Single-Cell RNA-Seq Analysis

Pang-Kuo Lo

11/9/2019

Single-Cell RNA-Seq Data Analysis using Seurat

The “Seurat” library package developed by Satija Lab is a useful R-based tool for single-cell genomics analysis. Seurat is used here to analyze the single-cell RNA-seq pbmc6k dataset. The pbmc6k dataset can be downloaded from the resources of the 10X genomics website.

```
# Load the required library packages
library(Seurat)
library(dplyr)
```

The first step is to use the Read10X syntax from Seurat to read the pbmc6k dataset including barcodes.tsv, genes.tsv and matrix.mtx.

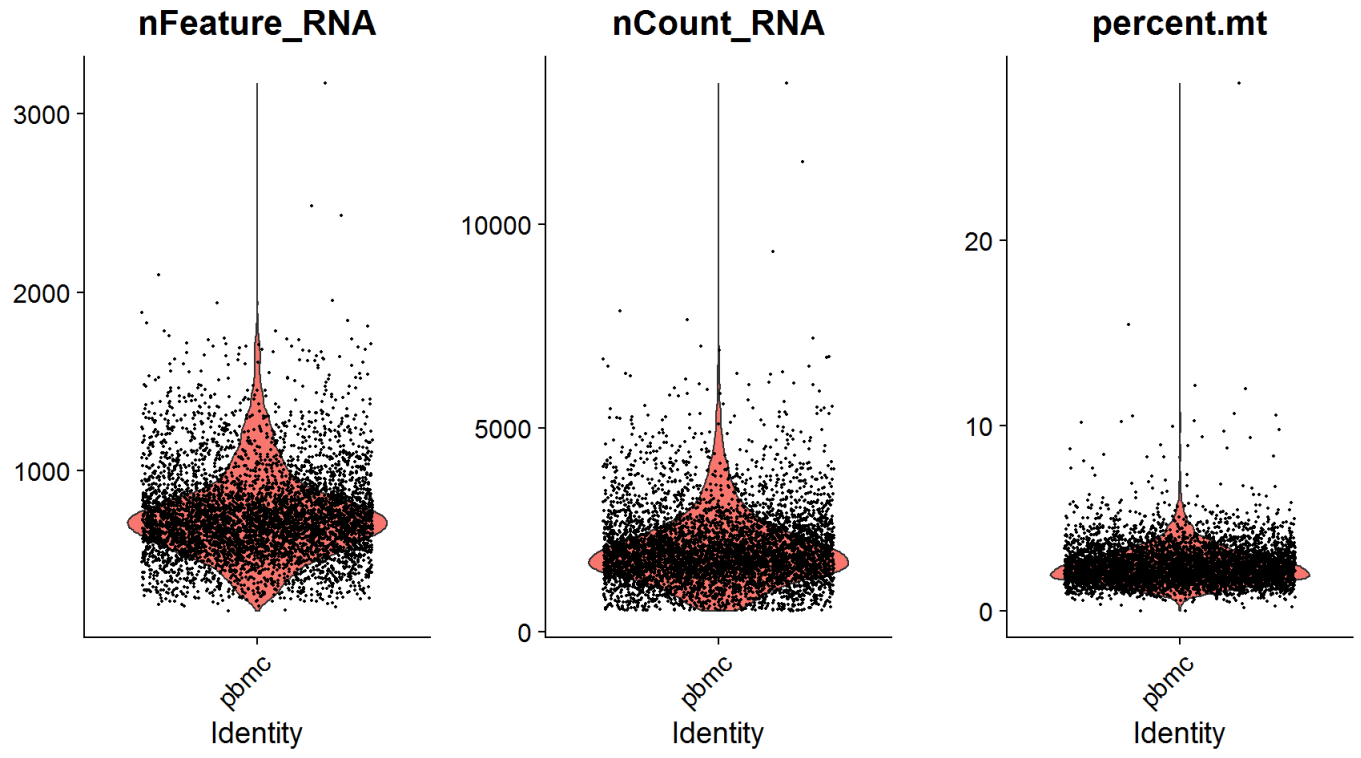
```
pbmc.data <- Read10X(data.dir = "/Users/Pang-Kuo/Desktop/NGS_ML_Analysis/single_cell_RNA-seq_analysis/pbmc6k/hg19", gene.c
olumn = 2)
```

The second step is to create a Seurat object using the CreateSeuratObject syntax.

```
pbmc <- CreateSeuratObject(counts = pbmc.data, project = "pbmc", min.cells = 3, min.features = 200)
```

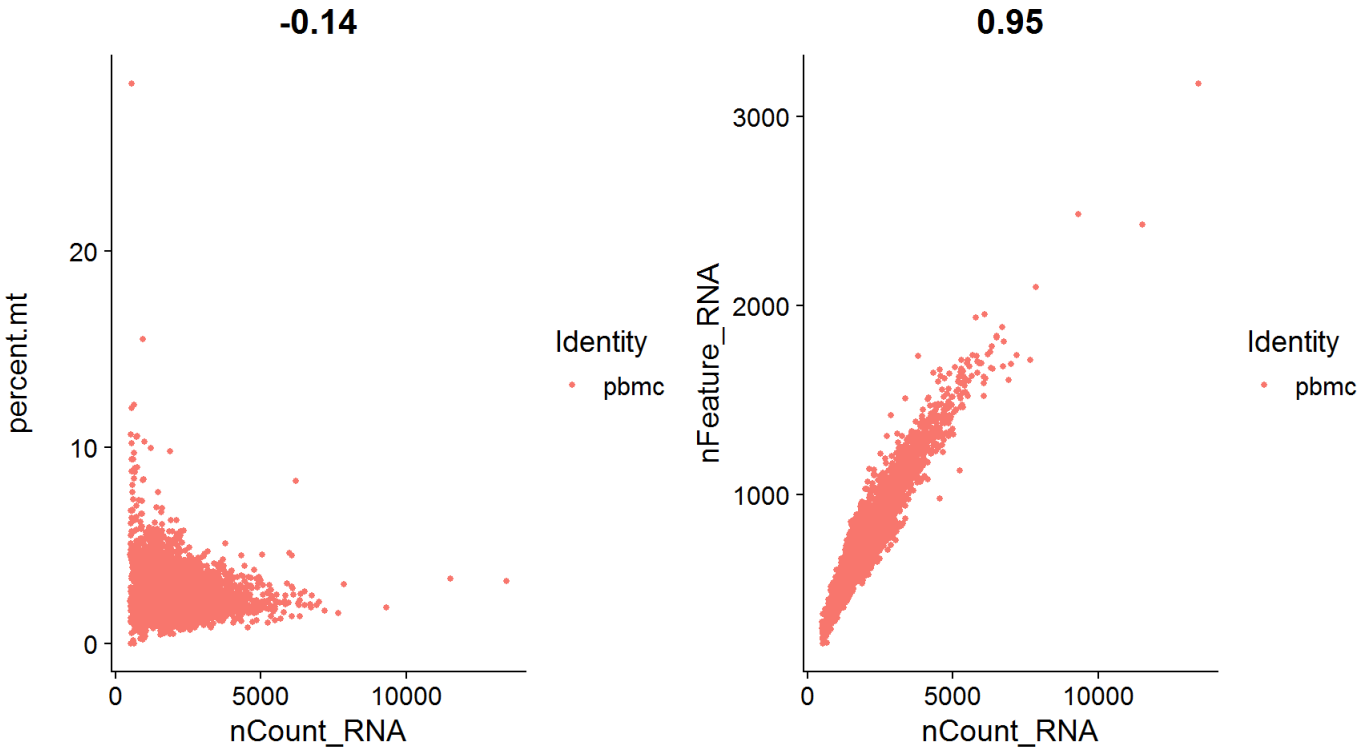
The third step is to calculate the percentage of detected mitochondrial genes.

```
pbmc[["percent.mt"]] <- PercentageFeatureSet(pbmc, pattern = "^MT-")
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3, pt.size = 0.3)
```



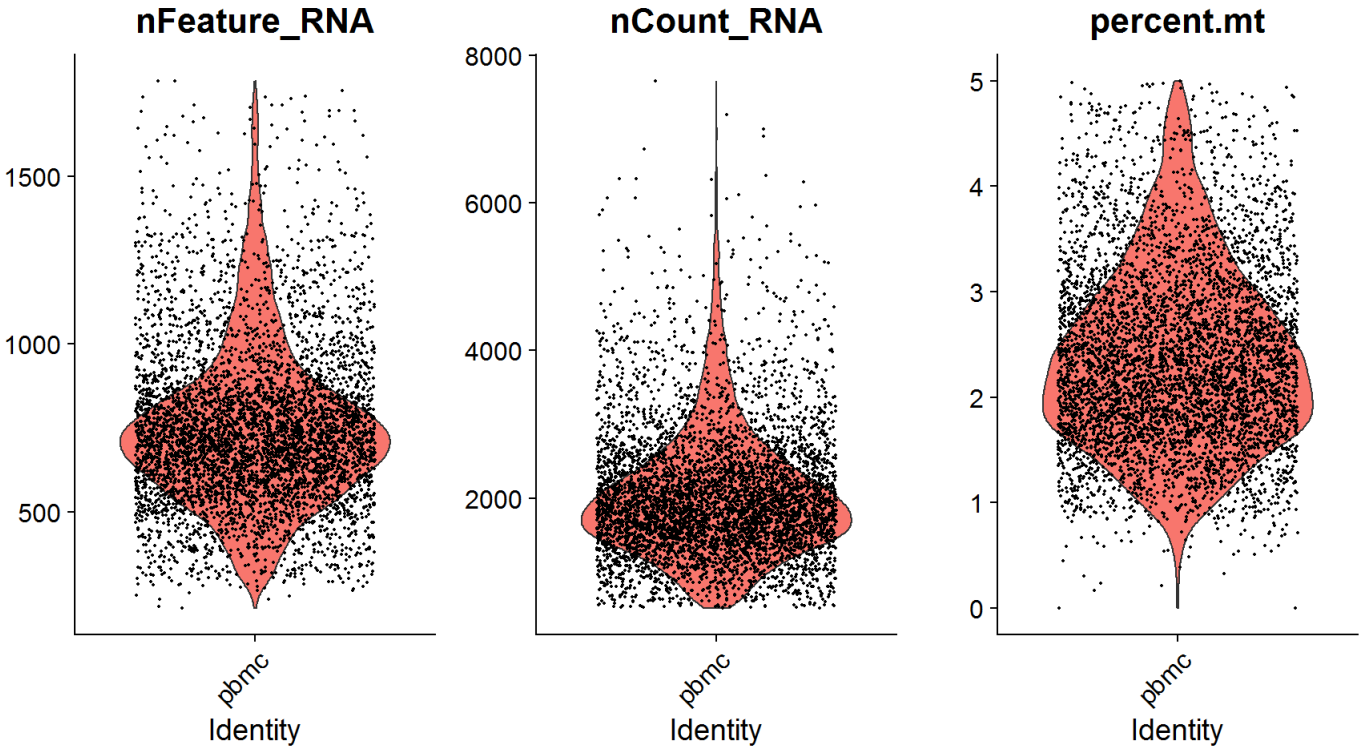
The fourth step is to perform feature scatter plot analysis of the pbmc dataset using the FeatureScatter syntax.

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



The fifth step is to use the subset syntax to remove cells with a too low or too high gene count and cells with a significant mitochondrial gene count (indicative of non-viable cells).

```
pbmc <- subset(pbmc, subset = nFeature_RNA > 200 & nFeature_RNA < 1800 & percent.mt < 5)
VlnPlot(pbmc, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3, pt.size = 0.3)
```

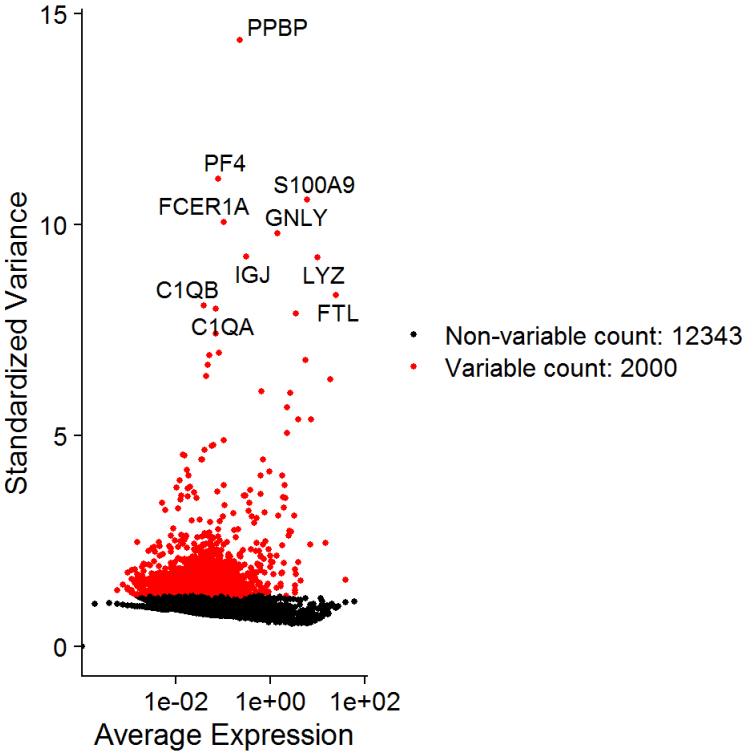


The sixth step is to perform normalization and identify variable expressed genes.

```
pbmc <- NormalizeData(pbmc, normalization.method = "LogNormalize", scale.factor = 10000)
pbmc <- FindVariableFeatures(pbmc, selection.method = "vst", nfeatures = 2000)
top10 <- head(VariableFeatures(pbmc), 10)
top10
```

```
##  [1] "PPBP"   "PF4"    "S100A9" "FCER1A" "GNLY"   "IGJ"    "LYZ"
##  [8] "FTL"    "C1QB"   "C1QA"
```

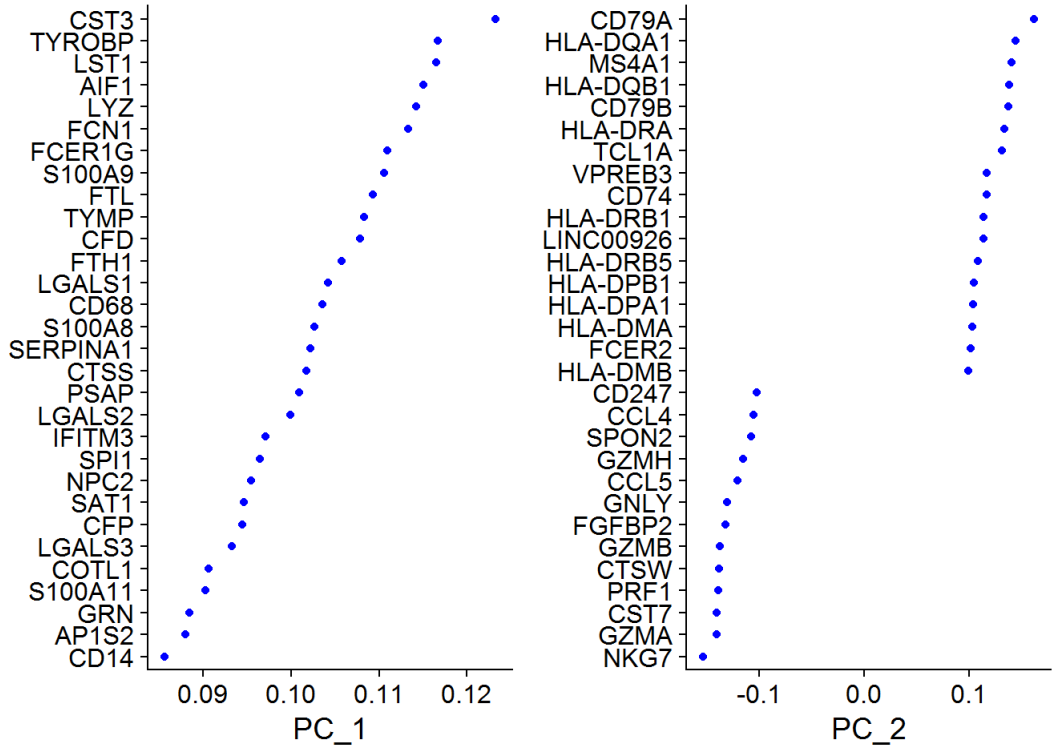
```
# Perform feature plot analysis of variable expressed genes
plot1 <- VariableFeaturePlot(pbmc)
LabelPoints(plot = plot1, points = top10, repel = TRUE, xnudge = 0, ynudge = 0)
```



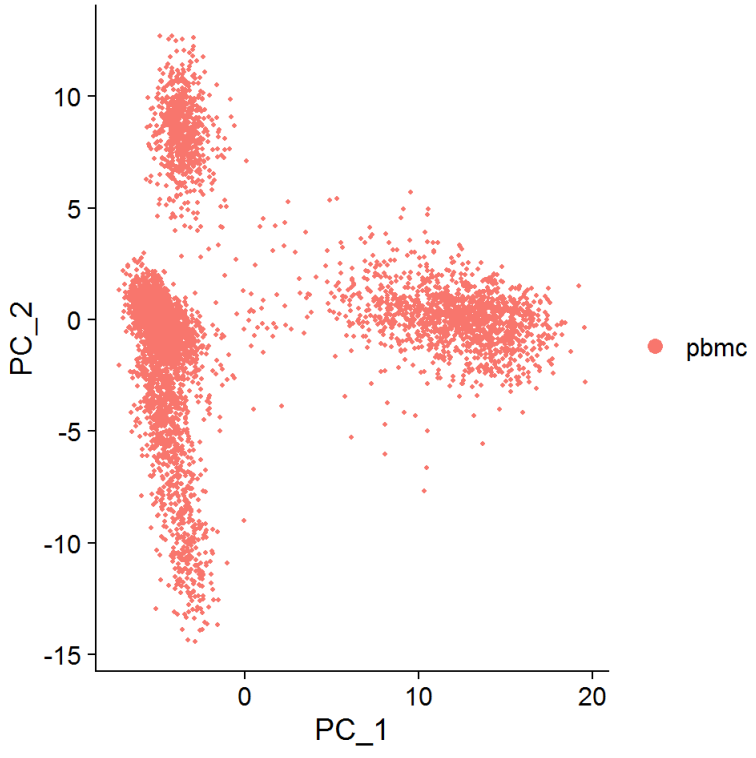
The seventh step is to perform linear scaling and run Principal Component Analysis (PCA) for dimensionality reduction.

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

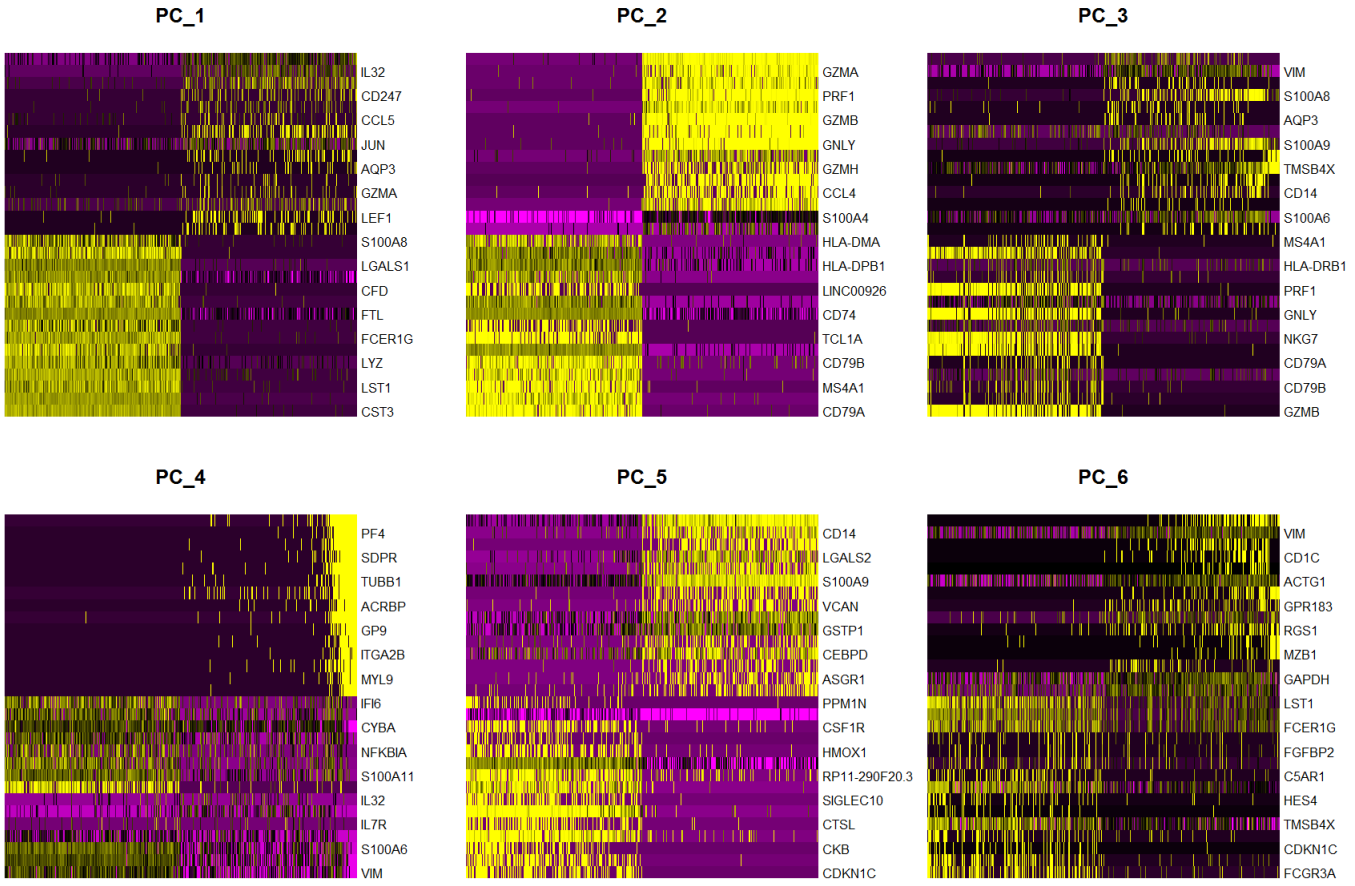
```
# VizDimLoadings analysis
VizDimLoadings(pbmc, dims = 1:2, reduction = "pca")
```



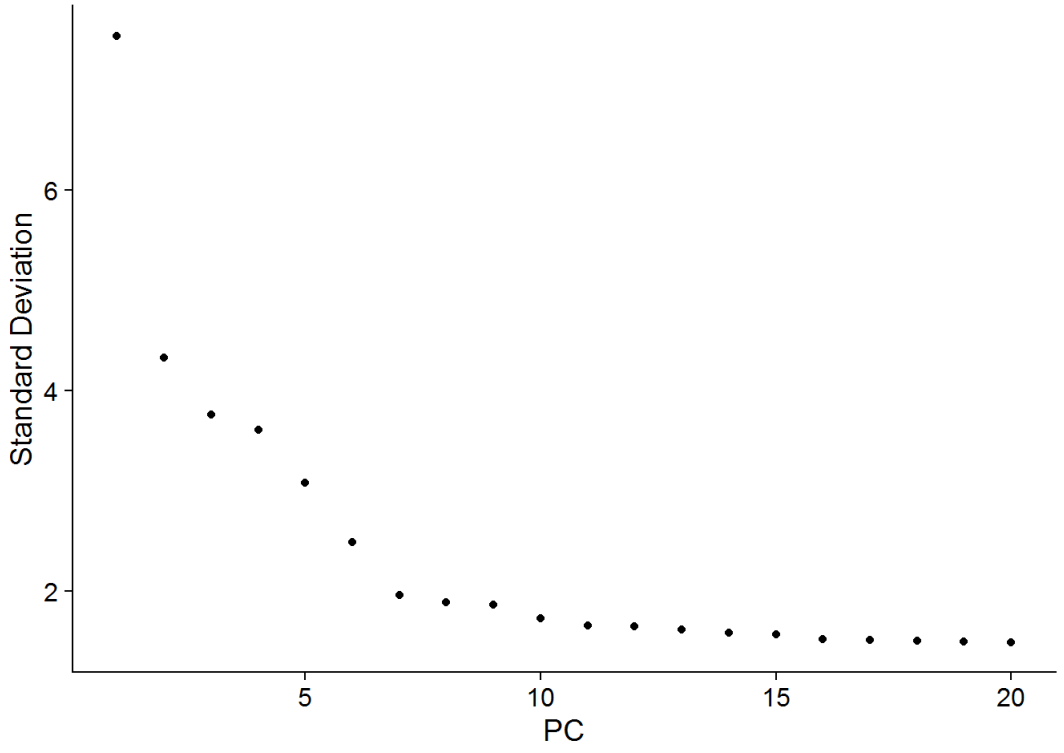

```
# PCA plot analysis
DimPlot(pbmc, reduction = "pca", pt.size = 0.7)
```




```
# DimHeatmap analysis
DimHeatmap(pbmc, dims = 1:6, cells = 500, balanced = TRUE)
```




```
# ElbowPlot analysis
ElbowPlot(pbmc)
```



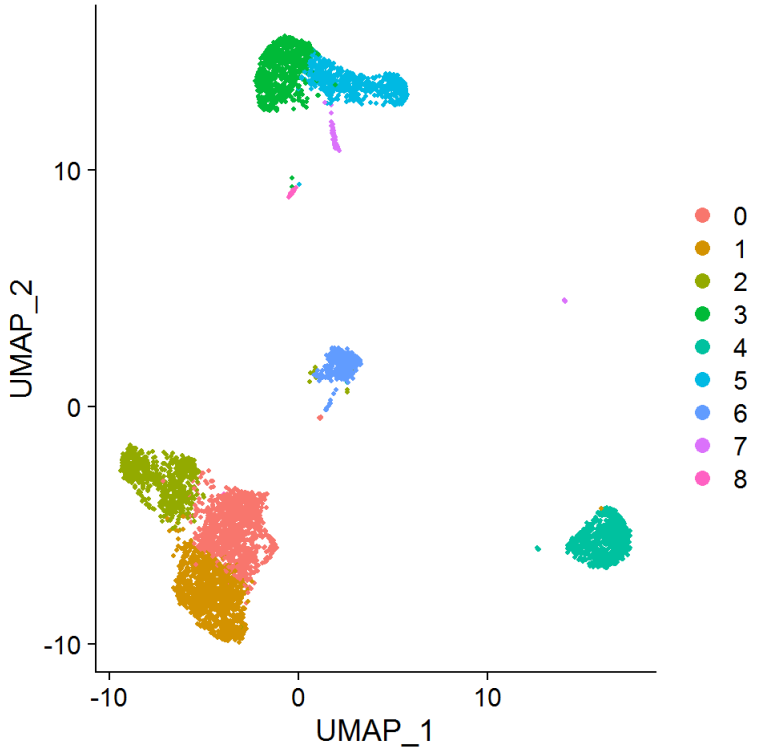
The eighth step is to perform neighboring and clustering analysis.

```
pbmc <- FindNeighbors(pbmc, dims = 1:15)
pbmc <- FindClusters(pbmc, resolution = 0.5)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 5283
## Number of edges: 200359
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8865
## Number of communities: 9
## Elapsed time: 0 seconds
```

The ninth step is to perform Uniform Manifold Approximation and Projection (UMAP) analysis for dimensionality reduction.

```
pbmc <- RunUMAP(pbmc, dims = 1:15)
DimPlot(pbmc, reduction = "umap", pt.size = 0.7)
```



The tenth step is to identify cell-type-specific gene markers using the FindMarkers syntax.

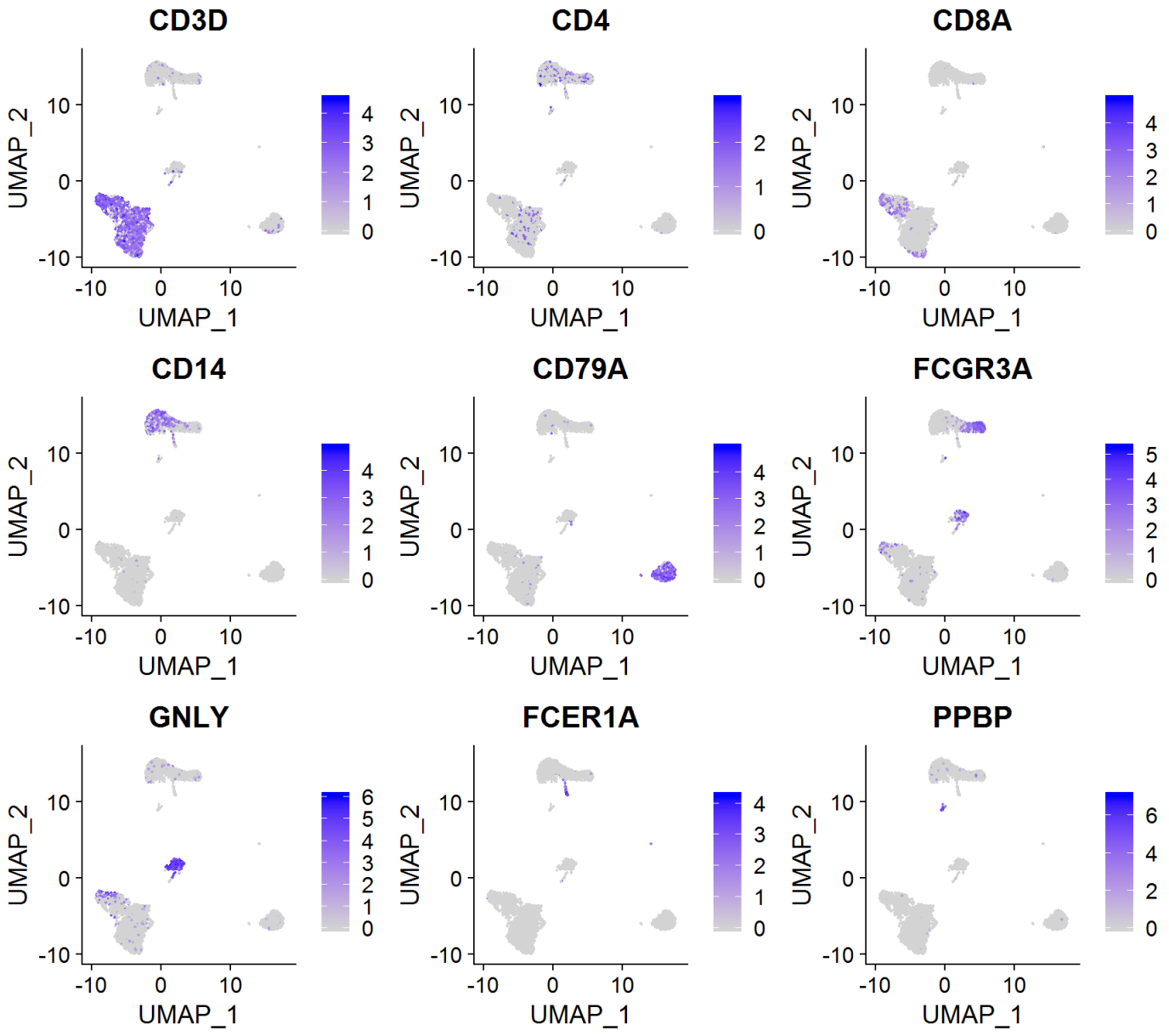
```
cluster_markers <- vector("list", 9)
names(cluster_markers) <- c(paste0("Cluster_",c(seq(1:9)-1)))
cluster_num <- c(seq(1:9)-1)
for (i in 1:9) {
  cluster_markers[[i]] <- FindMarkers(pbmc, ident.1 = cluster_num[i], min.pct = 0.25)
}
for (i in 1:9) {
  print(names(cluster_markers[i])); print(head(cluster_markers[[i]],8))
}
```

```
## [1] "Cluster_0"
##                p_val  avg_logFC pct.1 pct.2     p_val_adj
## LTB     1.945127e-202  0.8803064 0.957 0.577 2.789895e-198
## IL32    5.515849e-200  0.8819893 0.923 0.409 7.911382e-196
## IL7R    1.073432e-195  0.9063111 0.742 0.260 1.539624e-191
## LDHB    1.389207e-189  0.7727523 0.941 0.540 1.992540e-185
## AQP3    3.685784e-170  0.9329918 0.388 0.066 5.286520e-166
## CD3D    1.142594e-160  0.7163877 0.866 0.379 1.638822e-156
## HLA-DRA 1.186000e-146 -2.6645751 0.194 0.571 1.701079e-142
## CD74    2.260735e-138 -1.9832966 0.578 0.765 3.242573e-134
## [1] "Cluster_1"
##               p_val  avg_logFC pct.1 pct.2     p_val_adj
## RPS12 1.200383e-226  0.4939444 0.999 0.990 1.721709e-222
## RPL32 5.986060e-226  0.4337813 0.999 0.996 8.585806e-222
## RPS14 1.415911e-218  0.4401715 0.999 0.996 2.030841e-214
## CYBA  5.025829e-211 -1.1658016 0.552 0.891 7.208547e-207
## RPS27 7.127874e-211  0.4930438 0.996 0.990 1.022351e-206
## RPL31 3.786072e-204  0.5373341 0.994 0.971 5.430363e-200
## RPS6  4.106499e-201  0.4407961 1.000 0.996 5.889951e-197
## RPL13 1.540804e-197  0.3882065 1.000 0.996 2.209975e-193
## [1] "Cluster_2"
##              p_val avg_logFC pct.1 pct.2     p_val_adj
## CCL5  0.000000e+00  2.448939 0.971 0.153  0.000000e+00
## GZMK  0.000000e+00  2.124211 0.505 0.034  0.000000e+00
## CD8A  0.000000e+00  1.608061 0.489 0.040  0.000000e+00
## NKG7  0.000000e+00  1.582866 0.883 0.152  0.000000e+00
## CST7  0.000000e+00  1.504414 0.717 0.106  0.000000e+00
## GZMA  0.000000e+00  1.473542 0.702 0.096  0.000000e+00
## CTSW 2.226275e-262  1.222644 0.767 0.179 3.193146e-258
## GZMH 8.584534e-213  1.966541 0.422 0.052 1.231280e-208
## [1] "Cluster_3"
##         p_val avg_logFC pct.1 pct.2 p_val_adj
## S100A8      0  3.505254 0.997 0.150         0
## S100A9      0  3.295515 0.990 0.213         0
## LYZ         0  2.495039 1.000 0.393         0
## FCN1        0  2.080280 0.966 0.169         0
## CD14        0  1.994196 0.739 0.068         0
## LGALS2      0  1.982884 0.838 0.109         0
## TYROBP      0  1.778064 0.996 0.268         0
## S100A12     0  1.718246 0.440 0.010         0
## [1] "Cluster_4"
##          p_val avg_logFC pct.1 pct.2 p_val_adj
## CD79A        0  3.030731 0.929 0.026         0
## TCL1A        0  2.592535 0.574 0.013         0
## CD79B        0  2.519988 0.915 0.116         0
## MS4A1        0  2.372690 0.751 0.027         0
## HLA-DQA1     0  2.167423 0.885 0.110         0
## CD74         0  2.092161 1.000 0.685         0
## HLA-DQB1     0  2.077491 0.862 0.143         0
## HLA-DRA      0  1.907466 0.999 0.416         0
## [1] "Cluster_5"
##          p_val avg_logFC pct.1 pct.2 p_val_adj
## IFITM3       0  2.119298 0.901 0.115         0
## LST1         0  2.045152 0.997 0.236         0
## AIF1         0  1.998175 0.994 0.264         0
## FCER1G       0  1.788212 0.995 0.255         0
## SERPINA1     0  1.760088 0.902 0.122         0
## CST3         0  1.706757 0.998 0.247         0
## COTL1        0  1.598975 0.997 0.521         0
## MS4A7        0  1.587085 0.618 0.046         0
## [1] "Cluster_6"
##                p_val avg_logFC pct.1 pct.2     p_val_adj
## GNLY    0.000000e+00  3.746672 0.964 0.082  0.000000e+00
## GZMB    0.000000e+00  3.117224 0.921 0.066  0.000000e+00
## PRF1    0.000000e+00  2.714076 0.921 0.093  0.000000e+00
## FGFBP2  0.000000e+00  2.706371 0.826 0.057  0.000000e+00
## SPON2   0.000000e+00  2.298524 0.698 0.030  0.000000e+00
```

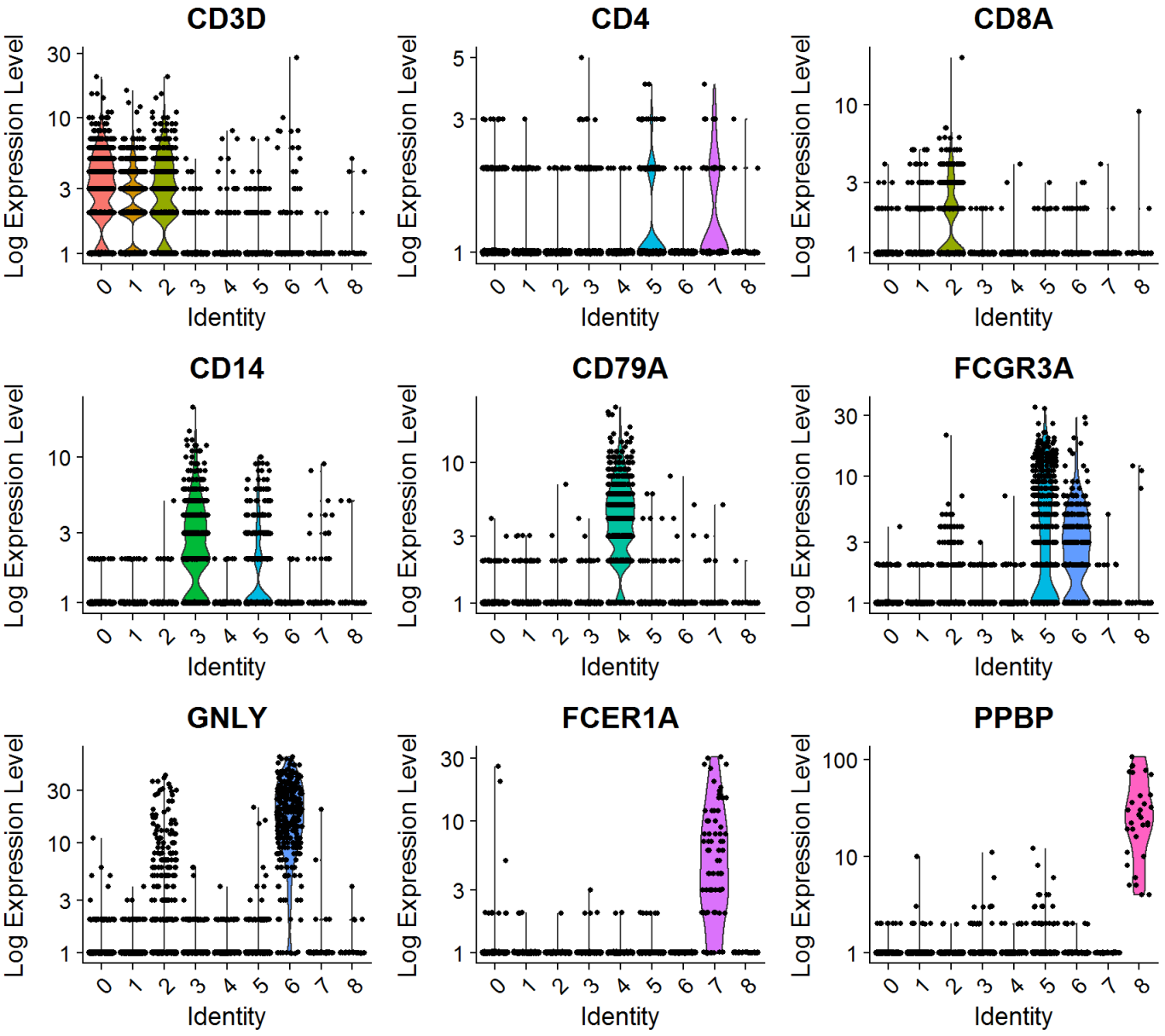
```
## CLIC3   0.000000e+00  1.840776 0.548 0.026  0.000000e+00
## AKR1C3  0.000000e+00  1.680990 0.456 0.012  0.000000e+00
## NKG7   4.583131e-298  2.710549 0.990 0.203 6.573585e-294
## [1] "Cluster_7"
##                 p_val avg_logFC pct.1 pct.2     p_val_adj
## FCER1A    0.000000e+00 2.6737184 0.857 0.005  0.000000e+00
## CLEC10A   0.000000e+00 1.8512462 0.675 0.012  0.000000e+00
## ENHO      0.000000e+00 0.9565024 0.442 0.002  0.000000e+00
## GSN      4.155209e-143 0.9580938 0.649 0.035 5.959817e-139
## SERPINF1 4.370337e-142 0.9737206 0.260 0.004 6.268374e-138
## CD1C     1.727550e-141 1.4478949 0.494 0.021 2.477824e-137
## PLD4     1.611417e-115 1.3448075 0.545 0.032 2.311255e-111
## CACNA2D3 1.942430e-113 0.7135865 0.377 0.014 2.786028e-109
## [1] "Cluster_8"
##        p_val avg_logFC pct.1 pct.2 p_val_adj
## PPBP       0  5.634794 1.000 0.016         0
## PF4        0  4.722368 1.000 0.004         0
## SDPR       0  4.142094 0.935 0.006         0
## GNG11      0  4.127555 0.968 0.006         0
## TUBB1      0  3.872728 0.968 0.007         0
## CLU        0  3.836917 1.000 0.009         0
## ACRBP      0  3.307028 0.871 0.005         0
## TREML1     0  3.261734 0.839 0.002         0
```

The eleventh step is to visualize expression of cell-type-specific gene markers in UMAP plots and VlnPlots.

```
# Visualization of cell-type-specific gene markers in UMAP plots
cell_type_markers <- c("CD3D", "CD4", "CD8A", "CD14", "CD79A", "FCGR3A", "GNLY", "FCER1A", "PPBP")
FeaturePlot(pbmc, features = cell_type_markers)
```

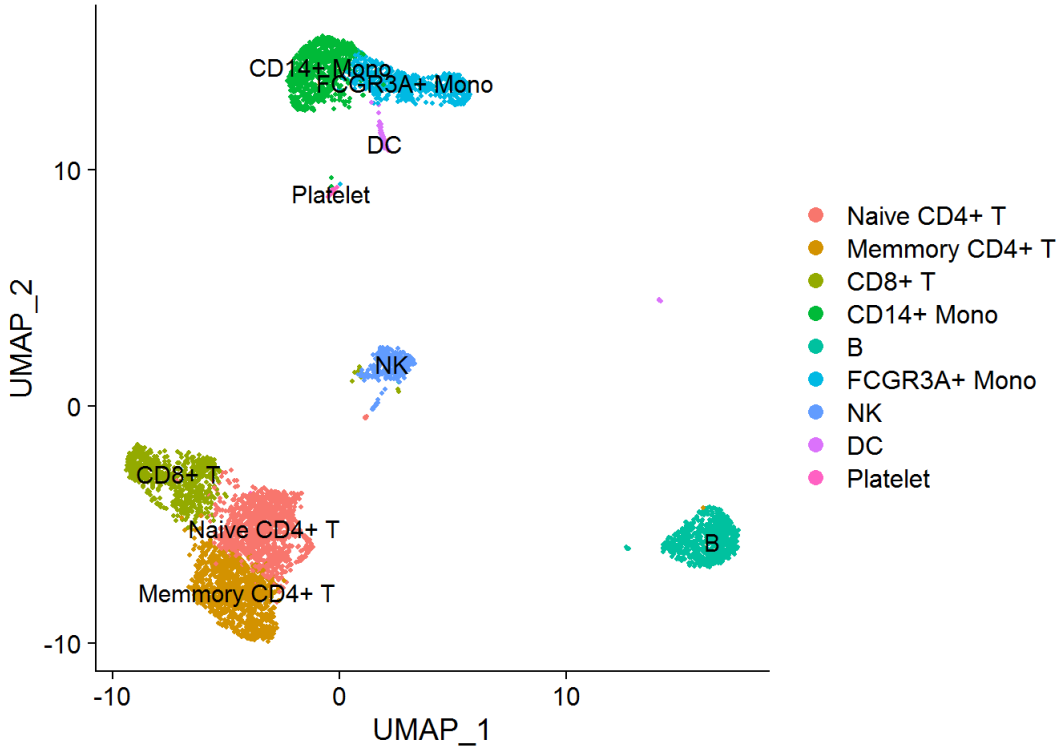



```
# Visualization of cell-type-specific gene markers in VlnPlots
VlnPlot(pbmc, features = cell_type_markers, slot = "counts", log = TRUE)
```



The final step is to annotate identified cell clusters with cell type names in the UMAP plot.

```
pbmc <- RenameIdents(pbmc, `0` = "Naive CD4+ T", `1` = "Memmory CD4+ T", `2` = "CD8+ T", `3` = "CD14+ Mono", `4` = "B", `5
` = "FCGR3A+ Mono", `6` = "NK", `7` = "DC", `8` = "Platelet")
DimPlot(pbmc, reduction = "umap", label = TRUE, pt.size = 0.7)
```

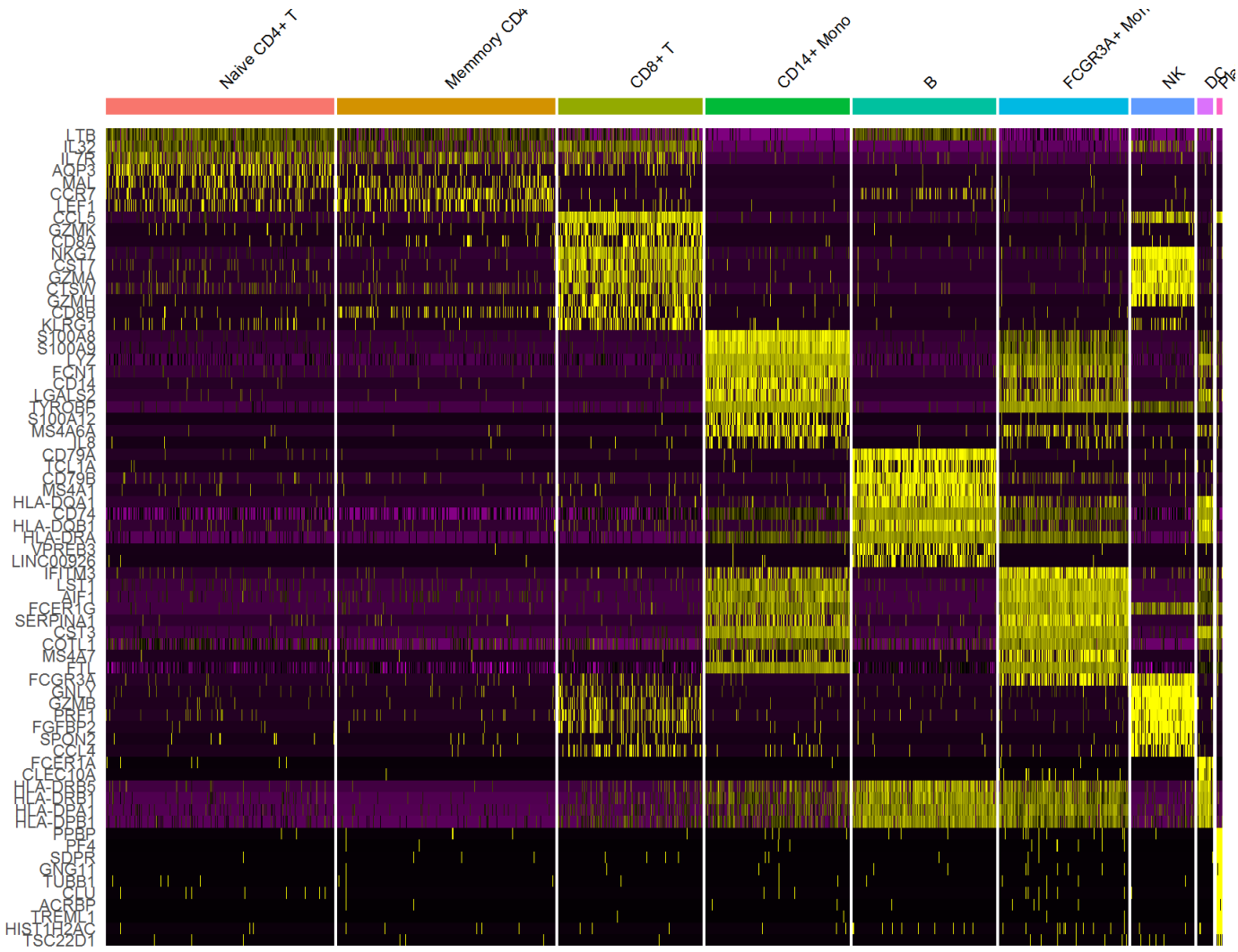


Heatmap analysis is performed to display differentially expressed genes in every cell type.

```
pbmc.markers <- FindAllMarkers(pbmc, only.pos = TRUE, min.pct = 0.25, logfc.threshold = 0.25)
pbmc.markers %>% group_by(cluster) %>% top_n(n = 4, wt = avg_logFC)
```

```
## # A tibble: 36 x 7
## # Groups:   cluster [9]
##        p_val avg_logFC pct.1 pct.2 p_val_adj cluster        gene
##        <dbl>     <dbl> <dbl> <dbl>     <dbl> <fct>          <chr>
##  1 1.95e-202     0.880 0.957 0.577 2.79e-198 Naive CD4+ T   LTB
##  2 5.52e-200     0.882 0.923 0.409 7.91e-196 Naive CD4+ T   IL32
##  3 1.07e-195     0.906 0.742 0.26  1.54e-191 Naive CD4+ T   IL7R
##  4 3.69e-170     0.933 0.388 0.066 5.29e-166 Naive CD4+ T   AQP3
##  5 1.89e-138     0.710 0.876 0.561 2.71e-134 Memmory CD4+ T LDHB
##  6 1.60e-118     0.920 0.384 0.102 2.30e-114 Memmory CD4+ T CCR7
##  7 3.42e- 88     0.788 0.306 0.085 4.91e- 84 Memmory CD4+ T LEF1
##  8 2.44e- 69     0.711 0.582 0.337 3.50e- 65 Memmory CD4+ T NOSIP
##  9 0.            2.45  0.971 0.153 0.        CD8+ T         CCL5
## 10 0.            2.12  0.505 0.034 0.        CD8+ T         GZMK
## # ... with 26 more rows
```

```
top10 <- pbmc.markers %>% group_by(cluster) %>% top_n(n = 10, wt = avg_logFC)
DoHeatmap(pbmc, features = top10$gene, size = 3) + NoLegend()
```



Here is the output of sessionInfo() on the system on which this document was compiled:

```
sessionInfo()
```

```
## R version 3.6.1 (2019-07-05)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 7 x64 (build 7601) Service Pack 1
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] dplyr_0.8.3  Seurat_3.1.1
##
## loaded via a namespace (and not attached):
##   [1] tsne_0.1-3          nlme_3.1-140        bitops_1.0-6
##   [4] RcppAnnoy_0.0.13    RColorBrewer_1.1-2  httr_1.4.1
##   [7] sctransform_0.2.0   tools_3.6.1         backports_1.1.5
##  [10] utf8_1.1.4          R6_2.4.0            irlba_2.3.3
##  [13] KernSmooth_2.23-15  uwot_0.1.4          lazyeval_0.2.2
##  [16] colorspace_1.4-1    withr_2.1.2         npsurv_0.4-0
##  [19] gridExtra_2.3       tidyselect_0.2.5    compiler_3.6.1
##  [22] cli_1.1.0           plotly_4.9.1        labeling_0.3
##  [25] caTools_1.17.1.2    scales_1.0.0        lmtest_0.9-37
##  [28] ggridges_0.5.1      pbapply_1.4-2       stringr_1.4.0
##  [31] digest_0.6.22       rmarkdown_1.16      R.utils_2.9.0
##  [34] pkgconfig_2.0.3     htmltools_0.4.0     bibtex_0.4.2
##  [37] htmlwidgets_1.5.1   rlang_0.4.1         zoo_1.8-6
##  [40] jsonlite_1.6        ica_1.0-2           gtools_3.8.1
##  [43] R.oo_1.23.0         magrittr_1.5        Matrix_1.2-17
##  [46] fansi_0.4.0         Rcpp_1.0.3          munsell_0.5.0
##  [49] ape_5.3             reticulate_1.13     lifecycle_0.1.0
##  [52] R.methodsS3_1.7.1   stringi_1.4.3       yaml_2.2.0
##  [55] gbRd_0.4-11         MASS_7.3-51.4       gplots_3.0.1.1
##  [58] Rtsne_0.15          plyr_1.8.4          grid_3.6.1
##  [61] parallel_3.6.1      gdata_2.18.0        listenv_0.7.0
##  [64] ggrepel_0.8.1       crayon_1.3.4        lattice_0.20-38
##  [67] cowplot_1.0.0       splines_3.6.1       SDMTools_1.1-221.1
##  [70] zeallot_0.1.0       knitr_1.25          pillar_1.4.2
##  [73] igraph_1.2.4.1      future.apply_1.3.0  reshape2_1.4.3
##  [76] codetools_0.2-16    leiden_0.3.1        glue_1.3.1
##  [79] evaluate_0.14       lsei_1.2-0          metap_1.1
##  [82] RcppParallel_4.4.4  data.table_1.12.6   vctrs_0.2.0
##  [85] png_0.1-7           Rdpack_0.11-0       gtable_0.3.0
##  [88] RANN_2.6.1          purrr_0.3.3         tidyr_1.0.0
##  [91] future_1.15.0       assertthat_0.2.1    ggplot2_3.2.1
##  [94] xfun_0.10           rsvd_1.0.2          RSpectra_0.15-0
##  [97] survival_2.44-1.1   viridisLite_0.3.0   tibble_2.1.3
## [100] cluster_2.1.0       globals_0.12.4      fitdistrplus_1.0-14
## [103] ROCR_1.0-7
```