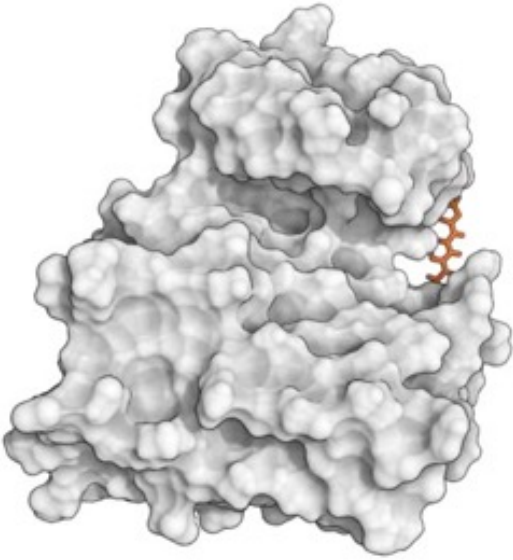


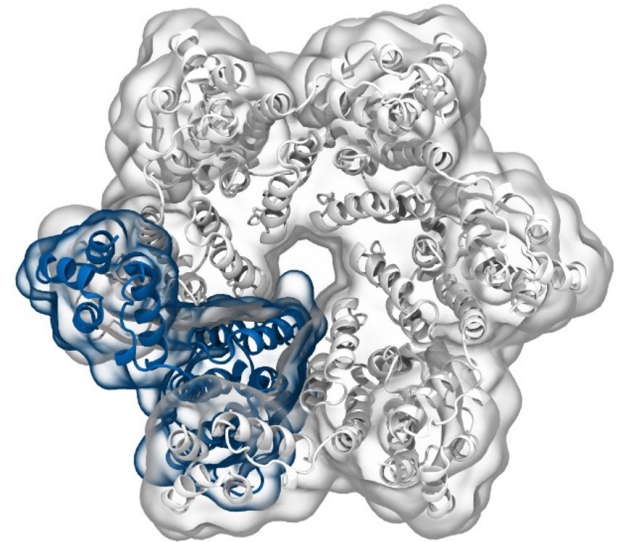
Simulation of Biomolecules

Clustering



Dr Matteo Degiacomi
Durham University

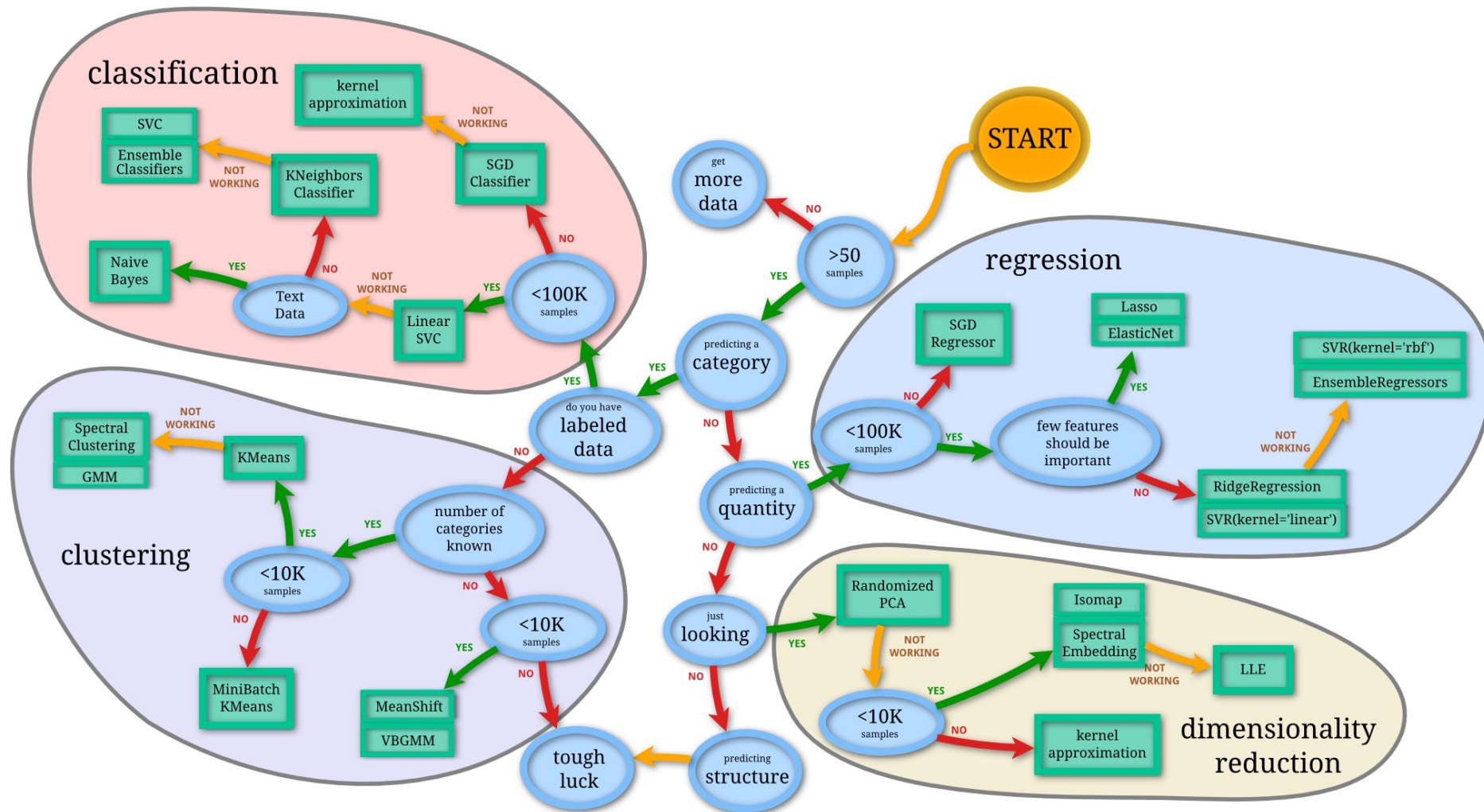
matteo.t.degiacomini@durham.ac.uk



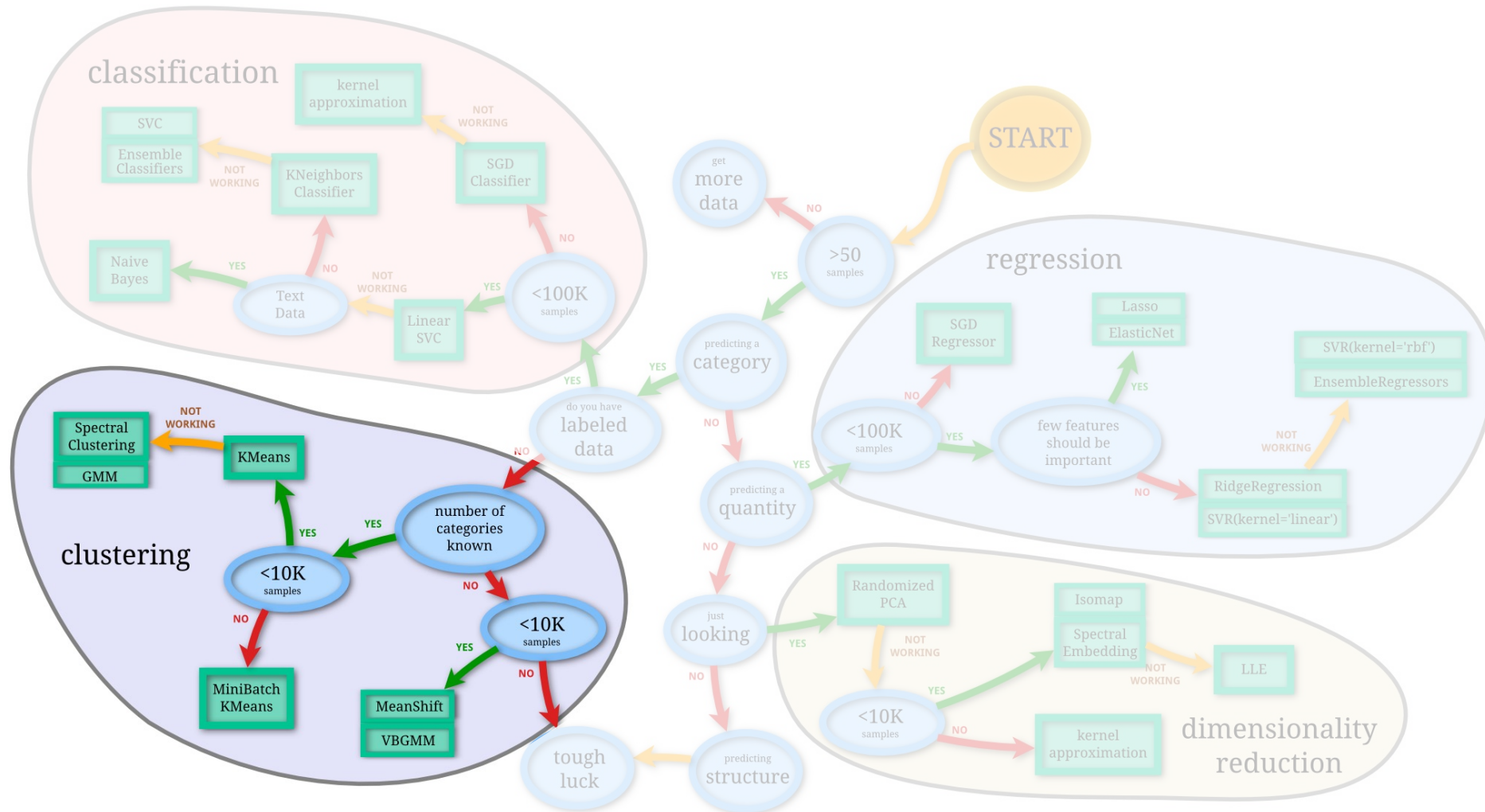
Dr Antonia Mey
University of Edinburgh

antonia.mey@ed.ac.uk

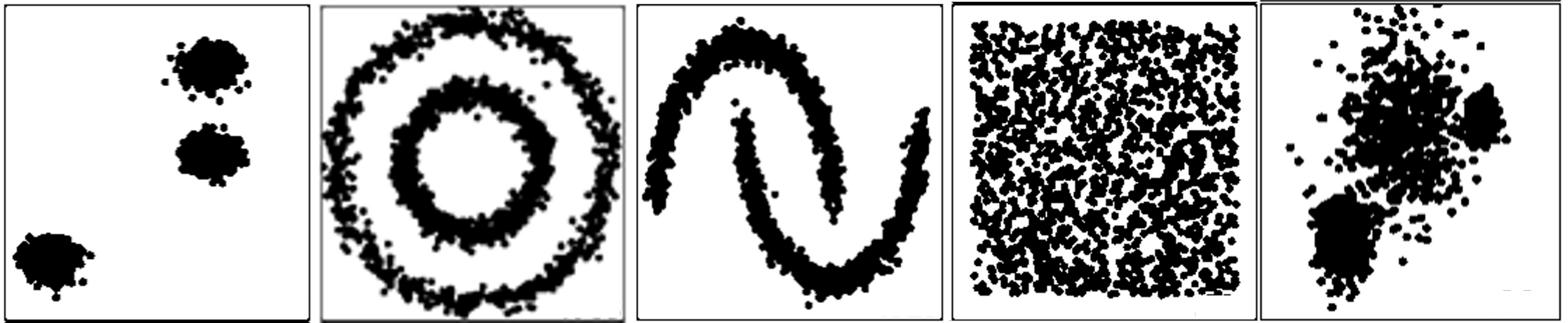
The Data Mining world



The Data Mining world

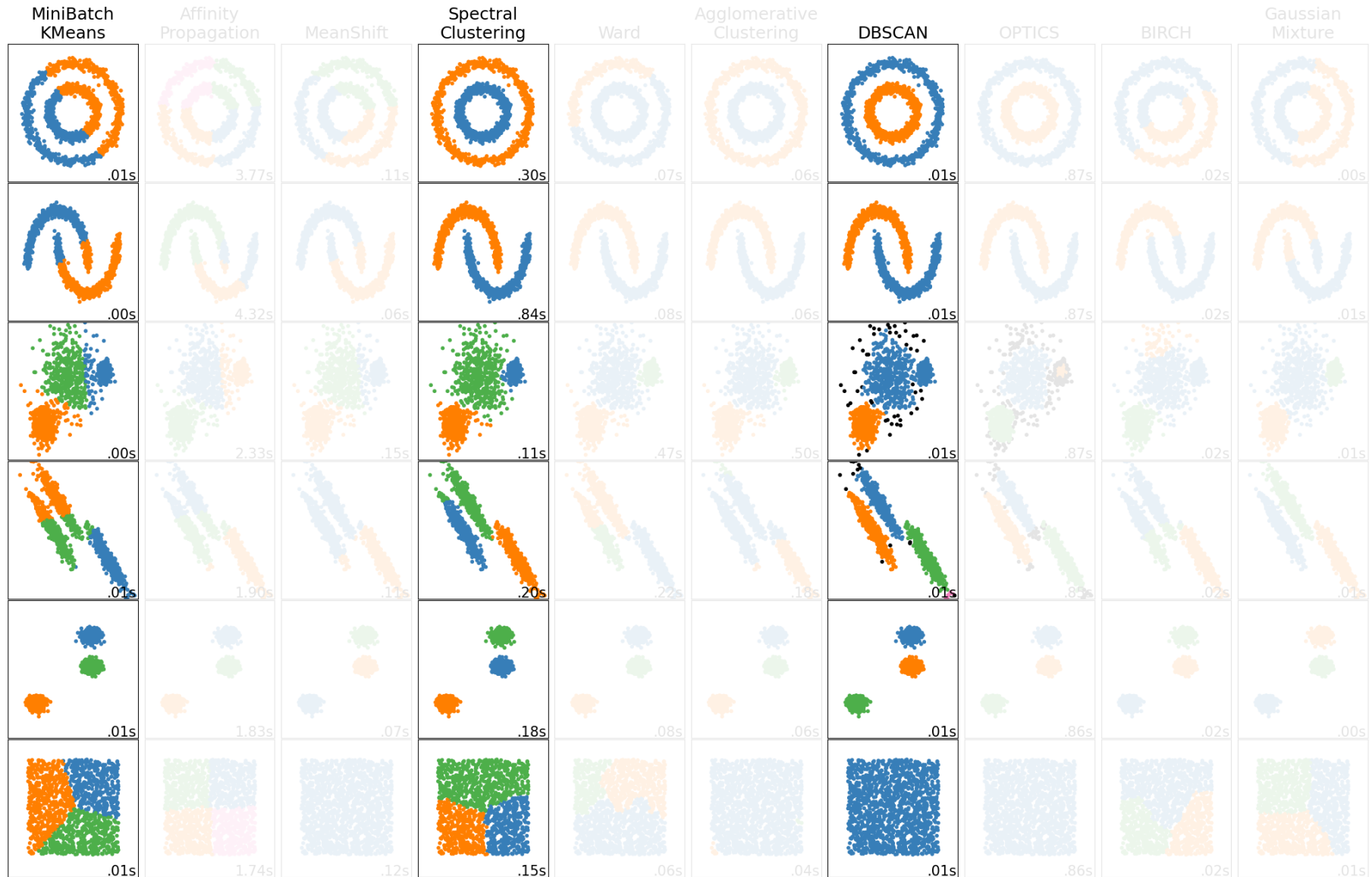


Clustering (i.e., unsupervised learning)



Known number of clusters? Flat geometry?
Even cluster size? Outliers? Centroids needed?

Clustering algorithms



How does k-means work?

Input: K, set of points $x_1 \dots x_n$ (can be in N-dimensional)

Place centroids, $c_1 \dots, c_n$ at random locations

Repeat until convergence:

For each point x_i :

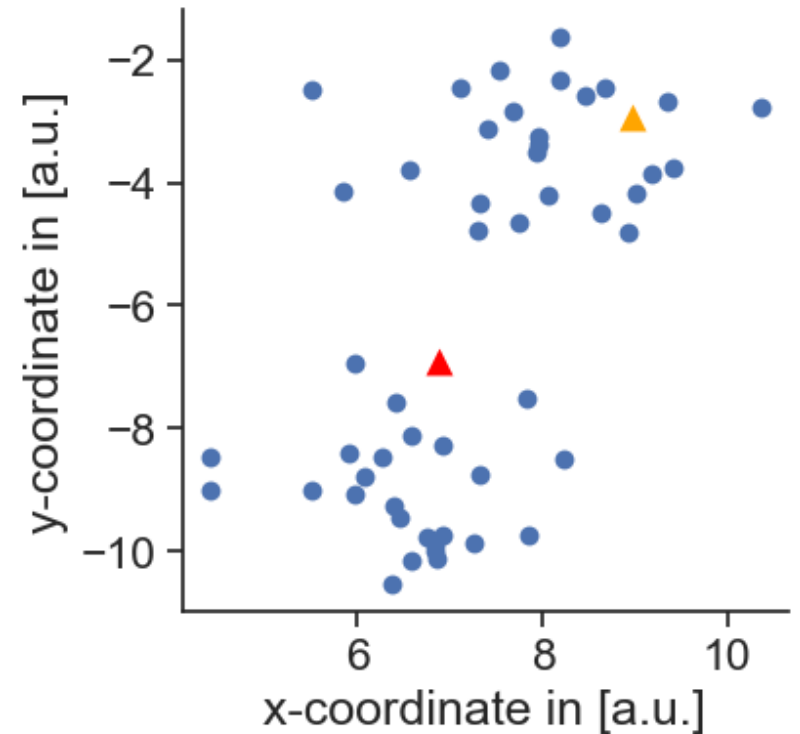
Find nearest centroid $c_j = \arg \min_j D(x_i, c_j)$

Assign the point x_i to cluster j

For each cluster $j = 1 \dots K$:

Compute the centroid mean for all points in one cluster and update the centroid

$$c_j(a) = \frac{1}{n_j} \sum_{x_i \rightarrow c_j} x_i(a)$$



How does k-means work?

Input: K, set of points $x_1 \dots x_n$ (can be in N-dimensional)

Place centroids, $c_1 \dots, c_n$ at random locations

Repeat until convergence:

For each point x_i :

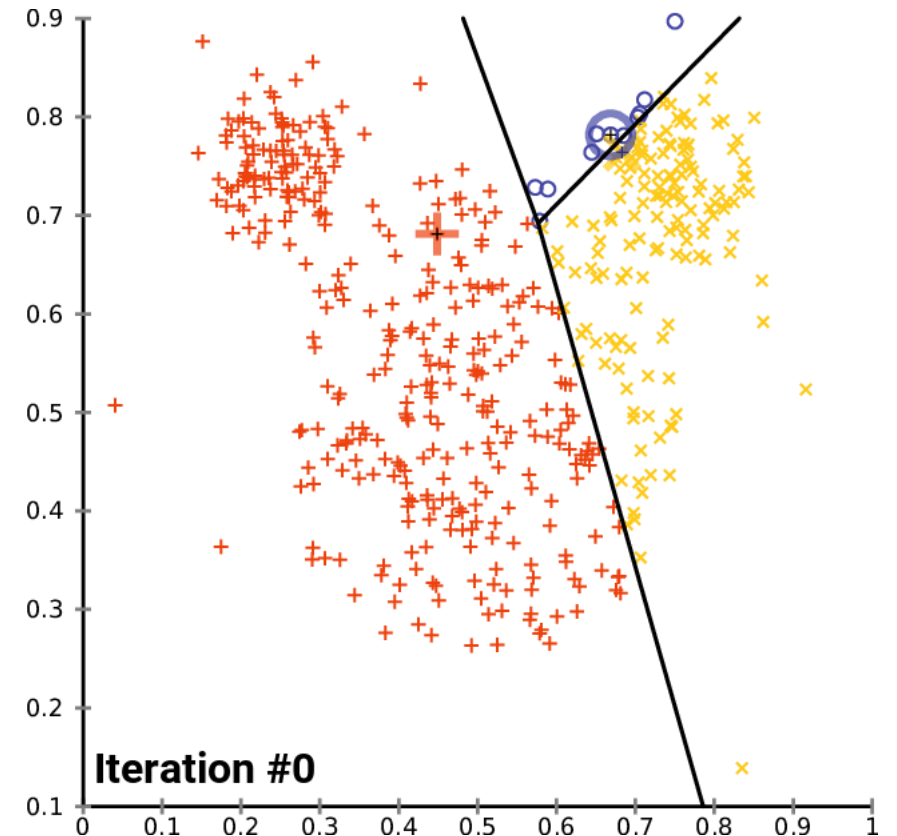
Find nearest centroid $c_j = \arg \min_j D(x_i, c_j)$

Assign the point x_i to cluster j

For each cluster $j = 1 \dots K$:

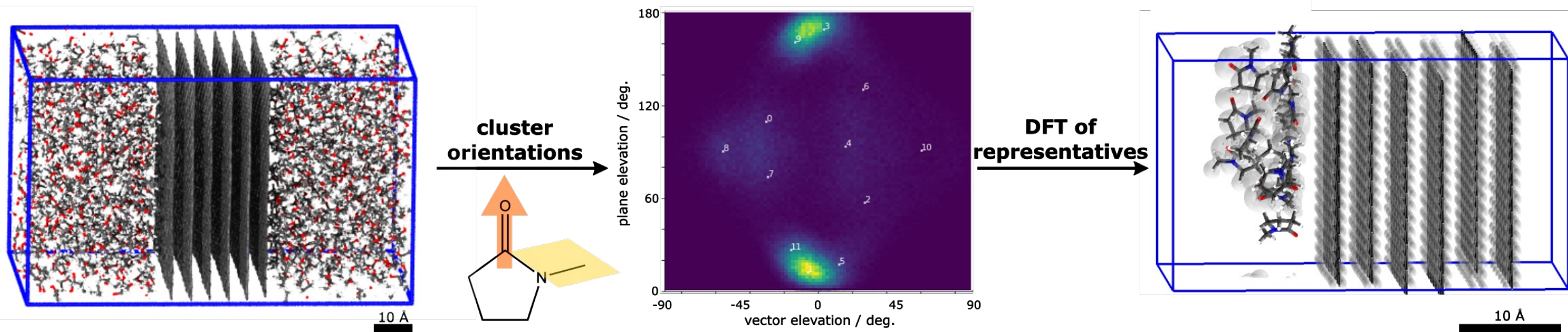
Compute the centroid mean for all points in one cluster and update the centroid

$$c_j(a) = \frac{1}{n_j} \sum_{x_i \rightarrow c_j} x_i(a)$$



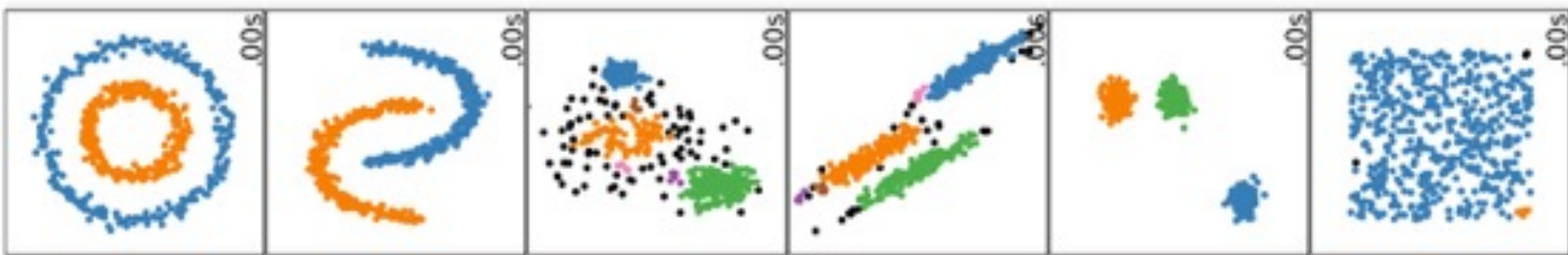
[Example] k-means vs solvent-graphite interactions

- Molecular Dynamics simulation of graphite immersed in solvents.
- >100k individual solvent-graphene interactions

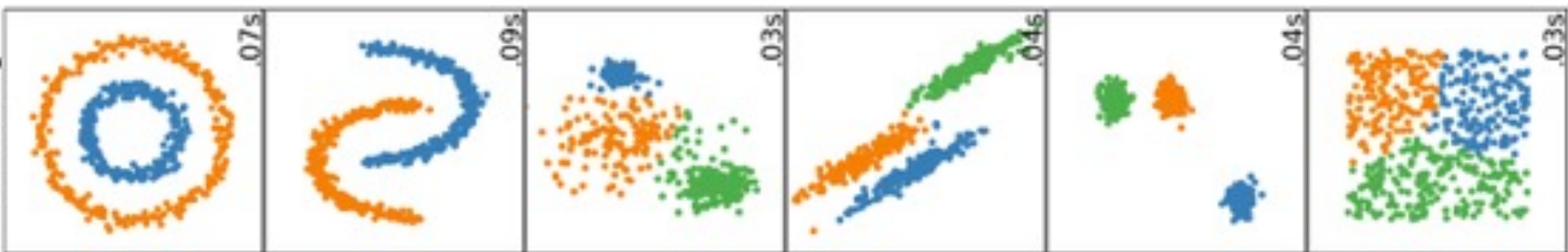


Density-based and spectral clustering

DBSCAN

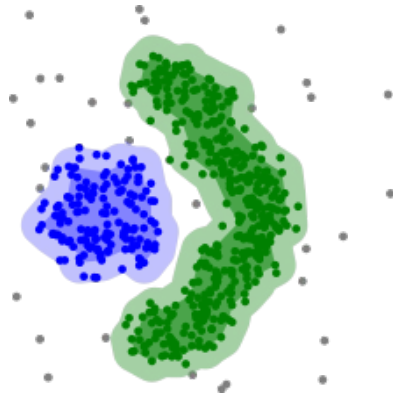


Spectral Clustering



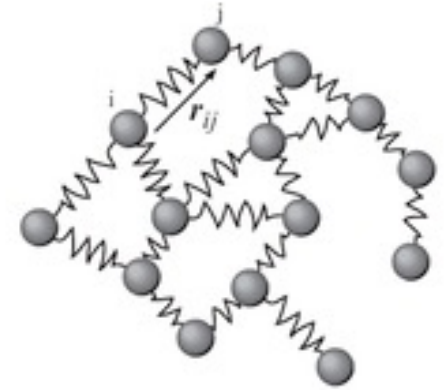
Density-based and spectral clustering

DBSCAN



- Find the points in the ε neighbourhood of every point, identify core points with more than n neighbours.
- Find the connected components of core points on the neighbour graph, ignoring all non-core points.
- Assign each non-core point to a nearby cluster if the cluster is an ε neighbour, otherwise assign to noise otherwise.

Spectral Clustering



- Calculate the Laplacian
- Calculate the first k eigenvectors
- Consider the matrix formed by the first k -eigenvectors
- Cluster the graph nodes based on these features (e.g. k-means)

[Example] DBSCAN for noise detection in EM maps

- Load electron density map and chose a threshold t
- Place pseudoatoms where intensity $> t$
- Cluster beads and delete small clusters ($<1\%$ of total beads)

