Organ Annotation Vignette

Preprocess

Enter the name of the organ you want to analyze.

```
organ_of_interest = "Liver"
```

We begin by loading a gene by cell matrix A, where A_{ij} is the number of reads (or UMIs) from cell j that aligned to gene i. The Seurat package keeps track of matrices and coordinates derived from A together with metadata; that information is stored in a large object called `tiss`. For the sake of explicitness, we describe below many of the mathematical transformations implemented inside of Seurat.

First, we filter out cells with fewer than 500 genes or 50,000 reads. (For UMI data, we filter out cells with fewer than 500 genes or 1,000 UMIs.)

```
tiss <- FilterCells(object = tiss, subset.names = c("nGene", "nReads"),
                    low.thresholds = c(500, 50000))
```

Next, we log-normalize counts for each cell, essentially to log counts per million (for reads) or counts per ten thousand (UMIs). More precisely, we set

$$N_{ij} = \log\left(1 + M\frac{A_{ij}}{\sum_{j'} A_{ij'}}\right),$$

where $M = 10^6$ for FACS and $M = 10^4$ for droplets. The log is base e. We chose those values of M to be close to the average number of counts per cell.

```
tiss <- NormalizeData(object = tiss, scale.factor = 1e6)
```

Then we shift and scale the rows of the normalized matrix, so each gene has mean zero and variance one.

$$X_{ij} = (N_{ij} - \mu_i)/\sigma_i,$$

where μ_i is the mean of N_{ij} and σ_i is the standard deviation of N_{ij}.

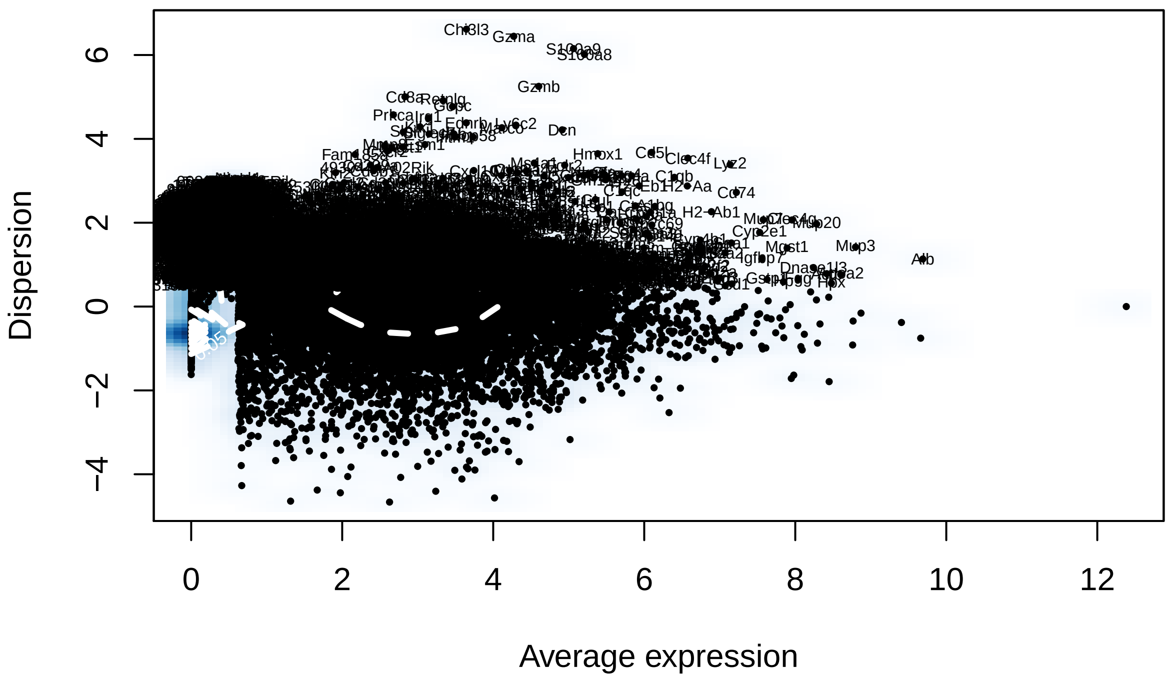
```
tiss <- ScaleData(object = tiss)
```

```
## [1] "Scaling data matrix"
##
  |
  |                                                                 |   0%
  |
  |=================================================================| 100%
```

We select variable genes (those with high standardized log dispersion, given their mean), as they will be more informative about differences between cell types in the population.

Concretely, the log dispersion d_i of a gene i is $d_i = \log(v_i/m_i)$, where v_i is the variance of $\exp(N_{ij})$ and m_i is the mean of $\exp(N_{ij})$. (In other words, m_i is 1 plus CPM.) We bin the genes into 20 equal-spaced bins based on $\log(m_i)$, then compute the mean and standard deviation of d_i within each bin. The standardized log dispersion $\bar{d}_i$ is the dispersion d_i shifted by the mean and rescaled by the standard deviation of the d_k within its bin. We retain genes with $\bar{d}_i > 0.5$ and $\log(m_i) > 0.1$.

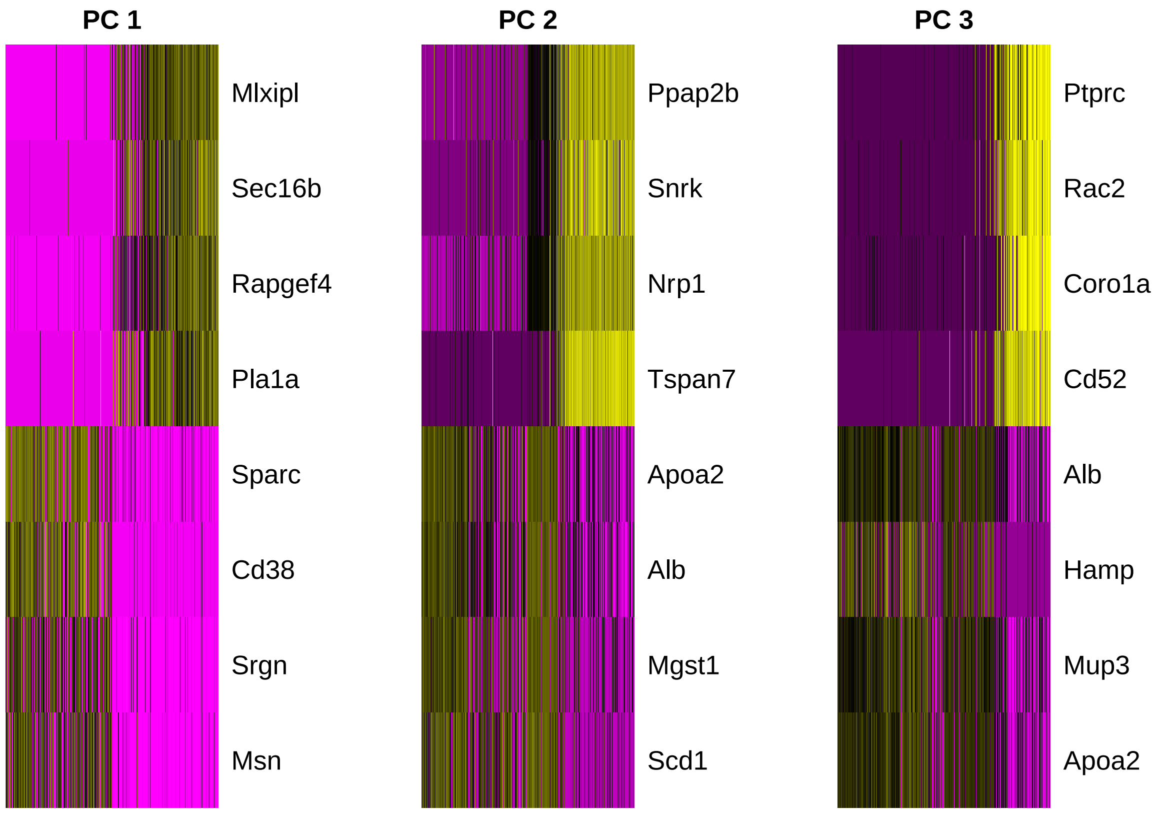
```
tiss <- FindVariableGenes(object = tiss, do.plot = TRUE,
                          x.high.cutoff = Inf, y.cutoff = 0.5, x.low.cutoff = 0.1)
```



Cells are projected onto a low-dimensional subspace using principal component analysis on the scaled expression X of the variable genes.

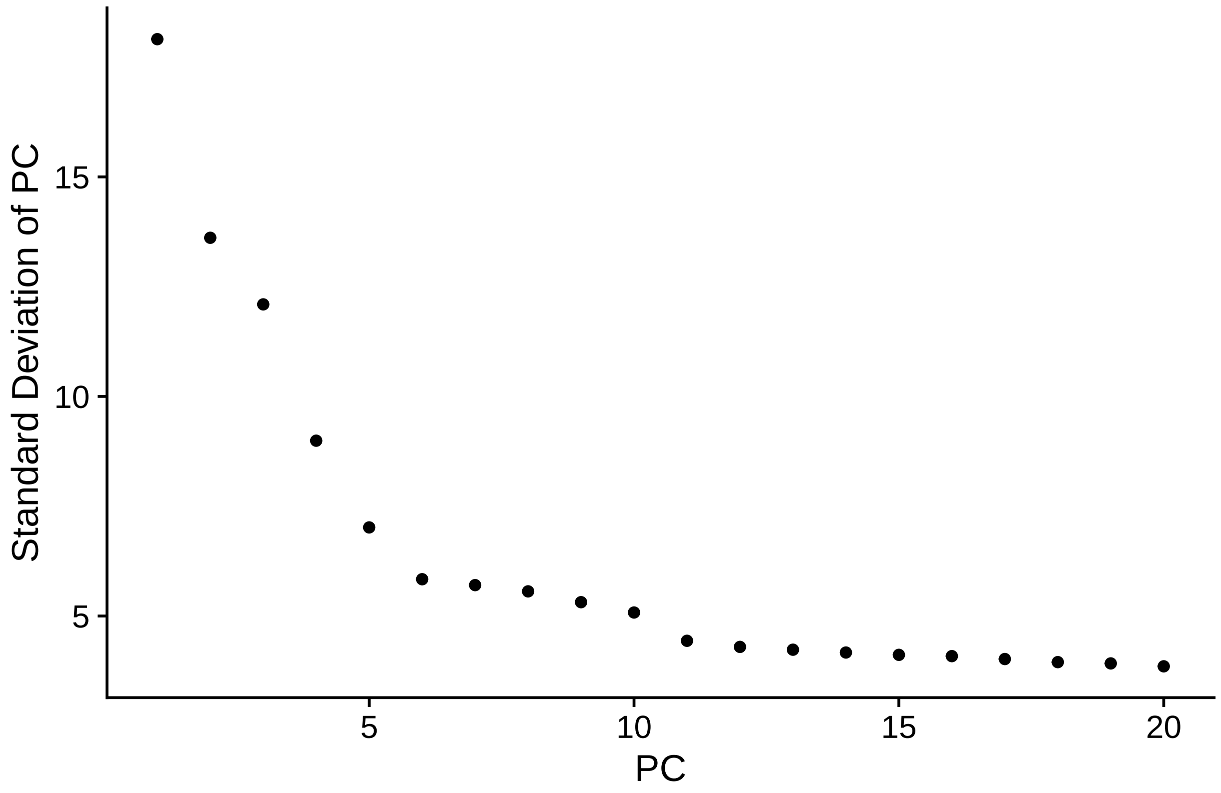
```
tiss <- RunPCA(object = tiss, do.print = FALSE)
tiss <- ProjectPCA(object = tiss, do.print = FALSE)
```

We can visualize top genes in each principal component.



We then project onto just the top principal components. This has the effect of keeping the major directions of variation in the data and, ideally, supressing noise. A decent rule of thumb is to pick the elbow in the plot below.

```r
PCElbowPlot(object = tiss)
```



Choose the number of principal components to use.

```r
n.pcs = 11
```

Cluster

The clustering is performed on a shared-nearest-neighbors graph on the cells.

The shared-nearest-neighbors graph was constructed based on the Euclidean distance in the low-dimensional subspace; cells are connected if their k-neighborhoods overlap. Indeed, let $\mathcal{N}(j)$ denote the k-nearest neighborhood (k = 30) for a cell j. The shared-nearest-neighbor graph G has a vertex for each cell and an edge of weight

$$w_{jk} = \frac{|\mathcal{N}(j) \cap \mathcal{N}(k)|}{|\mathcal{N}(j) \cup \mathcal{N}(k)|}$$

between cells j and k. Cells were clustered using a modified version of the Louvain method for modularity maximization. The modularity has a resolution parameter γ,

$$Q = \sum_{ij} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta(c_i, c_j),$$

where A_{ij} is the weighted adjacency matrix, k_i and k_j are the weighted degrees of cells i and j, m is the total weight of edges in the graph, c_i denotes cluster membership, and $\delta(c_i, c_j)$ is 1 if i and j are in the same cluster, and 0 otherwise.

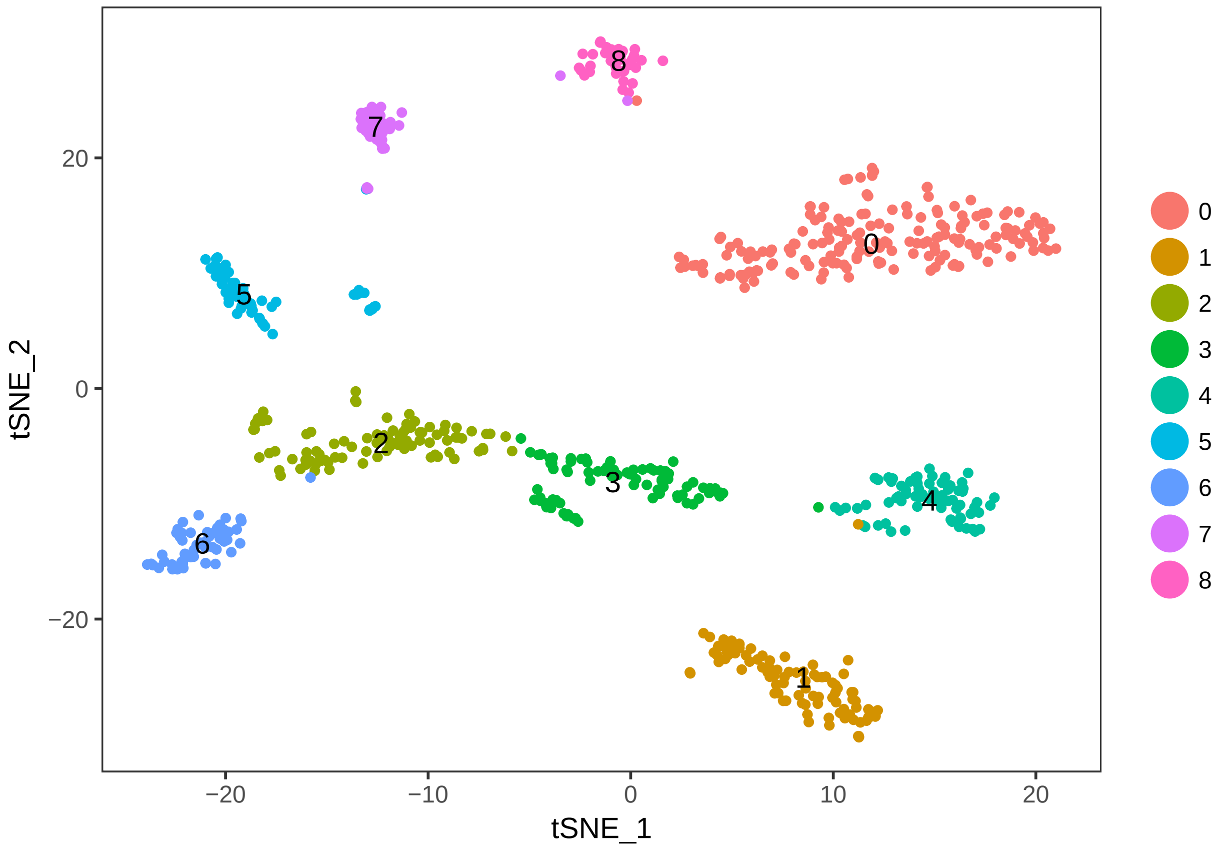
The resolution γ is a tuneable parameter in this analysis that sets the tradeoff between in-group connections and between-group connections. High resolution favors smaller clusters.

```
# Set resolution
res.used <- 1

tiss <- FindClusters(object = tiss, reduction.type = "pca", dims.use = 1:n.pcs,
    resolution = res.used, print.output = 0, save.SNN = TRUE)
```

We use tSNE solely to visualize the data.

```
tiss <- RunTSNE(object = tiss, dims.use = 1:n.pcs, seed.use = 10, perplexity=30)
```

```
TSNEPlot(object = tiss, do.label = T, pt.size = 1.2, label.size = 4)
```

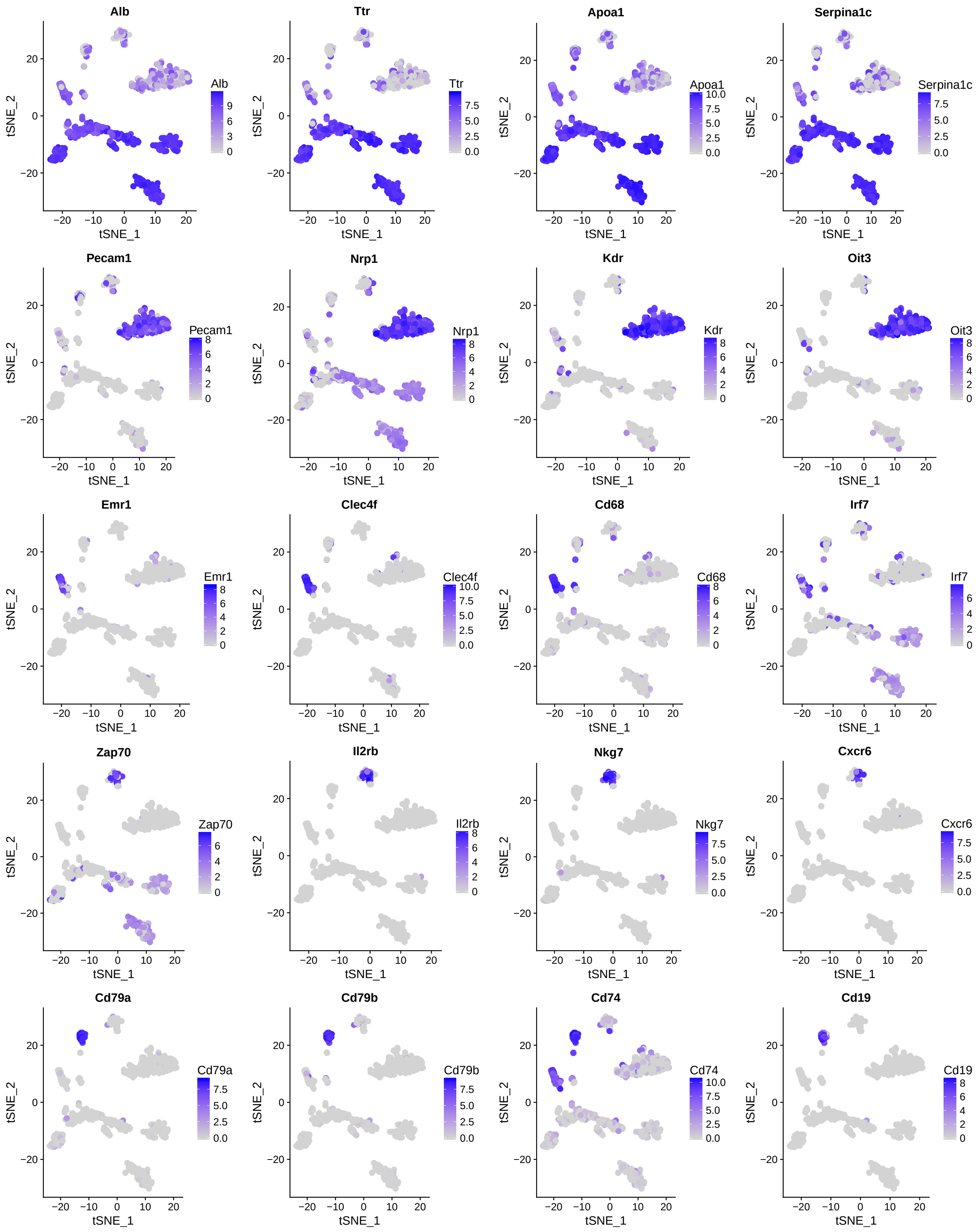


Label clusters using marker genes

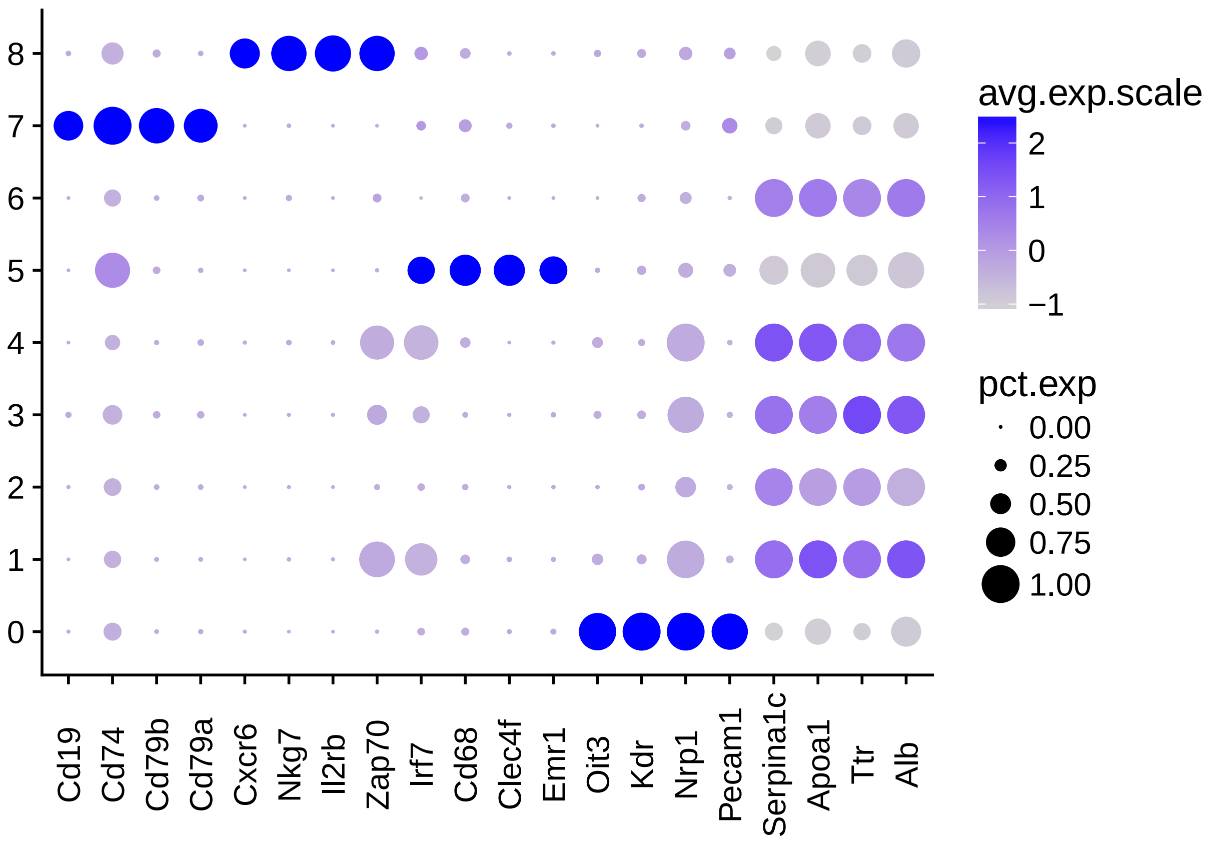
Check expression of genes useful for indicating cell type.

```
genes_hep = c('Alb', 'Ttr', 'Apoa1', 'Serpina1c') #hepatocyte
genes_endo = c('Pecam1', 'Nrp1', 'Kdr','Oit3') # endothelial
genes_kuppfer = c('Emr1', 'Clec4f', 'Cd68', 'Irf7') # Kuppfer cells
genes_nk = c('Zap70', 'Il2rb', 'Nkg7', 'Cxcr6') # Natural Killer cells
genes_b = c('Cd79a', 'Cd79b', 'Cd74', 'Cd19') # B Cells

genes_all = c(genes_hep, genes_endo, genes_kuppfer, genes_nk, genes_b)
```

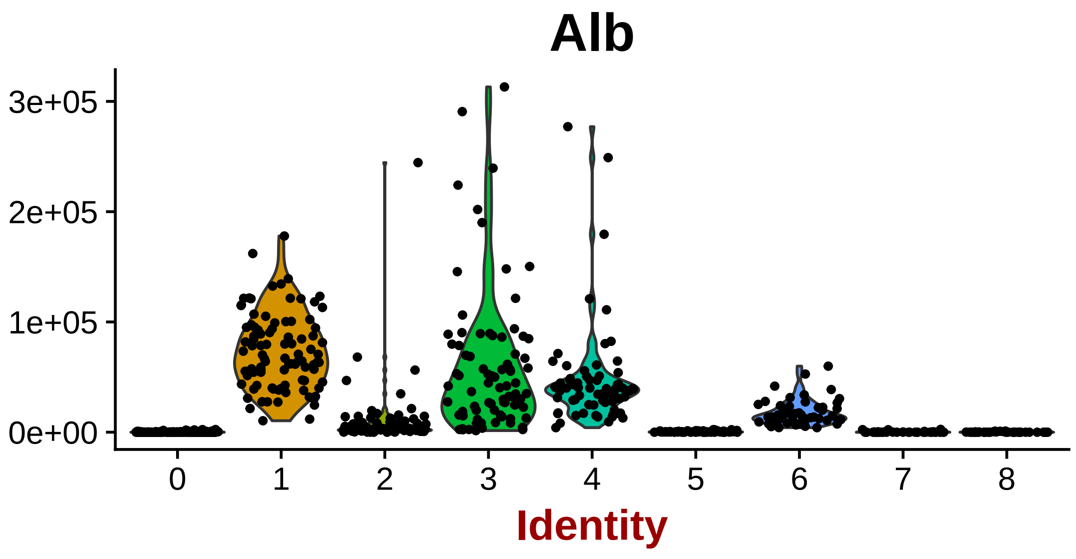
In the tSNE plots below, the intensity of each point represents the log-normalized gene expression N_{ij}.



Dotplots show, for each cluster and gene, the fraction of cells with at least one read for the gene (circle size) and the average scaled expression for that gene among the cells expressing it (circle color).



The low but nonzero levels of Albumin present in all clusters is consistent with a small amount of leakage, either through physical contamination or index hopping. Nevertheless, the absolute levels of expression confirm a sharp difference between the hepatocyte clusters and the others.



To confirm the identity of a cluster, you can inspect the genes differentially expressed in that cluster compared to the others.

```
clust.markers7 <- FindMarkers(object = tiss, ident.1 = 7,
                              only.pos = TRUE, min.pct = 0.25, thresh.use = 0.25)
```

The top markers for cluster 7 include histocompatibility markers H2-*, consistent with the expression of other

B-cell markers seen above.

```r
head(clust.markers7)
```

```
##                 p_val avg_logFC pct.1 pct.2    p_val_adj
## H2-DMb2 9.768214e-101  7.042620 0.902 0.024 2.279999e-96
## Pou2af1 1.585456e-100  6.108999 0.732 0.006 3.700612e-96
## Spib     1.271751e-88  6.278530 0.707 0.010 2.968394e-84
## Cd19     1.240657e-87  7.018641 0.756 0.016 2.895818e-83
## H2-Oa    6.025345e-85  5.833053 0.683 0.010 1.406376e-80
## H2-Ob    1.236266e-83  5.860218 0.780 0.022 2.885569e-79
```

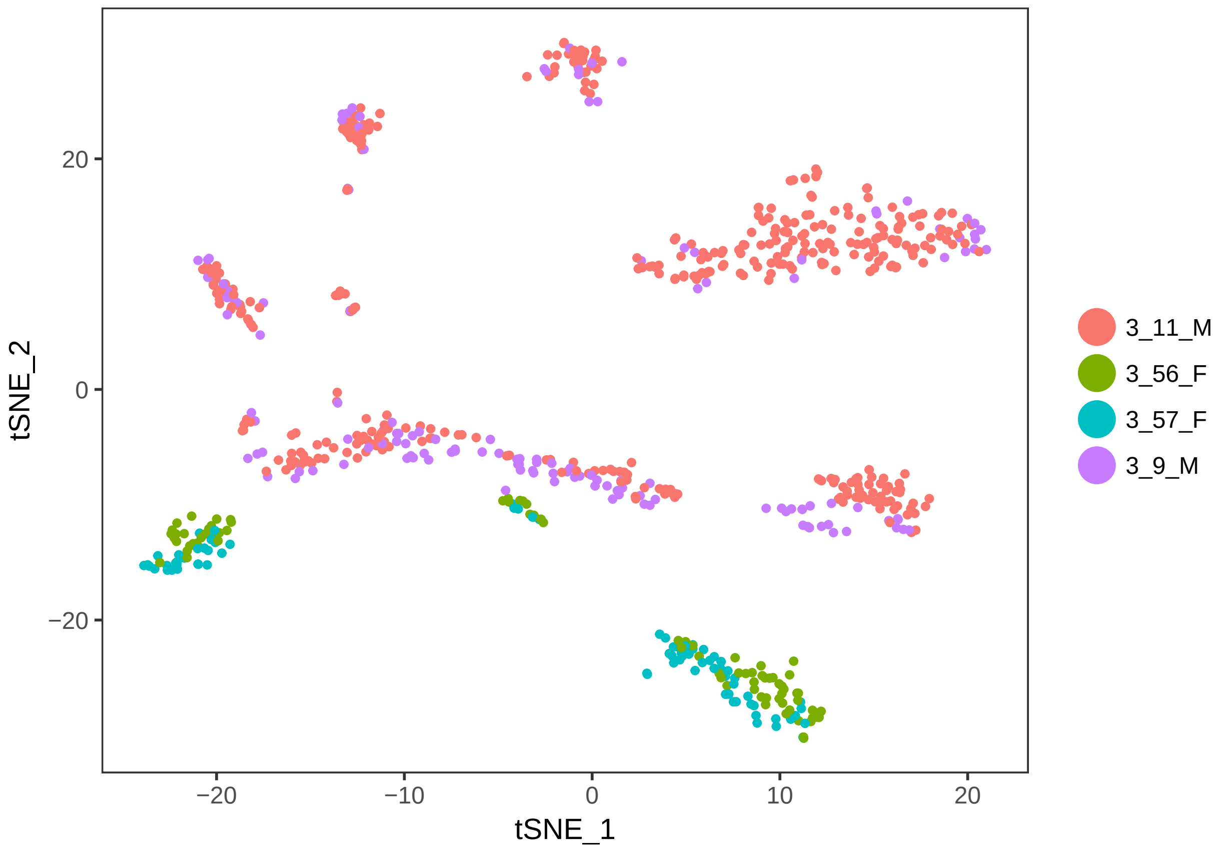
Using the markers, we can confidentaly label the clusters. We provide both a free annotation (where any name can be used) and a cell ontology class. The latter uses a controlled vocabulary for easy comparison between studies and different levels of the taxonomy.

```r
tiss <- StashIdent(object = tiss, save.name = "cluster.ids")

cluster.ids <- c(0, 1, 2, 3, 4, 5, 6, 7, 8)

free_annotation <- c(
  "endothelial cell",
  "hepatocyte",
  "hepatocyte",
  "hepatocyte",
  "hepatocyte",
  "kuppfer",
  "hepatocyte",
  "B cell",
  "NK/NKT cells")

cell_ontology_class <-c(
  "endothelial cell of hepatic sinusoid",
  "hepatocyte",
  "hepatocyte",
  "hepatocyte",
  "hepatocyte",
  "Kupffer cell",
  "hepatocyte",
  "B cell",
  "natural killer cell")

tiss = stash_annotations(tiss, cluster.ids, free_annotation, cell_ontology_class)
```

Checking for batch effects

Color by metadata, like plate barcode, to check for batch effects. Here we see that the clusters are segregated by sex.

```
TSNEPlot(object = tiss, do.return = TRUE, group.by = "mouse.id")
```



Nevertheless, every cluster contains cells from multiple mice.

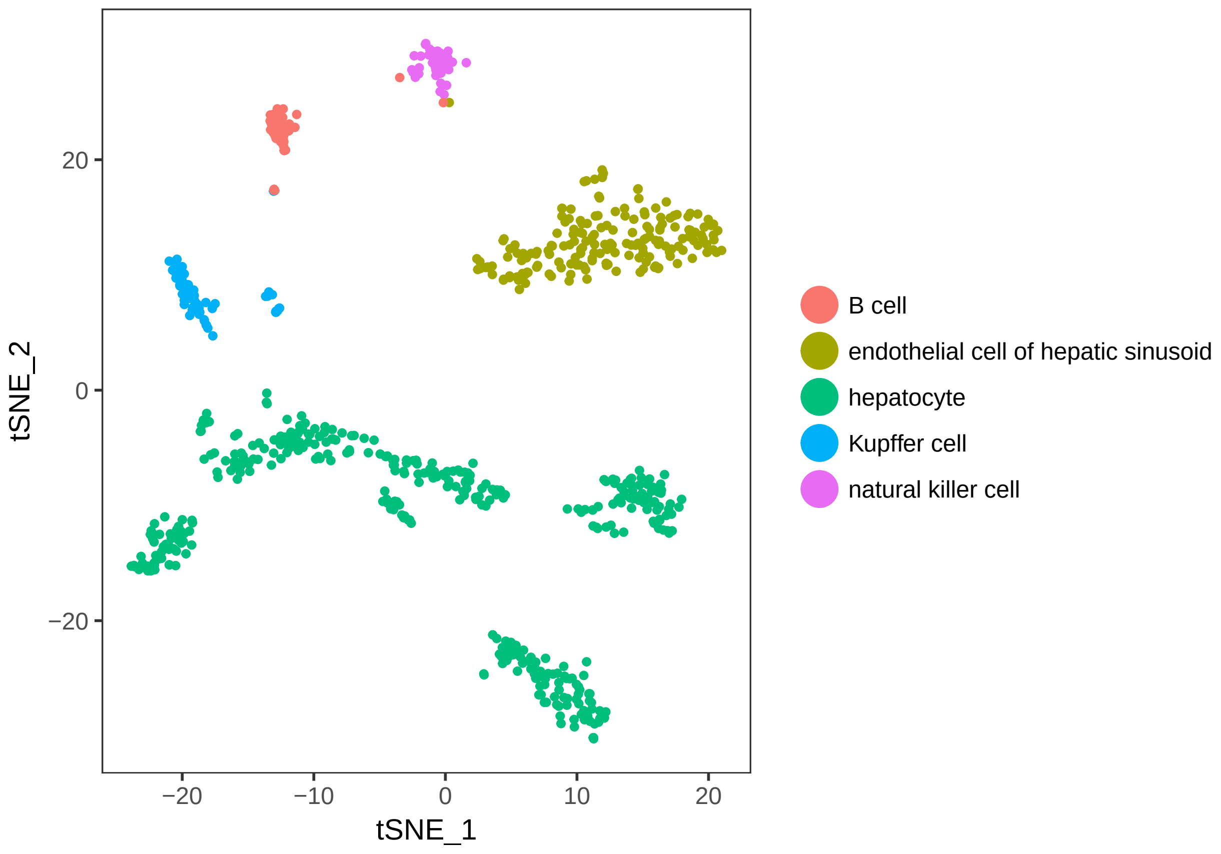
```
table(FetchData(tiss, c('mouse.id','ident')) %>% droplevels())
```

```
##          ident
## mouse.id   0   1   2   3   4   5   6   7   8
##   3_11_M 160   0  59  29  51  47   0  31  32
##   3_56_F   0  46   0  11   0   0  25   0   0
##   3_57_F   0  46   0   5   0   0  28   0   0
##   3_9_M   22   1  32  37  20  14   1  10   7
```

Final coloring

Color by cell ontology class on the original tSNE.

```
TSNEPlot(object = tiss, group.by = "cell_ontology_class")
```



Save the Robject for later

```
#filename = here('00_data_ingest', '04_tissue_robj_generated',
#                      paste0("facs_", organ_of_interest, "_seurat_tiss.Robj"))
#print(filename)
#save(tiss, file=filename)
```

```
# To reload a saved object
#filename = here('00_data_ingest', '04_tissue_robj_generated',
#                       paste0("facs_", organ_of_interest, "_seurat_subtiss.Robj"))
#load(file=filename)
```

Export the final metadata

```
#save_annotation_csv(tiss, organ_of_interest, "facs")
```