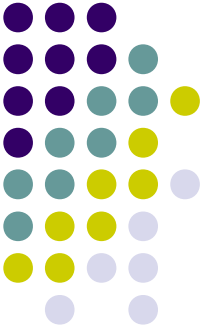
Lecture #9

Cluster Analysis



Outline

- Definitions
- General applications
- Methods
  - Hierarchical clustering
    - Distance metrics
    - Grouping metrics
    - Pros and cons
  - *k*-means
    - Pros and cons
  - SOM
    - Pros and cons

What is clustering

- Clustering is an unsupervised analysis technique used to group similar objects (genes or samples) together

- Builds structure to help explain the relationships that may exist between the objects
  - Dendrogram

Applications of clustering

- Like many techniques in microarray analysis, clustering is not a new approach for classification
- Clustering has been utilized as a classification method in multiple applications
  - **Phylogenetics (taxonomy)**
  - **Sociology (demographics)**
  - **Linguistics (word usage)**
  - **Business (money market)**

Some clustering methods

- There are multiple types of clustering methods
  - Hierarchical (HCA)
  - *k*-means
  - Self-organizing maps (SOM)
  - Clique partitioning

Some clustering methods

- There are multiple types of clustering methods
  - **Hierarchical (HCA)**
  - *k*-means
  - Self-organizing maps (SOM)

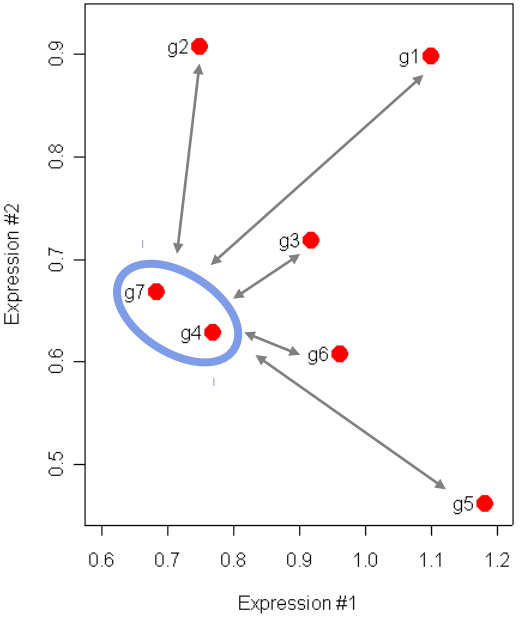
HCA Clustering Procedure (genes)

1.) Each gene begins as its own cluster

2.) Look for “closest” pair of clusters

3.) Merge them into their own cluster

4.) Evaluate all distances from this new cluster and the remaining clusters



HCA Clustering Procedure (genes)

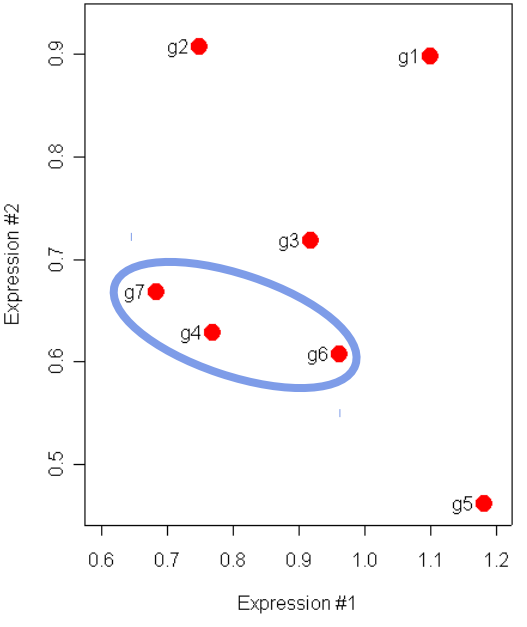
1.) Each gene begins as its own cluster

2.) Look for “closest” pair of clusters

3.) Merge them into their own cluster

4.) Evaluate all distances from this new cluster and the remaining clusters

5.) Repeat until no clusters remain



Distances and Similarities

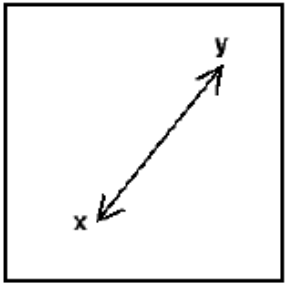
- The choice of distant or similarity metric can effect what we determine to be the “closest” genes
  - Euclidean
  - Manhattan
  - Pearson’s correlation

Euclidean Distance

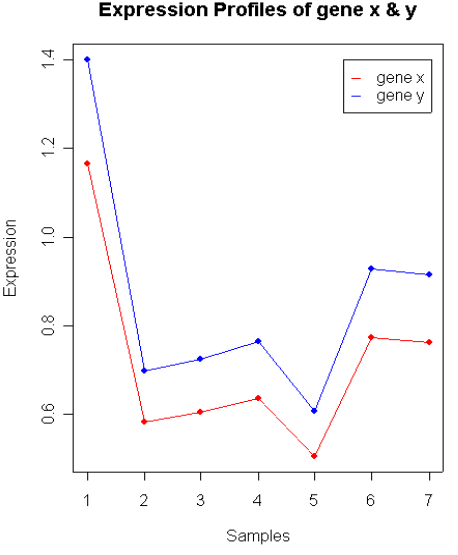
- Square root of summation of the squared differences between two gene vectors

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

where x_i is expression value at sample i
y_i is expression value at sample i
n is number of total samples



Euclidean

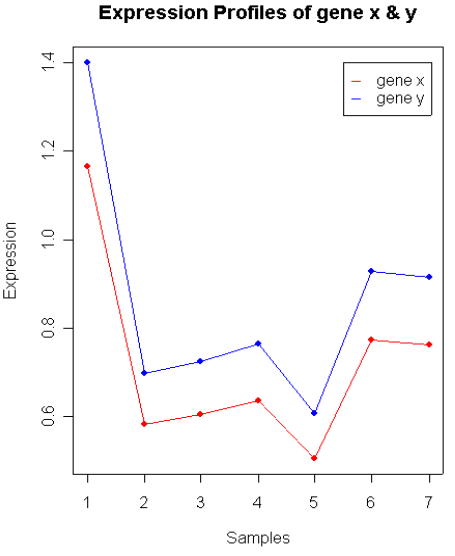


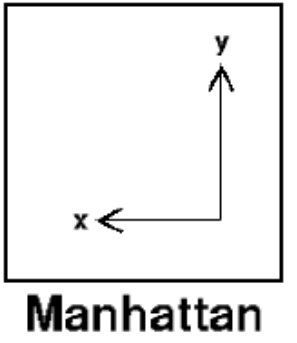
Manhattan Distance

- Absolute value of summation of the differences between two gene vectors

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

where x_i is expression value at sample i
y_i is expression value at sample i
n is number of total samples



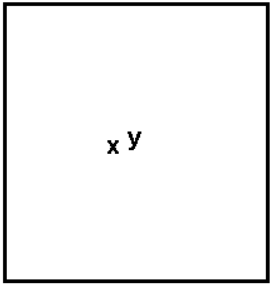


Pearson's Correlation

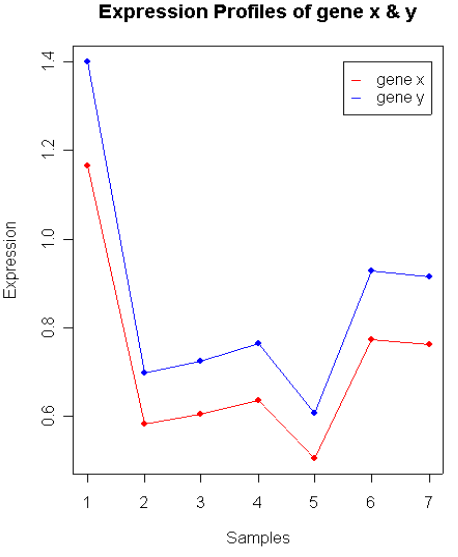
- Summation of products of two gene vectors, normalized by the product of their standard deviations

$$r = \frac{\sum XY - \frac{(\sum X)(\sum Y)}{n}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{n}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{n}\right)}}$$

where x_i is expression value at sample i
y_i is expression value at sample i
n is number of total samples



Pearson's r



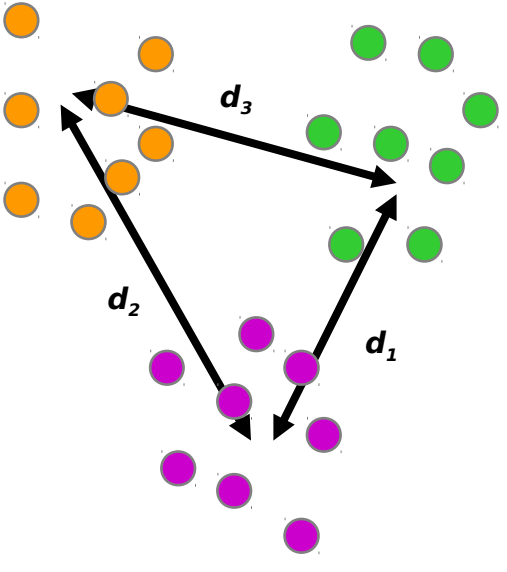
Grouping metrics

- The choice of grouping method can alter the cluster membership of the genes as they are partitioned
  - Average linkage (UPGMA)
  - Single linkage (nearest neighbor)
  - Complete linkage (furthest neighbor)
  - Ward's method

Average Linkage

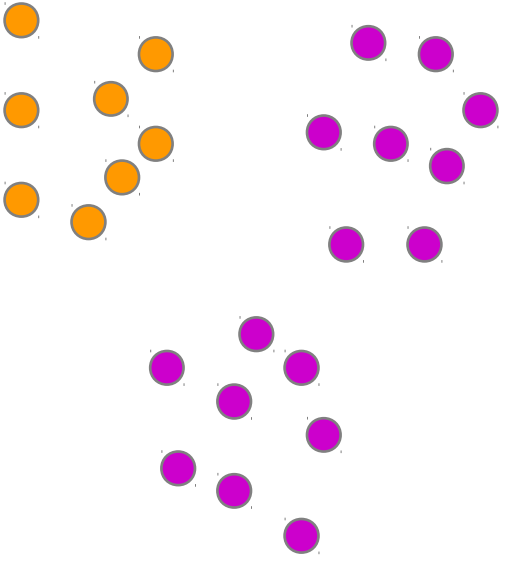
- Calculate the distance between each point (gene) in a cluster and all other points (genes) in another cluster.

- The two clusters with the lowest average distance are joined together to form the new cluster.



Average Linkage

- Two clusters are grouped to form new cluster

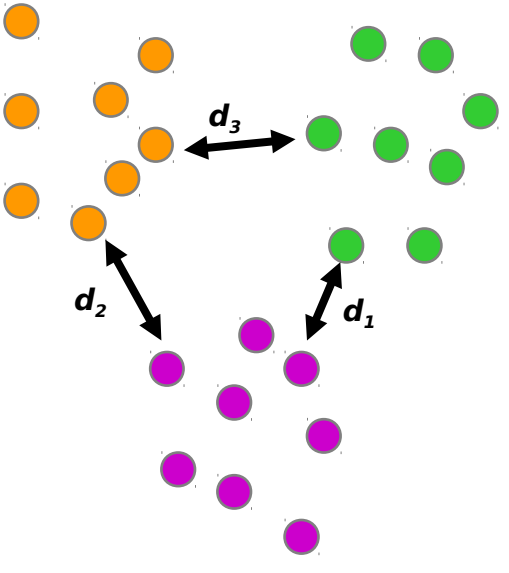


Single Linkage

- Calculate the distance between each point (gene) in a cluster and all other points (genes) in another cluster.

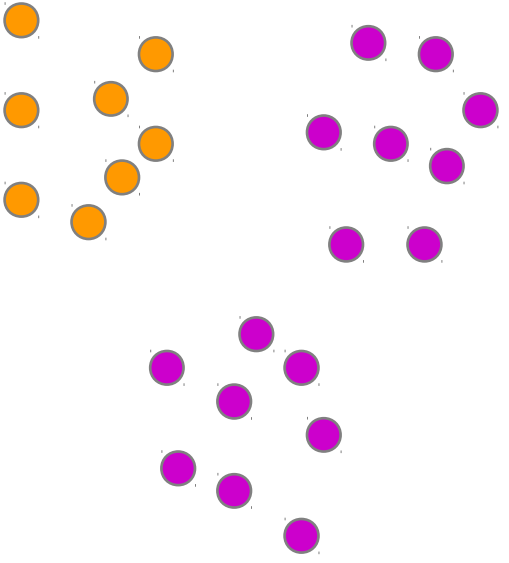
- The two clusters with members having the least dissimilarity are joined together (nearest neighbors).

$$d_j = \min (p_i - p_k)$$



Single Linkage

- Two clusters are grouped to form new cluster

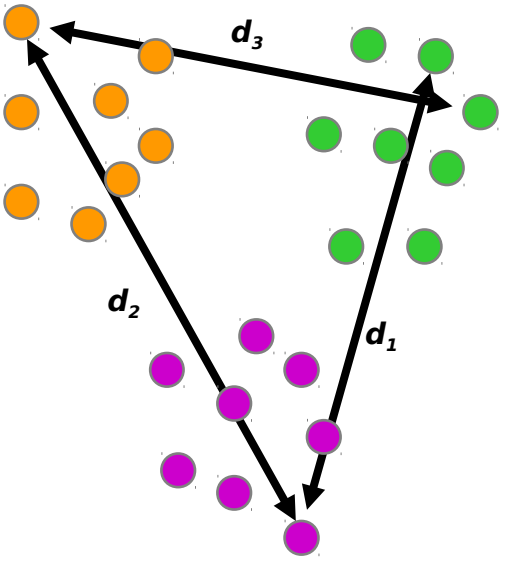


Complete Linkage

- Calculate the distance between each point (gene) in a cluster and all other points (genes) in another cluster.

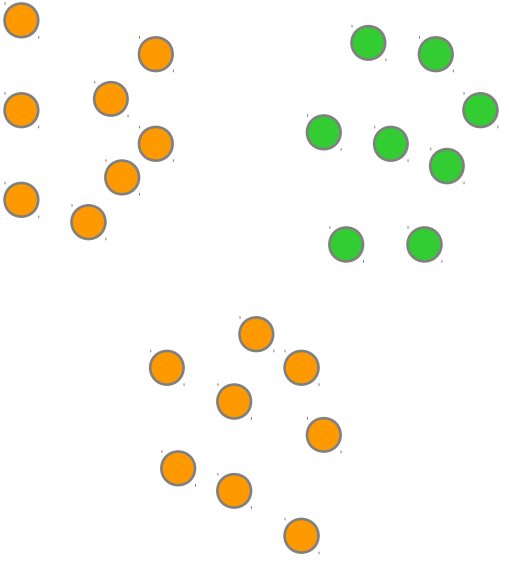
- The two clusters with members having the maximum dissimilarity are joined together (furthest neighbors).

$$d_j = \max (p_i - p_k)$$



Complete Linkage

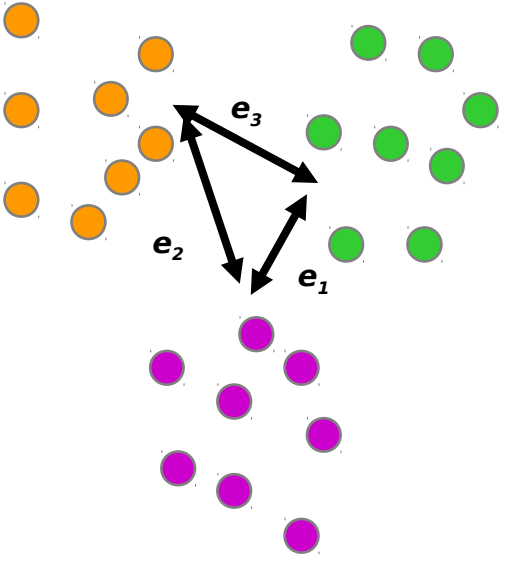
- Two clusters are grouped to form new cluster



Ward's Method

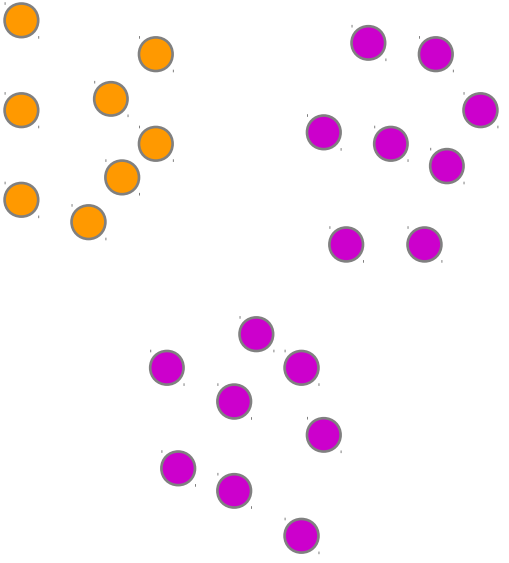
- Determine mean of each cluster from average linkage
- Calculate total sum of squared deviations from the mean of each cluster
- The two clusters with the smallest increase in error sum of squares are joined together

$e_j = \sum (p_\mu - p_i)^2$

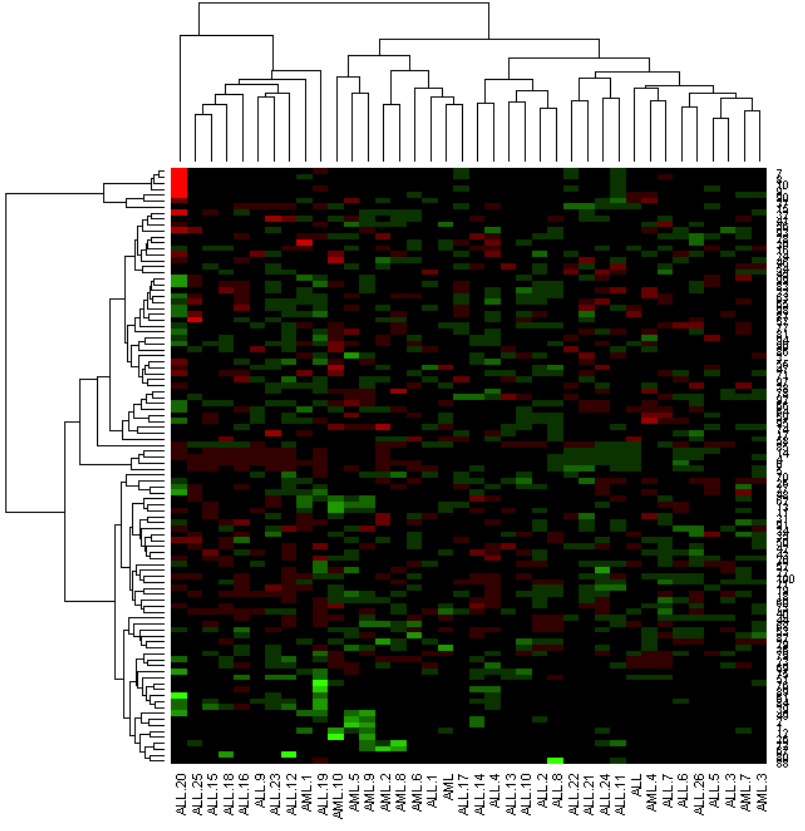


Ward's Method

- Two clusters are grouped to form new cluster



Dendrogram Example



HCA pros and cons

- Advantages
  - Provides an informative display of ordered objects
  - Sub-clusters may provide to be useful in discovery
  - Relatively simple to implement

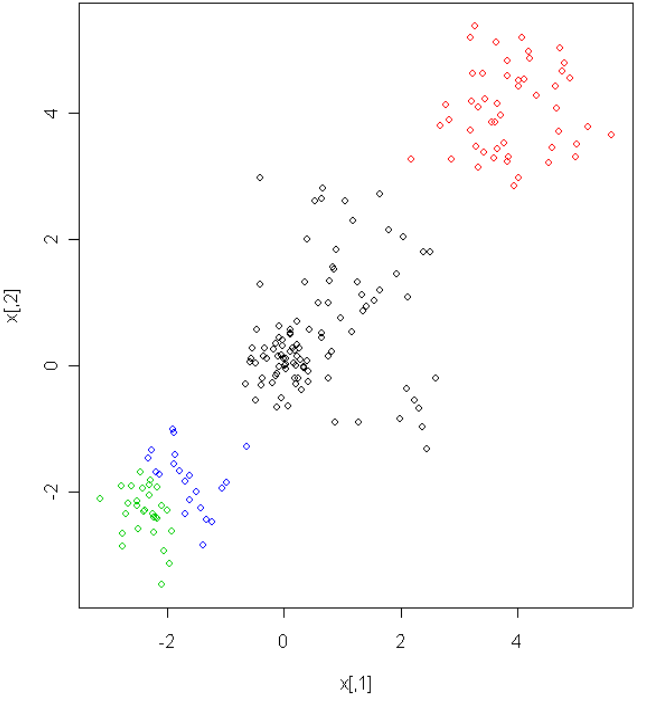
- Disadvantages
  - Early groupings are nested in later result groupings
  - Grouping metrics can provide inconsistent results
  - Tree structure is dynamic (i.e. not model-based)

Some clustering methods

- There are multiple types of clustering methods
  - Hierarchical (HCA)
  - ***k*-means**
  - Self-organizing maps (SOM)

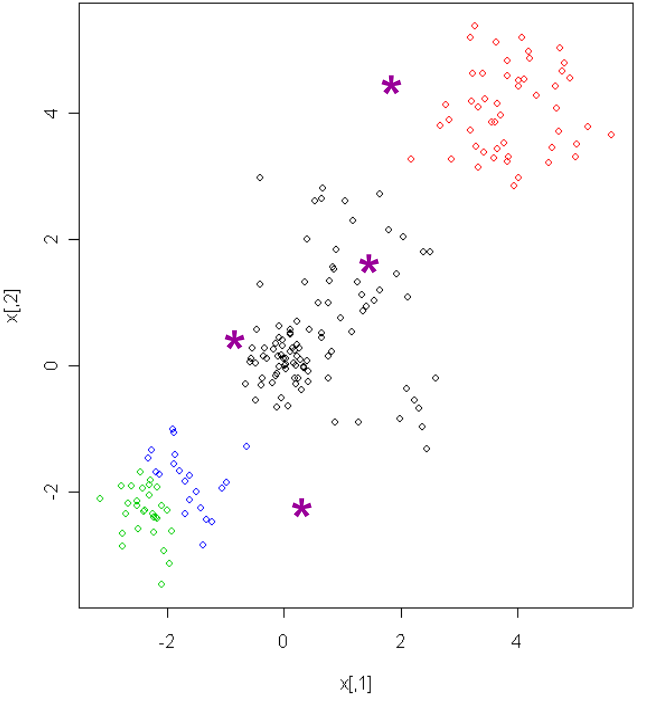
k-means procedure

- Start with a user-defined number of clusters (*k*-value)



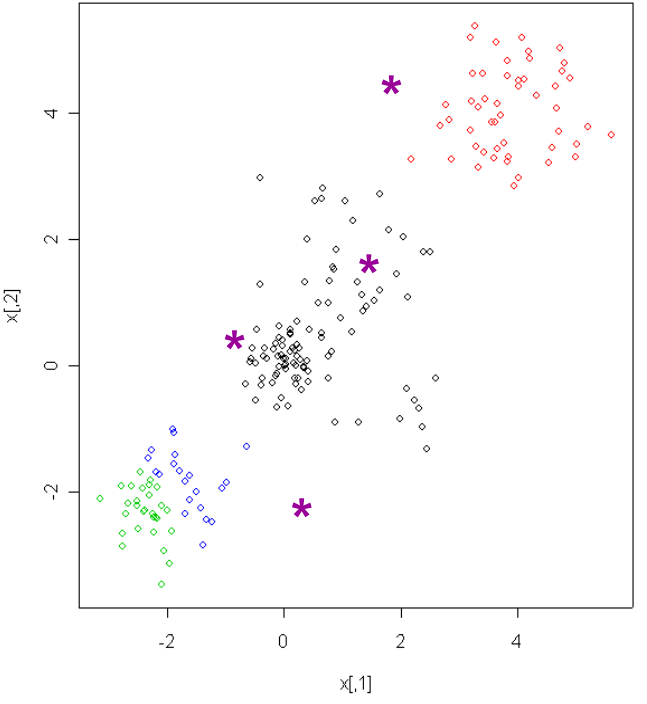
k-means procedure

- Start with a user-defined number of clusters (*k*-value)
- *k* points are projected into space to estimate the center of *k* separate clusters



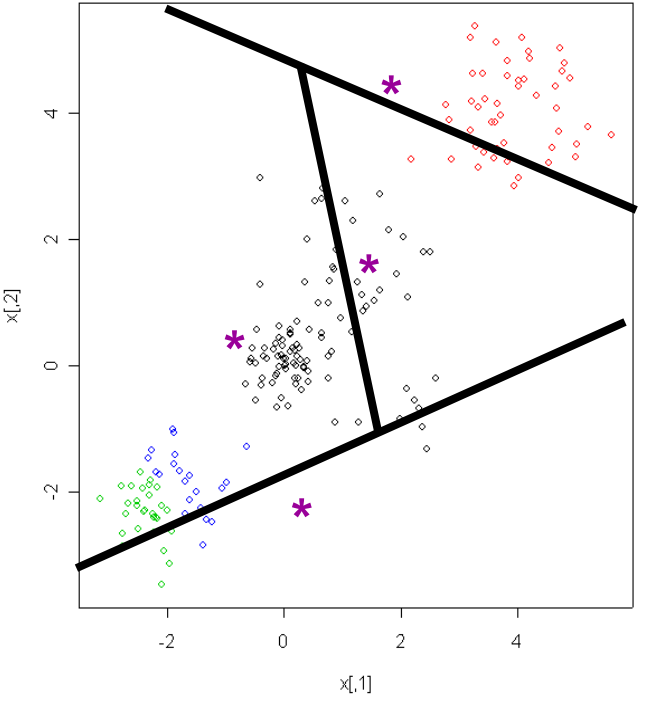
k-means procedure

- Start with a user-defined number of clusters (*k*-value)
- *k* points are projected into space to estimate the center of *k* separate clusters
- The distances from each point to all *k* clusters are calculated to determine which points are closest to which cluster centers



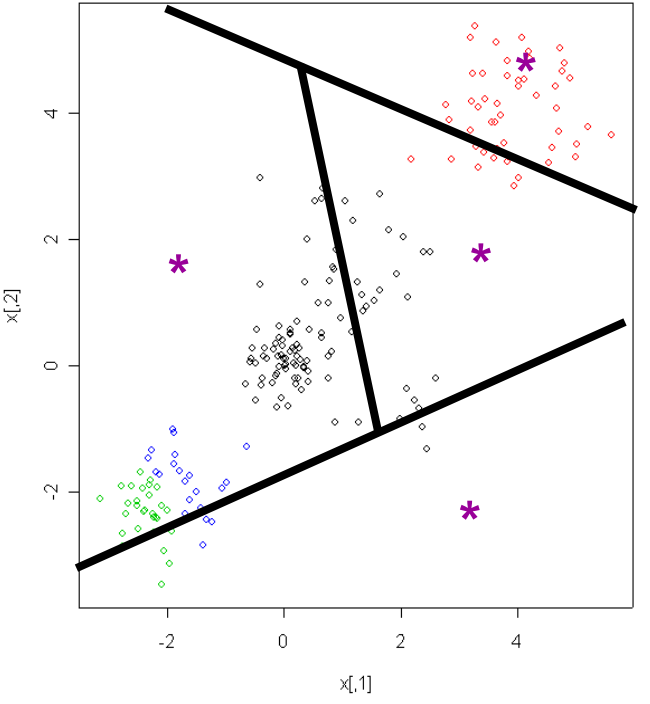
k-means procedure

- Start with a user-defined number of clusters (*k*-value)
- *k* points are projected into space to estimate the center of *k* separate clusters
- The distances from each point to all *k* clusters are calculated to determine which points are closest to which cluster centers
- The n-dimensional space is then partitioned into *k* regions of points



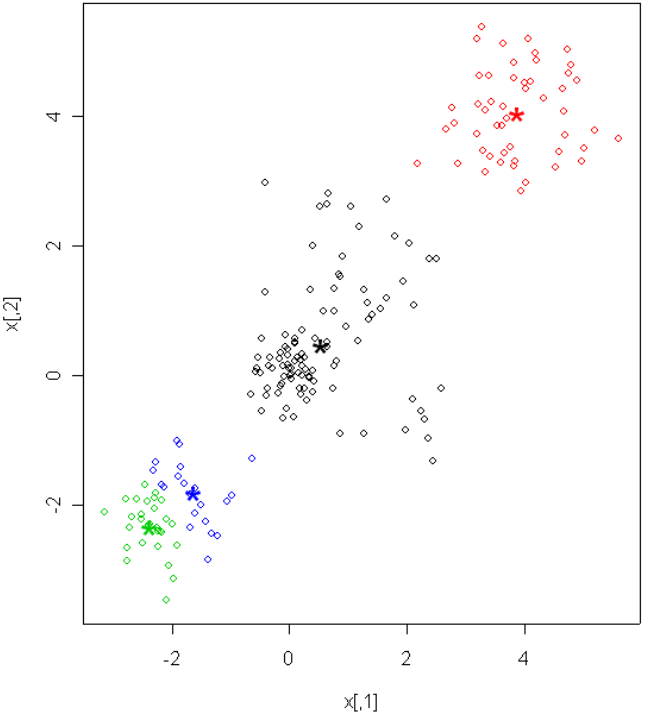
k-means procedure

- Start with a user-defined number of clusters (*k*-value)
- *k* points are projected into space to estimate the center of *k* separate clusters
- The distances from each point to all *k* clusters are calculated to determine which points are closest to which cluster centers
- The n-dimensional space is then partitioned into *k* regions of points
- The *k* cluster centers are adjusted to where the middle point (centroid) of the partitioned space is located



k-means procedure

- Start with a user-defined number of clusters (*k*-value)
- *k* points are projected into space to estimate the center of *k* separate clusters
- The distances from each point to all *k* clusters are calculated to determine which points are closest to which cluster centers
- The n-dimensional space is then partitioned into *k* regions of points
- The *k* cluster centers are adjusted to where the middle point (centroid) of the partitioned space is located
- This process is repeated n iterations or until convergence has occurred



k-means pros and cons

- Advantages
  - Good discovery tool
  - Can alleviate fuzzy partitioning of genes/samples
    - Specify *k* and algorithm will return *k* clusters
- Disadvantages
  - What is the appropriate number of clusters when the data structure is unknown?
  - Number of iterations necessary for convergence can be computationally expensive
  - The convergence criteria for *k*-means may only provide a local optima
  - Starting point may alter termination points
  - Robustness in distance algorithms (similar to HCA)

Some clustering methods

- There are multiple types of clustering methods
  - Hierarchical (HCA)
  - *k*-means
  - **Self-organizing maps (SOM)**

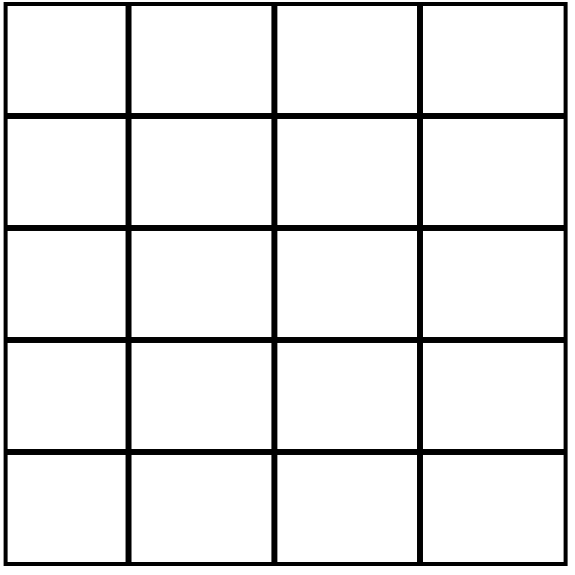
SOMs

- Self-organizing maps are an iterative fitting algorithm similar to k-means

- The grid shape and size are the parameters that are user defined
  - Similar to the *k* clusters from *k*-means
  - Hexagonal, rectangular, etc.
  - Dimensions (e.g. 3x4)

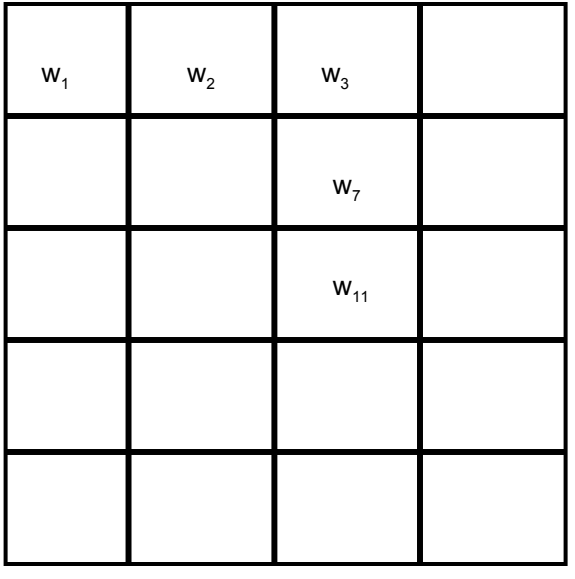
SOM procedure[3]

- A initial grid type of n x m dimensions is specified



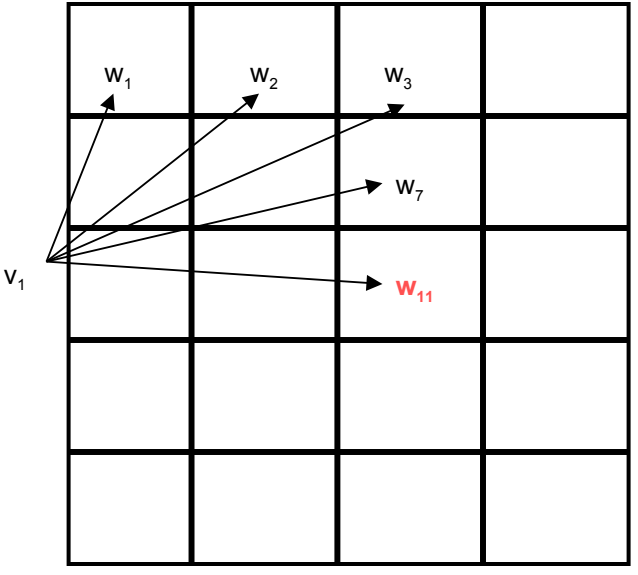
SOM procedure[3]

- A initial grid type of n x m dimensions is specified
- The cells in the grid are identified as nodes and each node is initialized with some random weight (vector of weights)



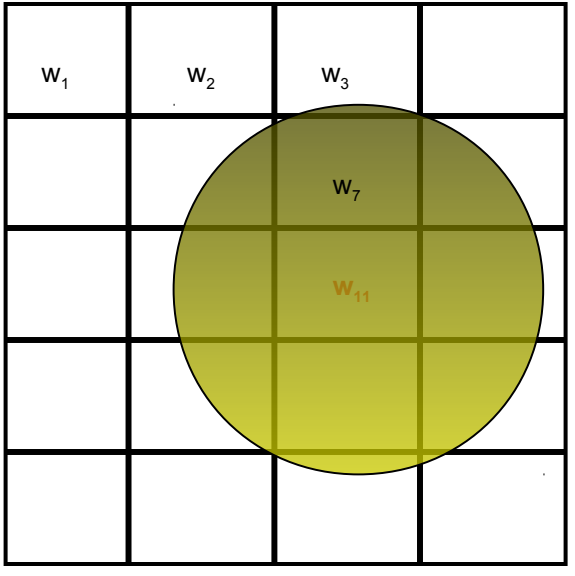
SOM procedure[3]

- A initial grid type of n x m dimensions is specified
- The cells in the grid are identified as nodes and each node is initialized with some random weight (vector of weights)
- A vector is chosen at random and compared to the grid nodes to determine which weight is the most similar to the input vector
  - Distance such as Euclidean distance is calculated between the input vector and all other node vectors
  - The most similar node is referred as the best matching node (BMU)



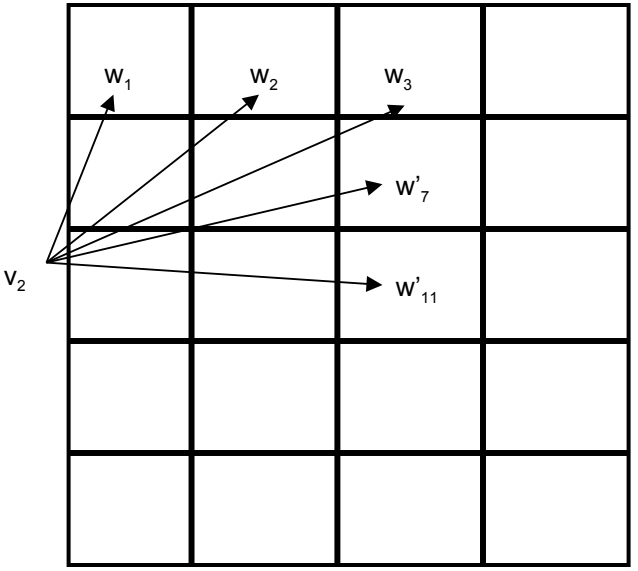
SOM procedure[3]

- A initial grid type of n x m dimensions is specified
- The cells in the grid are identified as nodes and each node is initialized with some random weight (vector of weights)
- A vector is chosen at random and compared to the grid nodes to determine which weight is the most similar to the input vector
  - Distance such as Euclidean distance is calculated between the input vector and all other node vectors
  - The most similar node is referred as the best matching node (BMU)
- A neighborhood radius is established around the BMU node and the weights for the neighbor nodes to the BMU are adjusted according to similarity to the input vector
  - Radius gets smaller and smaller each iteration by a decay function



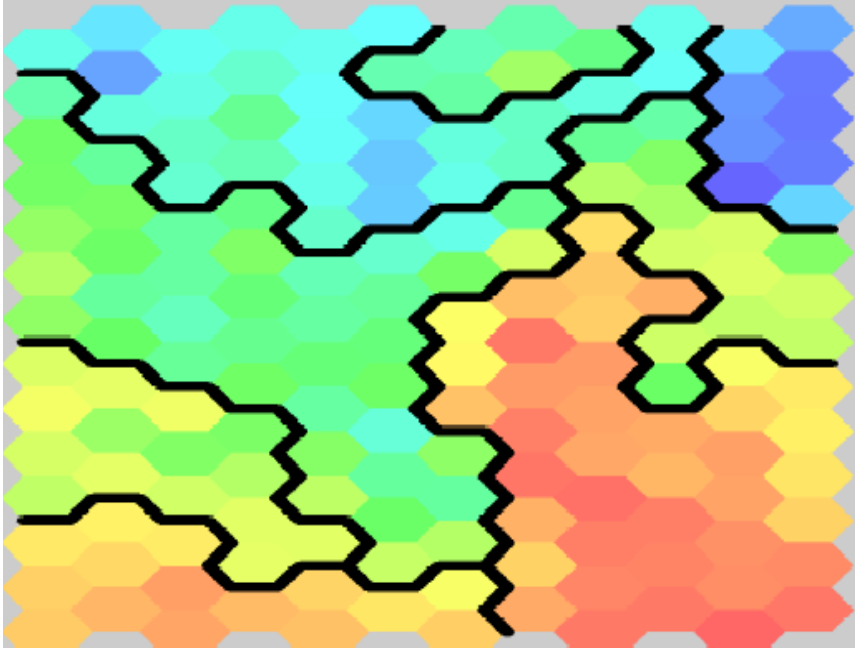
SOM procedure[3]

- A initial grid type of n x m dimensions is specified
- The cells in the grid are identified as nodes and each node is initialized with some random weight (vector of weights)
- A vector is chosen at random and compared to the grid nodes to determine which weight is the most similar to the input vector
  - Distance such as Euclidean distance is calculated between the input vector and all other node vectors
  - The most similar node is referred as the best matching node (BMU)
- This process is repeated N times,



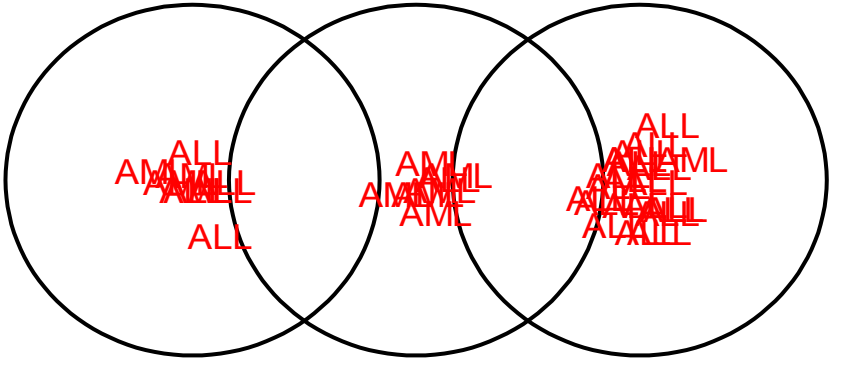
SOM procedure[3]

- Visual depiction of class structure



SOM Classification example

Golub et al. data with 1x3 grid



SOMs pros and cons

- Advantages
  - Good discovery tool
- Disadvantages
  - Can be computationally expensive
  - Much of the partitioning is dependent upon the grid design

References

- [1]Alizadeh A, et. al. (2000) Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*. **403**, 503-511.

- [2]Datta S and Datta S. (2003) Comparison and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics*. **19**, 459-466.

- [3]SOM tutorial
  - http://www.ai-junkie.com/som1.html

R Code

```
# Load Golub et al. data
library(multtest)
data(golub);
dat <- golub
dat <- as.data.frame(dat)
ann.dat2 <- golub.cl    # class labels ALL=0; AML=1
ann <- factor(c(rep("ALL",27),rep("AML",11)))

# HCA heat map of top 100 genes
# adjust colors for heat map
dat <- dat[1:100,]      # only use top 100 genes for computational speed
hm.rg <-
     c("#FF0000","#CC0000","#990000","#660000","#330000","#000000","#000000","#0A3300","#146600","#1F9900
     ","#29CC00","#33FF00")
names(dat) <- ann
heatmap(as.matrix(dat),col=hm.rg)

# kmeans clustering with random data
library(mva)
x <- rbind(matrix(rnorm(100, sd = 0.3),ncol=2),matrix(rnorm(100, mean = 1, sd =
     0.9),ncol=2),matrix( rnorm(100, mean = 4, sd = 0.75),ncol=2),matrix(rnorm(100, mean = -2, sd = 0.5),
     ncol = 2))
cl <- kmeans(x, centers=4, iter.max=20)
plot(x, col = cl$cluster,cex=1)
points(cl$centers, col = 1:4, pch = "*",cex=2.5)

# som clustering on Golub et al. samples
library(class)
dat <- t(dat)           # transponse data
dimnames(dat)[[1]] <- ann
gr <- somgrid(xdim=3,ydim=1,topo = "rectangular")          # specify grid type and dimensions
dat.som <- SOM(dat,gr)
```

R Code

```
# plot classifier
bins <- as.numeric(knn1(dat.som$code, dat,c(1:3)))
plot(crabs.som$grid, type = "n")
symbols(dat.som$grid$pts[, 1],dat.som$grid$pts[, 2],circles = rep(0.85, 3), inches = FALSE, add = TRUE)
text(dat.som$grid$pts[bins, ] + rnorm(76, 0, 0.1),as.character(ann),cex=0.6,col='red')
```