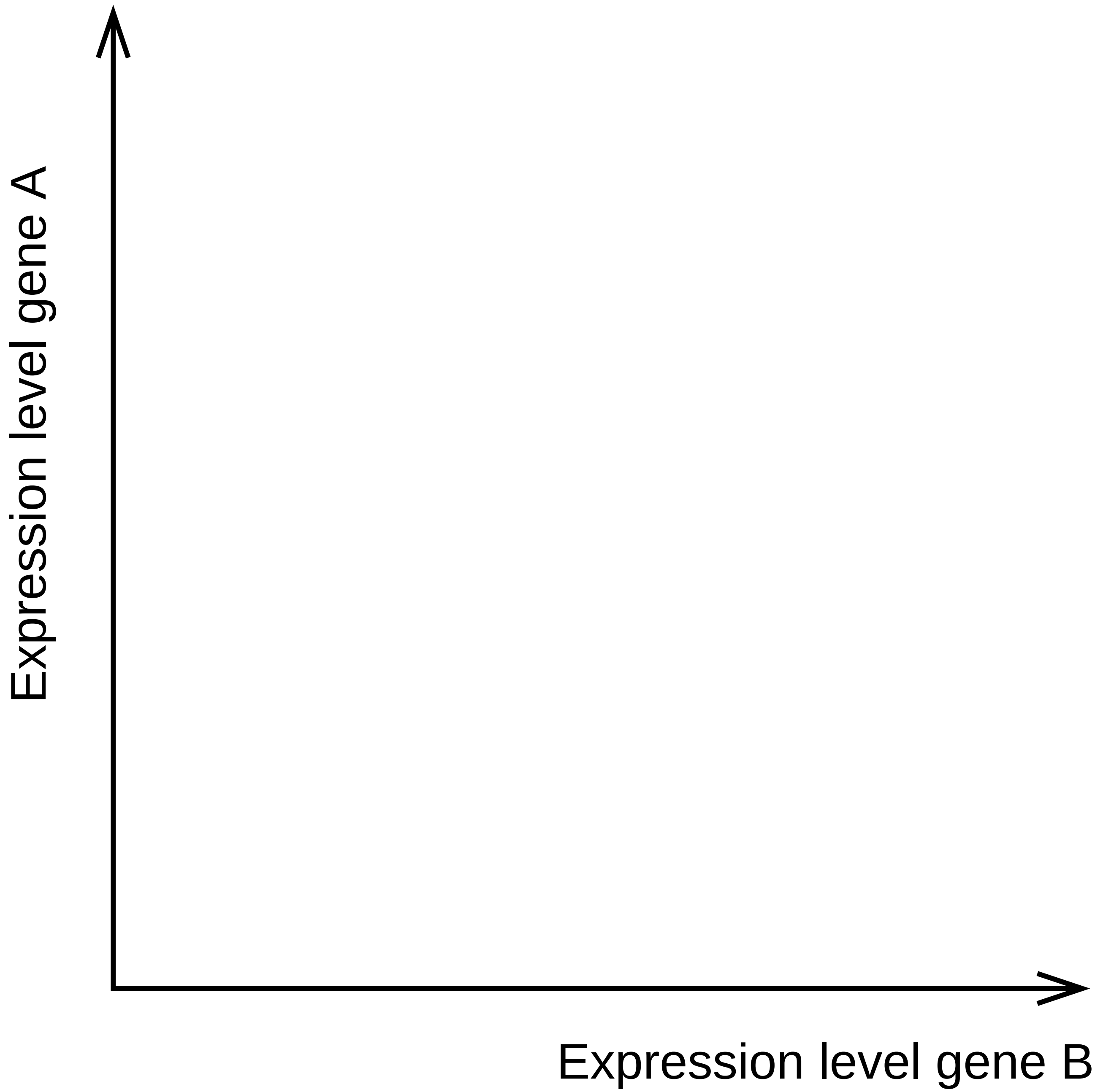
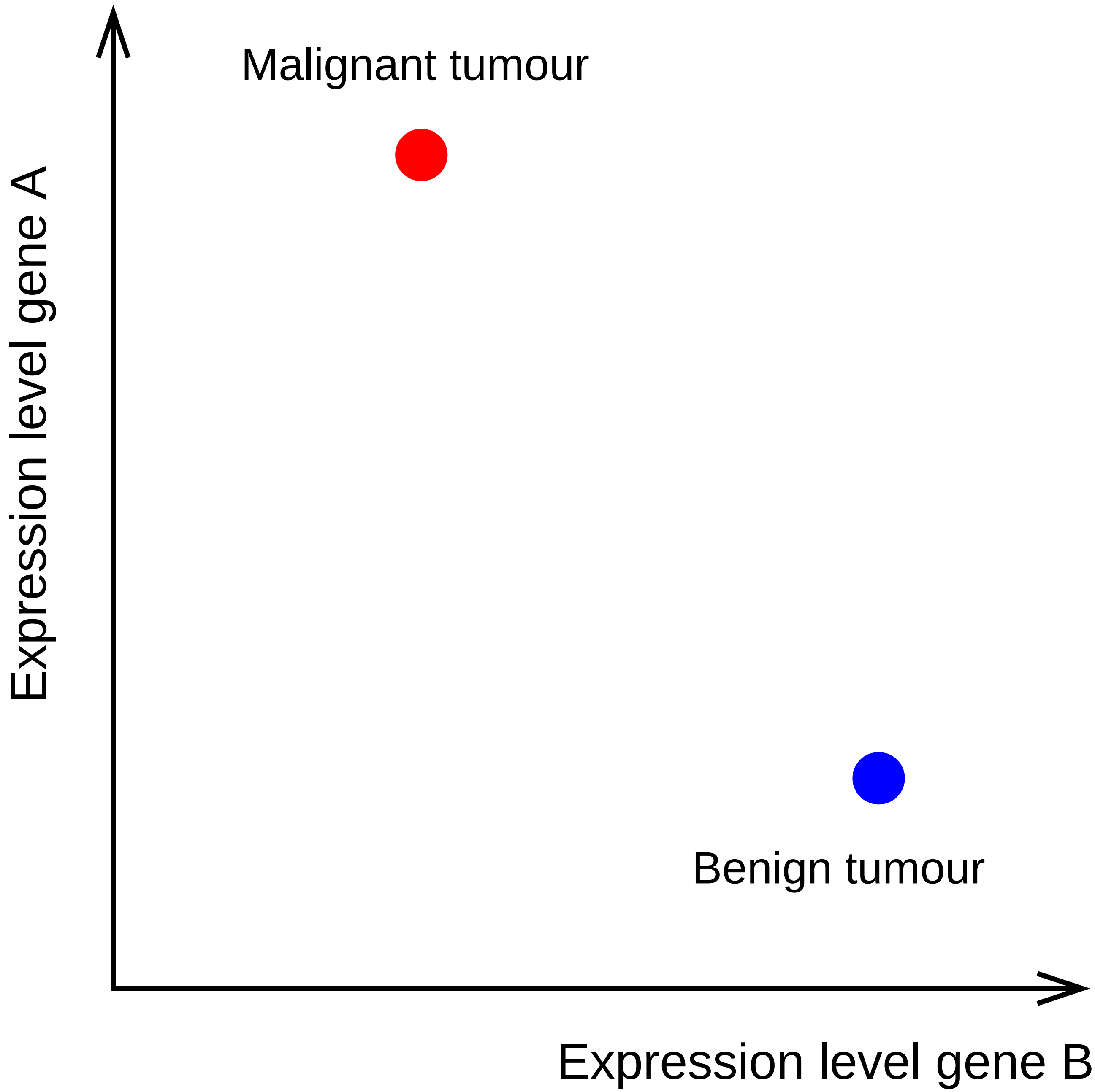
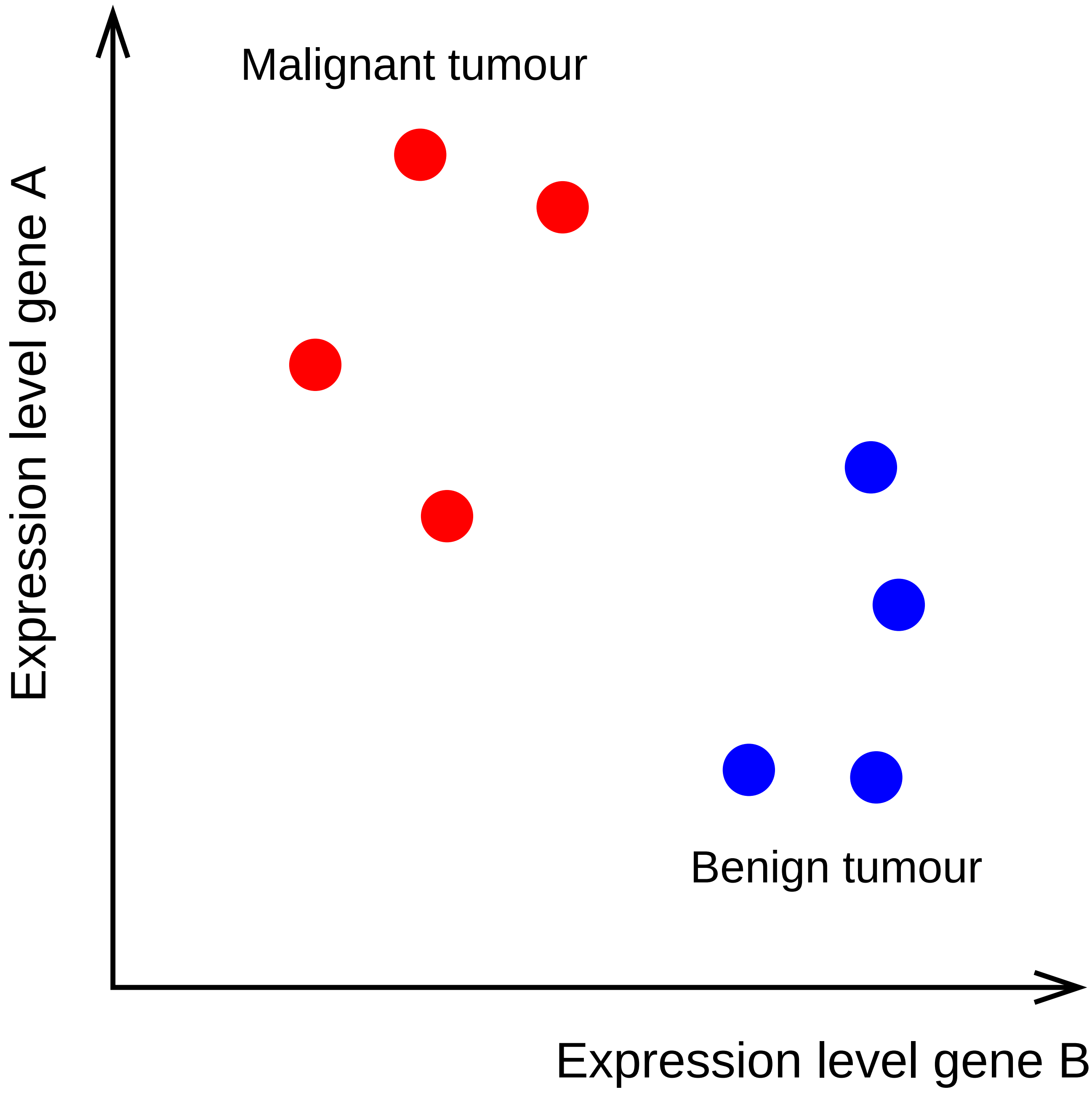
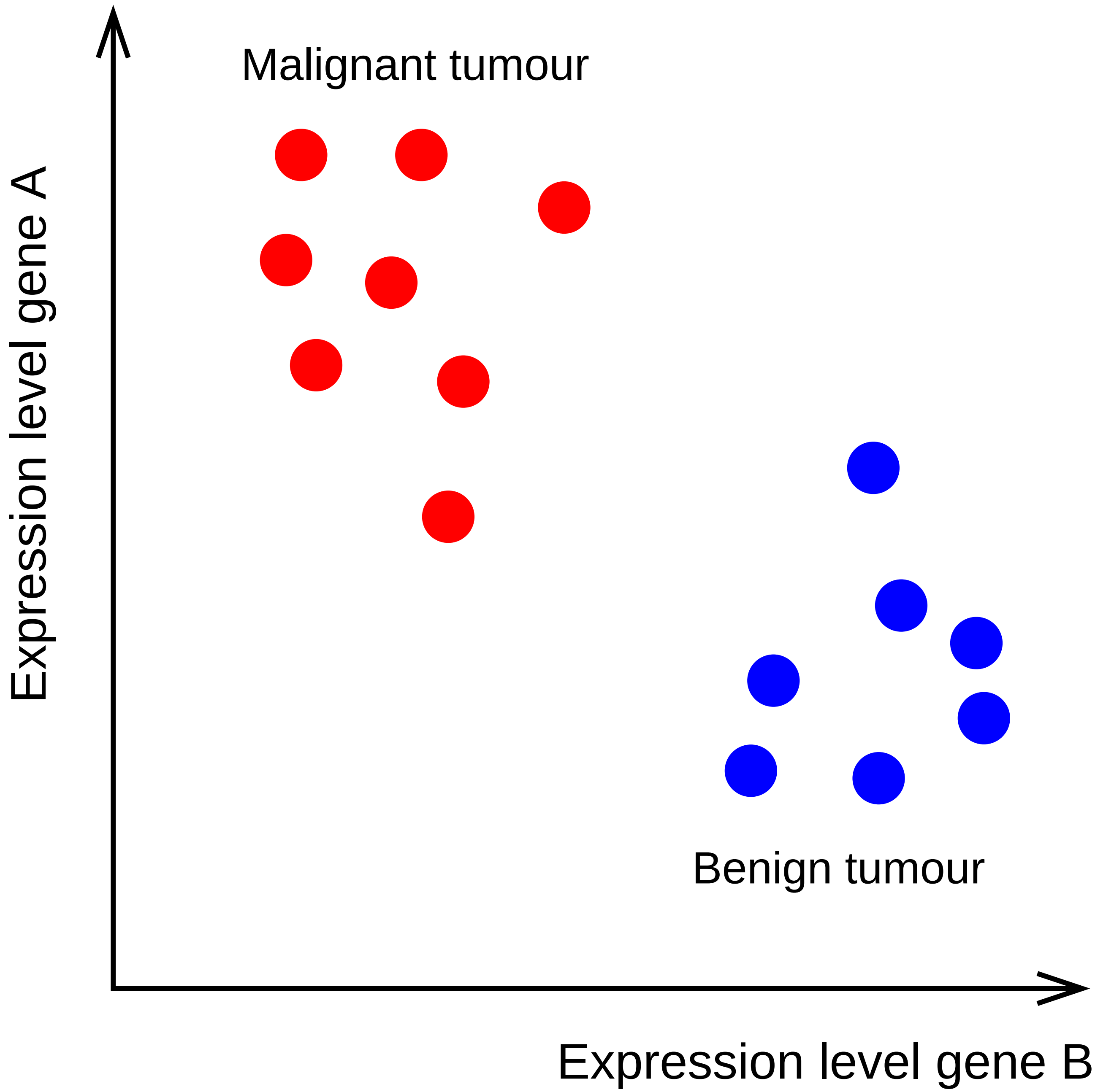
Objective of clustering

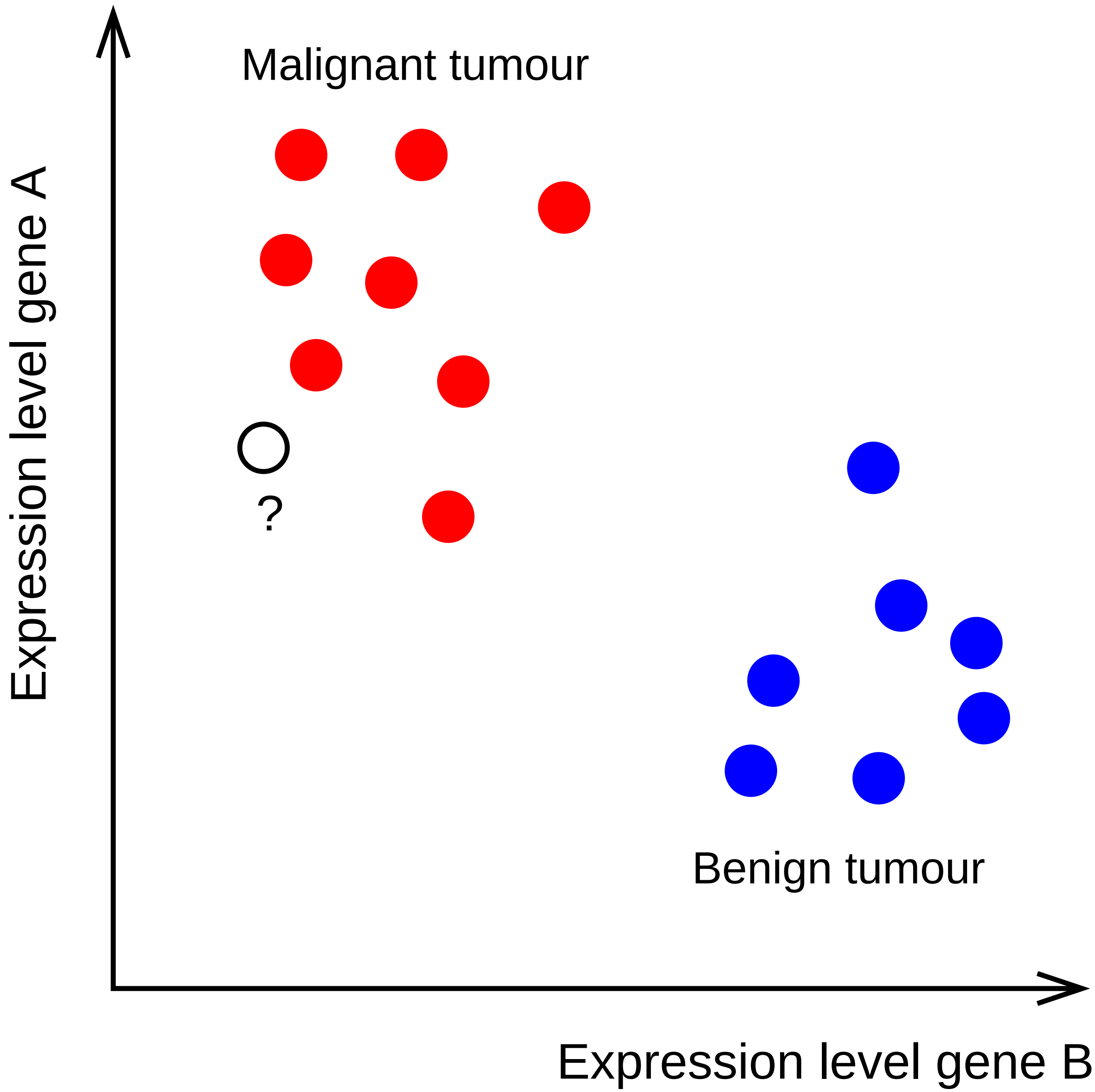
- Discover structures and patterns in high-dimensional data.
- Group data with similar patterns together.
- This reduces the complexity and facilitates interpretation.

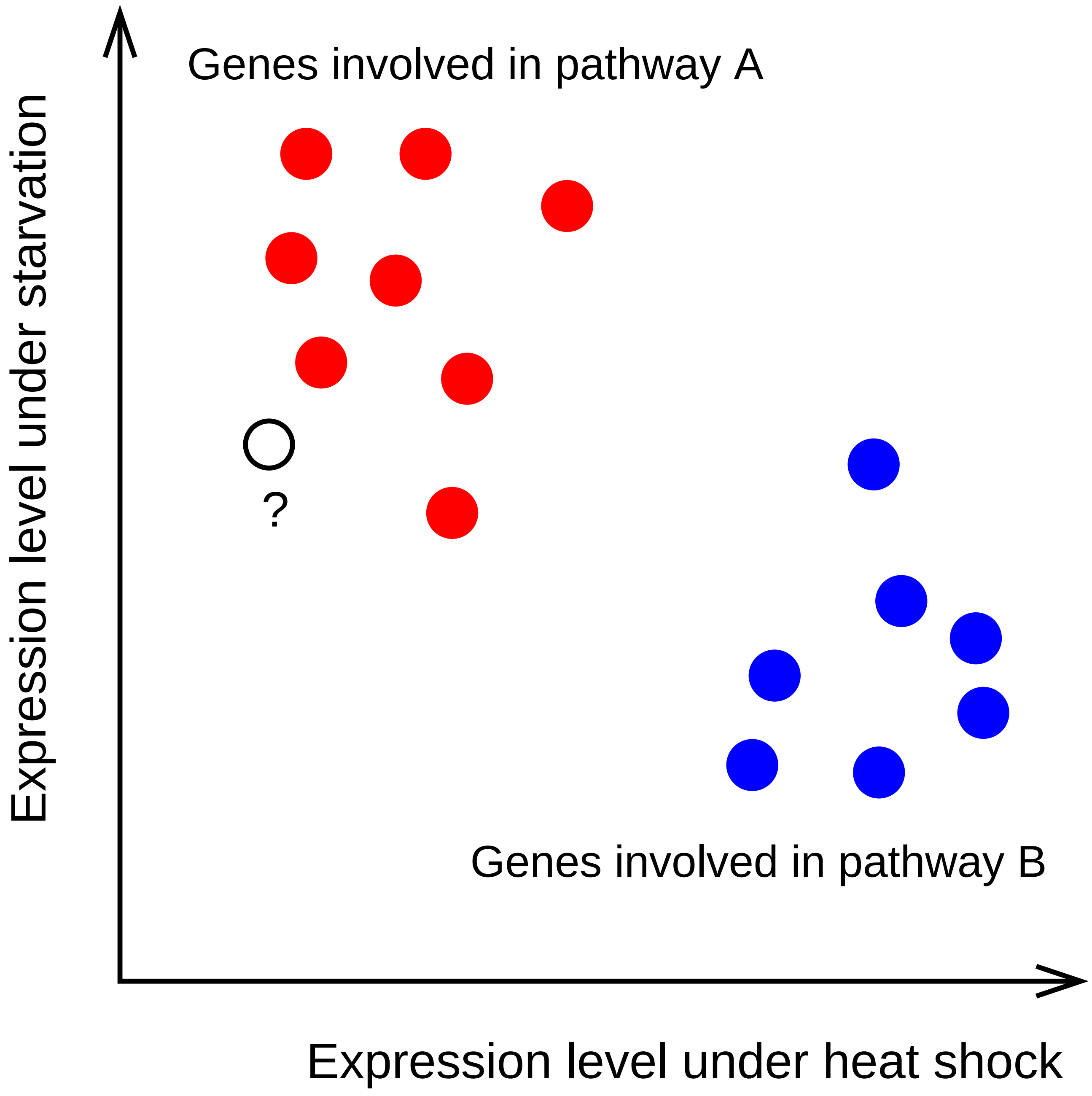

Expression level gene A
Expression level gene B


Malignant tumour
Expression level gene A
Benign tumour
Expression level gene B

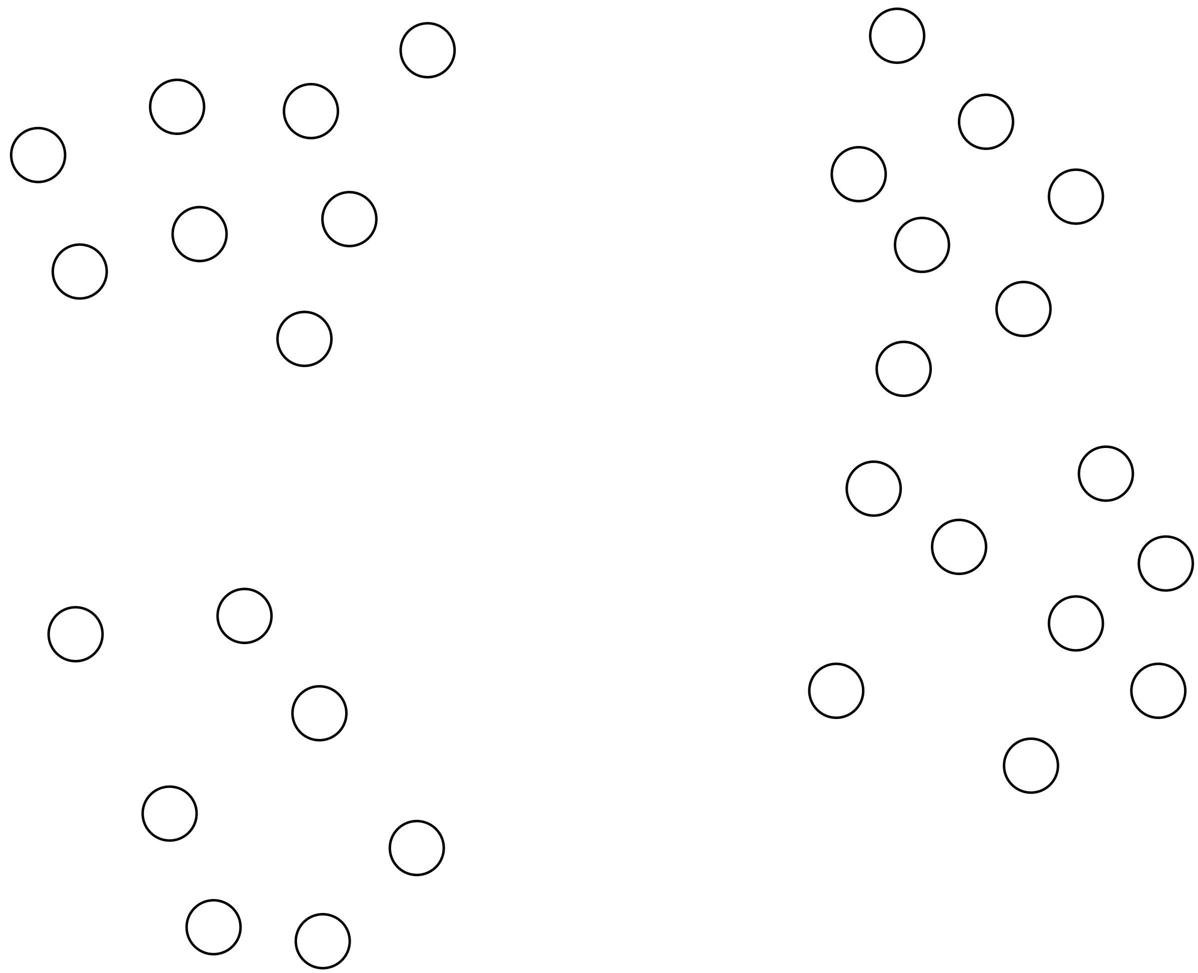

Malignant tumour
Benign tumour
Expression level gene A
Expression level gene B


Malignant tumour
Benign tumour
Expression level gene A
Expression level gene B

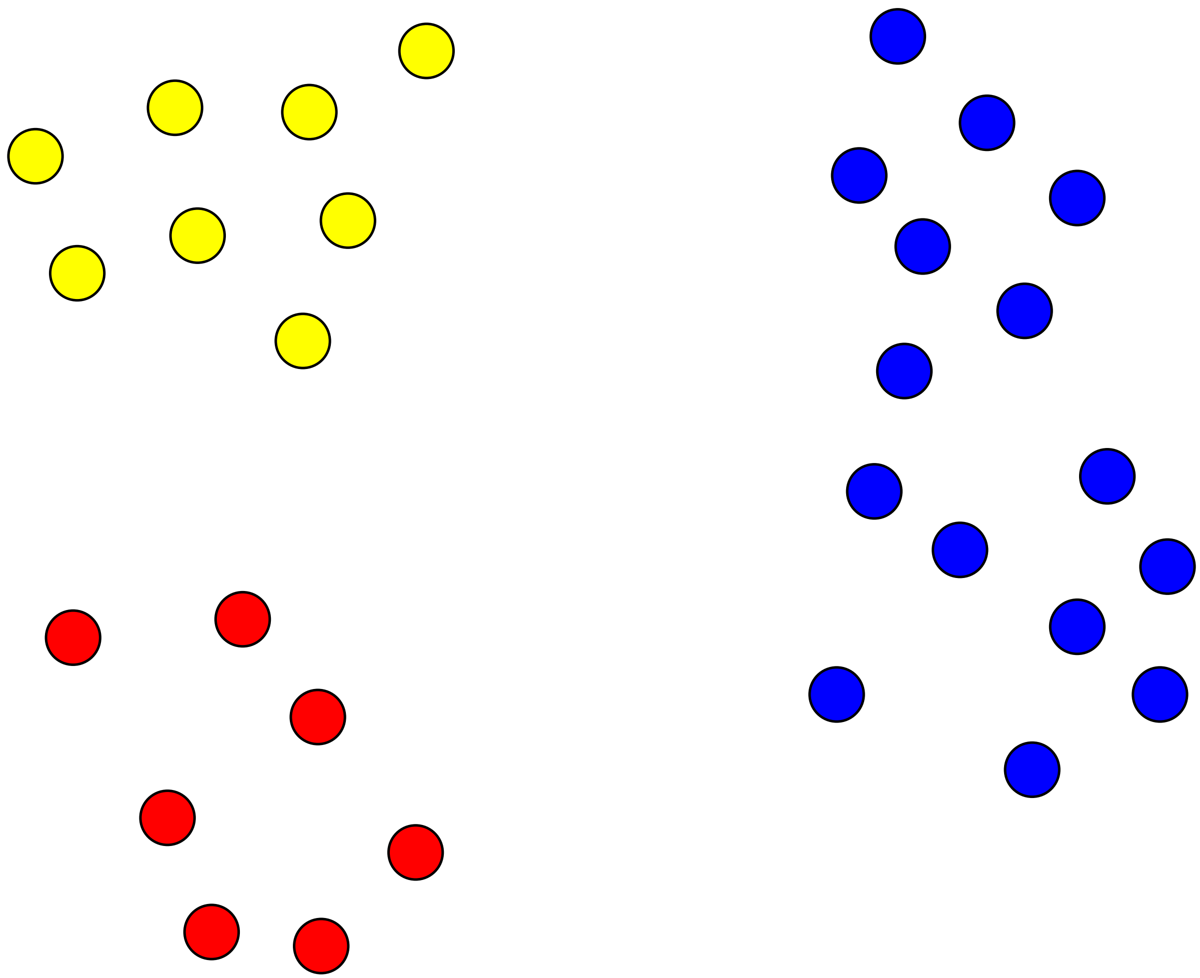

Malignant tumour
?
Benign tumour
Expression level gene A
Expression level gene B


Genes involved in pathway A
?
Genes involved in pathway B
Expression level under starvation
Expression level under heat shock

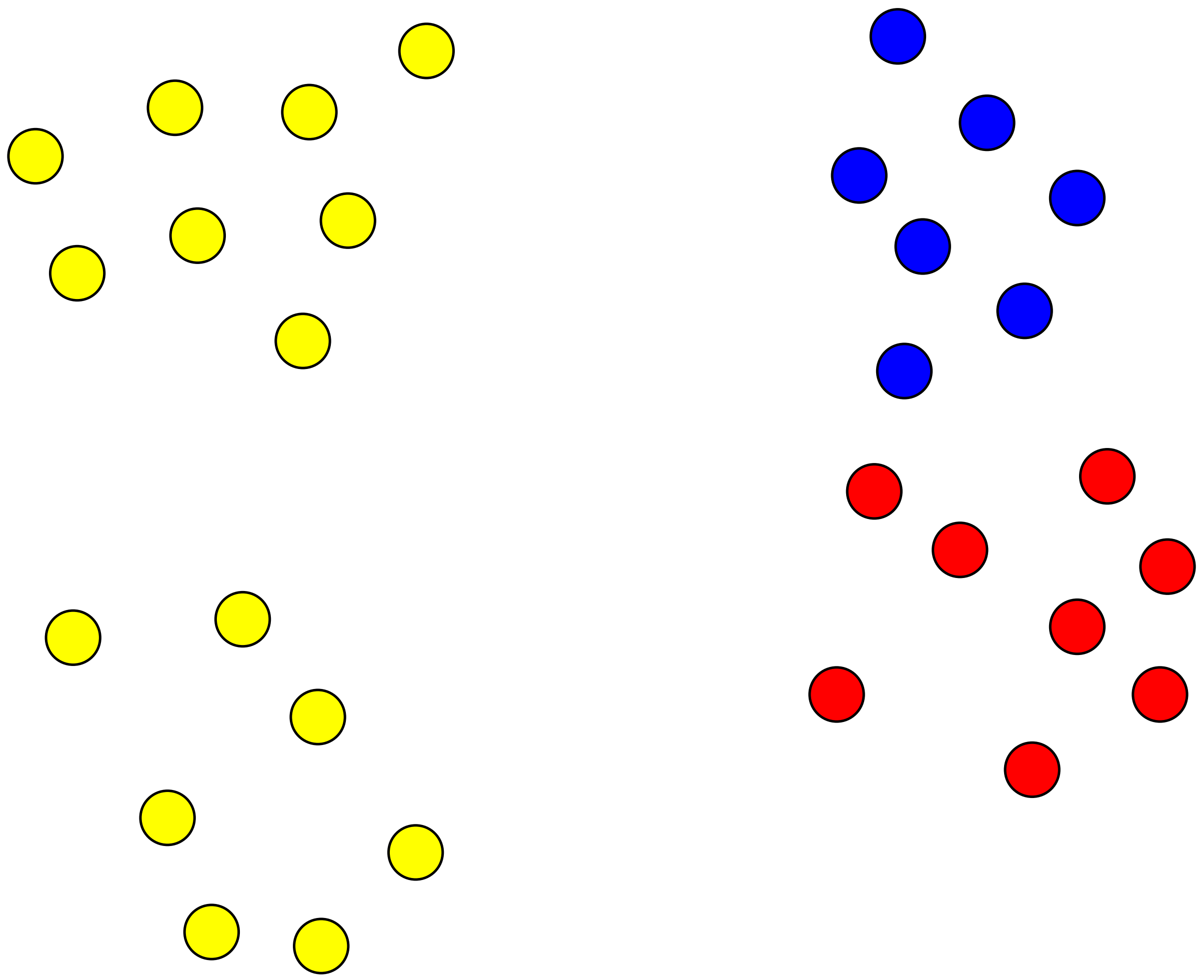
How shall we cluster the data?



Good clustering

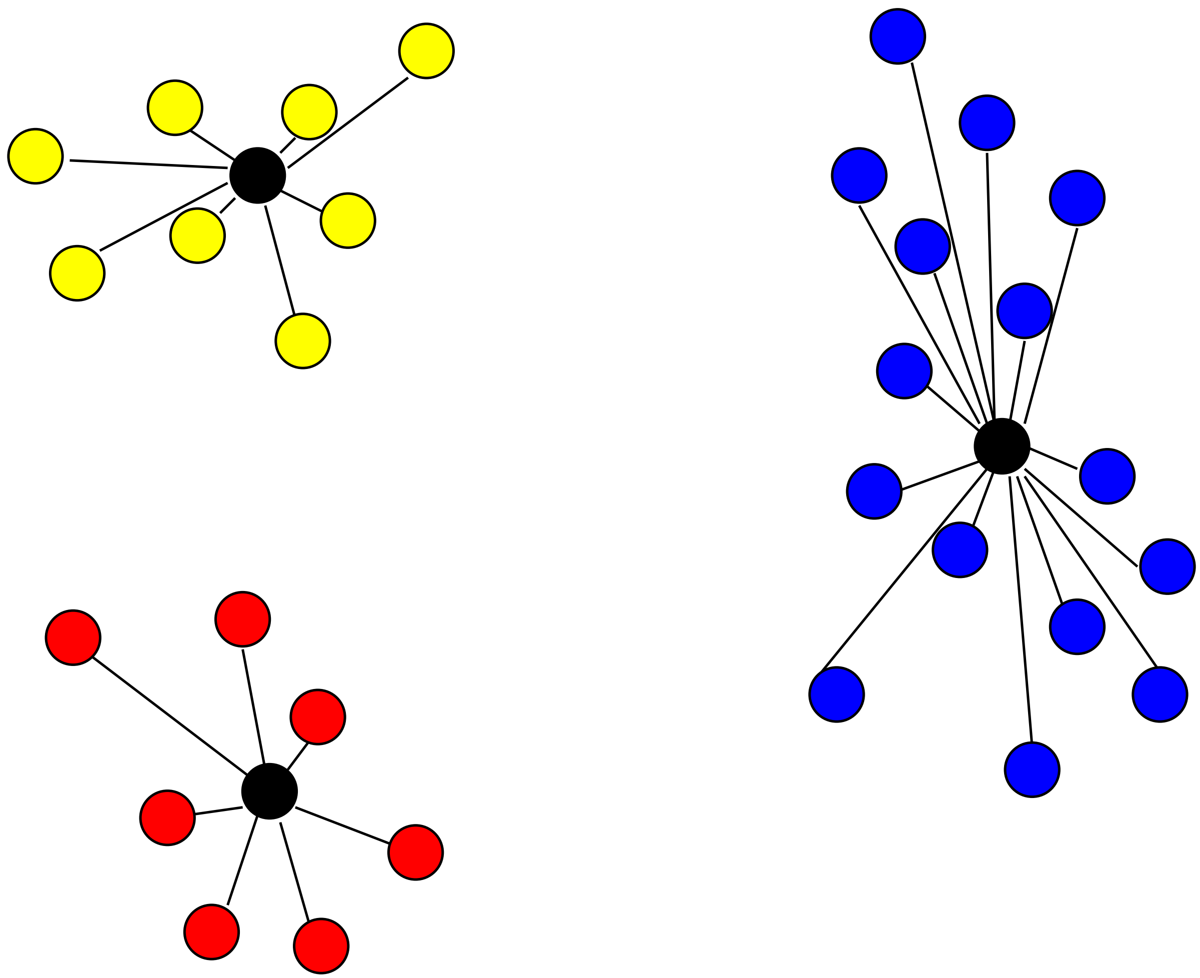


Bad clustering



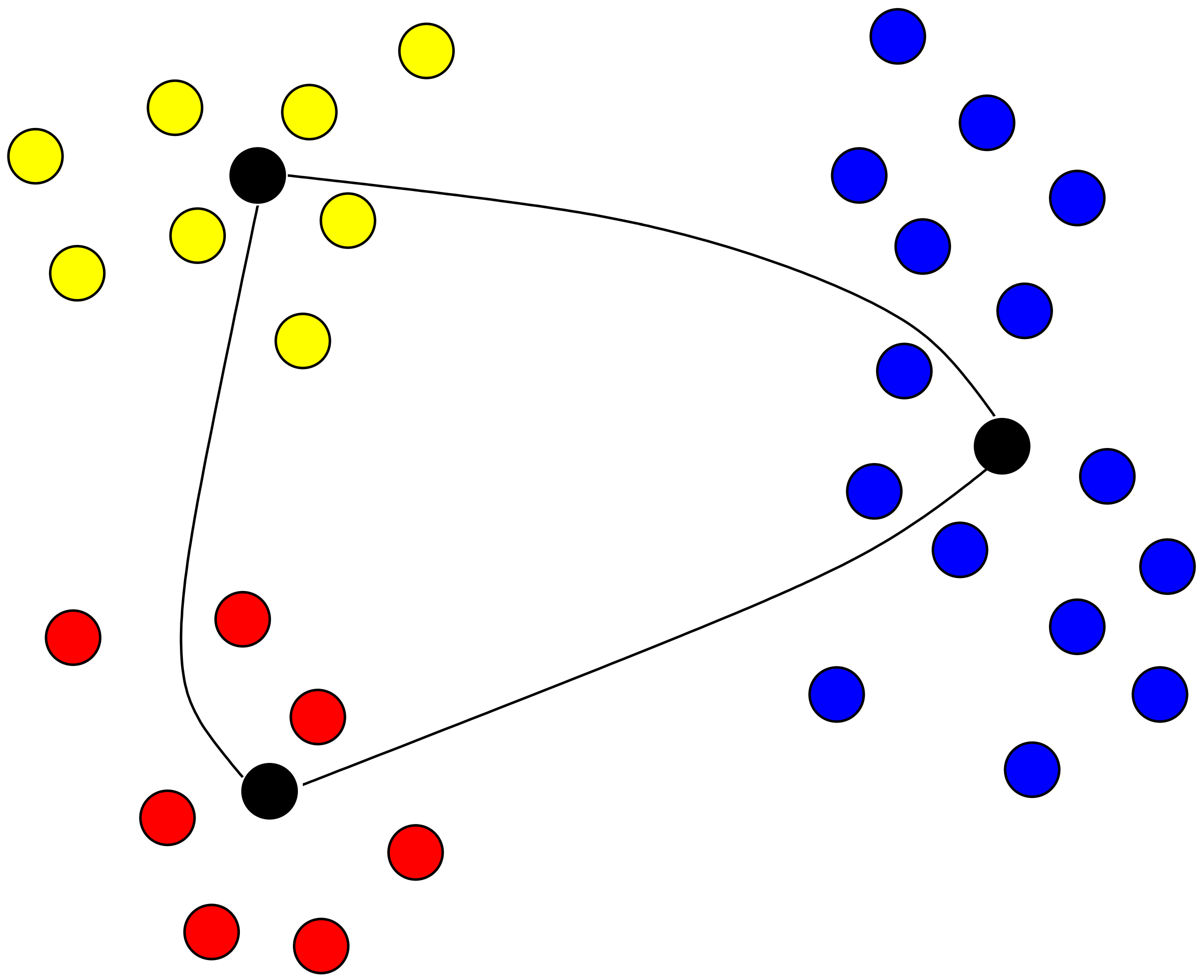
Within-group variance

Sum of squared vector-centroid distances



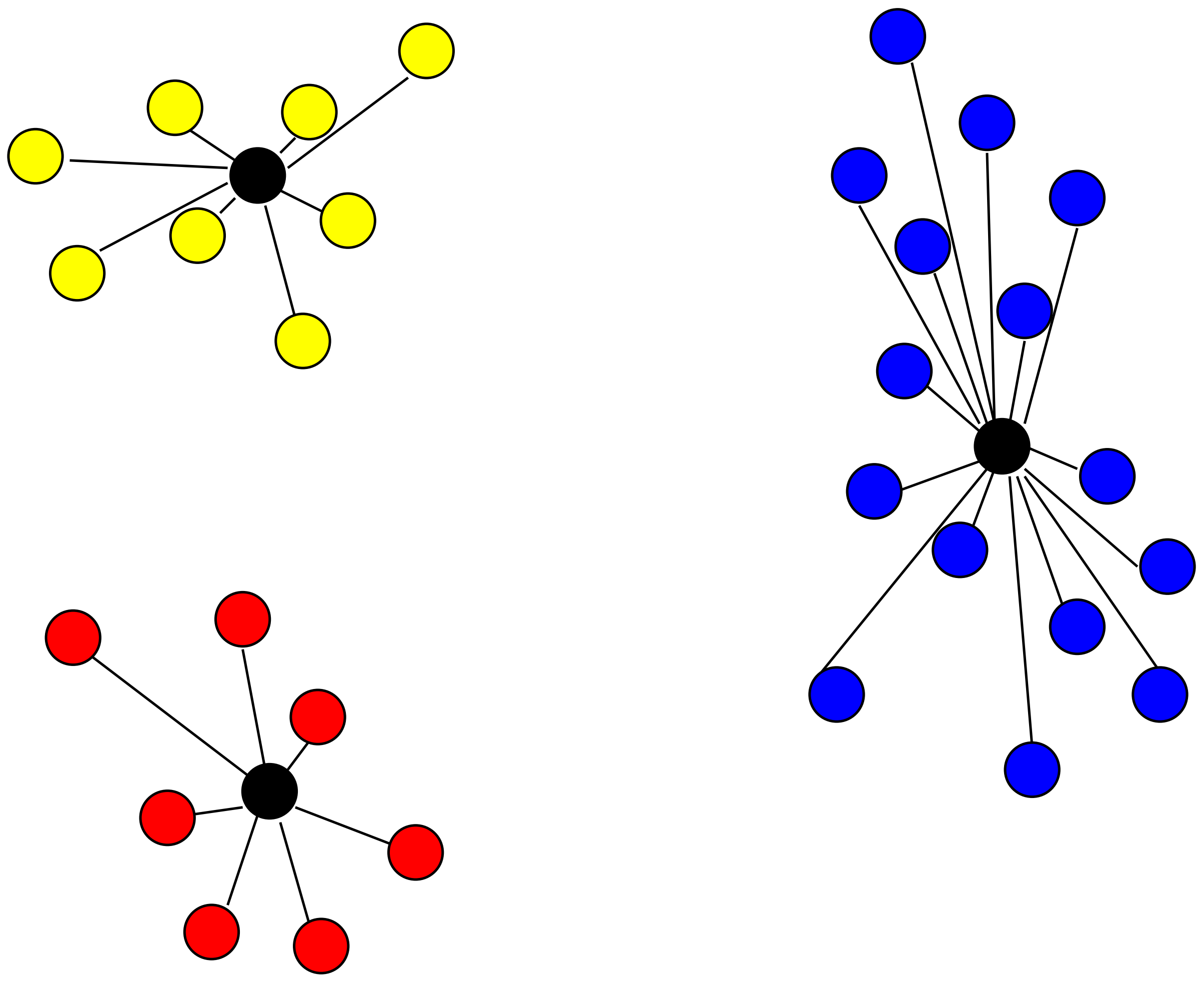
Between-group variance

Sum over squared distances between centroids



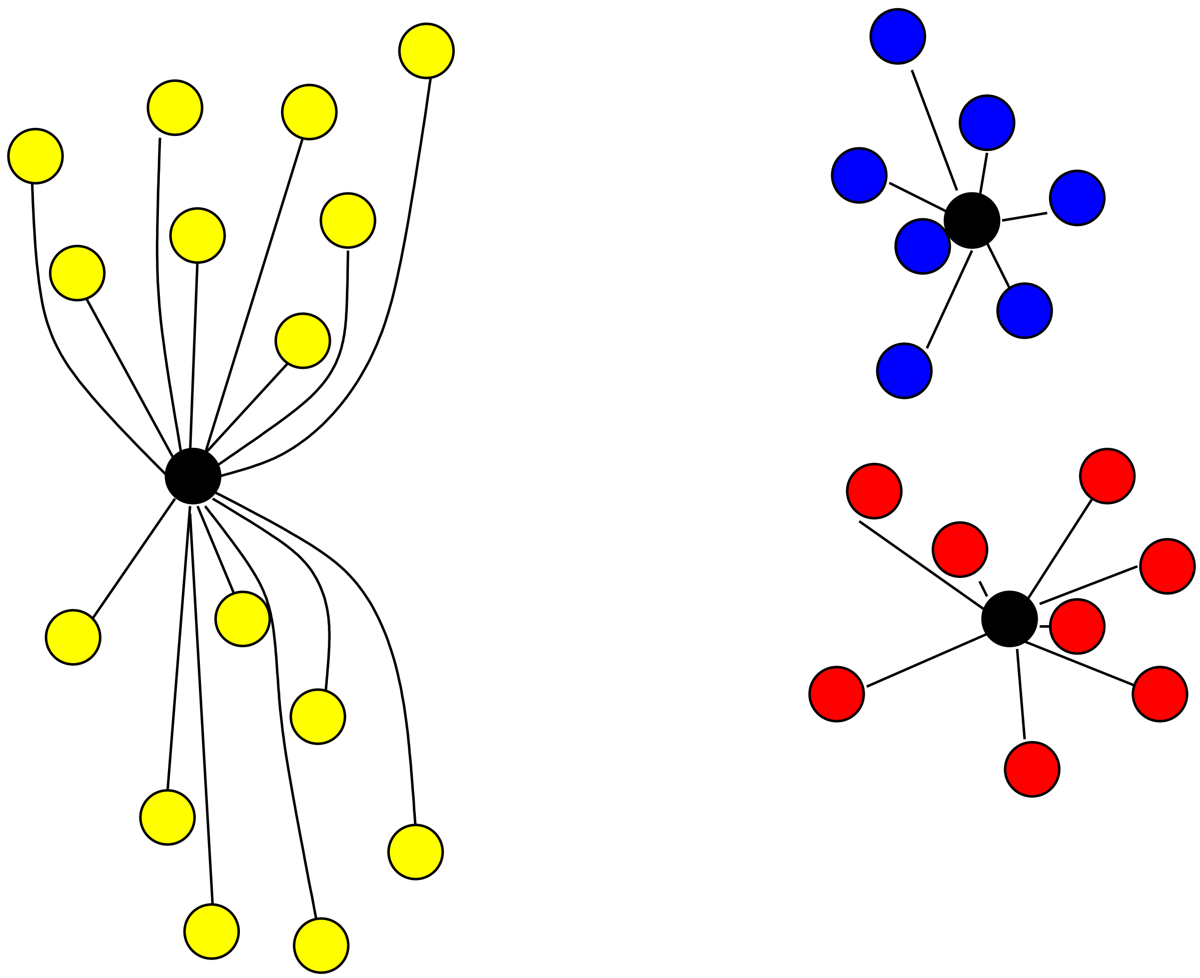
Within-group variance small

Tight clusters



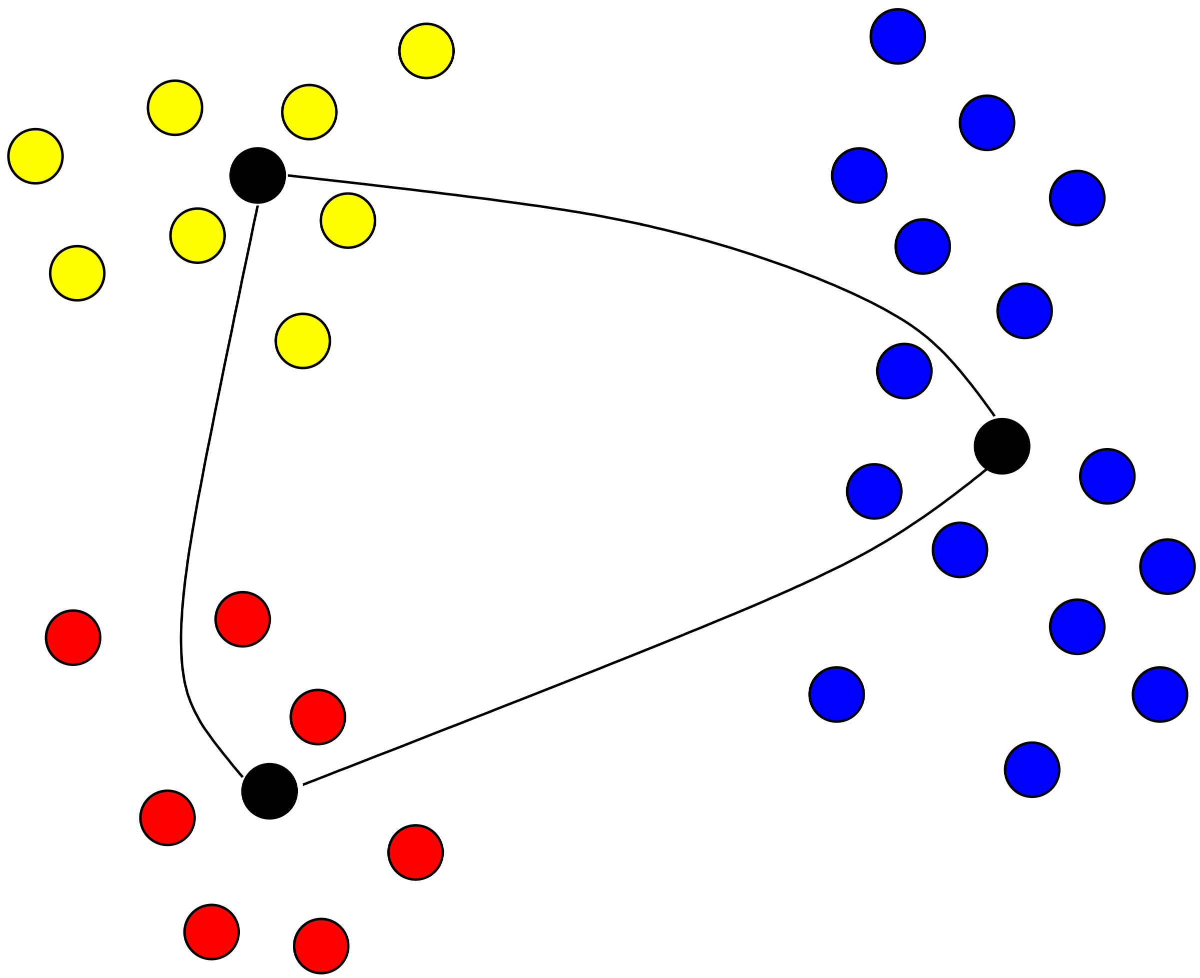
Within-group variance large

Diffuse clusters



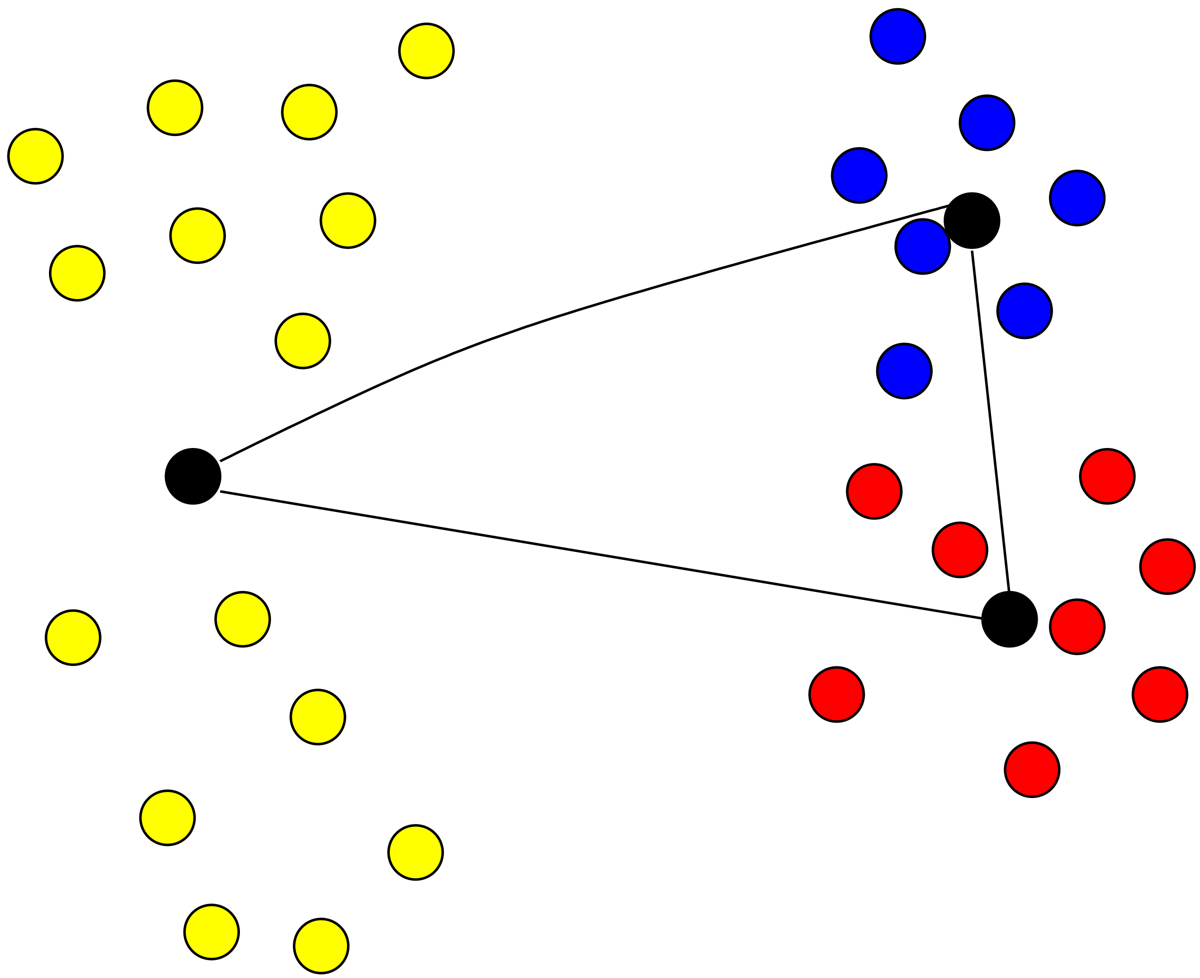
Between-group variance large

Clusters far apart



Between-group variance small

Clusters close together



How to cluster the data

How to cluster the data

- Minimize the within-group variance
  $\longrightarrow$ Tight clusters

How to cluster the data

- Minimize the within-group variance
  $\longrightarrow$ Tight clusters

- Maximize the between-group variance
  $\longrightarrow$ Clusters well separated

How to cluster the data

- Minimize the within-group variance
  $\longrightarrow$ Tight clusters

- Maximize the between-group variance
  $\longrightarrow$ Clusters well separated

- Problem NP-hard
  $\longrightarrow$ Heuristic algorithms and approximations are needed.

K-means clustering

K-means clustering

- Objective: Partition the data into a predefined number of clusters, K.

- Method: Alternatingly update
  - the cluster assignment of each data vector;
  - the cluster centroids.

- Decide on the number of clusters, K.

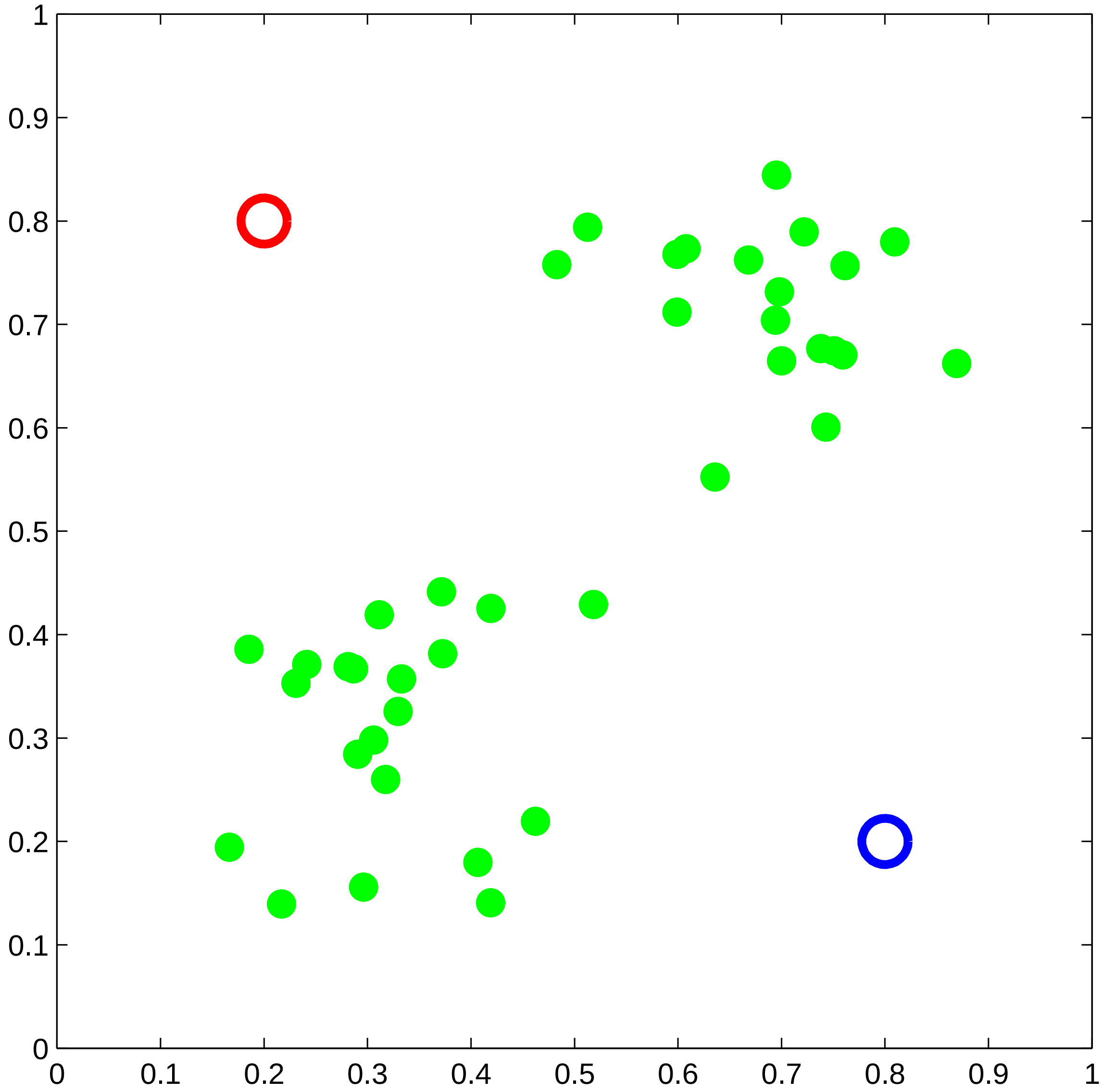
- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.

- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.
- Iteration (until cluster assignments remain unchanged):
  - For all data vectors $\mathbf{x}_i$, $i = 1, \ldots, N$, and all centroids $\mathbf{c}_k$, $k = 1, \ldots, K$: Compute the distance d_{ik} between the data vector $\mathbf{x}_i$ and the centroid $\mathbf{c}_k$.

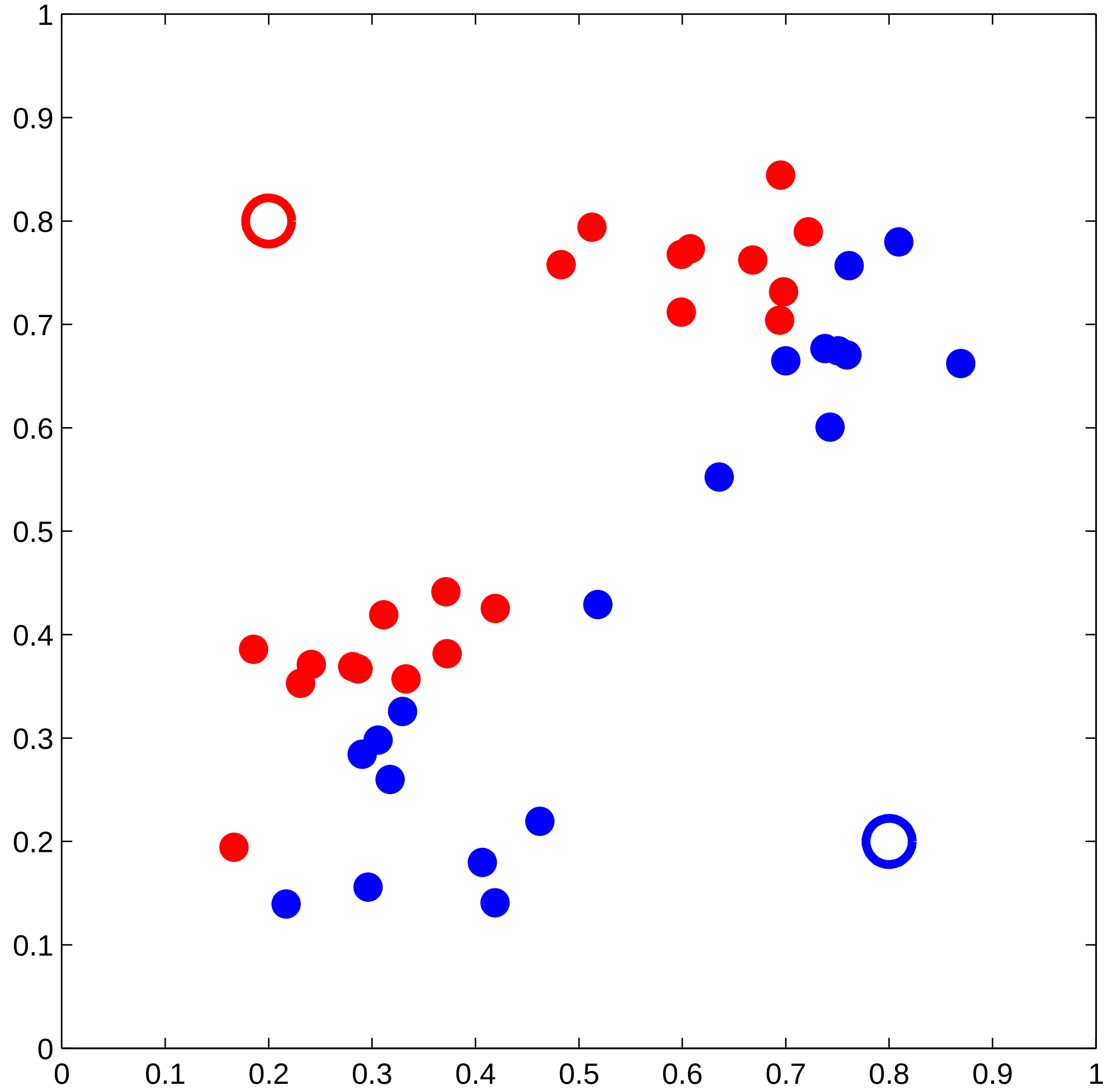
- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.
- Iteration (until cluster assignments remain unchanged):
  - For all data vectors $\mathbf{x}_i$, $i = 1, \ldots, N$, and all centroids $\mathbf{c}_k$, $k = 1, \ldots, K$: Compute the distance d_{ik} between the data vector $\mathbf{x}_i$ and the centroid $\mathbf{c}_k$.
  - Assign each data vector $\mathbf{x}_i$ to the closest centroid $\mathbf{c}_k$, that is, the one with minimal d_{ik}. Record the cluster membership in an indicator variable λ_{ik}, with $\lambda_{ik} = 1$ if $\mathbf{x}_i \rightarrow \mathbf{c}_k$ and $\lambda_{ik} = 0$ otherwise.

- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.
- Iteration (until cluster assignments remain unchanged):
  - For all data vectors $\mathbf{x}_i$, $i = 1, \ldots, N$, and all centroids $\mathbf{c}_k$, $k = 1, \ldots, K$: Compute the distance d_{ik} between the data vector $\mathbf{x}_i$ and the centroid $\mathbf{c}_k$.
  - Assign each data vector $\mathbf{x}_i$ to the closest centroid $\mathbf{c}_k$, that is, the one with minimal d_{ik}. Record the cluster membership in an indicator variable λ_{ik}, with $\lambda_{ik} = 1$ if $\mathbf{x}_i \rightarrow \mathbf{c}_k$ and $\lambda_{ik} = 0$ otherwise.
  - Set each cluster centroid to the mean of its assigned cluster: $\mathbf{c}_k = \frac{\sum_i \lambda_{ik} \mathbf{x}_i}{\sum_i \lambda_{ik}}$

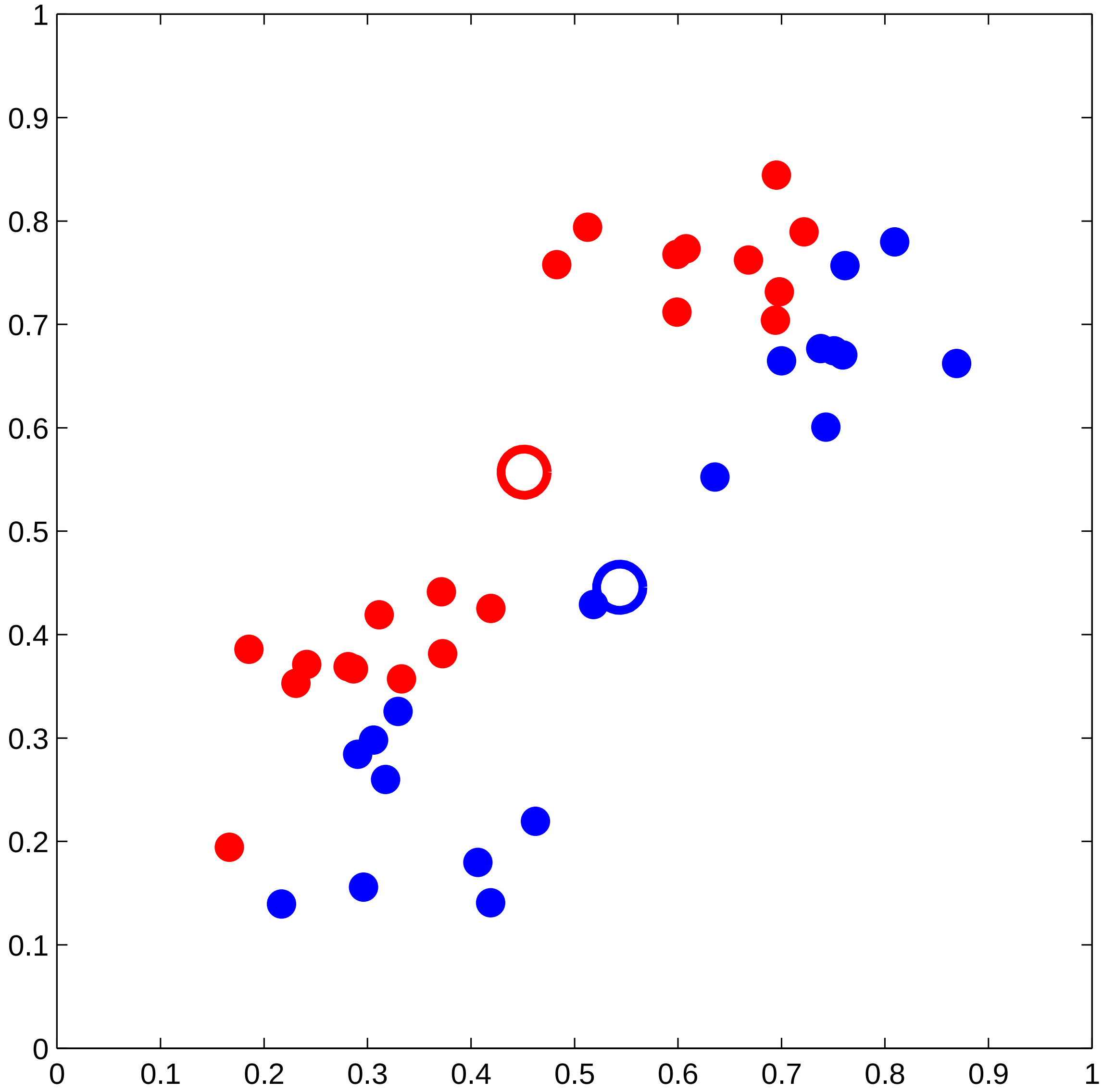
A good example of K-means clustering



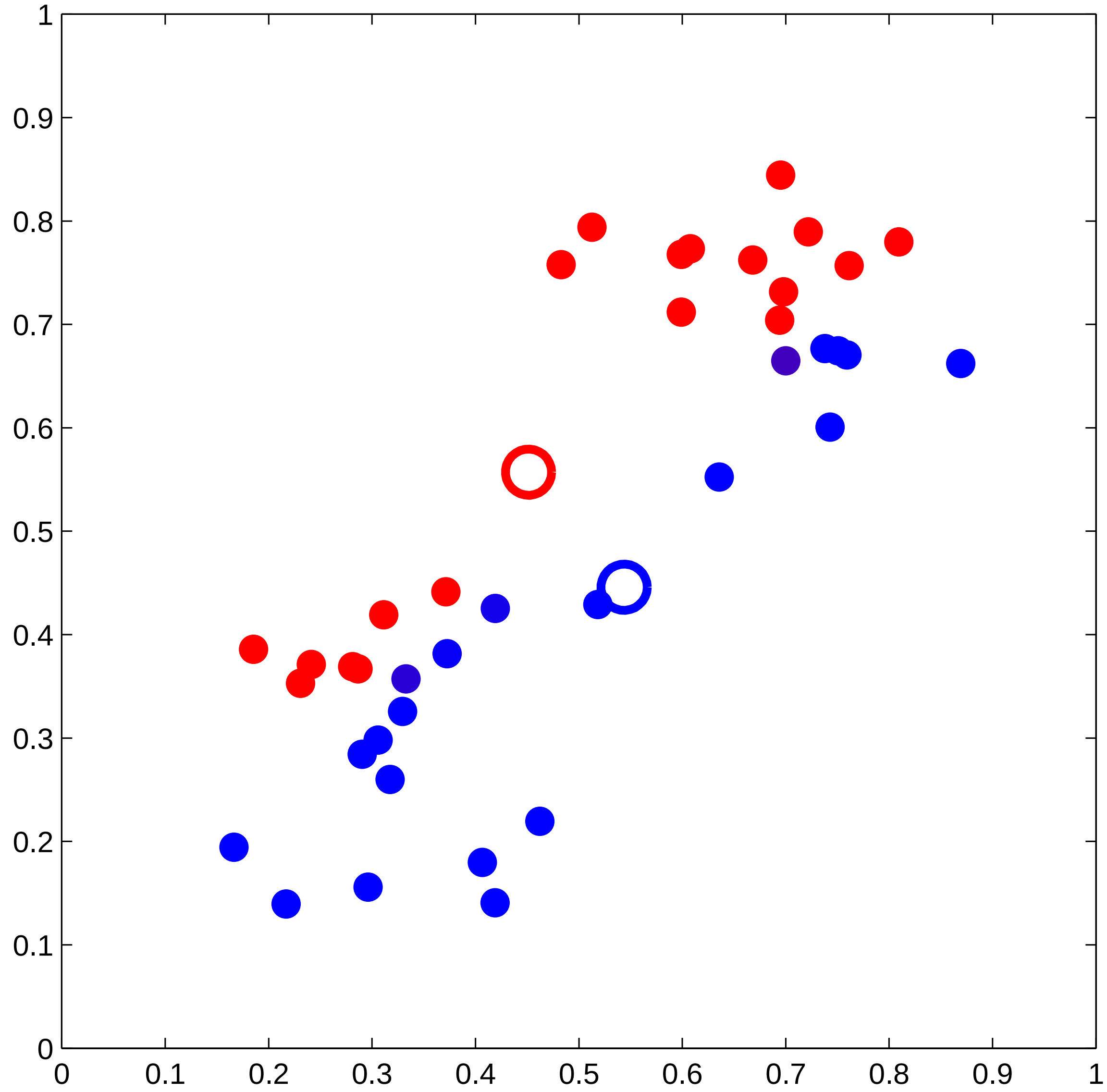
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



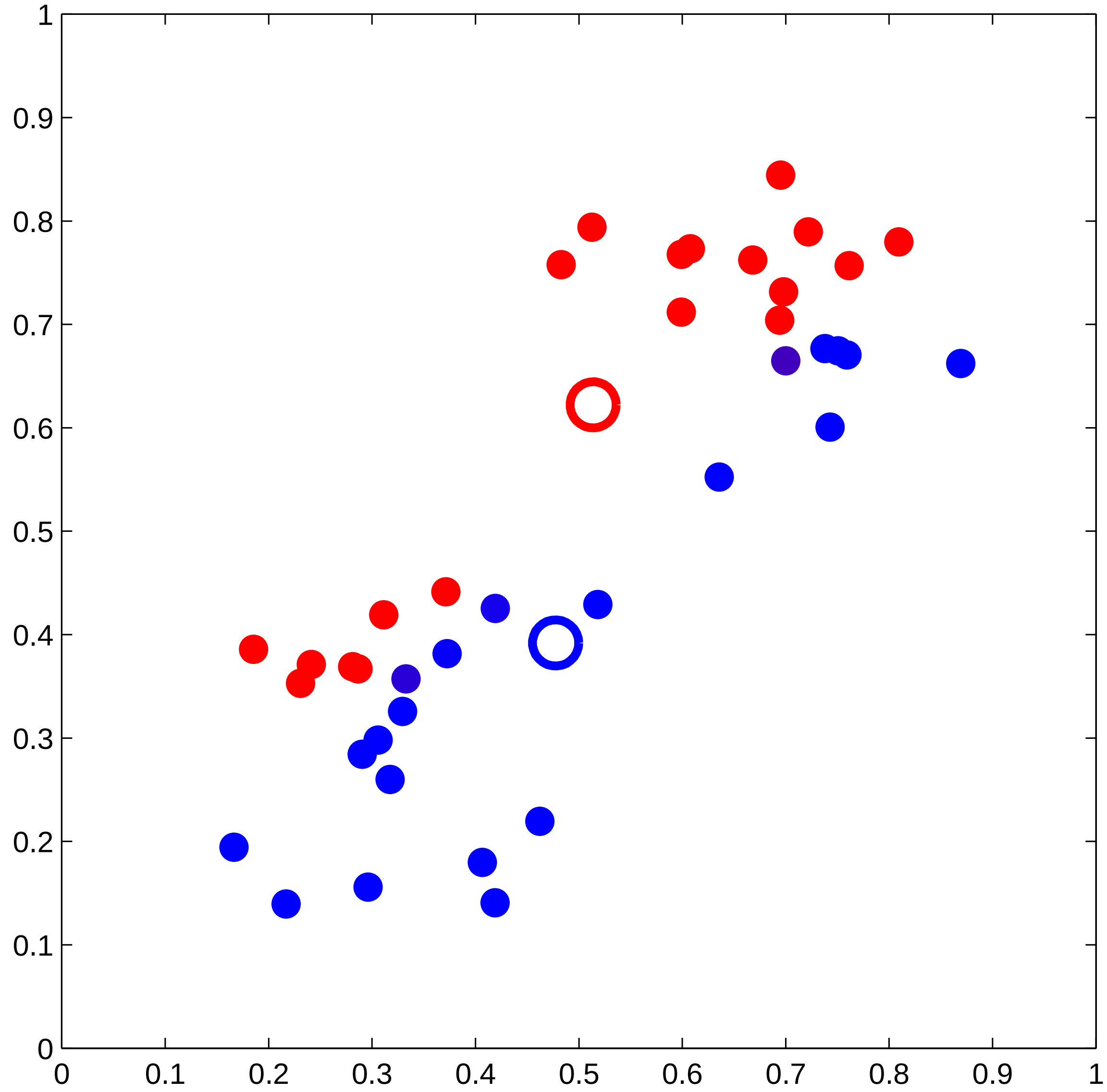
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



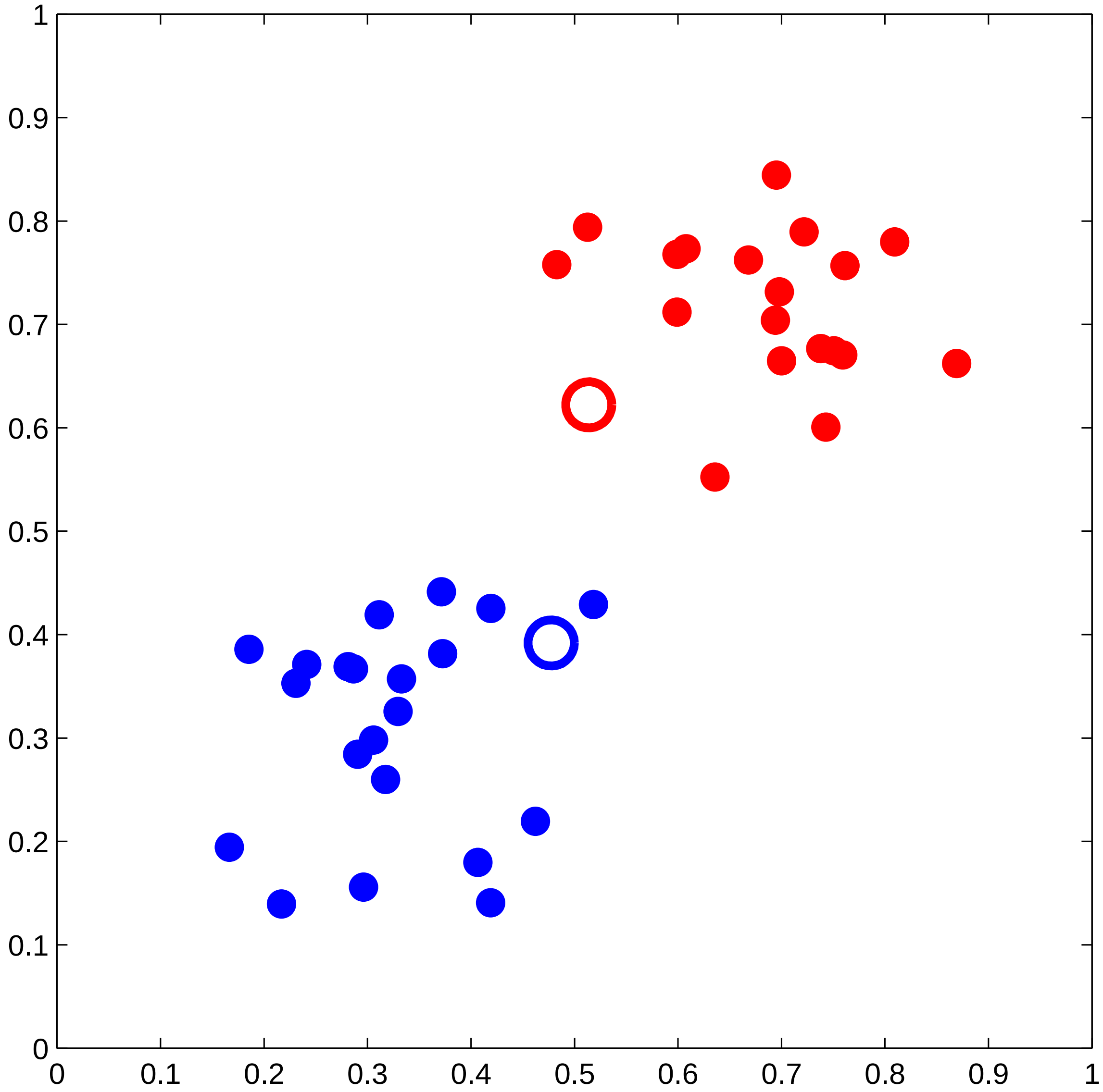
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

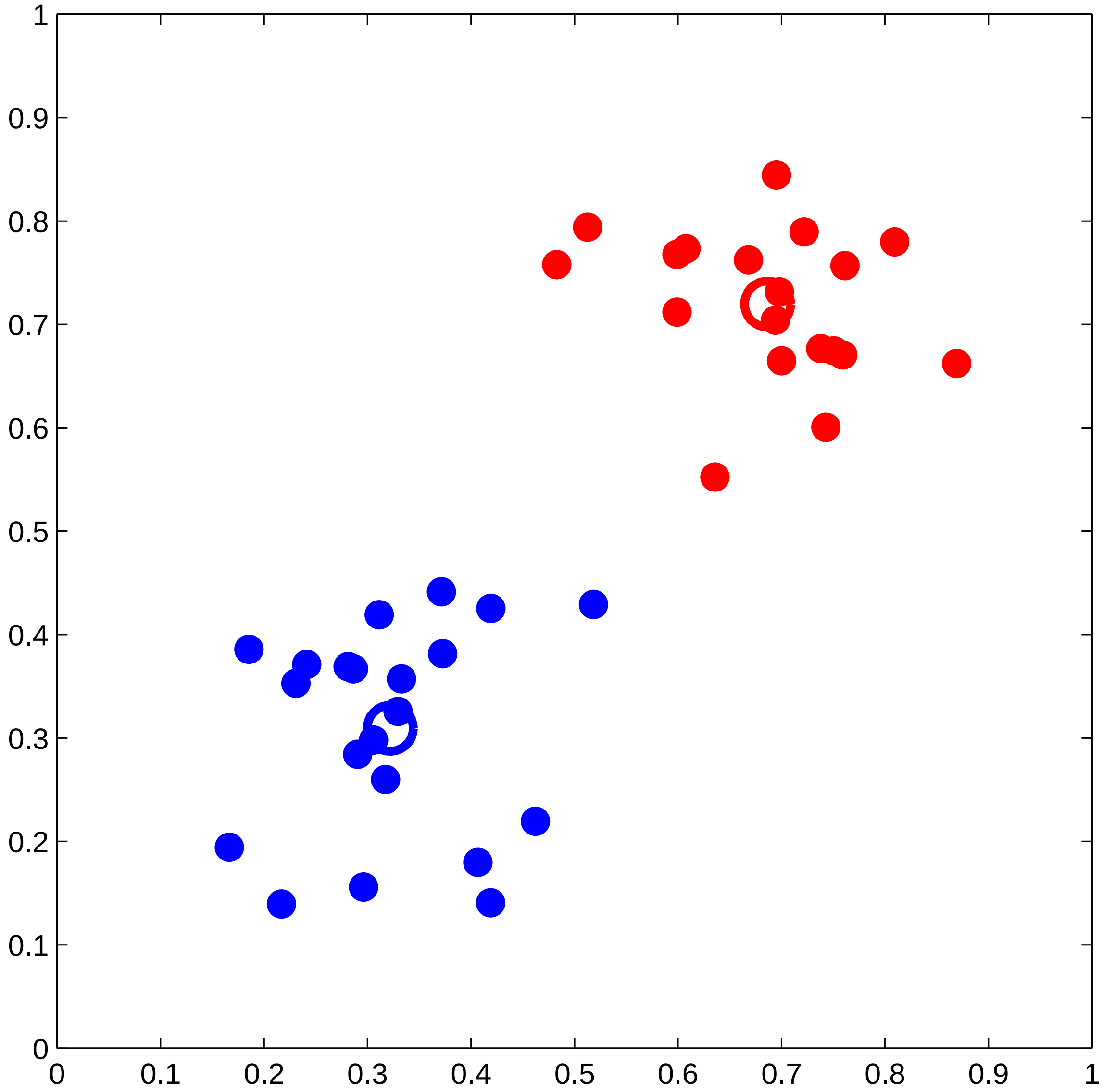


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

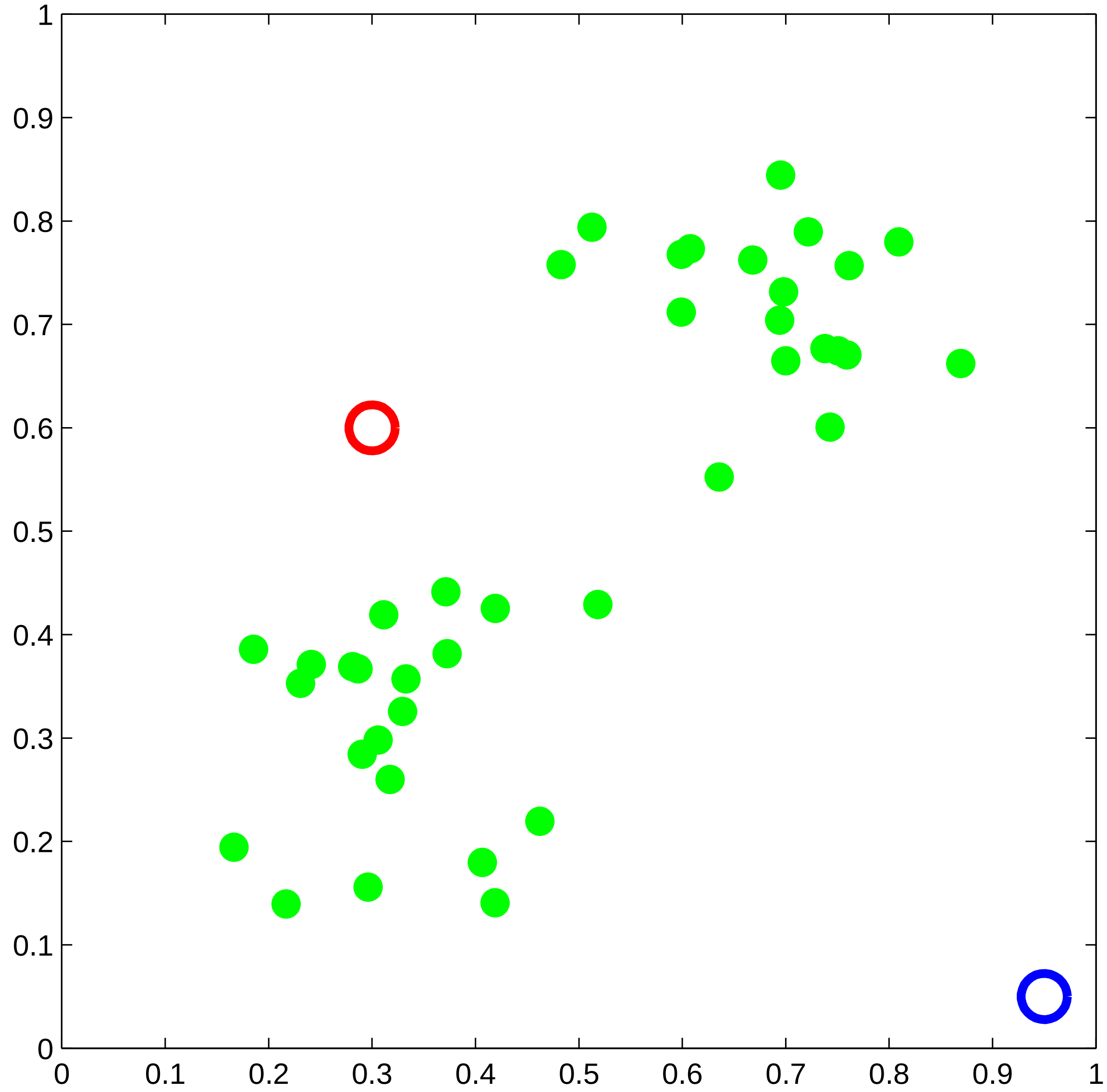


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

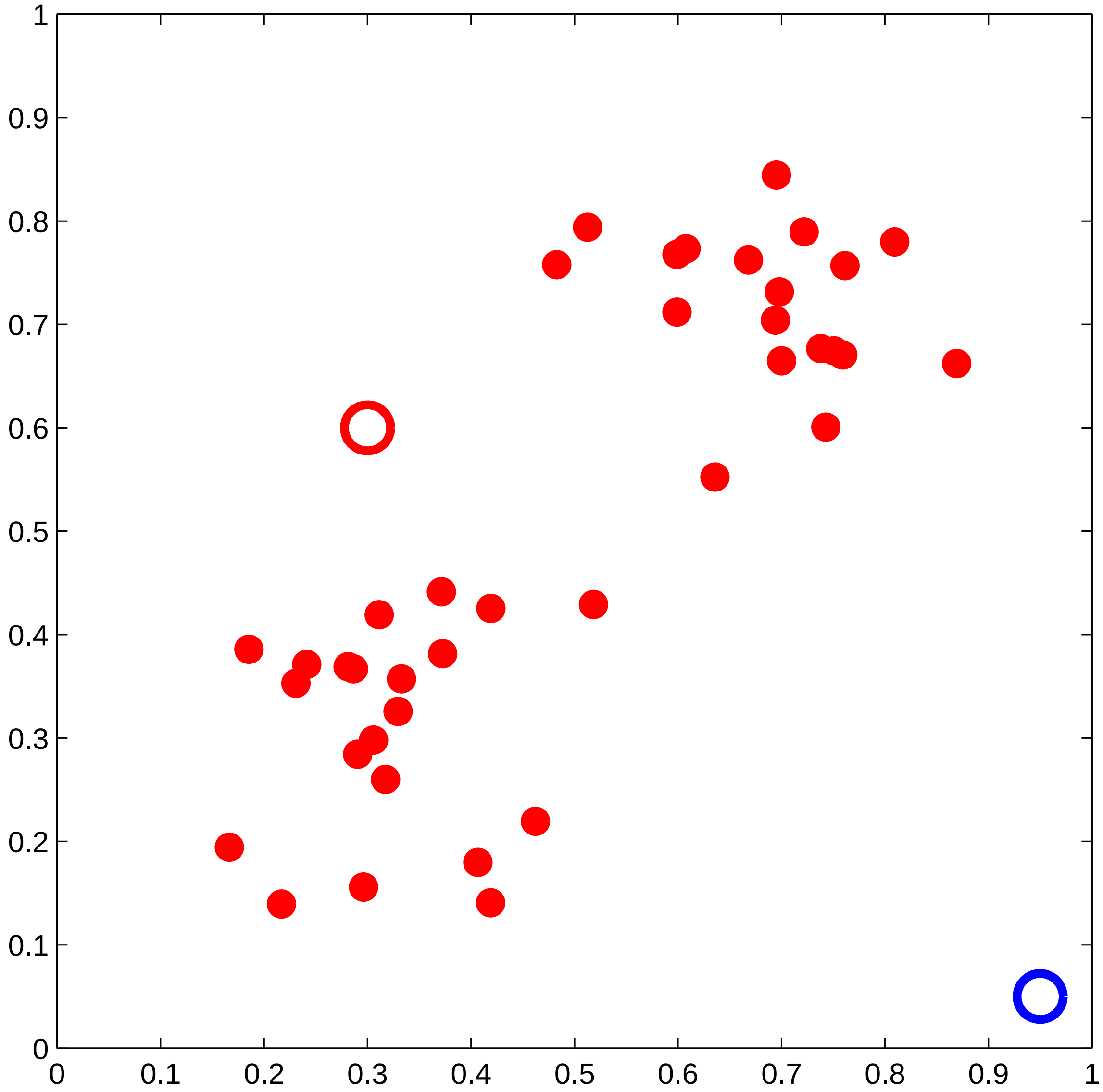

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

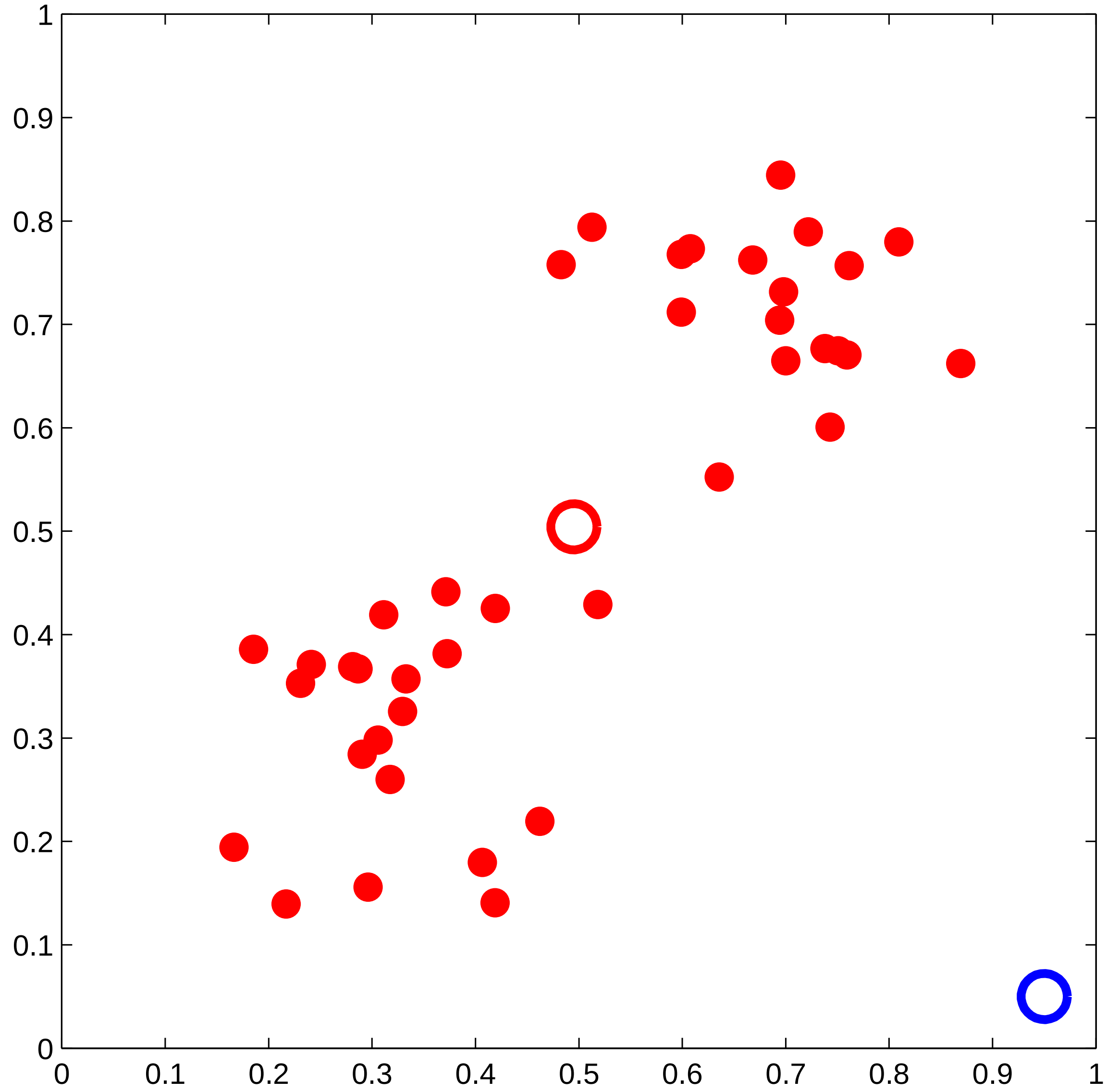
A bad example of K-means clustering



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Shortcoming of K-means clustering

- The algorithm can easily get stuck in suboptimal cluster formations.
- Use fuzzy or soft K-means.

Fuzzy and soft K-means clustering

Fuzzy and soft K-means clustering

- Objective: Soft or fuzzy partition of the data into a predefined number of clusters, K.
  - Each data vector may belong to more than one cluster, according to its degree of membership.
  - This is in contrast to K-means, where a data vector either wholly belongs to a cluster or not.

Fuzzy and soft K-means clustering

- Objective: Soft or fuzzy partition of the data into a predefined number of clusters, K.
  - Each data vector may belong to more than one cluster, according to its degree of membership.
  - This is in contrast to K-means, where a data vector either wholly belongs to a cluster or not.

- Method: Alternatingly update
  - the membership grade for each data vector;
  - the cluster centroids.

- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.

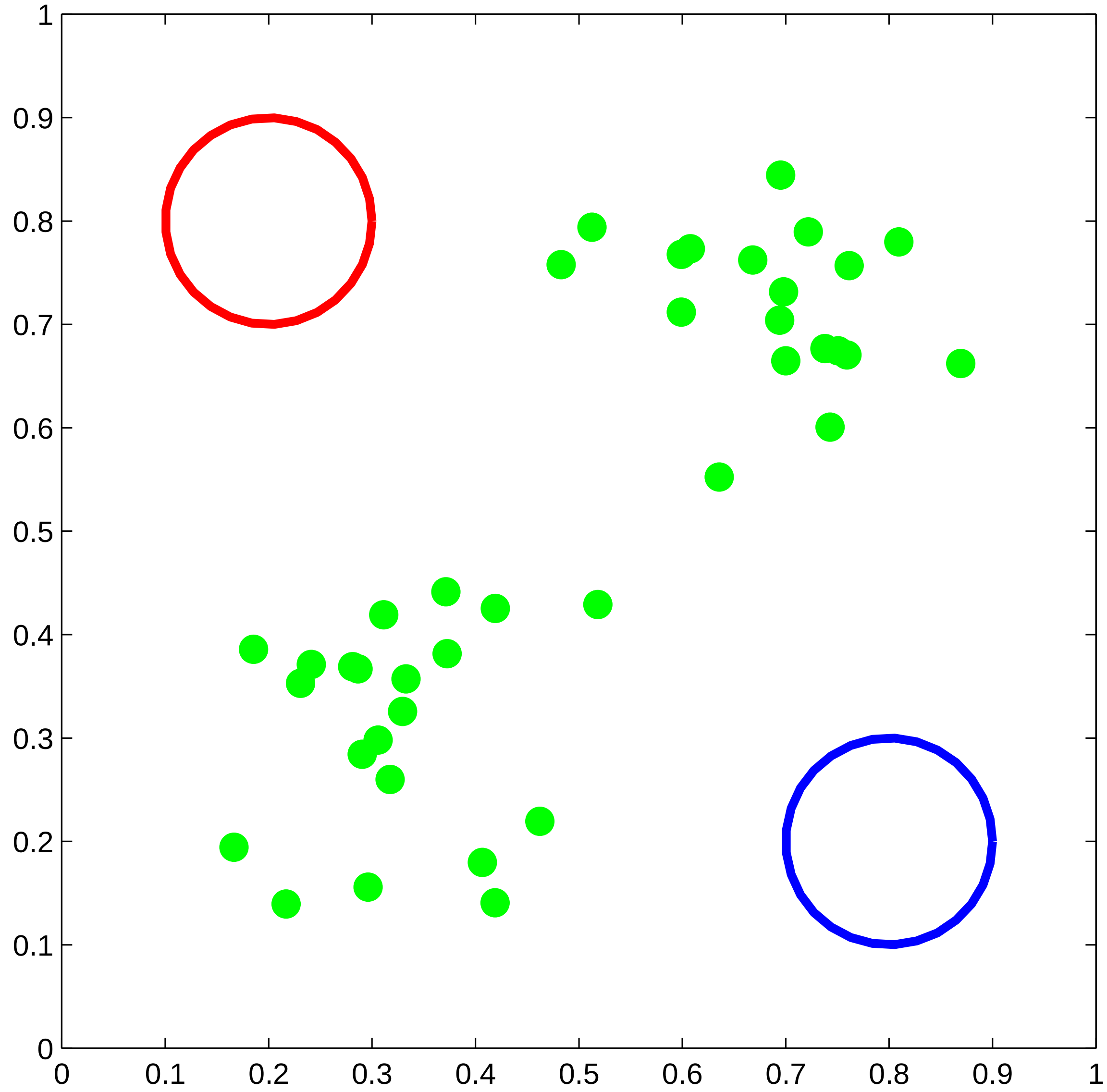
- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.
- Iteration (until membership grades remain unchanged):
  - For all data vectors $\mathbf{x}_i$, $i = 1, \ldots, N$, and all centroids $\mathbf{c}_k$, $k = 1, \ldots, K$: Compute the distance d_{ik} between the data vector $\mathbf{x}_i$ and the centroid $\mathbf{c}_k$.

- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.
- Iteration (until membership grades remain unchanged):
  - For all data vectors $\mathbf{x}_i$, $i = 1, \ldots, N$, and all centroids $\mathbf{c}_k$, $k = 1, \ldots, K$: Compute the distance d_{ik} between the data vector $\mathbf{x}_i$ and the centroid $\mathbf{c}_k$.
  - Compute the membership grades λ_{ik}. Note: $\lambda_{ik} \geq 0$ indicates the amount of association of data vector $\mathbf{x}_i$ with centroid $\mathbf{c}_k$ and depends on the distance d_{ik}: if $d_{ik} < d_{ik'}$, then $\lambda_{ik} > \lambda_{ik'}$. The detailed functional form (omitted) differs between soft and fuzzy K-means.

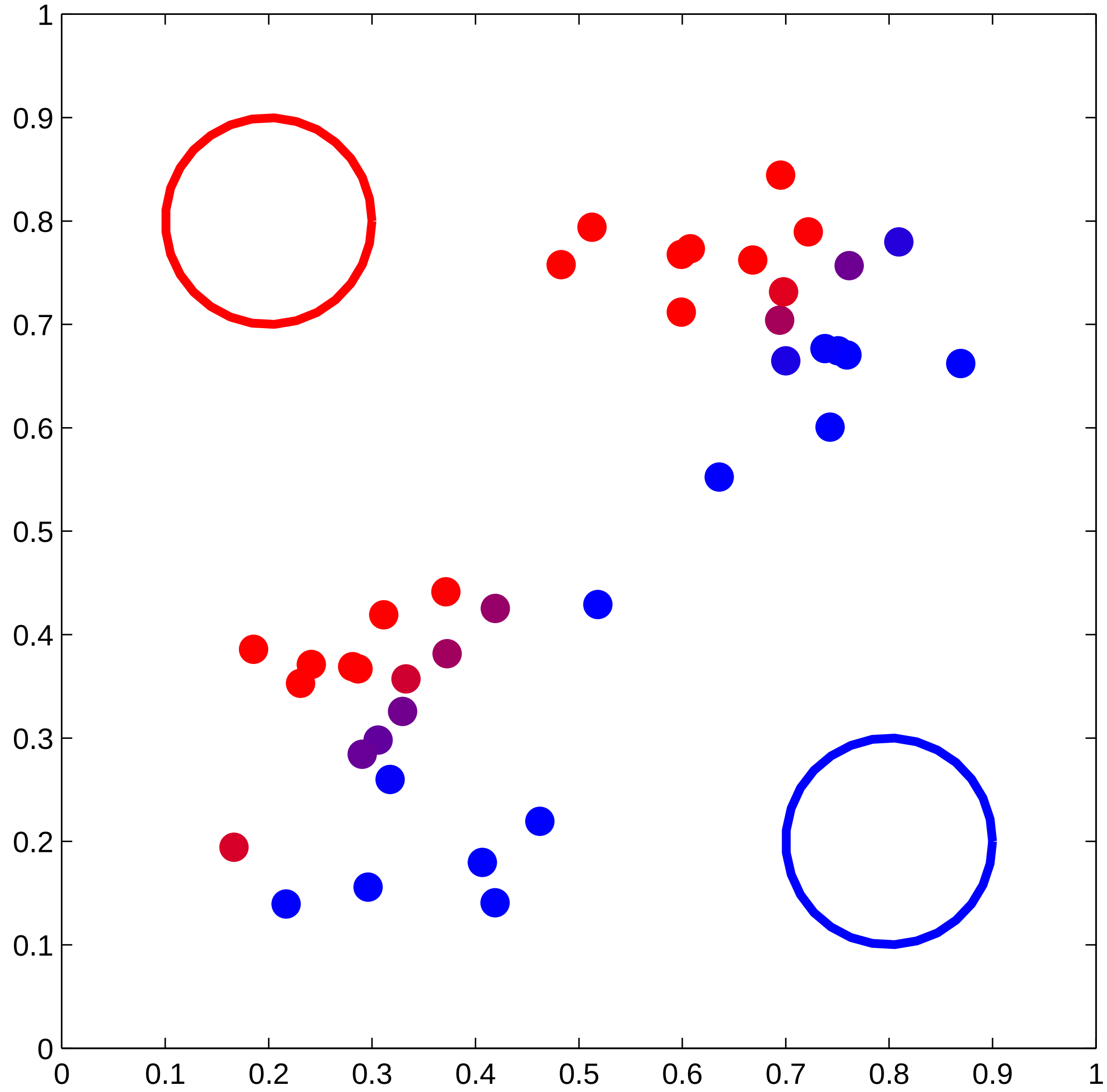
- Decide on the number of clusters, K.
- Start with a set of cluster centroids: $\mathbf{c}_1, \ldots, \mathbf{c}_K$.
- Iteration (until membership grades remain unchanged):
  - For all data vectors $\mathbf{x}_i$, $i = 1, \ldots, N$, and all centroids $\mathbf{c}_k$, $k = 1, \ldots, K$: Compute the distance d_{ik} between the data vector $\mathbf{x}_i$ and the centroid $\mathbf{c}_k$.
  - Compute the membership grades λ_{ik}. Note: $\lambda_{ik} \geq 0$ indicates the amount of association of data vector $\mathbf{x}_i$ with centroid $\mathbf{c}_k$ and depends on the distance d_{ik}: if $d_{ik} < d_{ik'}$, then $\lambda_{ik} > \lambda_{ik'}$. The detailed functional form (omitted) differs between soft and fuzzy K-means.
  - Recompute the cluster centroids: $\mathbf{c}_k = \frac{\sum_i \lambda_{ik} \mathbf{x}_i}{\sum_i \lambda_{ik}}$

Two examples of soft K-means clustering

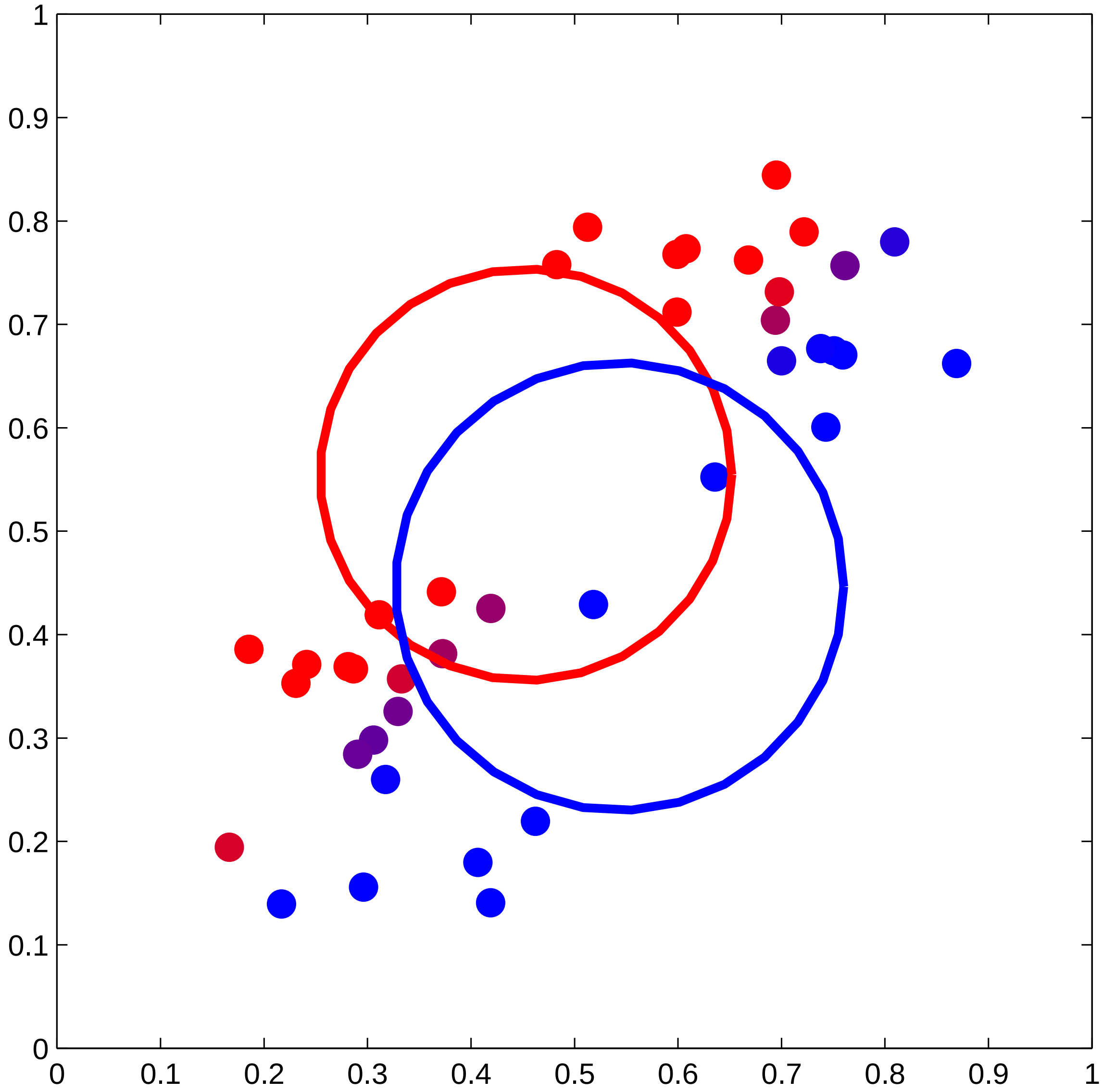
The posterior probability for a given data point is indicated by a colour scale ranging from pure red (corresponding to a posterior probability of 1.0 for the red component and 0.0 for the blue component) through to pure blue.



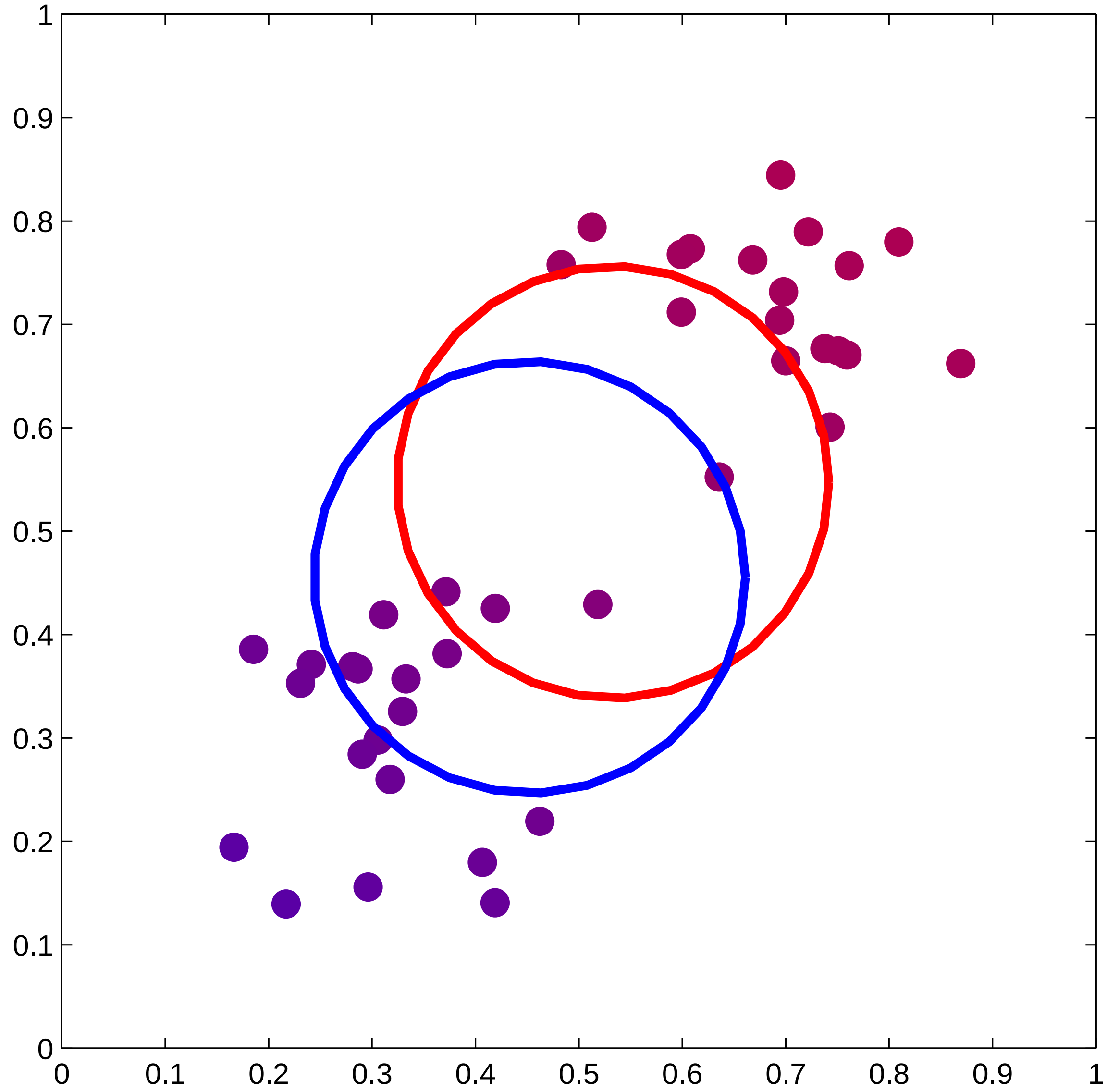
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



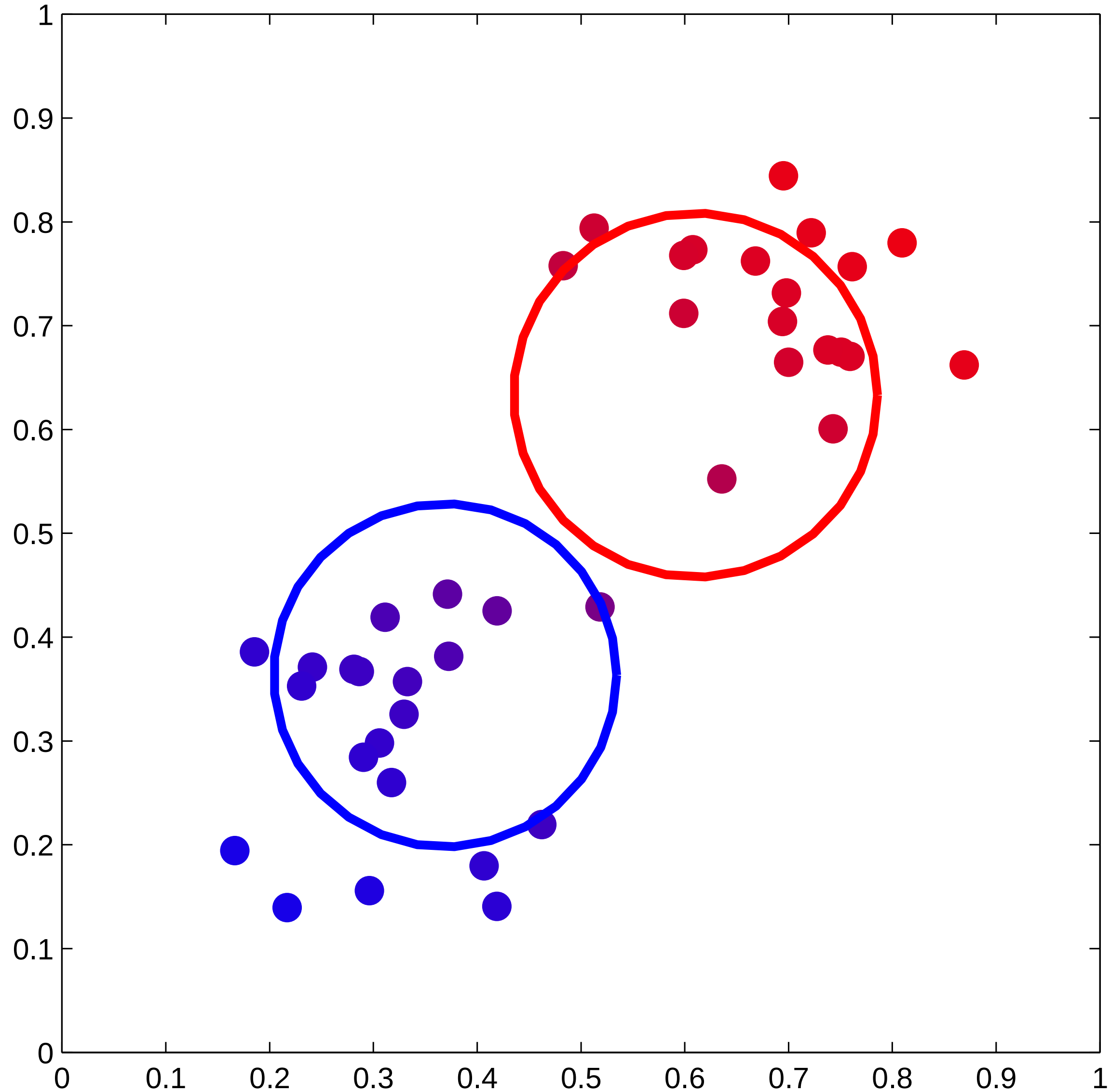
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

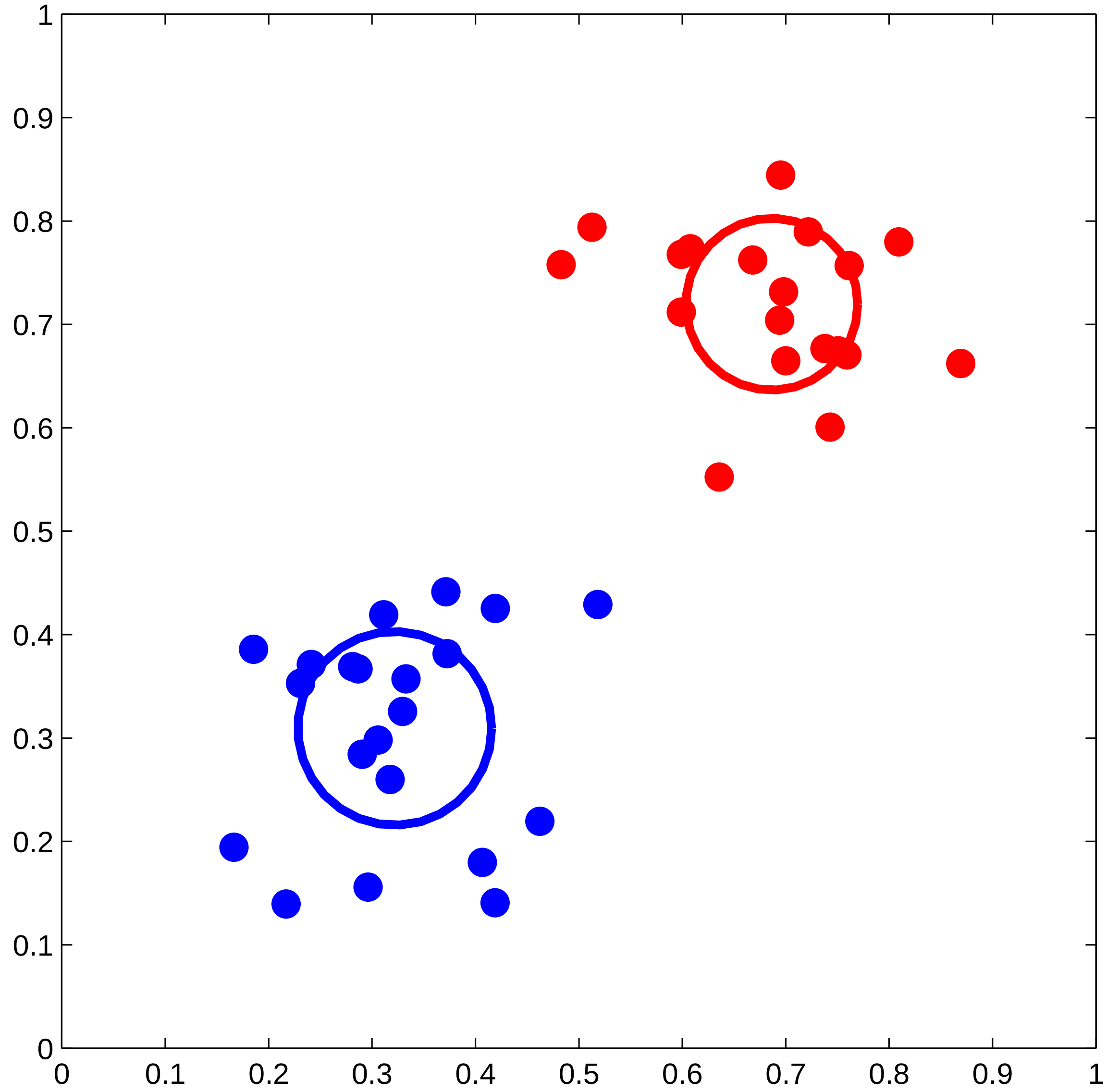


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



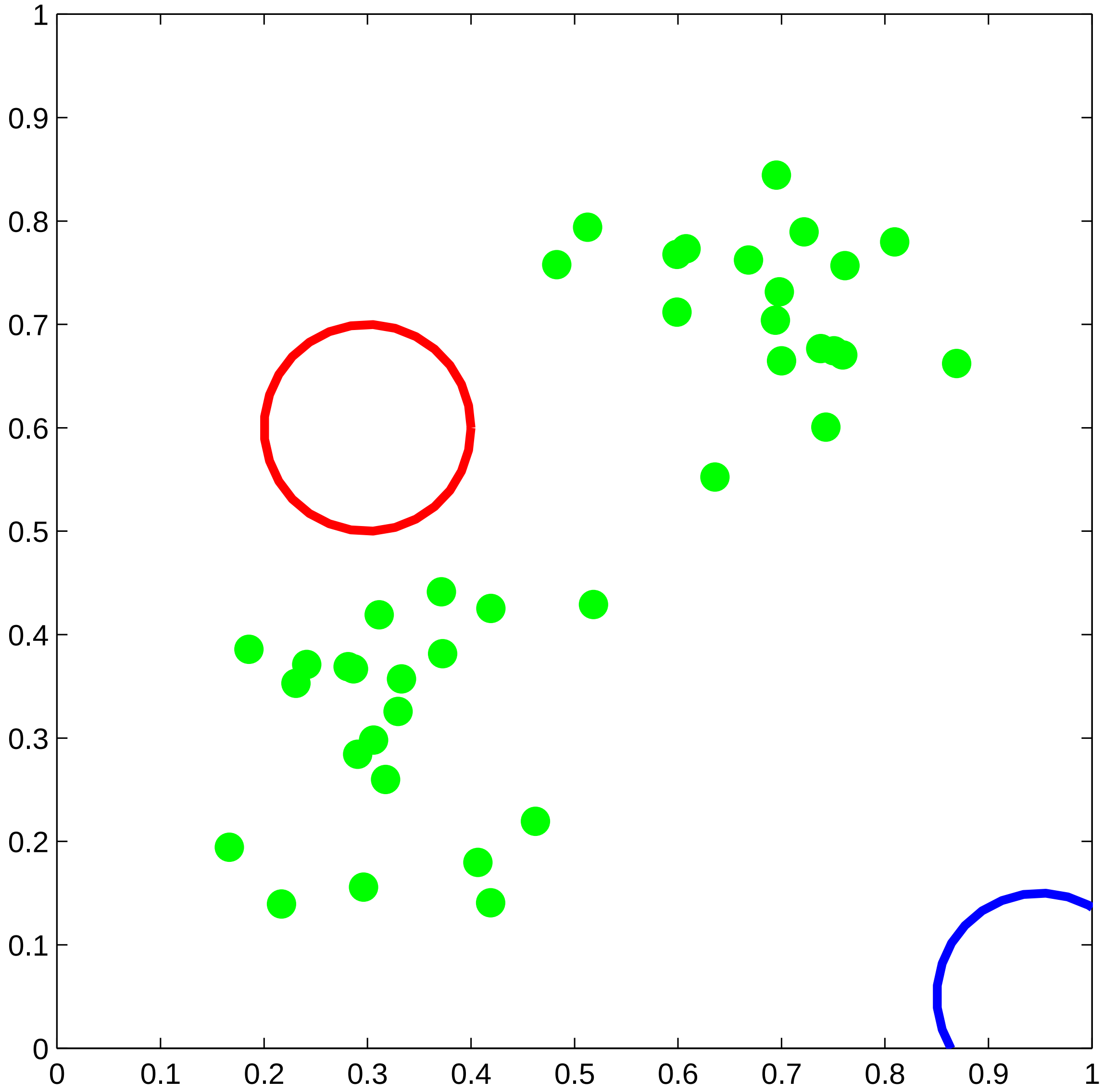
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

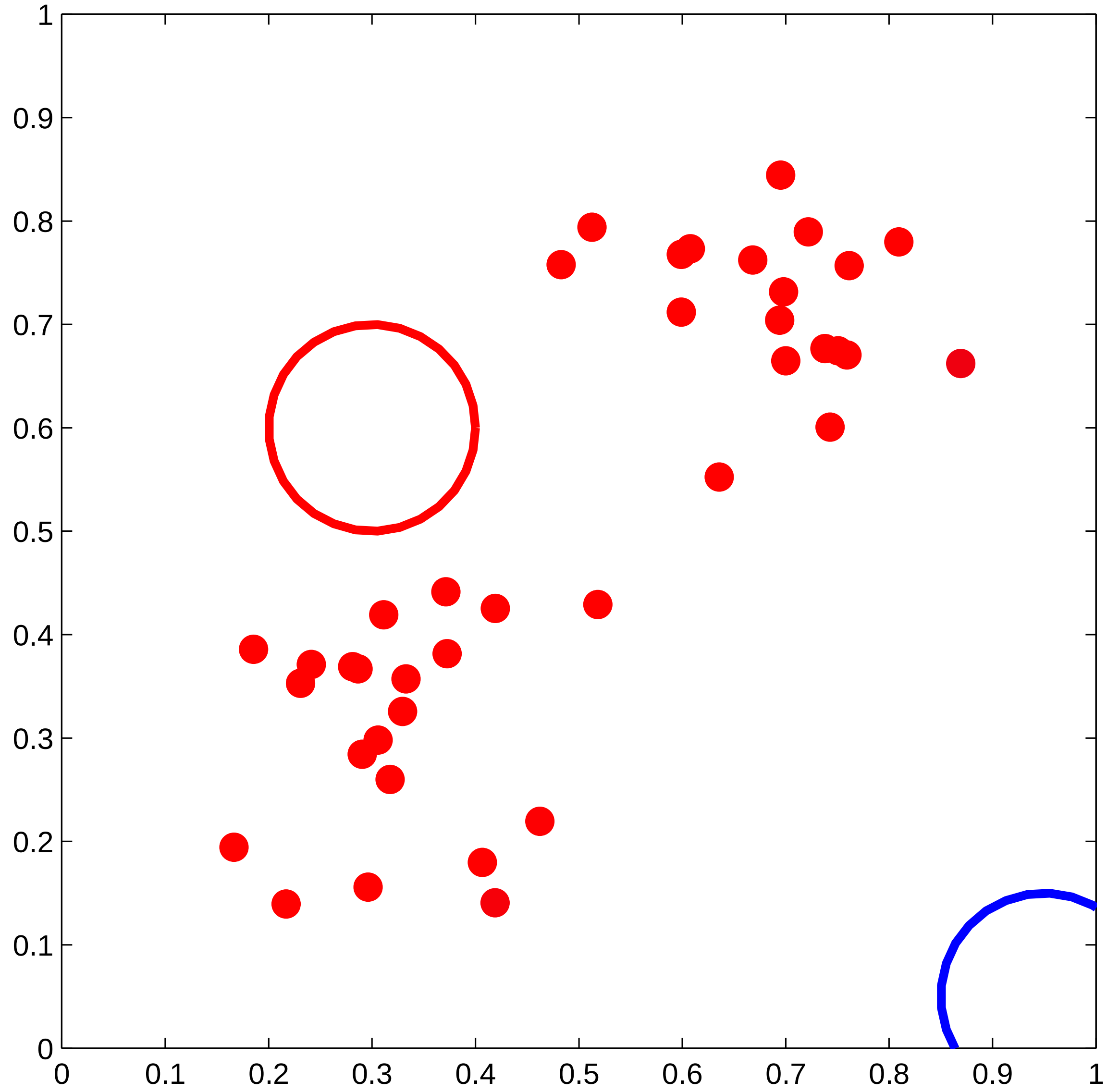


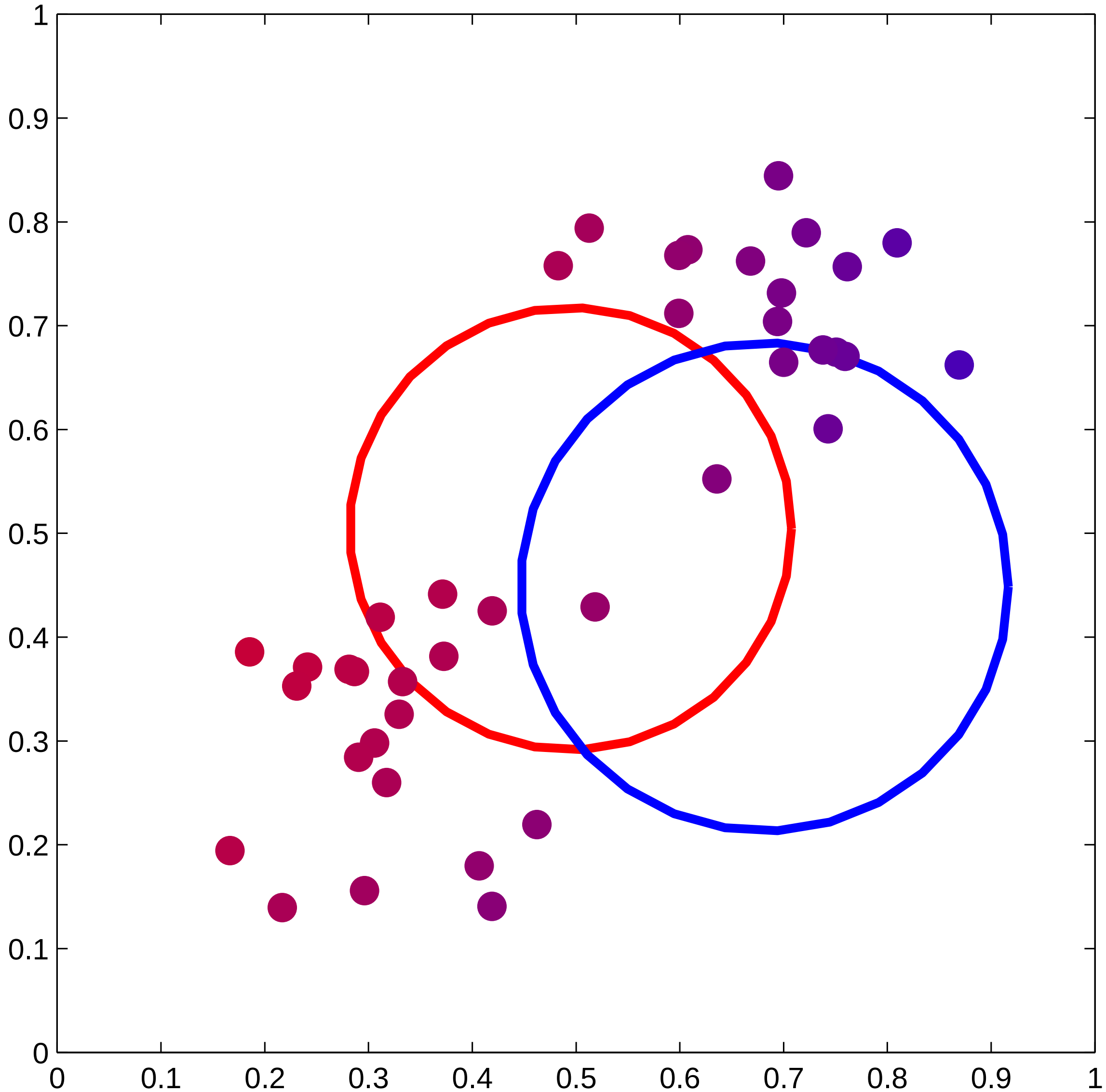
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Initialization for which K-means failed

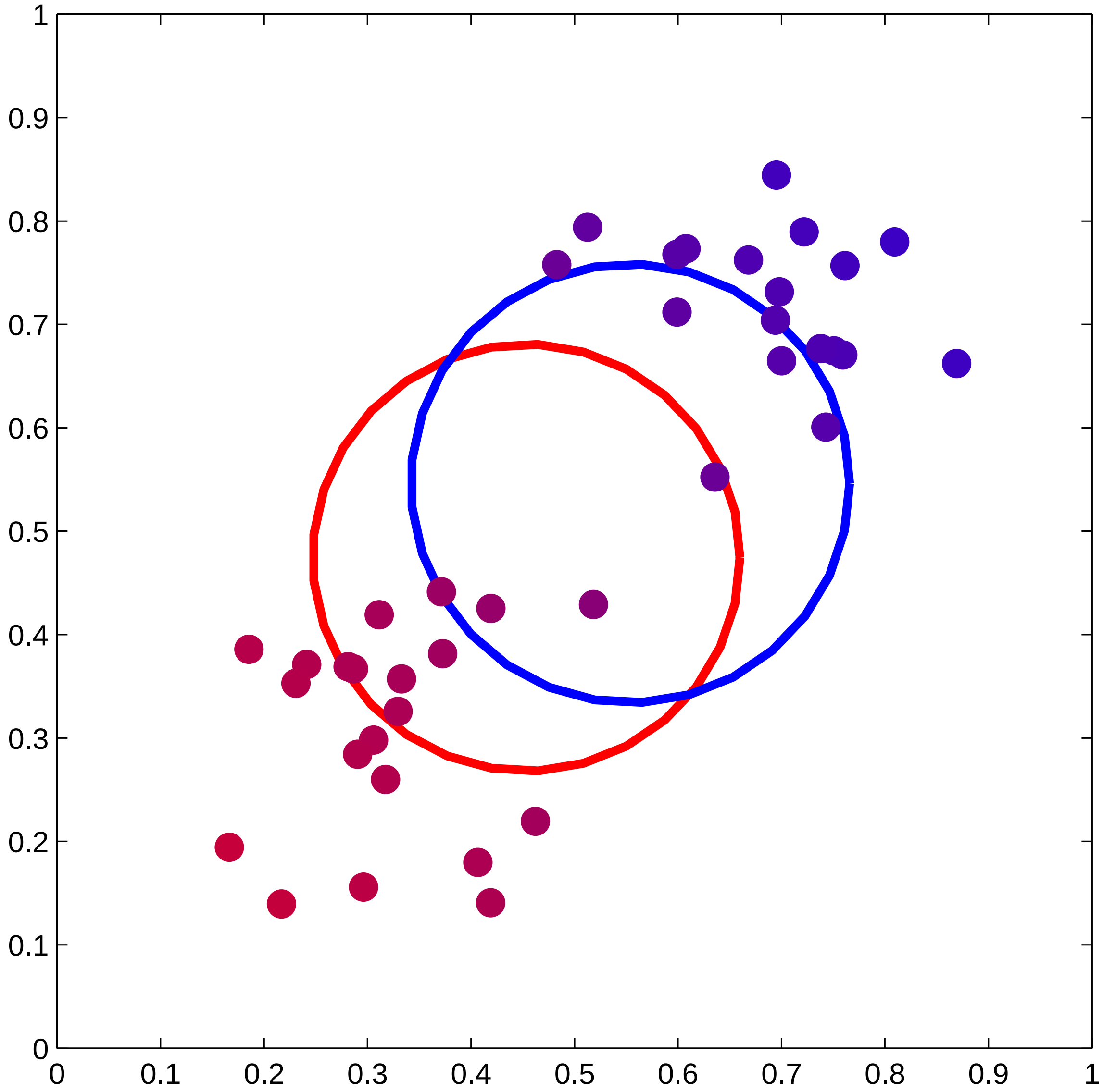


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

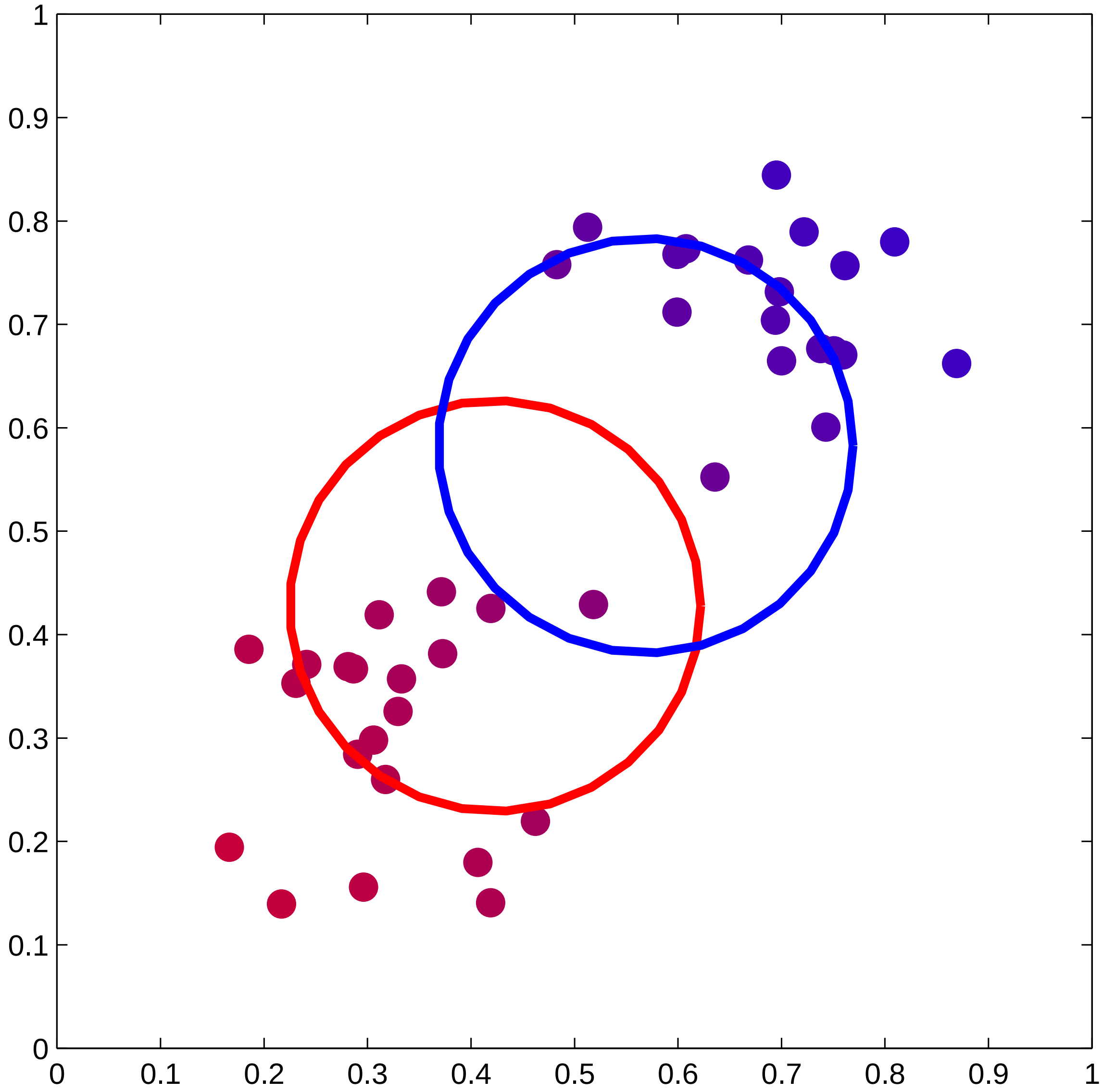

0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



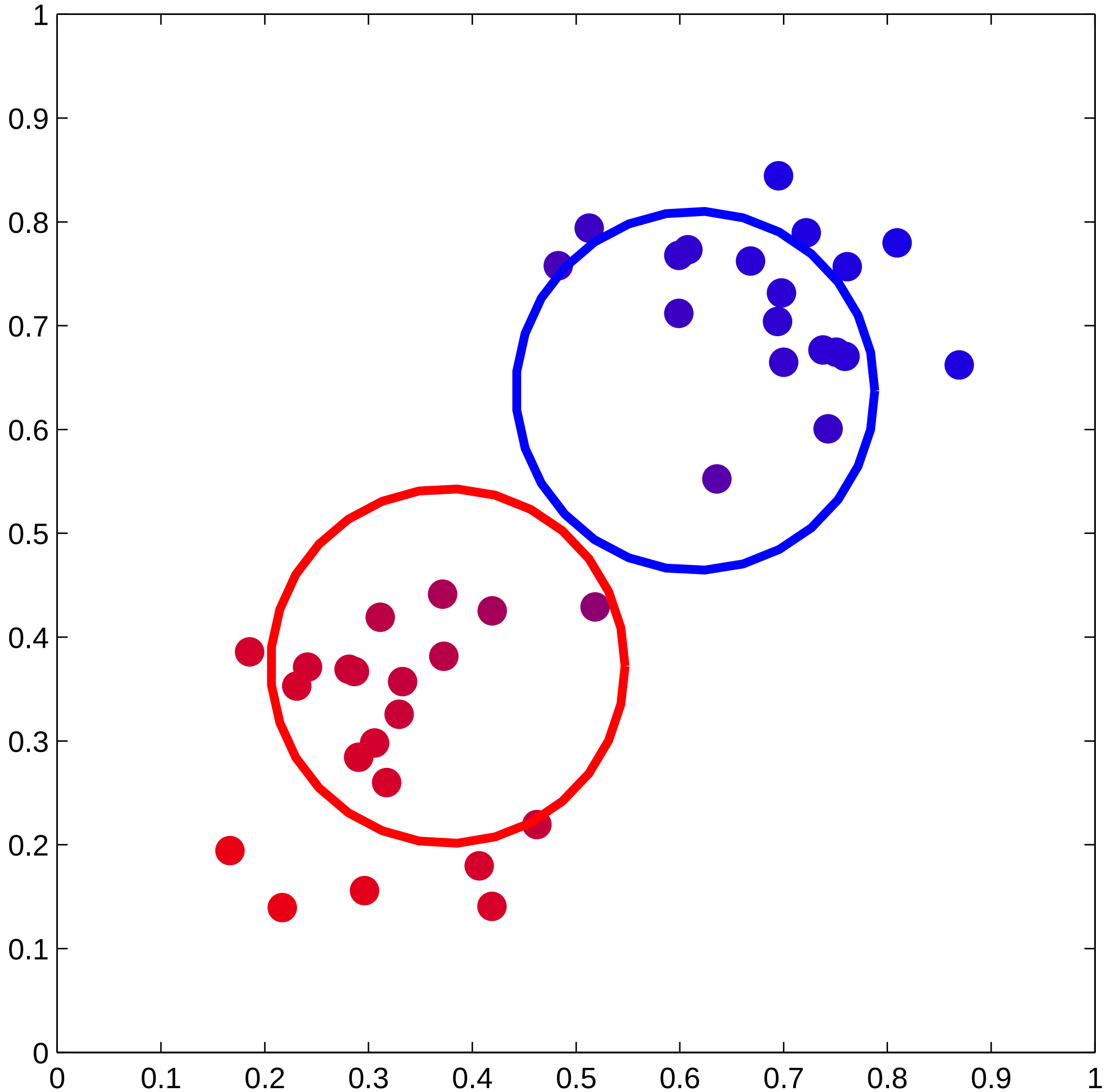
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



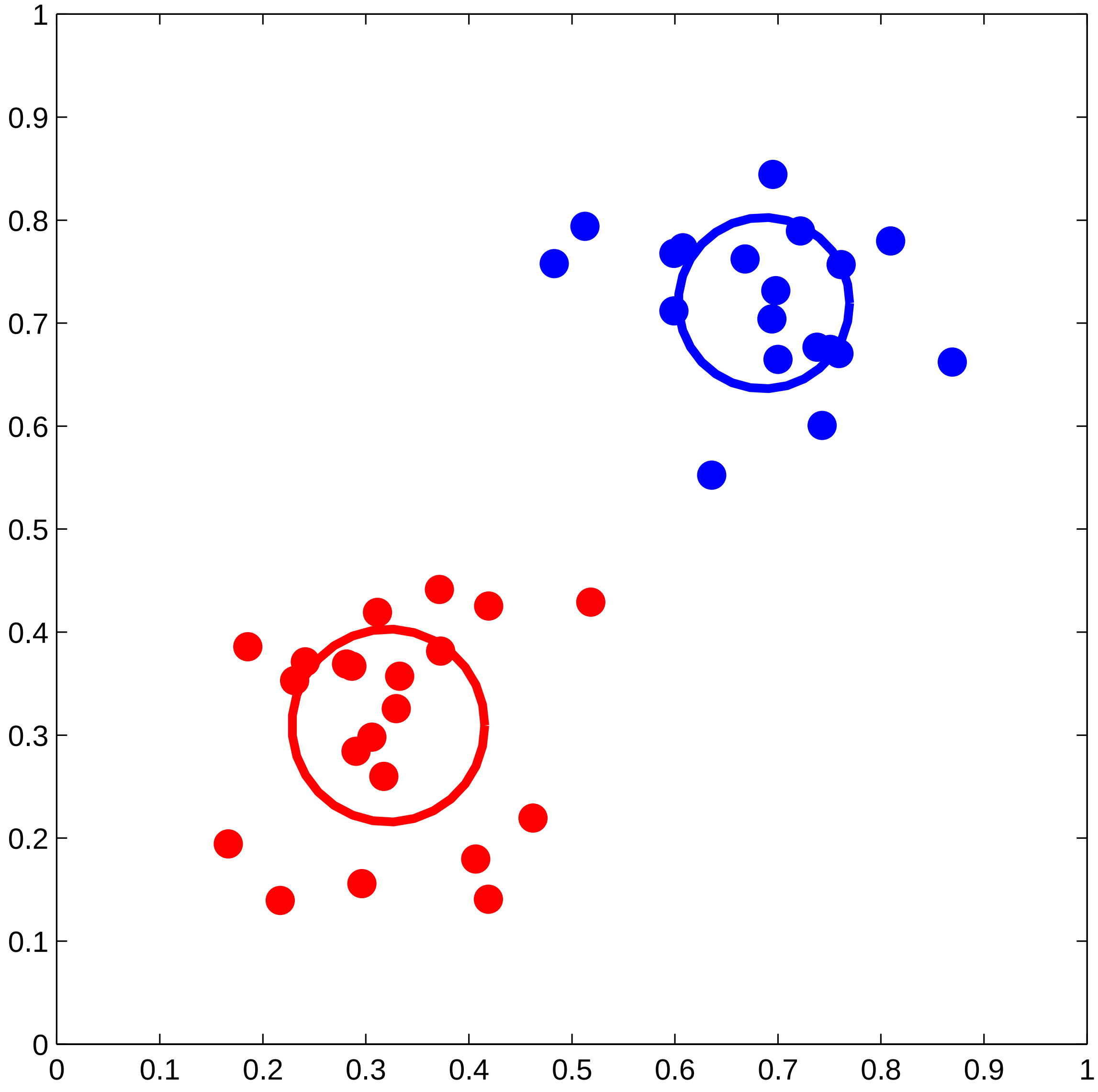
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1



1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

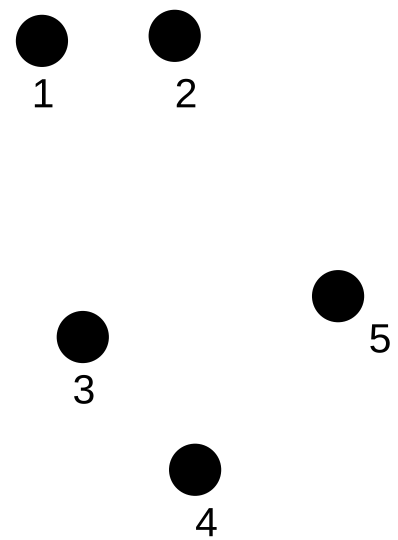


0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

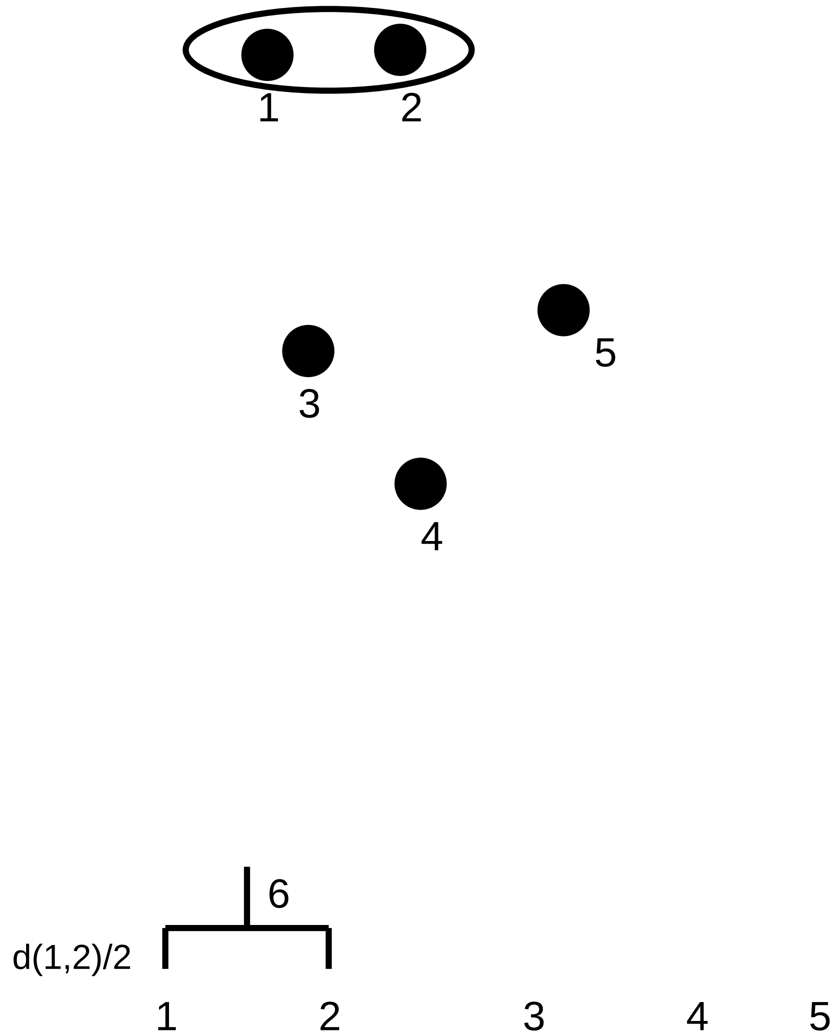


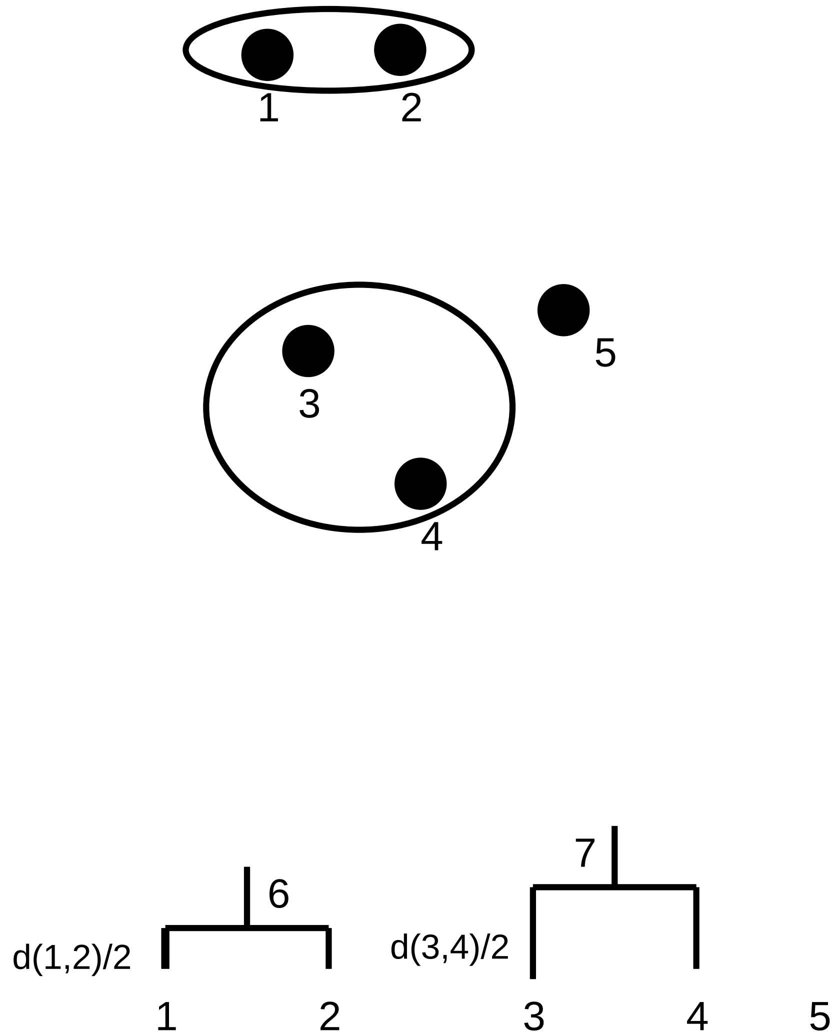
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Agglomerative hierarchical clustering: UPGMA (hierarchical average linkage clustering)

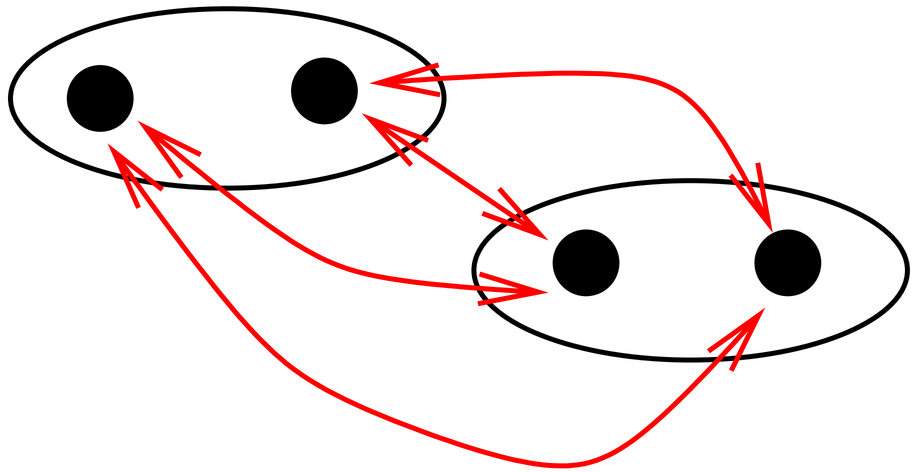


1 2 3 4 5

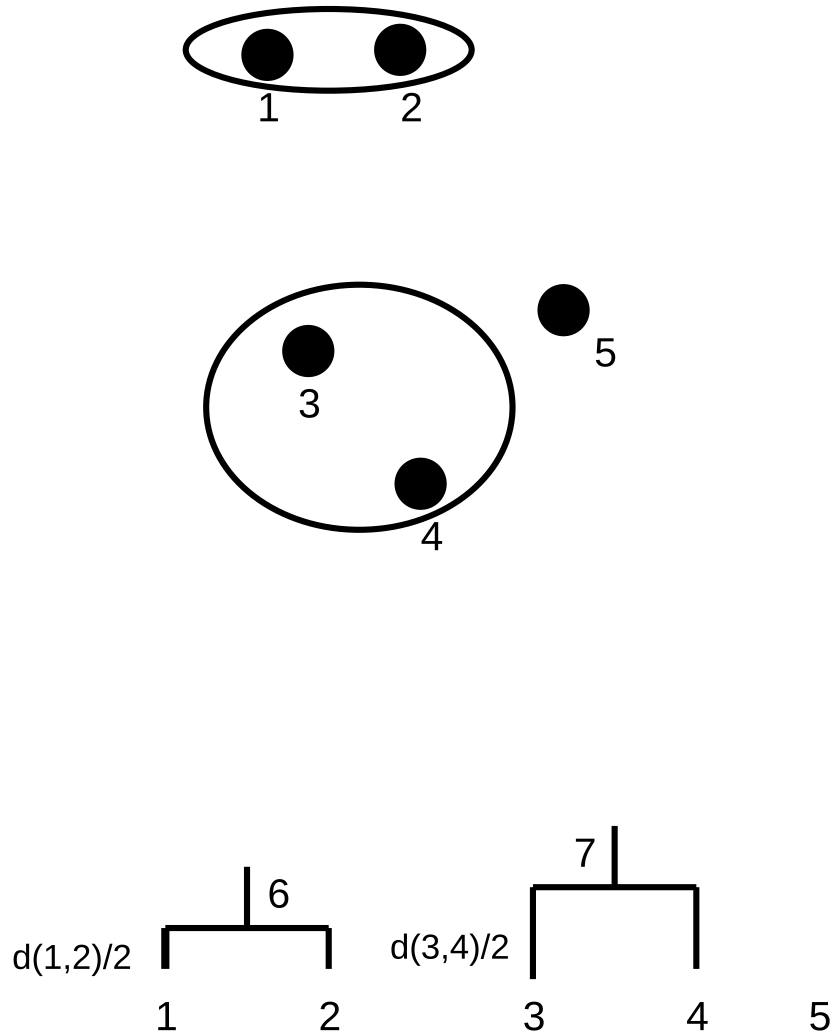

1
2
5
3
4
6
d(1,2)/2
1
2
3
4
5

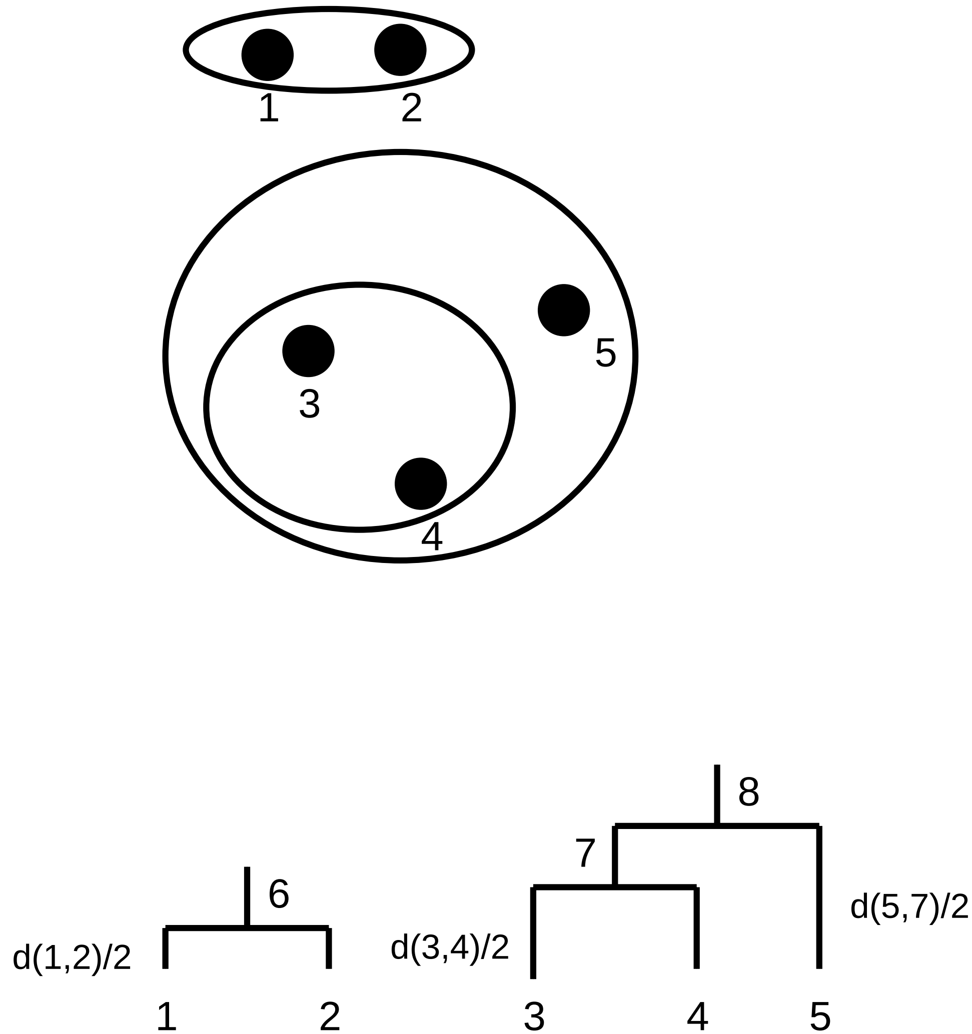

1
2
3
4
5
6
7
d(1,2)/2
d(3,4)/2
1
2
3
4
5

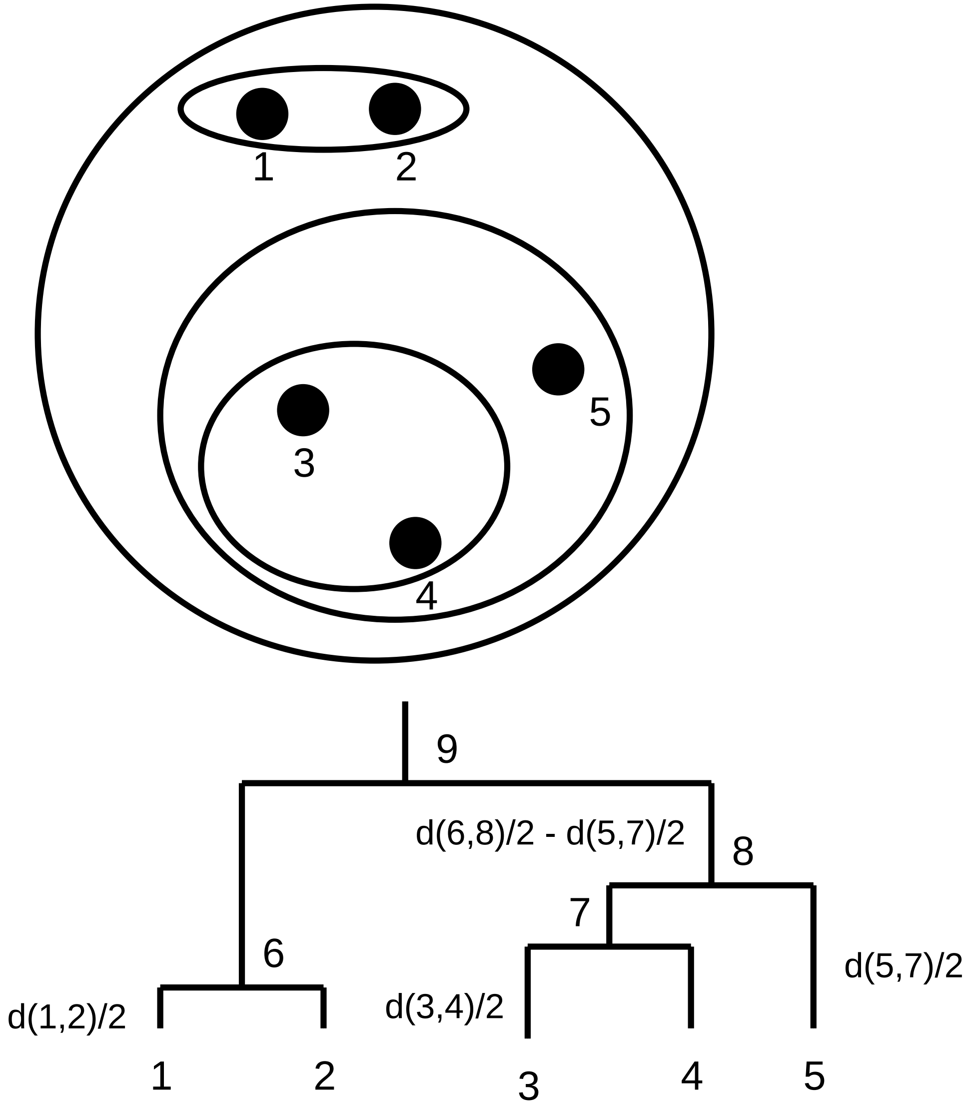
Distance between clusters

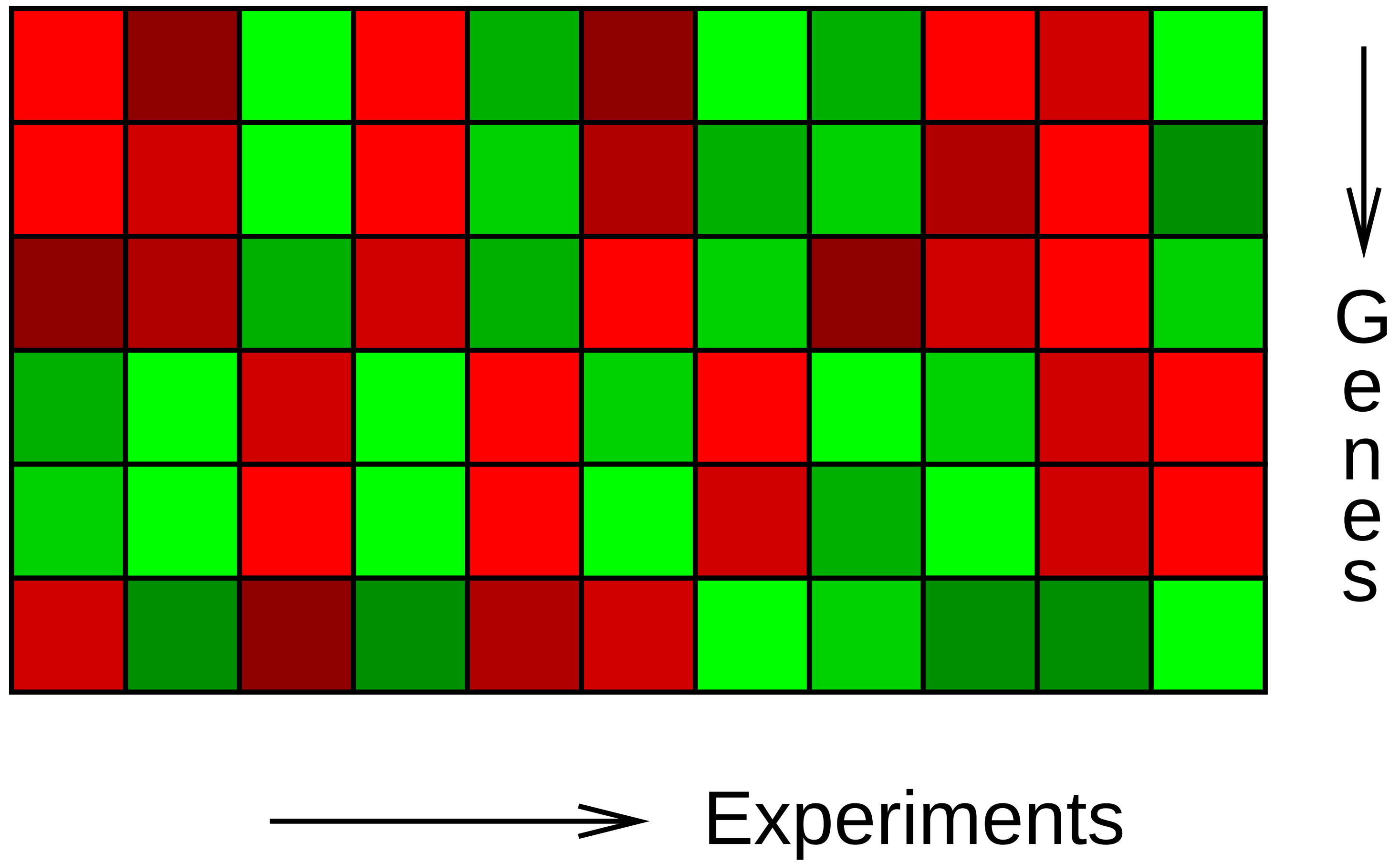


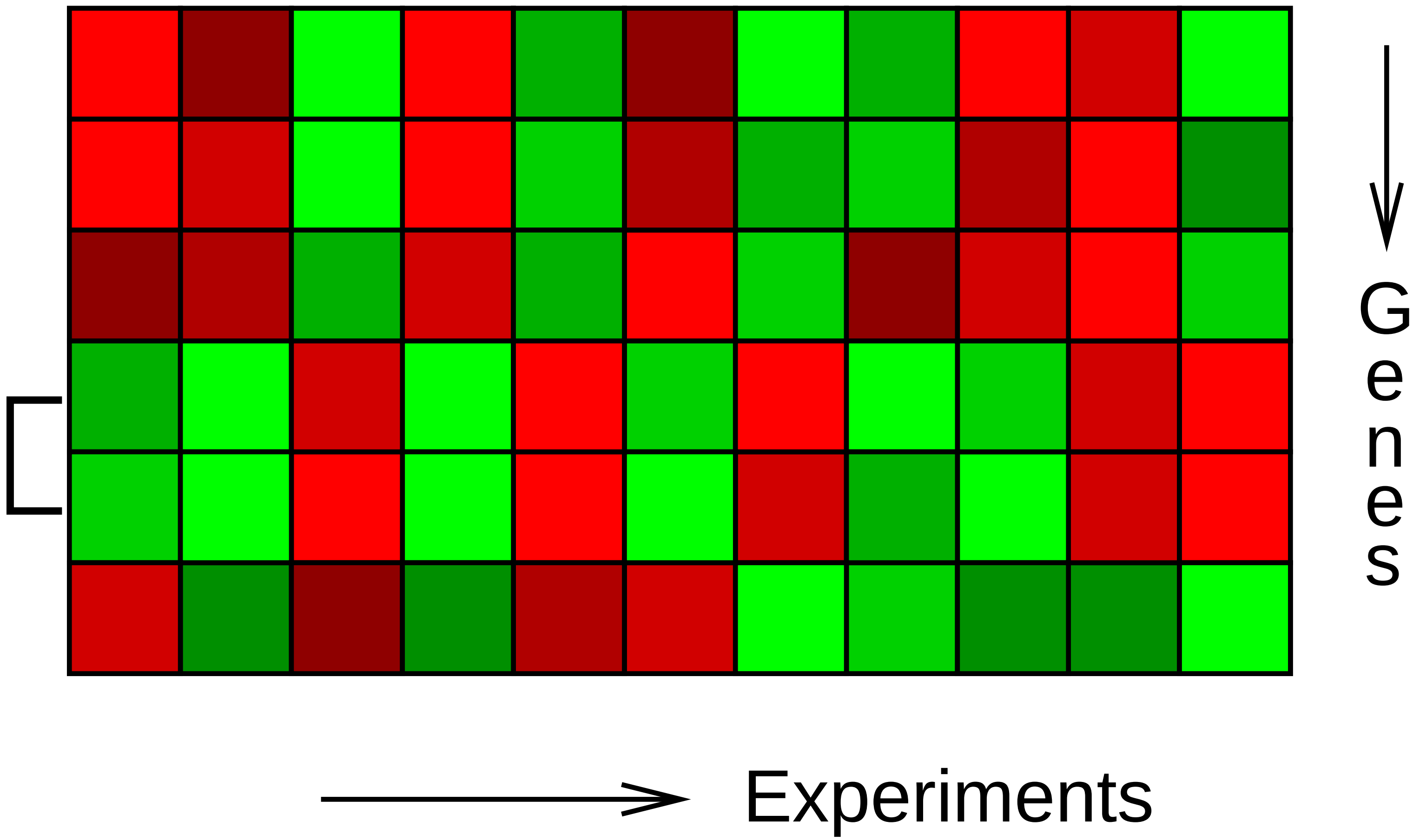
Average of individual distances.

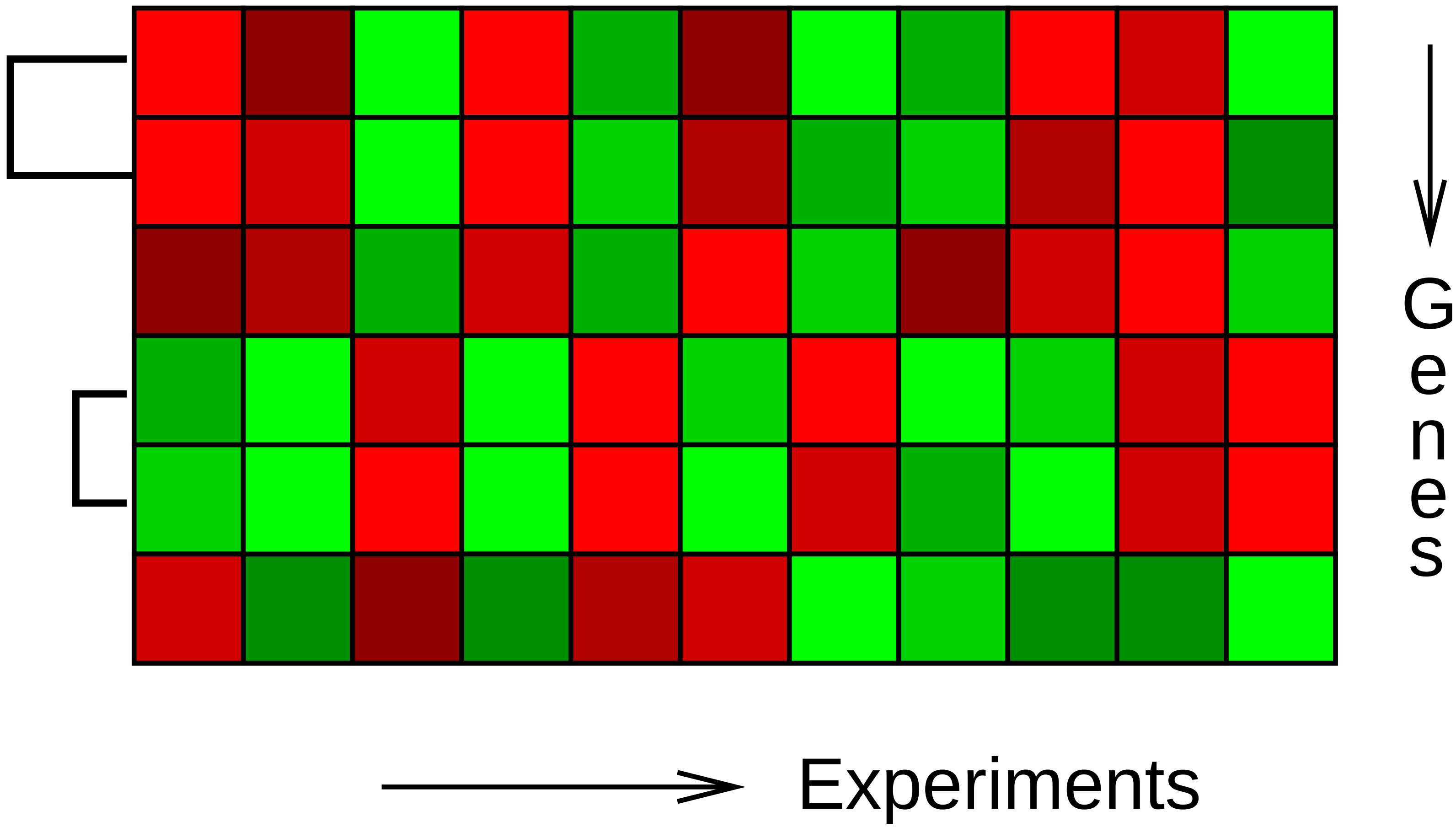

1
2
3
4
5
6
7
d(1,2)/2
d(3,4)/2
1
2
3
4
5

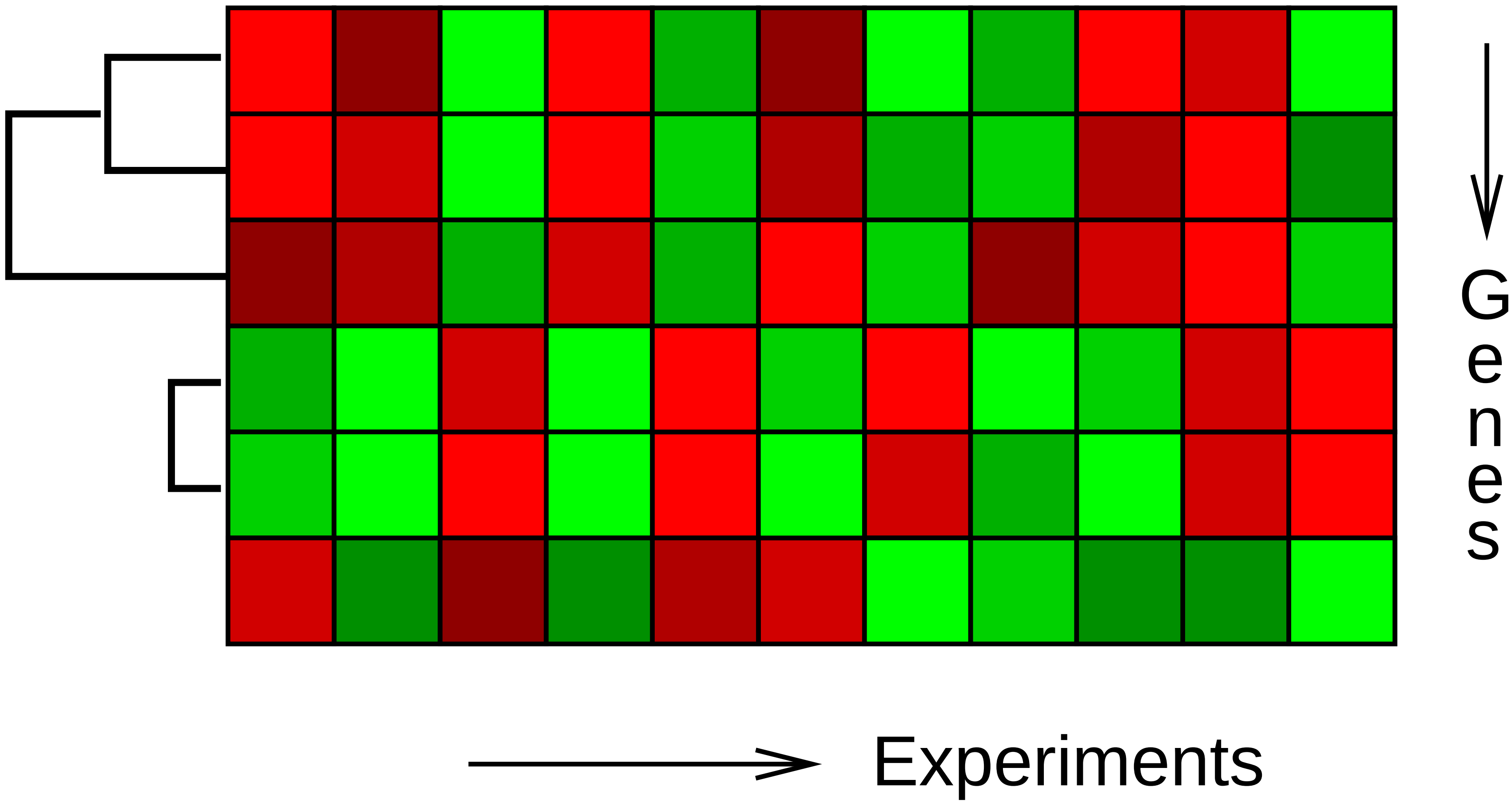

1
2
3
4
5
6
7
8
d(1,2)/2
d(3,4)/2
d(5,7)/2

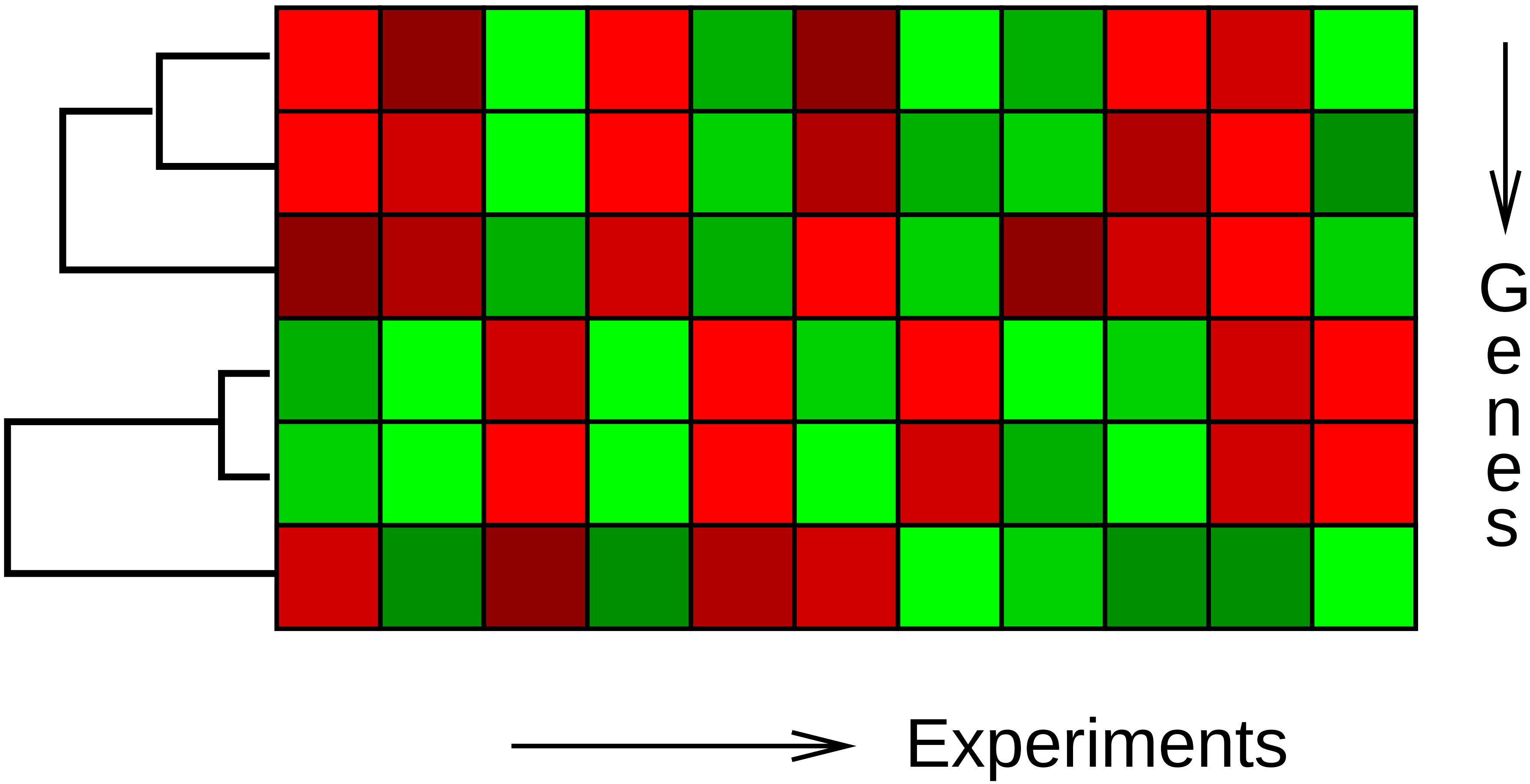

1
2
3
4
5
9
d(6,8)/2 - d(5,7)/2
8
7
6
d(5,7)/2
d(3,4)/2
d(1,2)/2
1
2
3
4
5

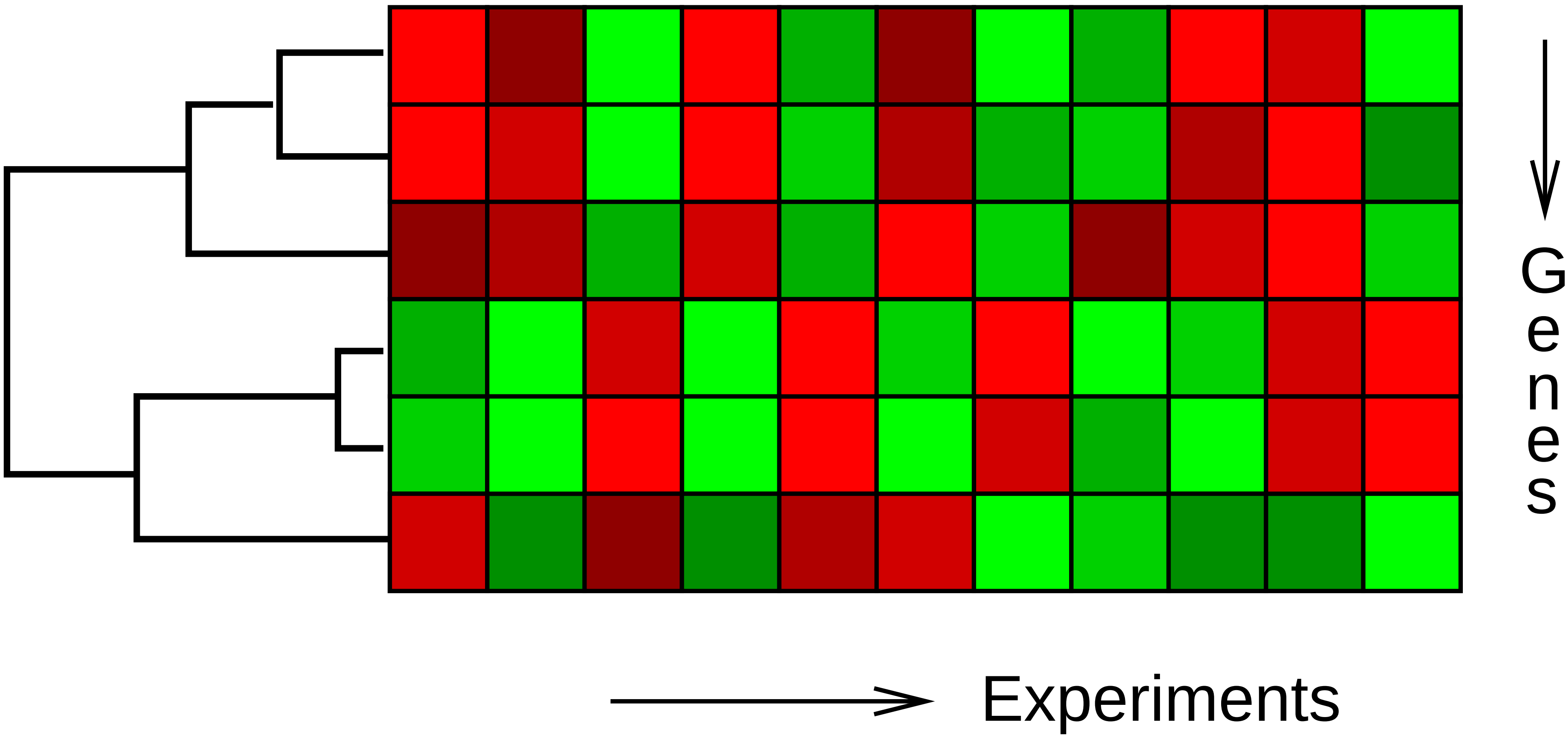

Genes
Experiments


Genes
Experiments

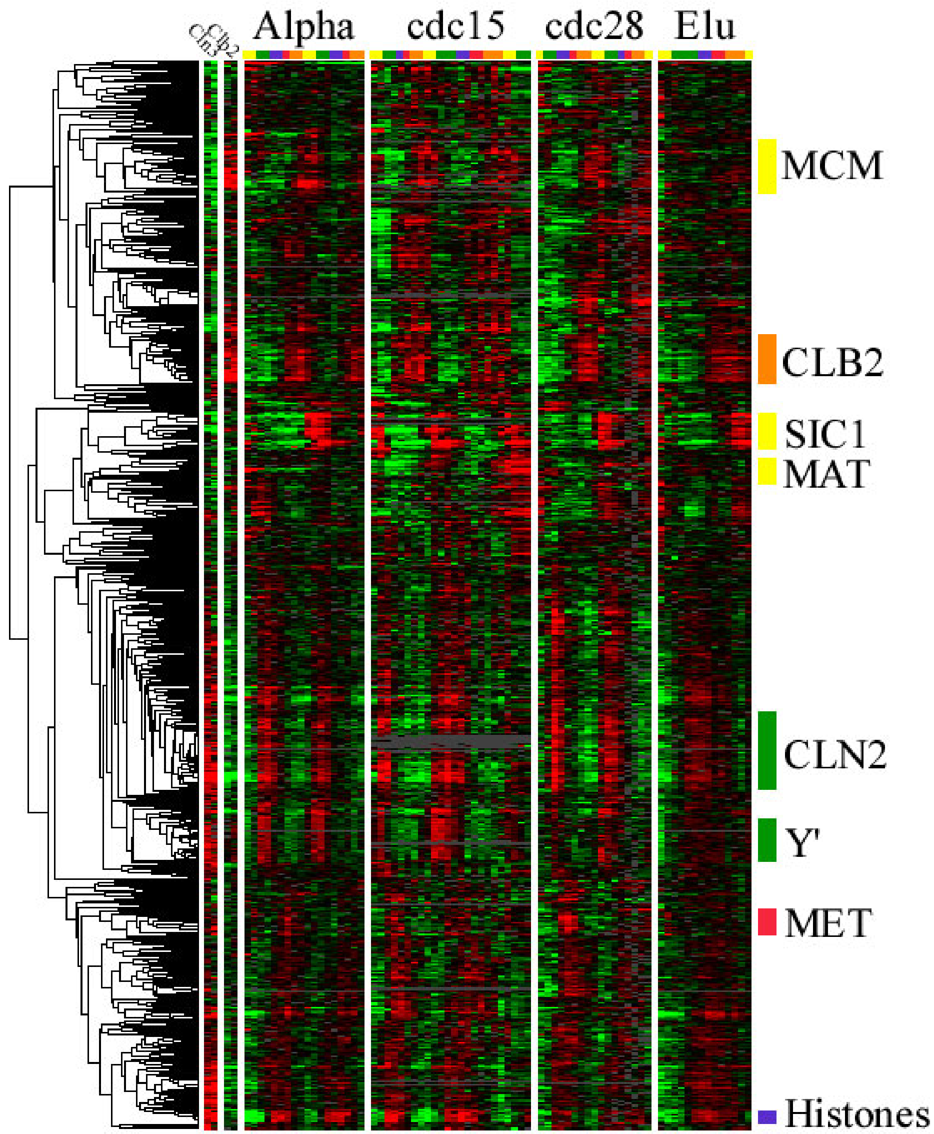

Genes
Experiments


Genes
Experiments


Genes
Experiments



Genes
Experiments

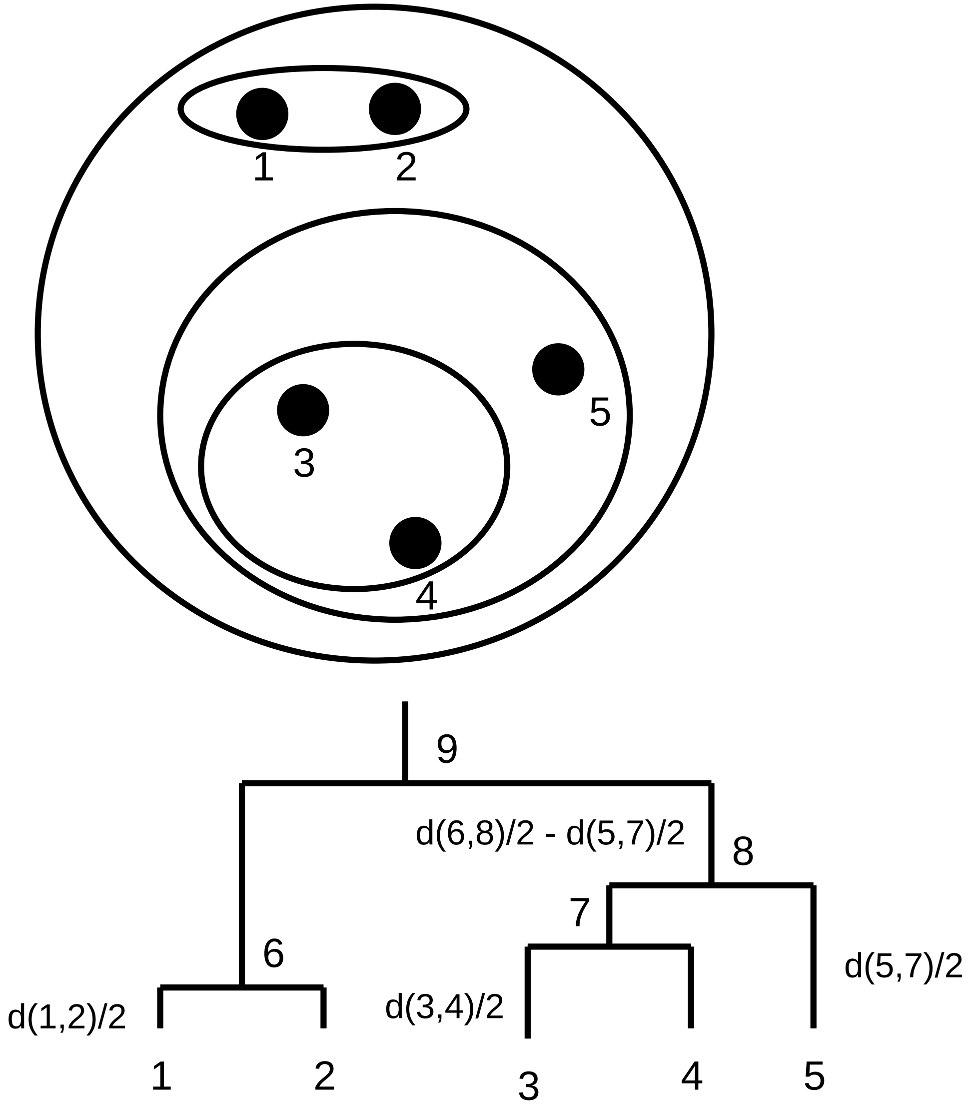


From Spellman et al., http://cellcycle-www.stanford.edu/

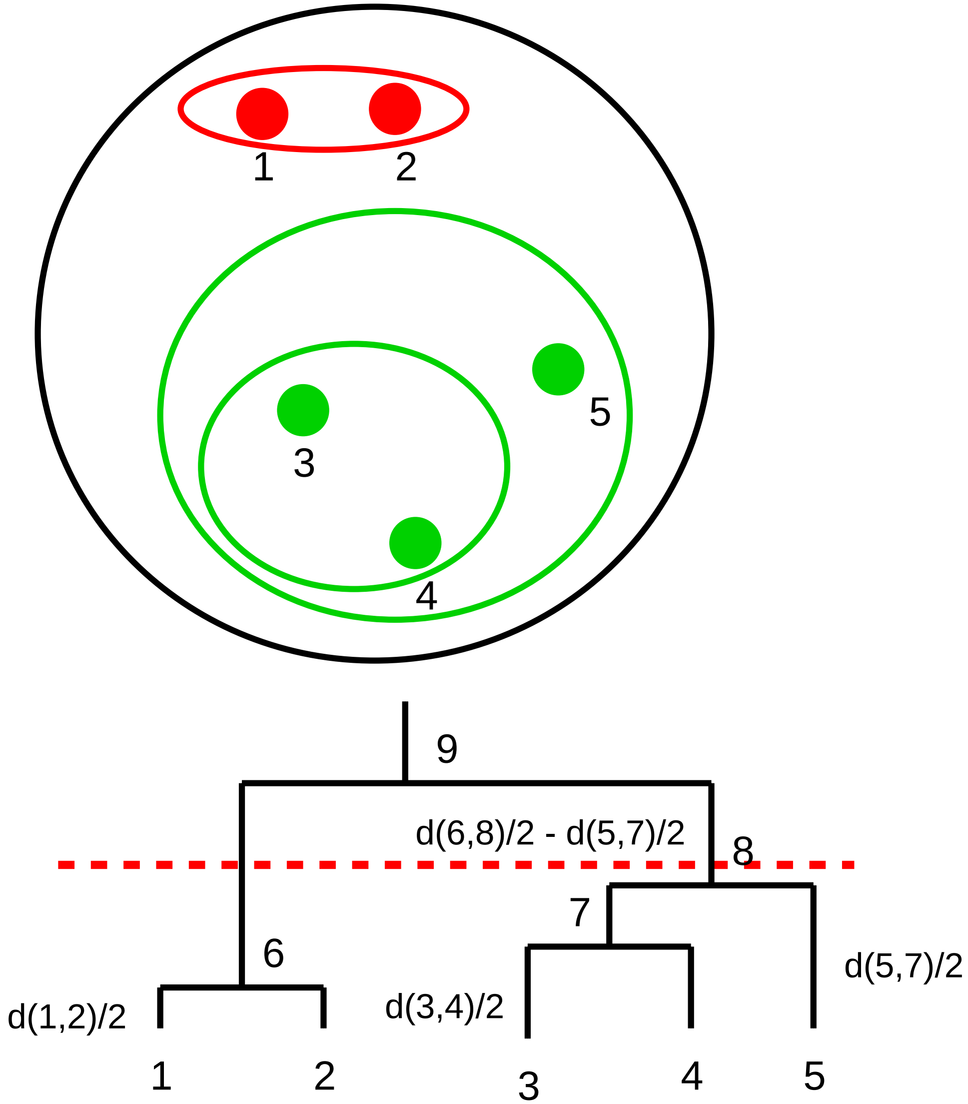
- Hierarchical clustering methods produce a tree or dendrogram $\rightarrow$ Allows the biologist to visualize and interpret the data.

- No need to specify how many clusters are appropriate $\longrightarrow$ partition of the data for each number of clusters K.

- Partitions are obtained from cutting the tree at different levels.



1
2
5
3
4
9
d(6,8)/2 - d(5,7)/2
8
7
6
d(5,7)/2
d(3,4)/2
d(1,2)/2
1
2
3
4
5



1
2
5
3
4
9
d(6,8)/2 - d(5,7)/2
8
7
6
d(5,7)/2
d(1,2)/2
d(3,4)/2
1
2
3
4
5

Principal clustering paradigms

- Non-hierarchical
  Cluster N vectors into K groups in an iterative process.

- Hierarchical
  Hierarchie of nested clusters; each cluster typically consists of the union of two smaller sub-clusters.

Hierarchical methods can be further subdivided

Hierarchical methods can be further subdivided

- Bottom-up or agglomerative clustering:
  Start with a single-object cluster and recursively merge them into larger clusters.

Hierarchical methods can be further subdivided

- Bottom-up or agglomerative clustering:
  Start with a single-object cluster and recursively merge them into larger clusters.

- Top down or divisive clustering:
  Start with a cluster containing all data and recursively divide it into smaller clusters.

Overview of clustering methods

	Hierarchical	Non-hierarchical
Top-down or divisive		K-means Fuzzy/soft K-means
Bottom-up or agglomerative	UPGMA	

Shortcoming of bottom-up agglomerative clustering

- Focus on local structures $\longrightarrow$ loses some of the information present in global patterns.

- Once a data vector has been assigned to a node,
  it cannot be reassigned to another node later
  when global patterns emerge.

Shortcoming of bottom-up agglomerative clustering

- Focus on local structures $\longrightarrow$ loses some of the information present in global patterns.

- Once a data vector has been assigned to a node, it cannot be reassigned to another node later when global patterns emerge.

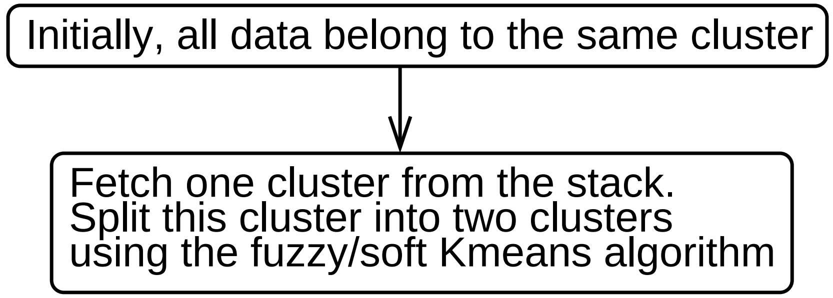
How can we devise a hierarchical top-down approach?

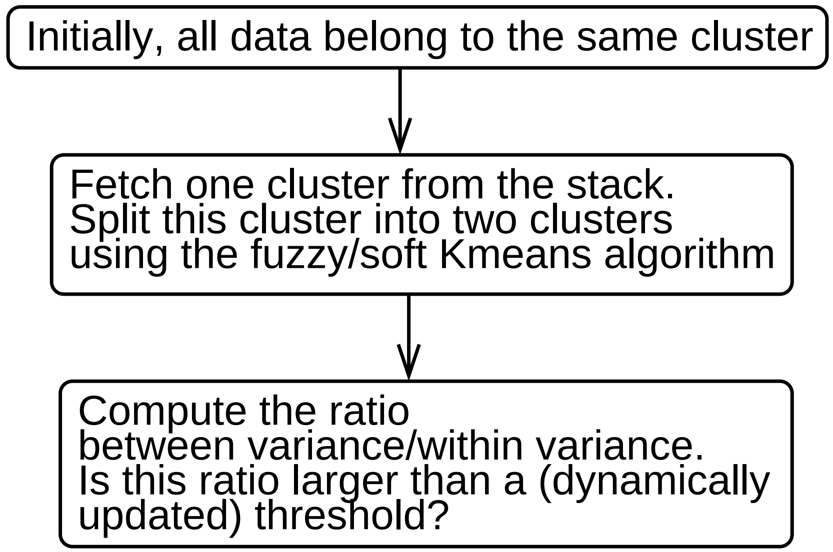
	Hierarchical	Non-hierarchical
Top-down or divisive	?	K-means Fuzzy/soft K-means
Bottom-up or agglomerative	UPGMA	

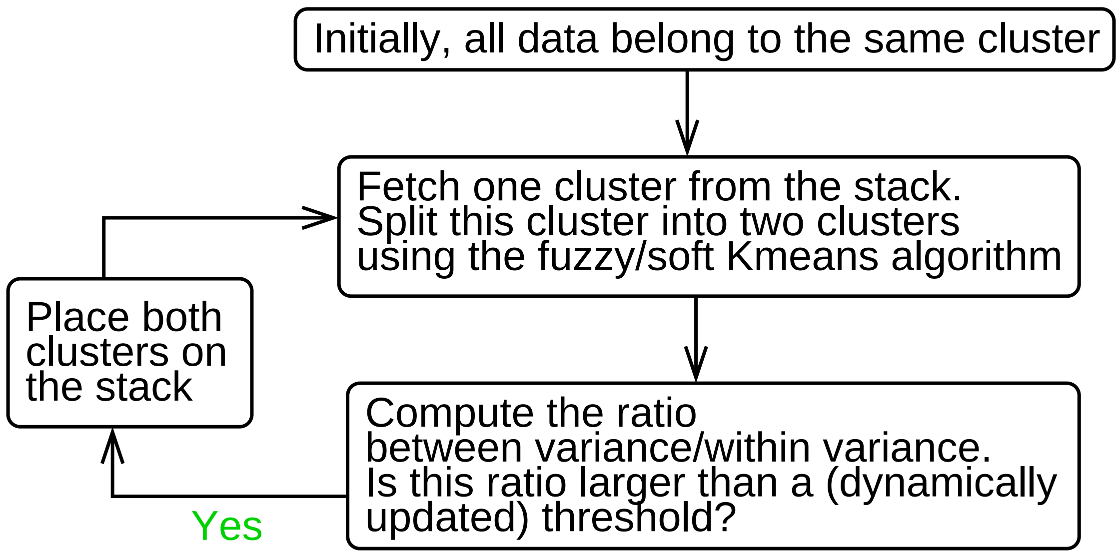
Divisive (top-down) hierarchical clustering:

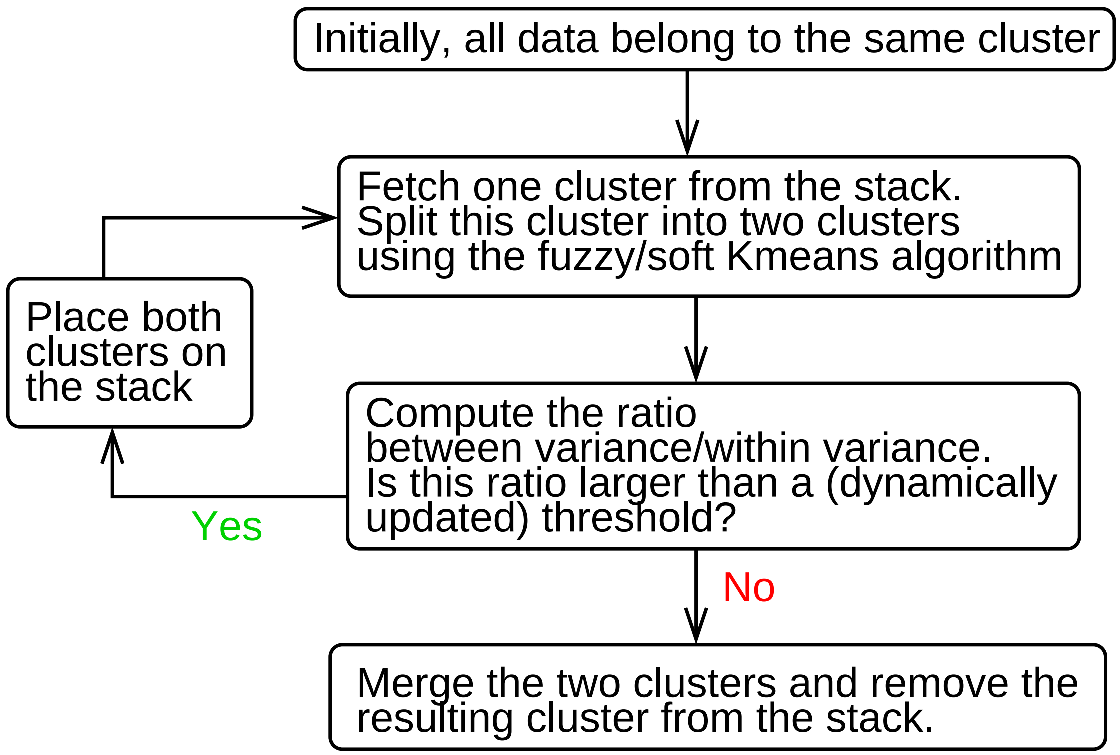
Binary tree-structured vector quantization (BTSVQ)

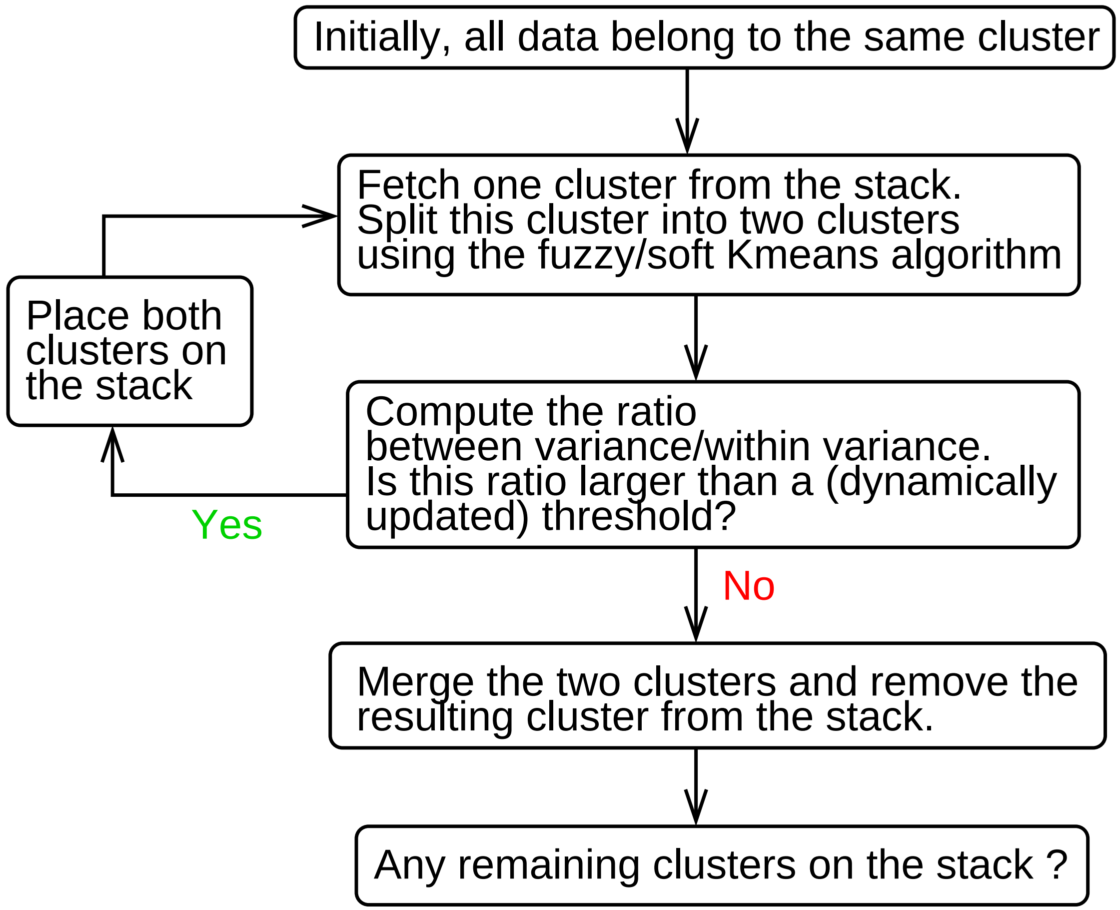
Initially, all data belong to the same cluster

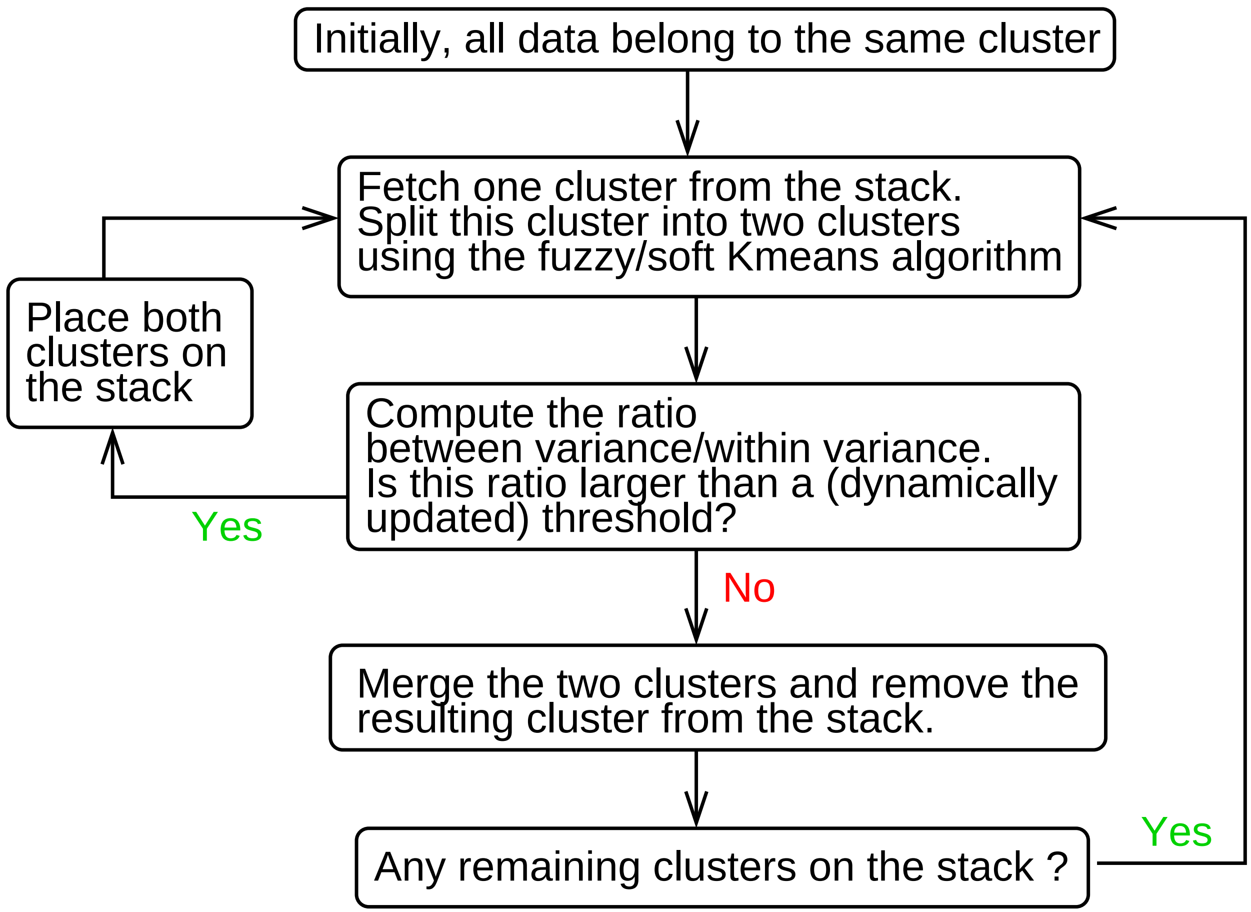

Initially, all data belong to the same cluster
Fetch one cluster from the stack.
Split this cluster into two clusters
using the fuzzy/soft Kmeans algorithm


Initially, all data belong to the same cluster
Fetch one cluster from the stack.
Split this cluster into two clusters
using the fuzzy/soft Kmeans algorithm
Compute the ratio
between variance/within variance.
Is this ratio larger than a (dynamically
updated) threshold?

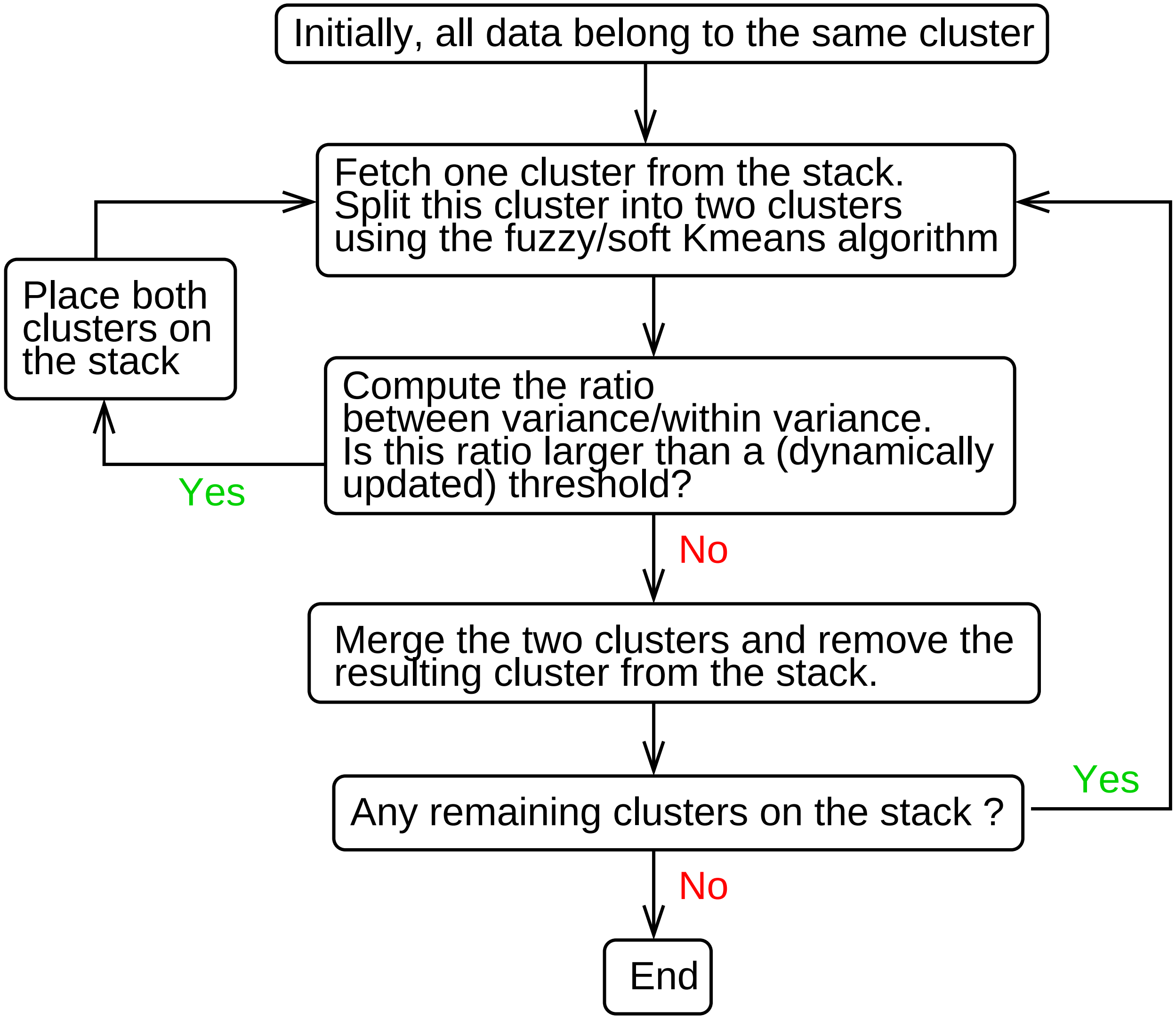

Initially, all data belong to the same cluster
Fetch one cluster from the stack.
Split this cluster into two clusters
using the fuzzy/soft Kmeans algorithm
Place both
clusters on
the stack
Compute the ratio
between variance/within variance.
Is this ratio larger than a (dynamically
updated) threshold?
Yes

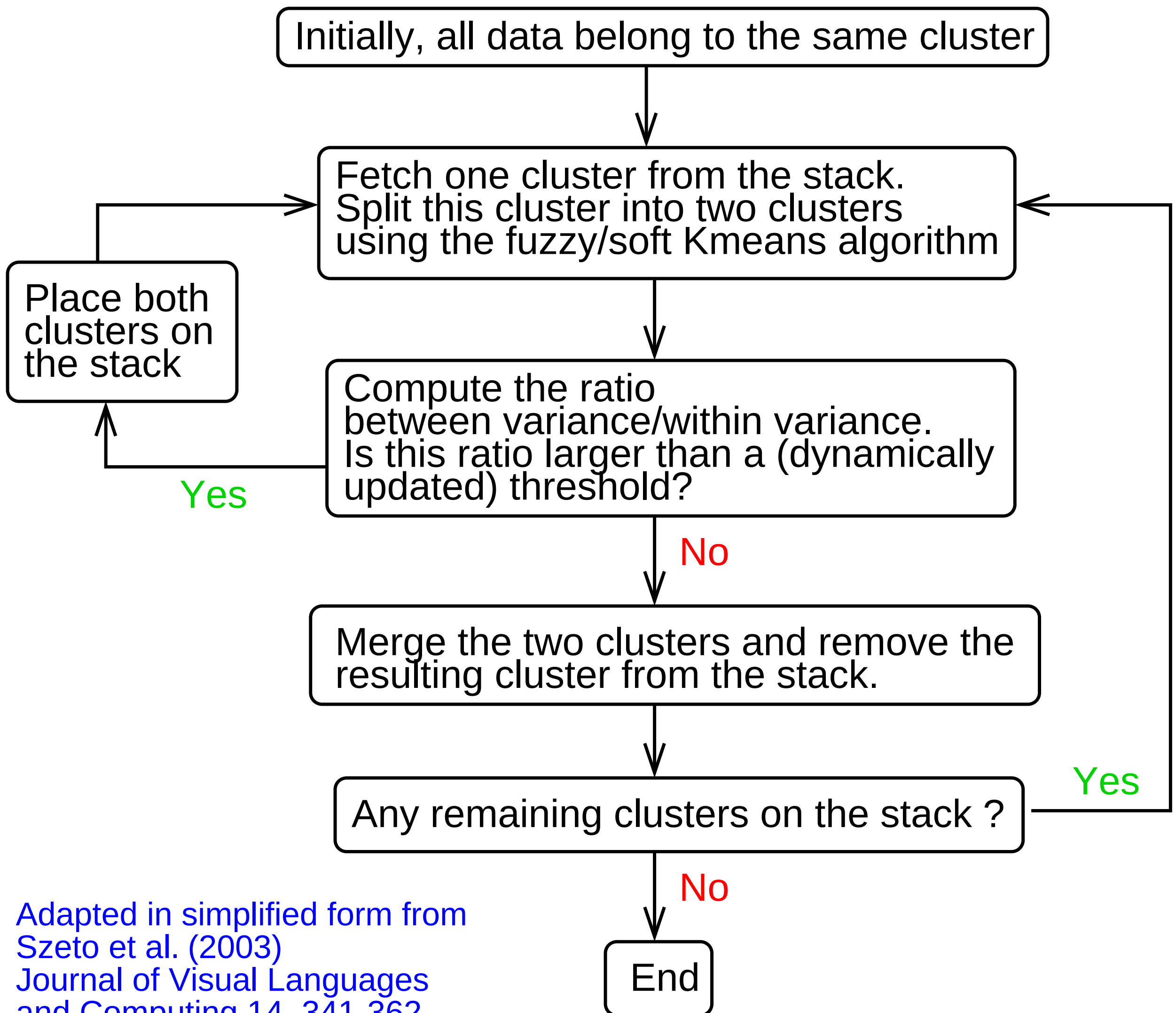

Initially, all data belong to the same cluster
Fetch one cluster from the stack.
Split this cluster into two clusters
using the fuzzy/soft Kmeans algorithm
Place both
clusters on
the stack
Compute the ratio
between variance/within variance.
Is this ratio larger than a (dynamically
updated) threshold?
Yes
No
Merge the two clusters and remove the
resulting cluster from the stack.


Initially, all data belong to the same cluster
Fetch one cluster from the stack. Split this cluster into two clusters using the fuzzy/soft Kmeans algorithm
Place both clusters on the stack
Compute the ratio between variance/within variance. Is this ratio larger than a (dynamically updated) threshold?
Yes
No
Merge the two clusters and remove the resulting cluster from the stack.
Any remaining clusters on the stack ?



Initially, all data belong to the same cluster
Fetch one cluster from the stack.
Split this cluster into two clusters
using the fuzzy/soft Kmeans algorithm
Place both
clusters on
the stack
Compute the ratio
between variance/within variance.
Is this ratio larger than a (dynamically
updated) threshold?
Yes
No
Merge the two clusters and remove the
resulting cluster from the stack.
Any remaining clusters on the stack ?
Yes


Initially, all data belong to the same cluster
Fetch one cluster from the stack.
Split this cluster into two clusters
using the fuzzy/soft Kmeans algorithm
Place both
clusters on
the stack
Compute the ratio
between variance/within variance.
Is this ratio larger than a (dynamically
updated) threshold?
Yes
No
Merge the two clusters and remove the
resulting cluster from the stack.
Any remaining clusters on the stack ?
Yes
No
End



Adapted in simplified form from
Szeto et al. (2003)
Journal of Visual Languages
and Computing 14, 341-362

Overview of clustering methods

	Hierarchical	Non-hierarchical
Top-down or divisive	BTSVQ	K-means Fuzzy/soft K-means
Bottom-up or agglomerative	UPGMA	

Pitfalls of clustering

Pitfalls of clustering

- Clustering algorithms always produce clusters even for uniformaly distrubuted data.

Pitfalls of clustering

- Clustering algorithms always produce clusters even for uniformaly distrubuted data.
- Difficult to test the null hypothesis of no clusters (current research).

Pitfalls of clustering

- Clustering algorithms always produce clusters even for uniformaly distrubuted data.
- Difficult to test the null hypothesis of no clusters (current research).
- Difficult to estimate the true number of clusters (current research).

Pitfalls of clustering

- Clustering algorithms always produce clusters even for uniformaly distrubuted data.
- Difficult to test the null hypothesis of no clusters (current research).
- Difficult to estimate the true number of clusters (current research).
- Risk of artifacts.
- Use clustering only for hypothesis generation.

Pitfalls of clustering

- Clustering algorithms always produce clusters even for uniformaly distrubuted data.
- Difficult to test the null hypothesis of no clusters (current research).
- Difficult to estimate the true number of clusters (current research).
- Risk of artifacts.
- Use clustering only for hypothesis generation.
- Independent experimental verification required.

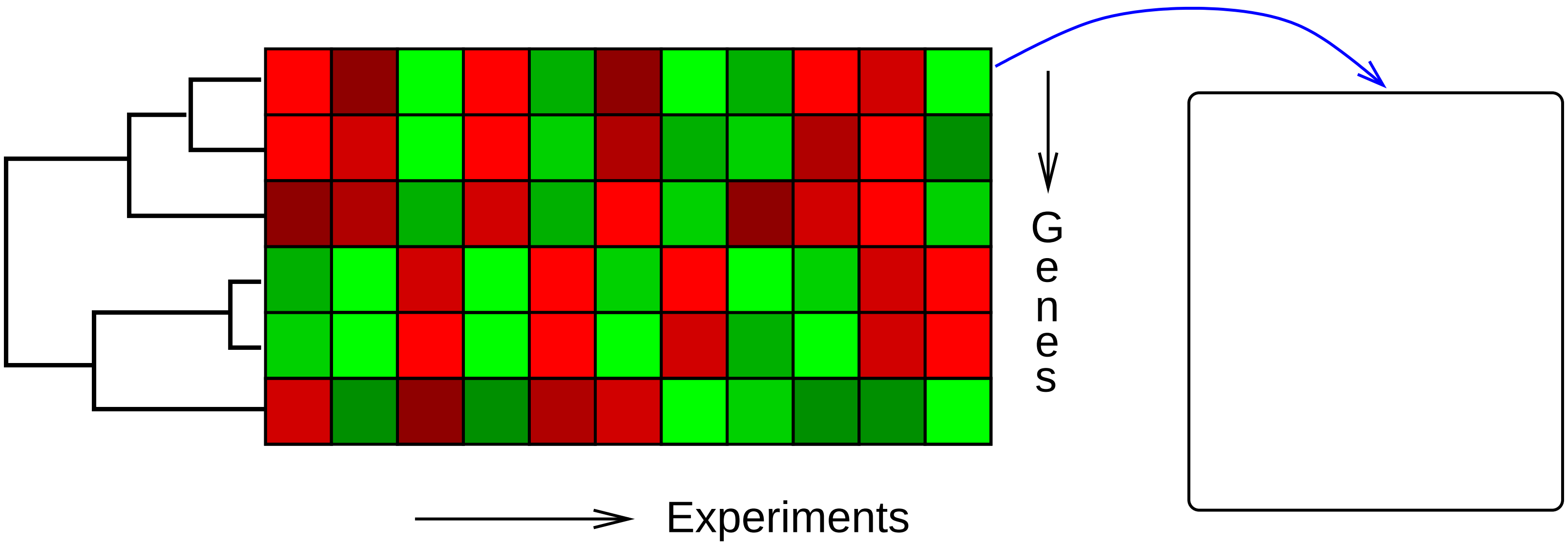
Deciding on the number of clusters: Gap statistic

Tibshirani, Walther, Hastie (2001), J. Royal Statistical Society B

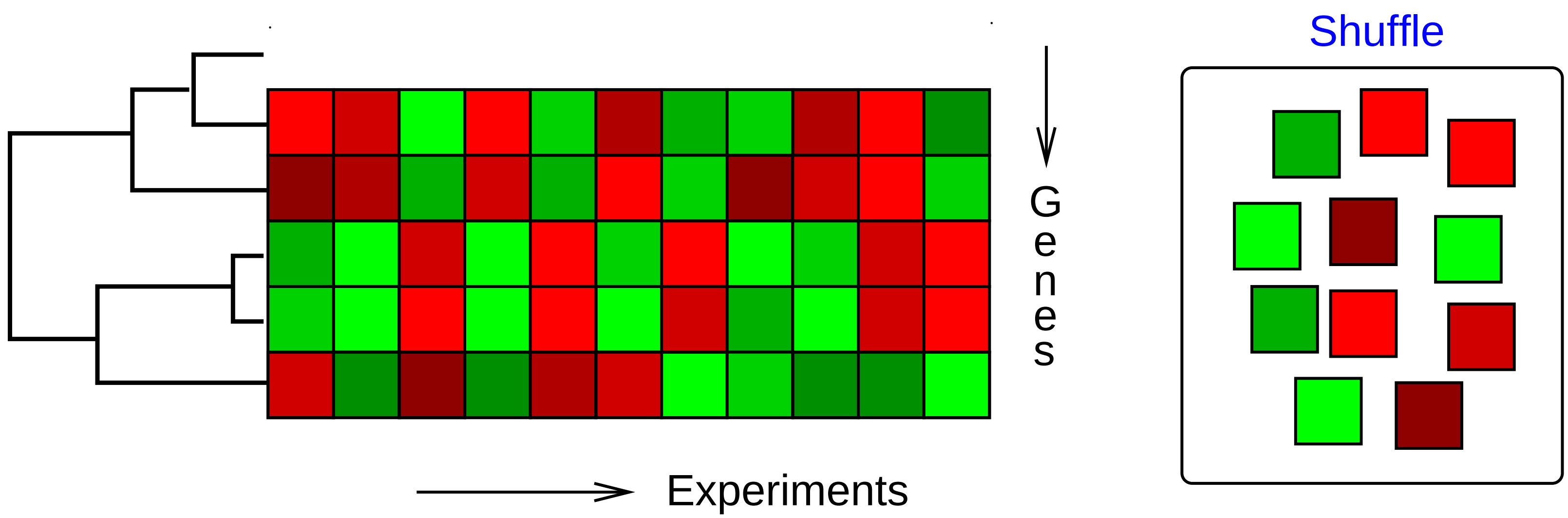
Idea:

- Compute E_K for randomized data.
- Compare this with E_K from real data.

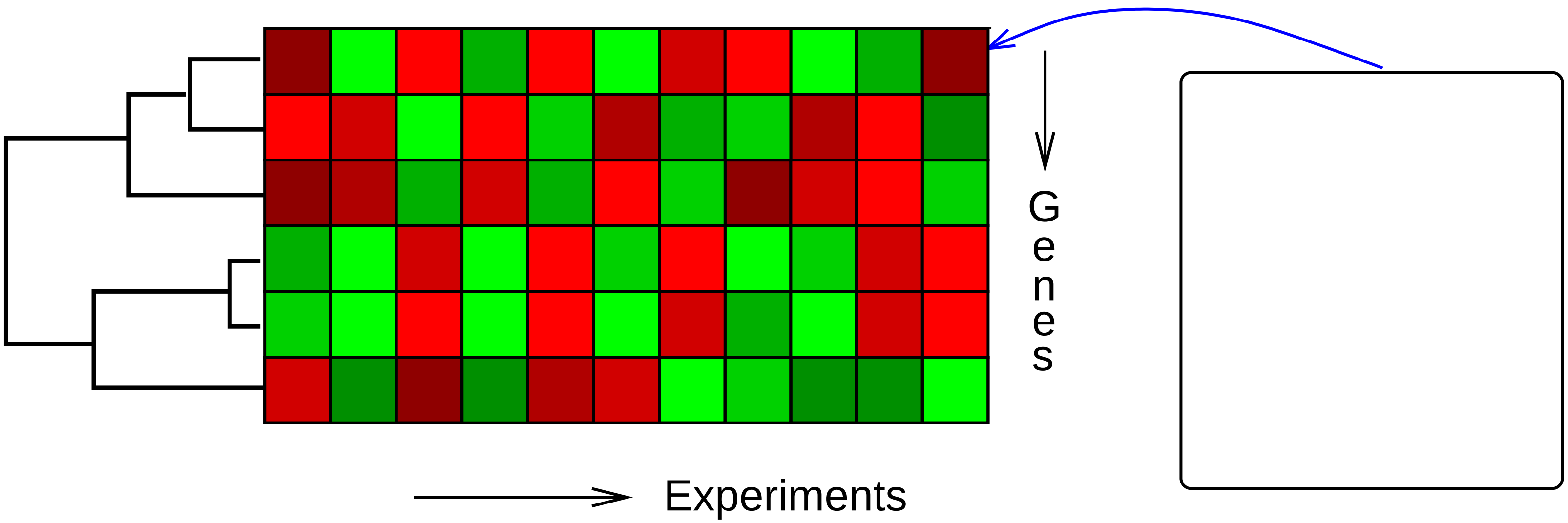
Randomize data



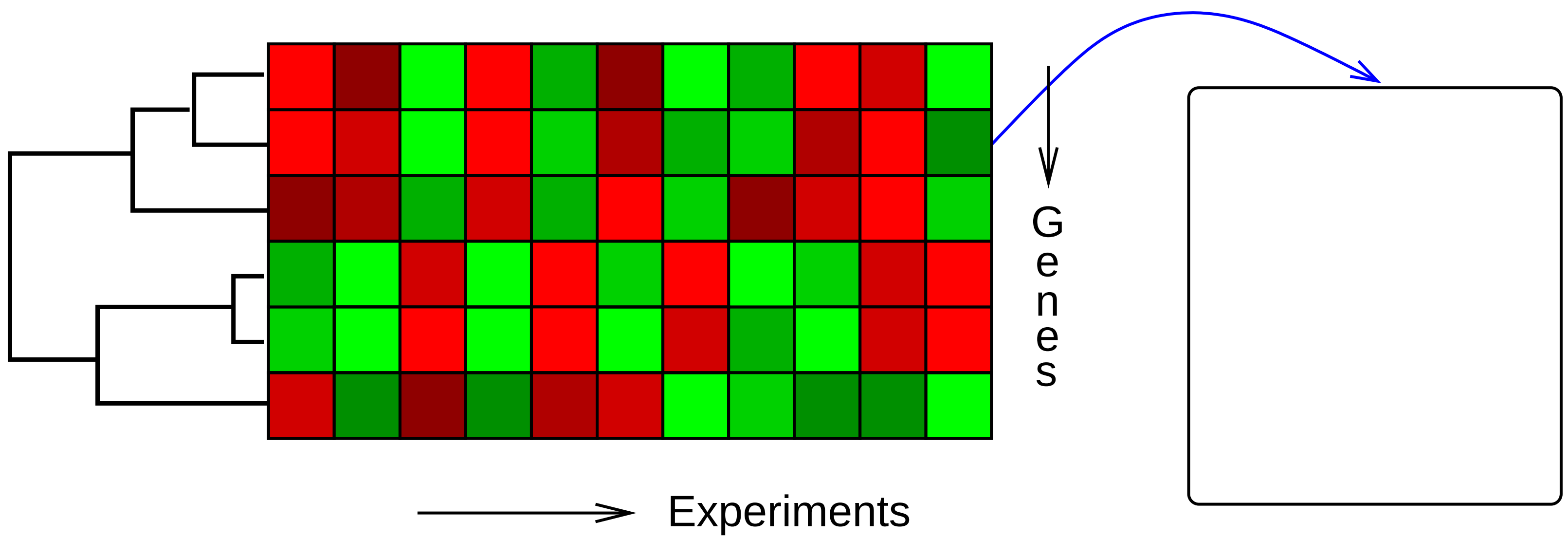
Randomize data



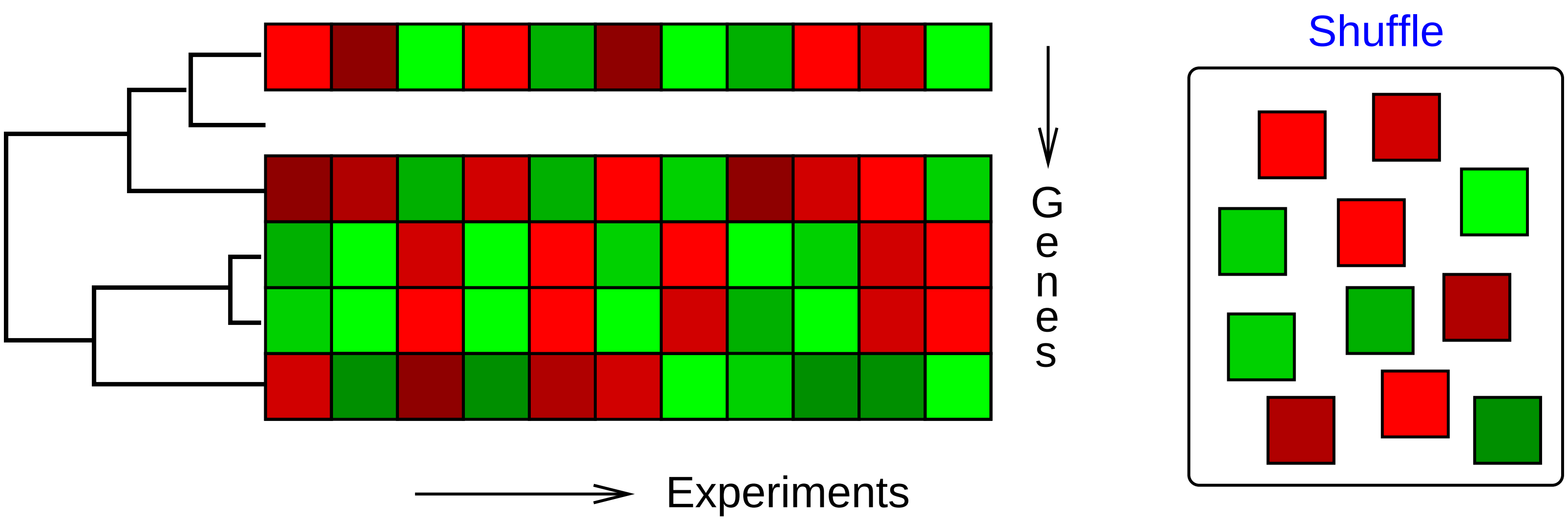
Randomize data



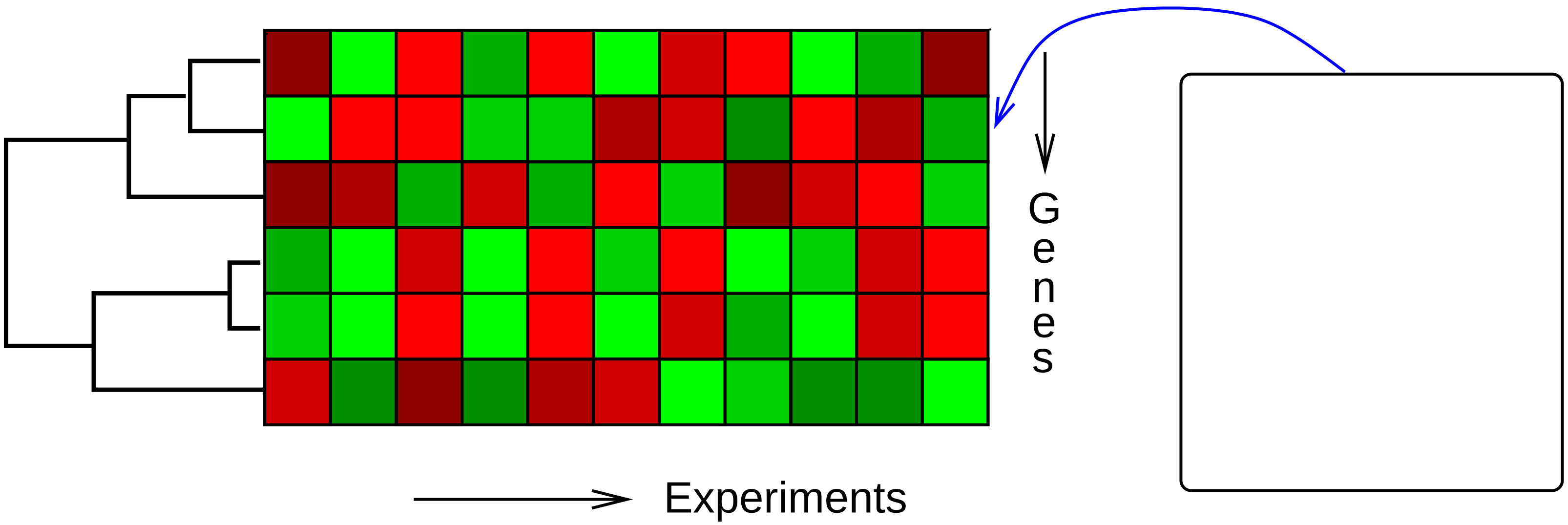
Randomize data



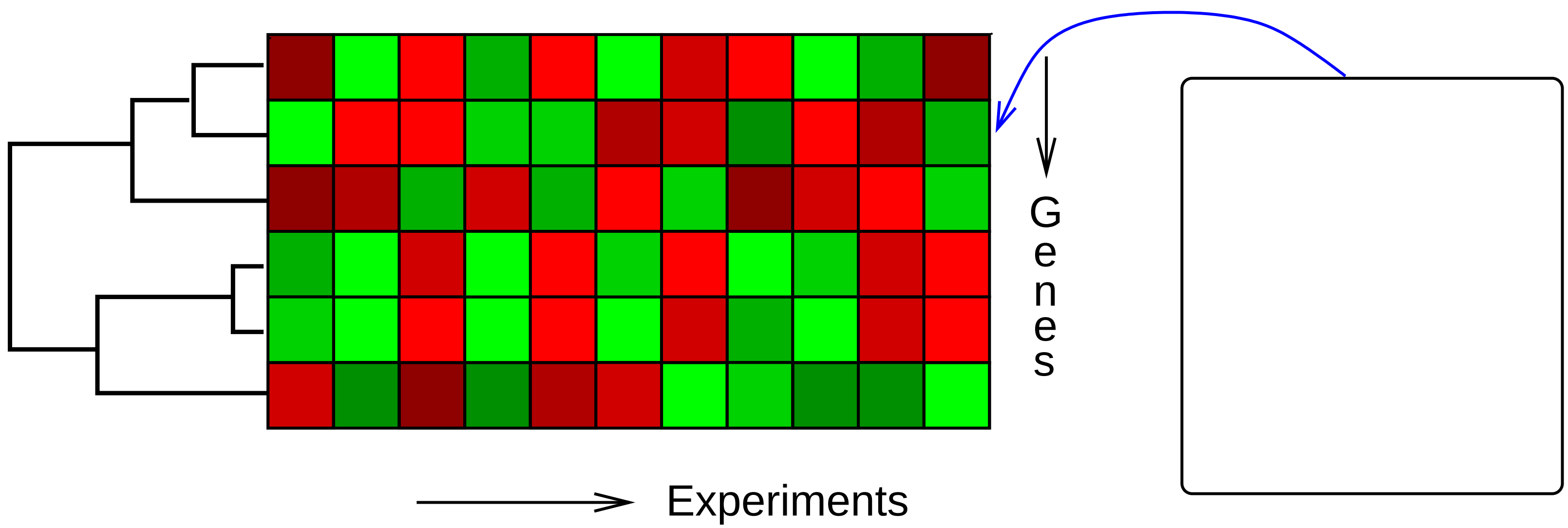
Randomize data



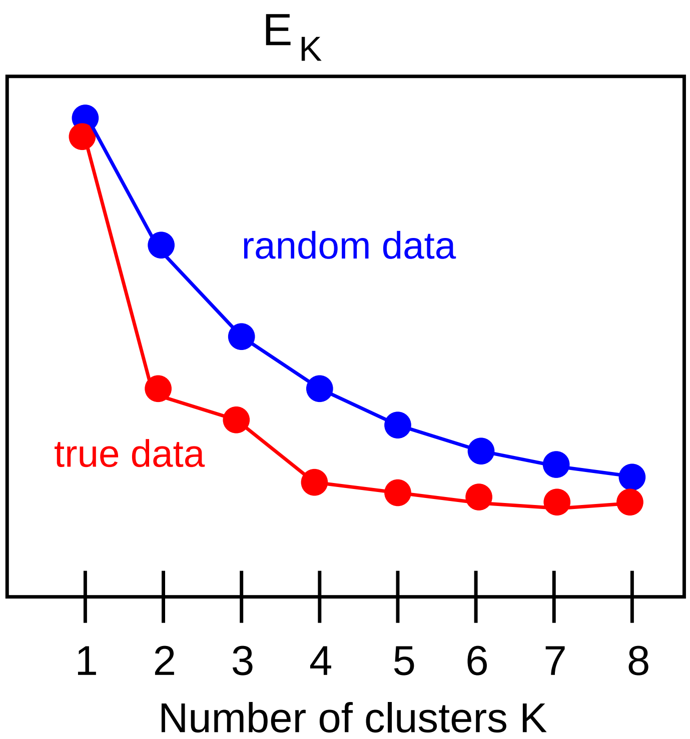
Randomize data

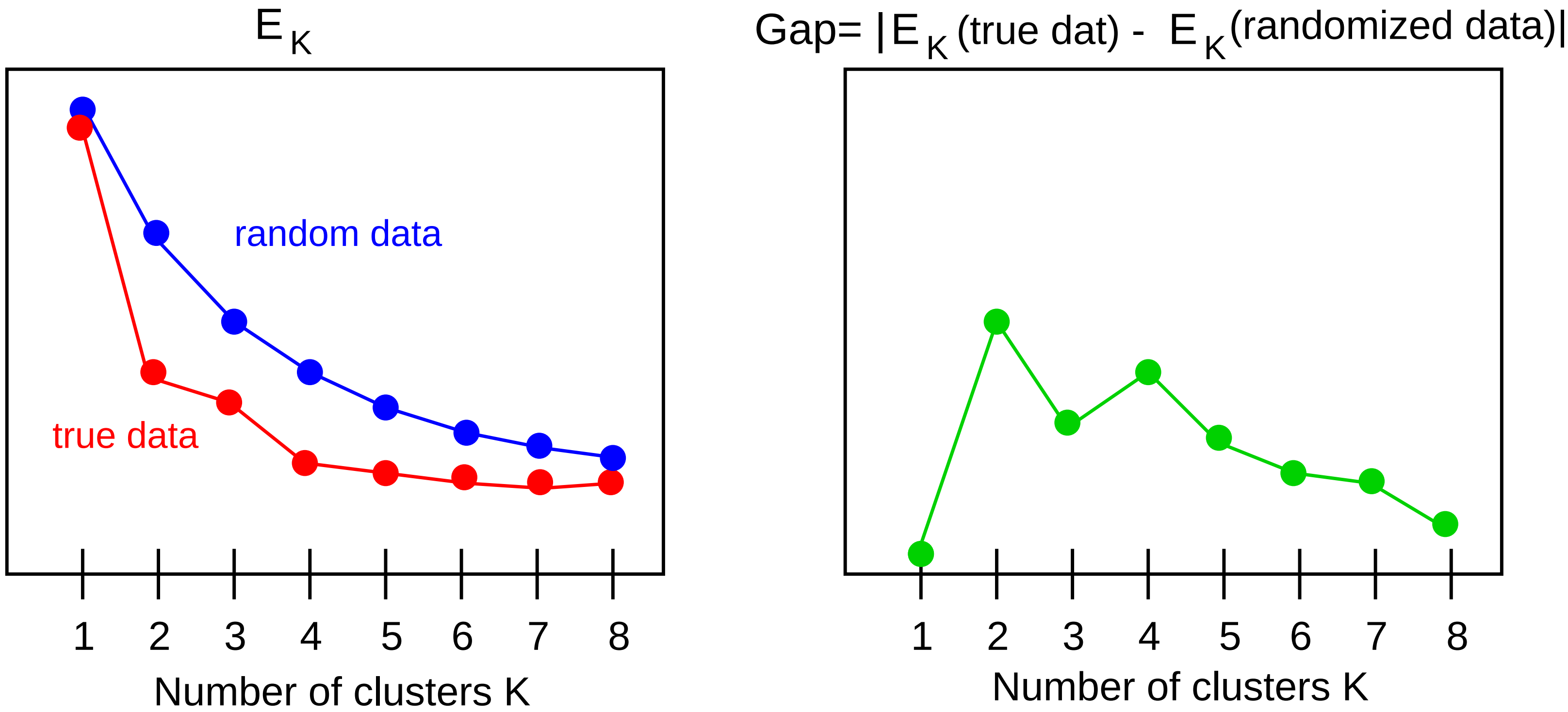


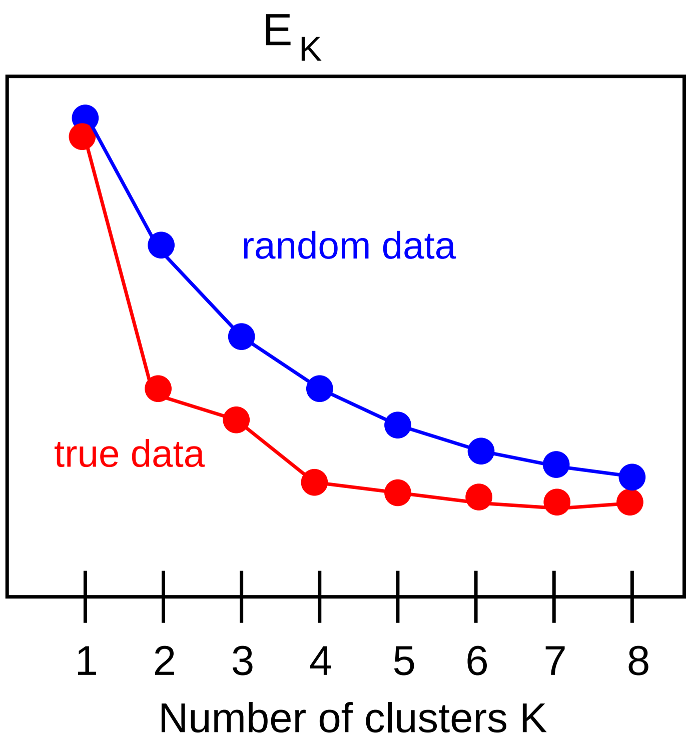
Randomize data

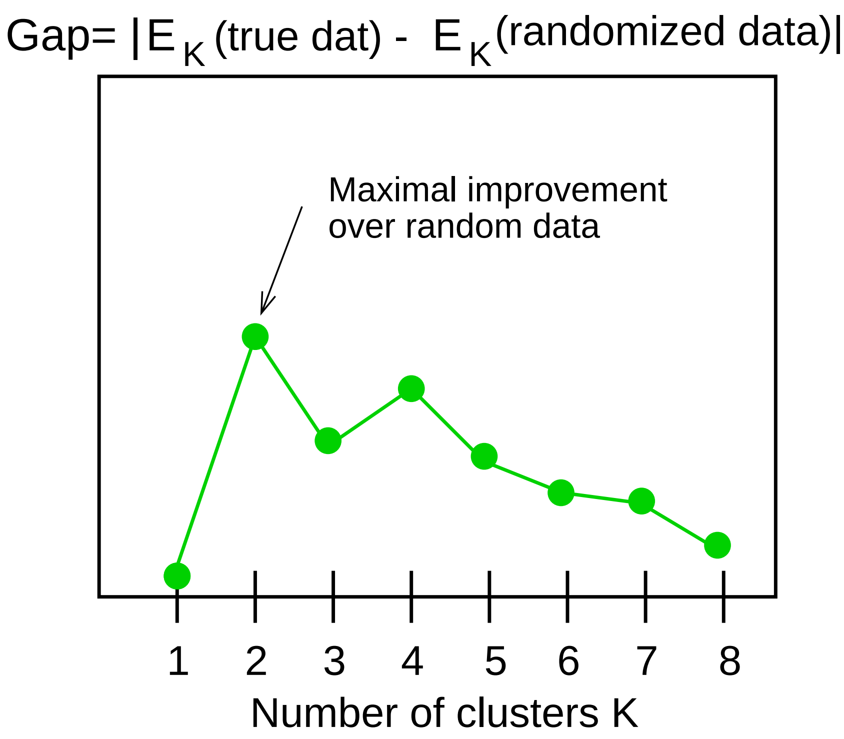


And so on . . .


E_K
random data
true data
1 2 3 4 5 6 7 8
Number of clusters K


E K
random data
true data
1 2 3 4 5 6 7 8
Number of clusters K
Gap= |E K (true dat) - E K (randomized data)|
1 2 3 4 5 6 7 8
Number of clusters K





Adapted from Hastie, Tibshiranie, Friedman: The Elements of Statistical Learning, Springer 2001