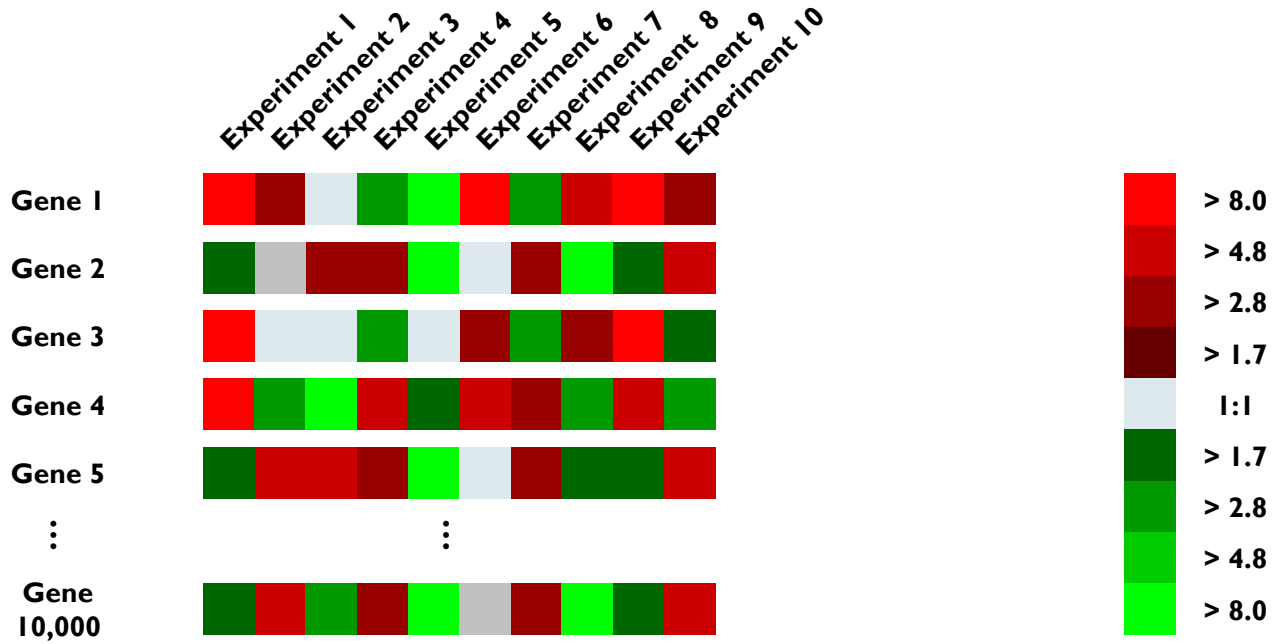
Genome wide analysis results in gene lists

- When analyzing high-throughput data, like Microarray experiment, the end result is often a list of genes.
  - Differentially expressed genes.
  - Cluster of highly correlated genes.
- A natural next step is to identify the commonality between the genes in the list.
  - Similar annotations
  - Same Pathway
  - Components of a Protein Complex

Gene lists as a discovery tool

- Depending on how the gene list was created, the genes can be used for discovering new things
  - For example if you have a cluster of highly correlated genes. One can look for novel Transcription Factor Binding sites by aligning the promoter regions of the genes in the cluster.
  - Many genes in the genome are still annotated as "unknown function". Finding an "unknown" gene in a list consisting of genes only up-regulated by a given treatment allows the biologists to provide a putative function for the unknown gene.

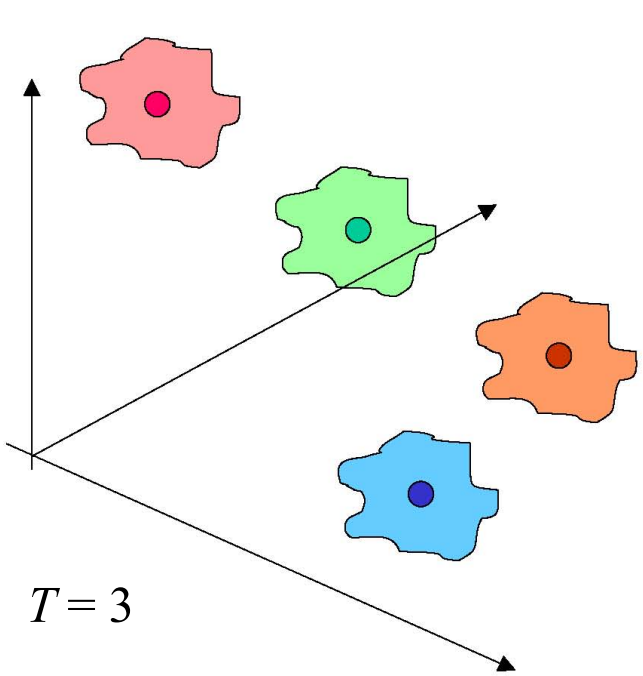
Gene expression can be assayed across many different conditions

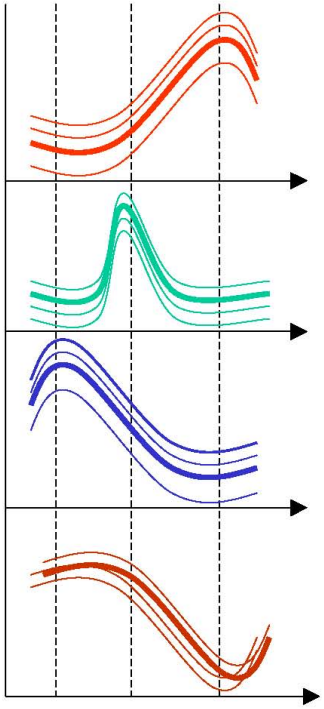


A separate microarray experiment is performed using mRNA isolated from each different “condition”, e.g.:

- Developmental time course
- Time course after exposure to some environmental stimulus (chemical, light/dark, etc.)
- Different tissues
- Normal vs. diseased tissue

Clustering (genes)



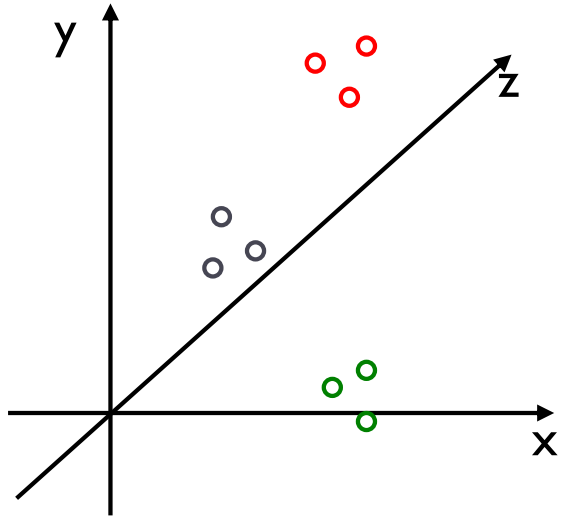


Genes with similar expression profiles are likely to have common or related functions, and possibly to be co-regulated

Similarly, ***conditions*** can be ***classified*** into different groups based on similarities in their expression profiles (all or subsets of genes).

Gene expression in multiple dimensions

Consider 3 experiments: x, y, and z



The expression vector for each gene can be represented as a point in 3-dimensional space, in which each axis represents the expression level in a different condition.

Genes with similar expression patterns fall nearby one another in this multi-dimensional space.

Coordinated Gene Expression

Which genes are co-expressed?

- Hierarchal clustering
- K-means clustering

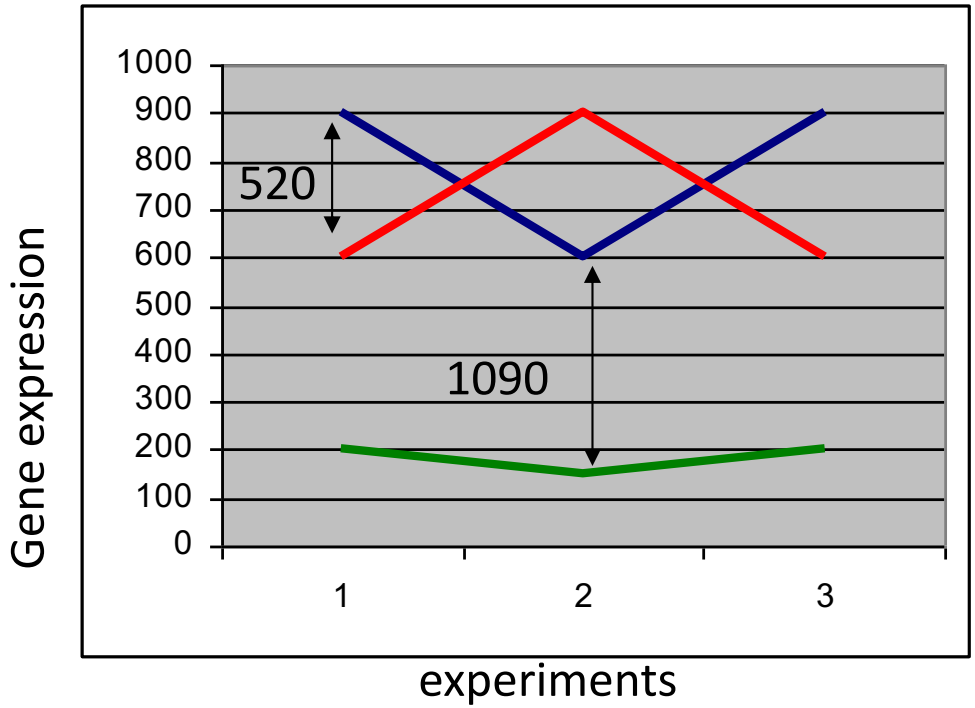
Calculating Distance

- Distance is the most natural method for numerical data
- Lower values indicate more similarity
- Distance metrics
  - Euclidean distance
  - Manhattan distance
  - Etc.
- Does not generalize well to non-numerical data
  - What is the distance between "male" and "female"?

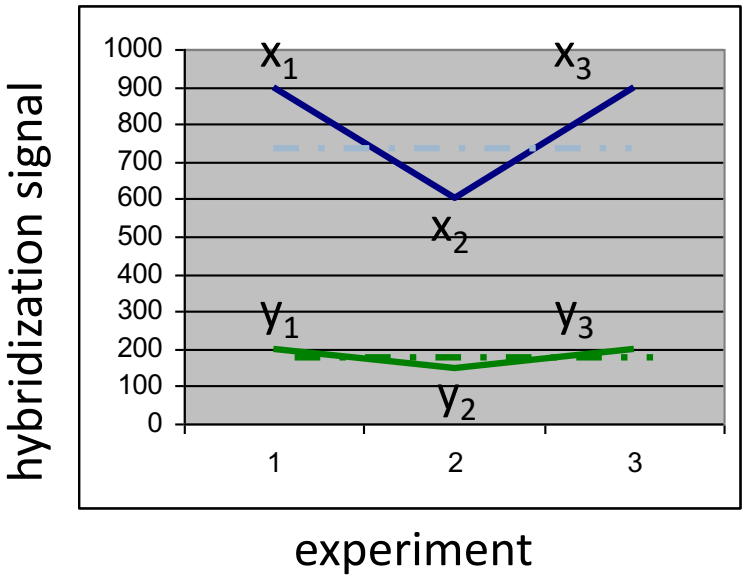
Euclidean distance

Implication for gene expression: **the magnitude of expression values will determine distances**

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



Variance and Covariance



variance measures dispersion from a mean value

$$\frac{(x_1-\bar{x}_b)^2+(x_2-\bar{x}_b)^2+(x_3-\bar{x}_b)^2}{n-1}$$

Intuitively, covariance is the measure of how much two variables vary together

$$\frac{(x_1-\bar{x}_b)(y_1-\bar{y}_g)+\ldots\ldots}{n-1}$$

covariance and correlation

Start with the concept of covariance

$$Cov_{xy} = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1}$$

Normalize the measure by taking the variance of two measurements

VarX and VarY

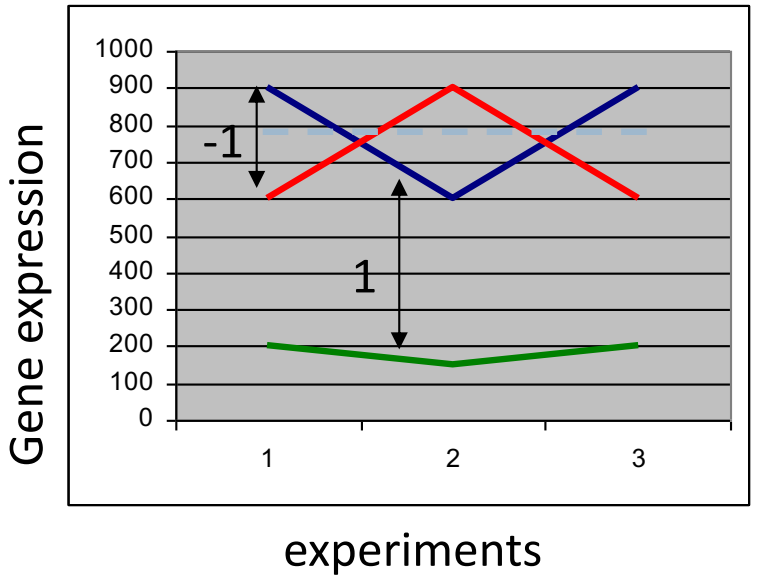
$$\sqrt{(VarX)(VarY)}$$

Pearson correlation has the nice property of varying between -1 and 1

Pearson correlation coefficient

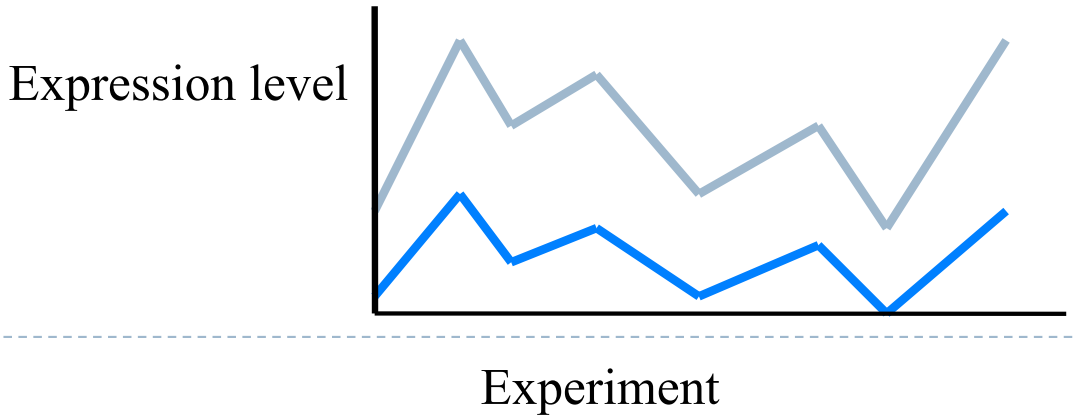
$$\frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{n-1} \Big/ \sqrt{(VarX)(VarY)}$$

Implication for gene expression:
the shape of gene expression responses will determine similarity



Calculating Numerical Similarity

- Traditionally over the range [0.0, 1.0]
  - 0.0 = no similarity, 1.0 = identity
- Converting distance to similarity
  - Distance and similarity are two sides of the same coin
  - To obtain similarity from distance, take the maximum pairwise distance and subtract from 1.0
- Pearson correlation
  - Removes magnitude effects
  - In range [-1.0, 1.0]
    - -1.0 = anti-correlated, 0.0 = no correlation, 1.0 = perfectly correlated
  - In the example below, the dark and light blue lines have high correlation, even though the distance between the lines is significant



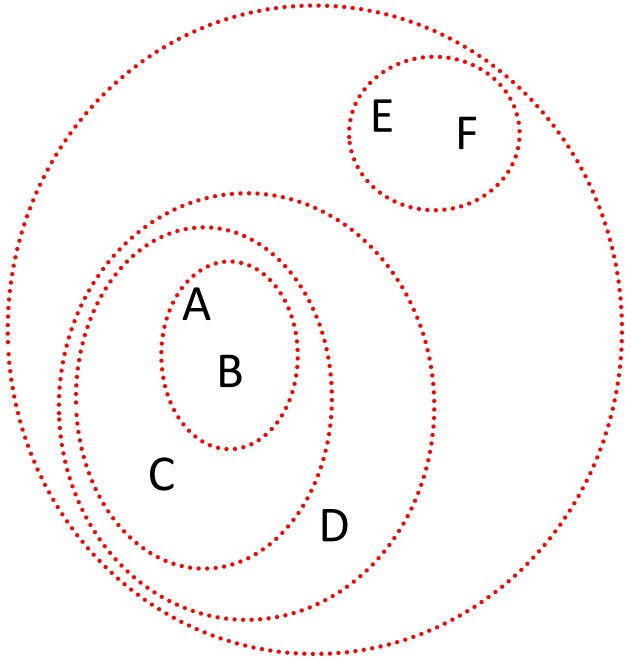
Clustering approaches

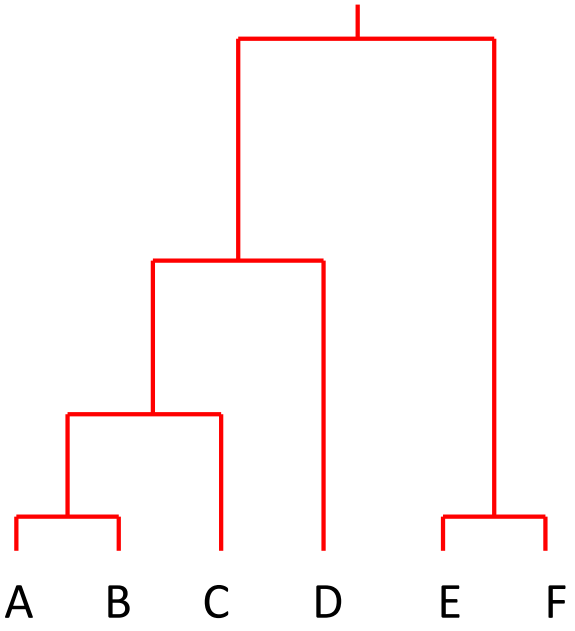
- Agglomerative: hierarchical
- Divisive: partitioning methods

Hierarchical Clustering

- Find the pair(s) with the highest pairwise similarity
- Join these as a group and calculate an "average" profile (*single*, *average*, or *complete* linkage)
- Iteratively join groups until all are linked

This example illustrates single-linkage clustering in Euclidean space on 6 points.

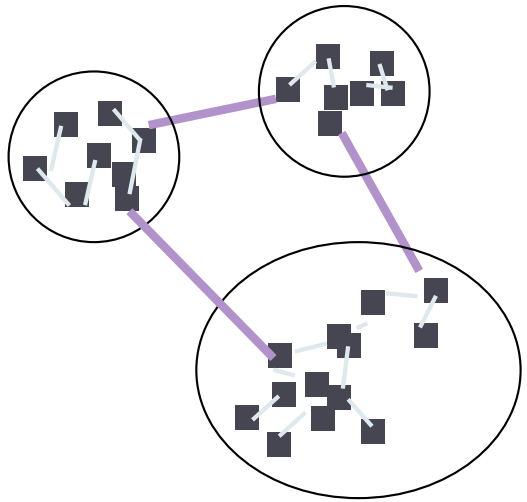




The UPGMA method of phylogenetic reconstruction uses average linking …

Clustering approaches

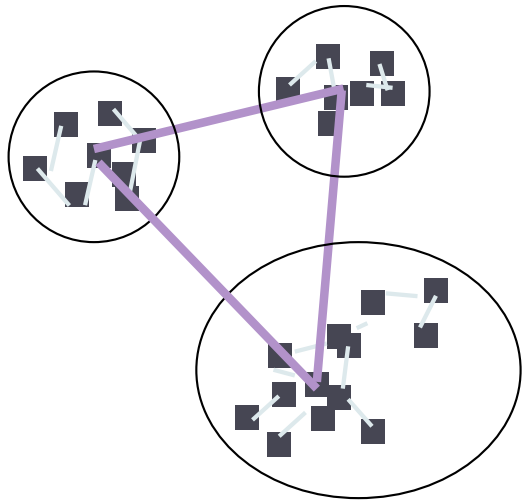
- Agglomerative
  - Single linkage



(closest points are used)

Clustering approaches

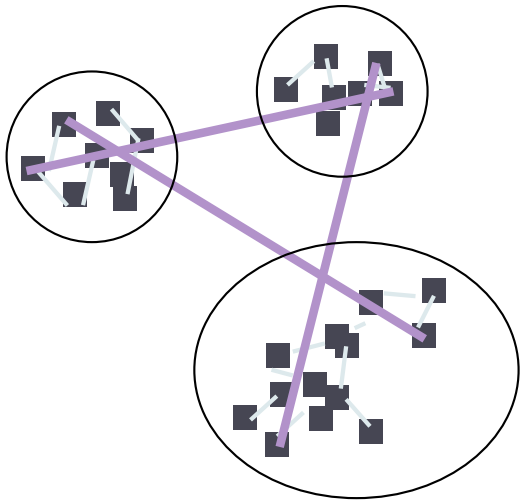
- Agglomerative
  - Single linkage
  - Centroid linkage



(center used for distance)

Clustering approaches

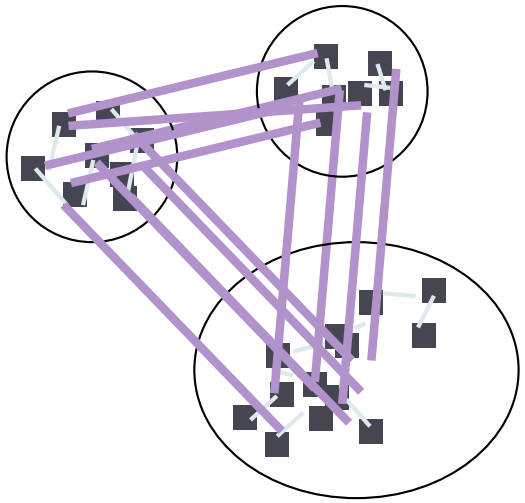
- Agglomerative
  - Single linkage
  - Centroid linkage
  - Complete linkage



(furthest points are used)

Clustering approaches

- Agglomerative
  - Single linkage
  - Centroid linkage
  - Complete linkage
  - Average linkage

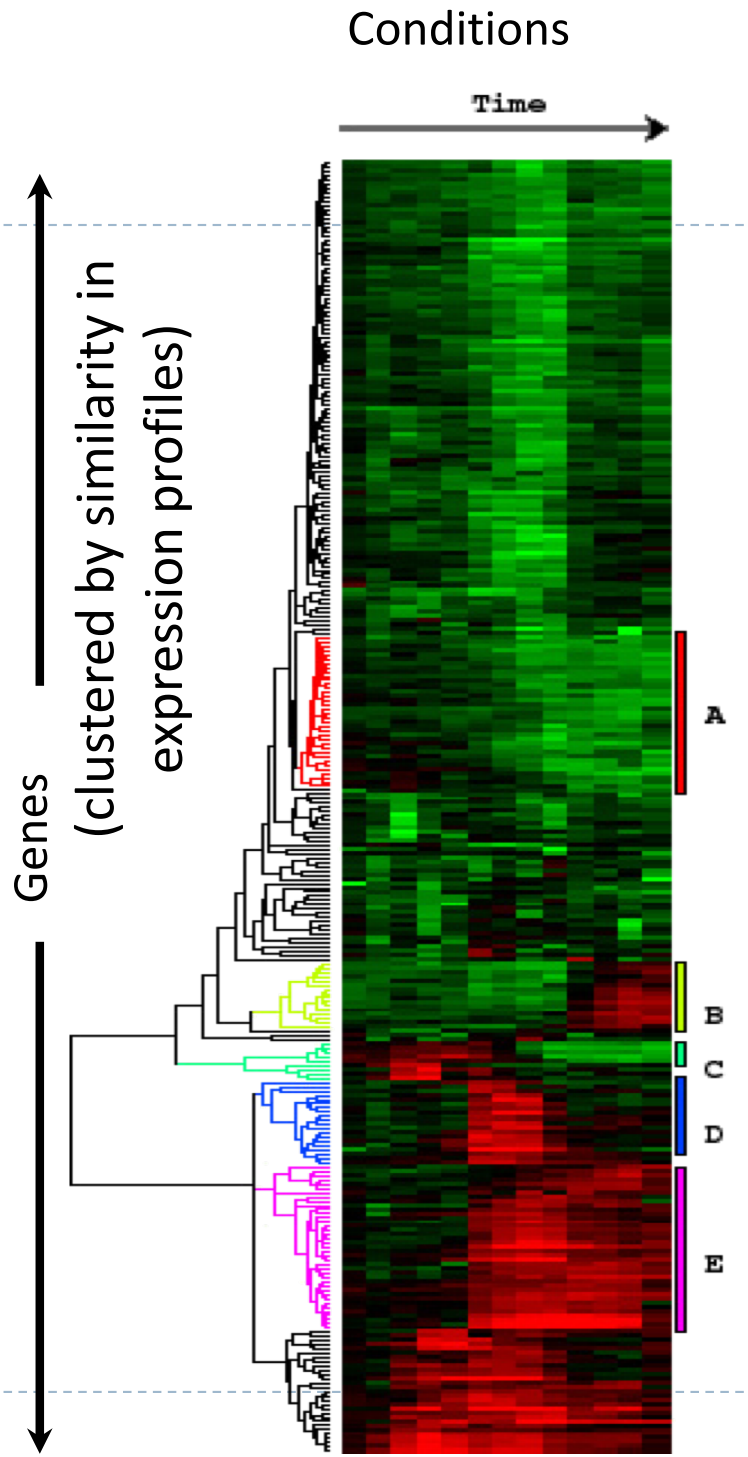


(average of all distances used)

End Result

Genes are grouped according to similarities in their expression levels across a variety of conditions.

- Place genes with similar expression profiles into clusters.
- Similarity is defined by Pearson correlation.



K-means: The Algorithm

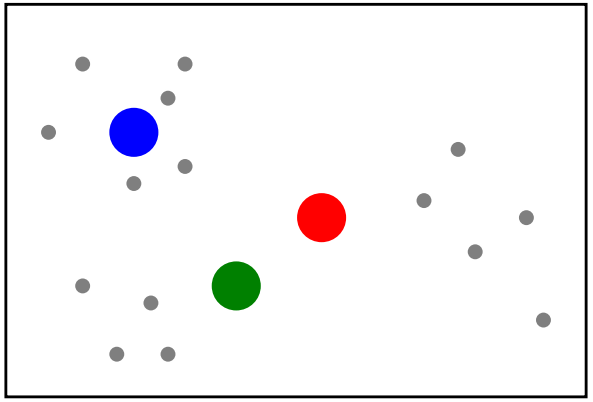
- Given a set of numeric points in *d* dimensional space, and integer *k*
- Algorithm generates *k* (or fewer) clusters as follows:

Assign all points to a cluster at random.

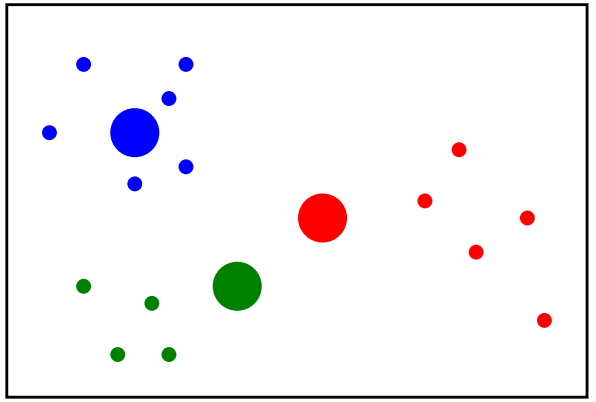
Repeat until stable:

- Compute centroid for each cluster
- Reassign each point to nearest centroid

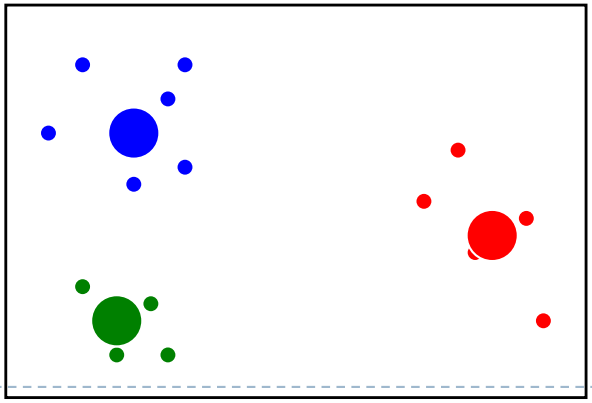
K-means: Example, k = 3



Step 1: Make random assignments and compute centroids (big dots)

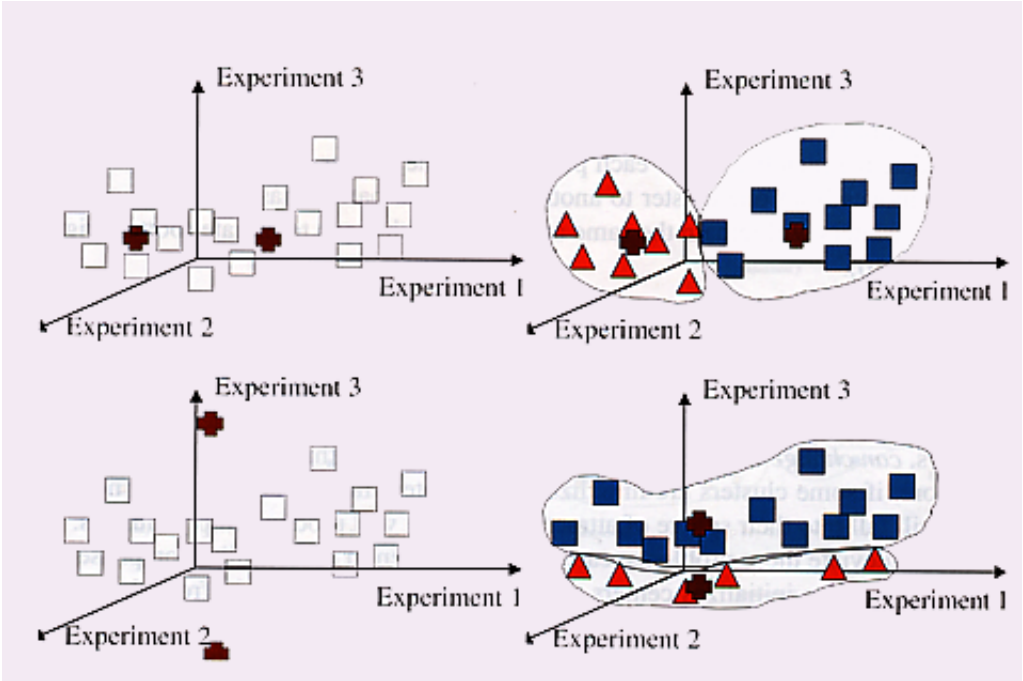


Step 2: Assign points to nearest centroids



Step 3: Re-compute centroids (in this example, solution is now stable)

K-means weaknesses: can give you a different result each time with exactly the same data



K-means: Summary

- Must choose parameter *k* in advance, or try many values.
- Data must be numerical and must be compared via Euclidean distance (there is a variant called the *k*-medians algorithm to address these concerns)
- The algorithm works best on data which contains spherical clusters; clusters with other geometry may not be found.
- The algorithm is sensitive to *outliers*---points which do not belong in any cluster. These can distort the centroid positions and ruin the clustering.

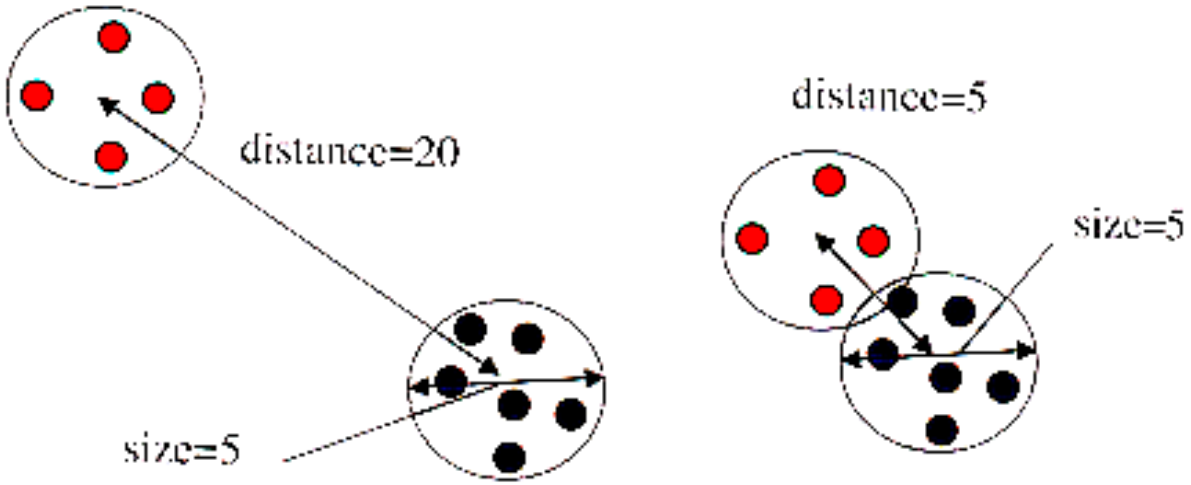
Clustering has no one answer

- Given a collection of objects, put objects into groups based on similarity.

- It really depends on how you measure similarity/dissimilarity

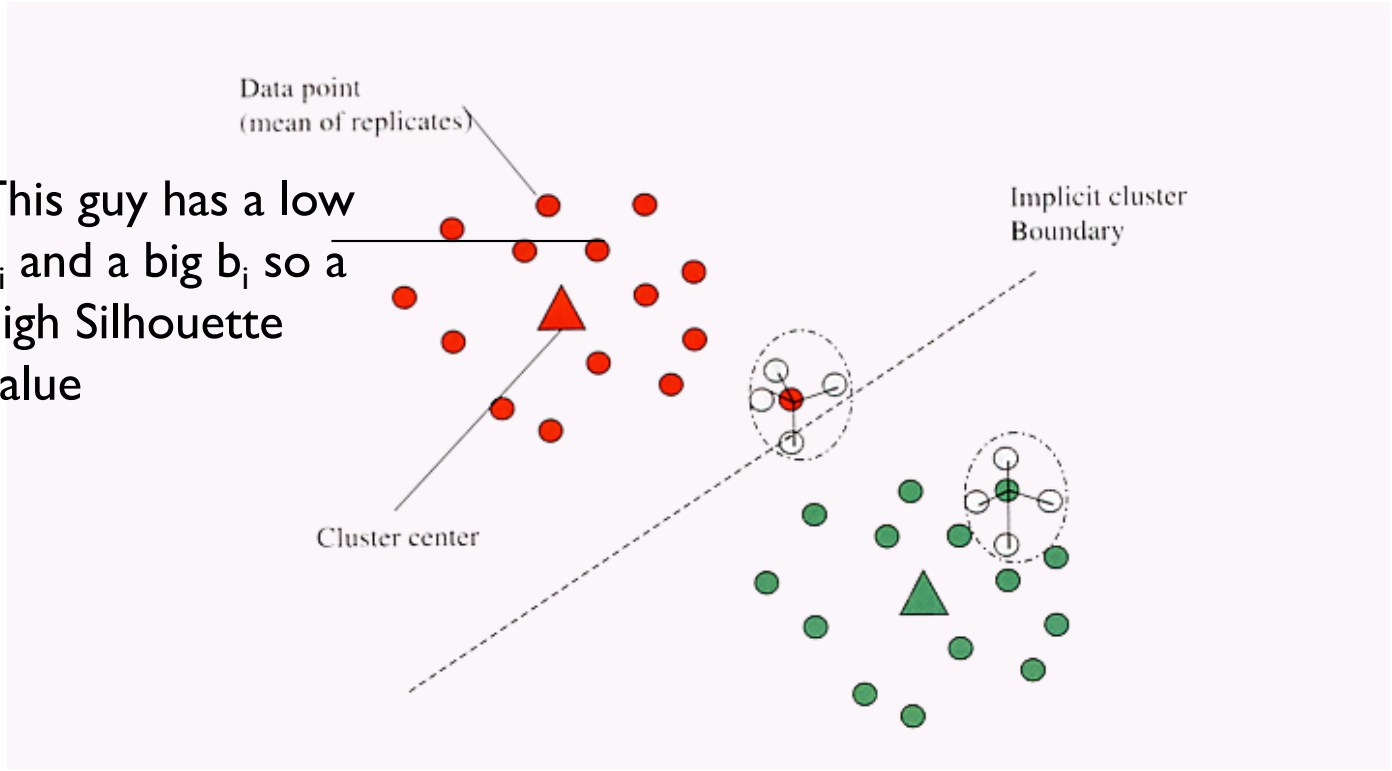
Judging the Quality of a Cluster

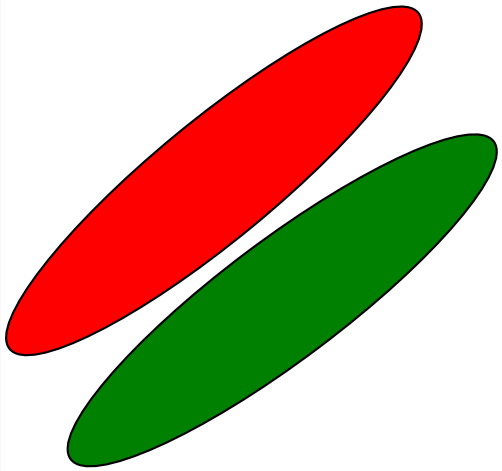
The idea is to classify distinct groups: other methods seek to directly optimize this trait in classification



Measuring the Quality of Clusters

This guy has a low a_i and a big b_i so a high Silhouette value





Will lead to low or negative Silhouette values

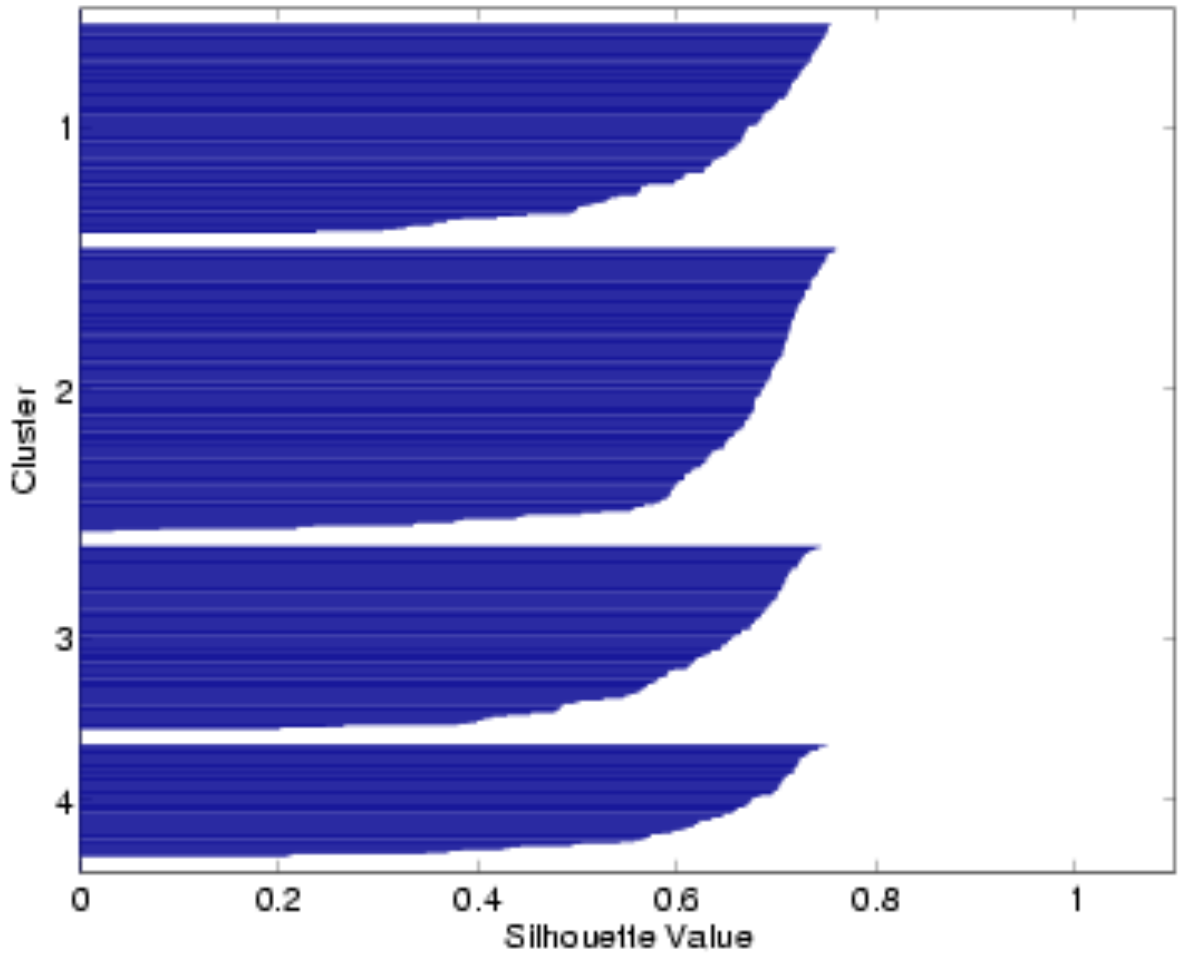
Silhouette Width

$Sil_i = (b_i - a_i)/max(a_i, b_i)$

a_i-average within cluster distance with respect to gene i

b_i-average between cluster distance with respect to gene i

Silhouette Plot



Clustering the Breast Cancer dataset

- What is the best way to cluster the 15k patients dataset?