t-SNE Analysis of Non-diabetic and Type 2 Diabetic Single Cell Ensemble Transcriptomes

Introduction

This file will detail the steps used to perform unsupervised t-SNE analysis on the combined non-diabetic and type 2 diabetic single cell transcriptomes. The data was reduced to two dimensions.

t-SNE Analysis

```r
suppressPackageStartupMessages(library(Biobase))
suppressPackageStartupMessages(library(edgeR))
suppressPackageStartupMessages(library(Rtsne))
suppressPackageStartupMessages(library(RColorBrewer))
library(Biobase)
library(edgeR)
library(Rtsne)
library(RColorBrewer)
rm(list = ls())
set.seed(123345)
setwd("/Users/lawlon/Documents/Final_RNA_Seq_3/Data/")
# File name
name = "NonT2D.and.T2D.log2.cpm"
# Load in Single Cell RNA-seq data
load("nonT2D.rdata")
p.anns <- featureData(cnts.eset)
probe.anns <- as(p.anns,"data.frame")
ND.anns <- pData(cnts.eset)
# Remove multiples and keep all other groups
ND.sel <- ND.anns[ND.anns$cell.type %in% c("INS", "PPY", "GCG", "SST",
                            "COL1A1", "KRT19", "PRSS1", "none"),]
# Calculate cpm
ND.counts <- exprs(cnts.eset)
ND.cpms <- cpm(x = ND.counts)
ND.cpm <- log2(ND.cpms+1)
ND.cpm.sel <- ND.cpm[, rownames(ND.sel)]
# Load in T2D single cell data
load("T2D.rdata")
T2D.anns <- pData(cnts.eset)
# Remove multiples and keep all other groups
T2D.sel <- T2D.anns[T2D.anns$cell.type %in% c("INS", "PPY", "GCG", "SST",
                            "COL1A1", "KRT19", "PRSS1", "none"),]
# Calculate cpm
T2D.counts <- exprs(cnts.eset)
T2D.cpms <- cpm(x = T2D.counts)
T2D.cpm <- log2(T2D.cpms+1)
T2D.cpm.sel <- T2D.cpm[, rownames(T2D.sel)]
# Combine sample anns and expression data
cpm.vals <- cbind(ND.cpm.sel, T2D.cpm.sel)
s.anns.sel <- rbind(ND.sel, T2D.sel)
```

```r
# Identify the epsilon cell (high GHRL expression)
g <- which(probe.anns$Associated.Gene.Name == "GHRL")
ghrl <- cpm.vals[g,]
samp <- which(ghrl > 15)
g.idx <- which(rownames(s.anns.sel) == names(samp))
# Change sample anns of cell to GHRL
s.anns.sel$cell.type[g.idx] <- "GHRL"
cpm.vals <- cpm.vals[, rownames(s.anns.sel)]
# Obtain highly expressed genes
r.max <- apply(cpm.vals,1,max)
cpm.sel <- cpm.vals[r.max > 10.5,]
# transpose the matrix
cpm1 <- t(cpm.sel)
# Remove groups that are all zeros
df <- cpm1[,apply(cpm1, 2, var, na.rm=TRUE) != 0]
# t-SNE ANALYSIS
rtsne_out <- Rtsne(as.matrix(df), dims = 2)
# Set rownames of matrix to tsne matrix
rownames(rtsne_out$Y) <- rownames(cpm1)
# Output tsne matrix to file
#write.csv(rtsne_out$Y, file = paste(name, "tsne.matrix.data.2D.csv", sep="."))

# Specify phenotype of each of single cell samples
group <- c(rep("NonT2D", dim(ND.sel)[1]), rep("T2D", dim(T2D.sel)[1]))

# Color Schema
grey <- brewer.pal(n=9, name="Greys")
# Color codes for each specific cell type
colorCodes <- c(INS="#e41a1c", GCG = "#377eb8", SST = "#4daf4a",
                PPY = "#984ea3", GHRL = "#ff7f00",
                COL1A1 = grey[9], PRSS1 = grey[7], KRT19 = grey[5],
                none = grey[3])

namelist <- c("Beta", "Alpha", "Delta", "Gamma", "Epsilon",
              "Stellate", "Acinar", "Ductal", "none")

# Specify the shape for T2D and NonT2D in the plot
type1 <- NULL
for (i in 1:length(group)){
  if ((group[i] %in% c("NonT2D")) == TRUE) {
    idx = 20
    type1 = c(type1, idx)
  } else {
    idx = 17
    type1 = c(type1, idx)
  }
}

# Match up colors and hormone labels
cols = NULL
for (i in 1:length(s.anns.sel$cell.type)) {
  if ((s.anns.sel$cell.type[i] %in% names(colorCodes)) == TRUE) {
    cols <- c(cols, colorCodes[s.anns.sel$cell.type[i]])
  }
}
# Match up cell name with hormone name
# Have cell type name and color
for (i in 1:length(cols)) {
  if (names(cols)[i] %in% names(namelist) == TRUE) {
    names(cols)[i] <- namelist[names(cols)[i]]
  }
}
# Plot the t-sne in 2-D
pdf(file = paste(name, "Tsne.2D.plot.pdf", sep="."))

plot(rtsne_out$Y[,1], rtsne_out$Y[,2], col = cols, pch = type1,
     xlab = "t-SNE 1", ylab = "t-SNE 2", cex = 0.75)
# May need to change the legend location upon running
legend("bottomleft", legend = c(expression(bold("Beta (INS)")), expression(bold("Alpha (GCG)")),
       expression(bold("Delta (SST)")), expression(bold("Gamma (PPY)")),  expression(bold("Epsilon (GHR
       expression(bold("Stellate (COL1A1)")), expression(bold("Acinar (PRSS1)")),
       expression(bold("Ductal (KRT19)")), expression(bold("None"))), text.col = colorCodes,
       col = colorCodes, cex = 0.75)
legend(-12,-15, legend = c("NonT2D", "T2D"), pch = c(20,17), cex = 0.75)

dev.off()
```

Session Information

```
suppressPackageStartupMessages(library(Biobase))
suppressPackageStartupMessages(library(edgeR))
```

```
## Warning: package 'limma' was built under R version 3.3.1
```

```
suppressPackageStartupMessages(library(Rtsne))
suppressPackageStartupMessages(library(RColorBrewer))
library(Biobase)
library(edgeR)
library(Rtsne)
library(RColorBrewer)
sessionInfo()
```

```
## R version 3.3.0 (2016-05-03)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.11.6 (El Capitan)
##
## 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] parallel  stats     graphics  grDevices utils     datasets  methods
## [8] base
##
## other attached packages:
## [1] RColorBrewer_1.1-2  Rtsne_0.11          edgeR_3.14.0
## [4] limma_3.28.21       Biobase_2.32.0      BiocGenerics_0.18.0
```

```
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.7     digest_0.6.10   assertthat_0.1  formatR_1.4
##  [5] magrittr_1.5    evaluate_0.10   stringi_1.1.2   rmarkdown_1.1
##  [9] tools_3.3.0     stringr_1.1.0   yaml_2.1.13     htmltools_0.3.5
## [13] knitr_1.14      tibble_1.2
```