# GraphDR: scaling to >10 million cells

$$Z = (I + \lambda L)^{-1} X W$$

Approximate nearest neighbor (ANN)

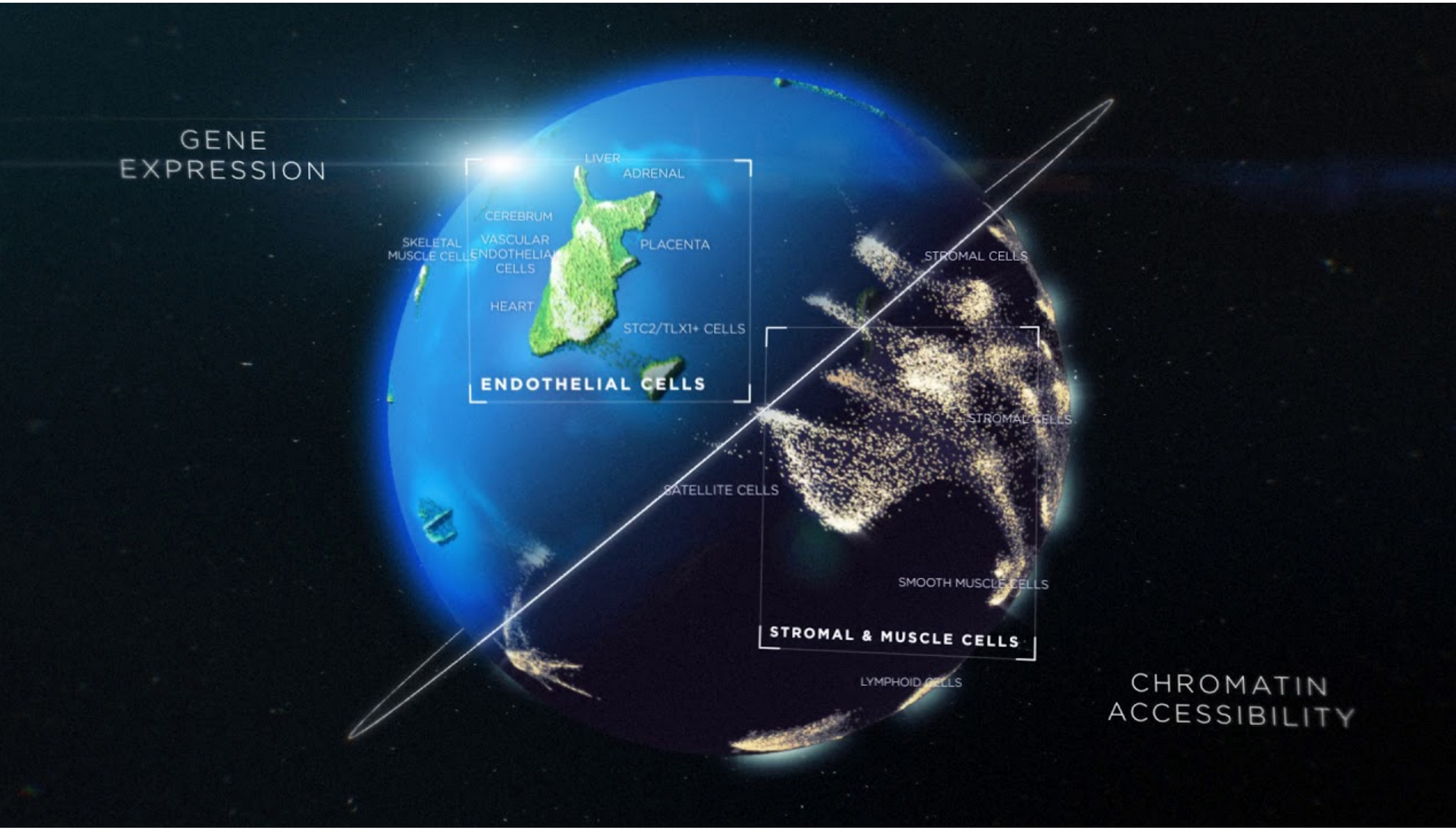
| | |
|---|---|
| Brute force | n^2 |
| KD-Tree / Ball-tree | n log n |
| ANN (**HNSW**, NN descent) | Close to O(n) |

Don’t do matrix inversion

$I + \lambda L$ is sparse,  the inverse of it is not sparse

**Solve the linear equation** $(I + \lambda L)Z = XW$ **with a sparse solver**

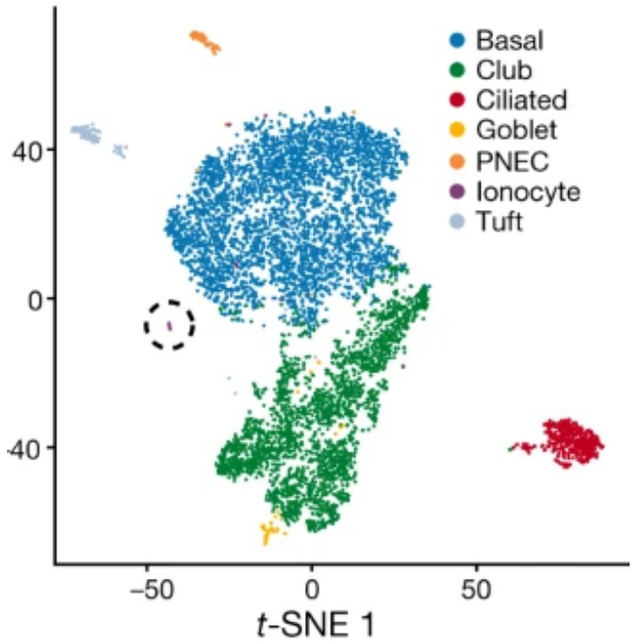
For further speed up, use GPU

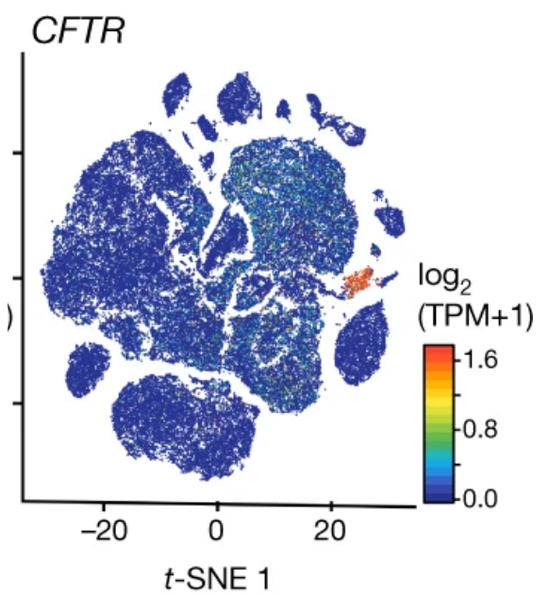

GENE EXPRESSION
LIVER
ADRENAL
CEREBRUM
SKELETAL MUSCLE CELLS
VASCULAR ENDOTHELIAL CELLS
PLACENTA
HEART
STC2/TLX1+ CELLS
ENDOTHELIAL CELLS
STROMAL CELLS
STROMAL CELLS
SATELLITE CELLS
SMOOTH MUSCLE CELLS
STROMAL & MUSCLE CELLS
LYMPHOID CELLS
CHROMATIN ACCESSIBILITY

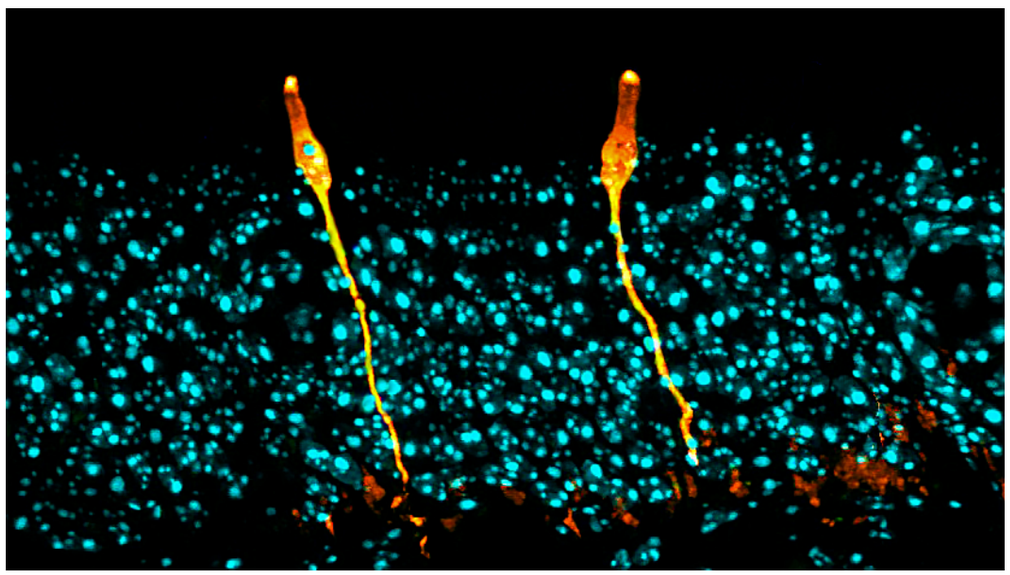
# Single-cell RNA-seq identified rare CFTR-expression cell types that is key to cystic fibrosis

Cystic fibrosis is caused by mutations from the CFTR gene, discovered in the 1980s

What cell type does CFTR function in?

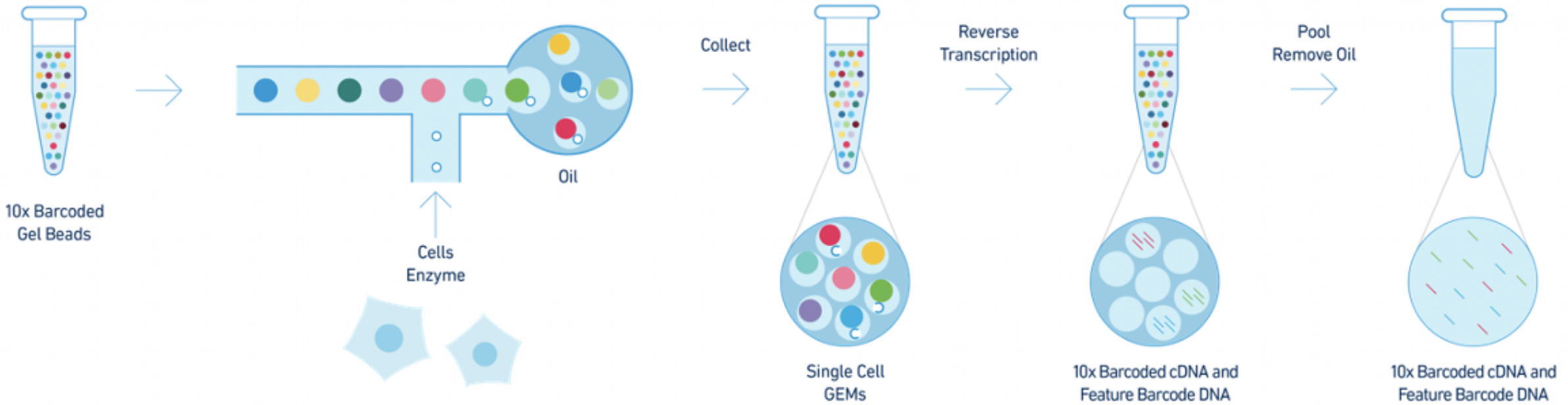


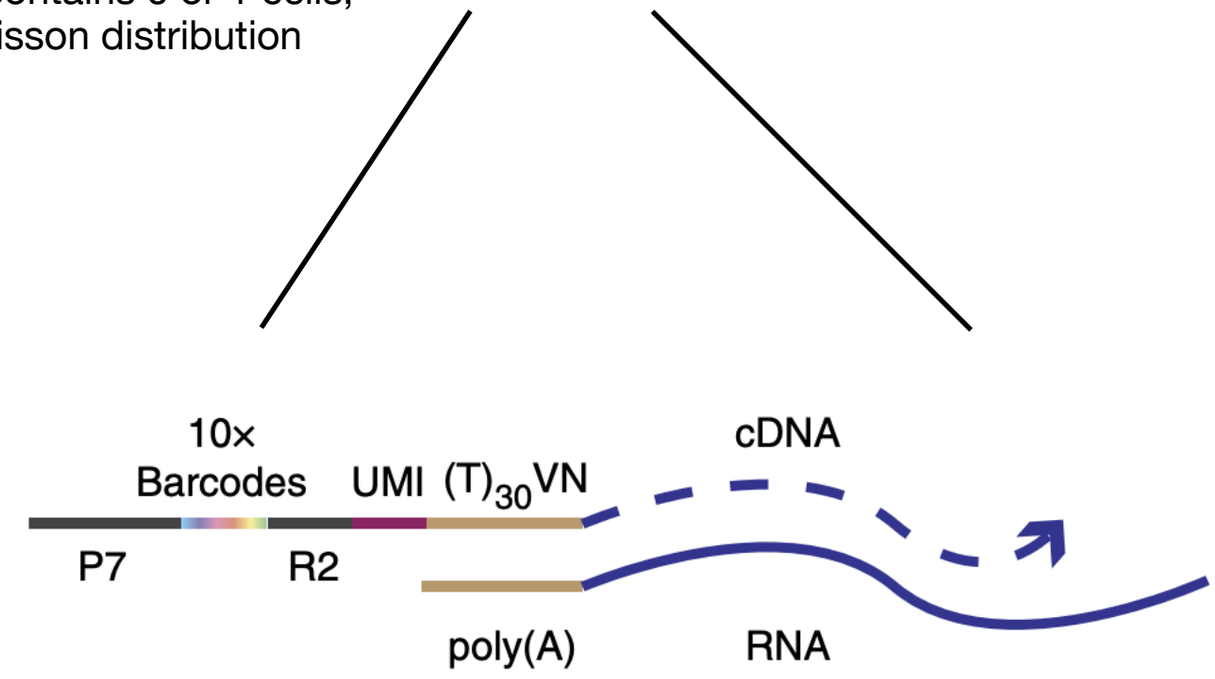



Plasschaert et al. 2018

# Understand the data: single cell RNA-seq

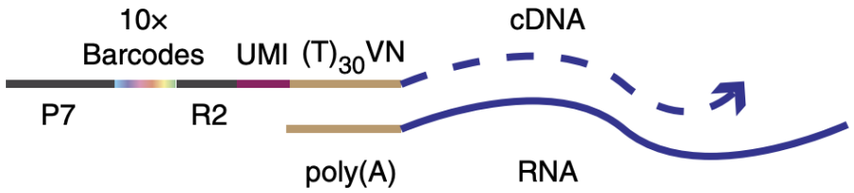


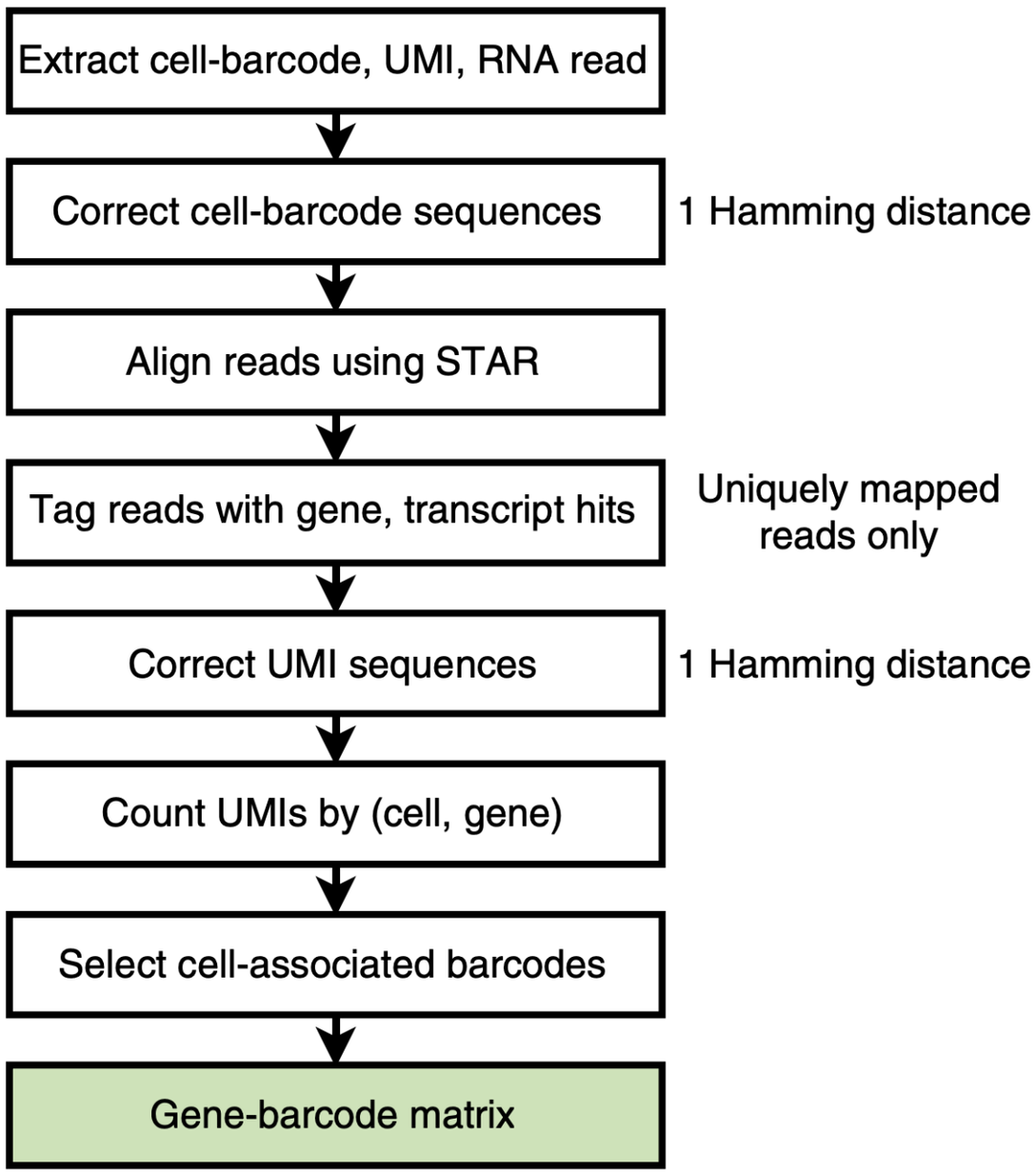

Most droplets contains 0 or 1 cells,
following Poisson distribution




Zheng et al, 2016, Massively parallel digital transcriptional profiling of single cells

# Understand the data: single cell RNA-seq




**Standard processing pipeline**

Filter out genes and cells with mostly zero counts

Filter out cells with high mitochondria gene content

Normalize by total UMI count per cell (library size)

Log-transformation with pseudo count log(x+1)

Optional: Standard scaling (zero mean and unit variance for each gene)

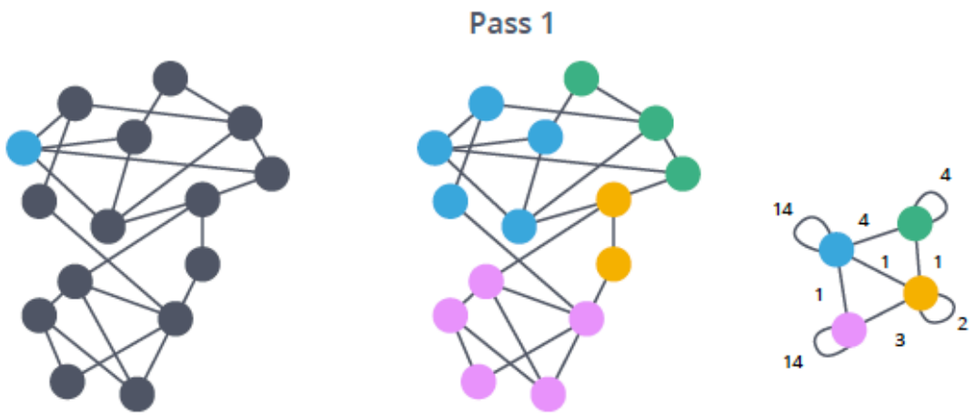
Zheng et al, 2016, Massively parallel digital transcriptional profiling of single cells

# Identify cell types: Graph-based clustering

**Use the topological structure of the data:**
Nearest neighbor graph / shared nearest neighbor graph is often good representation for identify clusters

https://github.com/vtraag/**louvain-igraph**

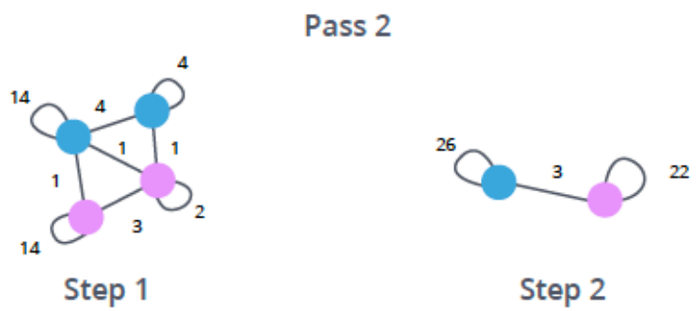
Modularity:
# of within-cluster edges -
expectation (based on product of node degrees)




Choose a start node and calculate the change in modularity that would occur if that node joins and forms a community with each of its immediate neighbors.

The start node joins the node with the highest modularity change. The process is repeated for each node with the above communities formed.

Communities are aggregated to create super communities and the relationships between these super nodes are weighted as a sum of previous links. (Self-loops represent the previous relationships now hidden in the super node.)

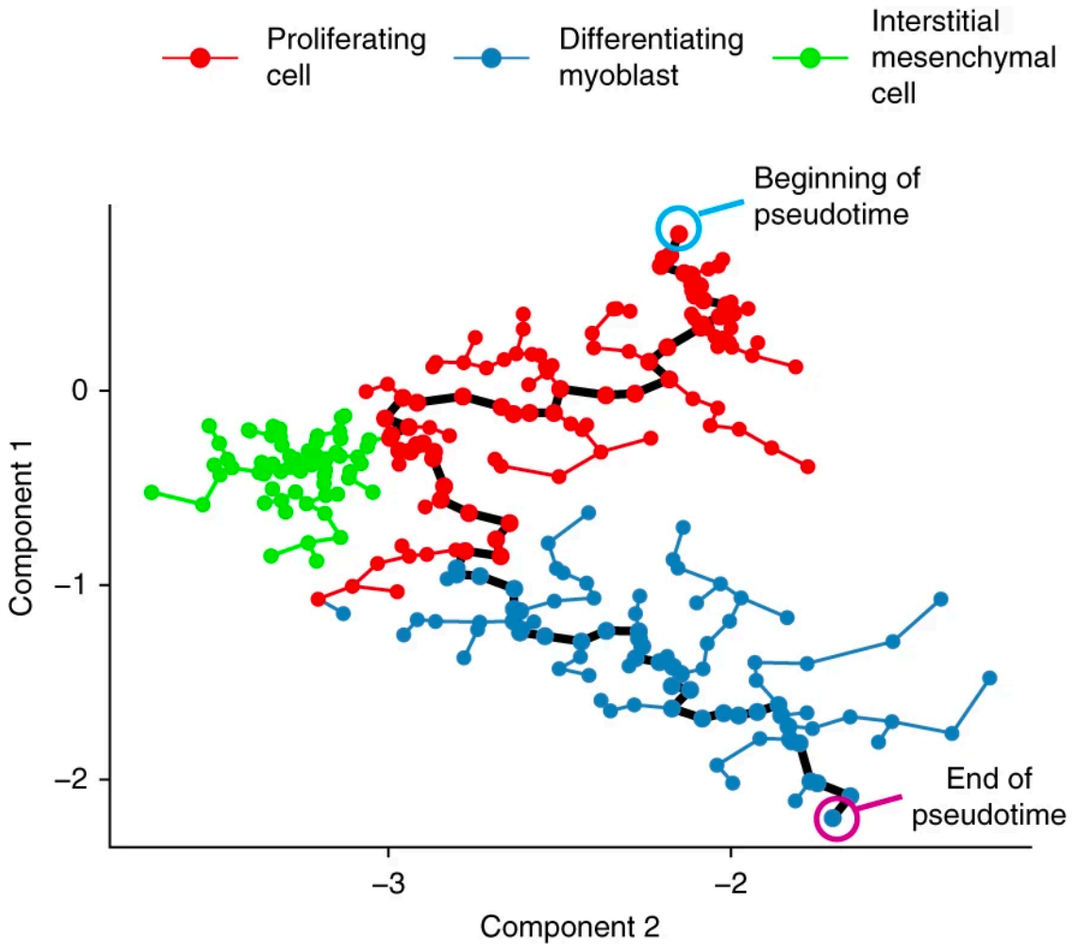


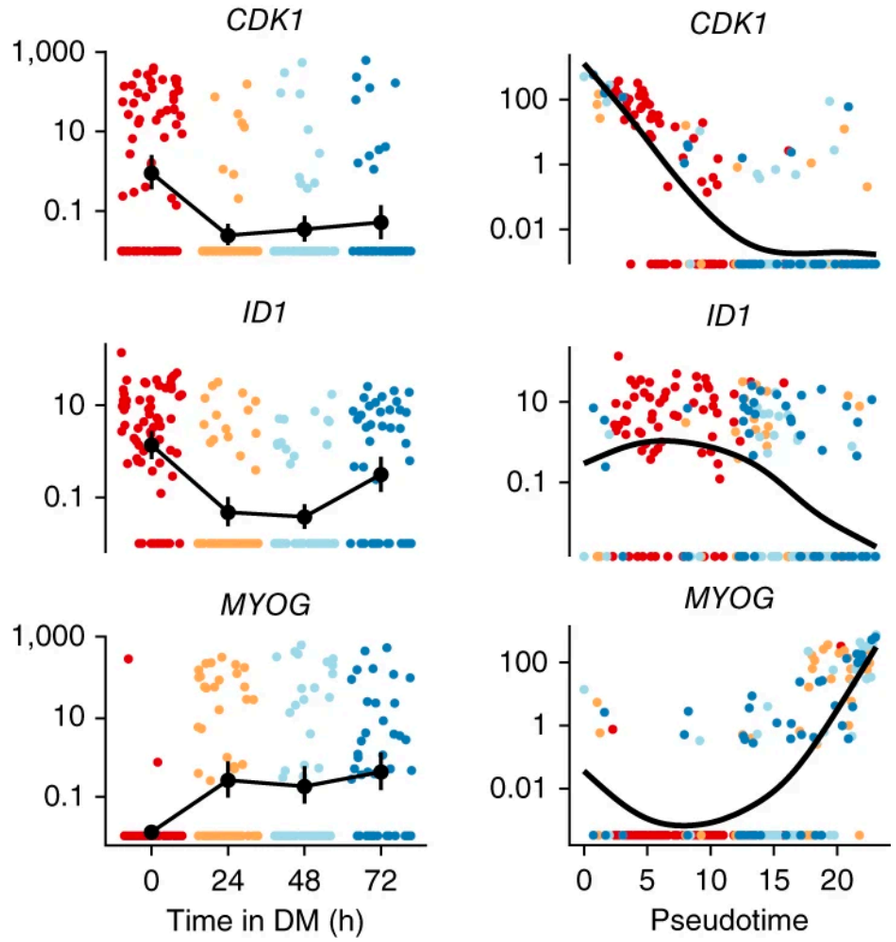

Steps 1 and 2 repeat in passes until there is no further increase in modularity or a set number of iterations have occurred.

***Louvain Modularity Algorithm***

https://neo4j.com/blog/graph-algorithms-neo4j-louvain-modularity/

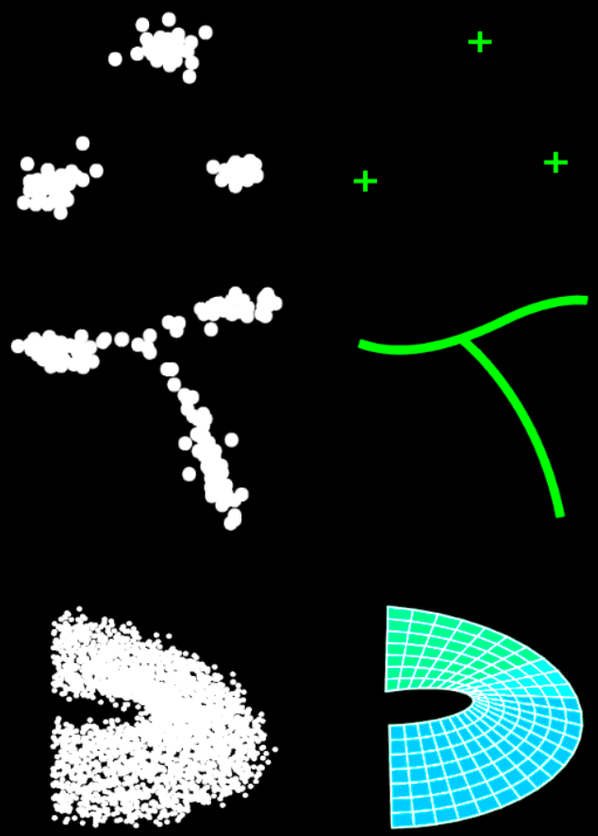
# Order cells by developmental state: pseudotime / trajectory analysis




Trapnell, C. et al. (2014)

# StructDR - unified structure extraction and inference of confidence set



Cluster

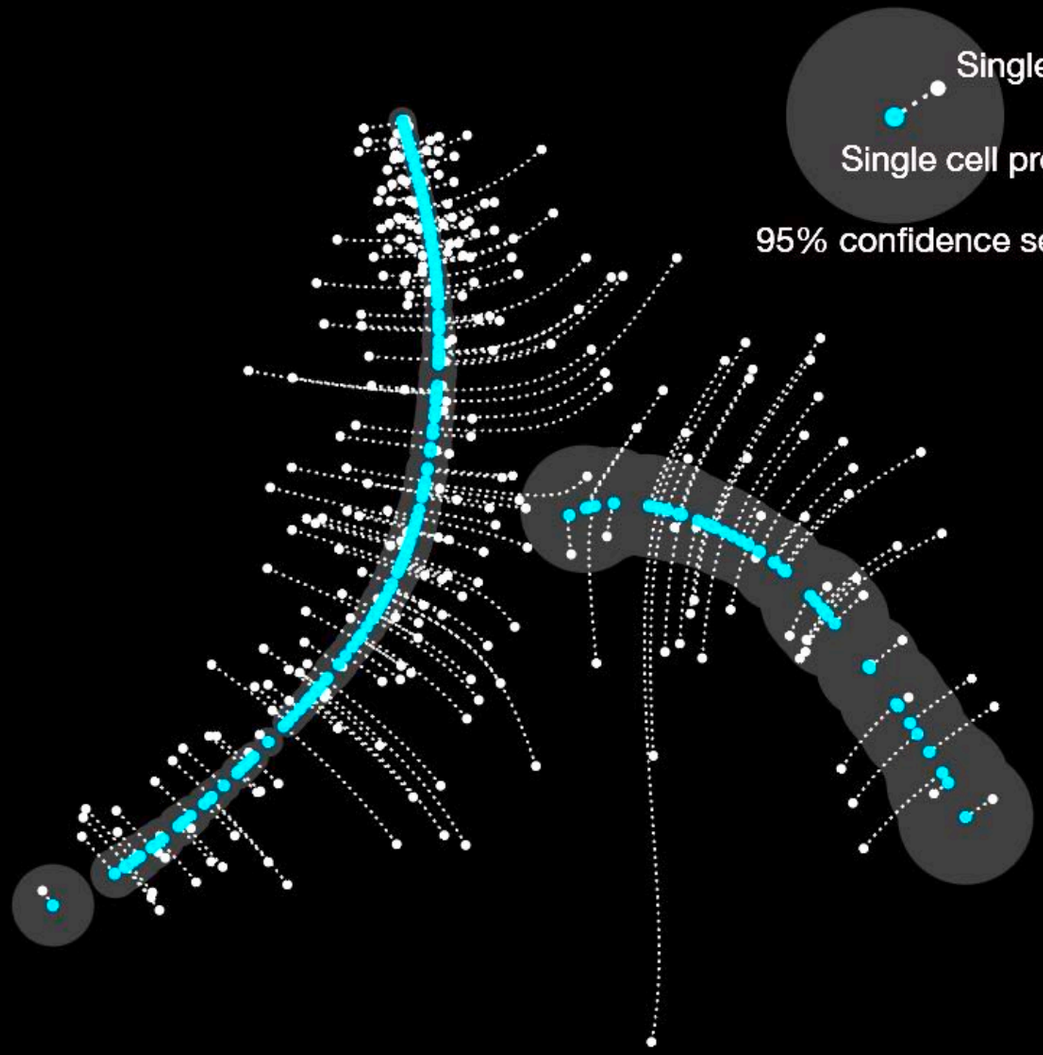
Cell type / state

Trajectory

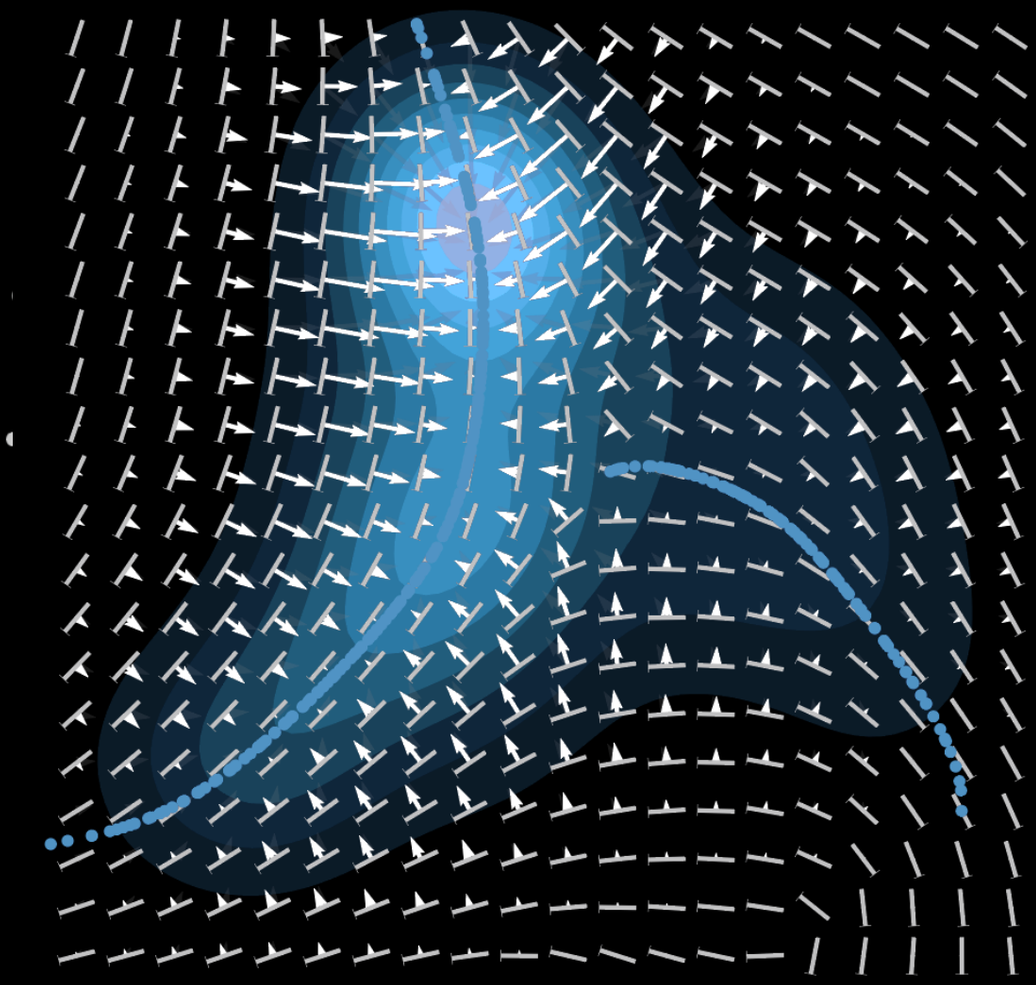
Developmental trajectories

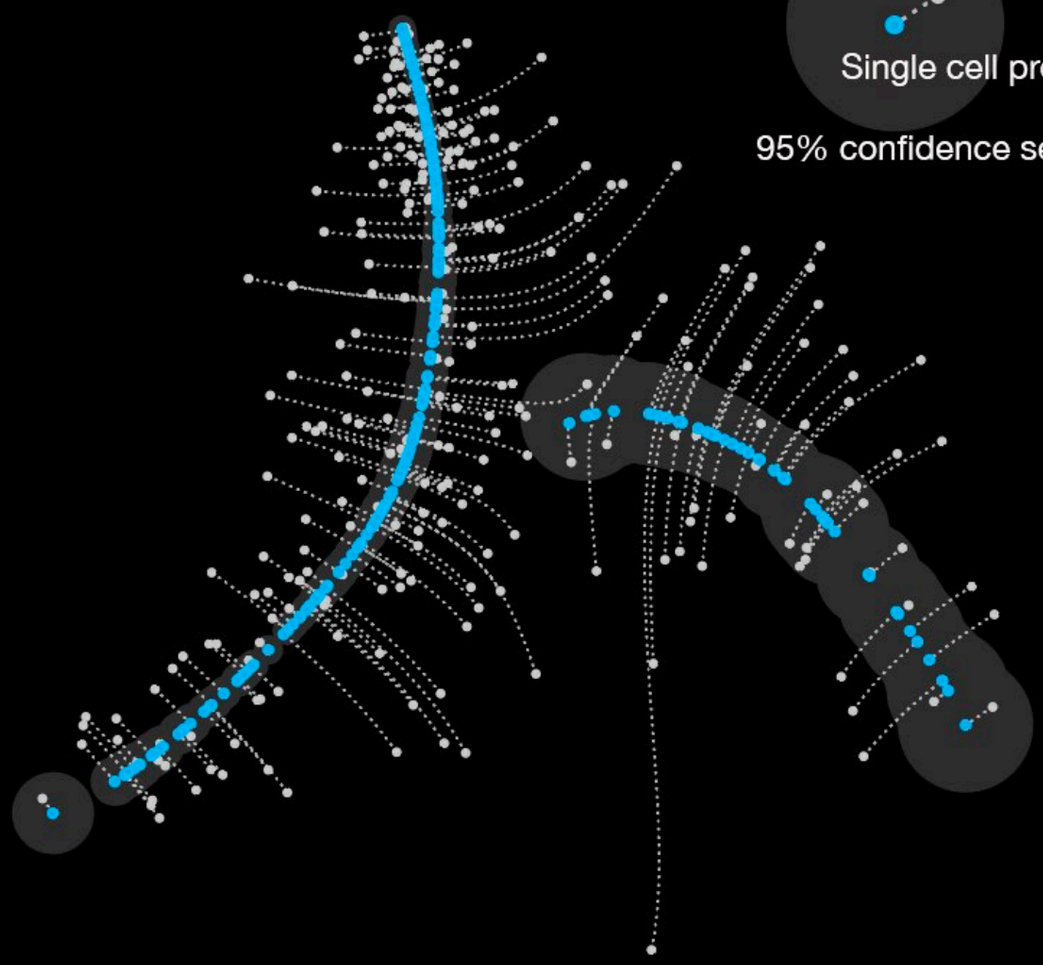
Surface

Spatial maps,
complex heterogeneous
population



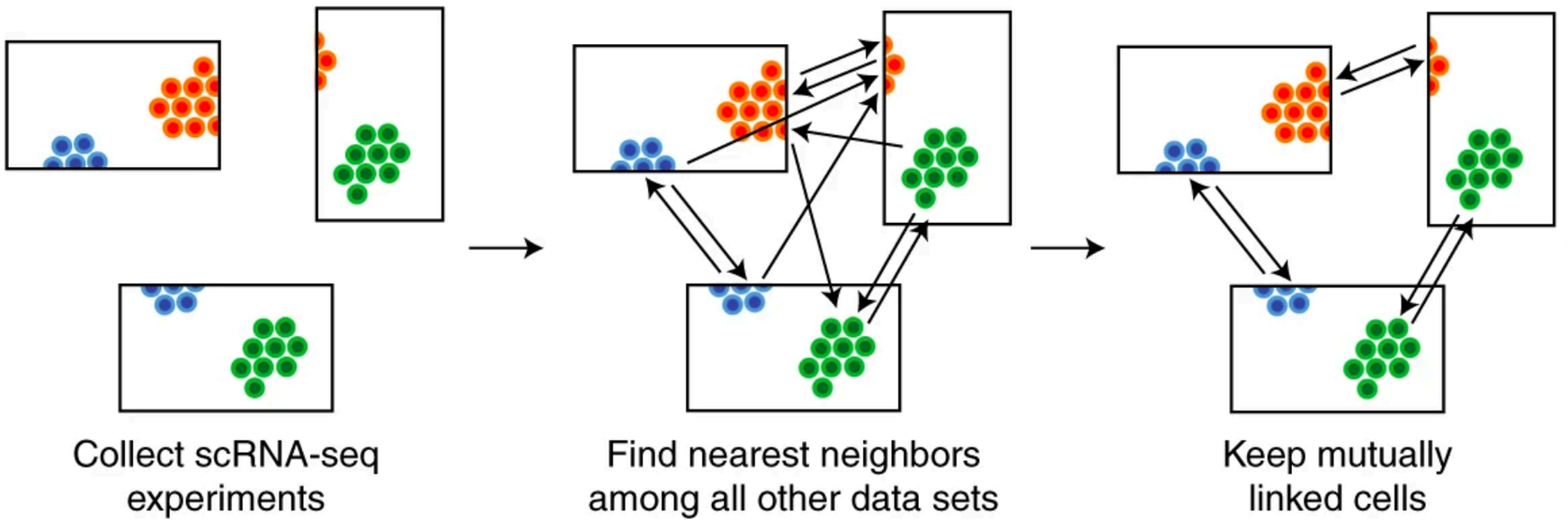
# StructDR - unified structure extraction and inference of confidence set





# Use mutual nearest neighbors to integrate datasets with batch effect

Mutual nearest neighbor




GraphDR graph construction with complex experimental design

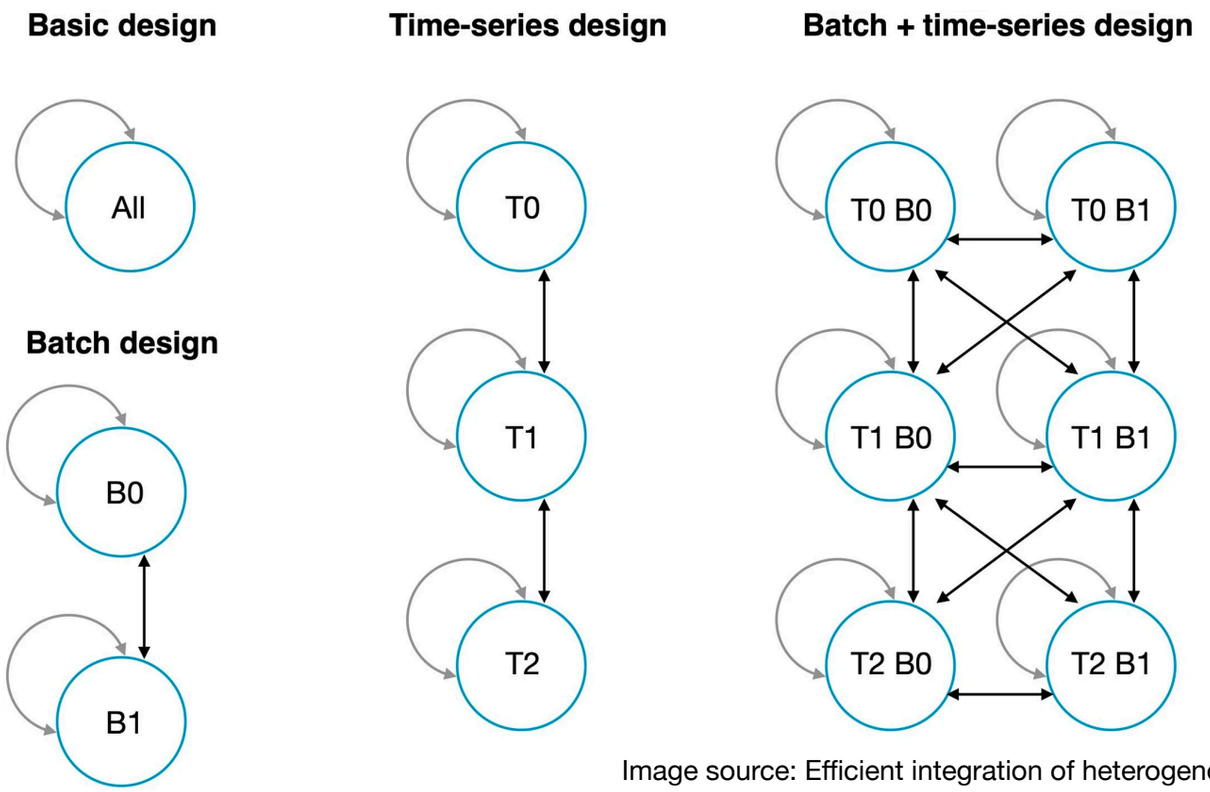
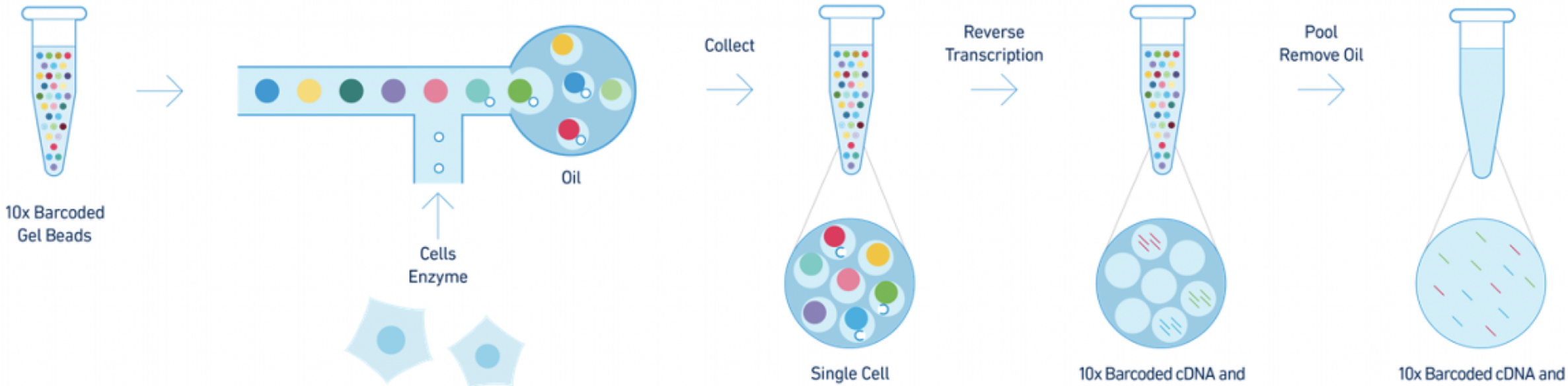
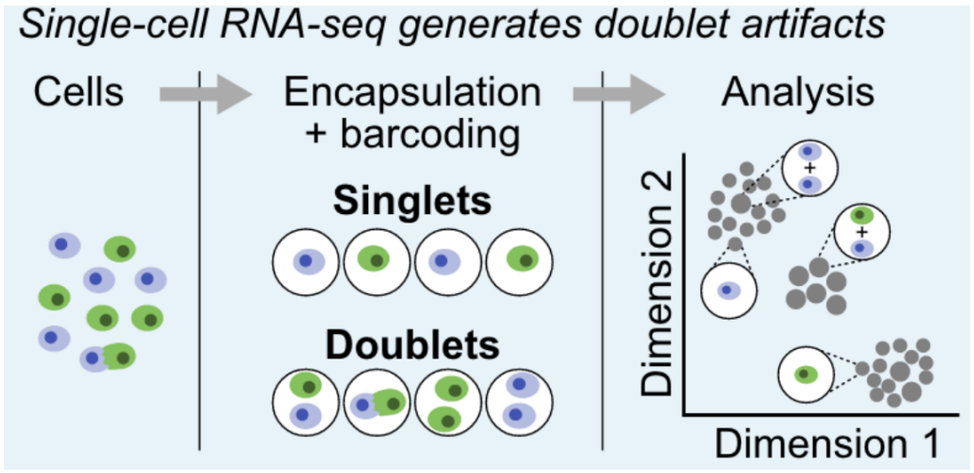


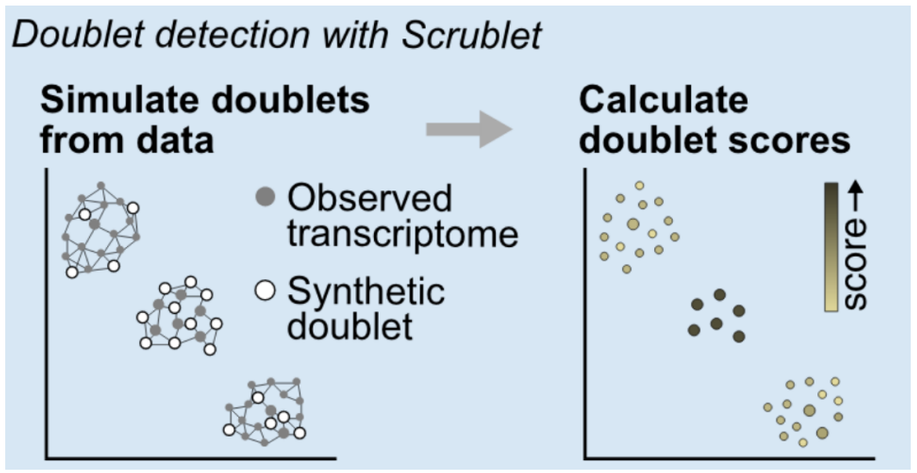

Image source: Efficient integration of heterogeneous single-cell transcriptomes using Scanorama, An analytical framework for interpretable and generalizable single-cell data analysis

# Doublet detection method with supervised learning




Most droplets contains 0 or 1 cells,
following Poisson distribution

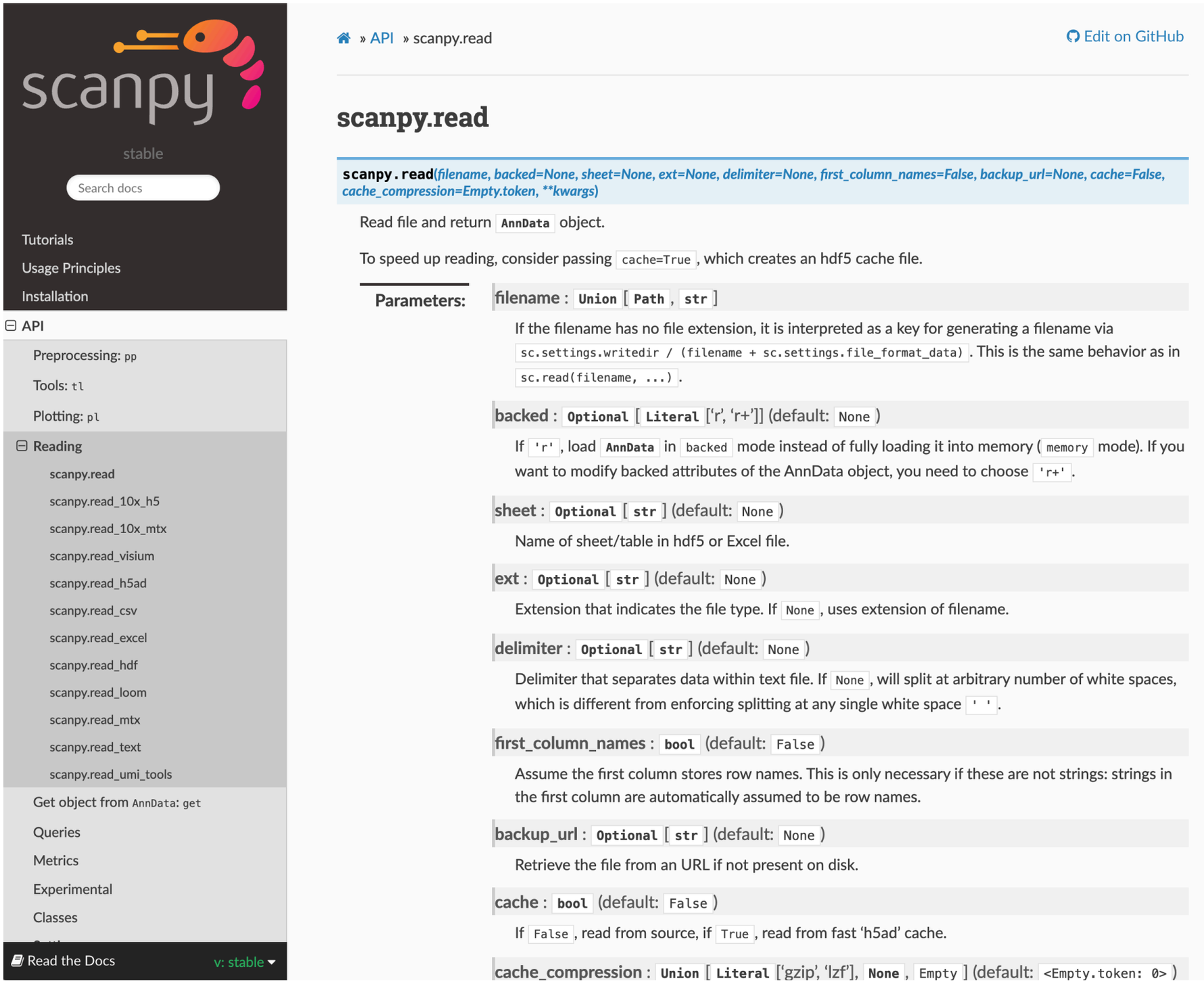



Cell Systems

**Scrublet: Computational Identification of Cell Doublets in Single-Cell Transcriptomic Data**

# Sphinx: Convert your docstring to a  documentation website



**Task for Day 4:**

1. **Generate documentation website with Sphinx**
2. **Write the README page for your repository.**
3. **Prepare for the demo**
4. **Optional: add capability of handling batch design following the provided example.**
5. **Optional: visualize a new single-cell dataset**