

# Cd52 and Macrophage differentiation

Domien Vanneste

2025-09-05 15:20:09 +0200

## Contents

<b>Introduction</b>	<b>3</b>
<b>Load packages</b>	<b>3</b>
<b>Download data Kayvanjoo et al 2024 (Elvira Mass Lab) from GEO (GSE225443)</b>	<b>3</b>
<b>Load data and construct Seurat Object</b>	<b>3</b>
<b>Standard pre-processing workflow</b>	<b>4</b>
QC and selecting cells for further analysis . . . . .	4
Normalizing the data . . . . .	6
Identification of highly variable features . . . . .	6
Scaling the data . . . . .	7
<b>Perform linear dimensional reduction</b>	<b>7</b>
Determine the ‘dimensionality’ of the dataset . . . . .	9
<b>Cluster the cells</b>	<b>10</b>
<b>Run non-linear dimensional reduction (UMAP)</b>	<b>11</b>
<b>Score for EMP, pMac, Mac and liverMac signatures (Mass et al 2016)</b>	<b>13</b>
<b>Subset EMP, pMac and Mac</b>	<b>21</b>
<b>Reprocess data</b>	<b>22</b>
<b>Recluster data</b>	<b>24</b>
<b>Assigning cell type identity to clusters</b>	<b>25</b>
<b>Finding differentially expressed features</b>	<b>30</b>

Perform Slingshot pseudotime analysis	31
Cd52 expression WT vs KO	35

# Introduction

To generate a single-cell transcriptomic map of macrophage differentiation, we selected clusters of mac and mac progenitors from recently published scRNA-seq data of FACS-sorted CD11blow/+ fetal liver cells at embryonic day of development E14.5 (Kayvanjoo et al., 2014), based on the expression EMP, preMac and Mac signature genes (Mass et al., 2016). Cells were reclustered using the FindClusters function from the Seurat package (4.3.0). To evaluate trajectory-based pseudotime analysis, we used the Slingshot package (2.14.0) with EPM as starting cluster and mac as ending cluster. Using Slingshot pseudotime, we identified Cd52 as highly expressed in pre-mac but downregulated at the pre-mac to mac transition, coinciding with increased Mafb expression. Interestingly, Cd52 was higher expressed in most Mafb-deficient RTMs as compared to their WT counterparts, with the exception of Ly6C+ CMs.

## Load packages

```
suppressMessages({  
  library(SeuratObject)  
  library(Seurat)  
  library(dplyr)  
  library(readxl)  
  library(slingshot)  
  library(DelayedMatrixStats)  
  library(ggplot2)  
})
```

## Download data Kayvanjoo et al 2024 (Elvira Mass Lab) from GEO (GSE225443)

Amir Hossein KayvanjooIva SplichalovaDavid Alejandro BejaranoHao HuangKatharina MauelNikola MakdissiDavid HeiderHui Ming TewNora Reka BalzerEric GretoCollins Osei-SarpongKevin BaßlerJoachim L SchultzeStefan UderhardtEva KiermaierMarc BeyerAndreas SchlitzerElvira Mass (2024) Fetal liver macrophages contribute to the hematopoietic stem cell niche by controlling granulopoiesis eLife 13:e86493. <https://doi.org/10.7554/eLife.86493>

```
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE225443
```

## Load data and construct Seurat Object

```
# Use the correct file path format for Windows  
dirs <- list.dirs(path = "C:/Users/domie/Documents/scRNA-Seq_Mass_2016/Mass_2023",  
  full.names = FALSE, recursive = FALSE)  
  
list_sample <- list()  
  
for (directory in dirs) {  
  # Construct a platform-independent path  
  data_path <- file.path("C:/Users/domie/Documents/scRNA-Seq_Mass_2016/Mass_2023",
```

```

    directory, "filtered_feature_bc_matrix")

# Read the 10x data
mass.data <- Read10X(data.dir = data_path, gene.column = 1)

# Create Seurat object
mass_group <- CreateSeuratObject(counts = mass.data, project = substr(directory,
  1, nchar(directory) - 27), min.cells = 3, min.features = 200)

# Append to list
list_sample <- append(list_sample, mass_group)
}

# Merge all Seurat objects
mass_cells <- merge(list_sample[[1]], y = list_sample[-1], add.cell.ids = as.character(seq_along(list_sample)),
  project = "mass_cells")

```

## Standard pre-processing workflow

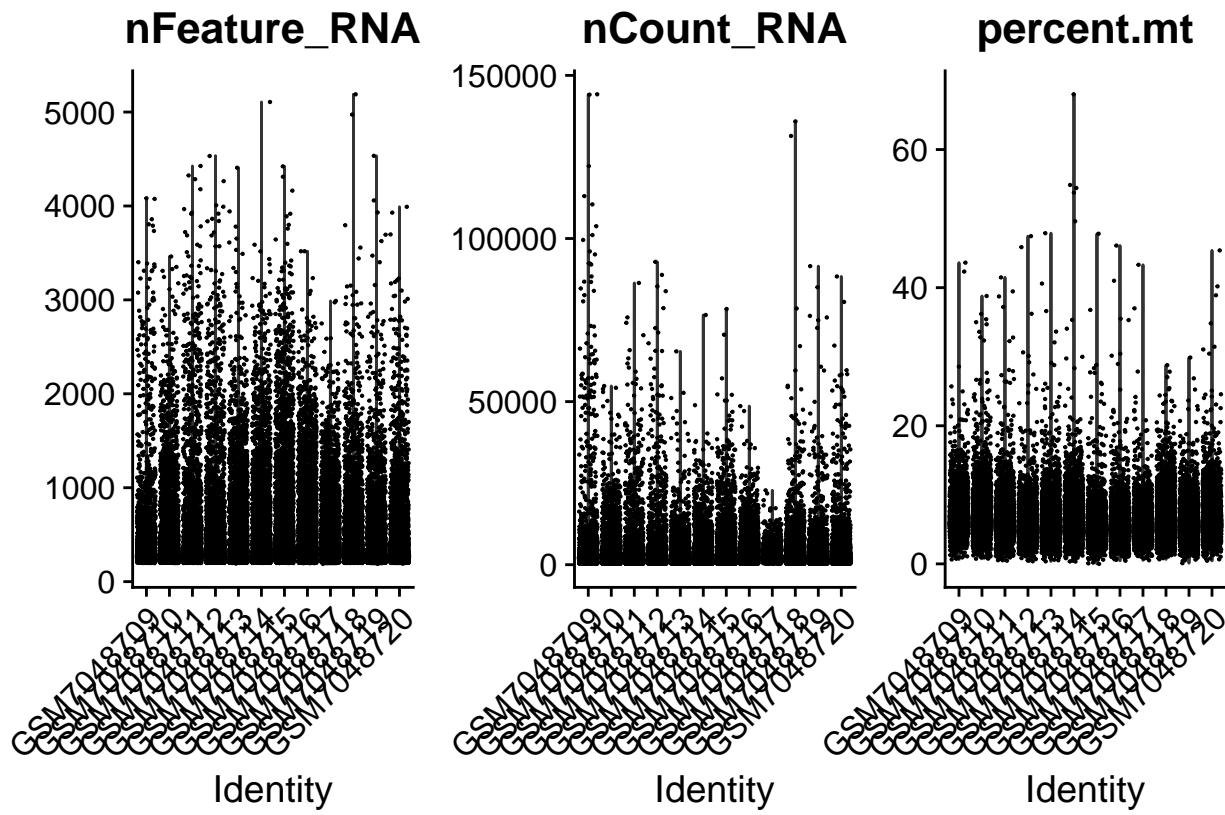
### QC and selecting cells for further analysis

```

mass_cells[["percent.mt"]] <- PercentageFeatureSet(mass_cells,
  pattern = "^\$mt\$")

VlnPlot(mass_cells, features = c("nFeature_RNA", "nCount_RNA",
  "percent.mt"), ncol = 3)

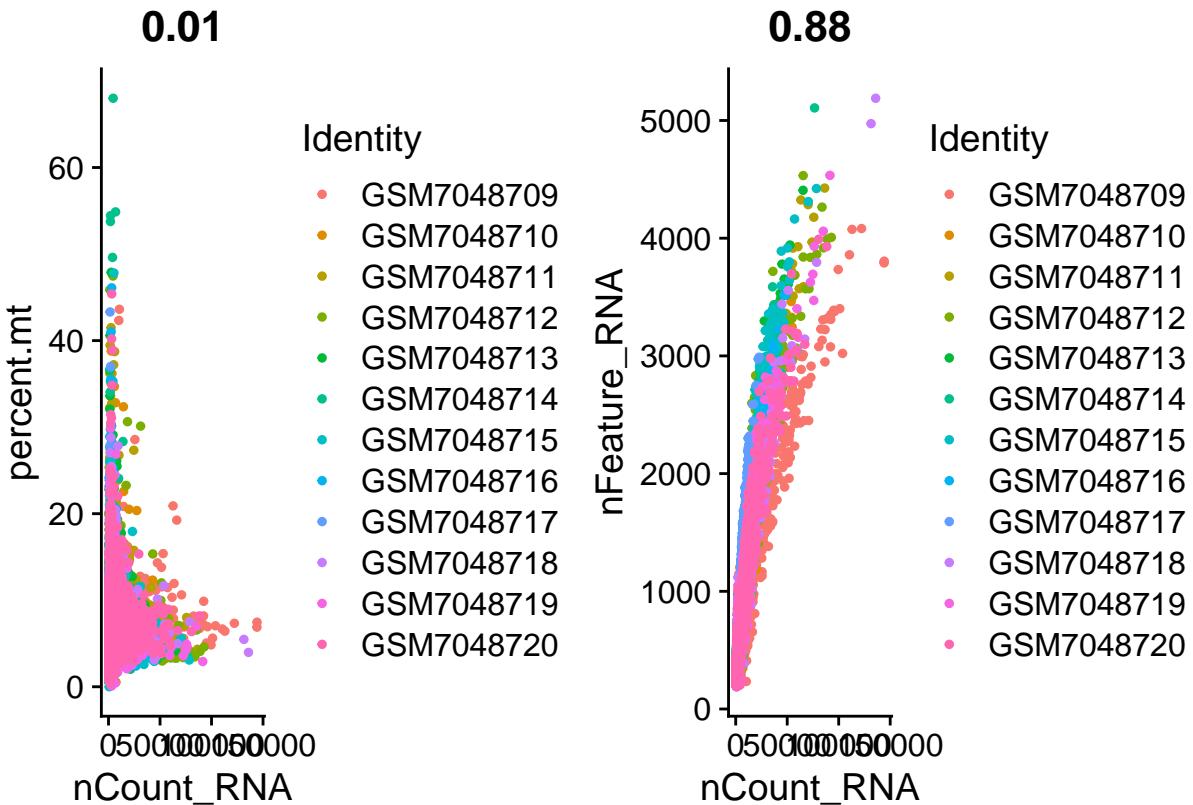
```



```

plot1 <- FeatureScatter(mass_cells, feature1 = "nCount_RNA",
                         feature2 = "percent.mt")
plot2 <- FeatureScatter(mass_cells, feature1 = "nCount_RNA",
                         feature2 = "nFeature_RNA")
plot1 + plot2

```



```
mass_cells <- subset(mass_cells, subset = nFeature_RNA > 200 &
nFeature_RNA < 3500 & percent.mt < 15)
```

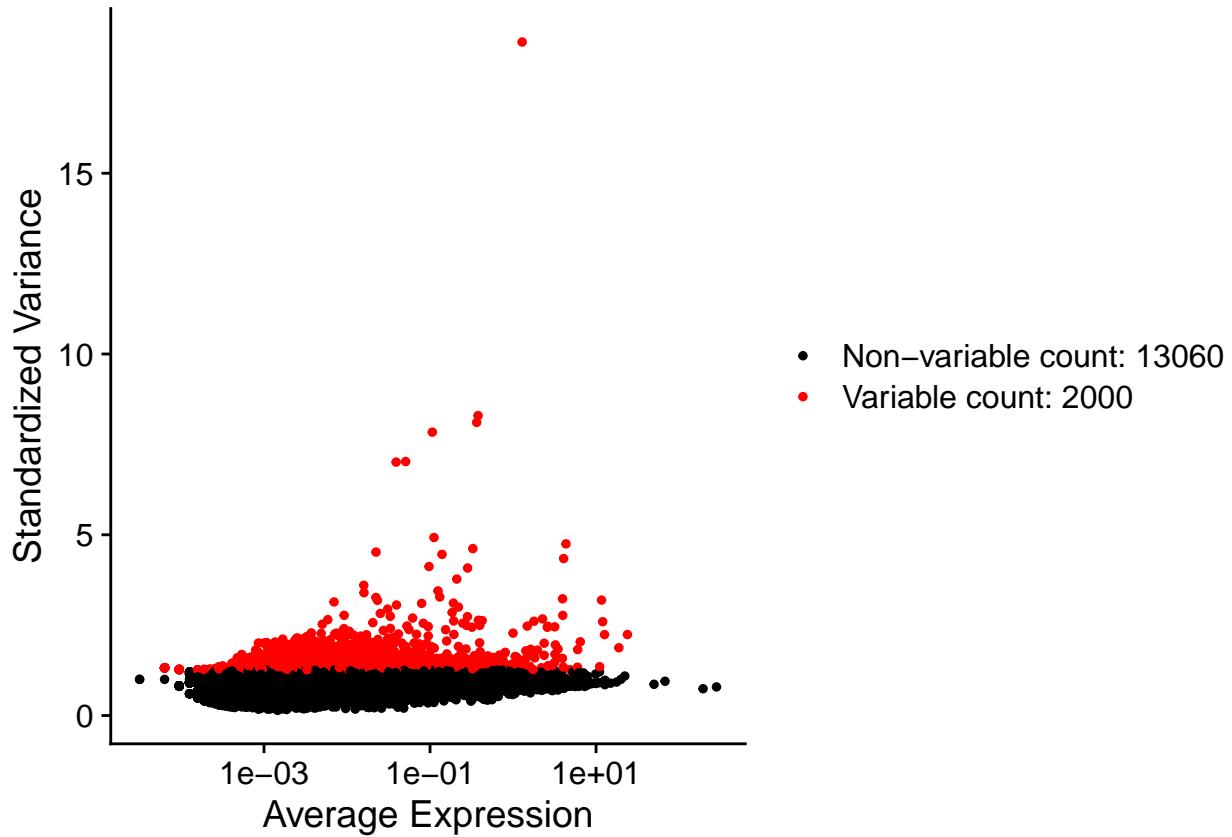
## Normalizing the data

```
mass_cells <- NormalizeData(mass_cells)
```

## Identification of highly variable features

```
mass_cells <- FindVariableFeatures(mass_cells, selection.method = "vst",
nfeatures = 2000)

# plot variable features without labels
VariableFeaturePlot(mass_cells)
```



## Scaling the data

```
all.genes <- rownames(mass_cells)
mass_cells <- ScaleData(mass_cells, features = all.genes)

## Centering and scaling data matrix
```

## Perform linear dimensional reduction

```
mass_cells <- RunPCA(mass_cells, features = VariableFeatures(object = mass_cells))

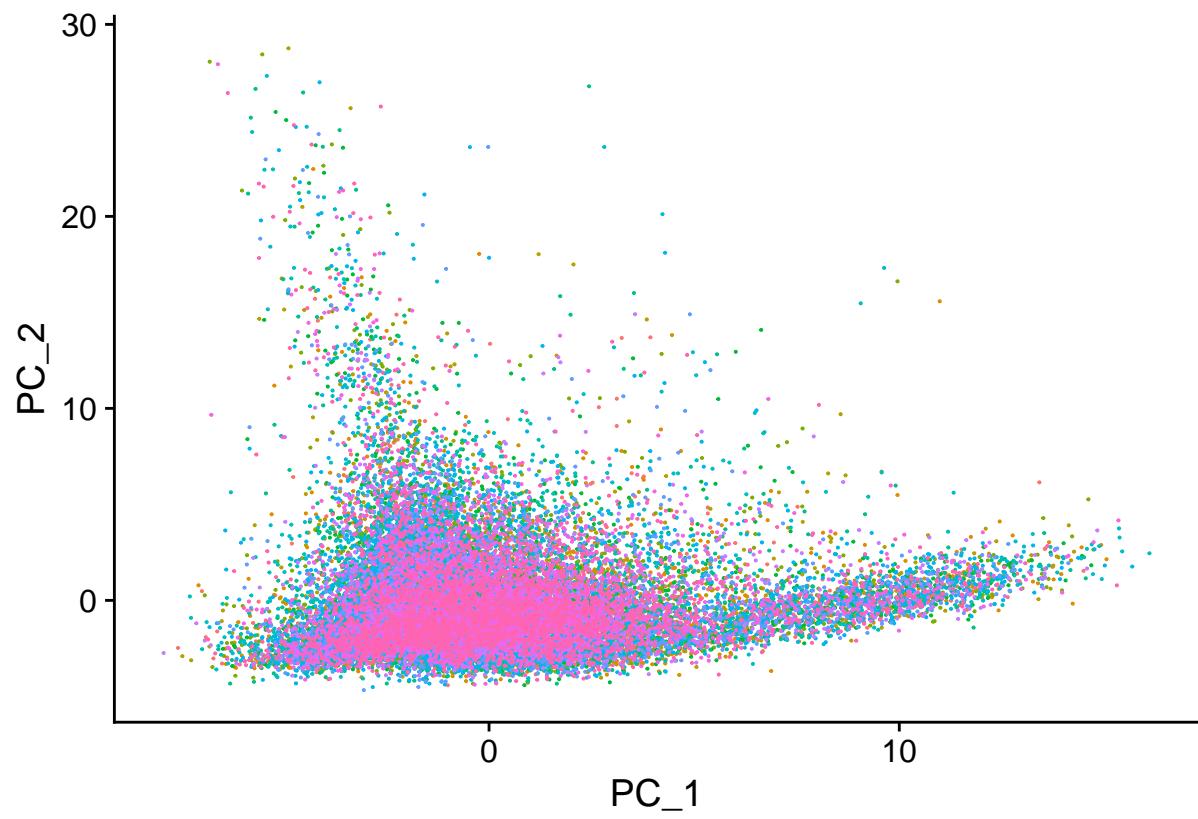
## PC_1
## Positive: S100a9, S100a8, Stfa1, Camp, BC100530, Ngp, Lcn2, Ltf, Olfm4, Gm5483
##     Stfa211, Syne1, Lyz2, Pglyrp1, C3, Fcnb, Thbs1, Hp, Cybb, Chill
##     Cd177, Serpinb1a, Mmp8, Ckap4, Cebpe, Retnlg, 6430548M08Rik, Mmp9, Ncam1, Elane
## Negative: H2afy, Hmgaa2, Ccnd2, Mybbp1a, Sox4, Ccnd1, Dnmt3b, Flt3, Cd93, Adgrg1
##     Jun, Egfl7, Tsc22d1, Irf8, Dusp5, Fos, Prmt1, Gimap6, Angpt1, Calr
##     Ighm, Cluh, Csf1r, Adgrl4, Slc22a3, Ctr9, Abl2, H19, Gimap1, Atp13a2
## PC_2
## Positive: C1qc, Ctss, C1qb, Ctsb, Slc40a1, Psap, Lpl, Mrc1, Csf1r, Mpeg1
```

```

##      Ctsc, Hmox1, Selenop, Clec7a, Lyz2, Cybb, Mafb, Sdc3, Abca1, C1qa
##      Fcna, Rassf4, Lgals3, Cfhl, Atf3, Fos, Grn, Cfp, Igf1, Cd51
## Negative: Mpo, Prtn3, Elane, Ctsg, Cdk6, Myb, Hmga2, Serpinb1a, Srgn, Cd63
##      H2afy, Adgrg1, Dnmt3b, Sox4, Cd93, Calr, Prss57, Atp8b4, Ccnd2, Mybbp1a
##      Alas1, Angpt1, Fut8, Kdm5b, Adgrl4, Egfl7, Gimap6, Runx3, Gpc1, Zbtb16
## PC_ 3
## Positive: Cd63, Srgn, Mpo, Atp8b4, Calr, Serpinb1a, Hdc, Cpa3, Gata2, Ctsg
##      Syne1, Fut8, Elane, Myb, Mcpt8, Prtn3, Cdk6, Ms4a2, Ifitm1, Lmo4
##      Apoe, Ncam1, Alas1, Tmed3, Cd200r3, Kdm5b, Prss57, Top2a, Egr1, Csf1
## Negative: Lyz2, Clec7a, Ahnak, Hbb-bs, Cd74, Ctsc, Ccr2, H2-Ab1, Mpeg1, Hbb-bt
##      Ctss, Vcan, Atf3, Rassf4, H2-Eb1, Psap, Klf4, Hba-a1, H2-Aa, Cfhl
##      Hba-a2, Lgals3, Nrp1, Olfm1, Il1b, Irf8, Siglech, Plcb1, Ciita, Slc4a1
## PC_ 4
## Positive: Prtn3, Elane, Mpo, Ctsg, Calr, Atp8b4, Prss57, F13a1, Alas1, Serpinb1a
##      Gpc1, Tmed3, Fcnb, Ctsb, Cdk6, Vcam1, Srgn, Myb, Hp, Gm11505
##      C1qb, C1qc, Mrc1, Creld2, Lpl, Slc40a1, Stfa1, Cd63, Chdh, Fcna
## Negative: Ifitm1, Gata2, Mcpt8, Cpa3, Dusp5, Ms4a2, Ltf, Cd200r3, Padi2, Lamb1
##      Ngp, Egr1, Csf1, Nfkbiz, Lcn2, Slc22a3, Stfa2l1, Fam129a, Mmp8, Il16
##      Sept1, Cd69, Cd177, Retnlg, Prss34, Angpt1, Chill, Il17r, Gimap1, Sox4
## PC_ 5
## Positive: Hmga2, Sox4, Dnmt3b, Flt3, H2afy, Gimap6, Ccnd2, Cd93, Tsc22d1, Ccnd1
##      Egfl7, Ltf, Ngp, Lcn2, Car2, Adgrl4, Trbc2, Adgrg1, Stfa2l1, Igdm
##      Hlf, H19, Camp, Mybbp1a, Rgl1, Ptp4a3, Gpr171, Gm5483, Plk2, Olfm4
## Negative: Mcpt8, Cpa3, Gata2, Ifitm1, Ms4a2, Cd200r3, Csf1, Lamb1, Hdc, Emilin2
##      Cd63, Padi2, Il16, Egr1, Srgn, Prss34, Mpo, Perp, F13a1, Fam129a
##      Ccr2, Atp8b4, Galnt6, Cyp11a1, Alox15, Itgb7, Ctsc, Ahnak, Nfkbiz, Apoe

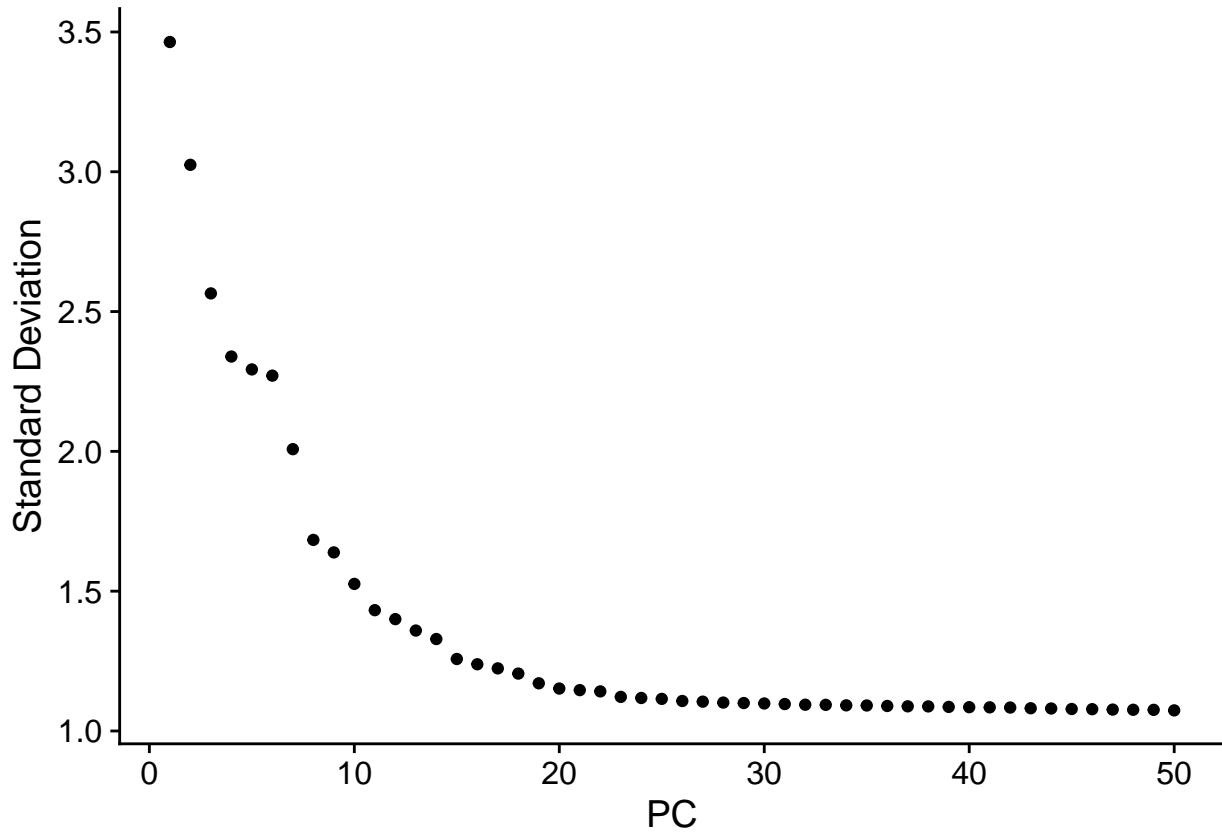
```

```
DimPlot(mass_cells, reduction = "pca") + NoLegend()
```



Determine the ‘dimensionality’ of the dataset

```
ElbowPlot(mass_cells, ndims = 50)
```



## Cluster the cells

```
mass_cells <- FindNeighbors(mass_cells, dims = 1:20)

## Computing nearest neighbor graph

## Computing SNN

mass_cells <- FindClusters(mass_cells, resolution = 1.2)

## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 31522
## Number of edges: 984336
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8180
## Number of communities: 22
## Elapsed time: 5 seconds
```

## Run non-linear dimensional reduction (UMAP)

```
mass_cells <- RunUMAP(mass_cells, dims = 1:20)
```

```
## Warning: The default method for RunUMAP has changed from calling Python UMAP via reticulate to the R
## To use Python UMAP via reticulate, set umap.method to 'umap-learn' and metric to 'correlation'
## This message will be shown once per session
```

## 15:22:28 UMAP embedding parameters a = 0.9922 b = 1.112

## 15:22:28 Read 31522 rows and found 20 numeric columns

## 15:22:28 Using Annoy for neighbor search, n\_neighbors = 30

```
## 15:22:28 Building Annoy index with metric = cosine, n_trees = 50
```

## 0% 10 20 30 40 50 60 70 80 90 100%

## \*\*\*\*|

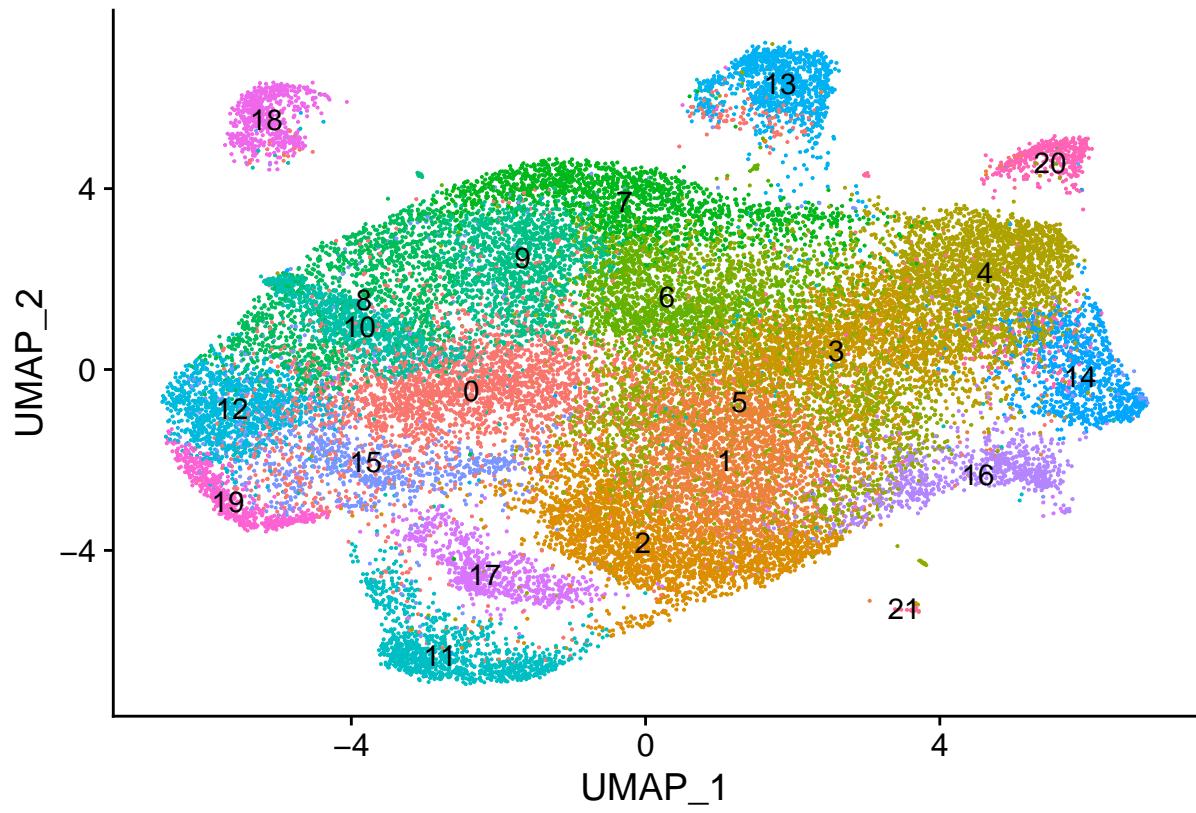
## 15:22:30 Writing NN index file to temp file C:\Users\domie\AppData\Local\Temp\NNIndex\NNIndex

## 15:22:30 Searching Annoy index using 1 thread, search\_k = 3000

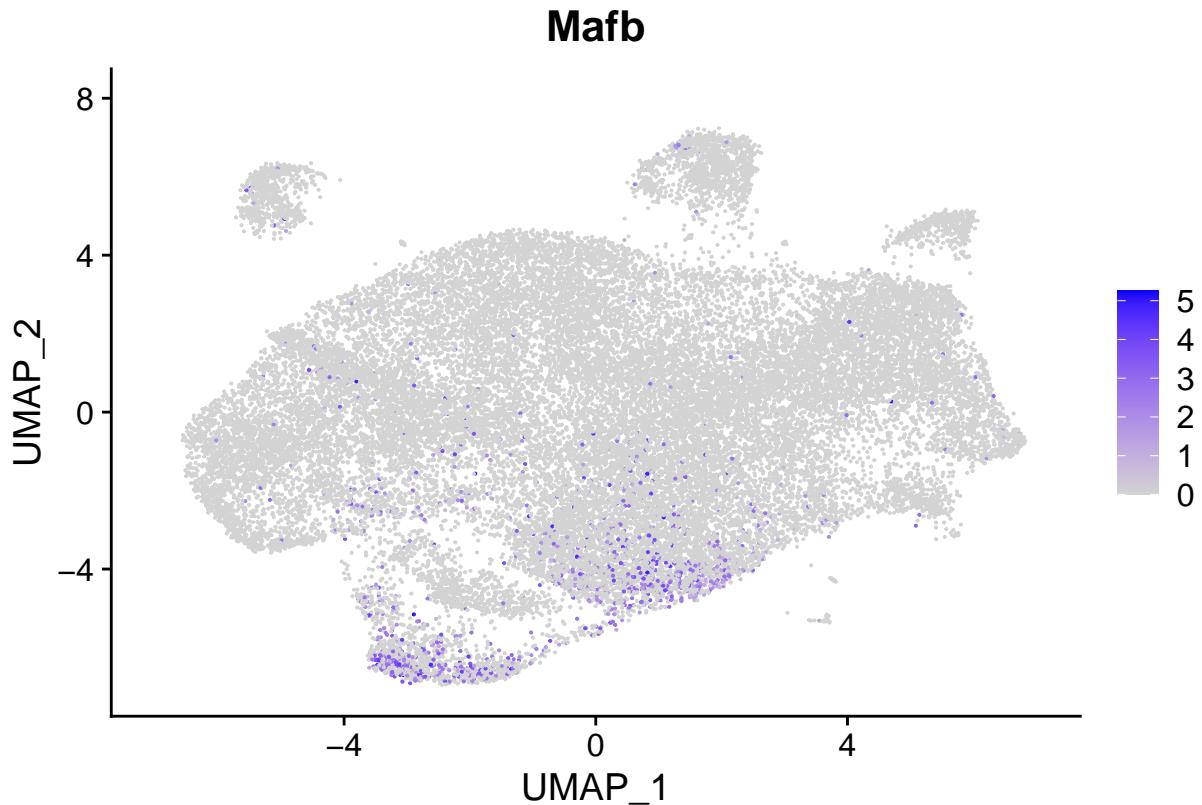
## 15:22:38 Annoy recall = 100%

```
## 15:22:39 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 15:22:41 Initializing from normalized Laplacian + noise (using irlba)
## 15:22:42 Commencing optimization for 200 epochs, with 1410922 positive edges
## 15:23:07 Optimization finished
```

```
DimPlot(mass_cells, reduction = "umap", label = T) + NoLegend()
```



```
FeaturePlot(mass_cells, features = "Mafb")
```



Score for EMP, pMac, Mac and liverMac signatures (Mass et al 2016)

```

signatures <- read_excel("signatures.xlsx")

EMP_sign <- list(na.omit(signatures$EMP_sign))

pMac_sign <- list(na.omit(signatures$pMac_sign))

Mac_sign <- list(na.omit(signatures$Mac_sign))

liverMac_sign <- list(na.omit(signatures$liverMac_sign))

mass_cells <- AddModuleScore(mass_cells, features = EMP_sign,
                               name = "EMP_sign")

```

```

## Warning: The following features are not present in the object: Diap1, Fam65a,
## Hba-x, Hbb-bh1, not searching for symbol synonyms

```

```

mass_cells <- AddModuleScore(mass_cells, features = pMac_sign,
                               name = "pMac_sign")

```

```

## Warning: The following features are not present in the object: 4632428N05Rik,
## not searching for symbol synonyms

mass_cells <- AddModuleScore(mass_cells, features = Mac_sign,
                               name = "Mac_sign")

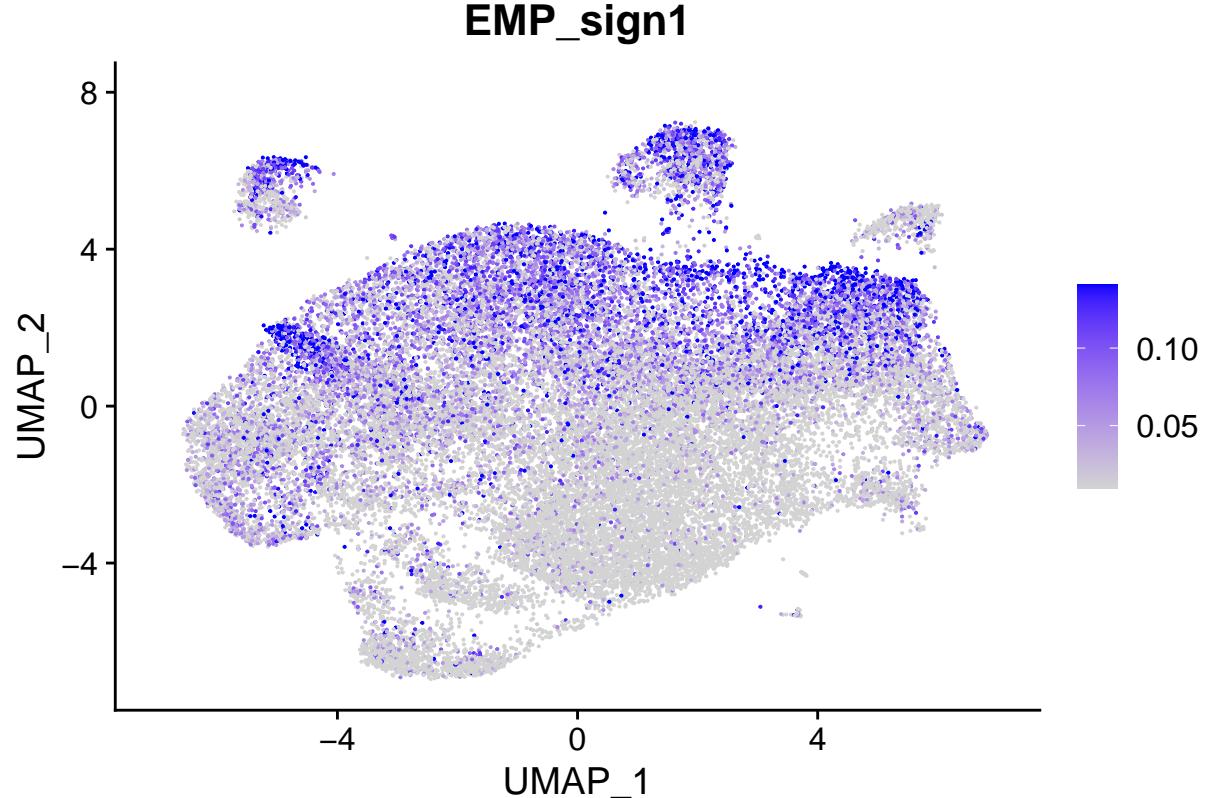
## Warning: The following features are not present in the object: Sepp1, not
## searching for symbol synonyms

mass_cells <- AddModuleScore(mass_cells, features = liverMac_sign,
                               name = "liverMac_sign")

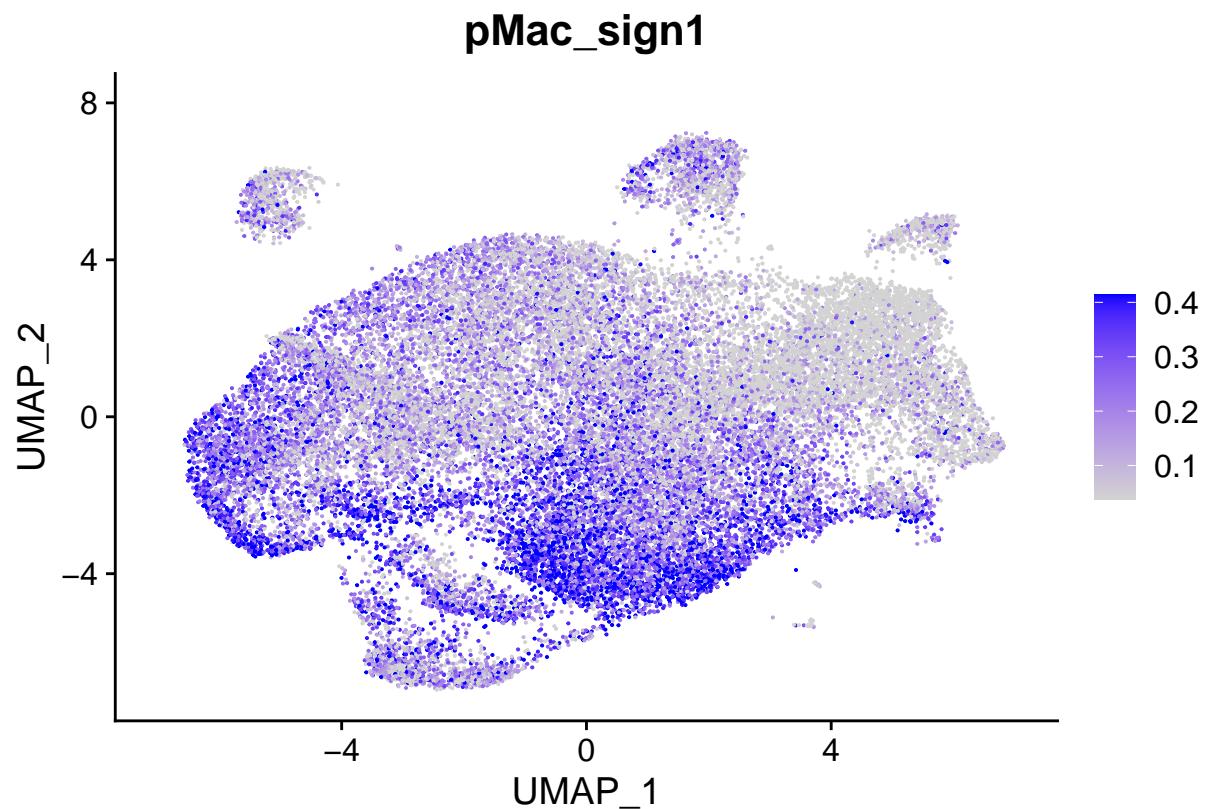
## Warning: The following features are not present in the object: 6030468B19Rik,
## A3galt2, Abcg4, Ackr1, Adh1, Adm, Agt, Aldob, Ambp, Apoa1, Apoc4, Atp7b,
## BC094916, Ccrn41, Cd207, Cd209f, Cdc42ep2, Ces1c, Clec1a, E130309D14Rik, Eltd1,
## Enpep, Epb4.1, Epb4.2, F2, F930017D23Rik, Fabp1, Fgb, Fgg, Fn3k, Gc, Gm11440,
## Gm15800, Gm28049, Gm826, Gpd1, Gpr182, Hbq1b, Hesx1, Hmgcs2, Hyal4, Hyal6,
## Igfbp1, Itih2, Itih3, Klf15, Mageb16, Mat1a, Mmp15, Mrap, Nhlrc4, Pcdha10,
## Pdia2, Ptgis, Rgn, Sardh, Sdpr, Serpina1a, Serpina1b, Serpina1c, Serpina1d,
## Serpinale, Sh3tc2, Shroom2, Slc38a5, Spats2l, Tmem56, Tslp, Ttc36, Umod11,
## Wdr65, not searching for symbol synonyms

FeaturePlot(mass_cells, features = "EMP_sign1", min.cutoff = "q10",
            max.cutoff = "q90")

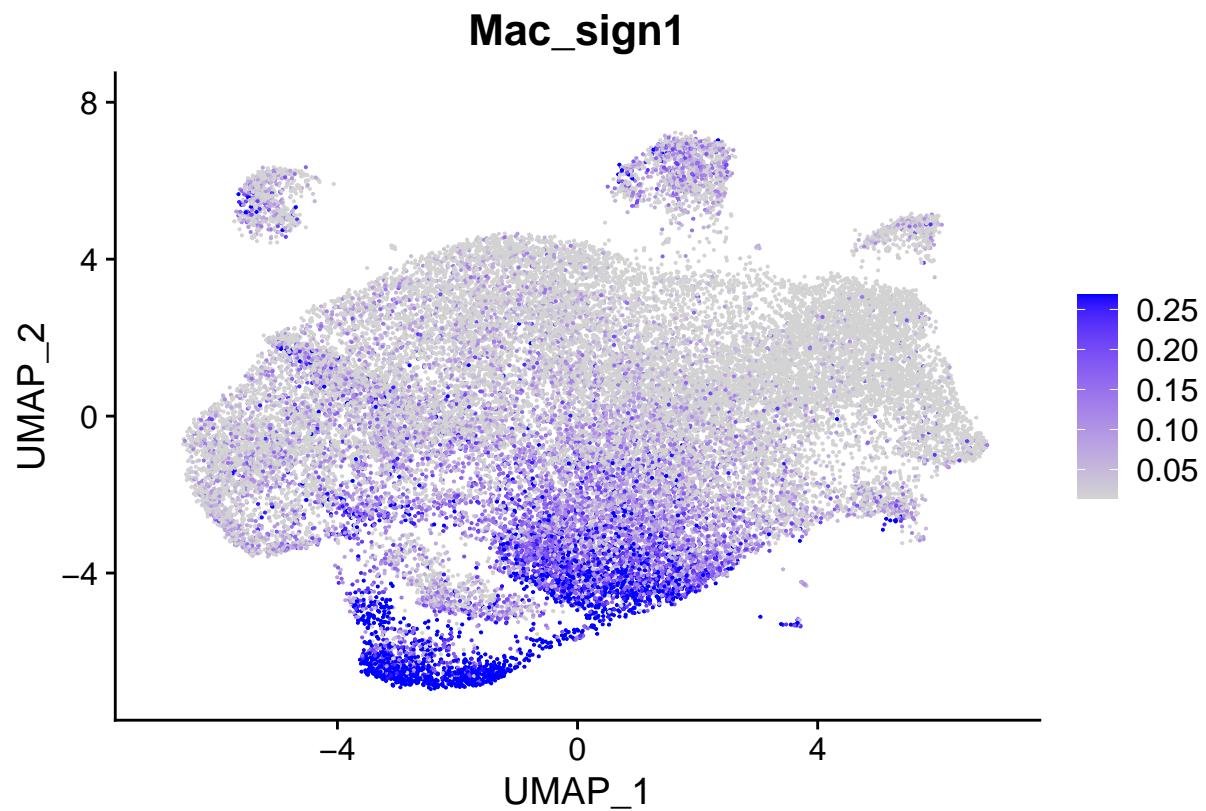
```



```
FeaturePlot(mass_cells, features = "pMac_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```

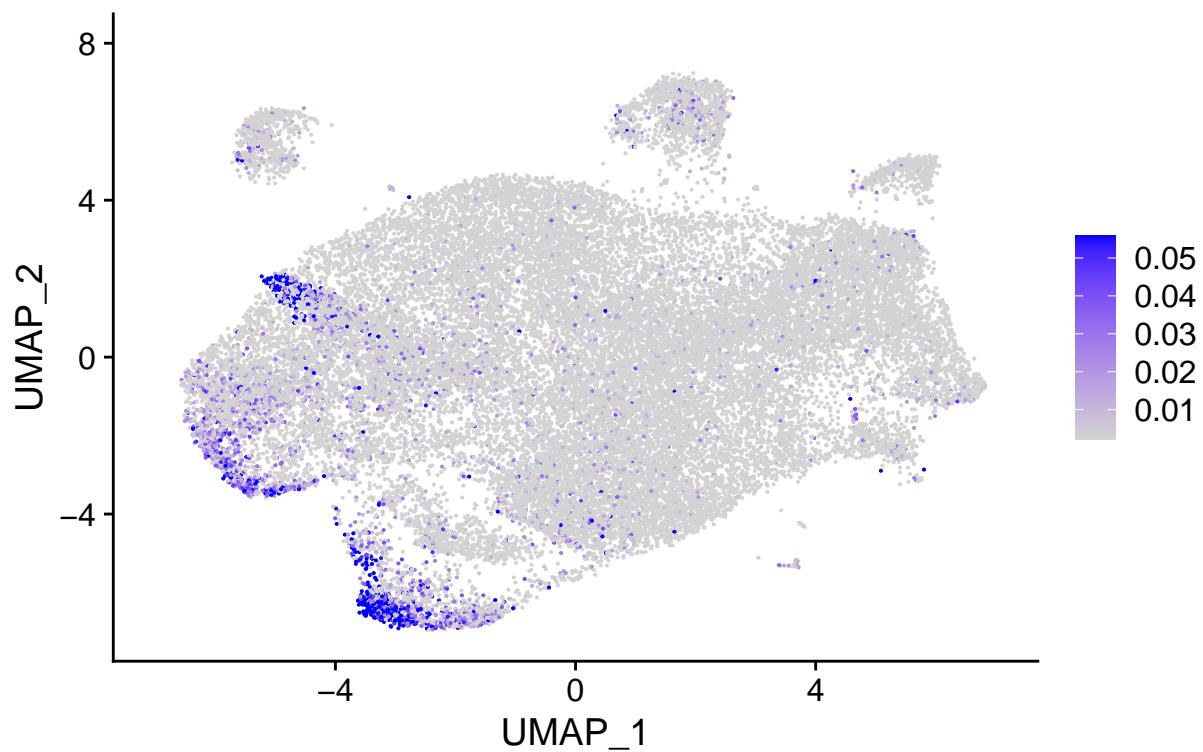


```
FeaturePlot(mass_cells, features = "Mac_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```



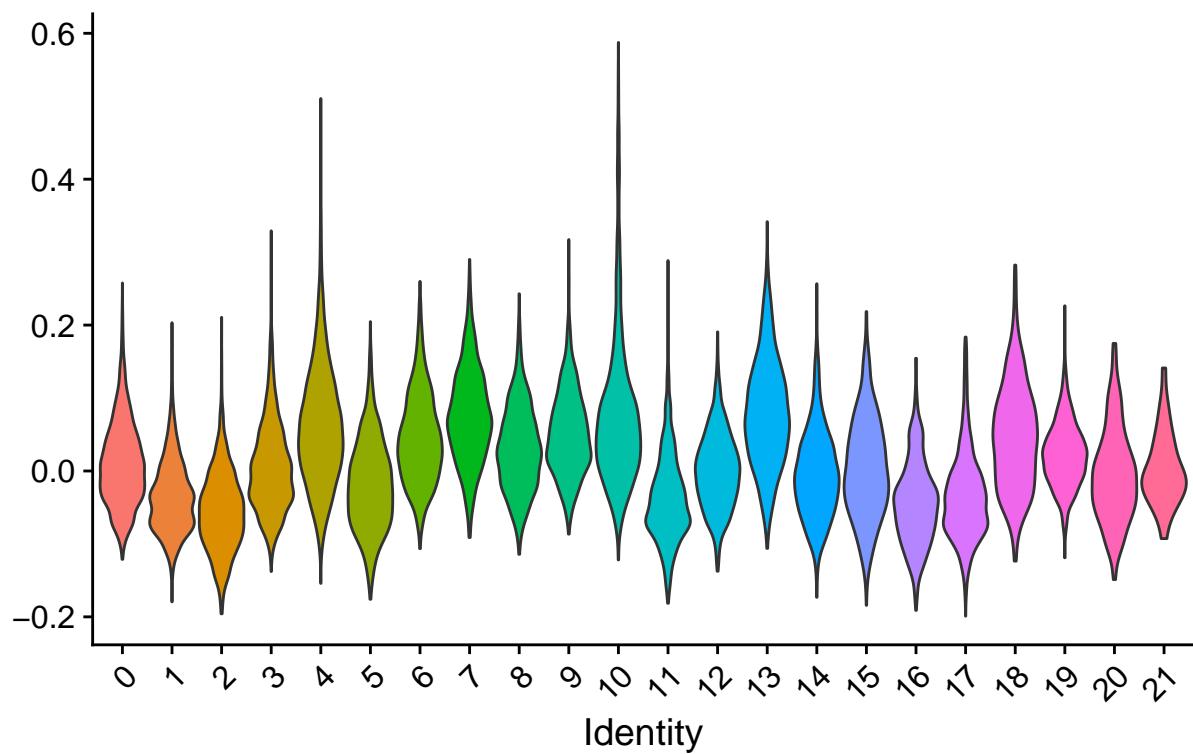
```
FeaturePlot(mass_cells, features = "liverMac_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```

**liverMac\_sign1**



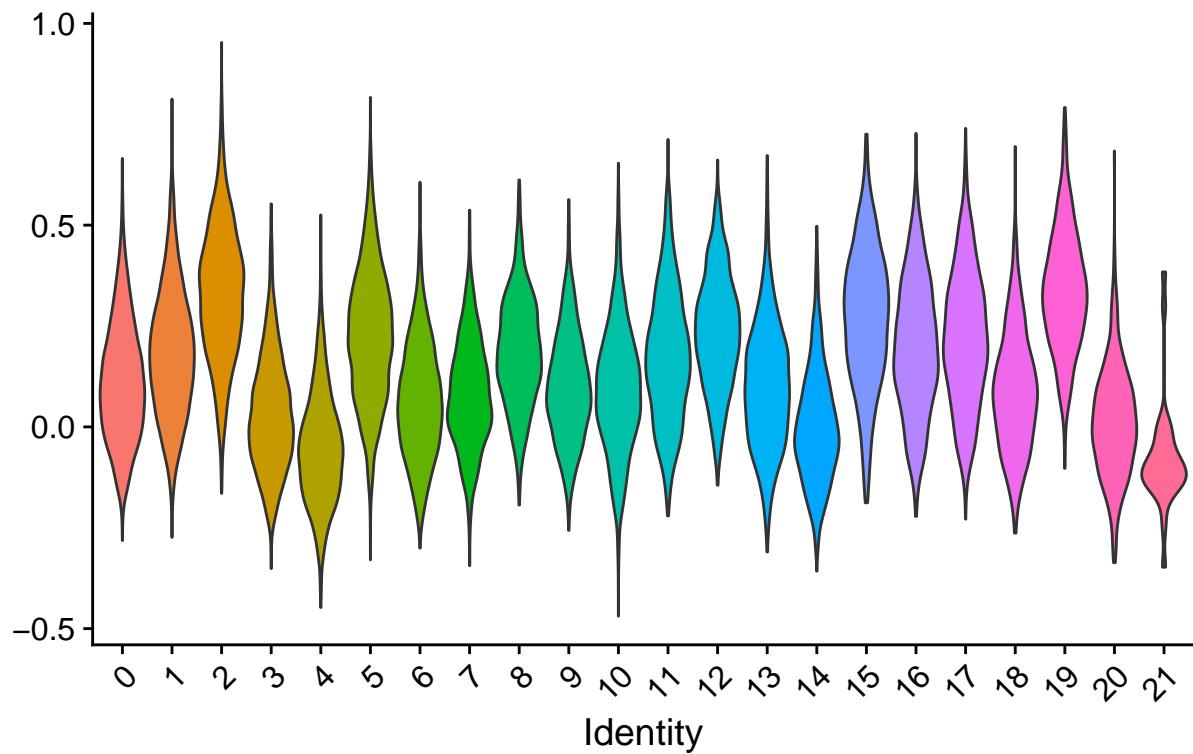
```
VlnPlot(mass_cells, features = "EMP_sign1", pt.size = 0) + NoLegend()
```

## EMP\_sign1



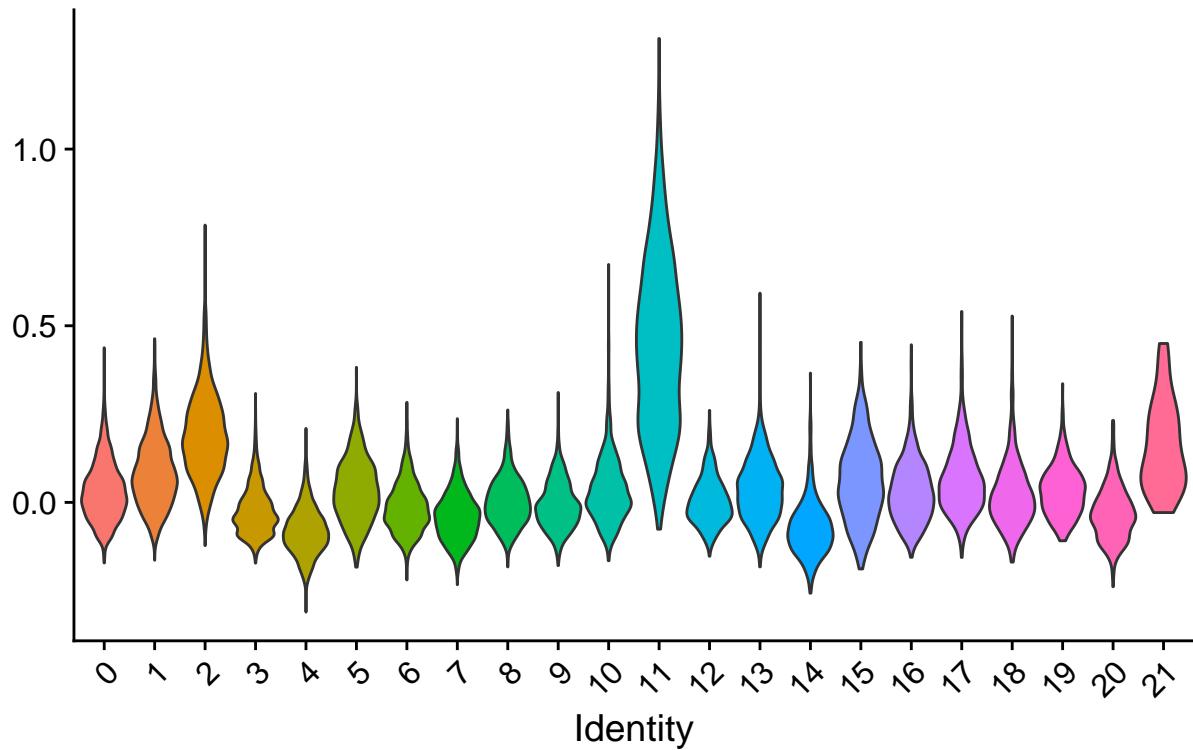
```
VlnPlot(mass_cells, features = "pMac_sign1", pt.size = 0) + NoLegend()
```

## pMac\_sign1



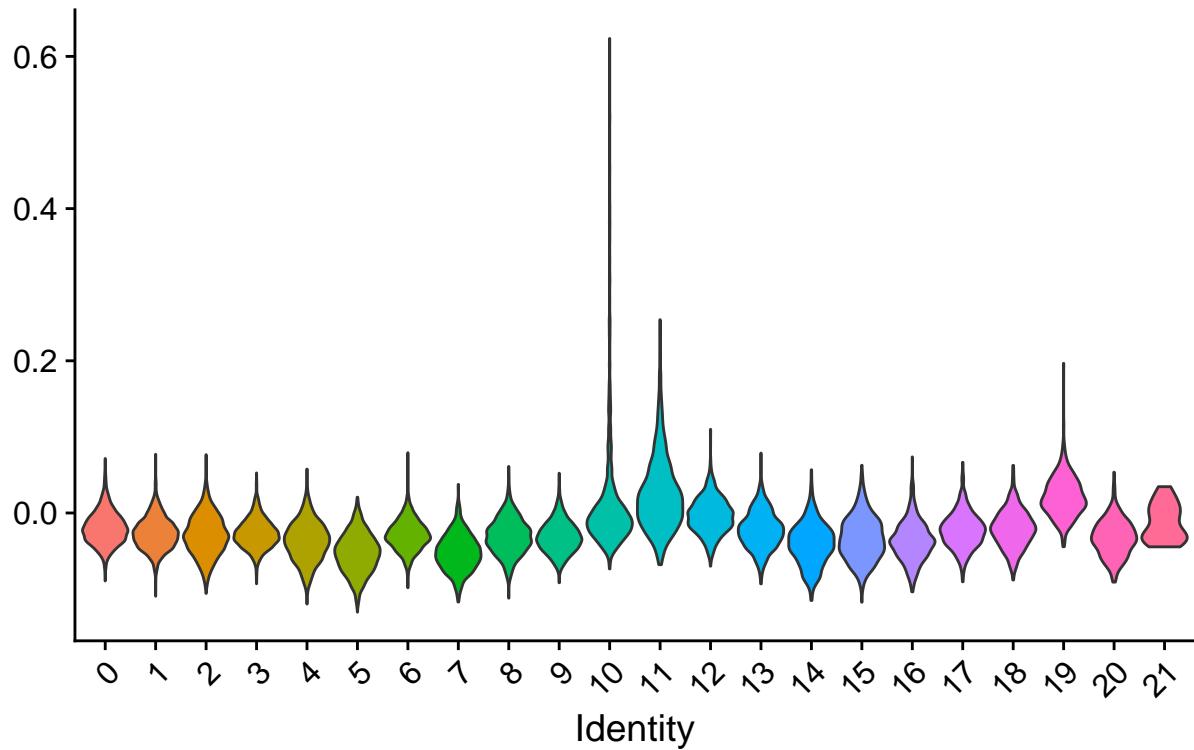
```
VlnPlot(mass_cells, features = "Mac_sign1", pt.size = 0) + NoLegend()
```

## Mac\_sign1



```
VlnPlot(mass_cells, features = "liverMac_sign1", pt.size = 0) +  
  NoLegend()
```

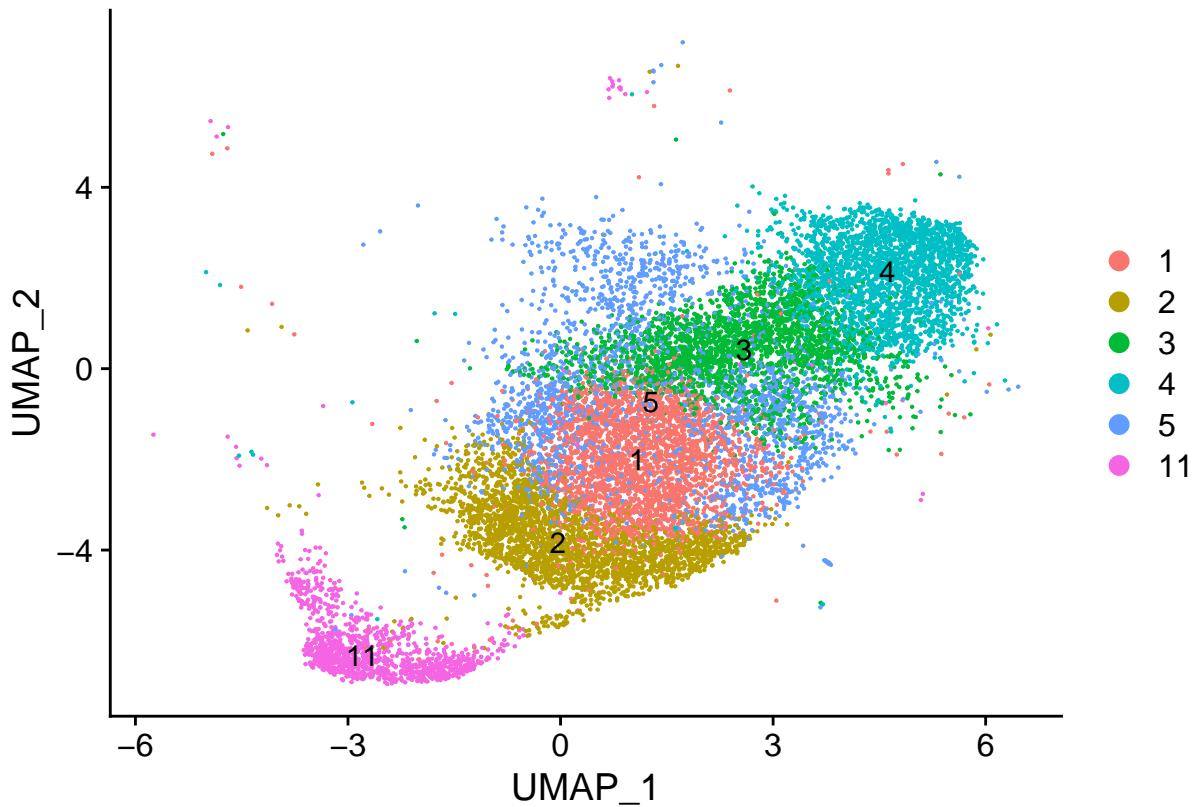
## liverMac\_sign1



```
saveRDS(mass_cells, file = "mass_cells.rds")
```

## Subset EMP, pMac and Mac

```
mac_diff <- subset(mass_cells, idents = c(1, 2, 3, 4, 5, 11))
DimPlot(mac_diff, label = T)
```



```
rm(mass_cells)
```

## Reprocess data

```
mac_diff <- FindVariableFeatures(mac_diff, selection.method = "vst",
                                    nfeatures = 2000)
all.genes <- rownames(mac_diff)
mac_diff <- ScaleData(mac_diff, features = all.genes)

## Centering and scaling data matrix

mac_diff <- RunPCA(mac_diff, features = VariableFeatures(object = mac_diff))

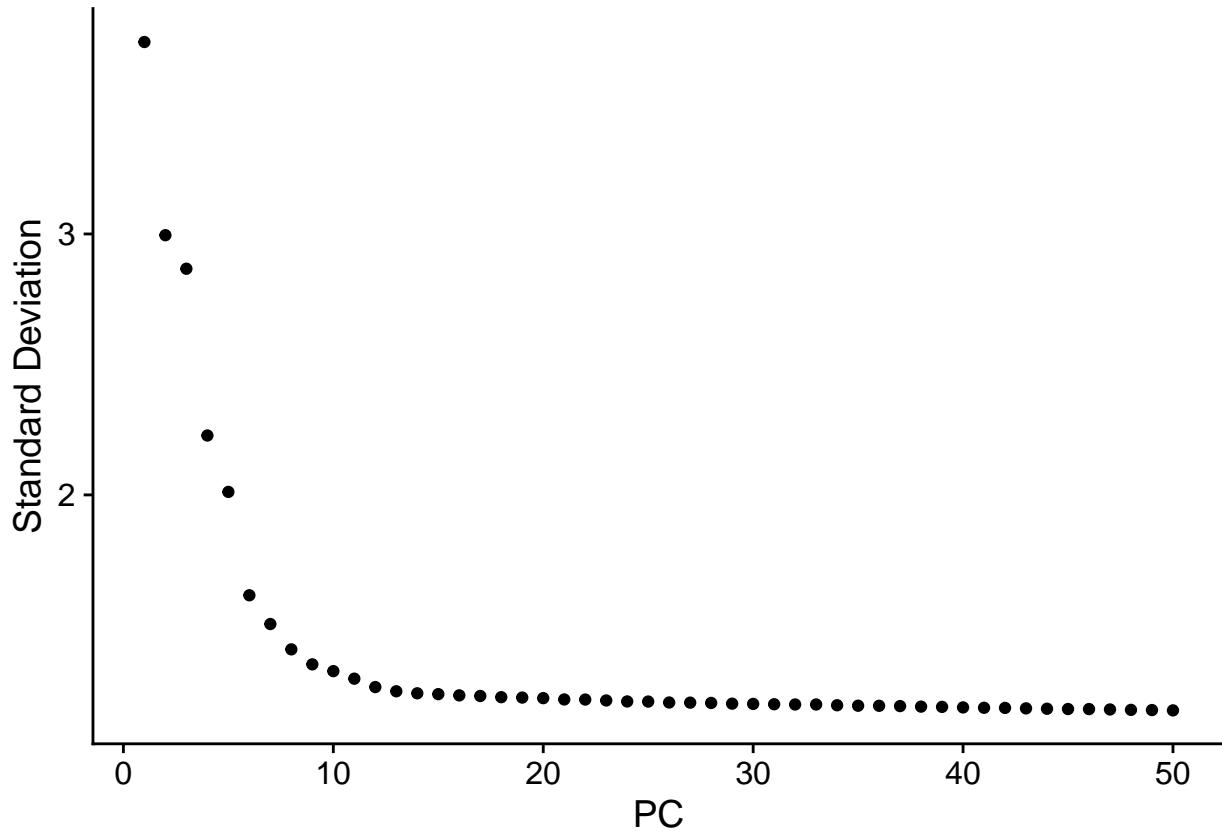
## PC_ 1
## Positive: Hmga2, H2afy, Nedd4, Sox4, Dnmt3b, Ccnd2, Adgrg1, Kit, Angpt1, Egfl7
##           Adgrl4, Cdk6, Hlf, Mpo, Slc22a3, Car2, Slco3a1, Gimap1, Spns2, Tespa1
##           Runx3, Atp2b4, Dusp5, Fgd5, Gemin5, Pcdh7, Gata2, Ablim1, Cyfip2, Prmt1
## Negative: Ctsb, Slc40a1, C1qc, C1qb, Mrcl, Lyz2, Ctss, Lpl, Selenop, Psap
##           Apoe, Hmox1, Mpeg1, Grn, Vcam1, Cybb, C1qa, Csf1r, Ctsd, Fcna
##           Cd51, Igf1, Maf, Abca1, Sdc3, Cfp, Marco, Mafb, Timd4, Igsf6
## PC_ 2
```

```

## Positive: Lyz2, F13a1, S100a8, Thbs1, Ahnak, Vim, Cybb, Hp, Stfa1, C3
##     Lgals3, S100a9, Clec7a, Mpeg1, Anxa2, Lyst, Vcan, Fcnb, Psap, Fos
##     Emilin2, Fn1, Igsf6, Mmp8, Flna, Csf2rb, Sell, Itga1, Cfh, Atf3
## Negative: Hmga2, Nedd4, Slc40a1, Ccnd2, Dnmt3b, Sox4, C1qc, Adgrg1, C1qb, Angpt1
##     Kit, Adgrl4, Cdk6, Mrc1, Egf17, Igf1, C1qa, Itga9, Slc22a3, H2afy
##     Hlf, Fcna, Car2, Vcam1, Cd51, Sdc3, Rgl1, Abca1, Fut8, Wdr43
## PC_ 3
## Positive: Slc40a1, C1qb, C1qc, Mrc1, C1qa, Fcna, Cd51, Vcam1, Sdc3, Igf1
##     Hmox1, Marco, Lpl, Rgl1, Abca1, Sema6d, Timd4, Dab2, Maf, Cdh5
##     Blvrb, Pf4, Mertk, Epb4113, Serpinb6a, Cadm1, Tgm2, Itga9, Plod1, Dmpk
## Negative: F13a1, Prtn3, Vim, Hp, Mpo, Top2a, Sell, Elane, Lyst, Stfa1
##     Klf6, Serpinb1a, Flna, Fcnb, Cybb, Igsf6, Irf8, H2afy, Lyz2, Cenpf
##     C3, Calr, S100a8, Itga1, Tifab, Csf2rb, Ahnak, Hist1h1b, Ckap4, Aspm
## PC_ 4
## Positive: S100a9, Camp, BC100530, Ngp, S100a8, Ltf, Stfa1, Lcn2, Olfm4, Gm5483
##     Syne1, Fcnb, Elane, Pglyrp1, Mpo, Gpc1, Serpinb1a, Cebpe, Cd63, Prtn3
##     C3, Itga1, Hp, Prss57, Cd177, Arsb, 4930438A08Rik, Sell, Cyfip2, Nedd4
## Negative: Clec7a, Atf3, Ctss, Fos, Csf1r, Psap, Cx3cr1, Fam129a, Cfh, Emilin2
##     Nrp1, Ahnak, Csf2rb, Hpgd, Jun, 5031439G07Rik, Rnf213, Vim, Arl4c, Mafb
##     Il1b, Mpeg1, Inf2, Lgals3, Cxcl2, Klf6, Tmem176b, Plxnb2, Dusp3, Egr1
## PC_ 5
## Positive: Mpo, Prtn3, Elane, Calr, Hspa5, Irf8, Creld2, Itga1, F13a1, Gpc1
##     Sell, Hyou1, Serpinb1a, Tifab, Zfp568, Cd63, Slc40a1, H2afy, Isyna1, Mrc1
##     Prss57, Fcna, Slc7a5, Pfkp, C1qa, Vcam1, Chaf1a, Cd51, Csf1r, Nrg2
## Negative: Thbs1, Adgrl4, Car2, Angpt1, Hlf, Adgrg1, Plk2, Vcan, Nfkbiz, S100a9
##     Slc22a3, H19, Atf3, Cxcl2, Gimap1, Chd3, Lgals3, Dusp5, Ltf, Fos
##     Ngp, Sox4, Il1b, Mmp8, Gm5483, Gcnt2, Ahnak, C3, Gata2, Arl4c

```

```
ElbowPlot(mac_diff, ndims = 50)
```



## Recluster data

```

mac_diff <- FindNeighbors(mac_diff, dims = 1:6)

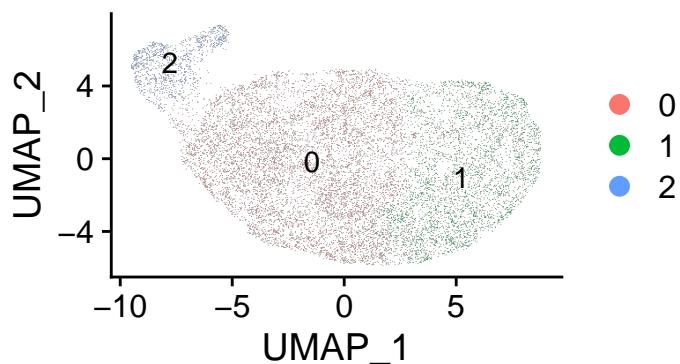
## Computing nearest neighbor graph

## Computing SNN

mac_diff <- FindClusters(mac_diff, resolution = 0.08)

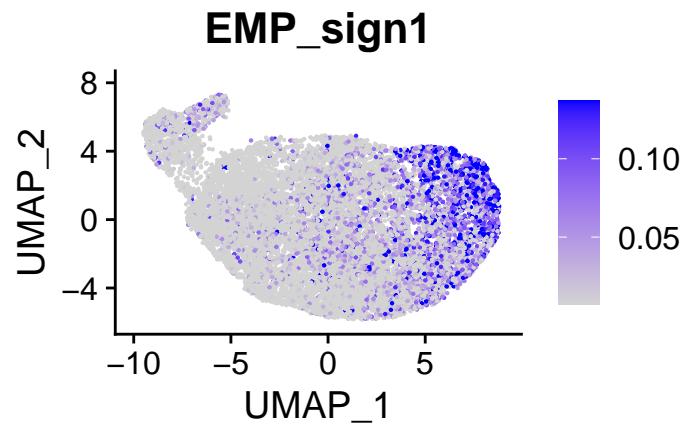
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 13815
## Number of edges: 405146
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9361
## Number of communities: 3
## Elapsed time: 0 seconds

```

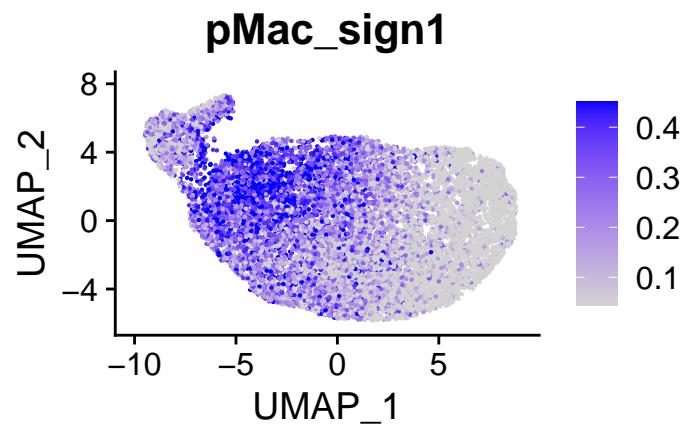


## Assigning cell type identity to clusters

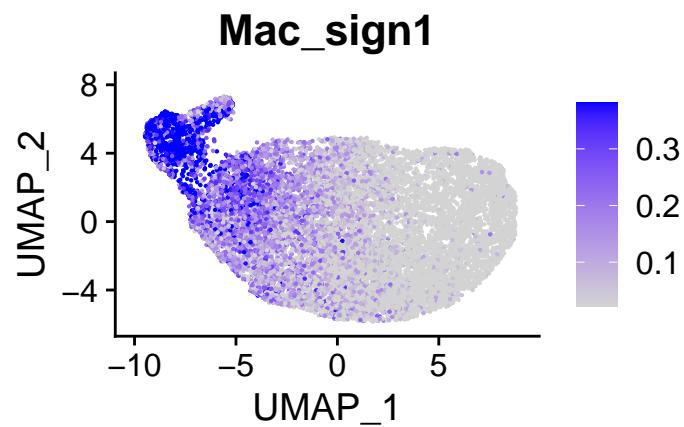
```
FeaturePlot(mac_diff, features = "EMP_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```



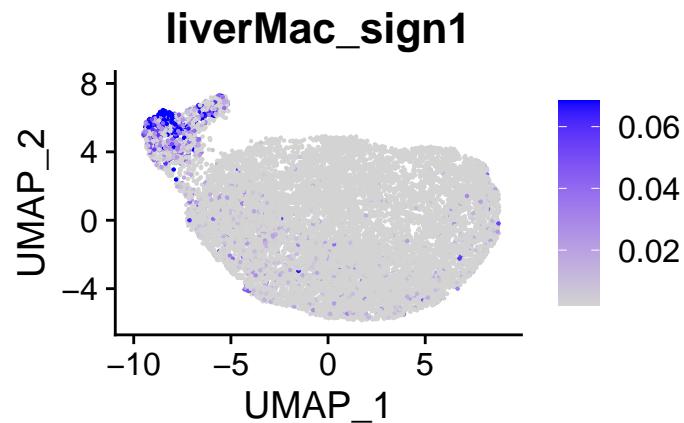
```
FeaturePlot(mac_diff, features = "pMac_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```



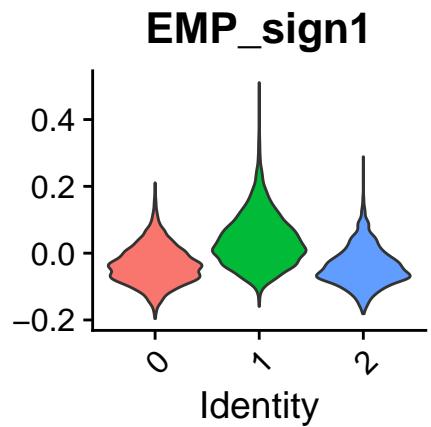
```
FeaturePlot(mac_diff, features = "Mac_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```



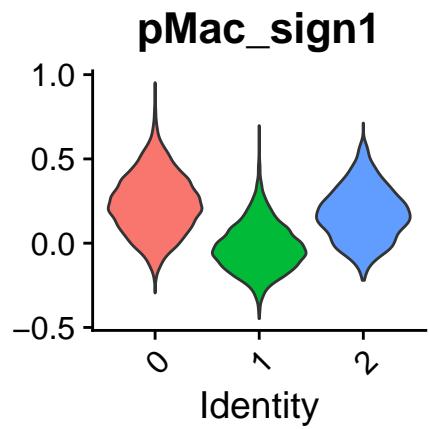
```
FeaturePlot(mac_diff, features = "liverMac_sign1", min.cutoff = "q10",
            max.cutoff = "q90")
```



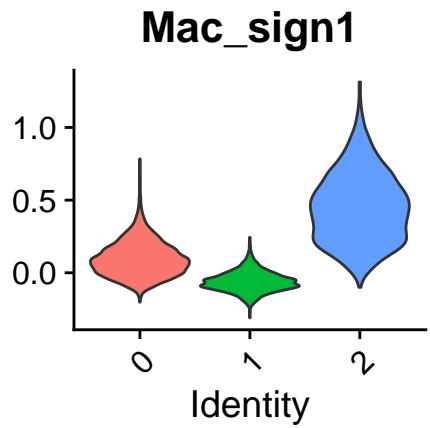
```
VlnPlot(mac_diff, features = "EMP_sign1", pt.size = 0) + NoLegend()
```



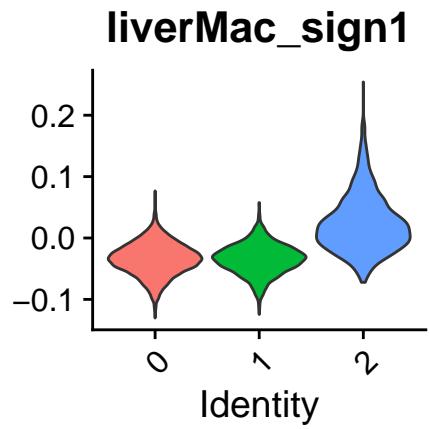
```
VlnPlot(mac_diff, features = "pMac_sign1", pt.size = 0) + NoLegend()
```



```
VlnPlot(mac_diff, features = "Mac_sign1", pt.size = 0) + NoLegend()
```



```
VlnPlot(mac_diff, features = "liverMac_sign1", pt.size = 0) +  
NoLegend()
```



```

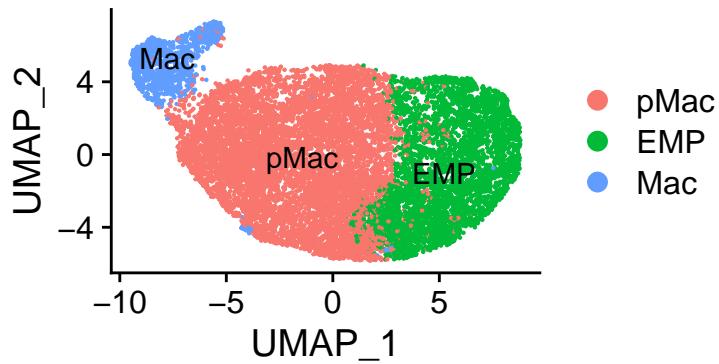
mac_diff$cell_type <- factor(Idents(mac_diff), labels = c("pMac",
  "EMP", "Mac"))

mac_diff$cell_type <- factor(mac_diff$cell_type, levels = c("pMac",
  "EMP", "Mac"))

Idents(mac_diff) <- "cell_type"

DimPlot(mac_diff, label = T) + ggtitle("")

```



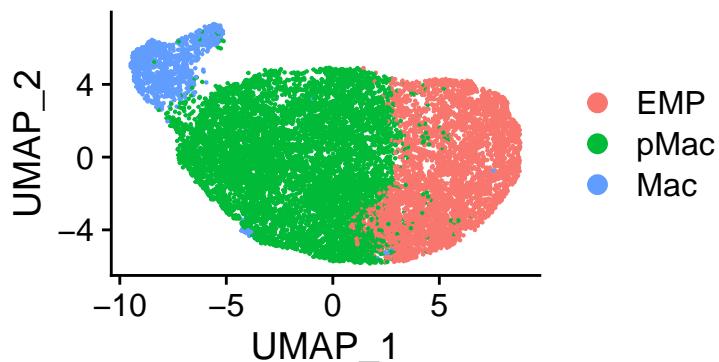
```

mac_diff$cell_type <- factor(mac_diff$cell_type, levels = c("EMP",
  "pMac", "Mac"))

Idents(mac_diff) <- "cell_type"

DimPlot(mac_diff) + ggtitle("")

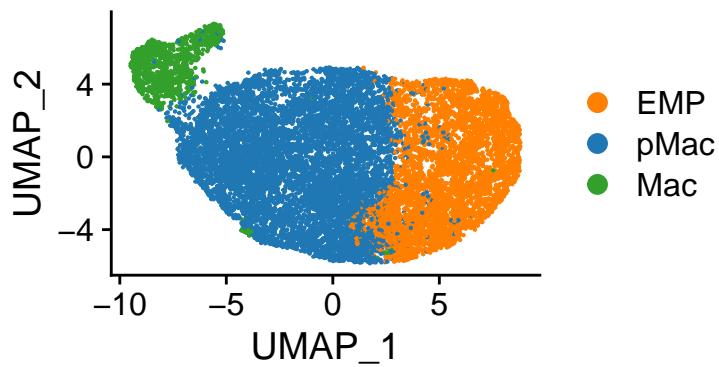
```



```

pal <- c("#ff7f00", "#1f78b4", "#33a02c")
DimPlot(mac_diff, cols = pal) + ggtitle("")

```



```
ggsave(filename = "UMAP_mac_diff.pdf", width = 4, height = 2.5)
```

## Finding differentially expressed features

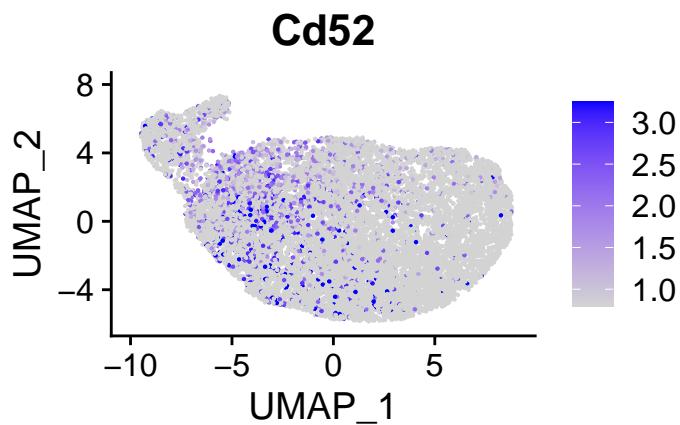
```
markers <- FindAllMarkers(mac_diff, only.pos = T)

## Calculating cluster EMP

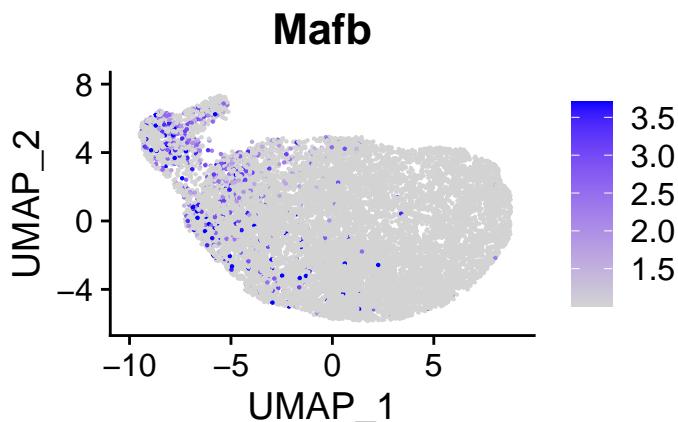
## Calculating cluster pMac

## Calculating cluster Mac

FeaturePlot(mac_diff, features = "Cd52", min.cutoff = "q10",
            max.cutoff = "q90")
```



```
FeaturePlot(mac_diff, features = "Mafb", min.cutoff = "q10",
            max.cutoff = "q90")
```



## Perform Slingshot pseudotime analysis

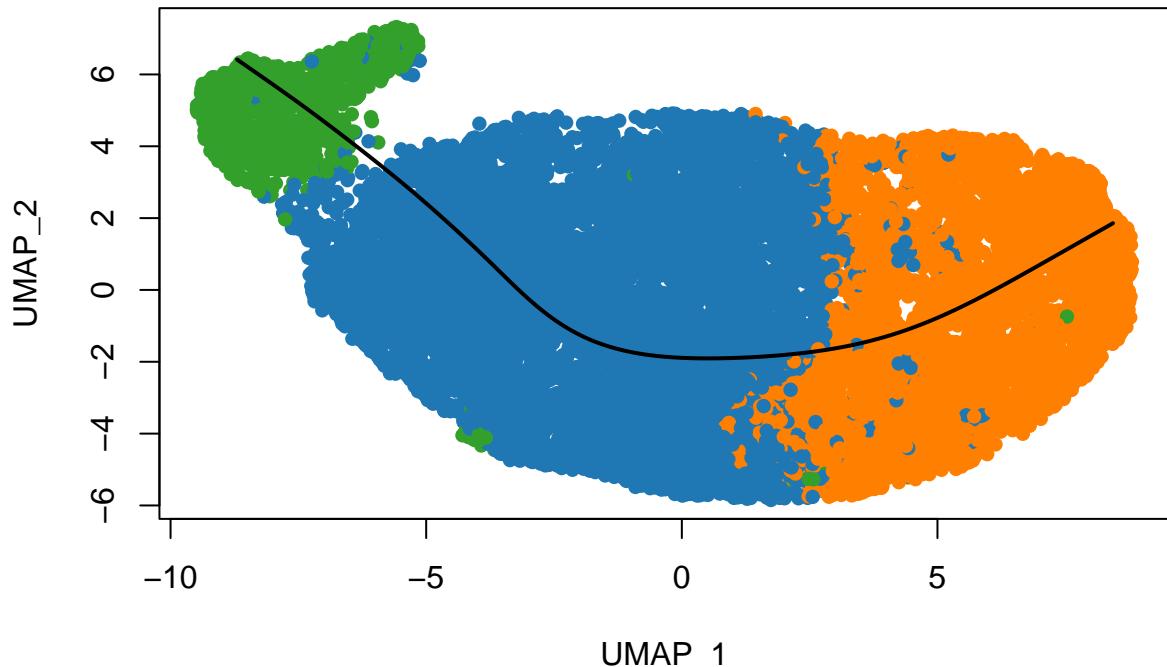
```
# Save the objects as separate matrices for input in
# slingshot
dimred <- mac_diff@reductions$umap@cell.embeddings
clustering <- mac_diff$cell_type
counts <- as.matrix(mac_diff@assays$RNA@counts[, mac_diff@assays$RNA@var.features,
      ])
# Run default Slingshot
set.seed(1)
lineages <- getLineages(data = dimred, clusterLabels = clustering,
      end.clus = c("Mac"), start.clus = "EMP")
lineages
## class: PseudotimeOrdering
## dim: 13815 1
## metadata(3): lineages mst slingParams
## pathStats(2): pseudotime weights
## cellnames(13815): 1_AAAACTGGAACT 1_AAAAGGGCAAGT ... 12_TTTTTCCCGAAC
##       12_TTTTTGCTGCCA
## cellData names(2): reducedDim clusterLabels
## pathnames(1): Lineage1
## pathData names(0):
# Define curves
curves <- as.SlingshotDataSet(getCurves(data = lineages, thresh = 0.1,
      stretch = 0.1, allow.breaks = F, approx_points = 1000))
curves
```

```

## class: SlingshotDataSet
##
##   Samples Dimensions
##       13815          2
##
##   lineages: 1
##   Lineage1: EMP  pMac  Mac
##
##   curves: 1
##   Curve1: Length: 21.863    Samples: 13815

pal <- c("#ff7f00", "#1f78b4", "#33a02c")
{
  plot(mac_diff@reductions$umap@cell.embeddings, col = pal[clustering],
    pch = 16)
  lines(curves, lwd = 2, col = "black")
}

```



```

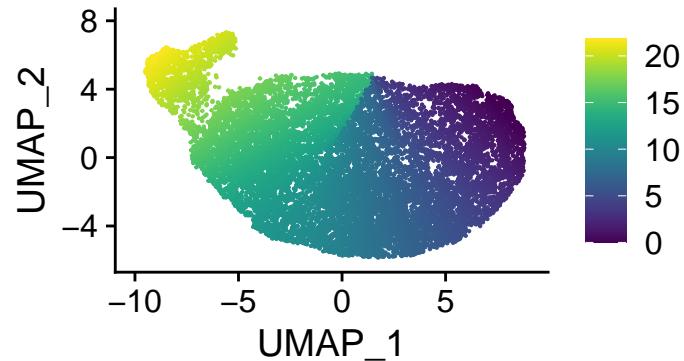
pseudotime <- slingPseudotime(curves, na = FALSE)

mac_diff$pseudotime <- pseudotime

```

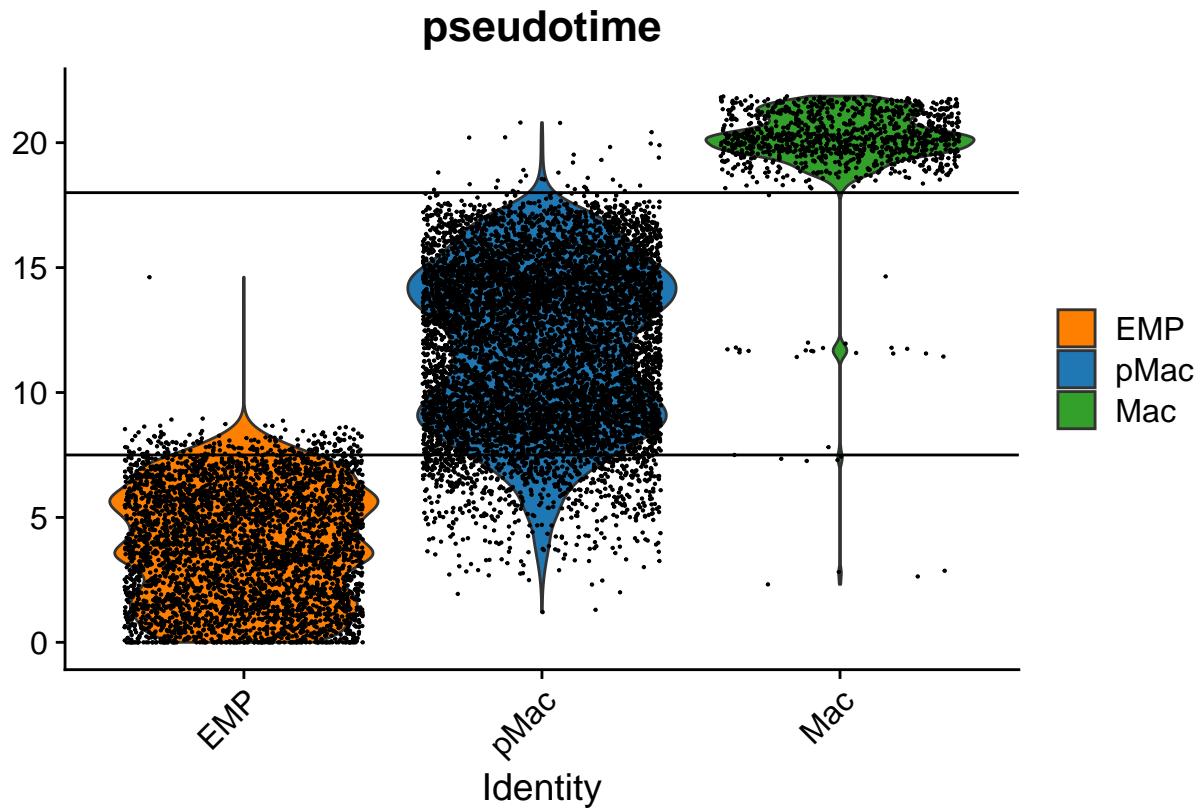
```
FeaturePlot(mac_diff, features = "pseudotime") + viridis::scale_color_viridis() +  
  ggtitle("")
```

```
## Scale for colour is already present.  
## Adding another scale for colour, which will replace the existing scale.
```



```
ggsave(filename = "UMAP_mac_diff_pseudotime.pdf", width = 3.8,  
       height = 2.5)
```

```
VlnPlot(mac_diff, features = "pseudotime", cols = pal) + geom_hline(yintercept = 7.5) +  
  geom_hline(yintercept = 18)
```

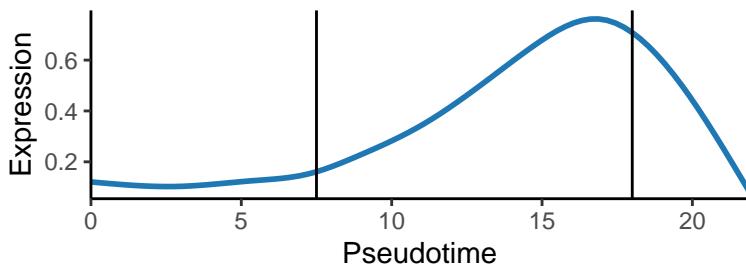


```

graph <- FetchData(mac_diff, vars = "pseudotime")
graph <- cbind(graph, FetchData(mac_diff, vars = "Cd52"))
graph <- cbind(graph, FetchData(mac_diff, vars = "Mafb"))
graph <- cbind(graph, FetchData(mac_diff, vars = "Kit"))

ggplot(graph, aes(x = pseudotime, y = Cd52)) + geom_smooth(color = "#1f78b4",
  se = F) + geom_vline(xintercept = 7.5) + geom_vline(xintercept = 18) +
  xlab("Pseudotime") + ylab("Expression") + scale_x_continuous(expand = c(0,
  0)) + theme_classic()
  
```

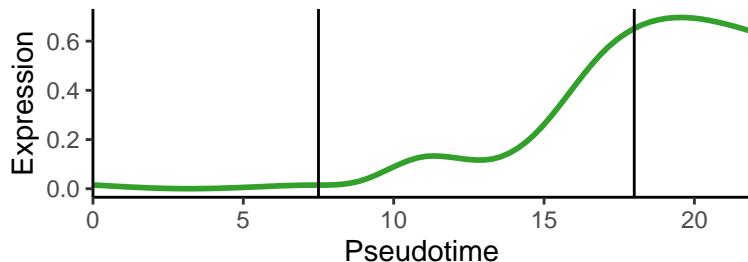
```
## 'geom_smooth()' using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
ggsave(filename = "Cd52_along_pseudotime.pdf", width = 4, height = 1.75)
```

```
ggplot(graph, aes(x = pseudotime, y = Mafb)) + geom_smooth(color = "#33a02c",
  se = F) + geom_vline(xintercept = 7.5) + geom_vline(xintercept = 18) +
  xlab("Pseudotime") + ylab("Expression") + scale_x_continuous(expand = c(0,
  0)) + theme_classic()
```

```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs = "cs")'
```



```
ggsave(filename = "Mafb_along_pseudotime.pdf", width = 4, height = 1.75)
```

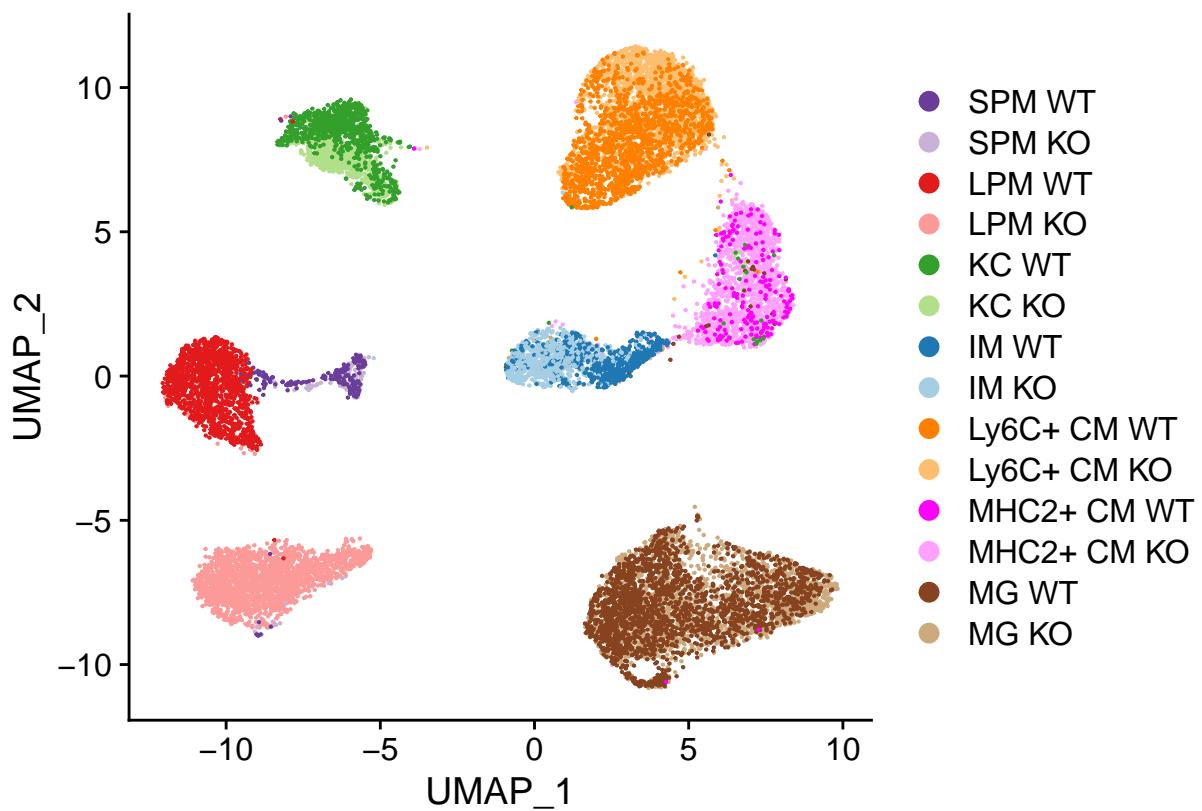
```
saveRDS(mac_diff, file = "mac_diff.rds")
```

## Cd52 expression WT vs KO

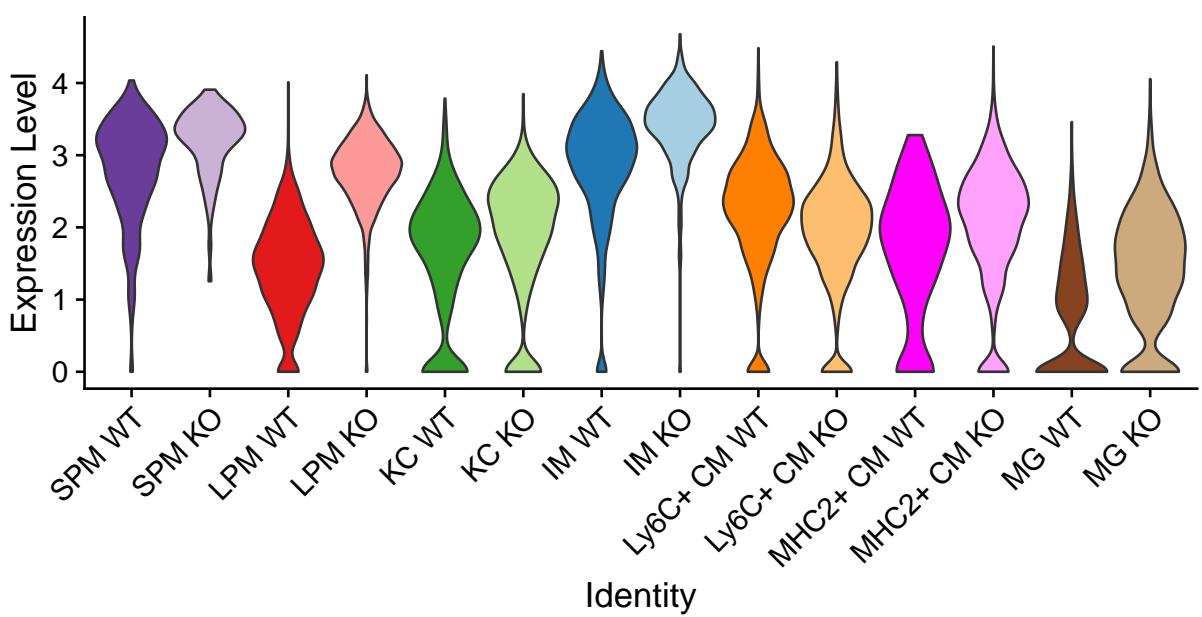
```
sc <- readRDS("sc.rds")

pal <- c("#6A3D9A", "#CAB2D6", "#E31A1C", "#FB9A99", "#33A02C",
  "#B2DF8A", "#1F78B4", "#A6CEE3", "#FF7F00", "#FDBF6F", "#FF00FA",
  "#FFA1FD", "#87421F", "#CDAA7D")

DimPlot(sc, cols = pal)
```



```
VlnPlot(sc, features = "Cd52", cols = pal, pt.size = 0) + NoLegend() +
  ggtitle("")
```



```

ggsave(filename = "VlnPlot_Cd52.pdf", width = 7, height = 4)

sessionInfo()

## R version 4.4.2 (2024-10-31 ucrt)
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_Belgium.utf8  LC_CTYPE=English_Belgium.utf8
## [3] LC_MONETARY=English_Belgium.utf8 LC_NUMERIC=C
## [5] LC_TIME=English_Belgium.utf8
##
## time zone: Europe/Brussels
## tzcode source: internal
##
## attached base packages:
## [1] stats4      stats       graphics    grDevices   utils       datasets    methods
## [8] base
##
## other attached packages:
##  [1] ggplot2_3.5.1           DelayedMatrixStats_1.28.1
##  [3] DelayedArray_0.32.0      SparseArray_1.6.1
##  [5] S4Arrays_1.6.0          abind_1.4-8
##  [7] Matrix_1.7-1            slingshot_2.14.0
##  [9] TrajectoryUtils_1.14.0   SingleCellExperiment_1.28.1
## [11] SummarizedExperiment_1.36.0 Biobase_2.66.0
## [13] GenomicRanges_1.58.0     GenomeInfoDb_1.42.3
## [15] IRanges_2.40.1          S4Vectors_0.44.0
## [17] BiocGenerics_0.52.0     MatrixGenerics_1.18.1
## [19] matrixStats_1.4.1        prcurve_2.1.6
## [21] readxl_1.4.3            dplyr_1.1.4
## [23] Seurat_4.3.0             SeuratObject_4.1.3
## [25] sp_2.1-4
##
## loaded via a namespace (and not attached):
##  [1] RColorBrewer_1.1-3      rstudioapi_0.17.1      jsonlite_1.8.9
##  [4] magrittr_2.0.3          ggbeeswarm_0.7.2       spatstat.utils_3.1-1
##  [7] farver_2.1.2            rmarkdown_2.29         zlibbioc_1.52.0
## [10] vctrs_0.6.5             ROCR_1.0-11           spatstat.explore_3.3-3
## [13] tinytex_0.54            hmltools_0.5.8.1      cellranger_1.1.0
## [16] sctransform_0.4.1       parallelly_1.41.0     KernSmooth_2.23-24
## [19] htmlwidgets_1.6.4        ica_1.0-3              plyr_1.8.9
## [22] plotly_4.10.4           zoo_1.8-12            igraph_2.1.2
## [25] mime_0.12               lifecycle_1.0.4       pkgconfig_2.0.3
## [28] R6_2.5.1                fastmap_1.2.0          GenomeInfoDbData_1.2.13
## [31] fitdistrplus_1.2-1      future_1.34.0         shiny_1.10.0
## [34] digest_0.6.37           colorspace_2.1-1       patchwork_1.3.0
## [37] tensor_1.5               irlba_2.3.5.1          labeling_0.4.3
## [40] progressr_0.15.1         spatstat.sparse_3.1-0 mgcv_1.9-1

```

```

## [43] httr_1.4.7
## [46] withr_3.0.2
## [49] MASS_7.3-61
## [52] lmtest_0.9-40
## [55] future.apply_1.11.3
## [58] glue_1.8.0
## [61] grid_4.4.2
## [64] reshape2_1.4.4
## [67] spatstat.data_3.1-4
## [70] data.table_1.16.4
## [73] RcppAnnoy_0.0.22
## [76] pillar_1.10.0
## [79] later_1.4.1
## [82] survival_3.7-0
## [85] miniUI_0.1.1.1
## [88] gridExtra_2.3
## [91] statmod_1.5.0
## [94] lazyeval_0.2.2
## [97] codetools_0.2-20
## [100] uwot_0.2.2
## [103] munsell_0.5.1
## [106] spatstat.random_3.3-2
## [109] spatstat.univar_3.1-1
## [112] listenv_0.9.1
## [115] ggridges_0.5.6
## [118] purrr_1.0.2
## [121] formatR_1.14

polyclip_1.10-7
viridis_0.6.5
tools_4.4.2
beeswarm_0.4.0
goftest_1.2-3
nlme_3.1-166
Rtsne_0.17
generics_0.1.3
R.methodsS3_1.8.2
XVector_0.46.0
ggrepel_0.9.6
stringr_1.5.1
splines_4.4.2
deldir_2.0-4
pbapply_1.7-2
scattermore_1.2
stringi_1.8.4
yaml_2.3.10
tibble_3.2.1
xtable_1.8-4
Rcpp_1.0.13-1
png_0.1-8
parallel_4.4.2
viridisLite_0.4.2
crayon_1.5.3
rlang_1.1.4

compiler_4.4.2
R.utils_2.12.3
viper_0.4.7
httpuv_1.6.15
R.oo_1.27.0
promises_1.3.2
cluster_2.1.6
gttable_0.3.6
tidyverse_1.3.1
spatstat.geom_3.3-4
RANN_2.6.2
limma_3.62.2
lattice_0.22-6
tidyselect_1.2.1
knitr_1.49
xfun_0.49
UCSC.utils_1.2.0
evaluate_1.0.1
cli_3.6.3
reticulate_1.40.0
globals_0.16.3
ggrastr_1.0.2
sparseMatrixStats_1.18.0
scales_1.3.0
leiden_0.4.3.1
cowplot_1.1.3

```