

# PAPER TITLE TO BE DEFINED (in common.yaml)

4-Compare Refilled IMs in Day4 depletion to control

2021-11-22 11:12:30 +0100

## Abstract

Lung interstitium macrophages (IMs) are non-alveolar resident tissue macrophages which contribute to the lung homeostasis. These cells were reported to be heterogeneous by our group and other teams, which contains two main distinct subpopulations: CD206+ IMs and CD206- IMs. However, the exact origin of IMs and the transcriptional programs that control IM differentiation remains unclear. In recent report, we analyzed the refilled IMs in the course of time after induced IM depletion with single-cell RNA sequencing (10X Genomics Chromium) and bulk RNA sequencing.

## Contents

|   |           |
|---|-----------|
| <b>1 Description</b>  | <b>2</b>  |
| <b>2 Load data and packages</b>                             | <b>2</b>  |
| <b>3 Build up sample metadata</b>                           | <b>2</b>  |
| <b>4 Clustering cells</b>                                   | <b>4</b>  |
| <b>5 Identify cells with SingleR</b>                        | <b>5</b>  |
| <b>6 Identify cells with marker expression</b>              | <b>6</b>  |
| <b>7 Combine SingleR and marker expression results</b>      | <b>7</b>  |
| <b>8 Remove other cell types than monocytes/macrophages</b> | <b>11</b> |
| <b>9 Phenotyping of lung monocytes/IMs</b>                  | <b>11</b> |
| 9.1 Clean data after filtering . . . . .                    | 11        |
| 9.2 Data re-processing . . . . .                            | 11        |
| 9.3 Population characterization . . . . .                   | 16        |
| 9.4 Cell-cycle analysis . . . . .                           | 18        |
| <b>10 Session information</b>                               | <b>21</b> |
| <b>11 References</b>  | <b>24</b> |

## 1 Description

## 2 Load data and packages

```
library(Seurat)
source("../R/seurat.make.integrated.R")

initiation.analysis.folder <- "../3-scrNAseq_initiation_the_IMs_in_Day4_
depletion/"
file.names <- list.files(initiation.analysis.folder, pattern = "*.rds")
sample.names <- sub(".rds", "", file.names)

table.samples <- data.frame(sample.names, file.names)

i <- sapply(table.samples, is.factor)
table.samples[i] <- lapply(table.samples[i], as.character) # this is for
convert the factor to character.

rm("file.names", "sample.names") # remove individual vector to avoid
confusion.

for (i in 1:length(table.samples$sample.names)) {
  assign(table.samples$sample.names[i], readRDS(file = file.path(
    initiation.analysis.folder, table.samples$file.names[i])))
}
```

## 3 Build up sample metadata

Here's sample order:

```
table.samples$sample.names
```

```
## [1] "CPlus_NGS20_Q147.seuratObject"      "Plusplus_NGS20_Q148.
seuratObject"
```

Make metadata table

```
# sample.names
sample.metadata <- data.frame(
  group = c("Cre+", "++"),
  cell.type = rep("lung_IM_niche_(including_
CD11c)", length(table.samples$sample.
names)),
  origin = rep("lung", length(table.samples$.
sample.names)),
  sample_prefix = gsub("_NGS[a-zA-Z0-9_.-]*",
 "", table.samples$sample.names))
rownames(sample.metadata) <- table.samples$sample.names
sample.metadata
```

```
## # A tibble: 2 x 4
##   group cell.type          origin sample_prefix

```

|   |               |   |
|---|---------------|---|
| ## <chr> <chr>                            | <chr> <chr>   | 3 |
| ## 1 Cre+ lung IM niche (including CD11c) | lung CPlus    | 4 |
| ## 2 ++ lung IM niche (including CD11c)   | lung Plusplus | 5 |

```

# add sample information:
for (i in table.samples$sample.names) {
  obj <- get(i)
  obj$treatment <- sample.metadata[i, ]$group
  assign(i, obj)
}

for (i in table.samples$sample.names) {
  obj <- get(i)
  obj$therapy <- sample.metadata[i, ]$cell.type
  assign(i, obj)
}

for (i in table.samples$sample.names) {
  obj <- get(i)
  obj$therapy <- sample.metadata[i, ]$origin
  assign(i, obj)
}

# add pre-fix to cellnames:
for (i in table.samples$sample.names) {
  obj <- get(i)
  obj <- RenameCells(obj, add.cell.id = sample.metadata[i, ]$sample_prefix
    )
  assign(i, obj)
}

```

Integrate:

```

results.dim30 <- seurat.make.integrated(seuratObjects = sapply(table.
  samples$sample.names, get),
                                         prefix = table.
                                         samples$sample.
                                         names,
                                         dimensionality =
                                         1:30,
                                         SCTransf = FALSE)

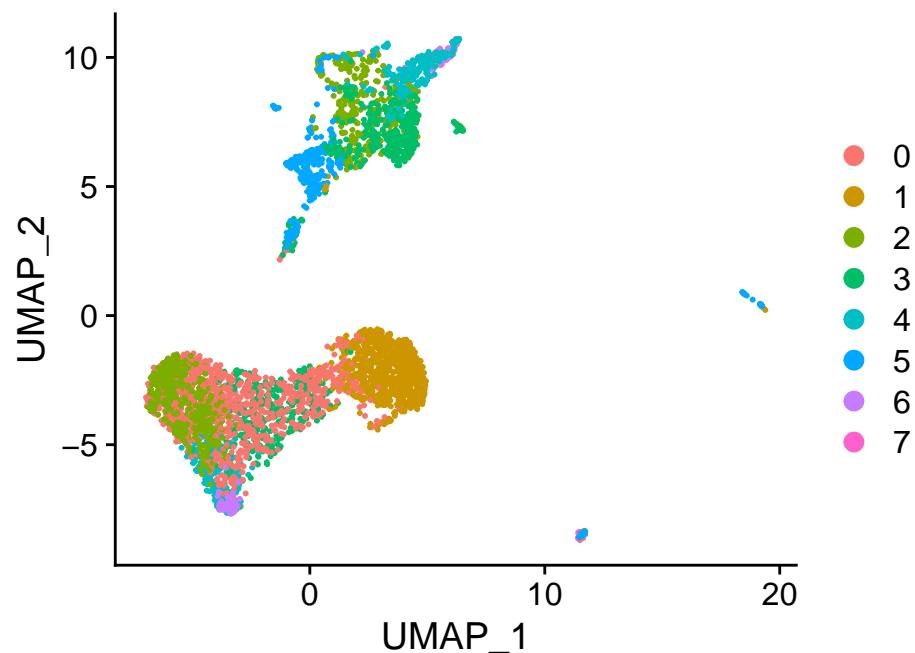
```

```

results.dim30$plots$plot.umap + ggtitle("dim_1:30")

```

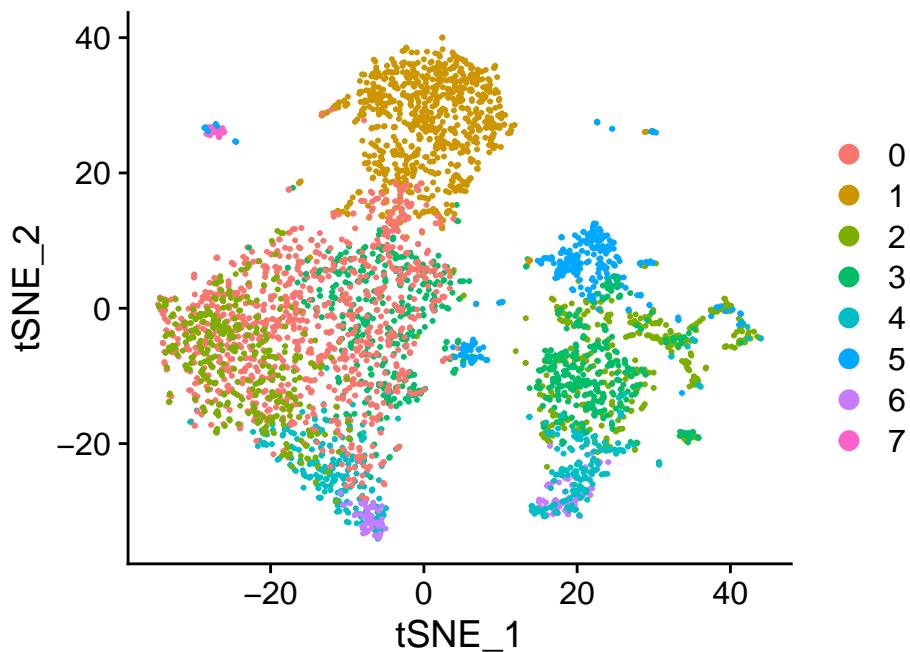
**dim = 1:30**



```
results.dim30$plots$plot.tsne + ggtitle("dim = 1:30")
```

1

**dim = 1:30**



```
results <- results.dim30$integrated.seuratObject
```

1

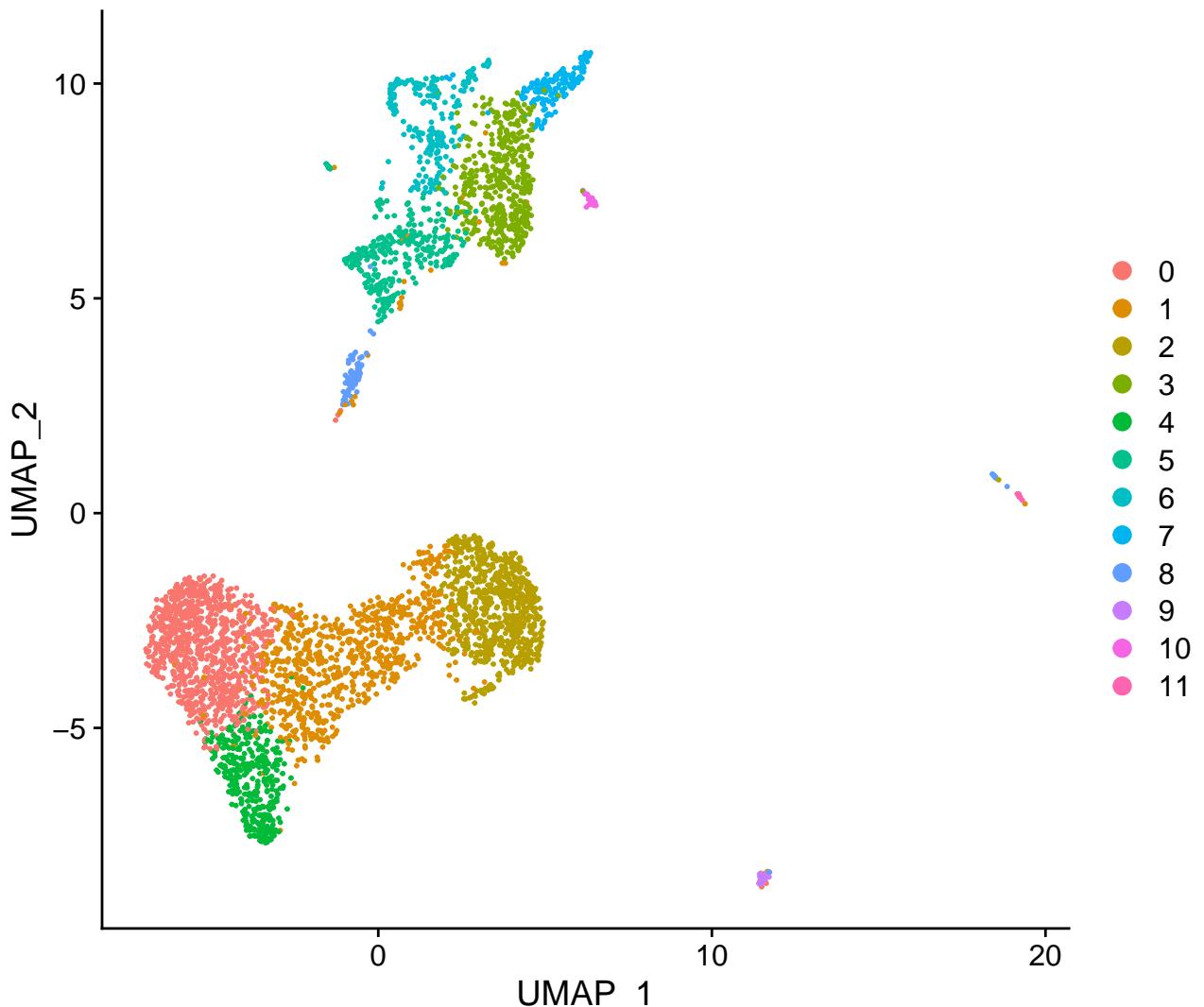
## 4 Clustering cells

```
results <- FindClusters(results, resolution = 0.5)
```

1

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck 1
## 2
## Number of nodes: 4169 3
## Number of edges: 161254 4
## 5
## Running Louvain algorithm... 6
## Maximum modularity in 10 random starts: 0.8723 7
## Number of communities: 12 8
## Elapsed time: 0 seconds 9
```

```
DimPlot(results) 1
```



```
# save data
saveRDS(results, file = "./results.integrated.Rds") 1
2
```

## 5 Identify cells with SingleR

```

source("~/Desktop/velocyto/Script/seurat2singleR.R")
Idents(results) <- "integrated_snn_res.0.5"
results.singleR <- seurat2singleR(results, ref = "ImmGenData")

```

```

saveRDS(results.singleR, file = "./results.ImmGenData.singleR.Rds")

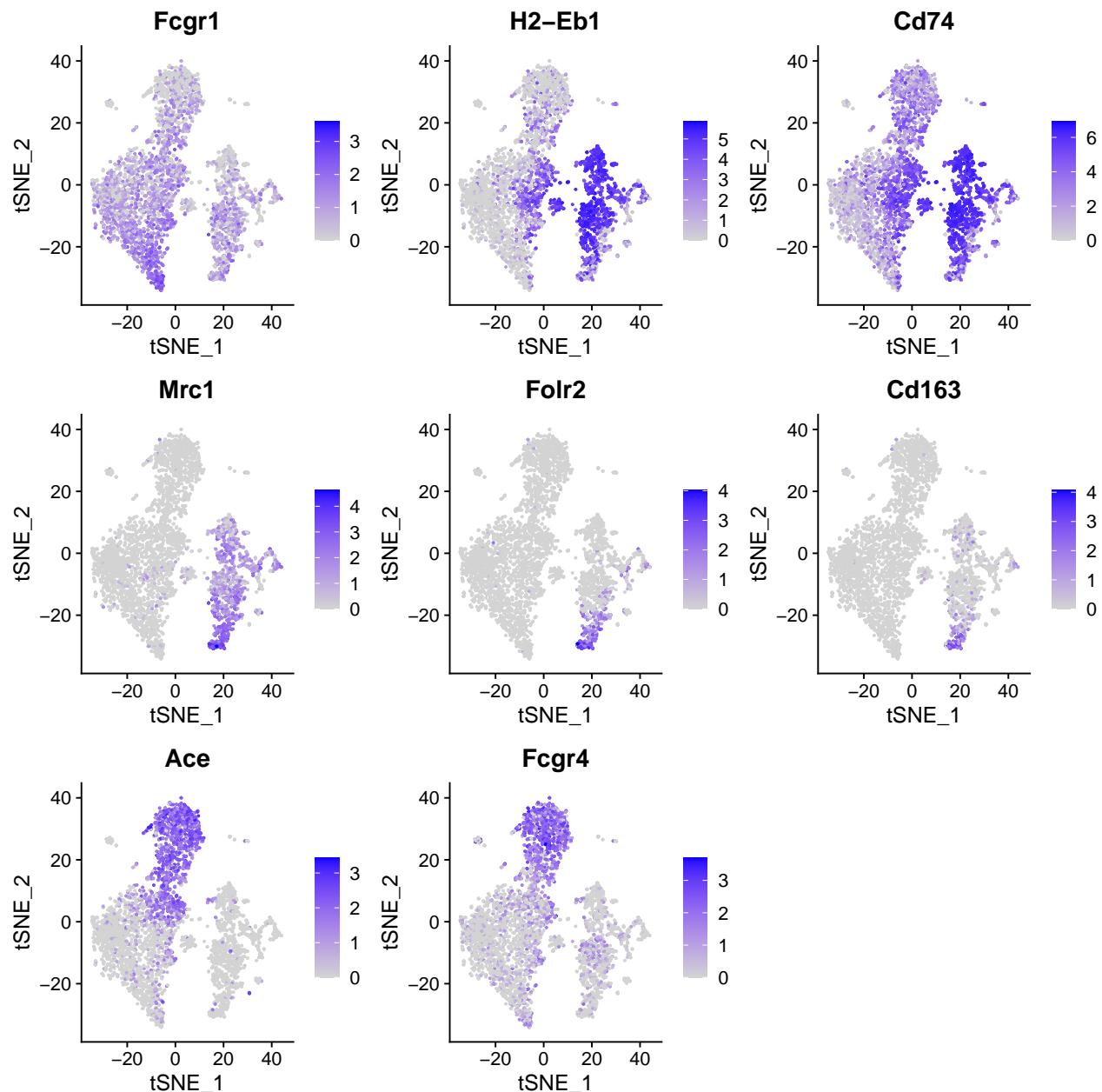
```

## 6 Identify cells with marker expression

```

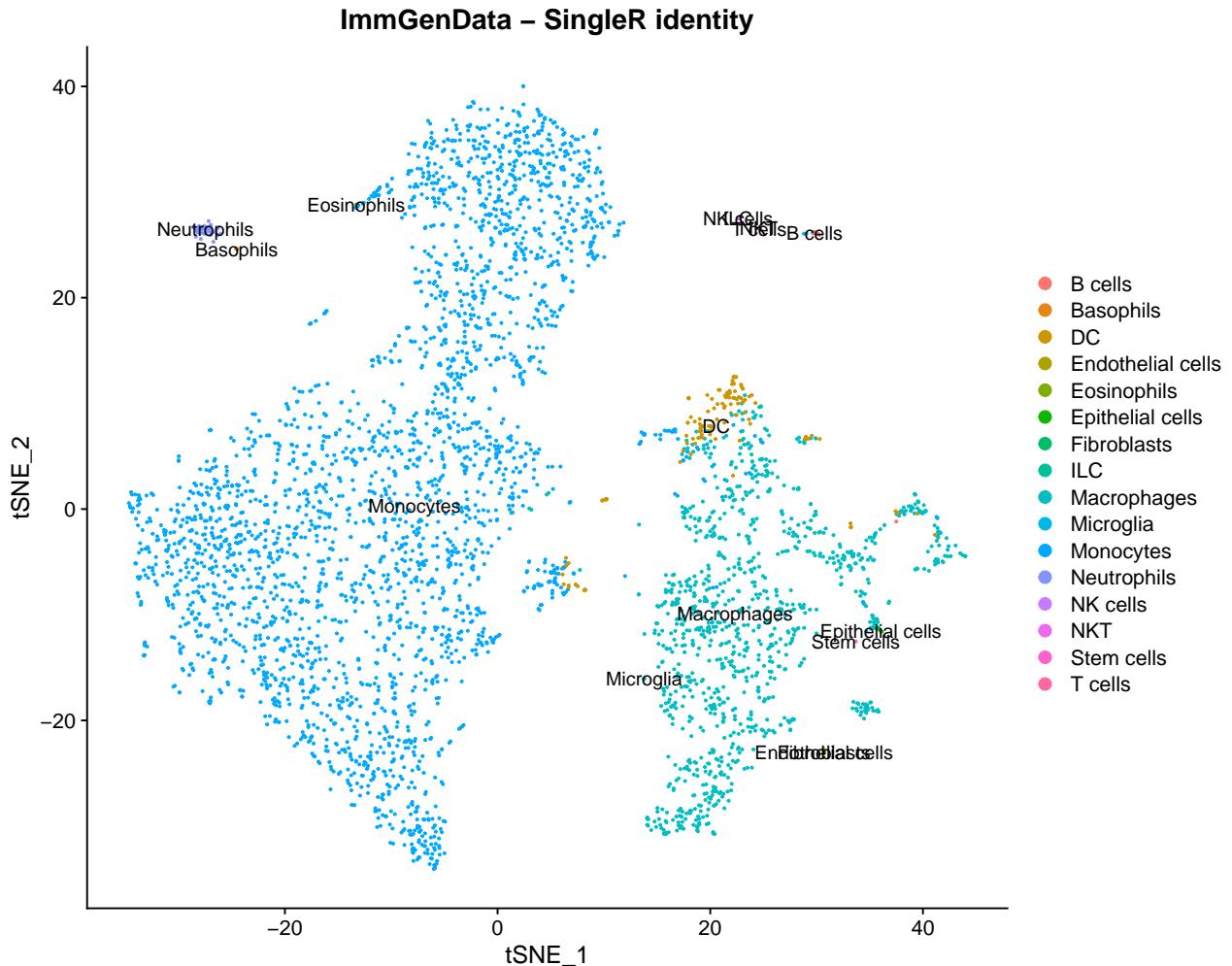
DefaultAssay(results) <- "RNA"
FeaturePlot(results, features = c("Fcgr1", "H2-Eb1", "Cd74", "Mrc1", "Folr2", "Cd163", "Ace", "Fcgr4"), reduction = "tsne")

```



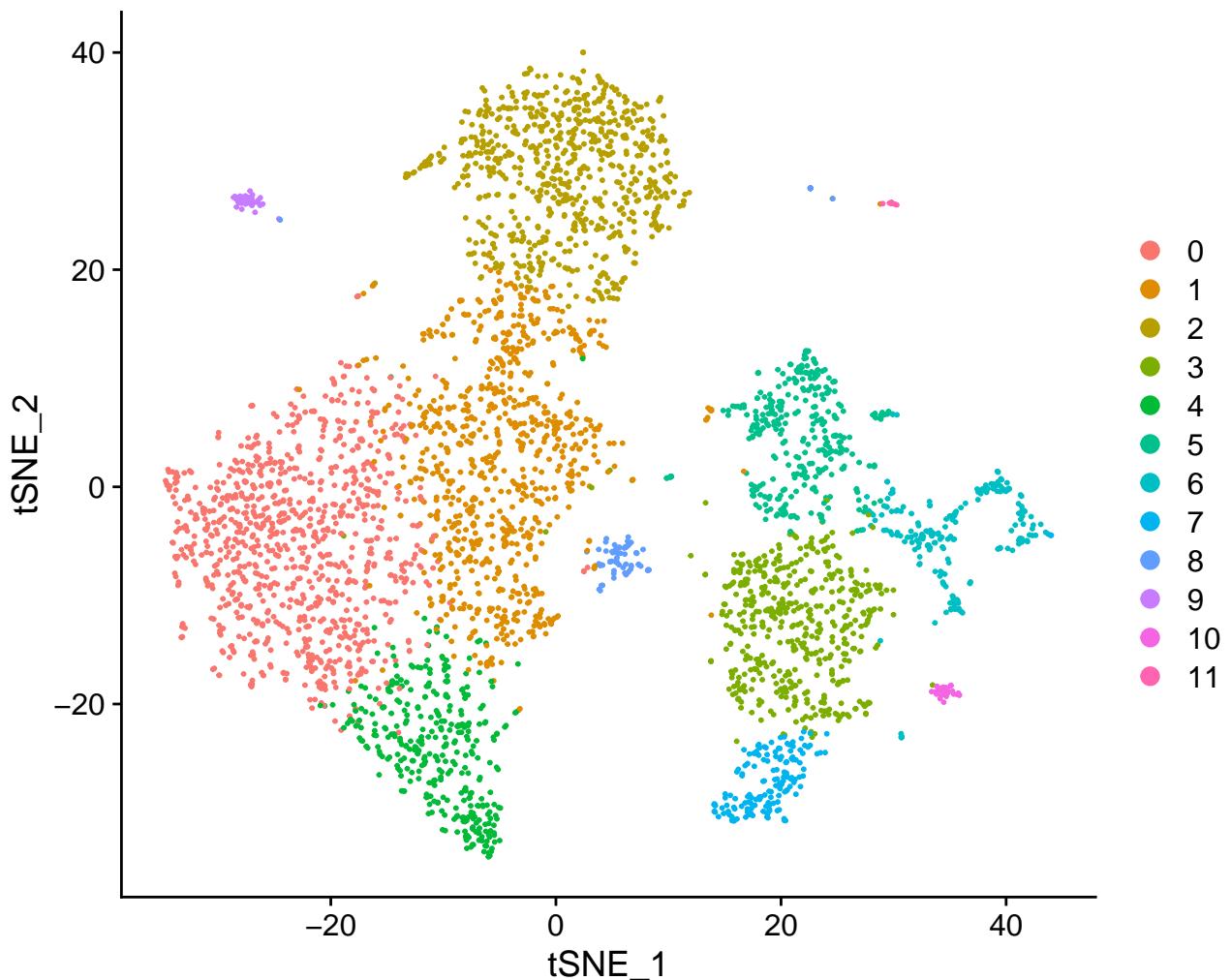
## 7 Combine SingleR and marker expression results

```
results$singleR.celltype <- results.singleR$labels  
DimPlot(results, group.by = "singleR.celltype", reduction = "tsne", label  
= T) + ggtitle("ImmGenData - SingleR identity")
```

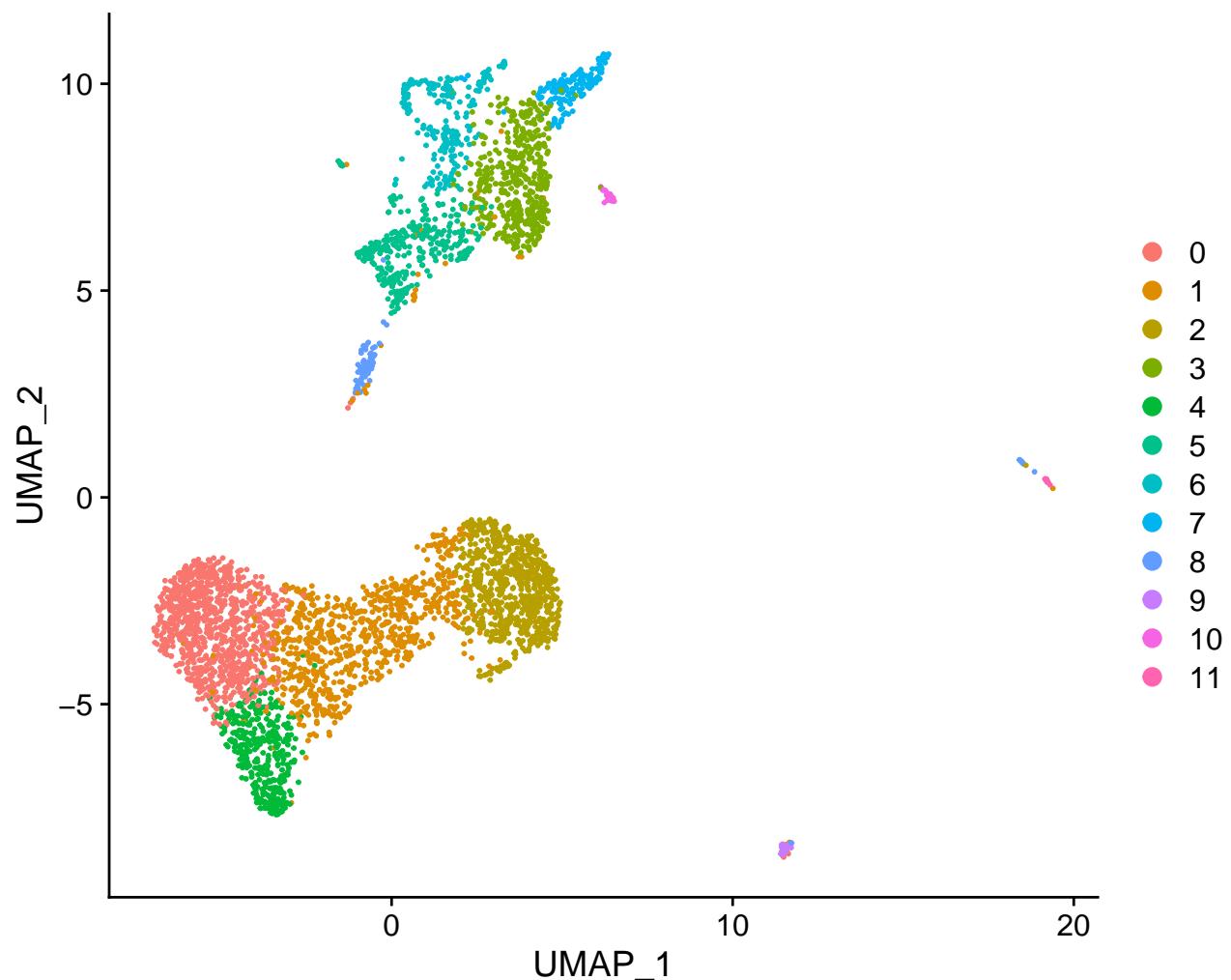


```
DimPlot(results, reduction = "tsne") + ggtitle("results - Clusters")
```

## results – Clusters

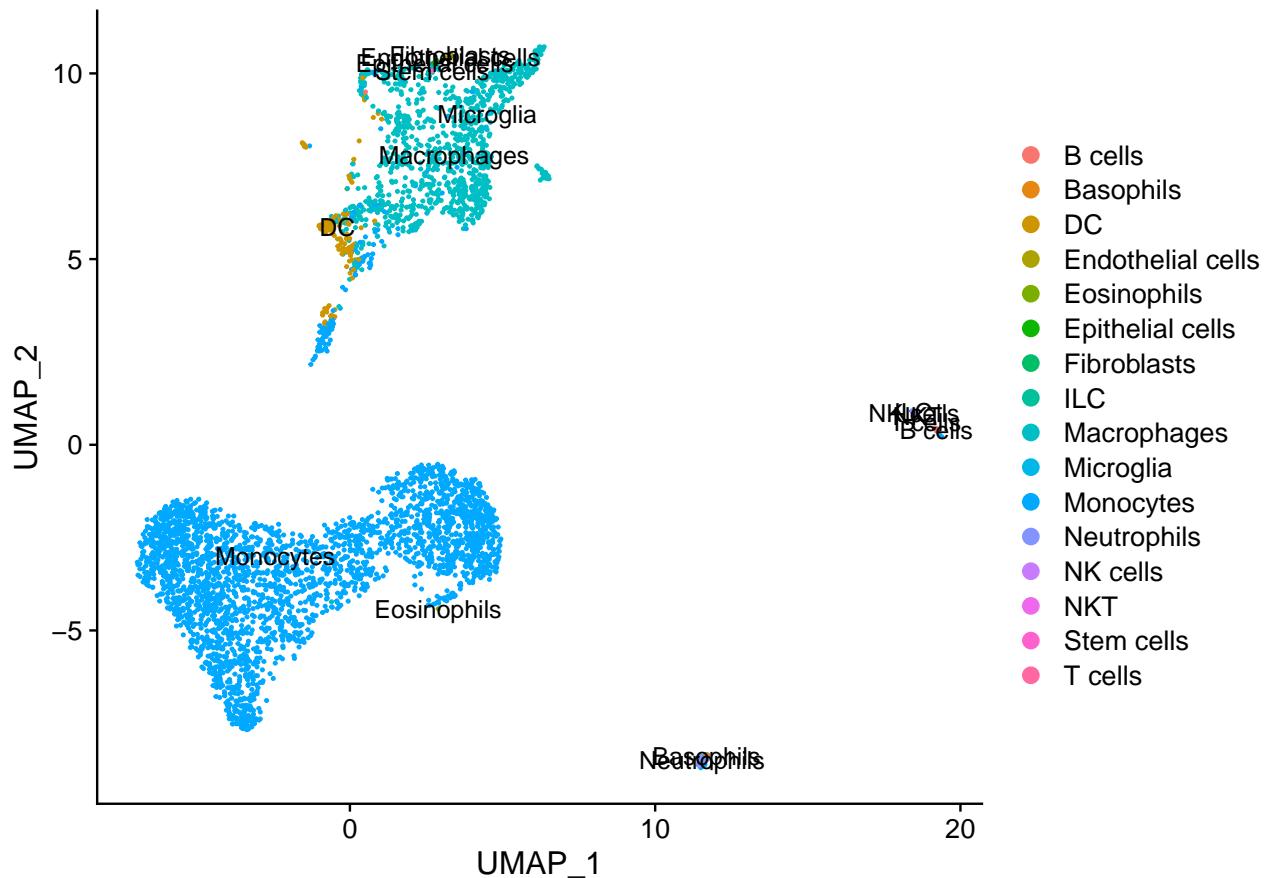


## results – Clusters

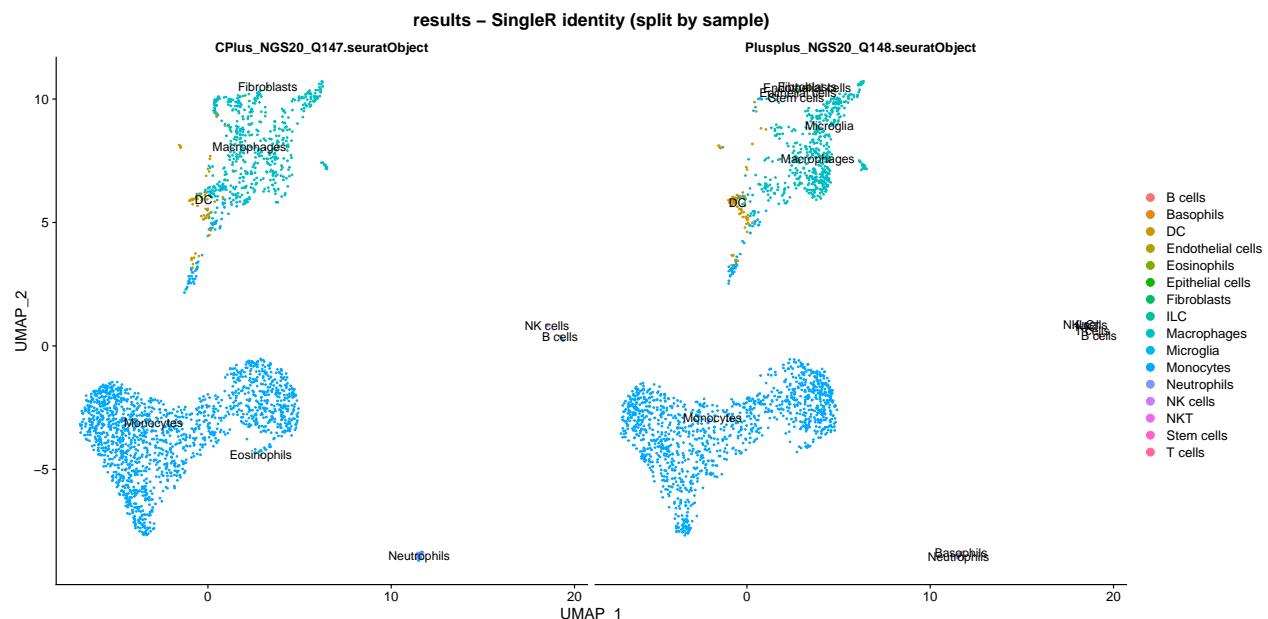


```
DimPlot(results, group.by = "singleR.celltype", reduction = "umap", label = T) + ggtitle("results - SingleR identity")
```

## results – SingleR identity



```
DimPlot(results, group.by = "singleR.celltype", reduction = "umap", label = T, split.by = "object_before_integrated", ncol = 2) + ggtitle("results - SingleR identity (split by sample)")
```



## 8 Remove other cell types than monocytes/macrophages

Cell numbers per cell type:

|                                  |   |
|----------------------------------|---|
| table(results\$singleR.celltype) | 1 |
| ##                               | 1 |
| ## B cells                       | 2 |
| ## 16                            | 3 |
| ## Eosinophils                   | 2 |
| ## 1                             | 3 |
| ## Macrophages                   | 4 |
| ## 1034                          | 5 |
| ## NK cells                      | 6 |
| ## 7                             | 7 |
| ## Basophils                     | 8 |
| ## Epithelial cells              | 9 |
| ## Fibroblasts                   | 1 |
| ## Microglia                     | 2 |
| ## Monocytes                     | 3 |
| ## NKT                           | 4 |
| ## Stem cells                    | 5 |
| ## T cells                       | 6 |
| ##                               | 7 |
| ## DC                            | 8 |
| ## Endothelial cells             | 9 |
| ## ILC                           | 1 |
| ## Neutrophils                   | 2 |
| ## 2921                          | 3 |
| ## 42                            | 4 |
| ## 1                             | 5 |

Keep only “Monocytes”, “Macrophages” and “Microglia”. As microglia are highly similar to lung IMs and we can exclude the possibility of contamination of brain cells in this experiment, we should keep them for the following analysis.

```
results <- results[, WhichCells(results, expression = singleR.celltype %in%
  % c("Monocytes", "Macrophages", "Microglia"))]
```

## 9 Phenotyping of lung monocytes/IMs

### 9.1 Clean data after filtering

Remove old snn:

```
results$RNA_snn_res.0.5 <- NULL
results$integrated_snn_res.0.5 <- NULL
```

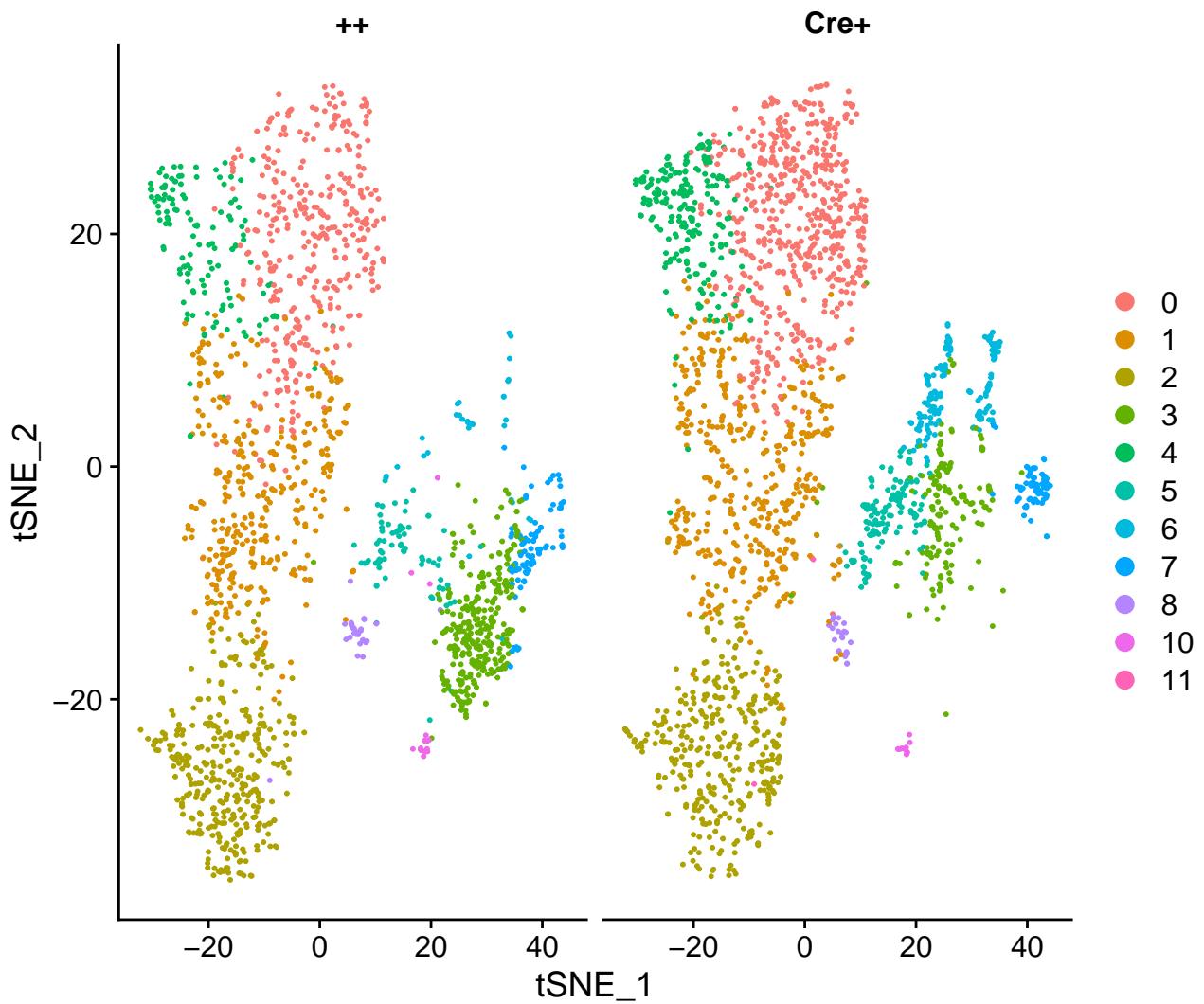
Save data

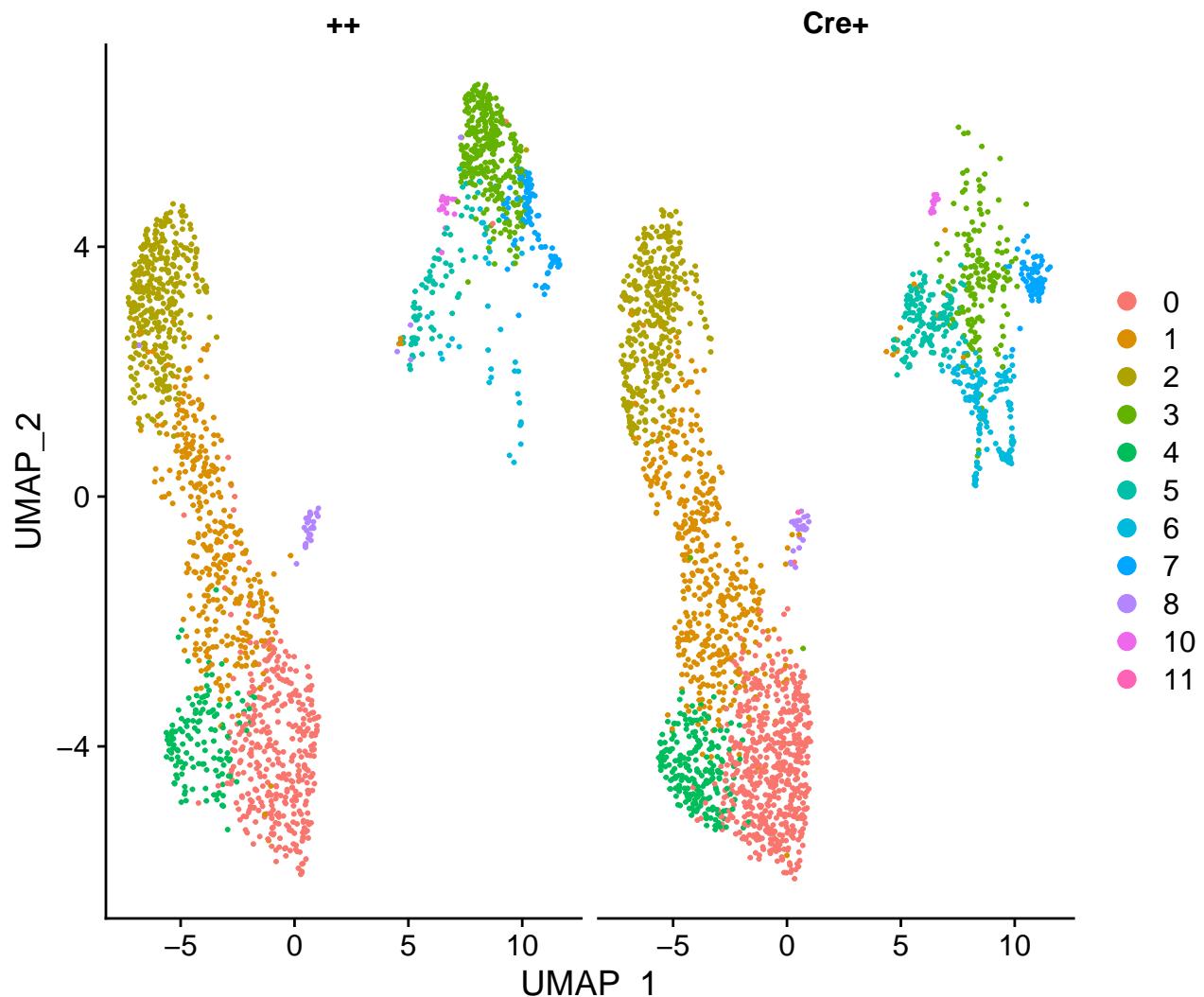
```
saveRDS(results, file = "./results.cellType_filtered.seuratObject.Rds")
```

### 9.2 Data re-processing

```
DefaultAssay(results) <- "RNA"
results <- NormalizeData(results)
results <- FindVariableFeatures(results, selection.method = "vst",
  nfeatures = 2000)
results <- ScaleData(results, features = rownames(results))
results <- RunPCA(results, features = VariableFeatures(results))
results <- RunTSNE(results, dims = 1:20)
results <- RunUMAP(results, dims = 1:20)

DimPlot(results, reduction = "tsne", split.by = "treatment")
```





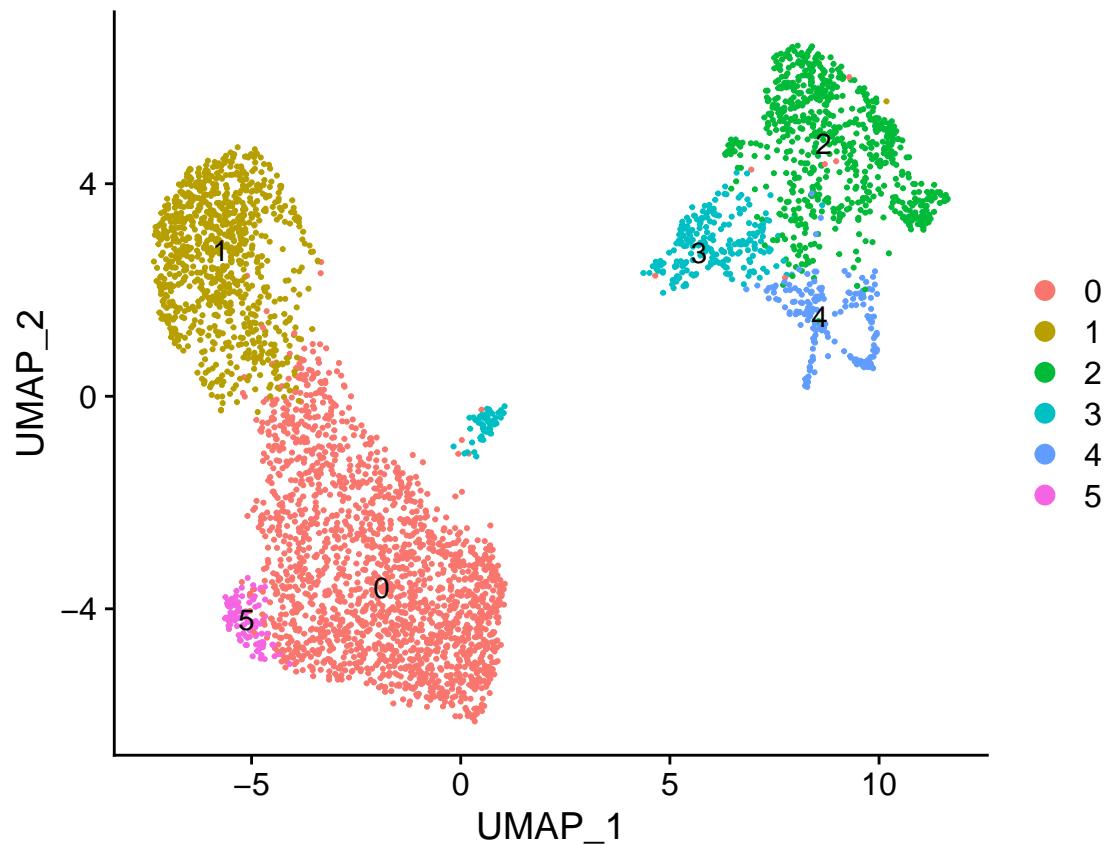
## Clustering of cells

```
results <- FindNeighbors(results, dims = 1:20)
results <- FindClusters(results, resolution = 0.3)
```

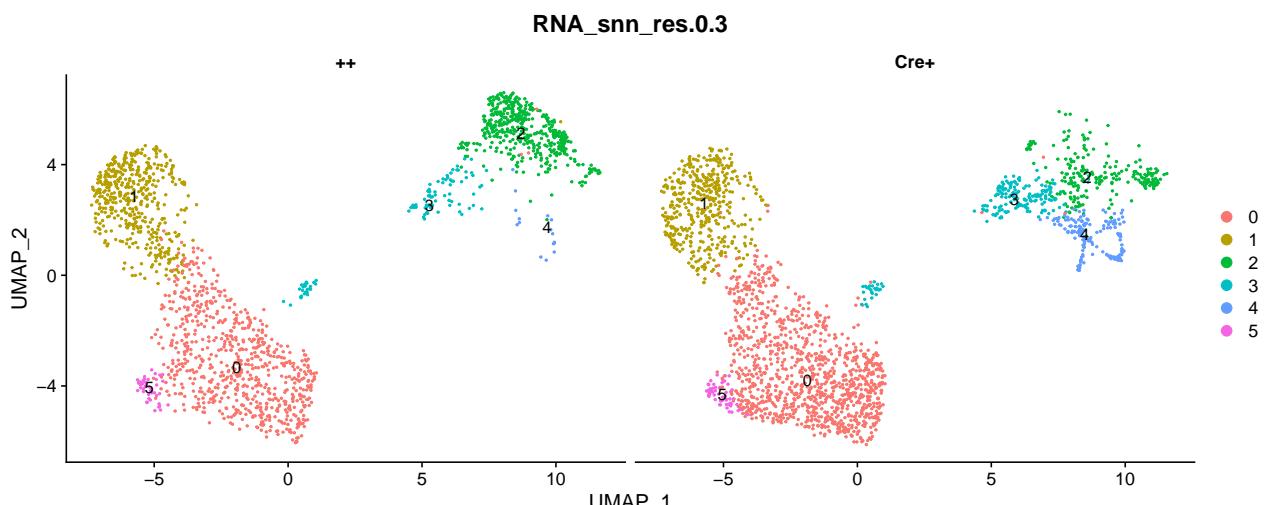
```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
## Number of nodes: 3958
## Number of edges: 139307
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8872
## Number of communities: 6
## Elapsed time: 0 seconds
```

```
Idents(results) <- "RNA_snn_res.0.3"
DimPlot(results, group.by = "RNA_snn_res.0.3", label = TRUE)
```

### RNA\_snn\_res.0.3

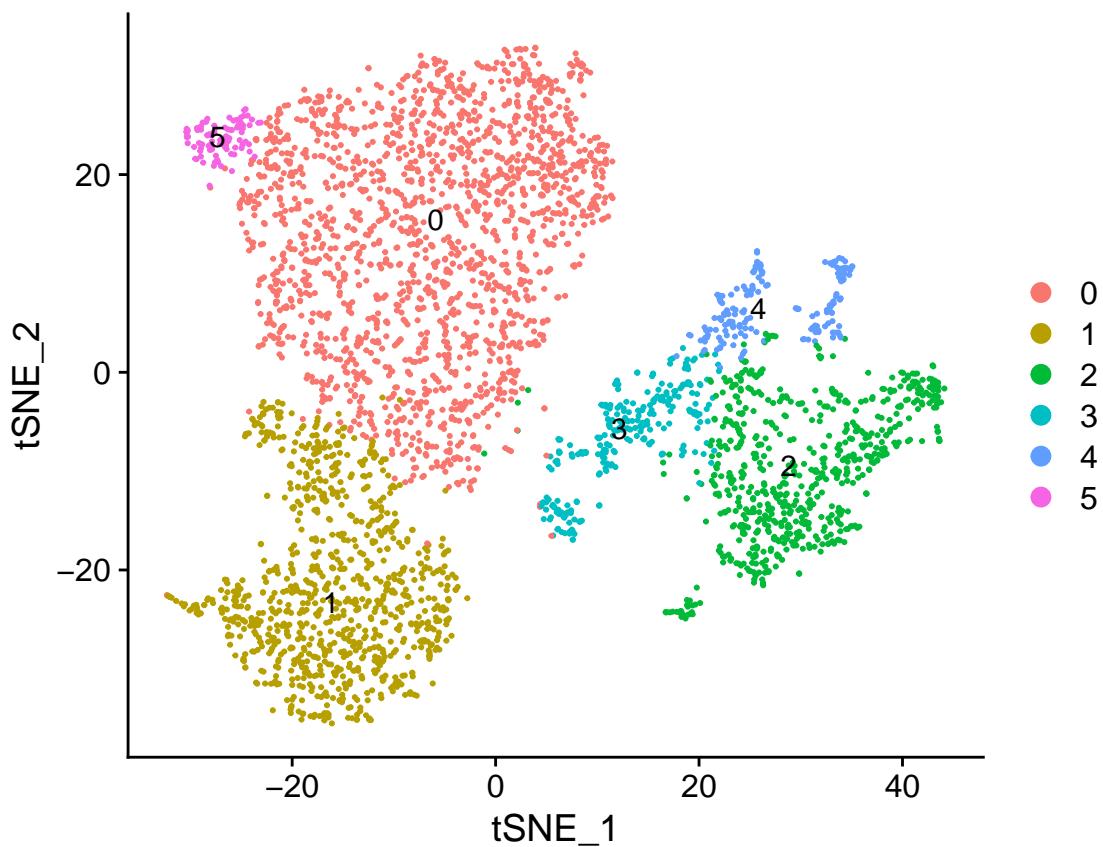


```
DimPlot(results, split.by = "treatment", group.by = "RNA_snn_res.0.3",  
        label = TRUE)
```

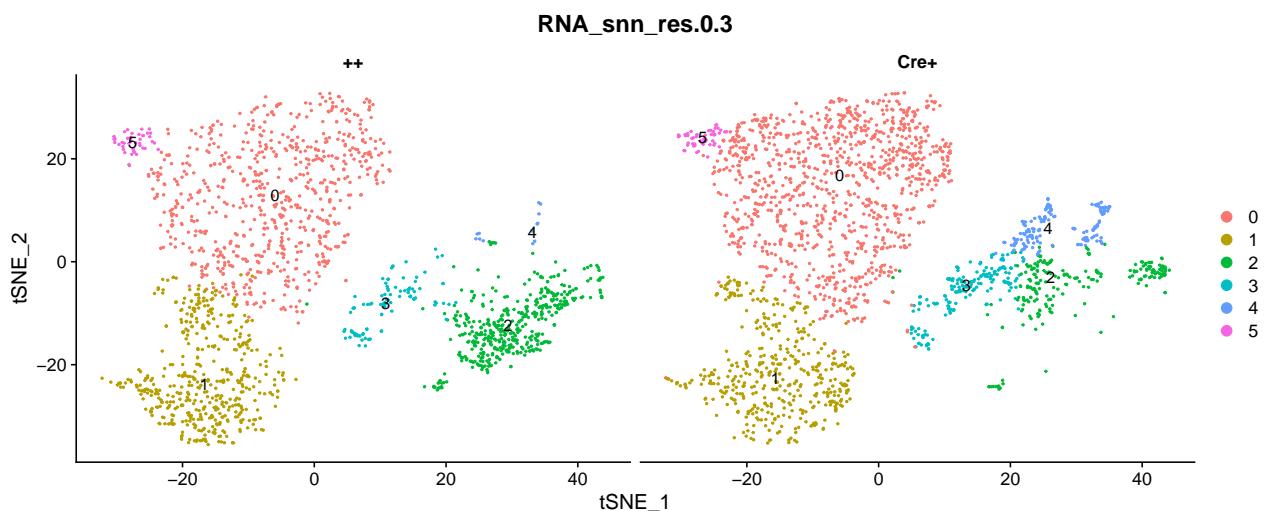


```
DimPlot(results, group.by = "RNA_snn_res.0.3", reduction = "tsne", label = TRUE)
```

## RNA\_snn\_res.0.3



```
DimPlot(results, split.by = "treatment", group.by = "RNA_snn_res.0.3",
        reduction = "tsne", label = TRUE)
```



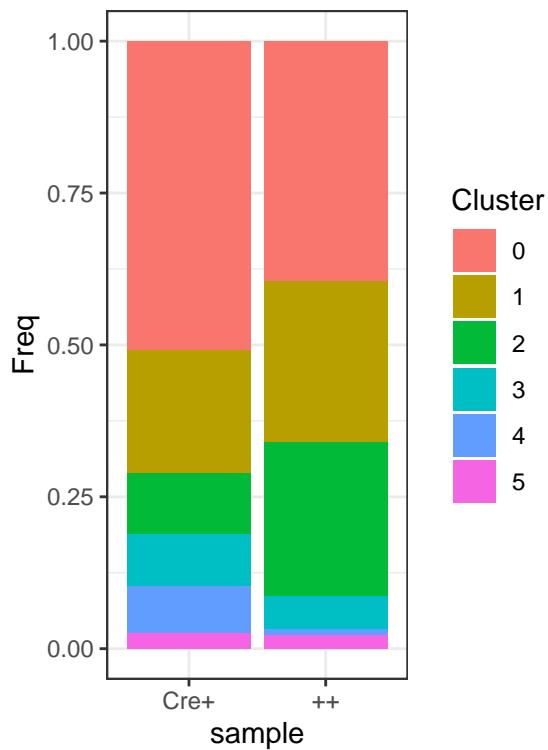
Distribution of each cluster across the samples

```
source("../R/SeuratFreqTable.R")
freq.celltype.list <- list(
  `Cre+` = Seurat2CellFreqTable(subset(results, subset = treatment == "Cre +
  +"), slotName = "RNA_snn_res.0.3"),
```

```

`++` = Seurat2CellFreqTable(subset(results, subset = treatment == "++"),
                           slotName = "RNA_snn_res.0.3")
)

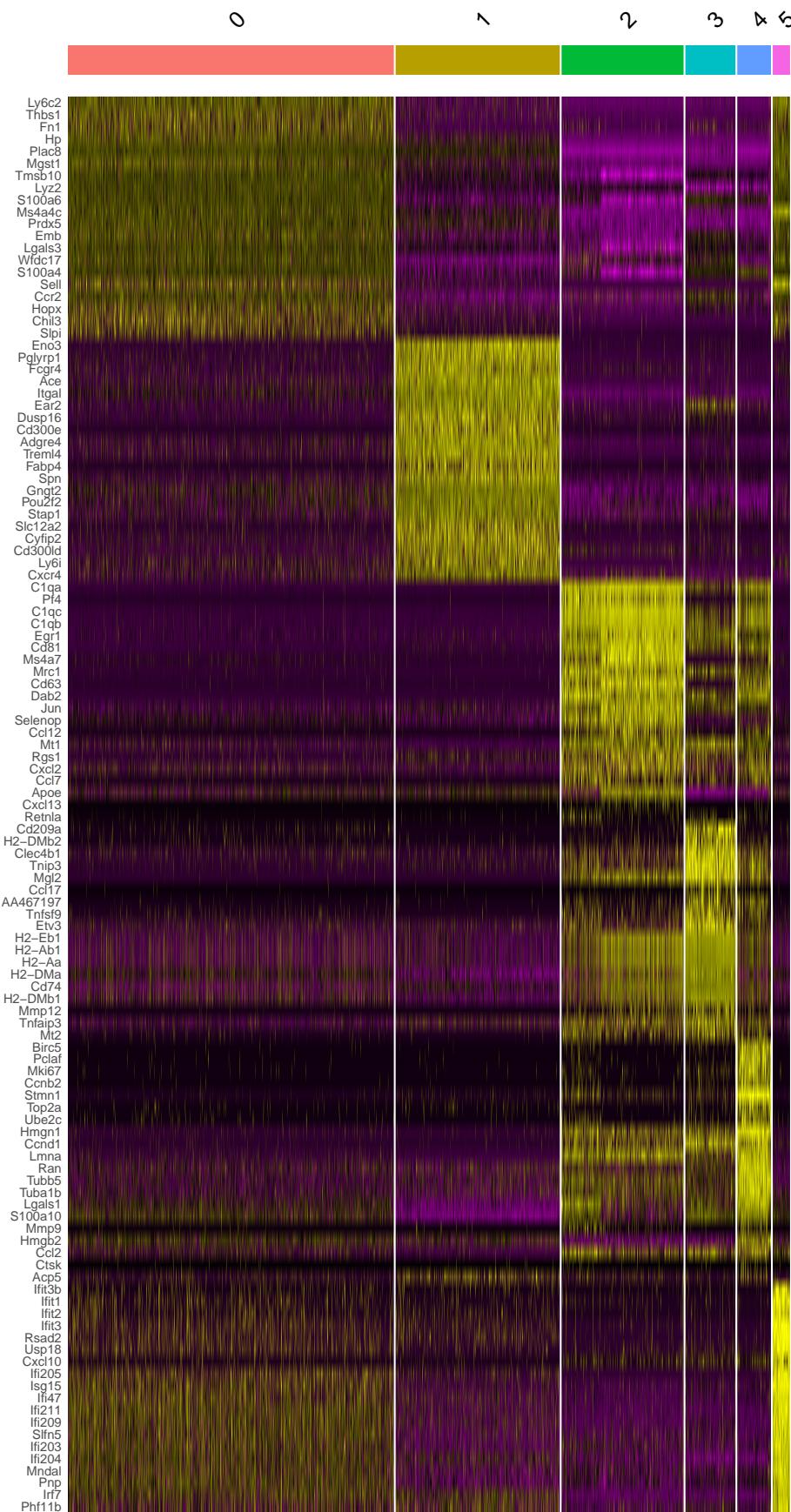
source("~/Desktop/velocity/Script/barChart.R")
barChart(freq.celltype.list) + labs(fill = "Cluster")
```



### 9.3 Population characterization

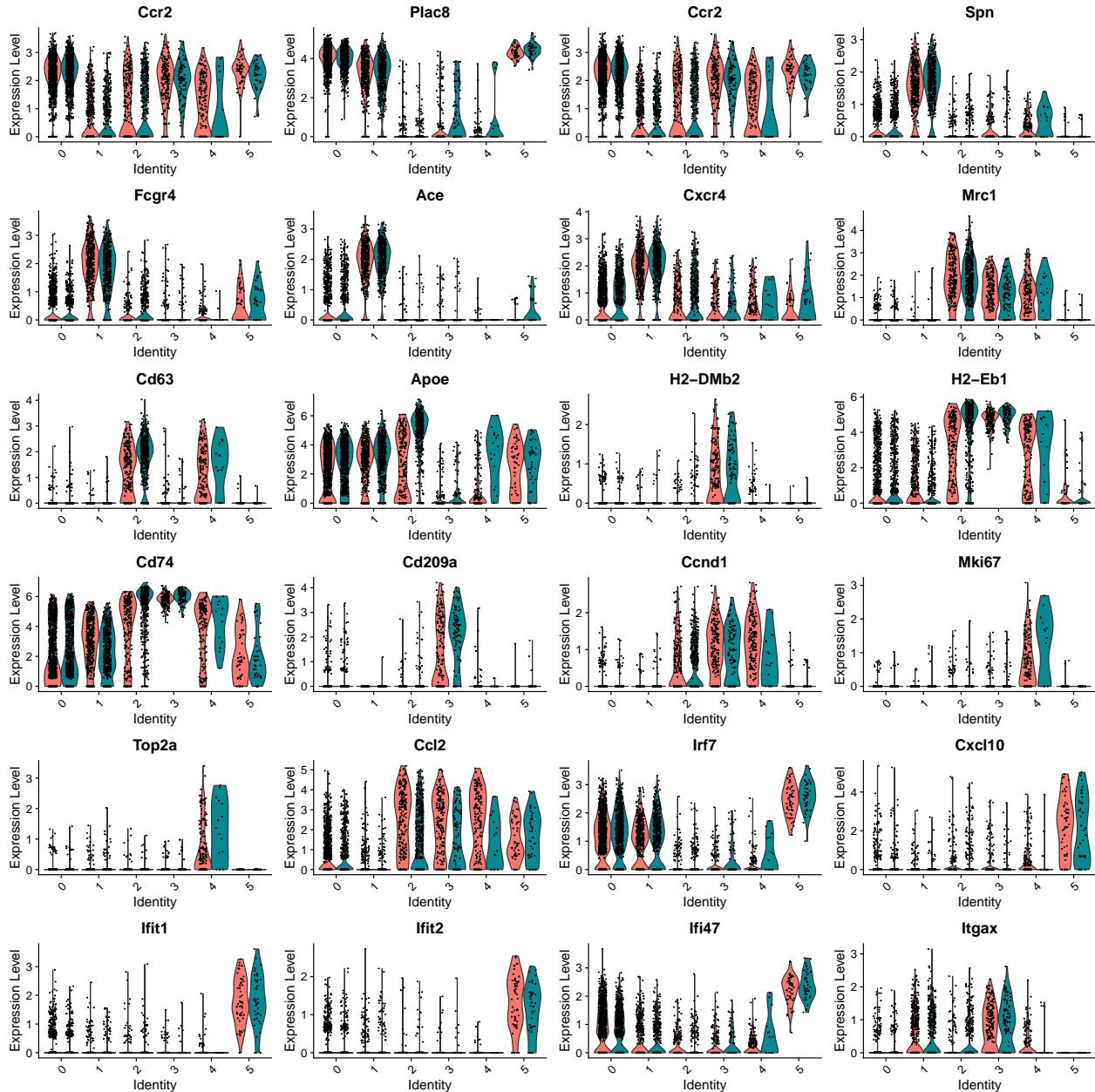
```

library(dplyr)
all_cluster.markers <- FindAllMarkers(results)
top20 <- all_cluster.markers %>% group_by(cluster) %>% top_n(n = 20, wt =
  avg_log2FC)
DoHeatmap(results, features = top20$gene) + NoLegend()
```



Expression of markers:

```
VlnPlot(results, features = c("Ccr2", "Plac8", "Ccr2", "Spn",
                               "Fcgr4", "Ace", "Cxcr4",
                               "Mrc1", "Cd63", "Apoe",
                               "H2-DMb2", "H2-Eb1", "Cd74", "Cd209a",
                               "Ccnd1", "Mki67", "Top2a", "Ccl2",
                               "Irf7", "Cxcl10", "Ifit1", "Ifit2", "Ifi47",
                               "Itgax"),
        split.by = "object_before_integrated", ncol = 4)
```



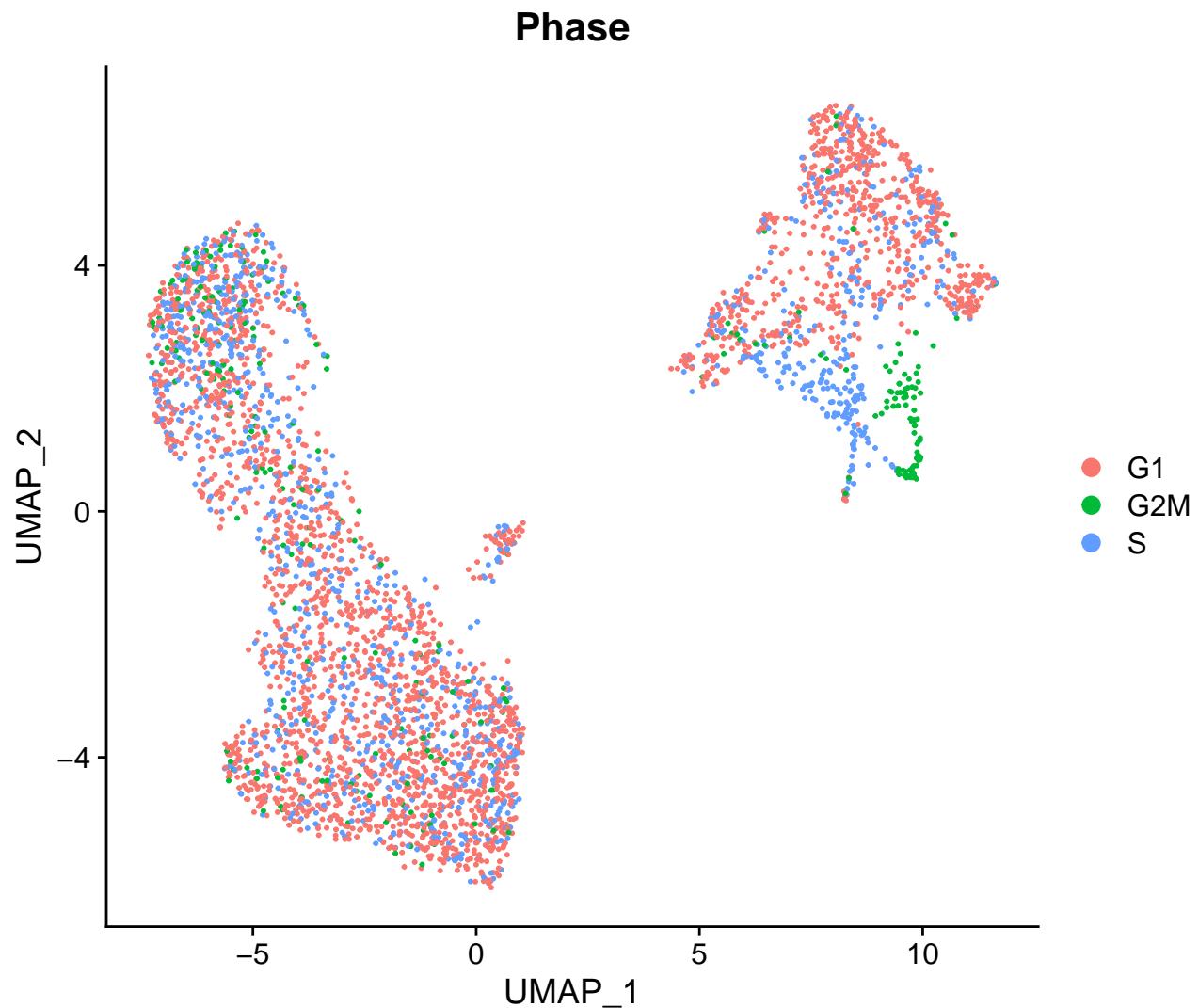
#### 9.4 Cell-cycle analysis

```
library(cowplot)
```

```

1 data("geneinfo_human", package = "nichenetr")
2 s.genes <- nichenetr::convert_human_to_mouse_symbols(cc.genes.updated.2019
3   $s.genes)
4 g2m.genes <- nichenetr::convert_human_to_mouse_symbols(cc.genes.updated
5   .2019$g2m.genes)
6 results <- CellCycleScoring(results, s.features = s.genes, g2m.features =
6   g2m.genes, set.ident = FALSE)
7 DimPlot(results, group.by = "Phase")

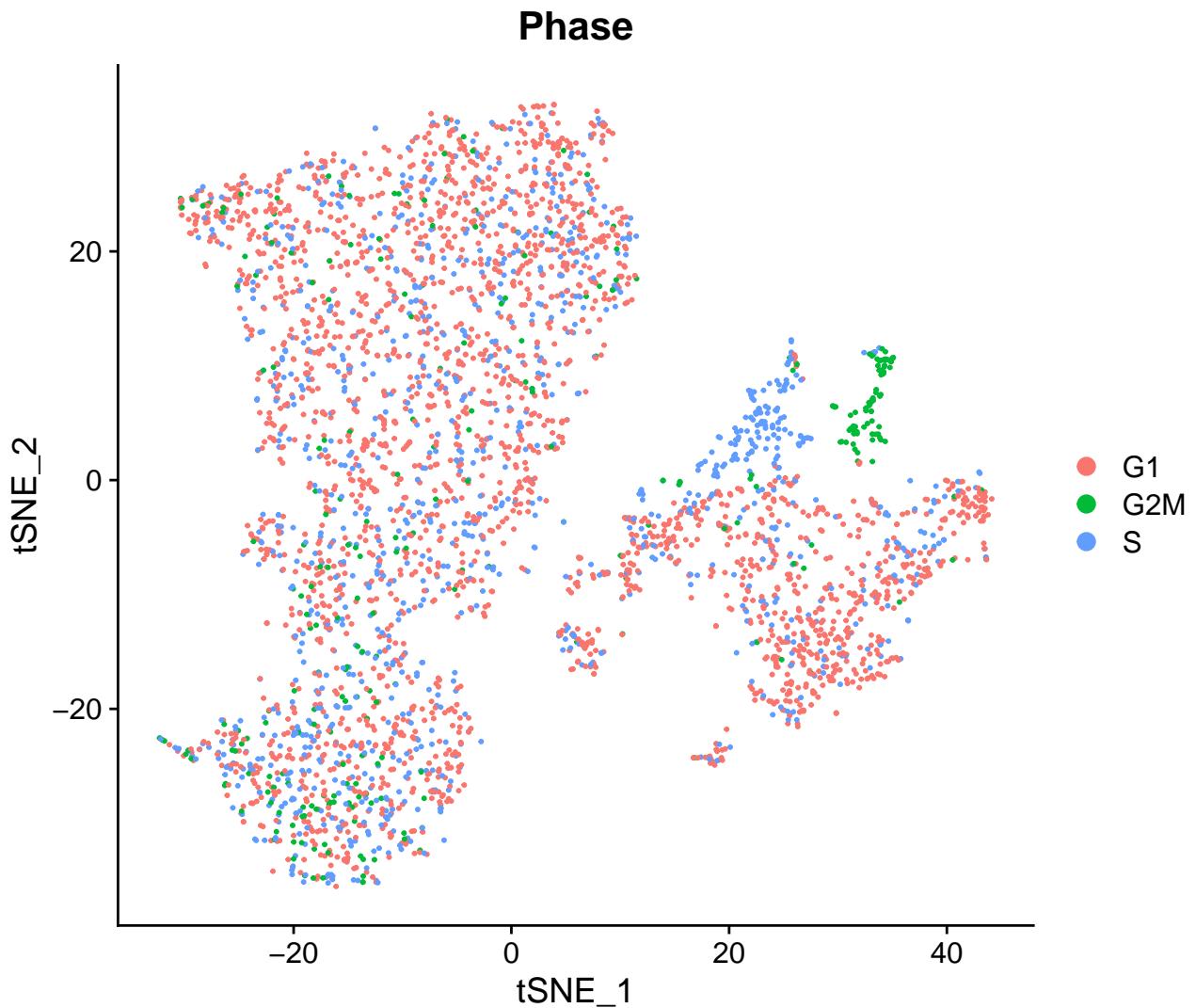
```



```

1 DimPlot(results, group.by = "Phase", reduction = "tsne")

```

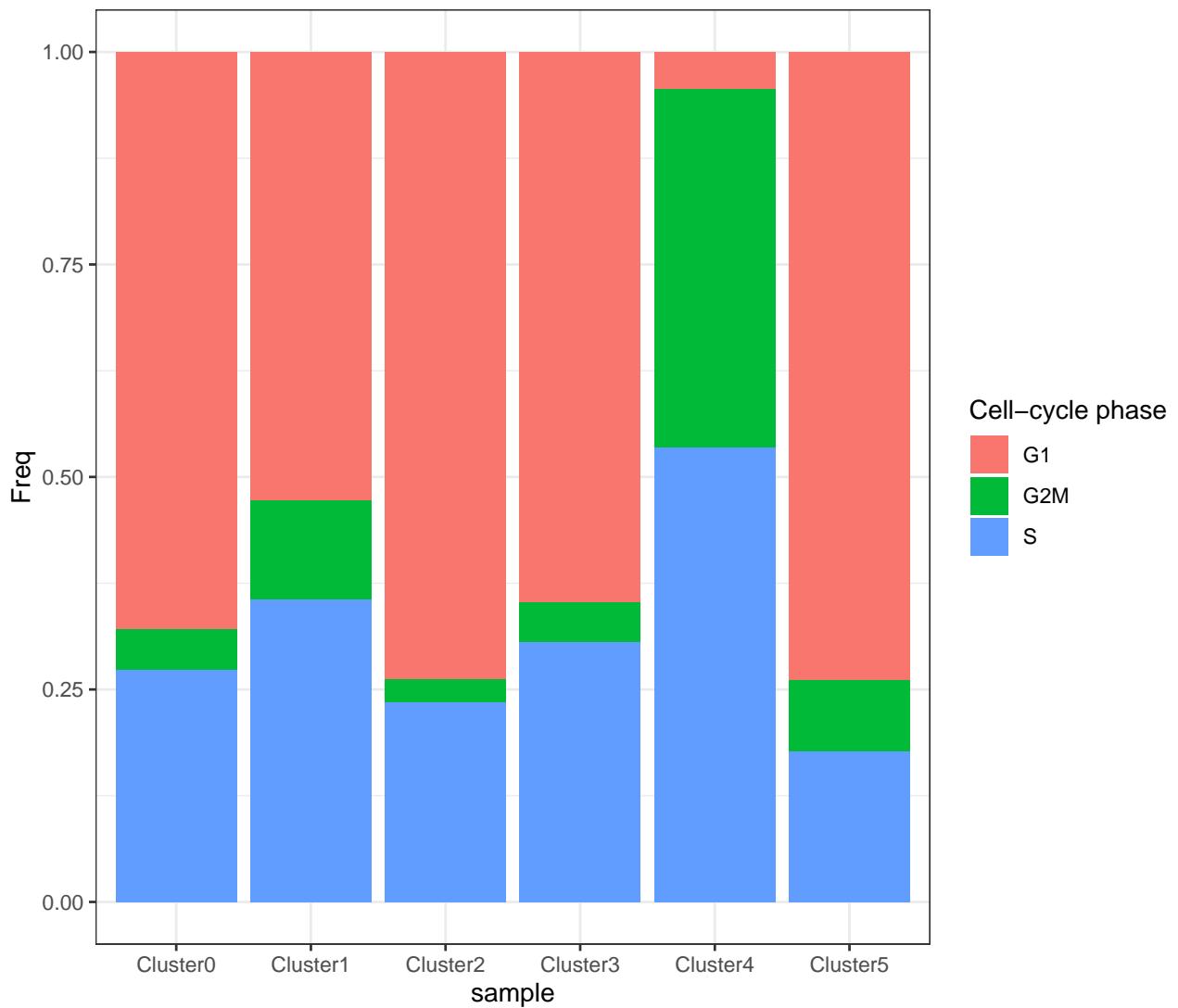


Distribution of cell phases

```

freq.celltype.list <- list(
  Cluster0 = Seurat2CellFreqTable(subset(results, ident = 0), slotName = "1
  Phase"),
  Cluster1 = Seurat2CellFreqTable(subset(results, ident = 1), slotName = "2
  Phase"),
  Cluster2 = Seurat2CellFreqTable(subset(results, ident = 2), slotName = "3
  Phase"),
  Cluster3 = Seurat2CellFreqTable(subset(results, ident = 3), slotName = "4
  Phase"),
  Cluster4 = Seurat2CellFreqTable(subset(results, ident = 4), slotName = "5
  Phase"),
  Cluster5 = Seurat2CellFreqTable(subset(results, ident = 5), slotName = "6
  Phase")
)
barChart(freq.celltype.list) + labs(fill = "Cell-cycle phase")7
8
9

```



## 10 Session information

```
sessionInfo()
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.3 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/liblapack.so.3
##
## locale:
## [1] LC_CTYPE=en_US.UTF-8          LC_NUMERIC=C
## [3] LC_TIME=en_GB.UTF-8          LC_COLLATE=en_US.UTF-8
## [5] LC_MONETARY=en_GB.UTF-8       LC_MESSAGES=en_US.UTF-8
## [7] LC_PAPER=en_GB.UTF-8          LC_NAME=C
## [9] LC_ADDRESS=C                  LC_TELEPHONE=C
```

```

## [11] LC_MEASUREMENT=en_GB.UTF-8 LC_IDENTIFICATION=C          15
##
## attached base packages:
## [1] parallel    stats      graphics   grDevices utils       16
##           datasets
## [8] methods     base        17
##
## other attached packages:
## [1] cowplot_1.1.1           dplyr_1.0.7            18
## [3] RColorBrewer_1.1-2      celldex_1.0.0          19
## [5] SingleR_1.4.1          SummarizedExperiment_1.20.0 20
## [7] Biobase_2.50.0          GenomicRanges_1.42.0    21
## [9] GenomeInfoDb_1.26.7      IRanges_2.24.1          22
## [11] S4Vectors_0.28.1        BiocGenerics_0.36.1    23
## [13] MatrixGenerics_1.2.1   matrixStats_0.60.0    24
## [15] ggplot2_3.3.5          SeuratObject_4.0.2    25
## [17] Seurat_4.0.3            26
##
## loaded via a namespace (and not attached):
## [1] utf8_1.2.2                reticulate_1.20         27
## [3] tidyselect_1.1.1           RSQLite_2.2.7           28
## [5] AnnotationDbi_1.52.0      htmlwidgets_1.5.3        29
## [7] grid_4.0.3                 BiocParallel_1.24.1      30
## [9] Rtsne_0.15                codetools_0.2-18        31
## [11] munsell_0.5.0              pROC_1.17.0.1          32
## [13] ica_1.0-2                 future_1.21.0          33
## [15] miniUI_0.1.1.1            withr_2.4.2             34
## [17] colorspace_2.0-2          highr_0.9               35
## [19] knitr_1.33                rstudioapi_0.13        36
## [21] ROCR_1.0-11               tensor_1.5              37
## [23] listenv_0.8.0              labeling_0.4.2          38
## [25] GenomeInfoDbData_1.2.4    polyclip_1.10-0         39
## [27] bit64_4.0.5               farver_2.1.0            40
## [29] parallelly_1.27.0          vctrs_0.3.8             41
## [31] generics_0.1.0              ipred_0.9-11            42
## [33] xfun_0.24                 BiocFileCache_1.14.0    43
## [35] randomForest_4.6-14        R6_2.5.0                44
## [37] rsvd_1.0.5                bitops_1.0-7            45
## [39] spatstat.utils_2.2-0       cachem_1.0.5            46
## [41] DelayedArray_0.16.3        assertthat_0.2.1        47
## [43] promises_1.2.0.1           scales_1.1.1            48
## [45] nnet_7.3-14               gtable_0.3.0            49
## [47] beachmat_2.6.4             globals_0.14.0          50
## [49] goftest_1.2-2              timeDate_3043.102       51
## [51] rlang_0.4.11               splines_4.0.3           52
## [53] lazyeval_0.2.2              ModelMetrics_1.2.2.2     53
## [55] checkmate_2.0.0             spatstat.geom_2.2-2      54
## [57] BiocManager_1.30.16         yaml_2.2.1              55
## [59] reshape2_1.4.4              abind_1.4-5             56
## [61] backports_1.2.1             httpuv_1.6.1            57
## [63] Hmisc_4.5-0                DiagrammeR_1.0.6.1      58
## [65] caret_6.0-88               tools_4.0.3              59
## [67] lava_1.6.9                 ellipsis_0.3.2          60
## [69] spatstat.core_2.3-0         proxy_0.4-26            61

```

|                                    |                               |     |
|------------------------------------|-------------------------------|-----|
| ## [71] ggridges_0.5.3             | Rcpp_1.0.7                    | 68  |
| ## [73] plyr_1.8.6                 | base64enc_0.1-3               | 69  |
| ## [75] visNetwork_2.0.0.9         | sparseMatrixStats_1.2.1       | 70  |
| ## [77] zlibbioc_1.36.0            | purrr_0.3.4                   | 71  |
| ## [79] RCurl_1.98-1.3             | rpart_4.1-15                  | 72  |
| ## [81] deldir_0.2-10              | pbapply_1.4-3                 | 73  |
| ## [83] zoo_1.8-9                  | nichenetr_1.0.0               | 74  |
| ## [85] ggrepel_0.9.1              | cluster_2.1.0                 | 75  |
| ## [87] magrittr_2.0.1             | data.table_1.14.0             | 76  |
| ## [89] RSpecsra_0.16-0            | scattermore_0.7               | 77  |
| ## [91] lmtest_0.9-38              | RANN_2.6.1                    | 78  |
| ## [93] fitdistrplus_1.1-5         | hms_1.1.0                     | 79  |
| ## [95] patchwork_1.1.1            | mime_0.11                     | 80  |
| ## [97] evaluate_0.14              | xtable_1.8-4                  | 81  |
| ## [99] jpeg_0.1-9                 | gridExtra_2.3                 | 82  |
| ## [101] compiler_4.0.3            | tibble_3.1.3                  | 83  |
| ## [103] KernSmooth_2.23-20        | crayon_1.4.1                  | 84  |
| ## [105] htmltools_0.5.1.1         | tzdb_0.1.2                    | 85  |
| ## [107] mgcv_1.8-33               | later_1.2.0                   | 86  |
| ## [109] Formula_1.2-4             | tidyr_1.1.3                   | 87  |
| ## [111] lubridate_1.7.10          | DBI_1.1.1                     | 88  |
| ## [113] ExperimentHub_1.16.1      | dbplyr_2.1.1                  | 89  |
| ## [115] MASS_7.3-53               | rappdirs_0.3.3                | 90  |
| ## [117] readr_2.0.0               | Matrix_1.3-4                  | 91  |
| ## [119] cli_3.0.1                 | gower_0.2.2                   | 92  |
| ## [121] igraph_1.2.6              | pkgconfig_2.0.3               | 93  |
| ## [123] foreign_0.8-81            | plotly_4.9.4.1                | 94  |
| ## [125] spatstat.sparse_2.0-0     | recipes_0.1.16                | 95  |
| ## [127] foreach_1.5.1             | XVector_0.30.0                | 96  |
| ## [129] prodlim_2019.11.13        | stringr_1.4.0                 | 97  |
| ## [131] digest_0.6.27             | sctransform_0.3.2             | 98  |
| ## [133] RcppAnnoy_0.0.19          | spatstat.data_2.1-0           | 99  |
| ## [135] rmarkdown_2.9             | leiden_0.3.9                  | 100 |
| ## [137] htmlTable_2.2.1           | uwot_0.1.10.9000              | 101 |
| ## [139] DelayedMatrixStats_1.12.3 | curl_4.3.2                    | 102 |
| ## [141] shiny_1.6.0               | lifecycle_1.0.0               | 103 |
| ## [143] nlme_3.1-152              | jsonlite_1.7.2                | 104 |
| ## [145] BiocNeighbors_1.8.2       | viridisLite_0.4.0             | 105 |
| ## [147] limma_3.46.0              | fansi_0.5.0                   | 106 |
| ## [149] pillar_1.6.2              | lattice_0.20-41               | 107 |
| ## [151] fastmap_1.1.0             | httr_1.4.2                    | 108 |
| ## [153] survival_3.2-7            | interactiveDisplayBase_1.28.0 | 109 |
| ## [155] glue_1.4.2                | fdrtool_1.2.16                | 110 |
| ## [157] png_0.1-7                 | iterators_1.0.13              | 111 |
| ## [159] BiocVersion_3.12.0        | bit_4.0.4                     | 112 |
| ## [161] class_7.3-17              | stringi_1.7.3                 | 113 |
| ## [163] blob_1.2.2                | BiocSingular_1.6.0            | 114 |
| ## [165] AnnotationHub_2.22.1      | caTools_1.18.2                | 115 |
| ## [167] latticeExtra_0.6-29       | memoise_2.0.0                 | 116 |
| ## [169] e1071_1.7-8               | irlba_2.3.3                   | 117 |
| ## [171] future.apply_1.7.0        |                               | 118 |

## 11 References