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
```

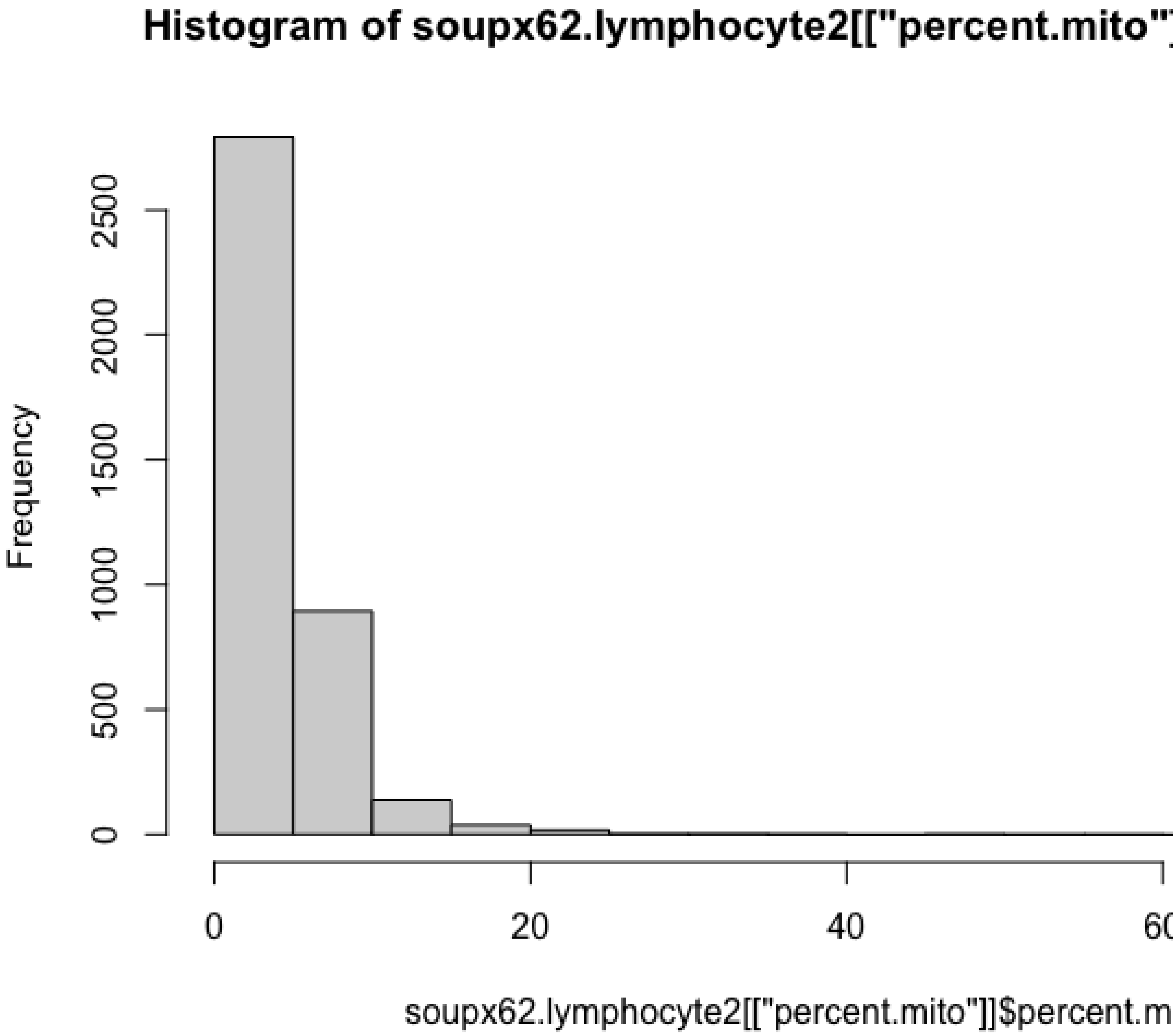
```
# Check the working directory.
getwd()

# Setting the wd to where the file is located.
setwd("/Users/tamto/Documents/GitHub/MolecularBiologyProjects/Projects/Project_ExpandingAcneDataset")

# Time to load the data in our global environment!
load("/Users/tamto/Documents/GitHub/MolecularBiologyProjects/Projects/Project_ExpandingAcneDataset/soup

# View the first 6 rows of the dataset.
head(soupx62.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(soupx62.lymphocyte2)
```

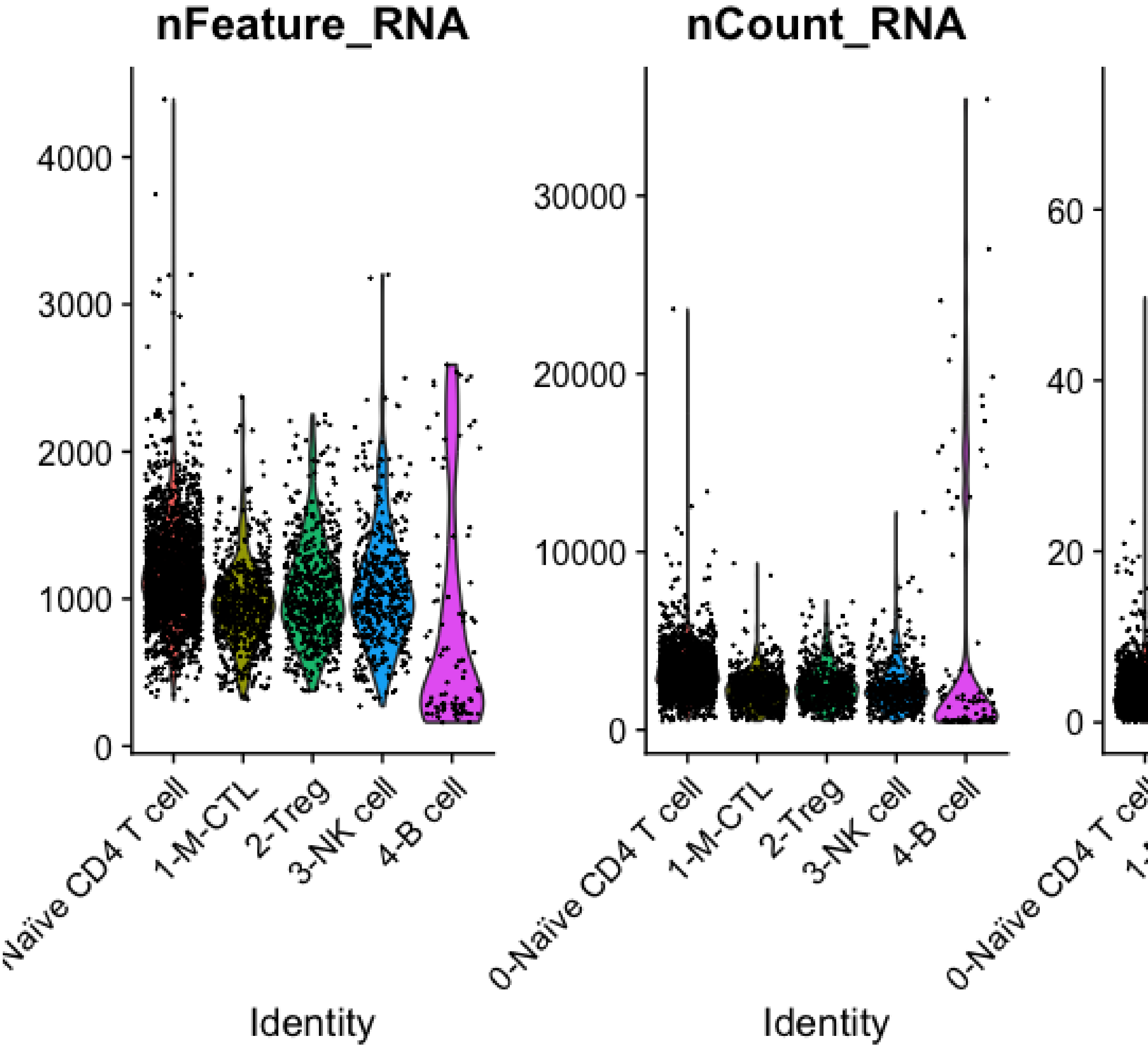
Checking Quality Control Metrics

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").

```
soupx62.lymphocyte2[["percent.mito"]] <- PercentageFeatureSet(soupx62.lymphocyte2, pattern = "^MT-")
# We can view this as a histogram to see what percentage of mitochondrial RNA the cells have. This hist
hist(soupx62.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(soupx62.lymphocyte2, features = c("nFeature_RNA", "nCount_RNA", "percent.mito"), ncol = 3)
```



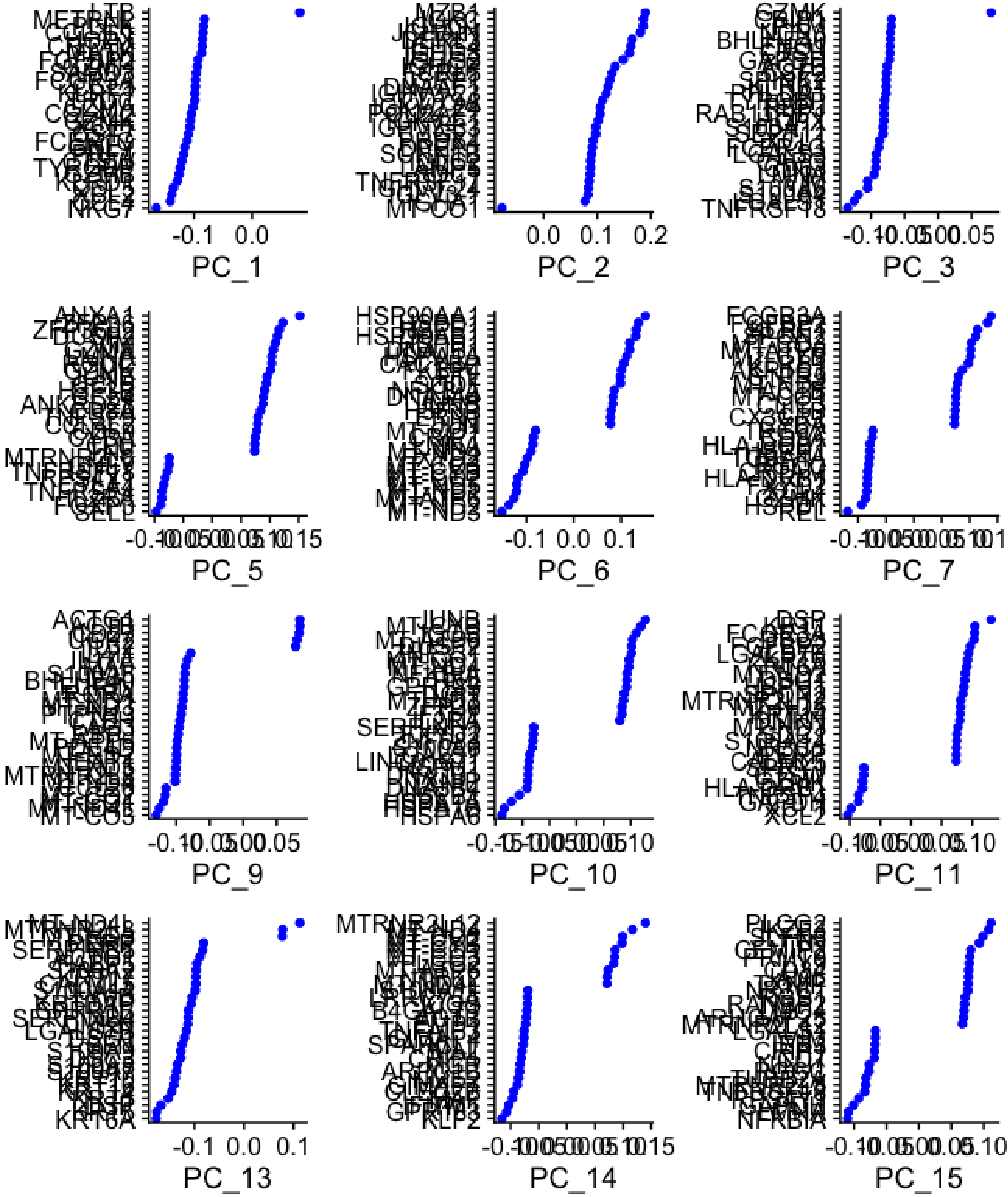


These plots show that we have mostly around 1000 genes, less than 10,000 gene count, and less than 15% mitochondrial RNA content.

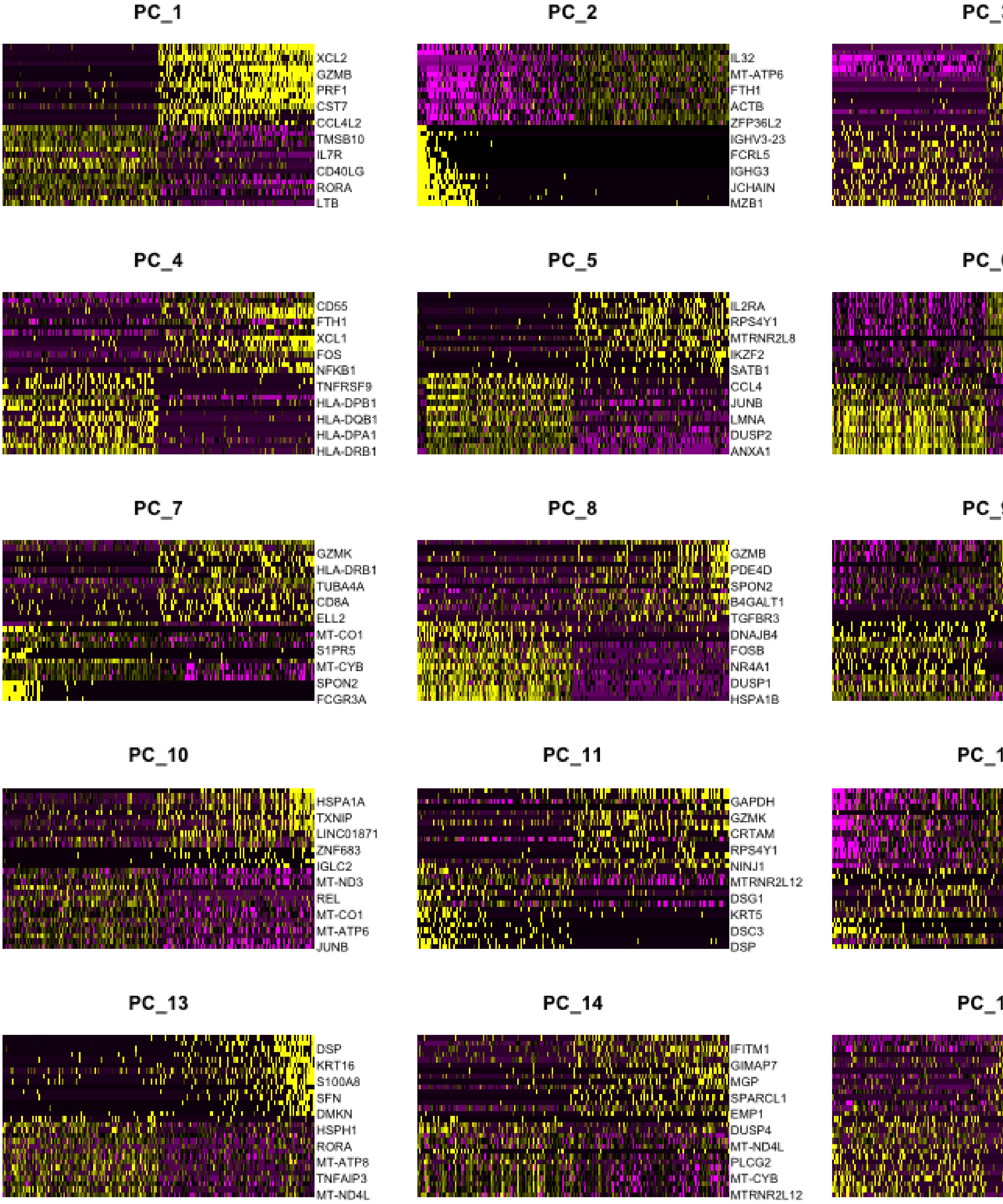
Normalization and Dimensionality Reduction

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(soupx62.lymphocyte2, features = VariableFeatures(object = soupx62.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')
```

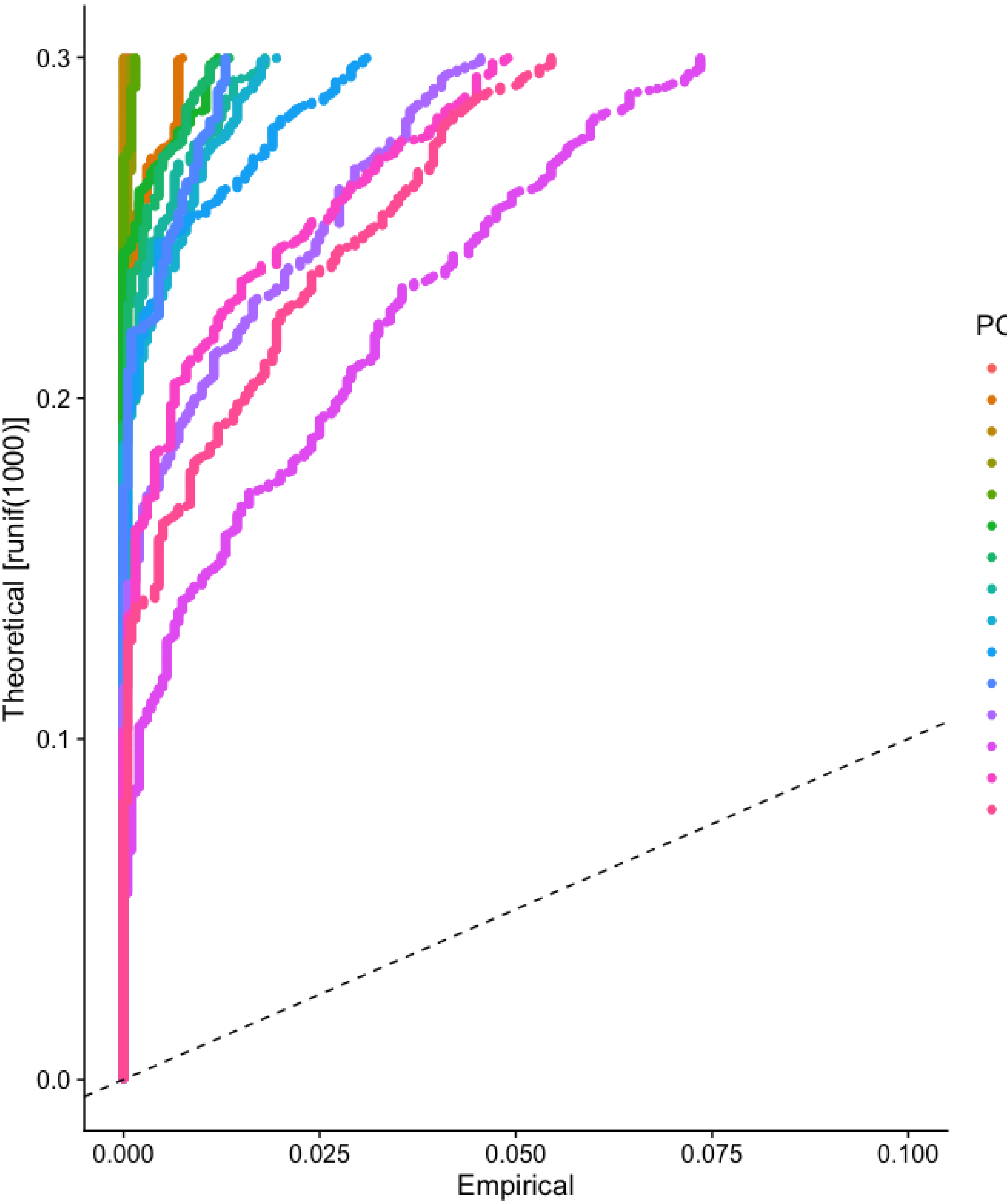

LTP
METRNL
NKG7
-0.1
0.0
PC_1
MZB1
MT-CO1
0.0
0.1
0.2
PC_2
GZMK
BHLHE40
RAB11FIP1
TNFRSF18
-0.10-0.050.000.05
PC_3
ANXA1
MTRNR2L8
SELL
-0.10-0.050.000.050.100.15
PC_5
HSP90AA1
MT-ND3
-0.1
0.0
0.1
PC_6
FCGR3A
HSPD1
-0.10-0.050.000.050.10
PC_7
ACTG1
BHLHE40
MT-CO3
-0.10-0.050.000.05
PC_9
JUNB
HSPA6
-0.15-0.10-0.050.000.050.10
PC_10
DSP
XCL2
-0.10-0.050.000.050.10
PC_11
MT-ND4L
KRT6A
-0.1
0.0
0.1
PC_13
MTRNR2L12
KLF2
-0.10-0.050.000.050.100.15
PC_14
PLCG2
NFKBIA
-0.10-0.050.000.050.10
PC_15

```
# DimHeatmap() function allows to explore heterogeneity in the dataset and decide which PCs to include 

DimHeatmap(PCALymphocyte, dims = 1:15, cells = 500, balanced = TRUE)
```


PC_1
XCL2
GZMB
PRF1
CST7
CCL4L2
TMSB10
IL7R
CD40LG
RORA
LTB
PC_2
IL32
MT-ATP6
FTH1
ACTB
ZFP36L2
IGHV3-23
FCRL5
IGHG3
JCHAIN
MZB1
PC_4
CD55
FTH1
XCL1
FOS
NFKB1
TNFRSF9
HLA-DPB1
HLA-DQB1
HLA-DPA1
HLA-DRB1
PC_5
IL2RA
RPS4Y1
MTRNR2L8
IKZF2
SATB1
CCL4
JUNB
LMNA
DUSP2
ANXA1
PC_7
GZMK
HLA-DRB1
TUBA4A
CD8A
ELL2
MT-CO1
S1PR5
MT-CYB
SPON2
FCGR3A
PC_8
GZMB
PDE4D
SPON2
B4GALT1
TGFBR3
DNAJB4
FOSB
NR4A1
DUSP1
HSPA1B
PC_10
HSPA1A
TXNIP
LINC01871
ZNF683
IGLC2
MT-ND3
REL
MT-CO1
MT-ATP6
JUNB
PC_11
GAPDH
GZMK
CRTAM
RPS4Y1
NINJ1
MTRNR2L12
DSG1
KRT5
DSC3
DSP
PC_13
DSP
KRT16
S100A8
SFN
DMKN
HSPH1
RORA
MT-ATP8
TNFAIP3
MT-ND4L
PC_14
IFITM1
GIMAP7
MGP
SPARCL1
EMP1
DUSP4
MT-ND4L
PLCG2
MT-CYB
MTRNR2L12

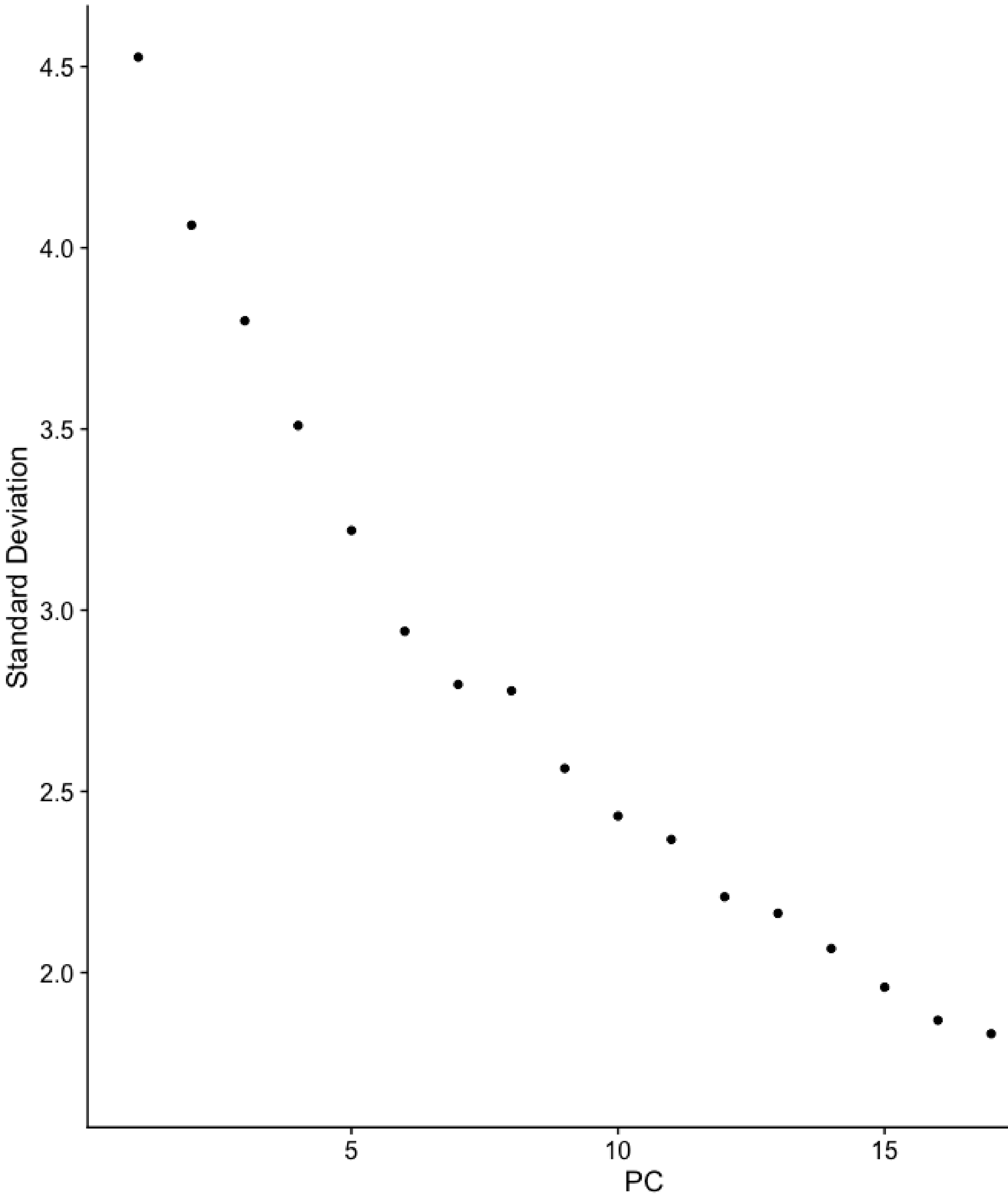
We can create a JackStraw plot using the JackStrawPlot() function to verify which PCs are significant. JackStraw() and ScoreJackStraw() compare the distribution of p-values for each PC with a uniform distribution (dashed line in the plot). Significant PCs show a strong enrichment of features with low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

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


0.3
0.2
0.1
0.0
Theoretical [runif(1000)]
0.000
0.025
0.050
0.075
0.100
Empirical
PC

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)
```


4.5
4.0
3.5
3.0
2.5
2.0
Standard Deviation
5
10
15
PC

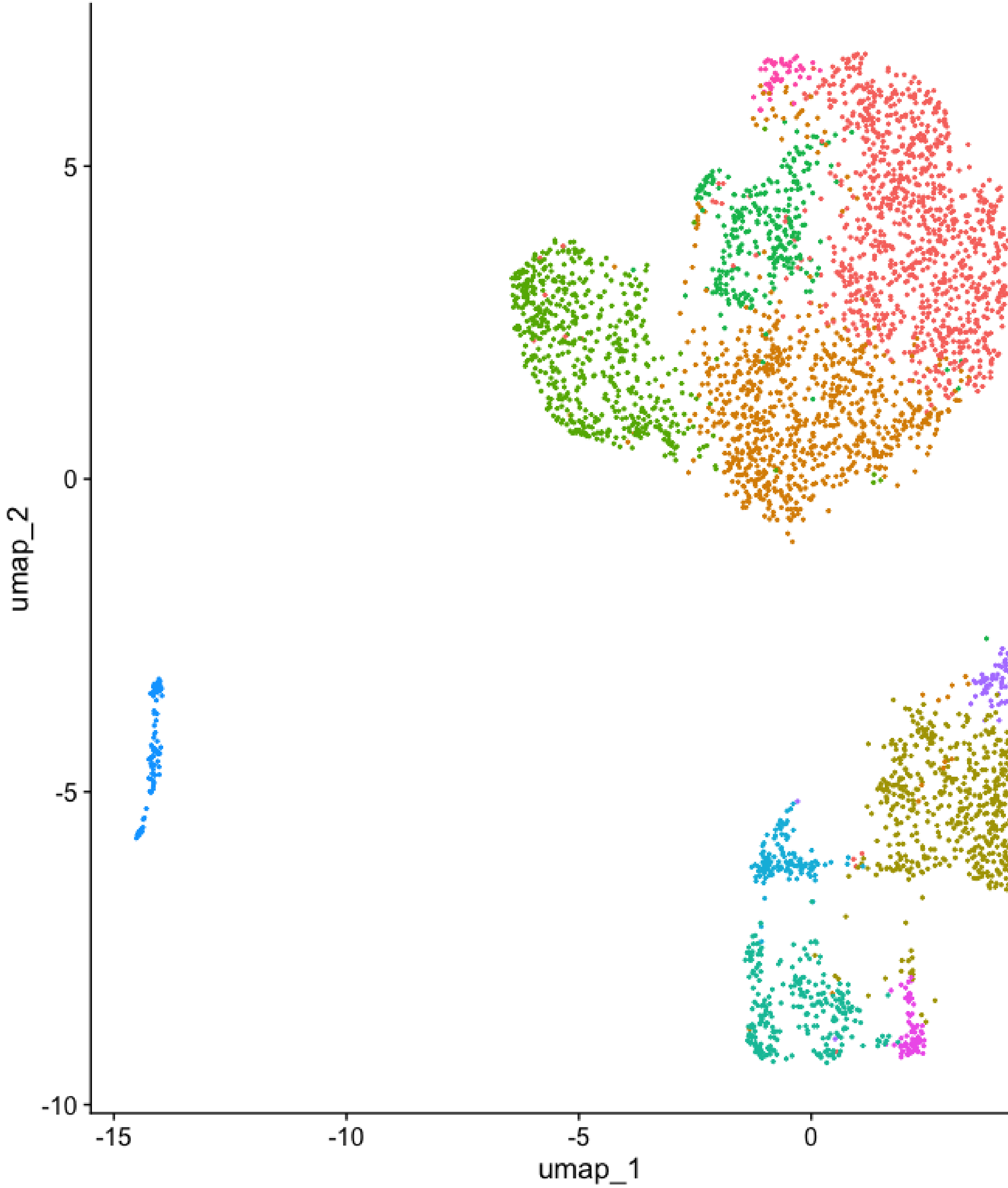
We see there is also an elbow at PC 11 so it verifies the cut off I'm choosing here.

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.

```r
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 cluste
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.

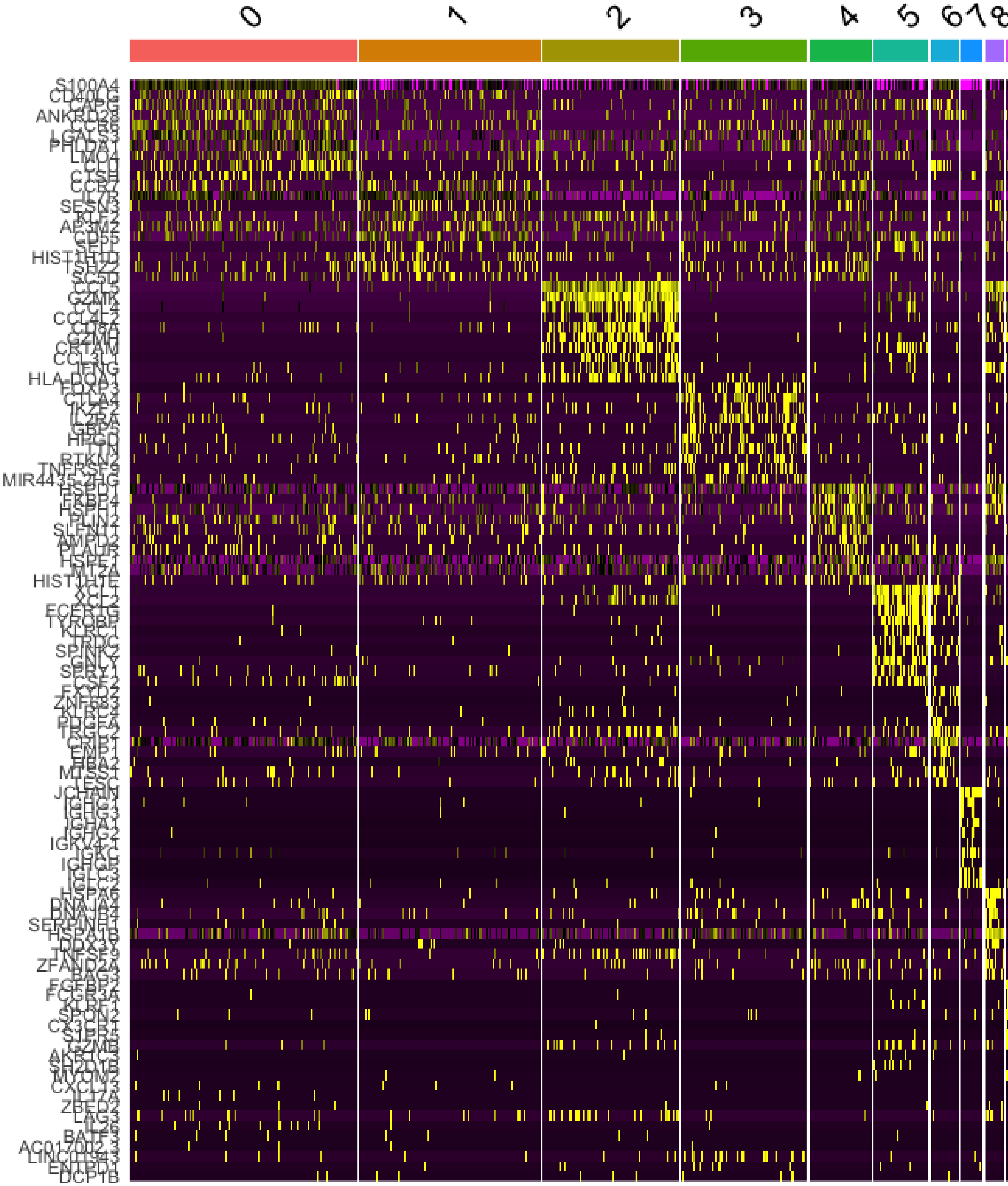

5
0
-5
-10
umap_2
-15
-10
-5
0
umap_1

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).

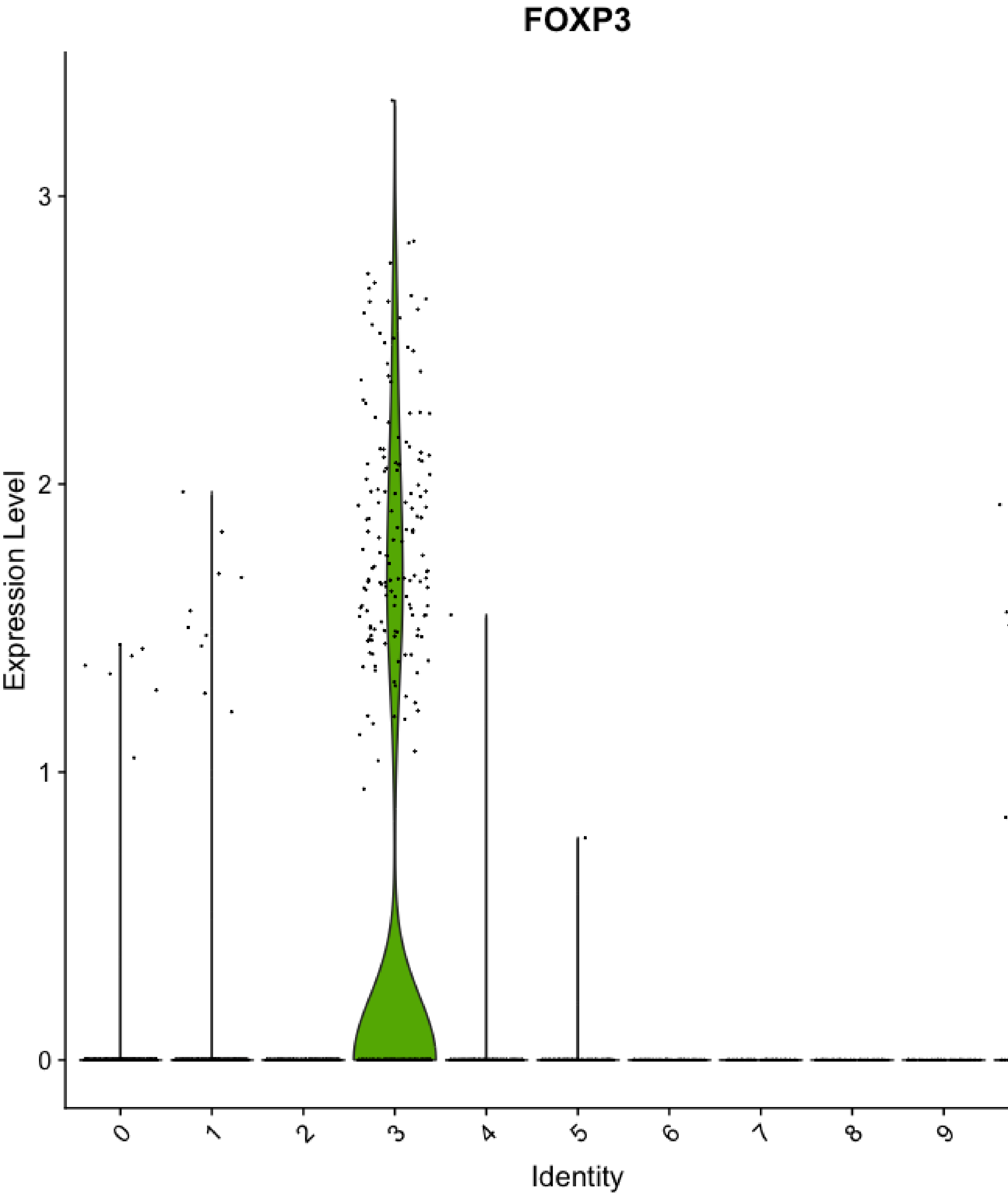
```
cluster_markers <- FindAllMarkers(NewLymphocyteClusters,
                                  only.pos = TRUE,
                                  min.pct = 0.25,
                                  logfc.threshold = 0.25)

# 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)
```

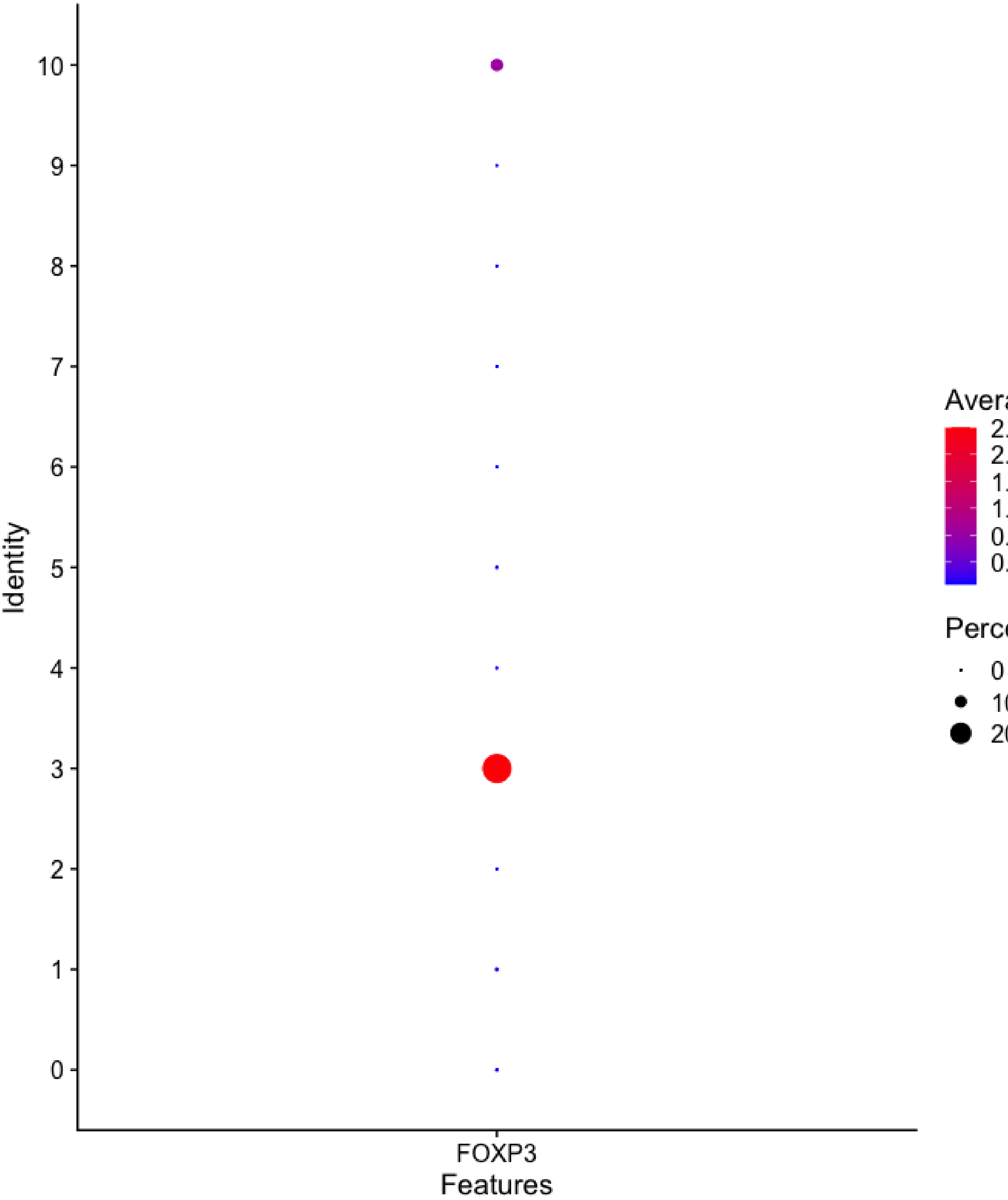
```
DoHeatmap(NewLymphocyteClusters, features = top10markers$gene)
```


0
1
2
3
4
5
6
7
8
S100A4
CD40LG
CAPG
ANKRD28
CCR5
LGALS3
PHLDA1
LMO4
CLU
CTSH
CCR7
IL7R
SESN3
KLF2
AP3M2
CD55
SELL
HIST1H1D
TSHZ2
SC5D
CCL5
GZMK
CCL4
CCL4L2
CD8A
GZMH
CRTAM
CCL3L1
IFNG
HLA-DQA1
FOXP3
CTLA4
IKZF2
IL2RA
GBP5
HPGD
TTN
RTKN2
TNFRSF9
MIR4435-2HG
HSPD1
FKBP4
HSPH1
PLIN2
SLFN11
AMPD3
PLAUR
HSPE1
MT2A
HIST1H1E
XCL1
XCL2
FCER1G
TYROBP
KLRC1
TRDC
SPINK2
GNLY
SPRY1
CSF2
FXYD2
ZNF683
KLRC4
PDGFA
TRGC2
CRIP1
EMP1
HBA2
MTSS1
TESC
JCHAIN
IGHG1
IGHG3
IGHA1
IGHG2
IGKV4-1
IGKC
IGHGP
IGLC3
IGLC2
HSPA6
DNAJA4
DNAJB4
SERPINE1
HSPA1B
DDX3Y
TNFSF9
ZFAND2A
BAG3
FGFBP2
FCGR3A
KLRF1
SPON2
CX3CR1
S1PR5
GZMB
AKR1C3
SH2D1B
MYOM2
CXCL13
IL17A
ZBED2
LAG3
IL26
BATF3
AC017002.3
LINC01943
ENTPD1
DCP1B

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"))
```


FOXP3
Expression Level
3
2
1
0
Identity
0
1
2
3
4
5
6
7
8
9

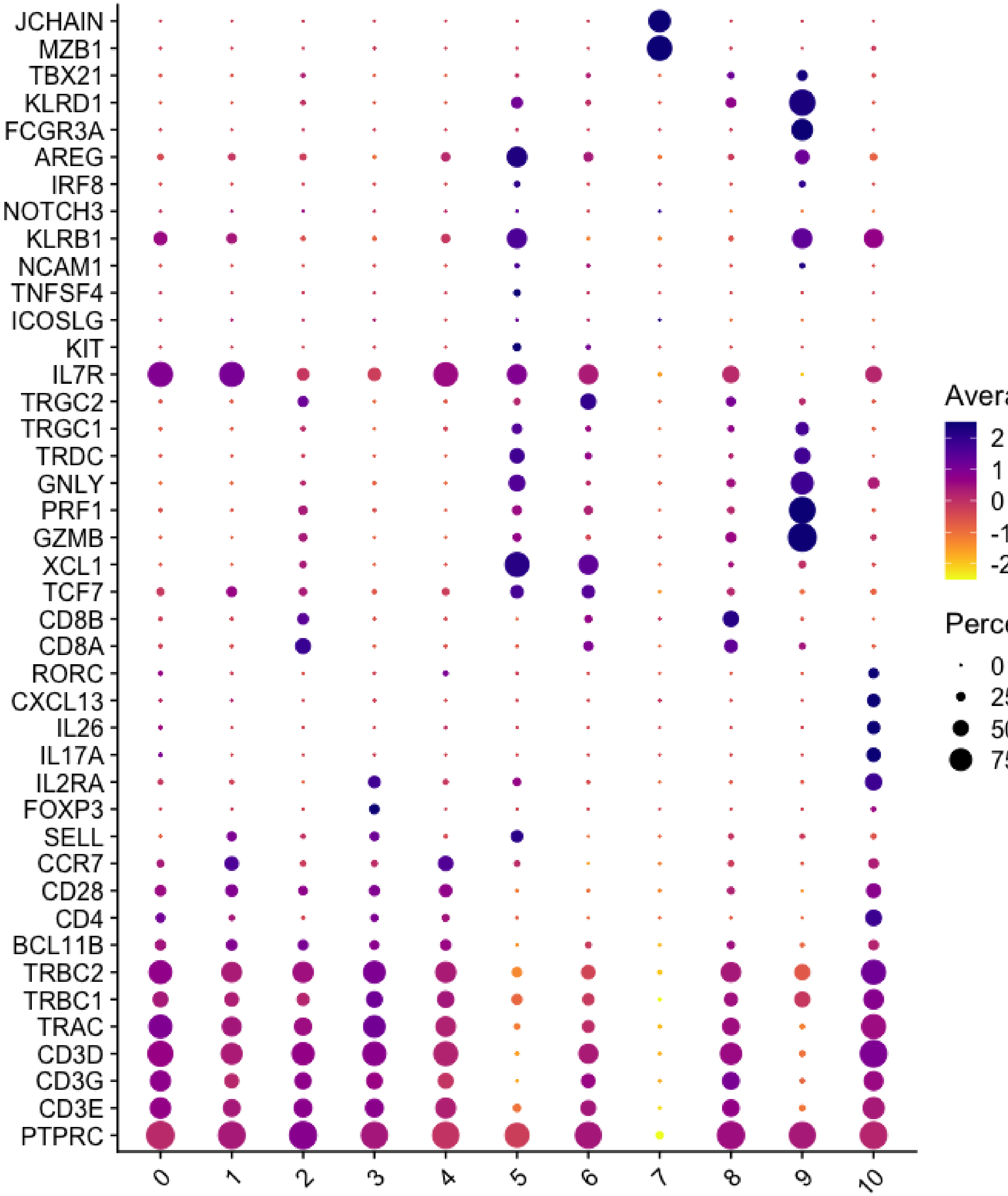

10
9
8
7
6
5
4
3
2
1
0
Identity
FOXP3
Features
Avera
2.
2.
1.
1.
0.
0.
Perc
0
10
20

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",

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

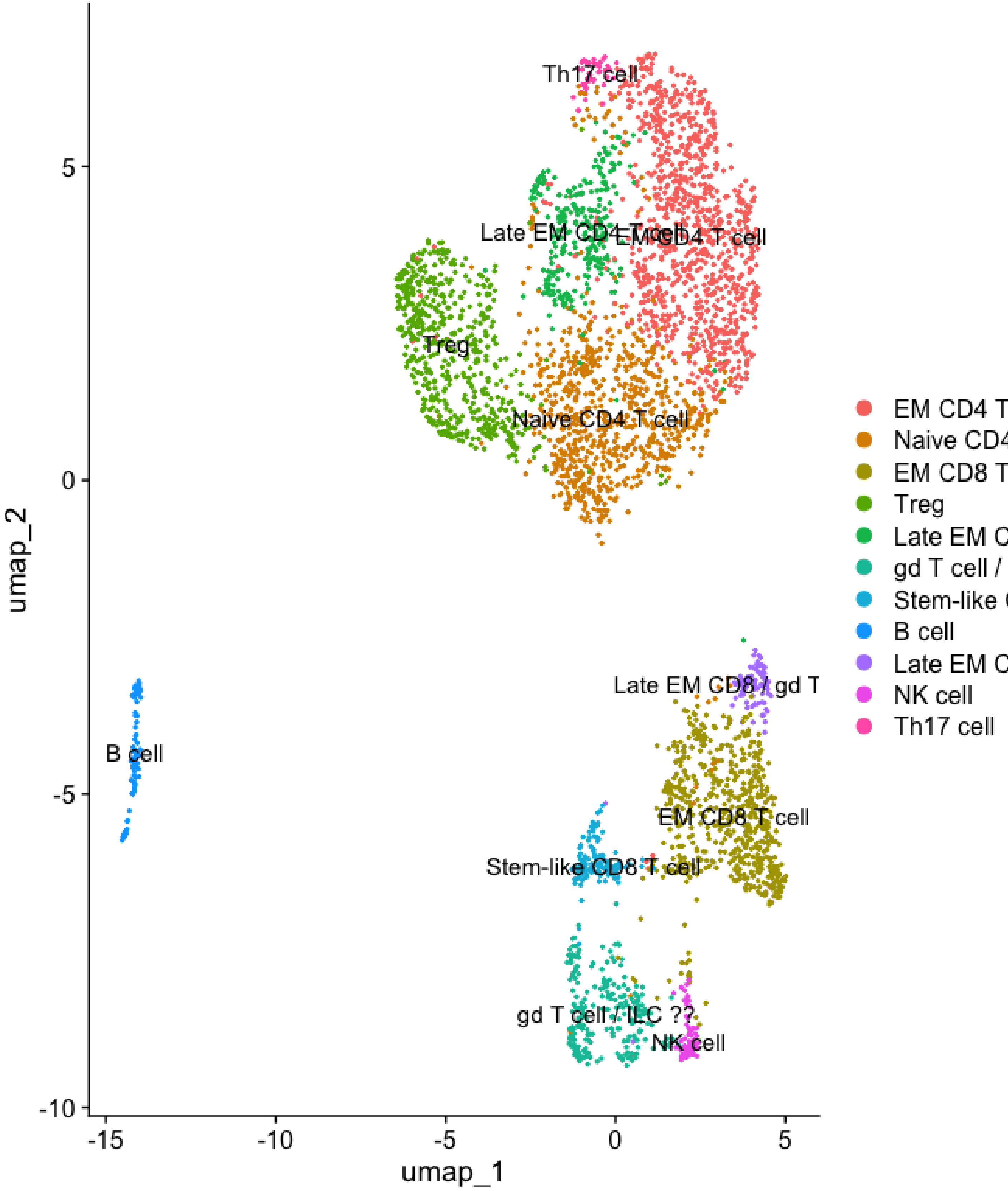

JCHAIN
MZB1
TBX21
KLRD1
FCGR3A
AREG
IRF8
NOTCH3
KLRB1
NCAM1
TNFSF4
ICOSLG
KIT
IL7R
TRGC2
TRGC1
TRDC
GNLY
PRF1
GZMB
XCL1
TCF7
CD8B
CD8A
RORC
CXCL13
IL26
IL17A
IL2RA
FOXP3
SELL
CCR7
CD28
CD4
BCL11B
TRBC2
TRBC1
TRAC
CD3D
CD3G
CD3E
PTPRC
0
1
2
3
4
5
6
7
8
9
10
Avera
2
1
0
-1
-2
Perc
0
25
50
75

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

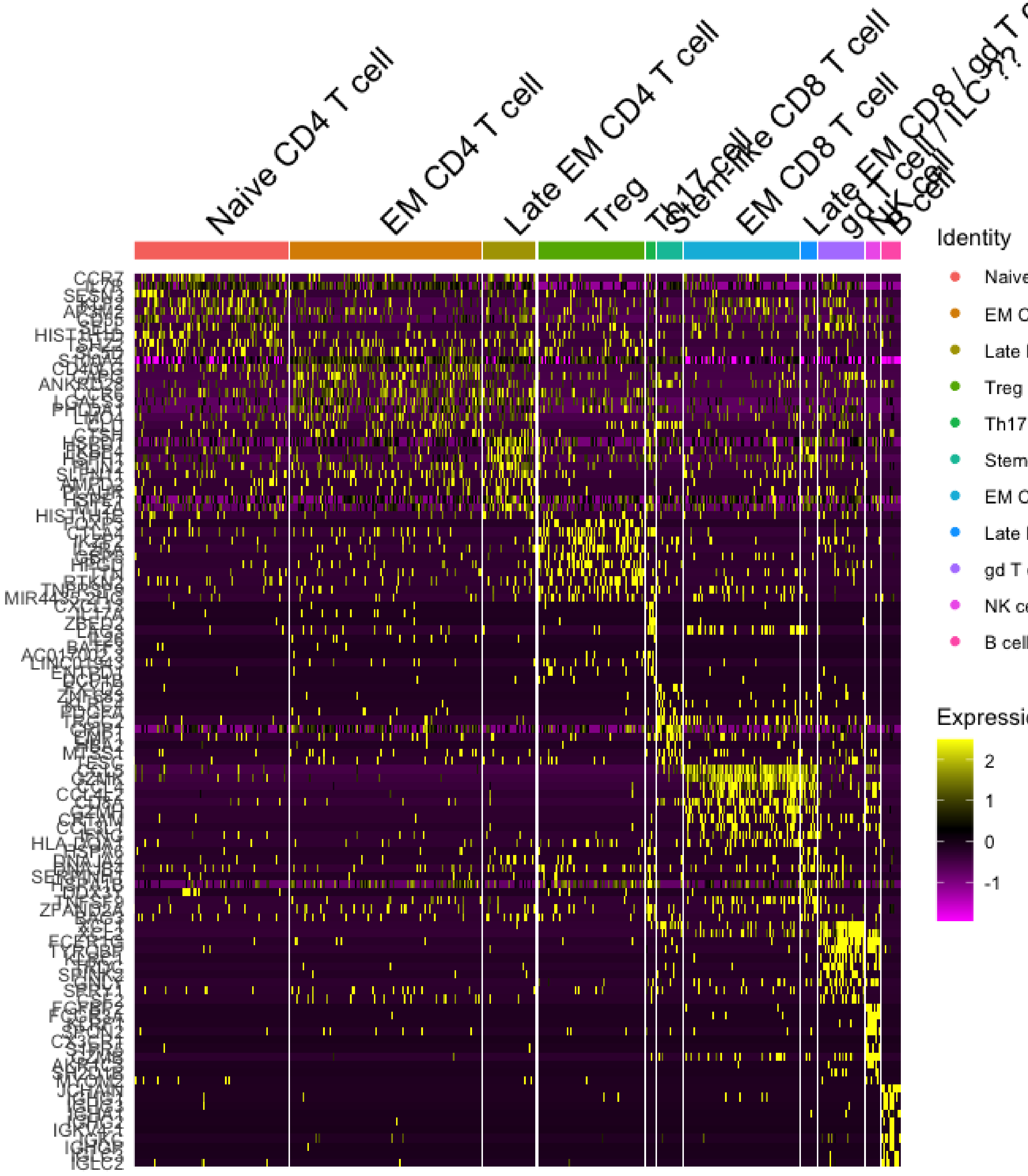
Now I've found out what each cluster may be based on their specific genes. It's time to rename the clusters with their correct cell type.

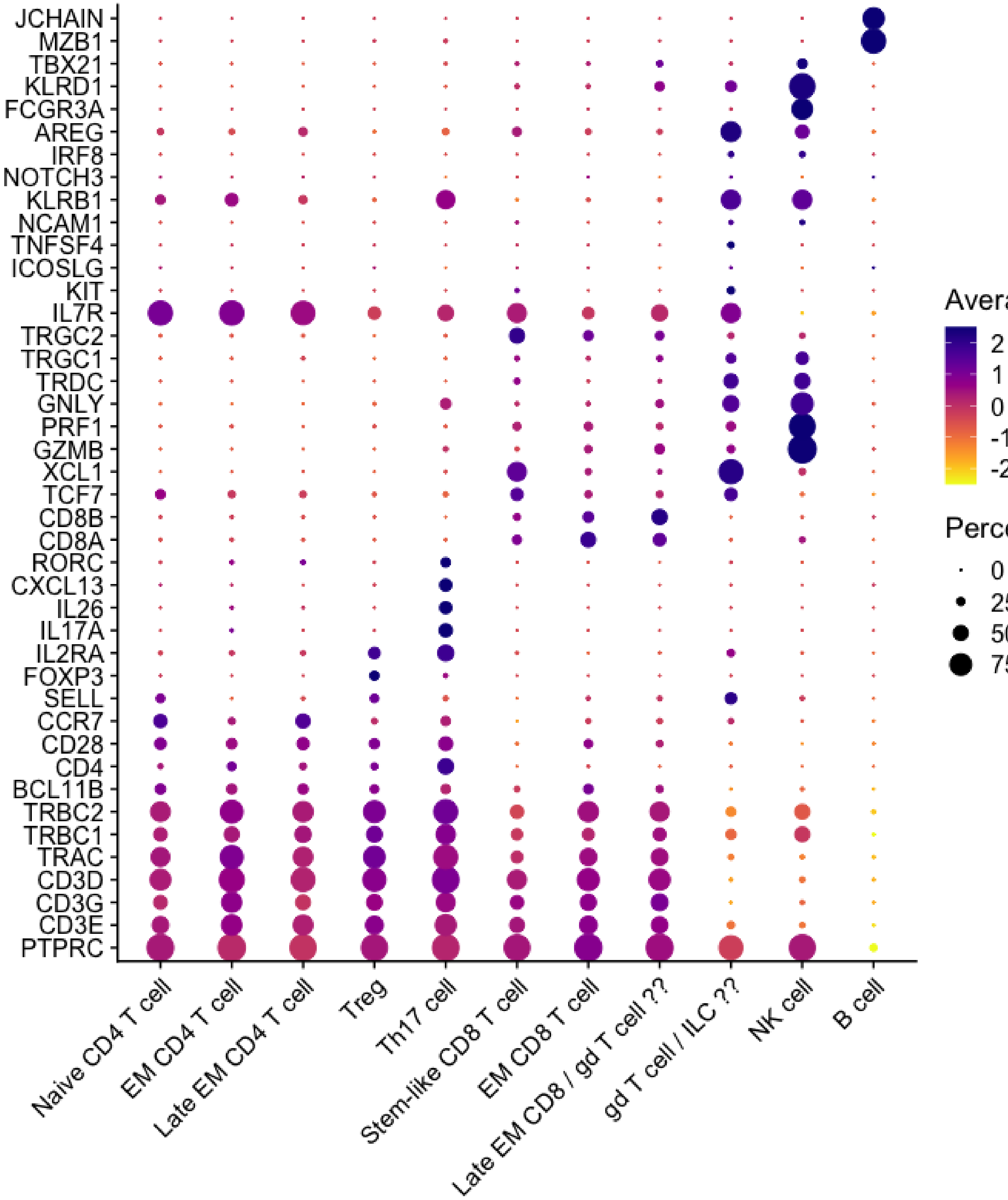
```
new.cluster.ids <- c("EM CD4 T cell", "Naive CD4 T cell", "EM CD8 T cell", "Treg", "Late EM CD4 T cell"

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)
```


Th17 cell
Late EM CD4 T cell
EM CD4 T cell
Treg
Naive CD4 T cell
Late EM CD8 / gd T
B cell
EM CD8 T cell
Stem-like CD8 T cell
gd T cell / ILC ??
NK cell
umap_2
umap_1
5
0
-5
-10
-15
-10
-5
0
5
EM CD4 T
Naive CD4
EM CD8 T
Treg
Late EM C
gd T cell /
Stem-like C
B cell
Late EM C
NK cell
Th17 cell

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.

```r
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",
                                                        "B cell")
                                             )

NewLymphocyteClusters$celltype <- Idents(NewLymphocyteClusters)

# Verify each cluster and their top genes in an updated Heatmap that shows all the renamed clusters. We

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

top10markers <- cluster_markers %>%
  group_by(cluster) %>%
  top_n(n = 10, wt = avg_log2FC)

DoHeatmap(NewLymphocyteClusters, features = top10markers$gene)

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


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 c
gd T cell
NK cell / ILC ??
B cell
Identity
Naive
EM C
Late
Treg
Th17
Stem
EM C
Late
gd T c
NK ce
B cell
Expressi
2
1
0
-1
CCR7
SESN3
CD40LG
ANKRD28
CXCL13
ZBED2
GZMK
CRTAM
FCER1G
TYROBP
JCHAIN
IGLC2


JCHAIN
MZB1
TBX21
KLRD1
FCGR3A
AREG
IRF8
NOTCH3
KLRB1
NCAM1
TNFSF4
ICOSLG
KIT
IL7R
TRGC2
TRGC1
TRDC
GNLY
PRF1
GZMB
XCL1
TCF7
CD8B
CD8A
RORC
CXCL13
IL26
IL17A
IL2RA
FOXP3
SELL
CCR7
CD28
CD4
BCL11B
TRBC2
TRBC1
TRAC
CD3D
CD3G
CD3E
PTPRC
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
Avera
2
1
0
-1
-2
Perce
0
25
50
75

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.

“‘