Analysis 1 - Top 25% of coefficient of variation Large SOM

Author: Ciera Martinez Date: October 23 - 31, 2014

AIM 1

Purpose:

In this analysis I am using the top 25% of genes based on co-efficient of variation, then proceeding to Self Organizing Map (SOM) clustering of gene co-expression across tissue. From discussions with Neelima, the only aspect that we are really interested in co-expression of genes through time in each tissue seperatley. We are not interested in the interaction between these tissues at this time. Ideally we are looking for genes that have co-expression patterns of up-regulation through time or down regulation through time.

Tissue Key:

SAM: Refers to shoot apices, likely with P0 - P4. Leaf: Likley P5

The plants were allowed to grow to 5 different ages (still need to talk with Yasu about specifics), the same tissue (SAM & Leaf), were extracted from plant of the five different ages (a1, b2, c3, d4, e5).

The tissue was dissected by Yasu.

Analysis

Required Libraries

```r
library(ggplot2)
library(reshape)
library(kohonen)
```

Cluster visualization functions. These are functions that are re-used throughout analysis.

```r
#ClusterVis
clusterVis <- function(clustNum){

  sub_cluster <- subset(plot.data, som$unit.classif==clustNum)
  sub_data <- sub_cluster[,c(1:11)] # just the sample types
  m.data <- melt(sub_data)
  p <- ggplot(m.data, aes(x=variable, y=value))
  p + geom_point(alpha=0.5,position="jitter", size=1) +
    geom_boxplot(alpha=0.75, outlier.size=0) +
    theme_bw() +
    theme(text = element_text(size=30),
          axis.text.x = element_text(angle=90,
                                     vjust=1)) +
    xlab("Library") +
    ylab("Scaled Gene Expression")
}
```

clusterVis_line

This function is used to plot gene expression profiles of clusters throughout time using a line plot.

To-do: [] Need to remove unused x-axis values between graphs.

```r
clusterVis_line <- function(clustNum) {
  sub_cluster <- subset(plot.data, som$unit.classif==clustNum)
  sub_data <- sub_cluster[,c(1:11)] # just the sample types
  m.data <- melt(sub_data)
  m.data$region <- ifelse(grepl("SAM", m.data$variable, ignore.case = T), "SAM",
                          ifelse(grepl("leaf", m.data$variable, ignore.case = T), "leaf", "other"))
  head(m.data)
  m.data <- within(m.data, lineGroup <- paste(gene,sep='.'))
  ggplot(m.data, aes(variable, value, group = lineGroup)) +
    geom_line(alpha = .1) +
    geom_point(alpha = .0) +
    theme_bw() +
    facet_grid(.~region) +
    theme(axis.text.x = element_text(size=20,
                                angle=90,
                                vjust=1))
  }
```

clusterVis_region

This function is not finished, but could be used to visually articulate age.

not finished.

```r
clusterVis_region <- function(clustNum){
  sub_cluster <- subset(plot.data, som$unit.classif==1)
  sub_data <- sub_cluster[,c(1:11)] # just the sample types
  m.data <- melt(sub_data)
  m.data$region <- ifelse(grepl("SAM", m.data$variable, ignore.case = T), "SAM",
                          ifelse(grepl("leaf", m.data$variable, ignore.case = T), "leaf", "other"))
  #Adds a column that specifies age
  m.data$age <- ifelse(grepl("a1", m.data$variable, ignore.case = T), "1",
                          ifelse(grepl("b2", m.data$variable, ignore.case = T), "2",
                            ifelse(grepl("c3", m.data$variable, ignore.case = T), "3",
                                ifelse(grepl("d4", m.data$variable, ignore.case = T), "4",
                                  ifelse(grepl("e5", m.data$variable, ignore.case = T), "5", "other")
                                  )
                                )
                             )
                          )

  head(m.data)
  p <- ggplot(m.data, aes(y=value, x=variable, fill = age))
  p + geom_point(alpha=0.5,position="jitter", size=1) +
    geom_boxplot(alpha=0.70, outlier.size=0) +
    scale_colour_manual(values = c("darkorchid1", "coral")) +
    theme(legend.text = element_text(
      size = 30,
      face = "bold"),
      text = element_text(size=40)) +
    theme_bw() +
    theme(text = element_text(size=30)) +
    facet_grid(.~region)
}
```

genesInCluster()

This function is used to identify which genes are in the cluster.

```
#Prereq annotation files for function

annotation1<- read.delim("../data/ITAG2.3_all_Arabidopsis_ITAG_annotations.tsv", header=FALSE)  #Change
colnames(annotation1) <- c("ITAG", "SGN_annotation")
annotation2<- read.delim ("../data/ITAG2.3_all_Arabidopsis_annotated.tsv")
annotation <- merge(annotation1,annotation2, by = "ITAG")

#Only Gene Name and ITAG
names(annotation)
```

```
##  [1] "ITAG"                 "SGN_annotation"      "AGI"
##  [4] "symbol"               "gene_name"           "X..identity"
##  [7] "alignment.length"     "e.value"             "bit.score"
## [10] "percent.query.align"
```

```
annotation <- annotation[,c(1,2,3,5)]

#fix with regex if ITAG does not include the last digits
#annotation$ITAG <- gsub("^(.*)[.].*", "\\1", annotation$ITAG)
#annotation$ITAG <- gsub("^(.*)[.].*", "\\1", annotation$ITAG)

###genesInClust()
#This looks at how many unique genes are in each cluster.

genesInClust <- function(clustNum, plot.data, annotation) {
  sub_cluster <- subset(plot.data, som$unit.classif==clustNum)
  sub_data <- as.data.frame(sub_cluster[,1])
  colnames(sub_data) <- "ITAG"
  resultsTable <- merge(sub_data,annotation,by = "ITAG", all.x=TRUE)
  print(nrow(unique(resultsTable)))
  return(unique(resultsTable))
  }
```

Get the co-efficient of variation.

```
countData <- read.csv("../data/normalized_count_file.csv")
#Then sort
#it adds numbers to them to make them unique but ignore
countData1 <- countData[,order(names(countData))] #sorting for easier assignment
names(countData1)
```

```
##  [1] "fifth.leaf.1"  "fifth.leaf.2"  "fifth.leaf.3"  "fifth.leaf.4"
##  [5] "fifth.SAM.1"   "fifth.SAM.2"   "fifth.SAM.3"   "fifth.SAM.4"  
##  [9] "first.leaf.2"  "first.leaf.3"  "first.leaf.4"  "first.SAM.1"  
## [13] "first.SAM.2"   "first.SAM.3"   "first.SAM.4"   "fourth.leaf.1"
## [17] "fourth.leaf.2" "fourth.leaf.3" "fourth.leaf.4" "fourth.SAM.5" 
## [21] "fourth.SAM.6"  "fourth.SAM.7"  "fourth.SAM.8"  "second.leaf.1"
## [25] "second.leaf.2" "second.leaf.3" "second.leaf.4" "second.SAM.1" 
## [29] "second.SAM.2"  "second.SAM.3"  "second.SAM.4"  "third.leaf.1" 
## [33] "third.leaf.2"  "third.leaf.3"  "third.leaf.4"  "third.leaf.5" 
## [37] "third.leaf.6"  "third.leaf.7"  "third.SAM.1"   "third.SAM.2"  
## [41] "third.SAM.3"   "third.SAM.4"   "third.SAM.5"   "third.SAM.6"  
## [45] "third.SAM.7"   "third.SAM.8"   "X"
```

```
countData1 <- subset(countData1, select=c(47,1:46)) #re-order
```

```
#remove low count libraries (3rd.leaf.7, 2nd.SAM.4, 5th.leaf.3)
dim(countData1) #check
```

```
## [1] 27741    47
```

```
names(countData1) #check
```

```
##  [1] "X"             "fifth.leaf.1"  "fifth.leaf.2"  "fifth.leaf.3" 
##  [5] "fifth.leaf.4"  "fifth.SAM.1"   "fifth.SAM.2"   "fifth.SAM.3"  
##  [9] "fifth.SAM.4"   "first.leaf.2"  "first.leaf.3"  "first.leaf.4" 
## [13] "first.SAM.1"   "first.SAM.2"   "first.SAM.3"   "first.SAM.4"  
## [17] "fourth.leaf.1" "fourth.leaf.2" "fourth.leaf.3" "fourth.leaf.4"
## [21] "fourth.SAM.5"  "fourth.SAM.6"  "fourth.SAM.7"  "fourth.SAM.8" 
## [25] "second.leaf.1" "second.leaf.2" "second.leaf.3" "second.leaf.4"
## [29] "second.SAM.1"  "second.SAM.2"  "second.SAM.3"  "second.SAM.4" 
## [33] "third.leaf.1"  "third.leaf.2"  "third.leaf.3"  "third.leaf.4" 
## [37] "third.leaf.5"  "third.leaf.6"  "third.leaf.7"  "third.SAM.1"  
## [41] "third.SAM.2"   "third.SAM.3"   "third.SAM.4"   "third.SAM.5"  
## [45] "third.SAM.6"   "third.SAM.7"   "third.SAM.8"
```

```
countData2 <- countData1[,-c(39,32,11)] #removal
names(countData2) #check
```

```
##  [1] "X"             "fifth.leaf.1"  "fifth.leaf.2"  "fifth.leaf.3" 
##  [5] "fifth.leaf.4"  "fifth.SAM.1"   "fifth.SAM.2"   "fifth.SAM.3"  
##  [9] "fifth.SAM.4"   "first.leaf.2"  "first.leaf.4"  "first.SAM.1"  
## [13] "first.SAM.2"   "first.SAM.3"   "first.SAM.4"   "fourth.leaf.1"
## [17] "fourth.leaf.2" "fourth.leaf.3" "fourth.leaf.4" "fourth.SAM.5" 
## [21] "fourth.SAM.6"  "fourth.SAM.7"  "fourth.SAM.8"  "second.leaf.1"
## [25] "second.leaf.2" "second.leaf.3" "second.leaf.4" "second.SAM.1" 
## [29] "second.SAM.2"  "second.SAM.3"  "third.leaf.1"  "third.leaf.2" 
## [33] "third.leaf.3"  "third.leaf.4"  "third.leaf.5"  "third.leaf.6" 
## [37] "third.SAM.1"   "third.SAM.2"   "third.SAM.3"   "third.SAM.4"  
## [41] "third.SAM.5"   "third.SAM.6"   "third.SAM.7"   "third.SAM.8"
```

```
dim(countData2) #check
```

```
## [1] 27741    44
```

```
#get row means per tissue type. This could be improved to be more manual.

countData2$a1.leaf <- rowMeans(subset(countData2[10:11]))
countData2$a1.SAM <- rowMeans(subset(countData2[12:15]))
countData2$b2.leaf <- rowMeans(subset(countData2[24:27]))
countData2$b2.SAM <- rowMeans(subset(countData2[28:30]))
countData2$c3.leaf <- rowMeans(subset(countData2[31:36]))
countData2$c3.SAM <- rowMeans(subset(countData2[37:44]))
countData2$d4.leaf <- rowMeans(subset(countData2[16:19]))
countData2$d4.SAM <- rowMeans(subset(countData2[20:23]))
countData2$e5.leaf <- rowMeans(subset(countData2[2:5]))
countData2$e5.SAM <- rowMeans(subset(countData2[6:10]))

dim(countData2) #check
```

```
## [1] 27741    54
```

```
names(countData2) #check
```

```
##  [1] "X"             "fifth.leaf.1"  "fifth.leaf.2"  "fifth.leaf.3"
##  [5] "fifth.leaf.4"  "fifth.SAM.1"   "fifth.SAM.2"   "fifth.SAM.3"
##  [9] "fifth.SAM.4"   "first.leaf.2"  "first.leaf.4"  "first.SAM.1"
## [13] "first.SAM.2"   "first.SAM.3"   "first.SAM.4"   "fourth.leaf.1"
## [17] "fourth.leaf.2" "fourth.leaf.3" "fourth.leaf.4" "fourth.SAM.5"
## [21] "fourth.SAM.6"  "fourth.SAM.7"  "fourth.SAM.8"  "second.leaf.1"
## [25] "second.leaf.2" "second.leaf.3" "second.leaf.4" "second.SAM.1"
## [29] "second.SAM.2"  "second.SAM.3"  "third.leaf.1"  "third.leaf.2"
## [33] "third.leaf.3"  "third.leaf.4"  "third.leaf.5"  "third.leaf.6"
## [37] "third.SAM.1"   "third.SAM.2"   "third.SAM.3"   "third.SAM.4"
## [41] "third.SAM.5"   "third.SAM.6"   "third.SAM.7"   "third.SAM.8"
## [45] "a1.leaf"       "a1.SAM"        "b2.leaf"       "b2.SAM"
## [49] "c3.leaf"       "c3.SAM"        "d4.leaf"       "d4.SAM"
## [53] "e5.leaf"       "e5.SAM"
```

```
#Average and Standard deviation
ave <- subset(countData2[45:54])
ave$sd <- apply(ave,1,function(d)sd(d))
ave$average <- rowMeans(subset(ave[1:10]))
ave$cv <- ave$sd / ave$average
dim(ave)#check
```

```
## [1] 27741    13
```

```
names(ave)#check
```

```
##  [1] "a1.leaf" "a1.SAM"  "b2.leaf" "b2.SAM"  "c3.leaf" "c3.SAM"  "d4.leaf"
##  [8] "d4.SAM"  "e5.leaf" "e5.SAM"  "sd"      "average" "cv"
```

```
#combine new columns to orginal
countData <- cbind(countData, countData2[45:54])
countData <- cbind(countData, ave[,11:13])

names(countData) #check
```

```
##  [1] "X"                "first.SAM.1"   "second.SAM.1"  "third.leaf.1"
##  [5] "third.SAM.1"    "third.SAM.2"   "fifth.leaf.1"  "fourth.SAM.5"
##  [9] "fourth.SAM.6"   "first.leaf.2"  "first.SAM.2"   "second.SAM.2"
## [13] "third.leaf.2"   "third.SAM.3"   "third.SAM.4"   "fifth.leaf.2"
## [17] "fifth.leaf.3"   "fifth.SAM.1"   "first.leaf.3"  "second.leaf.1"
## [21] "second.SAM.3"   "third.leaf.3"  "third.SAM.5"   "fourth.leaf.1"
## [25] "fourth.leaf.2" "fifth.leaf.4"  "fifth.SAM.2"   "first.leaf.4"
## [29] "second.leaf.2" "second.SAM.4"  "third.leaf.4"  "third.SAM.6"
## [33] "third.SAM.7"    "fourth.leaf.3" "fourth.SAM.7"  "fifth.SAM.3"
## [37] "first.SAM.3"    "second.leaf.3" "third.leaf.5"  "third.leaf.6"
## [41] "third.leaf.7"   "third.SAM.8"   "fourth.leaf.4" "fourth.SAM.8"
## [45] "fifth.SAM.4"    "first.SAM.4"   "second.leaf.4" "a1.leaf"
## [49] "a1.SAM"         "b2.leaf"       "b2.SAM"        "c3.leaf"
## [53] "c3.SAM"         "d4.leaf"       "d4.SAM"        "e5.leaf"
## [57] "e5.SAM"         "sd"            "average"       "cv"
```

```
quantile(countData$cv) #get quantile use 75% for subsetting top 25%
```

```
##      0%     25%     50%     75%    100%
## 0.00000 0.09877 0.25478 0.61264 3.16228
```

```
countData[is.na(countData)] <- 0 #get rid of NA
subCountData <- subset(countData, cv > 0.61264422) #top 25%

allGenes25 <- subCountData[,c(1,48:60)] #This is the subset of genes we will use for analysis
colnames(allGenes25)[1]<-"gene" #rename first column appropriatly
```

PCA

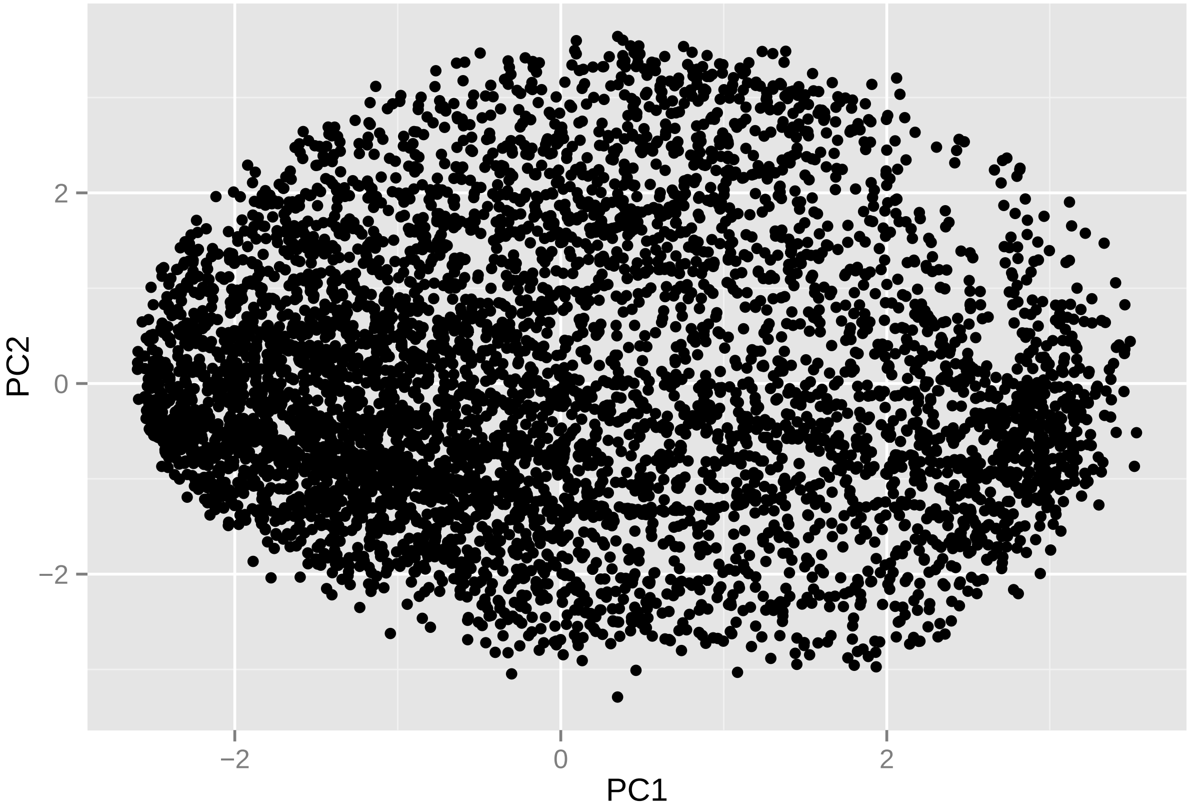
```
#write.csv(allGenes25, "../data/analysis4.top25.csv") #to write out data if needed.
scale_data <- as.matrix(t(scale(t(allGenes25[c(2:11)]))))  #scale data

#Principle Component Analysis
pca <- prcomp(scale_data, scale=TRUE)

summary(pca)
```

```
## Importance of components:
##                          PC1   PC2   PC3   PC4    PC5    PC6    PC7    PC8
## Standard deviation     1.453 1.337 1.089 1.008 0.9302 0.9175 0.8876 0.8735
## Proportion of Variance 0.211 0.179 0.119 0.102 0.0865 0.0842 0.0788 0.0763
## Cumulative Proportion  0.211 0.390 0.508 0.610 0.6966 0.7808 0.8596 0.9359
##                           PC9     PC10
## Standard deviation     0.8009 4.27e-15
## Proportion of Variance 0.0641 0.00e+00
## Cumulative Proportion  1.0000 1.00e+00
```

```
pca.scores <- data.frame(pca$x)

data.val.allGenes25 <- cbind(allGenes25, scale_data, pca.scores)
```

Visualizing the PCA

```
p <- ggplot(data.val.allGenes25, aes(PC1, PC2))
p + geom_point()
```



There are these swooping lines. Not sure what they are. Aashish informs me that they happen often, but if I have time I want to really understand what is causing them.

Self Organizing Map - (6,6) Large

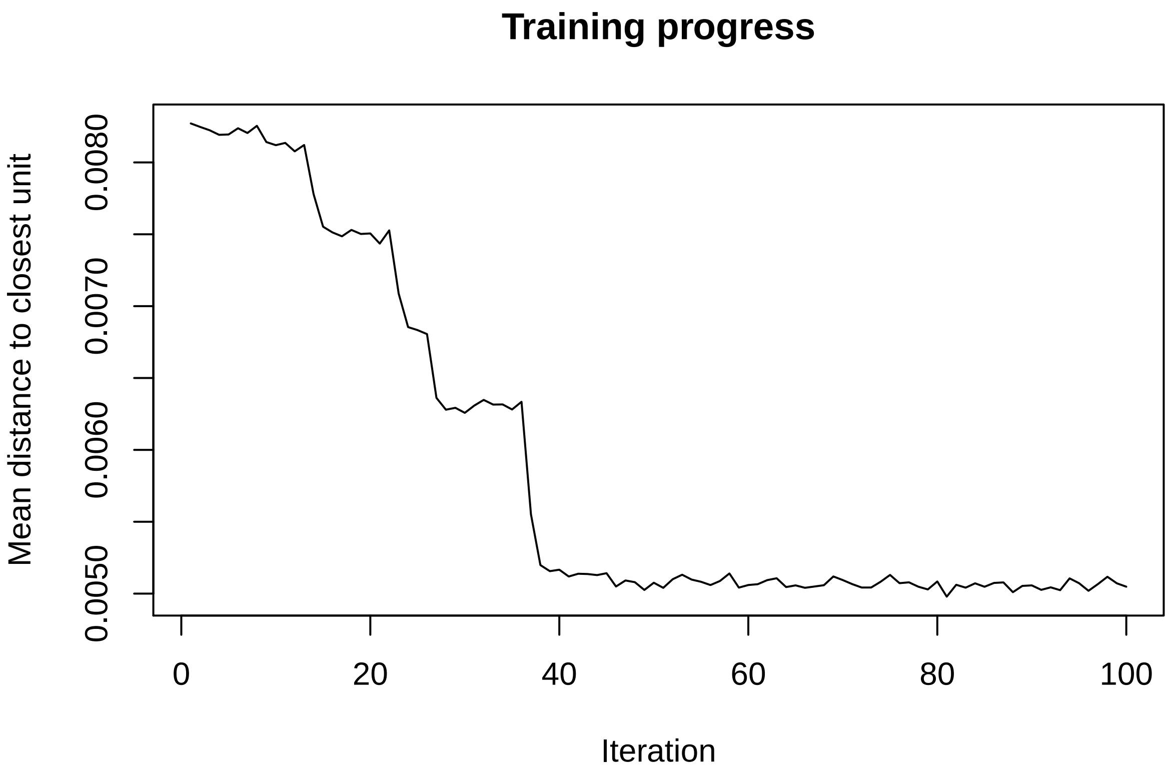
Since we are interested in particular co-expression pattern (up or down through time), I did a large SOM to explicitly find these clusters.

```
data.val <- data.val.allGenes25

som.data <- as.matrix(data.val[,c(15:24)])  #subset only the scaled gene expression values

set.seed(2)

som <- som(data=som.data, somgrid(6,6,"hexagonal")) # This is where you change the size of the map
summary(som)
```

```
## som map of size 6x6 with a hexagonal topology.
## Training data included; dimension is 6935 by 10
## Mean distance to the closest unit in the map: 1.771
```

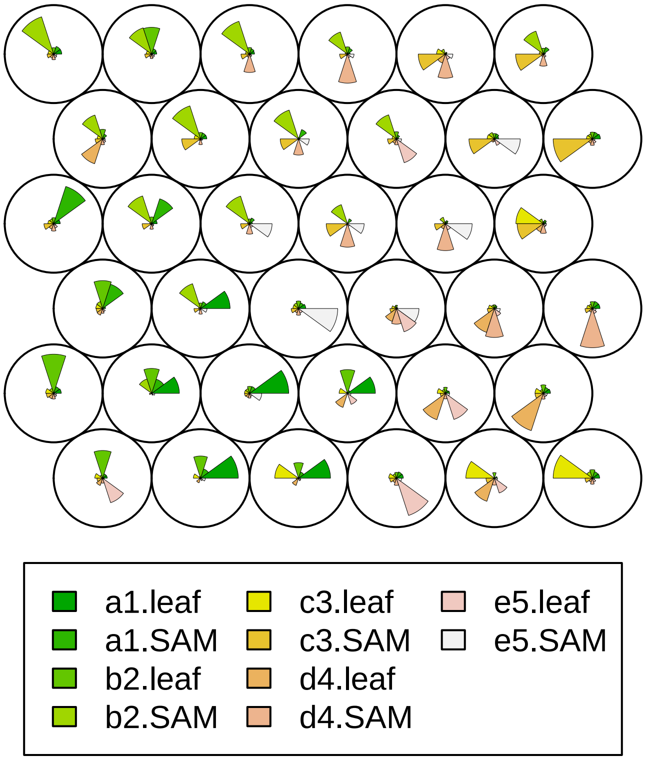
Training Plot ("changes")

```r
plot(som, type ="changes")
```



Code Plot - Large

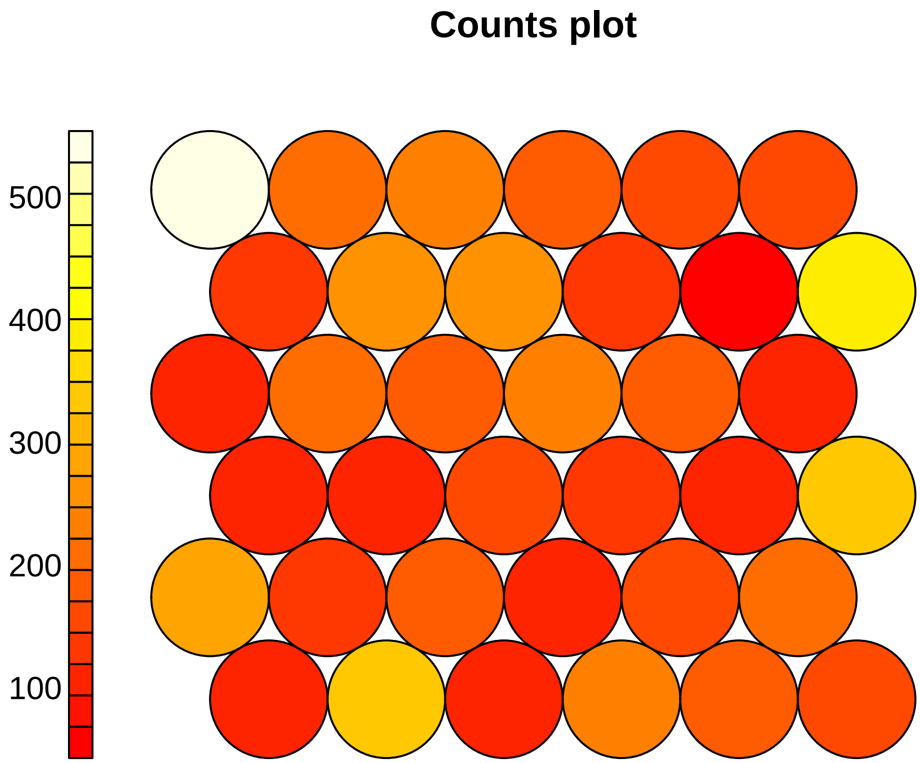
```r
plot(som, type = "codes")
```



Count Plot - Large

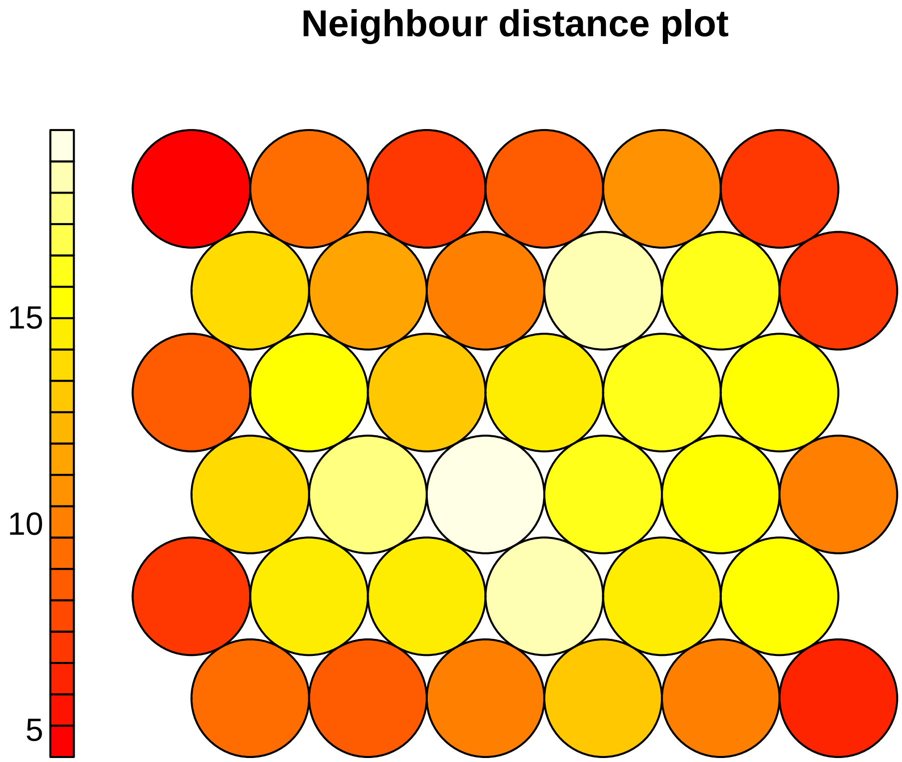
This tells you how many genes are in each of the clusters. The count plot can be used as a quality check. Ideally you want a uniform distribution. If there are some peaks in certain areas, this means you should likely increase the map size. If you have empty nodes you should decrease the map size [1].

```
plot(som, type = "counts")
```



Distance Neighbour Plot - Large

```
plot(som, type="dist.neighbours")
```

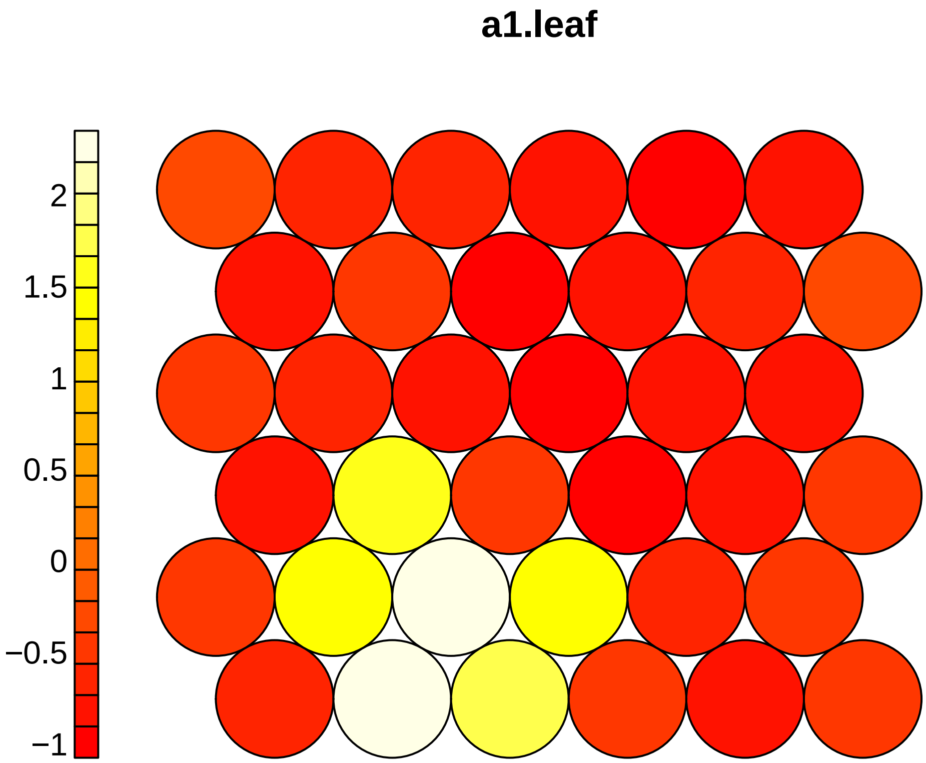


Heatmaps - large

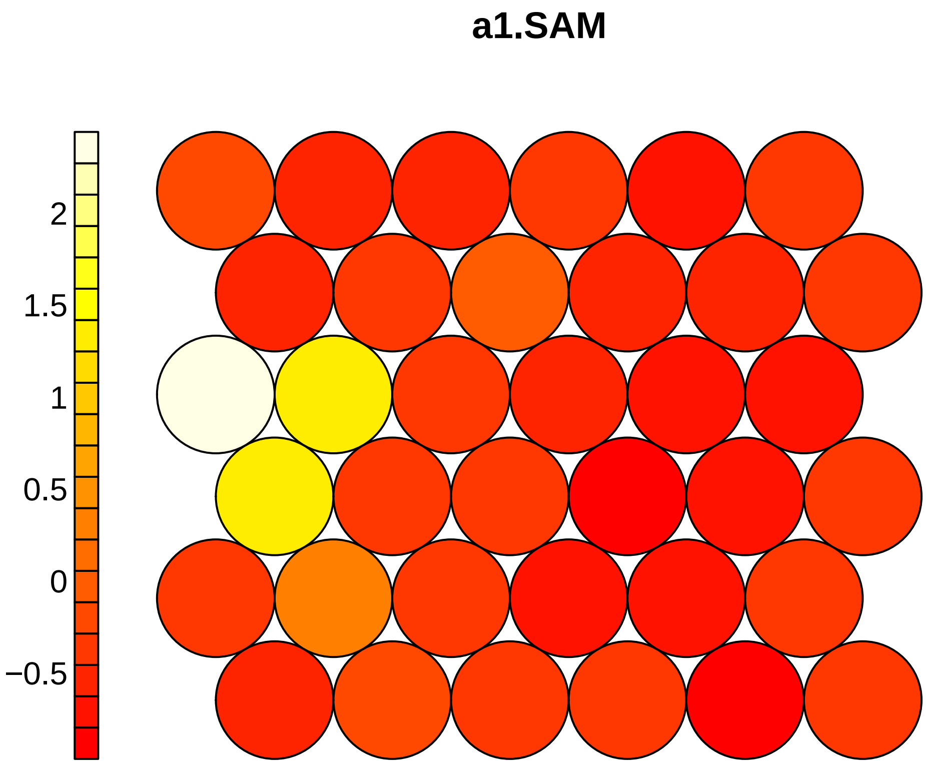
```
head(som$codes) #check
```

```
##         a1.leaf   a1.SAM  b2.leaf   b2.SAM c3.leaf    c3.SAM  d4.leaf   d4.SAM
## [1,] -0.6721 -0.5715  1.5783 -0.4611 -0.1570 -0.32349 -0.05554 -0.3900
## [2,]  2.2210 -0.1206  1.0958 -0.6010 -0.2098 -0.70994 -0.27230 -0.8129
## [3,]  1.7214 -0.4007  0.5164 -0.7382  1.1906 -0.62390 -0.05633 -0.8253
## [4,] -0.4736 -0.3913 -0.3417 -0.3440 -0.1685 -0.23156 -0.29809 -0.1940
## [5,] -0.8321 -0.9644 -0.3583 -0.7596  1.5999 -0.07862  1.26375 -0.2147
## [6,] -0.4335 -0.3615 -0.1052 -0.1374  2.5019 -0.17943 -0.33494 -0.2940
##         e5.leaf   e5.SAM
## [1,]  1.4833 -0.4308
## [2,] -0.4136 -0.1765
## [3,] -0.4024 -0.3815
## [4,]  2.5806 -0.1378
## [5,]  0.6487 -0.3047
## [6,] -0.2832 -0.3728
```

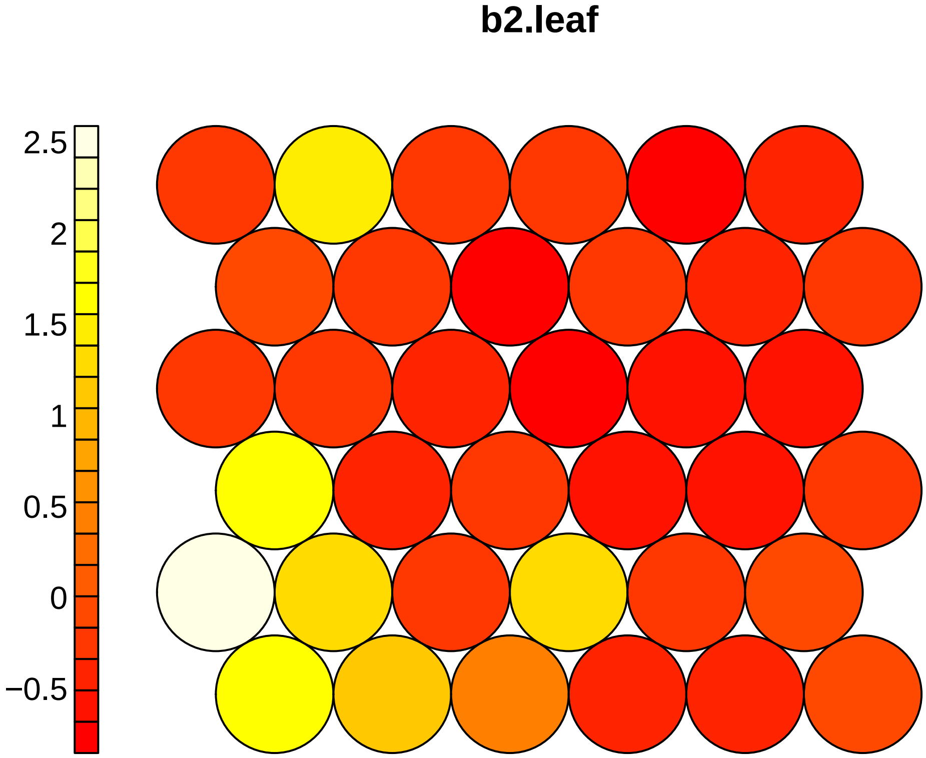
```
som$data <- data.frame(som$data) #changed to dataframe to extract column names easier.

#This is just a loop that plots the distribution of each tissue type across the map.
for (i in 1:10){
  plot(som, type = "property", property = som$codes[,i], main=names(som$data)[i])
  print(plot)
  }
```



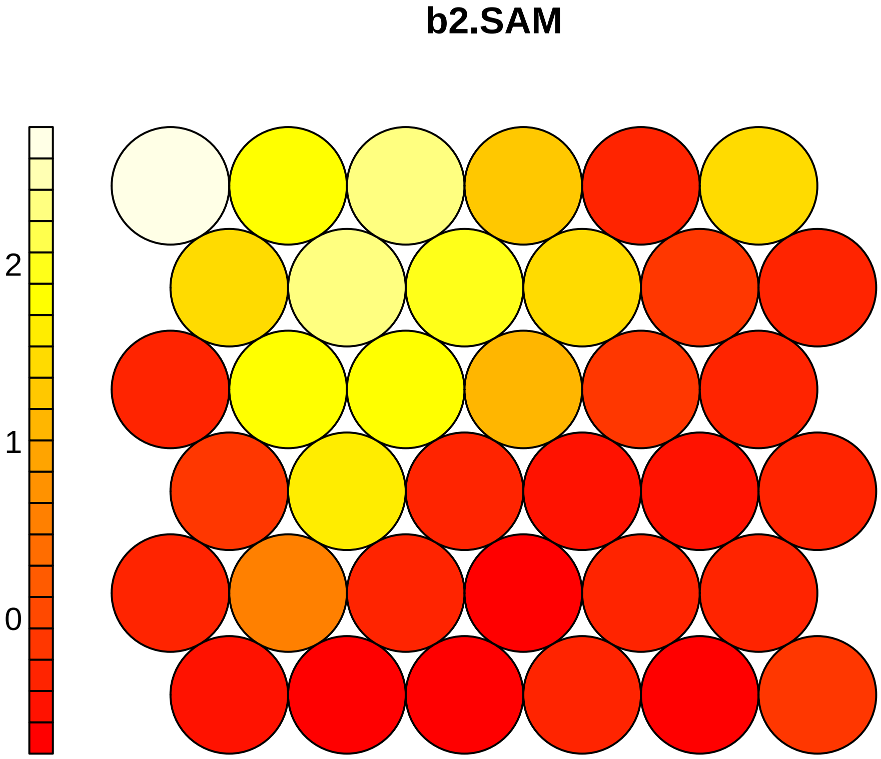
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



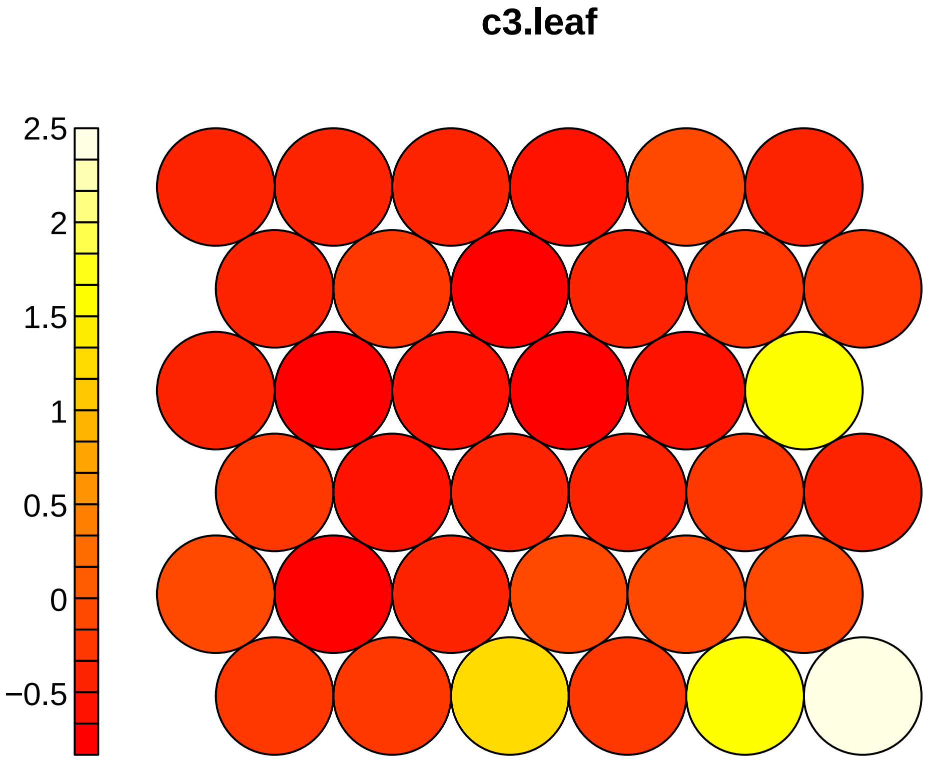
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



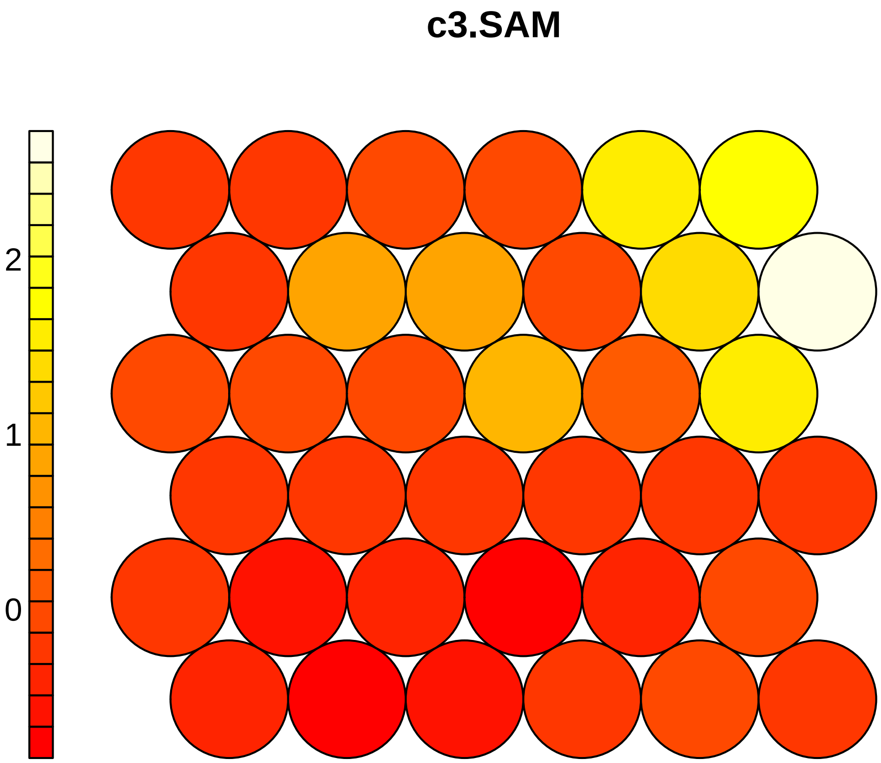
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



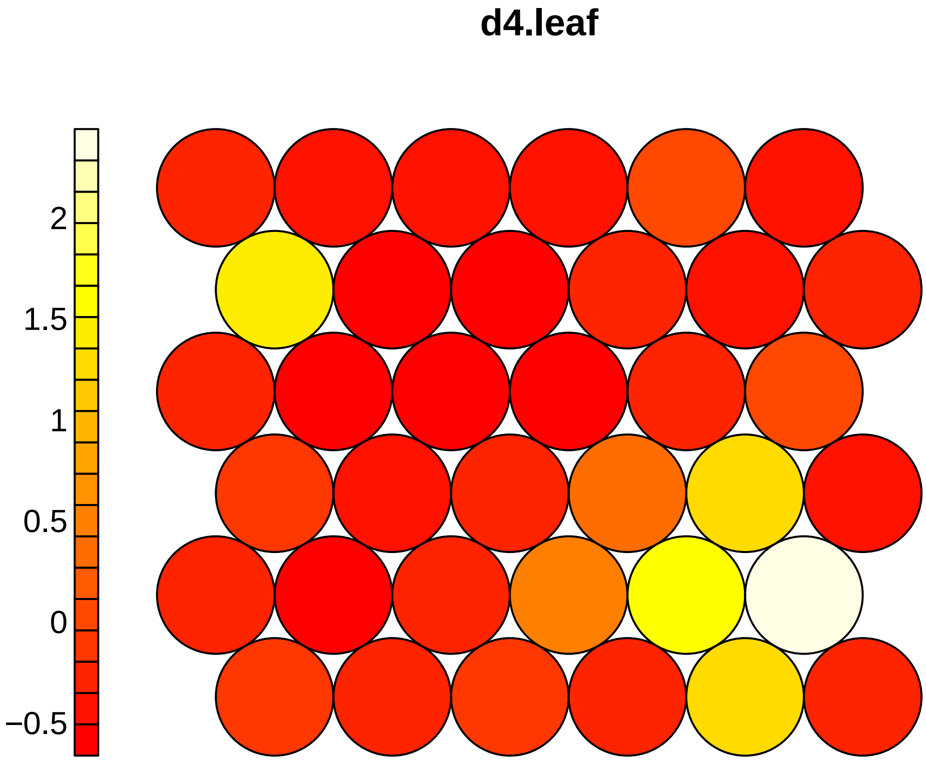
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



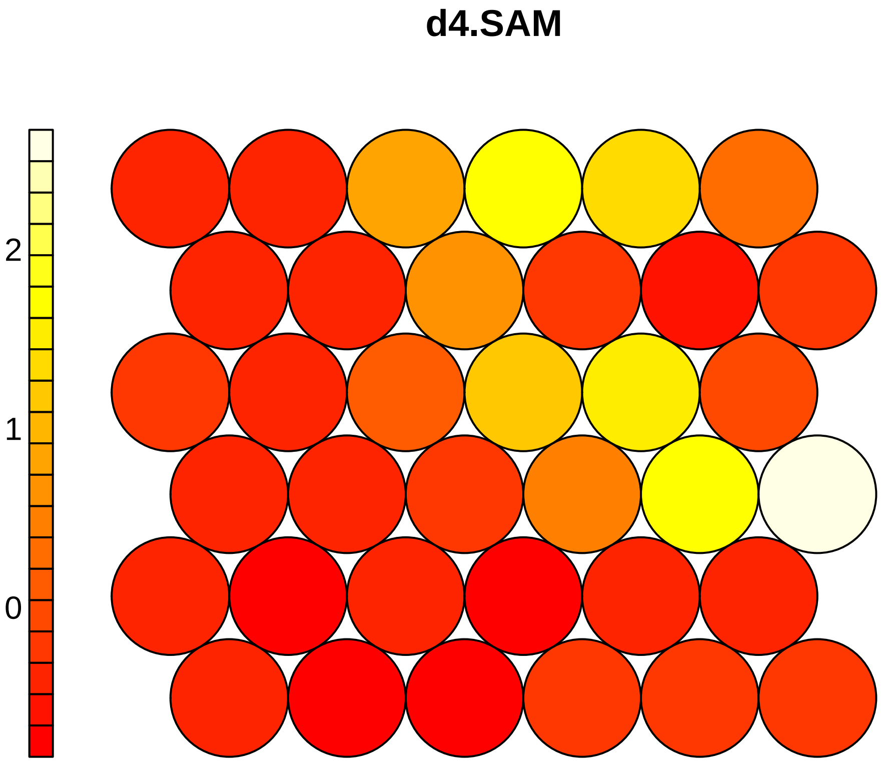
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



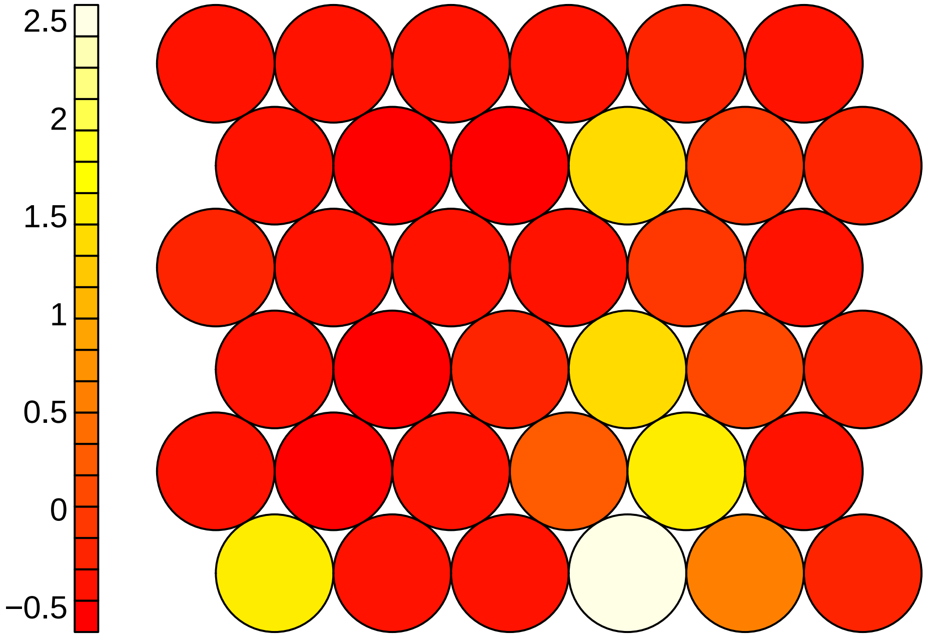
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

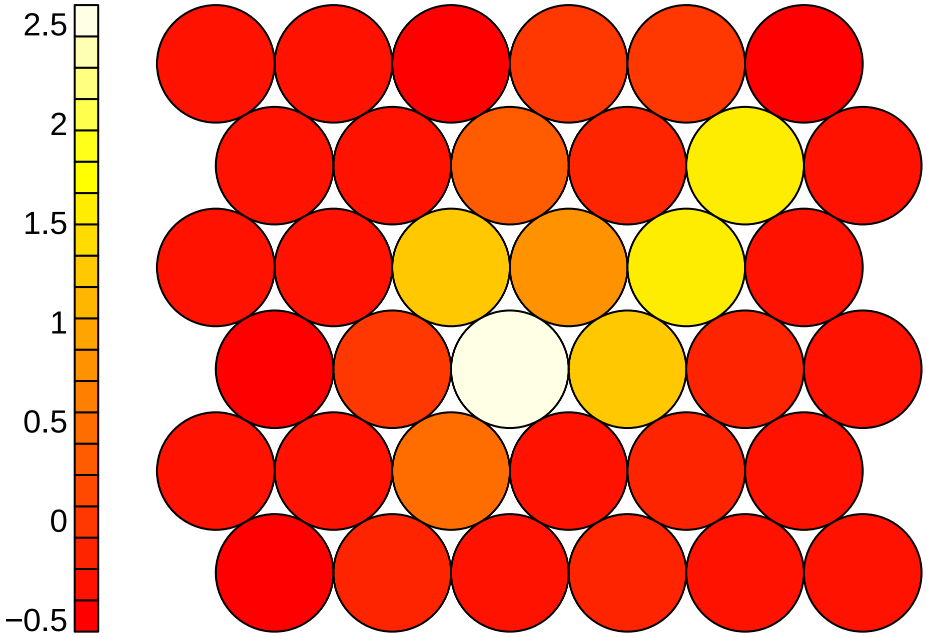


```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



e5.leaf

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



e5.SAM

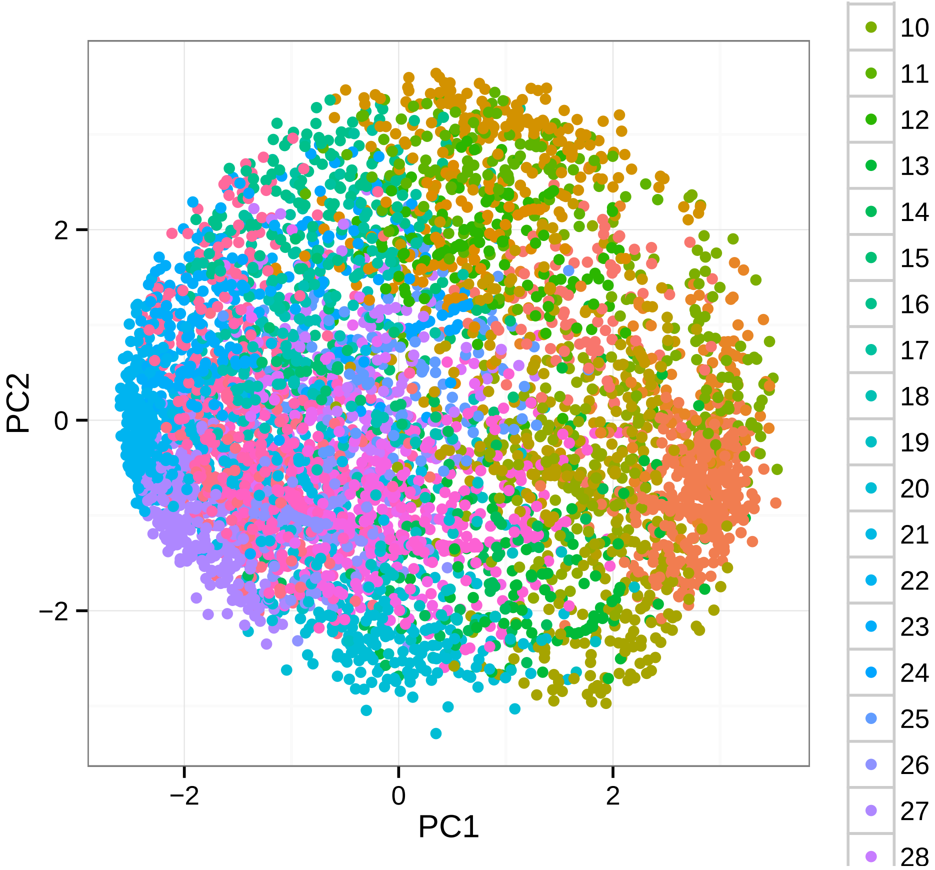
```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

Visualize by Cluster

```
##Bring the datasets back together for cluster specific visualizations
plot.data <- cbind(data.val[,c(1,15:34)],som$unit.classif,som$distances)
names(plot.data) #check
```

```
##  [1] "gene"              "a1.leaf"           "a1.SAM"
##  [4] "b2.leaf"           "b2.SAM"            "c3.leaf"
##  [7] "c3.SAM"            "d4.leaf"           "d4.SAM"
## [10] "e5.leaf"           "e5.SAM"            "PC1"
## [13] "PC2"               "PC3"               "PC4"
## [16] "PC5"               "PC6"               "PC7"
## [19] "PC8"               "PC9"               "PC10"
## [22] "som$unit.classif" "som$distances"
```

```
#too many cluster for anything to meaningful
p <- ggplot(data.val, aes(PC1, PC2, colour=factor(som$unit.classif)))
p + geom_point() + theme_bw()
```

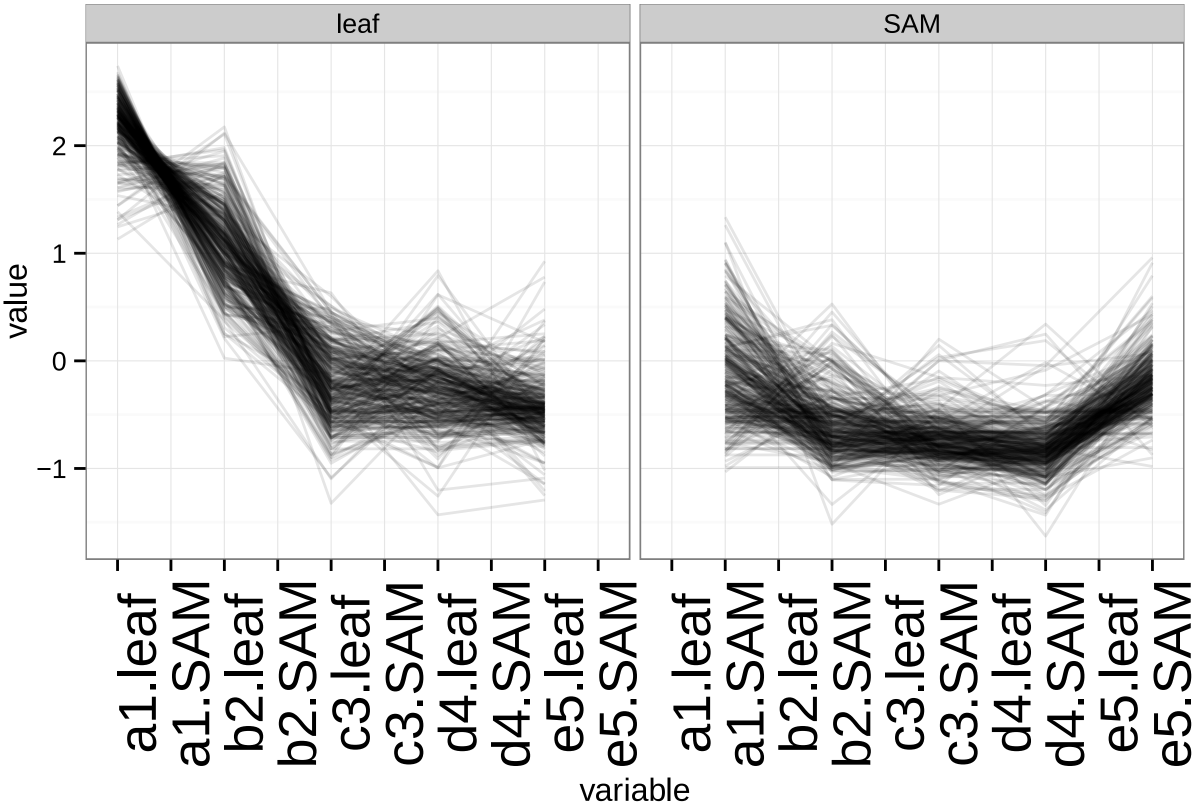


Visualize by cluster

I went through each cluster and tried to identify clusters that have the co-expression pattern we are interested in. The only problem I see with this is that we should possibly run the whole analysis seperatley for each tissue type. We might be able to get more explicit clustering because the other tissue type is not confounding the clustering. The clusters that were the most interesting are clusters 2, 11, 16.

```
# clusterVis_line(1)

clusterVis_line(2) #down through time in leaf
```

```
## Using gene as id variables
```

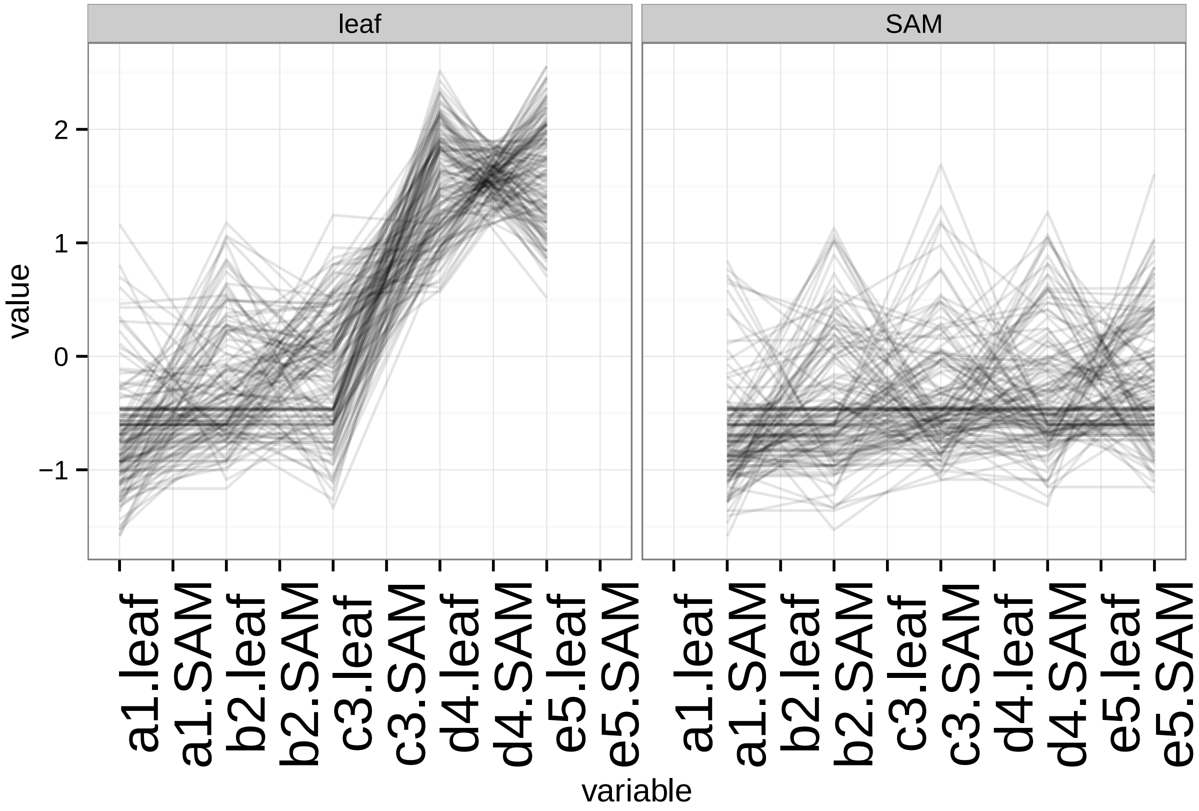


```
#What's in this cluster?
y <- genesInClust(2, plot.data, annotation)
```

```
## [1] 344
```

```
write.csv(y, "../clusterTables/analysis1.cluster2.csv")

# clusterVis_line(3)
# clusterVis_line(4)
# clusterVis_line(5)
# clusterVis_line(6)
# clusterVis_line(7)
# clusterVis_line(8)
# clusterVis_line(9)
# clusterVis_line(10)

clusterVis_line(11)#up through time in leaf
```

```
## Using gene as id variables
```

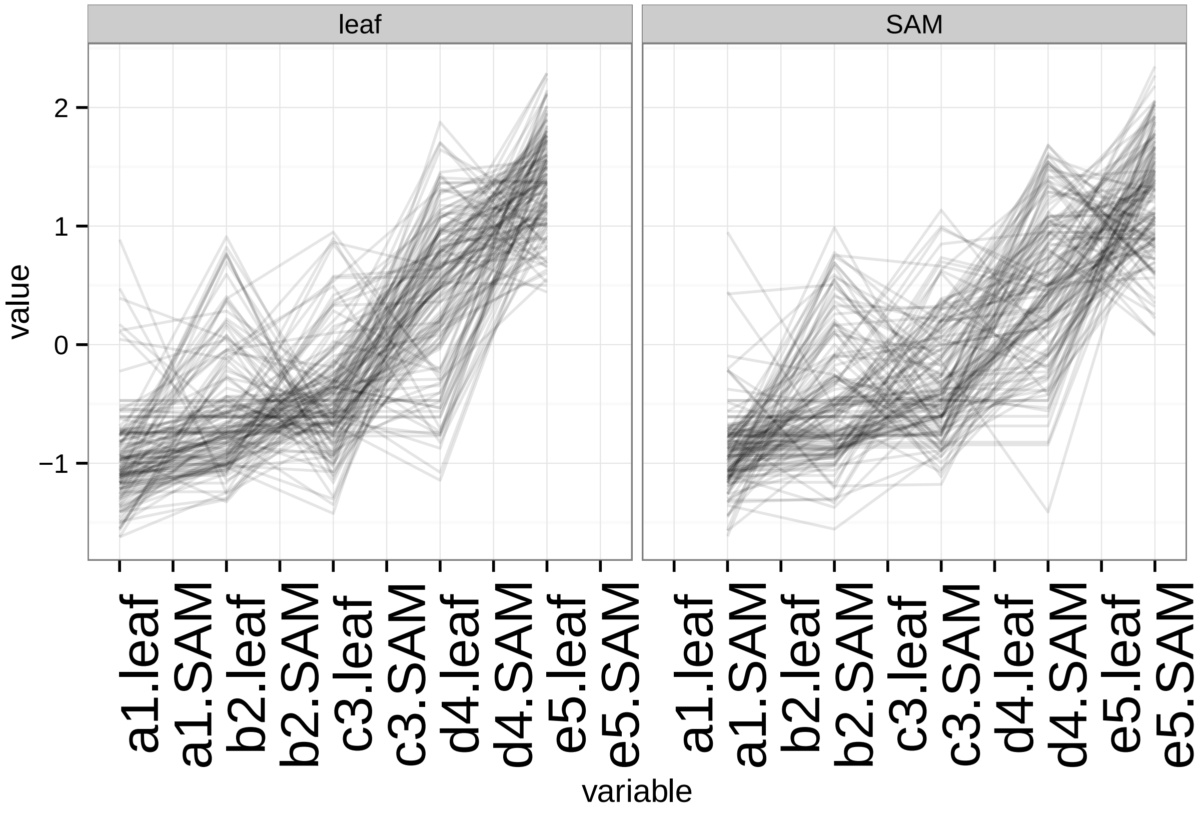


```
#what's in this cluster?
y <- genesInClust(11, plot.data, annotation)
```

```
## [1] 147
```

```
write.csv(y, "../clusterTables/analysis1.cluster11.csv")

# clusterVis_line(12)
# clusterVis_line(13)
# clusterVis_line(14)
# clusterVis_line(15)

clusterVis_line(16) #up through time in both SAM and leaves
```

```
## Using gene as id variables
```



```
#What's in this cluster
y <- genesInClust(16, plot.data, annotation)
```

```
## [1] 141
```

```
write.csv(y, "../clusterTables/analysis1.cluster16.csv")

# clusterVis_line(17)
# clusterVis_line(18)
# clusterVis_line(19)
# clusterVis_line(20)
# clusterVis_line(21)
# clusterVis_line(22)
# clusterVis_line(23)
# clusterVis_line(24)
# clusterVis_line(25)
# clusterVis_line(26)
# clusterVis_line(27)
# clusterVis_line(28)
# clusterVis_line(29)
# clusterVis_line(30)
# clusterVis_line(31)
# clusterVis_line(32)
# clusterVis_line(33)
# clusterVis_line(34)
# clusterVis_line(35)
# clusterVis_line(36)
```

```
#ect
```

Aim 2: Specific Genes

Talking to Dan Chitwood: we need to look into specific genes. Which clusters do they fall into? From Dan via email:

*The idea behind these experiments is a bit abstract, but let me try to convey it simply. 1) KNOXs are up in the leaf primordium in foliar shade. 2) As you would expect from this, leaves are statistically more complex in shade. 3) But shade also modulates the heteroblastic series. There is lots of classical literature on this. 4) Leaf complexity in tomato increases across the heteroblasty series already.

What we didn't know is whether KNOX gene expression increases in the primordia of successive leaves across the heteroblastic series or not. If so, it suggests a mechanism by which shape, heteroblasty, and environmental response are integrated. If not, it suggests that increases in KNOX expression in shade affect leaf shape more than heteroblasty per se for shade, and that mechanisms modulating increases in leaf complexity across the series are not mediated through KNOX genes (a recent commentary Neelima and I wrote on a piece by Detlef suggests that actually TCPs/CUCs mediate heteroblasty more than KNOXs in Arabidopsis).

For starters, how do the following Knotted-like genes behave in your dataset?

Solyc04g077210.2.1 Solyc05g005090.2.1 Solyc01g100510.2.1 Solyc11g069890.1.1 Solyc02g081120.2.1

Other genes to consider are the most significant in Dataset S2, which are those differentially expressed between constant sun and 28hr shade swapped leaf primordia.**

Make lists of genes.

```
#Genes that are differentially expressed between constand sum and 28 hr shade swapped leaf primordia vi

v9 <- read.csv("../data/DE_v9_DatasetS2.csv")
dim(v9) #there are 645 genes in this list
```

```
## [1] 645  14
```

```
#isolate the first column
v9.ITAG <- as.data.frame(v9[,1])
colnames(v9.ITAG)[1] <- "ITAG"
```

Merge `data.val` into each of these lists, do not keep the non-overlapp.

```
dim(plot.data)#check
```

```
## [1] 6935   23
```

```
names(plot.data)
```

```
##  [1] "gene"             "a1.leaf"          "a1.SAM"          
##  [4] "b2.leaf"          "b2.SAM"           "c3.leaf"         
##  [7] "c3.SAM"           "d4.leaf"          "d4.SAM"          
## [10] "e5.leaf"          "e5.SAM"           "PC1"             
## [13] "PC2"              "PC3"              "PC4"             
## [16] "PC5"              "PC6"              "PC7"             
## [19] "PC8"              "PC9"              "PC10"            
## [22] "som$unit.classif" "som$distances"
```

```
plot.data2 <- plot.data
colnames(plot.data2)[1]<-"ITAG"

dim(v9.ITAG) #check
```

```
## [1] 645   1
```

```
v9.cluster <- merge(v9.ITAG, plot.data2, by = "ITAG")
dim(v9.cluster) #check
```

```
## [1] 212  23
```

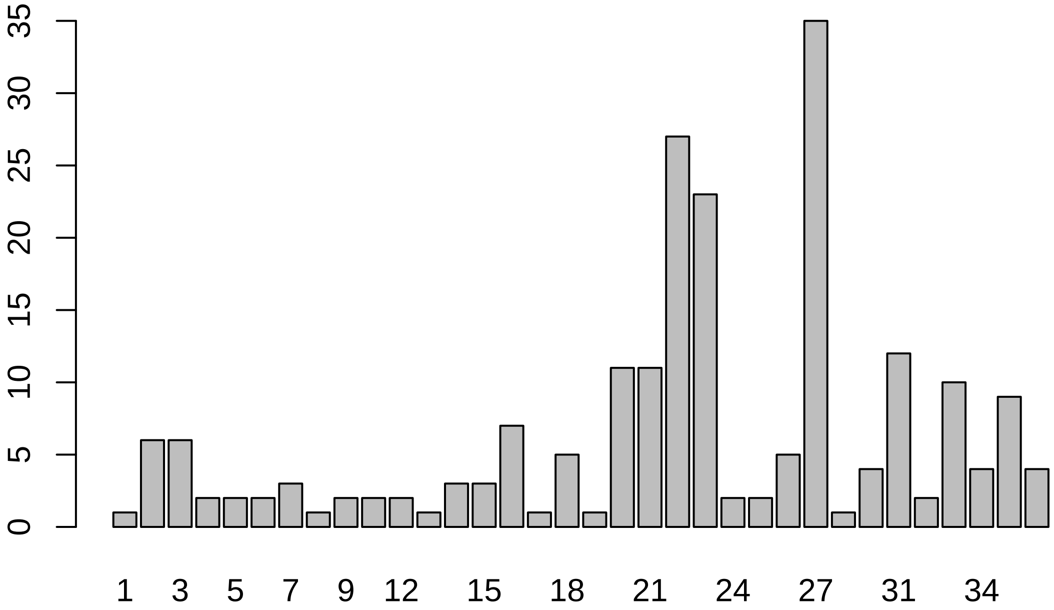
```
#Get only needed columns
v9.clusterIDs <- v9.cluster[,c(1,22,23)]
colnames(v9.clusterIDs)[2]<-"cluster"

#Visualize how many genes fall into which cluster
str(v9.clusterIDs) #need cluster to be factor
```

```
## 'data.frame':    212 obs. of  3 variables:
##  $ ITAG         : Factor w/ 645 levels "Solyc00g005050.2.1",..: 15 16 21 22 32 34 35 43 46 48 ...
##  $ cluster      : int  6 3 22 7 35 27 24 21 15 26 ...
##  $ som$distances: num  0.237 1.118 0.603 0.141 2.069 ...
```

```
v9.clusterIDs$cluster <- as.factor(v9.clusterIDs$cluster)

summary(v9.clusterIDs)
```

```
##                   ITAG        cluster   som$distances
##  Solyc01g007410.2.1:  1   27     :35   Min.   :0.017
##  Solyc01g007500.2.1:  1   22     :27   1st Qu.:0.524
##  Solyc01g010150.2.1:  1   23     :23   Median :1.100
##  Solyc01g014250.2.1:  1   31     :12   Mean   :1.372
##  Solyc01g073770.2.1:  1   20     :11   3rd Qu.:1.799
##  Solyc01g079950.2.1:  1   21     :11   Max.   :6.616
##  (Other)           :206   (Other):93
```

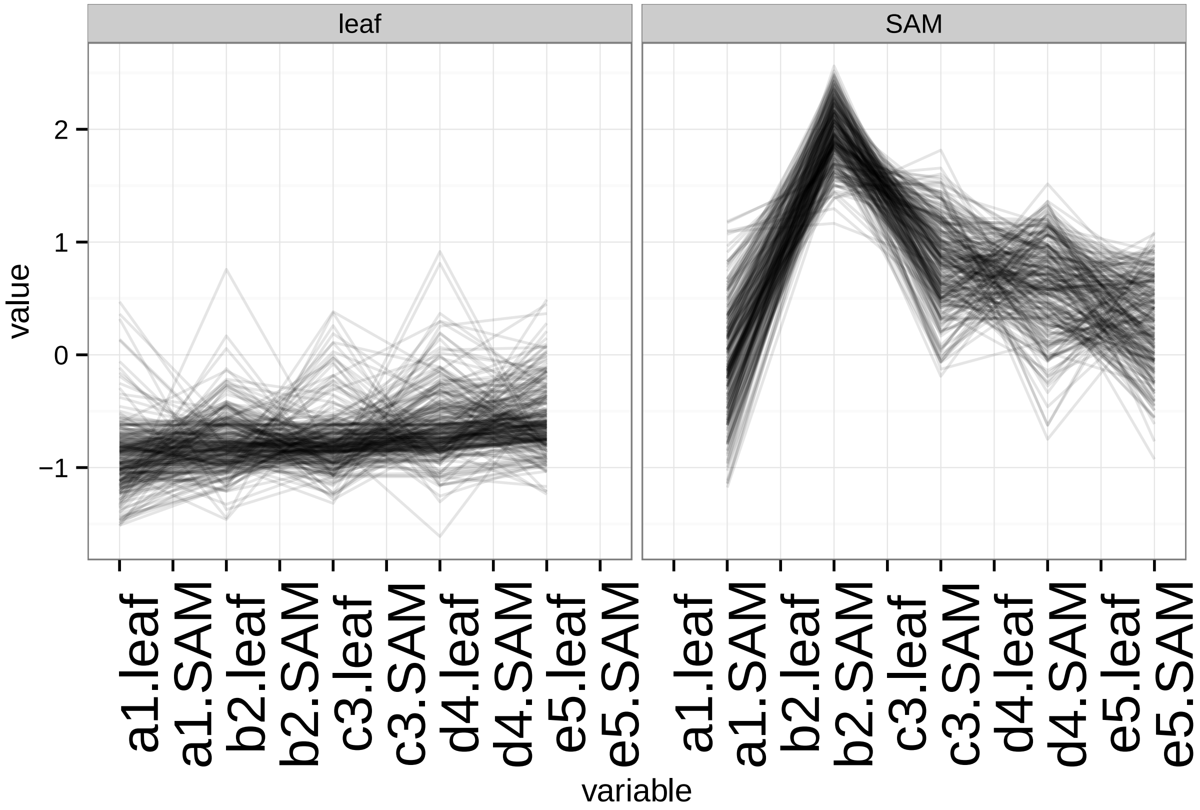
```
plot(v9.clusterIDs$cluster) #possible enriched in cluster #27, 24, and 23? Is there a way to statistica
```



There are 35 genes that are in cluster 27, but is this due to cluster size? What genes are in these clusters?

```
clusterVis_line(27)
```

```
## Using gene as id variables
```

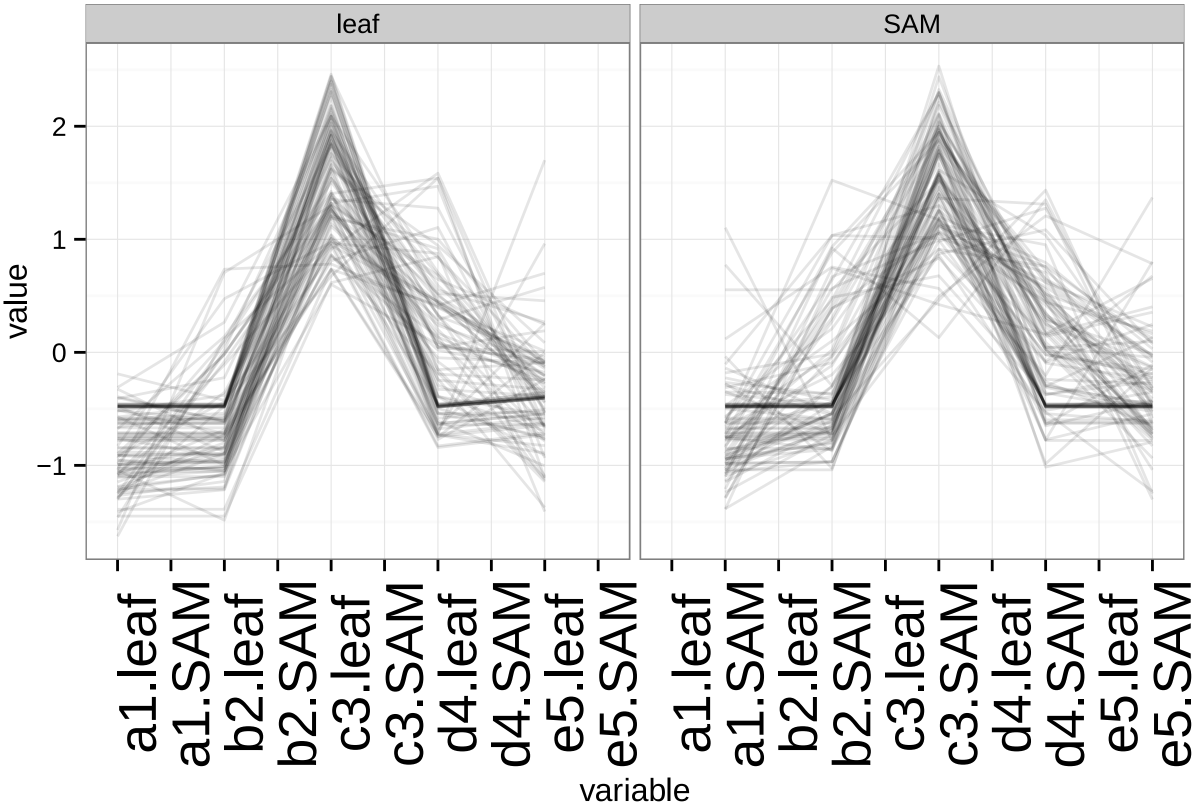


```
y <- genesInClust(27, plot.data, annotation)
```

```
## [1] 263
```

```
write.csv(y, "../clusterTables/analysis1.cluster27.csv")

clusterVis_line(24)
```

```
## Using gene as id variables
```

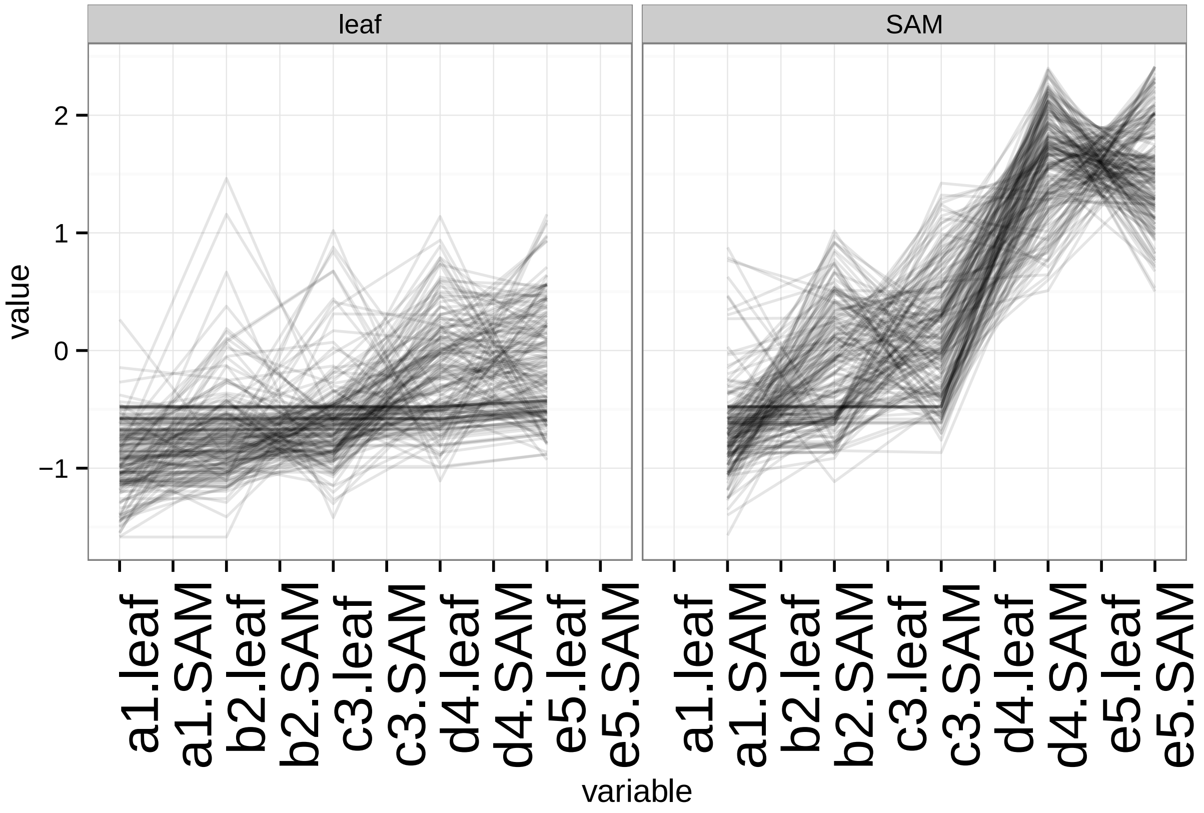


```
y <- genesInClust(24, plot.data, annotation)
```

```
## [1] 97
```

```
write.csv(y, "../clusterTables/analysis1.cluster24.csv")

clusterVis_line(23)
```

```
## Using gene as id variables
```



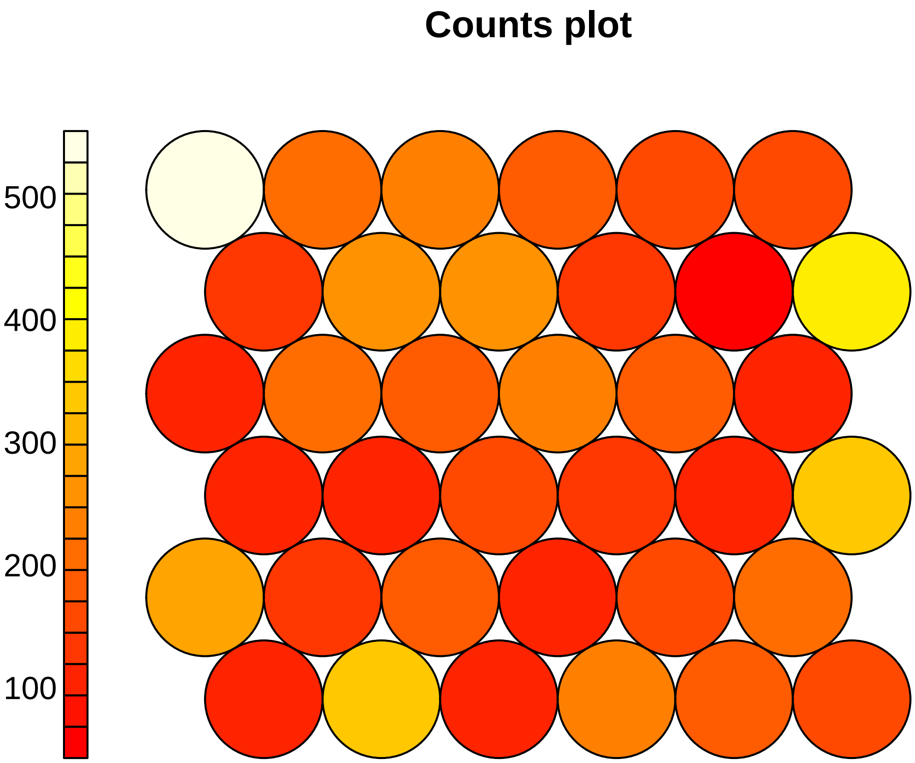
```
y <- genesInClust(23, plot.data, annotation)
```

```
## [1] 188
```

```
write.csv(y, "../clusterTables/analysis1.cluster23.csv")
```

Yes, cluster 27 is larger than the rest, but how large is it compared to other clusters in the SOM?

```
plot(som, type = "counts")
```



About Average. Are there statistics that can be done with this? What does the gene expression pattern in these clusters even mean?

Knotted - like genes

```
#Knotted-like
ITAG <- c("Solyc04g077210.2.1","Solyc05g005090.2.1","Solyc01g100510.2.1", "Solyc11g069890.1.1", "Solyc0

knottedGenes <- data.frame(ITAG)
#head(knottedGenes)

#names(plot.data2)
#names(knottedGenes)

knot.cluster <- merge(knottedGenes, plot.data2, by = "ITAG")

#Get only needed columns
names(knot.cluster)
```

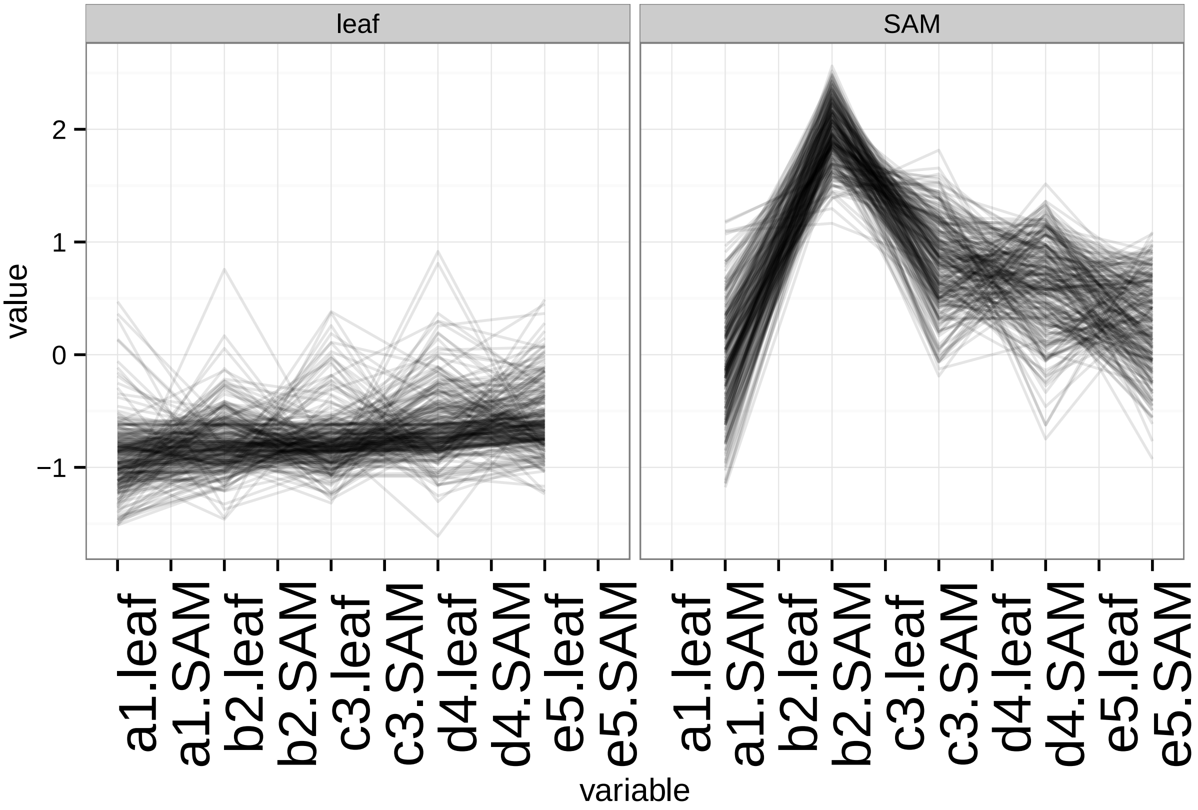
```
##  [1] "ITAG"              "a1.leaf"           "a1.SAM"
##  [4] "b2.leaf"           "b2.SAM"            "c3.leaf"
##  [7] "c3.SAM"            "d4.leaf"           "d4.SAM"
## [10] "e5.leaf"           "e5.SAM"            "PC1"
## [13] "PC2"               "PC3"               "PC4"
## [16] "PC5"               "PC6"               "PC7"
## [19] "PC8"               "PC9"               "PC10"
## [22] "som$unit.classif" "som$distances"
```

```
knot.clusterIDs <- knot.cluster[,c(1,22,23)]

knot.clusterIDs #clusters 21, 23, 27
```

```
##                    ITAG som$unit.classif som$distances
## 1 Solyc01g100510.2.1                22        0.5608
## 2 Solyc02g081120.2.1                21        1.6558
## 3 Solyc04g077210.2.1                27        0.3147
## 4 Solyc05g005090.2.1                27        0.4463
## 5 Solyc11g069890.1.1                27        0.5256
```

```
#Three out of five of them are cluster 27.

clusterVis_line(27)
```

```
## Using gene as id variables
```



Looking at the genes individually

Take the v9 subset and visualize the output all together, this is a little useless.

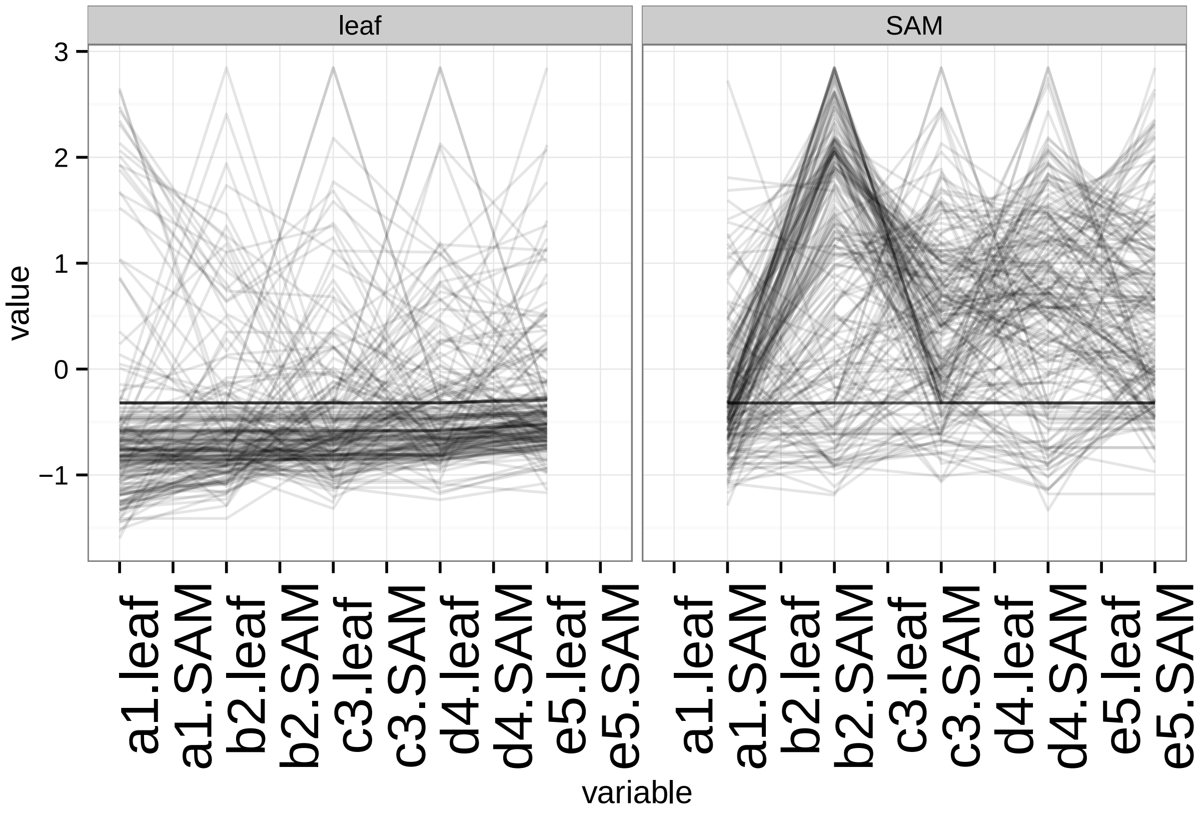
```
names(v9.cluster) #check
```

```
##  [1] "ITAG"             "a1.leaf"          "a1.SAM"
##  [4] "b2.leaf"           "b2.SAM"            "c3.leaf"
##  [7] "c3.SAM"            "d4.leaf"           "d4.SAM"
## [10] "e5.leaf"           "e5.SAM"            "PC1"
## [13] "PC2"               "PC3"               "PC4"
## [16] "PC5"               "PC6"               "PC7"
## [19] "PC8"               "PC9"               "PC10"
## [22] "som$unit.classif" "som$distances"
```

```
#Visualize
  sub_data <- v9.cluster[,c(1:11)] # just the sample types
  m.data <- melt(sub_data)
```

```
## Using ITAG as id variables
```

```
  m.data$region <- ifelse(grepl("SAM", m.data$variable, ignore.case = T), "SAM",
                         ifelse(grepl("leaf", m.data$variable, ignore.case = T), "leaf", "other"))
  m.data <- within(m.data, lineGroup <- paste(ITAG,sep='.'))
  ggplot(m.data, aes(variable, value, group = lineGroup)) +
    geom_line(alpha = .1) +
    geom_point(alpha = .0) +
    theme_bw() +
    facet_grid(.~region) +
    theme(axis.text.x = element_text(size=20,
                                     angle=90,
                                     vjust=1))
```



Maybe it is better to just visualize the genes from supp. v9 in cluster 27.

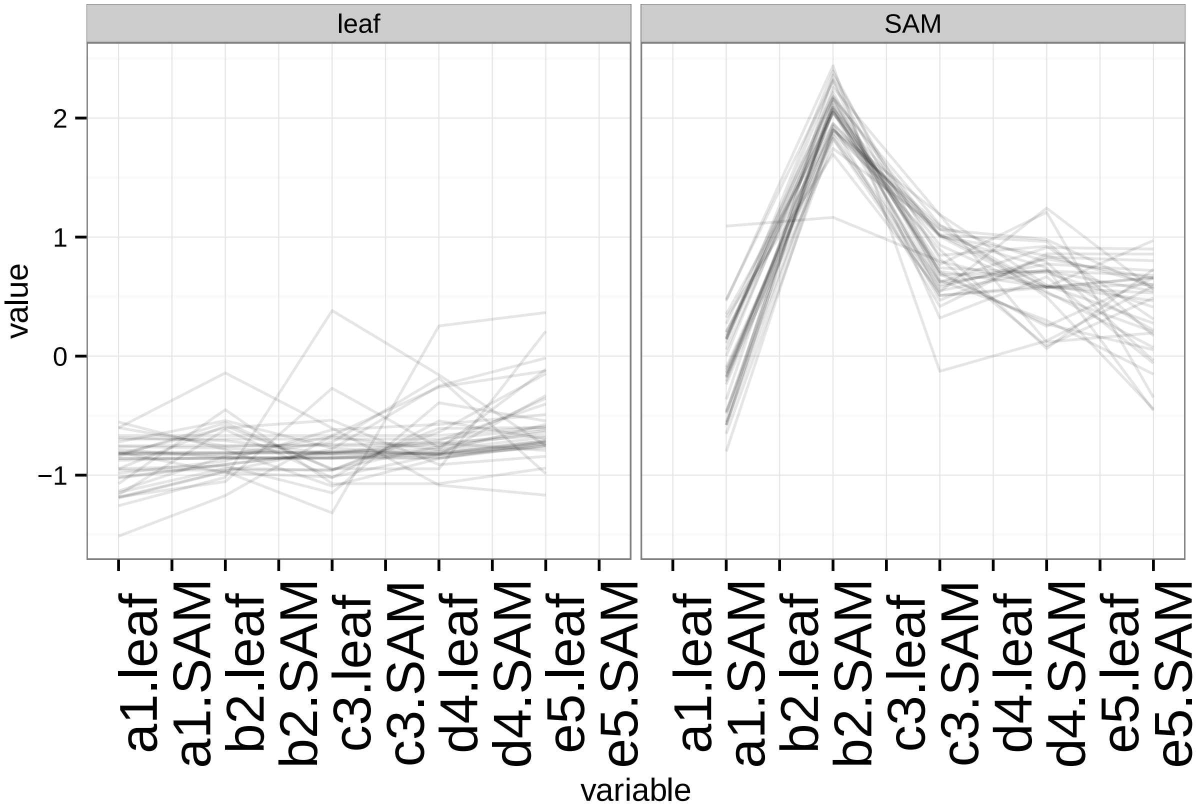
```
v9.cluster2 <- v9.cluster
colnames(v9.cluster2)[22]<-"cluster"
v9.cluster.sub <- subset(v9.cluster2, cluster == "27")
dim(v9.cluster.sub)
```

```
## [1] 35 23
```

```
  sub_data <- v9.cluster.sub[,c(1:11)] # just the sample types
  m.data <- melt(sub_data)
```

```
## Using ITAG as id variables
```

```
  m.data$region <- ifelse(grepl("SAM", m.data$variable, ignore.case = T), "SAM",
                          ifelse(grepl("leaf", m.data$variable, ignore.case = T), "leaf", "other"))
  m.data <- within(m.data, lineGroup <- paste(ITAG,sep='.'))
  ggplot(m.data, aes(variable, value, group = lineGroup)) +
    geom_line(alpha = .1) +
    geom_point(alpha = .0) +
    theme_bw() +
    facet_grid(.~region) +
    theme(axis.text.x = element_text(size=20,
                                     angle=90,
                                     vjust=1))
```



Overall Results and Future Analysis:

There are several clusters that could be looked at more closely. These are in the `clusterTables` directory. The clusters that were picked out for up or down regulation trends per tissue are clusters 2, 11 and 16. The clusters that were identified for “enrichment” of v9.supplementary genes are 27, 24, and 23. Cluster 27 not only had the most gene overlapp of the v9.supplementary genes, but also contained 3 out of 5 of the knotted-like genes. The expression pattern in this cluster is somewhat confusing though.

The clustering may be confounded between SAM and leaf tissue being forced into same cluster. I think looking at each of these tissues seperatley could be useful. See `dclcmSOM_analysis2_102814.Rmd`.

Also, varying SOM sizes could yield more explicit gene expression patterns if larger SOM or allow the ability to do GO-enrichemnt/promoter enrichment if smaller SOM.