

Results

programandoconro

7/19/2019

```
##### Required packages #####

library(dplyr)
library(Seurat)
library(ggplot2)
library(cowplot)

##### Exp_1 #####

##Read, normalize and find features in the data

R048_1 <- Read10X(data.dir = "~/Dropbox/DataScience/Fiver/R048_1/")%>%
  CreateSeuratObject(min.cells = 30, min.features = 2000,project = "R048 Exp.1")%>%
  NormalizeData(verbose = FALSE)%>%
  FindVariableFeatures(selection.method = "vst", nfeatures = 2000)

DMSO_1 <- Read10X(data.dir = "~/Dropbox/DataScience/Fiver/DMSO_1/")%>%
  CreateSeuratObject(min.cells = 30, min.features = 2000,project = "DMSO Exp.1")%>%
  NormalizeData(verbose = FALSE)%>%
  FindVariableFeatures(selection.method = "vst", nfeatures = 2000)

# Create variables for grouping later

R048_1$R048_1 <- 'R048_1'
DMSO_1$DMSO_1 <- 'DMSO_1'

# Integrate data

exp_1 <- FindIntegrationAnchors(object.list = list(DMSO_1,R048_1), dims = 1:20)%>%
  IntegrateData(dims = 1:20)

#Remove objects to save RAM

rm(DMSO_1,R048_1)

# PCA, t-SNE and Cluster in one pipe

DefaultAssay(exp_1) <- "integrated"

exp_1 <-ScaleData(exp_1,verbose = FALSE)%>%
  RunPCA(npcs = 30, verbose = FALSE)%>%
  RunUMAP(reduction = "pca", dims = 1:20)%>%
  FindNeighbors( reduction = "pca", dims = 1:20)%>%
  FindClusters( resolution = 0.5)

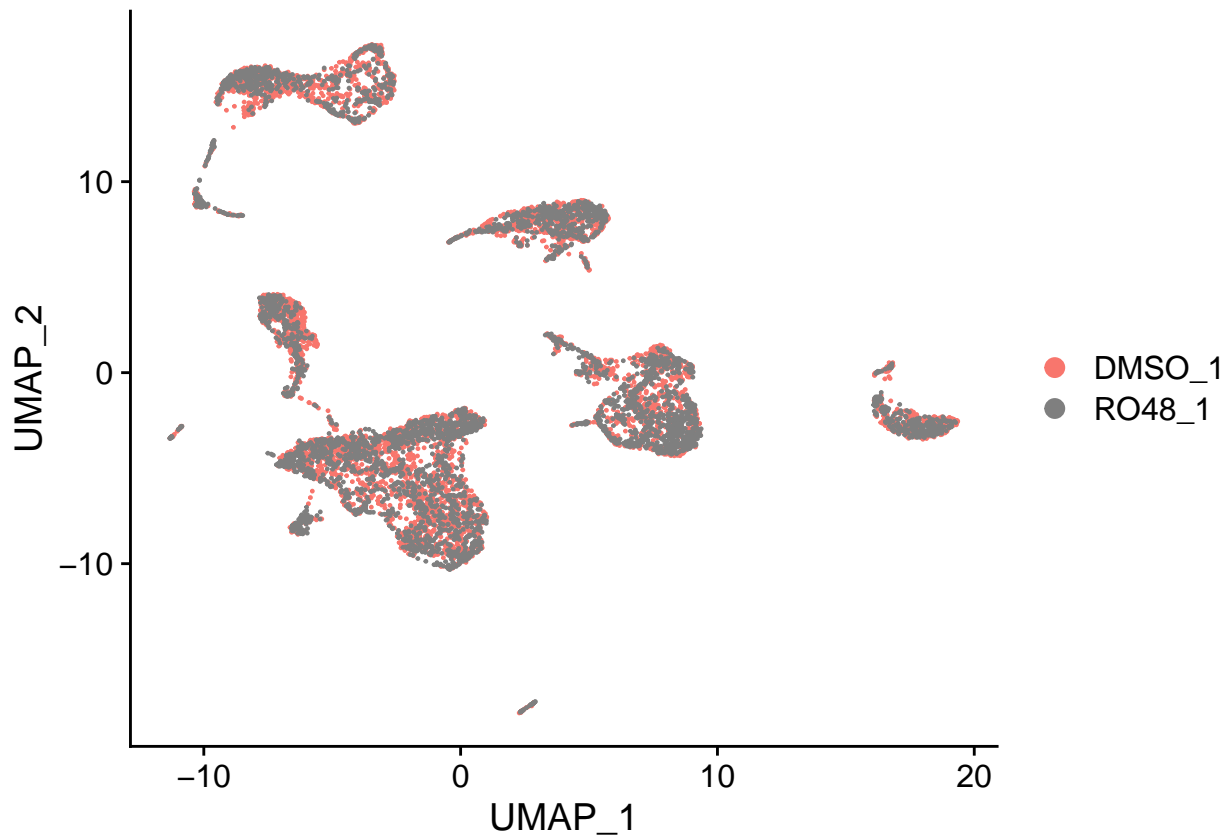
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
```

```
##
## Number of nodes: 8988
## Number of edges: 315515
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9331
## Number of communities: 18
## Elapsed time: 0 seconds
```

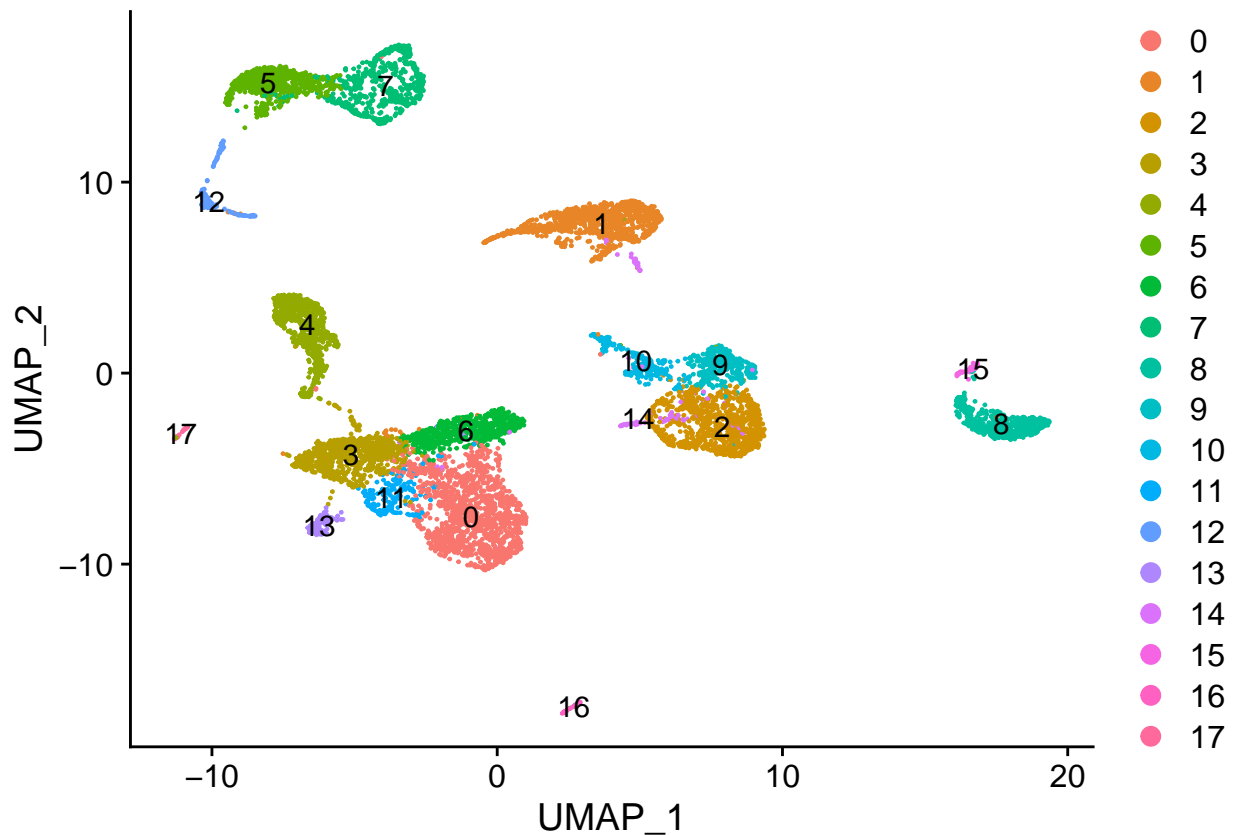
```
# Visualization
```

```
g1 <- DimPlot(exp_1, reduction = "umap", group.by = 'DMSO_1')
```

```
g1 + scale_color_discrete(labels=c('DMSO_1', 'R048_1'))
```



```
DimPlot(exp_1, reduction = "umap", label = TRUE)
```



```
rm(exp_1)

##### Exp_2 #####

##Read, normalize and find features in the data

R048_2 <- Read10X(data.dir = "~/Dropbox/DataScience/Fiver/R048_2/")%>%
  CreateSeuratObject(min.cells = 30, min.features = 2000,project = "R048 Exp.2")%>%
  NormalizeData(verbose = FALSE)%>%
  FindVariableFeatures(selection.method = "vst", nfeatures = 2000)

DMSO_2 <- Read10X(data.dir = "~/Dropbox/DataScience/Fiver/DMSO_2/")%>%
  CreateSeuratObject(min.cells = 30, min.features = 2000,project = "DMSO Exp.2")%>%
  NormalizeData(verbose = FALSE)%>%
  FindVariableFeatures(selection.method = "vst", nfeatures = 2000)

# Create variables for grouping later

R048_2$R048_2 <- 'R048_2'
DMSO_2$DMSO_2 <- 'DMSO_2'

# Integrate data

exp_2 <- FindIntegrationAnchors(object.list = list(DMSO_2,R048_2), dims = 1:20)%>%
  IntegrateData(dims = 1:20)

#Remove objects to save RAM
```

```

rm(DMSO_2,R048_2)

# PCA, t-SNE and Cluster in one pipe

DefaultAssay(exp_2) <- "integrated"

exp_2 <- ScaleData(exp_2, verbose = FALSE)%>%
  RunPCA(npcs = 30, verbose = FALSE)%>%
  RunUMAP(reduction = "pca", dims = 1:20)%>%
  FindNeighbors( reduction = "pca", dims = 1:20)%>%
  FindClusters( resolution = 0.5)

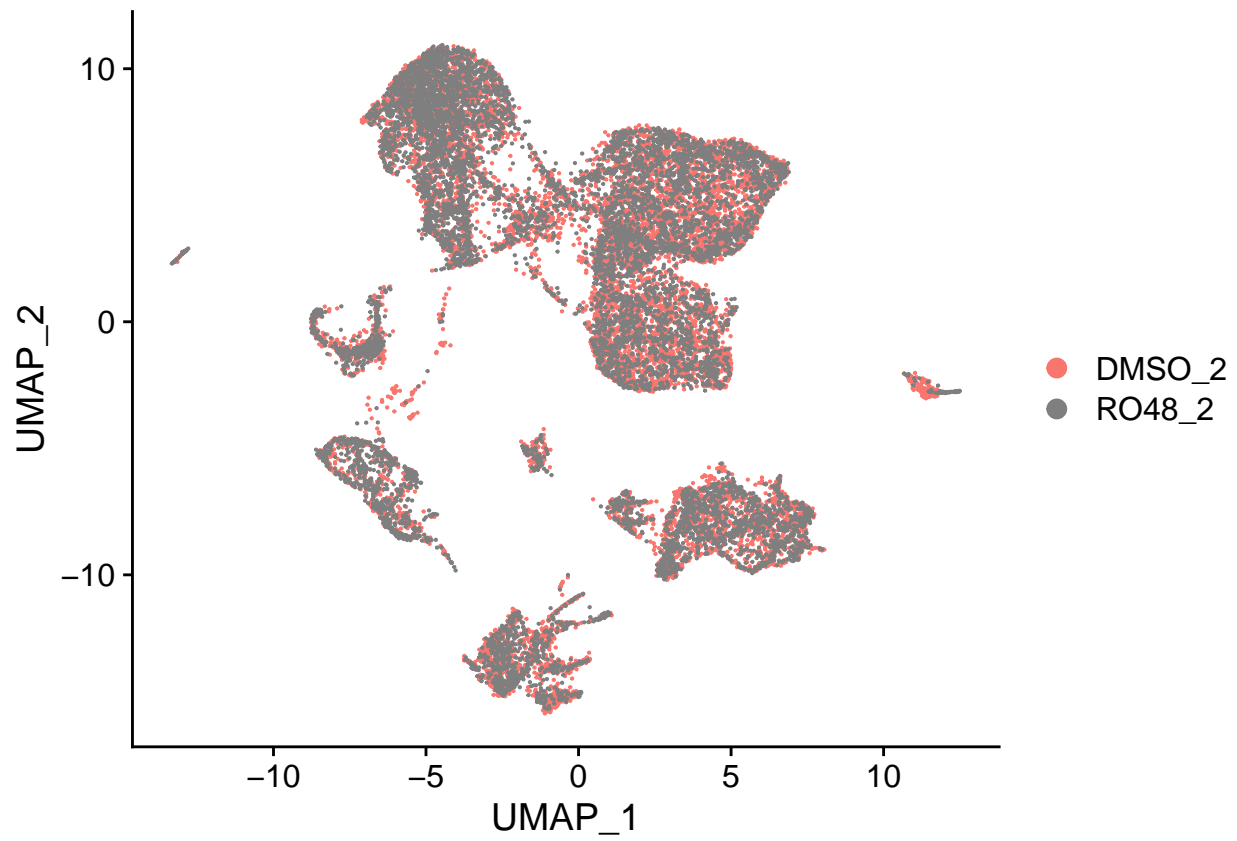
## Modularity Optimizer version 1.3.0 by Ludo Waltman and Nees Jan van Eck
##
## Number of nodes: 19093
## Number of edges: 666556
##
## Running Louvain algorithm...
## Maximum modularity in 10 random starts: 0.9189
## Number of communities: 18
## Elapsed time: 2 seconds

# Visualization

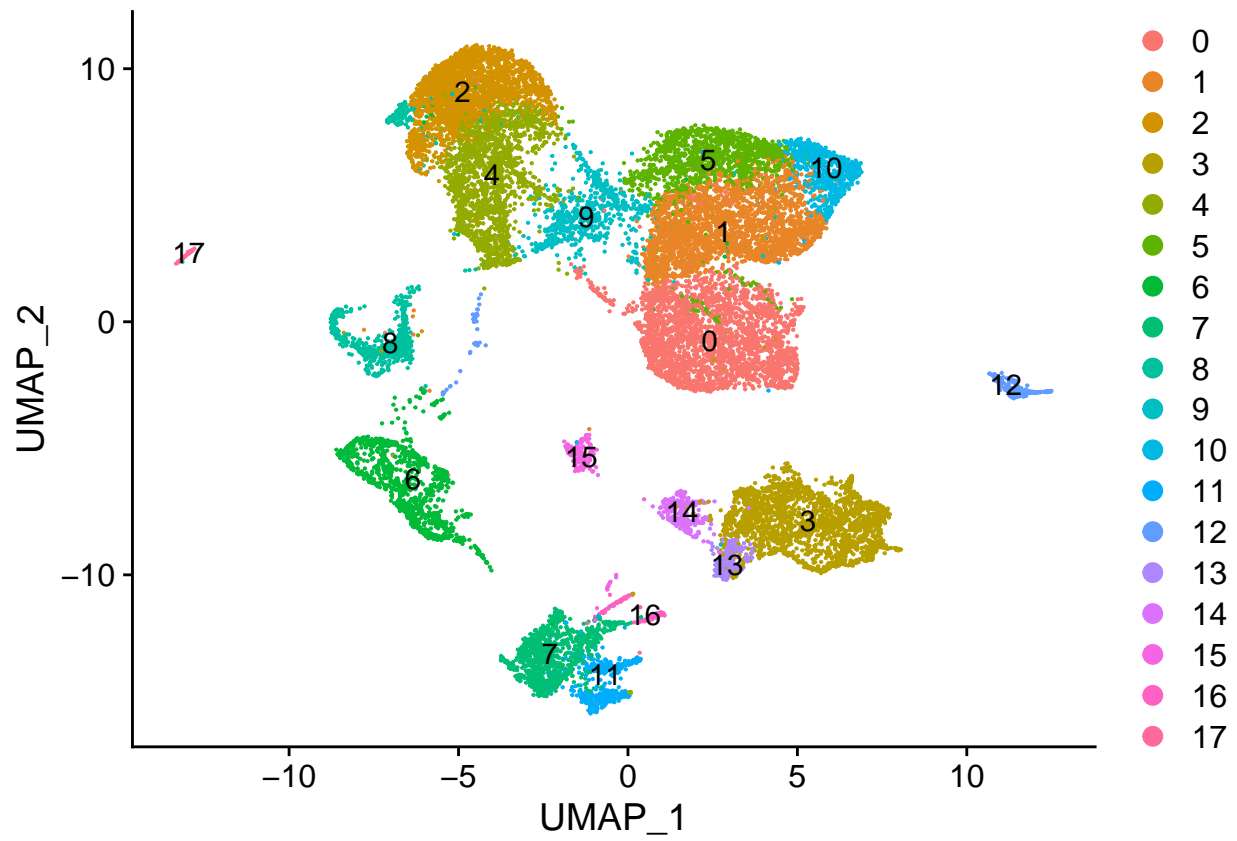
g2 <- DimPlot(exp_2, reduction = "umap", group.by = 'DMSO_2')

g2 + scale_color_discrete(labels=c('DMSO_2','R048_2'))

```



```
DimPlot(exp_2, reduction = "umap", label = TRUE)
```



```
rm(exp_2)
```

```
#####
```