output variables generated by this program are not saved automatically. You must use the “Save Variables” button to do this.

Upon clicking the “Save Variables” button, you will be prompted to specify the save location and name of the results file. The results file is a .mat file containing two variables: `points`, which is the x-y coordinates of the original data, and `regs`, which is a 1xn struct with the following fields:

- **points** - the x-y coordinates of the points in that ROI
- **ROI** - rectangular ROI that you selected
- **area** - area of the ROI in nm²
- **dist**s - pairwise distances between all points in the ROI
- **label** - (internal program use, ignore)
- **name** - (internal program use, ignore)
- **randoms** - a set of randomly generated points that would fit in the ROI, used for comparison between the real data and a similar set of random points (uniform random)
- **randists** - pairwise distance between all of the random points
- **disthist** - histogram of pairwise distances, excluding values above or below the designated cutoffs
- **cumhist** - cumulative histogram of same data as in disthist
- **cdf** - empirical cdf of pairwise distances
- **chidats** - result of chi² test between cdfs (this is only applicable if you care about a statistical comparison between regions on the same cell)
- **G** - a cell array containing the values of $G_i(d)$ for $d=[0, \text{cutoff})$ at each pixel in the ROI
- **maxG** - the maximum projection of pixel-by-pixel G values
- **clusts** - cell array containing all of the SR localizations contained in each cluster
- **aHull** - cell array containing the alpha-hull of each cluster, both inner and outer
- **Clarea** - a 2 x n array of the area of each cluster (nm²), based on alpha-hull, both inner and outer
- **compact** - a 2 x n array of cluster compactness measurements, both inner and outer
- **density** - a 2 x n array of the cluster density/whole cell density values, both inner and outer
- **nearestdist** - a 2 x n array of nearest cluster distances, both inner and outer
- **eqDiameter** - a 2 x n array of cluster equivalent diameters, both inner and outer
- **estMnDiam** - 1 x 2 array, the mean equivalent diameter, both inner and outer
- **hop** - the results of the Hopkins test
- **rip** - a 1x4 cell array containing the results of the Ripley test, in this order: $K(r)$, the distance step-size dr , $L(r)$, and $\hat{L}(r)$.
- **DBcounts** - cell array containing the number of SR localizations per DBSCAN cluster
- **DBhull** - cell array containing the convex hull of each DBSCAN cluster
- **DBdiam** - a 1 x n array of cluster equivalent diameters
- **DBmnDiam** - the mean cluster equivalent diameter
- **DBarea** - cell array containing the area of each DBSCAN cluster (based on convex hull)

- **paircorr** - a 1x4 cell array containing: the ROI image; a matrix with the radius values in column 1, the angularly averaged autocorrelation function in column 2, and the errors of the angularly averaged autocorrelation function in column 3; the average density rho; and the 2D autocorrelation function
- **fitresults** - a matrix containing the estimated fitting parameters (objects per domain, sigma for 2D Gaussian domain size, observations per object, sigma for Gaussian localization precision) with one row for each ROI or one row for the combined ROIs
- **model** - the fit model generated from these parameters
- **params** - the list of parameters used in generating these results.

Additionally, all plots will be automatically saved to the same directory. The plots are saved as .fig files, which facilitates format changes and allows the plots to be exported in a number of other formats by the user. The exception to this is the Hopkins plot and DBSCAN plot which, due to an as-yet unsolved bug, must be saved as a .png file (sorry for the inconvenience).

4 References

1. Getis, A., and J. K. Ord. 1992. The analysis of spatial association by use of distance statistics. *Geographical Analysis* 24:189-206.
2. Getis, A., and J. Aldstadt. 2004. Constructing the spatial weights matrix using a local statistic. *Geographical Analysis* 36:90-104.
3. Zhang, J. , Leiderman, K., Pfeiffer, J., Wilson, B., Oliver, J., and Steinberg, S. 2006. Characterizing the Topography and Interactions of Membrane Receptors and Signaling Molecules from Spatial Patterns Obtained using Nanometer-scale Electron-dense Probes and Electron Microscopy, *Micron*, 37: 14-34.
4. Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." *KDD*. Vol. 96. 1996.
5. Veatch, Sarah L., et al. "Correlation functions quantify super-resolution images and estimate apparent clustering due to over-counting." *PloS one* 7.2 (2012): e31457.