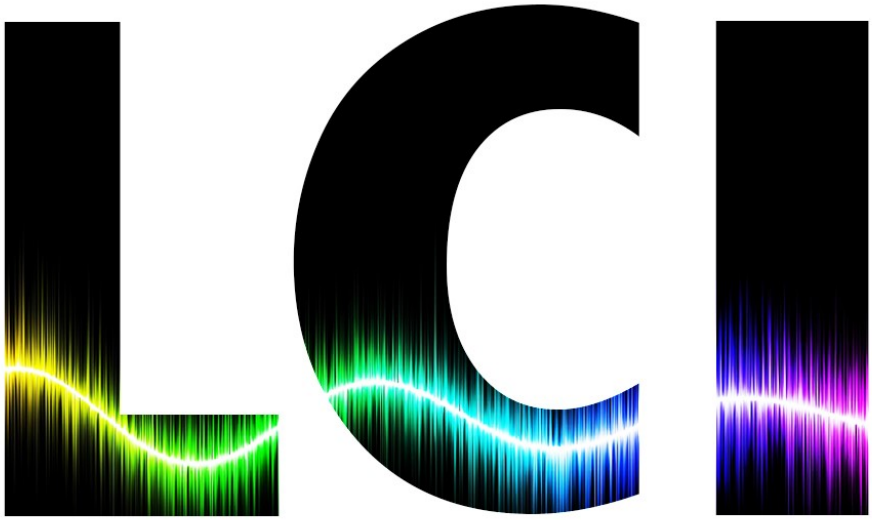


HDBSCAN Cluster Analysis Guide

Live Cell Imaging Resource Laboratory
Philip Ciunkiewicz

Contents

1 Introduction

This document will provide a standard operating procedure 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. 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.

In order to optimize parameters and interpret the results of clustering, it is important to understand how clusters are identified. The 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.

1.1.2 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 [2]. 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. 1). The next check is for the minimum number of neighbors, n (right side of Fig. 1). 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).

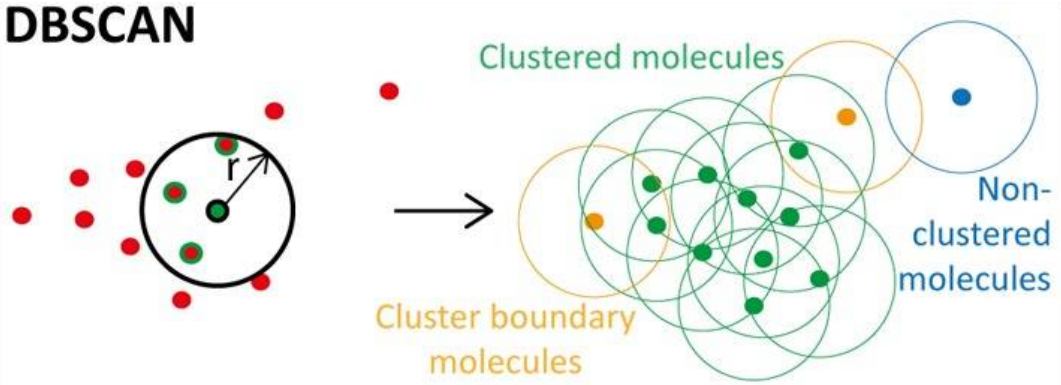
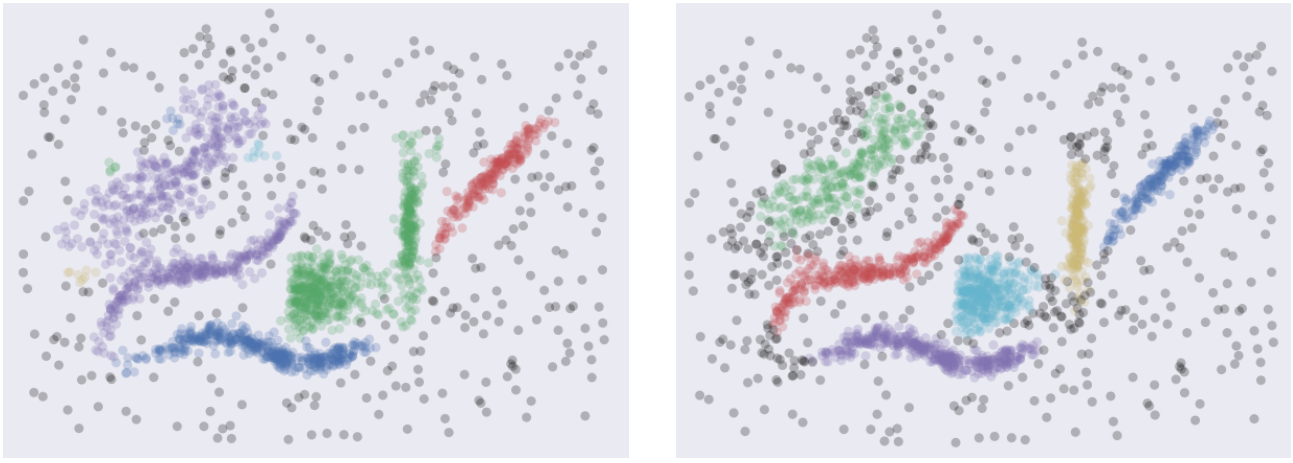


Figure 1: DBSCAN algorithm radius and minimum neighbors density-based clustering[8].

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.

1.1.3 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[7]. A detailed technical description of the algorithm is provided in McInnes et al.[6]. We can illustrate the difference that this approach makes by comparing the results of both algorithms for a challenging synthetic dataset (Fig. 2).



(a) DBSCAN (b) HDBSCAN

Figure 2: Comparison of DBSCAN and HDBSCAN clustering results for a synthetic dataset[5].

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. 2(b) are much more compelling than those of DBSCAN in Fig. 2(a). However, how can we quantitatively assess the quality of clustering to definitively say that one is better than the other?

1.1.4 Evaluating Clusters

Performing cluster analysis without knowing the correct results (ground truth) ahead of time is known 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[10]
- Calinski-Harabaz Index[4]
- Davies-Bouldin Index[1,3]

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 is an example for when qualitative assessment is as important or even more important than any quantitative measure.

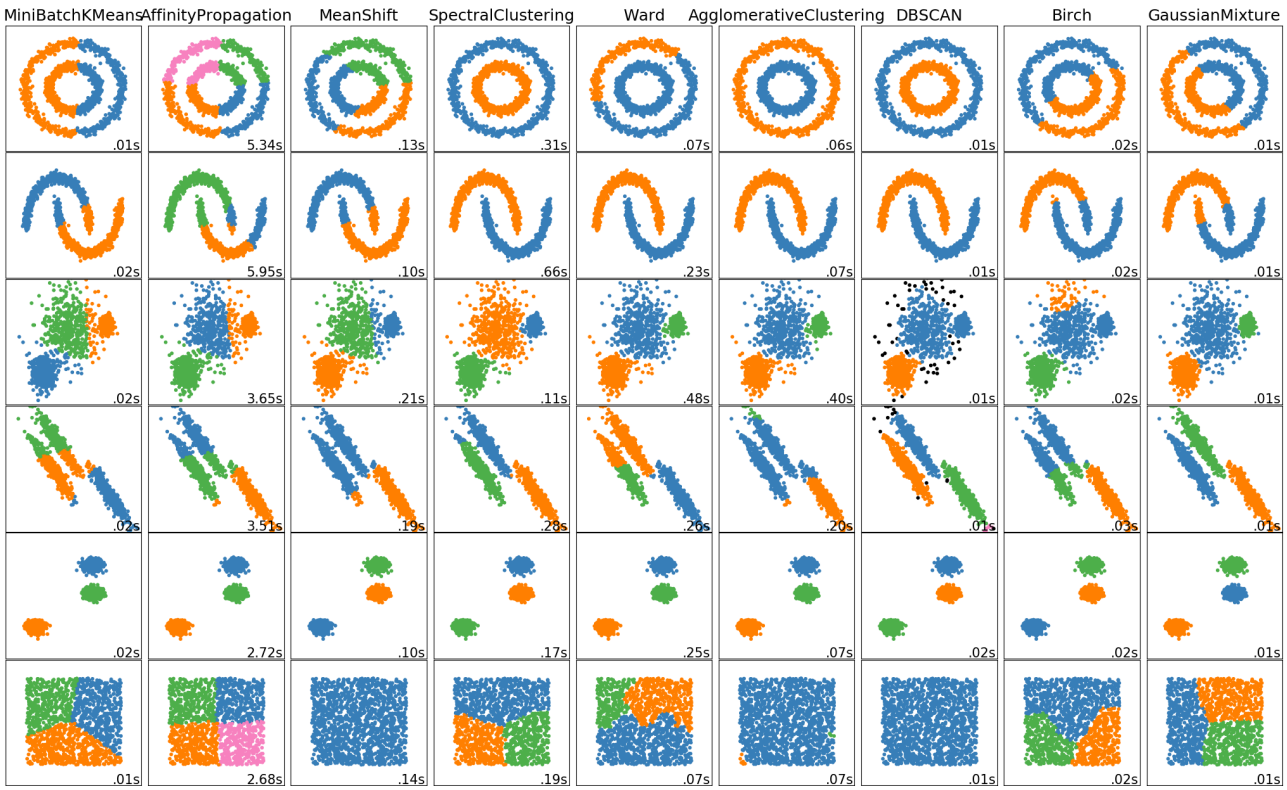


Figure 3: Clustering results for a variety of clustering methods[9].

1.2 Prerequisites

In order to perform any sort of cluster analysis on your data, there are a few preliminary steps which you must take.

1.2.1 SMLM Data

- ☐ ThunderSTORM reconstruction
- ☐ Validate the reconstruction using NanoJ SQUIRREL
- ☐ Hopefully good sample preparation and lowest noise possible

1.2.2 LCI Clustering Software

2 Getting Started

2.1 Installing the Software

2.2 Preparing your Data

2.3 Quick-Start Guide

3 Optimizing Parameters

3.1 Minimum Cluster Size

3.2 Minimum Samples

3.3 Comprehensive Parameter Search

4 Exploring Clustering Results

4.1 Full Dataset

4.1.1 Total Number of Points

4.1.2 Estimated Number of Clusters

4.1.3 Estimated Number of Outliers

4.2 Individual Cluster Statistics

4.2.1 Total Points

4.2.2 Probability Threshold

4.2.3 Radius of Gyration

4.2.4 Relative Density

5 Validating Clustering Results

5.1 Quantitative Evaluation

5.1.1 Silhouette Score

5.1.2 Calinski-Harabaz Index

5.1.3 Davies-Bouldin Index

5.2 Qualitative Evaluation

6 Interpreting Clustering Results

References

[1] 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/.

[2] 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.

[3] 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.

[4] Marcin Kozak. âĂIJA Dendrite Method for Cluster AnalysisâĂİ 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.

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

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

[7] 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.

[8] Sophie V Pageon, Philip R Nicovich, Mahdie Mollazade, Thibault Tabarin, and Katharina Gaus. Clus-DoC: 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.

[9] Fabian Pedregosa, GaÃńl 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 ÃĽdouard 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.

[10] 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.