COG-56 RNA velocity using scVelo in R

- scVelo supports a full dynamical model and various of utility functions, but only supports Python.
- By utlizing R package 'reticulate', we can perform RNA velocity by Python-based scVelo in R
- Reticulte is a package that we can run python code from R.
- Reference: https://statbiomed.github.io/SingleCell-Workshop-2021/RNA-velocity.html#getting-started
- Dataset: A549 cell line

(0) Create conda environment for scVelo

- Install Python package related to scVelo

```
(base) [tuchiyama@davinci tuchiyama]$ conda create -n scvelo python=3.10
&& conda activate scvelo
(scvelo) [tuchiyama@davinci tuchiyama]$ pip install numpy==1.23
pandas==1.5.3 matplotlib==3.6.0 scanpy==1.9.1 igraph==0.9.8 scvelo==0.2.4
loompy==3.0.6 anndata==0.8.0
```

- Test the work

```
import scanpy as sc
import scvelo as scv
from anndata import read as read_h5ad

scv.logging.print_version()

scv.settings.verbosity = 3  # show errors(0), warnings(1), info(2),
hints(3)
scv.settings.presenter_view = True  # set max width size for presenter
view
scv.set_figure_params('scvelo')  # for beautified visualization

file = '/wgbs_global1/davinci/tuchiyama/A549/output/A549.h5ad.gz'
adata = read_h5ad(file)

scv.pl.proportions(adata)

scv.pp.filter_and_normalize(adata, min_shared_counts=20, n_top_genes=2000)
scv.pp.moments(adata, n_pcs=30, n_neighbors=30)
sc.tl.umap(adata, n_components=2)
scv.tl.velocity(adata)
scv.tl.velocity_graph(adata)
```

```
scv.pl.velocity_embedding_stream(
    adata,
    basis='X_umap',
    size=120,
    alpha=0.4,
    title='A549 cell line',
    legend_fontweight='heavy',
    legend_fontsize=20,
    figsize=(10, 7)
)

scv.tl.velocity_pseudotime(adata)
scv.pl.velocity_embedding_stream(
    adata,
    basis='X_umap',
    color='velocity_pseudotime',
    color_map='Spectral_r',
    size=120,
    alpha=0.4,
    legend_fontweight='heavy',
    legend_fontsize=20,
    figsize=(10, 7)
)
```

(1) scVelo by R

- Install R via conda

```
(base) [tuchiyama@davinci ~]$ conda activate scvelo
(scvelo) [tuchiyama@davinci ~]$ conda install -c r r-essentials
(scvelo) [tuchiyama@davinci tuchiyama]$ which R
/wgbs/scratch/tuchiyama/anaconda3/envs/scvelo/bin/R

# install reticulate ---
> if (!require('BiocManager', quietly = TRUE))
    install.packages('BiocManager')

BiocManager::install('reticulate')
BiocManager::install('SingleCellExperiment')
BiocManager::install('zellkonverter')
install.packages('Matrix')
```

- Load MatrixData in current CogentDS fashion (Pattern A)

```
library(Matrix)
library(SingleCellExperiment)
library(zellkonverter)
Registered S3 method overwritten by 'zellkonverter':
  method                from
```

```
  py_to_r.numpy.ndarray reticulate

gene_mtx <-
readMM('/wgbs_global1/davinci/tuchiyama/A549/output/genes.mtx.gz')
barcode <-
read.delim('/wgbs_global1/davinci/tuchiyama/A549/output/barcodes.tsv',
header=FALSE)
colnames(barcode) <- 'cell'
gene <-
read.delim('/wgbs_global1/davinci/tuchiyama/A549/output/genes.tsv',
header=TRUE)
gene_mtx <- t(gene_mtx)
colnames(gene_mtx) <- barcode$cell
rownames(gene_mtx) <- gene$gene_name

sce <- SingleCellExperiment(assays = list(counts=as(gene_mtx,
'dgCMatrix')), colData=barcode, rowData=gene)

intron_mtx <-
readMM('/wgbs_global1/davinci/tuchiyama/A549/output/introns.mtx.gz')
intron_mtx <- t(intron_mtx)
colnames(intron_mtx) <- barcode$cell
rownames(intron_mtx) <- gene$gene_name
assay(sce, 'spliced') <- gene_mtx
assay(sce, 'unspliced') <- intron_mtx

cellgroup <-
'/wgbs_global1/davinci/tuchiyama/A549/metadata_sample_info.csv'
df_cellgroup <- read.csv(cellgroup)
df_cellgroup <- df_cellgroup[order(df_cellgroup$Barcode), ]
rownames(df_cellgroup) <- 1:nrow(df_cellgroup)
sce$clusters <- df_cellgroup$BC1

# Convert SingleCellExperiment class to AnnData object ---
adata <- SCE2AnnData(sce)

# Check output ---
adata
AnnData object with n_obs × n_vars = 11000 × 58735
    obs: 'cell', 'clusters'
    var: 'spliced', 'gene_name', 'gene_biotype'
    uns: 'X_name'
    layers: 'spliced', 'unspliced'
```

- Load MatrixData in Python fashion (Pattern B)

```
library(reticulate)
conda_list()
    name                                                    python
1   base             /wgbs/scratch/tuchiyama/anaconda3/bin/python
2 salmon /wgbs/scratch/tuchiyama/anaconda3/envs/salmon/bin/python
3 scvelo /wgbs/scratch/tuchiyama/anaconda3/envs/scvelo/bin/python
```

```
# choose conda environment ---
use_condaenv('scvelo')

# load Python packages ---
ann <- import('anndata')
scv <- import('scvelo')
sc <- import('scanpy')
plt <- import('matplotlib.pyplot', as='plt')

# check scVelo version ---
scv$logging$print_version()
Running scvelo 0.2.4 (python 3.10.13) on 2023-09-25 17:22.
ERROR: XMLRPC request failed [code: -32500]
RuntimeError: PyPI no longer supports 'pip search' (or XML-RPC search).
Please use https://pypi.org/search (via a browser) instead. See
https://warehouse.pypa.io/api-reference/xml-rpc.html#deprecated-methods
for more information.
scv$settings$presenter_view = TRUE
scv$set_figure_params('scvelo')

# Load MatrixData
file <- '/wgbs_global1/davinci/tuchiyama/A549/output/A549.h5ad.gz'
adata <- ann$read_h5ad(file)
```

- Preprocess MatrixData based on scVelo's tutorial

```
file <-
'/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.proportions.
png'
scv$pl$proportions(adata, show=FALSE, save=file)

saving figure to file
/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.proportions.p
ng
[[1]]
<AxesSubplot: >

[[2]]
<AxesSubplot: xlabel='proportions', ylabel='clusters'>

scv$pp$filter_and_normalize(adata, min_shared_counts=as.integer(20),
n_top_genes=as.integer(2000))
WARNING: Did not normalize X as it looks processed already. To enforce
normalization, set `enforce=True`.
WARNING: Did not normalize spliced as it looks processed already. To
enforce normalization, set `enforce=True`.
WARNING: Did not normalize unspliced as it looks processed already. To
enforce normalization, set `enforce=True`.
Extracted 2000 highly variable genes.
Logarithmized X.
```

```
scv$pp$moments(adata, n_pcs=as.integer(30), n_neighbors=as.integer(30))
computing neighbors
    finished (0:00:22) --> added
    'distances' and 'connectivities', weighted adjacency matrices
(adata.obsp)
computing moments based on connectivities
    finished (0:00:04) --> added
    'Ms' and 'Mu', moments of un/spliced abundances (adata.layers)

sc$tl$umap(adata, n_components=as.integer(2))
```

- Estimate RNA velocity

```
scv$tl$velocity(adata)
    finished (0:00:11) --> added
    'velocity', velocity vectors for each individual cell (adata.layers)

scv$tl$velocity_graph(adata)
computing velocity graph (using 1/64 cores)
WARNING: Unable to create progress bar. Consider installing `tqdm` as `pip
install tqdm` and `ipywidgets` as `pip install ipywidgets`,
or disable the progress bar using `show_progress_bar=False`.
    finished (0:01:57) --> added
    'velocity_graph', sparse matrix with cosine correlations (adata.uns)

# check output ---
adata
AnnData object with n_obs × n_vars = 11000 × 2000
    obs: 'cell', 'clusters', 'initial_size_spliced',
'initial_size_unspliced', 'initial_size', 'n_counts',
'velocity_self_transition'
    var: 'spliced', 'gene_name', 'gene_biotype', 'means', 'dispersions',
'dispersions_norm', 'highly_variable', 'velocity_gamma',
'velocity_qreg_ratio', 'velocity_r2', 'velocity_genes'
    uns: 'X_name', 'pca', 'neighbors', 'umap', 'velocity_params',
'velocity_graph', 'velocity_graph_neg'
    obsm: 'X_pca', 'X_umap'
    varm: 'PCs'
    layers: 'spliced', 'unspliced', 'Ms', 'Mu', 'velocity',
'variance_velocity'
    obsp: 'distances', 'connectivities'
```

- Run velocity_cellgroup by R

```
file <-
'/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.cell_group.p
ng'
scv$pl$velocity_embedding_stream(adata, basis='X_umap',
size=as.integer(120), alpha=0.4, title='A549 cell line',
```

```
legend_fontweight='heavy', legend_fontsize=20, figsize=c(10, 7),
show=FALSE, save=file)
computing velocity embedding
    finished (0:00:03) --> added
    'velocity_umap', embedded velocity vectors (adata.obsm)
saving figure to file
/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.cell_group.pn
g
```

- Run velocity_pseudotime by R

```
scv$tl$velocity_pseudotime(adata)
computing terminal states
    identified 1 region of root cells and 1 region of end points .
    finished (0:00:02) --> added
    'root_cells', root cells of Markov diffusion process (adata.obs)
    'end_points', end points of Markov diffusion process (adata.obs)

file <-
'/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.velocity_pse
udotime.png'
scv$pl$velocity_embedding_stream(adata, basis='X_umap',
color='velocity_pseudotime', color_map='Spectral_r', size=as.integer(120),
alpha=0.4, rescale_color=c(0, 1), legend_fontweight='heavy',
legend_fontsize=as.integer(20), figsize=c(10, 7), show=FALSE, save=file)
saving figure to file
/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.velocity_pseu
dotime.png
```

- Output as h5ad format file

```
file <-
'/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R/A549.scvelo.h5ad'
adata$write(file, compression='gzip')
```

- Final outputs

```
(scvelo) [tuchiyama@davinci scVelo_by_R]$ pwd
/wgbs_global1/davinci/tuchiyama/A549/output/scVelo_by_R

(scvelo) [tuchiyama@davinci scVelo_by_R]$ ls -lh .
total 319M
-rw-rw-r-- 1 tuchiyama tuchiyama 1015K Sep 25 18:32 A549.cell_group.png
-rw-rw-r-- 1 tuchiyama tuchiyama   20K Sep 25 17:43 A549.proportions.png
-rw-rw-r-- 1 tuchiyama tuchiyama  315M Sep 25 19:05 A549.scvelo.h5ad
-rw-rw-r-- 1 tuchiyama tuchiyama  3.0M Sep 25 18:51
A549.velocity_pseudotime.png
```