

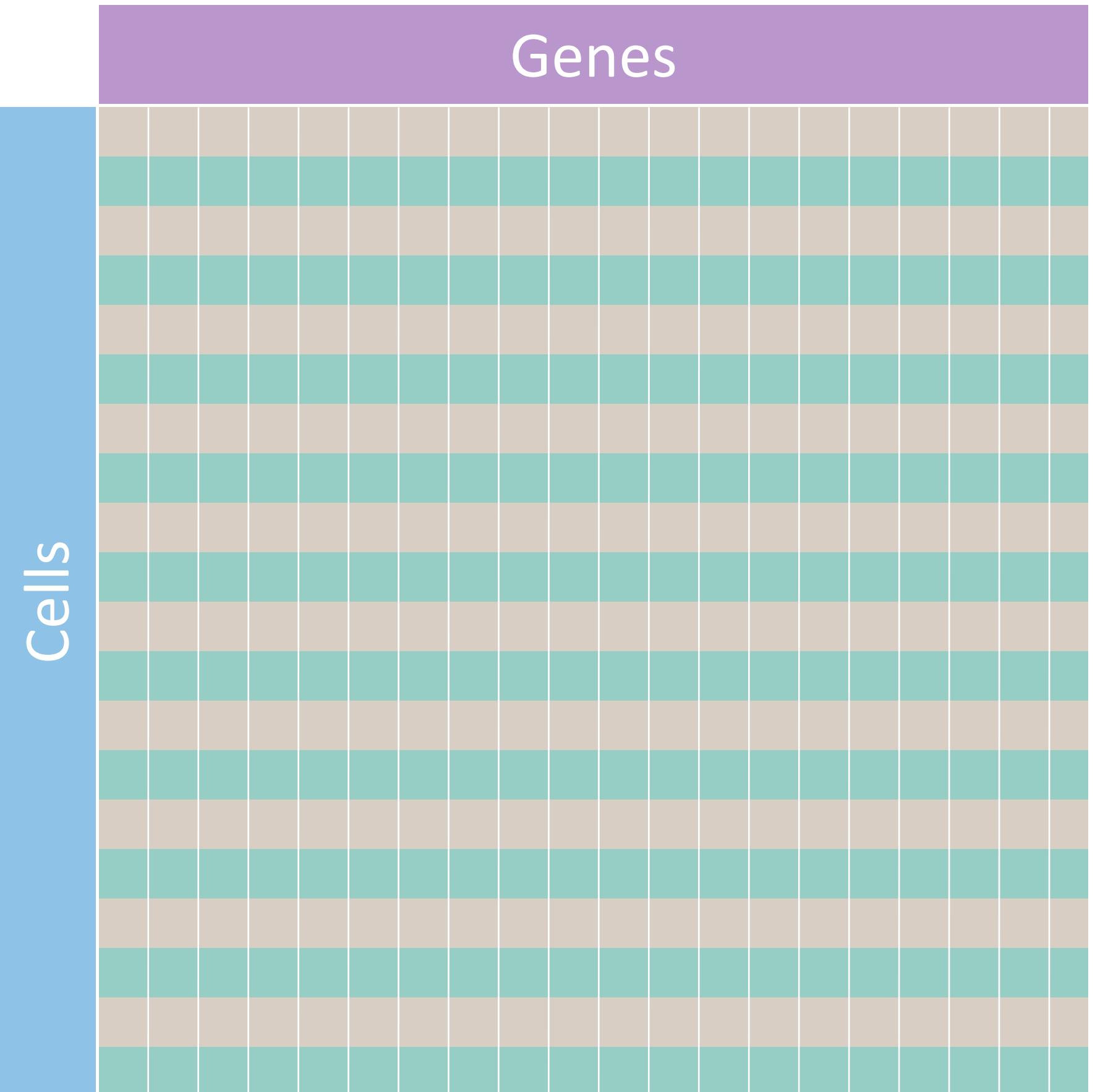
Interactive Visual Analysis with Dimensionality Reduction

Alexander Vieth - Computer Graphics & Visualization - TU Delft



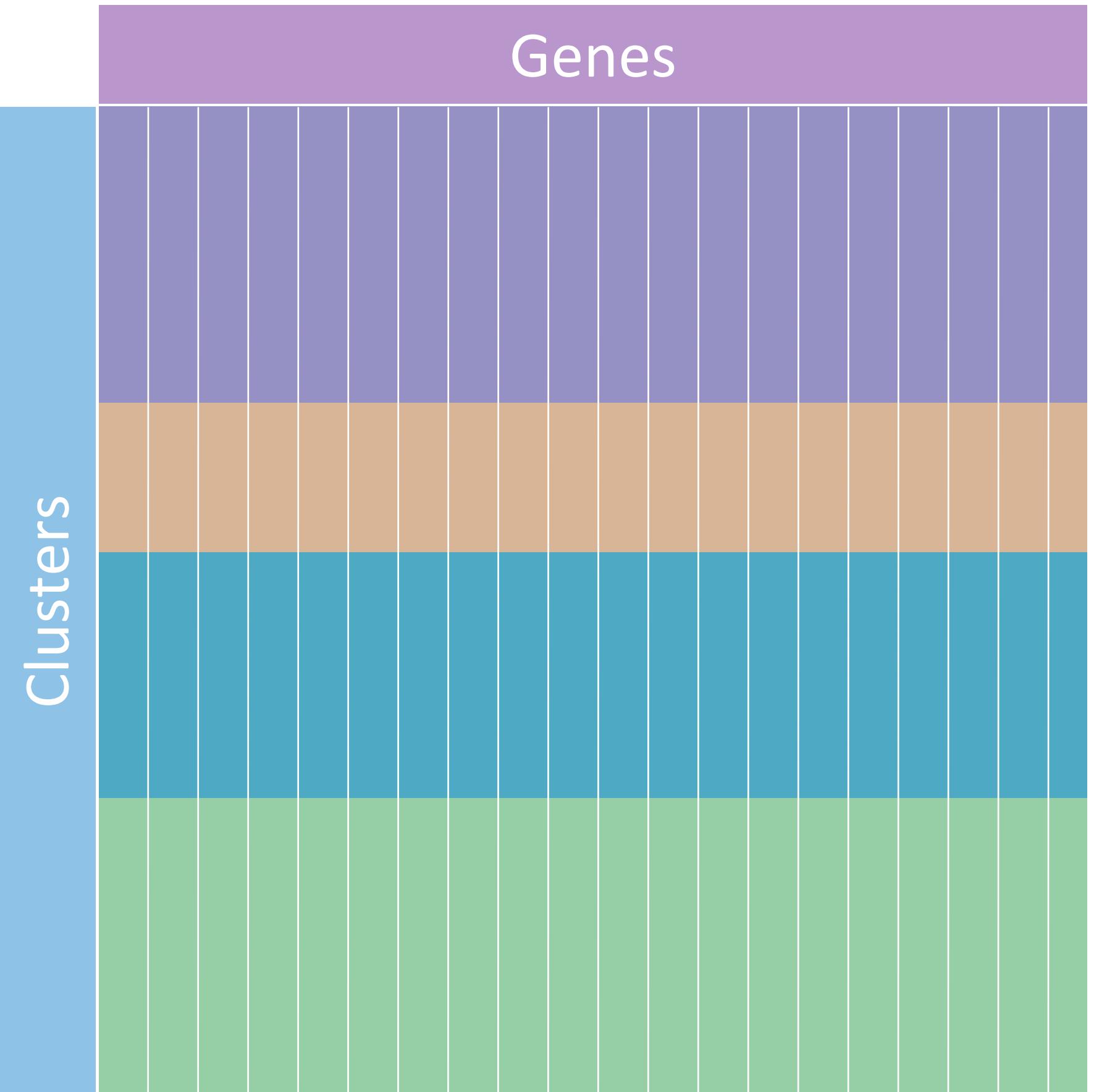
Dimensionality Reduction?

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



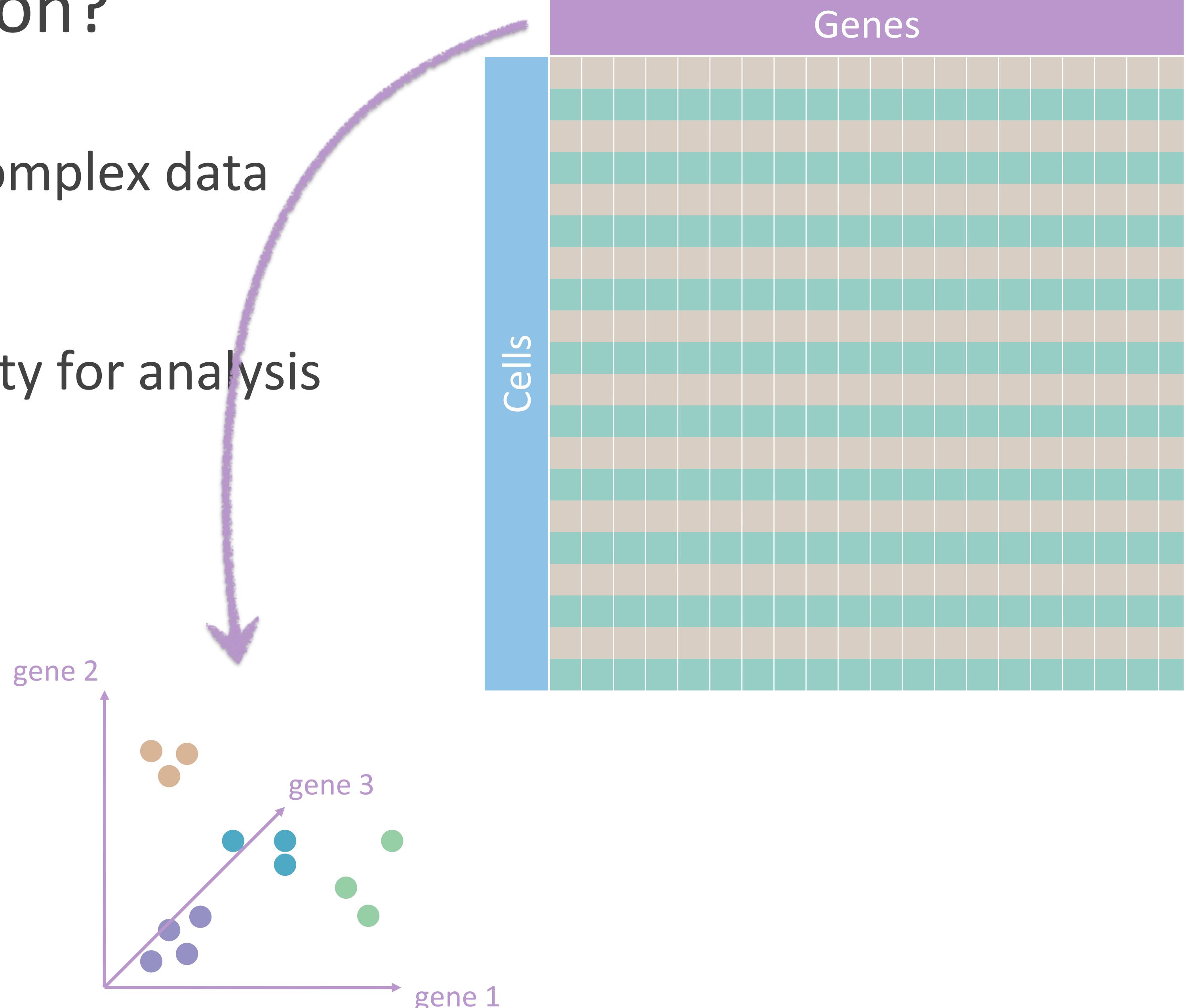
Dimensionality Reduction?

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



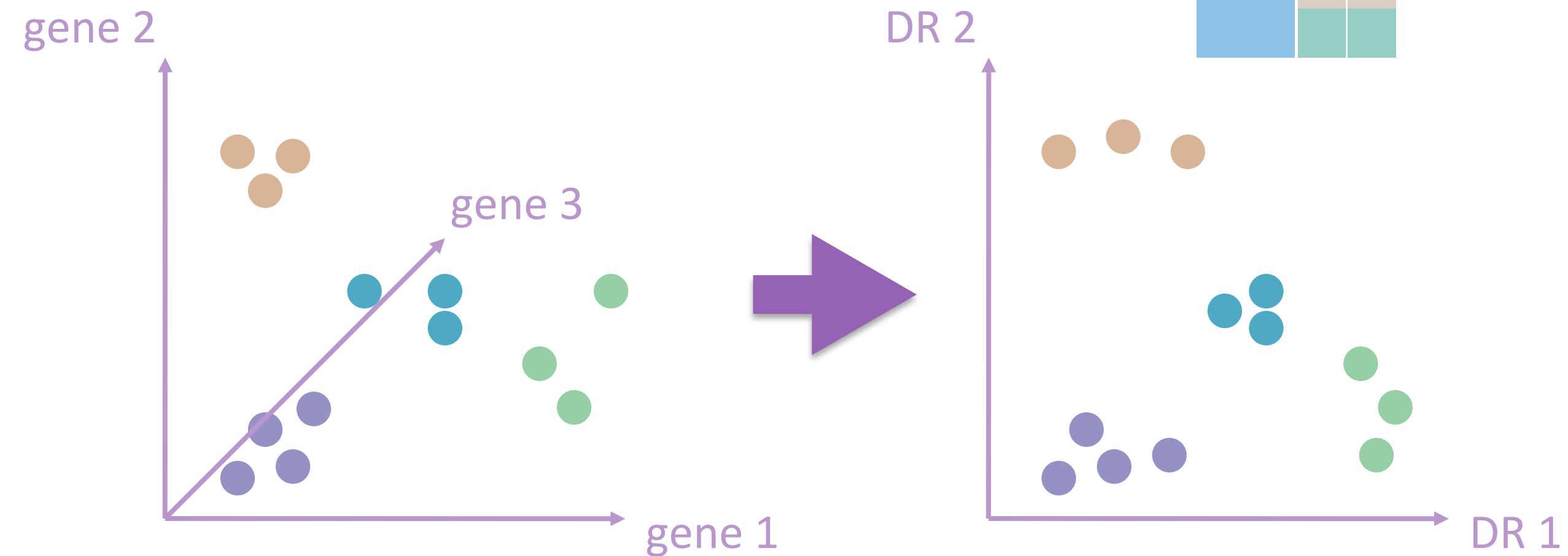
Dimensionality Reduction?

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



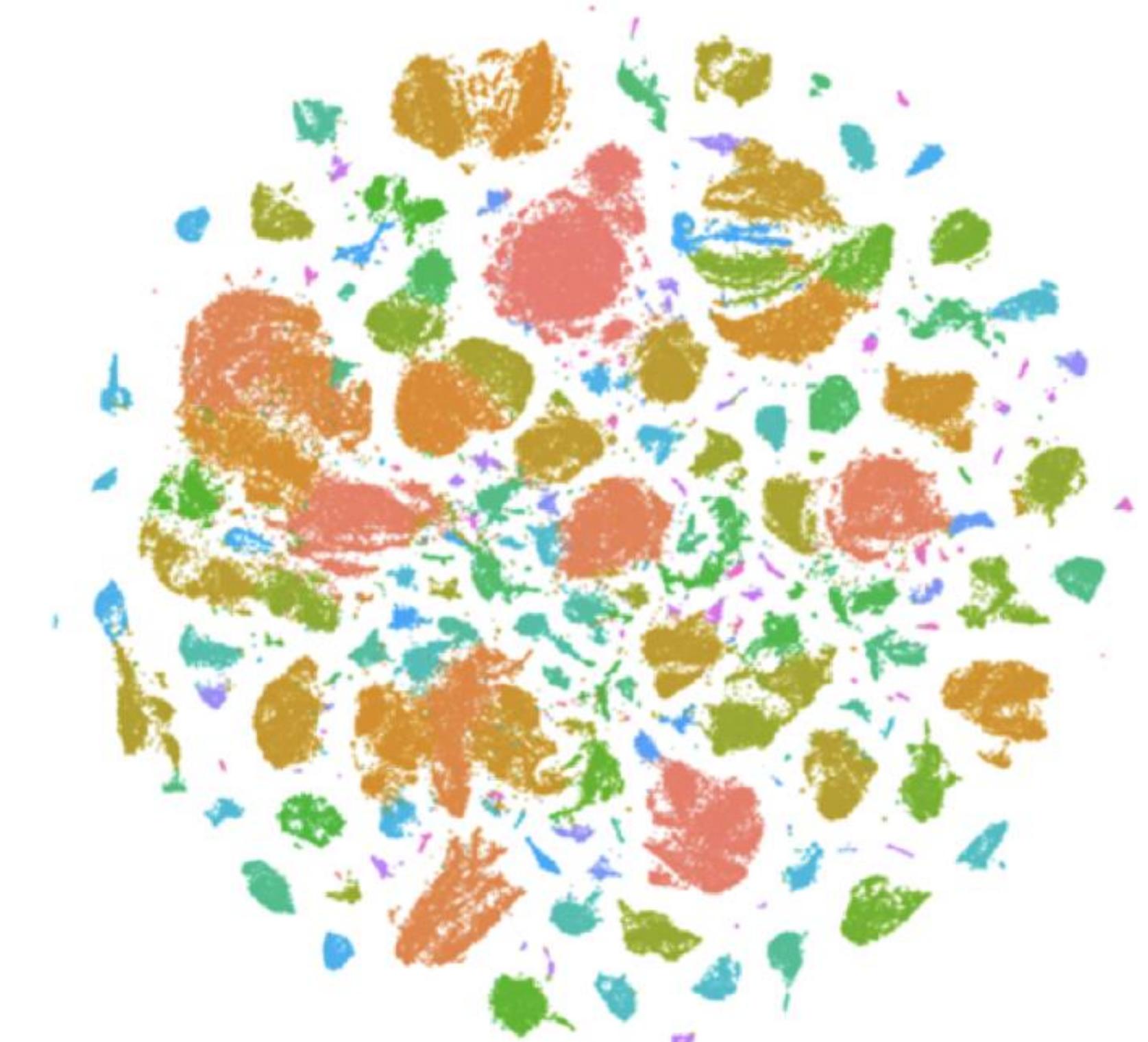
Dimensionality Reduction?

- 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



The case for interactive visual analysis

Numbers do not tell the whole story...

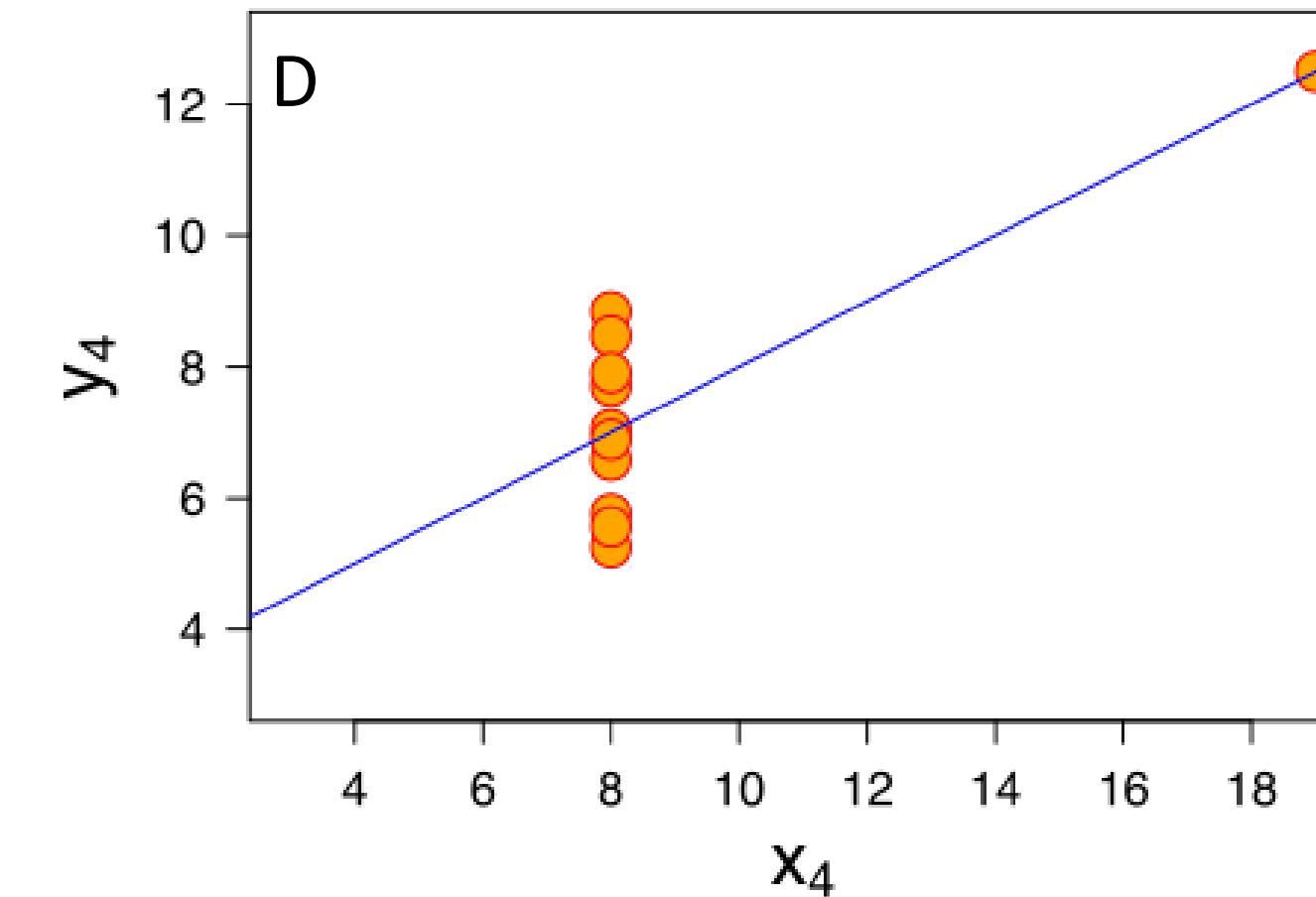
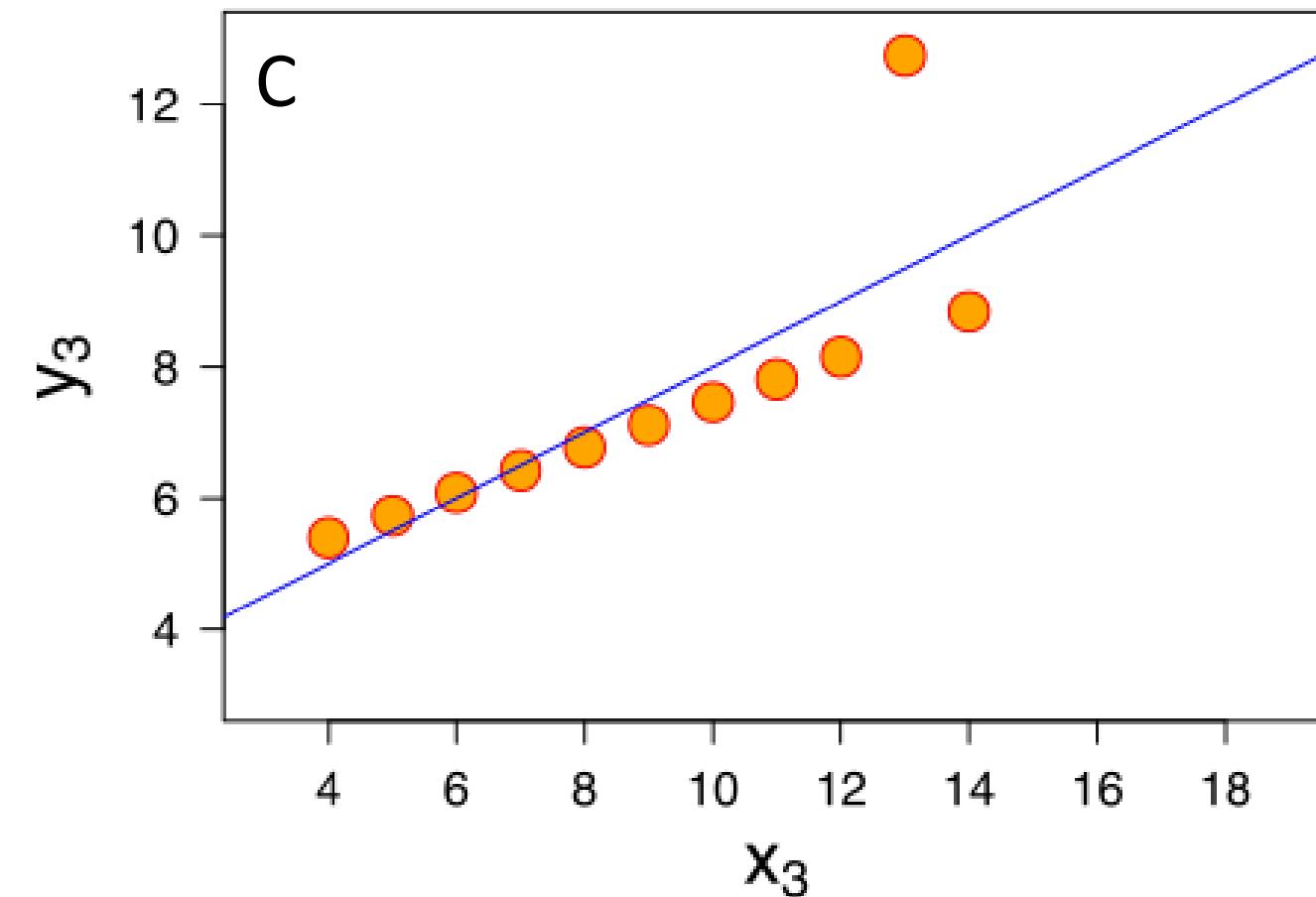
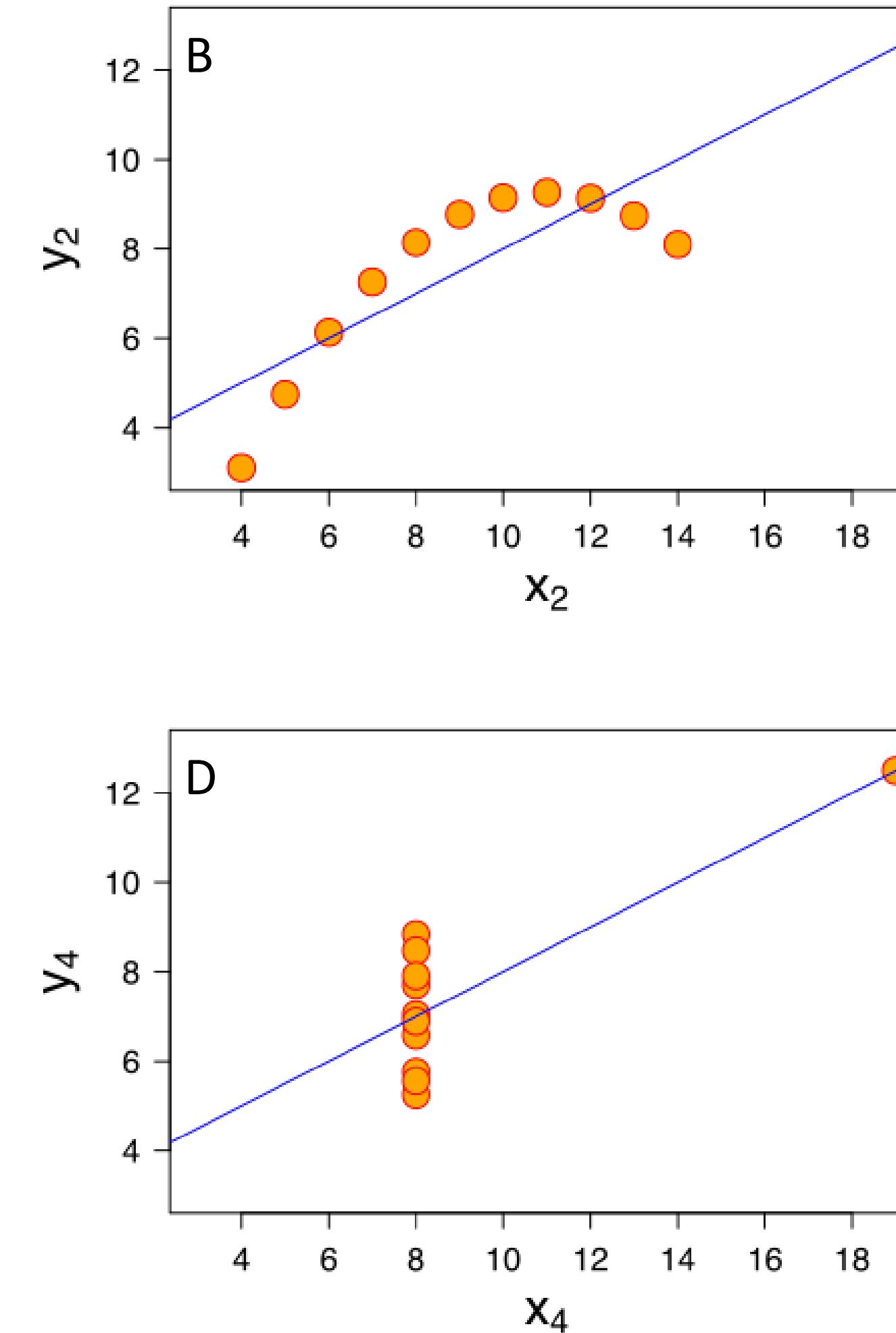
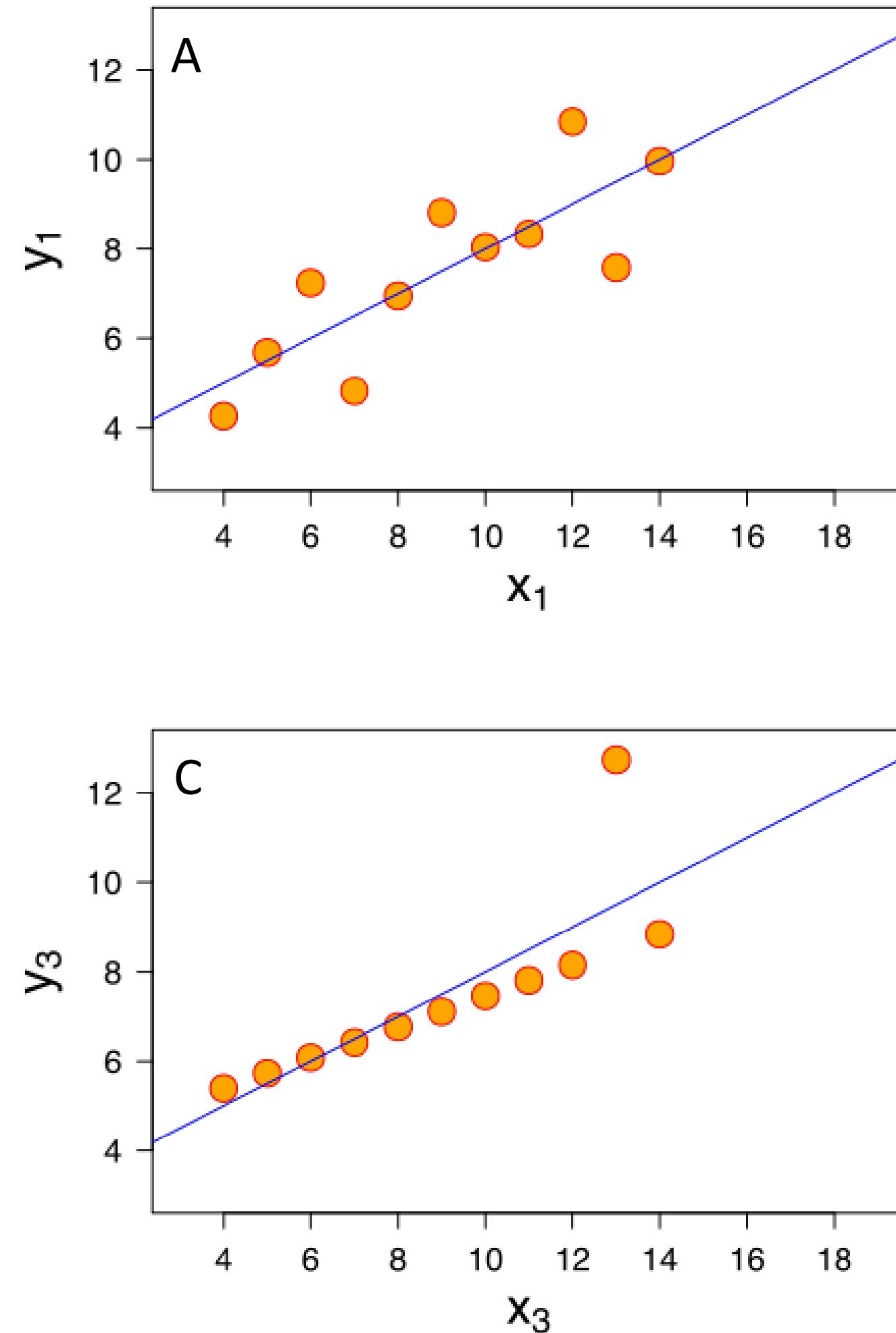
A		B		C		D	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89

Statistics:

- Mean(x): A=B=C=D=9 exact
- Variance(x): A=B=C=D=11 exact
- Mean(y): A=B=C=D=7.50 (2 decimals)
- Variance(y): A=B=C=D=4.125 (+/- 0.003)
- Correlation(x,y): A=B=C=D=816 (3 decimals)

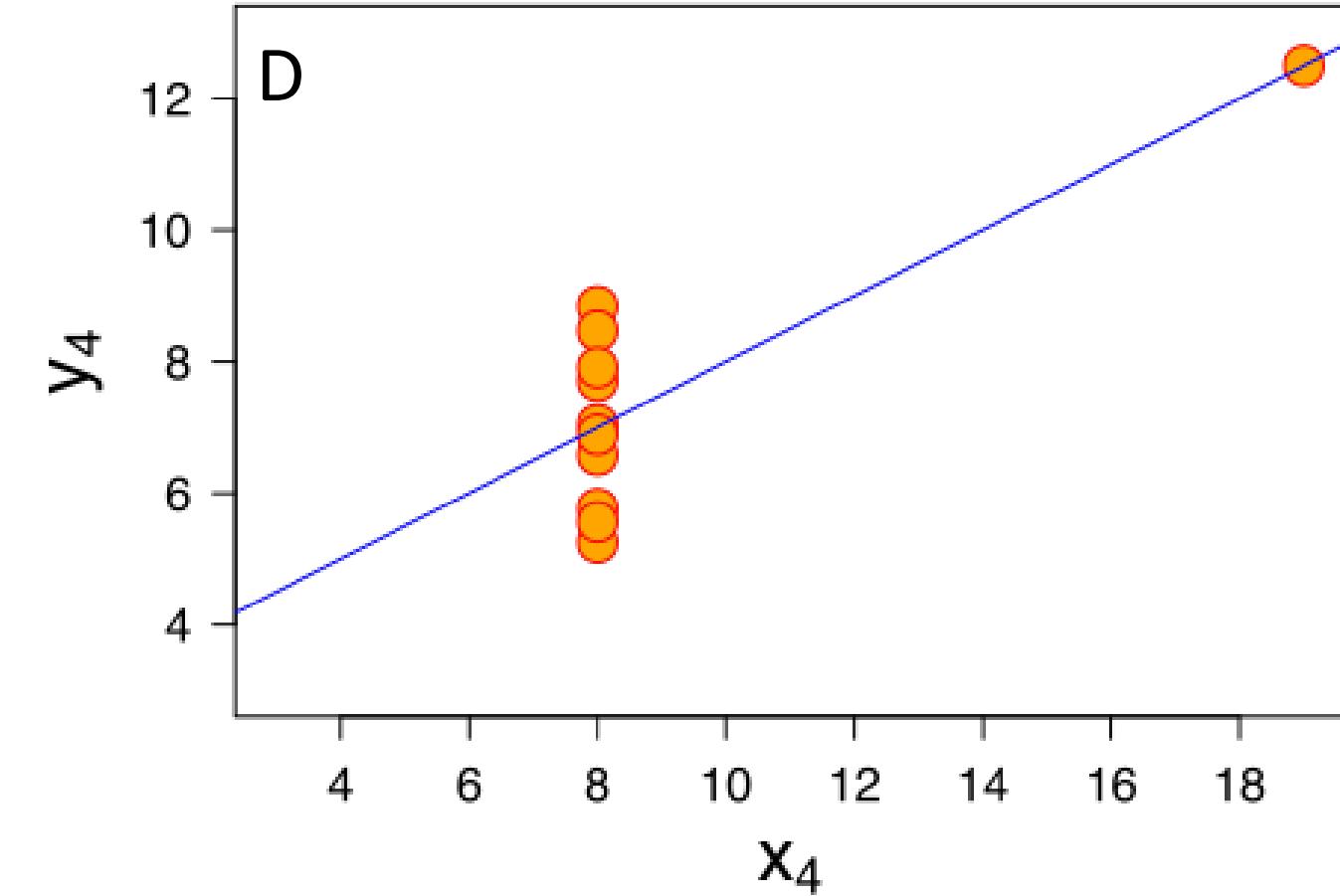
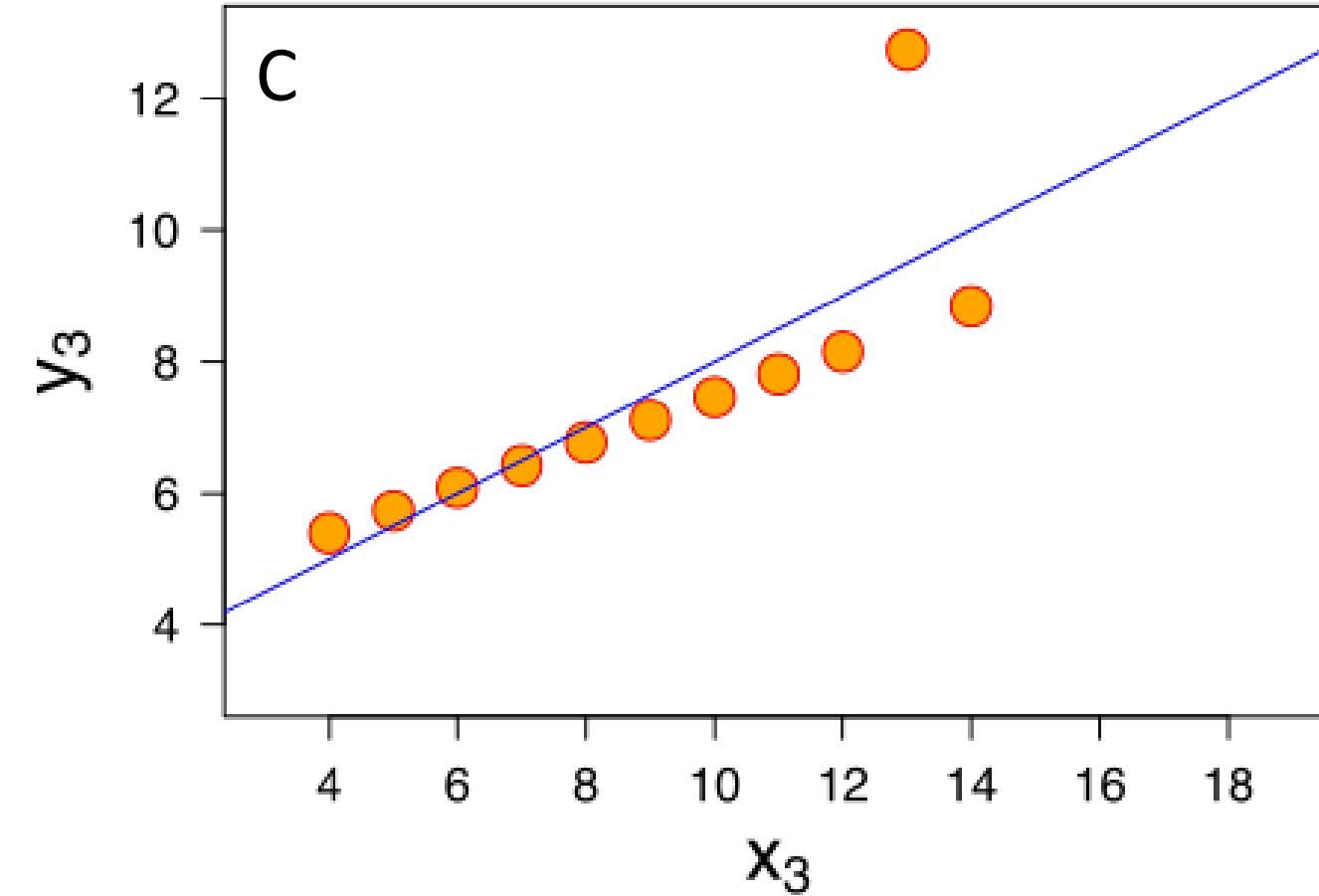
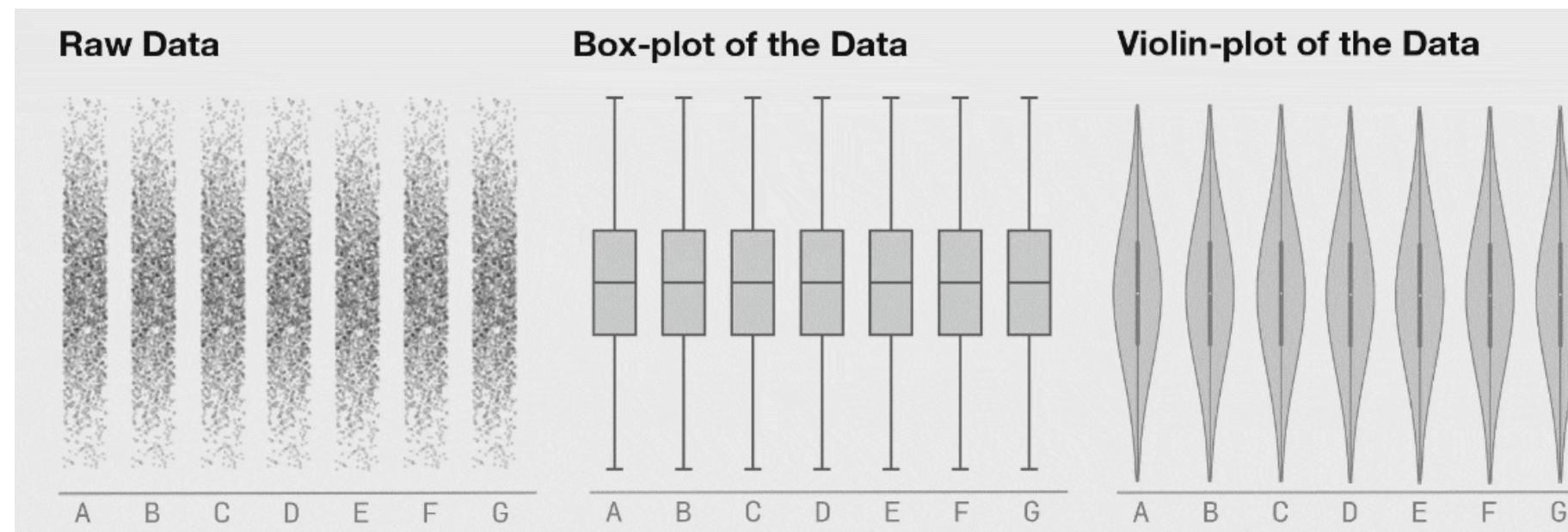
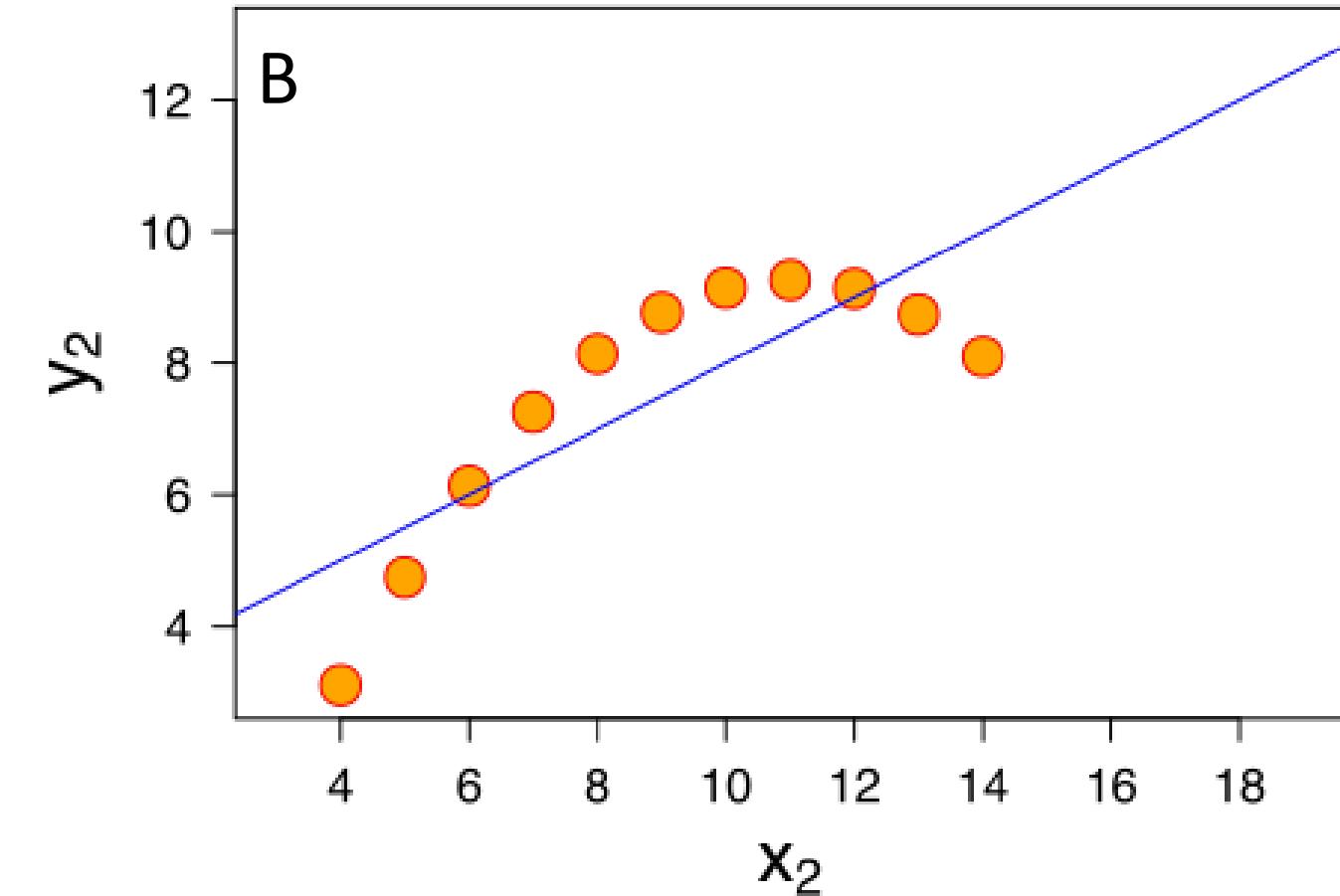
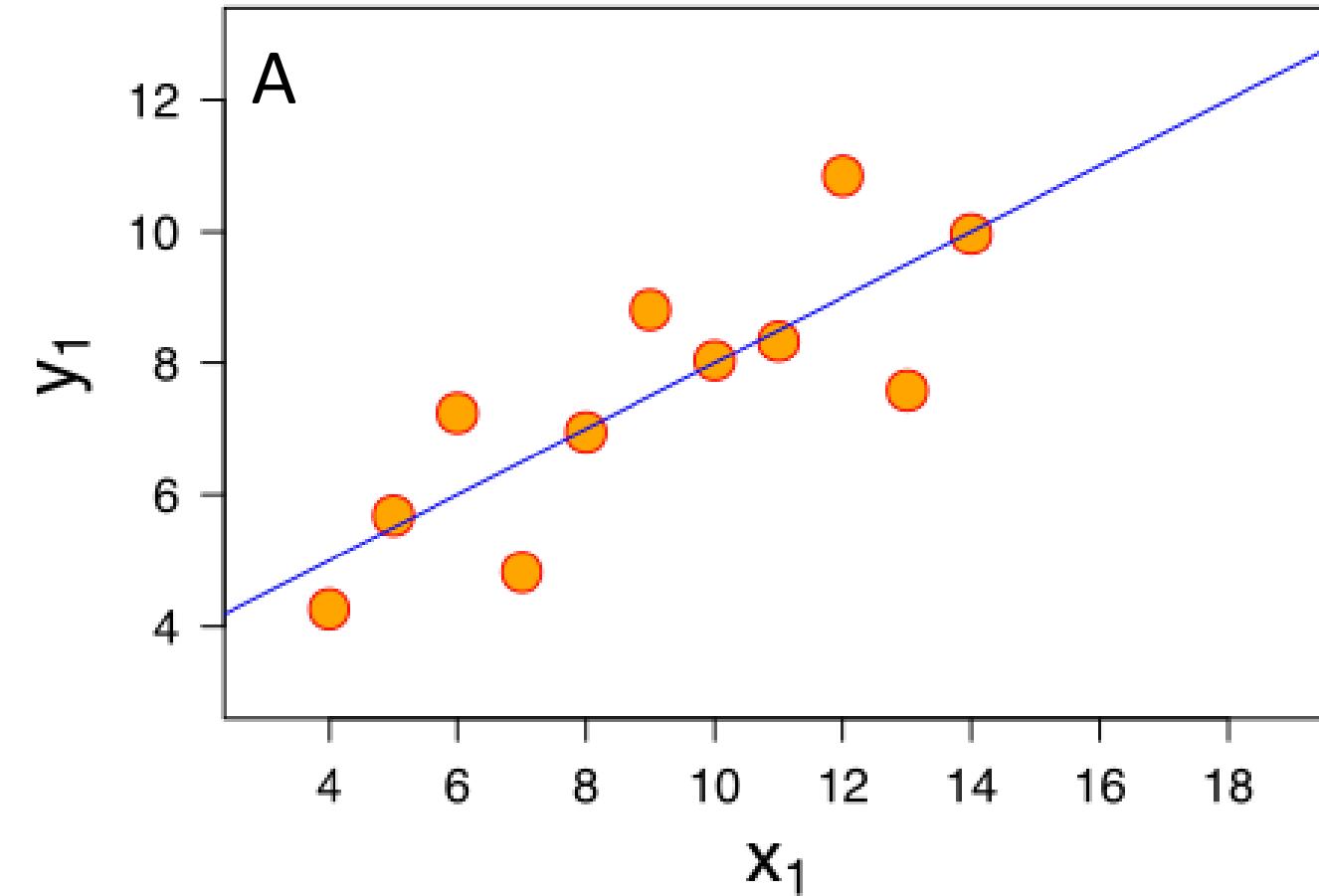
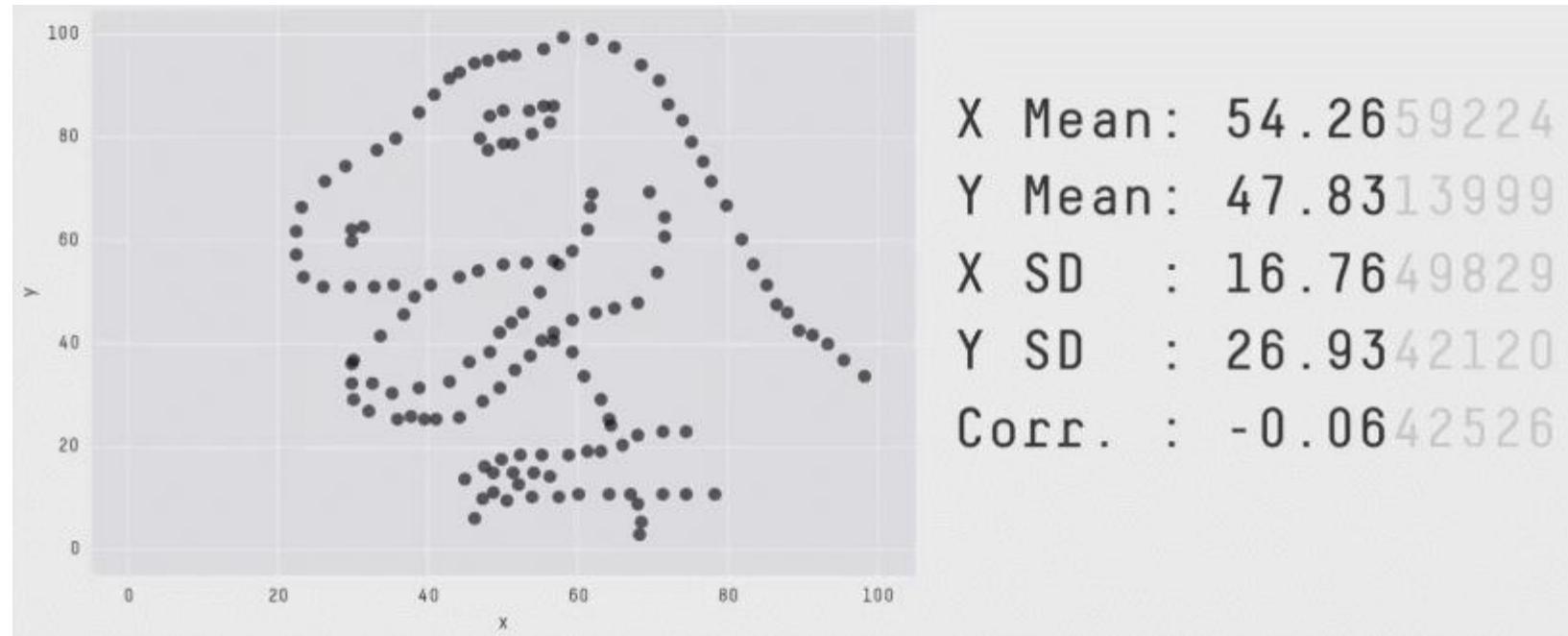
Numbers do not tell the whole story...

A		B		C		D	
x	y	x	y	x	y	x	y
10.0	8.04	10.0	9.14	10.0	7.46	8.0	6.58
8.0	6.95	8.0	8.14	8.0	6.77	8.0	5.76
13.0	7.58	13.0	8.74	13.0	12.74	8.0	7.71
9.0	8.81	9.0	8.77	9.0	7.11	8.0	8.84
11.0	8.33	11.0	9.26	11.0	7.81	8.0	8.47
14.0	9.96	14.0	8.10	14.0	8.84	8.0	7.04
6.0	7.24	6.0	6.13	6.0	6.08	8.0	5.25
4.0	4.26	4.0	3.10	4.0	5.39	19.0	12.50
12.0	10.84	12.0	9.13	12.0	8.15	8.0	5.56
7.0	4.82	7.0	7.26	7.0	6.42	8.0	7.91
5.0	5.68	5.0	4.74	5.0	5.73	8.0	6.89



Anscombe's Quartet, 1973

Numbers do not tell the whole story...

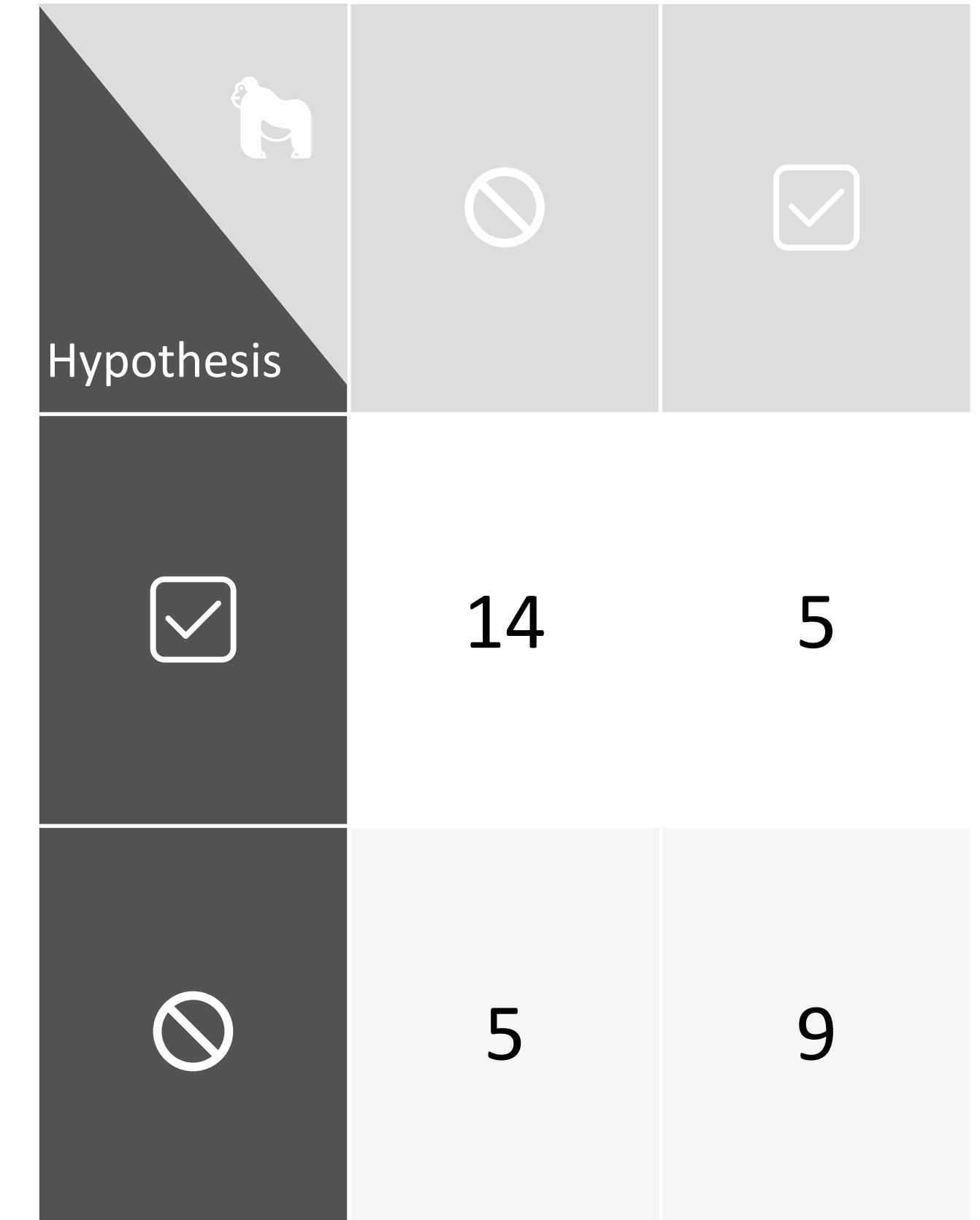
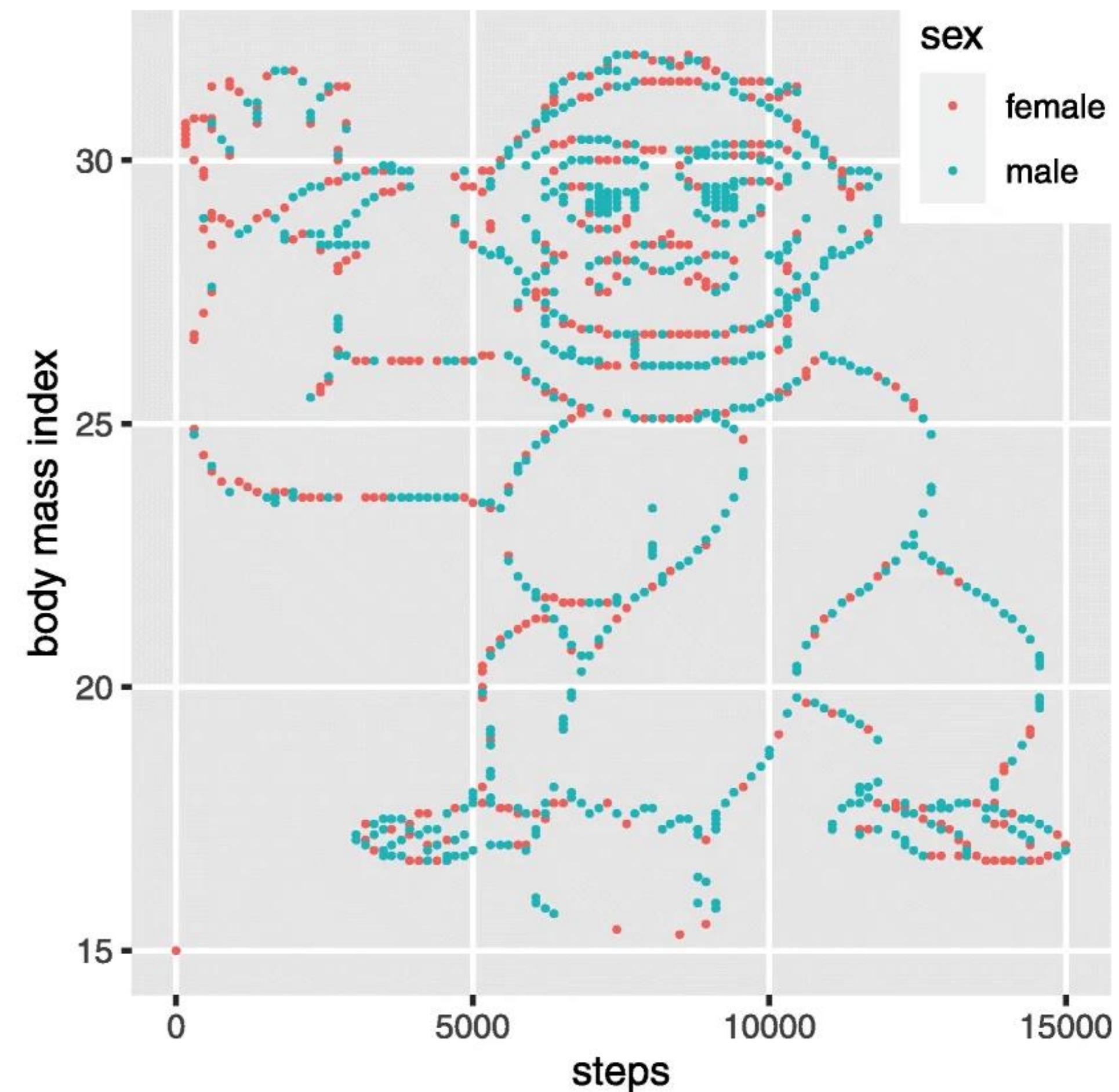


Justin Matejka and George Fitzmaurice. 2017
<https://doi.org/10.1145/3025453.3025912>

Anscombe's Quartet, 1973

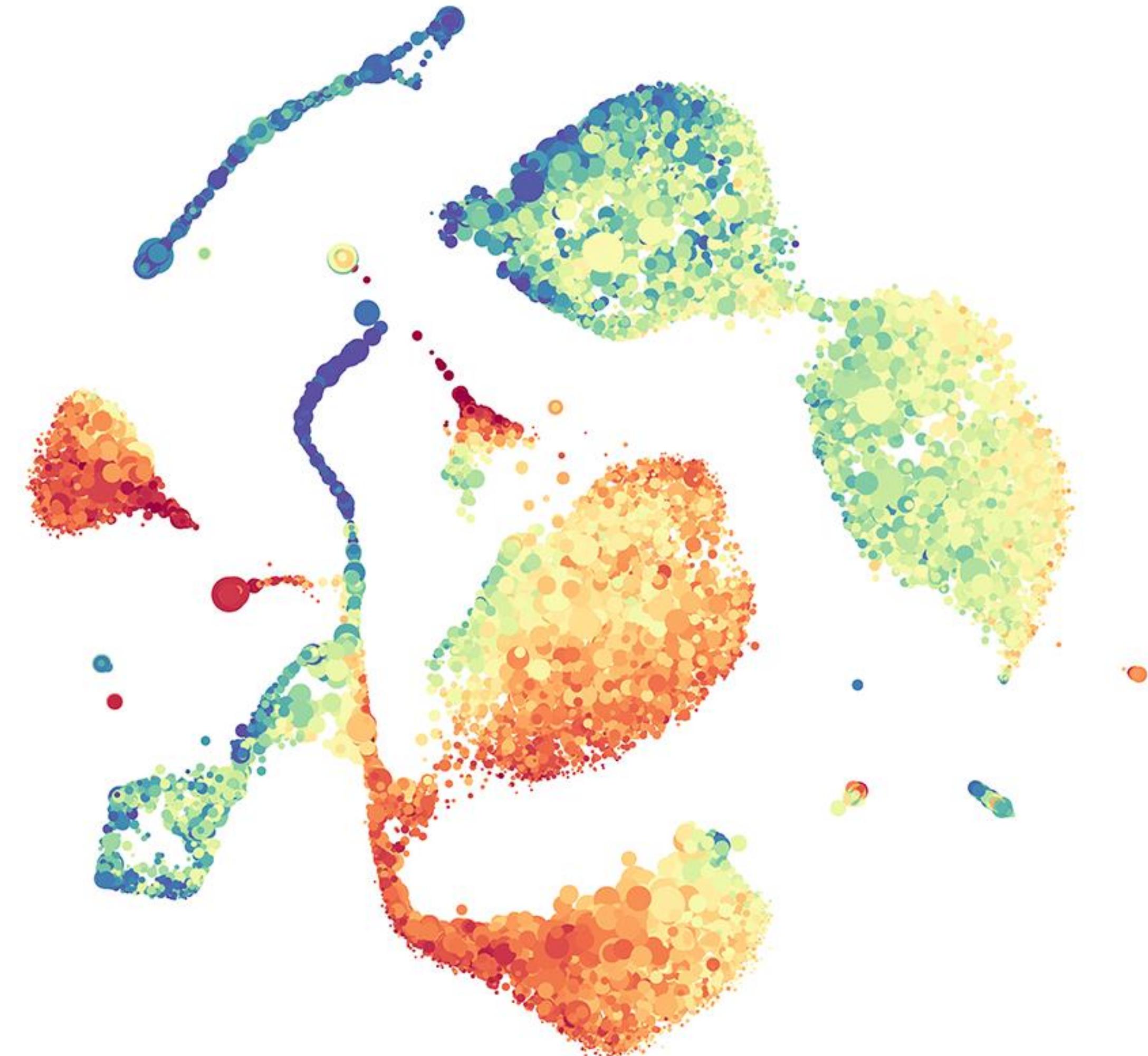
Do we look for the whole story?

ID	steps	bmi
3	15000	17.0
4	14861	17.2
5		
9		
12		
14		
15	1	16.9
16	2	16.9
21	6	16.8
23	7	16.8
26	8	17.3
28	10	20.5
31	11	20.6
33	13	20.5
34	17	20.4
35	18	20.4
36	19	19.8
38	20	19.7
39	22	19.7
41	24	19.6
44	25	19.6
45	27	19.6
<	29	17.4
30	14560	17.4
32	14398	20.9
37	14398	17.5
40	14398	17.1
42	14259	21.1
43	14259	21.1
44	14259	19.0

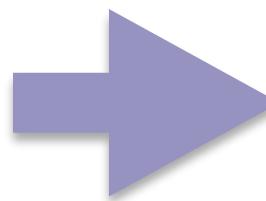


Why Visualization for Data Exploration?

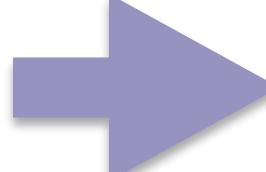
- Can't machines do (learn) that?
- Machine learning is great for
 - Well defined problems
 - Verifying Hypothesis
- ML not so great for
 - Finding the unknown
 - Fuzzy problems
 - Hypothesis generation



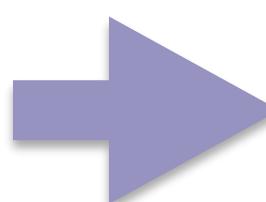
Algorithms



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
HSNE	non-linear	Graph, hierarchical	https://dx.doi.org/10.1038/s41467-017-01689-9
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

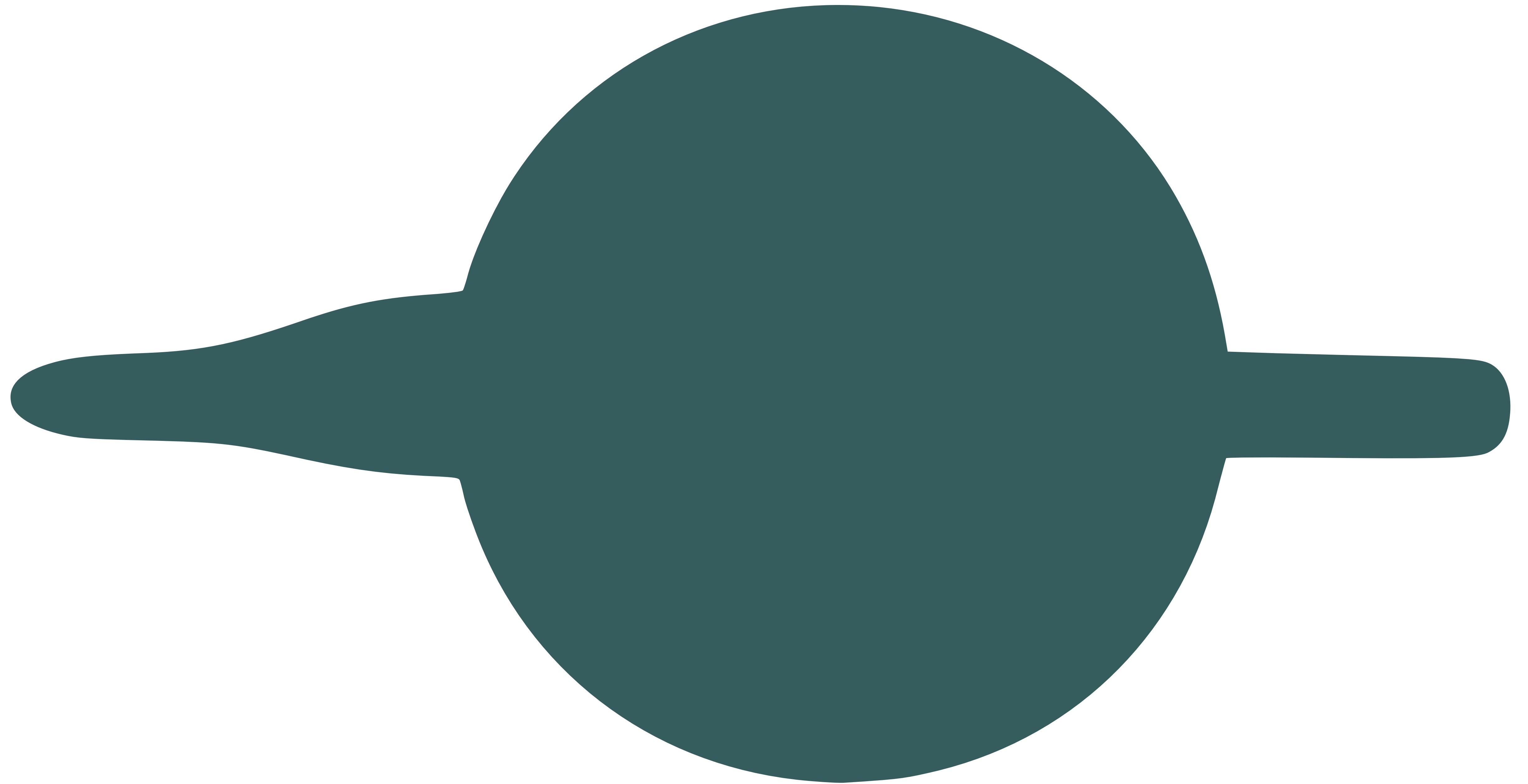


scvis	non-linear	Autoencoder	https://doi.org/10.1038/s41467-018-04368-5
VASC	non-linear	Autoencoder	https://doi.org/10.1016/j.gpb.2018.08.003

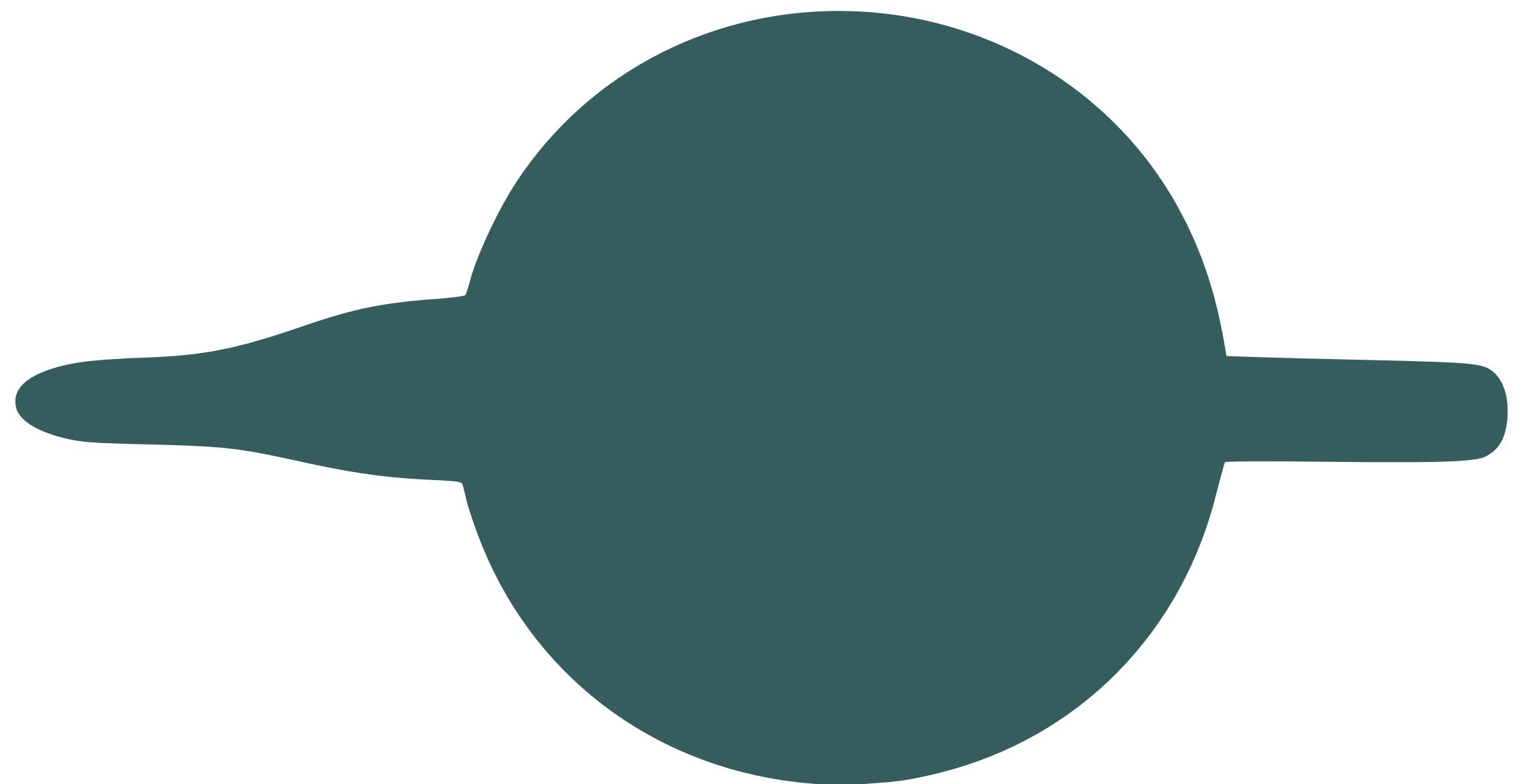
PCA

Principal Component Analysis







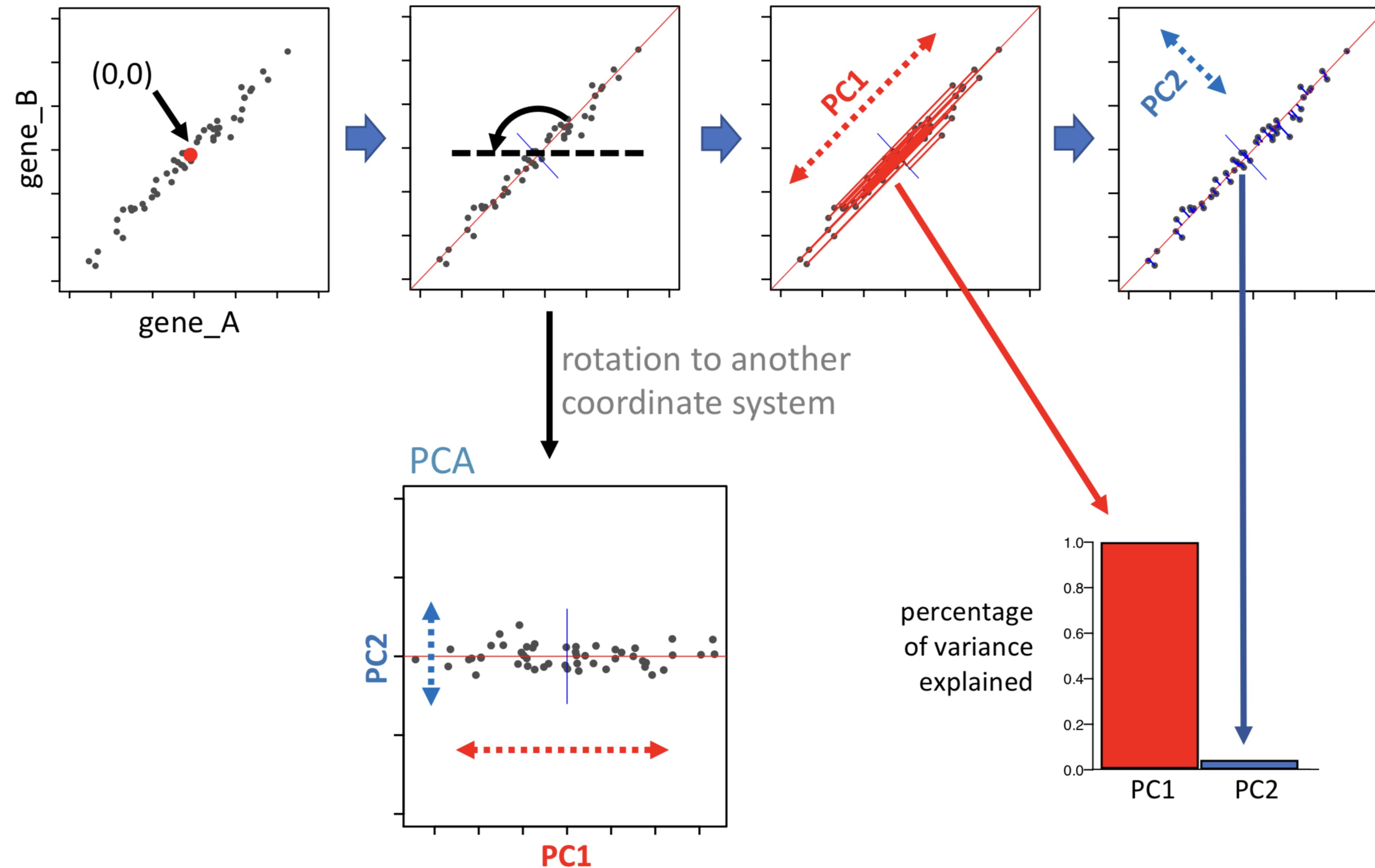


PCA - Intuition

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

PCA in Brief

→ <https://setosa.io/ev/principal-component-analysis/>

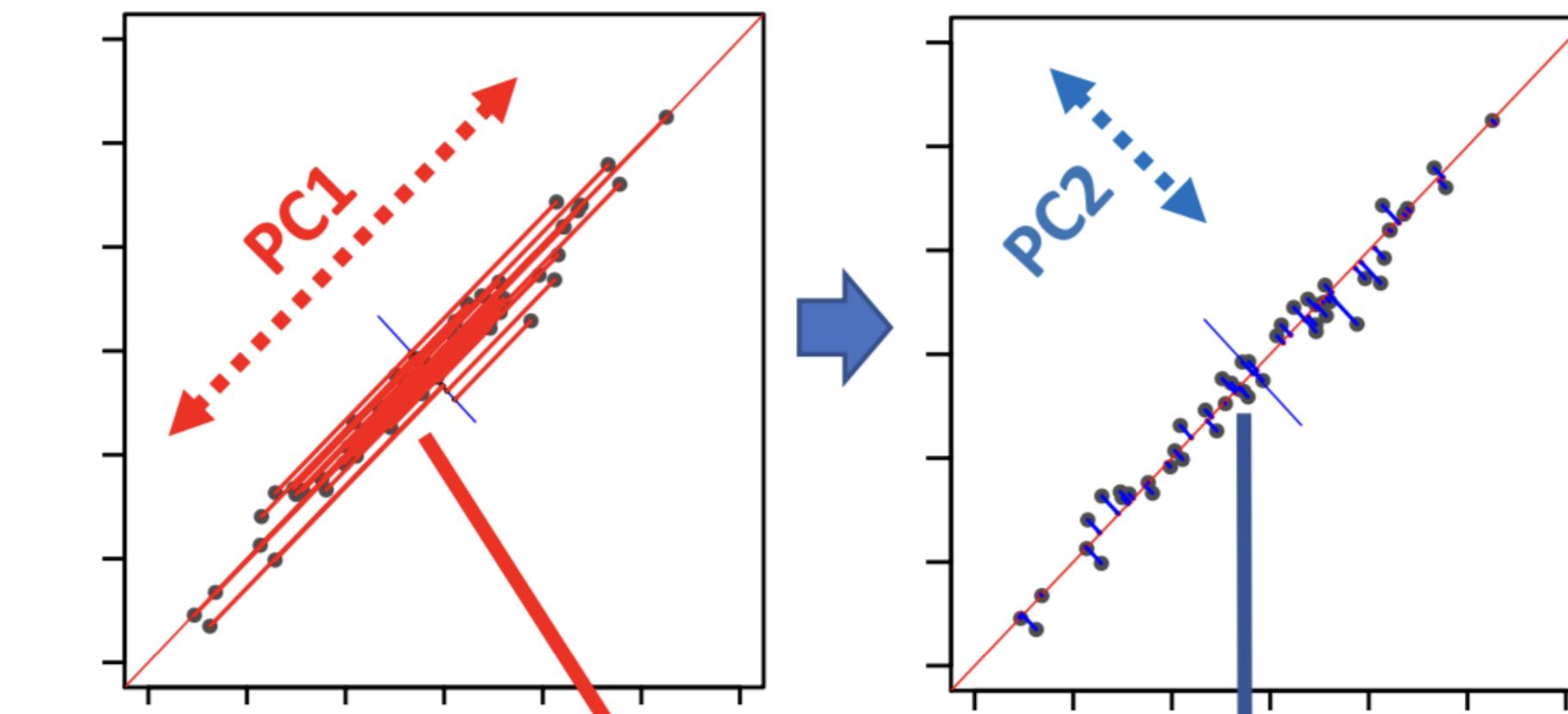


PCA in Brief

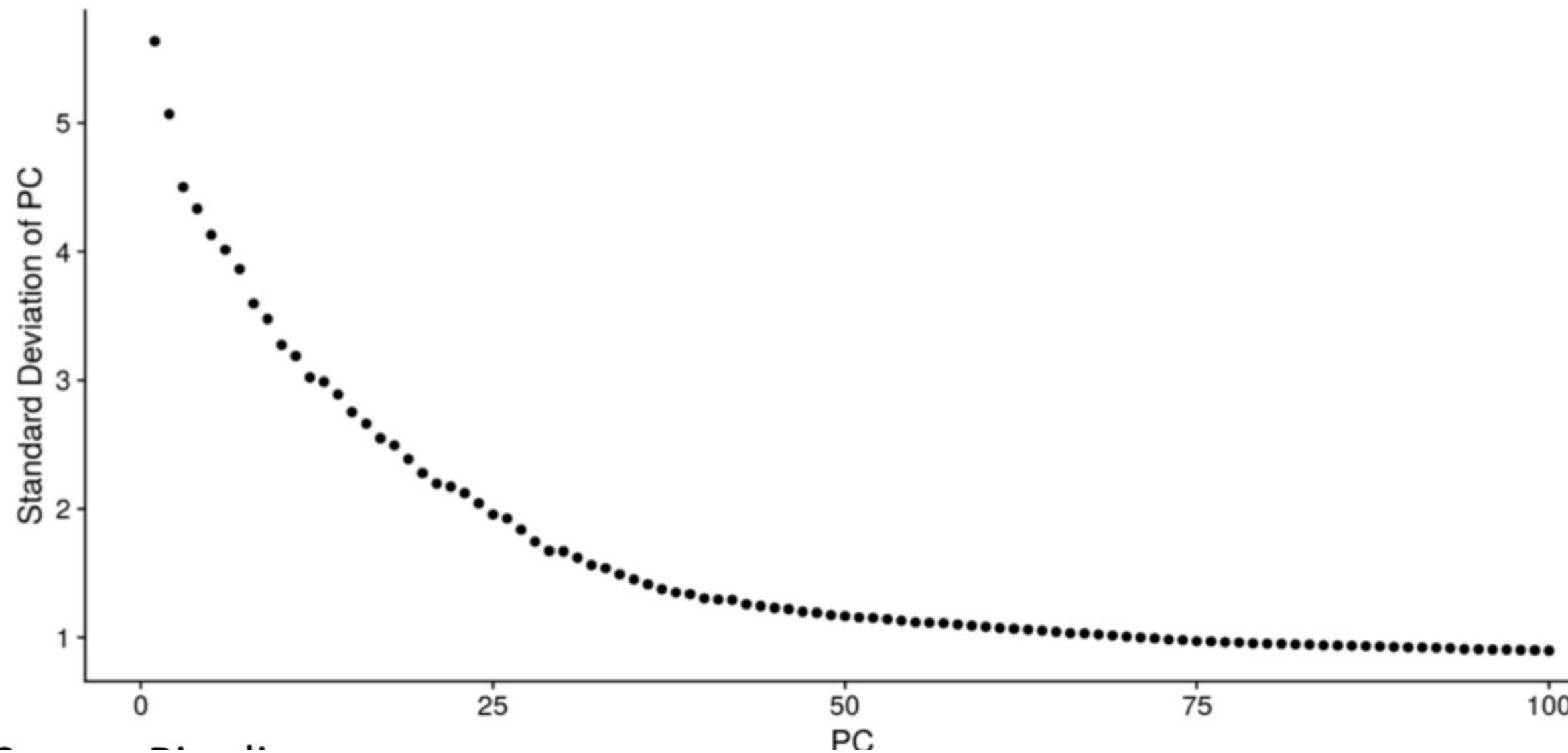
PC1 explains >98% of the variance

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

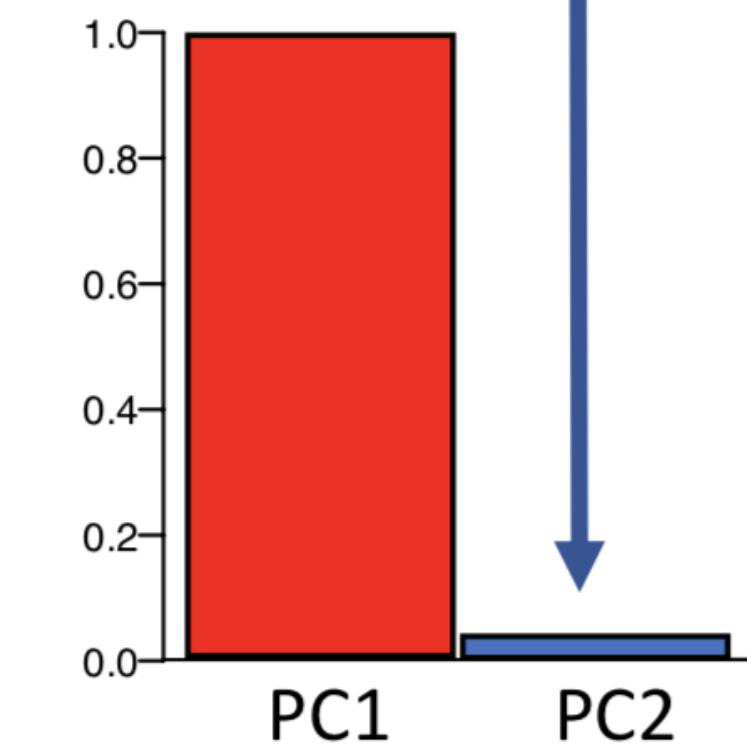
PC2 is nearly insignificant in this example
Could be disregarded



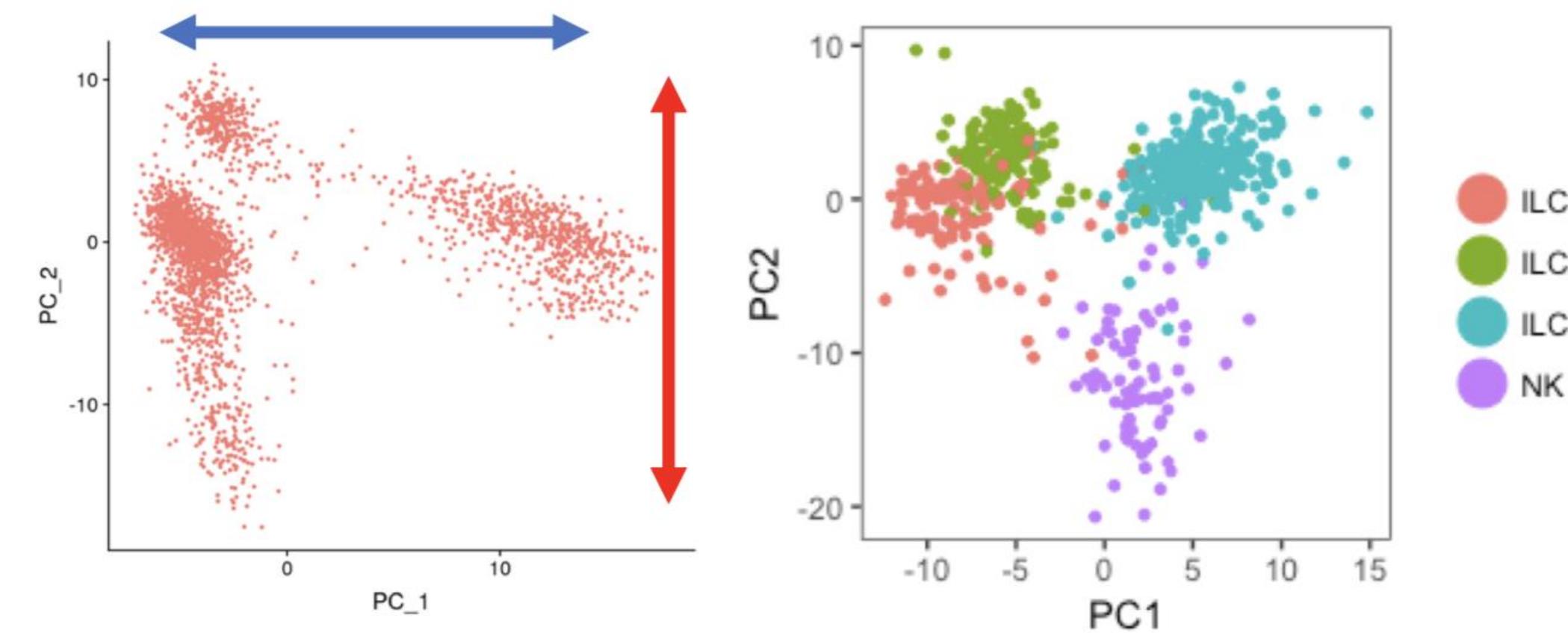
In real life ...



percentage
of variance
explained

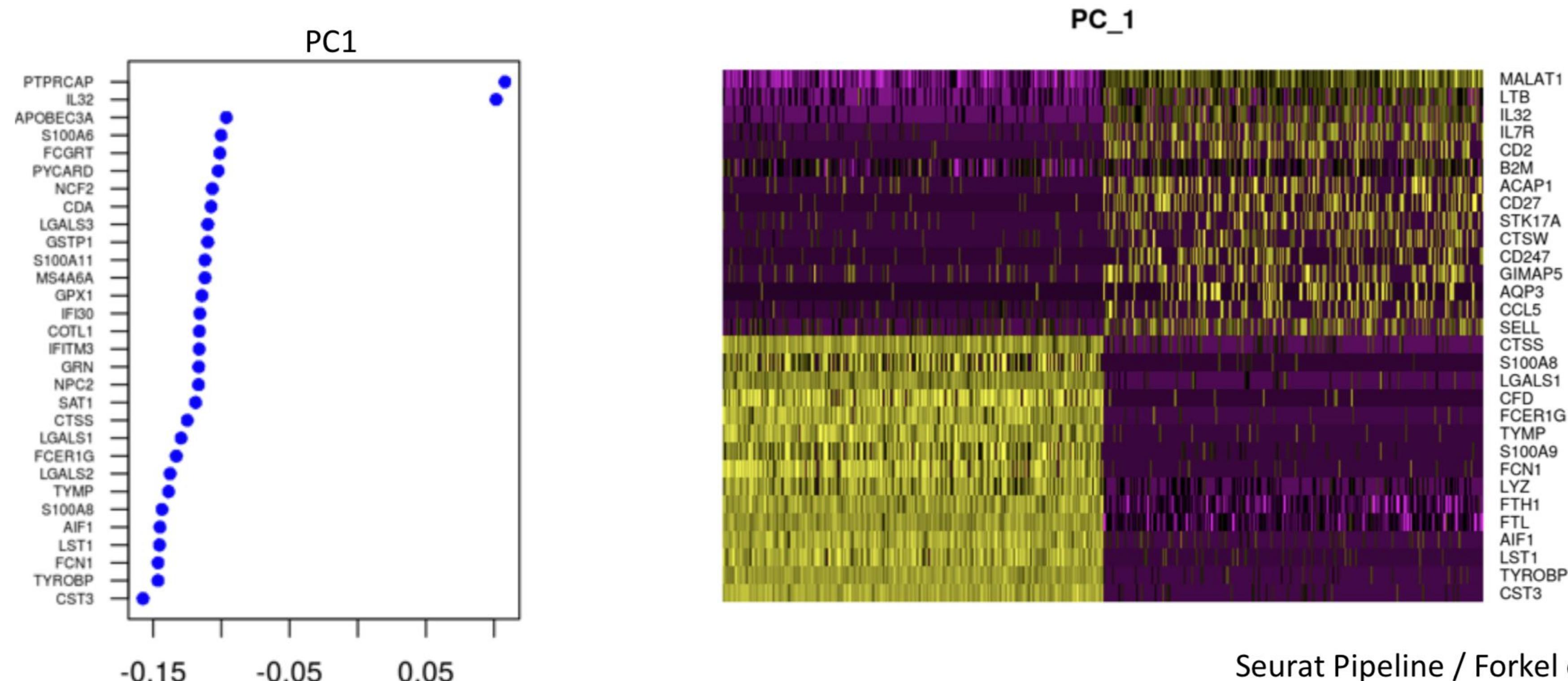


PCA in Brief



PC1 and PC2 are commonly correlated to sequencing depth and cell heterogeneity/complexity

(but not always ...)

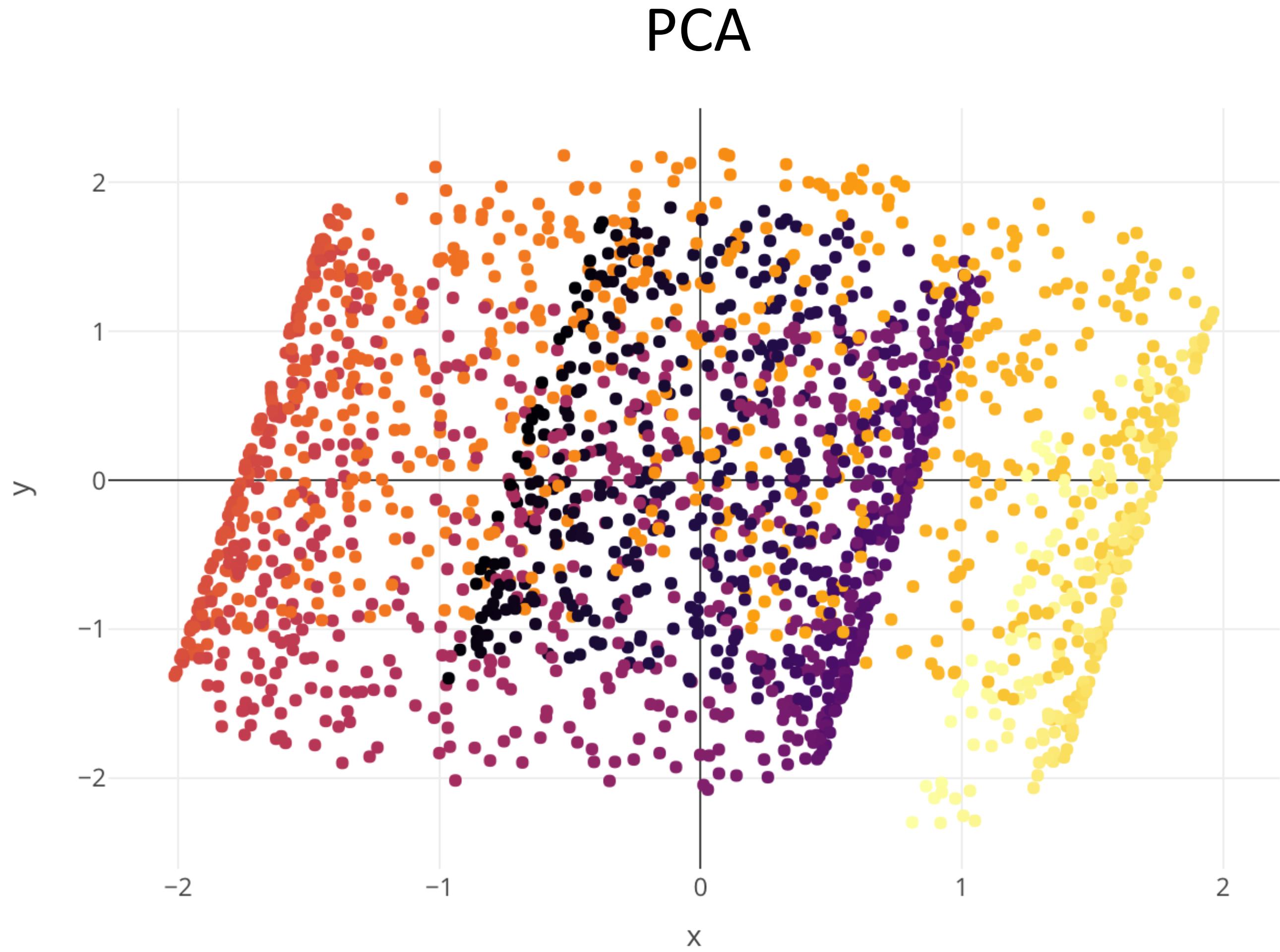
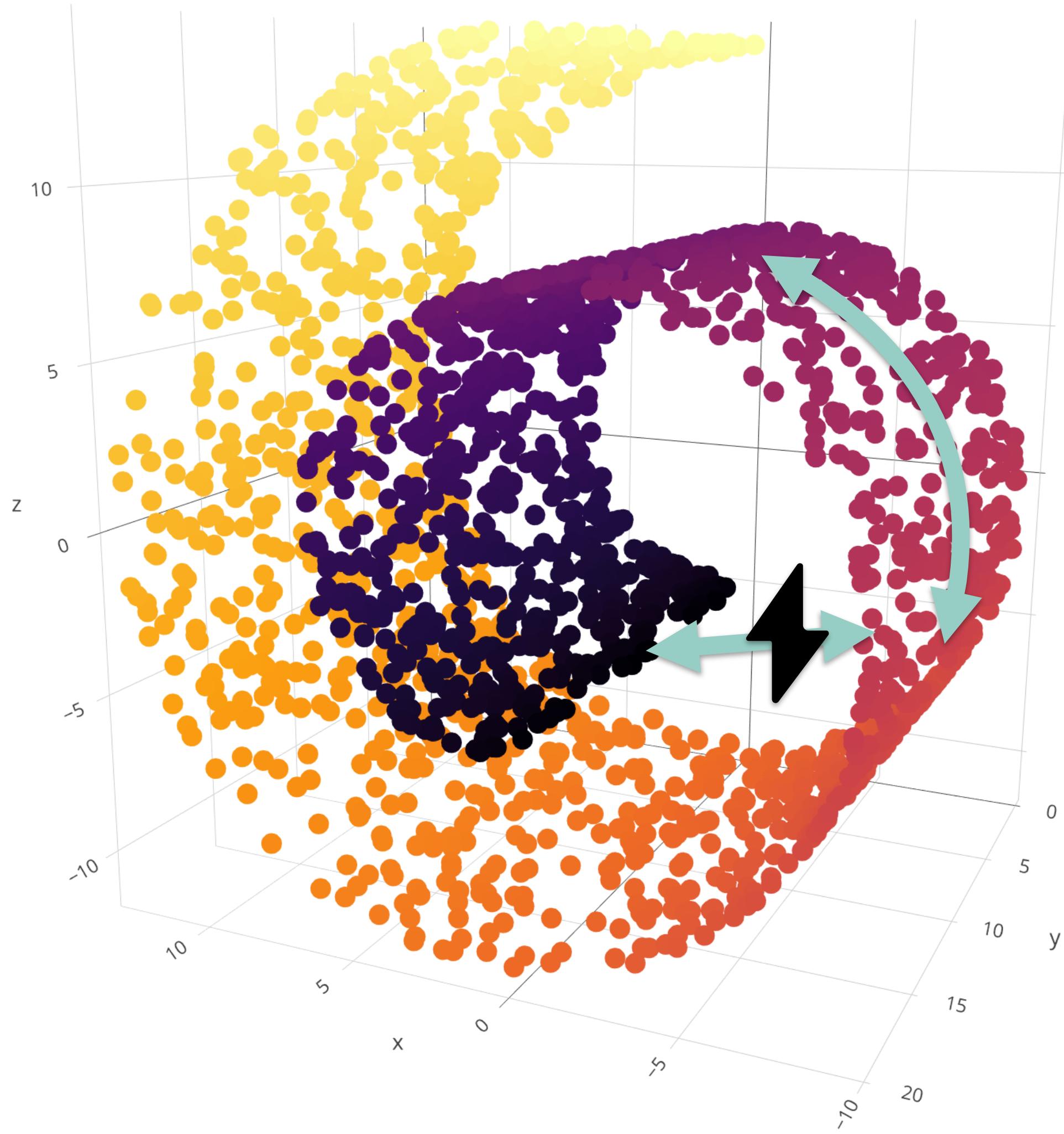


Summary: PCA

- 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

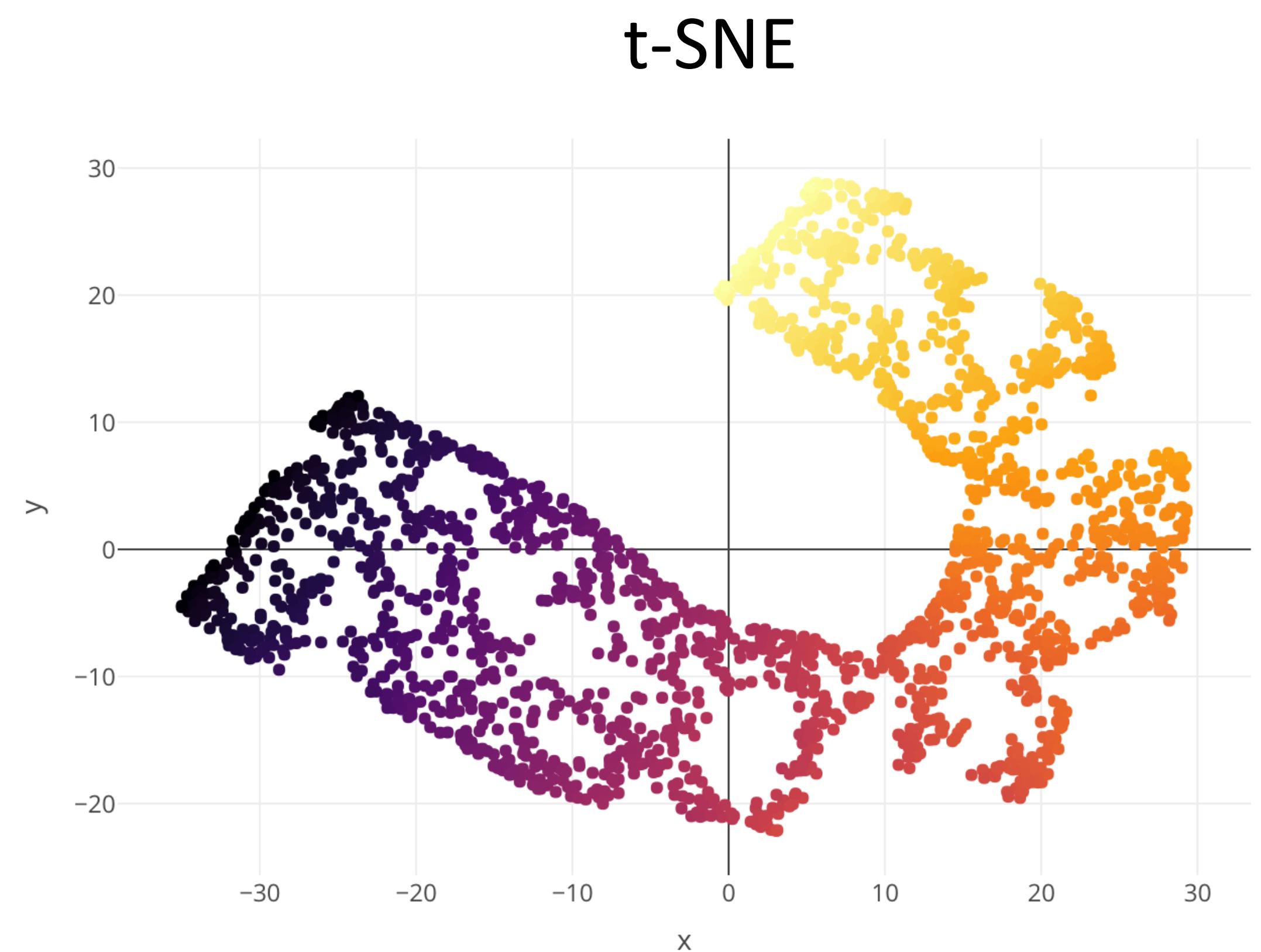
t-SNE
t-distributed Stochastic Neighborhood Embedding

Manifold Learning



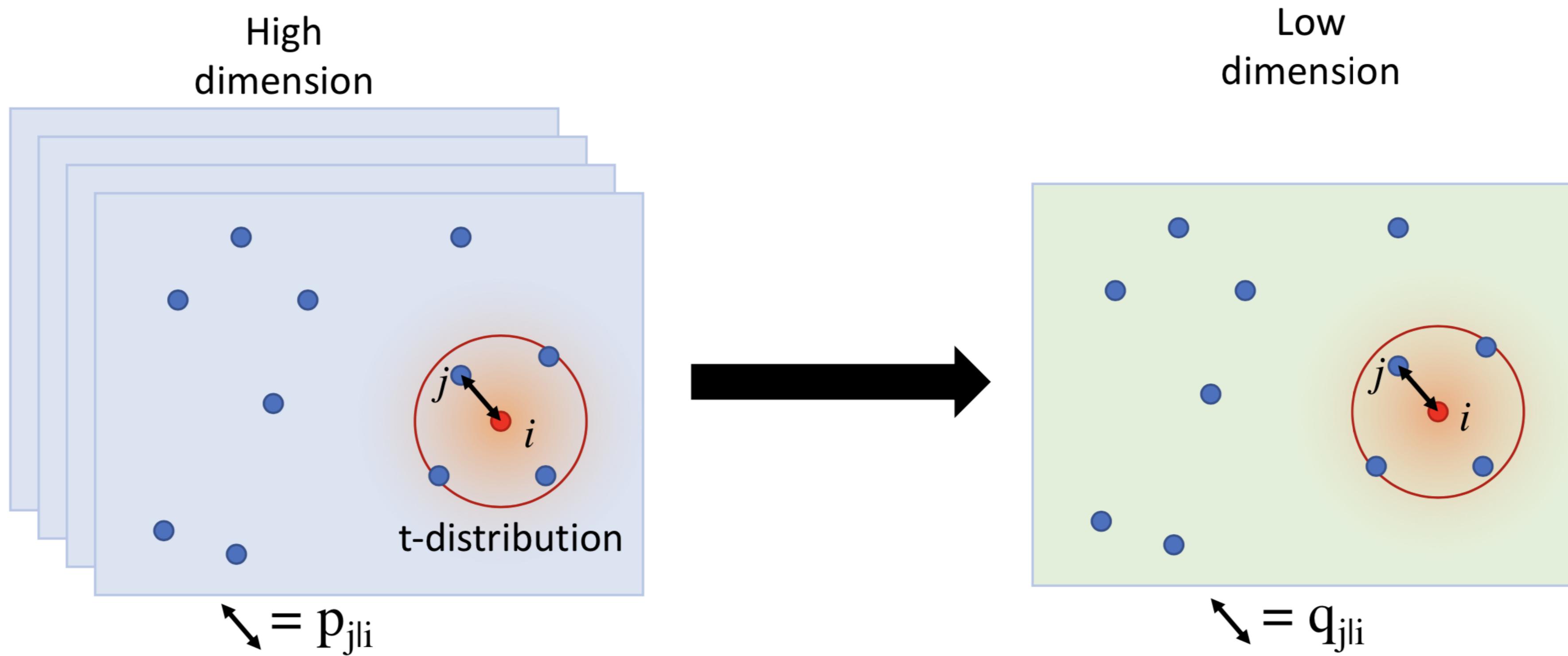
t-SNE Intuition

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

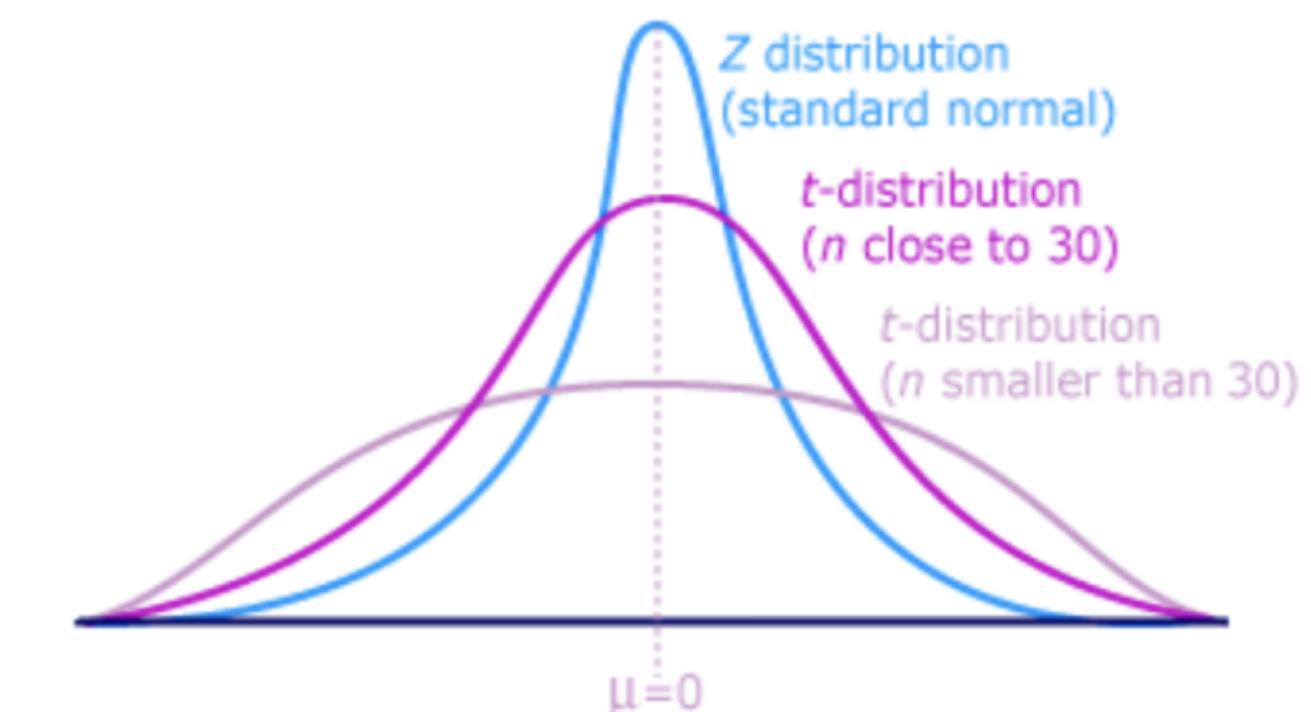


*actually also not great with Swiss Roll

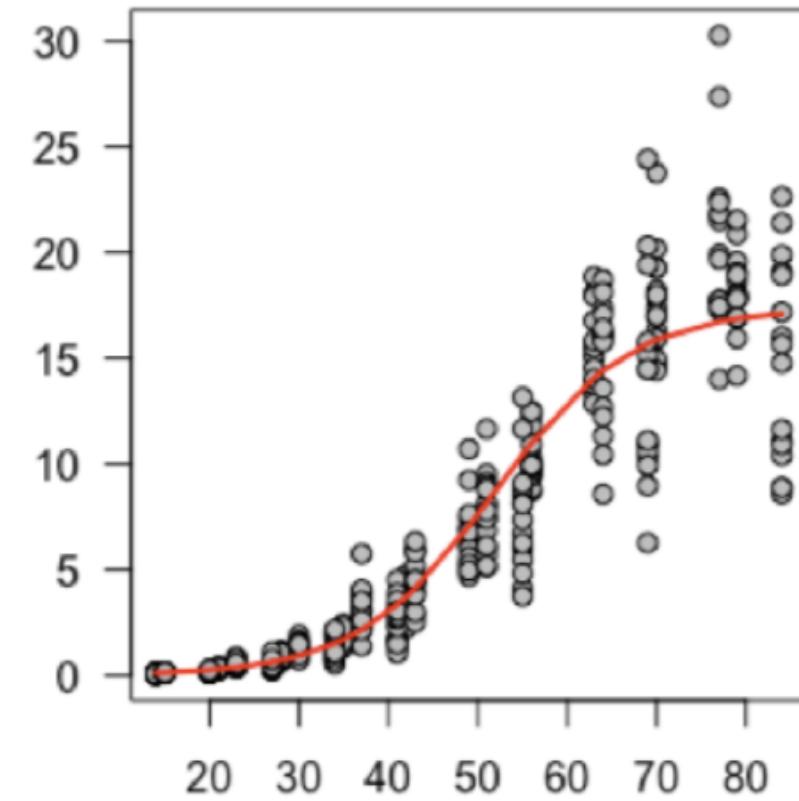
t-SNE in Brief



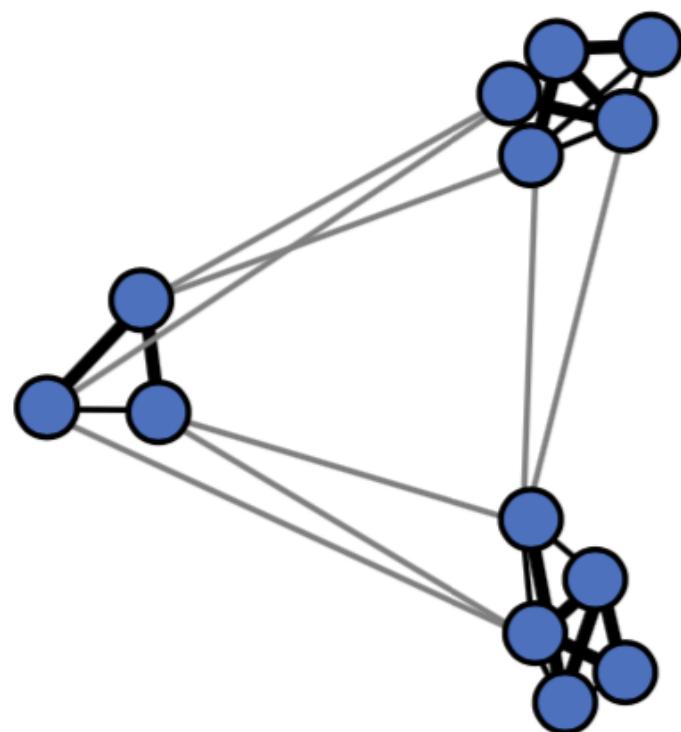
$p_{j|i}$ and $q_{j|i}$ measure the conditional probability that a point i would pick point j as its nearest neighbor, in high (p) and low (q) dimensional space respectively.



Sidestep: Graphs



This is a PLOT



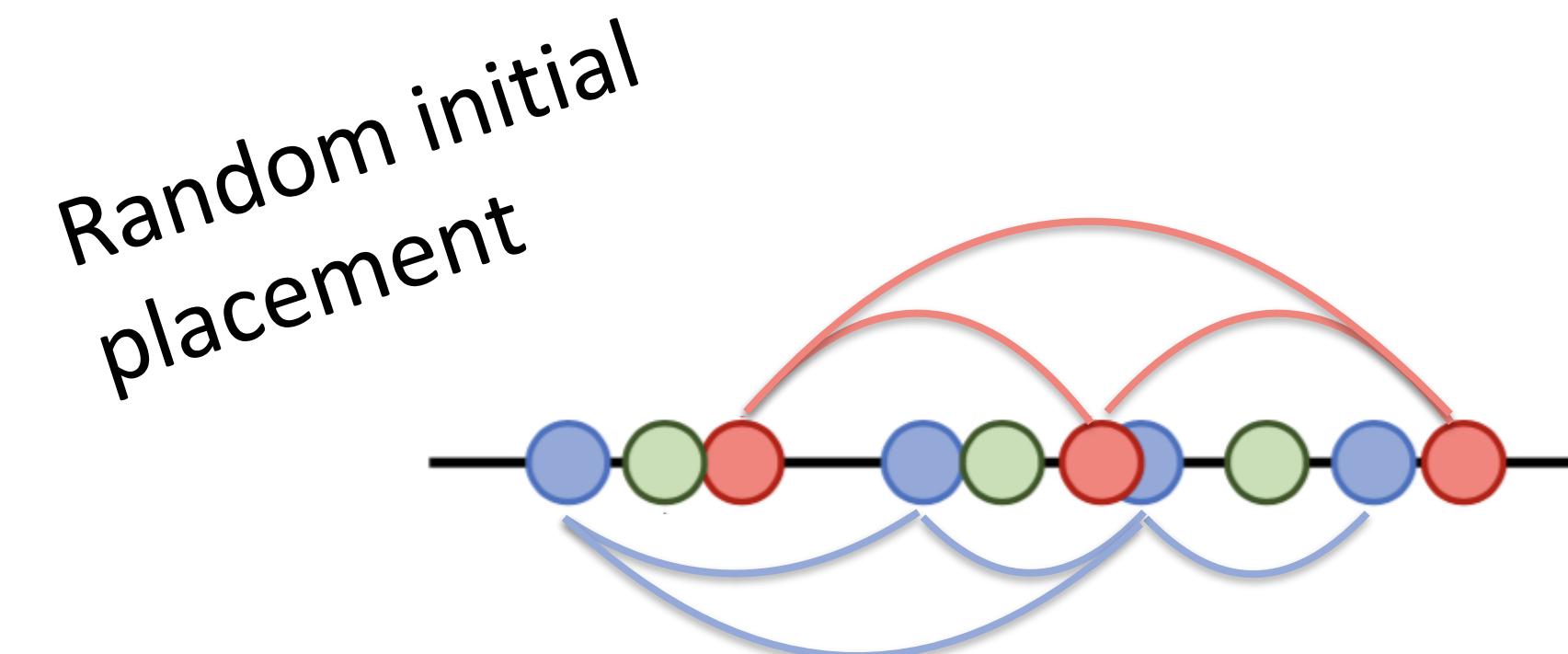
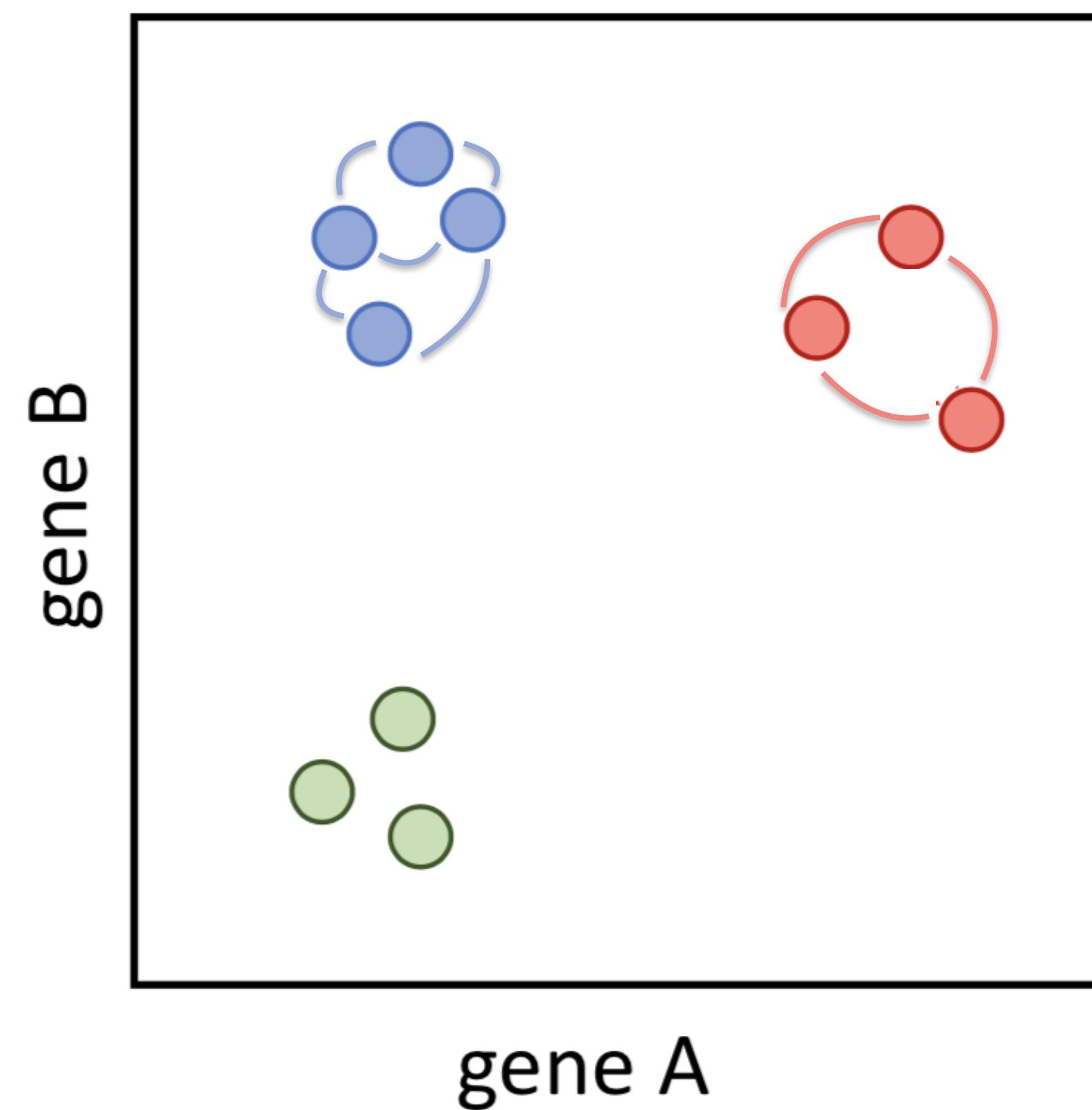
This is GRAPH
(a.k.a. network)

- 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.

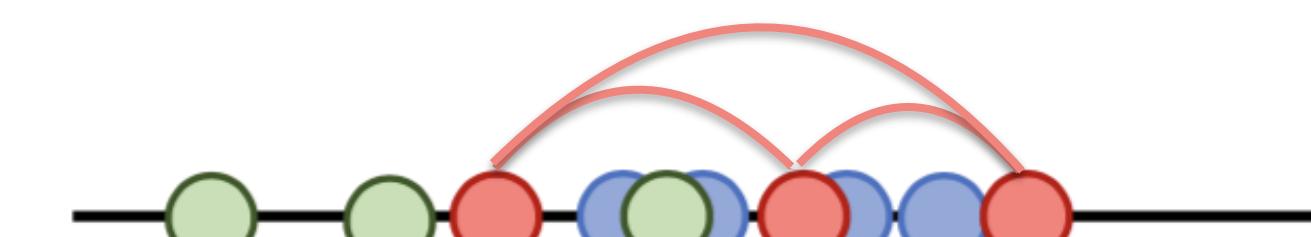
Graph-based dimensionality reduction algorithms can be divided into 2 main steps:

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

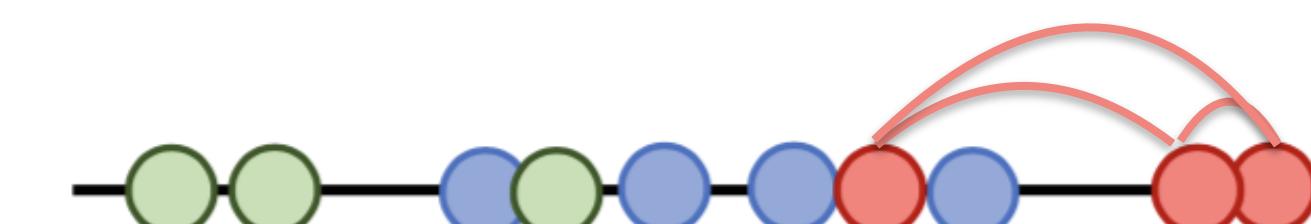
t-SNE in Brief II



iterations



iterations



iterations



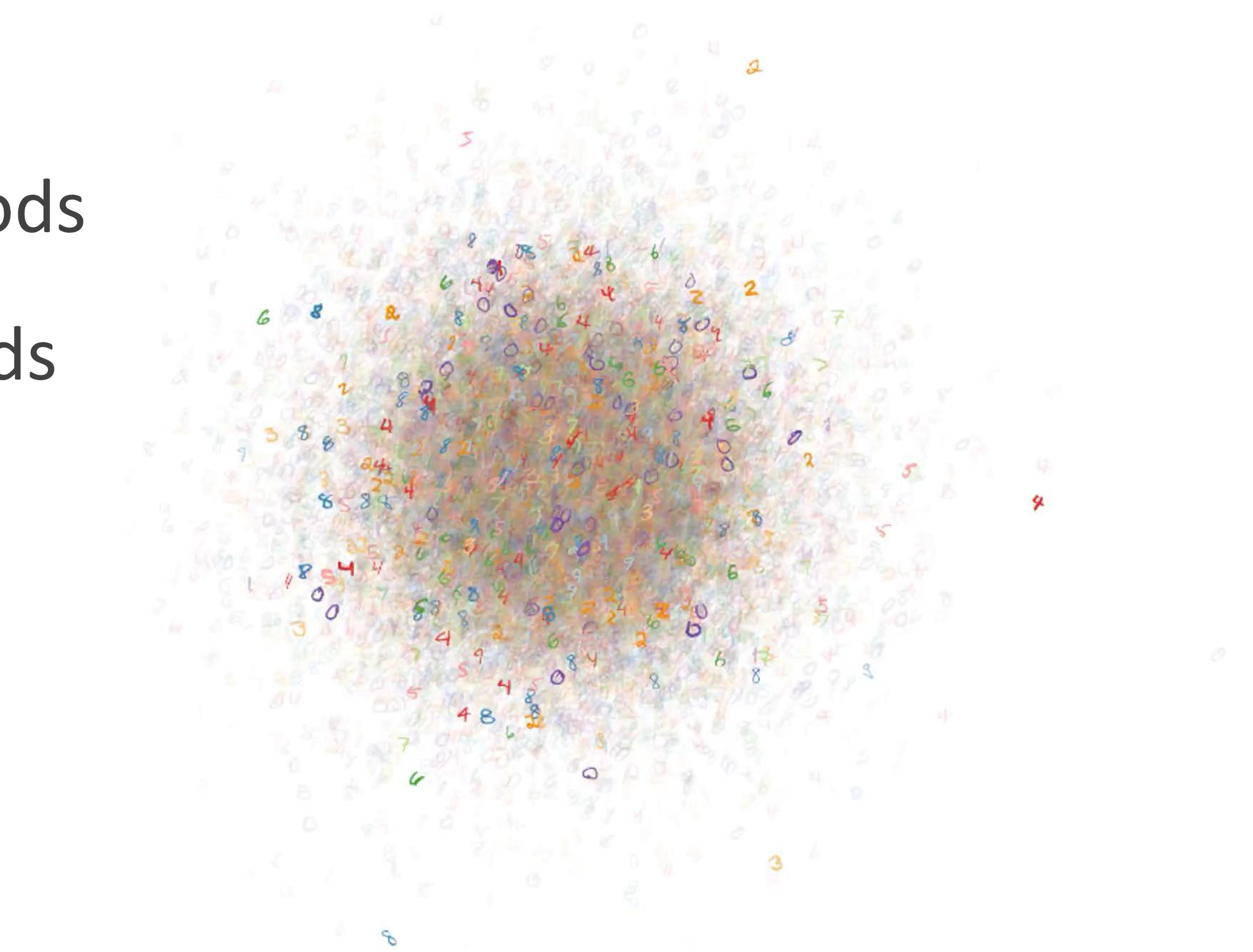
Lower KL divergence
(cost / error)

Iterative layout optimization

t-SNE in Brief III

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

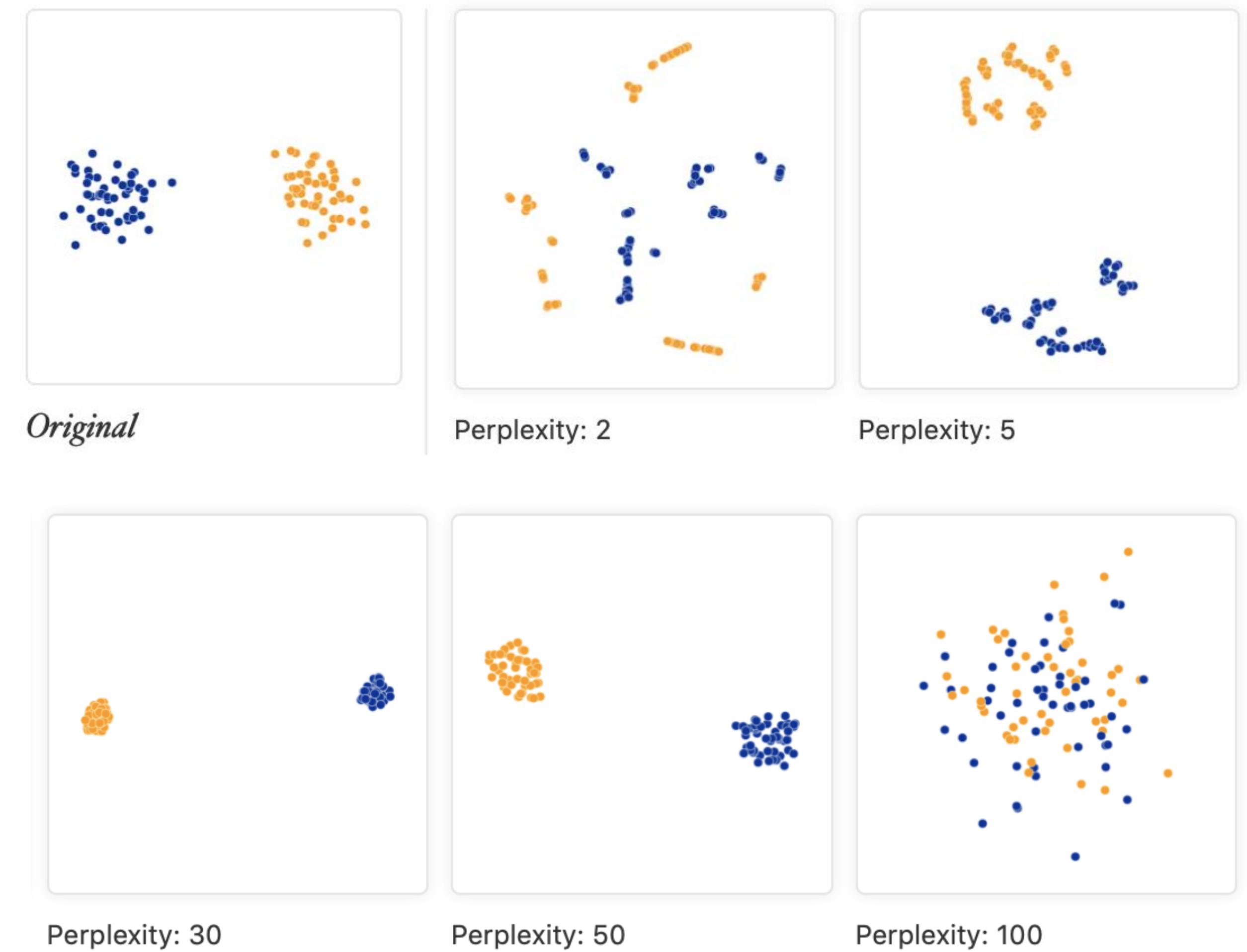
→ <https://distill.pub/2016/misread-tsne/>



t-SNE Parameters

- Perplexity
- Initialization
- Number of iterations
- Learning rate

...



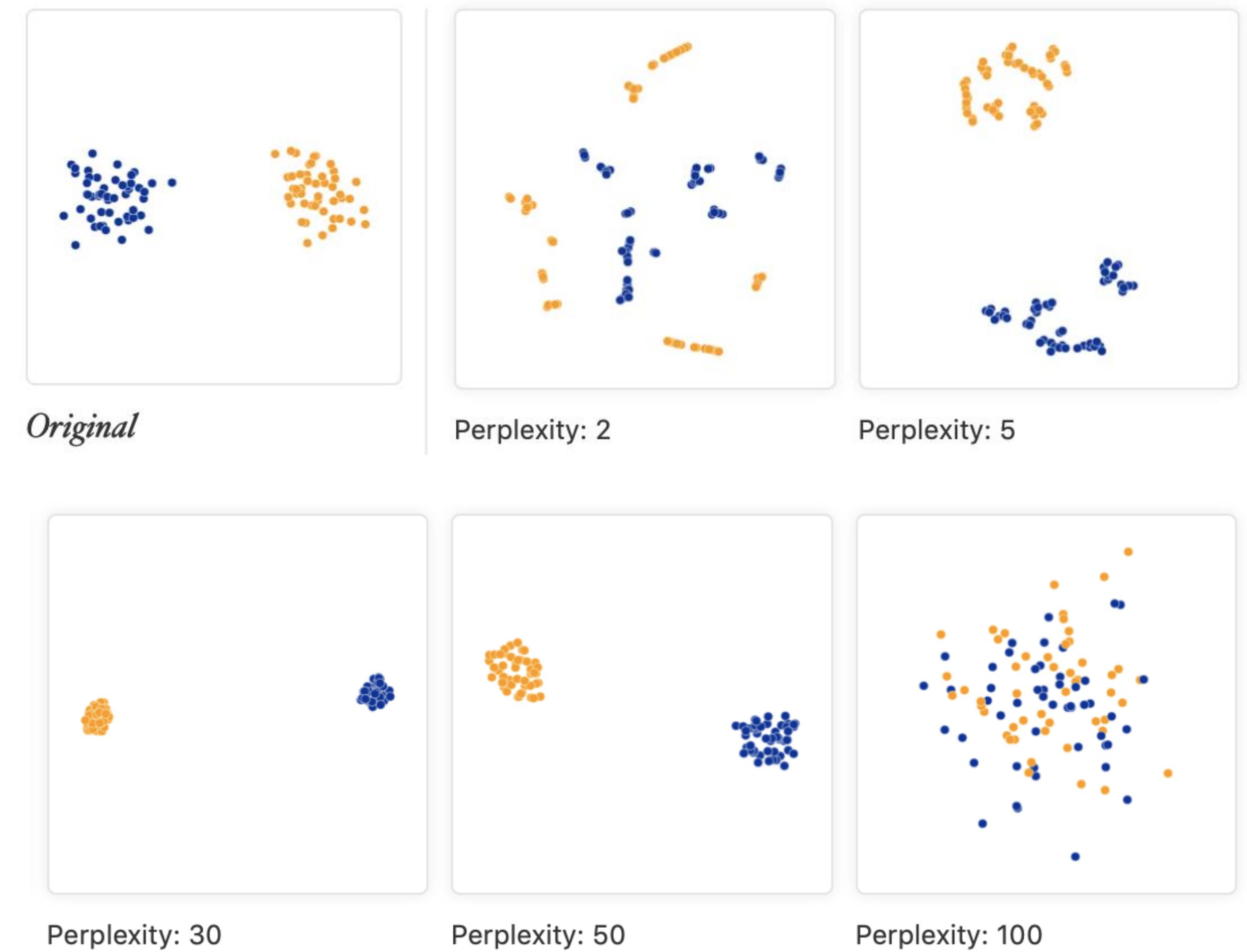
Kobak, D., Berens, P. **The art of using t-SNE for single-cell transcriptomics**. Nat Commun 10, 5416 (2019).
<https://doi.org/10.1038/s41467-019-13056-x>

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

t-SNE Parameters

- Perplexity
- Initialization
- Number of iterations
- Learning rate

...

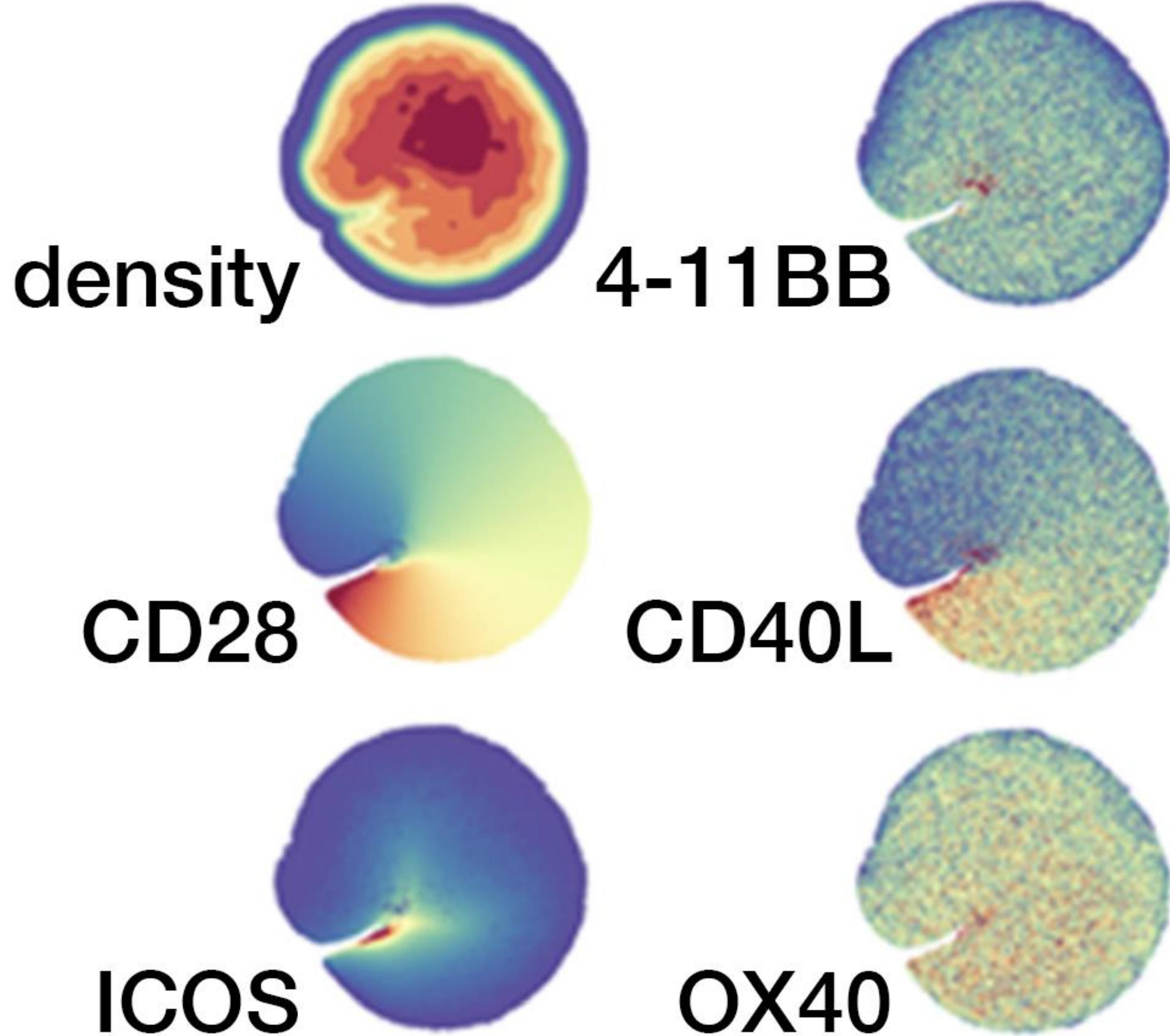


Kobak, D., Berens, P. **The art of using t-SNE for single-cell transcriptomics**. Nat Commun 10, 5416 (2019).
<https://doi.org/10.1038/s41467-019-13056-x>

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

t-SNE Parameters II

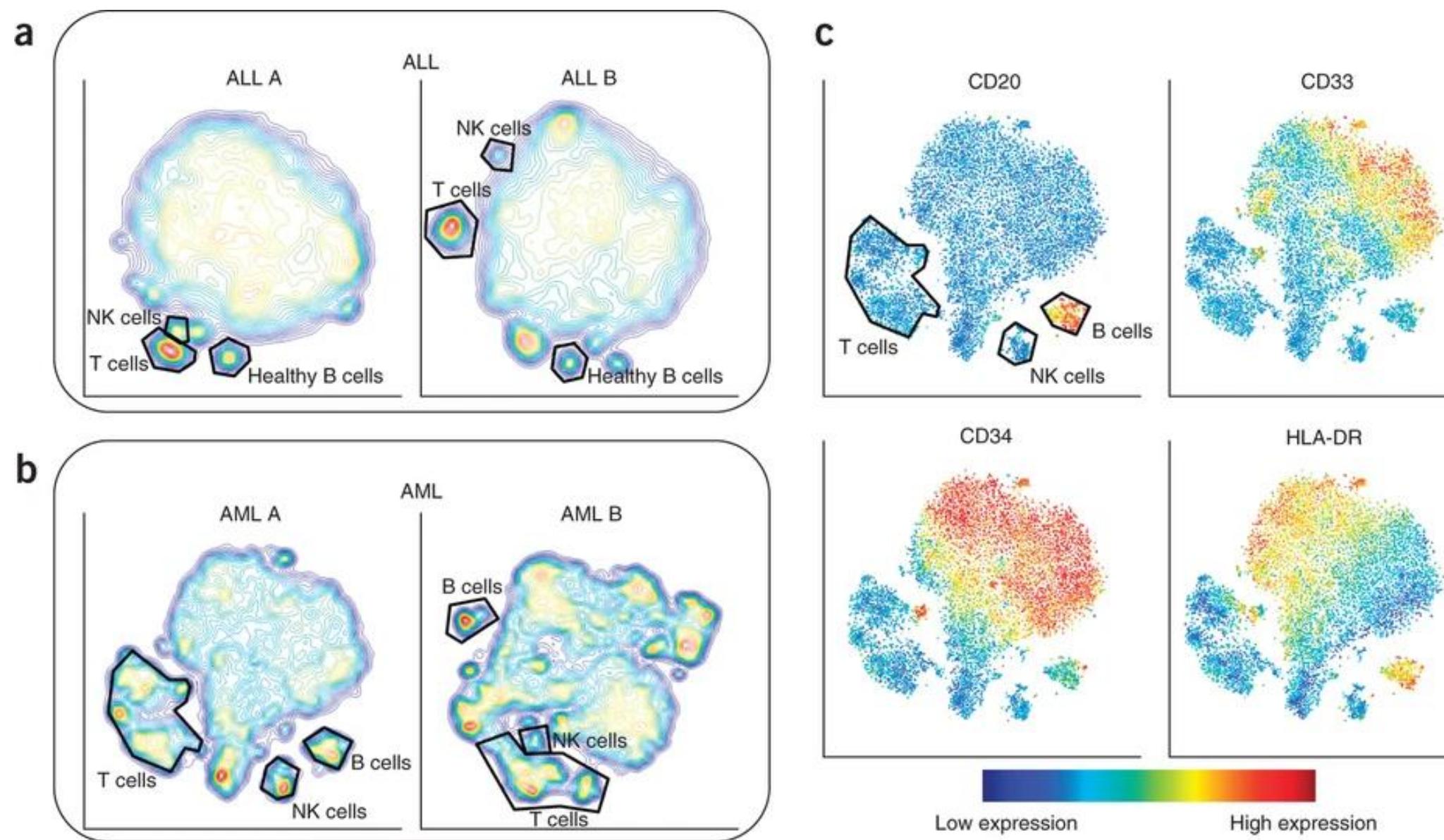
- Perplexity
- Initialization
- Number of iterations
- Learning rate
- ...



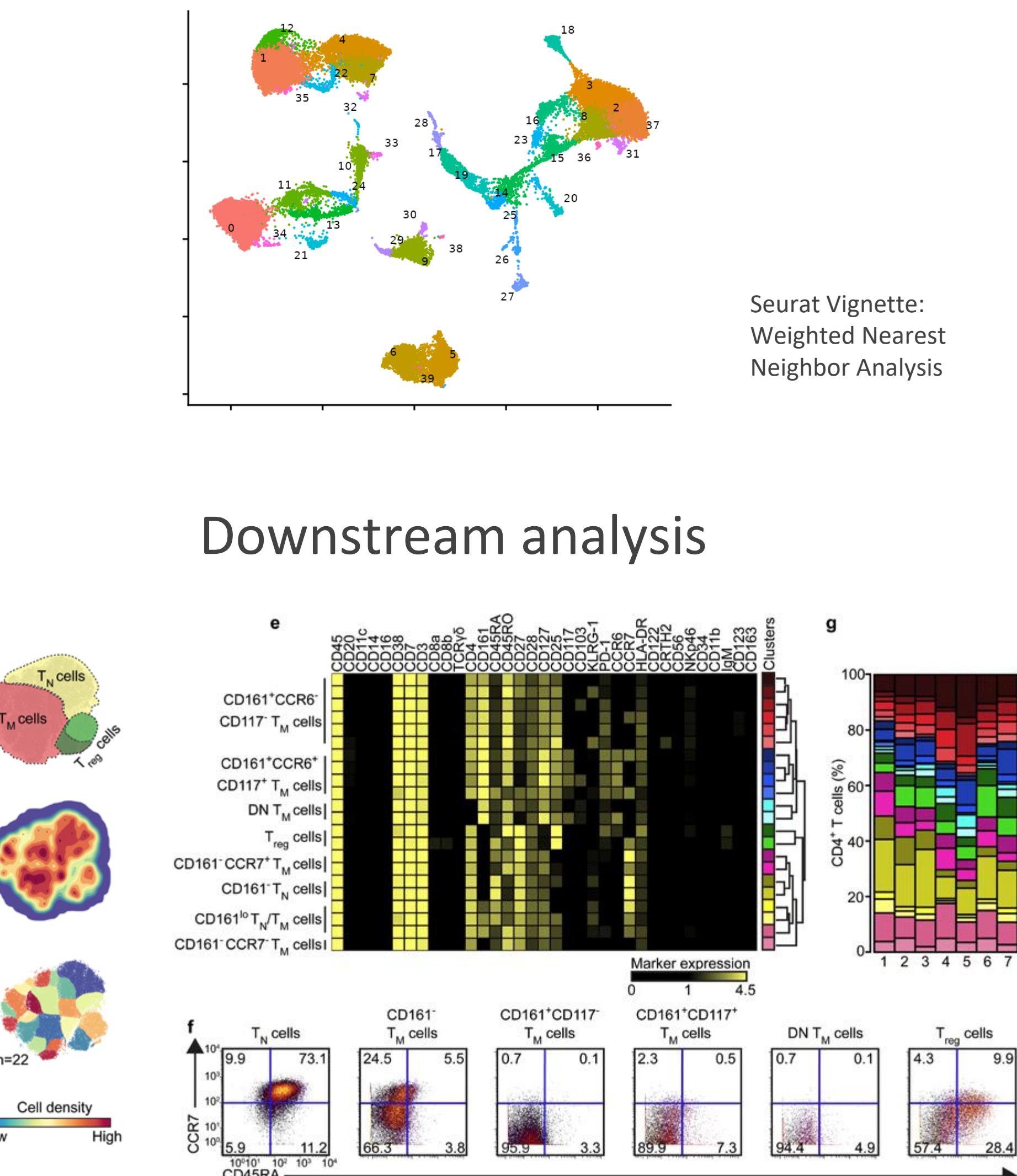
Embedding exploration

Clustering

Re-color and density



Amir, E.A., Davis, K., Tadmor, M. et al. **viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia**. Nat Biotechnol 31, 545–552 (2013).
<https://doi.org/10.1038/nbt.2594>



Li, N., van Unen, V., Abdelaal, T. et al. **Memory CD4+ T cells are generated in the human fetal intestine**. Nat Immunol 20, 301–312 (2019). <https://doi.org/10.1038/s41590-018-0294-9>

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.
→ There are approaches for overcoming this, e.g. [Kobak \(2019\)](#)

t-SNE Implementations

- Many implementations available

→ computation speed-up



pip install nptsne

<https://pypi.org/project/nptsne/>

<https://github.com/biovault/>

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](#)



Seurat -> FIt-SNE

<https://github.com/KlugerLab/FIt-SNE>

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

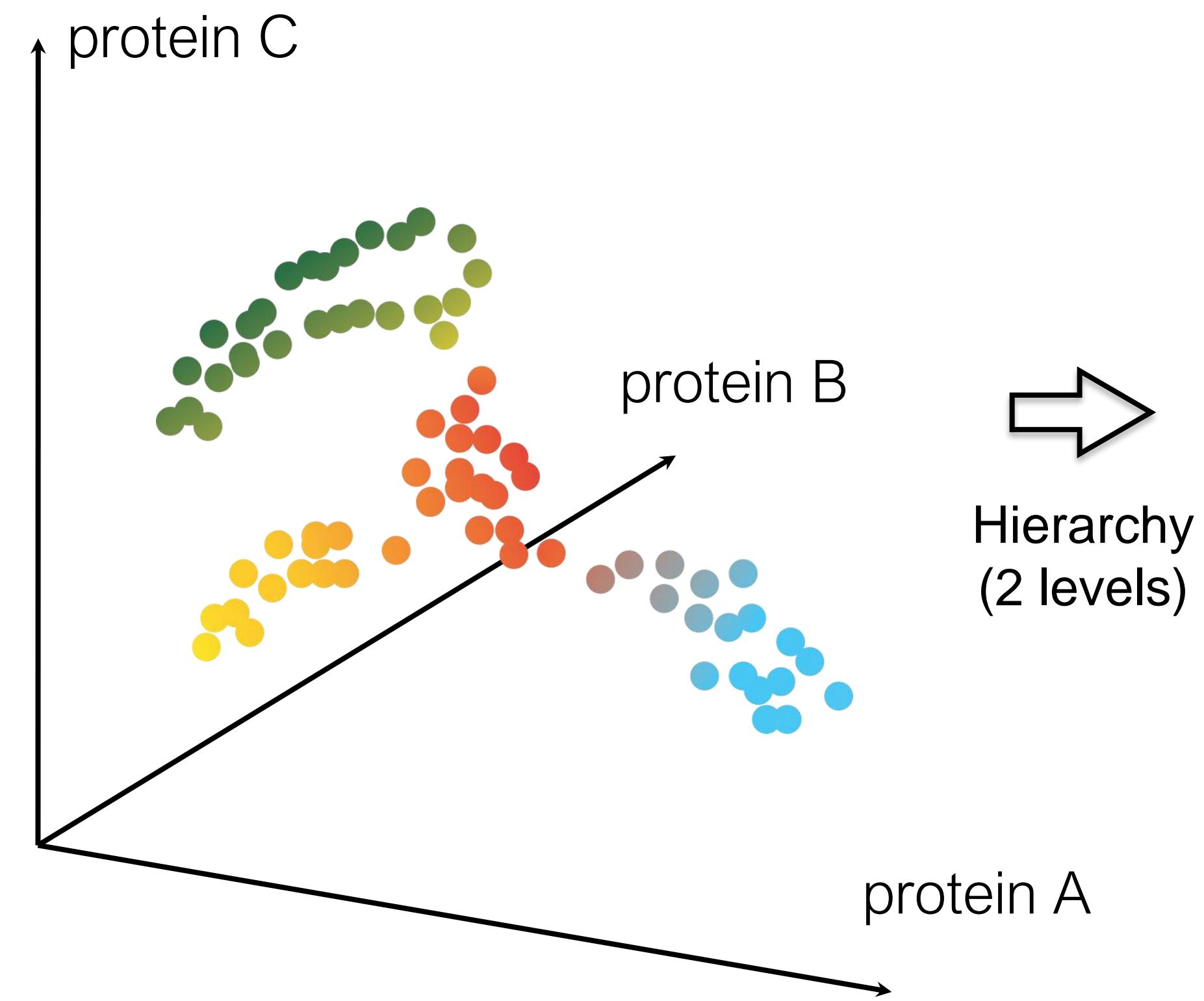
Summary: t-SNE

- NON-LINEAR method of dimensionality reduction
- Mostly used to embed to 2 dimensions
- Can be run from the top PCs (e.g.: PC1 to PC10)

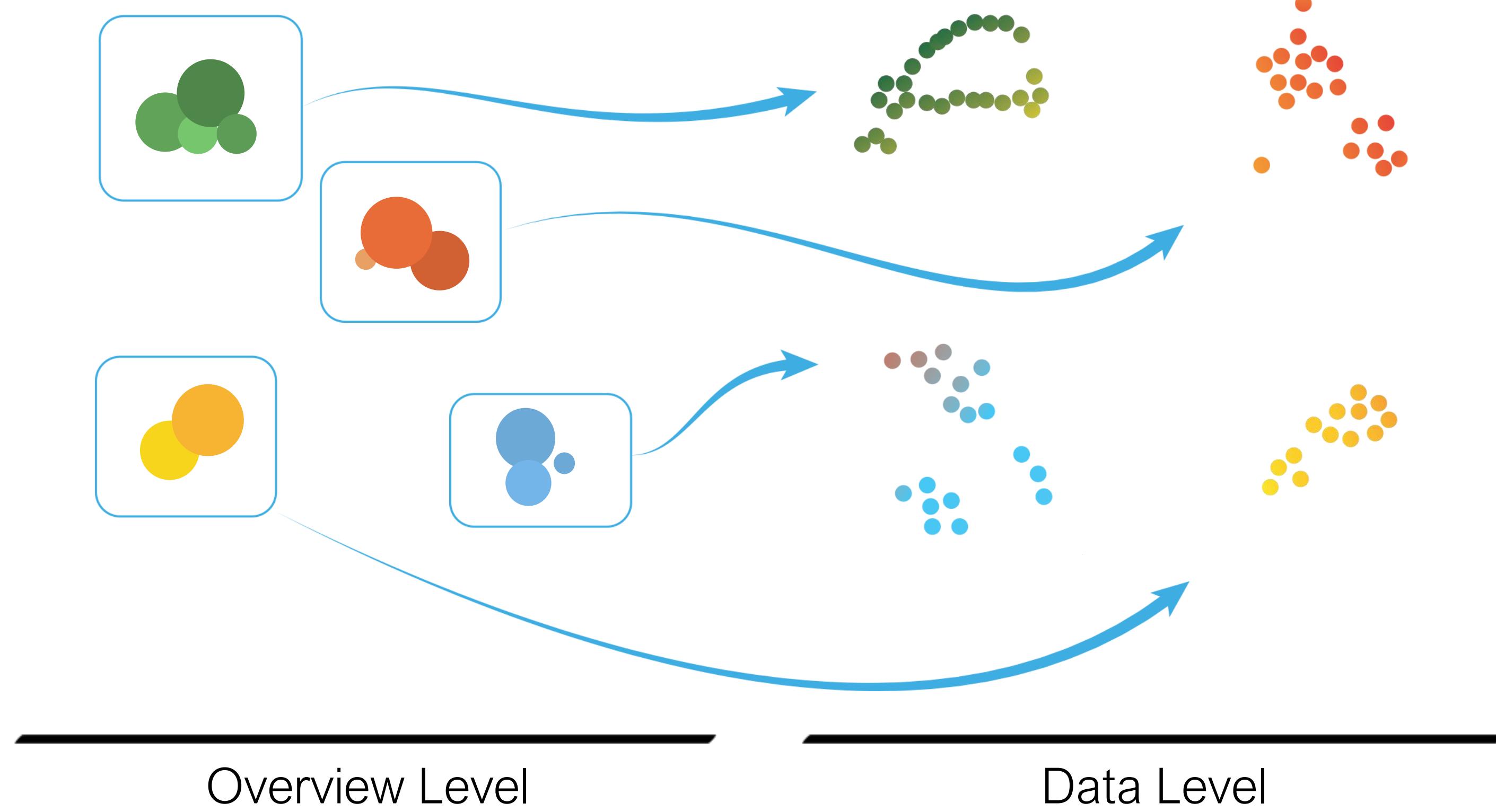
Problems:

- 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, without specific initializations)

Hierarchical SNE



Hierarchy
(2 levels)

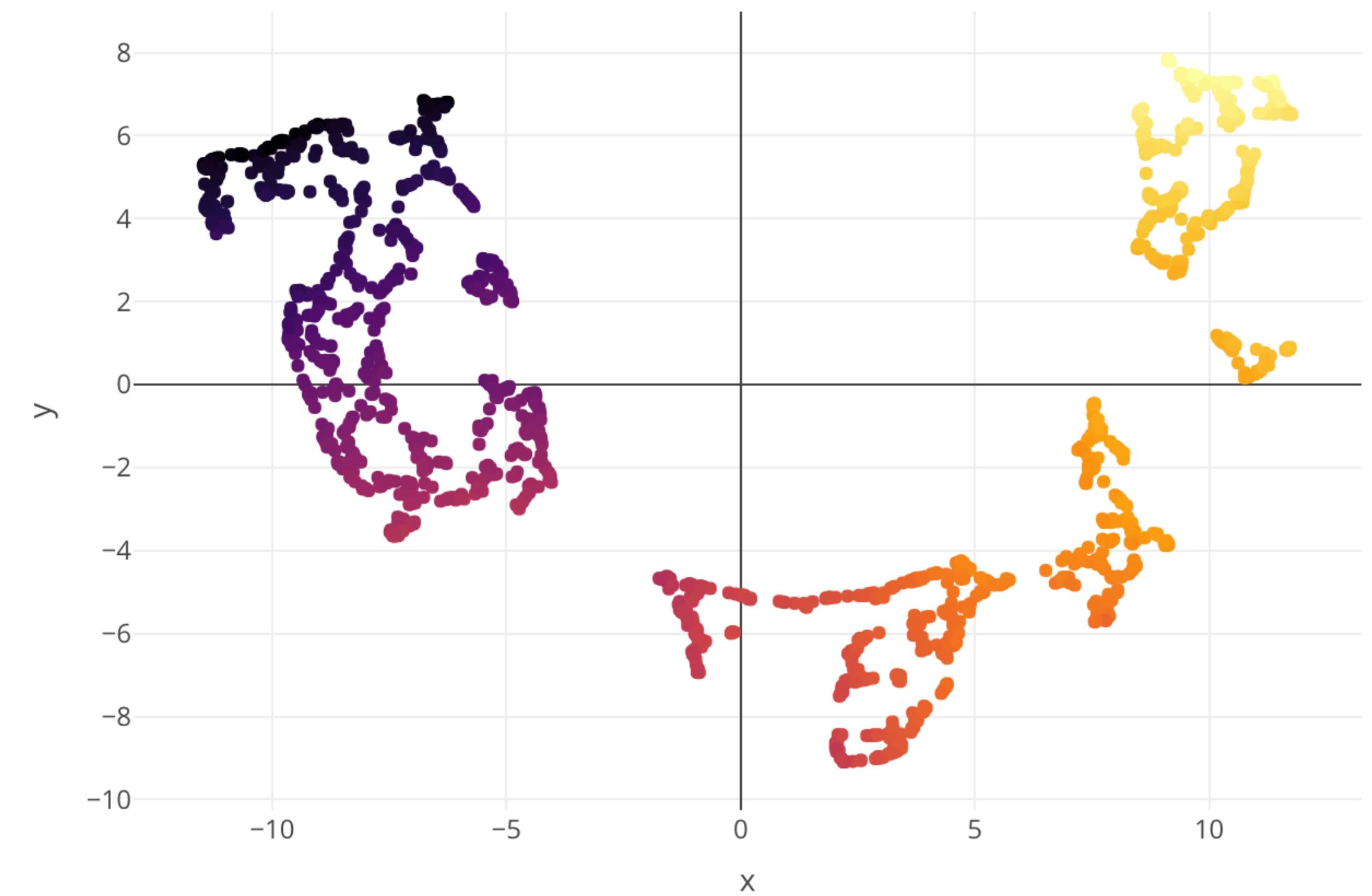


UMAP

Uniform Manifold Approximation and Projection

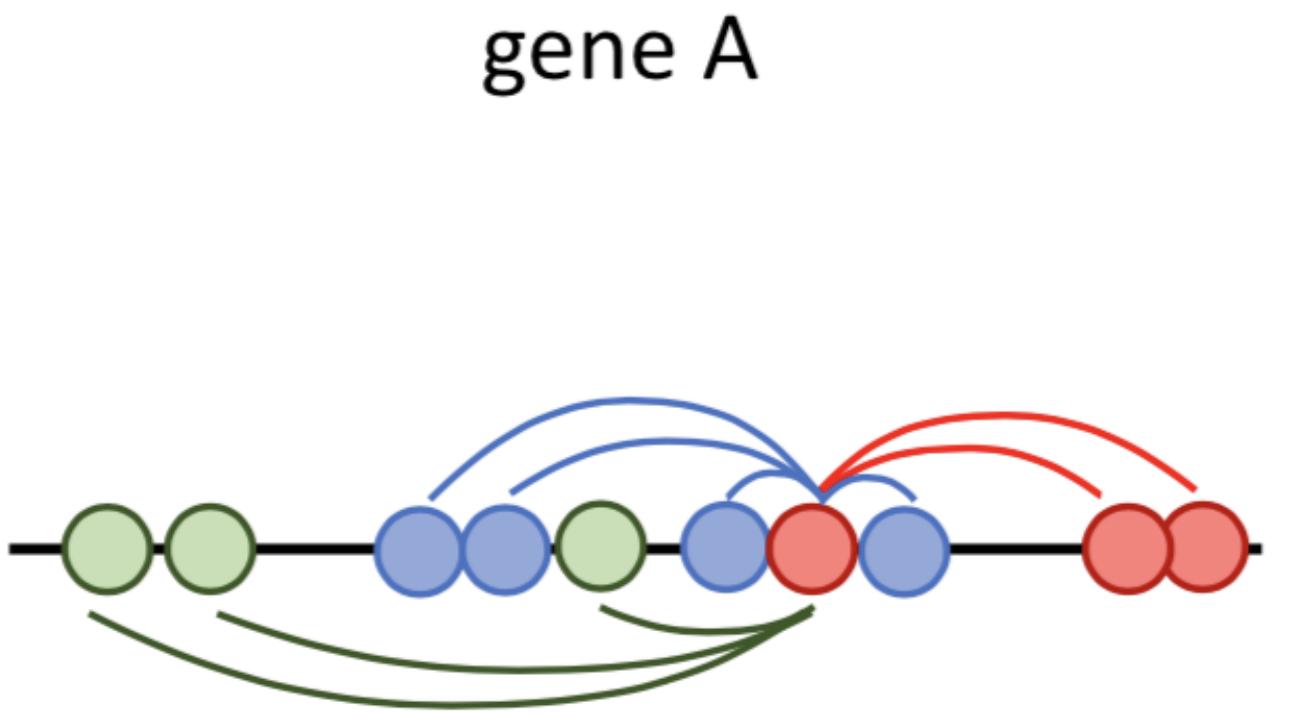
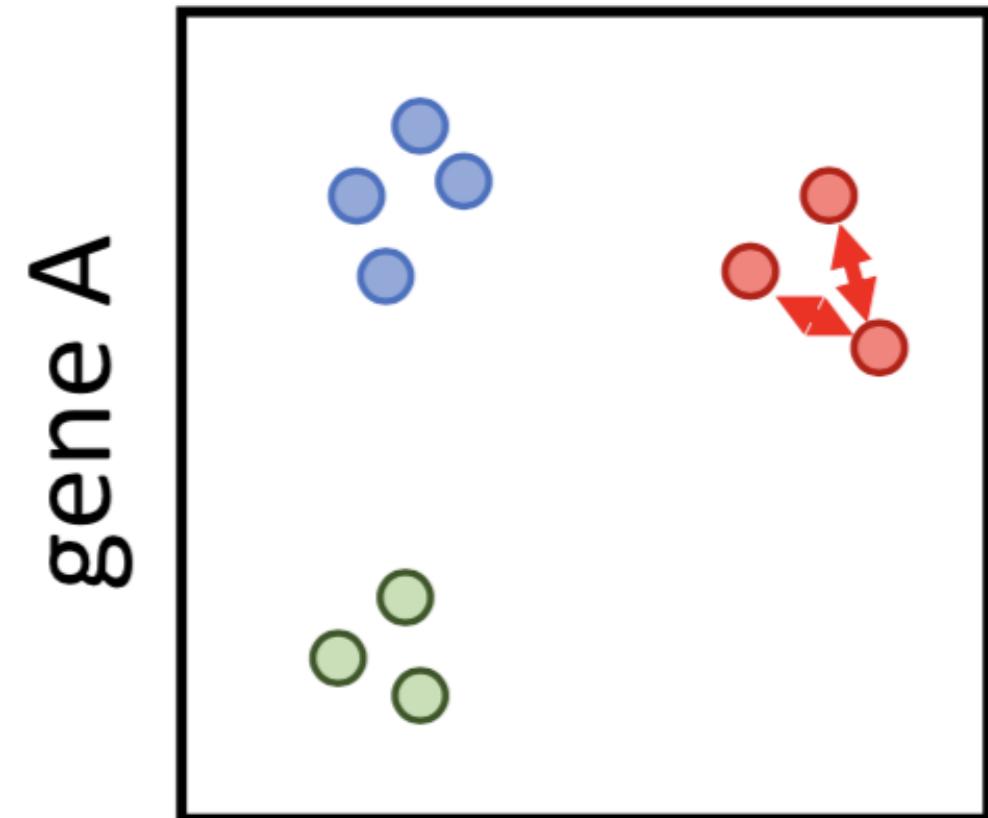
UMAP Intuition

- Similar principle as t-SNE
- Initializes with a non-heuristic “guess”
 - Same result every time
- Resolves global structure somewhat better (due to initialization)

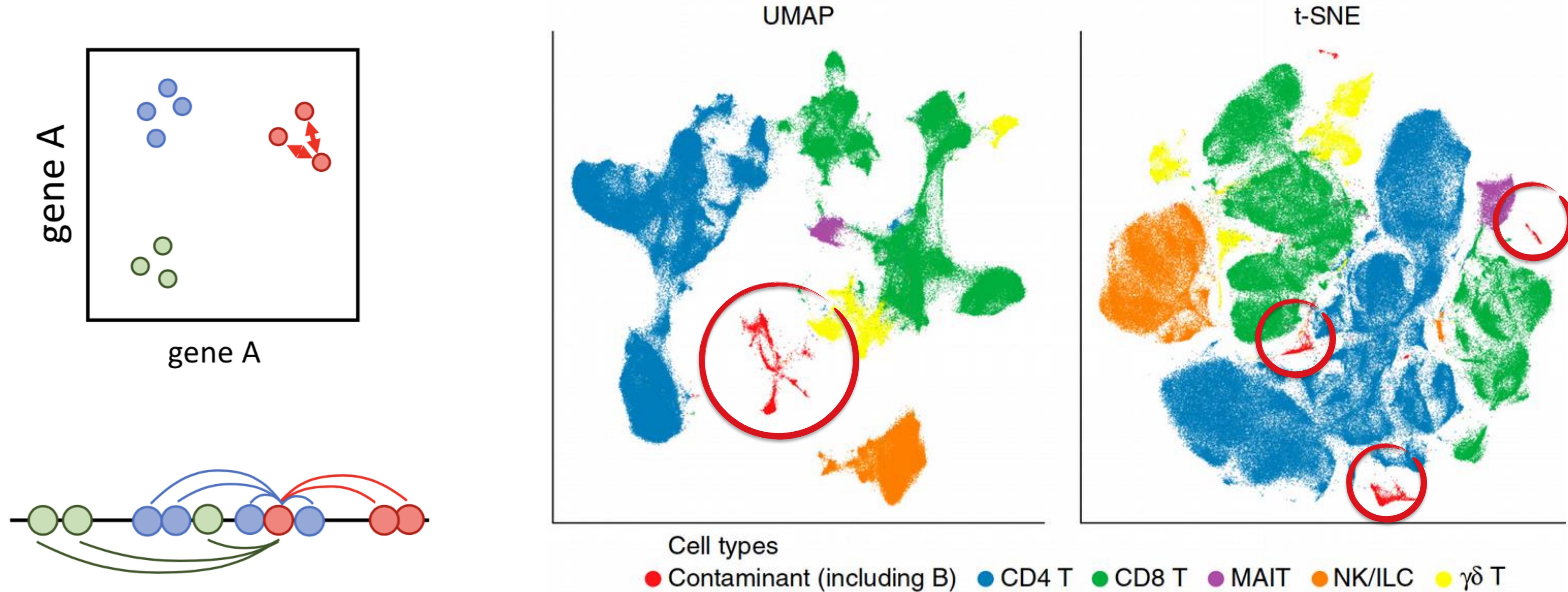


UMAP Intuition

- Similar principle as t-SNE
- Initializes with a non-heuristic “guess”
 - Same result every time
- Resolves global structure somewhat better (due to initialization)



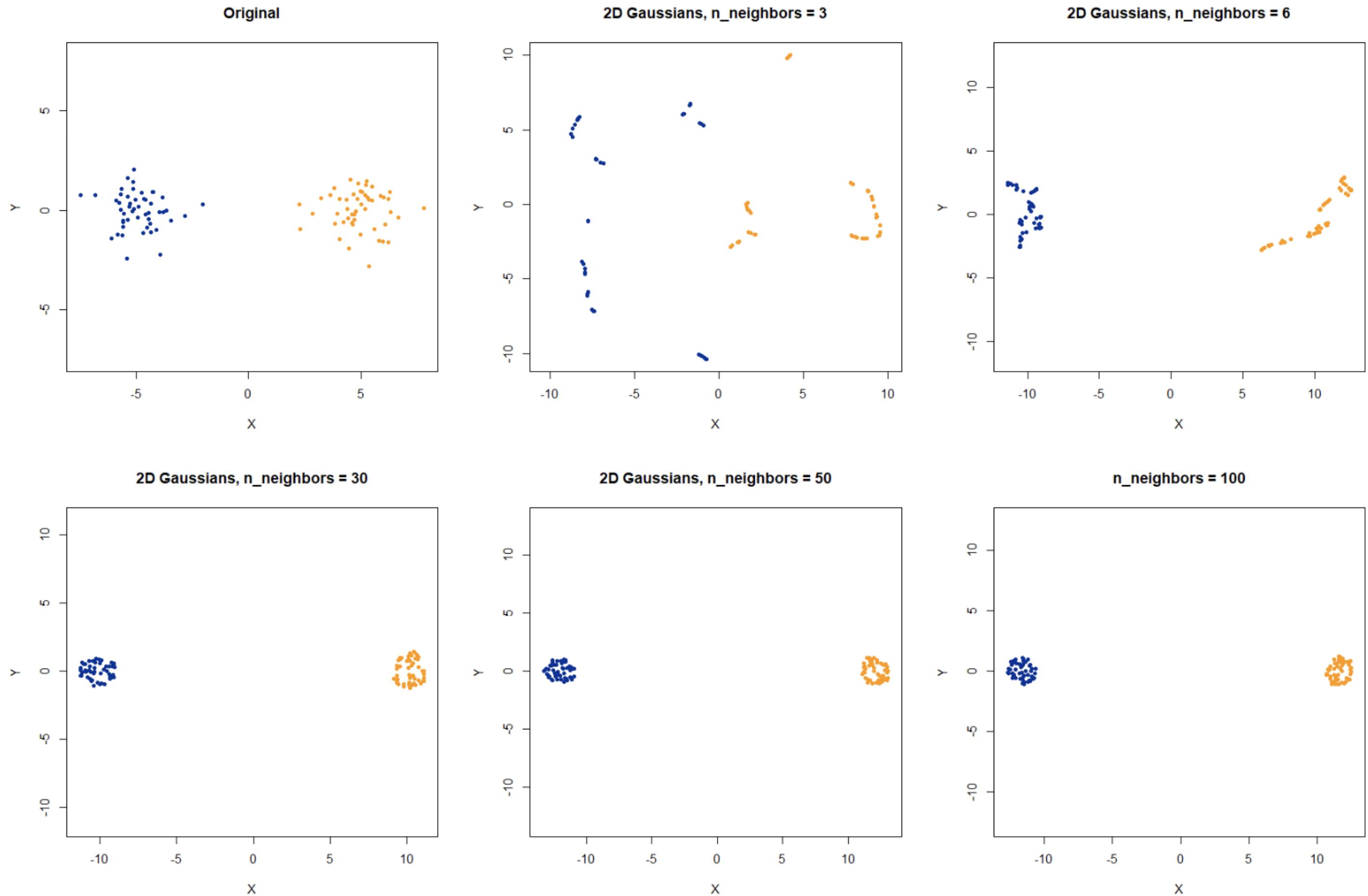
UMAP in Brief



UMAP Parameters

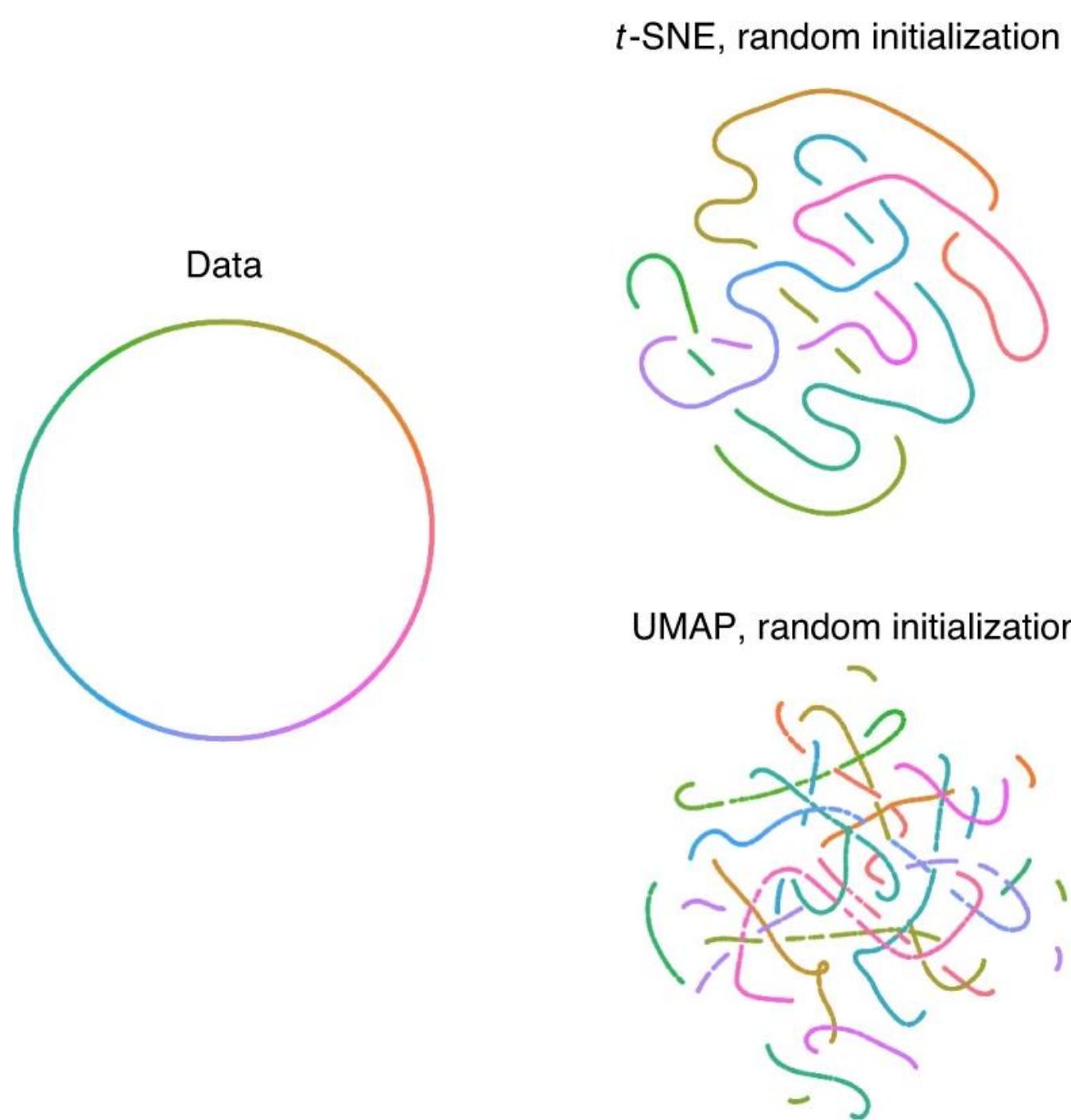
- Number of neighbors
- Number of iterations
- Minimum Distance (low-D)
- Metrics

...

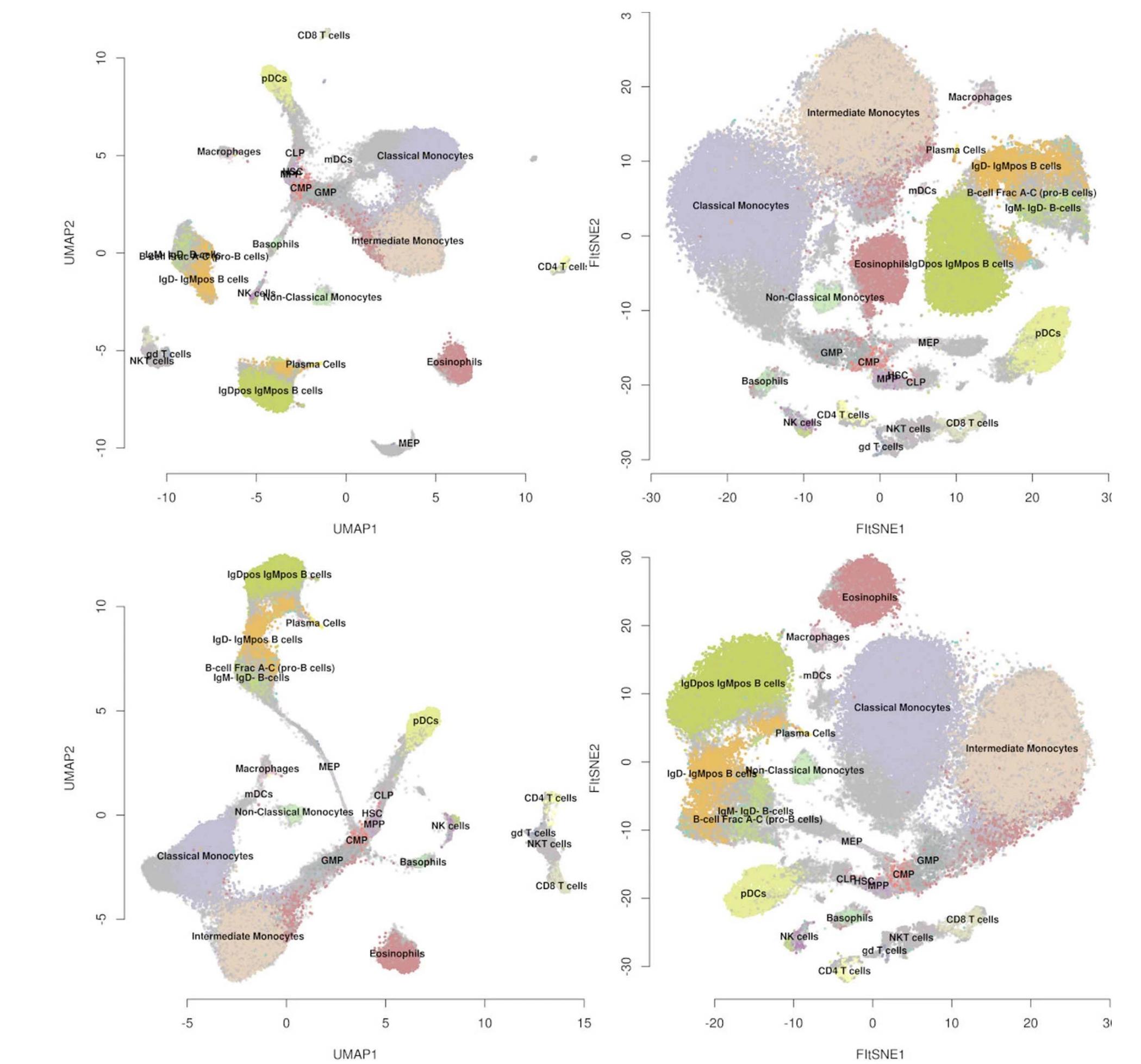


UMAP vs t-SNE

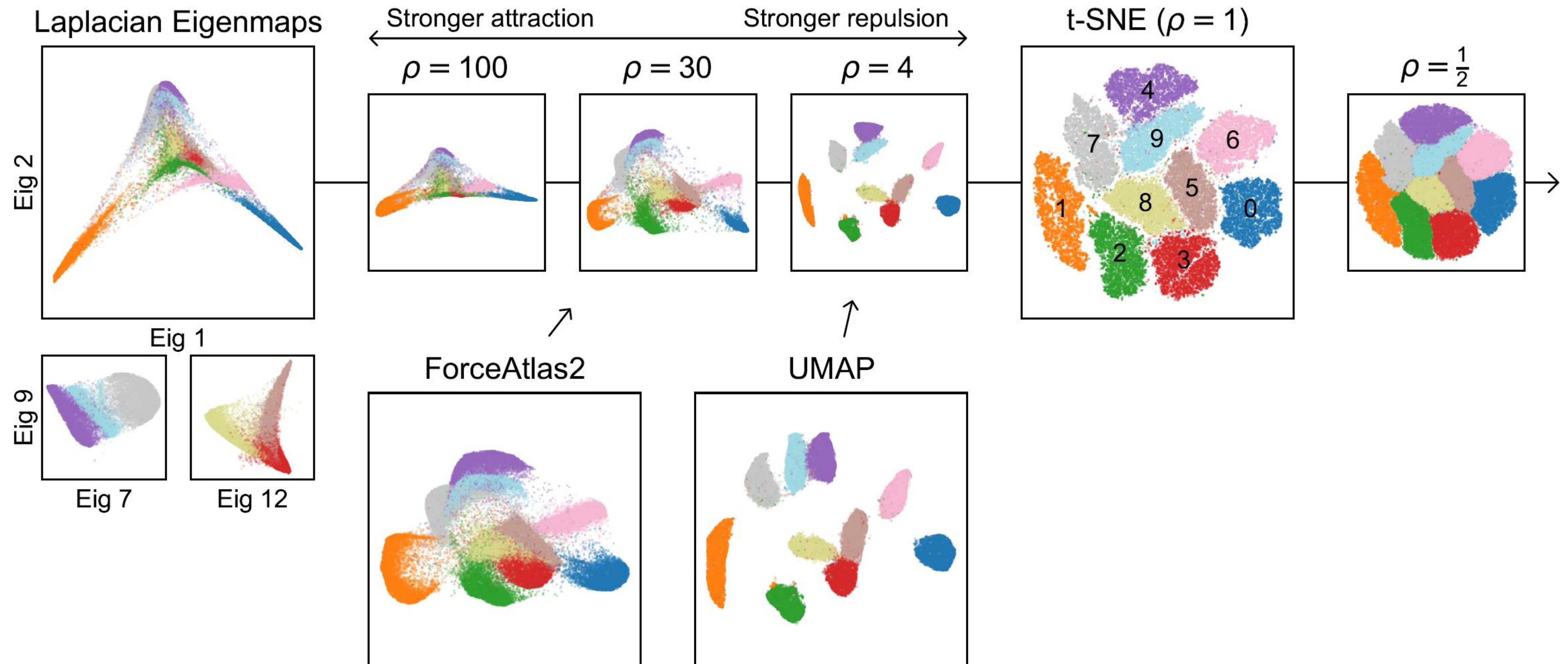
Artificial data



Real world data



UMAP vs t-SNE II



Summary: UMAP

- NON-LINEAR method of dimensionality reduction
- Can be run from the top PCs (e.g.: PC1 to PC10)
- Very efficient to compute, also for embedding dims > 2
- Is not randomly initialized by default
- It aims to 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

Further resources

- Paper: [UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction](#)
- Talk: [UMAP Uniform Manifold Approximation and Projection for Dimension Reduction | SciPy 2018](#)
- Code: [umap python](#) with its [docs](#) and especially on [How UMAP Works](#), [pynndescent](#), [umap.js](#)
- Paper: [Initialization is critical for preserving global data structure in both t-SNE and UMAP](#)
- Paper: [Dimensionality reduction for visualizing single-cell data using UMAP](#)
- Paper: [Parametric UMAP](#) and [Learning a Parametric Embedding by Preserving Local Structure](#)
- Paper: [Clustering with t-SNE, provably](#)
- Paper: [Fast interpolation-based t-SNE for improved visualization of single-cell RNA-seq data](#)
- Web: [The Beginner's Guide to Dimensionality Reduction](#)
- Another technique: [PHATE - Visualizing Transitions and Structure for Biological Data Exploration](#) and [Multiscale PHATE](#)

Dimensionality Reduction

Alexander Vieth, TU Delft

MGC/BioSB Course - Single Cell Analysis

11.10.2022

Comparing UMAP and t-SNE: Computations

UMAP

$$v_{j|i} = \exp\left(\frac{-\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\right)$$

ρ_i : distance to nearest neighbor

$$\sigma_i \text{ s.t. } \sum_{j=1}^k \exp\left(\frac{-\max(0, d(x_i, x_j) - \rho_i)}{\sigma_i}\right) = \log_2 k$$
$$v_{ij} = v_{j|i} + v_{i|j} - v_{j|i}v_{i|j}$$

High-dimensional similarities

$$w_{ij} = \left(1 + a\|y_i - y_j\|_2^{2b}\right)^{-1}$$

Where a, b depend on a hyper-parameter:
 $a \approx 1.929$ $b \approx 0.7915$

Low-dimensional similarities

Fuzzy set cross entropy

$$Cost = \sum_{i \neq j} v_{ij} \log \frac{v_{ij}}{w_{ij}} + (1 - v_{ij}) \log \left(\frac{1 - v_{ij}}{1 - w_{ij}} \right)$$

No random initialization
Stochastic gradient descent

Cost function for optimization process

t-SNE

$$v_{j|i} = \exp\left(\frac{-d(x_i, x_j) \square}{\sigma_i}\right) \quad \sigma_i \text{ s.t. } u = 2^{-\sum_j v_{j|i} \log_2 p_{j|i}} \quad p_{j|i} = \frac{v_{j|i}}{\sum_{k \neq i} v_{k|i}}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

Normalization step

$$w_{ij} = \left(1 + \|y_i - y_j\|_2^2\right)^{-1} \quad q_{ij} = \frac{w_{ij}}{\sum_{k \neq l} w_{kl}}$$

Kullback-Leibler divergence

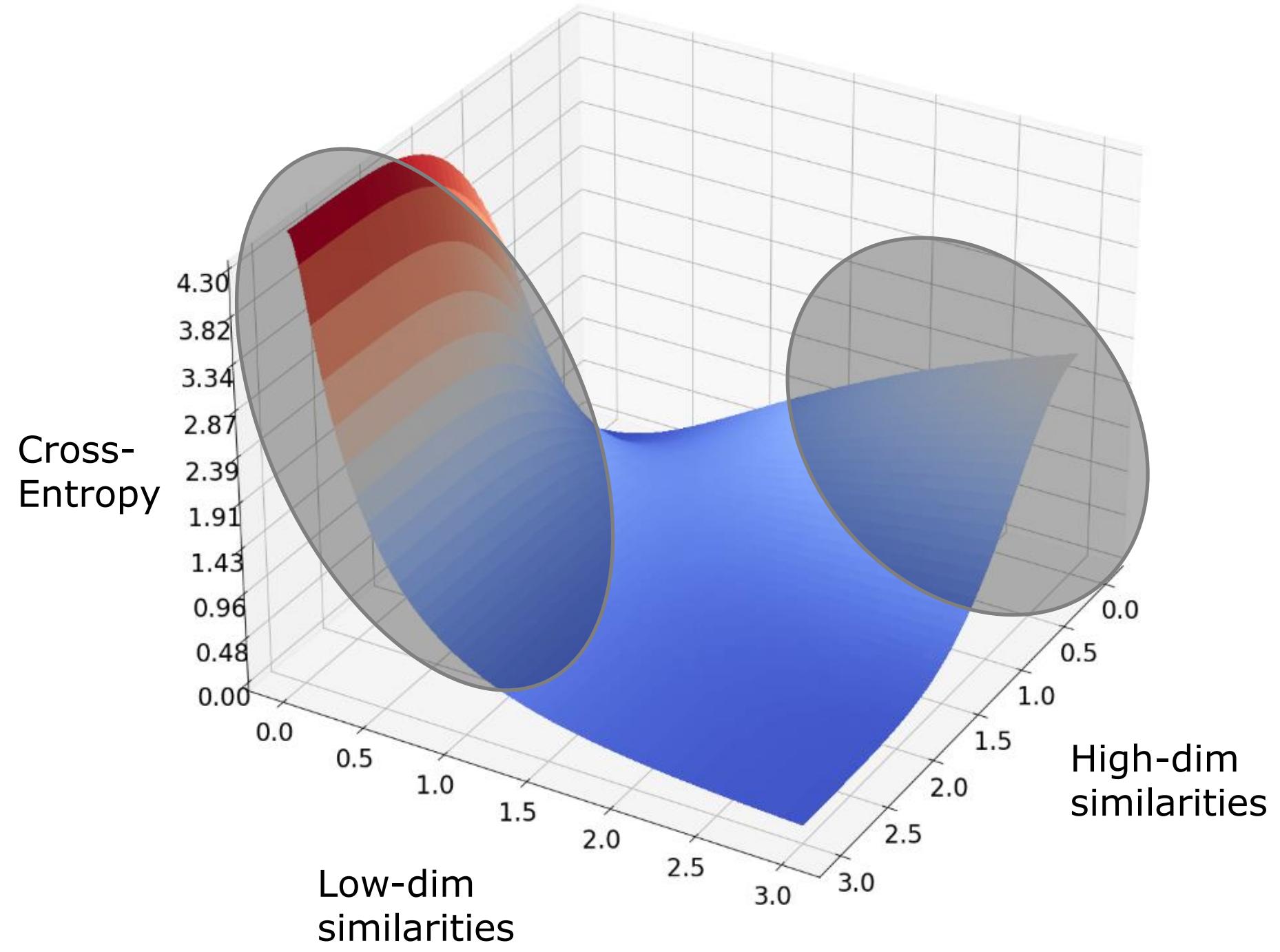
$$Cost = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Random initialization
Gradient descent

Comparing UMAP and t-SNE: Cost functions

But: t-SNE layout optimization results in repulsive forces between all data points
-> only those with low distances in high-dim space will be pulled together (their attractive forces are stronger than the general repulsive forces)

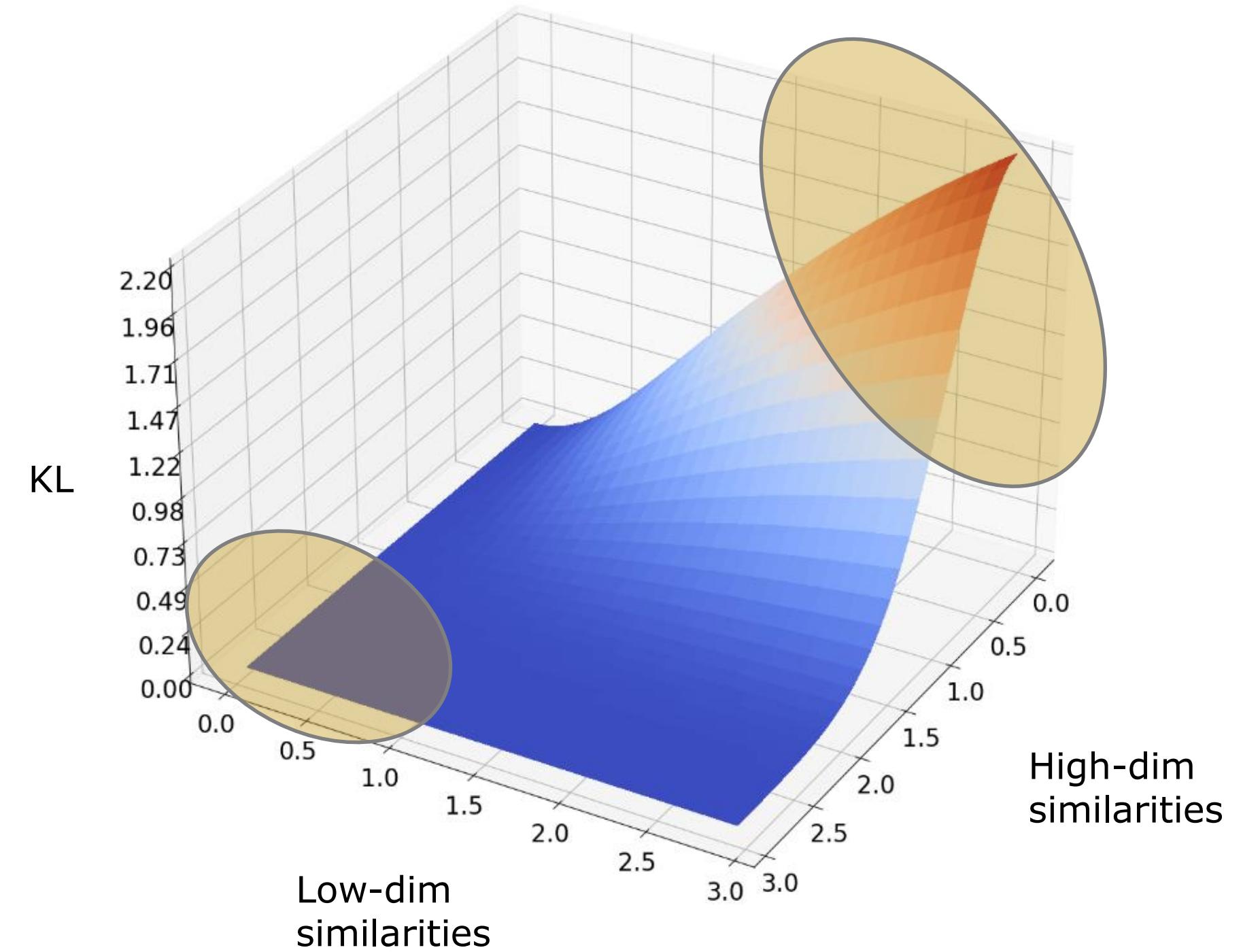
UMAP



Fuzzy set cross entropy

$$Cost = \sum_{i \neq j} v_{ij} \log \frac{v_{ij}}{w_{ij}} + (1 - v_{ij}) \log \left(\frac{1 - v_{ij}}{1 - w_{ij}} \right)$$

t-SNE

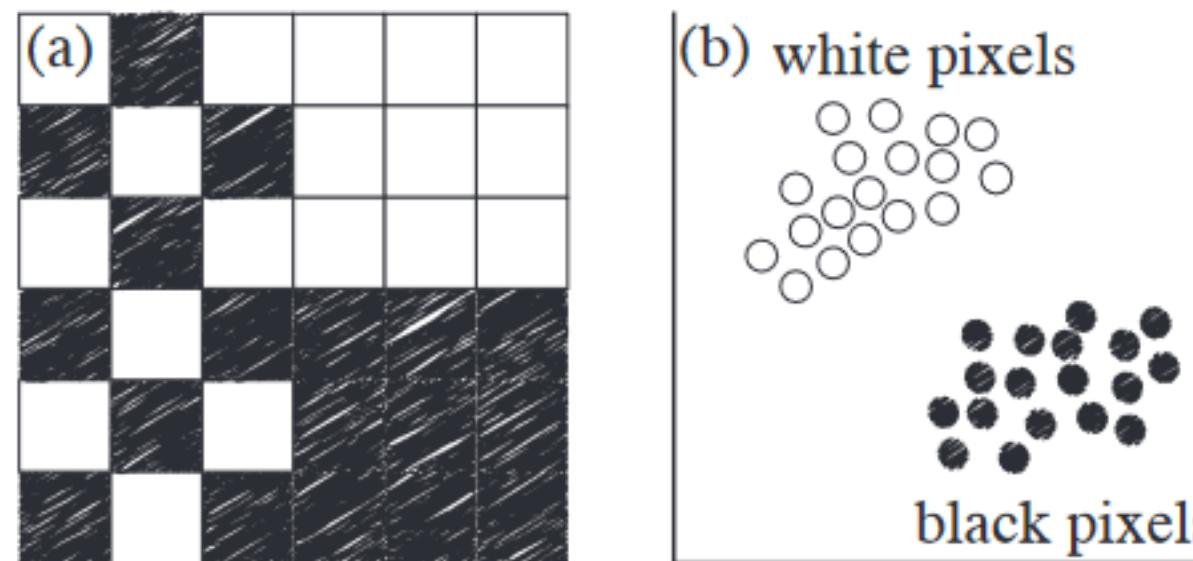


Kullback-Leibler (KL) divergence

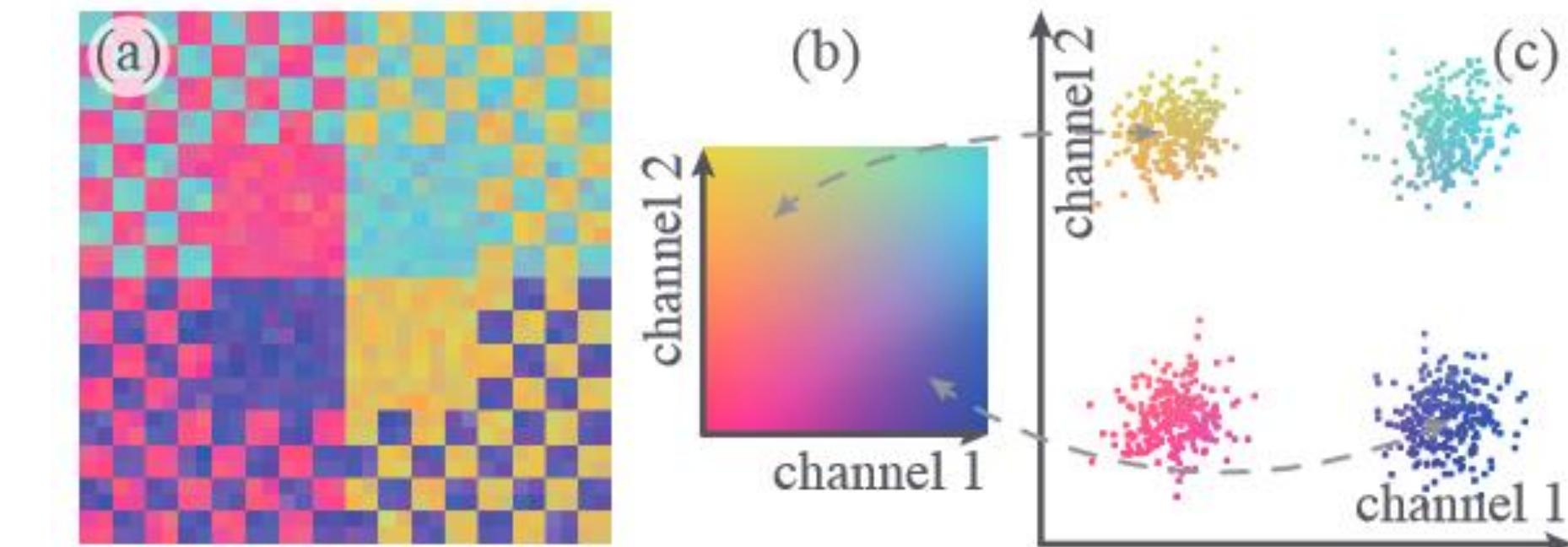
$$Cost = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Spatial Information in Dimensionality Reduction

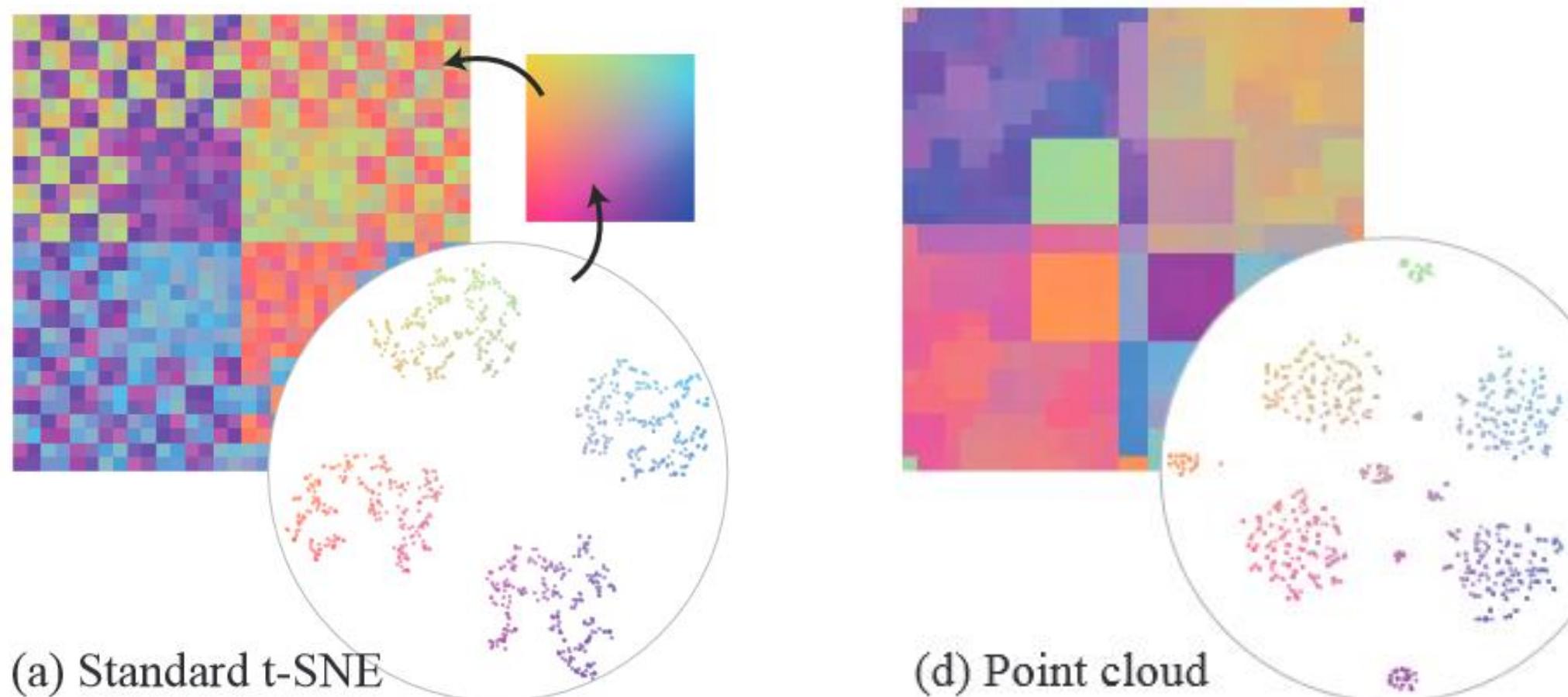
Idea



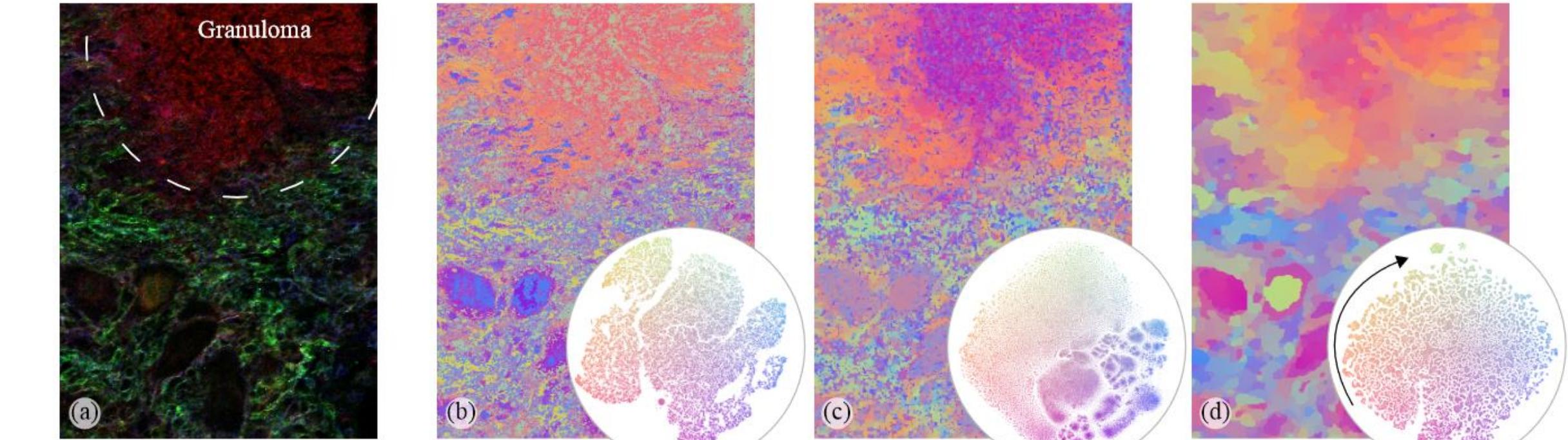
Data



Test



Example



A. Vieth, A. Vilanova, B. Lelieveldt, E. Eisemann and T. Höllt, "Incorporating Texture Information into Dimensionality Reduction for High-Dimensional Images," 2022 IEEE 15th Pacific Visualization Symposium (PacificVis), 2022, pp. 11-20, doi: 10.1109/PacificVis53943.2022.00010.
<https://github.com/biovault/Spidr>