

Single-cell RNA sequencing analysis of stretched cells

Allison M. Jevitt

2022-05-05

Analysis performed using the following software and package versions

R version 4.1.2 (2021-11-01) – “Bird Hippie”
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

Seurat 4.0.3
ggplot2 3.3.5
sctransform 0.3.2
viridis 0.6.1
viridisLite 0.4.0

Setting up

First, access the raw sequencing data by navigating to [Github](#). Instructions are provided in the README.md file to describe how download the correct data files from the GEO database and format the files so that the following script will run smoothly. This will result in three data files in a directory called “Data”.

Second, to ensure proper file structure, create a directory called “Markers” within your working directory for storing output files created during the analysis.

Finally, make sure this script, and the “Data” and “Markers” directories are all within a single working directory. Set this directory as your working directory with the “setwd()” command.

Load libraries and set options

```
# load libraries
library(Seurat)
library(ggplot2)
library(sctransform)
library(viridis)

# set options to allow larger global variable size
options(future.globals.maxSize = 4000000000000000000)
```

Load data and create a Seurat object

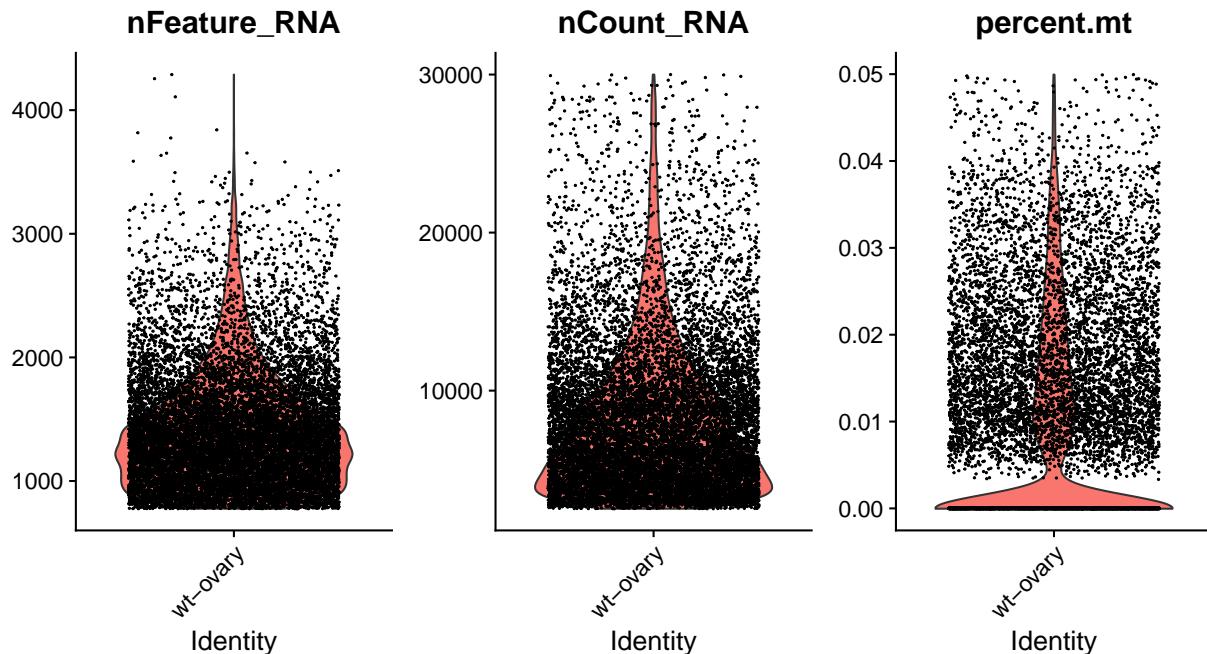
```
# load 10X sequencing data and create a Seurat object called data
read.data <- Read10X(data.dir = "Data")
data <- CreateSeuratObject(counts = read.data,
                           project = "wt-ovary",
                           min.cells = 3,
                           min.features = 200)
```

Quality control and select high quality cells for downstream analysis

```
# store mitochondrial percentage in object meta data in the Seurat object
data <- PercentageFeatureSet(data, pattern = "-m", col.name = "percent.mt")

# subset data to select high quality cells
data.sub <- subset(data,
                     subset = nFeature_RNA > 775
                     & nFeature_RNA < 55000
                     & nCount_RNA < 30000
                     & percent.mt < 0.05)

# plot QC metrics
VlnPlot(data.sub, features = c("nFeature_RNA",
                                "nCount_RNA",
                                "percent.mt"),
        ncol = 3)
```



SCTransform

```
# SCTransform -> replaces NormalizeData(), ScaleData(), and FindVariableFeatures()
data.SC <- SCTransform(data.sub,
                       vars.to.regress = c("percent.mt",
                                           "nCount_RNA"),
                       verbose = FALSE)
```

Dimensionality reduction using PCA and plot UMAP

```
# PCA
data.PCA <- RunPCA(data.SC,
```

```

        npcs = 50,
        verbose = FALSE)

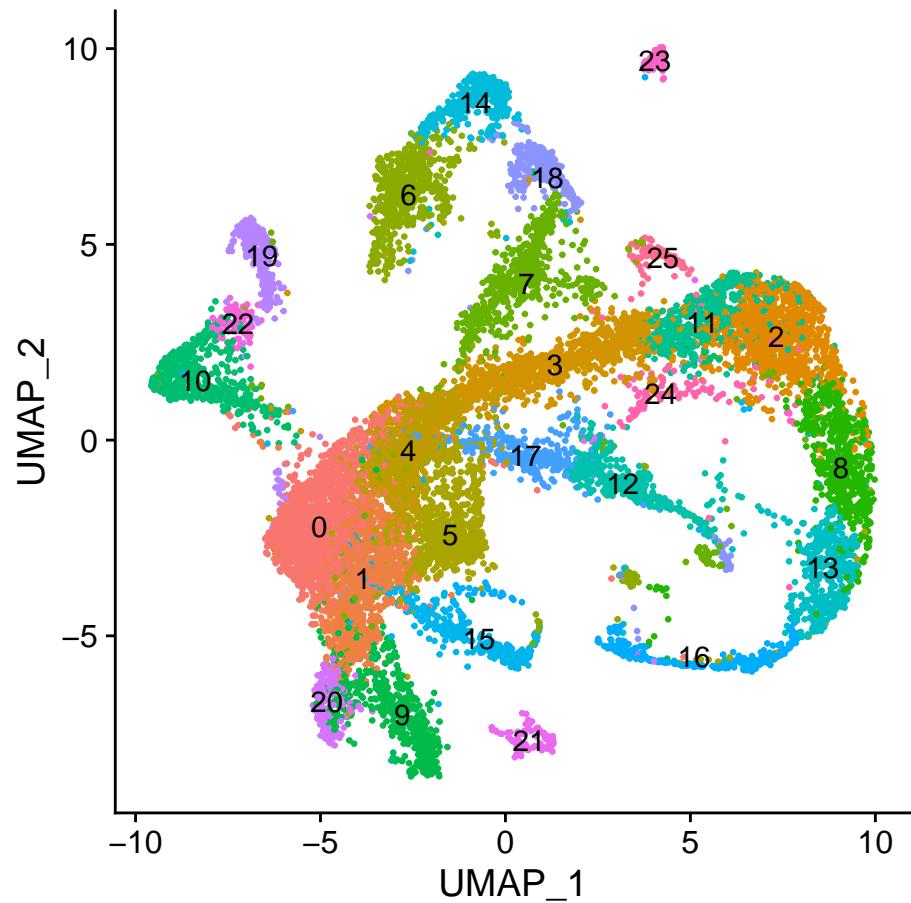
# UMAP
data.UMAP <- RunUMAP(data.PCA,
                      dims = 1:20,
                      verbose = FALSE)

data.nbrs <- FindNeighbors(data.UMAP,
                            dims = 1:20,
                            verbose = FALSE)

data <- FindClusters(data.nbrs,
                     resolution = 1,
                     verbose = FALSE)

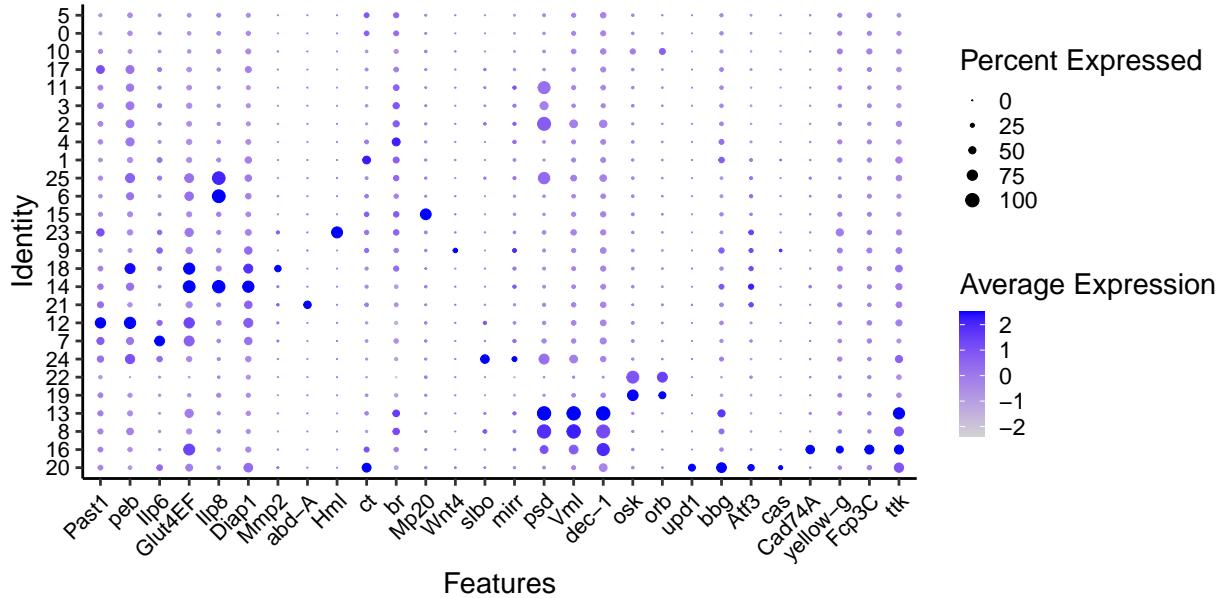
# plot UMAP
DimPlot(data,
        label = TRUE,
        pt.size = 0.5) + NoLegend()

```



Identify clusters using established marker genes and group similar clusters

```
# define the markers used to identify cell type and merge similar clusters
markers.to.plot <- c("Past1",
  "peb",
  "Ilp6",
  "Glut4EF",
  "Ilp8",
  "Diap1",
  "Mmp2",
  "abd-A",
  "Hml",
  "ct",
  "br",
  "Mp20",
  "Wnt4",
  "slbo",
  "mirr",
  "psd",
  "Vml",
  "dec-1",
  "osk",
  "orb",
  "upd1",
  "bbg",
  "Atf3",
  "cas",
  "Cad74A",
  "yellow-g",
  "Fcp3C",
  "ttk"
)
# make a dotplot of all clusters showing level of marker gene expression
DotPlot(data,
  features = markers.to.plot,
  cols = c("lightgrey", "blue"),
  cluster.idents = TRUE,
  dot.scale = 4) +
  RotatedAxis() +
  theme_classic(base_size = 20,) +
  theme(axis.text.x = element_text(angle = 45,
    hjust = 1,
    size = 15,
    color = "black")) +
  theme(axis.text.y = element_text(hjust = 1,
    size = 15,
    color = "black"))
```

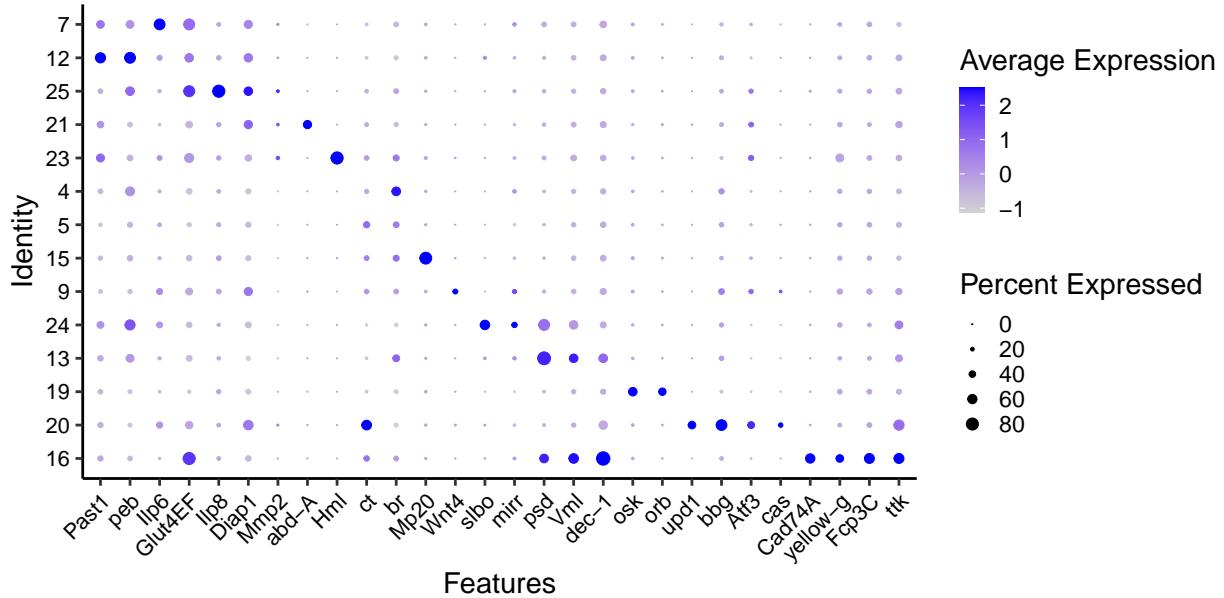


```

# merge clusters of similar type (identified from marker dotplot above)
data.merge <- RenameIdents(object = data, '10' = '22') # Germline
data.merge <- RenameIdents(object = data.merge, '22' = '19') # Germline
data.merge <- RenameIdents(object = data.merge, '3' = '11') # Vitellogenetic FCs
data.merge <- RenameIdents(object = data.merge, '0' = '1') # Mitotic FCs
data.merge <- RenameIdents(object = data.merge, '1' = '5') # Mitotic FCs
data.merge <- RenameIdents(object = data.merge, '11' = '2') # Vitellogenetic FCs
data.merge <- RenameIdents(object = data.merge, '2' = '8') # Vitellogenetic FCs
data.merge <- RenameIdents(object = data.merge, '8' = '13') # Vitellogenetic FCs
data.merge <- RenameIdents(object = data.merge, '6' = '14') # Pre-CL and CL Cells
data.merge <- RenameIdents(object = data.merge, '14' = '18') # Pre-CL and CL Cells
data.merge <- RenameIdents(object = data.merge, '18' = '25') # Pre-CL and CL Cells
data.merge <- RenameIdents(object = data.merge, '17' = '12') # Stretched Cells

# generate a dotplot with marker gene expression per cluster (after merging)
DotPlot(data.merge,
        features = markers.to.plot,
        cols = c("lightgrey", "blue"),
        cluster.idents = TRUE,
        dot.scale = 4) +
  RotatedAxis() +
  theme_classic(base_size = 20,) +
  theme(axis.text.x = element_text(angle = 45,
                                    hjust = 1,
                                    size = 15,
                                    color = "black"))+
  theme(axis.text.y = element_text(hjust = 1,
                                    size = 15,
                                    color = "black"))

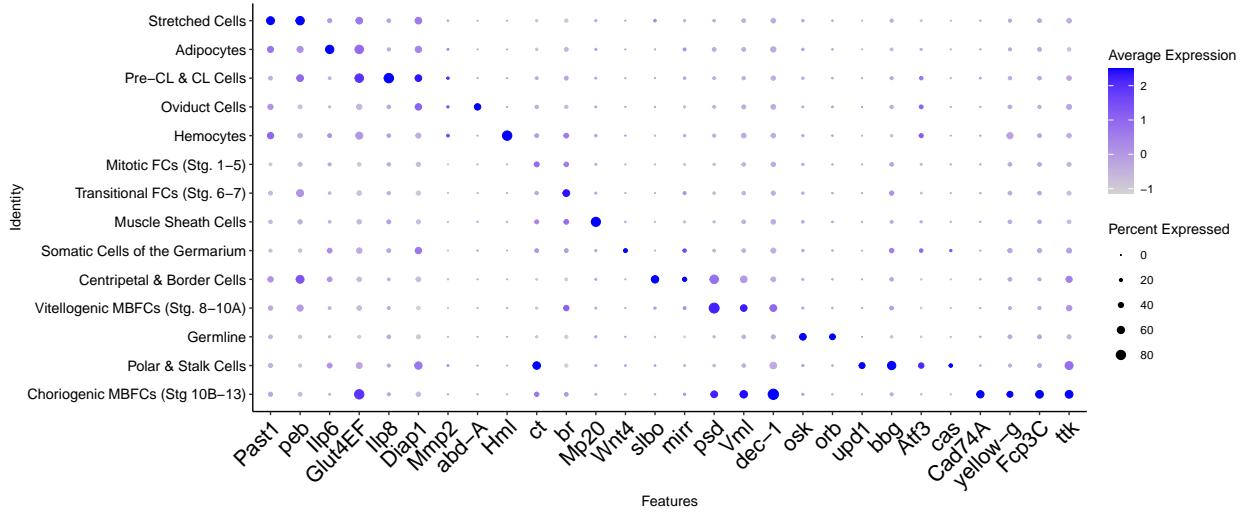
```



Rename clusters based on cell-type identity

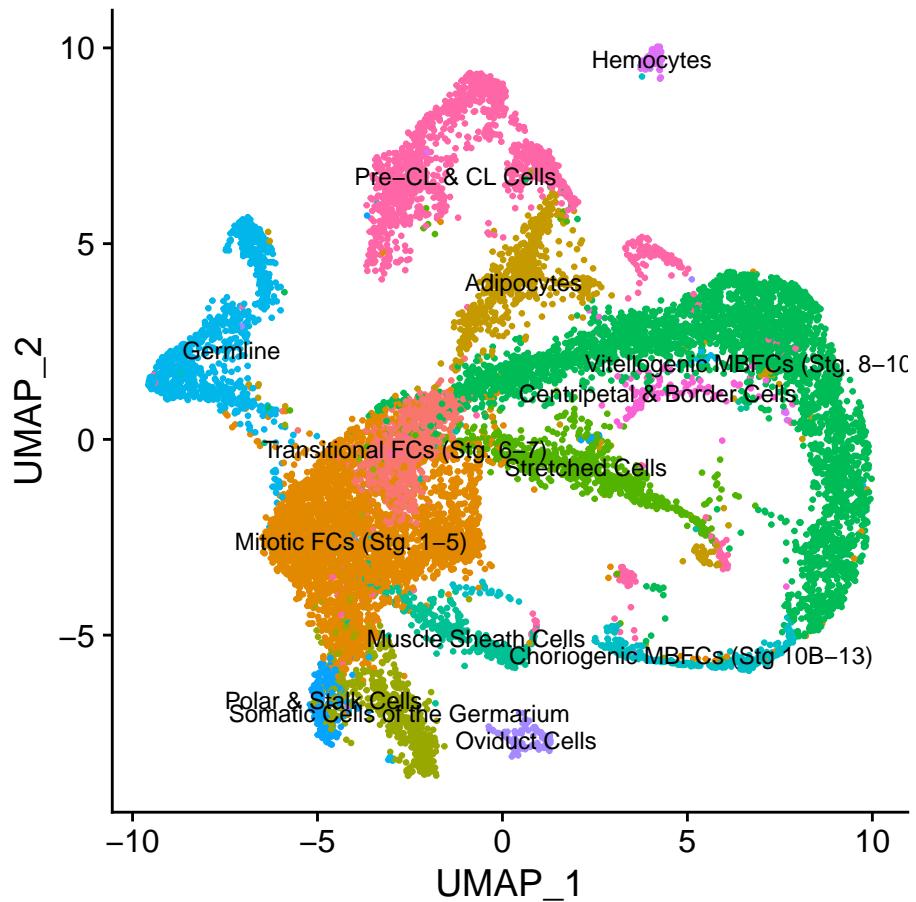
```
# rename each cluster based on gene expression patterns in previous dotplot
data.ID <- RenameIds(data.merge,
  `4` = "Transitional FCs (Stg. 6-7)",
  `5` = "Mitotic FCs (Stg. 1-5)",
  `7` = "Adipocytes",
  `9` = "Somatic Cells of the Germarium",
  `12` = "Stretched Cells",
  `13` = "Vitellogenetic MBFCs (Stg. 8-10A)",
  `15` = "Muscle Sheath Cells",
  `16` = "Choriogenic MBFCs (Stg 10B-13)",
  `19` = "Germline",
  `20` = "Polar & Stalk Cells",
  `21` = "Oviduct Cells",
  `23` = "Hemocytes",
  `24` = "Centripetal & Border Cells",
  `25` = "Pre-CL & CL Cells")

# generate dotplot with named clusters
DotPlot(data.ID, features = markers.to.plot,
  cols = c("lightgrey", "blue"),
  cluster.idents = TRUE,
  dot.scale = 3) +
  RotatedAxis() +
  theme_classic(base_size = 10,) +
  theme(axis.text.x = element_text(angle = 45,
    hjust = 1,
    size = 15,
    color = "black")) +
  theme(axis.text.y = element_text( hjust = 1,
    size = 10,
    color = "black"))
```



Final UMAP plot

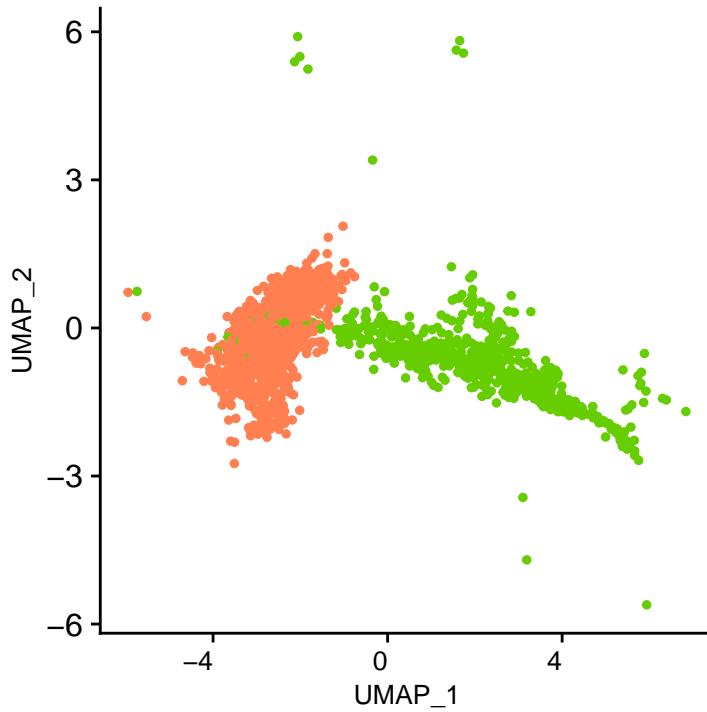
```
# generate final UMAP plot with cluster ID labels
DimPlot(data.ID,
         label = TRUE,
         pt.size = 0.5,
         label.size = 3) + NoLegend()
```



Create subset of data to focus just on cells undergoing the cuboidal-to-squamous transition

```
# subset data to only include Transitional FCs and Stretched cells
data.sub <- subset(data.ID,
                    idents = c("Stretched Cells",
                              "Transitional FCs (Stg. 6-7)"))

# plot UMAP for data subset with matching cluster colors
DimPlot(data.sub,
        label = FALSE,
        pt.size = 1,
        cols = c("coral", "chartreuse3"),
        label.size = 9) +
  NoLegend() +
  theme(title= element_text(size = 10),
        axis.text.x=element_text(hjust = 1, size = 10),
        axis.title = element_text(size = 10),
        axis.title.y.right = element_text(size = 3),
        legend.text=element_text(size = 20),
        legend.title=element_text(size = 10),
        axis.line = element_line(size = 0.5))
```



Find marker genes per cluster

```
# find marker genes defining each cluster
data.sub.markers <- FindAllMarkers(data.sub,
                                      min.pct = 0.25,
                                      logfc.threshold = 0.25)

# save a .csv file with markers defining each cluster in "Markers" directory
write.csv(data.sub.markers,
          file = "Markers/markers.csv")
```

Evaluate gene expression profiles and patterns between Stretched Cells and Transitional FCs

```
# differential expression testing
DEmarkers <- FindMarkers(data.sub,
                           ident.1 = "Stretched Cells",
                           ident.2 = "Transitional FCs (Stg. 6-7)")

# save a .csv file with DE results in "Markers" directory
write.csv(DEmarkers,
          file = "Markers/DE-markers.csv")
```

Heatmap plots of specific genes of interest resulting from DE analysis

```
# DoHeatmap
# define features involved in regulating cell shape/ morphogenesis
```

```

features1 <- c("scb",
             "mys",
             "Pax",
             "shg",
             "nrv2",
             "crb",
             "l(2)gl",
             "moody",
             "Fas3",
             "Moe",
             "Inx3",
             "Inx7",
             "ogre",
             "Inx2",
             "CLIP-190",
             "Gie",
             "Hsp83",
             "Jupiter",
             "betaTub60D",
             "Rab11",
             "CG34417",
             "Chd64",
             "Rtnl1",
             "sn",
             "Sep2",
             "Akap200",
             "Tm1",
             "Diap1",
             "Mlc-c",
             "Rac2",
             "Cortactin",
             "zip",
             "CG6891",
             "Fasi",
             "cindr",
             "Rho1",
             "Msp300",
             "Kank",
             "Col4a1",
             "vkg",
             "Plod")

# define features which are relevant transcription factors
features2 <- c("Hr4",
             "luna",
             "Glut4EF",
             "ftz-f1",
             "peb",
             "Six4",
             "Sox14",
             "cbt",
             "kay",
             "aop",

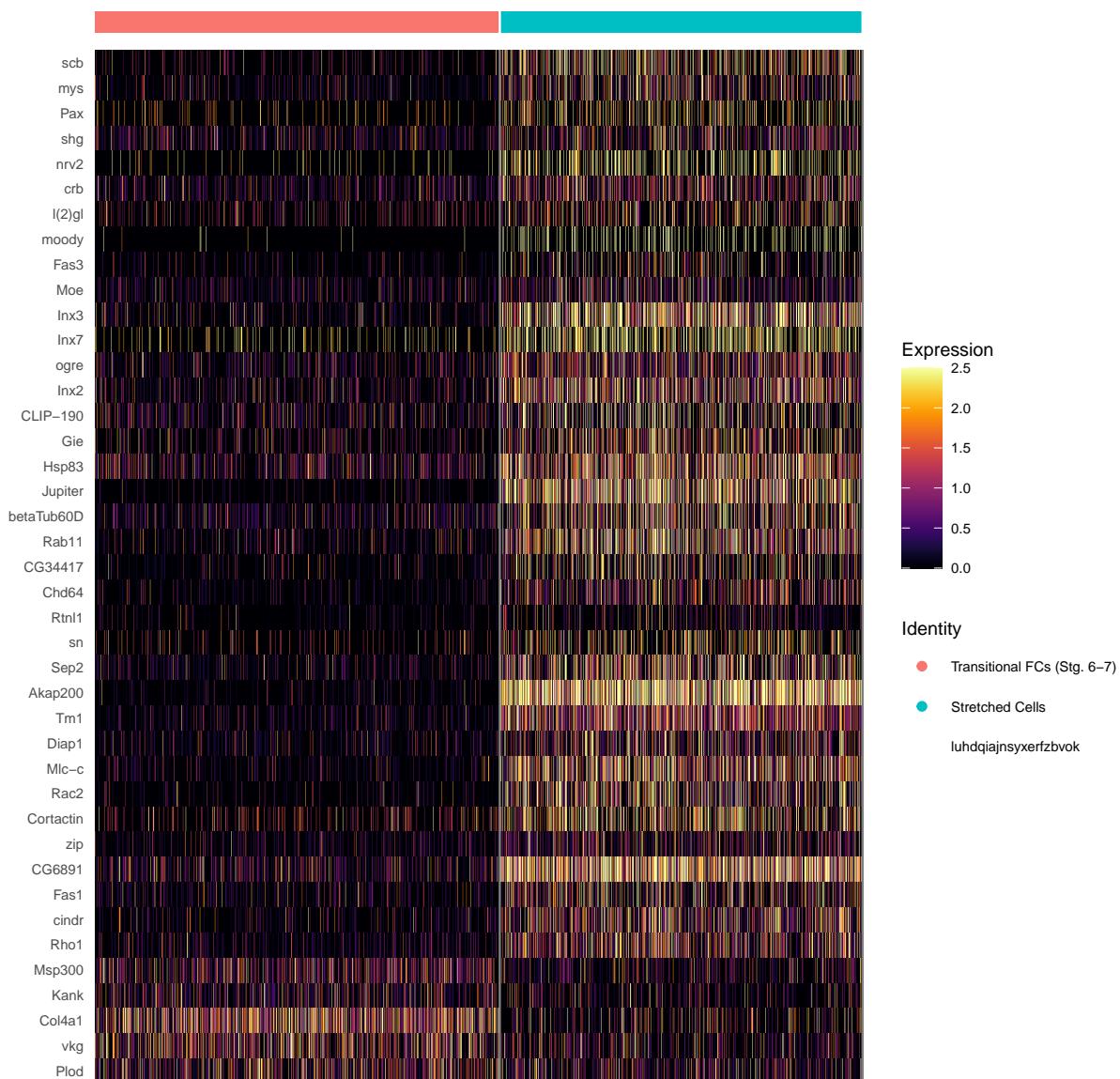
```

```

"apt",
"Hnf4",
"CG9932",
"Usf",
"Eip75B",
"Xbp1",
"E(spl)mbeta-HLH",
"Jra",
"bigmax",
"slbo",
"dl",
"E(spl)m3-HLH",
"Pdp1",
"Myc")

# generate heatmap of cell-shape-related genes
DoHeatmap(data.sub,
           features = features1,
           disp.min = 0,
           size = 8,
           label = FALSE) +
  scale_fill_viridis(option = "inferno") +
  theme(text = element_text(size = 8))

```



```
# generate heatmap of relevant transcription factor genes
DoHeatmap(data.sub,
           features = features2,
           disp.min = 0,
           size = 9,
           label = FALSE) +
  scale_fill_viridis(option = "inferno") +
  theme(text = element_text(size = 9))
```

