

Integrate Hive Samples Low filtering

Joan Abinet

2024-12-13 14:19:33 +0100

Contents

Load Packages	1
Loading data	1
Gene number	4
Annotation	4
Annotating cluster	5
Dotplot Annotated	6
Heatmap top genes per clusters	8

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 = "../Hive-1-Pre_Processing/Data",
                 full.names = T)

list_sample <- list()
for (dir in all_dirs) {

  files <- list.files(dir)
  Count_file <- files[grep("TCM.tsv.gz$", files)]
  Countdata <- read.table(paste0(dir, "/", Count_file),
                         sep = "\t", header = T, row.names = 1)

  Lavage_cellsHive <- CreateSeuratObject(counts = Countdata,
                                           project = str_sub(dir, -5, -1), min.features = 100)
```

```

Lavage_cellsHive[["percent.mt"]] <- PercentageFeatureSet(Lavage_cellsHive,
  pattern = "^\u00c9MT-") # MT : human cells
Lavage_cellsHive <- subset(Lavage_cellsHive, subset = nFeature_RNA <
  8000 & percent.mt < 20) # or 5
list_sample <- append(list_sample, Lavage_cellsHive)
}

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)

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",
  verbose = F)

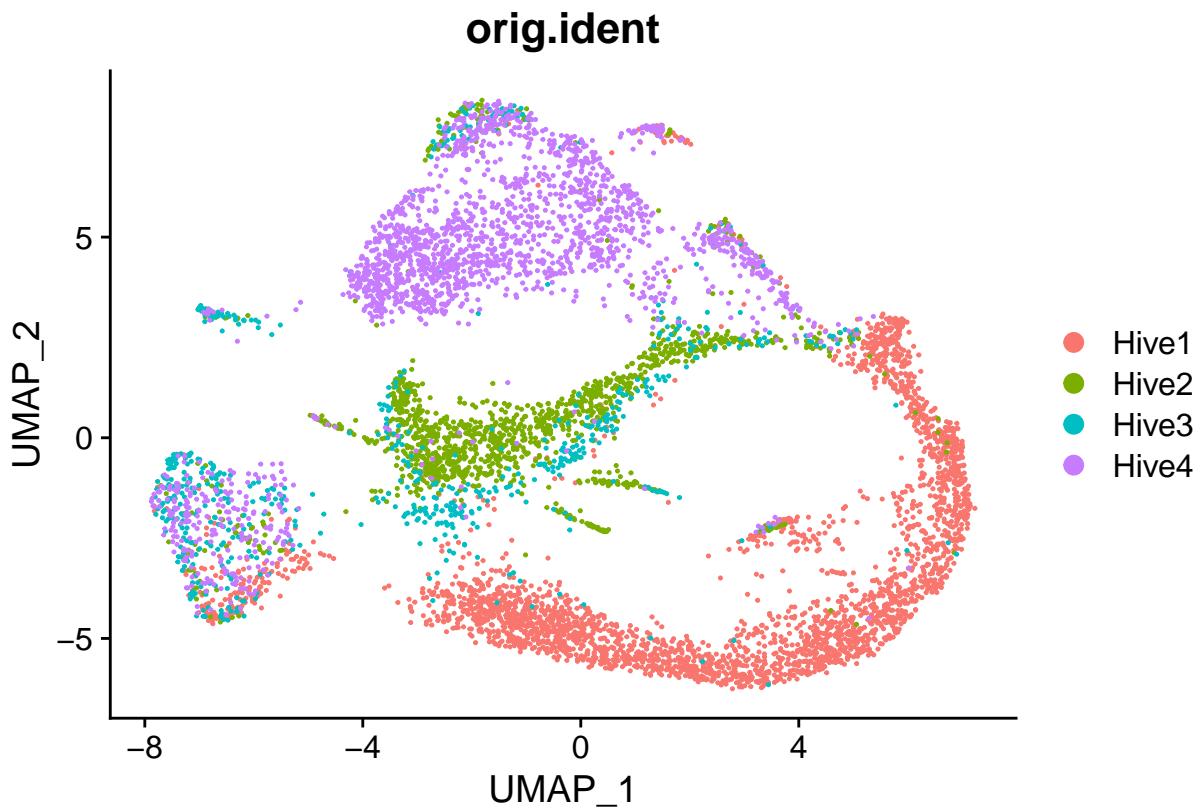
BAL_Hive.integrated <- IntegrateData(anchorset = BAL.anchors,
  verbose = F)

DefaultAssay(BAL_Hive.integrated) <- "integrated"

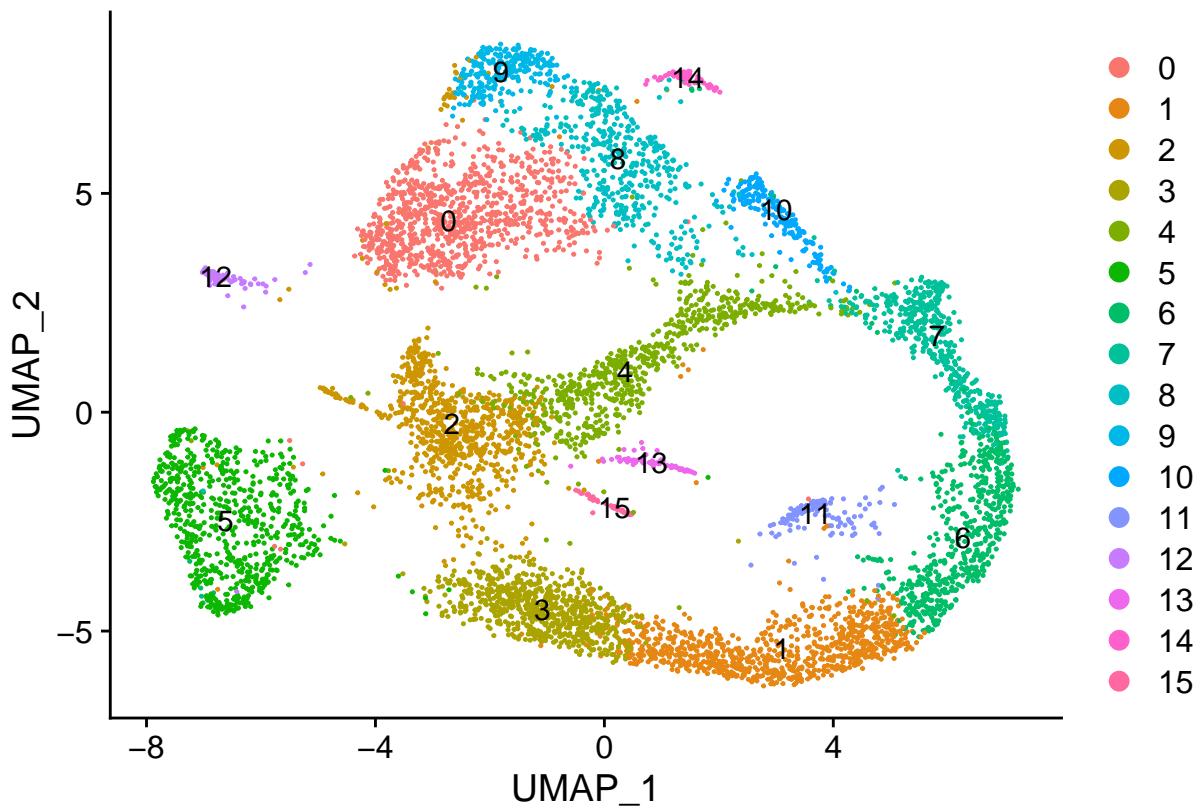
# Run the standard workflow for visualization and
# clustering
BAL_Hive.integrated <- ScaleData(BAL_Hive.integrated,
  verbose = FALSE)
BAL_Hive.integrated <- RunPCA(BAL_Hive.integrated,
  npcs = 30, verbose = FALSE)
BAL_Hive.integrated <- RunUMAP(BAL_Hive.integrated,
  reduction = "pca", dims = 1:15)
BAL_Hive.integrated <- FindNeighbors(BAL_Hive.integrated,
  reduction = "pca", dims = 1:15)
BAL_Hive.integrated <- FindClusters(BAL_Hive.integrated,
  resolution = 0.5)

DimPlot(BAL_Hive.integrated, reduction = "umap", group.by = "orig.ident")

```

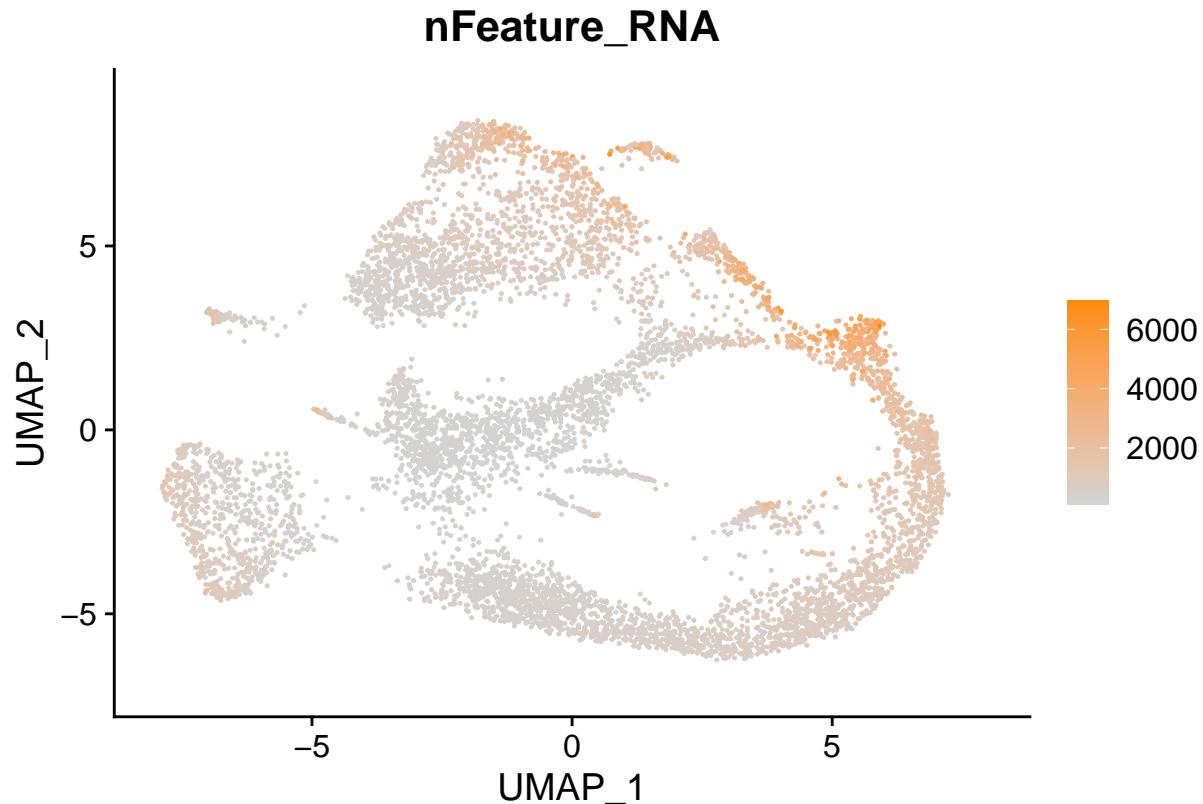


```
DimPlot(BAL_Hive.integrated, reduction = "umap", label = T)
```



Gene number

```
FeaturePlot(BAL_Hive.integrated, features = "nFeature_RNA",
            cols = c("lightgray", "darkorange"))
```



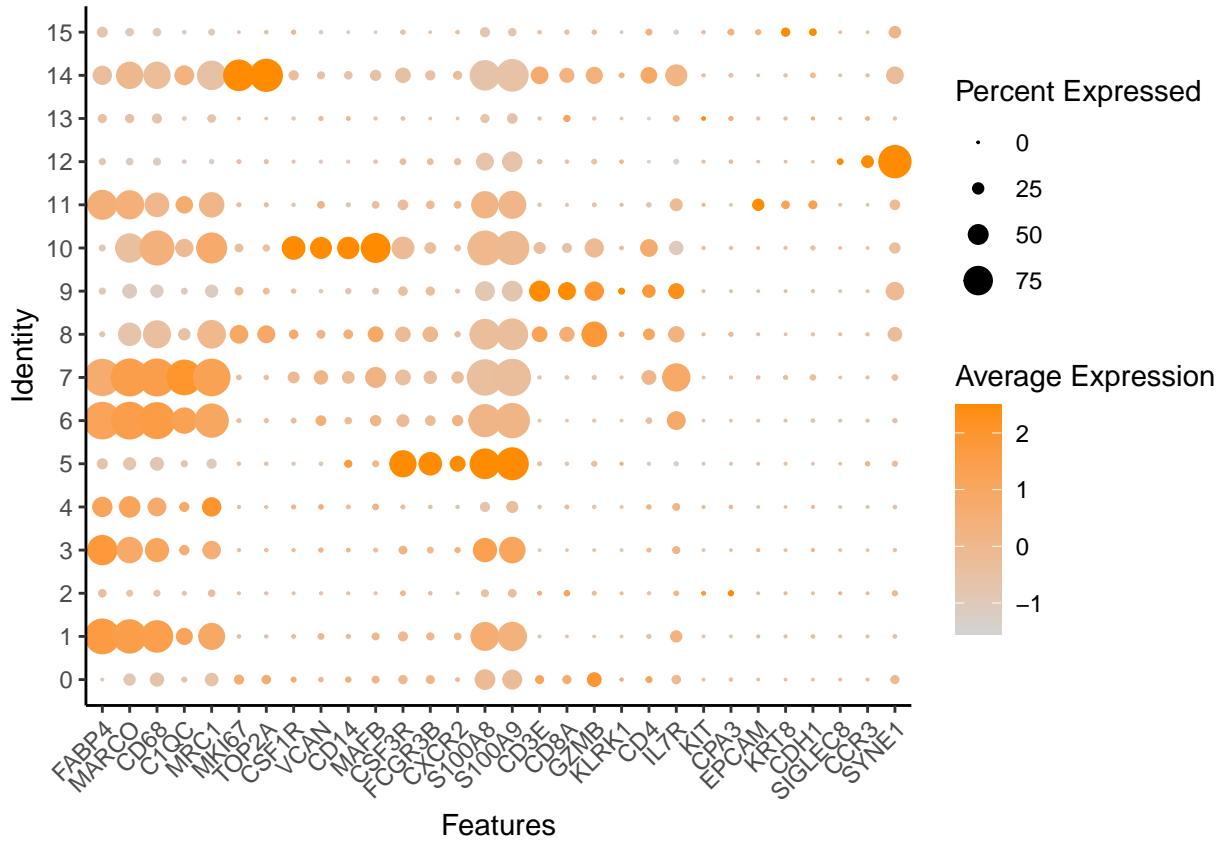
```
# ggsave('FeaturePlot_geneCount_low.pdf', width =
# 8, height = 4)
```

Annotation

```
DefaultAssay(BAL_Hive.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_Hive.integrated, features = Marker_gene,
        cols = c("lightgray", "darkorange")) + theme_classic() +
        theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



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

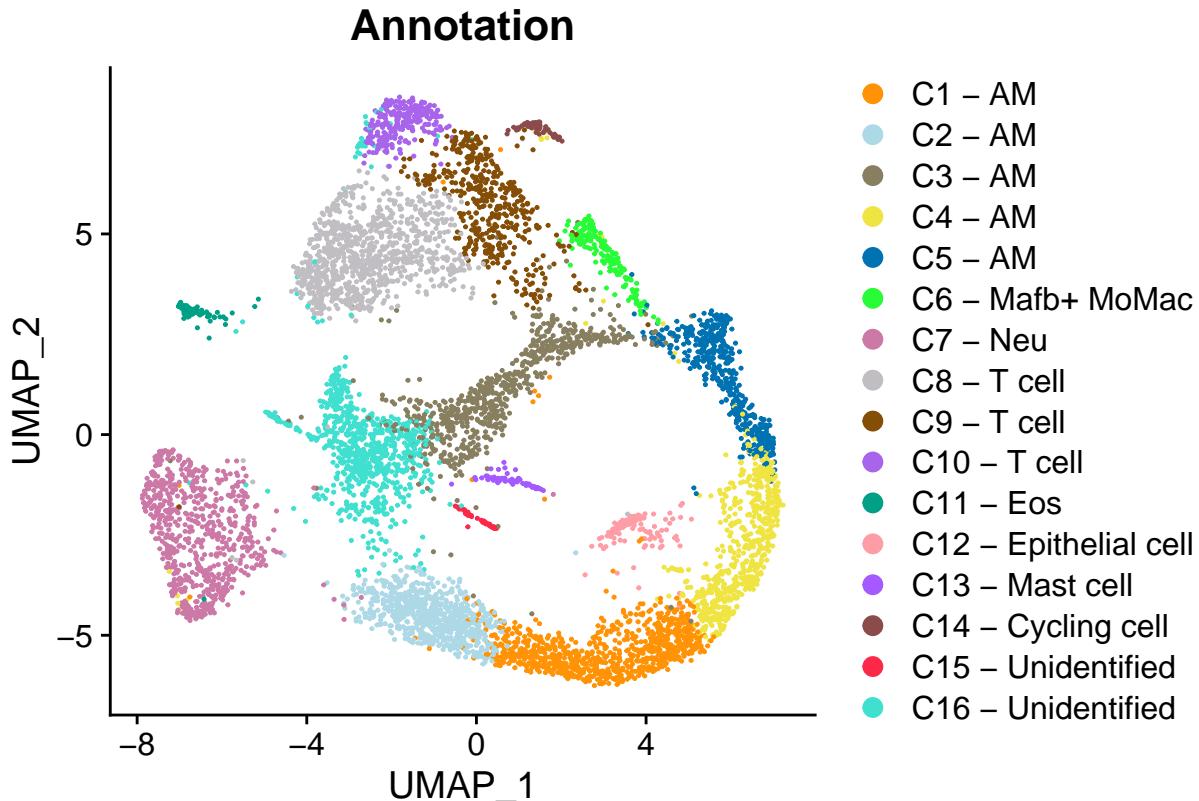
Annotating cluster

```
BAL_Hive.integrated$Annotation <- BAL_Hive.integrated$seurat_clusters
BAL_Hive.integrated$Annotation <- as.factor(BAL_Hive.integrated$Annotation)
levels(BAL_Hive.integrated$Annotation) <- c("C8 - T cell",
  "C1 - AM", "C16 - Unidentified", "C2 - AM", "C3 - AM",
  "C7 - Neu", "C4 - AM", "C5 - AM", "C9 - T cell",
  "C10 - T cell", "C6 - Mafb+ MoMac", "C12 - Epithelial cell",
  "C11 - Eos", "C13 - Mast cell", "C14 - Cycling cell",
  "C15 - Unidentified")

BAL_Hive.integrated$Annotation <- factor(BAL_Hive.integrated$Annotation,
  levels = c("C1 - AM", "C2 - AM", "C3 - AM", "C4 - AM",
  "C5 - AM", "C6 - Mafb+ MoMac", "C7 - Neu",
  "C8 - T cell", "C9 - T cell", "C10 - T cell",
  "C11 - Eos", "C12 - Epithelial cell", "C13 - Mast cell",
  "C14 - Cycling cell", "C15 - Unidentified",
  "C16 - Unidentified"))

my_palette <- c("#ff9305", "lightblue", "#877f5f",
  "#FOE442", "#0072B2", "#28FC37", "#CC79A7", "#c0bec2",
  "#854e07", "#a865eb", "#00A087", "#FF9DA7", "#A55AFF",
  "#8C4B4B", "#FC2848", "turquoise")
```

```
DimPlot(BAL_Hive.integrated, group.by = "Annotation",
        cols = my_palette)
```

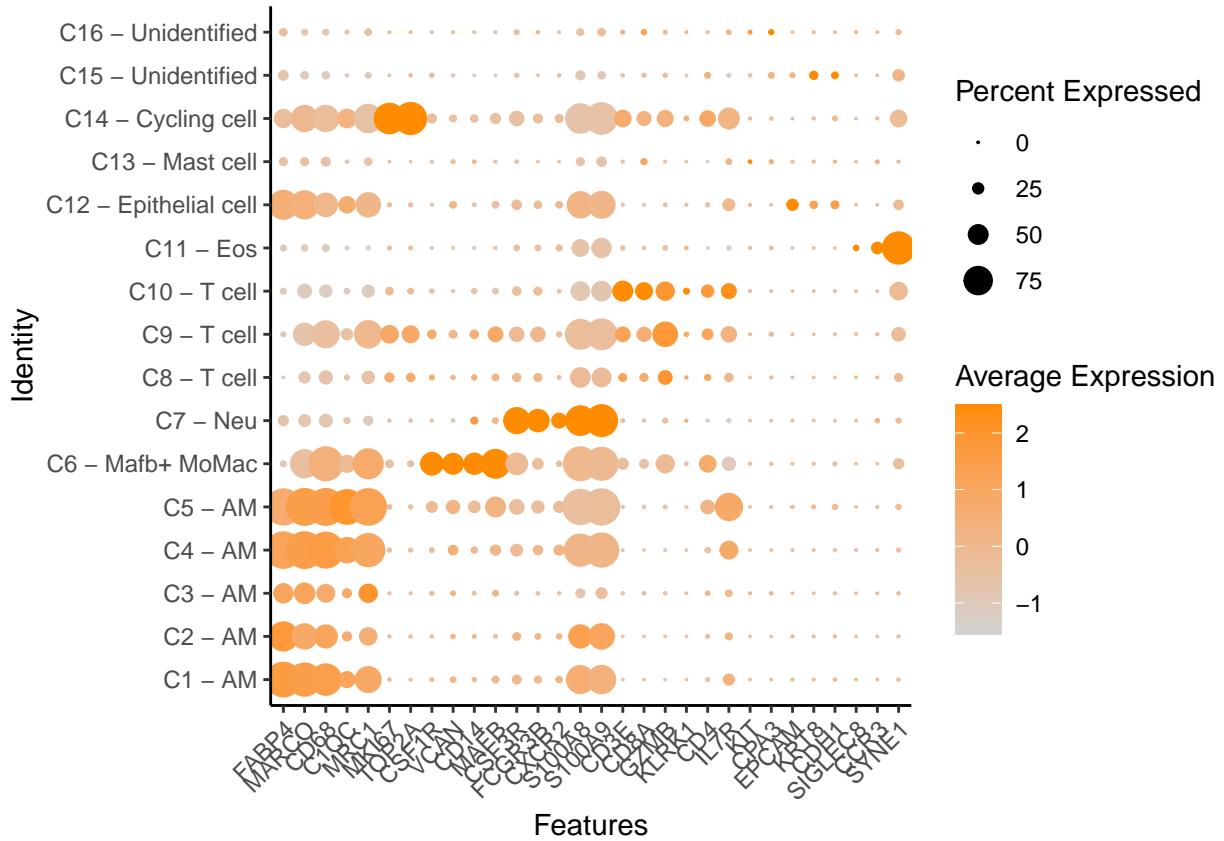


```
# ggsave('Umap_annotation_low.pdf', width = 8,
# height = 4)
```

Dotplot Annotated

```
DefaultAssay(BAL_Hive.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_Hive.integrated, features = Marker_gene,
        group.by = "Annotation", cols = c("lightgray",
                                         "darkorange")) + theme_classic() + theme(axis.text.x = element_text(angle = 45,
                                         hjust = 1))
```



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

```
Idents(BAL_Hive.integrated) <- "Annotation"
lavage.markers <- FindAllMarkers(BAL_Hive.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_Hive.integrated,
    slot = "data")[as.character(top10$gene), ])

df <- as.data.frame(BAL_Hive.integrated$Annotation)
colnames(df) <- "Clusters"
color_df <- list(Clusters = c(`C1 - AM` = "#ff9305",
    `C2 - AM` = "lightblue", `C3 - AM` = "#877f5f",
    `C4 - AM` = "#FOE442", `C5 - AM` = "#0072B2", `C6 - Mafb+ MoMac` = "#28FC37",
    `C7 - Neu` = "#CC79A7", `C8 - T cell` = "#c0bec2",
    `C9 - T cell` = "#854e07", `C10 - T cell` = "#a865eb",
    `C11 - Eos` = "#00A087", `C12 - Epithelial cell` = "#FF9DA7",
    `C13 - Mast cell` = "#A55AFF", `C14 - Cycling cell` = "#8C4B4B",
    `C15 - Unidentified` = "#FC2848", `C16 - Unidentified` = "turquoise")))

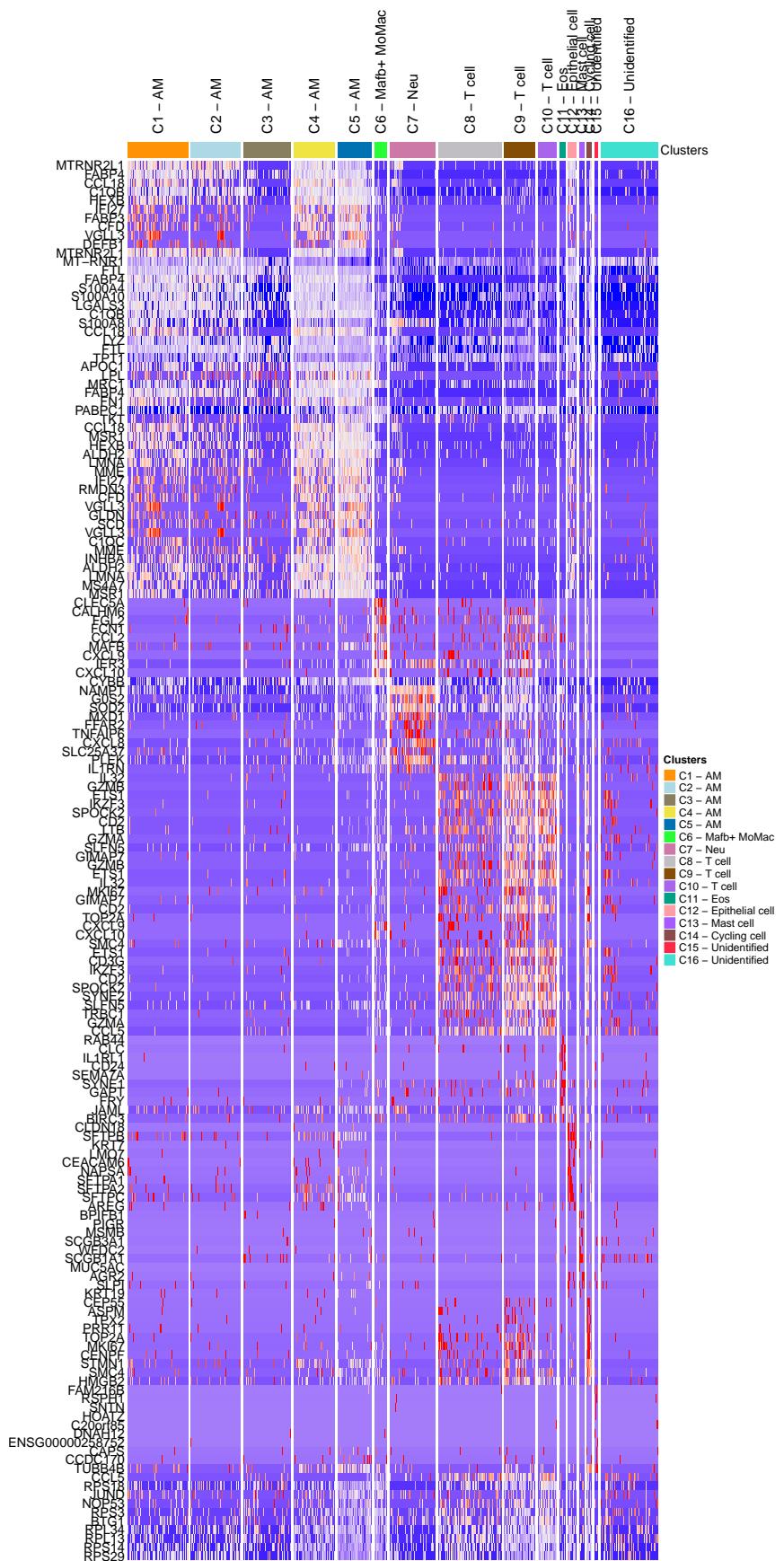
Heatmap <- Heatmap(t(scale(t(mat))), show_column_names = F,
    column_split = BAL_Hive.integrated$Annotation,
```

```
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_Hive_low.pdf', width = 30, height = 45,
# units ='cm')
```

Heatmap top genes per clusters

```
Heatmap
```



```

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-p0.3.20.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] RColorBrewer_1.1-3       shape_1.4.6
## [3] rstudioapi_0.14          jsonlite_1.8.7
## [5] magrittr_2.0.3            magick_2.7.5
## [7] spatstat.utils_3.0-3     farver_2.1.1
## [9] rmarkdown_2.23            GlobalOptions_0.1.2
## [11] zlibbioc_1.46.0          vctrs_0.6.3
## [13] ROCR_1.0-11             Cairo_1.6-2
## [15] spatstat.explore_3.2-1   RCurl_1.98-1.12
## [17] S4Arrays_1.2.1           htmltools_0.5.5
## [19] sctransform_0.3.5        parallelly_1.36.0
## [21] KernSmooth_2.23-22      htmlwidgets_1.6.2
## [23] ica_1.0-3                plyr_1.8.8
## [25] plotly_4.10.2            zoo_1.8-12
## [27] igraph_1.5.0.1           iterators_1.0.14
## [29] mime_0.12                 lifecycle_1.0.3
## [31] pkgconfig_2.0.3           Matrix_1.6-1
## [33] R6_2.5.1                 fastmap_1.1.1
## [35] clue_0.3-64              GenomeInfoDbData_1.2.10
## [37] MatrixGenerics_1.12.2    fitdistrplus_1.1-11
## [39] future_1.33.0             shiny_1.7.4.1
## [41] digest_0.6.33            colorspace_2.1-0
## [43] S4Vectors_0.38.1          tensor_1.5
## [45] irlba_2.3.5.1            GenomicRanges_1.52.0

```

```

## [47] labeling_0.4.2
## [49] fansi_1.0.4
## [51] httr_1.4.6
## [53] abind_1.4-5
## [55] doParallel_1.0.17
## [57] highr_0.10
## [59] DelayedArray_0.26.3
## [61] tools_4.3.3
## [63] httpuv_1.6.11
## [65] goftest_1.2-3
## [67] nlme_3.1-164
## [69] Rtsne_0.16
## [71] reshape2_1.4.4
## [73] gtable_0.3.3
## [75] tidyR_1.3.0
## [77] XVector_0.40.0
## [79] utf8_1.2.3
## [81] spatstat.geom_3.2-4
## [83] foreach_1.5.2
## [85] RANN_2.6.1
## [87] limma_3.56.2
## [89] later_1.3.1
## [91] splines_4.3.3
## [93] survival_3.5-8
## [95] tidyselect_1.2.0
## [97] miniUI_0.1.1.1
## [99] knitr_1.43
## [101] IRanges_2.34.0
## [103] scattermore_1.2
## [105] xfun_0.39
## [107] matrixStats_1.0.0
## [109] stringi_1.7.12
## [111] yaml_2.3.7
## [113] codetools_0.2-19
## [115] cli_3.6.1
## [117] xtable_1.8-4
## [119] munsell_0.5.0
## [121] GenomeInfoDb_1.36.0
## [123] spatstat.random_3.1-5
## [125] parallel_4.3.3
## [127] dotCall64_1.0-2
## [129] listenv_0.9.0
## [131] scales_1.2.1
## [133] crayon_1.5.2
## [135] purrr_1.0.1
## [137] rlang_1.1.1
## [139] formatR_1.14
progressr_0.13.0
spatstat.sparse_3.0-2
polyclip_1.10-4
compiler_4.3.3
withr_2.5.0
MASS_7.3-60.0.1
rjson_0.2.21
lmtest_0.9-40
future.apply_1.11.0
glue_1.6.2
promises_1.2.0.1
cluster_2.1.6
generics_0.1.3
spatstat.data_3.0-1
data.table_1.14.8
sp_2.0-0
BiocGenerics_0.46.0
RcppAnnoy_0.0.21
ggrepel_0.9.3
pillar_1.9.0
spam_2.9-1
circlize_0.4.15
lattice_0.22-5
deldir_1.0-9
SingleCellExperiment_1.22.0
pbapply_1.7-2
gridExtra_2.3
SummarizedExperiment_1.30.2
stats4_4.3.3
Biobase_2.60.0
pheatmap_1.0.12
lazyeval_0.2.2
evaluate_0.21
tibble_3.2.1
uwot_0.1.16
reticulate_1.30
Rcpp_1.0.11
globals_0.16.2
png_0.1-8
ellipsis_0.3.2
bitops_1.0-7
viridisLite_0.4.2
ggridges_0.5.4
leiden_0.4.3
GetoptLong_1.0.5
cowplot_1.1.1

```