scRNA-seq clustering and trajectory inference

Yumin Zheng

Dec. 1, 2022

Agenda

- 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 in single-cell
  - Dimensionality Reduction
    - Principal Components Analysis
  - Clustering methods
    - K-means based methods
    - Shared-nearest-neighbors methods
    - Hierarchical clustering 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

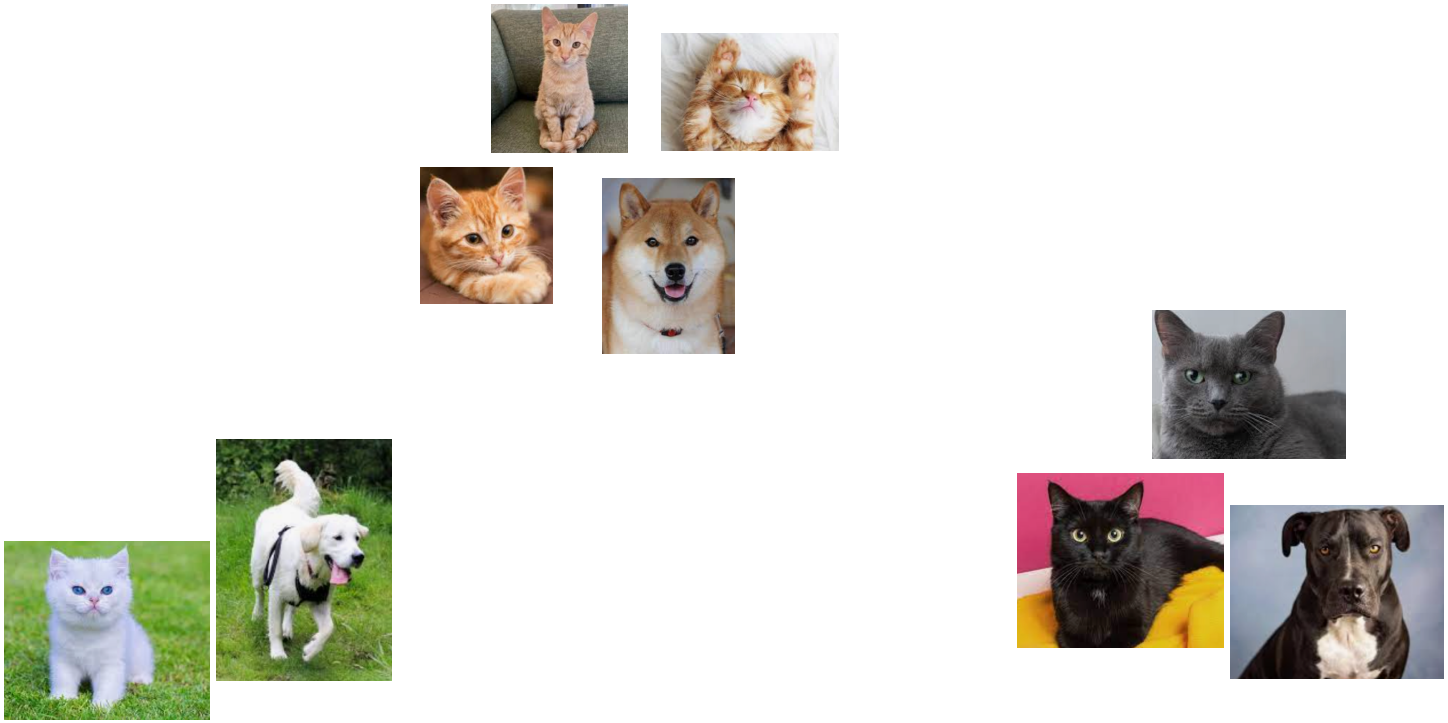
- Clustering vs Classification

- 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

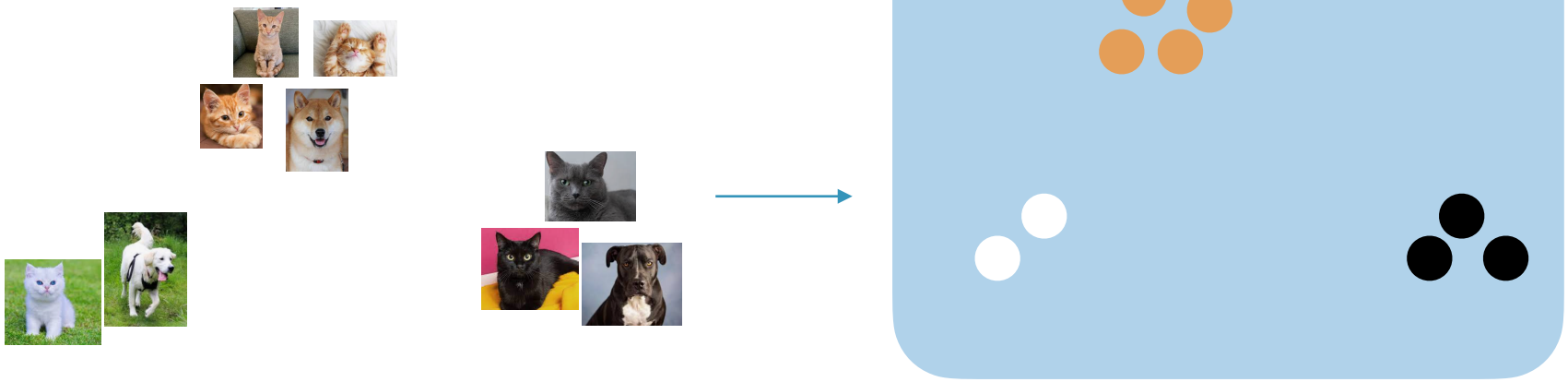
```
>NS500724:80:HWFYWBGXF:1:11101:10427:1039 1:N:0:143
GAACANACAGGCTGAGGCGTTGTGGAAGGGCTGGGCGAAGTCGCCGGACATTTCAGGCGCGCGGACGTTCCTAGATTTGCAGATTGCCCTTTGCCGTGAGTTGCTACAGACGGGCGAATGTTGGGTGCTCAAAACCCTGTCTCTTATAC
>NS500724:80:HWFYWBGXF:1:11101:26792:1051 1:N:0:143
GGCTTNTACCCAACACTCAATATCTCAGATGAGTTTTCTAGCAATGTTGCAAATTATCAAAAGGTTGGTATGCAAAAGTATTCTACCTGTCTCTTATACACATCTCCGAGCCCACGAGACGTTGGATGAAATATCGTATGCCGTCTTAG
>NS500724:80:HWFYWBGXF:1:11101:17792:1055 1:N:0:143
CCTTANAGCTGTTACAATAAGAGGCCATGCTAAATTAGGTGAATTGTCCATACTAATTTCACTAAGTTGAACAATTTTACTATCTGCATCTACAACCTGTTGGATTTCCCACAATGCTGATGCATAAGTAAATGTTGTACCATCACACG
>NS500724:80:HWFYWBGXF:1:11101:12348:1058 1:N:0:143
CACTTNCCTCGTGAAGGTGTCTTTGTTTCAAATGGCACACACTGGTTTGTAACACAAAGGAATTTTTATGAACCACAAATCATTACTACAGACAACACATTTGTGTCTGTCTCTTATACACATCTCCGAGCCCACGAGACGTTGGATGA
>NS500724:80:HWFYWBGXF:1:11101:19632:1072 1:N:0:143
GGTTAAGTGTCGTCTGCAACGAGCCGATGTTCAGTGCCCCAGAGAGACTCGGAAGTGGTGACCCGACCGAGCCGCCTGACAGGATCGTCCCGAGCTGCGACGCGAAATCCACCACATCATCGATTGCGTCTTCGATTGTGGCGCGAGCC
```

- Alignment and generate count matrix

$$\begin{bmatrix} 0\,0\,10\,2 \ldots 0\,3 \\ 1\,3 \quad 9\,3 \ldots 8\,0 \\ . \quad . \quad . \; . \ldots \; . \; . \\ 5\,9 \quad 3\,1 \ldots 5\,5 \end{bmatrix}$$

Introduction

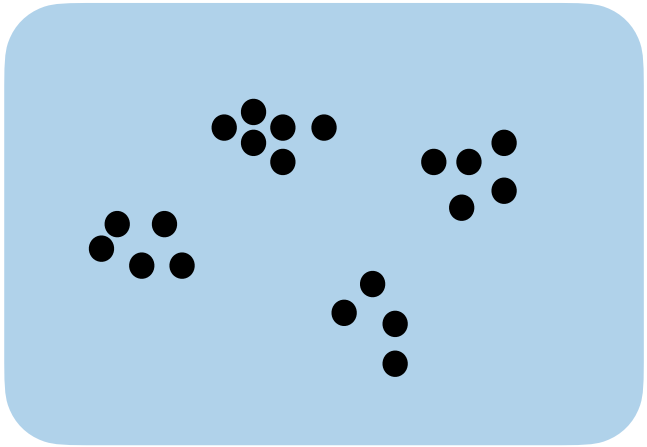
- Features



Dimensionality reduction

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

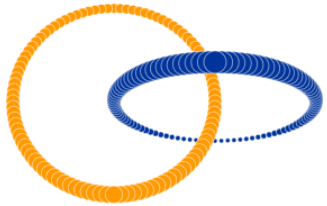
$$\begin{bmatrix} 0\,0\,10\,2\,\ldots\,0\,3 \\ 1\,3\;\;9\,3\,\ldots\,8\,0 \\ .\;\;.\;\;.\;\;.\ldots\;\;.\;\;. \\ 5\,9\;\;3\,1\,\ldots\,5\,5 \end{bmatrix}$$



Dimensionality reduction

- Linear
  - Principal Components Analysis (PCA)

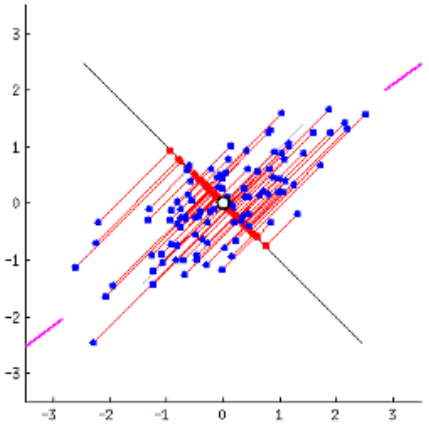
- Non-linear
  - t-SNE/UMAP
  - Auto-encoder



Non-linearly separable data

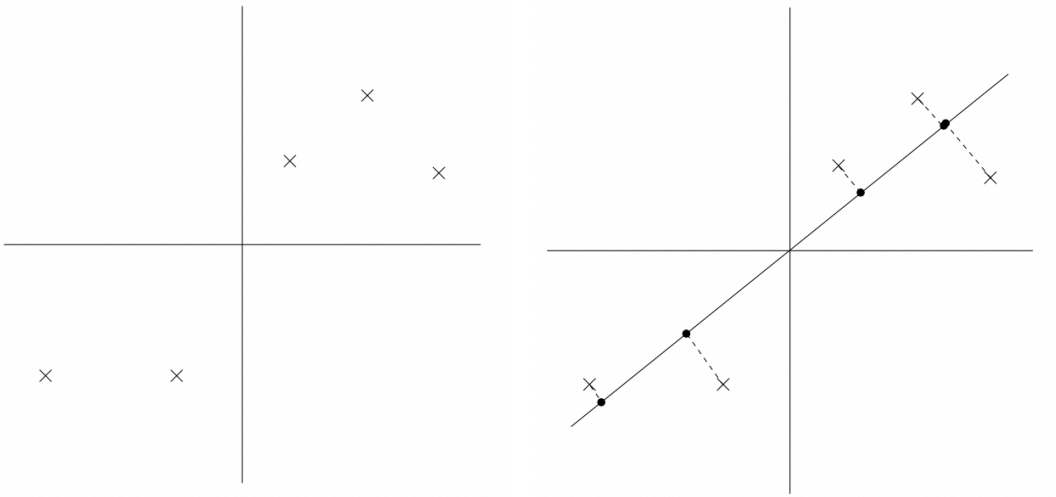
Dimensionality reduction

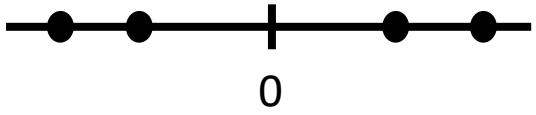
- Principal components analysis (PCA)
  - Find a linear transformation to project data from high-dimension to low-dimension space that maximize the variance between data points.



Dimensionality reduction

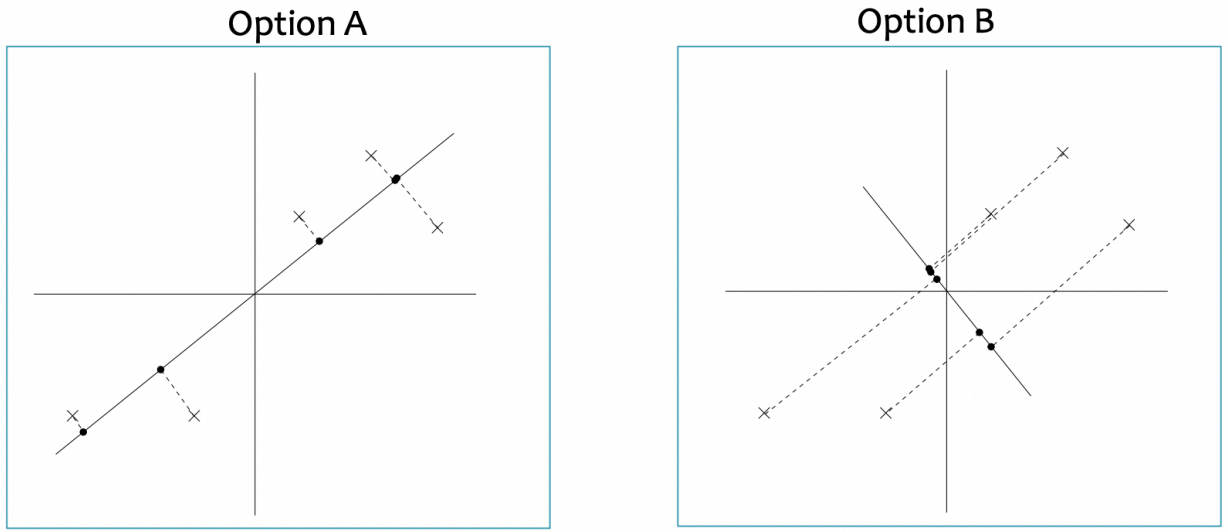
- Principal components analysis (PCA)
  - projection





Dimensionality reduction

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



Dimensionality reduction

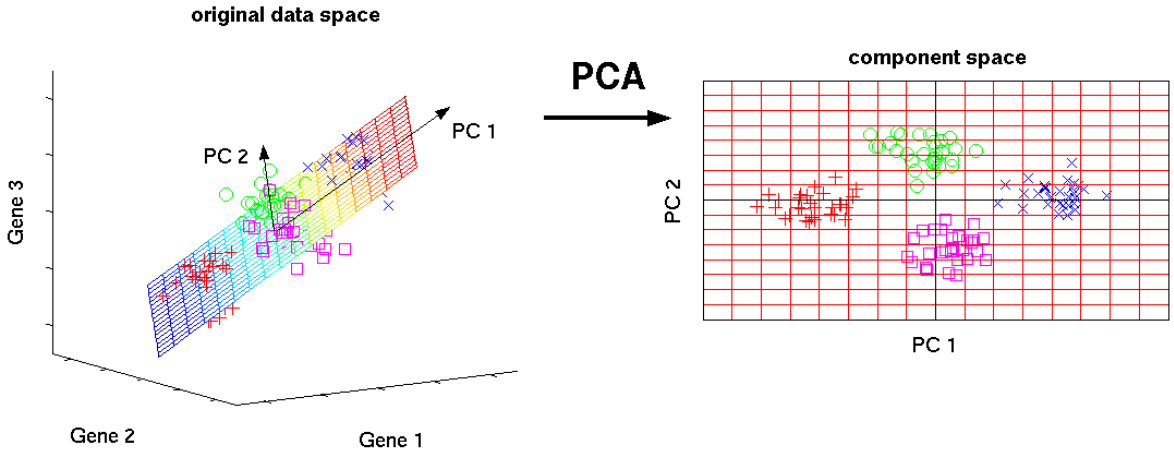
- Principal components analysis (PCA)
  - why we need the maximum variance component?

option A

option B

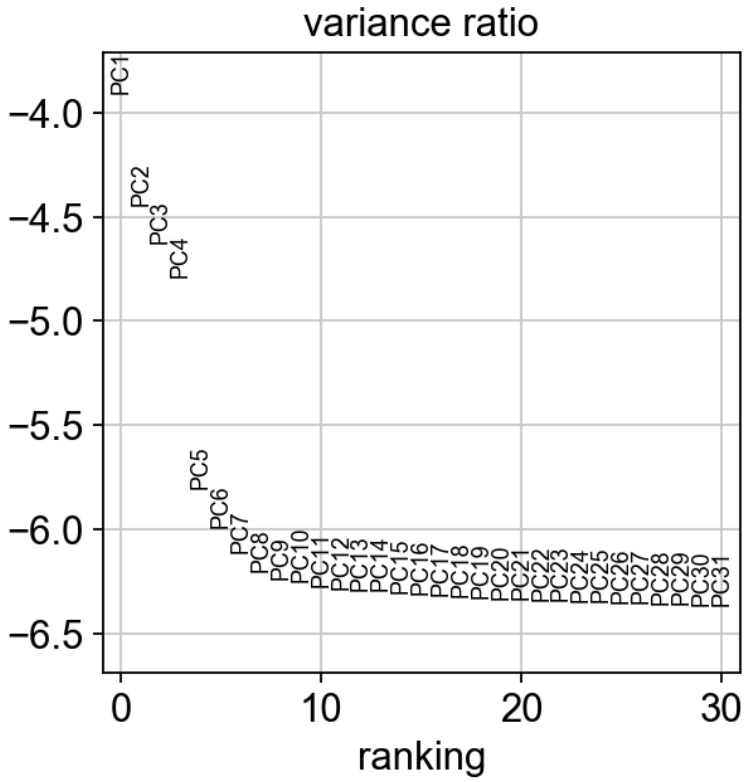
Dimensionality reduction

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



Dimensionality reduction

- Principal components analysis (PCA)
  - Variance



Clustering methods

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

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

- partition n cells into S groups

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

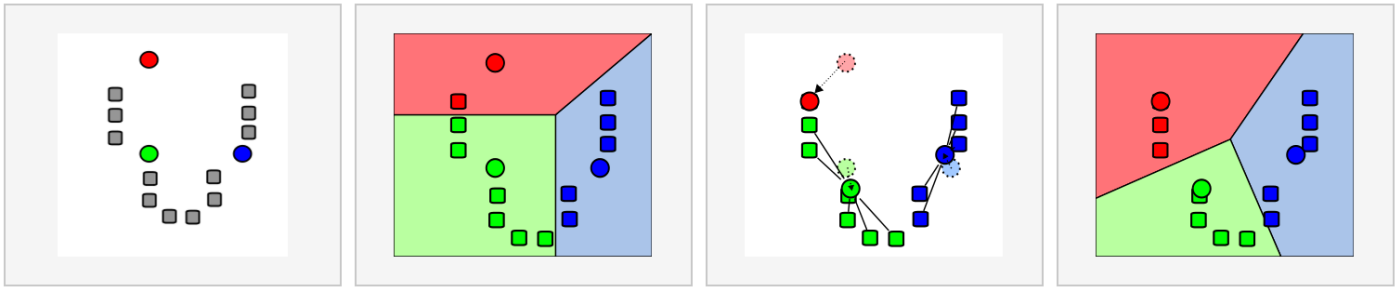
$(x_1, x_2, \ldots x_n)$: observations (cells)

$x_n = [g_1, g_2, .. g_d]$: genes

μ_i: the centroid point in S_i

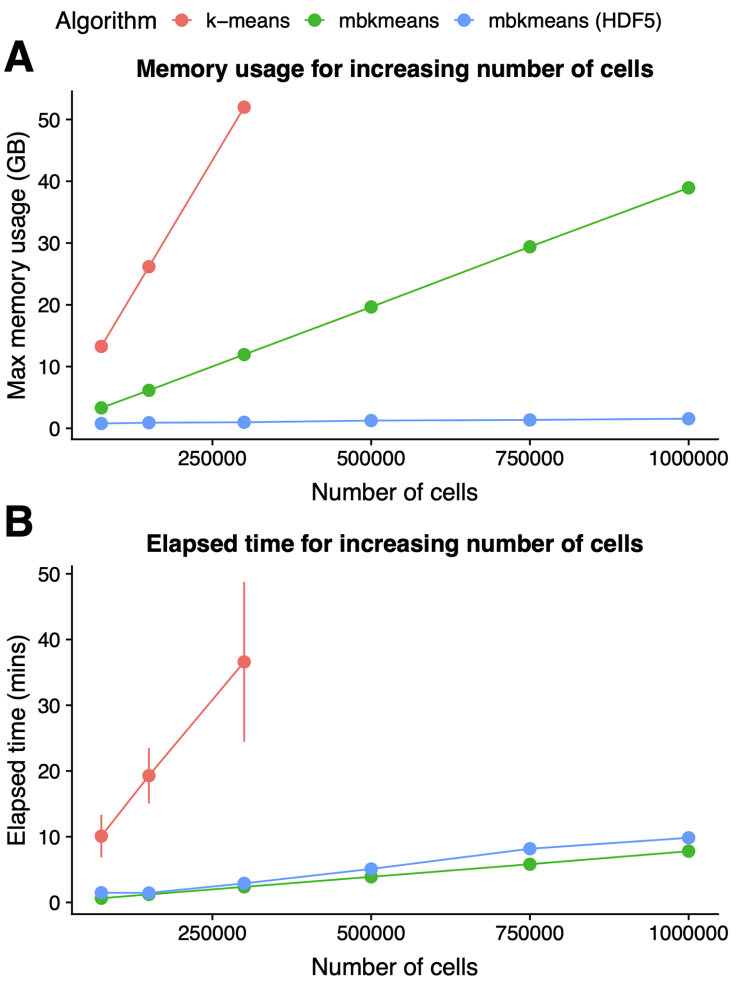
K-means clustering methods

- 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



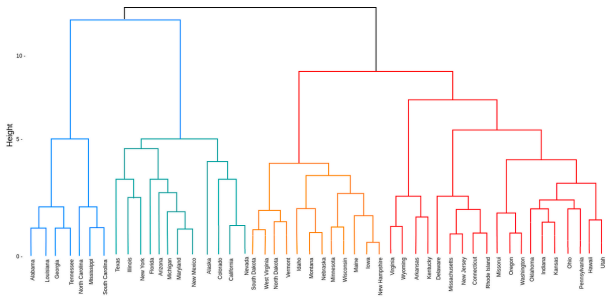
K-means clustering methods

- Applications
- mbkmeans
  - mini-batch k-means
  - k-means:
    - load all data
    - memory intense
  - load small batch
  - more economical



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.

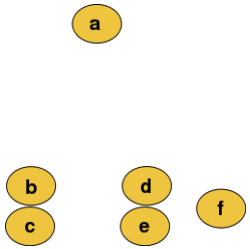
Hierarchical clustering methods

- Pros:
  - deterministic method
  - returns partitions at all levels along the dendrogram

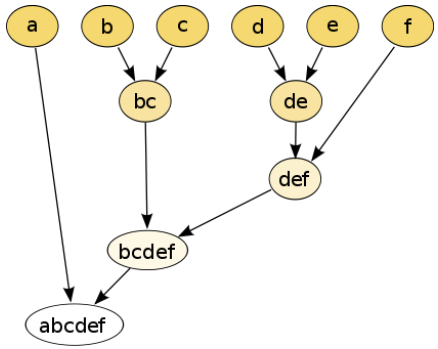
- Cons:
  - computationally expensive in time and memory
    - extra cost increase proportionally to the square of the number of data points

Hierarchical clustering methods

- bottom-up hierarchical clustering example



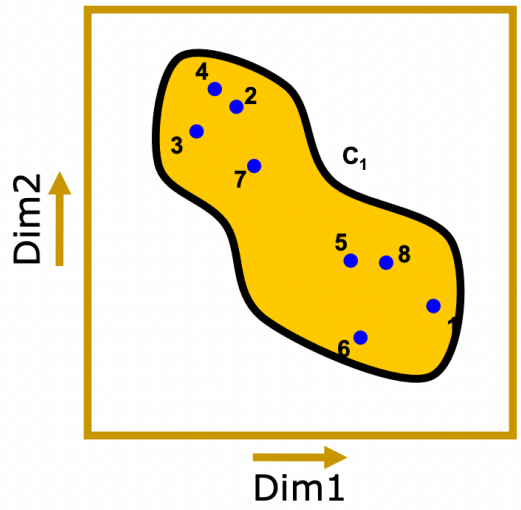
Data

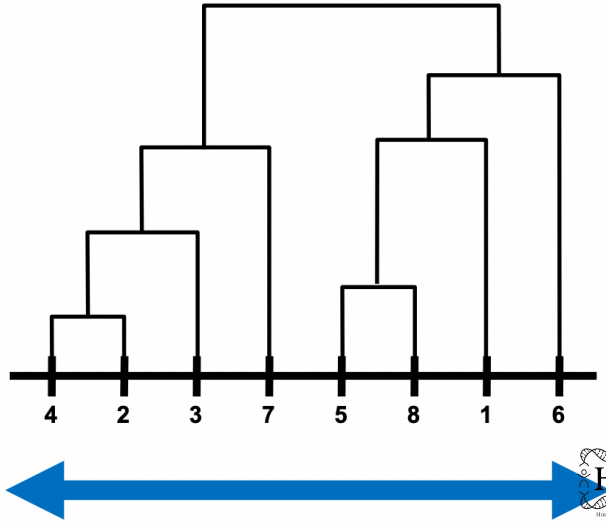


hierarchical clustering results

Hierarchical clustering methods

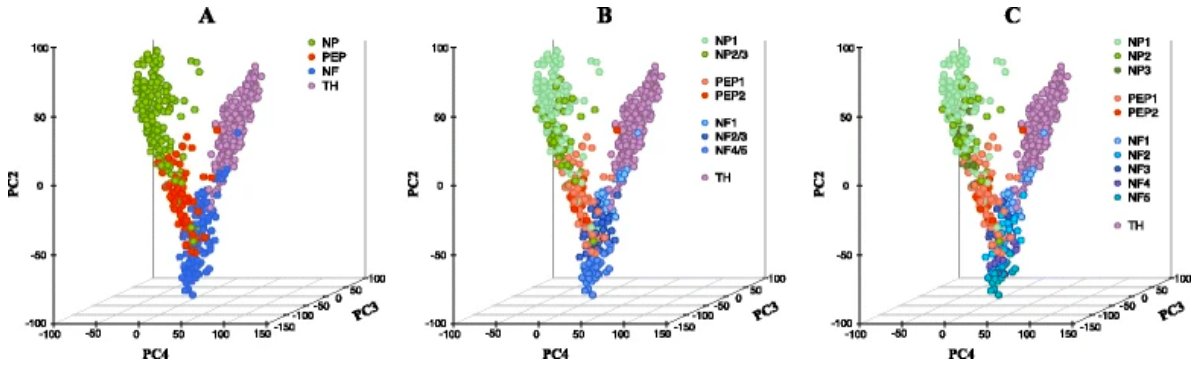
- bottom-up hierarchical clustering steps:
  - Define the distance
    - Euclidean distance
  - Find a pair of closest elements and merge them
  - Iteratively find and merge a pair of closest elements





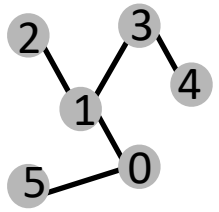
Hierarchical clustering methods

- Applications
  - pcaReduce
    - generate a cell state hierarchy where each cluster branch is associated with a principal component of variation that can be used to differentiate two cell states



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

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

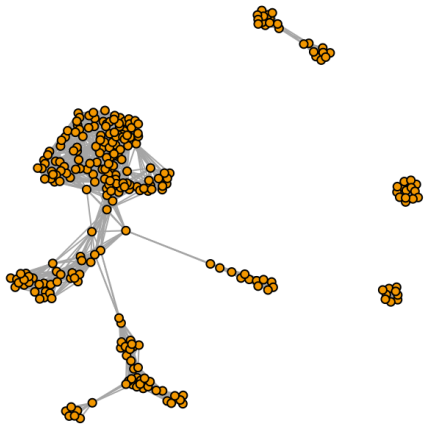
- shared neighbors network:
  - build a KNN graph
  - X and Y are k-nearest neighbors of each other
  - connect X and Y
  - similarity between X and Y:
    - X_neighbors ∩ Y_neighbors
  - remove edges if similarity is lower than a threshold

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

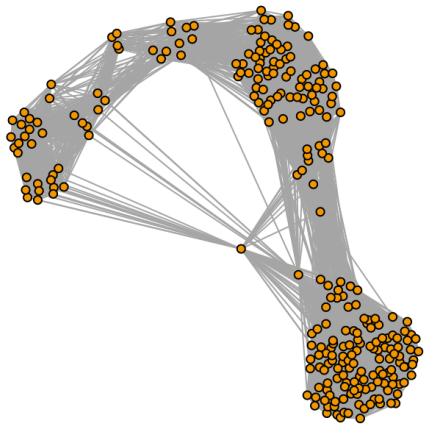
- Steps
  - construct the SNN graph
  - find SNN representative nodes
    - density/similarity within SNN higher than a threshold
  - discard points of density lower than a threshold (noise nodes)
  - align non-noise non-representative nodes to clusters

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

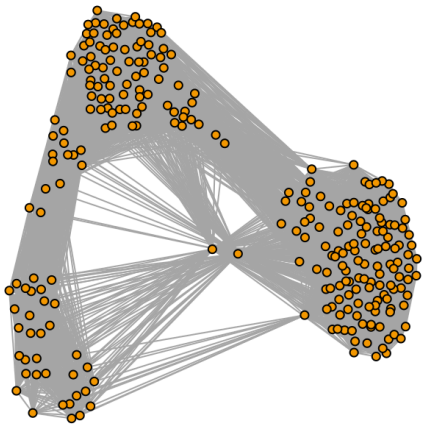
- Different neighbor-sizes



5-NN



15-NN



25-NN

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)

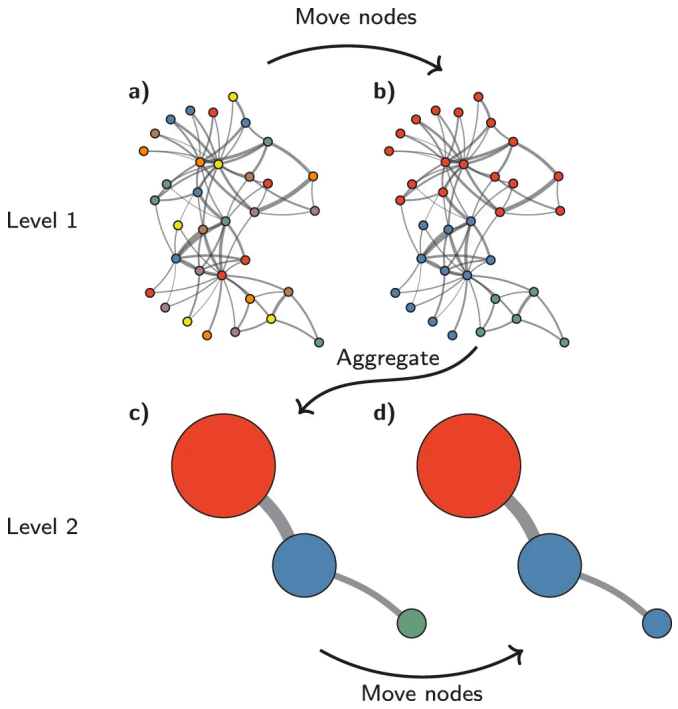
- Applications:
  - Louvain
  - Leiden

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 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.
  - Repeated until modularity stops increasing.

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

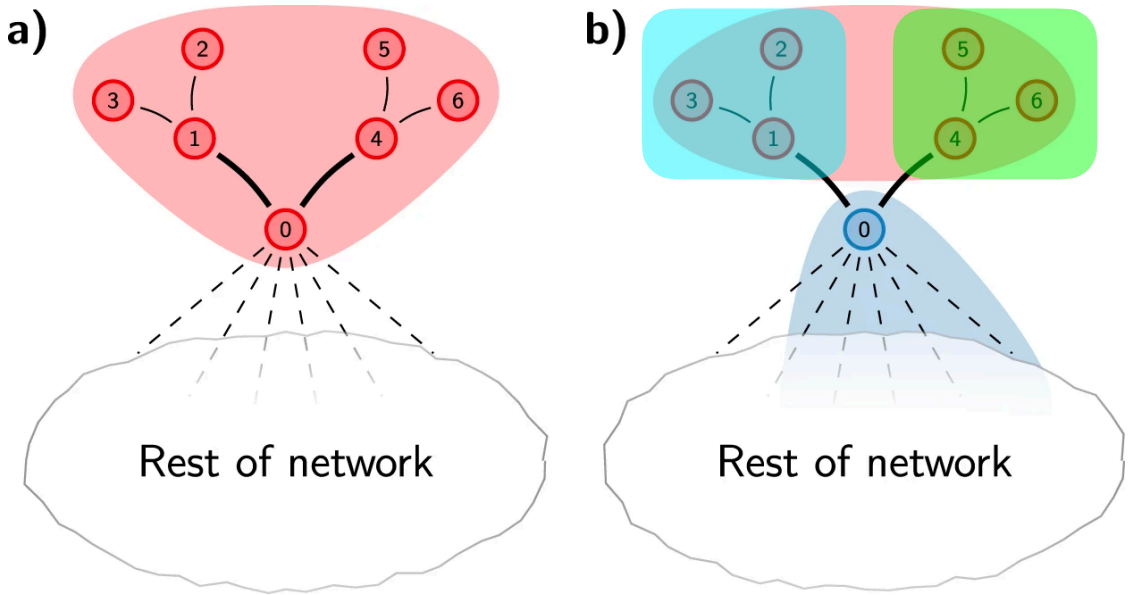
- Louvain clustering
  - each node is in its own community (partition)
  - moves individual nodes from one community to another
  - create an aggregate network
  - move individual nodes in the aggregate network



- Nodes in the aggregate network:
  - one cluster as a node

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

- Louvain clustering
  - Issue:
    - Disconnected community

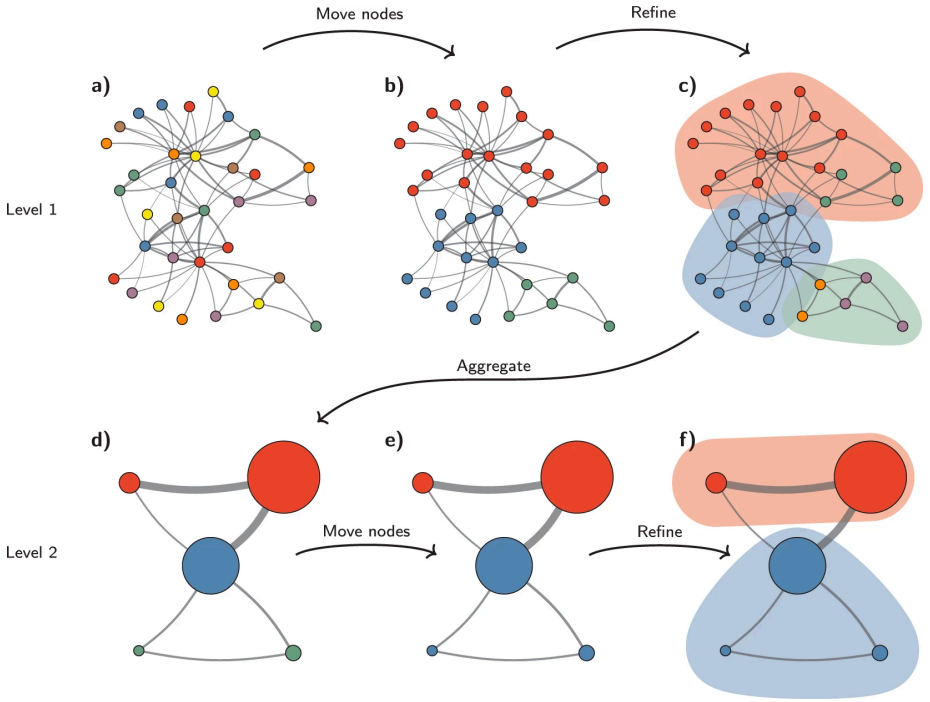


Graph-based clustering (Shared-nearest-neighbors methods)

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

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



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

Evaluate clustering results

- With labels:
  - Adjusted Rand index (ARI)
    - [-1, 1]
  - Normalized mutual information score (MNI)
    - [0, 1]

- Without labels:
  - silhouette coefficient
    - [-1, 1]

Recap

- K-means clustering

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

Hands-on

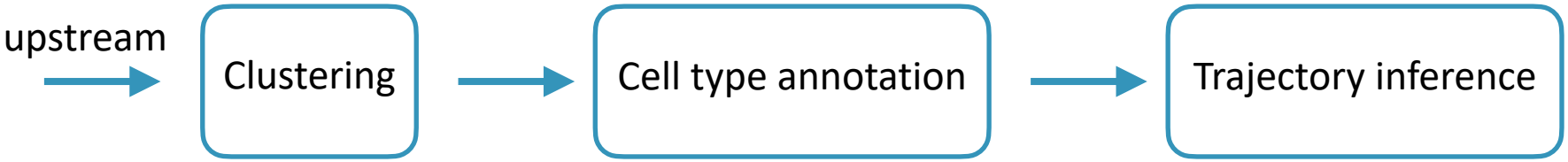
- scanpy
  - graph-based clustering

- sklearn
  - k-means clustering
  - hierarchical clustering

- Trajectory Inference
  - Introduction to Trajectory
    - What is trajectory
    - Why do we need to infer trajectories in single cell data
  - Inference methods
    - Gene counts based trajectory
    - RNA-velocity based trajectory
  - 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

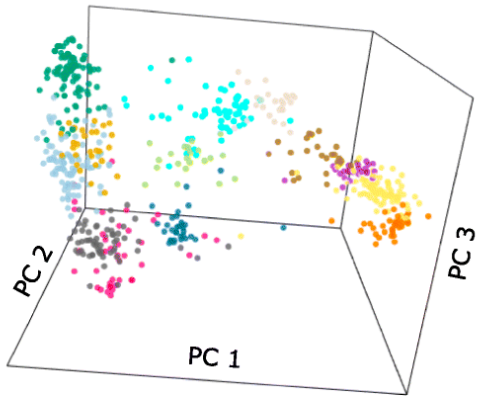


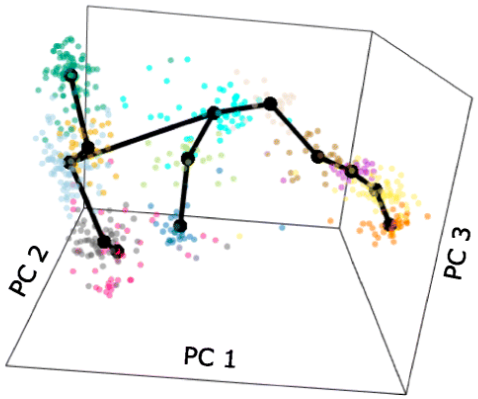
Introduction

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

Introduction

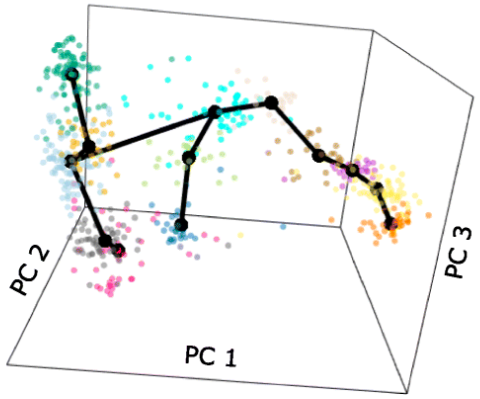
- Trajectory inference
  - characterize differences by
    - placing cells along a continuous path
  - represents the evolution of the process
  - instead of dividing cells into discrete clusters





Introduction

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

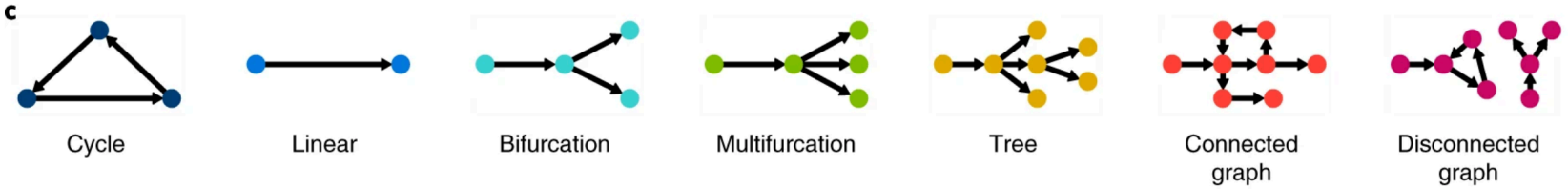


Introduction

- 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

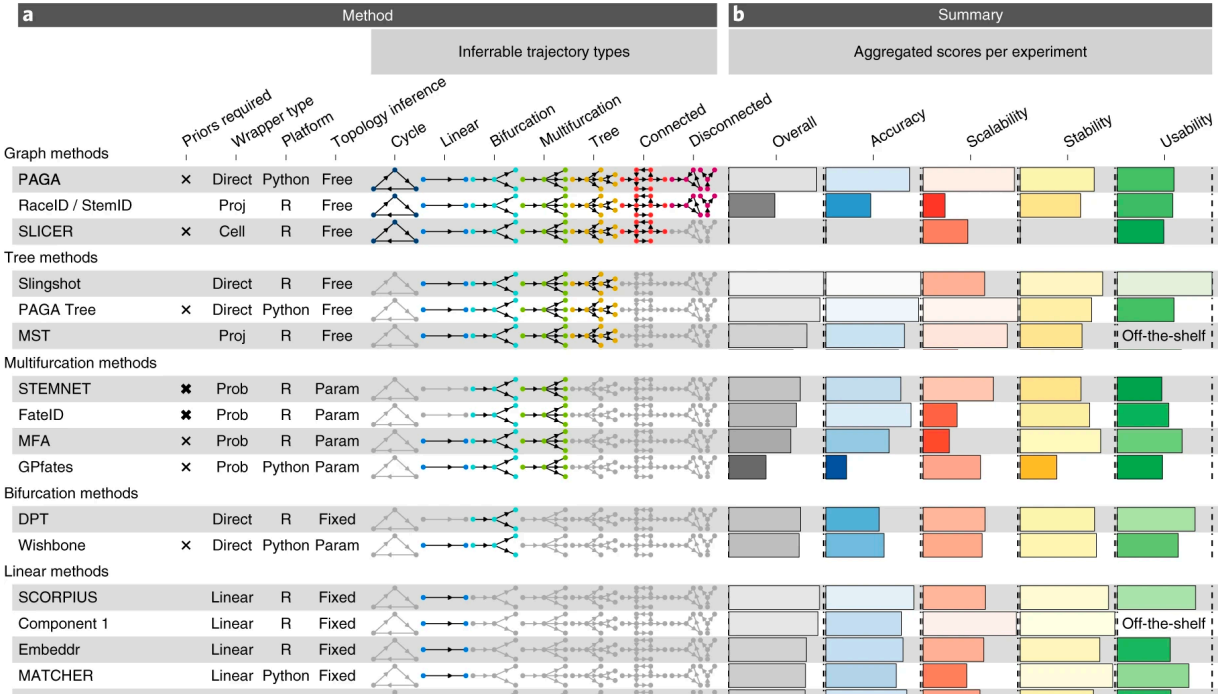
Introduction

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



Introduction

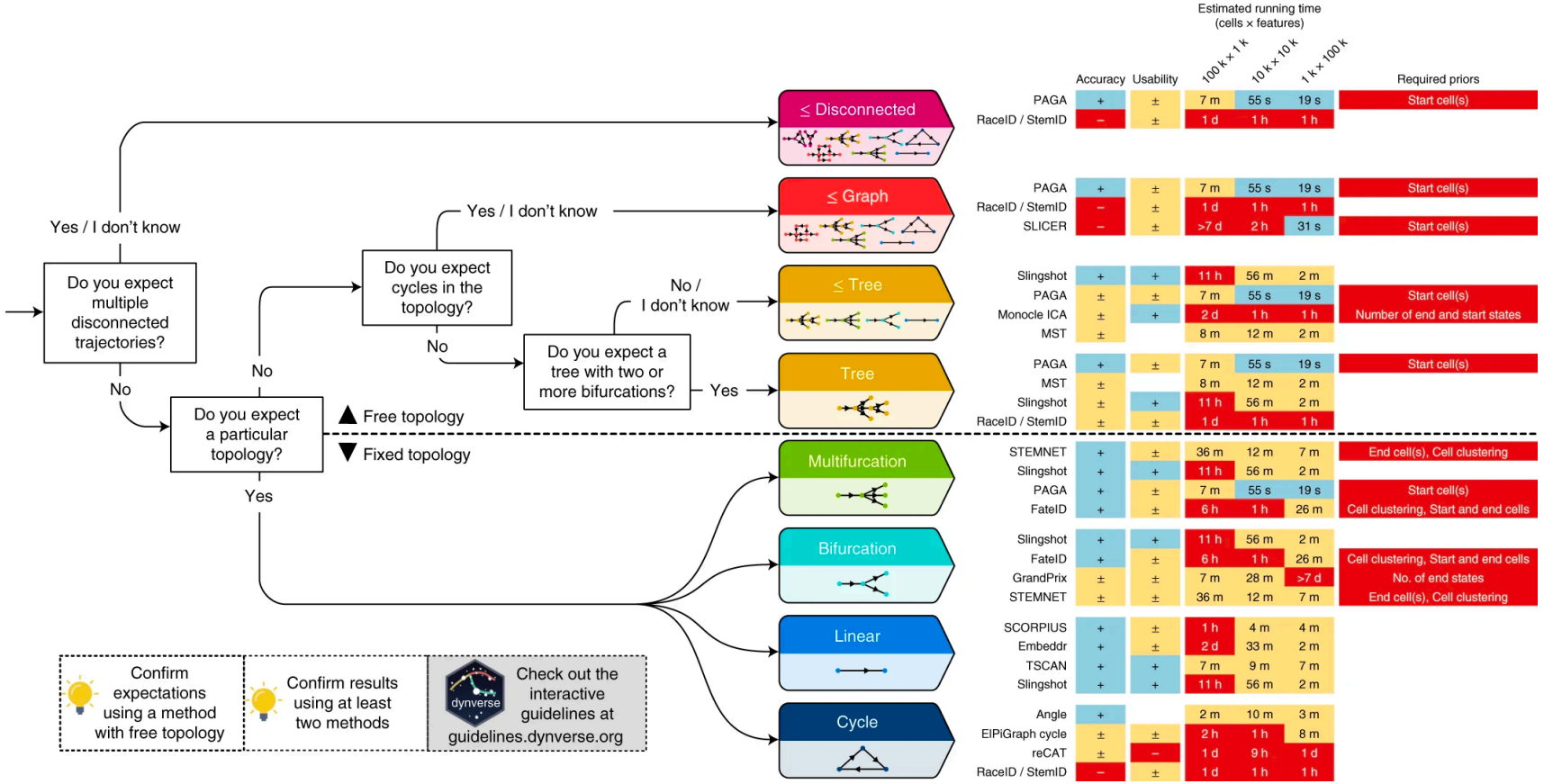
- Trajectory inference
  - Different trajectory inference methods



Saelens et al. (2019) A comparison of single-cell trajectory inference methods

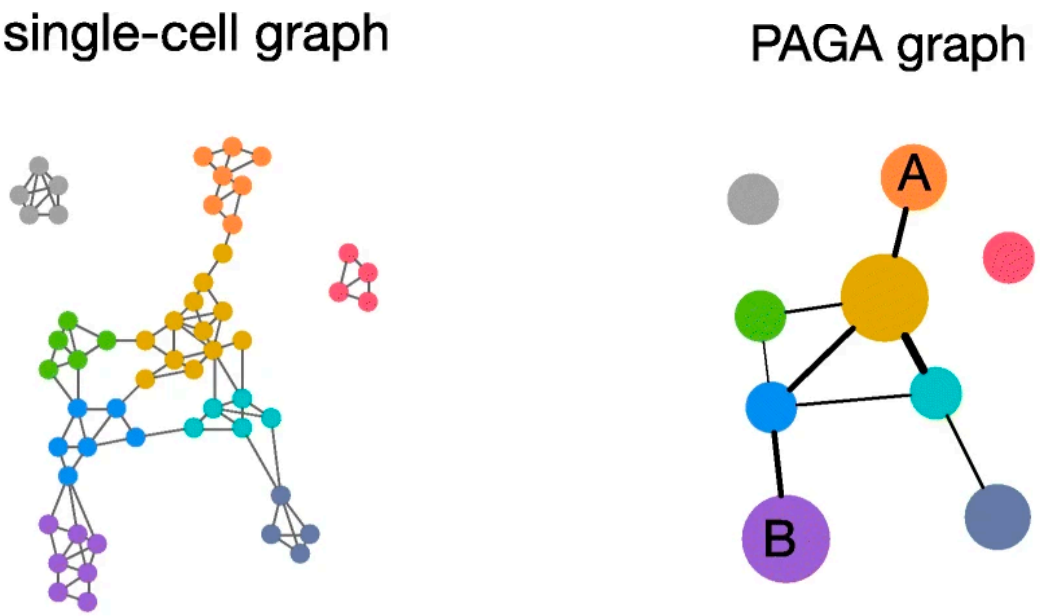
Trajectory inference

- Which methods to use?



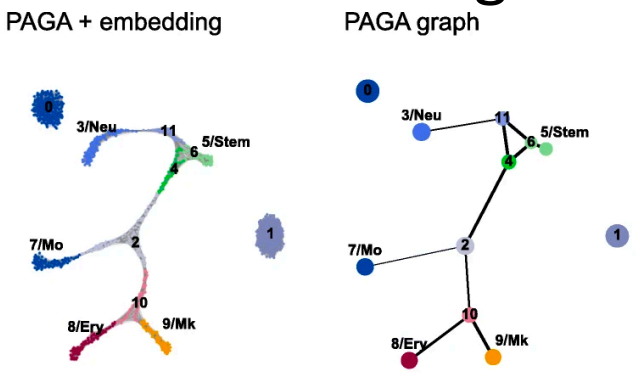
Trajectory inference

- Partition-based graph abstraction (PAGA)
  - nodes/partitions: cell populations
  - edges: distance/similarity



Trajectory inference

- Partition-based graph abstraction (PAGA)
  - initial graph from clustering results
  - use the connectivity between partitions as the weight of edges to connect partitions
  - discard spurious edges with low weights
  - order cells within each partition according to their distance from a root cell using walk-based distance measure



RNA-velocity

- Velocity:
  - a vector quantity that shows how fast and in what direction a point is moving
  - speed: rate
  - velocity: vector (speed+direction)

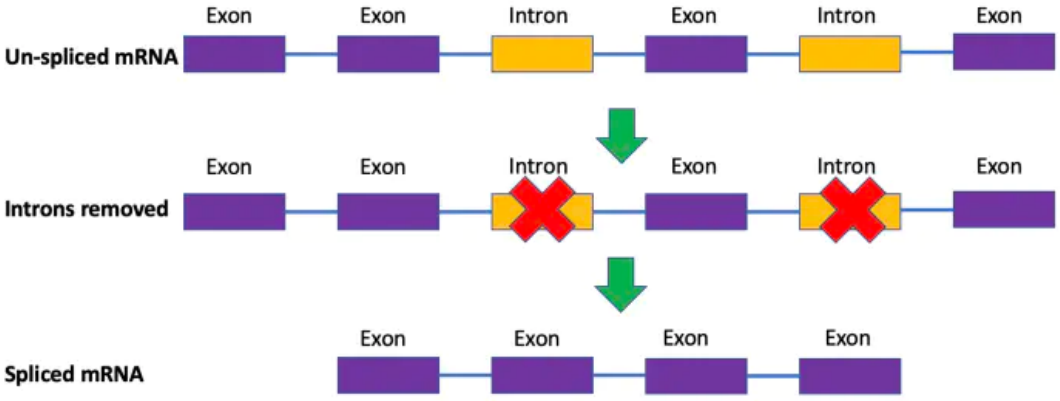
- $v = \frac{ds}{dt}$
  - s: distance moved
  - v: velocity
  - t: time

RNA-velocity

- 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

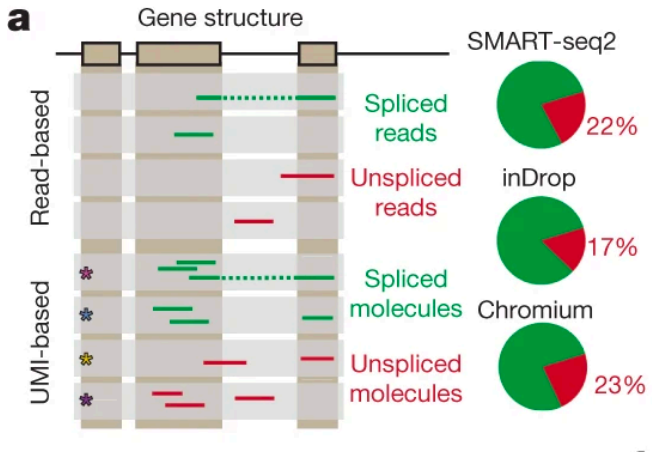
RNA-velocity

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



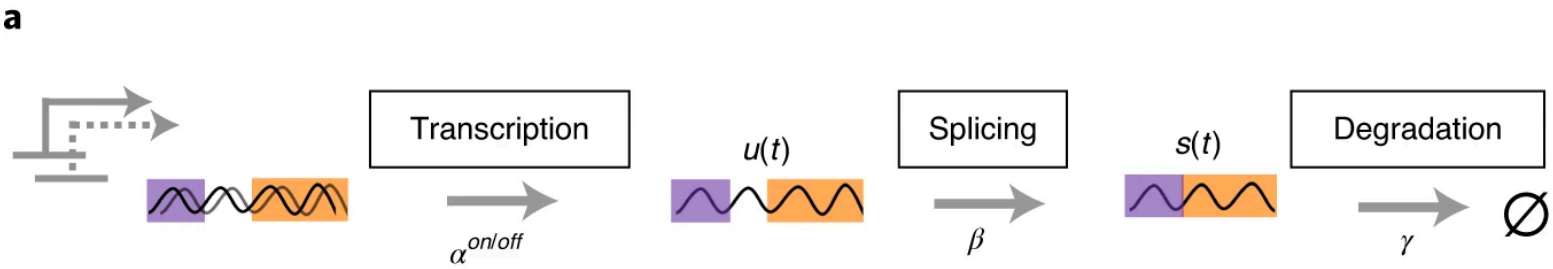
RNA-velocity

- 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 in trajectory inference



  - at time t
  - spliced mRNA reads changes =

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

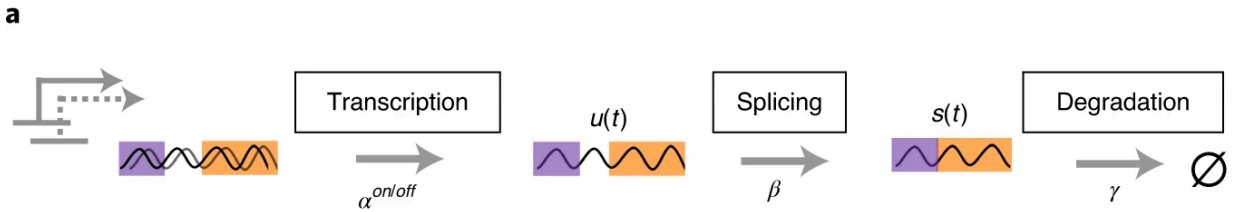
RNA-velocity

- RNA velocity in trajectory inference
  - the time derivative of the gene expression state
  - capturing transcription α
  - splicing rate β
  - degradation rates γ
  - 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



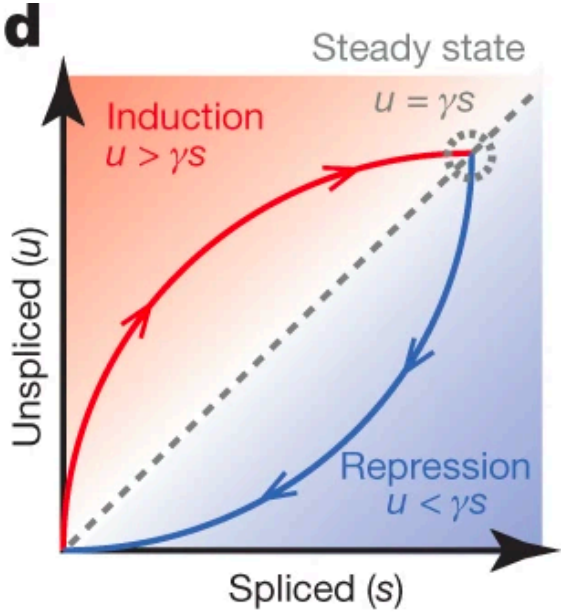
RNA-velocity

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

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

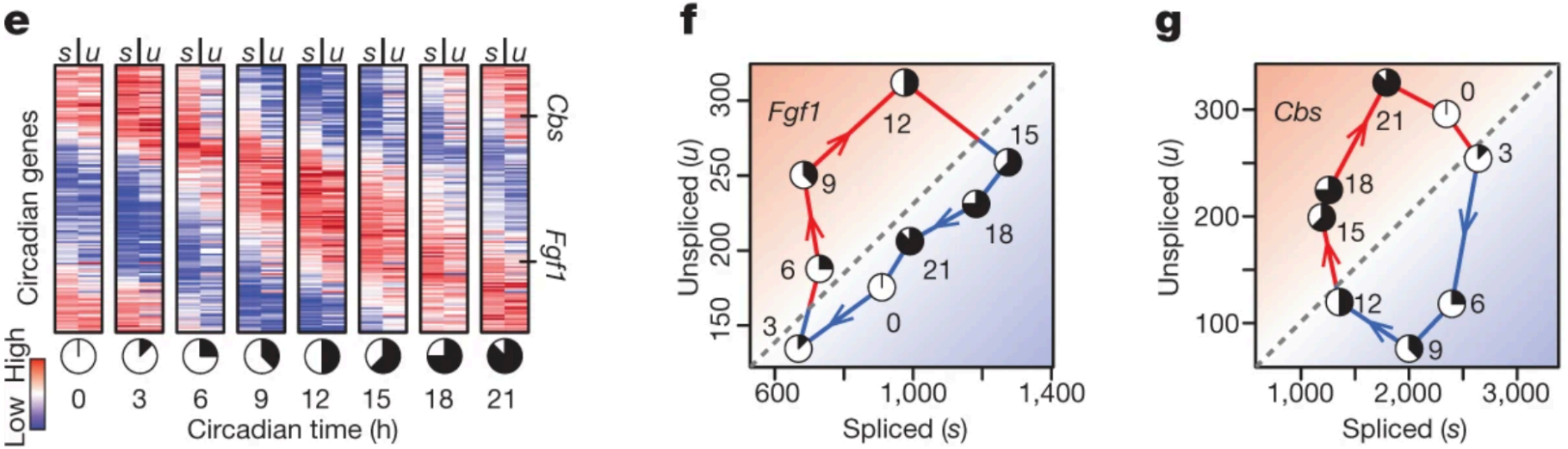
$$\frac{ds}{dt} = u - \gamma s \qquad \text{steady state: } \frac{ds}{dt} = 0$$

$$0 = u - \gamma s \qquad u = \gamma s$$



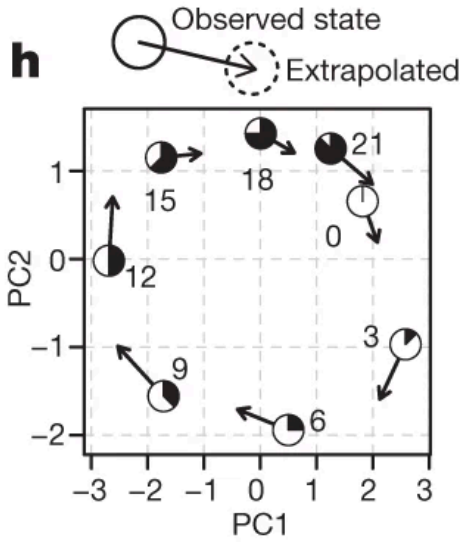
RNA-velocity

- RNA velocity in trajectory inference
  - examples in circadian time



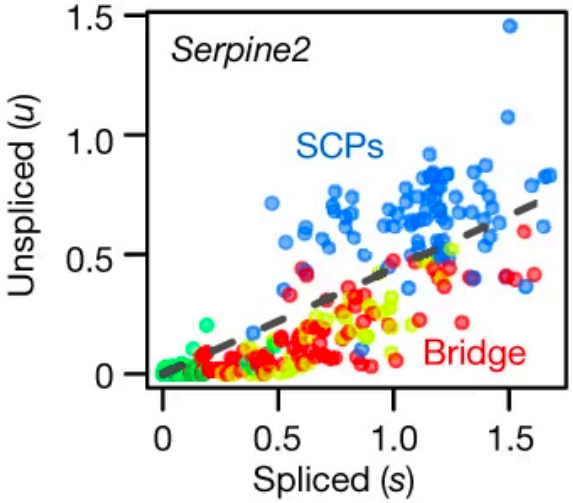
RNA-velocity

- 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



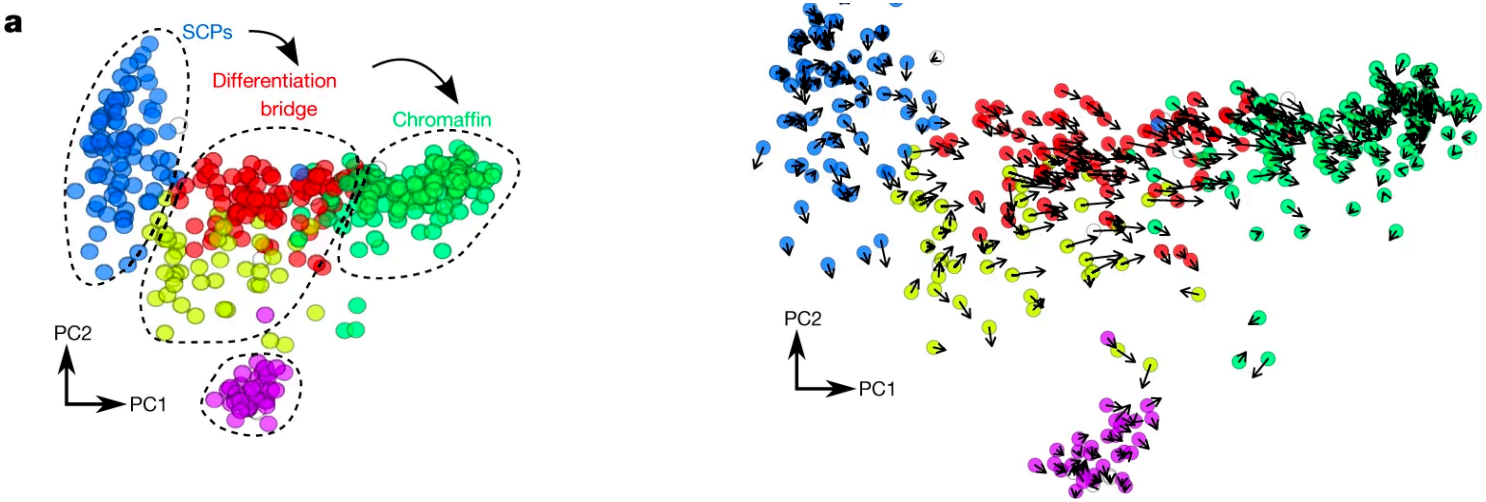
RNA-velocity

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



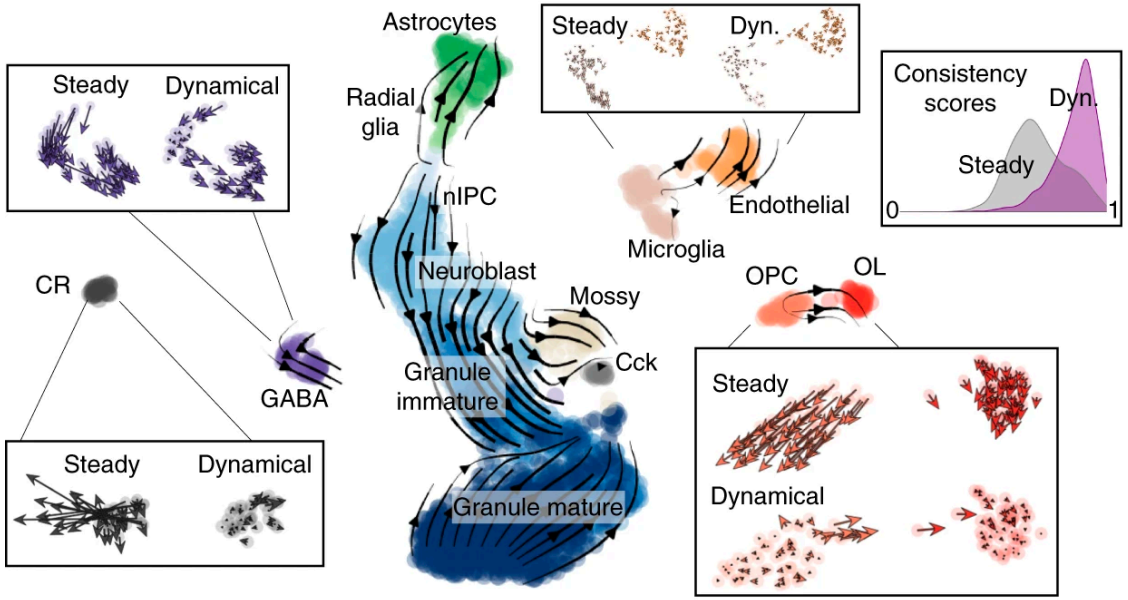
RNA-velocity

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



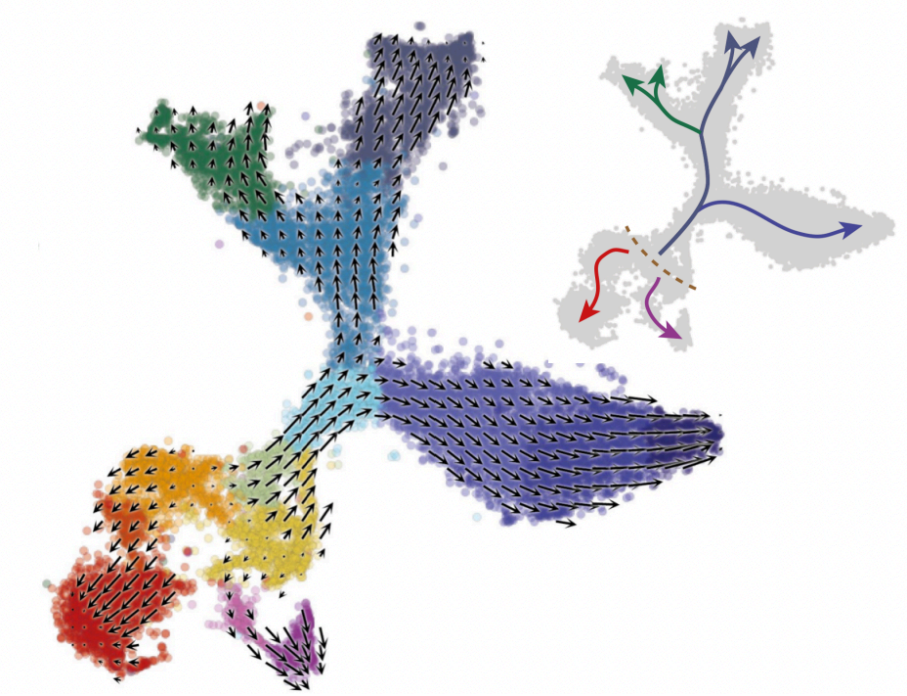
RNA-velocity

- 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