

scRNaseq RTM WT vs KO

Abinet Joan

2024-12-09 16:10:05 +0100

Contents

Introduction	2
Load packages	2
Loading Data	2
Importing Lung Macrophages	3
Combining Macrophage together	4
Quality control	4
Pre-processing workflow	5
Visualizing all clusters	6
Differential expression analysis	8
Removing contaminent, based on marker annotation.	8
Defining subset of interest	9
saving data	10

Introduction

Fifty thousand CD45+Lin-Ly6C-CD11b+CD115+ live cells from the peritoneal lavage, 50000 CD45+Lin-Ly6C-CD11bint/+F4/80+CD64+ live cells from the livers, 50000 CD45+Lin-CD11b+F4/80+CD64+ live cells from the colons and 50000 CD45+Lin-Ly6C-CD11b+F4/80+CD64+ live cells from the brains of Lyz2CreMafbfl/fl mice and Mafbfl/fl littermate controls were FACS-sorted in PBS with 5% FBS. Sorted cells were barcoded with TotalSeq™-B anti-mouse Hashtag antibodies (Biolegend). Hashtag barcoded cells were washed, pooled per genotype, spun down and resuspended in PBS with 0.04% UltraPure™ BSA (Invitrogen, AM2616) at an estimated final concentration of 1000 cells/ μ l. Cell suspensions were loaded into the Chromium Controller (10x Genomics) at a target recovery of 2000 cell per sorted population, to generate single-cell Gel Bead-in-EMulsion (GEMs). Single-cell RNA-Seq libraries were prepared using Chromium Single Cell 3'2032 GEM, Library & Gel Bead Kit v3 (10x Genomics) according to manufacturer's instructions.

The cDNAs were amplified and libraries compatible with Illumina sequencers were generated using Chromium Single Cell 3C GEM, Library & Gel Bead Kit v3 (10x Genomics). For Hash Tag Oligonucleotide (HTO) library, 1 μ l HTO additive primer v2 (0.2 μ M stock) were added to the mix at the cDNA amplification step. The libraries were sequenced on an Illumina NovaSeq sequencer on an SP100 cell flow (read 1, 28 cy; read 2, 76 cy; index 1, 10 cy; index 2, 10 cy) at a depth of 50,000 reads per cell. Demultiplexing, alignment to the mouse genome (GRCm38/mm10), filtering, unique molecular identifier counting and construction of gene-barcode matrices of the single cell RNA-Seq data were performed using Cell Ranger (7.1.0). Single cell RNA-Seq data were further analyzed using R Bioconductor (4.3.3) and the Seurat package (4.3.0). All samples were merged and integrated with previously published single cell RNA-seq data containing CD45+SSClCD11b+F4/80+CD64+ lung IM cells of Lyz2CreMafbfl/fl mice and Mafbfl/fl littermate controls (GSE193891). Next, filtered matrices containing cell IDs and feature names for each sample were used to construct a Seurat object. Quality control was performed by excluding cells with fewer than 200 detected genes, genes detected in fewer than three cells, and cells with more than 10% mitochondrial gene content. Gene counts for each sample were normalized using the default LogNormalize method, with a scale factor of 10,000 followed by log transformation. The top 2,000 highly variable features were identified using the vst method. Contaminating and actively proliferating cells were removed based on the expression of specific genes. Cells were clustered using the FindClusters function and the macrophage subsets were identified based on the expression of Hashtag barcodes and known macrophage subset specific genes. DEGs between macrophages subsets from Mafbfl/fl and Lyz2CreMafbfl/fl mice were determined using the FindMarkers method.

Load packages

```
suppressMessages(library(dplyr))
suppressMessages(library(Seurat))
suppressMessages(library(patchwork))
suppressMessages(library(ggplot2))
suppressMessages(library(stringr))
suppressMessages(library(SingleR))
suppressMessages(library(scuttle))
suppressMessages(library(scRNAseq))
suppressMessages(library(SingleCellExperiment))
suppressMessages(library(RColorBrewer))
suppressMessages(library(formatR))
```

Loading Data

The count files can be download from GEO

```
all_dirs <- dir(path = "Counts") # Specifiy your 10x directory
```

```

list_sample <- list()
for (i in 1:length(all_dirs)) {

  print(i)
  Seq_raw_file <- Read10X(data.dir = paste0("Counts/", all_dirs[i]))
  Seurat_file <- CreateSeuratObject(counts = Seq_raw_file$`Gene Expression`,
    all_dirs[i], min.cells = 3, min.features = 200)
  list_sample <- append(list_sample, Seurat_file)
}

Sc_Macro <- merge(list_sample[[1]], y = list_sample[-1], add.cell.ids = c("P2_1",
  "P2_2", "P1_1", "P1_2", "P2_4", "P2_5", "P1_4", "P1_5"),
  project = "Macro_whole_body")

```

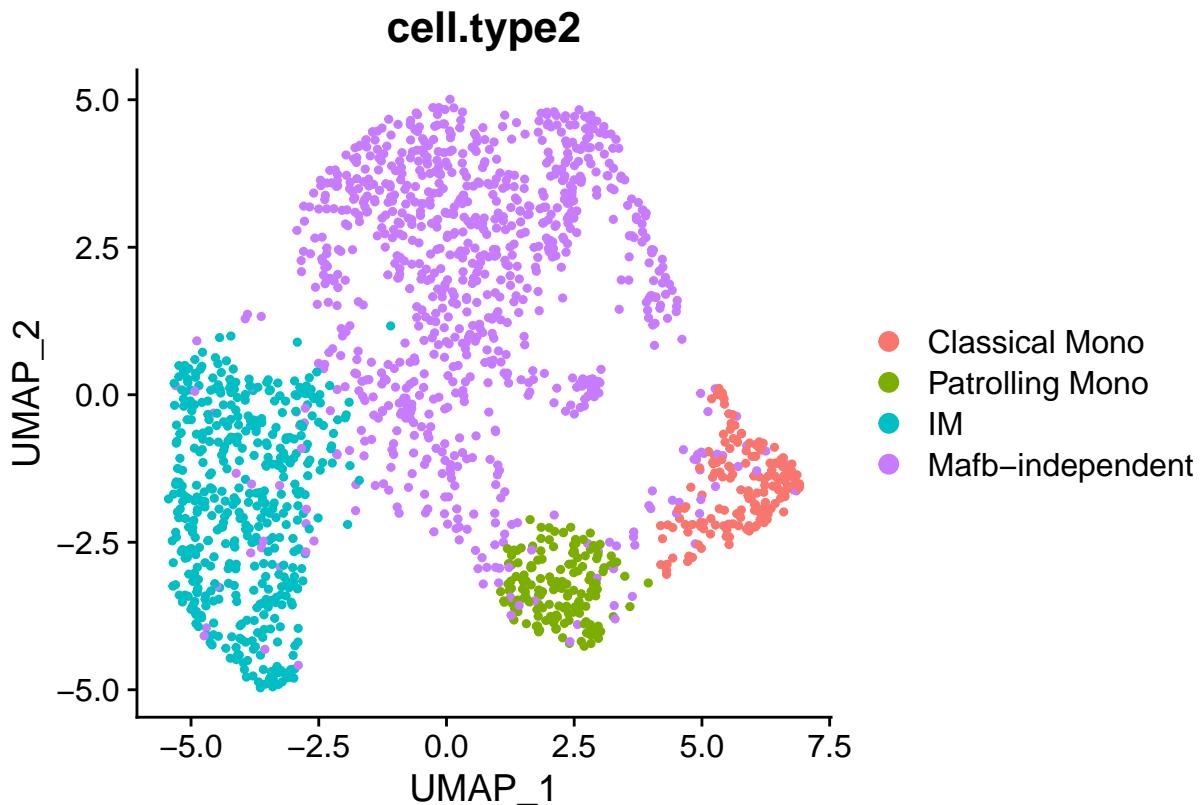
Importing Lung Macrophages

Can be obtained from (GSE193891)

```

Lung_MafbKO <- readRDS("MafbKO_Ctl.seuratObject.Rds")
DimPlot(Lung_MafbKO, group.by = "cell.type2")

```



```

Lung_MafbKO$orig.ident <- Lung_MafbKO$group
Lung_MafbKO <- subset(Lung_MafbKO, cell.type2 %in% c("IM", "Mafb-independent"))

Lung_MafbKO@meta.data <- Lung_MafbKO@meta.data[, c(1:6)]

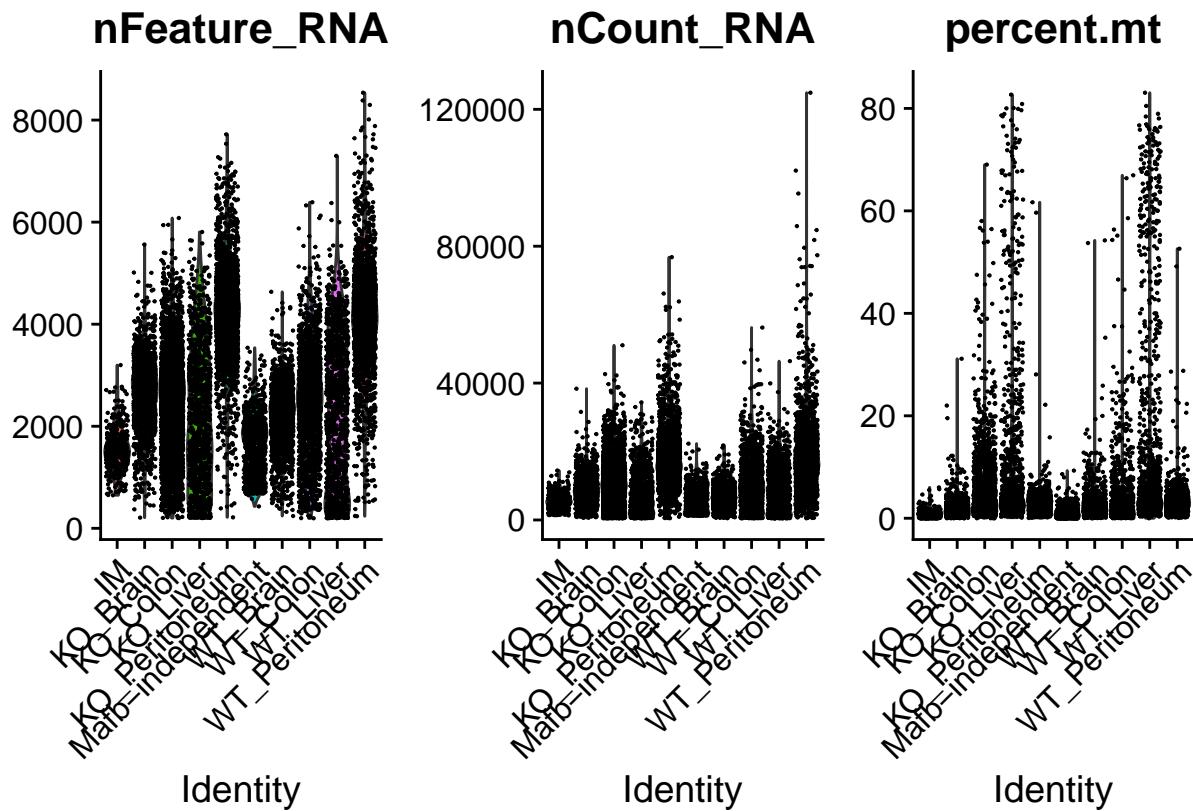
```

Combining Macrophage together

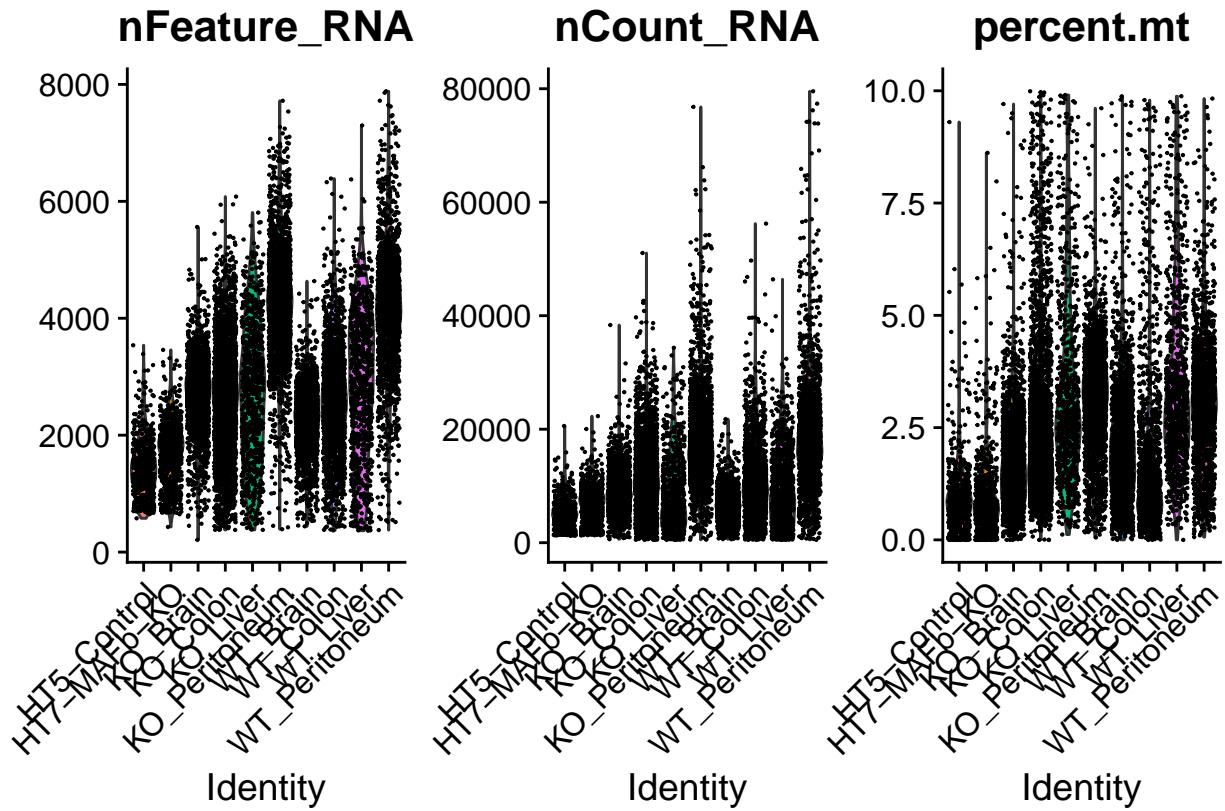
```
Lung_MafbKO <- RenameCells(Lung_MafbKO, new.names = paste0("P1_",
  colnames(Lung_MafbKO)))
Sc_Macro <- merge(Sc_Macro, Lung_MafbKO)
```

Quality control

```
Sc_Macro[["percent.mt"]] <- PercentageFeatureSet(Sc_Macro, pattern = "^\$mt\$")
VlnPlot(Sc_Macro, features = c("nFeature_RNA", "nCount_RNA",
  "percent.mt"), ncol = 3, pt.size = 0.1)
```



```
Sc_Macro <- subset(Sc_Macro, subset = nFeature_RNA > 200 & nFeature_RNA <
  8000 & percent.mt < 10 & nCount_RNA < 80000)
VlnPlot(Sc_Macro, features = c("nFeature_RNA", "nCount_RNA",
  "percent.mt"), group.by = "orig.ident", ncol = 3, pt.size = 0.1)
```



Pre-processing workflow

```

# Normalizing the data
Sc_Macro <- NormalizeData(Sc_Macro, normalization.method = "LogNormalize",
                           scale.factor = 10000)

# Feature selection
Sc_Macro <- FindVariableFeatures(Sc_Macro, selection.method = "vst",
                                   nfeatures = 2000)

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

# plot variable features without labels
plot1 <- VariableFeaturePlot(Sc_Macro)
plot1 <- LabelPoints(plot = plot1, points = top10, repel = TRUE)

# Scaling the data
all.genes <- rownames(Sc_Macro)
Sc_Macro <- ScaleData(Sc_Macro, features = all.genes)

# Linear dimensional reduction
Sc_Macro <- RunPCA(Sc_Macro, features = VariableFeatures(object = Sc_Macro))
plot2 <- DimPlot(Sc_Macro, reduction = "pca")

# Determine the 'dimensionality' of the dataset
Sc_Macro <- JackStraw(Sc_Macro, num.replicate = 100)

```

```

Sc_Macro <- ScoreJackStraw(Sc_Macro, dims = 1:20)
plot3 <- JackStrawPlot(Sc_Macro, dims = 1:20)
plot4 <- ElbowPlot(Sc_Macro, ndims = 30)

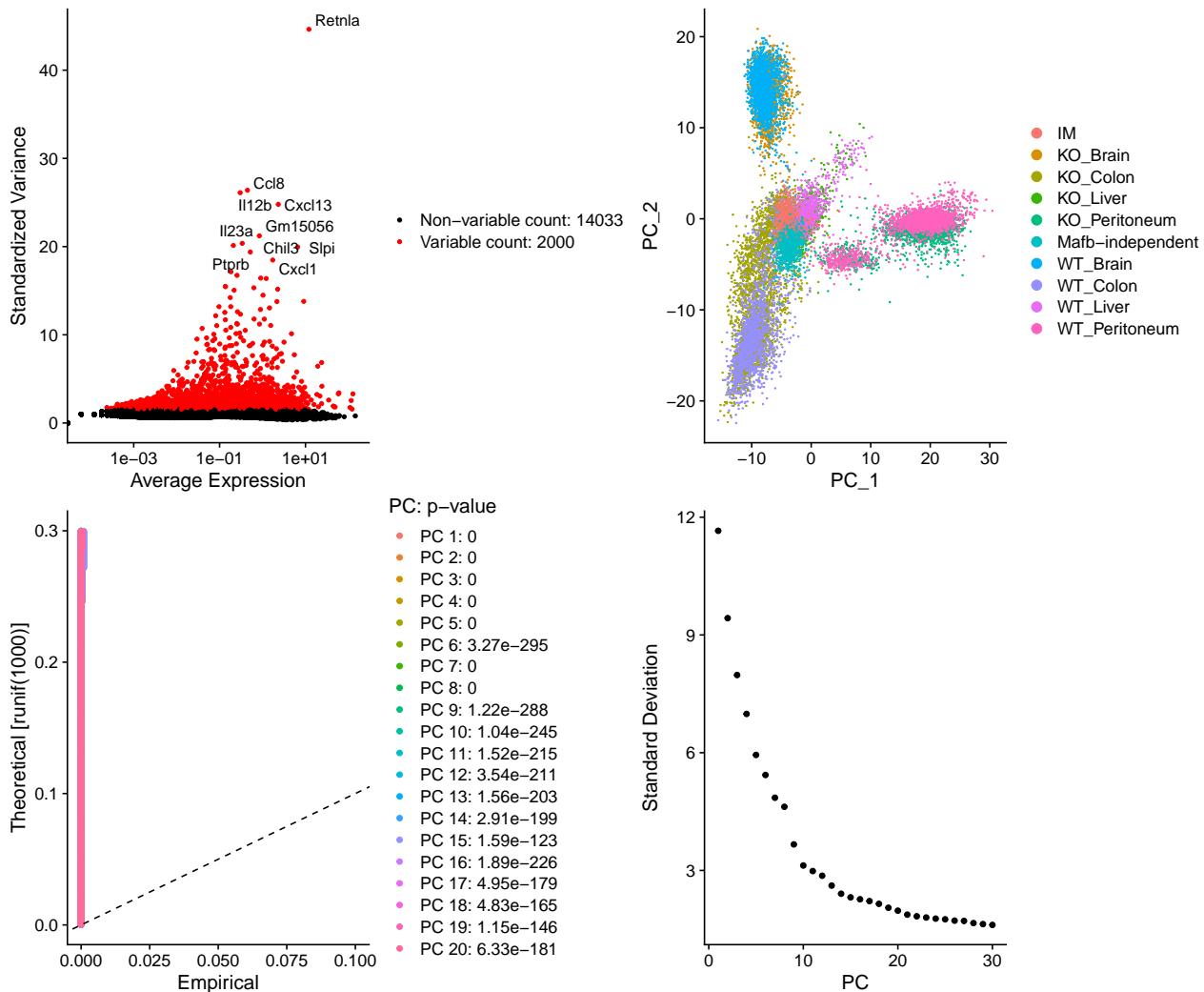
# Cluster cells in umap
Sc_Macro <- FindNeighbors(Sc_Macro, dims = 1:25)
Sc_Macro <- FindClusters(Sc_Macro, resolution = 0.8)
Sc_Macro <- RunUMAP(Sc_Macro, dims = 1:25)

plot1 + plot2 + plot3 + plot4

```

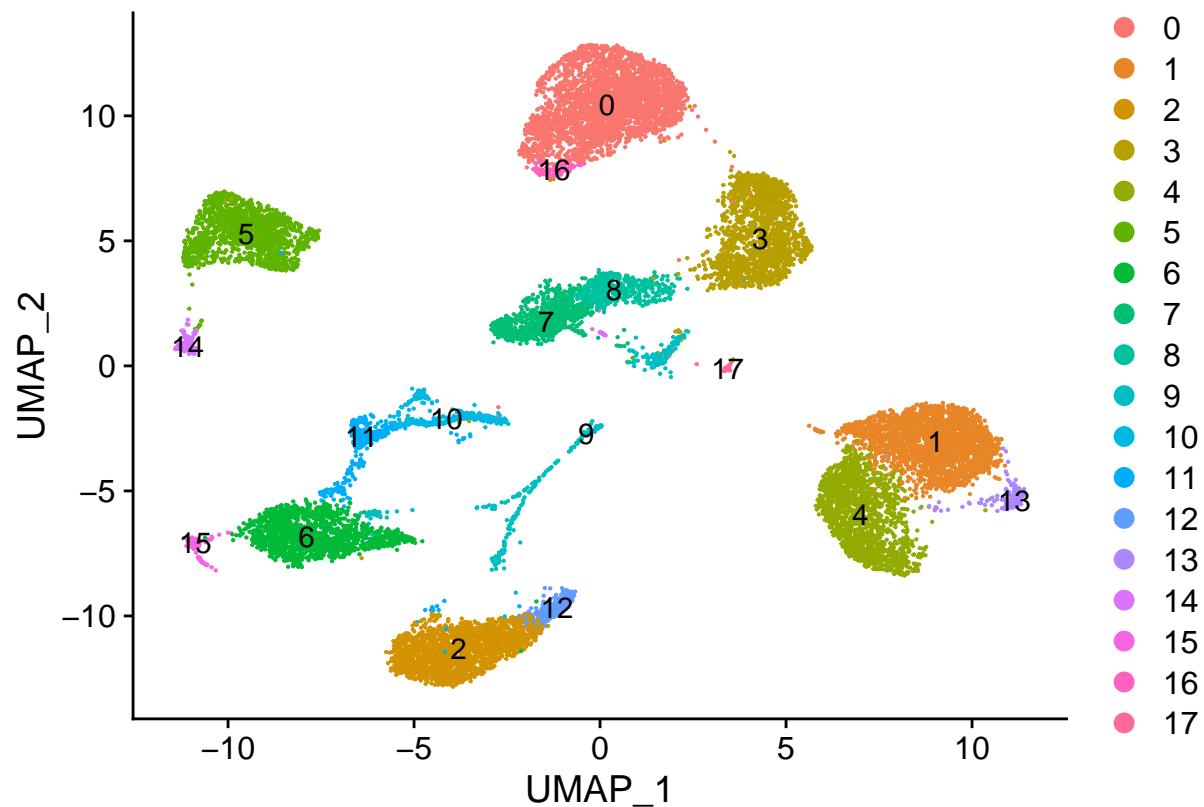
Warning: Transformation introduced infinite values in continuous x-axis

Warning: Removed 28000 rows containing missing values (`geom_point()`).

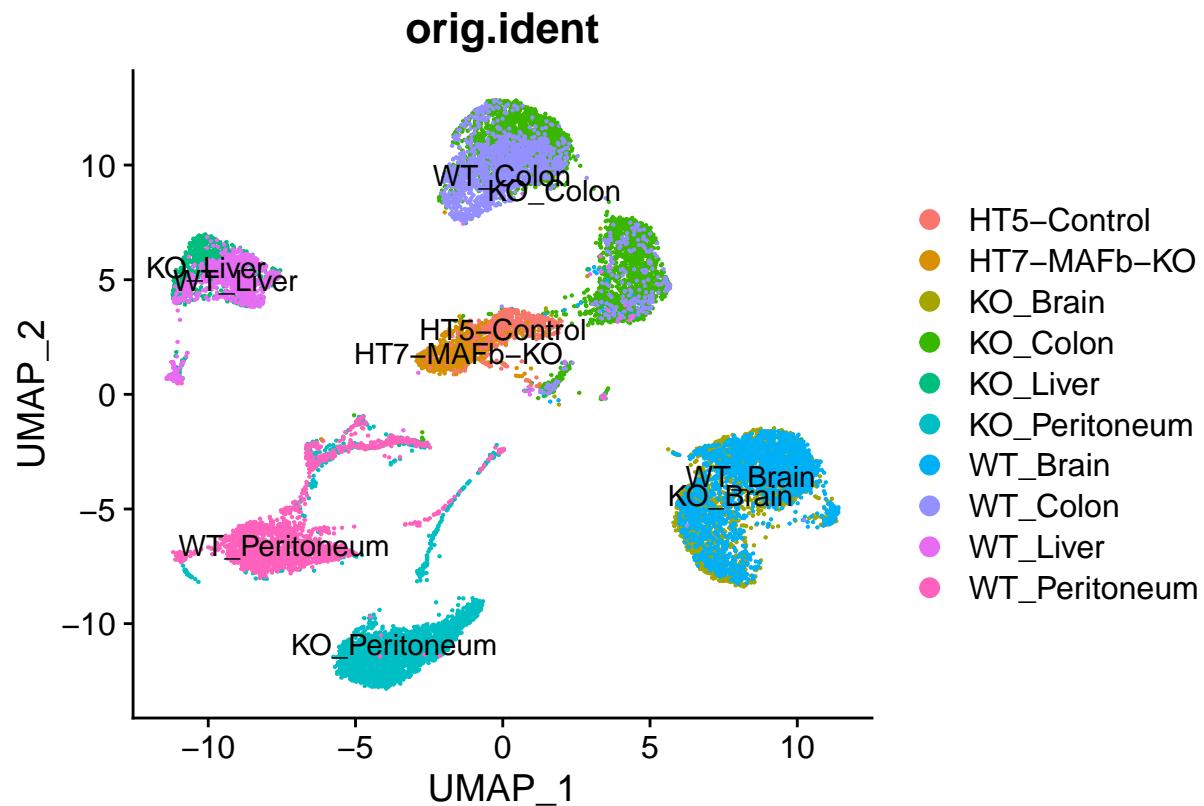


Visualizing all clusters

```
DimPlot(Sc_Macro, reduction = "umap", label = T)
```



```
DimPlot(Sc_Macro, reduction = "umap", label = T, group.by = "orig.ident")
```



Differential expression analysis

```
Macro.markers <- FindAllMarkers(Sc_Macro, only.pos = TRUE, min.pct = 0.25,
  logfc.threshold = 0.25)
Macro.markers %>%
  group_by(cluster) %>%
  slice_max(n = 10, order_by = avg_log2FC)
```

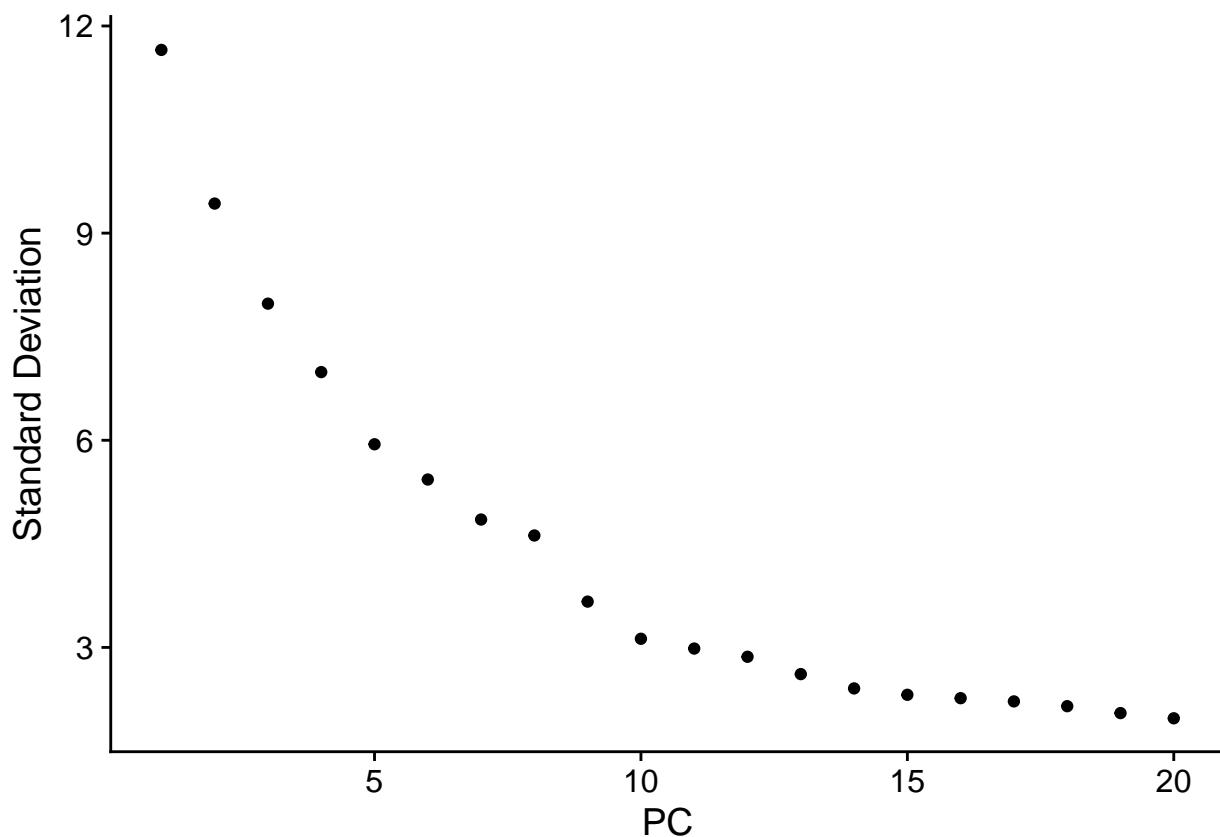
The cluster 9,10,14,15,17 were considered as non relevant and have been removed

- Cluster 9: proliferating cells
- Cluster 10: Dendritic cells
- Cluster 14: endothelial cells
- Cluster 15: unknown
- Cluster 17: unknown

Removing contaminent, based on marker annotation.

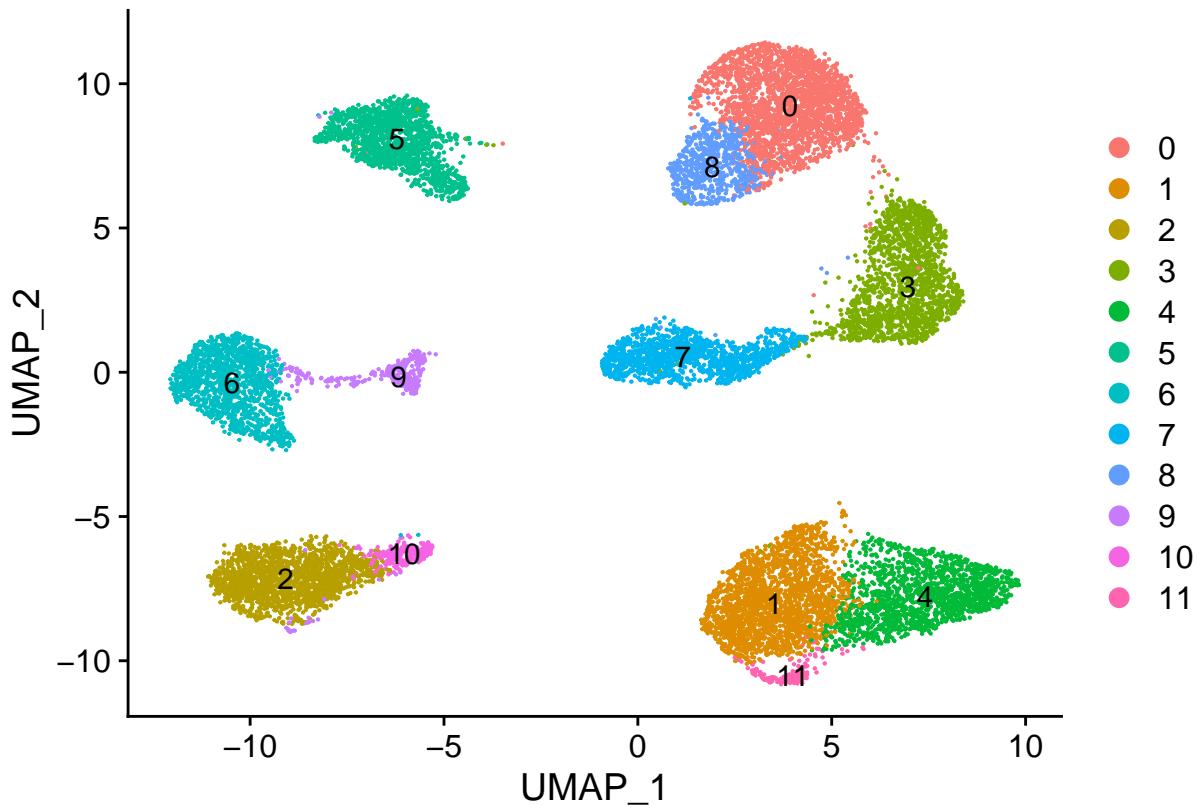
```
Sc_Macro_filter <- subset(Sc_Macro, seurat_clusters %in% c(9,
  10, 14, 15, 17), invert = T)

ElbowPlot(Sc_Macro_filter)
```



```
Sc_Macro_filter <- FindNeighbors(Sc_Macro_filter, dims = 1:15)
Sc_Macro_filter <- FindClusters(Sc_Macro_filter, resolution = 0.7)
Sc_Macro_filter <- RunUMAP(Sc_Macro_filter, dims = 1:15)
```

```
DimPlot(Sc_Macro_filter, reduction = "umap", label = T)
```



Defining subset of interest

```
Sc_Macro_filter$Condition <- Sc_Macro_filter$orig.ident  
  
`%notin%` <- function(x, table) {  
  !(x %in% table)  
}  
  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "KO_Colon" &  
  Sc_Macro_filter$seurat_clusters %in% c(0, 8))] <- "Ly6C+ CM KO"  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "WT_Colon" &  
  Sc_Macro_filter$seurat_clusters %in% c(0, 8))] <- "Ly6C+ CM WT"  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "KO_Colon" &  
  Sc_Macro_filter$seurat_clusters %notin% c(0, 8))] <- "MHC2+ CM KO"  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "WT_Colon" &  
  Sc_Macro_filter$seurat_clusters %notin% c(0, 8))] <- "MHC2+ CM WT"  
  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "KO_Peritoneum" &  
  Sc_Macro_filter$seurat_clusters %in% c(9))] <- "SPM KO"  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "WT_Peritoneum" &  
  Sc_Macro_filter$seurat_clusters %in% c(9))] <- "SPM WT"  
  
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "KO_Peritoneum" &
```

```

Sc_Macro_filter$seurat_clusters %notin% c(9))] <- "LPM KO"
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "WT_Peritoneum" &
  Sc_Macro_filter$seurat_clusters %notin% c(9))] <- "LPM WT"

Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "HT5-Control")] <- "IM WT"
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "HT7-MAFb-KO")] <- "IM KO"

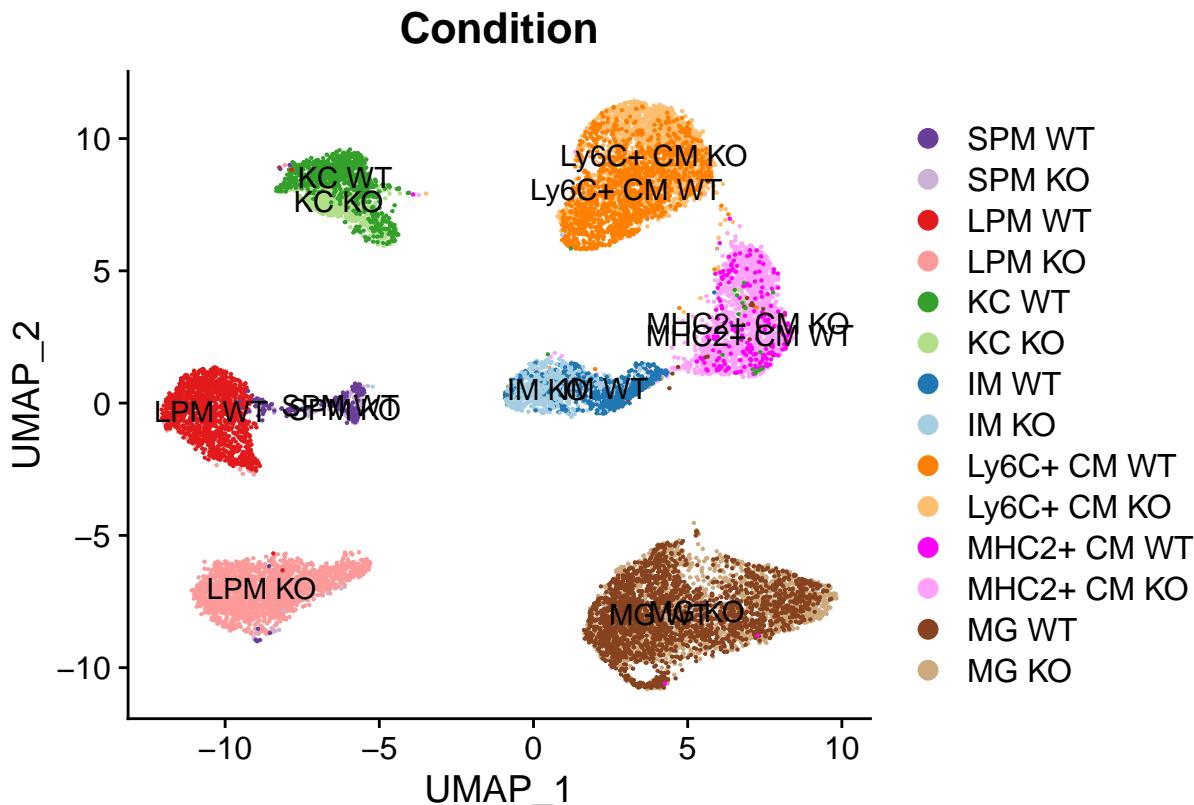
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "WT_Liver")] <- "KC WT"
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "KO_Liver")] <- "KC KO"

Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "WT_Brain")] <- "MG WT"
Sc_Macro_filter$Condition[(Sc_Macro_filter$orig.ident == "KO_Brain")] <- "MG KO"

Sc_Macro_filter$Condition <- factor(Sc_Macro_filter$Condition,
  levels = c("SPM WT", "SPM KO", "LPM WT", "LPM KO", "KC WT",
  "KC KO", "IM WT", "IM KO", "Ly6C+ CM WT", "Ly6C+ CM KO",
  "MHC2+ CM WT", "MHC2+ CM KO", "MG WT", "MG KO"))

pal <- c("#6A3D9A", "#CAB2D6", "#E31A1C", "#FB9A99", "#33A02C",
  "#B2DF8A", "#1F78B4", "#A6CEE3", "#FF7FO0", "#FDBF6F", "#FF00FA",
  "#FFA1FD", "#87421F", "#CDAA7D")
DimPlot(Sc_Macro_filter, group.by = "Condition", label = T, cols = pal)

```



saving data

```
saveRDS(Sc_Macro_filter, "Sc_Macro_filter.rds")
```

```

sessionInfo()

## R version 4.3.3 (2024-02-29)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 22.04.4 LTS
##
## Matrix products: default
## BLAS:    /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK:  /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblas-r0.3.20.so;  LAPACK version 3.10.0
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
## [3] LC_TIME=fr_BE.UTF-8          LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=fr_BE.UTF-8      LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=fr_BE.UTF-8         LC_NAME=C
## [9] LC_ADDRESS=C                 LC_TELEPHONE=C
## [11] LC_MEASUREMENT=fr_BE.UTF-8   LC_IDENTIFICATION=C
##
## time zone: Europe/Brussels
## tzcode source: system (glibc)
##
## attached base packages:
## [1] stats4     stats      graphics   grDevices  utils      datasets   methods
## [8] base
##
## other attached packages:
## [1] formatR_1.14           RColorBrewer_1.1-3
## [3] scRNAseq_2.14.0        scuttle_1.10.1
## [5] SingleCellExperiment_1.22.0 SingleR_2.2.0
## [7] SummarizedExperiment_1.30.2 Biobase_2.60.0
## [9] GenomicRanges_1.52.0    GenomeInfoDb_1.36.0
## [11] IRanges_2.34.0          S4Vectors_0.38.1
## [13] BiocGenerics_0.46.0    MatrixGenerics_1.12.2
## [15] matrixStats_1.0.0      stringr_1.5.0
## [17] ggplot2_3.4.2          patchwork_1.1.2
## [19] SeuratObject_4.1.3     Seurat_4.3.0
## [21] dplyr_1.1.2
##
## loaded via a namespace (and not attached):
## [1] RcppAnnoy_0.0.21       splines_4.3.3
## [3] later_1.3.1            BiocIO_1.10.0
## [5] filelock_1.0.2          bitops_1.0-7
## [7] R.oo_1.25.0             tibble_3.2.1
## [9] polyclip_1.10-4         XML_3.99-0.14
## [11] lifecycle_1.0.3         ensemblldb_2.24.0
## [13] globals_0.16.2          lattice_0.22-5
## [15] MASS_7.3-60.0.1         magrittr_2.0.3
## [17] plotly_4.10.2           rmarkdown_2.23
## [19] yaml_2.3.7              httpuv_1.6.11
## [21] sctransform_0.3.5       spam_2.9-1
## [23] sp_2.0-0                spatstat.sparse_3.0-2
## [25] reticulate_1.30         cowplot_1.1.1
## [27] pbapply_1.7-2           DBI_1.1.3
## [29] abind_1.4-5             zlibbioc_1.46.0

```

```

## [31] Rtsne_0.16
## [33] purrr_1.0.1
## [35] RCurl_1.98-1.12
## [37] GenomeInfoDbData_1.2.10
## [39] irlba_2.3.5.1
## [41] spatstat.utils_3.0-3
## [43] spatstat.random_3.1-5
## [45] parallelly_1.36.0
## [47] leiden_0.4.3
## [49] DelayedArray_0.26.3
## [51] tidyselect_1.2.0
## [53] ScaledMatrix_1.8.1
## [55] spatstat.explore_3.2-1
## [57] jsonlite_1.8.7
## [59] progressr_0.13.0
## [61] survival_3.5-8
## [63] tools_4.3.3
## [65] Rcpp_1.0.11
## [67] gridExtra_2.3
## [69] withr_2.5.0
## [71] fastmap_1.1.1
## [73] digest_0.6.33
## [75] R6_2.5.1
## [77] colorspace_2.1-0
## [79] tensor_1.5
## [81] spatstat.data_3.0-1
## [83] R.methodsS3_1.8.2
## [85] tidyR_1.3.0
## [87] data.table_1.14.8
## [89] prettyunits_1.1.1
## [91] htmlwidgets_1.6.2
## [93] uwot_0.1.16
## [95] gtable_0.3.3
## [97] lmtest_0.9-40
## [99] htmltools_0.5.5
## [101] ProtGenerics_1.32.0
## [103] png_0.1-8
## [105] rstudioapi_0.14
## [107] reshape2_1.4.4
## [109] nlme_3.1-164
## [111] cachem_1.0.8
## [113] KernSmooth_2.23-22
## [115] parallel_4.3.3
## [117] AnnotationDbi_1.62.1
## [119] restfulr_0.0.15
## [121] grid_4.3.3
## [123] RANN_2.6.1
## [125] BiocSingular_1.16.0
## [127] beachmat_2.16.0
## [129] cluster_2.1.6
## [131] evaluate_0.21
## [133] Rsamtools_2.16.0
## [135] compiler_4.3.3
## [137] crayon_1.5.2

R.utils_2.12.2
AnnotationFilter_1.24.0
rapdirs_0.3.3
ggrepel_0.9.3
listenv_0.9.0
goftest_1.2-3
fitdistrplus_1.1-11
DelayedMatrixStats_1.22.1
codetools_0.2-19
xml2_1.3.4
farver_2.1.1
BiocFileCache_2.8.0
GenomicAlignments_1.36.0
ellipsis_0.3.2
ggridges_0.5.4
progress_1.2.2
ica_1.0-3
glue_1.6.2
xfun_0.39
BiocManager_1.30.21
fansi_1.0.4
rsvd_1.0.5
mime_0.12
scattermore_1.2
biomaRt_2.56.1
RSQLite_2.3.1
utf8_1.2.3
generics_0.1.3
rtracklayer_1.60.0
httr_1.4.6
S4Arrays_1.2.1
pkgconfig_2.0.3
blob_1.2.4
XVector_0.40.0
dotCall64_1.0-2
scales_1.2.1
knitr_1.43
rjson_0.2.21
curl_5.0.1
zoo_1.8-12
BiocVersion_3.17.1
vigor_0.4.5
miniUI_0.1.1.1
ggrastr_1.0.2
pillar_1.9.0
vctrs_0.6.3
promises_1.2.0.1
dbplyr_2.3.2
xtable_1.8-4
beeswarm_0.4.0
GenomicFeatures_1.52.0
cli_3.6.1
rlang_1.1.1
future.apply_1.11.0

```

```
## [139] labeling_0.4.2          ggbeeswarm_0.7.2
## [141] plyr_1.8.8                stringi_1.7.12
## [143] viridisLite_0.4.2         deldir_1.0-9
## [145] BiocParallel_1.34.2       munsell_0.5.0
## [147] Biostrings_2.68.1         lazyeval_0.2.2
## [149] spatstat.geom_3.2-4       Matrix_1.6-1
## [151] ExperimentHub_2.8.0       hms_1.1.3
## [153] sparseMatrixStats_1.12.0   bit64_4.0.5
## [155] future_1.33.0             KEGGREST_1.40.0
## [157] shiny_1.7.4.1              highr_0.10
## [159] interactiveDisplayBase_1.38.0 AnnotationHub_3.8.0
## [161] ROCR_1.0-11                igraph_1.5.0.1
## [163] memoise_2.0.1              bit_4.0.5
```