R Notebook

```
library(Seurat)
```

```
## Attaching SeuratObject
```

```
## 'SeuratObject' was built under R 4.3.0 but the current version is
## 4.3.1; it is recomended that you reinstall 'SeuratObject' as the ABI
## for R may have changed
```

```
library(Matrix)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x tidyr::expand() masks Matrix::expand()
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tidyr::pack()   masks Matrix::pack()
## x tidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(biomaRt)
library(sctransform)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
##
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
mat <- readMM(file = "matrix.mtx")
feature.names = read.delim("genes.tsv",
                           header = FALSE,
                           stringsAsFactors = FALSE)
barcode.names = read.delim("barcodes.tsv",
                           header = FALSE,
                           stringsAsFactors = FALSE)
colnames(mat) = seq(1,80000)
rownames(mat) = feature.names$V1

MouseSampleTagList <- sapply(1:12, (function(x)paste0("MouseSampleTag", formatC(x,width = 2, format = "
MouseSampleTagData <- mat[feature.names$V1 %in% MouseSampleTagList,]
```

```
SeqData <- mat[!feature.names$V1 %in% MouseSampleTagList,]

SeqData <- CreateSeuratObject(counts = SeqData, min.cells = 3)


# Add MouseSampleTag data as a new assay independent from RNA
SeqData[["MouseSampleTag"]] <- CreateAssayObject(counts = MouseSampleTagData)

# Normalize MouseSampleTag data, here we use centered log-ratio (CLR) transformation
SeqData <- NormalizeData(SeqData, assay = "MouseSampleTag", normalization.method = "CLR")
```

```
## Normalizing across features
```

```
# Identify thresholds for filtering
SeqData[["percent.mt"]] <- PercentageFeatureSet(SeqData, pattern = "^mt-")
#VlnPlot(SeqData, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3, pt.size = 0)
#plot1 <- FeatureScatter(SeqData, feature1 = "nCount_RNA", feature2 = "percent.mt")
#plot2 <- FeatureScatter(SeqData, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
#plot3 <- FeatureScatter(SeqData, feature1 = "nFeature_RNA", feature2 = "percent.mt")
#plot1 + plot2 + plot3

# Filtering
# SeqData <- subset(SeqData, subset =  nFeature_RNA > 500 & percent.mt < 10 & nFeature_RNA < 4000)
SeqData <- subset(SeqData, subset = nFeature_RNA > 1600 & nCount_RNA < 60000 & percent.mt < 10)


# Demultiplexing based on mouse sample tags
SeqData <- HTODemux(SeqData, assay = "MouseSampleTag", positive.quantile = 0.99)
```

```
## Cutoff for MouseSampleTag05 : 33 reads
## Cutoff for MouseSampleTag03 : 39 reads
## Cutoff for MouseSampleTag02 : 311 reads
## Cutoff for MouseSampleTag10 : 26 reads
## Cutoff for MouseSampleTag04 : 19 reads
## Cutoff for MouseSampleTag09 : 24 reads
## Cutoff for MouseSampleTag11 : 10 reads
## Cutoff for MouseSampleTag08 : 43 reads
## Cutoff for MouseSampleTag07 : 10 reads
## Cutoff for MouseSampleTag01 : 8 reads
## Cutoff for MouseSampleTag06 : 7 reads
## Cutoff for MouseSampleTag12 : 5 reads
```

```
table(SeqData$MouseSampleTag_classification.global)
```

```
##
##  Doublet Negative  Singlet
##     6741       15     9412
```

```
rm(mat, barcode.names, feature.names, MouseSampleTagData)
```

```r
# Remove doubletts
Idents(SeqData) <- "MouseSampleTag_classification.global"
SeqData <- subset(SeqData, idents = "Singlet")



# Annotating datasets
Idents(SeqData) <- "MouseSampleTag_classification"

Maf_Seurat_WT <- subset(SeqData, idents = c("MouseSampleTag01", "MouseSampleTag08", "MouseSampleTag11"))
Maf_Seurat_KO <- subset(SeqData, idents = c("MouseSampleTag04", "MouseSampleTag06", "MouseSampleTag12"))

Tgfb_Seurat_WT <- subset(SeqData, idents = c("MouseSampleTag02", "MouseSampleTag05", "MouseSampleTag09"))
Tgfb_Seurat_KO <- subset(SeqData, idents = c("MouseSampleTag03", "MouseSampleTag07", "MouseSampleTag10"))


rm(SeqData)

# Normalization
Maf_Seurat_WT <- SCTransform(Maf_Seurat_WT, vars.to.regress = "percent.mt")
Maf_Seurat_KO <- SCTransform(Maf_Seurat_KO, vars.to.regress = "percent.mt")
Tgfb_Seurat_WT <- SCTransform(Tgfb_Seurat_WT, vars.to.regress = "percent.mt")
Tgfb_Seurat_KO <- SCTransform(Tgfb_Seurat_KO, vars.to.regress = "percent.mt")
```

```r
# Generate UMAP for wildtype Maf dataset

Maf_Seurat_WT <- RunPCA(Maf_Seurat_WT)
```

```
## PC_ 1
## Positive:  Malat1, Slfn5, Samhd1, Slfn1, Shisa5, Ifi27l2a, Rps27, Rflnb, Selenop, Ms4a4b
##     Gimap3, Izumo1r, Ifi209, Ifi213, Mxd4, H2-K1, Gm8995, Laptm5, Macf1, Trp53inp1
##     Rpl17, Ccnd2, Actn1, Sp100, Ifi208, Rnf213, Ifi206, Ifit1, Ifi203, Srgn
## Negative:  Mki67, Pclaf, Hmgb2, Stmn1, Ptma, Top2a, Tubb5, Ccna2, Birc5, Tuba1b
##     Rrm2, Smc2, Tpx2, Cenpe, Lmnb1, Tmpo, Cdk1, Nusap1, Kif11, Cdca8
##     Ncapg, Rrm1, Hmmr, Lig1, Tubb4b, Ube2c, Ccnb2, Mcm5, Prc1, Pcna
## PC_ 2
## Positive:  Hsp90ab1, Ncl, Eif5a, Hsp90aa1, Mif, Npm1, Maf, Pa2g4, Hspd1, Nop56
##     Nfatc1, Angptl2, Nolc1, Pcna, Nop58, Ddx21, Hspa8, Il21, Dkc1, Nme1
##     Set, Ccnd2, Hnrnpab, Ybx3, Eea1, Gnl3, Hspa9, Eif4a1, Srm, Tnfrsf4
## Negative:  Sell, Vim, S1pr1, Klf2, Slfn5, Ms4a4b, Gm8995, Cdc25b, Ifi209, Ifi203
##     Tmem71, Ifi208, S100a10, Slfn1, Ifi213, Ifit1, Cenpa, Ripor2, Ifi206, Samhd1
##     Cytip, Add3, 9930111J21Rik2, Anxa2, Cenpe, Ifi27l2a, Mndal, Hmgb2, Ablim1, Tmsb4x
## PC_ 3
## Positive:  Sell, Vim, S100a6, Tagln2, Arl4c, Klf2, Anxa2, Satb1, Icos, Cd52
##     Ncl, Ahnak, Crip1, Trib2, S1pr1, Lsp1, Dtx1, Anxa6, Eif5a, Nop56
##     Itgb7, Ccr7, Hspd1, Gnl3, Nolc1, Hsp90b1, Flna, Nop58, Npm1, Esyt1
## Negative:  Nfatc1, Maf, H2-Q2, Ccnd2, Tbc1d4, Ikzf2, Il21, Malat1, Angptl2, Tnfsf8
##     Izumo1r, Rgs10, Smc4, Rnaset2a, Bcl6, Cxcr5, Ptprc, Ephx1, Ucp2, P2rx7
##     Eea1, Fyn, Tox2, Myb, Kif15, Srgn, Cebpa, Ptms, Ptpn11, Tox
## PC_ 4
## Positive:  Tmsb4x, S1pr1, S100a6, Cenpa, Ccnb2, Ddit4, Trbc2, Vim, Mxd4, Rflnb
##     Klf2, Scd2, Pdcd4, Thy1, Ets1, Add3, Dtx1, H1f0, Cnn2, Selplg
##     Tcf7, Izumo1r, Lgals1, P2rx7, Tuba1a, Arl5c, Klf3, Laptm5, Cdc20, Hp1bp3
## Negative:  Ifit1, Isg15, Slfn5, Rsad2, Rnf213, Ifit3, Ifi47, Rtp4, Slfn1, Trim30a
```

```
##     Ifi213, Ifi208, Ifi203, Gm6545, Isg20, Oasl2, Ifih1, Ifi209, Gm8995, Ifit1bl1
##     Irgm1, Ifi206, Igtp, Usp18, Mndal, Slfn8, Gbp7, Ifit3b, Irf7, Dtx3l
## PC_ 5
## Positive:  Cenpf, Cenpa, Ccnb2, Cdc20, Hmmr, Aspm, Cdca8, Tpx2, Ube2c, Cenpe
##     Srgn, Hspa5, Icos, Cks2, Nfkbid, Cip2a, Tuba1c, Cep55, Kif23, Nucks1
##     Birc5, Prr11, Ckap2, Ccnb1, Knstrn, Trac, Tubb4b, Nek2, Ckap2l, Rilpl2
## Negative:  Pcna, Lig1, Mcm5, Mcm3, Uhrf1, Hells, Mcm6, Mcm4, Rrm2, Clspn
##     Cdc6, Mcm7, Ung, Atad5, Dhfr, Atad2, Chaf1a, Mcm2, Fen1, Dnmt1
##     Dtl, Rfc3, Tyms, Ccne2, Prim1, Ncapg2, Dut, Tcf19, Cdca7, Slbp
```

```
Maf_Seurat_WT <- RunUMAP(Maf_Seurat_WT, dims = 1:10, return.model = TRUE)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
```

```
## UMAP will return its model
```

```
## 12:41:11 UMAP embedding parameters a = 0.9922 b = 1.112
```

```
## 12:41:11 Read 2179 rows and found 10 numeric columns
```

```
## 12:41:11 Using Annoy for neighbor search, n_neighbors = 30
```

```
## 12:41:11 Building Annoy index with metric = cosine, n_trees = 50
```

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

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

```
## **************************************************|
## 12:41:12 Writing NN index file to temp file /var/folders/b4/j_k3f8zj6c93wxf7gbwh4q4m0000gp/T//Rtmp54
## 12:41:12 Searching Annoy index using 1 thread, search_k = 3000
## 12:41:12 Annoy recall = 100%
## 12:41:12 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 12:41:13 Initializing from normalized Laplacian + noise (using irlba)
## 12:41:13 Commencing optimization for 500 epochs, with 84442 positive edges
## 12:41:15 Optimization finished
```

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

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

```
Maf_Seurat_WT <- FindClusters(Maf_Seurat_WT, resolution = 0.3)
```

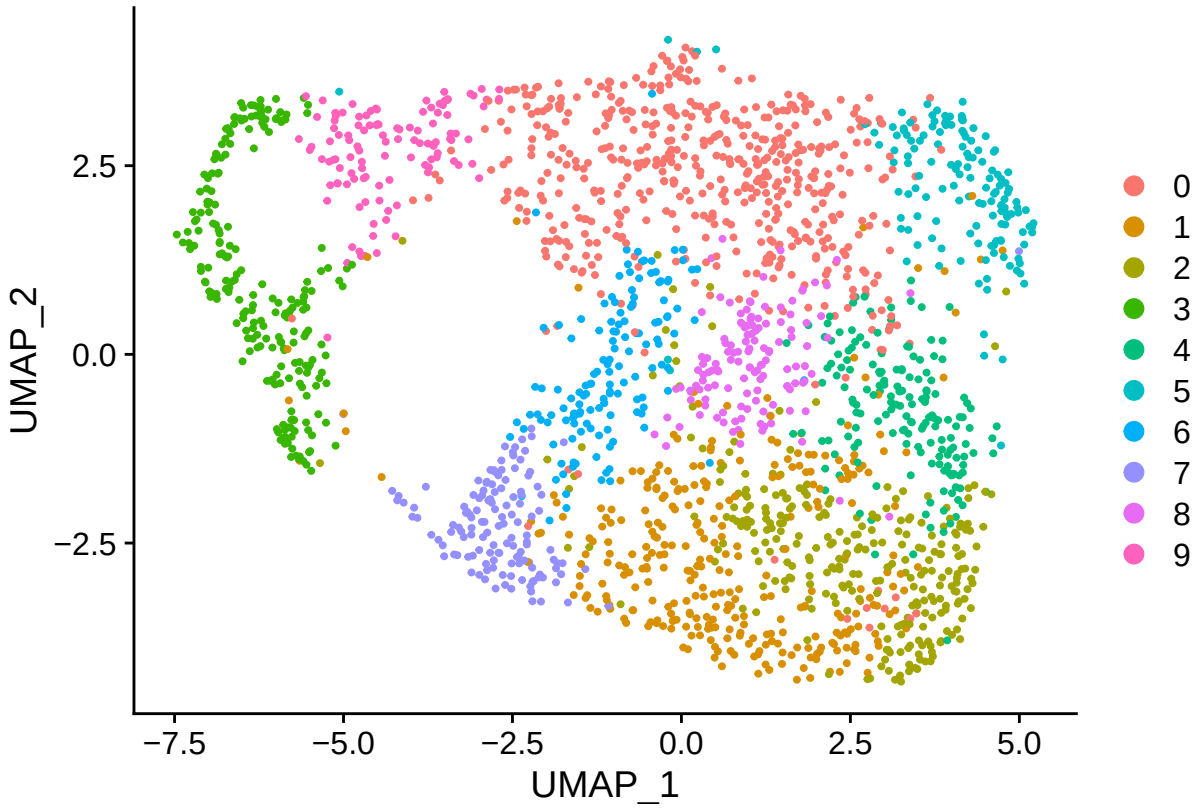
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2179
## Number of edges: 69986
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8381
## Number of communities: 6
## Elapsed time: 0 seconds
```

```
Maf_Seurat_WT <- FindClusters(Maf_Seurat_WT)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2179
```

```
## Number of edges: 69986
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7501
## Number of communities: 10
## Elapsed time: 0 seconds
```

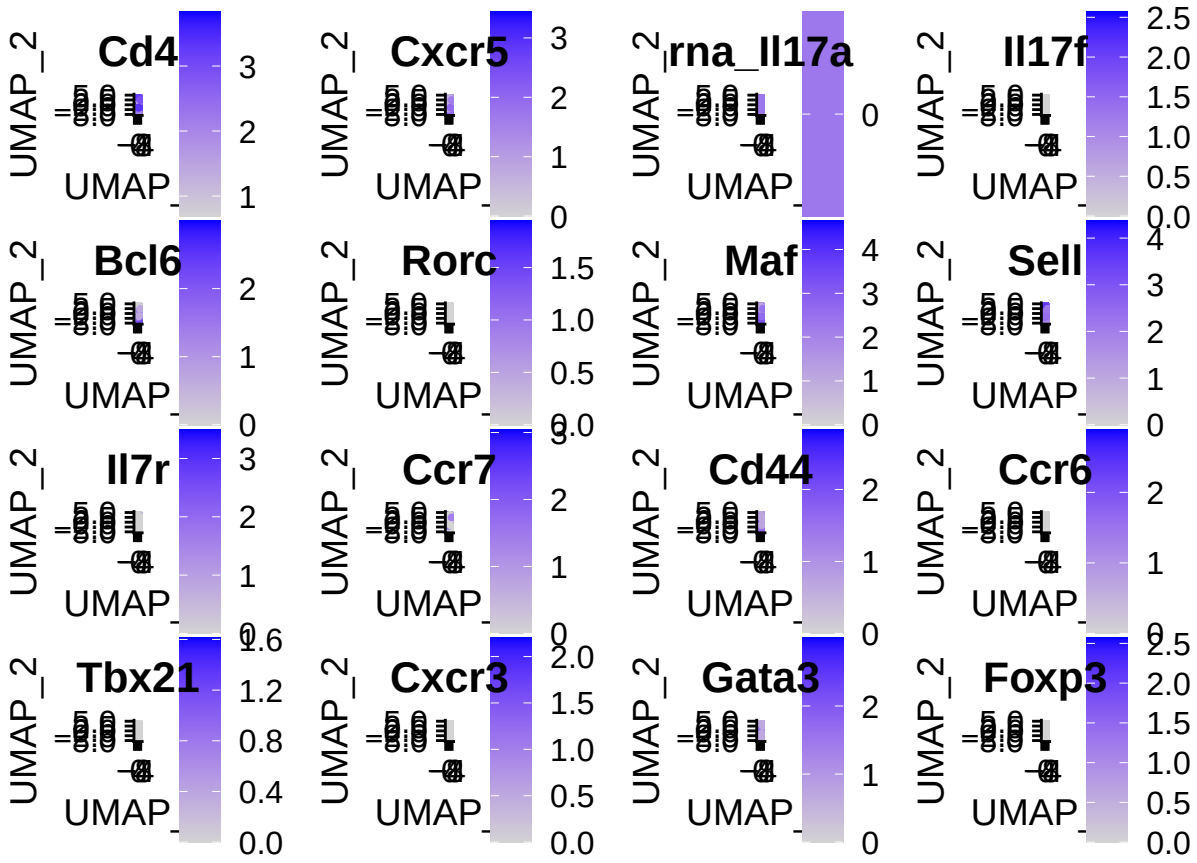
```
DimPlot(Maf_Seurat_WT, reduction = "umap")
```




```
# custom selection of genes of interest
FeaturePlot(Maf_Seurat_WT, features = c("Cd4", "Cxcr5", "Il17a", "Il17f", "Bcl6", "Rorc", "Maf", "Sell"
```

```
## Warning: Could not find Il17a in the default search locations, found in RNA
## assay instead
```

```
## Warning in FeaturePlot(Maf_Seurat_WT, features = c("Cd4", "Cxcr5", "Il17a", :
## All cells have the same value (0) of rna_Il17a.
```




```
# Project the knockout dataset onto the wldtype dataset

anchors <- FindTransferAnchors(reference = Maf_Seurat_WT, query = Maf_Seurat_KO)
```

```
## Warning in ValidateParams_FindTransferAnchors(reference = reference, query =
## query, : Both reference and query assays have been processed with
## SCTransform.Setting normalization.method = 'SCT' and continuing.
```

```
## Normalizing query using reference SCT model
```

```
## Performing PCA on the provided reference using 3000 features as input.
```

```
## Projecting cell embeddings
```

```
## Finding neighborhoods
```

```
## Finding anchors
```

```
##  Found 3475 anchors
```

```
## Filtering anchors
```

```
##  Retained 3311 anchors
```

```
#Maf_Seurat_KO <- MapQuery(anchorset = anchors, reference = Maf_Seurat_WT, query = Maf_Seurat_KO, refer


Maf_Seurat_KO <- IntegrateEmbeddings(anchorset = anchors, reference = Maf_Seurat_WT,
    query = Maf_Seurat_KO, new.reduction.name = "ref.pca")
```

```
## Requested to reuse weights matrix, but no weights found. Computing new weights.
```

```
##
## Integrating dataset 2 with reference dataset

## Finding integration vectors

## Finding integration vector weights

## Integrating data
```

```
Maf_Seurat_KO <- ProjectUMAP(query = Maf_Seurat_KO, query.reduction = "ref.pca", reference = Maf_Seurat_
    reference.reduction = "pca", reduction.model = "umap")
```

```
## Computing nearest neighbors

## Running UMAP projection

## 12:41:37 Read 2590 rows

## 12:41:37 Processing block 1 of 1

## 12:41:37 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 12:41:37 Initializing by weighted average of neighbor coordinates using 1 thread
## 12:41:37 Commencing optimization for 167 epochs, with 77700 positive edges
## 12:41:38 Finished
```

```
Maf_Seurat_KO <- RunPCA(Maf_Seurat_KO)
```

```
## PC_ 1
## Positive:  Mki67, Pclaf, Hmgb2, Stmn1, Top2a, Birc5, Ccna2, Tubb5, Tuba1b, Ptma
##     Tpx2, Rrm2, Cenpe, Smc2, Nusap1, Hmmr, Lmnb1, Ube2c, Kif11, Tubb4b
##     Cdk1, Cdca8, Cenpf, Tmpo, Ncapg, Rrm1, Knl1, Cenpa, Prc1, Lig1
## Negative:  Malat1, Ifi27l2a, Slfn5, Samhd1, Shisa5, Slfn1, H2-K1, Actn1, Mxd4, Izumo1r
##     Rflnb, Gimap3, Ccnd2, Trp53inp1, Ifi209, Cd52, Zfp36l1, Selenop, Ifi213, Macf1
##     Rps27, Ifi203, Mycbp2, Rpl17, Tbc1d4, Gimap4, Laptm5, Inpp4b, Ypel3, Ube2h
## PC_ 2
## Positive:  Hsp90ab1, Ncl, Eif5a, Hsp90aa1, Npm1, Mif, Nop56, Pa2g4, Nolc1, Tnfrsf4
##     Nop58, Hspd1, Pcna, Hnrnpab, Ptma, Hspa9, Gnl3, Mcm3, Mcm5, Set
##     Mybbp1a, Cdca7, Dkc1, Hells, Ddx21, Nme1, Mcm6, Tfrc, Srm, Il1r2
## Negative:  Sell, Vim, Slfn5, S100a6, Gm8995, Ms4a4b, S1pr1, Cdc25b, Itgb1, Ifi209
##     S100a4, S100a10, Cenpa, Arl4c, Klf2, Samhd1, Ly6a, Slfn1, Ifit1, Ifi208
##     Ifi213, Lsp1, Ube2c, Selplg, Ifi206, Cenpf, Cenpe, Tmem71, Ablim1, Add3
## PC_ 3
## Positive:  Malat1, Izumo1r, Icos, Smc4, Cd200, Tbc1d4, Arl6ip1, Cenpf, Zfp36l1, Hmmr
##     Tnfrsf4, Aspm, Nusap1, Tnfsf8, Tpx2, Cdc20, Inpp4b, Ube2c, Ccnb1, Rgs10
##     Tiam1, H2ax, Ckap2l, Agfg1, Tuba1c, H2-K1, Sesn3, Malt1, Kif23, Rnf19a
## Negative:  Vim, S100a4, Lgals1, S100a6, Crip1, Tagln2, Anxa2, S100a10, Cfl1, Pcna
##     Mcm3, Ly6a, Il18r1, Slfn5, Mcm5, Uhrf1, Lsp1, Cdc6, Ahnak, Calm1
##     Itgb7, Mcm6, Tmsb4x, Hells, Mcm4, Ccr2, Thy1, Flna, Hsp90b1, Esyt1
## PC_ 4
## Positive:  S100a4, Vim, Tmsb4x, S100a6, Lgals1, Cfl1, Dtx1, Cd7, Actb, Cd52
##     Ddit4, Itgb1, Cenpa, Cdc25b, Klf2, Arl4c, Crip1, Il18r1, Ccnb2, Anxa2
##     Cd22, S1pr1, Mxd4, mt-Cytb, Rps2, Ccr6, Tmsb10, Flna, Dap, Lcp1
## Negative:  Ifit1, Isg15, Slfn5, Rnf213, Ifit3, Ifi213, Ifi208, Trim30a, Rsad2, Rtp4
##     Ifit1bl1, Slfn1, Isg20, Oasl2, Ifih1, Mndal, Ifi206, Ifi203, Gm6545, Gbp7
##     Ifi209, Ifi27l2a, Ifi47, Slfn8, Ccnd2, Ddx60, Irf7, Stat1, Samd9l, Ddx58
## PC_ 5
## Positive:  Lig1, Pcna, Mcm5, Uhrf1, Mxd4, Rrm2, Mcm3, Hells, Mcm6, Mcm4
##     Clspn, Izumo1r, Cdc6, Atad2, Mcm7, Rflnb, Dhfr, Atad5, Ung, Fam111a
##     Fen1, Dtl, Ucp2, Prim1, Chaf1a, Smc4, Tyms, Ncapg2, Ccne2, Ezh2
```

```
## Negative:  Nfkbid, S100a4, Irf4, Srgn, Egr3, Hspa5, Nfkbia, Kdm6b, Rilpl2, Gnl3
##     Egr1, Nolc1, Tnfrsf4, Cenpa, Nr4a3, Nr4a1, Ddx21, Icos, Myc, Ccnb2
##     Tagap, Ncl, Hsp90b1, Nop58, Crip1, Pim1, Dusp2, Cenpf, Ly6a, Cd69
```

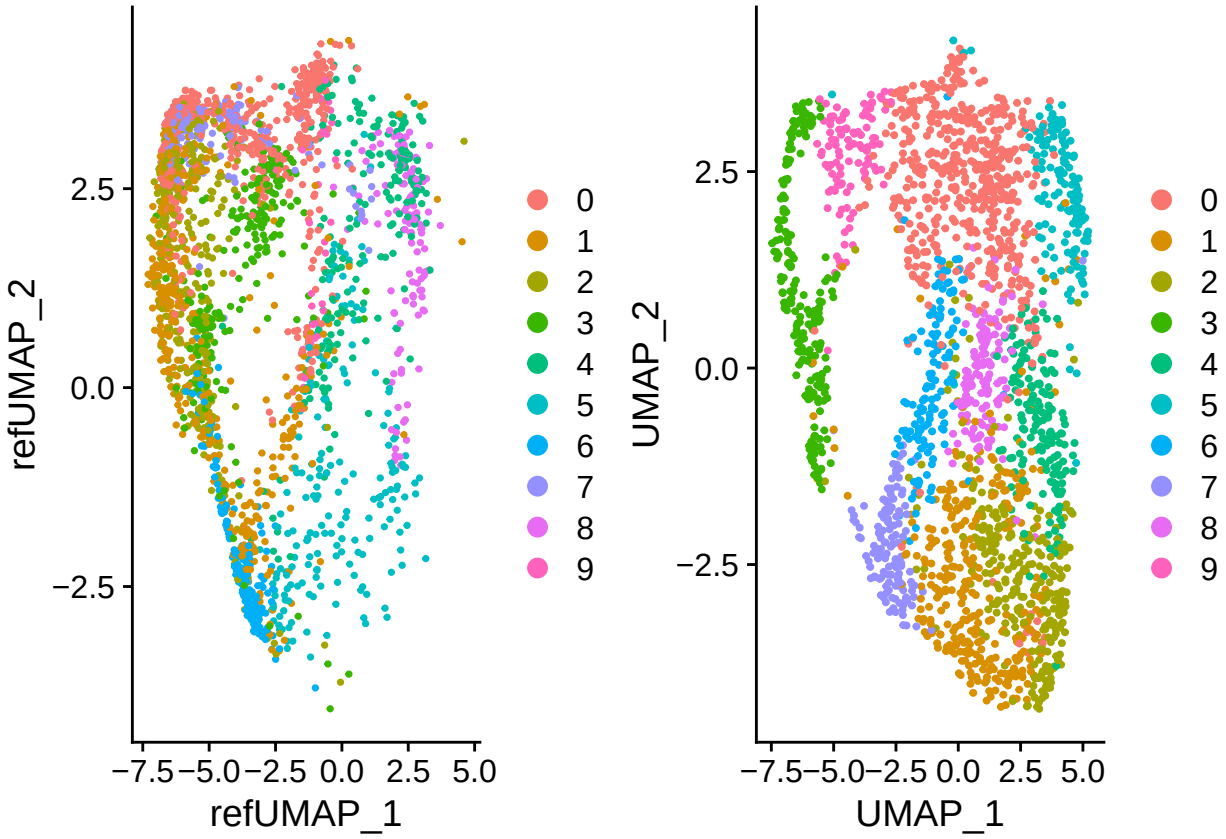
```
Maf_Seurat_KO <- FindNeighbors(Maf_Seurat_KO, dims = 1:10)
```

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

```
Maf_Seurat_KO <- FindClusters(Maf_Seurat_KO)
```

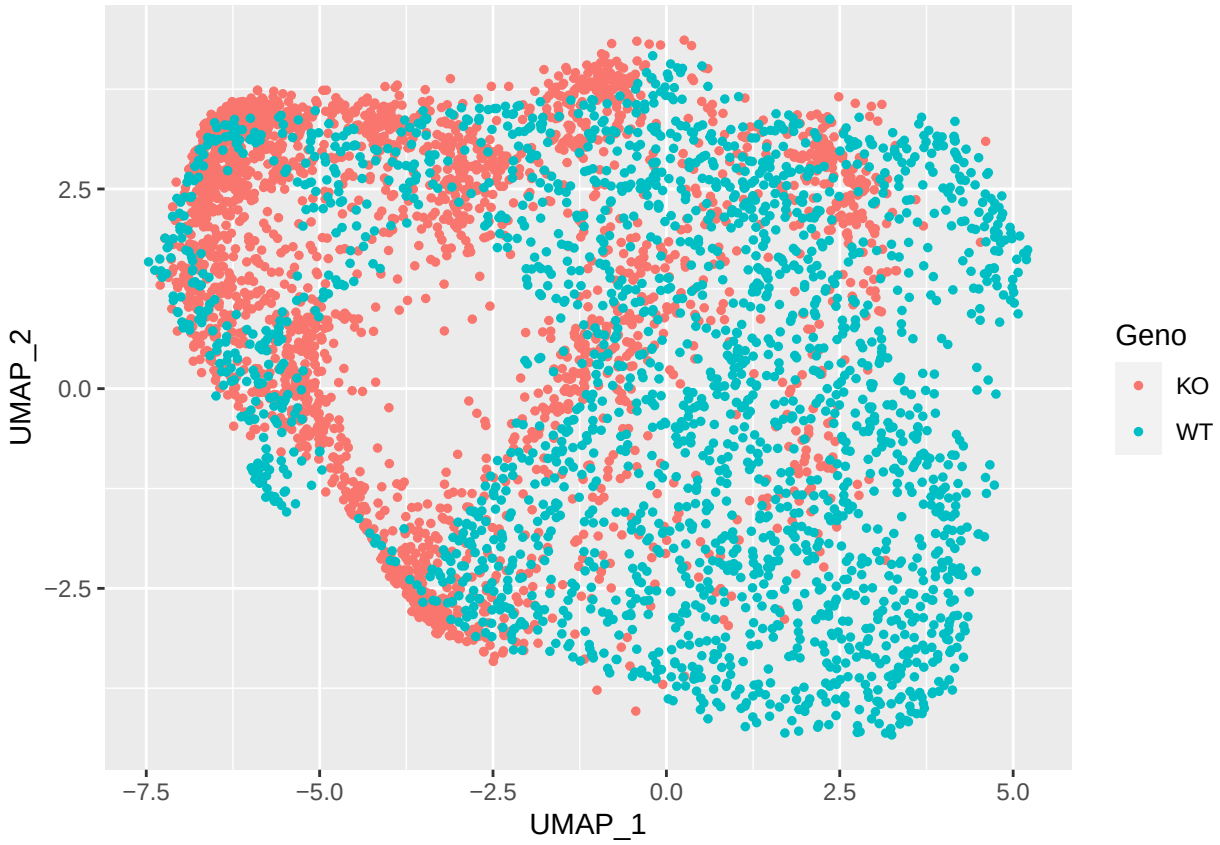
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 2590
## Number of edges: 83895
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.7726
## Number of communities: 10
## Elapsed time: 0 seconds
```

```
# Separate plotting
test_KO <- DimPlot(Maf_Seurat_KO, reduction = "ref.umap", combine = FALSE)[[1]]
test_WT <- DimPlot(Maf_Seurat_WT, reduction = "umap", combine = FALSE)[[1]]

grid.arrange(test_KO, test_WT, ncol=2)
```




```
# Combined plotting
test_KO_df <- test_KO$data %>% mutate(UMAP_1 = refUMAP_1, UMAP_2 = refUMAP_2, Geno = "KO") %>% dplyr::s
test_WT_df <- test_WT$data %>% mutate(Geno = "WT")
```

```
test <- bind_rows(test_KO_df, test_WT_df)

ggplot(test, aes(UMAP_1,UMAP_2, col = Geno)) + geom_point(size = 1)
```




```
# Plotting Tfh and Th17 features on the wildtype datasets
# Revert human gene formating to mgi
convertHumanGeneList <- function(x){
  return(unname(sapply(x, (function(y) paste0(substr(y, 1,1), tolower(substring(y,2)))))))
}

TfhVTh17_UP <- read.delim("genesets/GSE11924_TFH_VS_TH17_CD4_TCELL_UP.txt") %>% tail(-1) %>% .[[1]] %>%
  convertHumanGeneList() %>% list()
TfhVTh17_DN <- read.delim("genesets/GSE11924_TFH_VS_TH17_CD4_TCELL_DN.txt") %>% tail(-1) %>% .[[1]] %>%
  convertHumanGeneList() %>% list()

Maf_Seurat_WT <- AddModuleScore(
  object = Maf_Seurat_WT,
  features = TfhVTh17_UP,
  name = 'TfhVTh17_UP'
)
```

```
## Warning: The following features are not present in the object: Angptl1,
## C15orf65, Cables1, Cyp3a7, Dnaaf9, Fhip1a, Gnmt, Icoslg, Katnip, Kiaa1958,
## Mest, Mrgprg, Slc1a3, Stx4, Zbed6, Znf157, Znf708, Znf841, not searching for
## symbol synonyms
```

```
Maf_Seurat_WT <- AddModuleScore(
  object = Maf_Seurat_WT,
```

```
  features = TfhVTh17_DN,
  name = 'TfhVTh17_DN'
)
```

```
## Warning: The following features are not present in the object: Adh4, Apln, Boc,
## Cilp2, Crb1, Csmd3, Cysltr1, Depdc7, Dnah6, Ecrg4, Efs, Eif1ay, Fgf7, Fmod,
## Foxl1, Fstl4, Gpr20, Hla-doa, Kcne5, Kcnj12, Kcnj9, Kiaa0825, Krt33b, Lama5,
## Mepe, Mog, Npl, Pdgfc, Prl, Rflna, Rpe65, Rtkn, Sanbr, Sobp, Tll2, Tlr9,
## Tmem236, Zfta, Znf511, Znf706, Znf830, not searching for symbol synonyms
```

```
FeaturePlot(Maf_Seurat_WT, reduction = "umap", features = c("TfhVTh17_UP1", "TfhVTh17_DN1"))
```