SoupX

title: “SoupX for CM1_2” output: html_document —

```
library(SoupX)
library(Seurat)
library(ggplot2)
library(DropletUtils)
```

```
## Loading required package: SingleCellExperiment
```

```
## Loading required package: SummarizedExperiment
```

```
## Loading required package: GenomicRanges
```

```
## Loading required package: stats4
```

```
## Loading required package: BiocGenerics
```

```
## Loading required package: parallel
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which, which.max, which.min
```

```
## Loading required package: S4Vectors
```

```
##
## Attaching package: 'S4Vectors'
```

```
## The following object is masked from 'package:base':
##
##     expand.grid
```

```
## Loading required package: IRanges
```

```
##
## Attaching package: 'IRanges'
```

```
## The following object is masked from 'package:grDevices':
##
##     windows
```

```
## Loading required package: GenomeInfoDb
```

```
## Loading required package: Biobase
```

```
## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

```
## Loading required package: DelayedArray
```

```
## Loading required package: matrixStats
```

```
##
## Attaching package: 'matrixStats'
```

```
## The following objects are masked from 'package:Biobase':
##
##     anyMissing, rowMedians
```

```
##
## Attaching package: 'DelayedArray'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colMaxs, colMins, colRanges, rowMaxs, rowMins, rowRanges
```

```
## The following objects are masked from 'package:base':
##
##     aperm, apply, rowsum
```

```
##
## Attaching package: 'SummarizedExperiment'
```

```
## The following object is masked from 'package:Seurat':
##
##     Assays
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:matrixStats':
##
##     count
```

```
## The following object is masked from 'package:Biobase':
##
##     combine
```

```
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
```

```
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
```

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

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

Load 10X Data to SoupX and Seurat

```
DataDir = c('G://covid19scRNAseq/CM1_2/outs', 'G://covid19scRNAseq/CM1_2/outs/filtered_feature_b
c_matrix')
sc = load10X(DataDir[1])
```

```
## Loading raw count data
```

```
## Loading cell-only count data
```

```
## Loading extra analysis data where available
```

```
seu <- Read10X(DataDir[2])
```

running seurat, set up cluster, adjust count matrix

```
seu <- CreateSeuratObject(counts = seu, project = "cm1_2")
seu <- SCTransform(object = seu, verbose = T)
```

```
## Calculating cell attributes from input UMI matrix: log_umi
```

```
## Variance stabilizing transformation of count matrix of size 15704 by 7523
```

```
## Model formula is y ~ log_umi
```

```
## Get Negative Binomial regression parameters per gene
```

```
## Using 2000 genes, 7523 cells
```

```
##
  |
  |                                                                      |   0%
  |
  |==                                                                    |   3%
  |
  |====                                                                  |   6%
  |
  |=======                                                               |   9%
  |
  |=========                                                             |  12%
  |
  |===========                                                           |  16%
  |
  |==============                                                        |  19%
  |
  |================                                                      |  22%
  |
  |==================                                                    |  25%
  |
  |====================                                                  |  28%
  |
  |======================                                                |  31%
  |
  |=========================                                             |  34%
  |
  |===========================                                           |  38%
  |
  |============================                                          |  41%
  |
  |==============================                                        |  44%
  |
  |================================                                      |  47%
  |
  |==================================                                    |  50%
  |
  |====================================                                  |  53%
  |
  |======================================                                |  56%
  |
  |========================================                              |  59%
  |
  |==========================================                            |  62%
  |
  |============================================                          |  66%
  |
  |==============================================                        |  69%
  |
  |================================================                      |  72%
  |
  |==================================================                    |  75%
  |
  |====================================================                  |  78%
```

```
  |
  |=============================================================          |  81%
  |
  |==============================================================         |  84%
  |
  |================================================================       |  88%
  |
  |===================================================================    |  91%
  |
  |======================================================================= |  94%
  |
  |=========================================================================|  97%
  |
  |=========================================================================| 100%
```

```
## Computing corrected count matrix for 15704 genes
```

```
##
  |
  |                                                                      |   0%
  |
  |==                                                                    |   3%
  |
  |====                                                                  |   6%
  |
  |=======                                                               |   9%
  |
  |=========                                                             |  12%
  |
  |===========                                                           |  16%
  |
  |==============                                                        |  19%
  |
  |================                                                      |  22%
  |
  |==================                                                    |  25%
  |
  |====================                                                  |  28%
  |
  |======================                                                |  31%
  |
  |========================                                              |  34%
  |
  |==========================                                            |  38%
  |
  |============================                                          |  41%
  |
  |==============================                                        |  44%
  |
  |================================                                      |  47%
  |
  |==================================                                    |  50%
  |
  |====================================                                  |  53%
  |
  |======================================                                |  56%
  |
  |========================================                              |  59%
  |
  |==========================================                            |  62%
  |
  |============================================                          |  66%
  |
  |==============================================                        |  69%
  |
  |================================================                      |  72%
  |
  |==================================================                    |  75%
  |
  |====================================================                  |  78%
```

```
  |
  |=============================================================          |  81%
  |
  |==============================================================         |  84%
  |
  |================================================================       |  88%
  |
  |===================================================================    |  91%
  |
  |======================================================================= |  94%
  |
  |=========================================================================  |  97%
  |
  |============================================================================| 100%
```

```
## Calculating gene attributes
```

```
## Wall clock passed: Time difference of 2.298225 mins
```

```
## Determine variable features
```

```
## Set 3000 variable features
```

```
## Place corrected count matrix in counts slot
```

```
## Centering data matrix
```

```
## Set default assay to SCT
```

```
seu <- RunPCA(object = seu, verbose = T)
```

```
## PC_ 1
## Positive:  LYZ, S100A8, AC020656.1, CST3, S100A9, FCN1, S100A6, IFI27, IFITM3, S100A12
##     TYROBP, FTL, MNDA, FTH1, AIF1, S100A11, S100A4, S100A10, CEBPD, FCER1G
##     CTSS, NEAT1, VCAN, CTSD, CFD, CD14, CSTA, LGALS3, CXCL8, NCF1
## Negative:  GNLY, CCL5, NKG7, IL32, IL7R, CTSW, CD3D, RPL10, TRBC2, CD3G
##     RPS12, TRBC1, TRAC, CST7, CD3E, GZMH, RPL30, GZMA, RPS27A, IFITM1
##     CD2, KLRD1, LTB, RPS27, RPS4X, AES, LCK, RPS8, IL2RG, RPL5
## PC_ 2
## Positive:  GNLY, NKG7, CCL5, CTSW, CST7, GZMH, CCL4, KLRD1, GZMA, PRF1
##     GZMB, ACTB, FGFBP2, HOPX, S100A4, S100A8, IL32, TRGC2, GZMM, C12orf75
##     FCGR3A, LYZ, TRBC1, TRDC, CD8A, SAMD3, SPON2, CD3D, AC020656.1, ACTG1
## Negative:  IGKC, CD74, IGHM, MS4A1, HLA-DRB1, CD79A, HLA-DPA1, HLA-DRA, HLA-DQA1, BANK1
##     HLA-DQB1, LINC00926, IGLC2, RALGPS2, IGHD, IGLC3, SPIB, CD79B, JCHAIN, TCL1A
##     MEF2C, MZB1, GNG7, TNFRSF13C, LTB, VPREB3, RPS8, IL7R, AFF3, EAF2
## PC_ 3
## Positive:  CD74, IGKC, HLA-DRB1, GNLY, HLA-DPA1, NKG7, IGHM, HLA-DRA, MS4A1, CCL5
##     CD79A, HLA-DQB1, HLA-DQA1, HLA-DPB1, CTSW, BANK1, CST7, GZMH, LINC00926, CCL4
##     IGHD, SPIB, IGLC2, IGLC3, KLRD1, CD79B, ACTB, MEF2C, RALGPS2, TCL1A
## Negative:  IL7R, TRAC, LTB, TRAT1, TRBC2, LDHB, S100A9, MAL, IL32, S100A8
##     NOSIP, LEF1, GIMAP7, ETS1, TCF7, RPS12, LPAR6, RCAN3, AC058791.1, CAMK4
##     AQP3, CYLD, CD3E, TNFRSF25, SERINC5, BCL2, INPP4B, CD69, JUNB, NEAT1
## PC_ 4
## Positive:  NEAT1, S100A9, MT-CO1, VCAN, MT-CO2, MT-CO3, MTRNR2L12, MT-CYB, MT-ATP6, MT-ND1
##     MT-ND4, MT-ND2, ZEB2, CTSD, TYMP, MT-ND3, XIST, GRN, S100A8, MT-ND5
##     IGKC, MYO1F, JCHAIN, MZB1, PLCG2, CSF3R, PSAP, RNF213, PTCH2, MT-ND4L
## Negative:  FTH1, LYZ, AC020656.1, ACTB, CST3, RPL10, HLA-DRB1, B2M, RPS12, S100A4
##     HLA-DPA1, IFITM3, RPL30, IL7R, HLA-DRA, RPS8, S100A10, RPL32, RPS27A, CD74
##     RPL5, S100A6, RPS4X, RPS27, FTL, IL32, COTL1, S100A11, HLA-DQB1, SH3BGRL3
## PC_ 5
## Positive:  CD74, HLA-DRB1, HLA-DRA, NEAT1, HLA-DPA1, HLA-DQB1, HLA-DPB1, HLA-DQA1, CST3, MS4A
1
##     JUN, ZEB2, MT-CO1, JUND, VCAN, IFITM3, TYMP, MTRNR2L12, XIST, FOS
##     IGHD, ITGAX, FOSB, HLA-DMA, MT-ND1, SAT1, CLEC10A, AC103591.3, IER2, FTH1
## Negative:  JCHAIN, MZB1, HSP90B1, ITM2C, IGHG1, IGHG3, CD38, TNFRSF17, PPIB, IGHA1
##     SEC11C, FKBP11, IGKC, DERL3, IGHGP, IGHG4, LMAN1, TENT5C, SDF2L1, XBP1
##     SSR4, POU2AF1, MYBL2, SUB1, S100A8, HSPA5, AL133467.1, IGLC2, IGHA2, MANF
```

```
seu <- RunUMAP(object = seu, dims = 1:30,verbose = T)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate t
o the R-native UWOT using the cosine metric
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlatio
n'
## This message will be shown once per session
```

```
## 12:27:14 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 12:27:14 Read 7523 rows and found 30 numeric columns
```

```
## 12:27:14 Using Annoy for neighbor search, n_neighbors = 30
```

```
## 12:27:14 Building Annoy index with metric = cosine, n_trees = 50
```

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

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

```
## **************************************************|
## 12:27:15 Writing NN index file to temp file C:\Users\STRIPP~1\AppData\Local\Temp\RtmpWO9Hu0\f
ile3e8cee9197e
## 12:27:15 Searching Annoy index using 1 thread, search_k = 3000
## 12:27:17 Annoy recall = 100%
## 12:27:19 Commencing smooth kNN distance calibration using 1 thread
## 12:27:22 Initializing from normalized Laplacian + noise
## 12:27:22 Commencing optimization for 500 epochs, with 321492 positive edges
## 12:27:44 Optimization finished
```

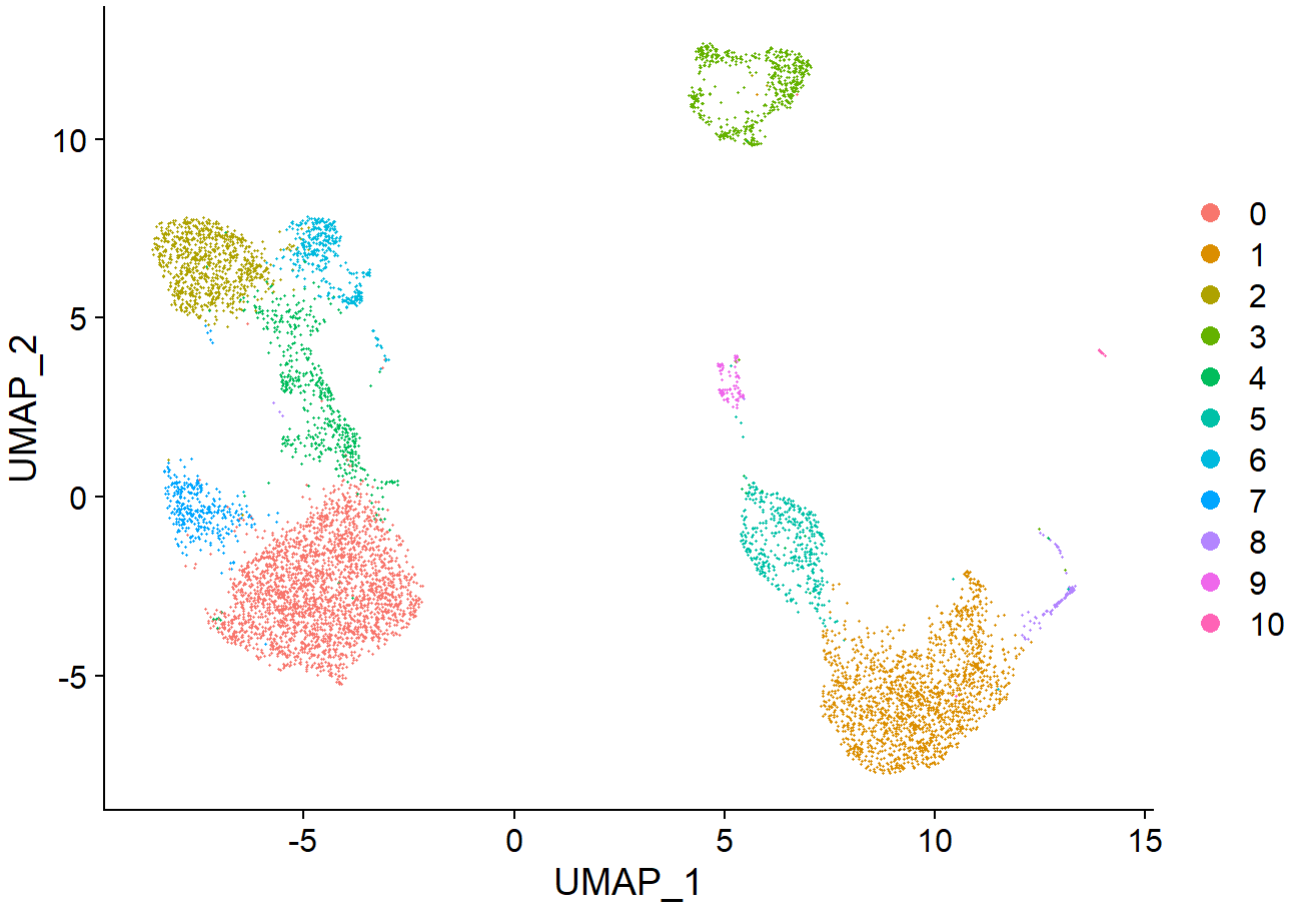
```
seu <- FindNeighbors(object = seu, dims = 1:30, verbose = T)
```

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

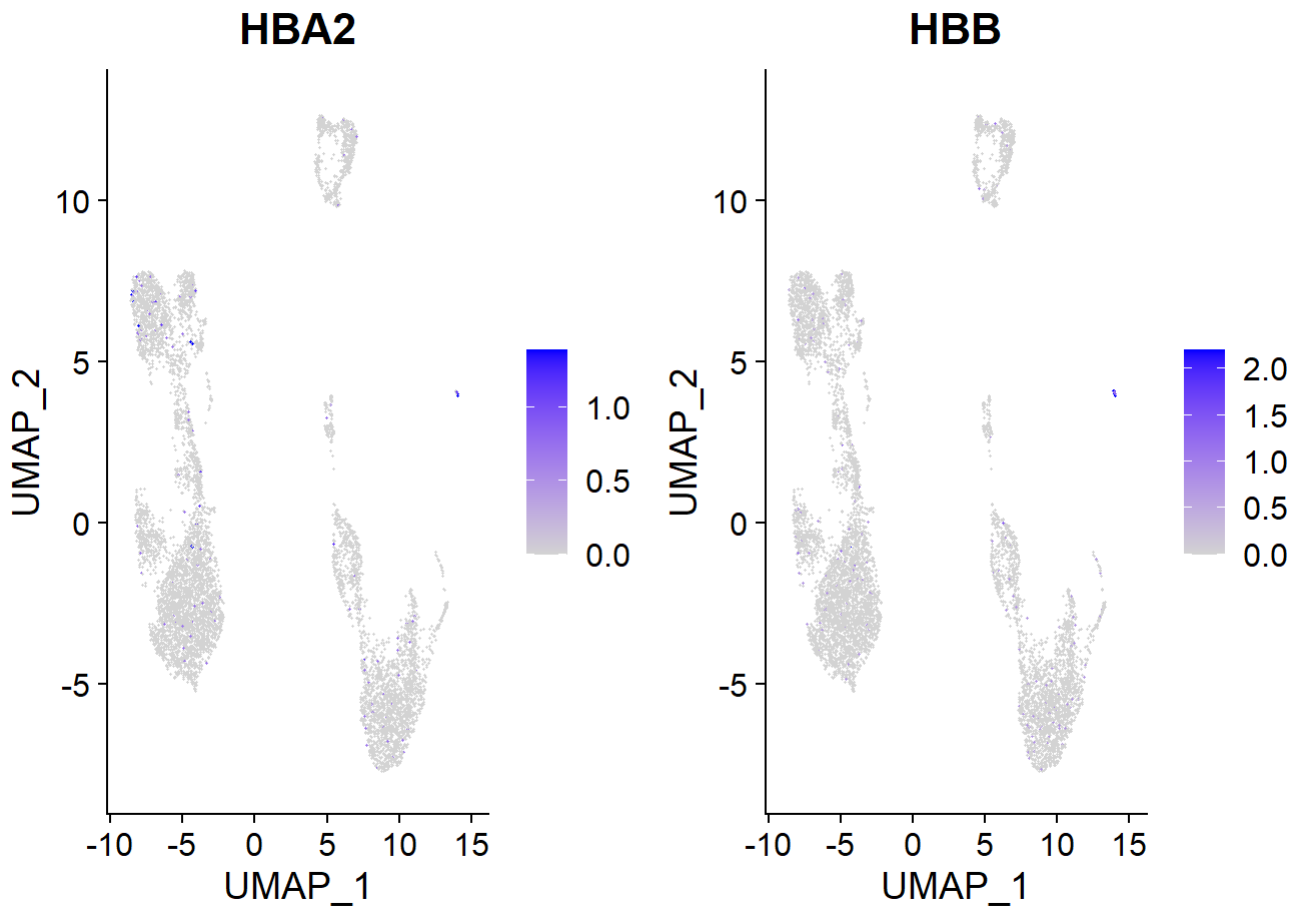
```
seu <- FindClusters(object = seu, resolution = 0.5, verbose = T)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 7523
## Number of edges: 272874
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8915
## Number of communities: 11
## Elapsed time: 1 seconds
```

```
DimPlot(seu)
```

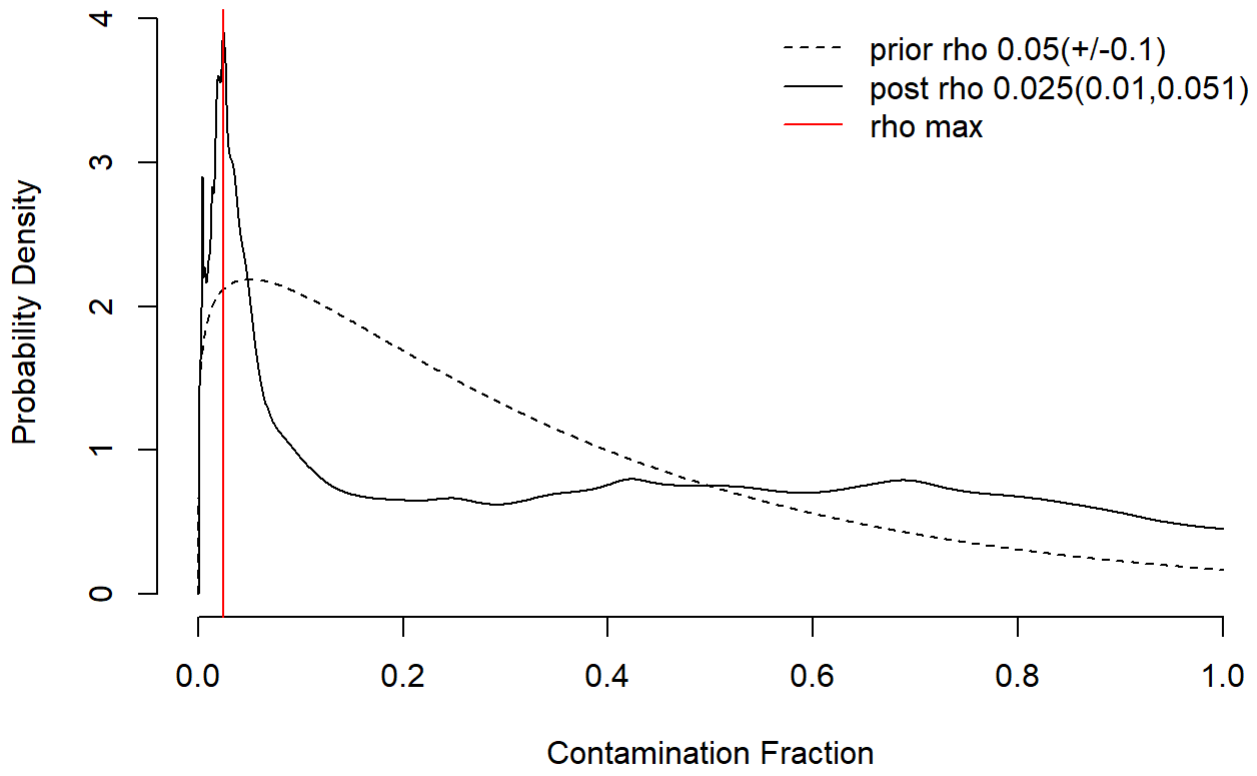



```
FeaturePlot(seu, features=c("HBA2", "HBB"), max.cutoff="q95")
```




```
Cluster <- seu$seurat_clusters
sc = setClusters(sc,Cluster)
sc = autoEstCont(sc)
```

```
## 528 genes passed tf-idf cut-off and 356 soup quantile filter.  Taking the top 100.
## Using 679 independent estimates of rho.
## Estimated global rho of 0.02
```




```
out = adjustCounts(sc)
```

```
## Expanding counts from 11 clusters to 7523 cells.
```

rerun Seurat for adjusted count matrix, check adjusted result

```
seu2 <- CreateSeuratObject(counts = out, project = "cm1_2")
seu2 <- SCTransform(object = seu2, verbose = T)
```

```
## Calculating cell attributes from input UMI matrix: log_umi
```

```
## Variance stabilizing transformation of count matrix of size 15704 by 7523
```

```
## Model formula is y ~ log_umi
```

```
## Get Negative Binomial regression parameters per gene
```

```
## Using 2000 genes, 7523 cells
```

```
##
  |
  |                                                                      |   0%
  |
  |==                                                                    |   3%
  |
  |====                                                                  |   6%
  |
  |=======                                                               |   9%
  |
  |=========                                                             |  12%
  |
  |===========                                                           |  16%
  |
  |==============                                                        |  19%
  |
  |================                                                      |  22%
  |
  |==================                                                    |  25%
  |
  |====================                                                  |  28%
  |
  |======================                                                |  31%
  |
  |========================                                              |  34%
  |
  |==========================                                            |  38%
  |
  |============================                                          |  41%
  |
  |==============================                                        |  44%
  |
  |================================                                      |  47%
  |
  |==================================                                    |  50%
  |
  |====================================                                  |  53%
  |
  |======================================                                |  56%
  |
  |========================================                              |  59%
  |
  |==========================================                            |  62%
  |
  |============================================                          |  66%
  |
  |==============================================                        |  69%
  |
  |================================================                      |  72%
  |
  |==================================================                    |  75%
  |
  |====================================================                  |  78%
```

```
  |
  |=============================================================          |  81%
  |
  |==============================================================         |  84%
  |
  |================================================================       |  88%
  |
  |===================================================================    |  91%
  |
  |======================================================================  |  94%
  |
  |========================================================================|  97%
  |
  |========================================================================| 100%
```

```
## Computing corrected count matrix for 15704 genes
```

```
##
  |
  |                                                                      |   0%
  |
  |==                                                                    |   3%
  |
  |====                                                                  |   6%
  |
  |=======                                                               |   9%
  |
  |=========                                                             |  12%
  |
  |===========                                                           |  16%
  |
  |==============                                                        |  19%
  |
  |================                                                      |  22%
  |
  |==================                                                    |  25%
  |
  |====================                                                  |  28%
  |
  |======================                                                |  31%
  |
  |=========================                                             |  34%
  |
  |===========================                                           |  38%
  |
  |============================                                          |  41%
  |
  |==============================                                        |  44%
  |
  |================================                                      |  47%
  |
  |==================================                                    |  50%
  |
  |====================================                                  |  53%
  |
  |======================================                                |  56%
  |
  |========================================                              |  59%
  |
  |==========================================                            |  62%
  |
  |============================================                          |  66%
  |
  |==============================================                        |  69%
  |
  |================================================                      |  72%
  |
  |==================================================                    |  75%
  |
  |====================================================                  |  78%
```

```
  |
  |=============================================================          |  81%
  |
  |==============================================================         |  84%
  |
  |================================================================       |  88%
  |
  |===================================================================    |  91%
  |
  |======================================================================= |  94%
  |
  |=========================================================================  |  97%
  |
  |============================================================================| 100%
```

```
## Calculating gene attributes
```

```
## Wall clock passed: Time difference of 2.158323 mins
```

```
## Determine variable features
```

```
## Set 3000 variable features
```

```
## Place corrected count matrix in counts slot
```

```
## Centering data matrix
```

```
## Set default assay to SCT
```

```
seu2 <- RunPCA(object = seu2, verbose = T)
```

```
## PC_ 1
## Positive:  GNLY, CCL5, NKG7, IL32, IL7R, CTSW, RPL10, CD3D, TRBC2, RPS12
##     CD3G, TRBC1, CST7, TRAC, GZMH, CD3E, GZMA, RPL30, IFITM1, KLRD1
##     CD2, LTB, AES, RPS27, LCK, RPS4X, RPSA, IL2RG, ETS1, RPS29
## Negative:  S100A8, LYZ, AC020656.1, CST3, S100A9, FCN1, S100A6, IFI27, IFITM3, S100A12
##     TYROBP, FTL, MNDA, S100A11, AIF1, FTH1, S100A10, S100A4, FCER1G, CEBPD
##     CTSS, NEAT1, VCAN, CTSD, CFD, CSTA, CD14, LGALS3, CXCL8, PLBD1
## PC_ 2
## Positive:  IGKC, CD74, IGHM, MS4A1, HLA-DRB1, CD79A, HLA-DPA1, HLA-DRA, HLA-DQA1, IGLC2
##     HLA-DQB1, BANK1, IGLC3, LINC00926, RALGPS2, IGHD, JCHAIN, SPIB, CD79B, TCL1A
##     MEF2C, MZB1, GNG7, TNFRSF13C, VPREB3, EAF2, AFF3, LTB, IGHA1, RPS8
## Negative:  GNLY, NKG7, CCL5, CTSW, CST7, GZMH, CCL4, KLRD1, GZMA, PRF1
##     ACTB, GZMB, FGFBP2, HOPX, S100A4, IL32, GZMM, S100A8, TRGC2, C12orf75
##     TRBC1, FCGR3A, LYZ, TRDC, CD8A, SAMD3, CD3D, SPON2, AC020656.1, ACTG1
## PC_ 3
## Positive:  CD74, IGKC, HLA-DRB1, GNLY, NKG7, HLA-DPA1, IGHM, HLA-DRA, MS4A1, CCL5
##     CD79A, HLA-DQB1, HLA-DQA1, HLA-DPB1, CTSW, BANK1, CST7, GZMH, IGLC2, CCL4
##     LINC00926, ACTB, IGHD, SPIB, KLRD1, IGLC3, CST3, CD79B, RALGPS2, TCL1A
## Negative:  IL7R, TRAC, LTB, TRAT1, TRBC2, LDHB, S100A9, MAL, IL32, NOSIP
##     LEF1, GIMAP7, S100A8, TCF7, RPS12, ETS1, LPAR6, RCAN3, AC058791.1, CAMK4
##     CYLD, AQP3, TNFRSF25, NEAT1, SERINC5, BCL2, CD69, JUNB, CD3E, INPP4B
## PC_ 4
## Positive:  S100A9, NEAT1, MT-CO1, VCAN, MT-CO2, MT-CO3, MTRNR2L12, MT-CYB, MT-ATP6, MT-ND1
##     MT-ND4, S100A8, MT-ND2, CTSD, ZEB2, MT-ND3, TYMP, XIST, IGKC, GRN
##     JCHAIN, MZB1, MT-ND5, MYO1F, PLCG2, CSF3R, PSAP, GNLY, RNF213, PTCH2
## Negative:  FTH1, ACTB, CST3, LYZ, AC020656.1, HLA-DRB1, RPL10, IFITM3, RPS12, HLA-DPA1
##     B2M, HLA-DRA, S100A4, RPL30, IL7R, CD74, S100A10, RPS8, COTL1, HLA-DQB1
##     RPL5, S100A6, FTL, HLA-DPB1, IL32, S100A11, RPS4X, RPS27, SH3BGRL3, ARPC1B
## PC_ 5
## Positive:  CD74, HLA-DRB1, NEAT1, HLA-DRA, HLA-DPA1, HLA-DQB1, MS4A1, HLA-DQA1, HLA-DPB1, MT-
CO1
##     VCAN, JUN, ZEB2, MTRNR2L12, CST3, TYMP, JUND, IGHD, XIST, MT-ND1
##     LINC00926, FOSB, ITGAX, MT-ND3, S100A9, MT-ND2, BANK1, AC103591.3, PSAP, MT-ND4
## Negative:  JCHAIN, MZB1, HSP90B1, ITM2C, IGHG1, IGHG3, CD38, TNFRSF17, PPIB, IGHA1
##     SEC11C, FKBP11, DERL3, IGHGP, IGHG4, LMAN1, TENT5C, SDF2L1, XBP1, MYBL2
##     POU2AF1, SSR4, IGKC, AL133467.1, AC020656.1, HSPA5, HRASLS2, MANF, IGHA2, PRDX4
```

```
seu2 <- RunUMAP(object = seu2, dims = 1:30,verbose = T)
```

```
## 12:30:36 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 12:30:36 Read 7523 rows and found 30 numeric columns
```

```
## 12:30:36 Using Annoy for neighbor search, n_neighbors = 30
```

```
## 12:30:36 Building Annoy index with metric = cosine, n_trees = 50
```

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

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

```
## **************************************************|
## 12:30:37 Writing NN index file to temp file C:\Users\STRIPP~1\AppData\Local\Temp\RtmpWO9Hu0\f
ile3e8c56837fef
## 12:30:37 Searching Annoy index using 1 thread, search_k = 3000
## 12:30:39 Annoy recall = 100%
## 12:30:41 Commencing smooth kNN distance calibration using 1 thread
## 12:30:44 Initializing from normalized Laplacian + noise
## 12:30:44 Commencing optimization for 500 epochs, with 317972 positive edges
## 12:31:06 Optimization finished
```

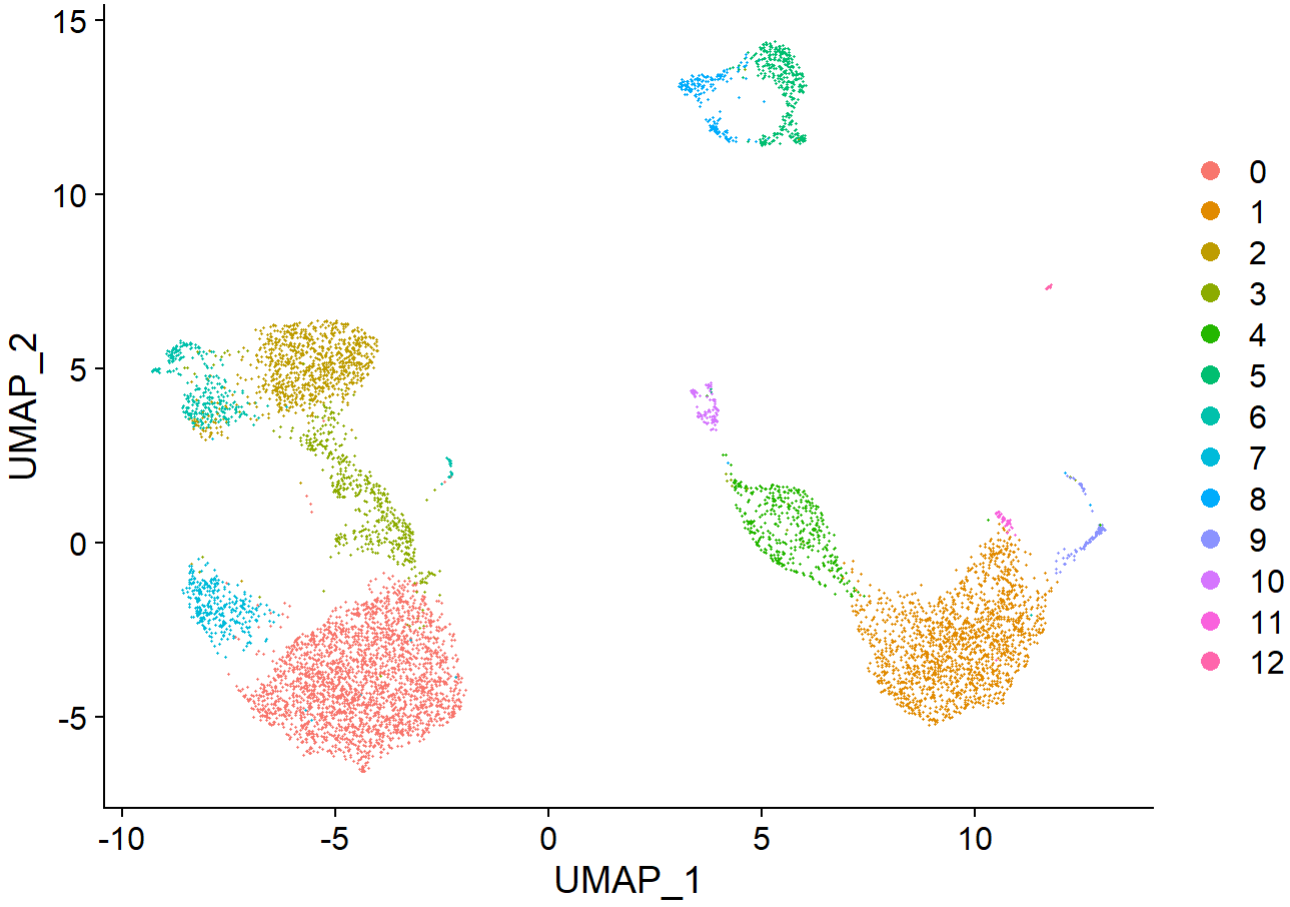
```
seu2 <- FindNeighbors(object = seu2, dims = 1:30, verbose = T)
```

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

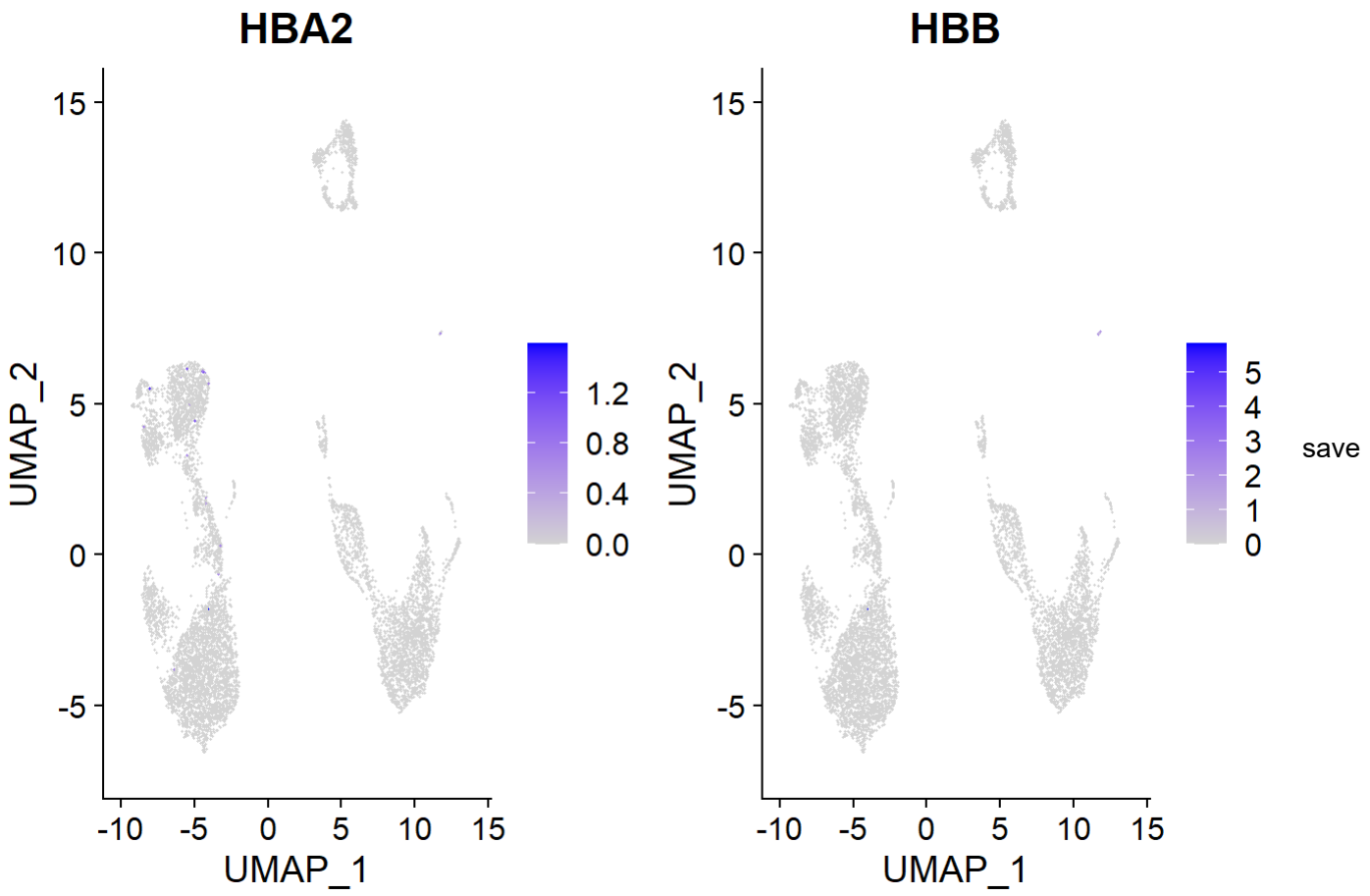
```
seu2 <- FindClusters(object = seu2, resolution = 0.5, verbose = T)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 7523
## Number of edges: 266992
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8943
## Number of communities: 13
## Elapsed time: 1 seconds
```

```
DimPlot(seu2)
```




```
FeaturePlot(seu2, features=c("HBA2", "HBB"), max.cutoff="q95")
```



SoupX adjust counted matrix

```
write10xCounts('G://covid19scRNAseq/CM1_2/outs/desoup5', out)
```