Self Organizing Maps (SOM): Example using RNAseq reads

Part 2b: Super Self Organizing Maps

Super self organizing maps allows you to add another level to your analysis. For example in this project, I ran my RNAseq analysis on two different genotypes, *tf-2* and wild type. Instead of clustering each gene one cluster, it forces the same gene from each genotype into the same cluster. Therefore, each gene is represented twice in this superSOM analysis (one for each genotype).

Required Libraries

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(reshape)
library(kohonen)
```

```
## Loading required package: class

##
## Attaching package: 'class'

## The following object is masked from 'package:reshape':
##
##     condense

## Loading required package: MASS
```

```
library(RColorBrewer)
```

Read in data and clean up.

```
mostDEgenes <- read.csv("../data/allGeneListBothGenotypes.csv")

#keep only needed columns (gene, genotype, type, mean)
mostDEgenes <- mostDEgenes[c(7, 2, 1, 4)]

#Change from long to wide data format
mostDEgene.long <- cast(mostDEgenes, genotype + gene ~ type, value.var = mean, fun.aggregate = "mean")
```

```
## Using mean as value column.  Use the value argument to cast to override this choice
```

```
# to keep attributes associated
mostDEgene.long <- as.data.frame(mostDEgene.long)
```

PCA

```r
mostDEgene.long <- as.data.frame(mostDEgene.long)
names(mostDEgene.long)
```

```
## [1] "genotype" "gene"     "Ambr"     "Aother"   "Bmbr"     "Bother"
## [7] "Cmbr"     "Cother"
```

```r
scale_data <- as.matrix(t(scale(t(mostDEgene.long[c(3:8)]))))
head(scale_data)
```

```
##         Ambr     Aother       Bmbr     Bother       Cmbr      Cother
## 1  0.6682306 -1.5249843 -0.8142005  1.0677854  0.6500786 -0.04690986
## 2 -0.3039380 -0.3362605  2.0337657 -0.4956094 -0.3798599 -0.51809787
## 3  0.4553713 -0.8054291  1.6148298 -0.5755183  0.3508907 -1.04014435
## 4 -0.5191322  1.6854463 -0.9561593 -0.1061458 -0.7444119  0.64040284
## 5 -0.1488347 -0.5336037  2.0228620 -0.4697011 -0.4084674 -0.46225509
## 6 -0.4345752 -0.4641784  2.0365771 -0.4264810 -0.2745967 -0.43674578
```

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

summary(pca)
```

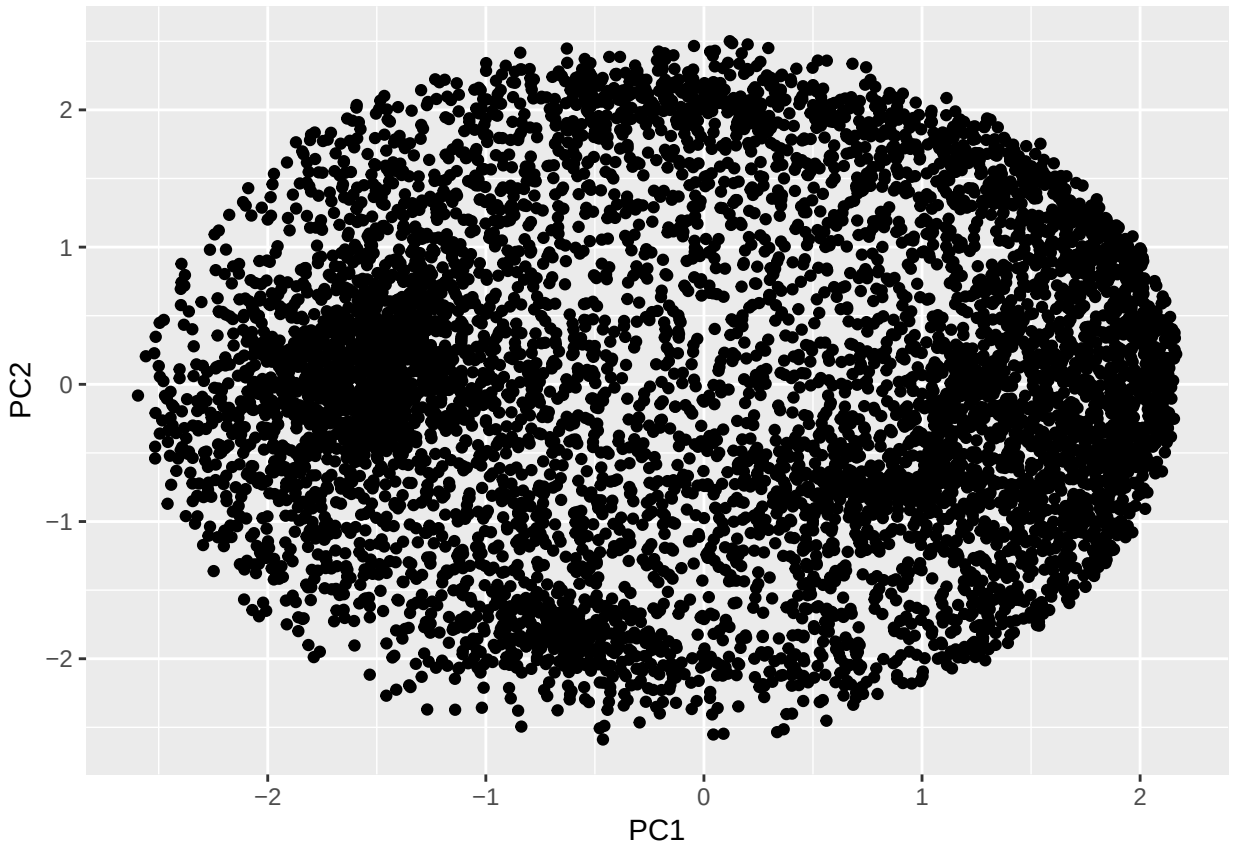
```
## Importance of components:
##                           PC1    PC2   PC3    PC4    PC5       PC6
## Standard deviation     1.3056 1.1248 1.036 1.0207 0.9563 2.895e-15
## Proportion of Variance 0.2841 0.2109 0.179 0.1736 0.1524 0.000e+00
## Cumulative Proportion  0.2841 0.4949 0.674 0.8476 1.0000 1.000e+00
```

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

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

Visualizing the PCA

Looks to be three major clusters.

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



SuperSOM

First we need to get the data scaled and in the correct format. In this example each genoype was scaled seperatly then brought back together in a list. Each gene will be represented twice.

```
tf2 <- as.matrix(subset(mostDEgene.long, genotype == "tf2", select = 3:8))
wt <- as.matrix(subset(mostDEgene.long, genotype == "wt", select = 3:8))

wt <- as.matrix(wt)
tf2 <- as.matrix(tf2)

sc.wt <- t(scale(t(wt)))
sc.tf2 <- t(scale(t(tf2)))

all.data <- list(sc.wt,sc.tf2)
str(all.data)
```

```
## List of 2
##  $ : num [1:3580, 1:6] -1.263 -0.186 0.501 -0.238 -0.737 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3580] "3581" "3582" "3583" "3584" ...
##   .. ..$ : chr [1:6] "Ambr" "Aother" "Bmbr" "Bother" ...
##   ..- attr(*, "scaled:center")= Named num [1:3580] 7.37 24.66 12.61 26.43 3.49 ...
##   .. ..- attr(*, "names")= chr [1:3580] "3581" "3582" "3583" "3584" ...
##   ..- attr(*, "scaled:scale")= Named num [1:3580] 2.85 40.21 6.81 28.1 4.55 ...
```

```
##   .. ..- attr(*, "names")= chr [1:3580] "3581" "3582" "3583" "3584" ...
##  $ : num [1:3580, 1:6] 0.668 -0.304 0.455 -0.519 -0.149 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:3580] "1" "2" "3" "4" ...
##   .. ..$ : chr [1:6] "Ambr" "Aother" "Bmbr" "Bother" ...
##   ..- attr(*, "scaled:center")= Named num [1:3580] 7.02 34.72 10.35 20.8 8.27 ...
##   .. ..- attr(*, "names")= chr [1:3580] "1" "2" "3" "4" ...
##   ..- attr(*, "scaled:scale")= Named num [1:3580] 3.75 61.02 3.18 13.91 14.51 ...
##   .. ..- attr(*, "names")= chr [1:3580] "1" "2" "3" "4" ...
```

Super SOM

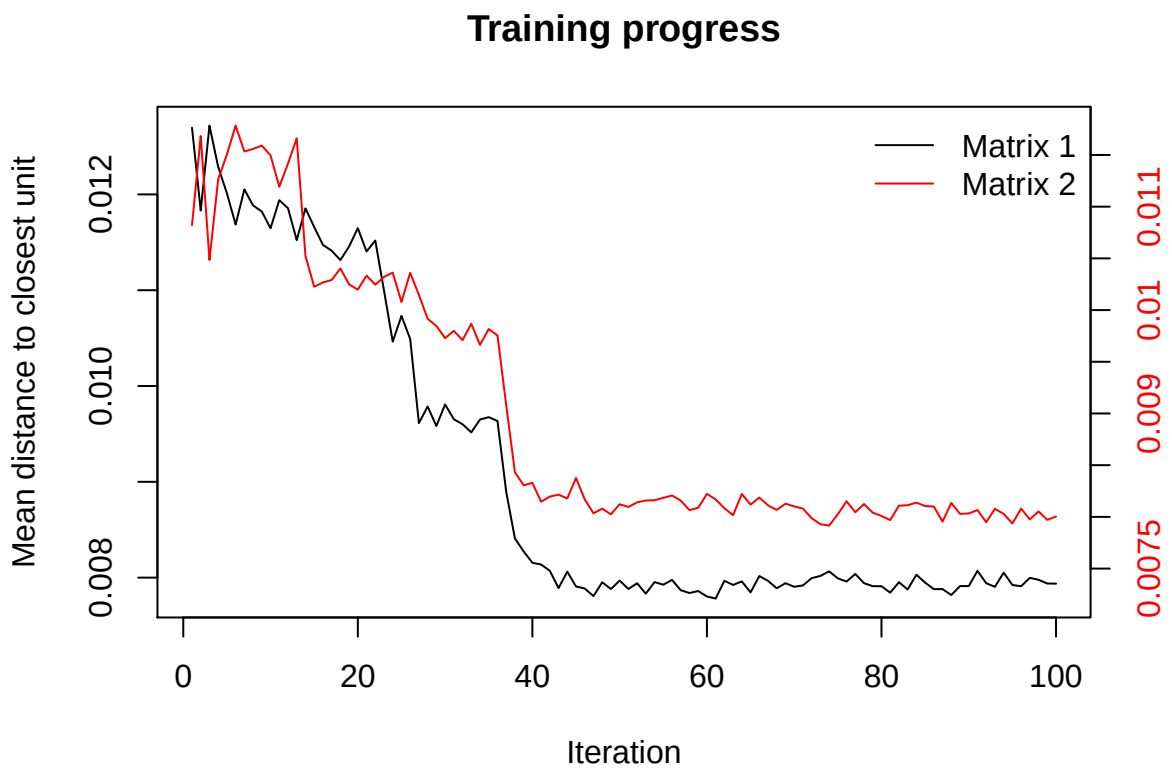
```
set.seed(2)
ssom <- supersom(all.data, somgrid(6, 6, "hexagonal"),weights=c(0.5,0.5))

summary(ssom)
```

```
## supersom map of size 6x6 with a hexagonal topology.
## Training data included of  3580 objects
## The number of layers is 2
## Mean distance to the closest unit in the map: 0.03614708
```

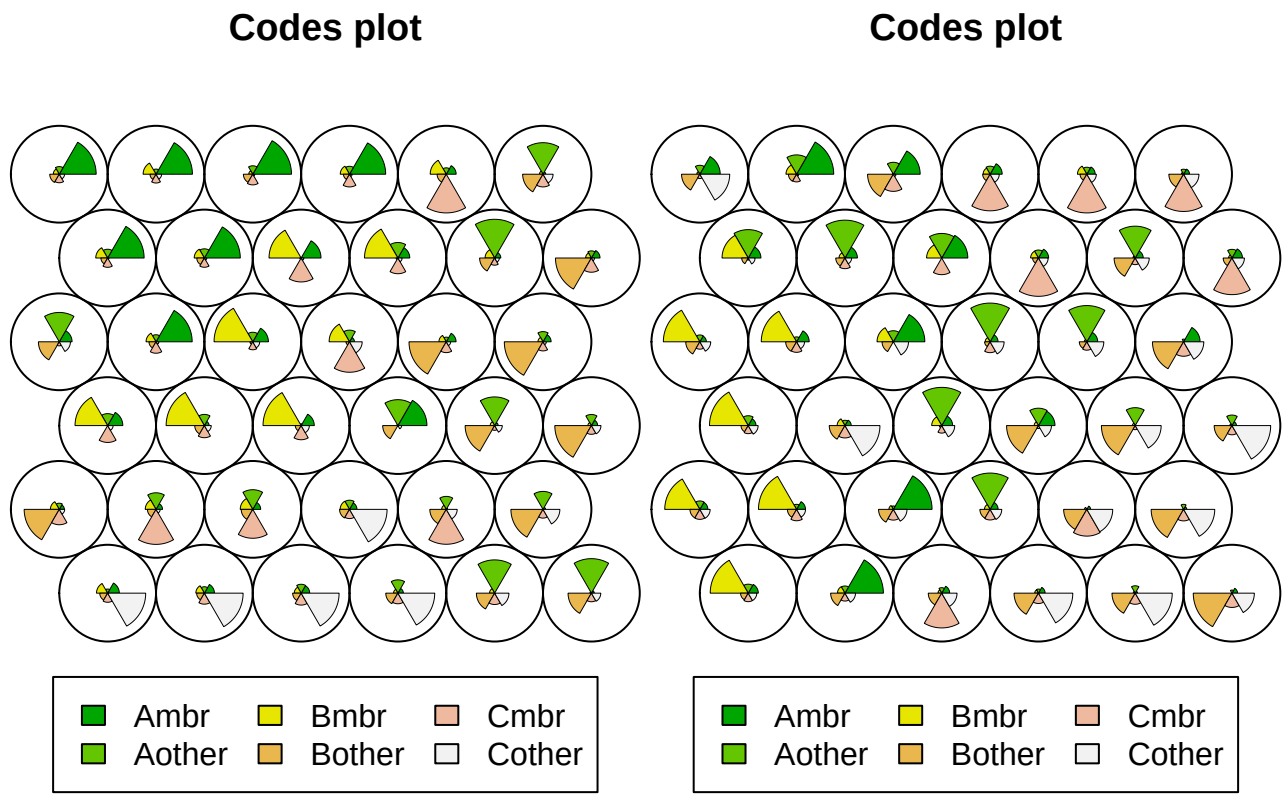
Notice the number of layers are now two. You could have more genotypes or different types of layers, like treatment. All depends on the experiment.

Let's look at the super SOM results. Now when you run the plots, you will get two plots.

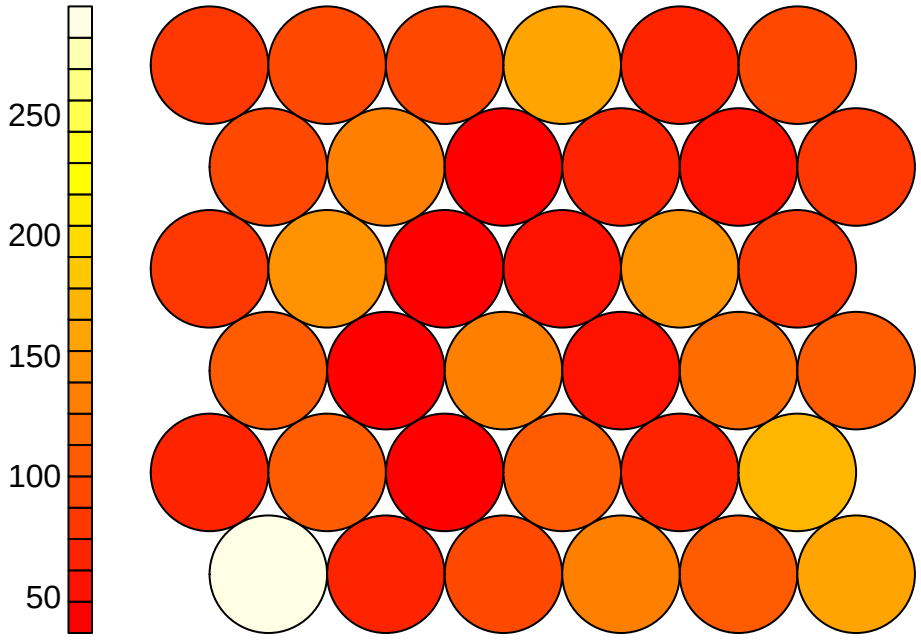
```
plot(ssom, type ="changes")
```



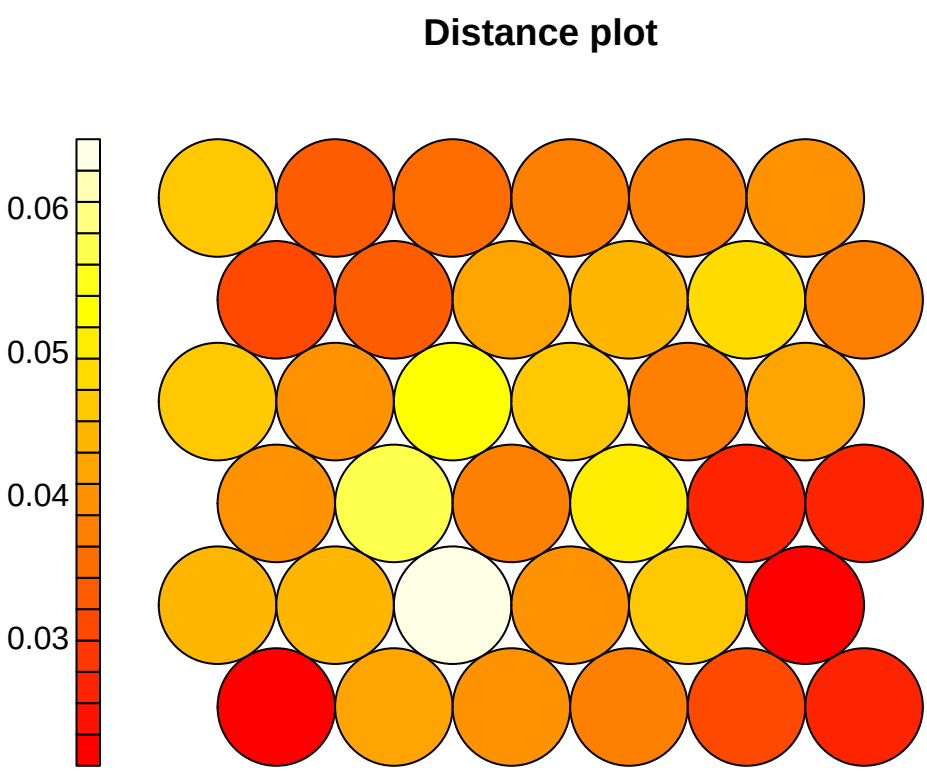
```
plot(ssom, type = "codes")
```




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



```
plot(ssom, type = "quality")
```




```
data.val <- cbind(data.val,ssom$unit.classif,ssom$distances)

head(data.val)
```

```
##   genotype               gene      Ambr     Aother       Bmbr    Bother
## 1      tf2 Solyc00g005050.2.1  9.525735  1.2969785   3.963779 11.024831
## 2      tf2 Solyc00g005070.1.1 16.174774 14.2025962 158.811326  4.479819
## 3      tf2 Solyc00g005080.1.1 11.796048  7.7875692  15.482330  8.518527
## 4      tf2 Solyc00g005840.2.1 13.584917 44.2405788   7.507857 19.327679
## 5      tf2 Solyc00g005870.1.1  6.110472  0.5291395  37.612382  1.456090
## 6      tf2 Solyc00g005880.1.1  1.839943  1.1235811  61.638800  2.035812
##        Cmbr    Cother       Ambr     Aother       Bmbr     Bother
## 1  9.457630  6.842589  0.6682306 -1.5249843 -0.8142005  1.0677854
## 2 11.542347  3.107667 -0.3039380 -0.3362605  2.0337657 -0.4956094
## 3 11.463872  7.041336  0.4553713 -0.8054291  1.6148298 -0.5755183
## 4 10.452300 29.708776 -0.5191322  1.6854463 -0.9561593 -0.1061458
## 5  2.344326  1.564099 -0.1488347 -0.5336037  2.0228620 -0.4697011
## 6  5.711226  1.787418 -0.4345752 -0.4641784  2.0365771 -0.4264810
##         Cmbr       Cother        PC1         PC2         PC3         PC4
## 1  0.6500786 -0.04690986  0.2265119 -1.62836740  0.3866619  1.5784047
## 2 -0.3798599 -0.51809787 -1.6467167 -0.09637224 -0.8791275 -1.3966283
## 3  0.3508907 -1.04014435 -2.2767329 -0.74387704 -0.1000315 -0.4976597
## 4 -0.7444119  0.64040284  1.0796206  1.71914907  0.2204601 -0.2686421
## 5 -0.4084674 -0.46225509 -1.6906182 -0.23661604 -1.0234687 -1.2019845
## 6 -0.2745967 -0.43674578 -1.5294584 -0.32073888 -0.8679183 -1.4646347
##          PC5           PC6 ssom$unit.classif ssom$distances
```

```
## 1  1.0599396  1.484923e-15               11    0.072268688
## 2  0.8451023 -1.665335e-15                1    0.001603610
## 3  0.9484308 -2.886580e-15                1    0.036801305
## 4 -1.0926003  1.110223e-16               23    0.009979580
## 5  0.8959657 -1.498801e-15                1    0.007301491
## 6  0.8818916 -1.332268e-15                1    0.010737197
```

```r
# Output for visualization
 write.table(data.val, file="../data/ssom.data.txt")
```