

ScRNA-Seq

Rishitha Pulakhandam

2025-02-16

ScRNA-Seq

In this project, I have analyzed a single-cell RNA-seq dataset using Seurat. The dataset consists of peripheral blood mononuclear cells (PBMCs), which include immune cell types such as T cells, B cells, monocytes, and natural killer (NK) cells.

```
library(dplyr)

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

library(Seurat)

## Warning: package 'Seurat' was built under R version 4.3.3
## Loading required package: SeuratObject

## Warning: package 'SeuratObject' was built under R version 4.3.3
## Loading required package: sp

## Warning: package 'sp' was built under R version 4.3.3
##
## Attaching package: 'SeuratObject'

## The following objects are masked from 'package:base':
##
##     intersect, t

library(SeuratObject)
library(patchwork)

## Warning: package 'patchwork' was built under R version 4.3.3
```

dplyr for data manipulation Seurat and SeuratObject are used to perform single cell RNA Analysis patchwork is used for combining multiple ggplots to single plot.

PBMC (Peripheral Blood Mononuclear Cells): The dataset consists of immune cells from human blood. Filtering criteria: Only genes detected in at least 3 cells are retained. Only cells with at least 200 detected genes are kept to remove low-quality cells.

```

#Load the PBMC dataset
raw_data <- Read10X(data.dir = "/Users/rishithapulakhandam/Downloads/filtered_gene_bc_matrices/hg19/")
# Initialize the Seurat Object with the raw data (not normalized).
pbmc <- CreateSeuratObject(counts = raw_data, project = "pbmc3k", min.cells = 3, min.features = 200)

## Warning: Feature names cannot have underscores ('_'), replacing with dashes
## ('-')

pbmc

## An object of class Seurat
## 13714 features across 2700 samples within 1 assay
## Active assay: RNA (13714 features, 0 variable features)
## 1 layer present: counts

#Examine the following genes present in first 30 cells
raw_data[c("CD3D", "TCL1A", "MS4A1"), 1:30]

## 3 x 30 sparse Matrix of class "dgCMatrix"
## [[ suppressing 30 column names 'AACATACAAACCAC-1', 'AACATTGAGCTAC-1', 'AACATTGATCAGC-1' ... ]]

##
## CD3D 4 . 10 . . 1 2 3 1 . . 2 7 1 . . 1 3 . 2 3 . . . . 3 4 1 5
## TCL1A . . . . . 1 . . . . . . . . . 1 . . . . .
## MS4A1 . 6 . . . . 1 1 1 . . . . . . 36 1 2 . . 2 . . .

CD3D: T-cell marker. TCL1A: B-cell marker. MS4A1: Also known as CD20, another B-cell marker. This step helps in validating whether known cell-type markers are detected in the dataset.

#Checking the size of Dense matrix
dense.size <- object.size(as.matrix(raw_data))
dense.size

## 709591472 bytes

#Checking the size of Sparse Matrix
sparse.size <- object.size(raw_data)
sparse.size

## 29905192 bytes

dense.size/sparse.size

## 23.7 bytes

```

Sparse matrices are used in scRNA-seq because most genes are not expressed in every cell, leading to a lot of zeros. Comparing dense vs. sparse matrix size shows the memory efficiency gained by using sparse representation.

QUALITY CONTROL ~Check unique genes present in each cell, as cell with less no.of genes may be low quality or empty droplets ~Cells with high gene counts could be doublets ~Total no.of molecules in each cell - correlates to no.of unique genes present in the cell, helps identify cells with abnormal counts. ~Calculate the percentage of reads mapping to mitochondrial genes, a high percentage od mitochondrial genes indicates dying or low quality cells

```

# Add a column to store the QC Stats
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "MT-")

```

Helps to identify cells which may have mitochondrial contamination

```
#QC metrics for the first 10 cells
head(pbmc@meta.data, 10)
```

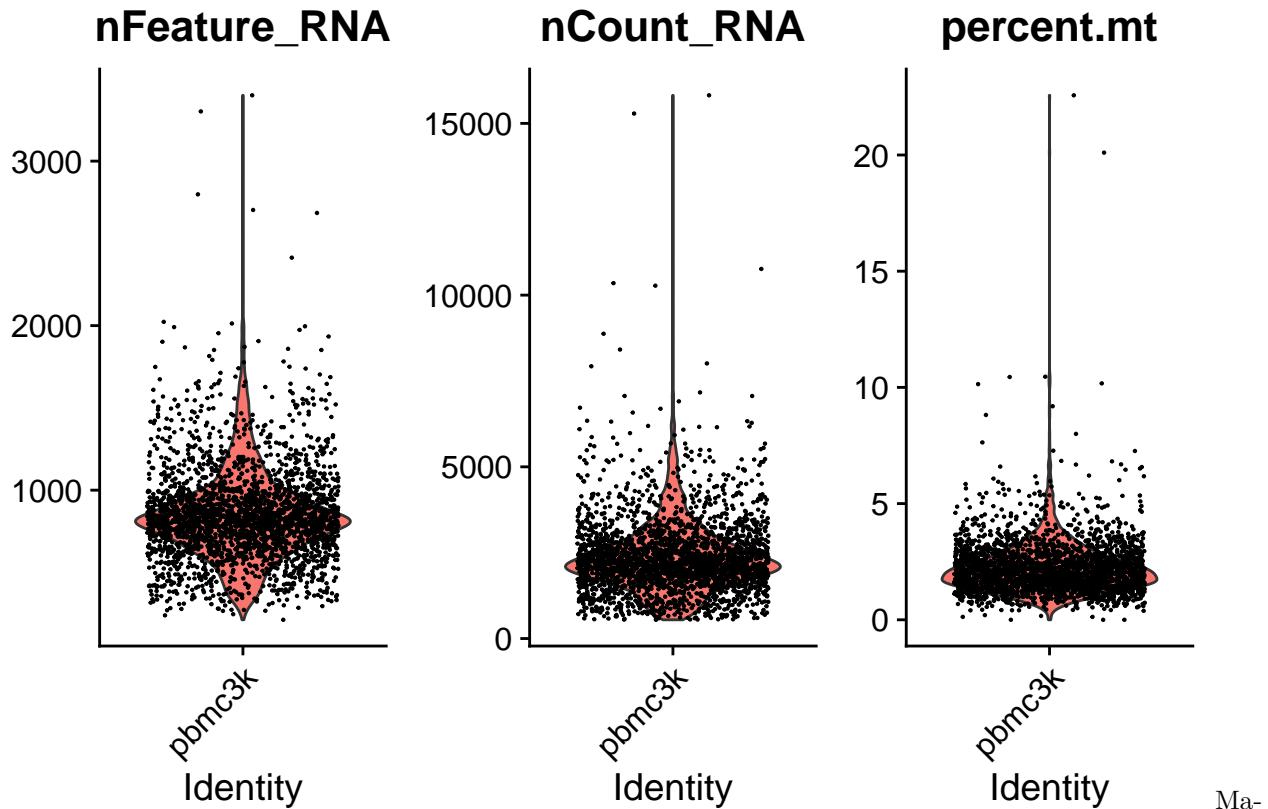
```
##          orig.ident nCount_RNA nFeature_RNA percent.mt
## AAACATACAACCAC-1    pbmc3k      2419        779  3.0177759
## AAACATTGAGCTAC-1    pbmc3k      4903       1352  3.7935958
## AAACATTGATCAGC-1    pbmc3k      3147       1129  0.8897363
## AAACCGTGTTCGG-1    pbmc3k      2639        960  1.7430845
## AAACCGTGTATGCG-1    pbmc3k      980        521  1.2244898
## AAACGCACTGGTAC-1    pbmc3k      2163        781  1.6643551
## AAACGCTGACCACT-1    pbmc3k      2175        782  3.8160920
## AAACGCTGGTTCTT-1    pbmc3k      2260        790  3.0973451
## AAACGCTGTAGCCA-1    pbmc3k      1275        532  1.1764706
## AAACGCTGTTCTG-1    pbmc3k      1103        550  2.9011786
```

percent.mt stores the percentage of mitochondrial genes in the cell. nFeature_RNA: Number of unique genes detected per cell. nCount_RNA: Total RNA molecules detected per cell. Cells with too few genes may be empty droplets. Cells with too many genes may be doublets (two cells merged into one).

```
#Vizualize the QC metrics as a Violin Plot
```

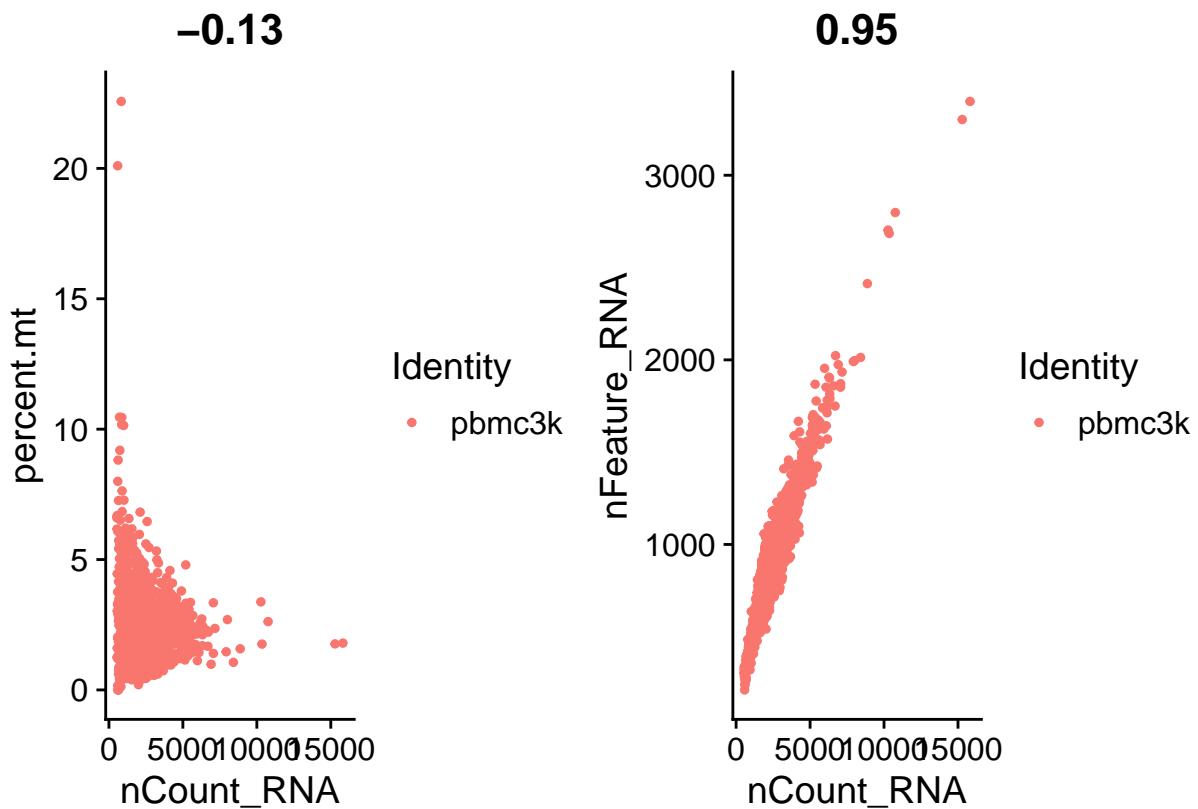
```
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```

```
## Warning: Default search for "data" layer in "RNA" assay yielded no results;
## utilizing "counts" layer instead.
```



Majority of the cells have 1000 unique genes. Majority of cells have RNA count at around 2000 -3000 range. Majority of the cells can be seen to contain 2-3 % mitochondrial gene contamination

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



there is a weak negative correlation between the number of RNA and percentage of mitochondrial gene. As seen in the second plot the higher the total number of RNA the higher the no.of genes present. There are presence of outliers in the second plot , the cell contains very high number of unique genes. It can be seen that the features total number of genes to the total number RNA transcripts have strong positive correlation.

Removing low-quality cells that have fewer than 200 genes. Removing potential doublets with more than 2500 genes. Filtering based on mitochondrial RNA percentage (usually <5%) ensures removal of dying cells.

```
#filter the data to remove high. values
pbmc <- subset(pbm, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)
pbmc
```

```
## An object of class Seurat
## 13714 features across 2638 samples within 1 assay
## Active assay: RNA (13714 features, 0 variable features)
## 1 layer present: counts
```

Features- no.of unique Genes Samples - no.of cells

Normalization corrects for differences in sequencing depth between cells. Log normalization is commonly used in scRNA-seq.

Normalize each cells gene expression by dividing by the total expression of that cell, multiplying by the scale factor (default 10000), then applying a log transformation *do not enter 10,000 enter 10000.

```
pbmc <- NormalizeData(pbm, normalization.method = "LogNormalize" , scale.factor = 10000)
```

```
## Normalizing layer: counts
```

or we can normalize the data by simply using the below command: pbmc <- NormalizeData(pbm)

Identifying highly variable features, the important biological signals in single cell data sets could be understood by focusing on the highly variable genes in the downstream analysis.

Not all genes are equally informative. The top 2,000 most variable genes are selected for downstream clustering.

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

```
## Finding variable features for layer counts
```

```
#Top 10 highly variable genes
```

```
top10 <- head(VariableFeatures(pbmc), 10)
```

```
# Plot showing the variable features with and without labels
```

```
plot1 <- VariableFeaturePlot(pbmc)
```

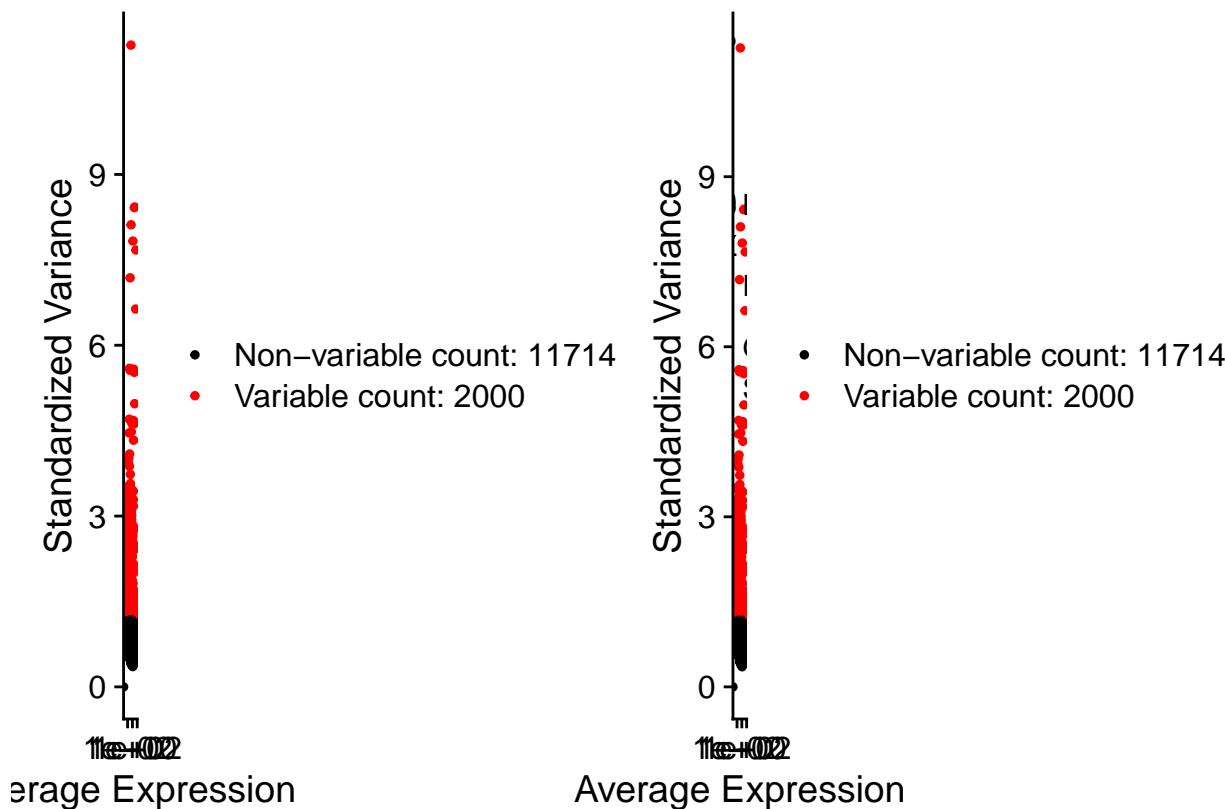
```
plot2 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)
```

```
## When using repel, set xnudge and ynudge to 0 for optimal results
```

```
plot1+plot2
```

```
## Warning: Transformation introduced infinite values in continuous x-axis
```

```
## Transformation introduced infinite values in continuous x-axis
```



vst stands for variance stabilizing transformation method.

PPBP is the highly variable gene.

shifting the expression of each gene so that the mean expression across the cell is 0. Scales the expression across the cell so that the variance is 1. The results are stored in pbmc[["RNA"]]\$scale.data By default only the variable features are scaled The scaling is applied to all the genes in the dataset

Scaling removes technical noise and ensures genes are on a comparable scale.

```

all.genes <- rownames(pbmc)
pbmc <- ScaleData(pbmc, features = all.genes)

```

```

## Centering and scaling data matrix

```

We could remove unwanted sources of variation from the single cell dataset, we could regress out heterogeneity associated with cell cycle stage or mitochondrial contamination by: pbmc <- ScaleData(pbmc, vars.to.regress = "percent.mt")

we are using the previously determined variable features as input (it is by default), but can change choosing a different subset in the features argument (must be passed through ScaleData() first)

PCA reduces dimensionality, making it easier to cluster cells. This helps identify major sources of variation in the dataset.

```

# Run PCA
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))

```

```

## PC_ 1
## Positive: CST3, TYROBP, LST1, AIF1, FTL, FTH1, LYZ, FCN1, S100A9, TYMP
##          FCER1G, CFD, LGALS1, S100A8, CTSS, LGALS2, SERPINA1, IFITM3, SPI1, CFP
##          PSAP, IFI30, SAT1, COTL1, S100A11, NPC2, GRN, LGALS3, GSTP1, PYCARD
## Negative: MALAT1, LTB, IL32, IL7R, CD2, B2M, ACAP1, CD27, STK17A, CTSW
##          CD247, GIMAP5, AQP3, CCL5, SELL, TRAF3IP3, GZMA, MAL, CST7, ITM2A
##          MYC, GIMAP7, HOPX, BEX2, LDLRAP1, GZMK, ETS1, ZAP70, TNFAIP8, RIC3
## PC_ 2
## Positive: CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1, HLA-DRA, LINC00926, CD79B, HLA-DRB1, CD74
##          HLA-DMA, HLA-DPB1, HLA-DQA2, CD37, HLA-DRB5, HLA-DMB, HLA-DPA1, FCRLA, HVCN1, LTB
##          BLNK, P2RX5, IGLL5, IRF8, SWAP70, ARHGAP24, FCGR2B, SMIM14, PPP1R14A, C16orf74
## Negative: NKG7, PRF1, CST7, GZMB, GZMA, FGFBP2, CTSW, GNLY, B2M, SPON2
##          CCL4, GZMH, FCGR3A, CCL5, CD247, XCL2, CLIC3, AKR1C3, SRGN, HOPX
##          TTC38, APMAP, CTSC, S100A4, IGFBP7, ANXA1, ID2, IL32, XCL1, RHOC
## PC_ 3
## Positive: HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPB1, HLA-DPA1, CD74, MS4A1, HLA-DRB1, HLA-DRA
##          HLA-DRB5, HLA-DQA2, TCL1A, LINC00926, HLA-DMB, HLA-DMA, CD37, HVCN1, FCRLA, IRF8
##          PLAC8, BLNK, MALAT1, SMIM14, PLD4, P2RX5, IGLL5, LAT2, SWAP70, FCGR2B
## Negative: PPBP, PF4, SDPR, SPARC, GNG11, NRGN, GP9, RGS18, TUBB1, CLU
##          HIST1H2AC, AP001189.4, ITGA2B, CD9, TMEM40, PTCRA, CA2, ACRBP, MMD, TREML1
##          NGFRAP1, F13A1, SEPT5, RUFY1, TSC22D1, MPP1, CMTM5, RP11-367G6.3, MYL9, GP1BA
## PC_ 4
## Positive: HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1, CD74, HIST1H2AC, HLA-DPB1, PF4, SDPR
##          TCL1A, HLA-DRB1, HLA-DPA1, HLA-DQA2, PPBP, HLA-DRA, LINC00926, GNG11, SPARC, HLA-DRB5
##          GP9, AP001189.4, CA2, PTCRA, CD9, NRGN, RGS18, CLU, TUBB1, GZMB
## Negative: VIM, IL7R, S100A6, IL32, S100A8, S100A4, GIMAP7, S100A10, S100A9, MAL
##          AQP3, CD2, CD14, FYB, LGALS2, GIMAP4, ANXA1, CD27, FCN1, RBP7
##          LYZ, S100A11, GIMAP5, MS4A6A, S100A12, FOLR3, TRABD2A, AIF1, IL8, IFI6
## PC_ 5
## Positive: GZMB, NKG7, S100A8, FGFBP2, GNLY, CCL4, CST7, PRF1, GZMA, SPON2
##          GZMH, S100A9, LGALS2, CCL3, CTSW, XCL2, CD14, CLIC3, S100A12, RBP7
##          CCL5, MS4A6A, GSTP1, FOLR3, IGFBP7, TYROBP, TTC38, AKR1C3, XCL1, HOPX
## Negative: LTB, IL7R, CKB, VIM, MS4A7, AQP3, CYTIP, RP11-290F20.3, SIGLEC10, HMOX1
##          LILRB2, PTGES3, MAL, CD27, HN1, CD2, GDI2, CORO1B, ANXA5, TUBA1B
##          FAM110A, ATP1A1, TRADD, PPA1, CCDC109B, ABRACL, CTD-2006K23.1, WARS, VM01, FYB

```

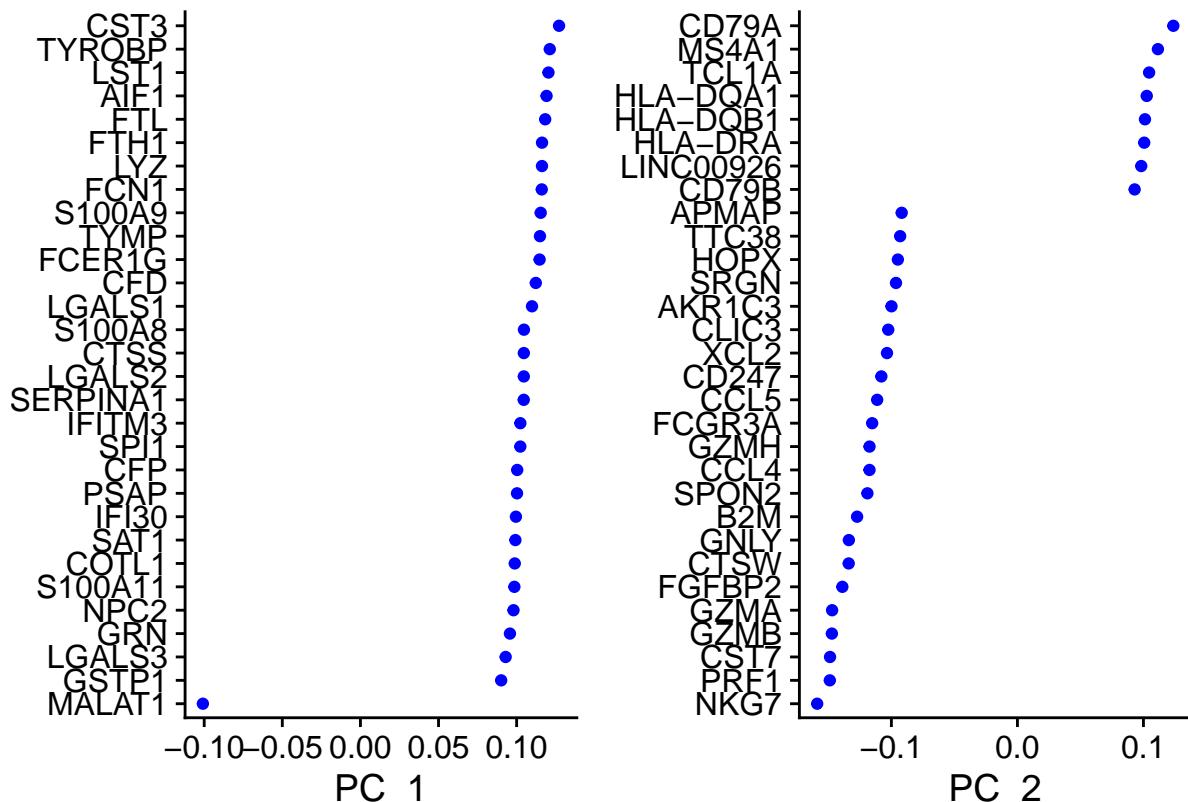
positive - high gene expression negative - the gene show higher levels of expression in the cells that score low on PC_1.

Genes in the positive loading are positive related to the cells , genes that are together can incide they are regulated of the the same pathway or are involved in the similar biological processes. Genes in the negative loading are inversely related to each other.

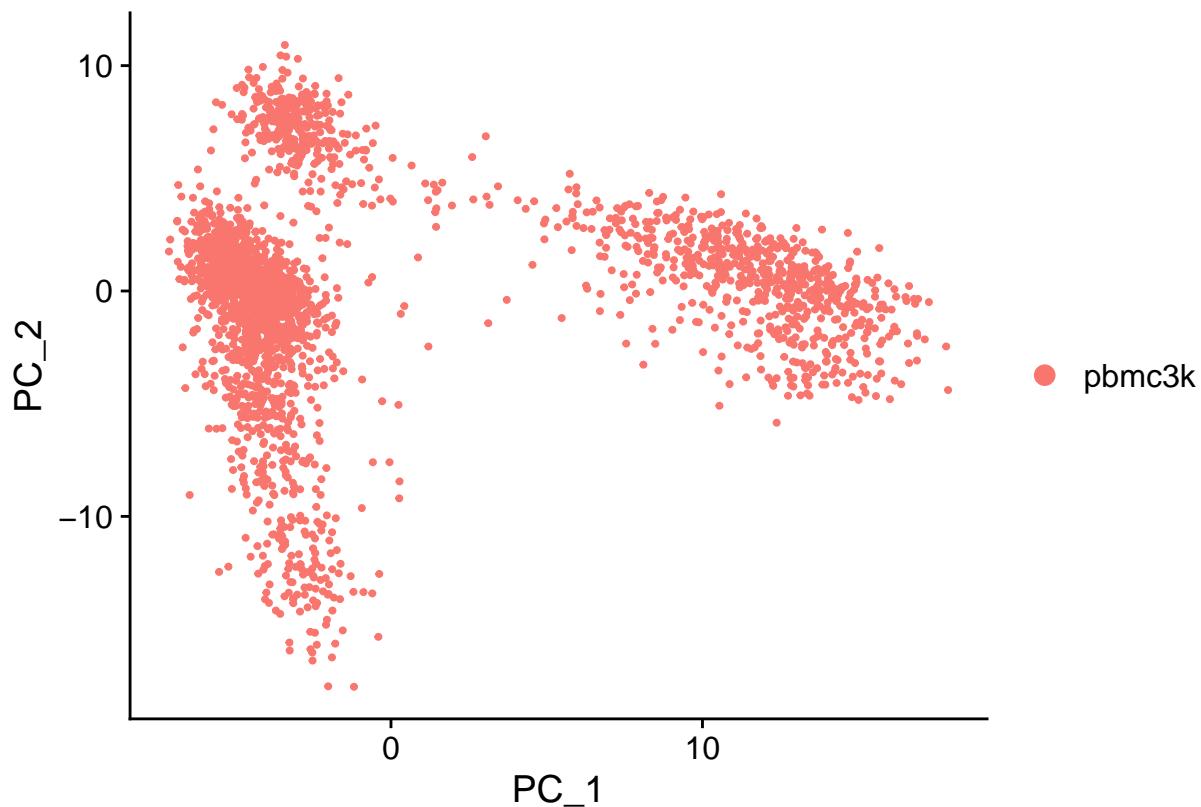
```
print(pbmc[["pca"]], dims = 1:5, nfeatures = 5)

## PC_ 1
## Positive: CST3, TYROBP, LST1, AIF1, FTL
## Negative: MALAT1, LTB, IL32, IL7R, CD2
## PC_ 2
## Positive: CD79A, MS4A1, TCL1A, HLA-DQA1, HLA-DQB1
## Negative: NKG7, PRF1, CST7, GZMB, GZMA
## PC_ 3
## Positive: HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPB1
## Negative: PPBP, PF4, SDPR, SPARC, GNG11
## PC_ 4
## Positive: HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1
## Negative: VIM, IL7R, S100A6, IL32, S100A8
## PC_ 5
## Positive: GZMB, NKG7, S100A8, FGFBP2, GNLY
## Negative: LTB, IL7R, CKB, VIM, MS4A7
```

```
# or
VizDimLoadings(pbmc, dims = 1:2, reduction = "pca")
```



```
# or
DimPlot(pbmc, reduction = "pca")
```

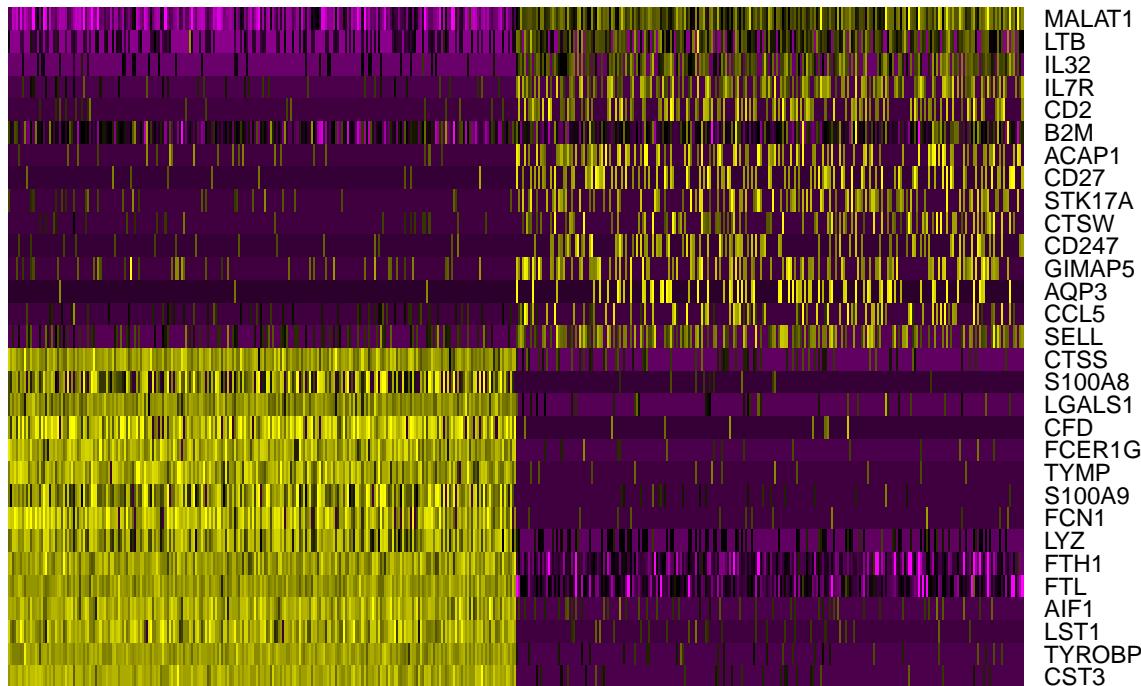


the first code results shows the top genes associated with each principal component, the second code shows a graphical representation of each principal component (we have mentioned 2). The third code shows how the cells are distributed based on the principal component.

The cells and features are ordered according to their PCA scores

```
DimHeatmap(pbmc, dims= 1, cells= 500, balanced = TRUE)
```

PC_1



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

PC_1

PC_2

PC_3

PC_4

PC_5

PC_6

PC_7

PC_8

PC_9

PC_10

PC_11

PC_12

PC_13

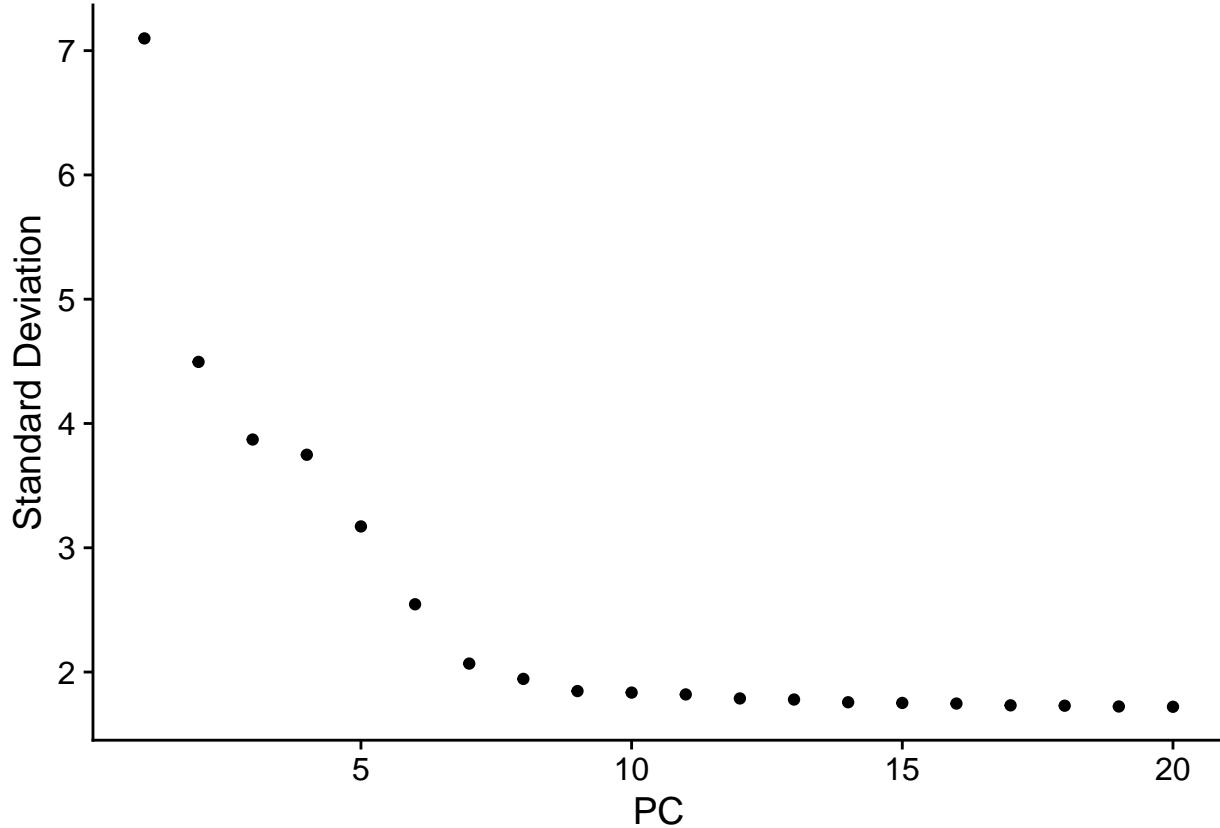
PC_14

PC_15

hysteric method generates an elbow plot are ranking of principal component based on the percentage of variance explained by each one in this example we can observe an elbow around PC 9 to 10 suggesting that

the majority of the two signal is captured in the first 10 PCs

```
ElbowPlot(pbmc)
```



After PC 10 the remaining components contribute much less to the overall values in all of the data

```
pbmc <- FindNeighbors(pbmc, dims = 1:10)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

```
pbmc <- FindClusters(pbmc, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
```

```
##
```

```
## Number of nodes: 2638
```

```
## Number of edges: 95927
```

```
##
```

```
## Running Louvain algorithm...
```

```
## Maximum modularity in 10 random starts: 0.8728
```

```
## Number of communities: 9
```

```
## Elapsed time: 0 seconds
```

```
# Cluster Id of the top5 cells
```

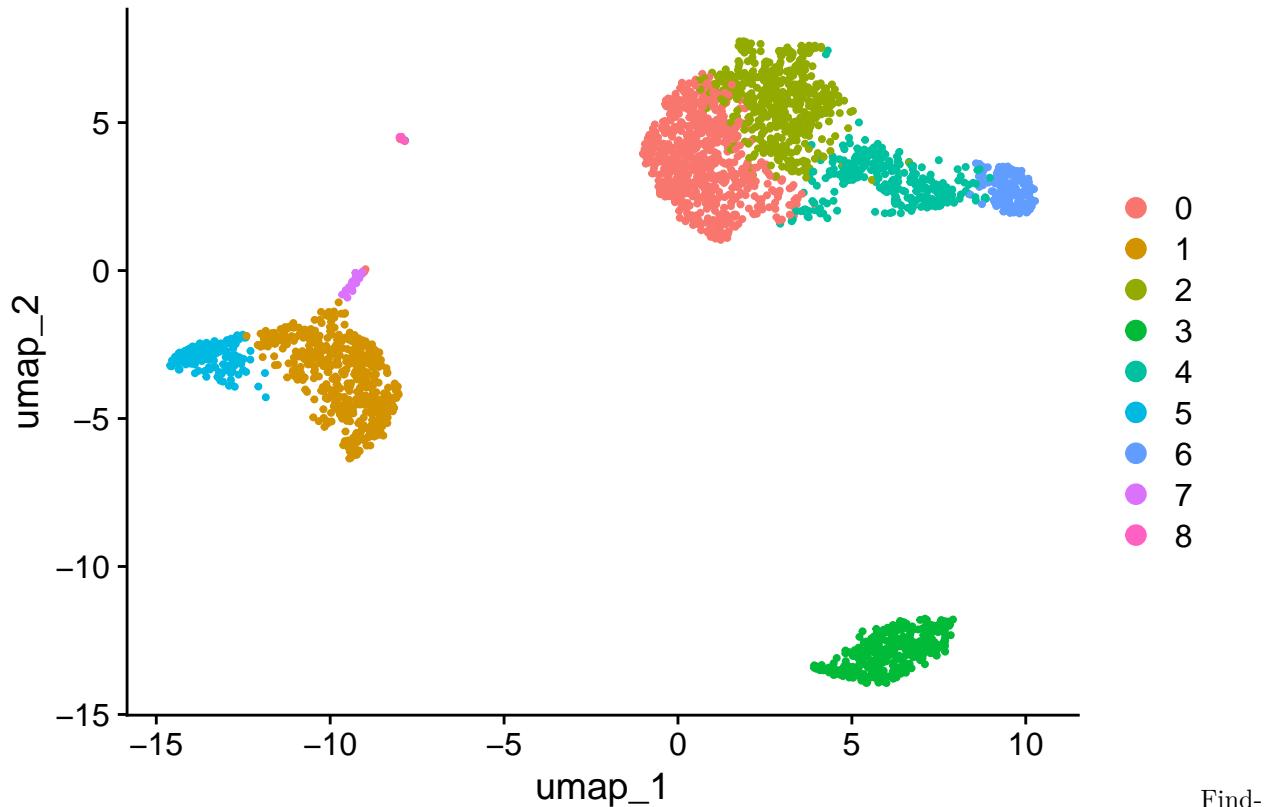
```
head(Ids(pbmc),5)
```

```
## AAACATACAACCAC-1 AAACATTGAGCTAC-1 AAACATTGATCAGC-1 AAACCGTGCTTCCG-1
```

```
## 2 3 2 1
```

```
## AAACCGTGTATGCG-1
```

```
## 6
```



ing differentially expressed features(Cluster Bio markers) Find markers that define clusters by there differntial expression, by default compares the positive and negative of a arkers of a single cluster, compared to all other cells

```
cluster2.markers <- FindMarkers(pbmc, ident.1 = 2)

## For a (much!) faster implementation of the Wilcoxon Rank Sum Test,
## (default method for FindMarkers) please install the presto package
## -----
## install.packages('devtools')
## devtools::install_github('immunogenomics/presto')
## -----
## After installation of presto, Seurat will automatically use the more
## efficient implementation (no further action necessary).
## This message will be shown once per session

head(cluster2.markers, n= 5)

##           p_val avg_log2FC pct.1 pct.2     p_val_adj
## IL32 2.892340e-90  1.3070772 0.947 0.465 3.966555e-86
## LTB  1.060121e-86  1.3312674 0.981 0.643 1.453850e-82
## CD3D 8.794641e-71  1.0597620 0.922 0.432 1.206097e-66
## IL7R 3.516098e-68  1.4377848 0.750 0.326 4.821977e-64
## LDHB 1.642480e-67  0.9911924 0.954 0.614 2.252497e-63
```

A very small p value close to 0 suggest that the gene is likely differentially expressed between the specified cluster and the other cells. Average log fold change is the average fold change of the gene expression between the clusters of interest and other cells, Positive values indicate higher expression in the cluster compared to the other cells

```
#Markers distinguishing cluster 5 from cluster 0 and 3
cluster5.marker <- FindMarkers(pbmc, ident.1 = 5, ident.2 = c(0,3))
head(cluster5.marker, n=5)

##           p_val avg_log2FC pct.1 pct.2     p_val_adj
## FCGR3A 8.246578e-205  6.794969 0.975 0.040 1.130936e-200
## IFITM3 1.677613e-195  6.192558 0.975 0.049 2.300678e-191
## CFD   2.401156e-193  6.015172 0.938 0.038 3.292945e-189
## CD68  2.900384e-191  5.530330 0.926 0.035 3.977587e-187
## RP11-290F20.3 2.513244e-186  6.297999 0.840 0.017 3.446663e-182

# Markers of every cluster compared to all the remaining cells and reporting only positive
pbmc.markers <- FindAllMarkers(pbmc, only.pos= TRUE)

## Calculating cluster 0
## Calculating cluster 1
## Calculating cluster 2
## Calculating cluster 3
## Calculating cluster 4
## Calculating cluster 5
## Calculating cluster 6
## Calculating cluster 7
## Calculating cluster 8
```

```

pbmc.markers %>%
  group_by(cluster) %>%
  dplyr::filter(avg_log2FC>1)

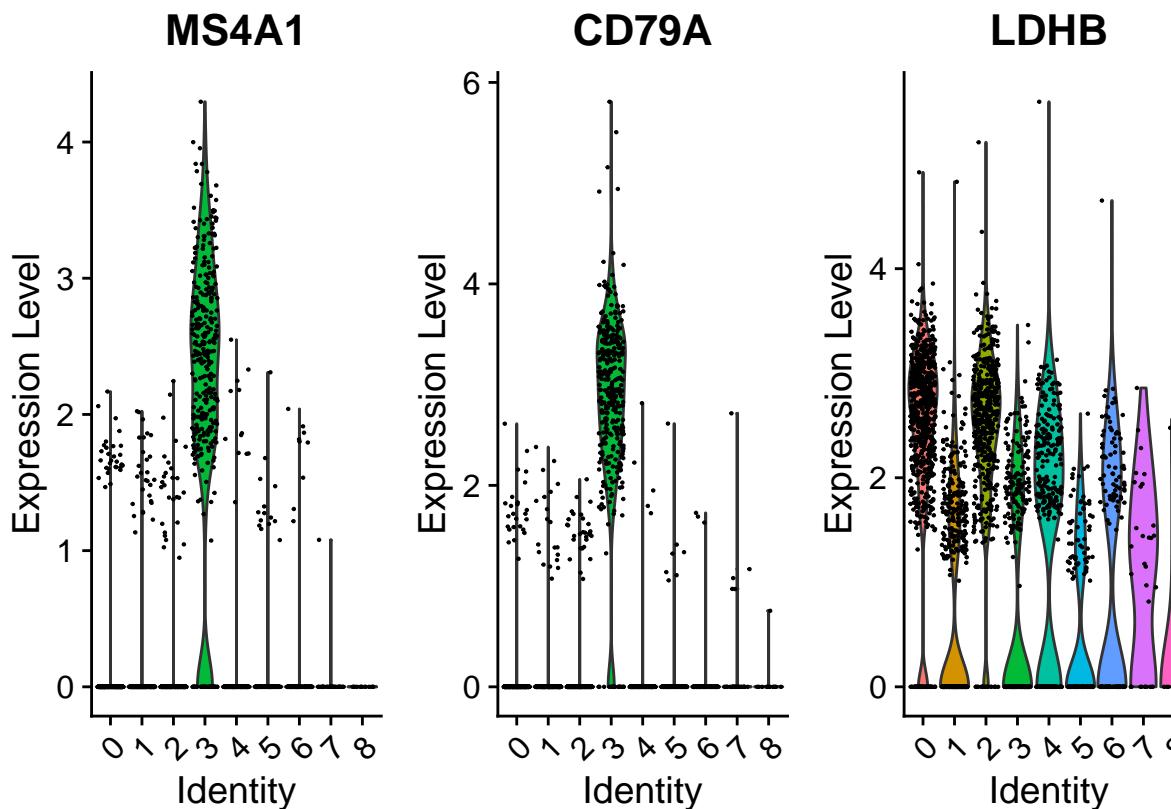
## # A tibble: 7,019 x 7
## # Groups:   cluster [9]
##       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##       <dbl>      <dbl> <dbl> <dbl>      <dbl> <fct> <chr>
## 1 3.75e-112    1.21  0.912  0.592  5.14e-108  0     LDHB
## 2 9.57e- 88    2.40  0.447  0.108  1.31e- 83  0     CCR7
## 3 1.15e- 76    1.06  0.845  0.406  1.58e- 72  0     CD3D
## 4 1.12e- 54    1.04  0.731  0.4     1.54e- 50  0     CD3E
## 5 1.35e- 51    2.14  0.342  0.103  1.86e- 47  0     LEF1
## 6 1.94e- 47    1.20  0.629  0.359  2.66e- 43  0     NOSIP
## 7 2.81e- 44    1.53  0.443  0.185  3.85e- 40  0     PIK3IP1
## 8 6.27e- 43    1.99  0.33   0.112  8.60e- 39  0     PRKCQ-AS1
## 9 1.16e- 40    2.70  0.2    0.04   1.59e- 36  0     FHIT
## 10 1.34e- 34   1.96  0.268  0.087  1.84e- 30  0     MAL
## # i 7,009 more rows

```

The below code finds genes that are differentially expressed in cluster 1. min.pct = 0.25 ensures that a gene is expressed in at least 25% of cells.

```
cluster0.markers <- FindMarkers(pbmc, ident.1 = 0, logfc.threshold=0.25, test.use="roc", only.pos= TRUE)
```

```
#Distribution of gene expression levels
VlnPlot(pbmc, features = c("MS4A1", "CD79A", "LDHB"))
```



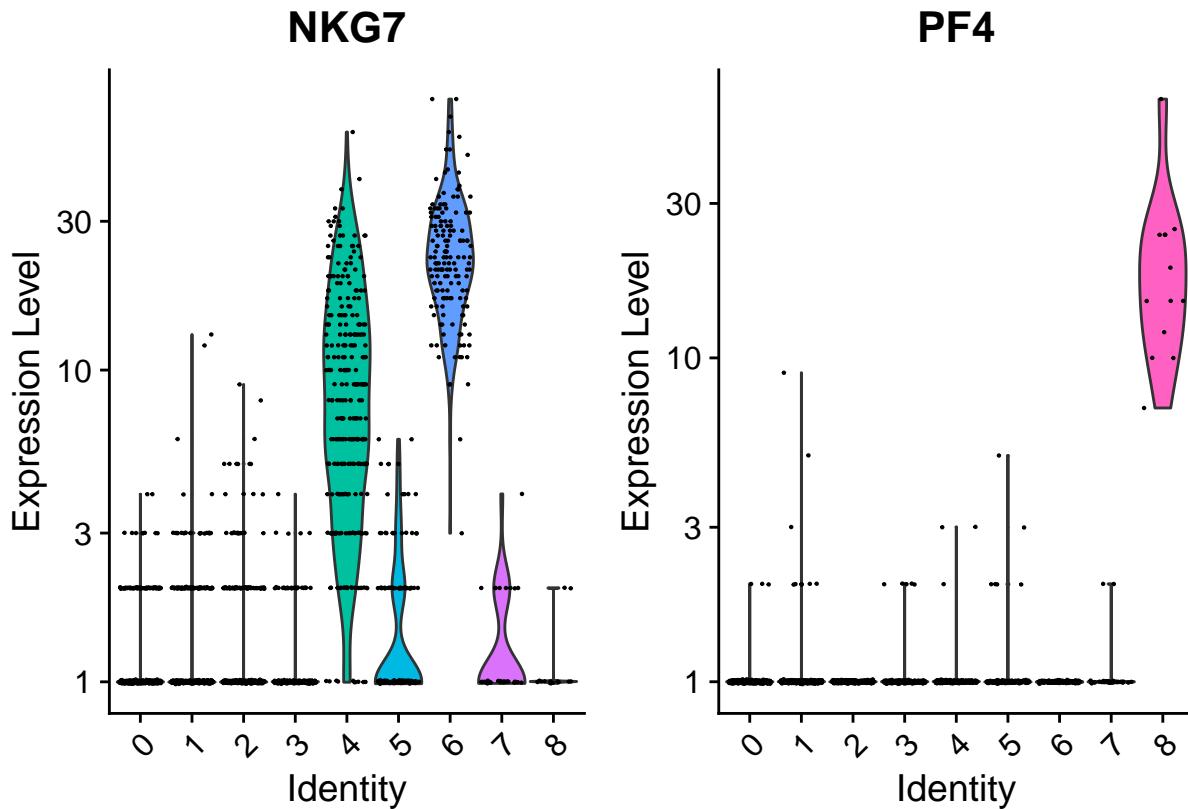
The width of the violin at any given expression level indicates the density of cells expressing the gene at that level. wider section - more cells have that expression level, narrower section means fewer cells

```

# View absolute gene expression levels and compare them across clusters
# with raw counts
VlnPlot(pbmc, features = c("NKG7", "PF4"), slot = "counts", log = TRUE)

## Warning: The `slot` argument of `VlnPlot()` is deprecated as of Seurat 5.0.0.
## i Please use the `layer` argument instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

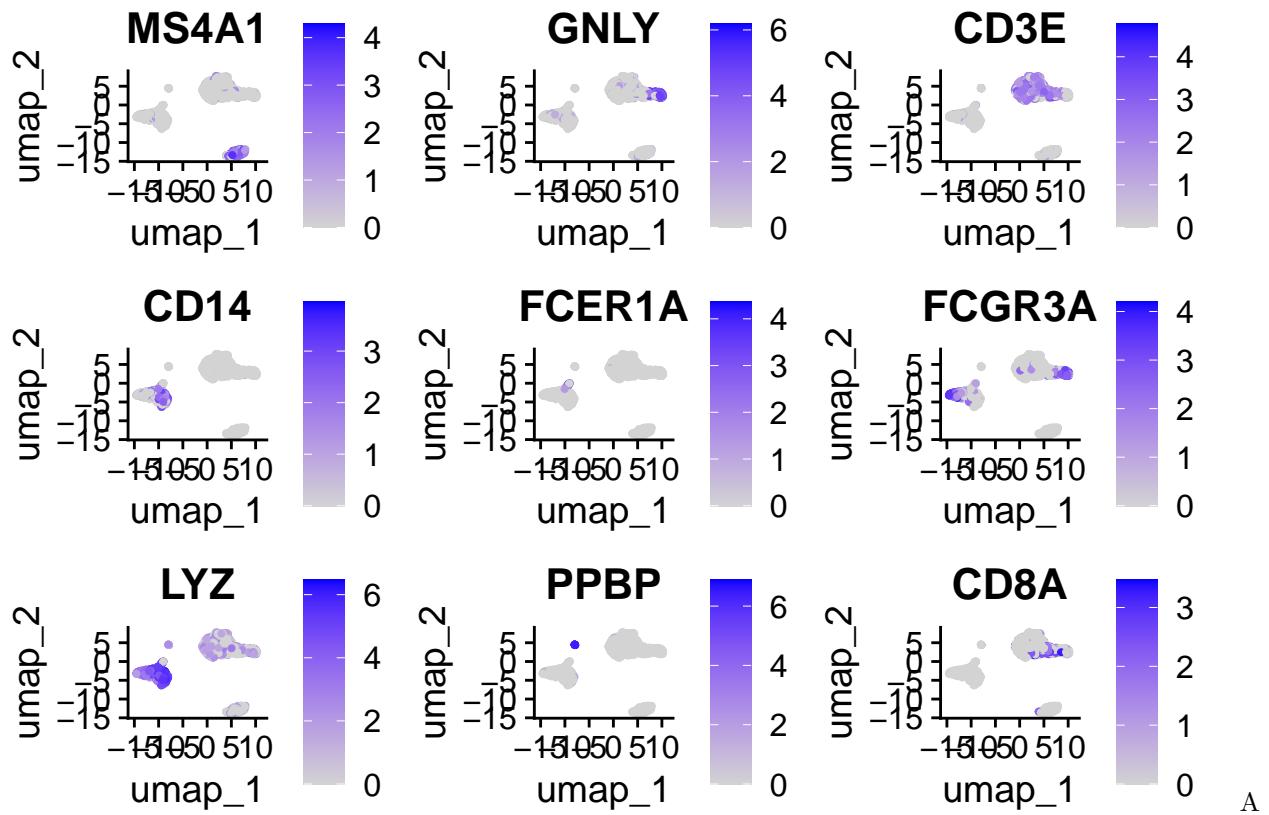
```



```

FeaturePlot(pbmc, features = c("MS4A1", "GNLY", "CD3E", "CD14", "FCER1A", "FCGR3A", "LYZ", "PPBP", "CD8A"))

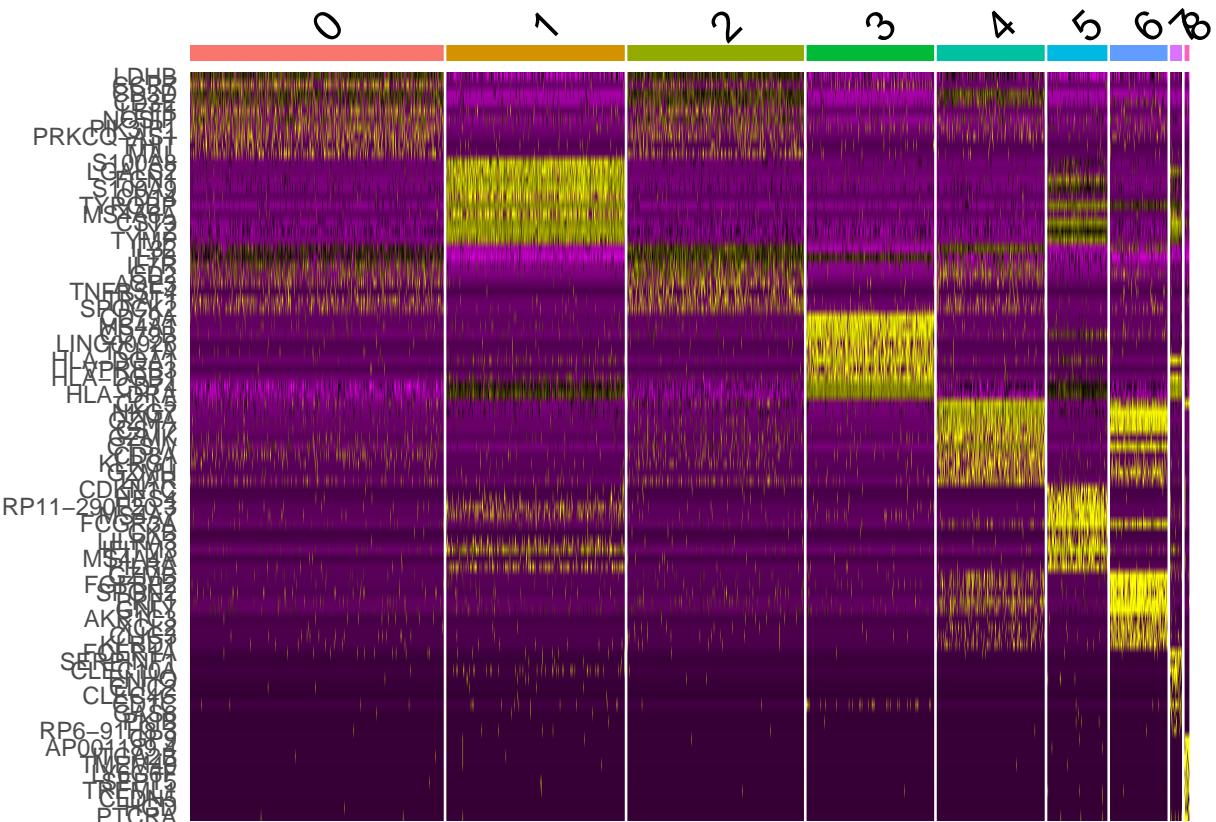
```



feature plot can display multiple genes at once.

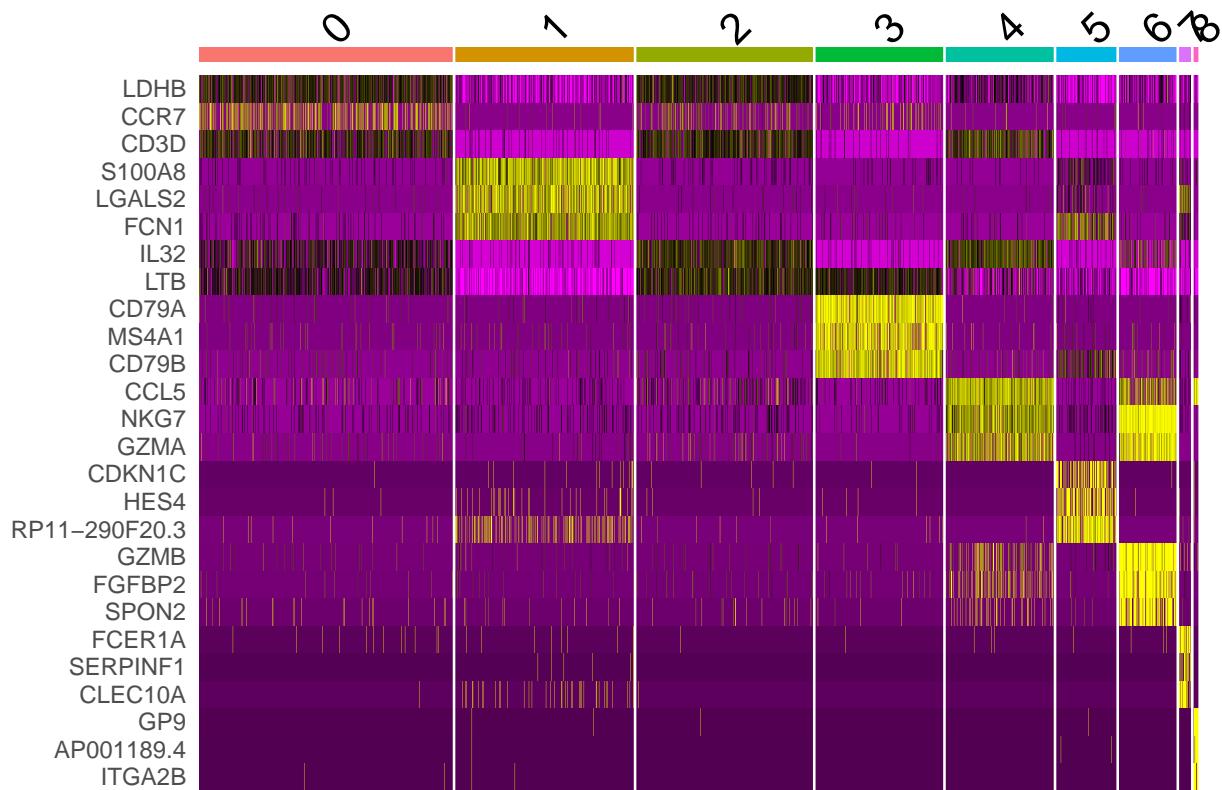
Top 10 markers of each cluster.

```
pbmc.markers%>%
  group_by(cluster)%>%
  dplyr::filter(avg_log2FC>1)%>%
  slice_head(n=10)%>%
  ungroup() ->top10
DoHeatmap(pbmc, features = top10$gene) +NoLegend()
```



For top 3 genes (to visualize better)

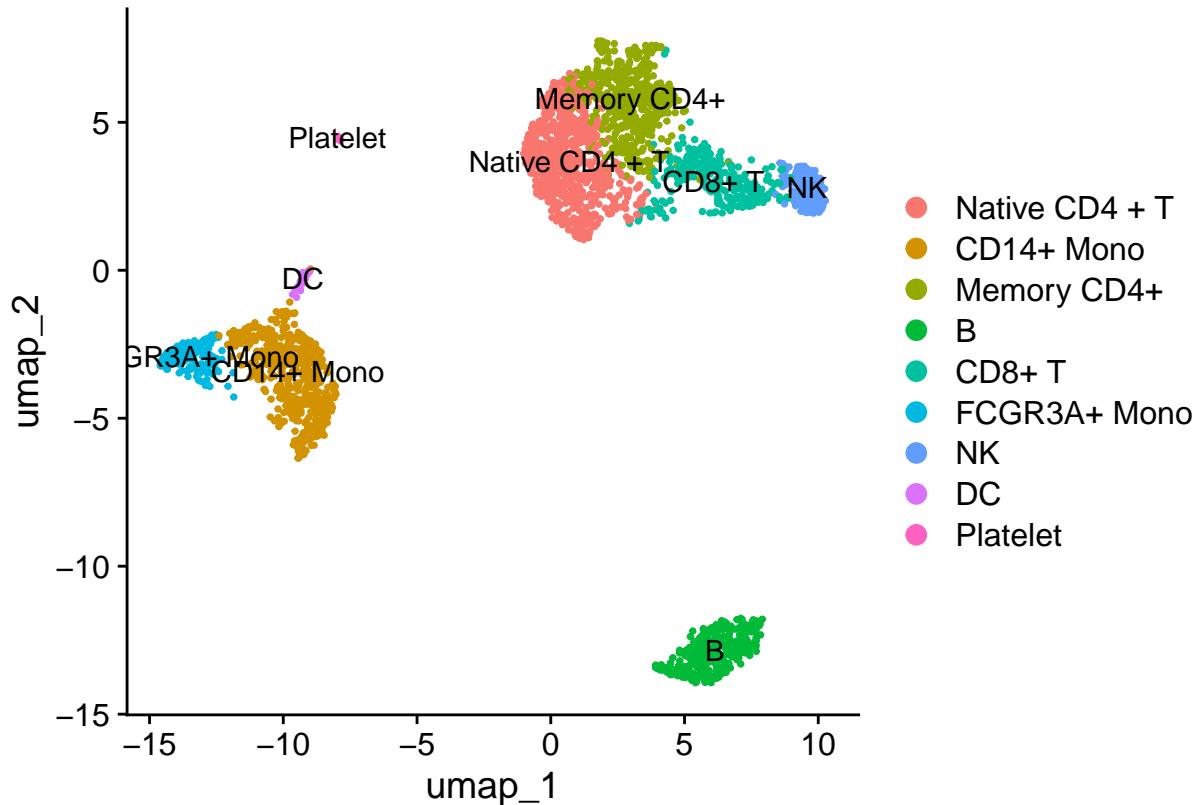
```
pbmc.markers%>%
  group_by(cluster)%>%
  dplyr::filter(avg_log2FC>1)%>%
  slice_head(n=3)%>%
  ungroup() ->top3
DoHeatmap(pbmc, features = top3 $gene) +NoLegend()
```



Assigning cell type identity to clusters Using Canonical Markers:

Cluster ID Markers cell type: 0 IL7R, CCR7 Native CD4 + T 1 CD14, LYZ CD14+ Mono 2 IL7R, S100A4 Memory CD4+ 3 MS4A1 B 4 CD8A CD8+ T 5 FCGR3A, MS4A7 FCGR3A+ Mono 6 GNLY, NKG7 NK 7 FCER1A, CST3 DC 8 PPBP Platelet

```
new.cluster.ids <- c("Native CD4 + T", "CD14+ Mono", "Memory CD4+", "B", "CD8+ T", "FCGR3A+ Mono", "NK", "DC", "Platelet")
names(new.cluster.ids) <- levels(pbmc)
pbmc <- RenameIds(pbmc, new.cluster.ids)
DimPlot(pbmc, reduction = "umap", label = TRUE, pt.size = 0.5)
```



Results and Conclusion

The UMAP plot presented here shows the clustering of different immune cell populations based on single-cell RNA sequencing (scRNA-seq) data. Each cluster represents a distinct immune cell type, including **T cells** (CD4+ and CD8+), **B cells**, **monocytes** (CD14+ and FCGR3A+), **natural killer (NK) cells**, **dendritic cells (DC)**, and **platelets**. The separation of these clusters in the UMAP space indicates that the transcriptomic profiles of these cell types are distinct.

- **T Cell Populations:** The **Native CD4+ T cells (red)** and **Memory CD4+ T cells (dark green)** are closely clustered, suggesting a shared transcriptomic profile with some variability. The **CD8+ T cells (cyan)** are distinct but remain in proximity, reflecting their functional relationship with CD4+ T cells.
- **B Cells:** The **B cells (green)** form a well-defined cluster, separated from the T cell populations, indicating distinct gene expression profiles.
- **Monocytes:** The **CD14+ Monocytes (brown)** and **FCGR3A+ Monocytes (blue)** form two separate clusters, highlighting differences in their functional states.
- **Dendritic Cells (DCs):** The **DC cluster (purple)** is relatively small and positioned near monocytes, consistent with their role as antigen-presenting cells.
- **NK Cells and Platelets:** **NK cells (periwinkle)** form a distinct cluster close to the T cell populations, reflecting their shared immune function. **Platelets (pink)** are the most distinct population, positioned far from the others, indicating a significantly different transcriptomic signature.

Conclusion

The UMAP visualization effectively differentiates immune cell types based on gene expression patterns. The clear clustering suggests that the scRNA-seq data successfully captures distinct transcriptomic signatures, supporting accurate cell-type identification. The separation between adaptive immune cells (T and B cells) and innate immune cells (monocytes, DCs, and NK cells) aligns with their expected biological roles, validating the quality of the data set.

References

-(N.d.). Retrieved from <https://www.youtube.com/watch?v=ngTiPKA8kMA> -Datasets. (n.d.). Retrieved from <https://www.10xgenomics.com/datasets?configure%5BhitsPerPage%5D=50&configure%5BmaxValue%5D=1000> -Tools for Single Cell Genomics. (n.d.). Retrieved from <https://satijalab.org/seurat/>