

Malignant Melanocytes Show Significant Upregulation of Class II Human Leukocytic Antigens

M. Gage

2025-08-14

Document Description

This document includes the applicable code, plots, and written assessments for scRNA-seq analysis using the Seurat-Guided Clustering Tutorial Vignette. The “10k Human DTC Melanoma, Chromium GEM-X Single Cell 3” data set was extracted from 10X Genomics for this workflow. This analysis was used to determine cell type clustering and to observe the distribution of HLA Class II genes across clusters (particularly the B Cell and Melanocyte clusters).

Read in Data (Setup the Seurat Object)

```
# Load in the appropriate packages
library(dplyr)
library(Seurat)
library(patchwork)
library(hdf5r)
library(SingleR)

# Load the dataset (using the .h5 file and Read10X_h5 function to conserve space
# locally)
melanoma.data <- Read10X_h5("melanoma_filtered_feature_bc_matrix.h5")

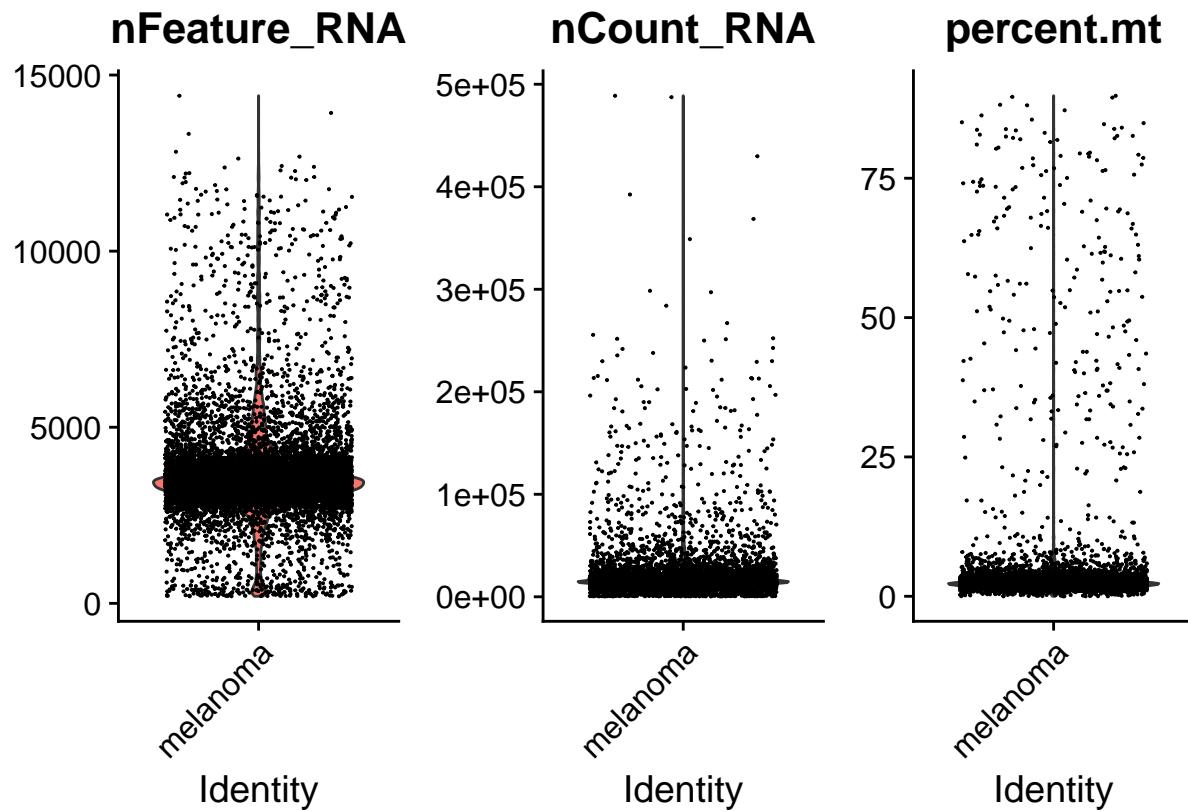
# Initialize the Seurat object with the raw (non-normalized data).
melanoma <- CreateSeuratObject(counts = melanoma.data,
                                 project = "melanoma",
                                 min.cells = 3,
                                 min.features = 200)
melanoma

# An object of class Seurat
# 29104 features across 10367 samples within 1 assay
# Active assay: RNA (29104 features, 0 variable features)
# 1 layer present: counts
```

Standard Pre-Processing Workflow

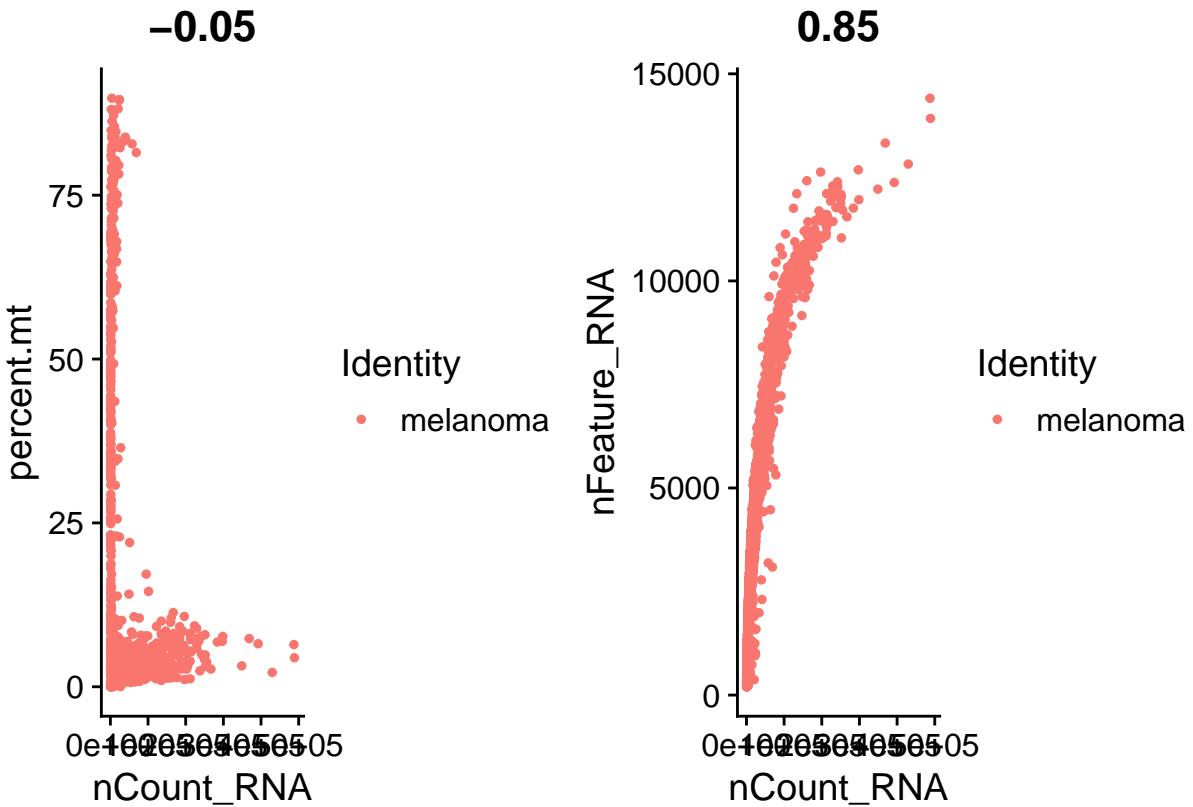
```
# The [[ operator can add columns to object metadata. This is a great place to stash QC
# stats. Here, we are creating the percentage of reads mapping to the mitochondrial
# genome. Human data uses "MT-", however, mouse data uses "mt-" to identify a set of
# mitochondrial genes. Low-quality/dying cells often have extensive mitochondrial
# contamination.
melanoma[["percent.mt"]] <- PercentageFeatureSet(melanoma, pattern = "^MT-")

# Visualize QC metrics as a violin plot. Here, we are comparing three key QC metrics:
# Number of genes detected per cell (nFeature_RNA), Total RNA molecules counts per
# cell (nCount_RNA), and percent of reads mapped to mitochondrial genes (percent.mt)
VlnPlot(melanoma, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```



```
# FeatureScatter is typically used to visualize feature-feature relationships, but can be
# used for anything calculated by the object, i.e. columns in object metadata, PC scores etc.
# Here, we see a very strong positive correlation (0.85) between nFeature and nCount and almost
# no correlation between percent.mt and nCount.
```

```
plot1 <- FeatureScatter(melanoma, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(melanoma, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
plot1 + plot2
```



```
# Removing unwanted cells based on predetermined thresholds from each feature. This allows us
# to omit outliers or doublet/multiplet cells captured together in scRNA-seq processing
melanoma <- subset(melanoma, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 12)
```

Normalize the Data

```
# Normalizing the data. "LogNormalize" normalizes the feature expression measurements for each cell
# by the total expression, multiplies this by a scale factor (10,000 by default), and log-transforms
# the result. This method relies on an assumption that each cell originally contains the same number
# of RNA molecules.
melanoma <- NormalizeData(melanoma)
```

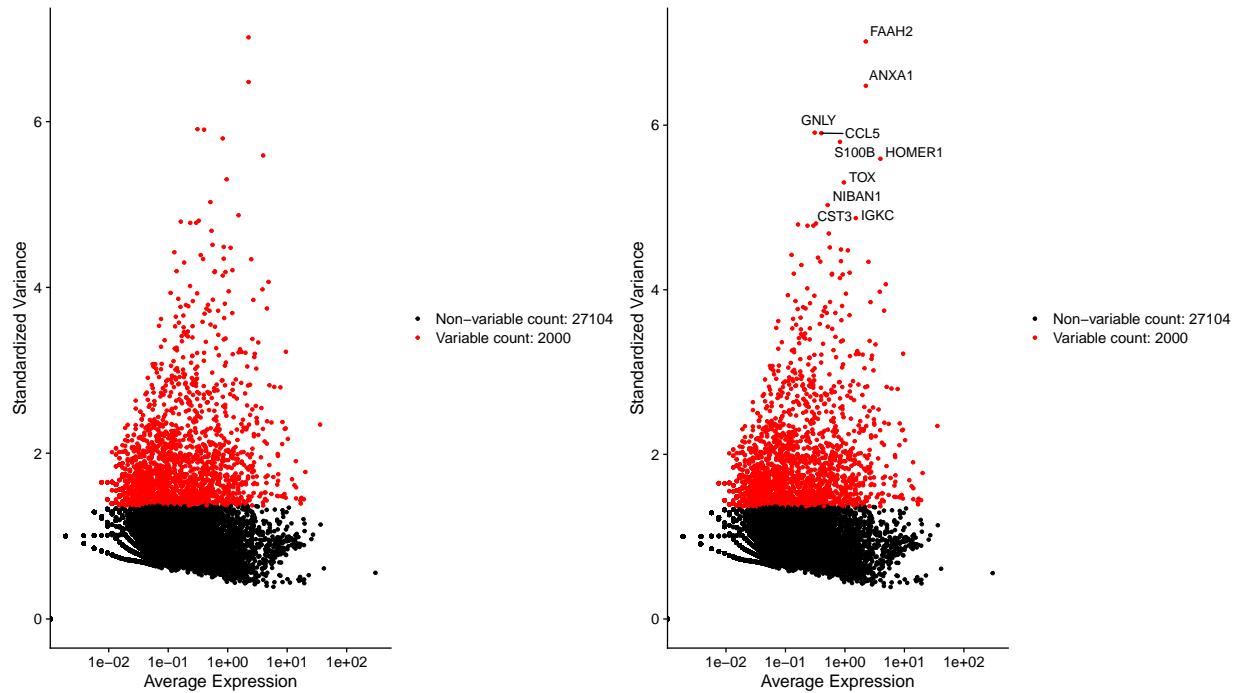
Feature Selection

```
# Finding variable features. This calculates a subset of features that have high cell-to-cell
# variation (highly expressed in some cells, and lowly expressed in others).
melanoma <- FindVariableFeatures(melanoma, selection.method = "vst", nfeatures = 2000)

# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(melanoma), 10)
```

```
# Plot variable features with and without labels
plot1 <- VariableFeaturePlot(melanoma)
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)

plot1+plot2
```



Scaling the Data

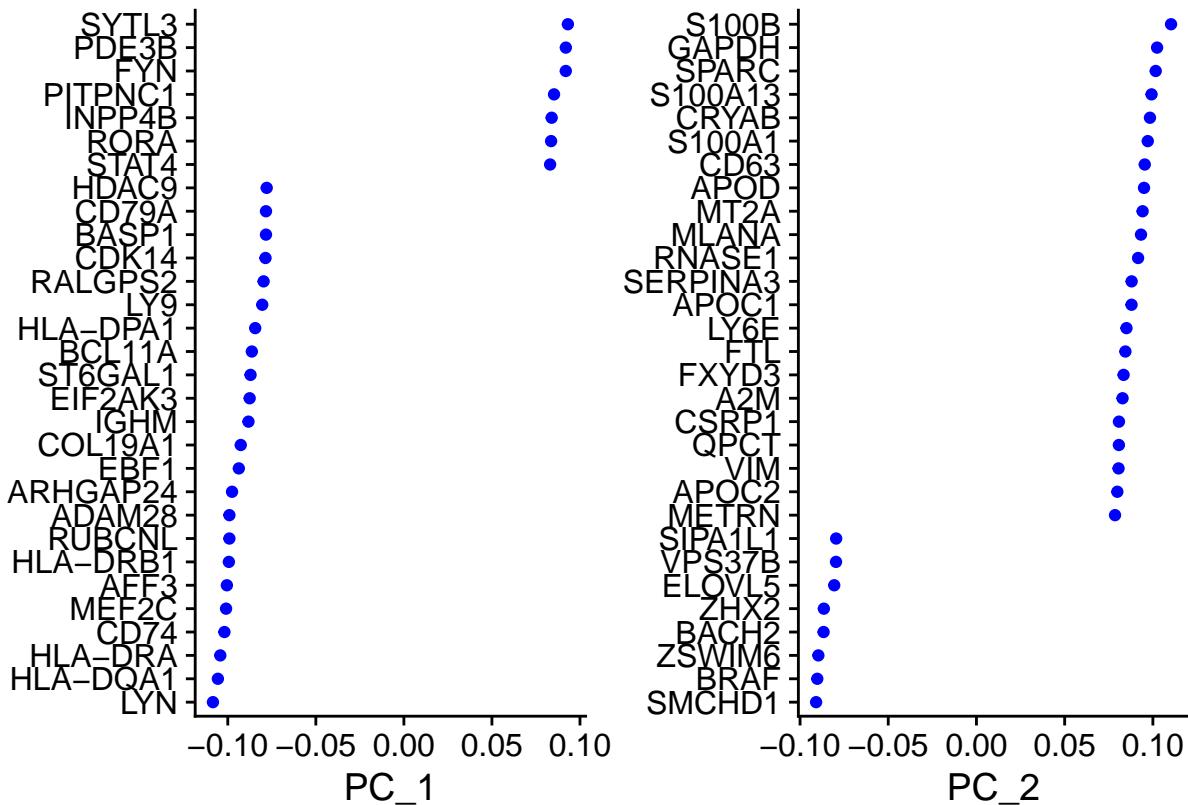
```
# Scaling the data by applying a linear transformation prior to dimensional reduction
all.genes <- rownames(melanoma)
melanoma <- ScaleData(melanoma, features = all.genes)
```

Linear Dimensional Reduction

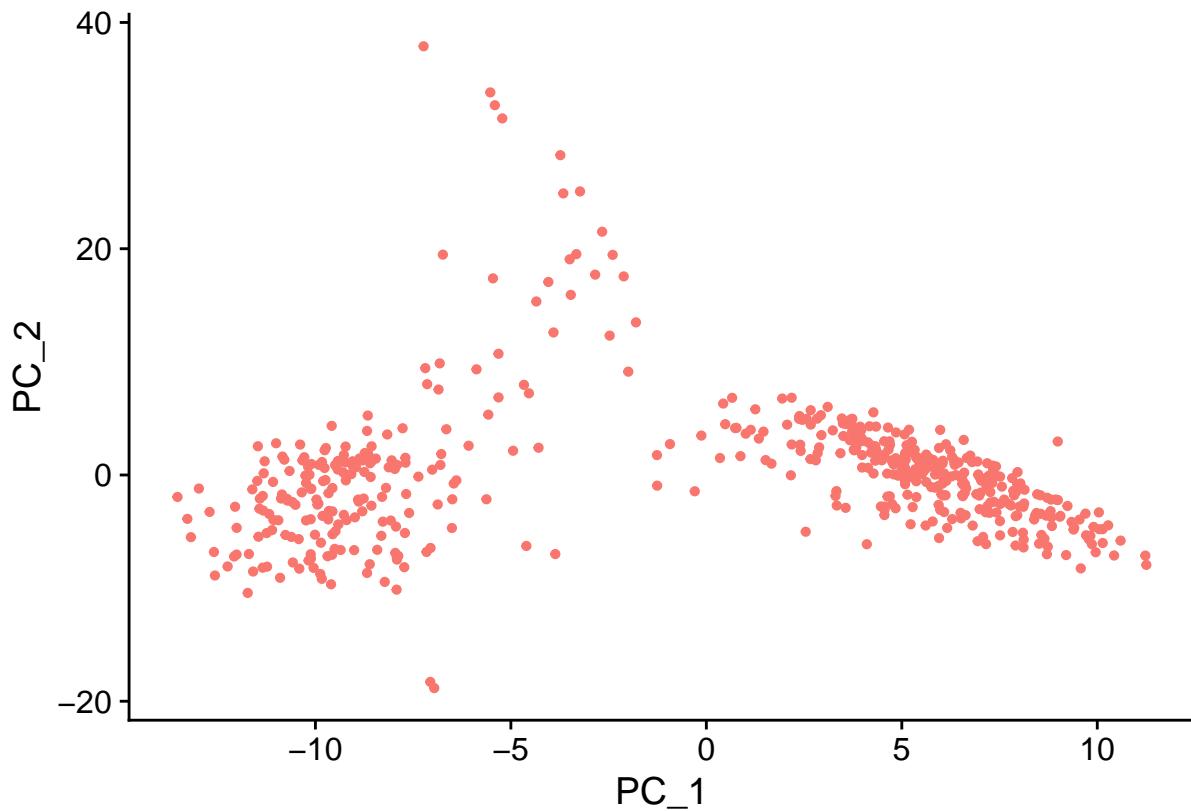
```
# Performing Principal Component Analysis on the scaled data
melanoma <- RunPCA(melanoma, features = VariableFeatures(object = melanoma))

# Examine the PCA results. Seurat outputs a list of genes with the most positive and
# negative loadings, representing modules of genes that exhibit either correlation
# (or anti-correlation) across single-cells in the dataset.
print(melanoma[["pca"]], dims = 1:5, nfeatures = 5)

# Visualize the PCA results in a few ways
VizDimLoadings(melanoma, dims = 1:2, reduction = "pca")
```

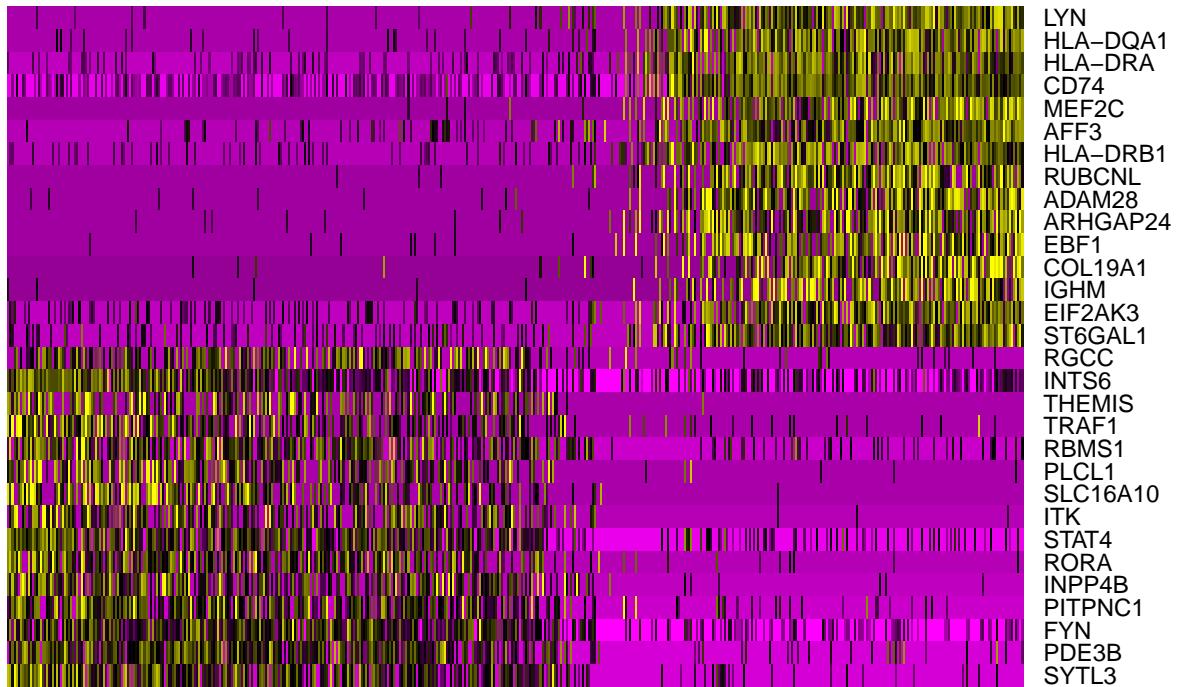


```
DimPlot(melanoma, reduction = "pca") + NoLegend()
```

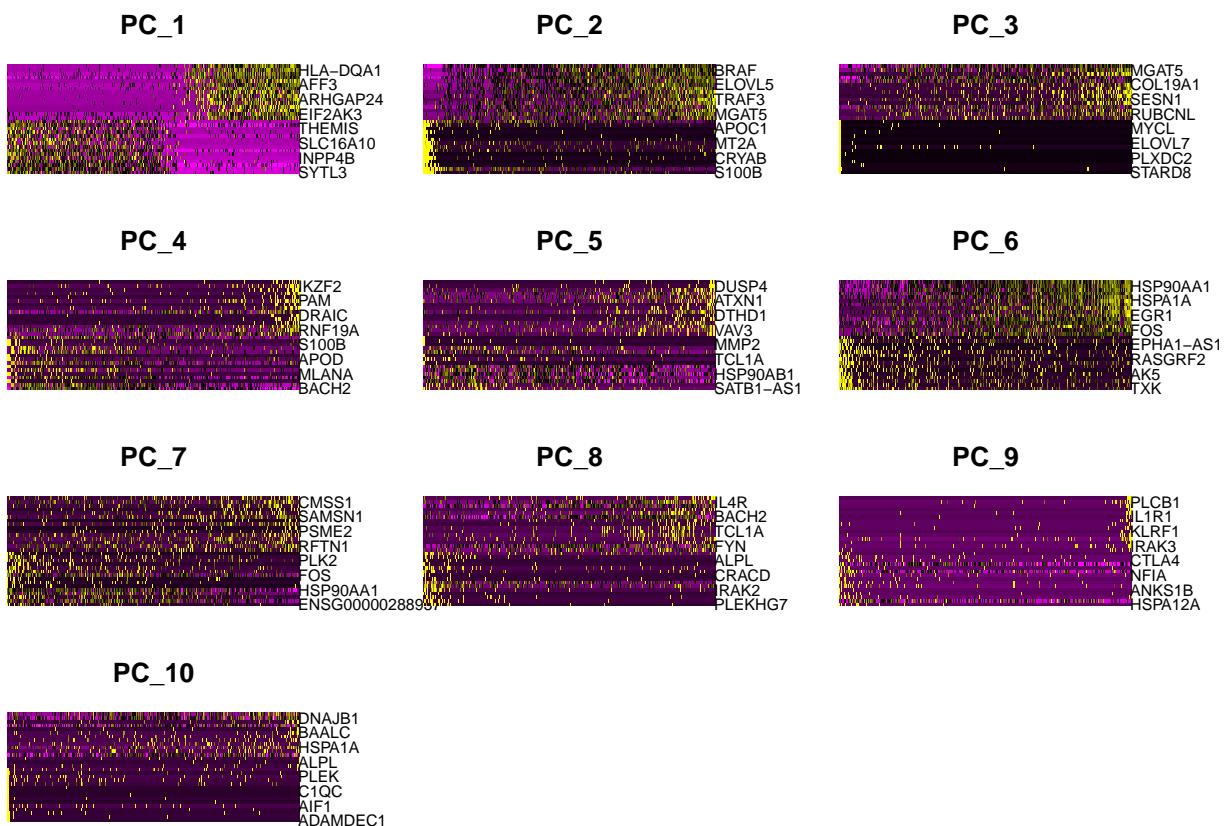


```
# Heatmap visuals are helpful in determining what Principal Components (PCs)
# to use for further downstream analysis. A strong bimodal pattern (a clear
# yellow vs purple separation) indicates that the PC is segregating cells strongly.
DimHeatmap(melanoma, dims = 1, cells = 500, balanced = TRUE)
```

PC_1

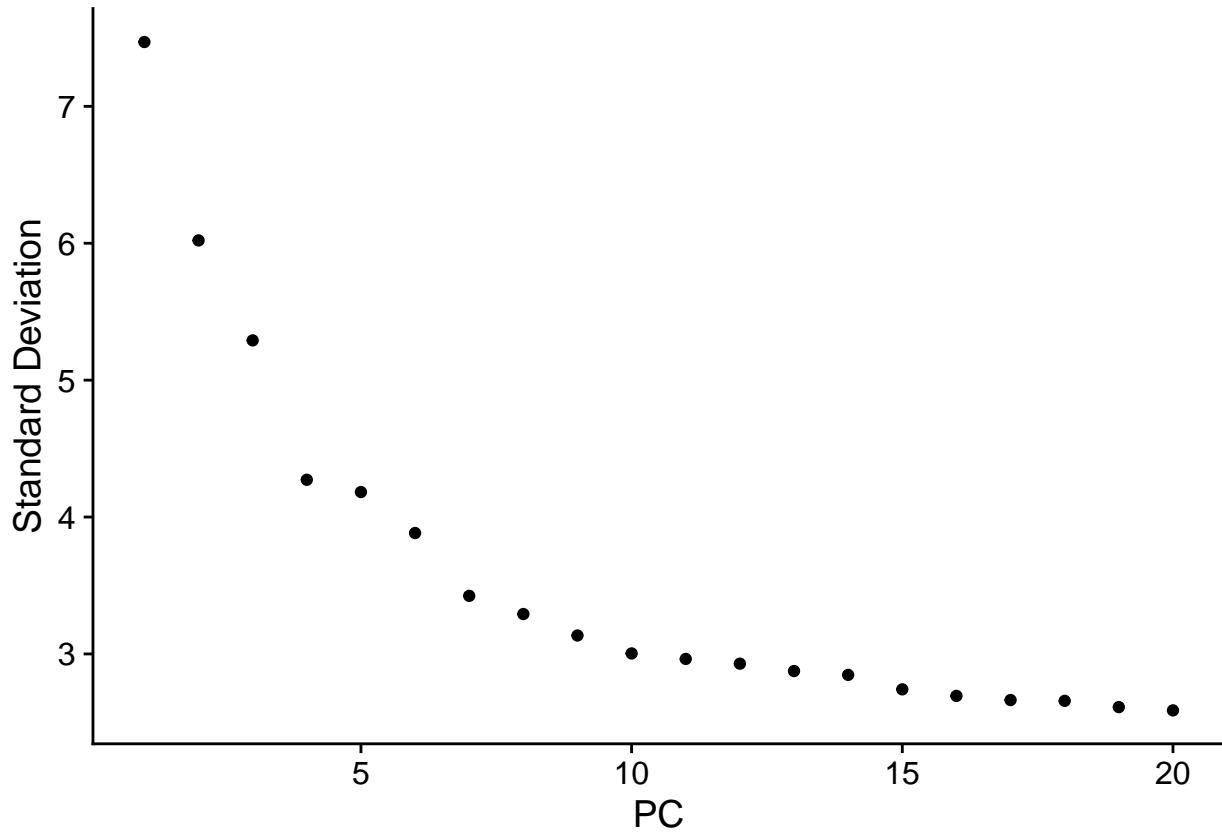


```
DimHeatmap(melanoma, dims = 1:10, cells = 500, balanced = TRUE)
```



Determine the Dimensionality of the Dataset

```
# A ranking of principle components based on the percentage of variance explained
# by each one (Elbow Plots). In this example, we see the majority of true signal is
# captured in the first 15 PCs. It is important to note that this is a qualitative
# observation that is somewhat subjective.
ElbowPlot(melanoma)
```



Cluster the Cells: Run Non-Linear Dimensional Reduction (UMAP/tSNE)

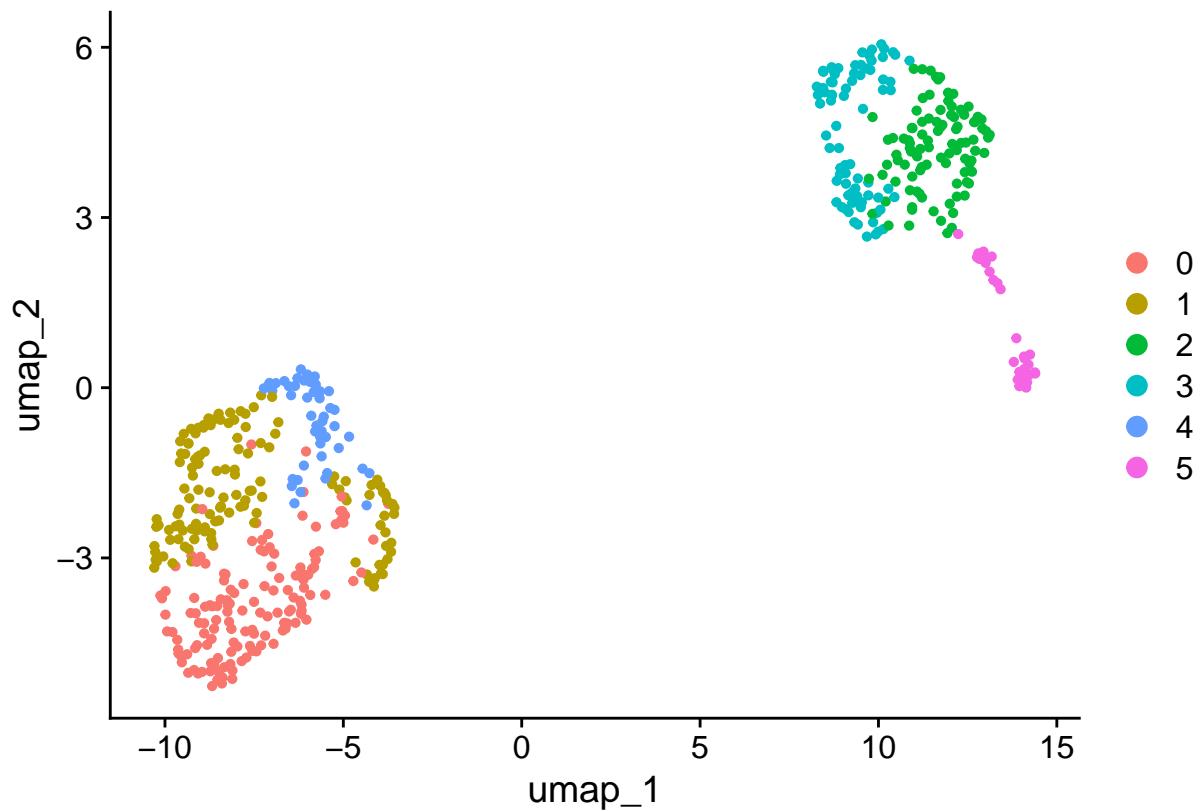
```
# Construct a KNN (K-nearest neighbor) graph based on the euclidean distance in PCA space,
# and refine the edge weights between any two cells based on the shared overlap in their
# local neighborhoods (Jaccard similarity). The first 15 PCs were selected based on the
# elbow plot observation.
melanoma <- FindNeighbors(melanoma, dims = 1:15)

# Apply modularity optimization technique (Default: Louvain algorithm)
melanoma <- FindClusters(melanoma, resolution = 0.5)

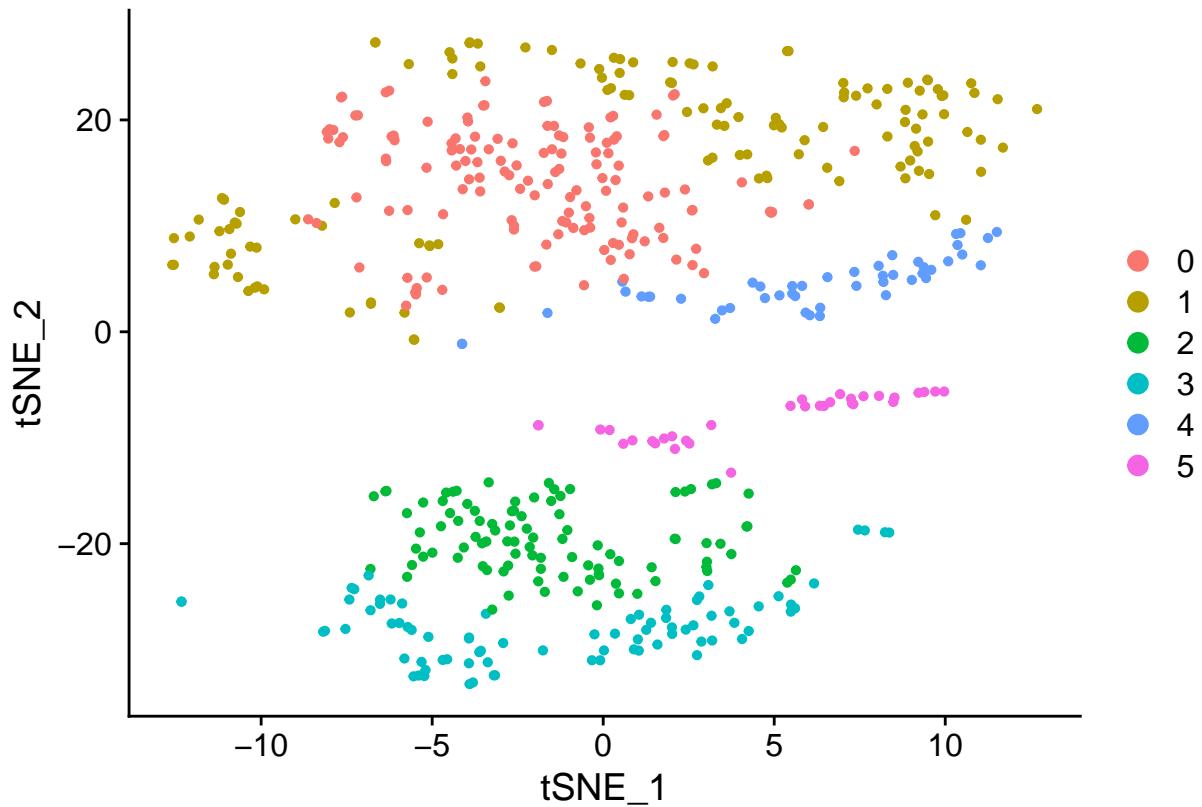
# Look at cluster IDs of the first 5 cells
head(Ids(melanoma), 5)

# Learn underlying structure in the dataset to place similar cells together in
# low-dimensional space (UMAP/tSNE). The dims= function determines which PCs to select

# UMAP
melanoma <- RunUMAP(melanoma, dims = 1:15)
DimPlot(melanoma, reduction = "umap")
```



```
# tSNE
melanoma <- RunTSNE(melanoma, dims = 1:15)
DimPlot(melanoma, reduction = "tsne")
```



Find Differentially Expressed Features (Cluster Biomarkers)

```

# Find features that define clusters via differential expression (DE). By default,
# Seurat identifies positive and negative markers of a single cluster compared to all other cells.

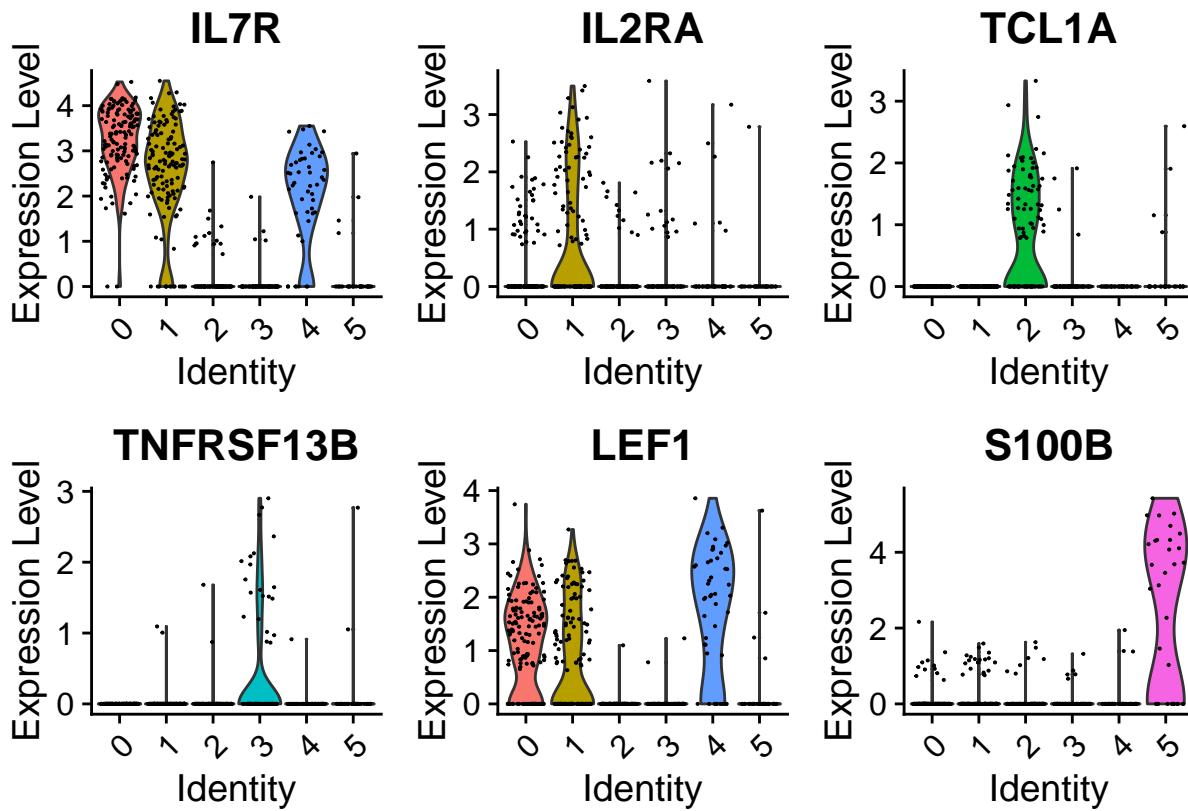
# Find top 10 markers of a specific cluster compared to the others.
# c() can contain a single cluster or list of other clusters to compare to.
# This was performed for each cluster to identify potential marker genes
cluster.markers <- FindMarkers(melanoma, ident.1 = 0, ident.2 = c(1,2,3,4,5), only.pos = TRUE)
head(cluster.markers, n = 15)
paste(head(rownames(cluster.markers), 100), collapse = ", ")

# Find markers for every cluster compared to all remaining cells, report only the positive
# ones. THIS IS THE DIFFERENTIAL GENE EXPRESSION STEP
melanoma.markers <- FindAllMarkers(melanoma, only.pos = TRUE)
melanoma.markers %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1)

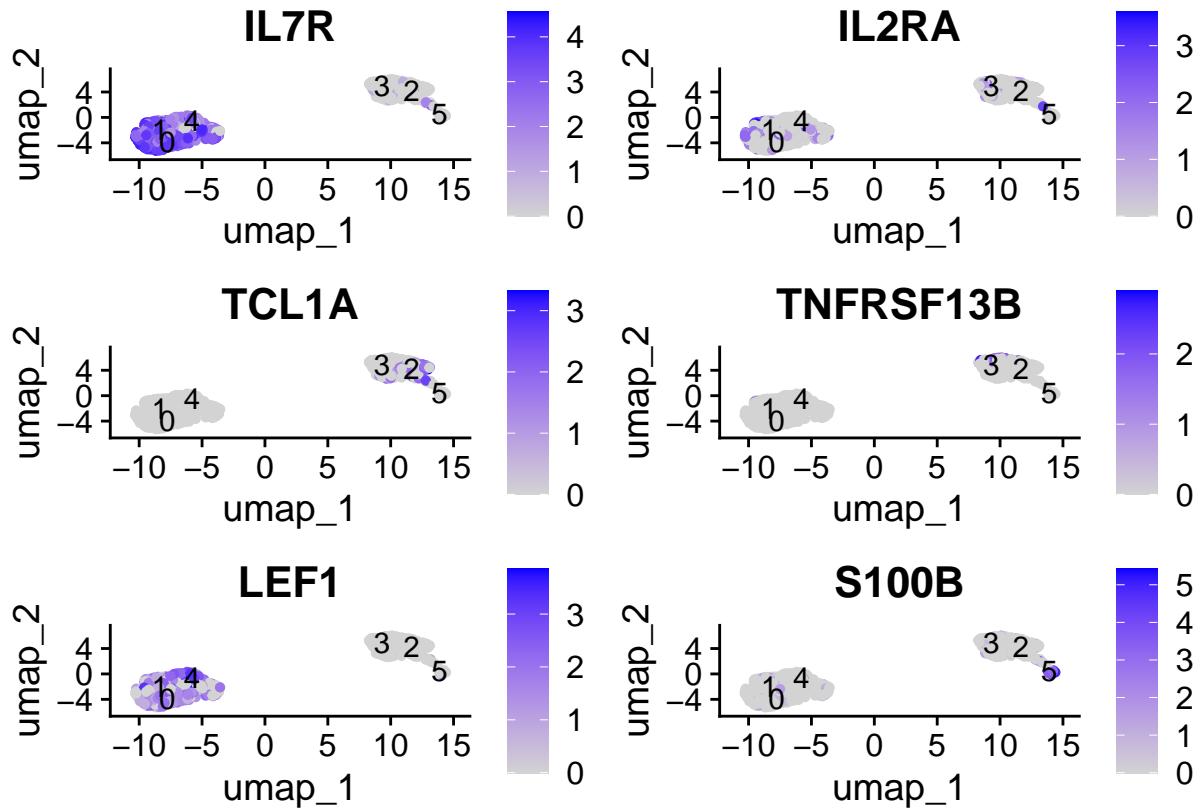
# Saving the gene list (differentially expressed features)
write.csv(melanoma.markers, "melanoma_markers.csv")

```

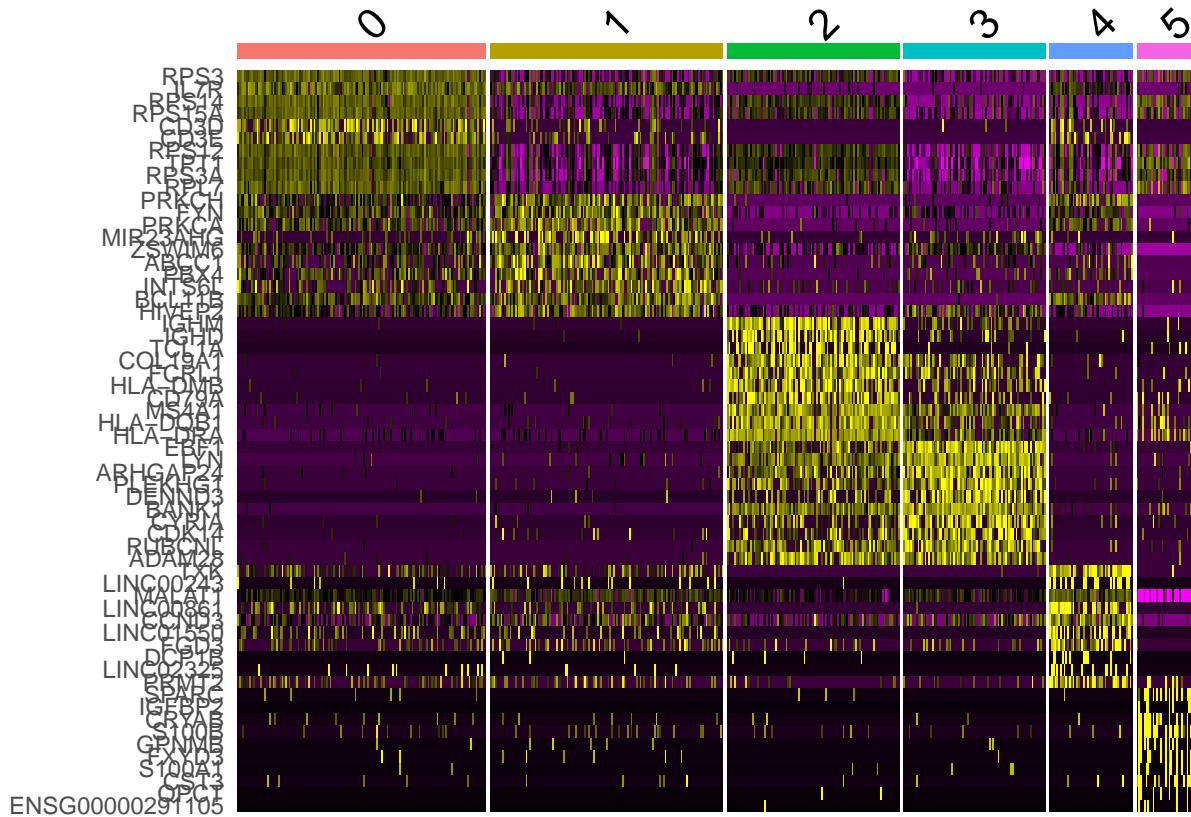
```
# Visualizing marker expression.
# Violin Plot (shows expression probability distributions across clusters)
VlnPlot(melanoma, features = c("IL7R", "IL2RA", "TCL1A", "TNFRSF13B", "LEF1", "S100B"))
```



```
# Feature Plot (visualizes feature expression on a tSNE or PCA plot)
FeaturePlot(melanoma, label = TRUE, features = c("IL7R", "IL2RA", "TCL1A", "TNFRSF13B", "LEF1", "S100B"))
```



```
# Expression Heatmap (expression heatmap for given cells and features)
melanoma.markers %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC > 1) %>%
  slice_head(n = 10) %>%
  ungroup() -> top10
DoHeatmap(melanoma, features = top10$gene) + NoLegend()
```



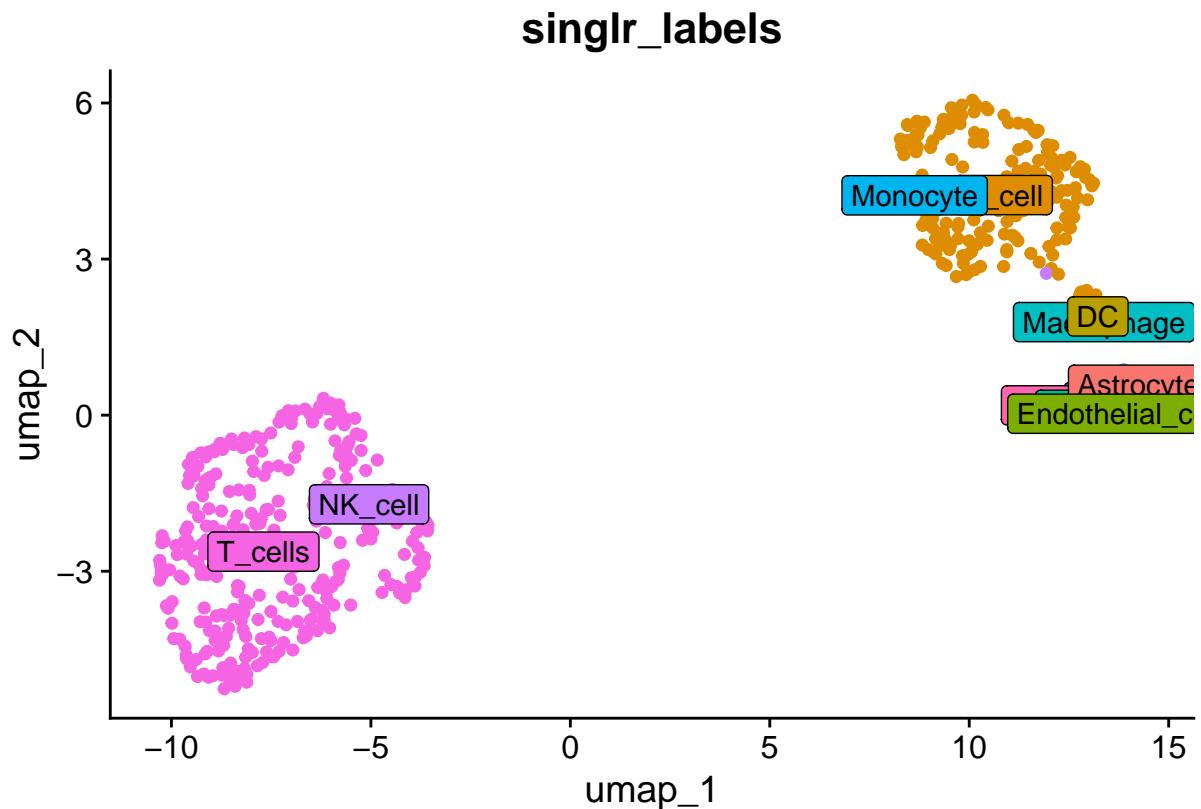
Assigning Cell Types to Identify Clusters (SingleR Method)

```
# Calling the reference data set from celldex. Here, I selected HumanPrimaryCellAtlasData()
ref <- celldex::HumanPrimaryCellAtlasData()

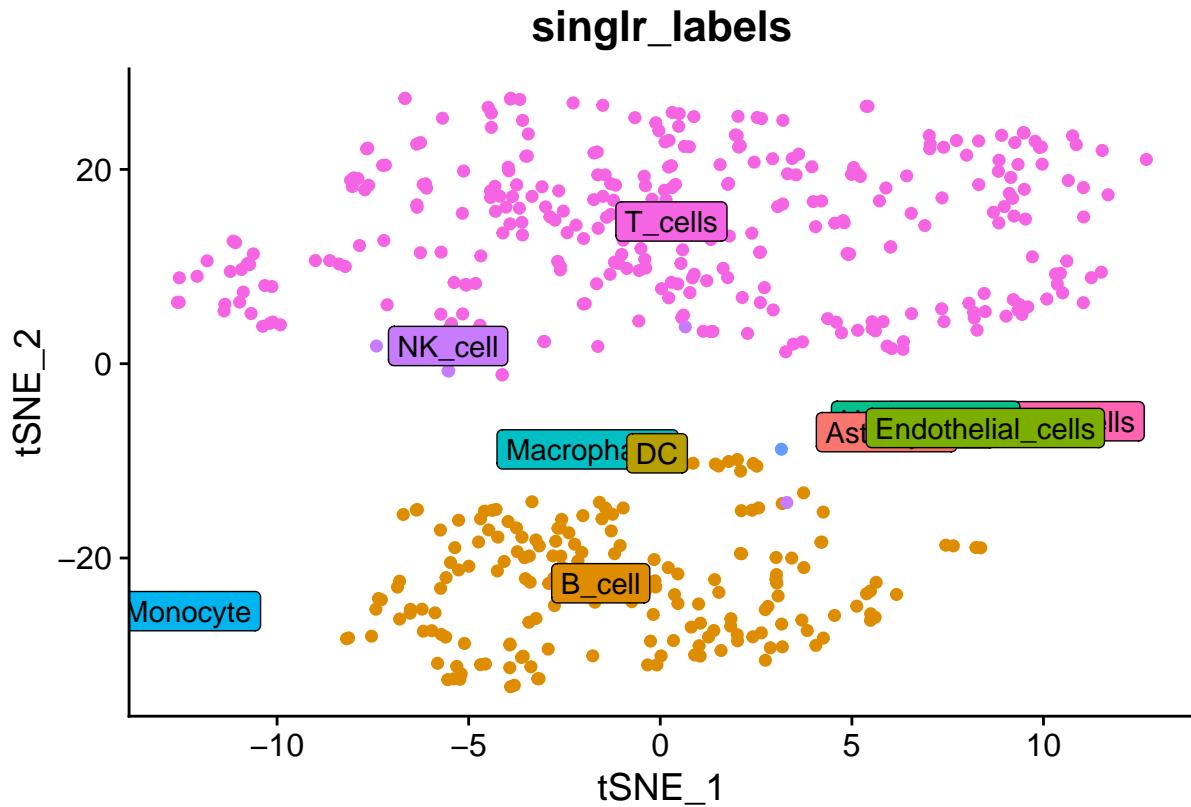
# Running SingleR to annotate clusters
results <- SingleR(test = as.SingleCellExperiment(melanoma), ref = ref, labels = ref$label.main)
results

# Creating a column of metadata to store the SingleR labels in my Seurat object
melanoma$singlr_labels <- results$labels

# Relabeled UMAP plot
DimPlot(melanoma, reduction = "umap", group.by = "singlr_labels", label = TRUE, label.box = TRUE,
        label.size = 4, pt.size = 1.5) + NoLegend()
```



```
# Relabeled tSNE plot
DimPlot(melanoma, reduction = "tsne", group.by = "singlr_labels", label = TRUE, label.box = TRUE,
        label.size = 4, pt.size = 1.5) + NoLegend()
```



Assigning Cell Types to Identify Clusters (Manual)

```
# Filtering top 10 expressed genes
top10 <- melanoma.markers %>%
  group_by(cluster) %>%
  top_n(n = 10, wt = avg_log2FC)

# Viewing genes based on cluster
top10 %>% filter(cluster == 0)
top10 %>% filter(cluster == 1)
top10 %>% filter(cluster == 2)
top10 %>% filter(cluster == 3)
top10 %>% filter(cluster == 4)
top10 %>% filter(cluster == 5)

# Forming a gene list of the top 10 genes in each cluster
gene_lists <- top10 %>%
  group_by(cluster) %>%
  summarize(genes = paste(gene, collapse = ",")) %>%
  arrange(as.numeric(as.character(cluster)))

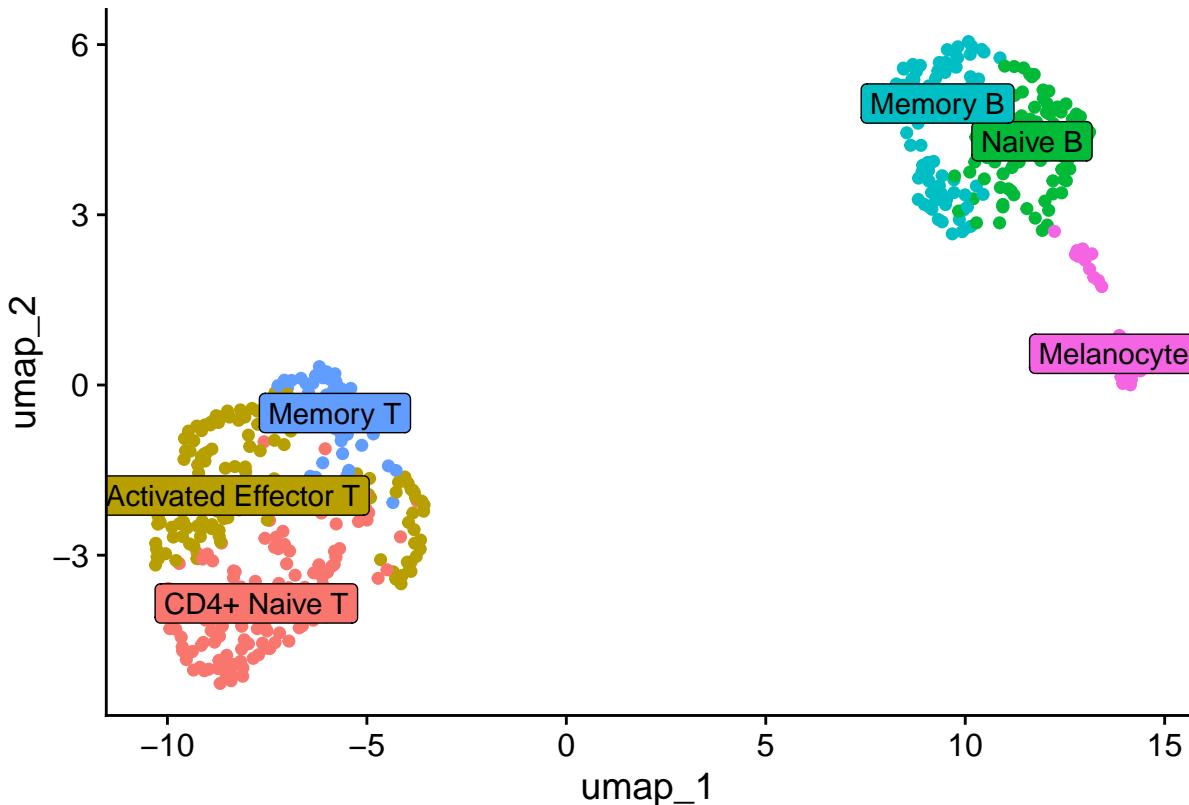
gene_lists
```

```

# Renaming the clusters based on cell type biology and associated marker genes.
new.cluster.ids <- c("CD4+ Naive T", "Activated Effector T", "Naive B",
                      "Memory B", "Memory T", "Melanocytes")
names(new.cluster.ids) <- levels(melanoma)
melanoma <- RenameIdent (melanoma, new.cluster.ids)

# Relabeled UMAP plot
DimPlot(melanoma, reduction = "umap", label = TRUE, label.size = 4, label.box = TRUE,
         pt.size = 1.5) + NoLegend()

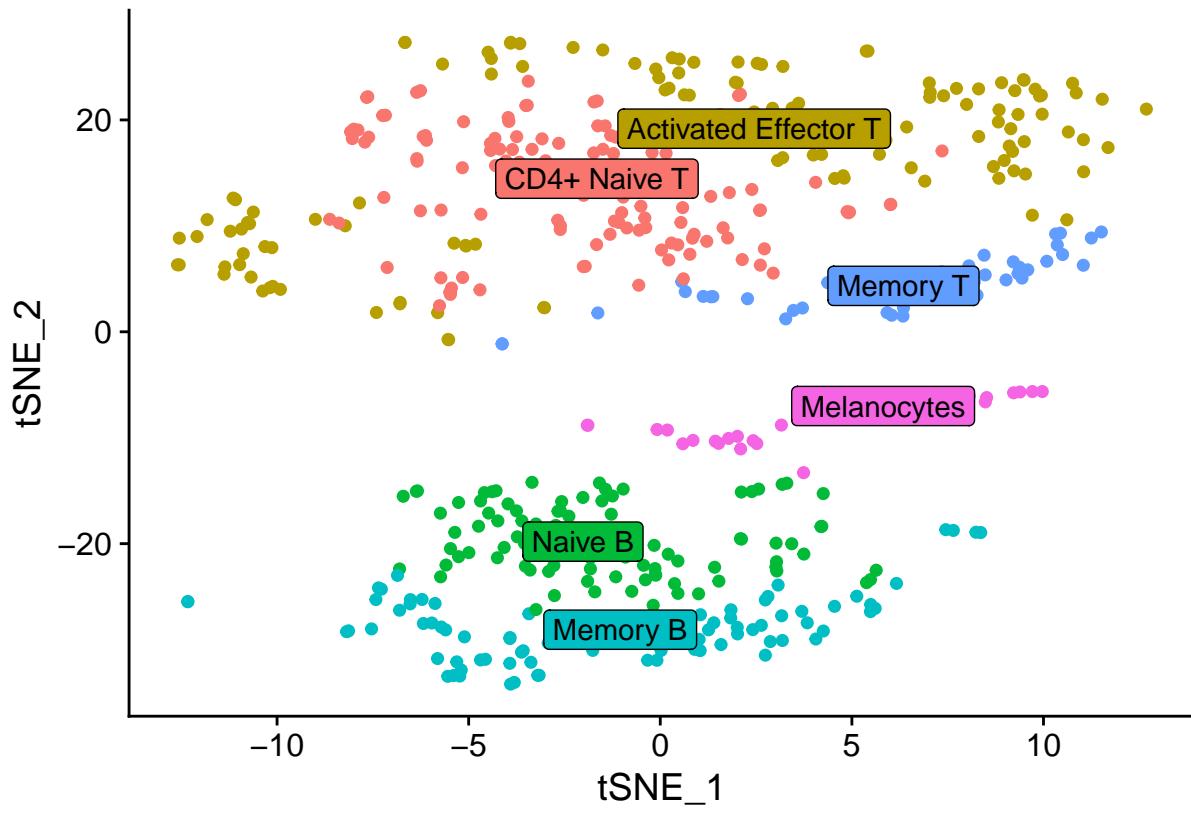
```



```

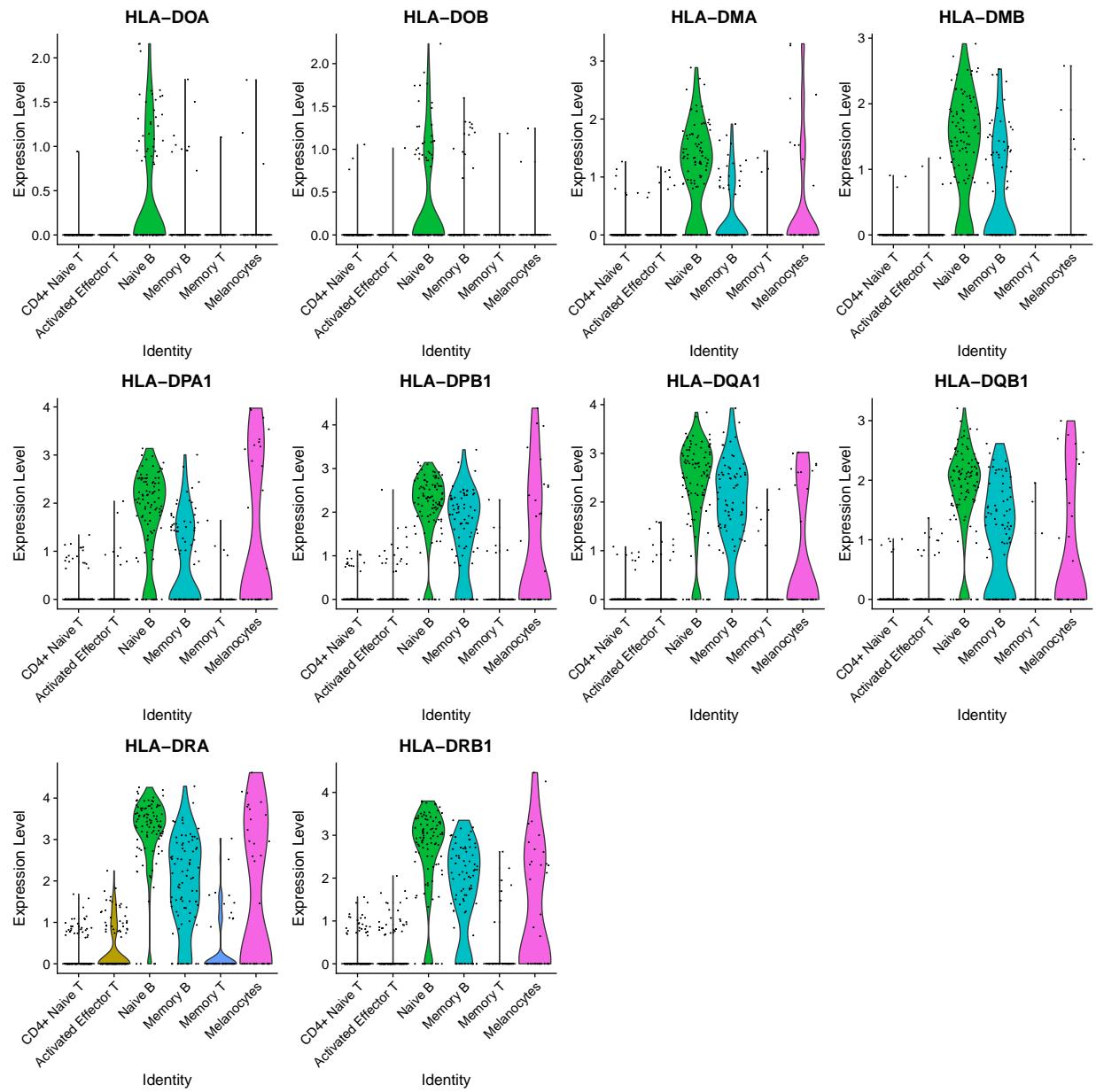
# Relabeled tSNE plot
DimPlot(melanoma, reduction = "tsne", label = TRUE, label.size = 4, label.box = TRUE,
         pt.size = 1.5) + NoLegend()

```



Visualizing HLA Class II Genes

```
# Visualizing marker expression.
# Violin Plot (shows expression probability distributions across clusters)
VlnPlot(melanoma, features = c("HLA-DOA", "HLA-DOB", "HLA-DMA", "HLA-DMB", "HLA-DPA1",
                               "HLA-DPB1", "HLA-DQA1", "HLA-DQB1", "HLA-DRA", "HLA-DRB1"))
```



```
# Feature Plot (visualizes feature expression on a tsNE or PCA plot)
```

```
FeaturePlot(melanoma,label = TRUE, features = c("HLA-DOA", "HLA-DOB", "HLA-DMA", "HLA-DMB",
"HLA-DPA1", "HLA-DPB1", "HLA-DQA1", "HLA-DQB1",
"HLA-DRA", "HLA-DRB1"))
```

