

# 3-Merge and celltyping

BAI Qiang\*

2022-03-25 16:42:15 +0100

## Contents

<b>1 Description</b>	<b>2</b>
<b>2 Load data:</b>	<b>2</b>
<b>3 Make metadata:</b>	<b>2</b>
<b>4 Merge data and preparation for celltyping</b>	<b>4</b>
4.1 Data processing: normalizaiton and scaling . . . . .	4
4.2 Dimension reduction to give a global view of all samples . . . . .	4
4.3 Cluster cells before celltyping . . . . .	4
4.4 Check for batch effects . . . . .	4
<b>5 Celltyping</b>	<b>10</b>
<b>6 Normalization and redo cluster in filtered “monocytes”</b>	<b>16</b>
6.1 Data processing in filtered “monocytes” . . . . .	16
6.2 Dimension reduction in filtered “monocytes” . . . . .	16
6.3 Clustering filtered “monocytes” . . . . .	17
<b>7 Characterization of clustered populations</b>	<b>22</b>
<b>8 Cell-cycle analysis</b>	<b>24</b>
<b>9 Session information</b>	<b>25</b>

---

\*University Liege, mail qiang.bai@uliege.be

# 1 Description

To identify and remove potential contaminated cell populations, cells from all samples were firstly merged and clustered with high resolution (res = 0.5) after the expression was normalized. Fourteen clusters were obtained (cluster 0 – 13). After careful check for the potential markers, the cluster 6, 12 and 13 were identified as lymphocytes (high CD3E expression), mucosal epithelial cells and ciliated epithelial cells (high expression of EPCAM, TEKT2 and CFAP74, low PTPRC expression).

To characterize the monocyte/macrophage populations, we repeated normalization, scaling and FindVariableFeatures on only filtered “monocytes” data. The 4 clusters shown in Figure 4A were identified in the integrated data using “FindClusters” with resolution of 0.1. To visualize the data, a non-linear dimension reduction was calculated using “RunUMAP”. The cell cycle estimate was made with Seurat function CellCycleScoring.

## 2 Load data:

```
# load package and data
library(Seurat)

initiation.analysis.folder <- "../2-Sample_QC_and_cell_filtering"
file.names <- list.files(initiation.analysis.folder, pattern = "*.rds")
sample.names <- sub(".rds", "", file.names)

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

# load individual samples under appropriate names
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 Make metadata:

The metadata of samples are also available with resource data in NCBI GEO database or click this (link to download)[<https://www.ncbi.nlm.nih.gov/geo/download/?acc=GSE183974&format=file&file=GSE183974%5Fmetadata%5Fall%5Ffiltered%5Fcells%2Ecsv%2Egz>]: (<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE183974>) [<https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE183974>]

Here is the sample order:

```
table.samples$sample.names
## [1] "NGS19_I415_Dim_LBA_Hum3.seuratObject"
## [2] "NGS19_I679_Dim_LBA_Hum4.seuratObject"
## [3] "NGS19_J028_Dim_LBA_Hum5.seuratObject"
## [4] "NGS19_J141_Dim_LBA_Hum7.seuratObject"
## [5] "NGS19_J142_Dim_LBA_Hum8.seuratObject"
## [6] "NGS19_J263_Dim_LBA_Hum9.seuratObject"
## [7] "NGS19_J264_Dim_LBA_Hum10.seuratObject"
## [8] "NGS19_K359_Dim_LBA_Hum11.seuratObject"
## [9] "NGS19_K360_Dim_LBA_Hum12.seuratObject"
```

```

sample.metadata <- data.frame(
  group = c("Healthy\u2225non-smokers", "Non-COPD\u2225smokers",
           "Non-COPD\u2225smokers", "Non-COPD\u2225smokers",
           "Healthy\u2225non-smokers", "COPD\u2225smokers",
           "Healthy\u2225non-smokers", "COPD\u2225smokers"),
  cell.type = rep("AM", length(table.samples$sample.names)),
  origin = rep("LBA", length(table.samples$sample.names)),
  sample_prefix = gsub(".*(Dim_)\|.seuratObject",
                        "\\\2", table.samples$sample.names))
rownames(sample.metadata) <- table.samples$sample.names
sample.metadata

```

	## # A tibble: 9 x 4	1			
##	group	cell.type	origin	sample_prefix	2
##	<chr>	<chr>	<chr>	<chr>	3
## 1	Healthy non-smokers	AM	LBA	LBA_Hum3	4
## 2	Non-COPD smokers	AM	LBA	LBA_Hum4	5
## 3	Non-COPD smokers	AM	LBA	LBA_Hum5	6
## 4	Non-COPD smokers	AM	LBA	LBA_Hum7	7
## 5	Healthy non-smokers	AM	LBA	LBA_Hum8	8
## 6	COPD smokers	AM	LBA	LBA_Hum9	9
## 7	Healthy non-smokers	AM	LBA	LBA_Hum10	10
## 8	COPD smokers	AM	LBA	LBA_Hum11	11
## 9	COPD smokers	AM	LBA	LBA_Hum12	12

Add metadata to cells before merging:

```

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

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

```

## 4 Merge data and preparation for celltyping

```
# add origine sample names:
for (i in table.samples$sample.names) {
  obj <- get(i)
  obj$origin <- sample.metadata[i, ]$sample_prefix
  assign(i, obj)
}

results <- merge(x = get(table.samples$sample.names[1]),
                  y = sapply(table.samples$sample.names[2:length(table.
samples$sample.names)], get ))
```

### 4.1 Data processing: normalizaiton and scaling

```
results <- NormalizeData(results, verbose = FALSE)
results <- FindVariableFeatures(results, selection.method = "vst",
nfeatures = 2000,
verbose = FALSE)
results <- ScaleData(results, features = rownames(results),
verbose = FALSE)
```

### 4.2 Dimension reduction to give a global view of all samples

```
results <- RunPCA(results, features = VariableFeatures(results), verbose =
FALSE)
results <- RunUMAP(results, dims = 1:15, verbose = FALSE)
results <- RunTSNE(results, dims = 1:15)
```

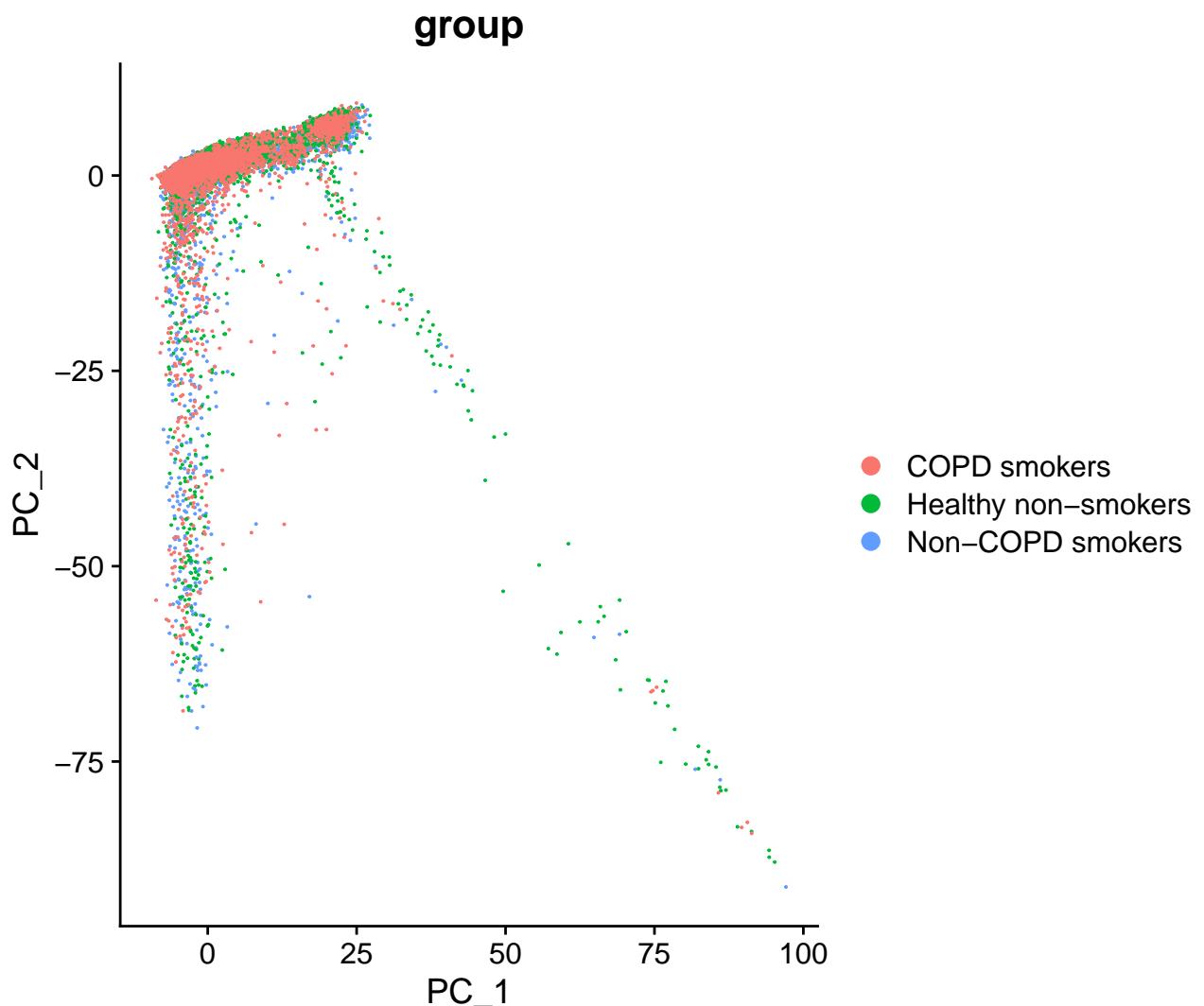
### 4.3 Cluster cells before celltyping

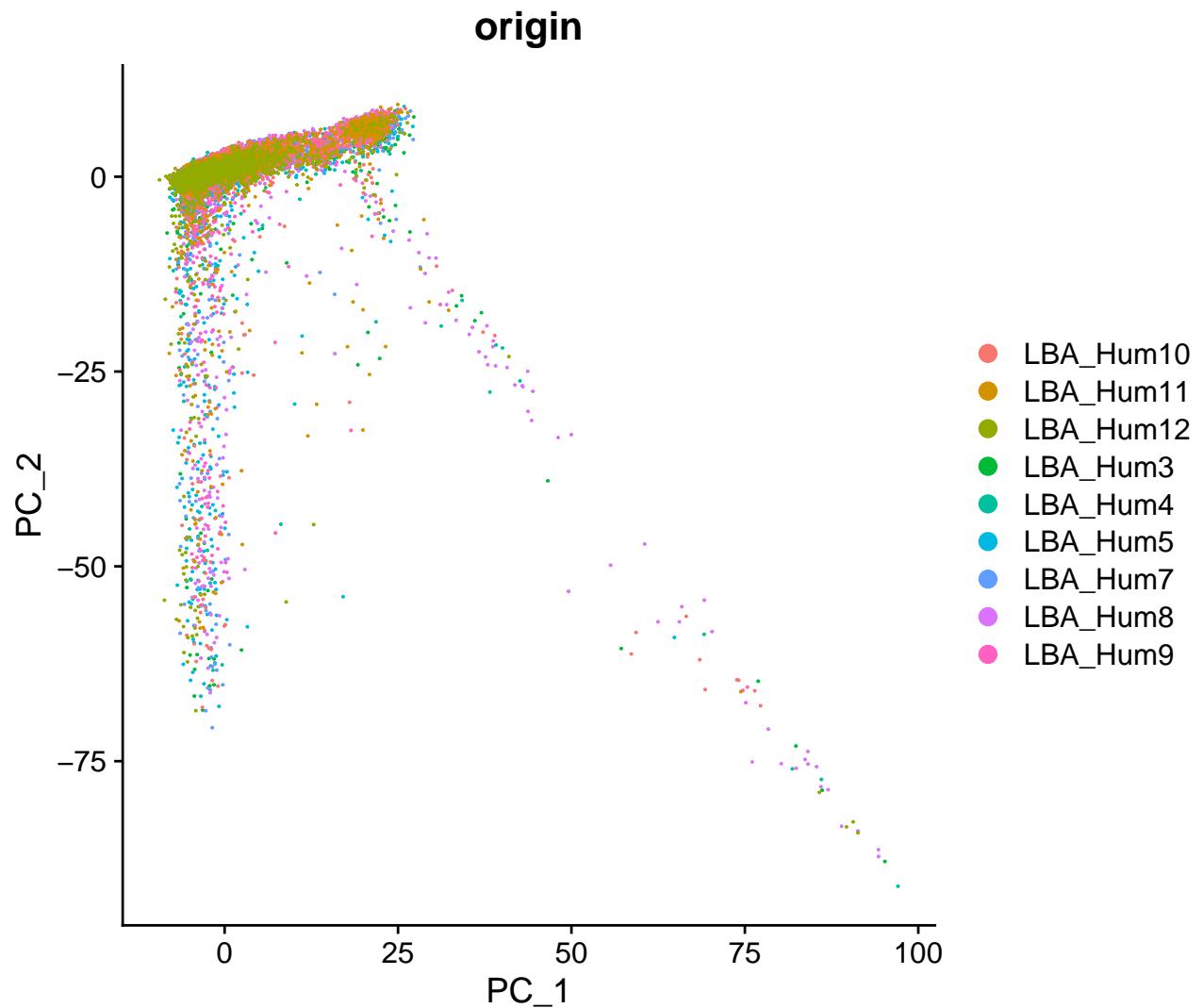
Cluster cells with a high resolution will help to find small contaminated cells in the next steps.

```
results <- FindNeighbors(results, dims = 1:15, verbose = FALSE)
results <- FindClusters(results, resolution = 0.5, verbose = FALSE)
```

### 4.4 Check for batch effects

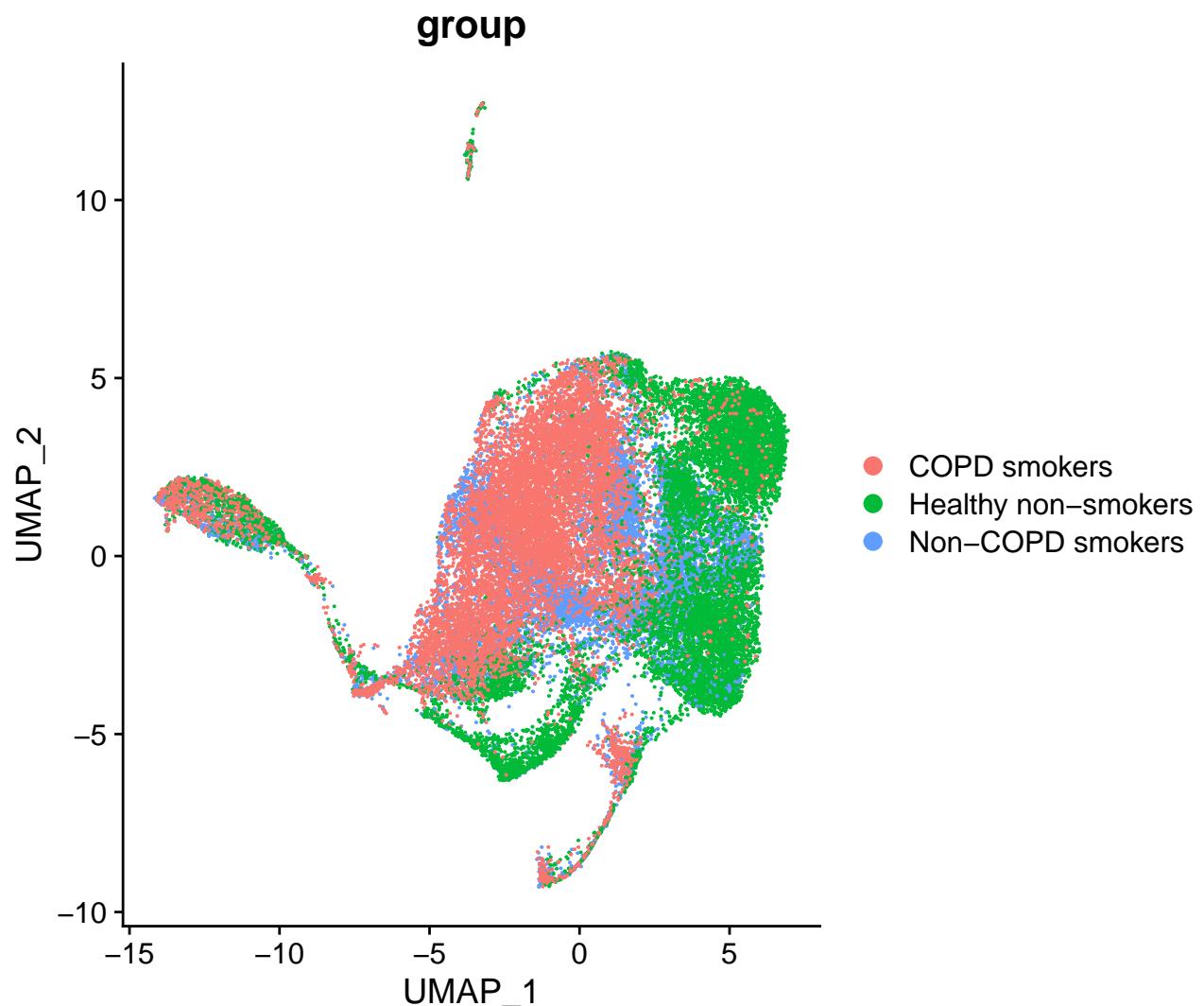
```
DimPlot(results, reduction = "pca", group.by = "group")
```

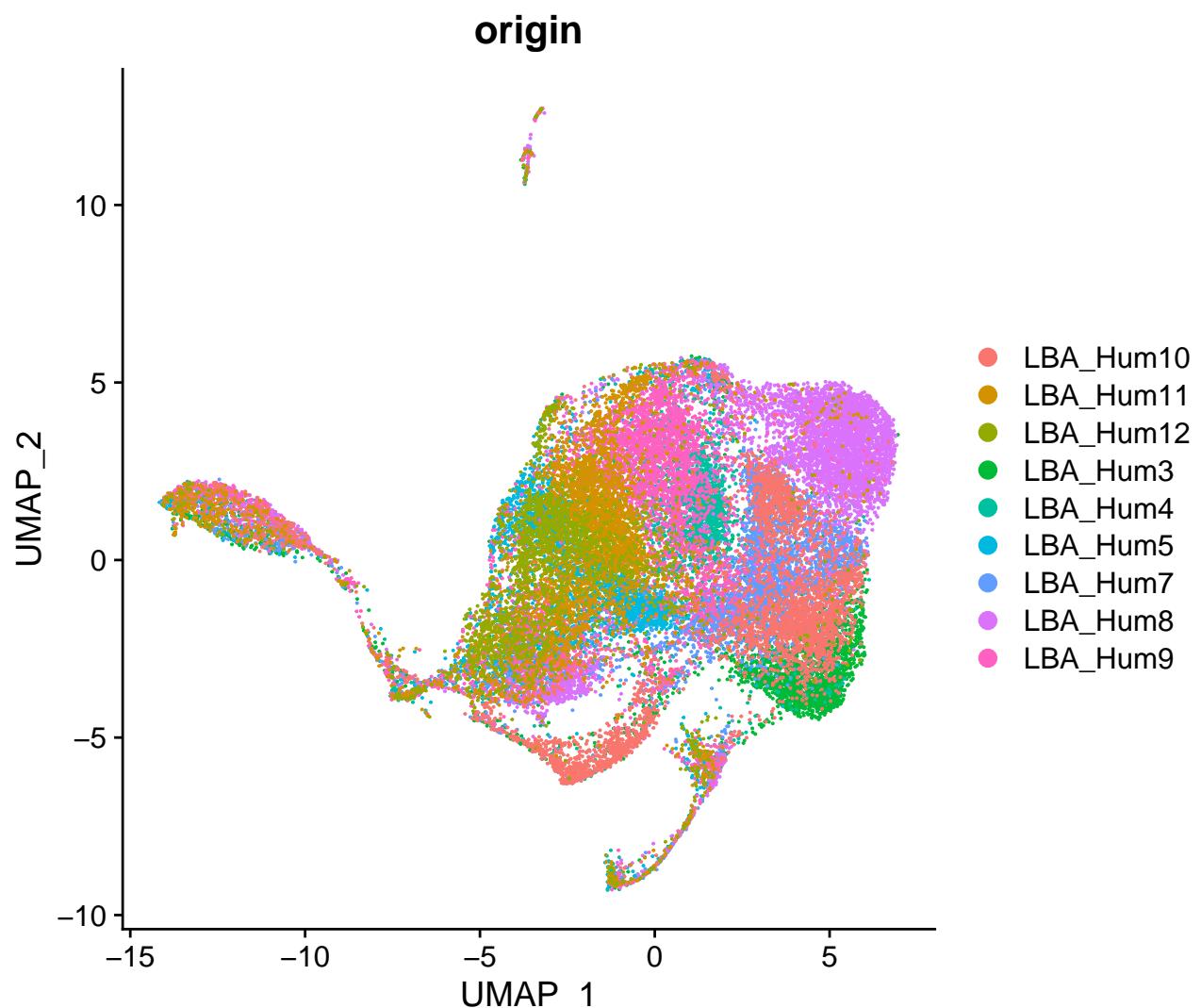




```
DimPlot(results, reduction = "umap", group.by = "group")
```

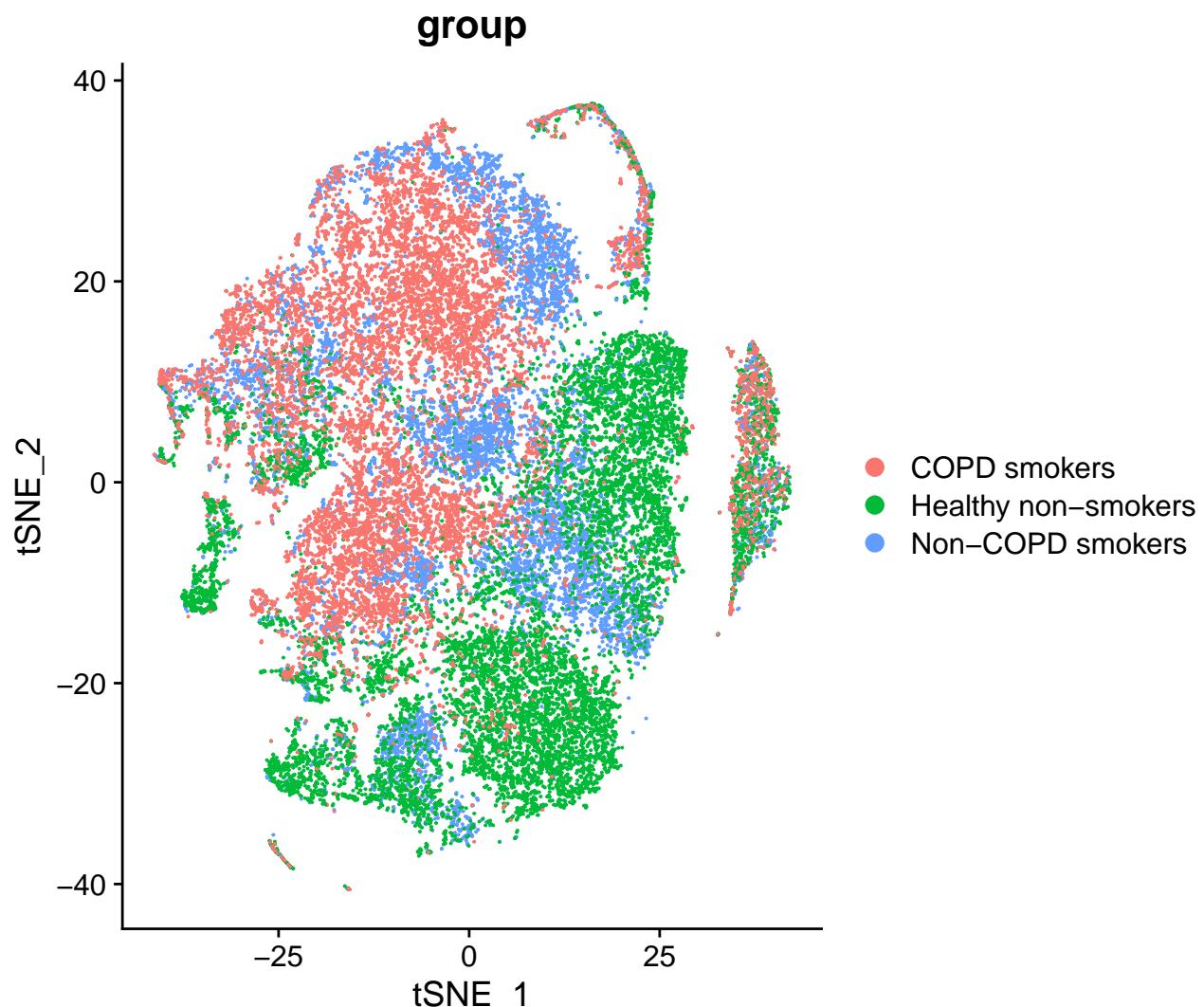
1

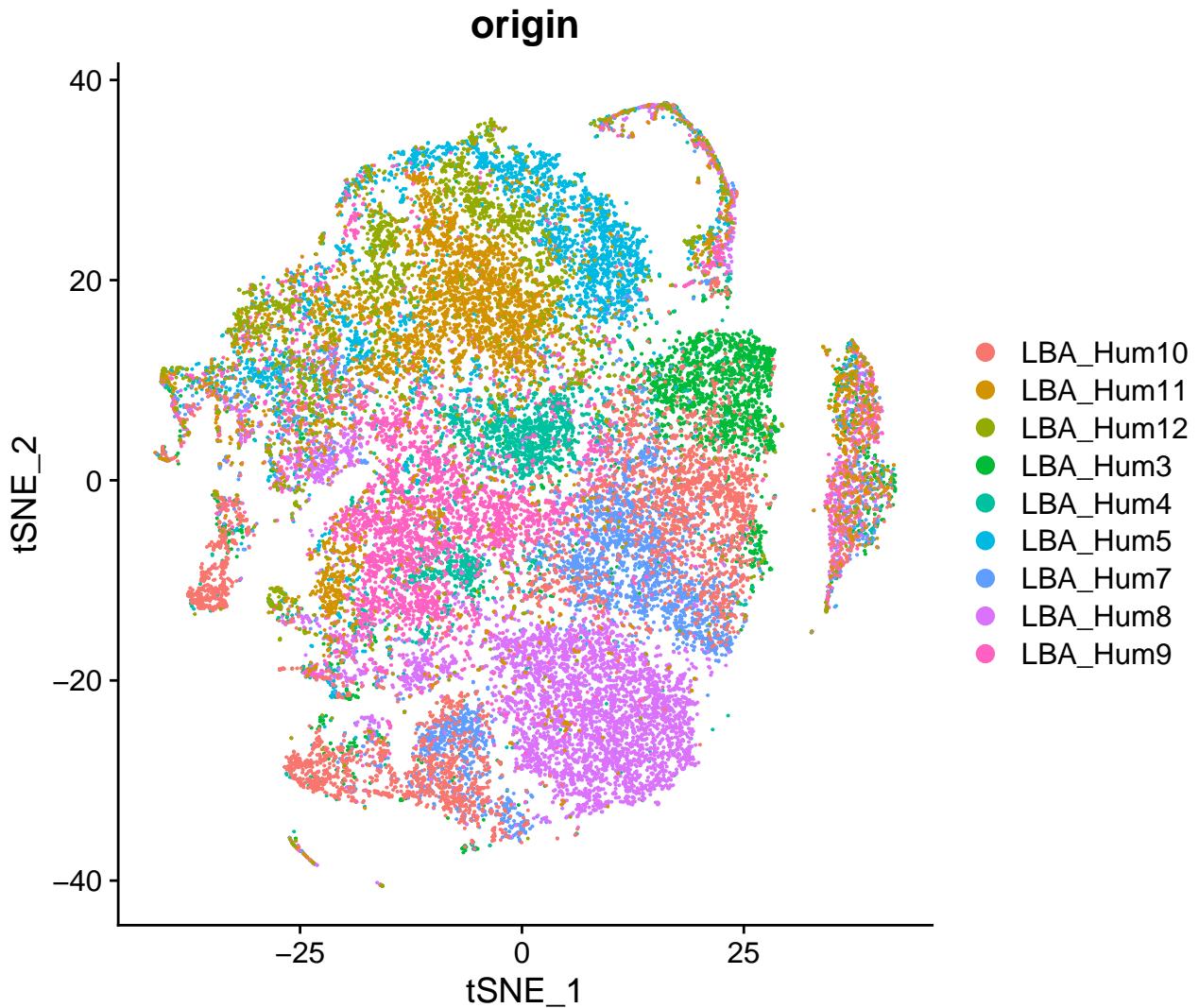




```
DimPlot(results, reduction = "tsne", group.by = "group")
```

1





From PCA plot, little batch effect can be observed. In non-linear dimension reduction, TSNE/UMAP plots, we observed difference of subset distribution across samples, but mainly across the samples with different groups. This difference could be subsets related to the sample's groups.

Save data for other analyses:

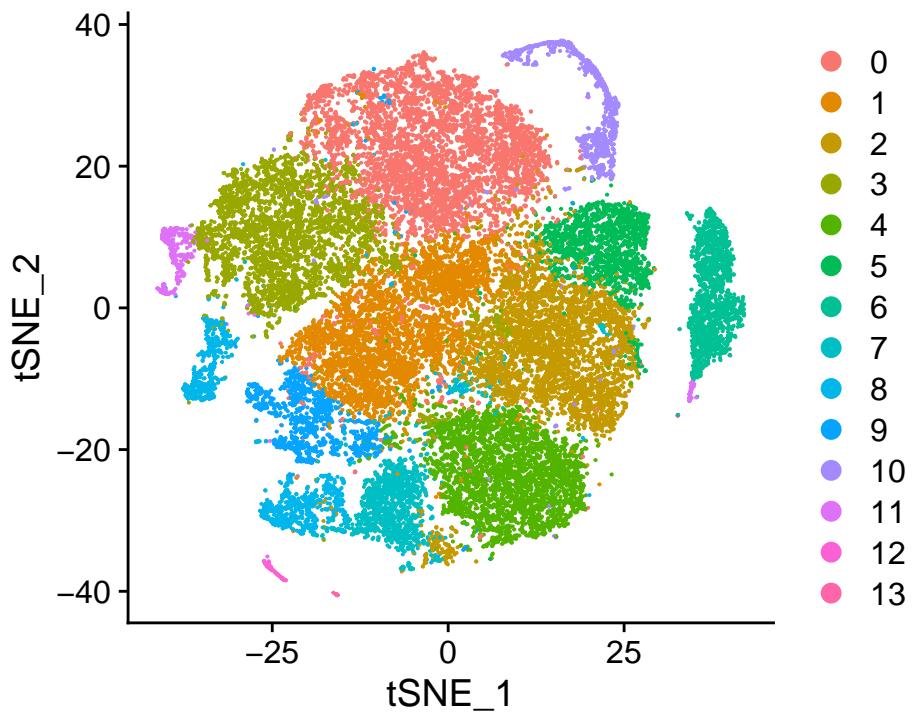
```
saveRDS(results, file = "./results.merged.Rds")
```

1

## 5 Celltyping

```
DimPlot(results, reduction = "tsne")
```

1



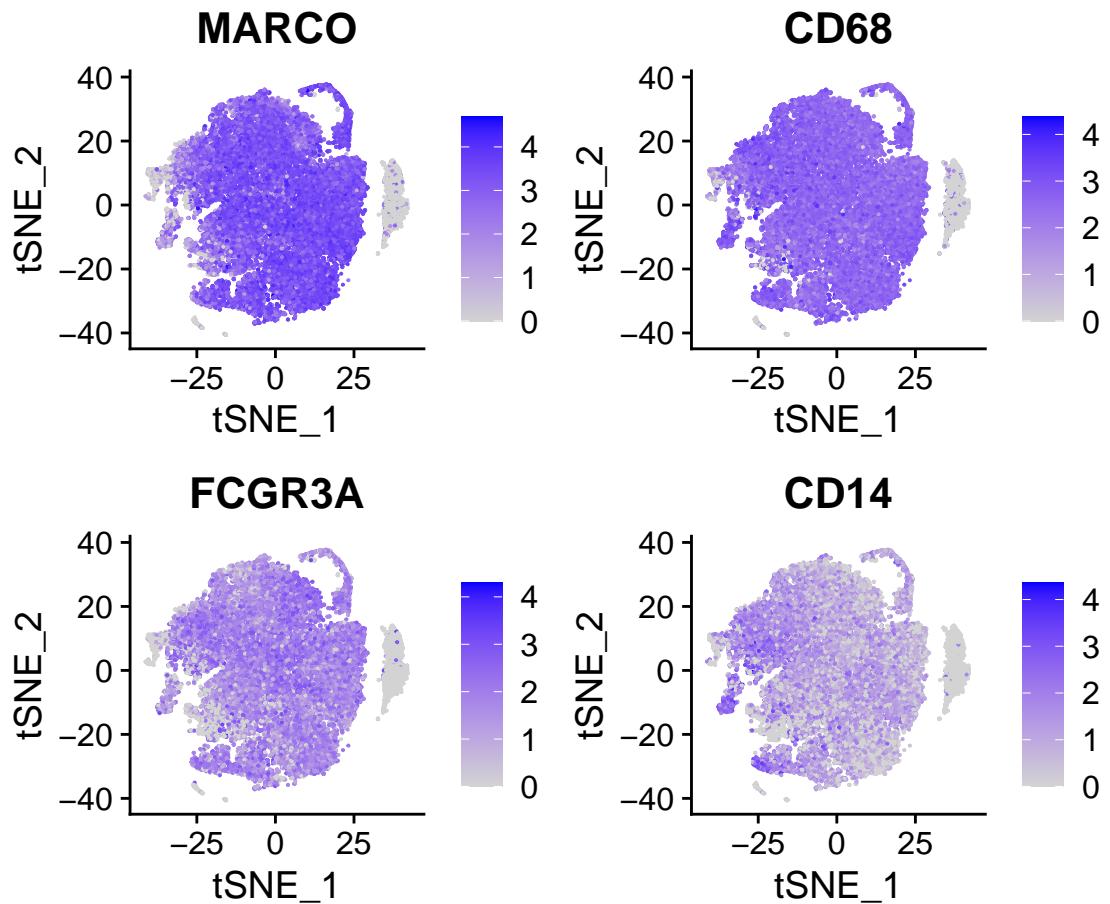
The cluster 6, 12 and 13 are

isolated clusters from other main clusters, indicating these cells are very different from the main cell type, thus susceptible to be contaminated cells.

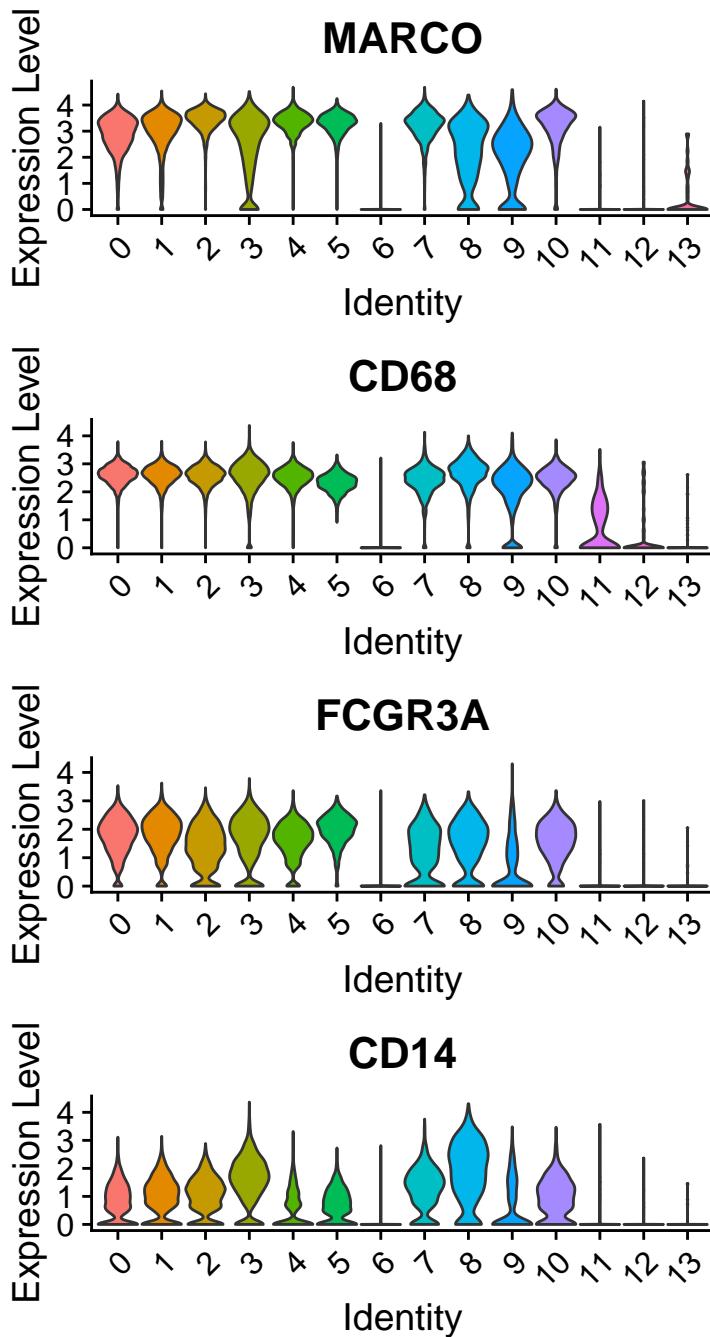
First check monocyte-macrophage markers:

```
FeaturePlot(results, features = c("MARCO", "CD68", "FCGR3A", "CD14"),
           reduction = "tsne")
```

1



```
VlnPlot(results, features = c("MARCO", "CD68", "FCGR3A", "CD14"), ncol = 1, pt.size = 0)
```

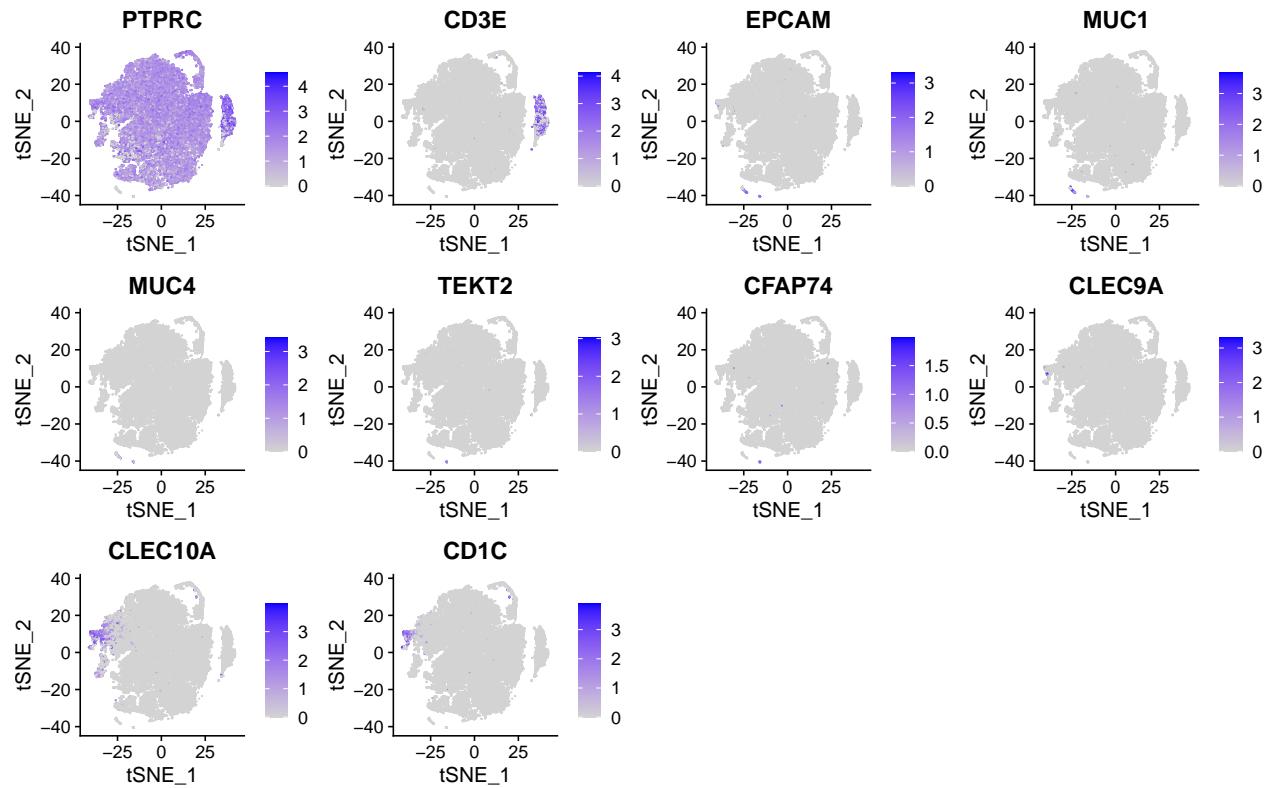


Then lineage markers:

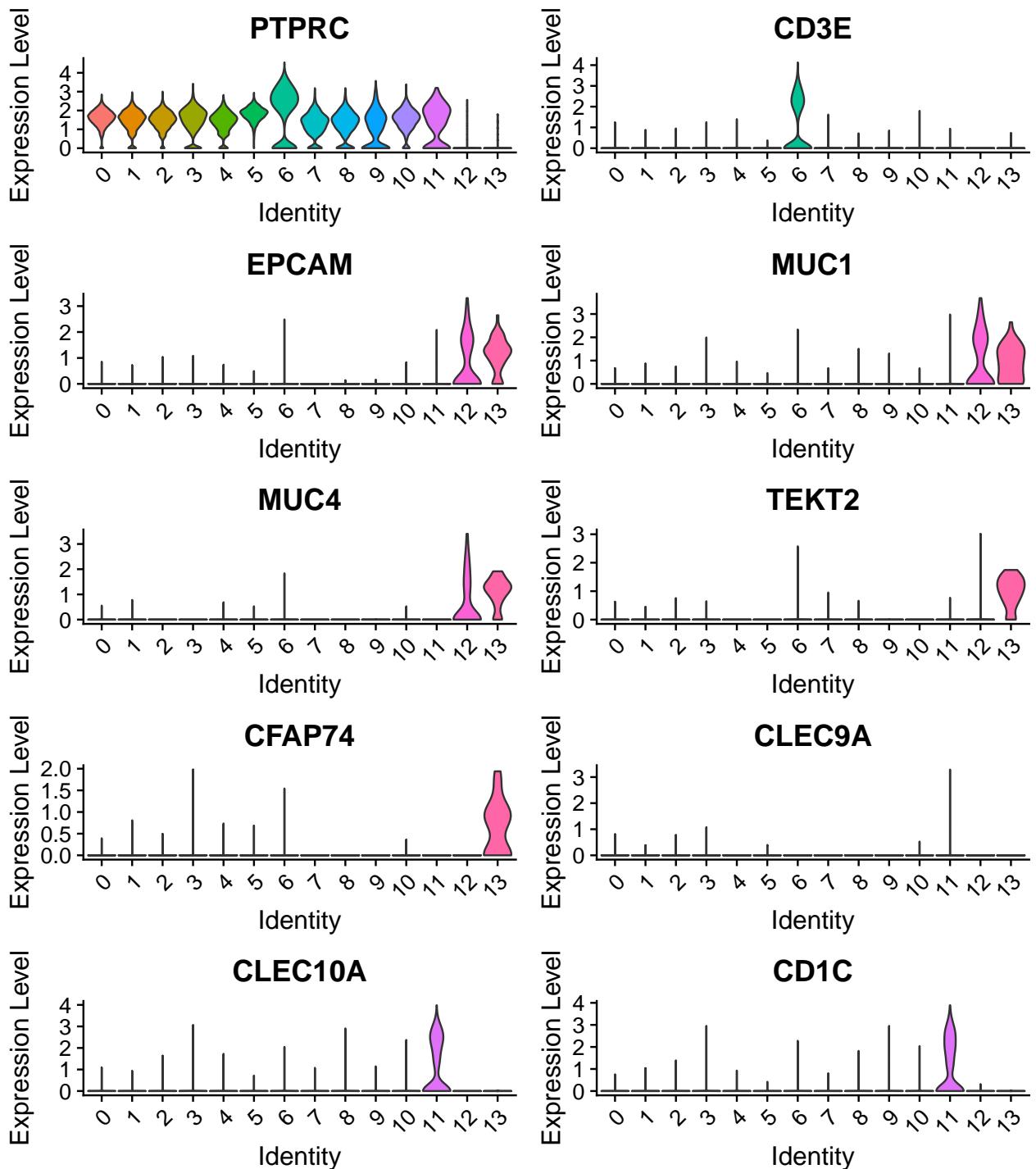
```

FeaturePlot(results, features = c("PTPRC", "CD3E",
                                 "EPCAM", "MUC1", "MUC4", # Epithelial
                                 cell markers
                                 "TEKT2", "CFAP74", # Mucosal epithelial
                                 cell markers
                                 "CLEC9A", "CLEC10A", "CD1C" # DC markers
                               ),
reduction = "tsne")

```



```
VlnPlot(results, features = c("PTPRC", "CD3E", "EPCAM", "MUC1", "MUC4", "TEKT2", "CFAP74", "CLEC9A", "CLEC10A", "CD1C"), ncol = 2, pt.size = 0) 1
                                                               2
```



Cluster 12 and 13 are not immune cells (CD45 lo). Cluster 12 should be mucosal epithelial cells (MUC1, MUC4, EPCAM). Cluster 13 should be ciliated cells (TEKT2, CFAP74, EPCAM). Cluster 11 is DC. But Cluster 11 is very close (high similarity) to the main monocyte-macrophage population. We should carefully investigate if they are true DCs or a subset of monocytes-macrophage population manifesting DC markers.

Remove contaminated populations: Cluster 6, 12 and 13.

```
results <- subset(results, idents = c(0:5, 7:11)) # here we leave cluster 11.
```

Let's keep old clustering information before filtering.

```
results$seurat_clusters_before_filter <- results$seurat_clusters
```

1

## 6 Normalization and redo cluster in filtered “monocytes”

### 6.1 Data processing in filtered “monocytes”

```
results <- NormalizeData(results, verbose = FALSE)
results <- FindVariableFeatures(results, selection.method = "vst",
                                nfeatures = 2000, verbose = FALSE)
results <- ScaleData(results, features = rownames(results))
```

1

2

3

```
## Centering and scaling data matrix
```

1

### 6.2 Dimension reduction in filtered “monocytes”

```
results <- RunPCA(results, features = VariableFeatures(results))
```

1

```
## PC_ 1
## Positive: MKI67, TOP2A, PCLAF, CDK1, TYMS, UBE2C, TPX2, BIRC5, ASPM,
## CENPF
## GTSE1, CDKN3, RRM2, HMMR, NCAPG, ANLN, NUSAP1, NDC80, CENPM, CKAP2L
## CEP55, KNL1, CCNB2, KIF11, DLGAP5, AURKB, CENPA, CCNA2, TK1, NUF2
## Negative: GPD1, GLRX, MSR1, PCOLCE2, INHBA, BCL2A1, CYP27A1, CD37, CD9
## , EVL
## ATP1B1, SERPING1, APOC1, CES1, LGALS3, AC026369.3, TFRC, PRDX1,
## ITM2B, TGM2
## SLC02B1, ALDH1A1, HBEGF, HSP90AB1, TPT1, ALDH2, MRC1, GCHFR, HCST,
## C1QA
## PC_ 2
## Positive: FCGR2B, CLEC10A, CORO1A, FGL2, EMP1, BASP1, MARCKS, CLEC5A,
## FPR3, PLEKH01
## LIMD2, ZFP36L1, PLXNC1, C15orf48, RASSF2, PLA2G7, CCL2, STAB1, IER3
## , IL4I1
## MATK, GPR183, RNASE1, PMP22, ARL4C, SPP1, TMEM176B, SGK1, CD48,
## ANTXR2
## Negative: FABP4, C1QA, C1QB, ALDH2, MARCO, SERPING1, SERPINA1,
## LGALS3BP, TREM1, CTSC
## CD52, APOC1, LDHB, GPD1, PCOLCE2, GCHFR, C1QC, CES1, PDLM1, ITM2B
## LGALS3, MCEMP1, MME, IGFBP2, ALOX5AP, CYP27A1, INHBA, ATP1B1,
## AC026369.3, PLBD1
## PC_ 3
## Positive: PRDX1, CSTB, LIPA, ACP5, GPNMB, FBP1, GSN, CAPG, CD9, ACTB
## TXN, CD63, NPC2, BCL2A1, CYBB, NRP2, TREM2, CTSB, HEXB, CTSD
## CYP1B1, PLA2G7, IGSF6, HSP90B1, CALR, ANXA2, FABP5, ASAHI, SQOR,
## LAMP1
## Negative: MTRNR2L12, ITIH5, LYZ, ZFAS1, TXNIP, AOC3, NET1, ANG, CFD,
## FABP4
## FCER1A, C8B, LAMB1, CAMP, PLAC8, RPS4X, PDGFD, HLA-C, LTB, SELENOM
## CD1C, SERPING1, PKIB, ICAM3, CD1E, ALDH2, DDIT4, CST7, TRHDE,
## GADD45B
```

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

```

## PC_ 4                                22
## Positive: NUPR1, APOE, CSTB, PLTP, APOC1, SPP1, CCL18, RARRES1, ABCA1,    23
  PLIN2
##          GPNMB, SOD2, CTSD, SCD, SQSTM1, PLD3, HS3ST2, NR1H3, CTSL, RBP4   24
##          MGST1, CYP27A1, ATP6VOD2, MGLL, PLPP3, MARCKS, SPARC, HES2, SLAMF7,  25
  KLHDC8B
## Negative: MRC1, S100A4, MCEMP1, ALOX5AP, FBP1, ACTB, AQP3, CYBB, RETN,   26
  ACTG1
##          PFN1, RGCC, YWHAH, S100A10, COTL1, STXBP2, MNDA, S100A8, ITGAM, GCA 27
##          HLA-DRB1, GSN, TGFB1, CLEC5A, CST3, MOB3B, VSIG4, VCAN, CORO1A,     28
  TXNIP
## PC_ 5                                29
## Positive: LGALS1, CFD, CTSB, CTSZ, C1QC, GLUL, ANXA2, PLTP, MS4A6A,      30
  S100A10
##          MGLL, EMP3, ABCA1, HLA-DRB1, CTSL, GRN, MARCKS, SCD, CCL18, ACP2  31
##          AHNAK, CD99, CD36, RNASE1, ALDH2, NR1H3, TXNIP, S100A4, C3AR1, LMNA 32
## Negative: TPT1, EIF1, CDKN1A, GDF15, DNASE2B, EVL, HLA-DRB5, MDM2,       33
  FAM213A, HLA-DQA2
##          CD9, TDRD3, PHPT1, DCSTAMP, RPS4X, SSBP3, BCL2A1, TREM2, AC022509 34
  .2, TGM2
##          ATP6VOD2, TMEM91, CRIM1, SYTL1, ZFAS1, GADD45A, CA2, RPS27L, IGSF6, 35
  CKLF

```

```

results <- RunUMAP(results, dims = 1:6, verbose = FALSE)                                1

```

### 6.3 Clustering filtered “monocytes”

```

results <- FindNeighbors(results, reduction = "pca", dims = 1:6)                            1

```

```

## Computing nearest neighbor graph                                                       1

```

```

## Computing SNN                                                               1

```

```

results <- FindClusters(results, resolution = 0.12)                                         1

```

```

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck 1
##                                                               2
## Number of nodes: 29827                                                 3
## Number of edges: 840369                                              4
##                                                               5
## Running Louvain algorithm...                                           6
## Maximum modularity in 10 random starts: 0.9153                         7
## Number of communities: 4                                              8
## Elapsed time: 4 seconds                                              9

```

See the cluster distribution by group:

```

library(ggplot2)                                                               1

```

```

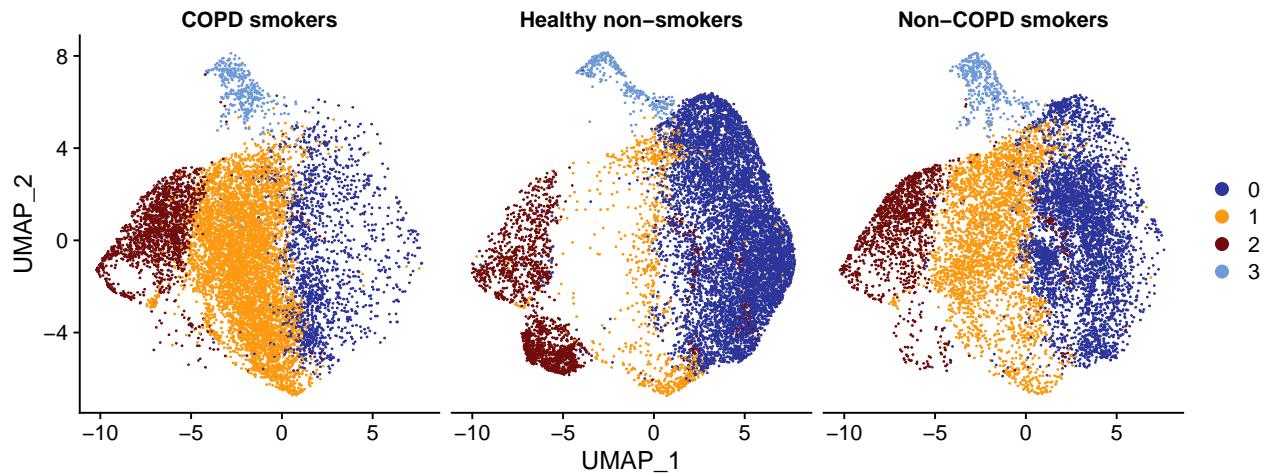
pal_4c <- c("#2E359A", "#FC990E", "#720DOD", "#6E9BD8") # new pal 20201214 2
  bis                                                               3

```

```

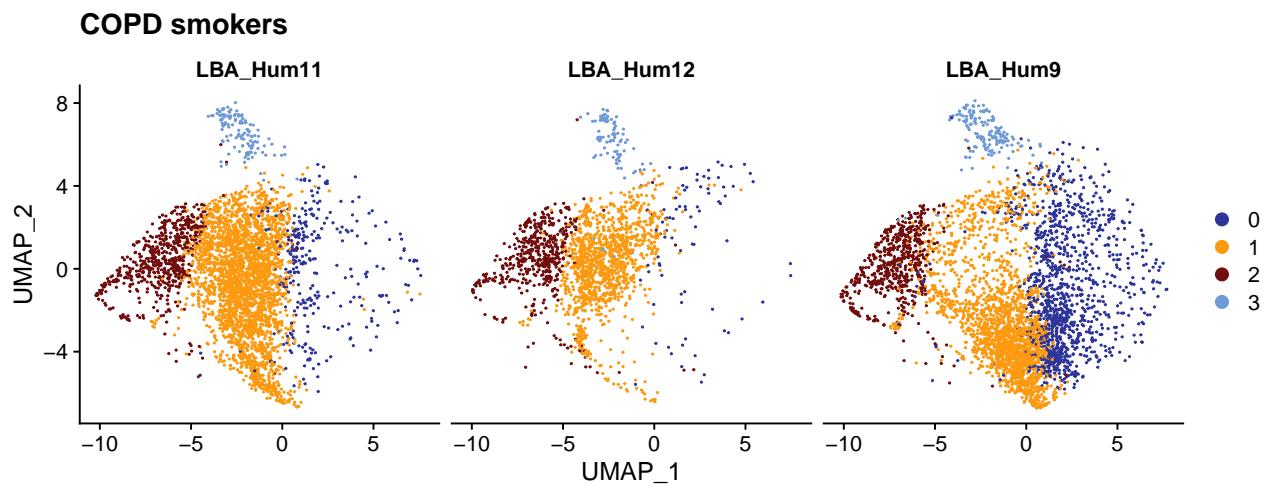
DimPlot(results, split.by = "group", cols = pal_4c)                           4

```



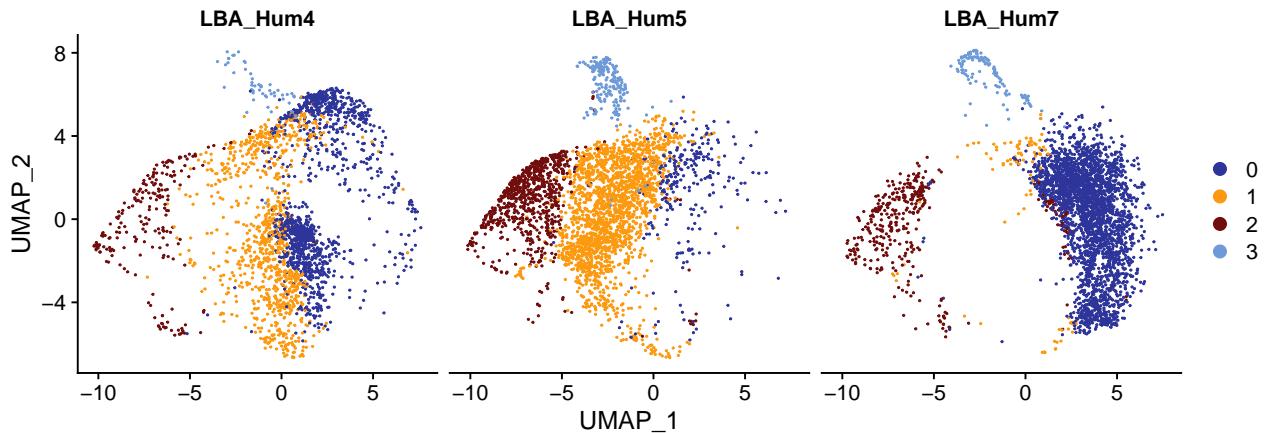
See the cluster distribution in samples (by group):

```
DimPlot(subset(results, subset = group == "COPD_smokers"),
        split.by = "origin",
        cols = pal_4c) + ggtitle("COPD_smokers")
```



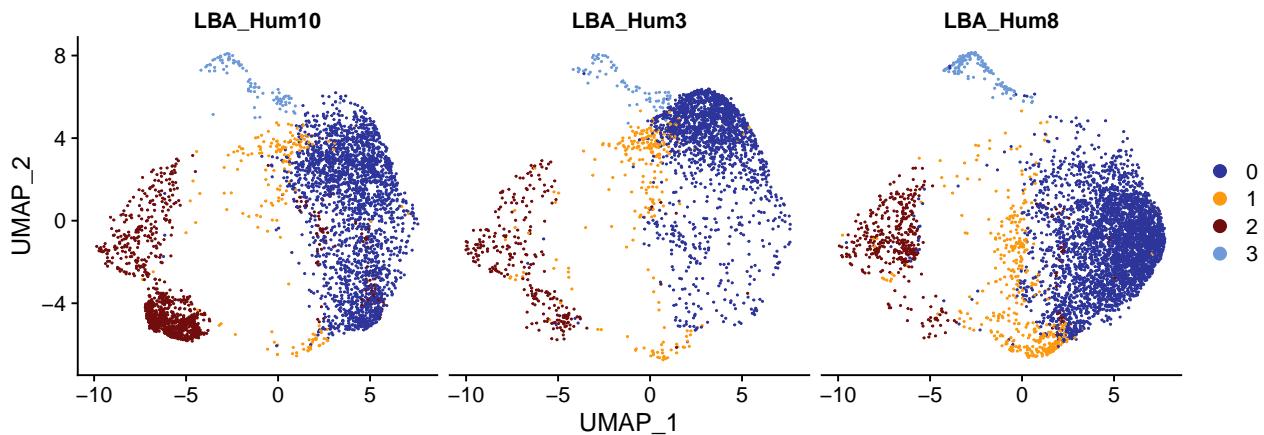
```
DimPlot(subset(results, subset = group == "Non-COPD_smokers"), split.by =
        "origin",
        cols = pal_4c) + ggtitle("Non-COPD_smokers")
```

### Non-COPD smokers



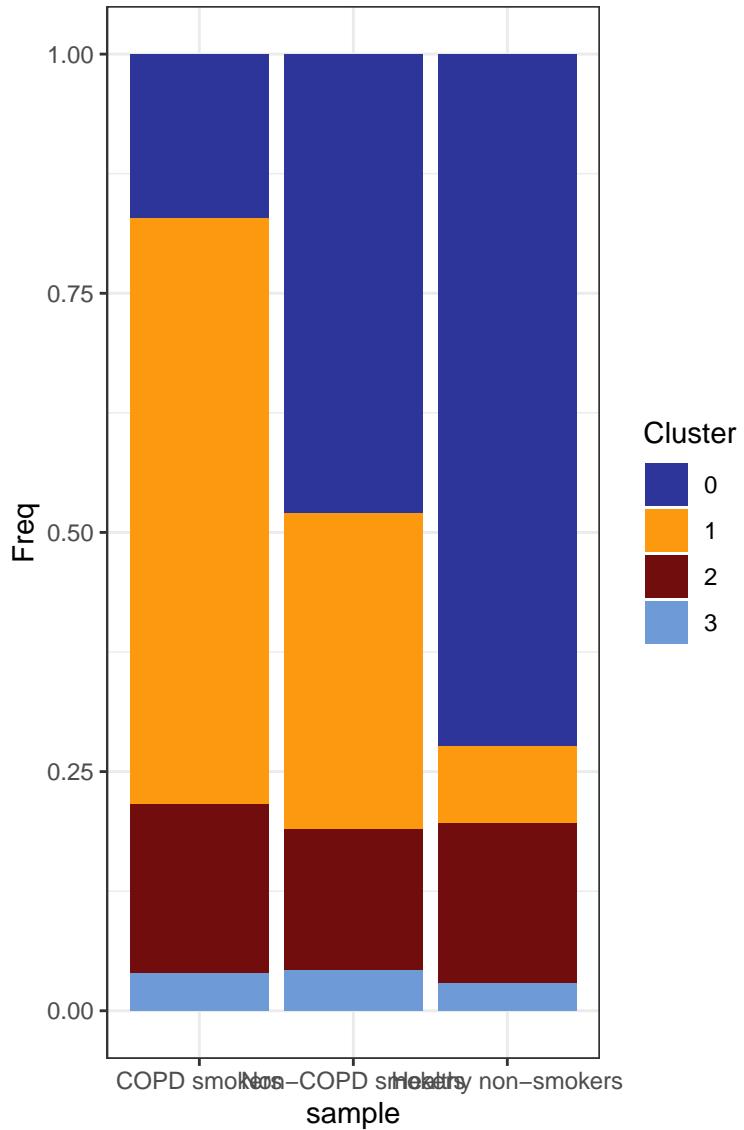
```
DimPlot(subset(results, subset = group == "Healthy\u2225non-smokers"), split.by = "origin",
        cols = pal_4c) + ggtitle("Healthy\u2225non-smokers")
```

### Healthy non-smokers



Show distribution of clusters in barplot:

```
source("../R/SeuratFreqTable.R")
freq.celltype.list <- list(
  `COPD smokers` = Seurat2CellFreqTable(subset(results, subset = group == "COPD\u2225smokers"),
                                          slotName = "RNA_snn_res.0.12"),
  `Non-COPD smokers` = Seurat2CellFreqTable(subset(results, subset = group == "Non-COPD\u2225smokers"),
                                              slotName = "RNA_snn_res.0.12"),
  `Healthy non-smokers` = Seurat2CellFreqTable(subset(results, subset = group == "Healthy\u2225non-smokers"),
                                                 slotName = "RNA_snn_res.0.12"))
)
source("../R/barChart.R")
barChart(freq.celltype.list) + labs(fill = "Cluster") + scale_fill_manual(values = pal_4c)
```



See cluster distribution in each samples:

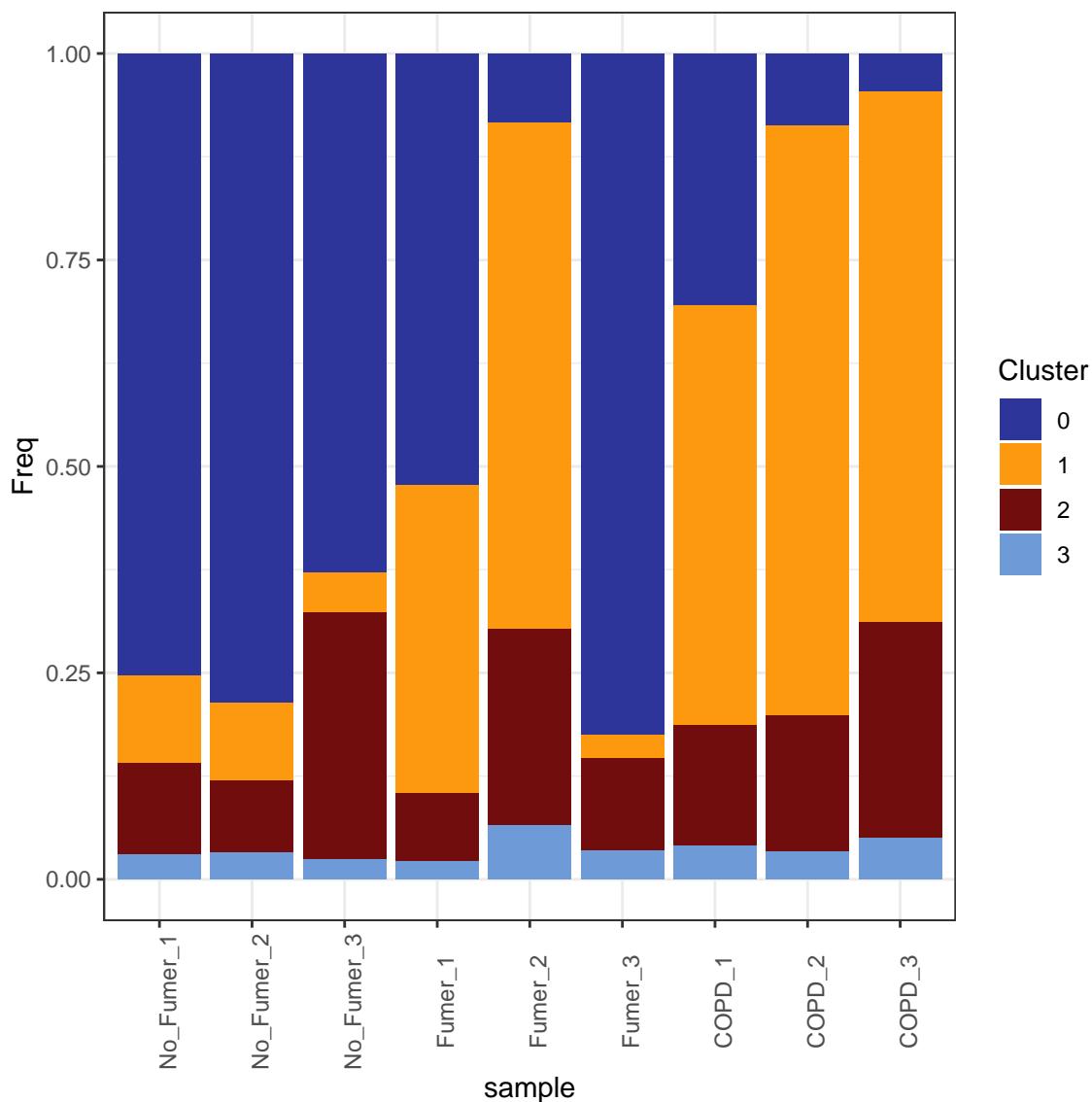
```

freq.celltype.list <- list(
  No_Fumer_1 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_Hum3"),
                                     slotName = "RNA_snn_res.0.12"),
  No_Fumer_2 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_Hum8"),
                                     slotName = "RNA_snn_res.0.12"),
  No_Fumer_3 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_Hum10"),
                                     slotName = "RNA_snn_res.0.12"),
  Fumer_1 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_Hum4"),
                                 slotName = "RNA_snn_res.0.12"),
  Fumer_2 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_Hum5"),
                                 slotName = "RNA_snn_res.0.12"),
  Fumer_3 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_Hum6"),
                                 slotName = "RNA_snn_res.0.12")
)
  
```

```

        Hum7" ),
                           slotName = "RNA_snn_res.0.12"),
COPD_1 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_ 13
                           Hum9"),
                           slotName = "RNA_snn_res.0.12"),
COPD_2 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_ 14
                           Hum11"),
                           slotName = "RNA_snn_res.0.12"),
COPD_3 = Seurat2CellFreqTable(subset(results, subset = origin == "LBA_ 15
                           Hum12"),
                           slotName = "RNA_snn_res.0.12"),
) 16
                           slotName = "RNA_snn_res.0.12") 17
) 18
                           slotName = "RNA_snn_res.0.12") 19
) 20
                           slotName = "RNA_snn_res.0.12") 21
) 22
barChart(freq.celltype.list) + labs(fill = "Cluster") + 23
scale_fill_manual(values = pal_4c) +
theme(axis.text.x = element_text(angle = 90)) 24

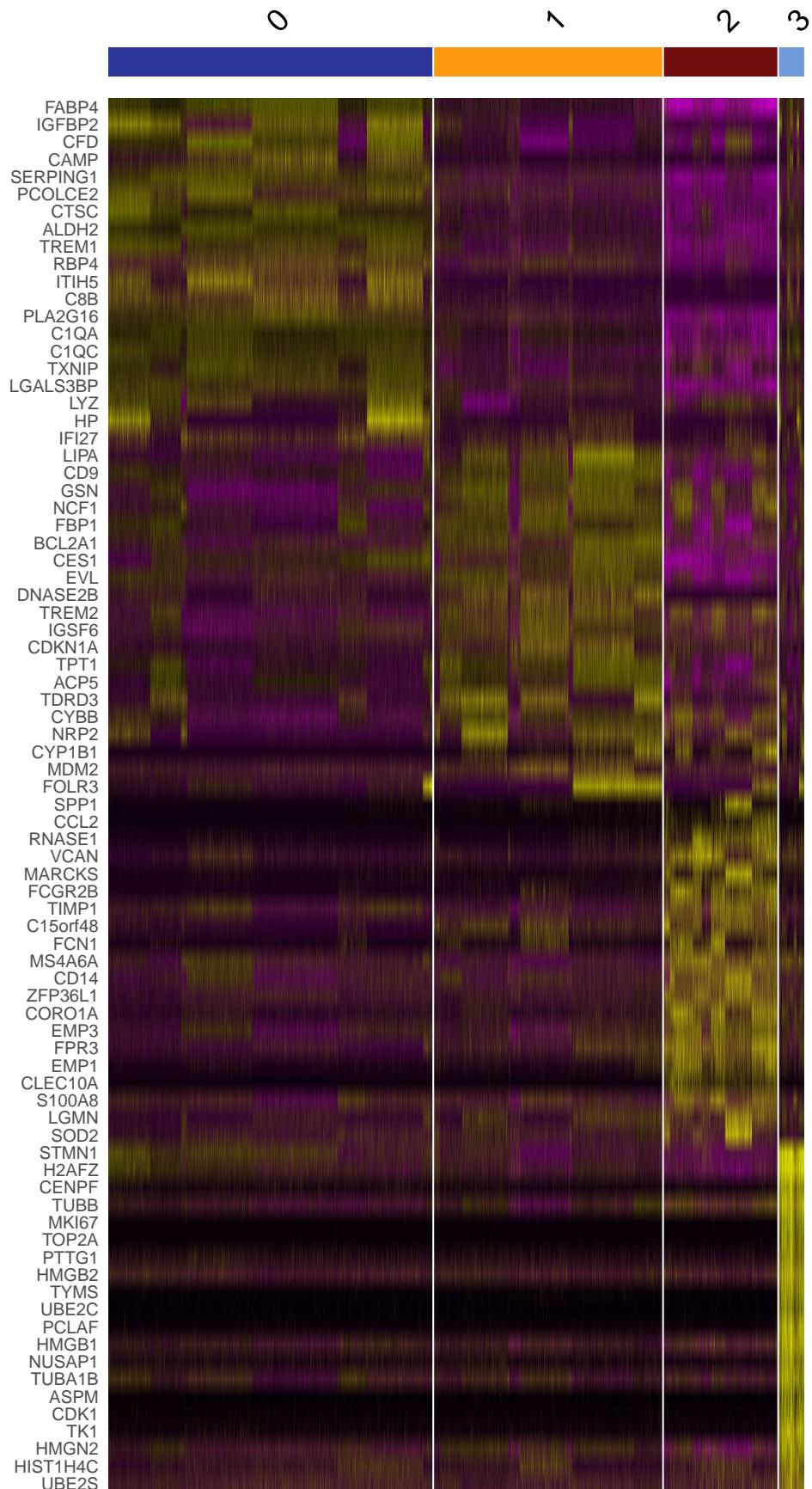
```



## 7 Characterization of clustered populations

See the markers of each cluster:

```
all_cluster.markers <- FindAllMarkers(results)                                1  
  
suppressMessages(library(dplyr))                                                 1  
top20 <- all_cluster.markers %>% group_by(cluster) %>% top_n(n = 20, wt = 2  
    avg_log2FC)  
DoHeatmap(results, features = top20$gene, group.colors = pal_4c) +            3  
    NoLegend()
```



## 8 Cell-cycle analysis

```

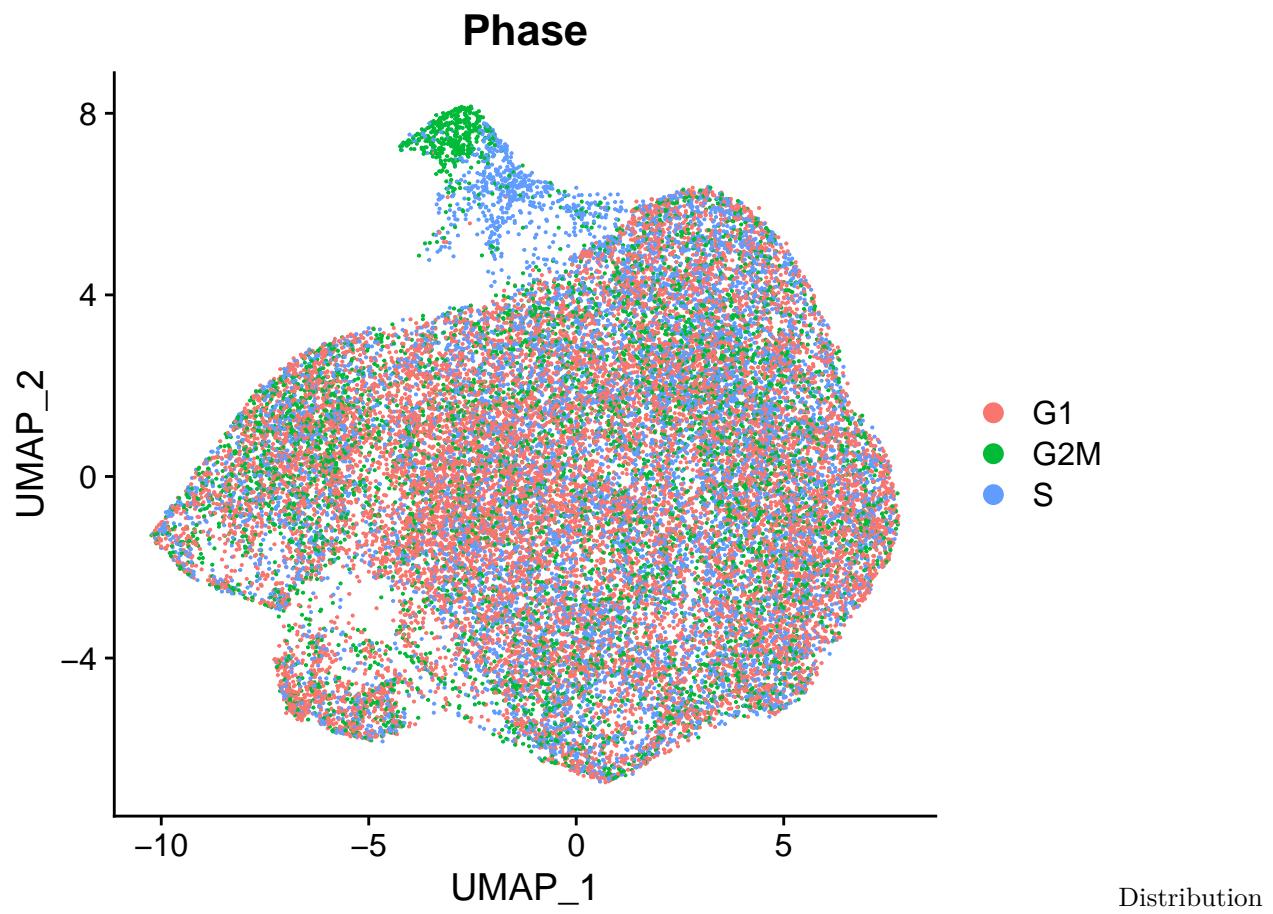
s.genes <- cc.genes$s.genes
g2m.genes <- cc.genes$g2m.genes
results <- CellCycleScoring(results, s.features = s.genes, g2m.features =
  g2m.genes, set.ident = FALSE)
1
2
3

## Warning: The following features are not present in the object: MLF1IP ,
not
## searching for symbol synonyms
1
2

## Warning: The following features are not present in the object: FAM64A ,
HN1 , not
## searching for symbol synonyms
1
2

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

```



```

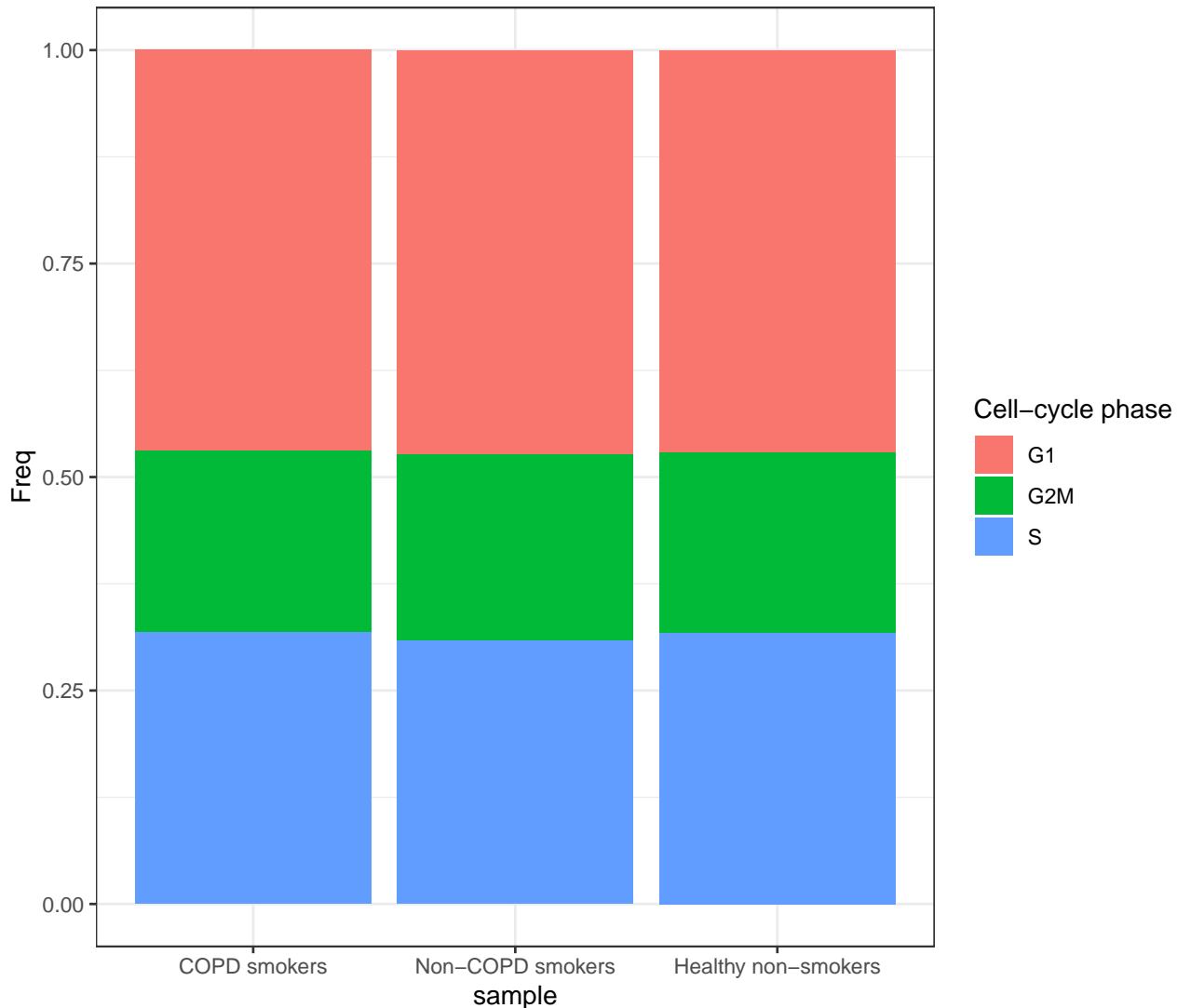
freq.celltype.list <- list(
  `COPD smokers` = Seurat2CellFreqTable(subset(results, subset = group ==
    "COPD smokers"), slotName = "Phase"),
  `Non-COPD smokers` = Seurat2CellFreqTable(subset(results, subset = group ==
    "Non-COPD smokers"), slotName = "Phase"),
1
2
3

```

```

`Healthy non-smokers` = Seurat2CellFreqTable(subset(results, subset =
  group == "Healthy non-smokers"), slotName = "Phase")
)
barChart(freq.celltype.list) + labs(fill = "Cell-cycle phase")

```



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

## 9 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    7
##
## locale:                                         8
##   [1] LC_CTYPE=en_US.UTF-8           LC_NUMERIC=C          9
##   [3] LC_TIME=en_GB.UTF-8          LC_COLLATE=en_US.UTF-8 10
##   [5] LC_MONETARY=en_GB.UTF-8      LC_MESSAGES=en_US.UTF-8 11
##   [7] LC_PAPER=en_GB.UTF-8         LC_NAME=C            12
##   [9] LC_ADDRESS=C                 LC_TELEPHONE=C       13
##  [11] LC_MEASUREMENT=en_GB.UTF-8  LC_IDENTIFICATION=C 14
##
## attached base packages:                      15
## [1] stats      graphics   grDevices  utils      datasets   methods    base 16
##
## other attached packages:                     17
## [1] dplyr_1.0.8        RColorBrewer_1.1-2  ggplot2_3.3.5 20
##     SeuratObject_4.0.4
## [5] Seurat_4.1.0
##
## loaded via a namespace (and not attached): 21
## [1] ggbeeswarm_0.6.0      Rtsne_0.15          colorspace_2.0-3 24
## [4] deldir_1.0-6          ellipsis_0.3.2     ggridges_0.5.3    25
## [7] rstudioapi_0.13       spatstat.data_2.1-2 leiden_0.3.9     26
## [10] listenv_0.8.0         farver_2.1.0       ggrepel_0.9.1    27
## [13] RSpectra_0.16-0       fansi_1.0.2        codetools_0.2-18 28
## [16] splines_4.0.3         knitr_1.37         polyclip_1.10-0  29
## [19] jsonlite_1.7.3       ica_1.0-2          cluster_2.1.0   30
## [22] png_0.1-7            uwot_0.1.11        shiny_1.7.1     31
## [25] sctransform_0.3.3    spatstat.sparse_2.1-0 compiler_4.0.3 32
## [28] httr_1.4.2           assertthat_0.2.1   Matrix_1.4-0    33
## [31] fastmap_1.1.0         lazyeval_0.2.2     limma_3.46.0    34
## [34] cli_3.2.0             later_1.3.0        htmltools_0.5.2 35
## [37] tools_4.0.3           igraph_1.2.11      gtable_0.3.0   36
## [40] glue_1.6.1            RANN_2.6.1         reshape2_1.4.4 37
## [43] Rcpp_1.0.8             scattermore_0.8    vctrs_0.3.8    38
## [46] nlme_3.1-155          lmtest_0.9-39      spatstat.random_2.1-0 39
## [49] xfun_0.29              stringr_1.4.0      globals_0.14.0 40
## [52] mime_0.12              miniUI_0.1.1.1    lifecycle_1.0.1 41
## [55] irlba_2.3.5           goftest_1.2-3      future_1.24.0 42
## [58] MASS_7.3-53            zoo_1.8-9          scales_1.1.1   43
## [61] spatstat.core_2.4-0    promises_1.2.0.1  spatstat.utils_2.3-0 44
## [64] parallel_4.0.3         yaml_2.3.5         reticulate_1.24 45
## [67] pbapply_1.5-0          gridExtra_2.3      ggrastr_1.0.1   46
## [70] rpart_4.1-15           stringi_1.7.6     highr_0.9     47
## [73] rlang_1.0.1            pkgconfig_2.0.3   matrixStats_0.61.0 48
## [76] evaluate_0.15          lattice_0.20-41  ROCR_1.0-11    49
## [79] purrr_0.3.4             tensor_1.5        patchwork_1.1.1 50
## [82] htmlwidgets_1.5.4      labeling_0.4.2    cowplot_1.1.1  51
## [85] tidyselect_1.1.1       parallely_1.30.0  RcppAnnoy_0.0.19 52
## [88] plyr_1.8.6              magrittr_2.0.2    R6_2.5.1      53
## [91] generics_0.1.2          DBI_1.1.2        withr_2.4.3    54
## [94] pillar_1.7.0            mgcv_1.8-33      fitdistrplus_1.1-6 55
## [97] survival_3.2-7          abind_1.4-5       tibble_3.1.6    56
## [100] future.apply_1.8.1     crayon_1.5.0     KernSmooth_2.23-20 57
## [103] utf8_1.2.2              spatstat.geom_2.3-2 plotly_4.10.0   58

```

## [106] rmarkdown_2.11	grid_4.0.3	data.table_1.14.2	60
## [109] digest_0.6.29	xtable_1.8-4	tidyR_1.2.0	61
## [112] httpuv_1.6.5	msnse11_0.5.0	beeswarm_0.4.0	62
## [115] viridisLite_0.4.0	vipor_0.4.5		63