Transcriptome
Lecture 3

Outline

- Multiple Testing Procedures
- Data Visualization, Distance Measures
- Clustering
- Gene Annotation and Enrichment Analysis

The Visualization

- MA plot
- Volcano plot
- Heatmap
- Dendrogram

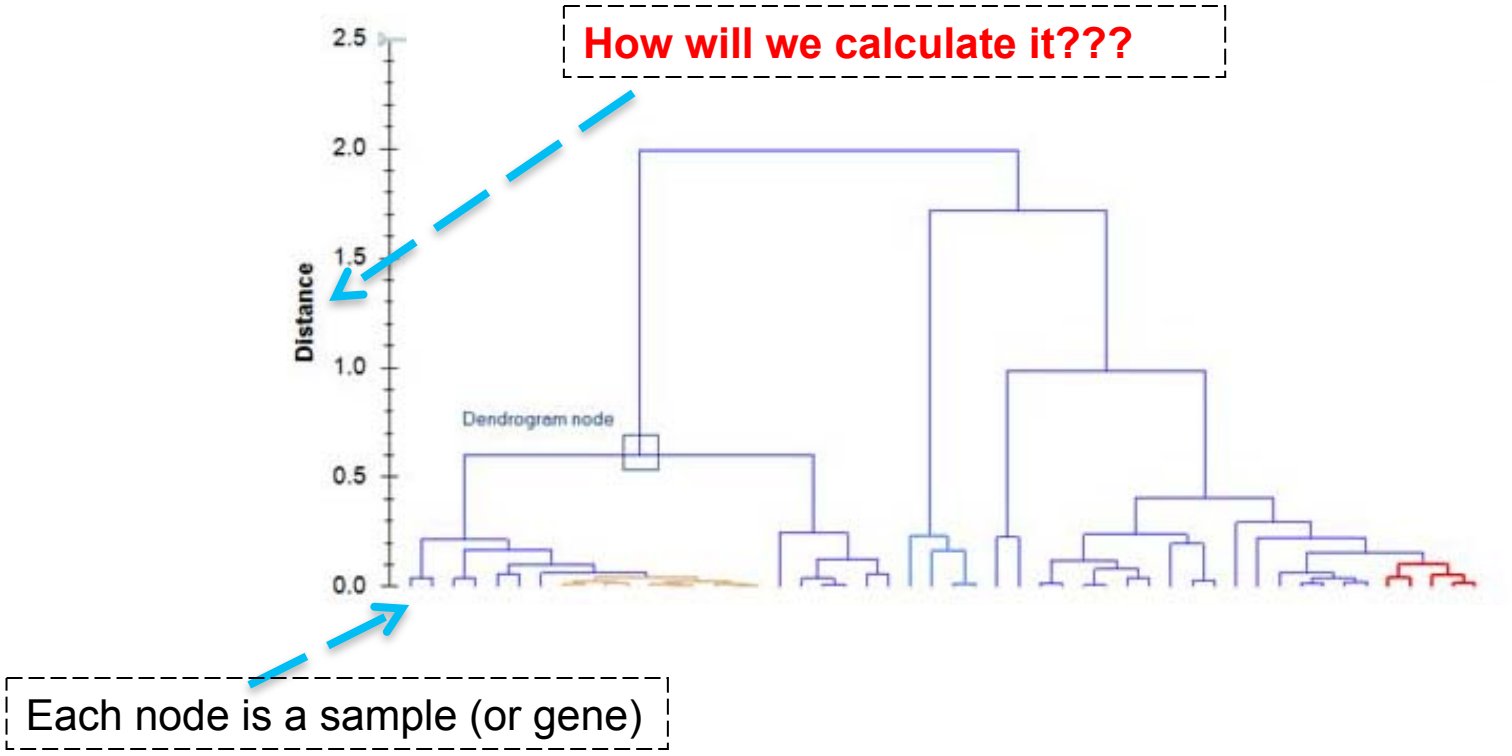
Dendrogram

- A **Dendrogram** is a tree diagram frequently used to illustrate the arrangement of the clusters produced by clustering.

- In Microarray/RNA-seq, it can represent the distance between a number of samples (or genes)

Dendrogram

- In Microarray/RNA-seq, a dendrogram can represent the distance between a number of samples (or genes)



Distance

- Distance between points
  - Minkowski metric
    - Euclidean metric
    - Manhattan metric
  - Correlation distance
    - Pearson sample correlation distance
    - Cosine correlation distance
    - Spearman sample correlation distance

Minkowski distance

For two points $X = (x_1,\ldots,x_m)$ and $Y=(y_1,\ldots,y_m)$

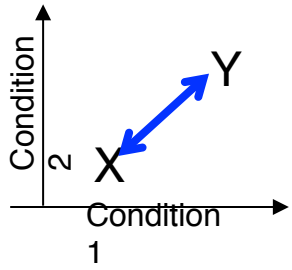
- Minkowski distance

$$F(z_1,\ldots,z_m) = \left(\sum_{k=1}^{m} z_k^{\lambda}\right)^{1/\lambda}$$

$$z_k = d_k(x_k, y_k) = |x_k - y_k|$$

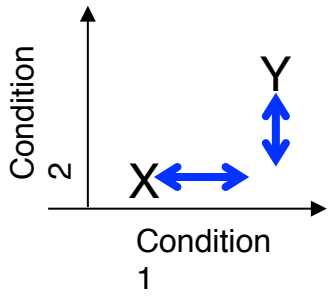
  - EUC Euclidean distance $\lambda = 2$

$$d_{euc}(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2}.$$



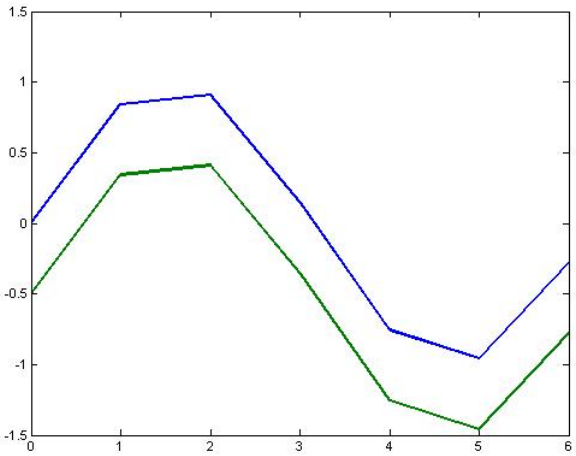
  - Man Manhattan distance $\lambda = 1$

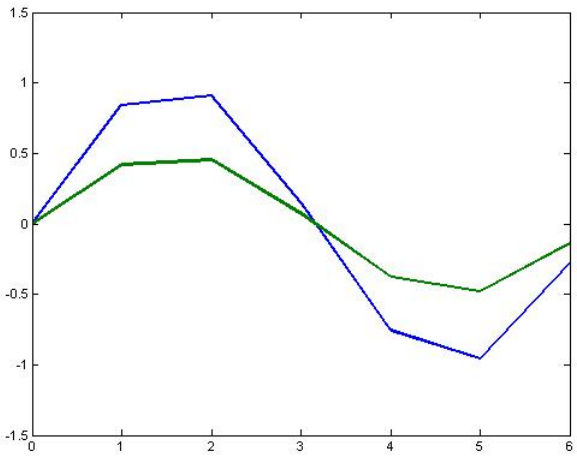
$$d_{man}(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m}|x_i - y_i|.$$



Distance

- In most case, we care more about the overall shape of expression profiles rather than the actual magnitudes
- That is, we might want to consider genes similar when they are “up” and “down” together





Distance: correlations

- Pearson correlation coefficient

$$r(\mathbf{x},\mathbf{y}) = \frac{\sum_{i=1}^{m}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i-\overline{x})^2\sum_{i=1}^{m}(y_i-\overline{y})^2}}.$$

$$r(x,y) \in [-1,1]$$

- Pearson's correlation reflects the degree of linear relationship between X and Y.

- 1 – perfect similarity(positive linear)

  0 – no similarity

  -1 – perfect dissimilarity(negative linear)

Gene Co-expression correlation

	T1	T2	T3	T4	T5
A	2.5	2.8	3.7	4.6	1.5
B	0.2	0.8	0.3	1.5	0.6
C	1.9	1.3	0.2	0.8	1.6
D	0.8	1.4	0.7	1.6	1.7
E	1.5	1.8	0.3	0.5	1.9

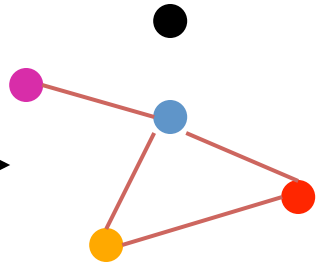
pair-wise

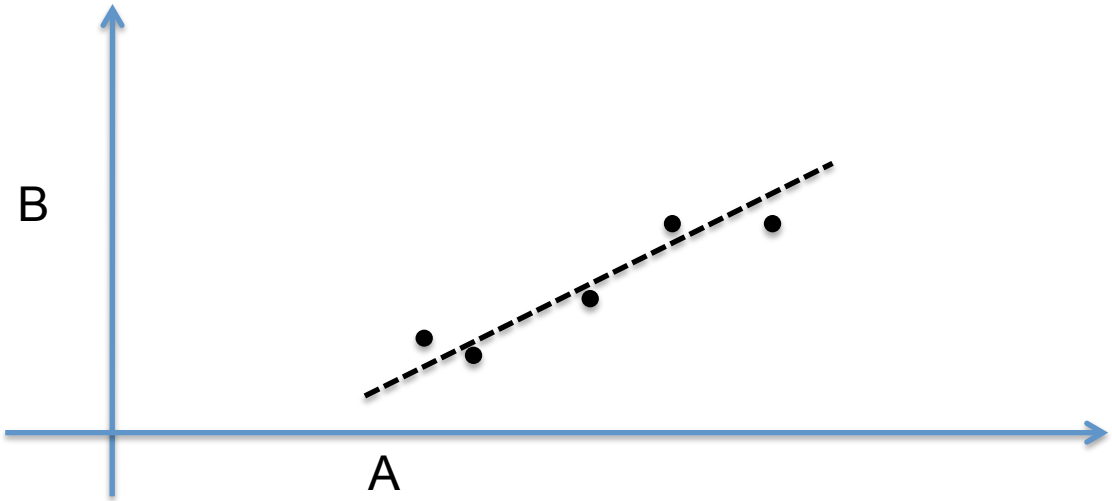
correlation

	C.C
A-B	0.76
A-C	0.90
A-D	0.50
…	0.83
D-E	0.42

cutoff

>= 0.6





Distance: correlations

- Correlation-based distance:
  - COR Pearson sample correlation distance

$$d_{cor}(\mathbf{x},\mathbf{y}) = 1 - r(\mathbf{x},\mathbf{y}) = 1 - \frac{\sum_{i=1}^{m}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{m}(x_i-\overline{x})^2\sum_{i=1}^{m}(y_i-\overline{y})^2}}.$$

  - EISEN Cosine correlation distance

$$d_{eisen}(\mathbf{x},\mathbf{y}) = 1 - \frac{\mathbf{x}'\mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{\sum_{i=1}^{m} x_i y_i}{\sqrt{\sum_{i=1}^{m} x_i^2 \sum_{i=1}^{m} y_i^2}}$$

    - Special case of Pearson sample correlation with $\overline{x}$ and $\overline{y}$ are both zero
    - This method only works with centered data, i.e., data which have been shifted by the sample mean

Correlation Coefficient in R

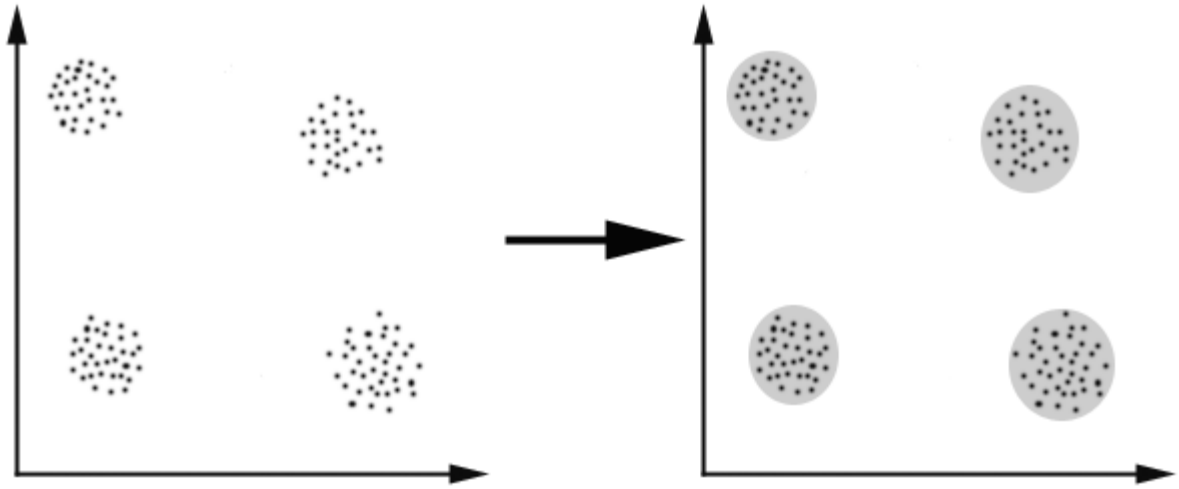
- cor.test(x,y, method="spearman")
- "pearson", "kendall", "spearman”

```
> d.exp=exprs(Dilution)
> cor.test(d.exp[1,],d.exp[2,], methods=”pearson")

   Pearson's product-moment correlation

data:  d.exp[1, ] and d.exp[2, ]
t = 2.8045, df = 2, p-value = 0.1071
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.4807171  0.9977571
sample estimates:
      cor
0.8928993
```

Outline

- Multiple Testing Procedures
- Data Visualization, Distance Measures
- Clustering
- Gene Annotation and Enrichment Analysis

Clustering: what is it?

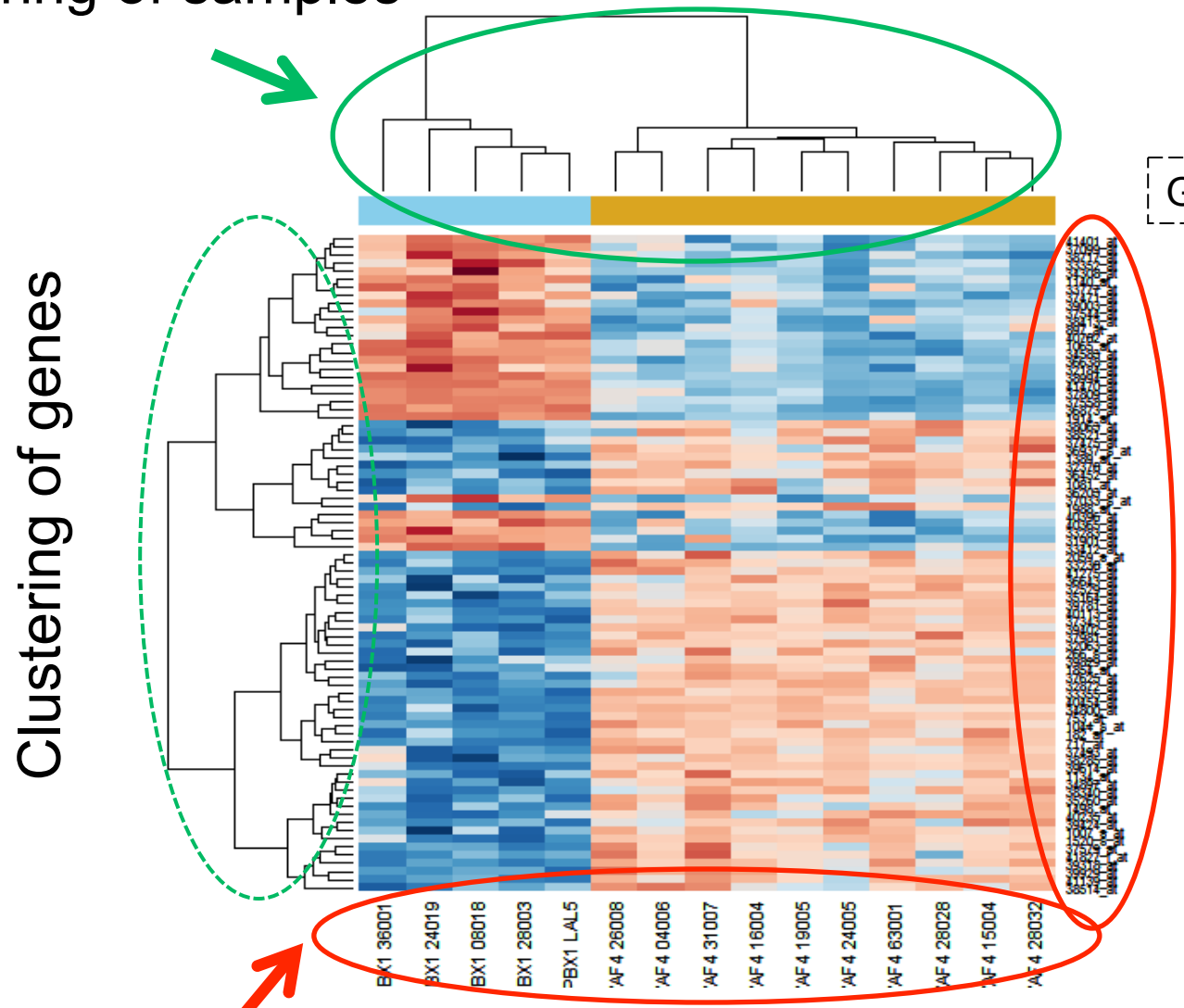


The goal of clustering could be to gather genes or samples into groups. We call those groups as clusters.

A *cluster* is a collection of objects which are “similar” between them and are “dissimilar” to the objects belonging to other clusters.

Clustering

Clustering of samples



Brown = Higher expression
Blue = Lower Expression

Clustering: why cluster genes?

- Identify groups of possibly co-regulated genes (e.g. genes regulated by one transcription factor).
- Identify typical temporal or spatial gene expression patterns (e.g. cell cycle data).
- Arrange a set of genes in a linear order that is at least not totally meaningless (for visualization).

Clustering: why cluster samples?

- Quality control: Detect experimental artifacts/ bad hybridizations
- Check whether samples are grouped according to known categories
- Identify new classes of biological samples (e.g. tumor subtypes)

Clustering: Basic principles

- Issues to be consider before performing a cluster analysis
  - ❑Which genes/arrays to be used?
  - ❑Which distance (similarity) measures?
  - ❑Which method is used to join clusters/ observations?
  - ❑Which clustering algorithm is applied?

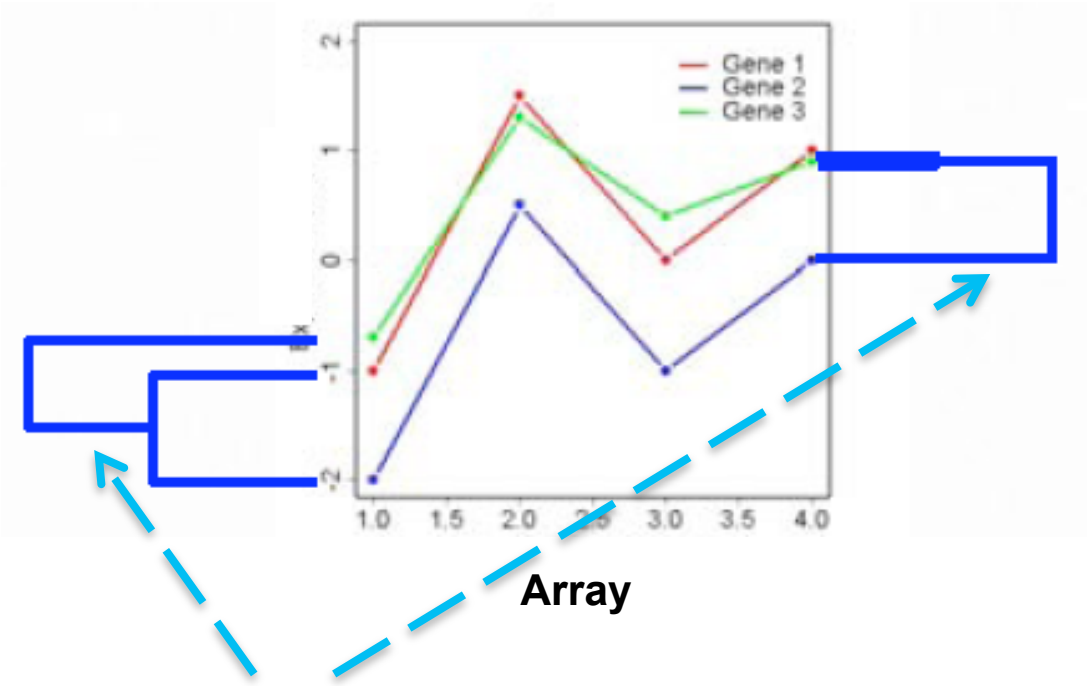
Clustering: Basic principles

- Issues to be consider before performing a cluster analysis
  - ❑ Which genes/arrays to be used?
    - It is **advisable to reduce the number of genes** from the full set to some more manageable number, before clustering.
    - A common approach is to perform a cluster analysis based on differentially expressed genes
  - ❑ Which distance (similarity or dissimilarity) measures?
  - ❑ Which method is used to link clusters?
  - ❑ Which clustering algorithm is applied?

Clustering: Basic principles

- Issues to be consider before performing a cluster analysis
  - ❑Which genes/arrays to be used?
  - ❑Which distance (similarity or dissimilarity) measures?
    - Correlation coefficient based distance (scale-independent)
    - Minkowski metric (scale-dependent)
  - ❑Which method is used to link clusters?
  - ❑Which clustering algorithm is applied?

Euclidean vs. Correlation



Are these clustering based on euclidean distance or correlation coeffecient?

Clustering: Basic principles

- Issues to be consider before performing a cluster analysis
  - ❑Which genes/arrays to be use?
  - ❑Which distance (similarity or dissimilarity) measures?
  - ❑Which method to use to link clusters?
    - ❑How to compute the cluster similarity in order to link them?
  - ❑Which clustering algorithm?

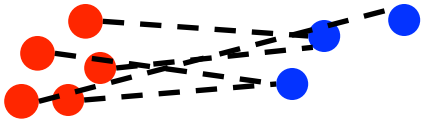
Clustering: Cluster similarity

- Four major methods to compute group similarity:
  - Given two clusters c1 and c2
    - Single-link: s(g1,g2)= similarity of the **closest pair** of points between the two clusters
    - Complete-link: s(g1,g2)= similarity of the **furtherest pair** of points between the two clusters
    - Average-link: s(g1,g2)= **average** of similarity of all pairs of points between the two clusters
    - Centroid-link: s(g1,g2)= distance between **centroids** of the two clusters

Clustering: cluster similarity

Single-link: similarity of the closest pair of points between the two clusters

Complete-link: similarity of the furtherest pair of points between the two clusters



Average-link: average of similarity of all pairs of points between the two clusters

Centroid-link: distance between centroids of the two clusters

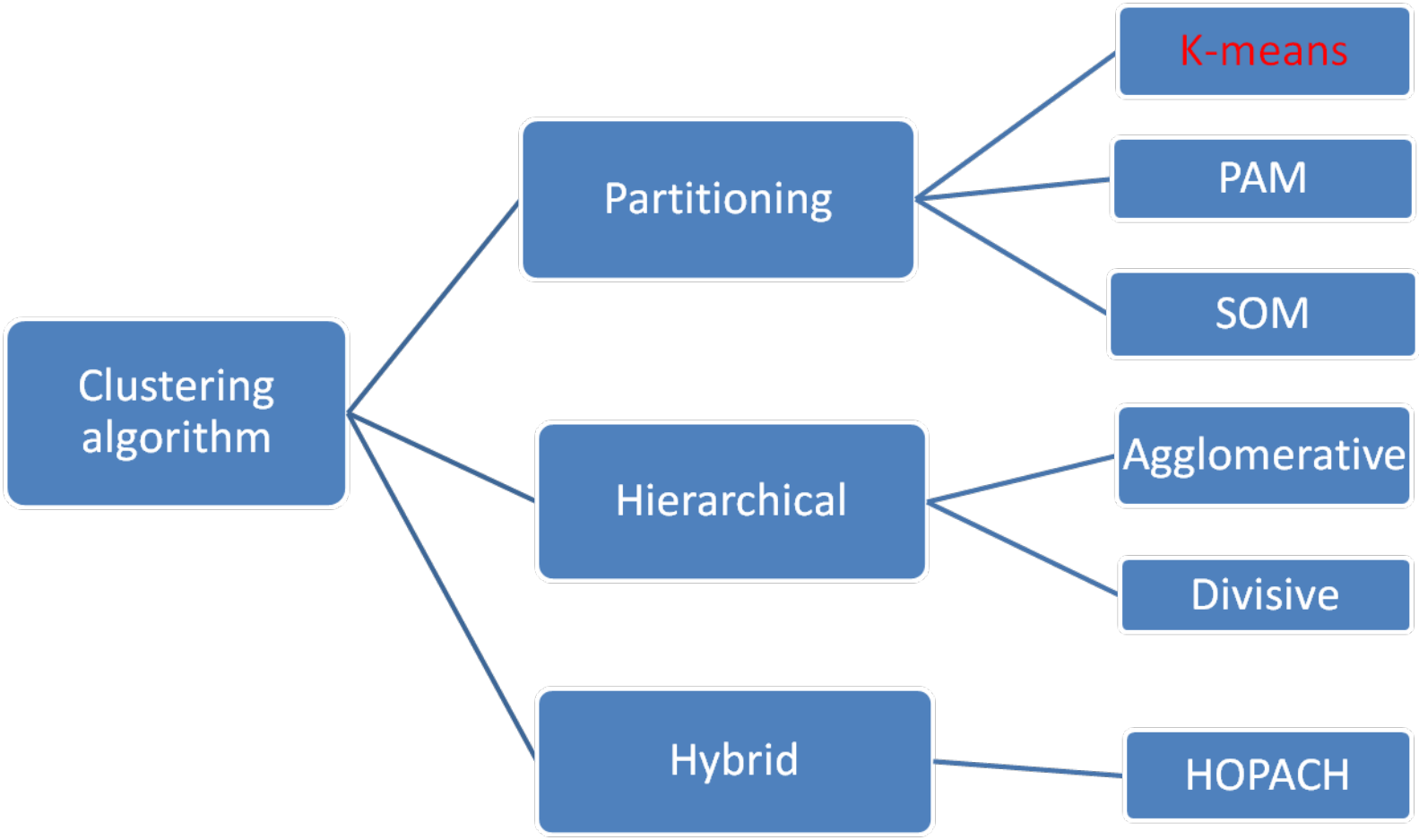
Clustering: Cluster similarity

- A comparison of cluster linkage methods:
  - Single-link and complete link: individual decision, more sensitive to outliers
  - Average-link and centroid-link: group decision, less sensitive to outliers

Clustering: Basic principles

- Issues to be consider before performing a cluster analysis
  - ❑Which genes/arrays to be use?
  - ❑Which distance (similarity or dissimilarity) measures?
  - ❑Which method to use to link clusters?
  - ❑Which clustering algorithm?

Type of Clustering algorithm



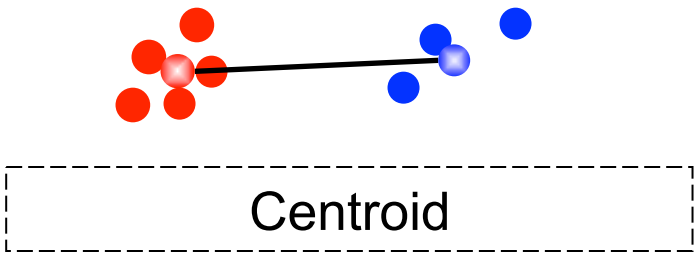
Partitioning: K-means

- A partitioning algorithm with a prefixed number k of clusters, that tries to minimize the sum of within-cluster-variances

$$\min\left(\sum_{i=1}^{k}\sum_{x_j\in S_i}\left\|x_j-\mu_i\right\|^2\right)$$

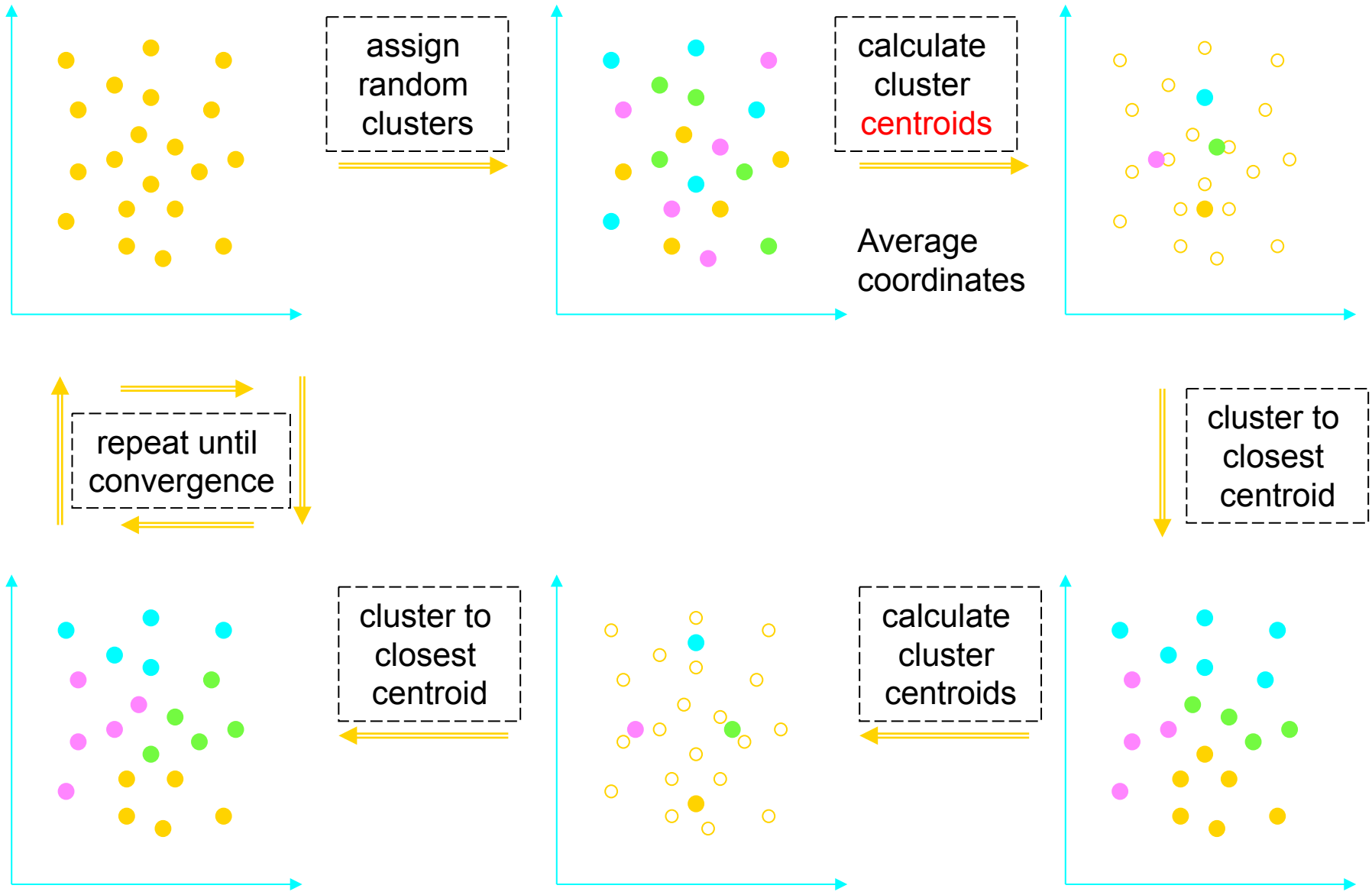
- Algorithm
  1. Randomly choose K points as the center of the K clusters
  2. Visit each point to its closest cluster
  3. Update the center of each newly formed cluster
  4. Repeat steps 2-4 until there is no change to the centers (centroids)(or reach the maximum cycles)

Clustering: cluster similarity



$$1/n \sum_{x_j \in S_i} x_j$$

Partitioning: K-means



Partitioning: K-means

- A partitioning algorithm with a prefixed number k of clusters, that tries to minimize the sum of within-cluster-variances
- MUST choose number of clusters K as a priori
  - If K =2, the data will be clustered (partitioned) into two clusters…
  - If K =4, the data will be clustered (partitioned) into two clusters..
  - …

Partitioning: K-means

- Use “cclust” package in R
- http://cran.r-project.org/web/packages/cclust/index.html

```
> source("http://bioconductor.org/biocLite.R")
> biocLite("cclust")

> library(cclust)
> ALL_exp=exprs(ALL)
> kc=cclust(ALL_exp,10,200,verbose=TRUE,
   dist="euclidean", method="kmean")
```

Distance: euclidean or manhattan

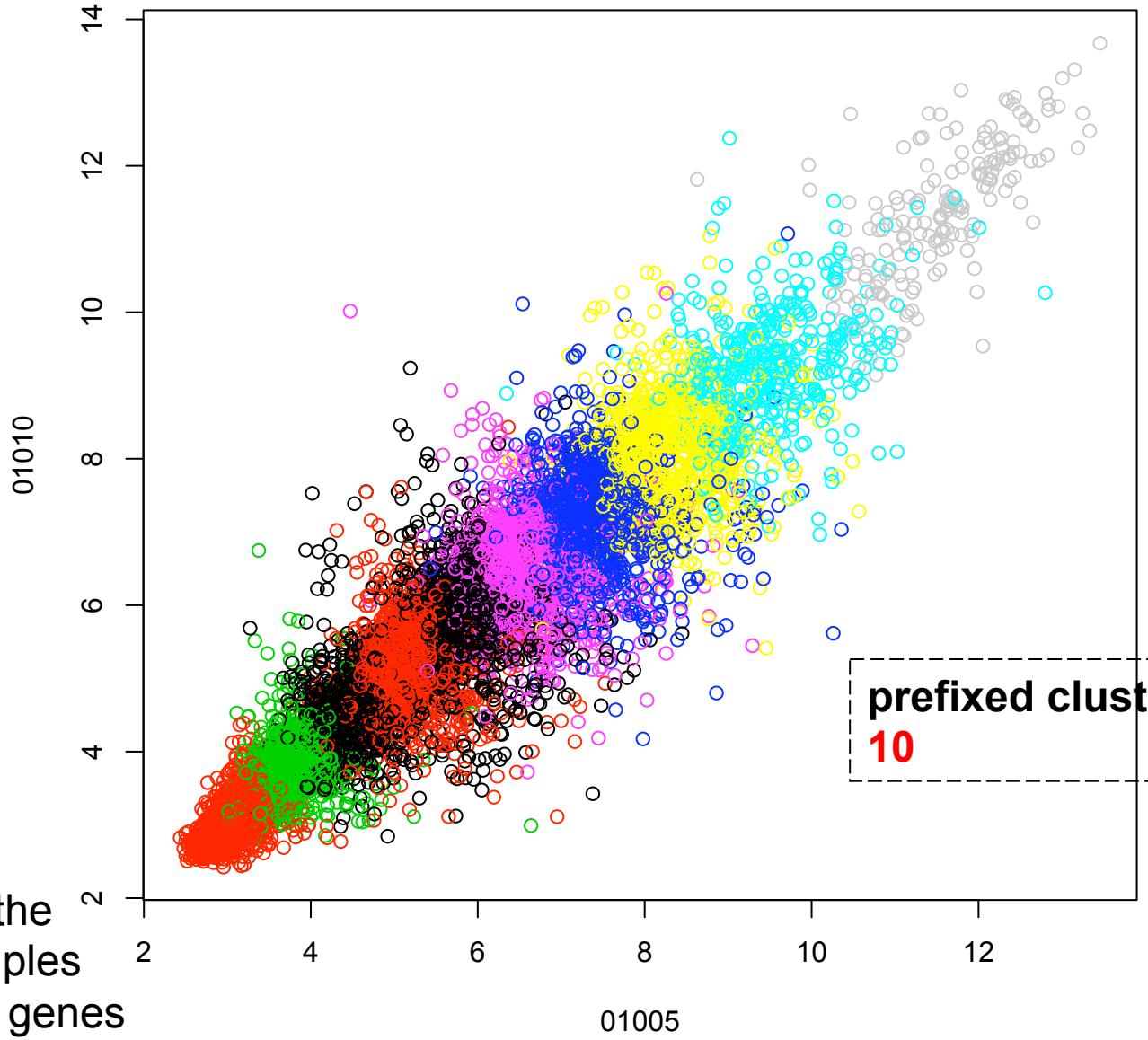
Partitioning: K-means

- Use "cclust" package in R

```
> kc=cclust(d,10,200,verbose=TRUE, dist="euclidean", method="kmean")
Iteration:   1    Changes:        10559
Iteration:   2    Changes:         2293
Iteration:   3    Changes:          929
Iteration:   4    Changes:          795
Iteration:   5    Changes:          789
Iteration:   6    Changes:          890
Iteration:   7    Changes:          871
Iteration:   8    Changes:          747
Iteration:   9    Changes:          645
Iteration:  10    Changes:          564
……..
Iteration: 117    Changes:           14
Iteration: 118    Changes:           11
Iteration: 119    Changes:            5
Iteration: 120    Changes:            6
Iteration: 121    Changes:            8
Iteration: 122    Changes:            5
Iteration: 123    Changes:            5
Iteration: 124    Changes:            6
Iteration: 125    Changes:            3
Iteration: 126    Changes:            3
Iteration: 127    Changes:            5
Iteration: 128    Changes:            3
Iteration: 129    Changes:            0
```

Partitioning: K-means

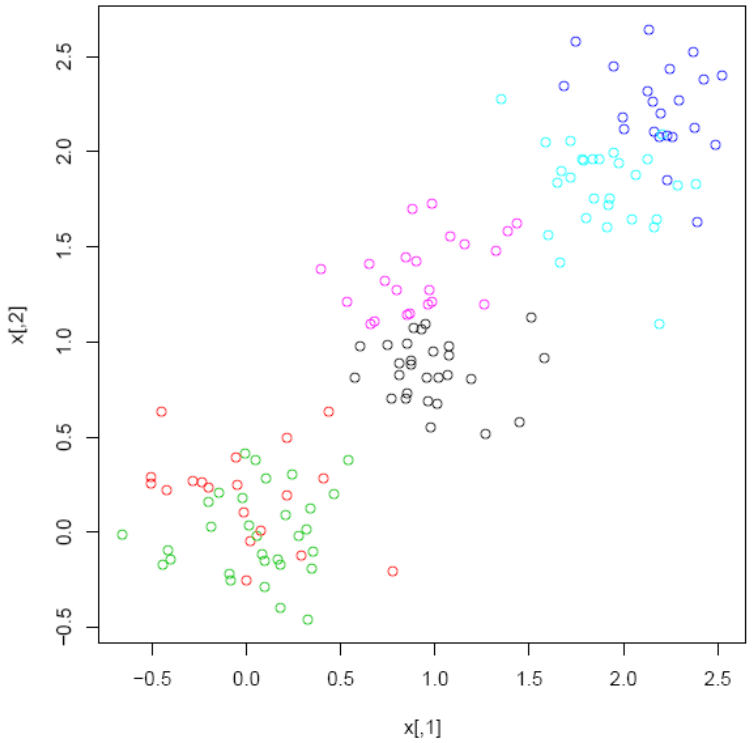
> plot(ALL_exp, col=kc$cluster)



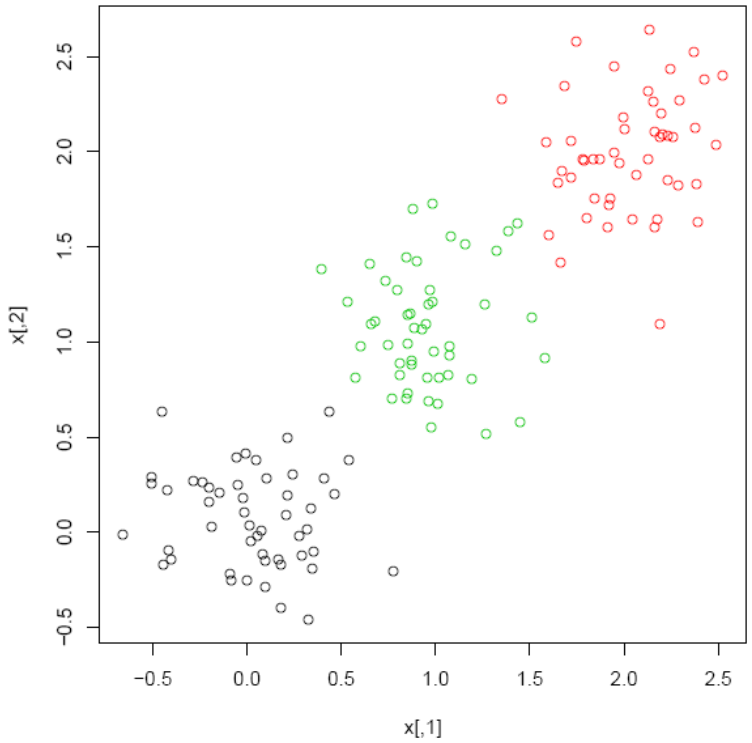
prefixed cluster number are 10

x,y-axis are the first two samples in ALL for all genes

Partitioning: K-means

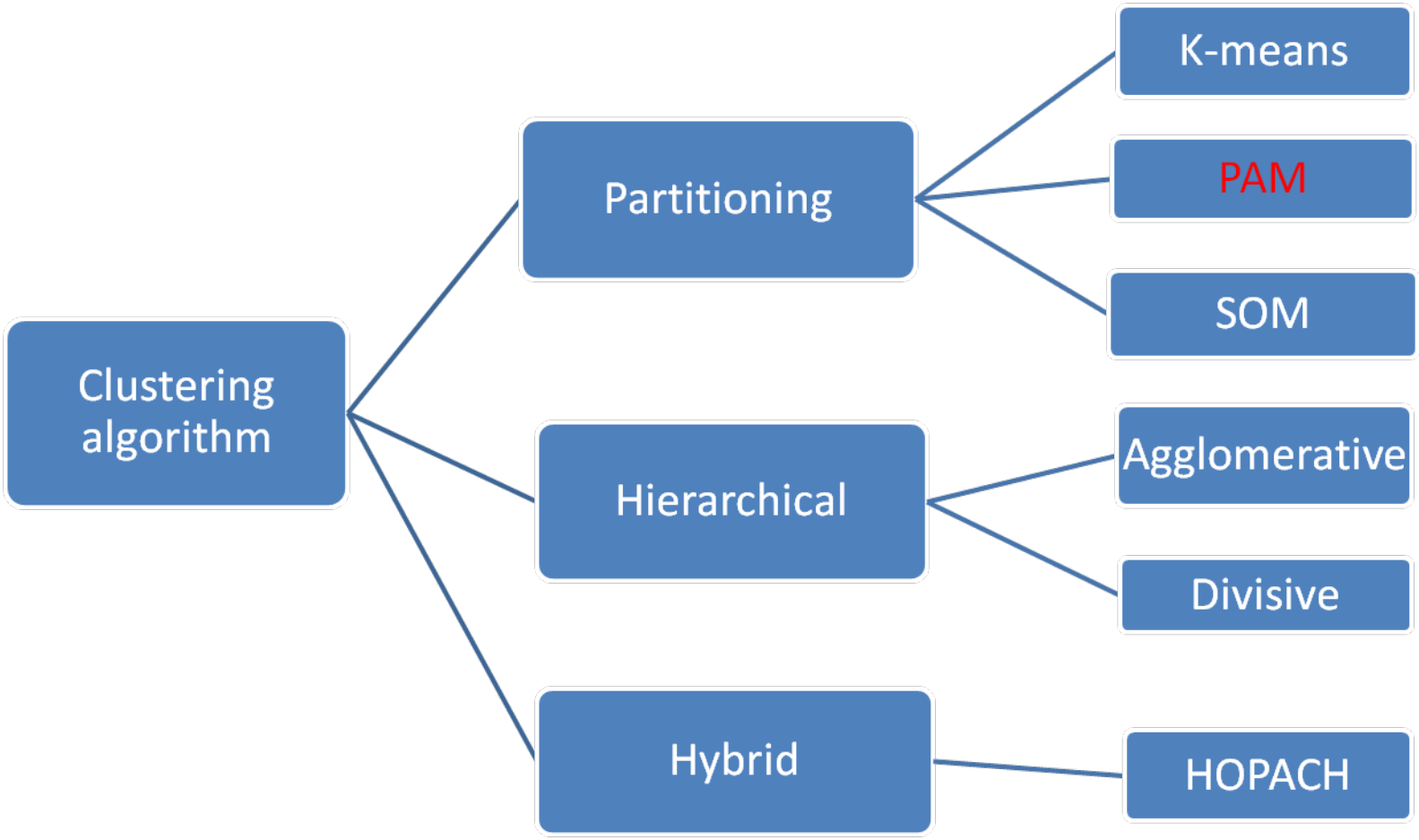


prefixed cluster number are 6



prefixed cluster number are 3

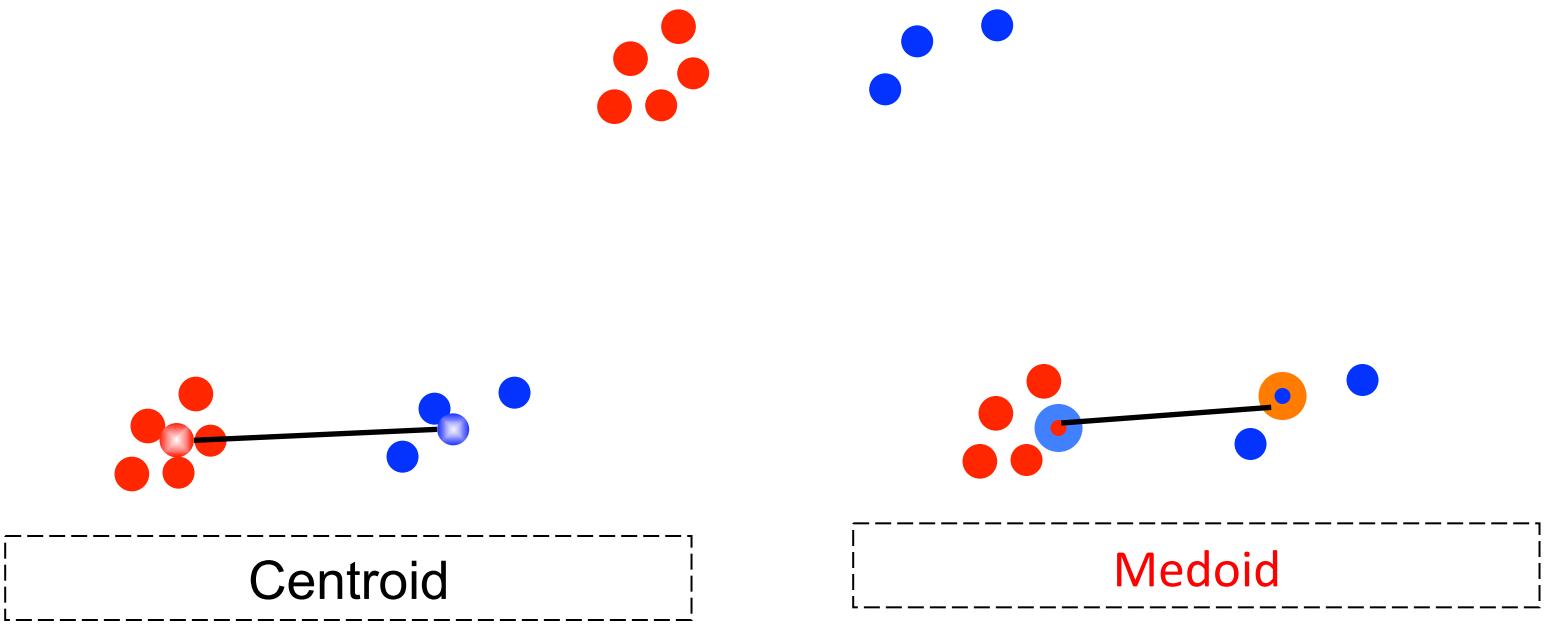
Type of Clustering algorithm



Partitioning: PAM

- PAM: Partitioning around medoids or k-medoids
- Mediod is the "representative point" within a cluster
  - It is different from "centroid" used by k-means, which is the average of the samples within a cluster
  - For example, medoid can be a point which has the smallest sum distance to all other points within the cluster
- The iterative procedure is analogous the one in *K*-means clustering

Clustering: cluster similarity



Partitioning: PAM

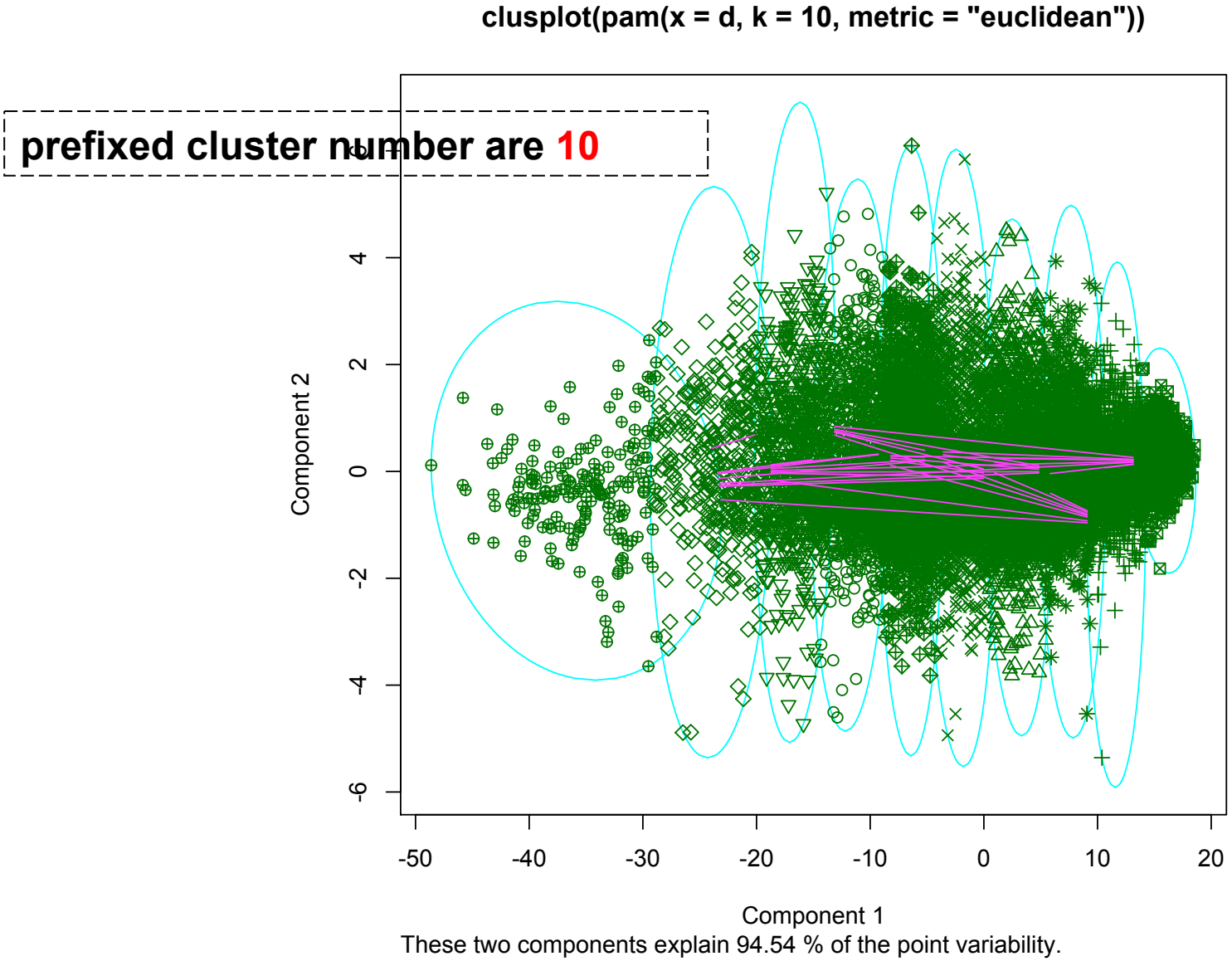
- Use “cluster” package in R
- http://cran.r-project.org/web/packages/cluster/index.html

```
> source("http://bioconductor.org/biocLite.R")
> biocLite(“cluster")

> library(cluster)
> ALL_exp=exprs(ALL)
> kc=pam(ALL_exp,10,metric = "euclidean”)
```

Distance (metric) : euclidean or manhattan

Partitioning: PAM

> plot(pamc)



Partitioning: PAM

- **pamx=pam(x,2)**
- **plot(pamx)**

K=-2, use two randomly selected observations as starting medoids

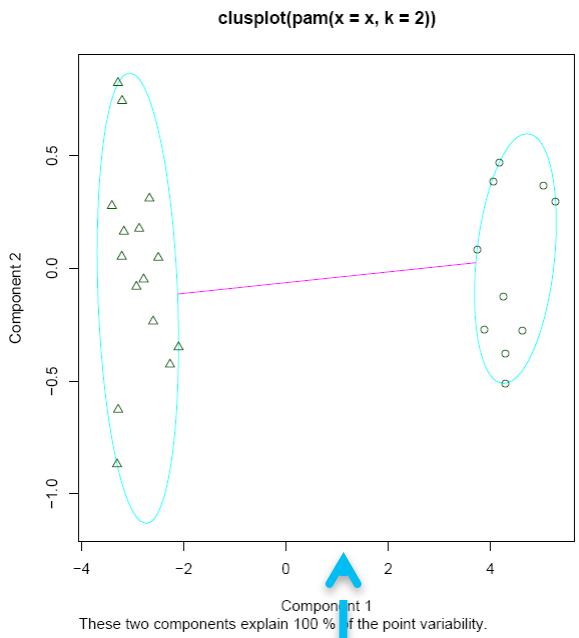
- **p2m=pam(x,2,medoids=c(1,16))**
- **plot(p2m)**

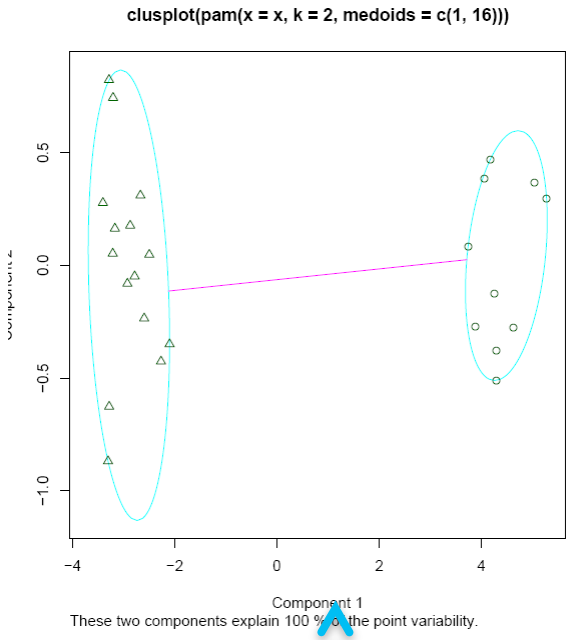
K=2, use observations 1 and 16 as starting medoids

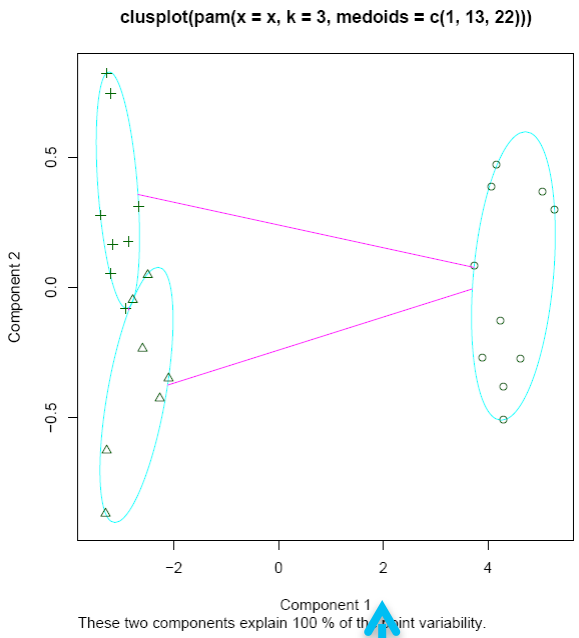
- **p3m=pam(x,3,medoids=c(1,13,22))**
- **plot(p3m)**

K=3, Use observations 1,13,22 as starting medoids

Partitioning: PAM





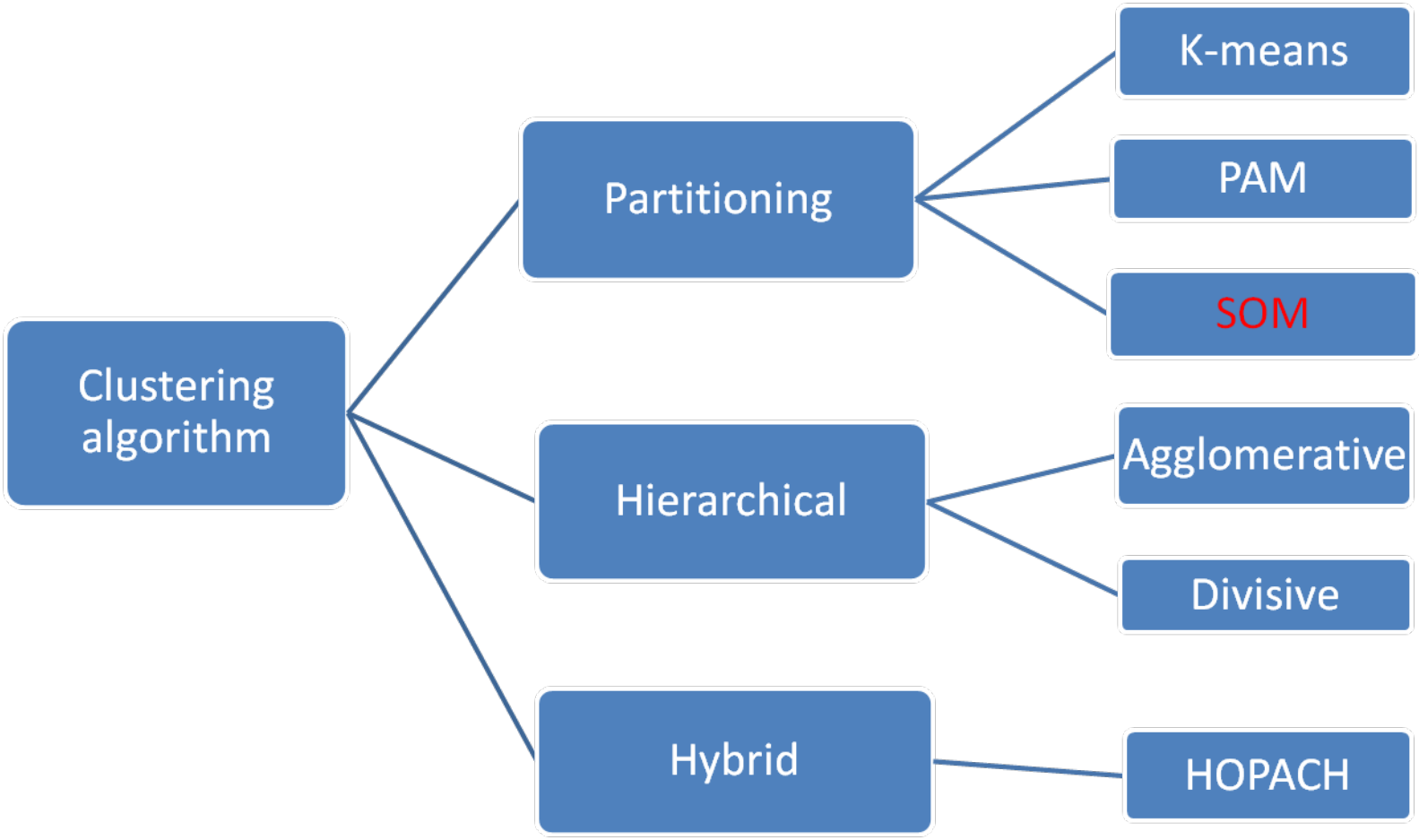


K=-2, use two randomly selected observations as starting medoids

K=2, use observations 1 and 16 as starting medoids

K=3, Use observations 1,13,22 as starting medoids

Type of Clustering algorithm



Partitioning: SOM

- SOM – Self-Organizing Map
- Based on work of Kohonen on learning/memory in the human brain
- Like *k*-means, we specify the number of clusters
- However, we also specify a topology – a 2D grid that gives the geometric relationships between the clusters (i.e., which clusters should be near or distant from each other)
- The algorithm learns a mapping from the high dimensional space of the data points onto the points of the 2D grid (there is one grid point for each cluster)

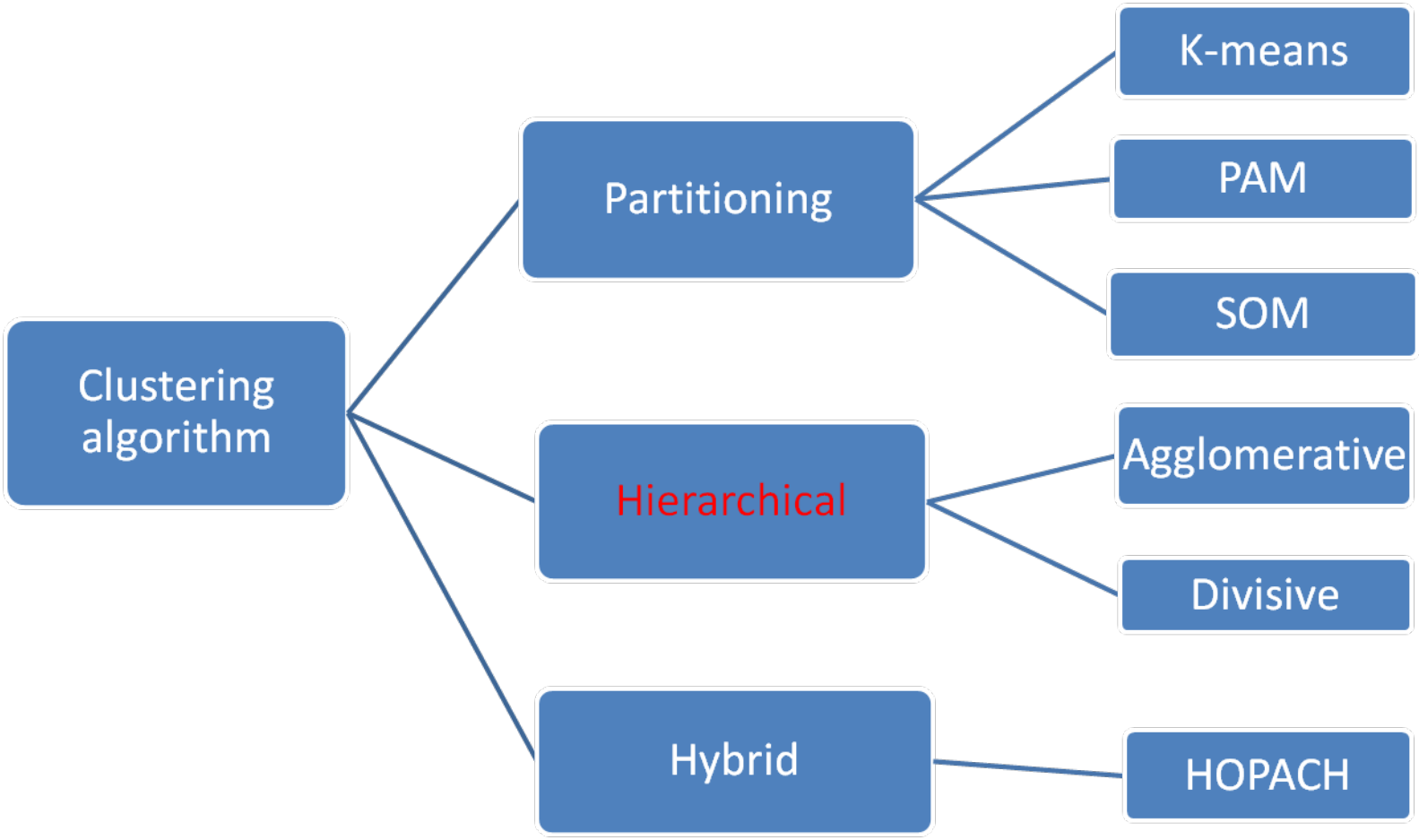
Partitioning: SOM

- The algorithm is complicated and there are a lot of parameters (such as the "learning rate") - these settings will affect the results

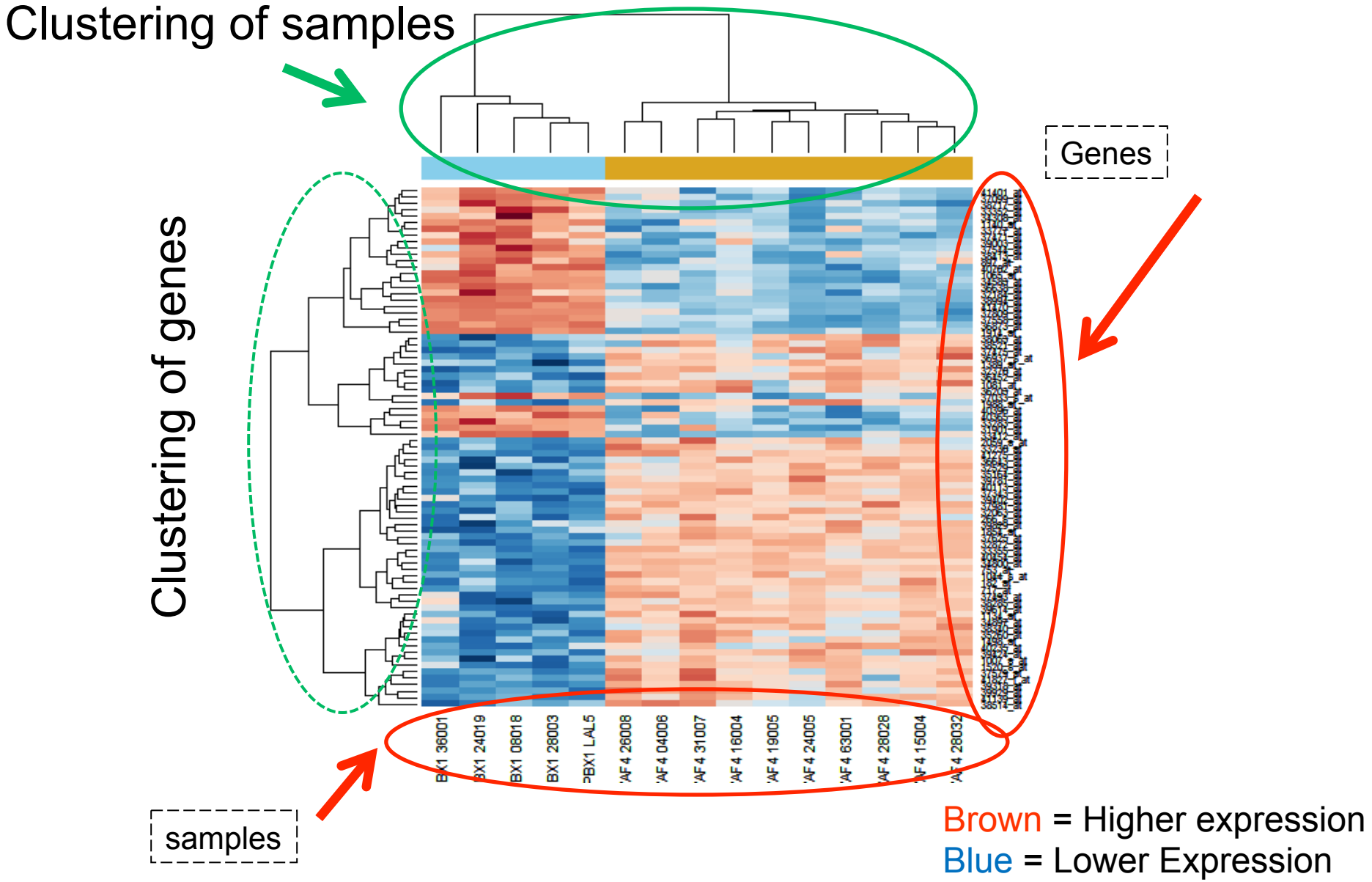
- The idea of a topology in high dimensional gene expression spaces is not exactly obvious
  - How do we know what topologies are appropriate?
  - In practice people often choose nearly square grids for no particularly good reason

- As with *k*-means, we still have to worry about how many clusters to specify…
  - Other choices?

Type of Clustering algorithm



Dendrogram: Hierarchical Clustering



Hierarchical clustering

- Advantage: no need to specify the number of clusters in advance.

- Similarity of objects is represented in a tree structure(dendrogram)

- Two types of hierarchical clustering:
  - **Agglomerative (bottom up)**
  - **Divisive(top down)(methods are less common)**

- Functions in R for hierarchical clustering : **agnes diana**

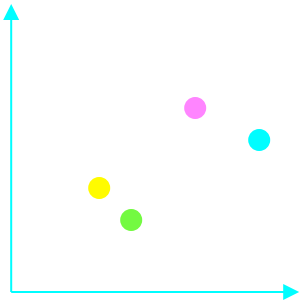
Agglomerative Hierarchical clustering

- Start with each object as an individual cluster.

- In each iteration, merge the two clusters with the minimal distance from each other - until you are left with a single cluster comprising all objects.

- Calculation of the distance between two clusters is based on the pairwise distances between members of the clusters:
  - Complete linkage: largest distance
  - Average linkage or centroid linkage: average distance
  - Single linkage: smallest distance

Agglomerative Hierarchical clustering



calculate distance matrix

	gene 1	gene 2	gene 3	gene 4
gene 1	0			
gene 2	2	0		
gene 3	8	7	0	
gene 4	10	12	4	0

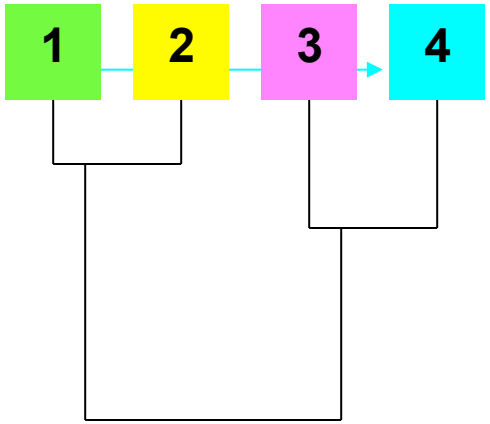
calculate averages of most similar

	gene 1,2	gene 3	gene 4
gene 1,2	0		
gene 3	7.5	0	
gene 4	11	4	0

calculate averages of most similar

	gene 1,2	gene 3,4
gene 1,2	0	
gene 3,4	9.25	0

Dendrogram



Hierarchical clustering

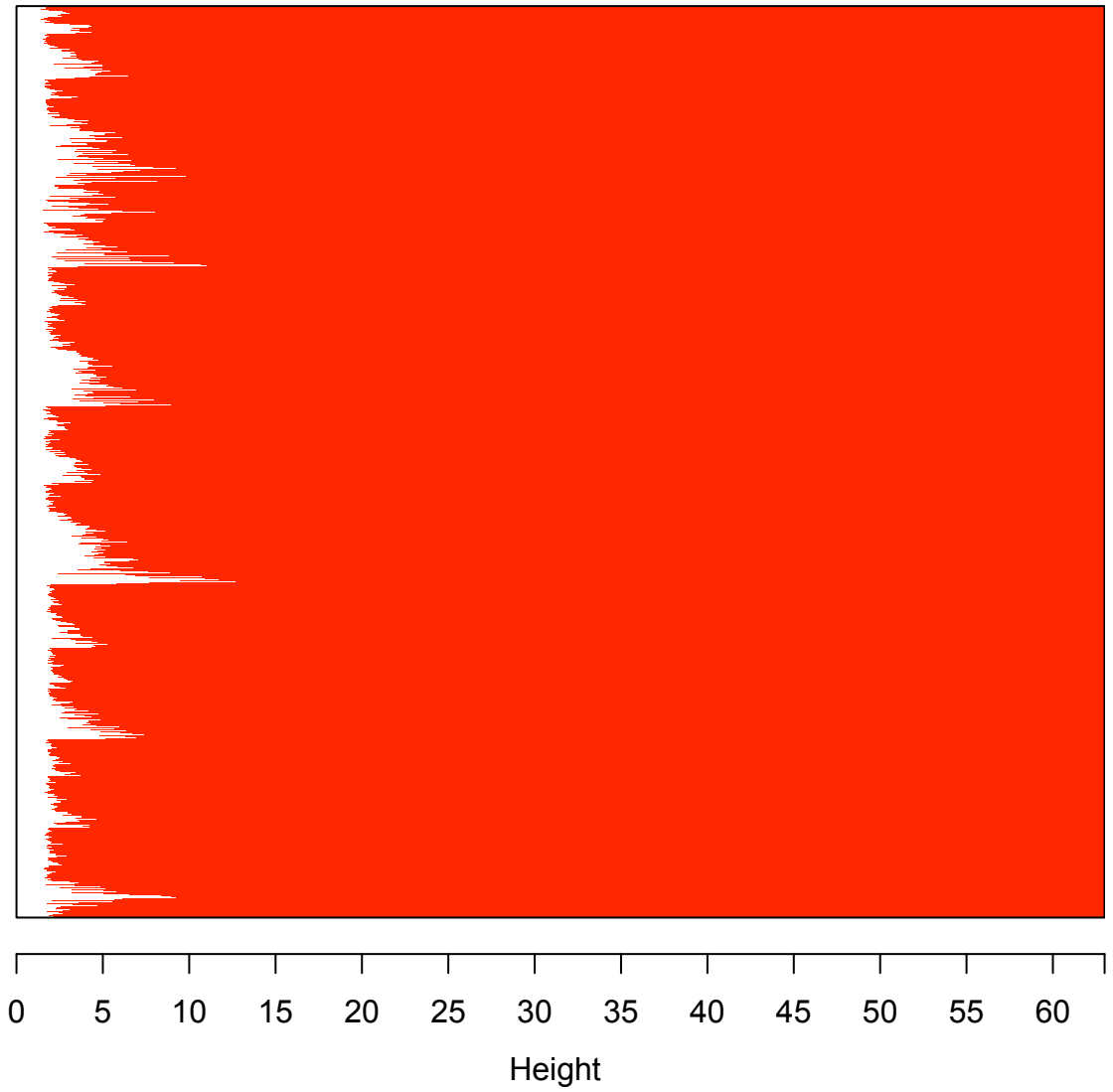
```
> library(cluster)
> ALL_exp=exprs(ALL)
> kc=agens(ALL_exp, metric =
   "euclidean”,method=“average”)
```

```
> kc=diana(ALL_exp,metric = "euclidean”)
```

Distance (metric) : euclidean or manhattan

agens

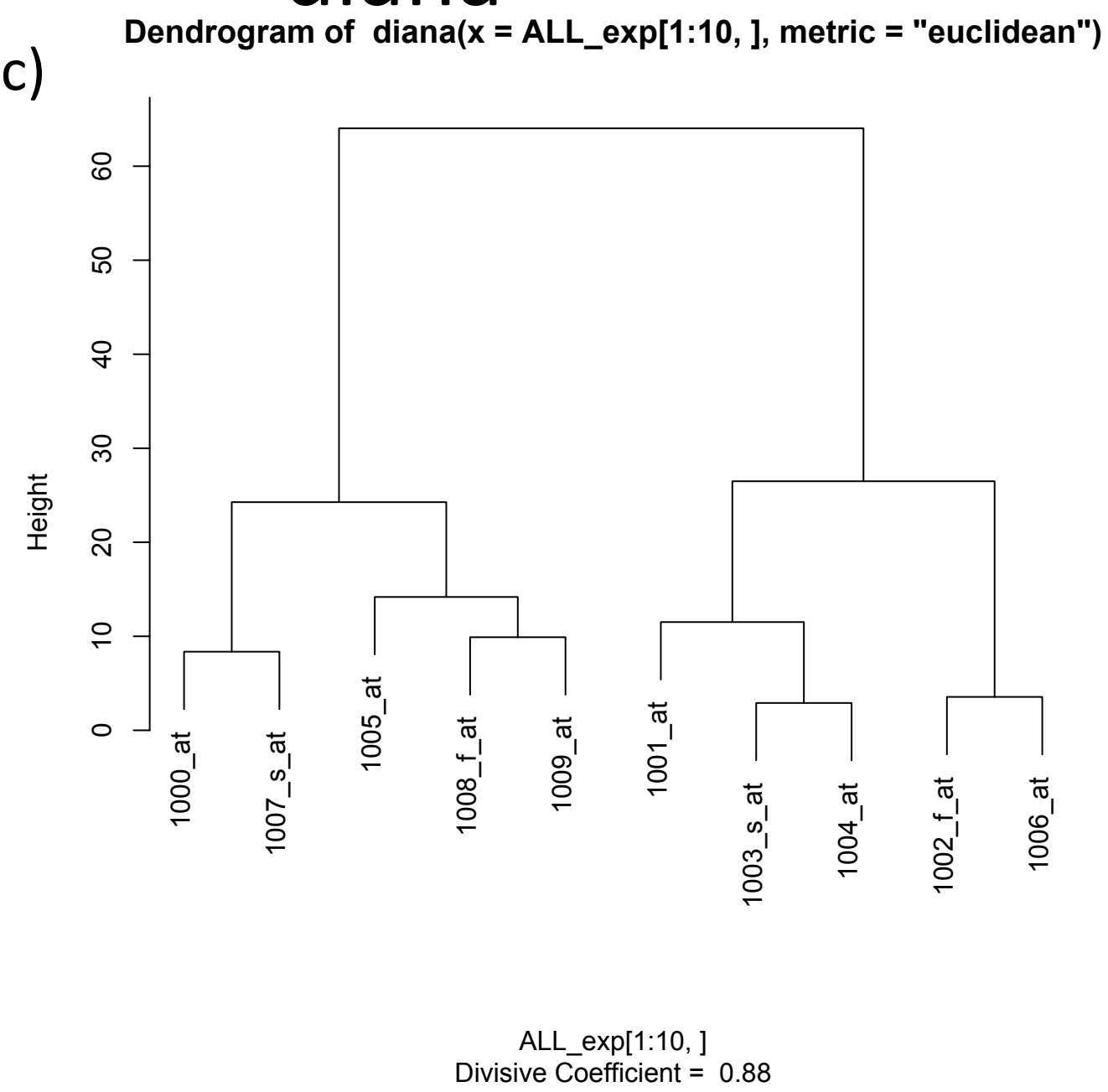
> plot(kc)



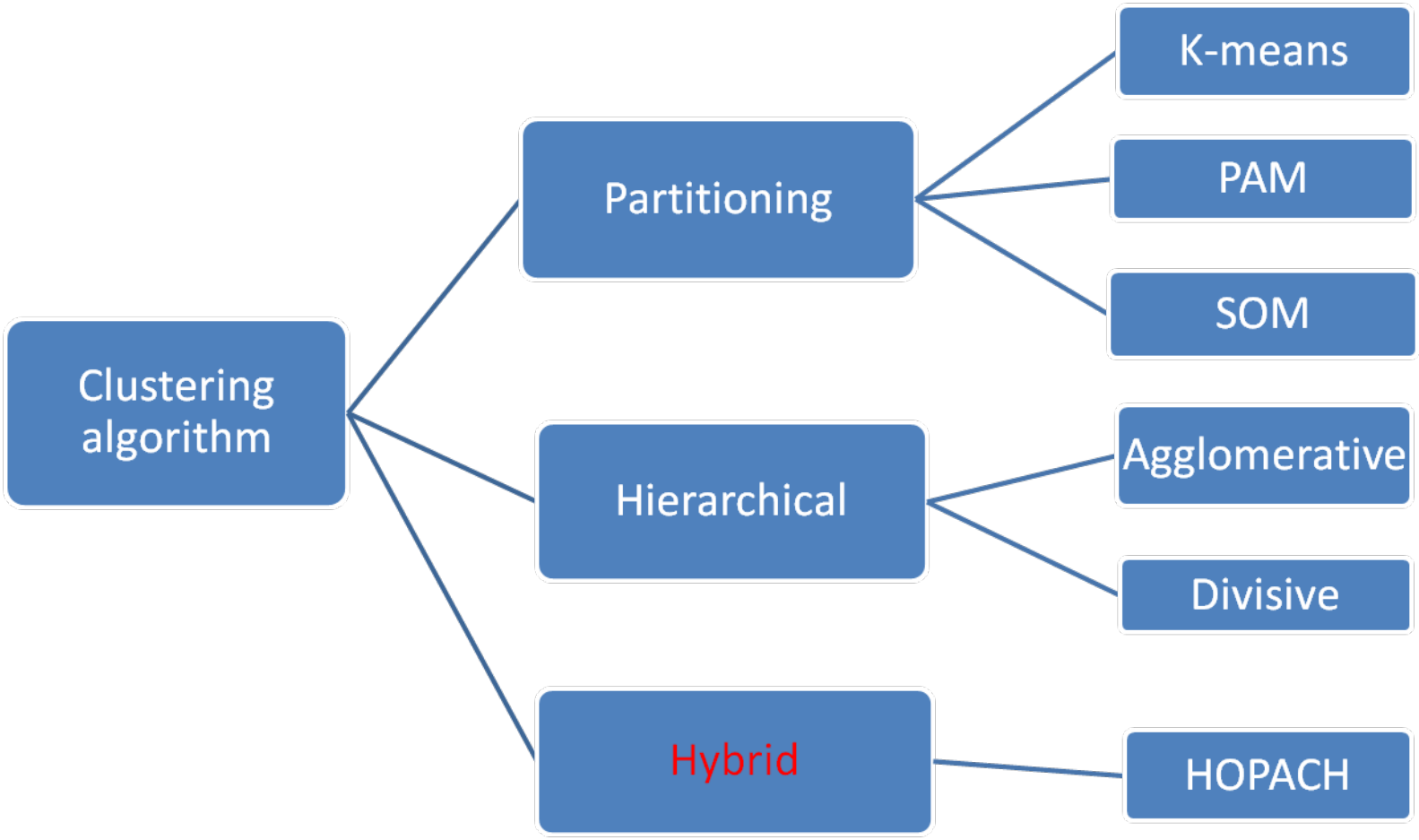
It is very slow for 20000 genes

diana

```
> plot(kc)
```



Type of Clustering algorithm



Hybrid clustering

- An example of hybrid clustering algorithm is HOPACH, the Hierarchical Ordered Partitioning And Collapsing Hybrid algorithm.

- It builds a tree of clusters, where the clusters in each level are ordered based on the pairwise dissimilarities between cluster medoids.

- The combination of recursive partitioning with a agglomerative collapsing step allows erroneously separated groups of elements to be reunited.

HOPACH algorithm

- Initial level: Begin with all elements at the root node
  - Partition
  - Order
  - Collapse
- Next level
  - Partition
  - Order
  - Collapse Iterate
- Iterate: Repeat until each node contains no more than 2 genes

HOPACH in R

http://cran.r-project.org/web/packages/hopach/index.html

> source("http://bioconductor.org/biocLite.R")

> biocLite(”hopach”)

> library(hopach)

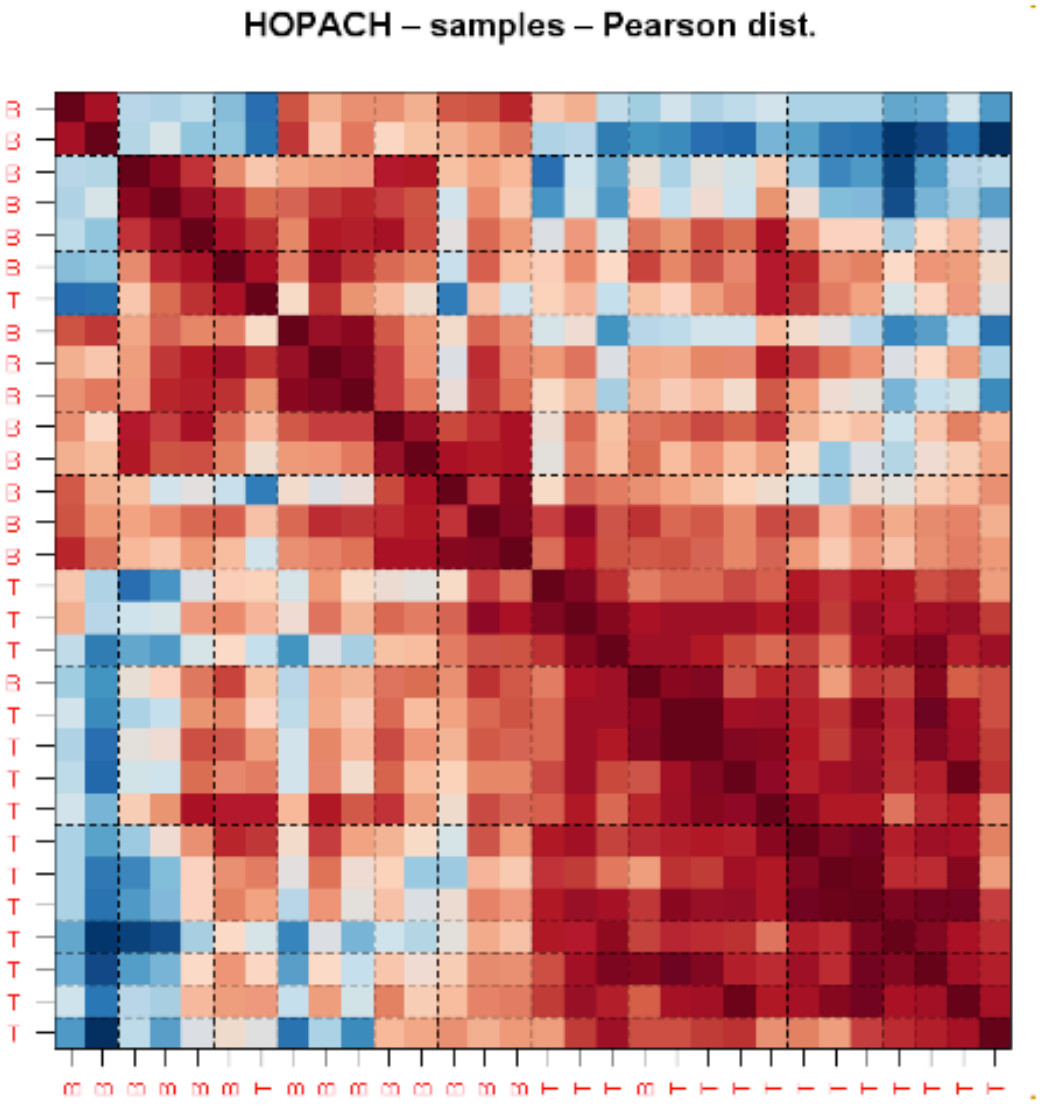
> ALL_exp=exprs(ALL)

> hc=hopach(ALL_exp, d = "cor”)

d= "cosangle”, "abscosangle”, "euclid”, "abseuclid”, "cor”, and "abscor”.

HOPACH in R

>dplot(ALL_exp[1:100,], kc)



- Heatmap shows the distance between genes
- Dotted lines: “breaks” between clusters

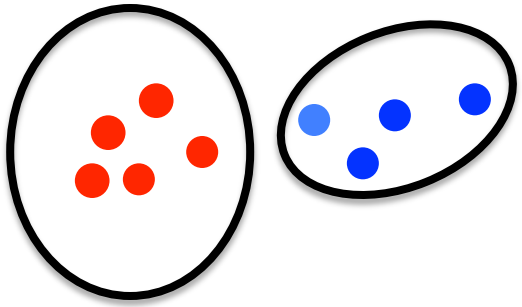
Fuzzy clustering

- In hard clustering, data is divided into distinct clusters, where each data element belongs to exactly one cluster.

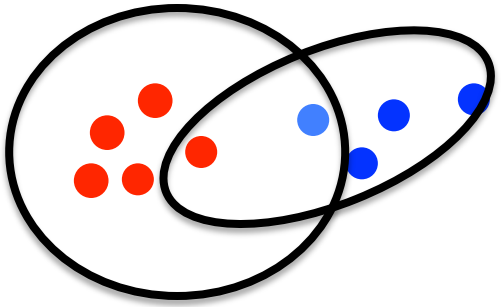
- In **fuzzy clustering** (also referred to as **soft clustering)**, data elements can belong to more than one cluster, and associated with each element is a set of membership levels. These indicate the strength of the association between that data element and a particular cluster.

- Fuzzy clustering can assign data elements to one or more clusters.

Fuzzy clustering



Hard clustering



Fuzzy Clustering

Fuzzy C-Means Algorithm

- Like the k-means algorithm, Fuzzy C-means (FCM) needs to pick a prefixed number C of clusters.

- FCM also tries to minimize the sum of within-cluster-variances

$$\min\left(\sum_{i=1}^{k}\sum_{x_j\in S_i}\left\|w_{ij}x_j-\mu_i\right\|^{2/(m-1)}\right)$$

- FCM differs from the k-means by the addition of the membership values w_{ij} and the fuzzifier m.
- The fuzzifier m determines the level of cluster fuzziness.

Fuzzy C-Means Algorithm

fanny() computes a fuzzy clustering of the data into k clusters.

```
> library(cluster)
> ALL_exp=exprs(ALL)
> fannyx <- fanny(ALL_exp, 3, memb.exp =2)
```

Outline

- Multiple Testing Procedures
- Data Visualization, Distance Measures
- Clustering
- Gene Annotation and Enrichment Analysis