

# scRNA-seq clustering and trajectory inference

Yumin Zheng

Dec. 1, 2022

- Clustering
  - Dimensionality Reduction
  - Clustering methods
  - Hands on
- Break
- Trajectory inference
  - Inference methods
  - Hands on

- Clustering
  - Introduction to clustering
    - What is clustering
    - Why we need clustering
  - Dimensionality Reduction
    - Principal Components Analysis
  - Clustering methods
    - K-means based methods
    - Hierarchical clustering methods
    - Shared-nearest-neighbors methods
  - Hands on
    - Scanpy for single-cell clustering analysis

# Introduction

- Clustering
  - Group a set of objects that objects in the same group are more similar to each other than to those in other groups.

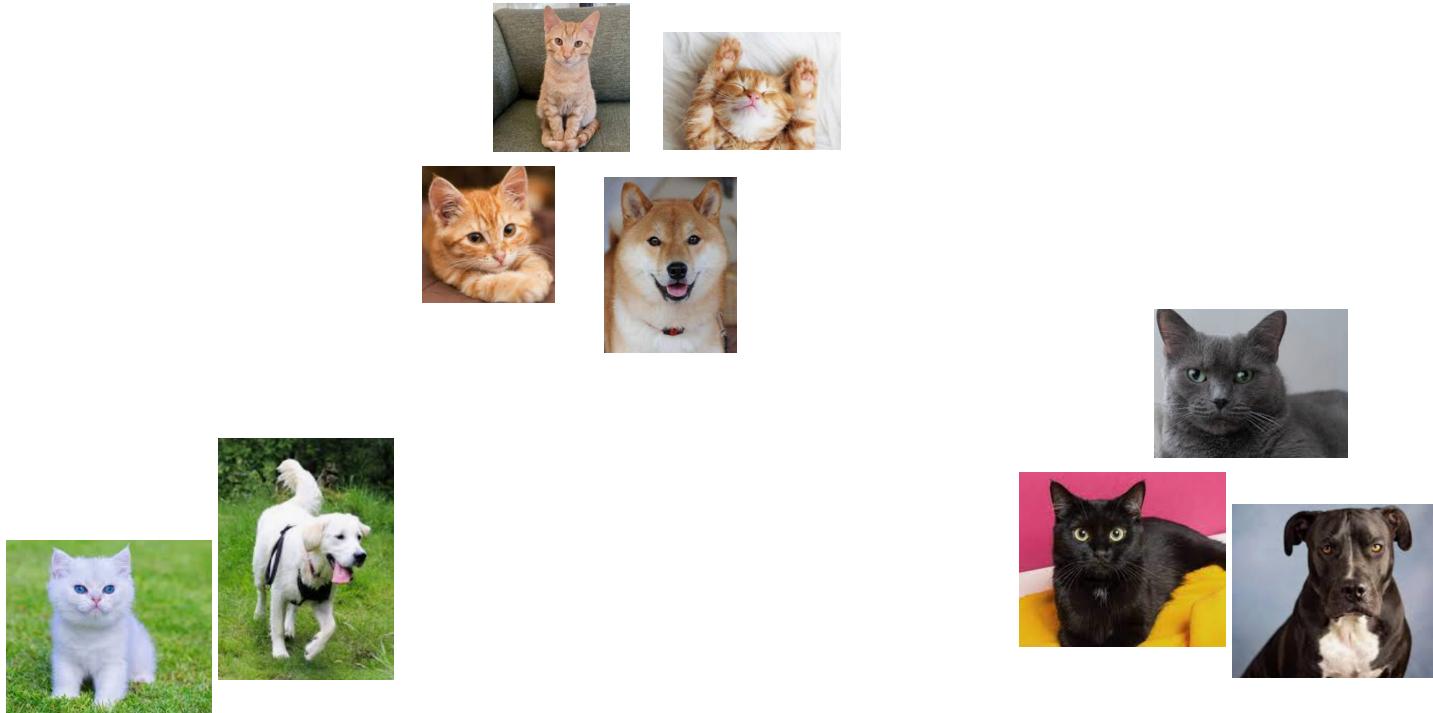


# Introduction

- Clustering vs Classification
- Clustering:
  - unlabeled dataset
- Classification:
  - labeled dataset

# Introduction

- Clustering based on visual similarity



## Introduction

- single-cell RNA-seq clustering
  - identify cell populations (especially, rare cell populations)
  - based on the similarities of transcriptomes
  - No cell type label

# Introduction

- single-cell RNA-seq data

- Raw sequences

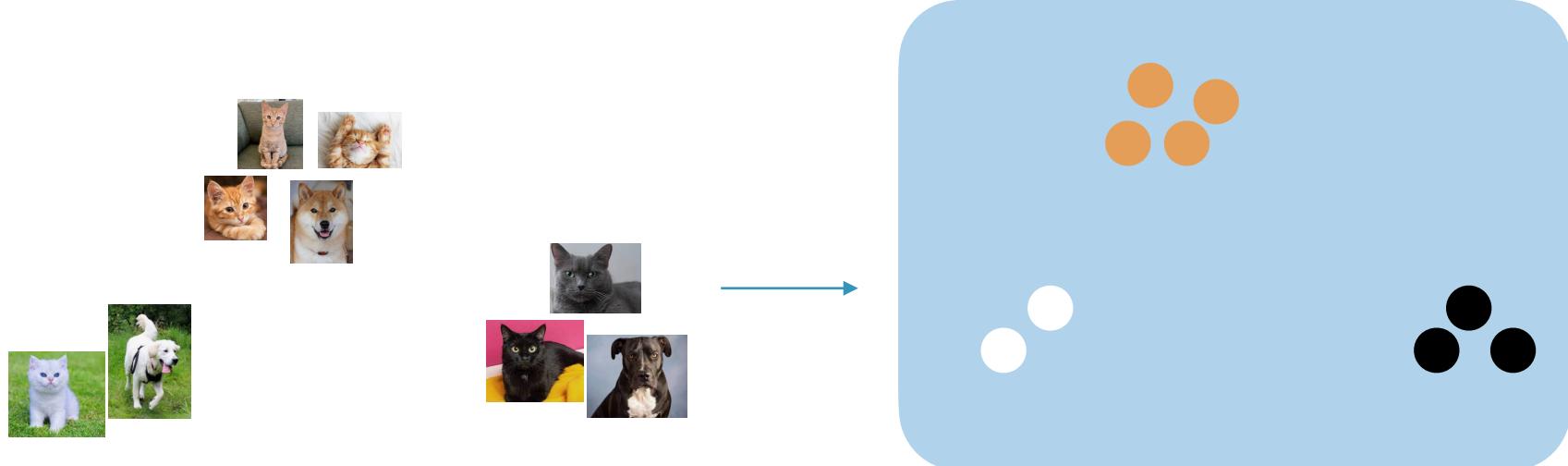
```
>NS500724:80:HWFYWBGX:1:11101:10427:1039 1:N:0:143
GAACANACAGGCTGGCGTTGTGGAAGGGCTGGCGAAGTCGCCGACATTCAAGCGCGCGACGTTCTAGATTGCAGATTGCCCTTGCCGTGAGTTGCTACAGACGGCGAATGTTGGGTGCTAAACCCGTCTTATAC
>NS500724:80:HWFYWBGX:1:11101:26792:1051 1:N:0:143
GGCTTNTACCCAACACTCAATATCTCAGATGAGTTCTAGCAATGTTGCAAATTAACAAAGGTTGGTATGCAAAGTATTCTACCTGTCTTATACACATCTCGAGGCCACGAGACGTTGGATGAAATATCGTATGCCGTCTTAG
>NS500724:80:HWFYWBGX:1:11101:17792:1055 1:N:0:143
CCTTANAGCTTTACAATAAGAGGCCATGCTAAATTAGGTGAATTGTCCATACTAAATTCAACTAAGTTGAAACAATTCTATCTGCATCTACAACCTGTGGATTCCCACAATGCTGATGCATAAGTAAATGTTGTAACCATCACACG
>NS500724:80:HWFYWBGX:1:11101:12348:1058 1:N:0:143
CACTTNCCCTGTGAAGGTGCTTTCTCAATGGCACACACTGGTTGTAACACAAAGGAATTTTATGAACCACAAATCATTACTACAGACAAACACATTGTGCTGTCTTATACACATCTCGAGGCCACGAGACGTTGGATGA
>NS500724:80:HWFYWBGX:1:11101:19632:1072 1:N:0:143
GGTTAAGTGTGCTCTGCAACCGAGCCGATGTTCAAGTGCCTTACAGTGCCCCAGAGAGACTGGAAAGTGGTACCCGACCGAGCCGCTGACAGGATCGCCGAGCTGCGACCGAATCCACACATCATCGATTGCGTCTCGATTGTGGCGCGAGCC
```

- Alignment and generate count matrix

$$\begin{bmatrix} 0 & 0 & 10 & 2 & \dots & 0 & 3 \\ 1 & 3 & 9 & 3 & \dots & 8 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 5 & 9 & 3 & 1 & \dots & 5 & 5 \end{bmatrix}$$

# Introduction

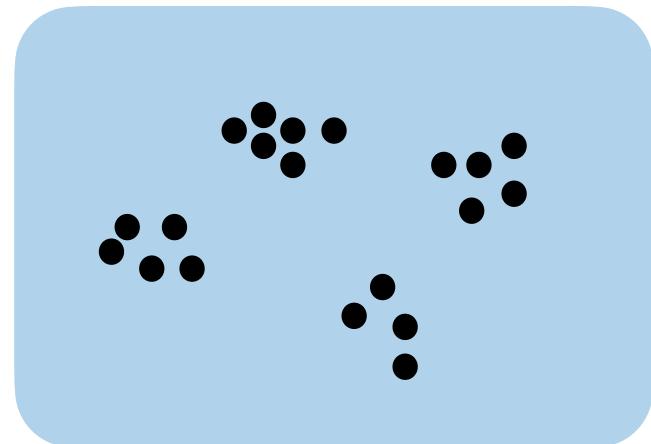
- Features



## Dimensionality reduction

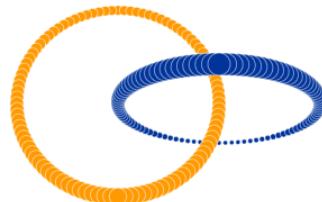
- Single-cell data
  - High dimension (various genes in each cell)
  - $[n, m]$  n cells, m genes

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 2 & \dots & 0 & 3 \\ 1 & 3 & & 9 & 3 & \dots & 8 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 5 & 9 & & 3 & 1 & \dots & 5 & 5 \end{bmatrix}$$



## Dimensionality reduction

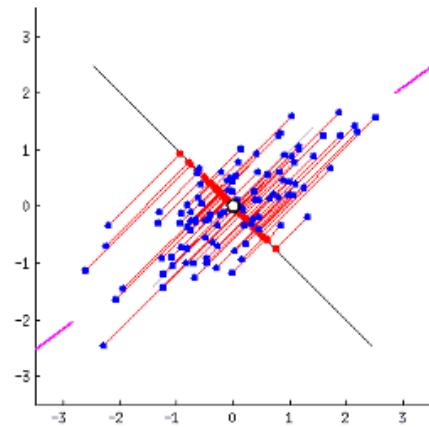
- Linear
  - Principal Components Analysis (PCA)
- Non-linear
  - t-SNE/UMAP
  - Auto-encoder



Non-linearly separable data

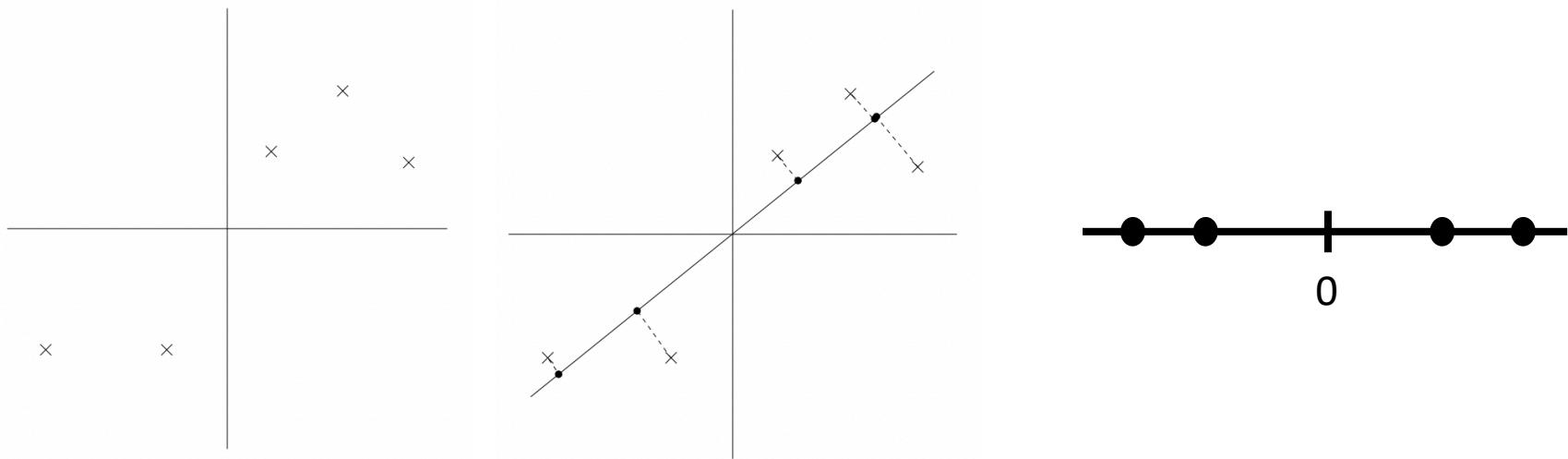
## Dimensionality reduction

- Principal components analysis (PCA)
  - Find a linear transformation to project data from high-dimension to low-dimension space that maximize the variance between data points.



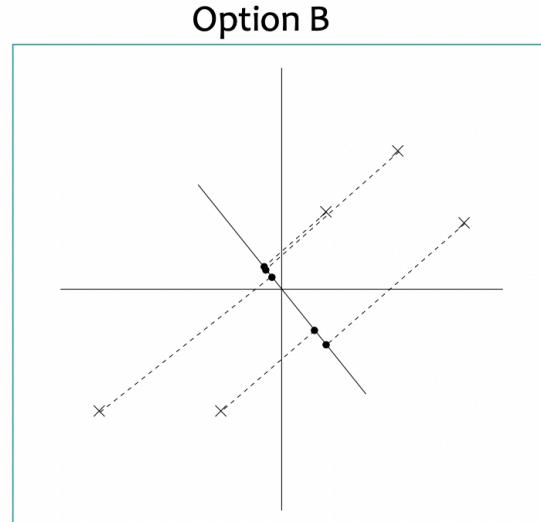
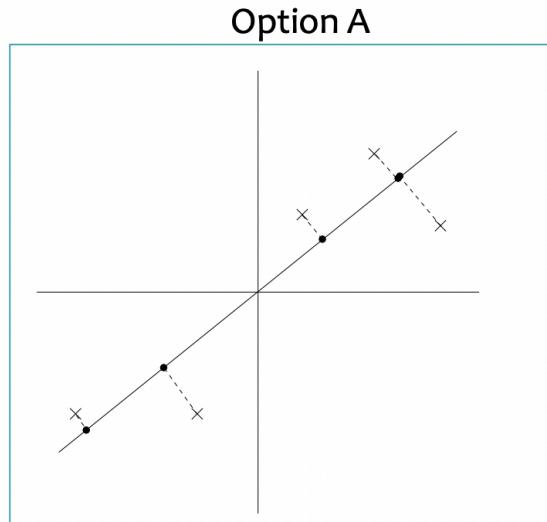
## Dimensionality reduction

- Principal components analysis (PCA)
  - projection



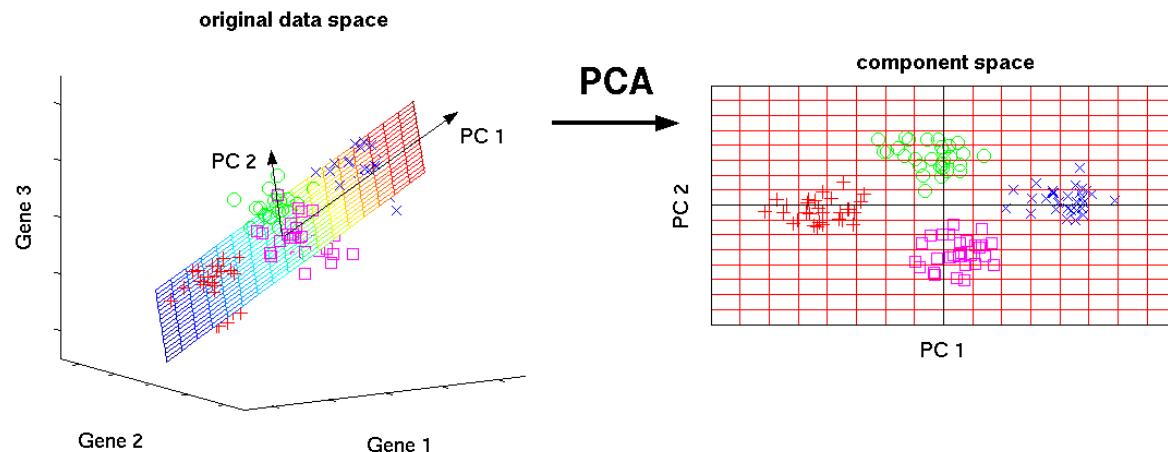
## Dimensionality reduction

- Principal components analysis (PCA)
  - maximize the variance of the projection in the residual subspace



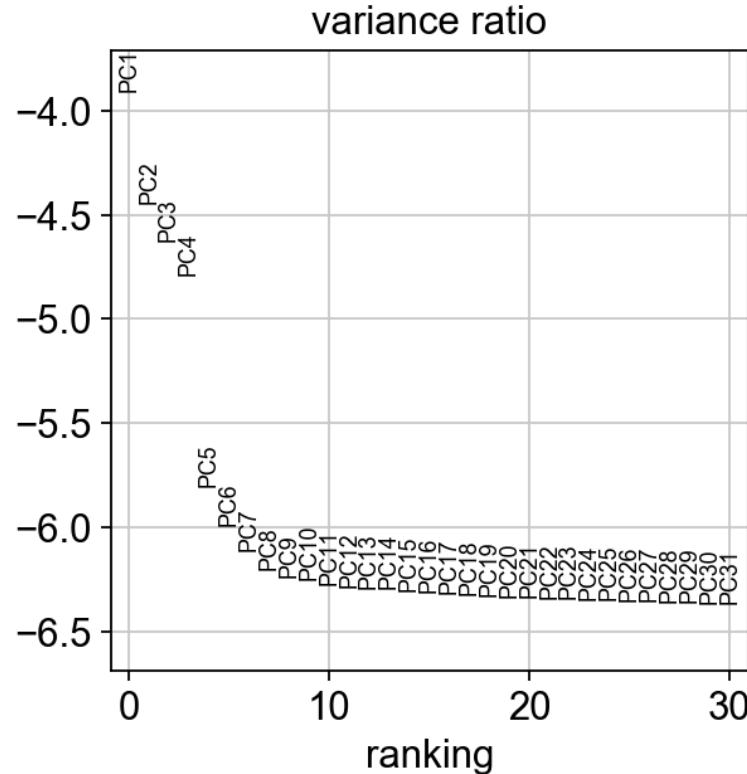
## Dimensionality reduction

- Principal components analysis (PCA)
  - First 2 dimensions to visualize



## Dimensionality reduction

- Principal components analysis (PCA)
  - Variance



## Clustering methods

- K-means based methods
- Hierarchical clustering methods
- Graph-based clustering

## K-means clustering methods

- Objective:
  - Partition cells in to K different clusters
- Aims:
  - Minimize within-cluster variation
  - Maximize between-clusters variation
- Process:
  - Find cluster centroids
  - Assign cells to its nearest cluster
  - Iterative refines clustering results

## K-means clustering methods

- Pros:
  - fast
- Cons:
  - assumes a pre-determined number of clusters
  - sensitive to outliers
  - tends to define equally-sized clusters

## K-means clustering methods

- Objective
  - partition n cells into S groups

$$\arg \min_{\mathbf{S}} \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg \min_{\mathbf{S}} \sum_{i=1}^k |S_i| \operatorname{Var} S_i$$

$(x_1, x_2, \dots, x_n)$ : observations (cells)

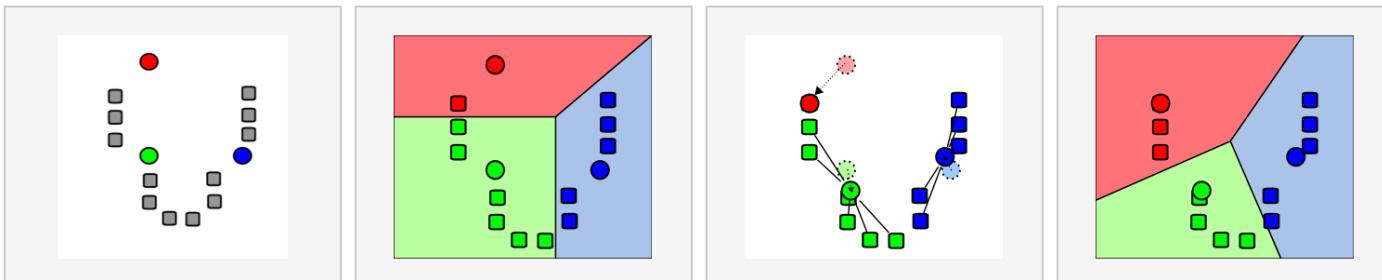
$x_n = [g_1, g_2, \dots, g_d]$ : genes

$\boldsymbol{\mu}_i$ : the mean of points in  $S_i$

## K-means clustering methods

- steps

- randomly select  $k$  data points to serve as initial cluster centers,
- for each point, 1) compute to centroids, 2) assign to closest cluster,
- calculate the mean of each cluster (the ‘mean’ in ‘k-mean’) to define its centroid,
- for each point compute the distance to these means to choose the closest,
- repeat until the distance between centroids and data points is minimal (ie clusters do not change) or the maximum number of iterations is reached,
- compute the total variation within clusters



## Hierarchical clustering methods

- Objective:
  - Build a hierarchy of clusters
  - Yielding a dendrogram (i.e. tree)
    - Groups together cells with similar expression patterns
- Types of hierarchical clustering
  - Agglomerative (bottom-up)
  - Divisive (top-down)

## Hierarchical clustering methods

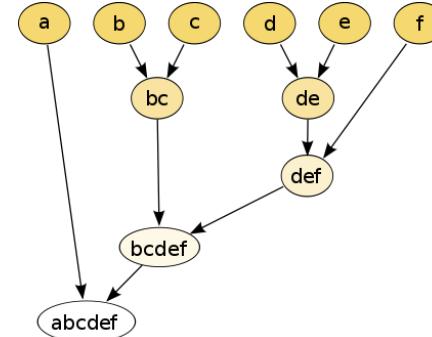
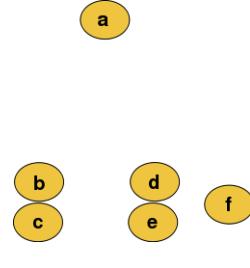
- Agglomerative (bottom-up):
  - each observation (cell) starts in its own cluster,
  - pairs of clusters are merged as one moves up the hierarchy.
- Divisive (top-down):
  - all observations (cells) start in one cluster,
  - splits are performed recursively as one moves down the hierarchy.

## Hierarchical clustering methods

- Pros:
  - deterministic method
  - returns partitions at all levels along the dendrogram
- Cons:
  - computationally expensive in time and memory
  - that increase proportionally
  - to the square of the number of data points

# Hierarchical clustering methods

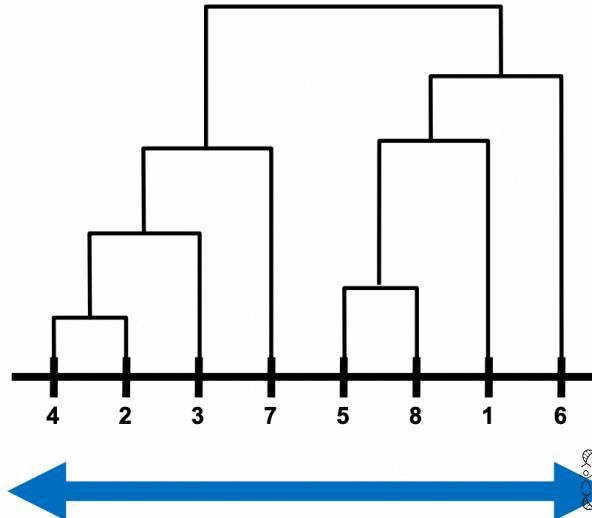
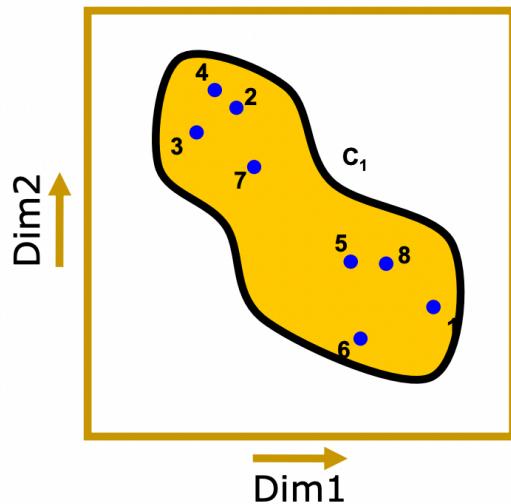
- bottom-up hierarchical clustering example



hierarchical clustering results

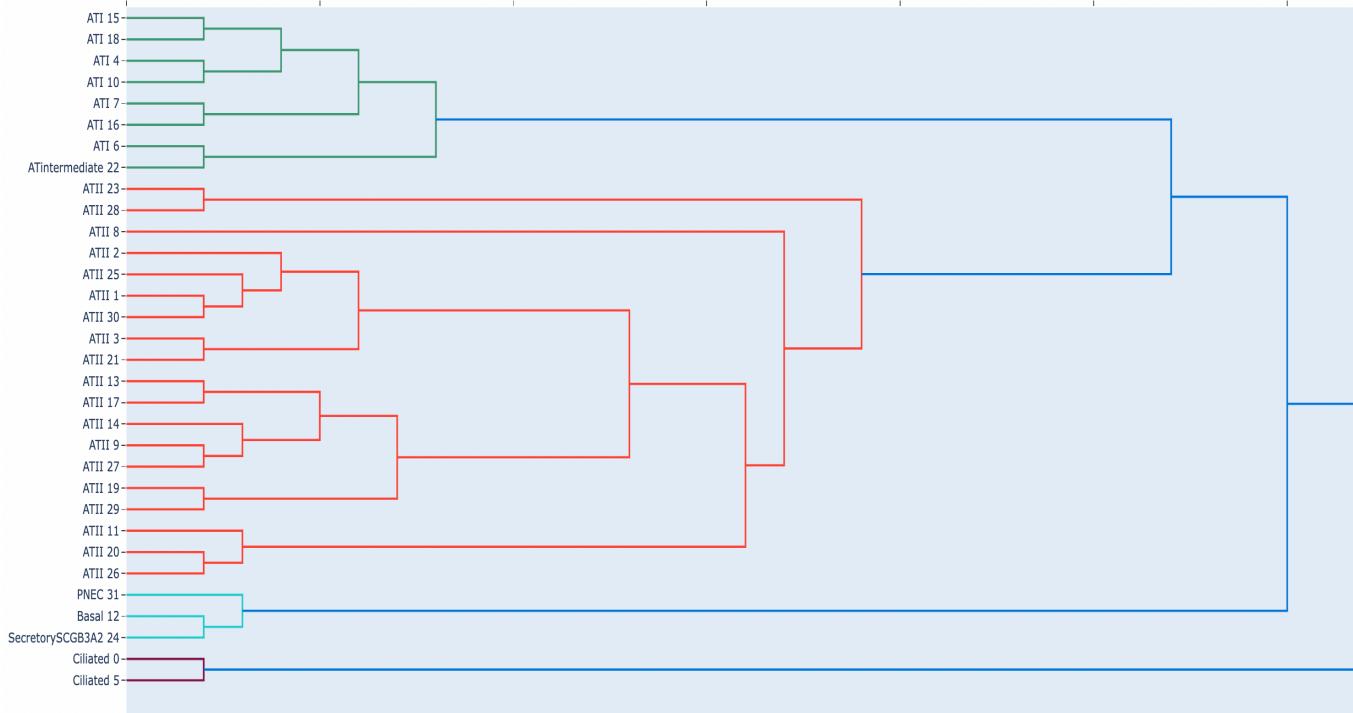
## Hierarchical clustering methods

- bottom-up hierarchical clustering steps:
  - Define the distance
    - Euclidean distance
  - Find a pair of closest elements and merge them
  - Iteratively find and merge a pair of closest elements



# Hierarchical clustering methods

- Applications

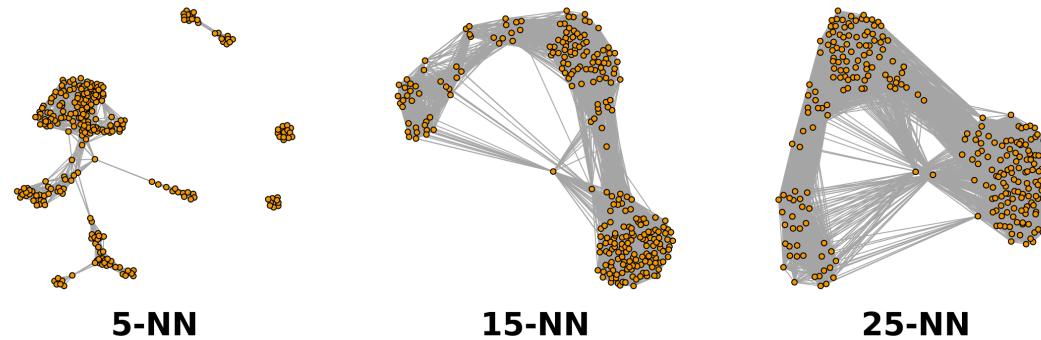


## Graph-based clustering (Shared-nearest-neighbors methods)

- Objective:
  - build a cell neighbors graph
    - cells -> nodes
    - similarities between cells -> edges
- Aim:
  - identify 'communities' of cells within the network
  - detect cluster
- Process:
  - Cell X and Y are connected
  - Find k cells with k-nearest distances from X
  - from Y to other cells for shared-NN

## Graph-based clustering (Shared-nearest-neighbors methods)

- Different neighbor-sizes



## Graph-based clustering (Shared-nearest-neighbors methods)

- Pros
  - fast and memory efficient
  - no assumptions on
    - the shape of the clusters
    - the distribution of cells within each cluster
    - number of clusters to identify
- Cons
  - loss of information beyond neighboring cells

## Graph-based clustering (Shared-nearest-neighbors methods)

- Modularity
  - measure the structure of graphs by measuring the strength of division of a network into clusters
- High modularity:
  - more dense within cluster
  - more sparse between clusters

## Graph-based clustering (Shared-nearest-neighbors methods)

- Applications:
- Louvain
- Leiden

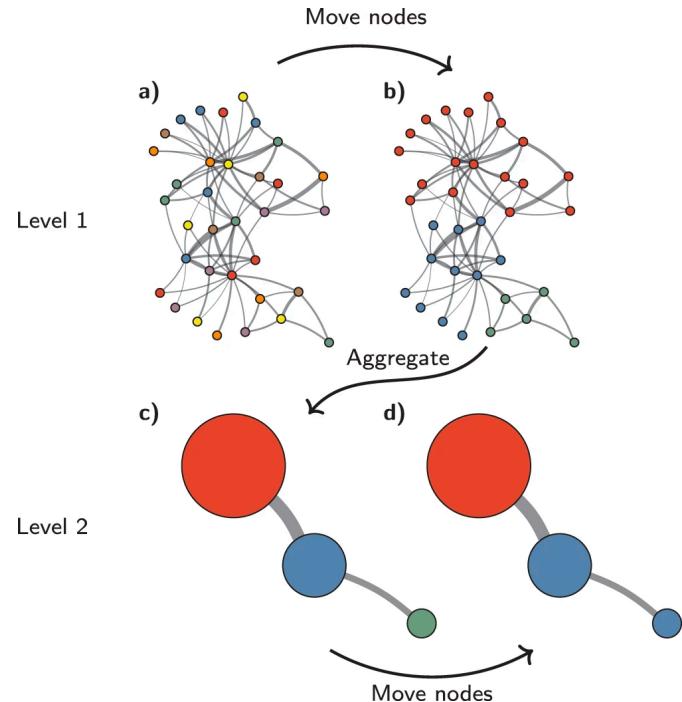
## Graph-based clustering (Shared-nearest-neighbors methods)

- Louvain clustering
  - hierarchical agglomerative idea
  - Nodes are first assigned their own community
  - Two-step iterations:
    - nodes are re-assigned one at a time to the community for which they increase modularity the most
    - a new, ‘aggregate’ network is built where nodes are the communities formed in the previous step.
  - This is repeated until modularity stops increasing.

## Graph-based clustering (Shared-nearest-neighbors methods)

- Louvain clustering

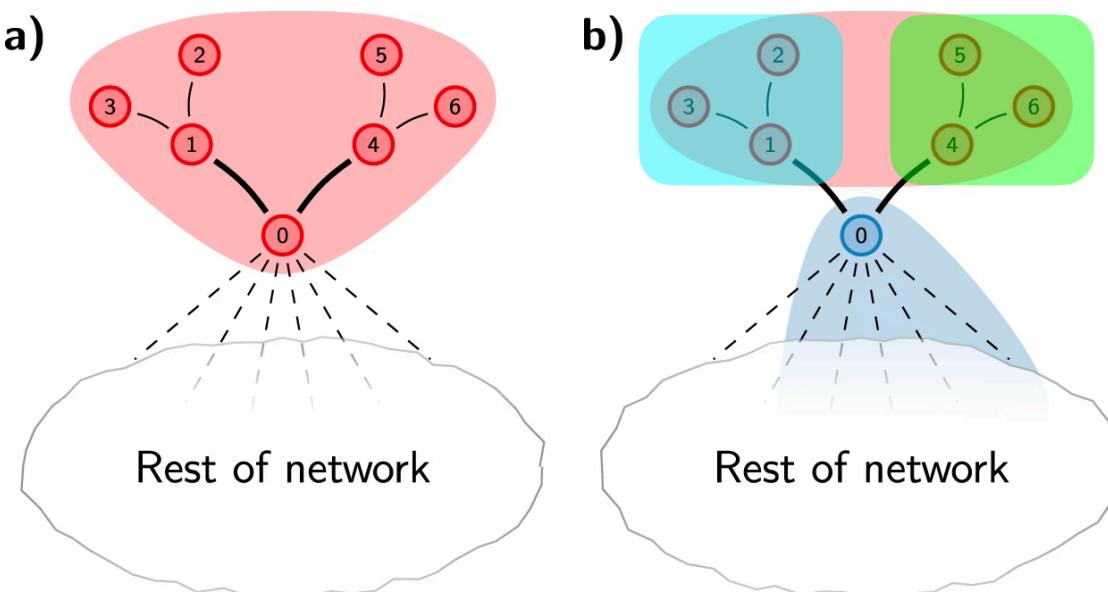
- each node is in its own community (partition)
- moves individual nodes from one community to another to find a partition
- create an aggregate network
- move individual nodes in the aggregate network



## Graph-based clustering (Shared-nearest-neighbors methods)

- Louvain clustering

- Issue:
  - Disconnected community



## Graph-based clustering (Shared-nearest-neighbors methods)

- Leiden clustering
  - add an extra step in iteration:
    - Refine the partition
      - one community may be split into multiple sub-communities
    - steps:
      - for each cluster/community/partition assign nodes to its own sub-community
      - Move nodes
      - locally merges nodes

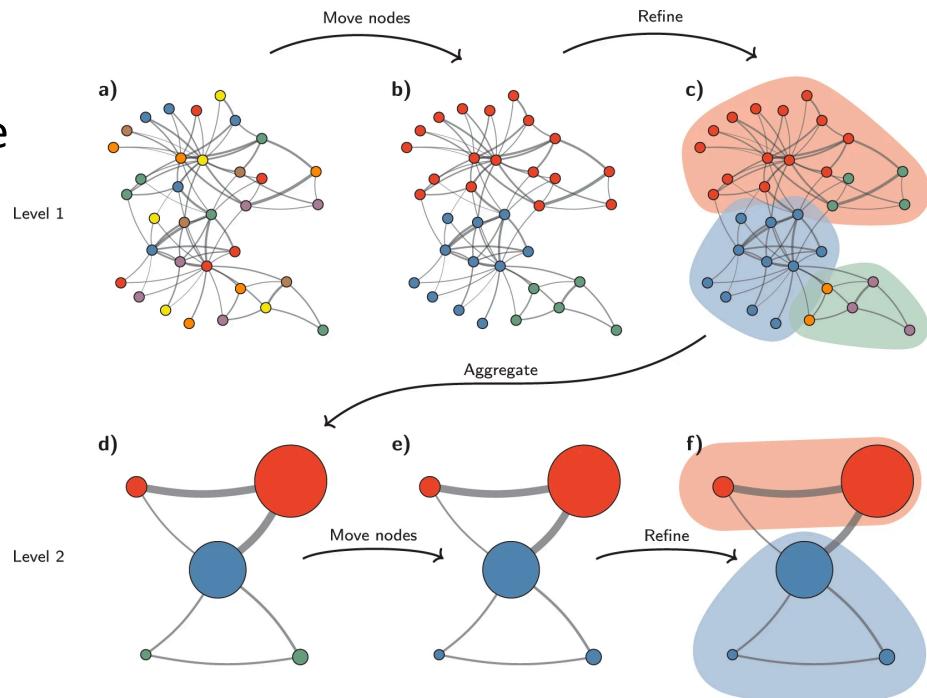
## Graph-based clustering (Shared-nearest-neighbors methods)

- Leiden clustering
  - speed up
    - only move nodes to connected communities
- Louvain:
  - try all communities
  - move to the one increase index the most

# Graph-based clustering (Shared-nearest-neighbors methods)

- Leiden clustering

- starts from a singleton partition
- moves individual nodes from one community to another to find a partition
- refine partition
- build an aggregate network
- moves individual nodes in the aggregate network
- refine the aggregate network



## Recap

- K-means clustering
- Hierarchical clustering
  - Agglomerative (bottom-up):
  - Divisive (top-down):
- Graph-based clustering
  - Louvain
  - Leiden

## Evaluate clustering results

- With labels:
  - Adjusted Rand index (ARI)
  - Mutual information based scores (MNI)
- Without labels:
  - silhouette coefficient

## Hands-on

- Scanpy
  - a python package

- Trajectory
  - Introduction to Trajectory
    - What is trajectory
    - Why do we need to infer trajectory
  - Inference methods
    - Gene expression inference
    - RNA-velocity based
  - Hands on
    - scvelo for single-cell trajectory inference

## Introduction

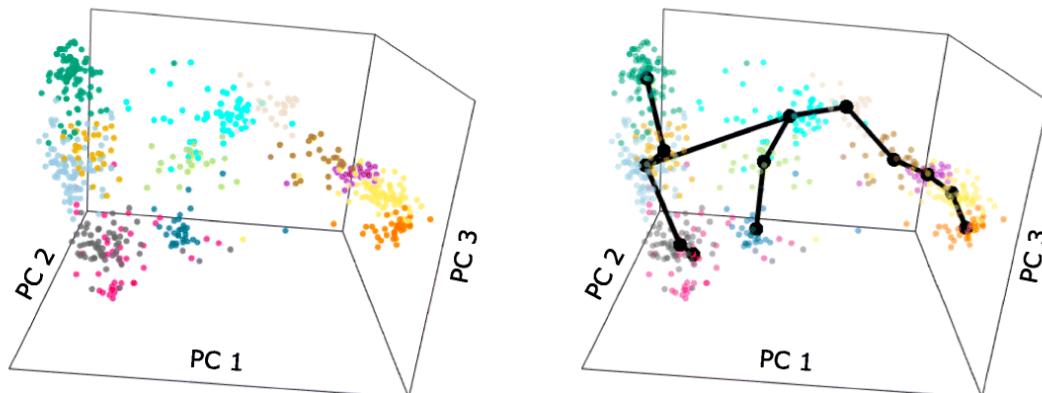
- Trajectory inference
  - computational technique
  - pattern of a dynamic process experienced by cells
  - arrange cells based on their progression through the process
  - after clustering and cell type annotation

## Introduction

- Trajectory inference
  - clustering
  - differences result from:
    - cell cycle
    - cell differentiation
    - response to an external stimuli
- characterize such differences by placing cells along a continuous path

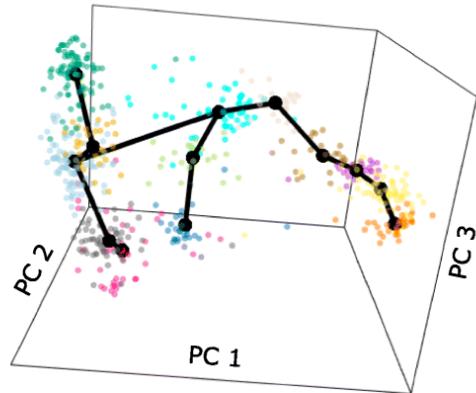
## Introduction

- Trajectory inference
  - characterize differences by
    - placing cells along a continuous path
    - represents the evolution of the process
    - instead of dividing cells into discrete clusters



# Introduction

- Trajectory inference
  - Pseudo-time analysis
    - start from the origin of cells
    - through different cellular states

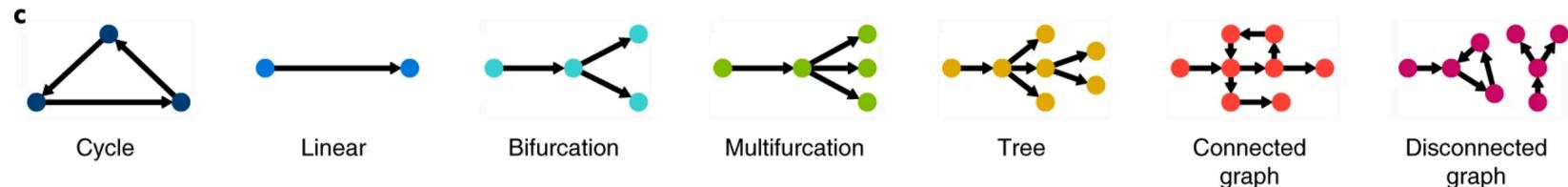


# Introduction

- Trajectory inference
  - Assumptions:
    - There is a developmental trajectory in the dataset
    - There are intermediate states in the dataset
    - Have branches in the trajectory or not
    - The dimensionality reduction and cell type annotation are good
  - Cautious:
    - ‘Trajectory’ can be inferred from any dataset without biological meaning

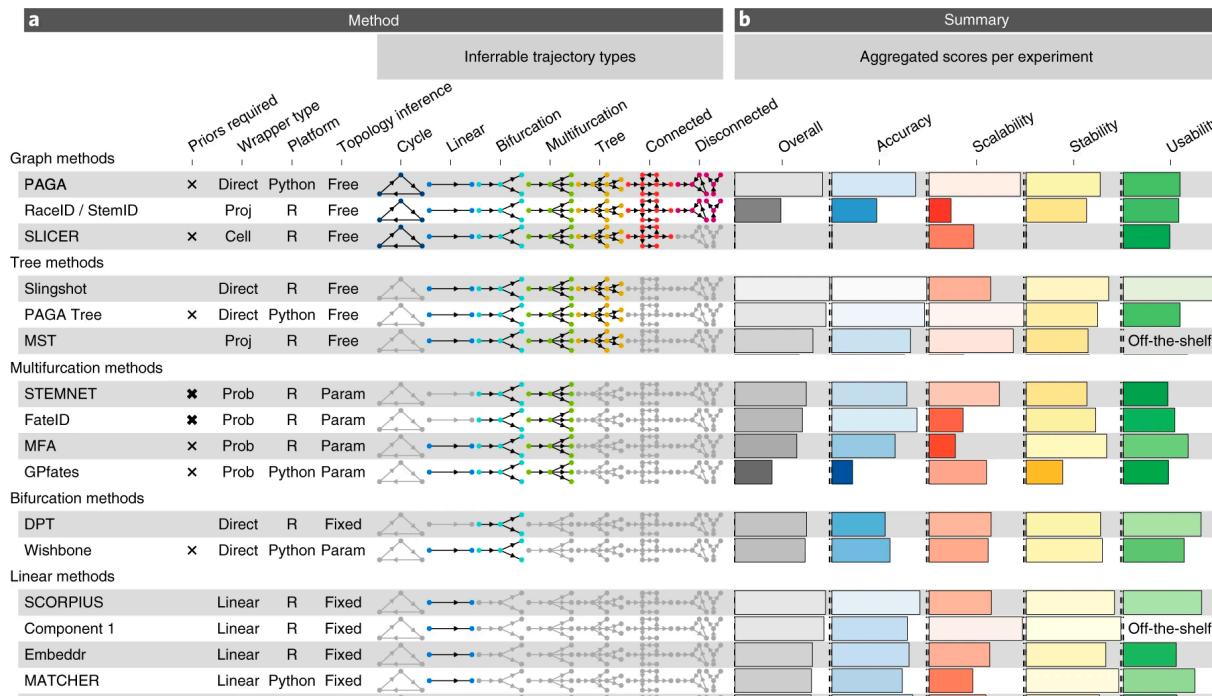
# Introduction

- Trajectory inference
  - Different trajectory topology
    - fixed topology: assumptions on topology
    - detected topology



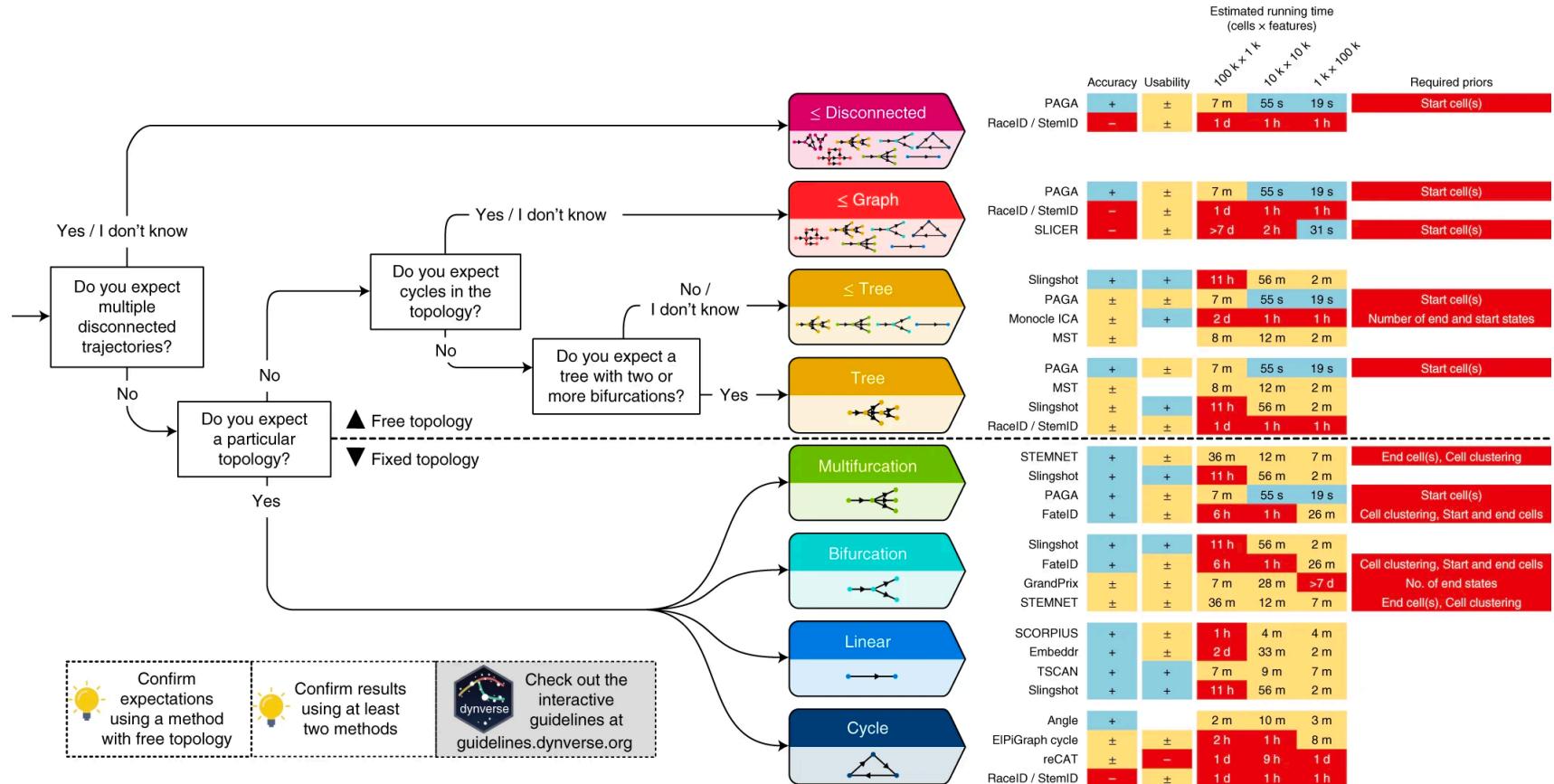
# Introduction

- Trajectory inference
  - Different trajectory inference methods



# Trajectory inference

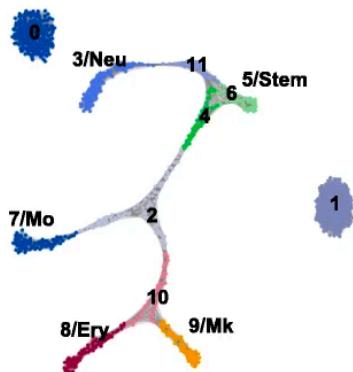
- Which methods to use?



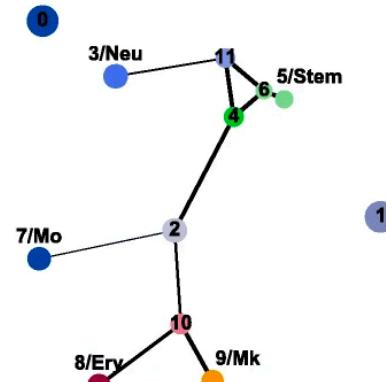
## Trajectory inference

- Partition-based graph abstraction (PAGA)
  - construct a symmetrized kNN-like graph
  - Euclidean distance as distance metric
  - nodes/partitions: cell populations
  - edges: distance/relationships

PAGA + embedding



PAGA graph

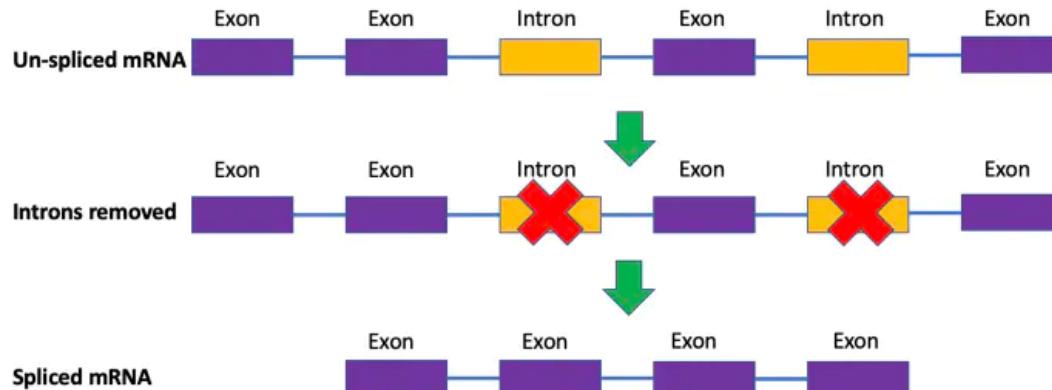


## RNA-velocity

- Velocity:
  - a vector quantity that designates how fast and in what direction a point is moving
  - speed: rate
  - velocity: vector (speed+direction)
- $v = \frac{ds}{dt}$ 
  - s: distance moved
  - v: velocity
  - t: time

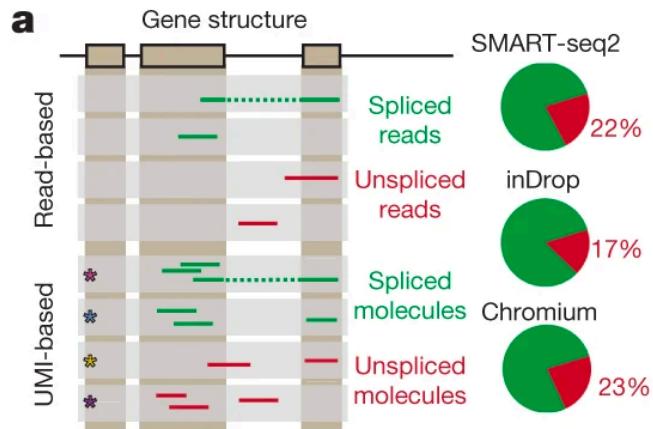
## RNA-velocity

- Eukaryotic pre-mRNA processing
  - unspliced mRNA
  - splice/remove introns
  - spliced mRNA



## RNA-velocity

- RNA velocity in trajectory inference
  - unspliced mRNA: nascent
  - spliced mRNA: mature
  - use unspliced RNA content as a proxy for spliced RNA content in the near future

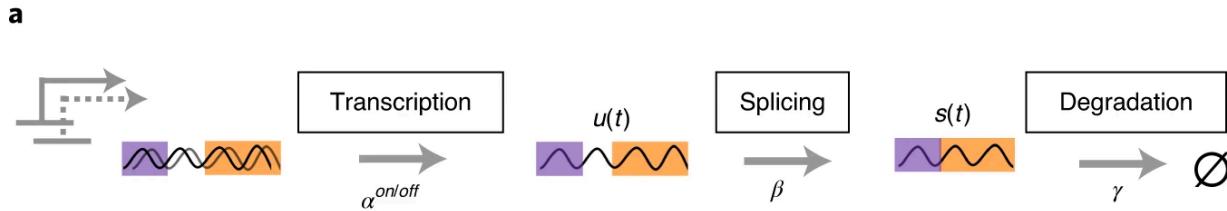


## RNA-velocity

- RNA velocity
  - the time derivative of the gene expression state
  - unspliced mRNA reads  $u$
  - spliced mRNA reads  $s$
- $\frac{du}{dt}$ : unspliced mRNA velocity
- $\frac{ds}{dt}$ : spliced mRNA velocity

## RNA-velocity

- RNA velocity in trajectory inference



- at time t
- spliced mRNA reads changes =

splicing rate \* unspliced mRNA reads - degradation rate \* spliced mRNA reads

## RNA-velocity

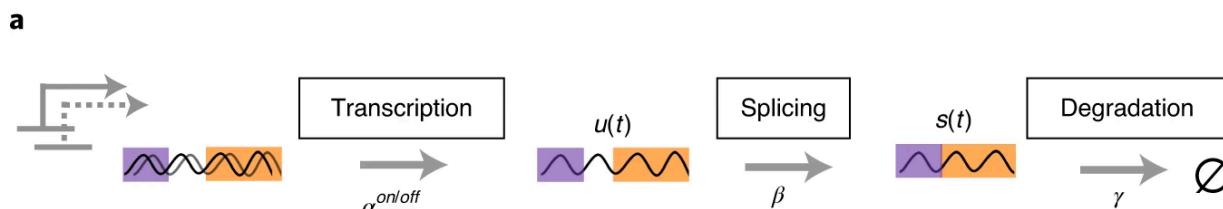
- RNA velocity in trajectory inference

- the time derivative of the gene expression state
- capturing transcription  $\alpha$
- splicing rate  $\beta$
- degradation rates  $\gamma$
- unspliced mRNA reads at time t  $u(t)$
- spliced mRNA reads at time t  $s(t)$

$$\frac{du}{dt} = \alpha(t) - \beta(t)u(t)$$

$$\frac{ds}{dt} = \beta(t)u(t) - \gamma(t)s(t)$$

The rate equations for a single gene



## RNA-velocity

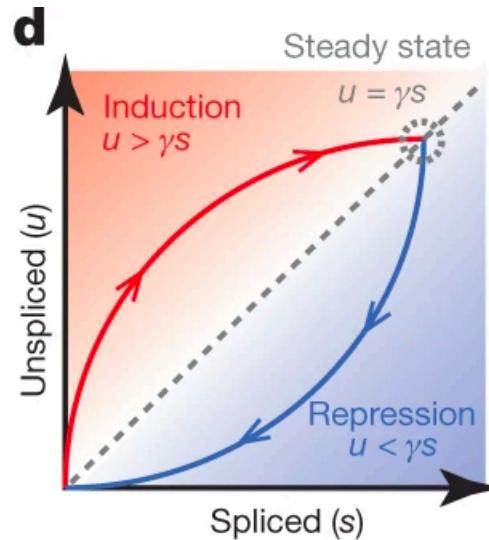
- RNA velocity in trajectory inference
  - the time derivative of the gene expression state
  - capturing transcription  $\alpha$
  - splicing  $\beta = 1$  (i.e. measuring all rates in units of the splicing rate),
  - degradation rates  $\gamma$
  - unspliced mRNA  $u(t)$
  - spliced mRNA  $s(t)$

$$\frac{ds}{dt} = \beta(t)u(t) - \gamma(t)s(t)$$

$$\frac{ds}{dt} = u - \gamma s \quad \text{steady state: } \frac{ds}{dt} = 0$$

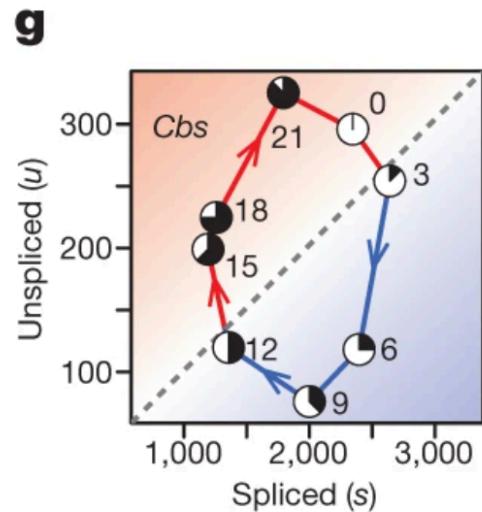
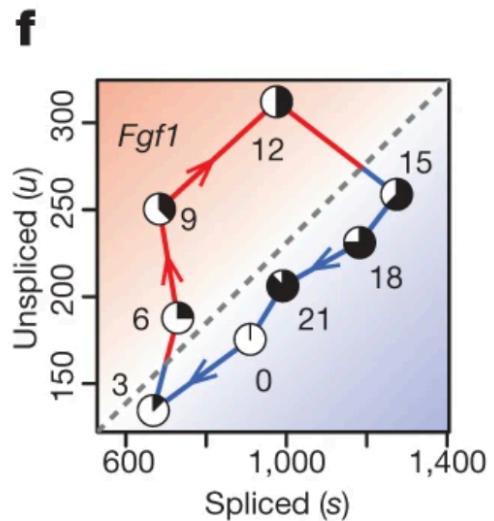
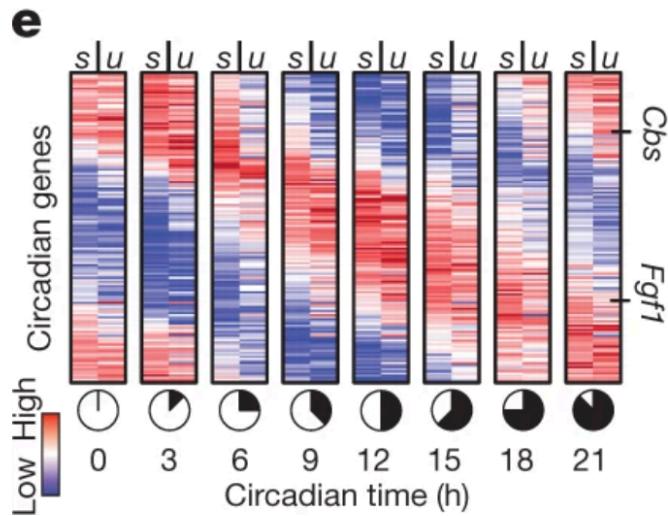
$$0 = u - \gamma s$$

$$u = \gamma s$$



## RNA-velocity

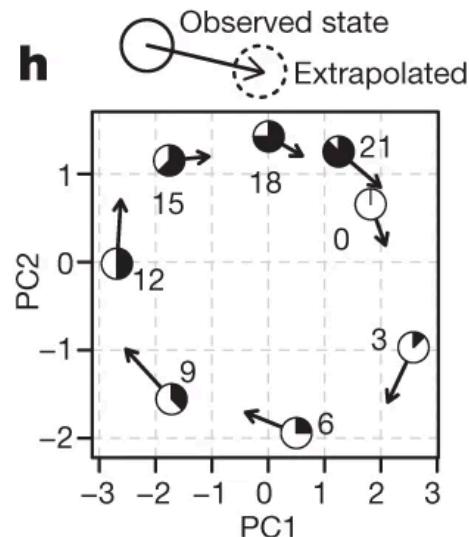
- RNA velocity in trajectory inference
  - examples in circadian time



## RNA-velocity

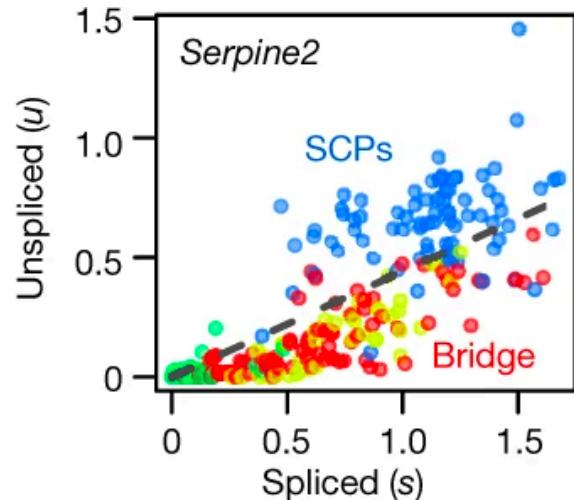
- RNA velocity in trajectory inference

- Change in expression state at a future time  $t$ , as predicted by the model
- arrow pointing to the position of the future state



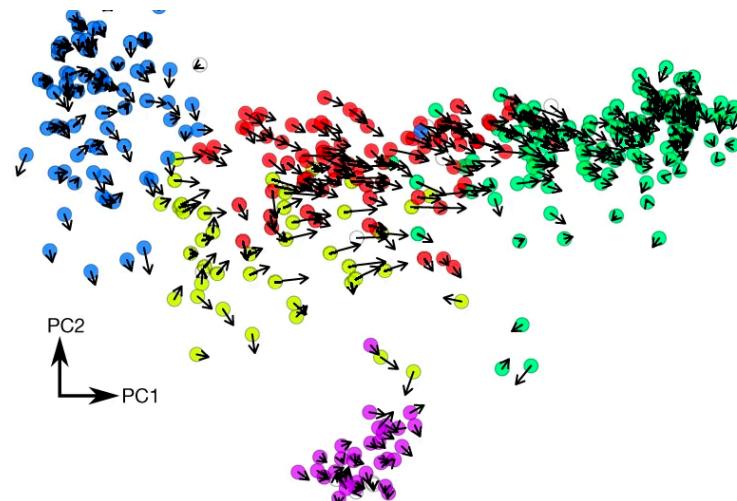
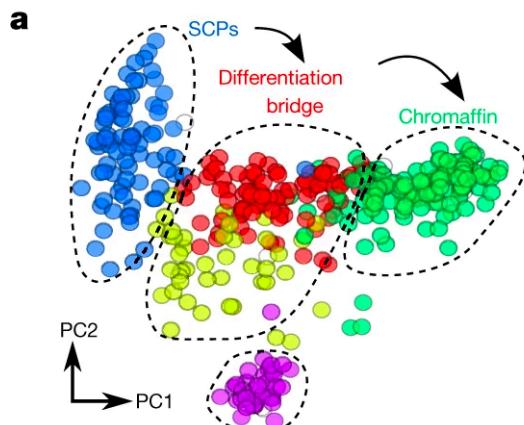
## RNA-velocity

- RNA velocity in trajectory inference
  - steady-state ratio
  - above the ratio line
    - increase expression of gene
  - below the ratio line
    - decrease expression of gene



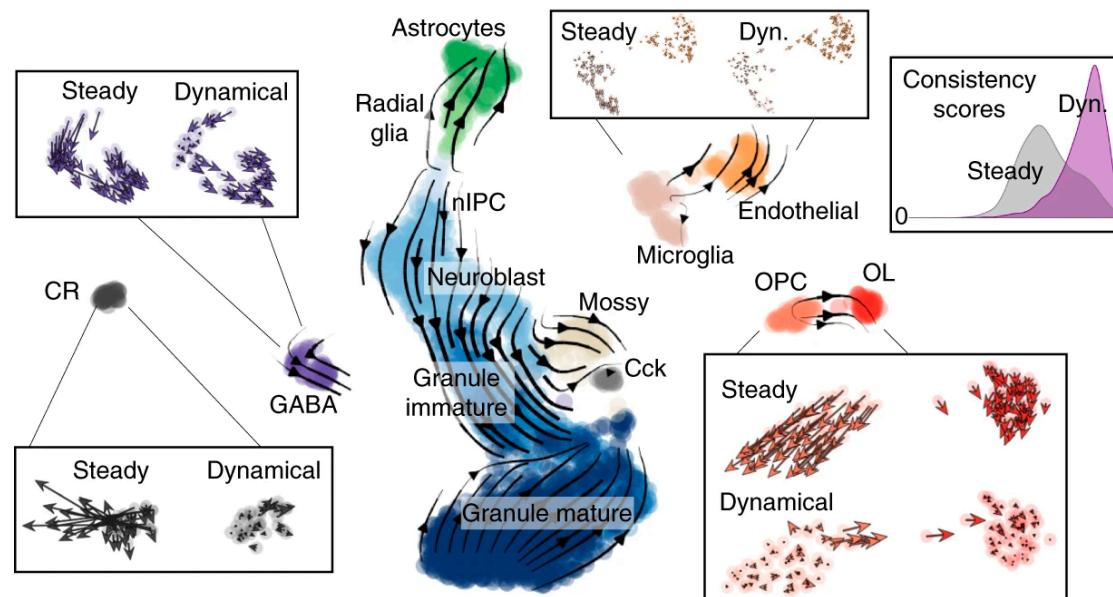
## RNA-velocity

- RNA velocity in trajectory inference
  - Arrows: extrapolated state given observed state
  - Different arrow directions: different extrapolated states



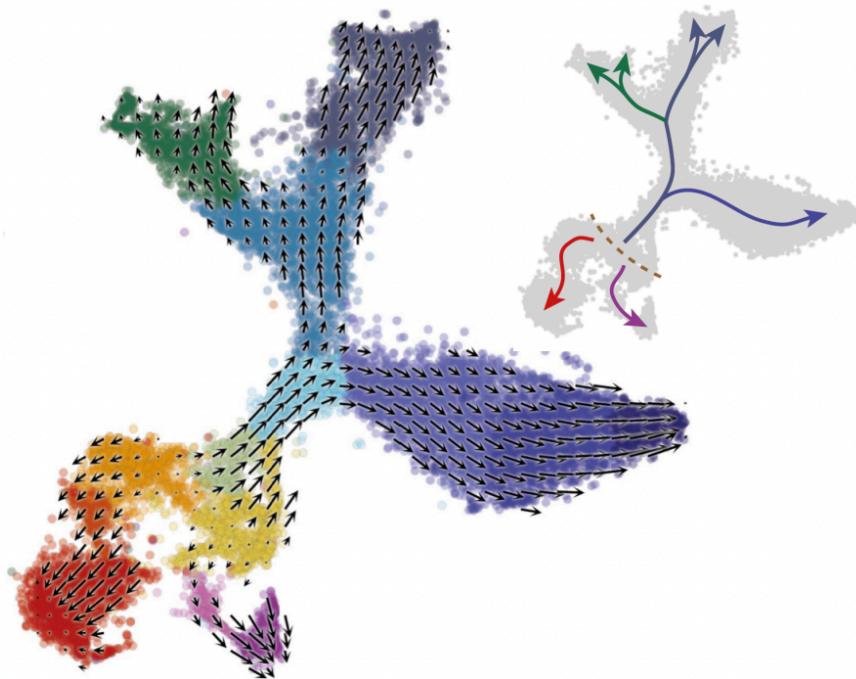
## RNA-velocity

- RNA velocity in trajectory inference
  - steady and dynamical arrows
  - different lengths: different speed rate



## RNA-velocity

- RNA velocity in trajectory inference
  - Trajectory based on RNA velocity



## Hands-on

- scvelo