In [ ]:
```
import scanpy as sc
import anndata
import anndata as ad
from scipy import io
from scipy.sparse import coo_matrix, csr_matrix
import numpy as np
import os
import pandas as pd
import scvelo as scv
import cellrank as cr
import loompy
import seaborn as sns
```

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

In [ ]:

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

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

In [ ]:
```
pca = pd.read_csv("cluster/all/mac.wt.pca.csv")
pca.index = adata.obs.index
adata.obsm['X_pca'] = pca.to_numpy()
adata.obsm['X_umap'] = np.vstack((adata.obs['UMAP_1'].to_numpy(), a
```

In [ ]:

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

In [ ]:
```
s = sc.read_mtx("data_setup/wt/wt_exon.mtx")
u = sc.read_mtx("data_setup/wt/wt_intron.mtx")
s_genes = pd.read_csv("data_setup/wt/genes_wt_exon.txt", header=Non
u_genes = pd.read_csv("data_setup/wt/genes_wt_intron.txt", header=N

s_bcs = pd.read_csv("data_setup/wt/barcodes_wt_exon.txt", header=No
```

In [ ]:
```
s.obs = s_bcs
s.obs.index = s_bcs[0].values
s.obs.columns = ["bcs"]

u.obs = u_bcs
u.obs.index = u_bcs[0].values
u.obs.columns = ["bcs"]

s.var = s_genes
s.var.index = s_genes[0].values
s.var.columns = ["gid"]

u.var = u_genes
u.var.index = u_genes[0].values
```

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

```

```
sadata = s[indo,indv]
uadata = u[indo,indv]

adata = sadata.copy()

adata.layers["spliced"] = sadata.X
adata.layers["unspliced"] = uadata.X

adata.obs["CellID"] = adata.obs.index
adata.var["Gene"] = adata.var.index

```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

```
sc.pl.umap(adata, color='seurat_clusters', palette={
                          '0':"#8AB6F9",
                         '1':"#B25690",
                         '2':"#ffc13b",
                         '3':"#04d4f0",
                         '4':"#5c3c92",
                         '5':"#077b8a",
                         '6':"#d72631",
                         '7':"#c38b72",
                         '8': "#80c904", '9': "#bbc0b6", '10': "#bbc0
```

In [ ]:

In [ ]:

```
scv.pp.filter_and_normalize(adata)
```

In [ ]:

```
scv.tl.velocity(adata, mode='stochastic')
```

In [ ]:

In [ ]:

In [ ]:

```
scv.pl.velocity_embedding_stream(adata, basis='umap', color=['seura
```

In [ ]:

```
#scv.pl.velocity_embedding_stream(adata, basis='umap', color=['velo
```

In [ ]:

```
scv.pl.velocity(adata, var_names=['Ly6c2'], color='seurat_clusters'
scv.pl.velocity(adata, var_names=['Cx3cr1'], color='seurat_clusters
scv.pl.velocity(adata, var_names=['Ccr2'], color='seurat_clusters',
scv.pl.velocity(adata, var_names=['Mrc1'], color='seurat_clusters',
scv.pl.velocity(adata, var_names=['Csf1r'], color='seurat_clusters'
scv.pl.velocity(adata, var_names=['Il12rb1'], color='seurat_cluster
scv.pl.velocity(adata, var_names=['Csf2rb'], color='seurat_clusters
scv.pl.velocity(adata, var_names=['Cd74'], color='seurat_clusters',
scv.pl.velocity(adata, var_names=['Il1b'], color='seurat_clusters',
scv.pl.velocity(adata, var_names=['Tnf'], color='seurat_clusters',
```

```
11
```

In [ ]:

In [ ]:
```
scv.tl.rank_velocity_genes(adata, groupby='seurat_clusters', min_co
df = scv.DataFrame(adata.uns['rank_velocity_genes']['names'])
df.head(100)
df.to_pickle('wt.rank.genes.csv')
```

In [ ]:
```
scv.pl.scatter(adata, df['0'][:5], ylabel='Cluster 0', frameon=Fals
scv.pl.scatter(adata, df['3'][:5], ylabel='Cluster 3', frameon=Fals
scv.pl.scatter(adata, df['4'][:5], ylabel='Cluster 4', frameon=Fals
scv.pl.scatter(adata, df['6'][:5], ylabel='Cluster 6', frameon=Fals
```

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

In [ ]:

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

In [ ]:
```
scv.tl.velocity_pseudotime(adata)
scv.pl.scatter(adata, color='velocity_pseudotime', cmap='gnuplot',t
```

In [ ]:
```
scv.tl.recover_dynamics(adata)
```

In [ ]:
```
scv.tl.velocity_graph(adata, mode_neighbors='connectivities')
scv.tl.terminal_states(adata)
scv.tl.latent_time(adata)
scv.pl.scatter(adata, color=[ 'root_cells', 'end_points'],title = '
```

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

In [ ]:
```
# current bug
adata.uns['neighbors']['distances'] = adata.obsp['distances']
adata.uns['neighbors']['connectivities'] = adata.obsp['connectiviti

scv.tl.paga(adata, groups='seurat_clusters')
df = scv.get_df(adata, 'paga/transitions_confidence', precision=2).

scv.pl.paga(adata, basis='umap', size=50, alpha=.25,
            min_edge_width=2, node_size_scale=1.5, title='', dpi=60
```

In [ ]:

In [ ]:
```
cur_celltypes = ['0', '1', '2', '3', '4', '5', '6', '7', '8']
adata_subset = adata[adata.obs['seurat_clusters'].isin(cur_celltype
sc.pl.umap(adata_subset, color=['seurat_clusters'],palette={
                        '0':"#8AB6F9",
                        '1':"#B25690",
                        '2':"#ffc13b",
                        '3':"#04d4f0",
                        '4':"#5c3c92",
                        '5':"#077b8a",
                        '6':"#d72631",
```

```
                    '7':"#c38b72",
                    '8': "#80c904"}, frameon=False, title = '',
```

In [ ]:

In [ ]:
```
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu
scv.pl.velocity_graph(adata_subset, threshold=.1, color='seurat_clu


```

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

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

In [ ]:
```
adata_subset.uns['neighbors']['distances'] = adata_subset.obsp['dis
adata_subset.uns['neighbors']['connectivities'] = adata_subset.obsp

scv.tl.paga(adata_subset, groups='seurat_clusters')
df = scv.get_df(adata_subset, 'paga/transitions_confidence', precis

scv.pl.paga(adata_subset, basis='umap', size=50, alpha=.1,
```

In [ ]:

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

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

In [ ]:
```
df = adata_subset.var
df = df[(df['fit_likelihood'] > .1) & df['velocity_genes'] == True]

kwargs = dict(xscale='log', fontsize=16)
with scv.GridSpec(ncols=3) as pl:
    pl.hist(df['fit_alpha'], xlabel='transcription rate', **kwargs)
    pl.hist(df['fit_beta'] * df['fit_scaling'], xlabel='splicing ra
    pl.hist(df['fit_gamma'], xlabel='degradation rate', xticks=[.1,

scv.get_df(adata_subset, 'fit*', dropna=True).head()
```

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

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

In [ ]:
```
top_genes = adata_subset.var['fit_likelihood'].sort_values(ascendin

scv.pl.heatmap(adata_subset,  var_names=top_genes,color_map='coolwa
```

```
                figsize=(4, 7), font_scale=0.8, sortby= 'latent time
```

In [ ]:
```
df = pd.DataFrame(top_genes)
```

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

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

```

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

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

In [ ]:
```
subset4.uns['neighbors']['distances'] = subset4.obsp['distances']
subset4.uns['neighbors']['connectivities'] = subset4.obsp['connecti

scv.tl.paga(subset4, groups='seurat_clusters')
df = scv.get_df(subset4, 'paga/transitions_confidence', precision=2

scv.pl.paga(subset4, basis='umap', size=50, alpha=.1,
```

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

In [ ]:
```

```

In [ ]:
```
scv.tl.velocity_graph(subset4, mode_neighbors='connectivities')
scv.tl.terminal_states(subset4)
scv.tl.latent_time(subset4)
scv.pl.scatter(subset4, color=[ 'root_cells', 'end_points'],color_m
```

In [ ]:
```
var_genes4 = subset4.var['fit_likelihood'].sort_values(ascending=Fa
plt.figure(dpi = 600)
ylabels=(None,None,'Cd44',None,None,None,None,None,None,None,None,N
scv.pl.heatmap(subset4, var_names=var_genes4, color_map='coolwarm',
               sortby= 'latent_time', sort=True, col_color='seurat_
```

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

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

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

In [ ]:
```
sc.pp.neighbors(subset3, n_neighbors=15, use_rep='X_pca')
scv.pp.filter_and_normalize(subset3)
```

```
scv.pp.moments(subset3)
scv.tl.velocity_pseudotime(subset3)
scv.pl.scatter(subset3, color='velocity_pseudotime', cmap='coolwarm
```

In [ ]:

```
subset3.uns['neighbors']['distances'] = subset3.obsp['distances']
subset3.uns['neighbors']['connectivities'] = subset3.obsp['connecti

scv.tl.paga(subset3, groups='seurat_clusters')
df = scv.get_df(subset3, 'paga/transitions_confidence', precision=2

scv.pl.paga(subset3, basis='umap', size=50, alpha=.1,
```

In [ ]:

```
scv.tl.recover_dynamics(subset3)
scv.tl.velocity(subset3, mode='dynamical')
```

In [ ]:

In [ ]:

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

In [ ]:

```
var_genes3 = subset3.var['fit_likelihood'].sort_values(ascending=Fa
plt.figure(dpi = 600)
ylabels3=(None,None,'Il1b',None,None,None,None,None,None,None,None,
)
scv.pl.heatmap(subset3, var_names=var_genes3, color_map='coolwarm',
               sortby= 'latent_time', sort=True, col_color='seurat_
```

In [ ]:

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

In [ ]:

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

In [ ]: