Clustering and Visualisation

Abinet Joan

2023-06-01 13:32:05 +0200

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

Importing the cells from the Mock group and the PR8 group.

```
myeloid_cells <- readRDS("Myeloid_cells_Part2.rds")
```

Visualizing cells

Defining parameters for the visualisation

```
colors <- c("#B2DF8A","#ABD61C","#1F78B4","#A6CEE3", "#E31A1C","#E3751C","#600078",
"#33A02C", "#FDBF6F", "#526317",  "#D4AAC6", "#784620")
```

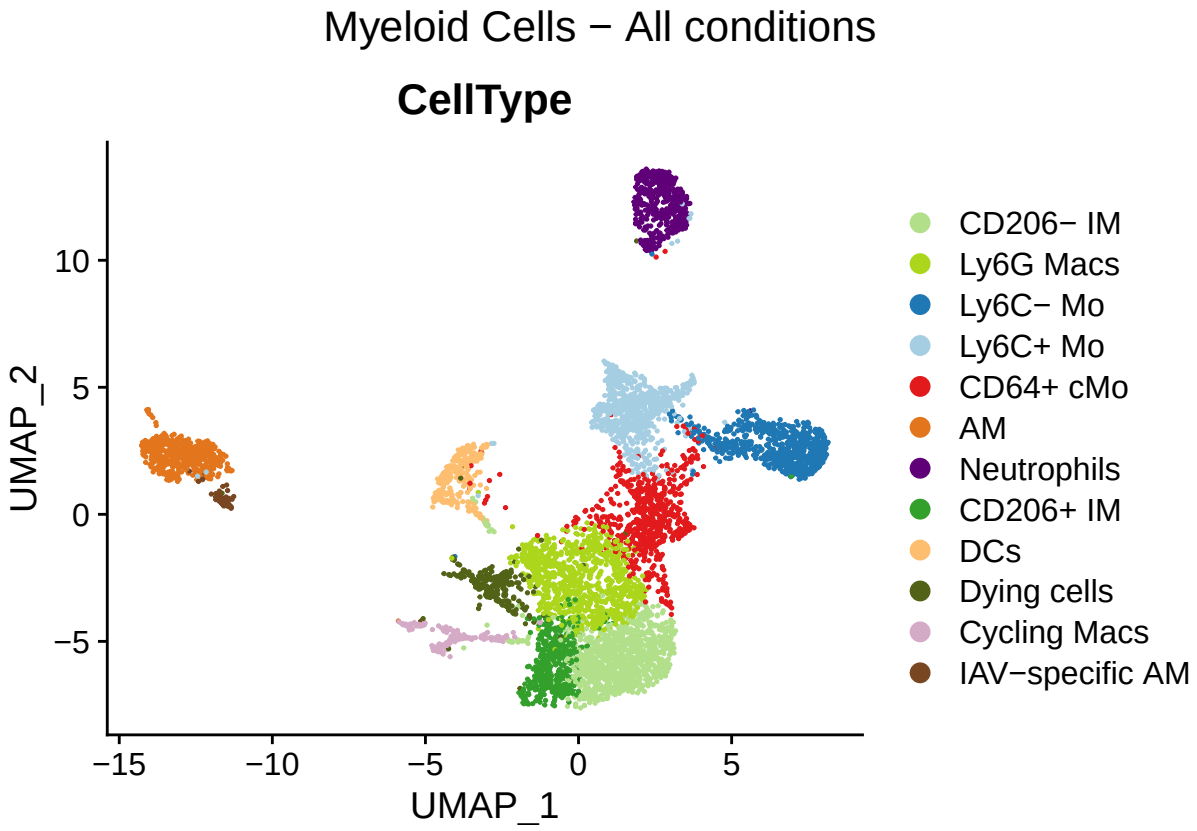
Adding the cell type to metadata

```
labels <- c("CD206- IM","Ly6G Macs","Ly6C- Mo","Ly6C+ Mo","CD64+
cMo","AM","Neutrophils","CD206+ IM","DCs","Dying cells","Cycling Macs","IAV-specific AM")

# create a new column in metadata
myeloid_cells$CellType <- myeloid_cells$seurat_clusters

# replace cluster number by cell type
levels(myeloid_cells$CellType) <-labels
```

```
DimPlot(myeloid_cells, reduction = "umap", cols = colors, group.by = "CellType") +
plot_annotation(
  title = 'Myeloid Cells - All conditions',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```




```
#ggsave("graph_Umap.pdf",  height = 12 , width = 19, units = "cm")
```

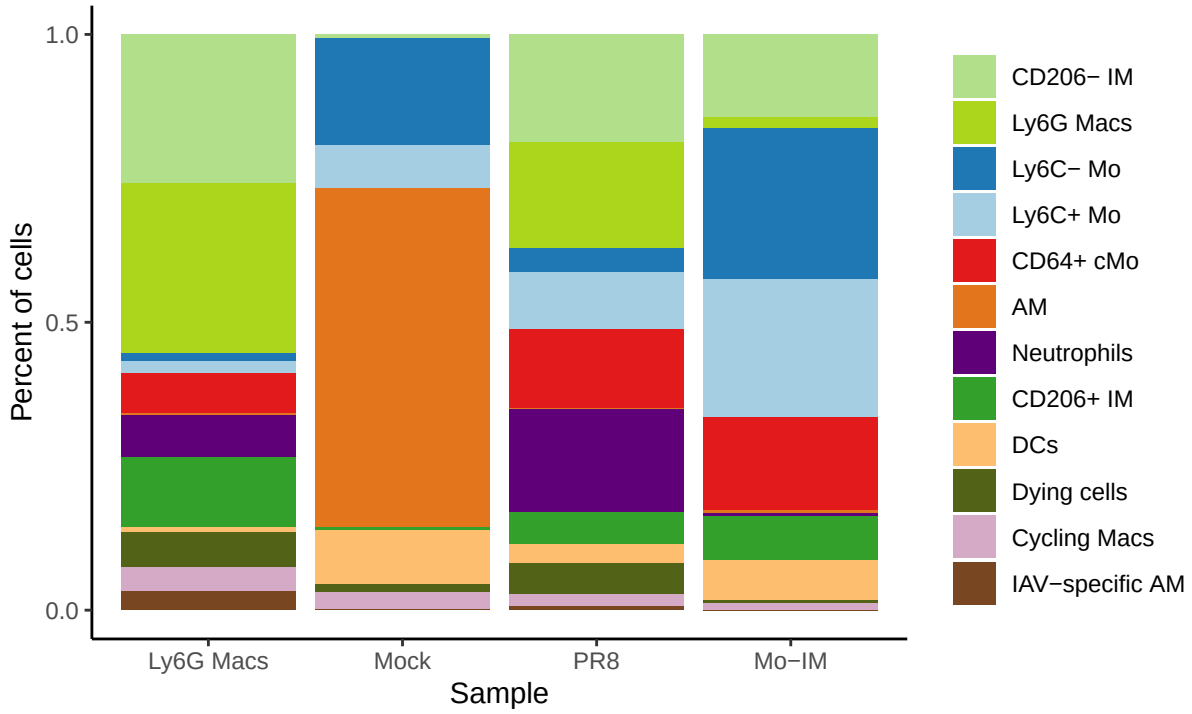
Visualsation of the frequency graph.

```
colors <- c("#B2DF8A","#ABD61C","#1F78B4","#A6CEE3", "#E31A1C","#E3751C","#600078",
"#33A02C", "#FDBF6F", "#526317",  "#D4AAC6", "#784620")

labels <- c("CD206- IM","Ly6G Macs","Ly6C- Mo","Ly6C+ Mo","CD64+
cMo","AM","Neutrophils","CD206+ IM","DCs","Dying cells","Cycling Macs","IAV-specific AM")

#var.labels.rename = labels
var_order <- c(1,2,5,6,7,8,9,10,11,12,3,4)

dittoBarPlot(myeloid_cells, "seurat_clusters", group.by = "orig.ident", color.panel =
colors, main ="", var.labels.reorder = var_order, var.labels.rename = labels, x.labels =
c("Ly6G Macs", "Mock", "PR8", "Mo-IM"), x.labels.rotate = F, xlab = "Sample")+
plot_annotation(
  title = 'Myeloid Cells - Cluster frequency',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```



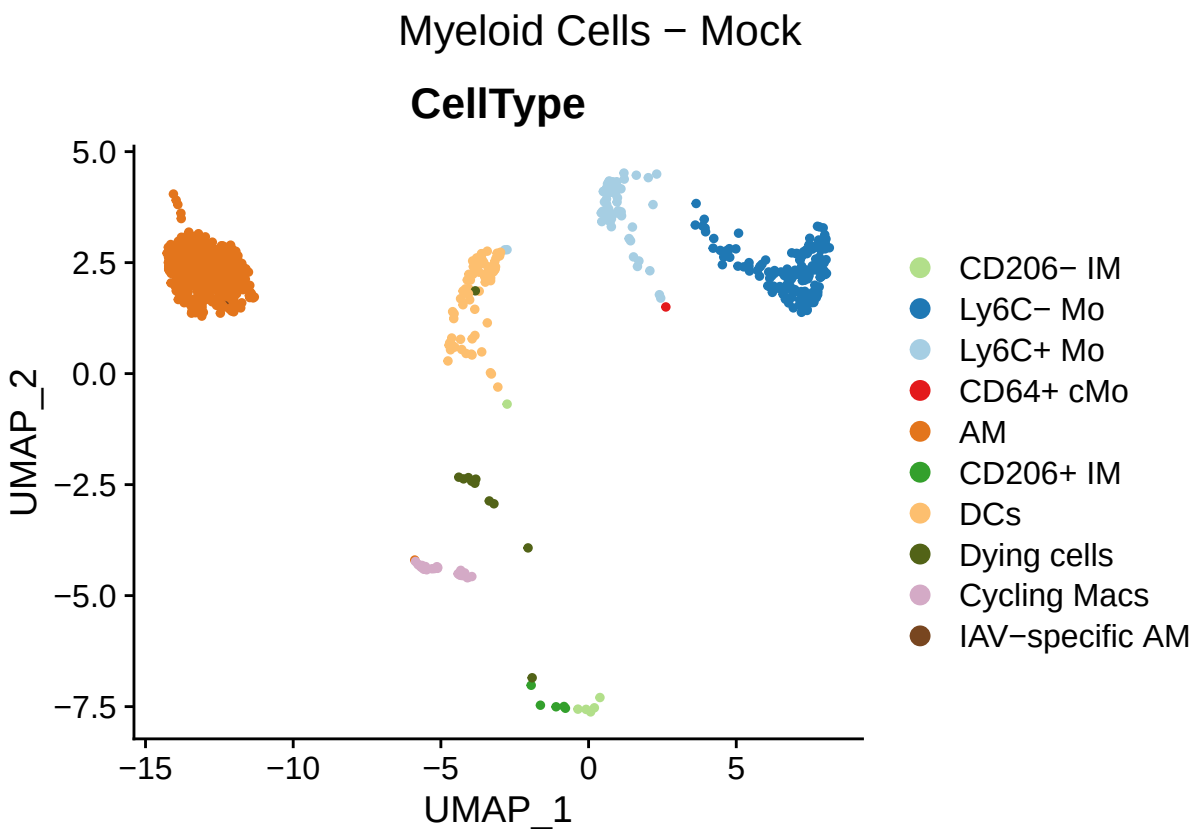

```
#ggsave("graphe_clusterFreq.pdf", height = 12 , width = 19, units = "cm")
```

Visualisation of the Mock cells

```
cell_myeloid_ctrl <- subset(myeloid_cells, orig.ident %in% "cell_myeloid_ctrl")

colors_mock <- c("#B2DF8A","#1F78B4","#A6CEE3", "#E31A1C","#E3751C", "#33A02C",
"#FDBF6F", "#526317",  "#D4AAC6", "#784620")

DimPlot(cell_myeloid_ctrl, reduction = "umap", cols = colors_mock, group.by = "CellType")
+ plot_annotation(
  title = 'Myeloid Cells - Mock',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

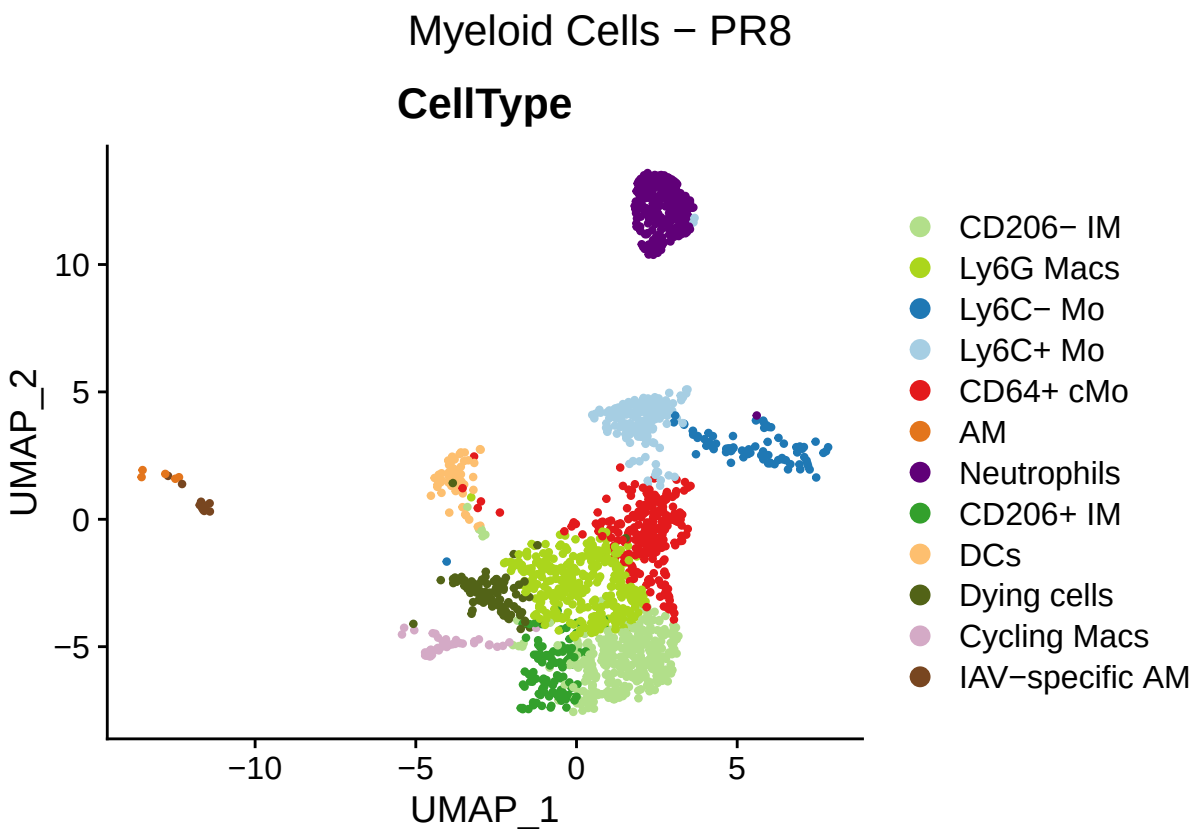



```
#ggsave("UMAP_mock.pdf", height = 12 , width = 19, units = "cm")
```

Visualisation of the PR8 cells

```
cell_myeloid_Pr8 <- subset(myeloid_cells, orig.ident %in% "cell_myeloidPr8")

DimPlot(cell_myeloid_Pr8, reduction = "umap", cols = colors, group.by = "CellType") +
plot_annotation(
  title = 'Myeloid Cells - PR8',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

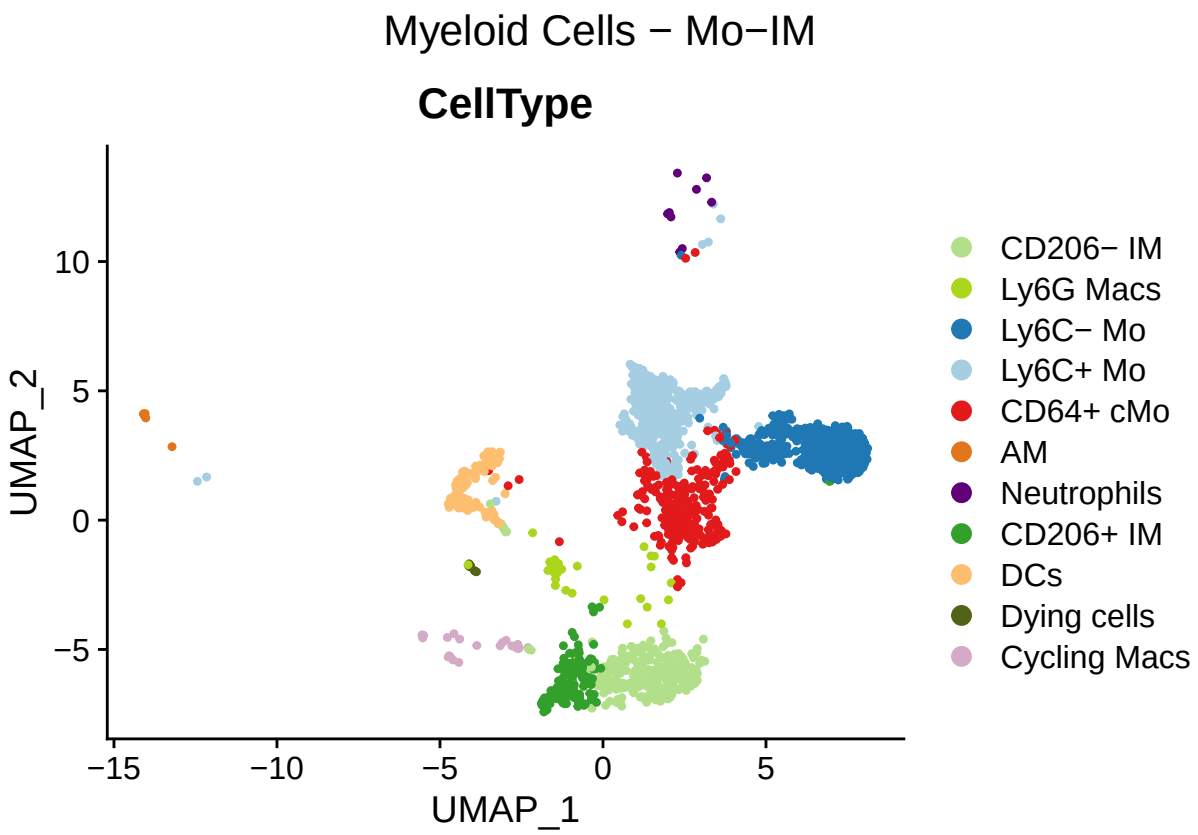



```
#ggsave("UMAP_PR8.pdf", height = 12 , width = 19, units = "cm")
```

Visualisation IMcells

```
cell_myeloid_IMcells <- subset(myeloid_cells, orig.ident %in% "IM_cells")

DimPlot(cell_myeloid_IMcells, reduction = "umap", cols = colors, group.by = "CellType") +
plot_annotation(
  title = 'Myeloid Cells - Mo-IM',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

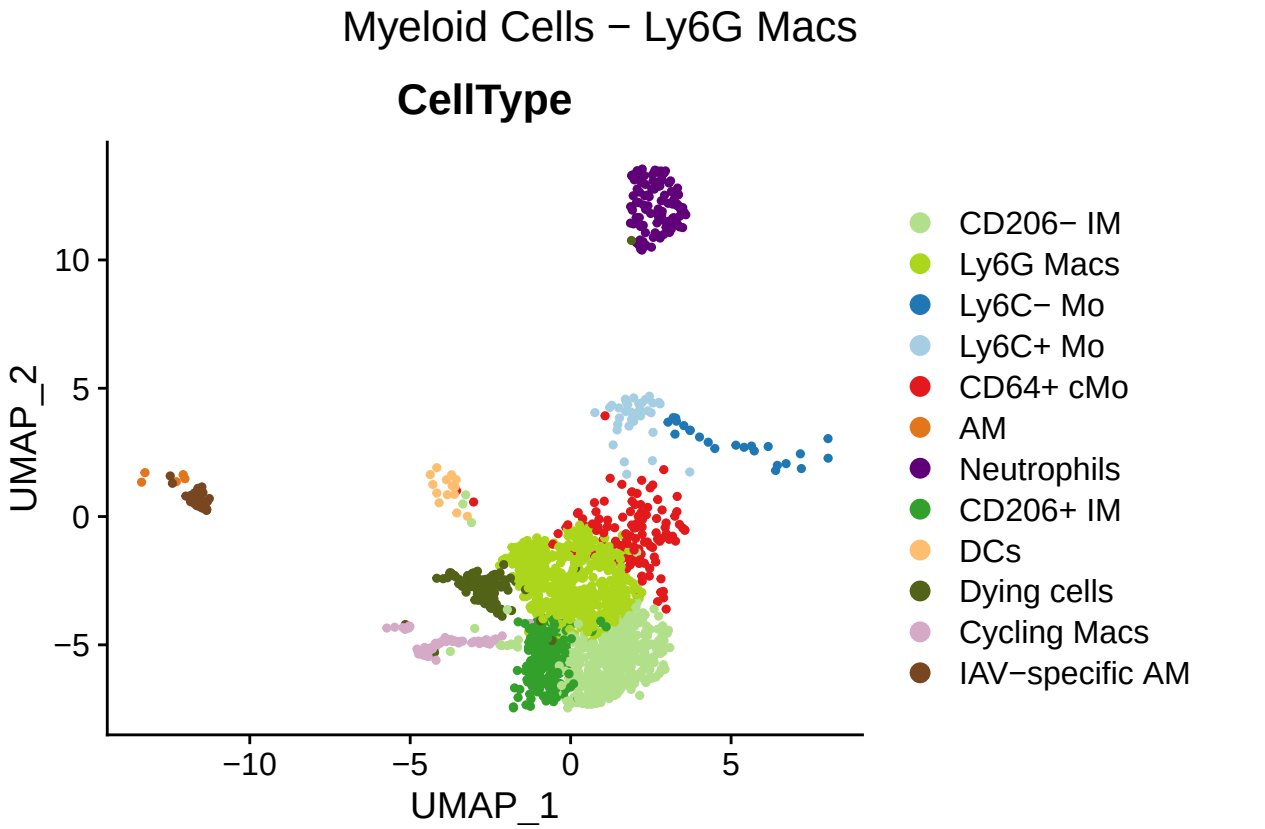



```
#ggsave("UMAP_PR8.pdf", height = 12 , width = 19, units = "cm")
```

Visualisation of Motro Cells

```
cell_myeloid_Ly6GMacs <- subset(myeloid_cells, orig.ident %in% "cell_Motro_Pr8")

DimPlot(cell_myeloid_Ly6GMacs, reduction = "umap", cols = colors, group.by = "CellType")
+ plot_annotation(
  title = 'Myeloid Cells - Ly6G Macs',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```




```
#ggsave("UMAP_Ly6GMacs.pdf", height = 12 , width = 19, units = "cm")
```

Save the result for further analysis

```
saveRDS(myeloid_cells, "Myeloid_cells_Final.rds")
```

```
DefaultAssay(myeloid_cells) <- "RNA"

DotPlot(myeloid_cells, features = c("C1qa", "Cd74", "Tmem119", "Ctsb", "Ctsz", "Lgals1",
"Lgals3", "Arg1", "Spp1", "Ace", "Nr4a1", "Fcgr4", "Ccr2","Ly6c2", "Irf7", "Chil3",
"Ear1", "Fabp1", "S100a8", "S100a9", "Mmp9", "C1qc", "Mrc1", "Maf", "H2-Ab1", "Cd209a",
"Flt3", "percent.mt", "Birc5", "Top2a", "Mki67", "Ear2"), group.by = "CellType") +
plot_annotation(
  title = 'Dot plot genes markers',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

Differential expression Analysis

```
# find markers for every cluster compared to all remaining cells, report only the
positive ones
myeloids.markers <- FindAllMarkers(myeloid_cells, only.pos = TRUE, min.pct = 0.25,
logfc.threshold = 0.25, test.use = "LR", latent.vars = "orig.ident") # The execution is
quite long
```

The results from DE can directly be load from a csv file

```
myeloids.markers <- read.csv("myeloids_markers.csv")
```

```
# select the top 10 markers the most diferentially expressed of each clusters
myeloids.markers %>%
    group_by(cluster) %>%
    top_n(n = 20, wt = avg_log2FC) -> top10

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

df <- as.data.frame(myeloid_cells$CellType)
colnames(df) <- "Clusters"

color_df <- list(Clusters =
                   c("CD206- IM" = "#B2DF8A",
                     "Ly6G Macs" = "#ABD61C",
                     "Ly6C- Mo" = "#1F78B4",
                     "Ly6C+ Mo" = "#A6CEE3",
                     "CD64+ cMo" = "#E31A1C",
                     "AM" = "#E3751C",
                     "Neutrophils" = "#600078",
                     "CD206+ IM" = "#33A02C",
                     "DCs" = "#FDBF6F",
                     "Dying cells" = "#526317",
                     "Cycling Macs" = "#D4AAC6",
                     "IAV-specific AM" = "#784620"))

heatmap_allGenes <- Heatmap(t(scale(t(mat))), name="expressop,", show_column_names =
FALSE,
        column_split = factor(myeloid_cells$CellType),
        cluster_column_slices = F,
        cluster_rows = F,
        show_column_dend = F,
        top_annotation = HeatmapAnnotation(df = df, col = color_df),
        column_title_rot = 90,
        row_names_gp = gpar(fontsize = 8),
        row_names_side = "left",
        use_raster=F,
        show_heatmap_legend = F)

#tidyHeatmap::save_pdf(heatmap_allGenes, "Heatmap.pdf", width = 30, height = 55, units =
"cm")
```

```
sessionInfo()
```

```
## R version 4.0.3 (2020-10-10)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.6 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=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
##
## attached base packages:
## [1] grid      stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] ComplexHeatmap_2.6.2 formatR_1.14         dittoSeq_1.2.6
## [4] ggplot2_3.4.0        patchwork_1.1.2      SeuratObject_4.1.3
## [7] Seurat_4.3.0         dplyr_1.0.10
##
## loaded via a namespace (and not attached):
##   [1] circlize_0.4.15            plyr_1.8.8
##   [3] igraph_1.4.1               lazyeval_0.2.2
##   [5] sp_1.6-0                   splines_4.0.3
##   [7] listenv_0.9.0              scattermore_0.8
##   [9] GenomeInfoDb_1.26.7        digest_0.6.31
##  [11] htmltools_0.5.4            magick_2.7.3
##  [13] fansi_1.0.4                magrittr_2.0.3
##  [15] tensor_1.5                 cluster_2.1.0
##  [17] ROCR_1.0-11                limma_3.46.0
##  [19] globals_0.16.2             matrixStats_0.63.0
##  [21] spatstat.sparse_3.0-0      colorspace_2.1-0
##  [23] ggrepel_0.9.2              xfun_0.37
##  [25] crayon_1.5.2               RCurl_1.98-1.10
##  [27] jsonlite_1.8.4             progressr_0.13.0
##  [29] spatstat.data_3.0-0        survival_3.2-7
##  [31] zoo_1.8-11                 glue_1.6.2
##  [33] polyclip_1.10-4            gtable_0.3.1
##  [35] zlibbioc_1.36.0            XVector_0.30.0
##  [37] leiden_0.4.3               GetoptLong_1.0.5
##  [39] DelayedArray_0.16.3        shape_1.4.6
##  [41] future.apply_1.10.0        SingleCellExperiment_1.12.0
##  [43] BiocGenerics_0.36.1        abind_1.4-5
##  [45] scales_1.2.1               pheatmap_1.0.12
##  [47] DBI_1.1.3                  edgeR_3.32.1
##  [49] spatstat.random_3.1-3      miniUI_0.1.1.1
```

```
##  [51] Rcpp_1.0.10                 viridisLite_0.4.1
##  [53] xtable_1.8-4                clue_0.3-64
##  [55] reticulate_1.27             stats4_4.0.3
##  [57] htmlwidgets_1.6.1           httr_1.4.5
##  [59] RColorBrewer_1.1-3          ellipsis_0.3.2
##  [61] ica_1.0-3                   farver_2.1.1
##  [63] pkgconfig_2.0.3             uwot_0.1.14
##  [65] deldir_1.0-6                locfit_1.5-9.4
##  [67] utf8_1.2.3                  labeling_0.4.2
##  [69] tidyselect_1.2.0            rlang_1.0.6
##  [71] reshape2_1.4.4              later_1.3.0
##  [73] munsell_0.5.0               tools_4.0.3
##  [75] cli_3.6.0                   generics_0.1.3
##  [77] ggridges_0.5.4              evaluate_0.20
##  [79] stringr_1.5.0               fastmap_1.1.1
##  [81] yaml_2.3.7                  goftest_1.2-3
##  [83] knitr_1.42                  fitdistrplus_1.1-8
##  [85] purrr_1.0.1                 RANN_2.6.1
##  [87] pbapply_1.7-0               future_1.32.0
##  [89] nlme_3.1-162                mime_0.12
##  [91] compiler_4.0.3              rstudioapi_0.14
##  [93] plotly_4.10.1               png_0.1-8
##  [95] spatstat.utils_3.0-1        tibble_3.1.8
##  [97] stringi_1.7.12              highr_0.10
##  [99] lattice_0.20-41             Matrix_1.5-3
## [101] vctrs_0.5.2                 pillar_1.8.1
## [103] lifecycle_1.0.3             spatstat.geom_3.0-6
## [105] lmtest_0.9-40               GlobalOptions_0.1.2
## [107] RcppAnnoy_0.0.20            data.table_1.14.8
## [109] cowplot_1.1.1               bitops_1.0-7
## [111] irlba_2.3.5.1               httpuv_1.6.9
## [113] GenomicRanges_1.42.0        R6_2.5.1
## [115] promises_1.2.0.1            KernSmooth_2.23-20
## [117] gridExtra_2.3               IRanges_2.24.1
## [119] parallelly_1.34.0           codetools_0.2-19
## [121] MASS_7.3-53                 assertthat_0.2.1
## [123] SummarizedExperiment_1.20.0 rjson_0.2.21
## [125] withr_2.5.0                 sctransform_0.3.5
## [127] S4Vectors_0.28.1            GenomeInfoDbData_1.2.4
## [129] parallel_4.0.3              tidyr_1.2.1
## [131] rmarkdown_2.19              MatrixGenerics_1.2.1
## [133] Cairo_1.6-0                 Rtsne_0.16
## [135] spatstat.explore_3.0-5      Biobase_2.50.0
## [137] shiny_1.7.4
```