seurat

Zitao Zhang

2024-11-19

```
# Load the PBMC dataset
pbmc.data <- Read10X(data.dir = "hg19/")
# Initialize the Seurat object with the raw (non-normalized data).
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc3k", min.cells = 3, min.features = 200)
pbmc
```

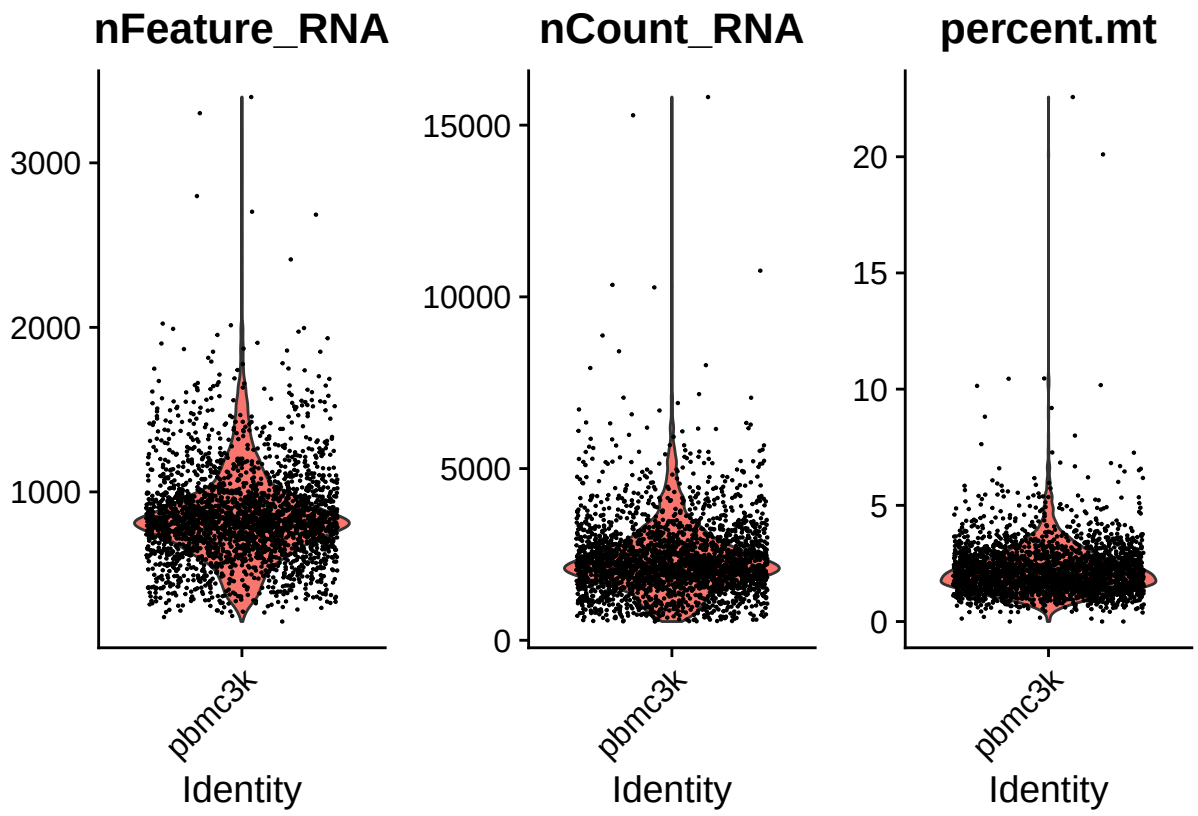
```
## An object of class Seurat
## 13714 features across 2700 samples within 1 assay
## Active assay: RNA (13714 features, 0 variable features)
##  1 layer present: counts
```

QC and selecting cells for further analysis

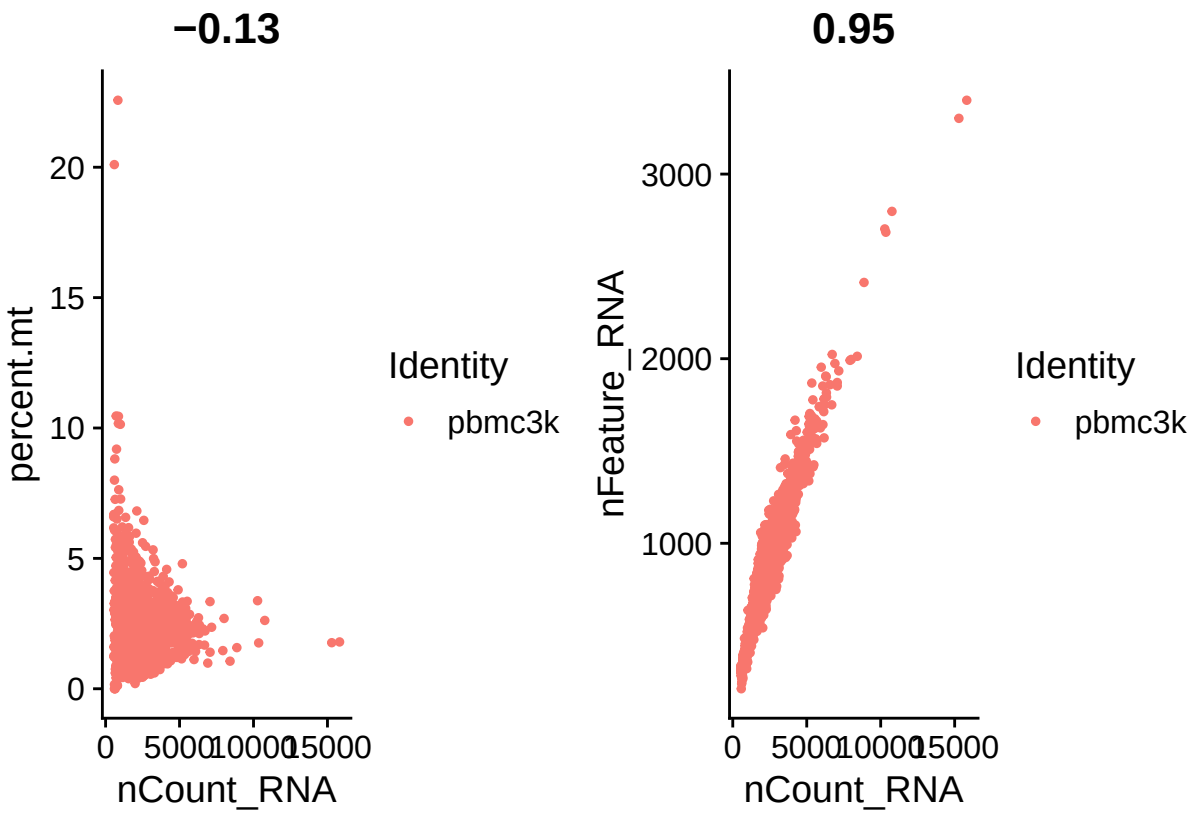
```
# The [[ operator can add columns to object metadata. This is a great place to stash QC stats
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")

# Show QC metrics for the first 5 cells
head(pbmc@meta.data, 5)
```

```
##                  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
## AAACCGTGCTTCCG-1     pbmc3k       2639          960  1.7430845
## AAACCGTGTATGCG-1     pbmc3k        980          521  1.2244898
```

```
# Visualize QC metrics as a violin plot
VlnPlot(pbmc, 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.

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




```
pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)
```

Normalizing the data

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

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

```
# pbmc <- NormalizeData(pbmc)
```

Identification of highly variable features (feature selection)

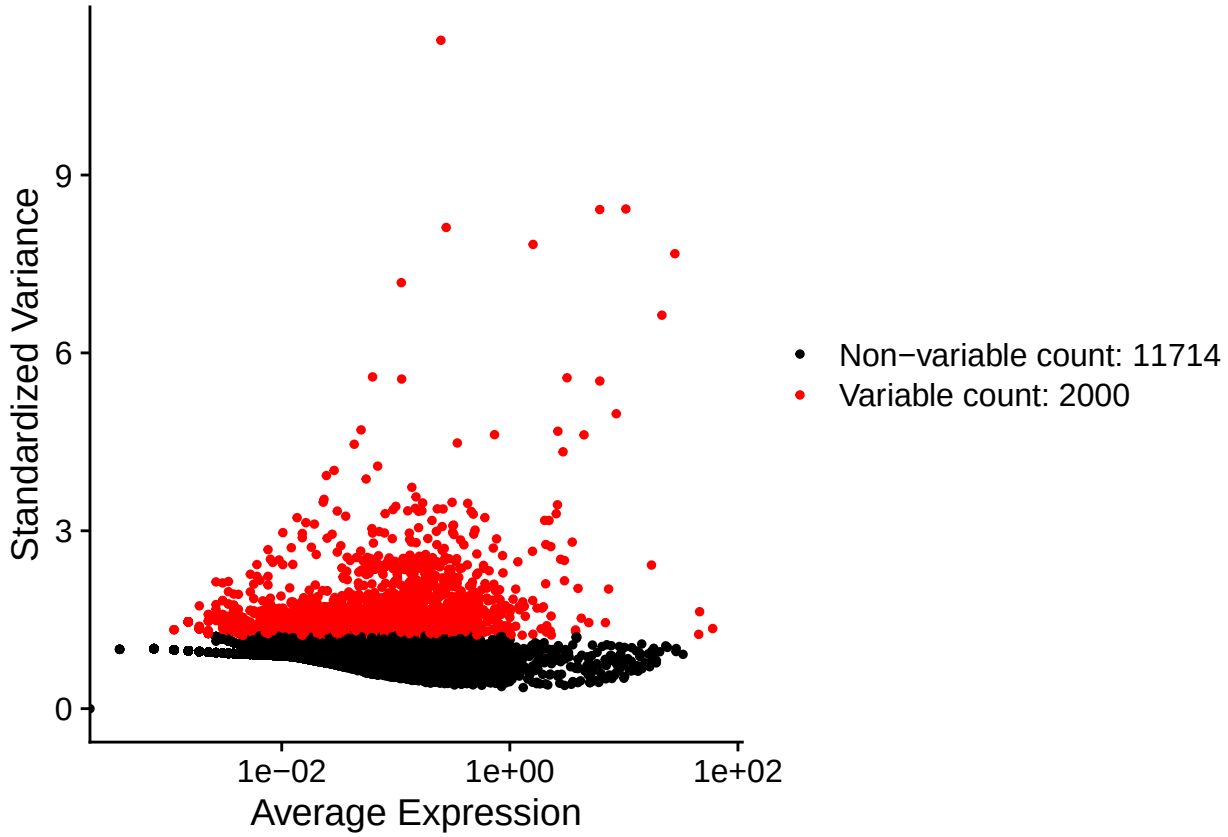
```
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
```

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

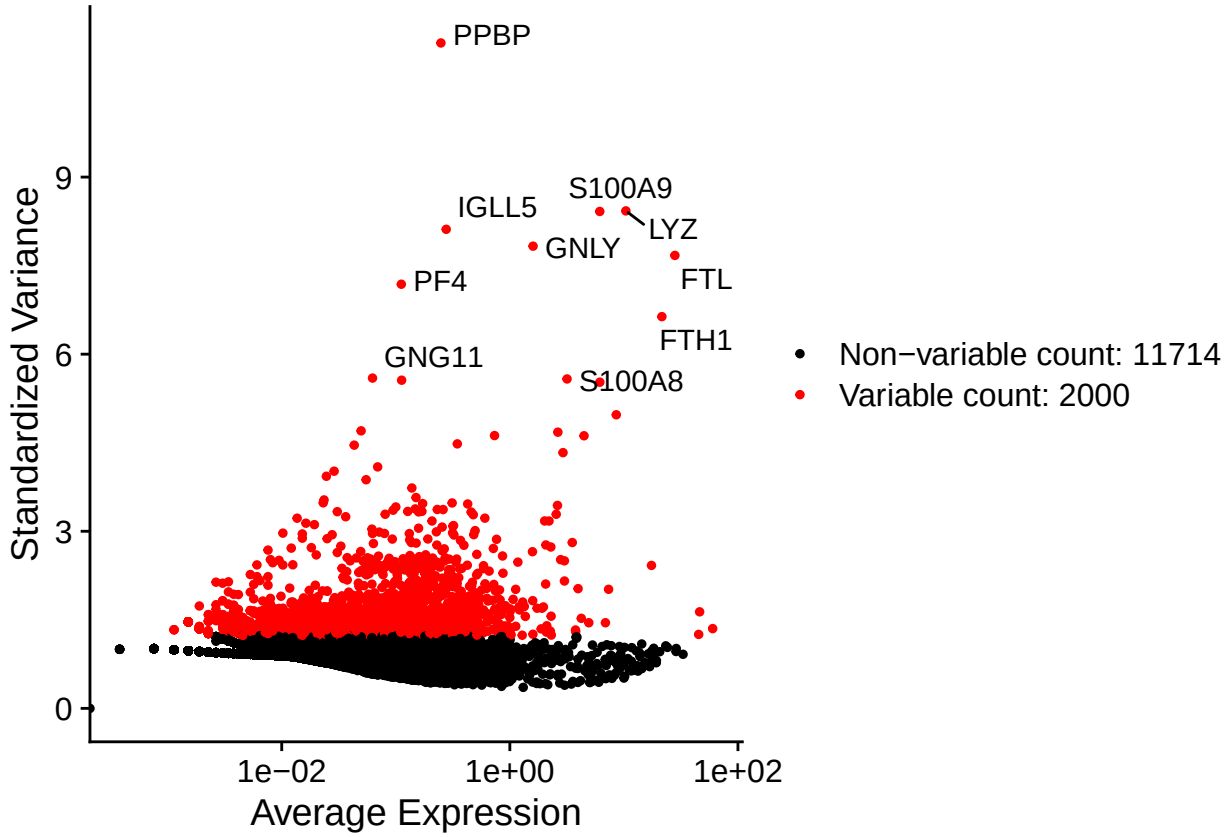
```r
# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(pbmc), 10)

# plot 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
```

```r
plot1
```




```r
plot2
```



Scaling the data

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

```
## Centering and scaling data matrix
```

```
pbmc <- ScaleData(pbmc, vars.to.regress = "percent.mt")
```

```
## Regressing out percent.mt

## Centering and scaling data matrix
```

Perform linear dimensional reduction

```
pbmc <- RunPCA(pbmc, features = VariableFeatures(object = pbmc))
```

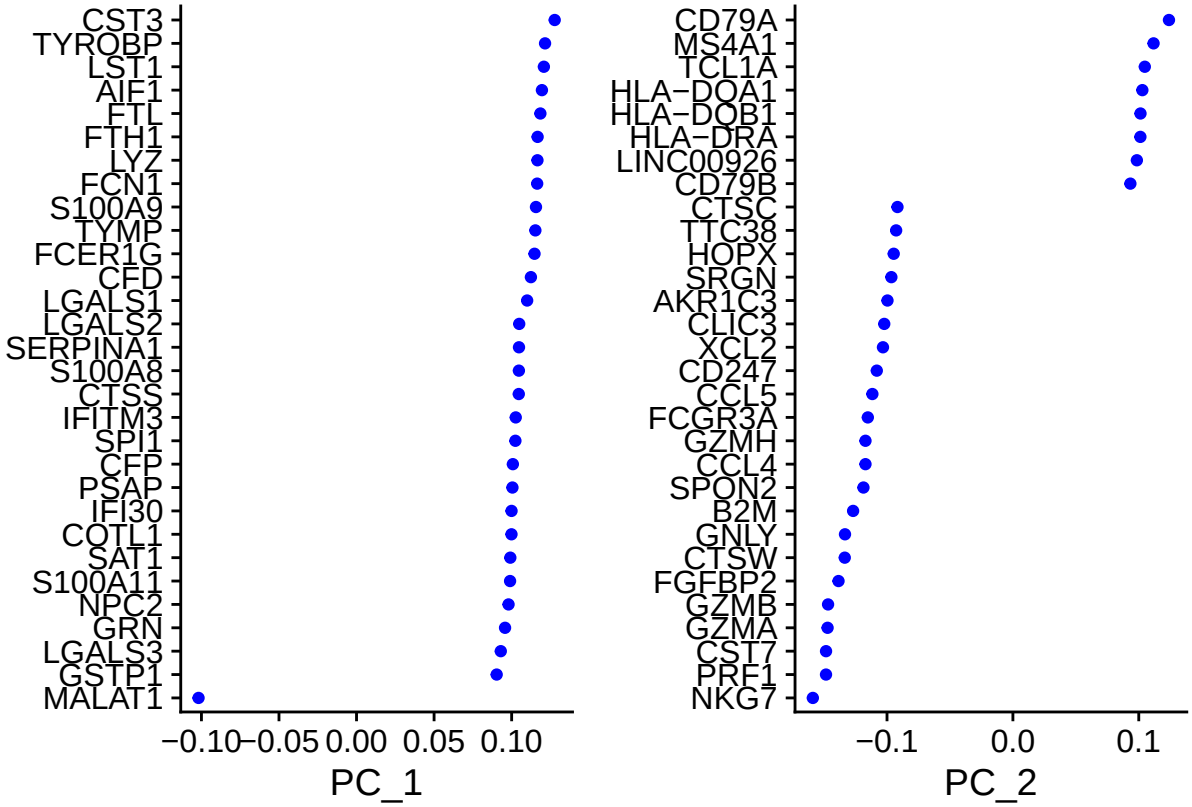
```
## PC_ 1
```

```
## Positive:  CST3, TYROBP, LST1, AIF1, FTL, FTH1, LYZ, FCN1, S100A9, TYMP
##     FCER1G, CFD, LGALS1, LGALS2, SERPINA1, S100A8, CTSS, IFITM3, SPI1, CFP
##     PSAP, IFI30, COTL1, SAT1, S100A11, NPC2, GRN, LGALS3, GSTP1, PYCARD
## Negative:  MALAT1, LTB, IL32, IL7R, CD2, B2M, ACAP1, CTSW, STK17A, CD27
##     CD247, CCL5, GIMAP5, GZMA, AQP3, CST7, TRAF3IP3, SELL, GZMK, HOPX
##     MAL, MYC, ITM2A, ETS1, LYAR, GIMAP7, KLRG1, NKG7, ZAP70, BEX2
## 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, GZMA, GZMB, FGFBP2, CTSW, GNLY, B2M, SPON2
##     CCL4, GZMH, FCGR3A, CCL5, CD247, XCL2, CLIC3, AKR1C3, SRGN, HOPX
##     TTC38, CTSC, APMAP, S100A4, IGFBP7, ANXA1, ID2, IL32, XCL1, RHOC
## PC_ 3
## Positive:  HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPA1, HLA-DPB1, 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, IGLL5, SWAP70, P2RX5, LAT2, FCGR3A
## 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, CMTM5, MPP1, MYL9, RP11-367G6.3, GP1BA
## PC_ 4
## Positive:  HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1, CD74, HLA-DPB1, HIST1H2AC, PF4, HLA-DRB1
##     HLA-DPA1, SDPR, TCL1A, HLA-DQA2, HLA-DRA, PPBP, LINC00926, GNG11, HLA-DRB5, SPARC
##     GP9, PTCRA, AP001189.4, CA2, CD9, NRGN, RGS18, CLU, TUBB1, GZMB
## Negative:  VIM, IL7R, S100A6, S100A8, IL32, S100A4, GIMAP7, S100A10, S100A9, MAL
##     AQP3, CD14, CD2, LGALS2, FYB, GIMAP4, ANXA1, RBP7, CD27, FCN1
##     LYZ, S100A12, MS4A6A, GIMAP5, S100A11, FOLR3, TRABD2A, AIF1, IL8, IFI6
## PC_ 5
## Positive:  GZMB, FGFBP2, S100A8, NKG7, GNLY, CCL4, PRF1, CST7, SPON2, GZMA
##     GZMH, LGALS2, S100A9, CCL3, XCL2, CD14, CLIC3, CTSW, MS4A6A, S100A12
##     GSTP1, RBP7, IGFBP7, FOLR3, AKR1C3, TYROBP, CCL5, TTC38, XCL1, APMAP
## Negative:  LTB, IL7R, CKB, MS4A7, RP11-290F20.3, AQP3, SIGLEC10, VIM, CYTIP, HMOX1
##     LILRB2, PTGES3, HN1, CD2, CD27, FAM110A, ANXA5, CTD-2006K23.1, MAL, VMO1
##     CORO1B, TUBA1B, LILRA3, GDI2, TRADD, IL32, ATP1A1, ABRACL, CCDC109B, PPA1
```

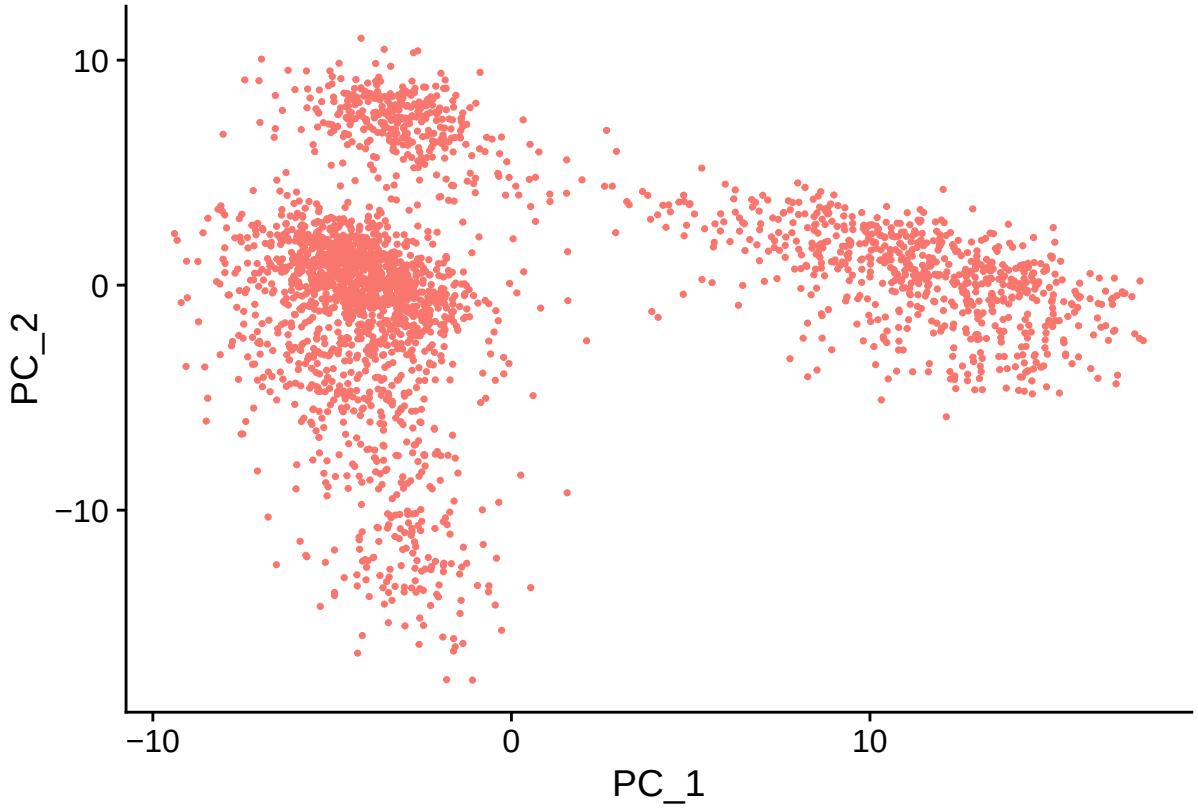
```
# Examine and visualize PCA results a few different ways
print(pbmc[["pca"]], dims = 1:5, nfeatures = 5)
```

```
## 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, GZMA, GZMB
## PC_ 3
## Positive:  HLA-DQA1, CD79A, CD79B, HLA-DQB1, HLA-DPA1
## Negative:  PPBP, PF4, SDPR, SPARC, GNG11
## PC_ 4
## Positive:  HLA-DQA1, CD79B, CD79A, MS4A1, HLA-DQB1
## Negative:  VIM, IL7R, S100A6, S100A8, IL32
## PC_ 5
## Positive:  GZMB, FGFBP2, S100A8, NKG7, GNLY
## Negative:  LTB, IL7R, CKB, MS4A7, RP11-290F20.3
```

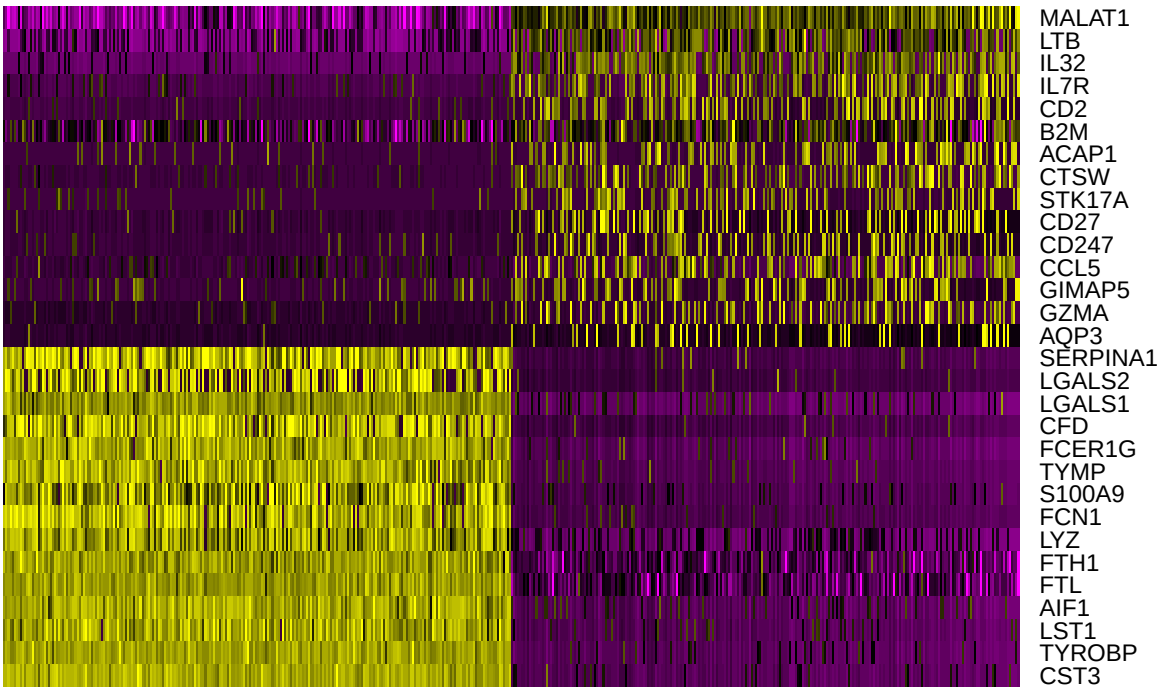
```
VizDimLoadings(pbmc, dims = 1:2, reduction = "pca")
```



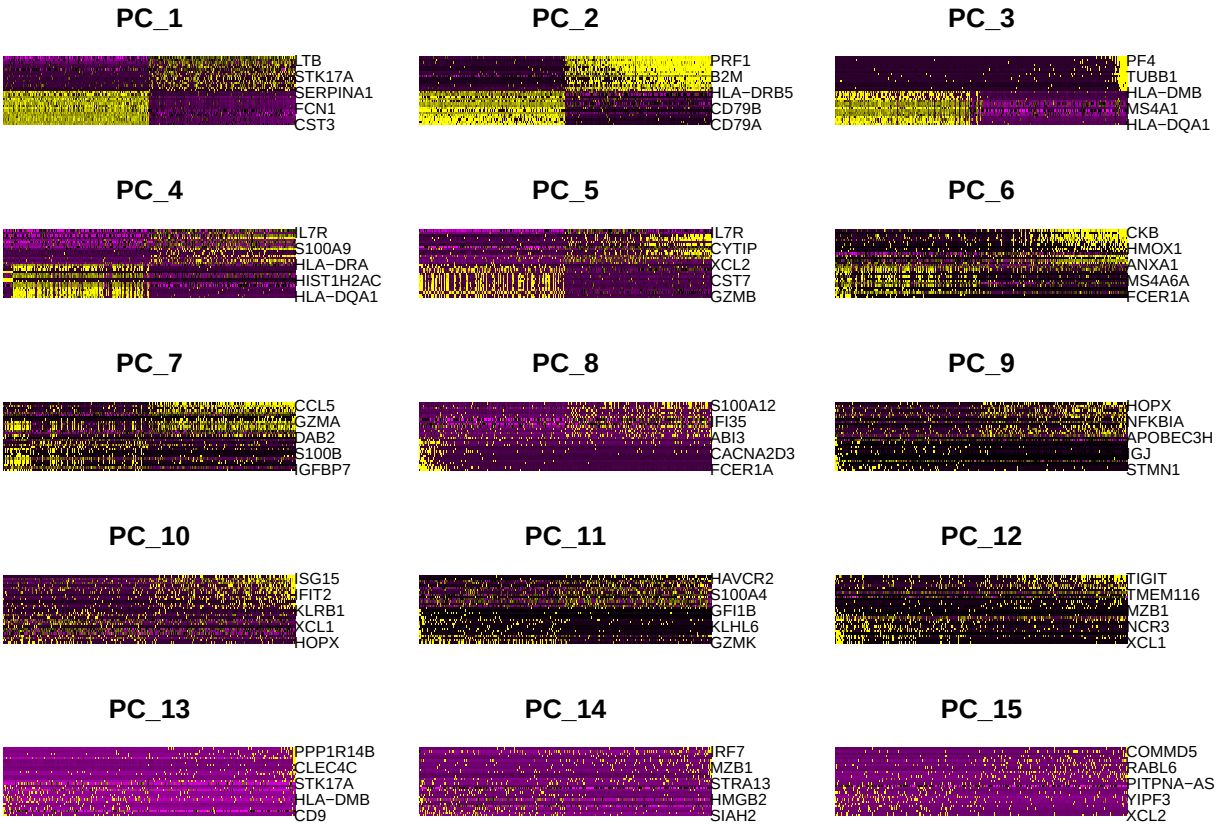

```
DimPlot(pbmc, reduction = "pca") + NoLegend()
```




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

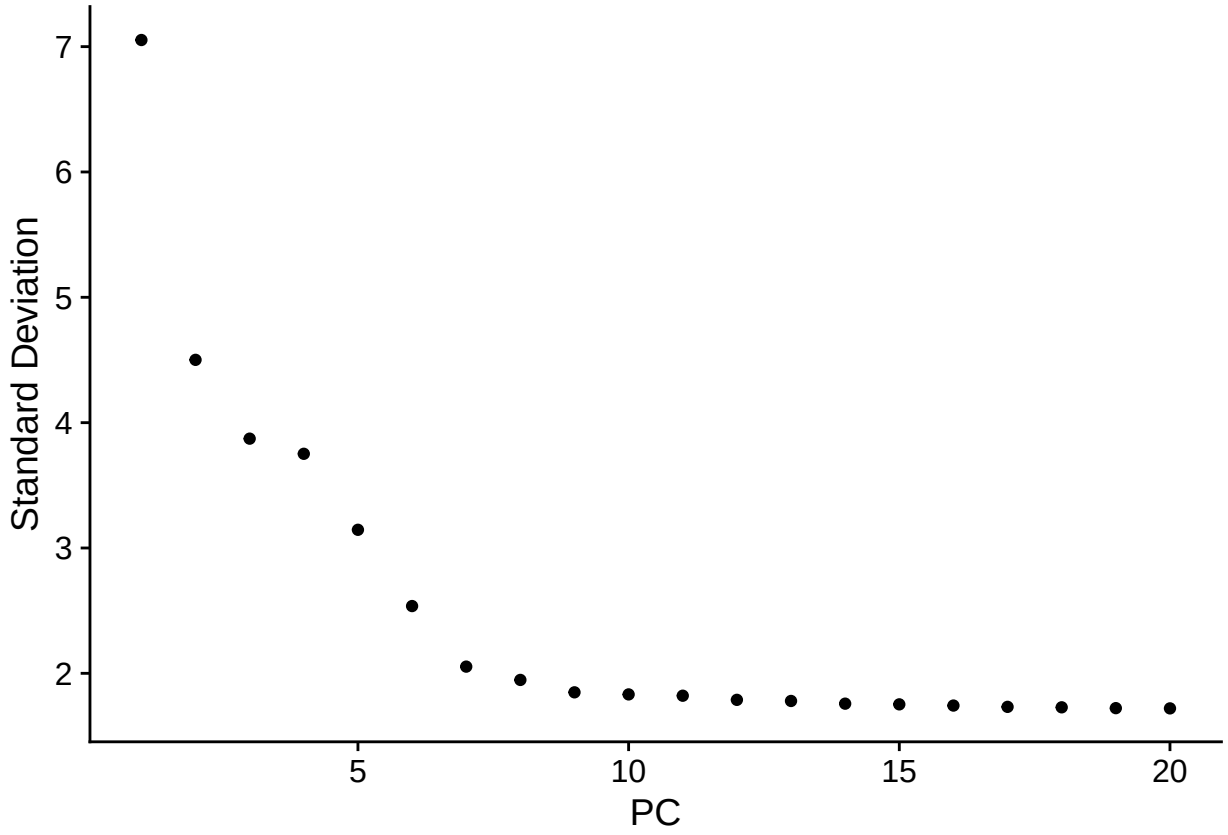



```
DimHeatmap(pbmc, dims = 1:15, cells = 500, balanced = TRUE)
```



Determine the 'dimensionality' of the dataset

```
ElbowPlot(pbmc)
```



Cluster the cells

```
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: 95905
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8735
## Number of communities: 9
## Elapsed time: 0 seconds
```

```r
# Look at cluster IDs of the first 5 cells
head(Idents(pbmc), 5)
```

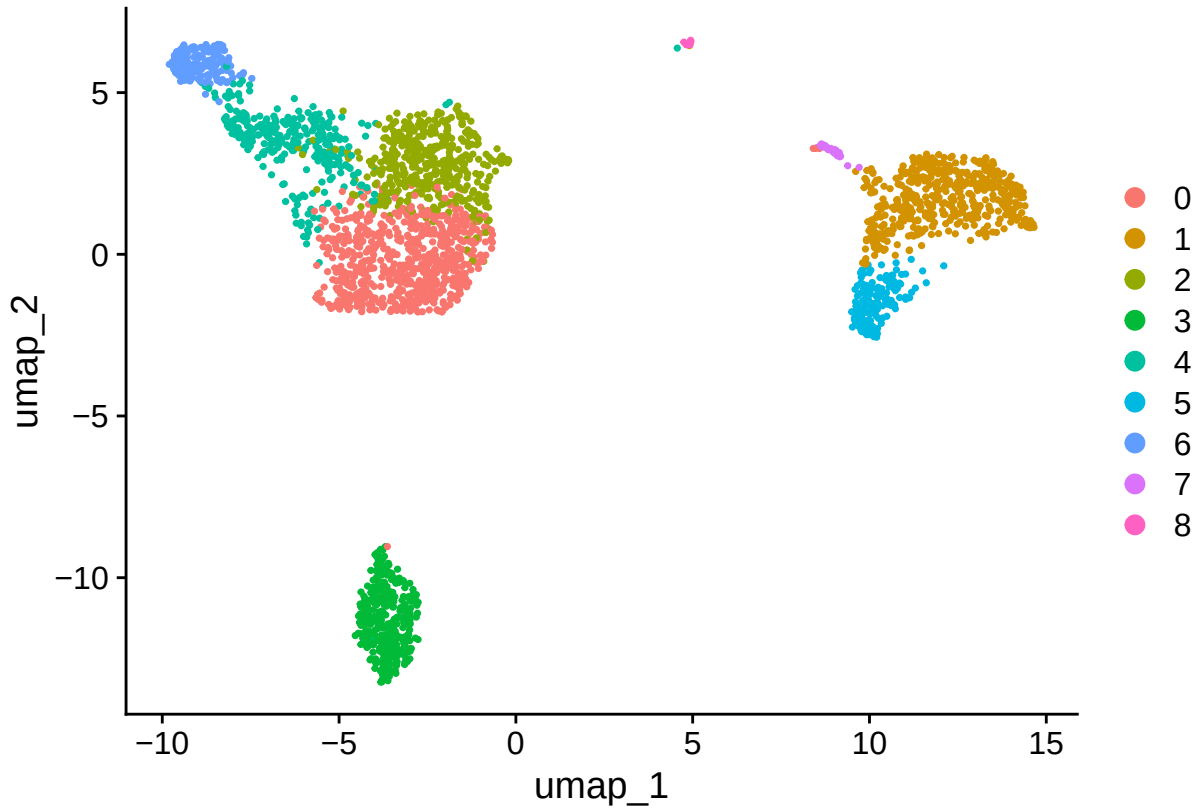
```
## AAACATACAACCAC-1 AAACATTGAGCTAC-1 AAACATTGATCAGC-1 AAACCGTGCTTCCG-1
##                0                3                2                1
## AAACCGTGTATGCG-1
##                6
## Levels: 0 1 2 3 4 5 6 7 8
```

Run non-linear dimensional reduction (UMAP/tSNE)

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

```
## 15:35:37 UMAP embedding parameters a = 0.9922 b = 1.112

## 15:35:37 Read 2638 rows and found 10 numeric columns

## 15:35:37 Using Annoy for neighbor search, n_neighbors = 30

## 15:35:37 Building Annoy index with metric = cosine, n_trees = 50

## 0%   10   20   30   40   50   60   70   80   90   100%

## [----|----|----|----|----|----|----|----|----|----|

## **************************************************|
## 15:35:37 Writing NN index file to temp file /var/folders/sk/sy0hv6gd795c1z2pgpd2ngcr0000gn/T//RtmphW
## 15:35:37 Searching Annoy index using 1 thread, search_k = 3000
## 15:35:37 Annoy recall = 100%
## 15:35:38 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 15:35:38 Initializing from normalized Laplacian + noise (using RSpectra)
## 15:35:38 Commencing optimization for 500 epochs, with 106310 positive edges
## 15:35:40 Optimization finished
```

```r
# note that you can set `label = TRUE` or use the LabelClusters function to help label
# individual clusters
DimPlot(pbmc, reduction = "umap")
```




```
saveRDS(pbmc, file = "pbmc_tutorial.rds")
```

Finding differentially expressed features (cluster biomarkers)

```
# find all markers of cluster 2
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
## LTB  1.709675e-83   1.330256 0.982 0.647 2.344648e-79
## IL32 5.076510e-83   1.242930 0.947 0.471 6.961926e-79
```

```
## LDHB 2.467055e-68   1.044820 0.967 0.615 3.383320e-64
## CD3D 1.817480e-66   1.058609 0.920 0.438 2.492492e-62
## IL7R 8.698894e-61   1.389909 0.744 0.333 1.192966e-56
```

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

```
##                       p_val avg_log2FC pct.1 pct.2     p_val_adj
## FCGR3A        5.972471e-204   6.795991 0.975 0.041 8.190647e-200
## IFITM3        5.671364e-195   6.201036 0.975 0.048 7.777708e-191
## CFD           2.389538e-193   6.081028 0.937 0.038 3.277012e-189
## CD68          1.800066e-189   5.472200 0.925 0.036 2.468611e-185
## RP11-290F20.3 6.852416e-189   6.390800 0.843 0.015 9.397404e-185
```

```
# find markers for every cluster compared to all remaining cells, report only the positive
# ones
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
```

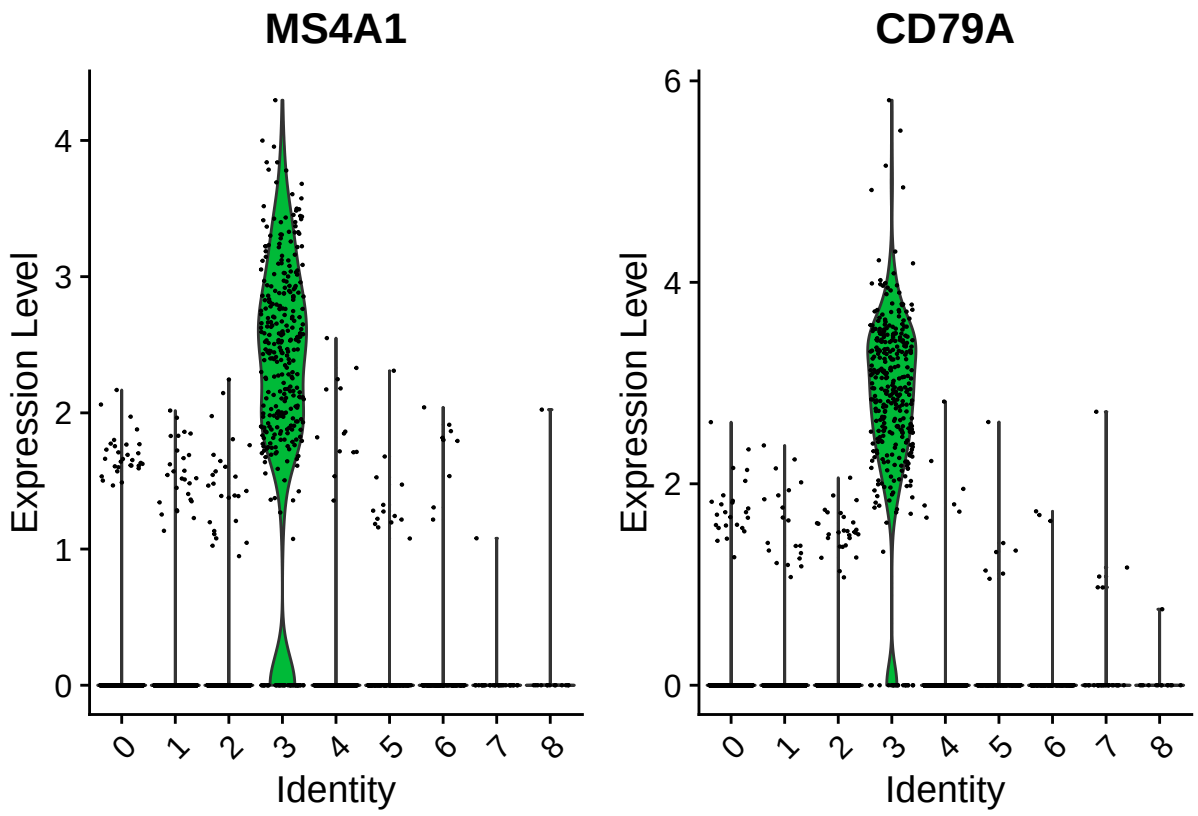
```
pbmc.markers %>%
    group_by(cluster) %>%
    dplyr::filter(avg_log2FC > 1)
```

```
## # A tibble: 7,024 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 5.32e-114       1.19 0.912 0.591 7.29e-110 0       LDHB
##  2 1.31e- 83       2.35 0.439 0.11  1.79e- 79 0       CCR7
##  3 2.61e- 78       1.06 0.85  0.403 3.58e- 74 0       CD3D
##  4 5.89e- 55       1.03 0.731 0.398 8.07e- 51 0       CD3E
##  5 3.91e- 50       2.11 0.338 0.104 5.36e- 46 0       LEF1
```

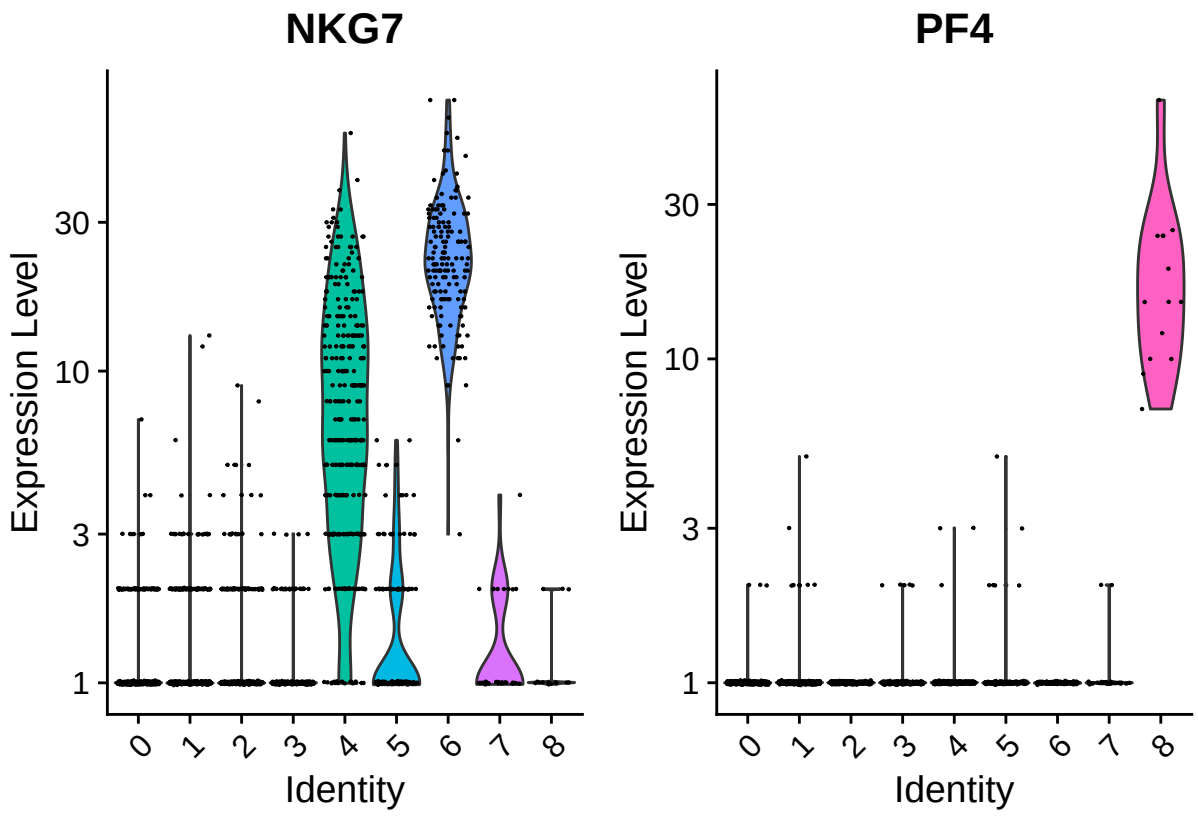
```
##  6 2.53e- 47       1.23 0.624 0.36  3.47e- 43 0       NOSIP
##  7 5.11e- 46       2.04 0.335 0.109 7.01e- 42 0       PRKCQ-AS1
##  8 5.49e- 43       1.51 0.438 0.186 7.52e- 39 0       PIK3IP1
##  9 9.17e- 41       2.73 0.199 0.04  1.26e- 36 0       FHIT
## 10 1.26e- 33       1.32 0.39  0.177 1.72e- 29 0       TCF7
## # i 7,014 more rows
```

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

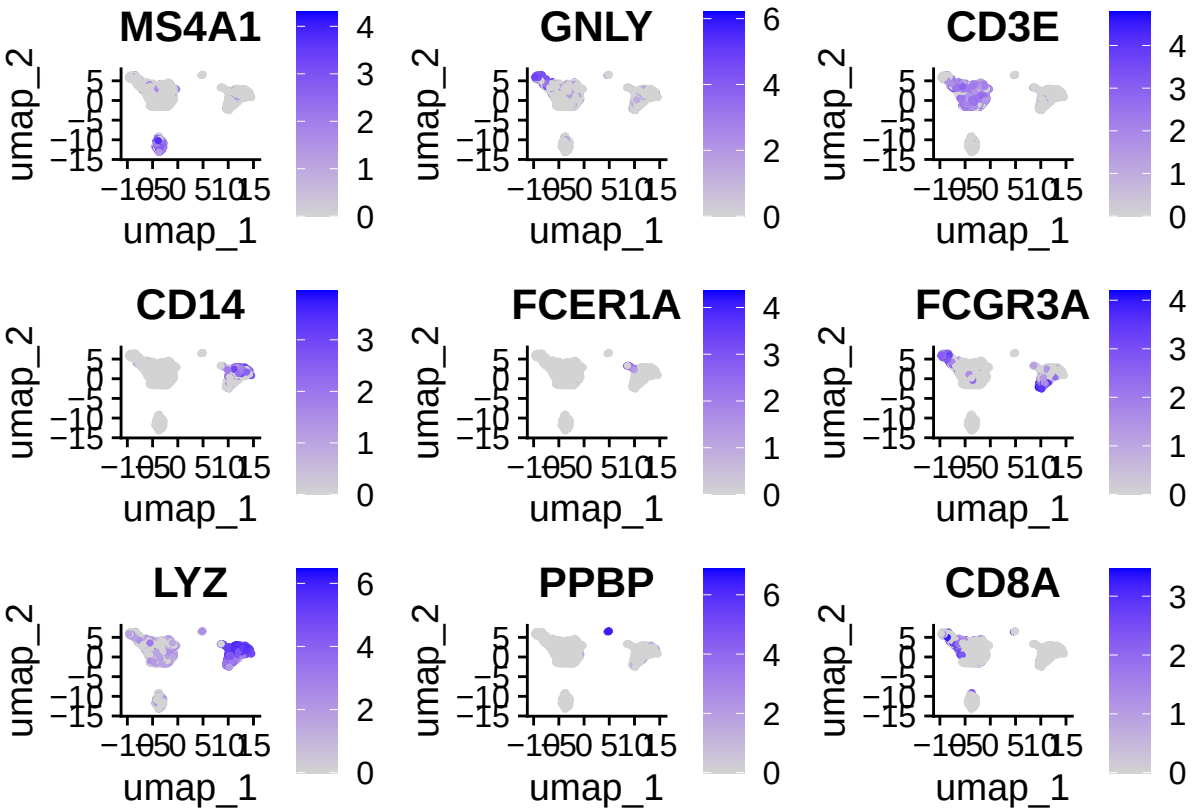
```
VlnPlot(pbmc, features = c("MS4A1", "CD79A"))
```



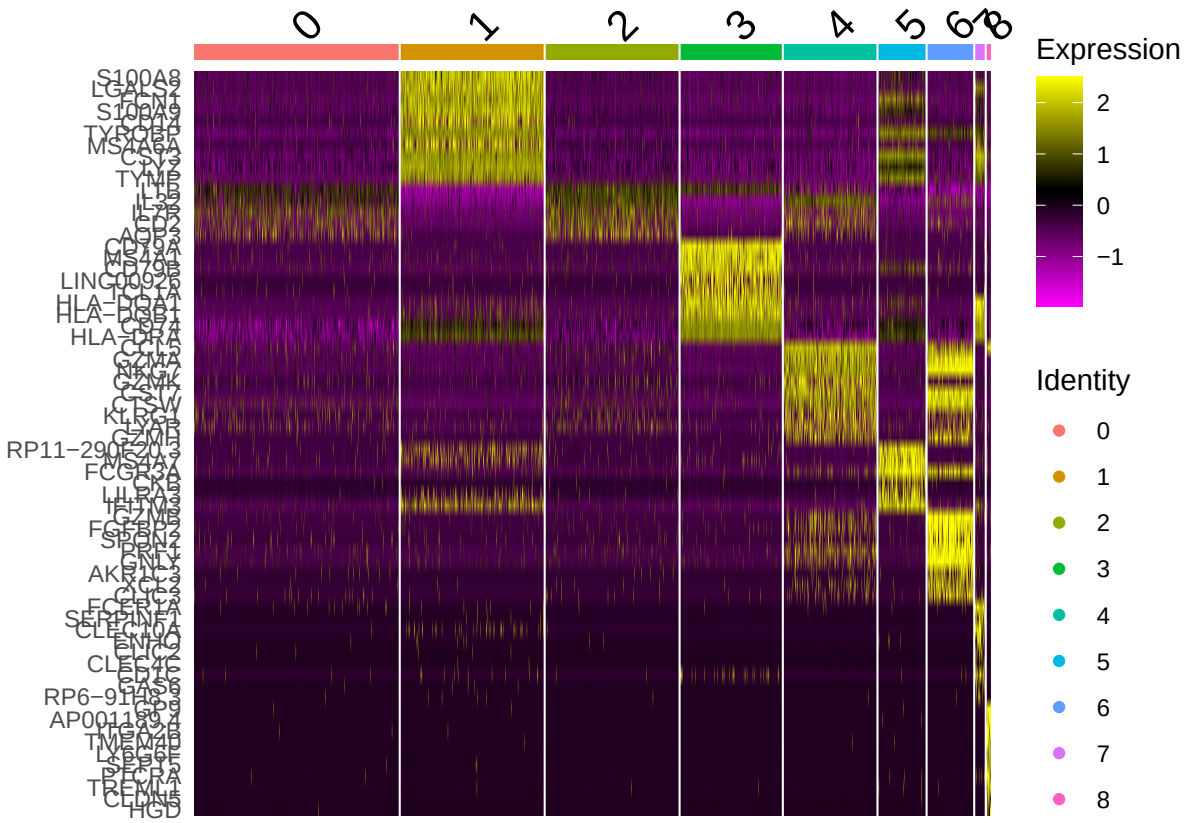

```
# you can plot raw counts as well
VlnPlot(pbmc, features = c("NKG7", "PF4"), slot = "counts", log = TRUE)
```




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

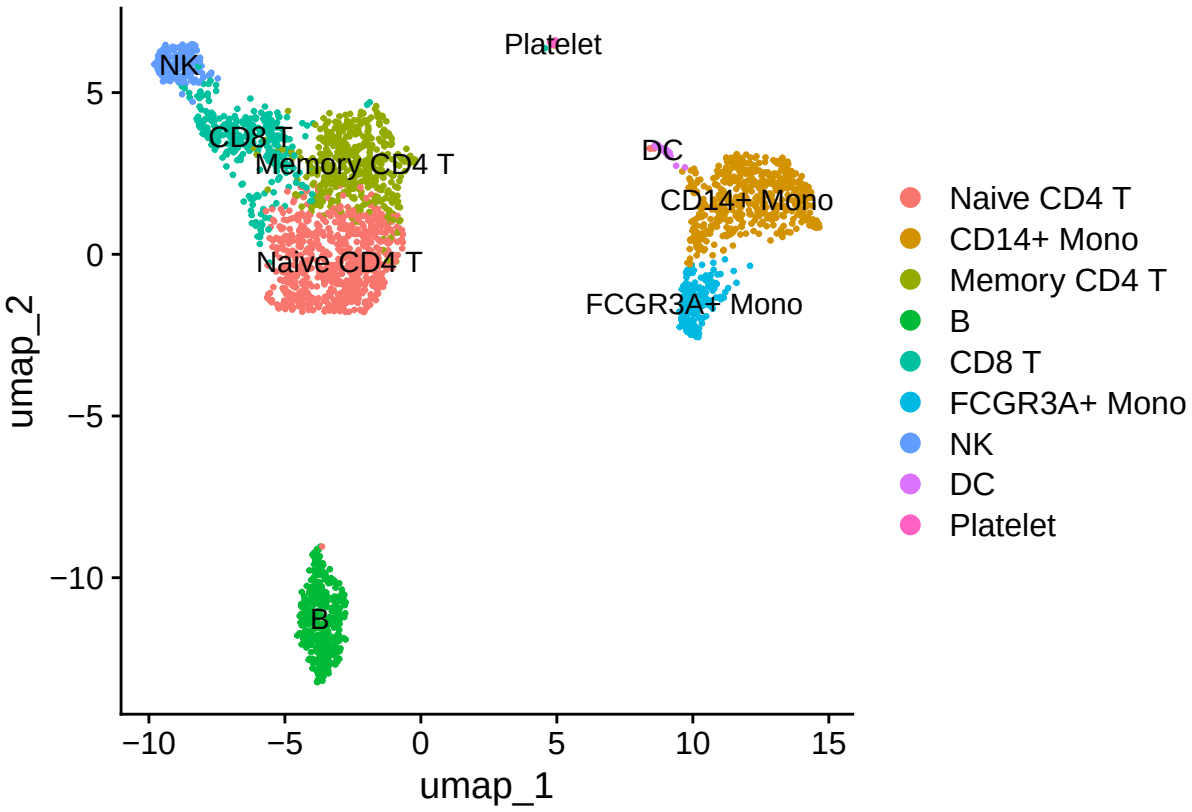



```
pbmc.markers %>%
    group_by(cluster) %>%
    dplyr::filter(avg_log2FC > 1) %>%
    slice_head(n = 10) %>%
    ungroup() -> top10
DoHeatmap(pbmc, features = top10$gene)
```



Assigning cell type identity to clusters

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




```
library(ggplot2)
plot <- DimPlot(pbmc, reduction = "umap", label = TRUE, label.size = 4.5) + xlab("UMAP 1") + ylab("UMAP
    theme(axis.title = element_text(size = 18), legend.text = element_text(size = 18)) + guides(colour =
ggsave(filename = "images/pbmc3k_umap.jpg", height = 7, width = 12, plot = plot, quality = 50)
```

```
saveRDS(pbmc, file = "pbmc3k_final.rds")
```