# script to perform trajectory analysis

# https://www.nature.com/articles/s41467-019-10291-0

# setwd("~/Desktop/demo/monocle3")

set.seed(1234)

library(monocle3)

library(SeuratWrappers)

library(Seurat)

library(ggplot2)

library(tidyverse)

# read in data

markers <- read.delim('ABC\_Marker.txt', header = T) # gene metadata

metadata <- read.delim('ABC\_Meta.txt', header = T) # cell metadata

expr <- read.delim('ABC\_umi\_matrix\_7551\_cells.csv', header = T, sep = ',') # expression matrix

# create seurat object ---------------

expr.t <- t(expr)

seu.obj <- CreateSeuratObject(counts = expr.t)

View(seu.obj@meta.data)

seu.obj@meta.data <- merge(seu.obj@meta.data, metadata, by.x = 'row.names', by.y = 'cell\_id')

View(seu.obj@meta.data)

seu.obj@meta.data <- seu.obj@meta.data %>%

column\_to\_rownames(var = 'Row.names')

seu.obj$mitopercent <- PercentageFeatureSet(seu.obj, pattern = '^MT-')

seu.obj.filtered <- subset(seu.obj, subset = nCount\_RNA > 800 &

nFeature\_RNA > 500 &

mitopercent < 10)

# subset my seurat object - B cells

unique(seu.obj.filtered@meta.data$population)

Idents(seu.obj.filtered) <- seu.obj.filtered$population

b.seu <- subset(seu.obj.filtered, idents = "b")

b.seu

unique(b.seu@meta.data$redefined\_cluster)

# pre-processing using seurat

b.seu <- NormalizeData(b.seu)

b.seu <- FindVariableFeatures(b.seu)

b.seu <- ScaleData(b.seu)

b.seu <- RunPCA(b.seu)

b.seu <- FindNeighbors(b.seu, dims = 1:30)

b.seu <- FindClusters(b.seu, resolution = 0.9)

b.seu <- RunUMAP(b.seu, dims = 1:30, n.neighbors = 50)

a1 <- DimPlot(b.seu, reduction = 'umap', group.by = 'redefined\_cluster', label = T)

a2 <- DimPlot(b.seu, reduction = 'umap', group.by = 'seurat\_clusters', label = T)

a1|a2

# MONOCLE3 WORKFLOW ---------------------

# monocle3 requires cell\_data\_set object

# convert seurat object to cell\_data\_set object for monocle3

# ...1 Convert to cell\_data\_set object ------------------------

cds <- as.cell\_data\_set(b.seu)

cds

# to get cell metadata

colData(cds)

# to gene metdata

fData(cds)

rownames(fData(cds))[1:10]

# since it misses the gene\_short\_name column, let's add it

fData(cds)$gene\_short\_name <- rownames(fData(cds))

# to get counts

counts(cds)

# ...2. Cluster cells (using clustering info from seurat's UMAP)---------------------------

# let's use the clustering information have

# assign paritions

reacreate.partition <- c(rep(1,length(cds@colData@rownames)))

names(reacreate.partition) <- cds@colData@rownames

reacreate.partition <- as.factor(reacreate.partition)

cds@clusters$UMAP$partitions <- reacreate.partition

# Assign the cluster info

list\_cluster <- b.seu@active.ident

cds@clusters$UMAP$clusters <- list\_cluster

# Assign UMAP coordinate - cell embeddings

cds@int\_colData@listData$reducedDims$UMAP <- b.seu@reductions$umap@cell.embeddings

# plot

cluster.before.trajectory <- plot\_cells(cds,

color\_cells\_by = 'cluster',

label\_groups\_by\_cluster = FALSE,

group\_label\_size = 5) +

theme(legend.position = "right")

cluster.names <- plot\_cells(cds,

color\_cells\_by = "redefined\_cluster",

label\_groups\_by\_cluster = FALSE,

group\_label\_size = 5) +

scale\_color\_manual(values = c('red', 'blue', 'green', 'maroon', 'yellow', 'grey', 'cyan')) +

theme(legend.position = "right")

cluster.before.trajectory | cluster.names

# ...3. Learn trajectory graph ------------------------

cds <- learn\_graph(cds, use\_partition = FALSE)

plot\_cells(cds,

color\_cells\_by = 'redefined\_cluster',

label\_groups\_by\_cluster = FALSE,

label\_branch\_points = FALSE,

label\_roots = FALSE,

label\_leaves = FALSE,

group\_label\_size = 5)

# ...4. Order the cells in pseudotime -------------------

cds <- order\_cells(cds, reduction\_method = 'UMAP', root\_cells = colnames(cds[,clusters(cds) == 5]))

plot\_cells(cds,

color\_cells\_by = 'pseudotime',

label\_groups\_by\_cluster = FALSE,

label\_branch\_points = FALSE,

label\_roots = FALSE,

label\_leaves = FALSE)

# cells ordered by monocle3 pseudotime

pseudotime(cds)

cds$monocle3\_pseudotime <- pseudotime(cds)

data.pseudo <- as.data.frame(colData(cds))

ggplot(data.pseudo, aes(monocle3\_pseudotime, reorder(redefined\_cluster, monocle3\_pseudotime, median), fill = redefined\_cluster)) +

geom\_boxplot()

# ...5. Finding genes that change as a function of pseudotime --------------------

deg\_bcells <- graph\_test(cds, neighbor\_graph = 'principal\_graph', cores = 4)

deg\_bcells %>%

arrange(q\_value) %>%

filter(status == 'OK') %>%

head()

FeaturePlot(b.seu, features = c('E2F2', 'STMN1', 'CD52'))

# visualizing pseudotime in seurat

b.seu$pseudotime <- pseudotime(cds)

Idents(b.seu) <- b.seu$redefined\_cluster

FeaturePlot(b.seu, features = "pseudotime", label = T)