

Clustering endothelial cells

Joan Abinet

2025-06-25 14:30:15 +0200

Contents

Introduction	2
Loading packages	2
Load Seurat objects	2
Reclustering the cells	2
Annotating cells	3
Cluster Frequency per condition	4
Visualisation	5
Mock cells	5
aCXCR2 Day 3 cells	6
aLy6G Day 3 cells	7
aLy6G Day 14 cells	8
aCXCR2 Day 14 cells	9
Dotplot Annotated	10
Heatmap top genes per clusters	12
saving results for later	14

Introduction

A total of 11 clusters of endothelial cells were obtained using the FindClusters function (15 Principal Components were included and a resolutions of 0.25 was selected) and the differentially expressed genes (DEGs) were calculated using the FindAllMarkers function (Seurat package). Endothelial cells were subsequently categorized into 7 subpopulations using canonical marker genes.

Loading packages

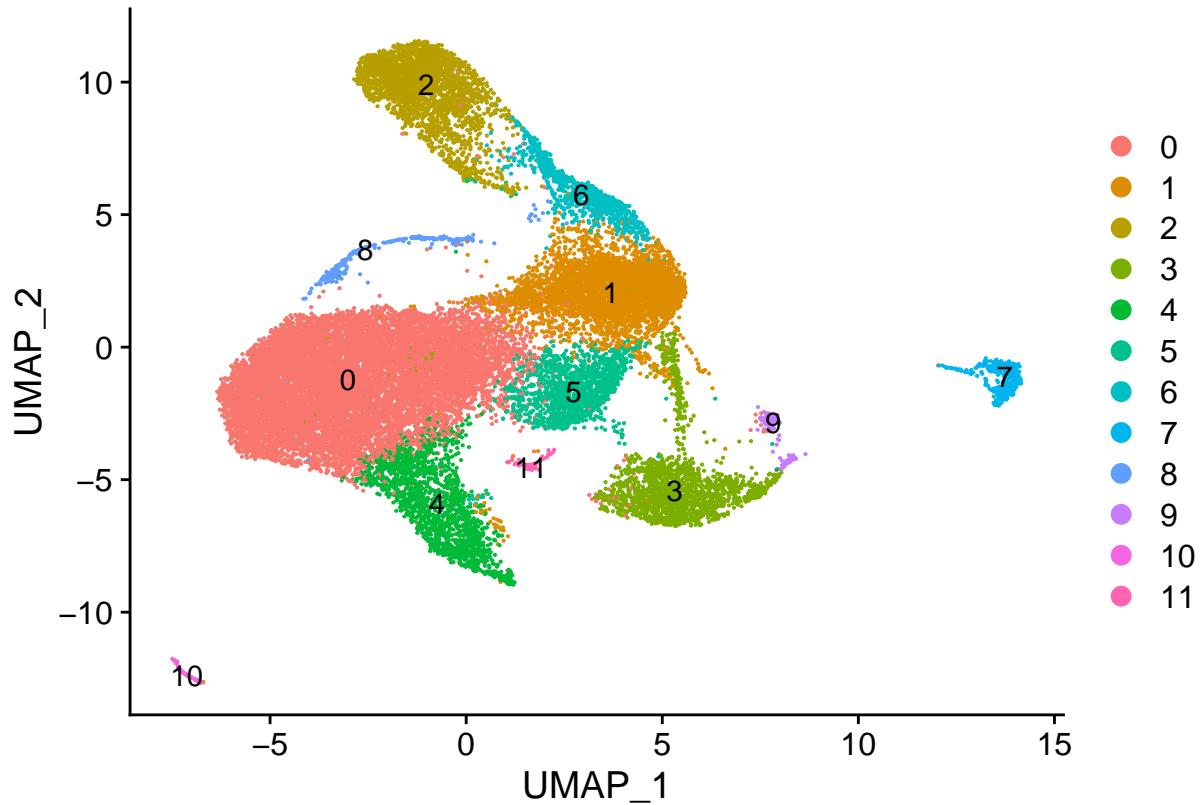
```
suppressMessages({  
  library(dplyr)  
  library(Seurat)  
  library(patchwork)  
  library(ggplot2)  
  library(formatR)  
  library(dittoSeq)  
  library(ComplexHeatmap)  
})
```

Load Seurat objects

```
endothelial_cells <- readRDS("../01-EndothelialCells_Preprocessing/endothelial_cells.rds")
```

Reclustering the cells

```
endothelial_cells <- FindNeighbors(endothelial_cells, dims = 1:15)  
endothelial_cells <- FindClusters(endothelial_cells, resolution = 0.25)  
endothelial_cells <- RunUMAP(endothelial_cells, dims = 1:15)  
  
DimPlot(endothelial_cells, reduction = "umap", label = T)
```



Annotating cells

```

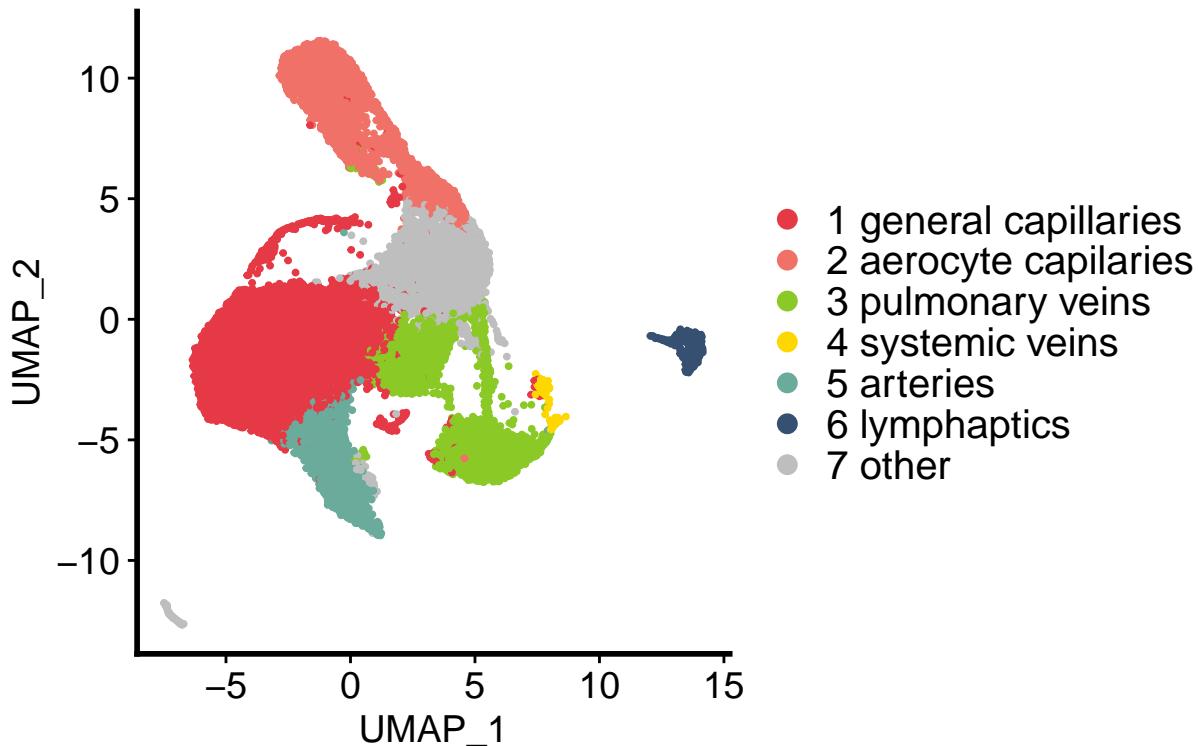
endothelial_cells$CellType <- endothelial_cells$seurat_clusters
levels(endothelial_cells$CellType) <- c("1 general capillaries", "7 other",
  "2 aerocyte capilaries", "3 pulmonary veins", "5 arteries", "3 pulmonary veins",
  "2 aerocyte capilaries", "6 lymphaptics", "1 general capillaries",
  "4 systemic veins", "7 other", "1 general capillaries")

endothelial_cells$CellType <- factor(endothelial_cells$CellType, levels = c("1 general capillaries",
  "2 aerocyte capilaries", "3 pulmonary veins", "4 systemic veins",
  "5 arteries", "6 lymphaptics", "7 other"))
Idents(endothelial_cells) <- "CellType"

color <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")

DimPlot(endothelial_cells, group.by = "CellType", cols = color, pt.size = 0.7) +
  theme(legend.text = element_text(size = 15), axis.line = element_line(linewidth = 1),
  axis.text = element_text(size = 14)) + ggtitle("")

```

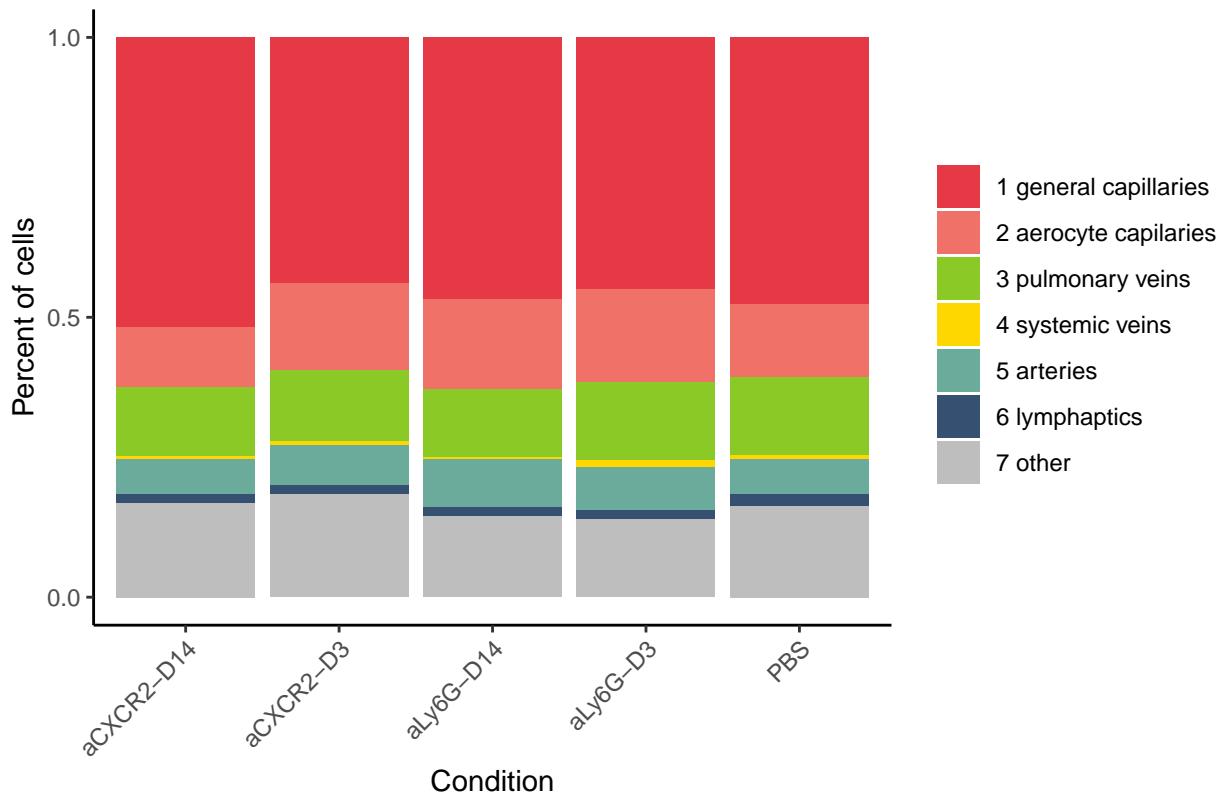


```
# ggsave('Plot/UMAP_whole.pdf', height = 12 , width = 19, units
# = 'cm')
```

Cluster Frequency per condition

```
endothelial_cells$Condition <- endothelial_cells$orig.ident
endothelial_cells$Condition <- as.factor(endothelial_cells$Condition)
levels(endothelial_cells$Condition) <- c("aCXCR2-D14", "aCXCR2-D3",
  "aLy6G-D14", "aLy6G-D3", "PBS")

bar_colors <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")
dittoBarPlot(endothelial_cells, "CellType", group.by = "Condition",
  color.panel = bar_colors, main = "")
```



```
# ggsave('bar_plot.pdf', width = 5, height = 3)
```

Visualisation

Mock cells

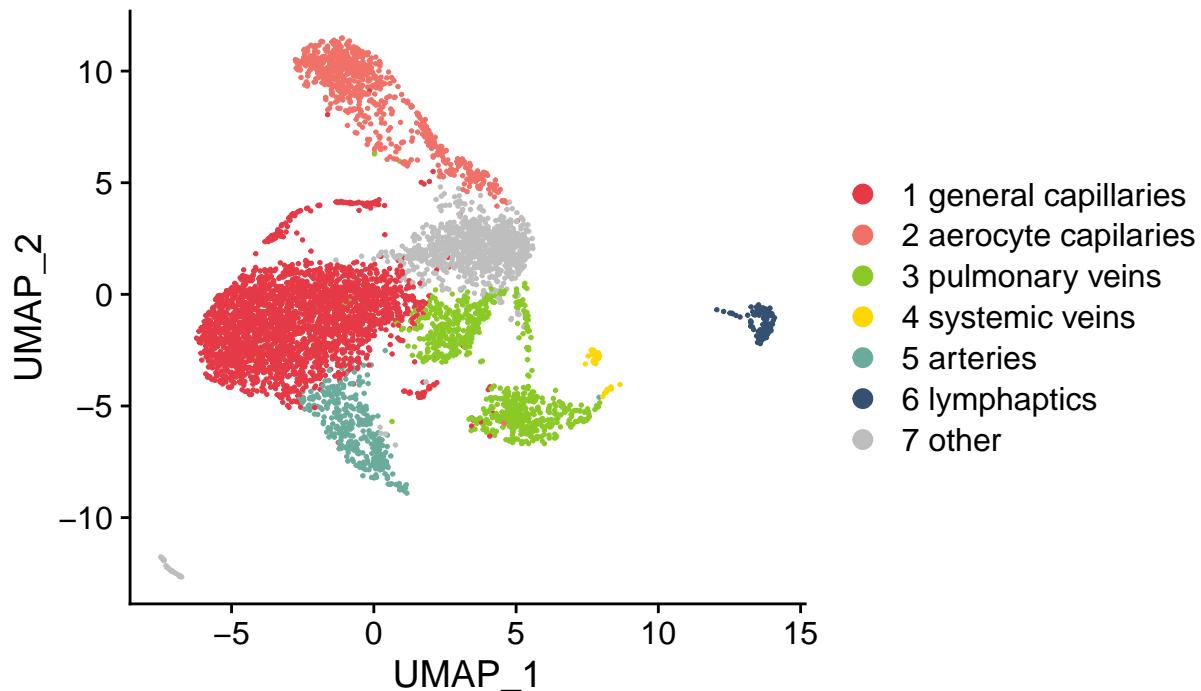
```
endothelial_cells_PBS <- subset(endothelial_cells, Condition %in%
  "PBS")

color <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")

DimPlot(endothelial_cells_PBS, reduction = "umap", cols = color, group.by = "CellType") +
  plot_annotation(title = "Endothelial - Cells - PBS ", theme = theme(plot.title = element_text(size =
    hjust = 0.5)))
```

Endothelial – Cells – PBS

CellType



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

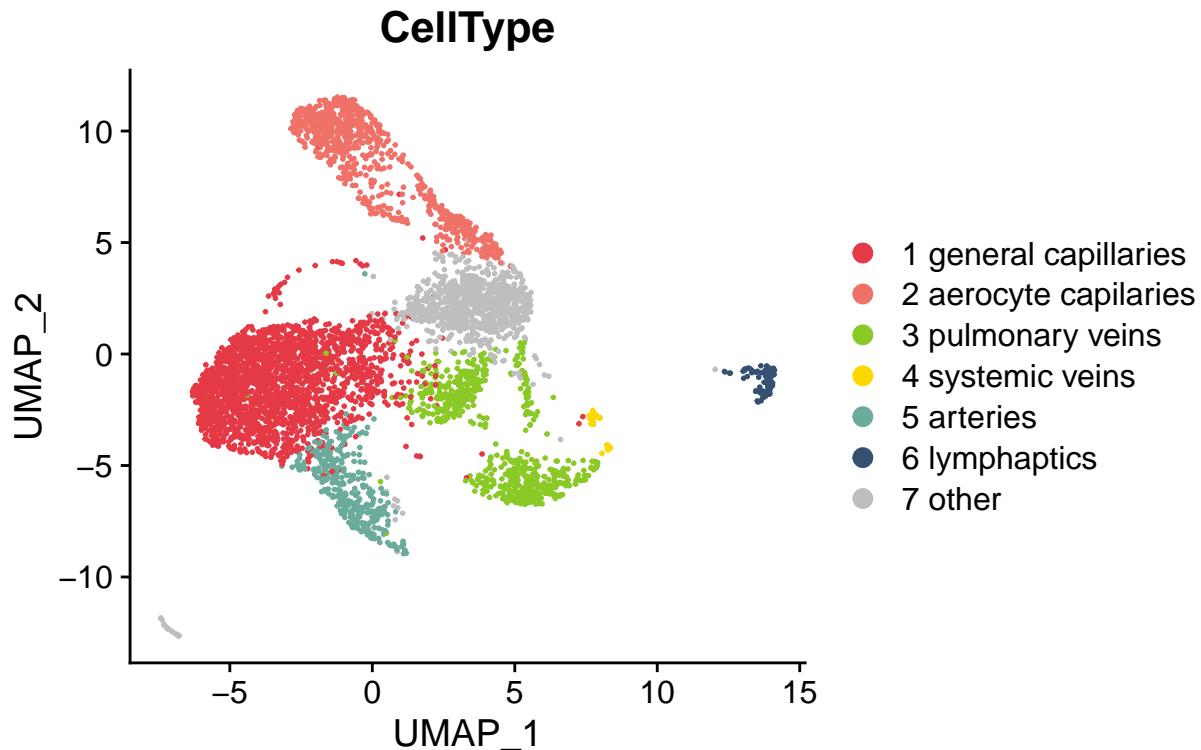
aCXCR2 Day 3 cells

```
endothelial_cells_aCXCR2_3 <- subset(endothelial_cells, Condition %in%
  "aCXCR2-D3")

color <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")

DimPlot(endothelial_cells_aCXCR2_3, reduction = "umap", cols = color,
  group.by = "CellType") + plot_annotation(title = "Endothelial Cells - aCXCR2 Day 3 ",
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

Endothelial Cells – aCXCR2 Day 3



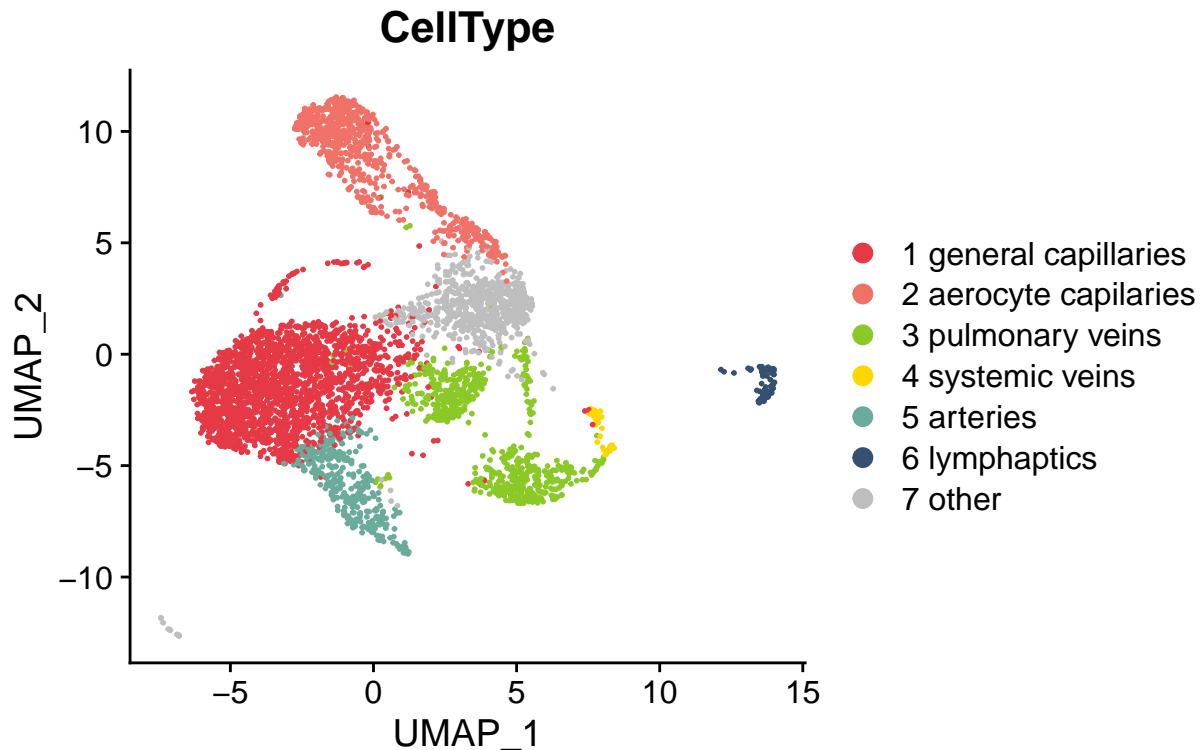
aLy6G Day 3 cells

```
endothelial_cells_aLy6G_3 <- subset(endothelial_cells, Condition %in%
  "aLy6G-D3")

color <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")

DimPlot(endothelial_cells_aLy6G_3, reduction = "umap", cols = color,
  group.by = "CellType") + plot_annotation(title = "Endothelial Cells - aLy6G Day 3 ",
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

Endothelial Cells – aLy6G Day 3



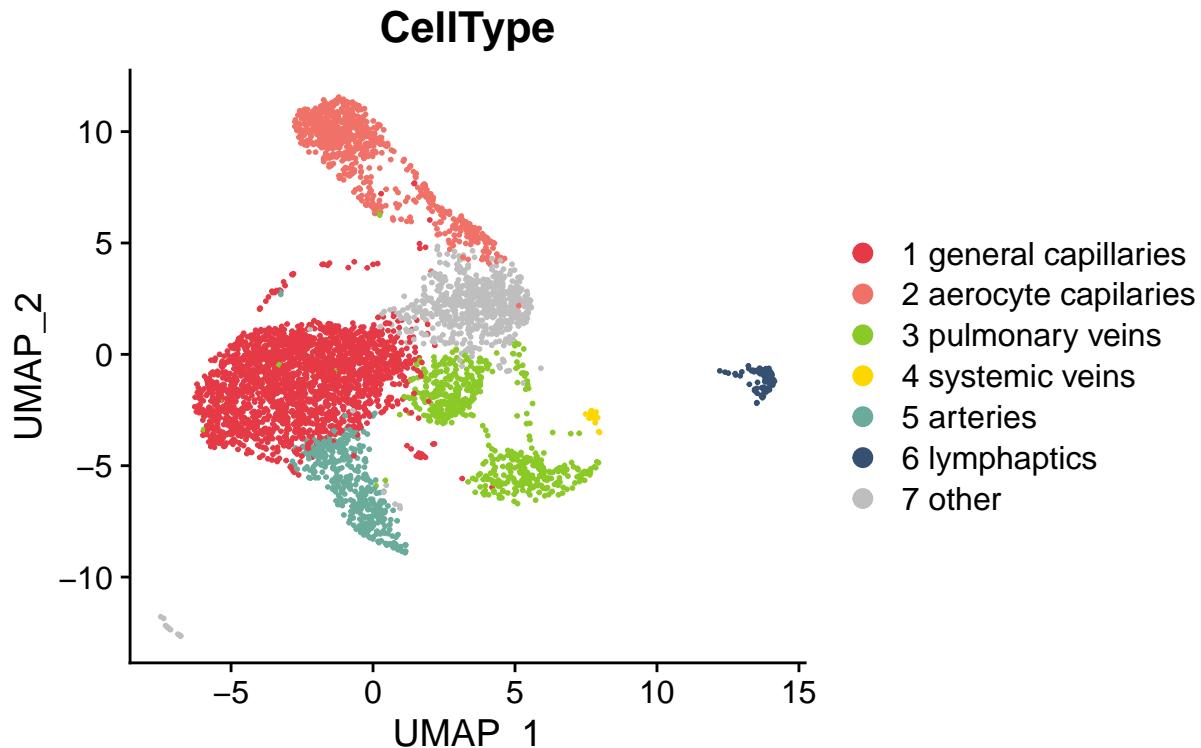
aLy6G Day 14 cells

```
endothelial_cells_aLy6G_14 <- subset(endothelial_cells, Condition %in%
  "aLy6G-D14")

color <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")

DimPlot(endothelial_cells_aLy6G_14, reduction = "umap", cols = color,
  group.by = "CellType") + plot_annotation(title = "Endothelial Cells - aLy6G Day 14 ",
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

Endothelial Cells – aLy6G Day 14



```
# ggsave('Plot/UMAP_aLy6G_D14.pdf', height = 12 , width = 19,
# units = 'cm')
```

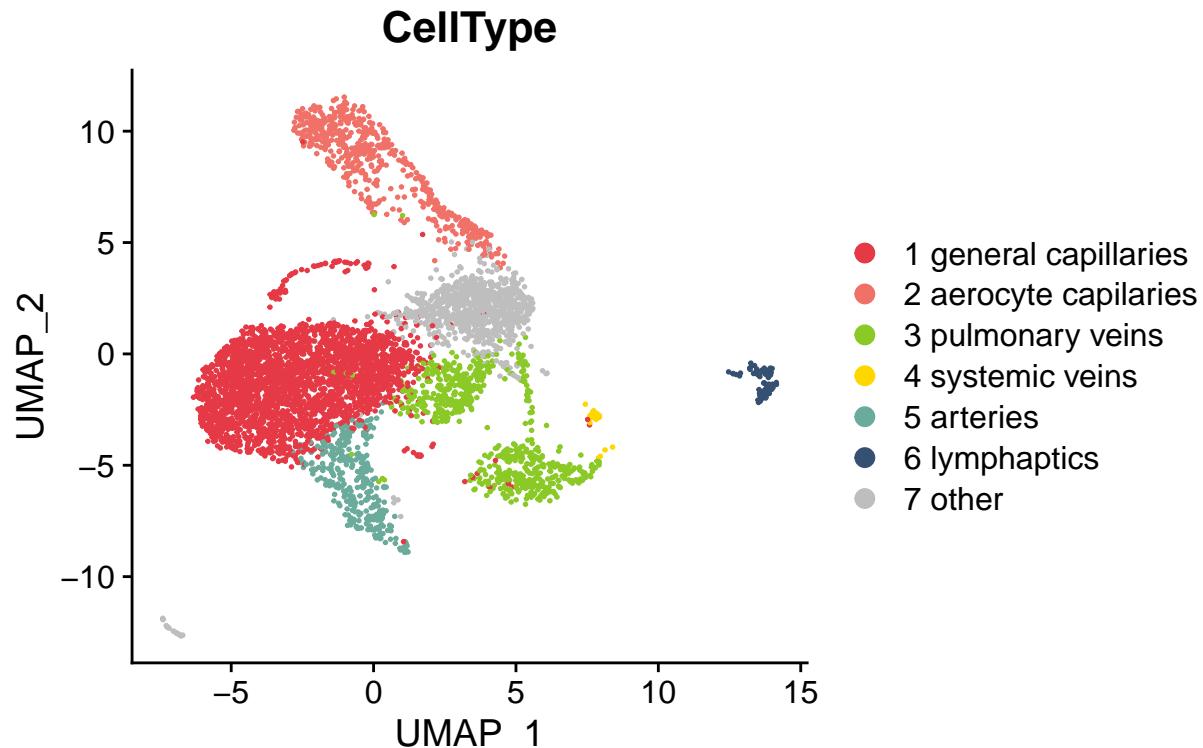
aCXCR2 Day 14 cells

```
endothelial_cells_aCXCR2_14 <- subset(endothelial_cells, Condition %in%
  "aCXCR2-D14")

color <- c("#E63946", "#F07167", "#8AC926", "#FFD700", "#6AAB9C",
  "#355070", "gray")

DimPlot(endothelial_cells_aCXCR2_14, reduction = "umap", cols = color,
  group.by = "CellType") + plot_annotation(title = "Endothelial Cells - aCXCR2 Day 14 ",
  theme = theme(plot.title = element_text(size = 16, hjust = 0.5)))
```

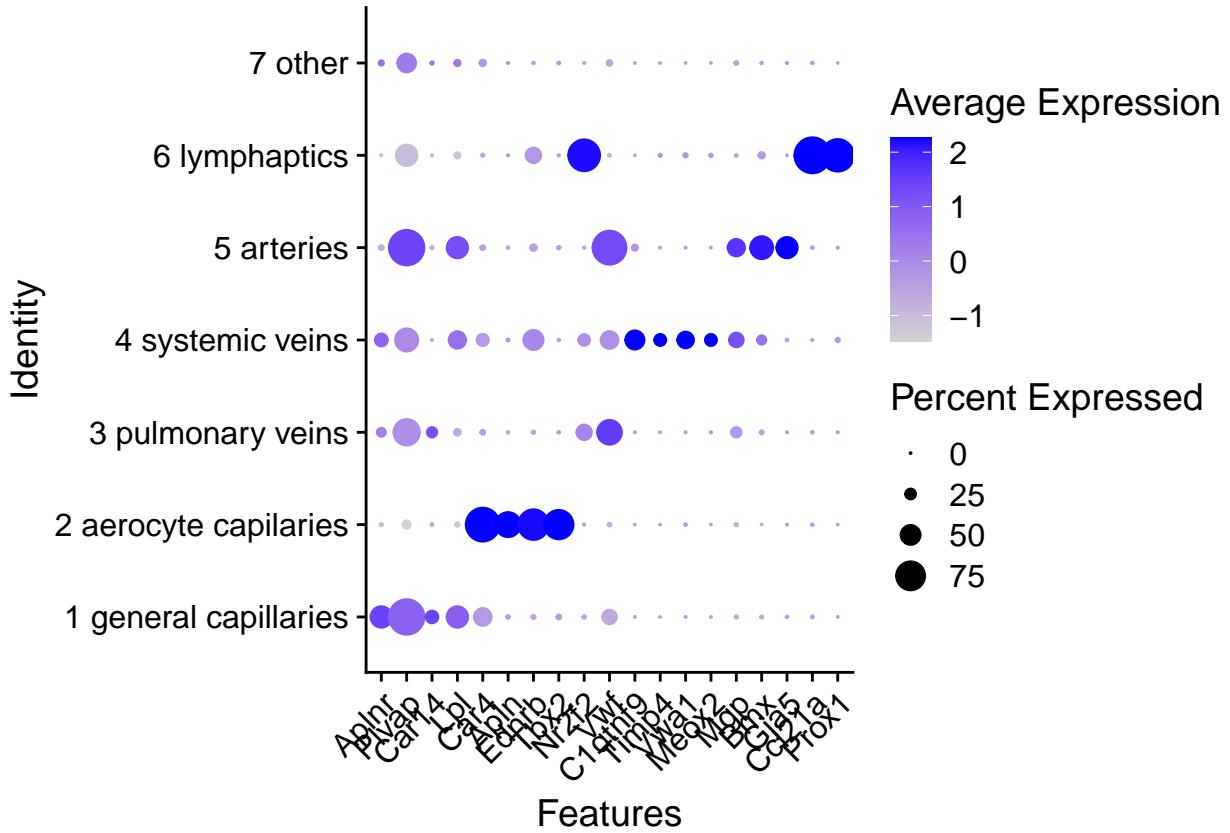
Endothelial Cells – aCXCR2 Day 14



Dotplot Annotated

```
DotPlot(endothelial_cells, features = c("Aplnr", "Col93", "Plvap",
  "Car14", "Lpl", "Car4", "Apln", "Ednrb", "Tbx2", "Nr2f2", "Vwf",
  "C1qtnf9", "Timp4", "Vwa1", "Meox2", "Mgp", "Bmx", "Gja5", "Ccl21a",
  "Prox1"), group.by = "CellType") + theme(axis.text.x = element_text(angle = 45,
  hjust = 1))

## Warning in FetchData.Seurat(object = object, vars = features, cells = cells):
## The following requested variables were not found: Col93
```



```
# ggsave('dotplot_endo.pdf', width = 8, height = 6)

endo.markers <- FindAllMarkers(endothelial_cells, only.pos = TRUE,
                                min.pct = 0.25)

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

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

df <- as.data.frame(endothelial_cells$CellType)
colnames(df) <- "Clusters"
color_df <- list(Clusters = c(`1 general capillaries` = "#E63946",
                               `2 aerocyte capilaries` = "#F07167", `3 pulmonary veins` = "#8AC926",
                               `4 systemic veins` = "#FFD700", `5 arteries` = "#6AAB9C", `6 lymphaptics` = "#355070",
                               `7 other` = "grey"))

Heatmap <- Heatmap(t(scale(t(mat))), show_column_names = F, column_split = endothelial_cells$CellType,
                     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, 'Plot/Heatmap_endo.pdf', width
# = 35, height = 50, units ='cm')
```

Heatmap top genes per clusters

Heatmap



saving results for later

```
saveRDS(endothelial_cells, "endothelial_cells_annotated.rds")  
  
sessionInfo()  
  
## R version 4.3.3 (2024-02-29)  
## Platform: x86_64-pc-linux-gnu (64-bit)  
## Running under: Ubuntu 24.04.2 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-p0.3.26.so;  LAPACK version 3.12.0  
##  
## 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      formatR_1.14  
## [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
## [41] digest_0.6.33
## [43] S4Vectors_0.38.1
## [45] irlba_2.3.5.1
## [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] stringr_1.5.0
## [89] spam_2.9-1
## [91] circlize_0.4.15
## [93] lattice_0.22-5
## [95] deldir_1.0-9
## [97] SingleCellExperiment_1.22.0
## [99] pbapply_1.7-2
## [101] gridExtra_2.3
## [103] SummarizedExperiment_1.30.2
## [105] stats4_4.3.3
## [107] Biobase_2.60.0
## [109] pheatmap_1.0.12
## [111] lazyeval_0.2.2
## [113] evaluate_0.21
## [115] tibble_3.2.1
## [117] uwot_0.1.16
## [119] reticulate_1.30
## [121] Rcpp_1.0.11
## [123] globals_0.16.2
## [125] png_0.1-8
## [127] ellipsis_0.3.2
## [129] bitops_1.0-7
## [131] viridisLite_0.4.2
## [133] ggridges_0.5.4
## [135] leiden_0.4.3
## [137] GetoptLong_1.0.5
## [139] cowplot_1.1.1

shiny_1.7.4.1
colorspace_2.1-0
tensor_1.5
GenomicRanges_1.52.0
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_2.2-0
BiocGenerics_0.46.0
RcppAnnoy_0.0.21
ggrepel_0.9.3
pillar_1.9.0
limma_3.56.2
later_1.3.1
splines_4.3.3
survival_3.5-8
tidyselect_1.2.0
miniUI_0.1.1.1
knitr_1.43
IRanges_2.34.0
scattermore_1.2
xfun_0.39
matrixStats_1.0.0
stringi_1.8.4
yaml_2.3.7
codetools_0.2-19
cli_3.6.1
xtable_1.8-4
munsell_0.5.0
GenomeInfoDb_1.36.0
spatstat.random_3.1-5
parallel_4.3.3
dotCall64_1.0-2
listenv_0.9.0
scales_1.2.1
crayon_1.5.2
purrr_1.0.1
rlang_1.1.1

```