

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 $O(n)$

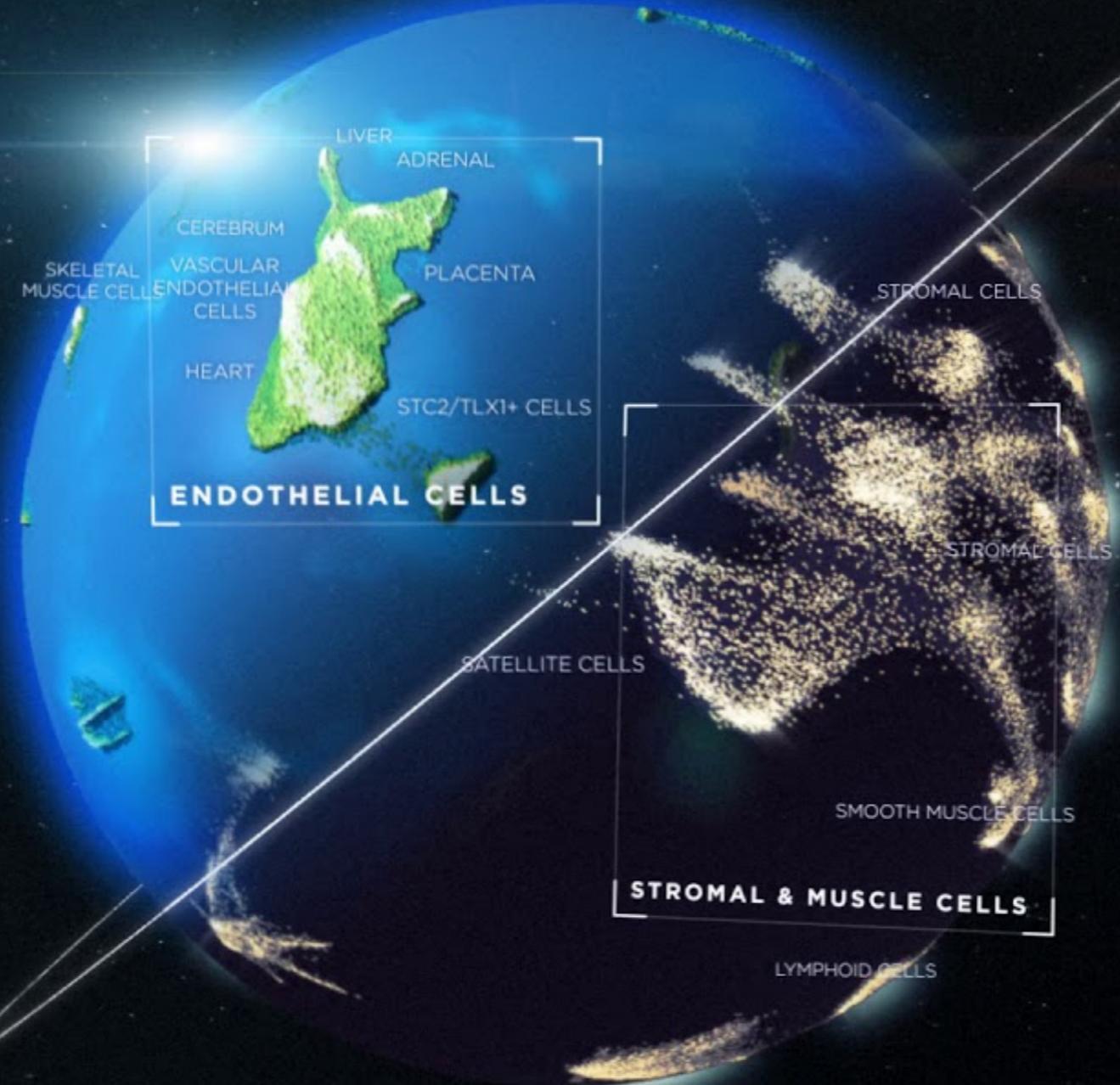
Don't do matrix inversion

$I + \lambda L$ 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

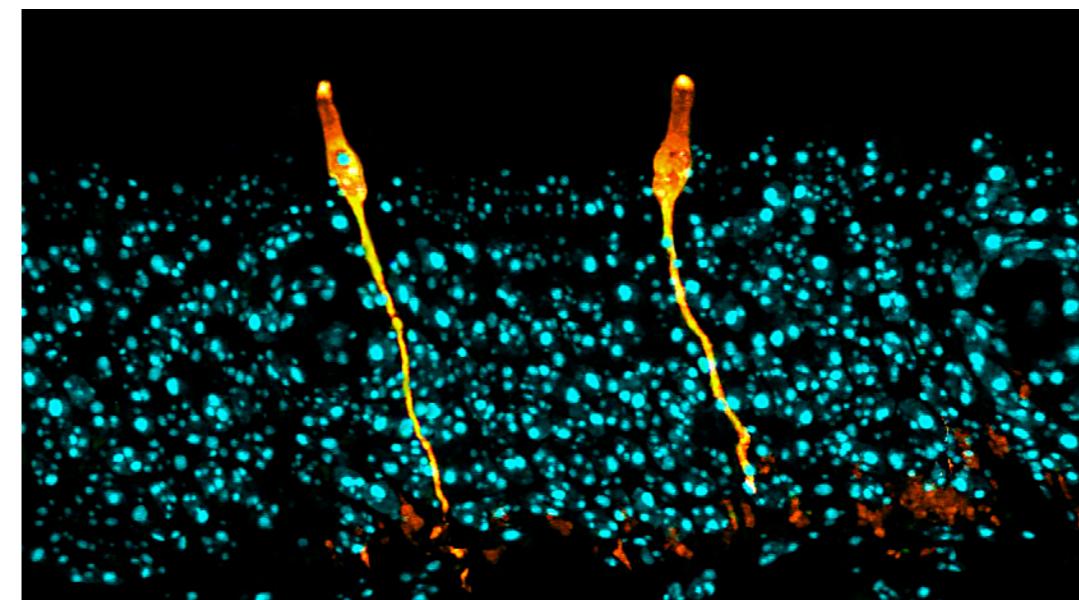
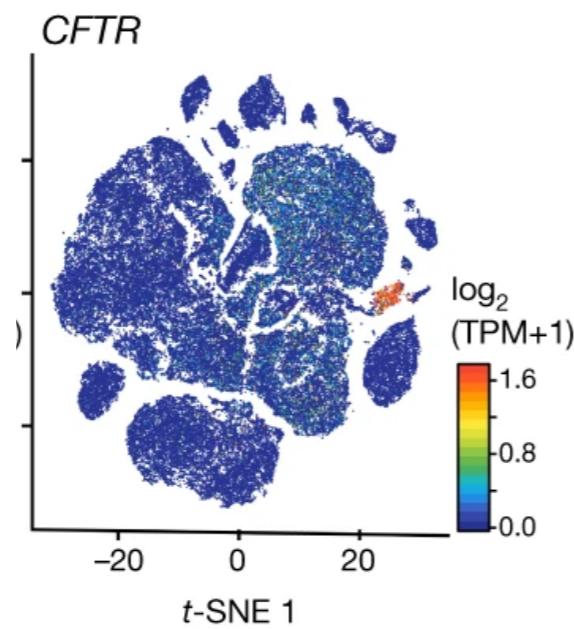
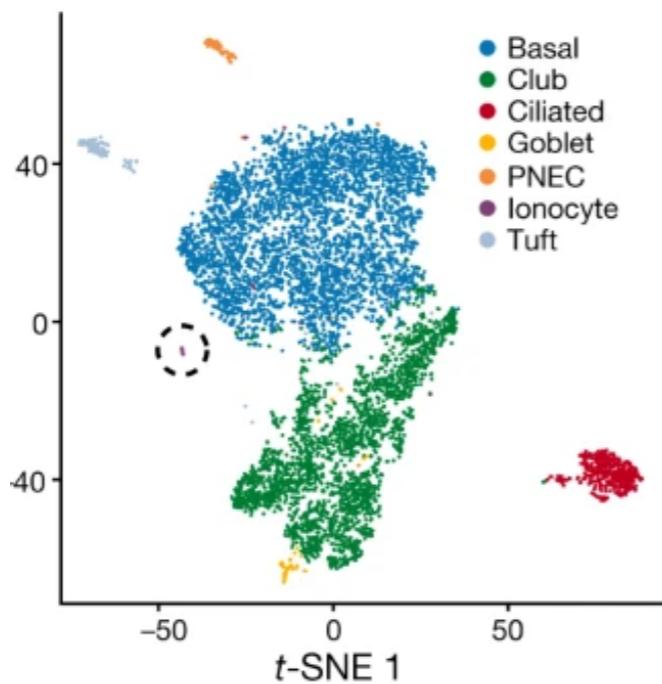


CHROMATIN ACCESSIBILITY

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

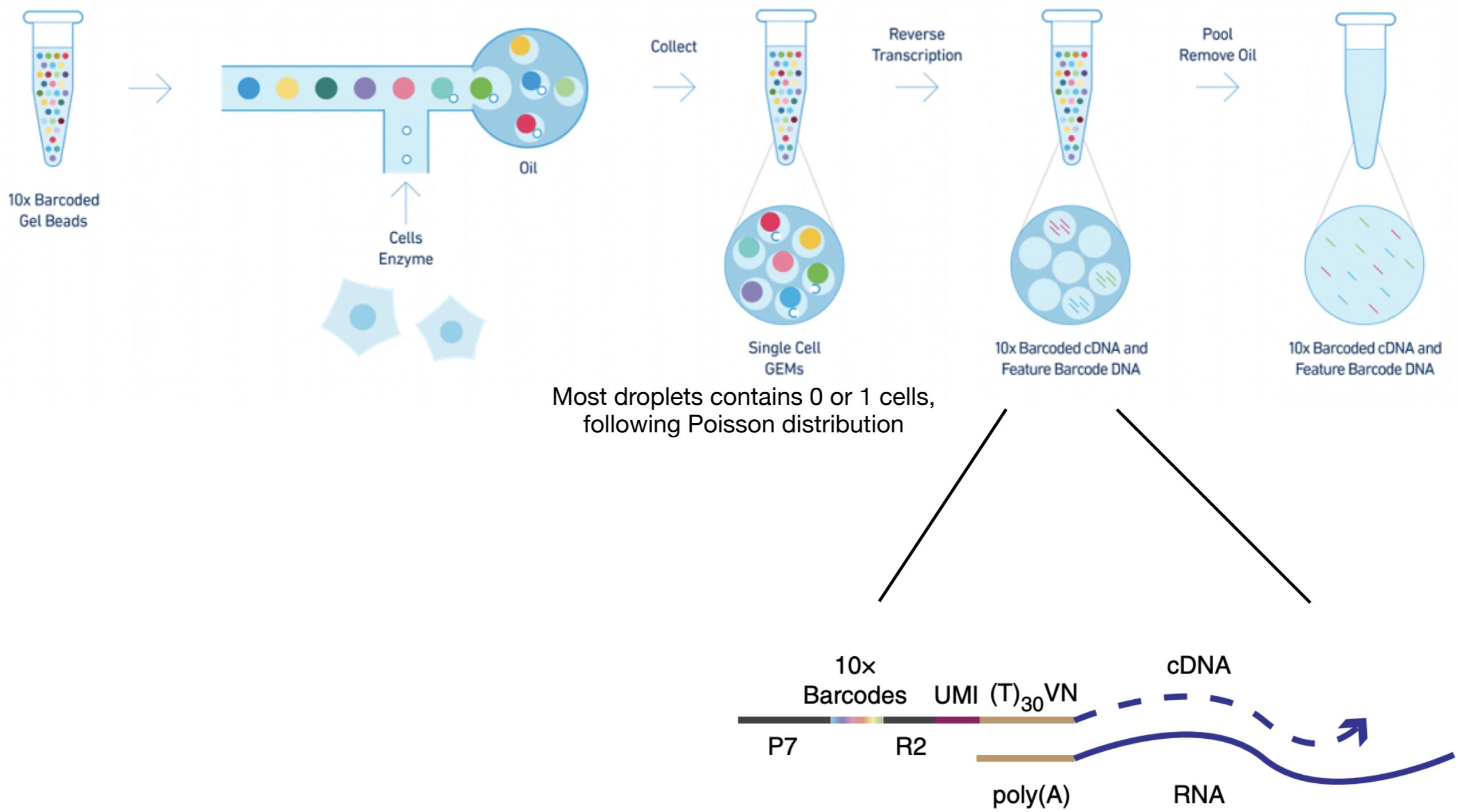
Cystic fibrosis is caused by mutations from the CFTR gene, discovered in the 1980s

What cell type does CFTR function in?

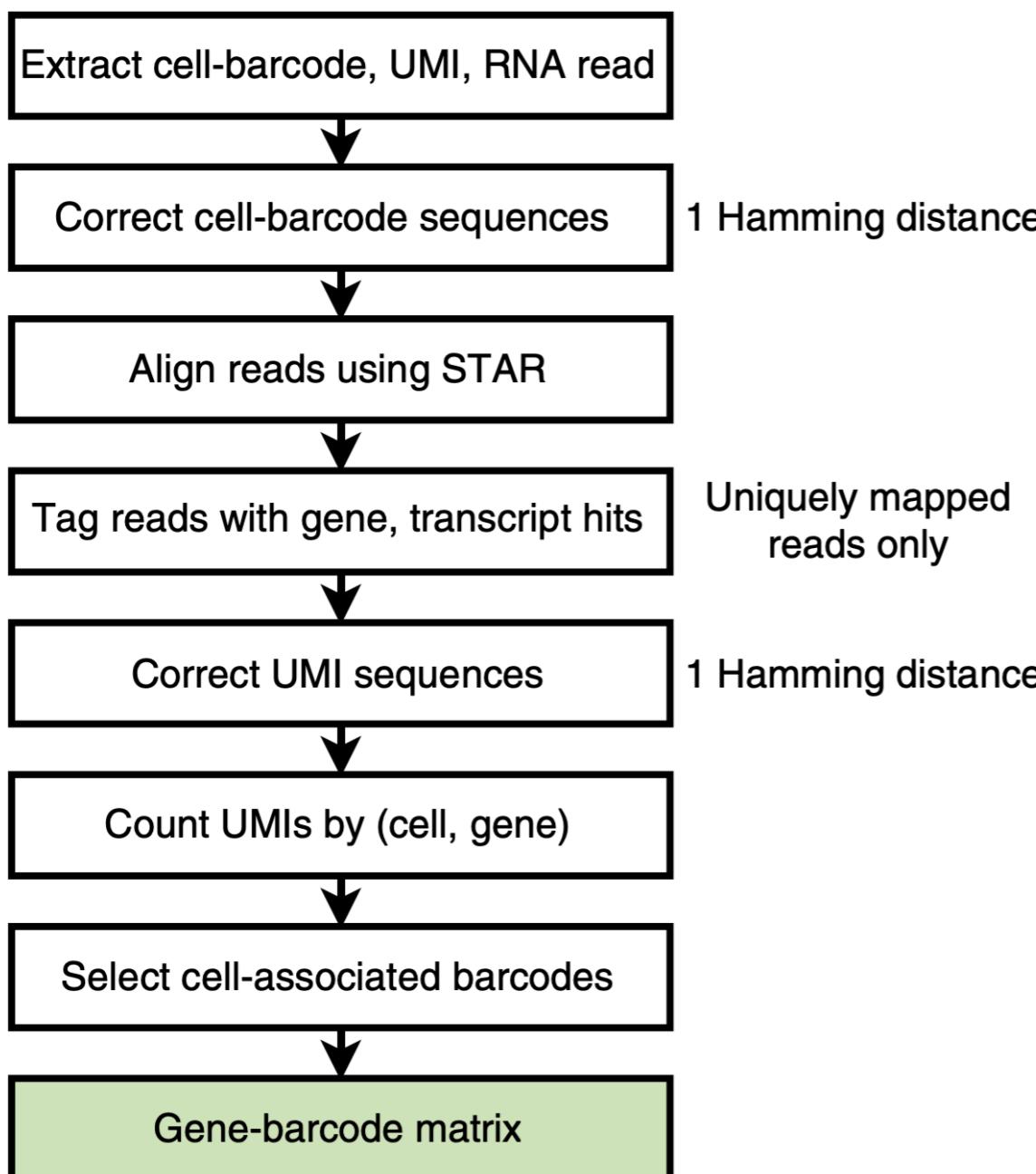
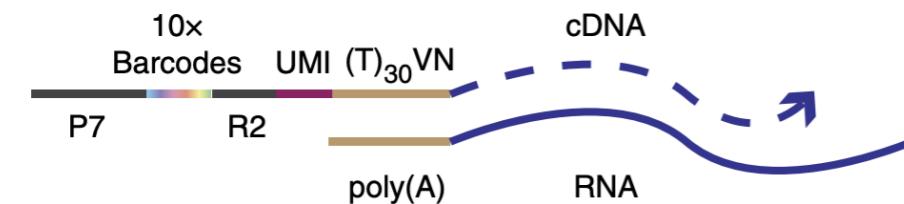


Plasschaert et al. 2018

Understand the data: single cell RNA-seq



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)

Identify cell types: Graph-based clustering

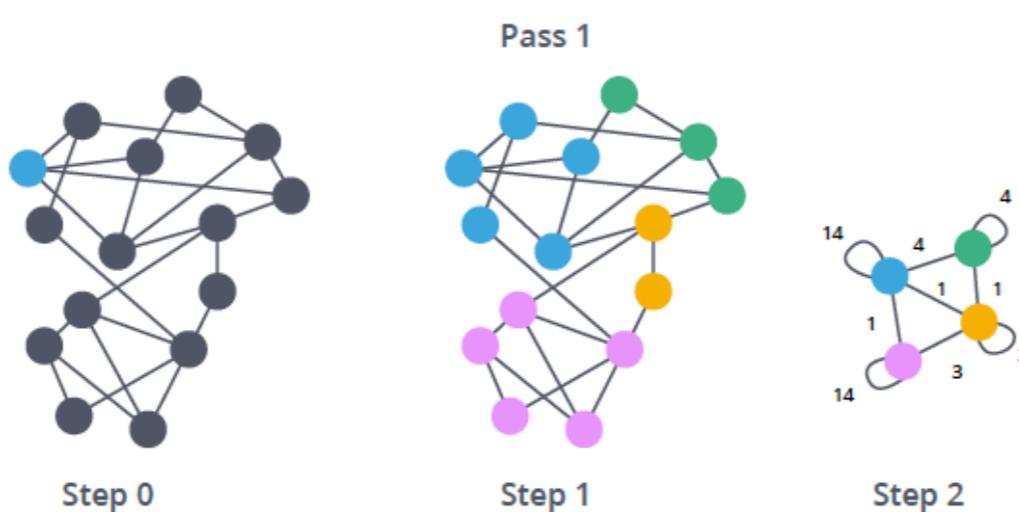
Use the topological structure of the data:

Nearest neighbor graph / shared nearest neighbor graph is often good representation for identify clusters

<https://github.com/vtraag/louvain-igraph>

Modularity:

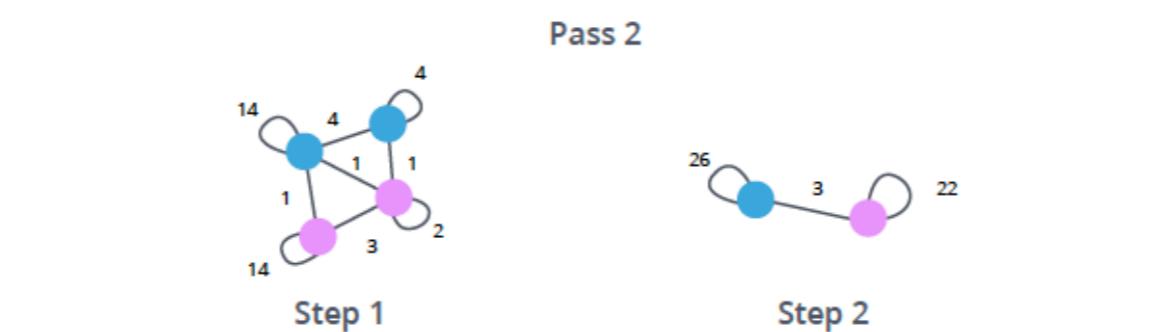
of within-cluster edges - expectation (based on product of node degrees)



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.

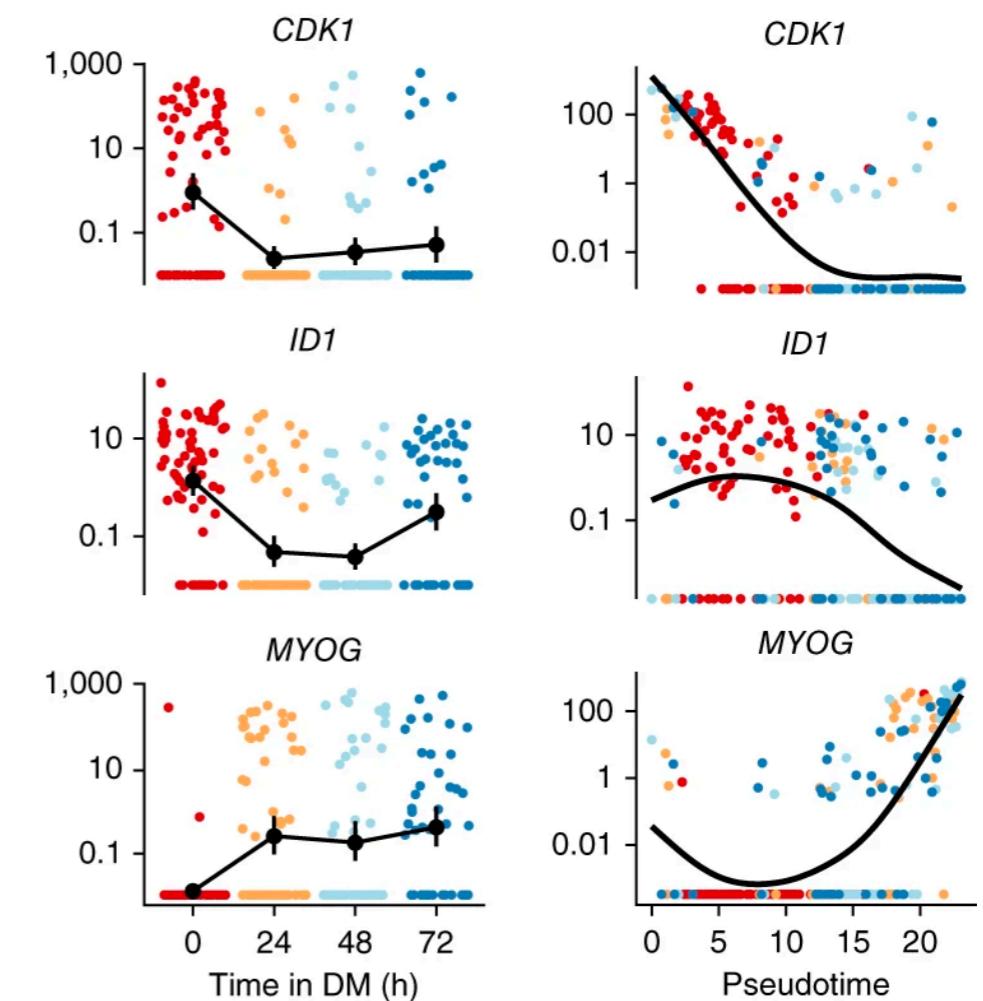
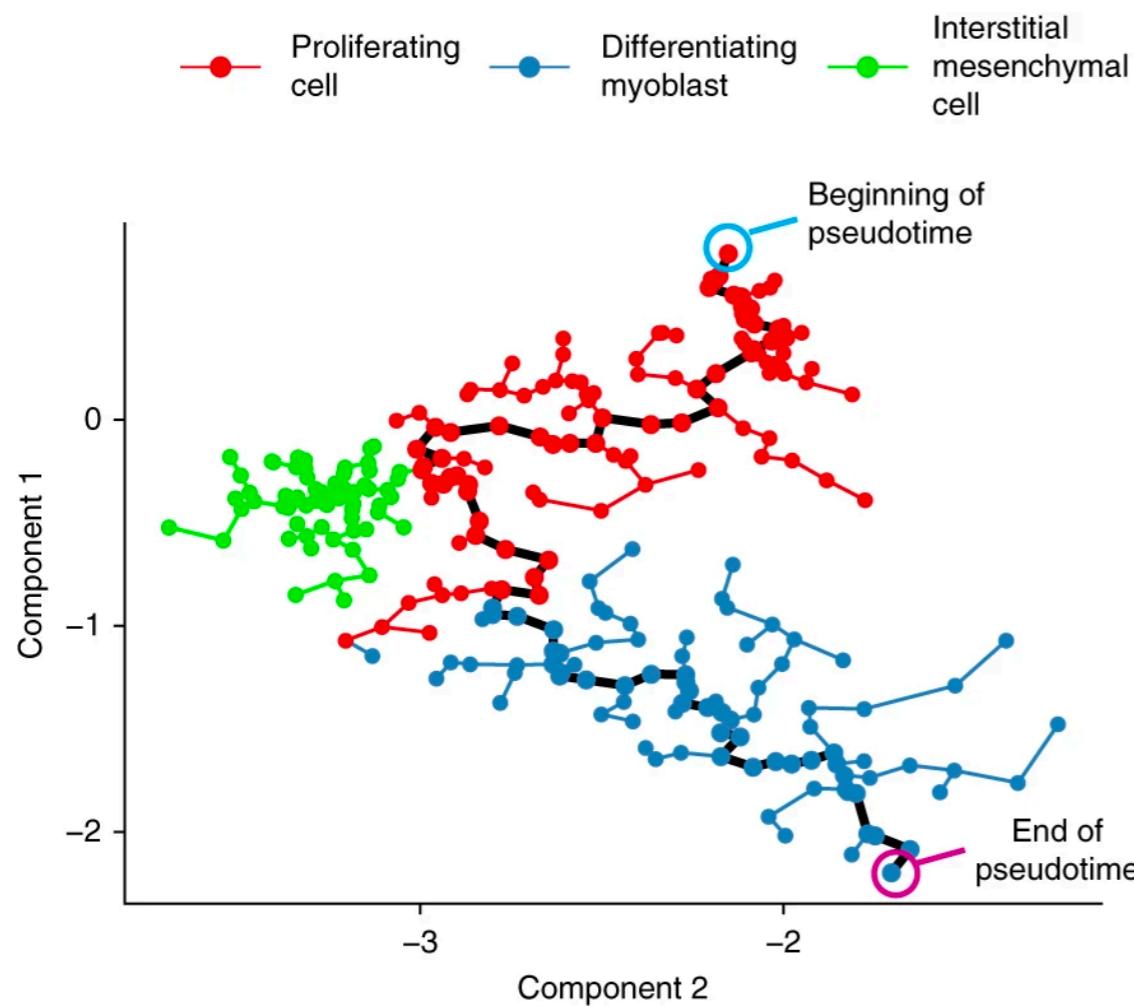
The start node joins the node with the highest modularity change. The process is repeated for each node with the above communities formed.

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.)

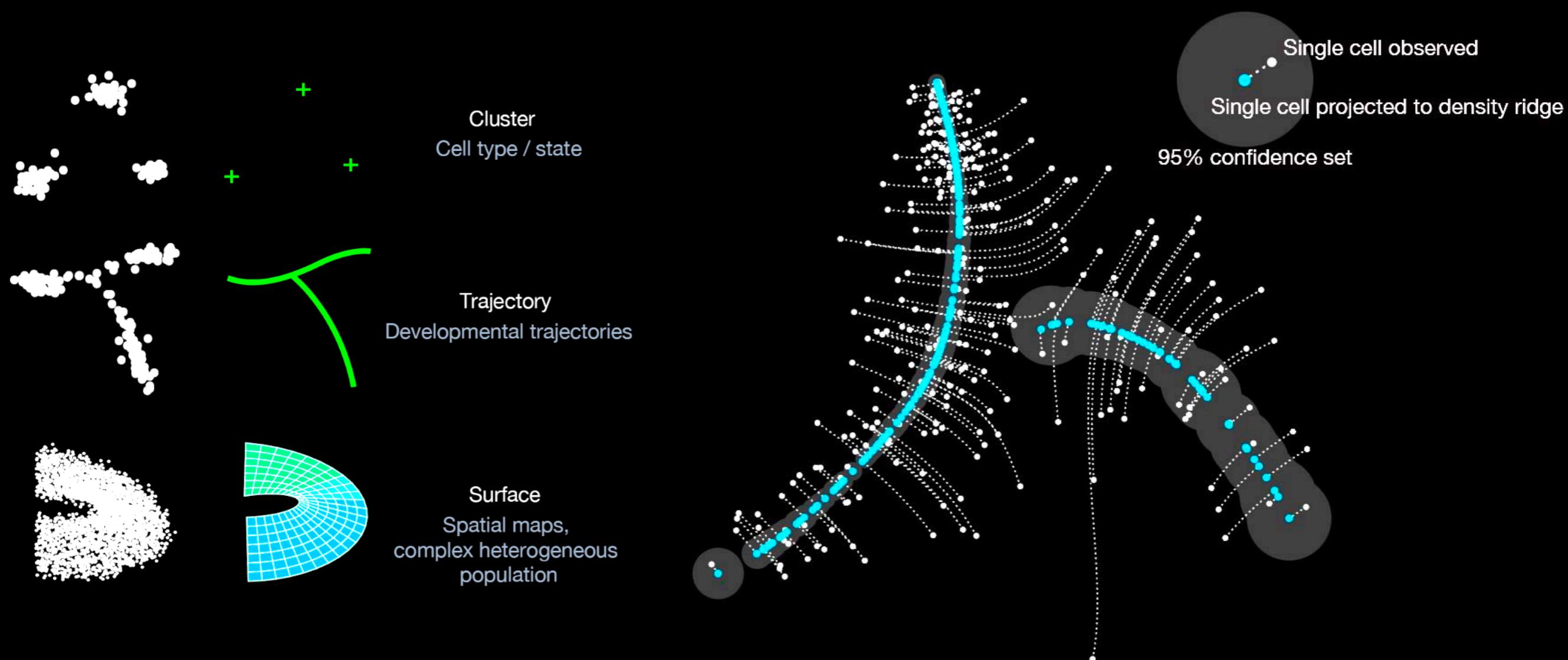


Steps 1 and 2 repeat in passes until there is no further increase in modularity or a set number of iterations have occurred.

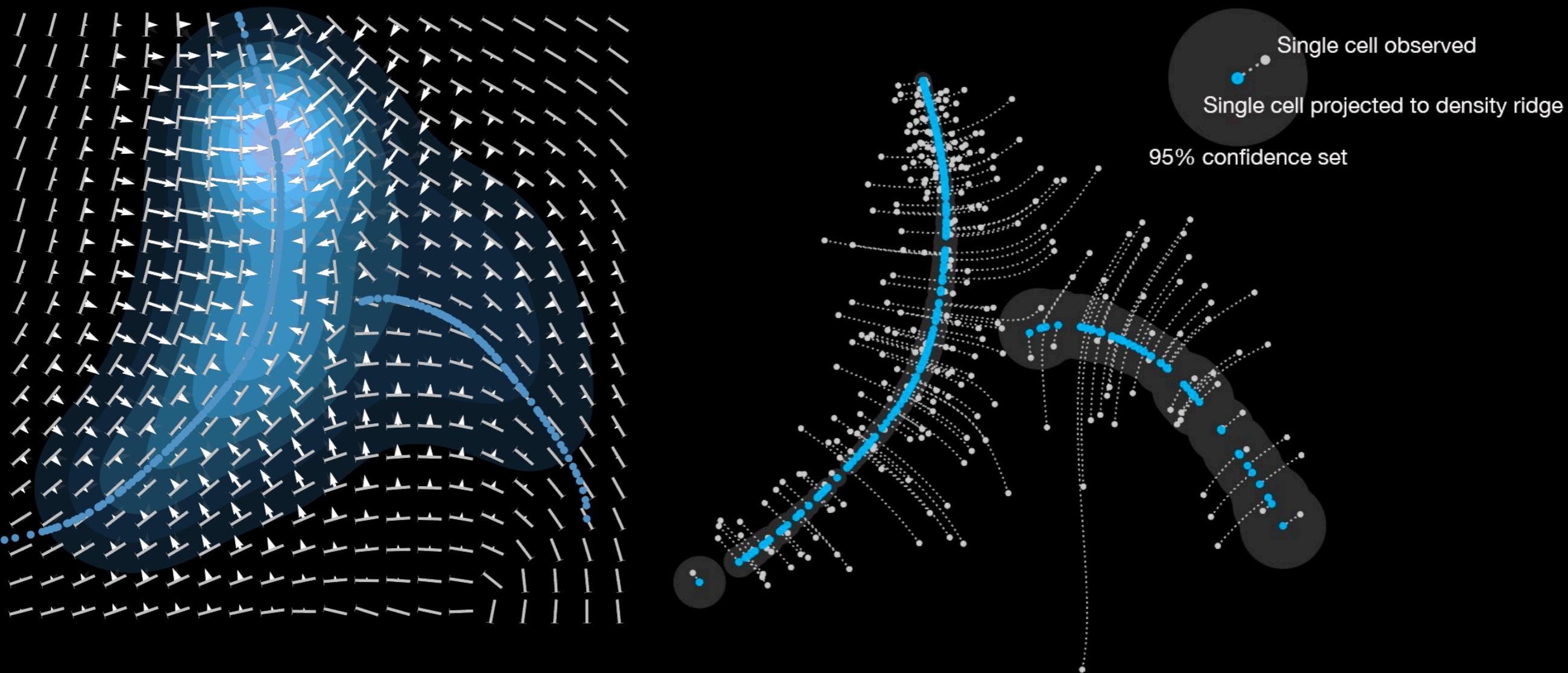
Order cells by developmental state: pseudotime / trajectory analysis



StructDR - unified structure extraction and inference of confidence set



StructDR - unified structure extraction and inference of confidence set



Use mutual nearest neighbors to integrate datasets with batch effect

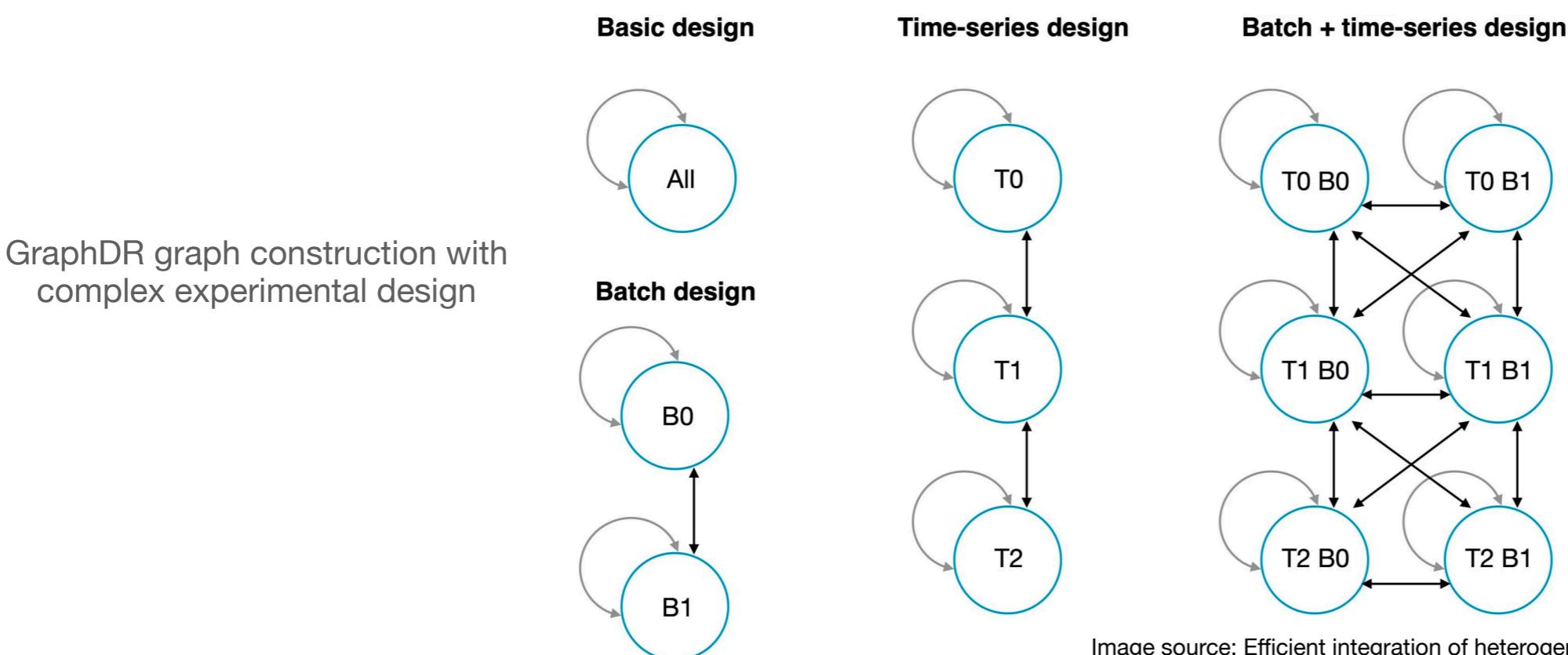
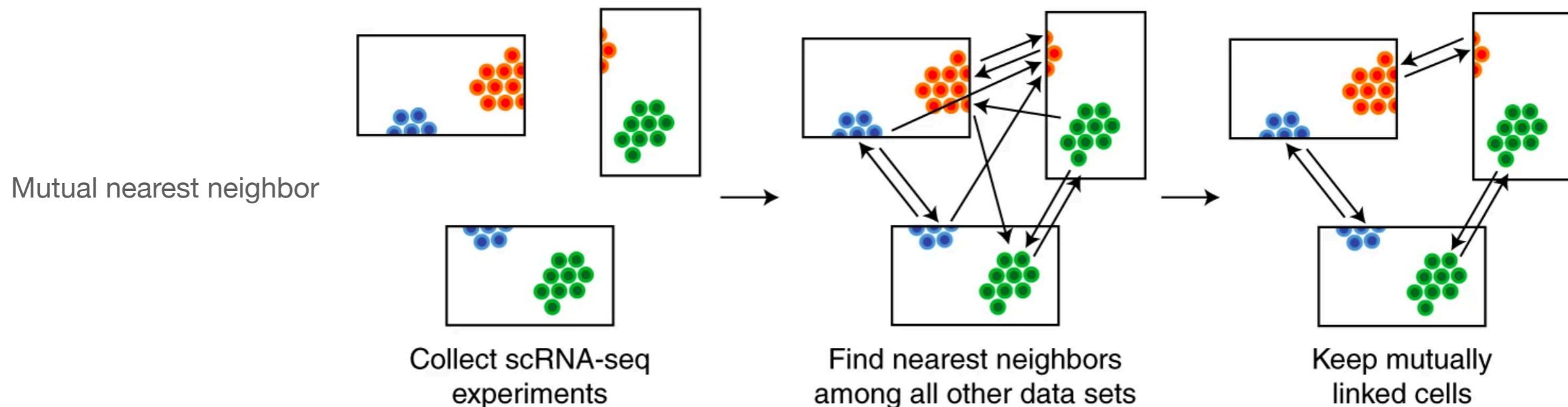
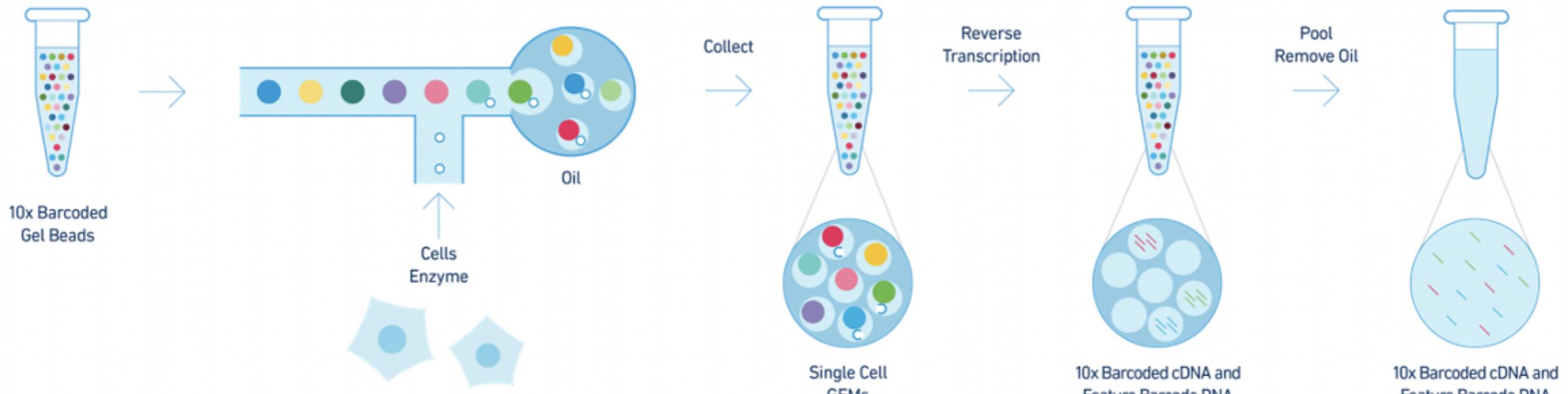
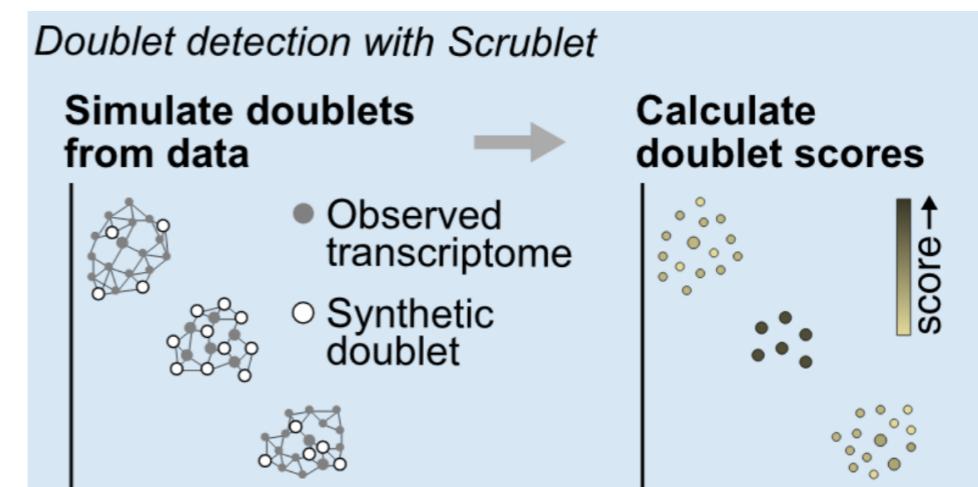
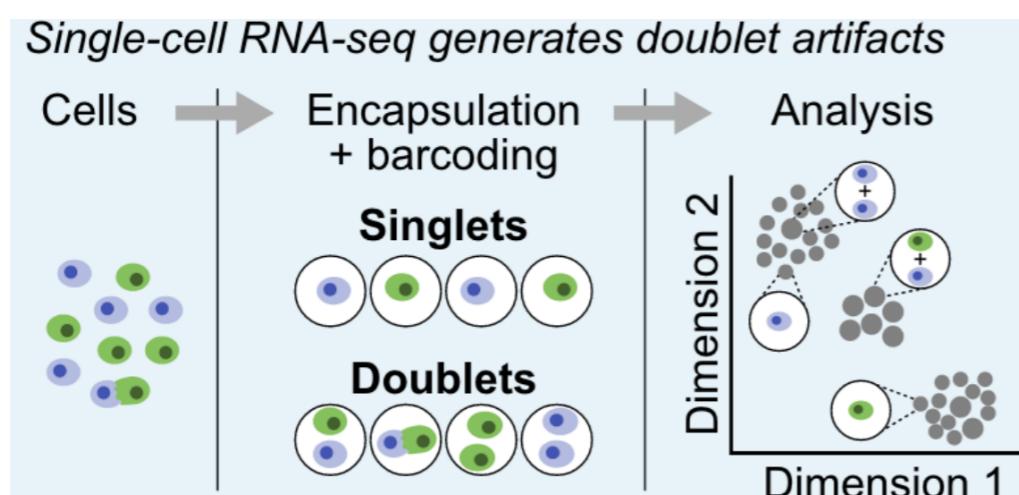


Image source: Efficient integration of heterogeneous single-cell transcriptomes using Scanorama, An analytical framework for interpretable and generalizable single-cell data analysis

Doublet detection method with supervised learning



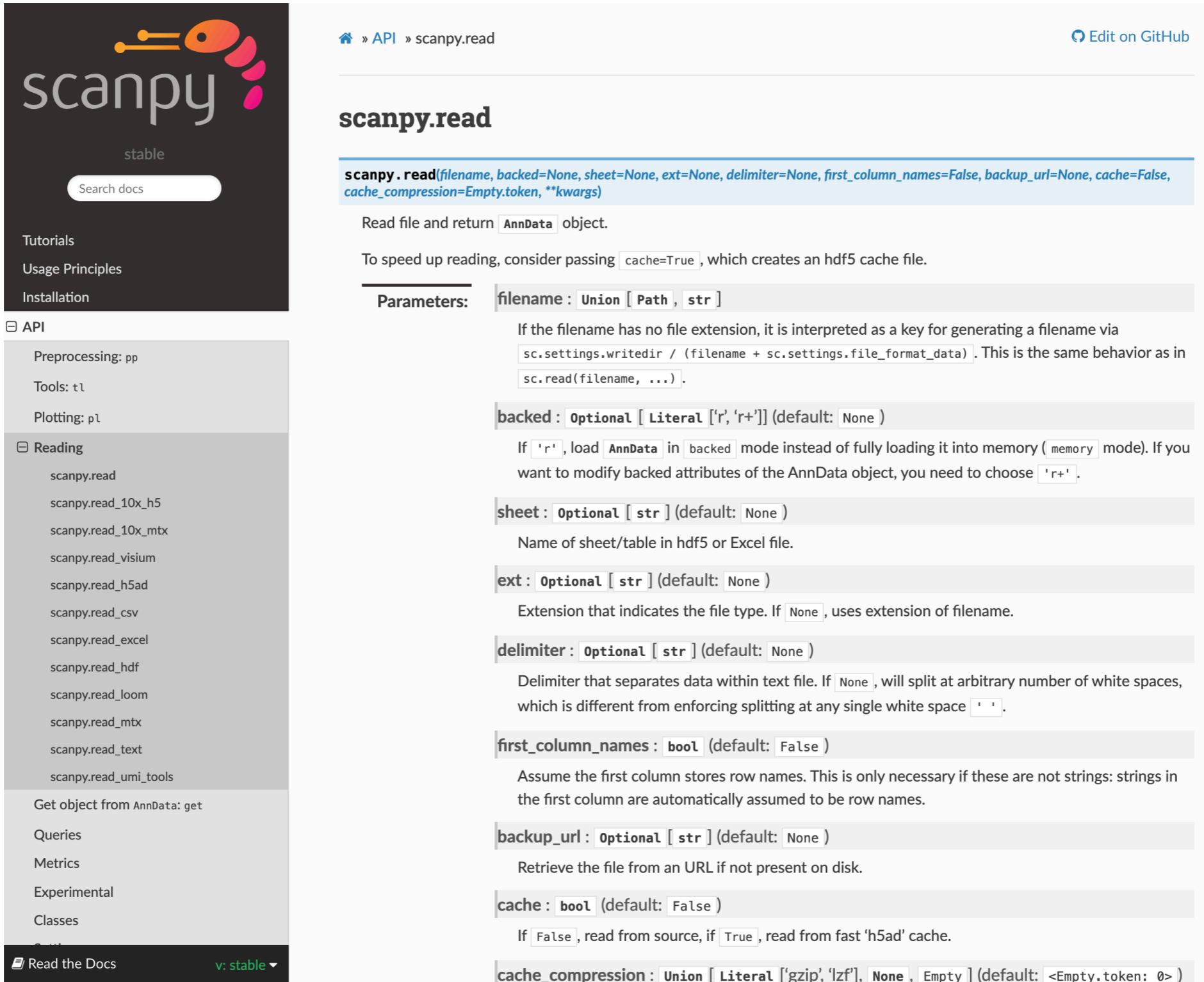
Most droplets contains 0 or 1 cells,
following Poisson distribution



Cell Systems

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

Sphinx: Convert your docstring to a documentation website



The screenshot shows the Scanpy documentation website. At the top left is the Scanpy logo with a stylized cell icon. Below it is the word "stable". A search bar says "Search docs". To its right are links for "Tutorials", "Usage Principles", and "Installation". On the far left is a sidebar with a tree view of the API documentation:

- Preprocessing: pp
- Tools: tl
- Plotting: pl
- Reading
 - scanpy.read
 - scanpy.read_10x_h5
 - scanpy.read_10x_mtx
 - scanpy.read_visium
 - scanpy.read_h5ad
 - scanpy.read_csv
 - scanpy.read_excel
 - scanpy.read_hdf
 - scanpy.read_loom
 - scanpy.read_mtx
 - scanpy.read_text
 - scanpy.read_umi_tools
- Get object from AnnData: get
- Queries
- Metrics
- Experimental
- Classes

At the bottom left are links for "Read the Docs" and "v: stable ▾". At the top right is a "Edit on GitHub" link.

The main content area shows the documentation for the `scanpy.read` function. The URL is `https://scanpy.readthedocs.io/en/stable/api/scanpy.read.html`. The function signature is:

```
scanpy.read(filename, backed=None, sheet=None, ext=None, delimiter=None, first_column_names=False, backup_url=None, cache=False, cache_compression=Empty.token, **kwargs)
```

The function reads file and returns `AnnData` object.

To speed up reading, consider passing `cache=True`, which creates an hdf5 cache file.

Parameters:

- filename : Union [Path , str]**
If the filename has no file extension, it is interpreted as a key for generating a filename via
`sc.settings.writedir / (filename + sc.settings.file_format_data)`. This is the same behavior as in
`sc.read(filename, ...)`.
- backed : Optional [Literal ['r', 'r+']] (default: None)**
If '`'r'`', load `AnnData` in `backed` mode instead of fully loading it into memory (`memory` mode). If you want to modify backed attributes of the `AnnData` object, you need to choose '`'r+'`'.
- sheet : Optional [str] (default: None)**
Name of sheet/table in hdf5 or Excel file.
- ext : Optional [str] (default: None)**
Extension that indicates the file type. If `None`, uses extension of filename.
- delimiter : Optional [str] (default: None)**
Delimiter that separates data within text file. If `None`, will split at arbitrary number of white spaces, which is different from enforcing splitting at any single white space `' '`.
- first_column_names : bool (default: False)**
Assume the first column stores row names. This is only necessary if these are not strings: strings in the first column are automatically assumed to be row names.
- backup_url : Optional [str] (default: None)**
Retrieve the file from an URL if not present on disk.
- cache : bool (default: False)**
If `False`, read from source, if `True`, read from fast 'h5ad' cache.
- cache_compression : Union [Literal ['gzip', 'lzf'], None , Empty] (default: <Empty.token: 0>)**

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**