

```
In [ ]: 1 import scanpy as sc
2 import anndata
3 import anndata as ad
4 from scipy import io
5 from scipy.sparse import coo_matrix, csr_matrix
6 import numpy as np
7 import os
8 import pandas as pd
9 import scvelo as scv
10 import cellrank as cr
11 import loompy
12 import seaborn as sns
13 import matplotlib.pyplot as plt

In [ ]: 1 X = io.mmread("cluster/all/mac.irf.counts.mtx")

In [ ]: 1

In [ ]: 1 cell_meta = pd.read_csv("cluster/all/mac.irf.metadata.csv")
2 with open("cluster/all/mac.irf.gene_names.csv", 'r') as f:

In [ ]: 1 adata.obs = cell_meta
2 adata.obs.index = adata.obs['barcode']

In [ ]: 1 pca = pd.read_csv("cluster/all/mac.irf.pca.csv")
2 pca.index = adata.obs.index
3 adata.obsm['X_pca'] = pca.to_numpy()
4 adata.obsm['X_umap'] = np.vstack((adata.obs['UMAP_1'].to_numpy(), a
5 adata.write("cluster/all/irf1.h5ad")

In [ ]: 1 scv.settings.verbosity = 3
2 scv.settings.set_figure_params('scvelo', facecolor='white', dpi=300

In [ ]: 1 s = sc.read_mtx("data_setup/irf/irf_exon.mtx")
2 u = sc.read_mtx("data_setup/irf/irf_intron.mtx")
3 s_genes = pd.read_csv("data_setup/irf/genes_irf_exon.txt", header=N
4 u_genes = pd.read_csv("data_setup/irf/genes_irf_intron.txt", header=
5
6 s_bcs = pd.read_csv("data_setup/irf/barcodes_irf_exon.txt", header=
7 u_bcs = pd.read_csv("data_setup/irf/barcodes_irf_intron.txt", heade
8

In [ ]: 1 s.obs = s_bcs
2 s.obs.index = s_bcs[0].values
3 s.obs.columns = ["bcs"]
4
5 u.obs = u_bcs
6 u.obs.index = u_bcs[0].values
7 u.obs.columns = ["bcs"]
8
9 s.var = s_genes
10 s.var.index = s_genes[0].values
11 s.var.columns = ["gid"]
12
13 u.var = u_genes
14 u.var.index = u_genes[0].values
15 u.var.columns = ["gid"]

In [ ]: 1 indo = (s.obs.index & u.obs.index)
```

```
2 indv = (s.var.index & u.var.index)
3
4 sadata = s[indv,indv]
5 uadata = u[indv,indv]
6
7 adata = sadata.copy()
8
9 adata.layers["spliced"] = sadata.X
10 adata.layers["unspliced"] = uadata.X
11
12 adata.obs["CellID"] = adata.obs.index
13 adata.var["Gene"] = adata.var.index
14
```

In [ ]: █

In [ ]: █

In [ ]: █

In [ ]: █

```
In [ ]: █ 1 sc.pl.umap(adata, color='seurat_clusters', palette={
2                      '0': "#8AB6F9",
3                      '1': "#B25690",
4                      '2': "#ffc13b",
5                      '3': "#04d4f0",
6                      '4': "#5c3c92",
7                      '5': "#077b8a",
8                      '6': "#d72631",
9                      '7': "#c38b72",
10                     '8': "#80c904", '9': "#bbc0b6", '10': "#bbc0b6"}
```

In [ ]: █

In [ ]: █ 1 scv.pp.filter\_and\_normalize(adata)

In [ ]: █ 1 scv.tl.velocity(adata, mode='stochastic')

In [ ]: █

In [ ]: █

In [ ]: █ 1 scv.pl.velocity\_embedding\_stream(adata, basis='umap', color=['seurat\_clusters'])

```
In [ ]: █ 1 scv.pl.velocity(adata, var_names=['Cx3cr1'], color='seurat_clusters')
2 scv.pl.velocity(adata, var_names=['Ccr2'], color='seurat_clusters'),
3 scv.pl.velocity(adata, var_names=['Mrc1'], color='seurat_clusters'),
4 scv.pl.velocity(adata, var_names=['Csf1r'], color='seurat_clusters')
5 scv.pl.velocity(adata, var_names=['Il11rb1'], color='seurat_clusters'),
6 scv.pl.velocity(adata, var_names=['Csf2rb'], color='seurat_clusters'),
7 scv.pl.velocity(adata, var_names=['Cd74'], color='seurat_clusters'),
8 scv.pl.velocity(adata, var_names=['Il11b'], color='seurat_clusters'),
9 scv.pl.velocity(adata, var_names=['Tnf'], color='seurat_clusters'),
```

```
In [ ]: 1 scv.tl.rank_velocity_genes(adata, groupby='seurat_clusters', min_co  
2 df = scv.DataFrame(adata.uns['rank_velocity_genes']['names'])  
3 df.head(100)
```

```
In [ ]: 1 scv.pl.scatter(adata, df['0'][:5], ylabel='Cluster 0', frameon=False  
2 scv.pl.scatter(adata, df['2'][:5], ylabel='Cluster 2', frameon=False  
3 scv.pl.scatter(adata, df['3'][:5], ylabel='Cluster 3', frameon=False  
4 scv.pl.scatter(adata, df['5'][:5], ylabel='Cluster 5', frameon=False)
```

```
In [ ]: 1 scv.tl.velocity_confidence(adata)  
2 keys = 'velocity_length', 'velocity_confidence'  
3 scv.pl.scatter(adata, c=keys, cmap='coolwarm', perc=[5, 95], size=8  
4
```

```
In [ ]:
```

```
In [ ]: 1 x, y = scv.utils.get_cell_transitions(adata, basis='umap', starting  
2 ax = scv.pl.velocity_graph(adata, c='lightgrey', edge_width=.05, sh  
3 ax = scv.pl.scatter(adata, x=x, y=y, s=120, c='ascending', cmap='gn
```

```
In [ ]: 1 scv.tl.velocity_pseudotime(adata)  
2 scv.pl.scatter(adata, color='velocity_pseudotime', cmap='gnuplot')
```

```
In [ ]: 1 scv.pl.velocity_embedding_stream(adata, basis='umap', color=['veloc
```

```
In [ ]: 1 # this is needed due to a current bug - bugfix is coming soon.  
2 adata.uns['neighbors']['distances'] = adata.obsp['distances']  
3 adata.uns['neighbors']['connectivities'] = adata.obsp['connectivities']  
4  
5 scv.tl.paga(adata, groups='seurat_clusters')  
6 df = scv.get_df(adata, 'paga/transitions_confidence', precision=2).  
7  
8 scv.pl.paga(adata, basis='umap', size=50, alpha=.25,  
9 min edge width=2. node size scale=1.5. title='', dpi=60
```

```
In [ ]: 1 cur_celltypes = ['0', '1', '2', '3', '4', '5', '6', '7', '8']  
2 adata_subset = adata[adata.obs['seurat_clusters'].isin(cur_celltype)  
3 scv.pl.umap(adata_subset, color=['seurat_clusters'], palette={  
4 '0': "#8AB6F9",  
5 '1': "#B25690",  
6 '2': "#ffc13b",  
7 '3': "#04d4f0",  
8 '4': "#5c3c92",  
9 '5': "#077b8a",  
10 '6': "#d72631",  
11 '7': "#c38b72",  
12 '8': "#80c904"}, frameon=False, title='IRF8k
```

```
In [ ]:
```

```
In [ ]: 1 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu  
2 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu  
3 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu  
4 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu  
5 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu  
6 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu  
7 scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
```

```
In [ ]: 1 sc.pp.neighbors(adata_subset, n_neighbors=15, use_rep='X_pca')
2 # pre-process
3 scv.pp.filter_and_normalize(adata_subset)
4 scv.pp.moments(adata_subset)
```

```
In [ ]: 1 scv.tl.velocity_pseudotime(adata_subset)
```

```
In [ ]: 1 adata_subset.uns['neighbors']['distances'] = adata_subset.obsp['dis
2 adata_subset.uns['neighbors']['connectivities'] = adata_subset.obsp
3
4 scv.tl.paga(adata_subset, groups='seurat_clusters')
5 df = scv.get_df(adata_subset, 'paga/transitions_confidence', precis
6
7 scv.pl.paga(adata_subset, basis='umap', size=50, alpha=.1, title = '
```

```
In [ ]:
```

```
In [ ]: 1 scv.tl.velocity(adata_subset, mode='dynamical')
```

```
In [ ]: 1 scv.pl.velocity_embedding_stream(adata_subset, basis='umap', color=
```

```
In [ ]: 1 df = adata_subset.var
2 df = df[(df['fit_likelihood'] > .1) & df['velocity_genes'] == True]
3
4 kwargs = dict(xscale='log', fontsize=16)
5 with scv.GridSpec(ncols=3) as pl:
6     pl.hist(df['fit_alpha'], xlabel='transcription rate', **kwargs)
7     pl.hist(df['fit_beta'] * df['fit_scaling'], xlabel='splicing ra
8     pl.hist(df['fit_gamma'], xlabel='degradation rate', xticks=[.1,
9
10 scv.get_df(adata_subset, 'fit*', dropna=True).head()
```

```
In [ ]: 1 scv.tl.latent_time(adata_subset)
```

```
In [ ]: 1 scv.tl.velocity_graph(adata_subset, mode_neighbors='connectivities')
2 scv.tl.terminal_states(adata_subset)
3 scv.tl.latent_time(adata_subset)
4 scv.pl.scatter(adata_subset, color=['root_cells', 'end_points'], co
5 scv.pl.scatter(adata_subset, color='latent_time', color_map='coolwar
```

```
In [ ]: 1 top_genes = adata_subset.var['fit_likelihood'].sort_values(ascendin
2 plt.figure(dpi = 600)
3 scv.pl.heatmap(adata_subset, var_names=top_genes, color_map='coolwar
4 col_color='seurat_clusters', n_convolve=50)
```

```
In [ ]: 1 top_genes = adata_subset.var['fit_likelihood'].sort_values(ascendin
2 scv.pl.scatter(adata_subset, color='seurat_clusters', basis=top_gen
```

```
In [ ]: 1 var_names = ['Nfkb1', 'Cd44', 'Il1b', 'Tnfrsf1b', 'Cd74', 'Mrc1']
2 scv.pl.scatter(adata_subset, var_names, color='seurat_clusters', fr
3 scv.pl.scatter(adata_subset, x='latent_time', y=var_names, color='s
4
```

```
In [ ]: 1 celltypes4 = ['0', '1', '4']
2 subset4 = adata[adata.obs['seurat_clusters'].isin(celltypes4)]
3 scv.pl.umap(subset4, color=['seurat_clusters'], palette={
4     '0': "#8AB6F9",
```

```
In [ ]: 1 sc.pp.neighbors(subset4, n_neighbors=15, use_rep='X_pca')
2 scv.pp.filter_and_normalize(subset4)
3 scv.pp.moments(subset4)
4 scv.tl.velocity_pseudotime(subset4)
5 scv.pl.scatter(subset4, color='velocity_pseudotime', cmap='coolwarm')

In [ ]: 1 subset4.uns['neighbors']['distances'] = subset4.obsp['distances']
2 subset4.uns['neighbors']['connectivities'] = subset4.obsp['connecti
3
4 scv.tl.paga(subset4, groups='seurat_clusters')
5 df = scv.get_df(subset4, 'paga/transitions_confidence', precision=2
6
7 scv.pl.paga(subset4, basis='umap', size=50, alpha=.1)

In [ ]: 1 scv.tl.recover_dynamics(subset4)
2 scv.tl.velocity(subset4, mode='dynamical')

In [ ]: 1 scv.tl.velocity_graph(subset4, mode_neighbors='connectivities')
2 scv.tl.terminal_states(subset4)
3 scv.tl.latent_time(subset4)
4 scv.pl.scatter(subset4, color=['root cells', 'end points'], color_m

In [ ]: 1 var_genes4 = subset4.var['fit_likelihood'].sort_values(ascending=False)
2 plt.figure(dpi = 600)
3 ylabels4=(None, None, None, None, None, None, None, None, None, None, No
4 scv.pl.heatmap(subset4, var_names=var_genes4, color_map='coolwarm',
5 sortby='latent_time', sort=True, col_color='seurat_)

In [ ]: 1 pd.options.display.max_seq_items = 1000

In [ ]: 1 df = pd.DataFrame(var_genes4)

In [ ]: 1 celltypes3 = ['2', '3', '6']
2 subset3 = adata[adata.obs['seurat_clusters'].isin(celltypes3)]
3 scv.pl.umap(subset3, color=['seurat_clusters'], palette={
4 '2': "#ffc13b",
5 '3': "#04d4f0",

In [ ]: 1 sc.pp.neighbors(subset3, n_neighbors=15, use_rep='X_pca')
2 scv.pp.filter_and_normalize(subset3)
3 scv.pp.moments(subset3)
4 scv.tl.velocity_pseudotime(subset3)
5 scv.pl.scatter(subset3, color='velocity_pseudotime', cmap='coolwarm')

In [ ]: 1 subset3.uns['neighbors']['distances'] = subset3.obsp['distances']
2 subset3.uns['neighbors']['connectivities'] = subset3.obsp['connecti
3
4 scv.tl.paga(subset3, groups='seurat_clusters')
5 df = scv.get_df(subset3, 'paga/transitions_confidence', precision=2
6
7 scv.pl.paga(subset3, basis='umap', size=50, alpha=.2, color_map='co

In [ ]: 1 scv.tl.recover_dynamics(subset3)
2 scv.tl.velocity(subset3, mode='dynamical')
```

```
In [ ]: █
```

```
In [ ]: █
```

```
1 scv.tl.velocity_graph(subset3, mode_neighbors='connectivities')
2 scv.tl.terminal_states(subset3)
3 scv.tl.latent_time(subset3)
4 scv.pl.scatter(subset3, color=['root cells', 'end points'], color_m
```

```
In [ ]: █
```

```
1 var_genes3 = subset3.var['fit_likelihood'].sort_values(ascending=False)
2 plt.figure(dpi = 600)
3 ylabels3=(None,None,None,None,None,None,None,None,None,None,None,None)
4 scv.pl.heatmap(subset3, var_names=var_genes3, color_map='coolwarm',
5                 sortby= 'latent_time', sort=True, col_color='seurat_
```

```
In [ ]: █
```

```
1 pd.options.display.max_seq_items = 1000
```

```
In [ ]: █
```

```
1 df = pd.DataFrame(var_genes3)
```