

Dimensionality Reduction

Thomas Höllt

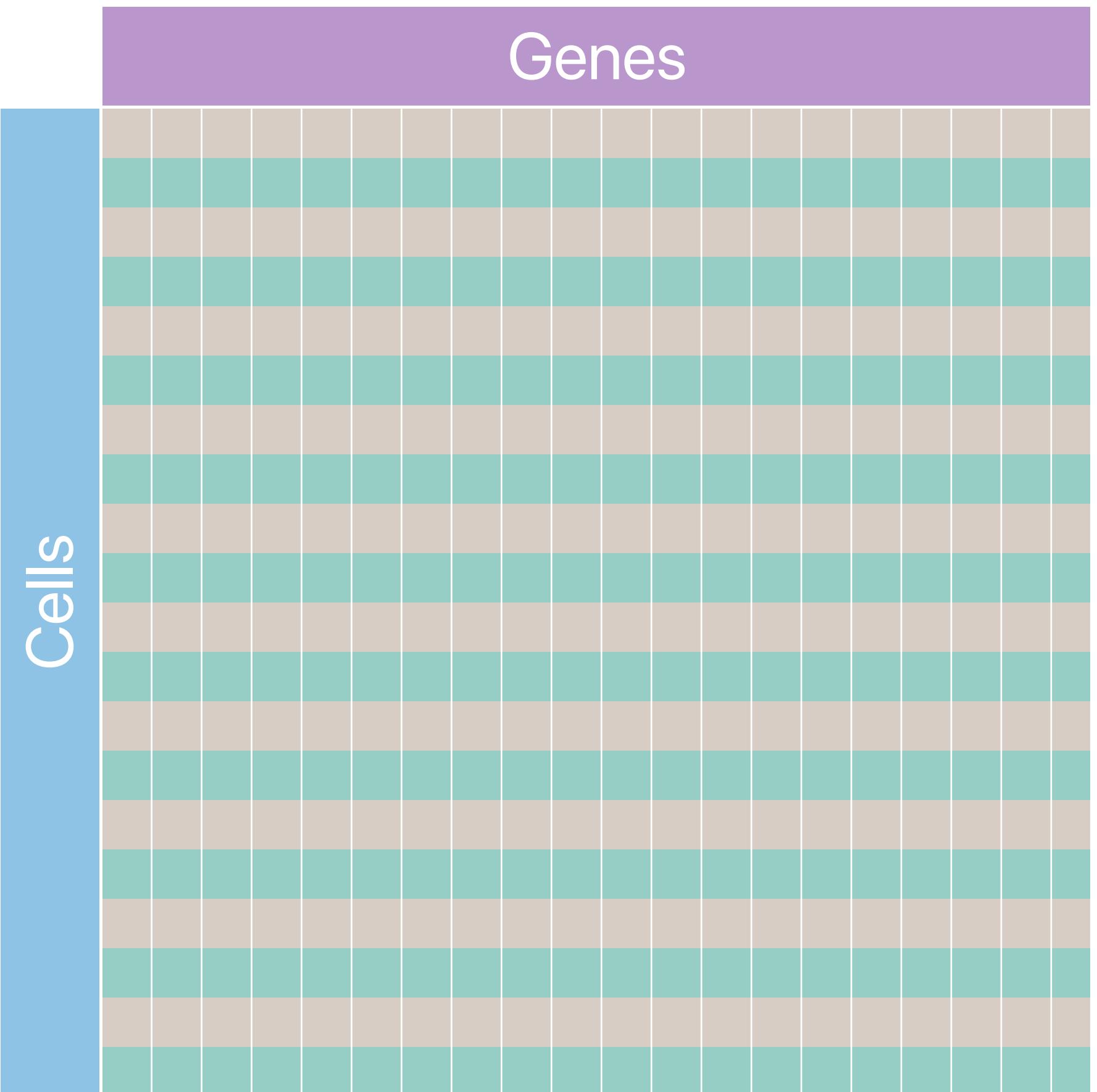


Leiden
Computational Biology Center



Dimensionality Reduction?

- We have huge amounts of complex data
(many cells x many genes)
- We want to reduce complexity for analysis
 - Clustering
 - Dimensionality Reduction



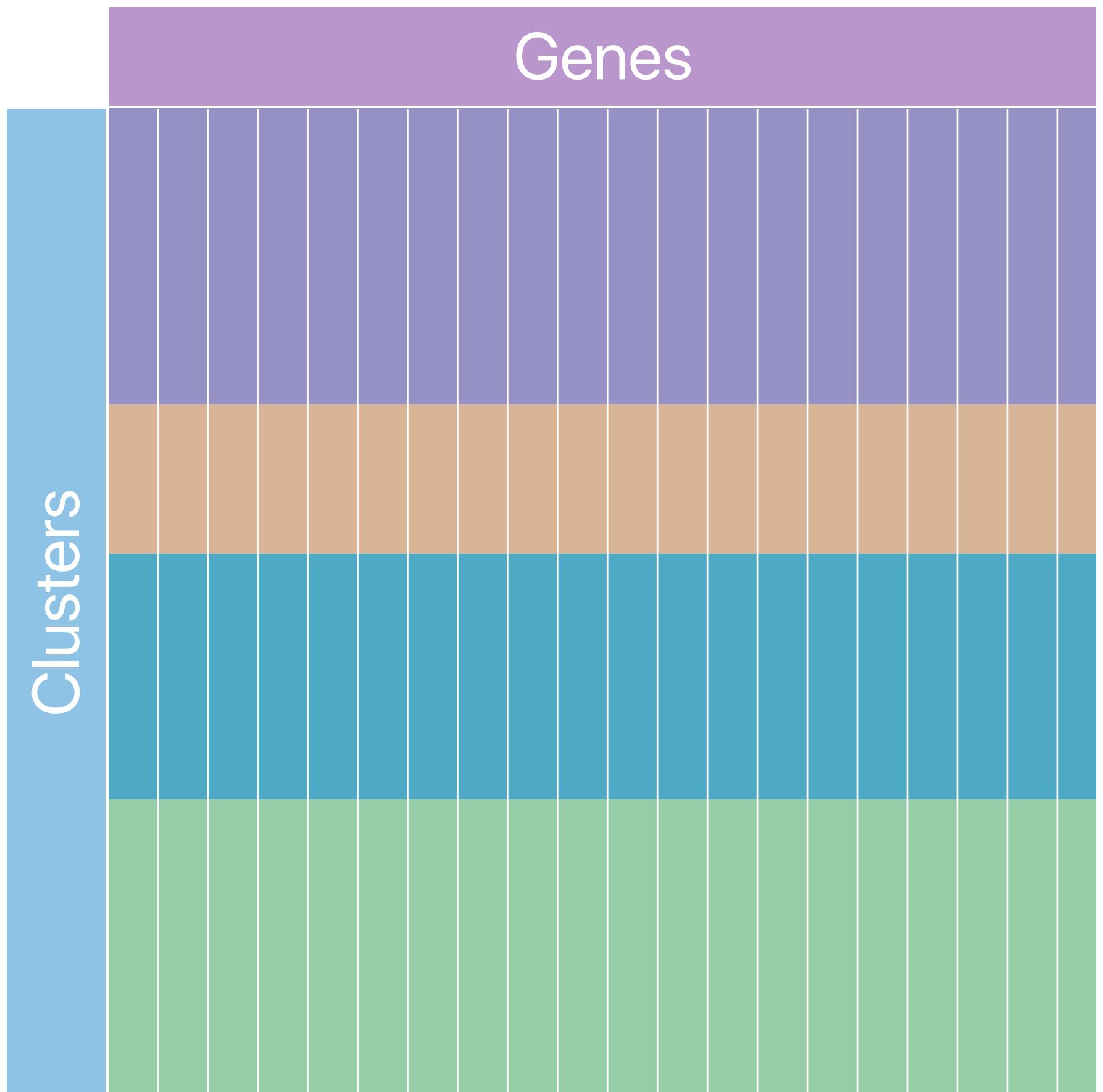
Dimensionality Reduction?

- We have huge amounts of complex data
(many cells x many genes)
- We want to reduce complexity for analysis
 - **Clustering**
 - Dimensionality Reduction



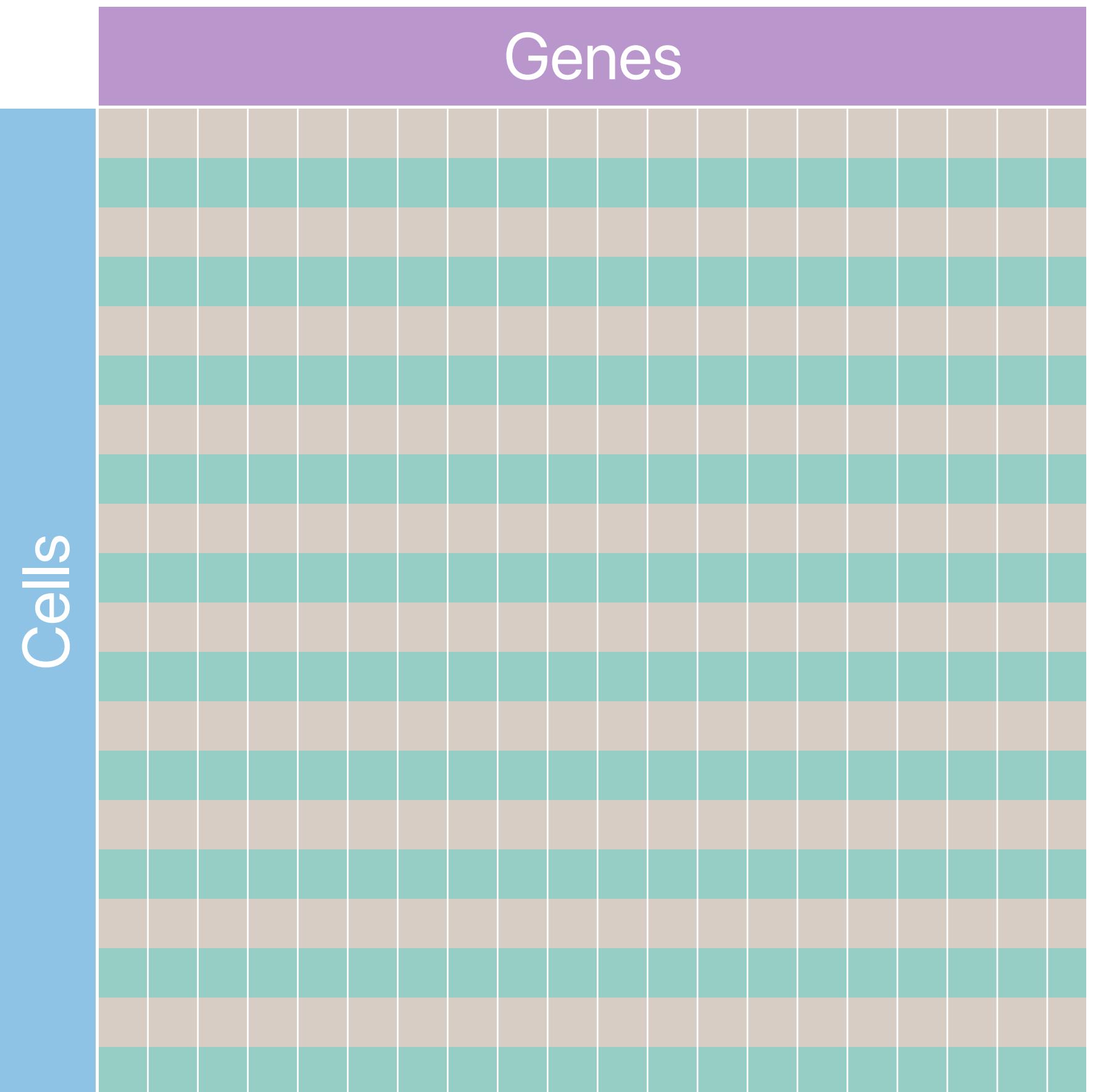
Dimensionality Reduction?

- We have huge amounts of complex data
(many cells x many genes)
- We want to reduce complexity for analysis
 - **Clustering**
 - Dimensionality Reduction



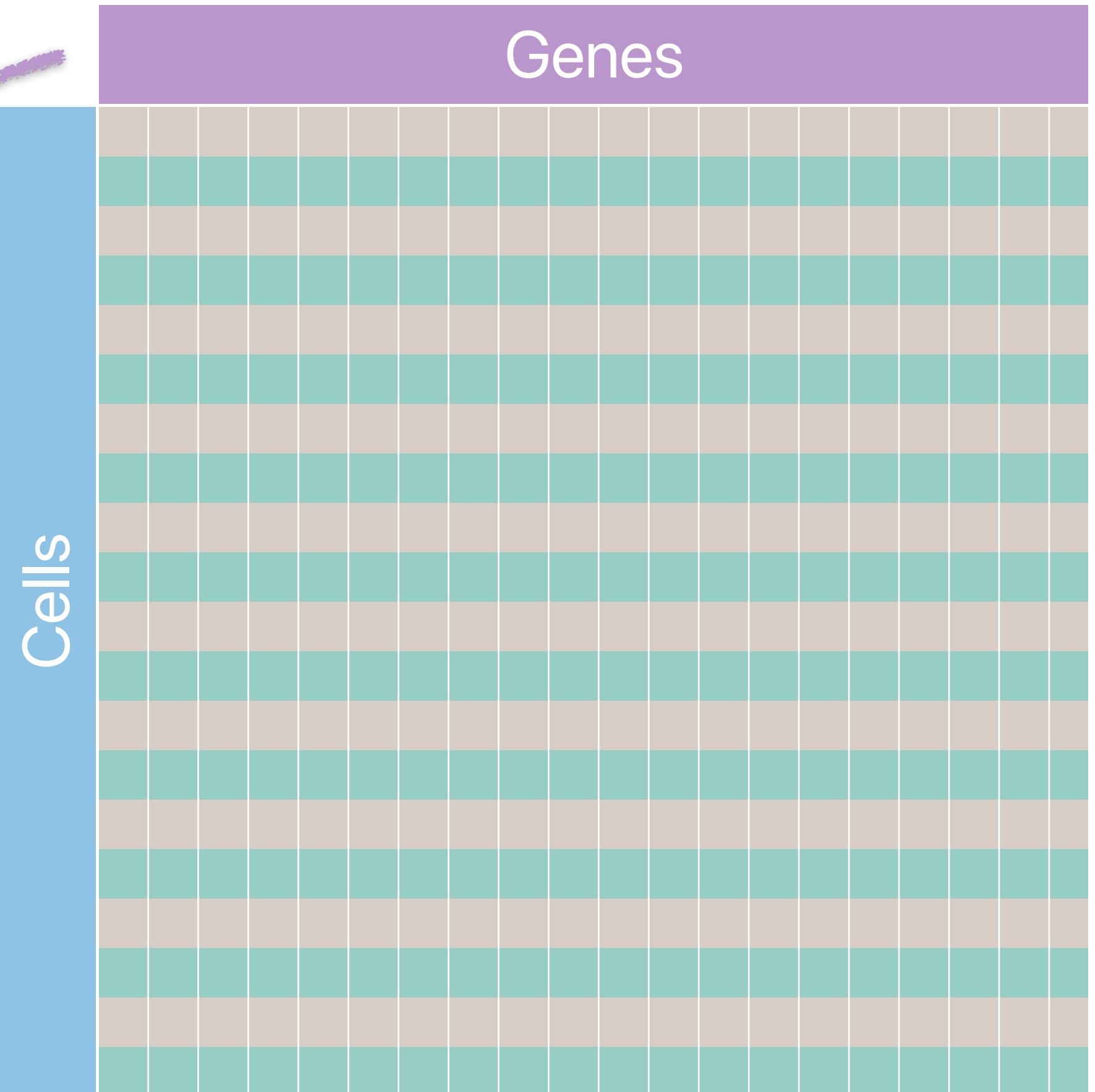
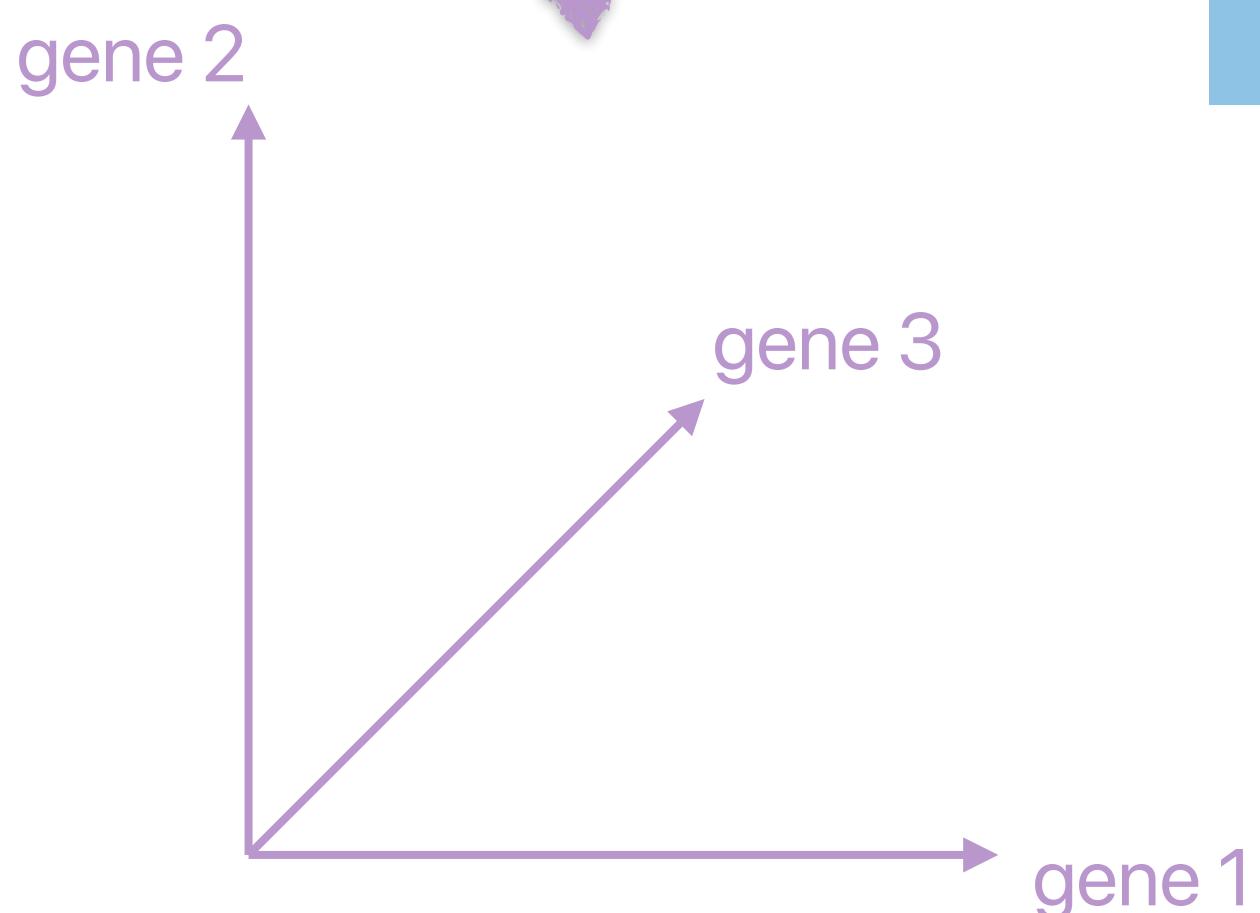
Dimensionality Reduction?

- We have huge amounts of complex data
(many cells x many genes)
- We want to reduce complexity for analysis
 - Clustering
 - **Dimensionality Reduction**



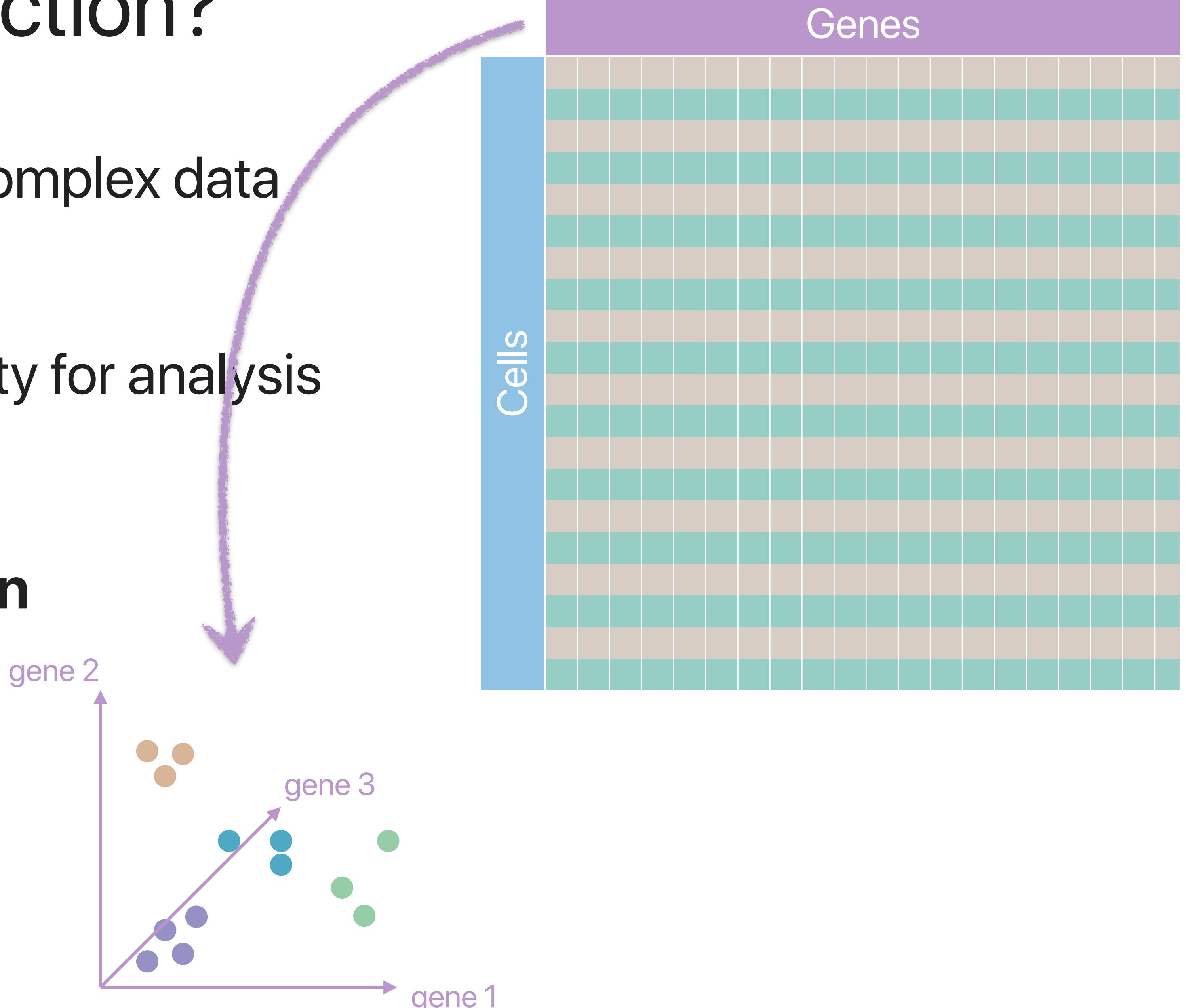
Dimensionality Reduction?

- We have huge amounts of complex data
(many cells x many genes)
- We want to reduce complexity for analysis
 - Clustering
 - **Dimensionality Reduction**



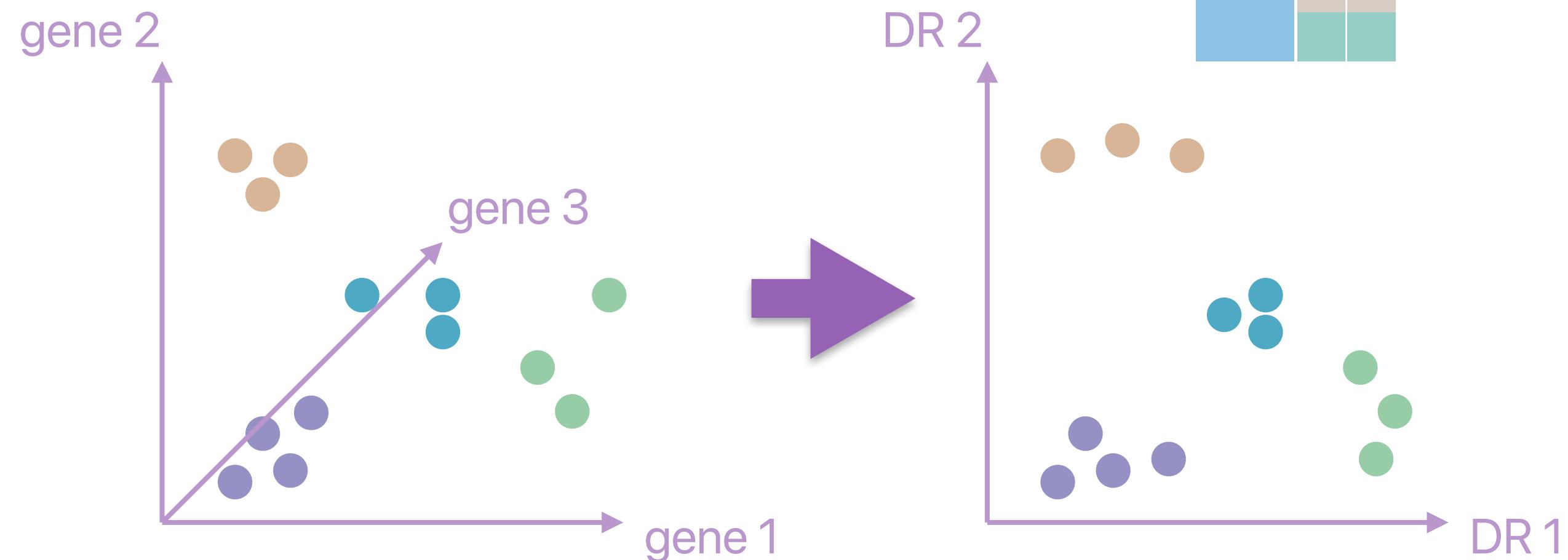
Dimensionality Reduction?

- We have huge amounts of complex data (many cells x many genes)
- We want to reduce complexity for analysis
 - Clustering
 - **Dimensionality Reduction**



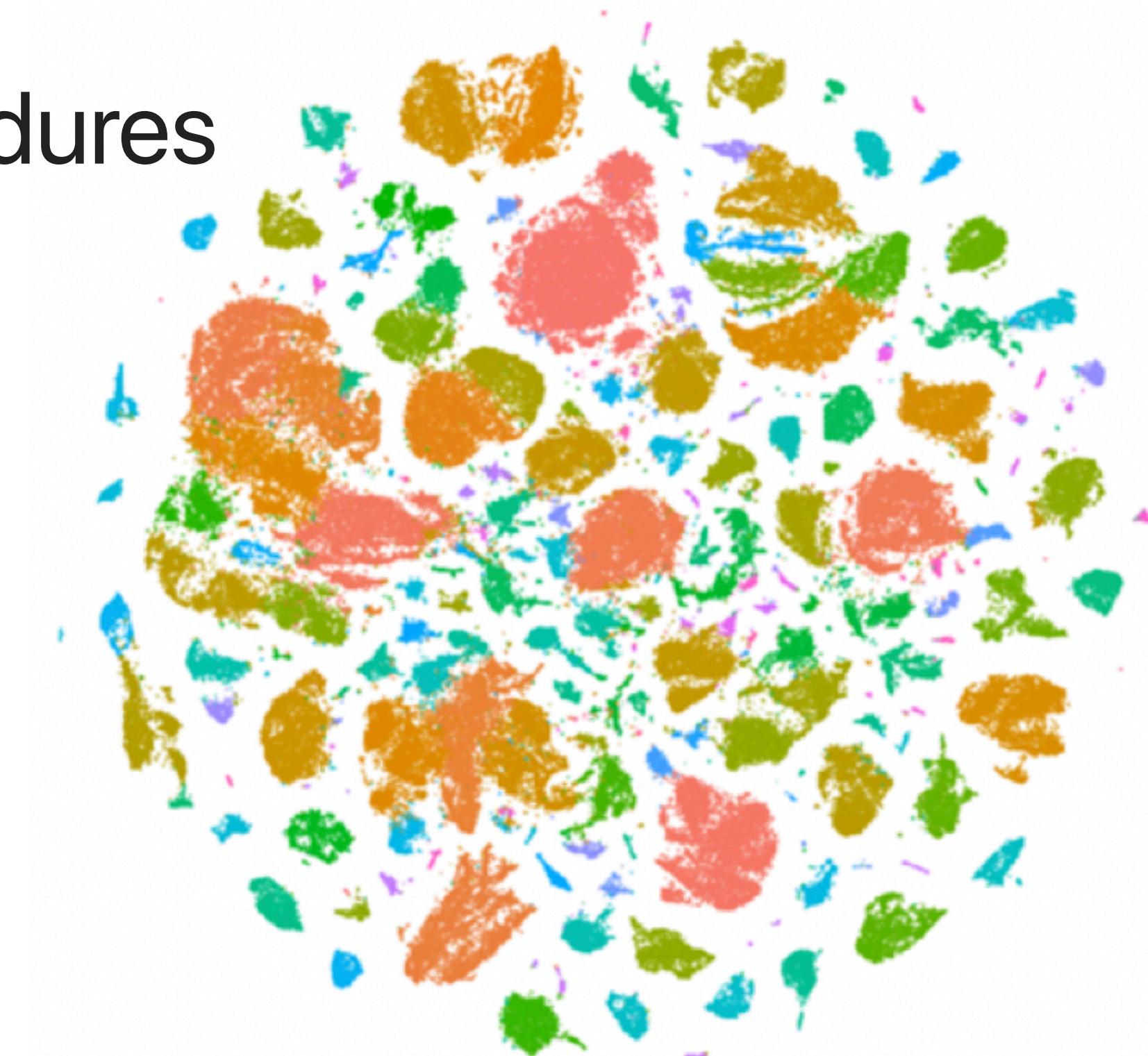
Dimensionality Reduction?

- We have huge amounts of complex data
(many cells x many genes)
- We want to reduce complexity for analysis
 - Clustering
 - **Dimensionality Reduction**

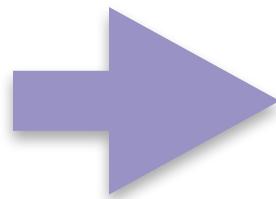


Why Dimensionality Reduction?

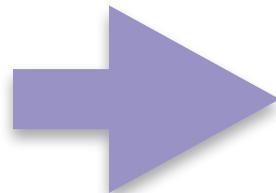
- Simplify complexity, so it becomes easier to work with
 - “Remove” redundancies in the data
 - Identify the most relevant information (find and filter noise)
- Reduce computational time for downstream procedures
 - e.g. clustering
- Visualization



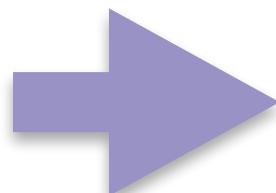
Algorithms



PCA	linear	Matrix Factorization	
ICA	linear	Matrix Factorization	
MDS	non-linear	Matrix Factorization	
cPCA	non-linear	Matrix Factorization	https://doi.org/10.1038/s41467-018-04608-8
ZIFA	non-linear	Matrix Factorization	https://doi.org/10.1186/s13059-015-0805-z
ZINB-WaVE	non-linear	Matrix Factorization	https://doi.org/10.1038/s41467-017-02554-5



Diffusion maps	non-linear	Graph-based	https://doi.org/10.1073/pnas.0500334102
Isomap	non-linear	Graph-based	https://doi.org/10.1126/science.290.5500.2319
t-SNE	non-linear	Graph-based	https://lvdmaaten.github.io/publications/papers/JMLR_2008.pdf
LargeVis	non-linear	Graph-based	arXiv:1602.00370
UMAP	non-linear	Graph-based	arXiv:1802.03426
PHATE	non-linear	Graph-based	https://doi.org/10.1101/120378

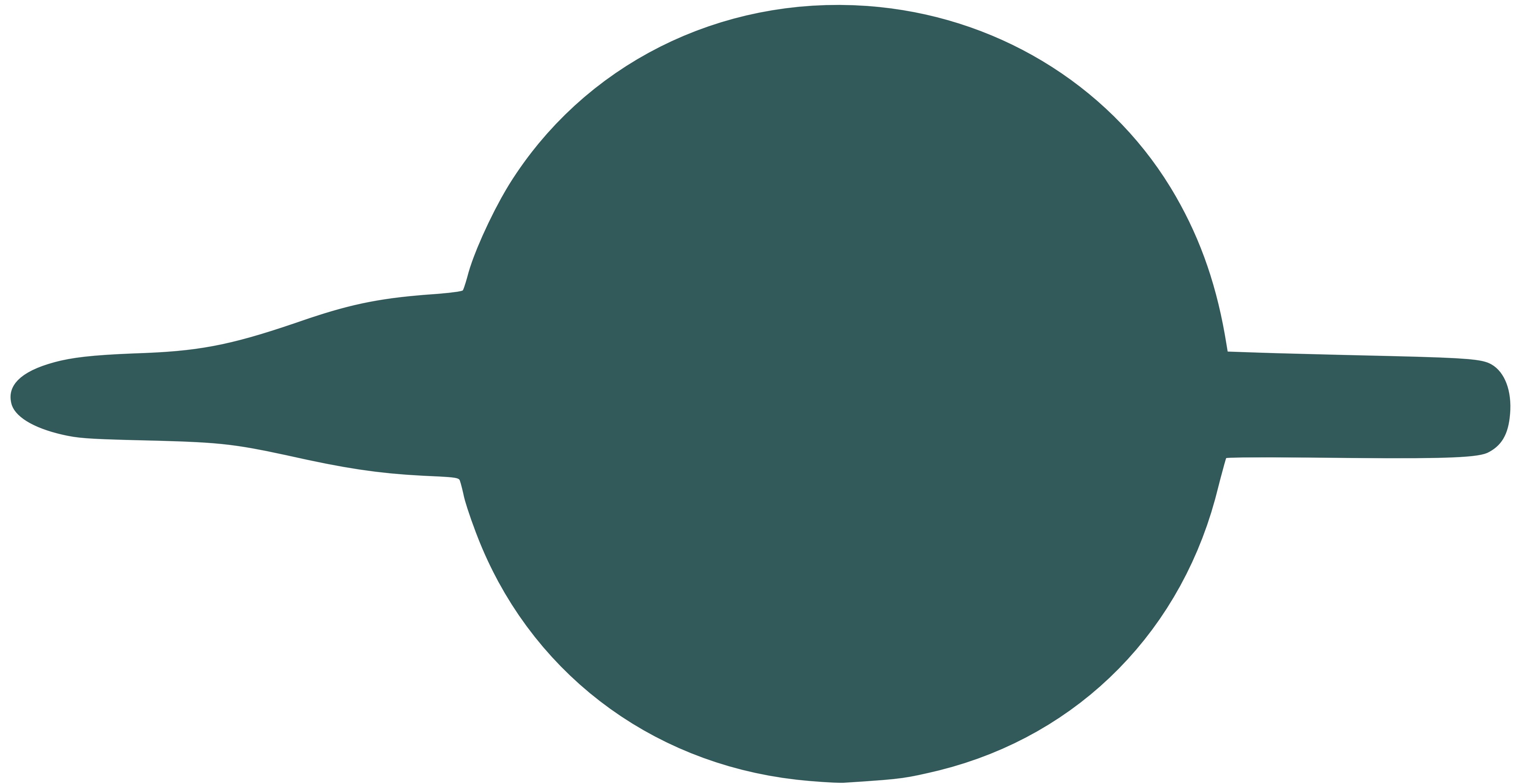


scvis	non-linear	Autoencoder	https://doi.org/10.1038/s41467-018-04368-5
VASC	non-linear	Autoencoder	https://doi.org/10.1016/j.gpb.2018.08.003

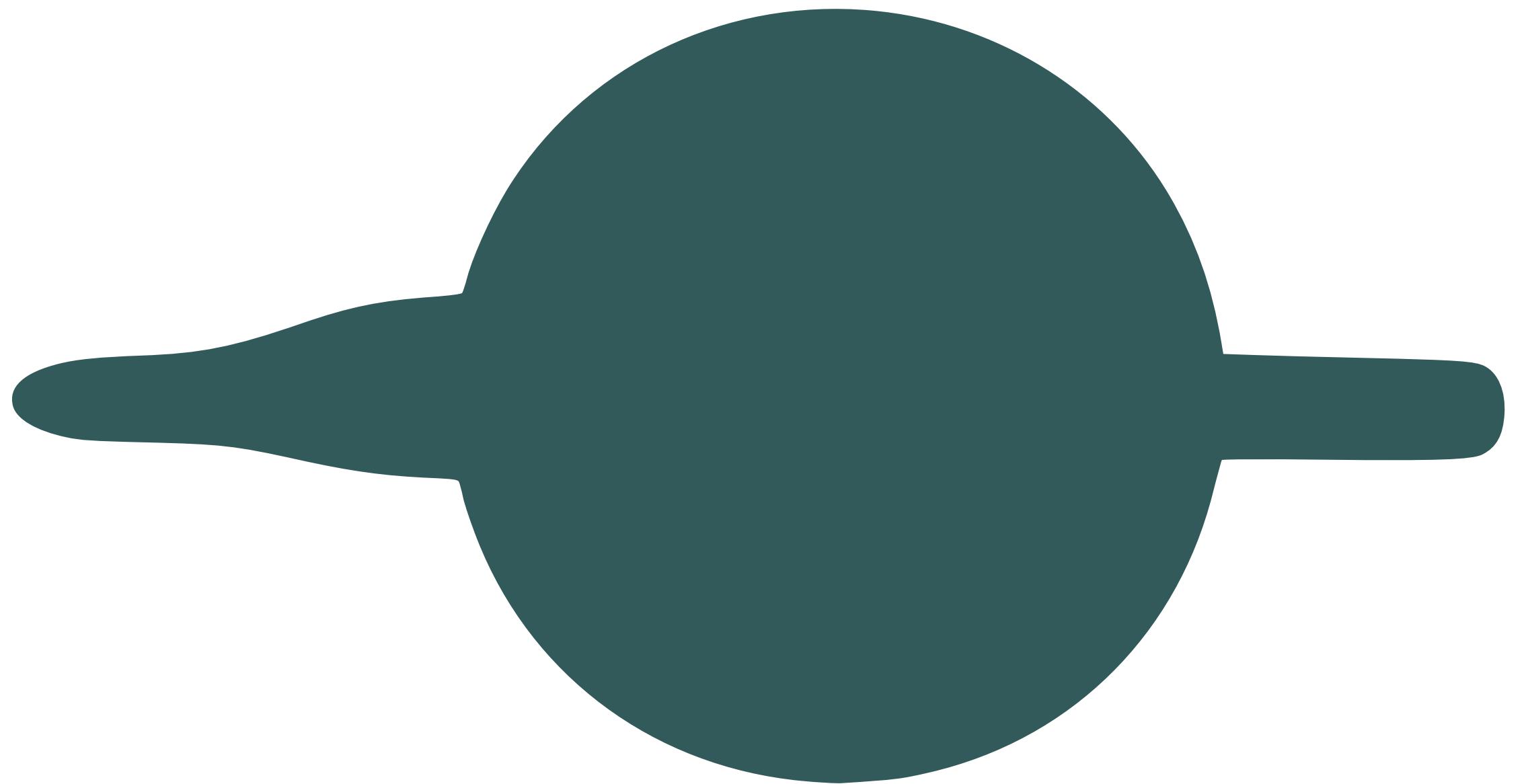
PCA

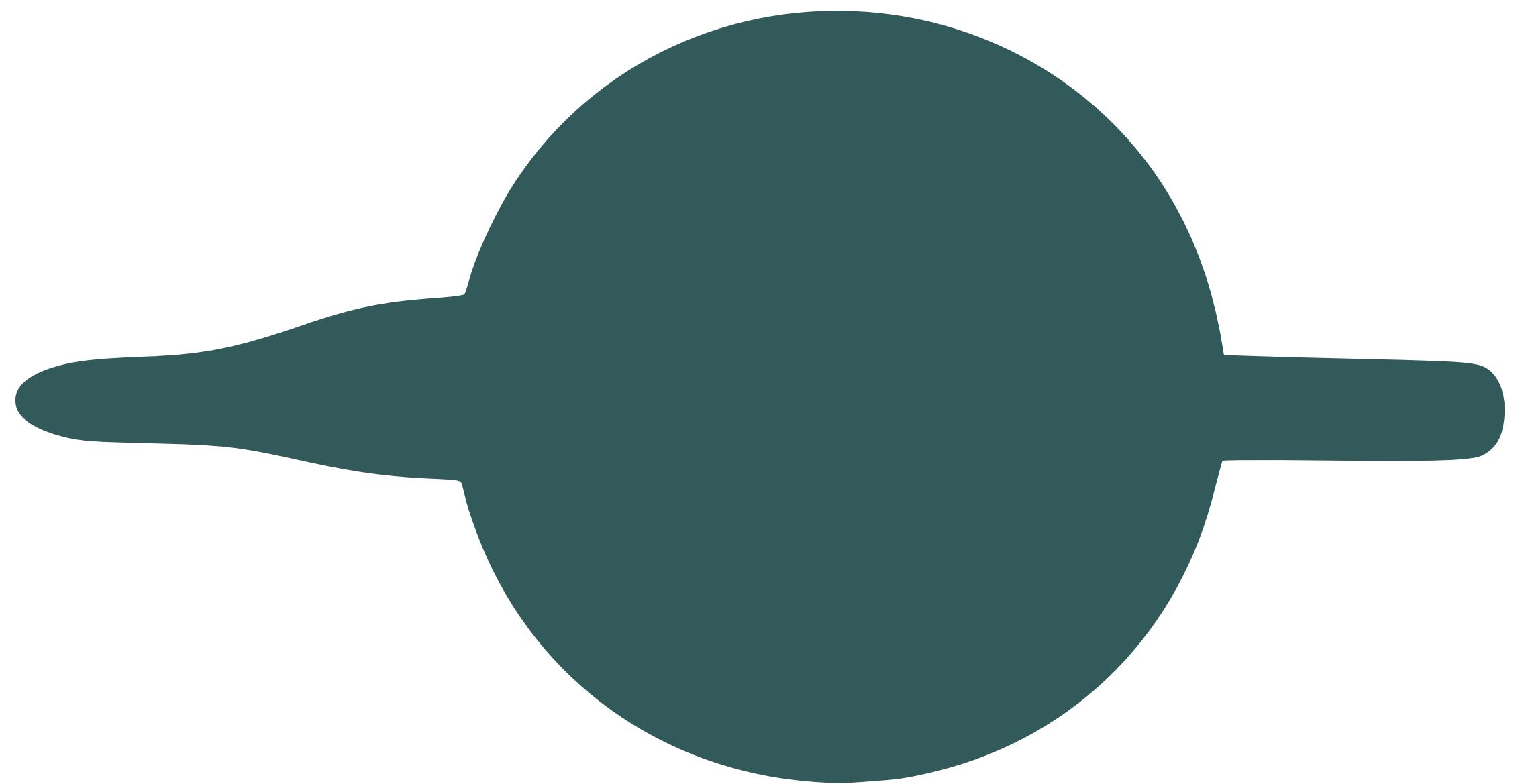
Principal Component Analysis











PCA - Intuition

- Given a dataset, compute/measure a number of features
- These features represent an N-dimensional problem
- PCA finds a new coordinate system obtained from the previous one by translation and rotation only
change the point of view
- Moves the center of the coordinate system with center of the data
- Moves the x-axis into the principal axis of variation
- Orders axes by amount of variation (importance)

PCA in Brief

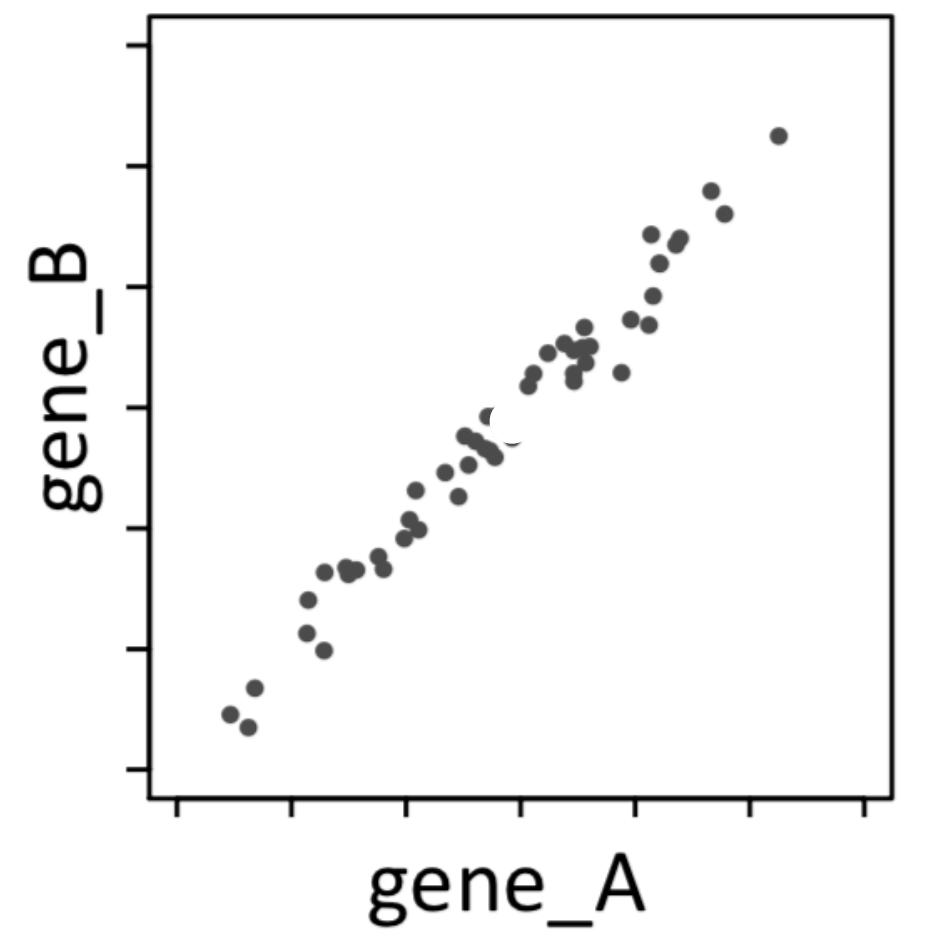
$$\mathbf{M} = \mathbf{U} \Sigma \mathbf{V}^*$$

Original data Center Rotation Scaling Planar rotation

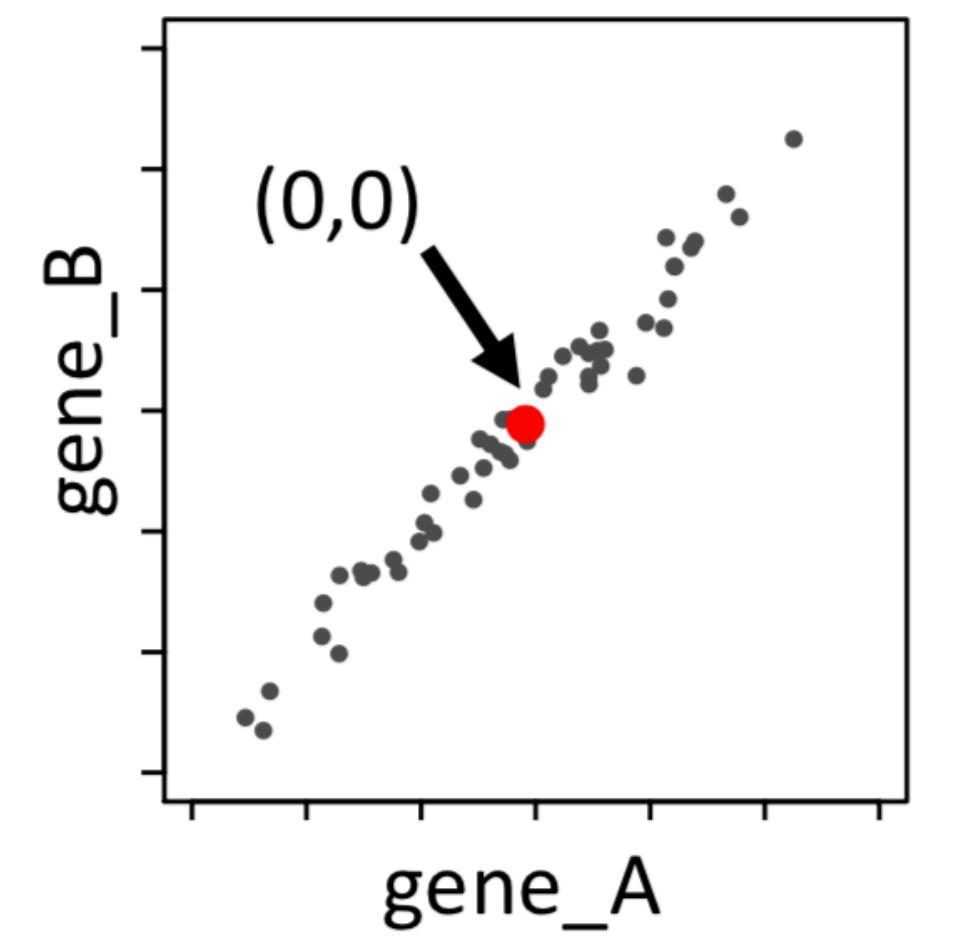
genes PCs genes PCs

cells cells cells cells

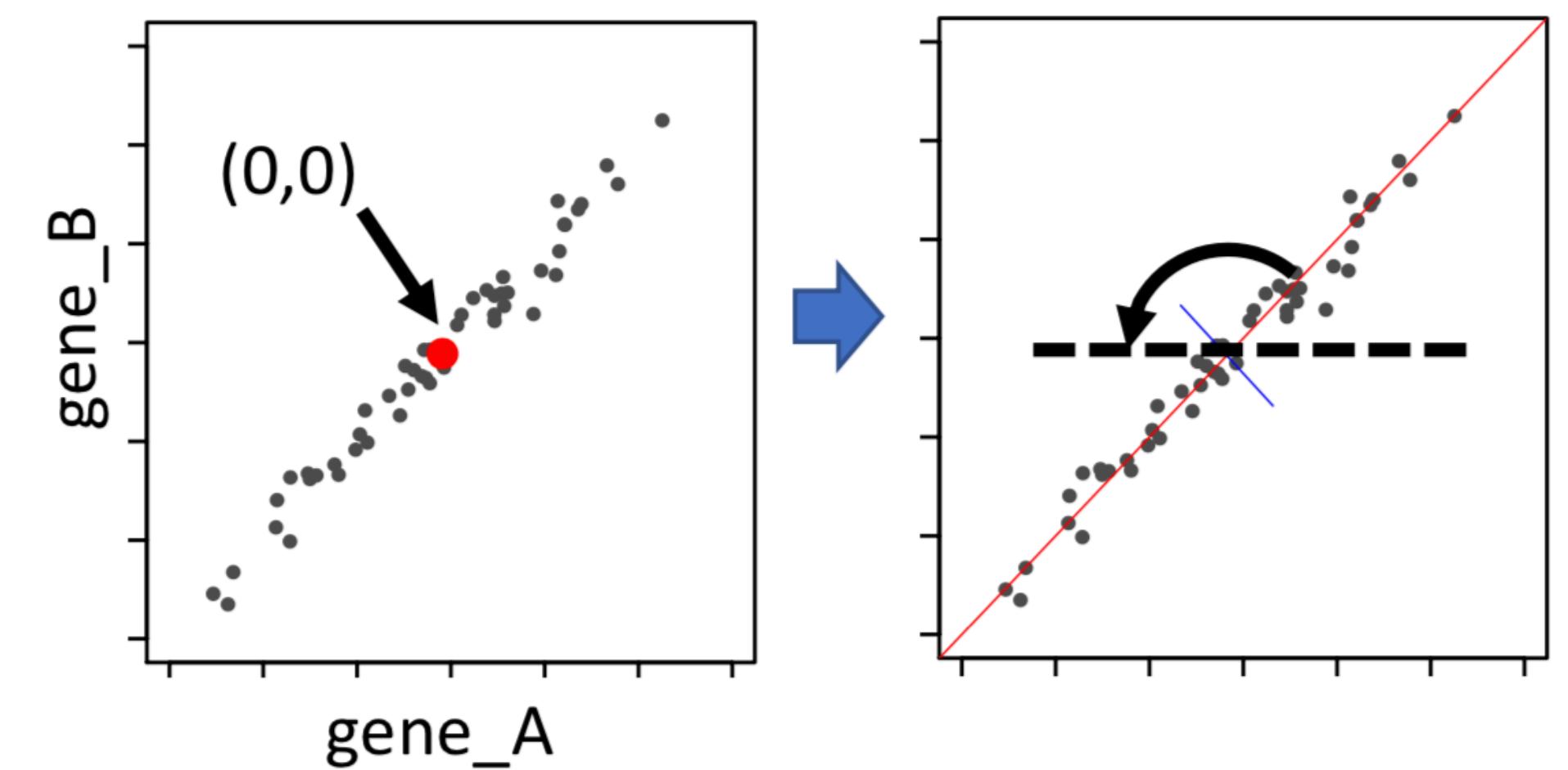
PCA in Brief



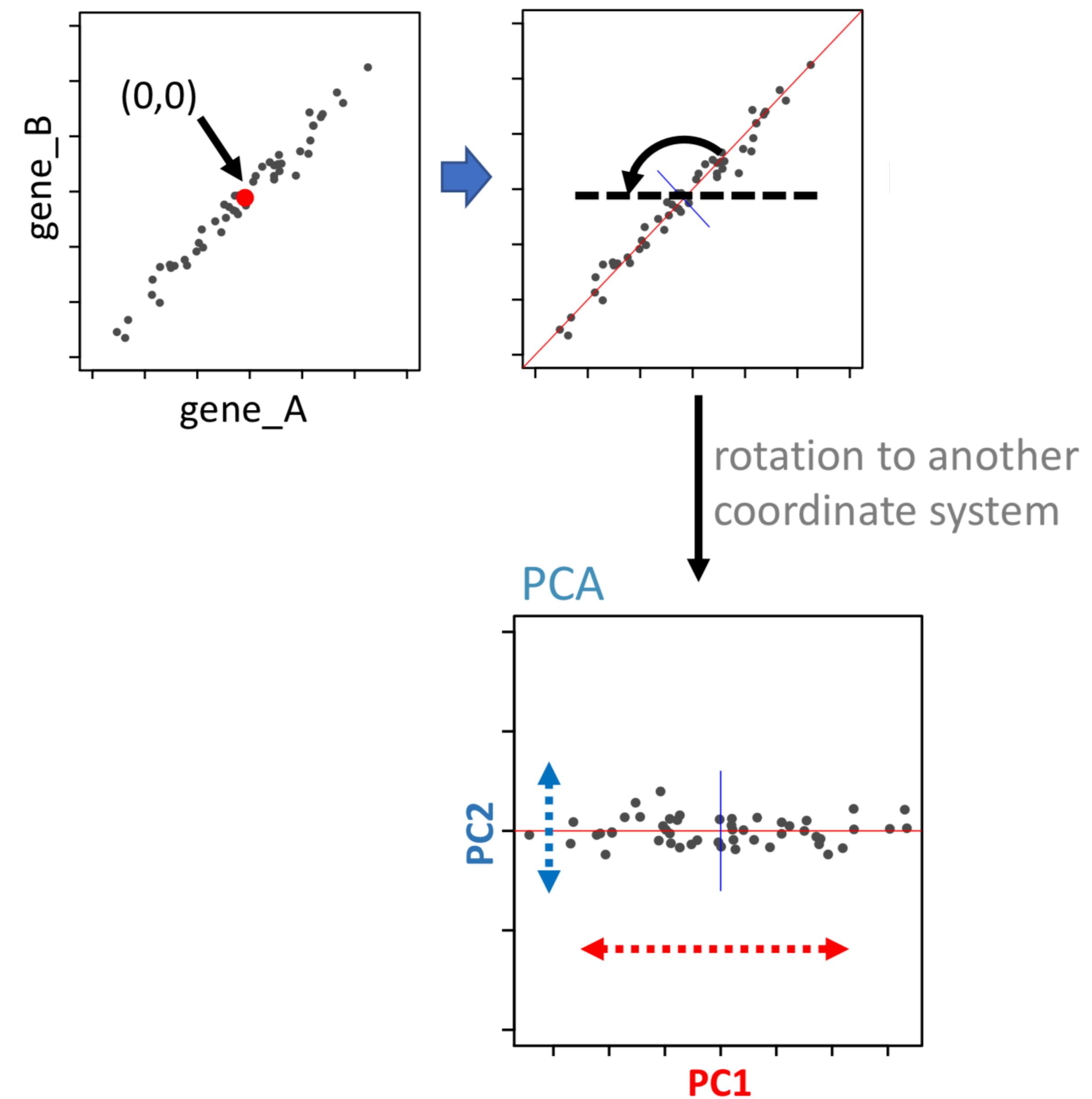
PCA in Brief



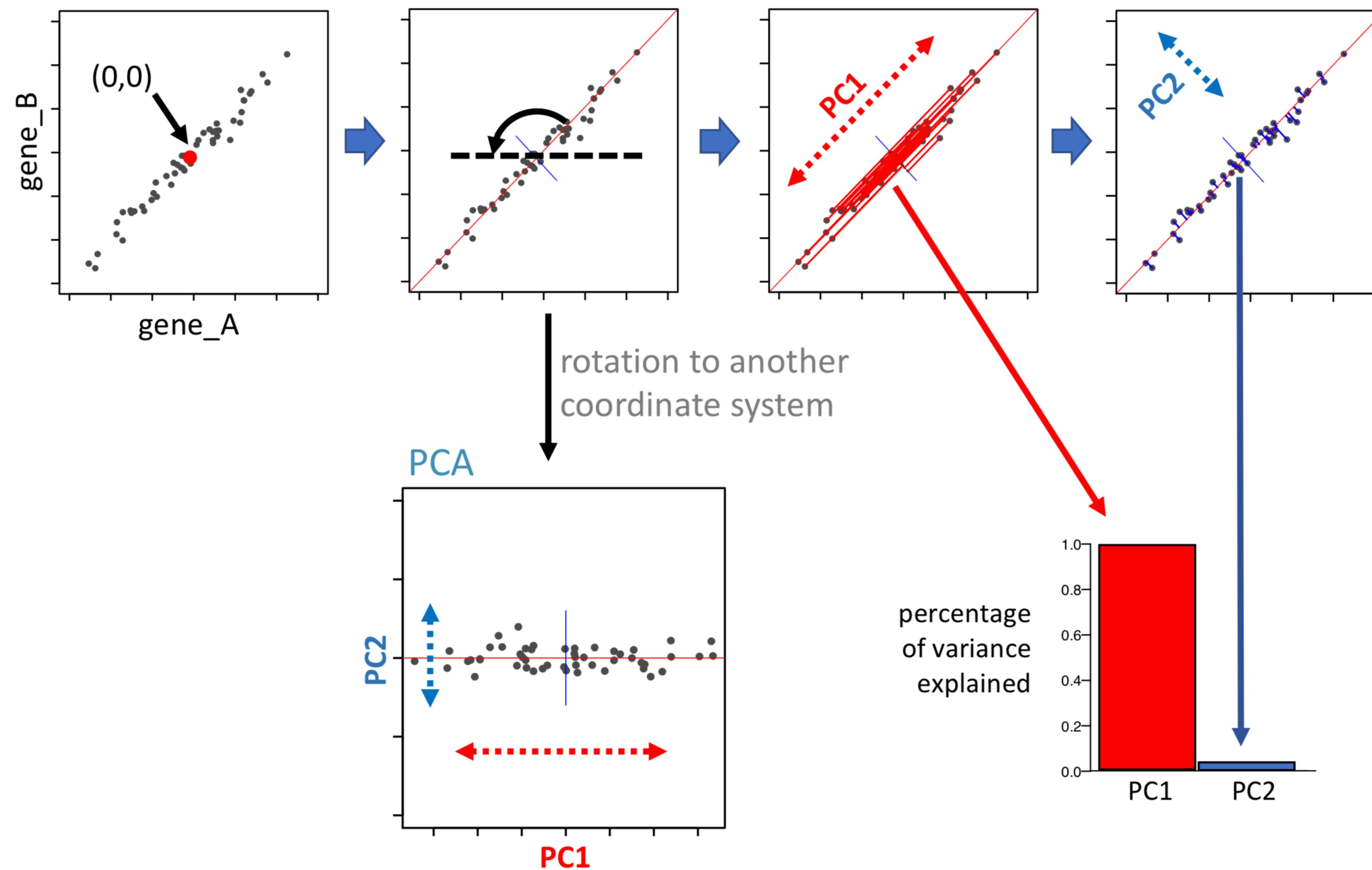
PCA in Brief



PCA in Brief



PCA in Brief

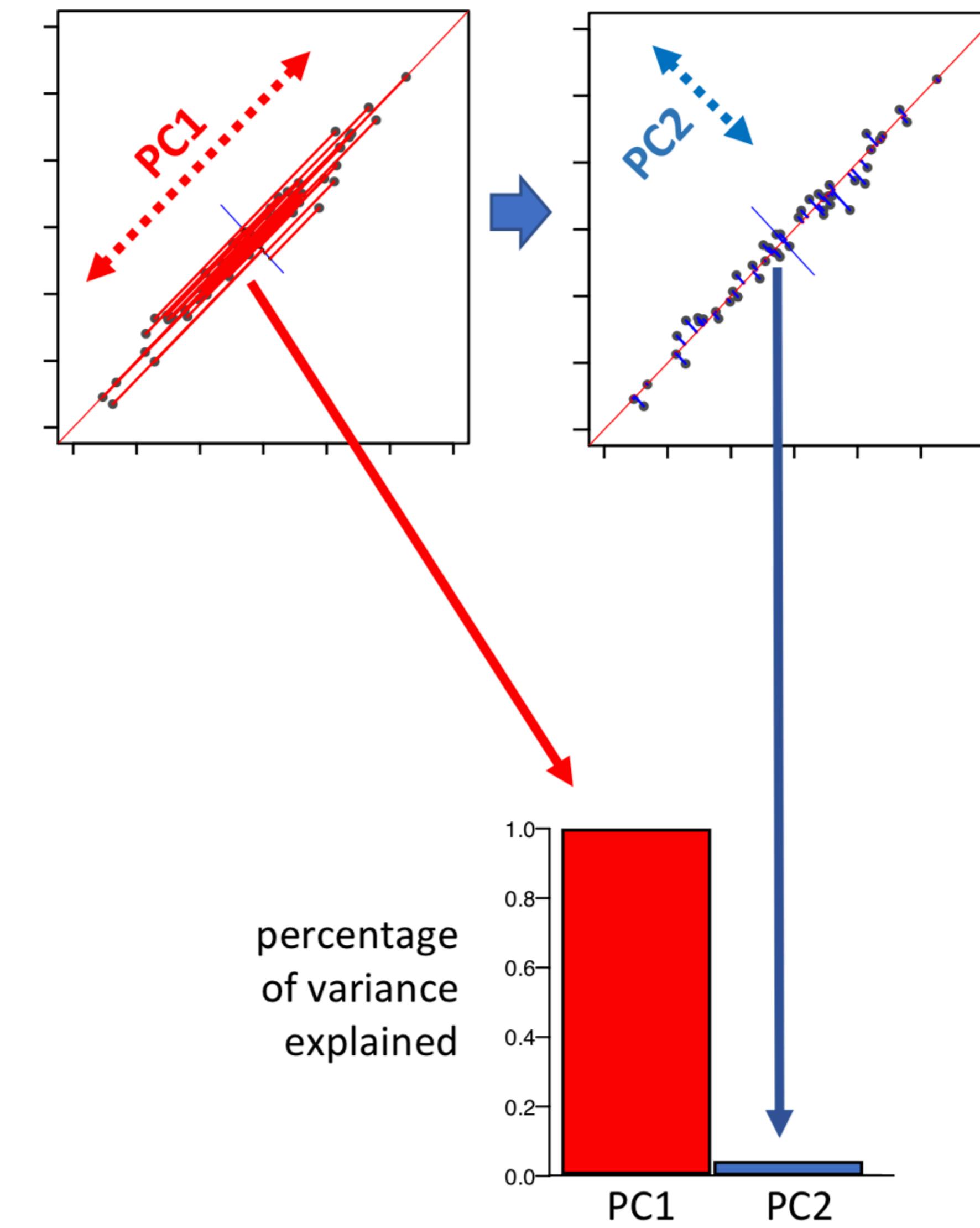


PCA in Brief

PC1 explains >98% of the variance

1 PC thus represents 2 genes very well
"Removing" redundancy

PC2 is nearly insignificant in this example
Could be disregarded

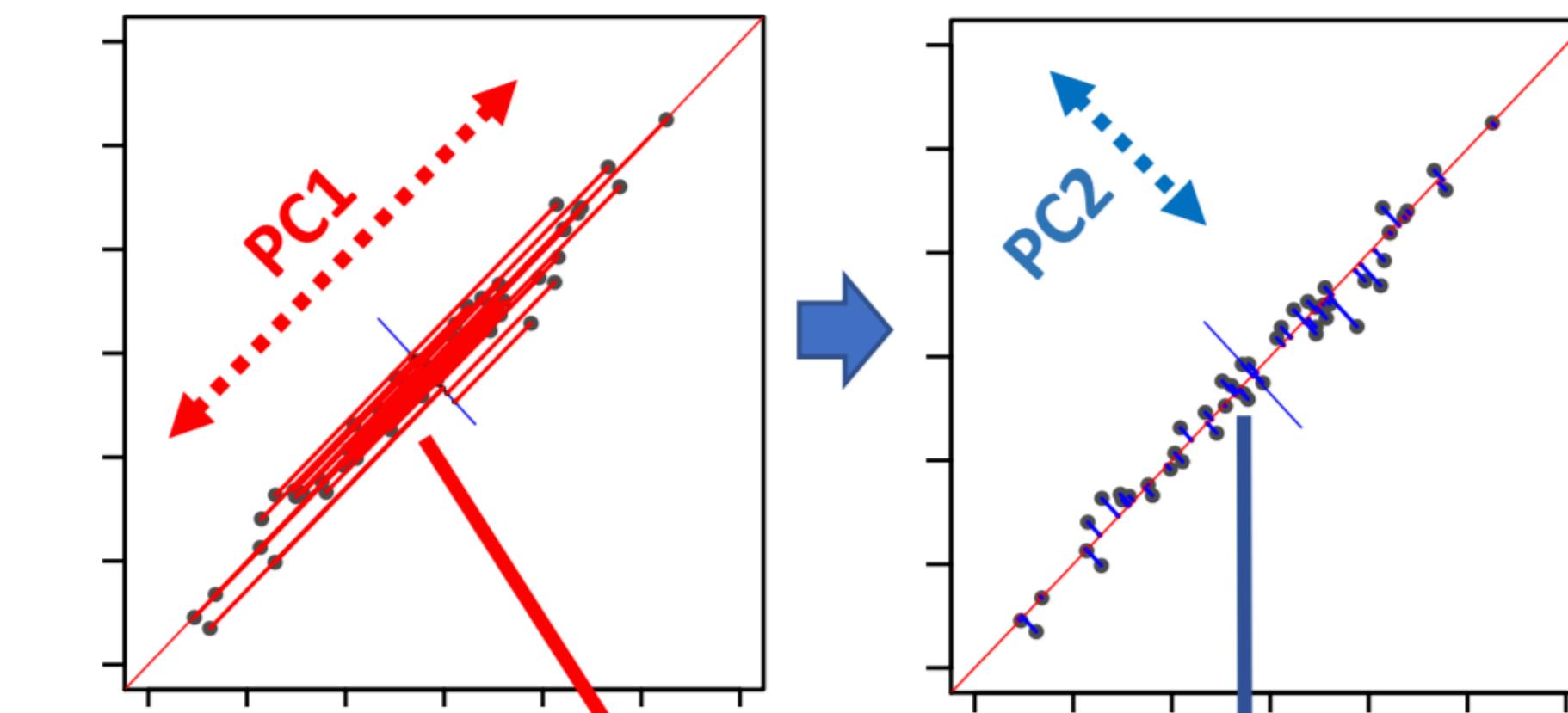


PCA in Brief

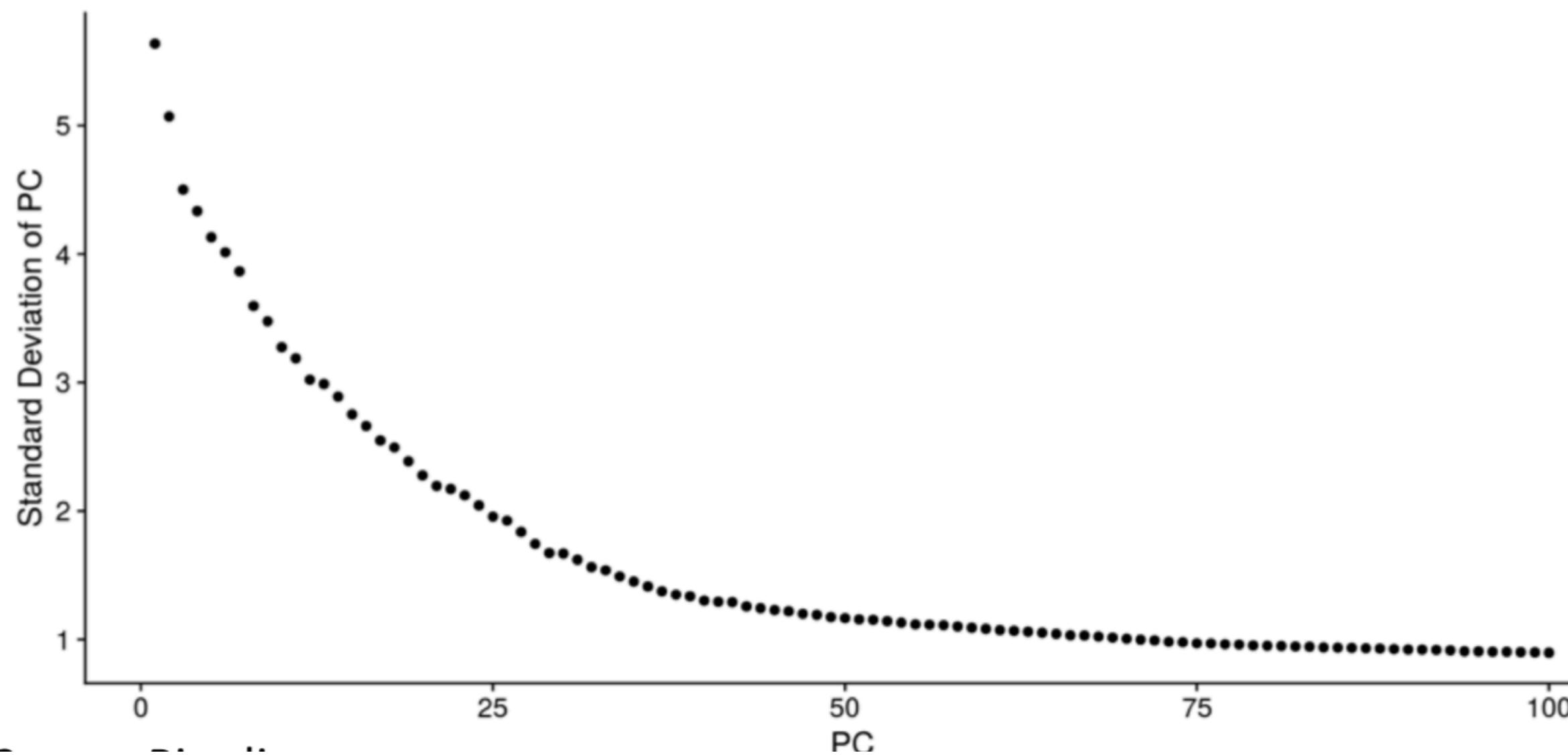
PC1 explains >98% of the variance

1 PC thus represents 2 genes very well
"Removing" redundancy

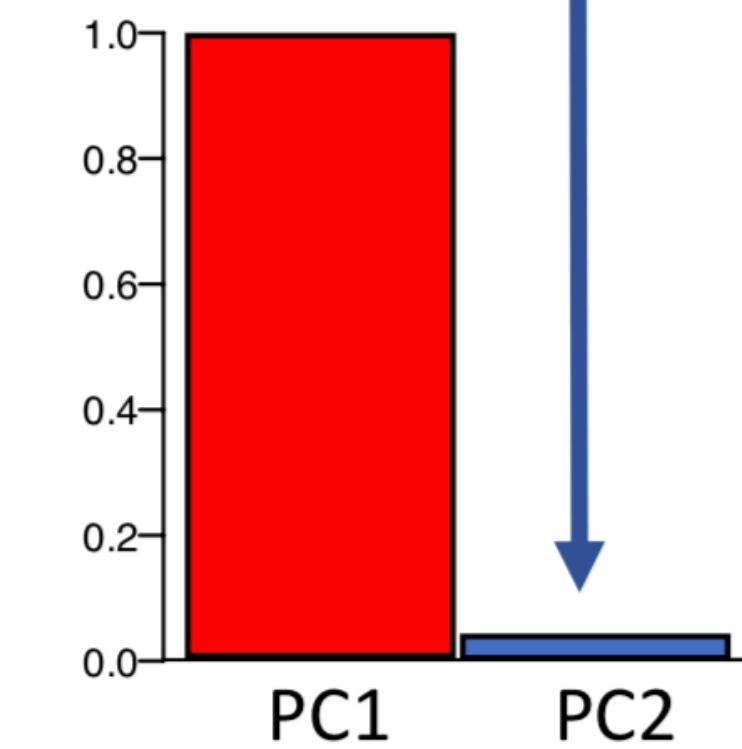
PC2 is nearly insignificant in this example
Could be disregarded



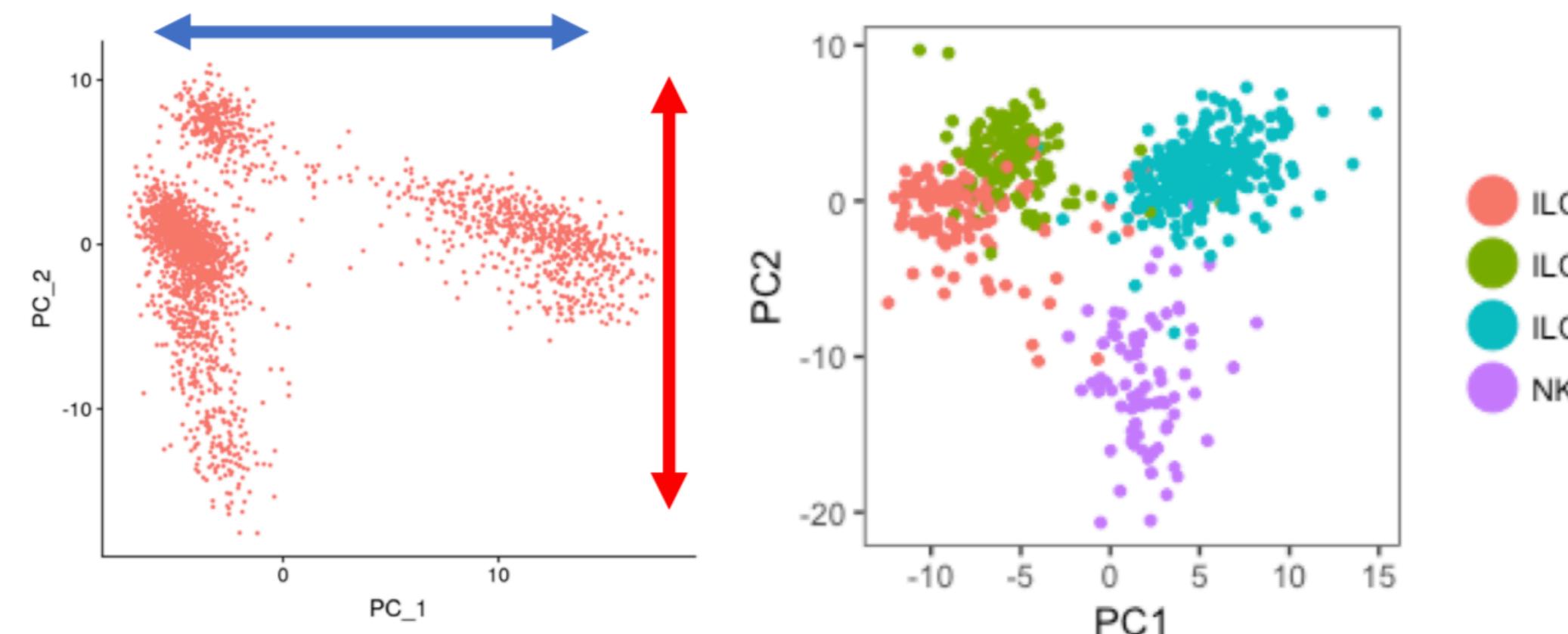
In real life ...



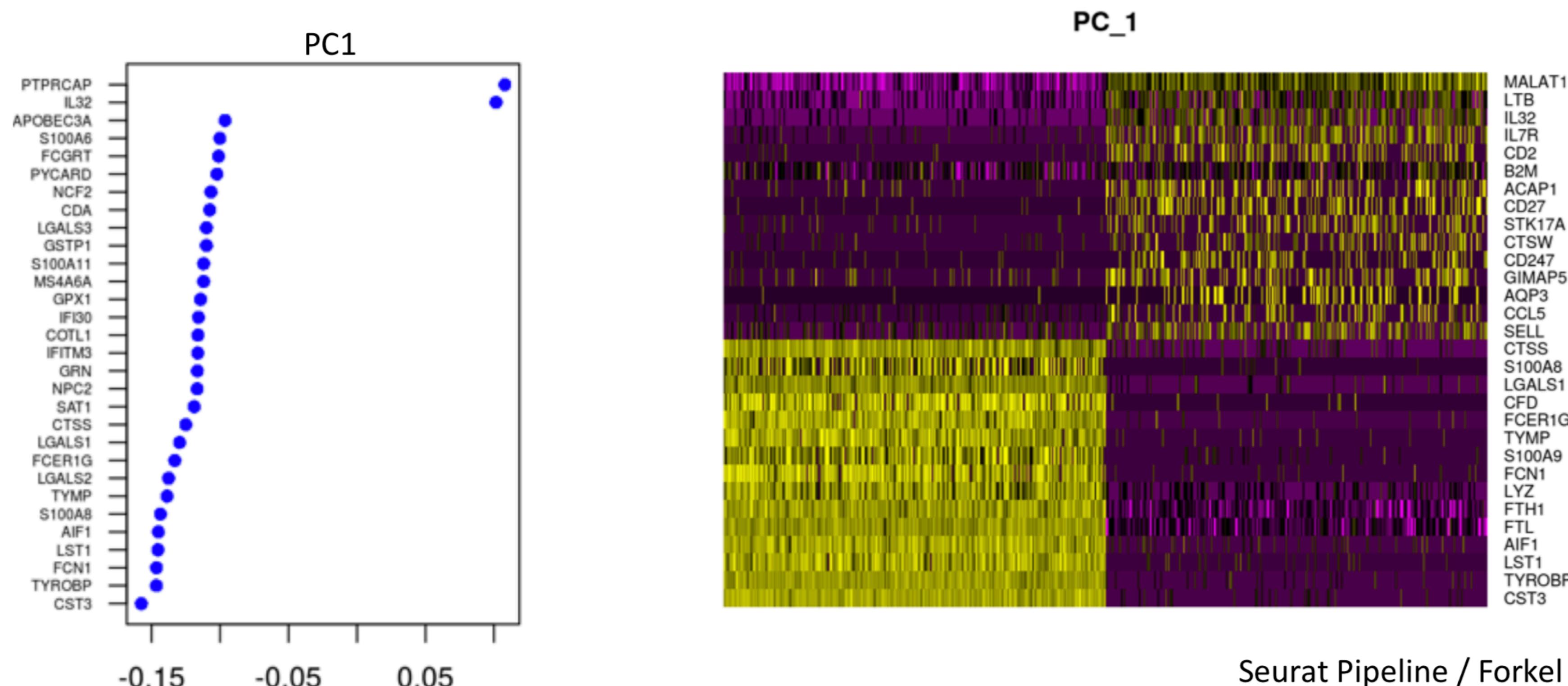
percentage
of variance
explained



PCA in Brief



PC1 and PC2 are commonly correlated to sequencing depth and cell heterogeneity/complexity
(but not always ...)



Summary: PCA

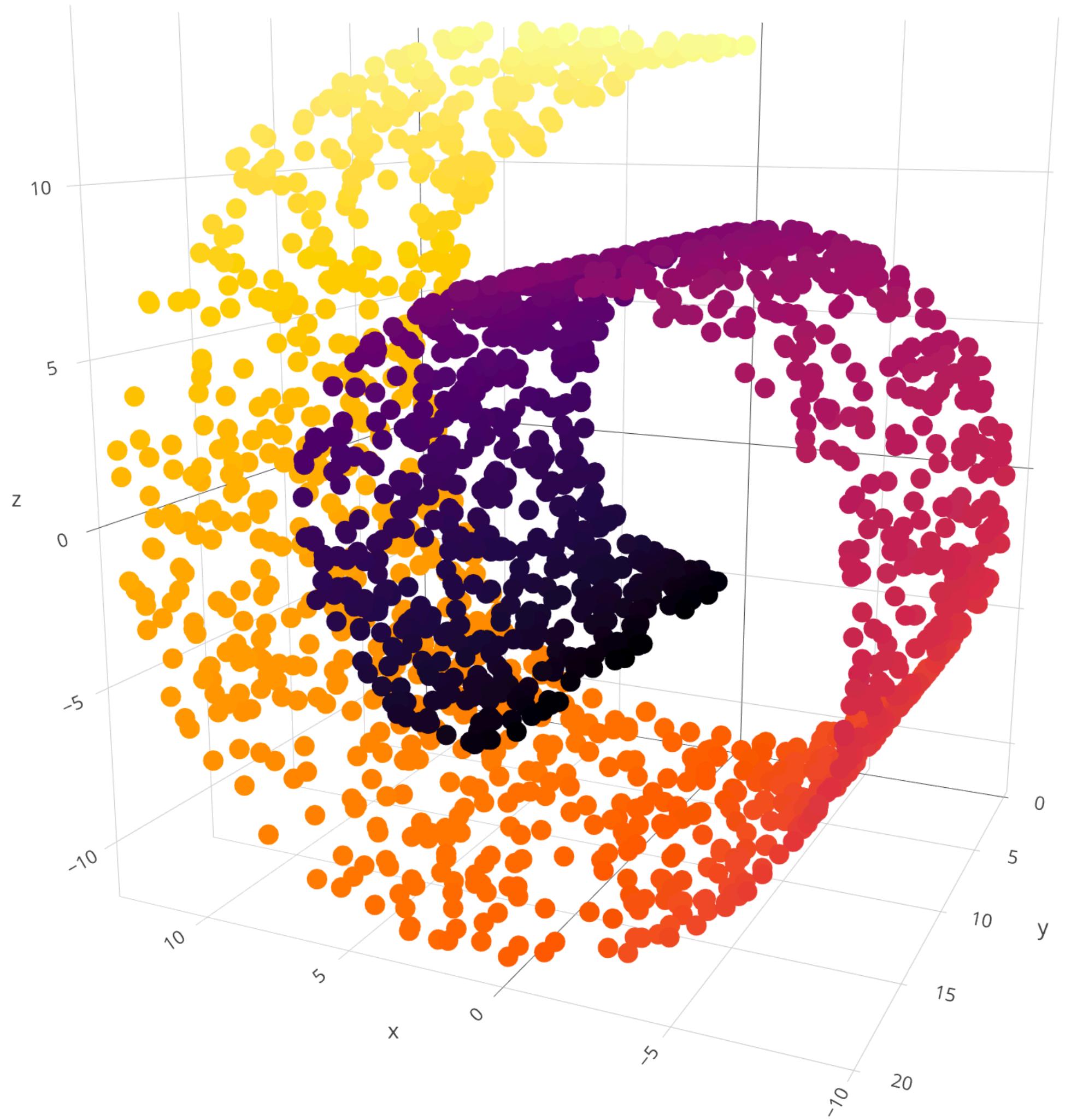
- LINEAR method of dimensionality reduction
- The TOP principal components contain higher variance from the data
- Can be used as FILTERING, by selecting only the top significant PCs
- *It is an interpretable/parametric dimensionality reduction*
- **Problems:**
 - It performs poorly to separate cells in 0-inflated data types
 - Cell sizes and sequencing depth are usually captured in the top PCs



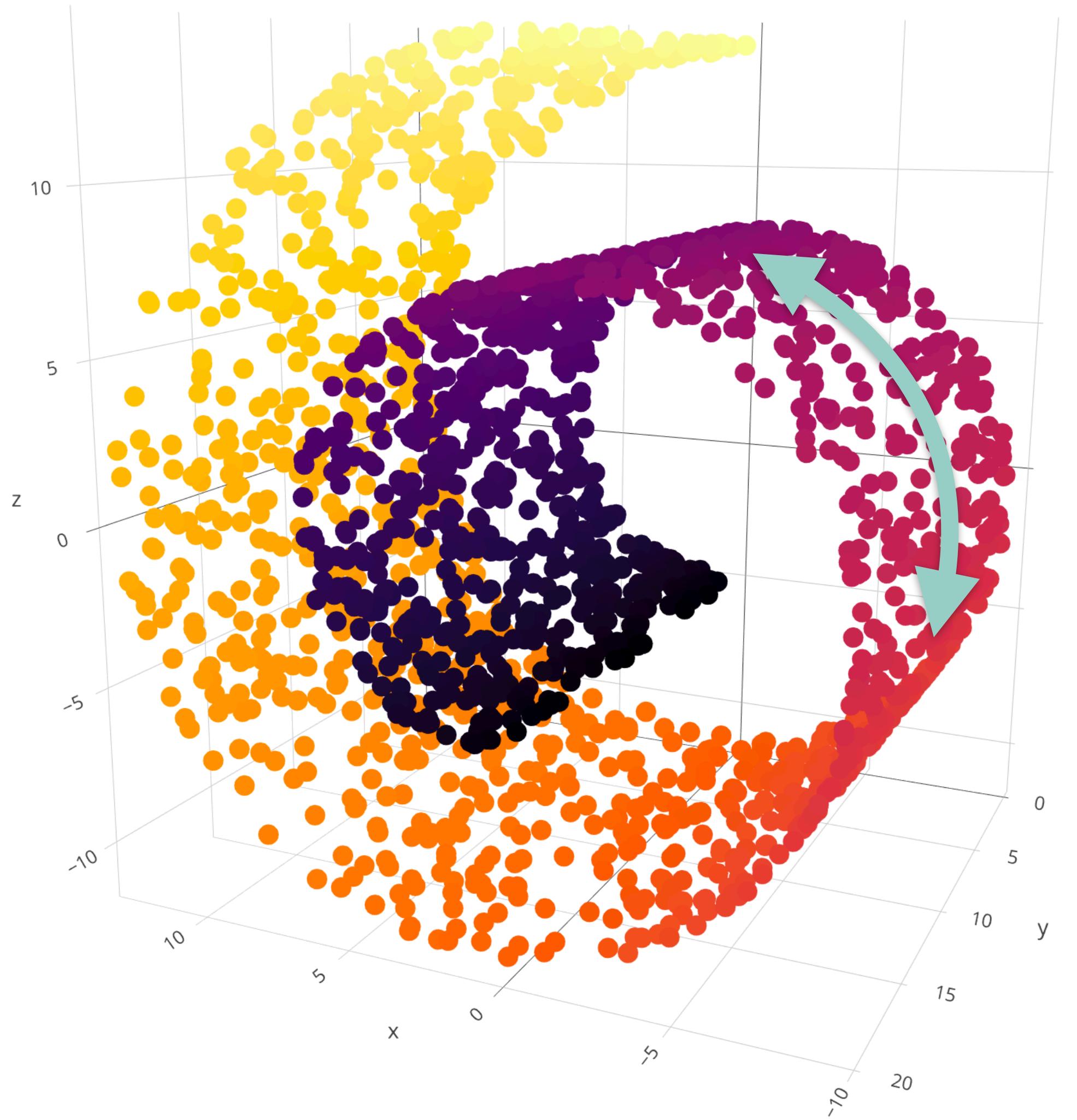
t-SNE

t-distributed Stochastic Neighborhood Embedding

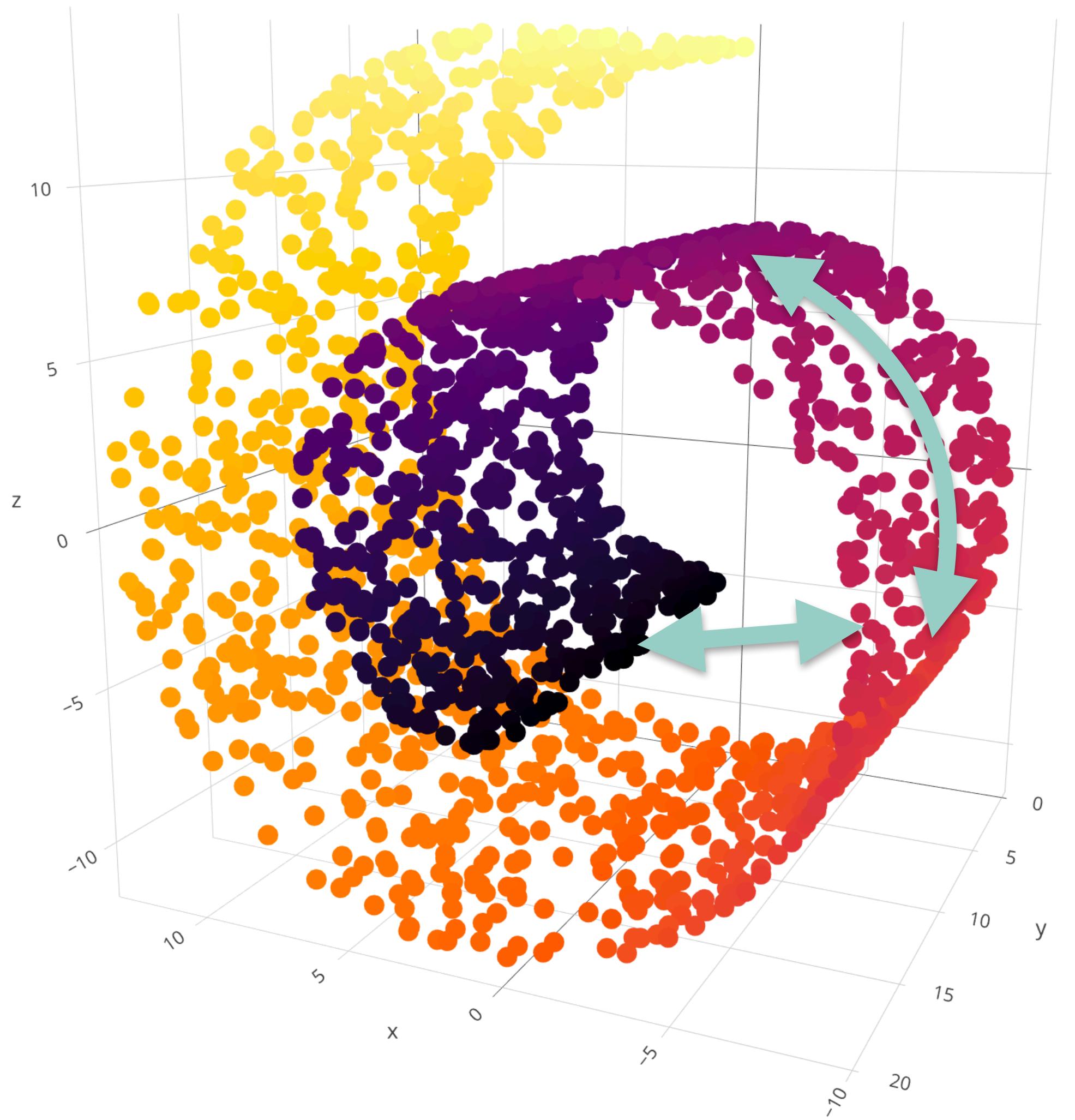
Manifold Learning



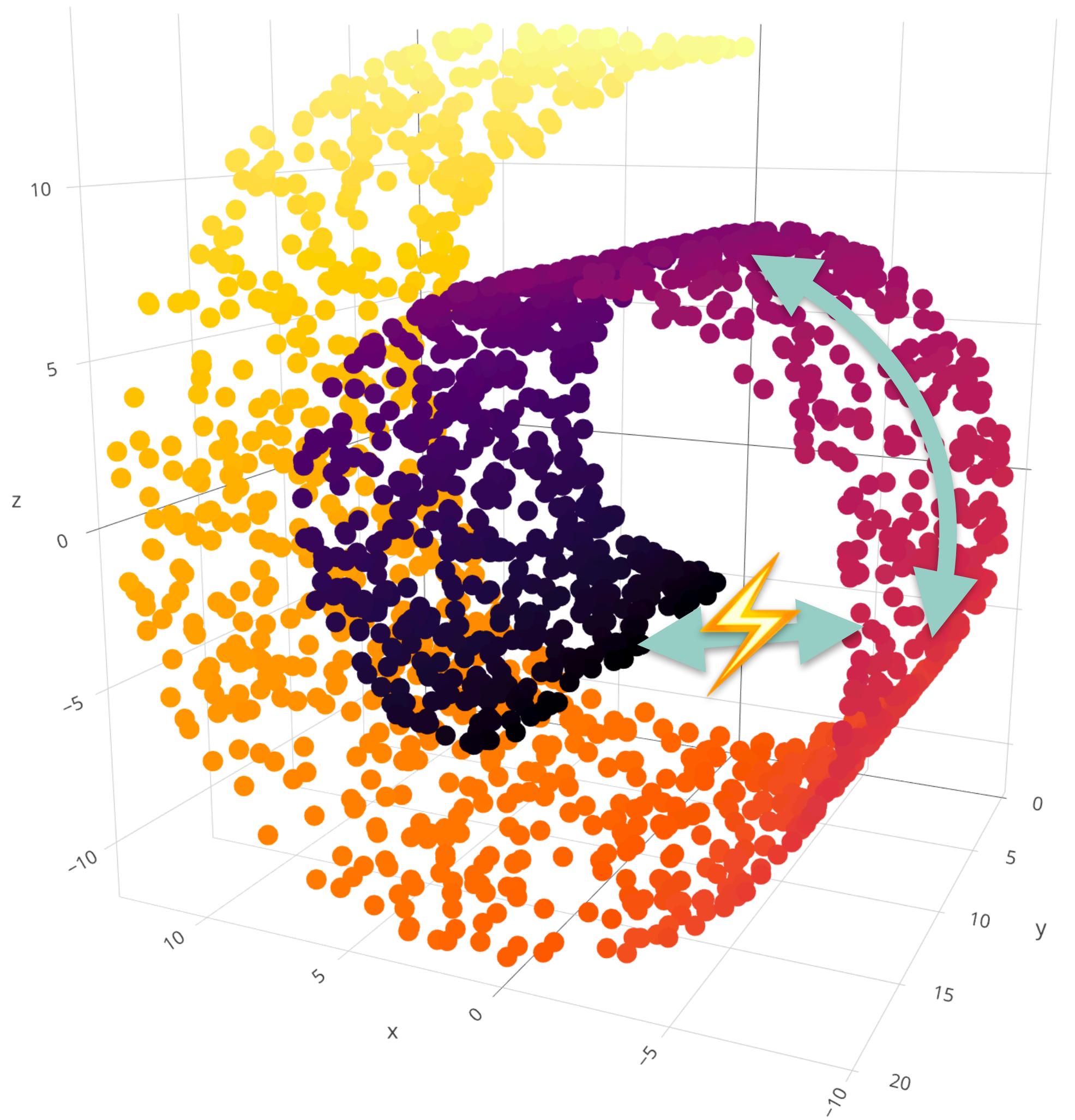
Manifold Learning



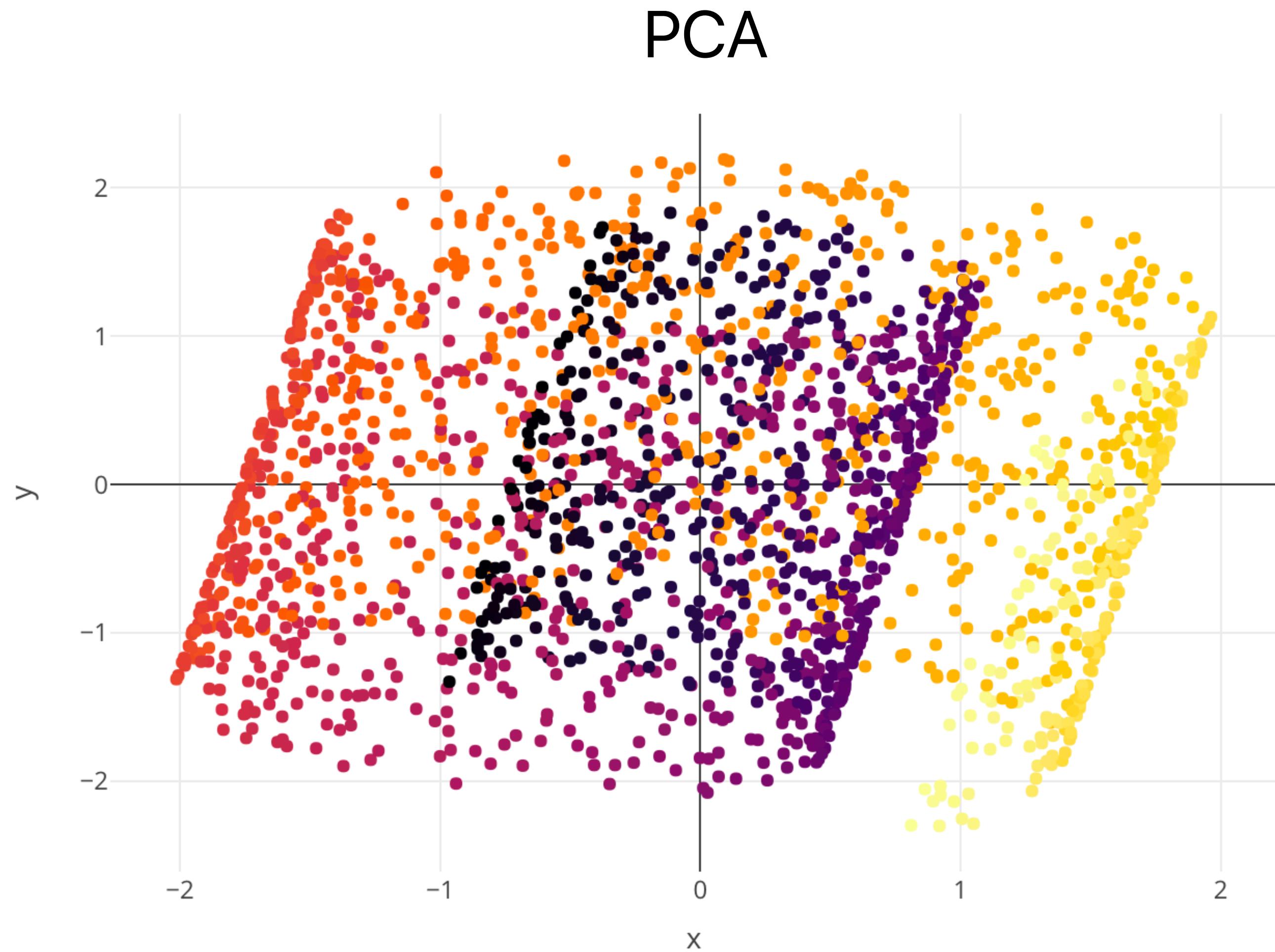
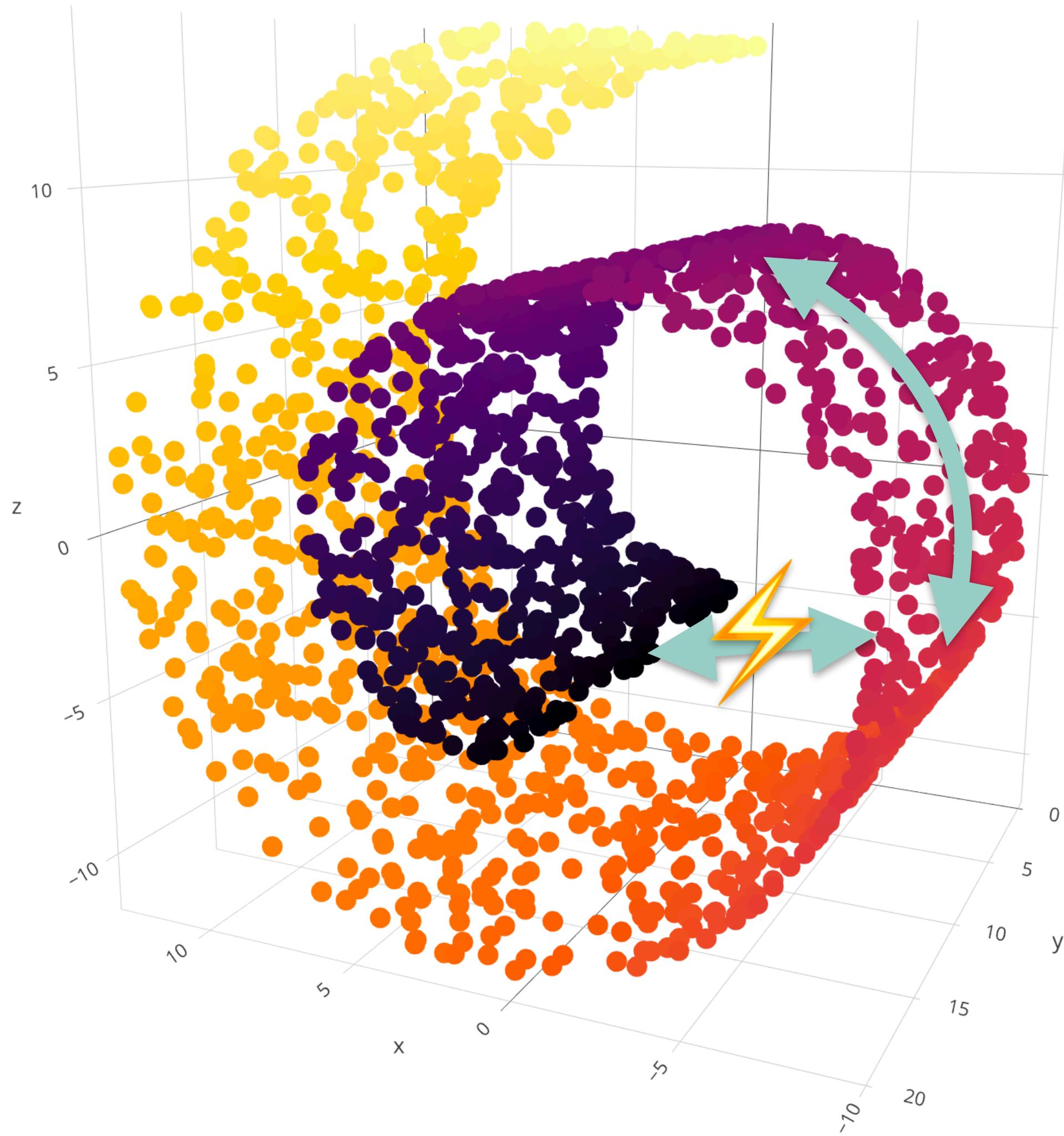
Manifold Learning



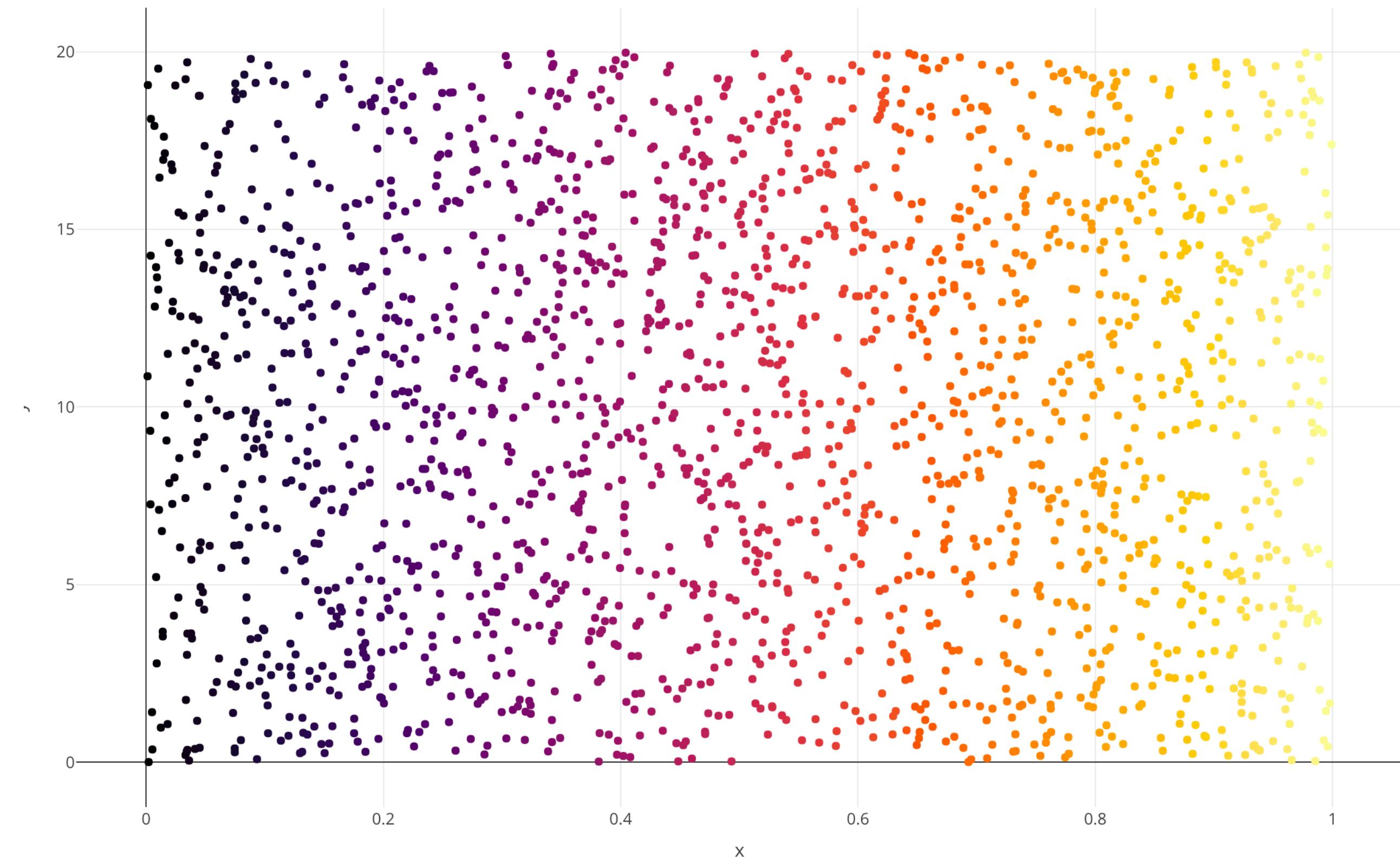
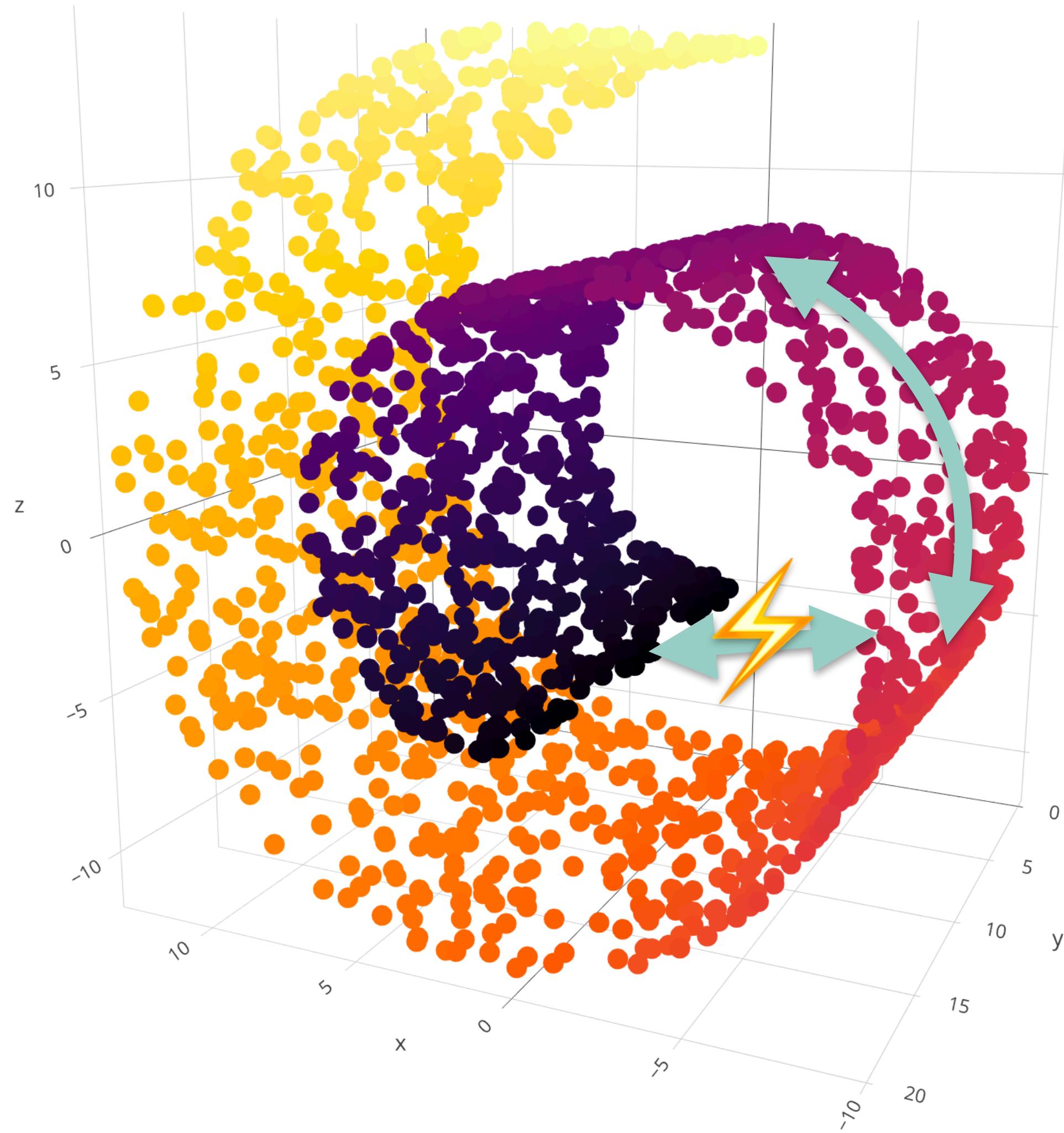
Manifold Learning



Manifold Learning



Manifold Learning

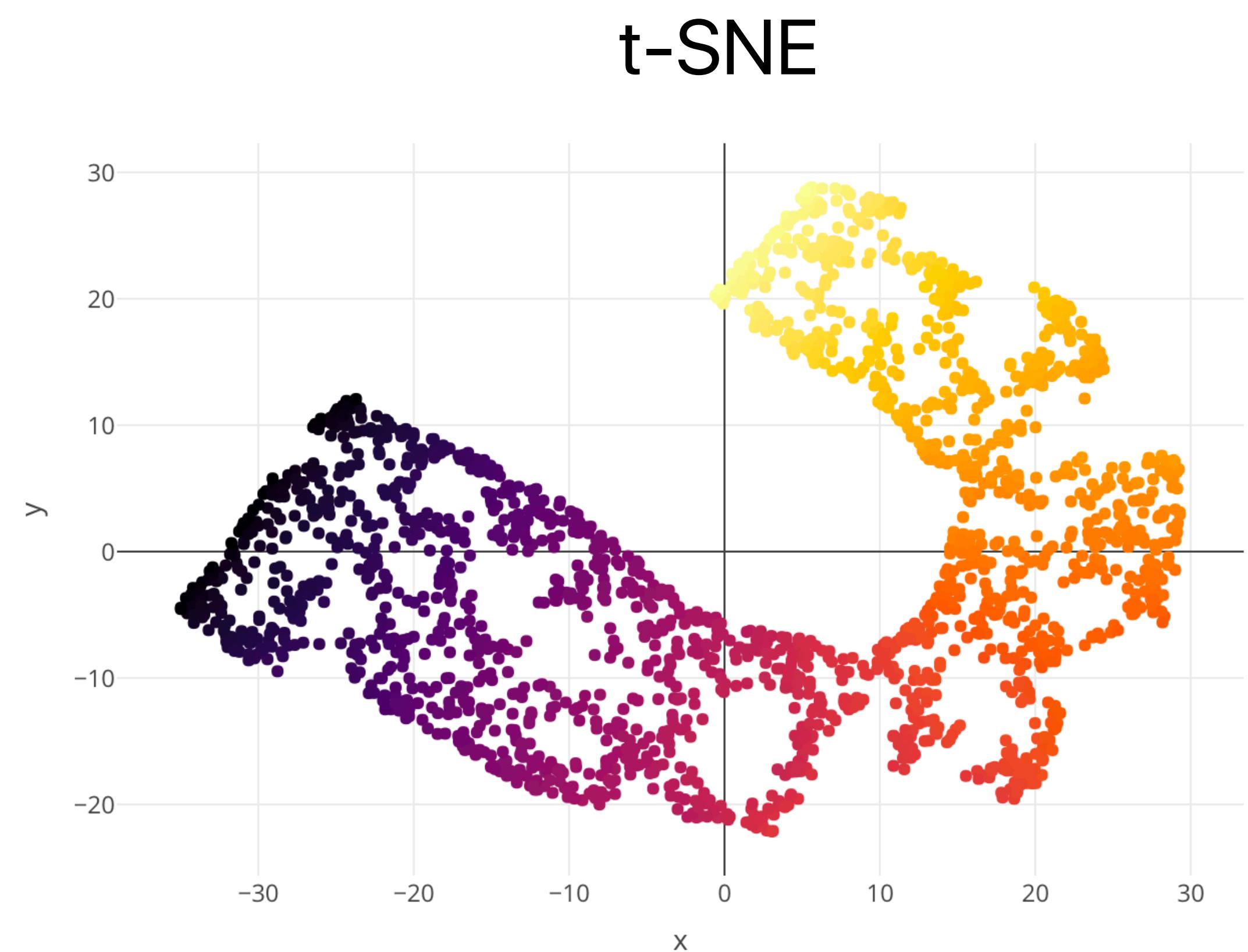


t-SNE Intuition

- Non-linear dimensionality reduction
 - Compute neighborhoods in hi-D
 - Model low-D to preserve neighborhoods
 - Preserves local neighborhoods
- ➡ Preserves high-D clusters!

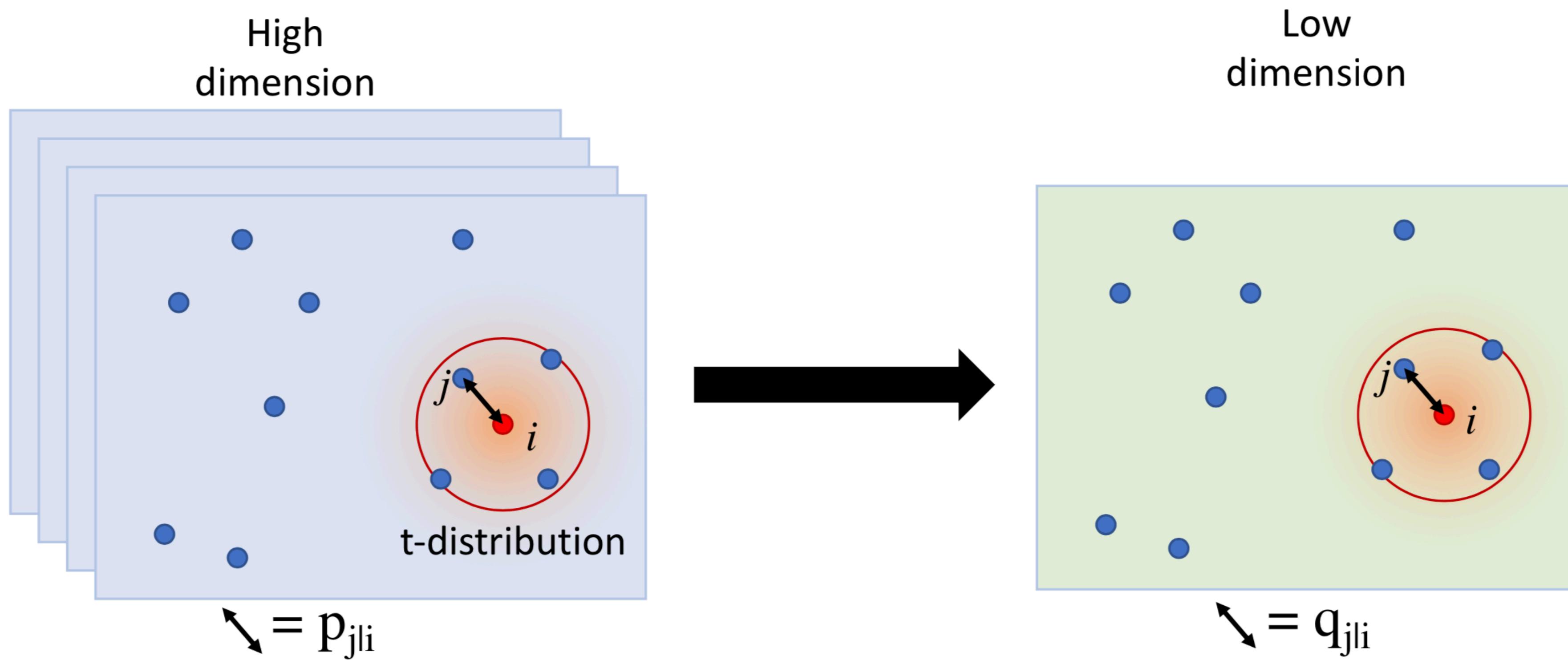
t-SNE Intuition

- Non-linear dimensionality reduction
 - Compute neighborhoods in hi-D
 - Model low-D to preserve neighborhoods
- Preserves local neighborhoods
- ➡ Preserves high-D clusters!

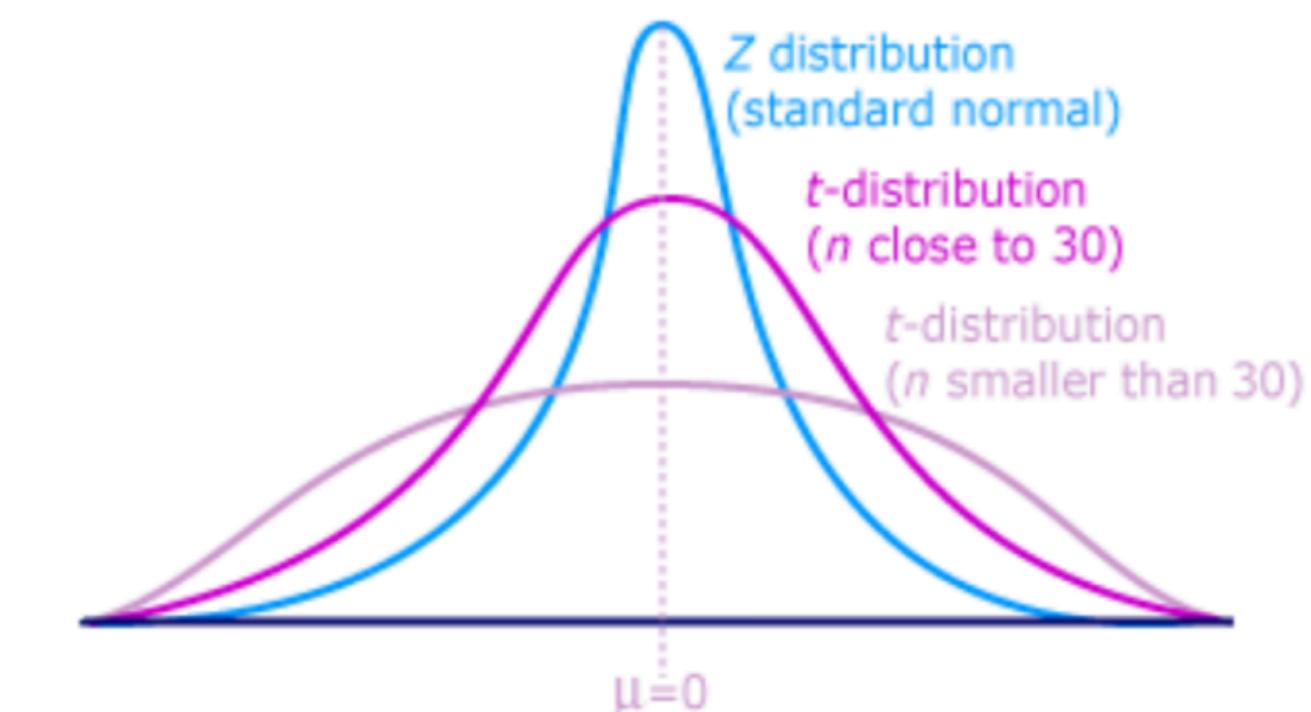


*actually also not great with Swiss Roll

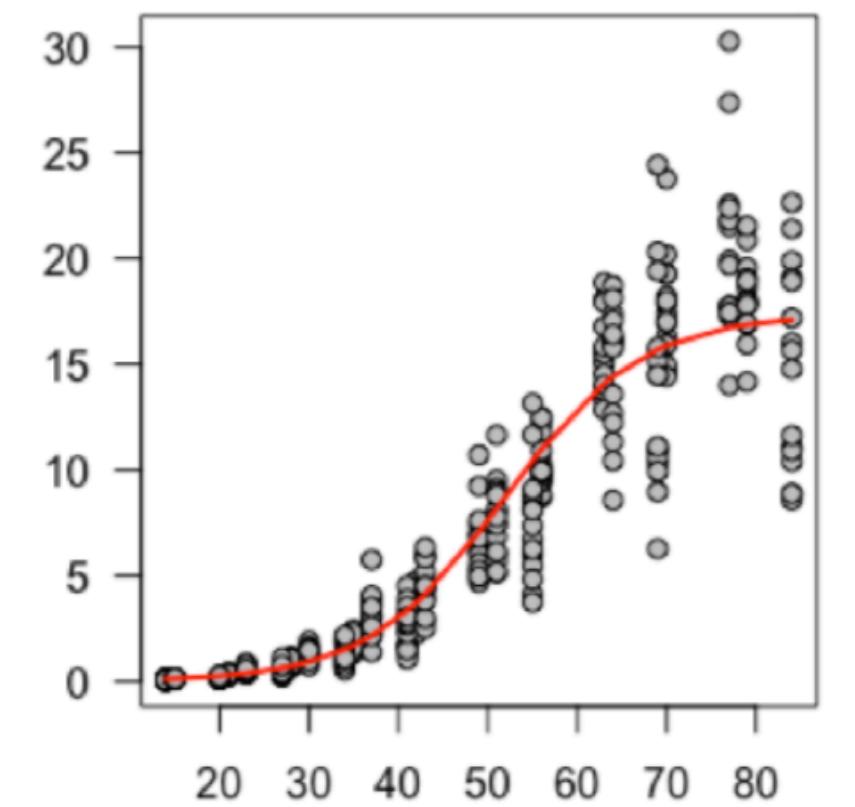
t-SNE in Brief



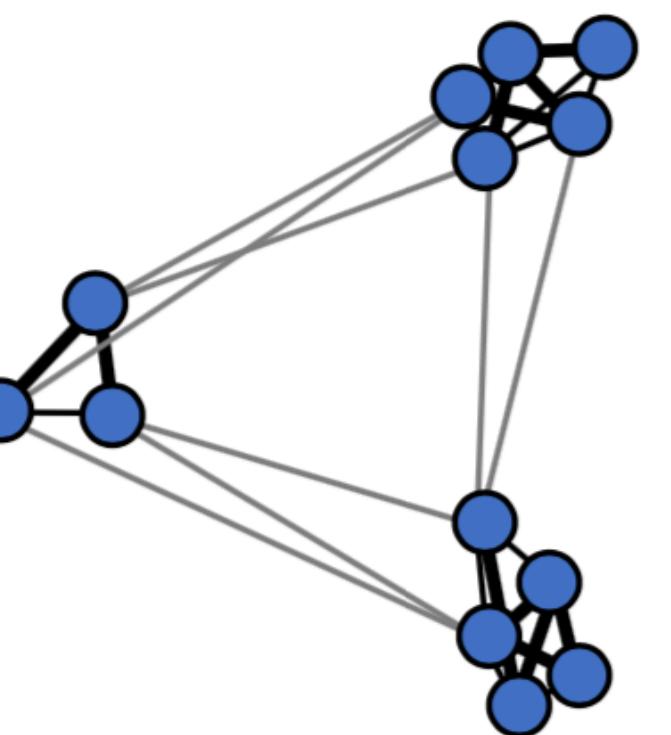
$p_{j|i}$ and $q_{j|i}$ measure the conditional probability that a point i would pick point j as its nearest neighbor, in high (p) and low (q) dimensional space respectively.



Sidestep: Graphs

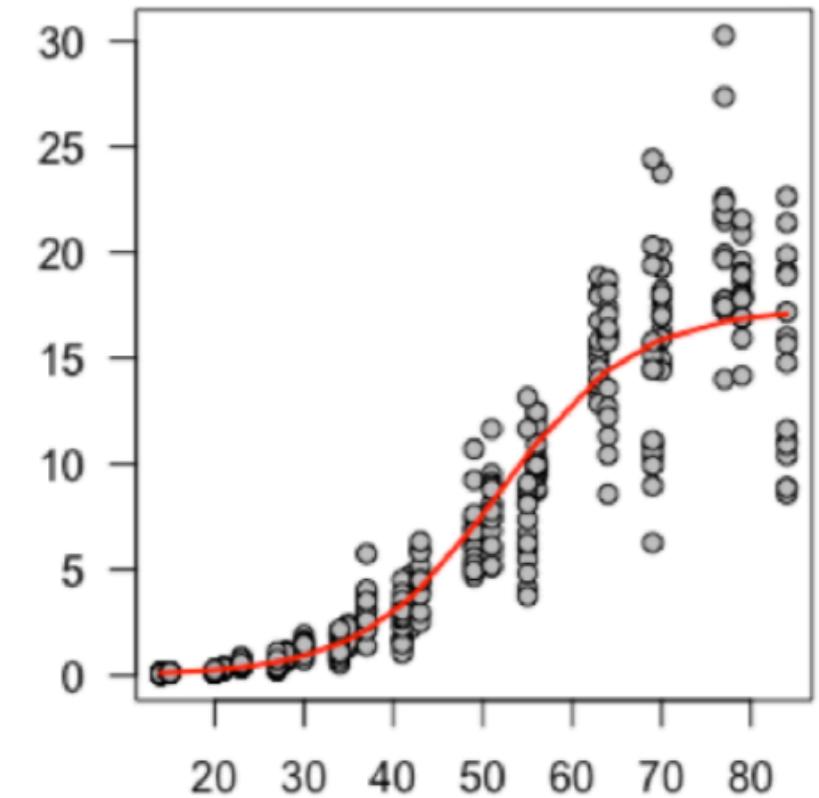


This is a PLOT

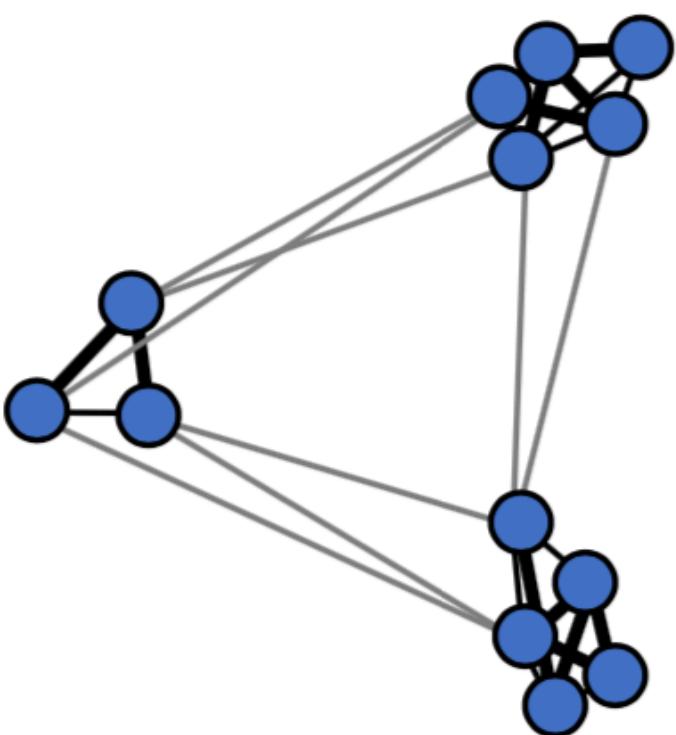


This is GRAPH
(a.k.a. network)

Sidestep: Graphs



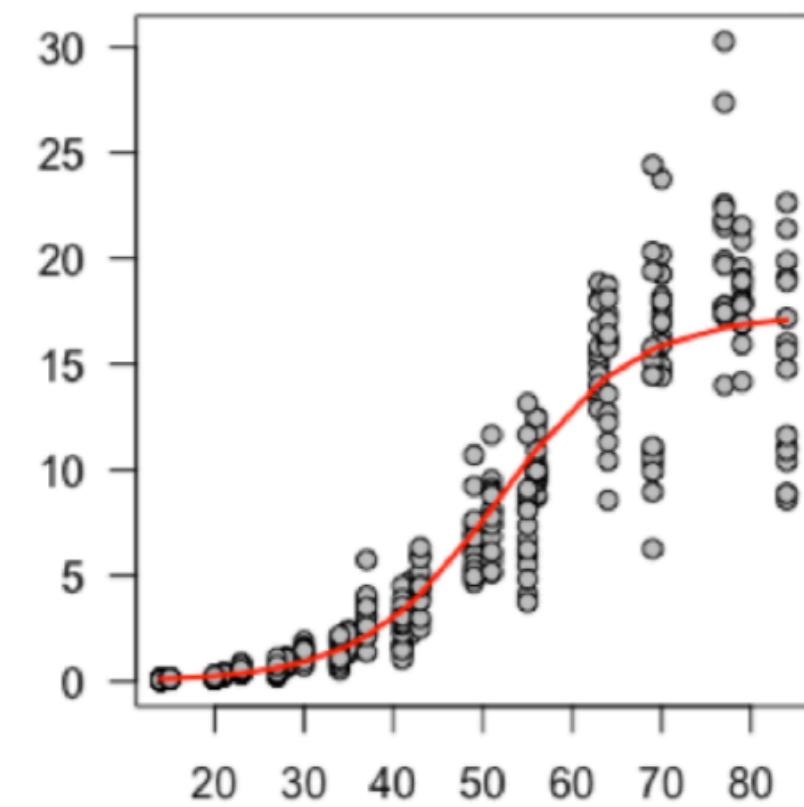
This is a PLOT



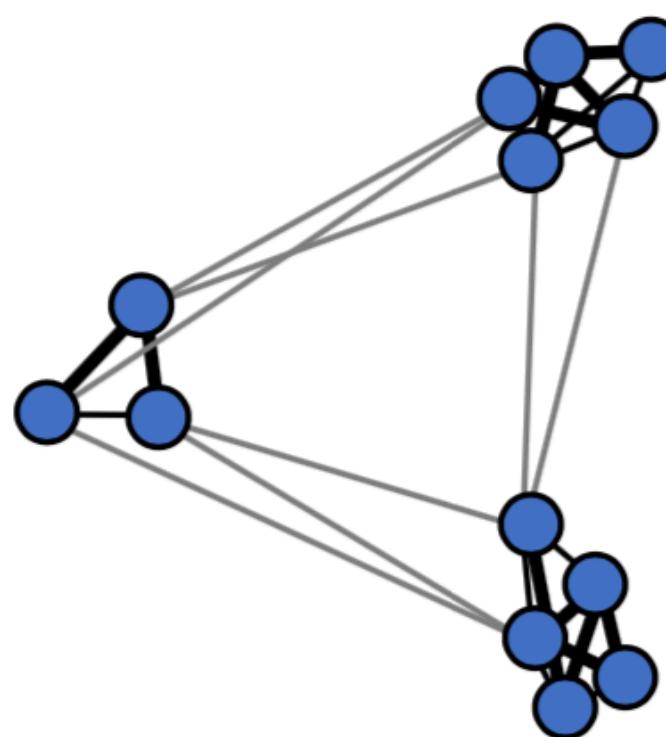
This is GRAPH
(a.k.a. network)

- Each dot is a cell (or a gene)
- Each line represents a connection between 2 cells
- Each connection can be weighted as a proximity between cells
 - Correlation (high and positive)
 - Euclidean distance (low)
 - etc.

Sidestep: Graphs



This is a PLOT



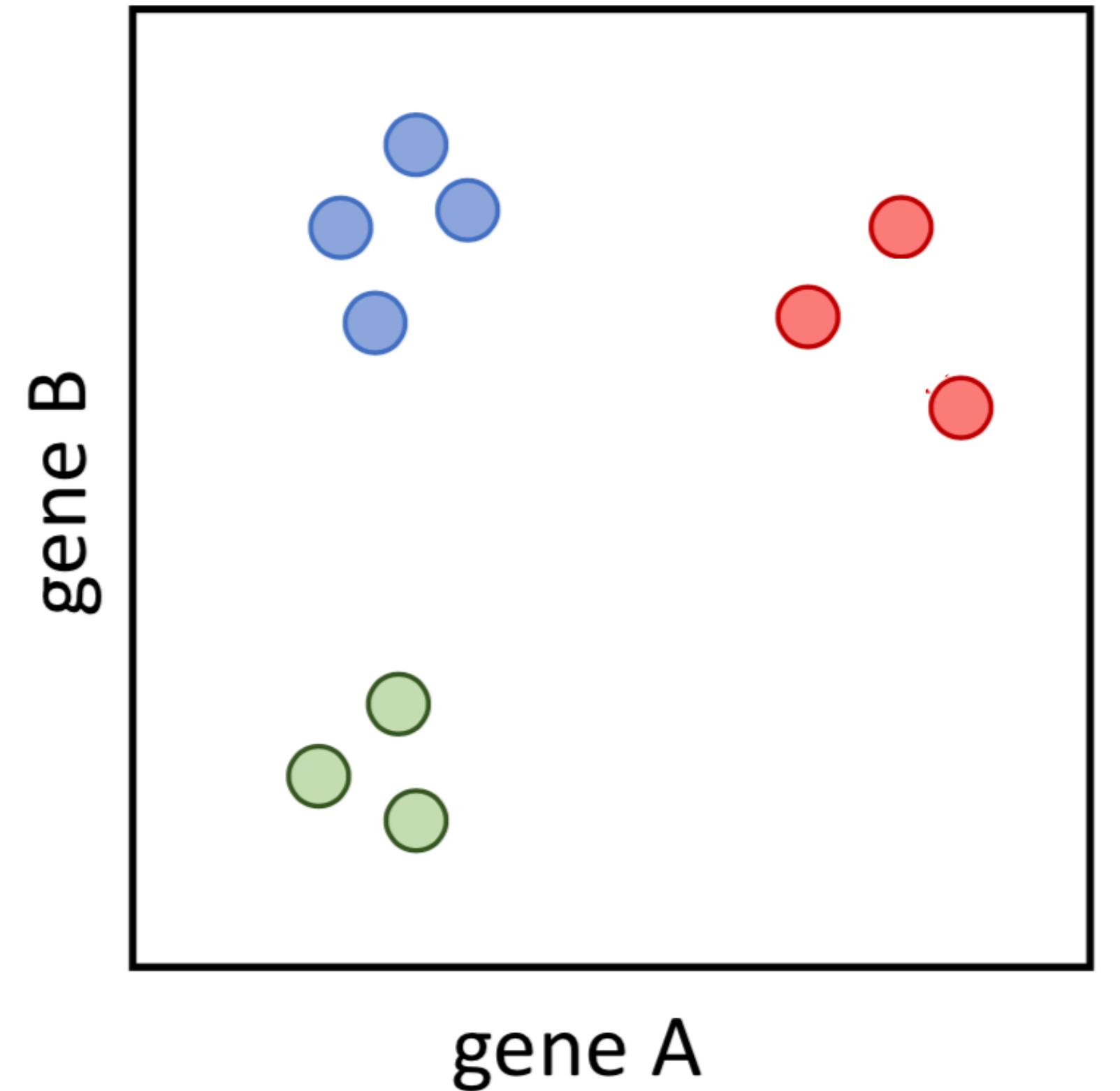
This is GRAPH
(a.k.a. network)

- Each dot is a cell (or a gene)
- Each line represents a connection between 2 cells
- Each connection can be weighted as a proximity between cells
 - Correlation (high and positive)
 - Euclidean distance (low)
 - etc.

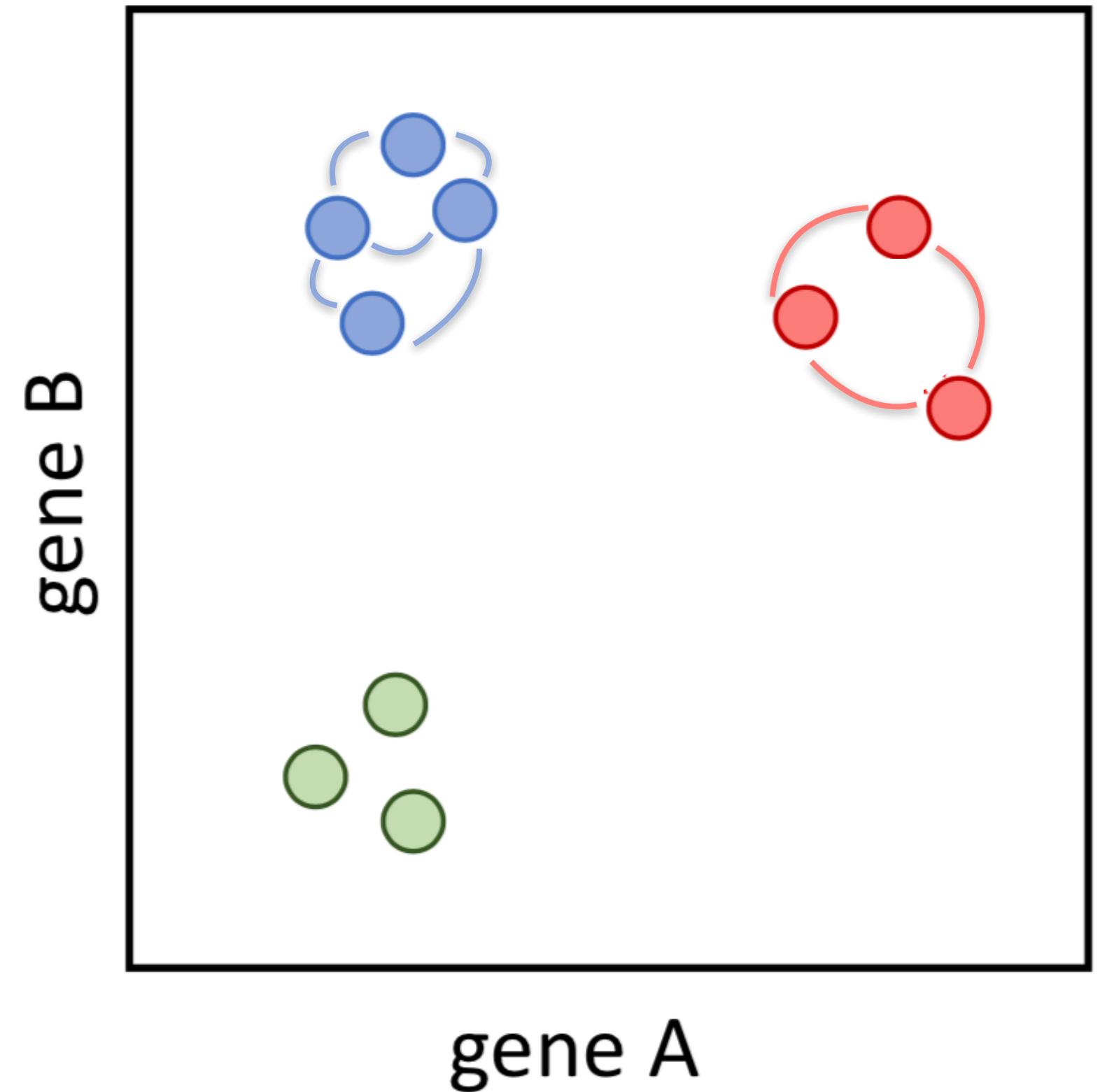
Graph-based dimensionality reduction algorithms can be divided into 2 main steps:

1. Construct a weighted graph based on the top k connections
(a.k.a. k -nearest neighbors, KNN)
2. The low dimensional layout of the graph is computed and optimized

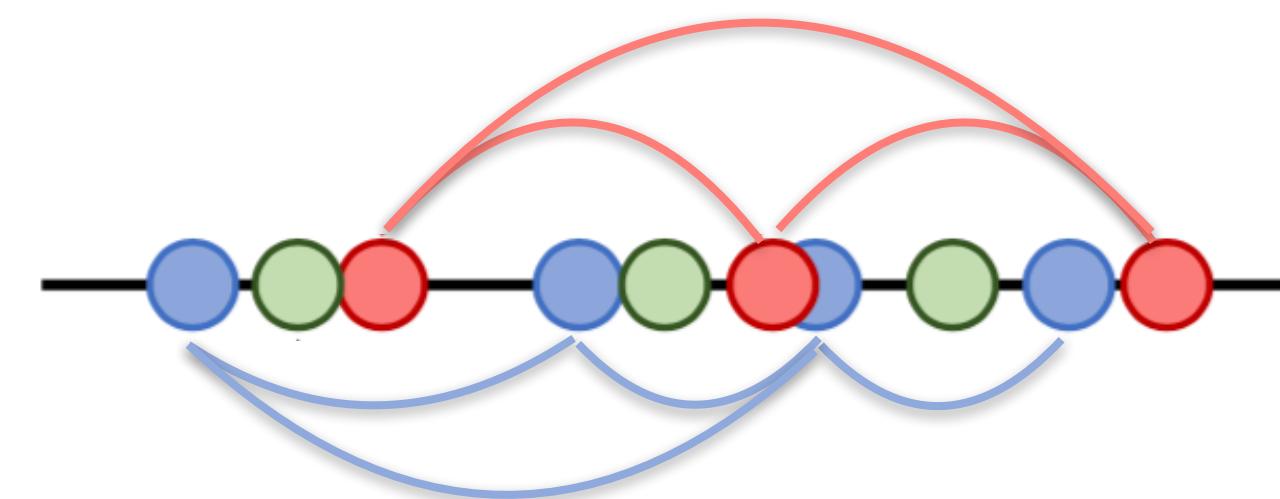
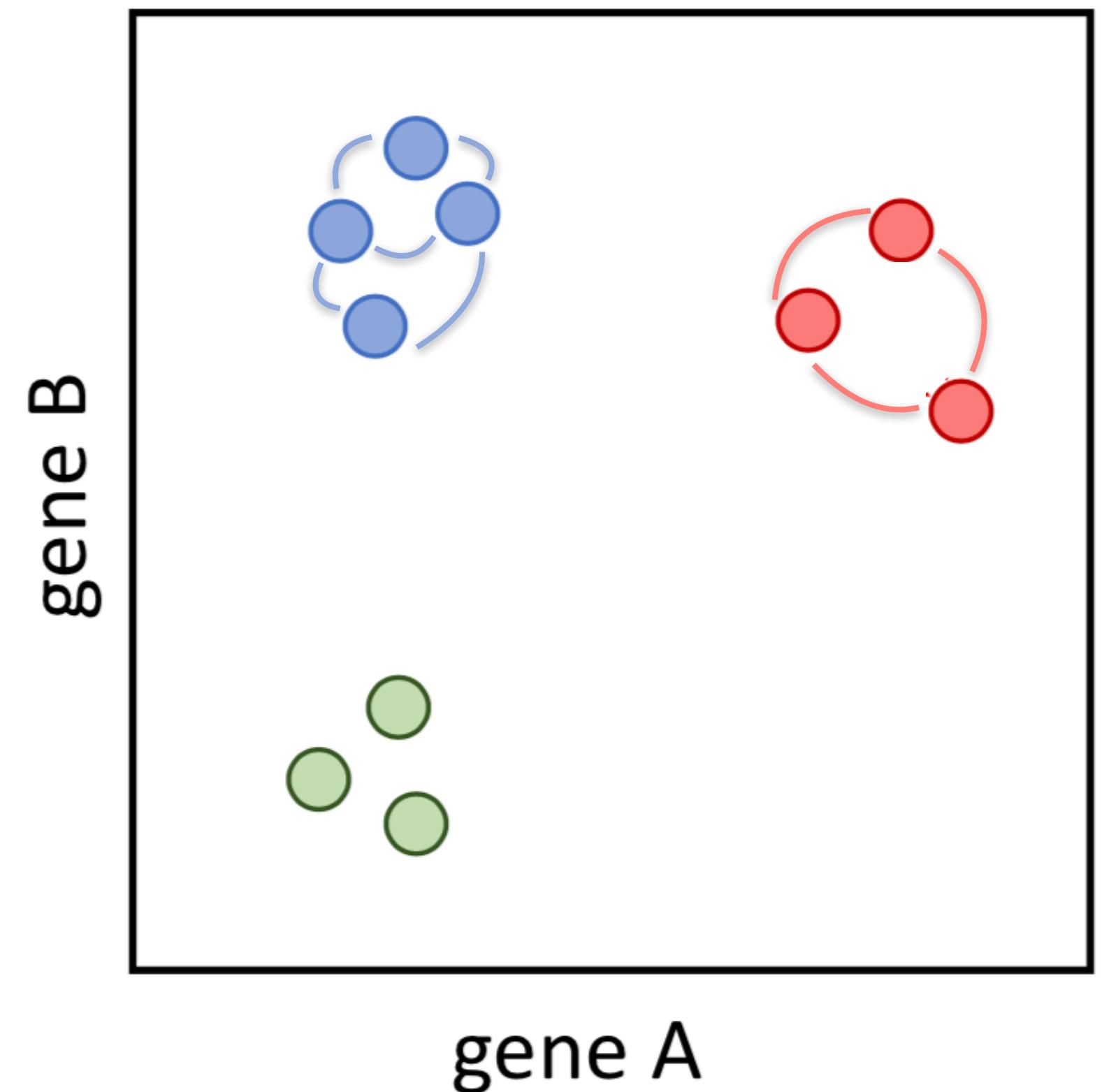
t-SNE in Brief



t-SNE in Brief

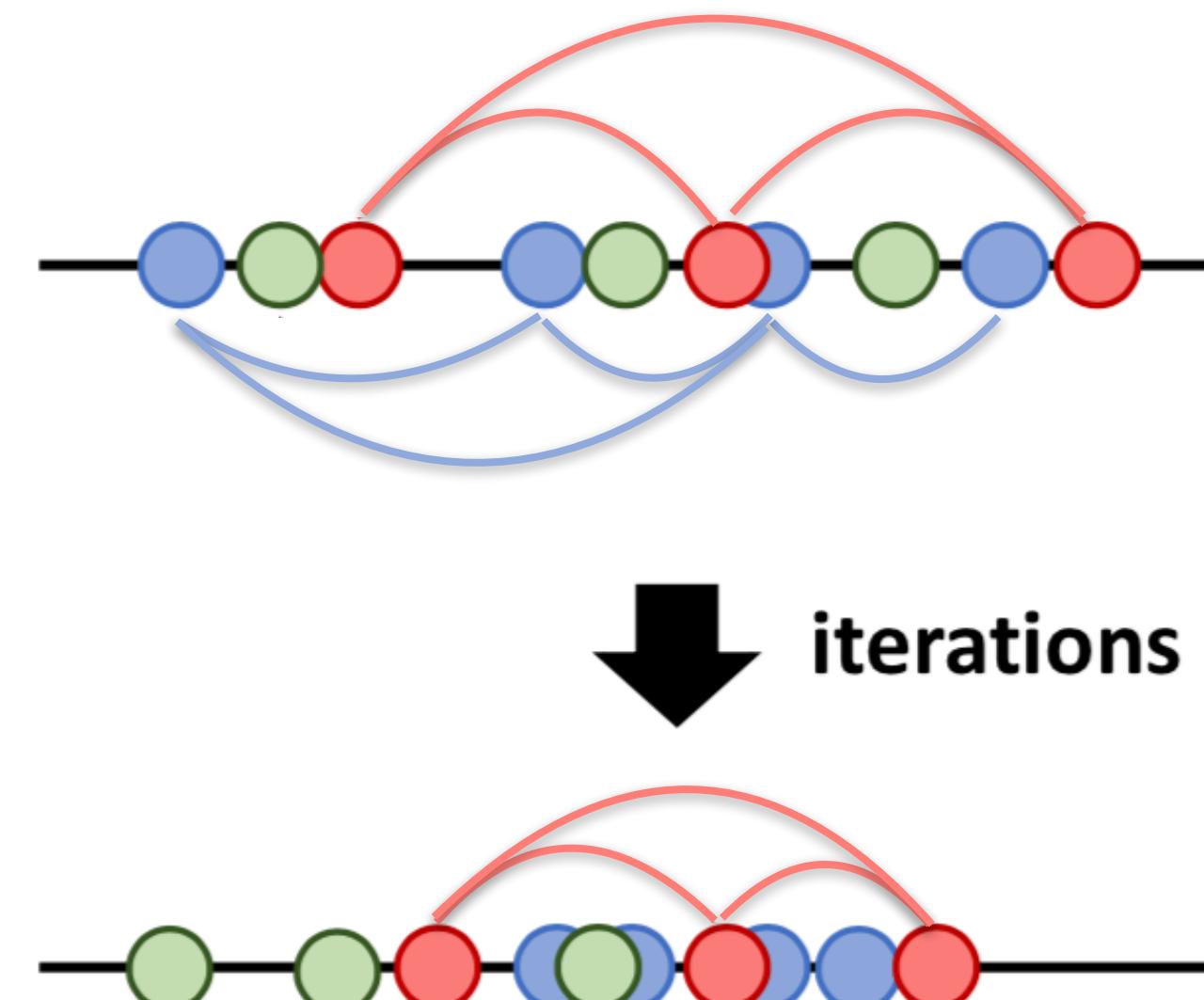
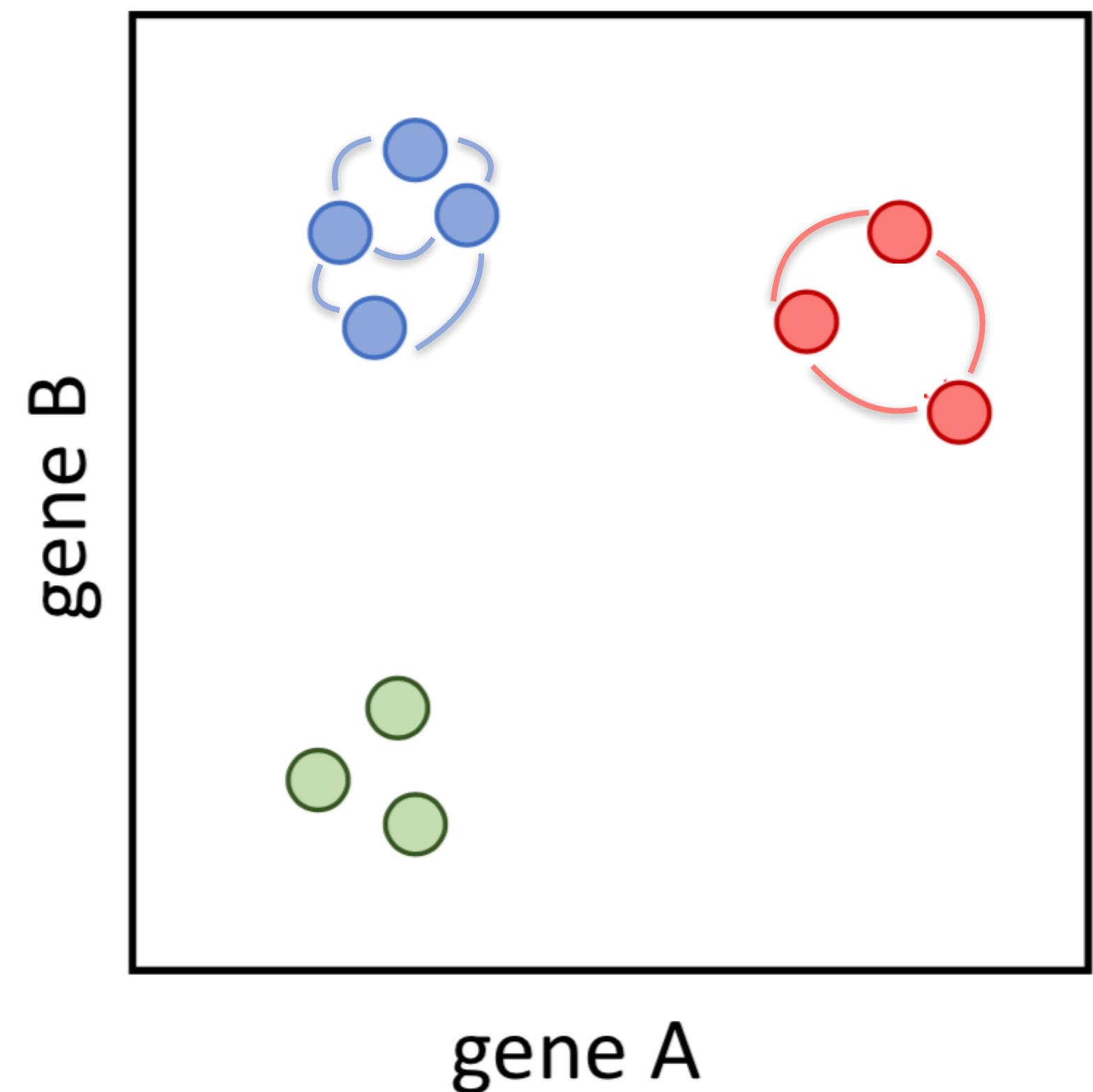


t-SNE in Brief



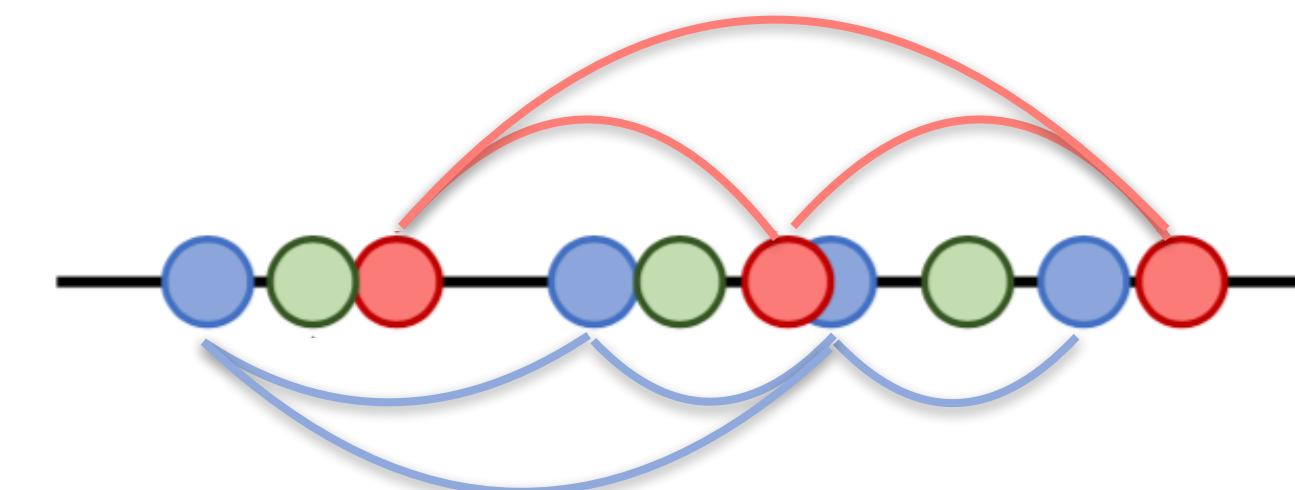
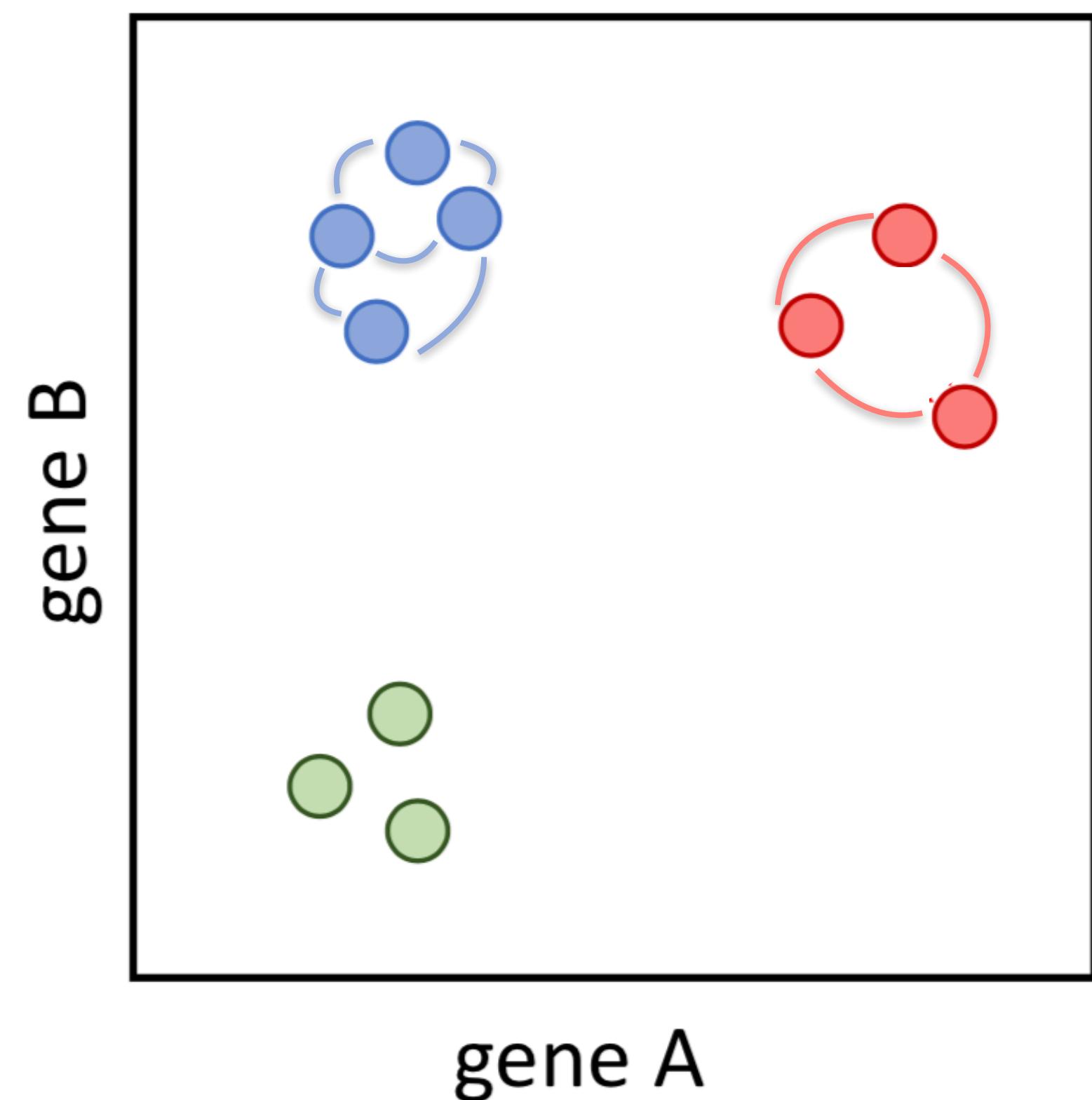
Higher KL divergence
(cost / error)

t-SNE in Brief



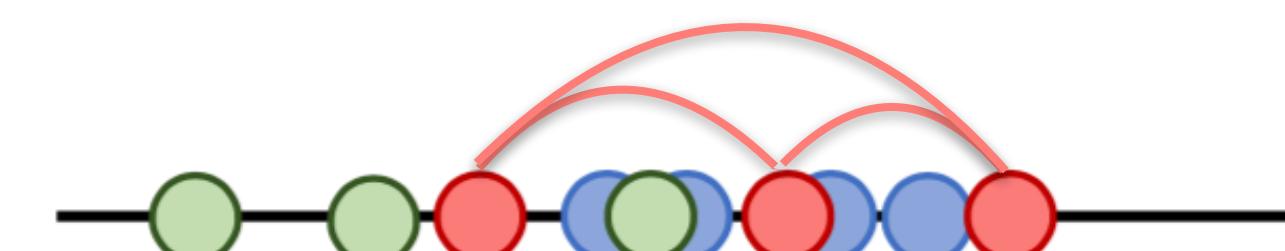
Higher KL divergence
(cost / error)

t-SNE in Brief

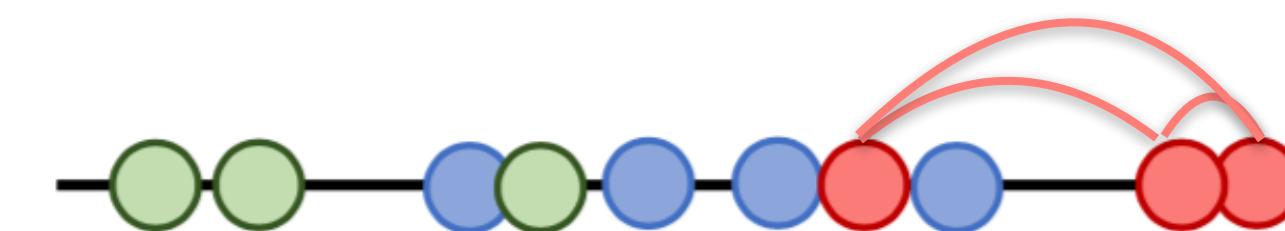


Higher KL divergence
(cost / error)

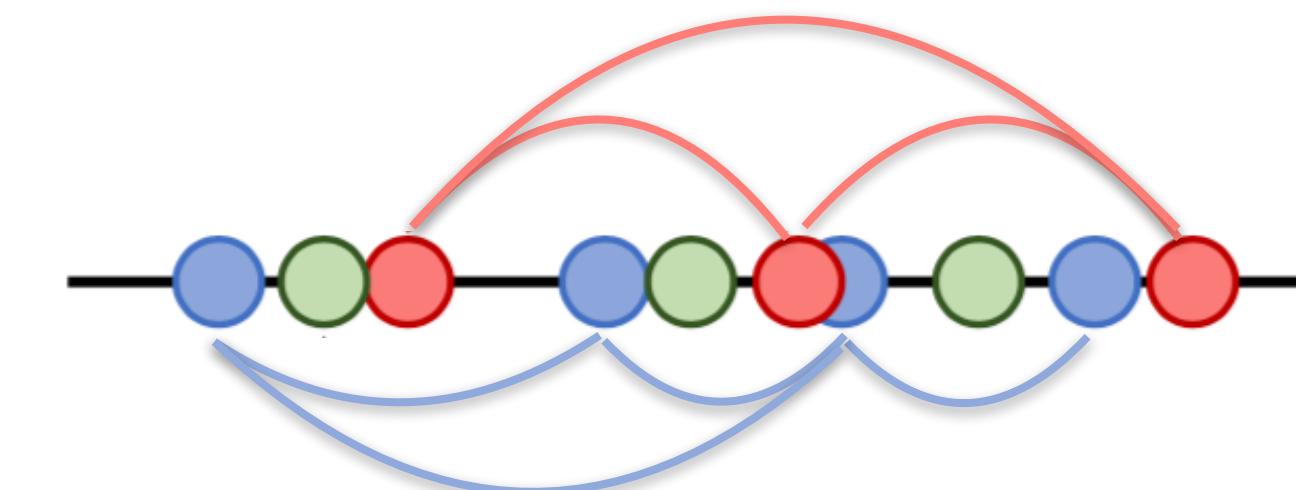
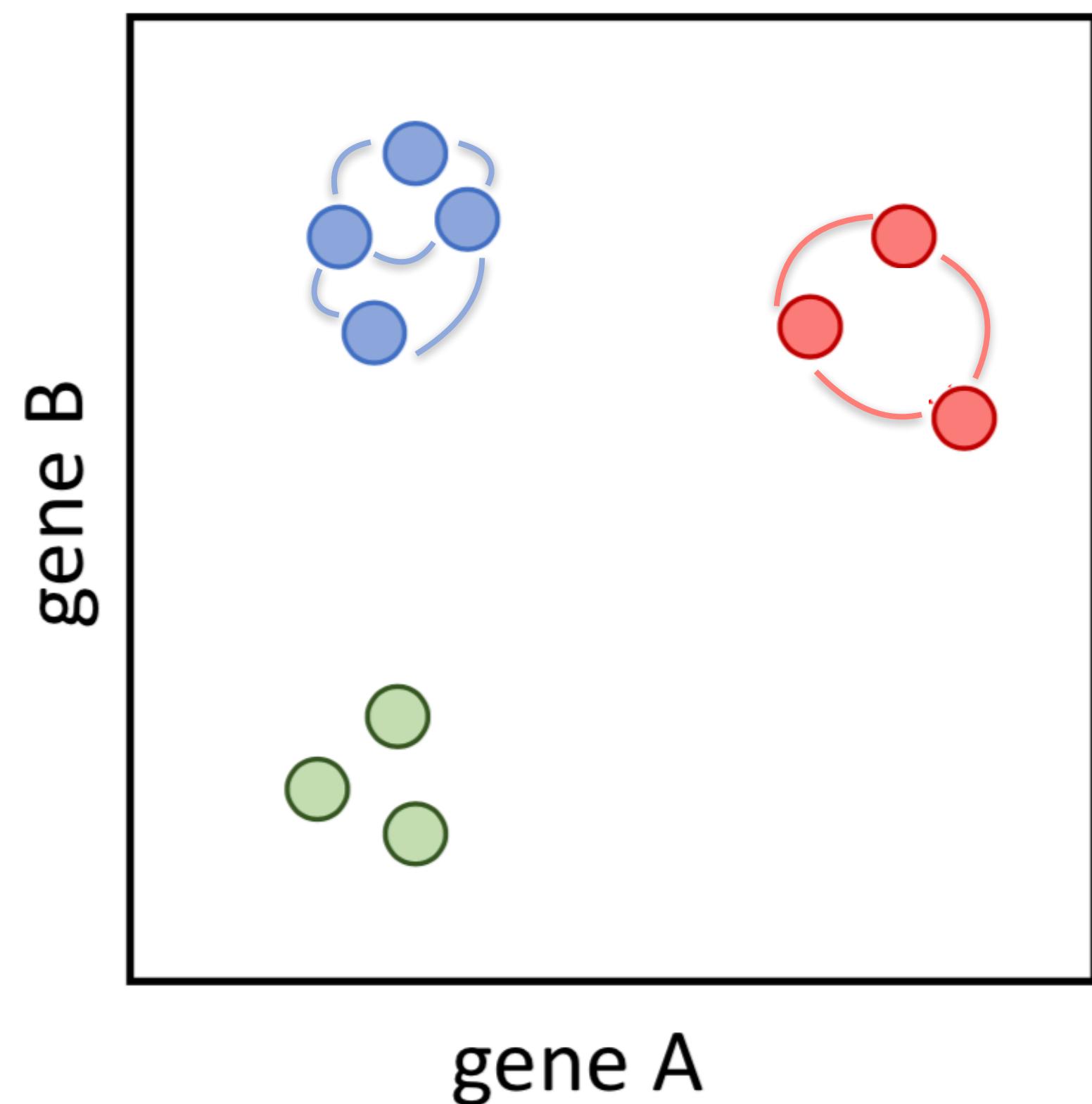
iterations



iterations

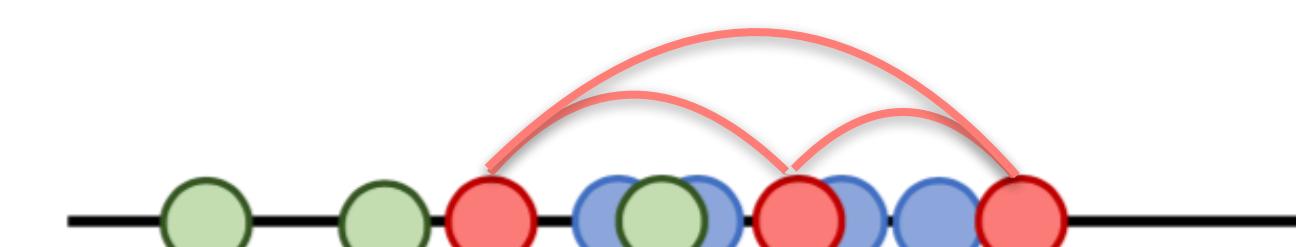


t-SNE in Brief

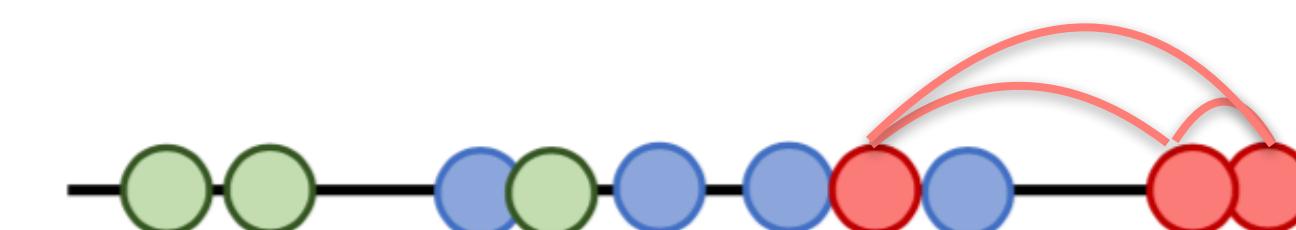


Higher KL divergence
(cost / error)

iterations



iterations



iterations

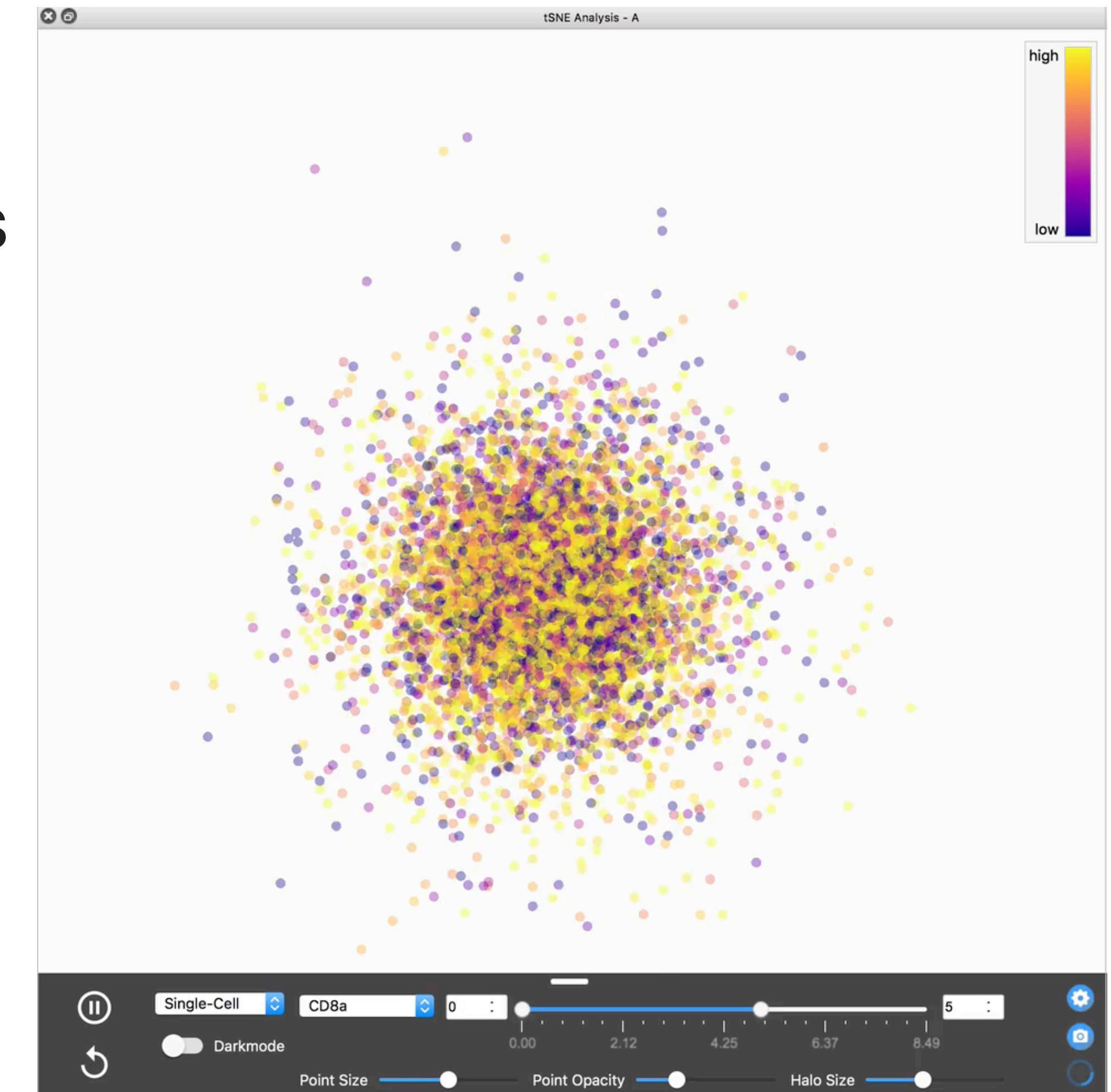


Lower KL divergence
(cost / error)

t-SNE in Brief

- 2 major computation parts in tSNE
 - compute high dimensional neighborhoods
 - optimize low dimensional neighborhoods
- Computationally intensive
 - Barnes-Hut implementation improves this
- Several parameters
 - Some can severely impact results

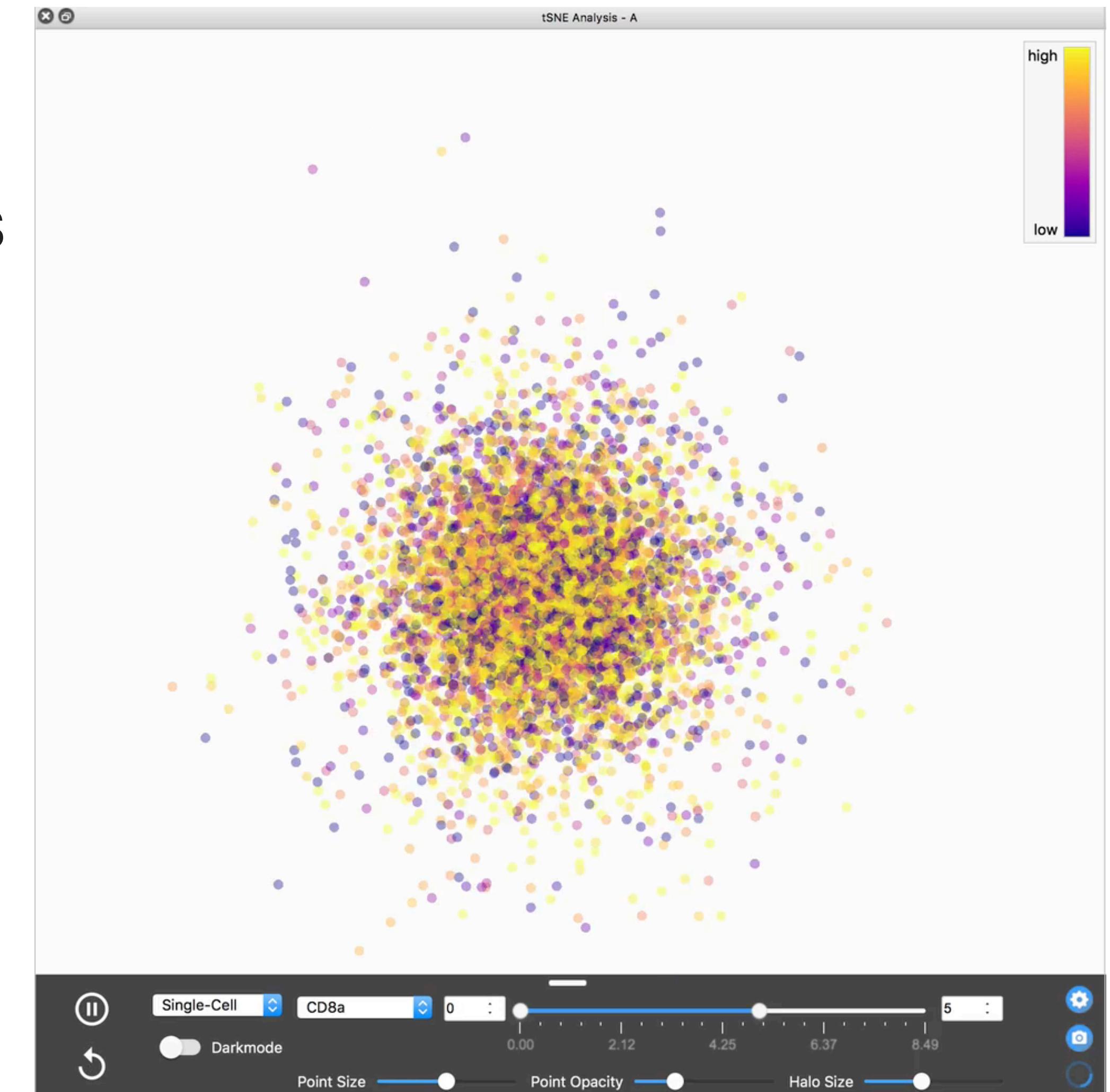
<https://distill.pub/2016/misread-tsne/>



t-SNE in Brief

- 2 major computation parts in tSNE
 - compute high dimensional neighborhoods
 - optimize low dimensional neighborhoods
- Computationally intensive
 - Barnes-Hut implementation improves this
- Several parameters
 - Some can severely impact results

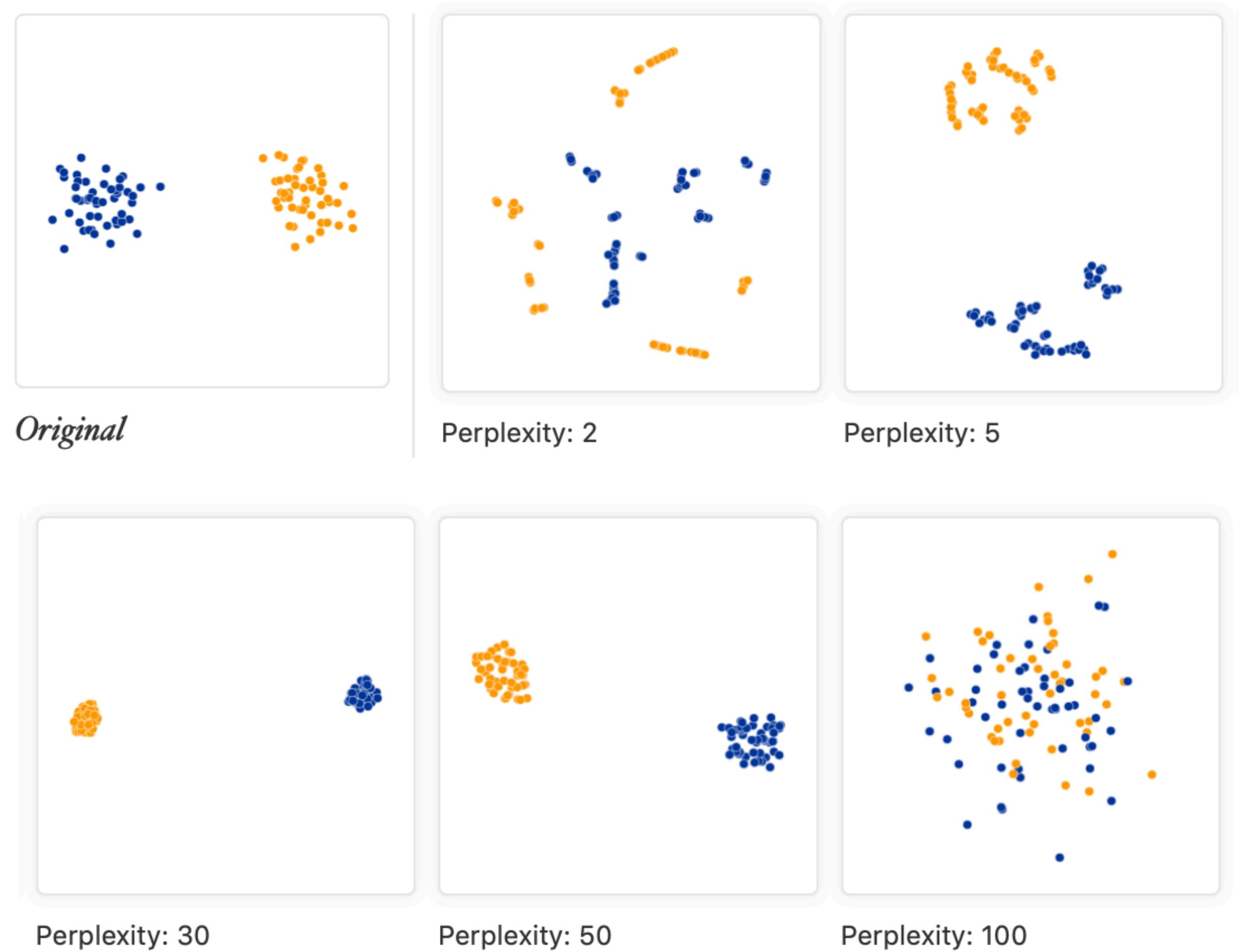
<https://distill.pub/2016/misread-tsne/>



t-SNE Parameters

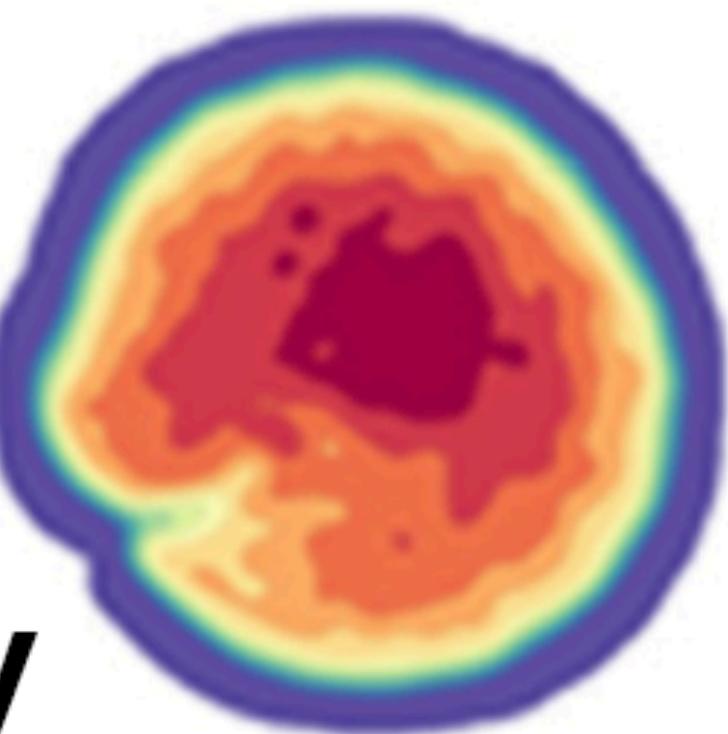
- **Perplexity**
- Number of iterations
- Learning rate
- Theta (for BH t-SNE)

...



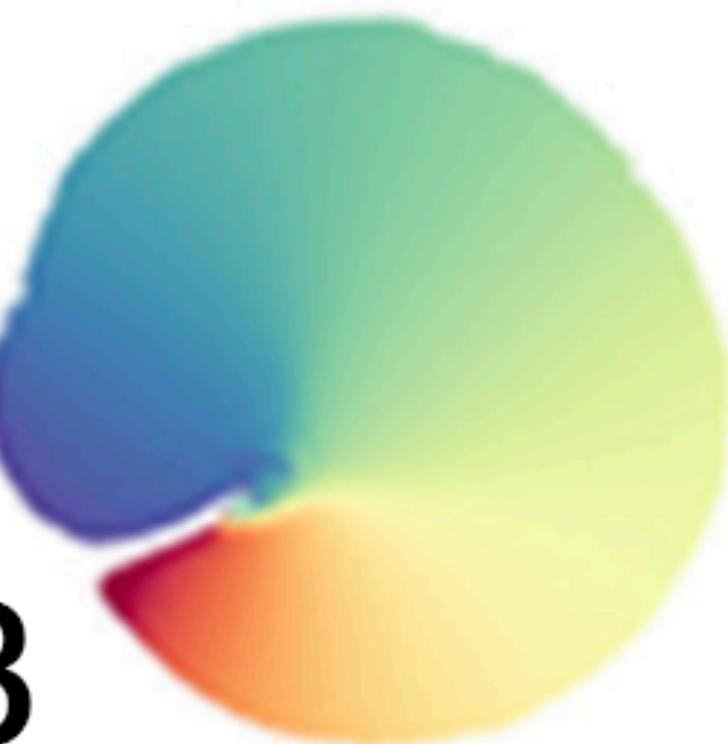
t-SNE Parameters

- Perplexity
- Number of iterations
- Learning rate
- Theta (for BH t-SNE)
- ...

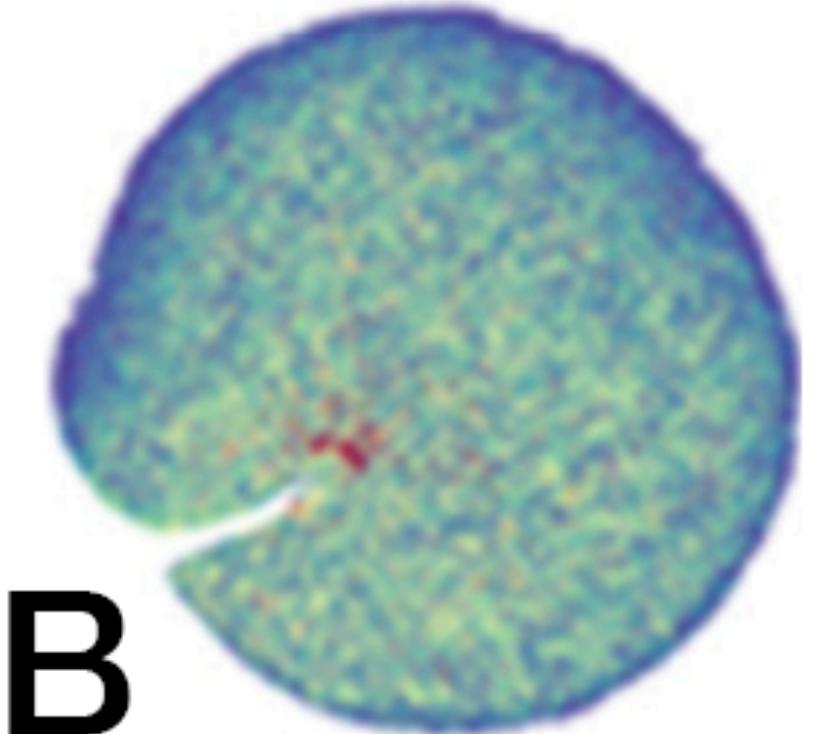


density

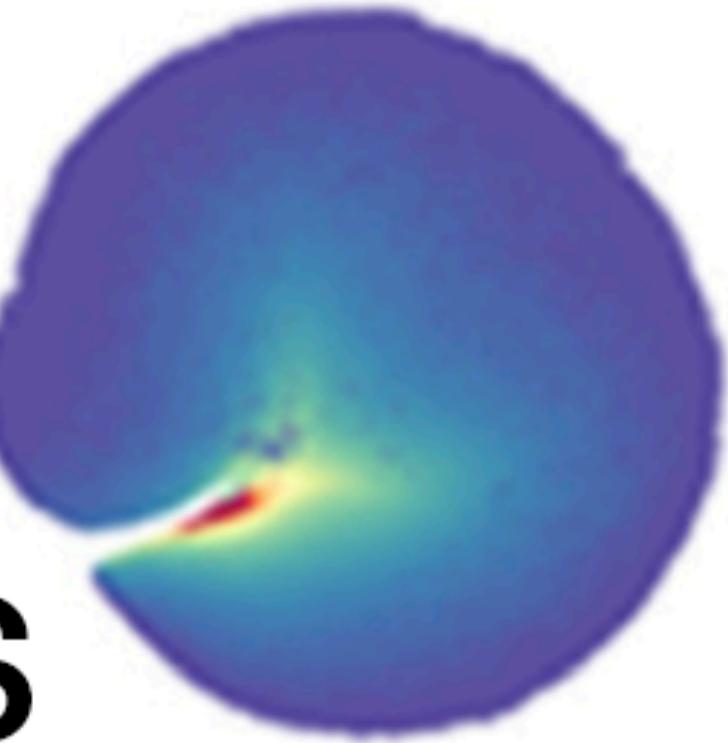
4-11BB



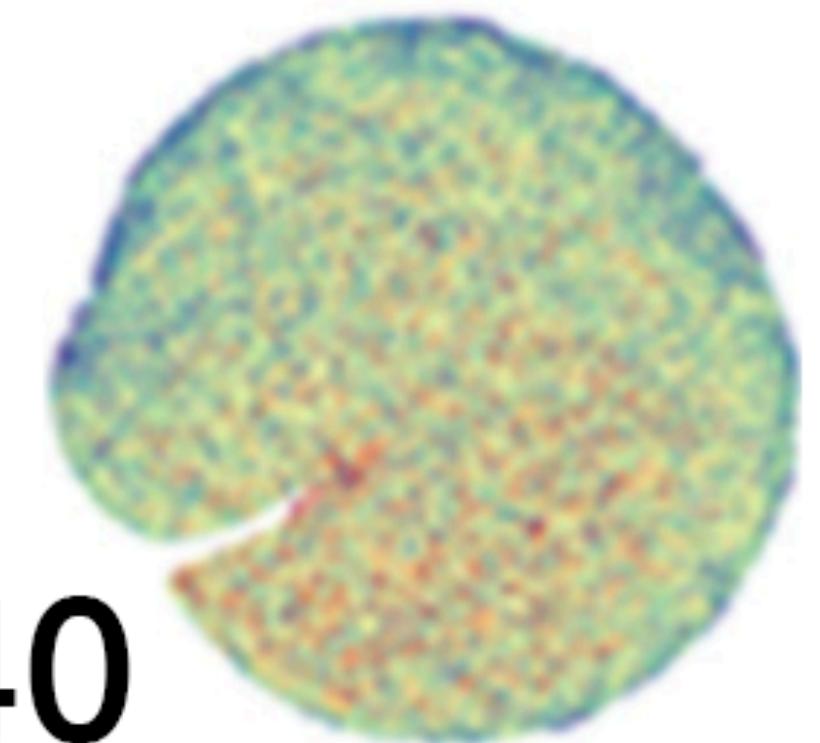
CD28



CD40L



ICOS



OX40

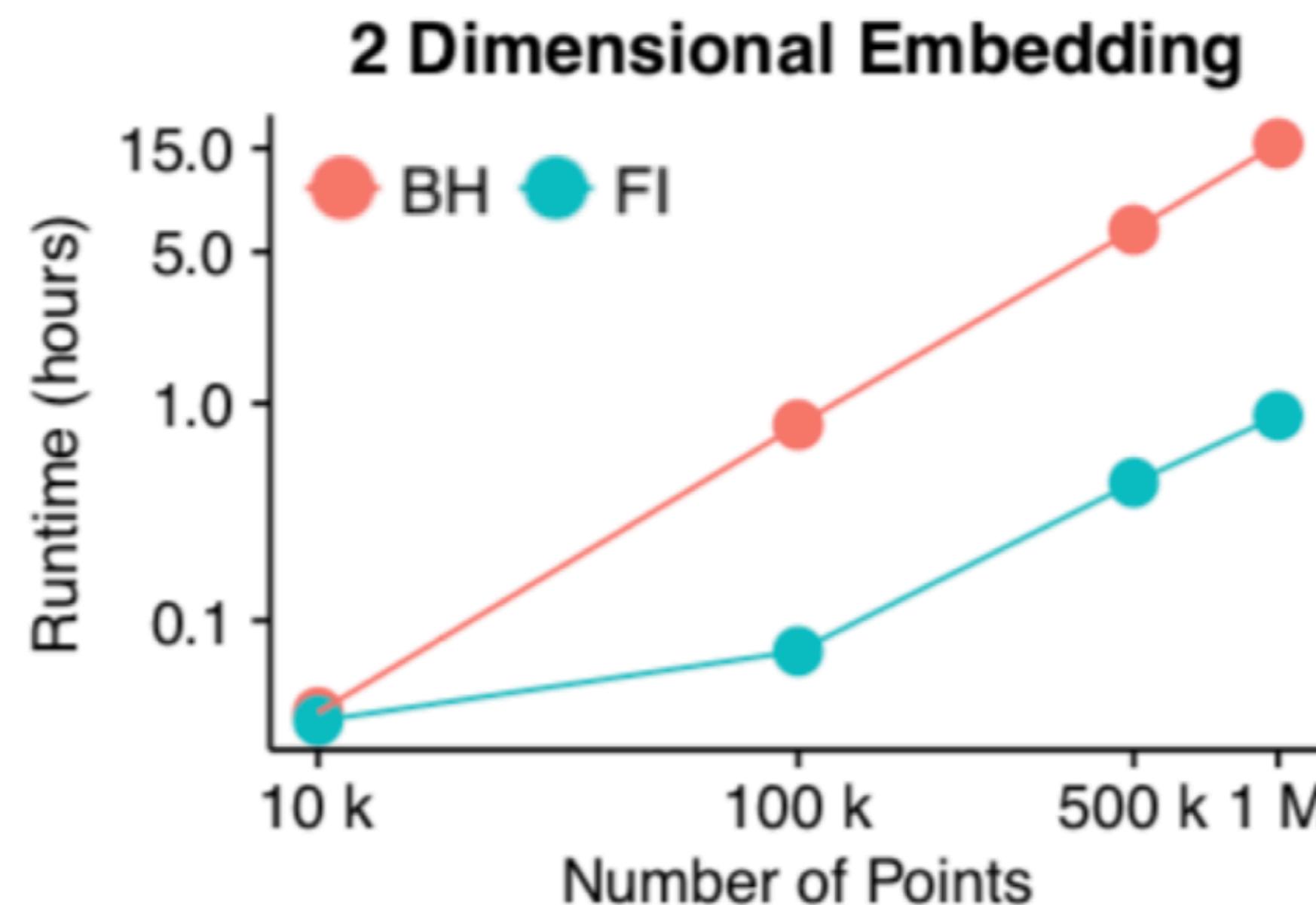
Important Notes

- Typically, the optimization is initialized randomly
Multiple runs will produce different results
- The cost function never reaches the minimum
- t-SNE optimizes the distance between close points (local embedding)
Distances within a group are slightly meaningful, but not between groups!
- To add more samples, you need to re-run the algorithm from start.

t-SNE Implementations

- Many Implementations available
- Fast Fourier Transform-accelerated

<https://www.nature.com/articles/s41592-018-0308-4>



Implementations

Below, implementations of t-SNE in various languages are available for download. Some of these implementations were developed by me, and some by other contributors. For the standard t-SNE method, implementations in Matlab, C++, CUDA, Python, Torch, R, Julia, and JavaScript are available. In addition, we provide a Matlab implementation of parametric t-SNE (described [here](#)). Finally, we provide a Barnes-Hut implementation of t-SNE (described [here](#)), which is the fastest t-SNE implementation to date, and which scales much better to big data sets.

You are free to use, modify, or redistribute this software in any way you want, but only for non-commercial purposes. The use of the software is at your own risk; the authors are not responsible for any damage as a result from errors in the software.

NOTE: t-SNE is now built-in functionality [in Matlab](#) and [in SPSS](#)!

Matlab implementation (user guide)	All platforms
CUDA implementation (by David, Roshan, and Forrest ; see paper)	All platforms
Python implementation	All platforms
Go implementation (by Daniel Salvadori)	All platforms
Torch implementation	All platforms
Julia implementation (by Leif Jonsson)	All platforms
Java implementation (by Leif Jonsson)	All platforms
R implementation (by Justin)	All platforms
JavaScript implementation (by Andrej; online demonstration)	All platforms
Parametric t-SNE (outdated; see here)	All platforms
Barnes-Hut t-SNE (C++, Matlab, Python, Torch, and R wrappers; see here)	All platforms / Github
MNIST Dataset	Matlab file

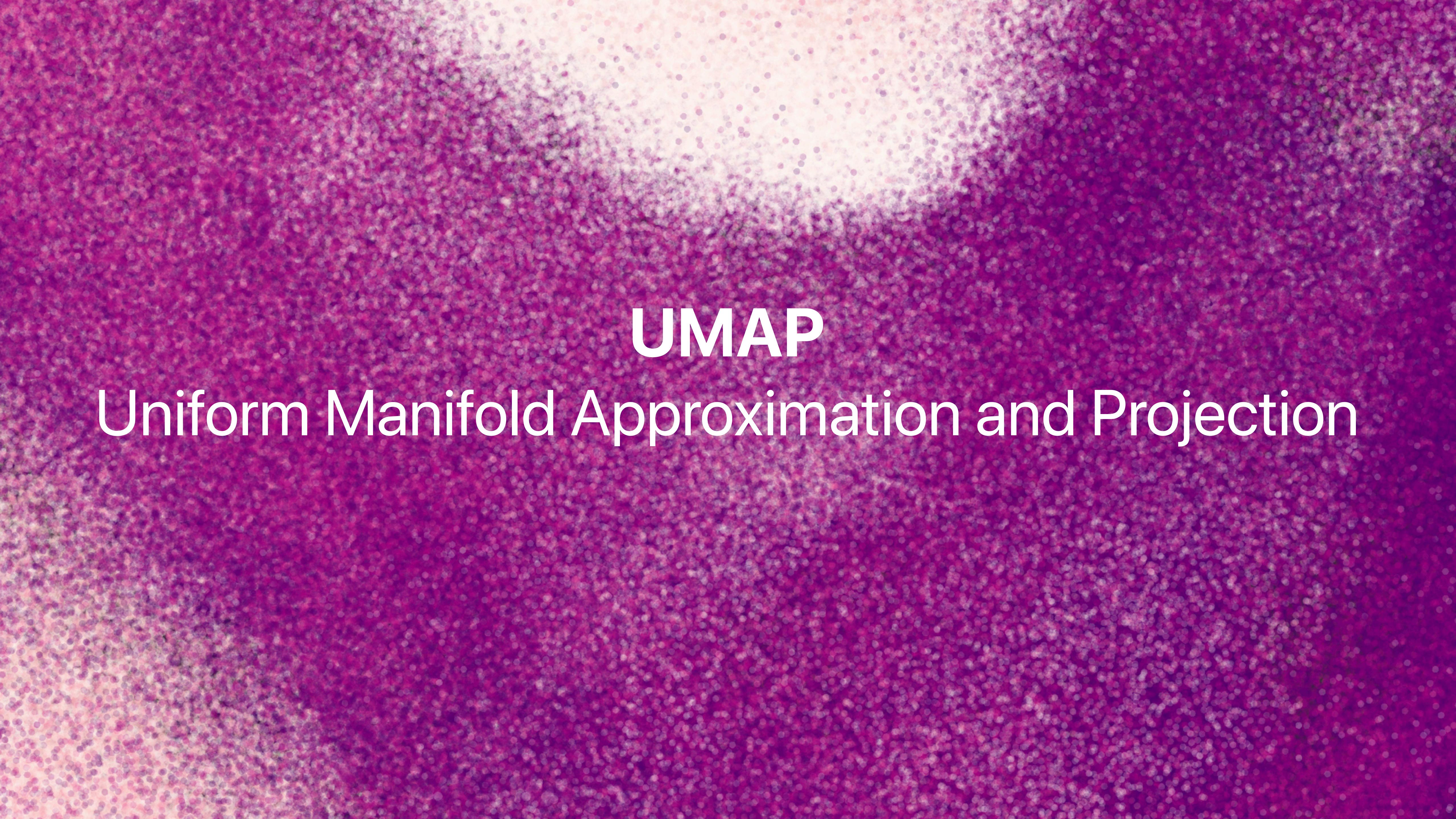
<https://lvdmaaten.github.io/tsne/>

Summary: t-SNE

- NON-LINEAR method of dimensionality reduction
- It is the current GOLD-STANDARD method in single cell data (including scRNA-seq)
- Can be run from the top PCs (e.g.: PC1 to PC10)

Problems:

- It does not learn an explicit function to map new points
- Its cost function is not convex – This means that the optimal t-SNE cannot be computed
- Many hyper-parameters need to be defined empirically (dataset-specific)
- It does not preserve global structure (in practice)



UMAP

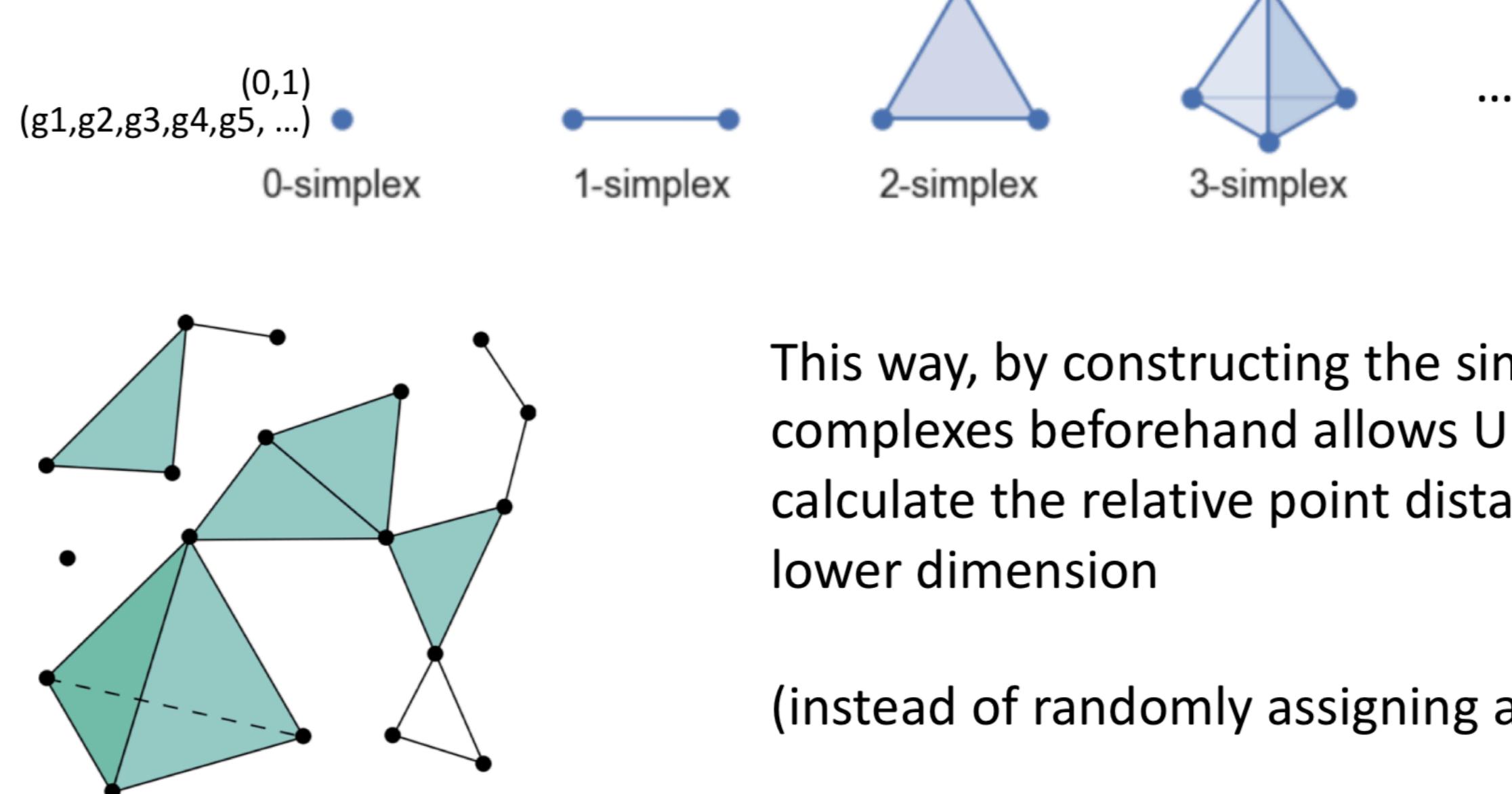
Uniform Manifold Approximation and Projection

UMAP in Brief

It is based on **topological structures** in multidimensional space (simplices)

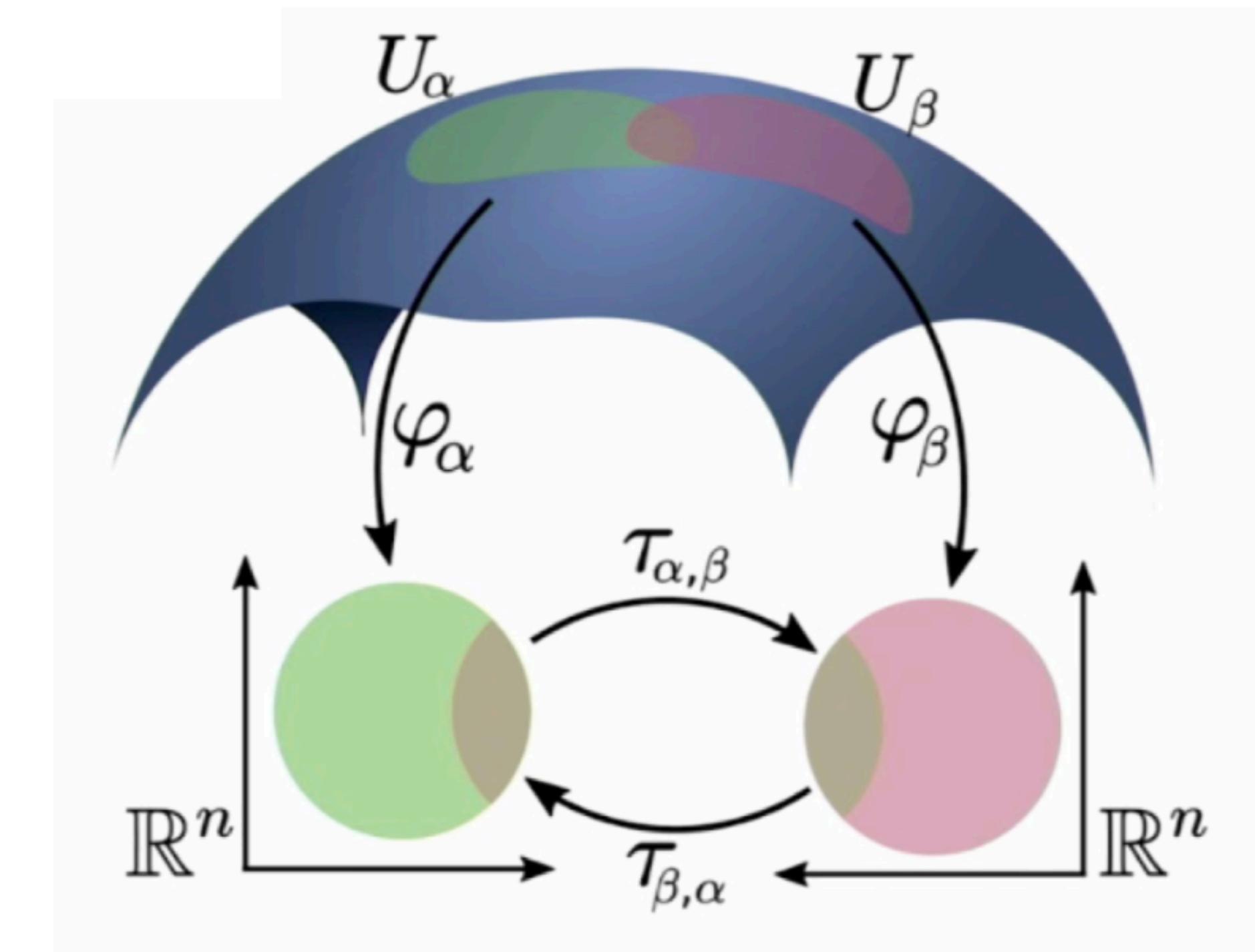
Points are connected with a line (edge) if the distance between them is below a threshold:

- Any distance metric can be used (euclidean)



This way, by constructing the simplicial complexes beforehand allows UMAP to calculate the relative point distances in the lower dimension

(instead of randomly assigning as in tSNE)

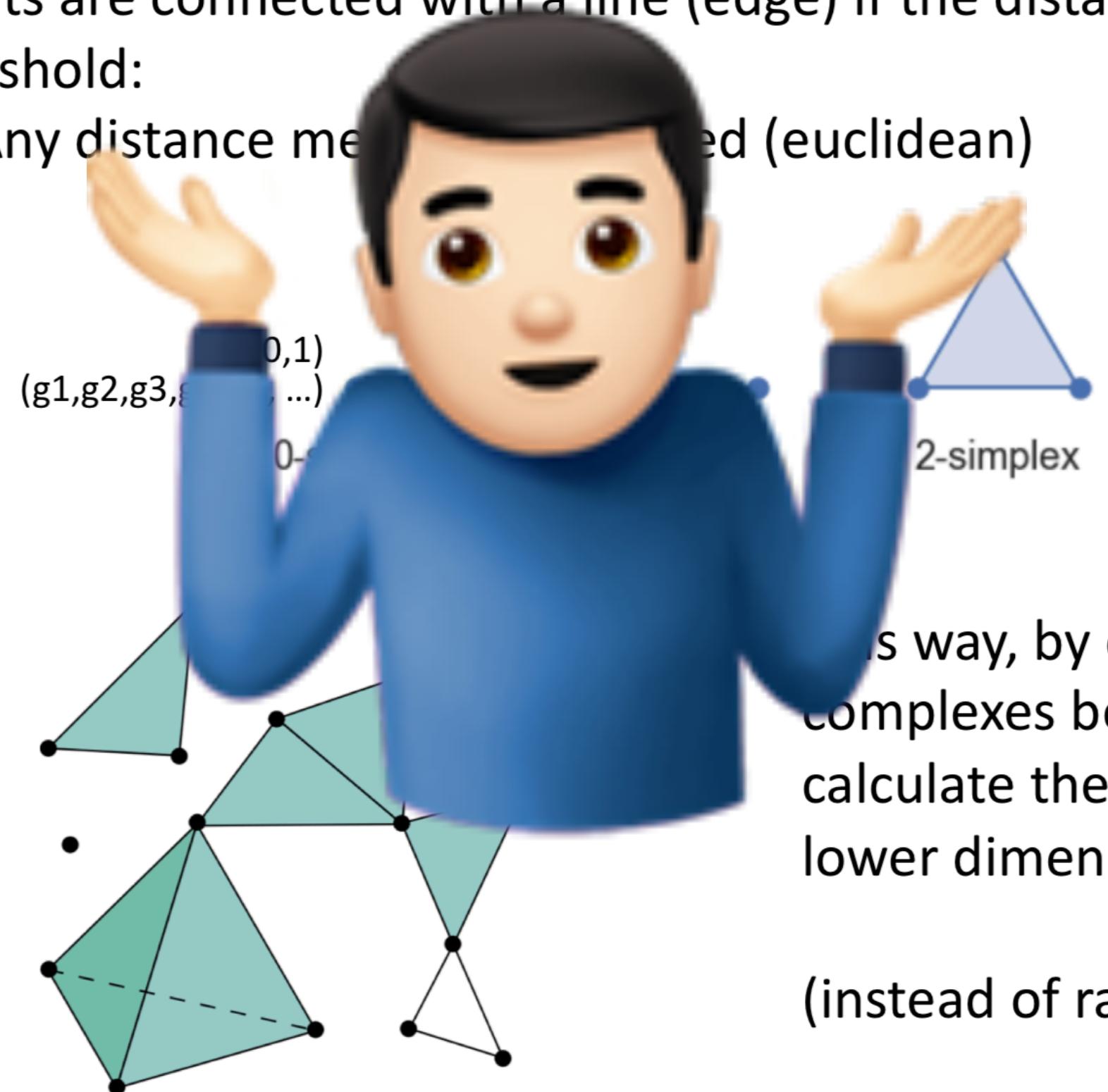


UMAP in Brief

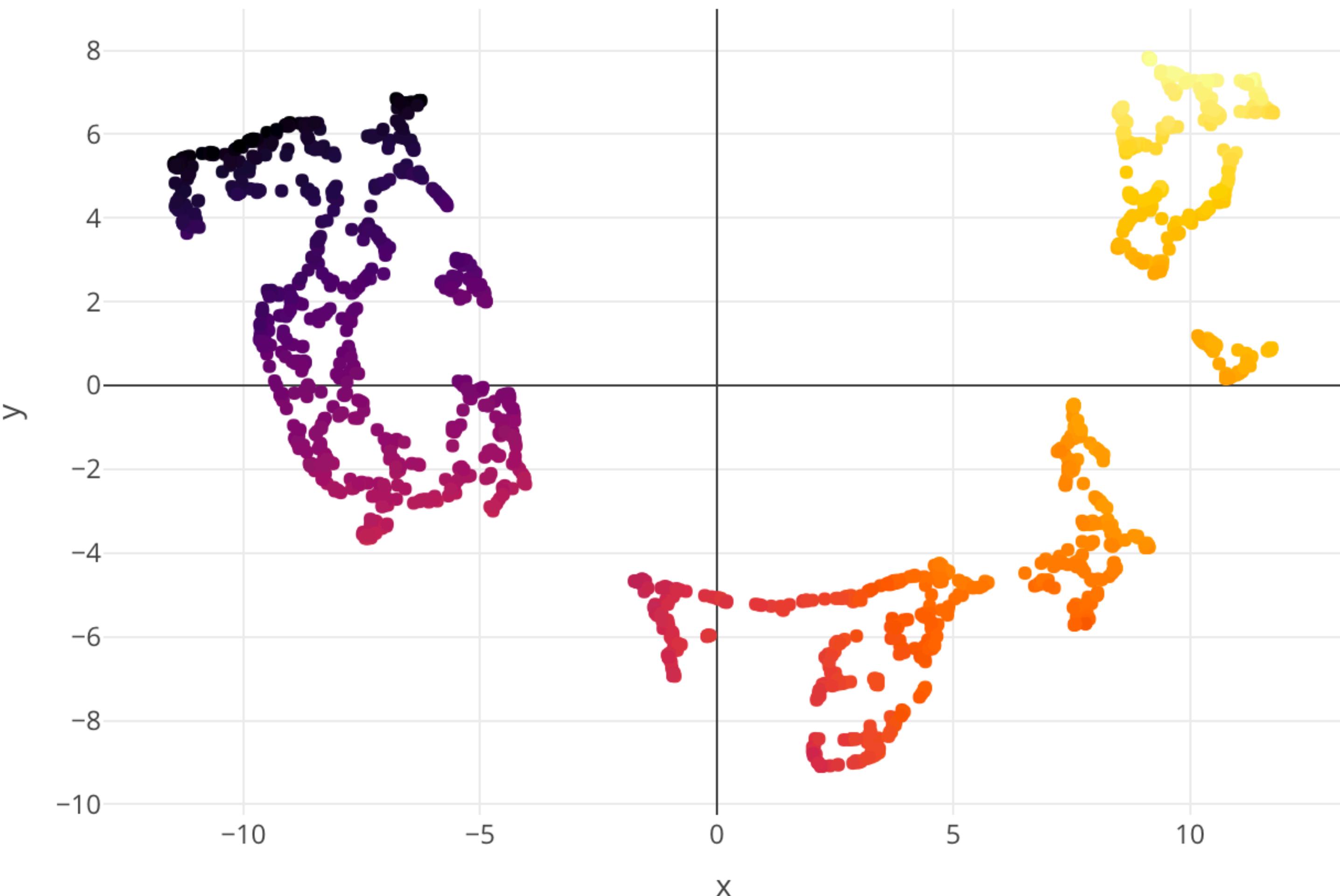
It is based on **topological structures** in multidimensi

Points are connected with a line (edge) if the distance threshold:

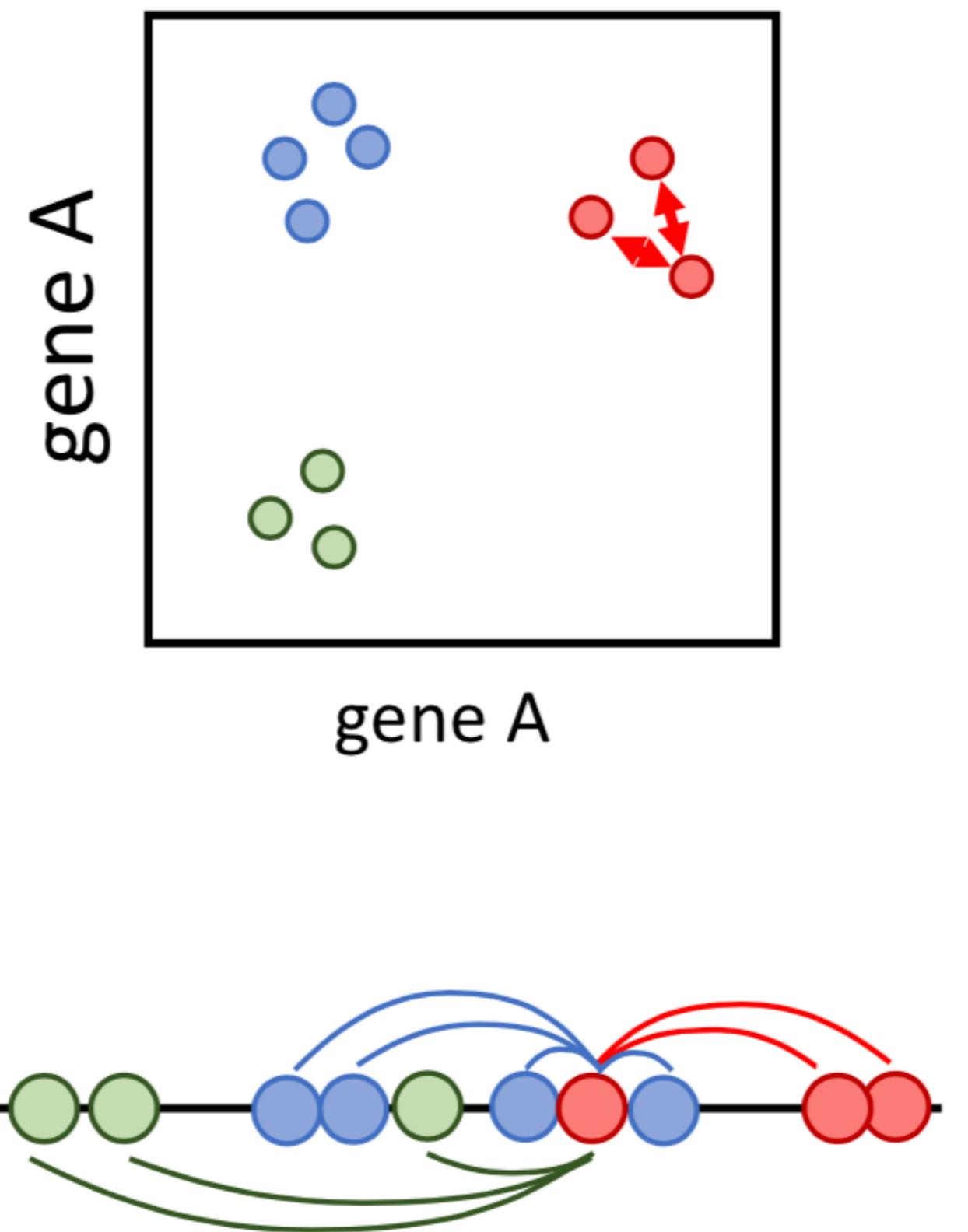
- Any distance metric can be used (euclidean)



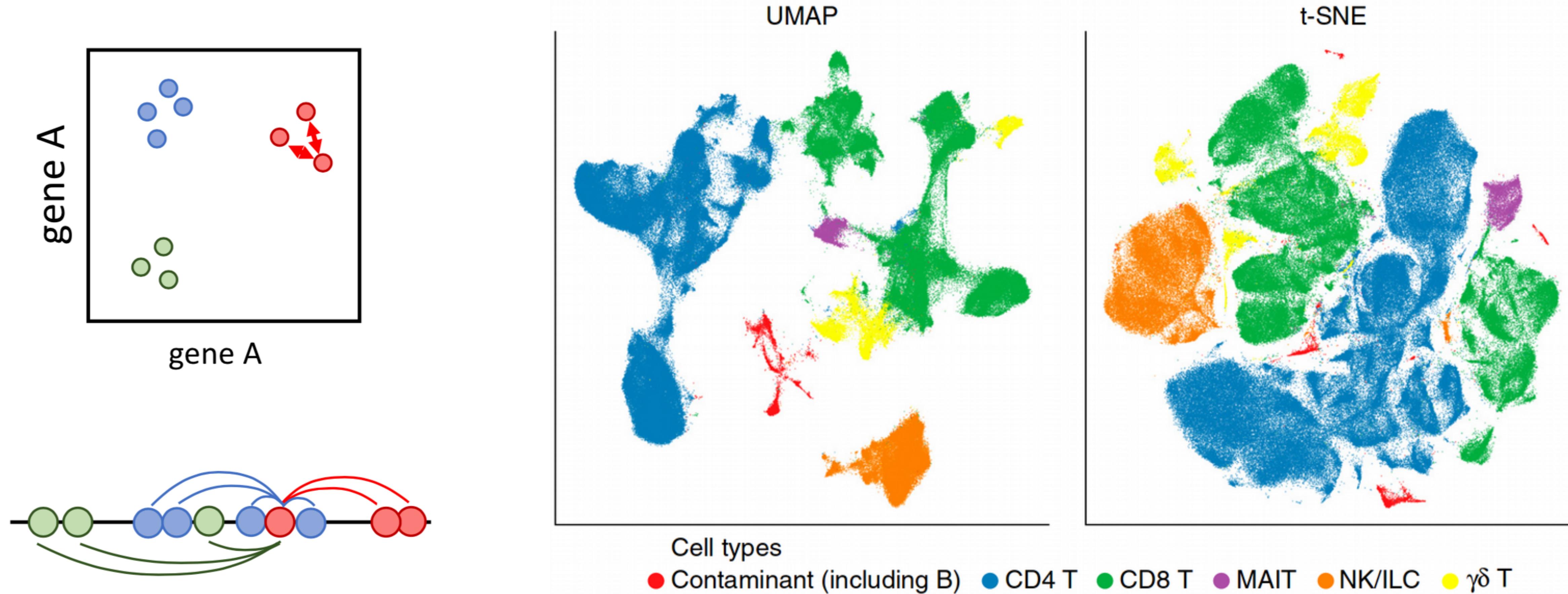
This way, by calculating the persistence of these complexes before calculating the reduced homology, we can calculate the representations of lower dimensional features (instead of random noise).



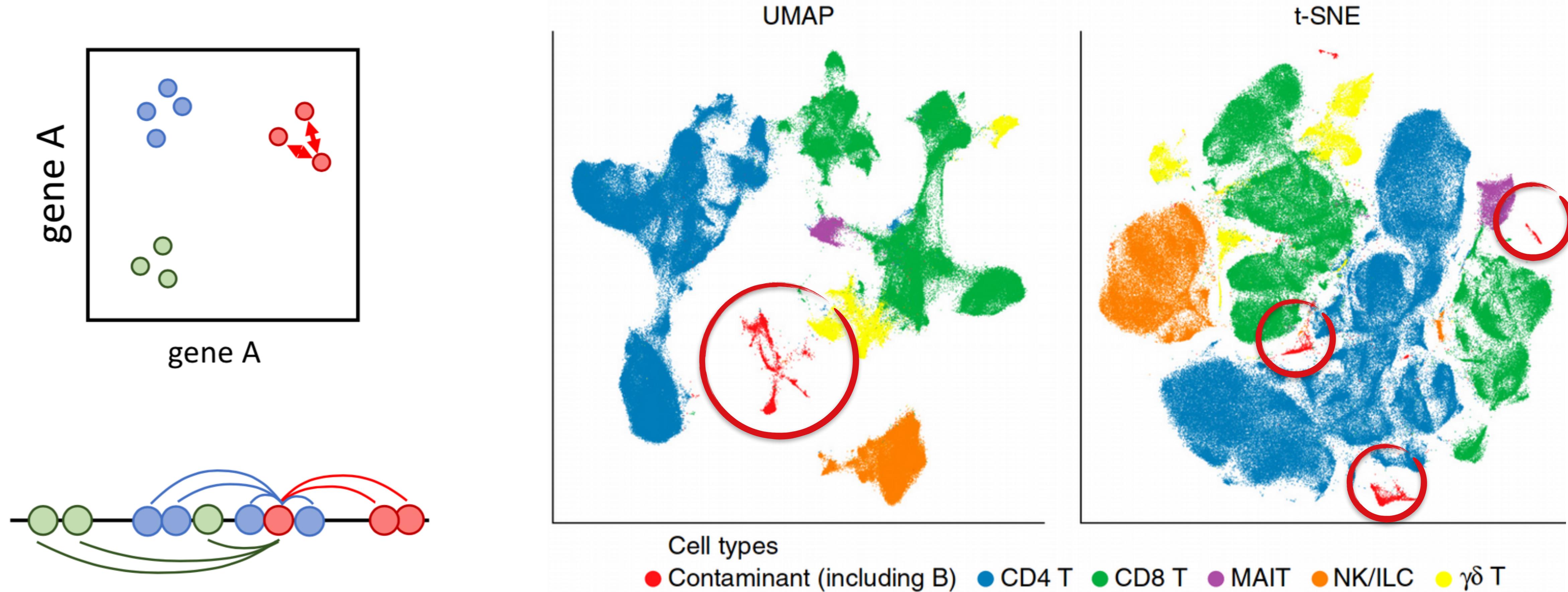
UMAP in Brief



UMAP in Brief



UMAP in Brief

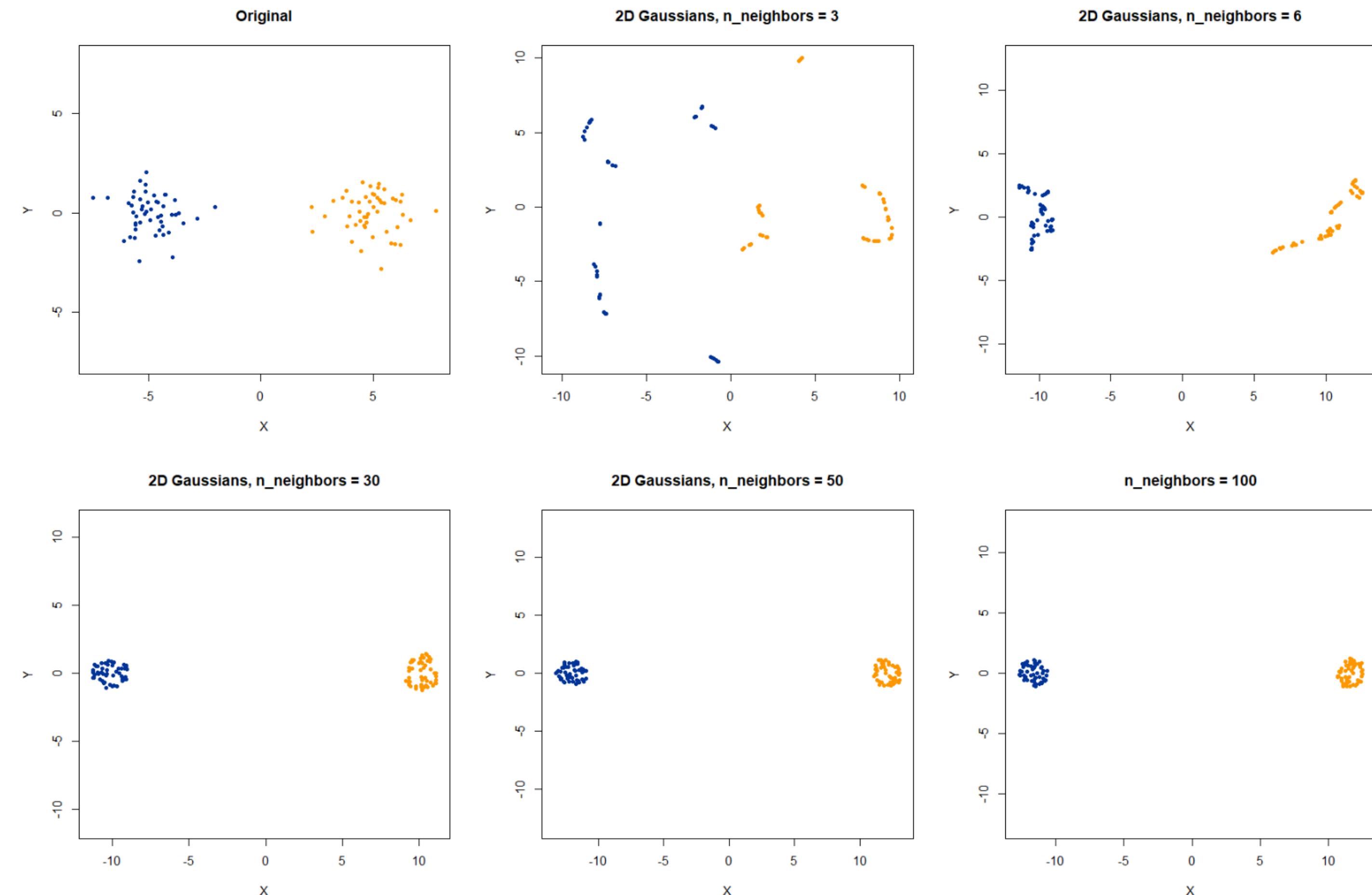


UMAP Parameters

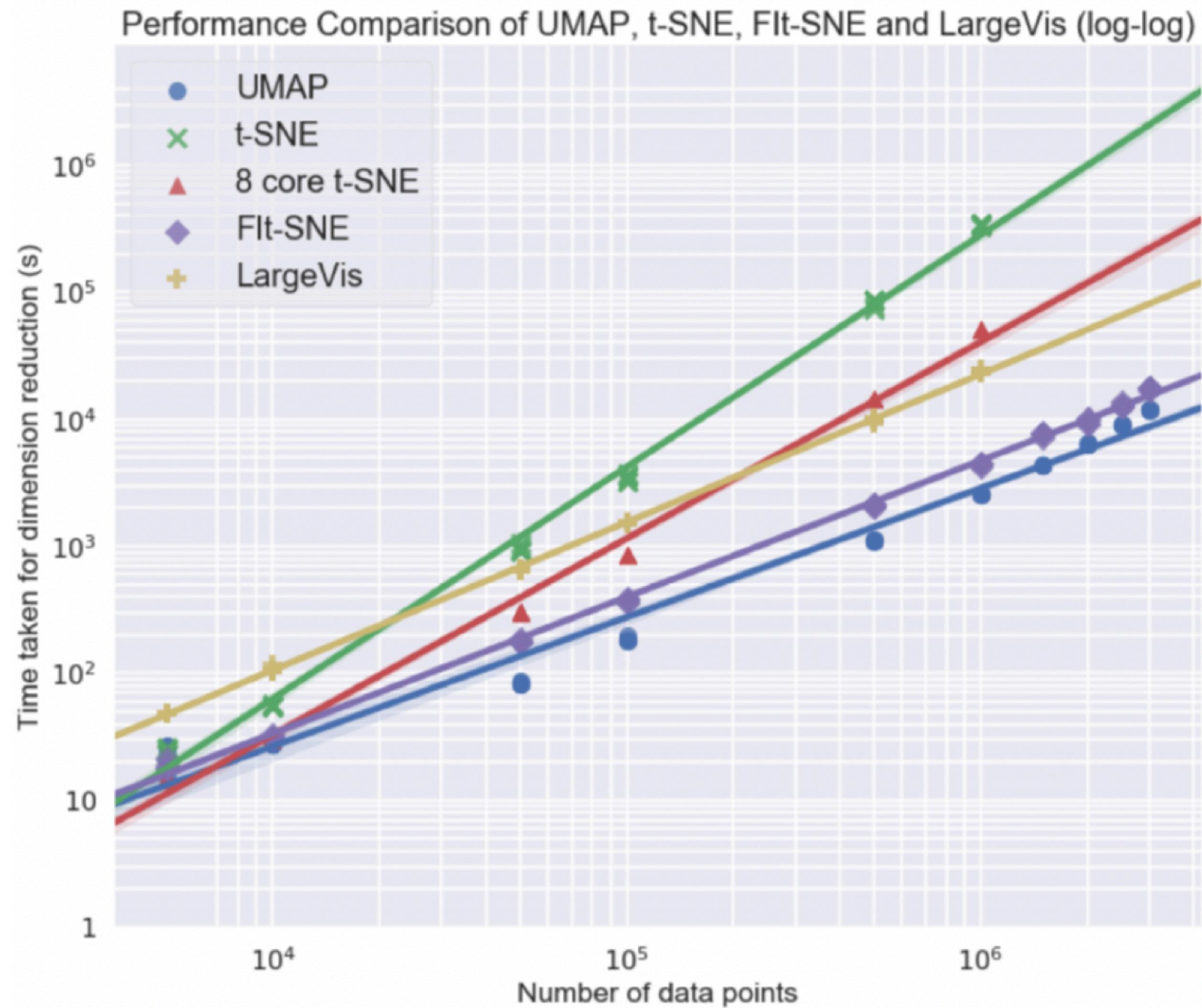
- Number of neighbors
- Number of iterations
- Minimum Distance (low-D)

- Metrics

...

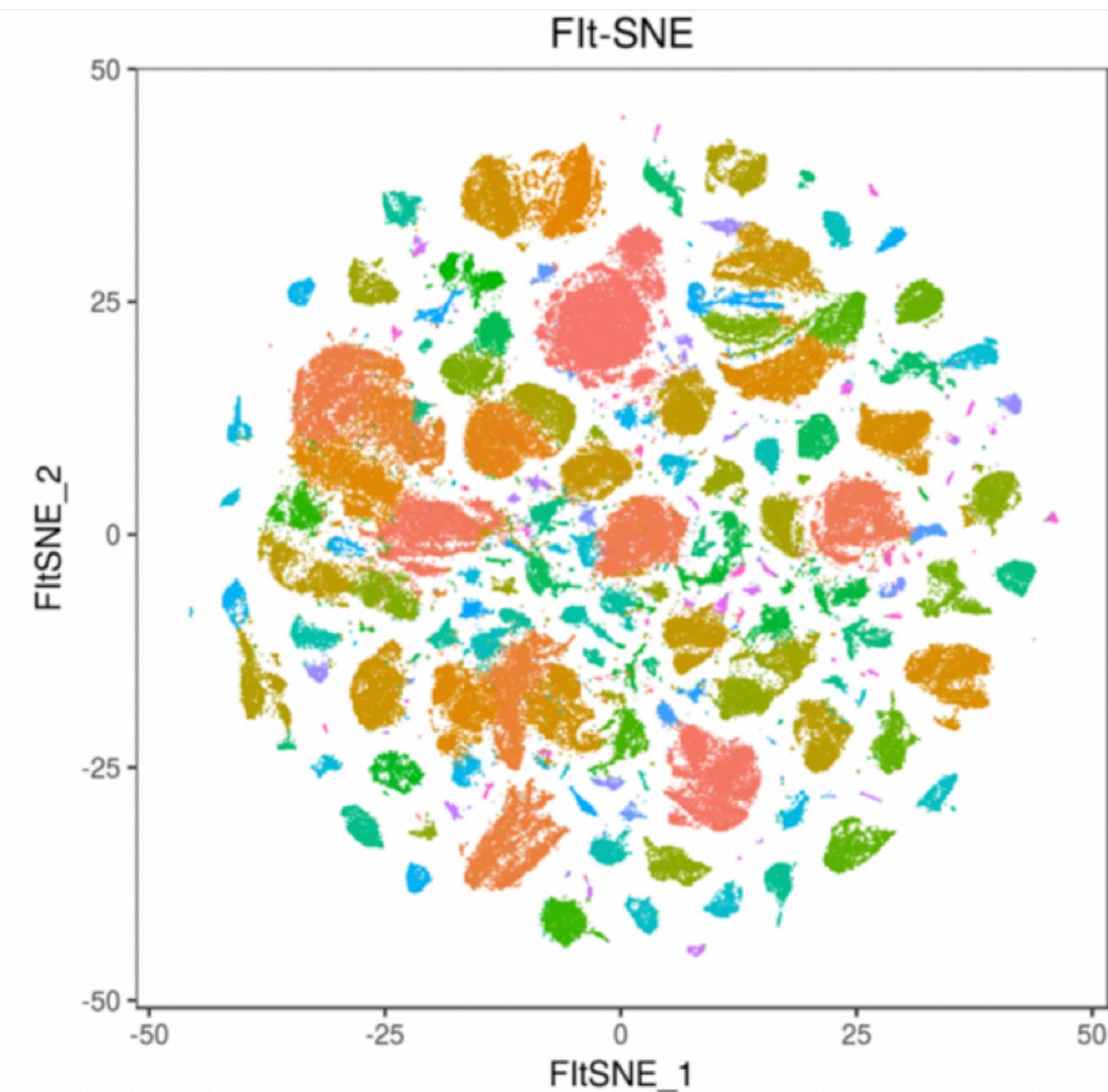


UMAP Parameters

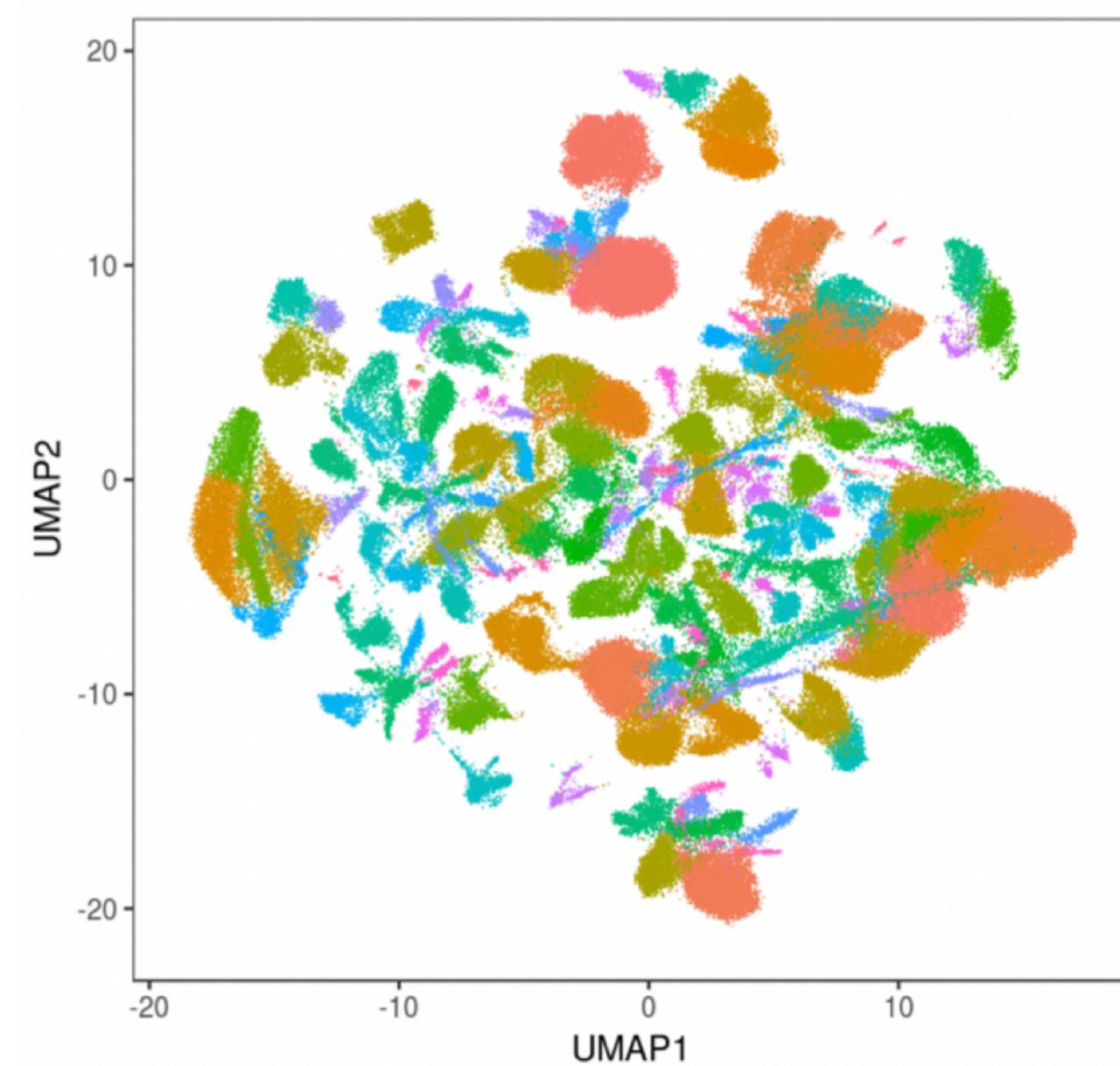


250k cells

Flt-SNE
1 hour



UMAP
7 minutes



Summary: UMAP

- NON-LINEAR method of dimensionality reduction
- Very efficient to compute
- Can be run from the top PCs (e.g.: PC1 to PC10)
- Is not randomly initialized and allows
- It should preserve global structure

Problems:

- It is designed to group cells stronger than t-SNE to show meaningful larger distances
- Similar number of hyper-parameters as t-SNE

<https://www.lcbc.nl>

Acknowledgements:
LKEB, IHB, LCBC @ LUMC
CGV @ TU Delft



Leiden
Computational Biology Center

