Self Organizing Maps (SOM): Example using RNAseq reads

Part 1: Formatting data for SOM

The goal of SOM analysis in the context of an RNAseq experiment is to find groups of genes that share similar expression patterns. Running SOM on all of your genes may make interpretation of SOM extremely difficult as the data set is simply too large to make sense of the clustering results. Only genes that vary significantly in expression across tissue types and can be compared between species are analyzed in this example. Another option is to use the top 25% of the genes with the largest variance across your samples.

In order to remove differences due to the magnitude of gene expression and focus only on gene expression profiles, expression values were mean centered and variance scaled.

Required Libraries

```
library(reshape)
```

Read in files and format data from raw files

To start, please set your working directory to the location of this source file.

Create a list of all DE analysi from the count data. These steps will vary considerably depending on the dataset. The count was generated and normalized using edgeR.

```
#Read in count data
countData <- read.csv("../data/normalized_read_count.csv")

#Melt count data
countData <- melt(countData)
colnames(countData) <- c("gene", "sample", "count")

#set genotype

countData$genotype <- ifelse(grepl("wt", countData$sample, ignore.case = T), "wt",
        ifelse(grepl("tf2", countData$sample, ignore.case = T), "tf2", "unknown"))

#set tissue

countData$tissue <- ifelse(grepl("other", countData$sample, ignore.case = T), "other",
        ifelse(grepl("mbr", countData$sample, ignore.case = T), "mbr", "unknown"))

#Set Region
countData$region <- ifelse(grepl("a", countData$sample, ignore.case = T), "A",
        ifelse(grepl("c", countData$sample, ignore.case = T), "C", "B"))

#Set type

countData$type <- paste(countData$region, countData$tissue,  sep = "")
head(countData)

#Subset by Genotype, since we will not be looking at tf2 at this stage
countData <- subset(countData, genotype == "wt")
```

Most Significantly DE genes

This is just reading the results from the pair-wise differential expression analysis between tissue in edgeR.

```
#Read in each list of DE expressed genes just to get the list of all genes present in my RNAseq analysi

wtaother_wtcother  <- read.table("../data/allSigGenes/wtaother_wtcother_DE_sig.txt", header = TRUE, fil
wtambr_wtaother <- read.table("../data/allSigGenes/wtambr_wtaother_DE_sig.txt", header = TRUE, fill = T
wtambr_wtbmbr <- read.table("../data/allSigGenes/wtambr_wtbmbr_DE_sig.txt", header = TRUE, fill = TRUE)
wtambr_wtcmbr <- read.table("../data/allSigGenes/wtambr_wtcmbr_DE_sig.txt", header = TRUE, fill = TRUE)
wtaother_wtbother <- read.table("../data/allSigGenes/wtaother_wtbother_DE_sig.txt", header = TRUE, fill
wtbmbr_wtbother <- read.table("../data/allSigGenes/wtbmbr_wtbother_DE_sig.txt", header = TRUE, fill = T
wtbmbr_wtcmbr <- read.table("../data/allSigGenes/wtbmbr_wtcmbr_DE_sig.txt", header = TRUE, fill = TRUE)
wtbother_wtcother <- read.table("../data/allSigGenes/wtbother_wtcother_DE_sig.txt", header = TRUE, fill
wtcmbr_wtcother <- read.table("../data/allSigGenes/wtcmbr_wtcother_DE_sig.txt", header = TRUE, fill = T

#merge them together
allGenes <- rbind(wtaother_wtcother, wtambr_wtaother, wtambr_wtbmbr, wtambr_wtcmbr, wtaother_wtbother,w

dim(allGenes)

#recieve just the list
allGenesITAG <- allGenes[,1]
length(allGenesITAG)
#Remove duplicates
allGenesITAG <- unique(allGenesITAG)
```

Mean count across samples

```
#make an empty table to hold all the genes
allGeneList <- data.frame(t(rep(NA,7)))
colnames(allGeneList) <- c("type", "genotype", "N", "mean", "sd", "se", "gene")
allGeneList <- allGeneList[-1,] #remove first row

# Takes some time to run

for(GENE in allGenesITAG) {

  if(length(grep(GENE, countData$gene)) < 1){ #this is just making sure that the list of sig genes
    next;
    }

  geneData <- subset(countData, grepl(GENE, countData$gene))

  sumGraph <- ddply(geneData, c("type", "genotype"), summarise,
               N    = length(count),
               mean = mean(count),
               sd   = sd(count),
               se   = sd / sqrt(N))

  sumGraph$gene <- GENE
```

```
allGeneList  <- rbind(allGeneList, sumGraph) #bind together all the new rows per loop.
}

dim(allGeneList)
head(allGeneList)

write.table(allGeneList, file = "../data/allGeneList.csv", sep = ",")
```