SA11a: Hydra URD Interstitial Lineage Subsetting

Jeff Farrell

October 1, 2018

Setup

```r
# Load required libraries
library(URD)
```

```
## Loading required package: ggplot2
## Warning: package 'ggplot2' was built under R version 3.4.4
## Loading required package: Matrix
## Warning: package 'Matrix' was built under R version 3.4.2
```

```r
# Set main
opts_chunk$set(root.dir = "~/Dropbox/HydraURDSubmission/")
main.path <- "~/Dropbox/HydraURDSubmission/"
```

Load Stefan's Seurat data and convert to URD

```r
# Load the IC Seurat object
hydra.seurat <- readRDS(paste0(main.path, "objects/Hydra_Seurat_IC.rds"))

# Convert hydra interstitial stem cell Seurat object to URD object
hydra.ic <- seuratToURD(hydra.seurat)
```

```
## [1] "Marchenko-Pastur eigenvalue null upper bound: 9.48365513324213"
## [1] "23 PCs have larger eigenvalues."
```

```r
# Two separate PCAs were used for the Seurat and URD analyses, which were
# performed in parallel. Both are stored in the Seurat object, but need to
# manually import the correct PCA here, as it is stored as 'ic.original.pca',
# which is not what URD imports by default.  Variable genes
ic.original.var <- scan(paste0(main.path, "cluster_results/ic-var.txt"), what = "character")
hydra.ic@var.genes <- ic.original.var
## Copy PCA over from '@dr$ic.original.pca'
hydra.ic@pca.load <- as.data.frame(hydra.seurat@dr$ic.original.pca@gene.loadings)
hydra.ic@pca.scores <- as.data.frame(hydra.seurat@dr$ic.original.pca@cell.embeddings)
hydra.ic@pca.sdev <- hydra.seurat@dr$ic.original.pca@sdev
## Recalculate significant PCs using Marchenko-Pastur rule
hydra.ic@pca.sig <- pcaMarchenkoPastur(M = dim(hydra.ic@pca.scores)[1], N = dim(hydra.ic@pca.load)[1],
    pca.sdev = hydra.ic@pca.sdev)
```

```
## [1] "Marchenko-Pastur eigenvalue null upper bound: 9.4649154132934"
## [1] "22 PCs have larger eigenvalues."
```

```r
# Remove the former Seurat object
rm(hydra.seurat)
shhh <- gc()

# Load NMF results
nmf.ic.cells <- as(as.matrix(t(read.csv(paste0(main.path, "nmf/ic_K75/GoodMeta_CellScores.csv"),
    row.names = 1))), "dgCMatrix")
rownames(nmf.ic.cells) <- gsub("X", "", gsub("\\.", "-", rownames(nmf.ic.cells)))

# Scale NMF results to 0-1
nmf.ic.cells <- sweep(nmf.ic.cells, 2, apply(nmf.ic.cells, 2, max), "/")

# Put NMF results into URD object
hydra.ic@nmf.c1 <- nmf.ic.cells
```

Remove outlier cells and doublets

Before attempting to build trajectories, it's important to clean the data by removing outlier cells that will confound the calculation of transition probabilities and removing doublets.

Remove clusters of doublets

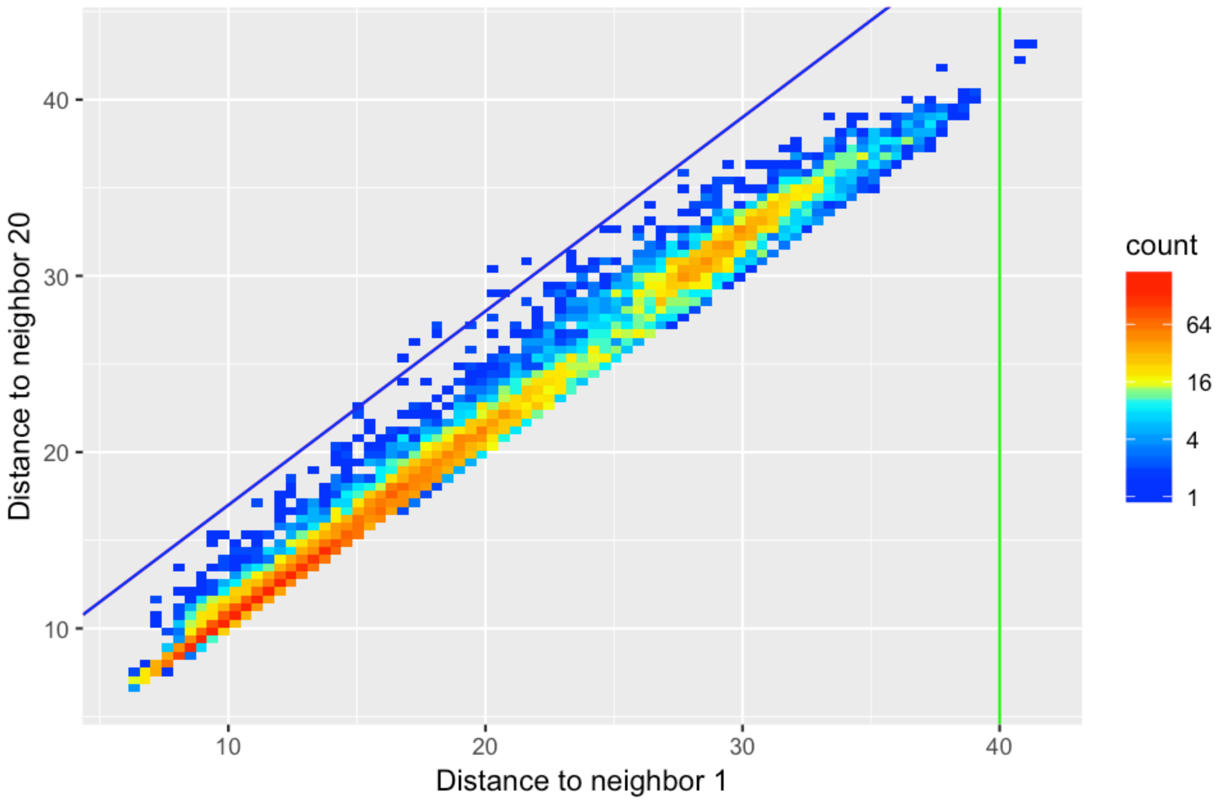
Two clusters represented clear doublets of endodermal-interstitial and ectodermal-interstitial cells. These were initially removed by their cluster identity.

```
# Remove some remaining doublets: endoderm-interstitial and
# ectoderm-interstitial, which formed clear and easy to identify clusters
germlayer.doublets <- cellsInCluster(hydra.ic, "res.1.5", c("31", "32"))
hydra.ic <- urdSubset(hydra.ic, cells.keep = setdiff(colnames(hydra.ic@logupx.data),
    germlayer.doublets))
```

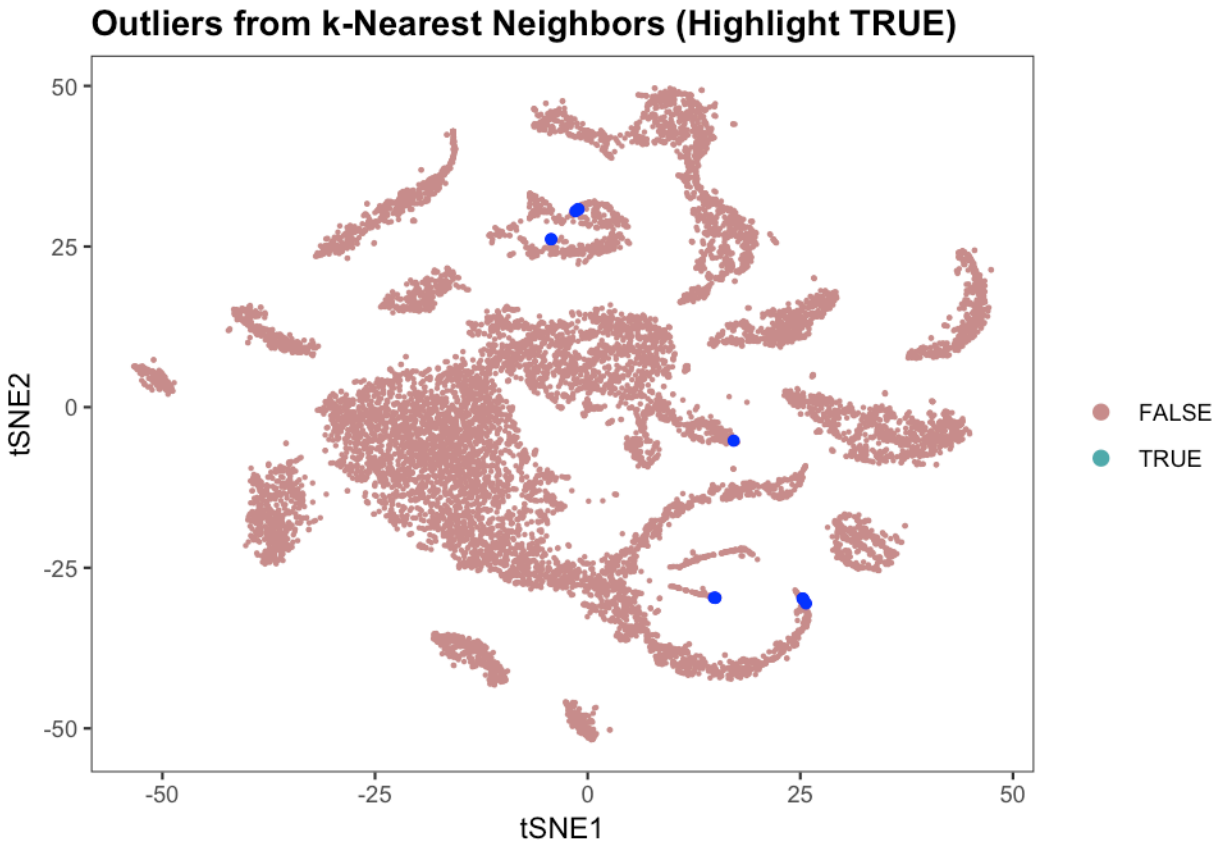
Calculate k-nearest neighbor networks

A few cells have unusually large distances to their nearest neighbors and will perform poorly in the diffusion map, so it is good to remove those.

```
# Calculate k-nearest neighbor network
hydra.ic <- calcKNN(hydra.ic, nn = 200)

# Choose outliers that have very unusual nearest neighbor distances
outliers.knn <- knnOutliers(hydra.ic, x.max = 40, slope.r = 1.1, int.r = 6, slope.b = 1.1,
    int.b = 6)  # 35 cells
```



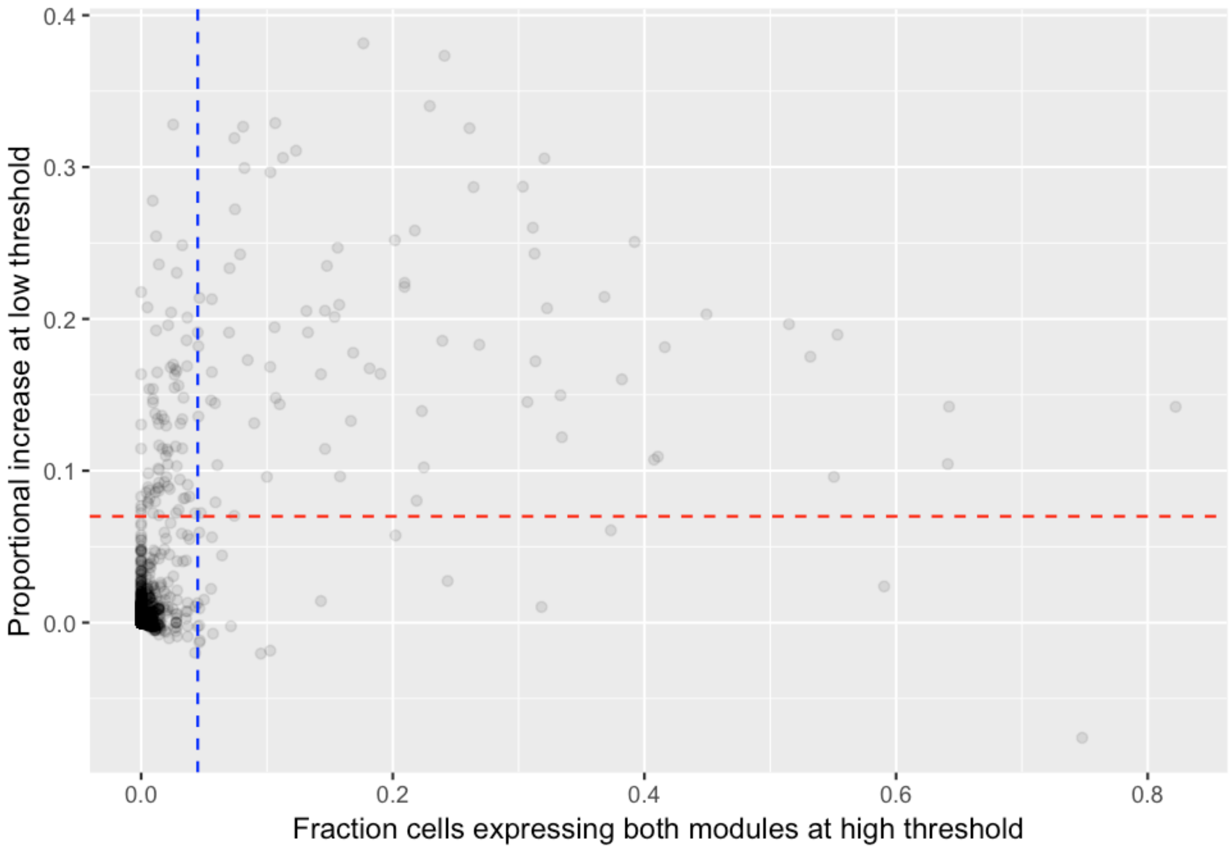
```
# Check out who those outliers are to make sure you're not cropping out all
# differentiated cells or something
hydra.ic <- groupFromCells(hydra.ic, group.id = "knn.outliers", cells = outliers.knn)
plotDimHighlight(hydra.ic, "knn.outliers", "TRUE", highlight.color = "blue", plot.title = "Outliers from k-Nearest Neighbors")
```



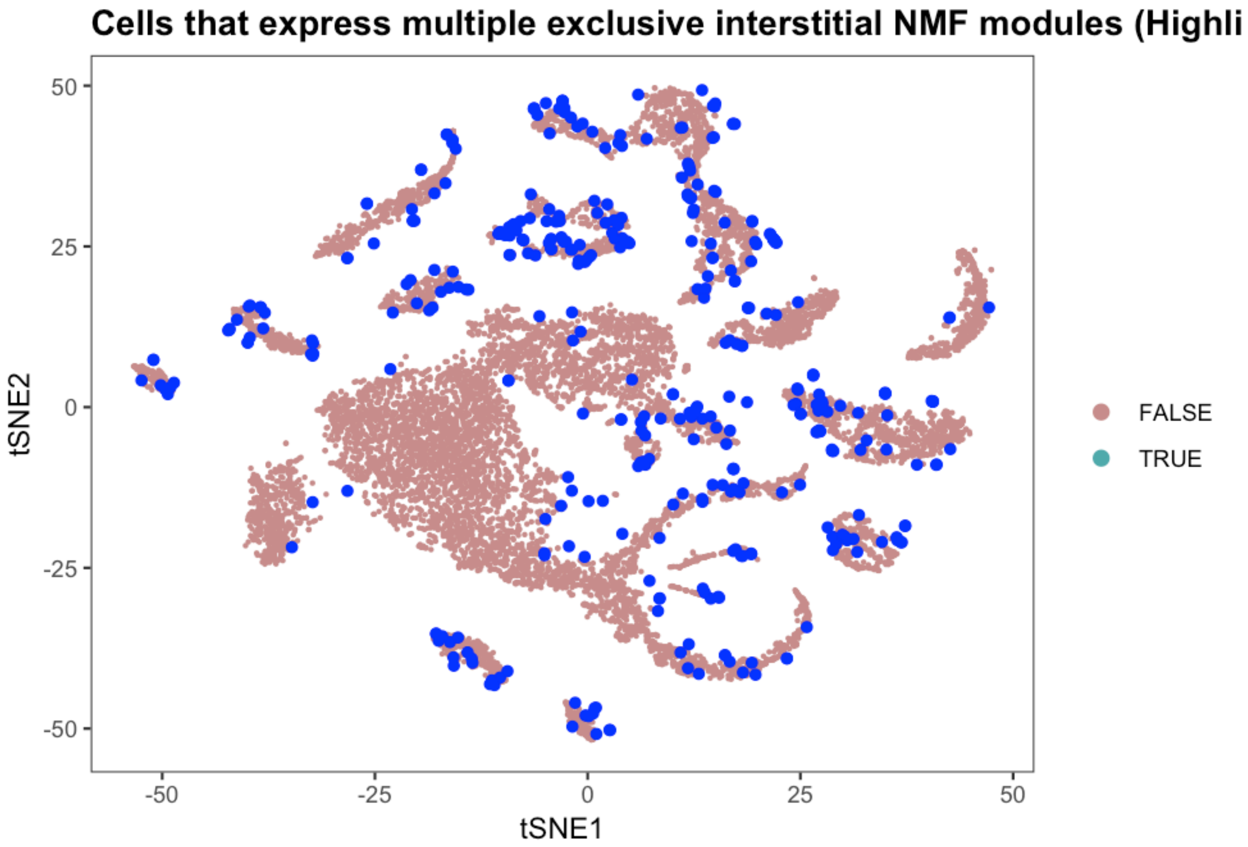
Doublets via NMF

Additionally, some cells are doublets of multiple IC lineage cell types. These are detected using NMF decomposition of the interstitial lineage. These doublets are cells that express multiple NMF modules that are non-overlapping in the data. (In other words, NMF modules that are mutually exclusive in the majority of the data.)

```
# Modules to use in the analysis: mostly Jack's 'good' modules, but with 35 and
# 43 excluded, which seem to be batch effects from the Biosearch brand beads.
ic.nmf.mods.use <- setdiff(colnames(hydra.ic@nmf.c1), c("ic35", "ic43"))

# Determine overlaps between module pairs
nmf.doublet.combos <- NMFDoubletsDefineModules(hydra.ic, modules.use = ic.nmf.mods.use,
    module.thresh.high = 0.3, module.thresh.low = 0.15)

# Determine which module pairs to use for doublets
NMFDoubletsPlotModuleThresholds(nmf.doublet.combos, frac.overlap.max = 0.045, frac.overlap.diff.max = 0.07)
```



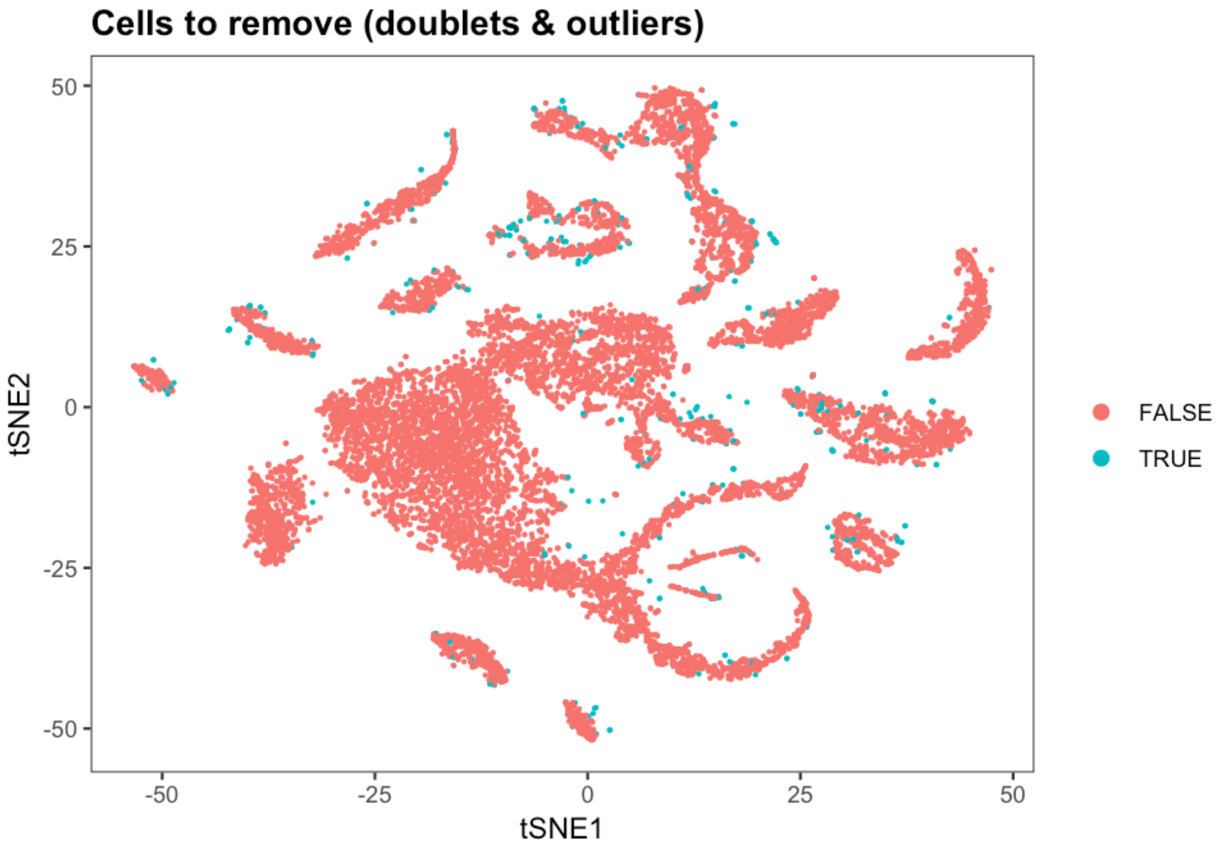
```
# Define doublet cells (now 472 doublets)
nmf.doublets <- NMFDoubletsDetermineCells(hydra.ic, nmf.doublet.combos, module.expressed.thresh = 0.25,
    frac.overlap.max = 0.045, frac.overlap.diff.max = 0.07)

# Visualize doublets
hydra.ic <- groupFromCells(hydra.ic, "nmf.doublets", nmf.doublets)
plotDimHighlight(hydra.ic, "nmf.doublets", "TRUE", highlight.color = "blue", plot.title = "Cells that express multiple exclusive
```



Remove outliers & doublets

Now we actually crop the data based on the previous two results.

```
# Define cell populations
doublets <- unique(c(outliers.knn, nmf.doublets))
cells.keep <- setdiff(colnames(hydra.ic@logupx.data), doublets)

# Plot to see where they are
hydra.ic <- groupFromCells(hydra.ic, "doublets", doublets)
plotDim(hydra.ic, "doublets", plot.title = "Cells to remove (doublets & outliers)")
```



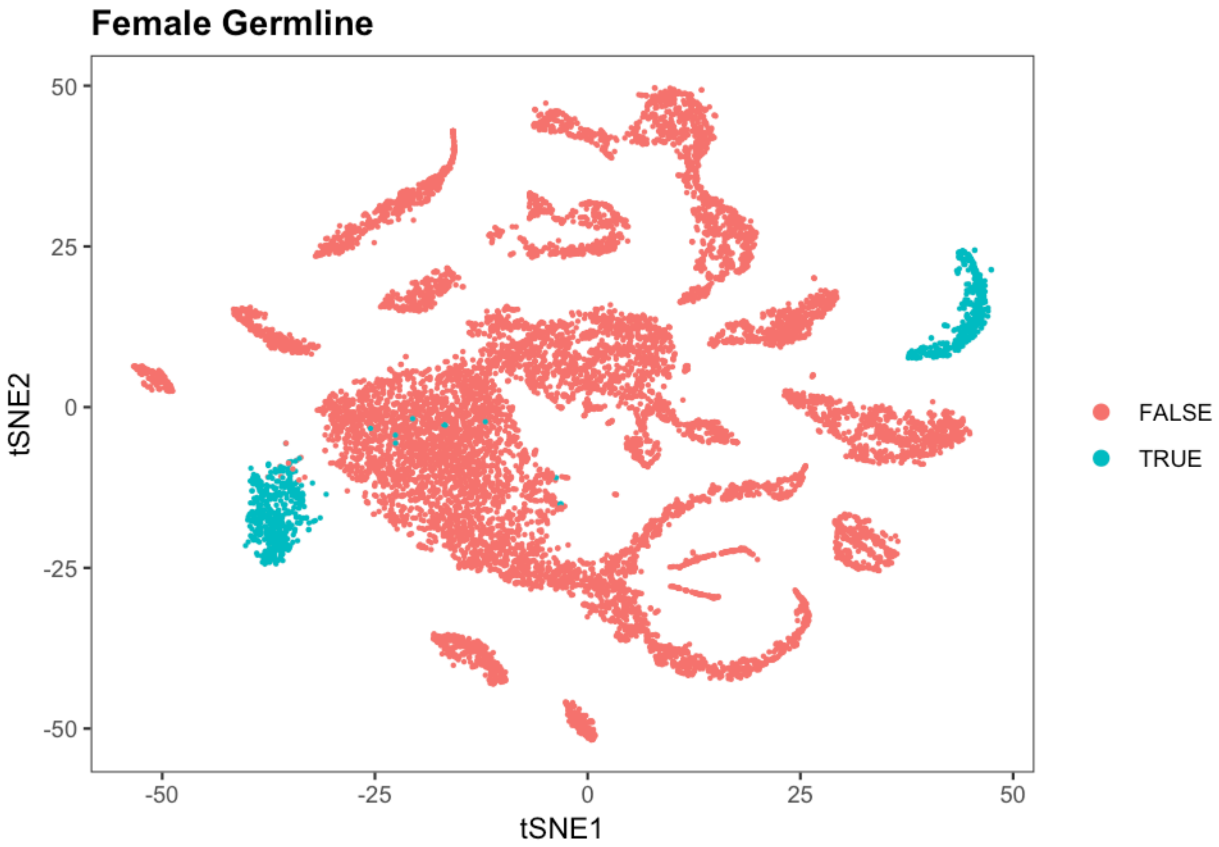
```
# Subset the data
hydra.ic <- urdSubset(hydra.ic, cells.keep = cells.keep)
```

Subset the data

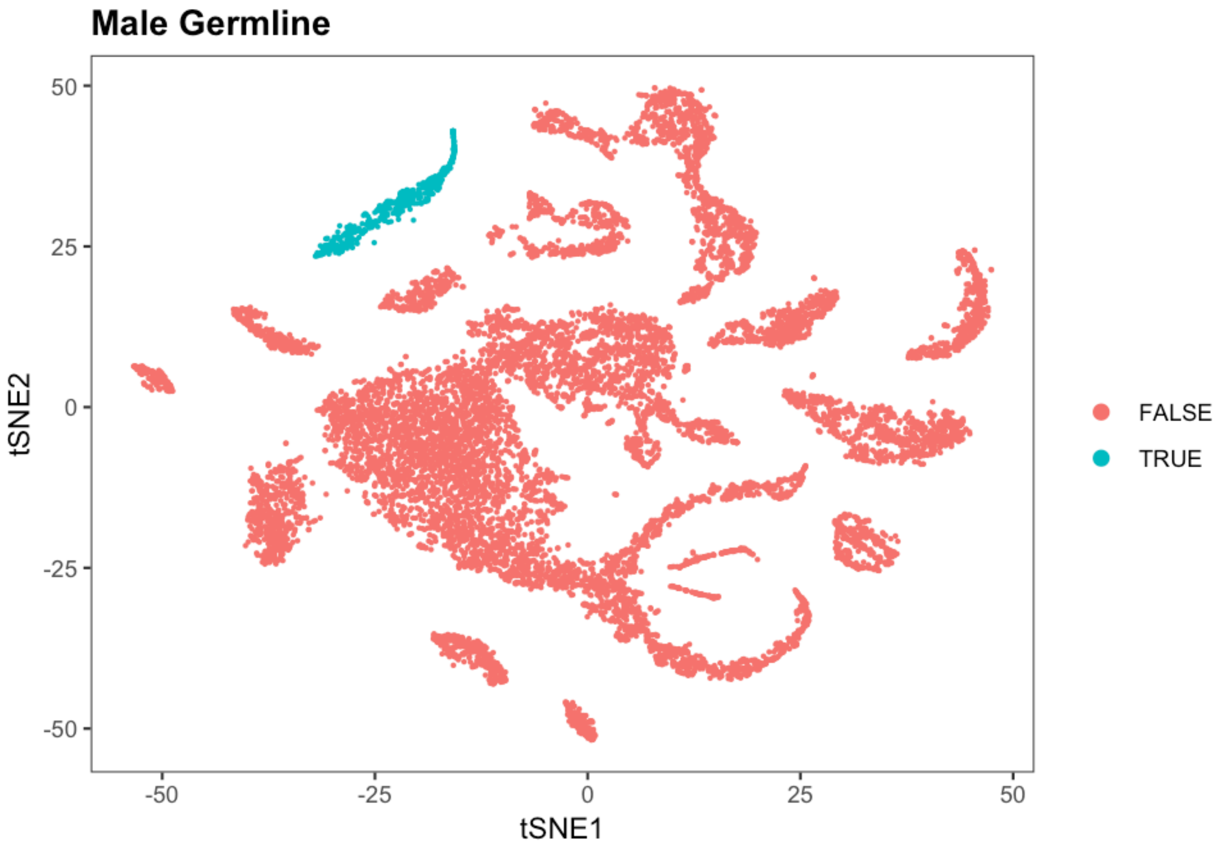
There are several lineages here that we want to process for independent spatial trajectories and the like. Thus, we carve up the data at this stage and output separate objects for each population to pursue in future sections of the analysis.

For the interstitial lineage moving forward for the tree, we remove the germline, as we have relatively few transitions here that make it challenging to reconstruct. This is likely because germline stem cells are primarily generated from the interstitial cell lineage under cases of damage, but we profiled healthy animals. We also remove the very terminal nematocytes; this population requires additional analysis, but it currently seems that the majority of these are loaded into batteries with other cell types, so very few of them have clean nematocyte transcriptomes that can be used for trajectory reconstruction. Finally, we remove two clusters that represent endoderm-interstitial and ectoderm-interstitial doublets.

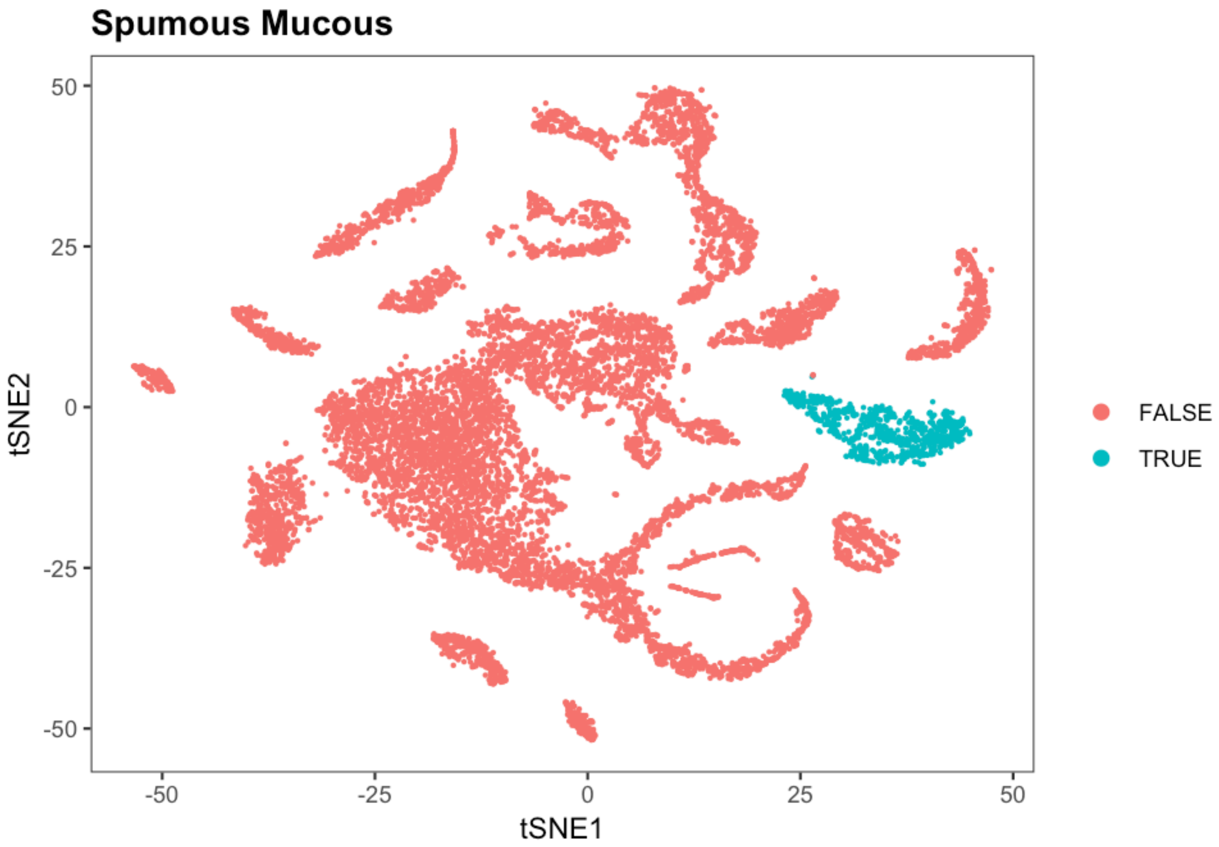
```
## FEMALE GERMLINE Determine cells in female germline by previous Seurat
## clustering
cells.female.germline <- cellsInCluster(hydra.ic, "clustering2", c("8", "10"))
# Identify cells on plot
hydra.ic <- groupFromCells(hydra.ic, "female.germline", cells.female.germline)
plotDim(hydra.ic, "female.germline", plot.title = "Female Germline")
```



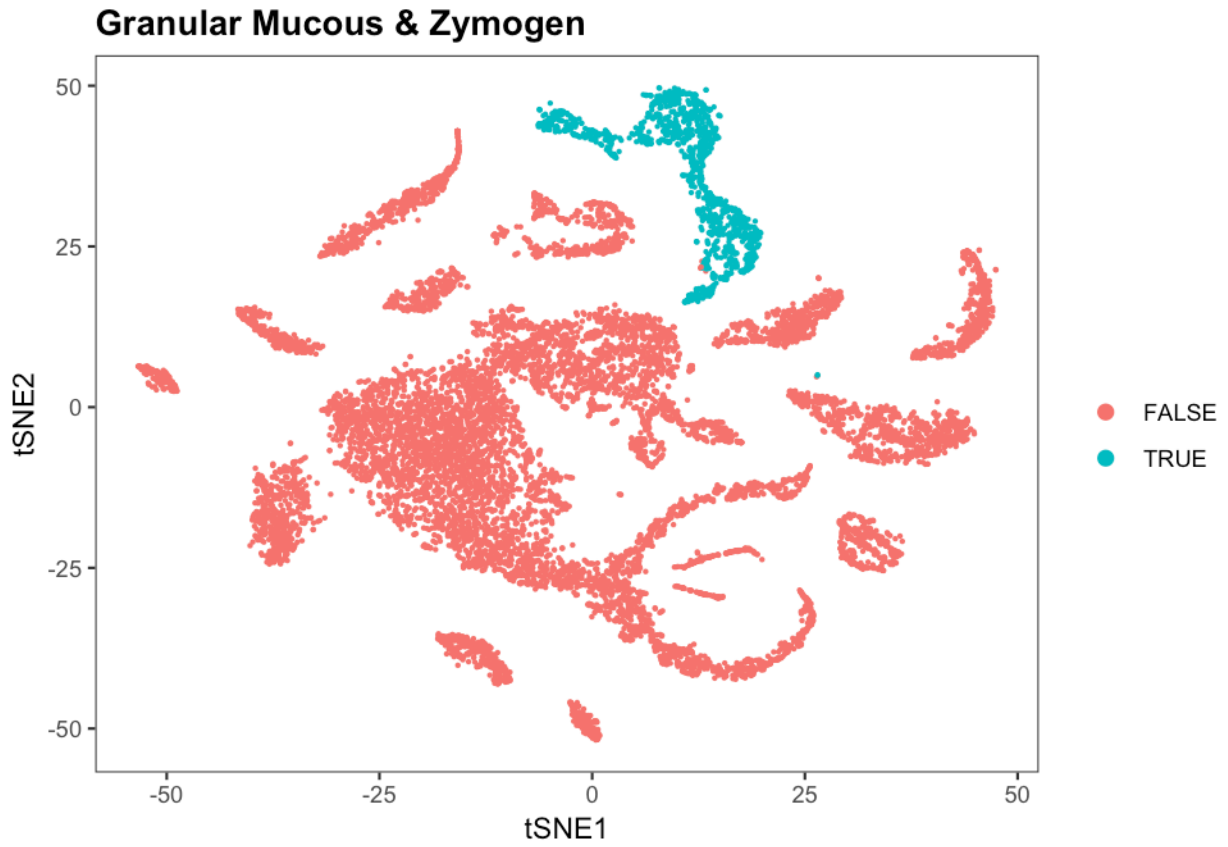
```
# Subset and save object
hydra.ic.subset <- urdSubset(hydra.ic, cells.female.germline)
saveRDS(hydra.ic.subset, file = paste0(main.path, "objects/Hydra_URD_Input_Female.rds"))

## MALE GERMLINE Determine cells in male germline by previous Seurat clustering
cells.male.germline <- cellsInCluster(hydra.ic, "clustering2", c("27", "13"))
# Identify cells on plot
hydra.ic <- groupFromCells(hydra.ic, "male.germline", cells.male.germline)
plotDim(hydra.ic, "male.germline", plot.title = "Male Germline")
```



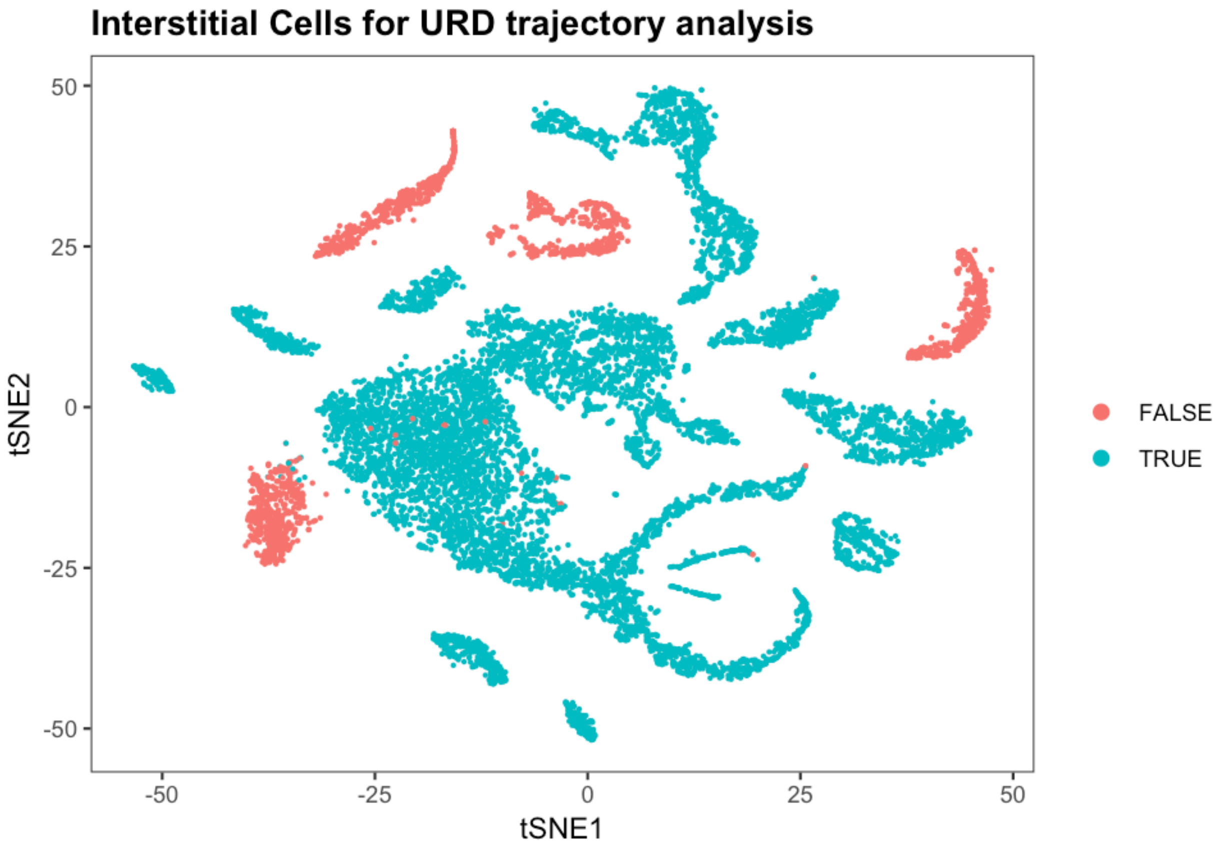
```
# Subset and save object
hydra.ic.subset <- urdSubset(hydra.ic, cells.male.germline)
saveRDS(hydra.ic.subset, file = paste0(main.path, "objects/Hydra_URD_Input_Male.rds"))

## SPUMOUS MUCOUS Determine cells in population by previous Seurat clustering
cells.spumous <- cellsInCluster(hydra.ic, "clustering2", c("23", "4"))
# Identify cells on plot
hydra.ic <- groupFromCells(hydra.ic, "spumous", cells.spumous)
plotDim(hydra.ic, "spumous", plot.title = "Spumous Mucous")
```



```
# Subset and save object
hydra.ic.subset <- urdSubset(hydra.ic, cells.spumous)
saveRDS(hydra.ic.subset, file = paste0(main.path, "objects/Hydra_URD_Input_Spumous.rds"))

## GRANULAR MUCOUS + ZYMOGEN Determine cells in population by previous Seurat
## clustering
cells.zymogen <- cellsInCluster(hydra.ic, "clustering2", c("22", "6", "11", "30"))
# Identify cells on plot
hydra.ic <- groupFromCells(hydra.ic, "zymogen", cells.zymogen)
plotDim(hydra.ic, "zymogen", plot.title = "Granular Mucous & Zymogen")
```



```
# Subset and save object
hydra.ic.subset <- urdSubset(hydra.ic, cells.zymogen)
saveRDS(hydra.ic.subset, file = paste0(main.path, "objects/Hydra_URD_Input_Zymogen.rds"))

## INTERSTITIAL LINEAGE Terminal nematocyte cells
terminal.nematocytes <- scan(paste0(main.path, "cluster_results/ic-terminal-nematocytes.txt"),
    what = "character")
# Determine cells to keep
cells.ic <- setdiff(colnames(hydra.ic@logupx.data), c(terminal.nematocytes, cells.male.germline,
    cells.female.germline))
# Identify cells on plot
hydra.ic <- groupFromCells(hydra.ic, "ic.for.tree", cells.ic)
plotDim(hydra.ic, "ic.for.tree", plot.title = "Interstitial Cells for URD trajectory analysis")
```



```
# Subset and save object
hydra.ic <- urdSubset(hydra.ic, cells.ic)
saveRDS(hydra.ic, file = paste0(main.path, "objects/Hydra_URD_Input_IC.rds"))
```