Advanced Topics in Single Cell Omics

RNA Velocity

Group 2

Grzegorz Maciag
Michael Teske
Adhideb Ghosh
Daryl Boey

Volker Bergen

Paulo Czarnewski

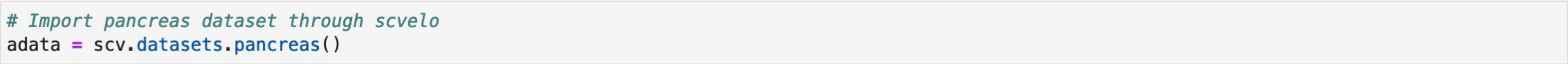
Objectives

1. Identify driver genes using RNA velocity based on:
    a. Genes contributing to vector fields in embedding
    b. Dynamic gene modelling
    c. Transiently expressed genes

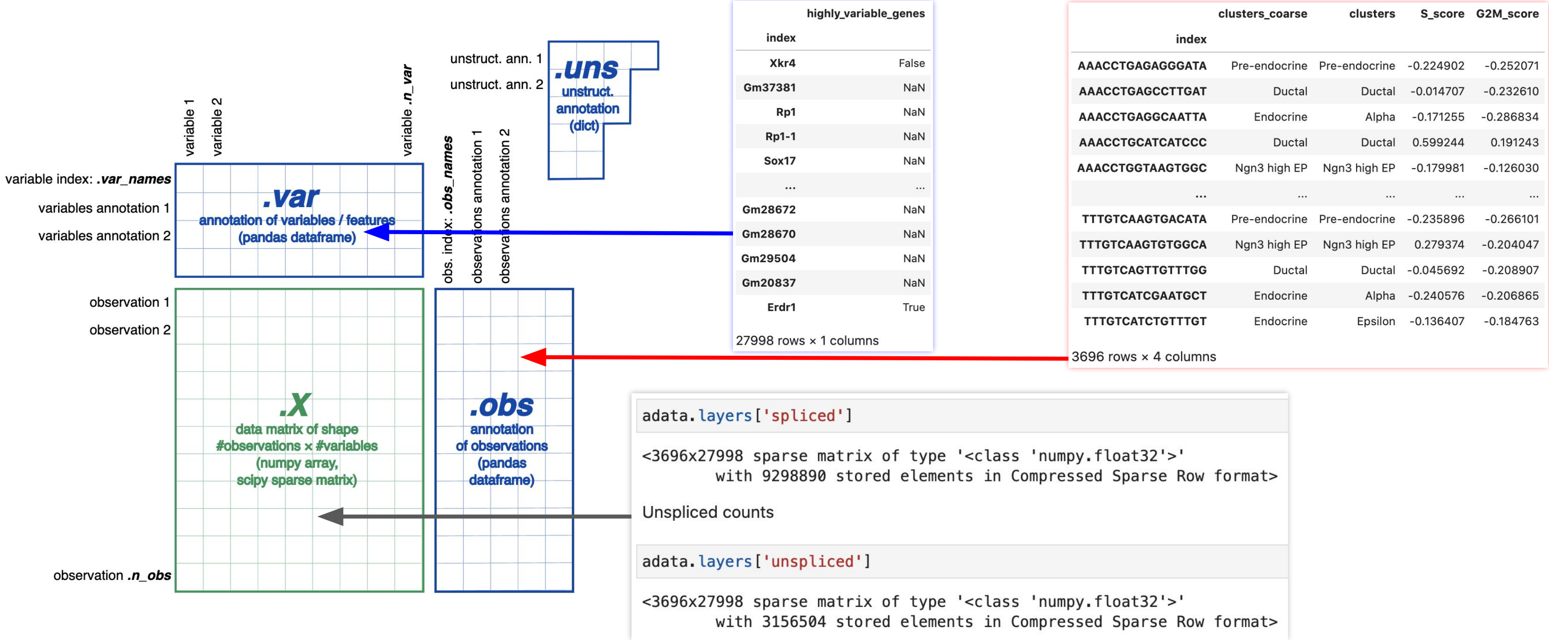
2. Based on the above, determine biologically relevant genes in differentiation

AnnData is the Launchpad

AnnData is a **popular format** for storing sc data used by scanpy and scVelo. It allows for comprehensive and scalable storage of **data** matrix **and annotation** information features and samples on different **layers**.



```
# Import pancreas dataset through scvelo
adata = scv.datasets.pancreas()
```



variable 1
variable 2
variable .n_var
variable index: .var_names
variables annotation 1
variables annotation 2
.var
annotation of variables / features
(pandas dataframe)
obs. index: .obs_names
observations annotation 1
observations annotation 2
unstruct. ann. 1
unstruct. ann. 2
.uns
unstruct. annotation (dict)
observation 1
observation 2
observation .n_obs
.X
data matrix of shape #observations × #variables
(numpy array, scipy sparse matrix)
.obs
annotation of observations
(pandas dataframe)

	highly_variable_genes
index	
Xkr4	False
Gm37381	NaN
Rp1	NaN
Rp1-1	NaN
Sox17	NaN
...	...
Gm28672	NaN
Gm28670	NaN
Gm29504	NaN
Gm20837	NaN
Erdr1	True

27998 rows × 1 columns

	clusters_coarse	clusters	S_score	G2M_score
index				
AAACCTGAGAGGGATA	Pre-endocrine	Pre-endocrine	-0.224902	-0.252071
AAACCTGAGCCTTGAT	Ductal	Ductal	-0.014707	-0.232610
AAACCTGAGGCAATTA	Endocrine	Alpha	-0.171255	-0.286834
AAACCTGCATCATCCC	Ductal	Ductal	0.599244	0.191243
AAACCTGGTAAGTGGC	Ngn3 high EP	Ngn3 high EP	-0.179981	-0.126030
...	...	...	...	...
TTTGTCAAGTGACATA	Pre-endocrine	Pre-endocrine	-0.235896	-0.266101
TTTGTCAAGTGTGGCA	Ngn3 high EP	Ngn3 high EP	0.279374	-0.204047
TTTGTCAGTTGTTTGG	Ductal	Ductal	-0.045692	-0.208907
TTTGTCATCGAATGCT	Endocrine	Alpha	-0.240576	-0.206865
TTTGTCATCTGTTTGT	Endocrine	Epsilon	-0.136407	-0.184763

3696 rows × 4 columns

```
adata.layers['spliced']
```

```
<3696x27998 sparse matrix of type '<class 'numpy.float32'>'
	with 9298890 stored elements in Compressed Sparse Row format>
```

Unspliced counts

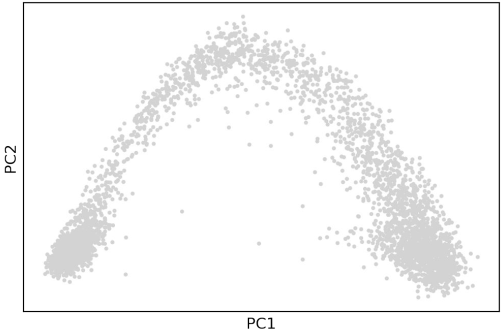
```
adata.layers['unspliced']
```

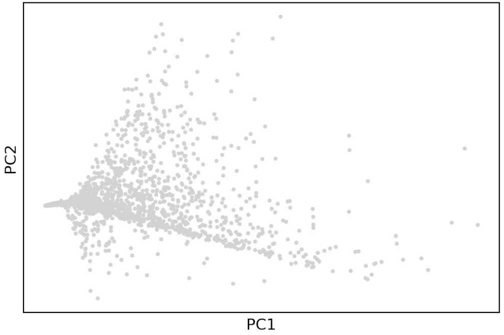
```
<3696x27998 sparse matrix of type '<class 'numpy.float32'>'
	with 3156504 stored elements in Compressed Sparse Row format>
```

Data Pre-processing

- Gene filtering:
  - Quality control
  - Eliminate covariates like dropouts, low/high gene counts in cells, high mitochondrial reads
  - Eliminate genes expressed only in small number of cells
- Variable gene selection:
  - Feature selection
- Normalisation:
  - Allows for cells to be intra-comparable
- Log transformation
  - Canonical way to measure gene expression
  - Mitigates mean-variance relationship
  - Reduces data skewness

Log-transformed



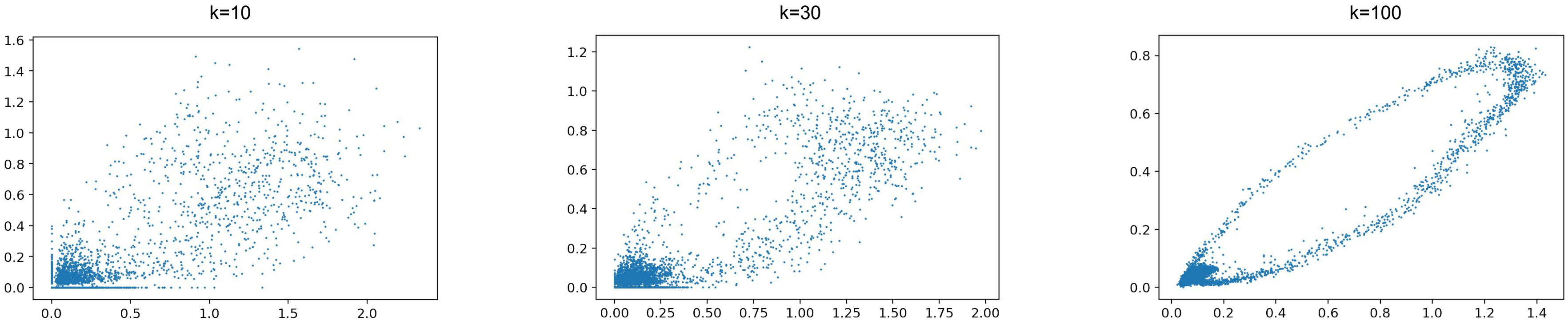


Non log-transformed

Data Imputation

- **kNN** graph represents distance and connectivity between cells, where each cell is connected to it’s k neighbors
- The kNN graph is used for computing the mean (first-order moments) and variance (second-order moments) of its k neighboring cells (**kNN imputation**)
- Number of neighbors, k impacts the imputation
  - Lower k results into noisy blob without any meaningful biological information
  - Higher k completely smoothes out the variance generating artificial results
  - Default value of k=30 seems to work fine, as it can already capture the induction and repression phase

Sulf2



k=10

k=30

k=100

Choice of velocity model matters

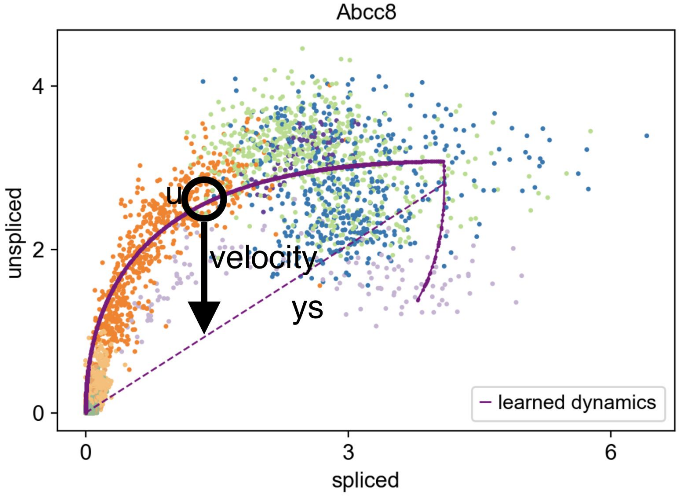
$$\frac{du}{dt} = \alpha(t) - \beta(t)\,u(t)$$

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

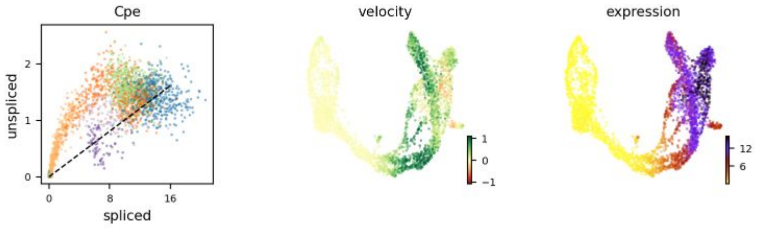
$$\xrightarrow[\beta(t)\,=\,1]{\gamma(t)\,=\,\gamma,}$$

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

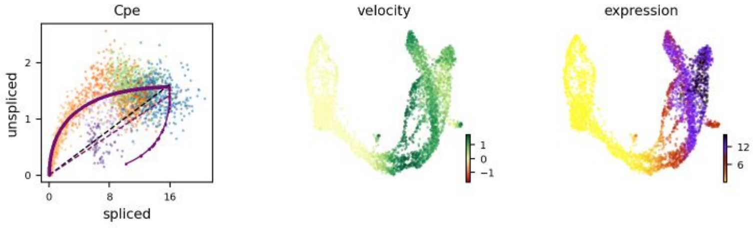
$$\frac{ds}{dt} = u(t) - \gamma s(t)$$



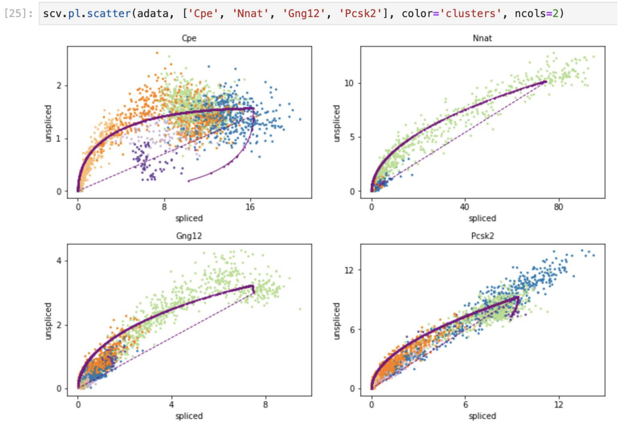
Stochastic



Dynamic

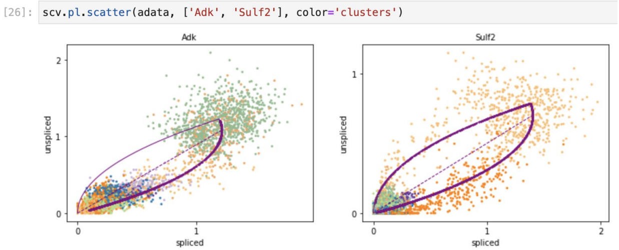


How to interpret velocity phase portraits?



```
[25]: scv.pl.scatter(adata, ['Cpe', 'Nnat', 'Gng12', 'Pcsk2'], color='clusters', ncols=2)
```

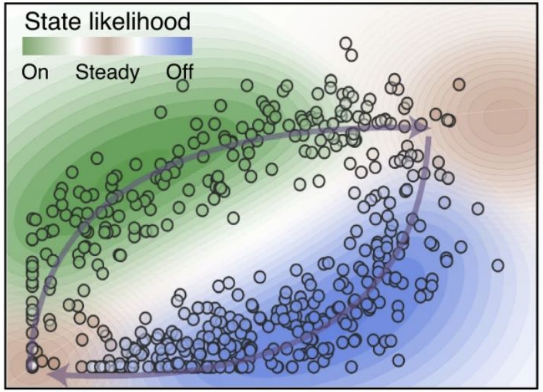
UP

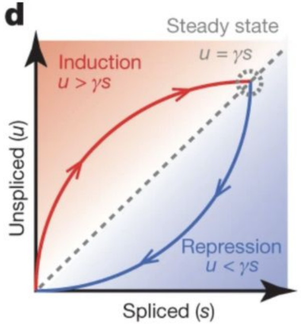


```
[26]: scv.pl.scatter(adata, ['Adk', 'Sulf2'], color='clusters')
```

DOWN

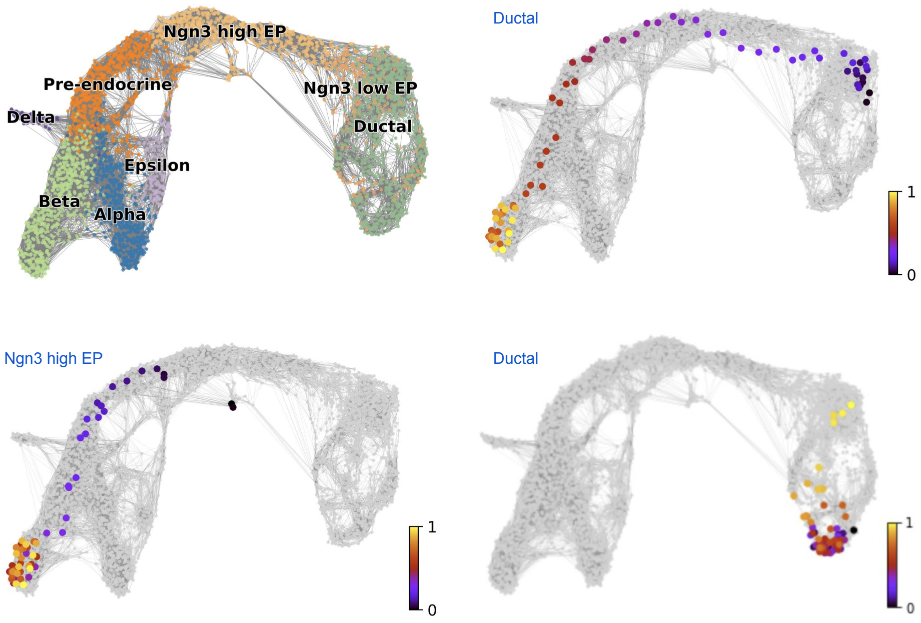
State assignment k_i





Discover the Velocity Graph

The velocity graph is a graph of cell-to-cell transitions inferred from velocity. For two cells, i and j, it represents cosine similarities between velocity vector vi and gene expression change $xj-xi$

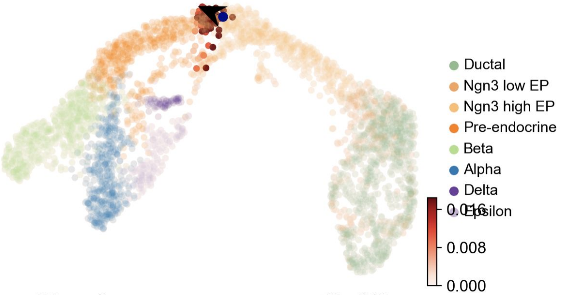


```
scv.tl.velocity_graph(adata)
```

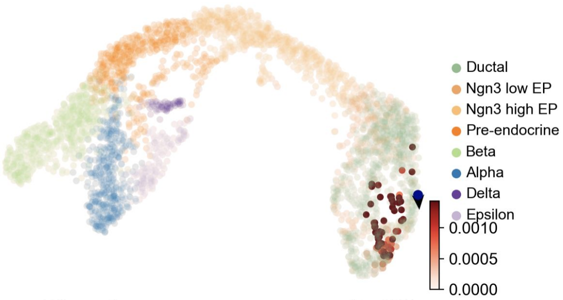
- At basal developmental stages cells can display more locally confined trajectories without clear transitions into other cell types/clusters, indicating cell **cycle-related velocity**.

- Cells from more developmentally advanced clusters will usually exhibit a **more clear trajectory** towards more mature/terminally differentiated cell types.

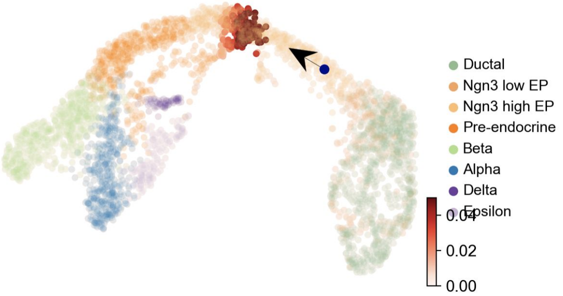
Velocity graph can be used to measure stochasticity



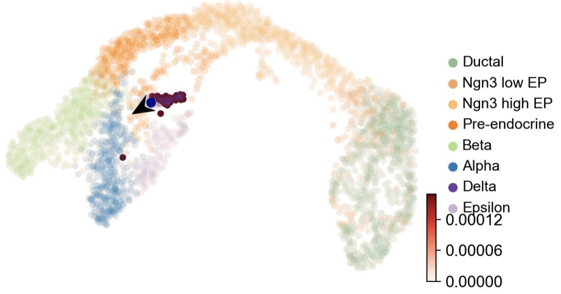
Variance score: 0.43



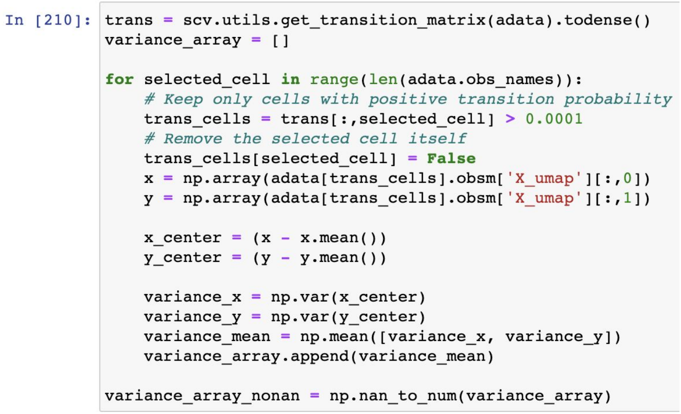
Variance score: 0.86



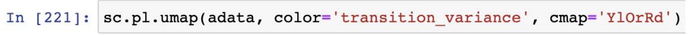
Variance score: 0.6



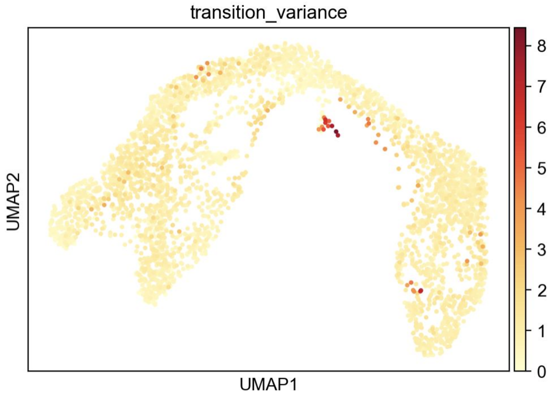
Variance score: 0.28



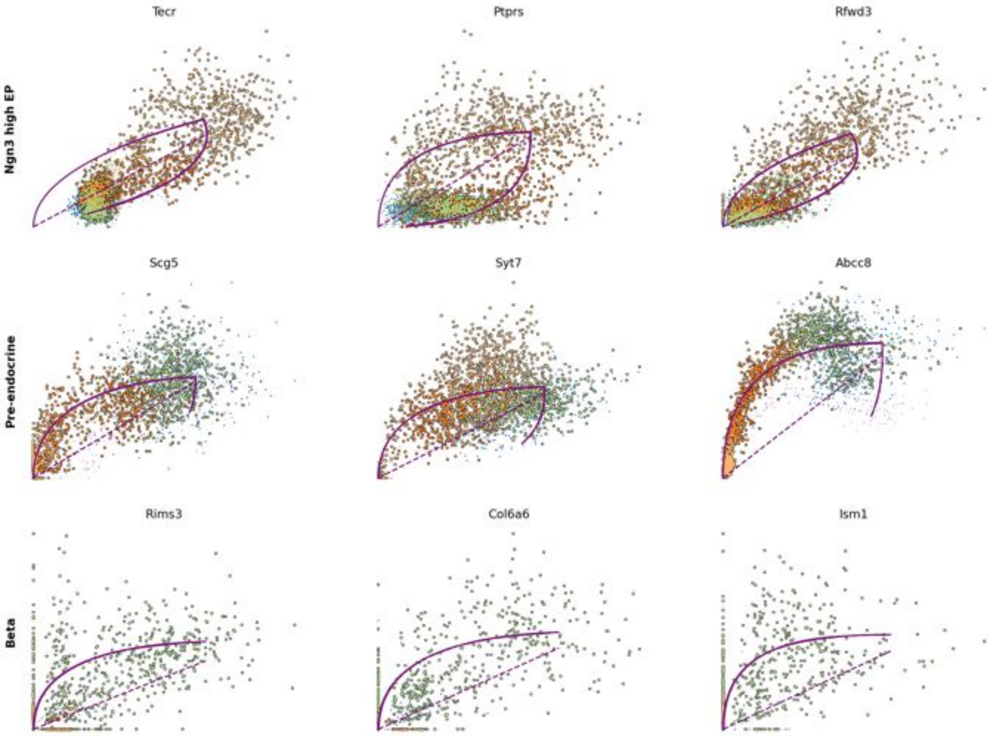
```
In [210]: trans = scv.utils.get_transition_matrix(adata).todense()
          variance_array = []

          for selected_cell in range(len(adata.obs_names)):
              # Keep only cells with positive transition probability
              trans_cells = trans[:,selected_cell] > 0.0001
              # Remove the selected cell itself
              trans_cells[selected_cell] = False
              x = np.array(adata[trans_cells].obsm['X_umap'][:,0])
              y = np.array(adata[trans_cells].obsm['X_umap'][:,1])

              x_center = (x - x.mean())
              y_center = (y - y.mean())

              variance_x = np.var(x_center)
              variance_y = np.var(y_center)
              variance_mean = np.mean([variance_x, variance_y])
              variance_array.append(variance_mean)

          variance_array_nonan = np.nan_to_num(variance_array)
```



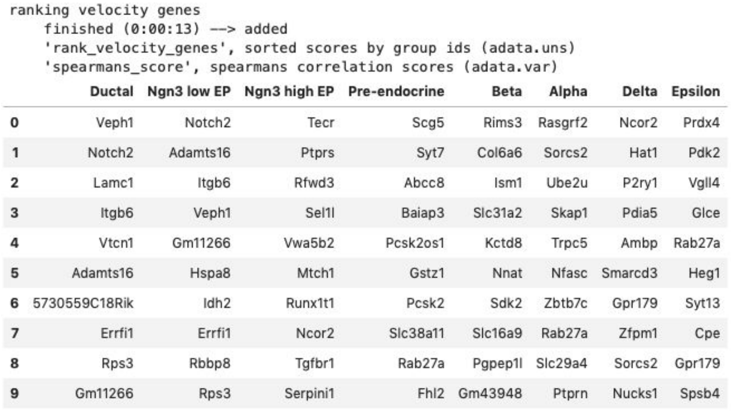
```
In [221]: sc.pl.umap(adata, color='transition_variance', cmap='YlOrRd')
```



Identify putative driver genes with Velocity



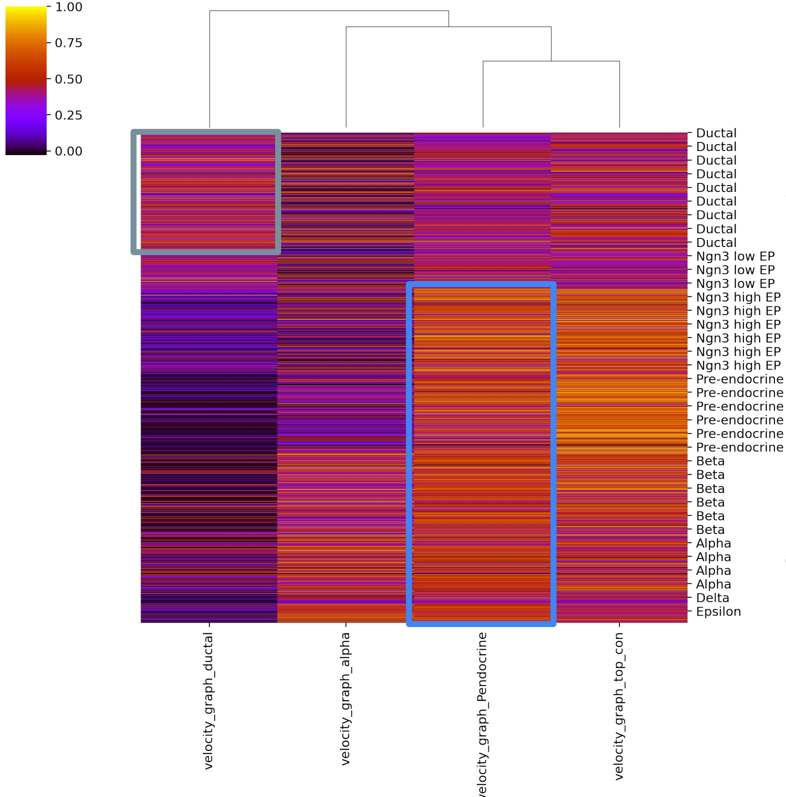
Genes contributing to velocities of cell types

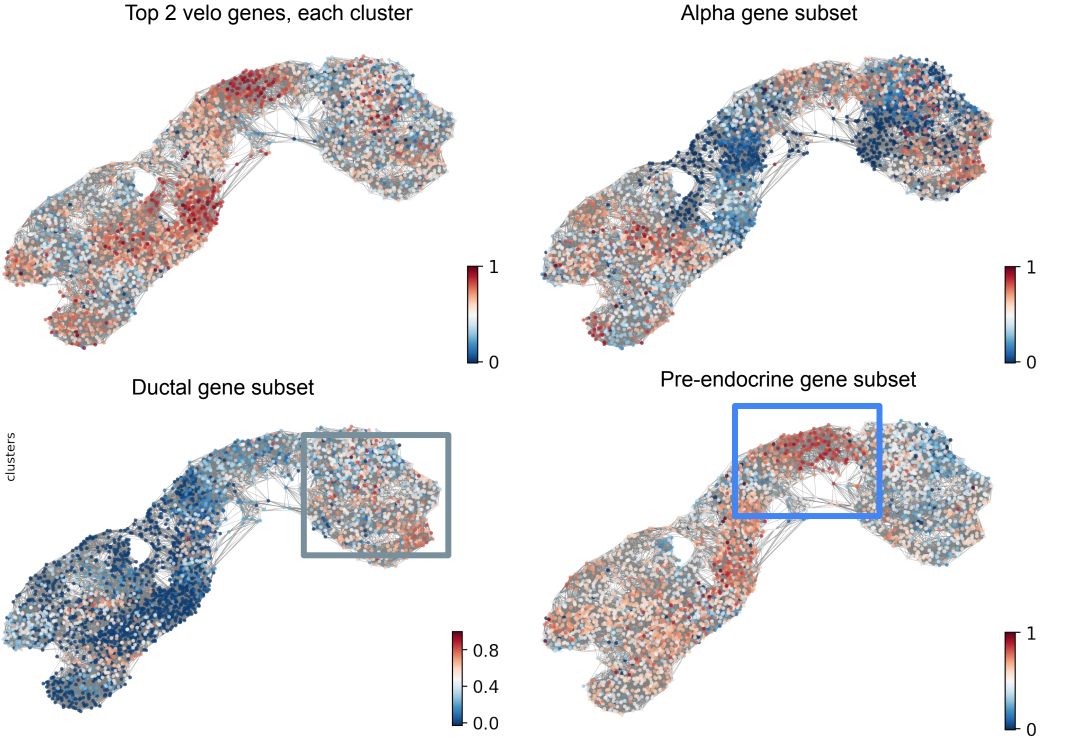


```
ranking velocity genes
    finished (0:00:13) --> added
    'rank_velocity_genes', sorted scores by group ids (adata.uns)
    'spearmans_score', spearmans correlation scores (adata.var)
```

	Ductal	Ngn3 low EP	Ngn3 high EP	Pre-endocrine	Beta	Alpha	Delta	Epsilon
0	Veph1	Notch2	Tecr	Scg5	Rims3	Rasgrf2	Ncor2	Prdx4
1	Notch2	Adamts16	Ptprs	Syt7	Col6a6	Sorcs2	Hat1	Pdk2
2	Lamc1	Itgb6	Rfwd3	Abcc8	Ism1	Ube2u	P2ry1	Vgll4
3	Itgb6	Veph1	Sel1l	Baiap3	Slc31a2	Skap1	Pdia5	Glce
4	Vtcn1	Gm11266	Vwa5b2	Pcsk2os1	Kctd8	Trpc5	Ambp	Rab27a
5	Adamts16	Hspa8	Mtch1	Gstz1	Nnat	Nfasc	Smarcd3	Heg1
6	5730559C18Rik	Idh2	Runx1t1	Pcsk2	Sdk2	Zbtb7c	Gpr179	Syt13
7	Errfi1	Errfi1	Ncor2	Slc38a11	Slc16a9	Rab27a	Zfpm1	Cpe
8	Rps3	Rbbp8	Tgfbr1	Rab27a	Pgpep1l	Slc29a4	Sorcs2	Gpr179
9	Gm11266	Rps3	Serpini1	Fhl2	Gm43948	Ptprn	Nucks1	Spsb4

Correlation of transition probabilities based on driver gene subsets



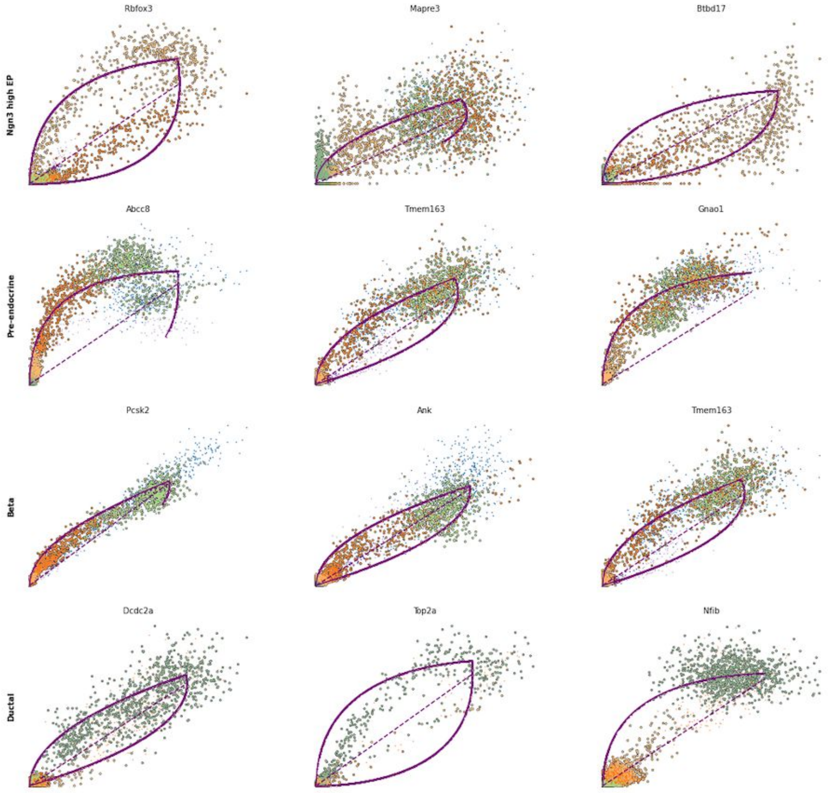

Top 2 velo genes, each cluster

Alpha gene subset

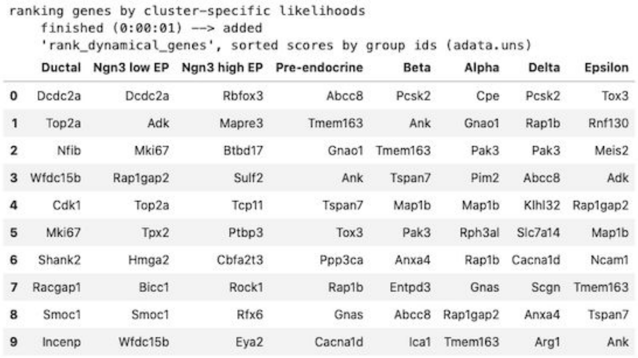
Ductal gene subset

Pre-endocrine gene subset

Identify putative driver genes with dynamic behavior



Dynamically activating genes in the differentiation process based on cluster-specific likelihood

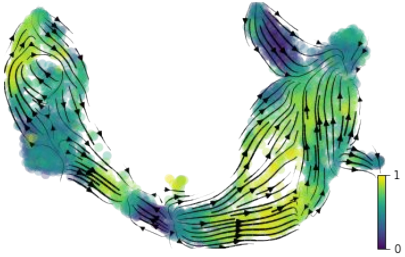


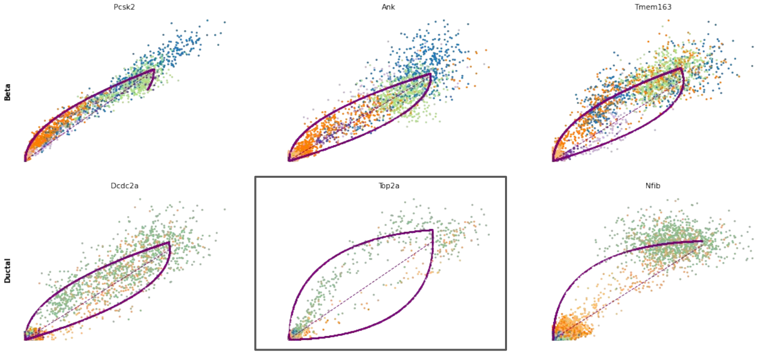
```
ranking genes by cluster-specific likelihoods
    finished (0:00:01) --> added
    'rank_dynamical_genes', sorted scores by group ids (adata.uns)
```

	Ductal	Ngn3 low EP	Ngn3 high EP	Pre-endocrine	Beta	Alpha	Delta	Epsilon
0	Dcdc2a	Dcdc2a	Rbfox3	Abcc8	Pcsk2	Cpe	Pcsk2	Tox3
1	Top2a	Adk	Mapre3	Tmem163	Ank	Gnao1	Rap1b	Rnf130
2	Nfib	Mki67	Btbd17	Gnao1	Tmem163	Pak3	Pak3	Meis2
3	Wfdc15b	Rap1gap2	Sulf2	Ank	Tspan7	Pim2	Abcc8	Adk
4	Cdk1	Top2a	Tcp11	Tspan7	Map1b	Map1b	Klhl32	Rap1gap2
5	Mki67	Tpx2	Ptbp3	Tox3	Pak3	Rph3al	Slc7a14	Map1b
6	Shank2	Hmga2	Cbfa2t3	Ppp3ca	Anxa4	Rap1b	Cacna1d	Ncam1
7	Racgap1	Bicc1	Rock1	Rap1b	Entpd3	Gnas	Scgn	Tmem163
8	Smoc1	Smoc1	Rfx6	Gnas	Abcc8	Rap1gap2	Anxa4	Tspan7
9	Incenp	Wfdc15b	Eya2	Cacna1d	Ica1	Tmem163	Arg1	Ank

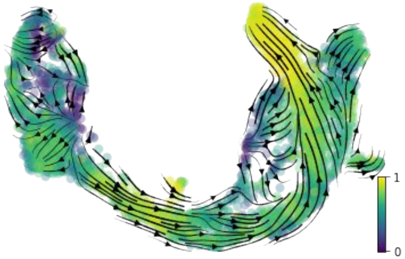
How to detect “relevant” genes?

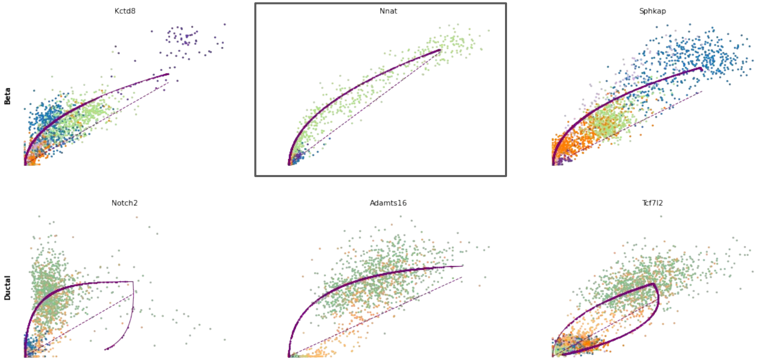
Top 5 dynamic genes per cluster





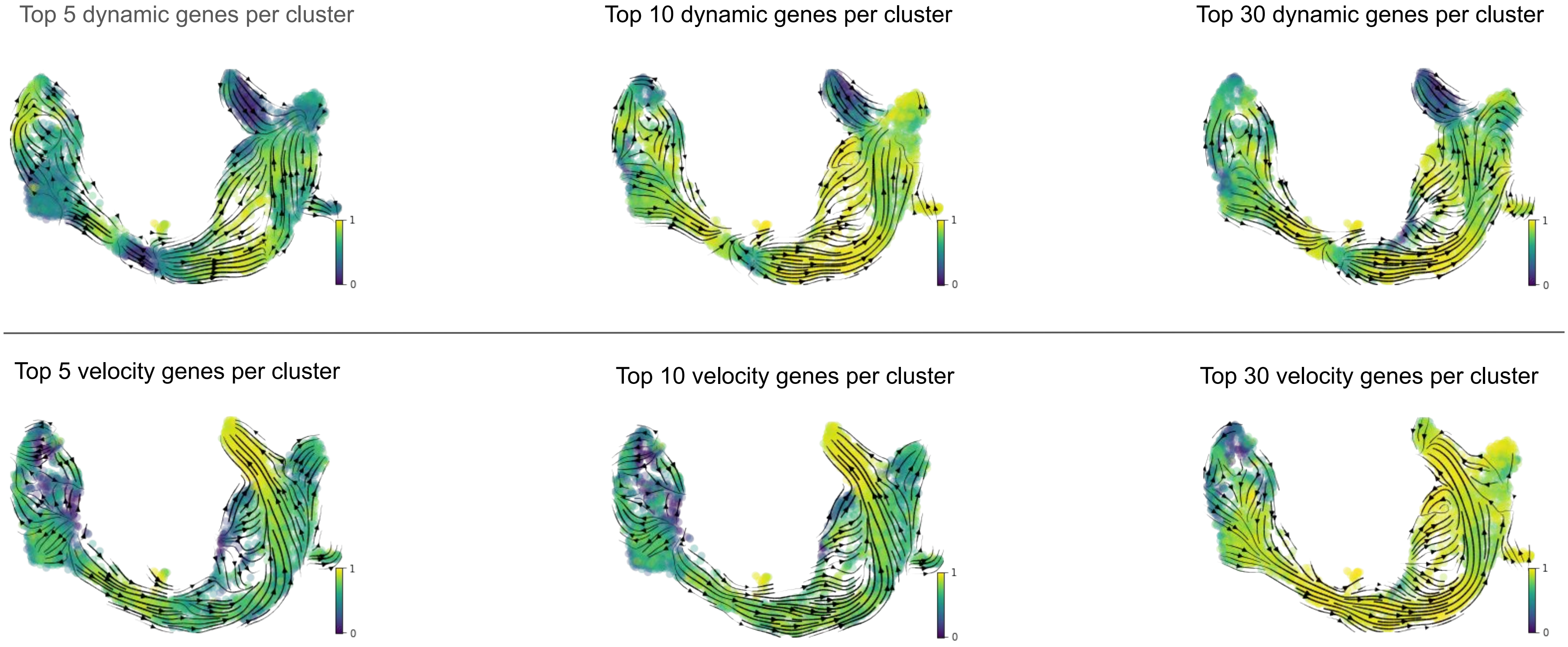
Top 5 velocity genes per cluster





Color scale: velocity correlation between gene-based projection & actual projection

How many “relevant” genes?



Top 5 dynamic genes per cluster

Top 10 dynamic genes per cluster

Top 30 dynamic genes per cluster

Top 5 velocity genes per cluster

Top 10 velocity genes per cluster

Top 30 velocity genes per cluster

Color scale: velocity correlation between gene-based projection & actual projection

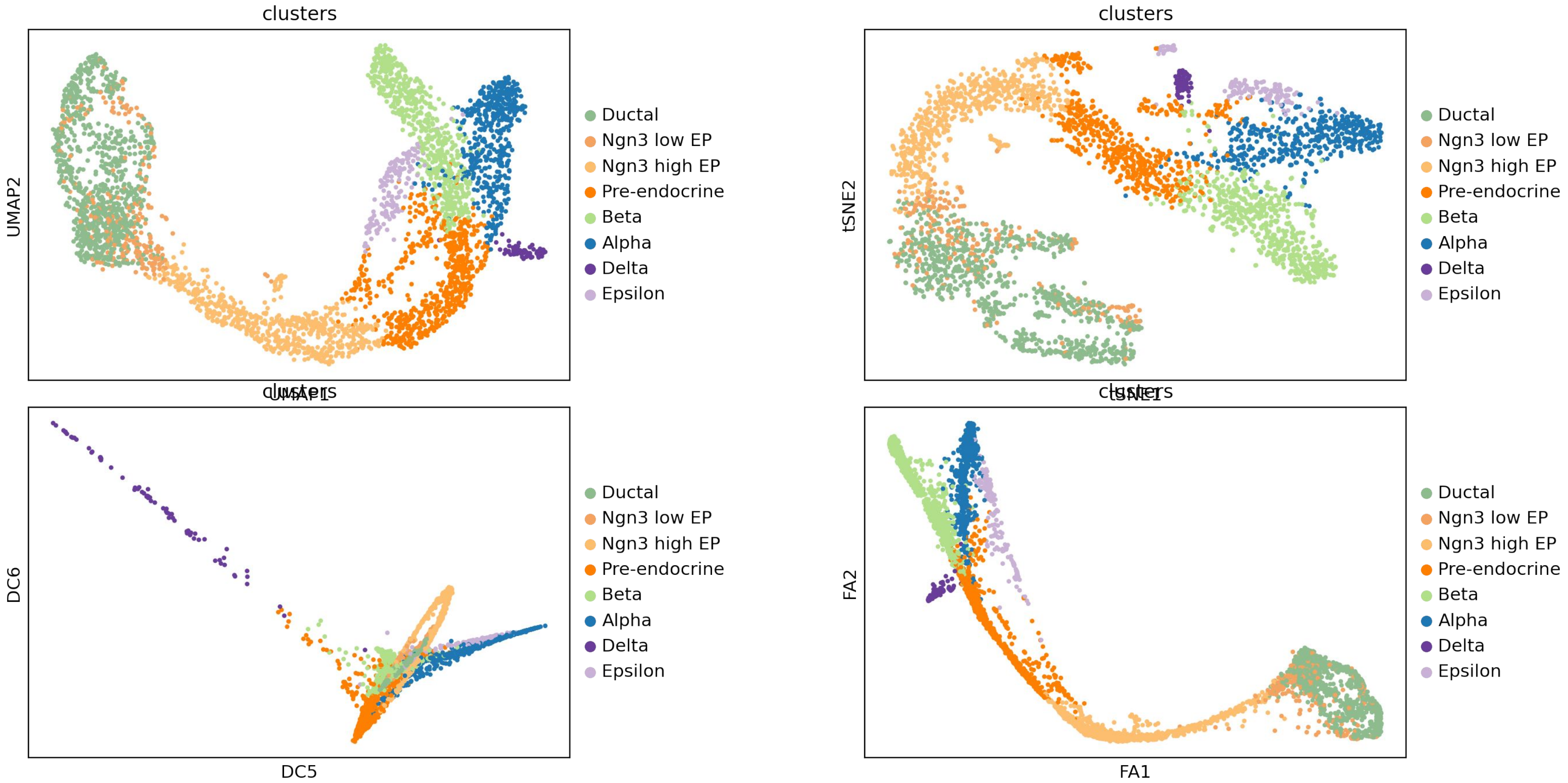
Discussion

- Transition matrix can be used to measure level of randomness in the velocity graph
- Driver genes can be detected based on different gene lists from literature, RNA velocity and dynamic modeling
  - Small number of gene velocities can account for velocity embedding
  - Quantification of embedding reconstruction based on velocity correlation
  - Biology is complex! - number of genes required for complete reconstruction of velocities varies from subtype to subtype

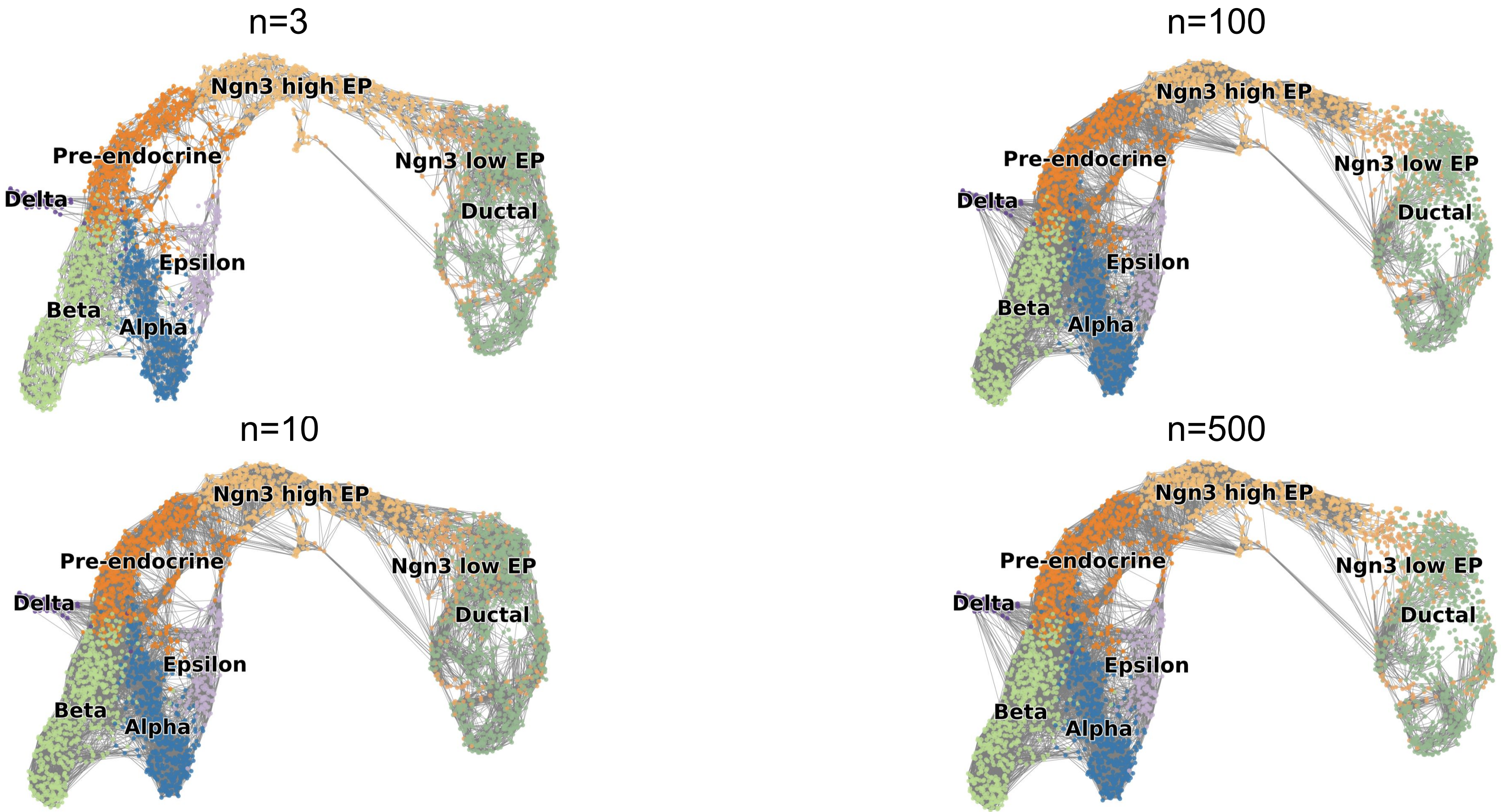
Thank you

Milestone 5

5.1.Dimensionality reduction methods / embeddings and topology



N_neighbors impacts the velocity vector field



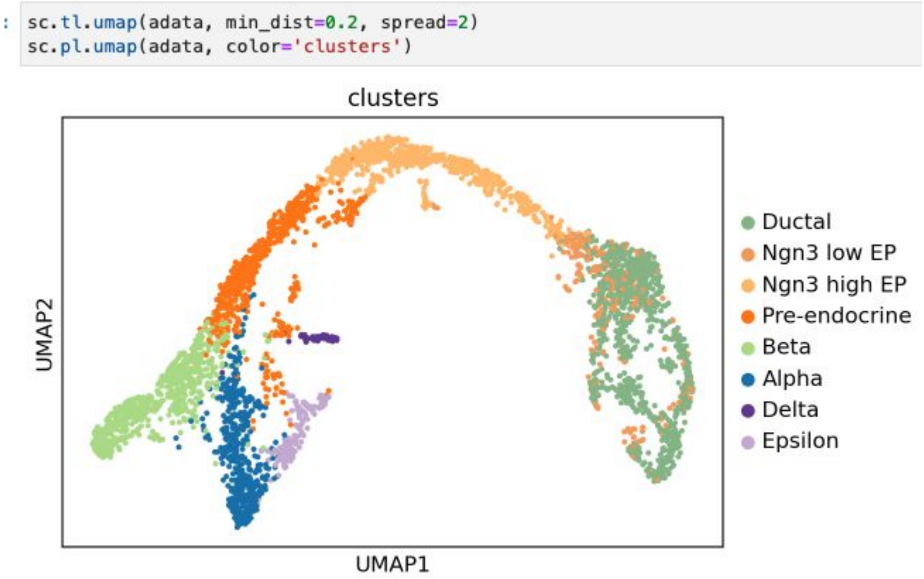
n=3

n=100

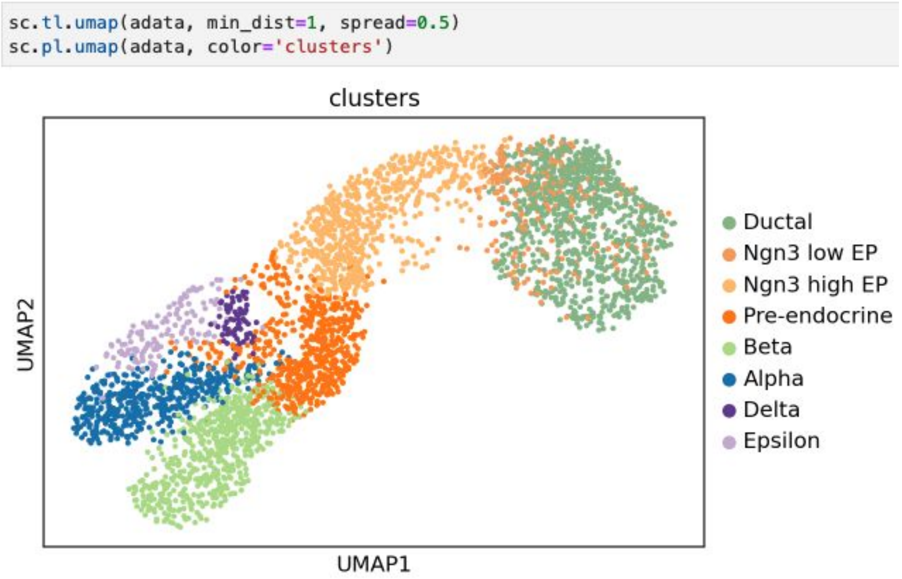
n=10

n=500

5.2. Main UMAP parameters impacts the embedding



```
sc.tl.umap(adata, min_dist=0.2, spread=2)
sc.pl.umap(adata, color='clusters')
```



```
sc.tl.umap(adata, min_dist=1, spread=0.5)
sc.pl.umap(adata, color='clusters')
```

UMAP is a decent trade-off between representing local and global topology, improvements can be made by adjusting parameters