

# Unsupervised Hierarchical Clustering of Non-diabetic Single Cell Ensemble Transcriptomes Without Hormone Marker Genes

## Introduction

This file will detail the steps used to perform unsupervised hierarchical clustering analysis on the non-diabetic and type 2 diabetic single cell transcriptomes without using pancreatic cell hormone marker genes (INS, GCG, SST, PPY, GHRL, COL1A1, PRSS1, and KRT19).

## Hierarchical Clustering

```
suppressPackageStartupMessages(library(Biobase))
suppressPackageStartupMessages(library(edgeR))
suppressPackageStartupMessages(library(ape))
suppressPackageStartupMessages(library(gplots))
suppressPackageStartupMessages(library(dendextend))
suppressPackageStartupMessages(library(RColorBrewer))
library(edgeR)
library(Biobase)
library(gplots)
library(dendextend)
library(ape)
library(RColorBrewer)
rm(list=ls())
set.seed(2135435)
# file name
fname = "NonT2D.and.T2D.log2.cpm.by.disease.no.hormones"
setwd("/Users/lawlon/Documents/Final_RNA_Seq_3/Data/")
# Load in nonT2D single cell 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.cpm <- cpm(x = T2D.counts)
T2D.cpm <- log2(T2D.cpm+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.max <- apply(cpm.vals,1,max)
# Use highly expressed genes
cpm.sel <- cpm.vals[r.max > 10.5,]
# Remove hormonal genes from list
horm <- which(p.anns$Associated.Gene.Name %in% c("INS","GCG","PPY","SST", "GHRL",
"COL1A1", "PRSS1", "KRT19"))

ids <- rownames(p.anns)[horm]
indices <- which(rownames(cpm.sel) %in% ids)
cpm.sel <- cpm.sel[-indices,]
# Save a copy of data
cpm.res <- cpm.sel
# Change column name labels to cell type and disease state
colnames(cpm.res)[1:dim(ND.sel)[1]] <- paste(ND.sel$cell.type, "NonT2D", sep="-")
colnames(cpm.res)[(dim(ND.sel)[1]+1):dim(cpm.res)[2]] <- paste(T2D.sel$cell.type, "T2D", sep="-")
# Change name of one KRT19 cell to ghrelin cell
g <- which(probe.anns$Associated.Gene.Name == "GHRL")
ghrl <- cpm.vals[g,]
samp <- which(ghrl > 15)
g.idx <- which(rownames(s.anns.sel) == names(samp))
colnames(cpm.res)[g.idx] <- "GHRL-NonT2D"
p.res <- probe.anns[rownames(cpm.res),]
# Combine probe anns with selected fpkm values
cpm.res.exp <- cbind(p.res,cpm.res)
# Write genes used for clustering to file
write.csv(cpm.res.exp, paste(fname, "genes_selected_for_cing.csv", sep = "."))
# Hclust object of samples
exp.sel <- cpm.res
d <- dist(t(exp.sel))
hc.final <- hclust(d,method="ward.D2")
# Change hclust to dendrogram
dend1 <- as.dendrogram(hc.final)
dendcol <- as.dendrogram(hc.final)
groupCodes1 <- s.anns.sel$cell.type
groupCodes <- c(rep("NonT2D", dim(ND.sel)[1]), rep("T2D", dim(T2D.sel)[1]))

# Color Schema
grey <- brewer.pal(n=9, name="Greys")
colorCodes1 <- c(INS="#e41a1c", GCG = "#377eb8", SST = "#4daf4a",
PPY = "#984ea3", GHRL = "#ff7f00",
COL1A1 = grey[9], PRSS1 = grey[7], KRT19 = grey[5],
none = grey[3])
colorCodes <- c(NonT2D="#bda2e5", T2D = "#10d2f0")

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

```

```

labels_colors(dend1) <- colorCodes[groupCodes][order.dendrogram(dend1)]
labels_colors(dendcol) <- colorCodes1[groupCodes1][order.dendrogram(dendcol)]
# Change dend to phylo object
dend2 <- as.phylo(dend1)
dend3 <- as.phylo(dendcol)
# Match up colors and labels
cols = NULL
for (i in 1:length(labels(dend2))) {
  if (grepl(x = dend2$tip.label[i], pattern = "NonT2D") == TRUE) {
    cols <- c(cols, colorCodes["NonT2D"])
  } else if (grepl(x = dend2$tip.label[i], pattern = "T2D") == TRUE) {
    cols <- c(cols, colorCodes["T2D"])
  }
}
# Match up cell type and color for dend3
col3 = NULL
for (i in 1:length(labels(dend3))) {
  if (grepl(x = dend3$tip.label[i], pattern = "INS") == TRUE) {
    col3 <- c(col3, colorCodes1["INS"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "GCG") == TRUE) {
    col3 <- c(col3, colorCodes1["GCG"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "SST") == TRUE) {
    col3 <- c(col3, colorCodes1["SST"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "PPY") == TRUE) {
    col3 <- c(col3, colorCodes1["PPY"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "COL1A1") == TRUE) {
    col3 <- c(col3, colorCodes1["COL1A1"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "PRSS1") == TRUE) {
    col3 <- c(col3, colorCodes1["PRSS1"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "KRT19") == TRUE) {
    col3 <- c(col3, colorCodes1["KRT19"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "GHRL") == TRUE) {
    col3 <- c(col3, colorCodes1["GHRL"])
  } else if (grepl(x = dend3$tip.label[i], pattern = "none") == TRUE) {
    col3 <- c(col3, colorCodes1["none"])
  }
}
#Use the long hyphen or the minus sign instead of regular hyphen symbol
labels(dend2) <- rep(x = "-", length(labels(dend2)))
labels(dend3) <- rep(x = "-", length(labels(dend3)))
# Make high resolution tiff image of plot, very big file!
tiff(file=paste(fname, "dendrogram.tiff", sep = "."),
      width = 11000, height = 11000, units = "px", res = 800)

plot(dend2, type = "fan", tip.color = col3, cex = 5.0, label.offset = 0)
legend("bottomleft", title = "Cell Types", title.col = "black",
      legend = c(expression(bold("Beta (INS)")), expression(bold("Alpha (GCG)")),
        expression(bold("Delta (SST)")), expression(bold("Gamma (PPY)")),
        expression(bold("Epsilon (GHRL)")),
        expression(bold("Stellate (COL1A1)")), expression(bold("Acinar (PRSS1)")),
        expression(bold("Ductal (KRT19)")), expression(bold("None"))), text.col = colorCodes1,
      cex = 0.9, xjust=0, yjust=0)
legend("bottomright", title = "Disease State", title.col = "black",

```

```

        legend = c(expression(bold("NonT2D")), expression(bold("T2D"))),
        text.col = colorCodes,
        cex = 1.0, xjust=0, yjust=0)
par(new = TRUE)
plot(dend3, type = "fan", tip.color = cols, cex = 5.0, label.offset = 40)
dev.off()

```

## Session Information

```

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

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

suppressPackageStartupMessages(library(ape))
suppressPackageStartupMessages(library(gplots))
suppressPackageStartupMessages(library(dendextend))
suppressPackageStartupMessages(library(RColorBrewer))
library(edgeR)
library(Biobase)
library(gplots)
library(dendextend)
library(ape)
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 dendextend_1.3.0 gplots_3.0.1
##  [4] ape_3.5 edgeR_3.14.0 limma_3.28.21
##  [7] Biobase_2.32.0 BiocGenerics_0.18.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.7 DEoptimR_1.0-6 formatR_1.4
##  [4] plyr_1.8.4 class_7.3-14 bitops_1.0-6
##  [7] tools_3.3.0 prabclus_2.2-6 digest_0.6.10
## [10] mclust_5.2 evaluate_0.10 tibble_1.2
## [13] nlme_3.1-128 gtable_0.2.0 lattice_0.20-34
## [16] yaml_2.1.13 mvtnorm_1.0-5 trimcluster_0.1-2
## [19] stringr_1.1.0 knitr_1.14 cluster_2.0.5
## [22] gtools_3.5.0 caTools_1.17.1 fpc_2.1-10
## [25] diptest_0.75-7 stats4_3.3.0 grid_3.3.0
## [28] nnet_7.3-12 robustbase_0.92-6 flexmix_2.3-13

```

```
## [31] rmarkdown_1.1      gdata_2.17.0      kernlab_0.9-25
## [34] ggplot2_2.1.0      magrittr_1.5       whisker_0.3-2
## [37] scales_0.4.0       htmltools_0.3.5    modeltools_0.2-21
## [40] MASS_7.3-45        assertthat_0.1     colorspace_1.2-7
## [43] KernSmooth_2.23-15 stringi_1.1.2      munsell_0.4.3
```