

CAF Single cell Transcriptomics Analysis

Mickael Coquerelle & Loik Galtier

2025-12-02

Table des matières

| | |
|--|----------|
| 1 Résumé | 2 |
| 2 Préparation des données. | 2 |
| 2.1 QC et filtrage des données | 3 |
| 3 Normalisation avec SC Transform. | 4 |
| 3.1 Le modèle mathématique derrière la normalisation SCTransform | 4 |
| 3.2 Application de la méthode | 5 |
| 4 Réduction de dimension, clustering et visualisation | 5 |
| 4.1 Interprétation PCA\ | 5 |
| 4.2 Interprétation TSNE} \ | 7 |

```
# Setup global chunk options and libraries.
knitr::opts_chunk$set(echo = TRUE, warning = FALSE, message = FALSE,
                      fig.width = 8, fig.height = 6)

# Tidyverse-like utilities
library(dplyr)
library(ggplot2)
# Seurat and single-cell specific tools
library(Seurat)
# Parallelization for Seurat (kept sequential by default in this report for reproducibility)
library(future)
# Annotation via Ensembl
library(AnnotationDbi)
library(EnsDb.Hsapiens.v86)
sessionInfo()
# Parallelisation:
plan(sequential)
options(future.globals.maxSize=10*1024^3) # 10 GB

# Resolve namespace conflicts (ensembldb::filter masks dplyr::filter)
filter <- dplyr::filter
arrange <- dplyr::arrange
mutate <- dplyr::mutate
select <- dplyr::select

" data loading : we load the three datasets corresponding to three different patients' CAF single-cell
```

```

p5 <- read.csv("~/Projets_GIT/Single_Cell_Project/Data/GSM4805570_CountsMatrix_20G00953M_TN.txt.gz",
sep="\t")
p4a <- read.csv("~/Projets_GIT/Single_Cell_Project/Data/GSM4805568_CountsMatrix_19G02977B_TN.txt.gz",
sep="\t")
p4b <- read.csv("~/Projets_GIT/Single_Cell_Project/Data/GSM4805566_CountsMatrix_19G02977A_TN.txt.gz",
sep="\t")

```

1 Résumé

L'objet de ce travail et de se familiariser avec l'analyse transcriptomique Single-Cell de CAF (Cancer Associated Fibroblasts) à partir d'une matrice de comptages issues des trois fichiers proposés. Notre pipeline d'analyse en se basant sur les éléments donnés en cours couvre le nettoyage (filtrage cellules/gènes), normalisation, réduction de dimension (PCA, t-SNE, UMPAP), clustering, identification de(s) signature(s) transcriptomique(s) pour marquer les clusters et enfin nous proposons une visualisation ciblée par heatmap et features plots.

2 Préparation des données.

We combine the three datasets and map Ensembl IDs to gene symbols and types. the objectif is to filter for protein-coding genes with unique symbols and no missing values. ## Pré-processing des données

```

caf.data <- data.matrix(cbind(p5, p4a, p4b))

# Ensembl mapping

ens <- read.csv("~/Projets_GIT/Single_Cell_Project/Data/ensembl-38-108-genes.txt", sep="\t")
ens2symb <- setNames(ens$Gene.name, ens$Gene.stable.ID)
ens2type <- setNames(ens$Gene.type, ens$Gene.stable.ID)

symbols <- ens2symb[rownames(caf.data)]
types <- ens2type[rownames(caf.data)]

good <- types == "protein_coding" & !is.na(symbols) & !duplicated(symbols)
symb <- symbols[good]

caf.data <- caf.data[good, ]
rownames(caf.data) <- symb

```

On crée un premier objet SeuratObject pour tirer profit des routines intégrées à la librairie (FindVariableFeatures, ScaleData, RunPCA, etc.) :

```

caf <- CreateSeuratObject(
counts = caf.data,
project = "CAF_raw",
min.cells = 0.01 * ncol(caf.data),
min.features = 1000
)
#caf

```

2.1 QC et filtrage des données

Puis on va définir des filtres. La détermination de ces derniers se fait sur des règles d'usage en *single-cell. min.cells = 0.01* au premier passage pour exclure les gènes exprimés dans très peu de cellules (ici adapté au CreateSeuratObject). On applique ensuite des QC aux cellules : - Les UMIs élevés (>50000) indiquent des duplicats il faut donc les gérer. - Les cellules avec peu de gènes détectés (< 1000) sont souvent de mauvaise qualité, il y'a donc nécessité de les filtrer. - Ensuite on ajoute une condition pour gérer la proportion de gènes mitochondriaux élevée (> 50), qui est un indicateur de cellules mortes. En combinant ces filtres, on nettoie notre dataset de manière avoir un jeu de cellule de haute qualité.

```
# Identify mitochondrial genes

ensdb.genes <- genes(EnsDb.Hsapiens.v86)
MT.names <- ensdb.genes[seqnames(ensdb.genes) == "MT"]$gene_name
counts <- GetAssayData(caf, "RNA")
umi.tot <- colSums(counts)
gene.tot <- colSums(counts > 0)
mito.pc <- colSums(counts[rownames(counts) %in% MT.names, ]) / umi.tot * 100

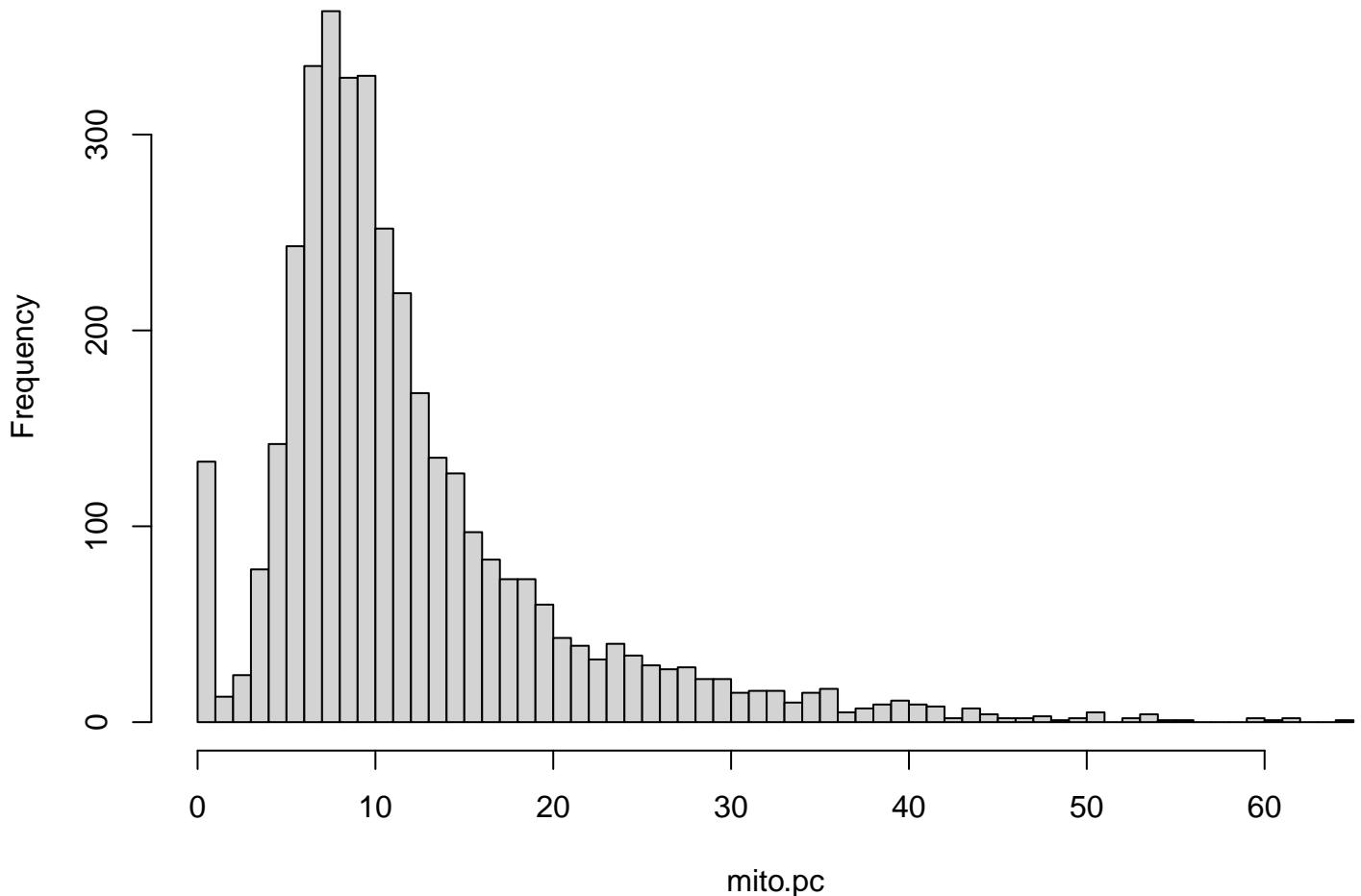
# Quality filters
bad.high <- umi.tot > 50000
bad.low <- gene.tot < 1000
bad.mito <- mito.pc > 50
bad <- bad.high | bad.low | bad.mito

# Cleaning
counts <- counts[, !bad]
good.genes <- rowSums(counts > 1) >= 0.01 * ncol(counts)
counts <- counts[good.genes, ]
dim(counts)
```

Nous pouvons visualiser la distribution de la proportion de gènes mitochondriaux, avant nettoyage :

```
hist(mito.pc, breaks=50)
```

Histogram of mito.pc



Nous mettons à jour notre objet Seurat avec les données nettoyées :

```
caf <- CreateSeuratObject(counts = counts, project = "CAF_clean")
caf[["percent.mt"]] <- PercentageFeatureSet(caf, pattern = "^MT-")
```

3 Normalisation avec SC Transform.

La normalisation étant une étape charnière et pouvant grandement changer les résultats finaux en Single-Cell. Nous avons choisi ici d'exploiter les fonctionnalités de normalisation de la librairie Seurat. Plutôt qu'une normalisation CPM (par millions) ou LogNormalize, nous appliquons ici **SCTransform**, une méthode plus moderne et robuste, qui est construite sur un modèle négatif binomial (Hafemeister & Satija, 2019). De ce que nous en avons compris, une telle approche va agir sur plusieurs aspects : - La normalisation par profondeur de séquençage,
- La stabilisation de variance,
- La sélection des gènes variables,
- La correction des effets techniques (ici regression des % mitochondriaux). Cette méthode est d'après la littérature **la référence Seurat** pour le pré-traitement.

3.1 Le modèle mathématique derrière la normalisation SCTransform

La normalisation SCTransform repose sur un modèle de **régression négative binomiale**.

Pour chaque gène g et chaque cellule i , on suppose : $y_{ig} \sim NB(\mu_{ig}, \theta_g)$ où y_{ig} est le nombre de transcripts observés, μ_{ig} la moyenne attendue de nos comptes et θ_g : le paramètre de dispersion du gène g . La moyenne μ_{ig} est modélisée via

un lien log-linéaire.

$$\log(\mu_{ig}) = \beta_{0g} + \beta_{1g} \cdot \log(\text{UMI}_i) + \sum_k \gamma_{kg} x_{ik}$$

avec : - β_{0g} : intercept spécifique au gène g

- UMI_i : total de counts de la cellule i (size factor)

- x_{ik} : variables techniques à régresser (ex. % mitochondriaux, batch)

- γ_{kg} : coefficients associés aux covariables La normalisation finale utilisée pour l'analyse (PCA, clustering) est obtenue via les **résidus Pearson standardisés** :

$$r_{ig} = \frac{y_{ig} - \hat{\mu}_{ig}}{\sqrt{\hat{\mu}_{ig} + \frac{\hat{\mu}_{ig}^2}{\theta_g}}}$$

Ces résidus sont **variance-stabilisés** et permettent d'atténuer l'effet de la profondeur de séquençage et du bruit technique. Avec cette approche, on à *à priori* un modèle individuel pour chaque gène, et une normalisation « cellule spécifique ».

3.2 Application de la méthode

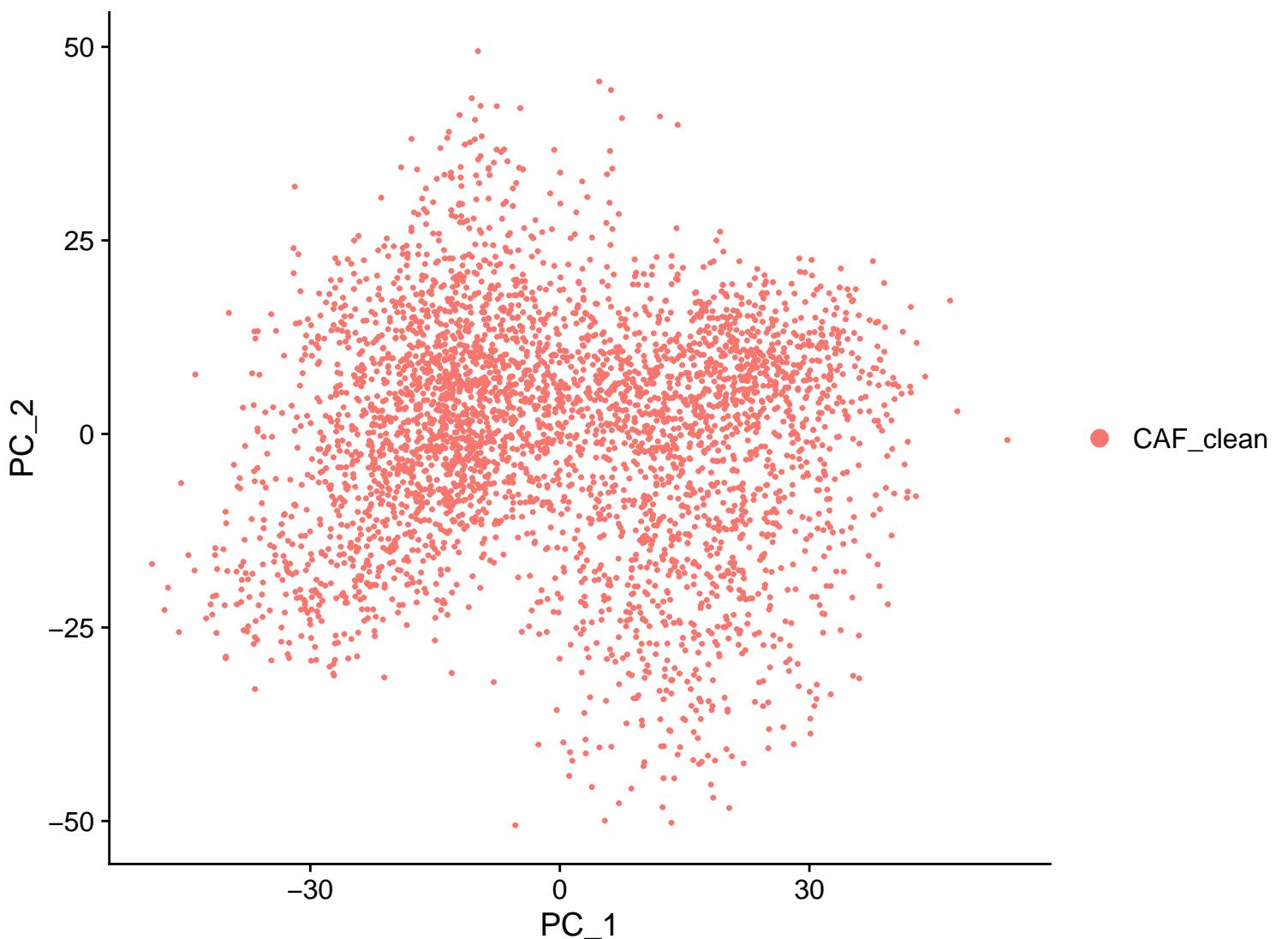
```
caf <- SCTransform(  
  caf,  
  vars.to.regress = "percent.mt",  
  verbose = TRUE)
```

4 Réduction de dimension, clustering et visualisation

Nous procédons à la réduction de dimension via PCA, puis UMAP pour la visualisation. Nous effectuons ensuite le clustering des cellules et identifions les marqueurs de chaque cluster.\

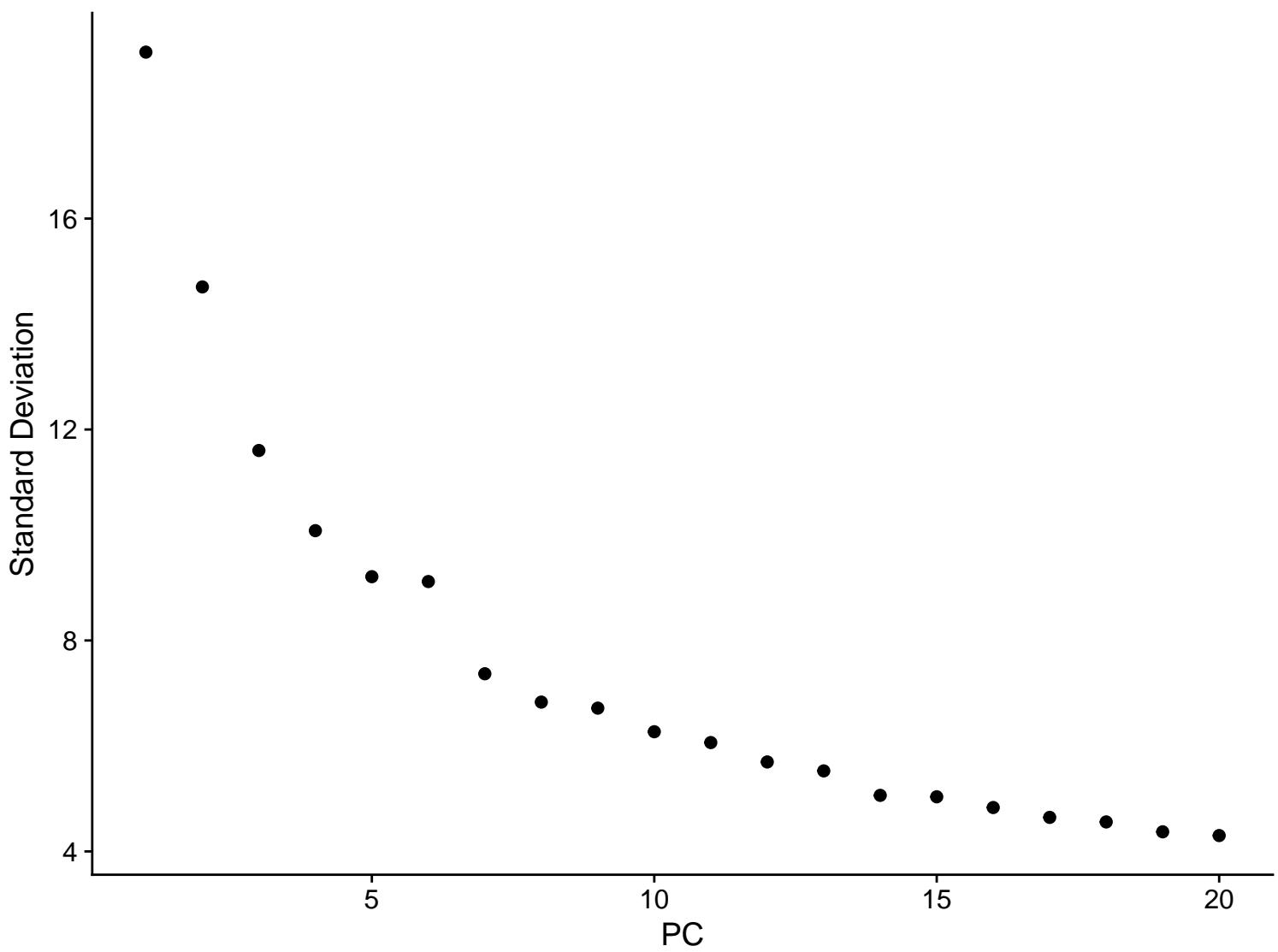
4.1 Interprétation PCA\

```
caf <- RunPCA(caf, verbose = FALSE)  
DimPlot(caf, reduction = "pca")
```



La PCA des cellules CAF nettoyées nous montre ici un nuage diffus sans clusters franchement identifiable. C'est le reflet d'une grande variabilité plutôt que de sous-populations clairement séparées. On peut expliquer cela en partie par la diversité des CAF, mais aussi par le fait que la PCA capture surtout la variance linéaire qui est dominante. Avant SCTransform, des effets techniques comme la profondeur de séquençage ou le % mitochondrial peuvent encore masquer la structure biologique réelle. Les signatures transcriptionnelles typiques des CAF sont ici illustrés par cette dispersion dans l'espace PCA. Cette première exploration nous confirme la qualité que des méthodes non linéaires (UMAP, t-SNE) sont nécessaires pour que nous allions plus loin dans les investigations.

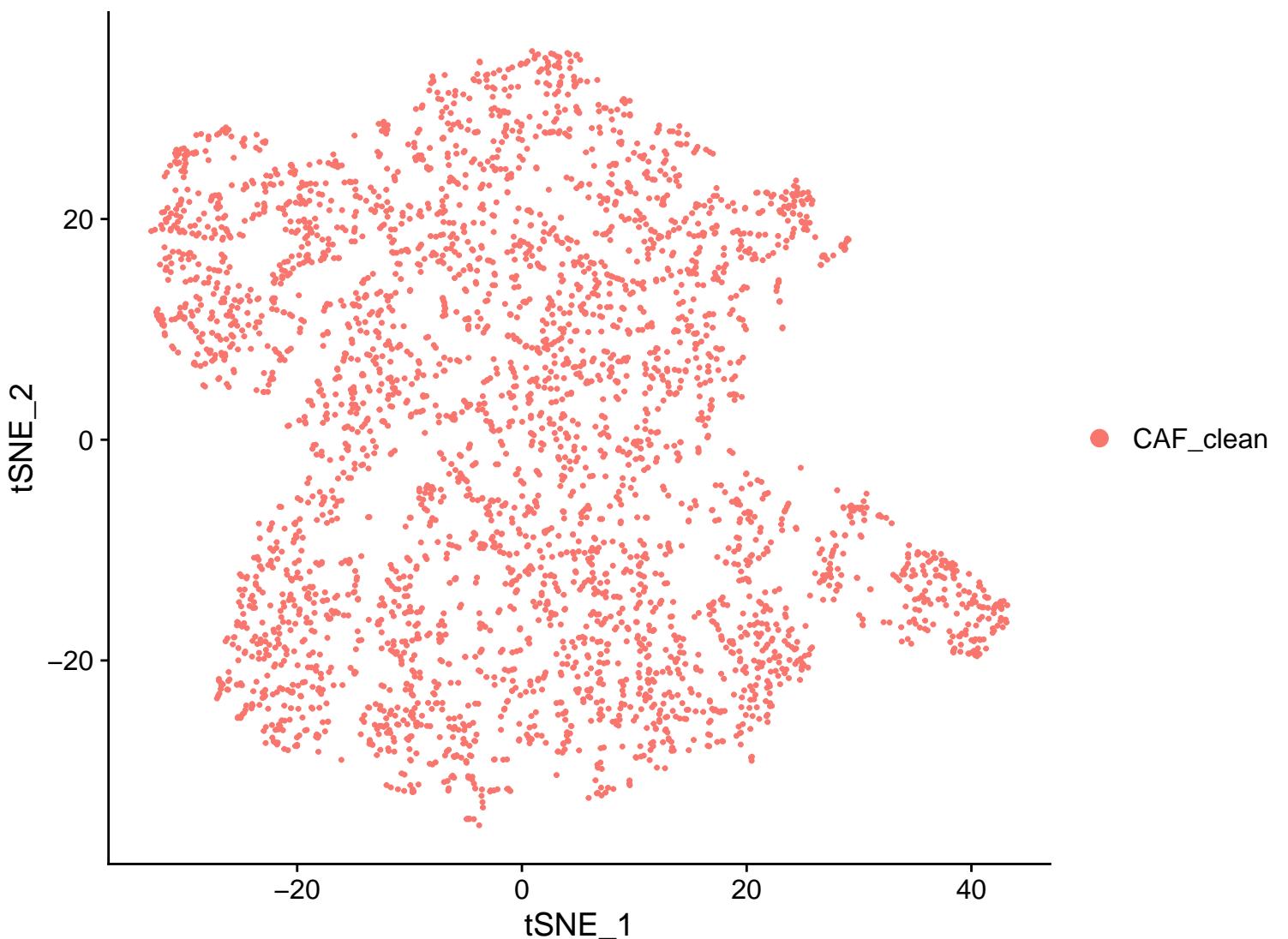
```
ElbowPlot(caf)
```



Le graphique ElbowPlot nous montre que les 10 premières composantes principales capturent la majorité de la variance significative dans les données (ce qui paraît cohérent avec les notions expliquées en cours). Au-delà de ce seuil, l'ajout de composantes supplémentaires contribue peu à l'explication de la variance totale, on a donc un point d'inflexion clair. Par conséquent, nous choisissons d'utiliser les 10 premières dimensions pour les analyses ultérieures telles que UMAP et t-SNE. Cela nous permet donc de capturer la structure transcriptionnelle des CAF sans intégrer trop de bruit.

4.2 Interprétation TSNE} \

```
caf_tsne <- RunTSNE(caf, dims = 1:10, perplexity = 30)
DimPlot(caf_tsne, reduction = "tsne")
```



Interprétation UMAP et clustering :

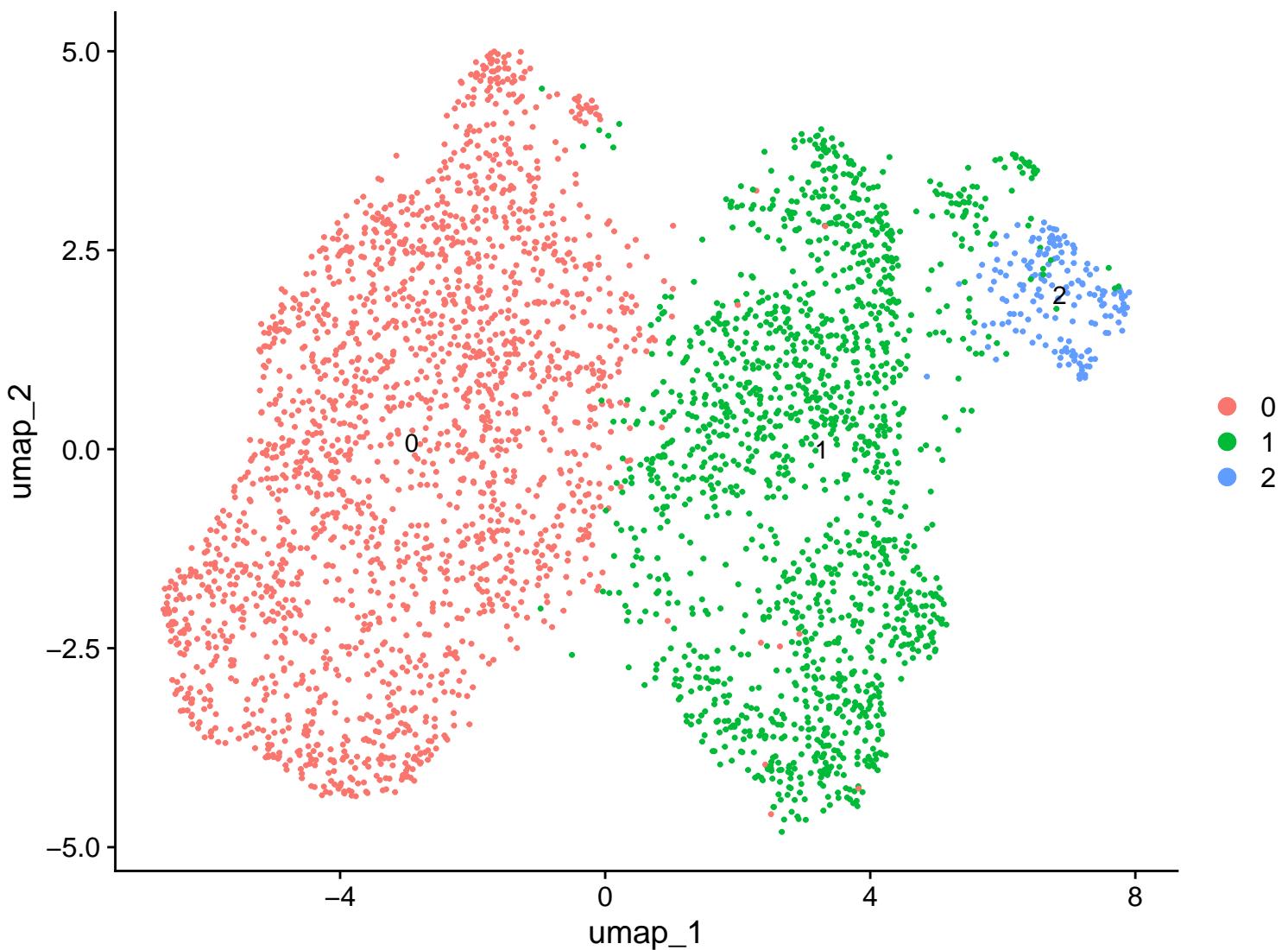
```

caf_umap <- RunUMAP(caf, dims = 1:10)
caf_umap <- FindNeighbors(caf_umap, dims = 1:10)
caf_umap <- FindClusters(caf_umap, resolution = 0.1)

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 3728
## Number of edges: 117803
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9287
## Number of communities: 3
## Elapsed time: 0 seconds

DimPlot(caf_umap, reduction = "umap", label = TRUE)

```



Interprétation Clustering :

Chaque ligne correspond à une famille d'origine différente. La première représente les VSMC, la seconde ligne correspond à HSC, la troisième ligne représente les SAMes

```

signatureGene <- c(
  "PLN", "SORBS2", "PHLDA2", "SNCG", "MT1M", "MYH11",
  "PTGDS", "FBLN1", "DCN", "LUM", "COL1A1", "LTBP2",
  "FABP5", "HIGD1B", "AGT", "RGS5", "CPE", "SSTR2")

caf_markers <- FindAllMarkers(
  caf_umap,
  only.pos = TRUE,
  min.pct = 0.25,
  logfc.threshold = 0.25)

caf_markers %>% group_by(cluster) %>% slice_max(n = 2, order_by = avg_log2FC)

## # A tibble: 6 x 7
## # Groups:   cluster [3]
##       p_val avg_log2FC pct.1 pct.2 p_val_adj cluster gene
##       <dbl>      <dbl> <dbl> <dbl>     <dbl> <fct>  <chr>
## 1      0.001  0.250000 0.001  0.001  0.000100  0     PLN
## 2      0.001  0.250000 0.001  0.001  0.000100  0     SORBS2
## 3      0.001  0.250000 0.001  0.001  0.000100  0     PHLDA2
## 4      0.001  0.250000 0.001  0.001  0.000100  0     SNCG
## 5      0.001  0.250000 0.001  0.001  0.000100  0     MT1M
## 6      0.001  0.250000 0.001  0.001  0.000100  0     MYH11
## 7      0.001  0.250000 0.001  0.001  0.000100  1     PTGDS
## 8      0.001  0.250000 0.001  0.001  0.000100  1     FBLN1
## 9      0.001  0.250000 0.001  0.001  0.000100  1     DCN
## 10     0.001  0.250000 0.001  0.001  0.000100  1     LUM
## 11     0.001  0.250000 0.001  0.001  0.000100  1     COL1A1
## 12     0.001  0.250000 0.001  0.001  0.000100  1     LTBP2
## 13     0.001  0.250000 0.001  0.001  0.000100  1     FABP5
## 14     0.001  0.250000 0.001  0.001  0.000100  1     HIGD1B
## 15     0.001  0.250000 0.001  0.001  0.000100  1     AGT
## 16     0.001  0.250000 0.001  0.001  0.000100  1     RGS5
## 17     0.001  0.250000 0.001  0.001  0.000100  1     CPE
## 18     0.001  0.250000 0.001  0.001  0.000100  1     SSTR2
## 19     0.001  0.250000 0.001  0.001  0.000100  2     PLN
## 20     0.001  0.250000 0.001  0.001  0.000100  2     SORBS2
## 21     0.001  0.250000 0.001  0.001  0.000100  2     PHLDA2
## 22     0.001  0.250000 0.001  0.001  0.000100  2     SNCG
## 23     0.001  0.250000 0.001  0.001  0.000100  2     MT1M
## 24     0.001  0.250000 0.001  0.001  0.000100  2     MYH11
## 25     0.001  0.250000 0.001  0.001  0.000100  2     PTGDS
## 26     0.001  0.250000 0.001  0.001  0.000100  2     FBLN1
## 27     0.001  0.250000 0.001  0.001  0.000100  2     DCN
## 28     0.001  0.250000 0.001  0.001  0.000100  2     LUM
## 29     0.001  0.250000 0.001  0.001  0.000100  2     COL1A1
## 30     0.001  0.250000 0.001  0.001  0.000100  2     LTBP2
## 31     0.001  0.250000 0.001  0.001  0.000100  2     FABP5
## 32     0.001  0.250000 0.001  0.001  0.000100  2     HIGD1B
## 33     0.001  0.250000 0.001  0.001  0.000100  2     AGT
## 34     0.001  0.250000 0.001  0.001  0.000100  2     RGS5
## 35     0.001  0.250000 0.001  0.001  0.000100  2     CPE
## 36     0.001  0.250000 0.001  0.001  0.000100  2     SSTR2
## 37     0.001  0.250000 0.001  0.001  0.000100  3     PLN
## 38     0.001  0.250000 0.001  0.001  0.000100  3     SORBS2
## 39     0.001  0.250000 0.001  0.001  0.000100  3     PHLDA2
## 40     0.001  0.250000 0.001  0.001  0.000100  3     SNCG
## 41     0.001  0.250000 0.001  0.001  0.000100  3     MT1M
## 42     0.001  0.250000 0.001  0.001  0.000100  3     MYH11
## 43     0.001  0.250000 0.001  0.001  0.000100  3     PTGDS
## 44     0.001  0.250000 0.001  0.001  0.000100  3     FBLN1
## 45     0.001  0.250000 0.001  0.001  0.000100  3     DCN
## 46     0.001  0.250000 0.001  0.001  0.000100  3     LUM
## 47     0.001  0.250000 0.001  0.001  0.000100  3     COL1A1
## 48     0.001  0.250000 0.001  0.001  0.000100  3     LTBP2
## 49     0.001  0.250000 0.001  0.001  0.000100  3     FABP5
## 50     0.001  0.250000 0.001  0.001  0.000100  3     HIGD1B
## 51     0.001  0.250000 0.001  0.001  0.000100  3     AGT
## 52     0.001  0.250000 0.001  0.001  0.000100  3     RGS5
## 53     0.001  0.250000 0.001  0.001  0.000100  3     CPE
## 54     0.001  0.250000 0.001  0.001  0.000100  3     SSTR2
## 55     0.001  0.250000 0.001  0.001  0.000100  4     PLN
## 56     0.001  0.250000 0.001  0.001  0.000100  4     SORBS2
## 57     0.001  0.250000 0.001  0.001  0.000100  4     PHLDA2
## 58     0.001  0.250000 0.001  0.001  0.000100  4     SNCG
## 59     0.001  0.250000 0.001  0.001  0.000100  4     MT1M
## 60     0.001  0.250000 0.001  0.001  0.000100  4     MYH11
## 61     0.001  0.250000 0.001  0.001  0.000100  4     PTGDS
## 62     0.001  0.250000 0.001  0.001  0.000100  4     FBLN1
## 63     0.001  0.250000 0.001  0.001  0.000100  4     DCN
## 64     0.001  0.250000 0.001  0.001  0.000100  4     LUM
## 65     0.001  0.250000 0.001  0.001  0.000100  4     COL1A1
## 66     0.001  0.250000 0.001  0.001  0.000100  4     LTBP2
## 67     0.001  0.250000 0.001  0.001  0.000100  4     FABP5
## 68     0.001  0.250000 0.001  0.001  0.000100  4     HIGD1B
## 69     0.001  0.250000 0.001  0.001  0.000100  4     AGT
## 70     0.001  0.250000 0.001  0.001  0.000100  4     RGS5
## 71     0.001  0.250000 0.001  0.001  0.000100  4     CPE
## 72     0.001  0.250000 0.001  0.001  0.000100  4     SSTR2
## 73     0.001  0.250000 0.001  0.001  0.000100  5     PLN
## 74     0.001  0.250000 0.001  0.001  0.000100  5     SORBS2
## 75     0.001  0.250000 0.001  0.001  0.000100  5     PHLDA2
## 76     0.001  0.250000 0.001  0.001  0.000100  5     SNCG
## 77     0.001  0.250000 0.001  0.001  0.000100  5     MT1M
## 78     0.001  0.250000 0.001  0.001  0.000100  5     MYH11
## 79     0.001  0.250000 0.001  0.001  0.000100  5     PTGDS
## 80     0.001  0.250000 0.001  0.001  0.000100  5     FBLN1
## 81     0.001  0.250000 0.001  0.001  0.000100  5     DCN
## 82     0.001  0.250000 0.001  0.001  0.000100  5     LUM
## 83     0.001  0.250000 0.001  0.001  0.000100  5     COL1A1
## 84     0.001  0.250000 0.001  0.001  0.000100  5     LTBP2
## 85     0.001  0.250000 0.001  0.001  0.000100  5     FABP5
## 86     0.001  0.250000 0.001  0.001  0.000100  5     HIGD1B
## 87     0.001  0.250000 0.001  0.001  0.000100  5     AGT
## 88     0.001  0.250000 0.001  0.001  0.000100  5     RGS5
## 89     0.001  0.250000 0.001  0.001  0.000100  5     CPE
## 90     0.001  0.250000 0.001  0.001  0.000100  5     SSTR2
## 91     0.001  0.250000 0.001  0.001  0.000100  6     PLN
## 92     0.001  0.250000 0.001  0.001  0.000100  6     SORBS2
## 93     0.001  0.250000 0.001  0.001  0.000100  6     PHLDA2
## 94     0.001  0.250000 0.001  0.001  0.000100  6     SNCG
## 95     0.001  0.250000 0.001  0.001  0.000100  6     MT1M
## 96     0.001  0.250000 0.001  0.001  0.000100  6     MYH11
## 97     0.001  0.250000 0.001  0.001  0.000100  6     PTGDS
## 98     0.001  0.250000 0.001  0.001  0.000100  6     FBLN1
## 99     0.001  0.250000 0.001  0.001  0.000100  6     DCN
## 100    0.001  0.250000 0.001  0.001  0.000100  6     LUM
## 101    0.001  0.250000 0.001  0.001  0.000100  6     COL1A1
## 102    0.001  0.250000 0.001  0.001  0.000100  6     LTBP2
## 103    0.001  0.250000 0.001  0.001  0.000100  6     FABP5
## 104    0.001  0.250000 0.001  0.001  0.000100  6     HIGD1B
## 105    0.001  0.250000 0.001  0.001  0.000100  6     AGT
## 106    0.001  0.250000 0.001  0.001  0.000100  6     RGS5
## 107    0.001  0.250000 0.001  0.001  0.000100  6     CPE
## 108    0.001  0.250000 0.001  0.001  0.000100  6     SSTR2
## 109    0.001  0.250000 0.001  0.001  0.000100  7     PLN
## 110    0.001  0.250000 0.001  0.001  0.000100  7     SORBS2
## 111    0.001  0.250000 0.001  0.001  0.000100  7     PHLDA2
## 112    0.001  0.250000 0.001  0.001  0.000100  7     SNCG
## 113    0.001  0.250000 0.001  0.001  0.000100  7     MT1M
## 114    0.001  0.250000 0.001  0.001  0.000100  7     MYH11
## 115    0.001  0.250000 0.001  0.001  0.000100  7     PTGDS
## 116    0.001  0.250000 0.001  0.001  0.000100  7     FBLN1
## 117    0.001  0.250000 0.001  0.001  0.000100  7     DCN
## 118    0.001  0.250000 0.001  0.001  0.000100  7     LUM
## 119    0.001  0.250000 0.001  0.001  0.000100  7     COL1A1
## 120    0.001  0.250000 0.001  0.001  0.000100  7     LTBP2
## 121    0.001  0.250000 0.001  0.001  0.000100  7     FABP5
## 122    0.001  0.250000 0.001  0.001  0.000100  7     HIGD1B
## 123    0.001  0.250000 0.001  0.001  0.000100  7     AGT
## 124    0.001  0.250000 0.001  0.001  0.000100  7     RGS5
## 125    0.001  0.250000 0.001  0.001  0.000100  7     CPE
## 126    0.001  0.250000 0.001  0.001  0.000100  7     SSTR2
## 127    0.001  0.250000 0.001  0.001  0.000100  8     PLN
## 128    0.001  0.250000 0.001  0.001  0.000100  8     SORBS2
## 129    0.001  0.250000 0.001  0.001  0.000100  8     PHLDA2
## 130    0.001  0.250000 0.001  0.001  0.000100  8     SNCG
## 131    0.001  0.250000 0.001  0.001  0.000100  8     MT1M
## 132    0.001  0.250000 0.001  0.001  0.000100  8     MYH11
## 133    0.001  0.250000 0.001  0.001  0.000100  8     PTGDS
## 134    0.001  0.250000 0.001  0.001  0.000100  8     FBLN1
## 135    0.001  0.250000 0.001  0.001  0.000100  8     DCN
## 136    0.001  0.250000 0.001  0.001  0.000100  8     LUM
## 137    0.001  0.250000 0.001  0.001  0.000100  8     COL1A1
## 138    0.001  0.250000 0.001  0.001  0.000100  8     LTBP2
## 139    0.001  0.250000 0.001  0.001  0.000100  8     FABP5
## 140    0.001  0.250000 0.001  0.001  0.000100  8     HIGD1B
## 141    0.001  0.250000 0.001  0.001  0.000100  8     AGT
## 142    0.001  0.250000 0.001  0.001  0.000100  8     RGS5
## 143    0.001  0.250000 0.001  0.001  0.000100  8     CPE
## 144    0.001  0.250000 0.001  0.001  0.000100  8     SSTR2
## 145    0.001  0.250000 0.001  0.001  0.000100  9     PLN
## 146    0.001  0.250000 0.001  0.001  0.000100  9     SORBS2
## 147    0.001  0.250000 0.001  0.001  0.000100  9     PHLDA2
## 148    0.001  0.250000 0.001  0.001  0.000100  9     SNCG
## 149    0.001  0.250000 0.001  0.001  0.000100  9     MT1M
## 150    0.001  0.250000 0.001  0.001  0.000100  9     MYH11
## 151    0.001  0.250000 0.001  0.001  0.000100  9     PTGDS
## 152    0.001  0.250000 0.001  0.001  0.000100  9     FBLN1
## 153    0.001  0.250000 0.001  0.001  0.000100  9     DCN
## 154    0.001  0.250000 0.001  0.001  0.000100  9     LUM
## 155    0.001  0.250000 0.001  0.001  0.000100  9     COL1A1
## 156    0.001  0.250000 0.001  0.001  0.000100  9     LTBP2
## 157    0.001  0.250000 0.001  0.001  0.000100  9     FABP5
## 158    0.001  0.250000 0.001  0.001  0.000100  9     HIGD1B
## 159    0.001  0.250000 0.001  0.001  0.000100  9     AGT
## 160    0.001  0.250000 0.001  0.001  0.000100  9     RGS5
## 161    0.001  0.250000 0.001  0.001  0.000100  9     CPE
## 162    0.001  0.250000 0.001  0.001  0.000100  9     SSTR2
## 163    0.001  0.250000 0.001  0.001  0.000100  10    PLN
## 164    0.001  0.250000 0.001  0.001  0.000100  10    SORBS2
## 165    0.001  0.250000 0.001  0.001  0.000100  10    PHLDA2
## 166    0.001  0.250000 0.001  0.001  0.000100  10    SNCG
## 167    0.001  0.250000 0.001  0.001  0.000100  10    MT1M
## 168    0.001  0.250000 0.001  0.001  0.000100  10    MYH11
## 169    0.001  0.250000 0.001  0.001  0.000100  10    PTGDS
## 170    0.001  0.250000 0.001  0.001  0.000100  10    FBLN1
## 171    0.001  0.250000 0.001  0.001  0.000100  10    DCN
## 172    0.001  0.250000 0.001  0.001  0.000100  10    LUM
## 173    0.001  0.250000 0.001  0.001  0.000100  10    COL1A1
## 174    0.001  0.250000 0.001  0.001  0.000100  10    LTBP2
## 175    0.001  0.250000 0.001  0.001  0.000100  10    FABP5
## 176    0.001  0.250000 0.001  0.001  0.000100  10    HIGD1B
## 177    0.001  0.250000 0.001  0.001  0.000100  10    AGT
## 178    0.001  0.250000 0.001  0.001  0.000100  10    RGS5
## 179    0.001  0.250000 0.001  0.001  0.000100  10    CPE
## 180    0.001  0.250000 0.001  0.001  0.000100  10    SSTR2
## 181    0.001  0.250000 0.001  0.001  0.000100  11    PLN
## 182    0.001  0.250000 0.001  0.001  0.000100  11    SORBS2
## 183    0.001  0.250000 0.001  0.001  0.000100  11    PHLDA2
## 184    0.001  0.250000 0.001  0.001  0.000100  11    SNCG
## 185    0.001  0.250000 0.001  0.001  0.000100  11    MT1M
## 186    0.001  0.250000 0.001  0.001  0.000100  11    MYH11
## 187    0.001  0.250000 0.001  0.001  0.000100  11    PTGDS
## 188    0.001  0.250000 0.001  0.001  0.000100  11    FBLN1
## 189    0.001  0.250000 0.001  0.001  0.000100  11    DCN
## 190    0.001  0.250000 0.001  0.001  0.000100  11    LUM
## 191    0.001  0.250000 0.001  0.001  0.000100  11    COL1A1
## 192    0.001  0.250000 0.001  0.001  0.000100  11    LTBP2
## 193    0.001  0.250000 0.001  0.001  0.000100  11    FABP5
## 194    0.001  0.250000 0.001  0.001  0.000100  11    HIGD1B
## 195    0.001  0.250000 0.001  0.001  0.000100  11    AGT
## 196    0.001  0.250000 0.001  0.001  0.000100  11    RGS5
## 197    0.001  0.250000 0.001  0.001  0.000100  11    CPE
## 198    0.001  0.250000 0.001  0.001  0.000100  11    SSTR2
## 199    0.001  0.250000 0.001  0.001  0.000100  12    PLN
## 200    0.001  0.250000 0.001  0.001  0.000100  12    SORBS2
## 201    0.001  0.250000 0.001  0.001  0.000100  12    PHLDA2
## 202    0.001  0.250000 0.001  0.001  0.000100  12    SNCG
## 203    0.001  0.250000 0.001  0.001  0.000100  12    MT1M
## 204    0.001  0.250000 0.001  0.001  0.000100  12    MYH11
## 205    0.001  0.250000 0.001  0.001  0.000100  12    PTGDS
## 206    0.001  0.250000 0.001  0
```

```

## 1 1.75e-124      6.63 0.295 0.006 1.57e-120 0      COX4I2
## 2 5.39e-107      6.13 0.266 0.009 4.83e-103 0      RERGL
## 3 8.12e-182      7.27 0.377 0.023 7.27e-178 1      C3
## 4 6.70e-150      4.55 0.32 0.017 6.00e-146 1      SCN7A
## 5 1.55e-151      7.37 0.297 0.007 1.39e-147 2      PCSK6
## 6 6.37e-173      7.09 0.267 0.003 5.71e-169 2      NSG1

```

```

signatureGene <- c(
  "PLN", "SORBS2", "PHLDA2", "SNCG", "MT1M", "MYH11",
  "PTGDS", "FBLN1", "DCN", "LUM", "COL1A1", "LTBP2",
  "FABP5", "HIGD1B", "AGT", "RGS5", "CPE", "SSTR2")

```

```

signatureGeneMarker <- caf_markers %>%
  filter(gene %in% signatureGene) %>%
  mutate(gene = factor(gene, levels = signatureGene)) %>%
  arrange(gene)

```

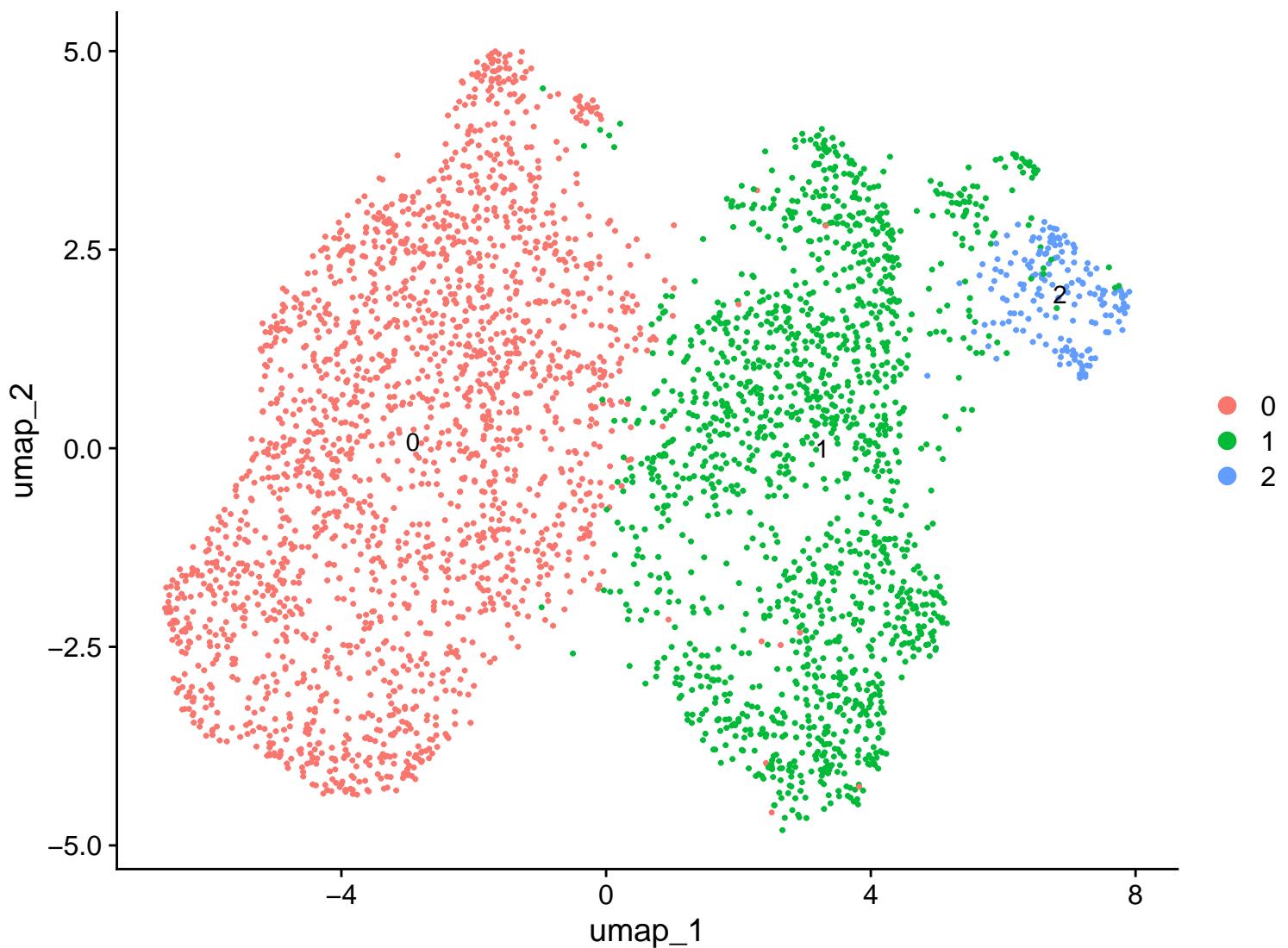
```
DoHeatmap(caf_umap, features = signatureGene) + NoLegend()
```



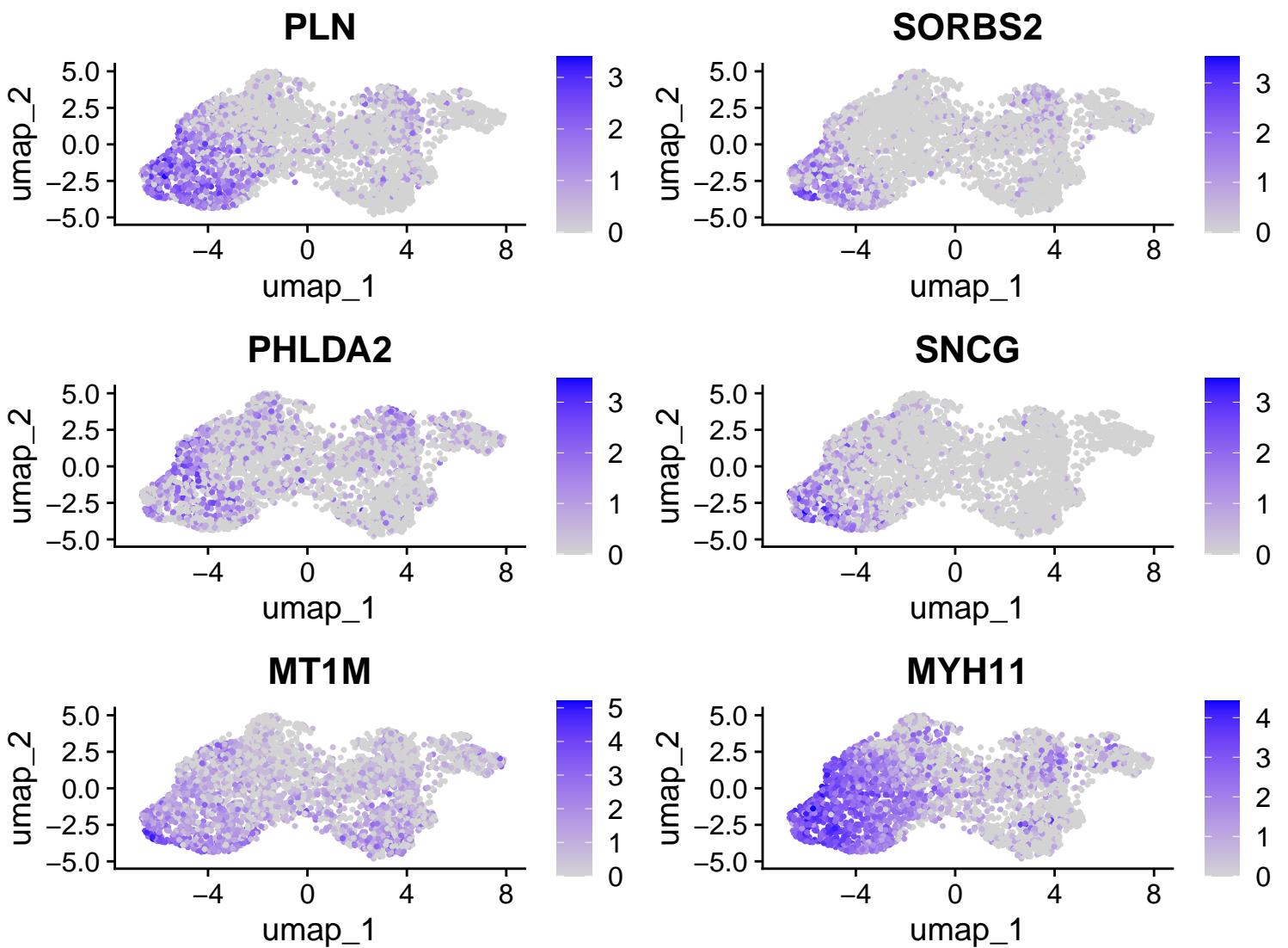
```

# UMAP clusters
DimPlot(caf_umap, reduction = "umap", label = TRUE)

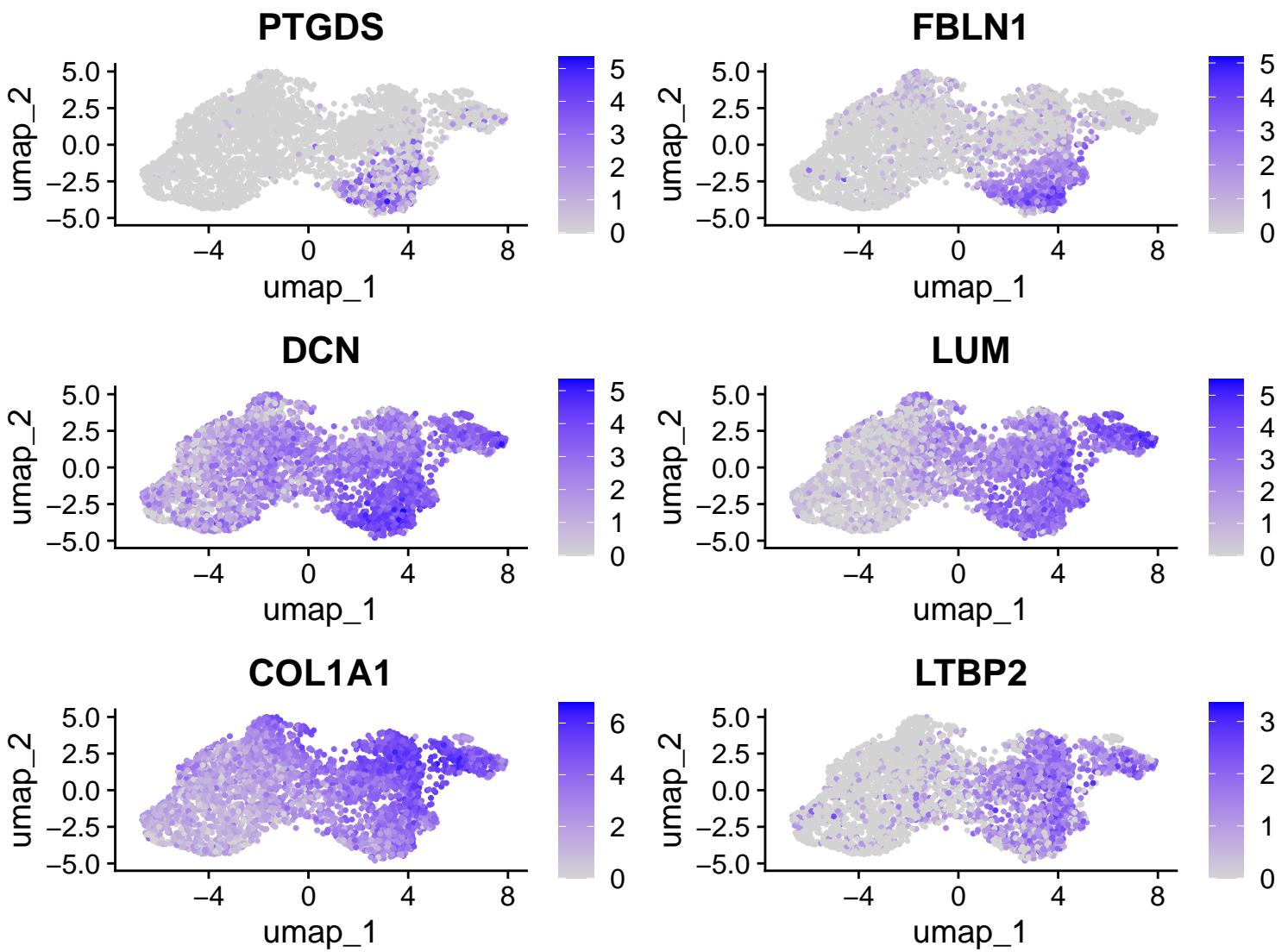
```



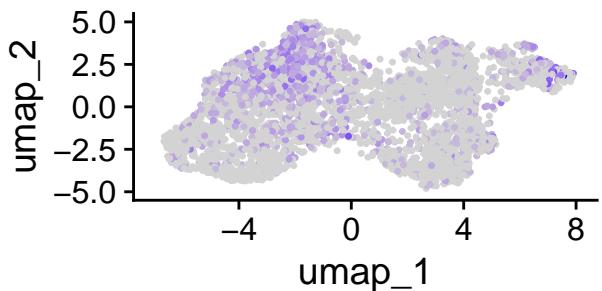
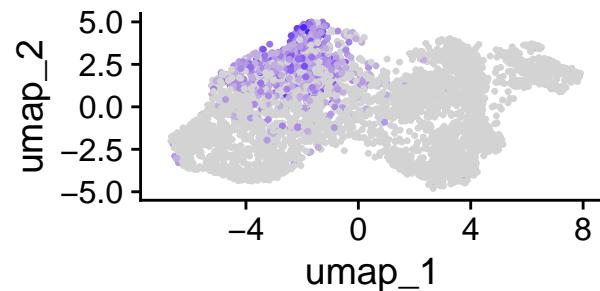
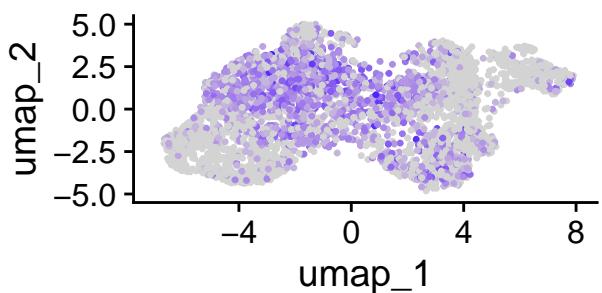
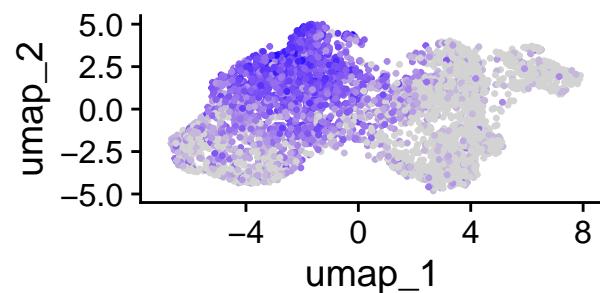
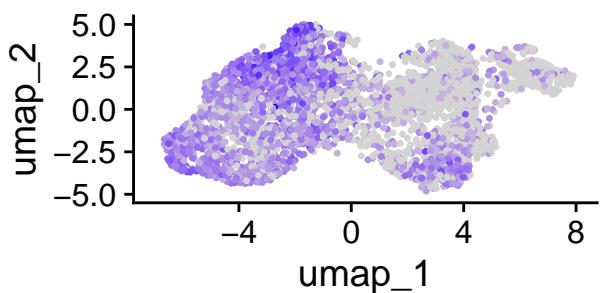
```
# FeaturePlots par famille
FeaturePlot(caf_umap, features = c("PLN", "SORBS2", "PHLDA2", "SNCG", "MT1M", "MYH11"))
```



```
FeaturePlot(caf_umap, features = c("PTGDS", "FBLN1", "DCN", "LUM", "COL1A1", "LTBP2"))
```



```
FeaturePlot(caf_umap, features = c("FABP5", "HIGD1B", "AGT", "RGS5", "CPE", "SSTR2"))
```

FABP5**HIGD1B****AGT****RGS5****CPE****SSTR2**