

Projections and clusterings

Formation IA biodiversité

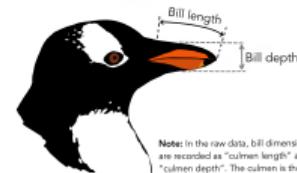
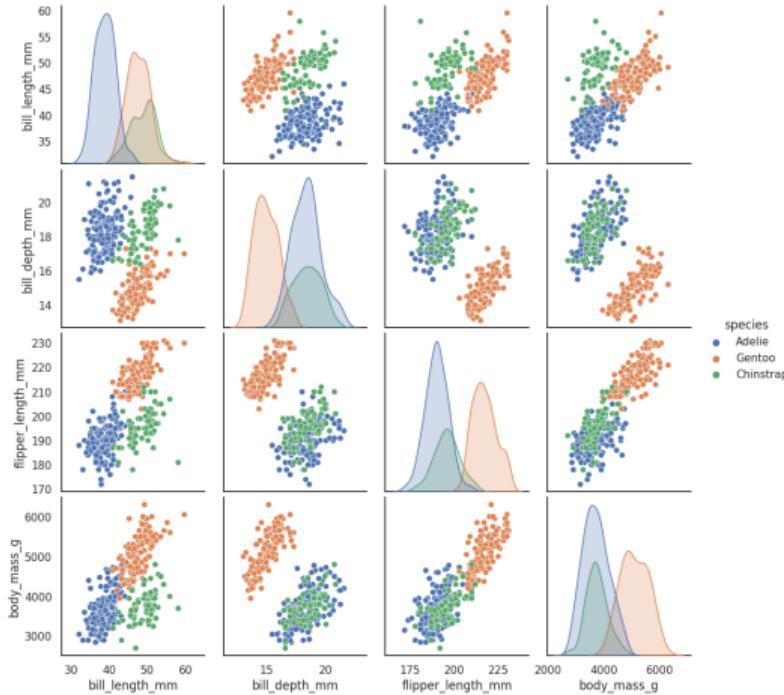
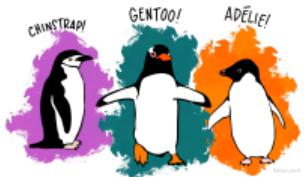
Paul Tresson

May 14, 2025

UMR AMAP

Introduction

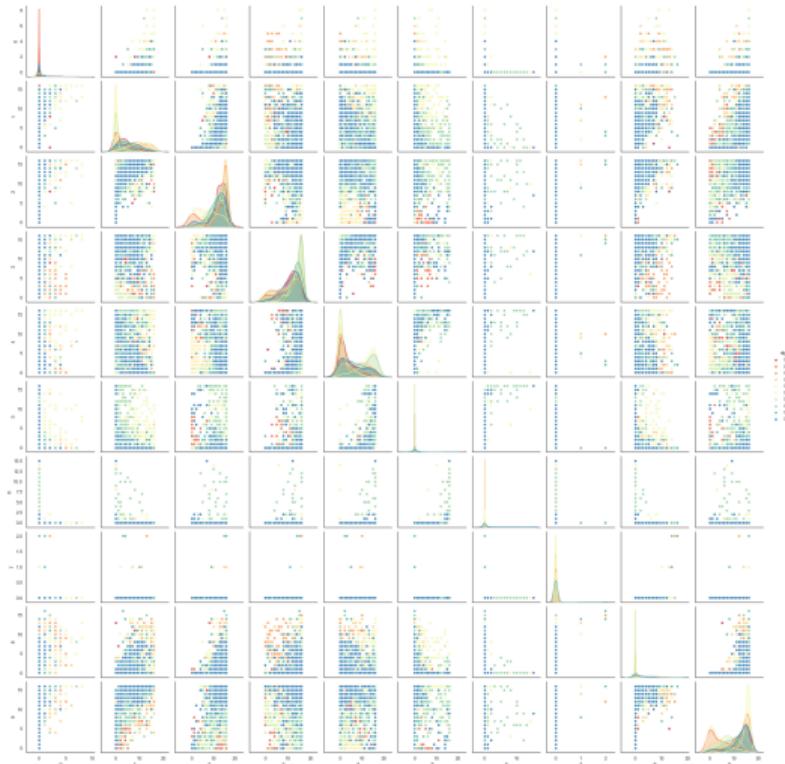
Visualize what is happening in higher dimensions



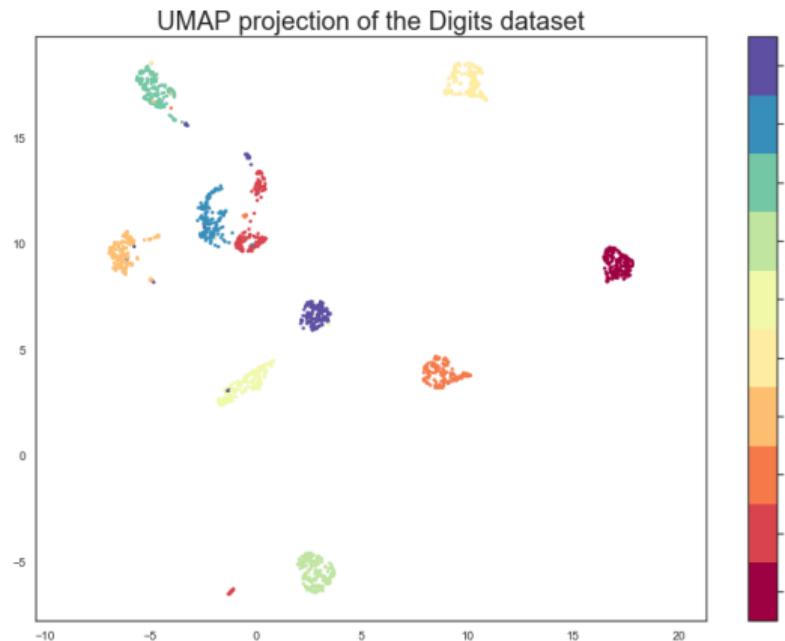
Note: In the raw data, bill dimensions are recorded as "culmen length" and "culmen depth". The culmen is the dorsal ridge atop the bill.

Visualize what is happening in higher dimensions

Visualize what is happening in higher dimensions



Visualize what is happening in higher dimensions

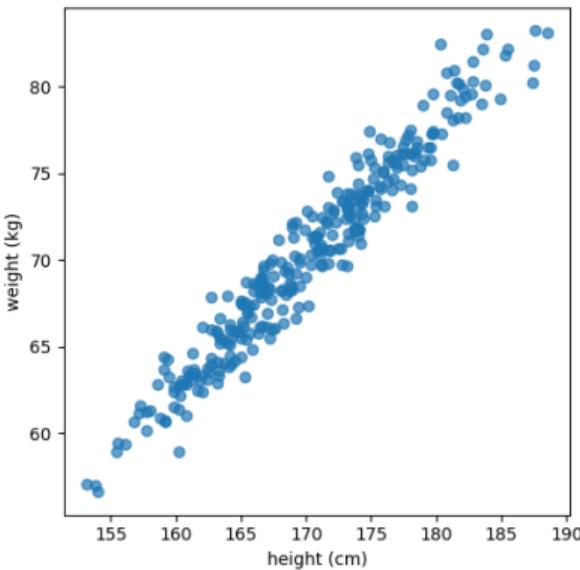


Projections

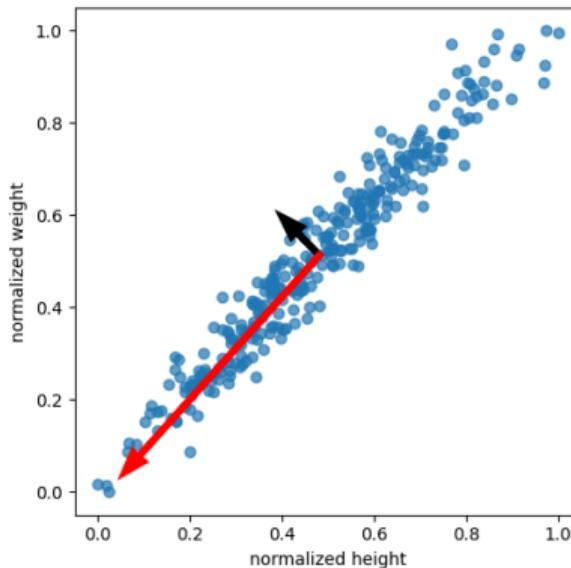
Projections

PCA

PCA



PCA



PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

$$C = \frac{1}{n-1} X^T X$$

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

$$C = \frac{1}{n-1} X^T X$$

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

$$Cv = \lambda v$$

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \lambda \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

PCA calculation

$$X = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ \dots & \dots \end{bmatrix}$$

$$C = \frac{1}{n-1} X^T X$$

$$C = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix}$$

$$Cv = \lambda v$$

$$\begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{yx} & \sigma_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \lambda \begin{bmatrix} v_x \\ v_y \end{bmatrix}$$

$$P = \begin{bmatrix} v_x & v_y \end{bmatrix}$$

PCA calculation

```
# normalize
normalized_data = (points - np.min(points, axis=0)) / (
    np.max(points, axis=0) - np.min(points, axis=0)
)

# get covariance
cov = np.cov(normalized_data, rowvar=False)

# calculate eigenvalues and eigenvectors of the covariance matrix
eigvals, eigvecs = np.linalg.eig(cov)

# scale eigenvectors
scaled_eigvecs = eigvecs * np.sqrt(eigvals)
```

PCA advantages and drawbacks

Advantages

- fast

Drawbacks

PCA advantages and drawbacks

Advantages

- fast
- scales well

Drawbacks

PCA advantages and drawbacks

Advantages

- fast
- scales well
- explanatory

Drawbacks

PCA advantages and drawbacks

Advantages

- fast
- scales well
- explanatory

Drawbacks

- **not suited for non-linear data**

Non linear data

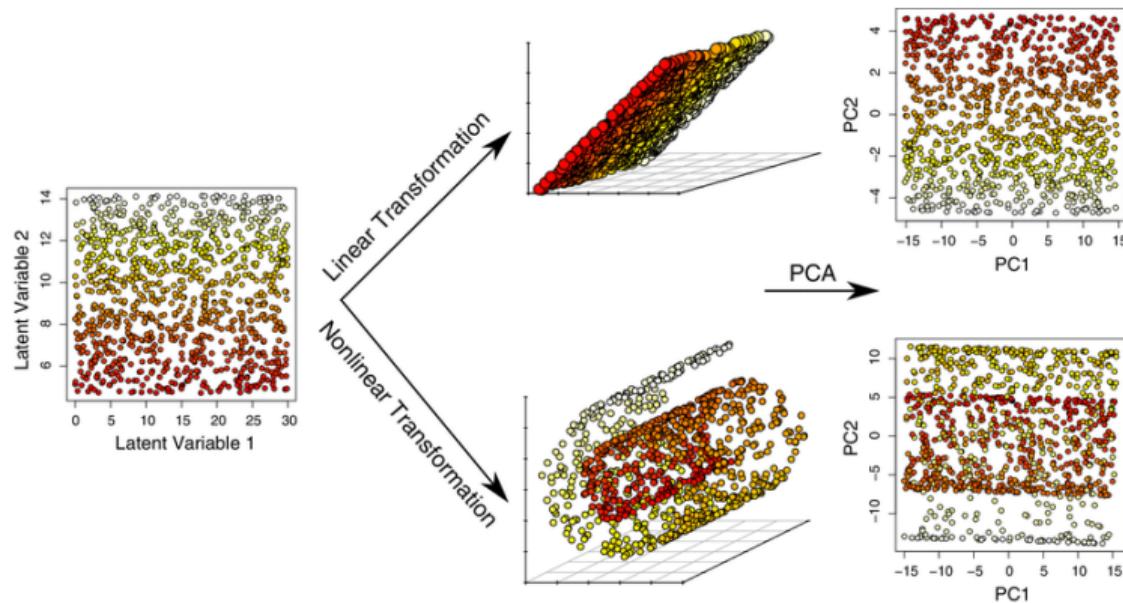


Figure from Du, 2019

Projections

UMAP

How to work with non linear data

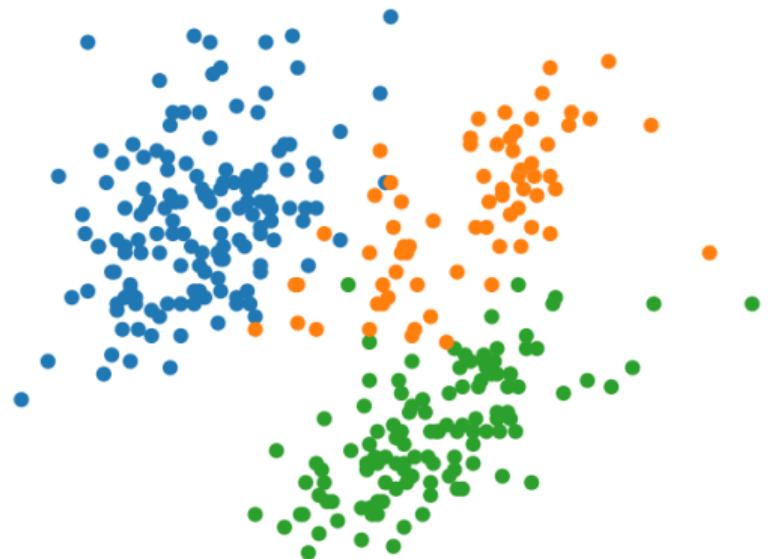
1. Find a good representation of the data in high dimension
2. Fit a good representation of in low dimension

- **SNE** (Stochastic Neighbor Embedding) Hinton and Roweis, 2002

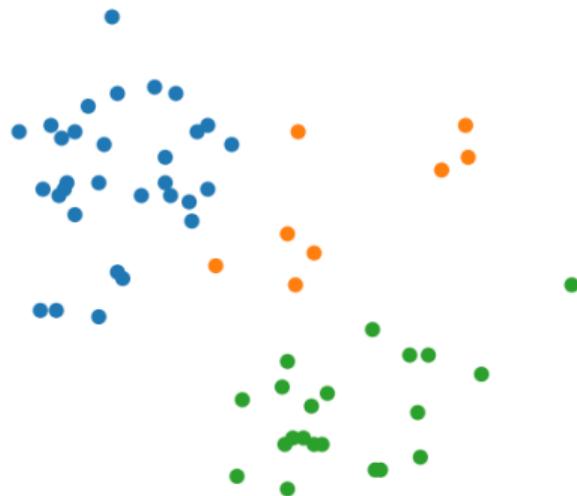
- **SNE** (Stochastic Neighbor Embedding) Hinton and Roweis, 2002
- **T-SNE** (T-distributed SNE) Van der Maaten and Hinton, 2008

- **SNE** (Stochastic Neighbor Embedding) Hinton and Roweis, 2002
- **T-SNE** (T-distributed SNE) Van der Maaten and Hinton, 2008
- **UMAP** (Uniform Manifold Approximation and Projection) McInnes et al., 2018

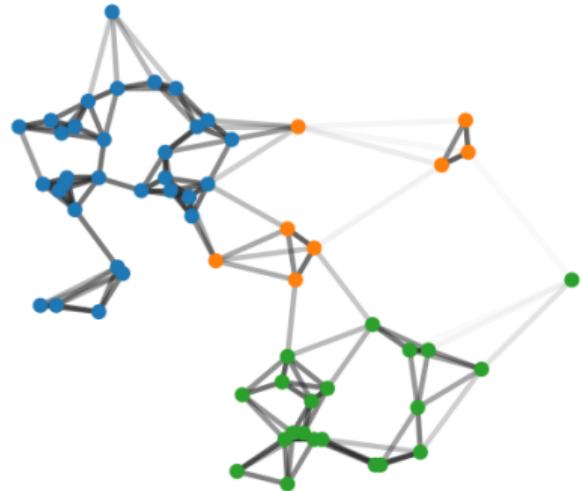
UMAP : Finding high dimension graph



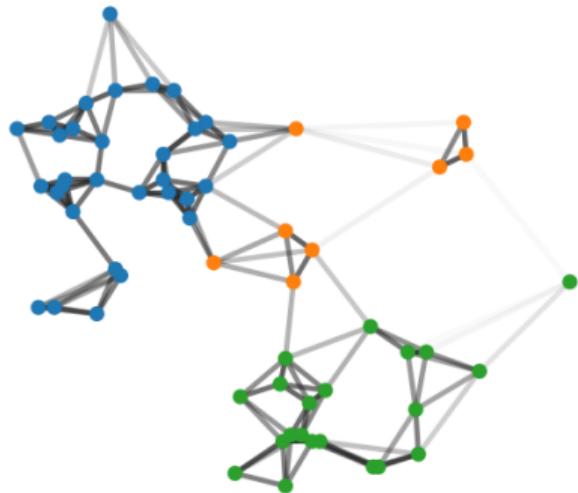
UMAP : Finding high dimension graph



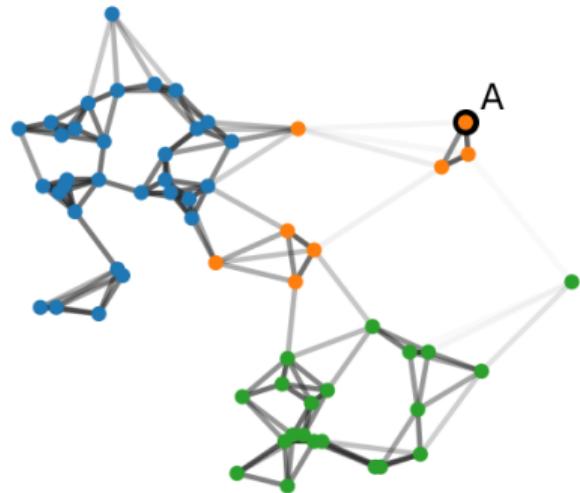
UMAP : Finding high dimension graph



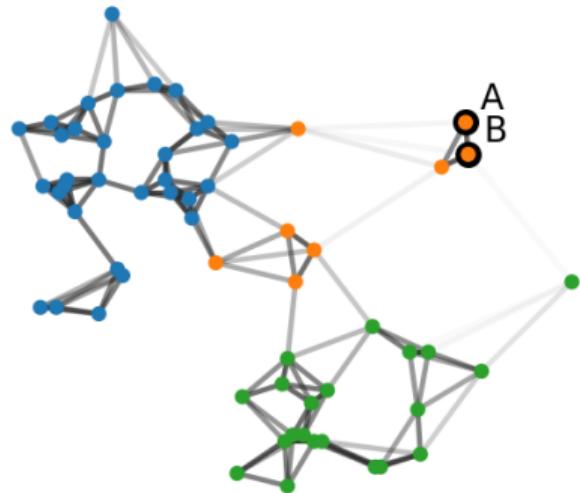
UMAP : Fitting low dimensional representation



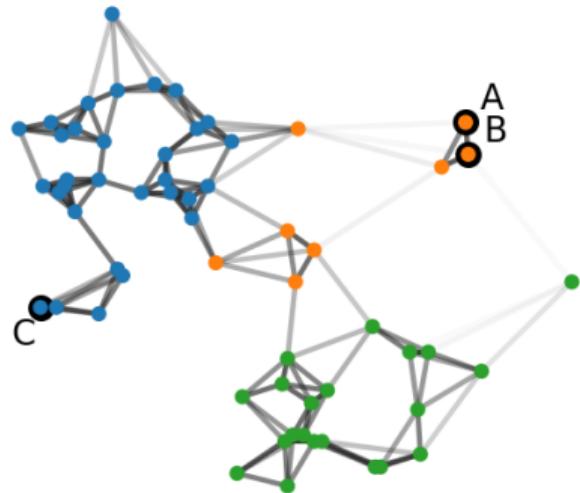
UMAP : Fitting low dimensional representation



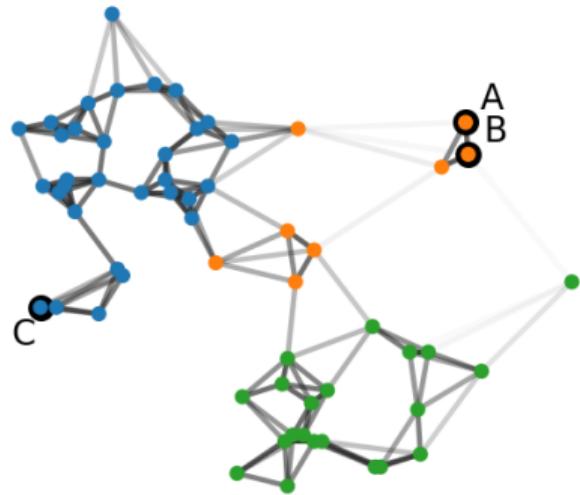
UMAP : Fitting low dimensional representation



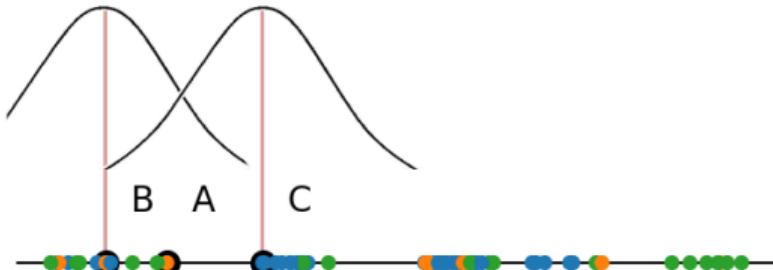
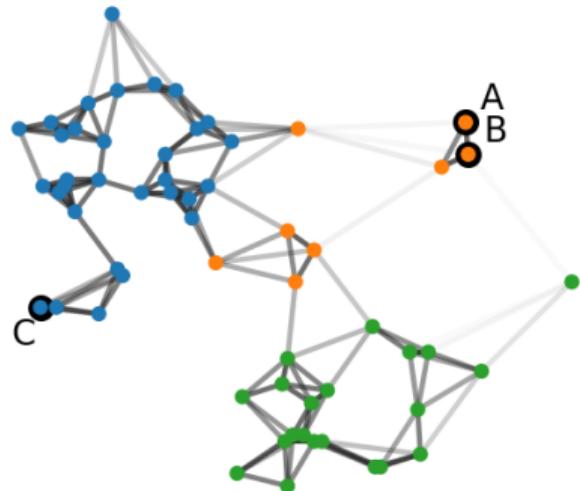
UMAP : Fitting low dimensional representation



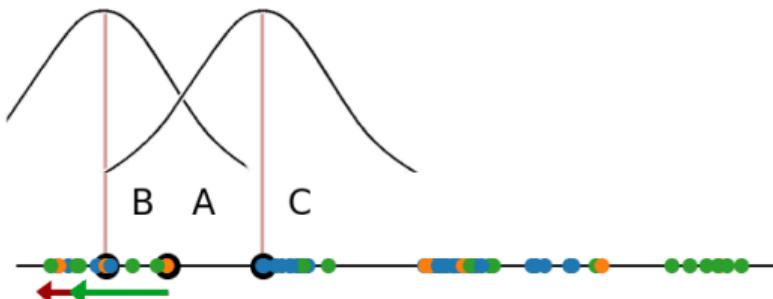
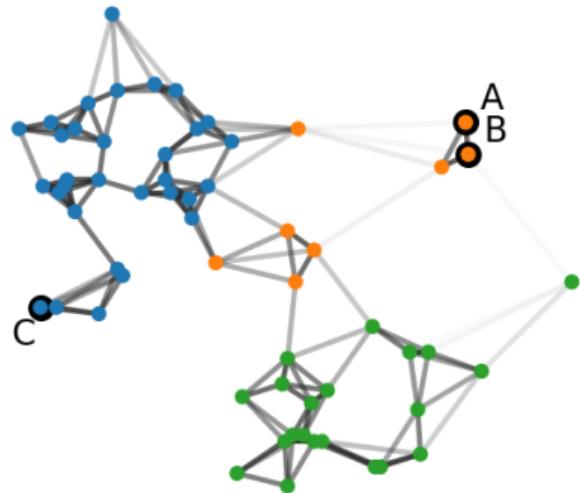
UMAP : Fitting low dimensional representation



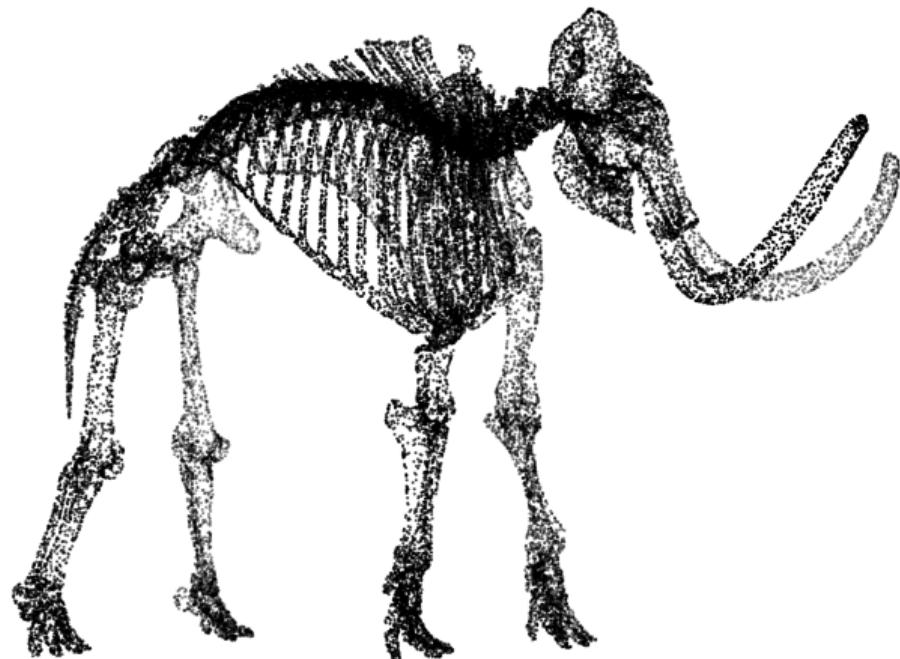
UMAP : Fitting low dimensional representation



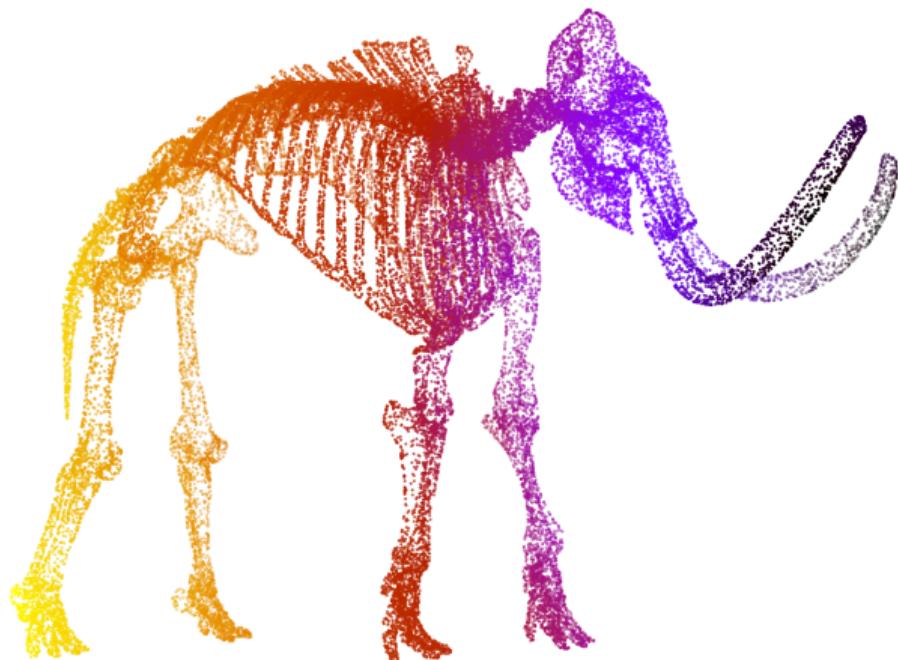
UMAP : Fitting low dimensional representation



Mammoth example



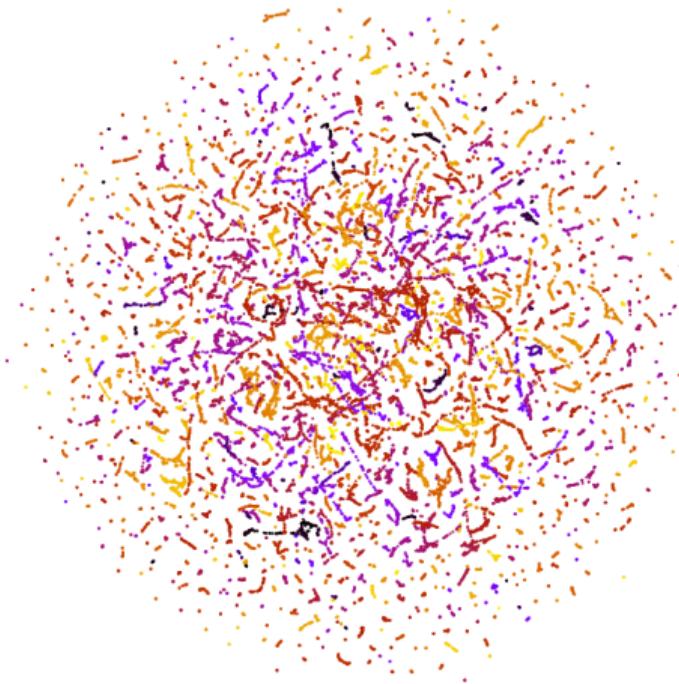
Mammoth example



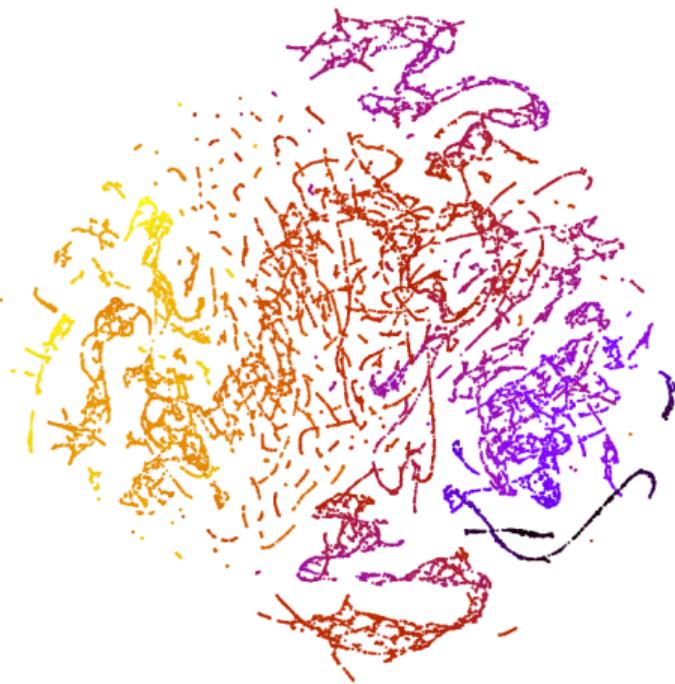
Mammoth example



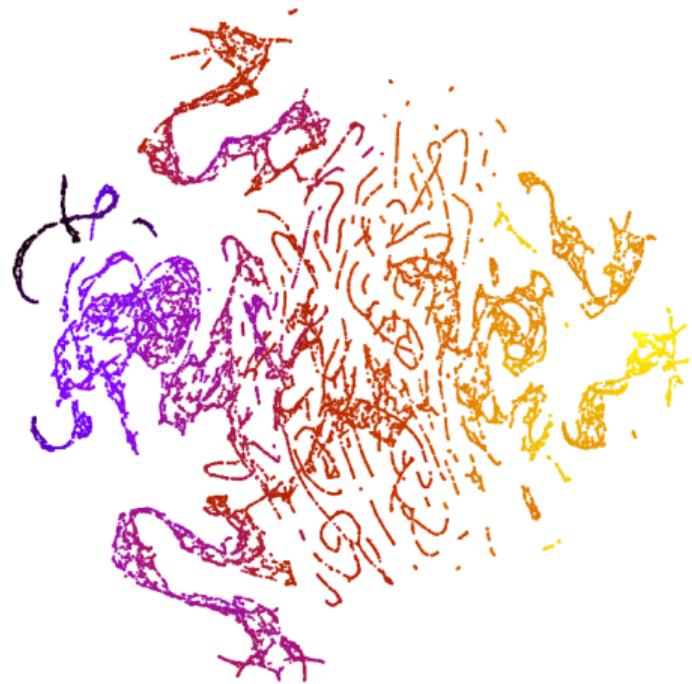
Mammoth example



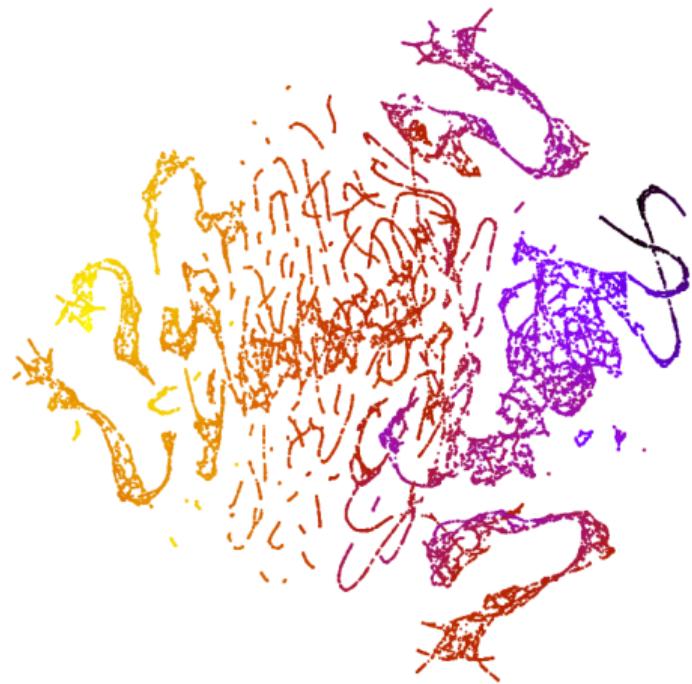
Mammoth example



Mammoth example



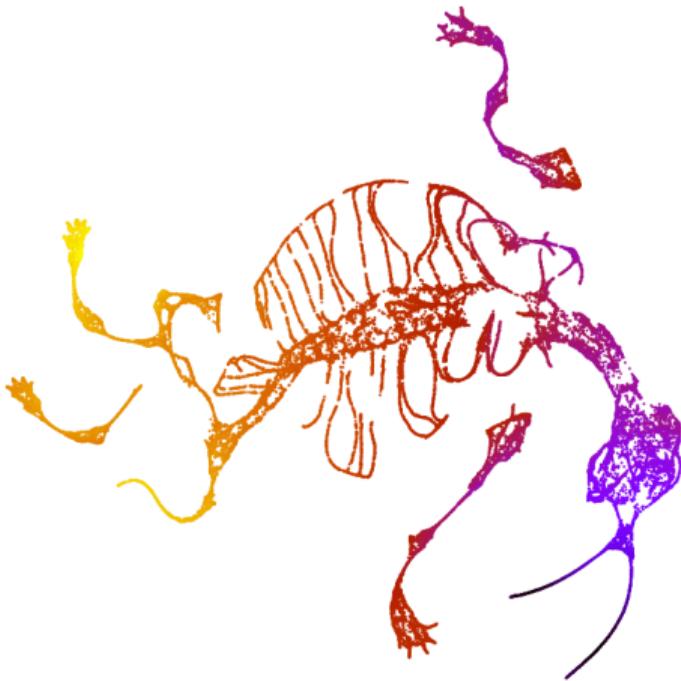
Mammoth example



Mammoth example



Mammoth example



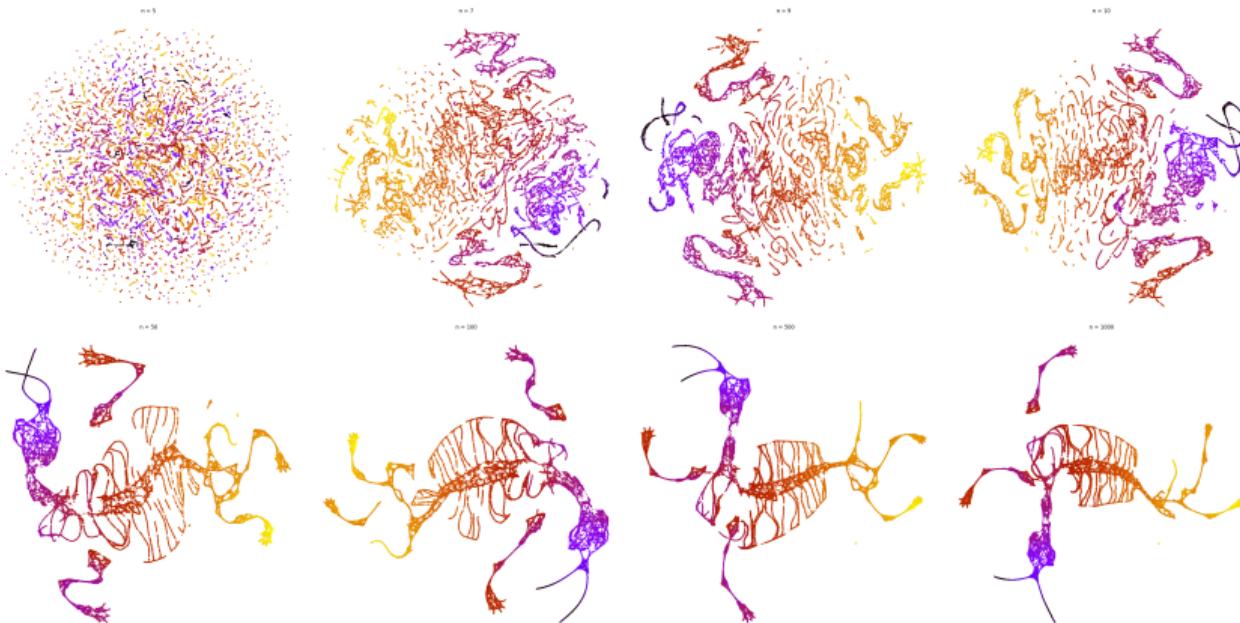
Mammoth example



Mammoth example



Mammoth example



UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic

T-SNE

- probabilities

UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic
- deterministic initialization

T-SNE

- probabilities
- random initialization

UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic
- determinist initialization
- \log_2 similarity scores

T-SNE

- probabilities
- random initialization
- gaussian distribution similarity

UMAP vs. T-SNE

UMAP

- graph theory, fuzzy logic
 - determinist initialization
 - \log_2 similarity scores
 - update by pairs
- scales well for large datasets

T-SNE

- probabilities
 - random initialization
 - gaussian distribution similarity
 - update all points
- scales less

Scaling

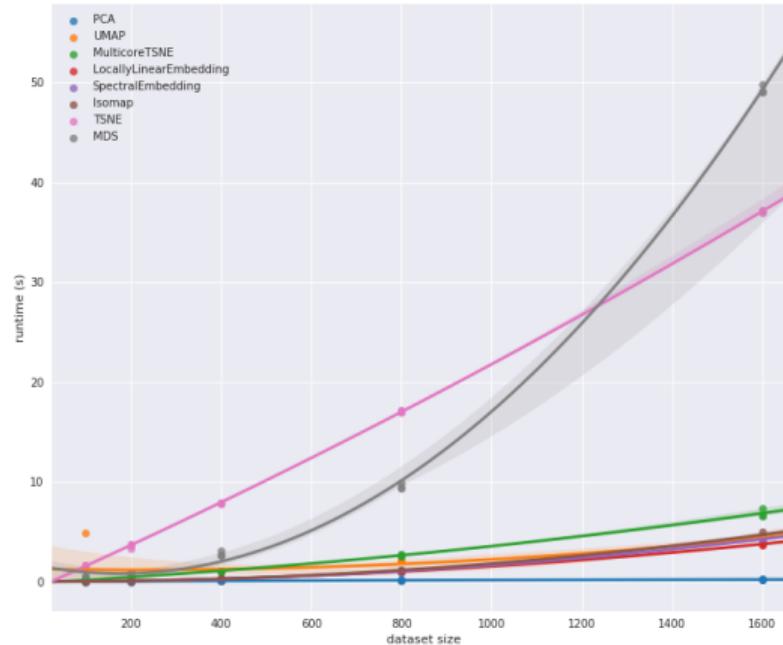


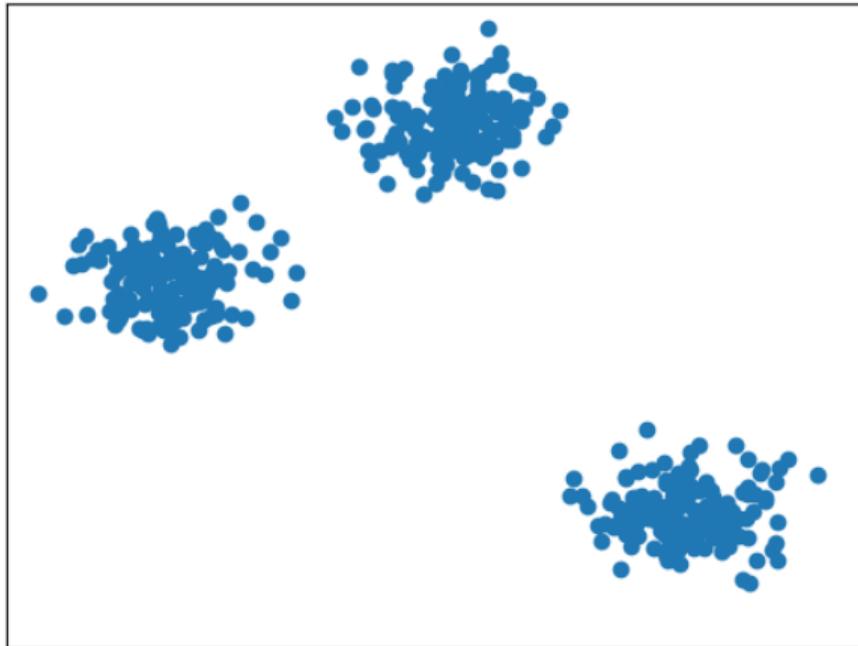
Figure from umap-learn documentation

Clustering

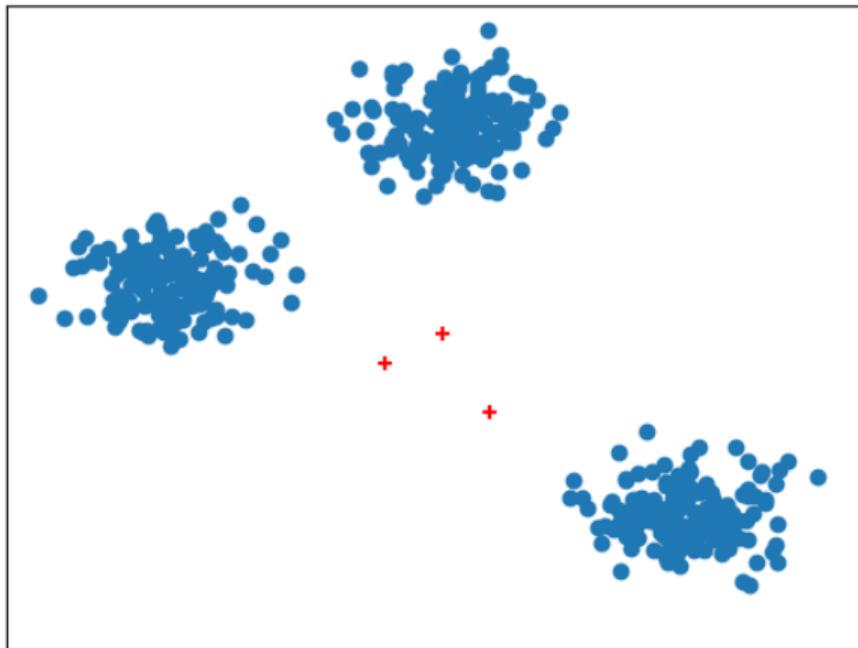
Clustering

K-means

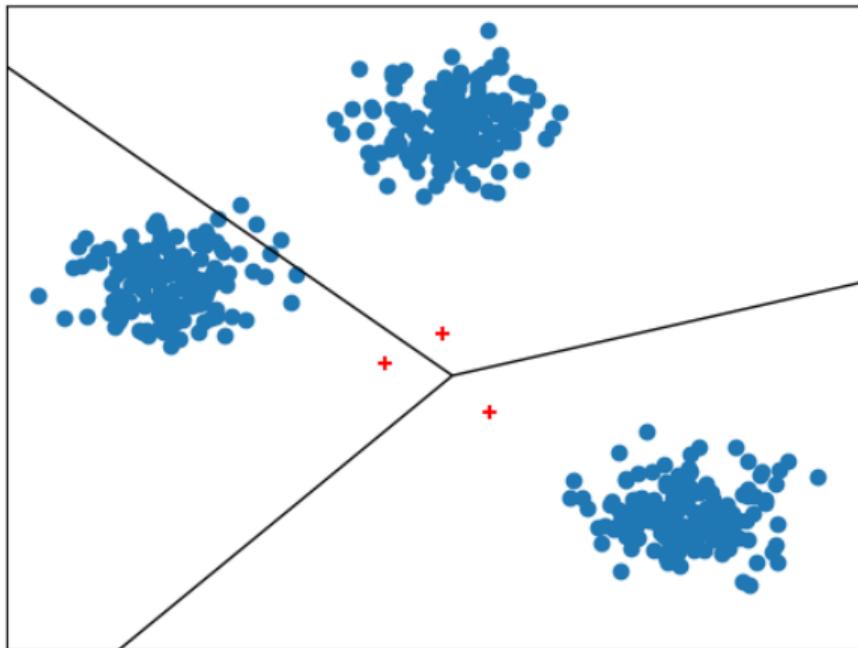
K-means



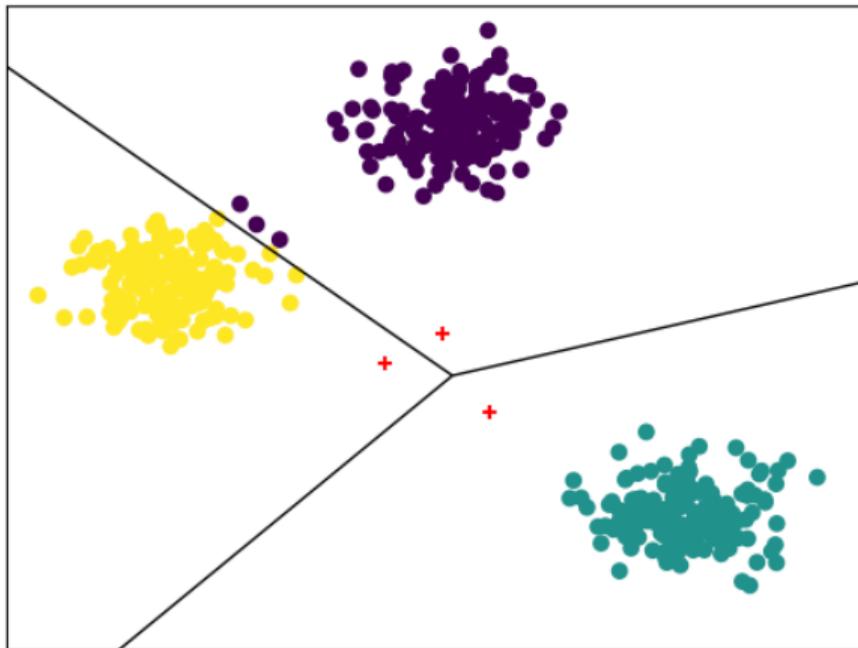
K-means



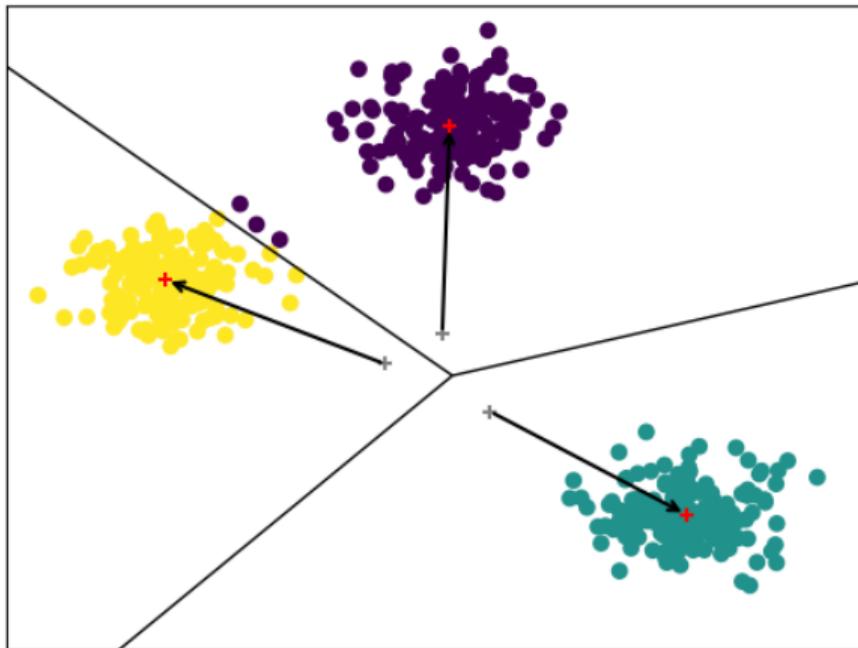
K-means



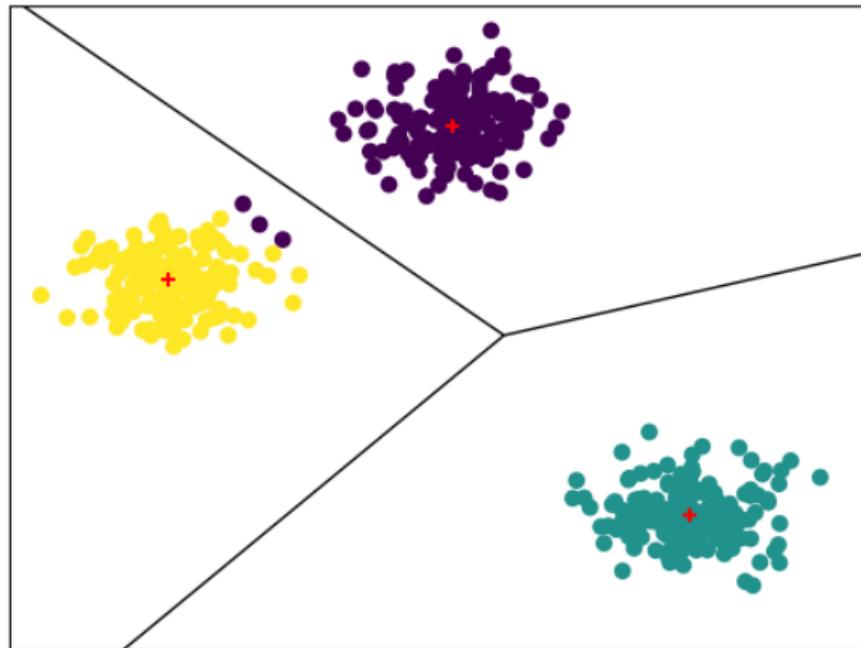
K-means



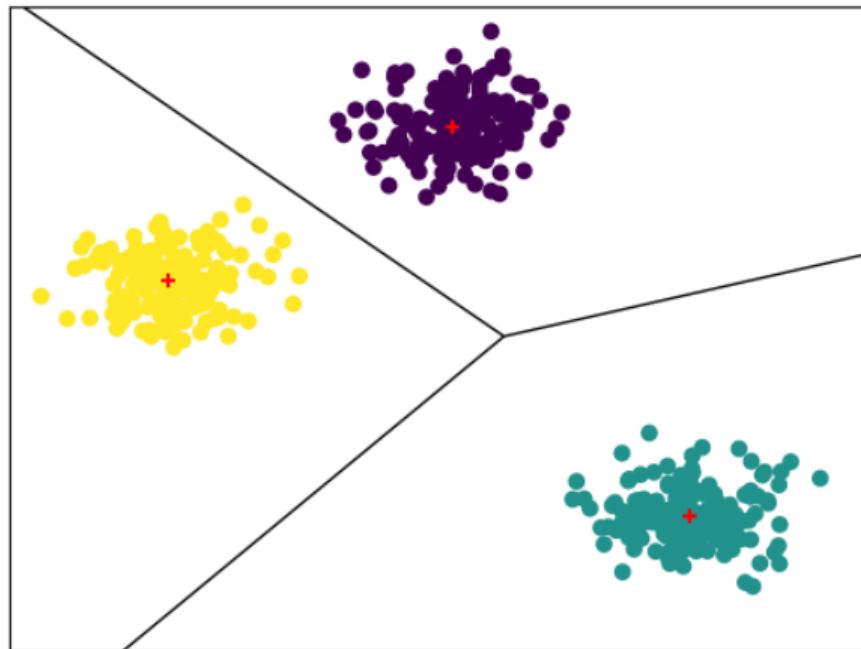
K-means



K-means



K-means



Determine the good K ? Elbow method

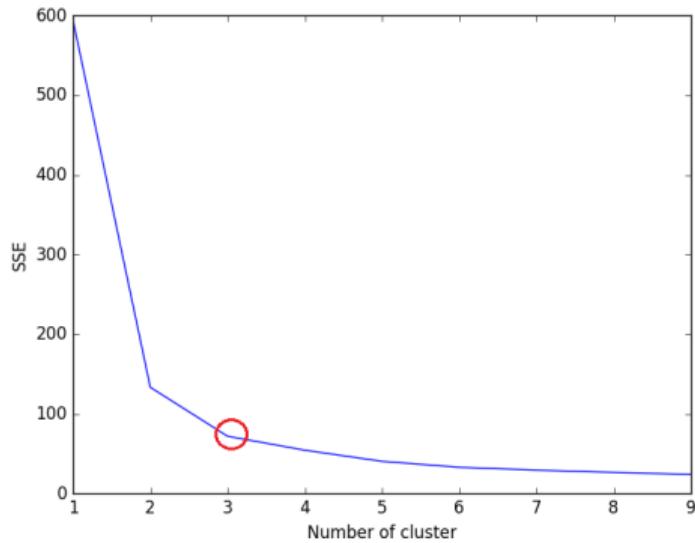
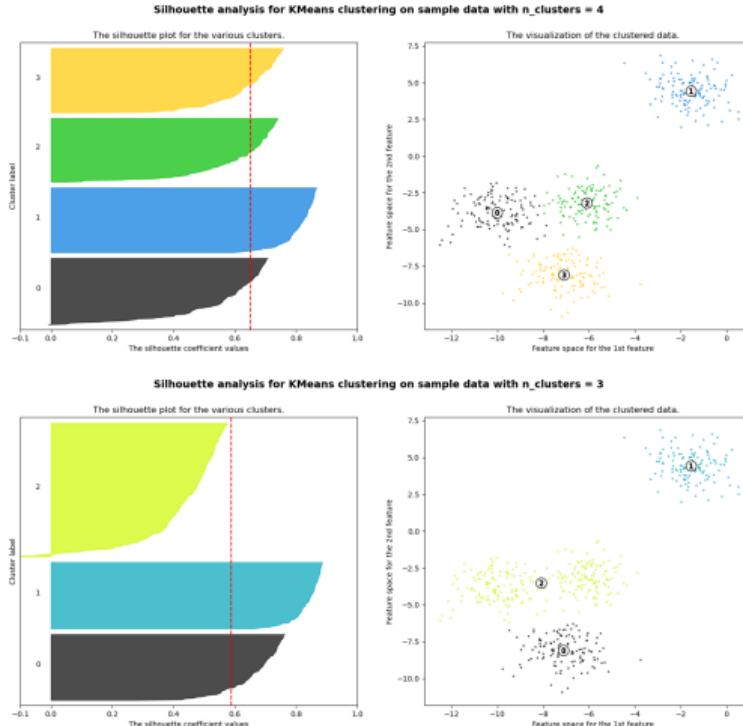


Figure from SO

Determine the good K ? Silhouette



Figures from scikit-learn documentation

K-means advantages and drawbacks

Advantages

- fast

Drawbacks

K-means advantages and drawbacks

Advantages

- fast
- scales well

Drawbacks

K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well

Drawbacks

K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well
- robust

Drawbacks

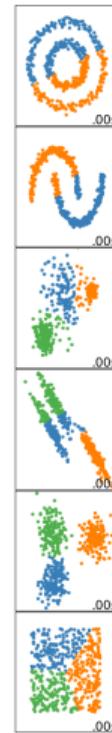
K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well
- robust

Drawbacks

- **not suited for non-linear data**



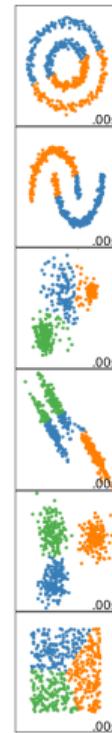
K-means advantages and drawbacks

Advantages

- fast
- scales well
- converges well
- robust

Drawbacks

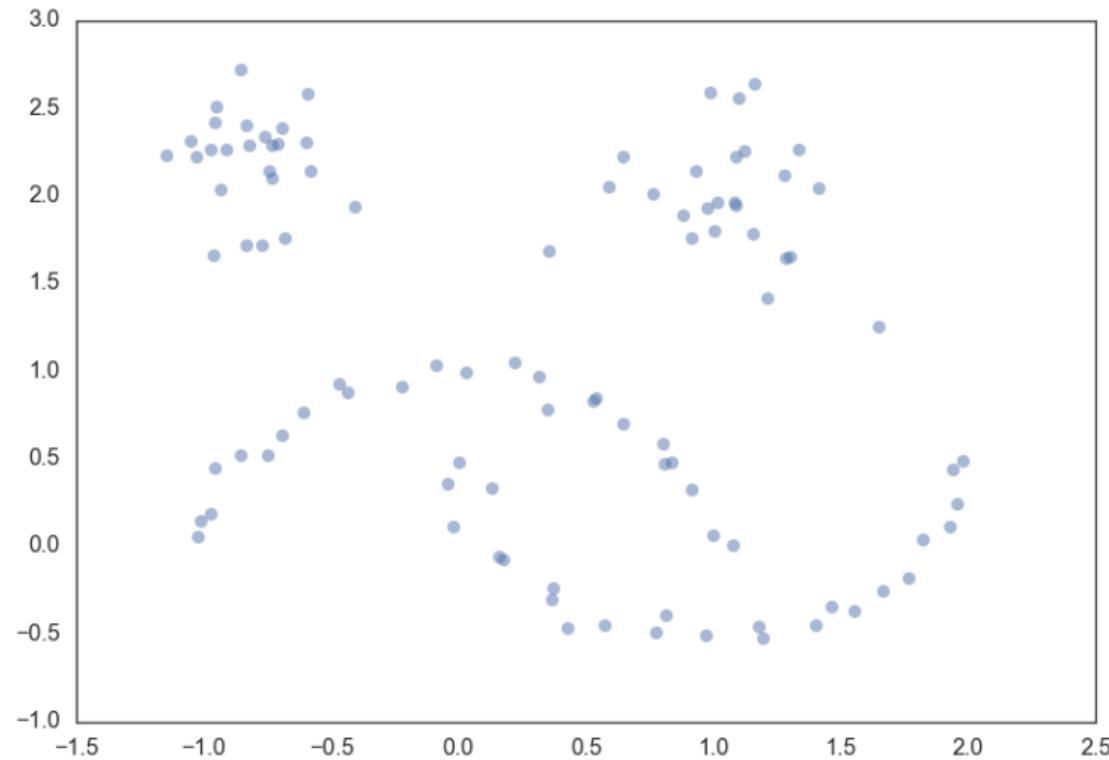
- **not suited for non-linear data**
→ Density based clustering



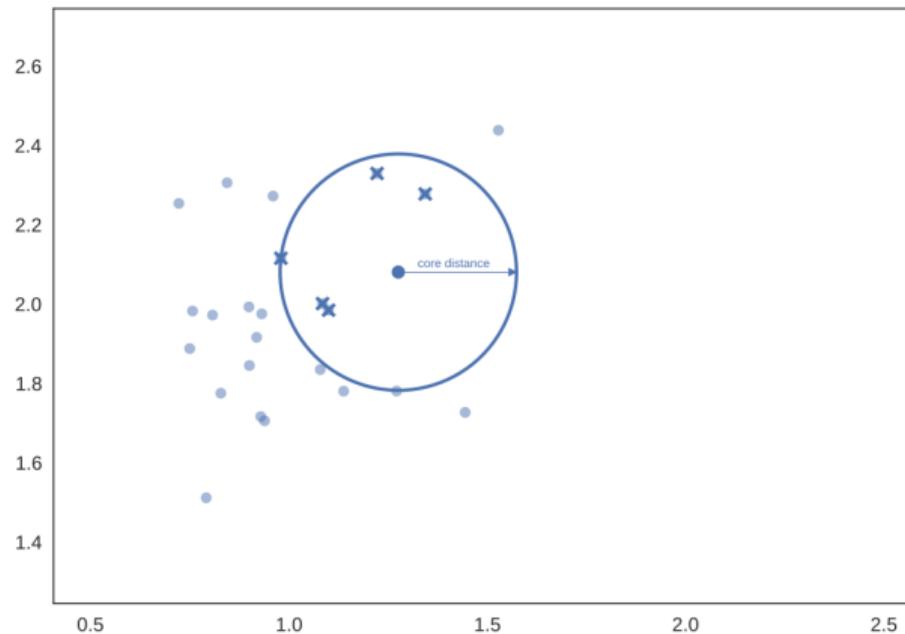
Clustering

HDBSCAN

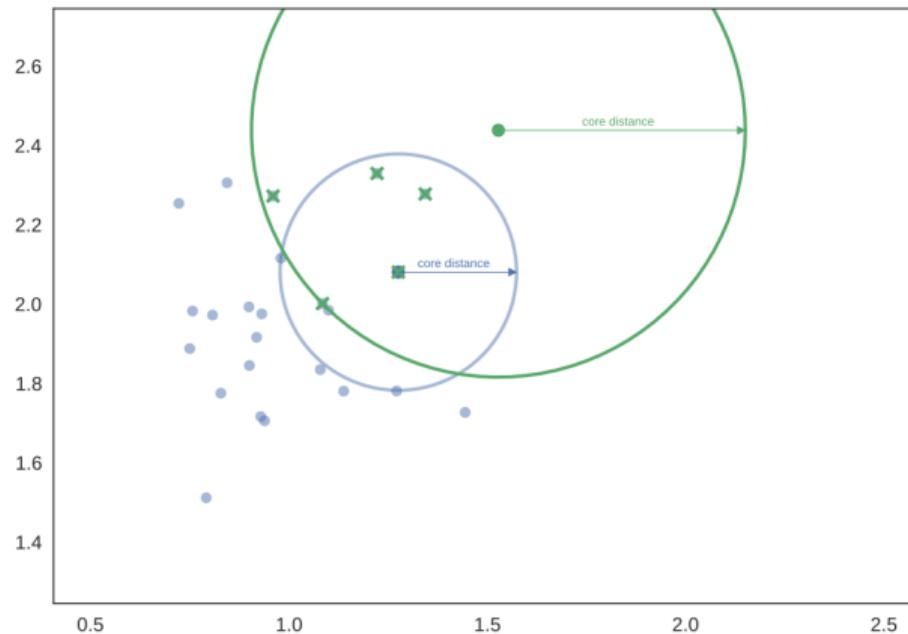
HDBSCAN



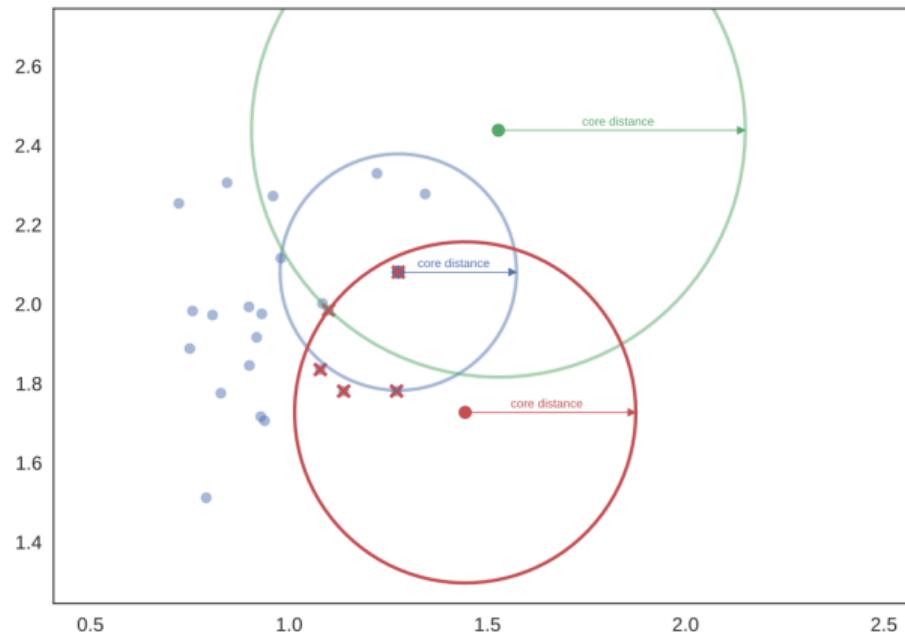
HDBSCAN



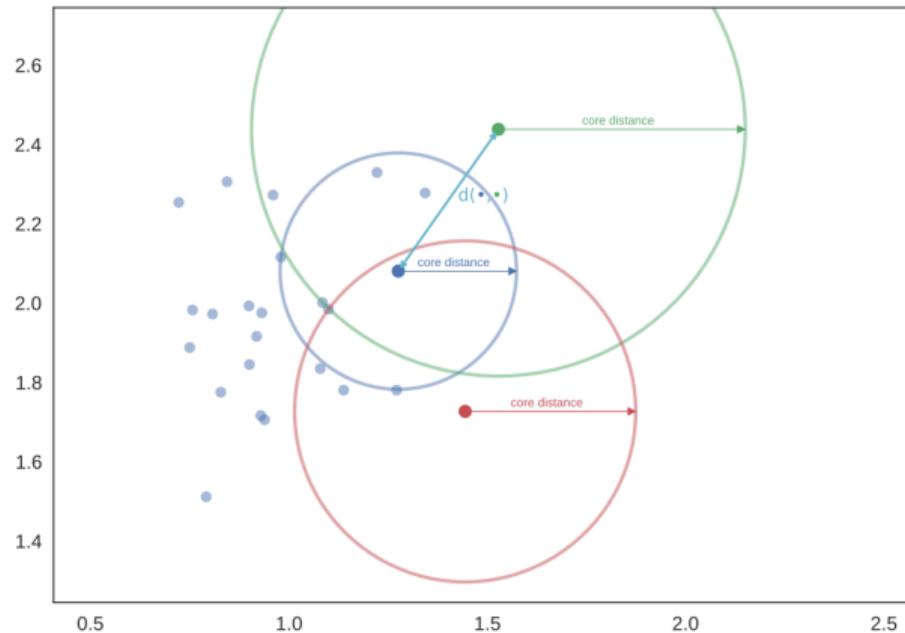
HDBSCAN



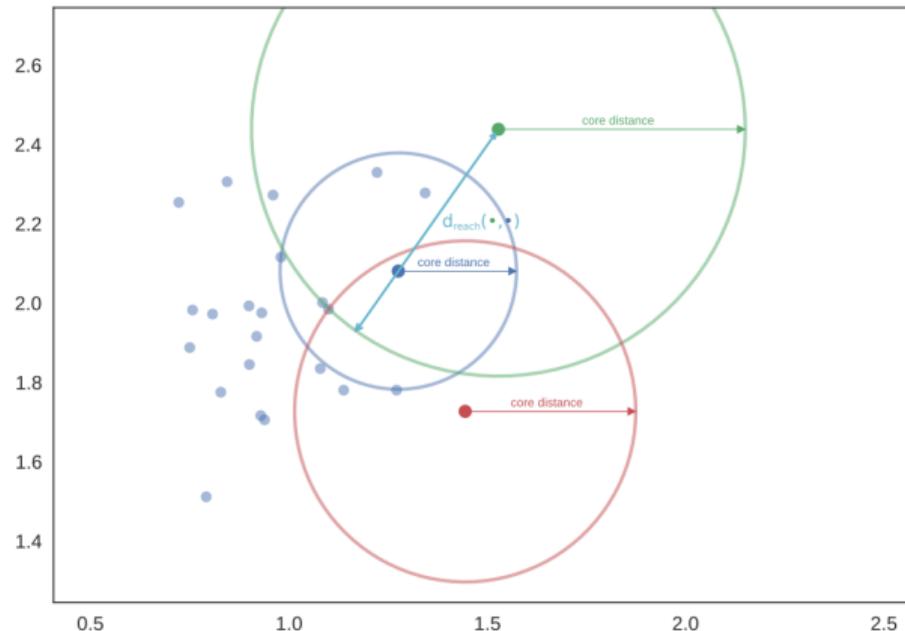
HDBSCAN



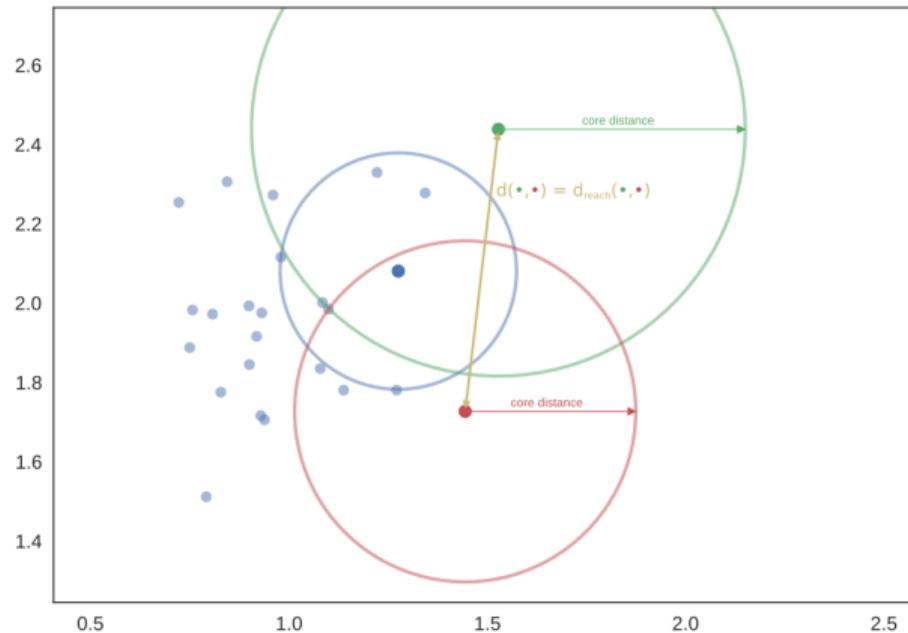
HDBSCAN



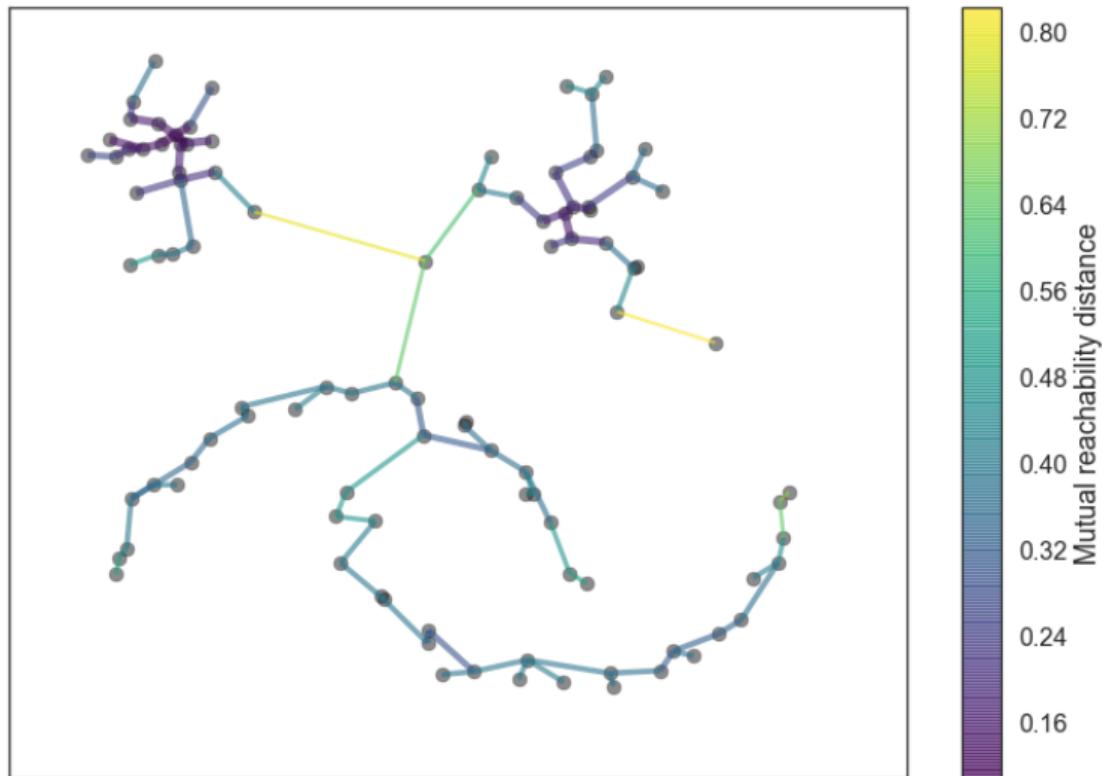
HDBSCAN



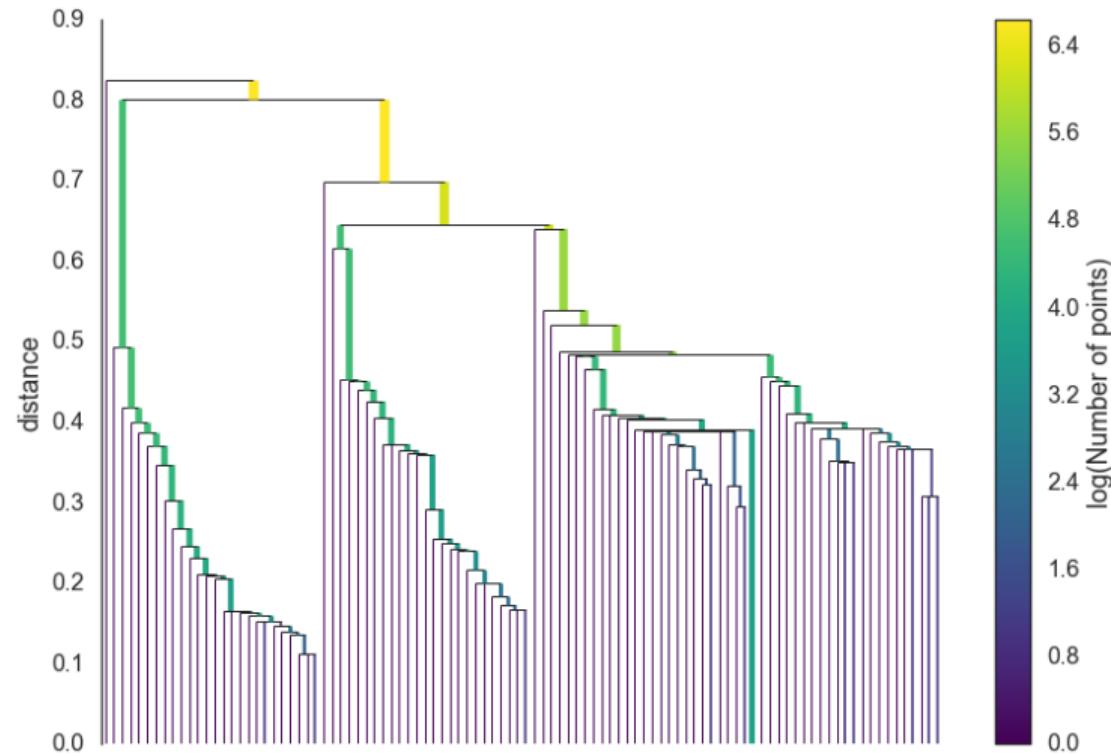
HDBSCAN



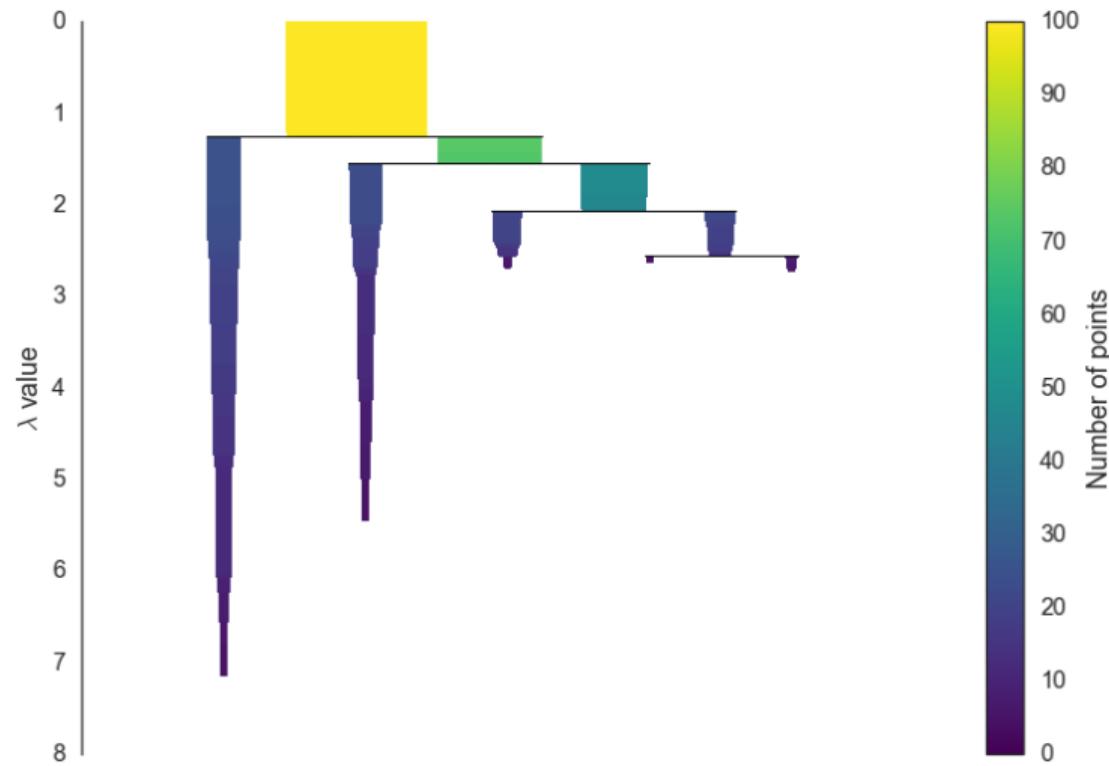
HDBSCAN



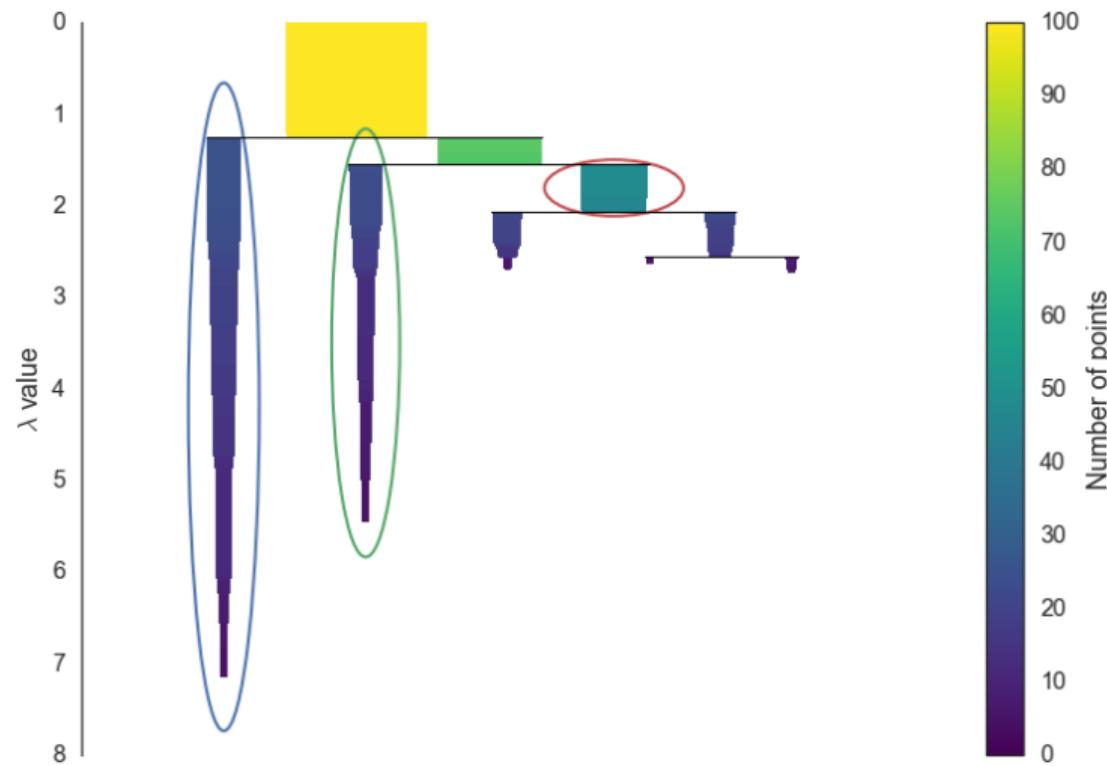
HDBSCAN



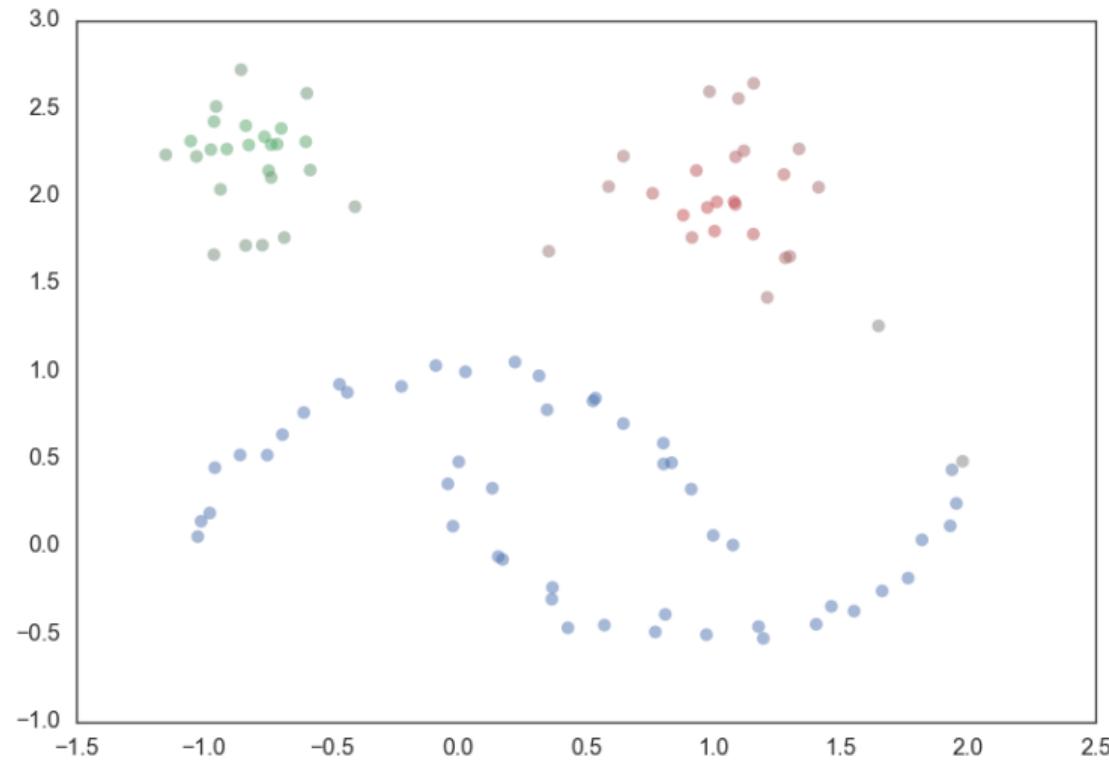
HDBSCAN



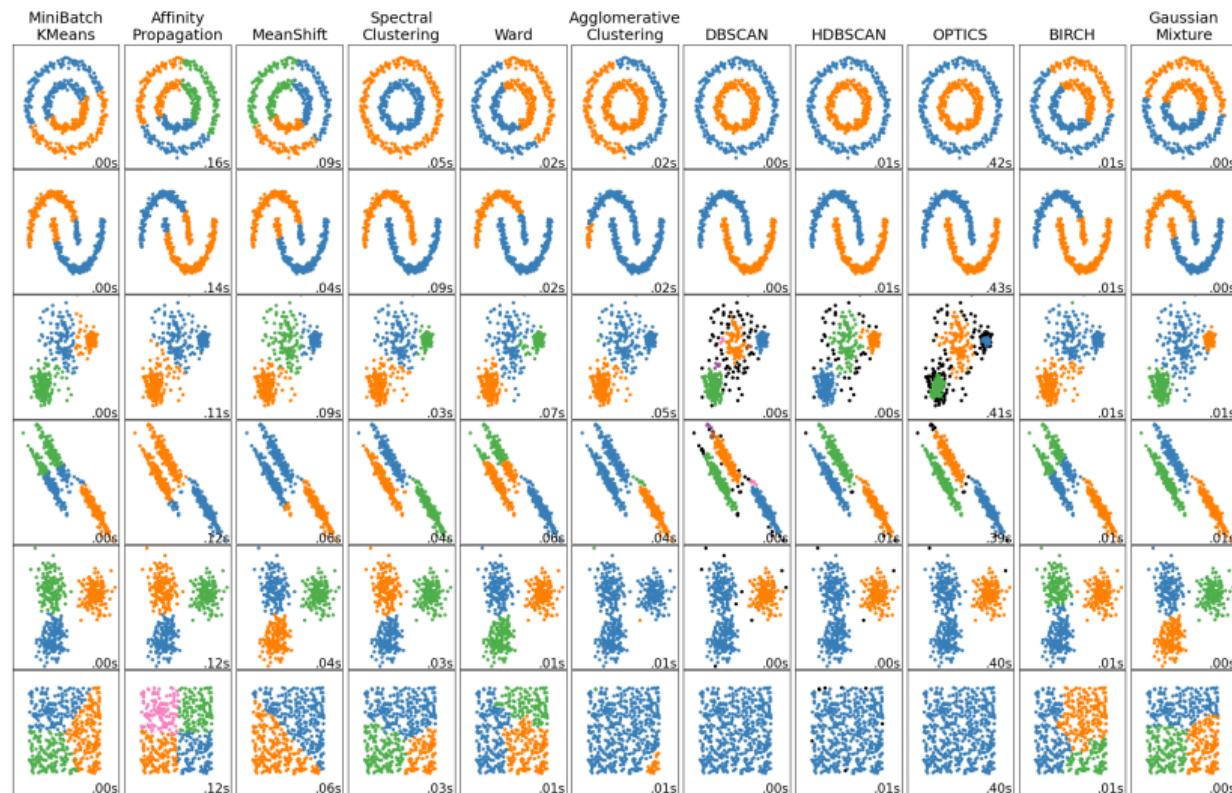
HDBSCAN



HDBSCAN



A lot of options !



Usefull ressources

- scikit-learn docs !
- deepia
- StatQuest

Thanks for you attention !

Let's practice !

References i

- Du, Trina Y (2019). “**Dimensionality reduction techniques for visualizing morphometric data: Comparing principal component analysis to nonlinear methods**”. In: *Evolutionary Biology* 46.1, pp. 106–121.
- Hinton, Geoffrey E and Sam Roweis (2002). “**Stochastic neighbor embedding**”. In: *Advances in neural information processing systems* 15.
- McInnes, Leland, John Healy, and James Melville (2018). “**Umap: Uniform manifold approximation and projection for dimension reduction**”. In: *arXiv preprint arXiv:1802.03426*.
- Van der Maaten, Laurens and Geoffrey Hinton (2008). “**Visualizing data using t-SNE**”. In: *Journal of machine learning research* 9.11.