STAT 530 Homework 14

Shuyu Jia (shuyuj2)

Due by midnight on 5/4/2021

Instructions

Homeworks must be submitted as PDFs. **You must complete this homework using an R Notebook.**

Introduction

This and the following homeworks ask you to analyze single-cell RNA-seq data from Pandey et al. (2018). The article is available from the course website and the data can be downloaded from https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE109158.

Download and uncompress `GSE109158_RAW.tar`. We will be analyzing larval data (`GSM2818521_larva_counts_matrix.txt.g` and one of the adult data (`GSM2818522_adultr1_counts_matrix.txt.gz`) runs.

In this homework you will analyze Pandey et al. (2018) data using the R package `Seurat`, currently one of the most widely-used scRNA-seq analysis pipelines. The purpose of this homework is for you to learn how to learn how to use `Seurat` on your own. Follow the tutorial on the `Seurat` website. In practice, learning on your own is the main way you will learn bioinformatics. **Therefore, for this homework neither Robin nor I will be answering any questions about coding. We will only answer clarification or conceptual questions.** If you have questions about coding, ask the internet or each other.

The purpose of this entire class has been to give you the foundations you need to learn bioinformatics on your own. To illustrate how much you have learned, in this homework you will also be analyzing these data again using the `signac` functions you have written. You will find that over the past few homeworks, you have implemented much of the core `Seurat` functionality from scratch. Recognizing the parallels between your `signac` functions and the `Seurat` functions will hopefully demystify bioinformatics in R.

A final note: don't expect `Seurat` and `signac` to give you the same results, because though they conceptually perform nearly the same analyses, the underlying implementations are still different.

Load software

Download the file `signac.R` and save it to the same working directory as your Homework 14 `.Rmd` file. This file contains all of the functions we have written so far to analyze scRNA-seq data. Load these functions using the following command. You can think of this as loading our own custom library.

```
source("signac.R")
```

Now load the `Seurat` package.

```
library(Seurat)
```

In past homeworks I added `rm()` and `gc()` commands to help you with RAM concerns. In this homework I will not write them, but please add them if you find your computer running out of memory.

Problem 1: Load data and quality control (12 points)

Load the Pandey et al. (2018) feature-barcode matrices using the following code.

```
library(data.table)

## load data
larval = fread("GSM2818521_larva_counts_matrix.txt.gz")
adult = fread("GSM2818522_adultr1_counts_matrix.txt.gz")

## convert to count matrices
larval_counts = as.matrix(larval[, -1])
rownames(larval_counts) = larval[, V1]
colnames(larval_counts) = colnames(larval)[-1]
adult_counts = as.matrix(adult[, -1])
rownames(adult_counts) = adult[, V1]
colnames(adult_counts) = colnames(adult)[-1]

rm(list = c("larval", "adult"))
```

a. Create `signac` objects using the the larval and adult count matrices. Remove genes detected in less than 30 cells and remove cells that contain less than 500 detected genes. (2 points)

```
## create signac object
larval_signac = create_object(counts = larval_counts, min.cells = 30, min.features = 500)
adult_signac = create_object(counts = adult_counts, min.cells = 30, min.features = 500)
```

b. Create `Seurat` objects using the the larval and adult count matrices. Remove genes detected in less than 30 cells and remove cells that contain less than 500 detected genes. (2 points)

```
larval_seurat = CreateSeuratObject(counts = larval_counts, min.cells = 30, min.features = 500)
adult_seurat = CreateSeuratObject(counts = adult_counts, min.cells = 30, min.features = 500)

rm(list = c("larval_counts", "adult_counts"))
```

c. Using the `signac` objects, calculate the percentage of reads that map to the mitochondrial genome in the larval and adult datasets and add the percentage as a column in the meta data. (2 points)

```
## filter by mitochondrial content
larval_mt_genes = grep("^MT-", rownames(larval_signac$assay_data))
adult_mt_genes = grep("^MT-", rownames(adult_signac$assay_data))

larval_signac$meta_data[["percent.mt"]]=percentage_feature_set(larval_signac, features=larval_mt_genes)
adult_signac$meta_data[["percent.mt"]]=percentage_feature_set(adult_signac, features=adult_mt_genes)

rm(list = c("larval_mt_genes", "adult_mt_genes"))
```

d. Using the Seurat objects, calculate the percentage of reads that map to the mitochondrial genome in the larval and adult datasets and add the percentage as a column in the meta data. (2 points)

```
larval_seurat[["percent.mt"]] = PercentageFeatureSet(larval_seurat, pattern = "^MT-")
adult_seurat[["percent.mt"]] = PercentageFeatureSet(adult_seurat, pattern = "^MT-")
```

e. Using the signac objects, filter cells that have > 6% mitochondrial counts in the larval and adult datasets. (2 points)

```
larval_signac=subset_obj(larval_signac, subset=which(larval_signac$meta_data[["percent.mt"]]<=0.06))
adult_signac=subset_obj(adult_signac, subset=which(adult_signac$meta_data[["percent.mt"]]<=0.06))
```

f. Using the Seurat objects, filter cells that have > 6% mitochondrial counts in the larval and adult datasets. (2 points)

```
larval_seurat = subset(larval_seurat, subset = percent.mt < 6)
adult_seurat = subset(adult_seurat, subset = percent.mt < 6)

gc()
```

```
##            used  (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells  2650813 141.6    4694678  250.8   4694678  250.8
## Vcells 72097313 550.1  274820815 2096.8 343526018 2620.9
```

Problem 2: Normalize data (8 points)

a. Normalize the signac objects to be log-counts per 10,000 (plus 1). (2 points)

```
## normalize data
larval_signac = normalize_data(larval_signac)
adult_signac = normalize_data(adult_signac)
```

b. Normalize the Seurat objects to be log-counts per 10,000 (plus 1). (2 points)

```
larval_seurat = NormalizeData(larval_seurat)
adult_seurat = NormalizeData(adult_seurat)
```

c. In the signac objects, remove unwanted variation as measured by the percent of mitochondrial reads. (2 points)

```
## remove unwanted sources of variation
larval_signac = regress_out(larval_signac, c("percent.mt"))
adult_signac = regress_out(adult_signac, c("percent.mt"))
```

d. In the Seurat objects, remove unwanted variation as measured by the percent of mitochondrial reads. In the Seurat tutorial, this step occurs after finding the most variable genes; come to office hours or set up an appointment with me if you want to know why. In this homework, we will instead remove variation before finding the most variable genes. (2 points)

```r
larval_seurat = ScaleData(larval_seurat, vars.to.regress = "percent.mt")
adult_seurat = ScaleData(adult_seurat, vars.to.regress = "percent.mt")

gc()
```

```
##              used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells   2668991  142.6    4694678  250.8   4694678  250.8
## Vcells 185561724 1415.8  329864978 2516.7 343526018 2620.9
```

Problem 3: Dimension reduction (8 points)

a. Identify the 2000 most variable genes in the `signac` objects. (2 points)

```r
larval_signac = find_variable_features(larval_signac)
adult_signac = find_variable_features(adult_signac)
```

b. Identify the 2000 most variable genes in the `Seurat` objects. (2 points)

```r
larval_seurat = FindVariableFeatures(larval_seurat, selection.method = "vst", nfeatures = 2000)
adult_seurat = FindVariableFeatures(adult_seurat, selection.method = "vst", nfeatures = 2000)
```

c. Using the `signac` objects, perform dimension reduction using PCA; calculate 20 principal components. (2 points)

```r
larval_signac = run_pca(larval_signac, npcs = 20)
adult_signac = run_pca(adult_signac, npcs = 20)
```

d. Using the `Seurat` objects, perform dimension reduction using PCA; calculate 20 principal components. (2 points)

```r
larval_seurat=RunPCA(larval_seurat, features=VariableFeatures(object=larval_seurat), npcs=20)
adult_seurat=RunPCA(adult_seurat, features=VariableFeatures(object=adult_seurat), npcs=20)
```

Problem 4: Clustering and visualization (12 points)

a. In the `signac` objects, find clusters using a 20 nearest neighbors and a shared nearest neighbor threshold of 3. (2 points)

```r
larval_signac = find_clusters(larval_signac, k = 20, kt = 3)
adult_signac = find_clusters(adult_signac, k = 20, kt = 3)
```

b. In the `Seurat` objects, find clusters using the first 20 principal components to find neighbors and a resolution of 0.5 to find clusters. (2 points)

```
larval_seurat = FindNeighbors(larval_seurat, dims = 1:20)
adult_seurat = FindNeighbors(adult_seurat, dims = 1:20)

larval_seurat = FindClusters(larval_seurat, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 4343
## Number of edges: 162885
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9176
## Number of communities: 15
## Elapsed time: 0 seconds
```

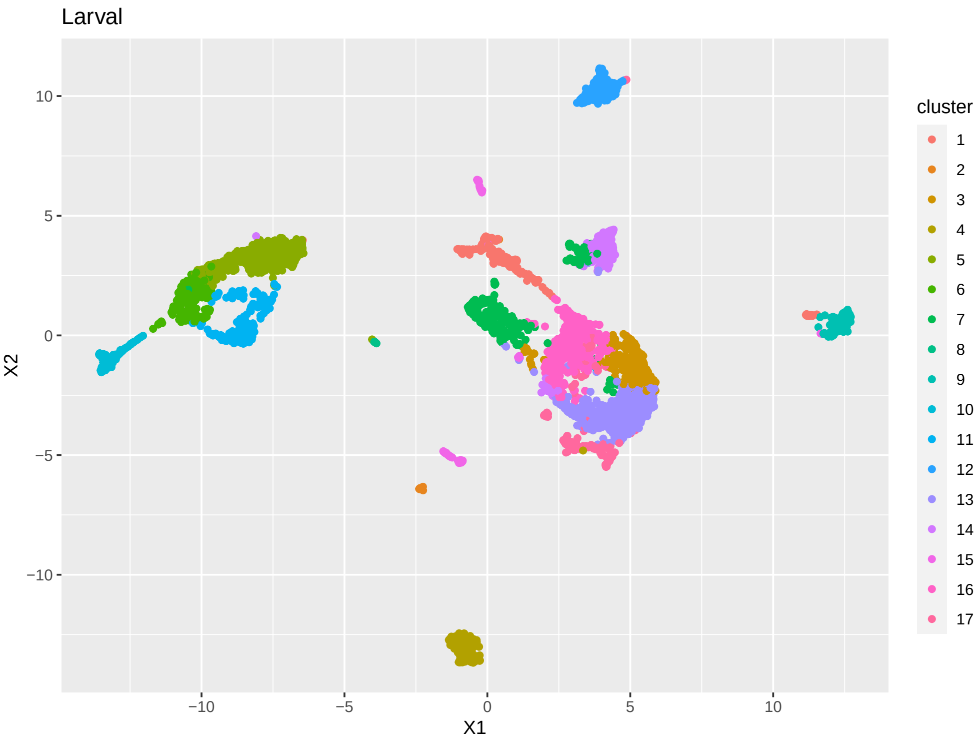
```
adult_seurat = FindClusters(adult_seurat, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 3048
## Number of edges: 129490
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8629
## Number of communities: 10
## Elapsed time: 0 seconds
```

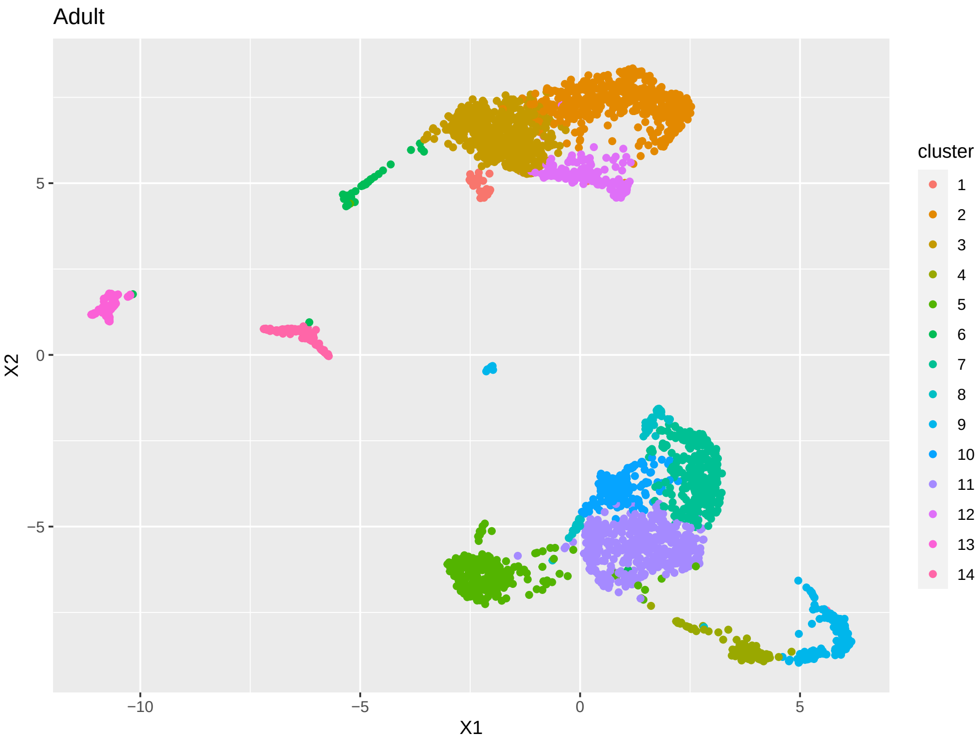
c. In the `signac` objects, run UMAP, visualize the cells using a scatterplot in the UMAP dimensions, and color the cells by cluster. (2 points)

```
## run UMAP and visualize
larval_signac = run_umap(larval_signac)
adult_signac = run_umap(adult_signac)

## visualize larval
df = data.frame(larval_signac$reductions$UMAP,
                cluster = larval_signac$meta_data$cluster)
ggplot(data = df) +
  geom_point(mapping = aes(x = X1, y = X2, color = cluster)) +
  ggtitle("Larval")
```



```r
## visualize adult
df = data.frame(adult_signac$reductions$UMAP,
                cluster = adult_signac$meta_data$cluster)
ggplot(data = df) +
  geom_point(mapping = aes(x = X1, y = X2, color = cluster)) +
  ggtitle("Adult")
```

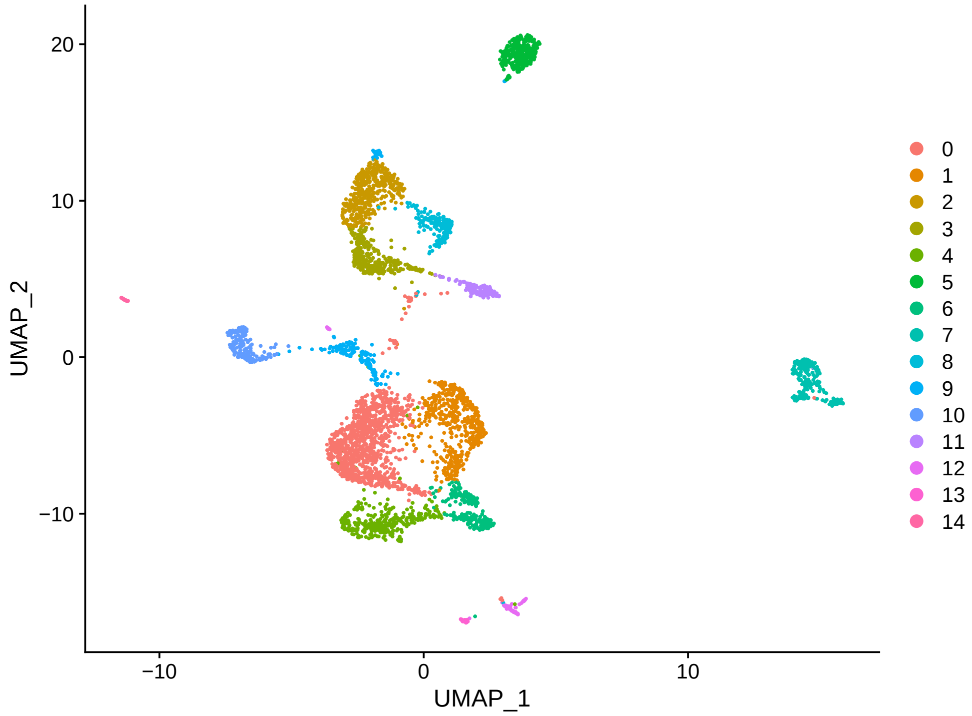


```
rm(list = c("df"))
```

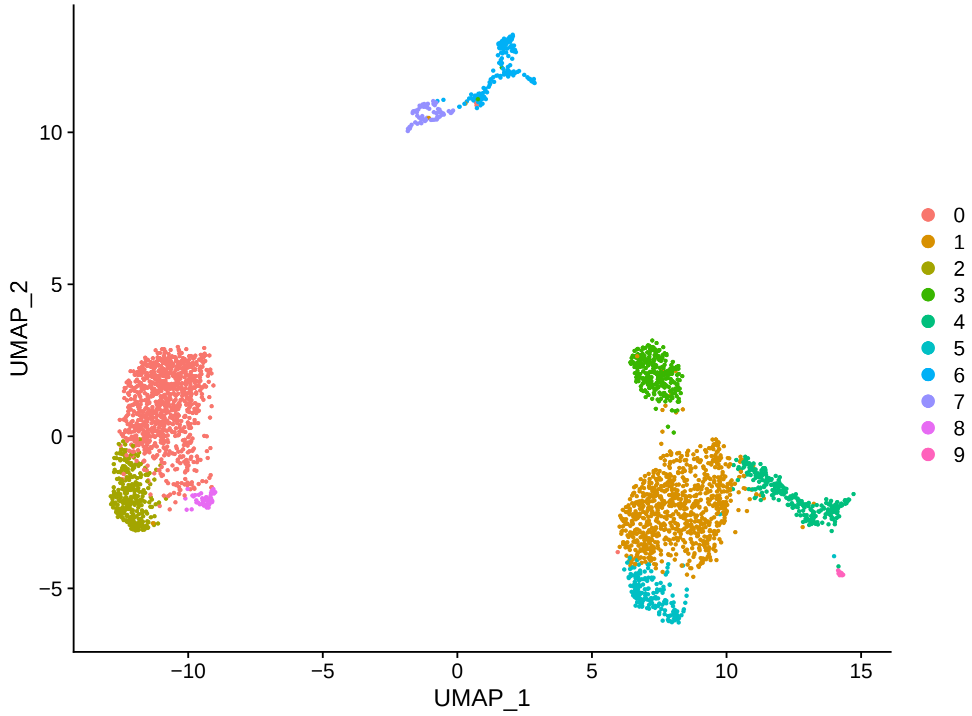
d. In the `Seurat` objects, run UMAP using the first 20 principal components, visualize the cells using a scatterplot in the UMAP dimensions, and color the cells by cluster. (2 points)

```
larval_seurat = RunUMAP(larval_seurat, dims = 1:20)
adult_seurat = RunUMAP(adult_seurat, dims = 1:20)

DimPlot(larval_seurat, reduction = "umap")
```



```r
DimPlot(adult_seurat, reduction = "umap")
```



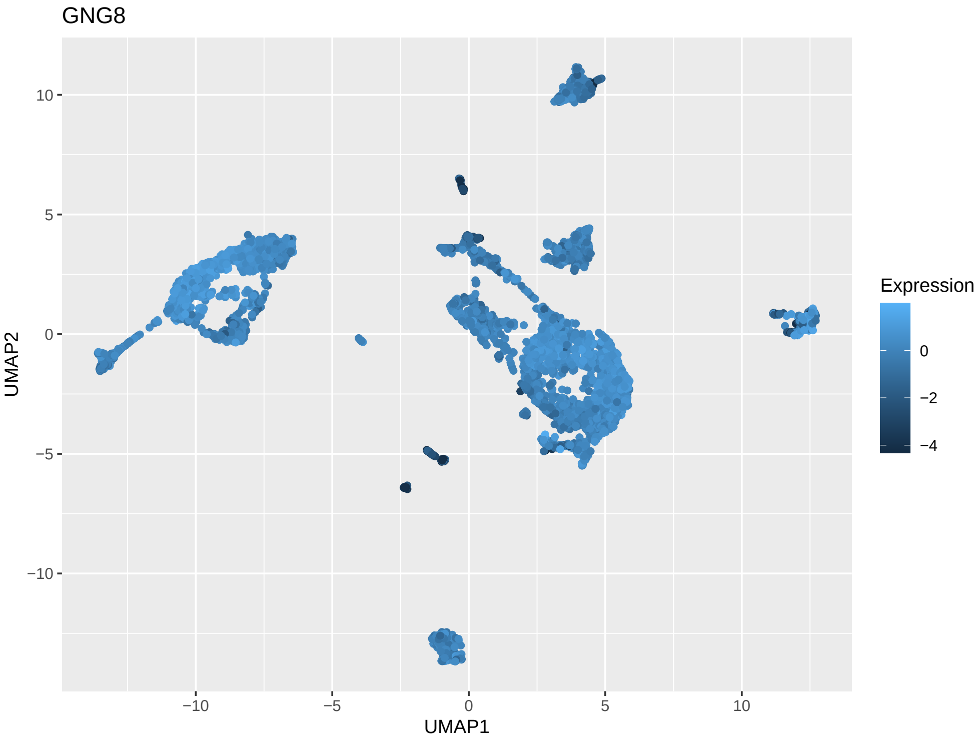
```
gc()
```

```
##              used   (Mb) gc trigger   (Mb)  max used   (Mb)
## Ncells    3007312  160.7    4694678  250.8   4694678  250.8
## Vcells 188308915 1436.7  329864978 2516.7 343526018 2620.9
```

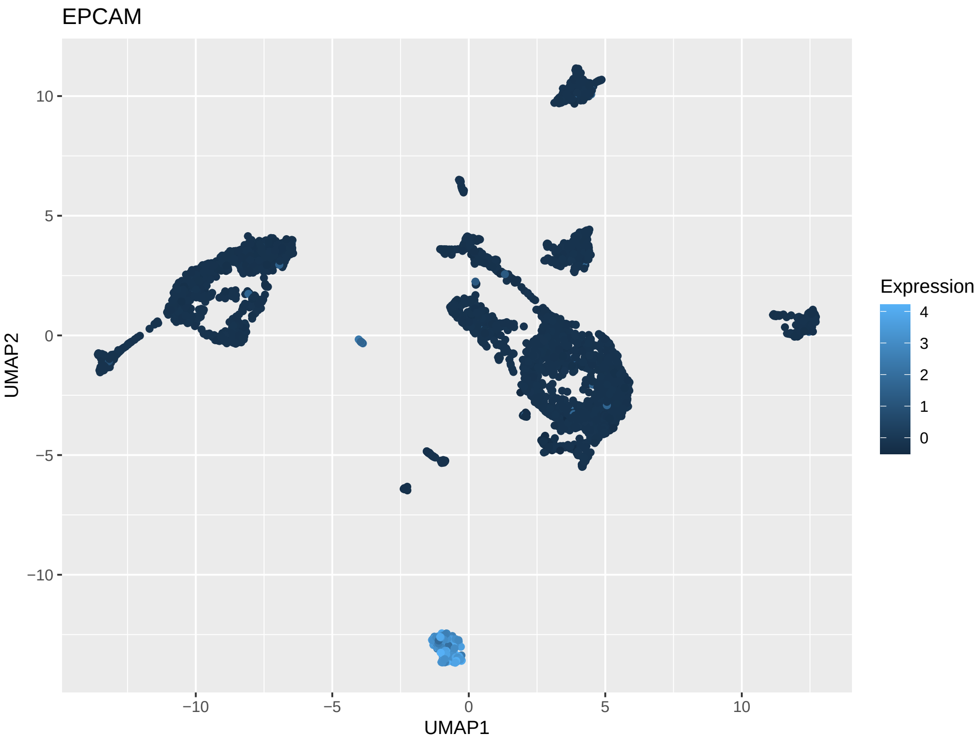
e. Using the `signac` results, visualize the genes `GNG8` and `EPCAM`. (2 points)

```
features = c("GNG8", "EPCAM")
feature_plot(larval_signac, features)
```

```
## [[1]]
```

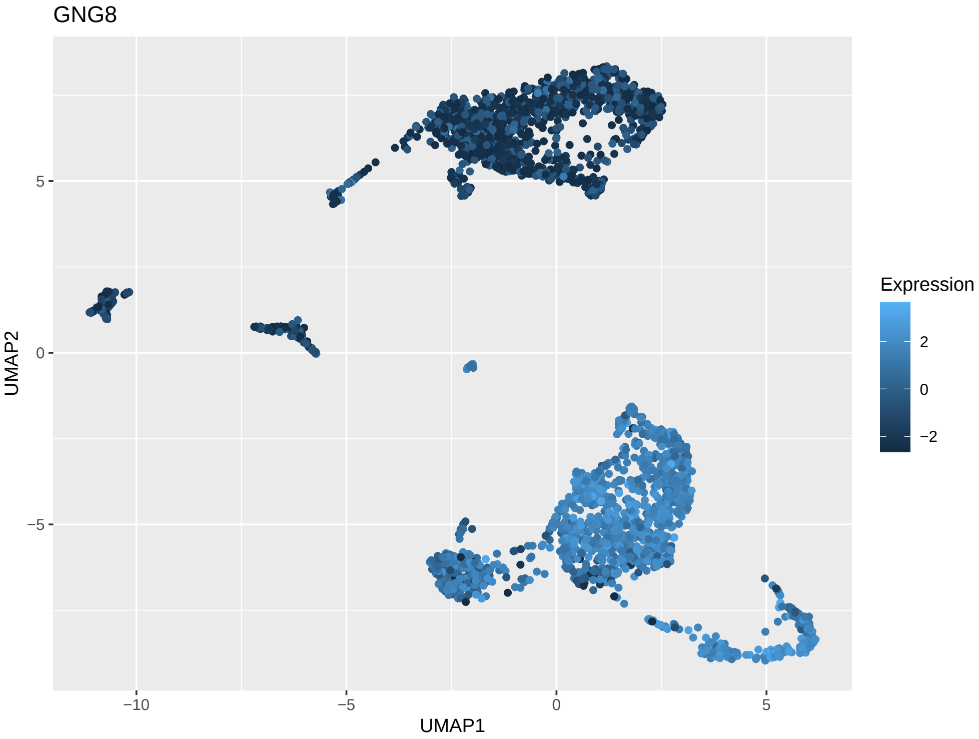


```
##
## [[2]]
```



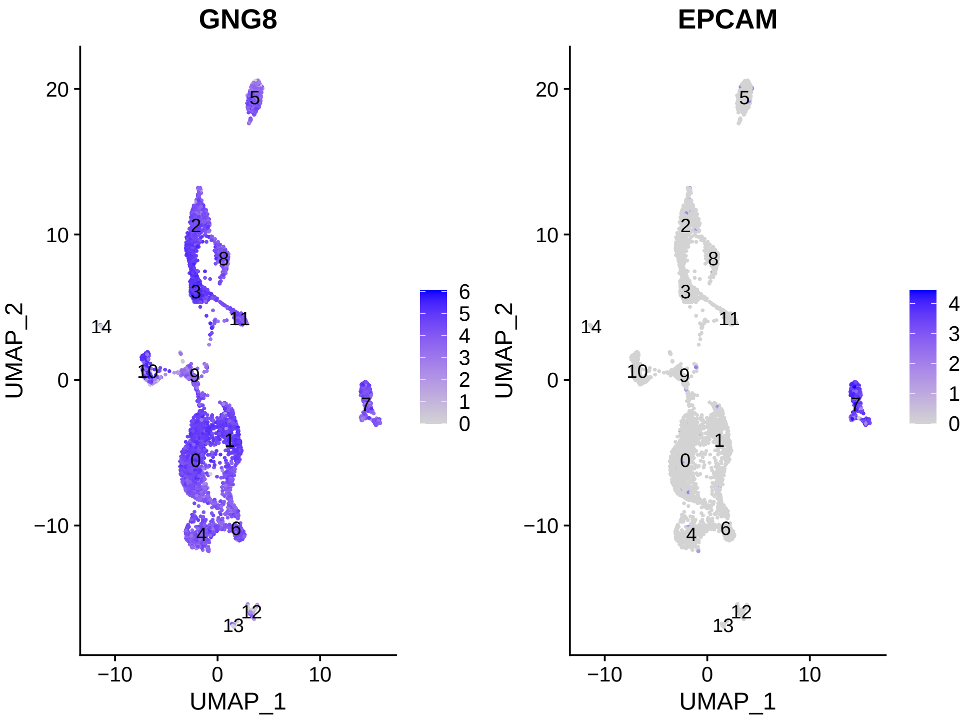
```r
feature_plot(adult_signac, features)
```

```
## [[1]]
```

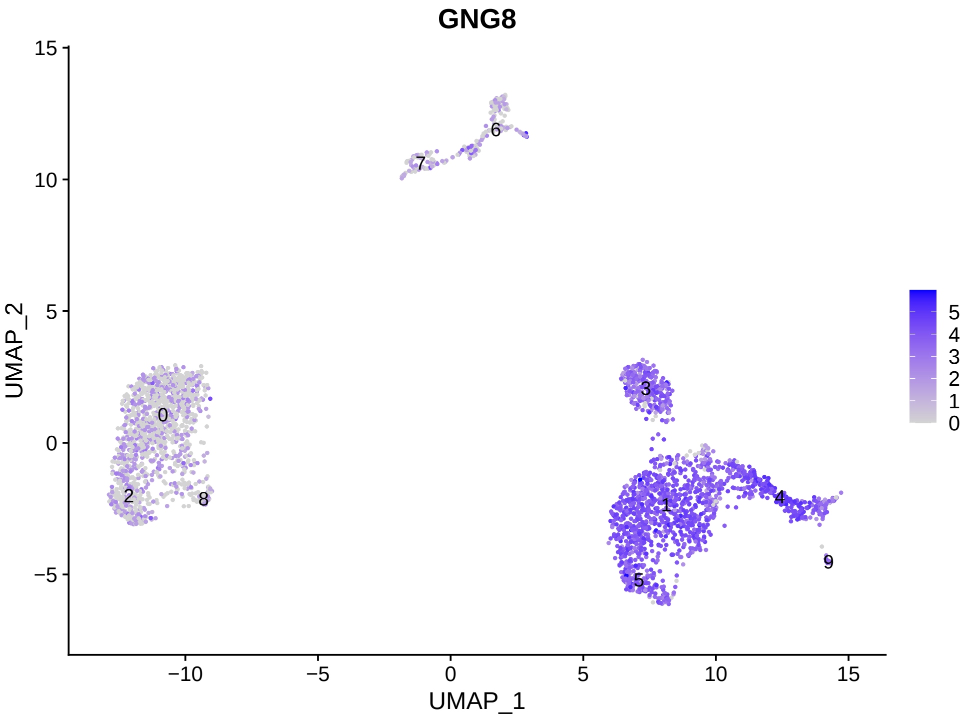


f. Using the `Seurat` results, visualize the genes `GNG8` and `EPCAM`. (2 points)

```
FeaturePlot(larval_seurat, features = features, label = TRUE)
```



```
FeaturePlot(adult_seurat, features = features, label = TRUE)
```



Problem 5: Discover marker genes (4 points)

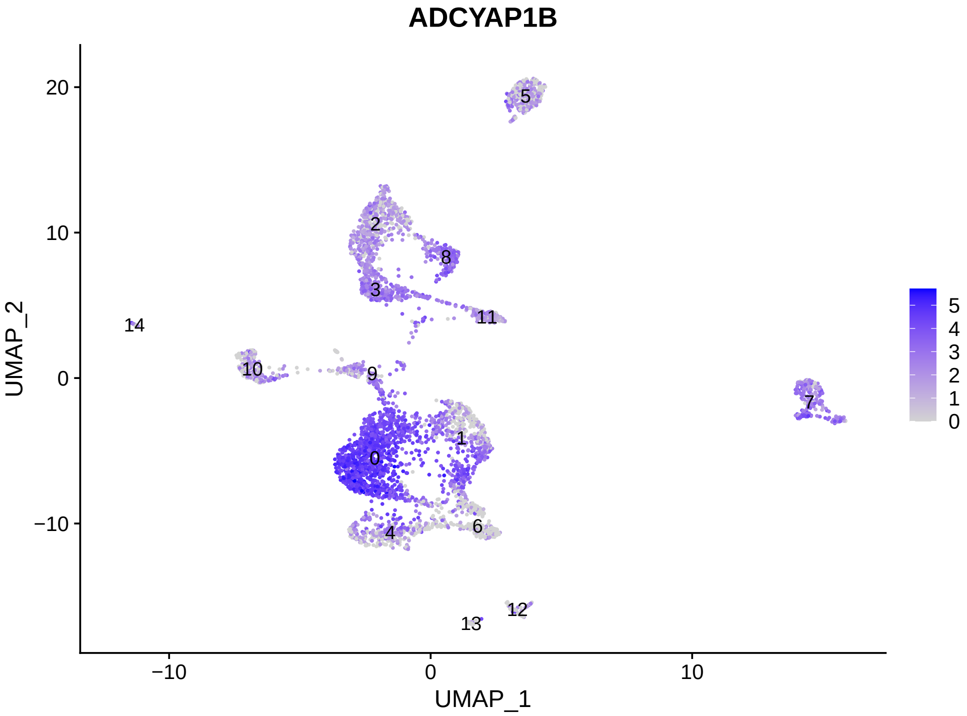
a. Using the `signac` larval object, find genes that are differentially expressed between cells in cluster 13 and all other cells. Print the top 5 genes significant at the 0.01 level after Bonferroni adjustment and with higher average expressions in cluster 13 than in all other cells. (1 point)

```
signac13 = find_markers(larval_signac, 13)
gene_list_signac = row.names(signac13[which((signac13$p_val_adj<0.01)&(signac13$ave_expr_diff>0)),])
(top5_signac = gene_list_signac[1:5])
```

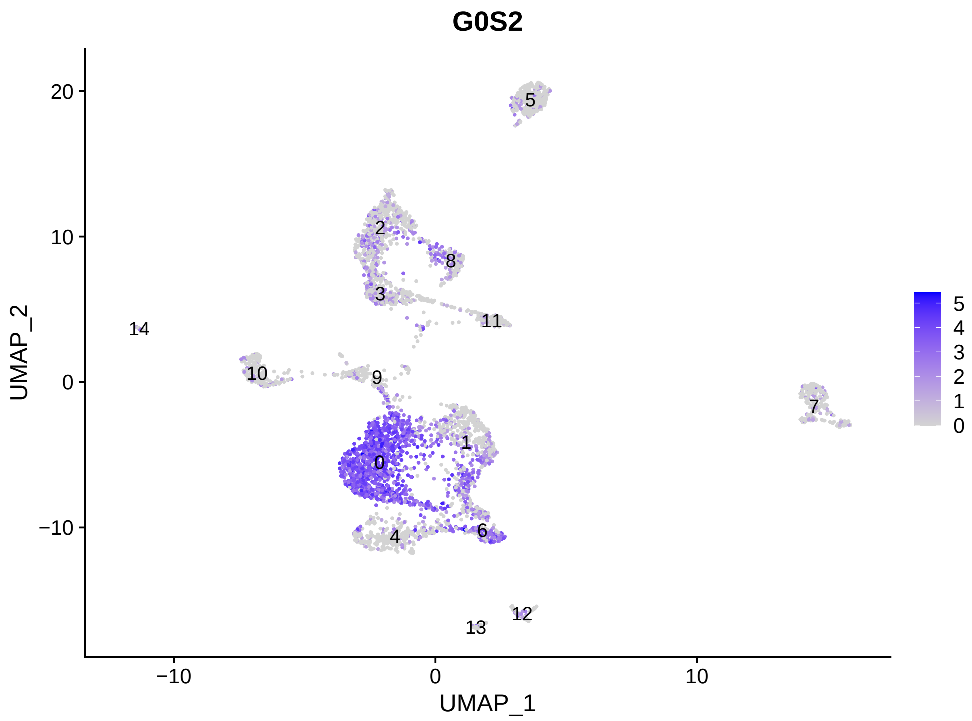
```
## [1] "ADCYAP1B" "G0S2"     "FXYD1"    "TMEFF1B"  "TAC3A"
```

b. Which cluster in the `Seurat` larval object do you think corresponds to cluster 13 in the `signac` larval object? (1 point)

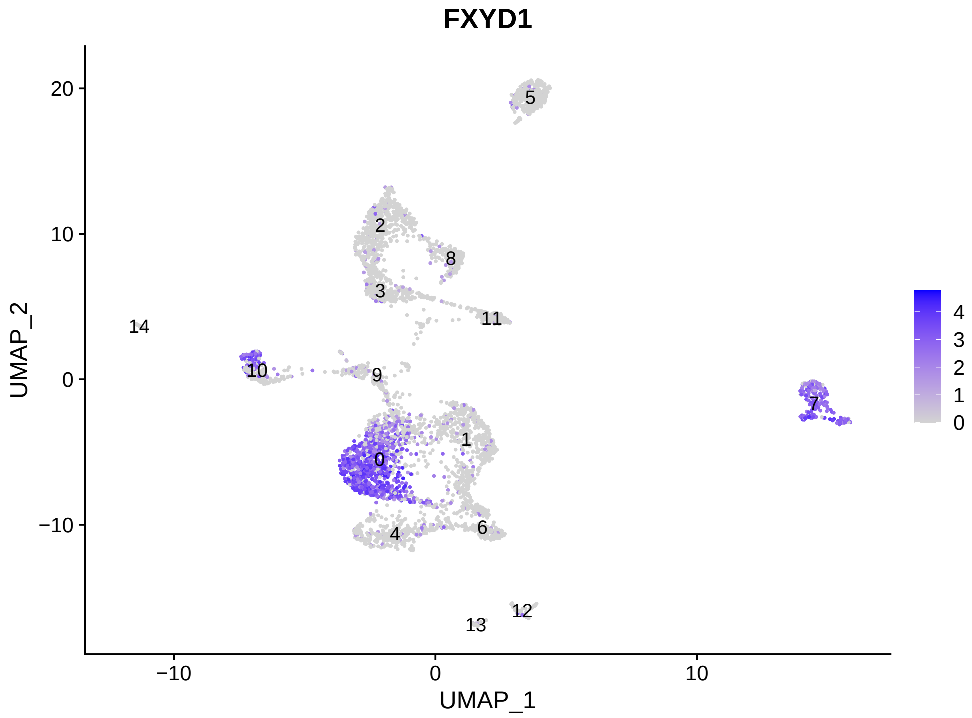
```
FeaturePlot(larval_seurat, features = "ADCYAP1B", label = TRUE)
```



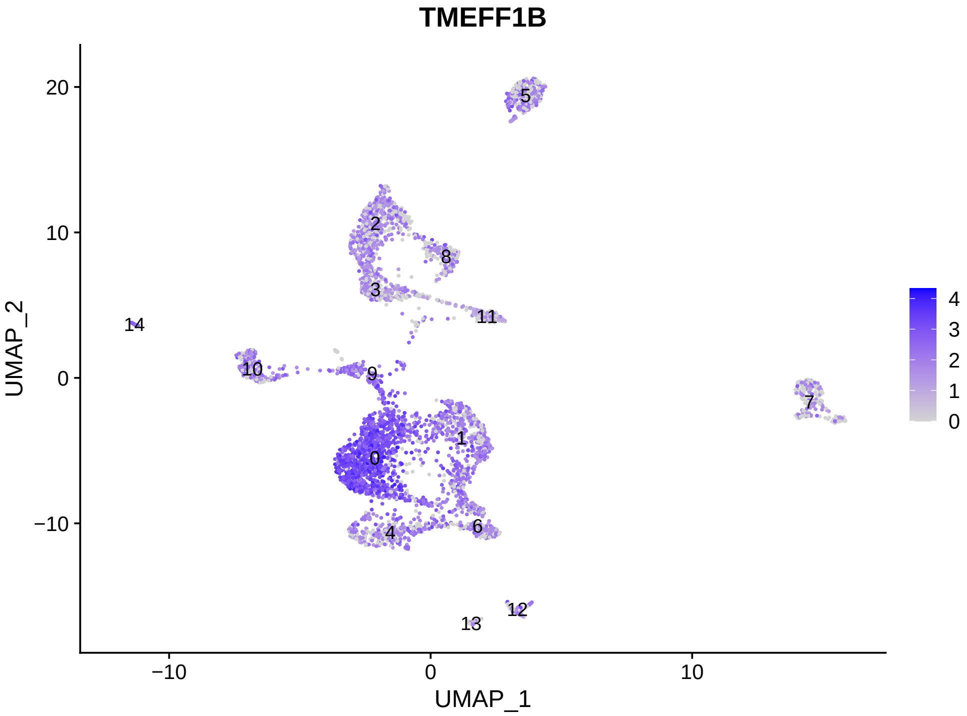
```
FeaturePlot(larval_seurat, features = "G0S2", label = TRUE)
```



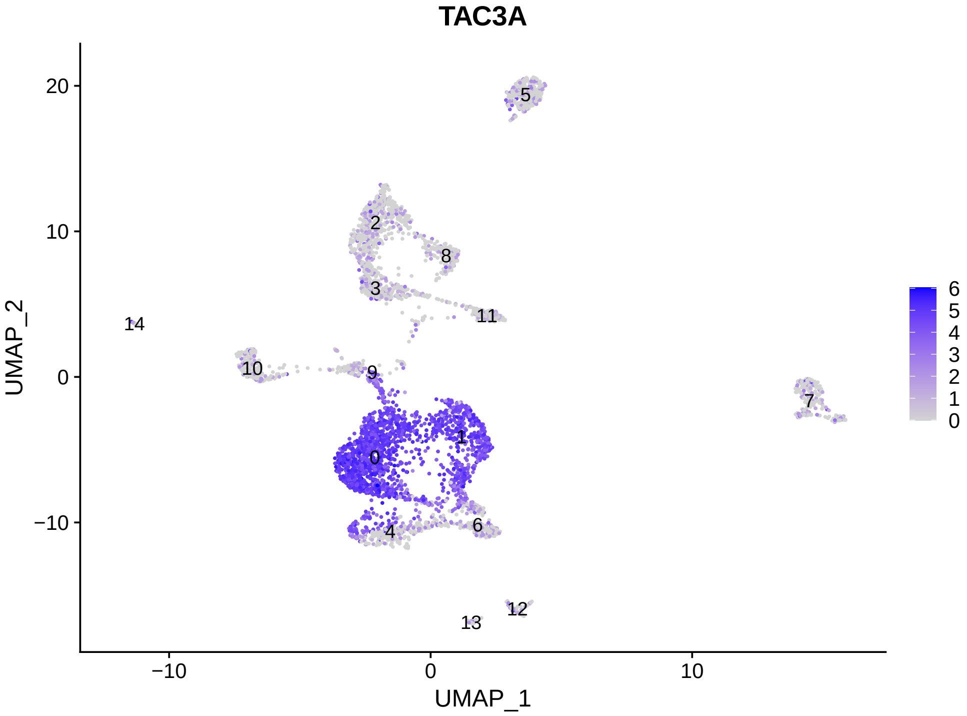
```
FeaturePlot(larval_seurat, features = "FXYD1", label = TRUE)
```



```r
FeaturePlot(larval_seurat, features = "TMEFF1B", label = TRUE)
```



```r
FeaturePlot(larval_seurat, features = "TAC3A", label = TRUE)
```



According to the feature plots of the top 5 genes in part (a), we can see that Cluster 0 has the highest expression level in all of them. Therefore, **Cluster 0** in the `Seurat` larval object corresponds to cluster 13 in the `signac` larval object.

c. Using the `Seurat` larval object, find genes that are differentially expressed between cells in cluster you identified in part (b), versus all other cells. Print the top 5 genes significant at the 0.01 level after Bonferroni adjustment and with higher average expressions in this cluster than in all other cells. (1 point)

```
seurat0 = FindMarkers(larval_seurat, ident.1 = 0)
gene_list_seurat = row.names(seurat0[which((seurat0$p_val_adj<0.01)&(seurat0$avg_log2FC>0)),])
(top5_seurat = gene_list_seurat[1:5])
```

```
## [1] "ADCYAP1B"  "TACR3L"    "FXYD1"     "PPP1R14AB" "TAC3A"
```

d. How many genes are shared between parts (a) and (c)?

```
length(intersect(top5_signac, top5_seurat))
```

```
## [1] 3
```

Three genes are shared between the top 5 genes in part (a) and part (c).