Getting the most sig. differentiated genes between tissue

Purpose

This is using the basic some with both genotypes. There are two identical ITAGs represented for each tissue type, one for each genotype. Because this is a basic SOM, the genes are free to be in any cluster, as opposed to in superSOMs where the same ITAG are forced to the same cluster.

Required Libraries

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

Self Organizing Maps

1.pca.R

First read in file that came from mostSigDEgenes.Rmd. This is a list of genes from all DE analysis in WT and *tf2*. They were all cancatenated, then duplicate genes were removed. In addition the mean was calculated from the replicates of each type.

The first step is to get it into the right format. First column being the genes, while the subsequent columns are the different libraries (type).

```
mostDEgenes <- read.csv("../data/allGeneListBothGenotypes_analysis5b.csv")
head(mostDEgenes)
```

```
##     type genotype N    mean      sd      se               gene
## 1   Ambr      tf2 4  75.159 144.465  72.233 Solyc00g014800.1.1
## 2   Ambr       wt 3   8.643   8.246   4.761 Solyc00g014800.1.1
## 3 Aother      tf2 4  15.792  12.929   6.465 Solyc00g014800.1.1
## 4 Aother       wt 5   3.723   2.930   1.310 Solyc00g014800.1.1
## 5   Bmbr      tf2 3 124.304 215.300 124.304 Solyc00g014800.1.1
## 6   Bmbr       wt 4  57.467 114.934  57.467 Solyc00g014800.1.1
```

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

#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
```

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

At this point I am going to subset on genotype and scale *seperately* before adding them back together.

```
wt <- subset(mostDEgene.long, genotype == "wt")
tf2 <- subset(mostDEgene.long, genotype == "tf2")

scale_data.wt <- as.matrix(t(scale(t(wt[c(3:8)]))))#transformation.
scale_data.tf2 <- as.matrix(t(scale(t(tf2[c(3:8)]))))#transformation.
scale_data <- rbind(scale_data.wt, scale_data.tf2)
```

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

summary(pca)
```

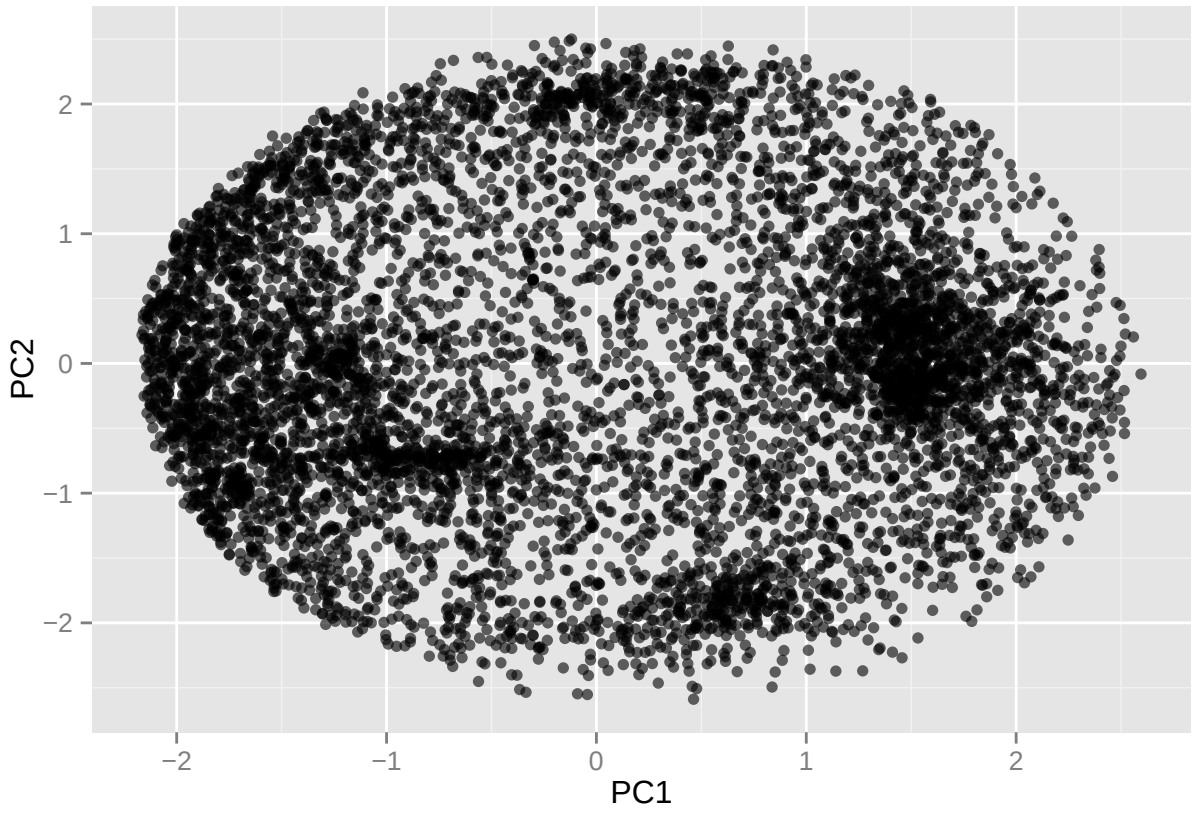
```
## Importance of components:
##                           PC1   PC2   PC3   PC4   PC5      PC6
## Standard deviation     1.306 1.125 1.036 1.021 0.956 1.84e-15
## Proportion of Variance 0.284 0.211 0.179 0.174 0.152 0.00e+00
## Cumulative Proportion  0.284 0.495 0.674 0.848 1.000 1.00e+00
```

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

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

Visualizing the PCA

Looks to be three major clusters.

```
p <- ggplot(data.val, aes(PC1, PC2))
p + geom_point(alpha = .6)
```



I am skipping the large Map and going straight to the small

2. Self Organizing Map- Small (3,2)

The size of the map is something that may cause differences in the genes that are clustered. Using a small map size (3,2), I found they cluster in according to tissue type. See below.

```
som.data <- as.matrix(data.val[,c(9:14)])
head(som.data)
```

```
##       Ambr  Aother      Bmbr    Bother     Cmbr  Cother
## 1 -1.2632 -0.3389 -0.86967  0.39833  0.7013  1.3721
## 2 -0.1857 -0.5008 -0.29882 -0.52721 -0.5093  2.0219
## 3  0.5010 -0.3641 -0.74069 -0.60368 -0.6158  1.8232
## 4 -0.2381 -0.4399 -0.53216  2.03154 -0.3781 -0.4433
## 5 -0.7372 -0.6256 -0.07972 -0.15567 -0.3751  1.9733
## 6 -0.8319 -0.4821  0.03808 -0.06021 -0.5925  1.9286
```

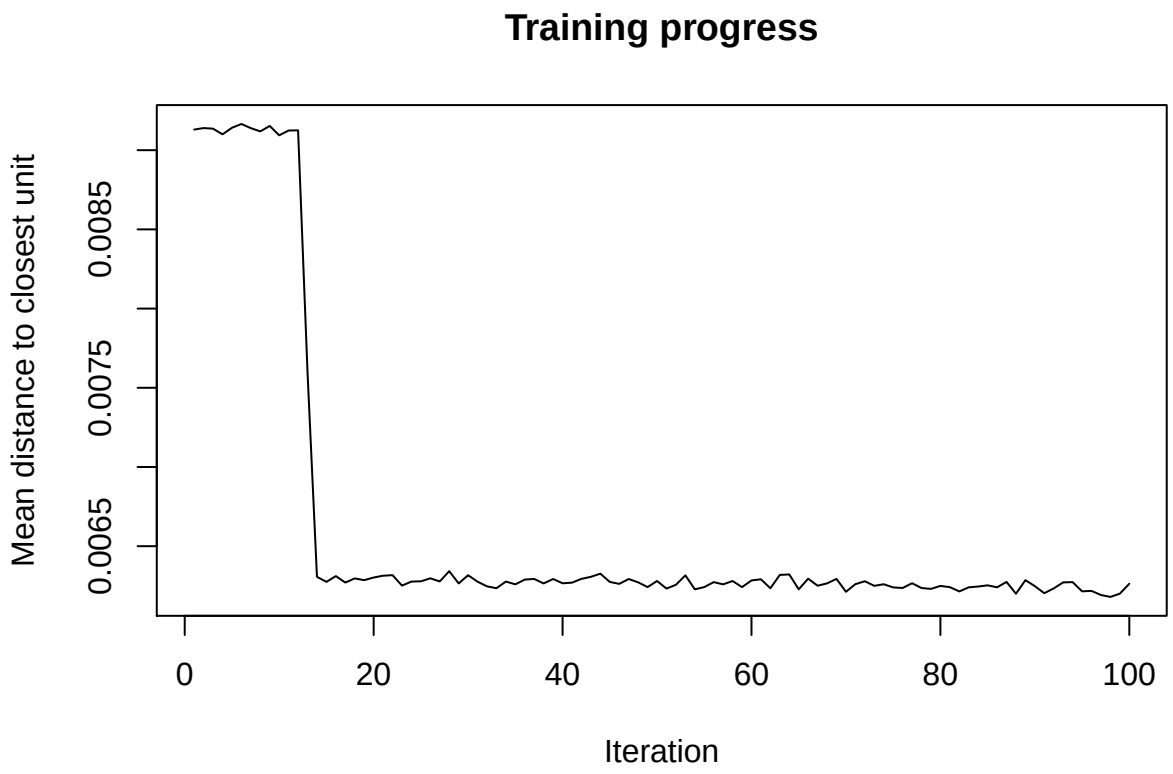
```
set.seed(5)

som <- som(data=som.data, somgrid(3,2,"hexagonal")) #set SOM size
summary(som)
```

```
## som map of size 3x2 with a hexagonal topology.
## Training data included; dimension is 7160 by 6
## Mean distance to the closest unit in the map: 1.67
```

Training Plot ("changes") - Small

This shows a hundred iterations.

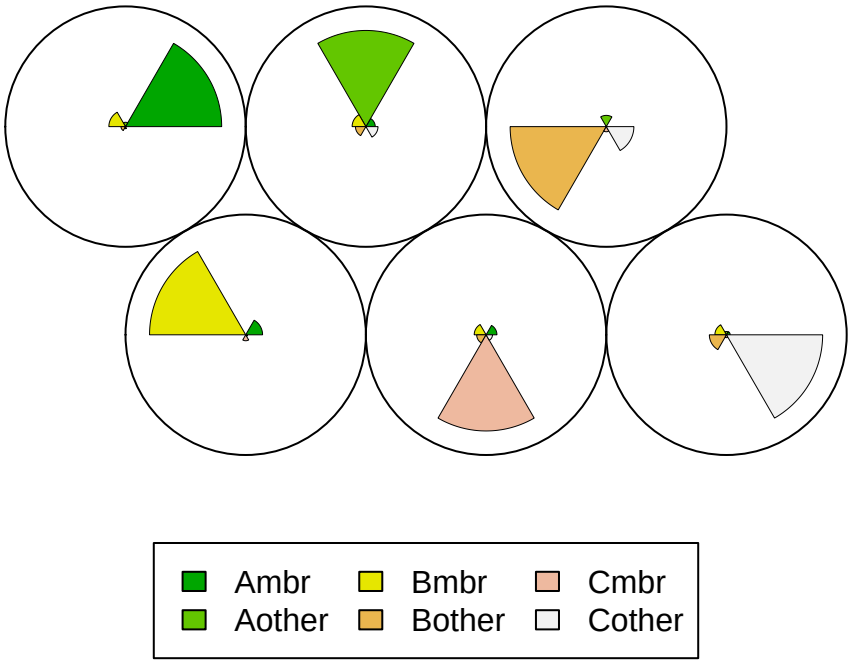
```
plot(som, type ="changes")
```



Code Plot - Small

Here with the small map, each tissue has a tissue specific cluster.

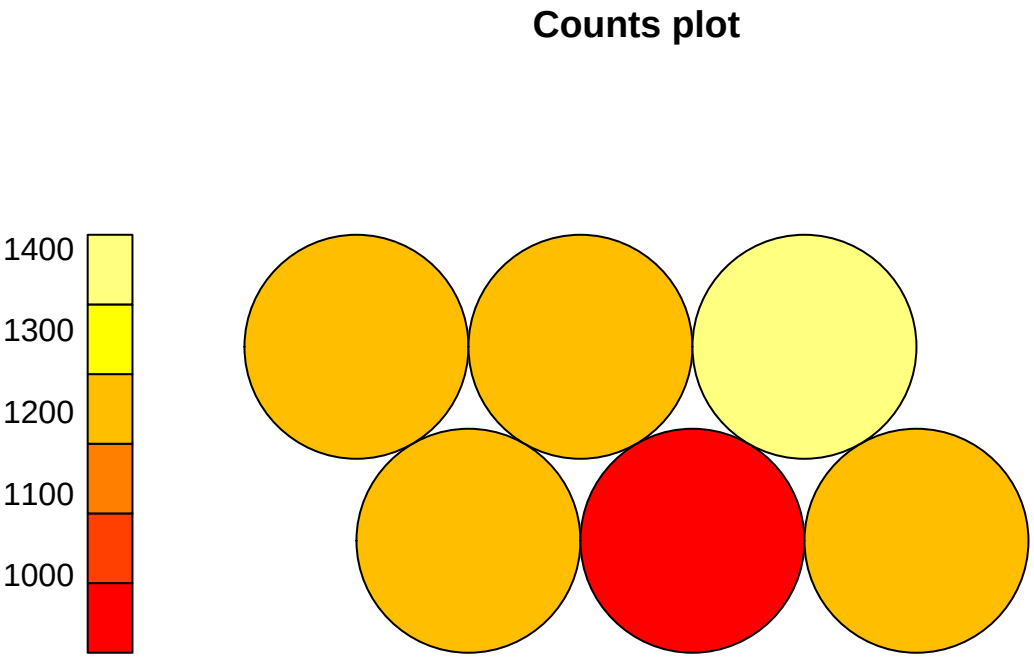
```
plot(som, type = "codes")
```



Count Plot - Small

This tells you how many genes are in each of the clusters.

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

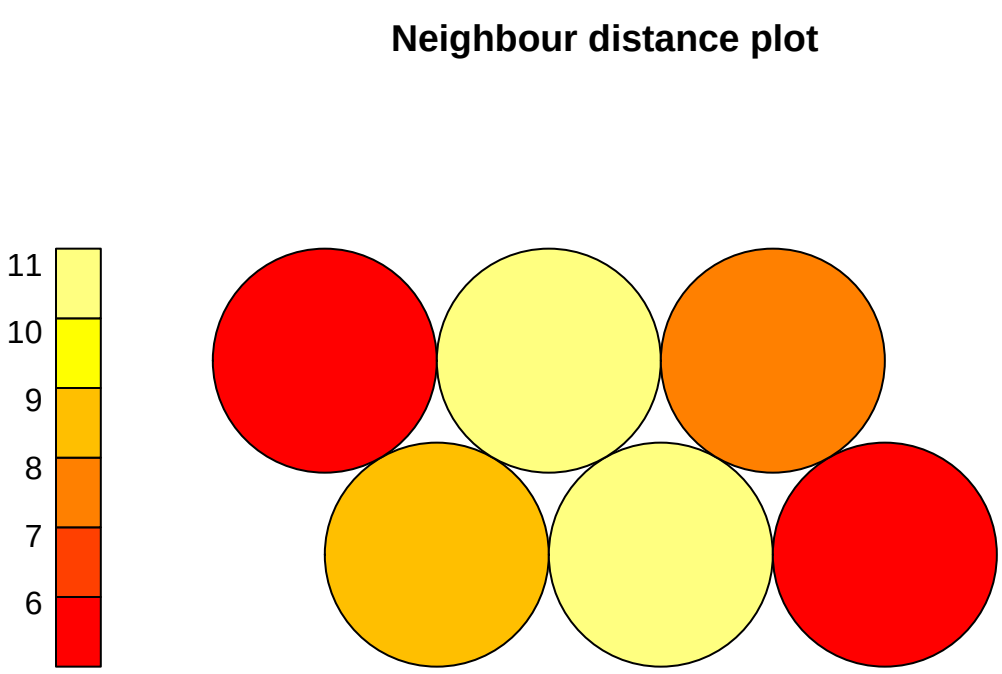


Distance Neighbour Plot- Small

This is sometimes called the "U-Matrix", it can help identify further clustering. Areas of low neighbour distance indicate groups of nodes that are similar and the further apart nodes indicate natural "borders" in

the map.

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



Heatmaps - Small

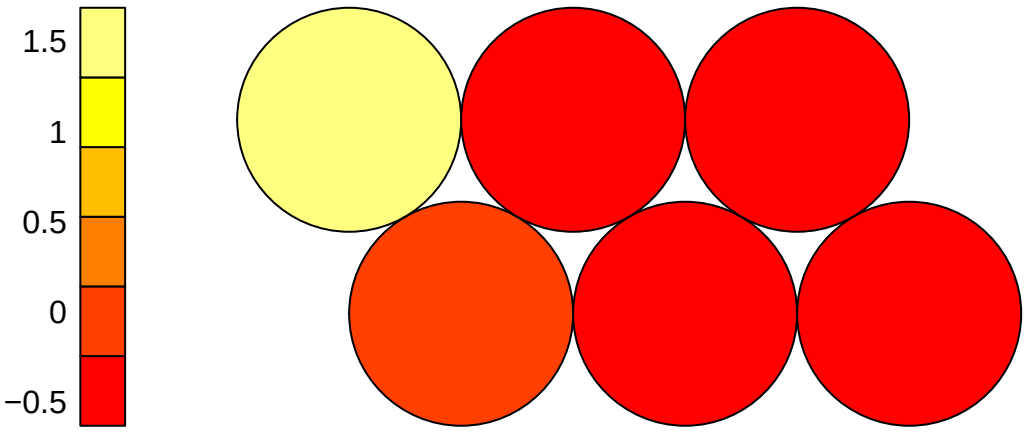
This shows the distribution of each type of tissue. This doesn't really work too well when the the map is so small. Bother is the only tissue type that contributes to two clusters.

```
head(som$codes)
```

```
##         Ambr  Aother    Bmbr  Bother    Cmbr  Cother
## [1,] -0.2249 -0.3466  1.7002 -0.4888 -0.2881 -0.3518
## [2,] -0.3703 -0.3453 -0.3880 -0.2911  1.6476 -0.2530
## [3,] -0.5426 -0.2812 -0.3952 -0.1283 -0.3544  1.7018
## [4,]  1.6873 -0.2603 -0.2699 -0.3937 -0.3717 -0.3917
## [5,] -0.4066  1.5607 -0.3410 -0.2691 -0.4157 -0.1284
## [6,] -0.6316 -0.1237 -0.6824  1.5342 -0.3055  0.2090
```

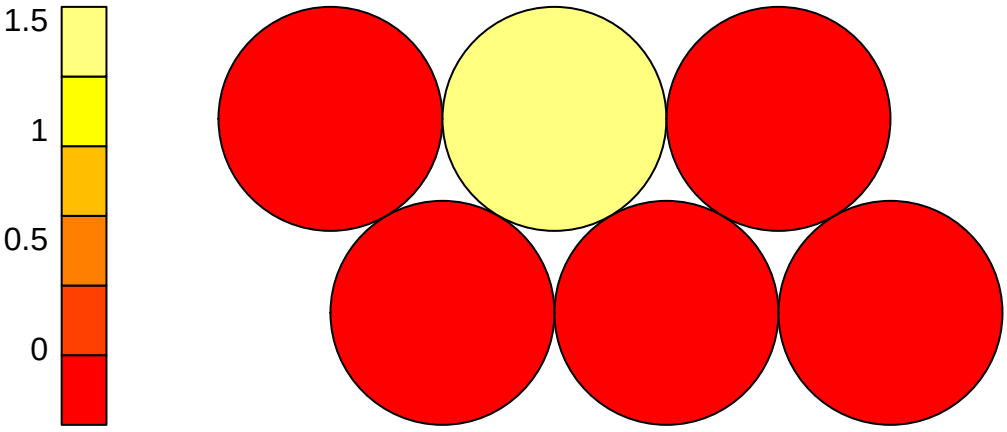
```
som$data <- data.frame(som$data) #changed to dataframe to extract column names easier.

#This is just a loop that plots the distribution of each
#tissue type across the map.
for (i in 1:6){
  plot(som, type = "property", property = som$codes[,i], main=names(som$data)[i])
  print(plot)
  }
```

Ambr



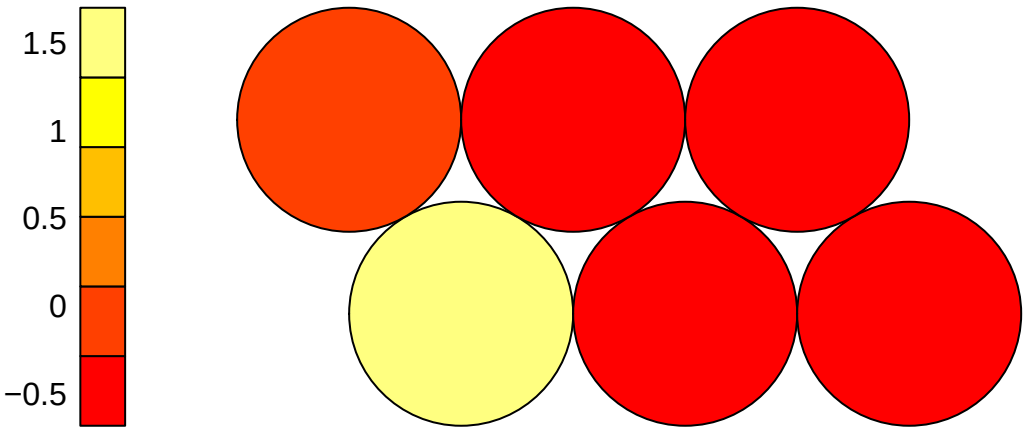

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa40237fcd0>
## <environment: namespace:graphics>
```

Aother



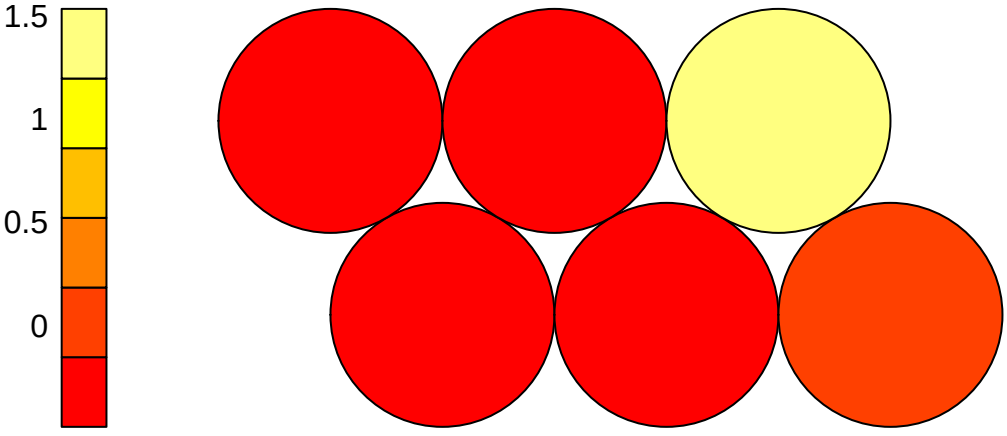

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa40237fcd0>
## <environment: namespace:graphics>
```

Bmbr



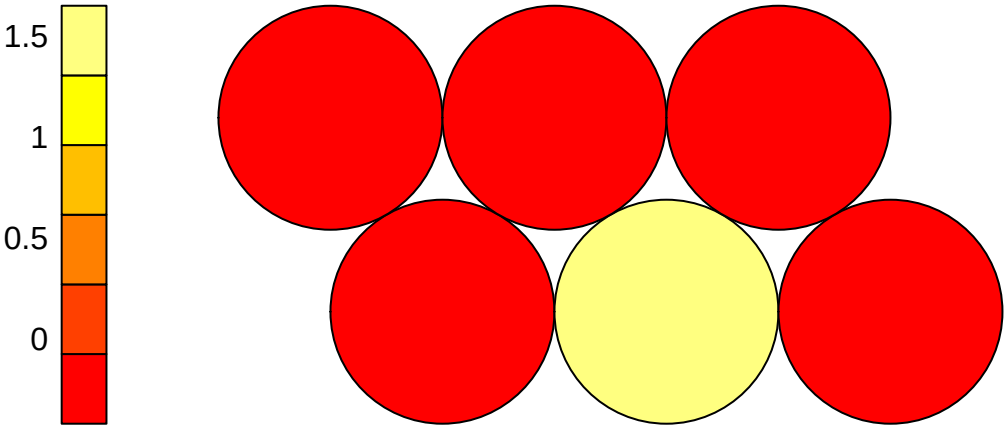

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa40237fcd0>
## <environment: namespace:graphics>
```

Bother



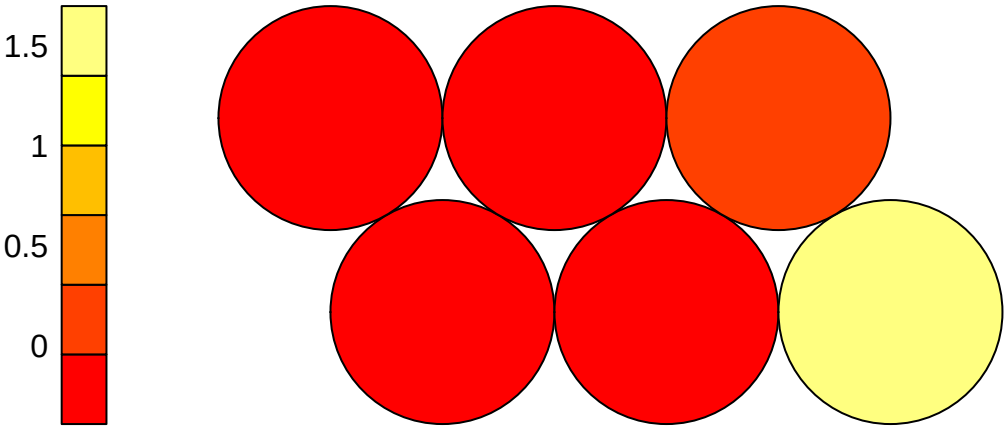

```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa40237fcd0>
## <environment: namespace:graphics>
```

Cmbr




```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa40237fcd0>
## <environment: namespace:graphics>
```

Cother




```
## function (x, y, ...)
## UseMethod("plot")
## <bytecode: 0x7fa40237fcd0>
## <environment: namespace:graphics>
```

Output

```
data.val.small <- cbind(data.val,som$unit.classif,som$distances)
#Make sure that there is just one of each value som$unit.classif and distances column.
names(data.val.small)
```

```
##  [1] "genotype"         "gene"             "Ambr"
##  [4] "Aother"           "Bmbr"             "Bother"
##  [7] "Cmbr"             "Cother"           "Ambr"
## [10] "Aother"           "Bmbr"             "Bother"
## [13] "Cmbr"             "Cother"           "PC1"
## [16] "PC2"              "PC3"              "PC4"
## [19] "PC5"              "PC6"              "som$unit.classif"
## [22] "som$distances"
```

```
summary(data.val.small)
```

```
##  genotype                    gene           Ambr           Aother
##  tf2:3580   Solyc00g005050.2.1:   2   Min.   :    0   Min.   :   0
##  wt :3580   Solyc00g005070.1.1:   2   1st Qu.:    3   1st Qu.:   4
##             Solyc00g005080.1.1:   2   Median :    8   Median :  10
##             Solyc00g005840.2.1:   2   Mean   :   52   Mean   :  40
##             Solyc00g005870.1.1:   2   3rd Qu.:   24   3rd Qu.:  25
##             Solyc00g005880.1.1:   2   Max.   :18688   Max.   :6882
##             (Other)           :7148
##       Bmbr           Bother           Cmbr           Cother
##  Min.   :   0   Min.   :   0   Min.   :   0   Min.   :   0.0
##  1st Qu.:   3   1st Qu.:   4   1st Qu.:   4   1st Qu.:   4.6
##  Median :   8   Median :  10   Median :   9   Median :  11.2
##  Mean   :  38   Mean   :  34   Mean   :  32   Mean   :  35.8
##  3rd Qu.:  24   3rd Qu.:  24   3rd Qu.:  22   3rd Qu.:  26.7
##  Max.   :4188   Max.   :3270   Max.   :6064   Max.   :2528.2
##
##       Ambr              Aother             Bmbr              Bother
##  Min.   :-2.0034   Min.   :-1.9913   Min.   :-1.9897   Min.   :-1.9507
##  1st Qu.:-0.7383   1st Qu.:-0.5578   1st Qu.:-0.7484   1st Qu.:-0.6005
##  Median :-0.3747   Median :-0.2458   Median :-0.4099   Median :-0.2519
##  Mean   :-0.0787   Mean   : 0.0468   Mean   :-0.0932   Mean   : 0.0473
##  3rd Qu.: 0.3793   3rd Qu.: 0.6183   3rd Qu.: 0.3381   3rd Qu.: 0.7505
##  Max.   : 2.0411   Max.   : 2.0412   Max.   : 2.0412   Max.   : 2.0412
##
##       Cmbr              Cother              PC1                PC2
##  Min.   :-1.8947   Min.   :-1.9805   Min.   :-2.1655   Min.   :-2.5887
##  1st Qu.:-0.6085   1st Qu.:-0.4891   1st Qu.:-1.2229   1st Qu.:-0.7821
##  Median :-0.3433   Median :-0.0958   Median :-0.0173   Median :-0.0575
##  Mean   :-0.0859   Mean   : 0.1637   Mean   : 0.0000   Mean   : 0.0000
##  3rd Qu.: 0.2657   3rd Qu.: 0.7967   3rd Qu.: 1.2493   3rd Qu.: 0.7752
##  Max.   : 2.0410   Max.   : 2.0399   Max.   : 2.5950   Max.   : 2.5007
##
##       PC3                PC4                PC5                PC6
##  Min.   :-2.4549   Min.   :-2.2853   Min.   :-2.2632   Min.   :-4.54e-15
##  1st Qu.:-0.8497   1st Qu.:-0.7254   1st Qu.:-0.6983   1st Qu.:-1.55e-15
##  Median :-0.0675   Median :-0.0808   Median :-0.0707   Median :-1.11e-16
##  Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   :-1.20e-17
##  3rd Qu.: 0.6962   3rd Qu.: 0.5879   3rd Qu.: 0.7475   3rd Qu.: 1.22e-15
```

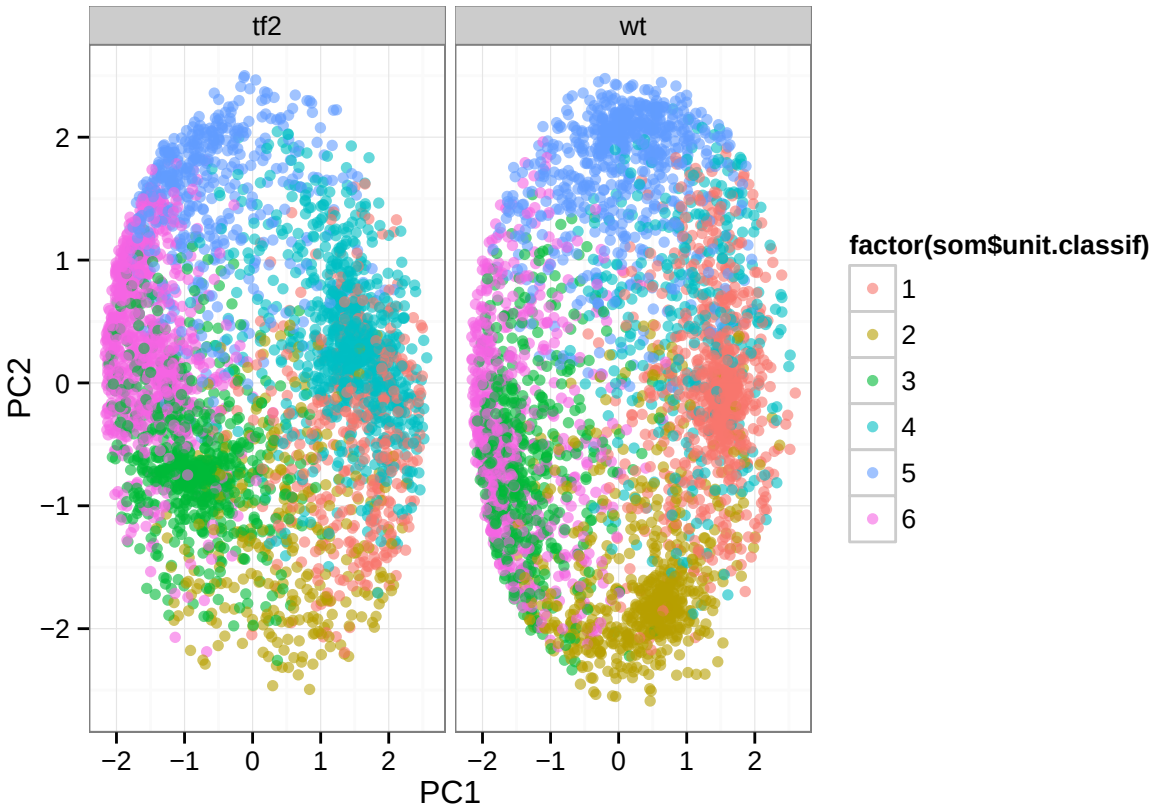
```
##  Max.   : 2.5966   Max.   : 2.2998   Max.   : 2.4924   Max.   : 4.89e-15
##
##  som$unit.classif som$distances
##  Min.   :1.00     Min.   :0.132
##  1st Qu.:2.00     1st Qu.:0.589
##  Median :4.00     Median :1.368
##  Mean   :3.62     Mean   :1.670
##  3rd Qu.:5.00     3rd Qu.:2.543
##  Max.   :6.00     Max.   :6.409
##
```

Visualize back to PC space

```
plot.data <- data.val.small
names(plot.data)
```

```
##  [1] "genotype"         "gene"             "Ambr"
##  [4] "Aother"           "Bmbr"             "Bother"
##  [7] "Cmbr"             "Cother"           "Ambr"
## [10] "Aother"           "Bmbr"             "Bother"
## [13] "Cmbr"             "Cother"           "PC1"
## [16] "PC2"              "PC3"              "PC4"
## [19] "PC5"              "PC6"              "som$unit.classif"
## [22] "som$distances"
```

```
p <- ggplot(plot.data, aes(PC1, PC2, colour=factor(som$unit.classif))) #use unit.classif for smaller da
p + geom_point(alpha = .6) + facet_grid(.~genotype) + theme_bw()
```



References

1. R self Organizing map tutorial