TK_54: Analysis of scRNAseq data from mouse adipose tissue (dataset GSE157281)

Hernan Lorenzi

2023-05-08

Goal: to utilize a published scRNA seq dataset (GSE157281) to extract information about the expression of some genes which are relevant to Asmita's project, mainly the purinergic receptors expression (P2Y, P2X and adenosine family receptors) in mouse.

Load required libraries

```
library(Seurat)
library(tidyverse)
library(cowplot)
library(ggplot2)
library(plot3D)
```

Load data from manuscript

Manuscript reference data source

```
# 1. read raw counts ------
counts_magic <- read.delim('./data/GSE157281_HFD.LFD.magic.txt', header = T, sep = ' ')
counts_raw <- read.delim('./data/GSE157281_HFD.LFD.txt', header = T, sep = ' ')
# ___create Seurat Object with count data --------
# include only genes that are are expressed in 3 or more cells and cells with complexity of 200 genes o
gse <- CreateSeuratObject(counts = counts_raw, project = "GSE157281", min.cells = 3, min.features = 200)
str(gse)
```

```
## Formal class 'Seurat' [package "SeuratObject"] with 13 slots
##   ..@ assays      :List of 1
##   .. ..$ RNA:Formal class 'Assay' [package "SeuratObject"] with 8 slots
##   .. .. .. .. ..@ counts       :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   .. .. .. .. .. .. ..@ i       : int [1:487958] 1 3 12 16 24 31 36 41 48 55 ...
##   .. .. .. .. .. .. ..@ p       : int [1:455] 0 1071 2054 3912 5433 6438 7725 8619 11006 12974 ...
##   .. .. .. .. .. .. ..@ Dim     : int [1:2] 9557 454
##   .. .. .. .. .. .. ..@ Dimnames:List of 2
##   .. .. .. .. .. .. .. ..$ : chr [1:9557] "Mrpl15" "Lypla1" "Tcea1" "Atp6v1h" ...
##   .. .. .. .. .. .. .. ..$ : chr [1:454] "LFD1_Org_CGAACCGATCGT" "LFD1_Org_AATATTGAAAGC" "LFD1_Org_GCCT
##   .. .. .. .. .. .. ..@ x       : num [1:487958] 2.57 1.61 1.61 1.61 1.61 ...
##   .. .. .. .. .. .. ..@ factors : list()
```

```
##   .. .. .. ..@ data         :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   .. .. .. .. .. ..@ i       : int [1:487958] 1 3 12 16 24 31 36 41 48 55 ...
##   .. .. .. .. .. ..@ p       : int [1:455] 0 1071 2054 3912 5433 6438 7725 8619 11006 12974 ...
##   .. .. .. .. .. ..@ Dim     : int [1:2] 9557 454
##   .. .. .. .. .. ..@ Dimnames:List of 2
##   .. .. .. .. .. .. ..$ : chr [1:9557] "Mrpl15" "Lypla1" "Tcea1" "Atp6v1h" ...
##   .. .. .. .. .. .. ..$ : chr [1:454] "LFD1_Org_CGAACCGATCGT" "LFD1_Org_AATATTGAAAGC" "LFD1_Org_GCCTC
##   .. .. .. .. .. ..@ x       : num [1:487958] 2.57 1.61 1.61 1.61 1.61 ...
##   .. .. .. .. .. ..@ factors : list()
##   .. .. .. ..@ scale.data   : num[0 , 0 ]
##   .. .. .. ..@ key          : chr "rna_"
##   .. .. .. ..@ assay.orig   : NULL
##   .. .. .. ..@ var.features : logi(0)
##   .. .. .. ..@ meta.features:'data.frame':   9557 obs. of  0 variables
##   .. .. .. ..@ misc         : list()
##   ..@ meta.data   :'data.frame': 454 obs. of  3 variables:
##   .. ..$ orig.ident  : Factor w/ 5 levels "HFD1","HFD2",..: 4 4 4 4 4 4 4 4 4 4 ...
##   .. ..$ nCount_RNA  : num [1:454] 2071 1793 2324 2328 1962 ...
##   .. ..$ nFeature_RNA: int [1:454] 1071 983 1858 1521 1005 1287 894 2387 1968 1055 ...
##   ..@ active.assay: chr "RNA"
##   ..@ active.ident: Factor w/ 5 levels "HFD1","HFD2",..: 4 4 4 4 4 4 4 4 4 4 ...
##   .. ..- attr(*, "names")= chr [1:454] "LFD1_Org_CGAACCGATCGT" "LFD1_Org_AATATTGAAAGC" "LFD1_Org_GCC
##   ..@ graphs      : list()
##   ..@ neighbors   : list()
##   ..@ reductions  : list()
##   ..@ images      : list()
##   ..@ project.name: chr "GSE157281"
##   ..@ misc        : list()
##   ..@ version     :Classes 'package_version', 'numeric_version'  hidden list of 1
##   .. ..$ : int [1:3] 4 1 3
##   ..@ commands    : list()
##   ..@ tools       : list()
```

```
# count matrix
gse@assays$RNA@counts[1:10,1:10]
```

```
## 10 x 10 sparse Matrix of class "dgCMatrix"

##   [[ suppressing 10 column names 'LFD1_Org_CGAACCGATCGT', 'LFD1_Org_AATATTGAAAGC', 'LFD1_Org_GCCTGTC

##
## Mrpl15  .        1.429502 1.305719 1.159301 .       .       .        1.1280565
## Lypla1  2.570134 1.429502 .        1.681835 .       .       .        1.1280565
## Tcea1   .        .        .        .        .       1.30298 .        .
## Atp6v1h 1.613933 .        .        .        1.61781 .       .        .
## Rb1cc1  .        .        .        1.159301 .       .       1.672813 .
## Pcmtd1  .        .        1.305719 1.159301 .       .       1.672813 1.1280565
## Rrs1    .        .        .        .        .       .       .        .
## Adhfe1  .        .        .        .        .       .       .        0.7153112
## Vcpip1  .        .        .        .        .       .       .        0.7153112
## Sgk3    .        .        .        .        .       .       .        .
##
## Mrpl15  .         1.973496
```

```
## Lypla1  1.3549214 .
## Tcea1   .         .
## Atp6v1h .         .
## Rb1cc1  0.8912716 1.410473
## Pcmtd1  1.3549214 .
## Rrs1    .         .
## Adhfe1  0.8912716 .
## Vcpip1  .         .
## Sgk3    .         .
```

```
# 2. QC --------
gse[["percent.mt"]] <- PercentageFeatureSet(gse, pattern = "^mt-")
str(gse)
```

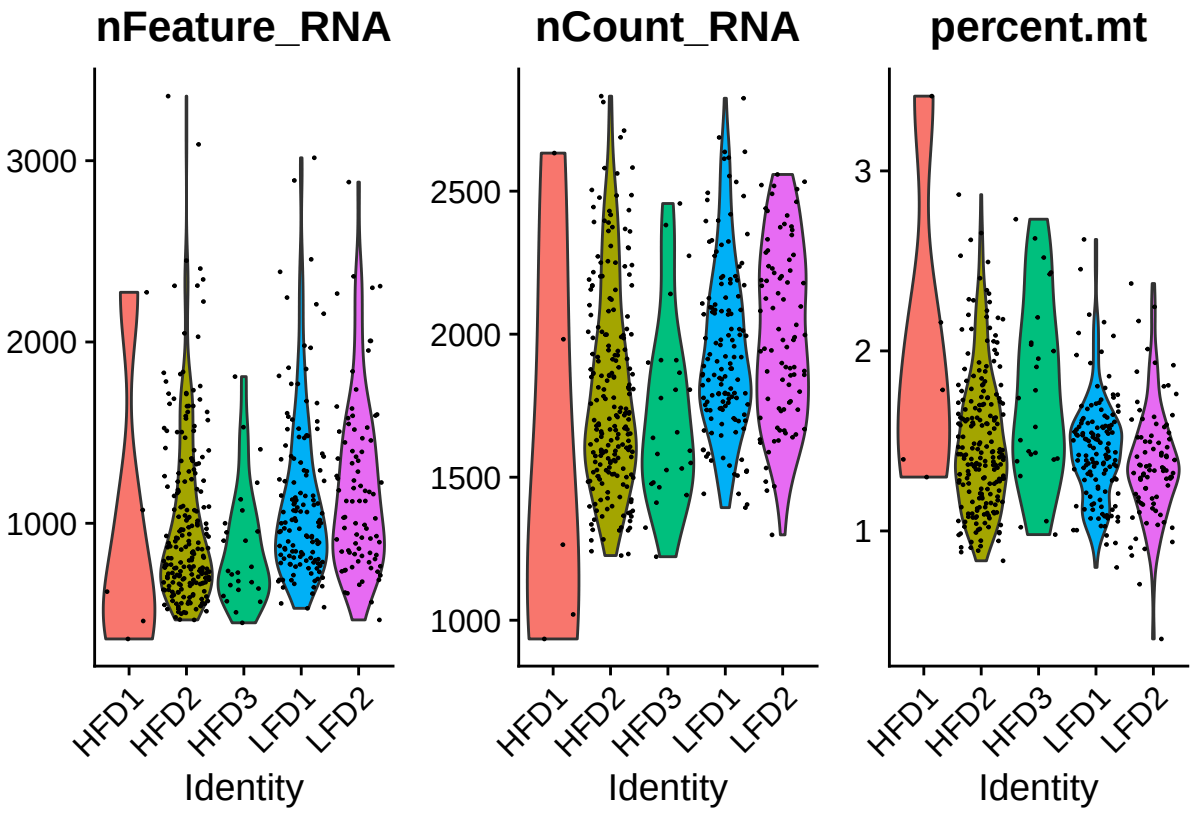
```
## Formal class 'Seurat' [package "SeuratObject"] with 13 slots
##   ..@ assays      :List of 1
##   .. ..$ RNA:Formal class 'Assay' [package "SeuratObject"] with 8 slots
##   .. .. .. .. ..@ counts       :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   .. .. .. .. .. .. ..@ i       : int [1:487958] 1 3 12 16 24 31 36 41 48 55 ...
##   .. .. .. .. .. .. ..@ p       : int [1:455] 0 1071 2054 3912 5433 6438 7725 8619 11006 12974 ...
##   .. .. .. .. .. .. ..@ Dim     : int [1:2] 9557 454
##   .. .. .. .. .. .. ..@ Dimnames:List of 2
##   .. .. .. .. .. .. .. ..$ : chr [1:9557] "Mrpl15" "Lypla1" "Tcea1" "Atp6v1h" ...
##   .. .. .. .. .. .. .. ..$ : chr [1:454] "LFD1_Org_CGAACCGATCGT" "LFD1_Org_AATATTGAAAGC" "LFD1_Org_GCCT
##   .. .. .. .. .. .. ..@ x       : num [1:487958] 2.57 1.61 1.61 1.61 1.61 ...
##   .. .. .. .. .. .. ..@ factors : list()
##   .. .. .. ..@ data         :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
##   .. .. .. .. .. .. ..@ i       : int [1:487958] 1 3 12 16 24 31 36 41 48 55 ...
##   .. .. .. .. .. .. ..@ p       : int [1:455] 0 1071 2054 3912 5433 6438 7725 8619 11006 12974 ...
##   .. .. .. .. .. .. ..@ Dim     : int [1:2] 9557 454
##   .. .. .. .. .. .. ..@ Dimnames:List of 2
##   .. .. .. .. .. .. .. ..$ : chr [1:9557] "Mrpl15" "Lypla1" "Tcea1" "Atp6v1h" ...
##   .. .. .. .. .. .. .. ..$ : chr [1:454] "LFD1_Org_CGAACCGATCGT" "LFD1_Org_AATATTGAAAGC" "LFD1_Org_GCCT
##   .. .. .. .. .. .. ..@ x       : num [1:487958] 2.57 1.61 1.61 1.61 1.61 ...
##   .. .. .. .. .. .. ..@ factors : list()
##   .. .. .. ..@ scale.data   : num[0 , 0 ]
##   .. .. .. ..@ key          : chr "rna_"
##   .. .. .. ..@ assay.orig   : NULL
##   .. .. .. ..@ var.features : logi(0)
##   .. .. .. ..@ meta.features:'data.frame':   9557 obs. of  0 variables
##   .. .. .. ..@ misc         : list()
##   ..@ meta.data   :'data.frame': 454 obs. of  4 variables:
##   .. ..$ orig.ident  : Factor w/ 5 levels "HFD1","HFD2",..: 4 4 4 4 4 4 4 4 4 4 ...
##   .. ..$ nCount_RNA  : num [1:454] 2071 1793 2324 2328 1962 ...
##   .. ..$ nFeature_RNA: int [1:454] 1071 983 1858 1521 1005 1287 894 2387 1968 1055 ...
##   .. ..$ percent.mt  : num [1:454] 1.05 1.08 1.37 1.08 1.39 ...
##   ..@ active.assay: chr "RNA"
##   ..@ active.ident: Factor w/ 5 levels "HFD1","HFD2",..: 4 4 4 4 4 4 4 4 4 4 ...
##   .. ..- attr(*, "names")= chr [1:454] "LFD1_Org_CGAACCGATCGT" "LFD1_Org_AATATTGAAAGC" "LFD1_Org_GCC
##   ..@ graphs      : list()
##   ..@ neighbors   : list()
##   ..@ reductions  : list()
##   ..@ images      : list()
##   ..@ project.name: chr "GSE157281"
```

```
##   ..@ misc        : list()
##   ..@ version     :Classes 'package_version', 'numeric_version'  hidden list of 1
##   .. ..$ : int [1:3] 4 1 3
##   ..@ commands    : list()
##   ..@ tools       : list()
```

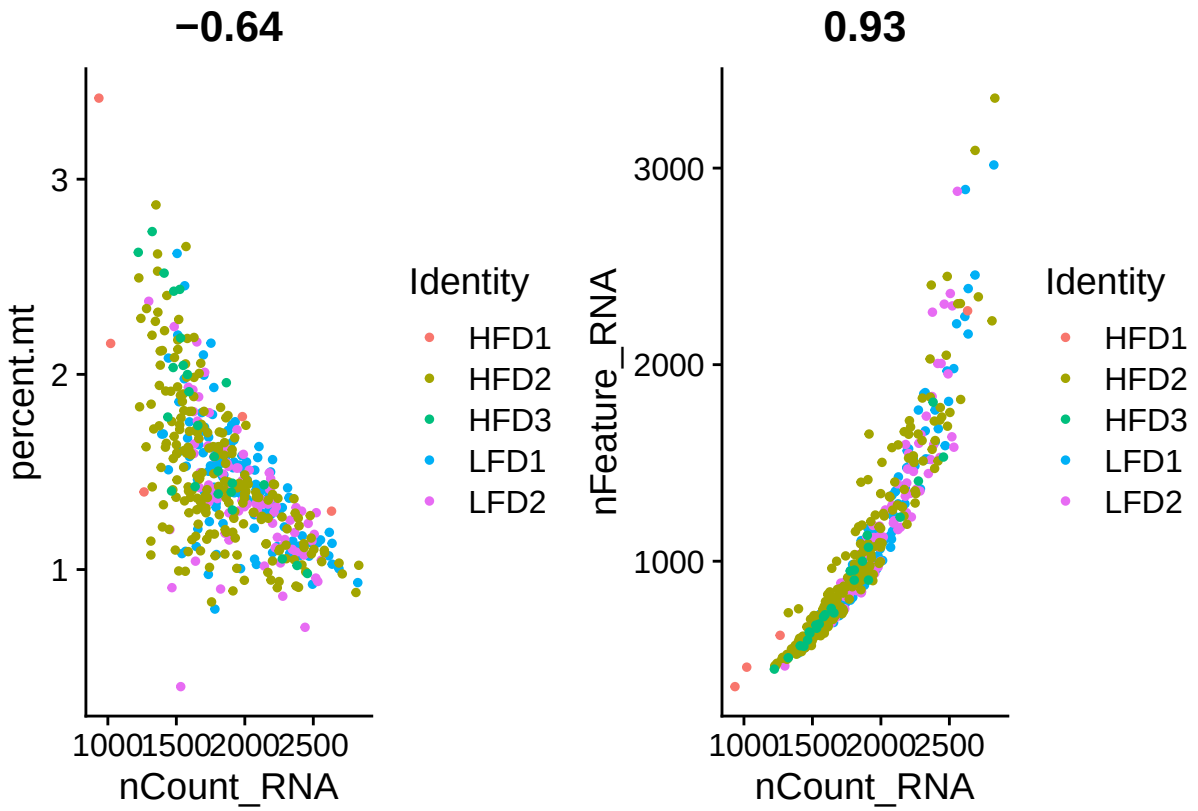
```
# Show QC metrics for the first 5 cells
head(gse@meta.data, 5)
```

```
##                       orig.ident nCount_RNA nFeature_RNA percent.mt
## LFD1_Org_CGAACCGATCGT       LFD1   2071.046         1071   1.053280
## LFD1_Org_AATATTGAAAGC       LFD1   1793.102          983   1.076999
## LFD1_Org_GCCTGTCAGAGG       LFD1   2323.641         1858   1.365507
## LFD1_Org_TGCAGTTACTGA       LFD1   2328.142         1521   1.082970
## LFD1_Org_AATAATTTAGTG       LFD1   1962.231         1005   1.392017
```

```
# Add treatment groups to metadata
gse@meta.data['groups'] <- stringr::str_remove_all(string = gse@meta.data$orig.ident, pattern = "1|2|3")


# We filter cells that have unique feature counts over 2,500 or less than 200
# We filter cells that have >5% mitochondrial counts

# ___Visualize QC metrics as a violin plot -------
VlnPlot(gse, features = c("nFeature_RNA", "nCount_RNA", "percent.mt"), ncol = 3)
```



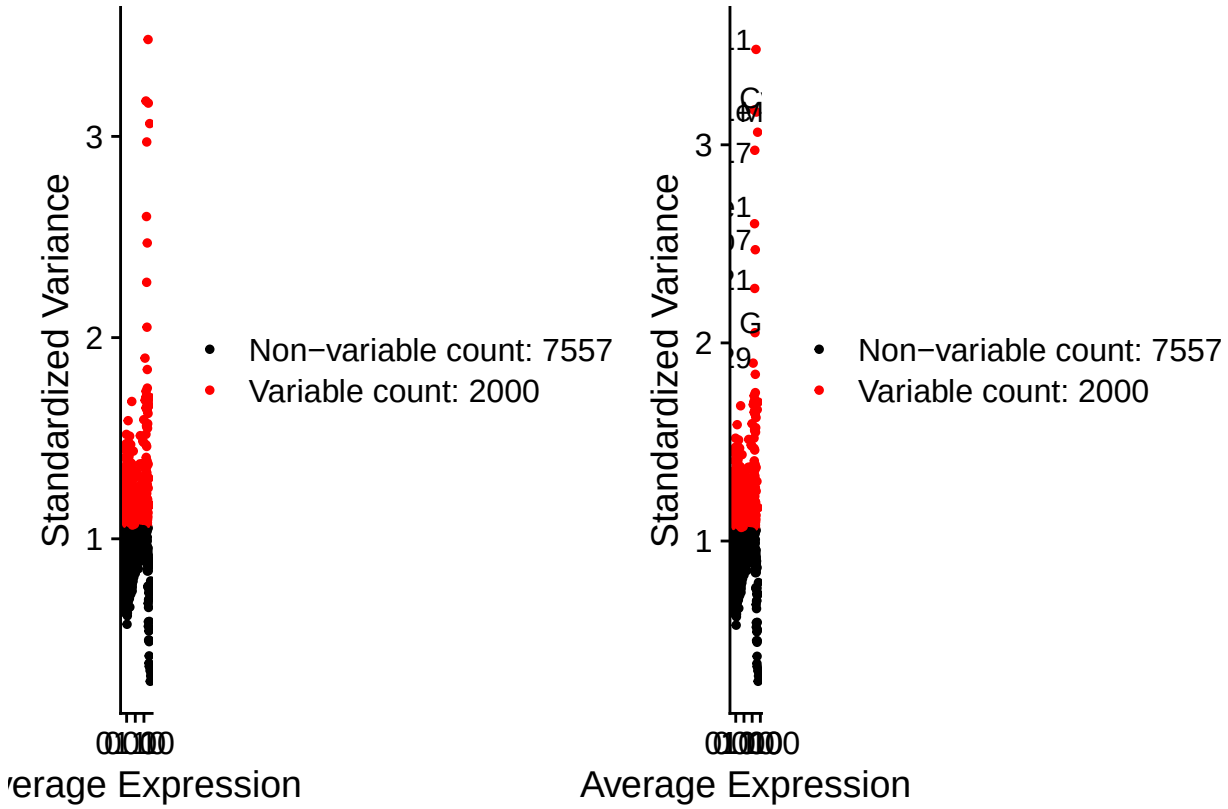
```
# ___feature-feature or gene-gene relationship --------
plot1 <- FeatureScatter(gse, feature1 = "nCount_RNA", feature2 = "percent.mt")
plot2 <- FeatureScatter(gse, feature1 = "nCount_RNA", feature2 = "nFeature_RNA")
plot1 + plot2
```




```
# what does plot 1 show/How to interpret plot1? what does gene-gene relationship mean?

gse <- subset(gse, subset = nFeature_RNA > 200 & nFeature_RNA < 2500 & percent.mt < 5)
# left with 987/1176 cells


# 3. Normalization ----------
gse <- NormalizeData(gse, normalization.method = "LogNormalize", scale.factor = 10000)
#str(gse)

# ___identification of highly variable features ---------
gse <- FindVariableFeatures(gse, selection.method = "vst", nfeatures = 2000)

# Identify the 10 most highly variable genes
top10 <- head(VariableFeatures(gse), 10)

# ___plot variable features with and without labels ---------
plot3 <- VariableFeaturePlot(gse)
plot4 <- LabelPoints(plot = plot3, points = top10, repel = TRUE)
```

```
## When using repel, set xnudge and ynudge to 0 for optimal results
```

```
plot3 + plot4
```




```
# 4. scaling the data (performed prior to linear dim reduction) ---------
all.genes <- rownames(gse)
gse <- ScaleData(gse, features = all.genes)
```

```
## Centering and scaling data matrix
```

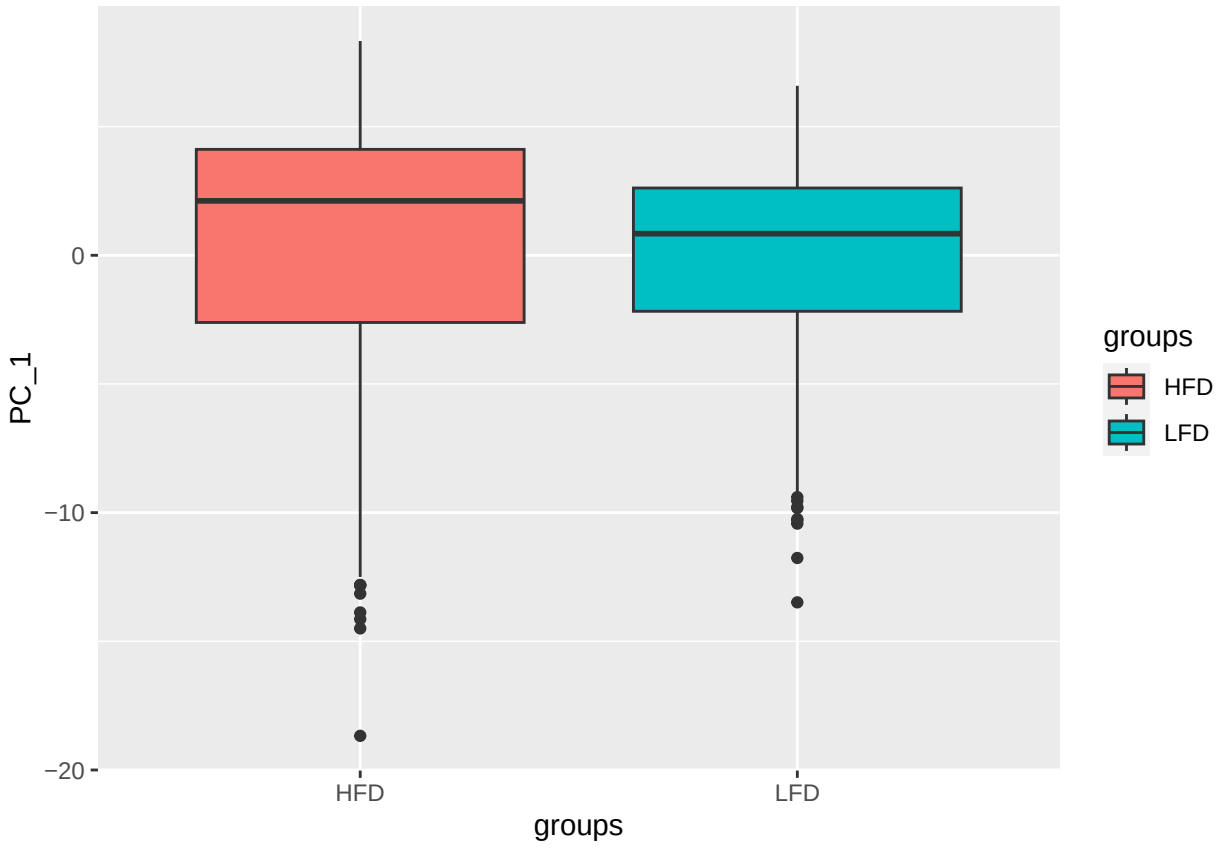
```
#str(gse)


# 5. Linear Dimensionality Reduction ----------
#gse <- RunPCA(gse, features = VariableFeatures(object = gse))
gse <- RunPCA(gse, features = VariableFeatures(object = gse))
```

```
## PC_ 1
## Positive:  Mup20, Cyp2f2, Mup3, Hsd17b13, Arg1, Fbp1, Hrsp12, Cps1, Pigr, Sds
##     mt-Cytb, Amdhd1, Hal, Hpx, Serpina1e, Ass1, Pck1, Gstp1, mt-Nd4, Mug2
##     Serpina12, Slc3a1, Hsd17b6, mt-Rnr2, mt-Co1, Tat, Angptl3, Cp, mt-Nd5, Aldh1b1
## Negative:  Rgn, Cyp2c50, Gulo, Cyp2e1, Cyp2c29, Cyp1a2, Cyp2c37, Aldh3a2, Oat, Nr1i3
##     Cyp2a5, Lect2, Mup17, Gm13775, Slc22a1, Fitm1, Csad, Lhpp, Clstn3, Cyp4a10
##     Cyp7a1, Aldh1a1, Slco1b2, Acaa1b, Slc1a2, Vnn1, 1810058I24Rik, Rnase4, Gsta3, Aldh2
## PC_ 2
## Positive:  Cyp3a11, Selenbp2, Cyp2c70, Mup7, Nudt7, Ces2a, Orm1, Mup16, Gstp1, Mup1
##     Upp2, C8b, Mup11, Cyp7b1, Mup12, Egfr, Inmt, Mup17, Urah, Rps14
##     Aldh1l1, Apom, Rnase4, Hsd3b5, Cyp2d9, Cyb5r3, C9, Serpina1e, Mdh1, Slc38a3
```

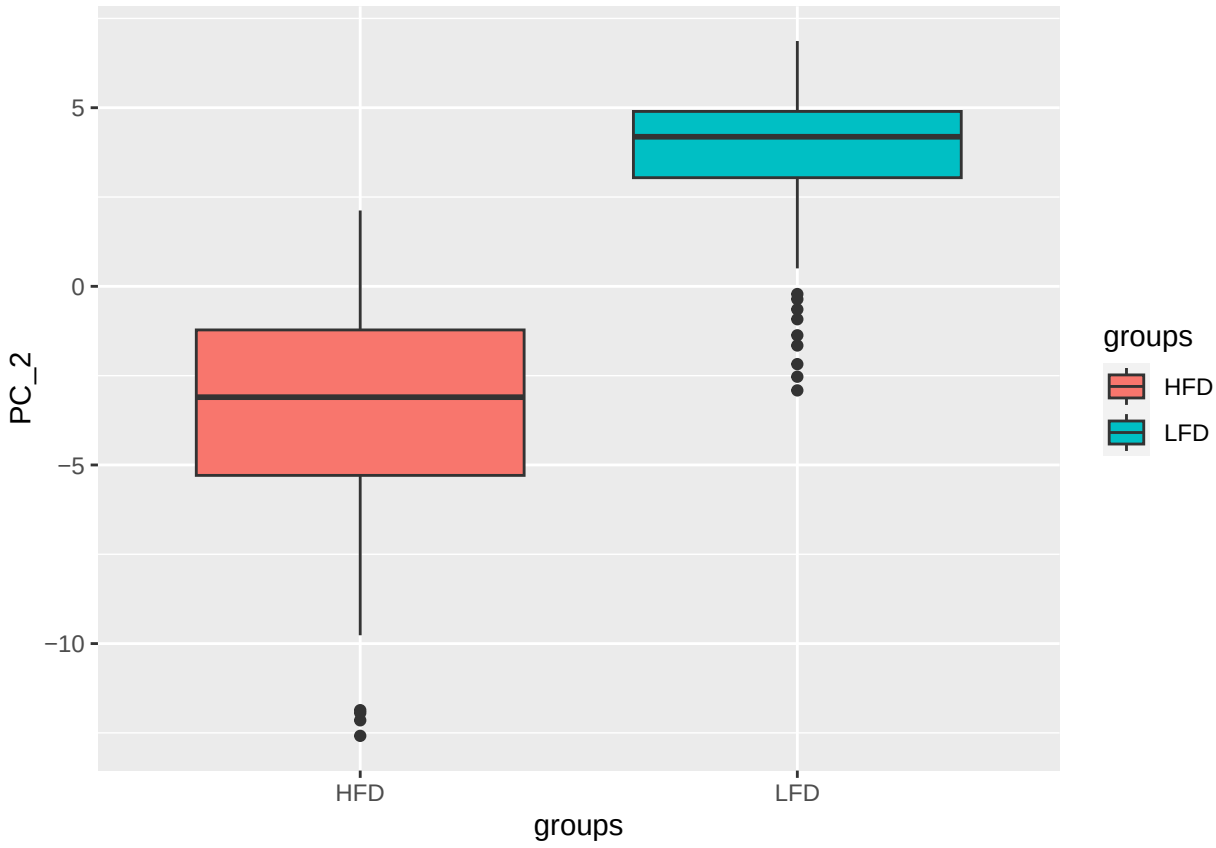
```
## Negative:  mt-Rnr2, Mup3, Apoa4, Malat1, Cyp4a14, Gm6484, G0s2, Elovl5, Zfp36l1, Dusp1
##     Csad, Jun, Abcb4, mt-Rnr1, Plxna2, Fasn, Gpc1, Leap2, Hes1, Tsc22d1
##     Mgst3, Vnn1, Hbb-bs, Kdr, Klf10, Pex11a, Net1, Nr1i3, Rrp36, Crot
## PC_ 3
## Positive:  Mup3, mt-Rnr2, Cyb5, Hrsp12, Cyp2e1, Serpina1e, Mup17, Mup20, Akr1c6, Tmsb4x
##     Hamp, Hbb-bs, Glul, Mup1, Mup7, Scd1, Mup9, Rgn, Slco1b2, Mrc1
##     Cyp2c29, Oat, Rps27l, Mup11, Actl6a, Calm1, Exoc2, 1100001G20Rik, Usp50, Fitm1
## Negative:  Klf7, F3, Ppp4r4, Scarna2, Gpn2, D1Ertd622e, Micu3, Plgrkt, Gbp6, Gbp10
##     Tia1, Rsad2, Frk, Fbxl14, Galns, Acot9, Otud1, Cc2d1b, Rbm14, Ifit3
##     Sardh, Ythdc2, Oasl1, Ifit1, Abl1, Ankrd54, Snx21, Ccdc84, Cecr2, Rgs12
## PC_ 4
## Positive:  mt-Rnr2, Mup3, Cyp3a11, Ifit1, Selenbp2, Ifit3, Otud1, Cxcl9, Ankrd54, Isg15
##     Mup17, Gin1, Cc2d1b, F3, Oat, Acot9, Ppp4r4, Gbp10, Gbp6, Oasl1
##     Tpst1, Ythdc2, Rsad2, Galns, Fam46a, Cecr2, Gpn2, Fbxl14, Rbm14, Abl1
## Negative:  Itih4, Ldha, Ndufs6, Fabp2, Atp5f1, Apoa4, Slc27a2, Slc17a8, Vcp, Ndufc1
##     Psmc1, Elovl5, G0s2, Ndufa3, Cyp4a14, 1500017E21Rik, Asgr1, Ndufa1, Apoa5, Igfbp4
##     Gchfr, Leap2, Aadac, Ugt2b36, Ass1, Abcb4, Csad, Crot, Pon1, Hypk
## PC_ 5
## Positive:  Atp5j, Cox6b1, Hint1, Mup21, Cox6a1, Cyb5, Mup10, Acat1, Ifit1, Prdx1
##     Ndufa1, Rps27l, Fabp2, Mup1, Leap2, Isg15, Mpc2, G0s2, Akap7, Ndufa3
##     Id2, Pdcd5, Hamp, Plgrkt, 1100001G20Rik, Ifit3, Itm2b, Lgals1, Oasl1, Cc2d1b
## Negative:  Malat1, Hpx, Gm26924, Gas2l1, mt-Rnr1, Selenbp2, Eef2, Mug1, Vtn, Rnf6
##     Slc7a2, Pomt2, Plg, Gm23935, Cars2, Eif4e3, Neat1, Ces2a, Vcp, Cbs
##     Pyroxd2, Gmcl1, Itih3, Ubc, Aldh1b1, Nktr, Cyp3a11, Egfr, Invs, mt-Rnr2
```

```
# ___Examine and visualize PCA results a few different ways -------
# print(gse[["pca"]], dims = 1:4, nfeatures = 5)
```

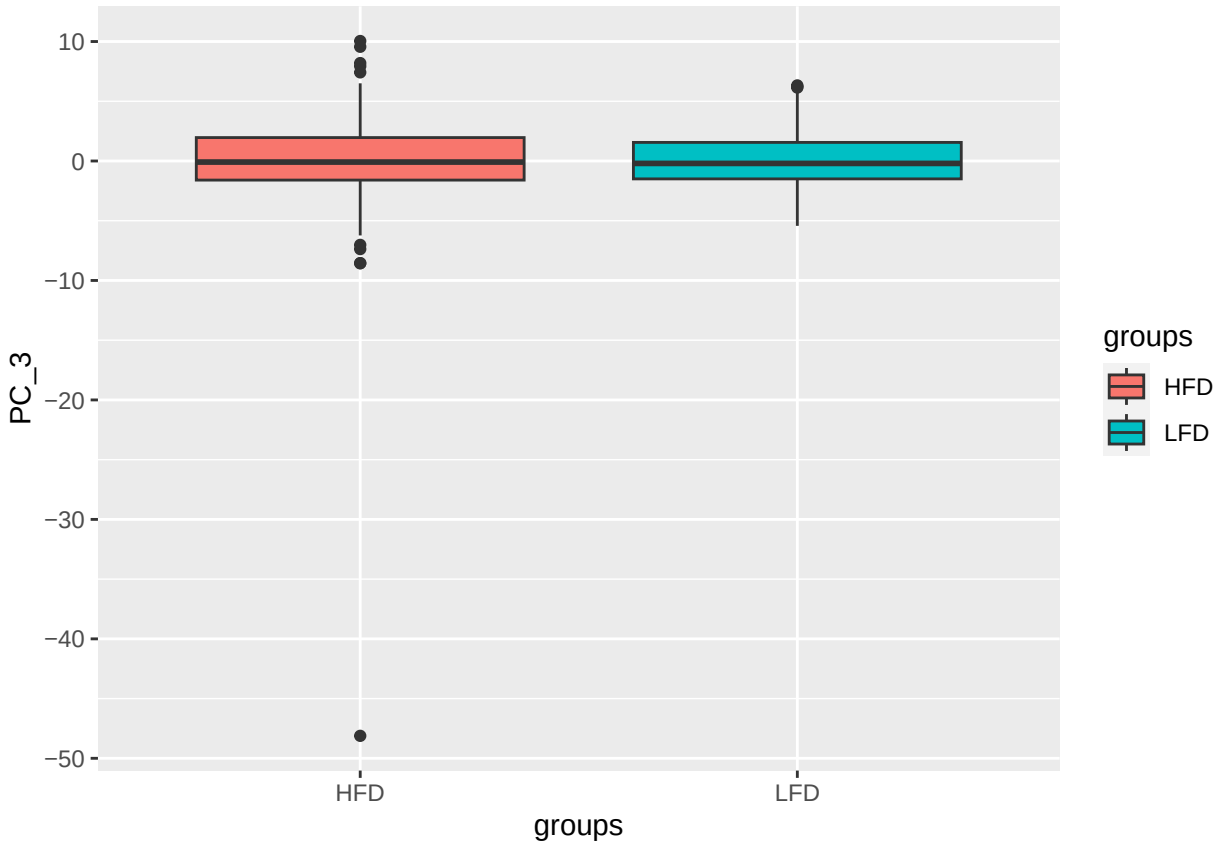
```
pca_data <- tibble(as.data.frame(gse[["pca"]]@cell.embeddings)) %>%
  mutate(groups = gse@meta.data$groups)

pca_data %>% ggplot(aes(x=groups, y=PC_1, fill=groups)) + geom_boxplot()
```



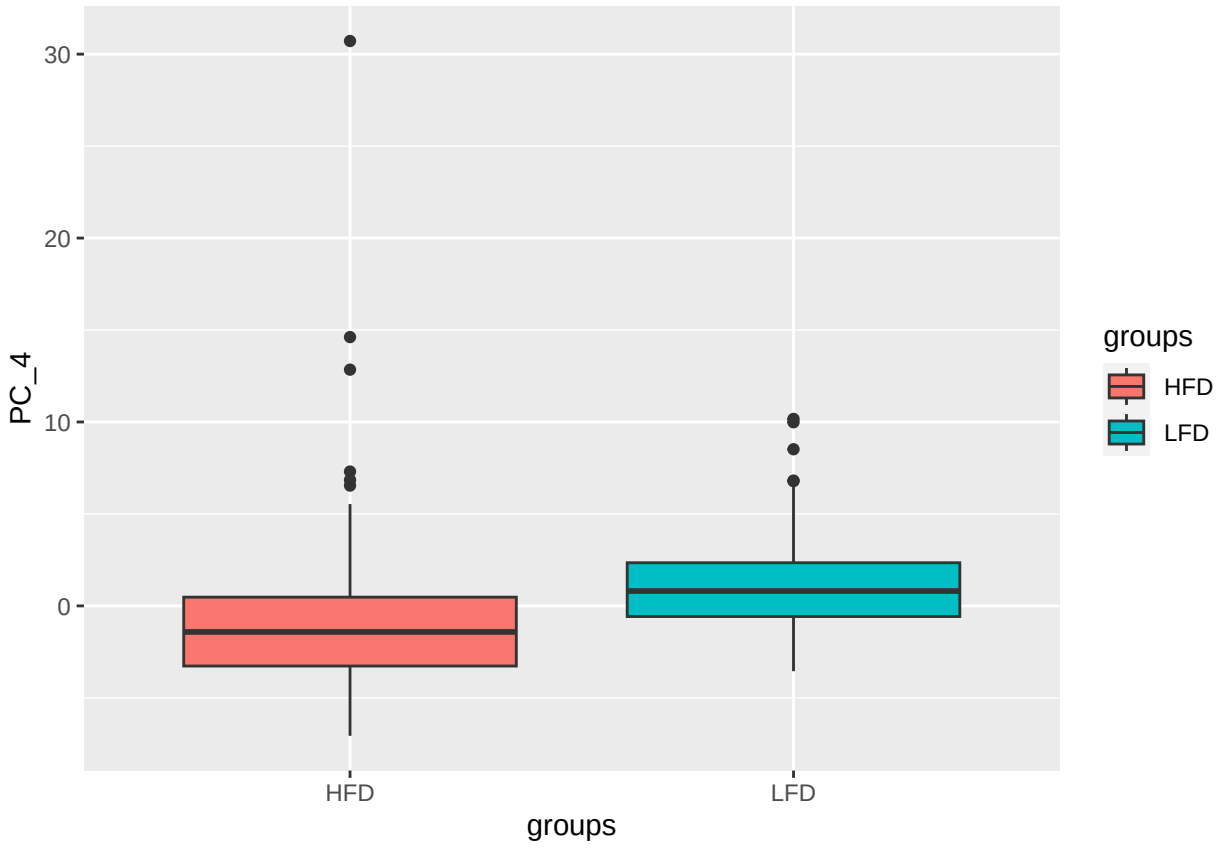

```
pca_data %>% ggplot(aes(x=groups, y=PC_2, fill=groups)) + geom_boxplot()
```



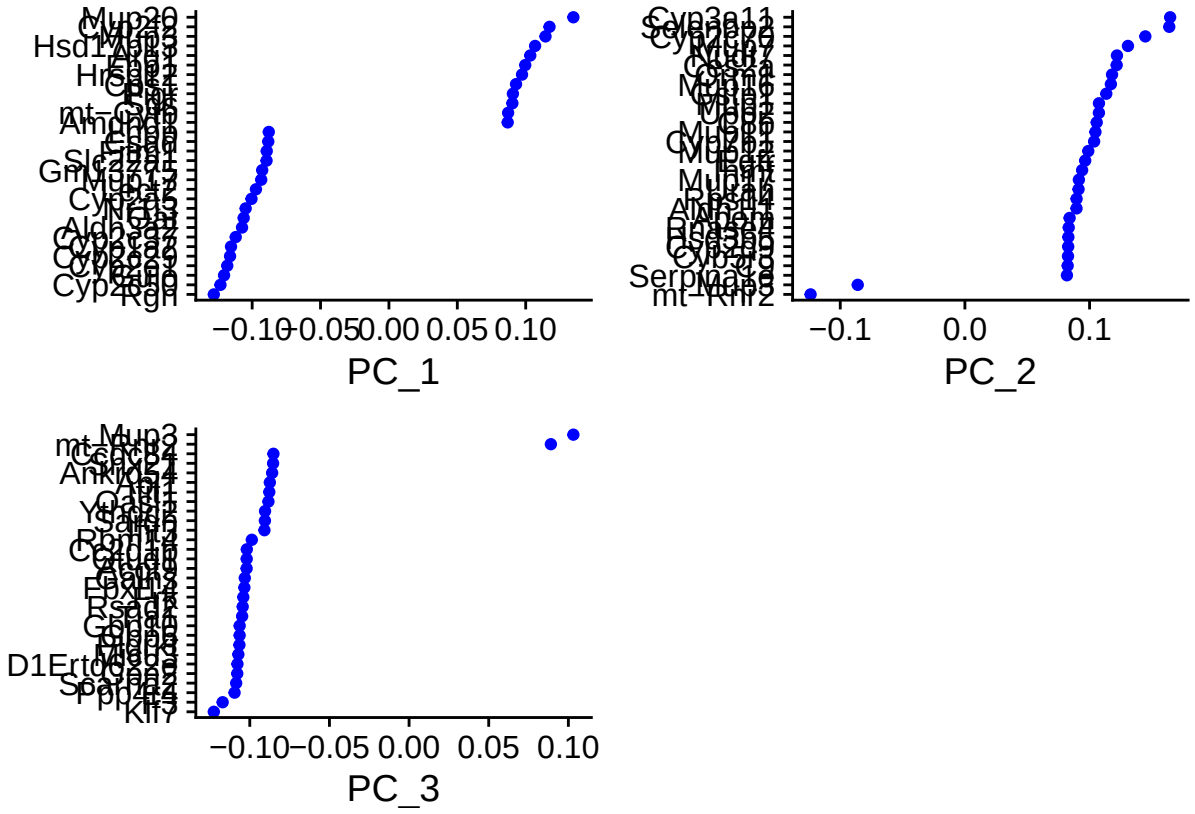

```
pca_data %>% ggplot(aes(x=groups, y=PC_3, fill=groups)) + geom_boxplot()
```



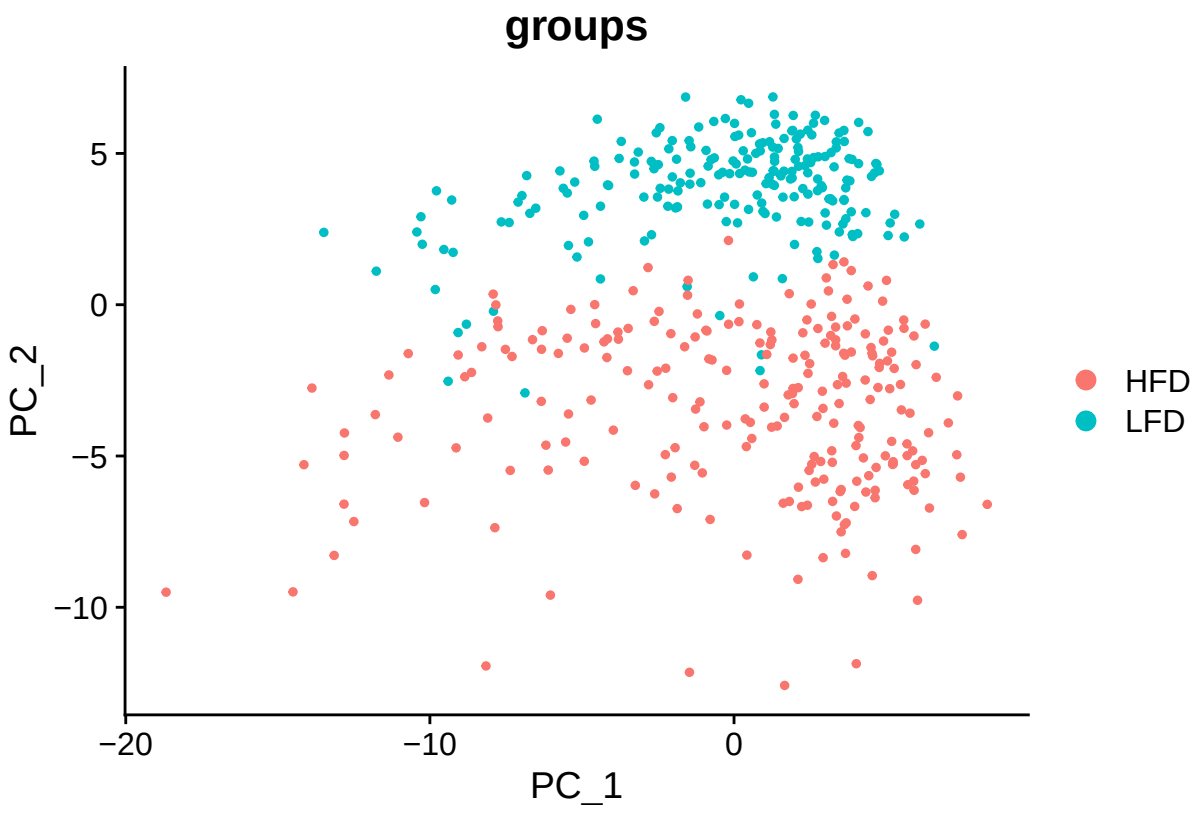

```
pca_data %>% ggplot(aes(x=groups, y=PC_4, fill=groups)) + geom_boxplot()
```



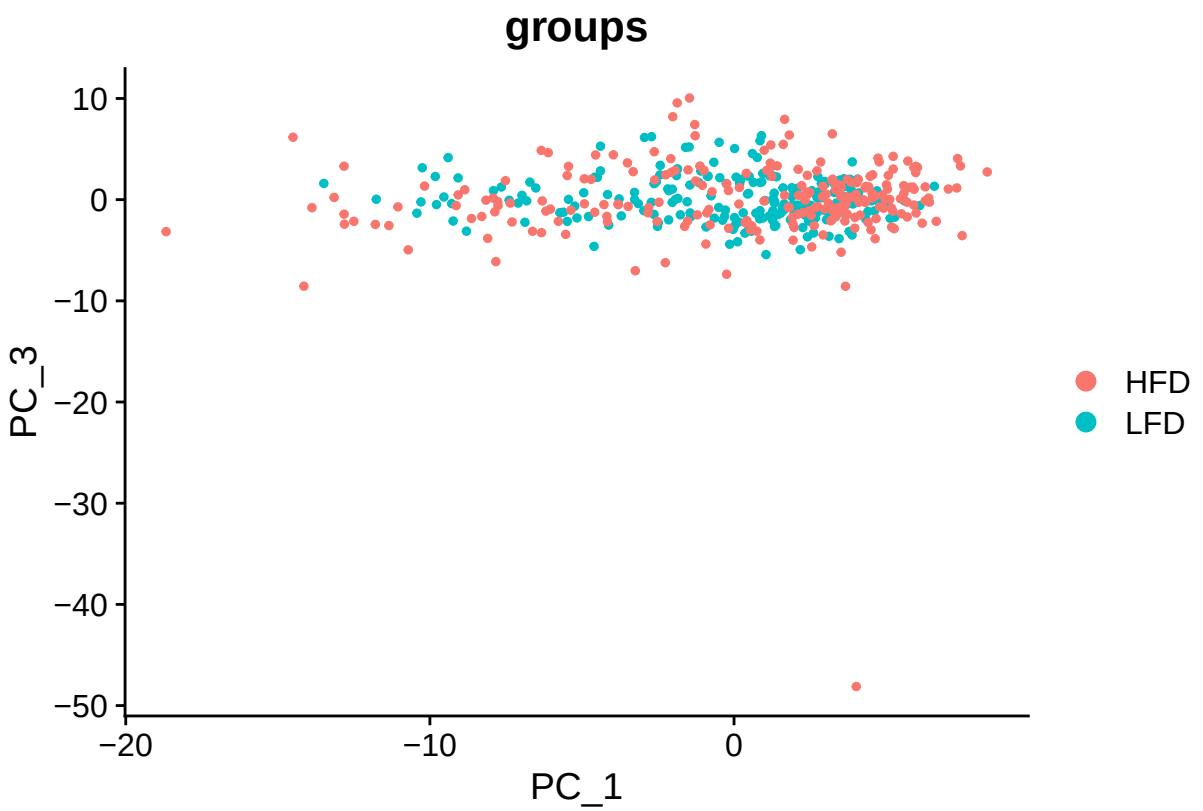

```
# -------- plot-1 --------
VizDimLoadings(gse, dims = 1:3, reduction = "pca")
```



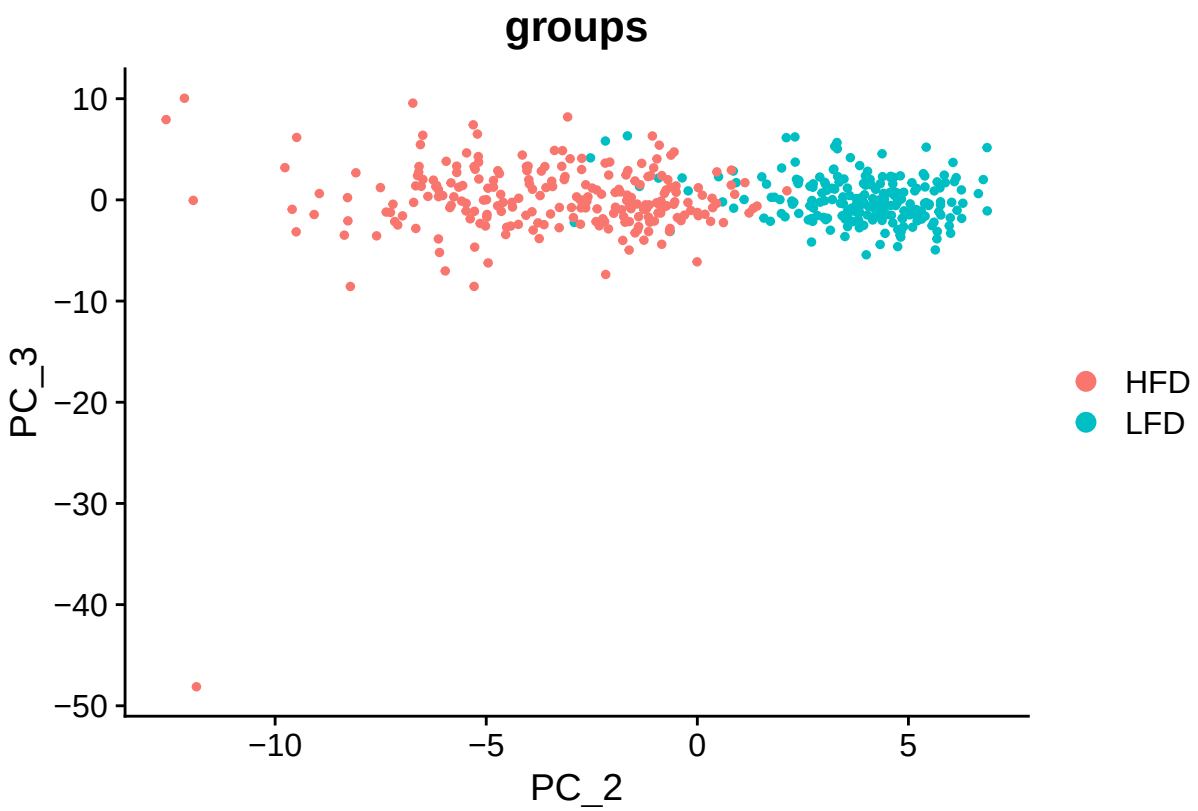

```
# -------- plot-2 by diet --------
DimPlot(gse, reduction = "pca", dims = 1:2, group.by = "groups")
```



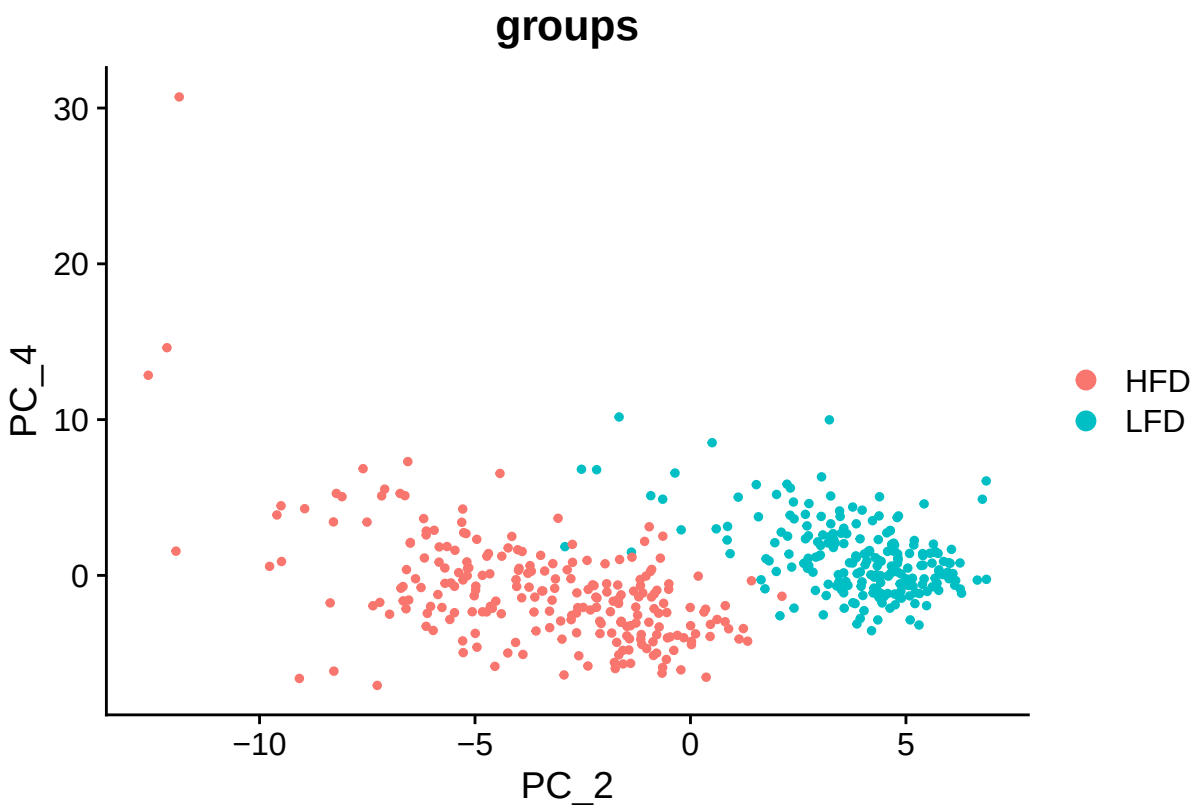
```
DimPlot(gse, reduction = "pca", dims = c(1,3), group.by = "groups")
```



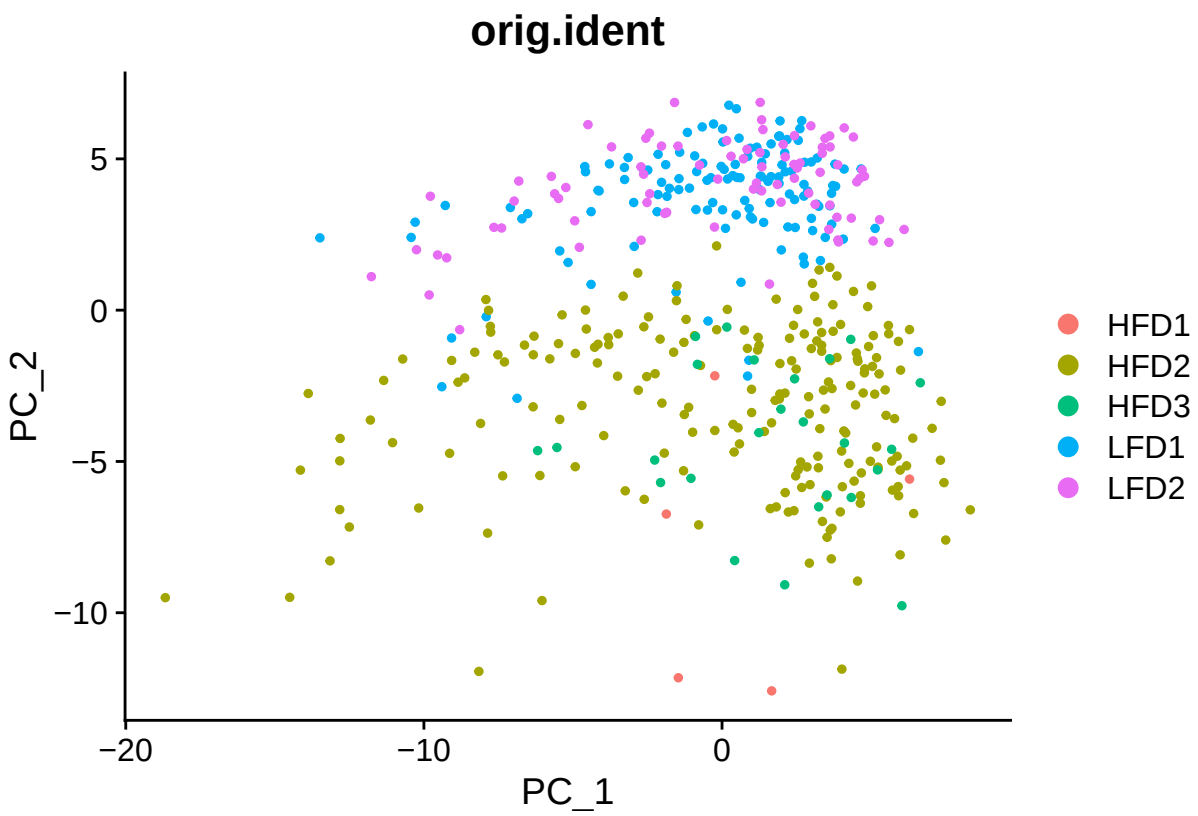

```
DimPlot(gse, reduction = "pca", dims = 2:3, group.by = "groups")
```



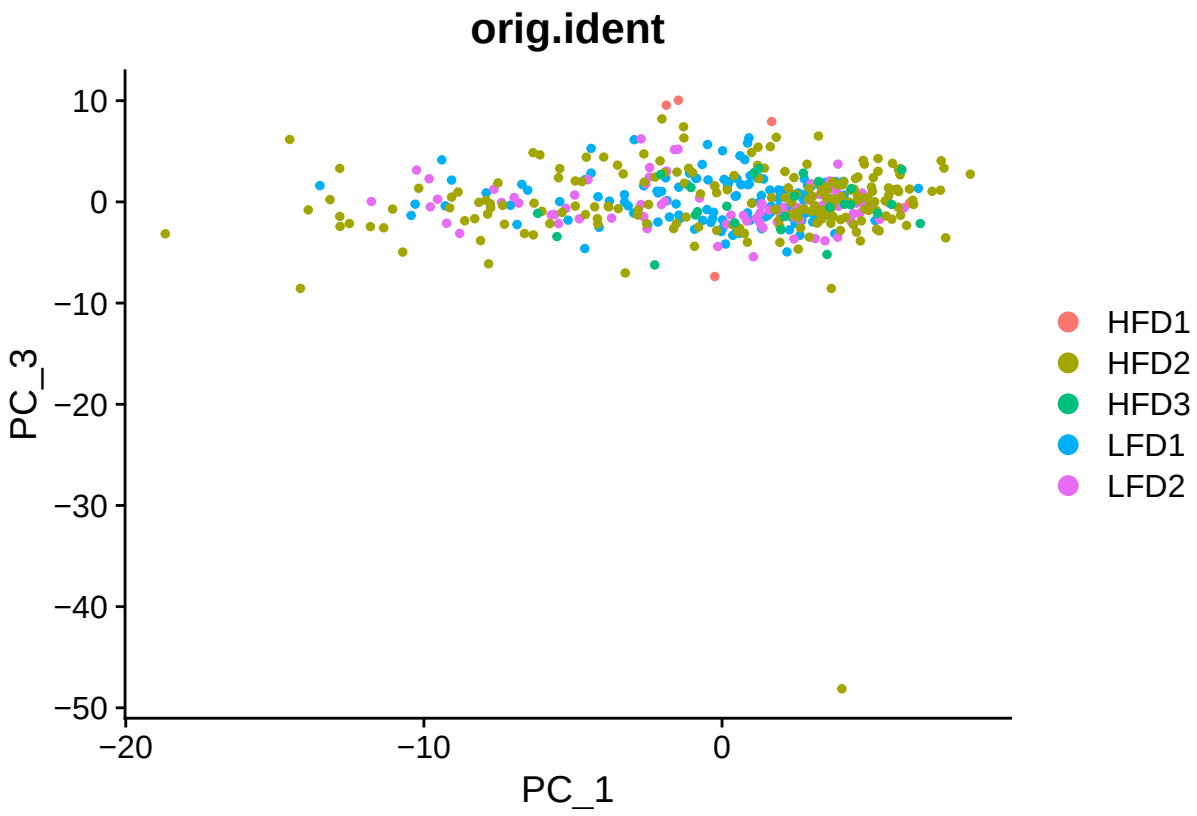

```
DimPlot(gse, reduction = "pca", dims = c(2,4), group.by = "groups")
```



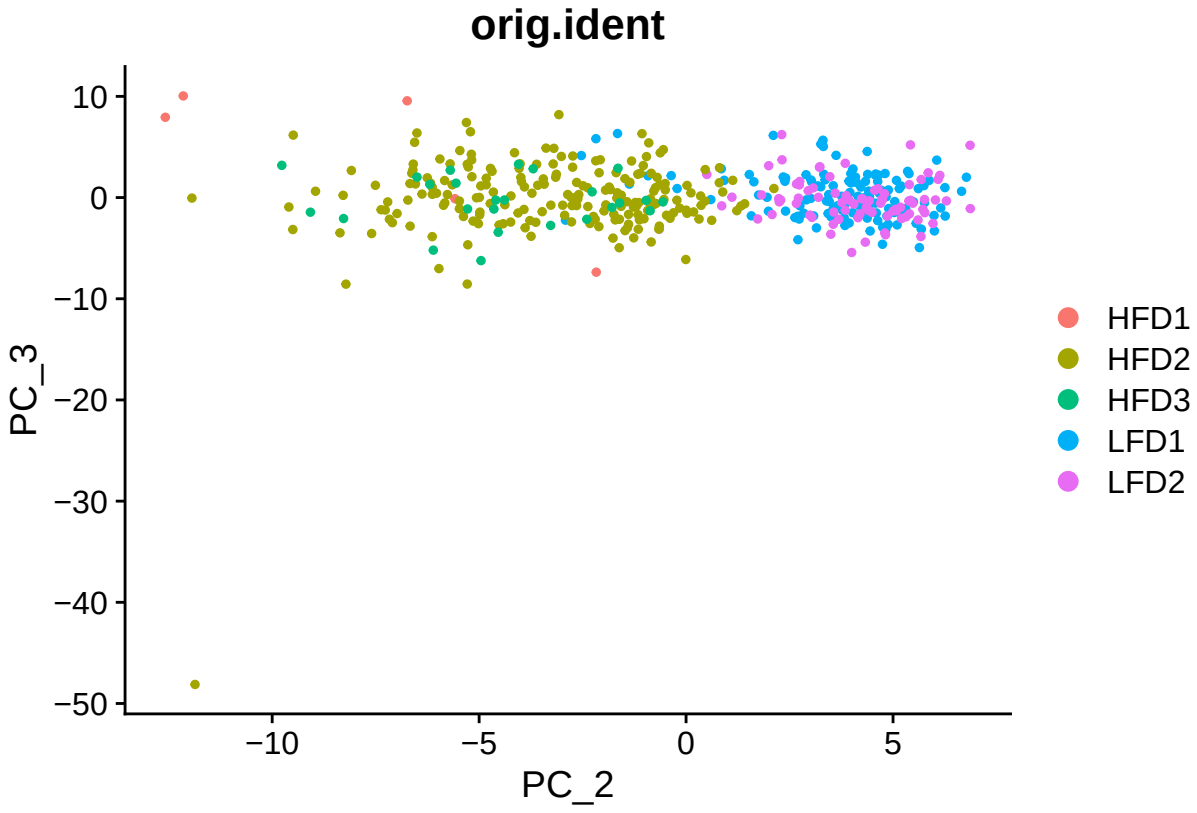
```r
# -------- plot-3 by sample --------
DimPlot(gse, reduction = "pca", dims = 1:2, group.by = "orig.ident")
```



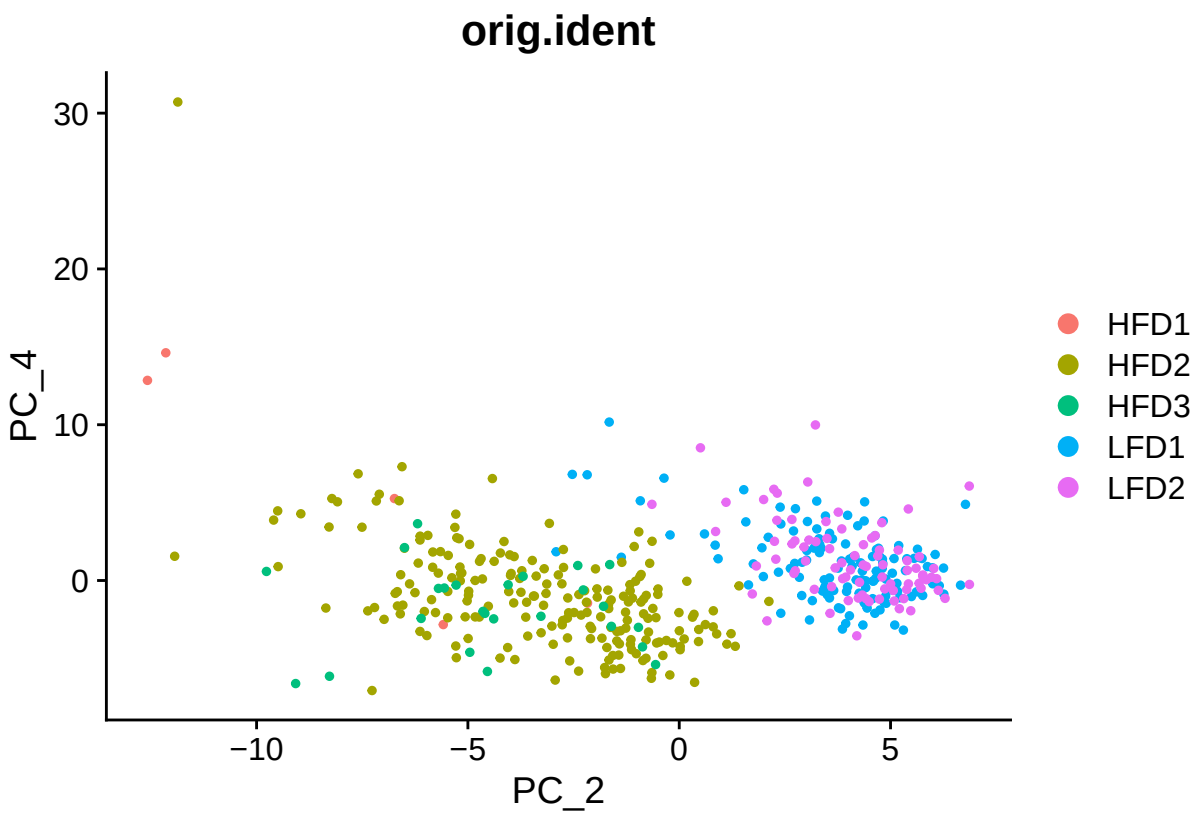

```r
DimPlot(gse, reduction = "pca", dims = c(1,3), group.by = "orig.ident")
```




```
DimPlot(gse, reduction = "pca", dims = 2:3, group.by = "orig.ident")
```

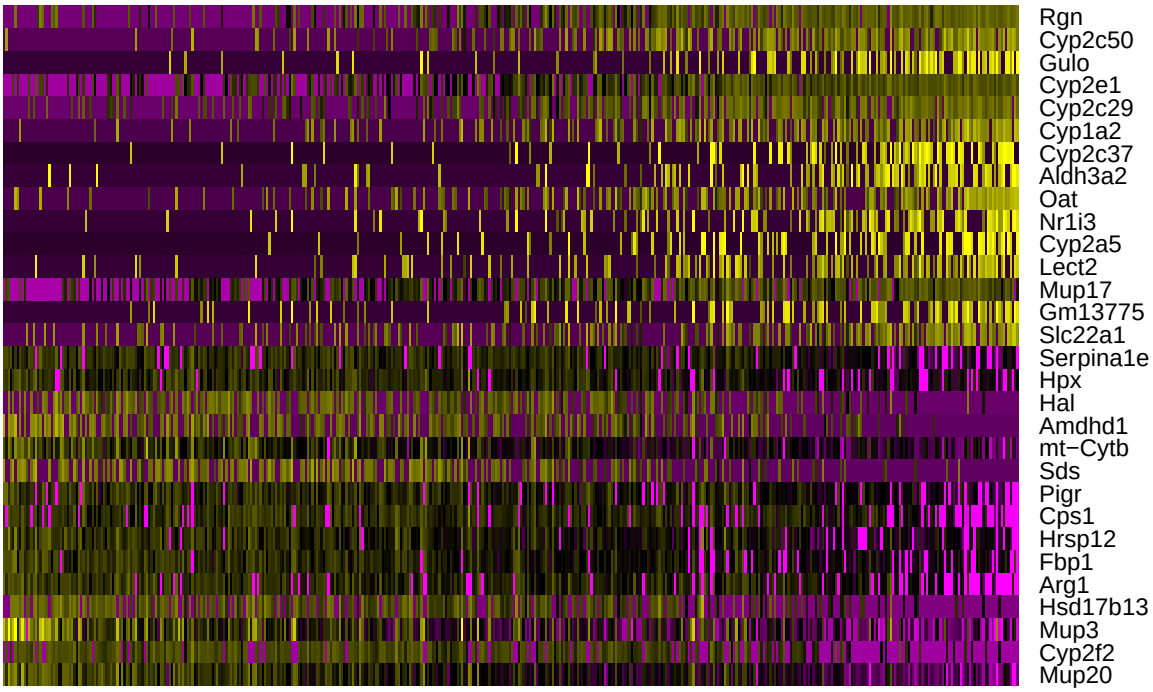


```
DimPlot(gse, reduction = "pca", dims = c(2,4), group.by = "orig.ident")
```



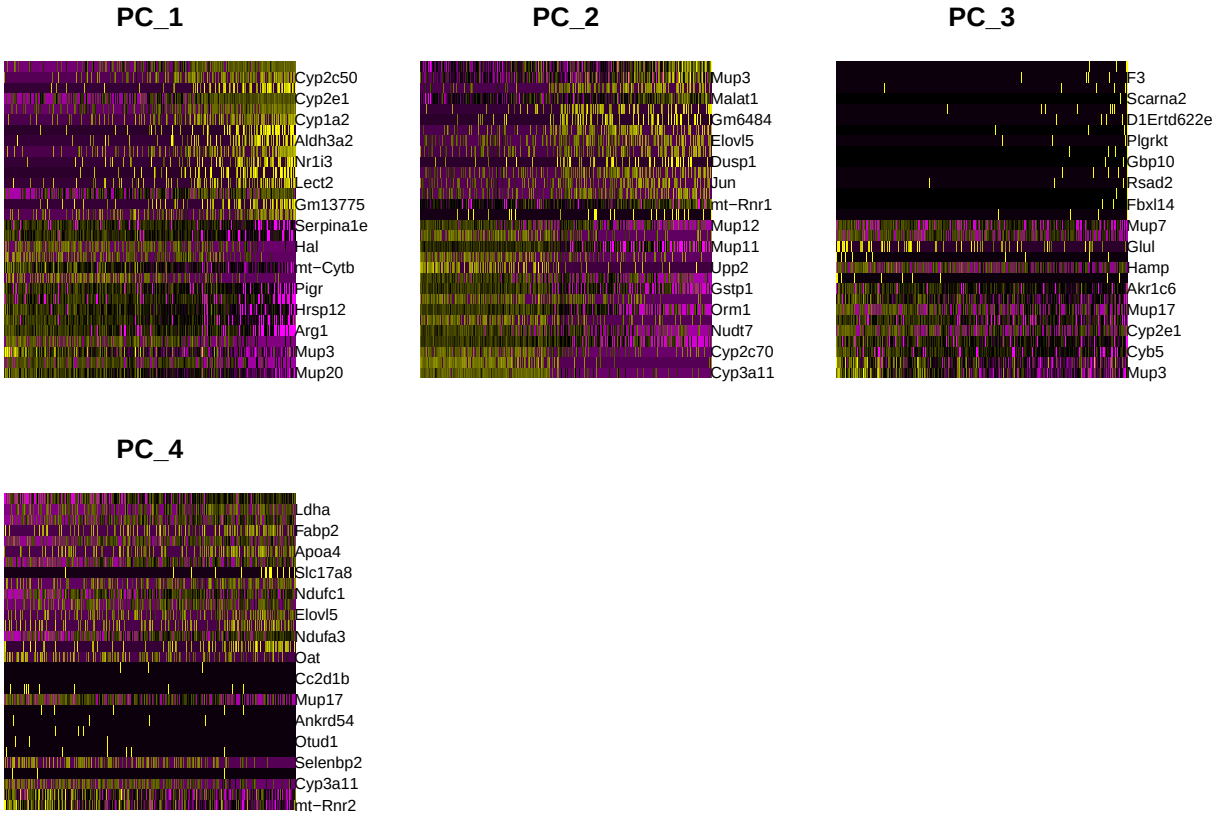

```
# ___plot-3 heatmap -------
# allows for easy exploration of the primary sources of heterogeneity in a dataset
# and can be useful when trying to decide which PCs to include for further downstream analyses
DimHeatmap(gse, dims = 1, cells = 500, balanced = TRUE)
```

```
## Warning: Requested number is larger than the number of available items (449).
## Setting to 449.
```




```
DimHeatmap(gse, dims = 1:4, cells = 500, balanced = TRUE)
```

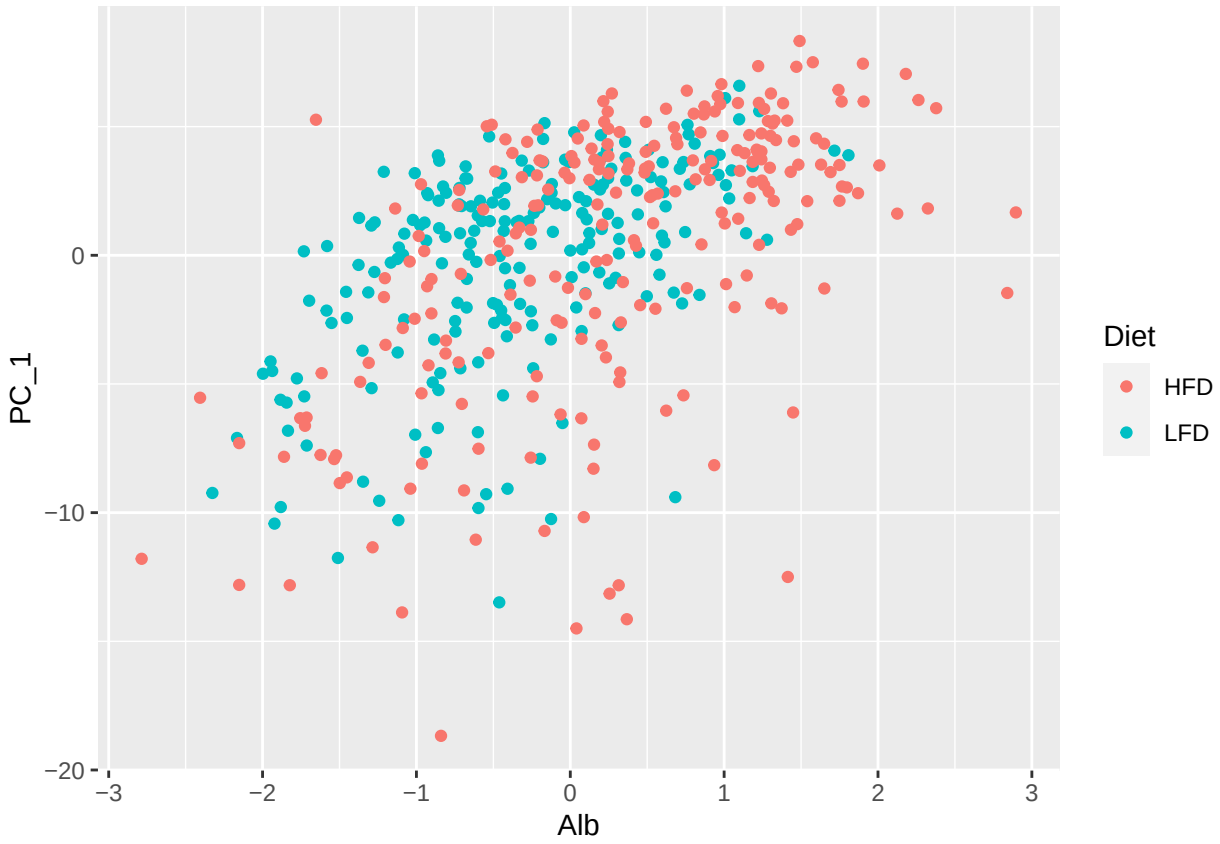
```
## Warning: Requested number is larger than the number of available items (449).
## Setting to 449.

## Warning: Requested number is larger than the number of available items (449).
## Setting to 449.

## Warning: Requested number is larger than the number of available items (449).
## Setting to 449.

## Warning: Requested number is larger than the number of available items (449).
## Setting to 449.
```




```
alb_pc1 <- tibble(PC_1 = as.data.frame(gse[["pca"]]@cell.embeddings)$PC_1,
       Alb = gse@assays$RNA@scale.data['Alb',],
       Diet = gse@meta.data$groups)

cor.test(x = alb_pc1$Alb, y = alb_pc1$PC_1)
```

```
##
##  Pearson's product-moment correlation
##
## data:  alb_pc1$Alb and alb_pc1$PC_1
## t = 13.21, df = 447, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4598864 0.5933245
## sample estimates:
##      cor
## 0.529877
```

```
alb_pc1 %>%
  ggplot(aes(x=Alb, y=PC_1, col=Diet)) + geom_point()
```




```
# Add hepatocyte Zonation score (Arg1 - Cyp2e1 imputed expression levels within cells)
# Using magic counts to replicate paper

zone_diff <- counts_magic['Arg1',] - counts_magic['Cyp2e1',]
gse@meta.data['zonation'] <- as.numeric(zone_diff[rownames(gse@meta.data)])
gse@meta.data['zone_group'] <- ifelse(gse@meta.data['zonation'] >= (2.375 - 1.84) & gse@meta.data$groups
                                      ifelse(gse@meta.data['zonation'] >= (2.35 - 1.78) & gse@meta.data
                                        ifelse(gse@meta.data['zonation'] >= (2.35 - 1.9) & gse@meta.data
                                            ifelse(gse@meta.data['zonation'] >= (2.31 - 1.9) & gse@meta
                                              'Zone_3')
                                            )
                                        )
                                      )


cyp2e1 <- counts_magic['Cyp2e1',]
arg1 <- counts_magic['Arg1',]

arg1_pc1 <- tibble(PC_1 = as.data.frame(gse[["pca"]]@cell.embeddings)$PC_1,
                   PC_2 = as.data.frame(gse[["pca"]]@cell.embeddings)$PC_2,
                   PC_3 = as.data.frame(gse[["pca"]]@cell.embeddings)$PC_3,
                   PC_4 = as.data.frame(gse[["pca"]]@cell.embeddings)$PC_4,
                   Arg1 = gse@assays$RNA@scale.data['Arg1',],
                   Arg1_magic = as.numeric(arg1[names(gse@assays$RNA@scale.data['Arg1',])]),
                   Cyp2e1 = gse@assays$RNA@scale.data['Cyp2e1',],
                   Cyp2e1_magic = as.numeric(cyp2e1[names(gse@assays$RNA@scale.data['Cyp2e1',])]),
                   Diet = gse@meta.data$groups,
```

```
                  Zone = as.character(gse@meta.data$zone_group))

# ----- correlations ------
print(cor.test(x = arg1_pc1$Arg1, y = arg1_pc1$PC_1))
```

```
##
##  Pearson's product-moment correlation
##
## data:  arg1_pc1$Arg1 and arg1_pc1$PC_1
## t = 11.818, df = 447, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.4140749 0.5553840
## sample estimates:
##       cor
## 0.4879197
```

```
print(cor.test(x = arg1_pc1$Arg1_magic, y = arg1_pc1$PC_1))
```

```
##
##  Pearson's product-moment correlation
##
## data:  arg1_pc1$Arg1_magic and arg1_pc1$PC_1
## t = 30.994, df = 447, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7942864 0.8534050
## sample estimates:
##       cor
## 0.8261055
```

```
print(cor.test(x = arg1_pc1$Cyp2e1, y = arg1_pc1$PC_1))
```

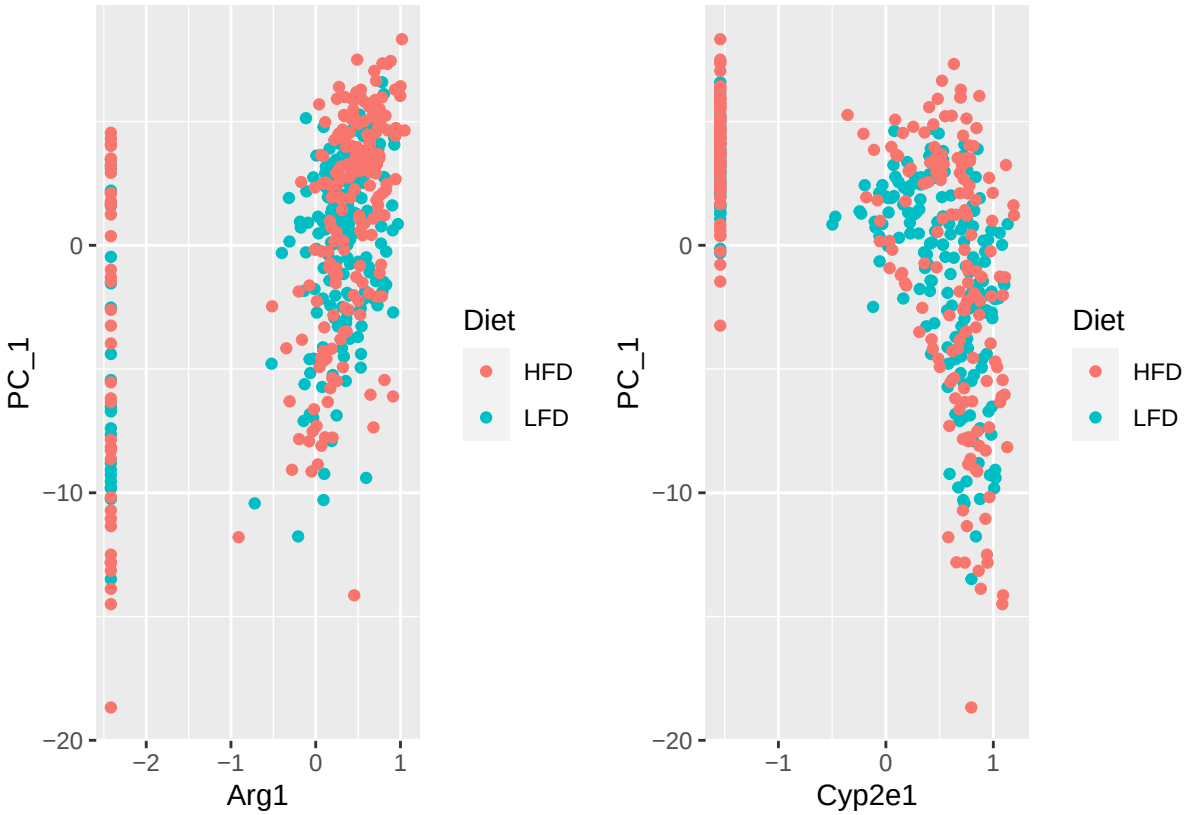
```
##
##  Pearson's product-moment correlation
##
## data:  arg1_pc1$Cyp2e1 and arg1_pc1$PC_1
## t = -14.218, df = 447, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6186445 -0.4908597
## sample estimates:
##       cor
## -0.5580517
```

```
print(cor.test(x = arg1_pc1$Cyp2e1_magic, y = arg1_pc1$PC_1))
```

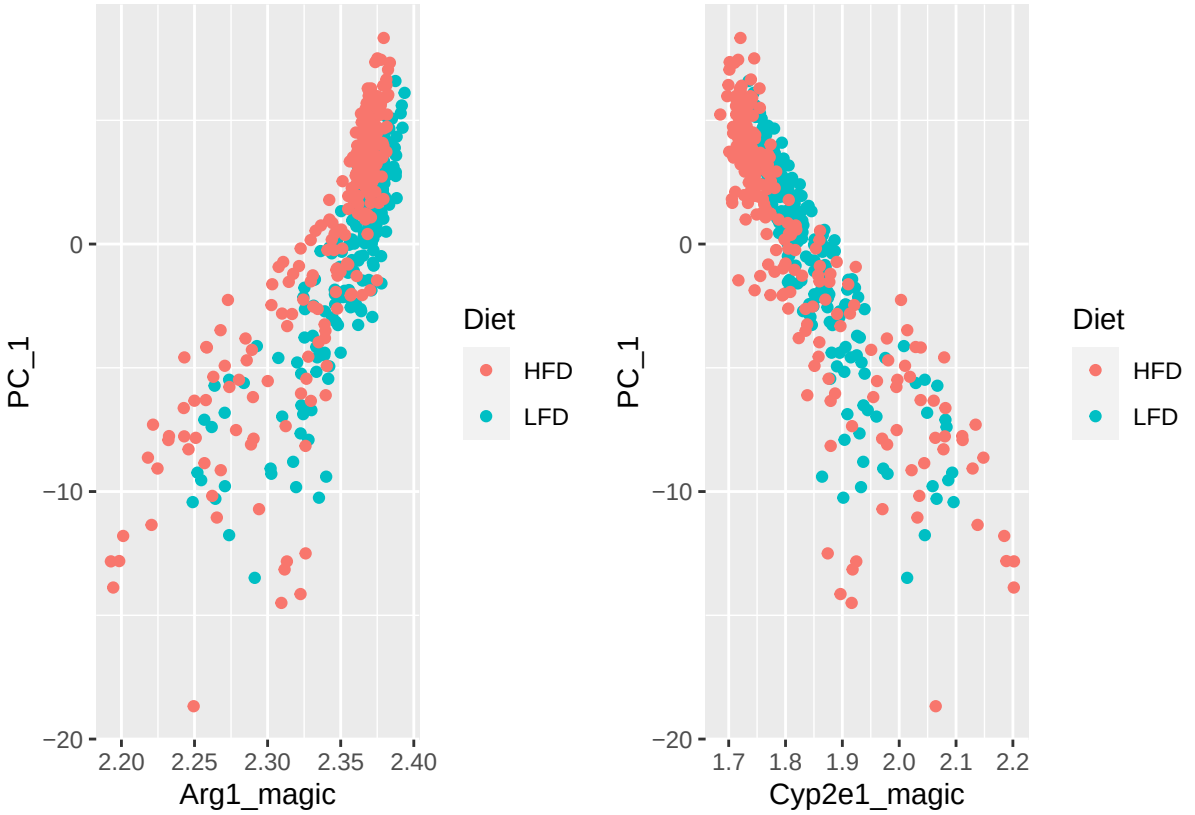
```
##
##  Pearson's product-moment correlation
##
## data:  arg1_pc1$Cyp2e1_magic and arg1_pc1$PC_1
```

```
## t = -38.39, df = 447, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.8958701 -0.8525141
## sample estimates:
##        cor
## -0.8759493
```

```
# ----- plot correlations -----
arg1.p<- arg1_pc1 %>%
  ggplot(aes(x=Arg1, y=PC_1, col=Diet)) + geom_point()
cyp2e1.p<- arg1_pc1 %>%
  ggplot(aes(x=Cyp2e1, y=PC_1, col=Diet)) + geom_point()

arg1.p + cyp2e1.p
```

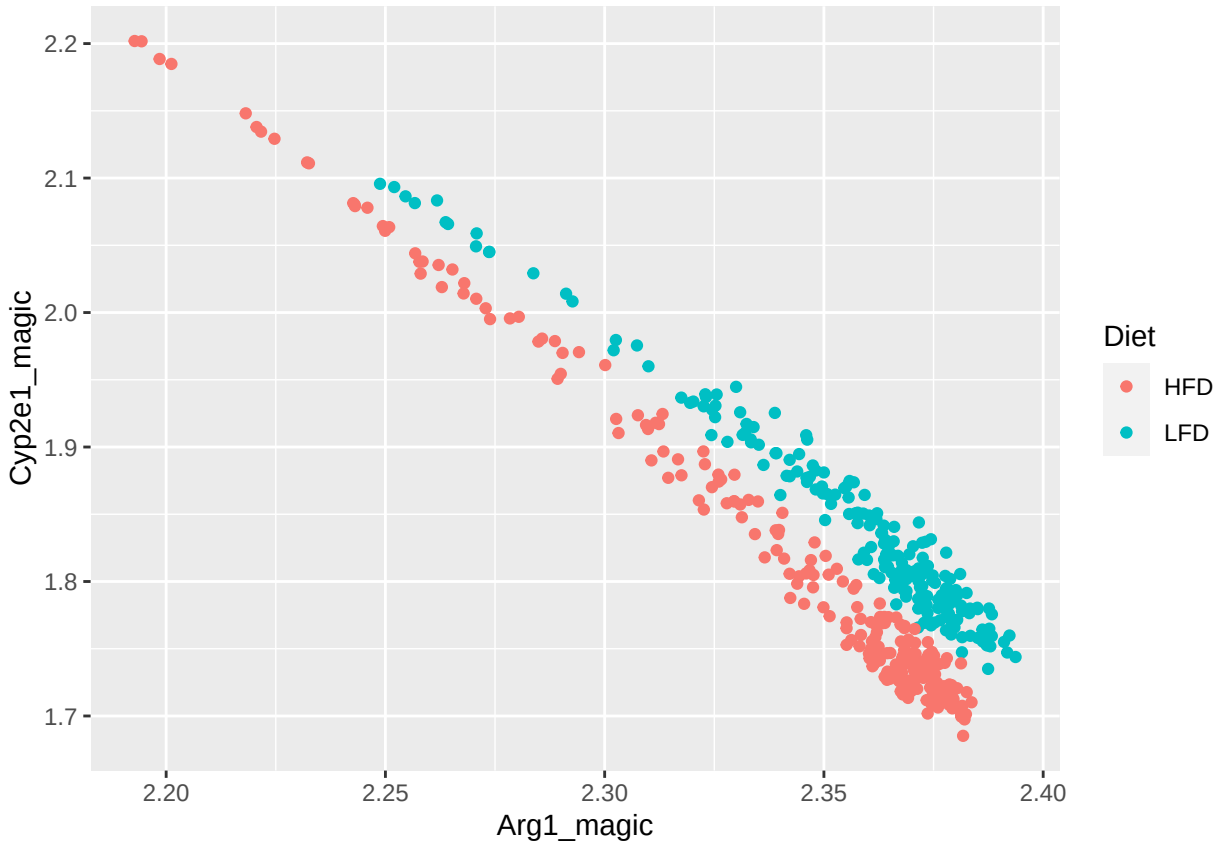



```
arg1_magic.p<- arg1_pc1 %>%
  ggplot(aes(x=Arg1_magic, y=PC_1, col=Diet)) + geom_point()
cyp2e1_magic.p<- arg1_pc1 %>%
  ggplot(aes(x=Cyp2e1_magic, y=PC_1, col=Diet)) + geom_point()

arg1_magic.p + cyp2e1_magic.p
```

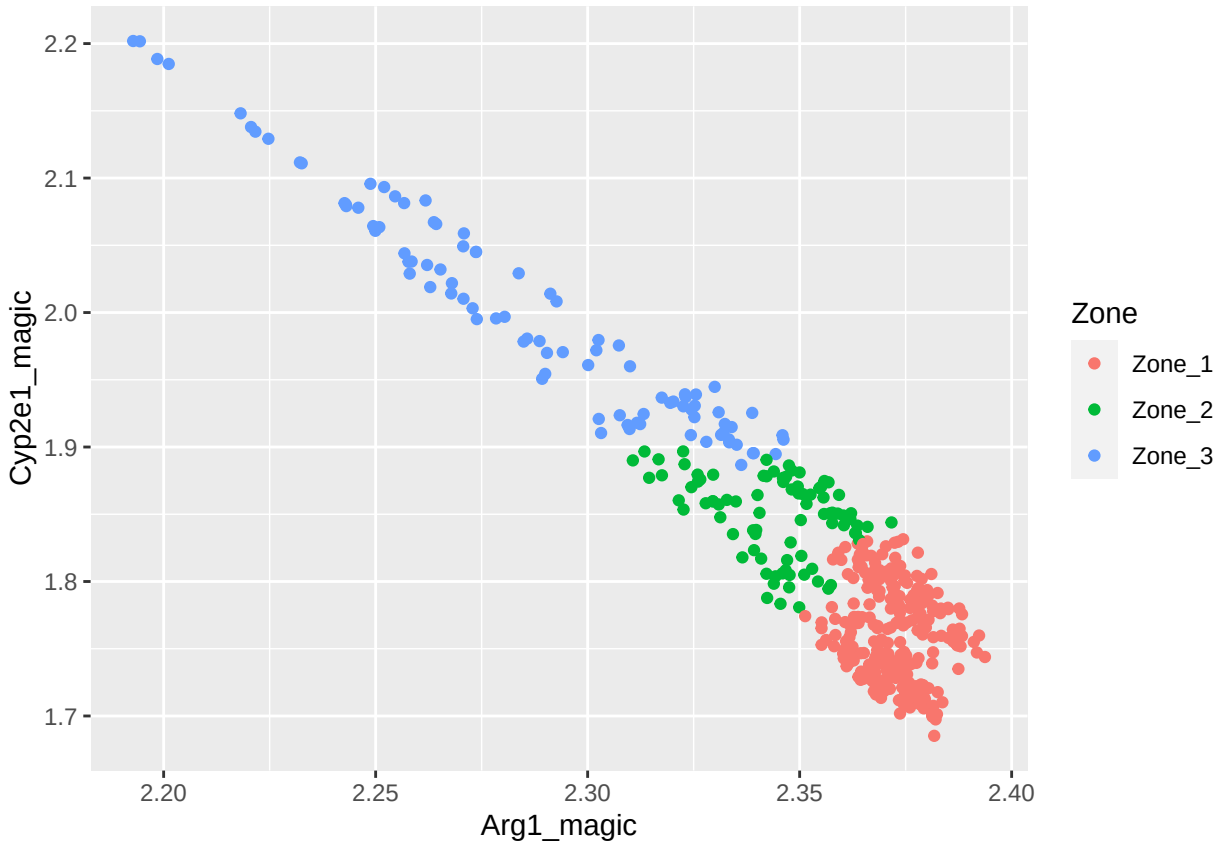


Generate scatter plot of Arg1 vs Cyp2e1 gene markers colored by diet or zone, as shown in the original paper

```
cyp2e1_arg1_magic.p <- arg1_pc1 %>%
  ggplot(aes(x=Arg1_magic, y=Cyp2e1_magic, col=Diet)) + geom_point()
cyp2e1_arg1_magic.p
```

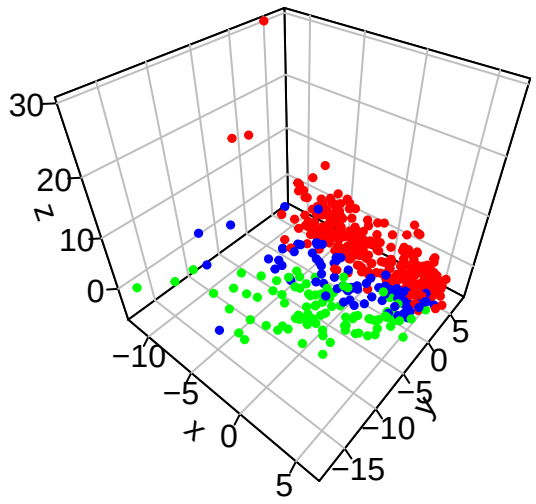



```
cyp2e1_arg1_magic.p <- arg1_pc1 %>%
  ggplot(aes(x=Arg1_magic, y=Cyp2e1_magic, col=Zone)) + geom_point()
cyp2e1_arg1_magic.p
```

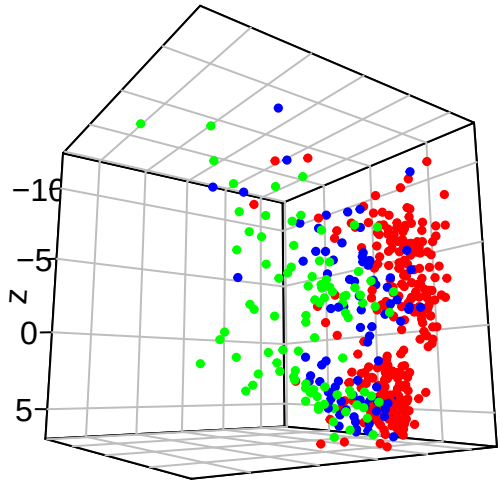


Generate 3D PCA plots

```
scatter3D(y = arg1_pc1$PC_1,
          x = arg1_pc1$PC_2,
          z = arg1_pc1$PC_4,
          colvar = NULL,
          col = c("red","blue","green")[as.factor(arg1_pc1$Zone)],
          pch = 19,
          cex = 0.5,
          bty = "b2",
          ticktype = "detailed"
          )
```

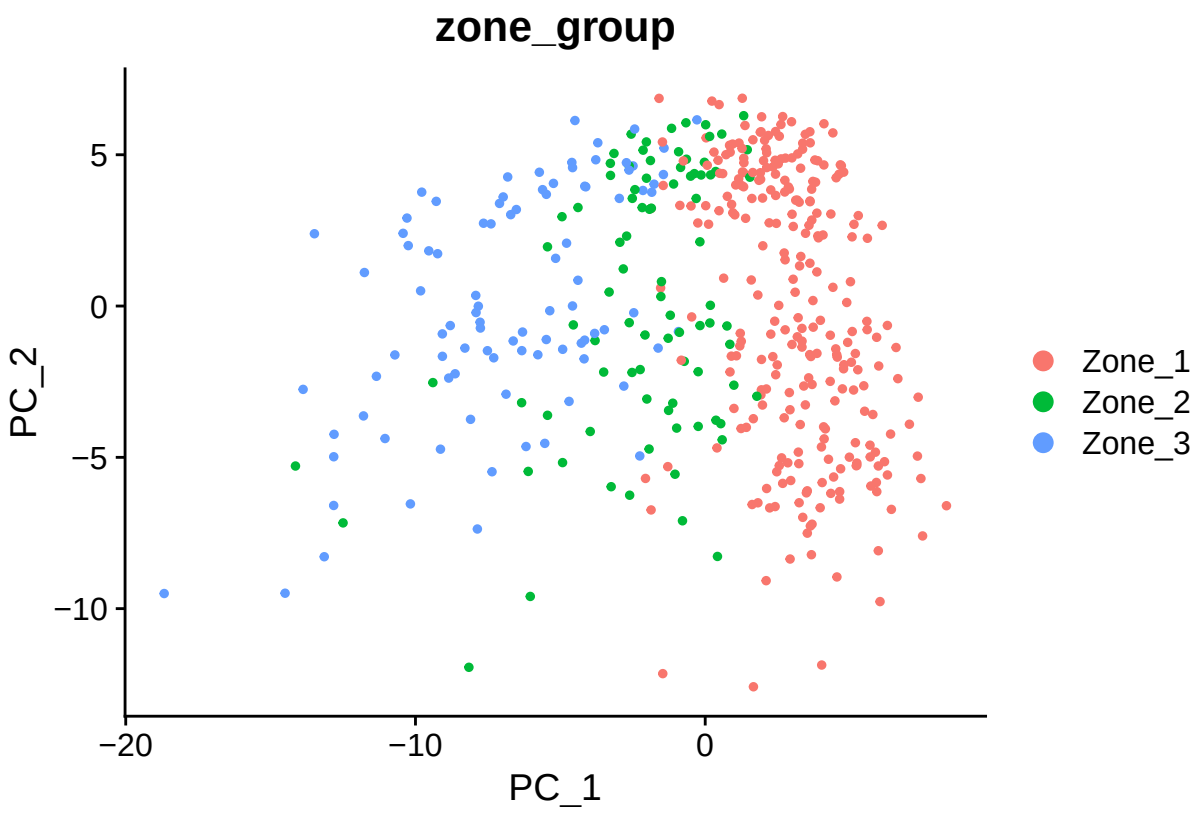



```
scatter3D(y = arg1_pc1$PC_1,
          z = arg1_pc1$PC_2,
          x = arg1_pc1$PC_4,
          colvar = NULL,
          col = c("red","blue","green")[as.factor(arg1_pc1$Zone)],
          pch = 19,
          cex = 0.5,
          bty = "b2",
          ticktype = "detailed",
          theta = 125, phi = 170
          )
```



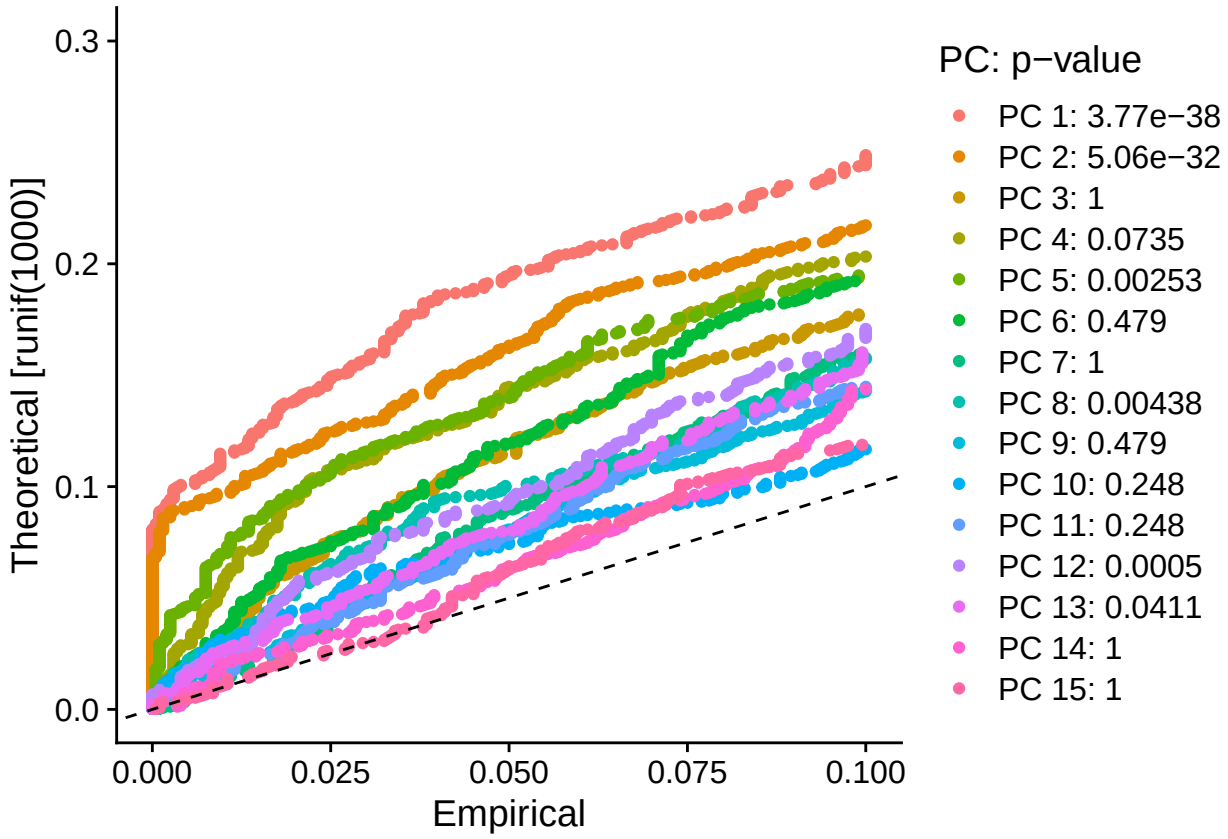
Plot PCA coloring by zone.

```
pca.12.p <- DimPlot(gse, reduction = "pca", dims = 1:2, group.by = "zone_group")
pca.12.p
```

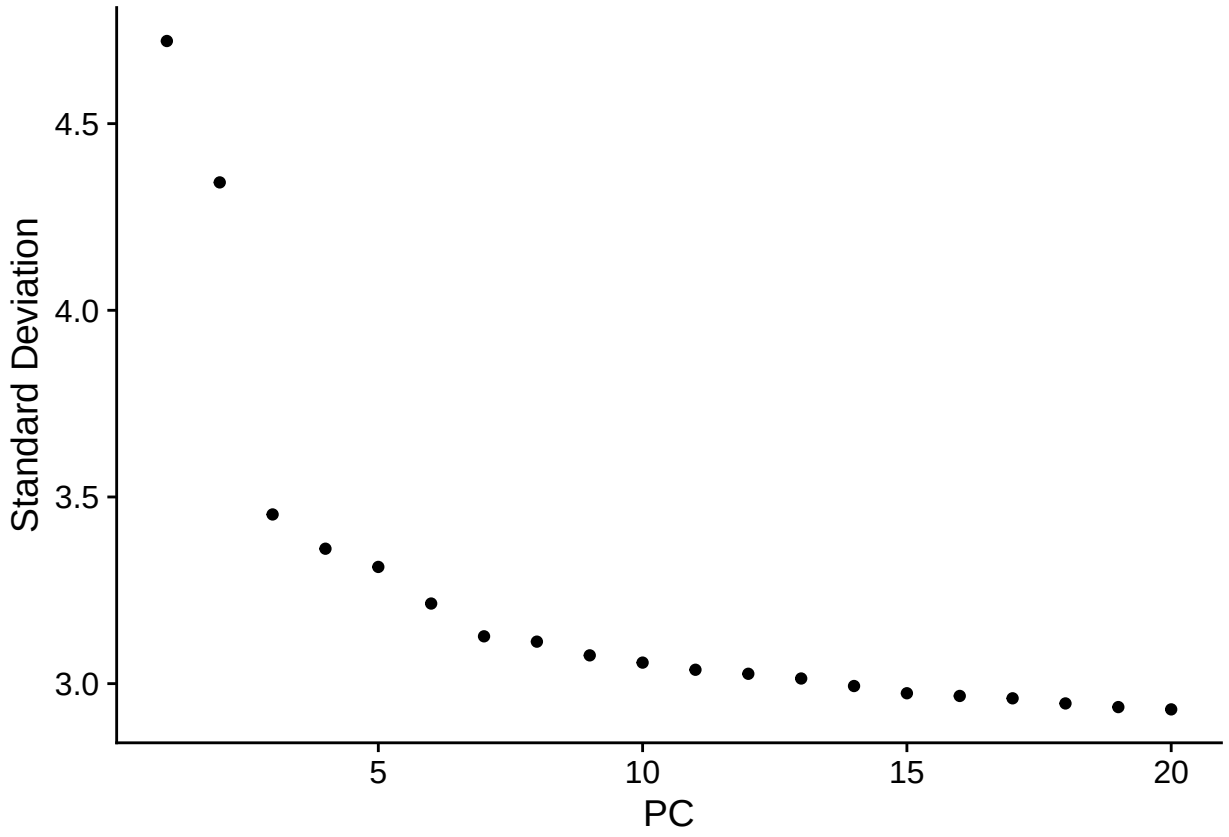



```
# ___to dertermine "dimensionality" of the dataset -------
# essentially determine how many PCs to consider - we would ideally want to consider PCs that show maxi

# JackStraw Procedure!
# identify 'significant' PCs as those who have a strong enrichment of low p-value features.
# NOTE: This process can take a long time for big datasets, comment out for expediency. More
# approximate techniques such as those implemented in ElbowPlot() can be used to reduce
# computation time

gse <- JackStraw(gse, num.replicate = 100)
gse <- ScoreJackStraw(gse, dims = 1:20)

JackStrawPlot(gse, dims = 1:15)
```

```
## Warning: Removed 24895 rows containing missing values (`geom_point()`).
```




```
# The JackStrawPlot() function provides a visualization tool for comparing the distribution of p-values
# 'Significant' PCs will show a strong enrichment of features with low p-values (solid curve above the


# An alternative heuristic method generates an 'Elbow plot': a ranking of principle components based on
ElbowPlot(gse)
```




```
# from the plot, it looks like majority of true signal is captured in the first 15 PCs.
# PCs to consider = 15
```

```
# 6. Cluster cells --------
gse <- FindNeighbors(gse, dims = 1:15)
```

```
## Computing nearest neighbor graph
```

```
## Computing SNN
```

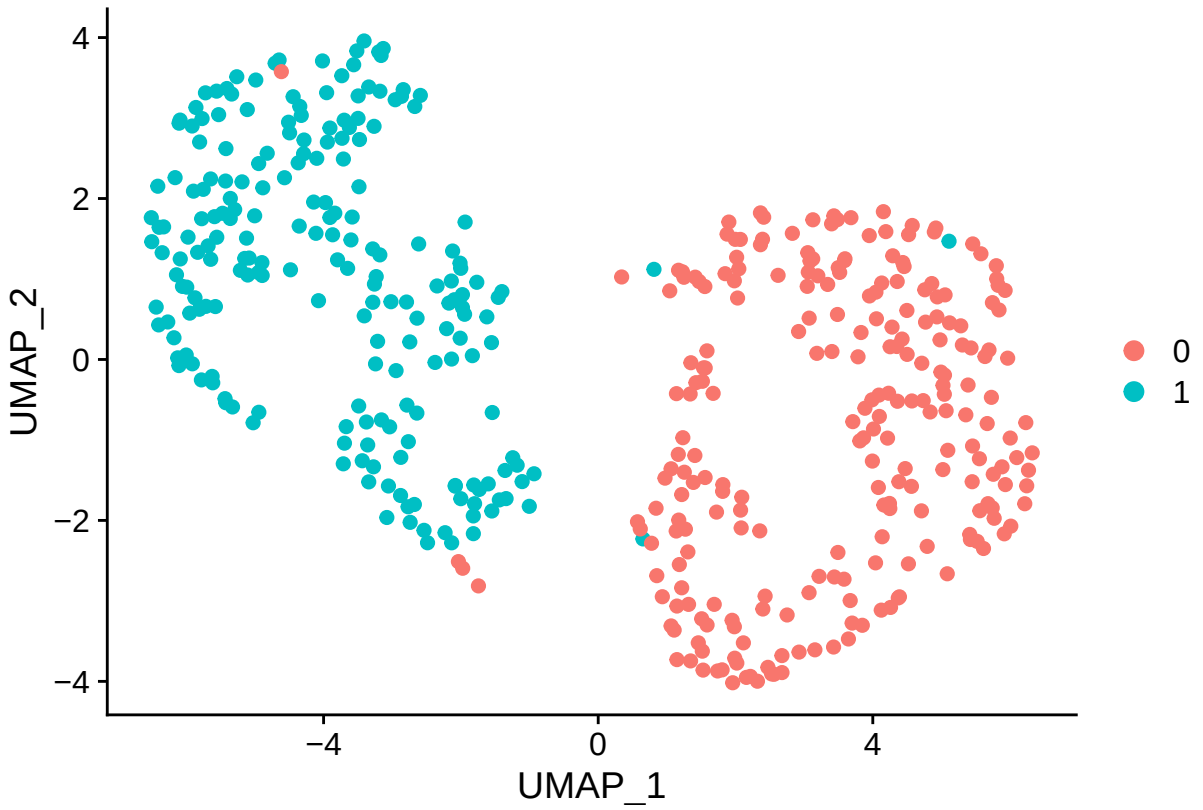
```
# The FindClusters() function contains a resolution parameter that sets the 'granularity' of the downst
# We find that setting this parameter between 0.4-1.2 typically returns good results for single-cell da
# Optimal resolution often increases for larger datasets.
gse <- FindClusters(gse, resolution = 0.2)
```

```
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 449
## Number of edges: 19680
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.8614
## Number of communities: 2
## Elapsed time: 0 seconds
```

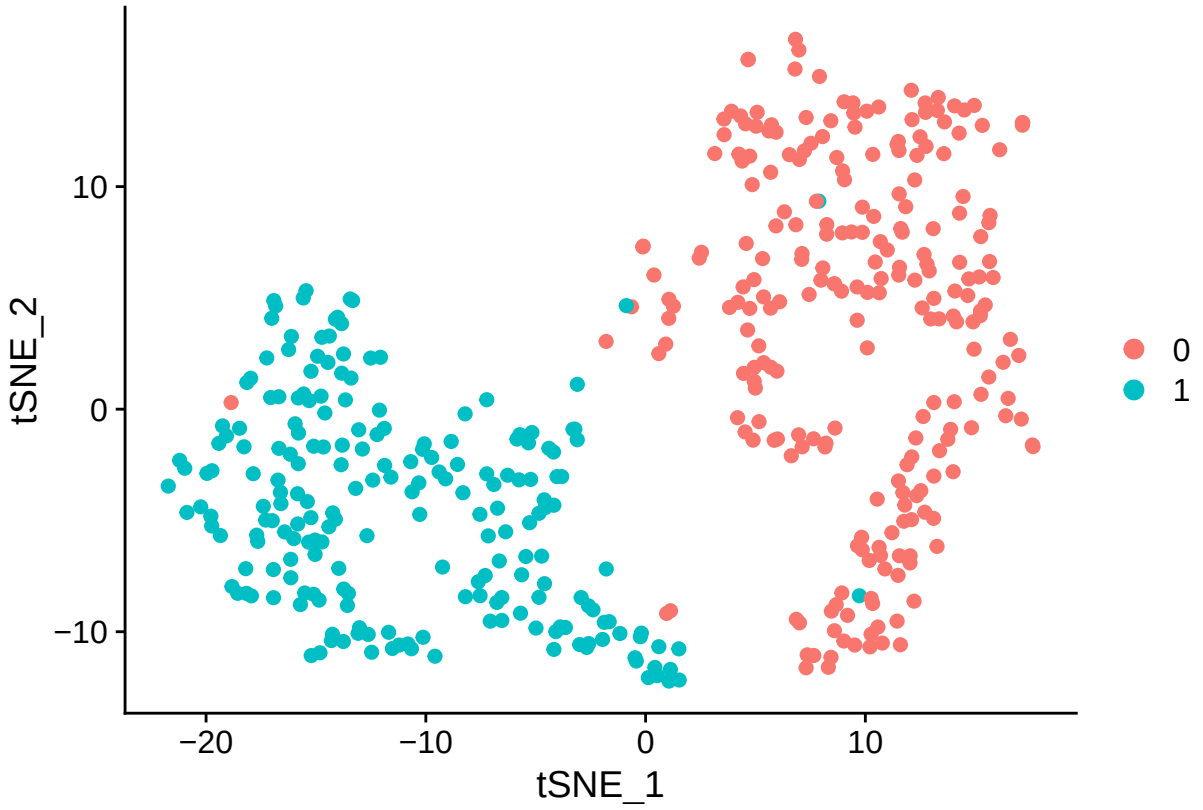
```r
# Look at cluster IDs of the first 5 cells
#head(Idents(gse), 5)



# 7. Run non-linear dimensional reduction (UMAP/tSNE) ---------
gse <- RunUMAP(gse, dims = 1:4)
```

```
## 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

## 11:39:01 UMAP embedding parameters a = 0.9922 b = 1.112

## 11:39:01 Read 449 rows and found 4 numeric columns

## 11:39:01 Using Annoy for neighbor search, n_neighbors = 30

## 11:39:01 Building Annoy index with metric = cosine, n_trees = 50

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

## [----|----|----|----|----|----|----|----|----|----|

## **************************************************|
## 11:39:01 Writing NN index file to temp file /var/folders/39/4_ycgqx97cn_x4d12skq1bv4d4jbsj/T//RtmpQj
## 11:39:01 Searching Annoy index using 1 thread, search_k = 3000
## 11:39:01 Annoy recall = 100%
## 11:39:01 Commencing smooth kNN distance calibration using 1 thread with target n_neighbors = 30
## 11:39:01 Initializing from normalized Laplacian + noise (using irlba)
## 11:39:01 Commencing optimization for 500 epochs, with 15820 positive edges
## 11:39:02 Optimization finished
```

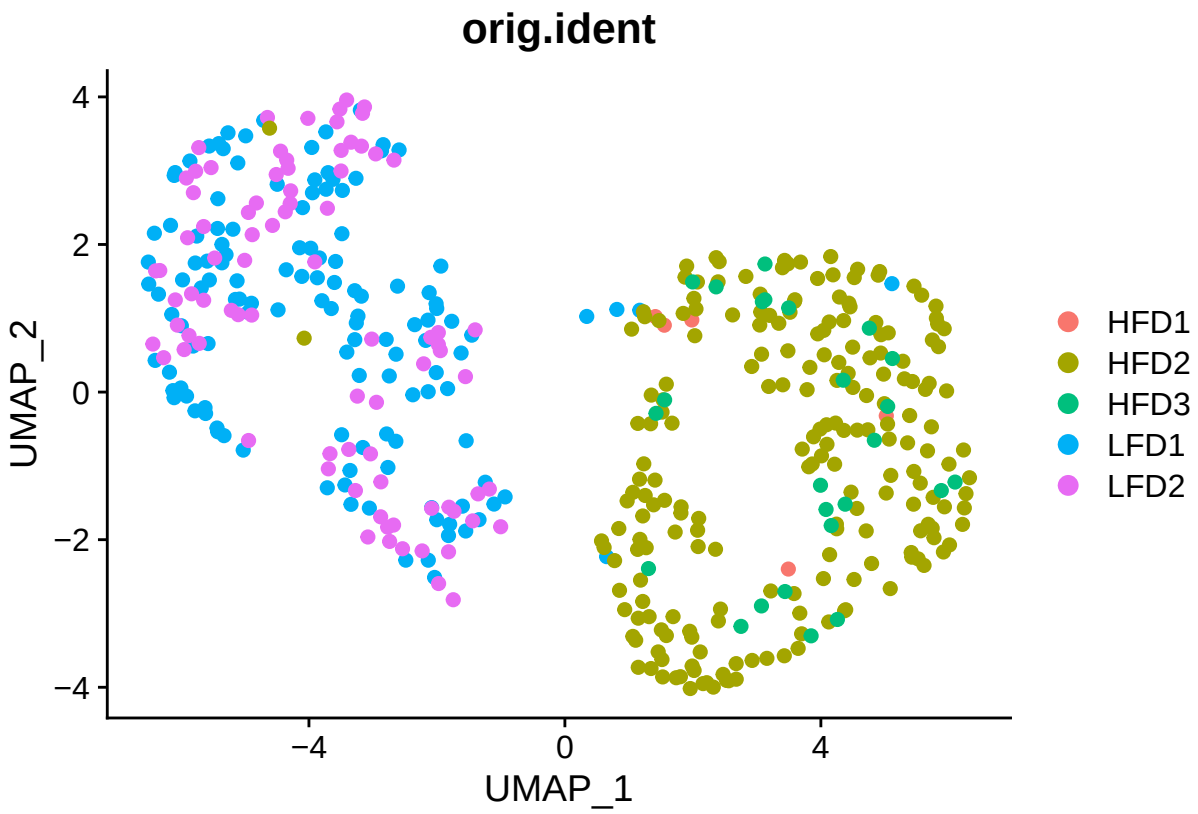
```r
gse <- RunTSNE(gse, dims = 1:4)

# note that you can set `label = TRUE` or use the LabelClusters function to help label
# individual clusters
DimPlot(gse, reduction = "umap", pt.size = 2)
```



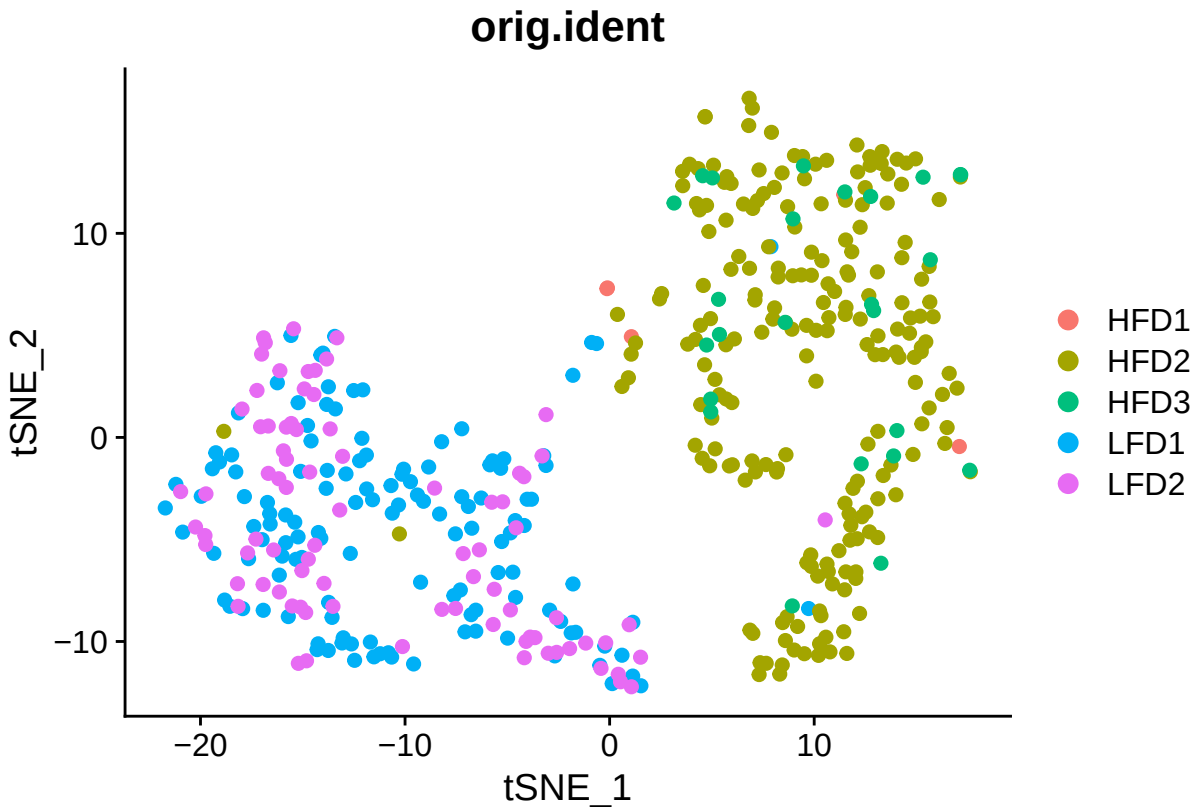

```
DimPlot(gse, reduction = "tsne", pt.size = 2)
```



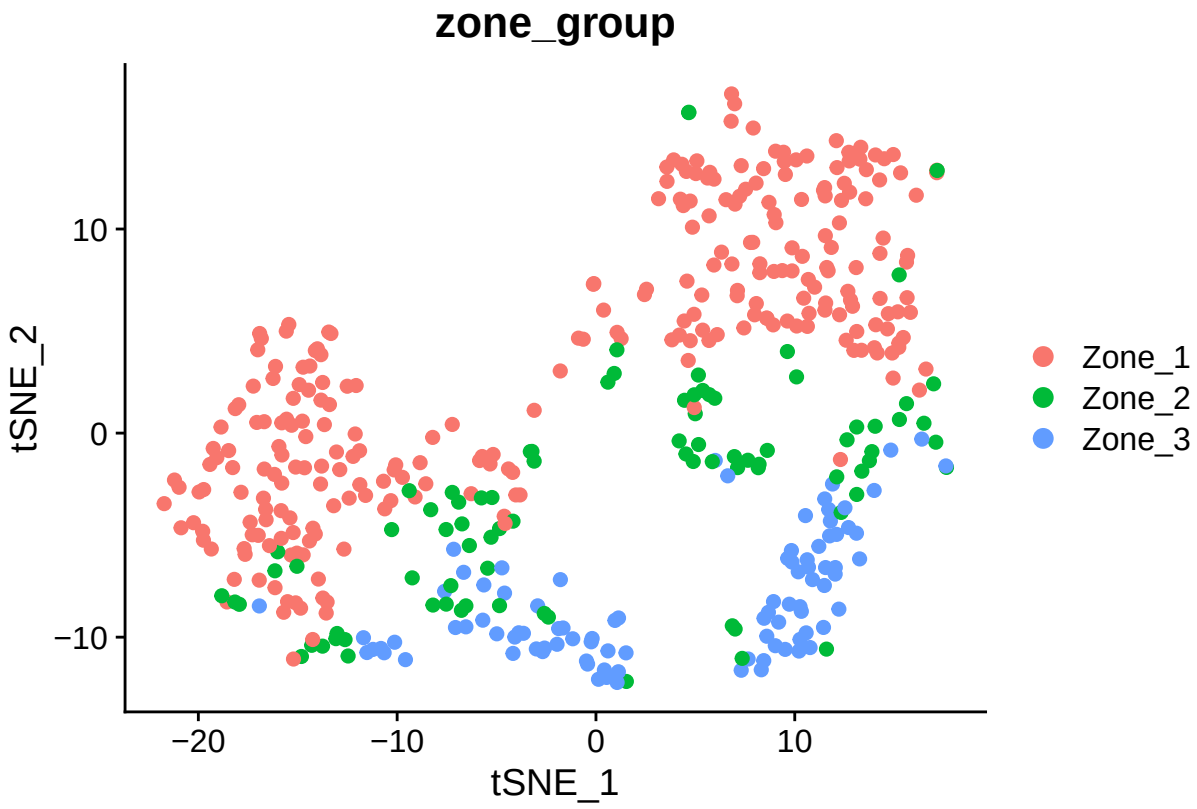
```
DimPlot(gse, reduction = "umap", group.by = "orig.ident", pt.size = 2)
```




```
DimPlot(gse, reduction = "tsne", group.by = "orig.ident", pt.size = 2)
```



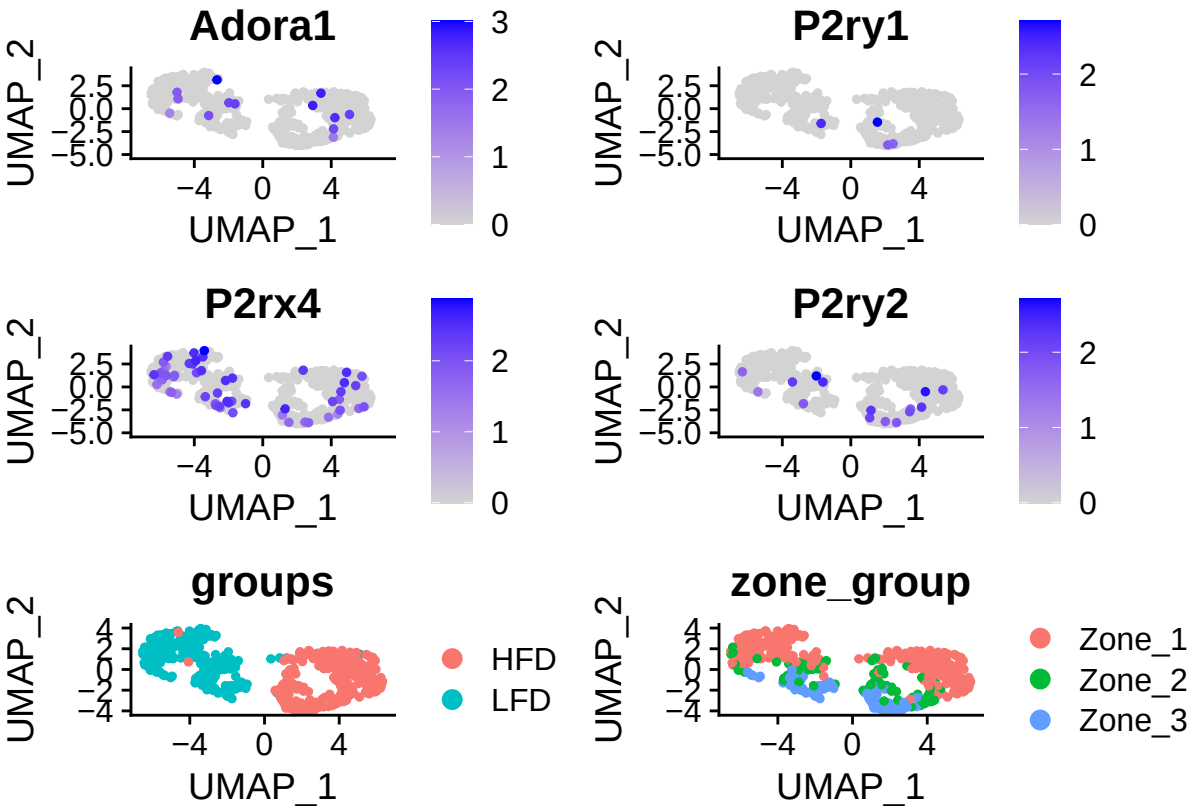

```
DimPlot(gse, reduction = "tsne", group.by = "zone_group", pt.size = 2)
```



Get features of interest

```
features = c(grep("P2RX|P2RY|ADORA1|ADORA2B|ADORA3",rownames(gse@assays$RNA@data),ignore.case = T, valu
```

```
umap_diet.p <- DimPlot(gse, reduction = "umap", group.by = "groups", pt.size = 1)
umap_zone.p <- DimPlot(gse, reduction = "umap", group.by = "zone_group", pt.size = 1)

feat.p <- FeaturePlot(object = gse, features = features, , pt.size = 1, order = TRUE)
print(feat.p + umap_diet.p + umap_zone.p)
```




```
ggsave2(filename = "GSE157281_umap_genes_of_int.png", plot = feat.p + umap_diet.p + umap_zone.p, width =
```

Find differentially expressed features (genes) across clusters (HFD vs LFD)

```
# find diff expressed genes in cluster 0 vs cluster 1
cluster0_vs_1.deg <- FindMarkers(gse,
                                 ident.1 = 0,
                                 ident.2 = 1,
                                 min.pct = 0,
                                 logfc.threshold = 0,
                                 return.thresh = 1,
                                 )
head(cluster0_vs_1.deg, n = 50)
```

```
##                  p_val avg_log2FC pct.1 pct.2    p_val_adj
## Cyp3a11   9.009109e-70 -2.6417452 0.183 0.971 8.610005e-66
## Selenbp2  2.009112e-59 -2.8794132 0.062 0.828 1.920108e-55
## Cyp2c70   1.383226e-37 -1.6541498 0.250 0.833 1.321949e-33
## Ces2a     2.863656e-30 -1.5944789 0.212 0.737 2.736796e-26
## Orm1      3.505133e-28 -0.8431652 0.654 0.986 3.349856e-24
## Gstp1     4.215748e-27 -0.7882024 0.683 0.971 4.028991e-23
## Mup7      1.468779e-24 -0.7750625 0.658 0.986 1.403712e-20
## Upp2      1.931931e-22 -1.5138133 0.100 0.536 1.846346e-18
## Cyp7b1    2.321539e-22 -1.2223012 0.296 0.732 2.218695e-18
## Fabp1     2.829128e-21  0.2644758 1.000 1.000 2.703797e-17
## Apoa4     5.353821e-21  1.7571015 0.558 0.163 5.116647e-17
## Nudt7     2.116697e-20 -0.8213059 0.554 0.904 2.022927e-16
## Mup11     8.557097e-20 -0.6147867 0.754 1.000 8.178017e-16
## Aldob     3.122492e-19  0.3063153 0.996 0.990 2.984165e-15
## C8b       5.156696e-18 -1.1216919 0.271 0.684 4.928254e-14
## Cyp4a14   1.447945e-17  1.6887837 0.421 0.067 1.383801e-13
## Egfr      2.743505e-17 -1.2161412 0.200 0.603 2.621967e-13
## G0s2      7.123477e-17  1.5228538 0.508 0.148 6.807907e-13
## Mup16     1.022987e-16 -1.0500000 0.329 0.708 9.776685e-13
## Hmgcs2    1.003475e-14  0.3848374 0.929 0.890 9.590212e-11
## Mup17     1.063560e-14 -0.7505481 0.579 0.947 1.016444e-10
## Gm6484    6.729865e-14  1.6517478 0.333 0.053 6.431732e-10
## Mat1a     9.487540e-14  0.3197482 0.971 0.962 9.067242e-10
## Dbi       1.096272e-13  0.2272108 1.000 1.000 1.047707e-09
## Inmt      1.340209e-13 -0.8038378 0.417 0.785 1.280837e-09
## Aldh1l1   1.603801e-13 -0.5736704 0.667 0.914 1.532753e-09
## Hsd3b5    1.990356e-13 -1.1687999 0.154 0.474 1.902184e-09
## Onecut1   9.452284e-13 -1.2360822 0.042 0.292 9.033548e-09
## Leap2     4.036487e-12  0.7719259 0.738 0.545 3.857671e-08
## Igfbp2    4.621209e-12 -0.9123304 0.288 0.612 4.416490e-08
## Apoa2     1.071205e-11  0.1762849 1.000 1.000 1.023751e-07
## Cyp4a12a  1.076958e-11 -1.1275147 0.071 0.330 1.029248e-07
## Mup1      1.743155e-11 -0.6169517 0.613 0.876 1.665934e-07
## Hsd17b2   2.412525e-11 -0.9577865 0.238 0.536 2.305650e-07
## Elovl5    2.448648e-11  1.1436913 0.554 0.311 2.340173e-07
## Alas1     2.667472e-11 -0.9489730 0.188 0.498 2.549303e-07
## Alb       3.348843e-11  0.1867395 1.000 1.000 3.200489e-07
## Csad      5.467095e-11  1.1178980 0.550 0.316 5.224903e-07
## Atp5d     5.472966e-11 -0.6623681 0.462 0.756 5.230513e-07
## Mup21     7.369106e-11  0.5907422 0.783 0.766 7.042655e-07
## Ang       1.291413e-10 -0.8869050 0.221 0.526 1.234203e-06
## Atp5k     1.576488e-10  0.3330863 0.912 0.876 1.506649e-06
## Cyp2c29   1.966791e-10 -0.7464290 0.404 0.732 1.879662e-06
## Serpina3k 2.220456e-10 -0.1670691 1.000 1.000 2.122090e-06
## Cyb5r3    3.720973e-10 -0.5544899 0.592 0.904 3.556133e-06
## Ces3b     4.867581e-10 -0.8712559 0.254 0.545 4.651947e-06
## Crot      4.959537e-10  0.5273290 0.821 0.703 4.739829e-06
## Serpina1e 5.353983e-10 -0.3931649 0.846 0.981 5.116802e-06
## Ttr       5.460310e-10  0.1667793 1.000 1.000 5.218418e-06
## Fbp1      5.510824e-10  0.3054372 0.938 0.904 5.266695e-06
```

```
# Save resutls to a file
write.table(x = cluster0_vs_1.deg,
```

```
            file = "GSE157281_HFD_vs_LFD.txt",
            sep = "\t", col.names = NA)
```

```
sessionInfo()
```

```
## R version 4.2.2 (2022-10-31)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Ventura 13.2.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.2-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] plot3D_1.4         cowplot_1.1.1      lubridate_1.9.2    forcats_1.0.0
##  [5] stringr_1.5.0      dplyr_1.1.1        purrr_1.0.1        readr_2.1.4
##  [9] tidyr_1.3.0        tibble_3.2.1       ggplot2_3.4.2      tidyverse_2.0.0
## [13] SeuratObject_4.1.3 Seurat_4.3.0
##
## loaded via a namespace (and not attached):
##   [1] Rtsne_0.16             colorspace_2.1-0       deldir_1.0-6
##   [4] ellipsis_0.3.2         ggridges_0.5.4         rstudioapi_0.14
##   [7] spatstat.data_3.0-1    farver_2.1.1           leiden_0.4.3
##  [10] listenv_0.9.0          ggrepel_0.9.3          fansi_1.0.4
##  [13] codetools_0.2-18       splines_4.2.2          knitr_1.42
##  [16] polyclip_1.10-4        jsonlite_1.8.4         ica_1.0-3
##  [19] cluster_2.1.4          png_0.1-8              uwot_0.1.14
##  [22] shiny_1.7.4            sctransform_0.3.5      spatstat.sparse_3.0-1
##  [25] compiler_4.2.2         httr_1.4.5             Matrix_1.5-4
##  [28] fastmap_1.1.1          lazyeval_0.2.2         limma_3.54.2
##  [31] cli_3.6.1              later_1.3.0            htmltools_0.5.5
##  [34] tools_4.2.2            misc3d_0.9-1           igraph_1.4.2
##  [37] gtable_0.3.3           glue_1.6.2             RANN_2.6.1
##  [40] reshape2_1.4.4         Rcpp_1.0.10            scattermore_0.8
##  [43] vctrs_0.6.2            spatstat.explore_3.1-0 nlme_3.1-160
##  [46] progressr_0.13.0       lmtest_0.9-40          spatstat.random_3.1-4
##  [49] xfun_0.39              globals_0.16.2         timechange_0.2.0
##  [52] mime_0.12              miniUI_0.1.1.1         lifecycle_1.0.3
##  [55] irlba_2.3.5.1          goftest_1.2-3          future_1.32.0
##  [58] MASS_7.3-58.1          zoo_1.8-12             scales_1.2.1
##  [61] ragg_1.2.5             hms_1.1.3              promises_1.2.0.1
##  [64] spatstat.utils_3.0-2   parallel_4.2.2         RColorBrewer_1.1-3
##  [67] yaml_2.3.7             reticulate_1.28        pbapply_1.7-0
##  [70] gridExtra_2.3          stringi_1.7.12         highr_0.10
##  [73] systemfonts_1.0.4      rlang_1.1.0            pkgconfig_2.0.3
##  [76] matrixStats_0.63.0     evaluate_0.20          lattice_0.20-45
##  [79] ROCR_1.0-11            tensor_1.5             labeling_0.4.2
```

```
##  [82] patchwork_1.1.2        htmlwidgets_1.6.2      tidyselect_1.2.0
##  [85] parallelly_1.35.0      RcppAnnoy_0.0.20       plyr_1.8.8
##  [88] magrittr_2.0.3         R6_2.5.1               generics_0.1.3
##  [91] DBI_1.1.3              withr_2.5.0            pillar_1.9.0
##  [94] fitdistrplus_1.1-8     survival_3.4-0         abind_1.4-5
##  [97] sp_1.6-0               future.apply_1.10.0    KernSmooth_2.23-20
## [100] utf8_1.2.3             spatstat.geom_3.1-0    plotly_4.10.1
## [103] tzdb_0.3.0             rmarkdown_2.21         grid_4.2.2
## [106] data.table_1.14.8      digest_0.6.31          xtable_1.8-4
## [109] httpuv_1.6.9           textshaping_0.3.6      munsell_0.5.0
## [112] viridisLite_0.4.1      tcltk_4.2.2
```