####### 7. GEO single cell Analogue time series analysis #########

rm(list = ls())

load("scRNA\_anno.Rda")

head(scRNA)

###Seurat object to monocle object

# Expression Matrix

expression\_matrix <- scRNA@assays$RNA@counts

expression\_matrix[1:3, 1:3]

dim(expression\_matrix)

# Cell Annotation Information

cell\_metadata <- scRNA@meta.data

dim(cell\_metadata)

# Gene annotation files

gene\_metadata <- data.frame(gene\_id = rownames(expression\_matrix),

gene\_short\_name = rownames(expression\_matrix))

rownames(gene\_metadata) <- gene\_metadata$gene\_short\_name

# Building cds objects

cds <- new\_cell\_data\_set(expression\_matrix,

cell\_metadata = cell\_metadata,

gene\_metadata = gene\_metadata)

####Data preprocessing##

# Data standardisation

# preprocess\_cdsThe data is first normalised by taking the log and then PCA downscaling is performed

cds <- preprocess\_cds(cds,

method = c("PCA"), # ATAC data selection using the LSI method

norm\_method = c("log"),

num\_dim = 100)

save(cds, file = "cds.log.Rda")

load(file = "cds.log.Rda")

# Visualisation of Principal Component Analysis

p1 <- plot\_pc\_variance\_explained(cds)

p1

# dimensionality reduction analysis

cds <- reduce\_dimension(cds,

reduction\_method = c("UMAP"), # "tSNE", "PCA"

preprocess\_method = c(c("PCA")))

colData(cds)

p2 <- plot\_cells(cds,

x = 1,

y = 2,

reduction\_method = c("UMAP"),

color\_cells\_by = "singleR\_type",# Must be the column name of the colData data in the cds object

group\_label\_size = 5,

labels\_per\_group = 2,

graph\_label\_size = 2,

show\_trajectory\_graph = TRUE,

label\_cell\_groups = TRUE,

label\_groups\_by\_cluster = TRUE,

)

p2

plot\_cells(cds,

genes = c("CDKN2A", "GPX3", "SLC7A11"))

plot\_cells(cds,

genes = c("CDKN2A"),

group\_label\_size = 4,

graph\_label\_size = 4,

cell\_size = 1.5,

alpha = 1)

plot\_cells(cds,

genes = c("GPX3"),

group\_label\_size = 4,

graph\_label\_size = 4,

cell\_size = 1.5,

alpha = 1)

plot\_cells(cds,

genes = c("SLC7A11"),

group\_label\_size = 4,

graph\_label\_size = 4,

cell\_size = 1.5,

alpha = 1)

plot\_cells(cds,

genes = c("H1F0"),

group\_label\_size = 4,

graph\_label\_size = 4,

cell\_size = 1.5,

alpha = 1)

# # Checking for batch effects

# p3 <- plot\_cells(cds,

# color\_cells\_by = "Group",

# label\_cell\_groups = TRUE)

# p3

# # The batch effect was significant, so the batch effect was removed

# cds <- align\_cds(cds,

# num\_dim = 100,

# alignment\_group = c("Group"))

# cds <- reduce\_dimension(cds)

# p4 <- plot\_cells(cds,

# color\_cells\_by = "group",

# label\_cell\_groups = FALSE)

# p3 + p4

#

# # Explore possible batches

# head(colData(cds))

# plot\_cells(cds,

# color\_cells\_by = "Phase",

# label\_cell\_groups = FALSE)

# # Inability to remove batch-to-batch differences again, possibly due to biological effects or other reasons

####cell clustering###

cds <- cluster\_cells(cds,

reduction\_method = "UMAP",

cluster\_method = "leiden",

k = 130, # The smaller K is, the more clusters there are.

random\_seed = 123456)

plot\_cells(cds,

group\_label\_size = 5)

cds\_cluster <- cds@clusters@listData[["UMAP"]][["clusters"]]

head(cds@colData)

colData(cds)$cds\_cluster <- as.character(clusters(cds))

# Finding marker genes

marker\_test\_res <- top\_markers(cds,

group\_cells\_by = "cds\_cluster",

reference\_cells = 1000,

cores = 8)

head(marker\_test\_res)

# fraction\_expressing Expression Score

plot\_cells(cds,

genes = c("CDKN2A", "GPX3", "SLC7A11")

)

plot\_cells(cds,

genes = c("CDKN2A"))

plot\_cells(cds,

genes = c("GPX3"))

plot\_cells(cds,

genes = c("SLC7A11"))

plot\_cells(cds,

genes = c("H1F0"))

top\_specific\_markers <- marker\_test\_res %>%

filter(marker\_test\_q\_value < 0.01 &

specificity >= 0.2) %>%

group\_by(cell\_group) %>%

top\_n(5, pseudo\_R2)

# Extract marker name

top\_specific\_markers1 <- top\_specific\_markers %>%

group\_by(gene\_short\_name) %>%

filter(n() == 1)

top\_specific\_marker\_ids <- top\_specific\_markers1 %>%

pull(gene\_id) # pull = $notation

# visualisation

plot\_genes\_by\_group(cds,

top\_specific\_marker\_ids,

group\_cells\_by = "cds\_cluster",

ordering\_type = "maximal\_on\_diag",

max.size = 3)

# Generate markers file

generate\_garnett\_marker\_file(top\_specific\_markers1, file="./marker\_file.txt")

marker\_file <- read\_tsv("marker\_file.txt")

# Perform cellular annotation

# Find literature for manual annotation

colData(cds)$assigned\_cell\_type <- as.character(partitions(cds))

colData(cds)$assigned\_cell\_type <- dplyr::recode(colData(cds)$cds\_cluster,

"1" = "Epithelial cells",

"2" = "iPS cells",

"3" = "Fibroblasts",

"4" = "B cells",

"5" = "T cells",

"6" = "Marcrophage",

"7" = "Endothelial cells"

)

plot\_cells(cds,

group\_cells\_by = "cluster",

color\_cells\_by = "assigned\_cell\_type",

group\_label\_size = 6,

labels\_per\_group = 2,

graph\_label\_size = 2,

cell\_size = 1,

alpha = 0.5)

# Gene of Interest Expression Map

ciliated\_genes <- c("CDKN2A", "GPX3", "SLC7A11")

plot\_cells(cds,

genes = ciliated\_genes,

label\_cell\_groups = FALSE,

show\_trajectory\_graph = FALSE,

cell\_size = 1,

alpha = 1)

# Monocle3 clustering partition

# Distinguishing between different trajectories

cds <- cluster\_cells(cds)

p1 <- plot\_cells(cds,

show\_trajectory\_graph = FALSE) +

ggtitle("label by clusterID")

p1

p2 <- plot\_cells(cds,

color\_cells\_by = "partition",

show\_trajectory\_graph = FALSE,

group\_label\_size = 4,

labels\_per\_group = 1,

graph\_label\_size = 2,) +

ggtitle("label by partitionID")

p2

# Instead of the previous steps, build the cds object directly and then run the following

# Recognition of trajectories

cds <- learn\_graph(cds)

p3 <- plot\_cells(cds,

label\_groups\_by\_cluster = F,

label\_leaves = F,

label\_branch\_points = F,

cell\_size = 1,

alpha = 1)

p3

# Cells are sorted in chronological order

# Officially a function is given, here a time\_bin is defined which selects the earliest time point interval.

# a helper function to identify the root principal points:

## The following set of codes can also produce diagrams

# get\_earliest\_principal\_node <- function(cds, time\_bin = "Macrophage"){

# cell\_ids <- which(colData(cds)[, "singleR\_type"] == 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

# }

# cds <- order\_cells(cds, root\_pr\_nodes=get\_earliest\_principal\_node(cds))

#

# plot\_cells(cds,

# color\_cells\_by = "pseudotime",

# label\_cell\_groups = FALSE,

# label\_leaves = FALSE,

# label\_branch\_points = FALSE,

# graph\_label\_size = 1.5,

# group\_label\_size = 4,

# cell\_size = 1.5)

## This study uses the following set of code to produce a diagram. Note: Run this step directly and a dialogue box will appear, mark the places in the diagram where the story takes place yourself, and then produce the diagrams

cds <- order\_cells(cds,

reduction\_method = "UMAP",

root\_pr\_nodes = NULL,

root\_cells = NULL,

verbose = FALSE)

plot\_cells(cds,

color\_cells\_by = "pseudotime",

label\_cell\_groups = F,

label\_leaves = F,

label\_branch\_points = F,

graph\_label\_size = 5,

group\_label\_size = 4,

cell\_size = 1.5)

# Finding differential genes (analysed using only top\_specific\_marker\_ids)

ciliated\_genes <- top\_specific\_marker\_ids

cds\_subset <- cds[rowData(cds)$gene\_short\_name %in% ciliated\_genes,]

# If all genes are used for analysis, run the following sentence directly

gene\_fits <- fit\_models(cds\_subset, model\_formula\_str = "~singleR\_type")

# coefficient\_table()The function extracts the results from the model

fit\_coefs <- coefficient\_table(gene\_fits)

head(fit\_coefs)

table(fit\_coefs$term)

# Selection of differential genes

emb\_time\_terms <- fit\_coefs %>%

filter(term == "singleR\_typeEndothelial\_cells") %>%

# filter (q\_value < 0.05) %>%

select(gene\_short\_name, term, q\_value, estimate)

genes <- head(emb\_time\_terms)$gene\_short\_name

plot\_genes\_violin(cds\_subset[genes,],

group\_cells\_by = "singleR\_type",

ncol = 2) +

theme(axis.text.x = element\_text(angle=45,

hjust=1))

# View Citation

get\_citations(cds)