gastD11

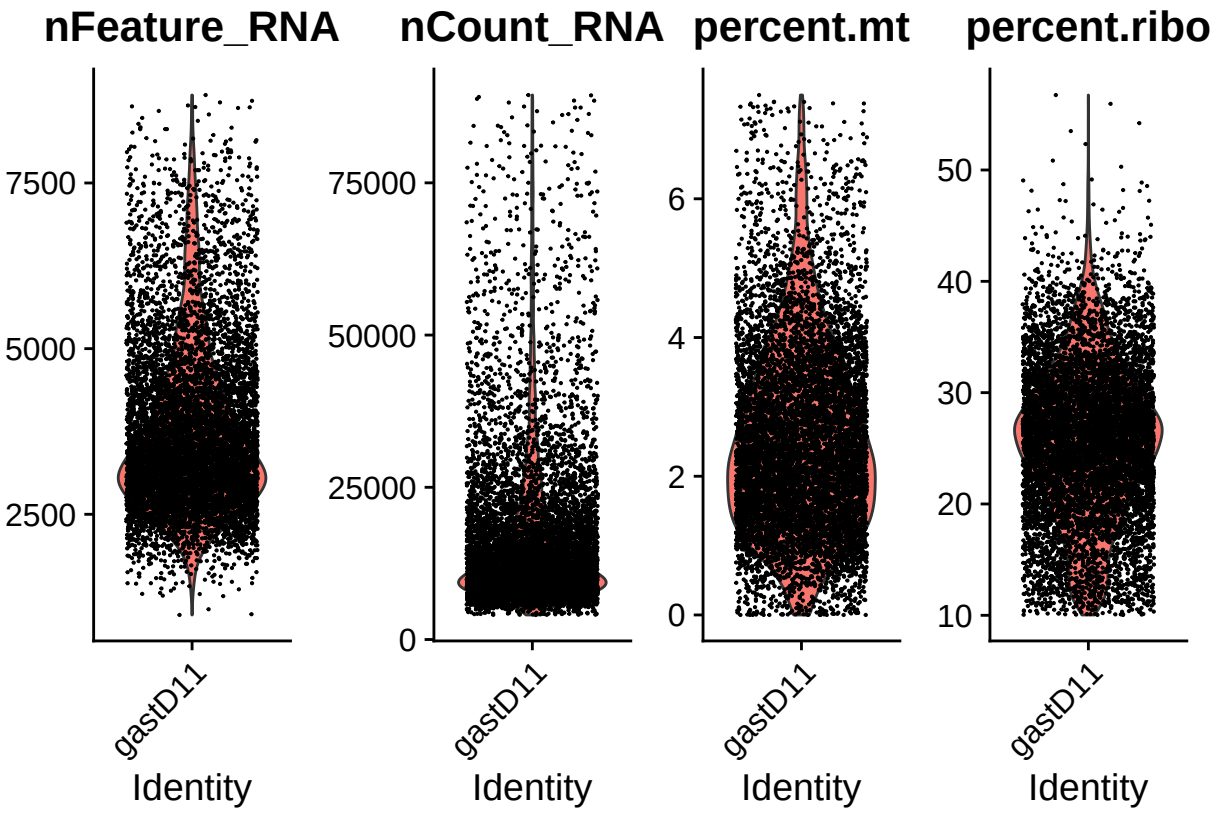
BULAVKA

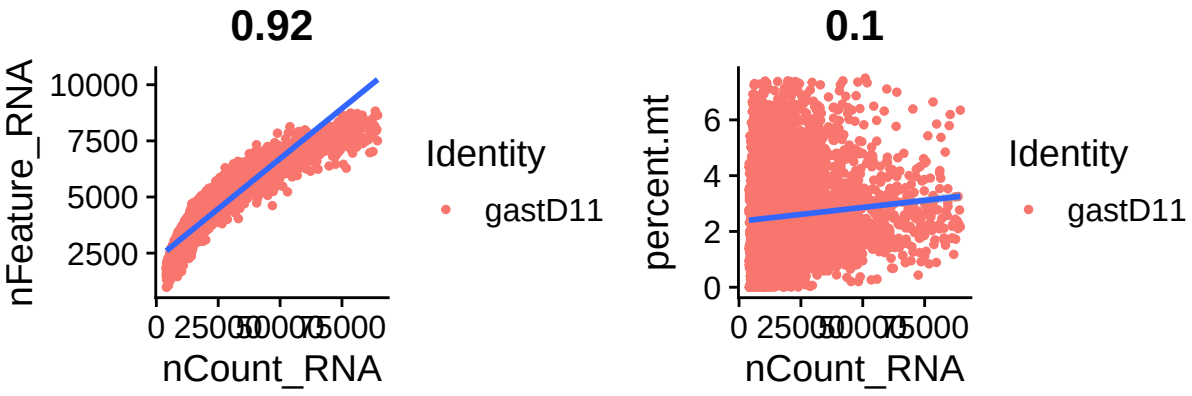
2023-05-17

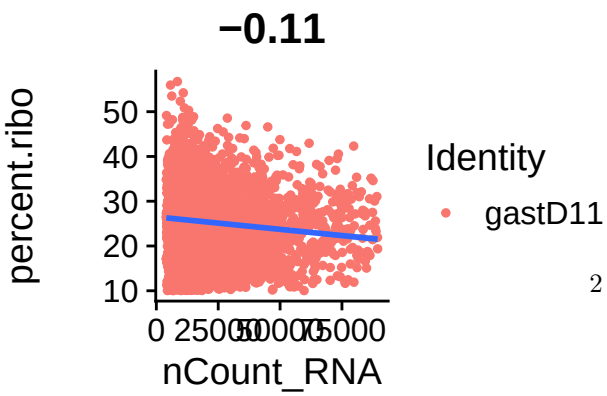
1 Load the dataset

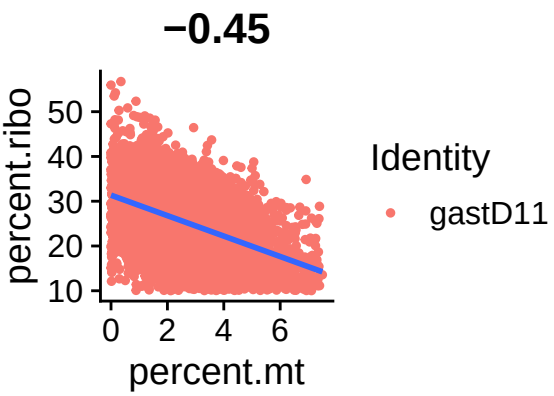
2. Parametres

3. QC



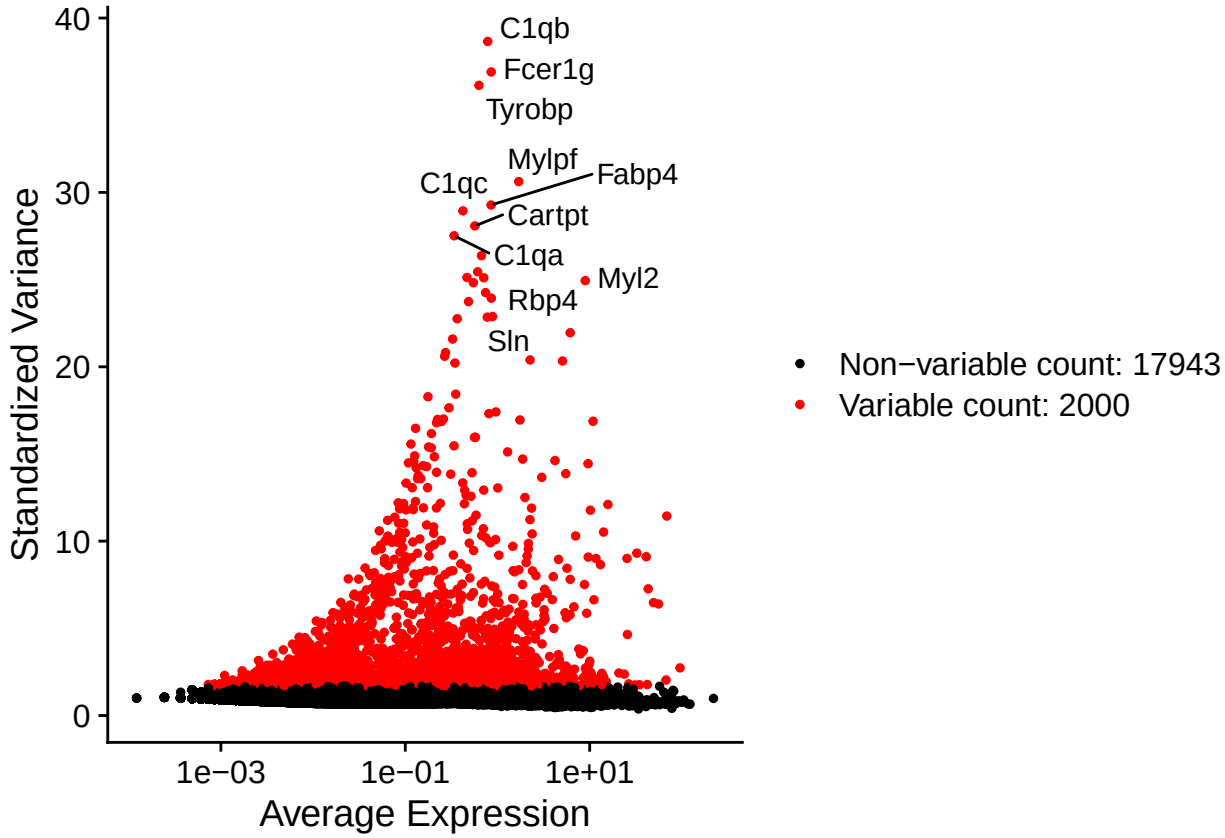






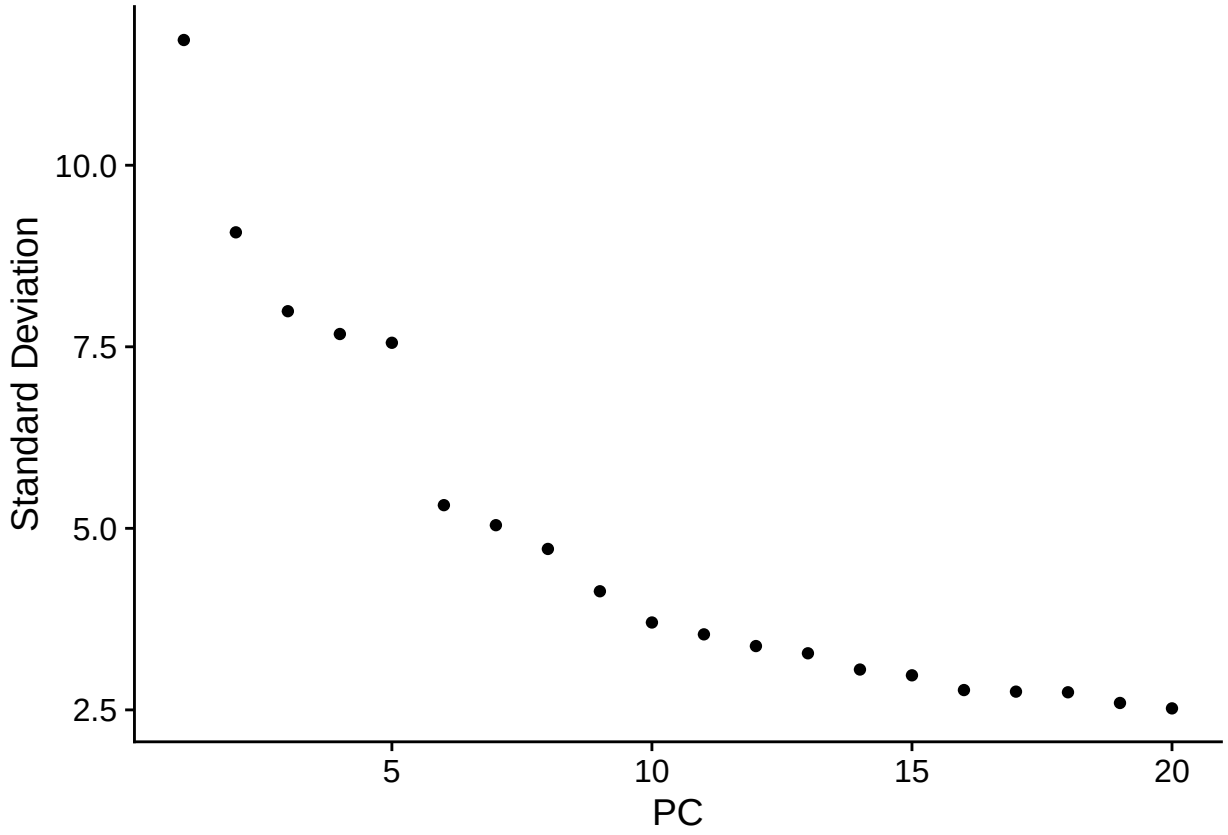
4. Normalization and Identification of highly variable features (feature selection)

```
gs <- NormalizeData(gs, normalization.method = norm_method, scale.factor = 10000, verbose = FALSE)
gs <- FindVariableFeatures(gs, selection.method = var_feat_method, nfeatures = 2000, verbose = FALSE)
## Identify the 10 most highly variable genes----------------------------------
top20 <- head(VariableFeatures(gs), 20)
## plot variable features with and without labels ----------------------------
plot1 <- VariableFeaturePlot(gs)
plot2 <- LabelPoints(plot = plot1, points = top20, repel = TRUE)
plot2
```

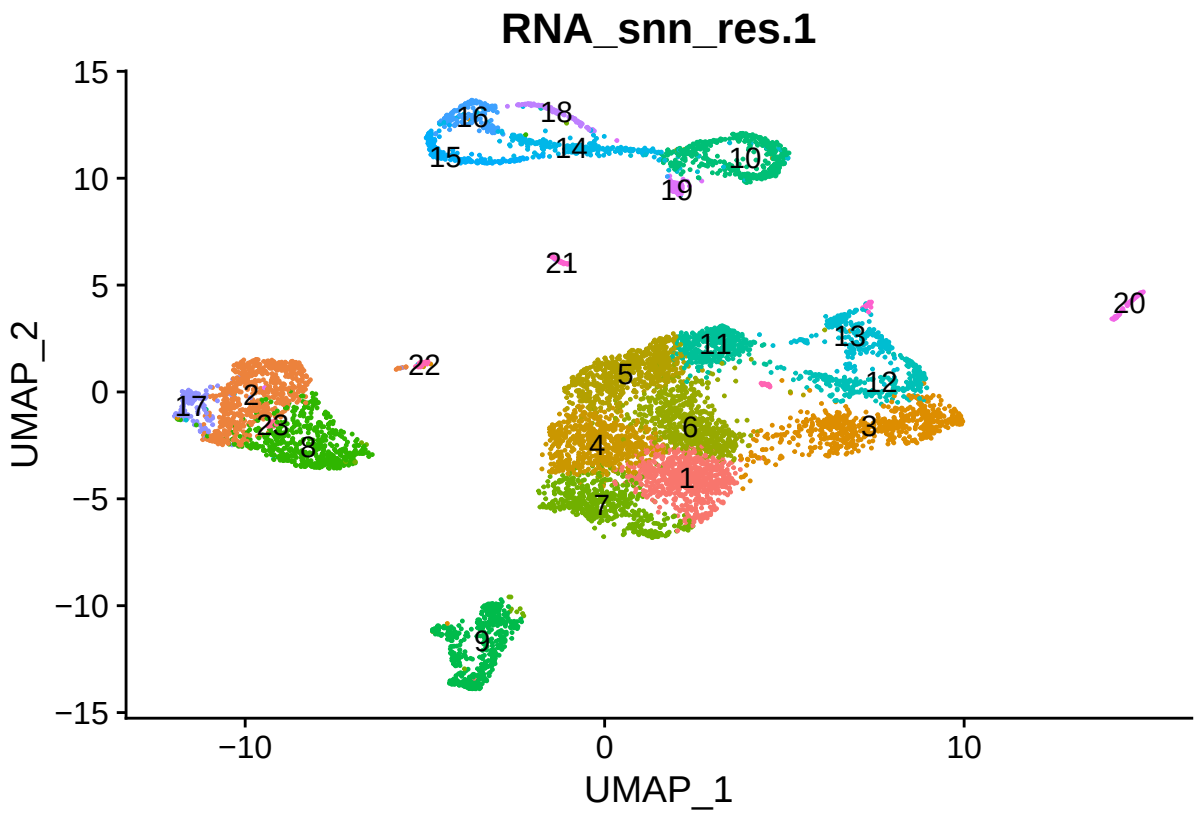


5. Scaling the data and perform linear dimensional reduction

```
all.genes <- rownames(gs)
gs <- ScaleData(gs, features = all.genes, verbose = FALSE)
gs <- RunPCA(gs, features = VariableFeatures(object = gs), npcs = 20, nfeatures.print = 10, verbose=TRU
ElbowPlot(gs, ndims = 20)
```

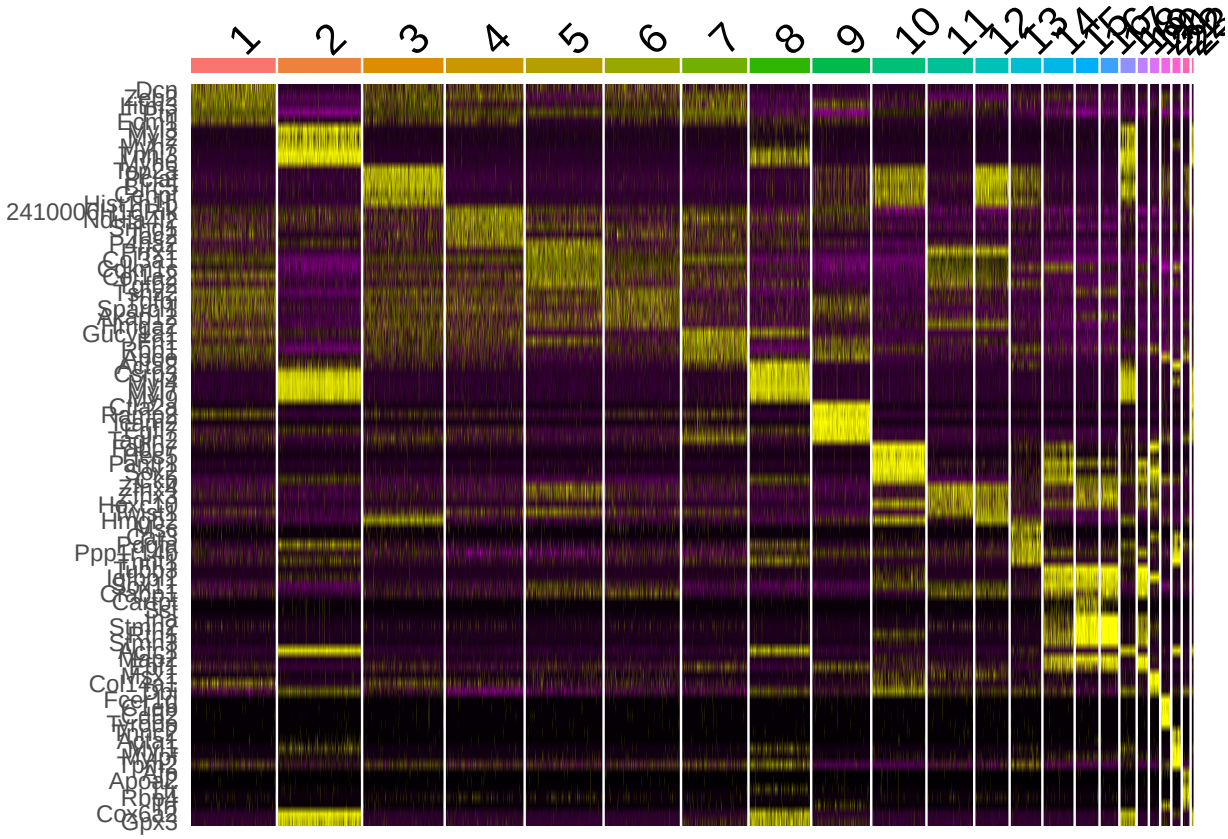



```
gs <- FindNeighbors(gs, dims = 1:20, verbose=FALSE)
gs <- FindClusters(gs, resolution = res_var, algorithm = Leiden, random.seed = general.seed) # algorith
gs <- RunUMAP(gs, dims = 1:20, seed.use = general.seed, verbose = FALSE)
DimPlot(gs, group.by = "RNA_snn_res.1" , reduction = "umap", label = TRUE) + NoLegend()
```



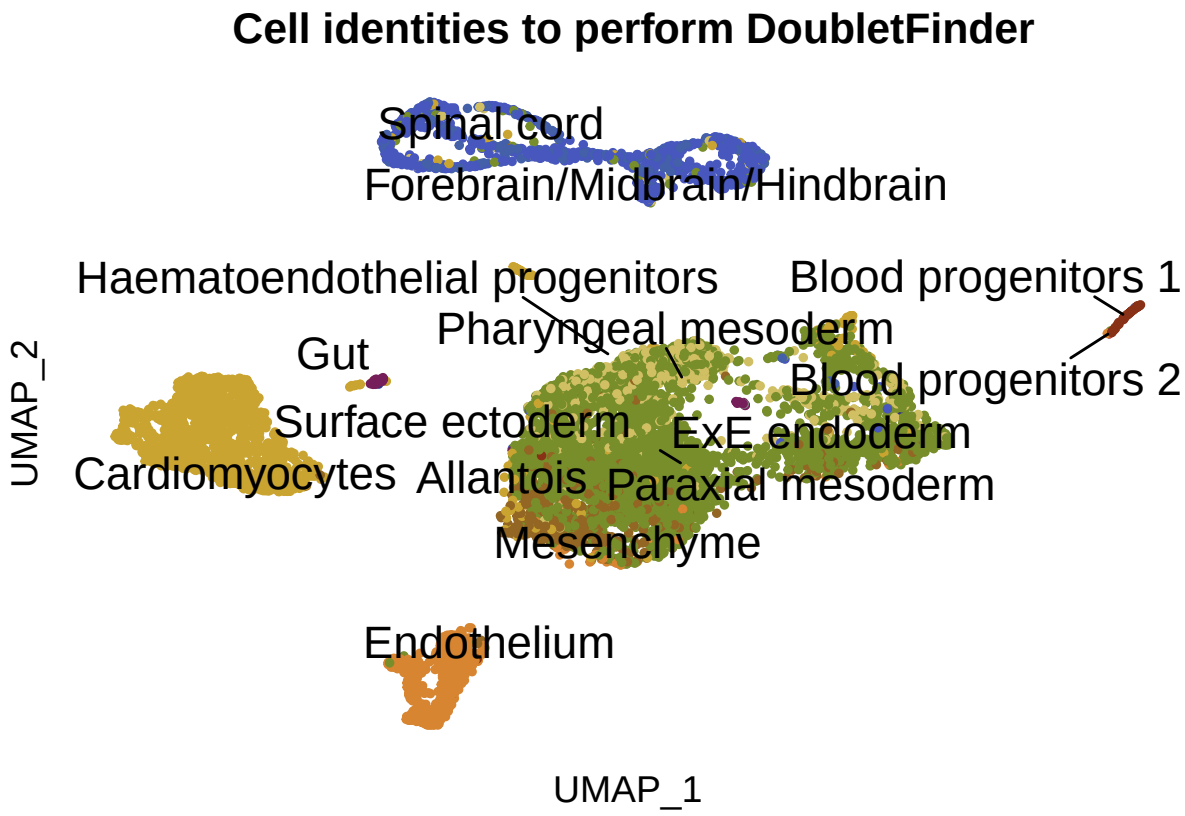
6. Finding differentially expressed features (cluster biomarkers)

```
gs.markers <- FindAllMarkers(gs, group.by = "RNA_snn_res.1", only.pos = TRUE, min.pct = 0.25, logfc.thr
gs.markers %>%
  group_by(cluster) %>%
  top_n(n = 5, wt = avg_log2FC) -> top10
DoHeatmap(gs,  group.by = "RNA_snn_res.1", features = top10$gene) + NoLegend()
```



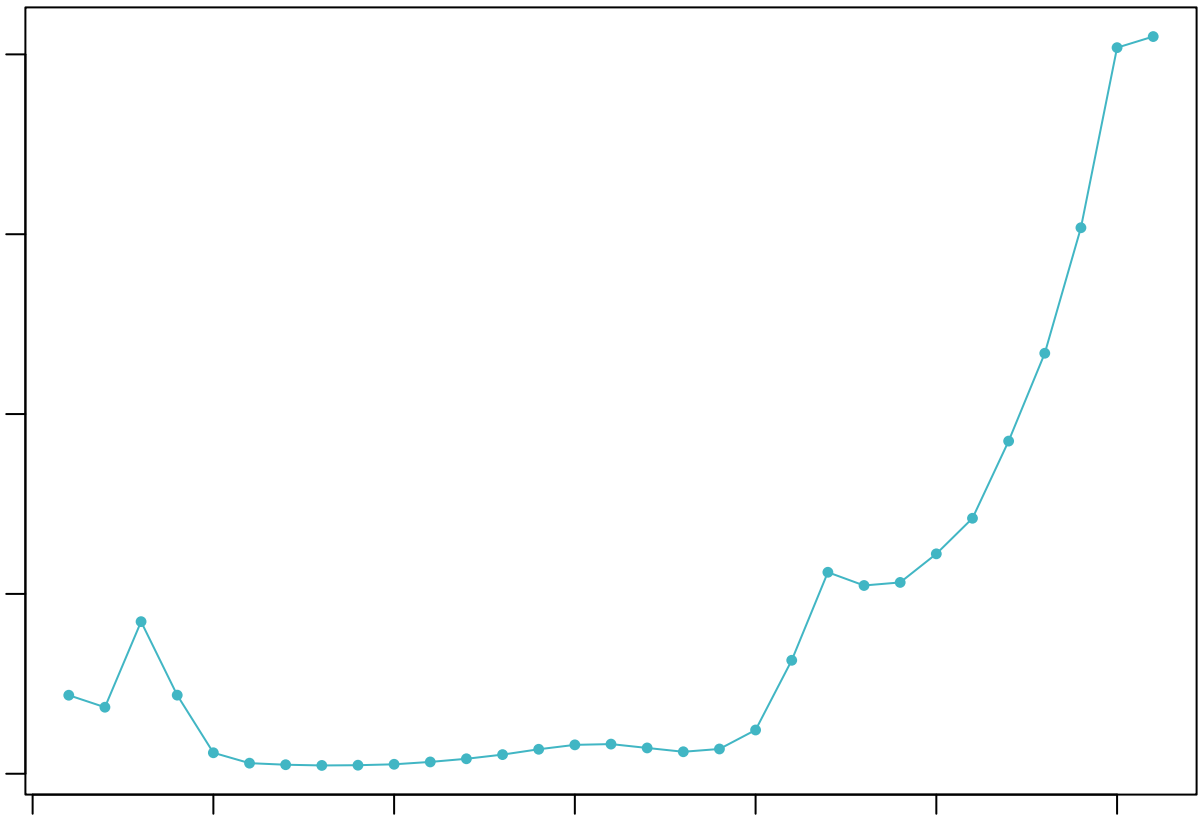
7. Doubets Finder

```
atlas.subset <- readRDS("/mnt/DATA_4TB/projects/gastruloids_sc_Lescroart/analysis/embryE9gastD10/50_rds
anchors <- FindTransferAnchors(reference = atlas.subset, query = gs,
                               dims = 1:20, reference.reduction = "pca")
predictions <- TransferData(anchorset = anchors, refdata = atlas.subset$celltype,
                            dims = 1:20)
gs <- AddMetaData(gs, metadata = predictions[,"predicted.id"], col.name = "celltype_DF")

#
Idents(gs) <- gs@meta.data$celltype_DF

p1 <- DimPlot(gs,
        pt.size = 1,
        repel = TRUE,
        label = TRUE,
        label.size = 6,
        cols = colors.celltype[levels(Idents(gs))]) +
  ggtitle( " Cell identities to perform DoubletFinder") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text = element_blank(),
        line = element_blank()) +
  NoLegend()
p1
```

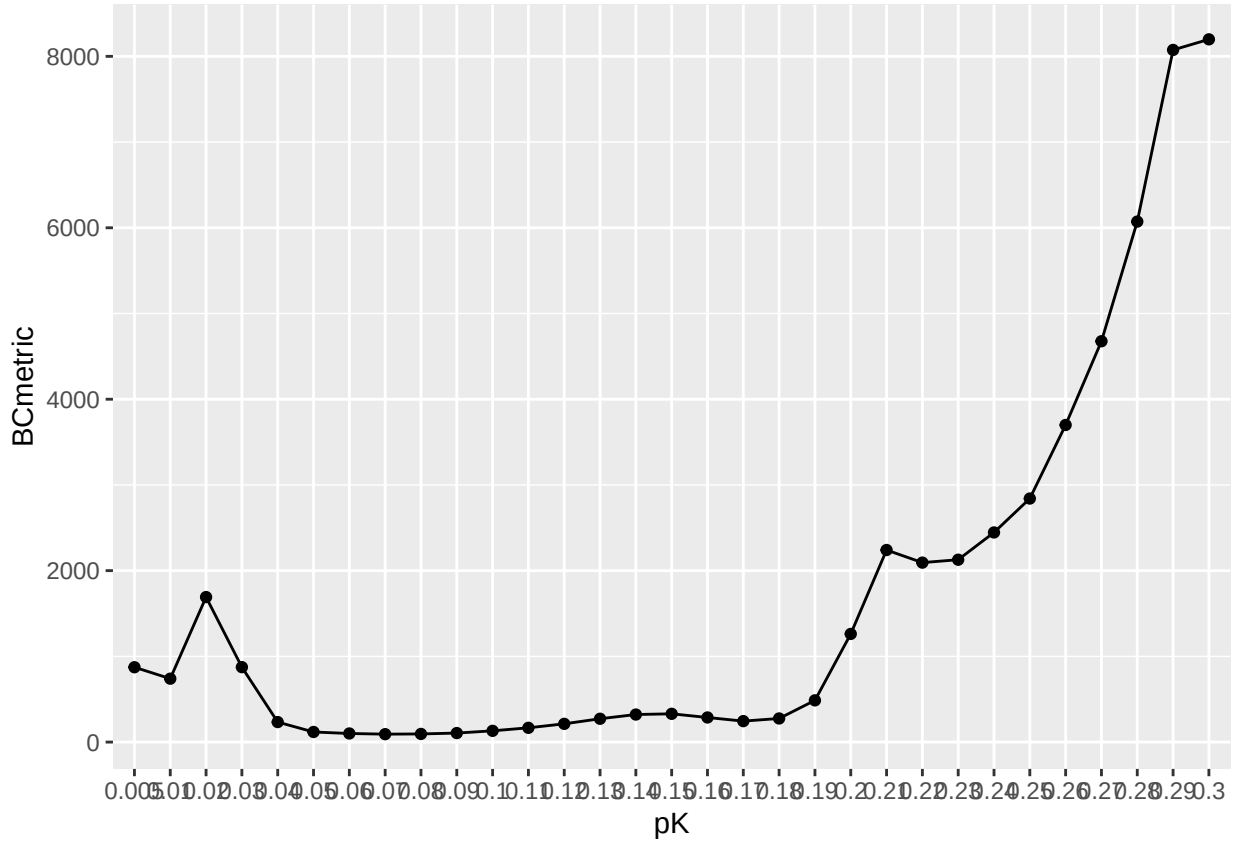


pK identifikation and Homotypic Doublets Proportion Estimate

```
## pK identifikation
sweep.res <- paramSweep_v3(gs, PCs = 1:20) # as estimated from PC elbowPlot
sweep.stats_gs <- summarizeSweep(sweep.res, GT= FALSE)
Sys.sleep(0.5)
bcmvn_gs <- find.pK(sweep.stats_gs)
```

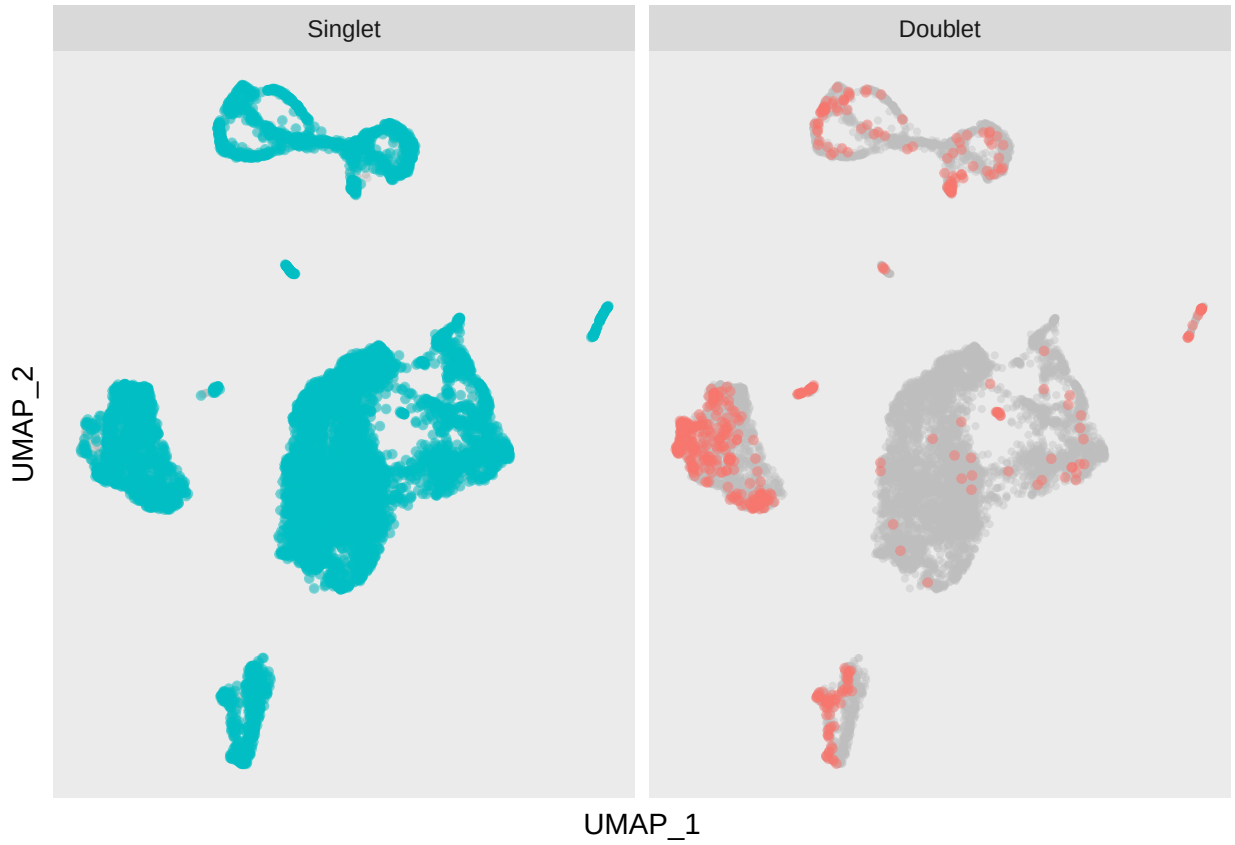


```
ggplot(bcmvn_gs, aes(pK, BCmetric, group = 1)) +
  geom_point() +
  geom_line()
```




```
pK <- bcmvn_gs %>% # select the pK that corresponds to max bcmvn to optimize doublet detection
  filter(BCmetric == max(BCmetric)) %>%
  select(pK)

pK <- as.numeric(as.character(pK[[1]]))

## Homotypic Doublets Proportion Estimate
annotations <- gs@meta.data$celltype_DF
homotypic.prop <- modelHomotypic(annotations)
nDoublets <- round(ncol(gs)*dblt.rate/100) # poi
nDoublets_nonhomo <- round(nDoublets*(1-homotypic.prop)) #poi.adj
## run doubletFinder: creating artificial doublets
gs <- doubletFinder_v3(gs,
                       PCs = 1:20,
                       pN = 0.25,
                       pK = pK,
                       nExp = nDoublets_nonhomo)

## Plot the singlets, doublets and the count of UMIs in each cells
col_dblts <- grep("DF.classifications", colnames(gs@meta.data), value=TRUE)

Idents(gs) <- col_dblts
cellsData <- data.frame(gs@reductions[["umap"]]@cell.embeddings, gs@meta.data[,col_dblts])
colnames(cellsData) <- c(colnames(gs@reductions[["umap"]]@cell.embeddings), "col_dblts")
p1 <- ggplot(cellsData[c('UMAP_1', 'UMAP_2')], # Omit the column used for facetting to get all points r
             aes( x = UMAP_1,
```

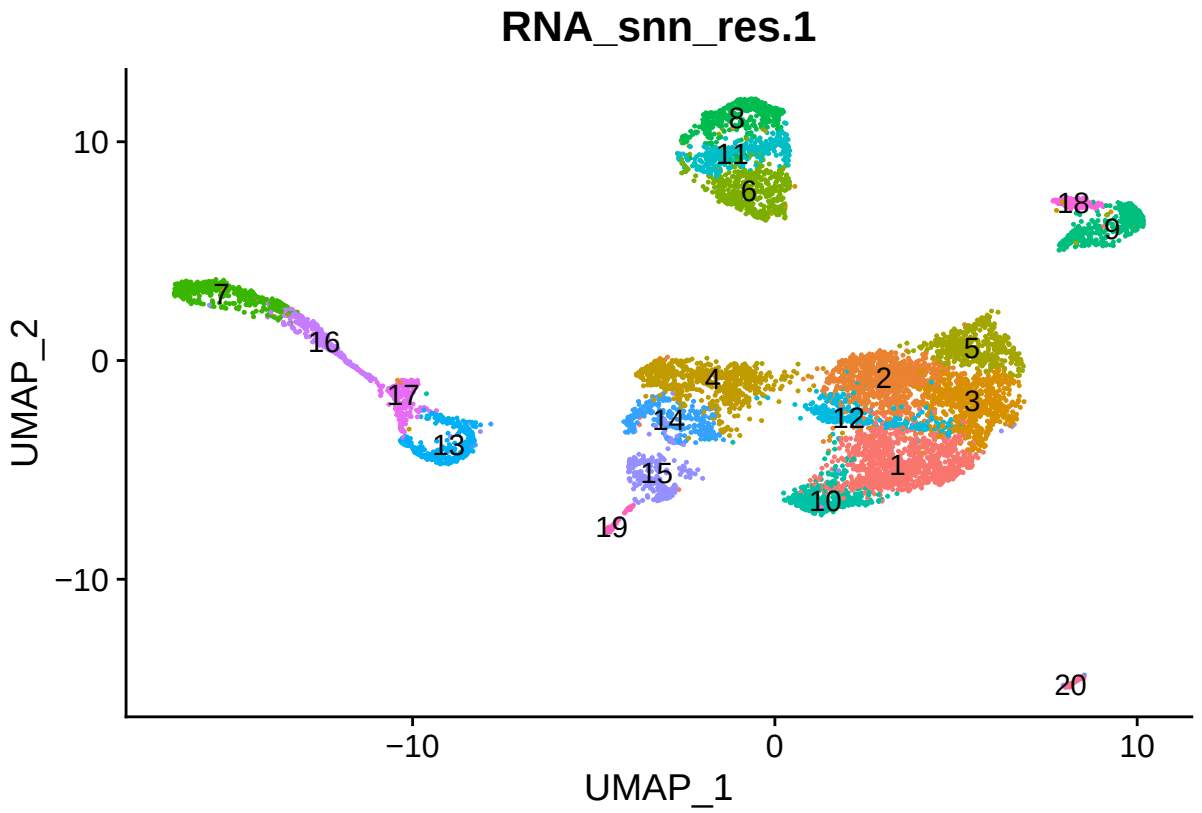
```
                  y = UMAP_2)) +
  geom_point( alpha = .4,
              size  = .75,
              color = "grey") +
  geom_point( data = cellsData, # Now provide data including column for facetting
              aes(color = col_dblts),
              alpha = .5,
              size  = 1.2) +
  facet_wrap(facets = vars(factor(col_dblts, levels = c("Singlet", "Doublet"))),
             ncol = 2) +
  NoLegend() +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text = element_blank(),
        line = element_blank())


p1
```



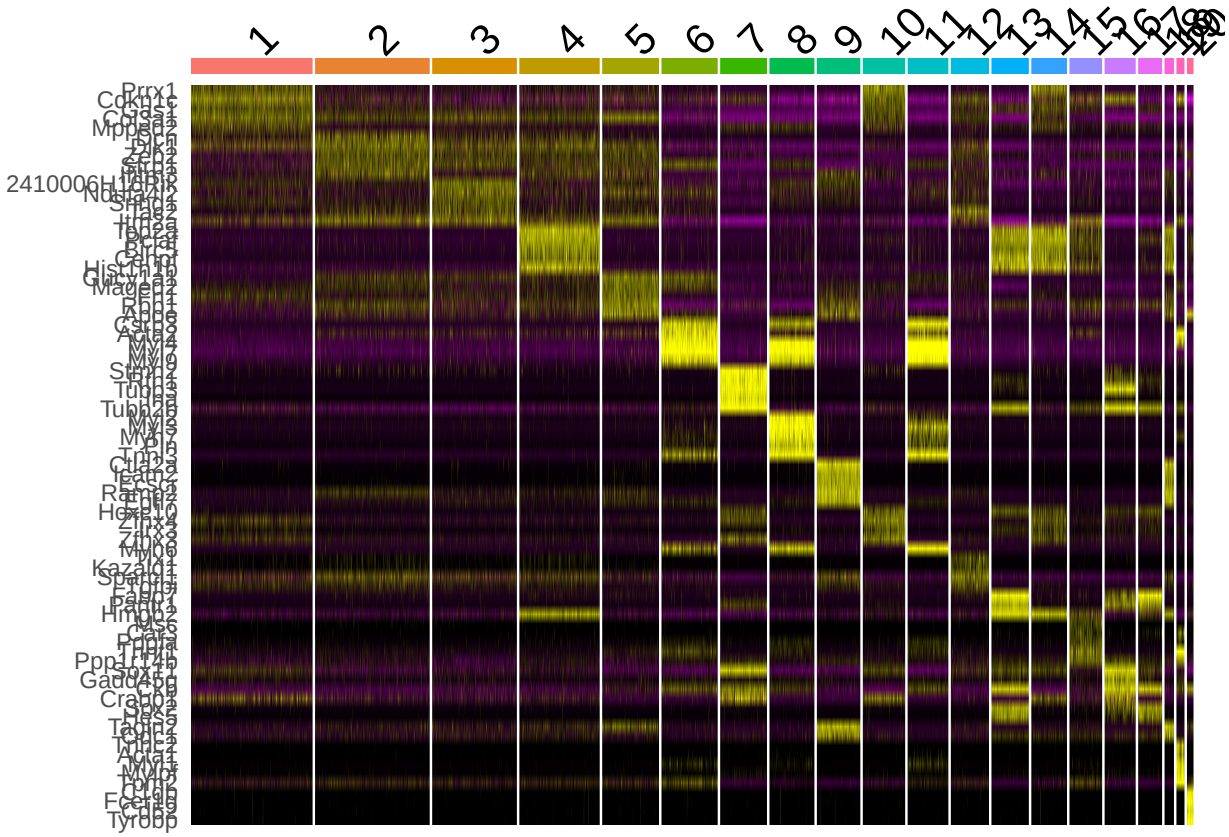

```
cellsData <- data.frame(gs@reductions[["umap"]]@cell.embeddings, gs@meta.data$nCount_RNA)
colnames(cellsData) <- c(colnames(gs@reductions[["umap"]]@cell.embeddings), "nCount_RNA")
Db <- table(gs@meta.data[col_dblts])["Doublet"]
saveRDS(gs, file = paste0(rdsObject, "gs.rds"))
```

8. Preprocessing workflow (after doublets rgsoval)

```
gs.sing <- subset(gs, idents='Singlet')
saveRDS(gs.sing, file = paste0(rdsObject, "gs_singlets.rds"))
gs.sing <- FindVariableFeatures(gs.sing, selection.method = var_feat_method, nfeatures=2000, verbose=FA
gs.sing <- ScaleData(gs.sing, features=rownames(gs.sing), do.scale=FALSE, verbose=FALSE)
gs.sing <- RunPCA(gs.sing, npcs = 20, nfeatures.print = 10, seed.use = general.seed, verbose=TRUE)
# Clustering
gs.sing <- FindNeighbors(gs.sing, dims = 1:20, verbose=FALSE)
gs.sing <- FindClusters(gs.sing, resolution = res_var, algorithm = Leiden, random.seed = general.seed,
## Run non-linear dimensional reduction (UMAP/tSNE)
gs.sing <- RunUMAP(gs.sing, dims = 1:20, seed.use = general.seed, verbose = FALSE)
DimPlot(gs.sing, group.by = "RNA_snn_res.1" , reduction = "umap", label = TRUE) + NoLegend()
```



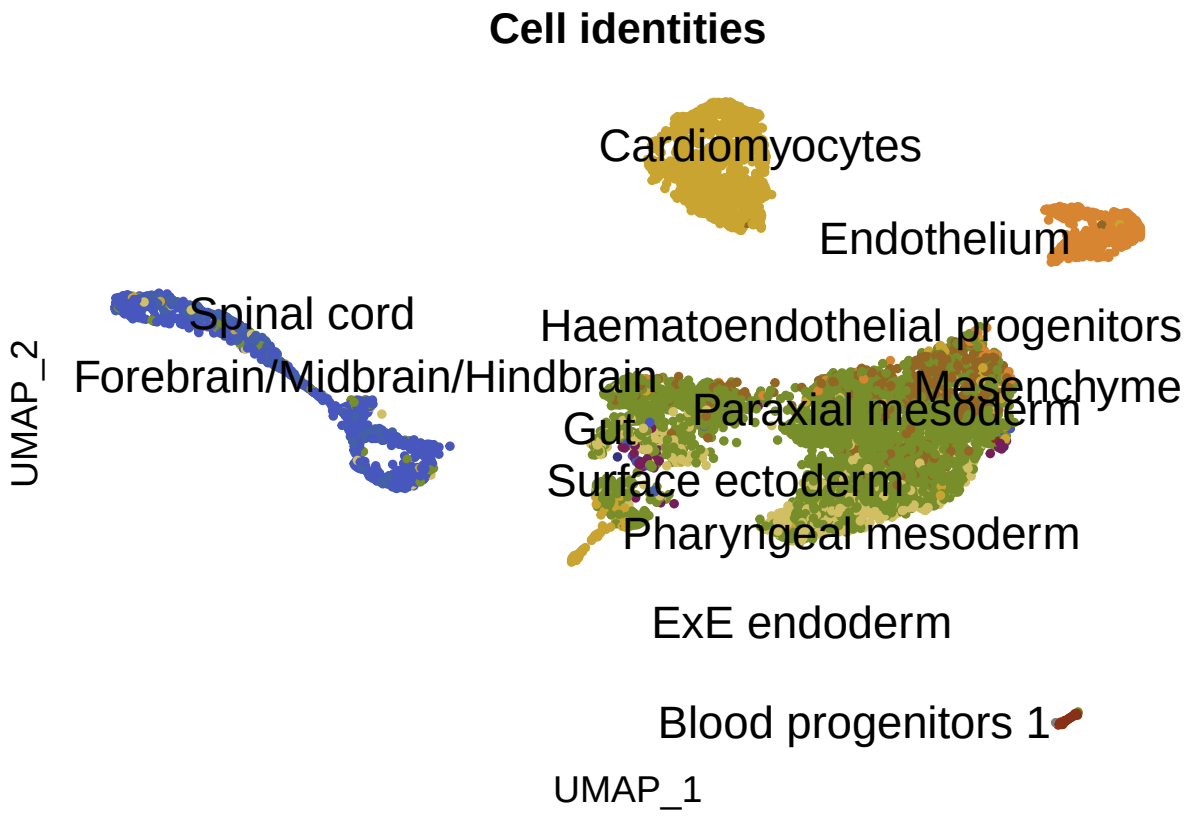

```
# Finding differentially expressed features (cluster biomarkers)------------
gs.sing.markers <- FindAllMarkers(gs.sing, group.by = "RNA_snn_res.1" , only.pos = TRUE, min.pct = 0.25
top10 <- gs.sing.markers %>%
  group_by(cluster) %>%
  top_n(n = 5, wt = avg_log2FC)
DoHeatmap(gs.sing, features = top10$gene, group.by = "RNA_snn_res.1") + NoLegend()
```




```
write.table(gs.sing.markers, file = paste0(table.path, "marquers_gastruloides_total.csv"))
```

9. Cell identities after perform DoubletFinder

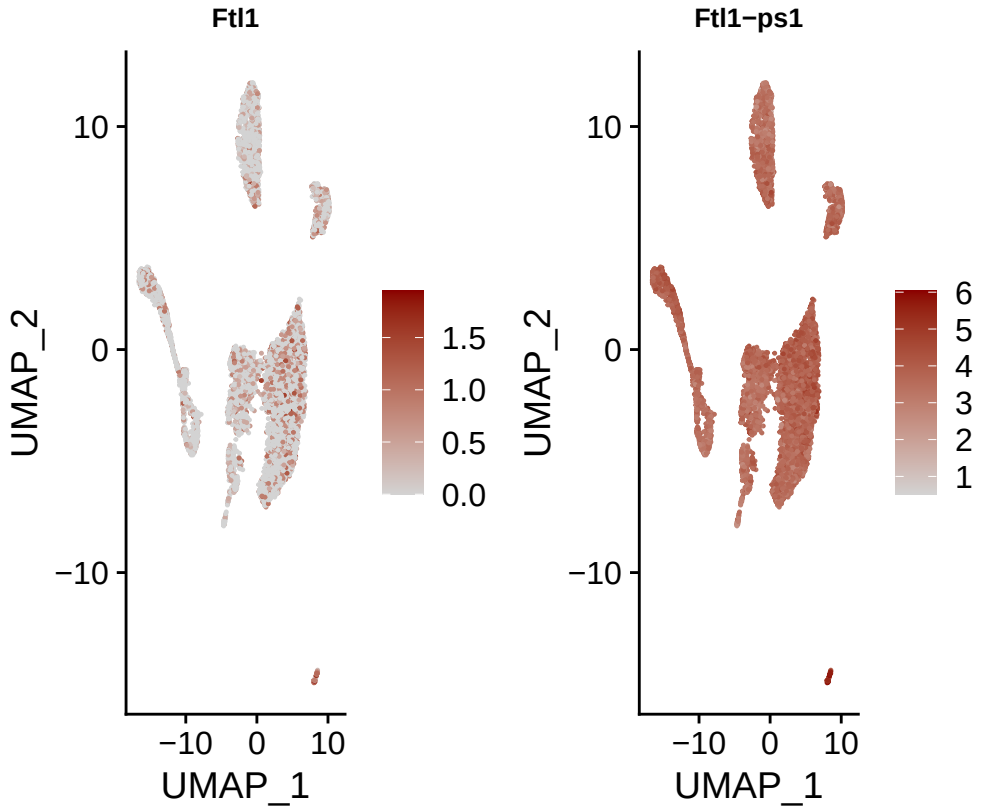
```
anchors <- FindTransferAnchors(reference = atlas.subset, query = gs.sing,
                               dims = 1:20, reference.reduction = "pca")
predictions <- TransferData(anchorset = anchors, refdata = atlas.subset$celltype,
                            dims = 1:20)
gs.sing <- AddMetaData(gs.sing, metadata = predictions[,"predicted.id"], col.name = "celltype_sing")
Idents(gs.sing) <- gs.sing@meta.data$celltype_sing

DimPlot(gs.sing,
        pt.size = 1,
        repel = TRUE,
        label = TRUE,
        label.size = 6,
        cols = colors.celltype[levels(Idents(gs.sing))]) +
  ggtitle( "Cell identities") +
  theme(plot.title = element_text(hjust = 0.5),
        axis.text = element_blank(),
        line = element_blank()) +
  NoLegend()
```




```
saveRDS(gs.sing, file = paste0(rdsObject,"gs_sing_process.rds"))
```

Expression of Flt1 and Flt1-ps

```
FeaturePlot(gs.sing, features = c("Ftl1","Ftl1-ps1"),  reduction = "umap", cols = c("lightgrey", "darkr
```




```
saveRDS(gs.sing, file = paste0(rdsObject,"gs_sing_total.rds"))
```