~~~
---
title: "pseudotime_stromal cells_PDR"
author: "Ben Thomson, Katia Corano Scheri"
date: "8/4/2022"
output:
  ###pdf_document: default
  html_document: default
  html_notebook: default
---

```{r setup, include=FALSE}

#knitr::opts_chunk$set(echo = TRUE, cache = TRUE, fig.width = 10, fig.height = 6, warning
= FALSE, message = FALSE)

#if (!requireNamespace("BiocManager", quietly = TRUE))
#install.packages("BiocManager")
#BiocManager::install(version = "3.14")

#BiocManager::install(c('BiocGenerics', 'DelayedArray', 'DelayedMatrixStats',
#                        'limma', 'lme4', 'S4Vectors', 'SingleCellExperiment',
#                        'SummarizedExperiment', 'batchelor', 'Matrix.utils',
#                        'HDF5Array', 'terra', 'ggrastr'))

#install.packages("devtools")
#devtools::install_github('cole-trapnell-lab/monocle3')

#brew install pkg config
#brew install gdal

#install.packages("sf", configure.args = "--with-proj-lib=/usr/local/lib/")

#Load libraries
  library(Seurat)
  library(SeuratWrappers)
  library(monocle3)
  library(BiocGenerics)
  library(dplyr)
  library(gdata)
  library(sctransform)
  library(cowplot)
  library(ggplot2)
  library(gridExtra)
  library(data.table)
theme_set(theme_cowplot())


```
~~~

Open stroma dataset and convert to monocle object

~~~
```{r}
  data.seurat <- readRDS("./stroma.reclustered.rds")
  data.seurat[["celltype"]] <- Idents(object = data.seurat)

```

```{r convertToMonocle}


  ##data.seurat <- FindVariableFeatures(data.seurat)
~~~

~~~
  #Extract data, phenotype data, and feature data from the SeuratObject
  ##data <- GetAssayData(object = data.seurat, slot = "counts", assay = "RNA")

  ##pd <- data.frame(data.seurat@meta.data)

  ##fData <- data.frame(gene_short_name = row.names(data), row.names = row.names(data))
  ##fd <- new('AnnotatedDataFrame', data = fData)

  #Construct monocle cds
  ##data.monocle <- new_cell_data_set(data, gene_metadata = fData, cell_metadata = pd)


##data.monocle <- preprocess_cds(data.monocle, num_dim = 50)
##data.monocle <- align_cds(data.monocle, alignment_group = "orig.ident")

##data.monocle <- reduce_dimension(data.monocle)

##data.monocle <- cluster_cells(data.monocle)



# a helper function to identify the root principal points:
get_earliest_principal_node <- function(cds, time_bin="130-170"){
  cell_ids <- which(colData(cds)[, "celltype"] == time_bin)

closest_vertex <- cds@principal_graph_aux[["UMAP"]]$pr_graph_cell_proj_closest_vertex
closest_vertex <- as.matrix(closest_vertex[colnames(cds), ])
root_pr_nodes <- igraph::V(principal_graph(cds)
[["UMAP"]])$name[as.numeric(names(which.max(table(closest_vertex[cell_ids,]))))]

root_pr_nodes
}


## Step 5: Learn a graph
data.monocle <- as.cell_data_set(data.seurat)
data.monocle <- cluster_cells(data.monocle)

data.monocle <- learn_graph(data.monocle)

## Step 6: Order cells -- change time_bin to preferred root node!!
data.monocle <- order_cells(data.monocle,
root_pr_nodes=get_earliest_principal_node(data.monocle, time_bin = "Stroma_1"))

plot_cells(data.monocle, color_cells_by = "pseudotime", graph_label_size = 4,
group_label_size = 4)
plot_cells(data.monocle, color_cells_by = "celltype", label_groups_by_cluster = T,
graph_label_size = 4, group_label_size = 4)


```



```{r, fig.height=10, include=FALSE}



##test.res <- graph_test(data.monocle, neighbor_graph = "principal_graph", cores = 16)
##test.res <- graph_test(data.monocle)

```

```{r, fig.height=10}
##pr_deg_ids <- row.names(subset(test.res, q_value < 0.05))
~~~

~~~
##gene_module_df <- find_gene_modules(data.monocle[pr_deg_ids,],
resolution=c(10^seq(-6,-1)))

##cell_group_df <- tibble::tibble(cell=row.names(colData(data.monocle)),
##                                cell_group=colData(data.monocle)$celltype)
##agg_mat <- aggregate_gene_expression(data.monocle, gene_module_df, cell_group_df)
##row.names(agg_mat) <- stringr::str_c("Module ", row.names(agg_mat))
##pheatmap::pheatmap(agg_mat,
##                   scale="column", clustering_method="ward.D2")

```


```{r}
sessionInfo()
```
~~~