Analyse scRNA-seq Lung Cancer

Archimede PATIPE

2026-02-16

Introduction

Ce document présente une analyse scRNA-seq d'un dataset humain de cellules tumorales pulmonaires et PBMC, en utilisant Seurat et SingleR. L'objectif est de :

- Prétraiter et normaliser les données
- Identifier les clusters cellulaires
- Annoter les types cellulaires
- Identifier les gènes marqueurs et réaliser une analyse de l'expression différentielle

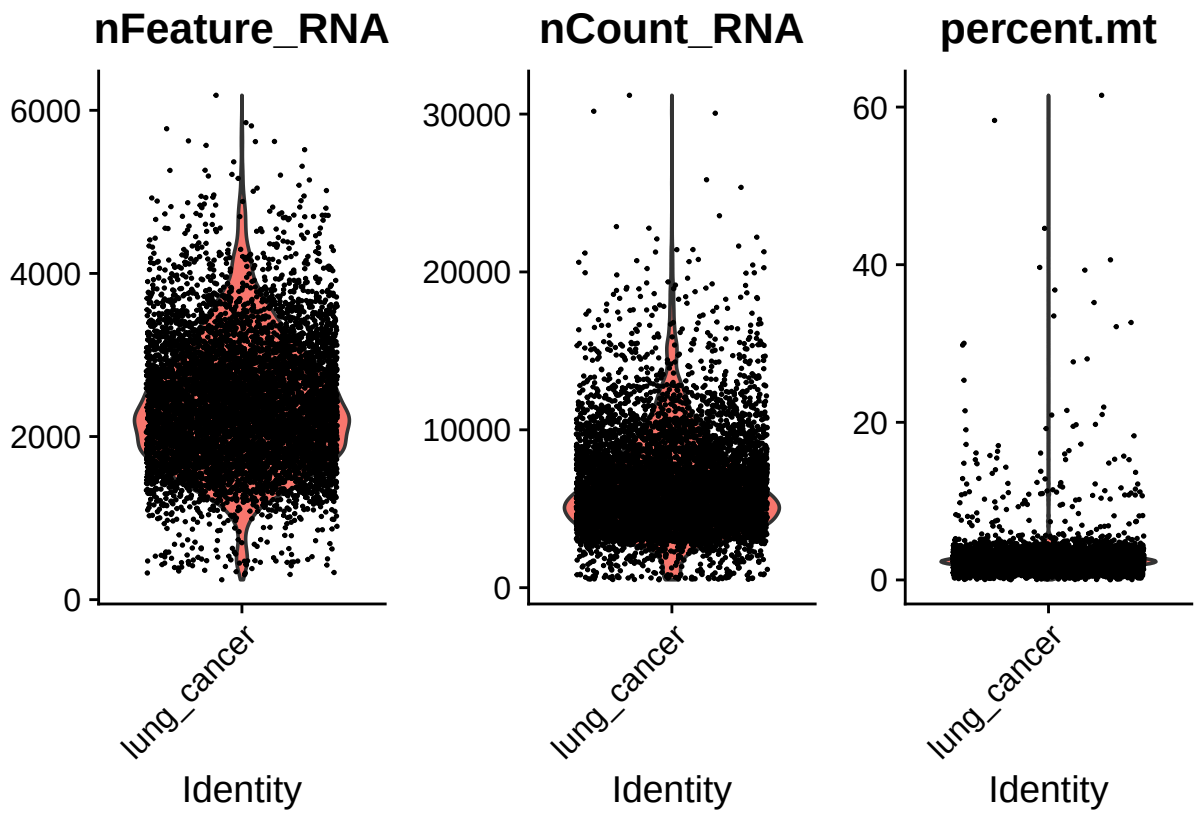
---

1. Préparation de l'environnement

```
# Packages
library(Seurat)
library(dplyr)
library(ggplot2)
library(patchwork)
library(SingleR)
library(celldex)
library(RColorBrewer)

# Set working directory
setwd("~/Single_analysis/project/data")

# Seed pour reproductibilité
set.seed(1234)

# Vérification version Matrix
packageVersion("Matrix")
```

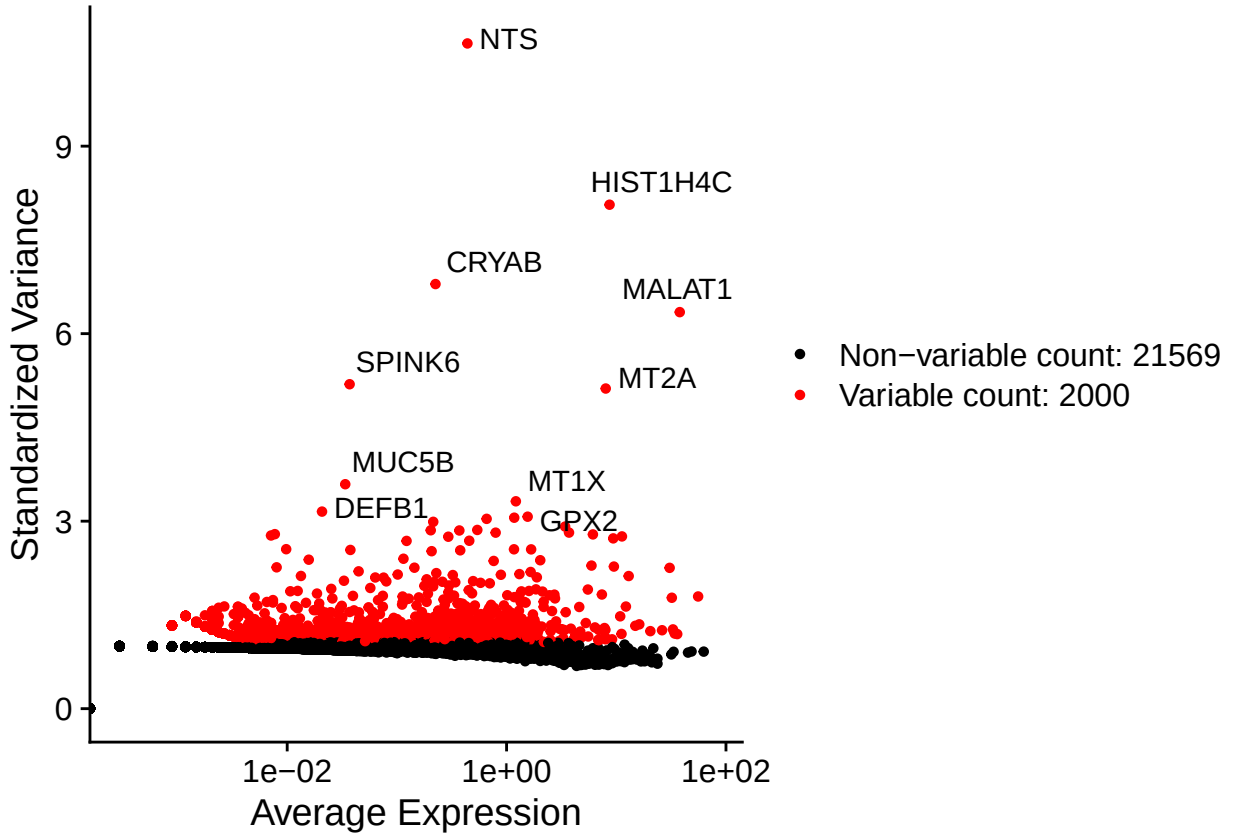
```
## [1] '1.7.4'
```

```r
# Lecture du dataset 10X
raw_data <- Read10X(data.dir = "~/Single_analysis/project/data")

# Création de l'objet Seurat
seurat_obj <- CreateSeuratObject(counts = raw_data[['Gene Expression']],
                                 project = "lung_cancer",
                                 min.cells = 3,
                                 min.features = 200)

# Calcul du pourcentage de gènes mitochondriaux
seurat_obj[["percent.mt"]] <- PercentageFeatureSet(seurat_obj, pattern = "^MT-")

# Visualisation QC metrics
VlnPlot(seurat_obj, features=c("nFeature_RNA","nCount_RNA","percent.mt"),
        pt.size = 0.1, ncol=3)
```

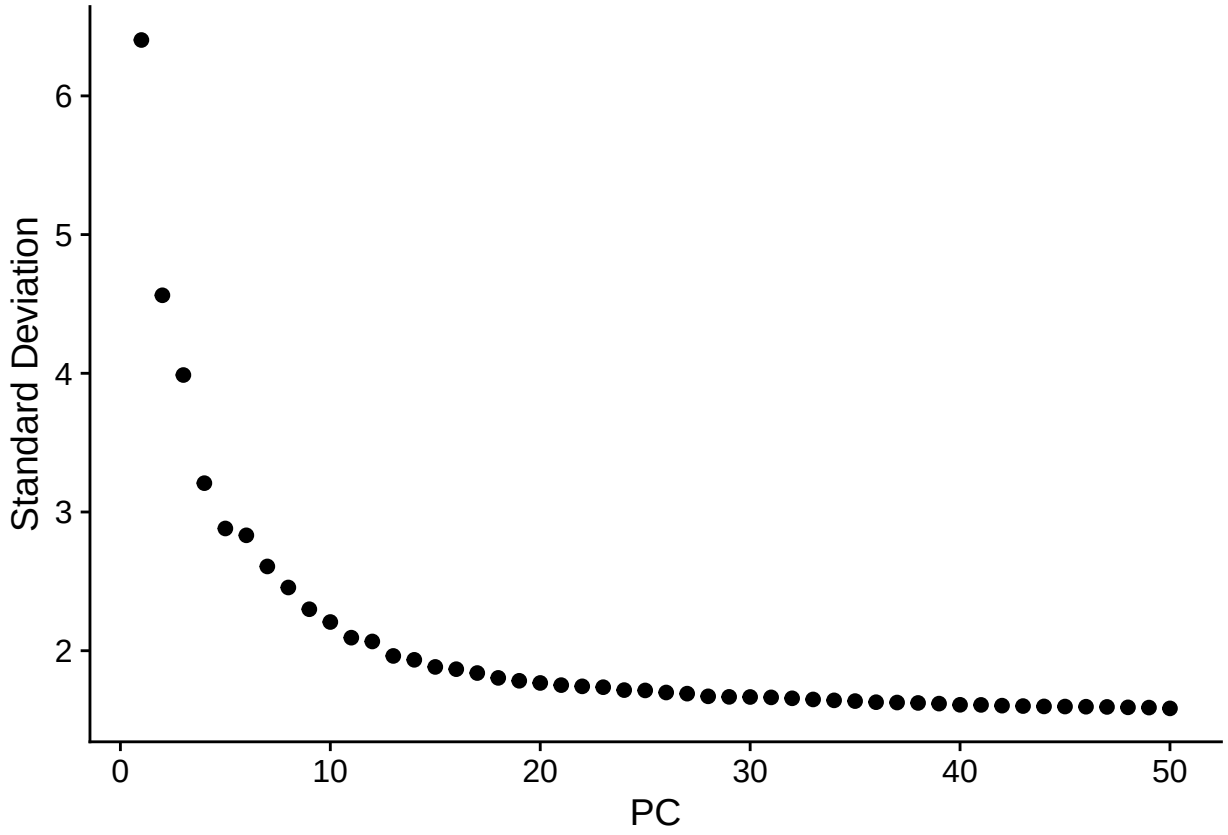



```r
# Filtrage des cellules de faible qualité
seurat_obj <- subset(seurat_obj,
                     subset = nFeature_RNA > 500 &
                              nCount_RNA < 6000 &
                              percent.mt < 8)
# Normalisation
seurat_obj <- NormalizeData(seurat_obj, normalization.method = "LogNormalize", scale.factor = 10000)

# Identification des 2000 gènes les plus variables
```

```r
seurat_obj <- FindVariableFeatures(seurat_obj, selection.method = "vst", nfeatures = 2000)

# Visualisation des gènes variables
var_plot <- VariableFeaturePlot(seurat_obj)
LabelPoints(plot = var_plot, points = head(VariableFeatures(seurat_obj),10), repel = TRUE)
```



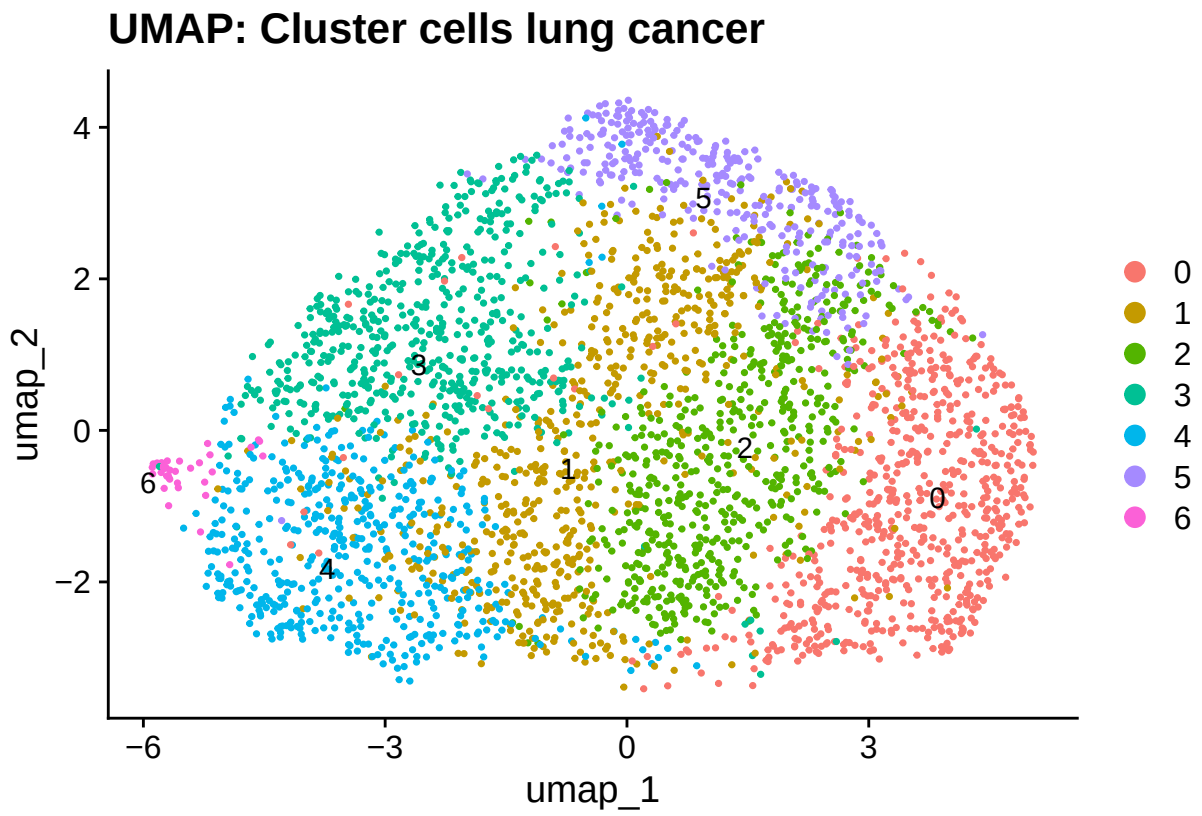

```r
# Mise à l'échelle
seurat_obj <- ScaleData(seurat_obj)

# PCA
seurat_obj <- RunPCA(seurat_obj, npcs = 50)

# Elbow Plot pour choisir le nombre de PCs
ElbowPlot(seurat_obj, ndims = 50)
```




```
# Sélection de 11 PCs pour clustering et UMAP
pcs <- 11
# KNN et clustering
seurat_obj <- FindNeighbors(seurat_obj, dims = 1:pcs)
seurat_obj <- FindClusters(seurat_obj, resolution = 0.6)
```

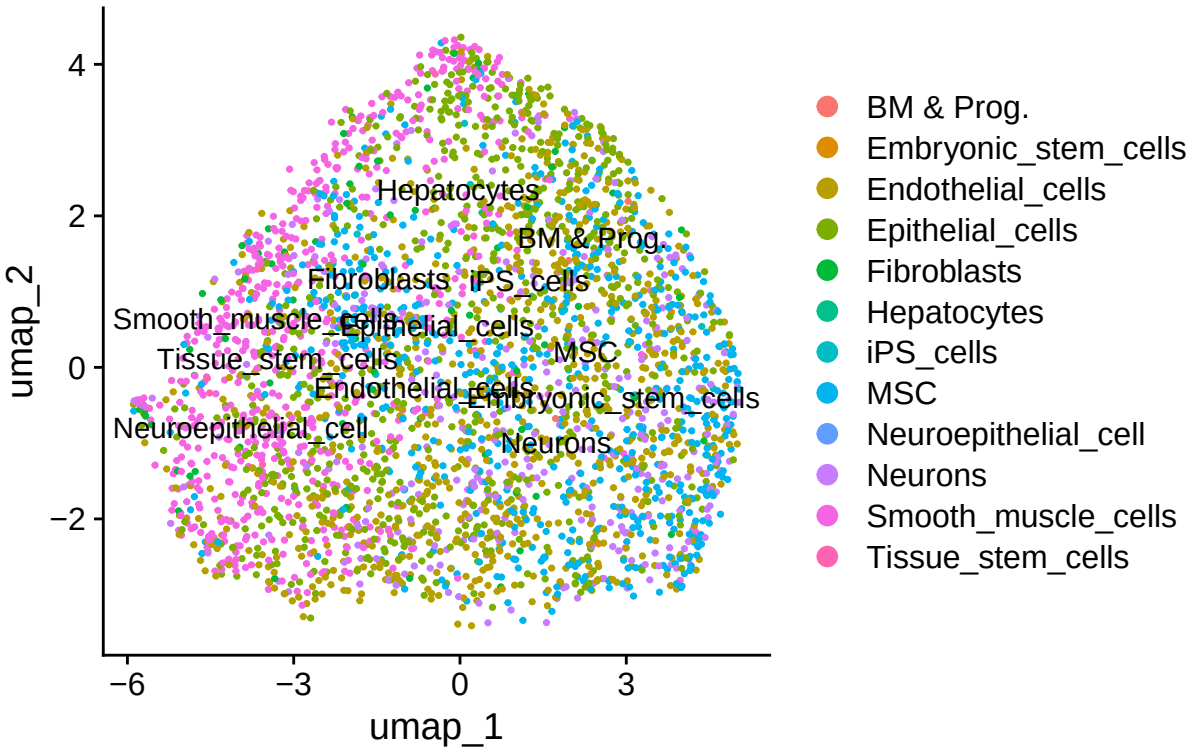
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 3371
## Number of edges: 106312
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7199
## Number of communities: 7
## Elapsed time: 0 seconds
```

```
# UMAP
seurat_obj <- RunUMAP(seurat_obj, dims = 1:pcs)

# Visualisation UMAP avec clusters
DimPlot(seurat_obj, reduction = "umap", label = TRUE, repel = TRUE) + ggtitle("UMAP: Cluster cells lung
```

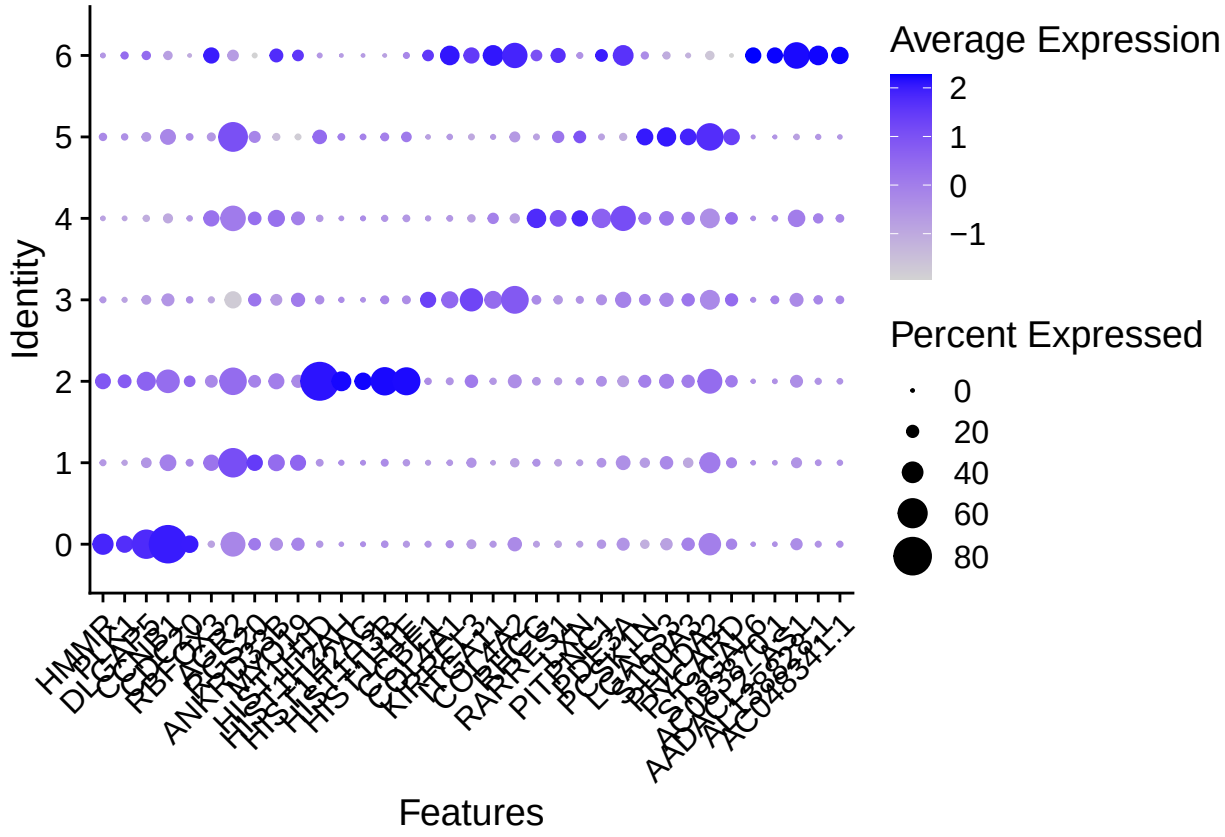



```r
# Charger le jeu de référence humain
ref <- celldex::HumanPrimaryCellAtlasData()

# Annotation SingleR
annotations <- SingleR(test = GetAssayData(seurat_obj, layer = "data"),
                       ref = ref, labels = ref$label.main)

# Ajouter les annotations à l'objet Seurat
seurat_obj$celltype <- annotations$labels

# UMAP annoté
DimPlot(seurat_obj, group.by = "celltype", label = TRUE, repel = TRUE) +
  ggtitle("UMAP: Annotated cell Types lung cancer Human")
```

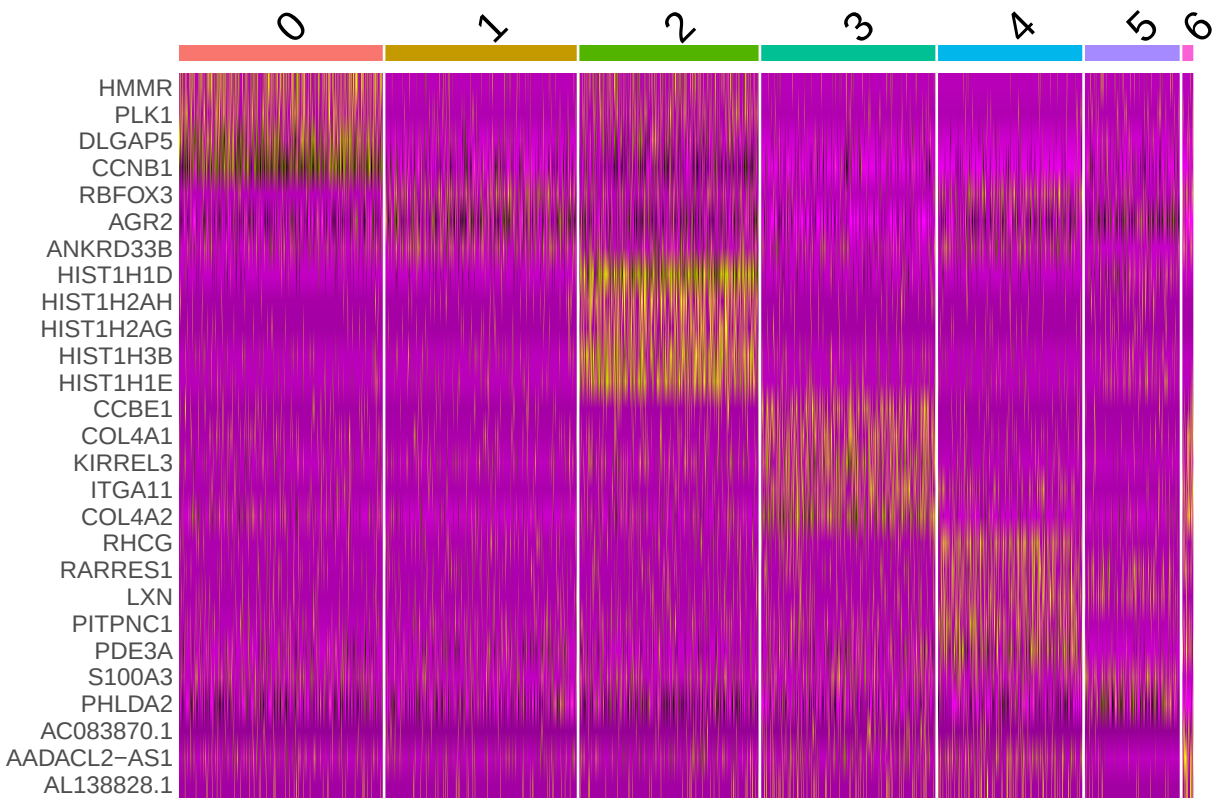
MAP: Annotated cell Types lung cancer Human




```r
# Find all positive markers
markers <- FindAllMarkers(seurat_obj,
                          only.pos = TRUE,
                          min.pct = 0.25,
                          logfc.threshold = 0.25)

# Top 5 markers par cluster
top_markers <- markers %>% group_by(cluster) %>% slice_max(n = 5 ,order_by = avg_log2FC)

# Export CSV
write.csv(top_markers,"cluster_markers.csv")
# DotPlot
DotPlot(seurat_obj, features = unique(top_markers$gene)) + RotatedAxis()
```




```
# Heatmap
DoHeatmap(seurat_obj, features = unique(top_markers$gene)) + NoLegend()
```




```r
# Comparaison entre cluster 0 (tumor) et cluster 1 (normal-like ou autre)
de_genes <- FindMarkers(seurat_obj, ident.1 = 0, ident.2 = 1,
                        min.pct = 0.25, logfc.threshold = 0.25)

# Aperçu
head(de_genes)
```

```
##                p_val avg_log2FC pct.1 pct.2     p_val_adj
## TOP2A 4.459114e-151   2.160567 0.983 0.644 1.050969e-146
## CENPF 5.470952e-126   2.236979 0.924 0.382 1.289449e-121
## CKS2  5.154900e-104   2.020118 0.884 0.474  1.214958e-99
## TPX2   1.773338e-92   2.475621 0.750 0.223  4.179579e-88
## CCNB1  4.069374e-84   1.925829 0.797 0.282  9.591107e-80
## AURKA  1.643655e-78   2.880019 0.619 0.132  3.873931e-74
```

```r
# Export CSV
write.csv(de_genes, "cluster_0_vs_cluster_1_DEGs.csv")
de_genes$gene <- rownames(de_genes)
de_genes$significant <- ifelse(de_genes$p_val_adj < 0.05 & abs(de_genes$avg_log2FC) > 0.5, "Yes", "No")

ggplot(de_genes, aes(x = avg_log2FC, y = -log10(p_val_adj), color = significant)) +
  geom_point(alpha = 0.8) +
  scale_color_manual(values = c("grey", "red")) +
  theme_minimal() +
  labs(title = "Volcano Plot: Cluster_0 vs Cluster_1",
```

```
       x = "log2 Fold Change",
       y = "-Log10 adjusted P value")
```