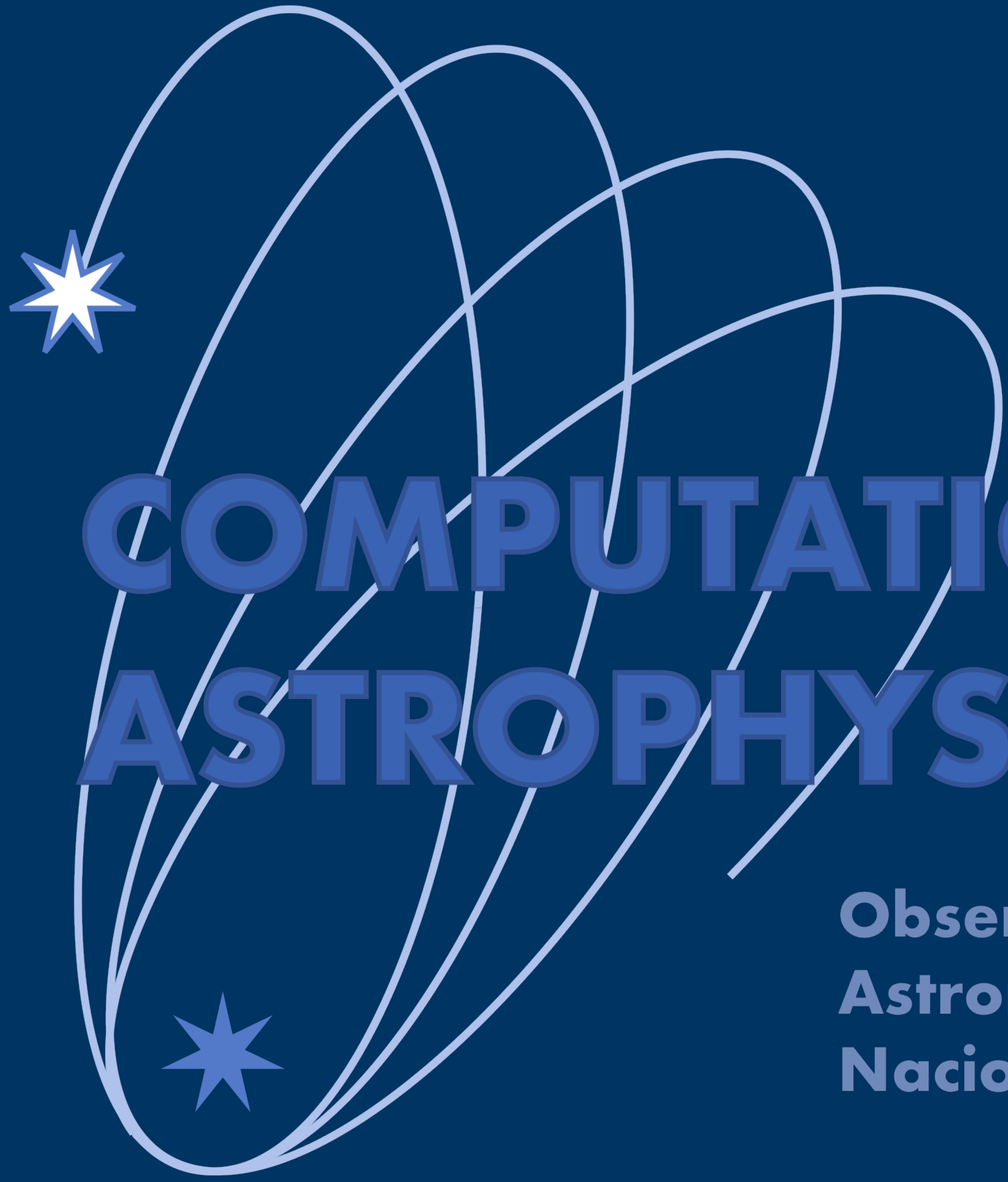


COMPUTATIONAL ASTROPHYSICS

Observatorio
Astronómico
Nacional

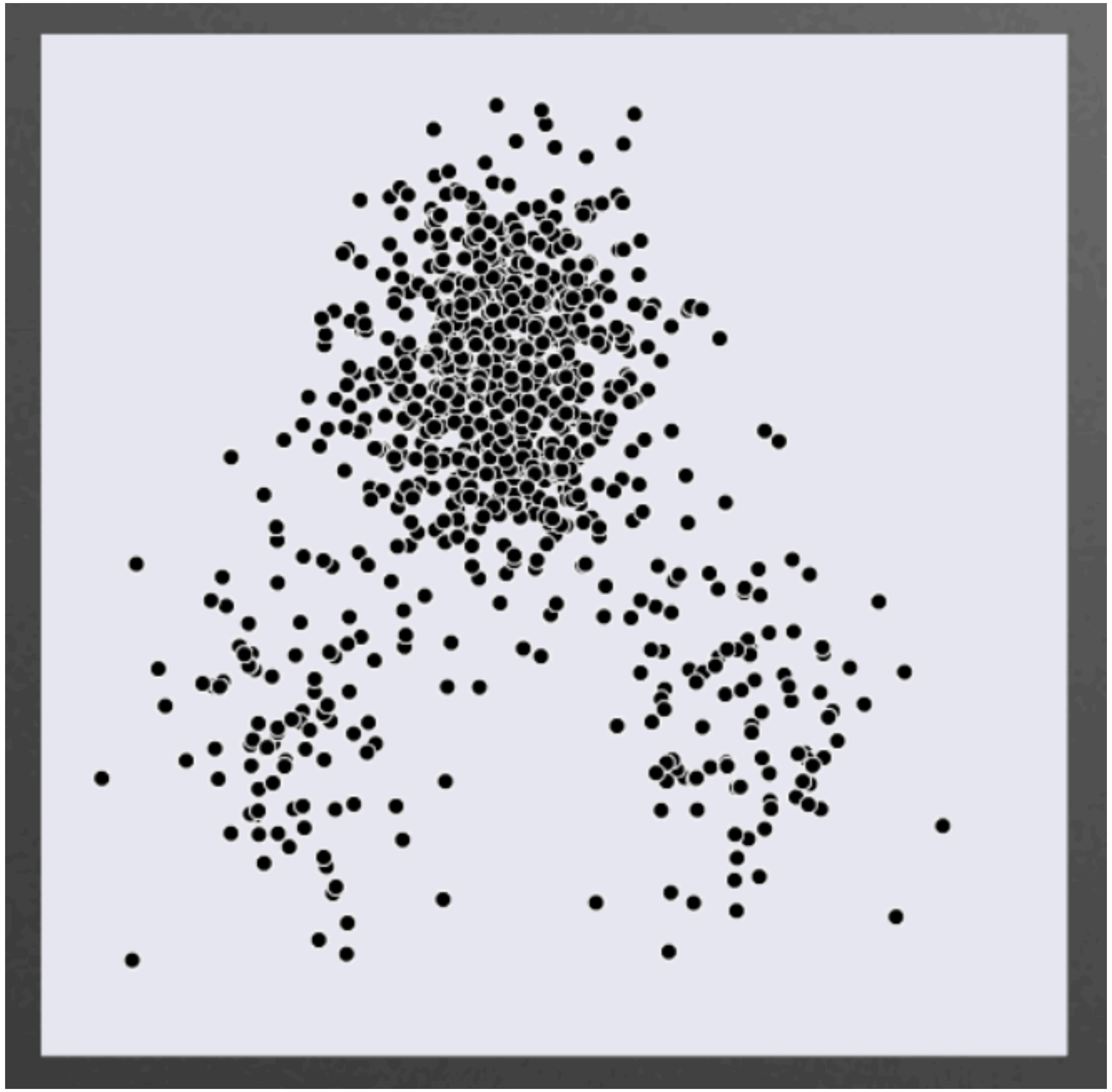
Computational Astrophysics

10. Clustering

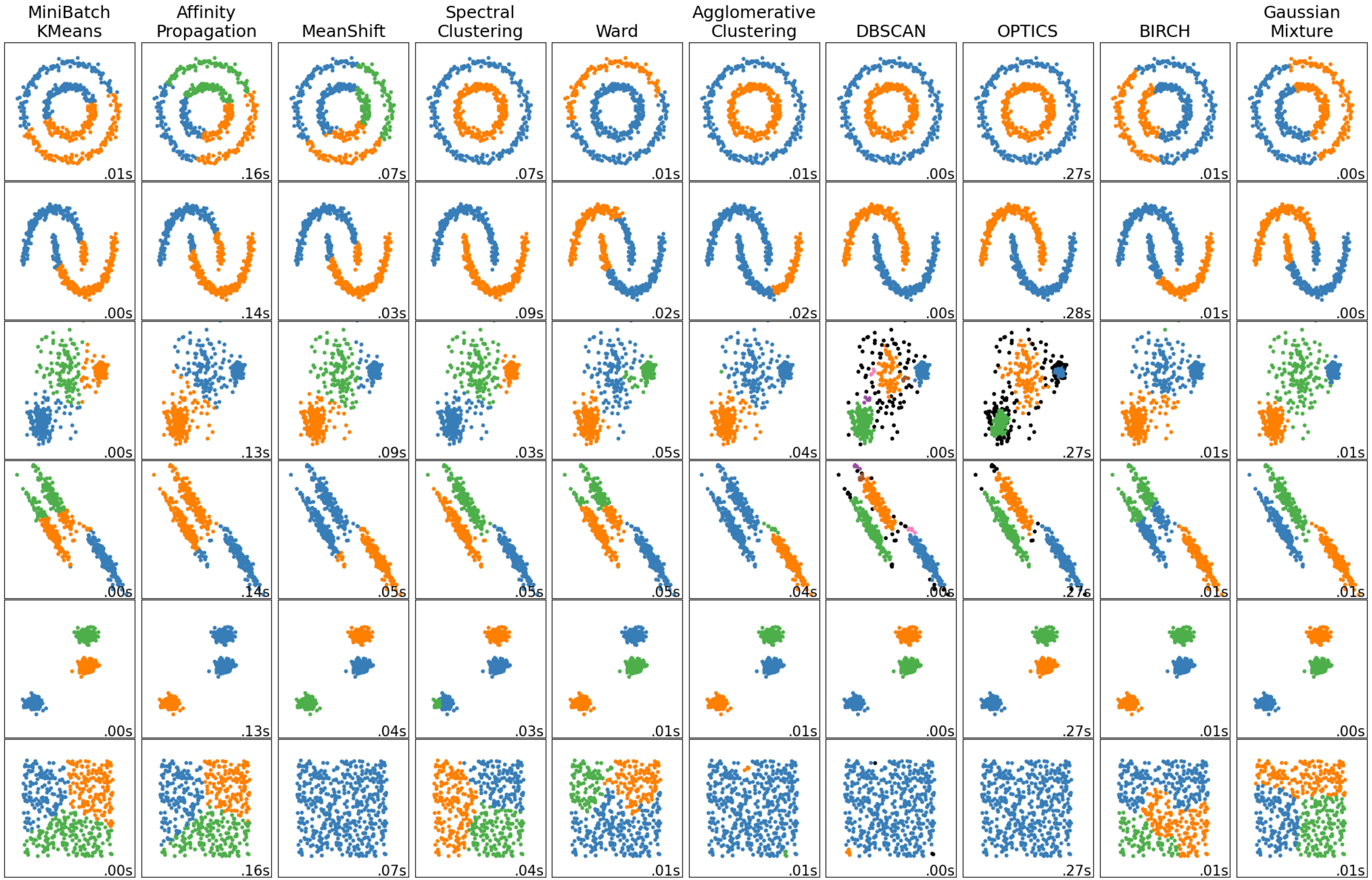
Eduard Larrañaga
Observatorio Astronómico Nacional
Universidad Nacional de Colombia

Clustering
(Unsupervised Classification)

Data with no-labels



Clustering in Sci-kit learn



K-Means

K-Means

The **KMeans** algorithm clusters data by trying to separate samples in groups of equal variance, minimizing a criterion known as the *inertia* or within-cluster sum-of-squares.

This algorithm requires the number of clusters to be specified.

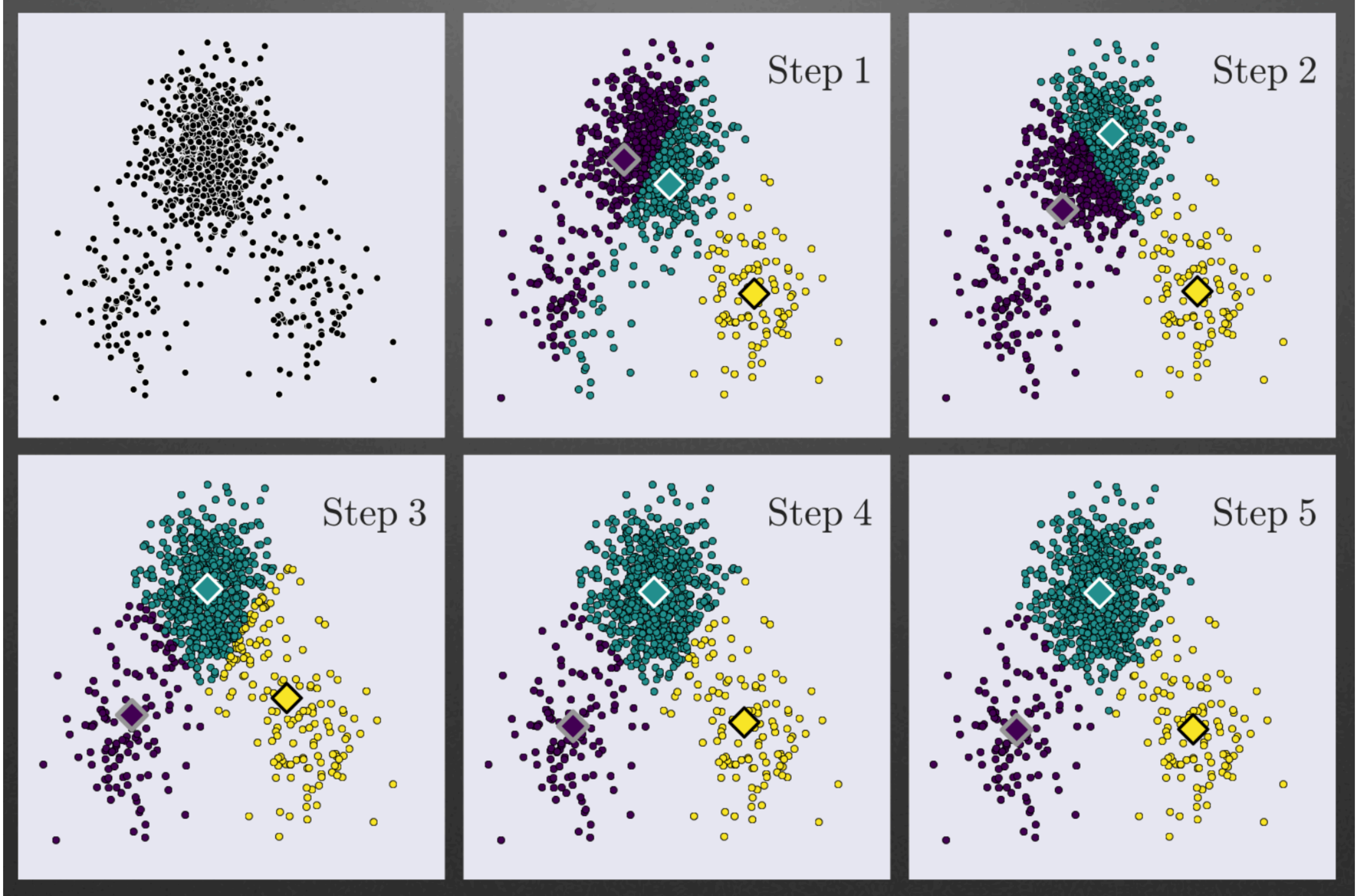
The k-means algorithm divides a set of N samples $X = \{x_1, x_2, \ldots, x_N\}$ into K disjoint clusters $C = \{c_1, c_2, \ldots, c_K\}$, each described by the mean μ_j ($j = 1,2,...,K$) of the samples in the cluster. The means are commonly called the cluster *"centroids", but* note that they are not, in general, points from X, although they live in the same space.

K-Means

The K-means algorithm aims to choose centroids that minimise the inertia, or within-cluster sum-of-squares criterion,

$$I = \sum_{j=1}^{K} \sum_{x \in C_j} |x - \mu_j|^2$$

Inertia can be recognized as a measure of how internally coherent clusters are.

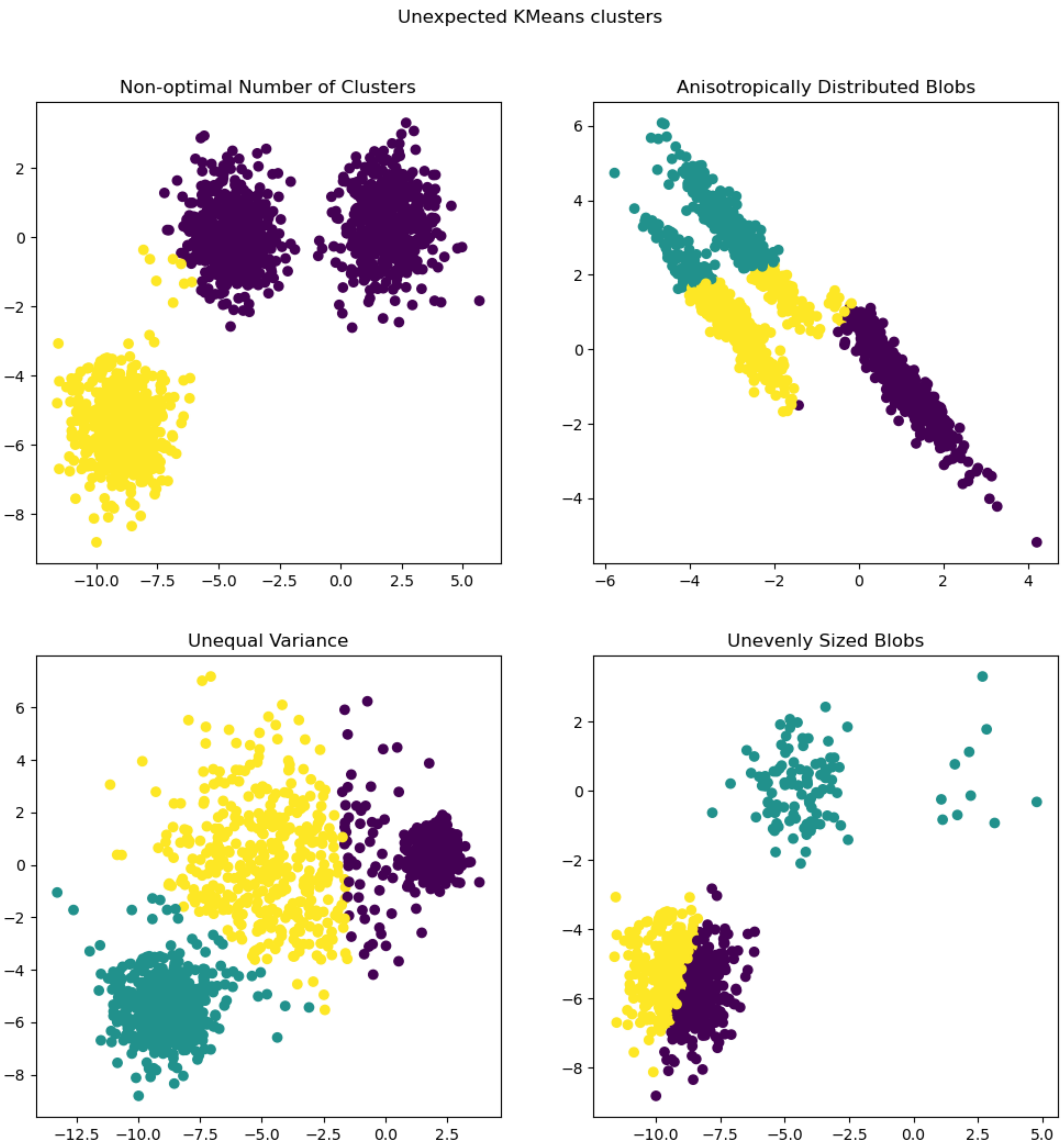

Step 1
Step 2
Step 3
Step 4
Step 5

K-Means

K-Means suffers from various drawbacks:

- Inertia makes the assumption that clusters are convex and isotropic, which is not always the case. *It responds poorly to elongated clusters, or manifolds with irregular shapes*.

- Inertia is not a normalized metric: we just know that lower values are better and zero is optimal. But in very high-dimensional spaces, Euclidean distances tend to become inflated (this is an instance of the so-called “curse of dimensionality”).
Running a dimensionality reduction algorithm such as Principal component analysis (PCA) prior to k-means clustering can alleviate this problem and speed up the computations.


Unexpected KMeans clusters
Non-optimal Number of Clusters
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
Anisotropically Distributed Blobs
6
4
2
0
−2
−4
−6
−4
−2
0
2
4
Unequal Variance
6
4
2
0
−2
−4
−6
−8
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
Unevenly Sized Blobs
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0

DBSCAN

DBSCAN

DBSCAN - **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise. Finds core samples of high density and expands clusters from them. Good for data which contains clusters of similar density.

The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. Due to this rather generic view, clusters found by DBSCAN can be *any shape*, as opposed to K-means, which assumes that clusters are convex shaped.

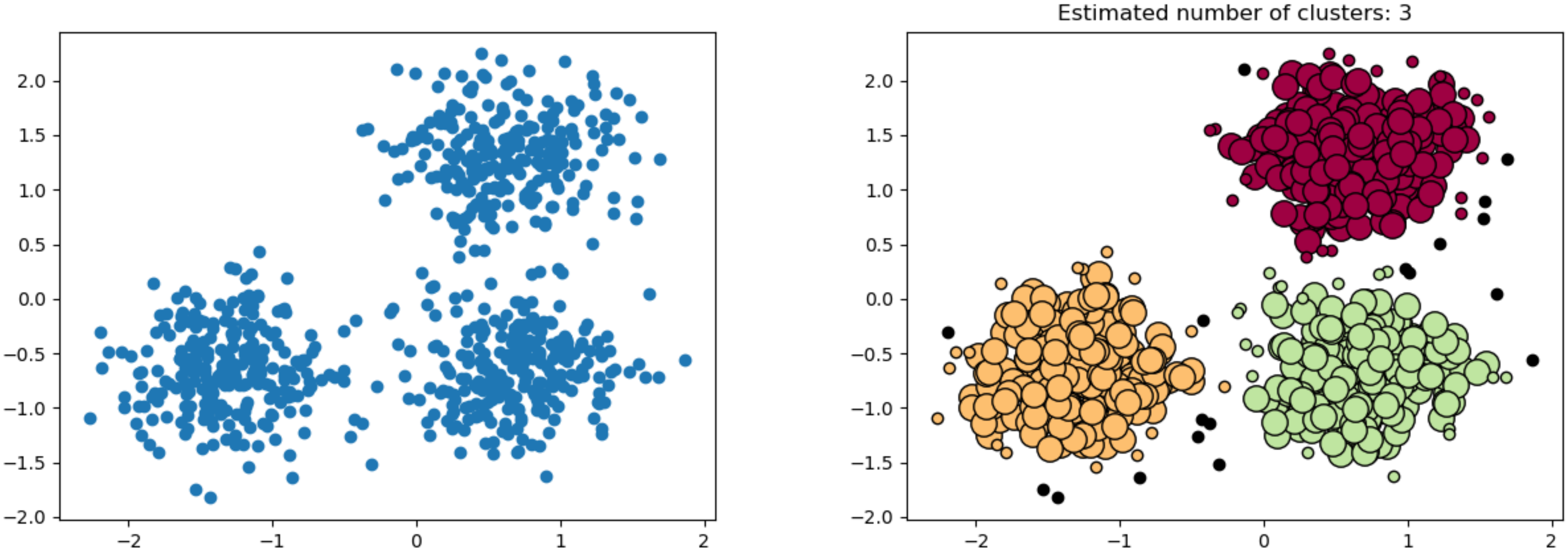
DBSCAN

The central component to the DBSCAN is the concept of **core samples**, which are samples that are in areas of high density. A cluster is therefore a set of core samples, each close to each other (measured by some distance measure) and a set of non-core samples that are close to a core sample (but are not themselves core samples).

There are two parameters in the algorithm:

- *min_samples:* The number of samples in a neighborhood for a point to be considered as a core point (this includes the point itself).

- *eps:* The maximum distance between two samples for one to be considered as in the neighborhood of the other.


Estimated number of clusters: 3
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−2
−1
0
1
2

@astronomiaOAN

AstronomiaOAN

@astronomiaOAN