

# Expanding Acne Dataset

TamTo

2023-10-07

## Introduction

I'm working with a Seurat object for this analysis. Seurat objects are a representation of single-cell expression data for R, where we can analyze cell types and feature level (gene expression) of our sample conditions.

In this project, I'm identifying cell clusters from the lymphocyte dataset using key markers. In the previous analysis of this dataset, only 5 cell clusters were identified. I'm going to see how we can make this more specific.

As a brief overview, this dataset includes human skin tissue biopsies from normal nonlesional skin and acne lesional skin.

I've downloaded the lymphocyte dataset for this project. More information on this data and the paper published is available [here](#).

```
library(tidyverse) # load the tidyverse
library(Seurat) # we need this to work with Seurat objects

# Check the working directory.
getwd()

# Setting the wd.
setwd("/Users/tamto/Documents/GitHub/MolecularBiologyProjects/Projects/Project_ExpandingAcneDataset")

# Time to load the data in our global environment!
load("/Users/tamto/Desktop/soup62.lymphocyte2.Rdata")

# View the first 6 rows of the dataset.
head(soup62.lymphocyte2, 6)

# I like to look at it using view() since the data is so big and it's easier to visualize in Rstudio.
view(soup62.lymphocyte2)
```

## Checking Quality Control Metrics

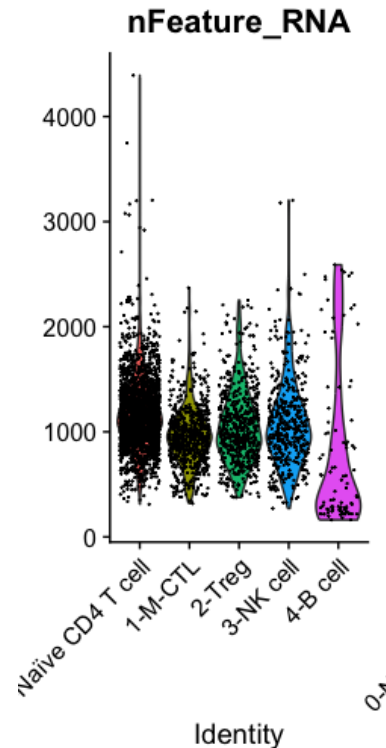
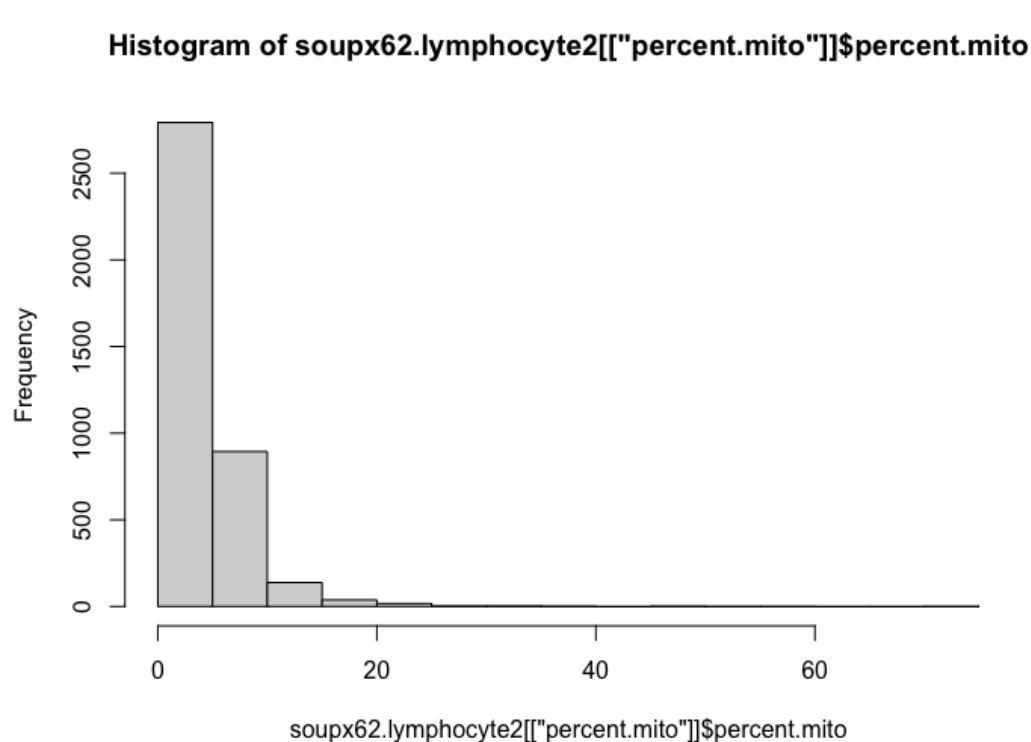
I like to just to double check the QC metrics of this data (includes mitochondrial content, feature count, etc.).

We have to be cautious of mitochondrial RNA because the mitochondria has its own RNA that can get mixed up with the RNA of our cells of interest. The PercentageFeatureSet() functions allows us to calculate the percentage of counts originating from features that contain our pattern of interest. In this case, we want to see how many genes are mitochondrial (i.e. start with "MT").

```
soup62.lymphocyte2[["percent.mito"]] <- PercentageFeatureSet(soup62.lymphocyte2, pattern = "^MT-")

# We can view this as a histogram to see what percentage of mitochondrial RNA the cells have. This hist
hist(soup62.lymphocyte2[["percent.mito"]])$percent.mito)

# We can plot a violin plot to make sure we have a good reading on enough genes, there's a good reading
VlnPlot(soup62.lymphocyte2, features = c("nFeature_RNA", "nCount_RNA", "percent.mito"), ncol = 3)
```



These plots show that we have mostly around 1000 genes, less than 10,000 gene count, and less than 15% mitochondrial RNA content.

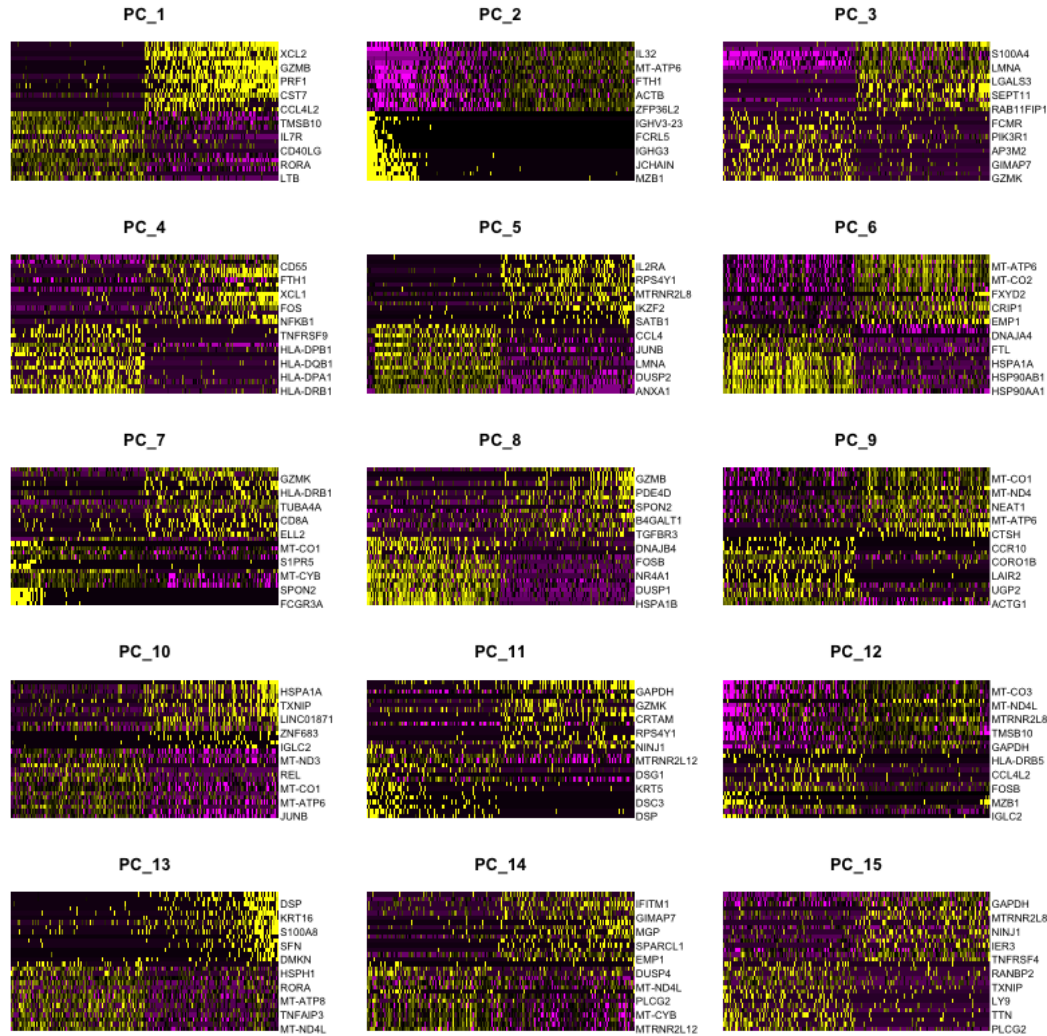
## Normalization and Dimensionality Reduction

This dataset has also already been normalized and scaled to make the values more comparable to each other. One problem is that our data is multi-dimensional (many genes) so it's not practical to analyze data with so many dimensions. Here I'm using principal component analysis (PCA) for dimensionality reduction using the RunPCA function on the Seurat object.

```
PCALymphocyte <- RunPCA(soup62.lymphocyte2, features = VariableFeatures(object = soup62.lymphocyte2))

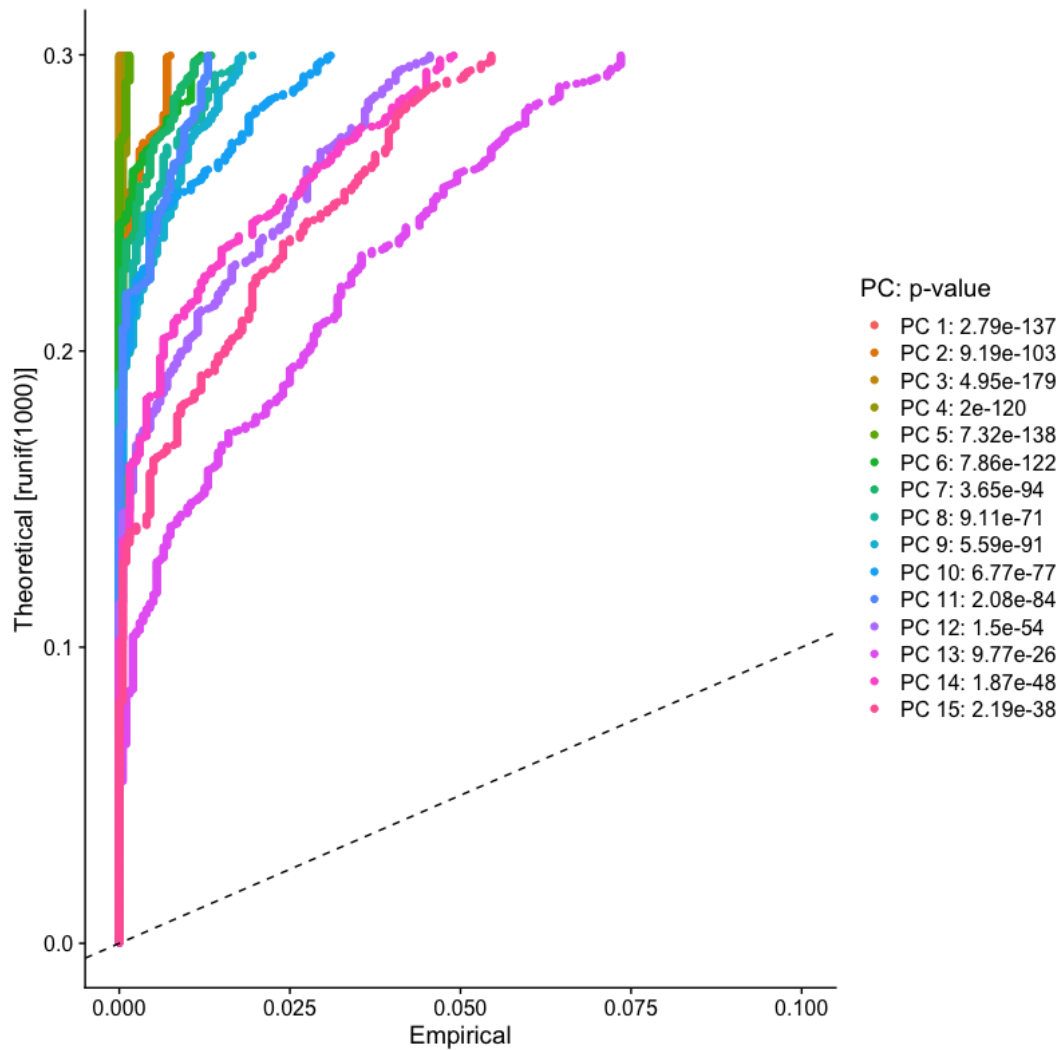
# I chose to look at 15 different dimensional variations since I'm trying to expand the dataset. We can
print(PCALymphocyte[["pca"]], dims = 1:15, nfeatures = 5)
VizDimLoadings(PCALymphocyte, dims = 1:15, reduction = 'pca')
```





We can create a JackStraw plot using the `JackStrawPlot()` function to verify which PCs are significant. `JackStraw()` and `ScoreJackStraw()` compare the distribution of p-values for each PC with a uniform distribution (dashed line in the plot). Significant PCs show a strong enrichment of features with low p-values (solid curves in the plot). The closer / more similar they are to the dashed line, the less meaningful/significant they are.

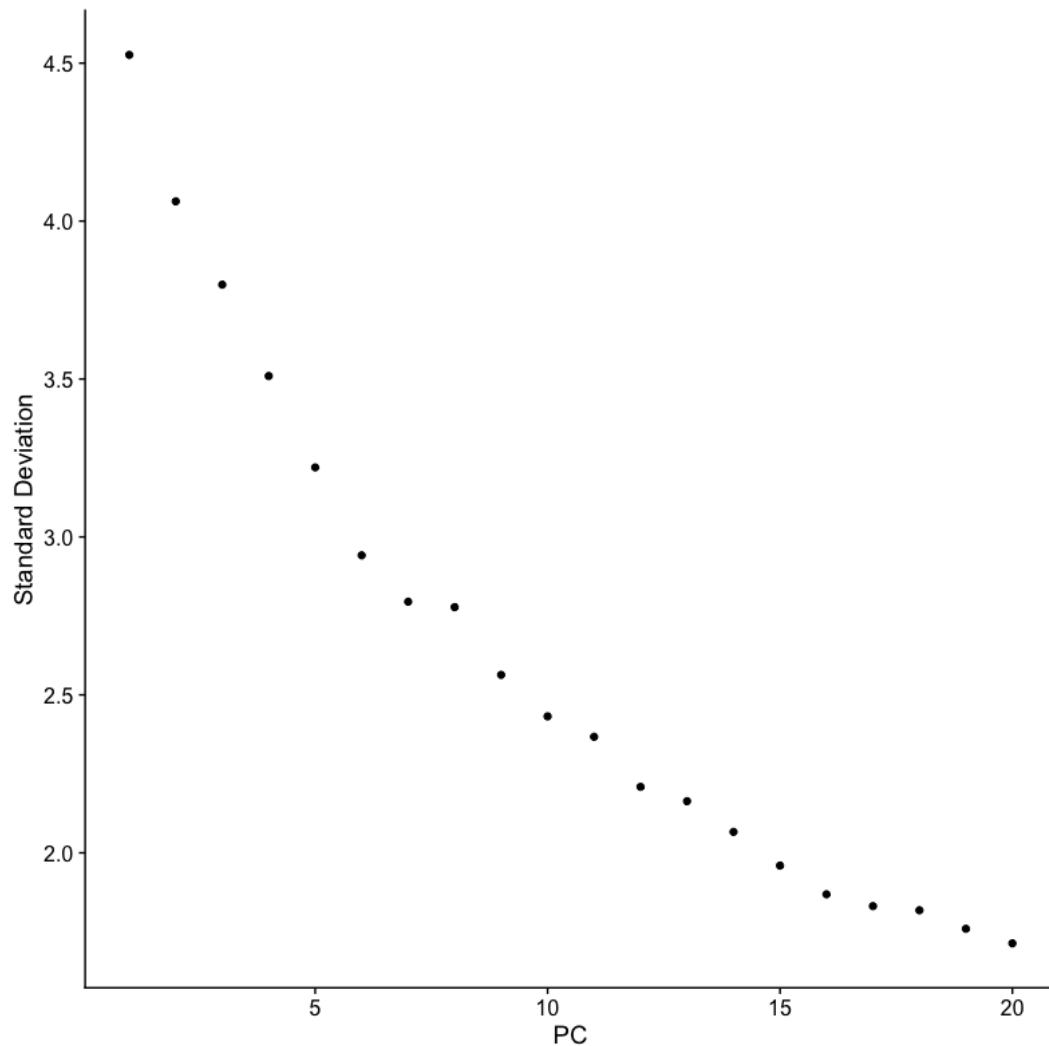
```
PCALymphocyte <- JackStraw(PCALymphocyte, num.replicate = 100)
PCALymphocyte <- ScoreJackStraw(PCALymphocyte, dims = 1:15)
JackStrawPlot(PCALymphocyte, dims = 1:15)
```



The plot shows all 15 lines are above the dashed line, but PC 12-15 start to deviate a bit. I'm going to cut it off at PC 11.

```
JackStrawPlot(PCALymphocyte, dims = 1:11)

# We can verify this another way using an Elbow plot with the ElbowPlot() function that ranks PCs based
ElbowPlot(PCALymphocyte)
```



We see there is also an elbow at PC 11 so it verifies the cut off I'm choosing here.

## Generating the UMAP

This is a dimensionality reduction technique used for visualization. It captures the manifold (topology/shape) of the data organization in higher dimensions and embeds a neighborhood of points. This uses the K-NN/KNN (K-nearest neighbor) algorithm to find similar gene expression profiles between 2 cells.

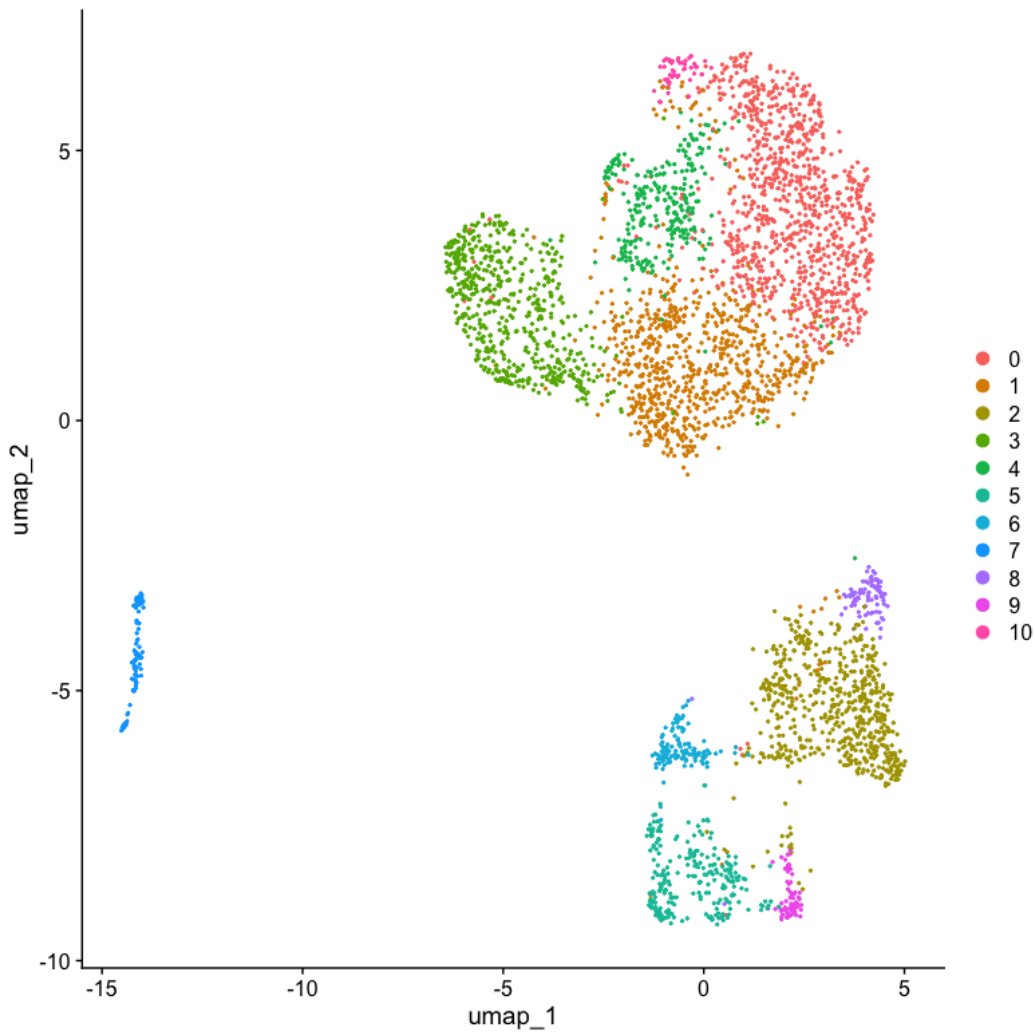
```
NewLymphocyteClusters <- FindNeighbors(PCALymphocyte, dims = 1:11)
NewLymphocyteClusters <- FindClusters(PCALymphocyte, resolution = 0.5)

# Now we can verify what the cluster IDs of the first 5 cells are and that we have a total of 11 clusters
head(Idents(NewLymphocyteClusters), 5)

NewLymphocyteClusters <- RunUMAP(NewLymphocyteClusters, reduction = "pca", dims = 1:11)
DimPlot(NewLymphocyteClusters, reduction = "umap")
```

Looks like we have 11 total levels (cell types) and they are named by number. We can create a UMAP to

view the graph-based clusters and verify these cell types cluster on the map. Next is identifying key markers for each cluster to name the cell type.



## Identifying Cluster Key Markers

Now we need to find out what each of these cells are. We can do this by finding out their top genes with `FindAllMarkers()` and deduce which cells they are based on the expression of these genes. I like to find the top 10 markers of each cluster, `min.pct` argument is used to restrict to features detected in a minimum fraction of the chosen percentage of cells (i.e. here it has to be expressed in at least 25% of the cells). I chose a minimum log2 fold change threshold as 0.25 for average expression of a gene in one cluster relative to other clusters (i.e. `logfc.threshold = 0.25`).

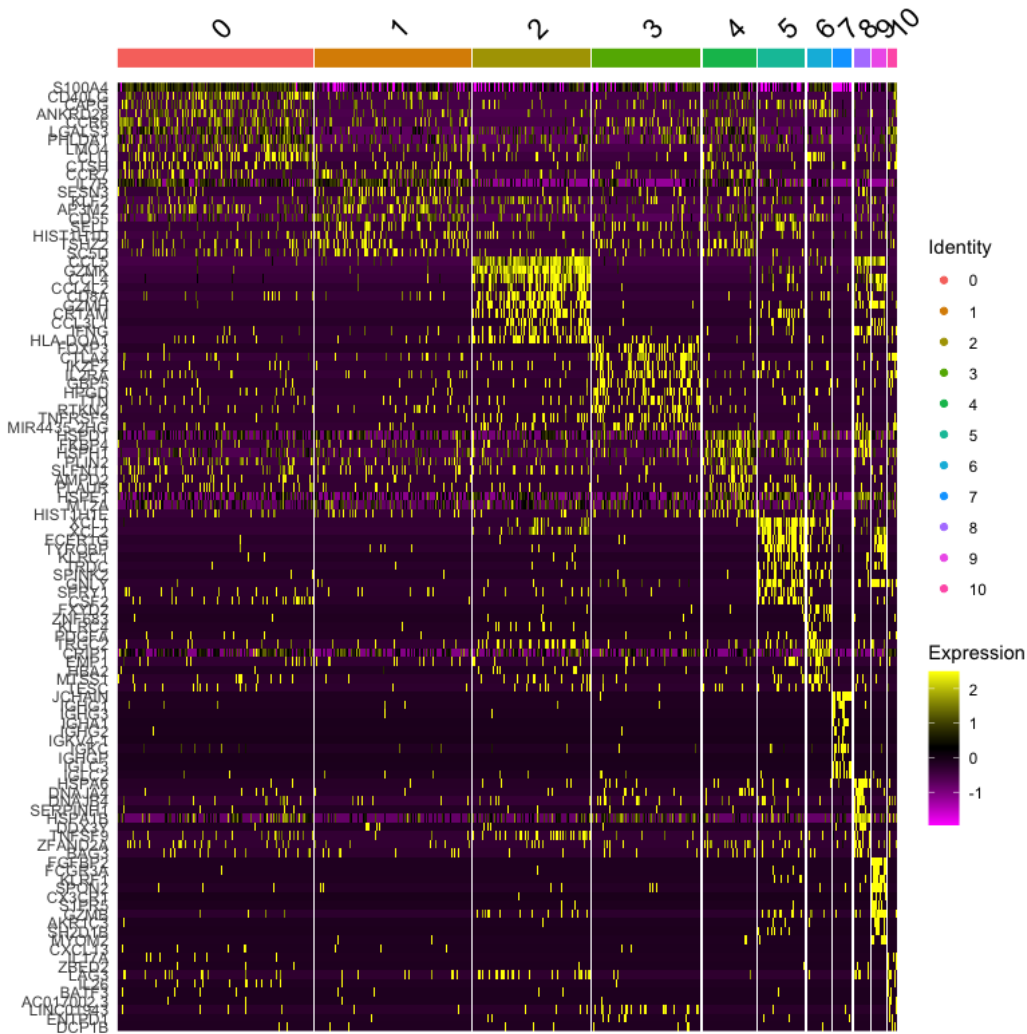
```
cluster_markers <- FindAllMarkers(NewLymphocyteClusters,
                                   only.pos = TRUE,
                                   min.pct = 0.25,
                                   logfc.threshold = 0.25)

# You can view the top genes for each cluster using slice_max().
```

```
cluster_markers %>%
  group_by(cluster) %>%
  slice_max(n = 1, order_by = avg_log2FC)

# I want to make a heatmap of the top 10 genes since this gives us more information than just one top g
top10markers <- cluster_markers %>%
  group_by(cluster) %>%
  top_n(n = 10, wt = avg_log2FC)

DoHeatmap(NewLymphocyteClusters, features = top10markers$gene)
```



Since the heatmap can get quite overwhelming with a lot of clusters, we can look at one cluster at a time. I'm going to start with cluster 0 and then this would be repeated for each cluster to find out their top genes.

```
cluster0.markers <- FindMarkers(NewLymphocyteClusters, ident.1 = 0, min.pct = 0.25)
head(cluster0.markers, n = 10)

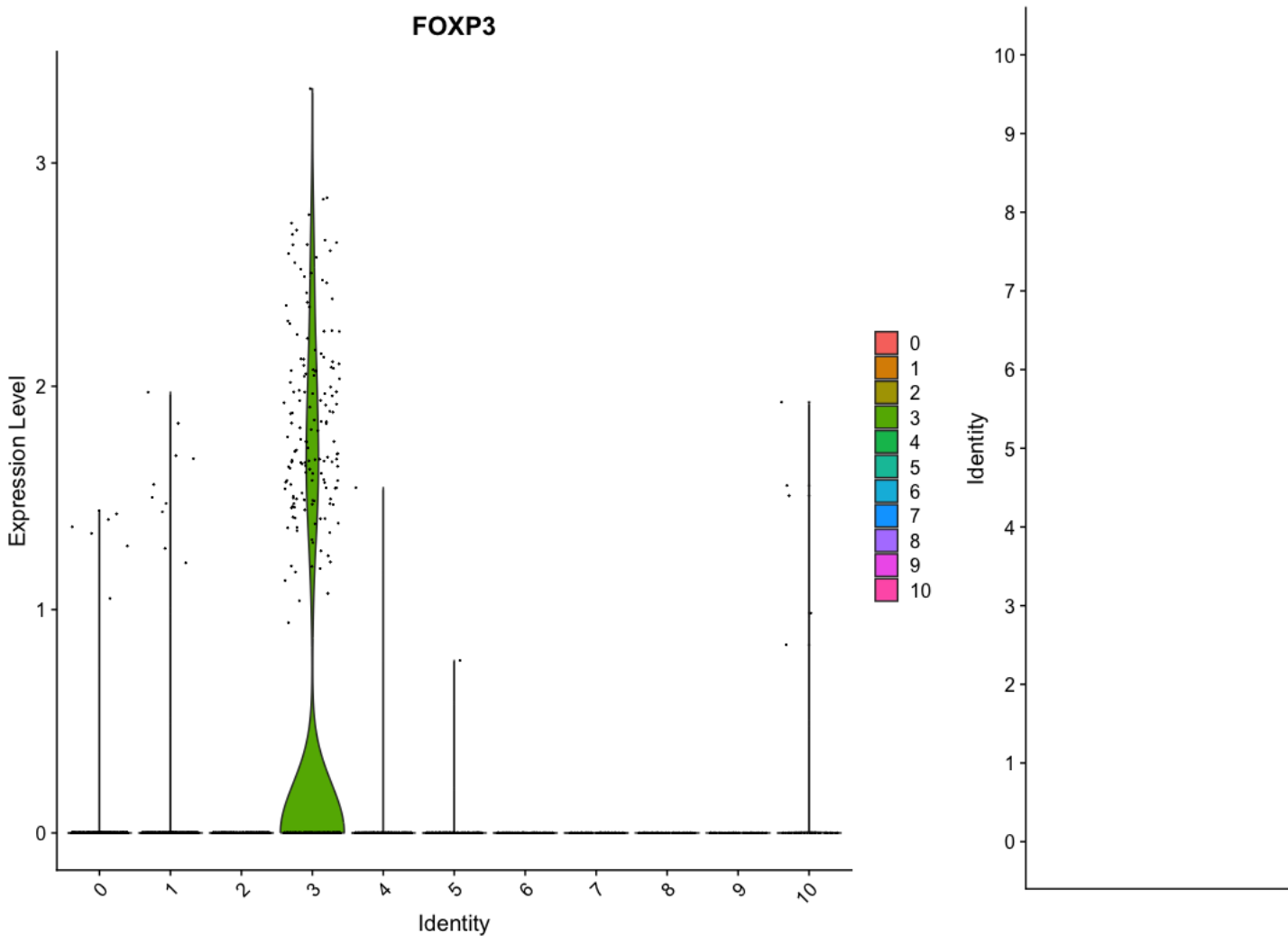
# If it is difficult to distinguish a certain cluster from another cluster, we can use the following li
```



```
cluster8.markers <- FindMarkers(NewLymphocyteClusters, ident.1 = 8, ident.2 = 2, min.pct = 0.25)
head(cluster8.markers, n = 10)

# I'm more of a visual analyzer, so I recommend making violin plots or dot plots of unique genes that a

VlnPlot(NewLymphocyteClusters, features = c("FOXP3"))
DotPlot(NewLymphocyteClusters, features = c("FOXP3"), cols = c("blue", "red"))
```



Looks like cluster 3 is a Treg cell cluster. We can then repeat this with more unique genes to find out the remaining cell types for each cluster.

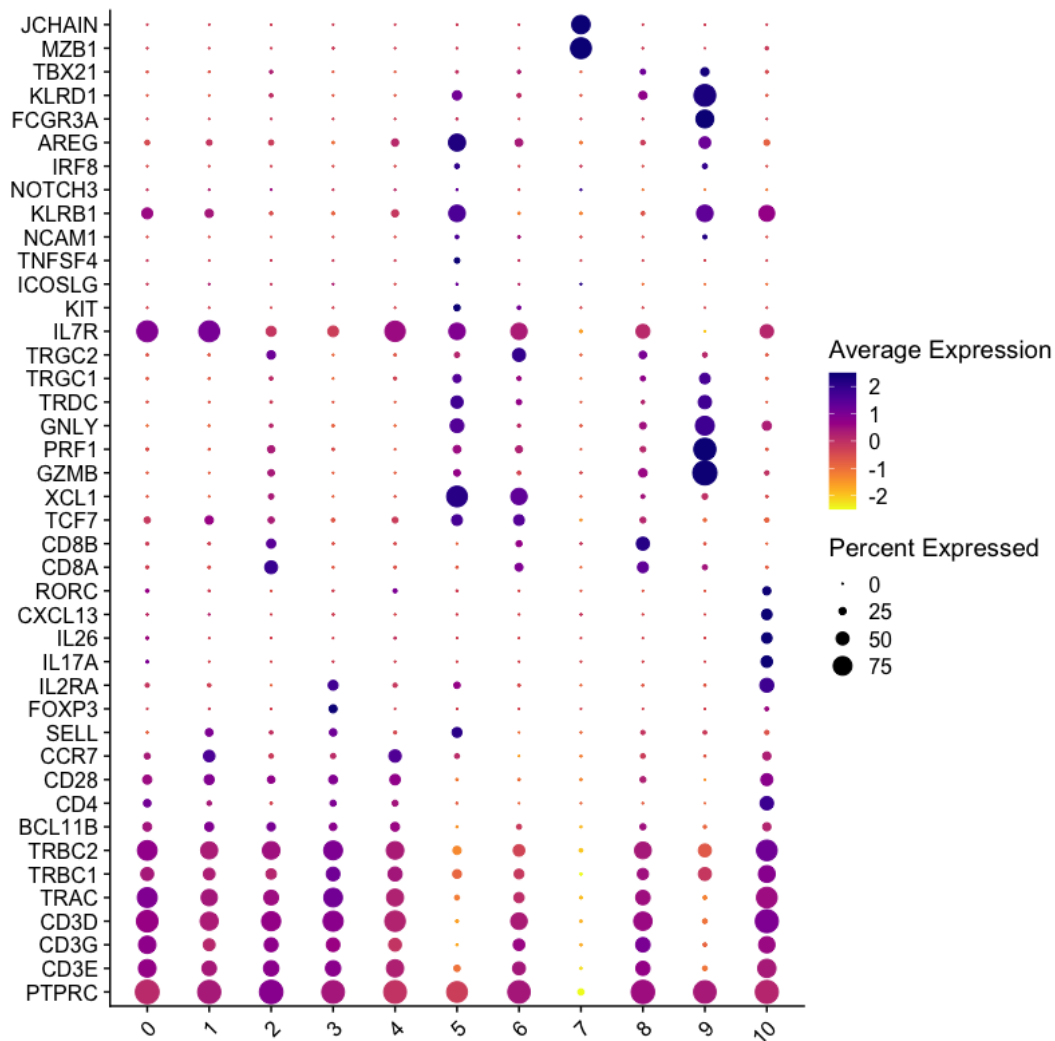
## Verifying Clusters

Although we might be certain of a cluster based on specific genes, we also need to verify that these cell types are actually true by comparing the genes across all clusters. We can do this by plotting genes and clusters using the `scCustomize` package.

```
# install.packages("scCustomize")
library(scCustomize)

genes <- c("PTPRC", "CD3E", "CD3G", "CD3D", "TRAC", "TRBC1", "TRBC2", "BCL11B", "CD4", "CD28", "CCR7", "IL2RA", "FOXP3", "SELL", "CD8B", "CD8A", "RORC", "CXCL13", "IL26", "IL17A", "IL2RA", "FOXP3", "SELL", "CCR7", "CD28", "CD4", "BCL11B", "TRBC2", "TRBC1", "TRAC", "CD3D", "CD3G", "CD3E", "PTPRC")

DotPlot_scCustom(seurat_object = NewLymphocyteClusters, features = genes, flip_axes = T, x_lab_rotate = 90)
```



Each cluster seems to have their own high expression of genes that correlate to certain cell types. Viewing it on a dotplot makes it easy to visualize and confirm our findings!

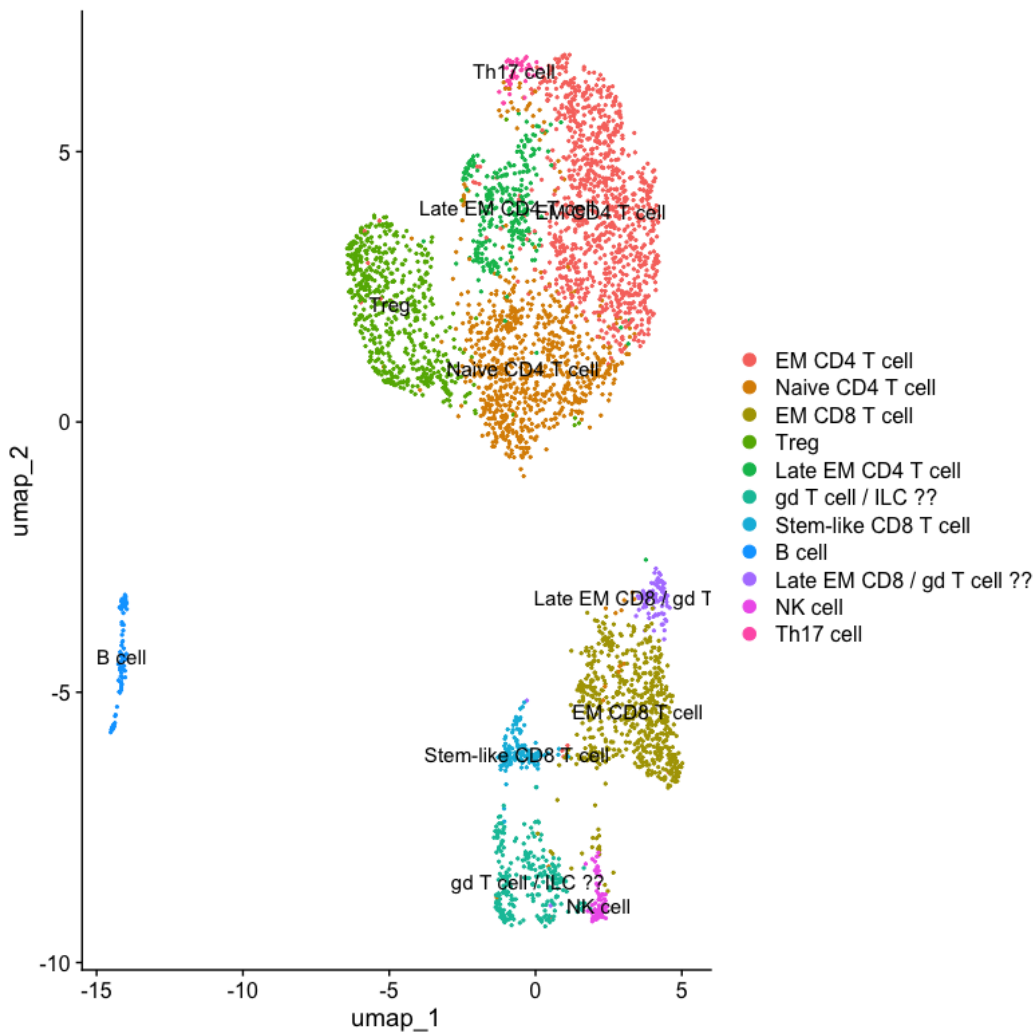
## Renaming the Clusters

Now I've found out what each cluster may be based on their specific genes. It's time to rename the clusters with their correct cell type.

```
new.cluster.ids <- c("EM CD4 T cell", "Naive CD4 T cell", "EM CD8 T cell", "Treg", "Late EM CD4 T cell")
names(new.cluster.ids) <- levels(NewLymphocyteClusters)
```

```
NewLymphocyteClusters <- RenameIdents(NewLymphocyteClusters, new.cluster.ids)

# View the new UMAP with the updated clusters!
DimPlot(NewLymphocyteClusters, reduction = 'umap', label = TRUE, pt.size = 0.4)
```



I want to reorder the cell types so that it's easier to analyze when we make downstream analyses. Make the new ordered cell types the active identities.

```
NewLymphocyteClusters@active.ident <- factor(NewLymphocyteClusters@active.ident,
                                              levels = c("Naive CD4 T cell",
                                                         "EM CD4 T cell",
                                                         "Late EM CD4 T cell",
                                                         "Treg",
                                                         "Th17 cell",
                                                         "Stem-like CD8 T cell",
                                                         "EM CD8 T cell",
                                                         "Late EM CD8 / gd T cell ??",
                                                         "gd T cell / ILC ??",
                                                         "NK cell",
```



## Saving the Data

This data can now be used for downstream analyses (shown in the next project).

```
save(NewLymphocyteClusters, file = "NewLymphocyteClusters.Rdata")
```

## CONCLUSION

Upon expanding this dataset, we were able to go from 5 clusters to 11 clusters (“Naive CD4 T cell”, “EM CD4 T cell”, “Late EM CD4 T cell”, “Treg”, “Th17 cell”, “Stem-like CD8 T cell”, “EM CD8 T cell”, “Late EM CD8 / gd T cell ??”, “gd T cell / ILC ??”, “NK cell”, “B cell”). There may be gd T cells and ILCs mixed within the same cluster because the ILC cluster seems to have cells expressing genes that represent gd T cells (TRDC, TRGC2, TRGC1) when they should not be positive for these genes. Moreover, gd T cells may be present in the Late EM CD8 T cell cluster as well due to similar functional gene signatures. In literature, these cell types have been shown to be quite similar, so we would need a dataset with lots more cells to make a more concrete verification that those are the correctly named clusters.

Overall, there are now 11 cell types distinguished in the lymphocyte dataset. Researchers can use this new information to generate hypotheses about genes of interest in more specific cell types.

Another method for annotating cell types may be to use the online source, Azimuth, at <https://azimuth.hubmapconsortium.org/>. It’s super quick and easy to use because you just upload your data to their reference datasets that can then name the clusters, however, there are very limited reference datasets available. In this case, there is no human skin dataset on the site that we can refer to.

CELLxGENE (<https://cellxgene.cziscience.com/>) is another useful source where you can find single-cell data and explore gene expression across many tissues/cell types. It allows you to download and integrate data as well.

““