HDBSCAN Cluster Analysis

Standard Operating Protocol v0.5.0

Philip Ciunkiewicz

GitHub | Google Scholar | LinkedIn | YouTube

Contents

List of Figures

1 Introduction

This document will provide a standard operating protocol for performing cluster analysis on single molecule localization (SMLM) data. The goal is to prepare you, the researcher, to be able to identify clustering behavior within your own data and apply computational tools to fully explore your images. This exploration includes optimizing parameters within the clustering tool to fit your data, evaluating the quality of the final clusters, and lastly interpreting the results.

1.1 Background

1.1.1 What is Cluster Analysis?

Simply put, clustering is a data analysis technique where objects are grouped based on similarities between their features. The fundamental idea is that all data points identified as part of a group (cluster) should be more similar to each other than to the members of other groups [5]. For SMLM, we are mainly interested in spatial clustering. Physical location (X, Y, Z) is the corresponding set of features in spatial clustering, with the distance between objects providing us with information regarding how "similar" they are. Tightly grouped molecules are more likely to correspond to a single structure, and thus they are more likely to be identified as part of the same cluster. Conversely, sparse or uniform distributions of molecules are likely to be identified as a single large cluster or even noise. This form of classification reflects the conventional idea of what a cluster is, allowing for some robust and intuitive insights to be extracted from data.

1.1.2 Why is Cluster Analysis Useful?

Clustering can be a powerful investigative tool for datasets where the "correct" results are unknown, commonly referred to as "unsupervised" clustering. Supervision in clustering corresponds to the ability to supervise (teach) the clustering algorithm to make better decisions based on a priori results. In the context of biological data (specifically SMLM images), results are not known a priori, however visual inspection of the data can provide an idea of which points should belong to clusters. Performing cluster analysis on the dataset offers robust quantitative information to support this visual intuition, as well as offering a number of additional avenues of investigation [6,16]:

- Data reduction
  - partitioning the dataset into clusters allows for interesting regions to be identified and studied in more detail than would be possible for the entire dataset
- Hypothesis generation and testing
  - without a priori knowledge, clustering can provide avenues for generating hypotheses for your investigation as well as methods for testing those hypotheses
- Prediction of new samples
  - from the results of initial clustering, the trained algorithm can provide predictions for new samples and classify those samples into either preexisting or new clusters

1.2 Cluster Analysis in Practice

In practice, there are a few steps required for performing cluster analysis. Figure 1 illustrates the steps required for a generic clustering problem, however some of these steps are already taken care of in the context of SMLM. The features of interest in SMLM clustering are the spatial positions of the localizations, so feature selection is completed by default. Intensity is an additional feature to investigate, however it is more suited for being used as a sample "weight" for each localization rather than a distinct feature.

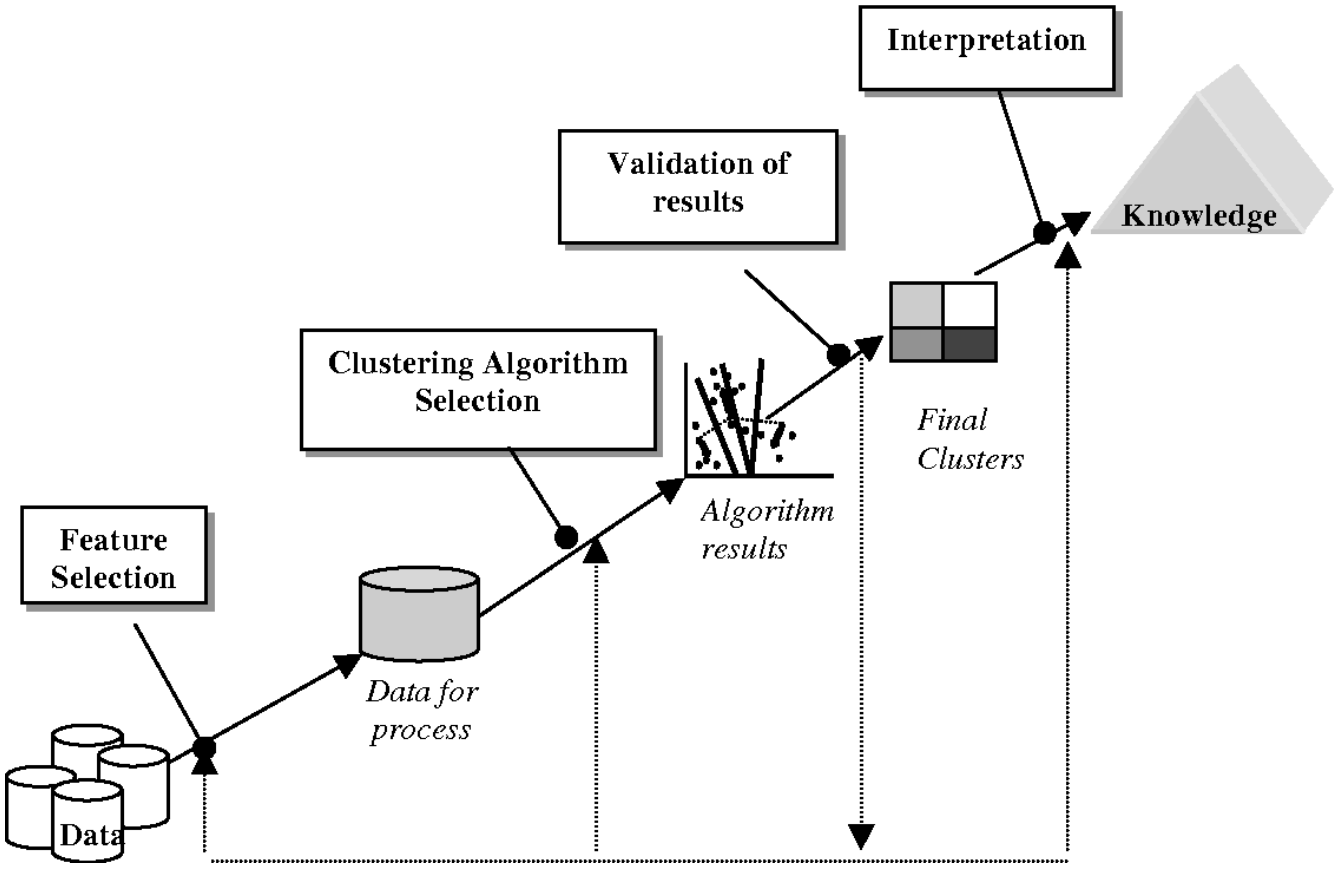


Figure 1: Steps of the clustering process[6].

As with most scientific tools, clustering algorithms are not a one-size-fits-all solution for investigating your dataset. Every clustering algorithm carries with it a set of parameters, formally referred to as hyper parameters, which need to be tuned to provide optimal results for your specific application. In order to optimize parameters and interpret the results of clustering, it is important to understand how clusters are identified. The specific algorithm being used is called HDBSCAN: Hierarchical Density-Based Spatial Clustering for Applications with Noise. This is a modified version of a very popular clustering algorithm DBSCAN, providing certain advantages for non-homogeneous cluster densities.

1.2.1 DBSCAN

DBSCAN works by examining the density of points within the data, assigning cluster labels to points based on two parameters: proximity and minimum number of neighbors[3]. The proximity is represented by a radius r. Points within this radius are considered neighbors, and points outside of this radius are excluded. This proximity check is done for all points in the data and the number of neighbors are counted (left side of Fig. 2). The next check is for the minimum number of neighbors, n (right side of Fig. 2). If a point has at least n neighbors within a radius r, it is considered a clustered molecule (green). If a point does not have n neighbors but is within the radius of a clustered molecule, it is considered as a boundary molecule (yellow). If a point is neither a clustered nor a boundary molecule, it is counted as an outlier (blue).

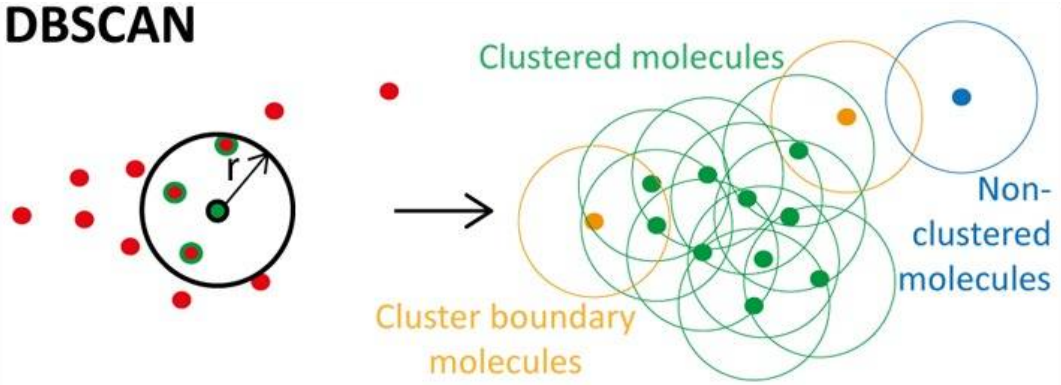
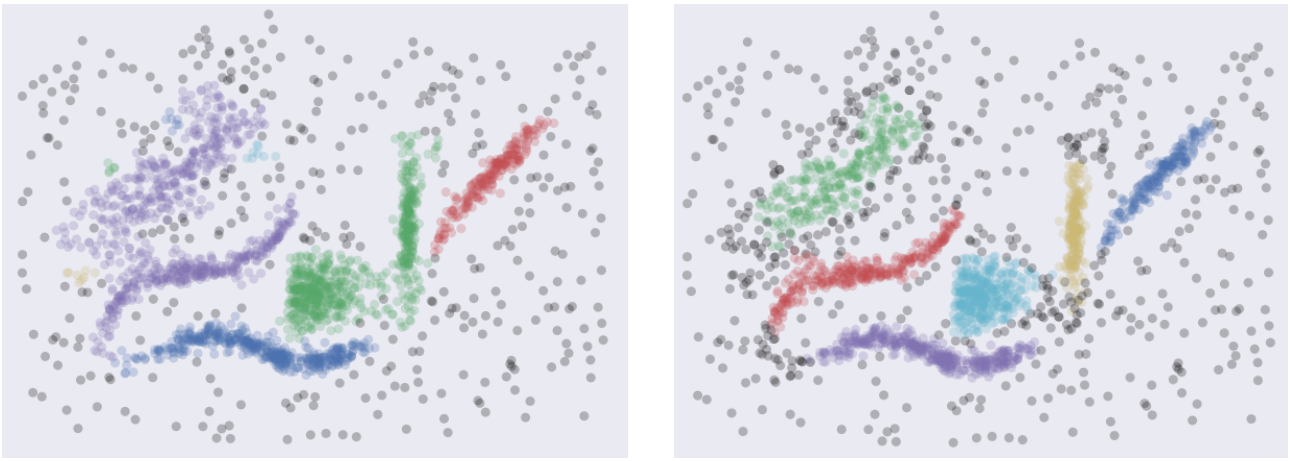


Figure 2: DBSCAN algorithm radius and minimum neighbors density-based clustering [11].

This density-based clustering approach is great if the structures that you are interested in all have similar densities, however the restriction imposed by setting an explicit radius makes it difficult to identify non-homogeneous cluster densities.

1.2.2 HDBSCAN

HDBSCAN improves upon DBSCAN by replacing the strict radius with a variable parameter epsilon. Epsilon serves the same purpose as the radius, however, it is never explicitly set by the user. Instead, the algorithm determines the best epsilon for each given point based on how stable the final cluster is against small changes in epsilon [10]. A detailed technical description of the algorithm is provided in McInnes et al. [9]. We can illustrate the difference that this approach makes by comparing the results of both algorithms for a challenging synthetic dataset (Fig. 3).



(a) DBSCAN (b) HDBSCAN

Figure 3: Comparison of DBSCAN and HDBSCAN clustering results for a synthetic dataset [8].

The dataset used contains both globular and non-globular clusters, as well as a significant amount of noise. We can see that the DBSCAN algorithm struggles with differentiating points within the green and purple clusters, as well as wrongly identifying multiple mini-clusters around the large purple boundary. Even to the untrained eye, the results of HDBSCAN clustering in Fig. 3(b) are much more compelling than those of DBSCAN in Fig. 3(a). However, how can we quantitatively assess the quality of clustering to definitively say that one is better than the other?

1.2.3 Validating Clusters

As mentioned earlier, performing cluster analysis without knowing the correct results (ground truth) ahead of time is referred to as unsupervised clustering. Not knowing what the objective truth is makes it challenging to assess the validity of results, however there do exist a few quantitative measures for cluster analysis:

- □ Silhouette Score [14]
- □ Calinski-Harabaz Index [7]
- □ Davies-Bouldin Index [2,6]

These will individually be elaborated upon in subsequent sections of this document. Along with a quantitative score of how well the algorithm clustered the data, a qualitative assessment by an expert in the respective field is also important. Performing a qualitative assessment is especially important when the quantitative measures score surprisingly high or low, as there are known limitations to each measure. Below (Fig. 4) is an example for when qualitative assessment is as important or even more important than any quantitative measure. Even to the untrained eye, many of the clustering outcomes shown in Fig. 4 can be immediately discarded as objectively poor. Identifying poor results is straightforward for toy-datasets such as the ones below, however with noisy and messy real data the task can become very challenging.

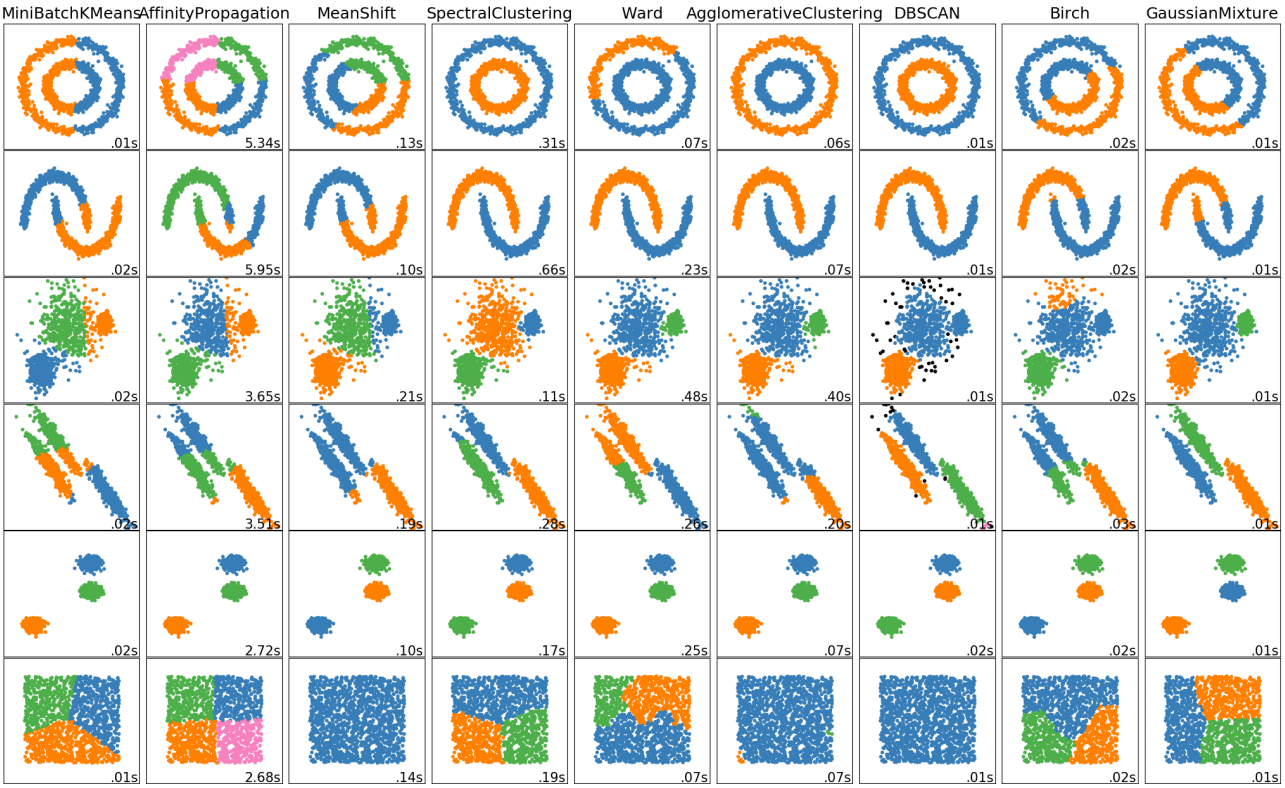


Figure 4: Clustering results for a variety of clustering methods [12].

1.3 Prerequisites

In order to perform any sort of cluster analysis on your data, there are a few preliminary steps which you must take. This section explores the checks you should perform before starting.

1.3.1 SMLM Data

It is useful to start with the data you wish to perform clustering on already collected and processed. The focus of this SOP is SMLM reconstruction data. Follow the procedures outlined by your SMLM reconstruction software package to generate your data. If you do not have any reconstruction software, the following repository provides an excellent starting point: https://srm.epfl.ch/SoftwareList. Note that if you are using software other than ThunderSTORM, you will need to follow further instructions on data preparation in subsequent sections.

It is also recommended to validate the reconstruction using a tool such as NanoJ SQUIRREL or similar. For best results, ensure good sample preparation before collection and aim for the lowest noise possible.

1.3.2 Clustering Software Utility

The included open source software package for SMLM cluster analysis with HDBSCAN is provided in the following GitHub repository: `https://github.com/PCiunkiewicz/smlm-clustering`. Installations instructions are present on the repository readme for ease of use.

2 Getting Started

2.1 Installing the Software

The software is distributed via GitHub and requires Python $\geq$ 3.6 to run. Installation instructions are included as part of the readme in the GitHub repository: `https://github.com/PCiunkiewicz/smlm-clustering`. Future versions of the software may be available as pre-compiled packages or through a package manager such as pip. This protocol as well as the readme will be updated accordingly to reflect any future changes.

2.2 Preparing your Data

The clustering software accepts any SMLM results tables which have been reconstructed using ThunderSTORM in ImageJ. These results tables are in the form of `.csv` files with the following structure:

Table 1: Generic ThunderSTORM results table structure.

id	frame	x [nm]	y [nm]	sigma [nm]	intensity [photon]
1	1	2474.999	6736.088	195.8157	1841.965
2	1	4356.137	14133.32	145.0092	5599.702
3	1	4542.775	9091.265	121.3134	221.4421
4	1	5977.444	7594.973	137.2764	155.5187
5	1	6079.46	11231.17	137.2616	2693.631

Note that you may have extra columns depending on the settings of your reconstruction. If your data was not reconstructed in ThunderSTORM but is in the form of a `.csv` file, you can manually identify which columns contain which information upon loading your dataset. If your data is not in `.csv` format, please convert it to `.csv` using Microsoft Excel or a similar spreadsheet software package.

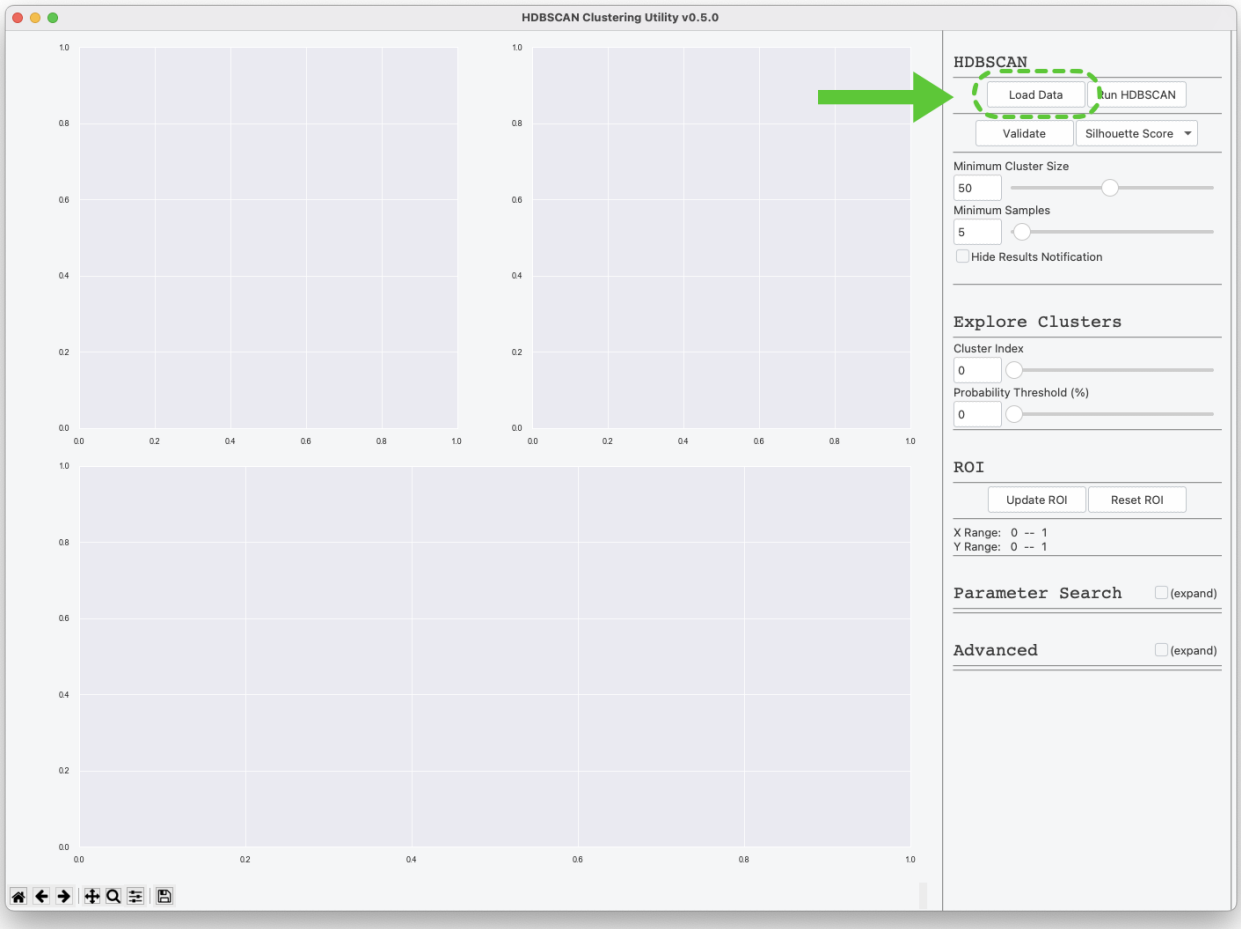
Localization position (x, y) must be included for the clustering algorithm to work, and due to the scale of most super resolution images, it should be provided in units of nm. Other functionality in the software will require more information alongside the localization position (such as intensity or channel number), however these requirements will be explicitly stated in the appropriate sections of this document.

2.3 Quick-Start Guide

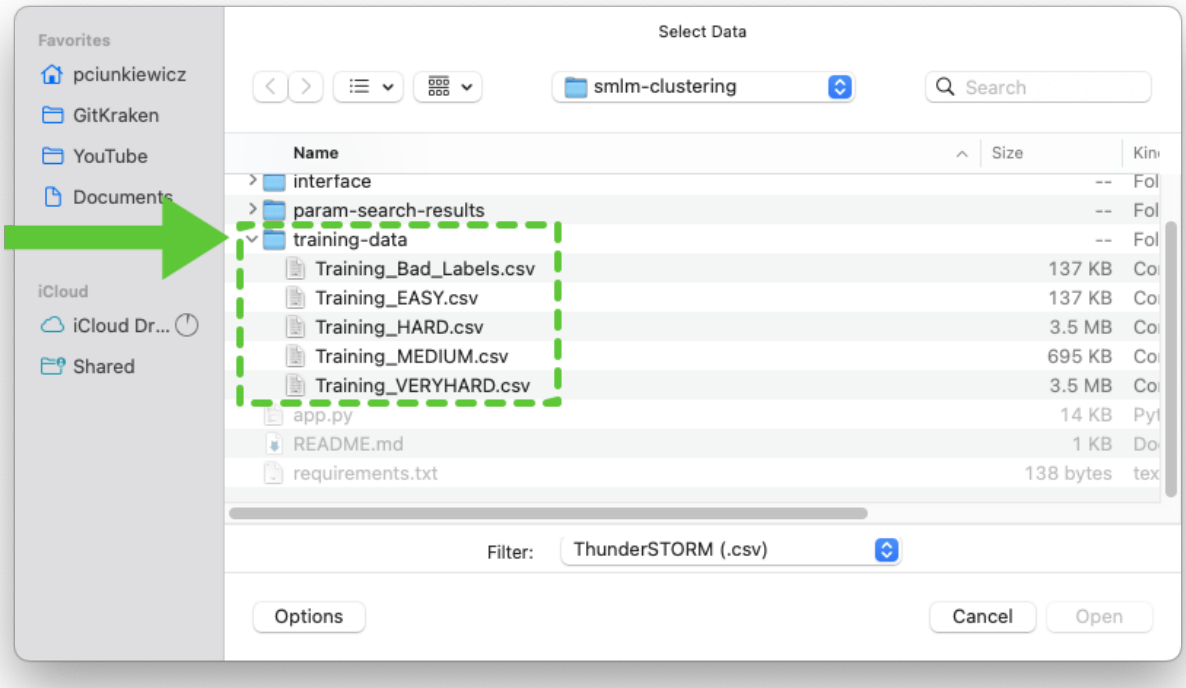
This quick-start guide takes you through the following steps to get up and running within minutes:

(1) Load Data
(2) Perform Clustering
(3) Explore Results

1. Load Data

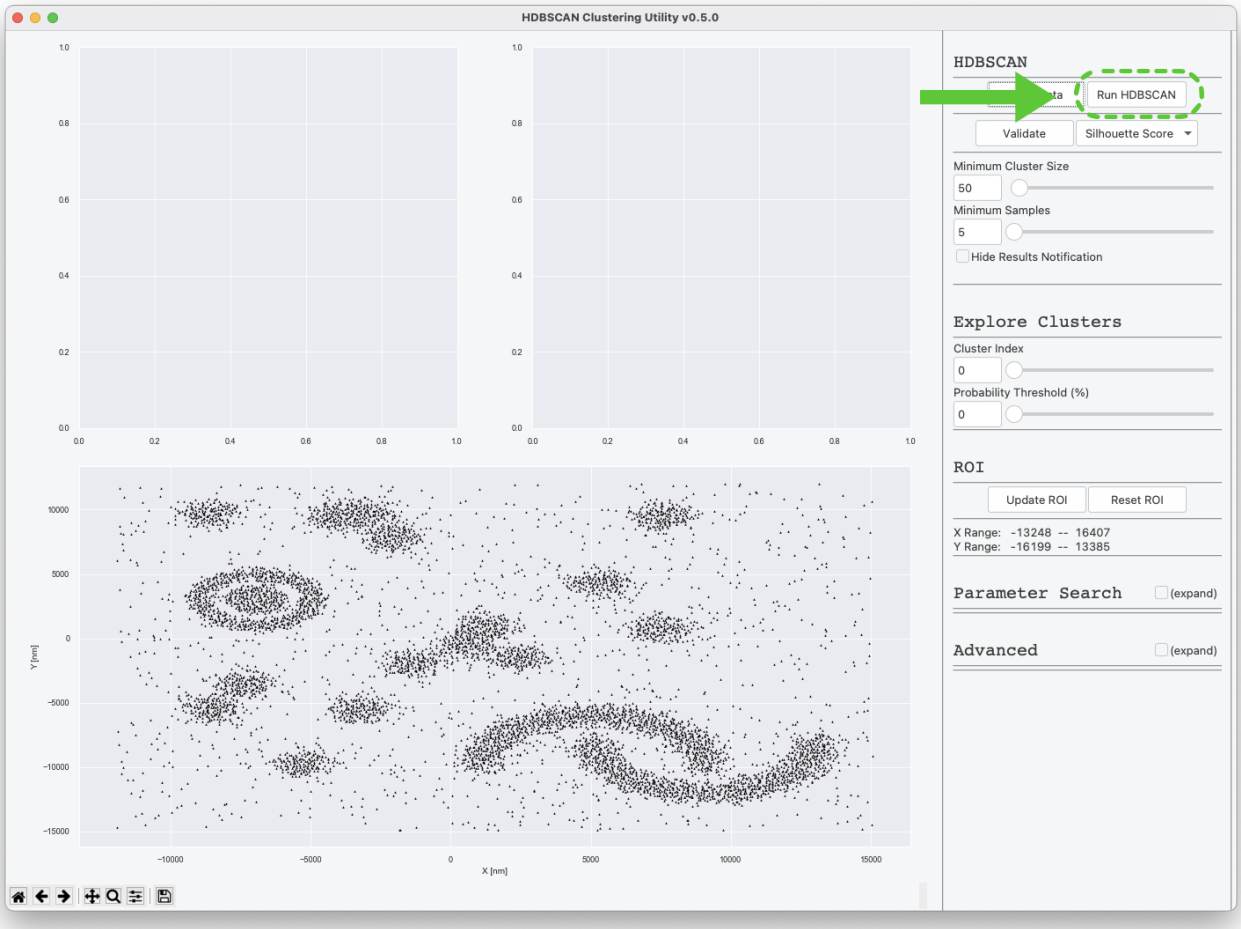


Click the "Load Data" button in the top right corner of the tool.



Find your data or use the provided samples under "training-data".

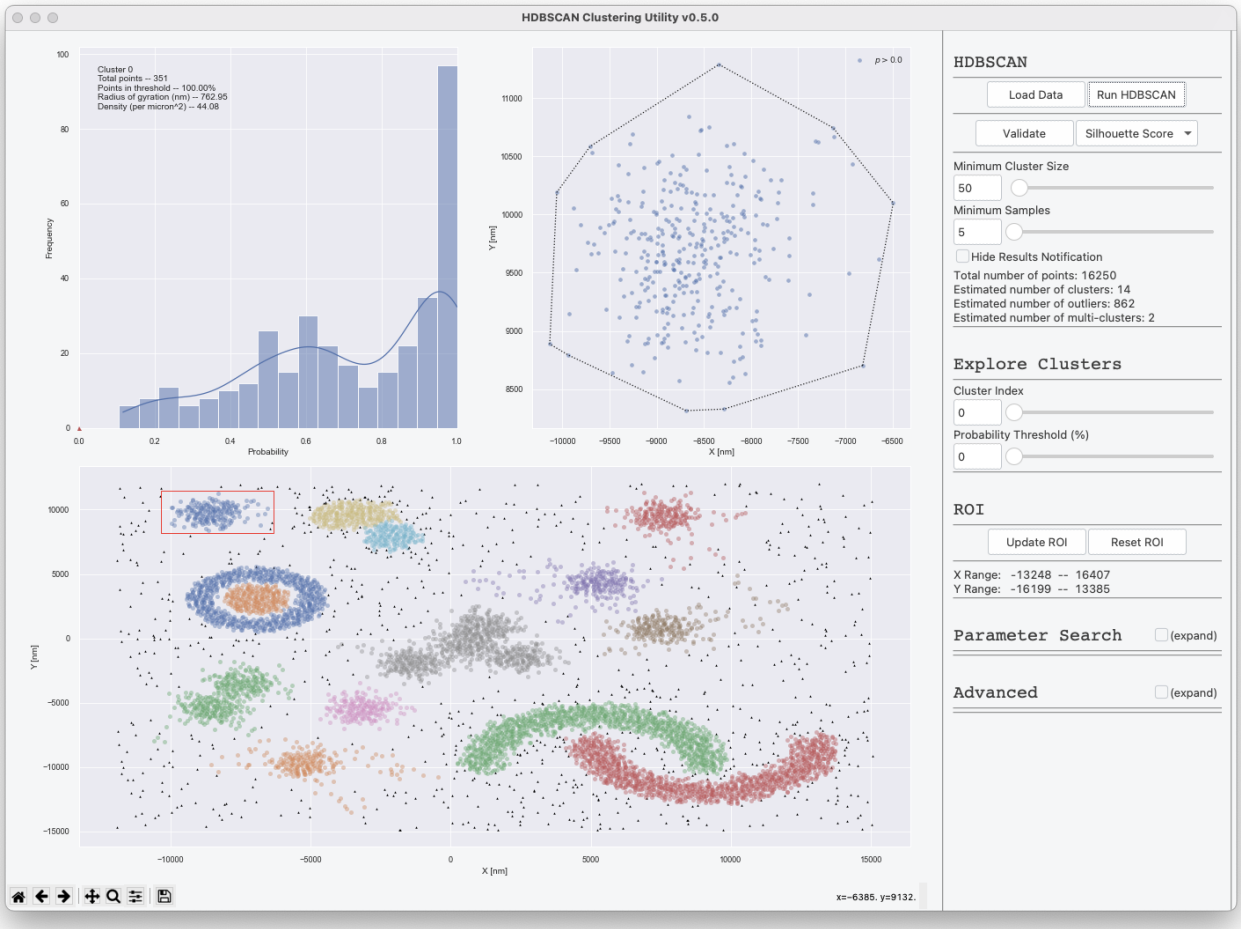
2. Perform Clustering



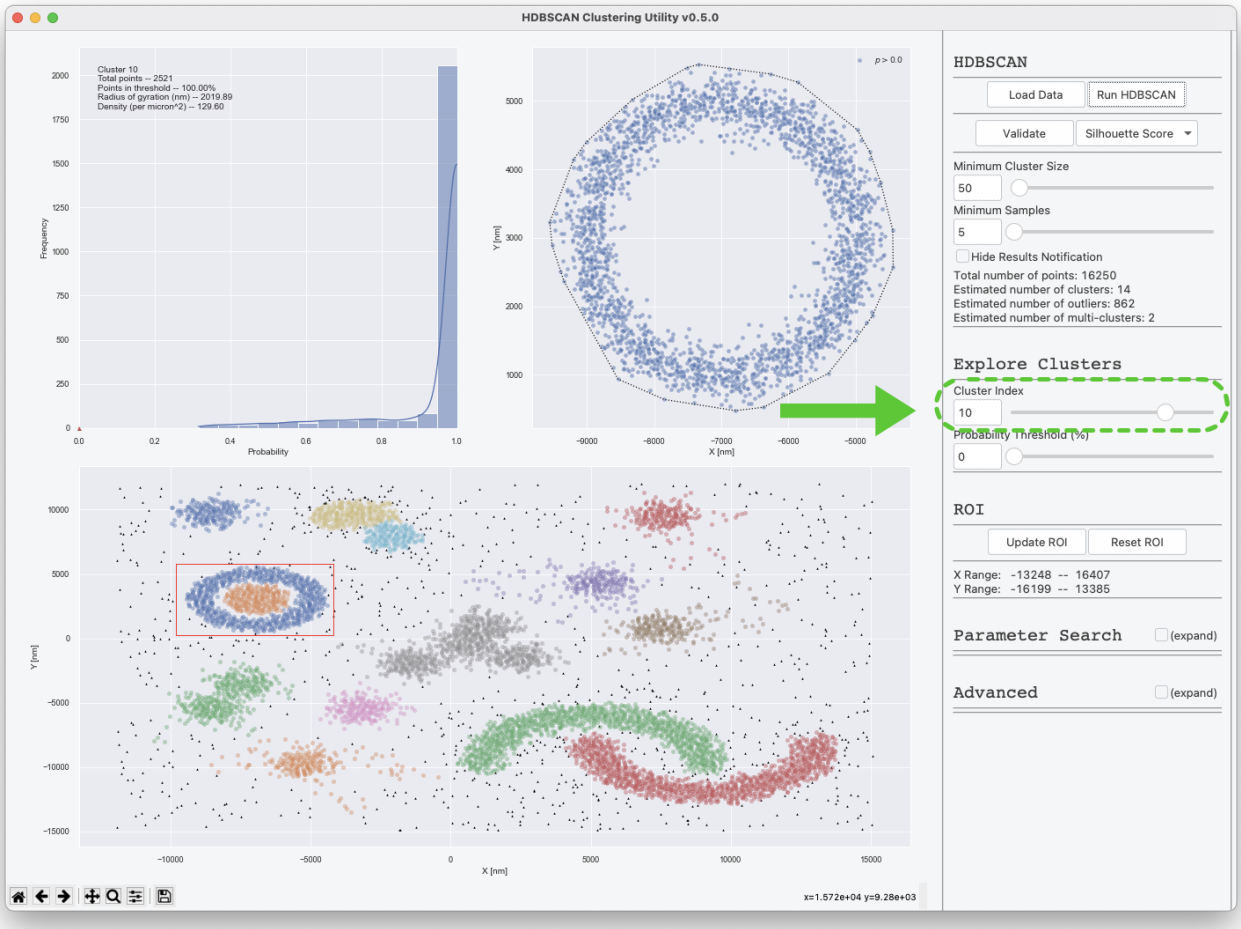
Click the "Run HDBSCAN" button in the top right corner of the tool.

You will get a summary notification detailing your results. This notification can be disabled by using the "Hide Results Notification" checkbox. This data is displayed in the main interface upon closing the notification.

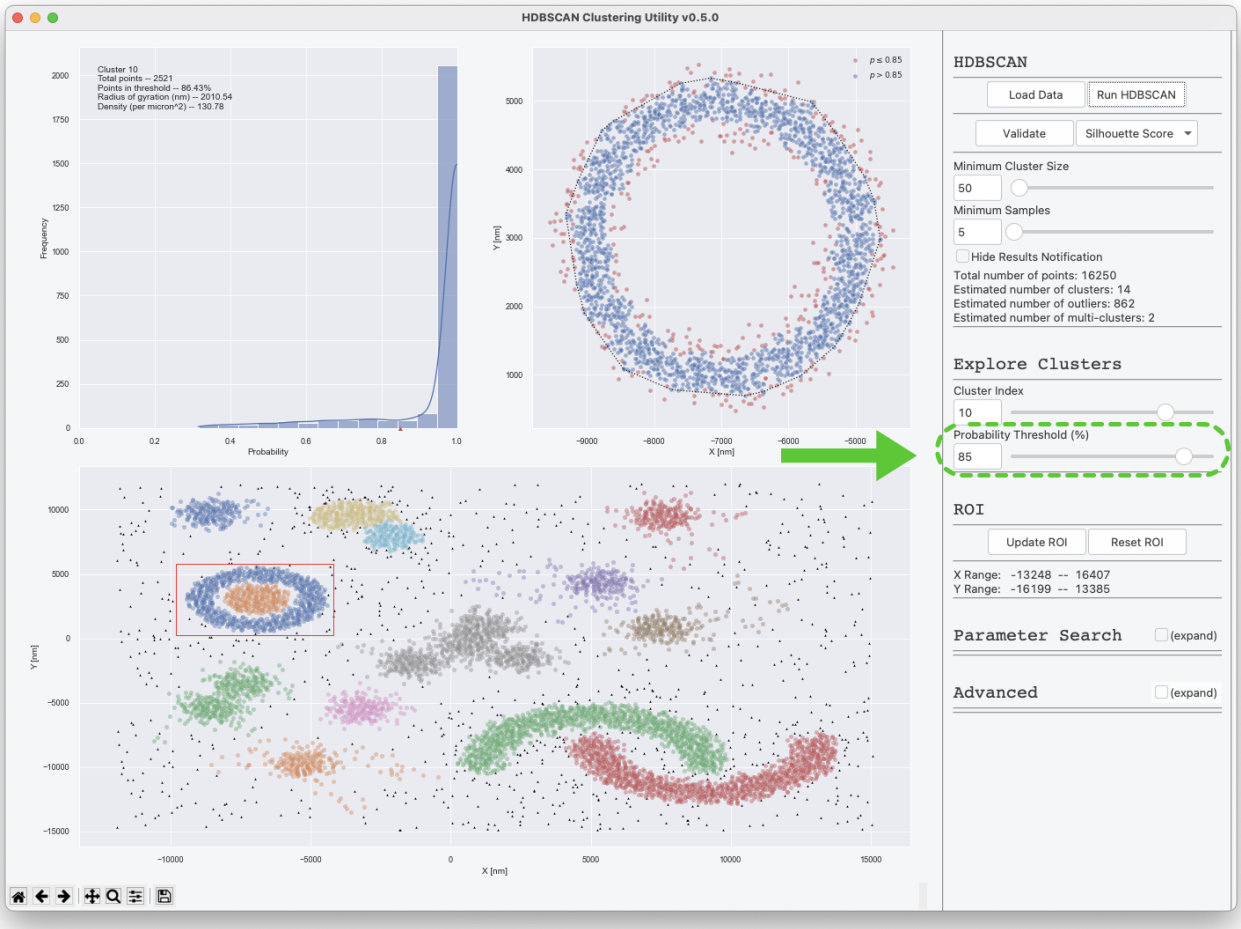
3. Explore Results



Clusters are color-coded in the bottom chart pane (note that there are limited colors in the palette, and thus colors are eventually re-used with enough clusters). A red bounding box surrounds the selected cluster in the bottom chart pane, and this cluster is expanded in detail with the top-right and top-left chart panes.



Use the "Cluster Index" slider to highlight different identified clusters for analysis.



Use the "Probability Threshold" slider to identify poorly fitting data points or potential multi-clusters.

3 Optimizing Parameters

In order to get meaningful clustering results, the parameters of HDBSCAN need to be tuned to your specific dataset. While HDBSCAN is significantly robust to parameter selection, choosing the appropriate values becomes increasingly important for dense and challenging datasets. This chapter will provide some intuition for each of the two parameters, minimum cluster size and minimum samples, as well as some guidelines for identifying a suitable combination of them for your data.

3.1 Minimum Cluster Size

Minimum cluster size, as the name suggests, represents the minimum number of objects needed to identify a cluster. This parameter only imposes a lower limit on the size of clusters, with the upper limit being the total size of your dataset. While intuitive in theory, the effects of this parameter on the clustering algorithm may be unexpected when combined with the minimum samples parameter.

3.1.1 Selecting Minimum Cluster Size

For an initial fixed minimum samples value, it is best to start at low minimum cluster sizes and work your way up. The ideal value of this parameter will generally scale with the number of points within your dataset, however this depends on the nature of your clusters. A safe starting point is the smallest possible value of 2, however for large datasets this can result in thousands of small clusters being identified which takes longer to visualize in the software. If you know the number of localizations in your dataset, as well as an estimate of the number of clusters you are expecting, you can use the following formula as a starting point:

$$\texttt{min\_cluster\_size} = 0.25 \times \left(\frac{\text{Total Points}}{\text{Expected Clusters}} \right) \tag{3.1}$$

This formula works best for many similar clusters. If your dataset has drastically different cluster densities or cluster sizes, start with a lower value.

Figure 5 illustrates the effects that changing the minimum cluster size parameter has on the clustering results with the minimum samples parameter held constant. For a minimum cluster size value of 5, the algorithm detects many more clusters than are represented in the data. Increasing the value to 10, we see that many of the tiny clusters found in the noise are now eliminated. Further increasing the value to 100, we see that only the clearly visible clusters are detected. Some of the clusters appear noisy and overly spread out. This cannot be addressed purely with the minimum cluster size, and we need to start using the minimum samples parameter.

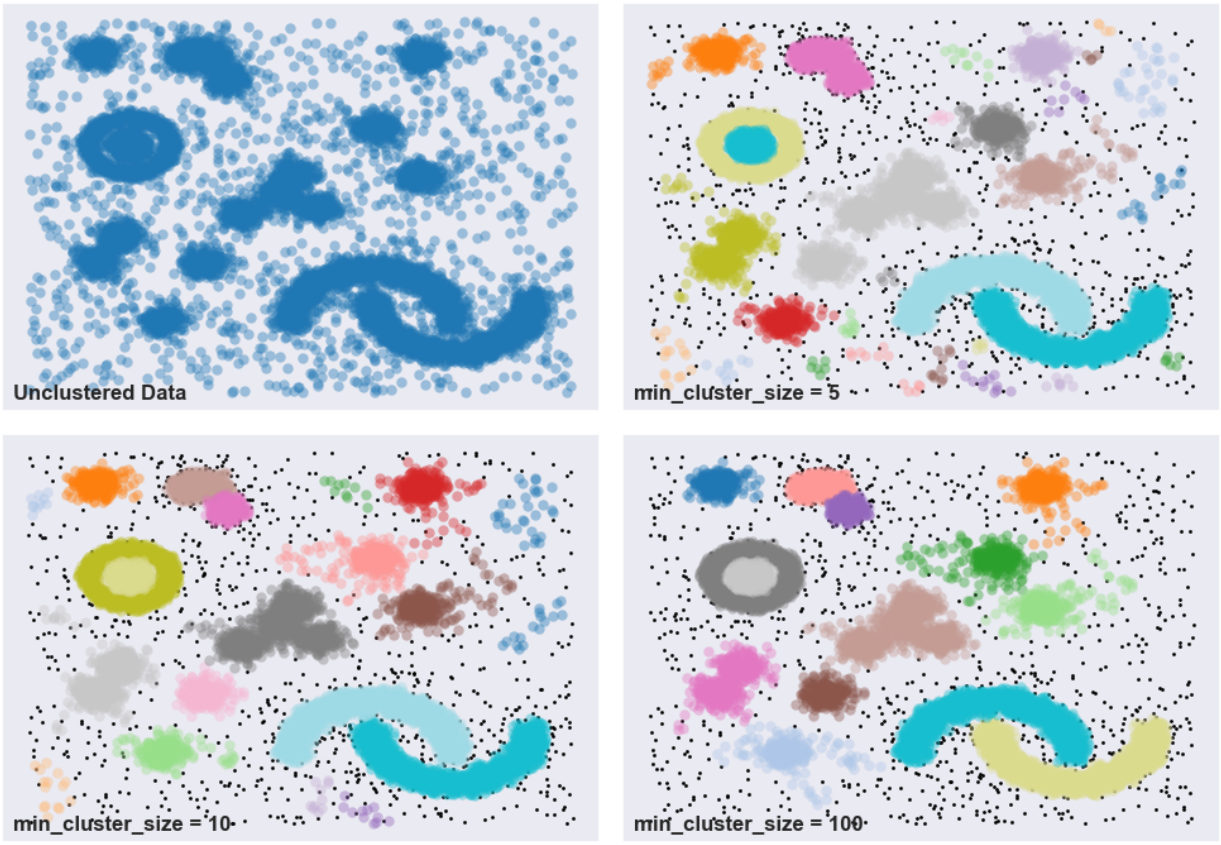


Figure 5: Effect of minimum cluster size on clustering results with a fixed minimum samples value of `min_samples`=5. The top left frame shows the raw unclustered data from the medium training dataset included with the software distribution. The remaining three frames display the clustering results for various minimum cluster size values.

3.2 Minimum Samples

Minimum samples is not as intuitive as minimum cluster size, however it can be summarized as a measure of how conservative the clustering algorithm is. The larger the value of minimum samples, the more conservative the algorithm will be when determining which points are members of a cluster. This parameter is best used to eliminate noise from clusters with too much spread and is very useful for data with lower signal to noise ratios (SNR).

3.2.1 Selecting Minimum Samples

As with selecting minimum cluster size for a fixed minimum samples value, it is best to start low and gradually increase this parameter. Large values of minimum samples can lead to incredibly taxing computations, thus it is recommended to use the lowest minimum samples value that still provides sensible results. For powerful workstations with large amounts of memory (>64Gb RAM), choosing large minimum samples values is possible however the computation will take time.

Unlike minimum cluster size, there is no formula for selecting a starting point for the minimum samples parameter. This parameter will be heavily dependent on the size of the dataset, as well as the SNR of the dataset. Some basic guidelines are as follows: for larger datasets, a larger value of this parameter is generally better; the same applies to low SNR datasets, as increasing this parameter will result in the algorithm being less likely to include noise in the identified clusters.

Figure 6 illustrates the effects that changing the minimum samples parameter has on the clustering results with the minimum cluster size parameter held constant. For a minimum samples value of 5, we recover the same results from our minimum cluster size exploration. This is an adequate clustering, however many of the clusters are noisy. Increasing the value to 10, we see that much of the noise is eliminated and many of the clusters are more well defined. In addition, we also discover two new clusters which were part of a larger cluster when minimum samples was set to 5. Further increasing the value to 100, we eliminate a significant amount of the noise for all of the clusters. We also lose some detail in a few of the tightly packed clusters, which is a necessary trade-off when optimizing these parameters.

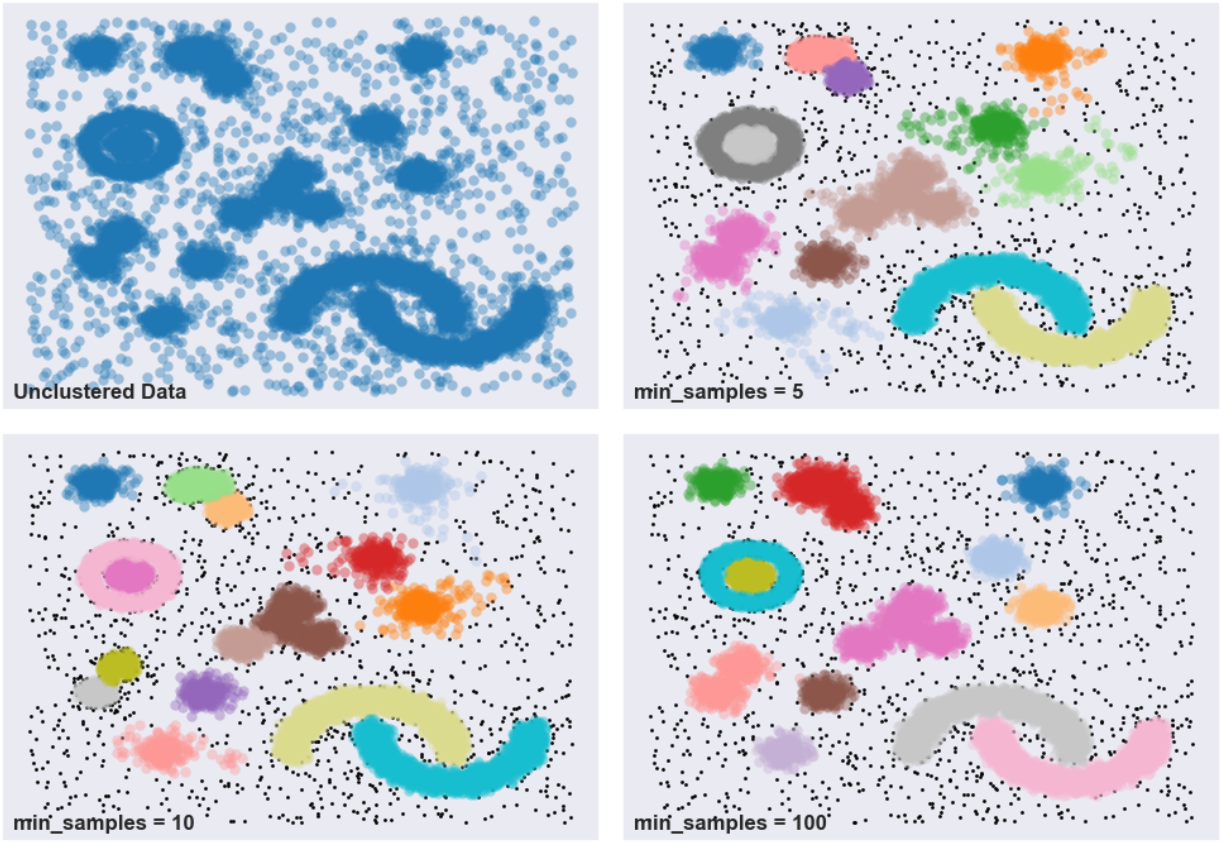


Figure 6: Effect of minimum samples on clustering results with a fixed minimum cluster size value of `min_cluster_size`=100. The top left frame shows the raw unclustered data from the medium training dataset included with the software distribution. The remaining three frames display the clustering results for various minimum samples values.

3.3 Comprehensive Parameter Search

For certain datasets, you will need to test dozens or even hundreds of parameter combinations before finding the one that works best. This can be a very time consuming process, which is why a comprehensive parameter search function is included in the software. This function accepts a set of upper and lower limits on each parameter, then it iterates over a range of possible combinations randomly. This process is referred to as a "Random Search" for hyper parameters in machine learning, differing from the commonly used "Grid Search" which iterates over every possible combination. The benefit of performing a random search stems mainly from the reduced computational time. Random search does not always find the absolute best parameter combination, however it often finds one of the best in a fraction of the time that grid search takes.

3.3.1 Initial Search

When performing a random search, the initial range should be very broad. In order to get an evenly-distributed parameter space, the random search has its full range reduced to 100x100 possible combinations. This reduction in resolution makes it more clear which regions of the parameter space provide better results, and provides a more evident path for refining the parameter search.

Once the parameter search is complete, a new window will open in the software containing the results of the search. The results are displayed as a heatmap of the parameter space, accompanied by the full table of results. For future use, the table is saved to a `.csv` file in the `Params` folder located in the main installation directory of the program.

3.3.2 Refining the Search

Using the heatmap and table of results from your initial search, regions of the parameter space which provide the highest scoring results can be further investigated. Generally, there will be multiple parameter combinations which provide similar scores, and sometimes these combinations are very different. To decide which regions to explore further, run the clustering algorithm for a few of the high-scoring combinations from each region and qualitatively assess the resulting clusters. Similar combinations will provide similar clustering patterns, however for small parameter values (such as minimum samples < 10) the differences between combinations may be large.

The resolution of the parameter values is limited to integer values, so the parameter search can only be refined so much. Often times, minimum cluster size and minimum samples will not offer enough detail to get perfect clustering results for every cluster in your dataset. To circumvent this issue, you can use the region of interest (ROI) tool to study select groups or individual clusters in your dataset.

3.4 Selecting ROIs

The ROI tool allows you to restrict your investigation to a specific spatial region of your dataset. Using this tool can provide a few universal benefits, as well as improvements for very specific use cases. Two immediate benefits are faster computations and easier parameter searching. Due to the reduced number of localizations within the ROI, the clustering algorithm can perform much faster. This improvement in efficiency applies equally to the parameter search, which is also made simpler by not having to balance potentially conflicting parameter combinations across as many clusters.

A more dataset-specific benefit would be analyzing clusters with varying levels of noise. Isolating noisy clusters or compact groups of clusters to individually perform cluster analysis on allows for more freedom in parameter selection. A further consequence of isolating suspected cluster groups is the improved level of detail and ability to identify individual clusters within the group, as illustrated in Fig. 7.

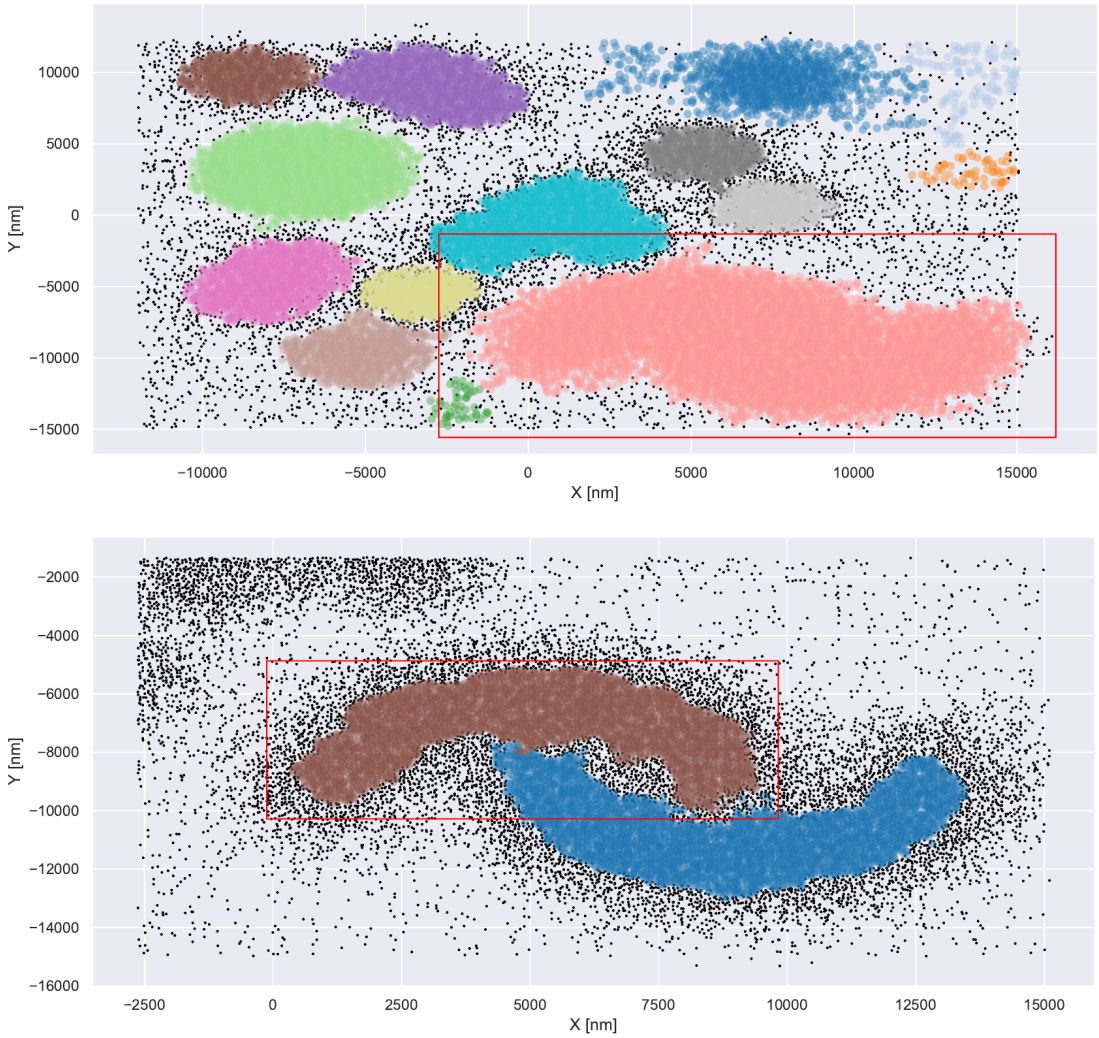


Figure 7: ROI selection for increased clustering detail.

Note: when isolating single clusters with the ROI tool, make sure to enable the "allow single clusters" setting. Without this setting enabled, the clustering algorithm is unable to return individual clusters for analysis.

3.5 Advanced Clustering Parameters

If neither minimum cluster size, minimum samples, nor the ROI tool provide enough flexibility for clustering your dataset, there are two additional advanced clustering parameters which may be used to extend the investigation: alpha and the cluster selection method. By default, these two parameters are set to the optimal values for almost all types of data, however for particularly challenging datasets, they exist. Additionally, there is also the option to allow a single cluster.

3.5.1 Alpha

In the original HDBSCAN documentation, the authors recommend against changing this parameter as it is part of the `RobustSingleLinkage` framework beneath HDBSCAN[10]. This parameter has a similar effect to the minimum samples parameter in determining how conservative the clustering algorithm is, however it is much more

aggressive. Increasing the value of alpha will provide more conservative clustering, however it is recommended to try multiple times using only minimum samples before changing alpha.

3.5.2 Cluster Selection Method

The cluster selection method determines how the algorithm identifies clusters from the condensed linkage tree generated by `RobustSingleLinkage`. By default, this option is set to use an "Excess of Mass" (EoM) algorithm. The EoM algorithm has a tendency to identify a few large clusters accompanied by multiple smaller clusters. If your dataset is expected to have many small homogeneous clusters, the "Leaf" clustering selection method is more appropriate. This method selects leaf nodes from the condensed linkage tree which tend to coincide with smaller homogeneous clusters.

3.5.3 Allow Single Cluster

For ROI selections containing only a single cluster or full datasets which are suspected to contain a single cluster, the option to allow a single cluster should be enabled. By default, the HDBSCAN algorithm is unable to detect single clusters and differentiate them from noise. Enabling this feature will allow single clusters to be identified and differentiated from noise. It is recommended to only use this feature if there is only one cluster expected in your data, otherwise the clustering algorithm will be biased towards identifying large clusters.

4 Exploring Clustering Results

Once the clustering is complete and results are displayed, it is time to explore those results. The results are broken up into two sections: the full dataset / current ROI, and individual cluster statistics. The first section provides general information about your dataset or current ROI, while the second section provides detailed information about all identified clusters individually. Exploring both of these sections and understanding what each quantity represents is important when deciding whether to continue the parameter selection process.

4.1 Full Dataset / Current ROI

Results for the full dataset / current ROI will be displayed in the HDBSCAN tools menu on the right side of the clustering utility (Fig. 8). These results update when the "Run HDBSCAN" button is pressed, and they include: the total number of points, estimated number of clusters, estimated number of outliers, and estimated number of multi-clusters. This results are also shown in a pop-up notification once the clustering is complete.

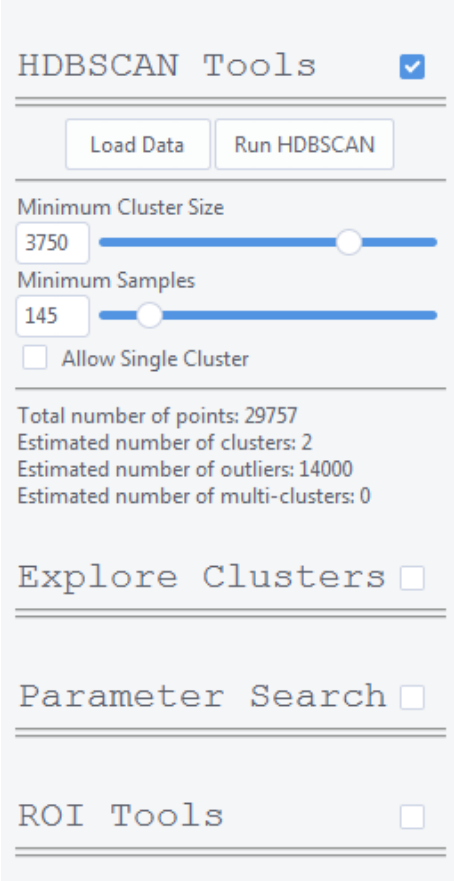


Figure 8: Full dataset / current ROI clustering results location.

4.1.1 Total Number of Points

The total number of points is self explanatory, however this quantity represents the number of points within the current selected ROI, not necessarily the full dataset.

4.1.2 Estimated Number of Clusters

The estimated number of clusters represents the number of unique labels (clusters) identified by the HDBSCAN algorithm. This quantity is an "estimated" quantity due to the fact that HDBSCAN does not know the true number of clusters ahead of time, and depending on which parameter values are used, this estimation may be inaccurate.

4.1.3 Estimated Number of Outliers

Similarly to the estimated number of clusters, the estimated number of outliers represents the size of the outlier cluster (label "-1"). These are all of the points which do not belong to any labelled cluster, and again the quantity is and "estimated" quantity.

4.1.4 Estimated Number of Multi-Clusters

The estimated number of multi-clusters represents the number of identified individual clusters which are likely to contain sub-clusters. This quantity can inform your decision to modify the clustering parameters, as a non-zero number of multi-clusters is indicative of poor clustering. Multi-clusters are identified by finding deep valleys in the cluster membership probability distribution, as shown in Fig. 9. Cluster membership probability represents the likelihood of a single localization belonging to the cluster that it is labelled as, and points with a probability of 0.0 are labelled as outliers. Deep valleys in the probability distribution often hint at sub-clusters which are physically disconnected from the core cluster. Cluster membership probability is elaborated upon in Section 4.2.2.

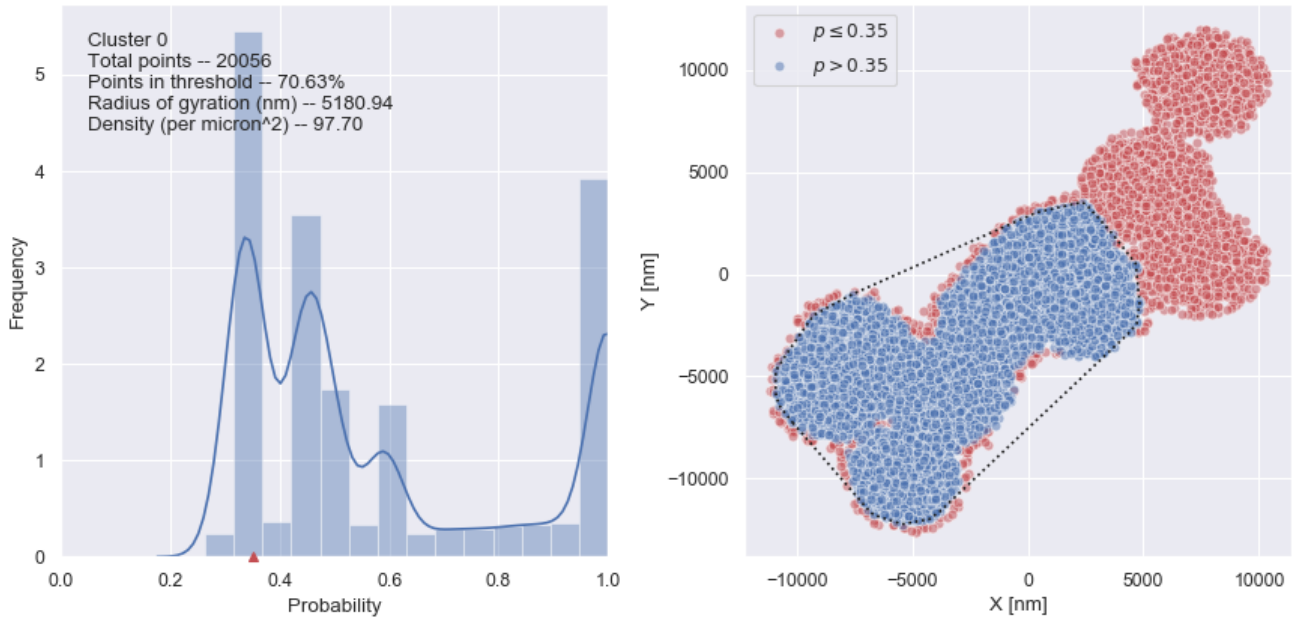


Figure 9: Multi-cluster identification using the cluster membership probability distribution curve. The membership probability distribution (left) contains multiple distinct peaks, each signifying a large sub-cluster within the identified cluster. At a threshold of $p = 0.35$, the identified cluster (right) loses a large percentage of members (shown in red). From the number of peaks in the distribution, it is likely that there are at least 4 sub-clusters present.

4.2 Individual Cluster Statistics

Exploring the results of the full dataset / current ROI provides a few fundamental pieces of clustering information, however these quantities do not give any insight to the properties of individual clusters. Individual cluster statistics are displayed in the top-left frame containing the probability membership distribution (Fig. 9, left). These results correspond to the individual cluster that is currently selected, and they include: the total points, probability threshold information, radius of gyration, and cluster density.

4.2.1 Total Points

The total points in a cluster represents the number of localizations which have been labelled as members of that cluster.

4.2.2 Probability Threshold

The probability distribution shown in the top-left frame of the clustering software represents the cluster membership probability for all points within that cluster. A probability threshold can be set to exclude "weak" members with low membership probability. This threshold is a powerful tool for optimizing individual clusters prior to performing further analysis. The percentage of points within the probability threshold is displayed on screen, as well as a red arrow marking the current position of the threshold. Real-time updating of the probability threshold is displayed, allowing for fast identification of outliers and multi-clusters.

4.2.3 Radius of Gyration

In various fields, the radius of gyration is commonly used to represent the physical size of a molecule or cluster of molecules[4,13,15]. This measure of the size of a cluster is robust in the sense that both the density and extent of the cluster are reflected. The radius of gyration, R_G, of a cluster containing N localizations is defined as

$$R_G^2 = \sum_{i=1}^{N} \frac{(w_i|\mathbf{r}_i - \mathbf{r}_{\text{cm}}|)^2}{N}, \tag{4.1}$$

where $\mathbf{r}_i = (x_i, y_i, z_i)$ is the location of particle i, $\mathbf{r}_{\text{cm}} = (x_{\text{cm}}, y_{\text{cm}}, z_{\text{cm}})$ is the centre of mass of the cluster, and w_i is the associated weight or "mass" of particle i. The larger the radius of gyration, the greater the physical extent or density of the cluster. Changes in the radius of gyration can be used to inform the probability threshold, with smaller radii generally corresponding to more compact clusters.

4.2.4 Density

Along with the radius of gyration, the density of the cluster in terms of localizations per unit area (scales either to μm^2 or nm^2 depending on magnitude) is provided. Unlike the radius of gyration, density is not robust to outliers and a single distant point in the cluster can significantly skew the value. The area or volume of the cluster is computed using the convex hull algorithm, which can be thought of as drawing straight lines or stretching a rubber band across the outer-most points[1]. This bounding perimeter can be seen in the right side of Fig. 9 as the dashed black line. A limitation to this approach is that concave clusters will experience artificially inflated areas / volumes, and consequently report lower densities.

- □ mathematical definition
- □ probability and tie-in to density (jumps in density, etc.)
- □ area computed using convex hull
  - associated limitations for concave clusters

5 Validating Clustering Results

For unsupervised clustering problems, evaluating the validity of clustering results is incredibly important if the results are to be interpreted. This importance is predominantly a consequence of optimal clustering results not being known a priori, and poor clustering may lead to misinterpreting trends in your dataset. There are two approaches to cluster validation: quantitative and qualitative methods. It is strongly recommended that both of these approaches are considered when validating results, as each has specific benefits and drawbacks. Additionally, there are a few fundamental concepts used for assessing the validity of clustering that are necessary for either approach.

5.1 Fundamental Concepts of Cluster Validity

5.1.1 Compactness

Compactness describes how close members of each cluster are to one another [6]. A higher degree of compactness corresponds to a lower variance or lower spread of points within the cluster. Minimizing the variance within a cluster is one of the steps required for achieving optimal clustering.

5.1.2 Separation

Separation describes how far clusters are from other clusters [6]. There are multiple approaches to measuring the inter-cluster separation, with common examples including: linkage, comparison of centroids, and mean pairwise distance. Single linkage measure the distance between the two closest members of the two clusters being compared, while complete linkage measures the distance between the two furthest members. Comparison of centroids measures the distance between the geometric centroids of the two clusters. Lastly, mean pairwise distance measures the pairwise distances of all points between both clusters and returns the mean value.

5.2 Quantitative Validation

There are three numerical validation techniques offered in the clustering software: the Silhouette score, Calinski-Harabasz index, and Davies-Bouldin index. Each of these three scores approaches the validation problem in a different manner, and as such there are strengths and weaknesses to each score.

5.2.1 Silhouette Score

The Silhouette is a graphical method for interpreting cluster validity proposed by Rousseeuw [14], and it incorporates both the compactness and separation present in clustering. The Silhouette score for each point in the dataset is defined as

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{5.1}$$

where a is a measure of the intra-cluster compactness and b is a measure of the inter-cluster separation. More specifically, $a(i)$ represents the mean distance between a point i and all other samples j within it's cluster C_i

$$a(i) = \frac{1}{N_i - 1} \sum_{j \in C_i, i \neq j} d(i, j),$$

where distance $d(i, j)$ is defined as

$$d(i, j) = |\mathbf{r}_i - \mathbf{r}_j|.$$

Similarly, $b(i)$ represents the mean distance between a point i and all other samples j within the next nearest cluster C_j

$$b(i) = \min_{i \neq j} \frac{1}{N_j} \sum_{j \in C_j} d(i,j).$$

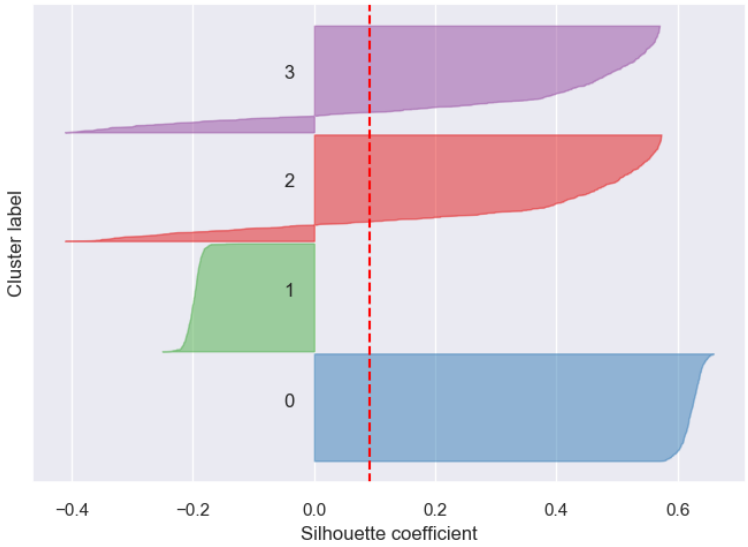


Figure 10: Silhouettes for four identified clusters. The dashed red line represents the mean Silhouette score, with higher scores denoting better clustering results. Negative scores in clusters "1", "2", and "3" represent misclassification.

The Silhouettes of multiple clusters are shown in Fig. 10, with the Silhouette score of the full dataset simply being the mean Silhouette score across all samples. Taking a closer look, we can see that cluster "0" shows high Silhouette coefficients for all of its members, indicating good clustering. Clusters "2" and "3" show sharp negative spikes, indicating a population of outliers. Cluster "1" has exclusively negative Silhouette coefficients, suggesting very poor labelling.

Advantages. The score is bounded on the range [-1,1], with -1 representing incorrect cluster labelling, 0 representing overlapping clusters, and +1 representing highly dense clustering[12]. This normalization makes it useful for comparison across different datasets, as well as for use with parameter optimization techqniques. The score is higher when both cluster compactness and cluster separation are clear, relating to the conventional idea of what a cluster is.

Drawbacks. The Silhouette score will be lower for concave clusters or oddly shaped density-based clusters due to it's mathematical formulation. The score is also very computationally intensive to compute, taking significantly longer than the remaining two quantitative scores.

5.2.2 Calinski-Harabasz Index

The Calinski-Harabasz index, also known as the Variance Ratio Criterion, is a ratio of the inter-cluster dispersion to the intra-cluster dispersion[7]. For k clusters, the Calinski-Harabasz index is defined as

$$CH(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1}, \tag{5.2}$$

where B_k is the between-cluster dispersion matrix

$$B_k = \sum_q n_q (c_q - c)(c_q - c)^T,$$

and W_k is the within-cluster dispersion matrix

$$W_k = \sum_{q=1}^{k} \sum_x (x - c_q)(x - c_q)^T.$$

Intra-cluster and inter-cluster dispersion can be thought of as a different formulation of the more general compactness and separation measures with which we are already familiar. A higher score indicates denser and more spread clusters.

Advantages. As with Silhouettes, this score is higher for dense and well separated clusters, agreeing with intuition. Additionally, the linear algebra approach to computing this score using matrices makes it incredibly fast to calculate when compared to the Silhouette score.

Drawbacks. As with the Silhouette score, the Calinski-Harabasz index will be lower for concave clusters or oddly shaped density-based clusters due to it's mathematical formulation.

5.2.3 Davies-Bouldin Index

The Davies-Bouldin index is defined as the average similarity between each cluster and the cluster most similar to it, given by

$$DB = \frac{1}{k} \sum_{i=1}^{k} \max_{i \neq j} R_{ij}. \tag{5.3}$$

The quantity R_{ij} is a measure of the cluster similarity

$$R_{ij} = \frac{s_i + s_j}{d_{ij}},$$

which trades off the compactness (s) of each cluster with the separation (d) between the two clusters. Each R_{ij} will be a $k \times k$ matrix where k is the number of identified clusters. The best possible score for this index is zero, and higher scores indicate worse clustering results.

Advantages. As with the Calinski-Harabasz index, this score is very fast compared to the Silhouette score.

Drawbacks. As with the Silhouette score and the Calinski-Harabasz index, the Davies-Bouldin index will be worse for concave clusters or oddly shaped density-based clusters due to it's mathematical formulation. The mathematical formulation also limits the score to Euclidean geometry, so the data must be two or three dimensional. Lastly, even if the score is good (low values), insightful clustering results are not at all guaranteed.

5.2.4 Limitations of Quantitative Validation

Numerical approaches to evaluating clustering results can be a powerful tool, however they do not always reveal the full story. There are many highly specific situations where quantitative validation fails and either scores a really poor clustering result well, or scores a really good clustering result poorly. It is thus always important to stop and carefully observe your clustering results qualitatively once a quantitative evaluation or parameter search has been completed.

5.3 Qualitative Validation

Clustering results can be qualitatively evaluated by an expert in the field using the tools available in the clustering software. Qualitative validation can also be done by non-experts, however this does not guarantee that the optimal clustering results will be achieved. The first important tool to use is the "Cluster Index" slider in the "Explore Clusters" section of the software. Clusters can be explored on an individual basis in great detail using this tool, with the added ability to apply probability thresholds. Probability thresholds allow you to determine how reliable the clustering results are, namely whether or not obvious outliers have high or low membership probabilities. If a qualitatively obvious outlier has a high membership probability, the clustering results are likely sub-optimal. Conversely, if there are many outliers but they all have low membership probability, the clustering results are likely not as poor as quantitative validation may suggest. Silhouettes can also be used for qualitatively evaluating the clustering results, even if they predominantly offer a quantitative evaluation. It is important for an expert in the field to conduct the qualitative validation for the final clustering results, as even the nicest looking clusters may be misleading.

6 Interpreting Clustering Results

- ☐ read paper on clustering validation techniques in detail [6]

References

[1] A.M. Andrew. Another efficient algorithm for convex hulls in two dimensions. *Information Processing Letters*, 9(5):216–219, 12 1979. ISSN 0020-0190. doi: 10.1016/0020-0190(79)90072-3. URL https://www.sciencedirect.com/science/article/pii/0020019079900723?via%3Dihub.

[2] David L. Davies and Donald W. Bouldin. A Cluster Separation Measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(2):224–227, 4 1979. ISSN 0162-8828. doi: 10.1109/TPAMI.1979.4766909. URL http://ieeexplore.ieee.org/document/4766909/.

[3] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Technical report, Institute for Computer Science, University of Munich, 1996. URL www.aaai.org.

[4] Marshall Fixman. Radius of Gyration of Polymer Chains. *The Journal of Chemical Physics*, 36(2):306–310, 1 1962. ISSN 0021-9606. doi: 10.1063/1.1732501. URL http://aip.scitation.org/doi/10.1063/1.1732501.

[5] Sudipto Guha, Rajeev Rastogi, Kyuseok Shim, Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. CURE: an efficient clustering algorithm for large databases. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data - SIGMOD '98*, volume 27, pages 73–84, New York, New York, USA, 1998. ACM Press. ISBN 0897919955. doi: 10.1145/276304.276312. URL http://portal.acm.org/citation.cfm?doid=276304.276312.

[6] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On Clustering Validation Techniques. *Journal of Intelligent Information Systems*, 17(2/3):107–145, 2001. ISSN 09259902. doi: 10.1023/A:1012801612483. URL http://link.springer.com/10.1023/A:1012801612483.

[7] Marcin Kozak. "A Dendrite Method for Cluster Analysis" by Caliński and Harabasz: A Classical Work that is Far Too Often Incorrectly Cited. *Communications in Statistics - Theory and Methods*, 41(12):2279–2280, 6 2012. ISSN 0361-0926. doi: 10.1080/03610926.2011.560741. URL http://www.tandfonline.com/doi/abs/10.1080/03610926.2011.560741.

[8] Leland McInnes, John Healy, and Steve Astels. Comparing Python Clustering Algorithms — hdbscan 0.8.1 documentation, 2016. URL https://hdbscan.readthedocs.io/en/latest/comparing_clustering_algorithms.html.

[9] Leland McInnes, John Healy, and Steve Astels. How HDBSCAN Works — hdbscan 0.8.1 documentation, 2016. URL https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html.

[10] Leland McInnes, John Healy, and Steve Astels. HDBSCAN: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), 3 2017. doi: 10.21105/joss.00205. URL https://doi.org/10.21105%2Fjoss.00205.

[11] Sophie V Pageon, Philip R Nicovich, Mahdie Mollazade, Thibault Tabarin, and Katharina Gaus. ClusDoC: a combined cluster detection and colocalization analysis for single-molecule localization microscopy data. *Molecular biology of the cell*, 27(22):3627–3636, 2016. ISSN 1939-4586. doi: 10.1091/mbc.E16-07-0478. URL http://www.ncbi.nlm.nih.gov/pubmed/27582387http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5221594.

[12] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011. ISSN ISSN 1533-7928. URL http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html.

[13] Theophanes Raptis and Vasilios Raptis. RDCI: A novel method of cluster analysis and applications thereof in sample molecular simulations. *arXiv*, 6 2013. URL http://arxiv.org/abs/1306.3460.

[14] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. URL https://www.sciencedirect.com/science/article/pii/0377042787901257?via%3Dihub.

[15] Michael J. Saxton. Single-Particle Tracking Analysis using the Radius of Gyration Tensor, Revisited. *Biophysical Journal*, 110(3):487a, 2 2016. ISSN 00063495. doi: 10.1016/j.bpj.2015.11.2604. URL https://linkinghub.elsevier.com/retrieve/pii/S000634951503787X.

[16] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern recognition*. Academic Press, 4th edition, 2009. ISBN 9780080949123.

7 To-Do List

7.1 Standard Operating Protocol

- ☐ add a tutorial section dealing with the training data (maybe after the quick-start guide)
  - specific goals of optimizing and identifying all clusters correctly
  - gain intuition for all parameters
- ☐ complete section 4.2.4 content
- ☐ complete section 6 content
- ☐ a few editing passes couldn't hurt

7.2 Software Utility

7.2.1 Features

- ☐ implement individual cluster saving (dumping dataframe to `.csv`)
  - needs to reflect probability threshold as well
  - maybe add an option for in depth individual cluster analysis in new `TopLevel` window
- ☐ batch mode for individual cluster stats
  - display results and offer to dump to `.csv`
- ☐ implement RipleyK coefficient
  - don't know how useful this is, do some research first
  - will need to add section in SOP if implemented
- ☐ higher dimensional clustering / sample weights on clustering
  - not entirely necessary but good for intensity-based investigations
- ☐ support multiple data channels
  - channel selection and clustering
  - degree of colocalization between channels[11]
- ☐ add option to manually sub-sample smaller datasets
- ☐ add loading bar for parameter search (currently in terminal)

7.2.2 Fixes

- ☐ None (for now...)

7.2.3 Miscellaneous

- ☐ look into software licenses