Integrate 10x Samples Low filtering

Joan Abinet

2024-12-12 16:23:21 +0100

Contents

Load Packages

```
suppressMessages(library(dplyr))
suppressMessages(library(Seurat))
suppressMessages(library(patchwork))
suppressMessages(library(ggplot2))
suppressMessages(library(stringr))
suppressMessages(library(dittoSeq))
suppressMessages(library(ComplexHeatmap))
```

Loading data

```
all_dirs <- dir(path = "../10x-1-Pre_Processing/Data",
    full.names = T)

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

    Seq_raw_file <- Read10X(data.dir = paste0(all_dirs[i]))
    Seurat_file <- CreateSeuratObject(counts = Seq_raw_file,
        project = str_sub(all_dirs[i], -8, -1), min.cells = 3,
        min.features = 100)

    Seurat_file[["percent.mt"]] <- PercentageFeatureSet(Seurat_file,
        pattern = "^MT-")
    Seurat_file <- subset(Seurat_file, subset = nFeature_RNA <
        8000 & percent.mt < 20)  # or 5

    list_sample <- append(list_sample, Seurat_file)
}
```

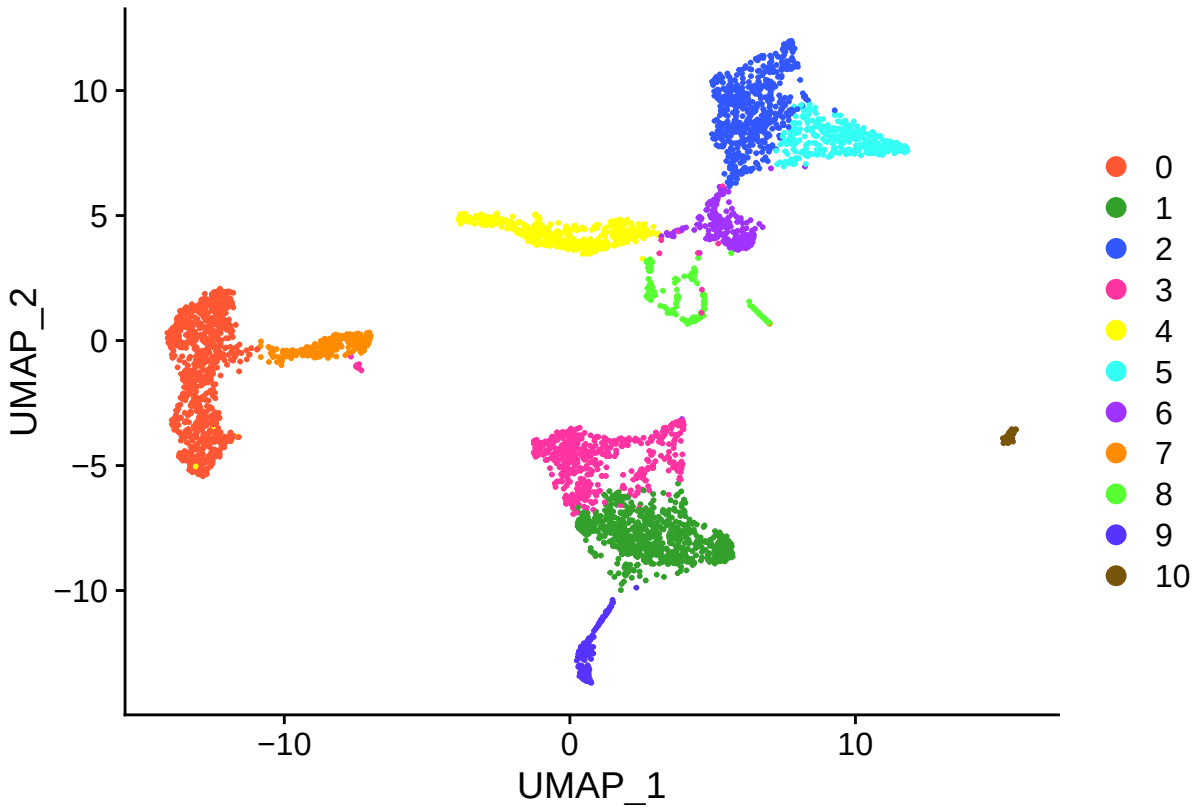
```
list_sample <- lapply(list_sample, function(x) {
    x <- NormalizeData(x, verbose = F)
    x <- FindVariableFeatures(x, selection.method = "vst",
        nfeatures = 2000, verbose = F)
})
```

```
features <- SelectIntegrationFeatures(list_sample,
    verbose = F)

list_sample <- lapply(list_sample, function(x) {
    x <- ScaleData(x, features = features, verbose = F)
    x <- RunPCA(x, features = features, verbose = F)
})
```

```
BAL.anchors <- FindIntegrationAnchors(object.list = list_sample,
    anchor.features = features, reduction = "rpca")

BAL_10x.integrated <- IntegrateData(anchorset = BAL.anchors)
```

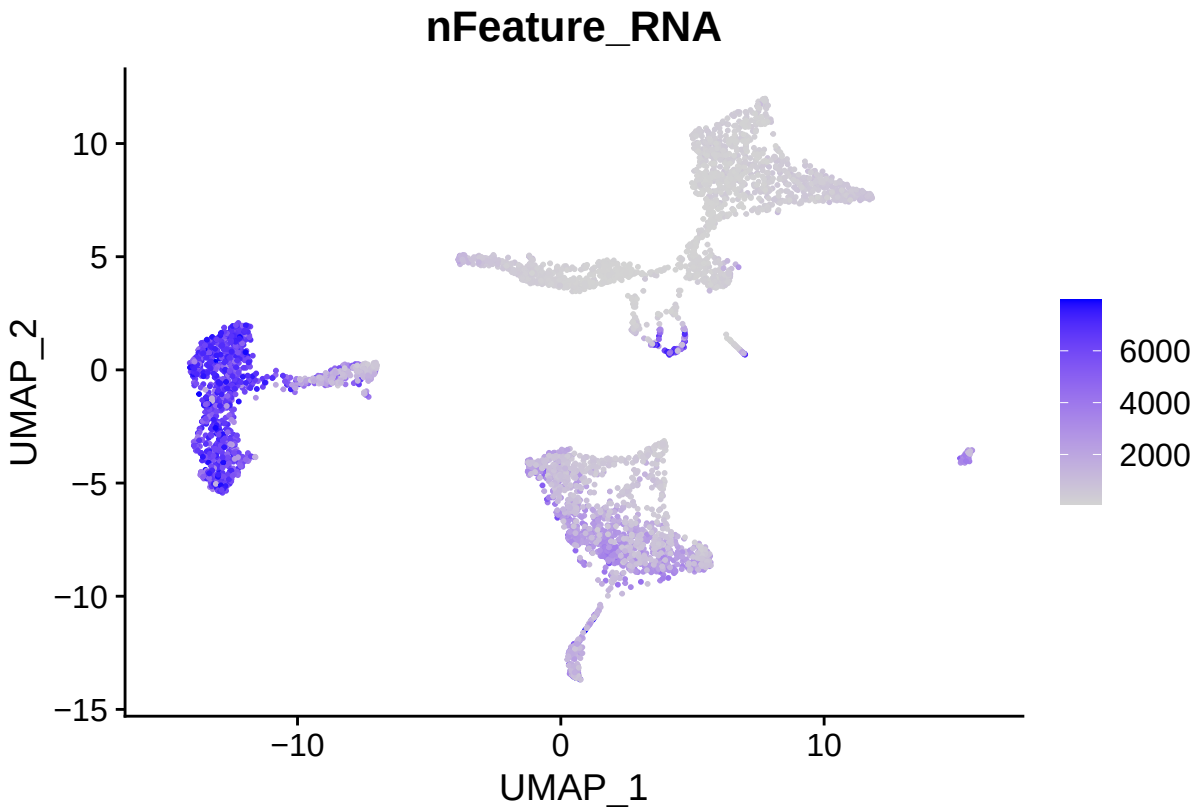
```
DefaultAssay(BAL_10x.integrated) <- "integrated"

# Run the standard workflow for visualization and
# clustering
BAL_10x.integrated <- ScaleData(BAL_10x.integrated,
    verbose = FALSE)
BAL_10x.integrated <- RunPCA(BAL_10x.integrated, npcs = 30,
    verbose = FALSE)
BAL_10x.integrated <- RunUMAP(BAL_10x.integrated, reduction = "pca",
    dims = 1:15)
BAL_10x.integrated <- FindNeighbors(BAL_10x.integrated,
    reduction = "pca", dims = 1:15)
BAL_10x.integrated <- FindClusters(BAL_10x.integrated,
    resolution = 0.35)
```

```
color_pallette <- c("#FF5733", "#33A02C", "#3357FF",
    "#FF33A1", "yellow", "#33FFF5", "#A133FF", "#FF8C00",
    "#57FF33", "#5733FF", "#78550b")

DimPlot(BAL_10x.integrated, cols = color_pallette)
```



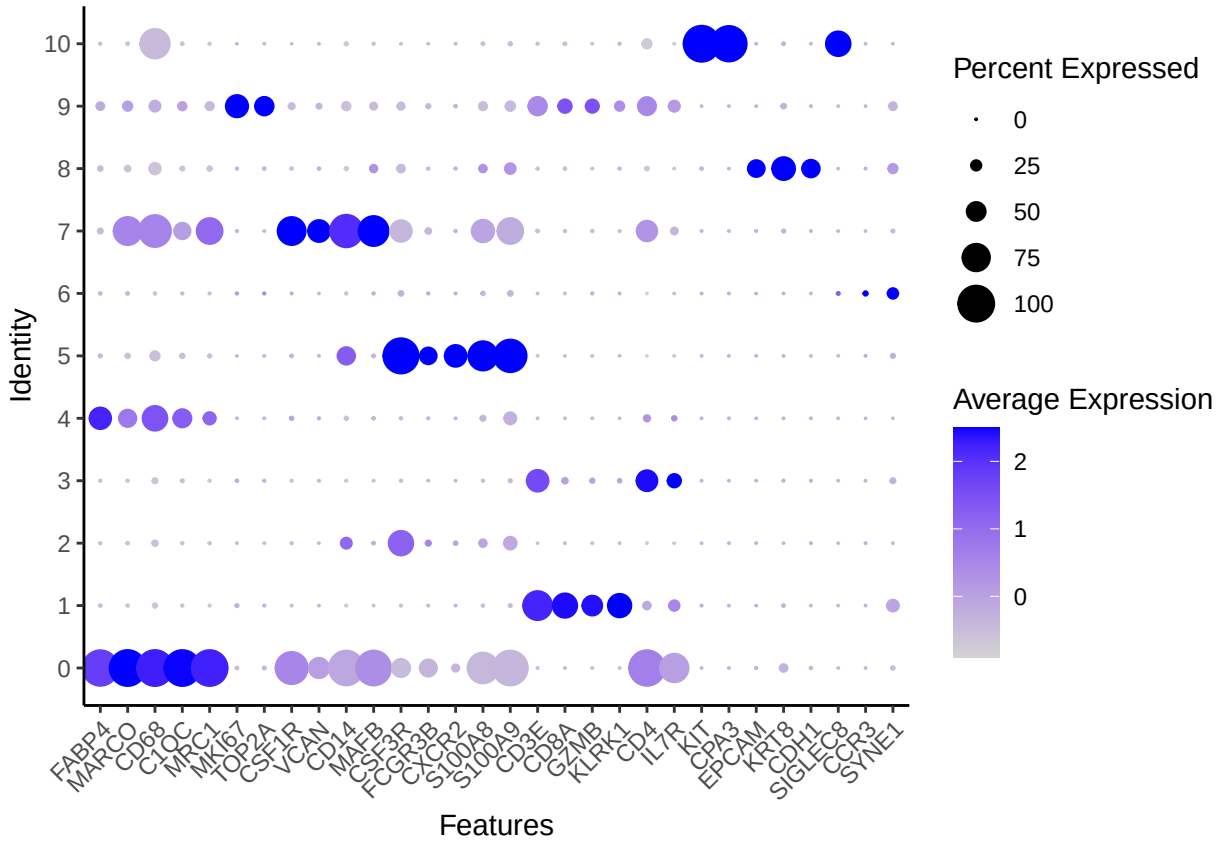

```
# ggsave('umap10x_lowQC.pdf', width = 8, height =
# 4)
```

```
FeaturePlot(BAL_10x.integrated, features = "nFeature_RNA")
```



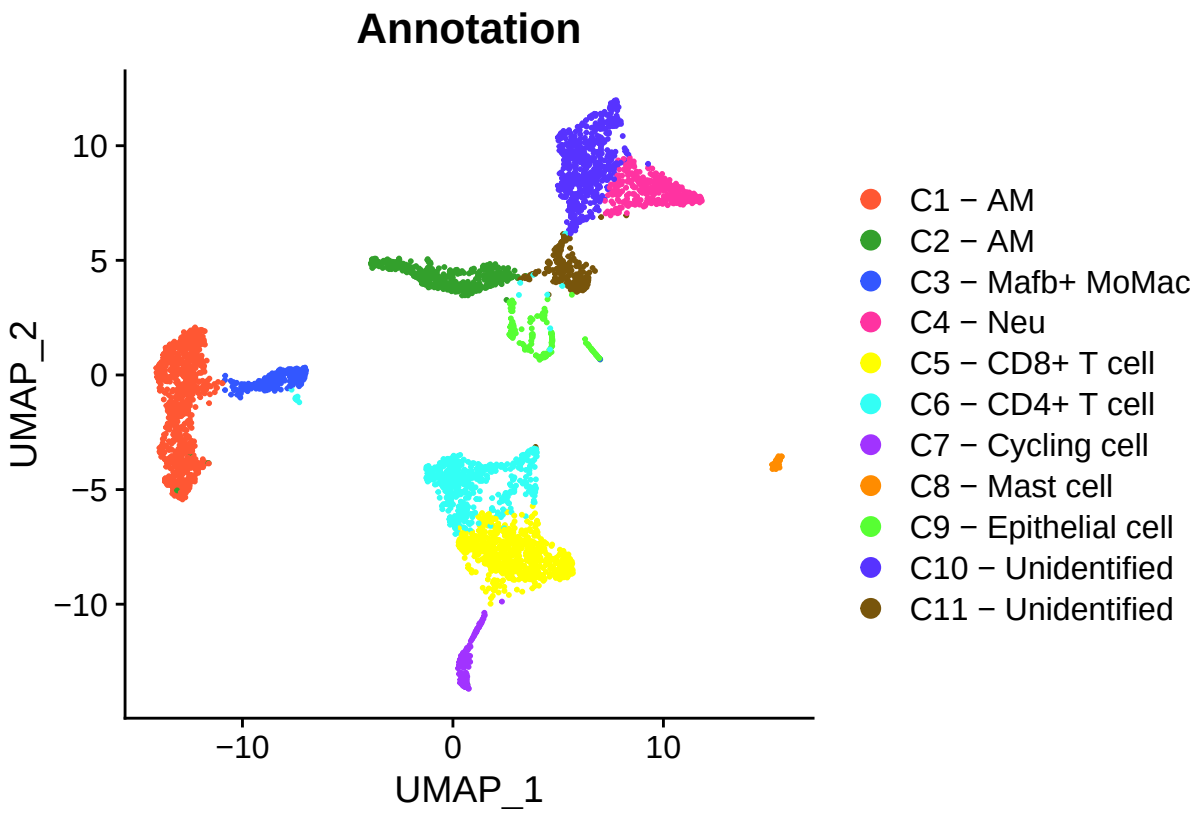

```
# ggsave('10x_lowQc_ngene.pdf', width = 8, height
# = 4)
```

```
DefaultAssay(BAL_10x.integrated) <- "RNA"
Marker_gene <- c("FABP4", "MARCO", "CD68", "C1QC",
    "MRC1", "MKI67", "TOP2A", "CSF1R", "VCAN", "CD14",
    "MAFB", "CSF3R", "FCGR3B", "CXCR2", "S100A8", "S100A9",
    "CD3E", "CD8A", "GZMB", "KLGR1", "KLRK1", "CD4",
    "IL7R", "KIT", "CPA3", "EPCAM", "KRT8", "CDH1",
    "SIGLEC8", "CCR3", "SYNE1")

DotPlot(BAL_10x.integrated, features = Marker_gene) +
    theme_classic() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1))
```



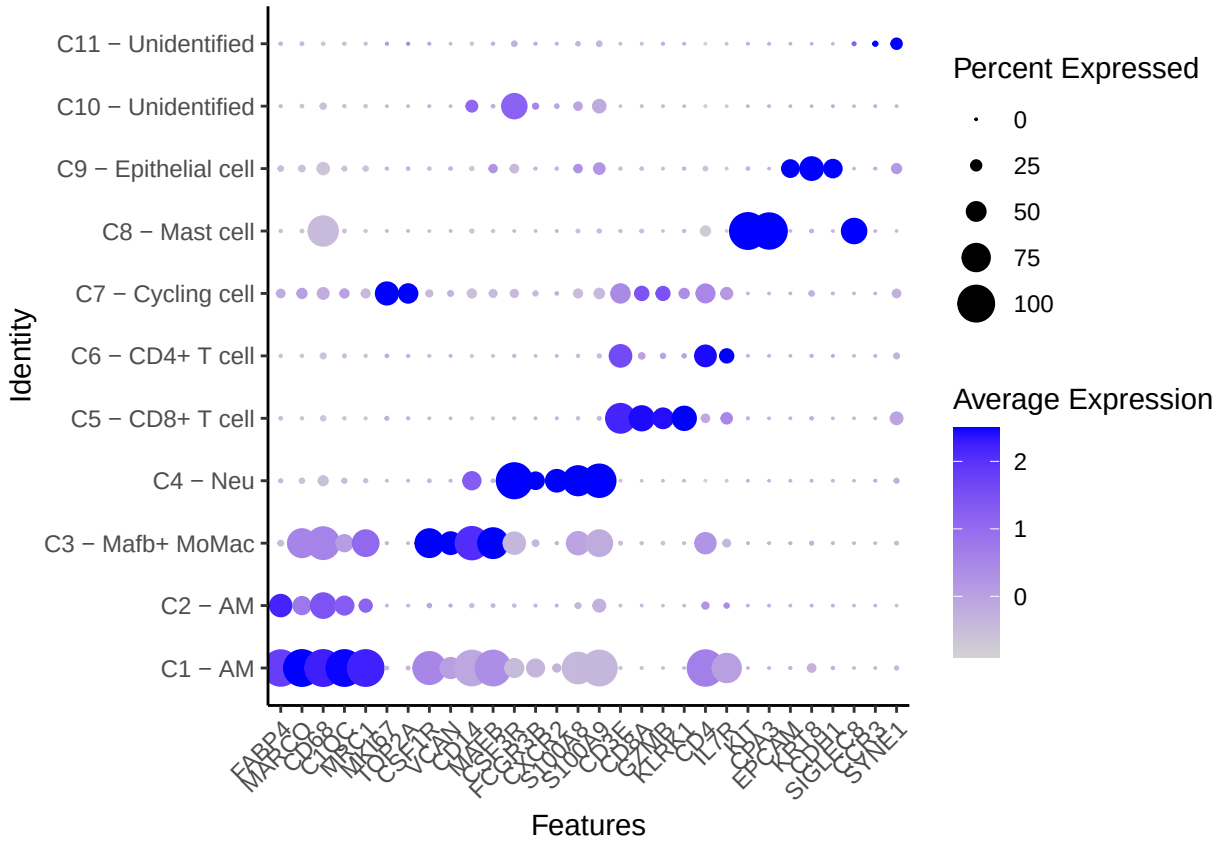

```
# ggsave('Dotplot_10X_LowQC_nonAnnotated.pdf',
# width = 35, height = 10)
```

```
BAL_10x.integrated$Annotation <- BAL_10x.integrated$seurat_clusters
BAL_10x.integrated$Annotation <- as.factor(BAL_10x.integrated$Annotation)

levels(BAL_10x.integrated$Annotation) <- c("C1 - AM",
    "C5 - CD8+ T cell", "C10 - Unidentified", "C6 - CD4+ T cell",
    "C2 - AM", "C4 - Neu", "C11 - Unidentified", "C3 - Mafb+ MoMac",
    "C9 - Epithelial cell", "C7 - Cycling cell", "C8 - Mast cell")

BAL_10x.integrated$Annotation <- factor(BAL_10x.integrated$Annotation,
    levels = c("C1 - AM", "C2 - AM", "C3 - Mafb+ MoMac",
        "C4 - Neu", "C5 - CD8+ T cell", "C6 - CD4+ T cell",
        "C7 - Cycling cell", "C8 - Mast cell", "C9 - Epithelial cell",
        "C10 - Unidentified", "C11 - Unidentified"))

color_pallette <- c("#FF5733", "#33A02C", "#3357FF",
    "#FF33A1", "yellow", "#33FFF5", "#A133FF", "#FF8C00",
    "#57FF33", "#5733FF", "#78550b")

DimPlot(BAL_10x.integrated, group.by = "Annotation",
    cols = color_pallette)
```




```
# ggsave('Umap_LowQC_Annotation.pdf', width = 8,
# height = 4)
```

```
DefaultAssay(BAL_10x.integrated) <- "RNA"
Marker_gene <- c("FABP4", "MARCO", "CD68", "C1QC",
    "MRC1", "MKI67", "TOP2A", "CSF1R", "VCAN", "CD14",
    "MAFB", "CSF3R", "FCGR3B", "CXCR2", "S100A8", "S100A9",
    "CD3E", "CD8A", "GZMB", "KLGR1", "KLRK1", "CD4",
    "IL7R", "KIT", "CPA3", "EPCAM", "KRT8", "CDH1",
    "SIGLEC8", "CCR3", "SYNE1")

DotPlot(BAL_10x.integrated, features = Marker_gene,
    group.by = "Annotation") + theme_classic() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1))
```



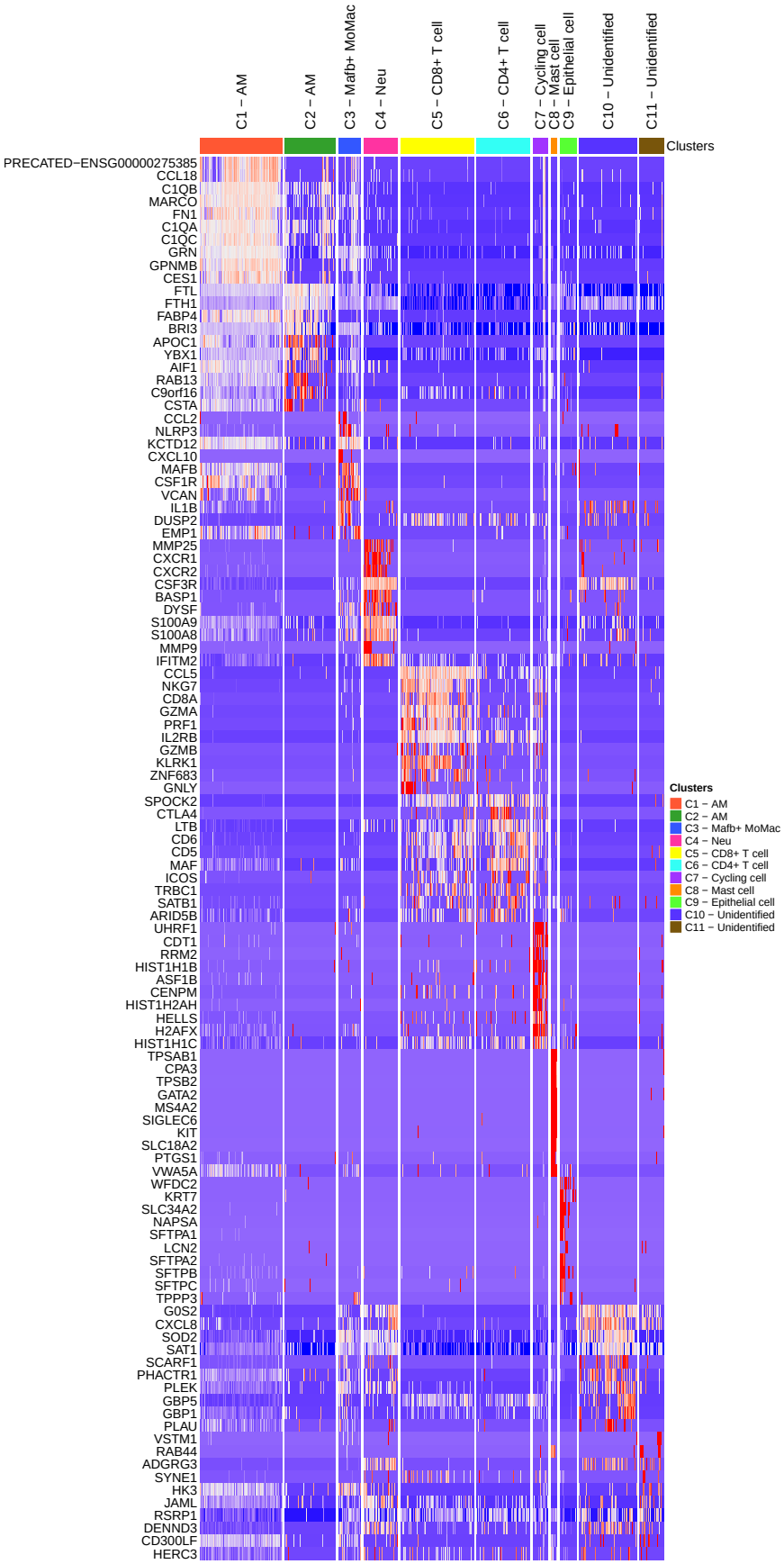

```
# ggsave('Dotplot_10X_LowQC_Annotated.pdf', width
# = 35, height = 10)
```

```
Idents(BAL_10x.integrated) <- "Annotation"

lavage.markers <- FindAllMarkers(BAL_10x.integrated,
    only.pos = TRUE, min.pct = 0.25)

lavage.markers %>%
    group_by(cluster) %>%
    top_n(n = 10, wt = avg_log2FC) -> top10

mat <- as.matrix(GetAssayData(object = BAL_10x.integrated,
    slot = "data")[as.character(top10$gene), ])

df <- as.data.frame(BAL_10x.integrated$Annotation)
colnames(df) <- "Clusters"
color_df <- list(Clusters = c(`C1 - AM` = "#FF5733",
    `C2 - AM` = "#33A02C", `C3 - Mafb+ MoMac` = "#3357FF",
    `C4 - Neu` = "#FF33A1", `C5 - CD8+ T cell` = "yellow",
    `C6 - CD4+ T cell` = "#33FFF5", `C7 - Cycling cell` = "#A133FF",
    `C8 - Mast cell` = "#FF8C00", `C9 - Epithelial cell` = "#57FF33",
    `C10 - Unidentified` = "#5733FF", `C11 - Unidentified` = "#78550b"))

Heatmap <- Heatmap(t(scale(t(mat))), show_column_names = F,
    column_split = BAL_10x.integrated$Annotation, cluster_column_slices = F,
    cluster_rows = F, top_annotation = HeatmapAnnotation(df = df,
```

```
        col = color_df), use_raster = F, show_heatmap_legend = F,
    show_column_dend = F, column_title_rot = 90, row_names_side = "left")

# tidyHeatmap::save_pdf(Heatmap,
# 'Heatmap_10x_low.pdf', width = 35, height = 50,
# units ='cm')
```

Heatmap top genes per clusters

```
Heatmap
```


C1 – AM
C2 – AM
C3 – Mafb+ MoMac
C4 – Neu
C5 – CD8+ T cell
C6 – CD4+ T cell
C7 – Cycling cell
C8 – Mast cell
C9 – Epithelial cell
C10 – Unidentified
C11 – Unidentified
Clusters
PRECATED−ENSG00000275385
CCL18
C1QB
MARCO
FN1
C1QA
C1QC
GRN
GPNMB
CES1
FTL
FTH1
FABP4
BRI3
APOC1
YBX1
AIF1
RAB13
C9orf16
CSTA
CCL2
NLRP3
KCTD12
CXCL10
MAFB
CSF1R
VCAN
IL1B
DUSP2
EMP1
MMP25
CXCR1
CXCR2
CSF3R
BASP1
DYSF
S100A9
S100A8
MMP9
IFITM2
CCL5
NKG7
CD8A
GZMA
PRF1
IL2RB
GZMB
KLRK1
ZNF683
GNLY
SPOCK2
CTLA4
LTB
CD6
CD5
MAF
ICOS
TRBC1
SATB1
ARID5B
UHRF1
CDT1
RRM2
HIST1H1B
ASF1B
CENPM
HIST1H2AH
HELLS
H2AFX
HIST1H1C
TPSAB1
CPA3
TPSB2
GATA2
MS4A2
SIGLEC6
KIT
SLC18A2
PTGS1
VWA5A
WFDC2
KRT7
SLC34A2
NAPSA
SFTPA1
LCN2
SFTPA2
SFTPB
SFTPC
TPPP3
G0S2
CXCL8
SOD2
SAT1
SCARF1
PHACTR1
PLEK
GBP5
GBP1
PLAU
VSTM1
RAB44
ADGRG3
SYNE1
HK3
JAML
RSRP1
DENND3
CD300LF
HERC3
Clusters
C1 – AM
C2 – AM
C3 – Mafb+ MoMac
C4 – Neu
C5 – CD8+ T cell
C6 – CD4+ T cell
C7 – Cycling cell
C8 – Mast cell
C9 – Epithelial cell
C10 – Unidentified
C11 – Unidentified

```r
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/libopenblasp-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] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ComplexHeatmap_2.16.0 dittoSeq_1.12.0       stringr_1.5.0
## [4] ggplot2_3.4.2         patchwork_1.1.2       SeuratObject_4.1.3
## [7] Seurat_4.3.0          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                 bitops_1.0-7
##   [5] tibble_3.2.1                R.oo_1.25.0
##   [7] polyclip_1.10-4             lifecycle_1.0.3
##   [9] doParallel_1.0.17           globals_0.16.2
##  [11] lattice_0.22-5              MASS_7.3-60.0.1
##  [13] magrittr_2.0.3              limma_3.56.2
##  [15] plotly_4.10.2               rmarkdown_2.23
##  [17] yaml_2.3.7                  httpuv_1.6.11
##  [19] sctransform_0.3.5           spam_2.9-1
##  [21] sp_2.0-0                    spatstat.sparse_3.0-2
##  [23] reticulate_1.30             cowplot_1.1.1
##  [25] pbapply_1.7-2               RColorBrewer_1.1-3
##  [27] abind_1.4-5                 zlibbioc_1.46.0
##  [29] Rtsne_0.16                  GenomicRanges_1.52.0
##  [31] purrr_1.0.1                 R.utils_2.12.2
##  [33] BiocGenerics_0.46.0         RCurl_1.98-1.12
##  [35] circlize_0.4.15             GenomeInfoDbData_1.2.10
##  [37] IRanges_2.34.0              S4Vectors_0.38.1
##  [39] ggrepel_0.9.3               irlba_2.3.5.1
##  [41] listenv_0.9.0               spatstat.utils_3.0-3
##  [43] pheatmap_1.0.12             goftest_1.2-3
##  [45] spatstat.random_3.1-5       fitdistrplus_1.1-11
```

```
##  [47] parallelly_1.36.0           leiden_0.4.3
##  [49] codetools_0.2-19            DelayedArray_0.26.3
##  [51] tidyselect_1.2.0            shape_1.4.6
##  [53] farver_2.1.1                matrixStats_1.0.0
##  [55] stats4_4.3.3                spatstat.explore_3.2-1
##  [57] jsonlite_1.8.7              GetoptLong_1.0.5
##  [59] ellipsis_0.3.2              progressr_0.13.0
##  [61] ggridges_0.5.4              survival_3.5-8
##  [63] iterators_1.0.14            foreach_1.5.2
##  [65] tools_4.3.3                 ica_1.0-3
##  [67] Rcpp_1.0.11                 glue_1.6.2
##  [69] gridExtra_2.3               xfun_0.39
##  [71] MatrixGenerics_1.12.2       GenomeInfoDb_1.36.0
##  [73] withr_2.5.0                 formatR_1.14
##  [75] fastmap_1.1.1               fansi_1.0.4
##  [77] digest_0.6.33               R6_2.5.1
##  [79] mime_0.12                   colorspace_2.1-0
##  [81] scattermore_1.2             Cairo_1.6-2
##  [83] tensor_1.5                  spatstat.data_3.0-1
##  [85] R.methodsS3_1.8.2           utf8_1.2.3
##  [87] tidyr_1.3.0                 generics_0.1.3
##  [89] data.table_1.14.8           httr_1.4.6
##  [91] htmlwidgets_1.6.2           S4Arrays_1.2.1
##  [93] uwot_0.1.16                 pkgconfig_2.0.3
##  [95] gtable_0.3.3                lmtest_0.9-40
##  [97] SingleCellExperiment_1.22.0 XVector_0.40.0
##  [99] htmltools_0.5.5             dotCall64_1.0-2
## [101] clue_0.3-64                 scales_1.2.1
## [103] Biobase_2.60.0              png_0.1-8
## [105] knitr_1.43                  rstudioapi_0.14
## [107] reshape2_1.4.4              rjson_0.2.21
## [109] nlme_3.1-164                zoo_1.8-12
## [111] GlobalOptions_0.1.2         KernSmooth_2.23-22
## [113] parallel_4.3.3              miniUI_0.1.1.1
## [115] pillar_1.9.0                vctrs_0.6.3
## [117] RANN_2.6.1                  promises_1.2.0.1
## [119] xtable_1.8-4                cluster_2.1.6
## [121] evaluate_0.21               magick_2.7.5
## [123] cli_3.6.1                   compiler_4.3.3
## [125] rlang_1.1.1                 crayon_1.5.2
## [127] future.apply_1.11.0         labeling_0.4.2
## [129] plyr_1.8.8                  stringi_1.7.12
## [131] viridisLite_0.4.2           deldir_1.0-9
## [133] munsell_0.5.0               lazyeval_0.2.2
## [135] spatstat.geom_3.2-4         Matrix_1.6-1
## [137] future_1.33.0               shiny_1.7.4.1
## [139] SummarizedExperiment_1.30.2 highr_0.10
## [141] ROCR_1.0-11                 igraph_1.5.0.1
```