Project\_Shatakshi Shewale

2025-03-03

######## Part 1 #############  
  
setwd("/Users/shatakshishewale/Desktop/Applications/interviews/Final project") # Manually set the working directory  
  
# Verify it has been set correctly  
getwd()

## [1] "/Users/shatakshishewale/Desktop/Applications/interviews/Final project"

set.seed(2024) # Set seed  
  
# Load the datasets  
bm1 <- readRDS("BoneMarrow\_dataset1.rds")  
bm2 <- readRDS("BoneMarrow\_dataset2.rds")  
  
dim(bm1)

## Loading required package: Matrix

## NULL

dim(bm2)

## [1] 26832 2870

#summary(bm1)  
#summary(bm2)

set.seed(2024) # Set seed  
  
library(Seurat)

## Warning: package 'Seurat' was built under R version 4.3.3

## Loading required package: SeuratObject

## Warning: package 'SeuratObject' was built under R version 4.3.3

## Loading required package: sp

## Warning: package 'sp' was built under R version 4.3.3

##   
## Attaching package: 'SeuratObject'

## The following objects are masked from 'package:base':  
##   
## intersect, t

bm1 <- CreateSeuratObject(  
 counts = bm1,   
 project = "BoneMarrow1",  
 min.cells = 3, # Removes genes detected in fewer than 3 cells  
 min.features = 200 # Cells with at least 200 genes detected  
)  
  
bm2 <- CreateSeuratObject(  
 counts = bm2,   
 project = "BoneMarrow2",  
 min.cells = 3,  
 min.features = 200  
)  
  
# Check Seurat object structure  
bm1

## An object of class Seurat   
## 17725 features across 2791 samples within 1 assay   
## Active assay: RNA (17725 features, 0 variable features)  
## 1 layer present: counts

bm2

## An object of class Seurat   
## 17985 features across 2870 samples within 1 assay   
## Active assay: RNA (17985 features, 0 variable features)  
## 1 layer present: counts

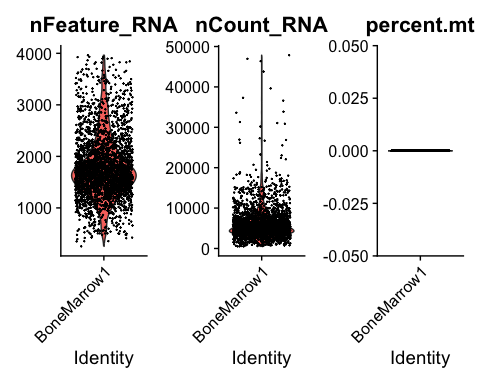
set.seed(2024) # Set seed  
  
# Add mitochondrial gene percentage  
bm1[["percent.mt"]] <- PercentageFeatureSet(bm1, pattern = "^MT-")  
bm2[["percent.mt"]] <- PercentageFeatureSet(bm2, pattern = "^MT-")  
  
# Visualize QC metrics  
VlnPlot(bm1, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

## Warning: Default search for "data" layer in "RNA" assay yielded no results;  
## utilizing "counts" layer instead.

## Warning: The `slot` argument of `FetchData()` is deprecated as of SeuratObject 5.0.0.  
## ℹ Please use the `layer` argument instead.  
## ℹ The deprecated feature was likely used in the Seurat package.  
## Please report the issue at <https://github.com/satijalab/seurat/issues>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

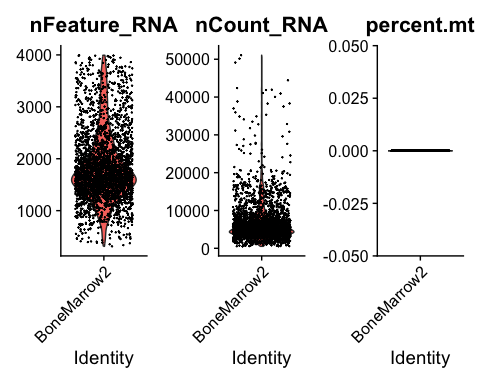
## Warning: `PackageCheck()` was deprecated in SeuratObject 5.0.0.  
## ℹ Please use `rlang::check\_installed()` instead.  
## ℹ The deprecated feature was likely used in the Seurat package.  
## Please report the issue at <https://github.com/satijalab/seurat/issues>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning in SingleExIPlot(type = type, data = data[, x, drop = FALSE], idents =  
## idents, : All cells have the same value of percent.mt.



# Visualize QC metrics  
VlnPlot(bm2, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

## Warning: Default search for "data" layer in "RNA" assay yielded no results;  
## utilizing "counts" layer instead.  
## Warning: All cells have the same value of percent.mt.



# Filter cells (adjust thresholds based on QC)  
bm1 <- subset(bm1, subset = nFeature\_RNA > 200 & nFeature\_RNA < 4000 & nCount\_RNA < 30000)  
bm2 <- subset(bm2, subset = nFeature\_RNA > 200 & nFeature\_RNA < 4000 & nCount\_RNA < 30000)  
  
# Check dimensions after filtering  
dim(bm1)

## [1] 17725 2778

dim(bm2)

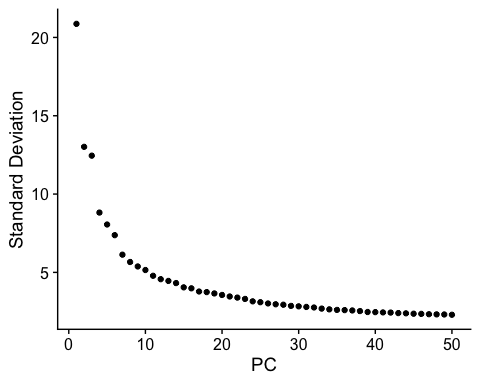
## [1] 17985 2844

set.seed(2024) # Set seed  
  
# run sctransform  
bm1 <- SCTransform(bm1, vars.to.regress = "nCount\_RNA", verbose = FALSE)

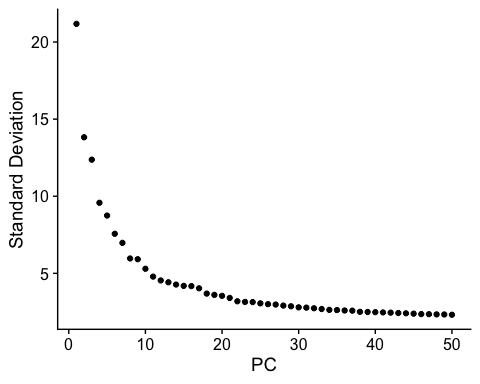
## Warning: The `slot` argument of `SetAssayData()` is deprecated as of SeuratObject 5.0.0.  
## ℹ Please use the `layer` argument instead.  
## ℹ The deprecated feature was likely used in the Seurat package.  
## Please report the issue at <https://github.com/satijalab/seurat/issues>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

## Warning: The `slot` argument of `GetAssayData()` is deprecated as of SeuratObject 5.0.0.  
## ℹ Please use the `layer` argument instead.  
## ℹ The deprecated feature was likely used in the Seurat package.  
## Please report the issue at <https://github.com/satijalab/seurat/issues>.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_lifecycle\_warnings()` to see where this warning was  
## generated.

bm2 <- SCTransform(bm2, vars.to.regress = "nCount\_RNA", verbose = FALSE)  
  
# run PCA  
bm1 <- RunPCA(bm1, verbose = FALSE)  
bm2 <- RunPCA(bm2, verbose = FALSE)  
  
ElbowPlot(bm1, ndims = 50)



ElbowPlot(bm2, ndims = 50)



# run UMAP  
bm1 <- RunUMAP(bm1, dims = 1:30, verbose = FALSE)

## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R-native UWOT using the cosine metric  
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'  
## This message will be shown once per session

bm2 <- RunUMAP(bm2, dims = 1:30, verbose = FALSE)

set.seed(2024) # Set seed  
  
bm1 <- FindNeighbors(bm1, dims = 1:30)

## Computing nearest neighbor graph

## Warning: package 'future' was built under R version 4.3.3

## Computing SNN

bm2 <- FindNeighbors(bm2, dims = 1:30)

## Computing nearest neighbor graph

## Computing SNN

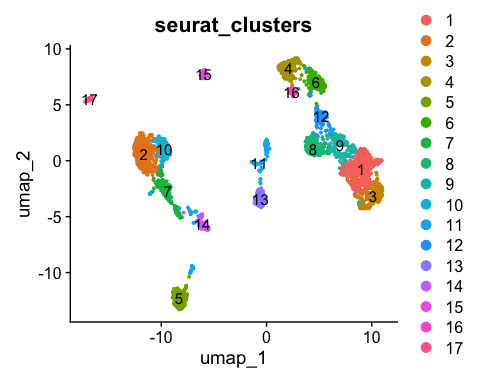
bm1 <- FindClusters(bm1, resolution = 1.5, algorithm = 4)

## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : `random.seed` must be greater than 0 for  
## leiden clustering, resetting `random.seed` to 1.

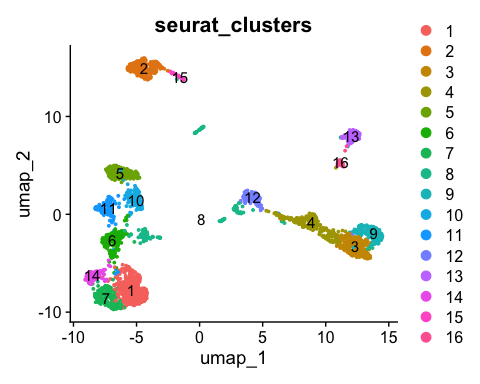
bm2 <- FindClusters(bm2, resolution = 1.5, algorithm = 4)

## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : `random.seed` must be greater than 0 for  
## leiden clustering, resetting `random.seed` to 1.

DimPlot(bm1, reduction = "umap", group.by = "seurat\_clusters", label = TRUE)



DimPlot(bm2, reduction = "umap", group.by = "seurat\_clusters", label = TRUE)



# Set seed for reproducibility  
set.seed(2024)  
  
# Load necessary libraries  
library(Seurat)  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.3.3

# Define marker genes with their corresponding Ensembl IDs  
cell\_type\_markers <- unique(c(  
 "ENSG00000170458", # CD14 (Classical monocyte cell & Macrophage cell)  
 "ENSG00000168329", # CX3CR1 (Non-classical monocyte cell)  
 "ENSG00000198178", "ENSG00000099250", # CLEC4C, NRP1 (Plasmacytoid dendritic cell)  
 "ENSG00000114013", # CD86 (Dendritic cell)  
 "ENSG00000100721", "ENSG00000128218", # TCL1A, VPREB3 (Precursor B cell)  
 "ENSG00000153064", # BANK1 (Memory B cell)  
 "ENSG00000105369", # CD79A (Naive B cell)  
 "ENSG00000137441", # FGFBP2 (CD16 natural killer cell)  
 "ENSG00000168685", # IL7R (Natural killer cell)  
 "ENSG00000137801", # THBS1 (Megakaryocyte)  
 "ENSG00000145708", "ENSG00000170891", "ENSG00000128040", # CRHBP, CYTL1, SPINK2 (Myeloid progenitor)  
 "ENSG00000174059", # CD34 (Hematopoietic stem cell)  
 "ENSG00000172116", "ENSG00000256039", "ENSG00000142546", # CD8B, LINC02446, NOSIP (CD8 T cell)  
 "ENSG00000163600", # ICOS (Regulatory T cell)  
 "ENSG00000113088" # GZMK (Mucosal-associated invariant T cell)  
))  
  
# List of datasets  
datasets <- list("bm1" = bm1, "bm2" = bm2)  
  
# Generate and save combined dot plots for each dataset  
for (dataset\_name in names(datasets)) {  
 dataset <- datasets[[dataset\_name]]  
   
 # Generate single dot plot for all marker genes  
 plot <- DotPlot(dataset, features = cell\_type\_markers, group.by = "seurat\_clusters") +  
 ggtitle(paste("Combined Dot Plot -", dataset\_name)) +  
 RotatedAxis()  
   
 # Save the plot as a single image file  
 ggsave(filename = paste0(dataset\_name, "\_combined\_dotplot.png"),  
 plot = plot, width = 10, height = 8, dpi = 300)  
}

# Set seed for reproducibility  
set.seed(2024)   
  
# Load necessary libraries  
library(Seurat)  
library(ggplot2)  
  
# Define marker genes with their corresponding Ensembl IDs (No duplicates)  
cell\_type\_markers <- unique(c(  
 "ENSG00000170458", # CD14 (Classical monocyte cell & Macrophage cell)  
 "ENSG00000168329", # CX3CR1 (Non-classical monocyte cell)  
 "ENSG00000198178", "ENSG00000099250", # CLEC4C, NRP1 (Plasmacytoid dendritic cell)  
 "ENSG00000114013", # CD86 (Dendritic cell)  
 "ENSG00000100721", "ENSG00000128218", # TCL1A, VPREB3 (Precursor B cell)  
 "ENSG00000153064", # BANK1 (Memory B cell)  
 "ENSG00000105369", # CD79A (Naive B cell)  
 "ENSG00000137441", # FGFBP2 (CD16 natural killer cell)  
 "ENSG00000137801", # THBS1 (Megakaryocyte)  
 "ENSG00000145708", "ENSG00000170891", "ENSG00000128040", # CRHBP, CYTL1, SPINK2 (Myeloid progenitor)  
 "ENSG00000135218", # CD36 (Erythroid progenitor)  
 "ENSG00000174059", # CD34 (Hematopoietic stem cell)  
 "ENSG00000172116", "ENSG00000256039", "ENSG00000142546", # CD8B, LINC02446, NOSIP (CD8 T cell)  
 "ENSG00000163600", # ICOS (Regulatory T cell)  
 "ENSG00000113088", "ENSG00000111796" # GZMK , KLRB1 (Mucosal-associated invariant T cell)  
))  
  
# Subset clusters  
bm1\_clusters <- c(1, 4, 9, 12,10,2,13,15,17,11,5,14)  
bm2\_clusters <- c(3, 9,12,13,16,15,2,5,1,14,6)  
  
# Subset the data for the selected clusters  
bm1\_subset <- subset(bm1, idents = bm1\_clusters)  
bm2\_subset <- subset(bm2, idents = bm2\_clusters)  
  
# Re-cluster using FindClusters after subsetting  
bm1\_subset <- FindClusters(bm1\_subset, resolution = 1.0, algorithm = 4)

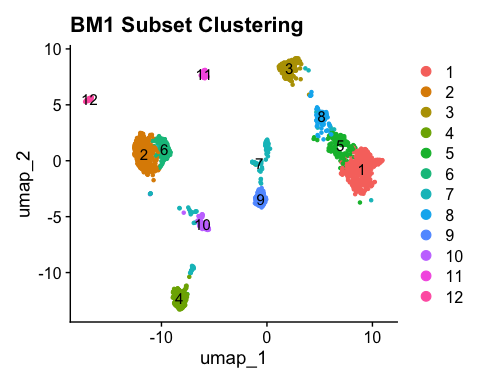
## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : `random.seed` must be greater than 0 for  
## leiden clustering, resetting `random.seed` to 1.

## Warning: Adding a command log without an assay associated with it

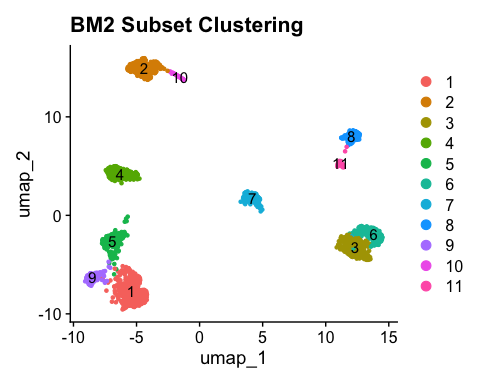
bm2\_subset <- FindClusters(bm2\_subset, resolution = 1.0, algorithm = 4)

## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : `random.seed` must be greater than 0 for  
## leiden clustering, resetting `random.seed` to 1.  
## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : Adding a command log without an assay  
## associated with it

# Plot UMAP for verification  
DimPlot(bm1\_subset, reduction = "umap", label = TRUE) + ggtitle("BM1 Subset Clustering")



DimPlot(bm2\_subset, reduction = "umap", label = TRUE) + ggtitle("BM2 Subset Clustering")



# Generate and save dot plots for bm1 and bm2  
datasets <- list("bm1\_subset" = bm1\_subset, "bm2\_subset" = bm2\_subset)  
  
for (dataset\_name in names(datasets)) {  
 dataset <- datasets[[dataset\_name]]  
   
 # Generate dot plot  
 plot <- DotPlot(dataset, features = cell\_type\_markers, group.by = "seurat\_clusters") +   
 ggtitle(paste("DotPlot for", dataset\_name)) +   
 RotatedAxis()  
   
 # Save the plot as 'split\_dotplot.png'  
 ggsave(filename = paste0(dataset\_name, "\_split\_dotplot.png"),  
 plot = plot, width = 10, height = 6, dpi = 300)  
}

set.seed(2024) # Set seed  
  
# Define the final cell type annotations based on the table  
bm1\_cell\_types <- list(  
 "6" = "Classical monocyte cell",  
 "9" = "Plasmacytoid dendritic cell",  
 "12" = "Dendritic cell",  
 "11" = "Dendritic cell",  
 "4" = "Naive B cell",  
 "3" = "CD16 natural killer cell",  
 "2" = "Megakaryocyte",  
 "10" = "Myeloid progenitor",  
 "1" = "Regulatory T cell",  
 "5" = "Mucosal-associated invariant T cell"  
)  
  
bm2\_cell\_types <- list(  
 "6" = "Classical monocyte cell",  
 "8" = "Plasmacytoid dendritic cell",  
 "11" = "Dendritic cell",  
 "10" = "Precursor B cell",  
 "2" = "Naive B cell",  
 "4" = "CD16 natural killer cell",  
 "3" = "Megakaryocyte",  
 "7" = "Myeloid progenitor",  
 "9" = "CD8 T cell",  
 "5" = "Mucosal-associated invariant T cell"  
)  
  
# Annotate the cell types in bm1\_subset  
bm1\_subset$cell\_type <- NA # Set NA for all cells initially  
for (cluster in names(bm1\_cell\_types)) {  
 bm1\_subset$cell\_type[bm1\_subset$seurat\_clusters == as.numeric(cluster)] <- bm1\_cell\_types[[cluster]]  
}  
  
# Annotate the cell types in bm2\_subset  
bm2\_subset$cell\_type <- NA # Set NA for all cells initially  
for (cluster in names(bm2\_cell\_types)) {  
 bm2\_subset$cell\_type[bm2\_subset$seurat\_clusters == as.numeric(cluster)] <- bm2\_cell\_types[[cluster]]  
}  
  
# Confirm the cell type annotation  
table(bm1\_subset$cell\_type)

##   
## CD16 natural killer cell Classical monocyte cell   
## 199 142   
## Dendritic cell Megakaryocyte   
## 81 326   
## Mucosal-associated invariant T cell Myeloid progenitor   
## 153 87   
## Naive B cell Plasmacytoid dendritic cell   
## 177 120   
## Regulatory T cell   
## 460

table(bm2\_subset$cell\_type)

##   
## CD16 natural killer cell CD8 T cell   
## 207 92   
## Classical monocyte cell Dendritic cell   
## 163 43   
## Megakaryocyte Mucosal-associated invariant T cell   
## 260 201   
## Myeloid progenitor Naive B cell   
## 123 262   
## Plasmacytoid dendritic cell Precursor B cell   
## 121 44

# Add NA to the full dataset for cells that are not in the subsets  
bm1$cell\_type <- NA  
bm2$cell\_type <- NA  
  
# Transfer the annotated labels from the subset to the full dataset  
bm1$cell\_type[Cells(bm1\_subset)] <- bm1\_subset$cell\_type  
bm2$cell\_type[Cells(bm2\_subset)] <- bm2\_subset$cell\_type  
  
# Confirm the cell type annotation  
table(bm1$cell\_type)

##   
## CD16 natural killer cell Classical monocyte cell   
## 199 142   
## Dendritic cell Megakaryocyte   
## 81 326   
## Mucosal-associated invariant T cell Myeloid progenitor   
## 153 87   
## Naive B cell Plasmacytoid dendritic cell   
## 177 120   
## Regulatory T cell   
## 460

table(bm2$cell\_type)

##   
## CD16 natural killer cell CD8 T cell   
## 207 92   
## Classical monocyte cell Dendritic cell   
## 163 43   
## Megakaryocyte Mucosal-associated invariant T cell   
## 260 201   
## Myeloid progenitor Naive B cell   
## 123 262   
## Plasmacytoid dendritic cell Precursor B cell   
## 121 44

# Save the annotated Seurat objects  
saveRDS(bm1, file = "bm1\_annotated.rds")  
saveRDS(bm2, file = "bm2\_annotated.rds")

set.seed(2024) # Set seed

############## Part 2 #####################  
  
set.seed(2024) # Set seed  
  
setwd("/Users/shatakshishewale/Desktop/Applications/interviews/Final project") # Manually set the working directory  
  
# Verify it has been set correctly  
getwd()

## [1] "/Users/shatakshishewale/Desktop/Applications/interviews/Final project"

# Load the dataset  
pancreas <- readRDS("Pancreas.rds")  
  
# Check the structure of the object  
#print(pancreas)  
dim(pancreas)

## [1] 26099 1610

#summary(pancreas)

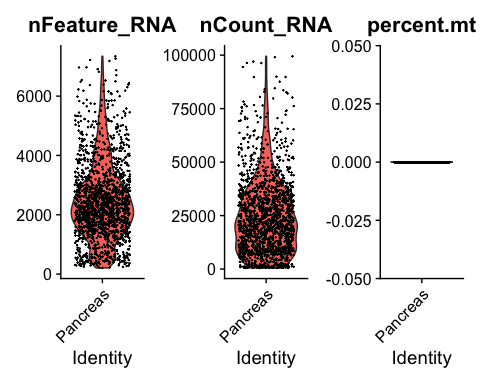
set.seed(2024) # Set seed  
  
  
library(Seurat)  
  
pancreas <- CreateSeuratObject(  
 counts = pancreas,   
 project = "Pancreas",  
 min.cells = 3, # Removes genes detected in fewer than 3 cells  
 min.features = 200 # Cells with at least 200 genes detected  
)  
  
# Check Seurat object structure  
pancreas

## An object of class Seurat   
## 17597 features across 1610 samples within 1 assay   
## Active assay: RNA (17597 features, 0 variable features)  
## 1 layer present: counts

set.seed(2024) # Set seed  
# Add mitochondrial gene percentage  
pancreas[["percent.mt"]] <- PercentageFeatureSet(pancreas, pattern = "^MT-")  
  
  
# Visualize QC metrics  
VlnPlot(pancreas, features = c("nFeature\_RNA", "nCount\_RNA", "percent.mt"), ncol = 3)

## Warning: Default search for "data" layer in "RNA" assay yielded no results;  
## utilizing "counts" layer instead.

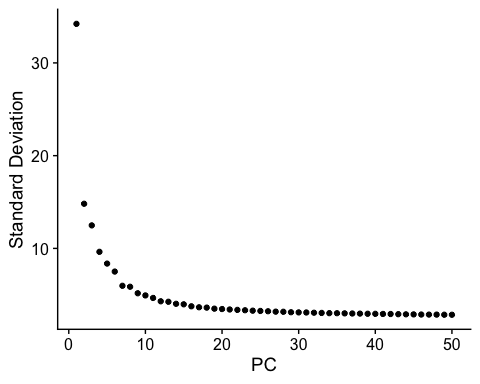
## Warning in SingleExIPlot(type = type, data = data[, x, drop = FALSE], idents =  
## idents, : All cells have the same value of percent.mt.



# Filter cells (adjust thresholds based on QC)  
pancreas <- subset(pancreas, subset = nFeature\_RNA > 200 & nFeature\_RNA < 6500 & nCount\_RNA < 100000)  
  
# Check dimensions after filtering  
dim(pancreas)

## [1] 17597 1597

set.seed(2024) # Set seed  
# run sctransform  
pancreas <- SCTransform(pancreas,vars.to.regress = "nCount\_RNA", verbose = FALSE)  
  
# run PCA  
pancreas <- RunPCA(pancreas, verbose = FALSE)  
  
ElbowPlot(pancreas, ndims = 50)



# run UMAP  
pancreas <- RunUMAP(pancreas, dims = 1:20, verbose = FALSE)

set.seed(2024) # Set seed  
  
pancreas <- FindNeighbors(pancreas, dims = 1:25)

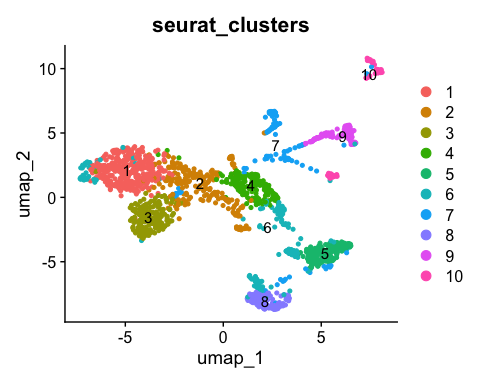
## Computing nearest neighbor graph

## Computing SNN

pancreas <- FindClusters(pancreas, resolution = 0.8, algorithm = 4)

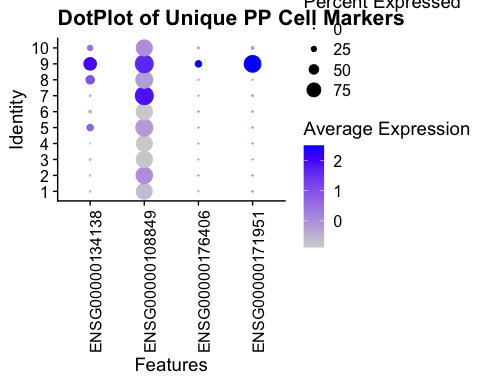
## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : `random.seed` must be greater than 0 for  
## leiden clustering, resetting `random.seed` to 1.

DimPlot(pancreas, reduction = "umap", group.by = "seurat\_clusters", label = TRUE)

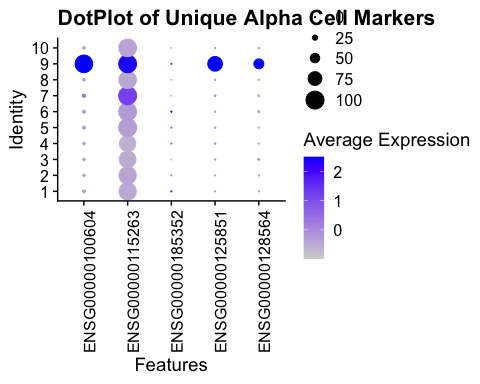


set.seed(2024) # Set seed  
  
library(ggplot2)  
  
# Define marker genes with Ensembl IDs for PP and Alpha cells  
pp\_markers <- c("ENSG00000089199", "ENSG00000109472", "ENSG00000134138", "ENSG00000157168",  
 "ENSG00000102109", "ENSG00000108849", "ENSG00000176406", "ENSG00000171951",  
 "ENSG00000166922", "ENSG00000118271")  
  
alpha\_markers <- c("ENSG00000100604", "ENSG00000089199", "ENSG00000109472", "ENSG00000115263",  
 "ENSG00000185352", "ENSG00000102109", "ENSG00000125851", "ENSG00000166922",  
 "ENSG00000118271", "ENSG00000128564")  
  
# Identify unique markers for each cell type  
unique\_pp\_markers <- setdiff(pp\_markers, alpha\_markers)  
unique\_alpha\_markers <- setdiff(alpha\_markers, pp\_markers)  
  
  
# DotPlot for PP cell markers (unique)  
DotPlot(pancreas, features = unique\_pp\_markers, group.by = "seurat\_clusters") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 ggtitle("DotPlot of Unique PP Cell Markers")

## Warning: The following requested variables were not found: ENSG00000157168



# DotPlot for Alpha cell markers (unique)  
DotPlot(pancreas, features = unique\_alpha\_markers, group.by = "seurat\_clusters") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 ggtitle("DotPlot of Unique Alpha Cell Markers")

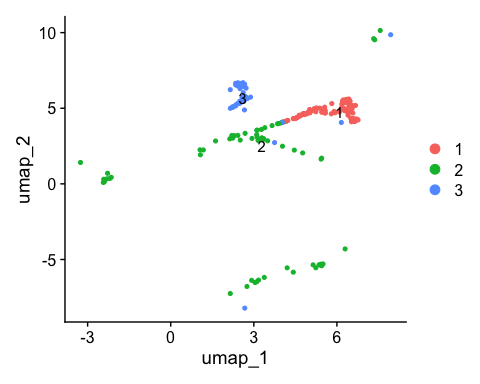


set.seed(2024) # Set seed  
  
# subset cluster 7 and 9 and then re-cluster  
  
pancreas\_subset <- subset(pancreas, idents = c(9, 7))  
pancreas\_subset <- FindClusters(pancreas\_subset, resolution = 0.5, algorithm = 4)

## Warning in RunLeiden(object = object, method = method, partition.type =  
## "RBConfigurationVertexPartition", : `random.seed` must be greater than 0 for  
## leiden clustering, resetting `random.seed` to 1.

## Warning: Adding a command log without an assay associated with it

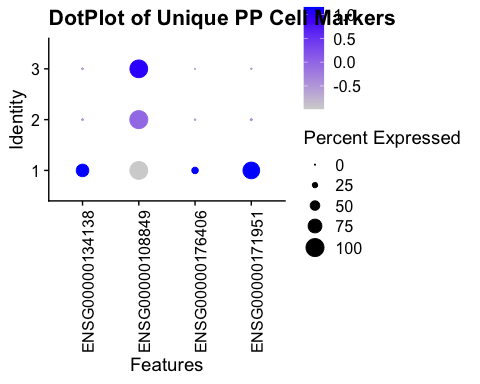
DimPlot(pancreas\_subset, reduction = "umap", label = TRUE)



set.seed(2024) # Set seed  
  
# DotPlot for PP cell markers (unique)  
DotPlot(pancreas\_subset, features = unique\_pp\_markers, group.by = "seurat\_clusters") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 ggtitle("DotPlot of Unique PP Cell Markers")

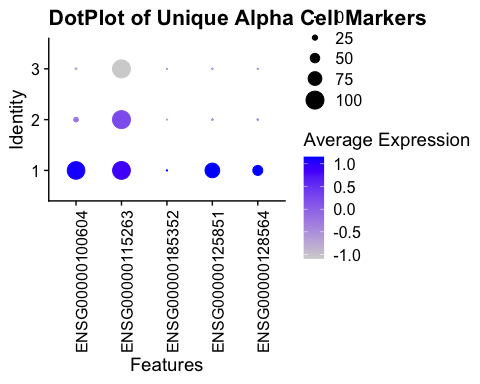
## Warning: The following requested variables were not found: ENSG00000157168

## Warning: Scaling data with a low number of groups may produce misleading  
## results



# DotPlot for Alpha cell markers (unique)  
DotPlot(pancreas\_subset, features = unique\_alpha\_markers, group.by = "seurat\_clusters") +  
 theme(axis.text.x = element\_text(angle = 90, hjust = 1)) +  
 ggtitle("DotPlot of Unique Alpha Cell Markers")

## Warning: Scaling data with a low number of groups may produce misleading  
## results



set.seed(2024) # Set seed  
  
# Initialize cell\_type column in pancreas object  
pancreas$cell\_type <- NA   
  
# Assign the same annotations from pancreas\_subset to the full dataset  
pancreas$cell\_type[pancreas$seurat\_clusters == 3] <- "PP"  
pancreas$cell\_type[pancreas$seurat\_clusters == 1] <- "alpha"  
  
# Check the distribution of annotated cell types  
table(pancreas$cell\_type)

##   
## alpha PP   
## 339 190

# Save the updated Seurat object for submission  
saveRDS(pancreas, file = "Pancreas\_Annotated\_Full.rds")

set.seed(2024) # Set seed  
  
#Q5  
# Identify differentially expressed genes for cluster 1 (alpha cells)  
alpha\_DEGs <- FindMarkers(pancreas\_subset, ident.1 = 1, logfc.threshold = 0.25, min.pct = 0.1)

## For a (much!) faster implementation of the Wilcoxon Rank Sum Test,  
## (default method for FindMarkers) please install the presto package  
## --------------------------------------------  
## install.packages('devtools')  
## devtools::install\_github('immunogenomics/presto')  
## --------------------------------------------  
## After installation of presto, Seurat will automatically use the more   
## efficient implementation (no further action necessary).  
## This message will be shown once per session

alpha\_DEGs <- alpha\_DEGs[order(alpha\_DEGs$avg\_log2FC, decreasing = TRUE), ] # Sort by log fold change  
  
# View the top markers  
head(alpha\_DEGs, 10)

## p\_val avg\_log2FC pct.1 pct.2 p\_val\_adj  
## ENSG00000136750 9.626837e-34 8.593402 0.828 0.000 1.559836e-29  
## ENSG00000007372 1.186518e-29 7.640536 0.747 0.000 1.922516e-25  
## ENSG00000108309 6.441037e-27 7.169464 0.690 0.000 1.043641e-22  
## ENSG00000152254 8.090872e-20 7.024322 0.552 0.008 1.310964e-15  
## ENSG00000010282 1.375415e-19 6.954196 0.529 0.000 2.228584e-15  
## ENSG00000117472 1.355596e-19 6.937518 0.529 0.000 2.196473e-15  
## ENSG00000153132 4.099901e-19 6.886292 0.517 0.000 6.643069e-15  
## ENSG00000128656 1.093474e-21 6.640536 0.575 0.000 1.771756e-17  
## ENSG00000166963 1.032611e-17 6.577342 0.483 0.000 1.673140e-13  
## ENSG00000187634 3.469973e-18 6.533621 0.494 0.000 5.622398e-14

# Identify differentially expressed genes for cluster 3 (PP cells)  
pp\_DEGs <- FindMarkers(pancreas\_subset, ident.1 = 3, logfc.threshold = 0.25, min.pct = 0.1)  
pp\_DEGs <- pp\_DEGs[order(pp\_DEGs$avg\_log2FC, decreasing = TRUE), ] # Sort by log fold change  
  
# View the top markers  
head(pp\_DEGs, 10)

## p\_val avg\_log2FC pct.1 pct.2 p\_val\_adj  
## ENSG00000245532 3.921122e-24 5.138909 1.000 0.796 6.353395e-20  
## ENSG00000106258 1.460864e-04 4.989775 0.136 0.013 1.000000e+00  
## ENSG00000065361 7.632650e-06 4.628319 0.203 0.026 1.236718e-01  
## ENSG00000142102 1.394730e-04 4.409679 0.153 0.020 1.000000e+00  
## ENSG00000147862 4.136615e-05 3.950247 0.169 0.020 6.702558e-01  
## ENSG00000162069 1.227496e-04 3.950247 0.169 0.026 1.000000e+00  
## ENSG00000152669 5.204793e-04 3.950247 0.102 0.007 1.000000e+00  
## ENSG00000113916 5.953014e-04 3.687213 0.119 0.013 1.000000e+00  
## ENSG00000176058 1.554201e-02 3.613212 0.136 0.046 1.000000e+00  
## ENSG00000120158 8.120470e-03 3.557930 0.136 0.039 1.000000e+00

set.seed(2024) # Set seed