Laura Žigutytė
Kather Lab, EKFZ, TU Dresden
laura.zigutyte@tu-dresden.de

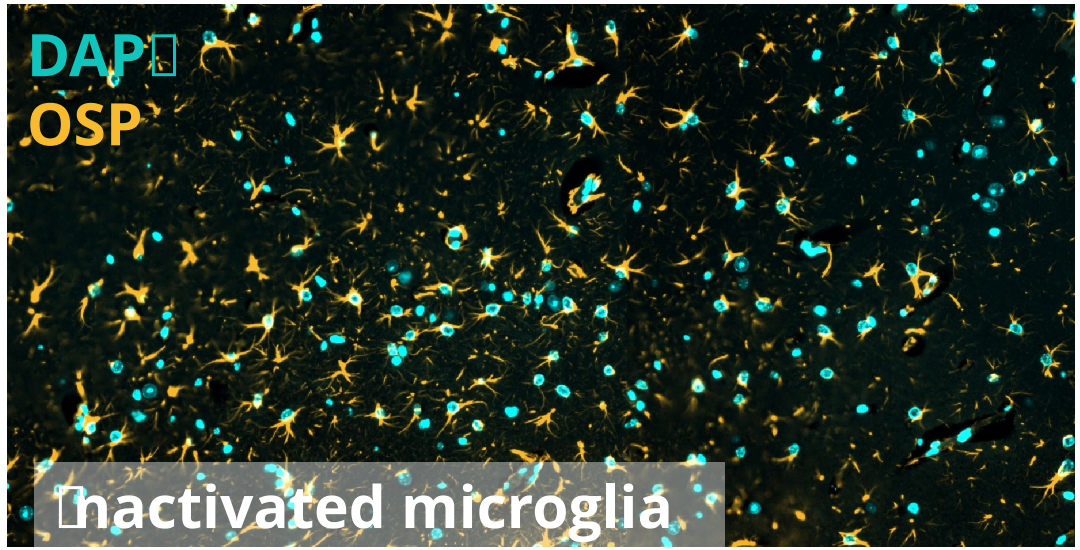
Matthias Täschner
ScaDS, Uni Leipzig
matthias.taeschner@uni-leipzig.de

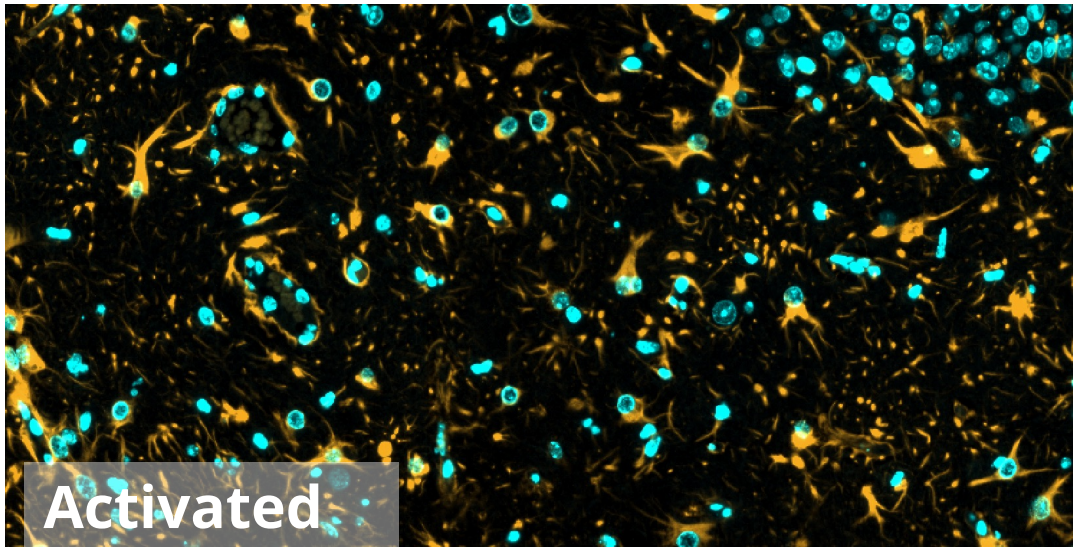
Explorative data science
Unsupervised machine learning

With materials from Robert Haase, Till Korten, Johannes Müller, Ryan Savill
ScaDS BIDS Training School on Bioimage and Data Science // May 15th

Ideal situation: We know about a measurable feature

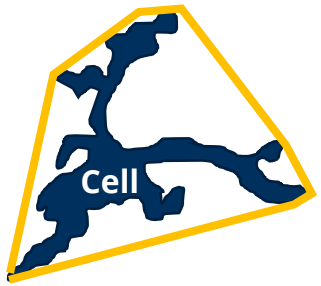
Example: Inactivated vs. activated microglia in mouse brain





Source: Slice2Volume, https://rodare.hzdr.de/record/1849, shared under CC-BY 4.0

Existing scores: "Ramification index" (Wittekind et al., 2022)



$$Rammification\ index = \frac{A_C}{A_P}$$

Solidity | Perimeter | Circularity

Ideal workflow

Get Data

Measure defined score/features

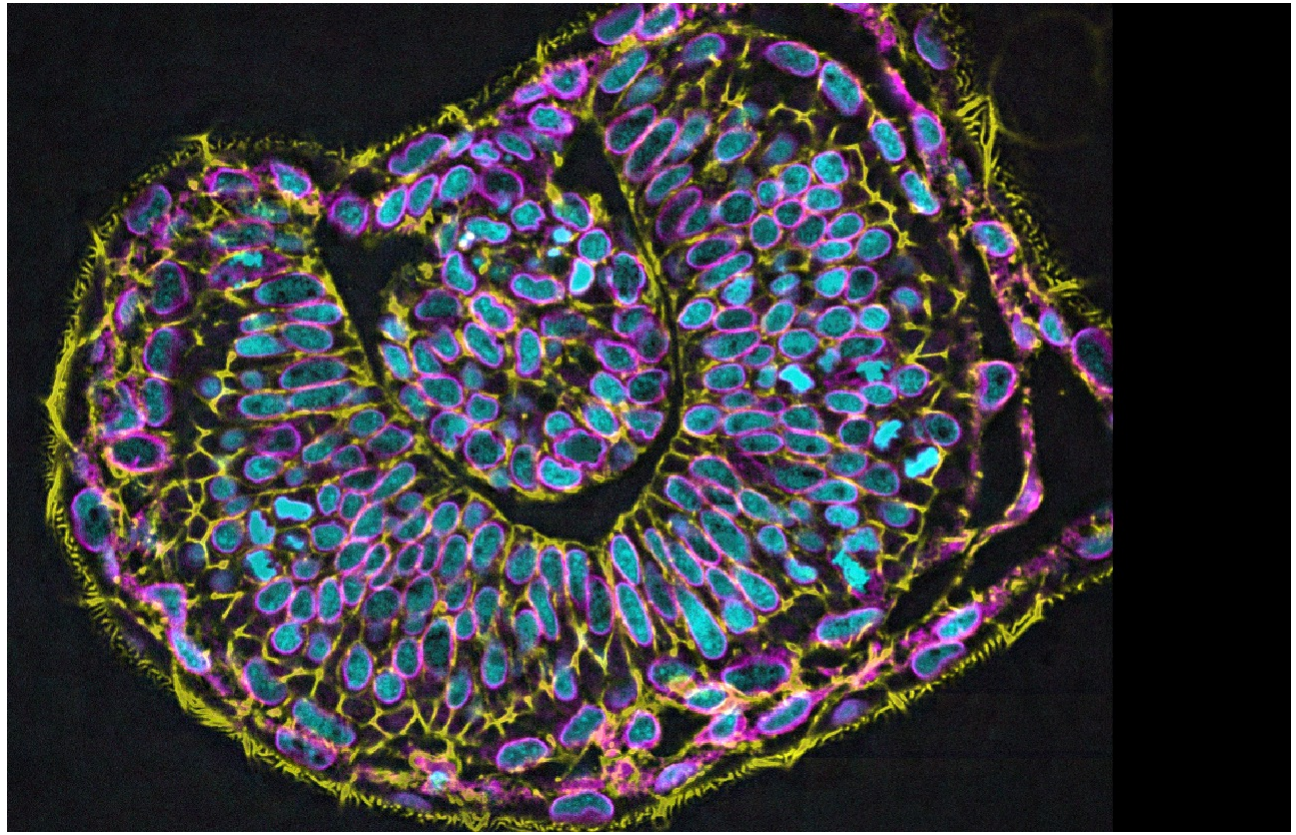
Compare conditions

Be done ☺

ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

More typical situation: We do not know about a feature

- We expect or know of a biological effect (e.g., through external cues, cell growth stages, etc.)
- We do not know how this effect can be measured or how it manifests itself



Source: Mauricio Rocha Martins, Norden lab, MP□CBG

Example:

Developing zebrafish eye

Hypothesis: Cells develop differently depending on where they are

Get Data

↓

Measure … what exactly?

↓

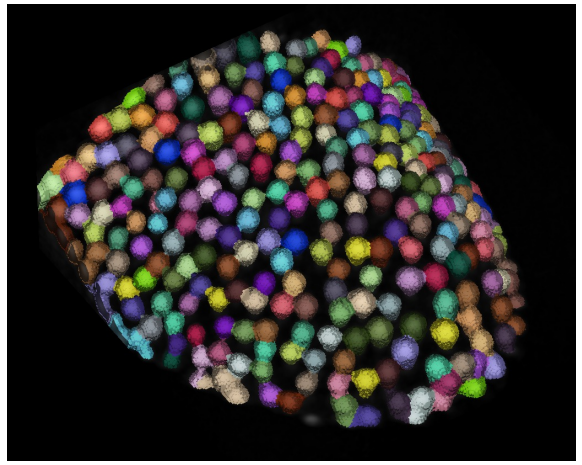
Compare … what exactly?

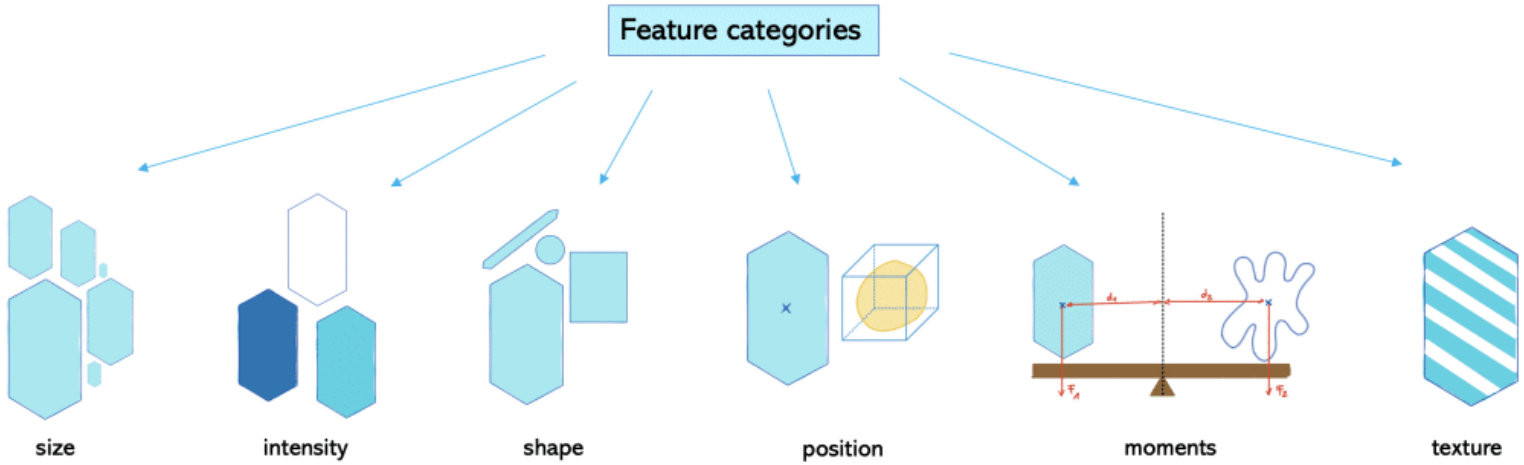
↓

Be stuck ☹

We can measure tons of features…

… but still have no idea about what's happening!





Source: Mara Lampert, *FocalPlane*, https://focalplane.biologists.com/2023/05/03/feature-extraction-in-napari/

Which of these features reflect interesting biology?

	label	area	bbox_area	convex_area	quivalent_diamete	max_intensity	mean_intensity	min_intensity	solidity	extent	eret_diameter_ma	local_centroid-0
1	1	3379	13949	5120	18.61786412639...	613.0	345.6717963894...	259.0	0.6599609375	0....	37.3496987939662	15.77952056821...
2	2	2319	7448	3491	16.42230229224...	421.0	297.8434670116...	240.0	0....	0....	38.65229618017...	4....
3	3	2304	14415	4281	16.38681751812...	456.0	300.8298611111...	245.0	0....	0....	34.19064199455...	17.73828125
4	4	3278	13804	5139	18.43048549951...	467.0	316.1446003660...	249.0	0....	0....	34.84250278036...	15.52287980475...
5	5	1501	3315	1681	14.20563625190...	458.0	302.147235176549	236.0	0....	0....	17.97220075561...	6....
6	6	2341	6061	2714	16.47407088948...	594.0	355.4446817599...	261.0	0....	0....	30.67572330035...	16.54250320375...
7	7	1725	3584	1940	14.87979081163...	568.0	343.7866666666...	257.0	0....	0....	17.72004514666...	7.80463768115942

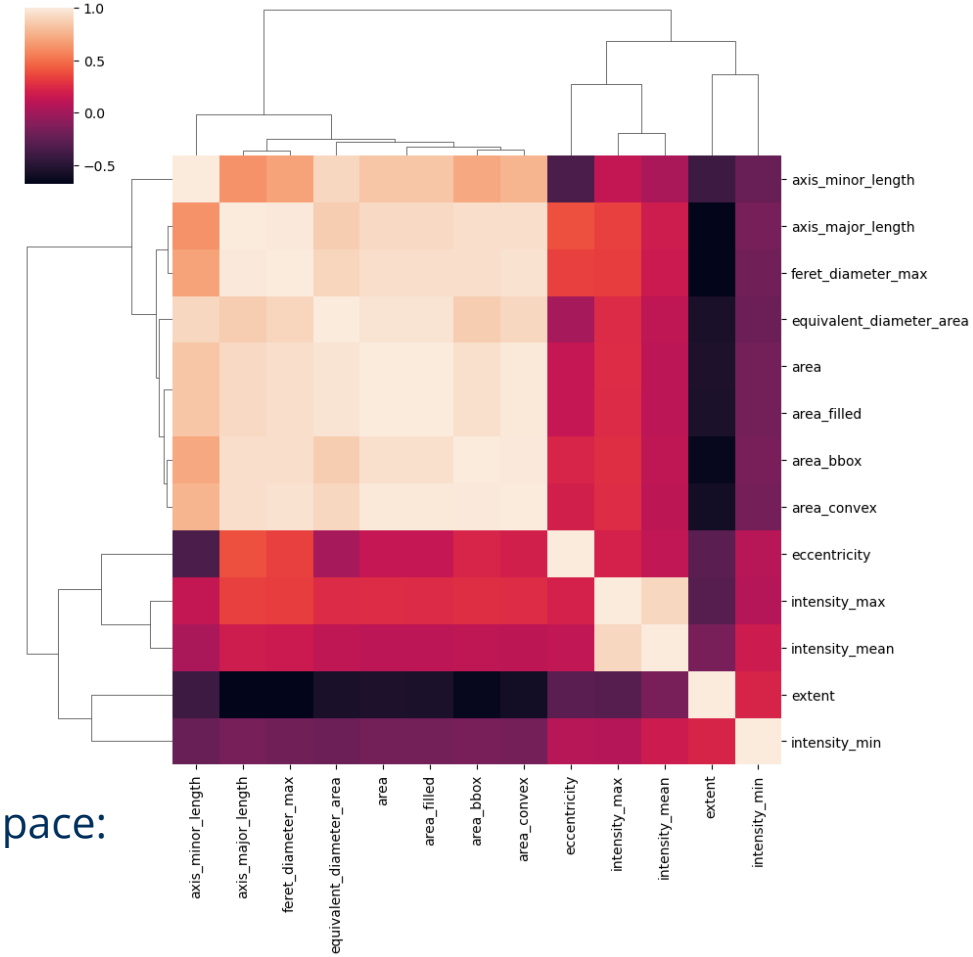
ScaDS.AI DRESDEN LEIPZIG · EKFZ Digital Health · TECHNISCHE UNIVERSITÄT DRESDEN

Identify the feature with the strongest effect

We could plot all features against our data and check which feature shows the strongest effect

But this would lead to following challenges:

- Features are not independent!
  - Area and diameter
  - Width and height
- A lot of redundant information
- Strongest effect might be a combination of features
- Risk of misinterpretation



- Need fewer and independent features
- Need to transform parameter space into lower dimensional space:

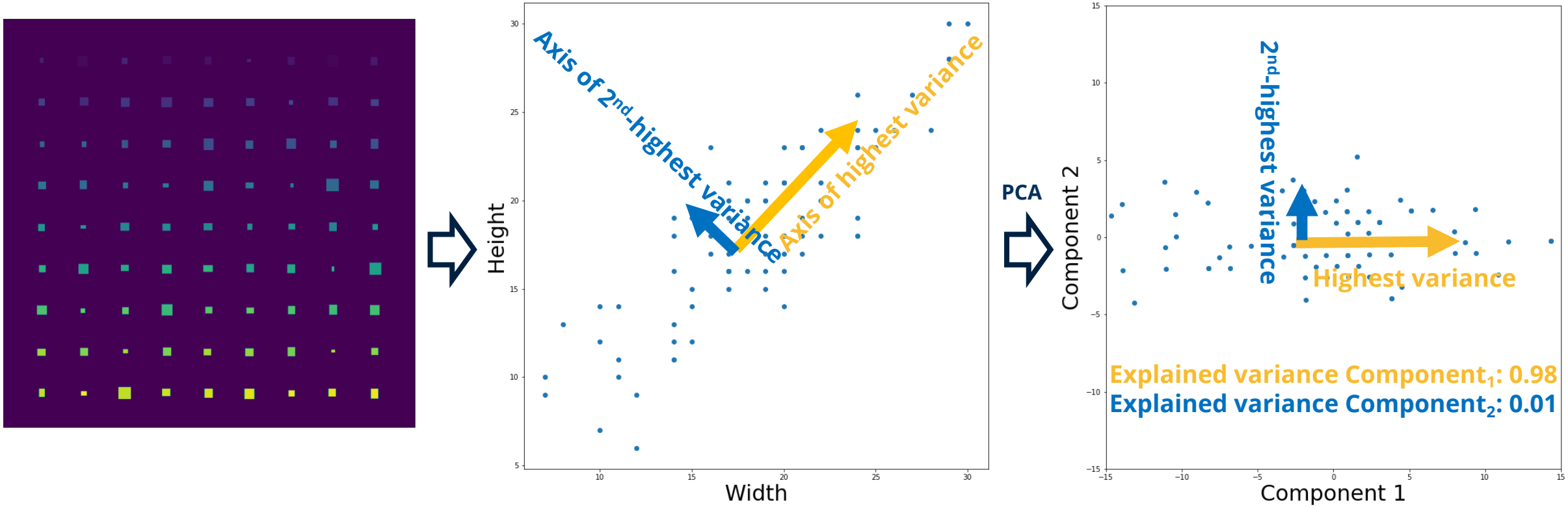
➢ **Matrix factorization methods**
  - Principal component analysis (PCA)

➢ **Neighbor Graphs**
  - t-Distributed Stochastic Neighbour Embedding (t-SNE)
  - Uniform Manifold Approximation and Projection (UMAP)

PCA: Principal Component Analysis

Decomposes data into linear combinations of features that explain the highest variance

Example: Squares of different size



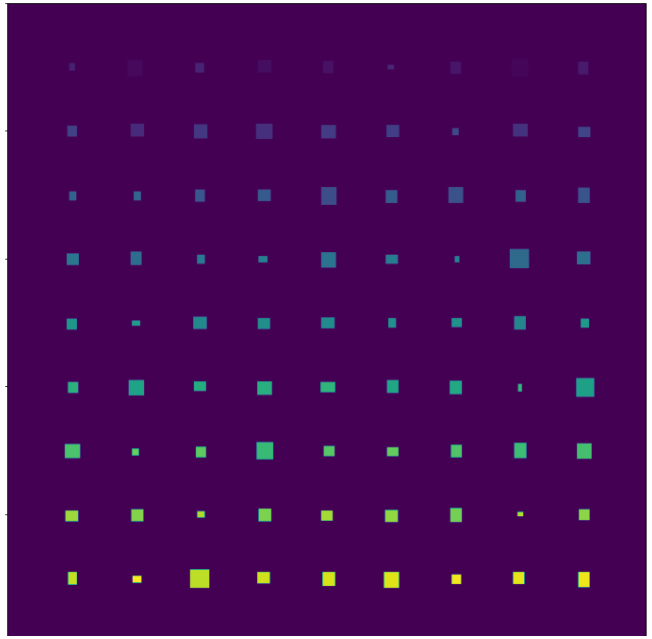
→ PCA transforms width/height measurements into a coordinate system that explains existing variance better

ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

PCA: Principal Component Analysis

Decomposes data into linear combinations of features that explain the highest variance

Example: Squares of different size



Step 1: Standardization

Case 1
Heights: 0 … 30
Widths: 0 … 30

Case 2
Area: 0 … 100
Circularity: 0 … 1

$$z = \frac{value - mean}{standard\ deviation}$$

Step 2: Covariance matrix computation to identify correlations

$$\begin{bmatrix} Cov(height, height) & Cov(height, width) \\ Cov(width, height) & Cov(width, width) \end{bmatrix}$$

Variances

+ *Cov* → variables correlated
- *Cov* → inversely correlated
= 0 → variables are independent

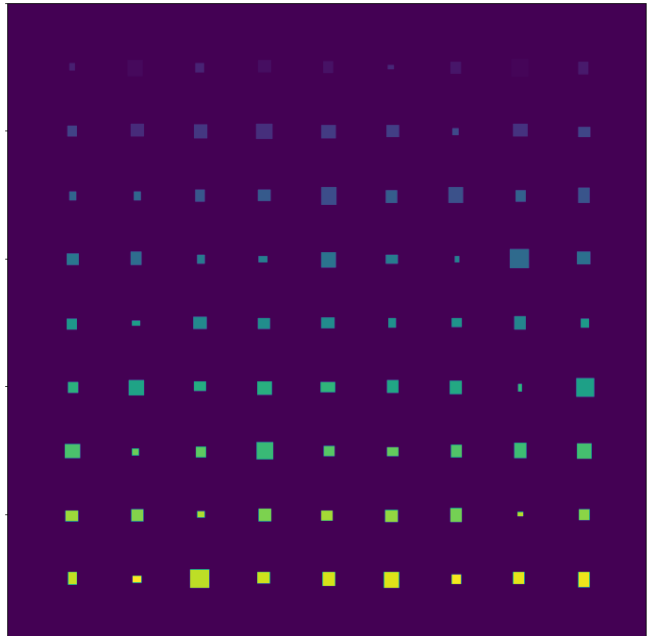
$$Cov(height, width) = \langle h \cdot w \rangle - \langle h \rangle \cdot \langle w \rangle = \left(\frac{1}{N}\sum_{i=0}^{N-1} h_i w_i\right) - \left(\frac{1}{N}\sum_{i=0}^{N-1} h_i\right)\left(\frac{1}{N}\sum_{i=0}^{N-1} w_i\right)$$

N – number of data points

PCA: Principal Component Analysis

Decomposes data into linear combinations of features that explain the highest variance

Example: Squares of different size



Step 3: Calculation of Eigenvectors and Eigenvalues

$\det(C - \lambda I) = 0 \qquad I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$

C – covariance matrix
λ - eigenvalues
I – identity matrix
v - eigenvectors

$$\det(C - \lambda I) = det \begin{bmatrix} Cov(height, height) - \lambda & Cov(height, width) \\ Cov(width, height) & Cov(width, width) - \lambda \end{bmatrix} = 0$$

→ Solve equation for eigenvalues (λ)

→ Find eigenvectors (v) by substituting each eigenvalue in $(C - \lambda I)\mathrm{v} = 0$

$$v_1 = \begin{bmatrix} v_{1,height} \\ v_{1,width} \end{bmatrix} \qquad v_2 = \begin{bmatrix} v_{2,height} \\ v_{2,width} \end{bmatrix}$$

PCA: Principal Component Analysis

Decomposes data into linear combinations of features that explain the highest variance

Example: Squares of different size

Step 4: Transformation to a new coordinate system

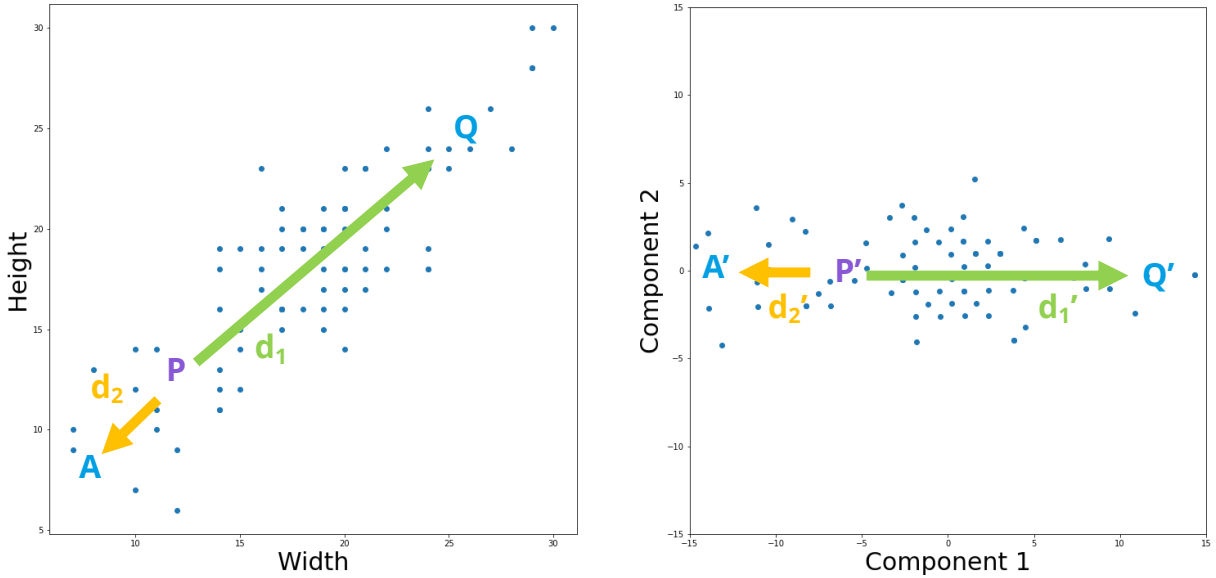
$$Transformation\ matrix\ T = \begin{bmatrix} v_{1,height} & v_{2,height} \\ v_{1,width} & v_{2,width} \end{bmatrix}$$

$$T \cdot \begin{pmatrix} width \\ height \end{pmatrix} = \begin{pmatrix} component\ 1 \\ component\ 2 \end{pmatrix}$$

This is a linear operation!
Metrics remain meaningful

This works for any number of features!

$$T \cdot \begin{pmatrix} feature\ 1 \\ \dots \\ feature\ N \end{pmatrix} = \begin{pmatrix} component\ 1 \\ component\ 2 \end{pmatrix}$$



$d_1 > d_2 \rightarrow d_1' > d_2'$

Relative distances are preserved!

Great visualization tool for learning PCA:
https://setosa.io/ev/principal-component-analysis/ by Victor Powell

ScaDS.AI DRESDEN LEIPZIG · EKFZ Digital Health · TECHNISCHE UNIVERSITÄT DRESDEN

PCA in Python: `sklearn.decomposition.PCA`

- Import package

```
from sklearn.decomposition import PCA
```

- Apply PCA

```
pca = PCA(n_components=2)
pca.fit(standardized_data)
```

- Transform data into new coordinate system

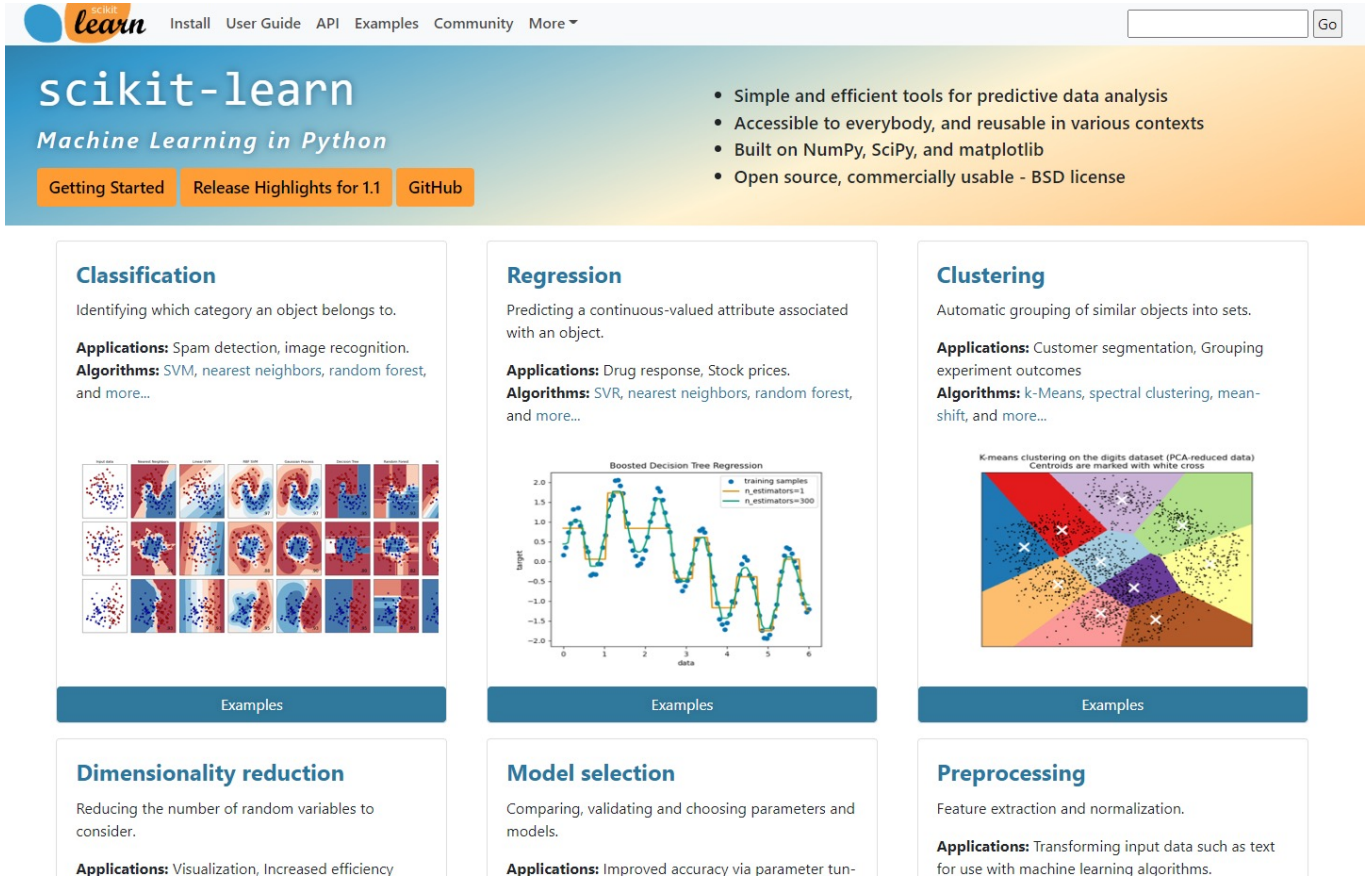
```
transformed_data = pca.transform(data)
```

Important!

Always check the explained variance along the PCA component axes!

```
pca.explained_variance_ratio_
```

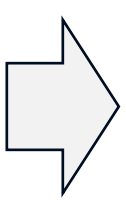
```
array([0.98773142, 0.01226858])
```

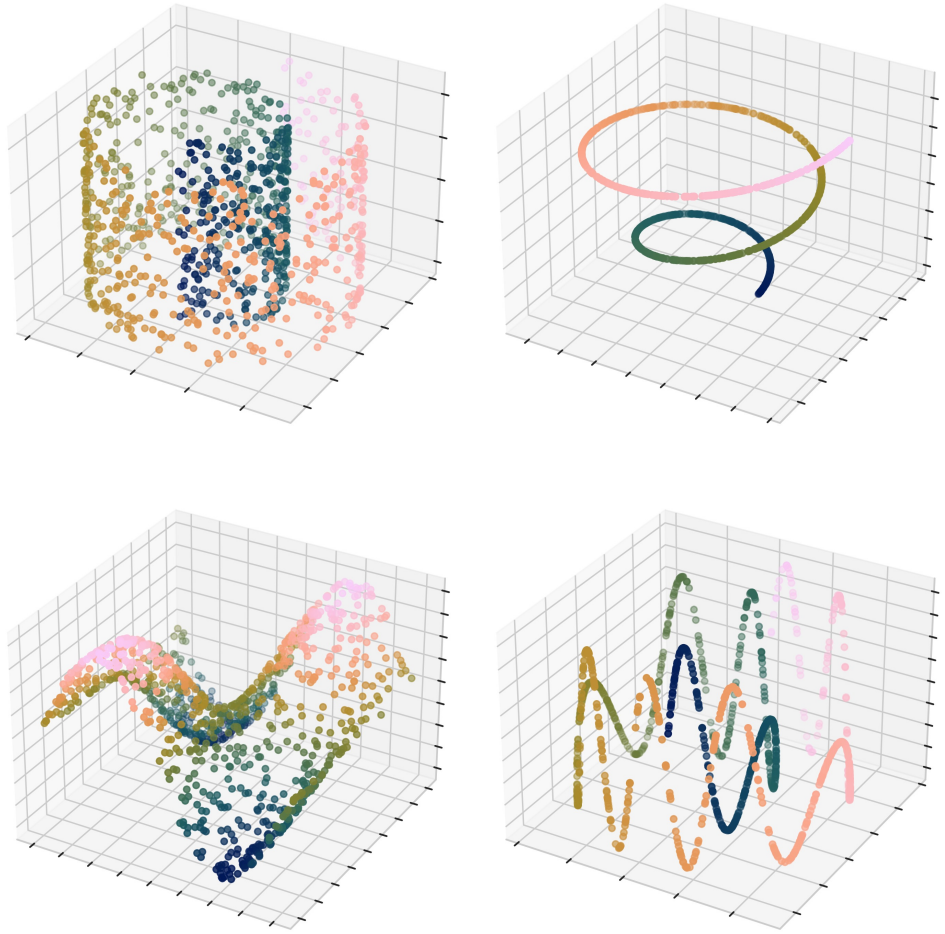


Disadvantages of PCA

- Sensitive to the scaling of the variables and outliers
- Linear algorithm → cannot represent complex relationships between features
- Loss of information

Need an algorithm to deal with linearly non separable data





Recap: Euclidean space

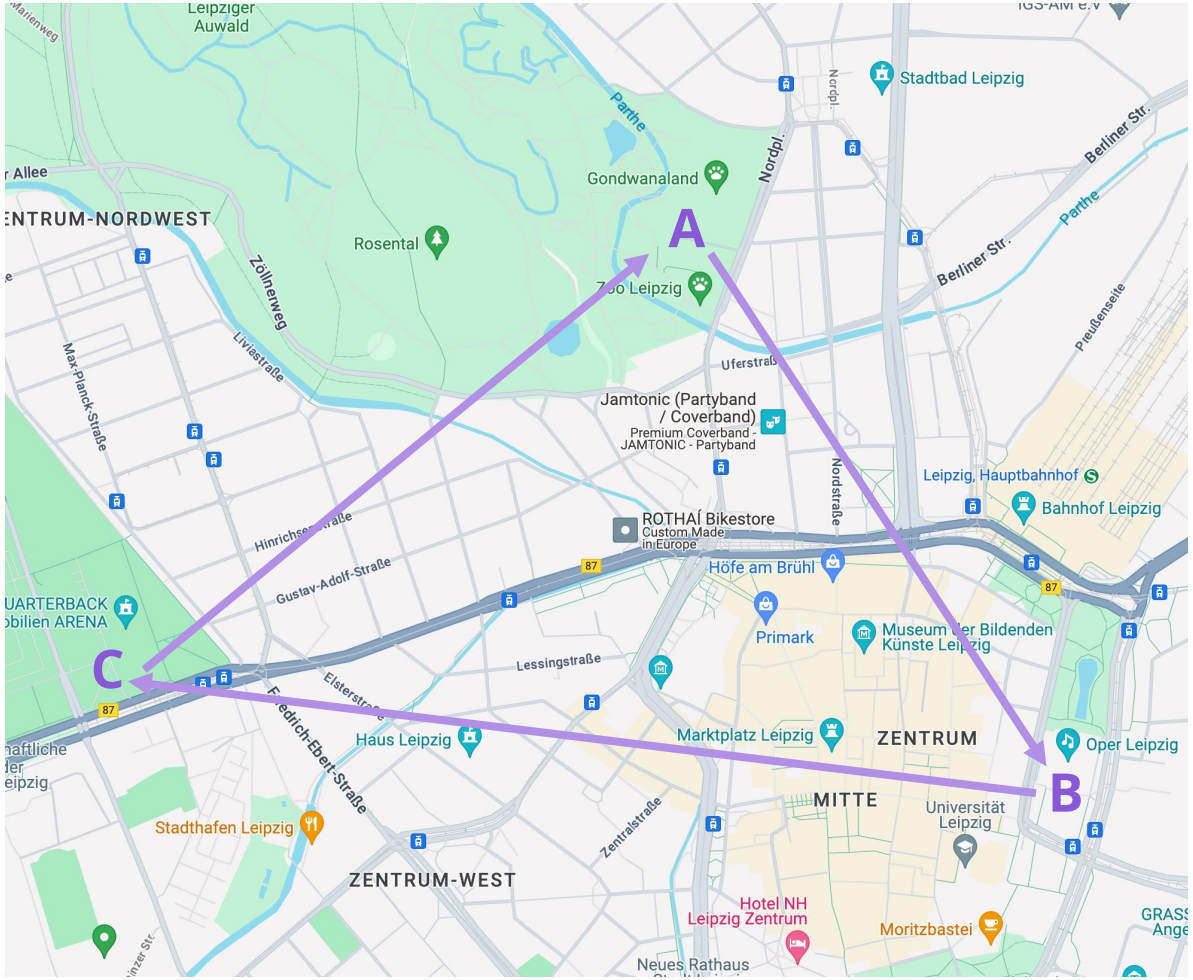
Characteristics:

- Distance between **A** and **B** is symmetric:
  - $d(A, B) = d(B, A)$
- Distance between **A** and **B** can be measured as the length (“norm”) of the vector $\overrightarrow{AB}$
- Distances satisfy the triangle inequality:

$$d(\mathrm{B}, \mathrm{C}) \leq d(\mathrm{C}, A) + d(\mathrm{A}, \mathrm{B})$$

In other words: there is no shorter path between two points other than a straight line

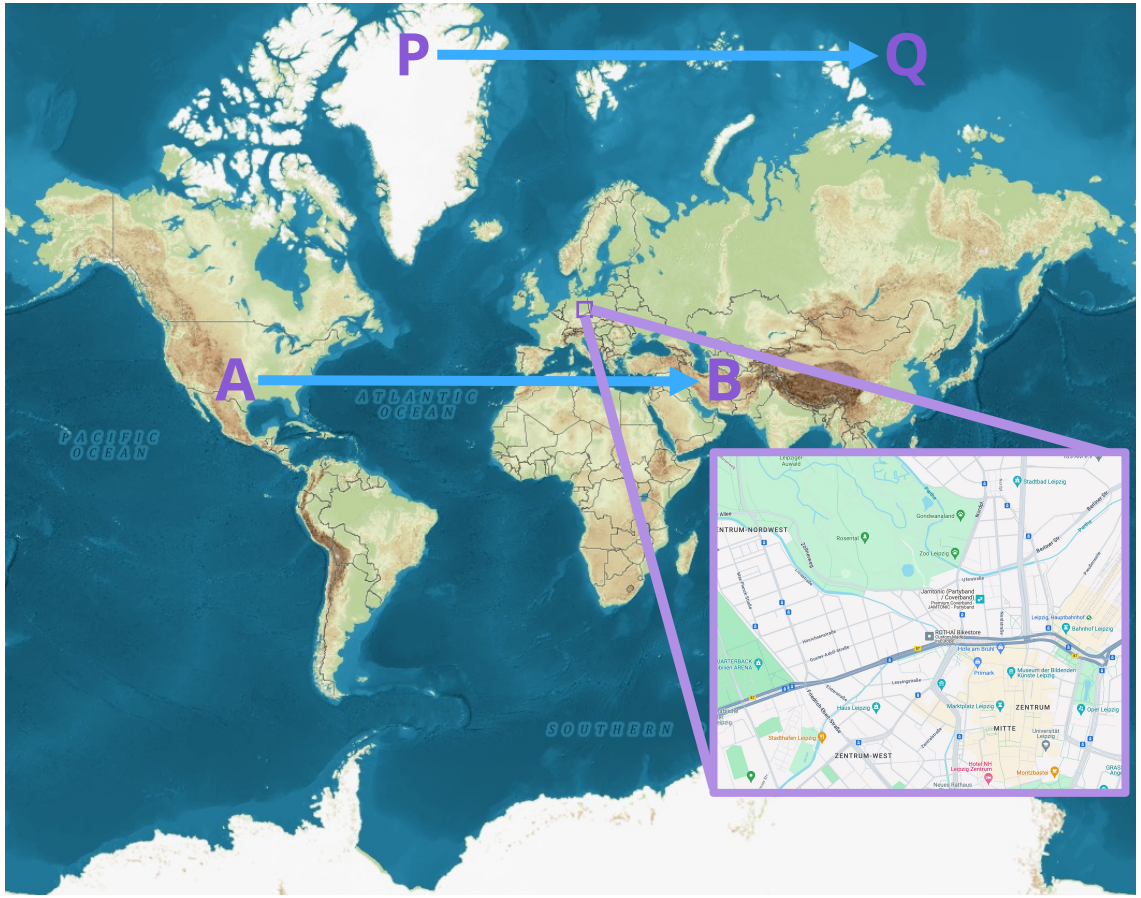
Example: (local) 2D space



Source: Maps.google.de
License: https://about.google/brand-resource-center/products-and-services/geo-guidelines/#google-maps

More complex concept: Manifolds

From Wikipedia: *"In mathematics, a manifold is a topological space that locally resembles Euclidean space near each point."*



Mapy.cz, https://licence.mapy.cz/?doc=mapy_pu

...This map is non-Euclidean!

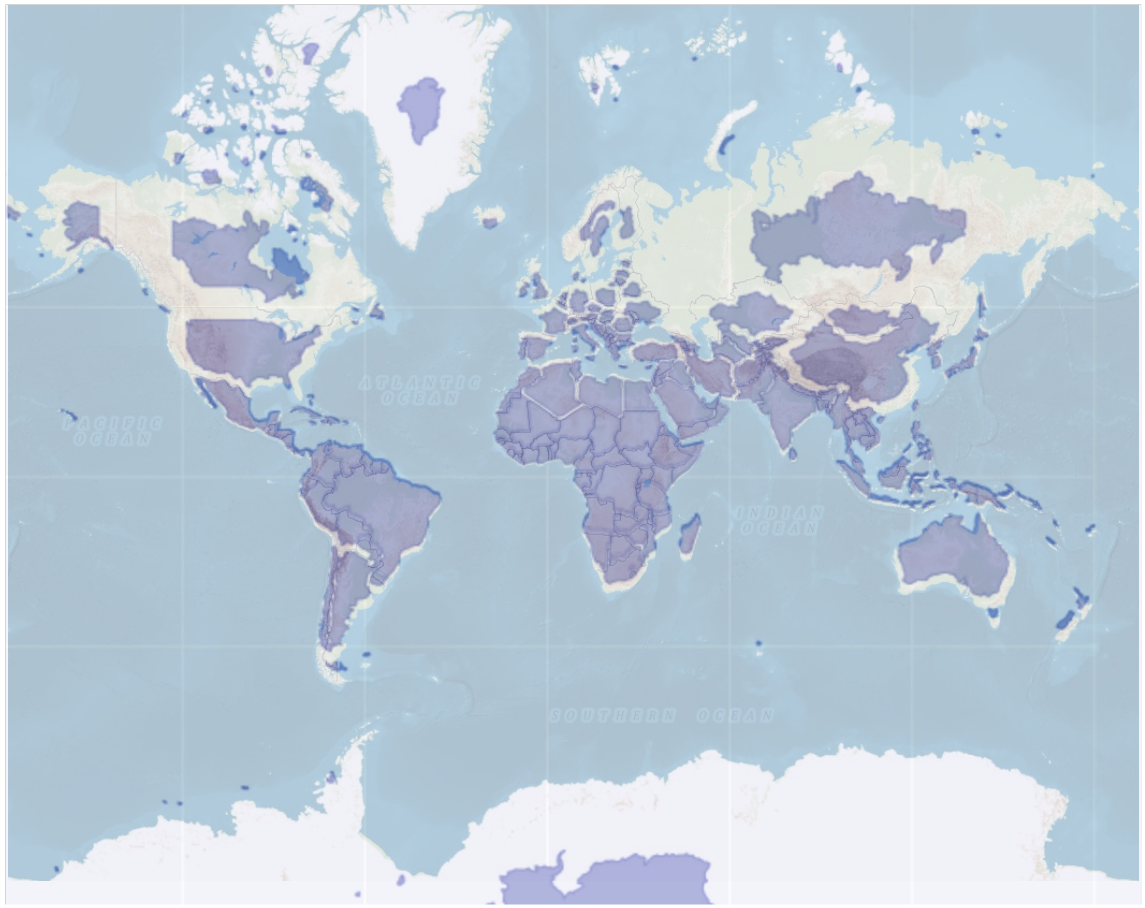
→ *The two vectors $\overrightarrow{PQ}$ and $\overrightarrow{AB}$ have the same length, but the real distances (the norm) of both are completely different!*

→ *Cropping a small piece from the map gives us a local Euclidean space, where the previous assumptions hold.*

ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

More complex concept: Manifolds

From Wikipedia: *"In mathematics, a manifold is a topological space that locally resembles Euclidean space near each point."*



Mapy.cz, https://licence.mapy.cz/?doc=mapy_pu

...This map is non-Euclidean!

→ *The two vectors $\overrightarrow{PQ}$ and $\overrightarrow{AB}$ have the same length, but the real distances (the norm) of both are completely different!*

→ *Cropping a small piece from the map gives us a local Euclidean space, where the previous assumptions hold.*

Approximately true country sizes
Source: Jakub Nowosad
(CC BY-SA 4.0)

ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

t-Distributed Stochastic Neighbour Embedding (t-SNE)

Reduce dimensionality preserving local structure (neighbours)

- Find a manifold that represents the data in fewer dimensions → ability to visualize the data
- Preserve local neighbours at the expense of distance distortions

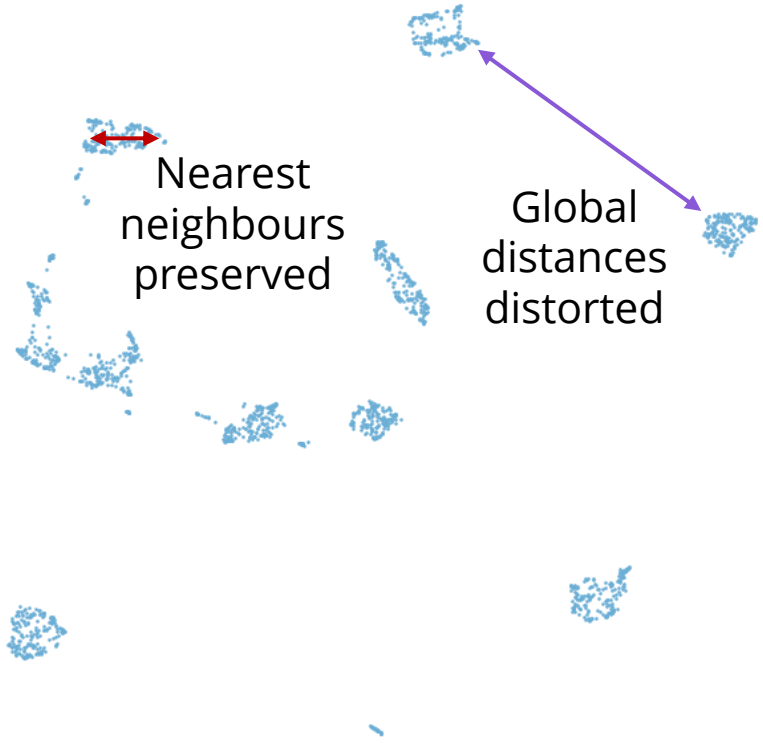
Many dimensions

Properties of Result of voronoi otsu labeling (clesperant

Copy to clipboard | Save as csv...

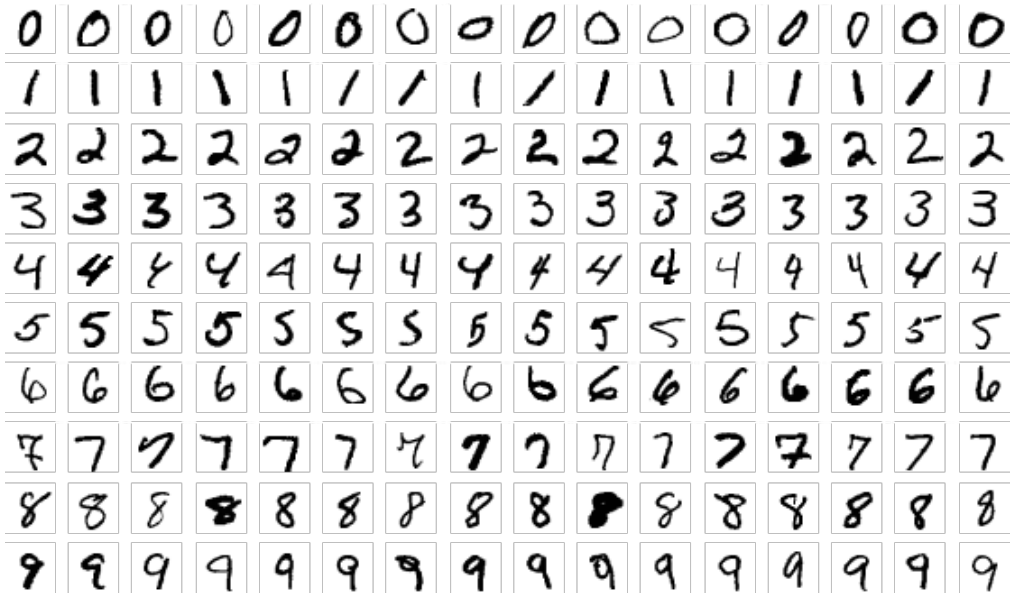
	label	area	bbox_area	convex_area	quivalent_diamete	max_intensity	mean_intensity	min_intensity	solidity	extent	eret_diameter_ma	local_centroid-0	l
1	1	3379	13949	5120	18.61786412639...	613.0	345.6717963894...	259.0	0.6599609375	0....	37.3496987939662	15.77952056821...	18
2	2	2319	7448	3491	16.42230229224...	421.0	297.8434670116...	240.0	0....	0....	38.65229618017...	4....	17
3	3	2304	14415	4281	16.38681751812...	456.0	300.8298611111...	245.0	0....	0....	34.19064199455...	17.73828125	13
4	4	3278	13804	5139	18.43048549951...	467.0	316.1446003660...	249.0	0....	0....	34.84250278036...	15.52287980475...	10
5	5	1501	3315	1681	14.20563625190...	458.0	302.147235176549	236.0	0....	0....	17.97220075561...	6....	6.
6	6	2341	6061	2714	16.47407088948...	594.0	355.4446817599...	261.0	0....	0....	30.67572330035...	16.54250320375...	6.
7	7	1725	3584	1940	14.87979081163...	568.0	343.7866666666...	257.0	0....	0....	17.72004514666...	7.804637681159442	7.
8	8	1502	3840	1753	14.20879025650...	431.0	290.0659121171...	235.0	0....	0....	18.57417562100...	8....	6.
9	9	1602	4080	1894	14.51737058294...	475.0	297.8008739076...	241.0	0....	0....	18.70828693386...	8....	8.
10	10	1395	3600	1624	13.86304166283...	424.0	304.8494623655...	247.0	0....	0.3875	17.60681686165...	7....	7.
11	11	609	1100	697	10.51654029260...	323.0	274.2528735632...	241.0	0....	0....	13.45362404707...	3....	4.
12	12	1686	3757	1894	14.76679738567...	460.0	303.8303677342...	240.0	0....	0....	17.97220075561...	9....	7.
13	13	2157	5184	2531	16.03062694504...	576.0	339.990264255911	270.0	0....	0....	19.54482028569...	8....	8.
14	14	863	2340	1032	11.81237949737...	327.0	272.4449594438...	237.0	0....	0....	16.0312195418814	6....	5.

Reduced space to 2 dimensions

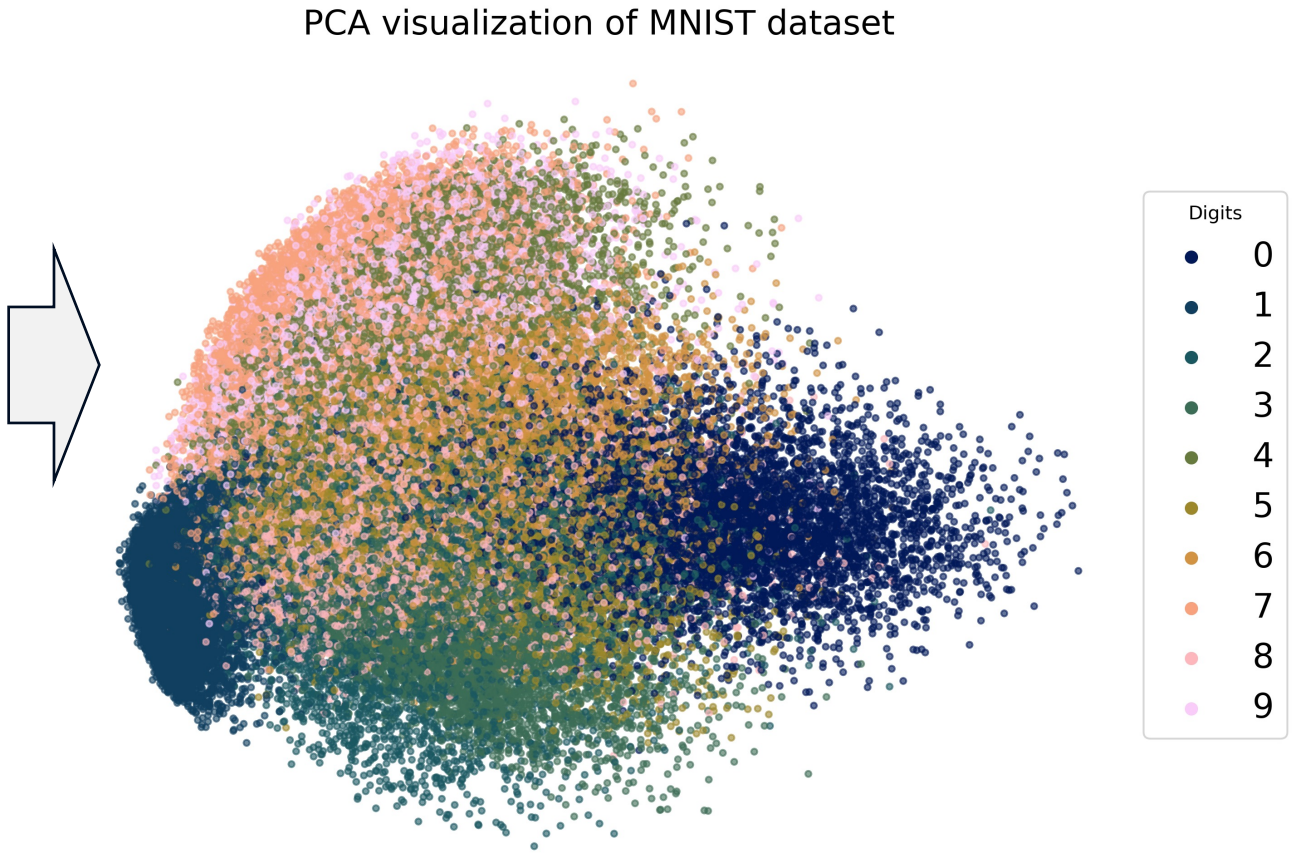


ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

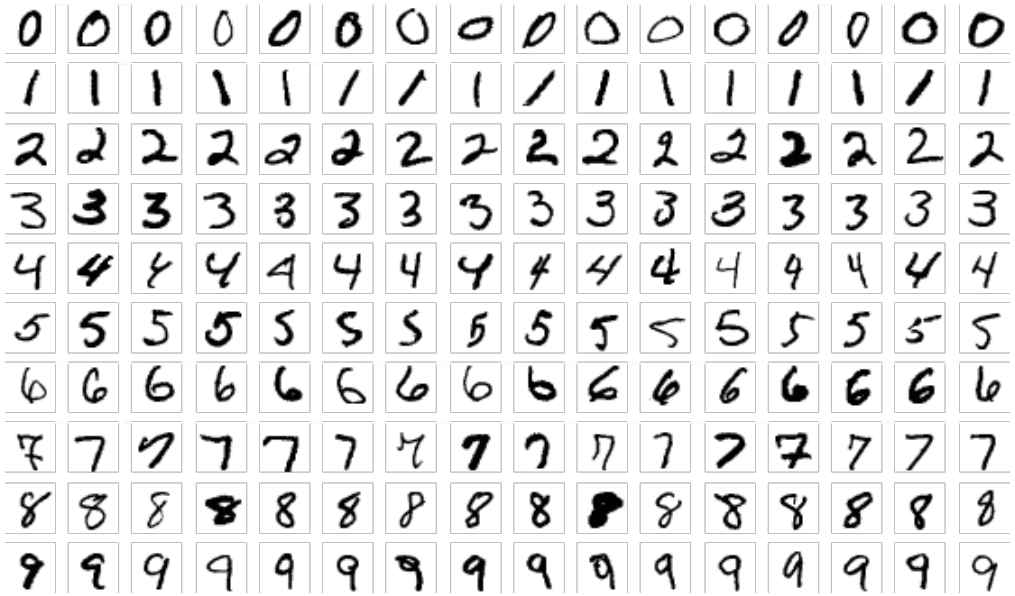
t-Distributed Stochastic Neighbour Embedding (t-SNE)



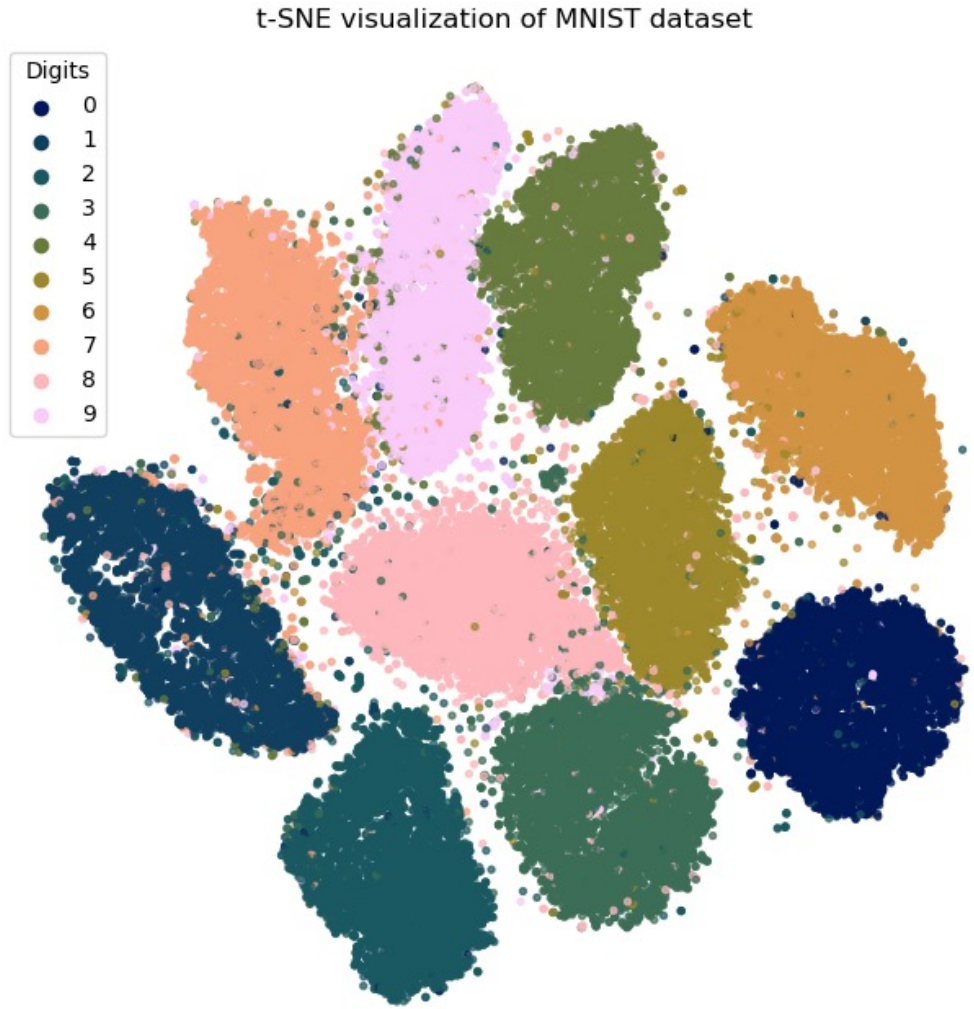
Source: Wikipedia (CC BY-SA 4.0)



t-Distributed Stochastic Neighbour Embedding (t-SNE)



Source: Wikipedia (CC BY-SA 4.0)



ScaDS.AI DRESDEN LEIPZIG

EKFZ Digital Health

TECHNISCHE UNIVERSITÄT DRESDEN

t-Distributed Stochastic Neighbour Embedding (t-SNE)

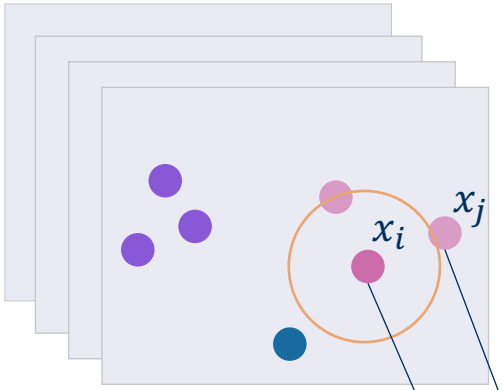
- Loss function – **Kullback-Leibner divergence ($\mathcal{L}$)** between pairwise similarities (affinities) in the high-dimensional and in the low-dimensional spaces. Similarities are defined such that they sum to 1.
- **High price for putting close neighbors far away.**

p – **High dimensional similarities**

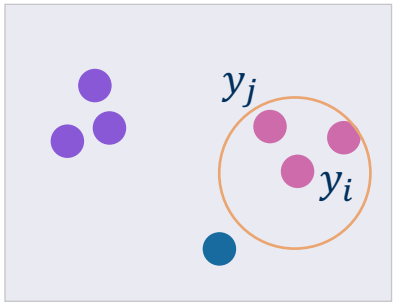
q – Low dimensional similarities

$$\mathcal{L} = \sum_{i,j} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

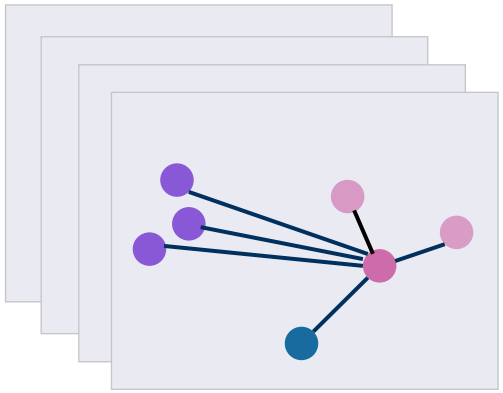
High dimensionality



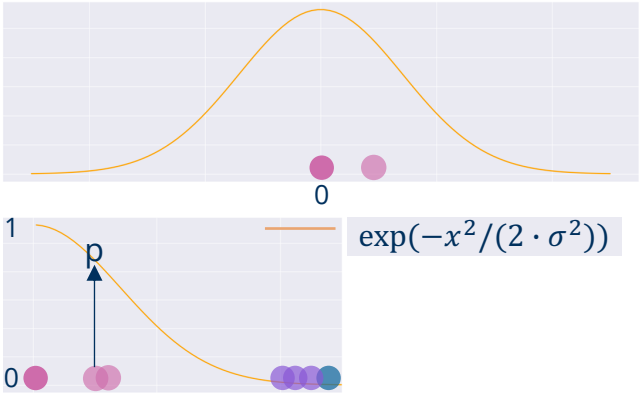
Low dimensionality



High dimensionality



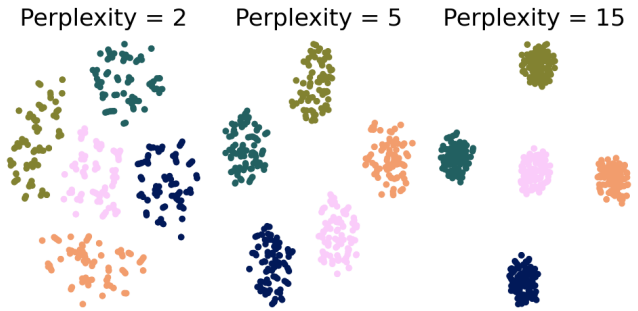
The larger the distance, the smaller $p_{j|i}$



$$p_{j|i} = \frac{\exp(-\left|\left|x_i - x_j\right|\right|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\left|\left|x_i - x_k\right|\right|^2 / 2\sigma_i^2)}$$

Euclidean distance (or any other)

Kernel width is adaptively chosen adaptively to achieve the desired ***perplexity***



Hinton et al. ***Stochastic neighbor embedding****, 2002*
Maaten et al. ***Visualizing data using t-SNE****, 2008*

ScaDS.AI DRESDEN LEIPZIG · EKFZ Digital Health · TECHNISCHE UNIVERSITÄT DRESDEN

t-Distributed Stochastic Neighbour Embedding (t-SNE)

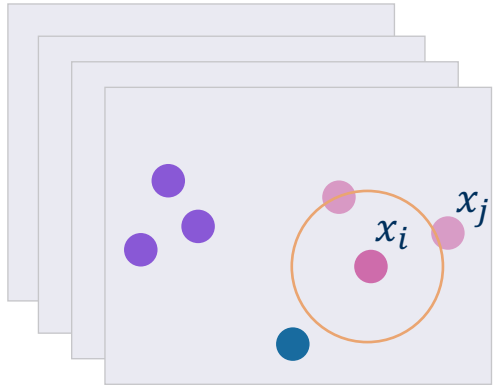
- Loss function – **Kullback-Leibner divergence ($\mathcal{L}$)** between pairwise similarities (affinities) in the high-dimensional and in the low-dimensional spaces. Similarities are defined such that they sum to 1.
- **High price for putting close neighbors far away.**

p – High dimensional similarities

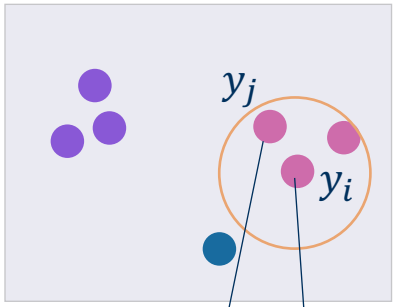
q – **Low dimensional similarities**

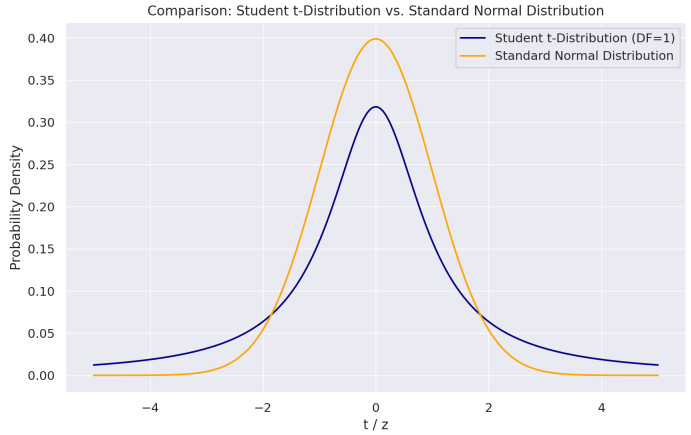
$$\mathcal{L} = \sum_{i,j} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

High dimensionality



Low dimensionality





$$k(d) = \exp(-d^2)$$

$$k(d) = \frac{1}{1+d^2}$$

$$q_{ij} = \frac{w_{ij}}{Z}$$

$$w_{ij} = k(\left|\left|y_i - y_j\right|\right|)$$

All pairwise distances over the entire dataset: $Z = \sum_{k \neq l} w_{kl}$

Similarity kernel (k):

- In the original SNE paper **Gaussian kernel**
- t-SNE paper introduced **Student t-distribution** with 1 degree of freedom (heavy tails) to avoid crowding issue

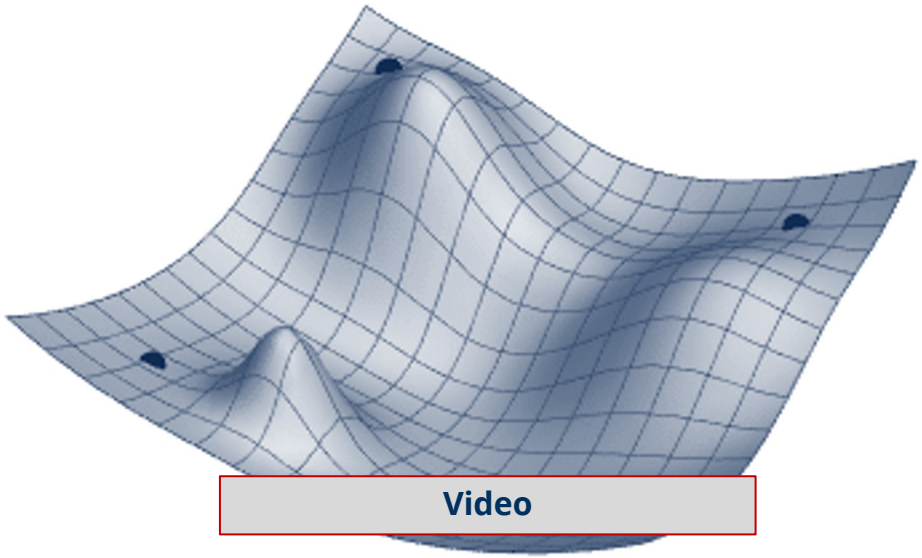
Hinton et al. ***Stochastic neighbor embedding****, 2002*
Maaten et al. ***Visualizing data using t-SNE****, 2008*

ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

t-Distributed Stochastic Neighbour Embedding (t-SNE)

Gradient Descent for Optimizing the Loss

- Starting from a **random** configuration of the same number of points in low dimensional space
- Close neighbours **attract** each other while all points **repulse** each other → High price for putting close neighbors far away
- Attraction-repulsion forces are computed for each data point, and a small step is made in the direction of this gradient (that is, you move all the points), and then the gradient is recomputed



$$\mathcal{L} = \sum_{i,j} \boldsymbol{p_{ij}} \boldsymbol{log} \frac{\boldsymbol{p_{ij}}}{\boldsymbol{q_{ij}}} = \overbrace{\sum_{i,j} p_{ij} \cdot \log(p_{ij})}^{\text{Constant}} - \sum_{i,j} p_{ij} \cdot \log(q_{ij}) = -\sum_{i,j} p_{ij} log \frac{w_{ij}}{Z} = -\sum_{i,j} p_{ij} \log(w_{ij}) + \underbrace{\sum_{ij} p_{ij}}_{=1} logZ$$

Derivative is taken to compute the gradient

p – High dimensional similarities

q – Low dimensional similarities

"Attraction of neighbors": the distance should be as small as possible in the low-dim space

"Repulsion" between all the pairs for balance

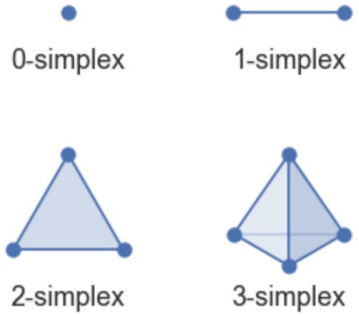
ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

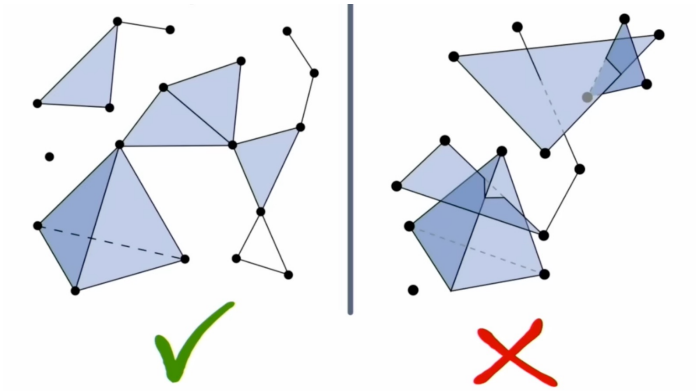
Uniform Manifold Approximation and Projection (UMAP)

Advantages over t-SNE:

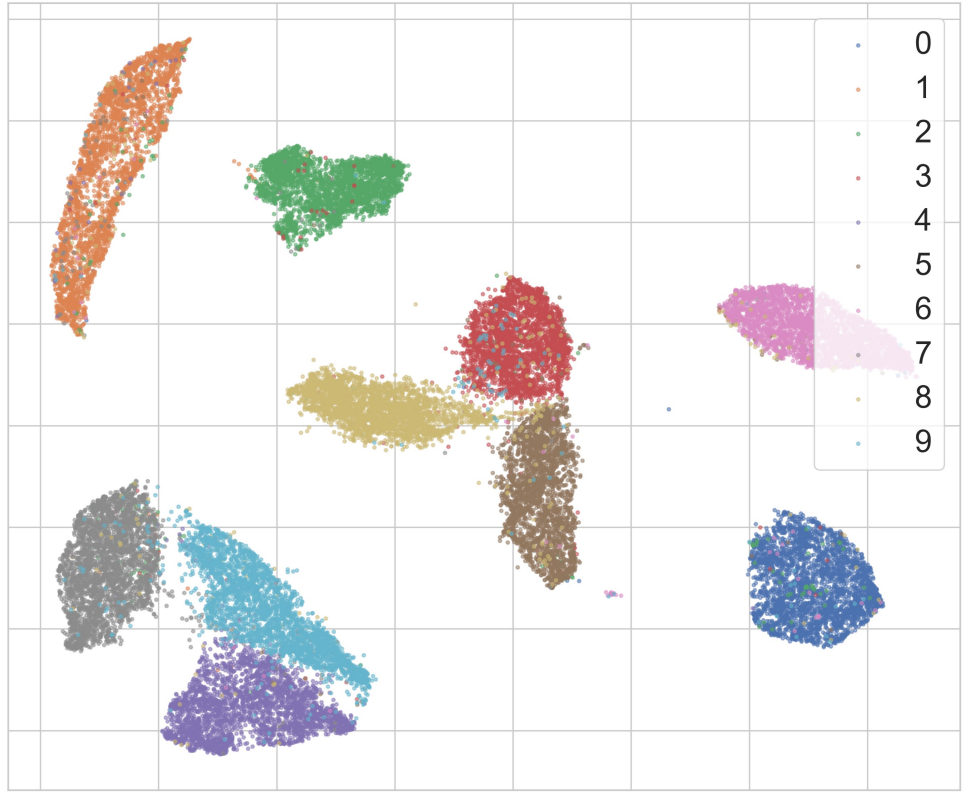
- Increased speed (projection of 70 000 point MNIST dataset < 3 minutes in comparison to 45 min for *scikit-learn's* t-SNE)
- Scales well in terms of both dataset and dimensionality
- Better preservation of the data's global structures
- Builds mathematical theory to justify the graph based approach

UMAP constructs a high dimensional graph representation of the data then optimizes a low-dimensional graph to be as structurally similar as possible.





UMAP projection of MNIST dataset



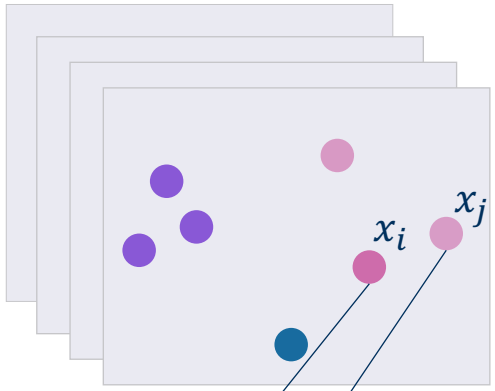
https://umap-learn.readthedocs.io/en/latest/how_umap_works.html
Interesting UMAPs https://johnhw.github.io/umap_primes/index.md.html

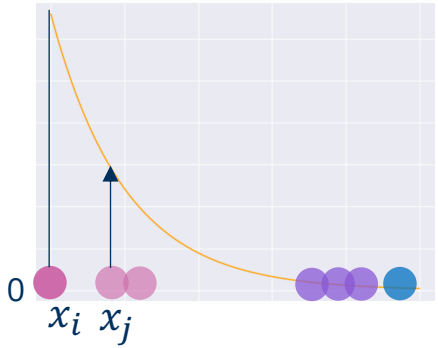
ScaDS.AI DRESDEN LEIPZIG
EKFZ Digital Health
TECHNISCHE UNIVERSITÄT DRESDEN

Uniform Manifold Approximation and Projection (UMAP)

- The data suggests an underlying structure (“topology”) but we do not have a model for it.
- UMAP constructs a high dimensional graph representation of the data then optimizes a low-dimensional graph to be as structurally similar as possible.

High dimensionality





Similarity score

$$p_{i|j} = \exp((-\left|\left|x_i - x_j\right|\right|^2 - \rho_i)/\sigma_i)$$

$log_2(\boldsymbol{nr\ of\ neighbors}) = log_2 3 = 1.6$

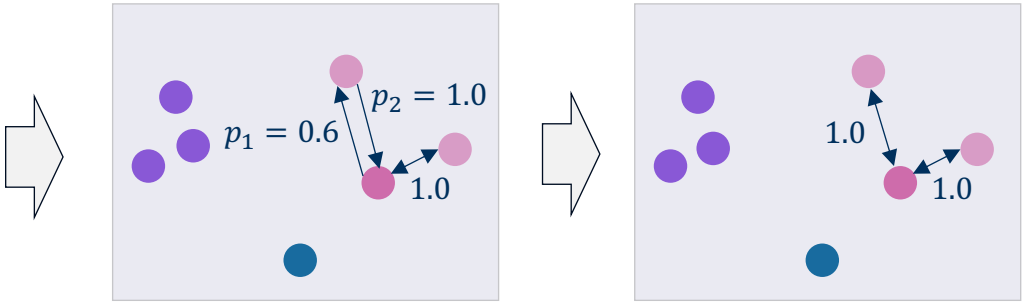
Distance to the nearest neighbor

Sum of similarity scores

Large number of neighbors → Global structure preserved
Small number of neighbors → Local structure preserved

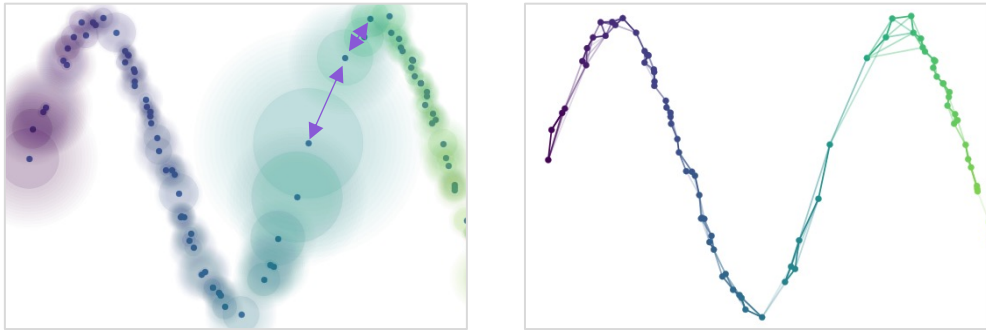
Scores are made symmetrical:
Fuzzy union operation

$$\boldsymbol{Symmetrical\ score} = (p_1 + p_2) - p_1 p_2$$



Both points have the **same similarity scores** even though one is further one due to the **theoretical framework of UMAP (topology and fuzzy sets)**

Same distances!



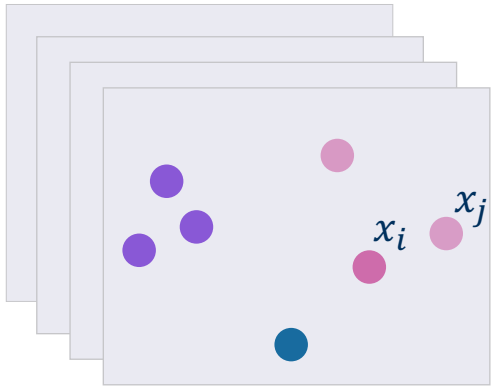
https://umap-learn.readthedocs.io/en/latest/how_umap_works.html

ScaDS.AI DRESDEN LEIPZIG | EKFZ Digital Health | TECHNISCHE UNIVERSITÄT DRESDEN

Uniform Manifold Approximation and Projection (UMAP)

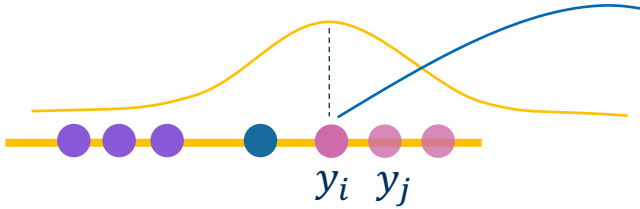
Goal: to optimize the low dimensional representation to have as close a fuzzy topological representation as possible as measured by **binary cross entropy** via the **stochastic gradient descent**

High dimensionality



Low dimensionality

Initialization of a low-dimensional graph using **Spectral Embedding**



Known weights of edges from high dimensional manifold approximation

"Repultion" of distant points

$$CE(y_i, v) = \sum_j \left(-p_{i|j} log\, v(d_{ij})\right) + \left(1 - p_{i|j}\right) log(1 - v(d_{ij}))$$

$$where\ d_{ij} = y_i - y_j$$

"Attraction" of neighbors:
the distance should be as small as possible in the low-dim space

Low dim similarity score

$$v(d_{ij}) = \frac{1}{1 + \alpha(y_i - y_j)^{2\beta}}$$

Default: $\alpha = 1.577$; $\beta = 0.8951$

NOTE: In contrast to t-SNE, which uses **random** initialization (unless other initialization is chosen, e.g. with PCA), resulting low-dimensional graph is the same each run on the same dataset

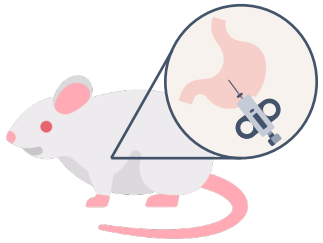
NOTE: if $\alpha = 1\ and\ \beta = 1$
then low-dim scores are equal to the ones that **t-SNE** uses
→ UMAP gives more control how tightly packed low-dim space ends up

Things to Consider

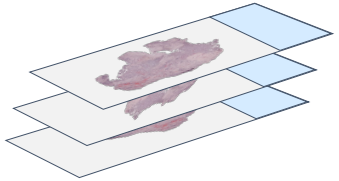
- Many parameters invite to “adjust” the data analysis, and final results depend a lot on hyperparameters
- Danger to over-interpret the visual “distance”: distances between clusters might not mean anything
- How much data structure is preserved is still a matter of debate
- Random noise might not always look random
- Cluster size might not mean anything

Dimensionality Reduction for Whole Slide Images

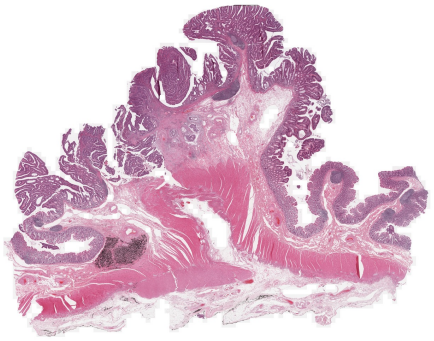
Biopsy/resection



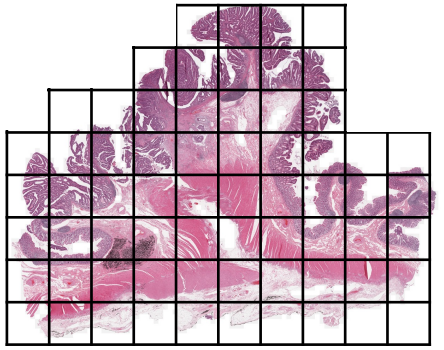
Microscope slide

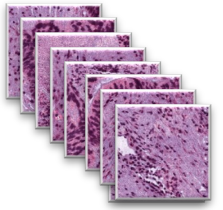


Whole slide image

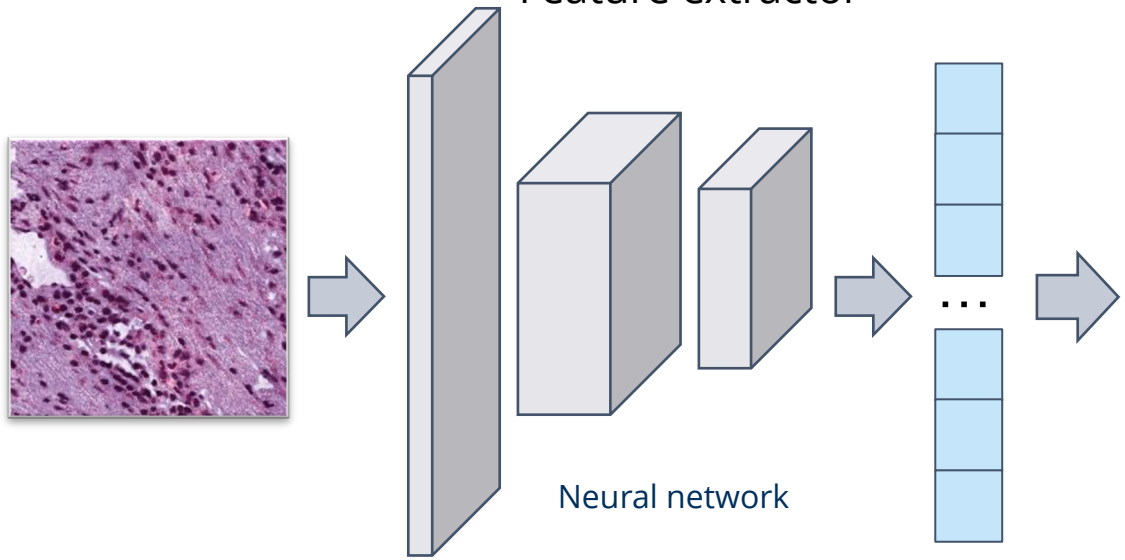


Tessellation

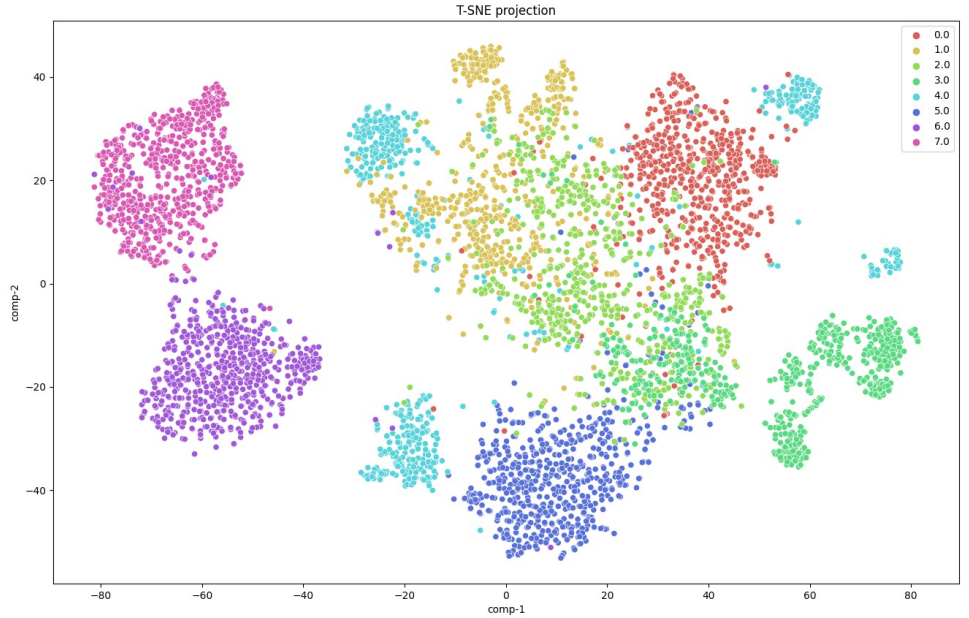




Feature extractor



Neural network



UMAP source: Didem Çifçi, Kather lab

Whole Slide Image Source: The Cancer Genome Atlas, National Cancer Institute

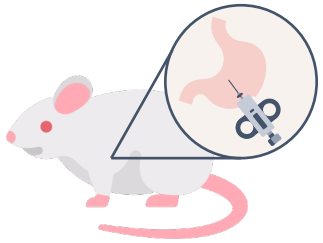
ScaDS.AI DRESDEN LEIPZIG

EKFZ Digital Health

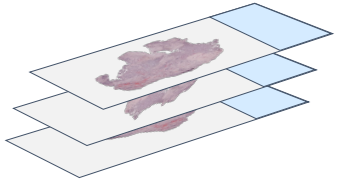
TECHNISCHE UNIVERSITÄT DRESDEN

Dimensionality Reduction for Whole Slide Images

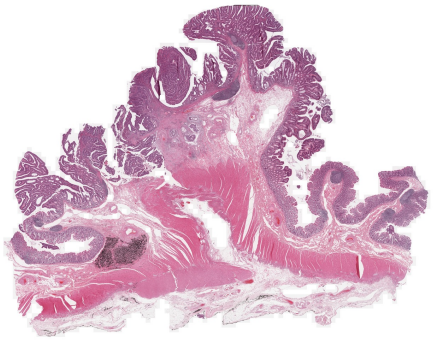
Biopsy/resection



Microscope slide

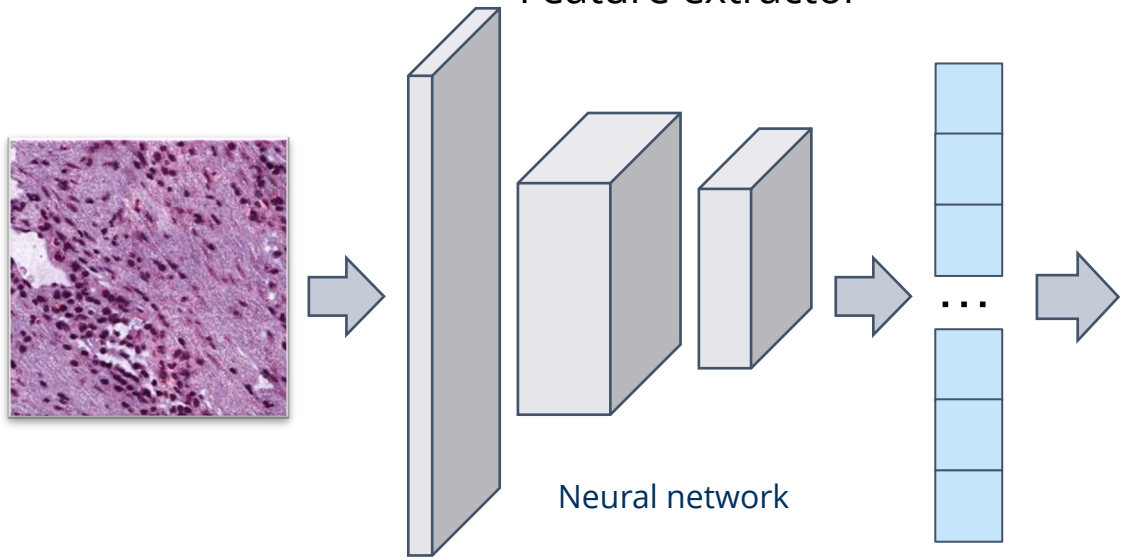


Whole slide image

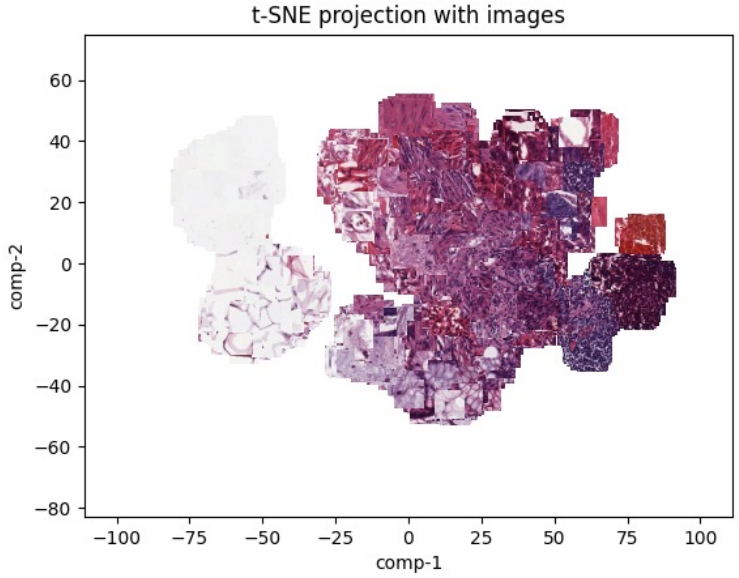


Tessellation

Feature extractor



Neural network



UMAP source: Didem Çifçi, Kather lab

Whole Slide Image Source: The Cancer Genome Atlas, National Cancer Institute

ScaDS.AI DRESDEN LEIPZIG

EKFZ Digital Health

TECHNISCHE UNIVERSITÄT DRESDEN

How to choose the best algorithm for your data?

- Depends on the dataset
- Subjective assessment of obtained results

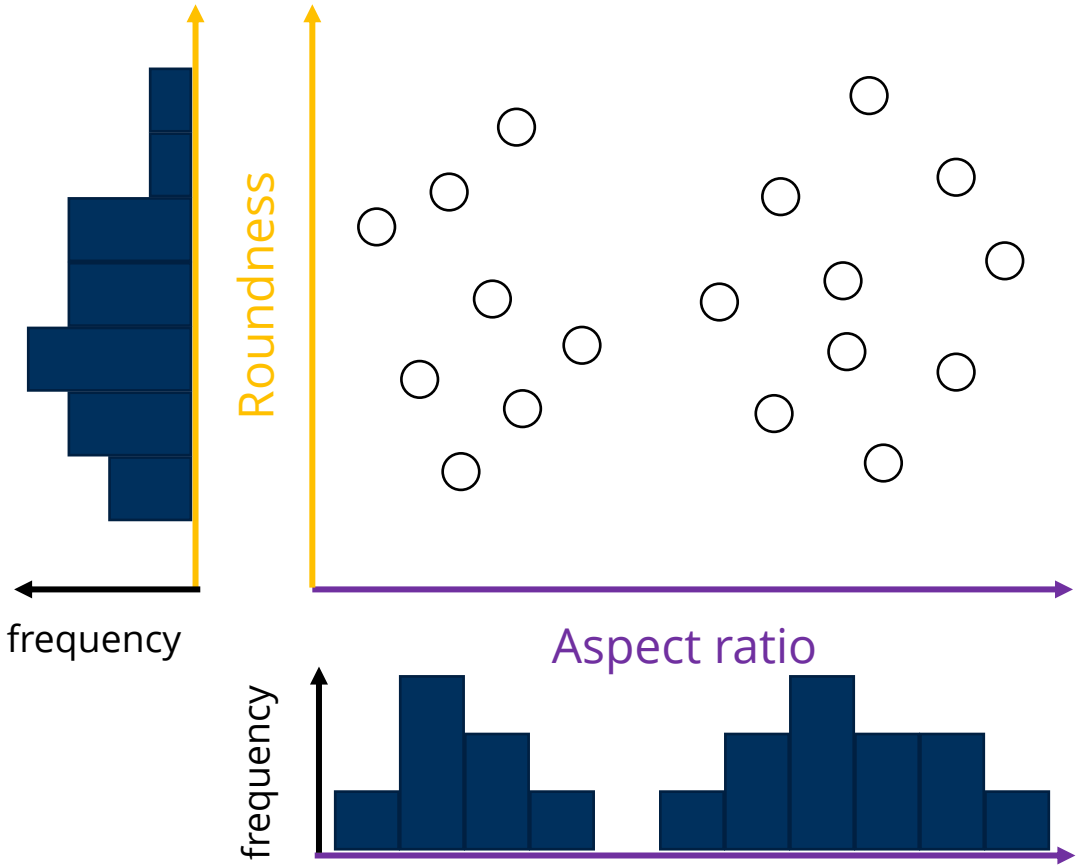
How to compare different low dimensional embeddings?

- Lack of robust statistical approaches available to compare different results
- There is some literature trying to fill this gap (Roca et al., 2023)

Roca et al., (2023). **A cross entropy test allows quantitative statistical comparison of t-SNE and UMAP representations**. *Cell Reports Methods*, *3*(1).

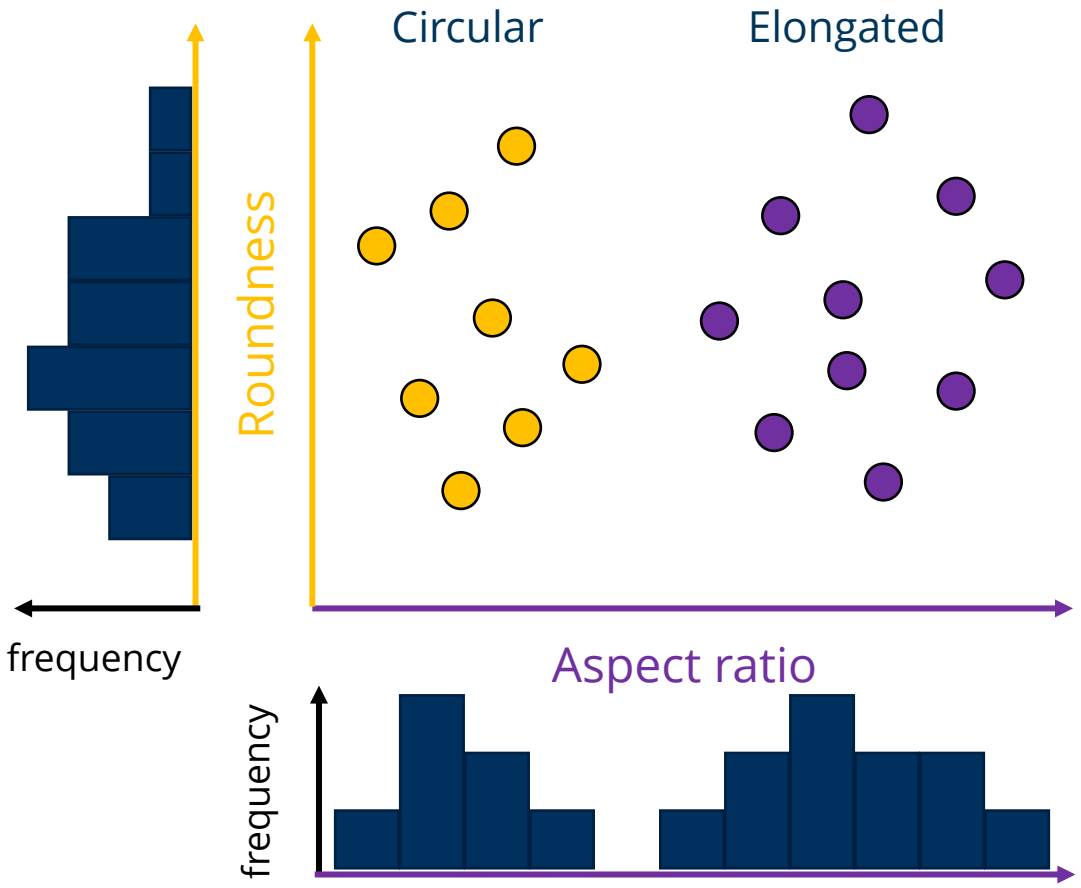
Unsupervised Machine Learning

Unsupervised machine learning algorithms try to find any similarities, differences, patterns, and structure in data by itself, without the provided ground truth (labels).



Unsupervised Machine Learning

Unsupervised machine learning algorithms try to find any similarities, differences, patterns, and structure in data by itself, without the provided ground truth (labels).



K-Means Clustering

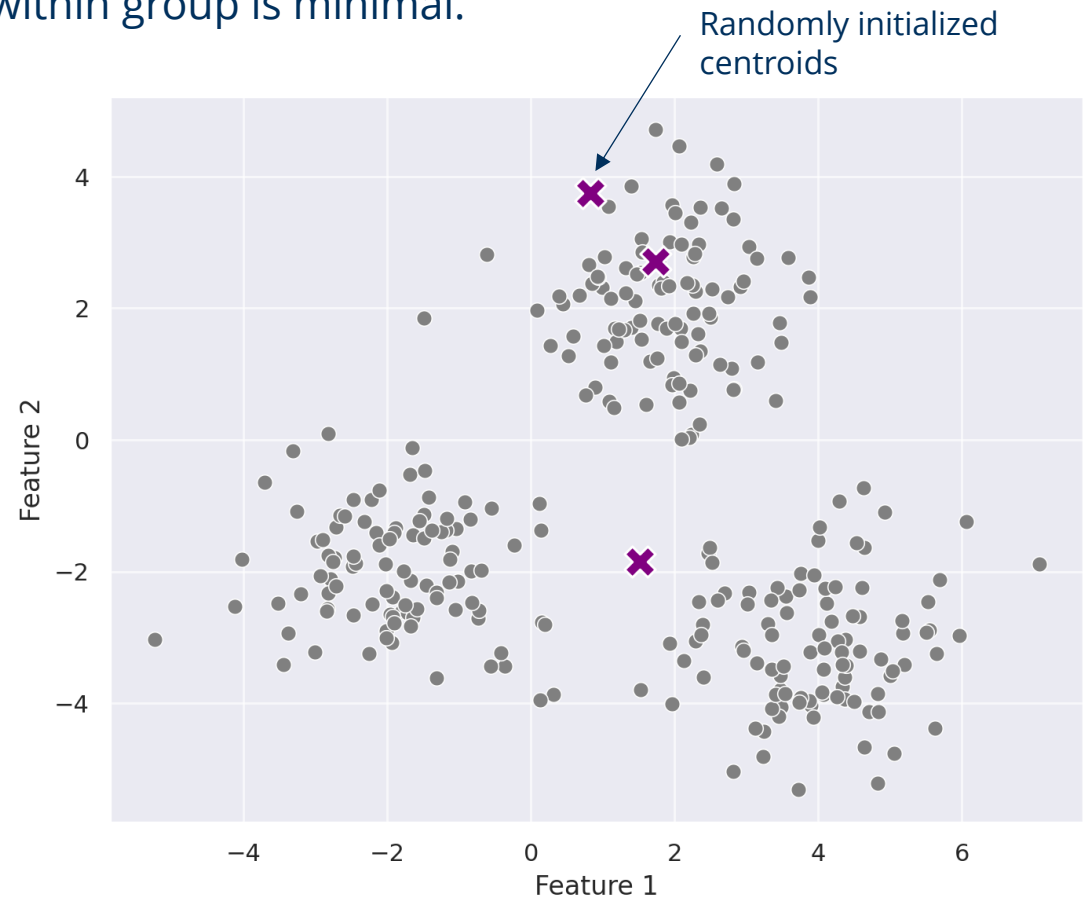
Goal: group data points into k groups so that variance within group is minimal.

STEP 1: k initial centroids are randomly initialized. These centroids are the "centers" of the initial clusters.

STEP 2: each data point is assigned to the nearest centroid. The "nearest" is typically determined by the **Euclidean distance** between the data point and the centroid. This forms k clusters.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$$

n – dimensionality, in this example $= 2$



K-Means Clustering

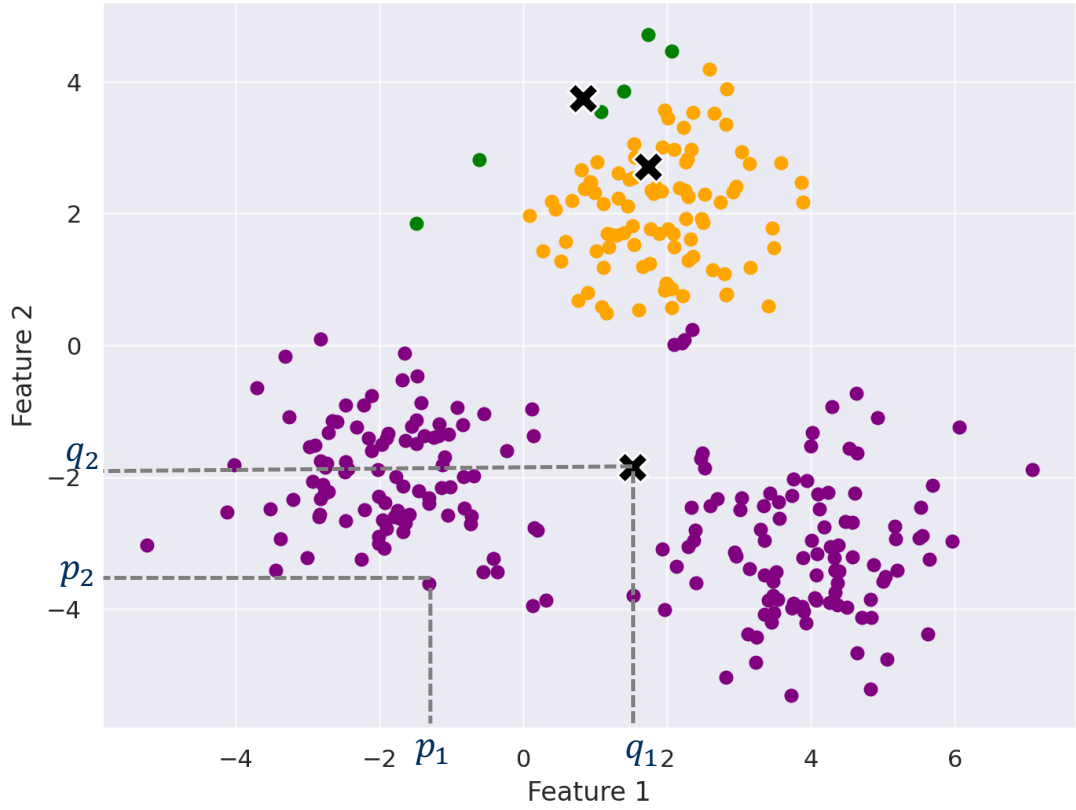
Goal: group data points into k groups so that variance within group is minimal.

STEP 1: k initial centroids are randomly initialized. These centroids are the "centers" of the initial clusters.

STEP 2: each data point is assigned to the nearest centroid. The "nearest" is typically determined by the **Euclidean distance** between the data point and the centroid. This forms k clusters.

$$d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2}$$

n – dimensionality, in this example = 2



K-Means Clustering

Goal: group data points into k groups so that variance within group is minimal.

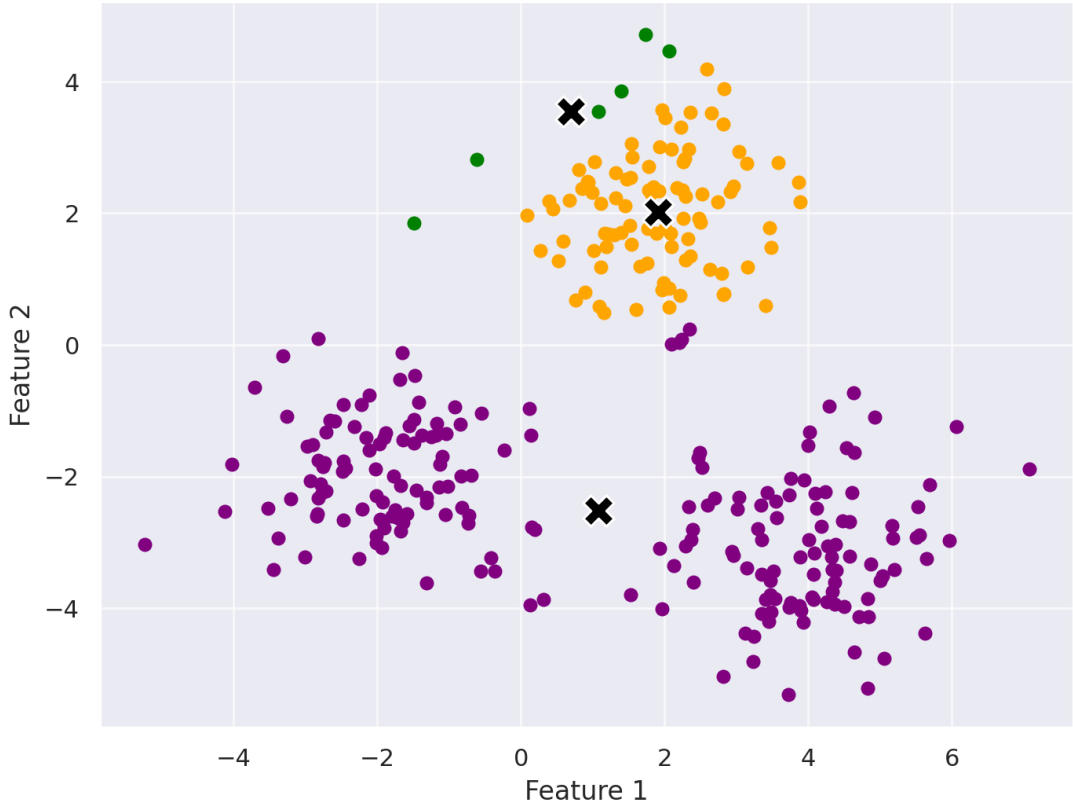
STEP 3: Recalculation of centroids of the clusters formed by taking the mean of all points assigned to each cluster.

$$\text{New centroid}_i = \frac{1}{|C_i|}\sum_{x \in C_i} x$$

C_i - the number of data points in cluster i

Repeat steps 2-3: the assignment and update steps are repeated iteratively until one of the following conditions is met:

- The centroids do not change (or their changes are below a certain tolerance).
- The assignments do not change (no data point moves to a different cluster).
- A predetermined number of iterations is reached.



K-Means Clustering

Goal: group data points into k groups so that variance within group is minimal.

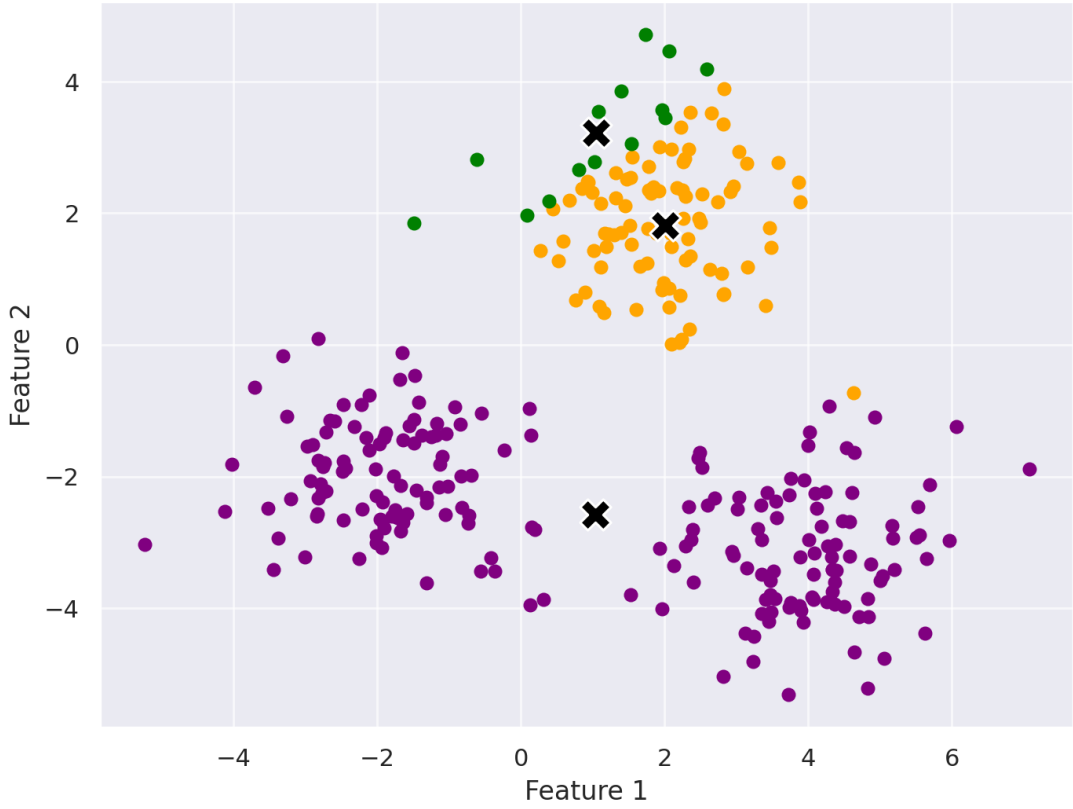
STEP 3: Recalculation of centroids of the clusters formed by taking the mean of all points assigned to each cluster.

$$\text{New centroid}_i = \frac{1}{|C_i|}\sum_{x \in C_i} x$$

C_i - the number of data points in cluster i

Repeat steps 2-3: the assignment and update steps are repeated iteratively until one of the following conditions is met:

- The centroids do not change (or their changes are below a certain tolerance).
- The assignments do not change (no data point moves to a different cluster).
- A predetermined number of iterations is reached.



K-Means Clustering

Goal: group data points into k groups so that variance within group is minimal.

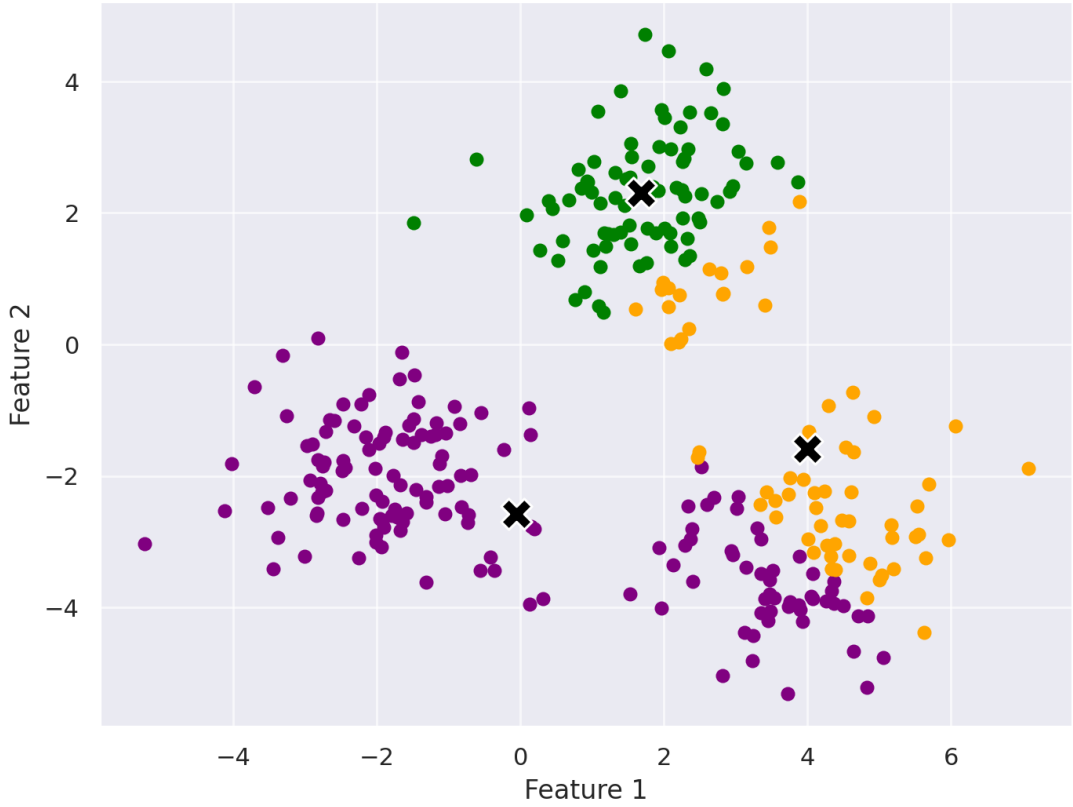
STEP 3: Recalculation of centroids of the clusters formed by taking the mean of all points assigned to each cluster.

$$\text{New centroid}_i = \frac{1}{|C_i|}\sum_{x \in C_i} x$$

C_i - the number of data points in cluster i

Repeat steps 2-3: the assignment and update steps are repeated iteratively until one of the following conditions is met:

- The centroids do not change (or their changes are below a certain tolerance).
- The assignments do not change (no data point moves to a different cluster).
- A predetermined number of iterations is reached.



K-Means Clustering

Goal: group data points into k groups so that variance within group is minimal.

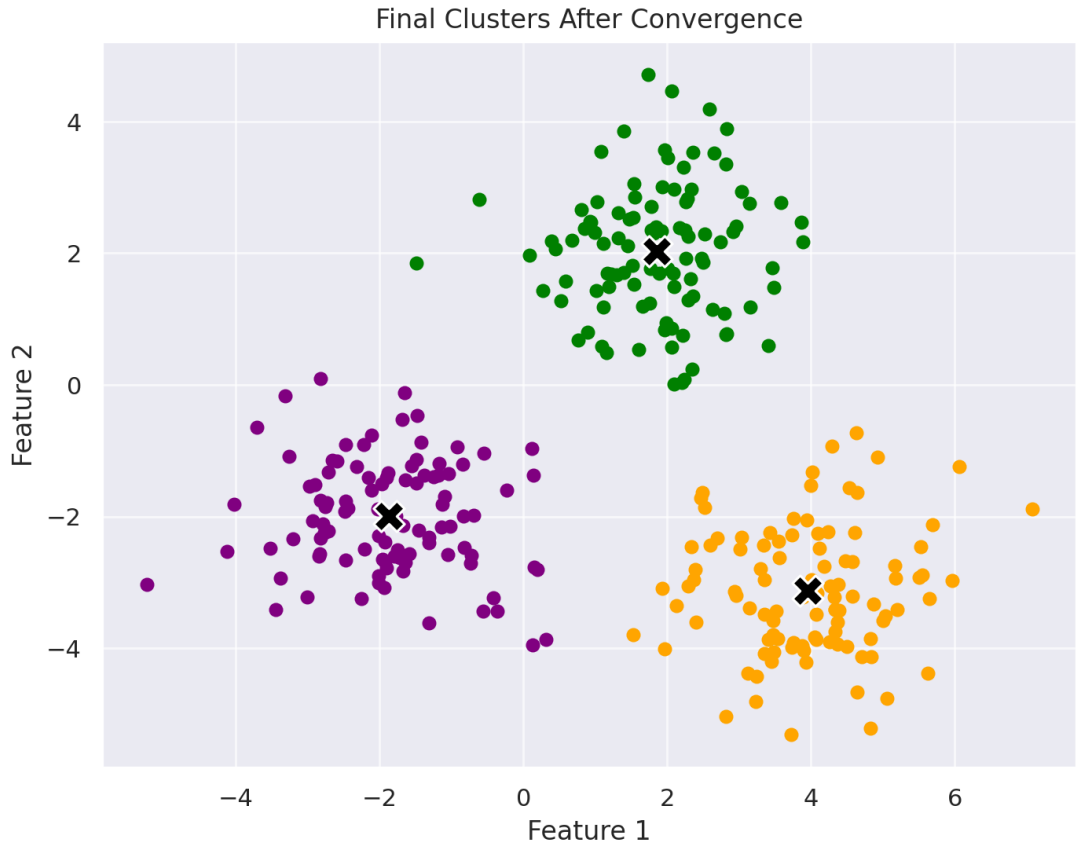
STEP 3: Recalculation of centroids of the clusters formed by taking the mean of all points assigned to each cluster.

$$\text{New centroid}_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$$

C_i - the number of data points in cluster i

Repeat steps 2-3: the assignment and update steps are repeated iteratively until one of the following conditions is met:

- The centroids do not change (or their changes are below a certain tolerance).
- The assignments do not change (no data point moves to a different cluster).
- A predetermined number of iterations is reached.



K-Means Clustering
In Python

```
from sklearn.cluster import KMeans
```

```
kmeans = KMeans(n_clusters=2, random_state=42)
kmeans.fit(simple_data)
```

```
# The cluster centers (means)
centroids = kmeans.cluster_centers_
```

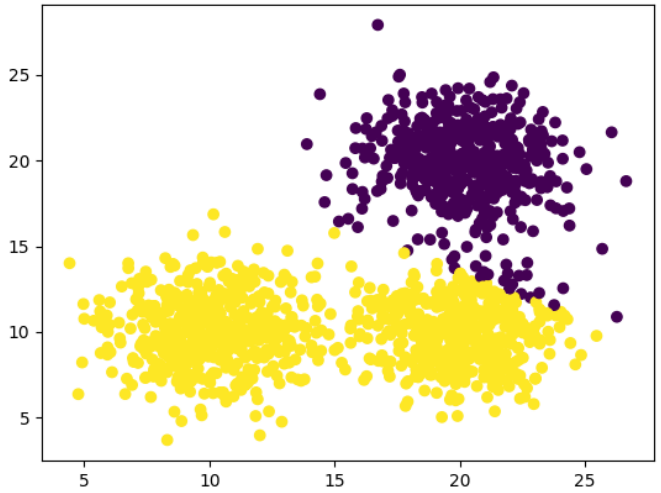
```
# Using the 'predict' method to assign new points to the nearest cluster centroid
new_points = np.array([[0, 0], [12, 3]])
predicted_labels = kmeans.predict(new_points)
```

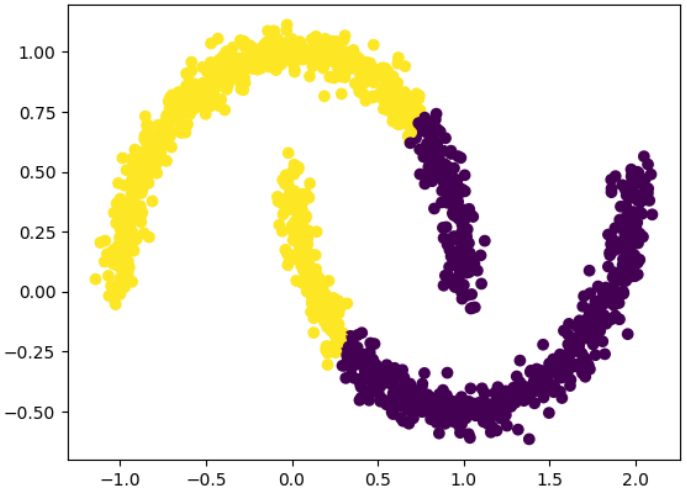
K-Means Clustering
In Python

Advantages & Disadvantages:

- Simplicity and Speed
- Easy to interpret results
- Well-suited for spherical clusters and of similar size
- Based on Euclidean distance → every new point can be assigned to a cluster

- Number of clusters needs to be known
- Vulnerability to outliers
- Difficulty with varying densities
- Convergence to local minima
- Clusters can not capture more complex topologies

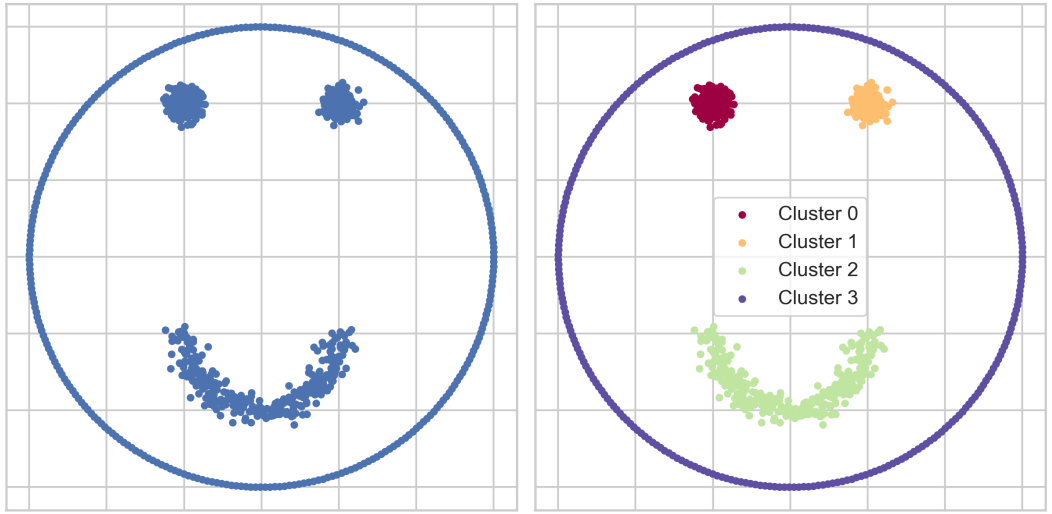


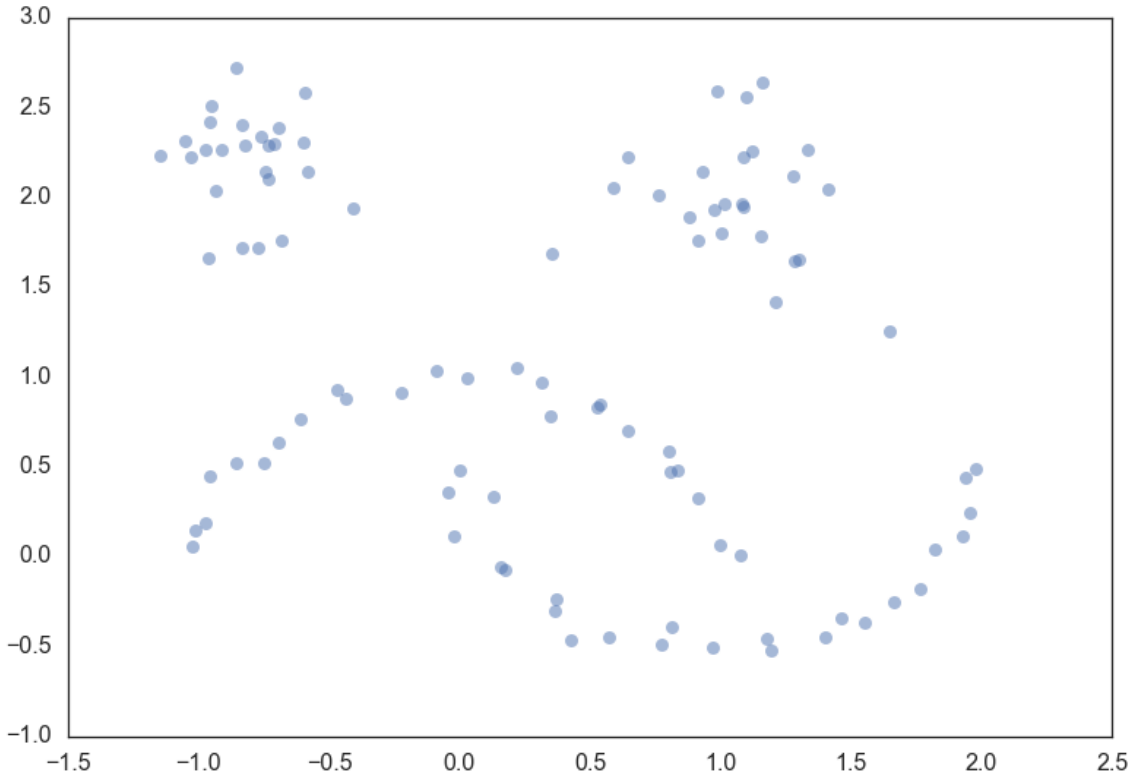


Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

Unlike K-means which uses centroid-based clustering, HDBSCAN relies on **density-based** clustering.

→ Assumes that clusters are defined as areas of higher density than the remainder of the dataset, which allows it to find arbitrarily shaped clusters and handle noise (outliers) effectively.

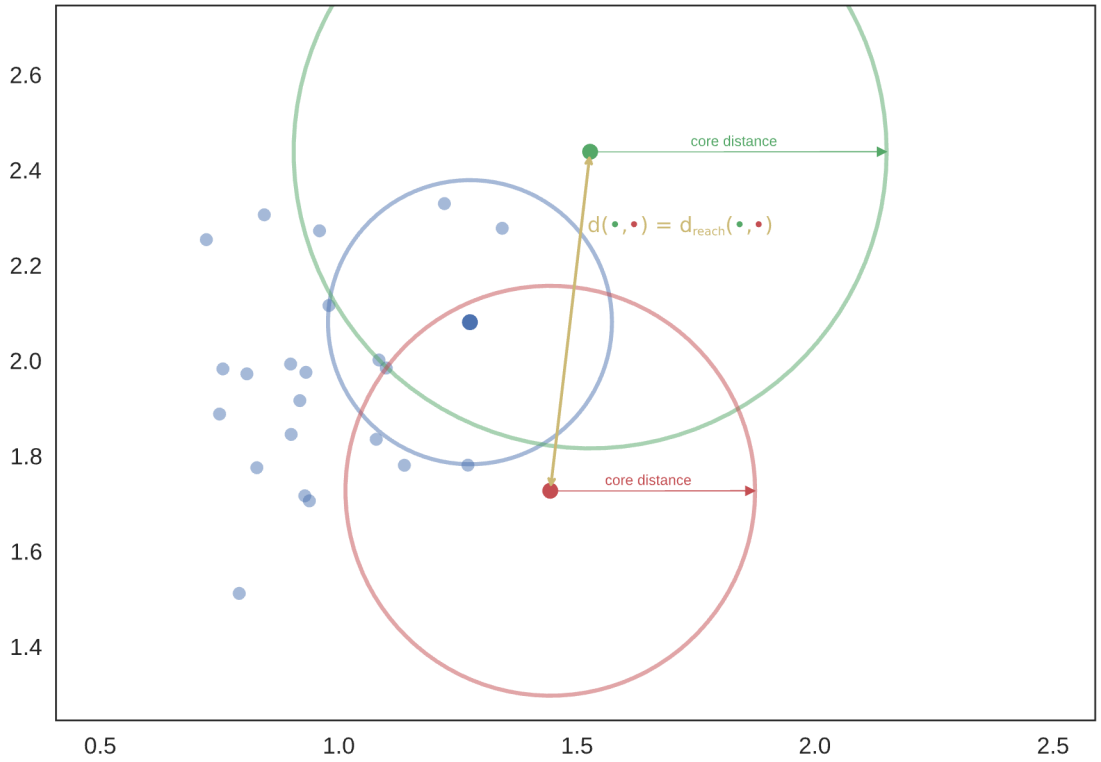




https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

STEP 1: Transform the space according to the density/sparsity.



Core distance: Distance to n-th nearest neighbor
Distance metric: Mutual reachability
Core distance of Q > d(P, Q) → $d_{new}(P,Q)$ = core distance
Core distance of Q < d(A, Q) → $d_{new}(A,Q)$ = d(A,Q)

→ Isolated points are pushed further away from clusters

"To find clusters we want to find the islands of higher density amid a sea of sparser noise [...] For practical purposes that means making 'sea' points more distant from each other and from the 'land'."

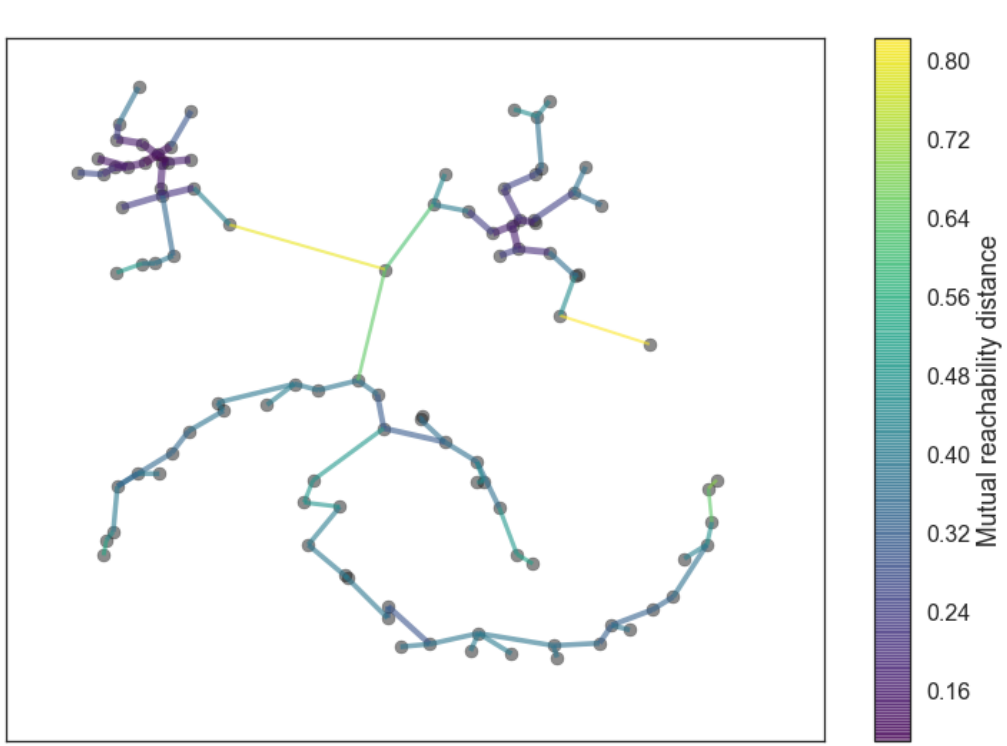
https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

ScaDS.AI DRESDEN LEIPZIG
EKFZ Digital Health
TECHNISCHE UNIVERSITÄT DRESDEN

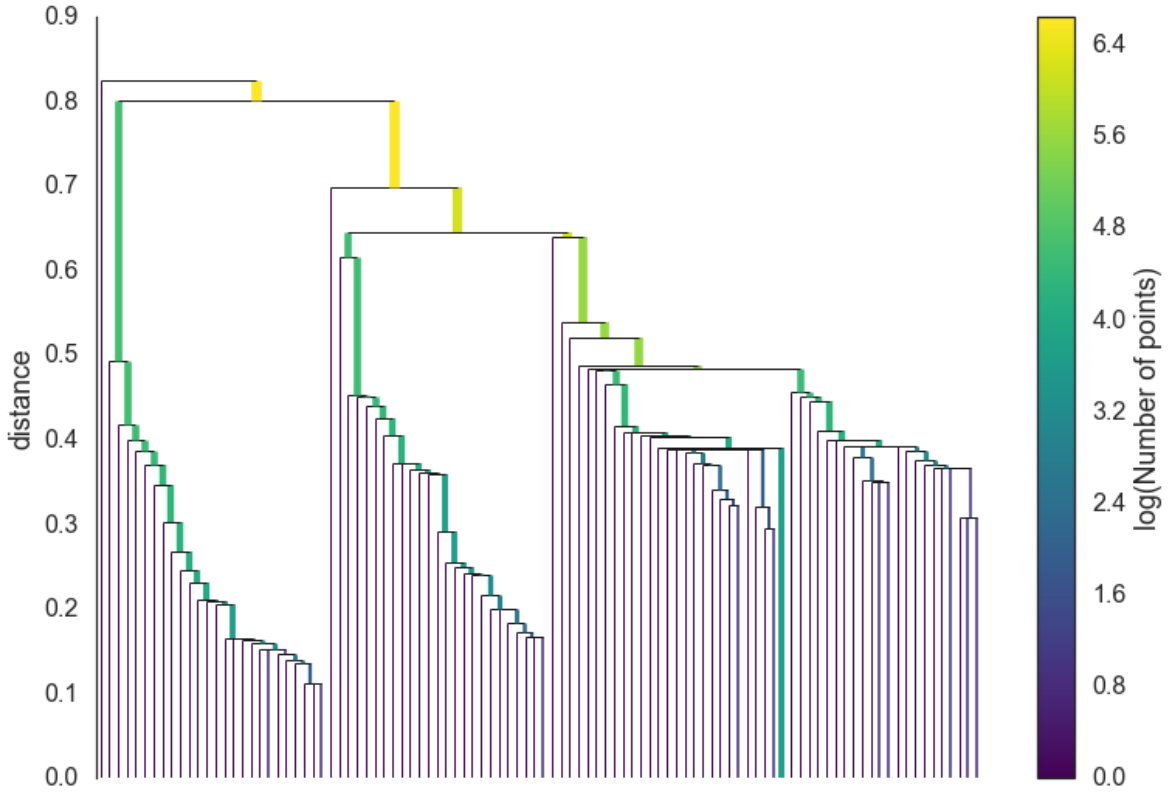
Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

STEP 2: Build the minimum spanning tree of the distance weighted graph.

STEP 3: Construct a cluster hierarchy of connected components.



Sort edges by distances & Merge close points

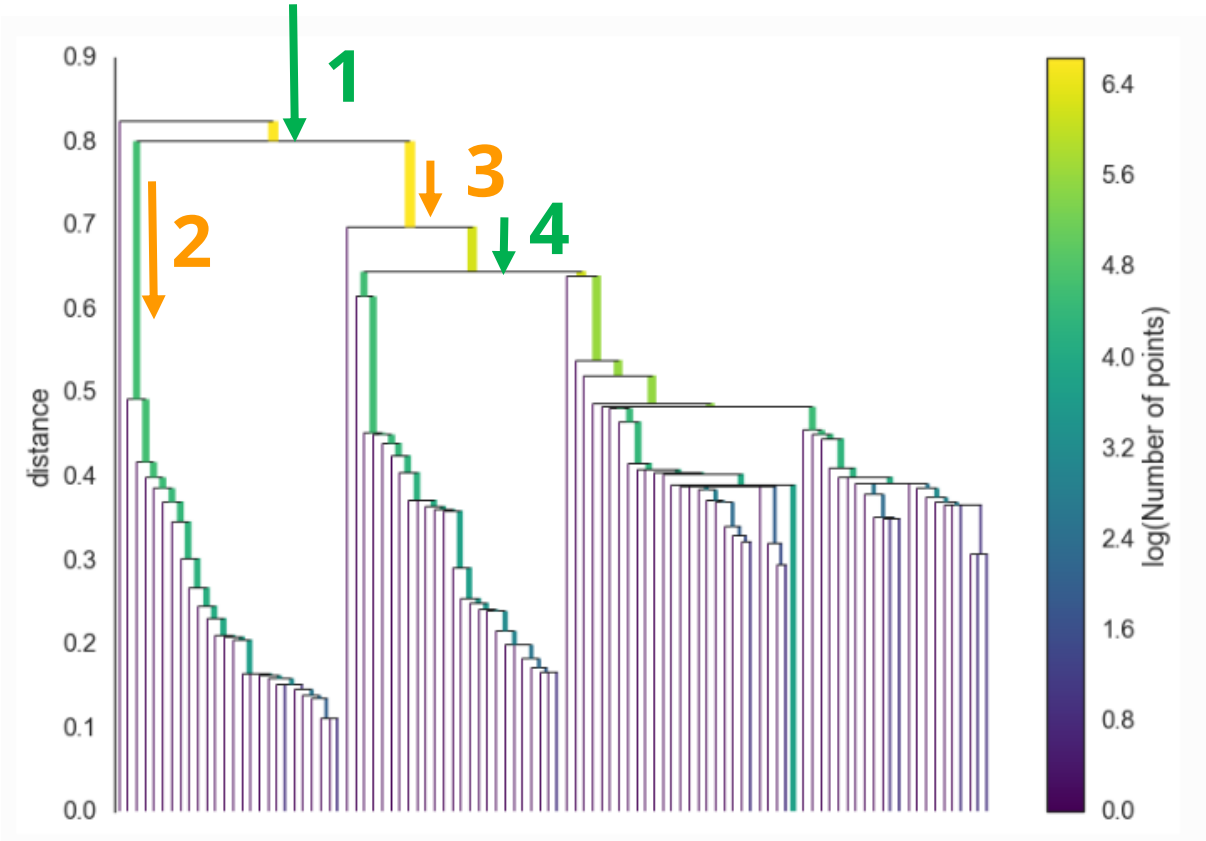


https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

ScaDS.AI DRESDEN LEIPZIG
EKFZ Digital Health
TECHNISCHE UNIVERSITÄT DRESDEN

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

STEP 4: Condense the cluster hierarchy based on minimum cluster size. Traverse graph from top to bottom and decide whether a new cluster is formed at every crossroads



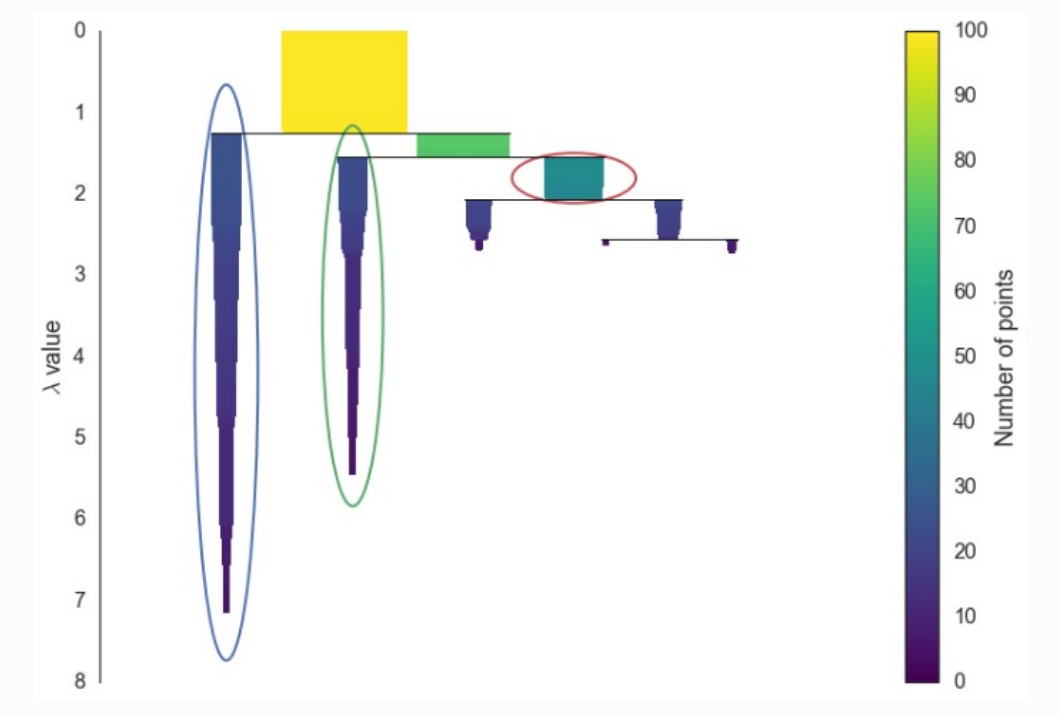
1. If points are split into clusters here – are both clusters larger than a size threshold? Yes
2. No – this part of the tree remains a single cluster
3. No – this part of the tree remains a single cluster
4. Yes – remaining points are split into new clusters here

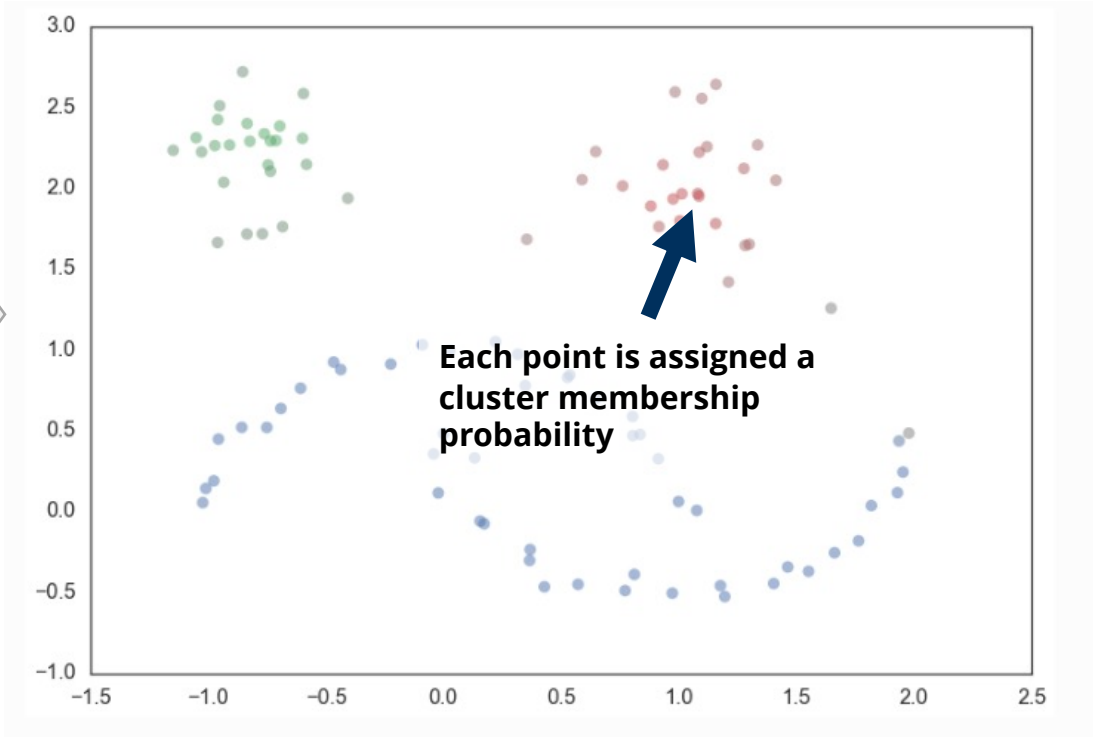
https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN)

STEP 5: Extract the stable clusters from the condensed tree.

Extracting the clusters with 'largest total ink area' leads to the final selection of clusters

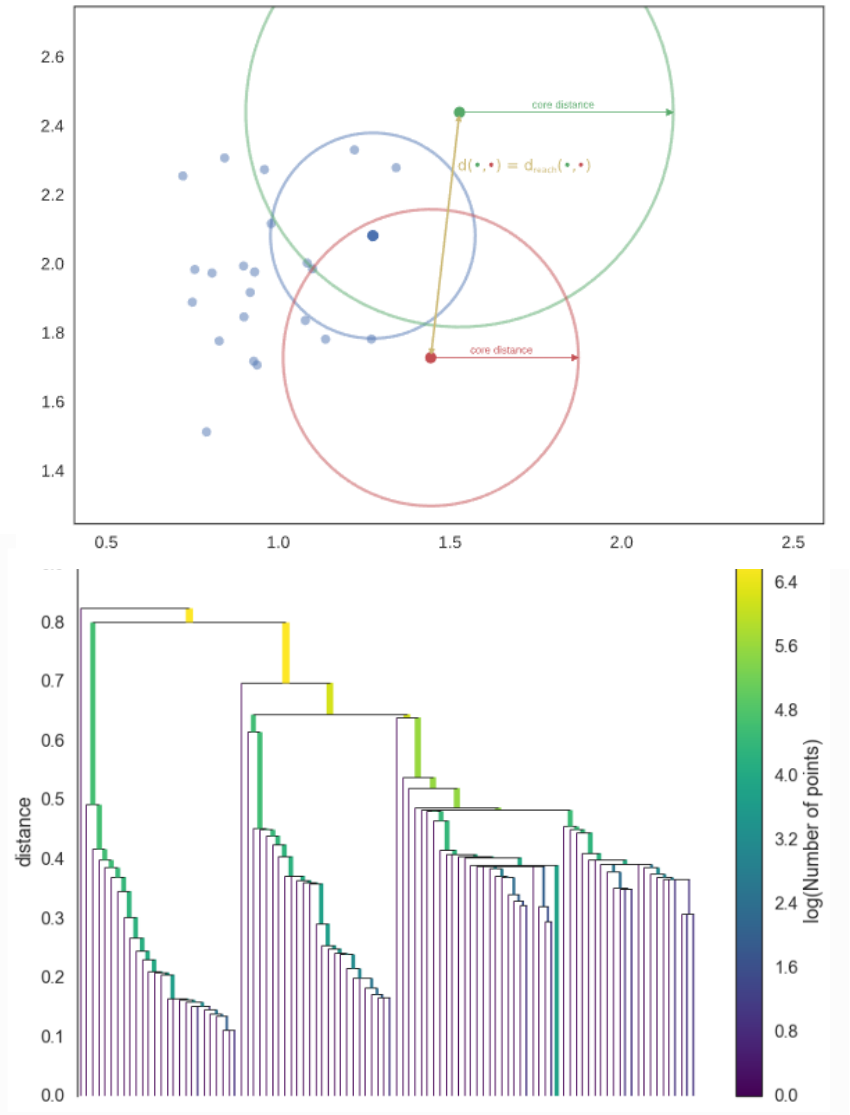




https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

ScaDS.AI DRESDEN LEIPZIG
EKFZ Digital Health
TECHNISCHE UNIVERSITÄT DRESDEN

Variants of Linkage-clustering



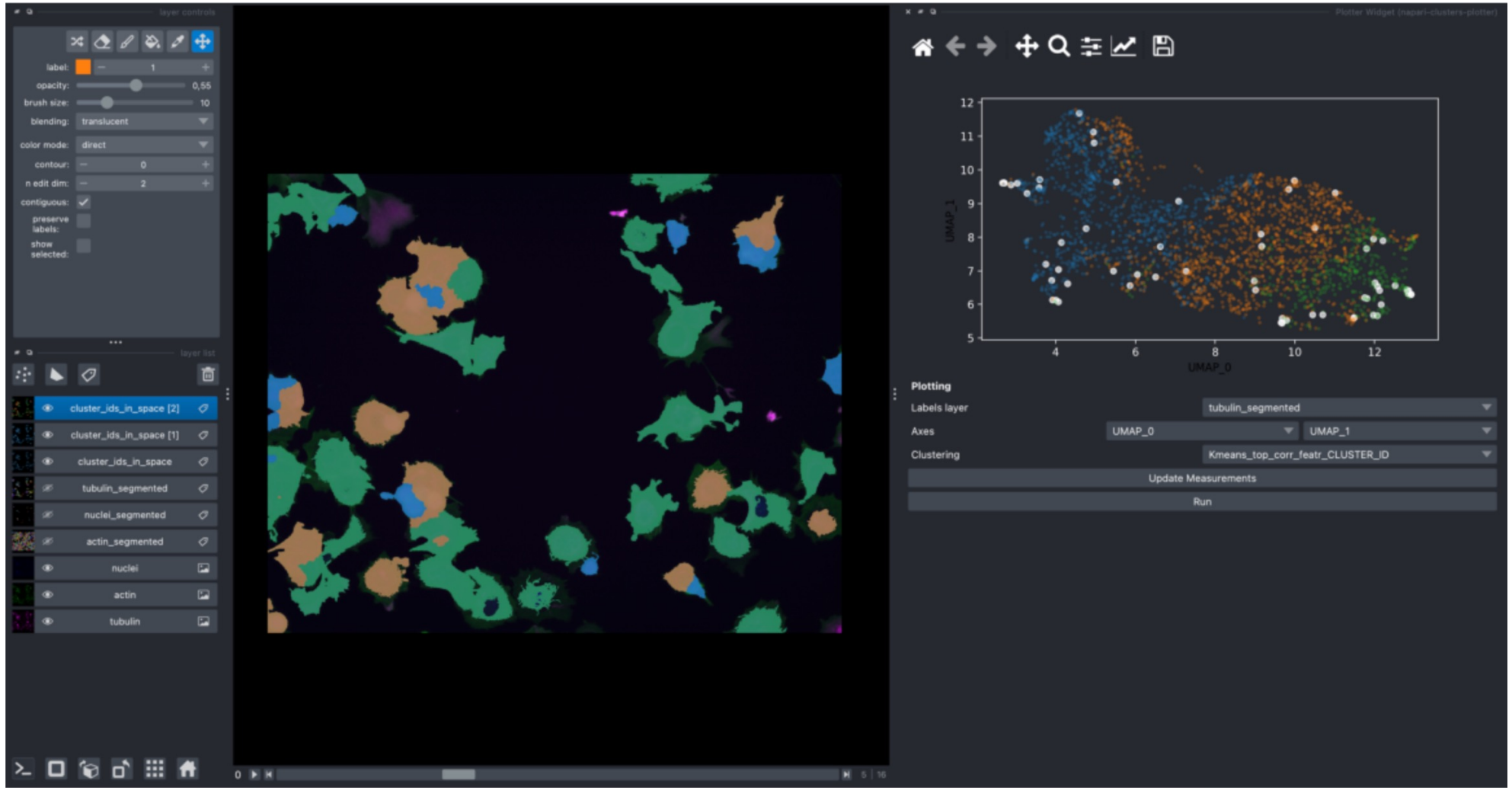
There are multiple ways to reconstruct the neighborhood graph and the clusters in the hierarchy schematic:

- Setting a maximum distance between two points to be considered neighbors → **DBSCAN**
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

- Aggregate points into clusters bottom-up → **Agglomerative clustering**
  https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

Interactive Hands-On Session with Napari

Data preparation, feature extraction, feature exploration, clustering, dimensionality reduction



ScaDS.AI DRESDEN LEIPZIG

EKFZ Digital Health

TECHNISCHE UNIVERSITÄT DRESDEN

Recap: Environment Preparation

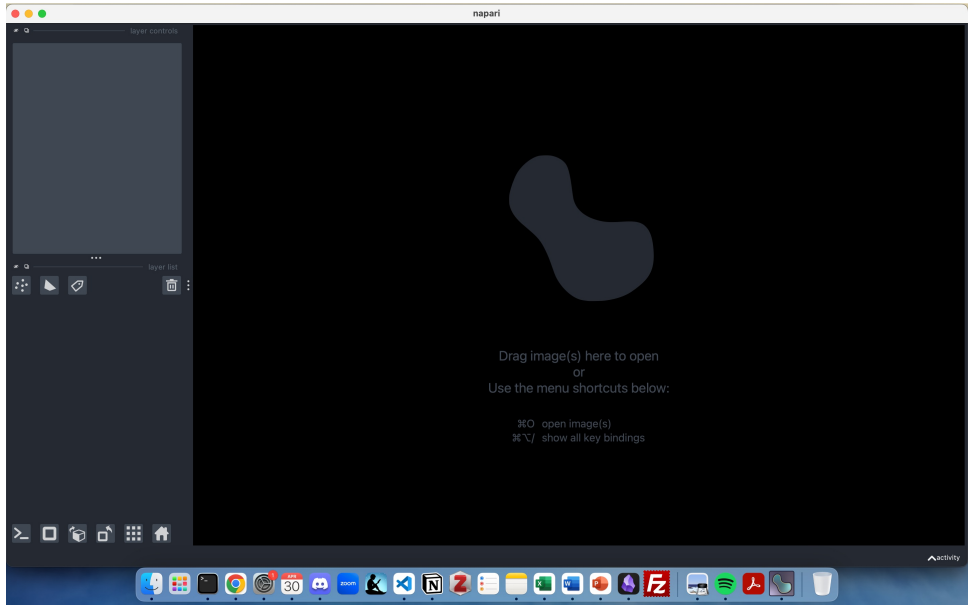
Install *conda/miniforge/mamba/micromamba* on your machine:

https://biapol.github.io/blog/mara_lampert/getting_started_with_mambaforge_and_python/readme.html

Follow installation instructions for ***devbio-napari*** collection of *napari* plugins:

https://github.com/haesleinhuepf/devbio-napari

```
Transaction finished

To activate this environment, use:

    micromamba activate devbio-napari-env

Or to execute a single command in this environment, use:

    micromamba run -n devbio-napari-env mycommand

laura@Lauras-MacBook-Air ~ % micromamba activate devbio-napari-env
(devbio-napari-env) laura@Lauras-MacBook-Air ~ % napari
```



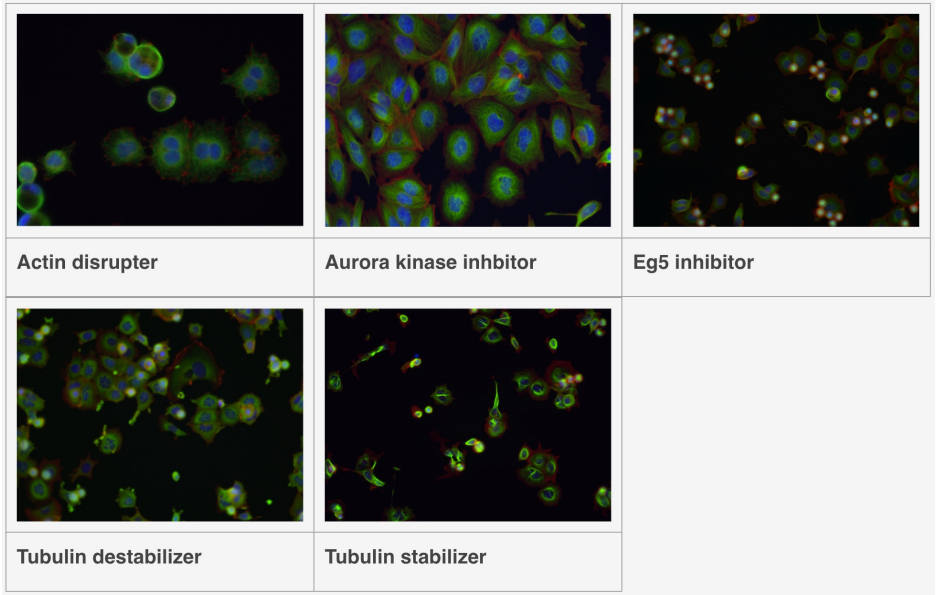
ScaDS.AI DRESDEN LEIPZIG

EKFZ Digital Health

TECHNISCHE UNIVERSITÄT DRESDEN

Dataset: Image Set of Human HT29 Colon-cancer Cells (BBBC021)

Broad Bioimage Benchmark Collection (BBBC)



Download data from:

https://bbbc.broadinstitute.org/BBBC021

The dataset is designed for evaluating the ability to predict **biological mechanisms of action** (MoA) based on **morphological changes** in cells caused by chemical compounds. The images have been treated with 113 different small-molecule compounds at various concentrations, resulting in a variety of **cellular phenotypes**.

Key Features of BBBC021:

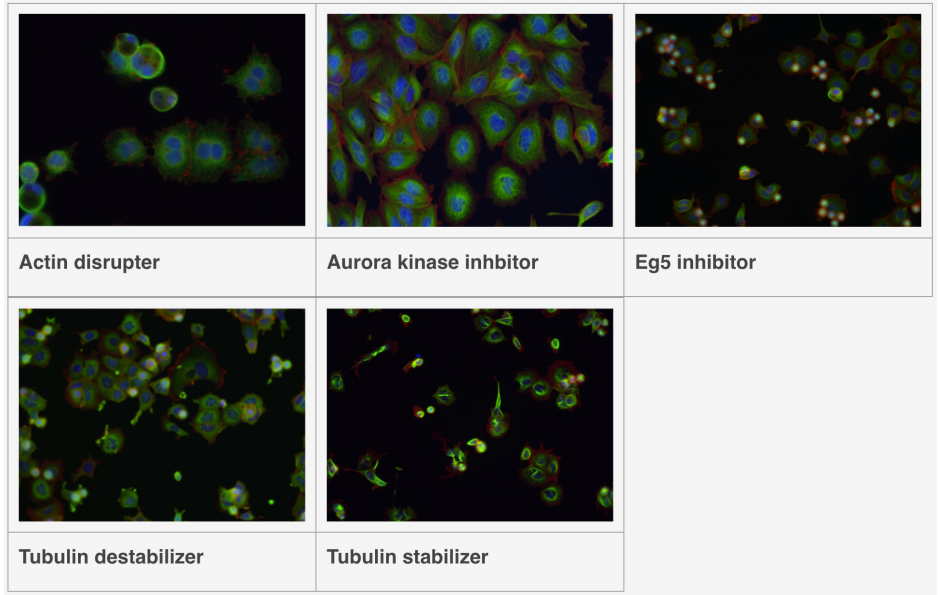
- **Images:** The dataset contains thousands of images, each corresponding to a well of a microplate where cells have been treated with a different compound.
- **Labels:** Each image is associated with a compound and often a MoA, providing a clear label for supervised learning tasks. We will use this as a ground truth to compare against the clusters discovered.
- **Metadata:** Includes details about the compound, dose, and batch, which can be used to perform more nuanced analyses and correct for batch effects. ← *Not part of this workshop*

The BBBC resource is described in the following publication: Ljosa V, Sokolnicki KL, Carpenter AE (2012). Annotated high-throughput microscopy image sets for validation. Nature Methods 9(7):637 / doi. PMID: 22743765 PMCID: PMC3627348. Available at http://dx.doi.org/10.1038/nmeth.2083

ScaDS.AI DRESDEN LEIPZIG · EKFZ Digital Health · TECHNISCHE UNIVERSITÄT DRESDEN

Dataset: Image Set of Human HT29 Colon-cancer Cells (BBBC021)

Broad Bioimage Benchmark Collection (BBBC)



Download data from:

https://bbbc.broadinstitute.org/BBBC021

The dataset is designed for evaluating the ability to predict **biological mechanisms of action** (MoA) based on **morphological changes** in cells caused by chemical compounds. The images have been treated with 113 different small-molecule compounds at various concentrations, resulting in a variety of **cellular phenotypes**.

BBBC021_v1_images_Week1_22123.zip (839436312 bytes)

BBBC021_v1_images_Week1_22141.zip (851400910 bytes)

BBBC021_v1_images_Week1_22161.zip (841371484 bytes)

BBBC021_v1_images_Week1_22361.zip (854598915 bytes)

BBBC021_v1_images_Week1_22381.zip (861576297 bytes)

BBBC021_v1_images_Week1_22401.zip (874848053 bytes)

The BBBC resource is described in the following publication: Ljosa V, Sokolnicki KL, Carpenter AE (2012). Annotated high-throughput microscopy image sets for validation. Nature Methods 9(7):637 / doi. PMID: 22743765 PMCID: PMC3627348. Available at http://dx.doi.org/10.1038/nmeth.2083

ScaDS.AI DRESDEN LEIPZIG EKFZ Digital Health TECHNISCHE UNIVERSITÄT DRESDEN

Data Preparation: Segmentation of all 3 channels

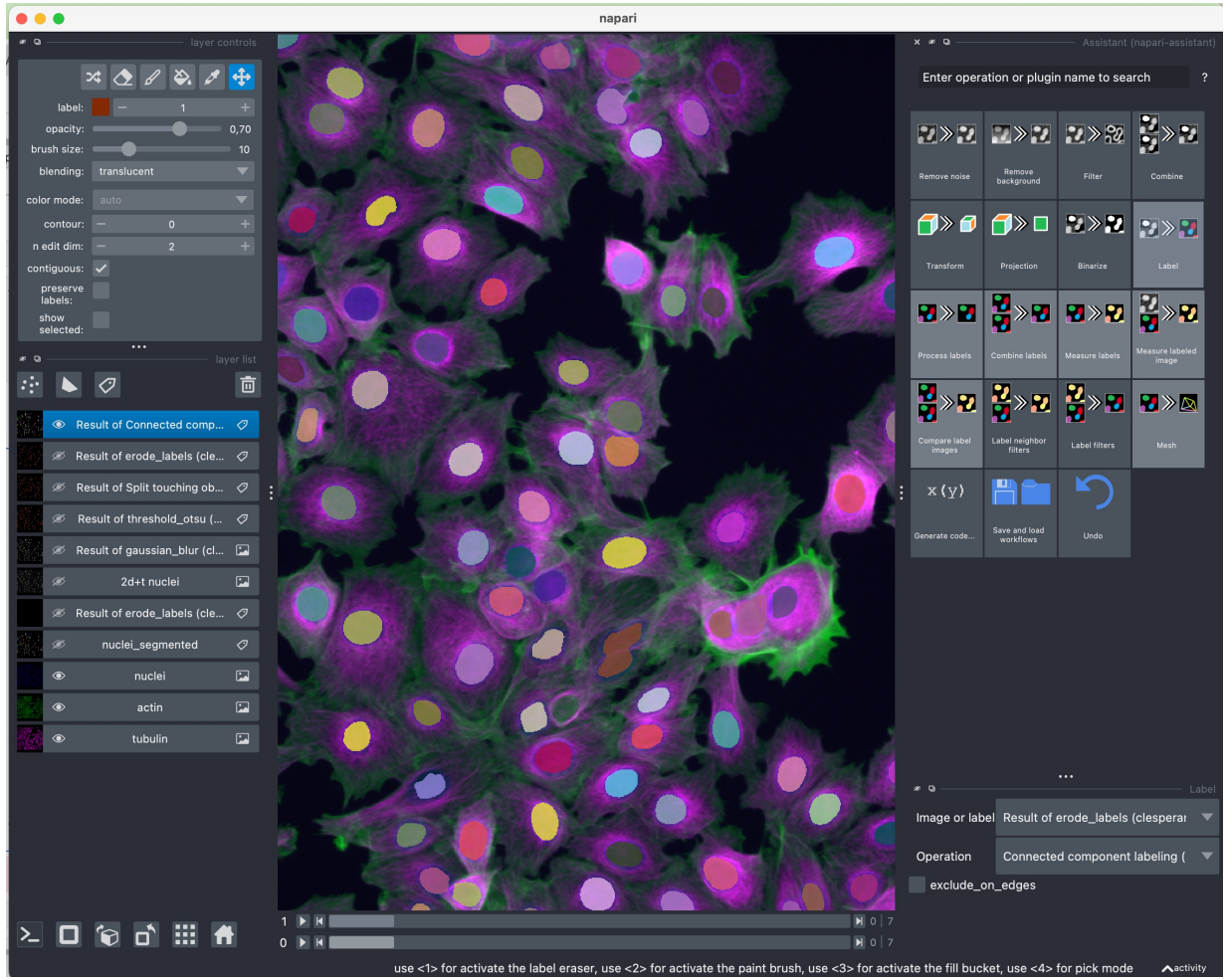
Nuclei (DAPI blue) channel

Perform interactive segmentation in napari (only one timepoint):

- Open one of the prepared dataset images
- Right click on the layer → split RGB
- Now for the layer that you want to process: Plugins → convert to 2D timelapse

Segment nuclei channel:

Plugins → Assistant (napari-assistant)
→ Remove noise (gaussian blur)
→ Binarize (Threshold Otsu)
→ Process labels (Split touching objects, sigma=4)
→ Process labels (Erode labels)
→ Label (Connected component labelling, scikit-image)



Same steps in the notebook!

ScaDS.AI DRESDEN LEIPZIG
EKFZ Digital Health
TECHNISCHE UNIVERSITÄT DRESDEN

Data Preparation: Segmentation of all 3 channels

Actin (Green) channel

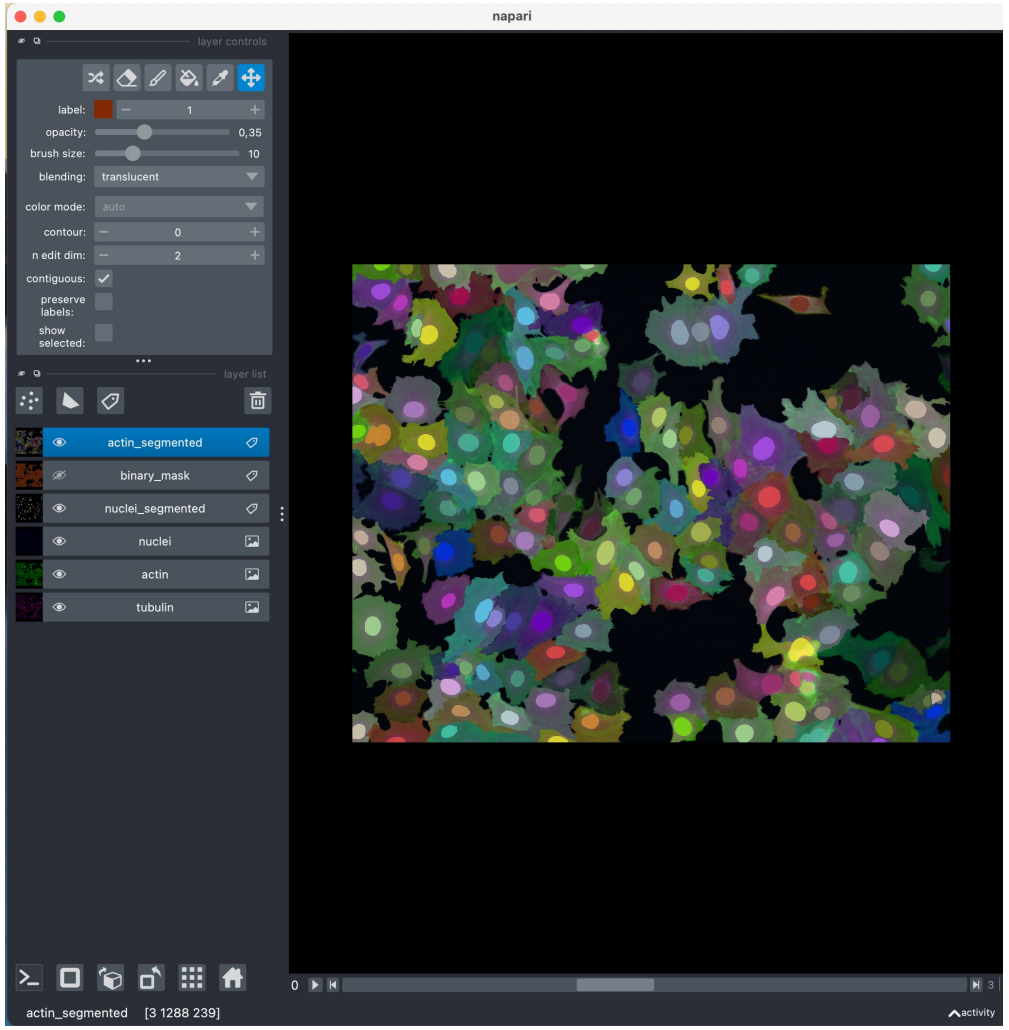
Perform interactive segmentation in napari (only one timepoint):

- Open one of the prepared dataset images
- Right click on the layer → split RGB
- Now for the layer that you want to process: Plugins → convert to 2D timelapse

Segment actin channel:

Plugins → Assistant (napari-assistant)

→ Binarize (Theshold Huang and Wang 1995)
→ Remove noise (Median sphere)
→ Filter (Sobel, Detect edges)
→ Label (Seeded watershed with nuclei as seeds and binary mask, ***only in the notebook!***)



Final result only in the notebook!

Data Preparation: Segmentation of all 3 channels

Tubulin (Red) channel

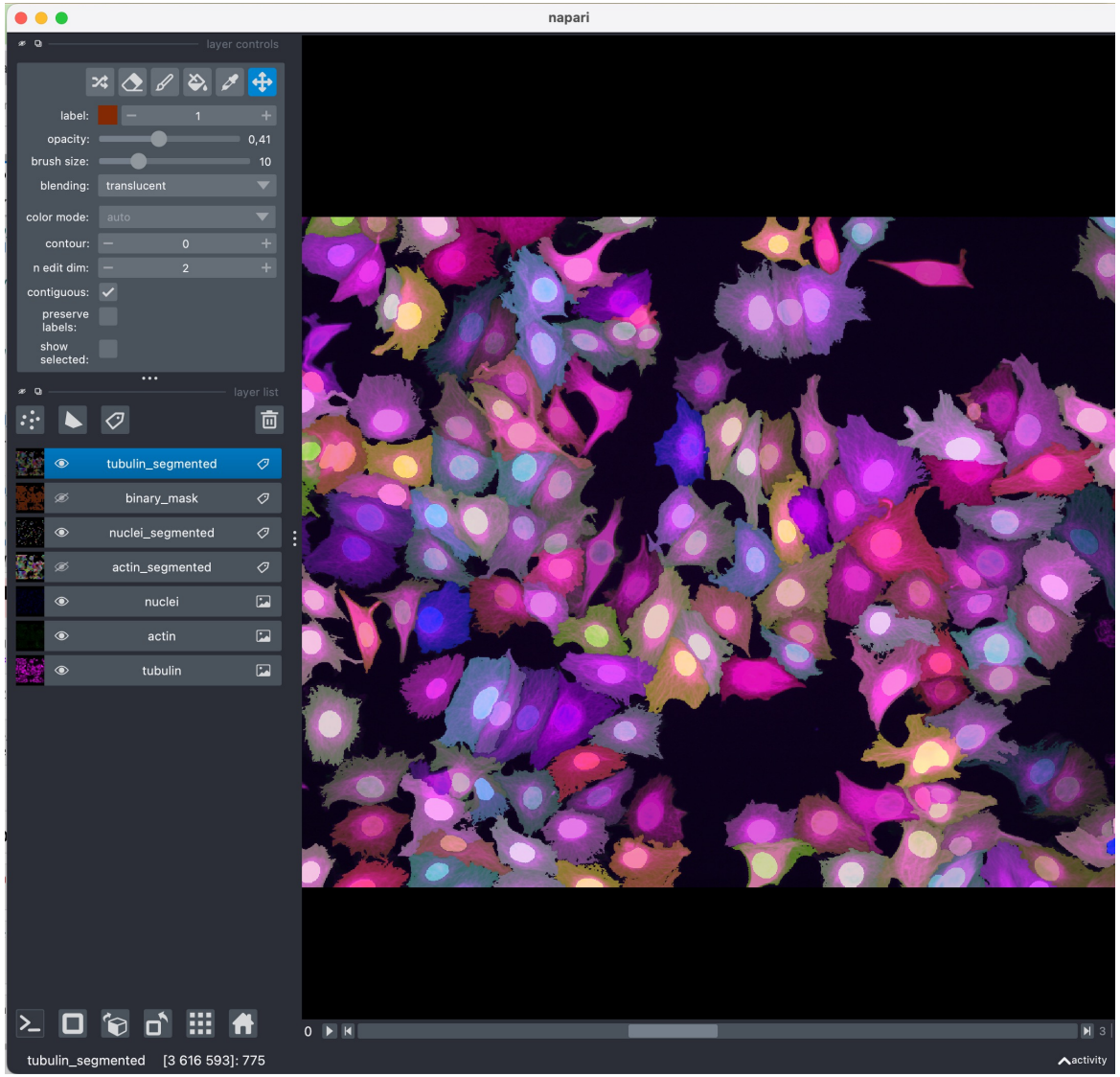
Perform interactive segmentation in napari (only one timepoint):

- Open one of the prepared dataset images
- Right click on the layer → split RGB
- Now for the layer that you want to process: Plugins → convert to 2D timelapse

Segment tubulin channel:

Plugins → Assistant (napari-assistant)

→ Binarize (Theshold Huang and Wang 1995)
→ Remove noise (Median sphere)
→ Label (Seeded watershed with nuclei as seeds and binary mask, ***only in the notebook!***)



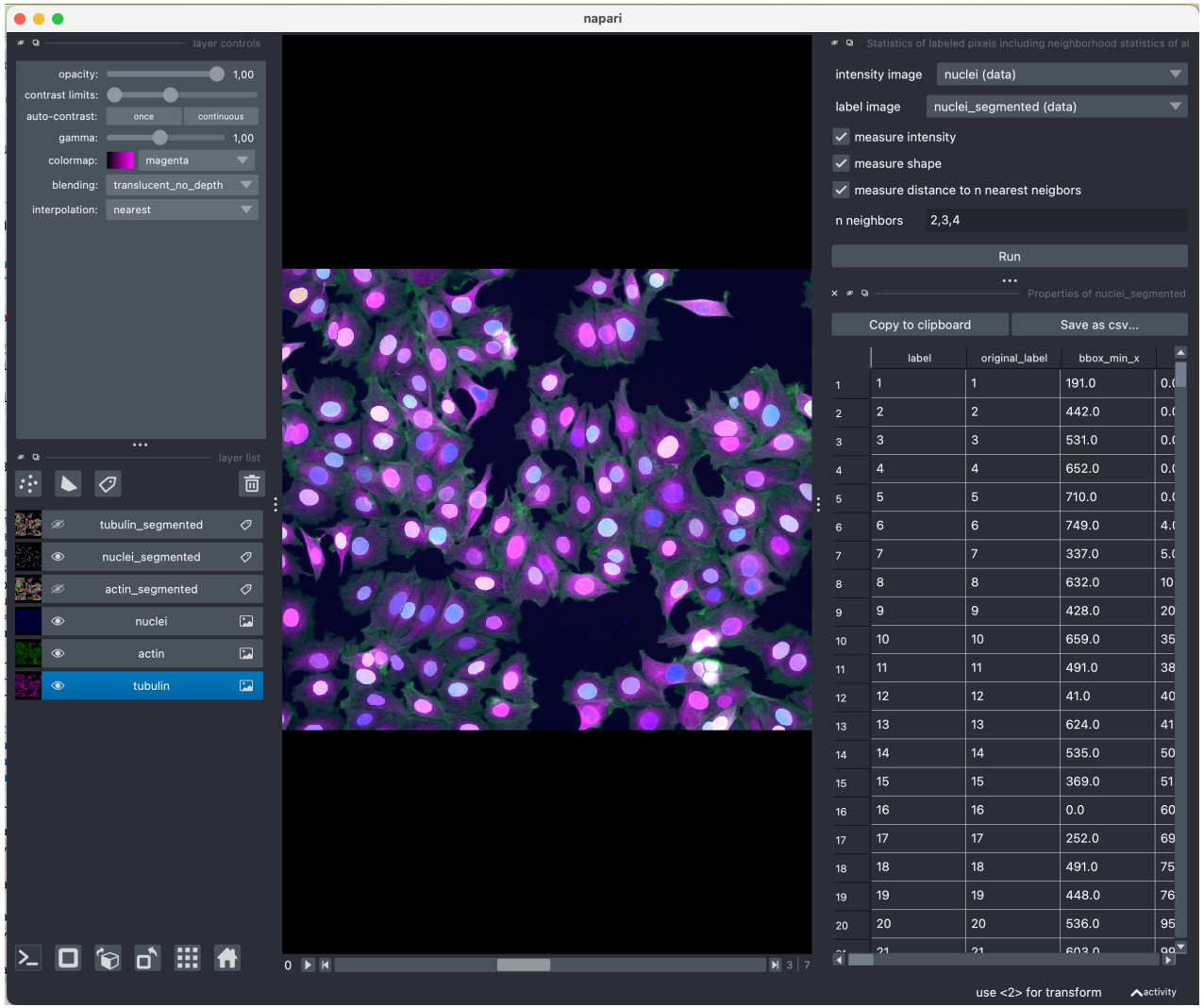
Final result only in the notebook!

Data Preparation: Extracting Quantitative Measurements

Perform interactive measurements extraction in napari:

- Open one of the prepared dataset images
- Right click on the layer → split RGB
- Open corresponding segmentation images
- Now for each layer:
  Plugins → Convert to 2D timelapse

- Tools → Measurement tables → Label statistics of all frames *(clEsperanto)*



Same steps in the notebook!

Dimensionality Reduction & Clustering

Interactively with *napari-clusters-plotter*

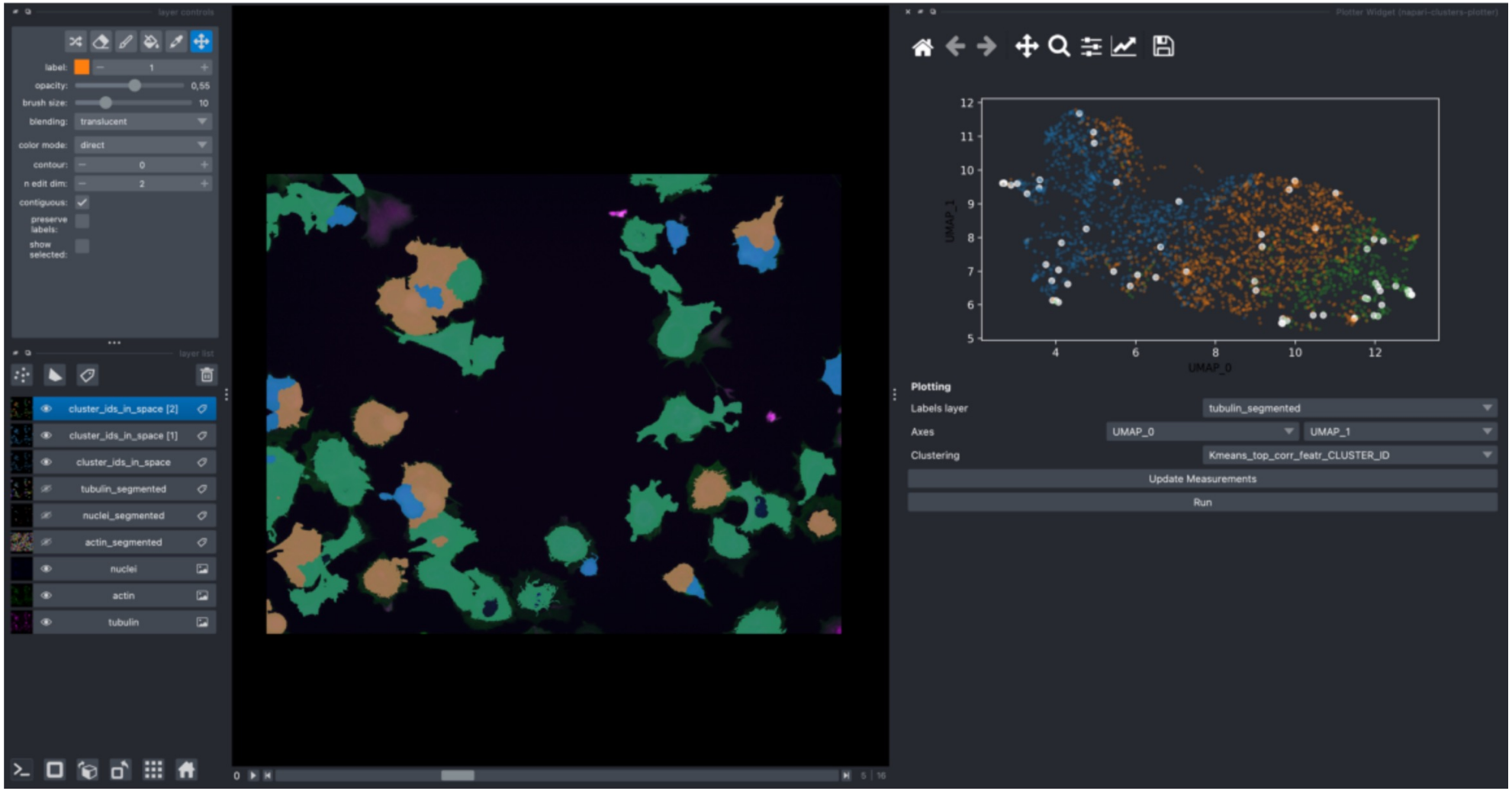


Image-based Profiling

Ground truth: 6 of the 12 mechanisms can be identified visually:

- Actin disruptors
- Aurora kinase inhibitors
- Eg5 inhibitors
- Microtubule destabilizers
- Microtubule stabilizers
- Epithelial

Can you identify any of the features that are important for any of these mechanisms?