# MuTrans Analysis of Epithelial-Mesenchymal Transition (EMT)

## Load the dataset and explore with Scanpy

In [1]:

```
import pandas as pd
import scanpy as sc
sc.settings.set_figure_params(dpi=100, frameon=False, figsize=(3, 3), facecolor='white')
import matplotlib.pyplot as plt
import pyMuTrans as pm

datadir = "../Data/"
adata = sc.read_csv(datadir+'data_emt.csv')
genes = pd.read_csv(datadir+'emt_genes.csv')
adata.var_names = genes['genes']
```

In [2]:

```
sc.pp.log1p(adata)
sc.pp.scale(adata)
sc.tl.tsne(adata,perplexity=40)
```

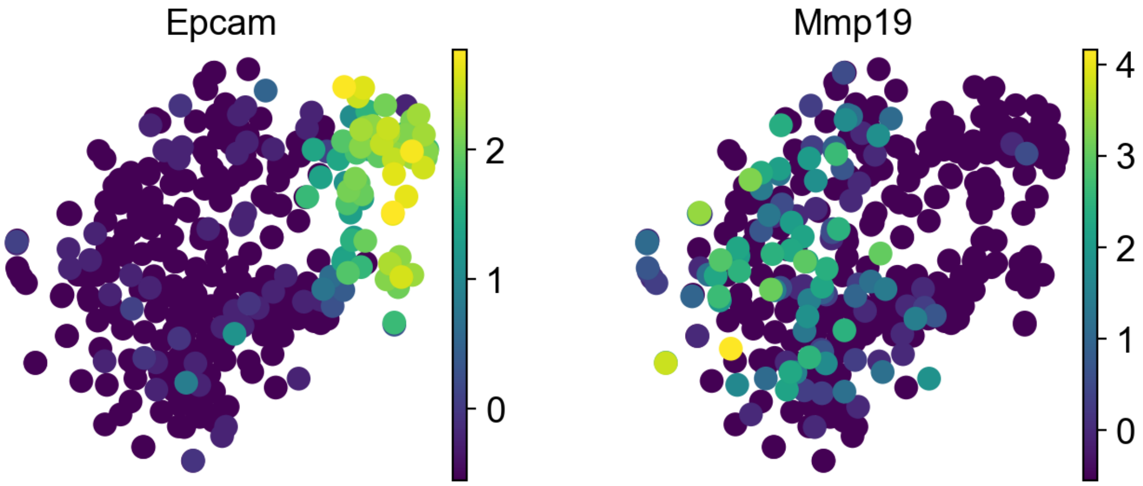
```
WARNING: You’re trying to run this on 1500 dimensions of `.X`, if you really want th
is, set `use_rep='X'`.
         Falling back to preprocessing with `sc.pp.pca` and default params.

/Users/cliffzhou/opt/anaconda3/lib/python3.7/site-packages/pyemma/__init__.py:92: Us
erWarning: You are not using the latest release of PyEMMA. Latest is 2.5.7, you have
2.5.6.
  .format(latest=latest, current=current), category=UserWarning)

WARNING: Consider installing the package MulticoreTSNE (https://github.com/DmitryUly
anov/Multicore-TSNE). Even for n_jobs=1 this speeds up the computation considerably
and might yield better converged results.
```

In [3]:

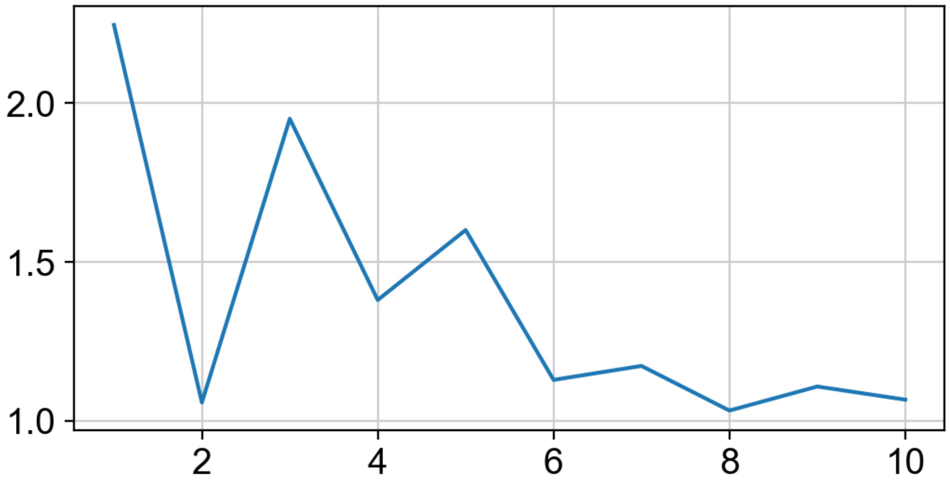
```
sc.pl.tsne(adata, color=['Epcam','Mmp19'])
```



## Estimate Cluster Number with EPI (Eigen-Peak Index)

In [4]:

```
fig = plt.figure(figsize=(6, 3))
par = {"choice_distance":"cosine"}
out = pm.plot_cluster_num(adata, par, k_plot= 10) #look for the peaks
```



## MuTrans Analysis Based on the AnnData Object

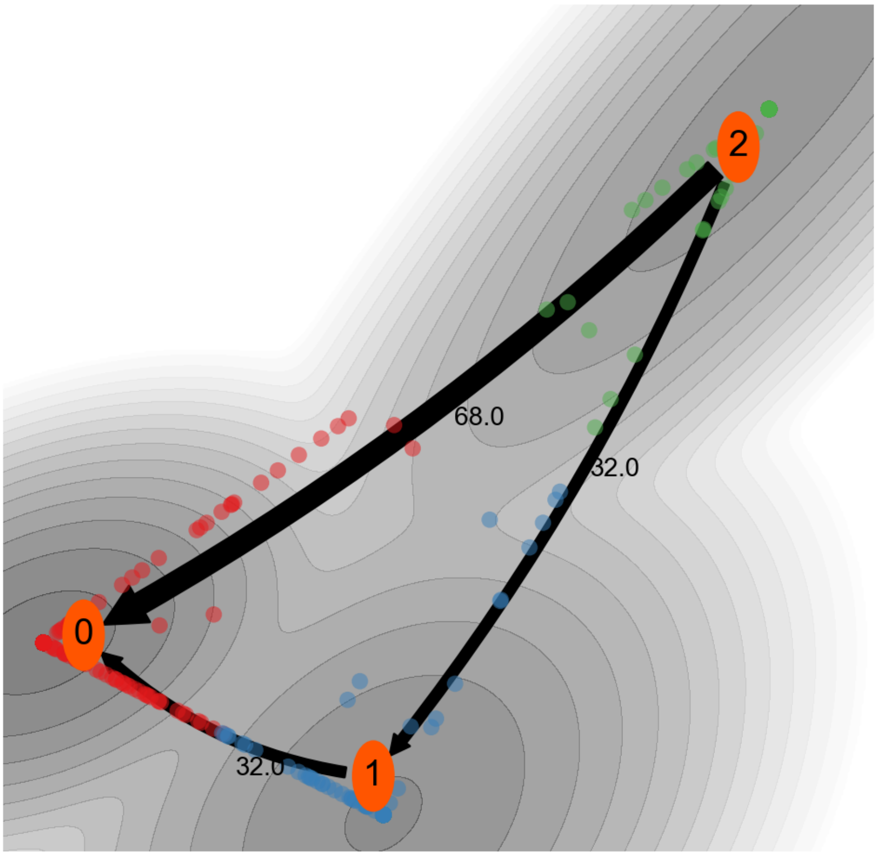
In [5]:

```
par = {"choice_distance":"cosine","perplex":150.0,"K_cluster":3.0,"trials":10}
adata = pm.dynamical_analysis(adata,par)
```

Infer the transition trajectories from Epithelial to Mesenchymal, with the MPPT (most probable path tree) approach

In [6]:

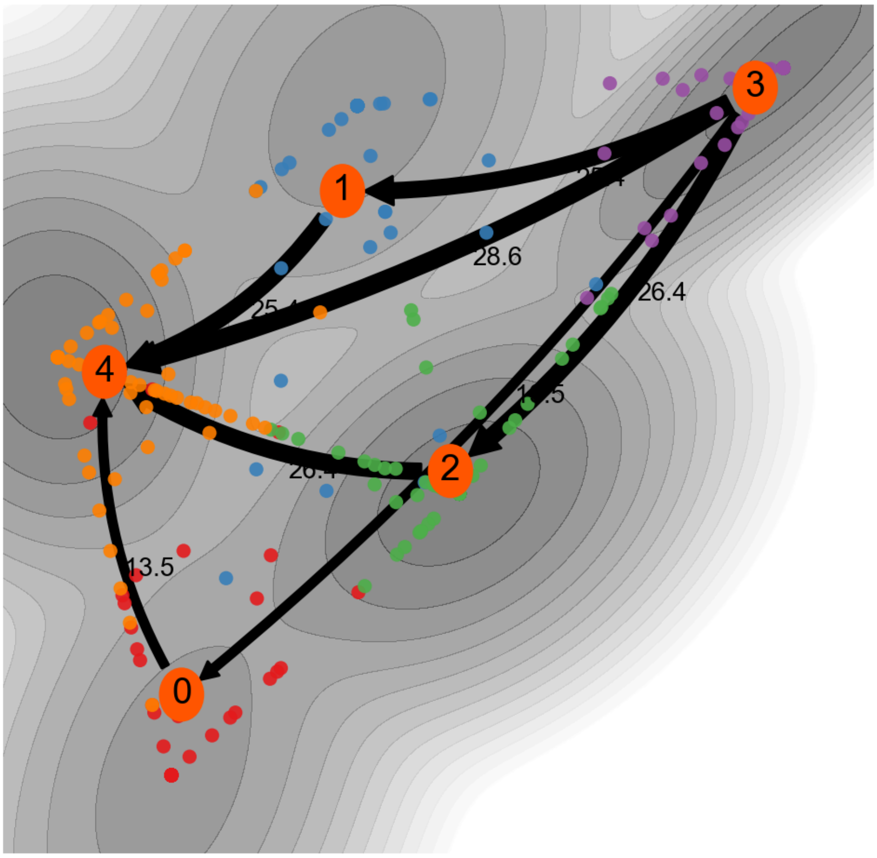
```
fig = plt.figure(figsize=(6, 6))
pm.infer_lineage(adata,si=2,sf=0,method = "MPPT",size_point =40, size_text = 10,alpha
_point = 0.5)
```



In the EPI figure, we actually observe another peak at K=5. What if we choose the parameter like this?

```
In [7]: par['K_cluster'] = 5.0
        adata = pm.dynamical_analysis(adata,par)
```

```
In [8]: fig = plt.figure(figsize=(6, 6))
        pm.infer_lineage(adata,si=3,sf=4,method = "MPPT",size_point =30, size_text = 10,alpha
        _point = 0.9)
```



Note that two new attractors appear (1 from the original "transition cells" and 0 from the "low-expression" state). However, we observe that the quanlitative conclusion about transition trajectory remains relatively robust, indicating that the transition flux mediated by transition state is larger than from low expression state.