Analysis 1 - Top 25% of coefficient of variation - SuperLarge 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. This time I am looking at superLarge SOMs to see if I can get more specific expression patterns.

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

Cluster visualization functions. These are functions that are re-used throughout analysis. These are not printed out in reports to save space. See .Rmd file for specifics of these functions.

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.

clusterVis_region

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

not finished.

genesInCluster()

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

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

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

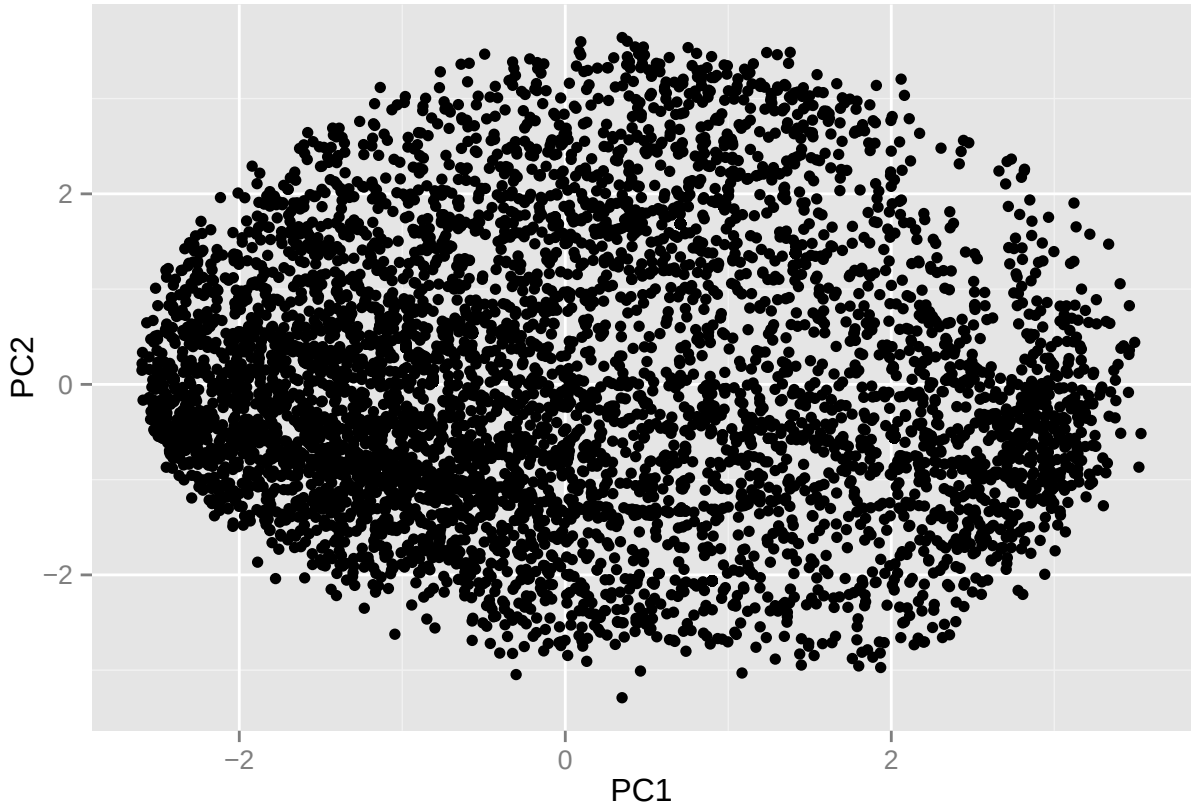
```
#write.csv(allGenes25, "../data/analysis4.top25.csv") #to write out data if needed.
scale_data <- as.matrix(t(scale(t(allGenes25[c(2:11)]))))  #scale data

#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()
```



Self Organizing Map - (6,6) Large

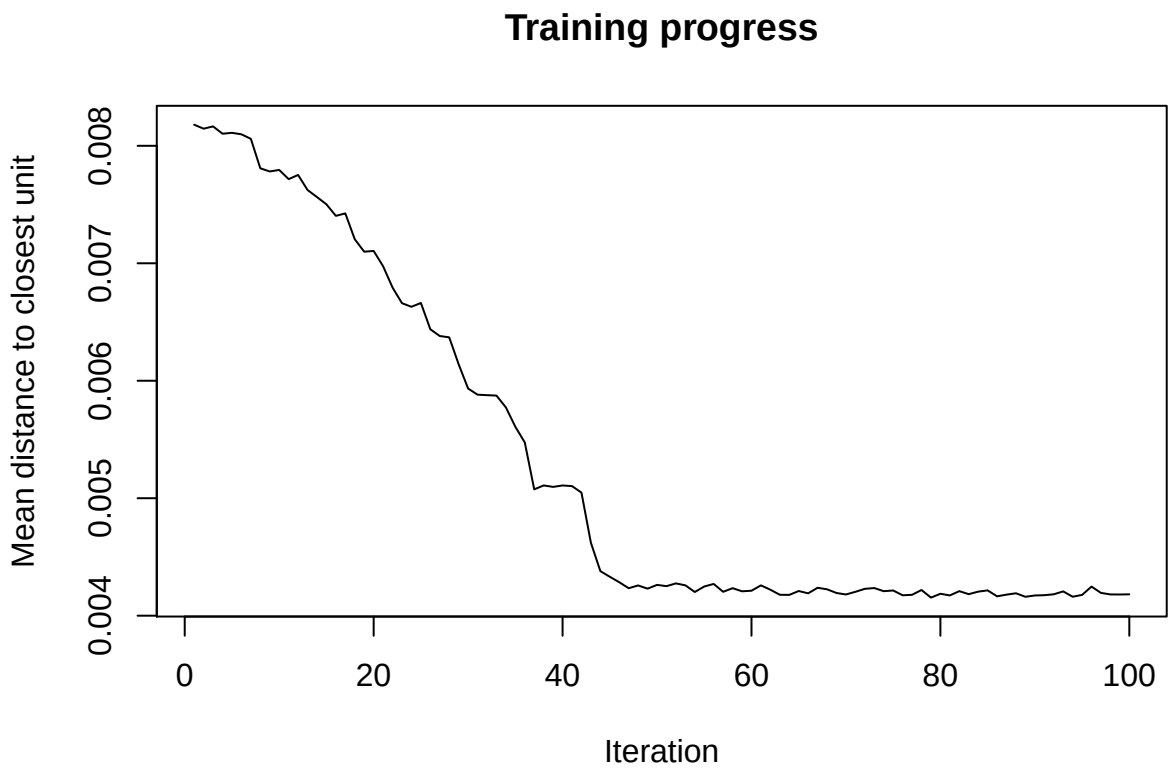
Since we are interested in particular co-expression pattern (up or down through time), I did a large SOM to explicitly find these clusters.

```r
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(10,10,"hexagonal")) # This is where you change the size of the map
summary(som)
```

```
## som map of size 10x10 with a hexagonal topology.
## Training data included; dimension is 6935 by 10
## Mean distance to the closest unit in the map: 1.213
```

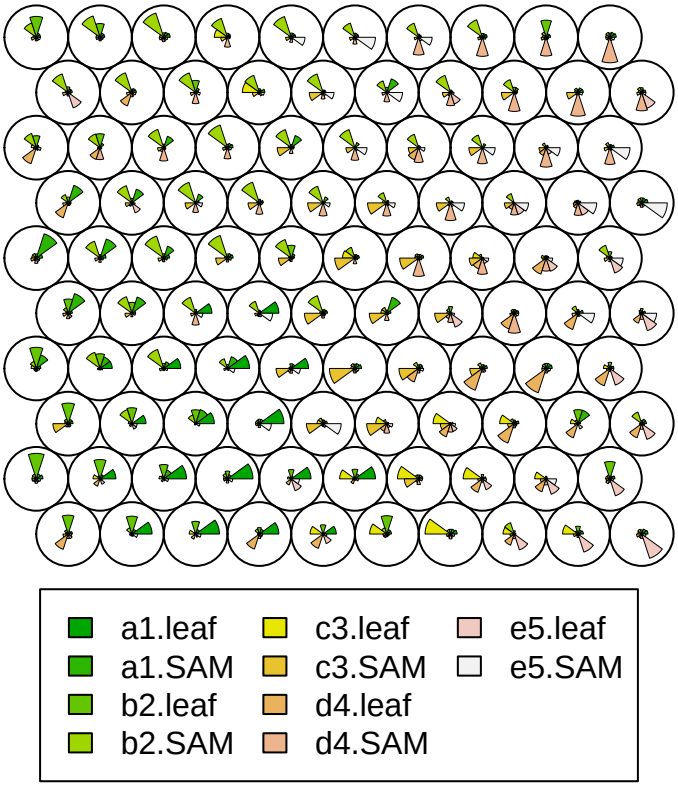
Training Plot ("changes")

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



Code Plot - Large

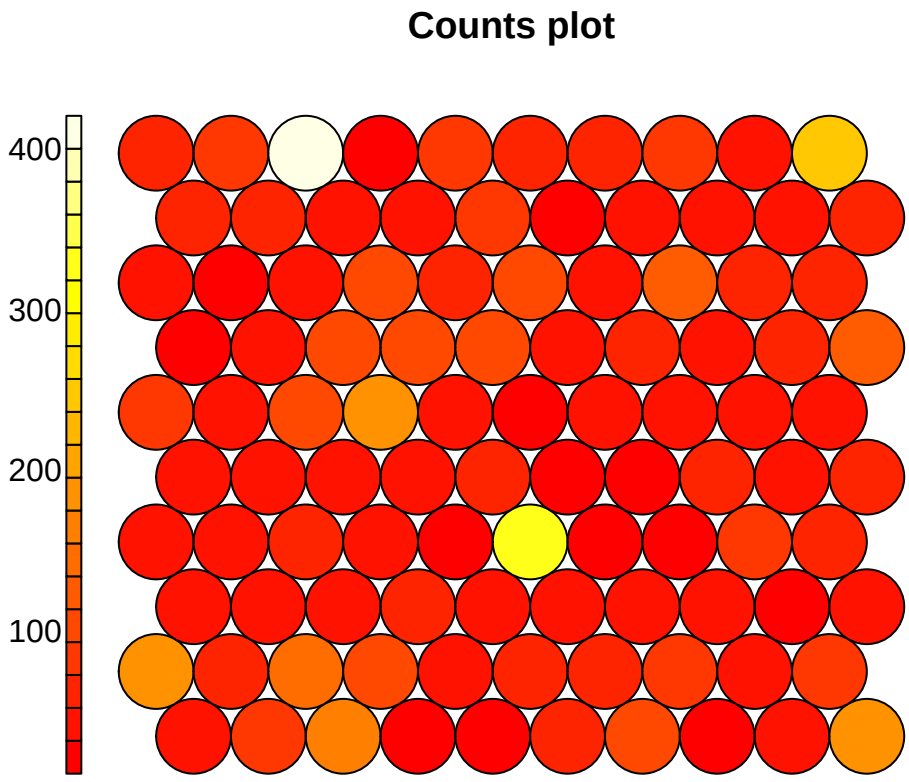
```r
plot(som, type = "codes")
```



Count Plot - superLarge

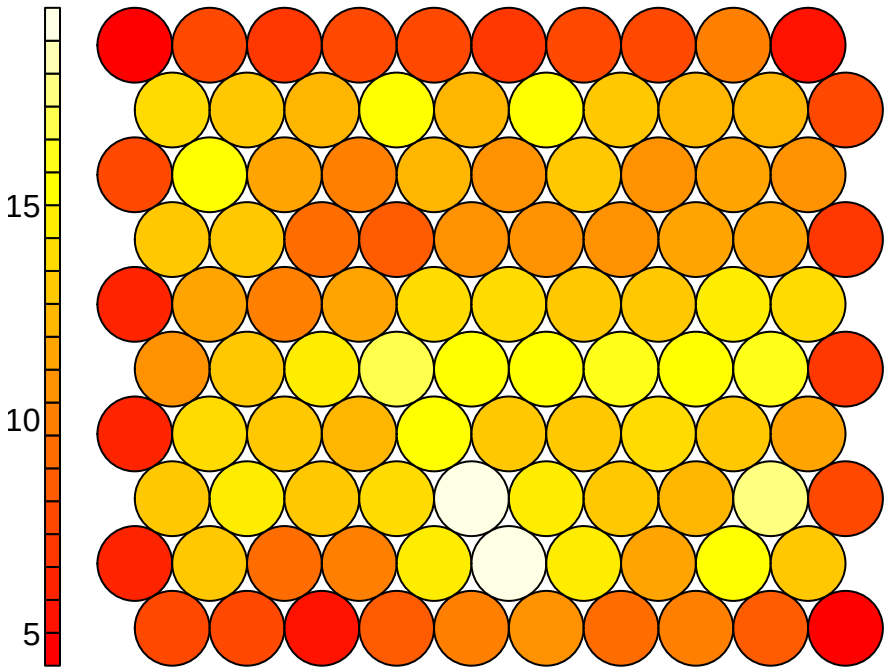
This tells you how many genes are in each of the clusters. The count plot can be used as a quality check. Ideally you want a uniform distribution. If there are some peaks in certain areas, this means you should likely increase the map size. If you have empty nodes you should decrease the map size [1].

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



Distance Neighbour Plot - superLarge

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



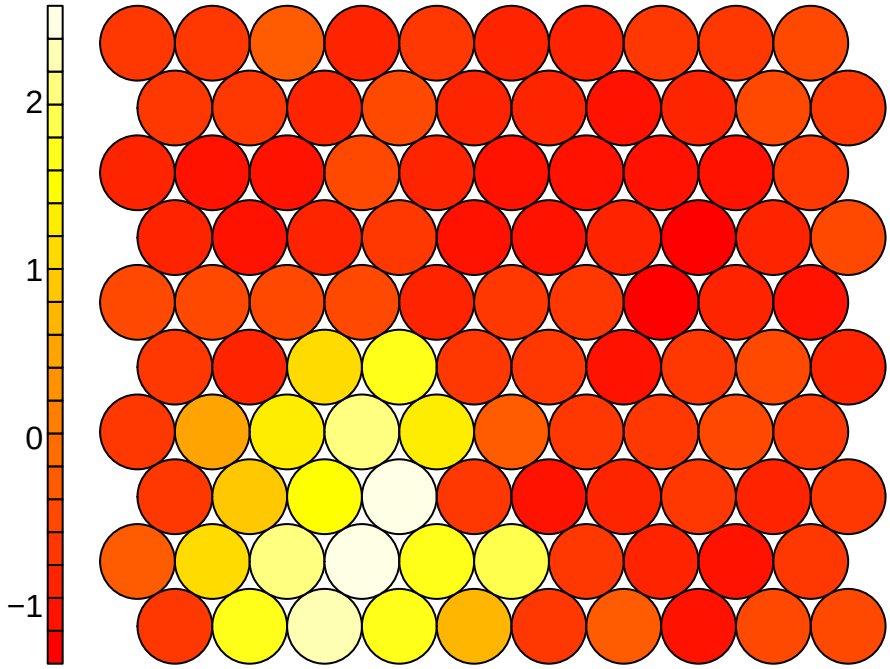
Heatmaps - superlarge

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

```
##      a1.leaf  a1.SAM  b2.leaf   b2.SAM  c3.leaf  c3.SAM  d4.leaf  d4.SAM
## [1,] -0.6952 -0.6491  1.81167 -0.41045 -0.51743 -0.1745  1.44025 -0.3049
## [2,]  1.7520 -0.3638  1.66058 -0.09341 -0.61023 -0.6058 -0.55284 -0.6689
## [3,]  2.3078 -0.3901  0.90455 -0.78526  0.02347 -0.7519 -0.04354 -0.8426
## [4,]  1.6437 -0.3390 -0.01588 -0.28023 -0.66291 -0.6980  1.39681 -0.3056
## [5,]  0.6812 -0.8386  0.38970 -1.09311  0.95518 -0.9378  1.29179 -0.6932
## [6,] -0.6255 -0.3893  1.73113 -0.09917  1.54791 -0.3688 -0.25504 -0.5603
##       e5.leaf  e5.SAM
## [1,] -0.09448 -0.4058
## [2,] -0.43366 -0.0840
## [3,] -0.30177 -0.1206
## [4,] -0.30569 -0.4332
## [5,]  0.59648 -0.3517
## [6,] -0.44025 -0.5407
```

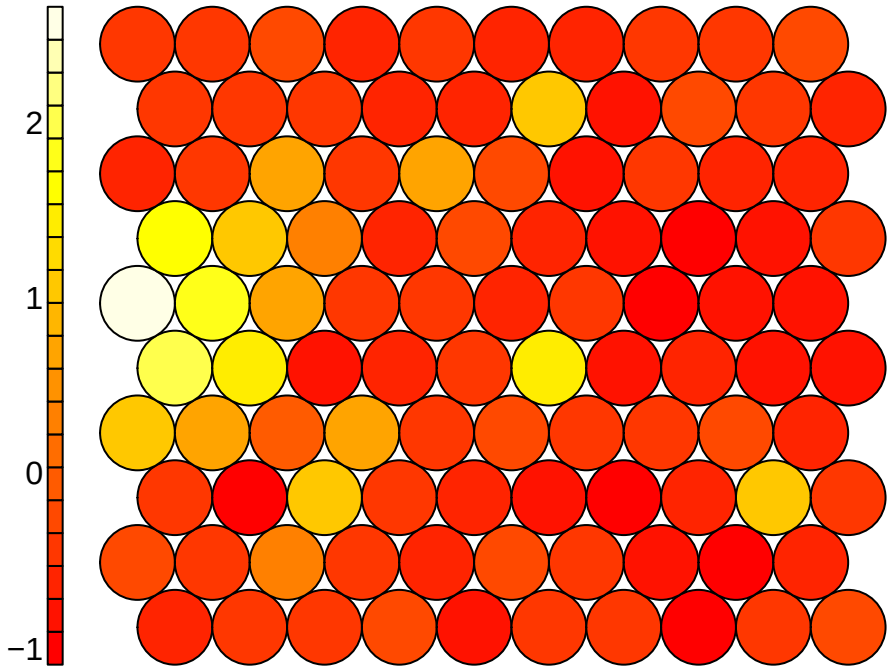
```
som$data <- data.frame(som$data) #changed to dataframe to extract column names easier.

#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)
}
```

a1.leaf

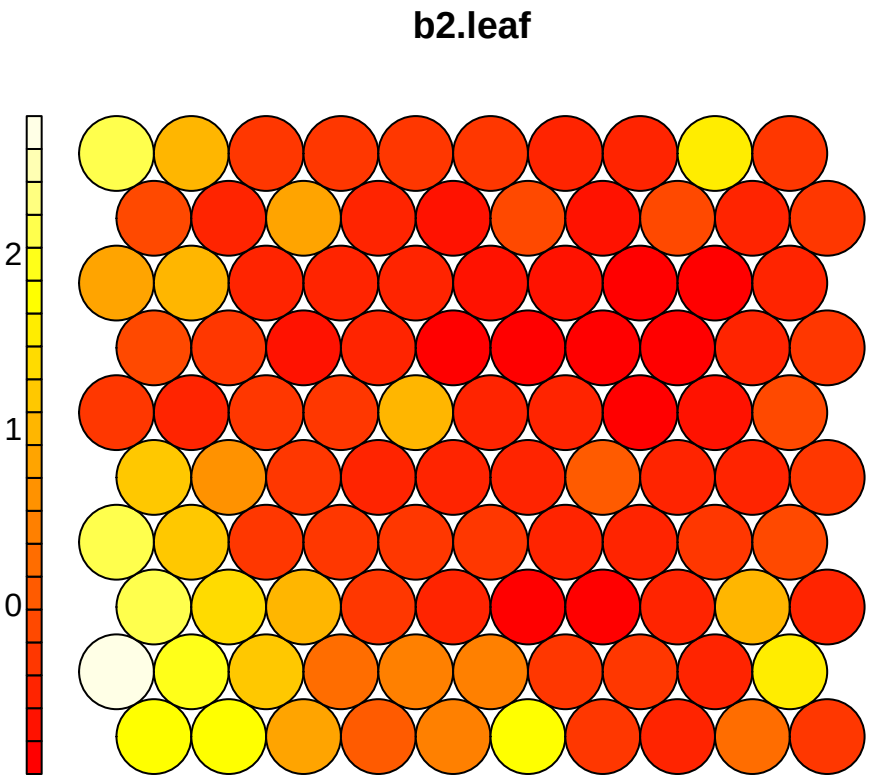



```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

a1.SAM

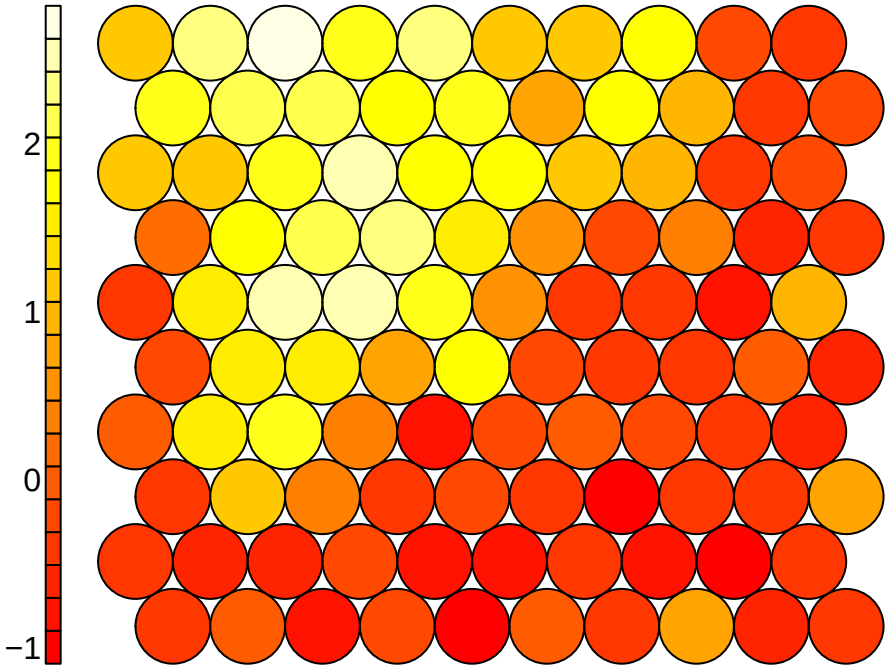


```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



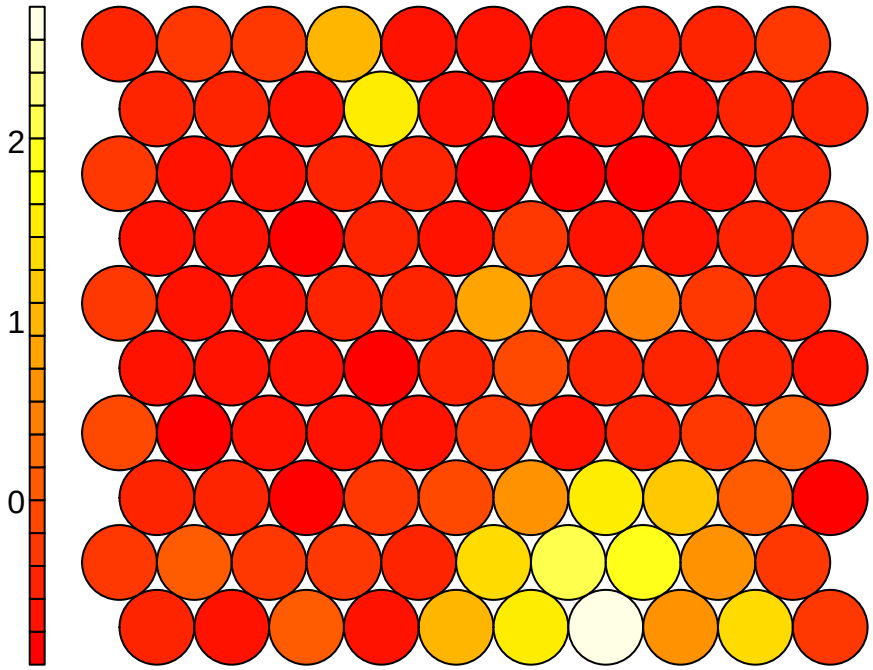

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

b2.SAM

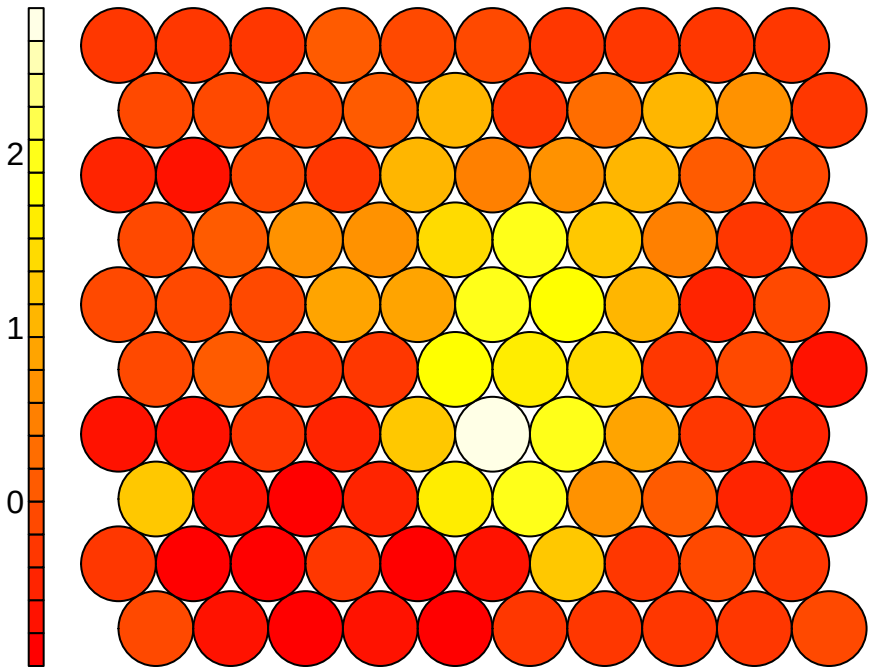



```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

c3.leaf

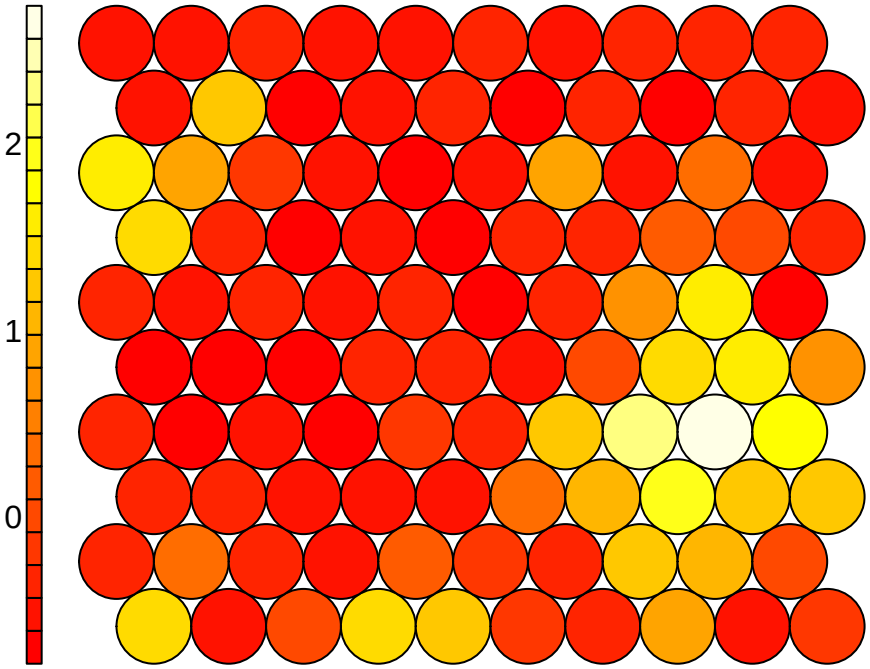


```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```



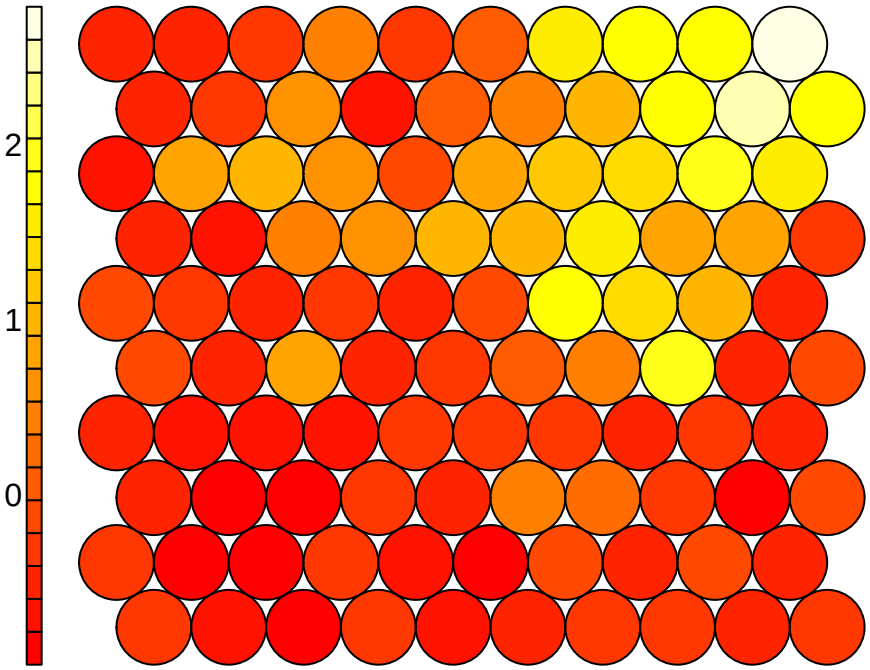

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

d4.leaf

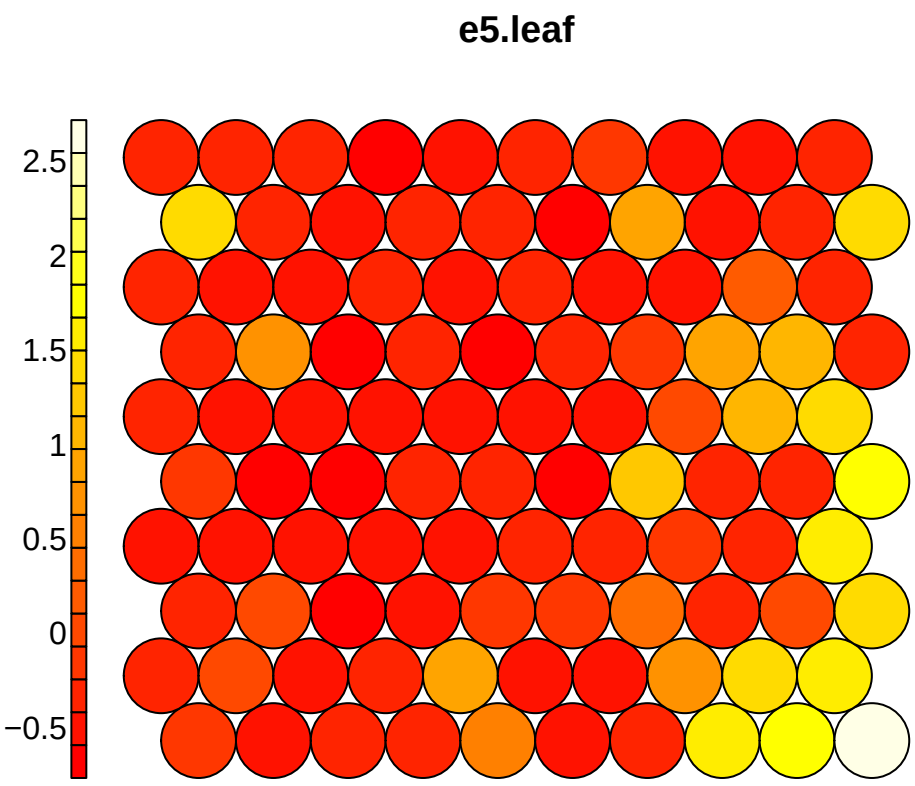



```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fd1a3c3b0d0>
## <environment: namespace:graphics>
```

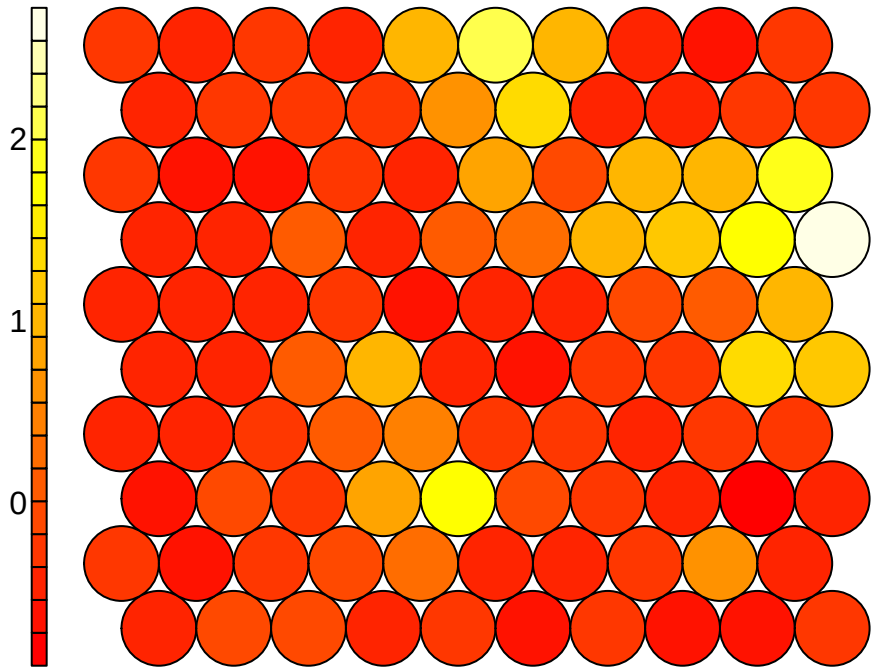
d4.SAM



```
## 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>
```

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
```

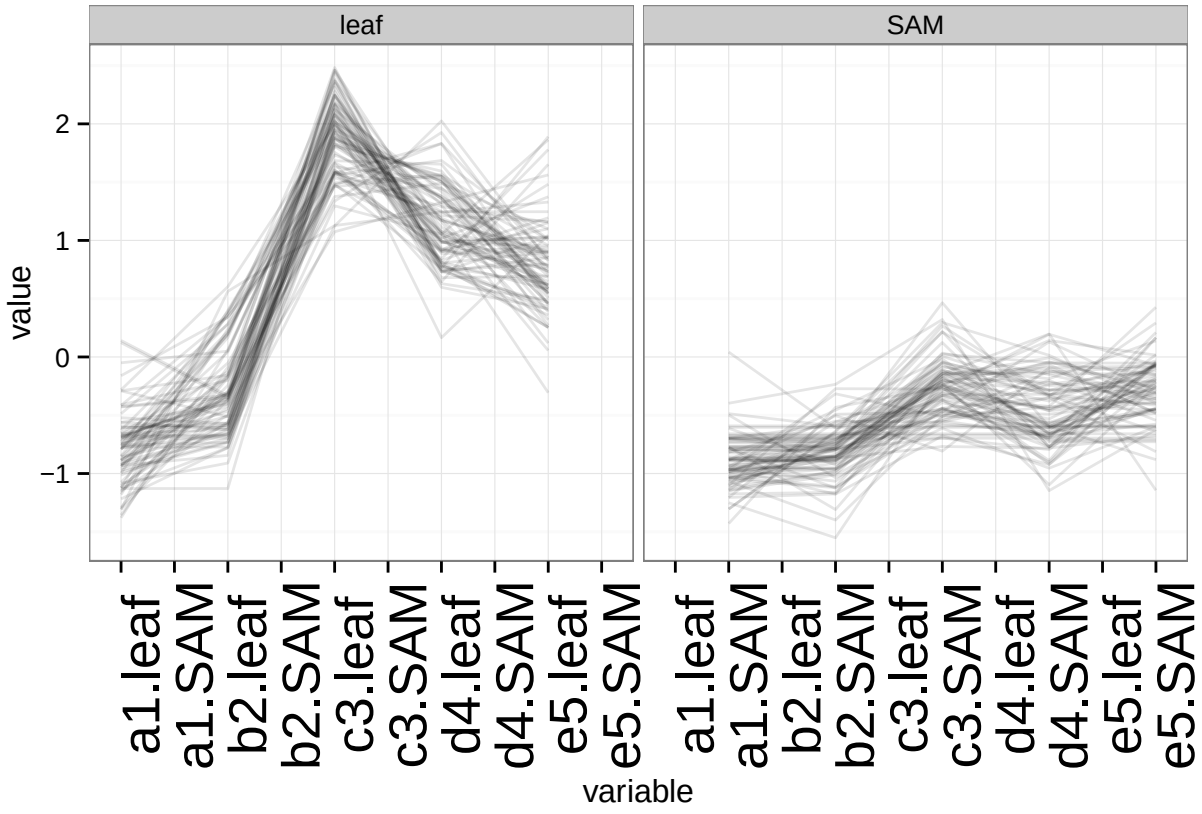
```
##  [1] "gene"             "a1.leaf"          "a1.SAM"
##  [4] "b2.leaf"          "b2.SAM"           "c3.leaf"
##  [7] "c3.SAM"           "d4.leaf"          "d4.SAM"
## [10] "e5.leaf"          "e5.SAM"           "PC1"
## [13] "PC2"              "PC3"              "PC4"
## [16] "PC5"             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

Visualize by cluster

```
# clusterVis_line(1)
# clusterVis_line(2)
```

```
# 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)
# clusterVis_line(12)
# clusterVis_line(13)
# clusterVis_line(14)
# clusterVis_line(15)
# clusterVis_line(16)
# clusterVis_line(17)
clusterVis_line(18) #up in SAM
```

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



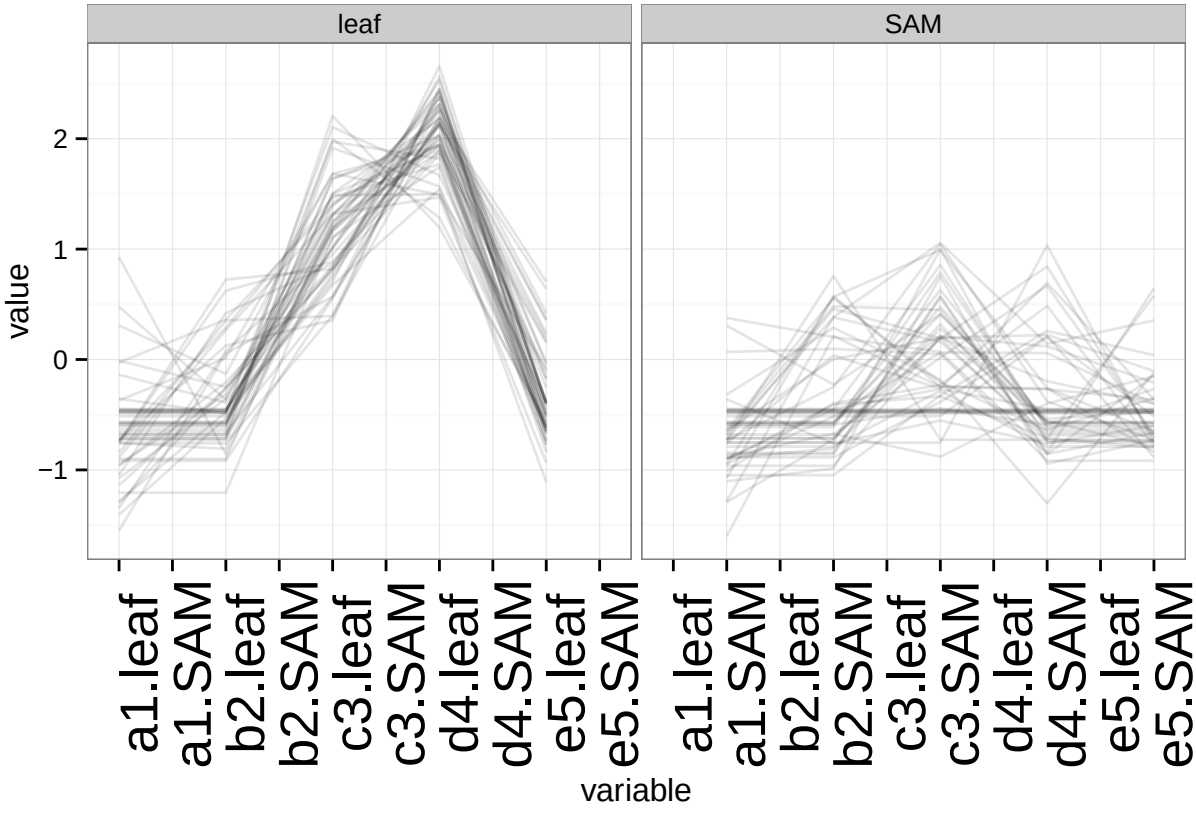

```
y <- genesInClust(18, plot.data, annotation)
```

```
## [1] 78
```

```
write.csv(y, "../clusterTables/analysis4.cluster18.csv")
```

```
# 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) #up in both
```

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



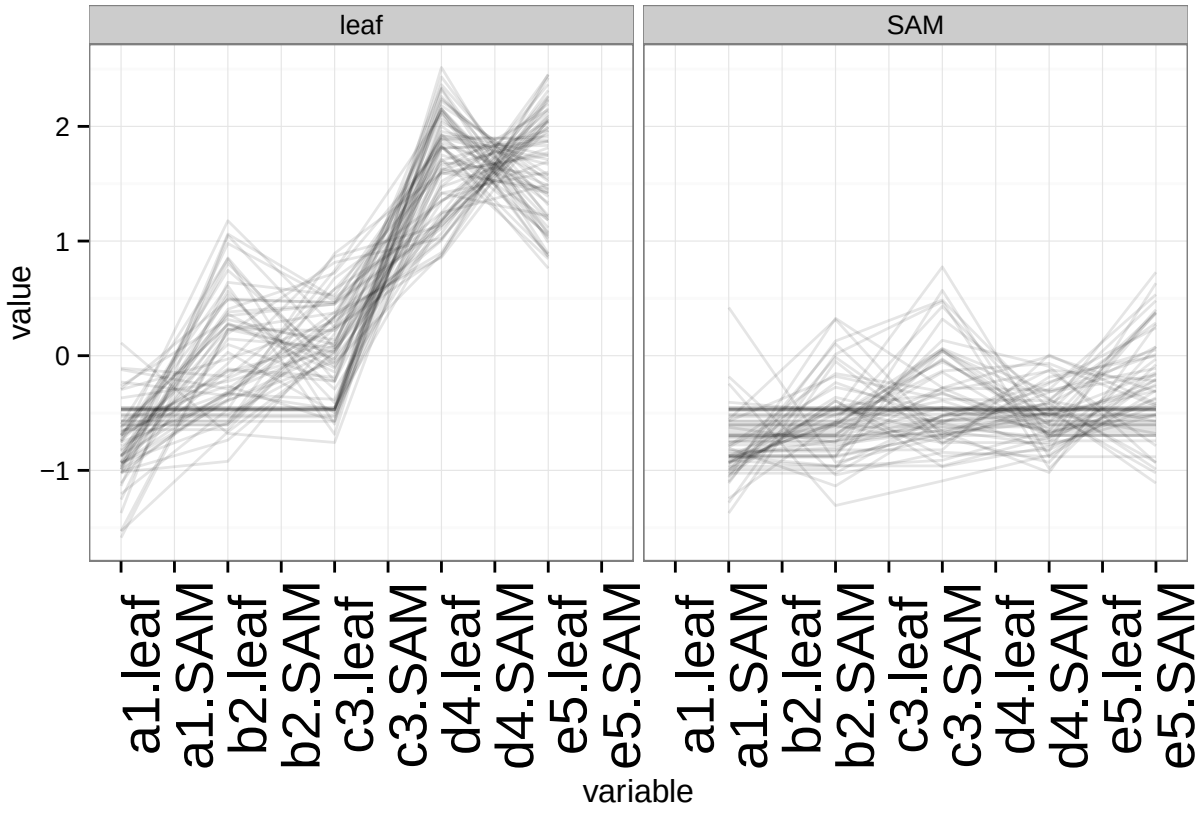

```
y <- genesInClust(28, plot.data, annotation)
```

```
## [1] 51
```

```
write.csv(y, "../clusterTables/analysis4.cluster28.csv")

# clusterVis_line(29)
# clusterVis_line(30)
# clusterVis_line(31)
# clusterVis_line(32)
# clusterVis_line(33)
```

```
# clusterVis_line(34)
# clusterVis_line(35)
# clusterVis_line(36)
# clusterVis_line(37)
# clusterVis_line(38)
# clusterVis_line(39)

clusterVis_line(40) #up in both
```

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

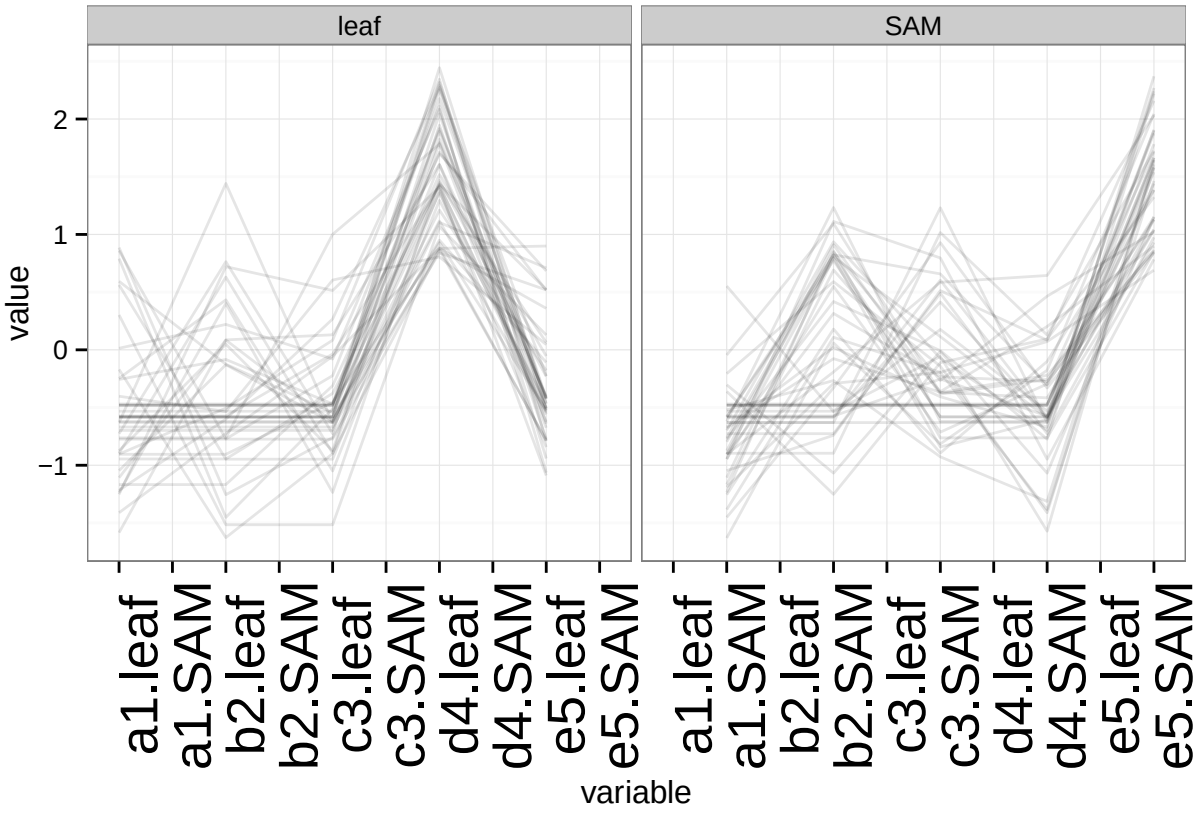



```
y <- genesInClust(40, plot.data, annotation)
```

```
## [1] 70
```

```
write.csv(y, "../clusterTables/analysis4.cluster40.csv")
# clusterVis_line(41)
# clusterVis_line(42)
# clusterVis_line(43)
# clusterVis_line(44)
# clusterVis_line(45)
# clusterVis_line(46)
# clusterVis_line(47)

clusterVis_line(49) #up in leaf
```

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



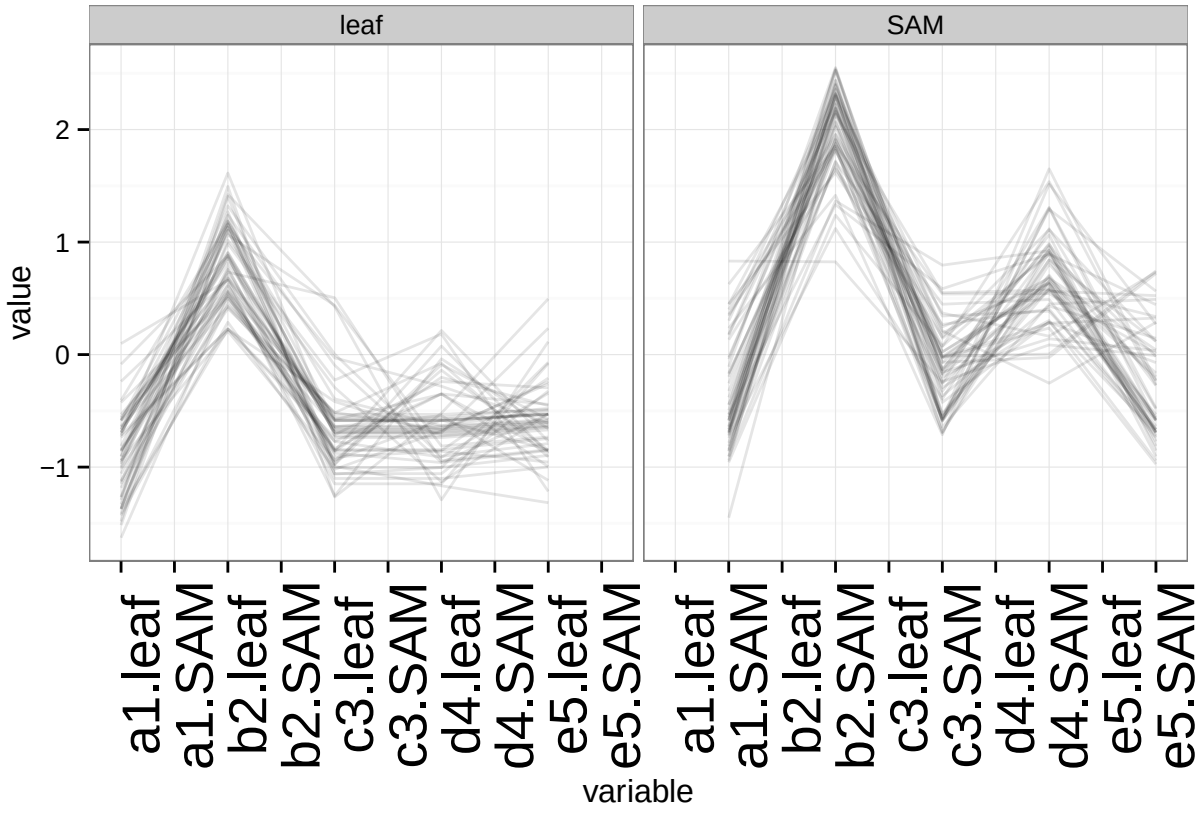

```
y <- genesInClust(49, plot.data, annotation)
```

```
## [1] 42
```

```
write.csv(y, "../clusterTables/analysis4.cluster49.csv")

# clusterVis_line(50)
# clusterVis_line(51)
# clusterVis_line(52)
# clusterVis_line(53)
# clusterVis_line(54)
# clusterVis_line(55)
# clusterVis_line(56)
# clusterVis_line(57)
# clusterVis_line(58)
# clusterVis_line(59)
# clusterVis_line(60)
# clusterVis_line(61)
# clusterVis_line(62)
# clusterVis_line(63)
# clusterVis_line(64)
# clusterVis_line(65)
# clusterVis_line(66)
# clusterVis_line(67)
# clusterVis_line(68)
```

```
# clusterVis_line(69)
# clusterVis_line(70)
# clusterVis_line(71)
# clusterVis_line(72)
# clusterVis_line(73)
# clusterVis_line(74)
# clusterVis_line(75)
# clusterVis_line(76)
# clusterVis_line(77)
# clusterVis_line(78)
# clusterVis_line(79)
# clusterVis_line(80)
# clusterVis_line(81)
# clusterVis_line(82)

clusterVis_line(83) #down in leaf
```

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



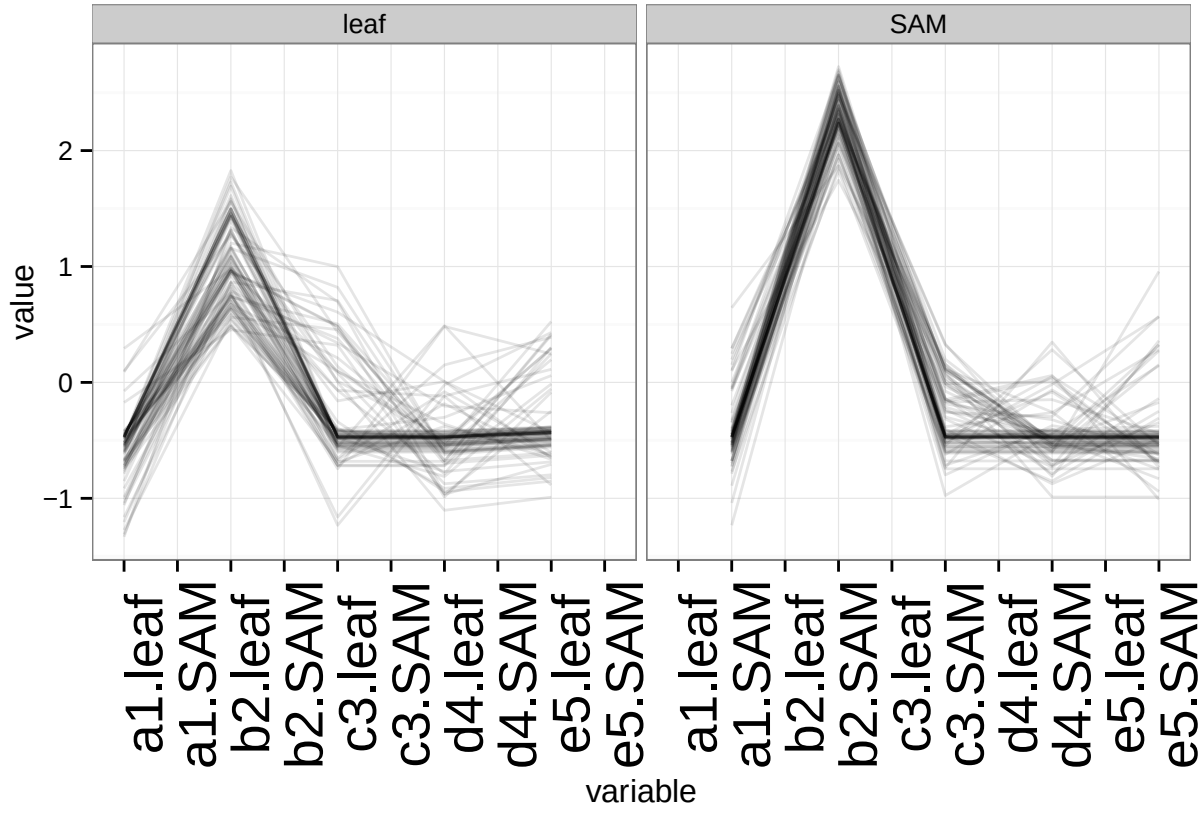

```
y <- genesInClust(83, plot.data, annotation)
```

```
## [1] 52
```

```
write.csv(y, "../clusterTables/analysis4.cluster83.csv")
```

```
# clusterVis_line(84)
# clusterVis_line(85)
# clusterVis_line(86)
# clusterVis_line(87)
# clusterVis_line(88)
# clusterVis_line(89)
# clusterVis_line(90)
# clusterVis_line(91)

clusterVis_line(92)  #down in leaf
```

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




```
y <- genesInClust(92, plot.data, annotation)
```

```
## [1] 85
```

```
write.csv(y, "../clusterTables/analysis4.cluster92.csv")

# clusterVis_line(93)
# clusterVis_line(94)
# clusterVis_line(95)
# clusterVis_line(96)
# clusterVis_line(97)
# clusterVis_line(98)
# clusterVis_line(99)
# clusterVis_line(100)
```

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
```

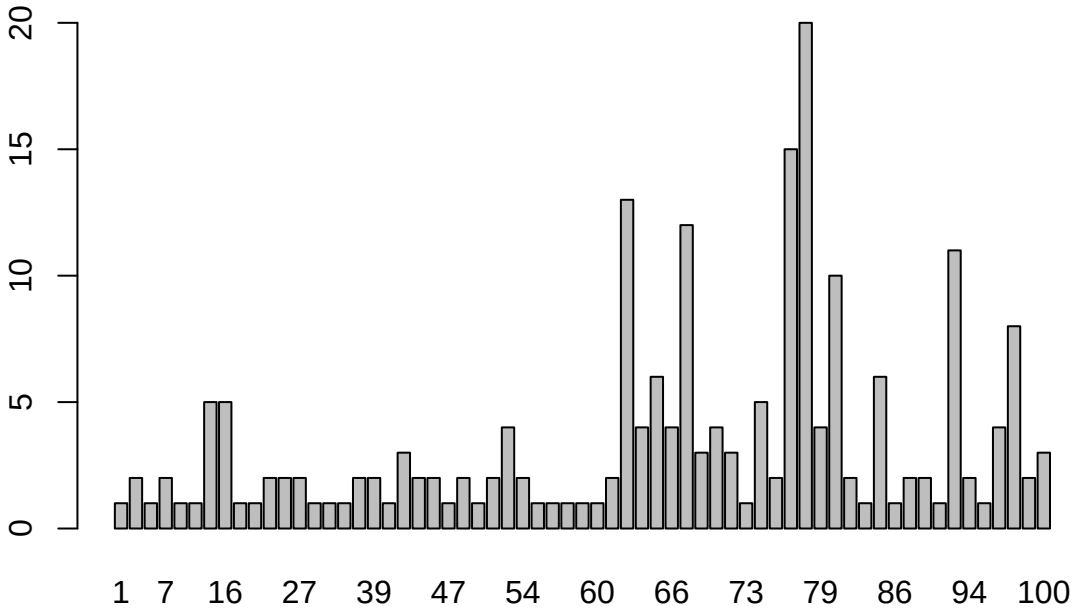
```
#Get only needed columns
v9.clusterIDs <- v9.cluster[,c(1,22,23)]
colnames(v9.clusterIDs)[2]<-"cluster"

#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  7 16 78 11 66 63 26 76 70 64 ...
##  $ som$distances: num  0.02312 0.95917 0.37605 0.00866 0.3109 ...
```

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

summary(v9.clusterIDs)
```

```
##                  ITAG        cluster    som$distances
##  Solyc01g007410.2.1:  1   78     : 20   Min.   :0.002
##  Solyc01g007500.2.1:  1   76     : 15   1st Qu.:0.375
##  Solyc01g010150.2.1:  1   63     : 13   Median :0.684
##  Solyc01g014250.2.1:  1   67     : 12   Mean   :0.922
##  Solyc01g073770.2.1:  1   93     : 11   3rd Qu.:1.199
##  Solyc01g079950.2.1:  1   80     : 10   Max.   :6.037
##  (Other)           :206   (Other):131
```

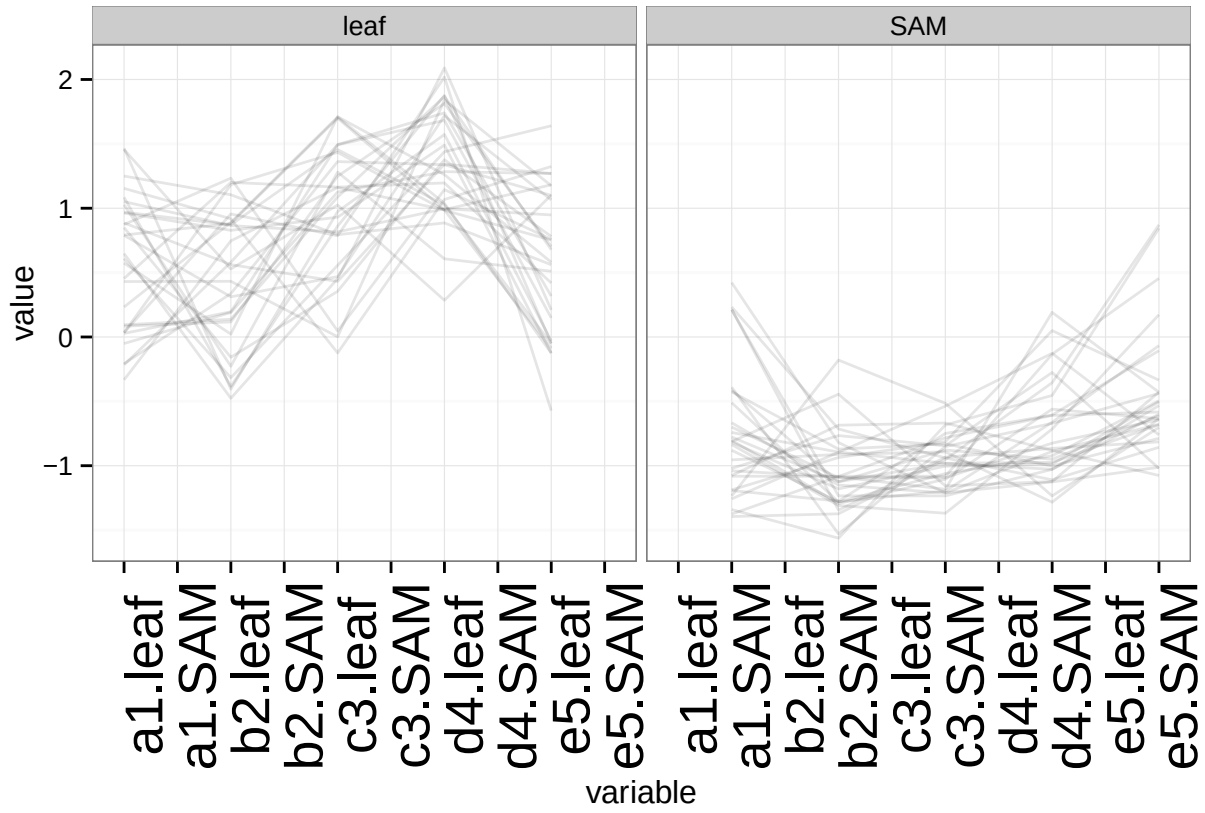
```
plot(v9.clusterIDs$cluster) #possible enriched in cluster #5, 6, and 17? Is there a way to statistically
```



Is this due to cluster size? What genes are in these clusters?

```
clusterVis_line(5)
```

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

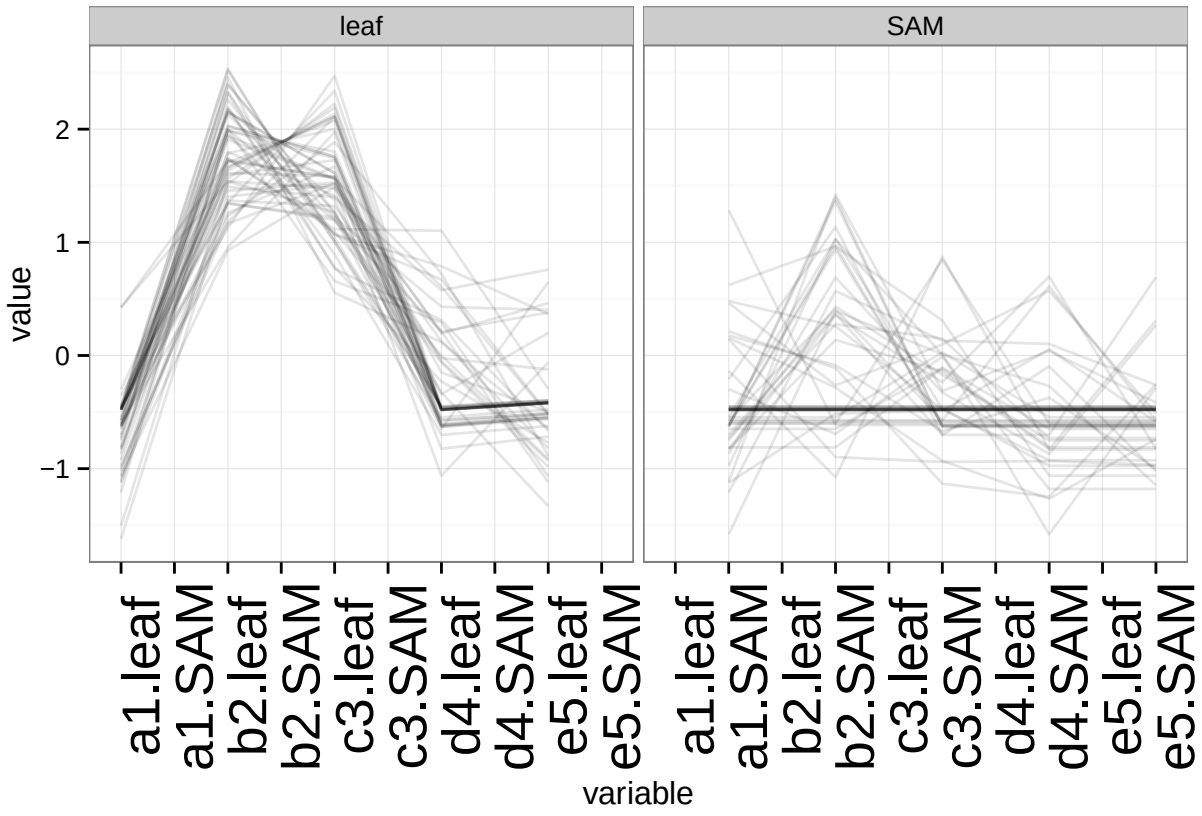


```
y <- genesInClust(5, plot.data, annotation)
```

```
## [1] 29
```

```
write.csv(y, "../clusterTables/analysis4.cluster5.csv")

clusterVis_line(6)
```

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

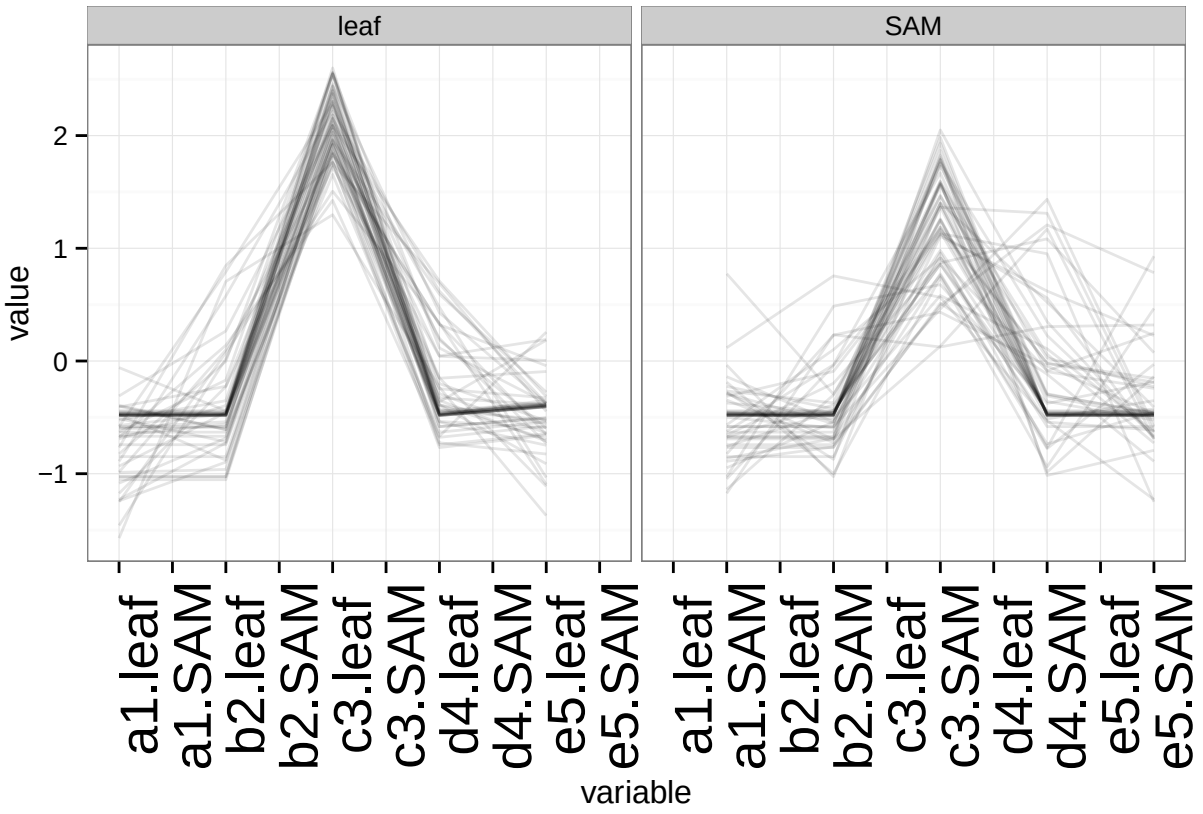



```
y <- genesInClust(6, plot.data, annotation)
```

```
## [1] 53
```

```
write.csv(y, "../clusterTables/analysis4.cluster6.csv")

clusterVis_line(17)
```

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



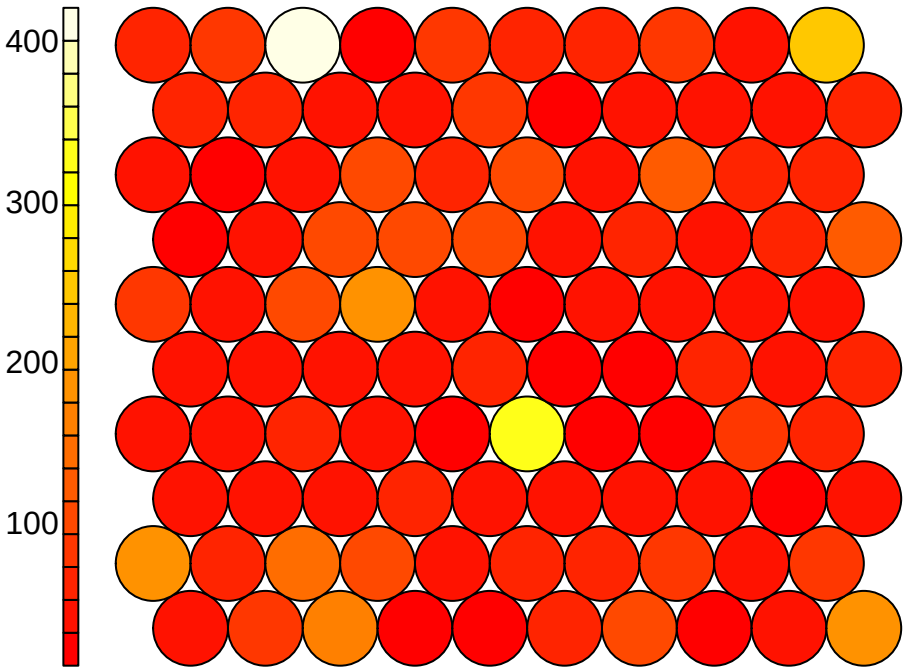

```
y <- genesInClust(17, plot.data, annotation)
```

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

```
write.csv(y, "../clusterTables/analysis4.cluster17.csv")
```

Cluster size does not appear to make a big difference

```
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)
```

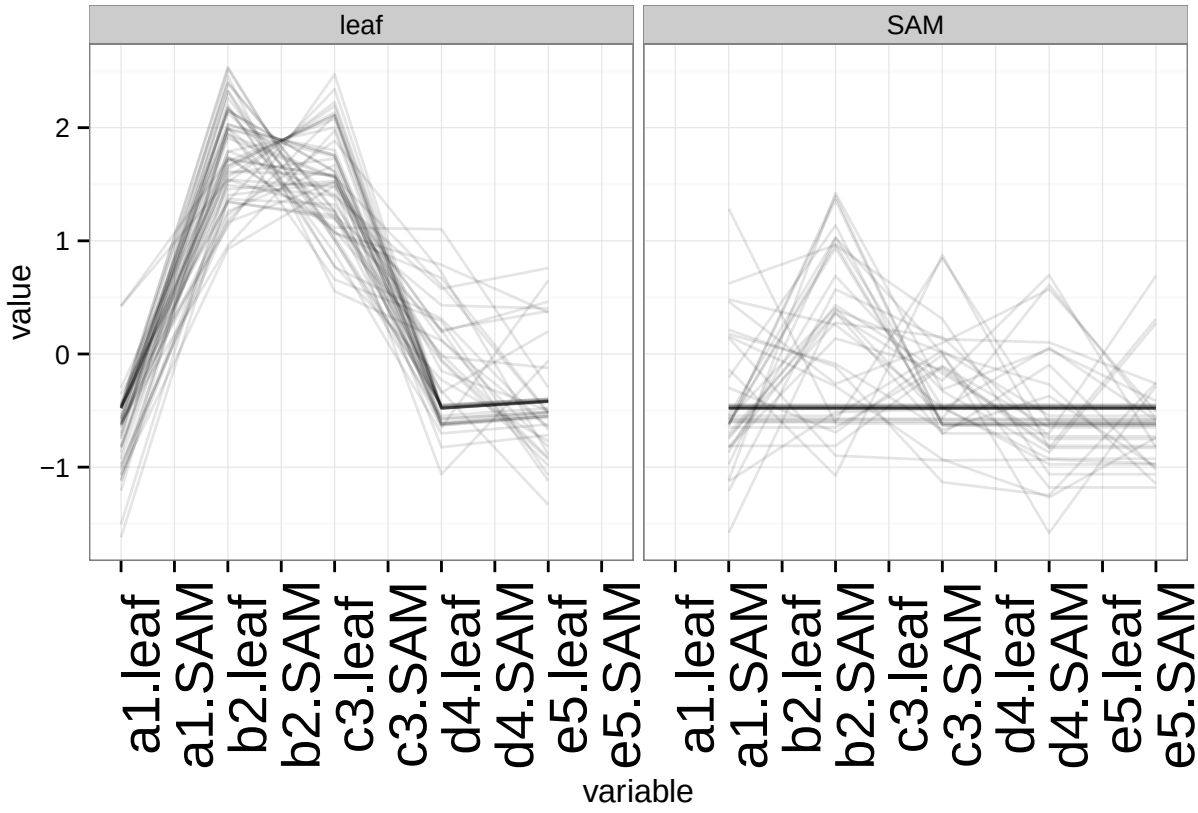
```
##  [1] "ITAG"             "a1.leaf"          "a1.SAM"
##  [4] "b2.leaf"          "b2.SAM"           "c3.leaf"
##  [7] "c3.SAM"           "d4.leaf"          "d4.SAM"
## [10] "e5.leaf"          "e5.SAM"           "PC1"
## [13] "PC2"              "PC3"              "PC4"
## [16] "PC5"             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

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

knot.clusterIDs #clusters 6 and 5
```

```
##                 ITAG som$unit.classif som$distances
## 1 Solyc01g100510.2.1               78        0.5804
## 2 Solyc02g081120.2.1               76        1.2069
## 3 Solyc04g077210.2.1               63        0.4949
## 4 Solyc05g005090.2.1               65        0.4362
## 5 Solyc11g069890.1.1               76        0.3977
```

```
clusterVis_line(6)
```

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

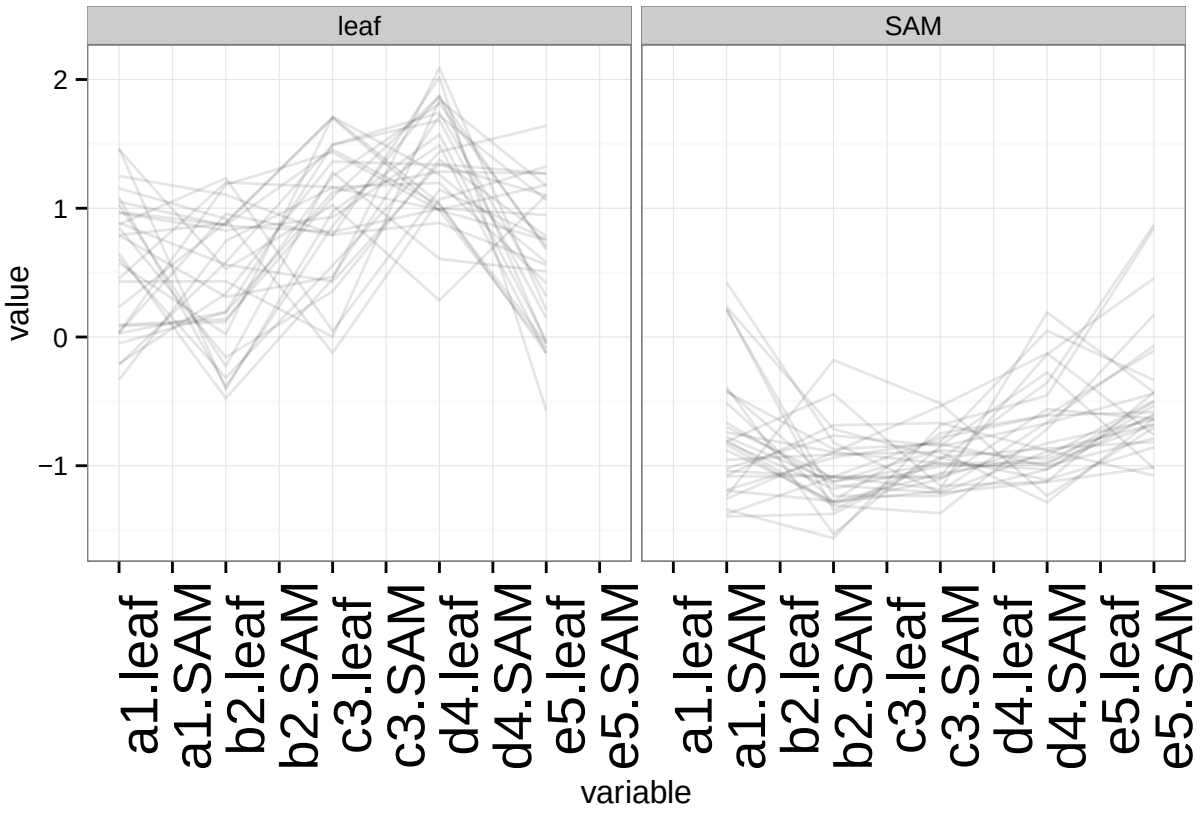



```
y <- genesInClust(6, plot.data, annotation)
```

```
## [1] 53
```

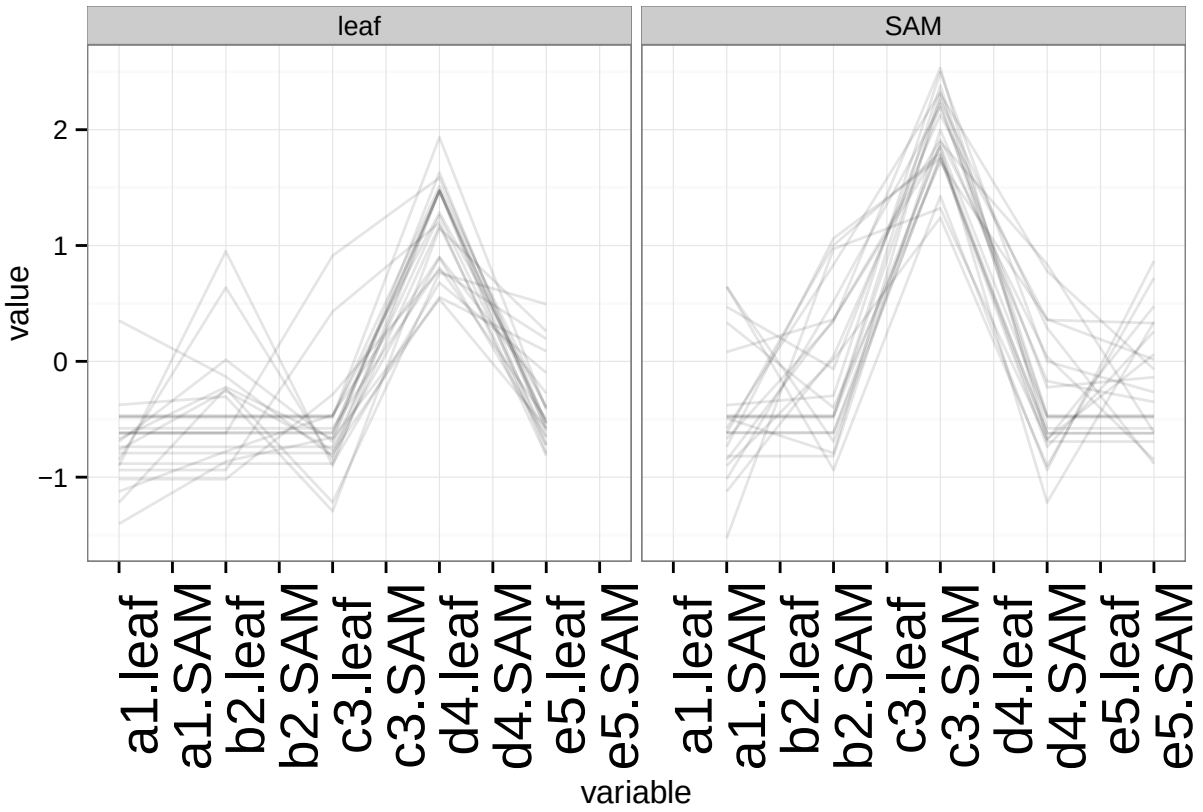
```
write.csv(y, "../clusterTables/analysis4.cluster6.csv")
clusterVis_line(5)
```

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




```
clusterVis_line(37)
```

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



Overall Results and Future Analysis:

There are several clusters that could be looked at more closely. These are in the `clusterTables` directory. The larger the SOM the tighter the expression patterns in the cluster.