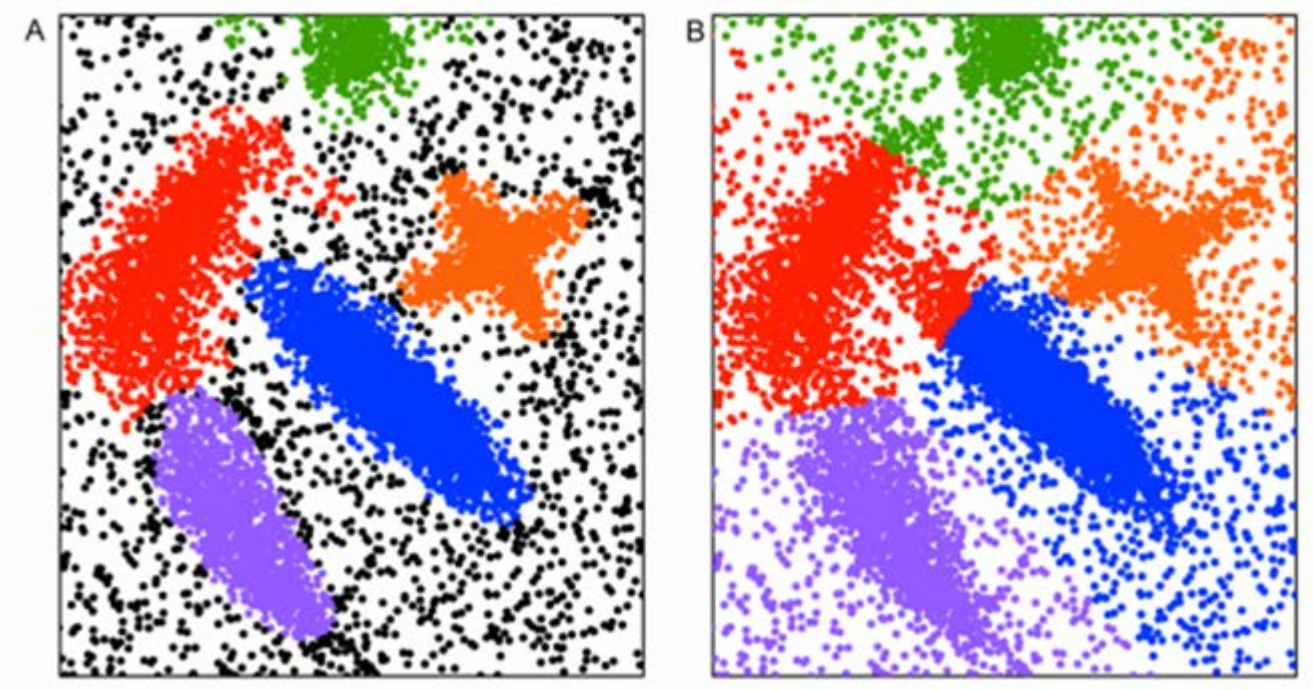
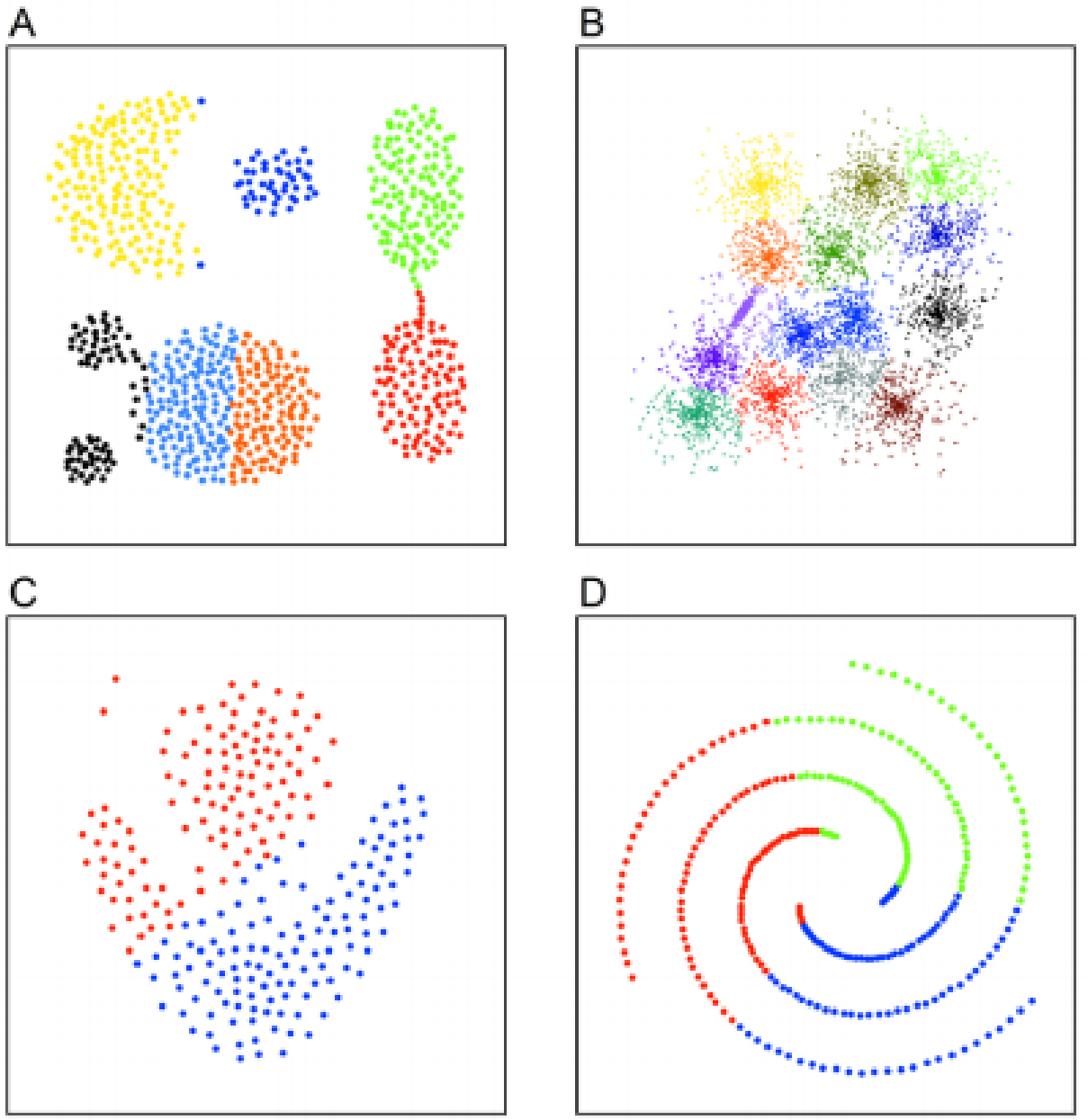
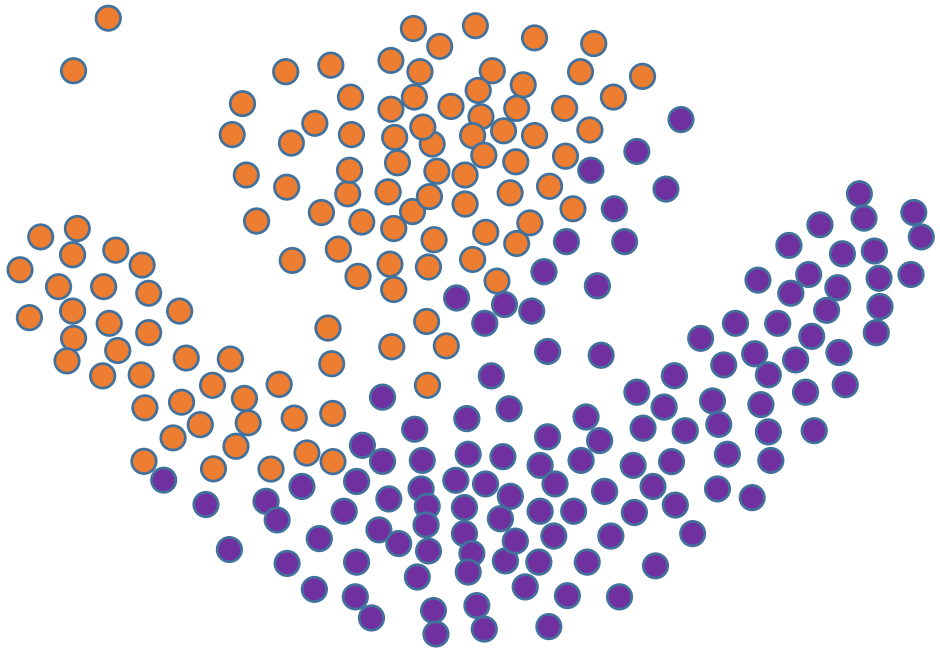
K-means



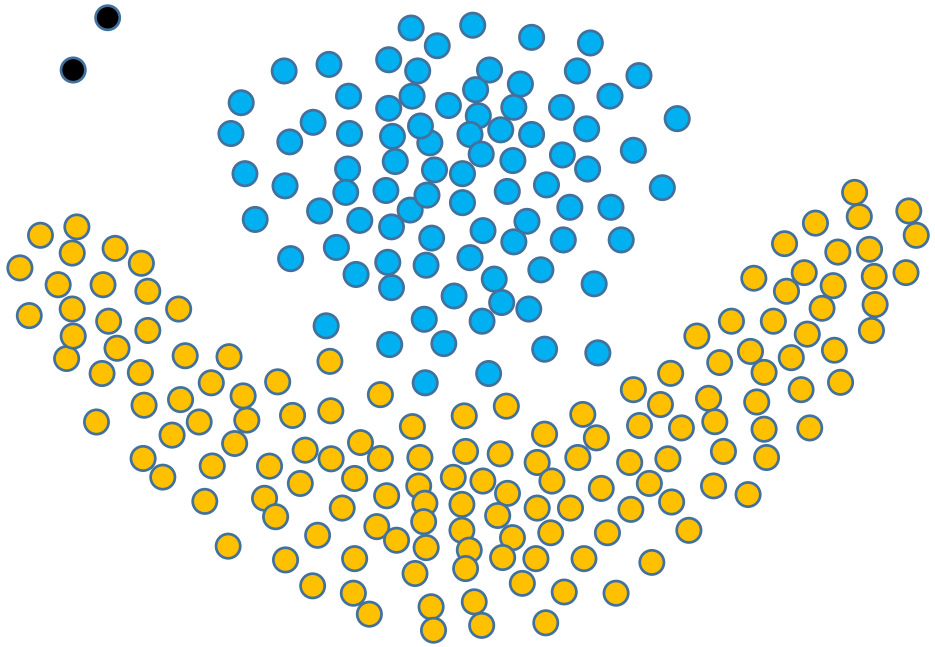
K-means



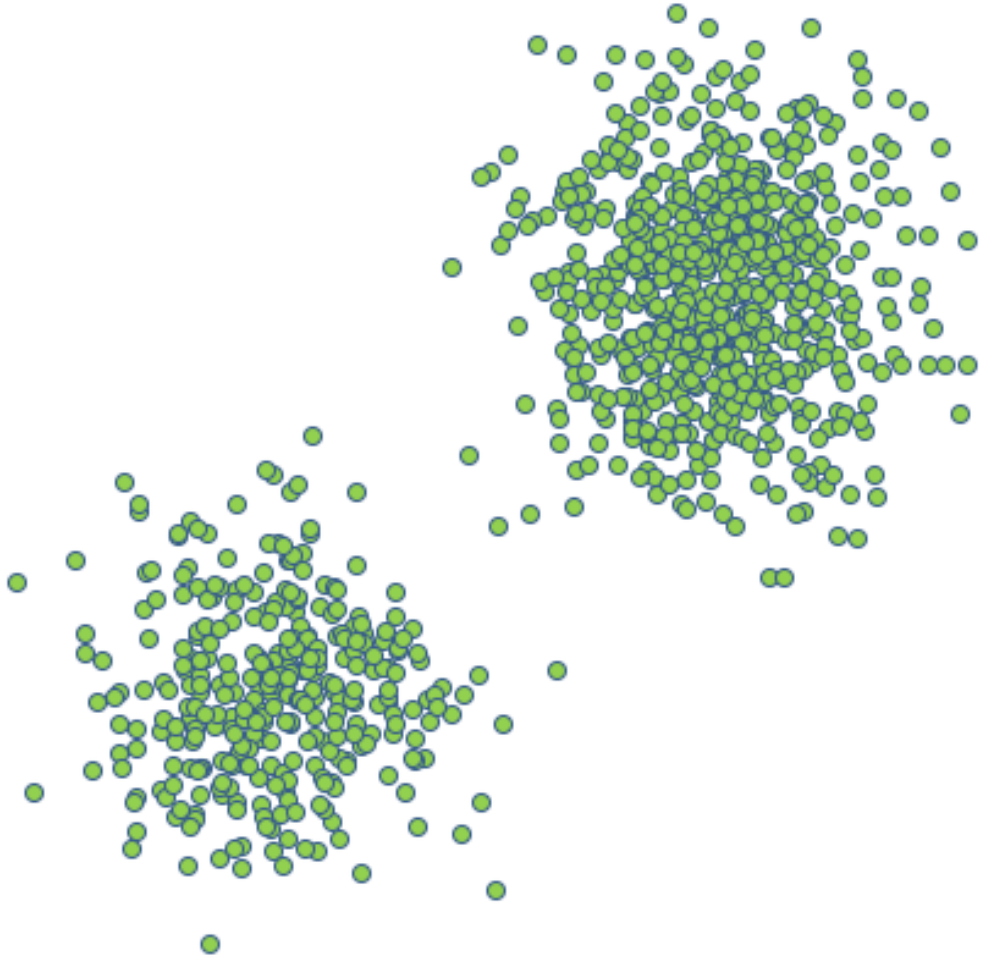


Density-based clustering

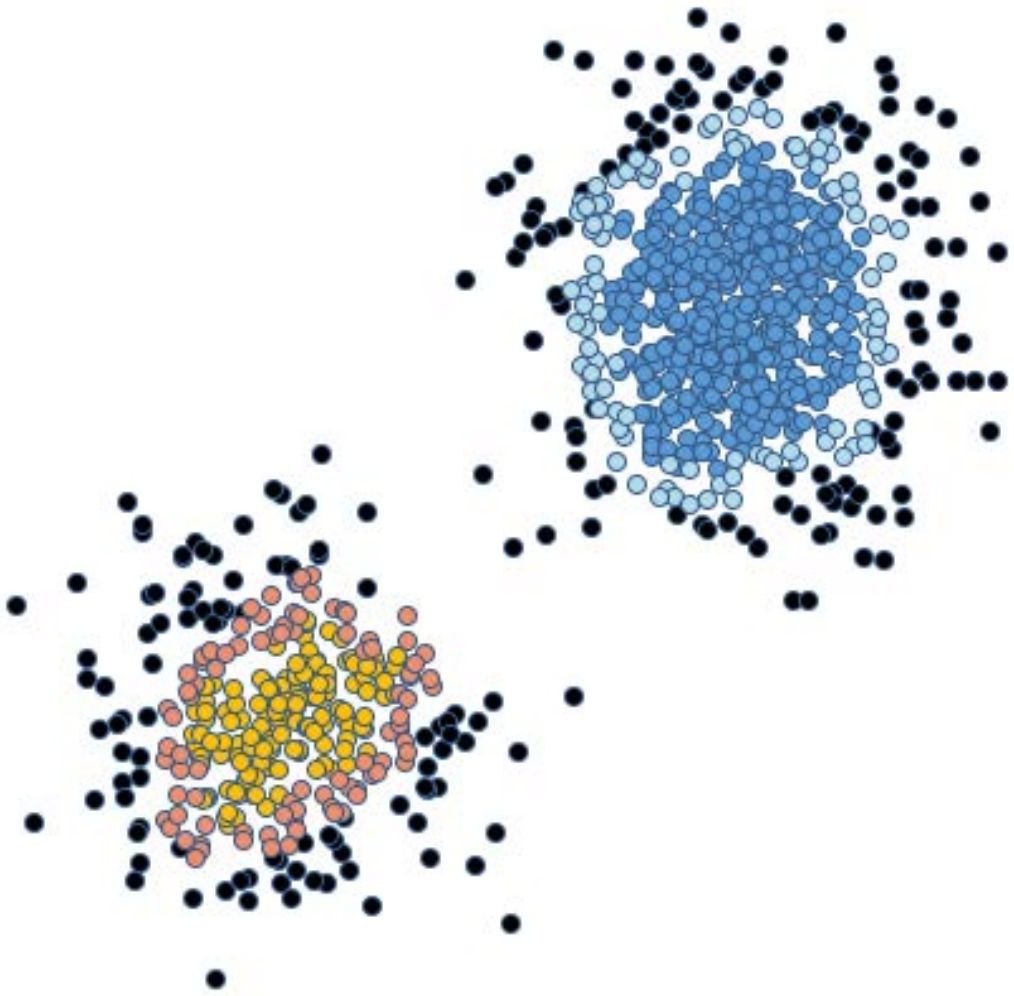
Non-Parametric – does not need to specify number of clusters in advance
Clusters are defined as areas of higher density than the remainder of the data set
Can find clusters of arbitrary shapes
For example: DBSCAN, MeanShift, OPTICS



DBSCAN, analyze the density

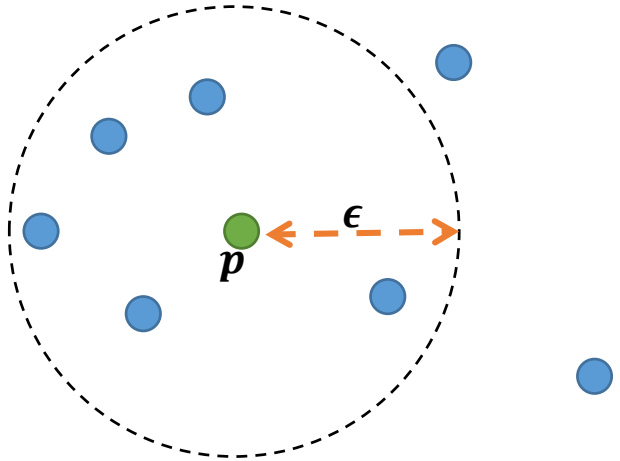


DBSCAN: Core and non-core points



DBSCAN *(Density-based spatial clustering of applications with noise)*

- Input parameters – **minPts** (integer), ϵ (distance)
- **Core points** – A point is a core point if there are at least **minPts** points with distance which is less than ϵ from it:

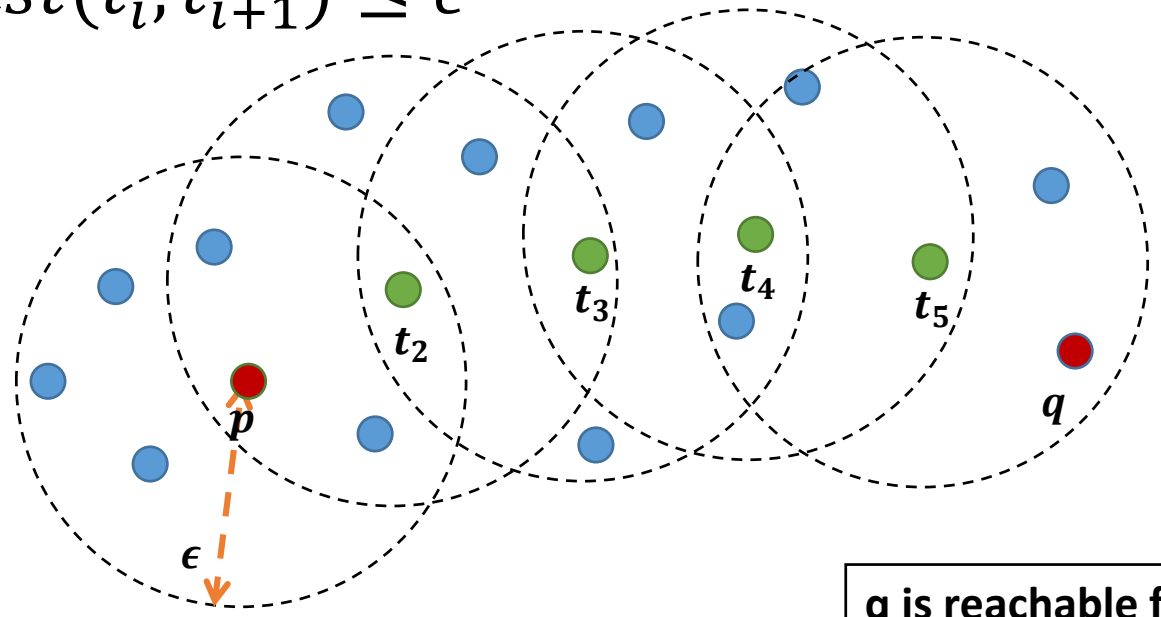


For example, point p here is a core point for **minPts<=5**

DBSCAN – cont'd

The main idea: connect core points that are in distance $\leq \epsilon$ from each other + their neighborhoods into clusters.

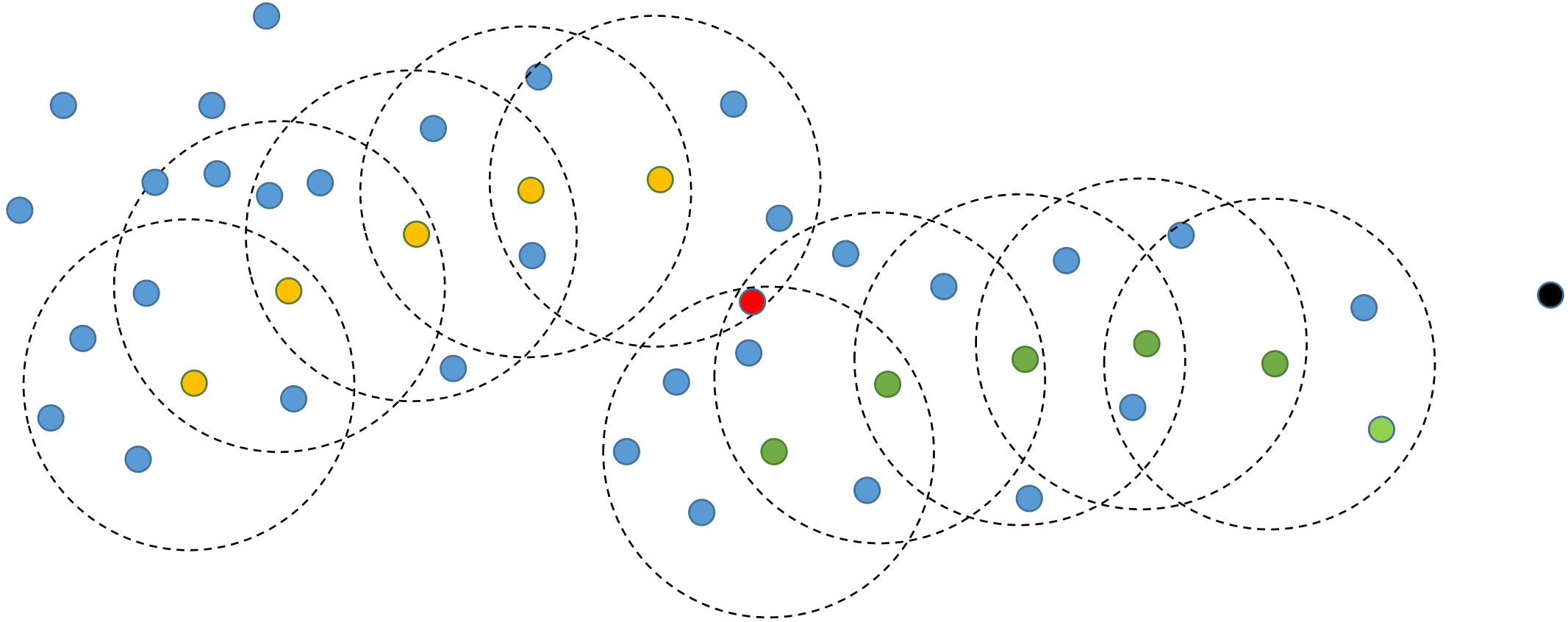
We say that a point **q** is **<u>reachable</u>** from point **p** if there is a series of core points $t_1, t_2, \ldots t_n$ and $t_1 = p, and\ dist(q, t_n) \leq \epsilon$, and also for every *i it holds that:* $dist(t_i, t_{i+1}) \leq \epsilon$



q is reachable from p

DBSCAN *(Density-based spatial clustering of applications with noise)*

A core point, p - form a cluster with all points that are reachable from it. And this cluster will contain all of the points that are reachable from the core points that are reachable from p, and so on..



DBSCAN: Sensitivity to TWO parameters

Cons: sensitive to values of minPts and ϵ.
Does not deal well with clusters of various densities.

minPts=5, ϵ=0.6

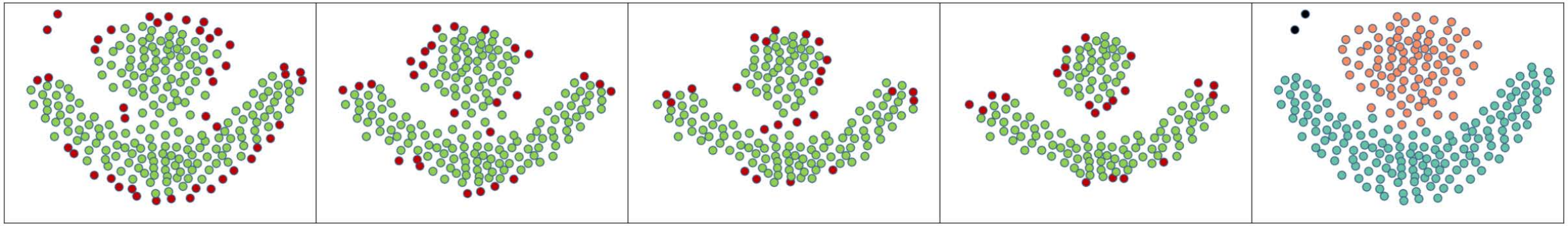
minPts=,5 ϵ=0.5

minPts=15, ϵ=0.7

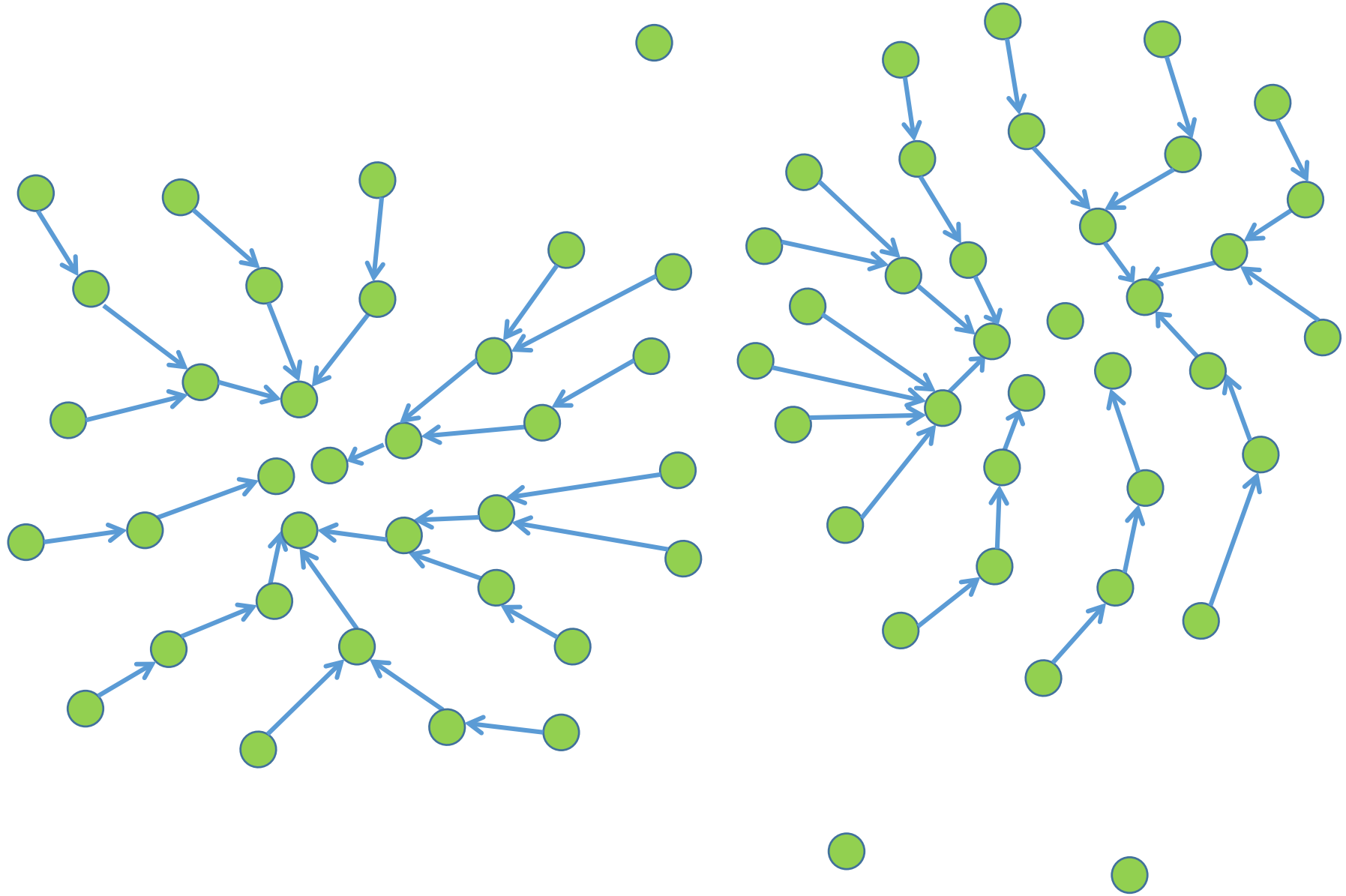
minPts=15, ϵ=0.6

Geometric Reasoning

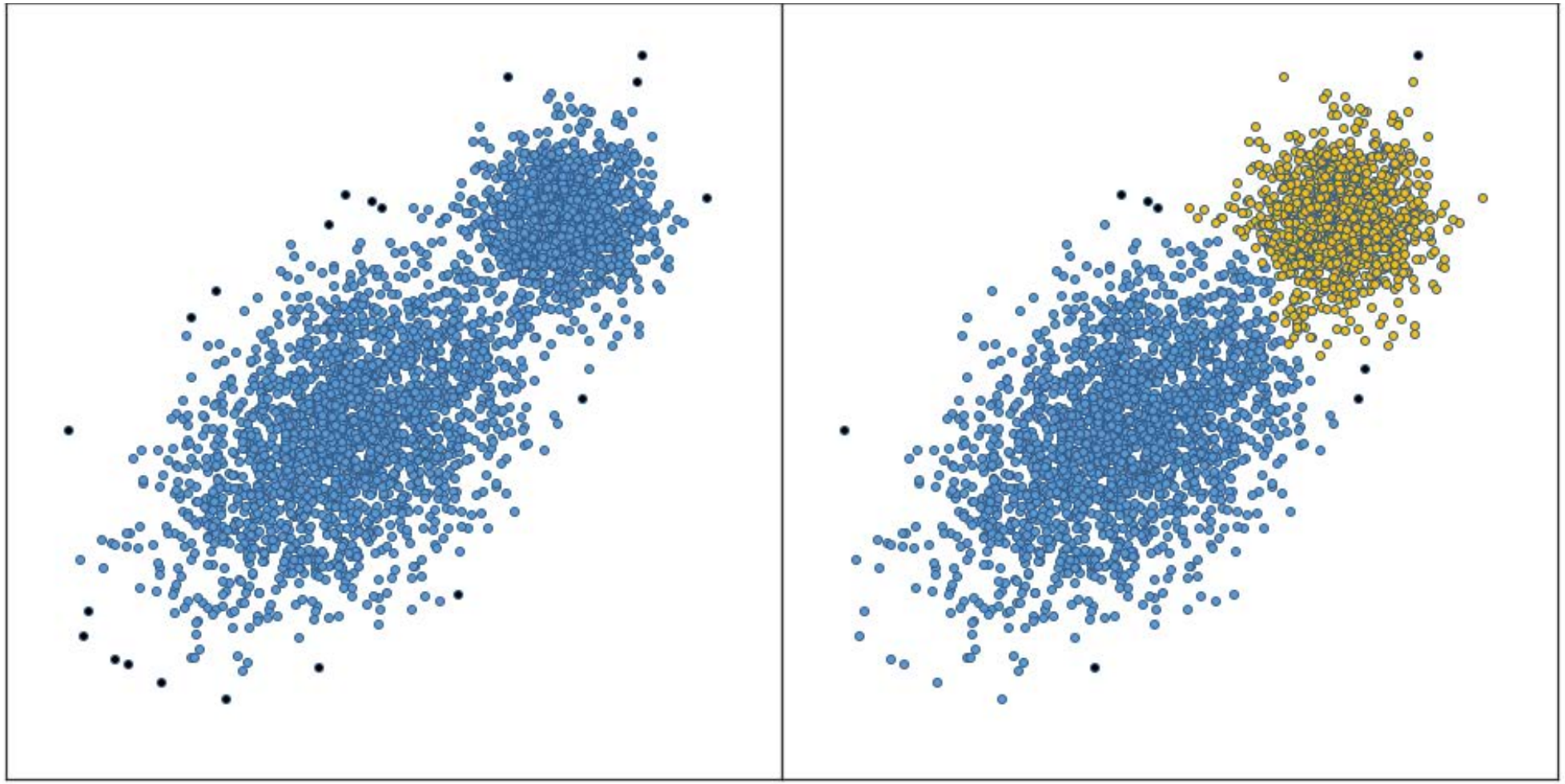
- “Peel” border areas iteratively until only the cluster cores remain
- At each iteration associate border point with neighboring non-border points
- Cluster the more separable core points after the peeling stops



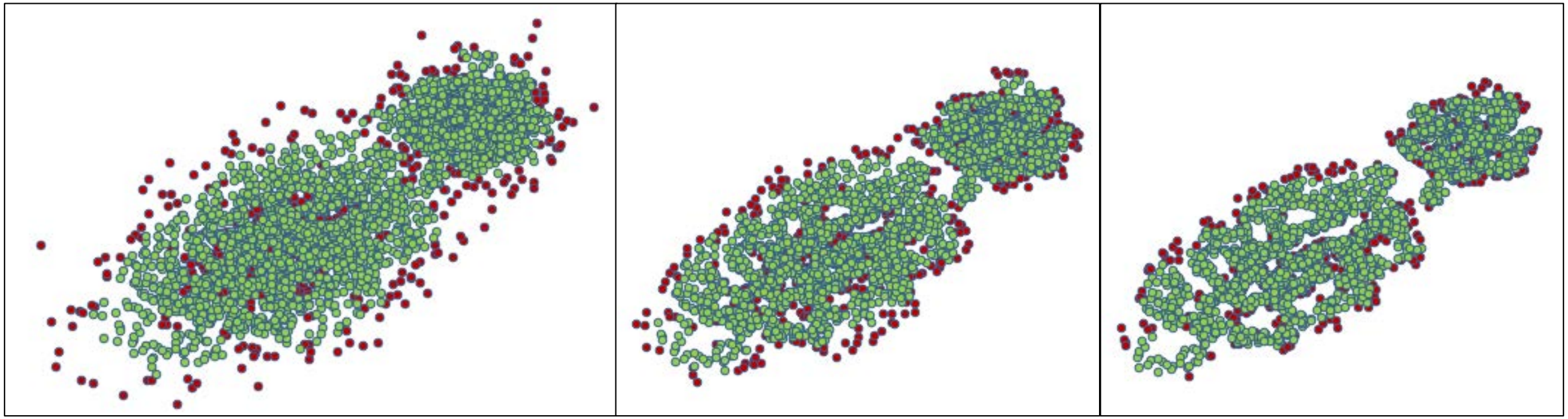
Border peel clustering



Border Peel clustering can discover finer clusters that DBSCAN merges

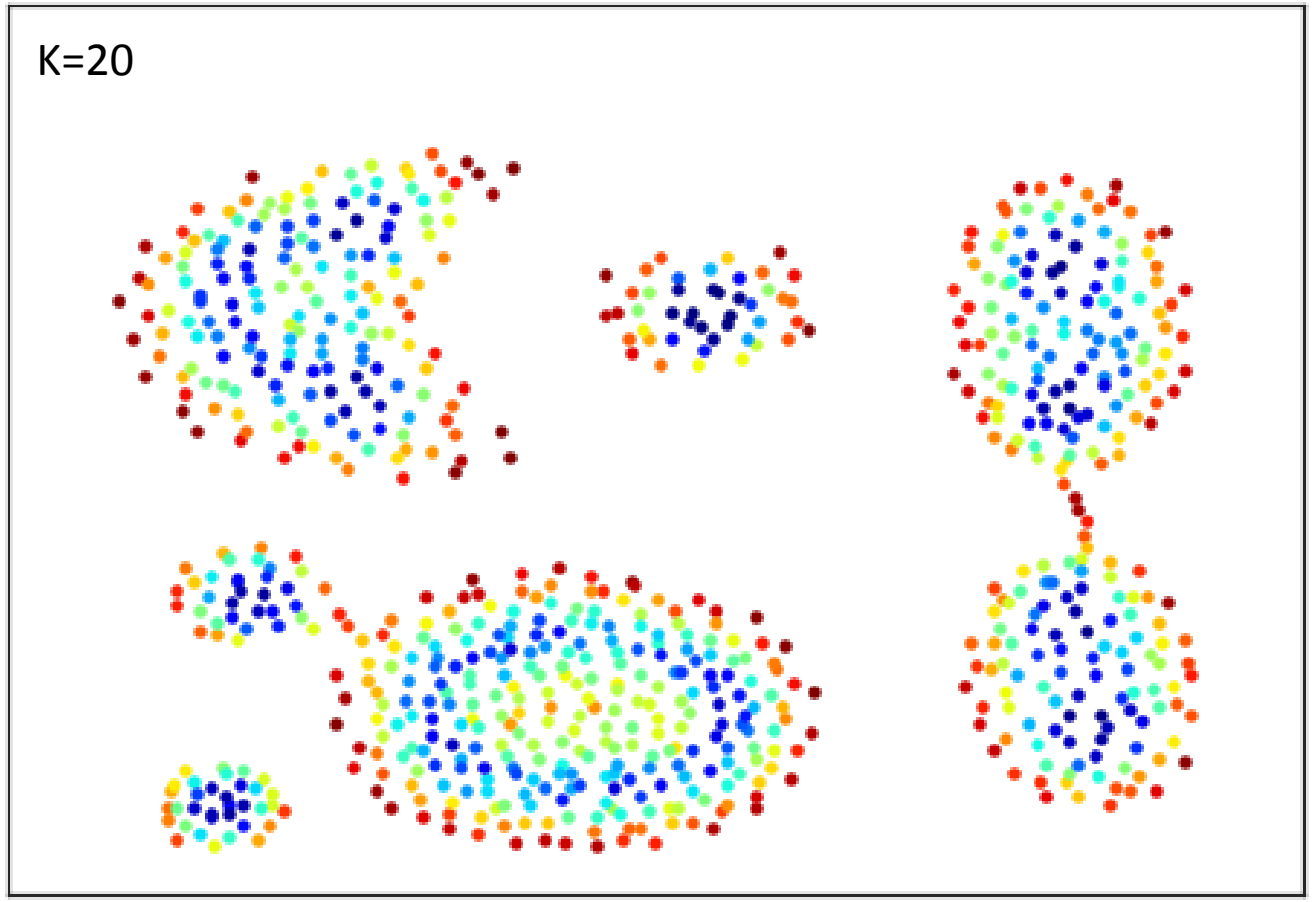


Border Peel clustering can discover finer clusters that DBSCAN merges

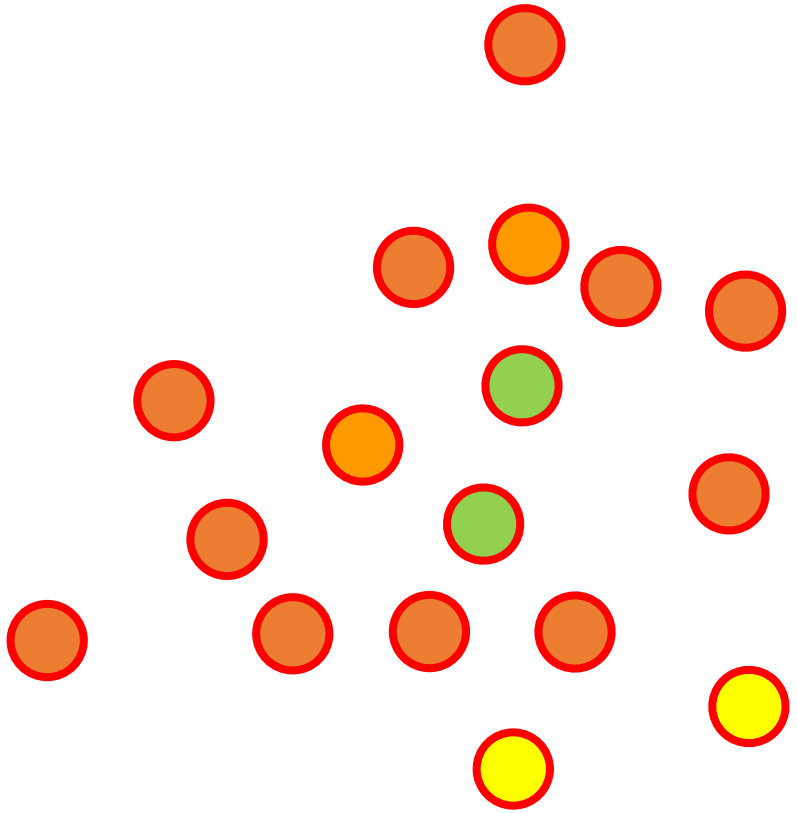


Border points classification

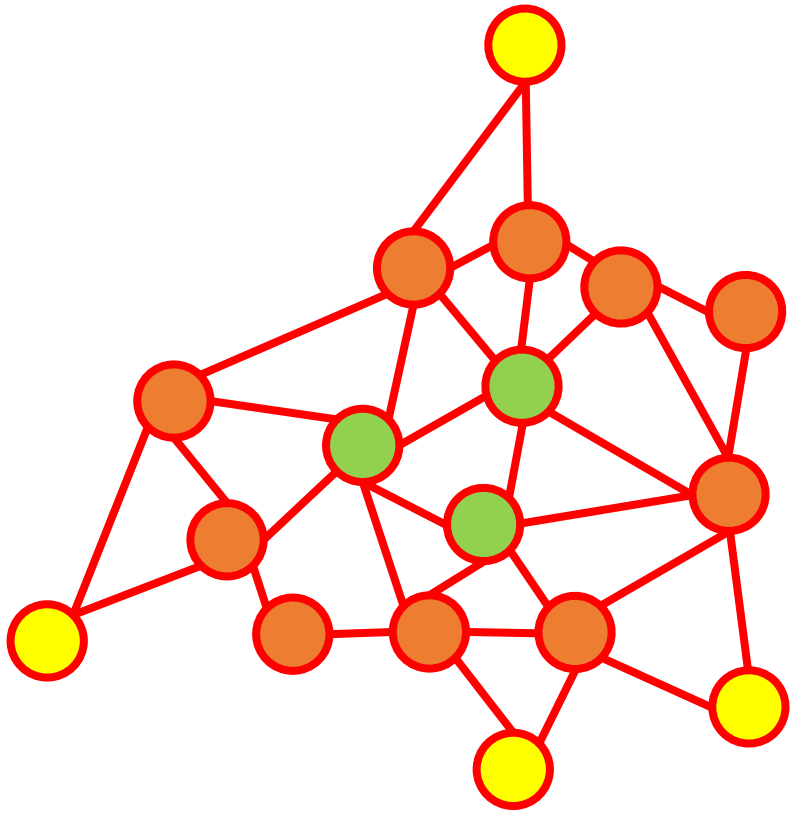
Data points with smaller values of border-ness are classified as “border points”:



How to define border-ness?



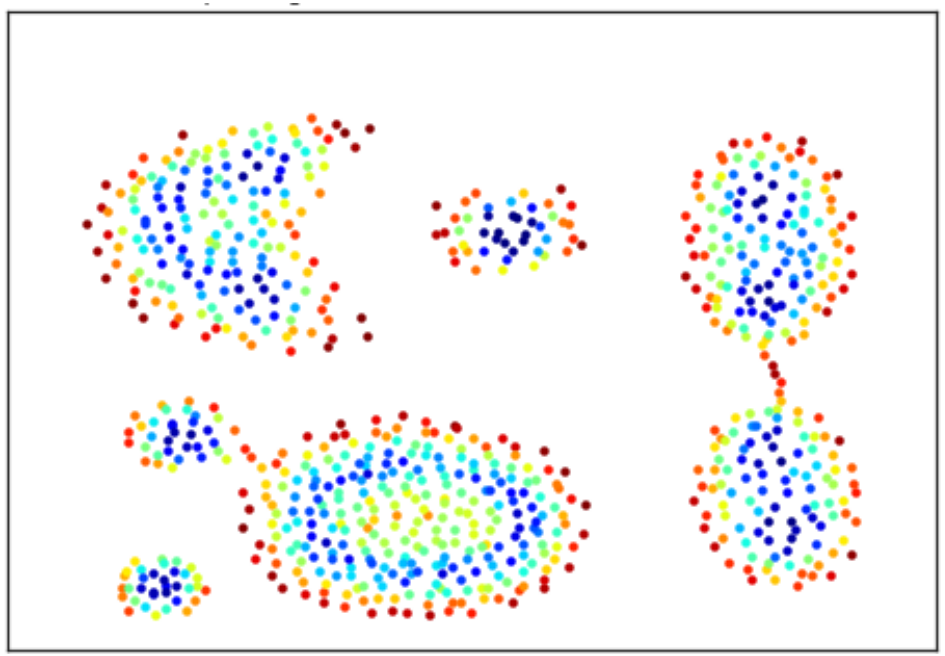
Border points of a cluster tend to have smaller sets of RKNN



Reverse k-nearest neighbor

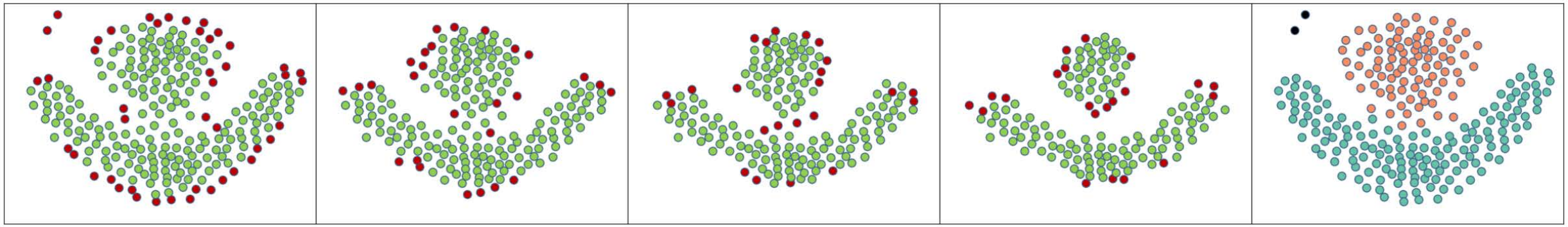
- $KNN(x_i)$ - K-nearest neighbors of x_i: The set of k data points that are closest to data point x_i

- $RKNN(x_i)$ - Reverse KNN of x_i: all of the data points that x_i is one of their k-nearest neighbors.

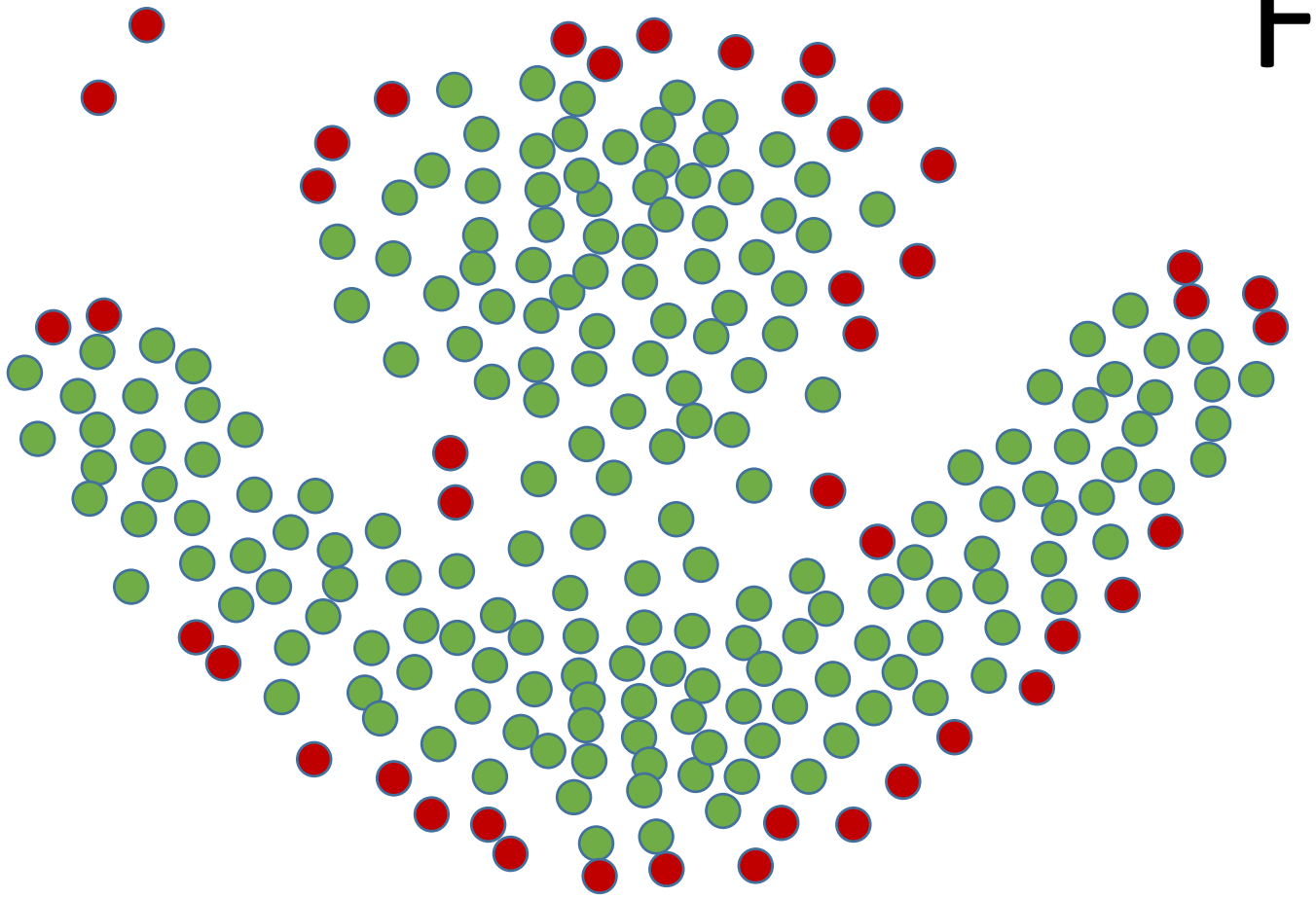


Border Peeling Clustering

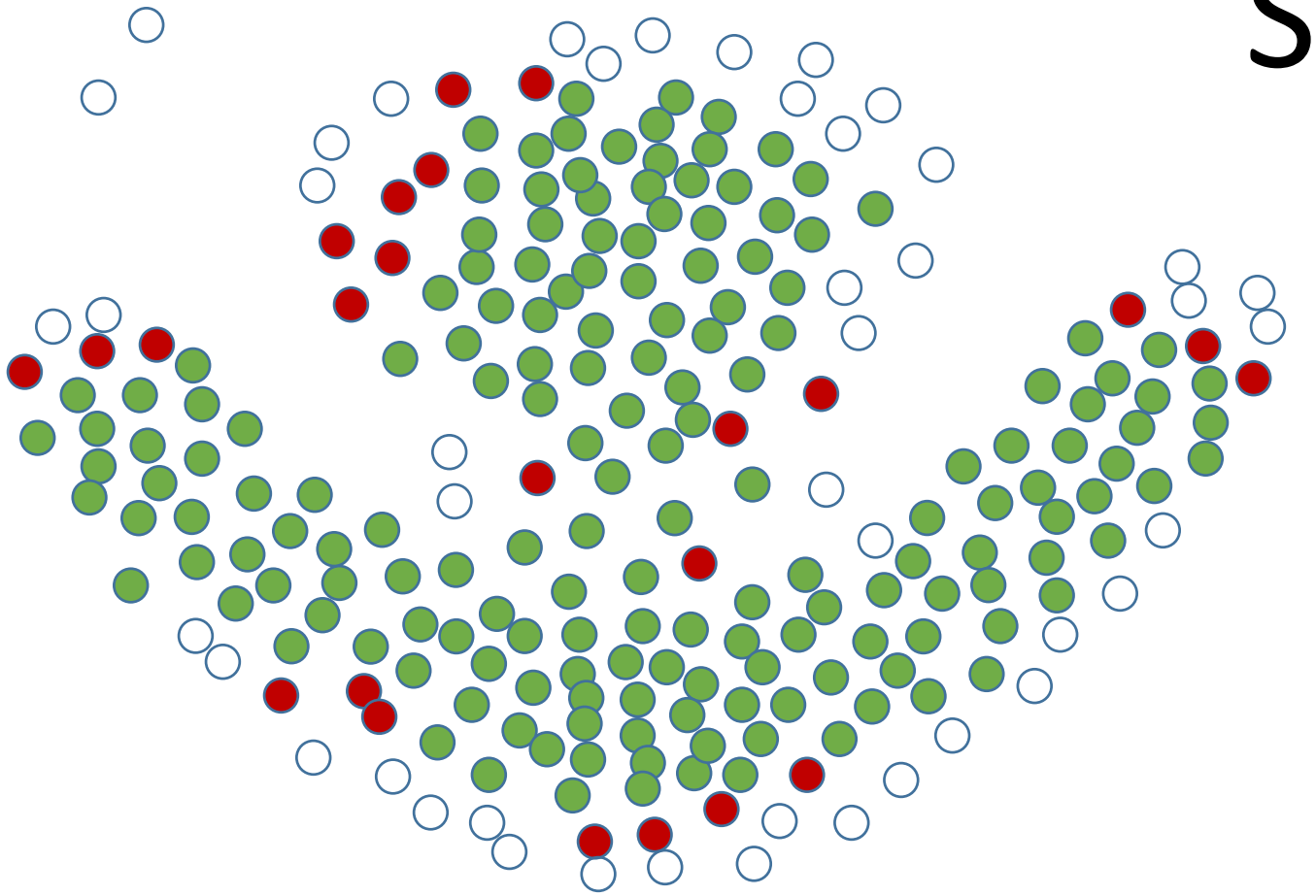
- Iteratively peel border points until the number of border points is small (lower than a predefined threshold)
- The remaining set of points are referred to as the “core points” of the clusters
- The core points are separable and easier to cluster according to their density (similar to DBSCAN, but using a spatially-invariant neighborhood.)
- The peeled border points are then transitively clustered using their association



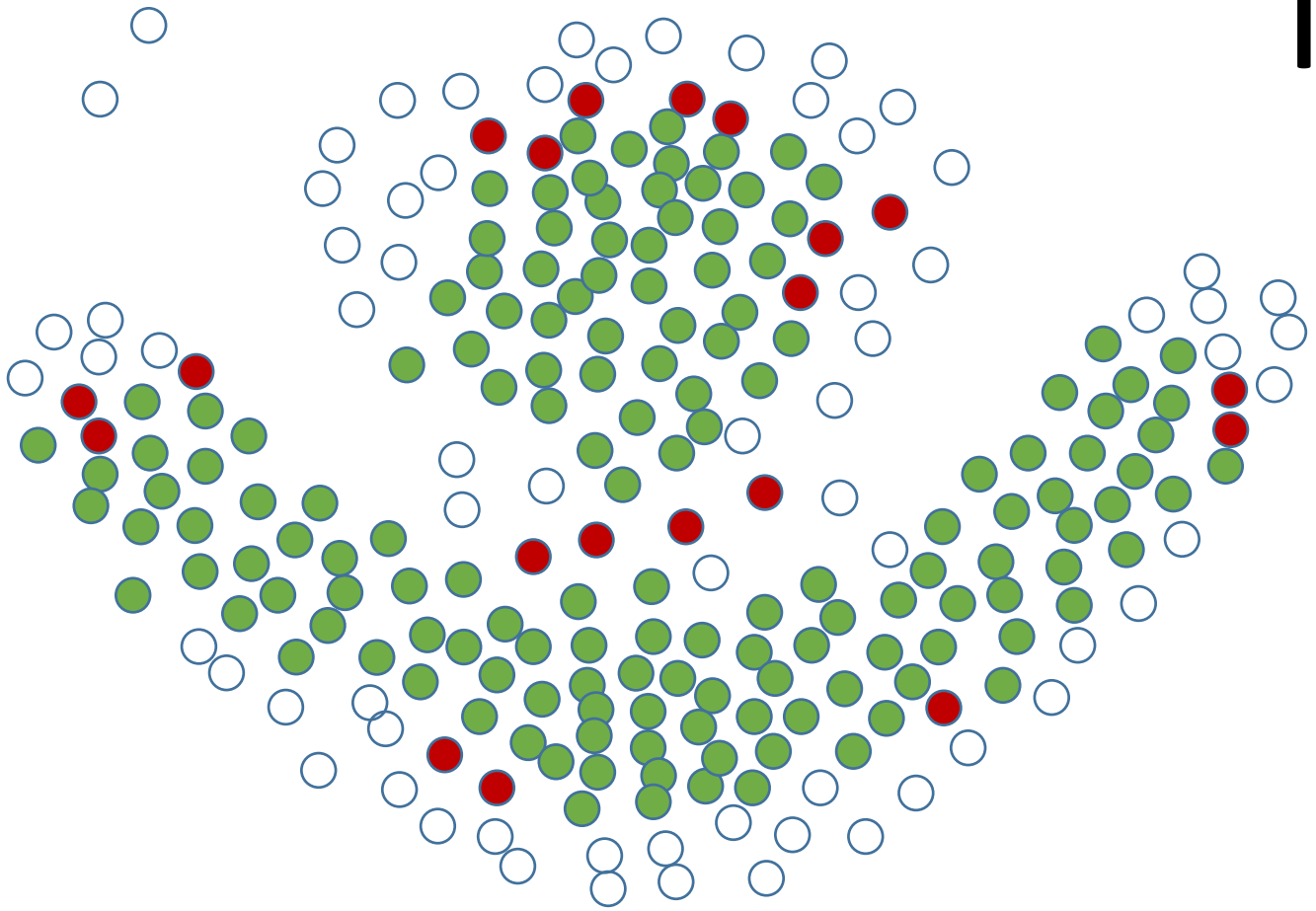
First layer



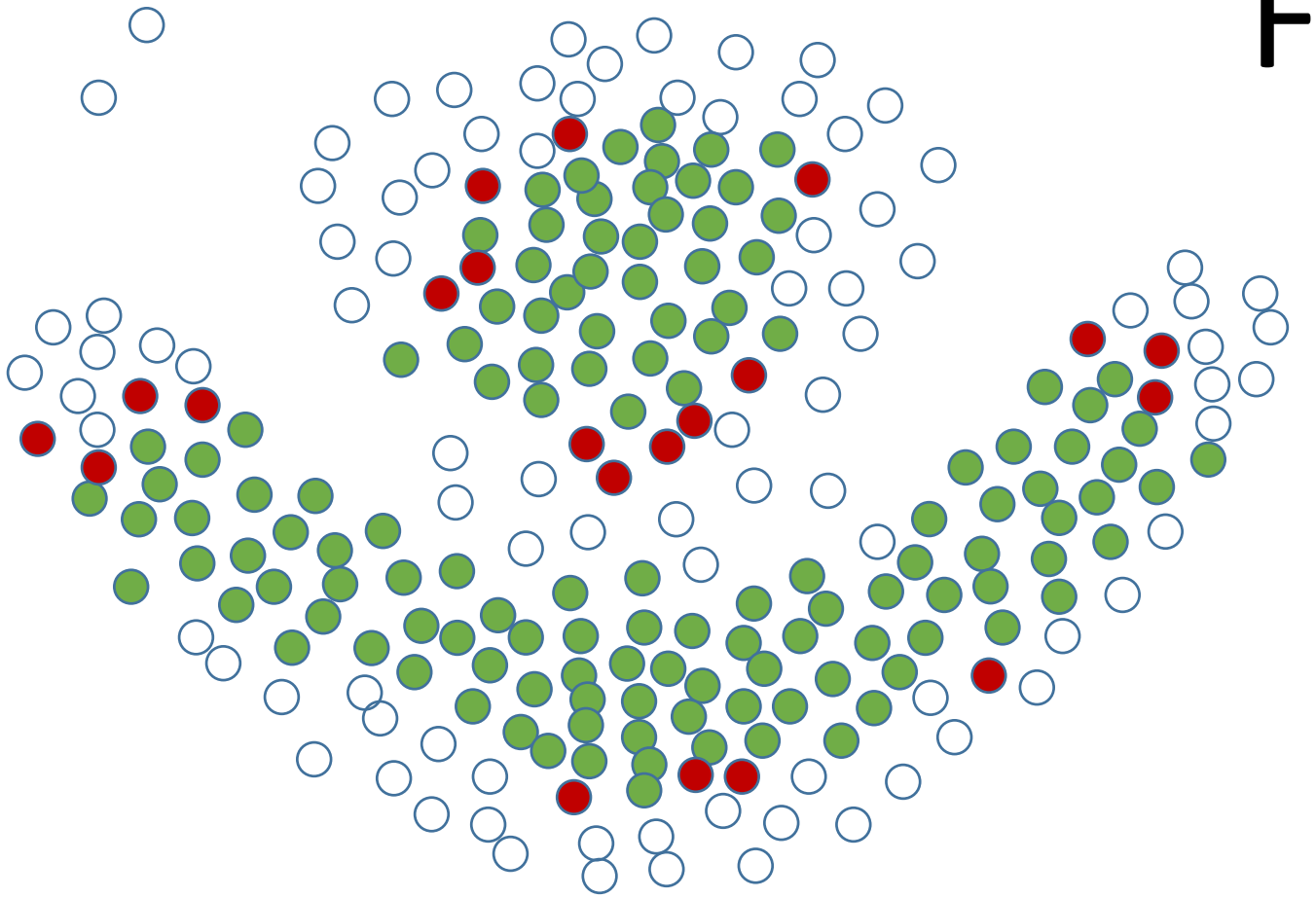
Second layer



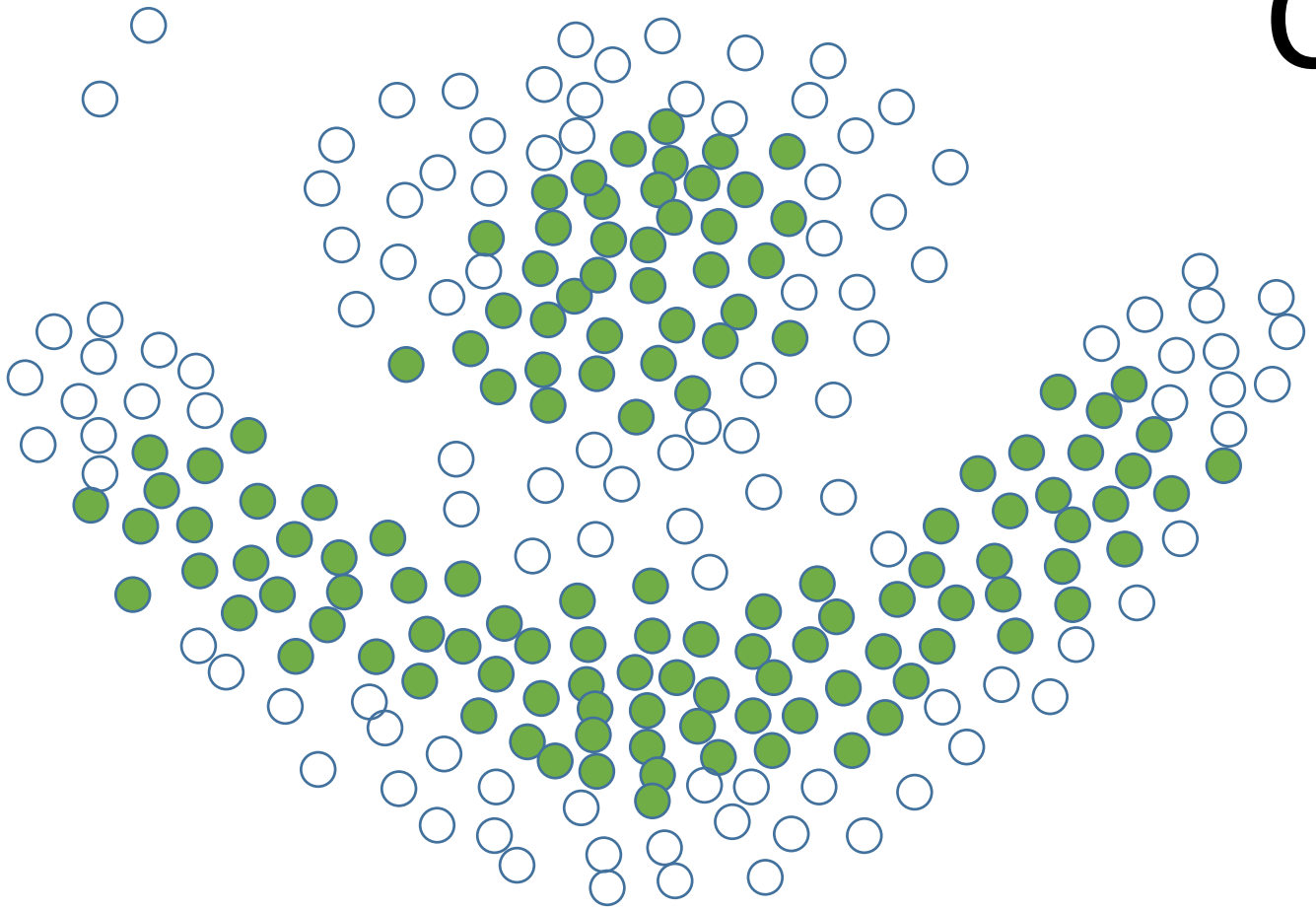
Third layer



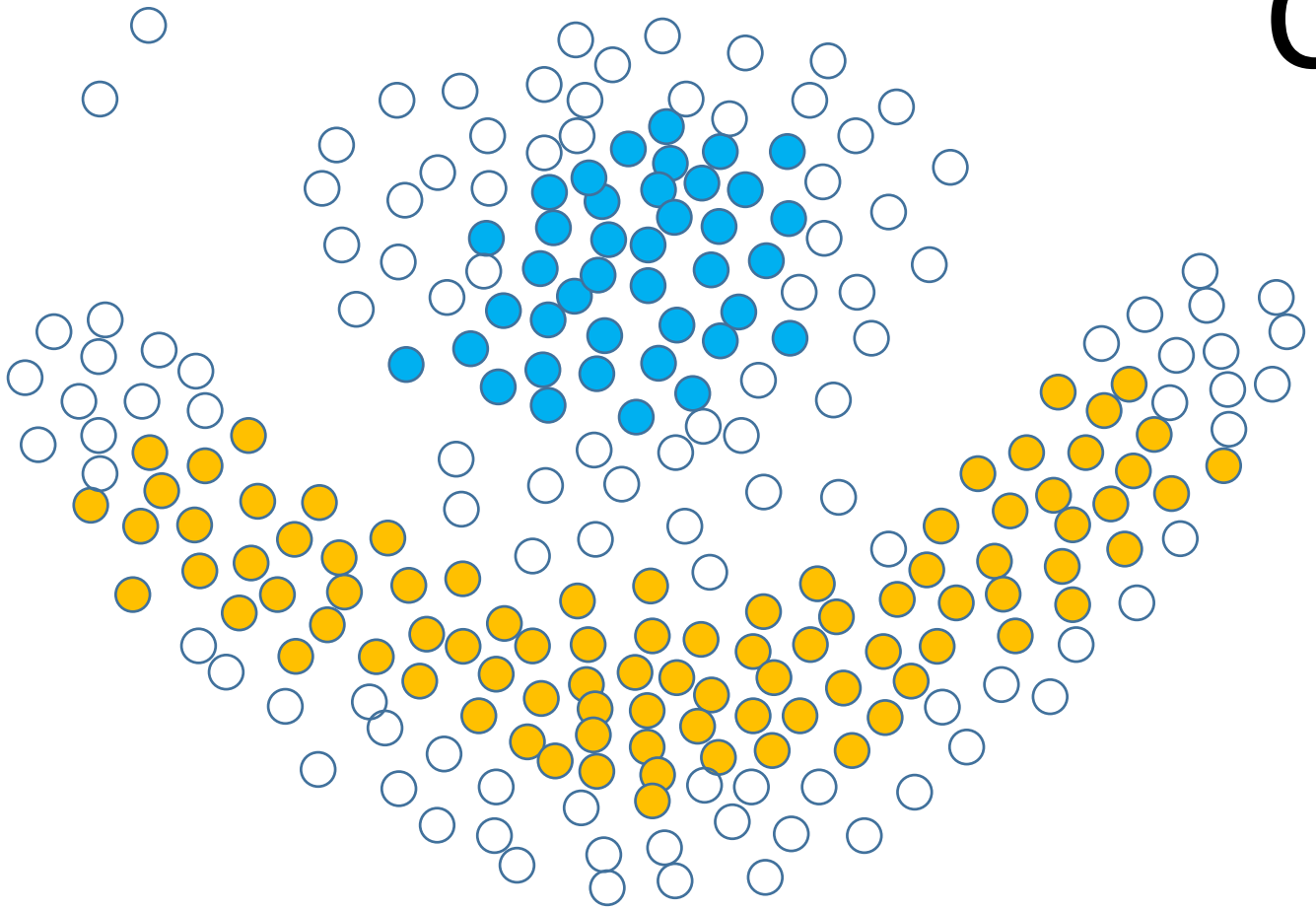
Fourth layer



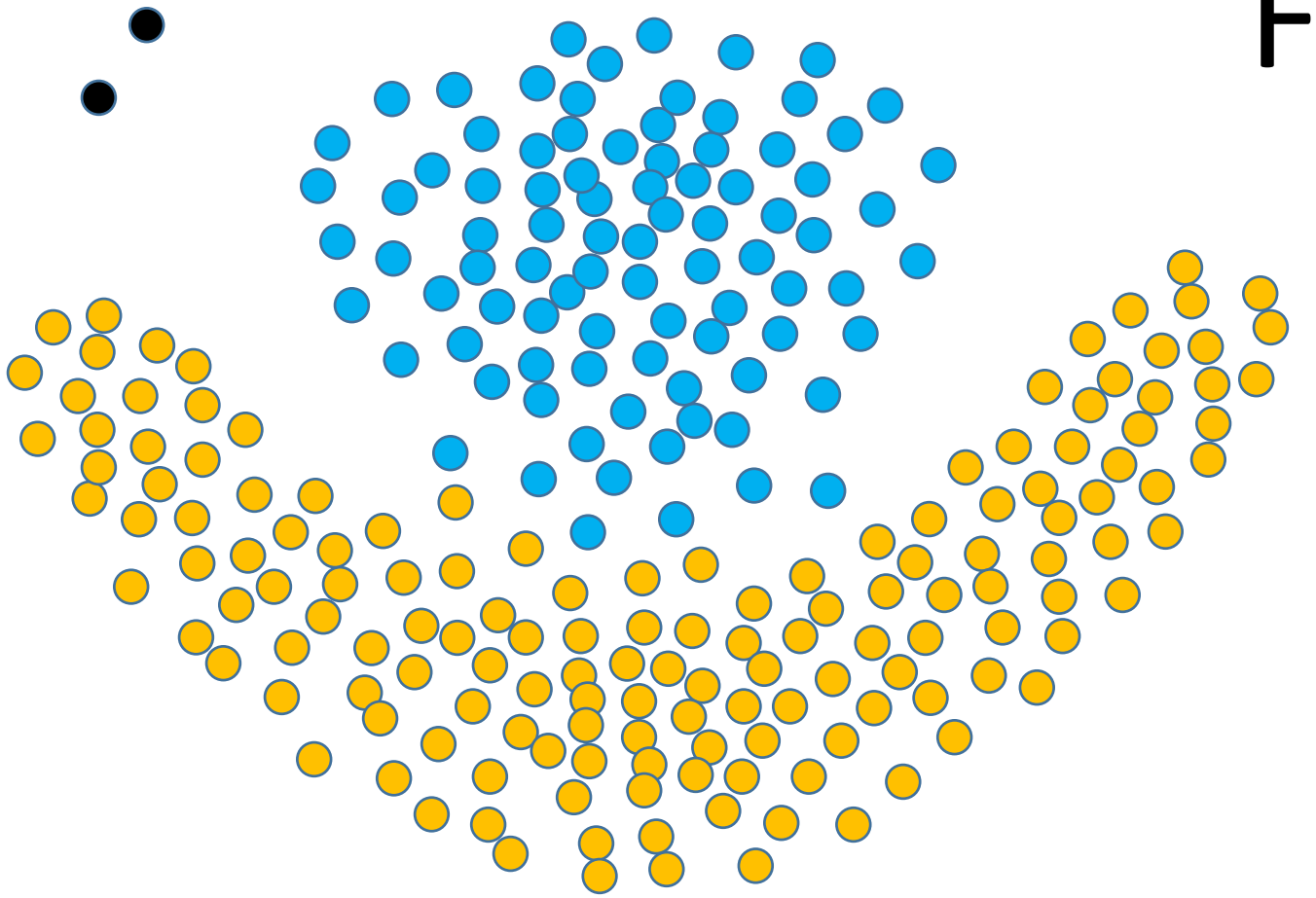
Core points



Cluster

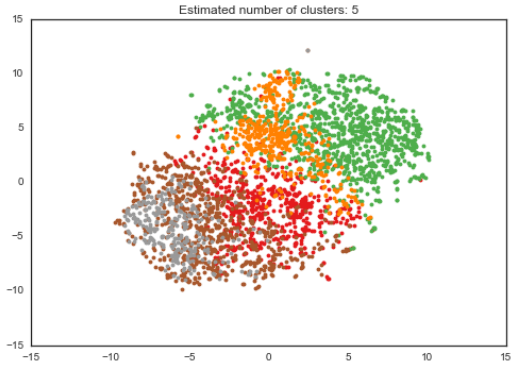


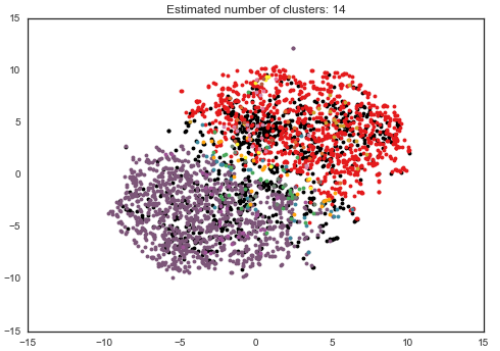
Final Result



Clustering results – (spectral embedding to 5d)

Method	Number of Clusters detected	Normalized Mutual Information	Adjusted Mutual Information	Adjusted Rand Index	Rand Index
Spectral Clustering	5	0.406	0.400	0.254	0.687
DBSCAN	15	0.023	0.006	-0.009	0.362
K-Means (original data)	5	0.356	0.329	0.242	0.711
Border Peel	4	0.425	0.424	0.428	0.769
HDBSCAN	2	0.007	0.004	0.009	0.498
Affinity Propagation	71	0.337	0.183	0.042	0.710
K-Means	5	0.406	0.377	0.323	0.740





Run command for the border peel:
BorderPeel.BorderPeel(iterations=40, k=20, plot_debug_output_dir = debug_output_dir, min_cluster_size = 10, dist_threshold = 0.43, convergence_constant = 0, link_dist_expansion_factor = 4, verbose = True, border_precentile = 0.1, stopping_precentile=0.05)