Clustering and Visualisation

Abinet Joan

2024-10-18 11:47:17 +0200

Loading packages

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

Loading data

```
myeloid_cells <- readRDS("../2-Integration_workflow/Myeloid_cells_Part2.rds")
```

The Condition Myeloid PR8 1 and Myeloid PR8 2 are merged into one condition IAV

and the IM cells are renamed Steady-State CD64

```
myeloid_cells$orig.ident <- as.factor(myeloid_cells$orig.ident)
levels(myeloid_cells$orig.ident) <-
c("myeloid_PR8","cell_myeloid_ctrl","myeloid_PR8","Steady-State CD64")
```

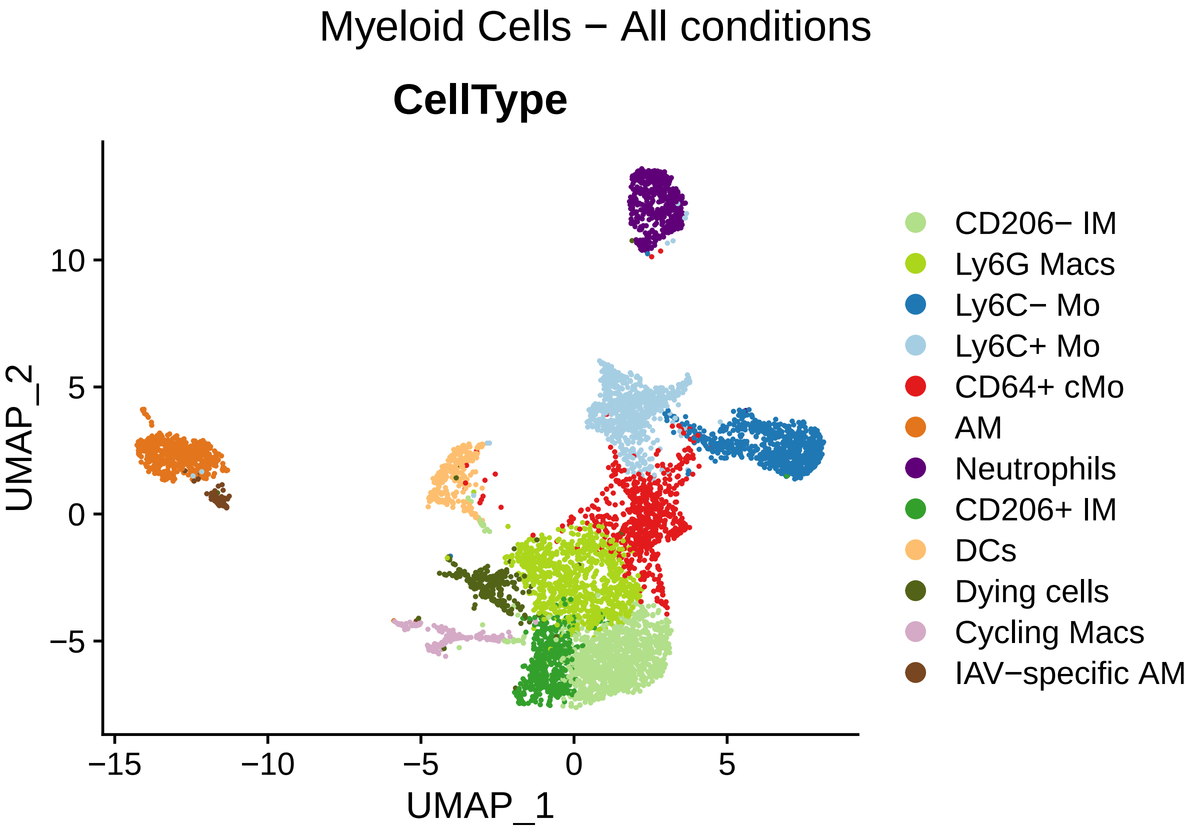
Adding the cell type to metadata

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

# 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")
```

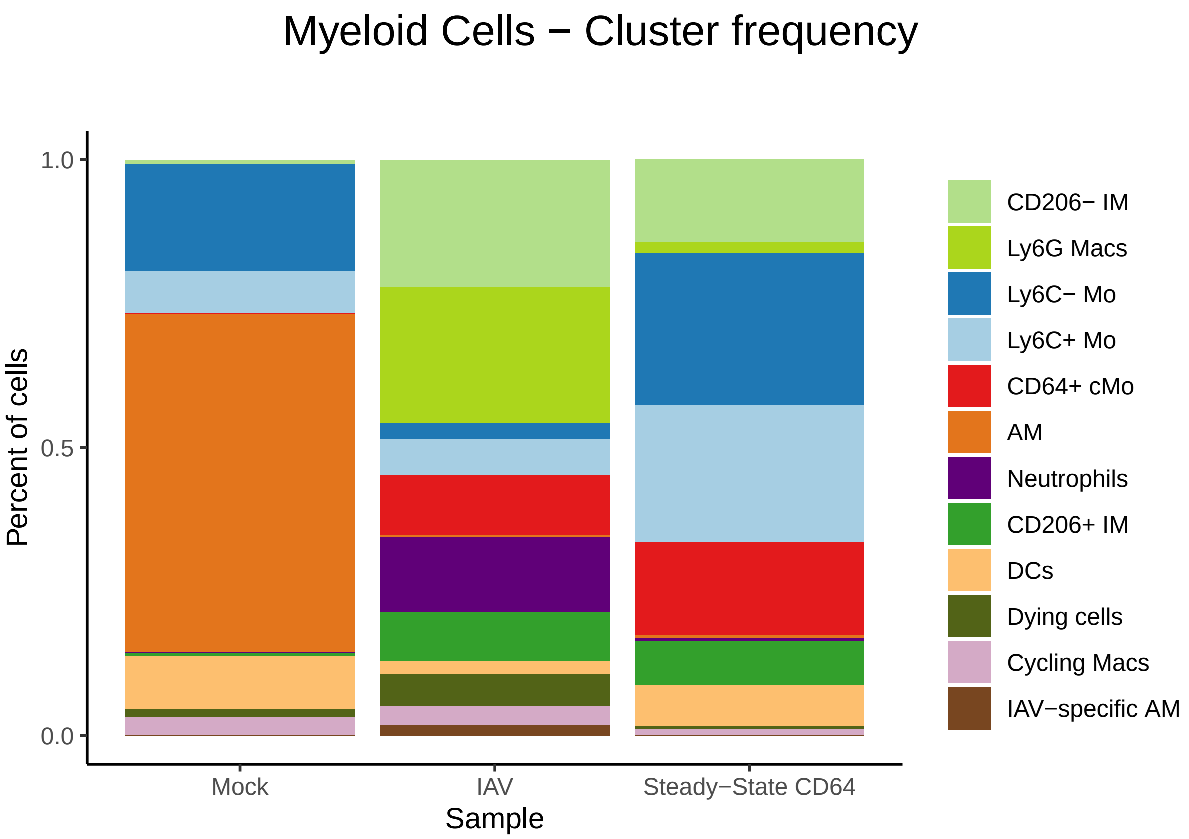
Visualsation of the frequency graph.

```
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.rotate = F, xlab = "Sample", x.labels = c("Mock","IAV","Steady-State CD64"))+
plot_annotation(
  title = 'Myeloid Cells - Cluster frequency',
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```



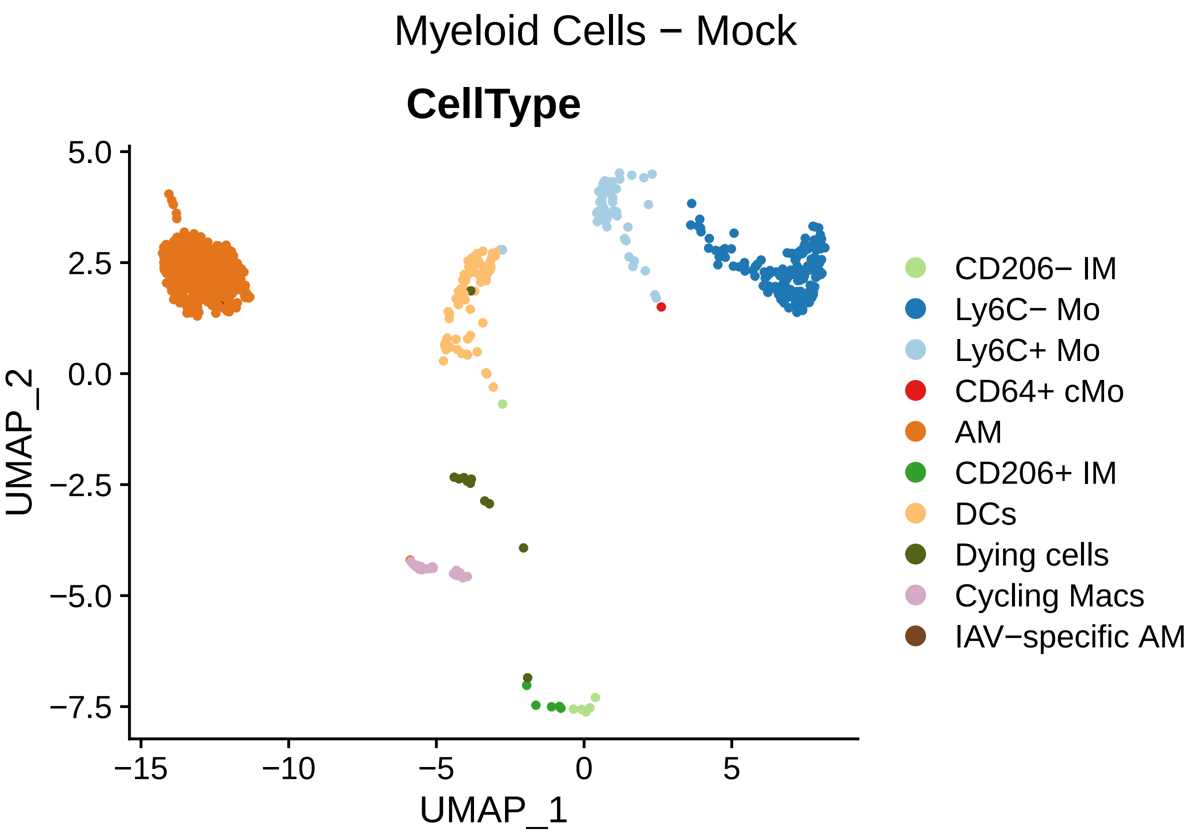
```
#ggsave("graphe_clusterFreq.pdf", height = 12 , width = 19, units = "cm")
```

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)))
```

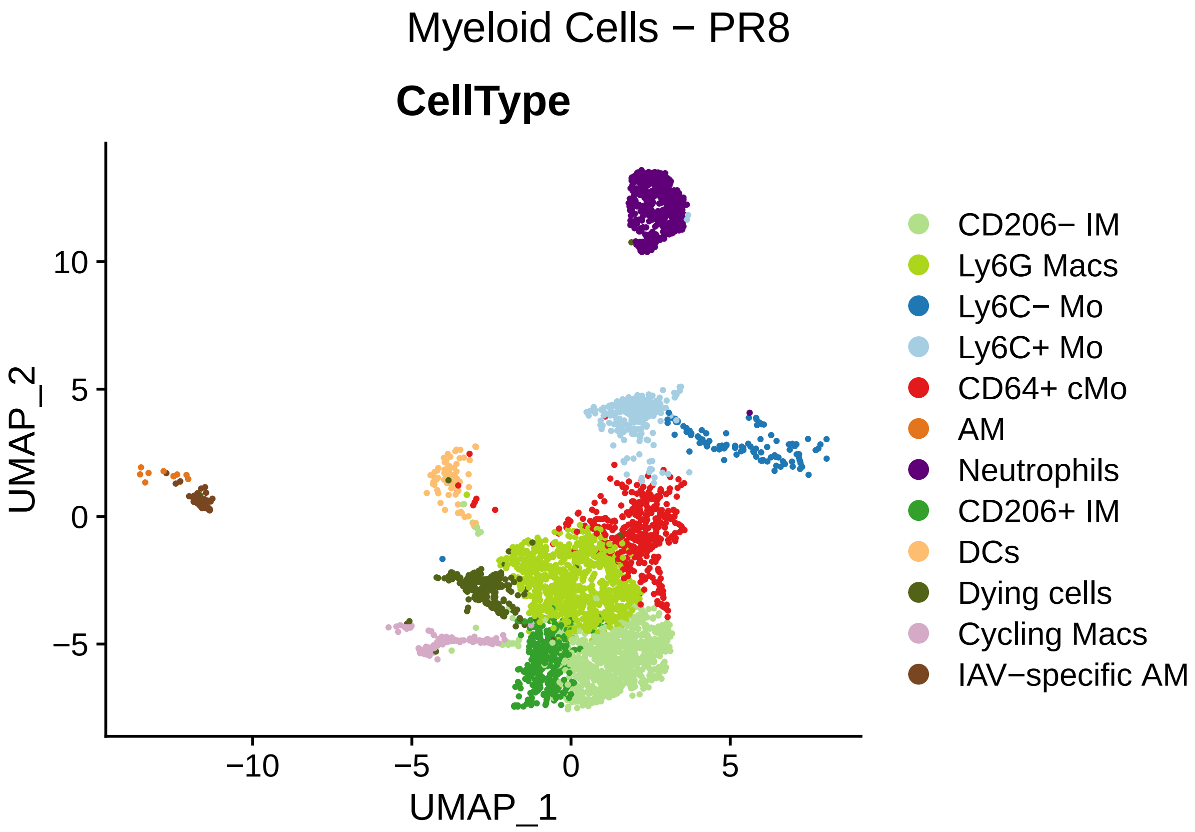


```r
#ggsave("UMAP_mock.pdf", height = 12 , width = 19, units = "cm")
```

Visualisation of the PR8 cells

```r
cell_myeloid_Pr8 <- subset(myeloid_cells, orig.ident %in% "myeloid_PR8")

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)))
```

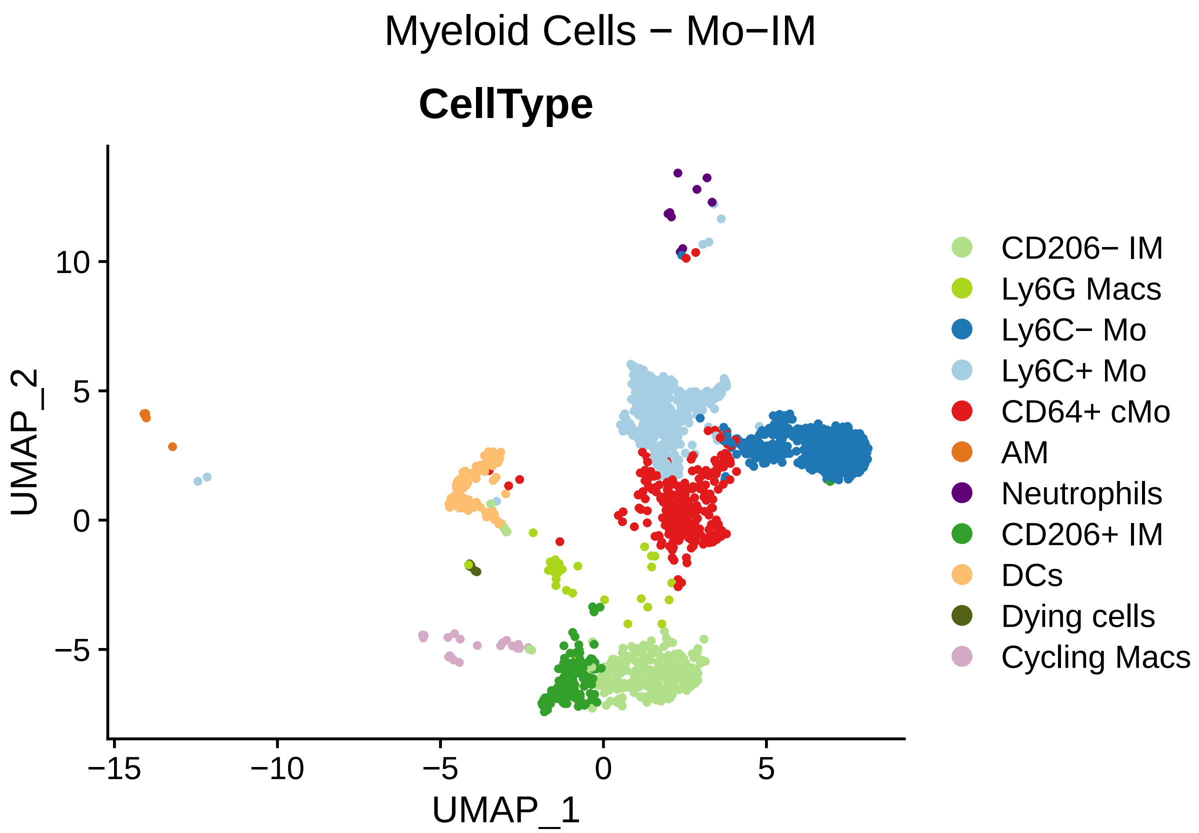


```r
#ggsave("UMAP_PR8.pdf", height = 12 , width = 19, units = "cm")
```

Visualisation IMcells

```r
cell_myeloid_IMcells <- subset(myeloid_cells, orig.ident %in% "Steady-State CD64")

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)))
```



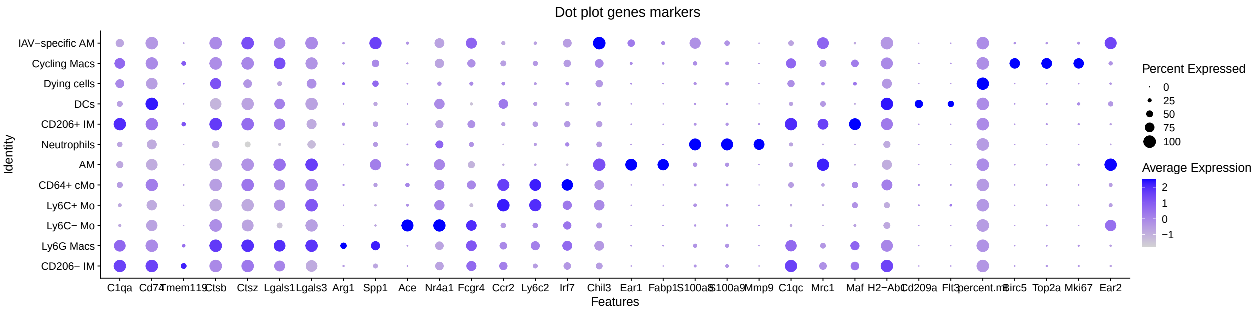
```
#ggsave("UMAP_PR8.pdf", height = 12 , width = 19, units = "cm")
```

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")
```

```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 formatR_1.14          dittoSeq_1.12.0      
## [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              progressr_0.13.0
##  [49] fansi_1.0.4                 spatstat.sparse_3.0-2
##  [51] httr_1.4.6                  polyclip_1.10-4
##  [53] abind_1.4-5                 compiler_4.3.3
##  [55] doParallel_1.0.17           withr_2.5.0
##  [57] highr_0.10                  MASS_7.3-60.0.1
##  [59] DelayedArray_0.26.3         rjson_0.2.21
##  [61] tools_4.3.3                 lmtest_0.9-40
##  [63] httpuv_1.6.11               future.apply_1.11.0
##  [65] goftest_1.2-3               glue_1.6.2
##  [67] nlme_3.1-164                promises_1.2.0.1
##  [69] Rtsne_0.16                  cluster_2.1.6
##  [71] reshape2_1.4.4              generics_0.1.3
##  [73] gtable_0.3.3                spatstat.data_3.0-1
##  [75] tidyr_1.3.0                 data.table_1.14.8
##  [77] XVector_0.40.0              sp_2.0-0
##  [79] utf8_1.2.3                  BiocGenerics_0.46.0
##  [81] spatstat.geom_3.2-4         RcppAnnoy_0.0.21
##  [83] foreach_1.5.2               ggrepel_0.9.3
##  [85] RANN_2.6.1                  pillar_1.9.0
##  [87] stringr_1.5.0               spam_2.9-1
##  [89] later_1.3.1                 circlize_0.4.15
##  [91] splines_4.3.3               lattice_0.22-5
##  [93] survival_3.5-8              deldir_1.0-9
##  [95] tidyselect_1.2.0            SingleCellExperiment_1.22.0
##  [97] miniUI_0.1.1.1              pbapply_1.7-2
##  [99] knitr_1.43                  gridExtra_2.3
## [101] IRanges_2.34.0              SummarizedExperiment_1.30.2
## [103] scattermore_1.2             stats4_4.3.3
## [105] xfun_0.39                   Biobase_2.60.0
## [107] matrixStats_1.0.0           pheatmap_1.0.12
## [109] stringi_1.7.12              lazyeval_0.2.2
## [111] yaml_2.3.7                  evaluate_0.21
## [113] codetools_0.2-19            tibble_3.2.1
## [115] cli_3.6.1                   uwot_0.1.16
## [117] xtable_1.8-4                reticulate_1.30
## [119] munsell_0.5.0               Rcpp_1.0.11
## [121] GenomeInfoDb_1.36.0         globals_0.16.2
## [123] spatstat.random_3.1-5       png_0.1-8
## [125] parallel_4.3.3              ellipsis_0.3.2
## [127] dotCall64_1.0-2             bitops_1.0-7
## [129] listenv_0.9.0               viridisLite_0.4.2
## [131] scales_1.2.1                ggridges_0.5.4
## [133] crayon_1.5.2                leiden_0.4.3
## [135] purrr_1.0.1                 GetoptLong_1.0.5
## [137] rlang_1.1.1                 cowplot_1.1.1
```