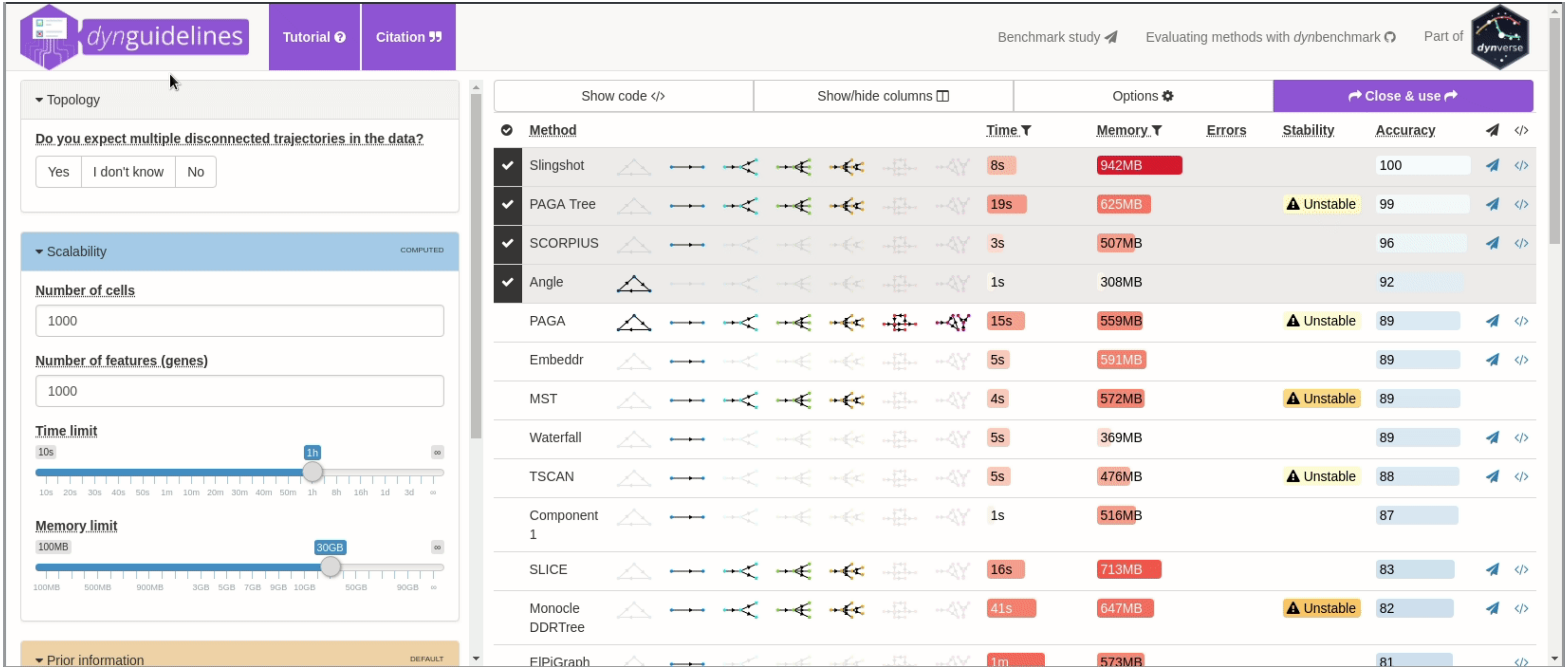
- TRABAJO

	Language	Platform	Input data	Approach
scVIA	Python	scanpy	*h5ad/ann data	trajectory
Monocle	R	seurat	*rds; *h5ad	trajectory
Monocle3	R	seurat	*rds;*h5ad	trajectory
Slingshot	R	SCE	*h5ad; *rds	trajectory
Diffusion map	R	SCE	*h5ad; *rds	trajectory
CAPITAL	Python	scanpy	*h5ad/ann data	alignment/trajectory

Each method is sensitive (exception DPT) to dimensionality reduction utilized to infer the trajectory!

dynverse/dynguidelines



dynguidelines

Tutorial ❓ Citation ❞

Benchmark study Evaluating methods with *dyn*benchmark Part of dynverse

▾ Topology

Do you expect multiple disconnected trajectories in the data?

Yes I don't know No

▾ Scalability COMPUTED

Number of cells

1000

Number of features (genes)

1000

Time limit

10s — 1h — ∞

10s 20s 30s 40s 50s 1m 10m 20m 30m 40m 50m 1h 8h 16h 1d 3d ∞

Memory limit

100MB — 30GB — ∞

100MB 500MB 900MB 3GB 5GB 7GB 9GB 10GB 50GB 90GB ∞

▾ Prior information DEFAULT

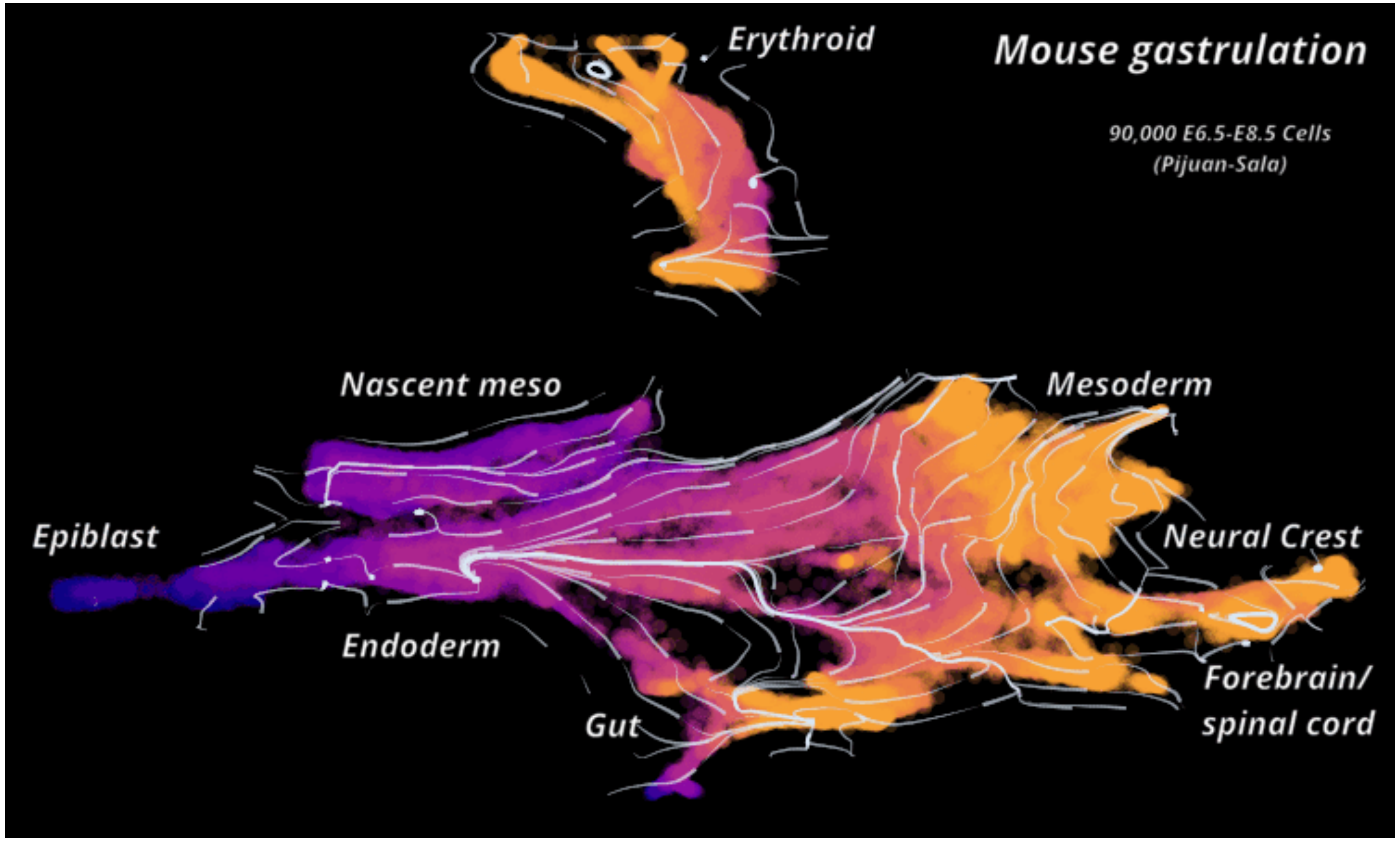
Show code </> Show/hide columns Options ⚙ Close & use

	Method	Time	Memory	Errors	Stability	Accuracy
✓	Slingshot	8s	942MB			100
✓	PAGA Tree	19s	625MB		⚠ Unstable	99
✓	SCORPIUS	3s	507MB			96
✓	Angle	1s	308MB			92
	PAGA	15s	559MB		⚠ Unstable	89
	Embeddr	5s	591MB			89
	MST	4s	572MB		⚠ Unstable	89
	Waterfall	5s	369MB			89
	TSCAN	5s	476MB		⚠ Unstable	88
	Component 1	1s	516MB			87
	SLICE	16s	713MB			83
	Monocle DDRTree	41s	647MB		⚠ Unstable	82
	ElPiGraph	1m	573MB			81

- scVIA

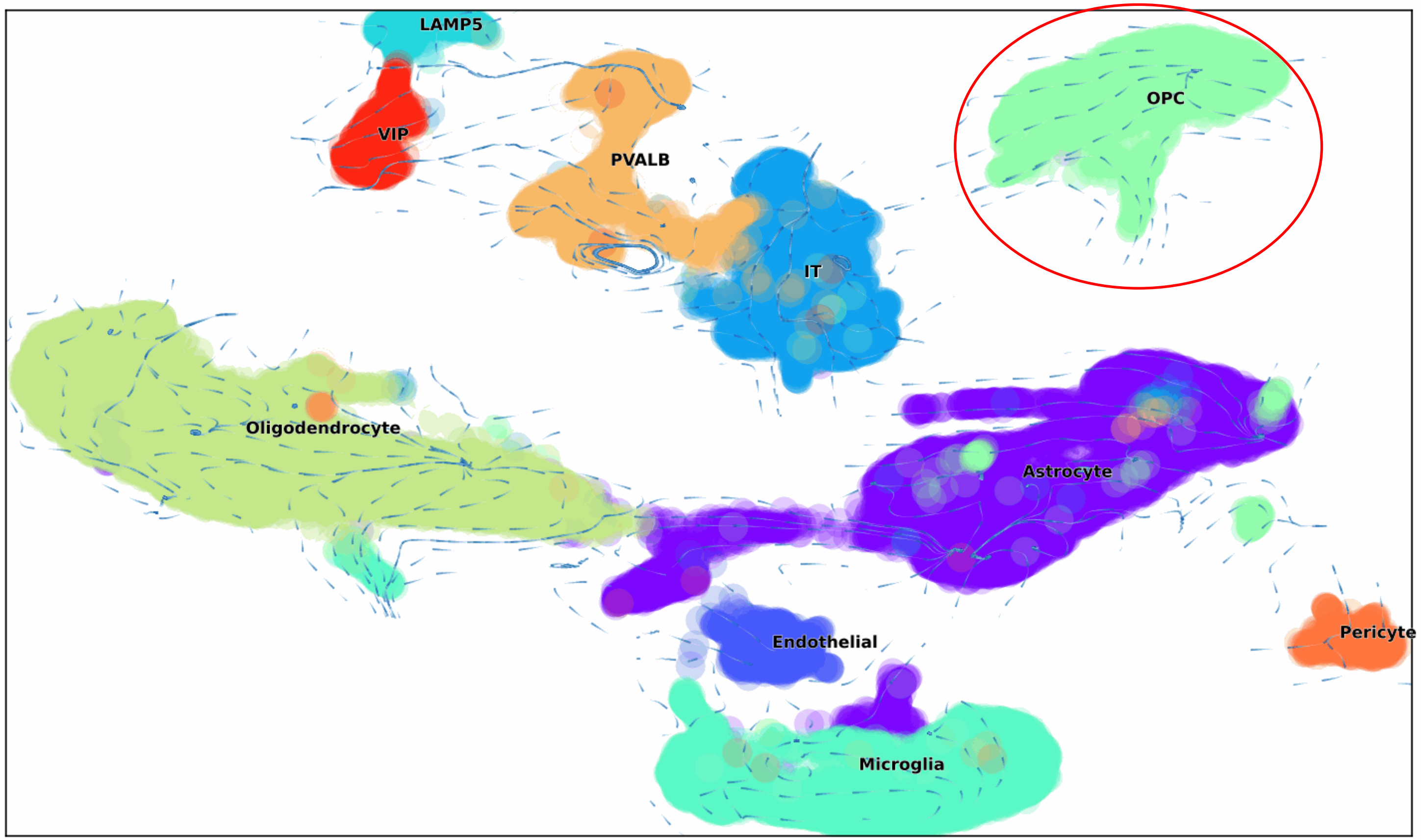
- VIA combines lazy-teleporting random walks and Monte-Carlo Markov Chain simulations to overcome common challenges such as

1) accurate terminal state and lineage inference
2) ability to capture combination of cyclic, disconnected and tree-like structures
3) scalability in feature and sample space
4) generalizability to multi-omic analysis



- scVIA

Potential temporal interactions between different cell types



Scanpy preprocessing
(or ann data with
pre-calculated UMAP)

Graph tree with
genes of potential interest

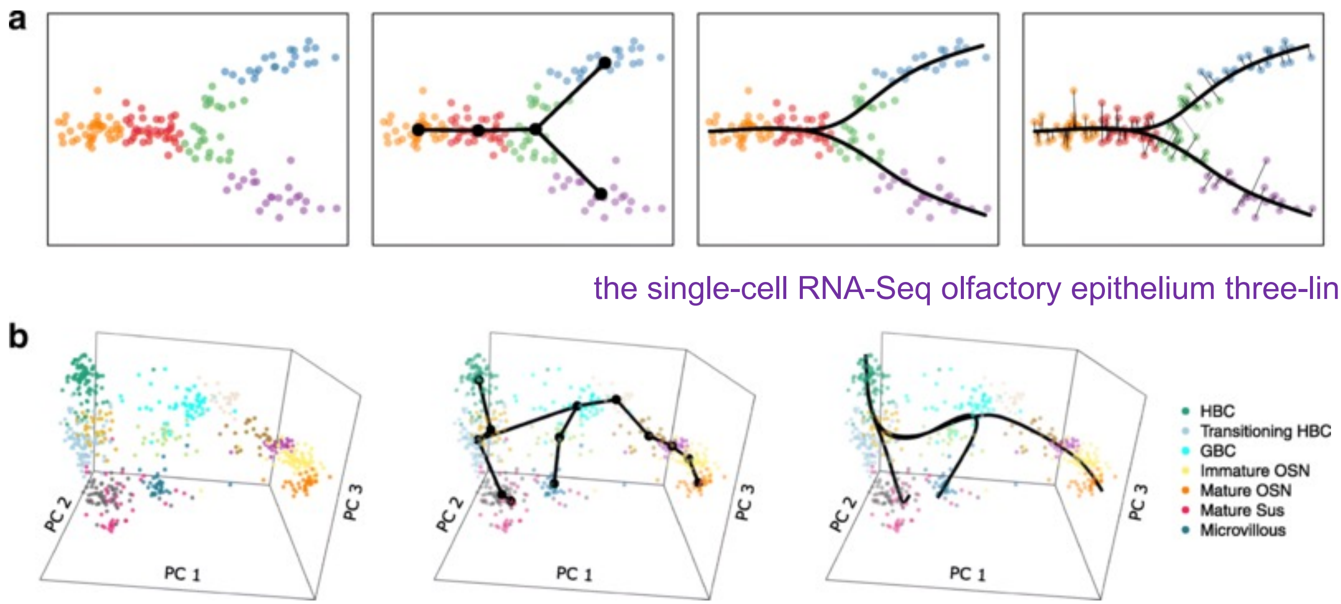
ACC brain-region
(3DG multiome, RNA-seq)

- Slingshot

use clusters of cells to uncover global structure and convert this structure into smooth lineages represented by one-dimensional variables, called “pseudotime”

2D

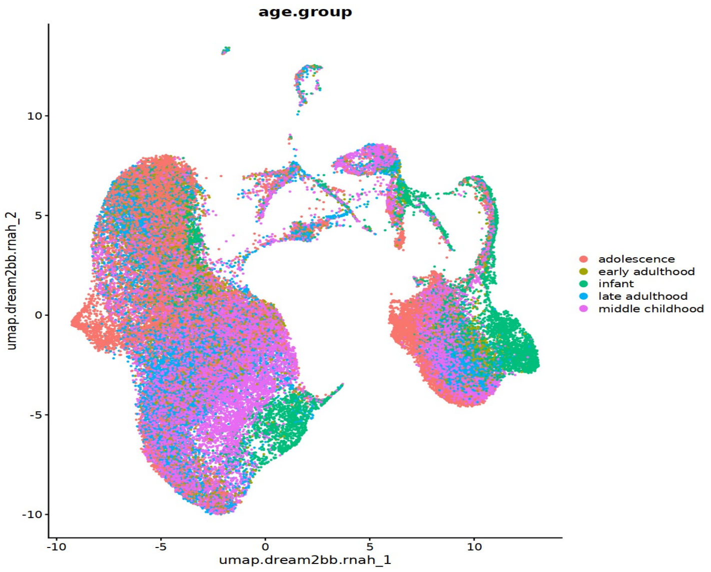
3D

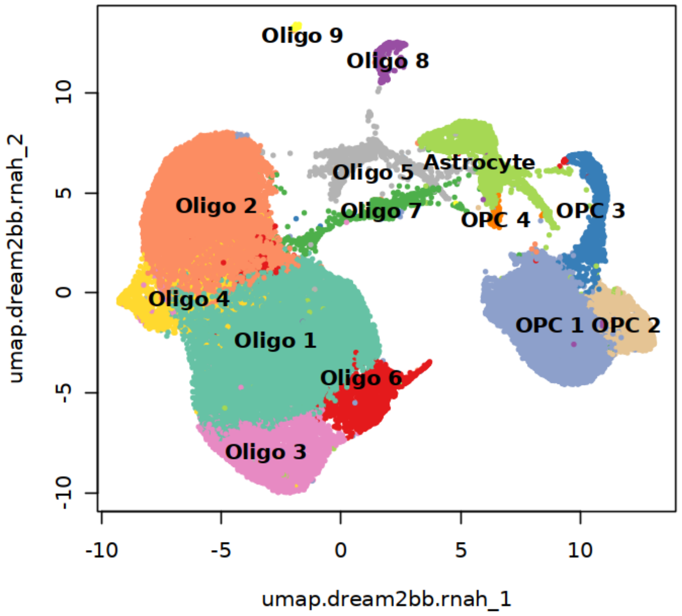


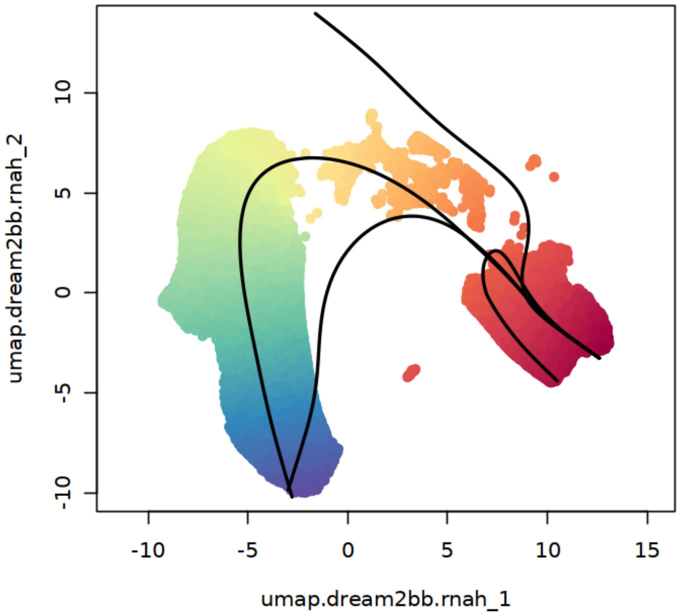
the single-cell RNA-Seq olfactory epithelium three-lineage dataset

Step 1: A minimum spanning tree is constructed on the clusters to determine the number and rough shape of lineages.
Step 2: Simultaneous principal curves are used to obtain smooth representations of each lineage.
Step 3: Pseudotime values are obtained by orthogonal projection onto the curves

- Slingshot


age.group
umap.dream2bb.rnah_2
umap.dream2bb.rnah_1
10
5
0
-5
-10
-10
-5
0
5
10
● adolescence
● early adulthood
● infant
● late adulthood
● middle childhood


Oligo 9
Oligo 8
Oligo 5
Astrocyte
Oligo 2
Oligo 7
OPC 4
OPC 3
Oligo 4
Oligo 1
OPC 1
OPC 2
Oligo 6
Oligo 3
umap.dream2bb.rnah_2
umap.dream2bb.rnah_1
10
5
0
-5
-10
-10
-5
0
5
10


umap.dream2bb.rnah_2
umap.dream2bb.rnah_1
10
5
0
-5
-10
-10
-5
0
5
10
15

Monocle3 vs Monocle/2

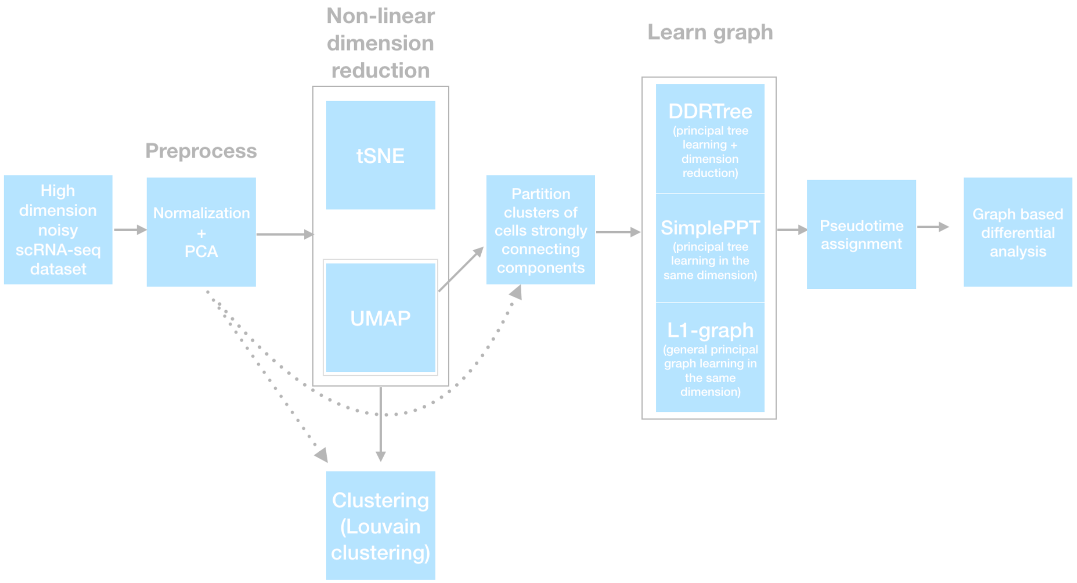
- Clustering, classifying, and counting cells.
- Constructing single-cell trajectories.
- Differential expression analysis.

Monocle3

(beta phase of development)

Improvement

- Support for the UMAP algorithm to initialize trajectory inference.
- Support for trajectories with multiple roots.
- Ways to learn trajectories that have loops or points of convergence.
- Algorithms that automatically partition cells to learn disjoint or parallel trajectories using ideas from "approximate graph abstraction".



High dimension noisy scRNA-seq dataset

Preprocess

Normalization + PCA

Non-linear dimension reduction

tSNE

UMAP

Clustering (Louvain clustering)

Partition clusters of cells strongly connecting components

Learn graph

DDRTree (principal tree learning + dimension reduction)

SimplePPT (principal tree learning in the same dimension)

L1-graph (general principal graph learning in the same dimension)

Pseudotime assignment

Graph based differential analysis

Monocle/2

ordering single cells in **pseudotime**

- placing them along a trajectory corresponding to a biological process
- by taking advantage of individual cell's asynchronous progression of those processes

- orders cells by learning an explicit principal graph from the scRNA data with advanced machine learning techniques (Reversed Graph Embedding)

The Ordering Workflow

Step 1: choosing genes that define progress (dpFeature: Selecting features from dense cell clusters)
Step 2: reducing the dimensionality of the data (Reverse Graph Embedding)
Step 3: ordering the cells in pseudotime

- assumes that the trajectory has a tree structure, with one end of it the "root", and the others the "leaves”
- Monocle's job is to fit the best tree it can to the data >> manifold learning

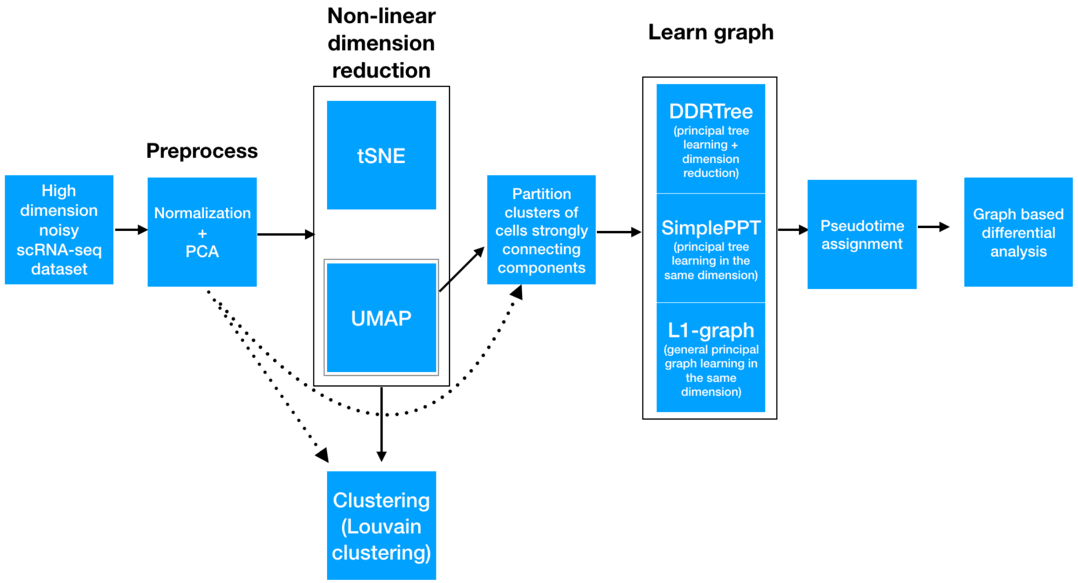
Monocle3 vs Monocle

(beta phase of development)

- Clustering, classifying, and counting cells.
- Constructing single-cell trajectories.
- Differential expression analysis.

Improvement

•Support for the UMAP algorithm to initialize trajectory inference.
•Support for trajectories with multiple roots.
•Ways to learn trajectories that have loops or points of convergence.
•Algorithms that automatically partition cells to learn disjoint or parallel trajectories using ideas from "approximate graph abstraction".



Non-linear dimension reduction
Learn graph
Preprocess
High dimension noisy scRNA-seq dataset
Normalization + PCA
tSNE
UMAP
Partition clusters of cells strongly connecting components
DDRTree (principal tree learning + dimension reduction)
SimplePPT (principal tree learning in the same dimension)
L1-graph (general principal graph learning in the same dimension)
Pseudotime assignment
Graph based differential analysis
Clustering (Louvain clustering)

ordering single cells in **pseudotime**

- placing them along a trajectory corresponding to a biological process
- by taking advantage of individual cell's asynchronous progression of those processes

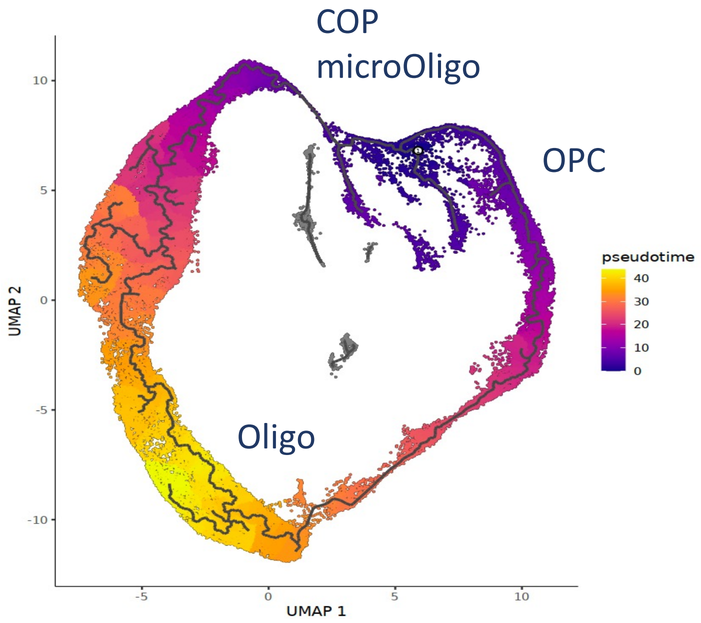
- orders cells by learning an explicit principal graph from the scRNA data with advanced machine learning techniques (Reversed Graph Embedding)

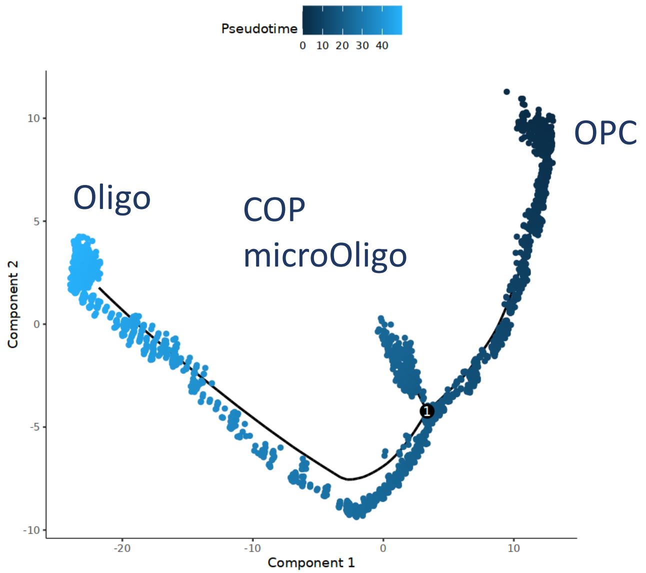
The Ordering Workflow

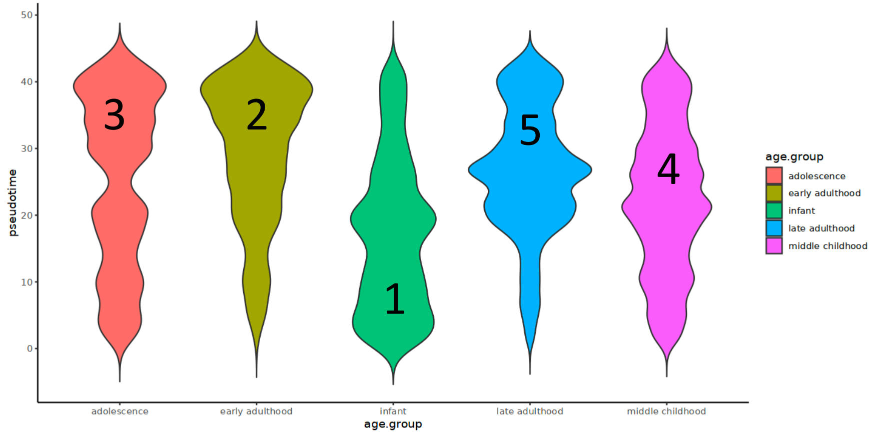
Step 1: choosing genes that define progress (dpFeature: Selecting features from dense cell clusters)
Step 2: reducing the dimensionality of the data (Reverse Graph Embedding)
Step 3: ordering the cells in pseudotime

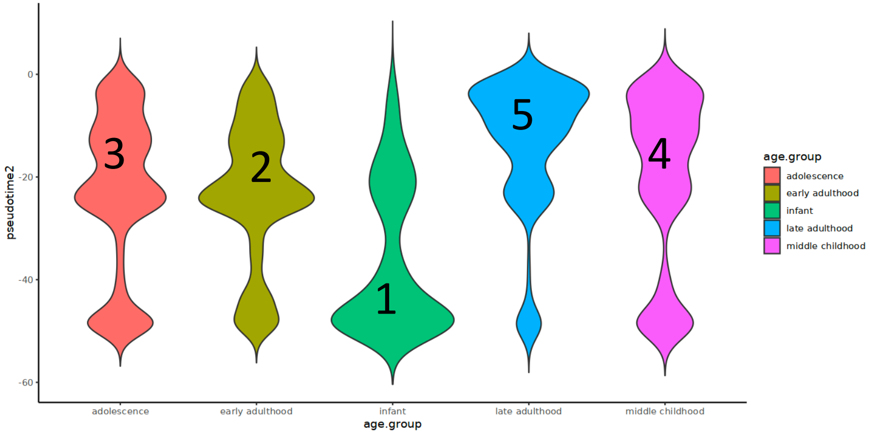
- assumes that the trajectory has a tree structure, with one end of it the "root", and the others the "leaves”
- Monocle's job is to fit the best tree it can to the data >> manifold learning

Monocle3 vs Monocle

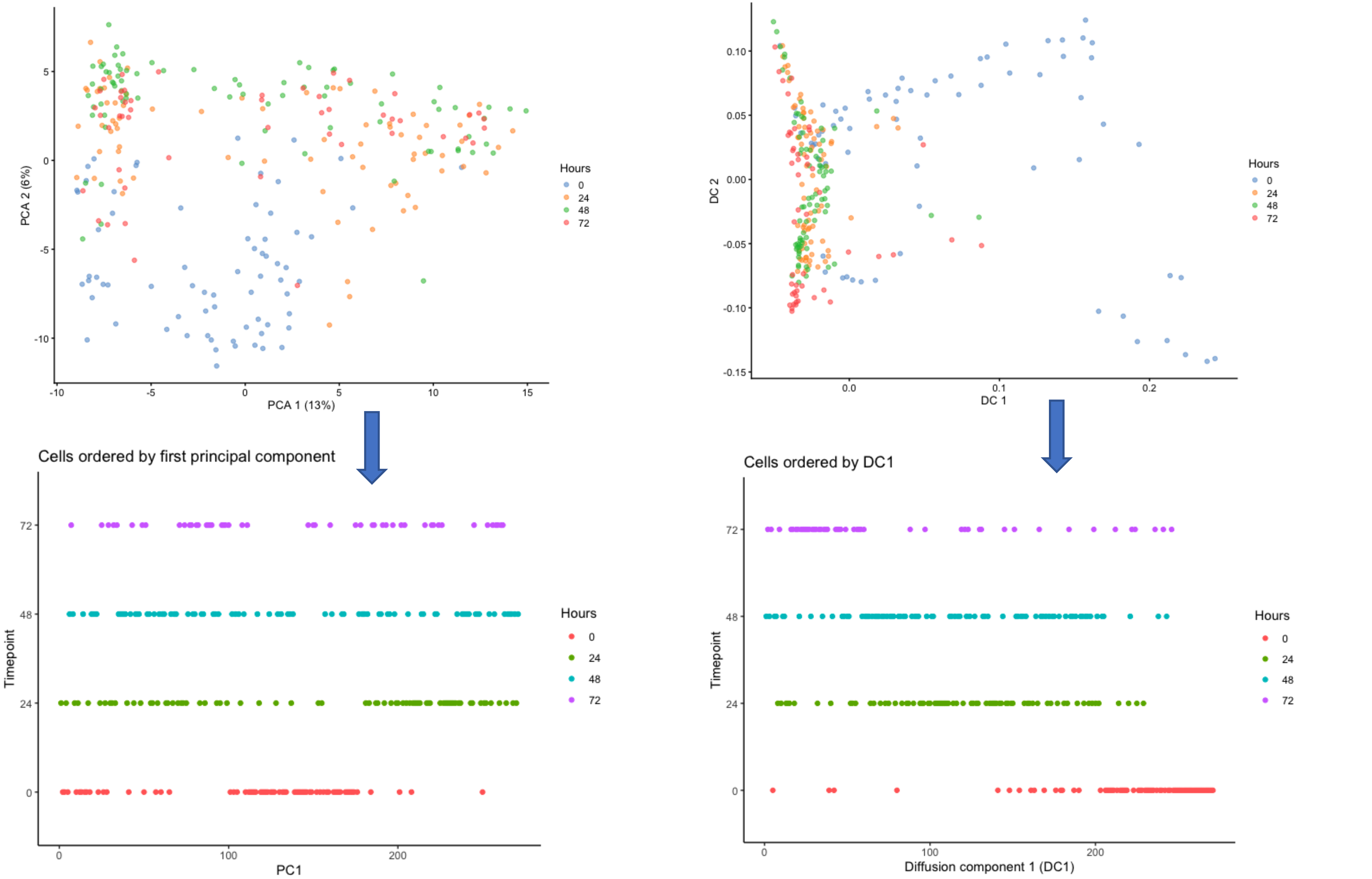








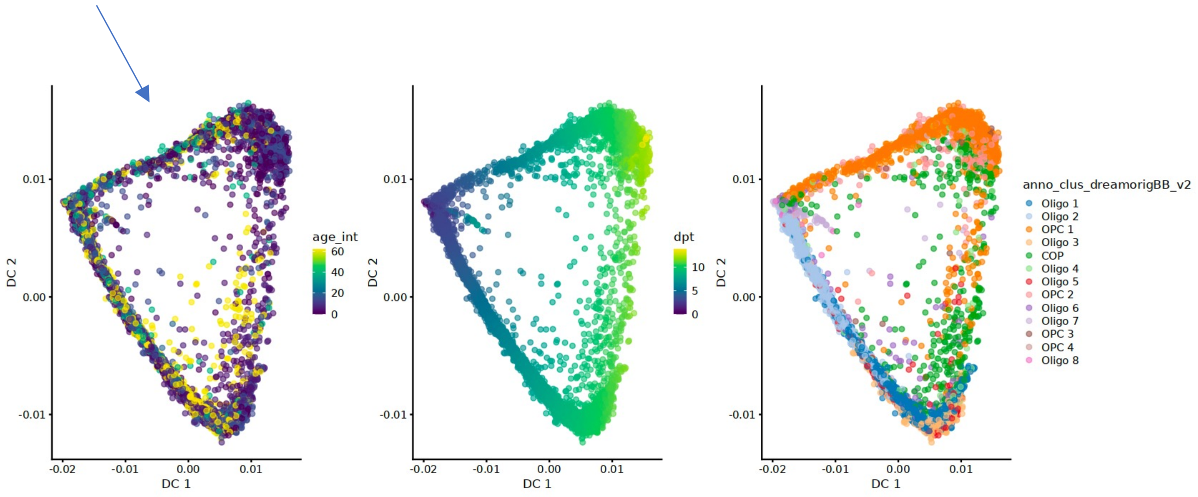
Diffusion map… how it should work

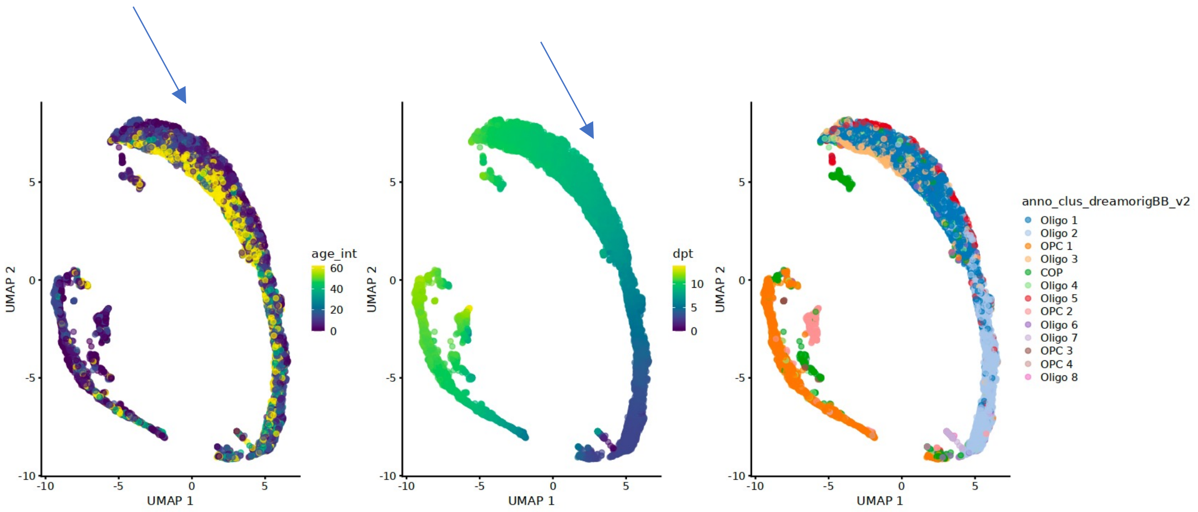


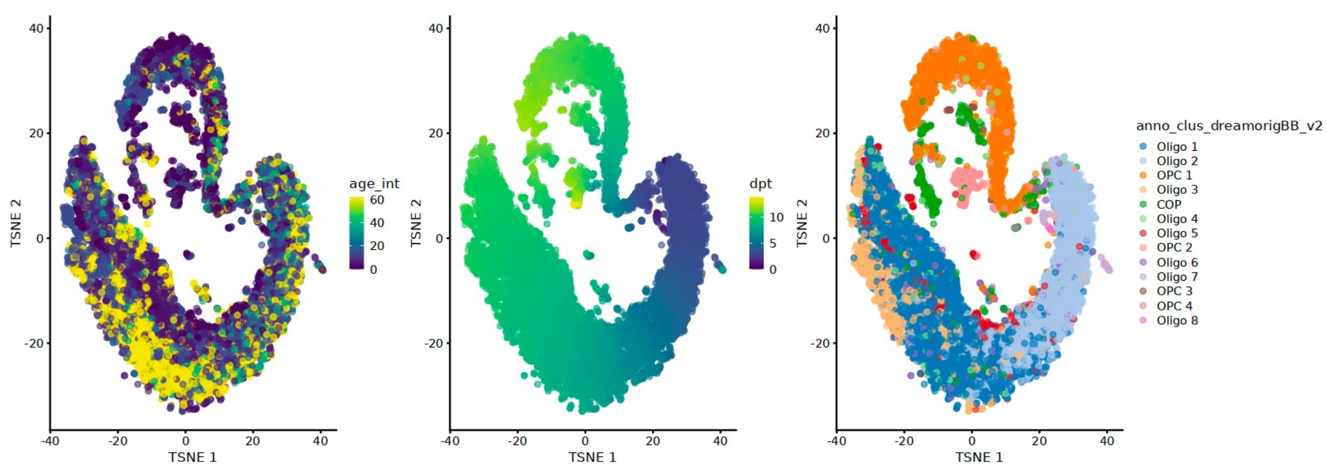
DC map

- the underlying idea is **to assume that the data are samples from a diffusion process**

Diffusion map… how it doesn't work

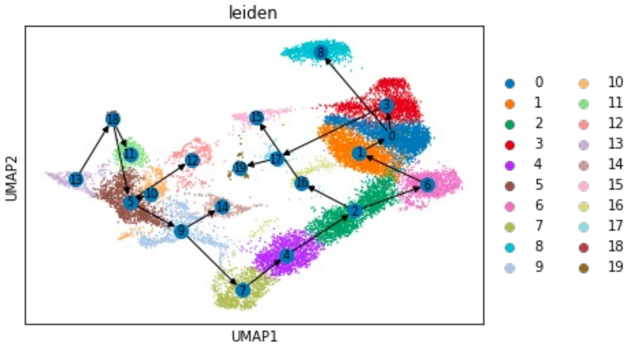


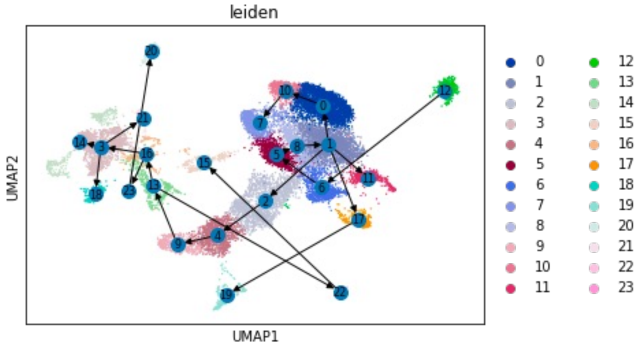


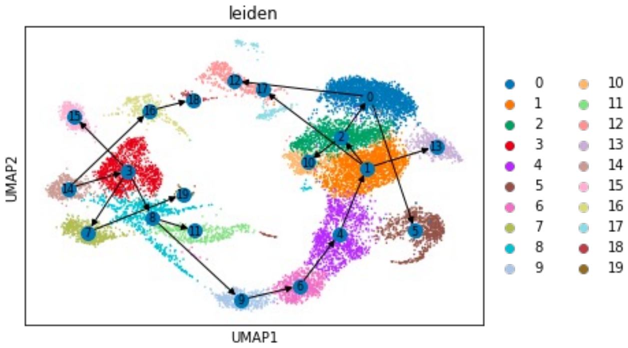


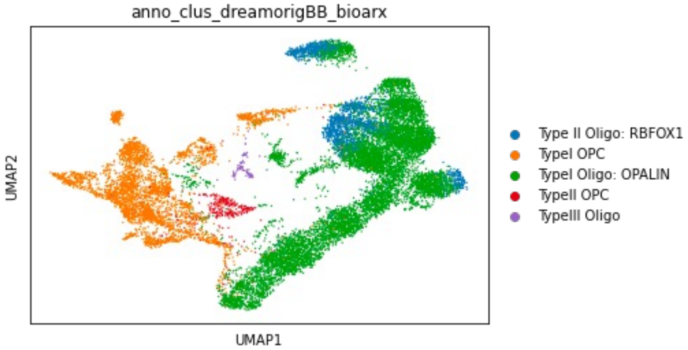
….and UMAP or tSNE is a better way to go

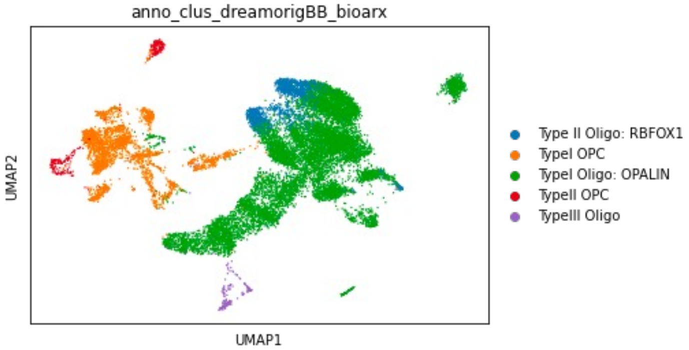
CAPITAL

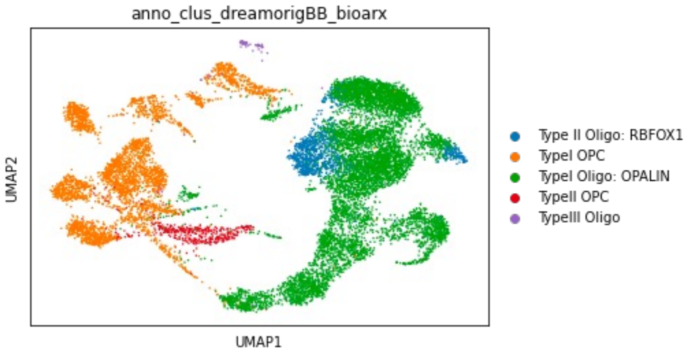

leiden
UMAP2
UMAP1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

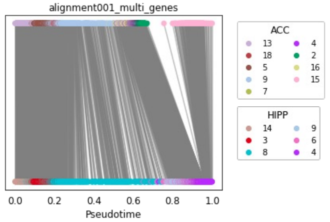

leiden
UMAP2
UMAP1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

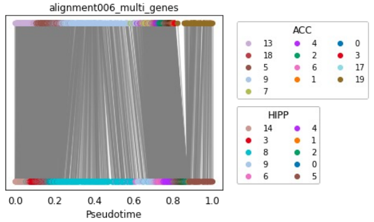

leiden
UMAP2
UMAP1
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

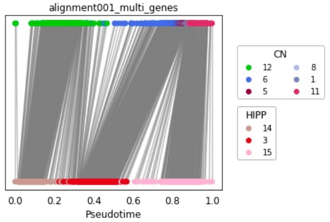

anno_clus_dreamorigBB_bioarx
UMAP2
UMAP1
Type II Oligo: RBFOX1
TypeI OPC
TypeI Oligo: OPALIN
TypeII OPC
TypeIII Oligo

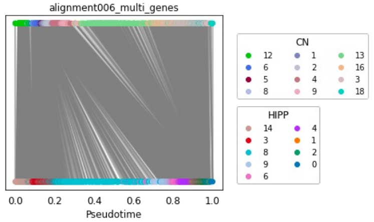

anno_clus_dreamorigBB_bioarx
UMAP2
UMAP1
Type II Oligo: RBFOX1
TypeI OPC
TypeI Oligo: OPALIN
TypeII OPC
TypeIII Oligo

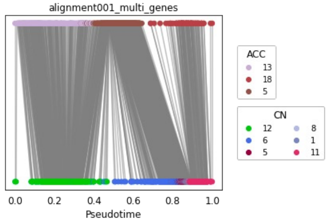

anno_clus_dreamorigBB_bioarx
UMAP2
UMAP1
Type II Oligo: RBFOX1
TypeI OPC
TypeI Oligo: OPALIN
TypeII OPC
TypeIII Oligo

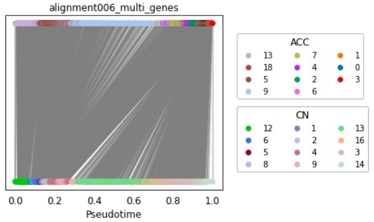

alignment001_multi_genes
0.0
0.2
0.4
0.6
0.8
1.0
Pseudotime
ACC
13
18
5
9
7
4
2
16
15
HIPP
14
3
8
9
6
4


alignment006_multi_genes
0.0
0.2
0.4
0.6
0.8
1.0
Pseudotime
ACC
13
18
5
9
7
4
2
6
1
0
3
17
19
HIPP
14
3
8
9
6
4
1
2
0
5


alignment001_multi_genes
0.0
0.2
0.4
0.6
0.8
1.0
Pseudotime
CN
12
6
5
8
1
11
HIPP
14
3
15


alignment006_multi_genes
0.0
0.2
0.4
0.6
0.8
1.0
Pseudotime
CN
12
6
5
8
1
2
4
9
13
16
3
18
HIPP
14
3
8
9
6
4
1
2
0


alignment001_multi_genes
0.0
0.2
0.4
0.6
0.8
1.0
Pseudotime
ACC
13
18
5
CN
12
6
5
8
1
11


alignment006_multi_genes
0.0
0.2
0.4
0.6
0.8
1.0
Pseudotime
ACC
13
18
5
9
7
4
2
6
1
0
3
CN
12
6
5
8
1
2
4
9
13
16
3
14