

Single cell RNA-seq analysis

Part I: data processing & analysis

BMI/CS 776

www.biostat.wisc.edu/bmi776/

Spring 2024

Daifeng Wang

daifeng.wang@wisc.edu

Thanks to Ting Jin for slides!

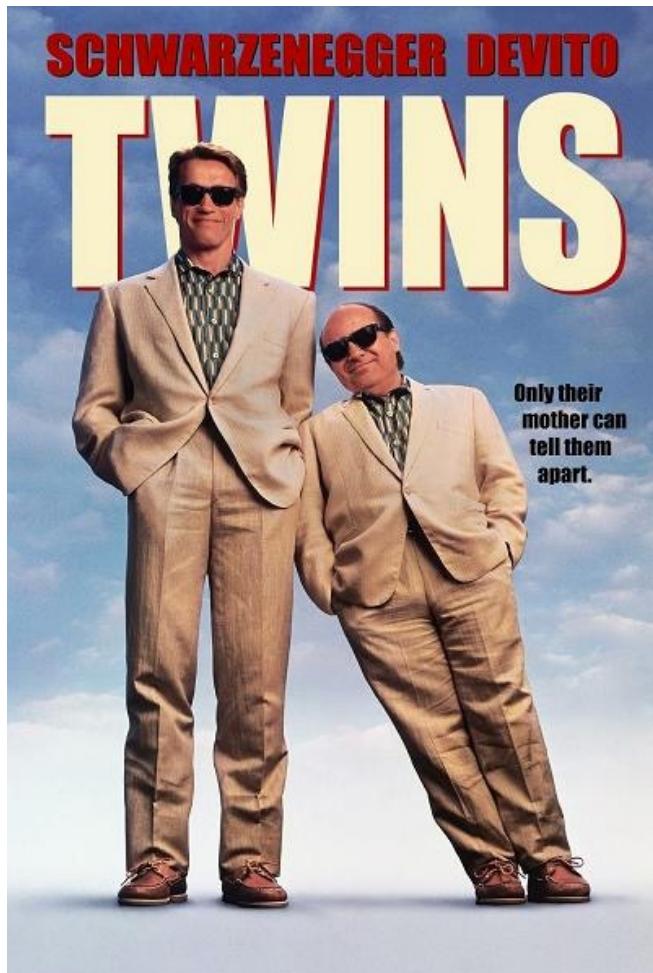
These slides, excluding third-party material, are licensed under [CC BY-NC 4.0](https://creativecommons.org/licenses/by-nc/4.0/) by Mark Craven, Colin Dewey, Anthony Gitter and Daifeng Wang

Outline

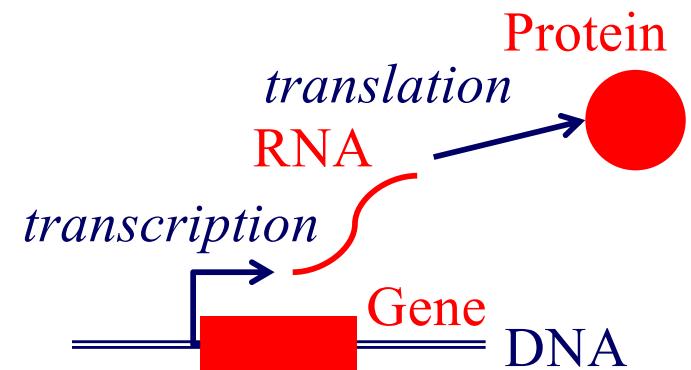
- **Introduction on single cell RNA sequencing**
- Single cell RNA sequencing (scRNA-seq) data processing
- scRNA-seq data analysis

Gene expression and regulation

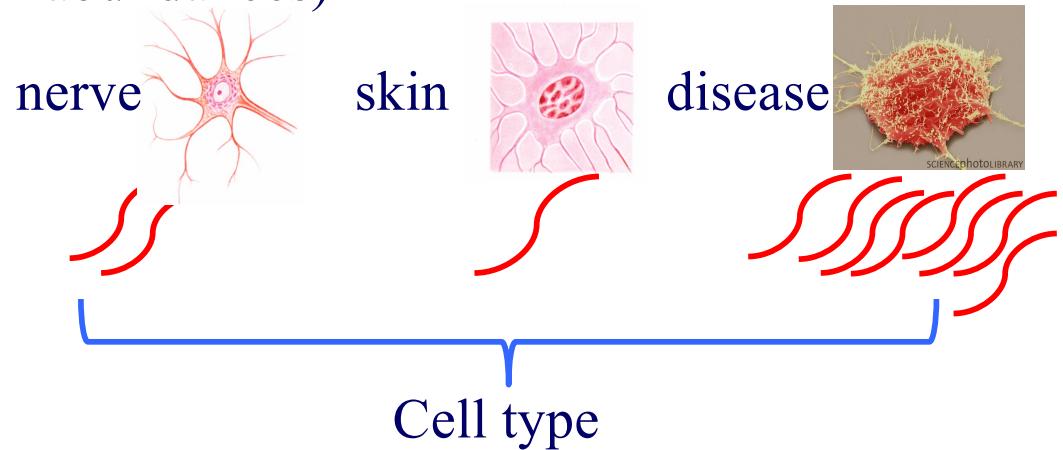
Identical DNA but different gene expression



Central dogma

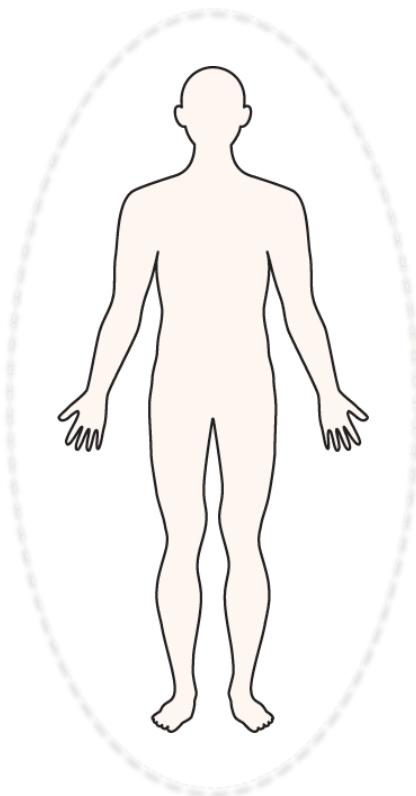


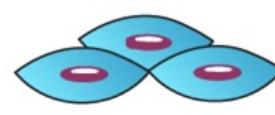
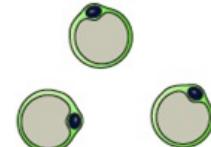
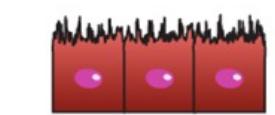
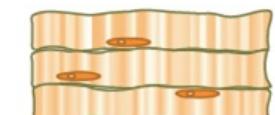
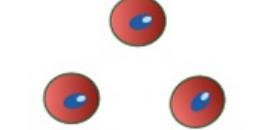
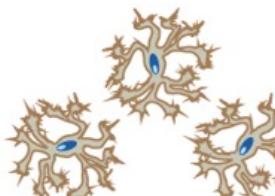
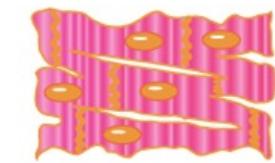
Gene expression levels (e.g., values to quantify RNA abundances)



Gene regulation: which & how genes express?

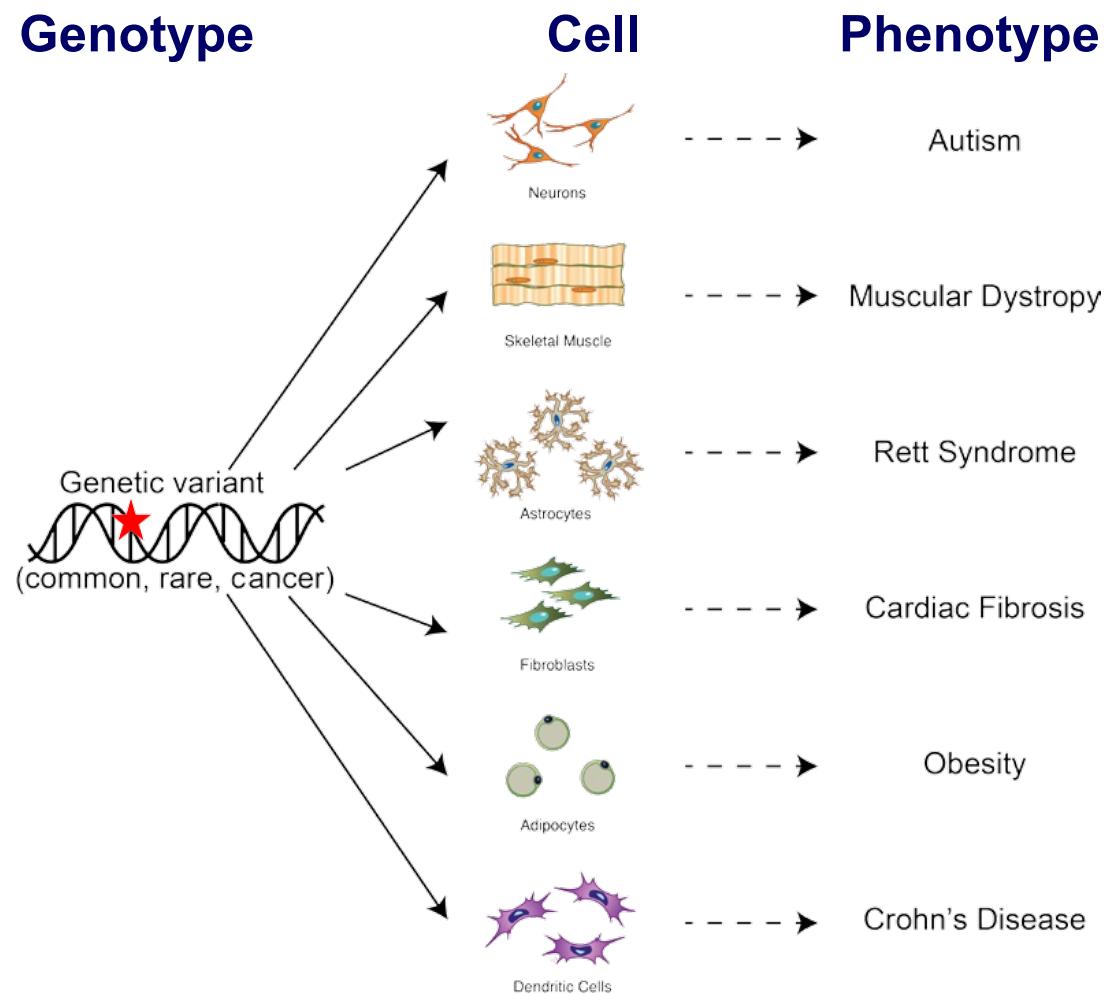
Why study single cells?



Skin Cells	Immune Cells	Brain Cells	Muscle Cells
 Fibroblasts	 Megakaryocytes	 Neurons	 Smooth Muscle
 Adipocytes	 Dendritic Cells	 Ependymal Cells	 Skeletal Muscle
 Epithelial Cells	 T Cells	 Astrocytes	 Cardiac Muscle

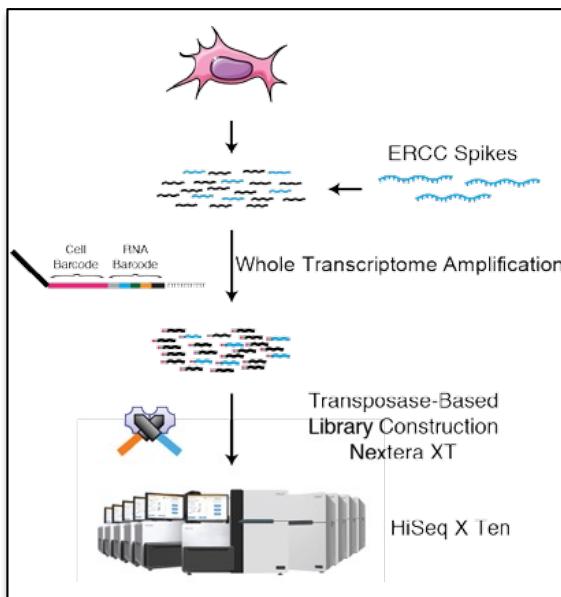
Cells are our core constituents, are classified by characteristic molecules, structures, and functions

Why study single cells?

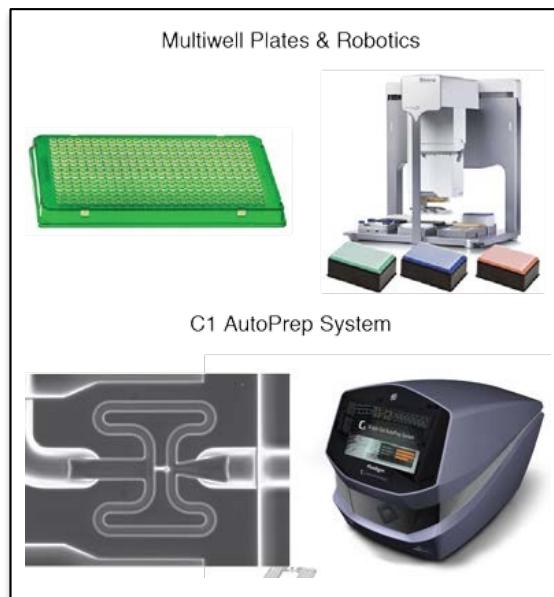


Cells are key intermediate from genotype to phenotype, e.g., essential functional dissection of genetic variants

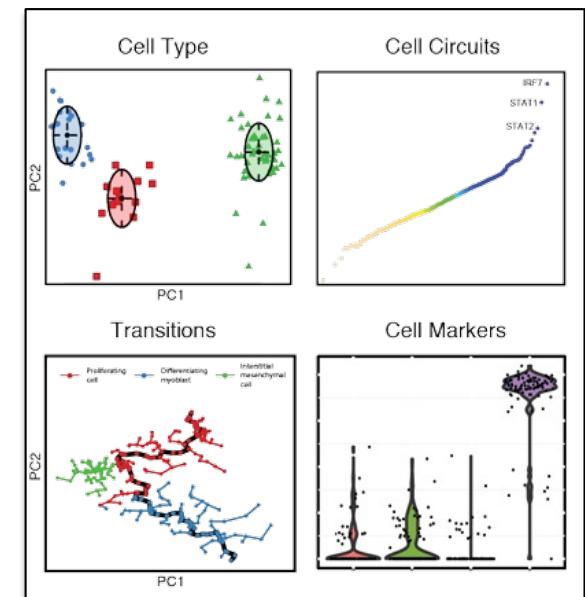
Sequencing RNAs in single cells



✓ Core technology



✓ Sample prep



✓ Computation

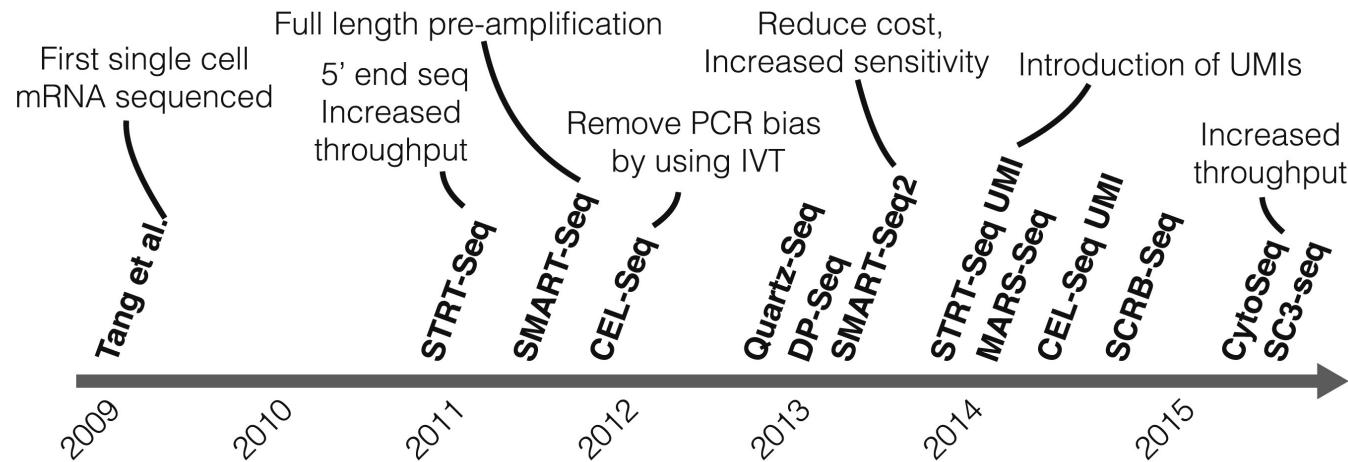
2012: 18 cells



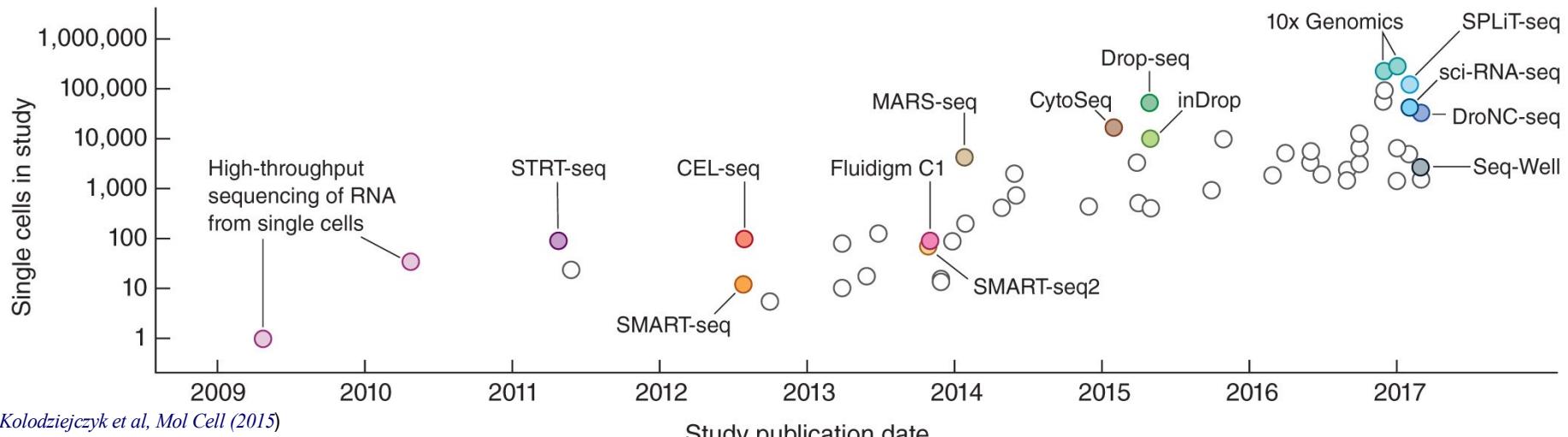
2020: ~100,000 cells

Evolution of single cell sequencing

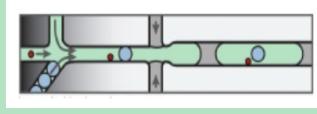
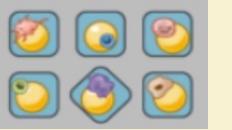
- High-throughput technologies enable the profiling of thousands of cells in parallel, providing an unbiased view of the heterogeneity of single cells within a population. *Fan et al., 2015*



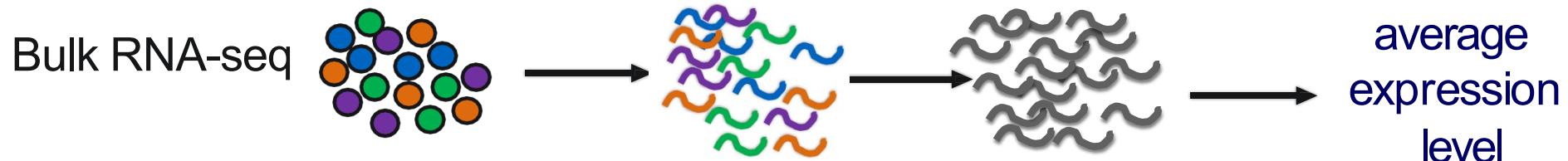
- Cell numbers reported in representative publications by publication date



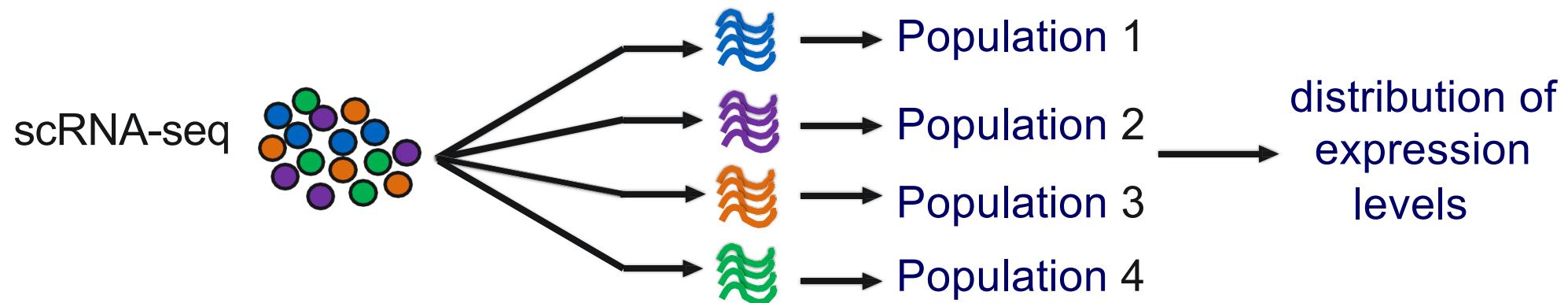
Single cell RNA sequencing (scRNA-seq) technology

	inDrops	10x Genomics	Drop-seq	Seq-well (Honeycomb)	SMART-seq
Cell capture efficiency	~70-80%	~50-70%	~10%	~80%	~80%
Time to capture 10k cells	~30min	10min	1-2 hours	5-10min	--
Encapsulation type	Droplet 	Droplet 	Droplet 	Nanolitre well 	Plate-based 
Library prep	CEL-seq Linear amplification by RT	SMART-seq Exponential PCR based amplification	SMART-seq Exponential PCR based amplification	SMART-seq Exponential PCR based amplification	SMART-seq Exponential PCR based amplification
Commercial	Yes	Yes	--	Yes (Summer 2020)	Yes
Cost (~\$ per cell)	~0.06	~0.2	~0.06	~0.15	1
Strengths	<ul style="list-style-type: none"> Good cell capture Cost-effective Real-time monitoring Customizable 	<ul style="list-style-type: none"> Good cell capture Fast and easy to run Parallel sample collection High gene / cell counts 	<ul style="list-style-type: none"> Cost-effective Customizable 	<ul style="list-style-type: none"> Good cell capture Cost-effective Real-time monitoring Customizable 	<ul style="list-style-type: none"> Good cell capture Good mRNA capture Full-length transcript No UMI
Weaknesses	Difficult to run	Expensive	Difficult to run & low cell capture efficiency	Available Soon	Expensive

Bulk vs scRNA-seq



- quantifying expression signatures from ensembles
- insufficient for studying heterogeneous system



- inference of gene regulatory networks across the cells
- heterogeneity of cell responses
- cell type identification

Outline

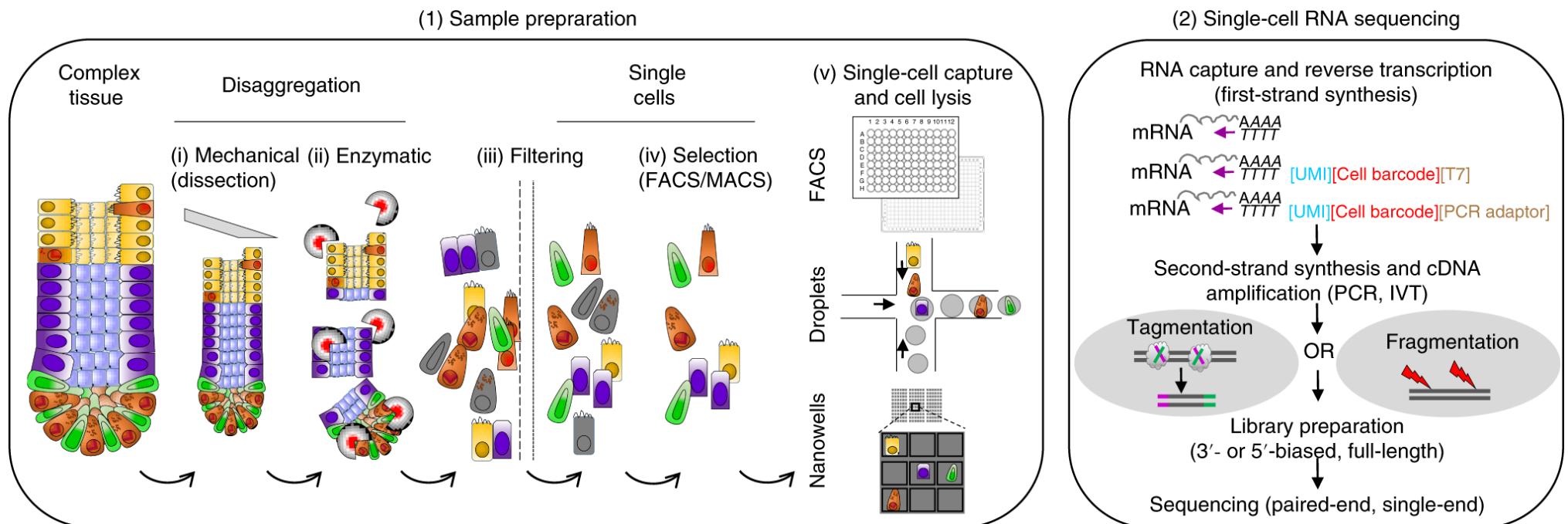
- Introduction on single cell RNA sequencing
- **Single cell RNA sequencing (scRNA-seq) data processing**
- scRNA-seq data analysis

Single-cell RNA sequencing (scRNA-seq) process

- Step1: Sample preparation
- Step2: Single-cell RNA sequencing
- Step3: Data processing
- Step4: Data analysis

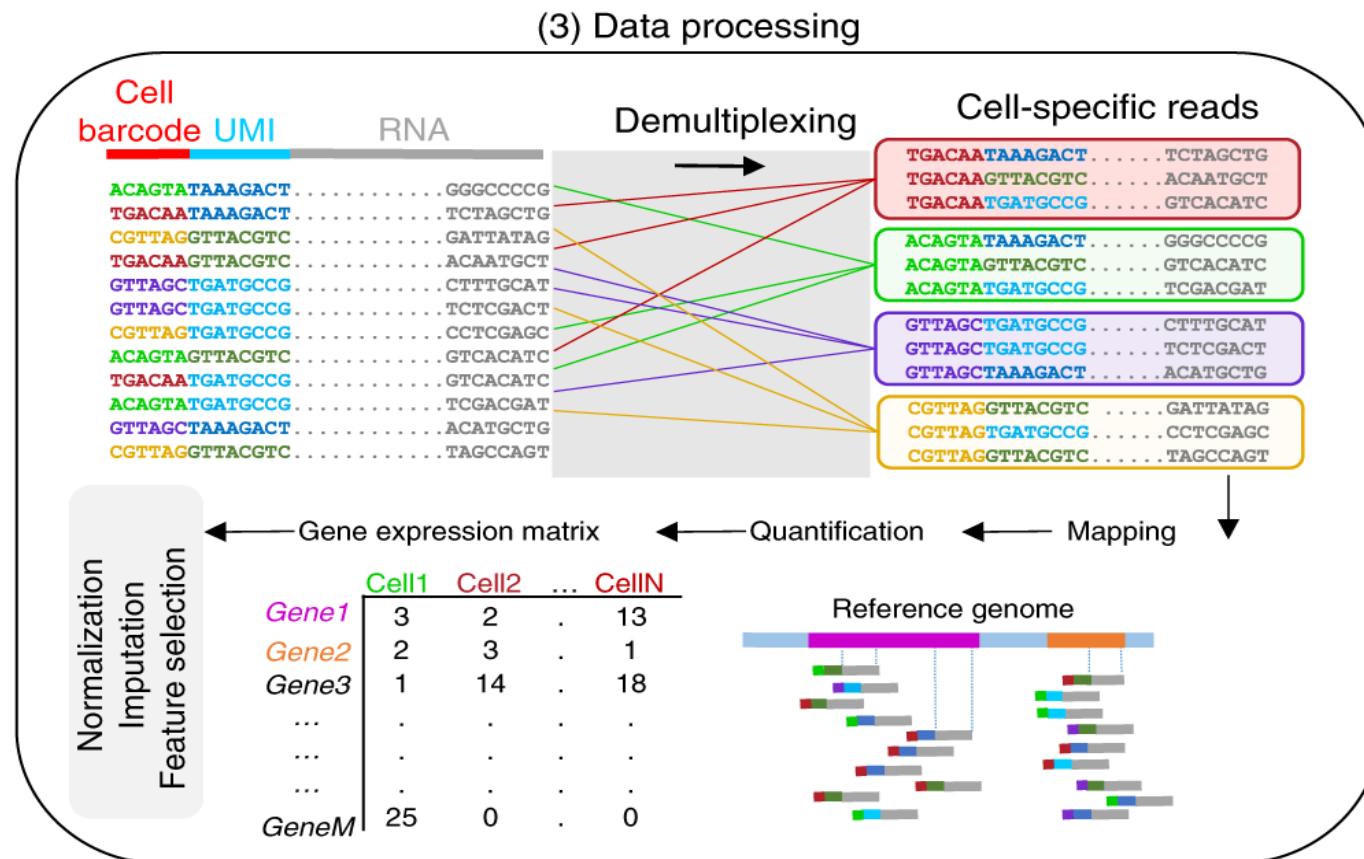
scRNA-seq process

- Step 1 - Sample preparation: cells are physically separated into a single-cell solution from which specific cell types can be enriched or excluded
- Step 2 - Single-cell RNA sequencing



scRNA-seq process

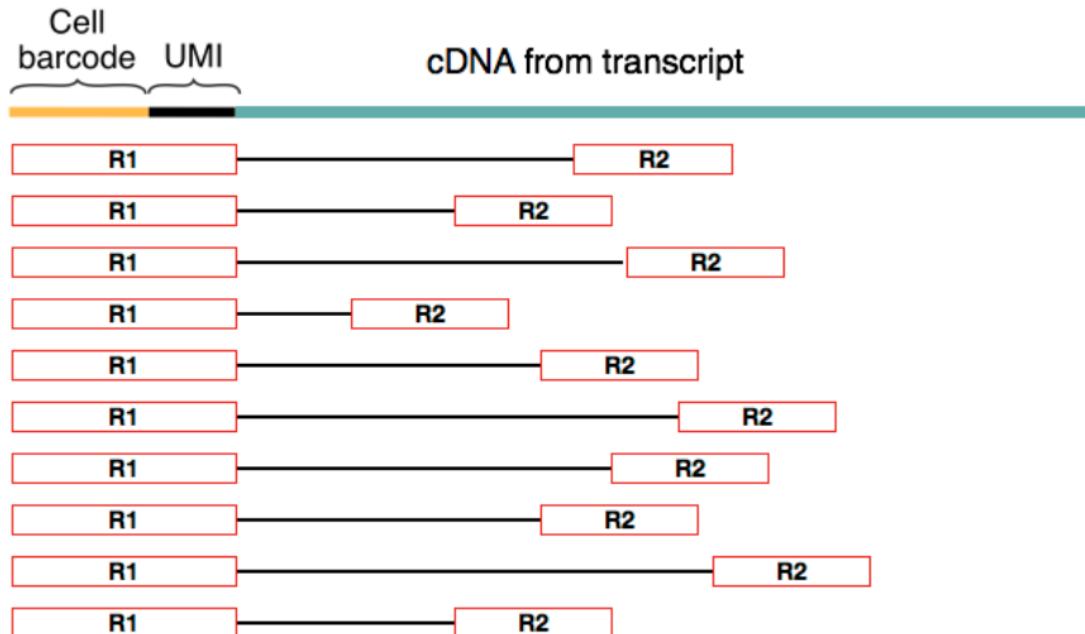
- Step 3 - Data processing
 - Unique molecular identifier (UMI)
 - Gene counts
 - Drop-outs in single cell
 - Imputation method: MAGIC



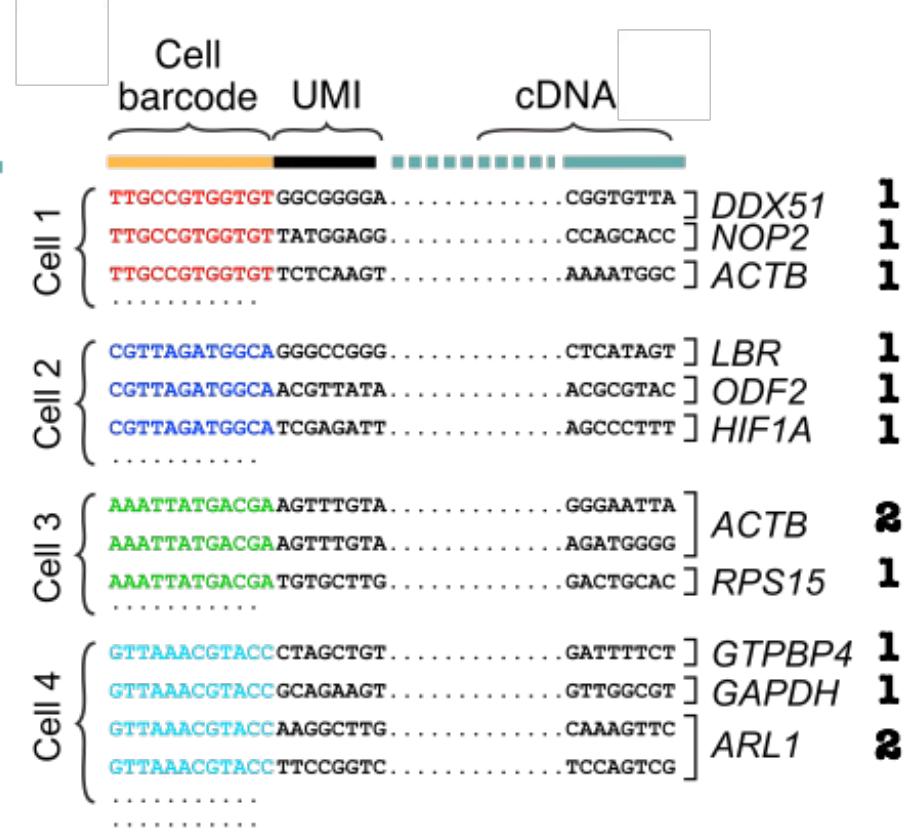
Unique molecular identifier (UMI)

- UMIs are short (4-10bp) random barcodes added to transcripts during reverse-transcription.

Biased paired-end reads

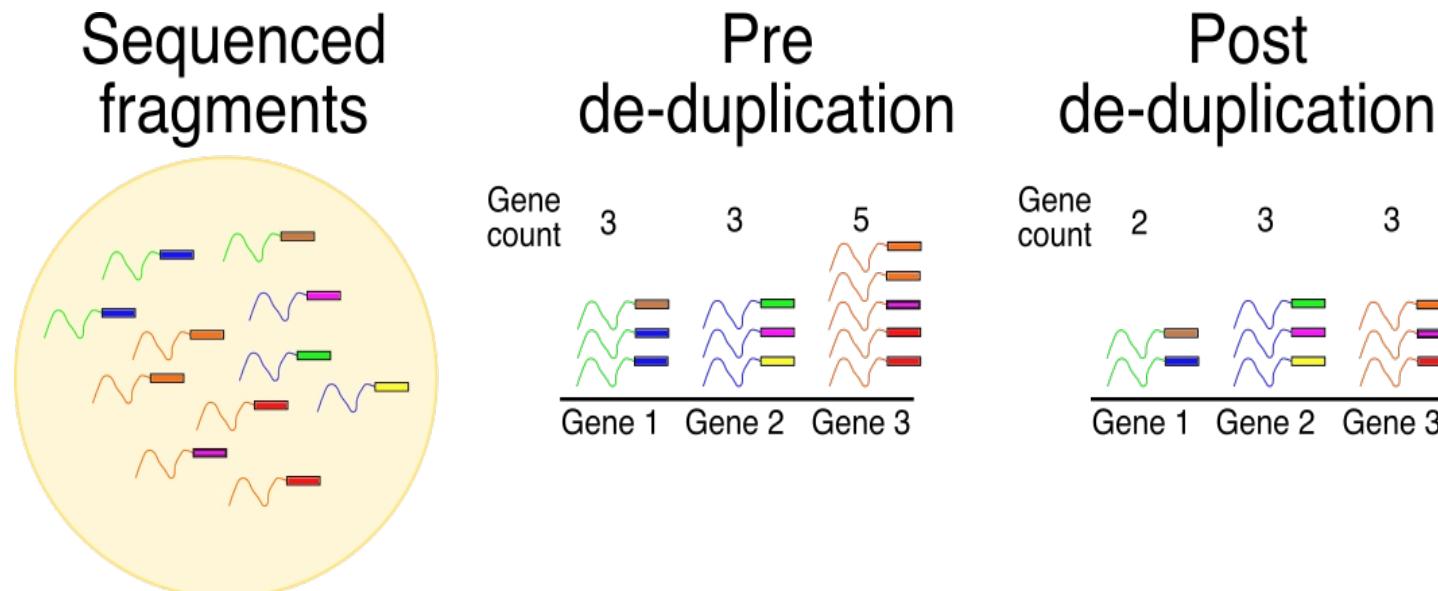


Grouping barcodes to assign reads to cells



Unique molecular identifier (UMI)

- They enable sequencing reads to be assigned to individual transcript molecules and thus the removal of amplification noise and biases from scRNA-Seq data
- They reduce the amplification noise by allowing (almost) complete de-duplication of fragments



Gene Counts

- In each gene, within each cell, the total number of unique UMI is counted and reported as the number of transcripts of that gene for a given cell.

Hundreds of millions of reads

	Cell barcode	UMI	cDNA
Cell 1			
	TTGCCGTGGTGT	GGCGGGGA.....	CGGTGTTA
	TTGCCGTGGTGT	TATGGAGG.....	CCAGCACC
	TTGCCGTGGTGT	TCTCAAGT.....	AAAATGGC
Cell 2			
	CGTTAGATGGCA	GGGCCGGG.....	CTCATAGT
	CGTTAGATGGCA	ACGTTATA.....	ACCGGTAC
	CGTTAGATGGCA	TCGAGATT.....	AGCCCTTT
Cell 3			
	AAATTATGACGA	AGTTTGTA.....	GGGAATTAA
	AAATTATGACGA	AGTTTGTA.....	AGATGGGG
	AAATTATGACGA	TGTGCTTG.....	GACTGCAC
Cell 4			
	GTAAACGTACC	CTAGCTGT.....	GATTTTCT
	GTAAACGTACC	GCAGAAAGT.....	GTGGCGT
	GTAAACGTACC	AAGGCTTG.....	CAAAGTTC
	GTAAACGTACC	TTCCGGTC.....	TCCAGTCG

cDNA alignment to
Genome and group
Results by cell

Thousands of cells

	Cell barcode	UMI	cDNA	
Cell 1				
	TTGCCGTGGTGT	GGCGGGGA.....	CGGTGTTA	DDX51 1
	TTGCCGTGGTGT	TATGGAGG.....	CCAGCACC	NOP2 1
	TTGCCGTGGTGT	TCTCAAGT.....	AAAATGGC	ACTB 1
Cell 2				
	CGTTAGATGGCA	GGGCCGGG.....	CTCATAGT	LBR 1
	CGTTAGATGGCA	ACGTTATA.....	ACCGGTAC	ODF2 1
	CGTTAGATGGCA	TCGAGATT.....	AGCCCTTT	HIF1A 1
Cell 3				
	AAATTATGACGA	AGTTTGTA.....	GGGAATTAA	ACTB 2
	AAATTATGACGA	AGTTTGTA.....	AGATGGGG	RPS15 1
	AAATTATGACGA	TGTGCTTG.....	GACTGCAC	
Cell 4				
	GTAAACGTACC	CTAGCTGT.....	GATTTTCT	GTPBP4 1
	GTAAACGTACC	GCAGAAAGT.....	GTGGCGT	GAPDH 1
	GTAAACGTACC	AAGGCTTG.....	CAAAGTTC	ARL1 2
	GTAAACGTACC	TTCCGGTC.....	TCCAGTCG	

Count unique UMIs
for each gene in each cell

Create digital
expression matrix

Gene counts matrix

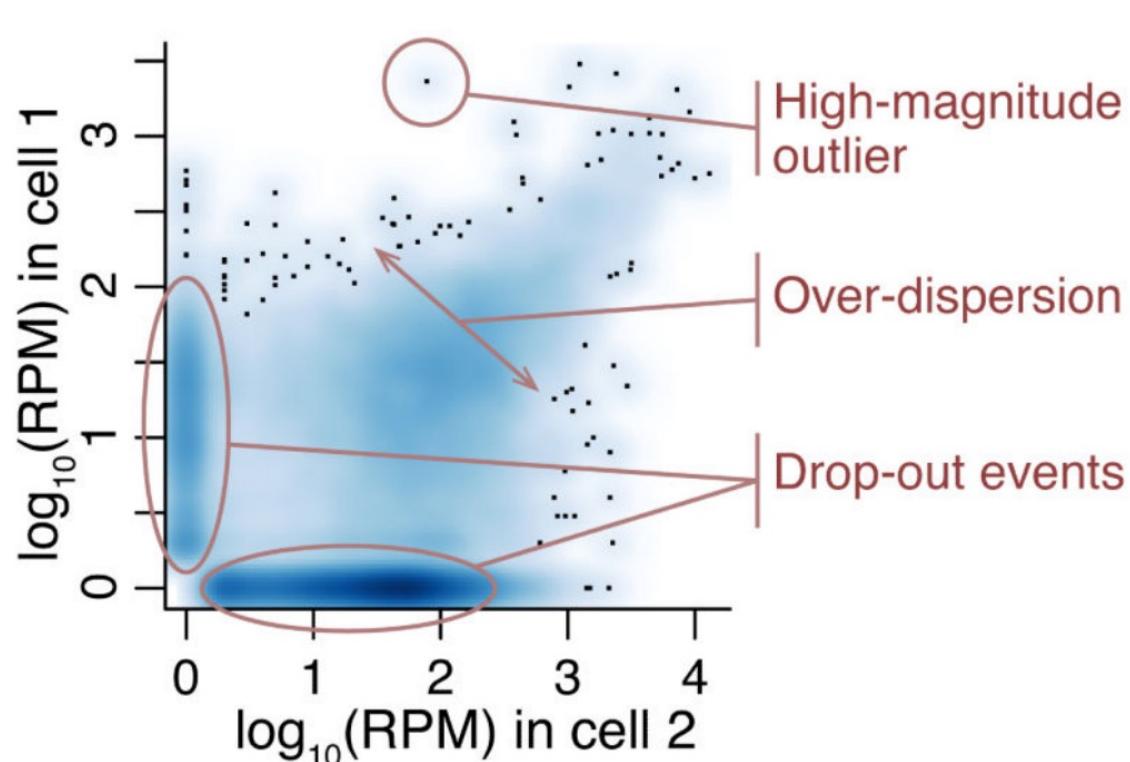
	Cell: 1	2	...	N
GENE	1	2		14
GENE 2	4	27		8
GENE 3	0	0		1
GENE M	6	2		0

Drop-outs in single cell

- a gene is observed at a moderate or high expression level in one cell but is not detected in another cell

Why do dropouts occur in single cell?

- technical artifacts
- cell type differences
- statistical sampling
- biological factors



Drop-outs in single cell

What should we do about dropouts?

- Ignore zero inflation
- Preprocess/reduce dimensions
- Impute scRNA-seq gene count matrix before analysis



Why do we need imputation methods?

- Downstream analyses relying on the accuracy of gene expression measurements



Imputation Methods

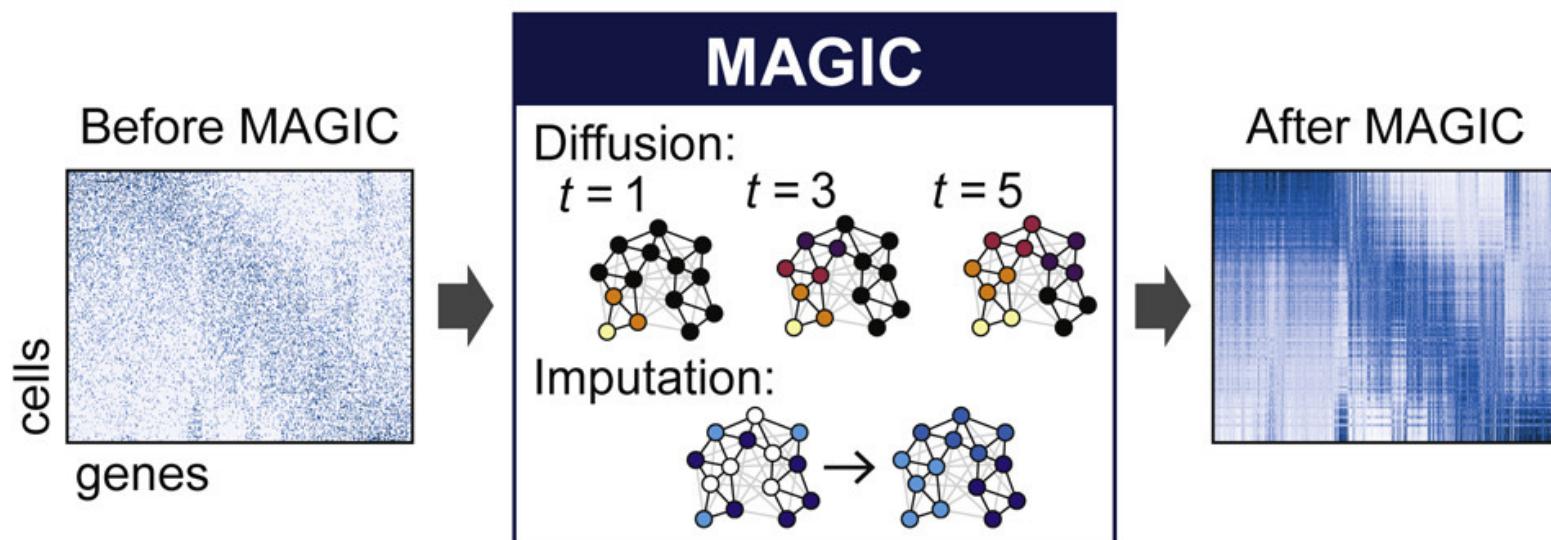
- MAGIC
- Droplet
- DrImpute
- scDoc



MAGIC

Markov affinity-based graph imputation of cells

- Denoise high-dimensional scRNA-seq data
- Impute missing expression values by sharing information across similar cells
- Similarity between two cells $A_{ij} = e^{-(\frac{Distance_{ij}}{\sigma})^2}$
- Transform the similarity matrix A into a Markov transition matrix M
- Raise the Markov matrix to the power of t : M^t , which determines the weights of cells

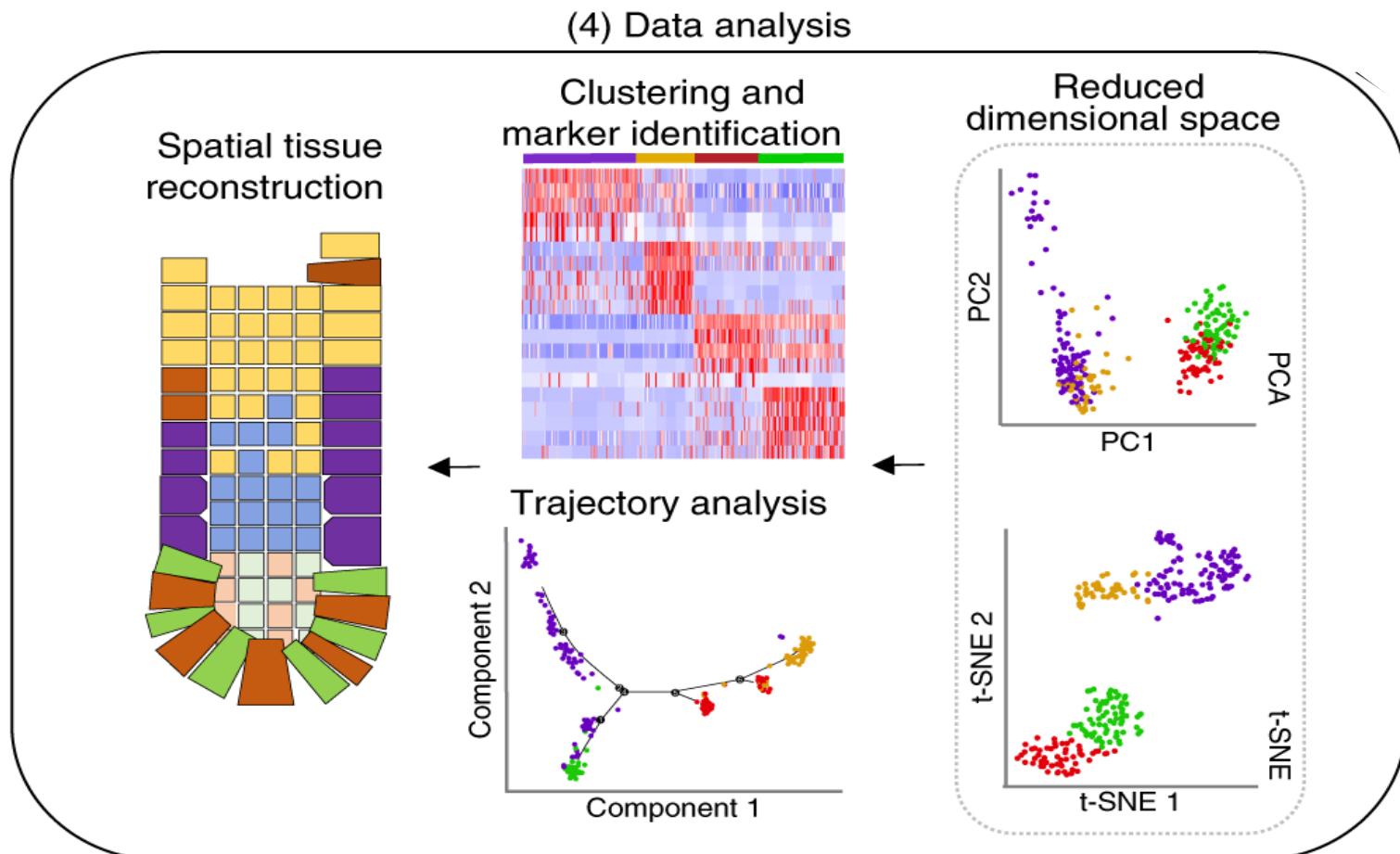


<http://jsb.ucla.edu/sites/default/files/scImpute.pdf>

van Dijk D, Sharma R, Nainys J, Yim K, Kathail P, Carr AJ, Burdziak C, Moon KR, Chaffer CL, Pattabiraman D, Bierie B, Mazutis L, Wolf G, Krishnaswamy S, Pe'er D. Recovering Gene Interactions from Single-Cell Data Using Data Diffusion. *Cell*. 2018 Jul 26;174(3):716-729.e27. doi: 10.1016/j.cell.2018.05.061. Epub 2018 Jun 28. PMID: 29961576; PMCID: PMC6771278.

scRNA-seq process

- Step 4 - Data analysis
 - Dimensionality reduction
 - Clustering and marker identification
 - Trajectory analysis



Outline

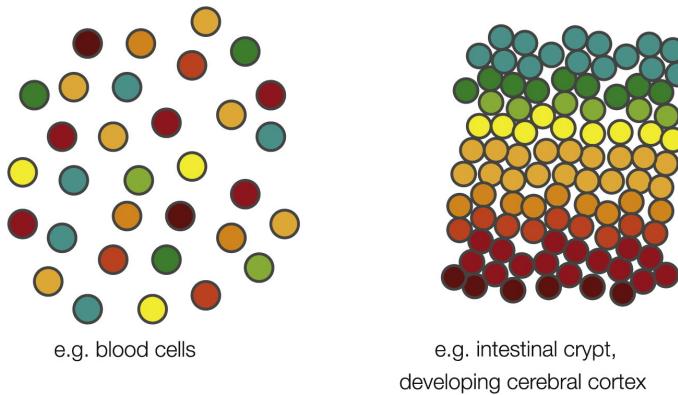
- Introduction on single cell RNA sequencing
- Single cell RNA sequencing (scRNA-seq) data processing
- **scRNA-seq data analysis**

scRNA-seq data analysis

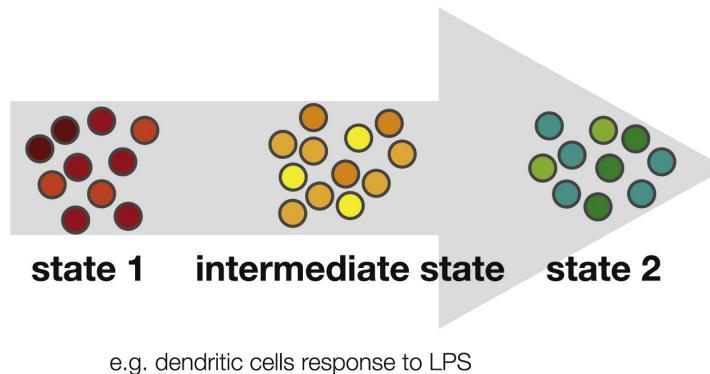
- **Introduction of single cell analysis**
- Dimensionality reduction
 - Linear: PCA
 - Non-linear: t-SNE, UMAP
- Cell clustering
 - K-means
 - hierarchical clustering
 - graph-based clustering

Why single-cell analysis ?

- Single-cell data opens the door to several types of biological questions:
 - What cell types exist within a population of cells in a particular sample?

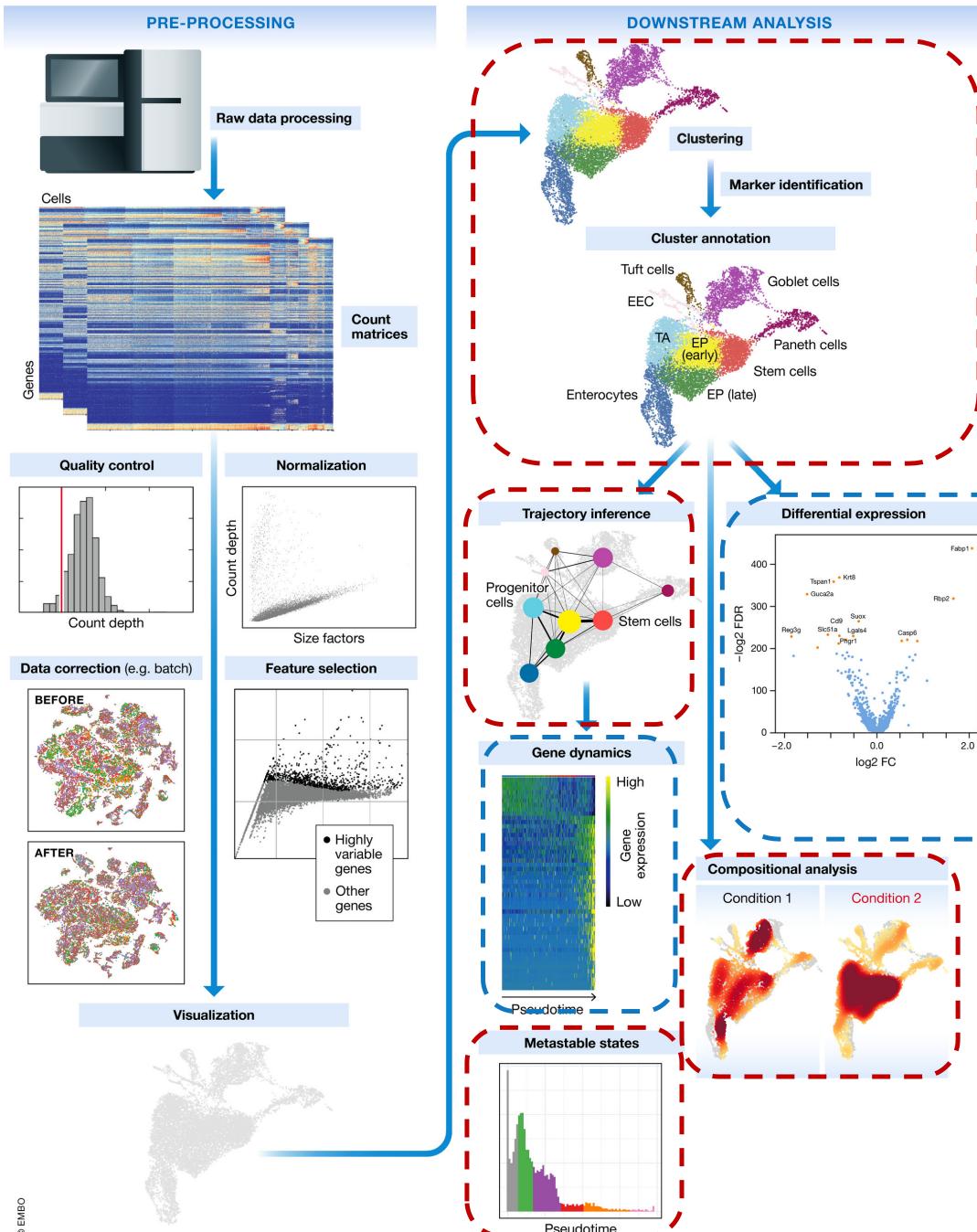


- What does this high-resolution view of cellular transitions tell us about switches in cell states?



- How can gene regulatory networks drive cell types or states?

scRNA-seq analysis



Cell-level analysis

- Marker gene identification
- Cluster analysis
- Trajectory analysis

Gene-level analysis

- Single-cell differential expression analysis
- Gene set analysis
- Gene regulatory networks

Python for scRNA-seq data analysis

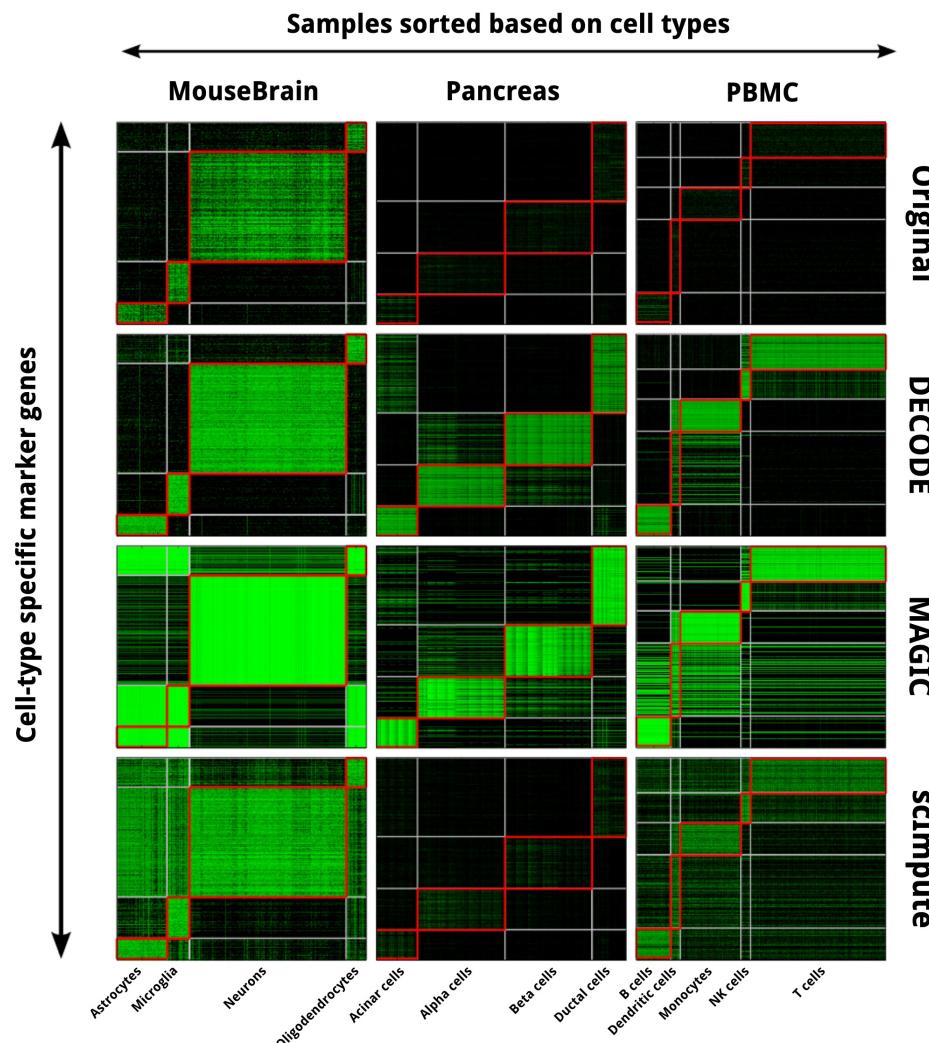


Single-cell RNA sequencing analysis

- Introduction of single cell analysis
- **Dimensionality reduction**
 - Linear: PCA
 - Non-linear: t-SNE, UMAP
- Cell clustering
 - K-means
 - hierarchical clustering
 - graph-based clustering

Dimensionality Reduction

- Dimensionality reduction announces that cells could cluster together into groups.



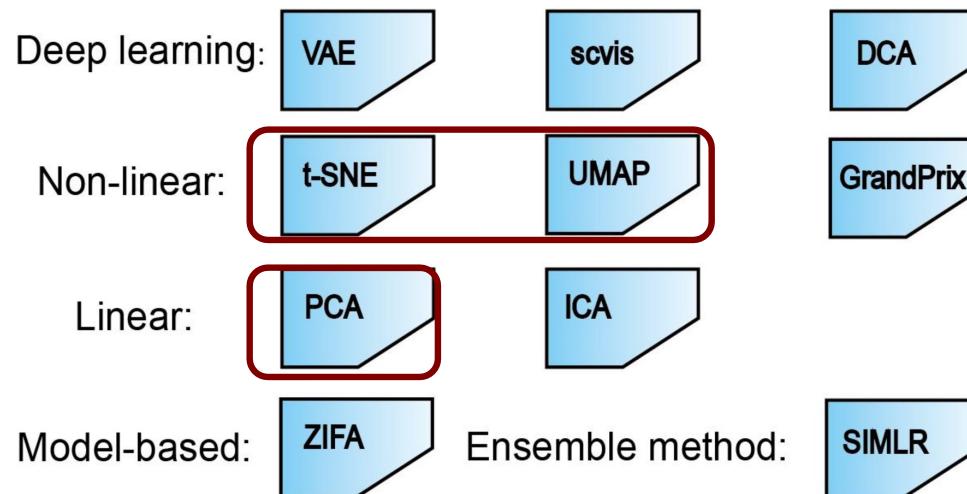
<https://doi.org/10.3389/fgene.2021.646936>

<https://www.biorxiv.org/content/10.1101/241646v1.full>

Luecken, MD and Theis, FJ. Current best practices in single-cell RNA-seq analysis: a tutorial, Mol Syst Biol 2019 (doi: <https://doi.org/10.15252/msb.20188746>)

Dimensionality Reduction

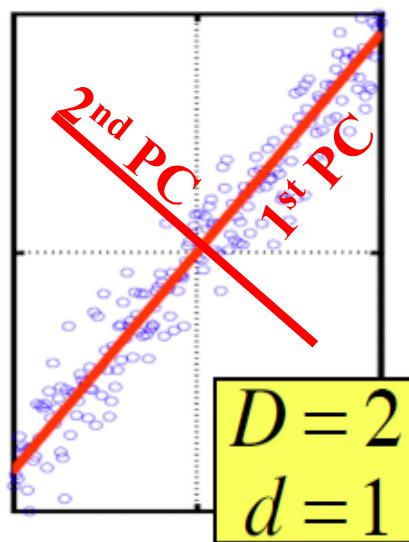
- Reduce noise, sparsity
- Easier for visualization and processing
- Linear methods:
 - PCA (principal component analysis)
- Non-linear methods:
 - t-SNE
 - UMAP



PCA

Principal component analysis

- It is a linear algebraic method of dimensionality reduction
- Any matrix can be decomposed as a multiplication of other matrices
- PC1 represents the most of variance for the data, then PC2, PC3

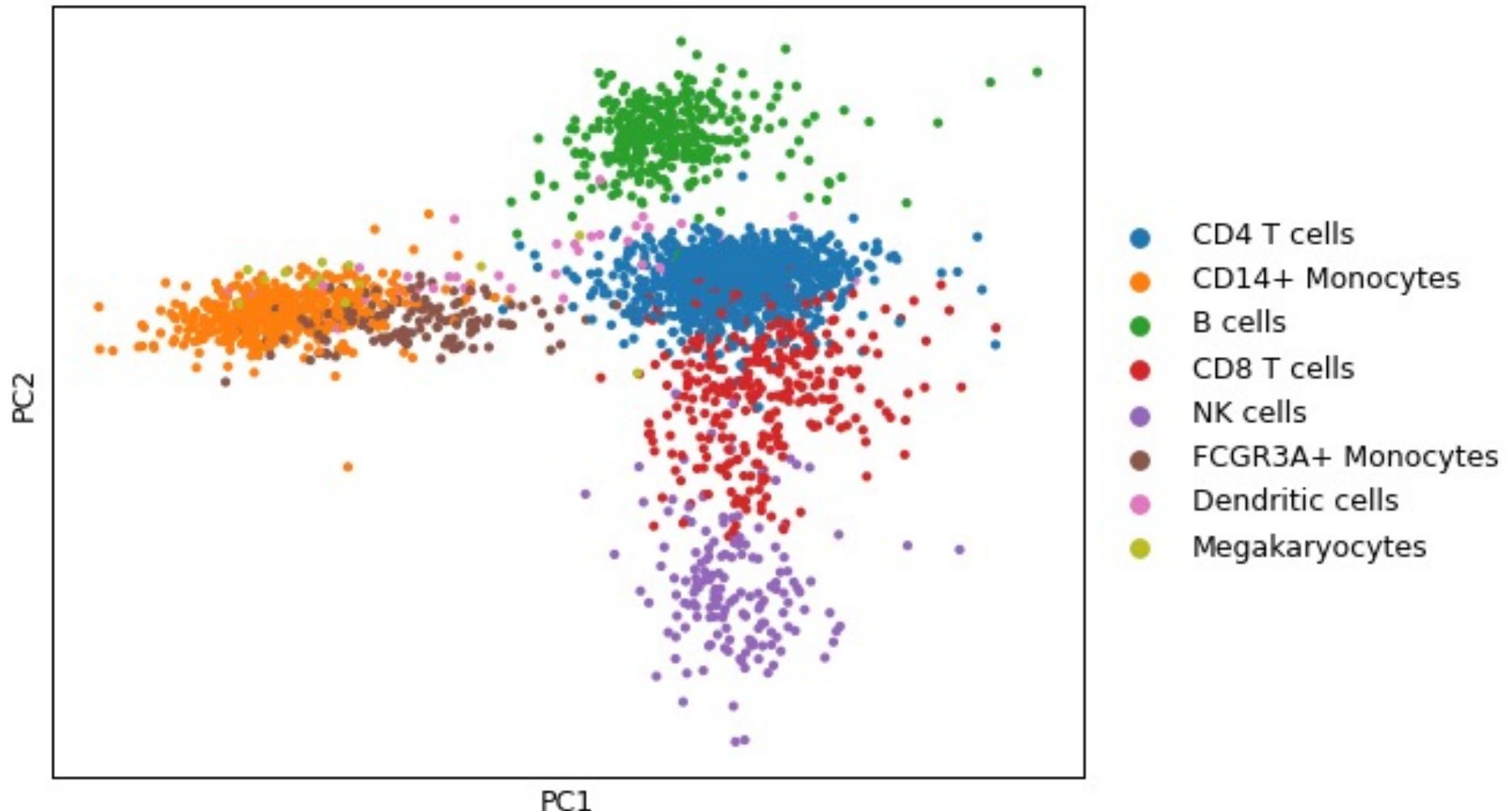


Reduce D genes to d PCs of cells, where $d \ll D$



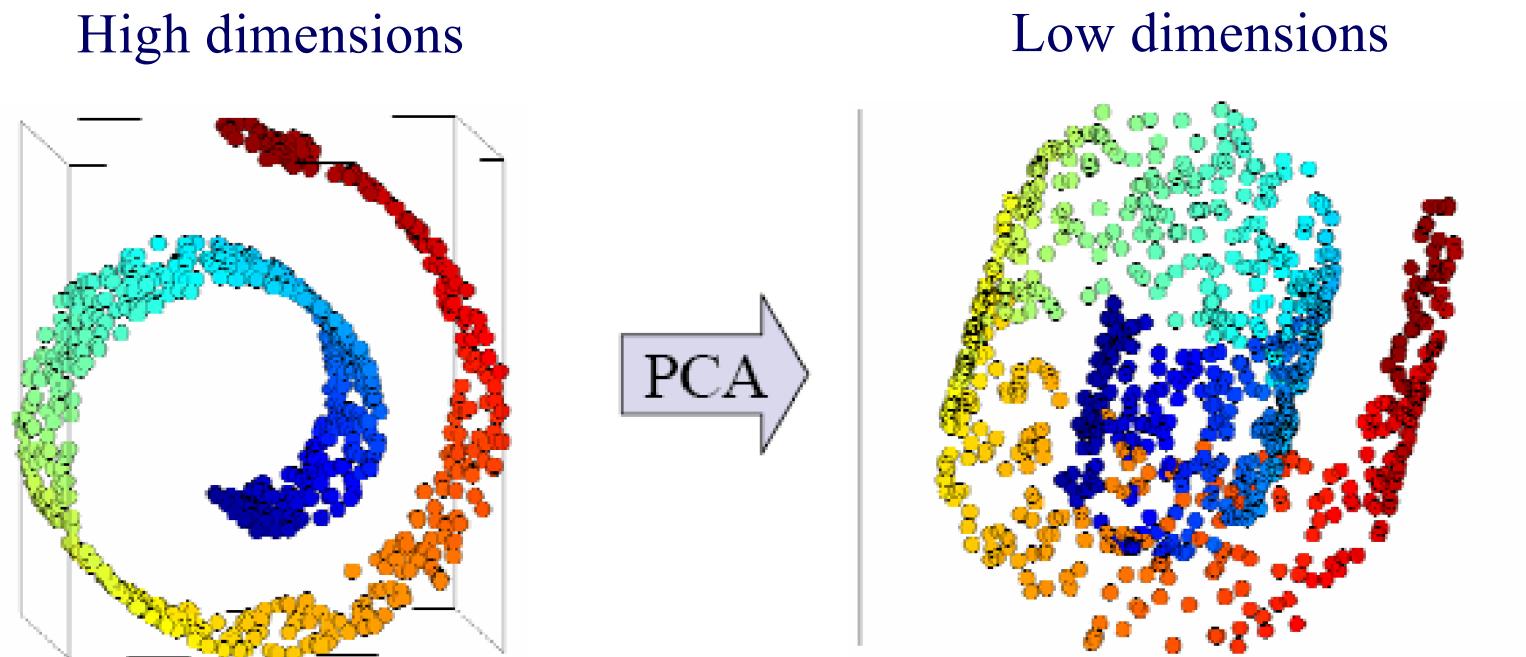
```
sc.pp.pca(adata, n_comps=50)
```

PCA of Peripheral Blood Mononuclear Cells (PBMC)



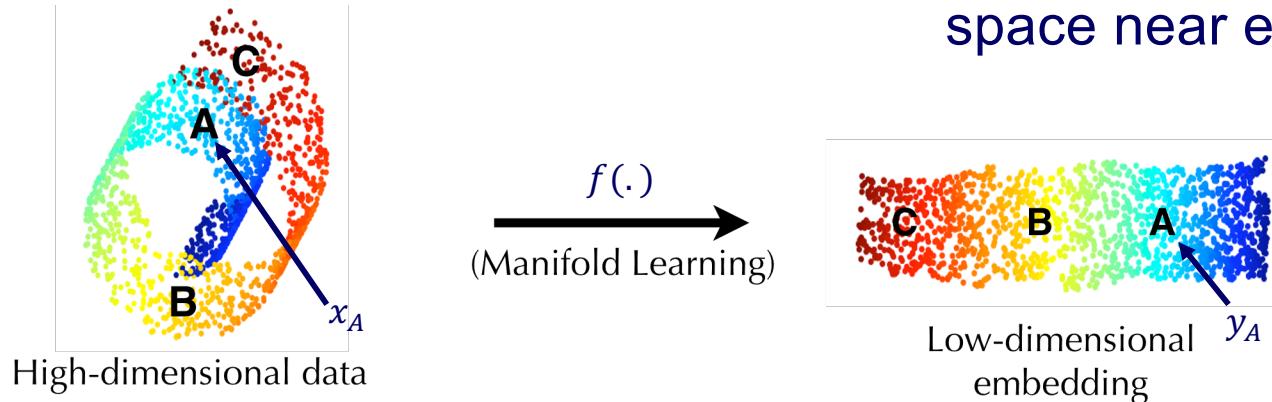
Limitation of PCA

- PCs represent linear combination of individual features (e.g., genes)
- PCA fails to find non-linear structure in the data



Manifold learning

- A manifold is a topological space that locally resembles Euclidean space near each point



A d dimensional manifold M is embedded in a m dimensional space, and there is an explicit mapping $f: \mathbb{R}^m \rightarrow \mathbb{R}^d$ where $d \leq m$.

Given Sample $x_i \in \mathbb{R}^m$ with noise

$$y_i = f(x_i)$$

→ find $f(\cdot)$ or y_i from given x_i

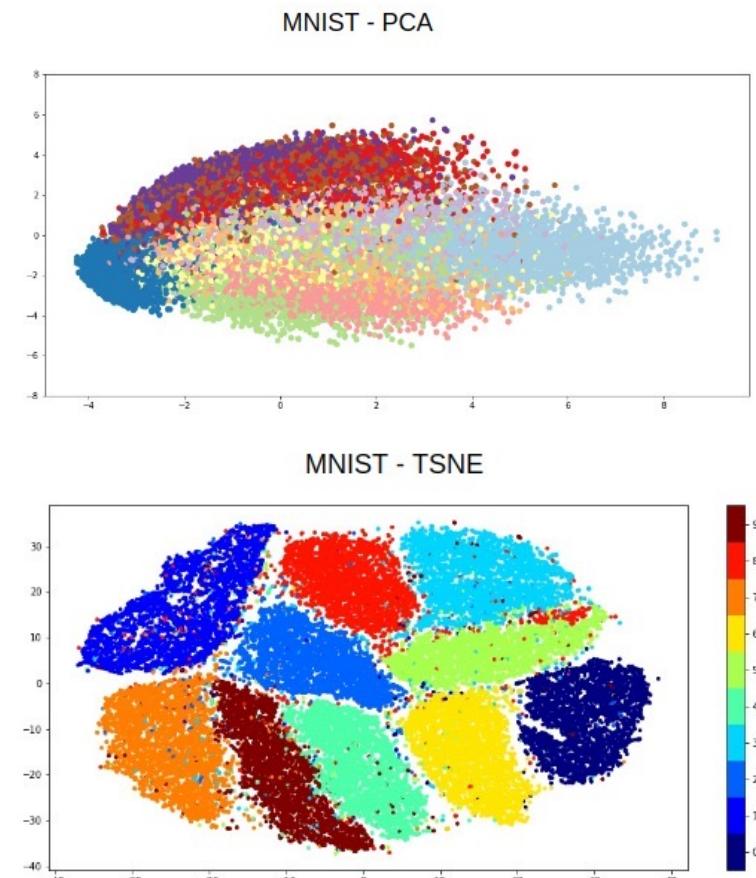
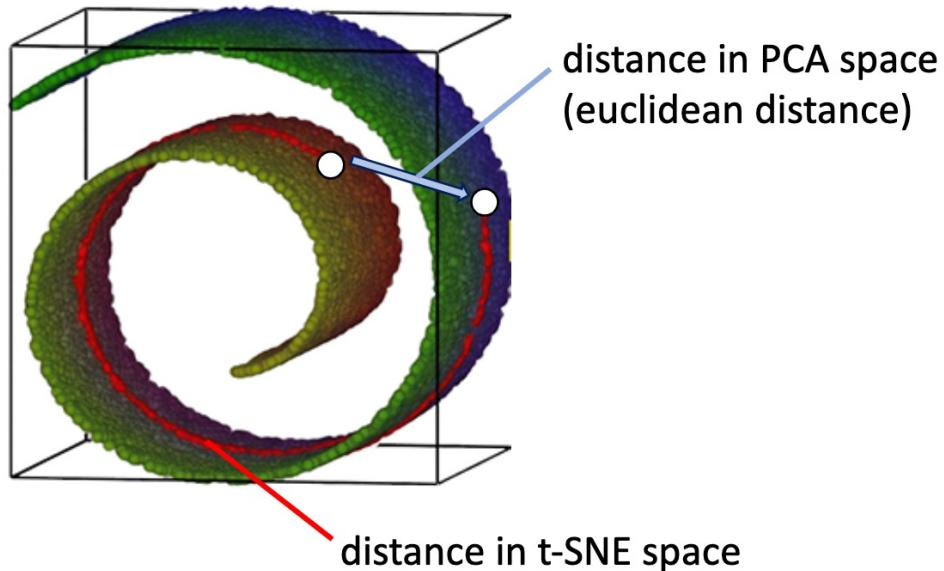
Parametric & generalizable (e.g., linear manifold learning)

Nonparametric & nongeneralizable (e.g., nonlinear manifold learning)

t-SNE

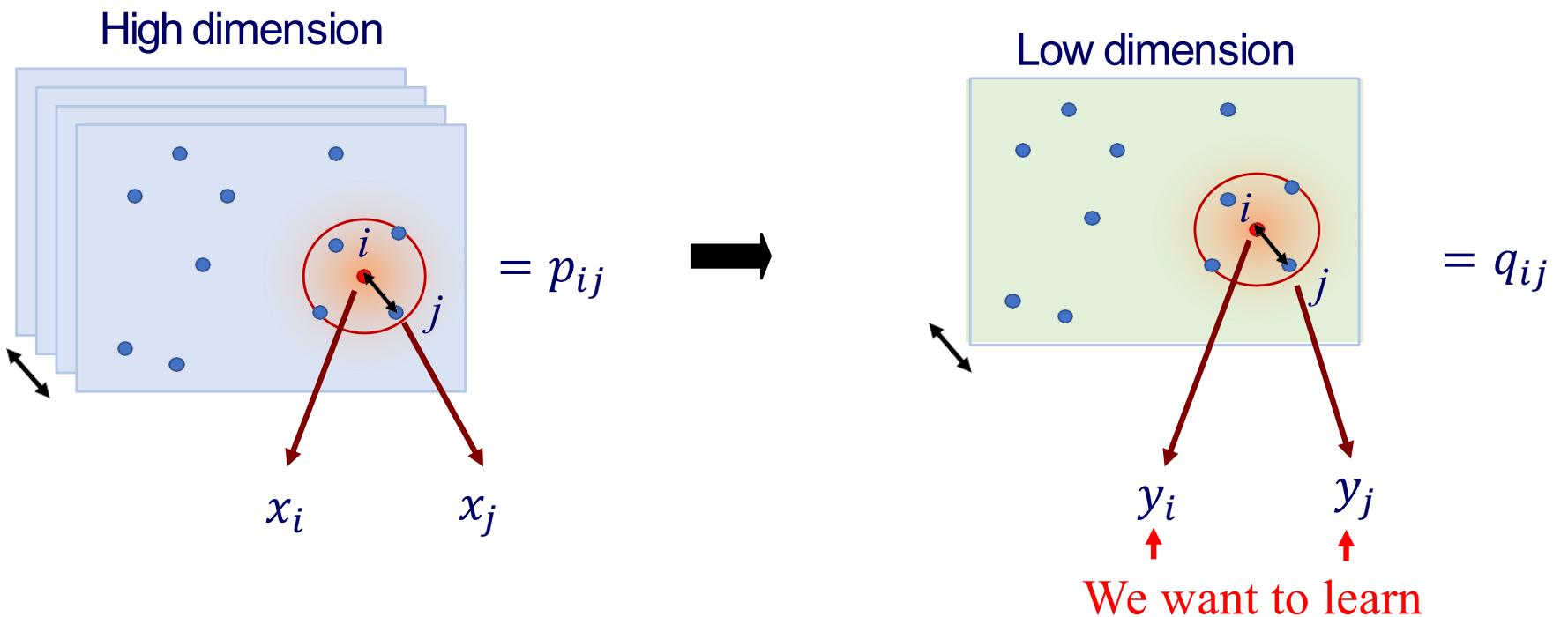
t-Stochastic Neighbor Embedding

- t-SNE is a manifold embedding algorithm for nonlinear dimensionality reduction ("keep manifold structures on low dimension space")



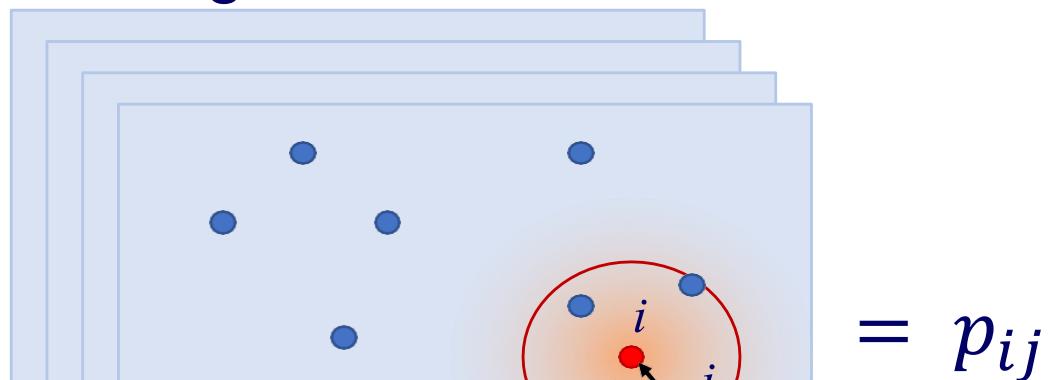
t-SNE

- Given a collection of points $X = \{x_1, \dots, x_n\} \subset R^d$, find a collection of points $Y = \{y_1, \dots, y_n\} \subset R^{d'}$, where $d \gg d'$. p_{ij} and q_{ij} measure the conditional probability that a point j would pick point i as its nearest neighbor, in high (p) and low (q) dimensional space, respectively.



Similarity matrix at high dimension

High dimension



$$= p_{ij}$$

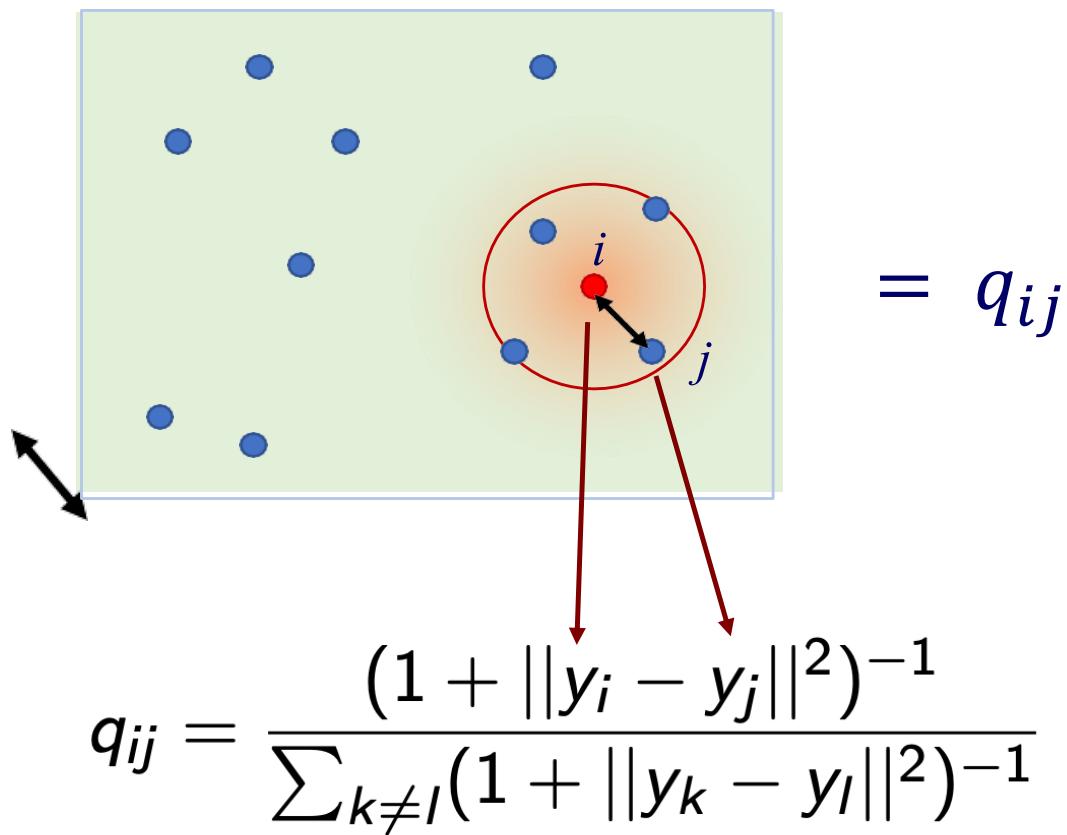
$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\tau_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\tau_i^2\right)}$$

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

where τ_i^2 is the variance for the Gaussian distribution centered around x_i

Similarity matrix at low dimension

Low dimension



Cost Function

Kullback-Leibler divergence

- Find optimal $\{y_i\}$: Minimize a single Kullback-Leibler divergence between a joint probability P , in the high-dimensional space and a joint probability Q , in the low-dimensional space.

$$C = \sum_i KL(P_i || Q_i) = \sum_{i=1}^n \sum_{j=1}^n p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

- The gradient is given by

$$\frac{dC}{dy_i} = 4 \sum_{j=1, j \neq i}^n (p_{ij} - q_{ij})(1 + \|y_i - y_j\|^2)^{-1}(y_i - y_j)$$

- The result of this optimization is a map that reflects the similarities between the high-dimensional inputs.

Summary of t-SNE

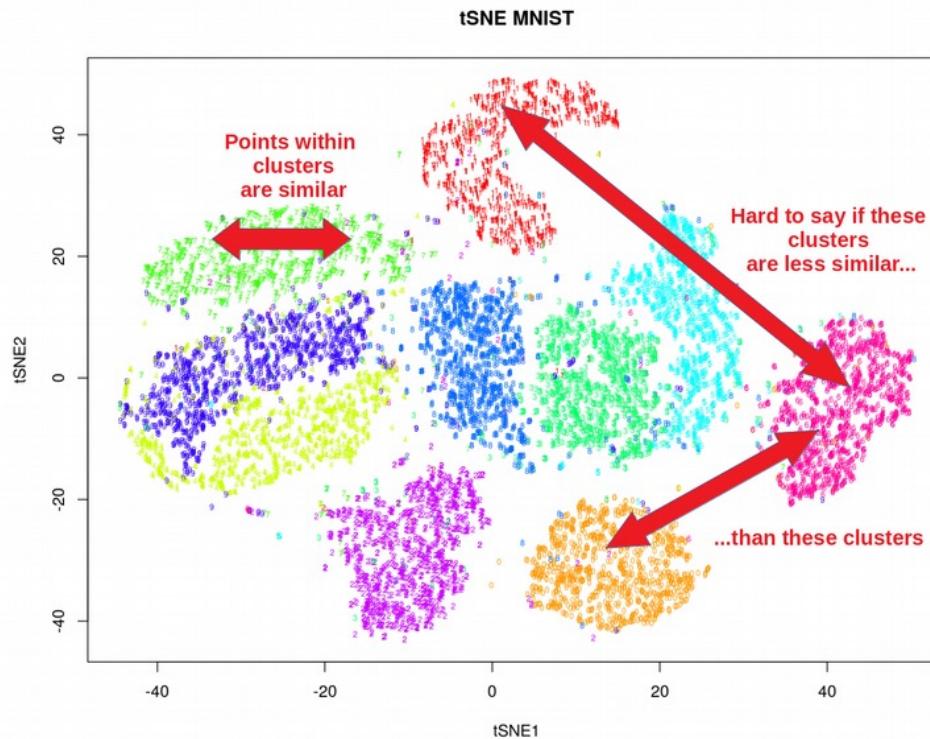
- t-SNE minimizes the divergence between two similarity distributions:
 - pairwise similarities of the input data points on the high dimensional space $\{p_{ij}\}$
 - pairwise similarities of the corresponding low-dimensional points $\{q_{ij}\}$

Main steps for t-SNE

1. Construct a similarity matrix over pairs of high-dimensional points
 - Similar data points are assigned a higher probability, while dissimilar points are assigned a lower probability.
2. Define a similarity matrix in the low-dimensional embedding space
3. Minimize the KL divergence between two similarity distributions using gradient descent to find optimal low dimensional coordinates of data points

Limitation of t-SNE

- t-SNE mainly preserves local similarity structure of data, not global similarity.
- Only within cluster distances are meaningful while between cluster similarities are not guaranteed.

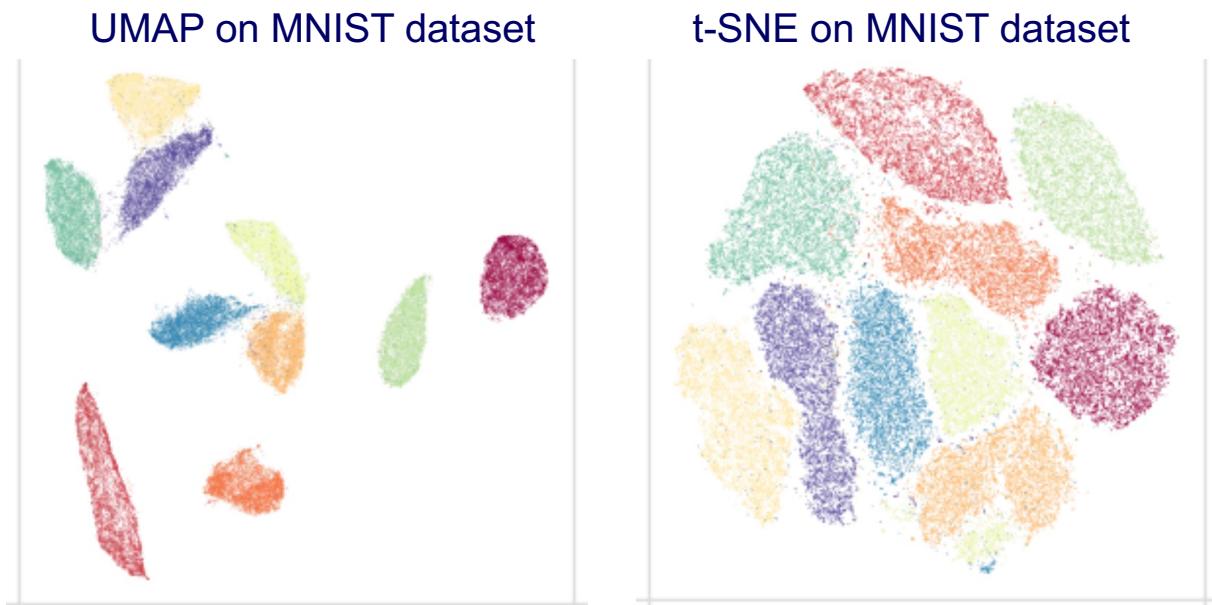


```
sc.tl.tsne(adata, use_rep="X_pca")
```

UMAP

Uniform Manifold Approximation and Projection

- Instead of randomly assigning *points* as in t-SNE, UMAP constructs a high dimensional *graph representation* of the data and then optimizes a low-dimensional graph to be as structurally similar as possible
- UMAP could preserve both global and local data structures



- <https://www.youtube.com/watch?v=nq6iPZVUxZU>
- https://nbisweden.github.io/excelerate-scRNAseq/session-dim-reduction/lecture_dimensionality_reduction.pdf
- https://umap-learn.readthedocs.io/en/latest/how_umap_works.html
- <https://pair-code.github.io/understanding-umap/>

Main steps for UMAP

1. Construct a weighted k nearest neighbors graph for defining a similarity matrix in high-dimensional space
 - The edge weights represent the likelihood that two data points are connected (i.e., “similarity”)
2. Define a similarity matrix in low-dimensional space
3. Minimize the cross-entropy cost function at each point using gradient descent to find the most *similar graph* in lower dimensions

Similarity matrix at high dimension

- ρ_i , the distance from x_i to the k nearest neighbors in high dimension (e.g., kNN graph)

$$\rho_i = \min\{d(x_i, x_{i_j}) \mid 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\}.$$

- UMAP uses an exponential probability distribution in high dimension

$$p_{j|i} = e^{-\frac{d(x_i, x_j) - \rho_i}{\sigma_i}}$$

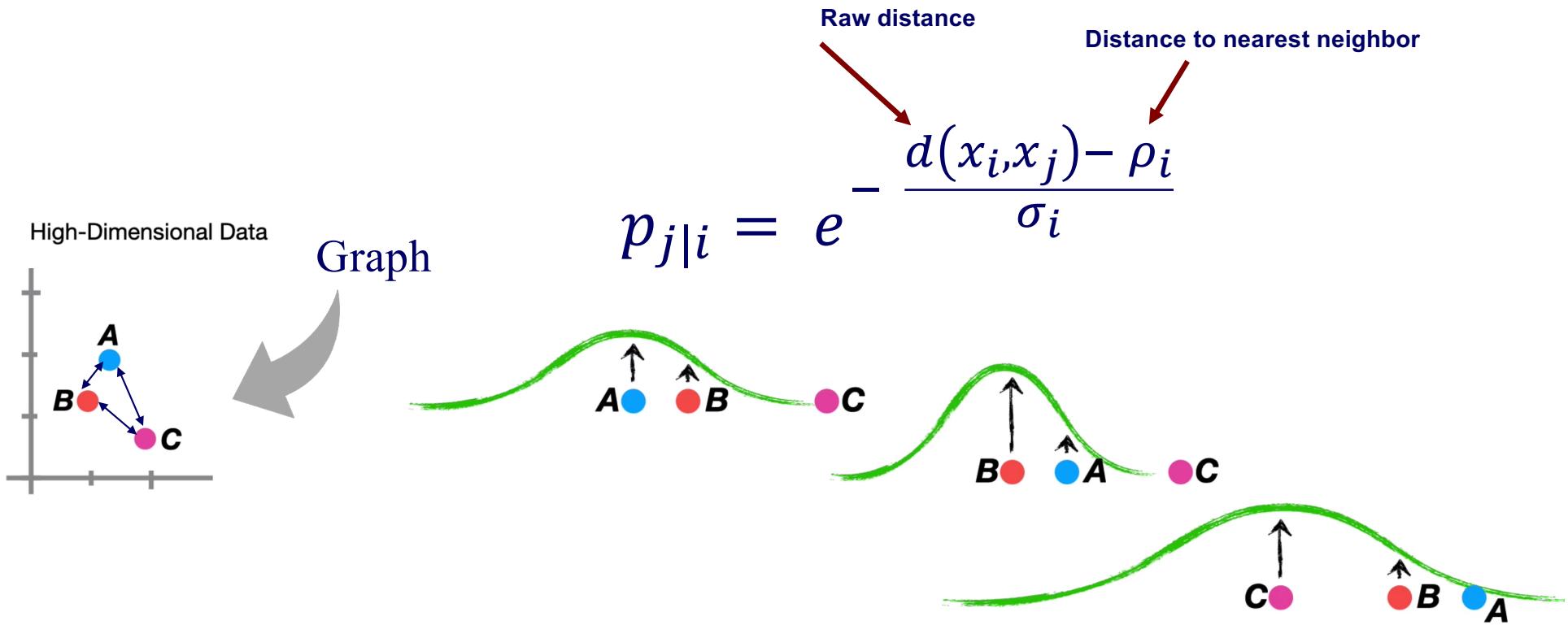
Raw distance Distance to nearest neighbor

- The symmetrization of high dimension similarity score and the definition of number of nearest neighbors

$$p_{ij} = p_{i|j} + p_{j|i} - p_{i|j}p_{j|i}$$

Similarity matrix at high dimension

- For each data point, it has a locally adaptive exponential kernel, so the distance metric varies from point to point.



Cost Function

binary cross-entropy (CE)

- Similarity matrix at low dimension

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

the family of curves $1/a \times y^{2b}$ for modelling distance probabilities in low dimensions, a and b are hyperparameters

- Cross-entropy cost function makes UMAP to capture the global and local data structures

$$CE(X, Y) = \sum_i \sum_j \left[p_{ij}(X) \log\left(\frac{p_{ij}(X)}{q_{ij}(Y)}\right) + (1 - p_{ij}(X)) \log\left(\frac{1 - p_{ij}(X)}{1 - q_{ij}(Y)}\right) \right]$$

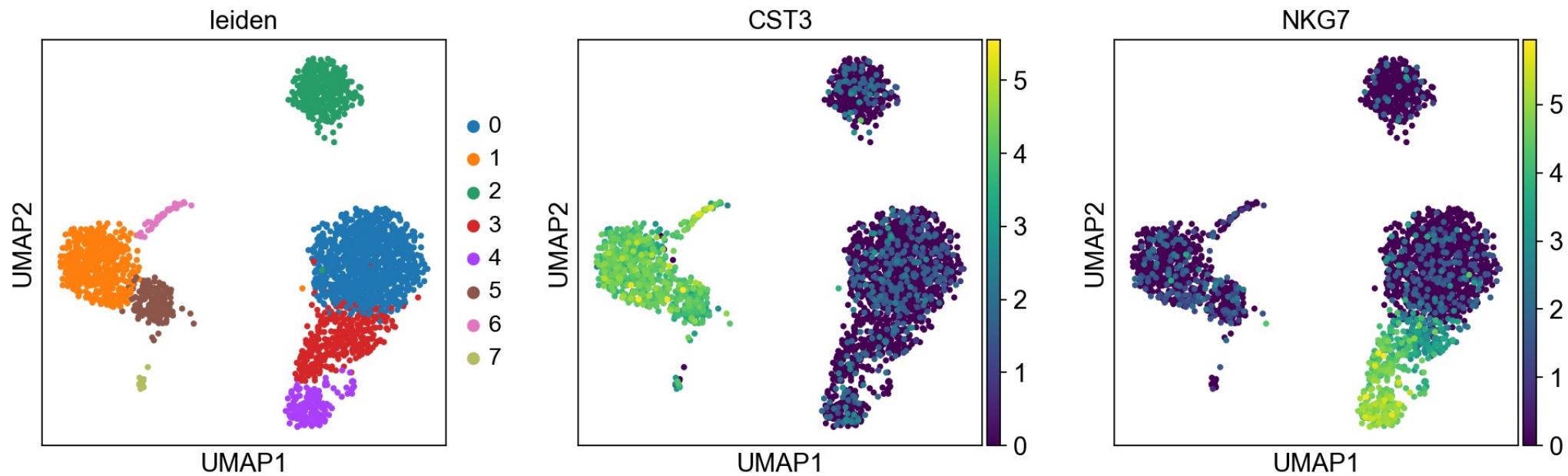
- The gradient of the CE to find optimal $\{y_i\}$

$$\frac{\delta CE}{\delta y_i} = \sum_j \left[\frac{2abd_{ij}^{2(b-1)}P(X)}{1 + ad_{ij}^{2b}} - \frac{2b(1 - P(X))}{d_{ij}^2(1 + ad_{ij}^{2b})} \right] \text{ where } d_{ij} = y_i - y_j$$

UMAP examples



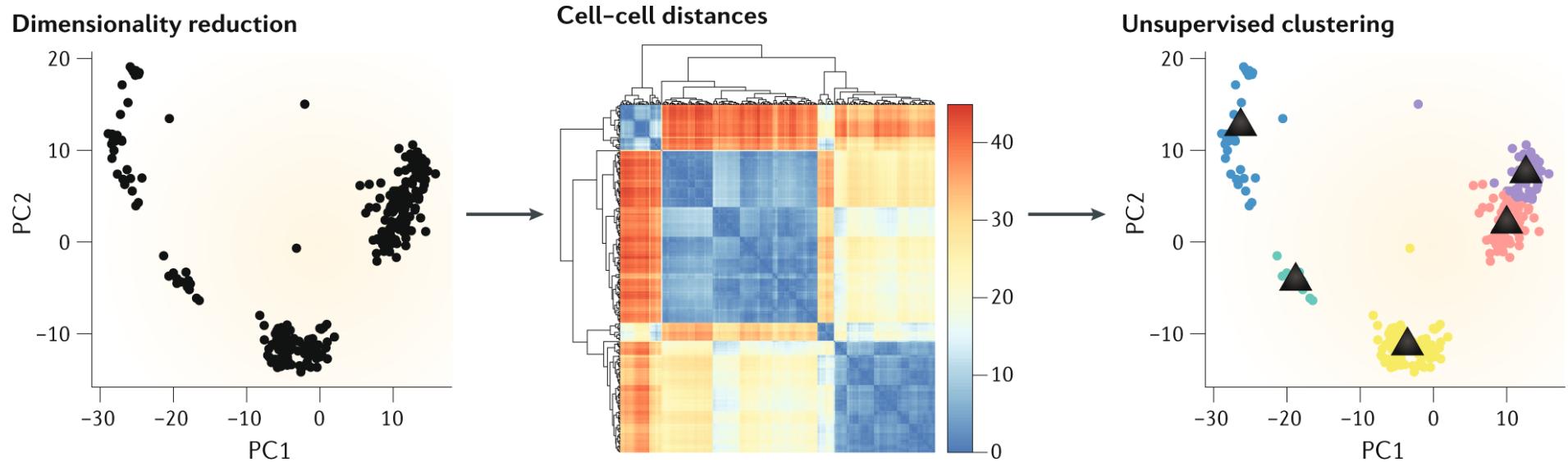
```
sc.pl.umap(adata, color=['leiden', 'CST3', 'NKG7'])
```



Single-cell RNA sequencing analysis

- Introduction of single cell analysis
- Dimensional reduction
 - Non-linear: t-SNE, UMAP
 - Linear: PCA
- **Cell clustering**
 - K-means
 - hierarchical clustering
 - graph-based clustering

Cell clustering

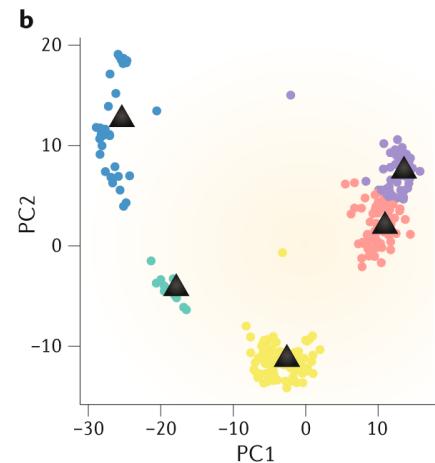


- Feature selection and dimensionality reduction extract the most informative genes and represented features from background noise, respectively
- Cell–cell distances are then calculated in the low dimensional space for clustering cells to clusters

Clustering methods for scRNA-seq

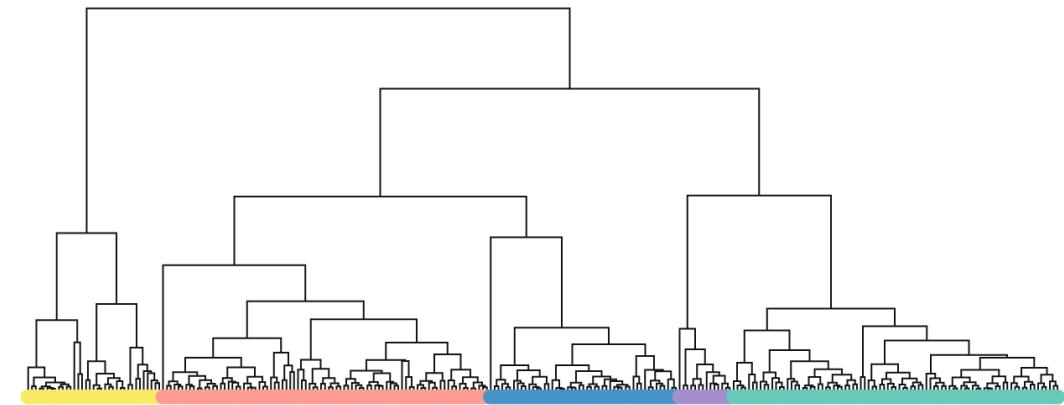
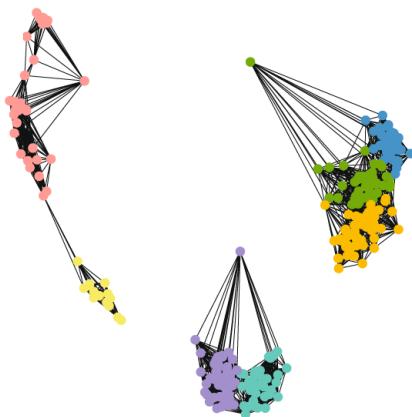
Clustering methods:

- K-means
- hierarchical clustering
- **graph-based clustering**



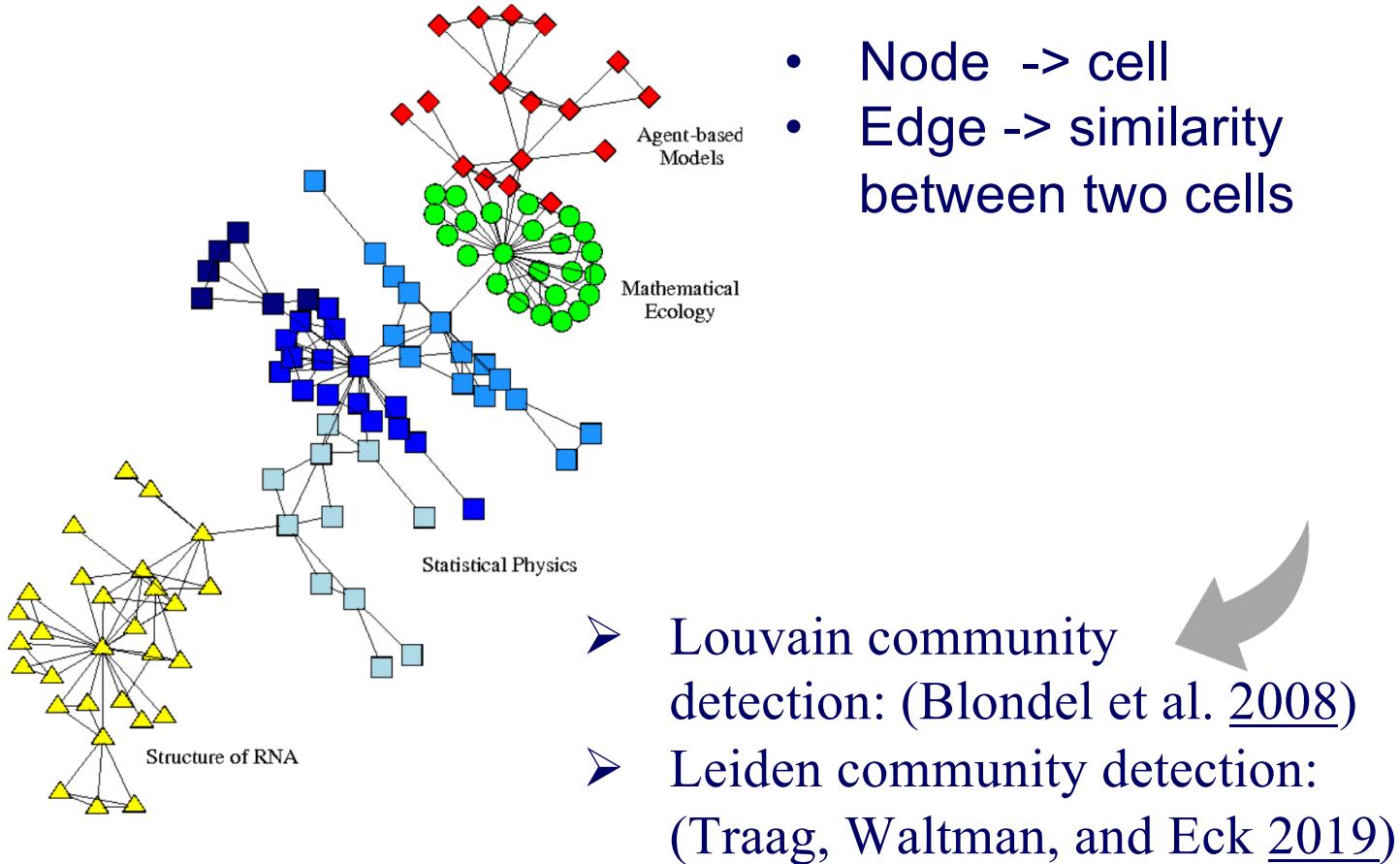
Tools for graph-based clustering:

- Seurat: Louvain, Leiden, SLM
- igraph:fast greedy, Louvain, optimal, walktrap, spinglass, infomap



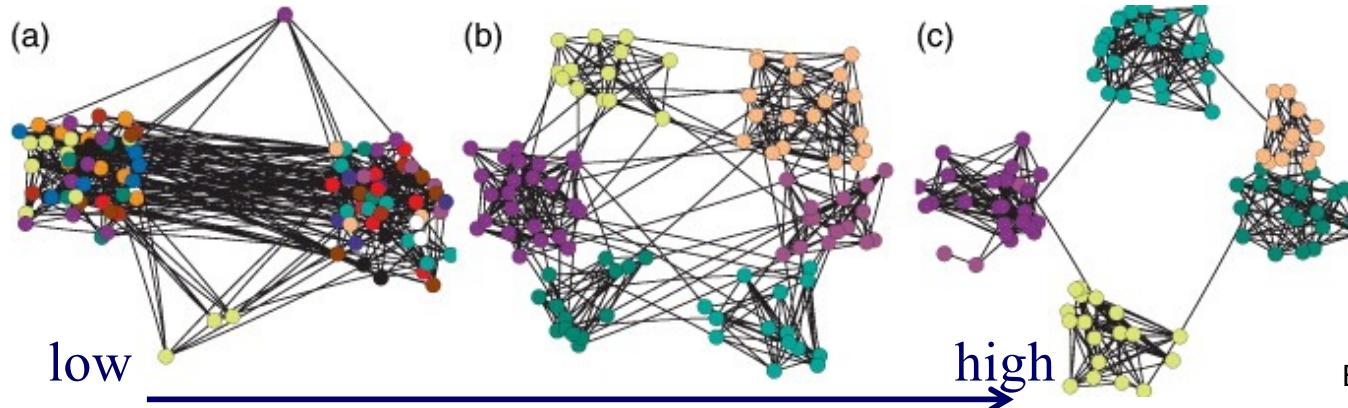
Graph-based clustering

- One of most popular clustering algorithms in scRNA-seq data analysis



- Freytag, Saskia, Luyi Tian, Ingrid Lönnstedt, Milica Ng, and Melanie Bahlo. 2018. “Comparison of Clustering Tools in R for Medium-Sized 10x Genomics Single-Cell Rna-Sequencing Data.” *F1000Research* 7. Faculty of 1000 Ltd.
- Wasserman, S. & Faust, K. (1994) Social Network Analysis (Cambridge Univ. Press, Cambridge, U.K.).
- https://biocellgen-public.svi.edu.au/mig_2019_scrnaseq-workshop/public/clustering-and-cell-annotation.html#ref-freytag2018comparison

Modularity



Brede, Europhysics Letters, 2010.

Modularity Q : measurement on strength of network division

$$Q = \frac{1}{2m} \sum_{i,j} \left(W_{ij} - \frac{k_i k_j}{2m} \right) \delta_{\sigma_i \sigma_j}$$

normalization
 m : total number of edges

edge weight between nodes i and j

$\frac{k_i k_j}{2m} = p_{ij}$ = expected edge weight that would go between i and j

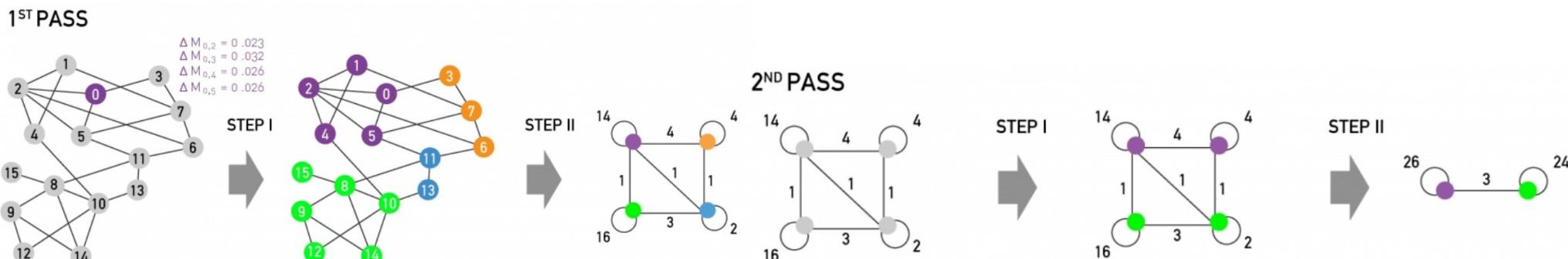
sum over nodes within a group (module)

Clustering goal: assign each node a module to maximize “modularity” as an objective function (module is a group of highly connected nodes)

Newman, PNAS, 2006.

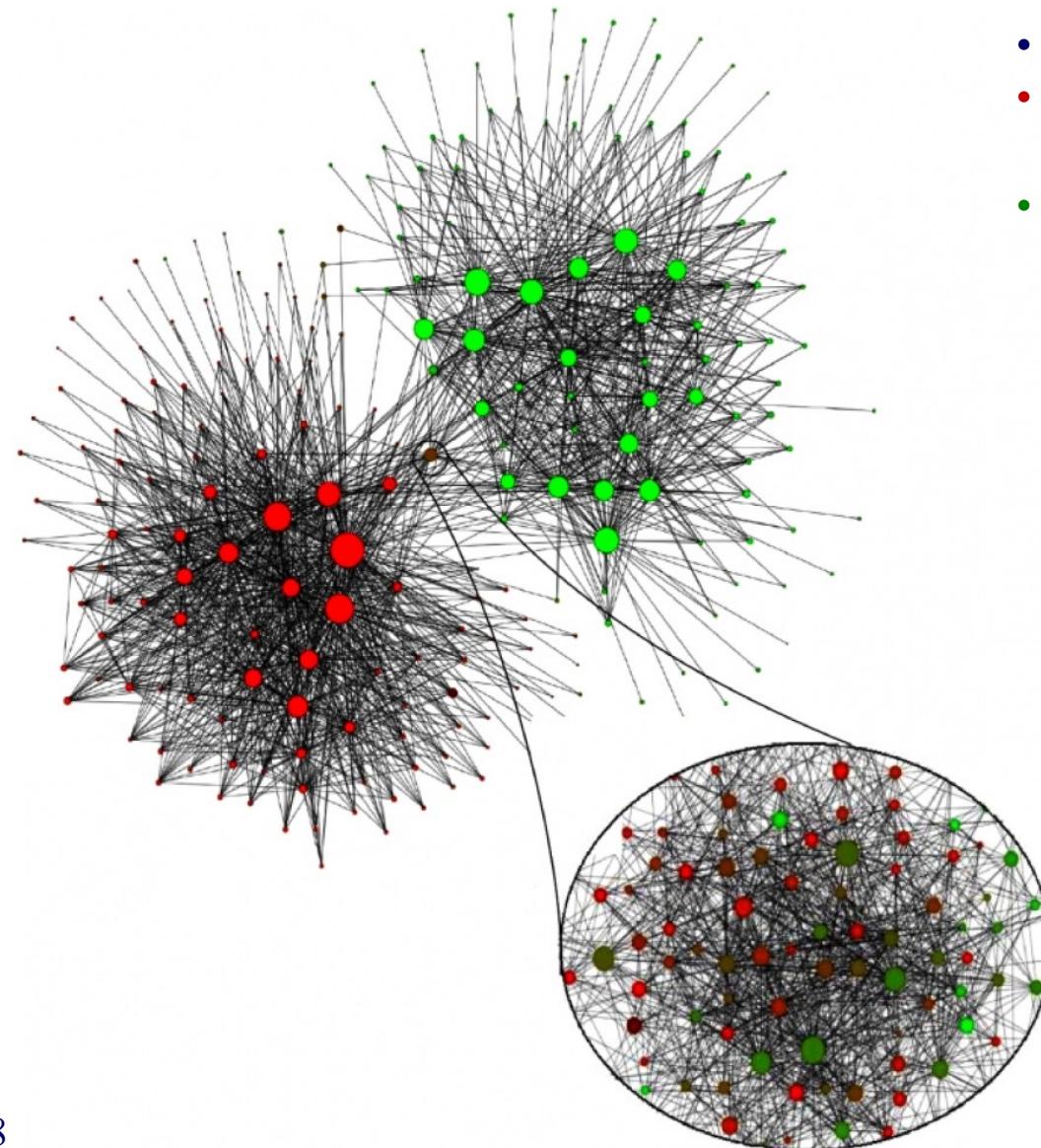
Louvain community detection

- Start with every node in its own community (i.e., module/cluster)
- Step1: Modularity optimization
 - Order the nodes and for each node i , move i to the community of neighbor j that leads to maximum ΔQ
 - If all $\Delta Q < 0$ the i remains in its current community
 - Repeatedly cycle through all nodes until $\Delta Q = 0$
- Step2 : Community aggregation
 - Create a weighted network of communities from Step1
 - Nodes : communities in Step1
 - Edge weights : sum of weights of edges between communities
 - Edges within a community become two self-loops
- Repeat: Apply Step1 & Step2 to resulting network, and so on until $\Delta Q = 0$



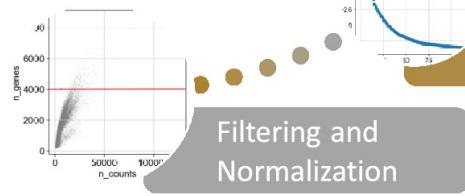
Examples of Louvain clustering

Belgian mobile phone network



- 2M nodes
- Red nodes: French speakers
- Green nodes: Dutch speakers

Examples of Louvain clustering

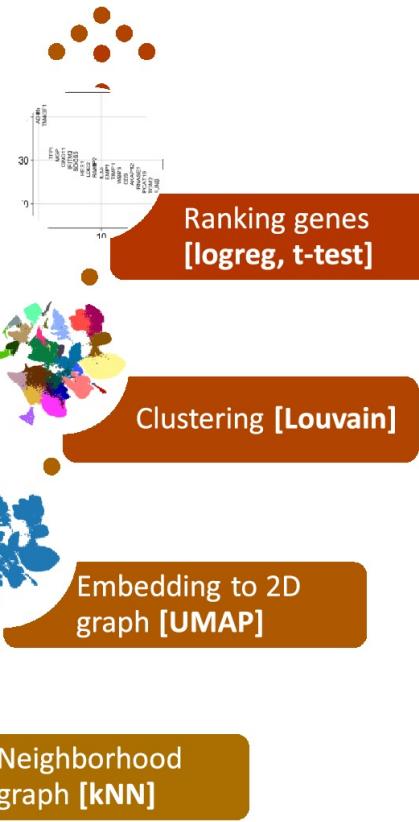


Filtering and
Normalization

Reduce the
dimensionality [PCA]



```
sc.tl.louvain(adata, resolution=1.0)
```



The Louvain algorithm clusters millions of cell with reasonable computational complexity.