


# Hierarchical Clustering

Mahdi Roozbahani  
Georgia Tech

# Outline

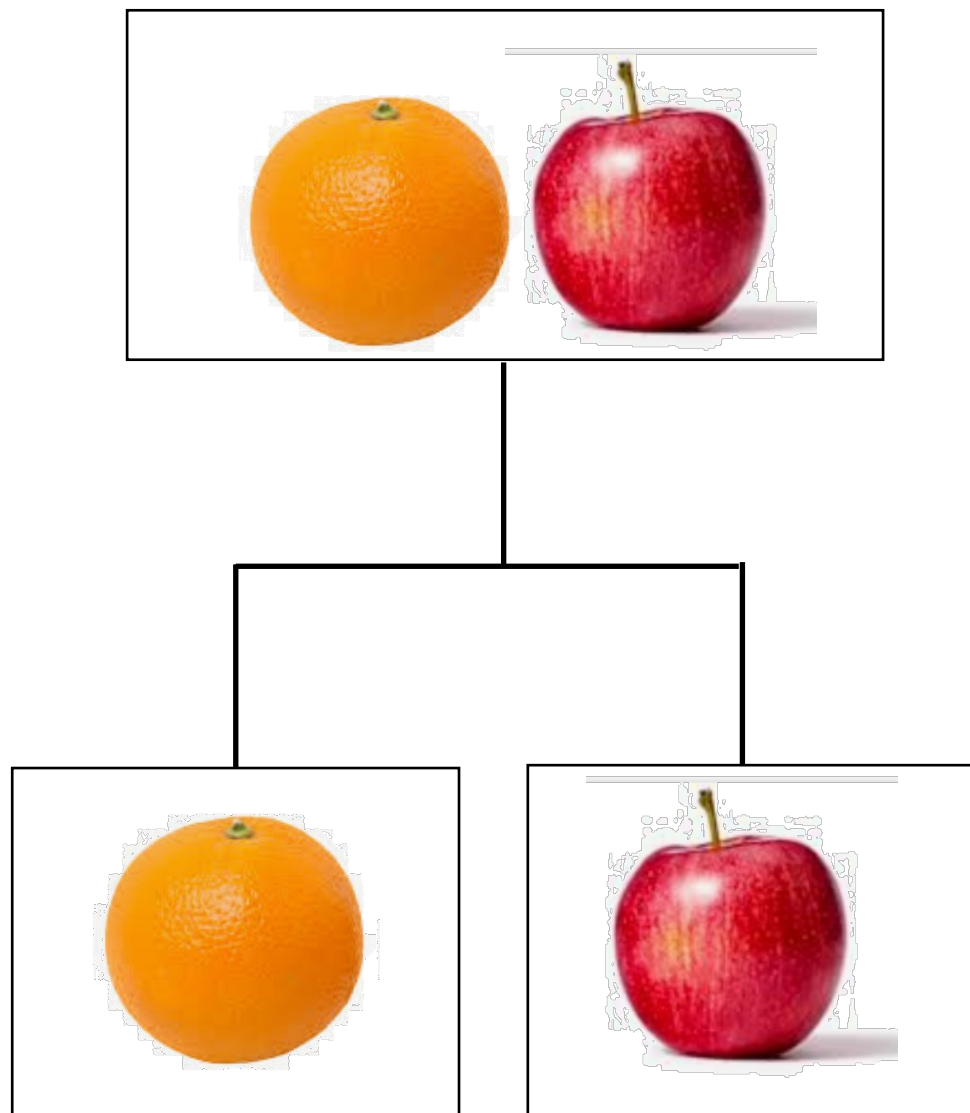
- Overview 
- Bottom-Up vs Top-Down Clustering
- Measuring Distance between Clusters

# Hierarchical Clustering vs Partitional Clustering

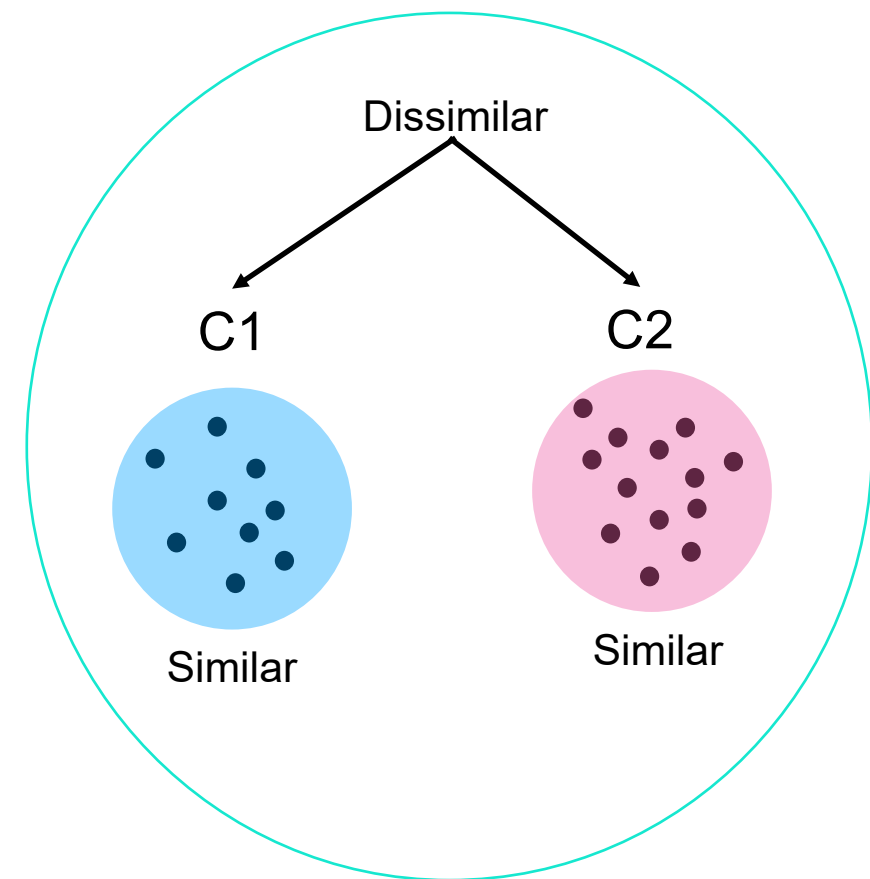
Agglomerative

Divisive

K-Means

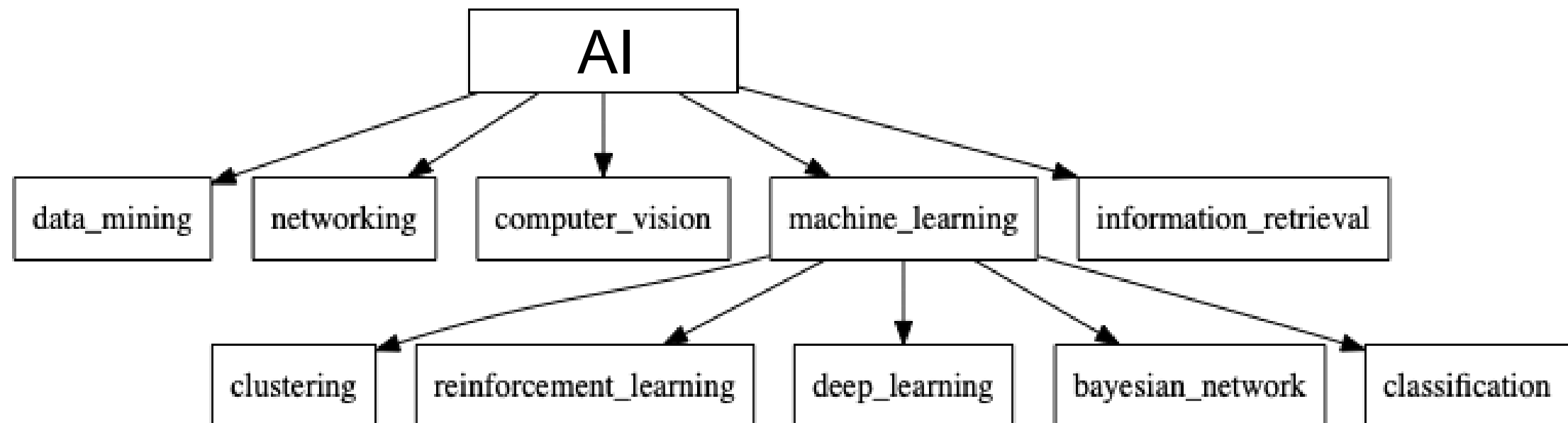


Tree structure (parent-child relationship)



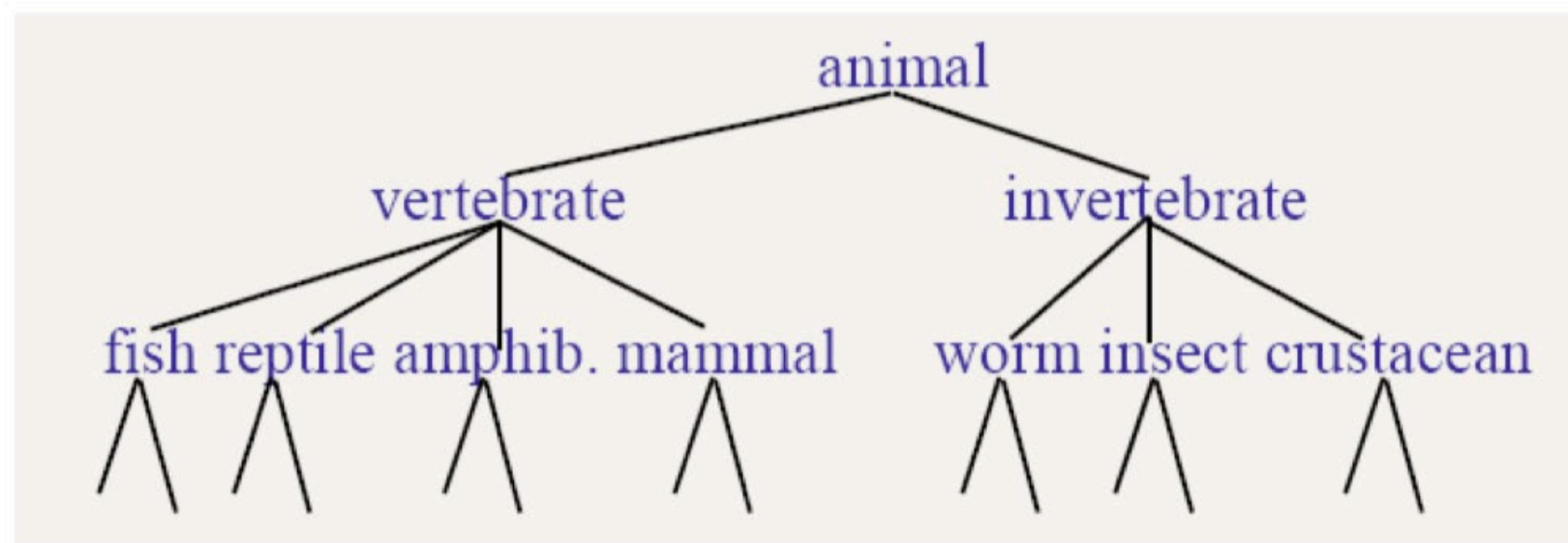
# Hierarchical Clustering

- How to organize a set of CS papers into a hierarchy?



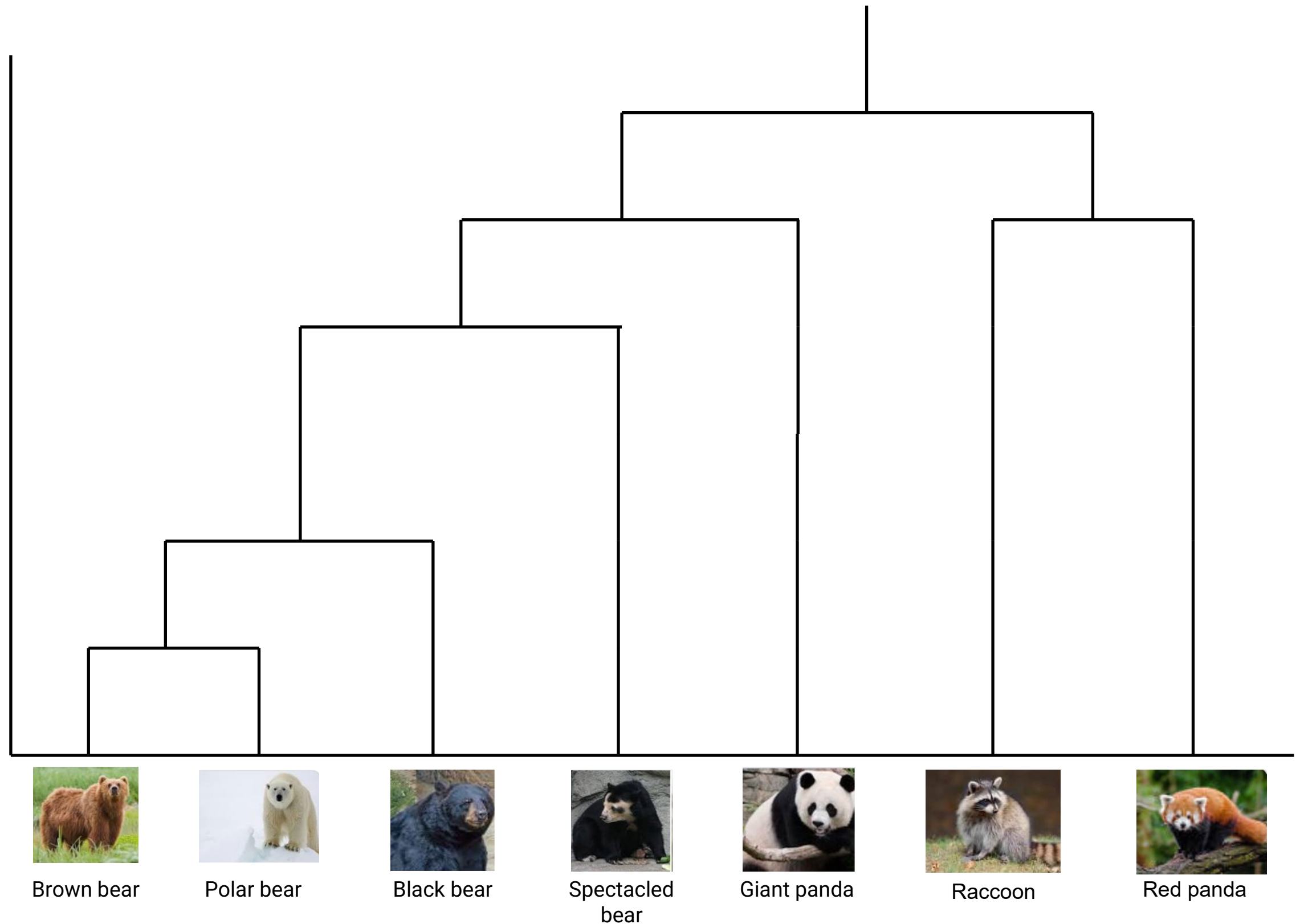
# Hierarchical Clustering

- Organize objects into a tree-based hierarchical taxonomy (dendrogram)



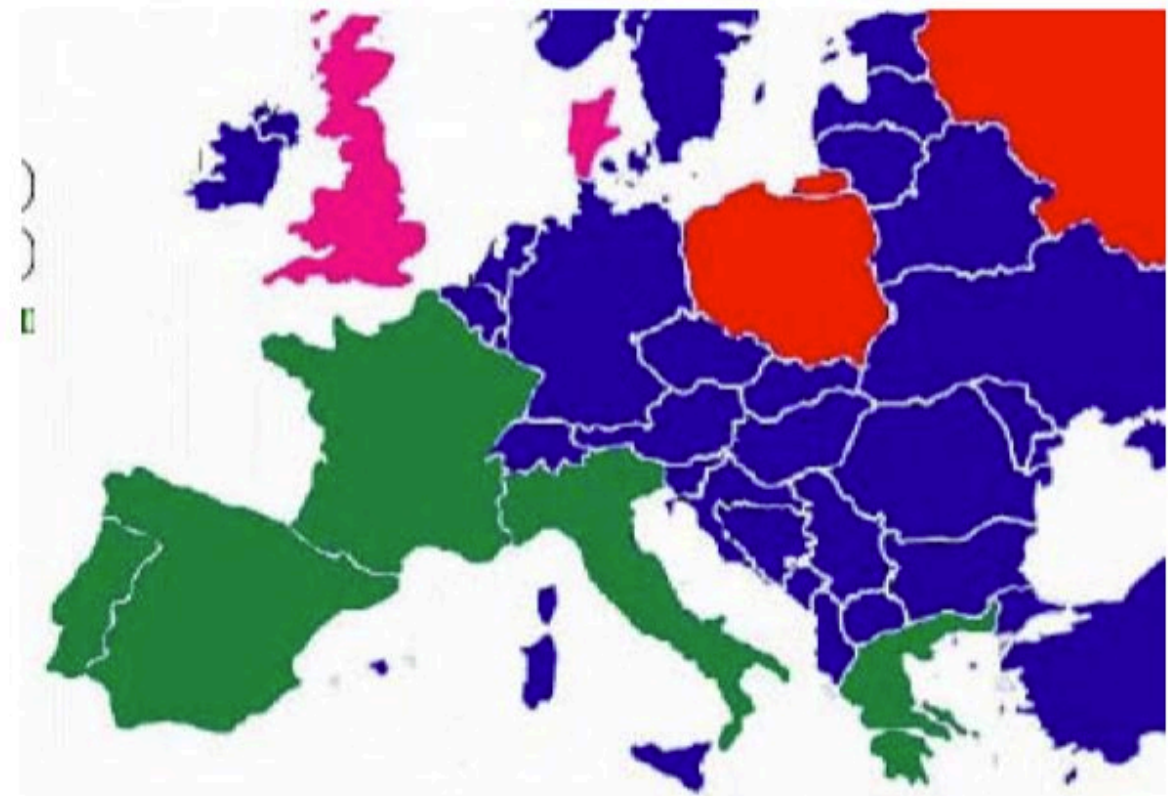
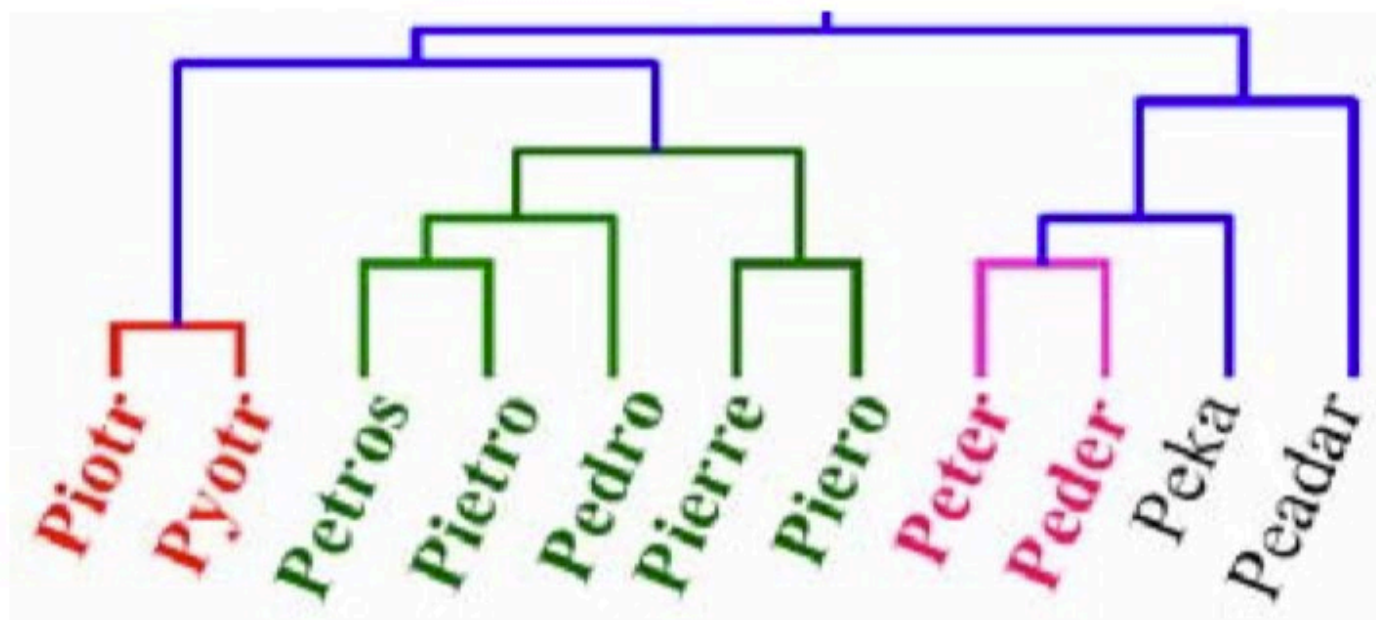
- Many applications in the real world
  - Web pages
  - News articles
  - Scientific papers

## DNA sequencing and hierarchical clustering to find the phylogenetic tree of animal evolution




# Hierarchical Clustering

- Organizing data at multiple granularities
- Cutting the dendrogram at a desired level leads to a sub-cluster: each connected component forms a cluster



# Outline

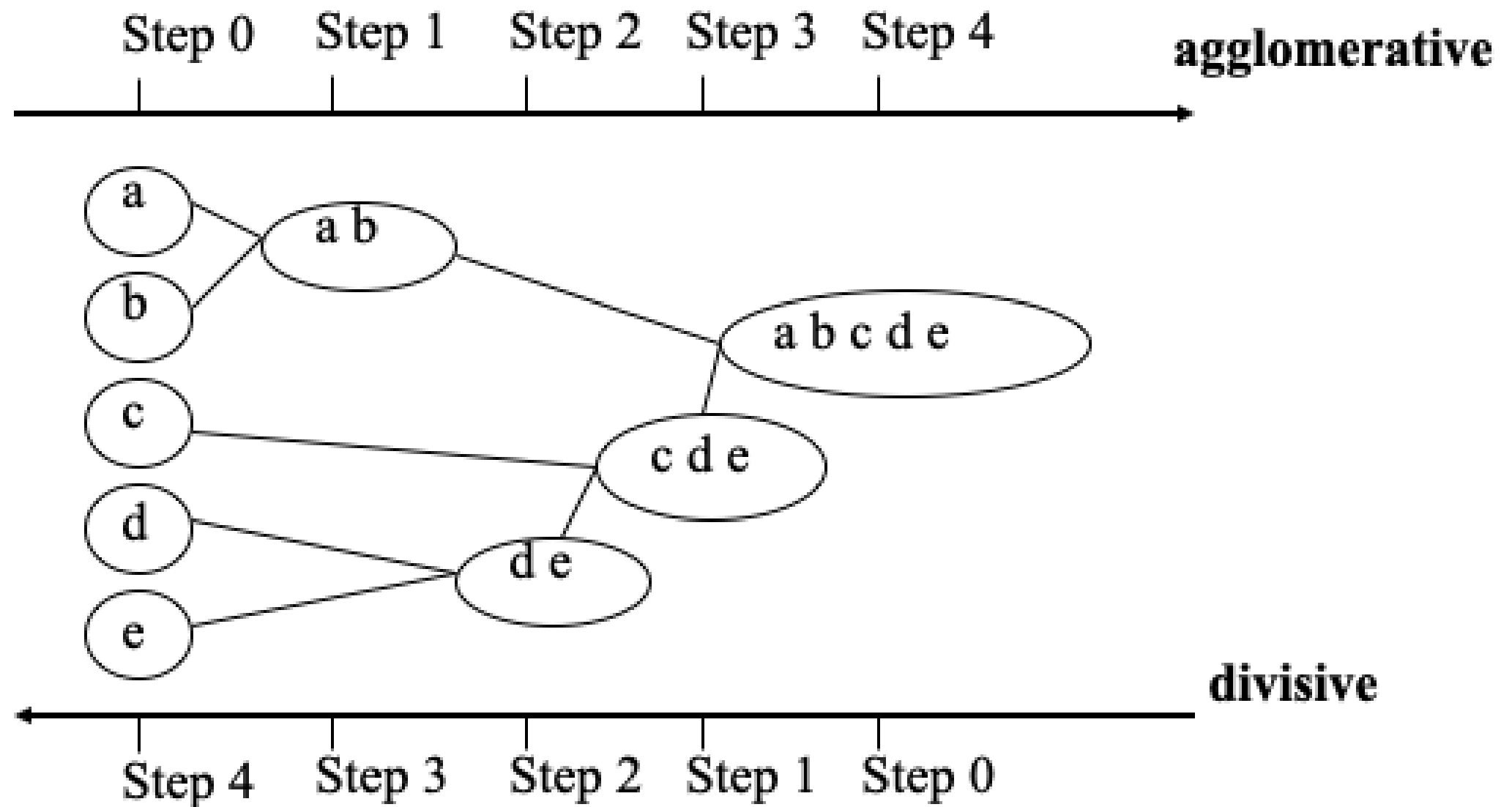
- Overview
- Bottom-Up vs Top-Down Clustering 
- Measuring Distance between Clusters



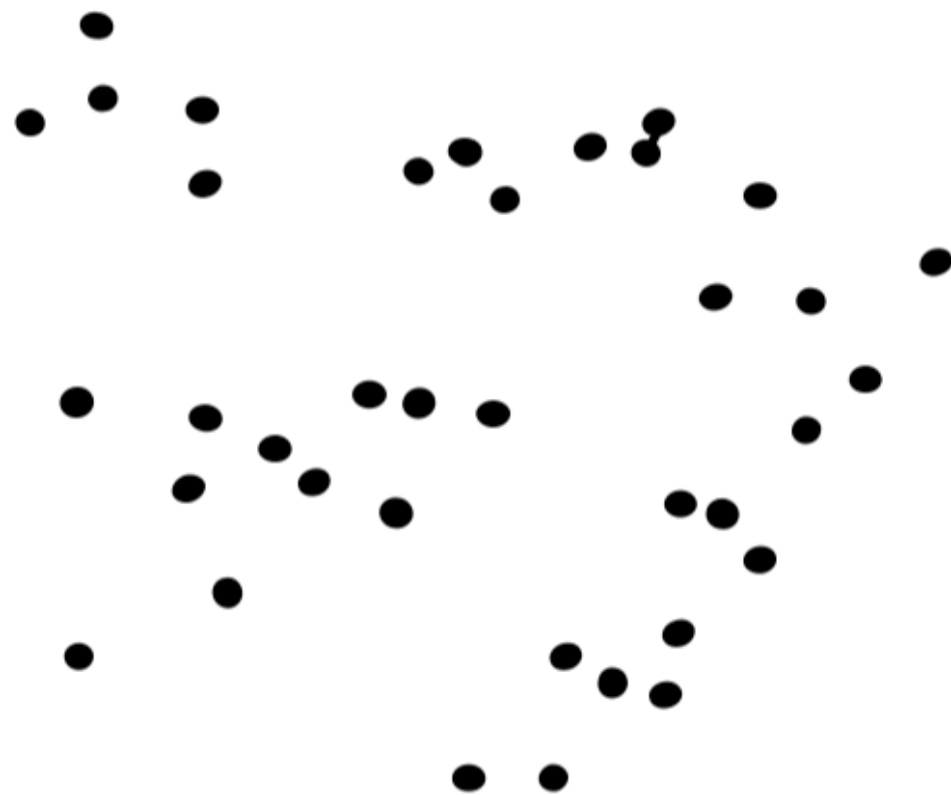
# Two Paradigms for Hierarchical Clustering

- Bottom-up Agglomerative Clustering
  - Start by considering each object as a separate cluster
  - Repeatedly join the closest pair of clusters
  - Stop when there is only one cluster left
- Top-Down Divisive Clustering
  - Start by considering all objects as one large cluster
  - Recursively divide each cluster into two sub-clusters
  - Stop when each cluster contains only one object

# Bottom-Up v.s. Top-Down

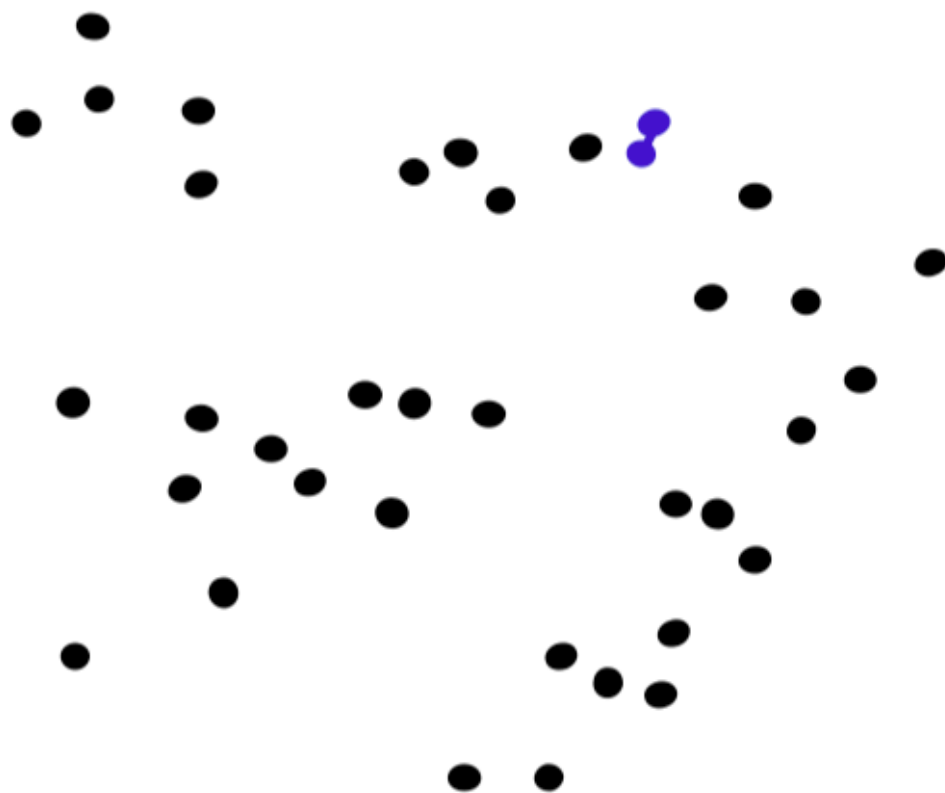


# Bottom-Up Agglomerative Clustering



1. Say "Every point is it's own cluster"

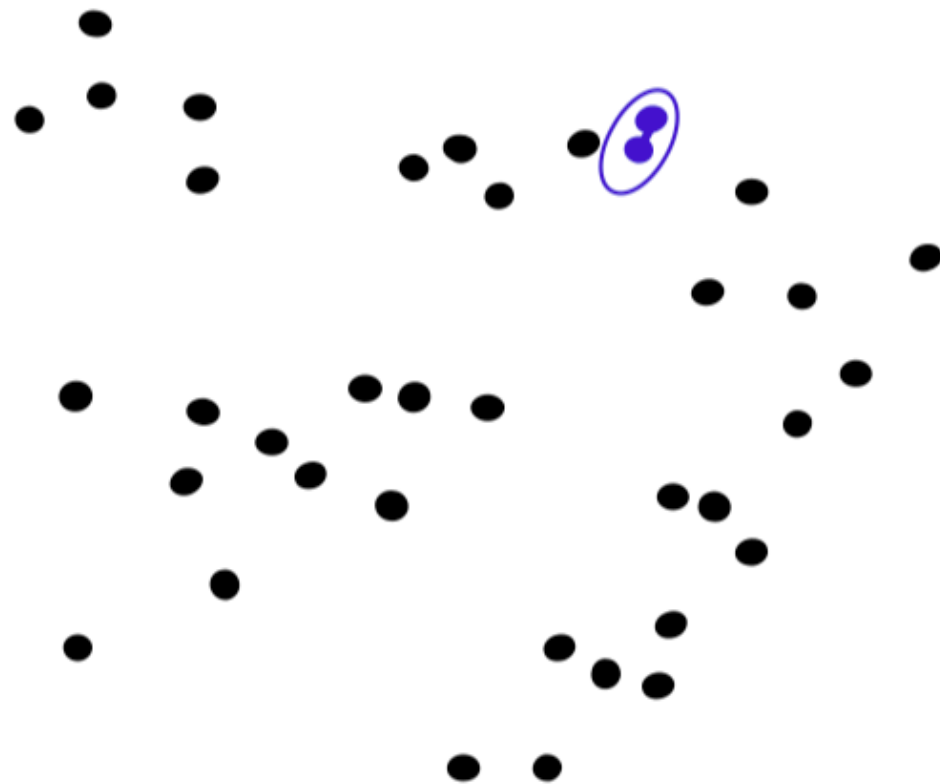
# Bottom-Up Agglomerative Clustering



1. Say "Every point is it's own cluster"
2. Find "most similar" pair of clusters



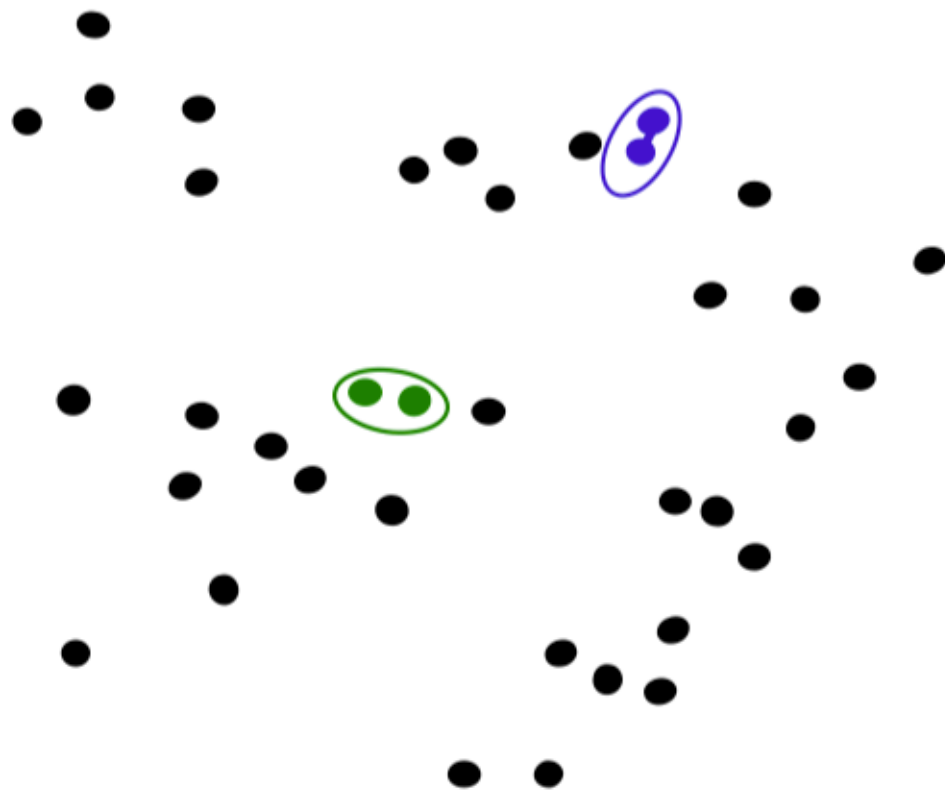
# Bottom-Up Agglomerative Clustering



1. Say "Every point is it's own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster



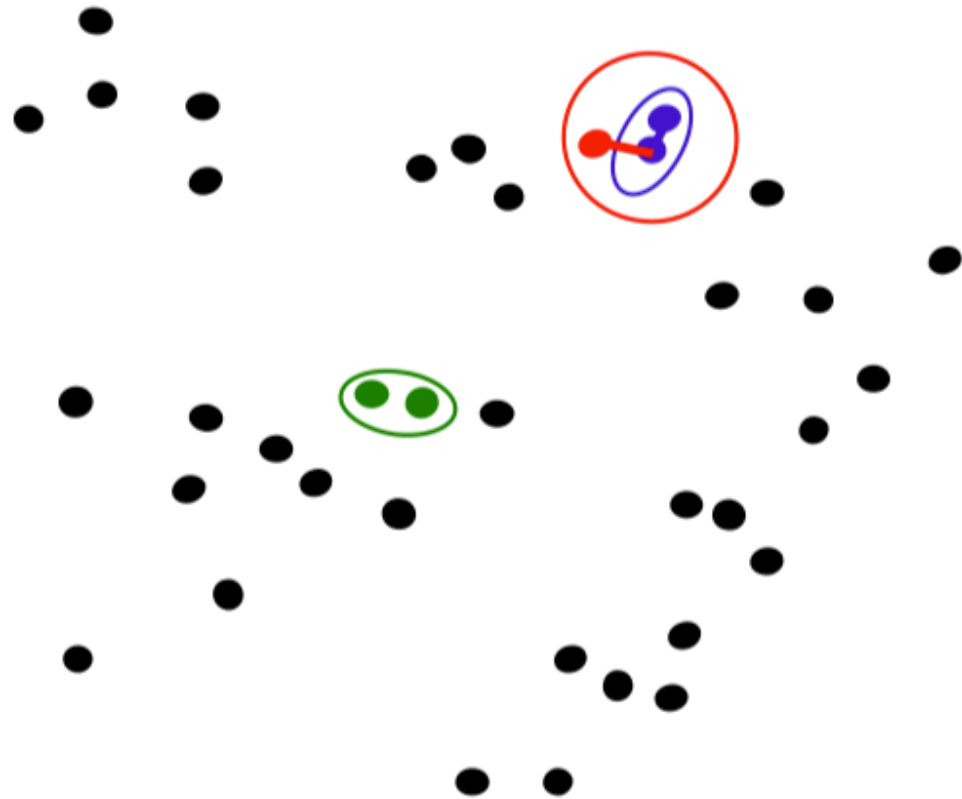
# Bottom-Up Agglomerative Clustering



1. Say "Every point is it's own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat




# Bottom-Up Agglomerative Clustering



1. Say "Every point is it's own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat

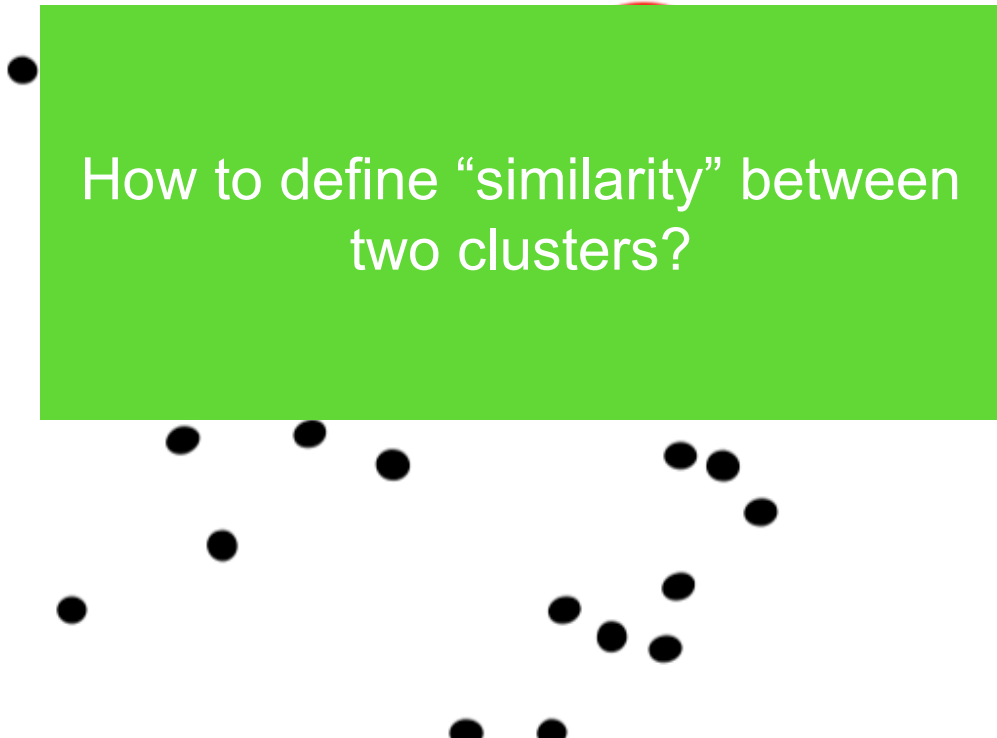


# Outline

- Overview
- Bottom-Up vs Top-Down Clustering
- Measuring Distance between Clusters 



# Key Question: Similarity Function



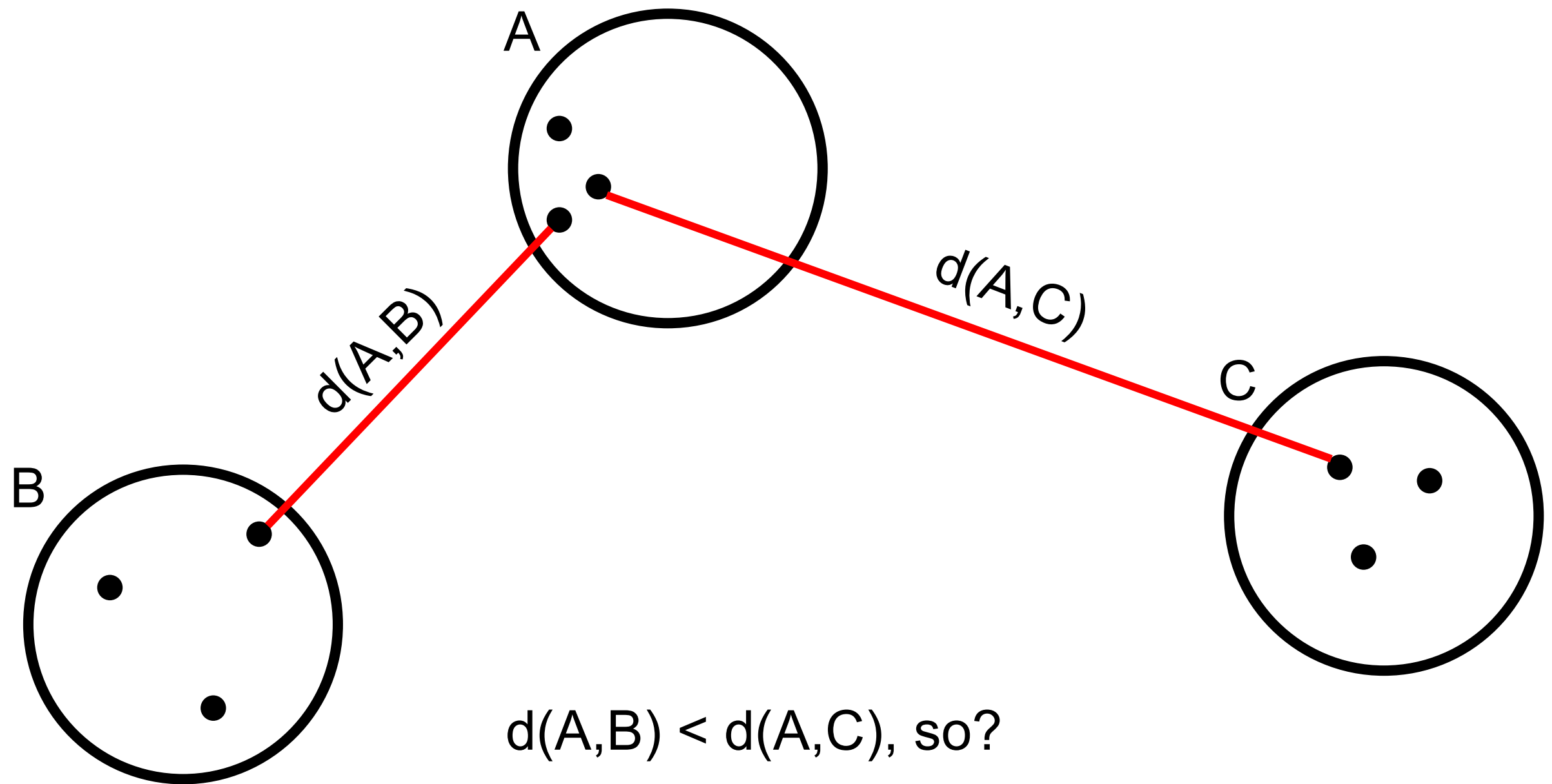
How to define “similarity” between two clusters?

A scatter plot of black dots is shown. A green rectangular box is overlaid on the plot, containing the text 'How to define “similarity” between two clusters?'. The dots are scattered, with a small cluster of four dots at the bottom right and several other dots scattered around.

1. Say “Every point is it’s own cluster”
2. Find “most similar” pair of clusters
3. Merge it into a parent cluster
4. Repeat

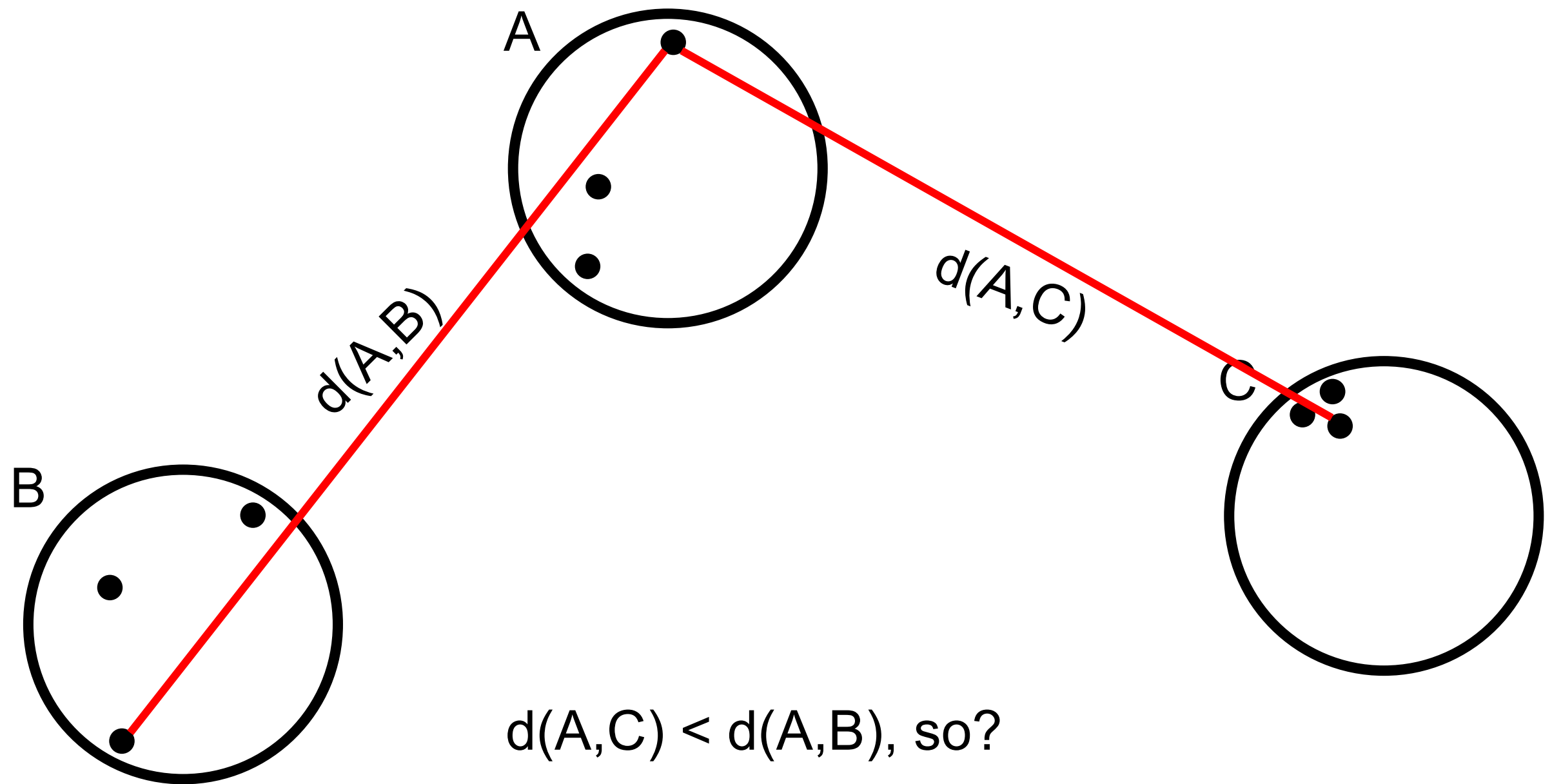


I am going to merge A with either B or C. Which one?



Single Link

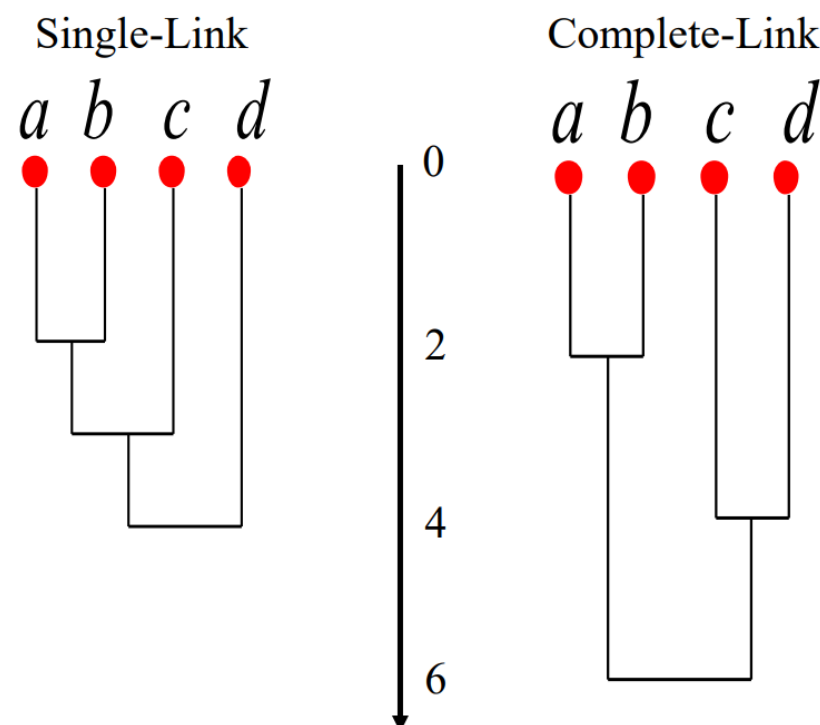
I am going to merge A with either B or C. Which one?



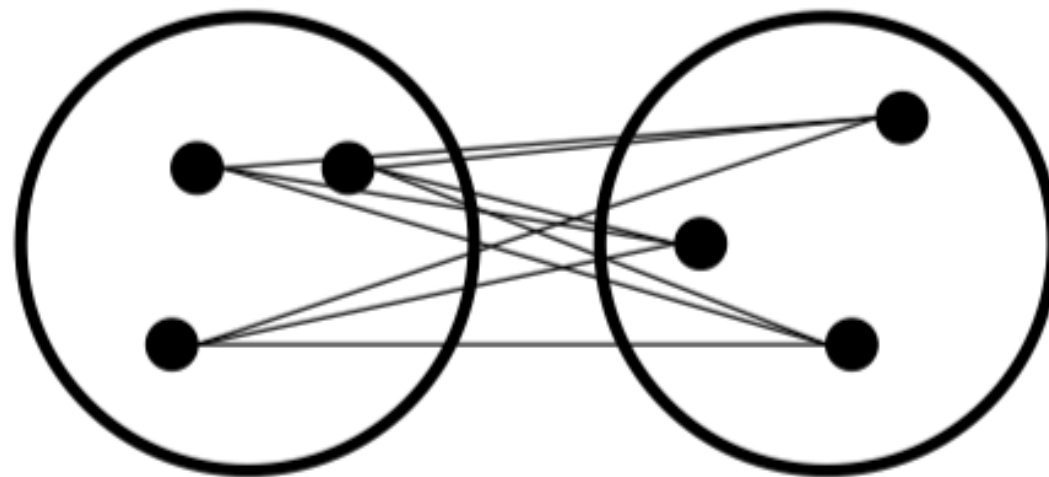
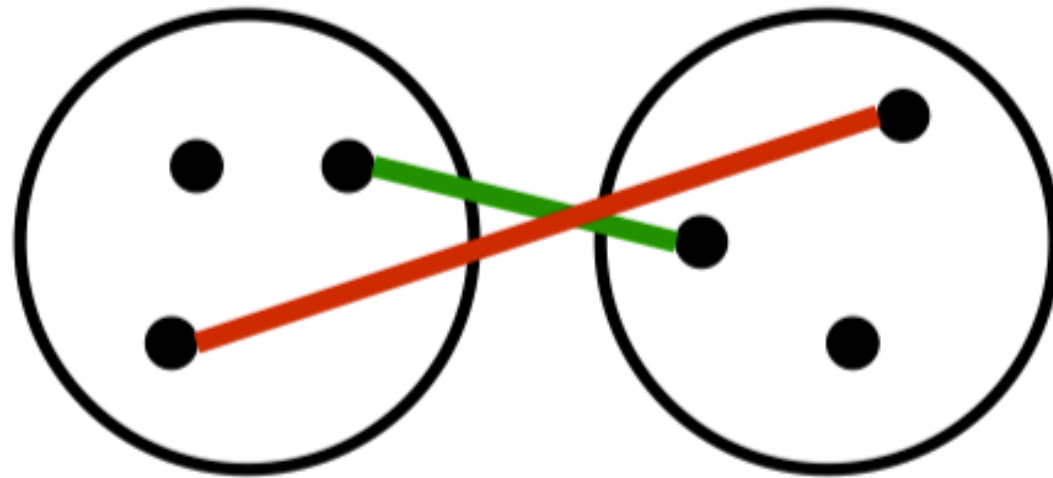
Complete Link

- **Single link:** A chain of points can be extended for long distances without regard to the overall shape of the emerging cluster. This effect is called *chaining*. It is also sensitive to outliers. It is faster in general.
- **Complete link:** Clusters are split into two groups of roughly equal size when we cut the dendrogram at the last merge. In general, this is a more useful organization of the data than a clustering with chains. It avoids chaining and more robust to outliers. Generally slower.
- **Average link:** When you don't know which one may be better for you, start it with the average link method.

## Dendrograms



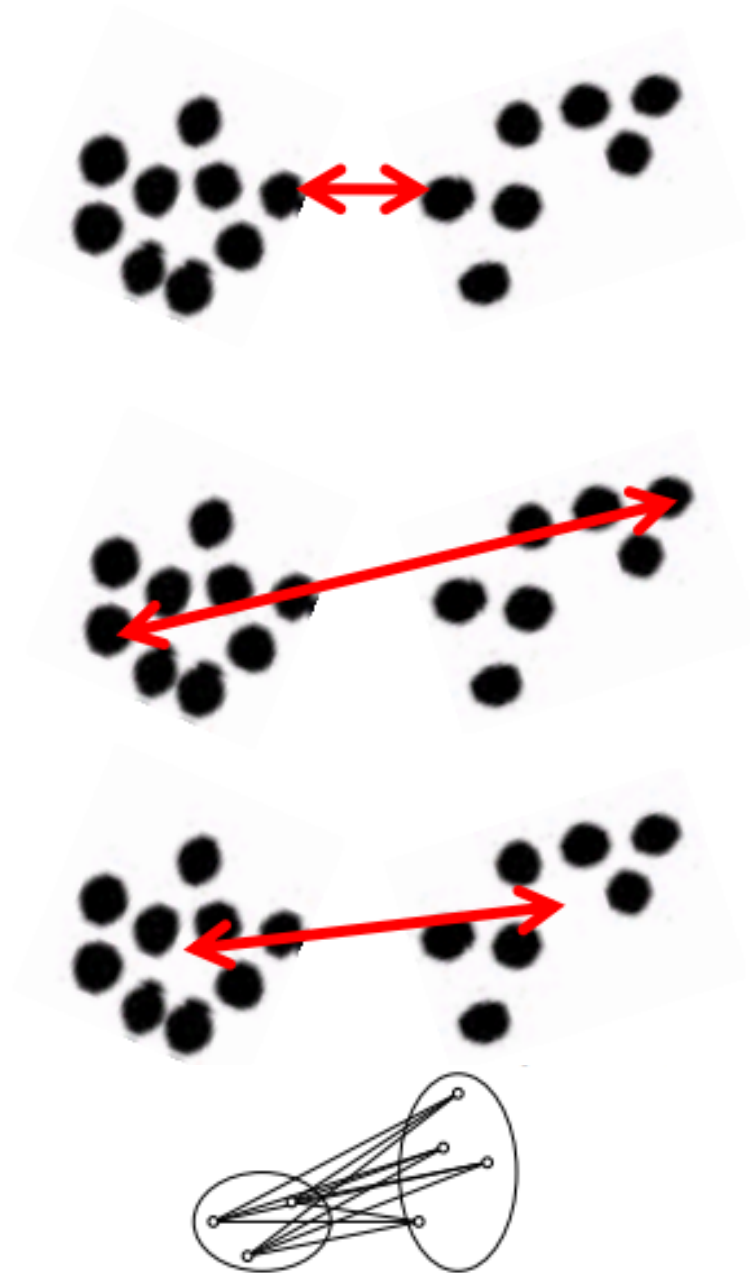
# How to Define Distance Between Two Clusters?



# Bottom-up Agglomerative clustering

Different algorithms differ in how the similarities are defined (and hence updated) between two clusters

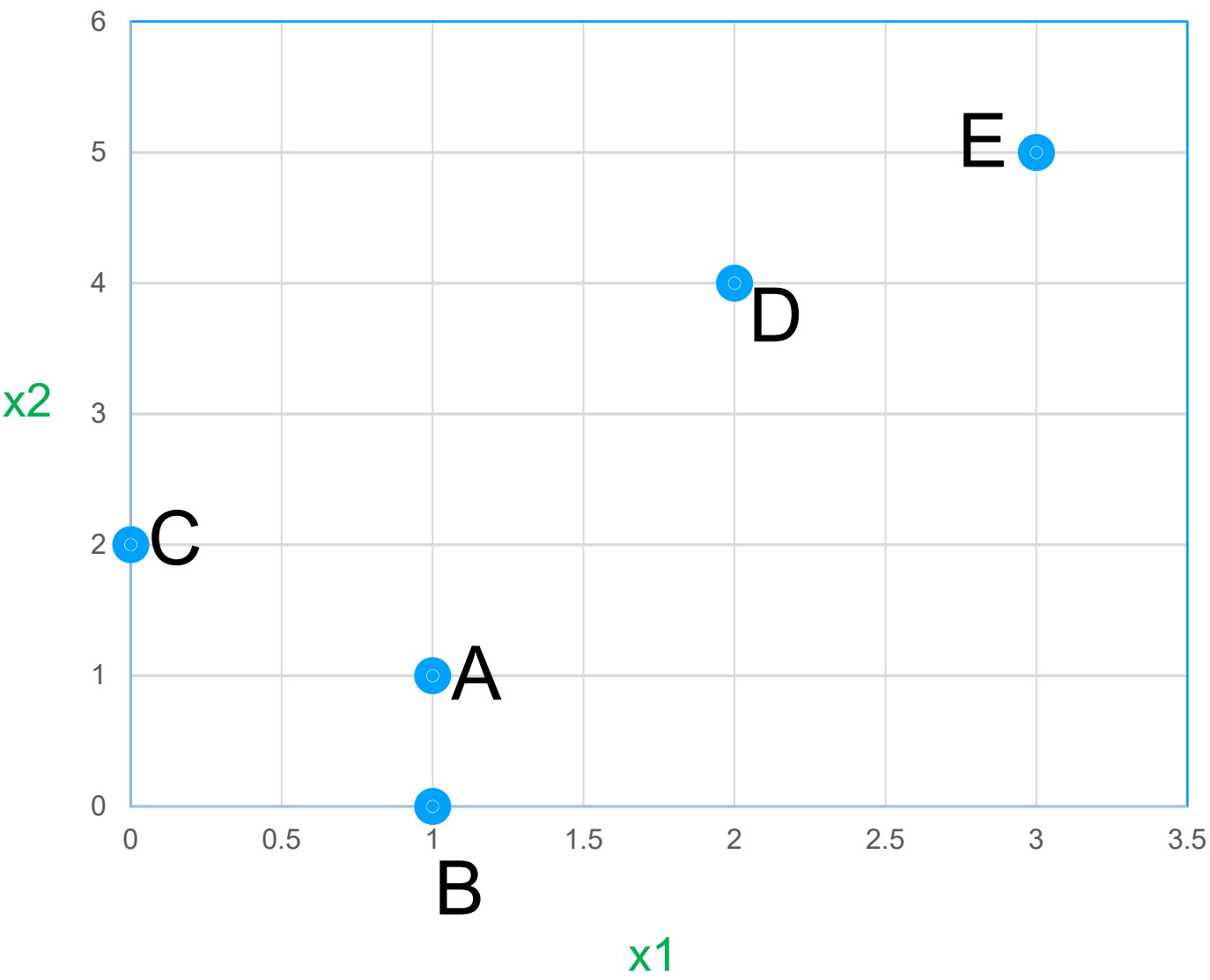
- Single-Link
  - Nearest Neighbor: similarity between their closest members.
- Complete-Link
  - Furthest Neighbor: similarity between their furthest members.
- Centroid
  - Similarity between the centers of gravity
- Average-Link
  - Average similarity of all cross-cluster pairs.



# Distance Between Clusters

**Different distance functions can lead to different results!**

i	X1	X2
A	1	1
B	1	0
C	0	2
D	2	4
E	3	5



**EUCLIDEAN DISTANCE**

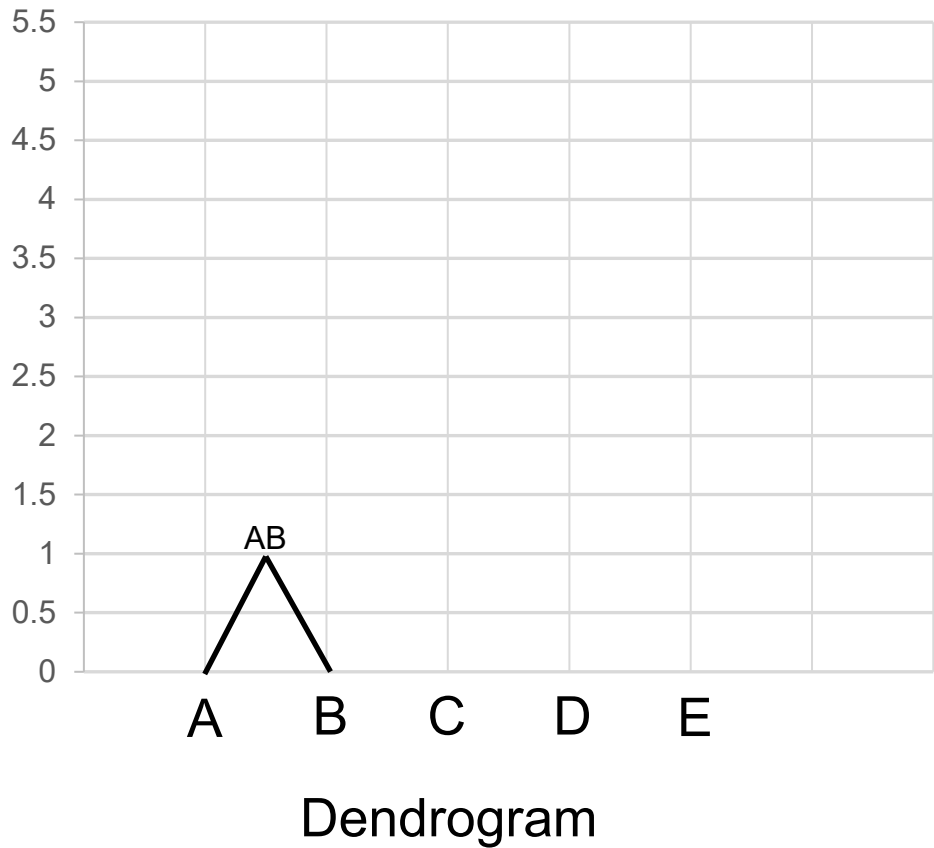
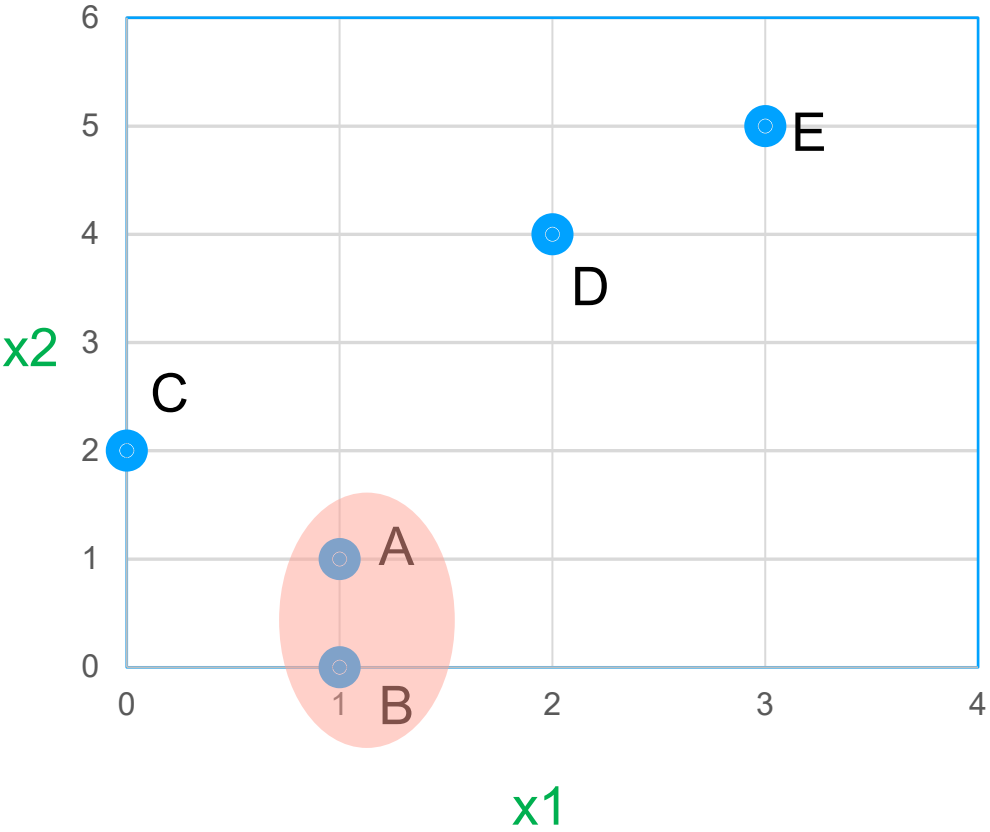
	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0



# Distance based on Average point (Bottom-Up Clustering)

	A	B	C	D	E
A	0	1	1.4	3.2	4.5
B	1	0	2.2	4.1	5.4
C	1.4	2.2	0	2.8	4.2
D	3.2	4.1	2.8	0	1.4
E	4.5	5.4	4.2	1.4	0

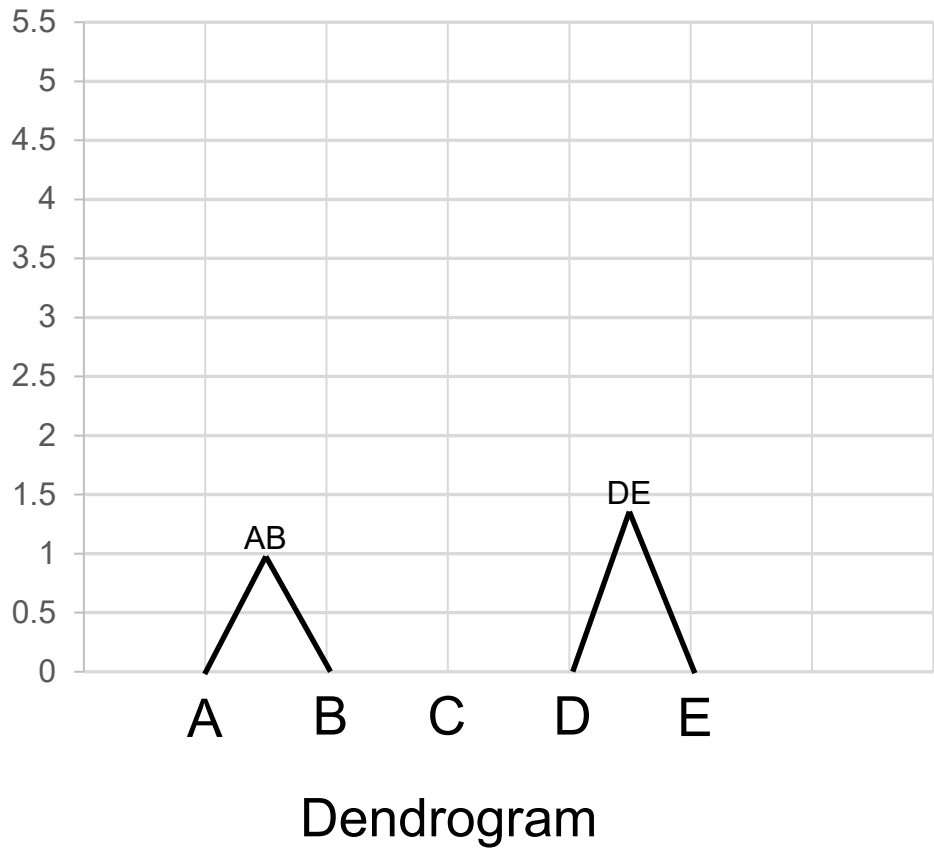
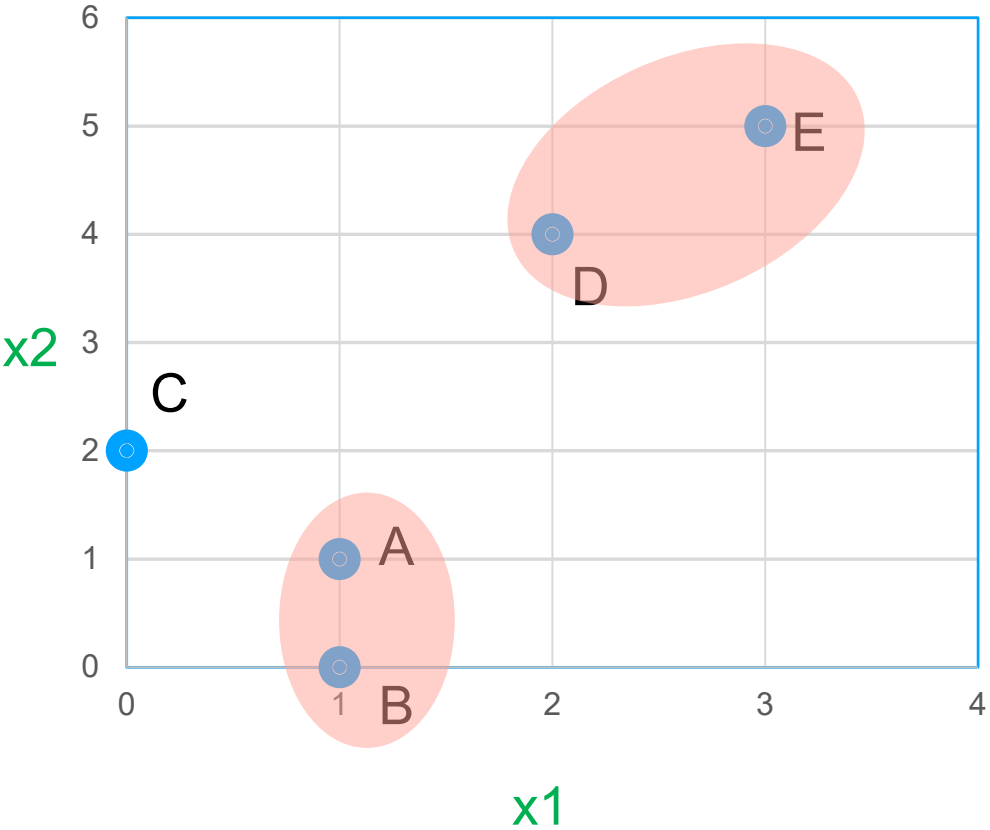
	(A,B)	C	D	E
(A,B)	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0



# Distance based on average point (Bottom-Up Clustering)

	(A,B)	C	D	E
(A,B)	0	1.8	3.6	4.9
C	1.8	0	2.8	4.2
D	3.6	2.8	0	1.4
E	4.9	4.2	1.4	0

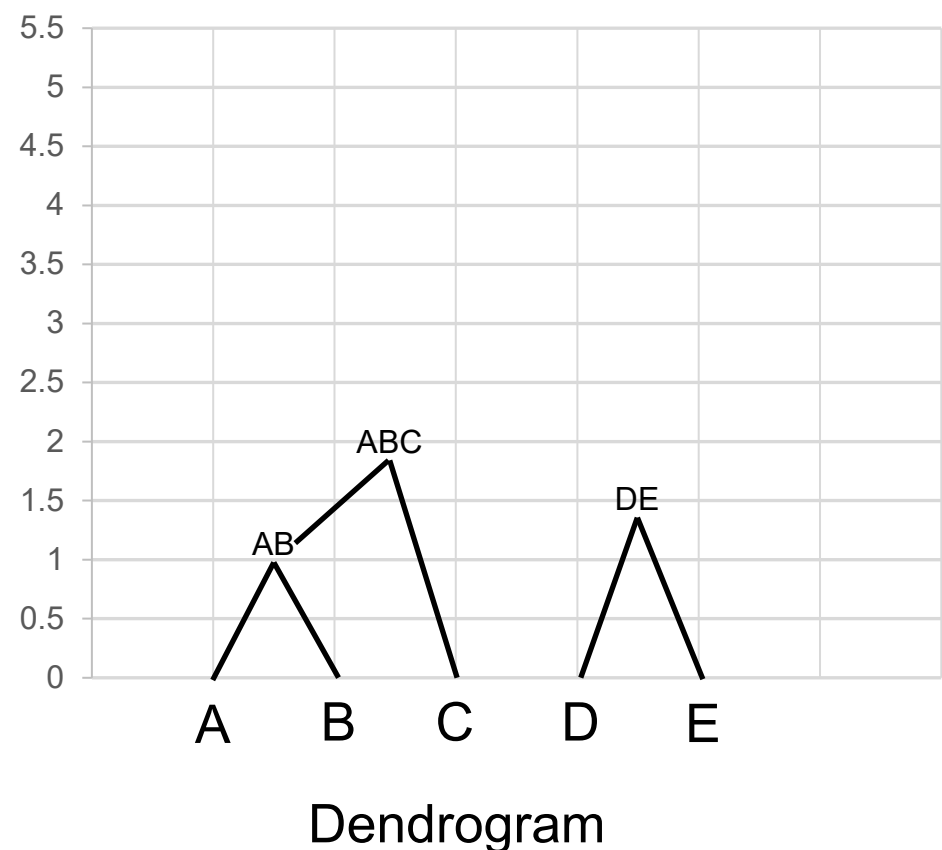
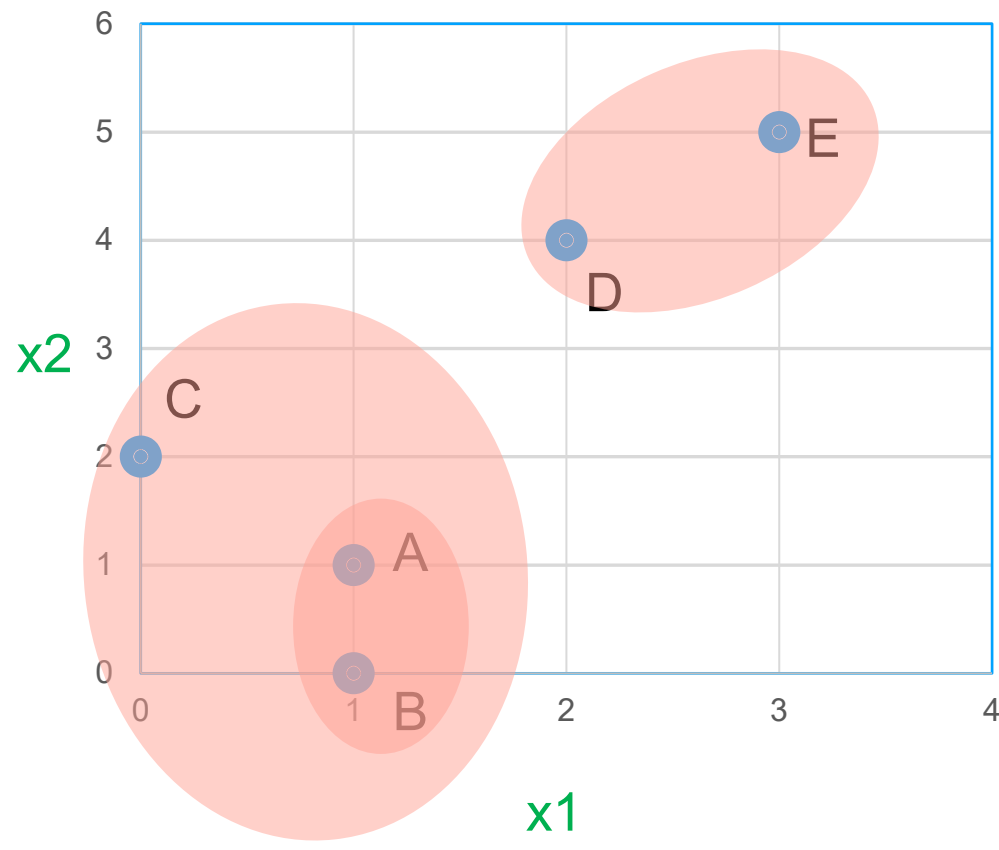
	(A,B)	C	(D,E)
(A,B)	0	1.8	4.25
C	1.8	0	3.5
(D,E)	4.25	3.5	0



# Distance based on average point (Bottom-Up Clustering)

	(A,B)	C	(D,E)
(A,B)	0	1.8	4.25
C	1.8	0	3.5
(D,E)	4.25	3.5	0

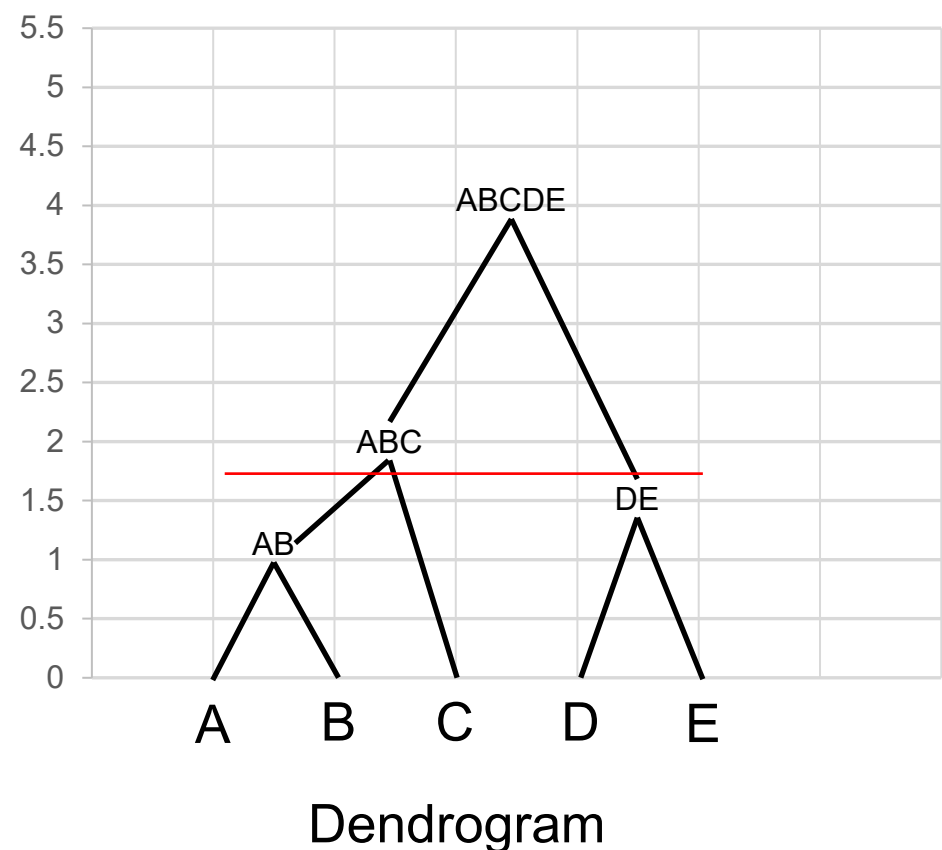
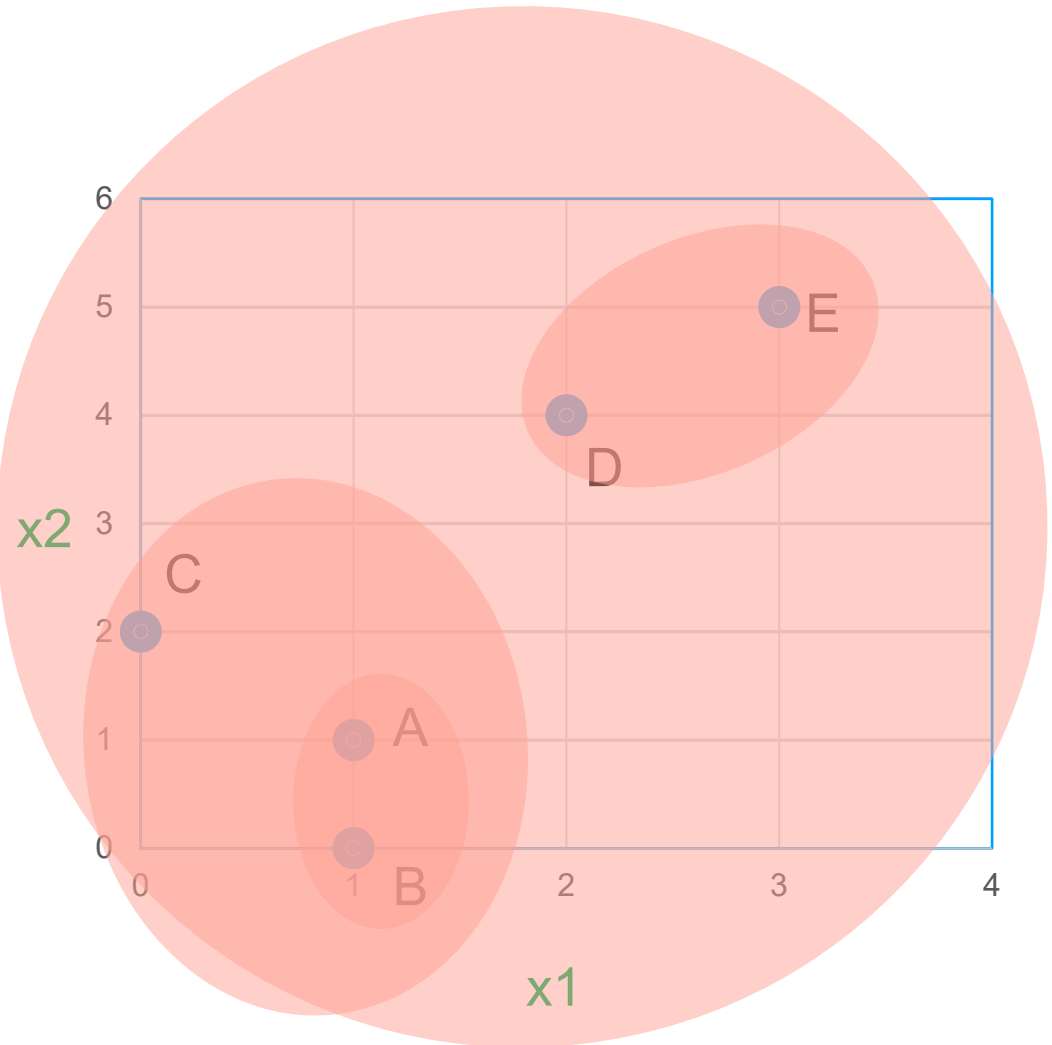
	((A,B),C)	(D,E)
((A,B),C)	0	3.875
(D,E)	3.875	0



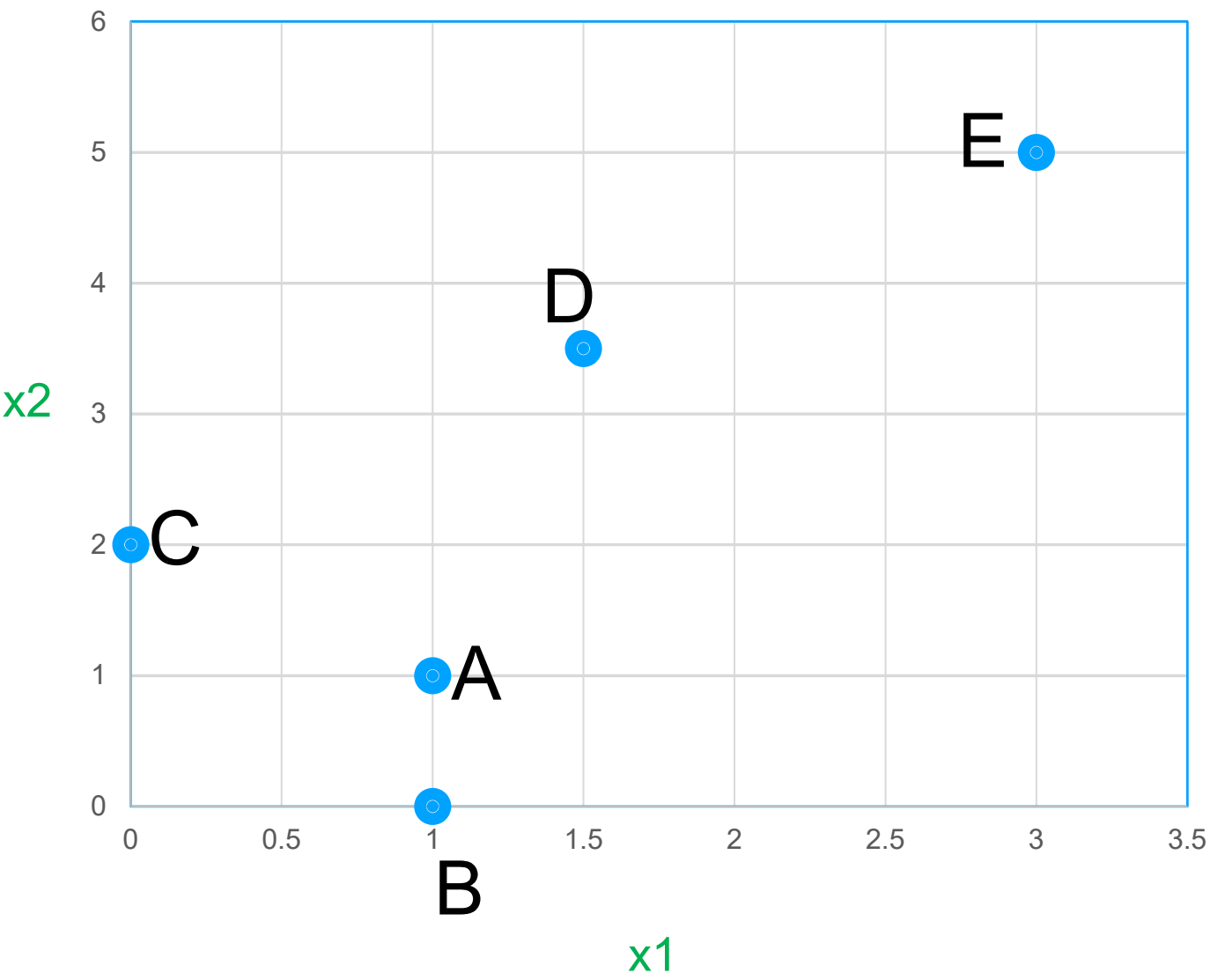
# Distance based on average point (Bottom-Up Clustering)

	$((A,B),C)$	$(D,E)$
$((A,B),C)$	0	3.875
$(D,E)$	3.875	0

	$((((A,B),C),D),E)$
$((((A,B),C),D),E)$	0



i	X1	X2
A	1	1
B	1	0
C	0	2
D	1.5	3.5
E	3	5



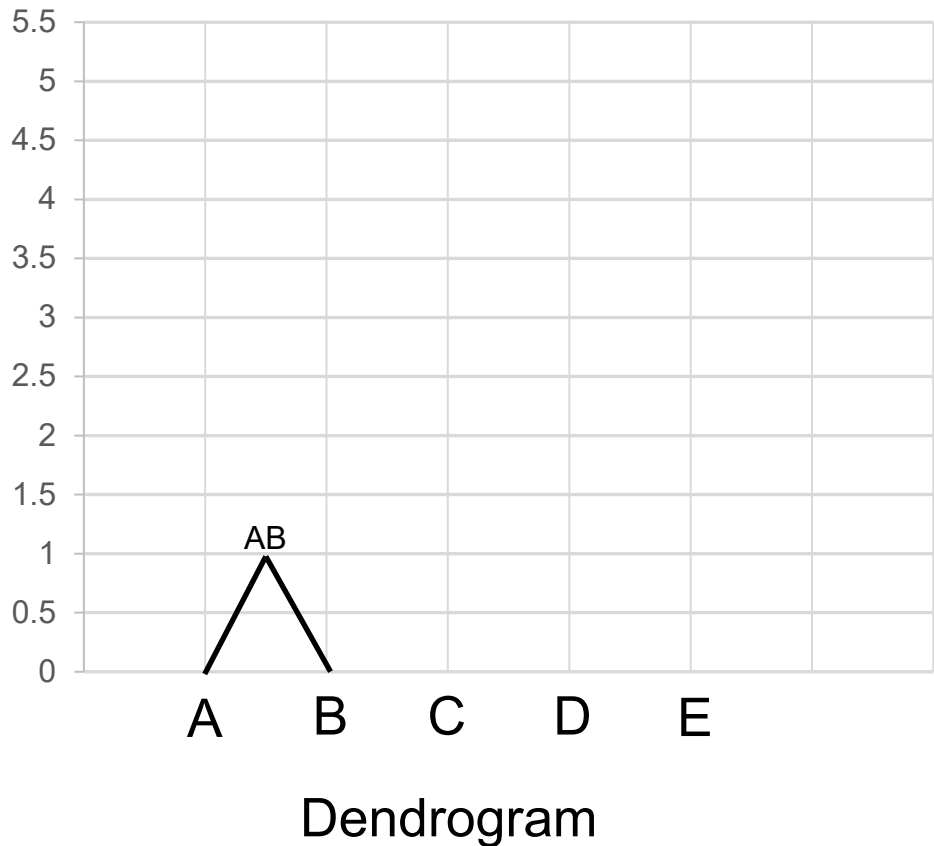
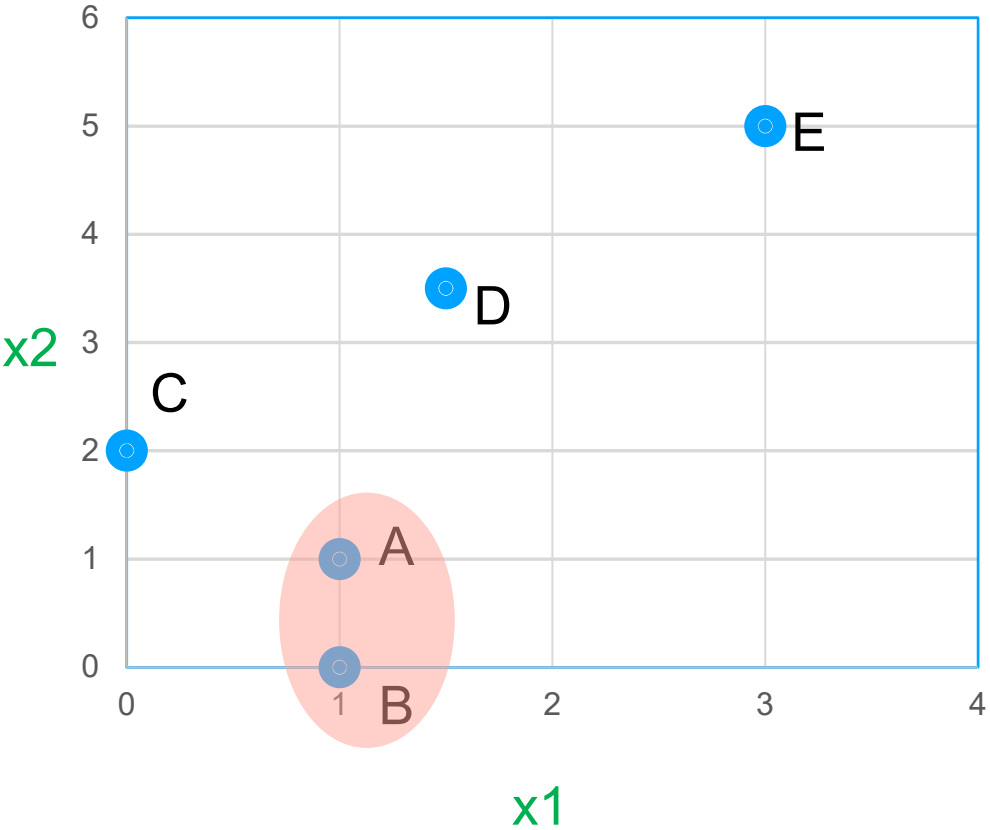
**EUCLIDEAN DISTANCE**

	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

# Distance based on Single Link (Bottom-Up Clustering)

	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

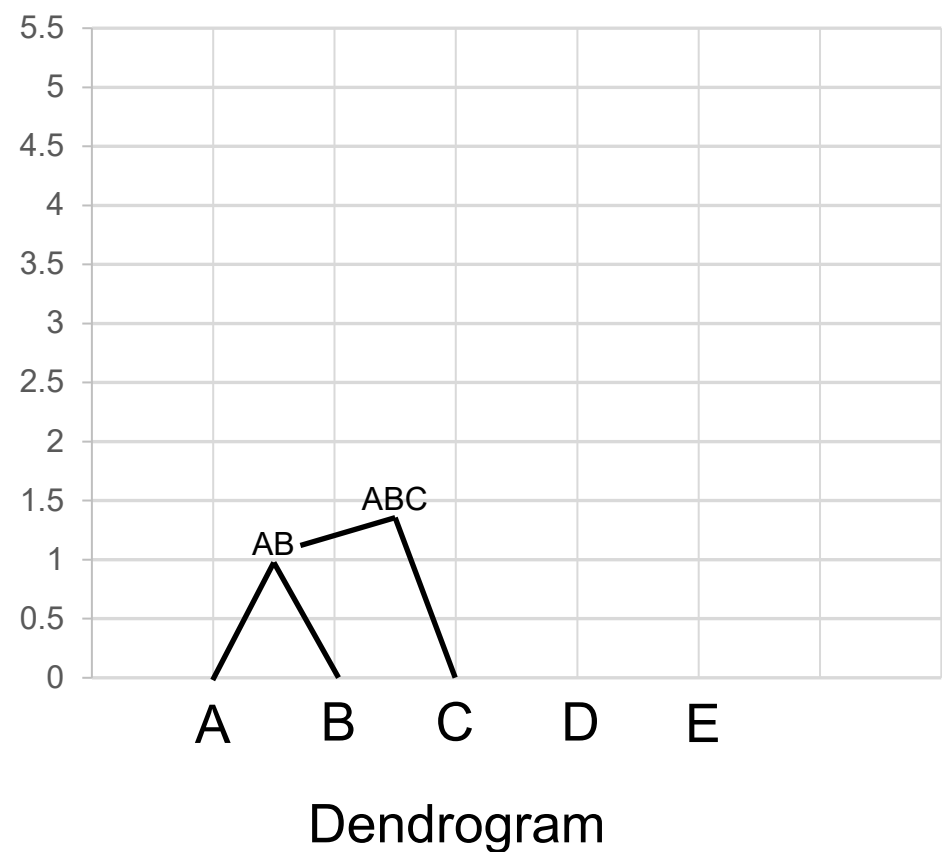
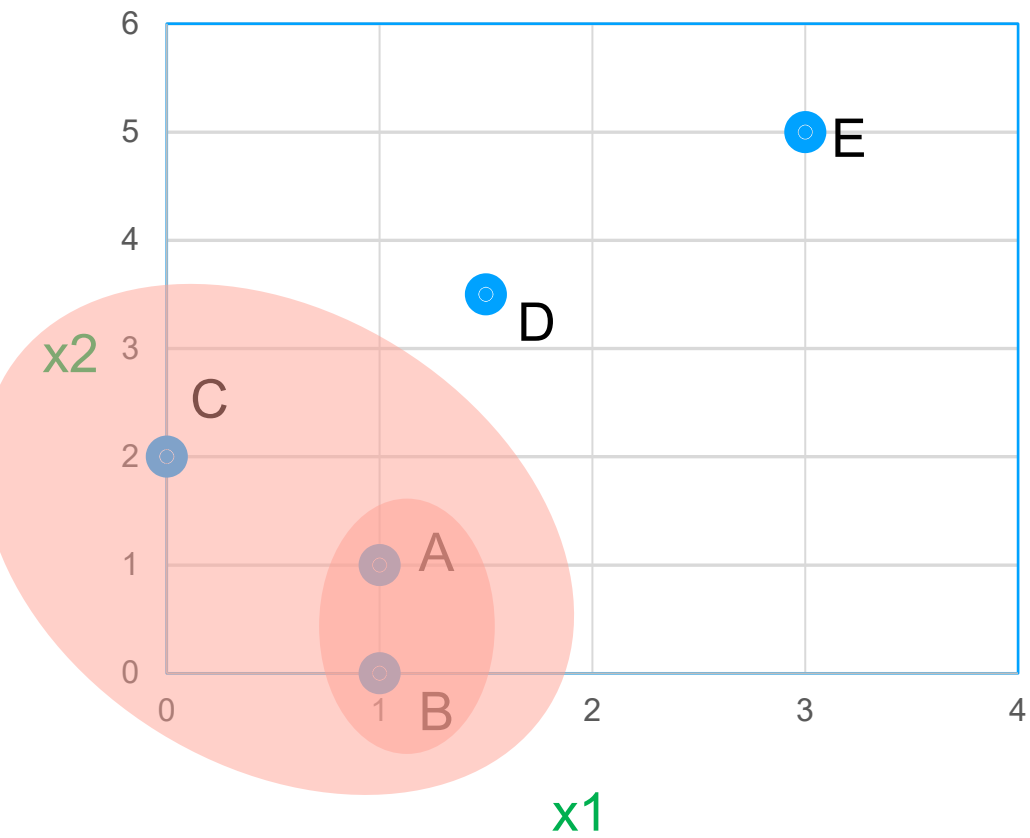
	(A,B)	C	D	E
(A,B)	0	1.4	2.55	4.5
C	1.4	0	2.12	4.2
D	2.55	2.12	0	2.12
E	4.5	4.2	2.12	0



# Distance based on Single Link (Bottom-Up Clustering)

	(A,B)	C	D	E
(A,B)	0	1.4	2.55	4.5
C	1.4	0	2.12	4.2
D	2.55	2.12	0	2.12
E	4.5	4.2	2.12	0

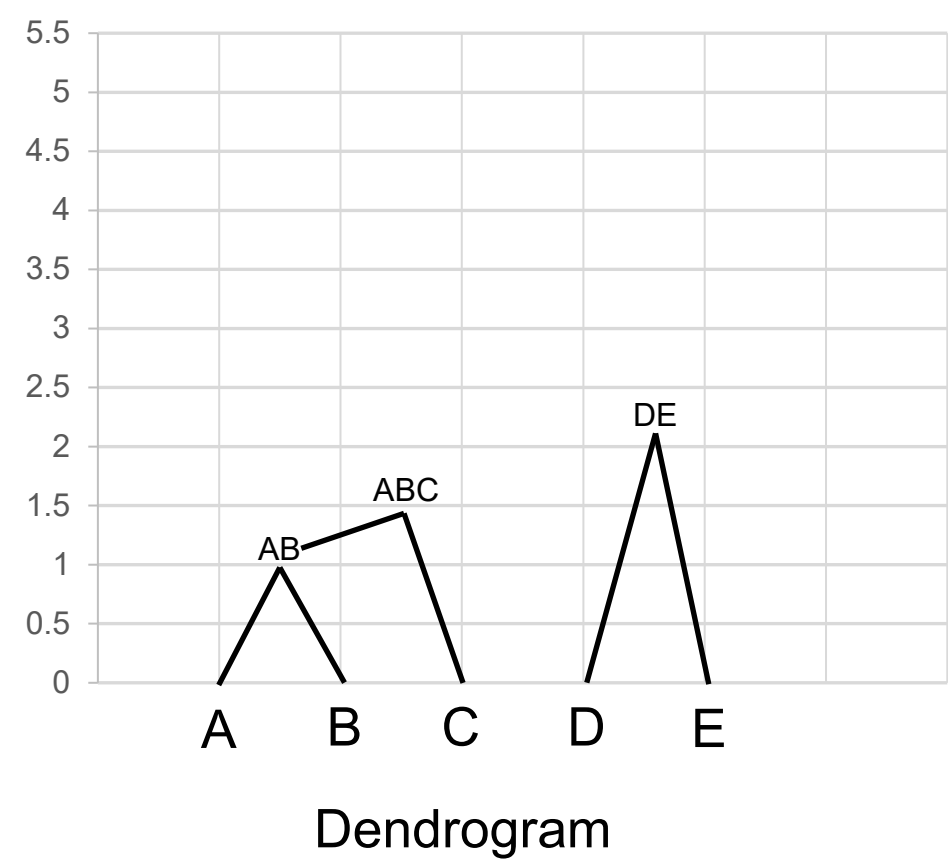
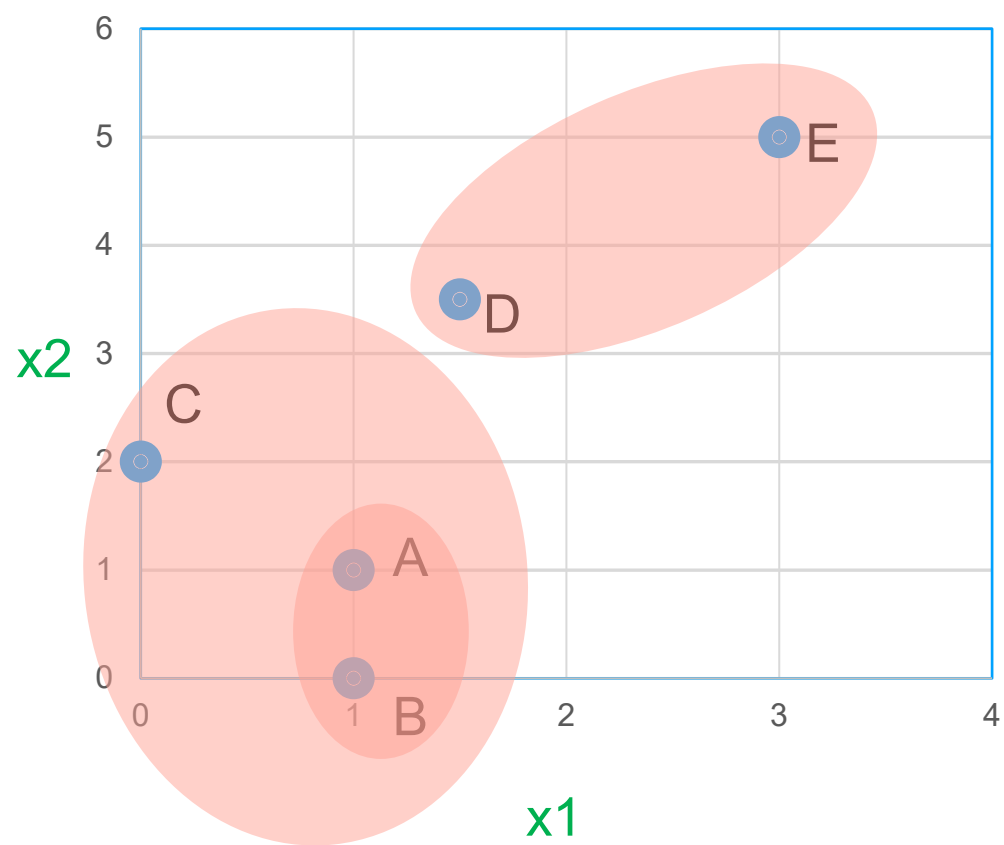
	(A,B),C	D	E
(A,B),C	0	2.12	4.2
D	2.12	0	2.12
E	4.2	2.12	0



# Distance based on Single Link (Bottom-Up Clustering)

	(A,B), C	D	E
(A,B), C	0	2.12	4.2
D	2.12	0	2.12
E	4.2	2.12	0

	((A,B),C)	(D,E)
((A,B),C)	0	2.12
(D,E)	2.12	0

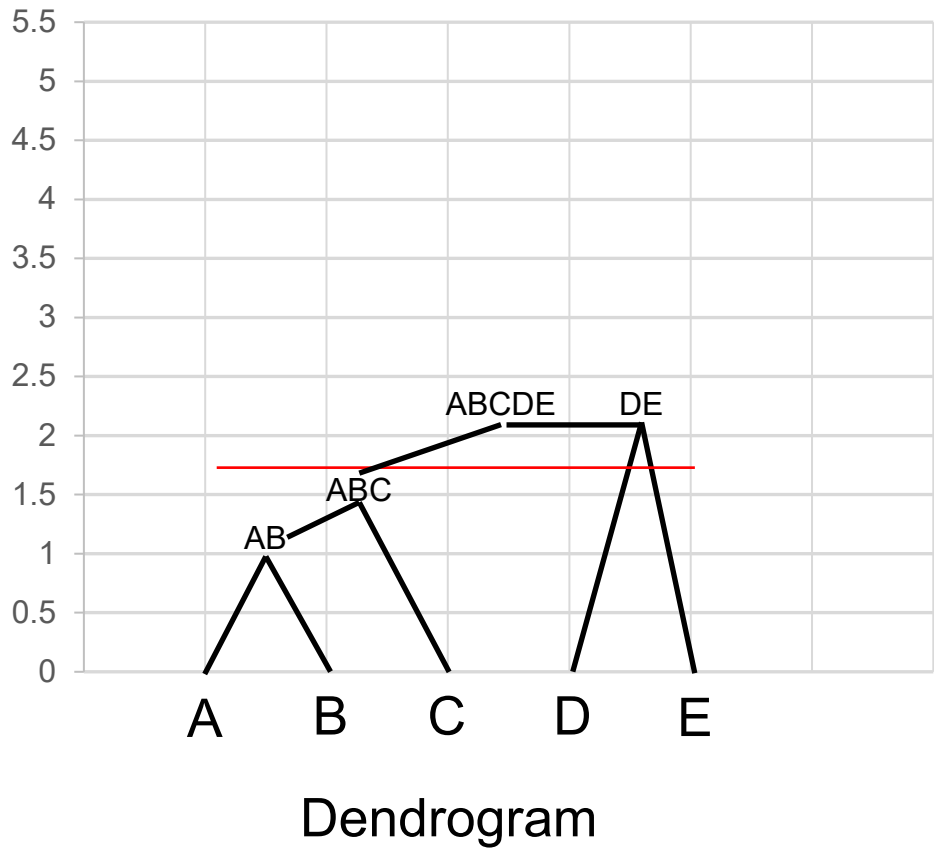
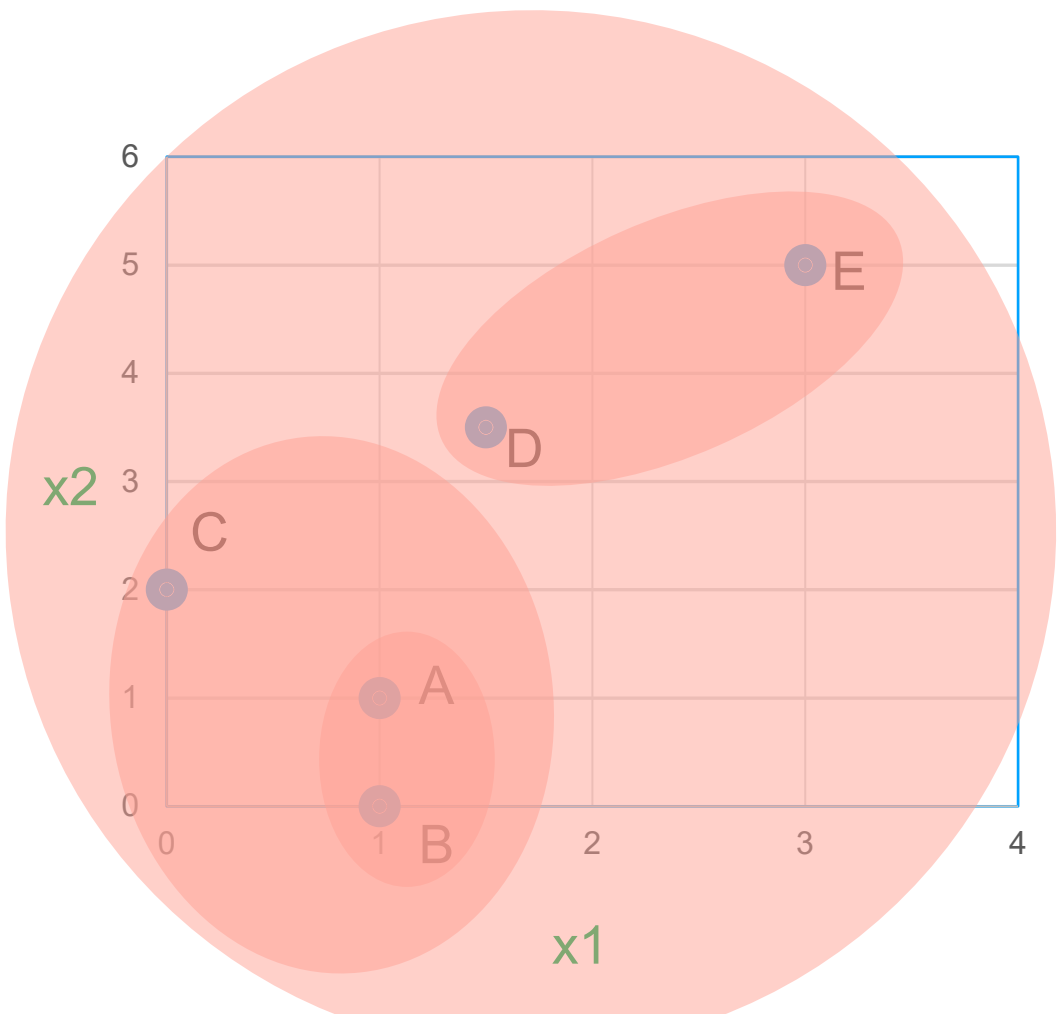




# Distance based on Single Link (Bottom-Up Clustering)

	$((A,B),C)$	$(D,E)$
$((A,B),C)$	0	2.12
$(D,E)$	2.12	0

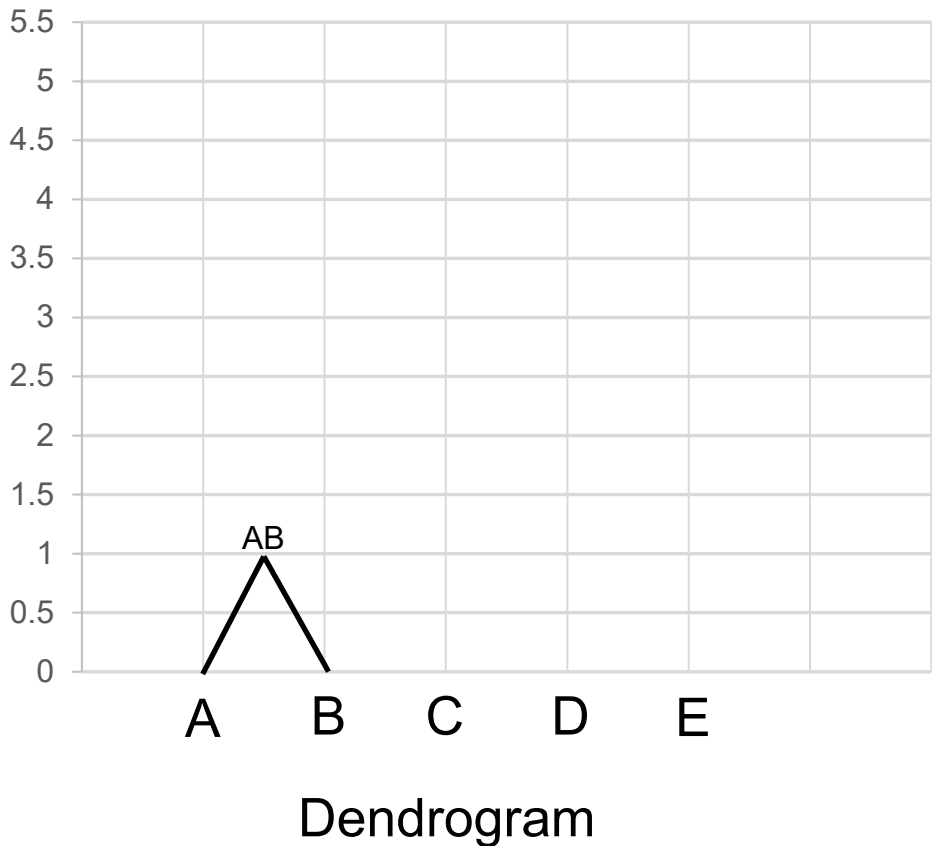
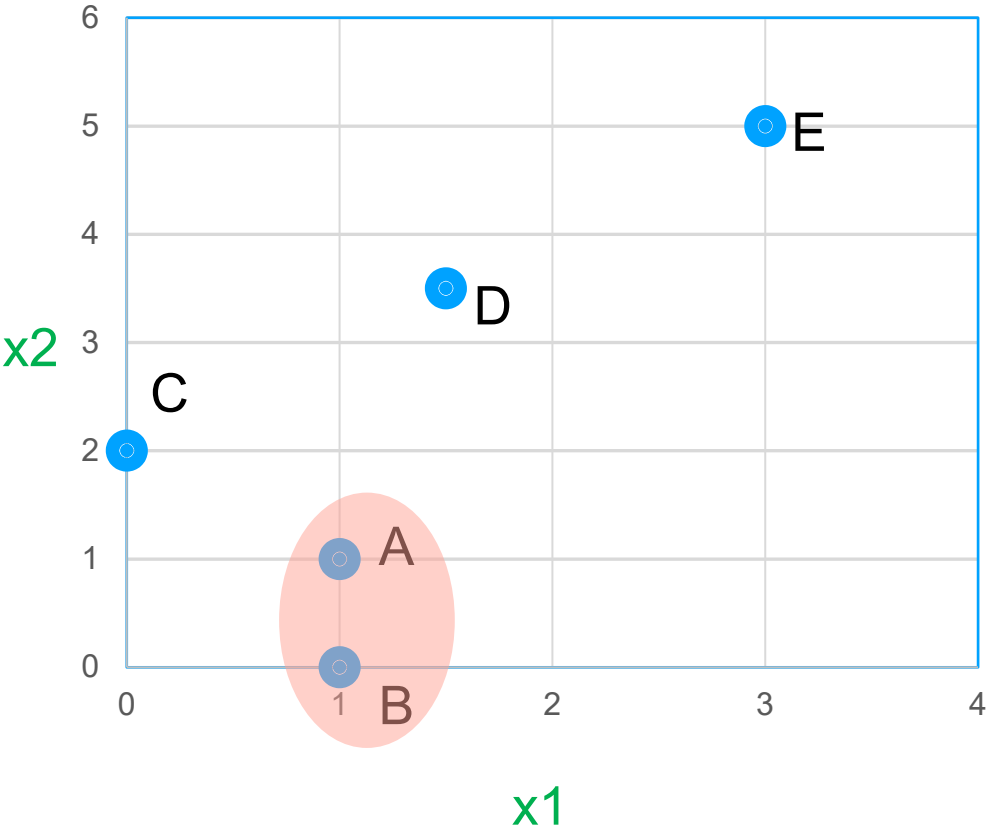
	$((((A,B),C),D),E)$
$((((A,B),C),D),E)$	0



# Distance based on Complete Link (Bottom-Up Clustering)

	A	B	C	D	E
A	0	1	1.4	2.55	4.5
B	1	0	2.2	3.53	5.4
C	1.4	2.2	0	2.12	4.2
D	2.55	3.53	2.12	0	2.12
E	4.5	5.4	4.2	2.12	0

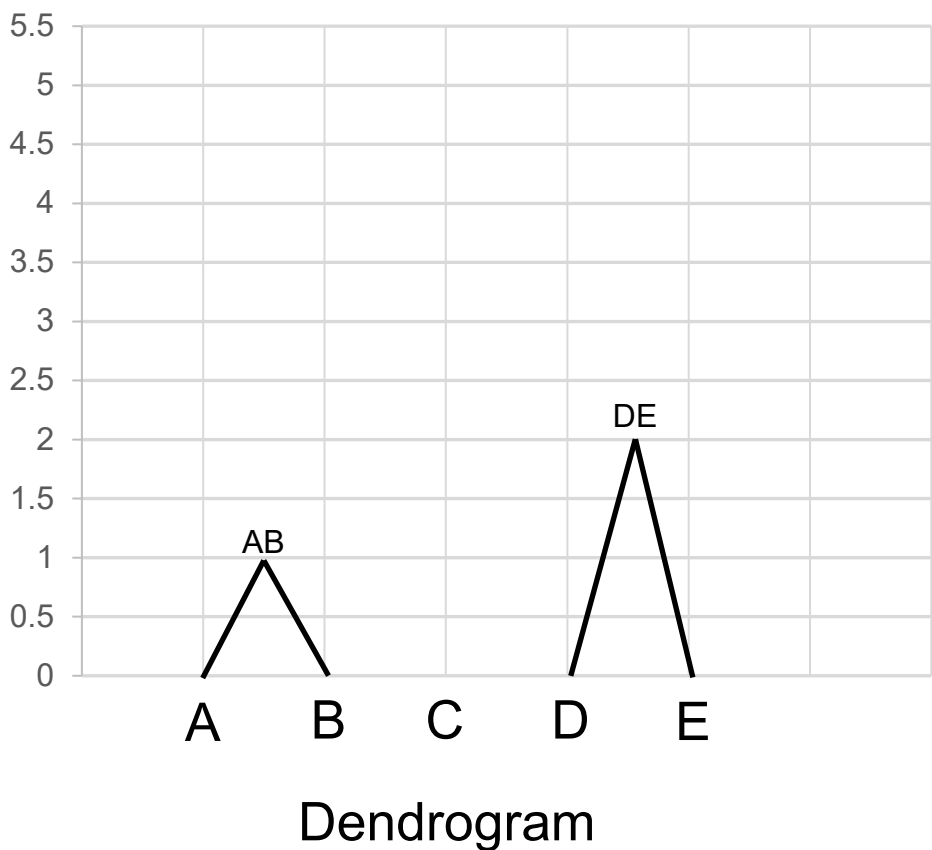
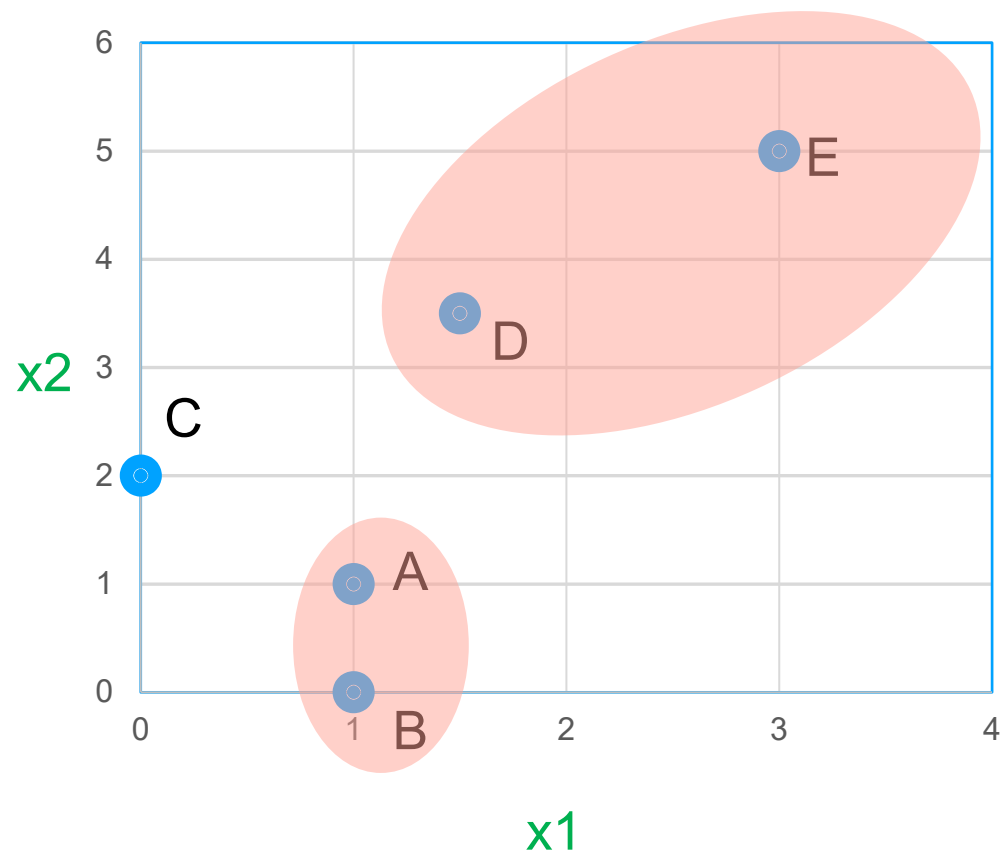
	(A,B)	C	D	E
(A,B)	0	2.2	3.55	5.4
C	2.2	0	2.12	4.2
D	3.55	2.12	0	2.12
E	5.4	4.2	2.12	0



# Distance based on Complete Link (Bottom-Up Clustering)

	(A,B)	C	D	E
(A,B)	0	2.2	3.55	5.4
C	2.2	0	2.12	4.2
D	3.55	2.12	0	2.12
E	5.4	4.2	2.12	0

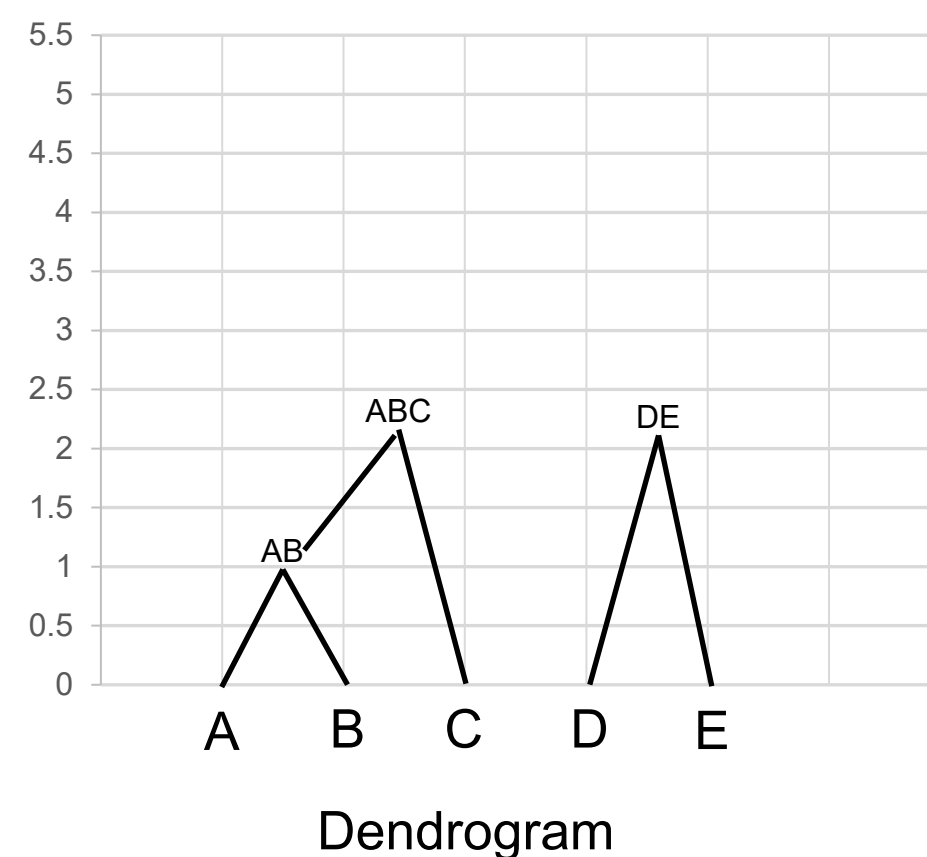
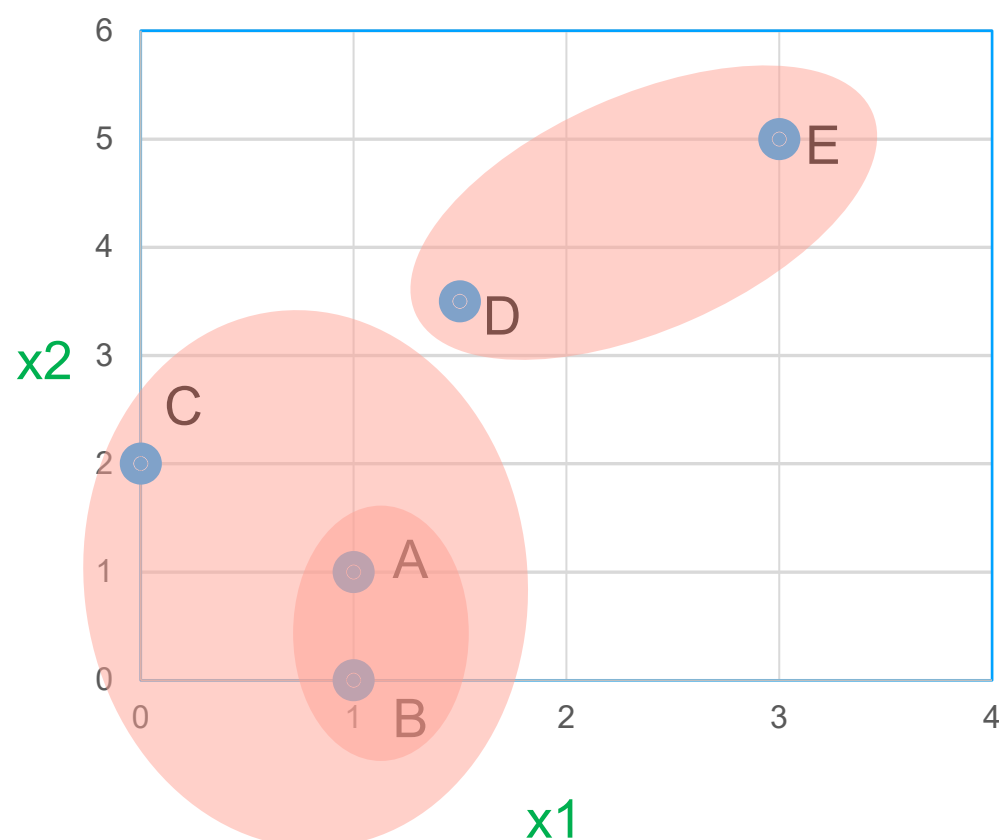
	(A,B)	C	(D,E)
(A,B)	0	2.2	5.4
C	2.2	0	4.2
(D,E)	5.4	4.2	0



# Distance based on Single Link (Bottom-Up Clustering)

	(A,B)	C	(D,E)
(A,B)	0	2.2	5.4
C	2.2	0	4.2
(D,E)	5.4	4.2	0

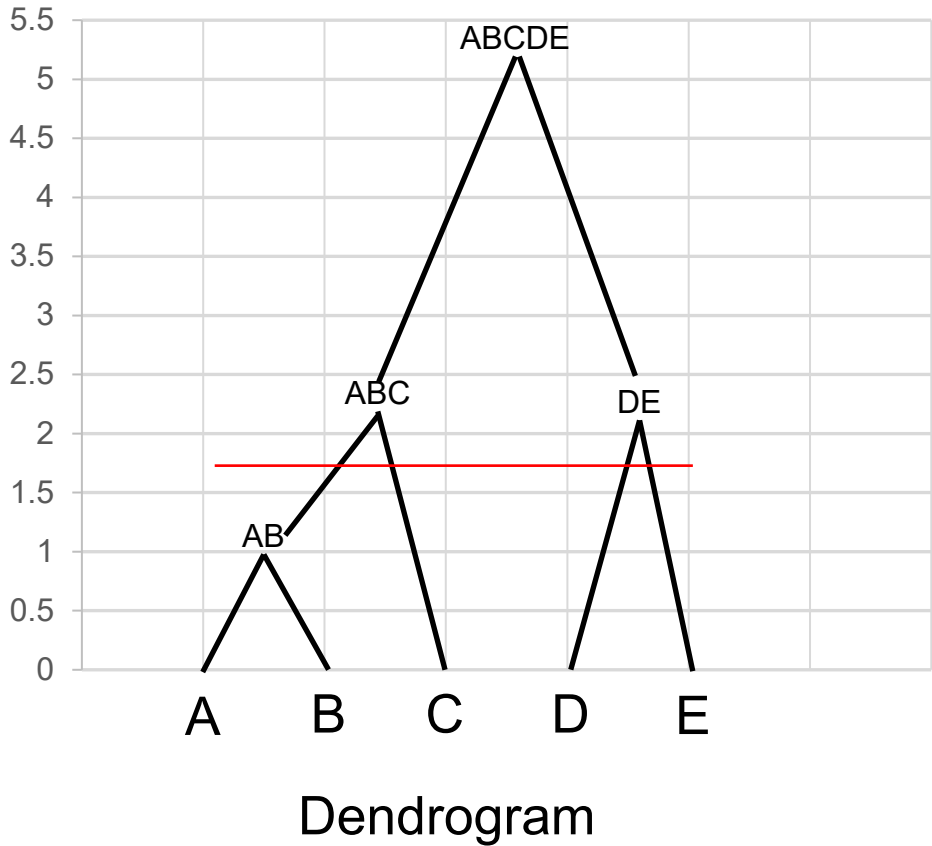
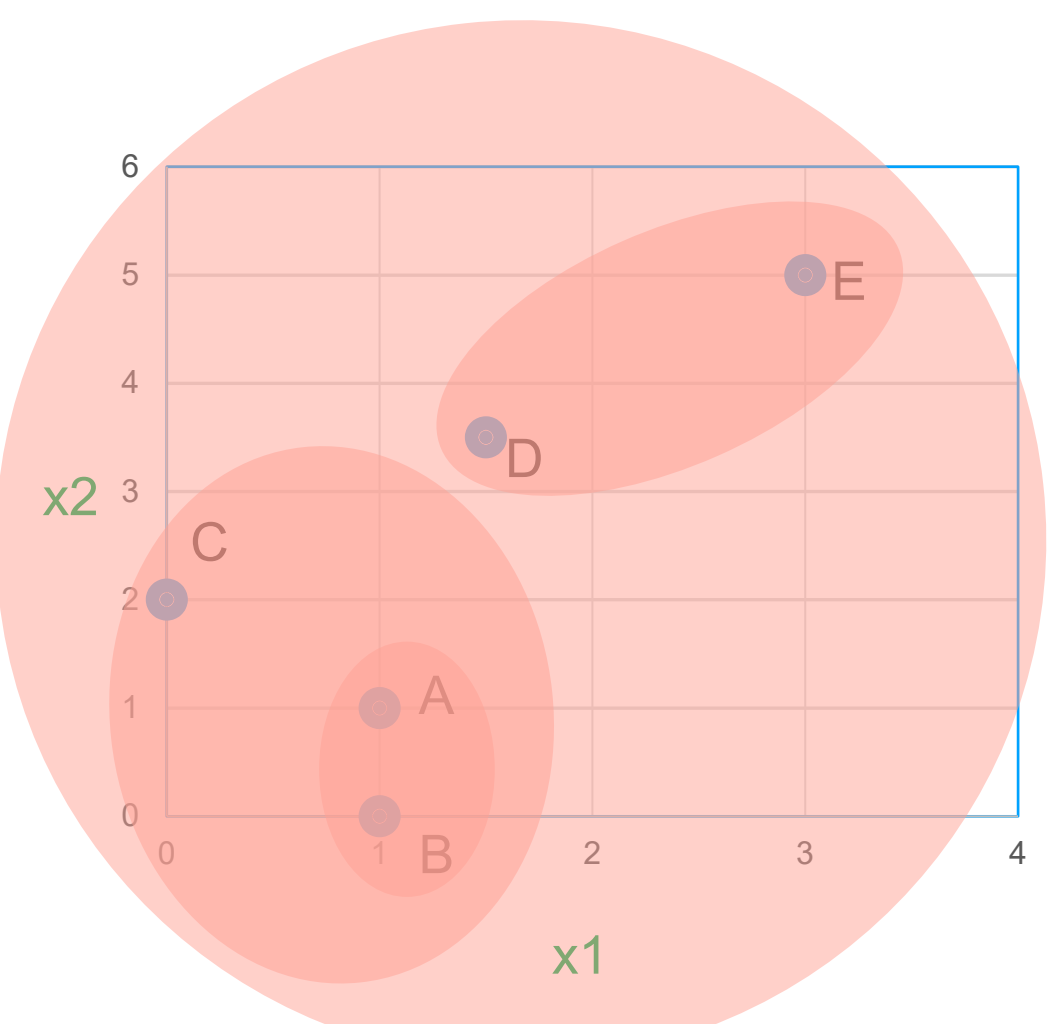
	((A,B),C)	(D,E)
((A,B),C)	0	5.4
(D,E)	5.4	0

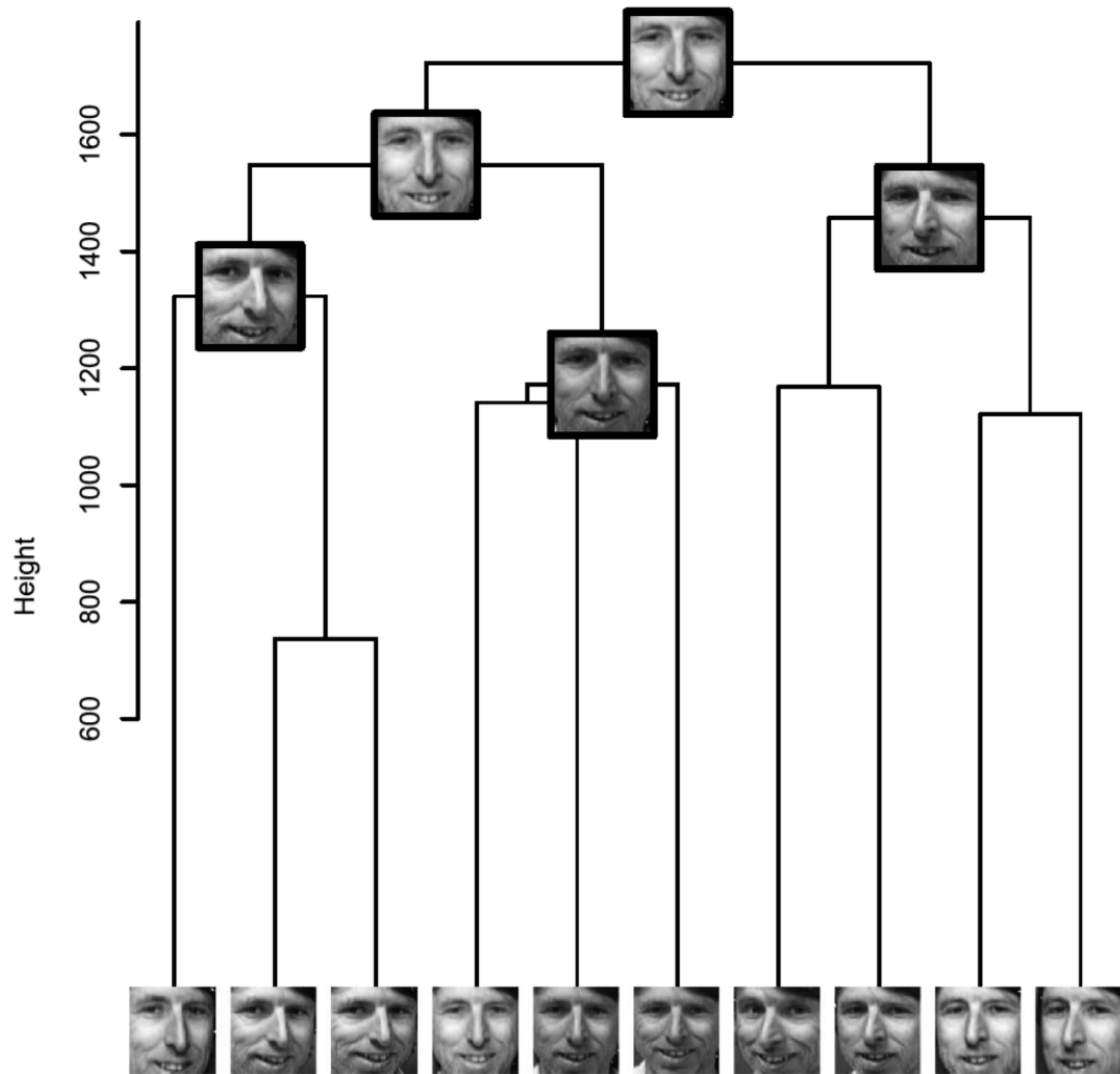


# Distance based on Complete Link (Bottom-Up Clustering)

	$((A,B),C)$	$(D,E)$
$((A,B),C)$	0	5.4
$(D,E)$	5.4	0

	$((((A,B),C),D),E)$
$((((A,B),C),D),E)$	0

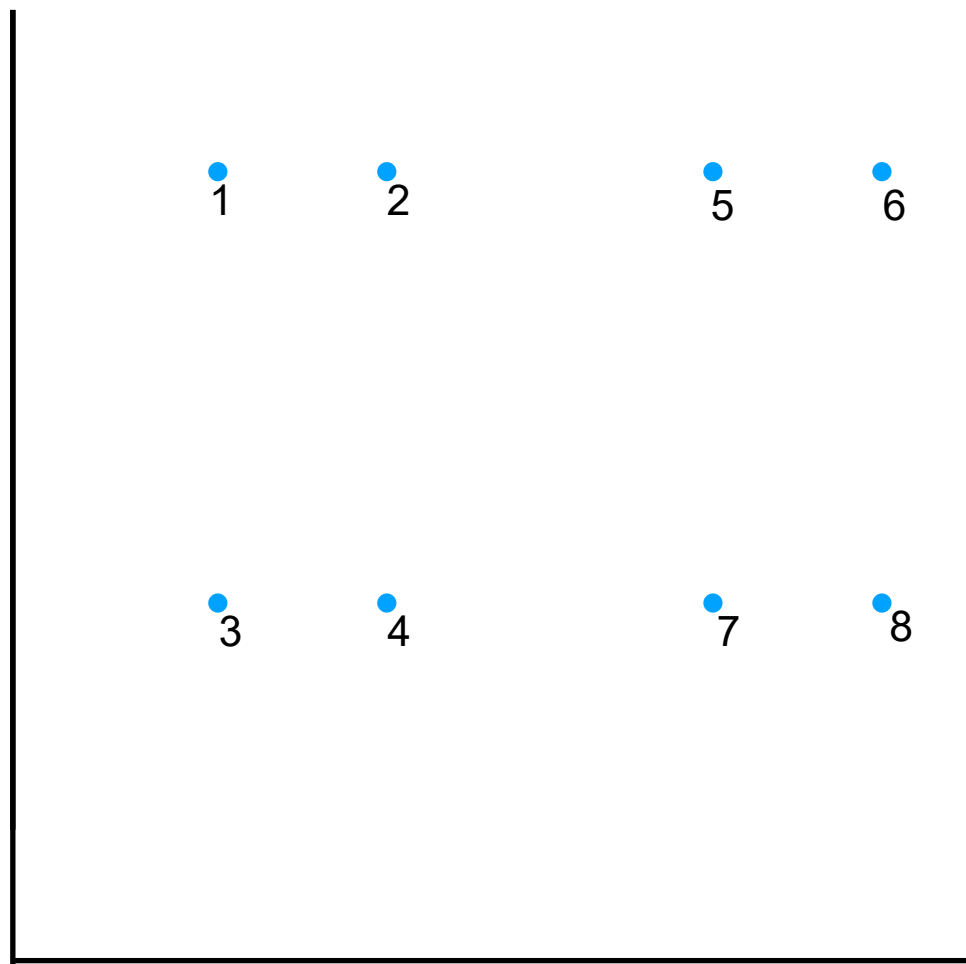




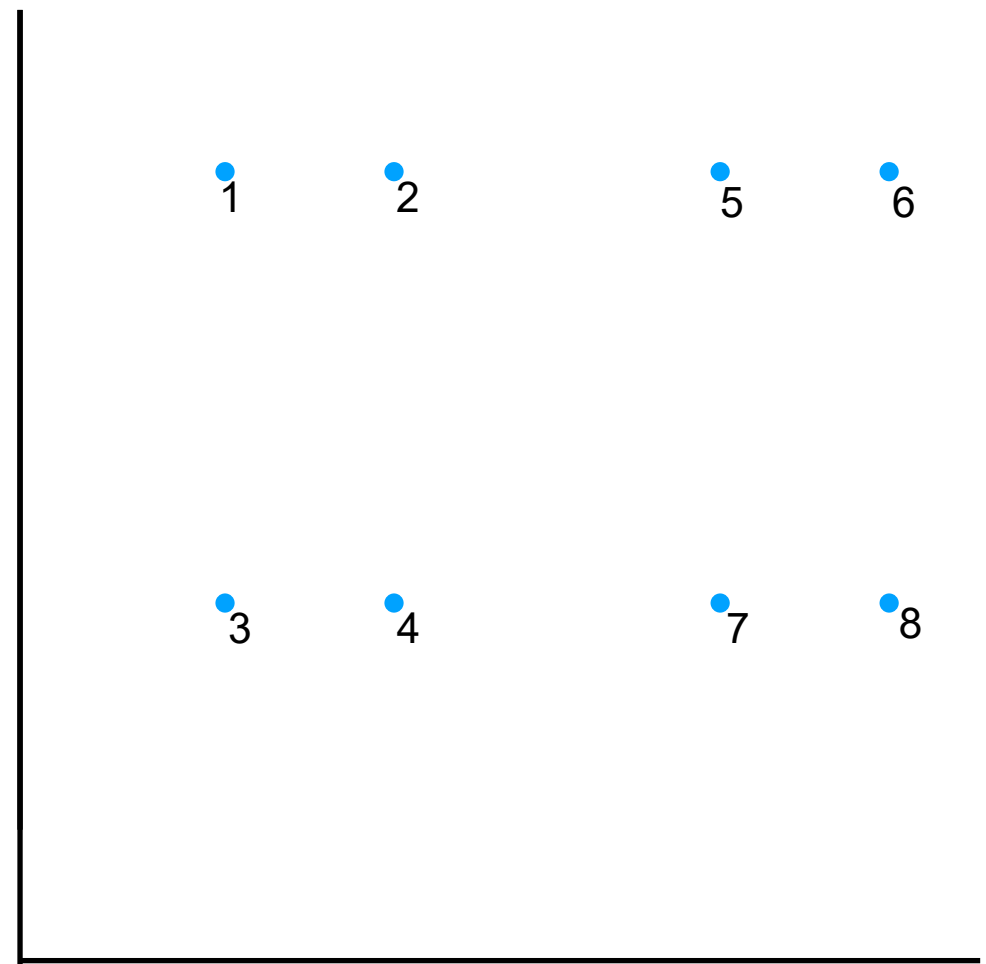
(From Bien et al. (2011))

# Another Example

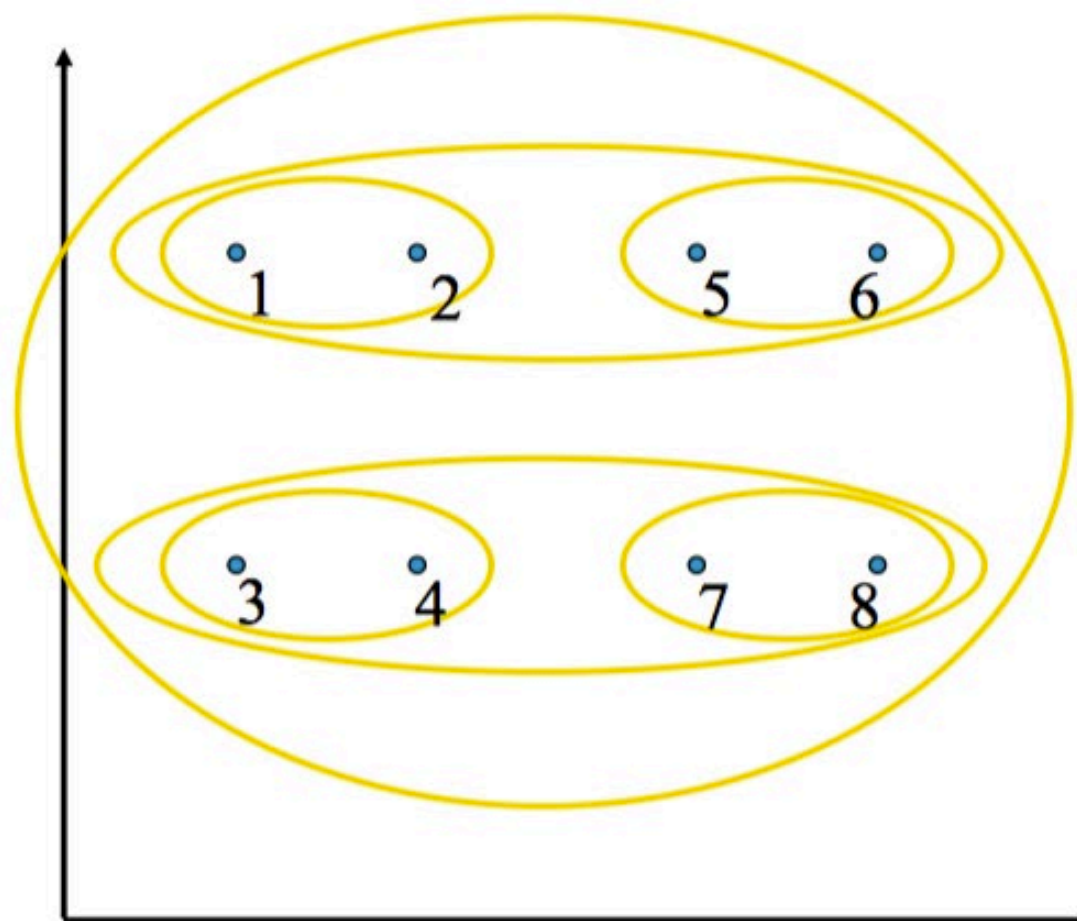
Single Link clustering (Closest pair)



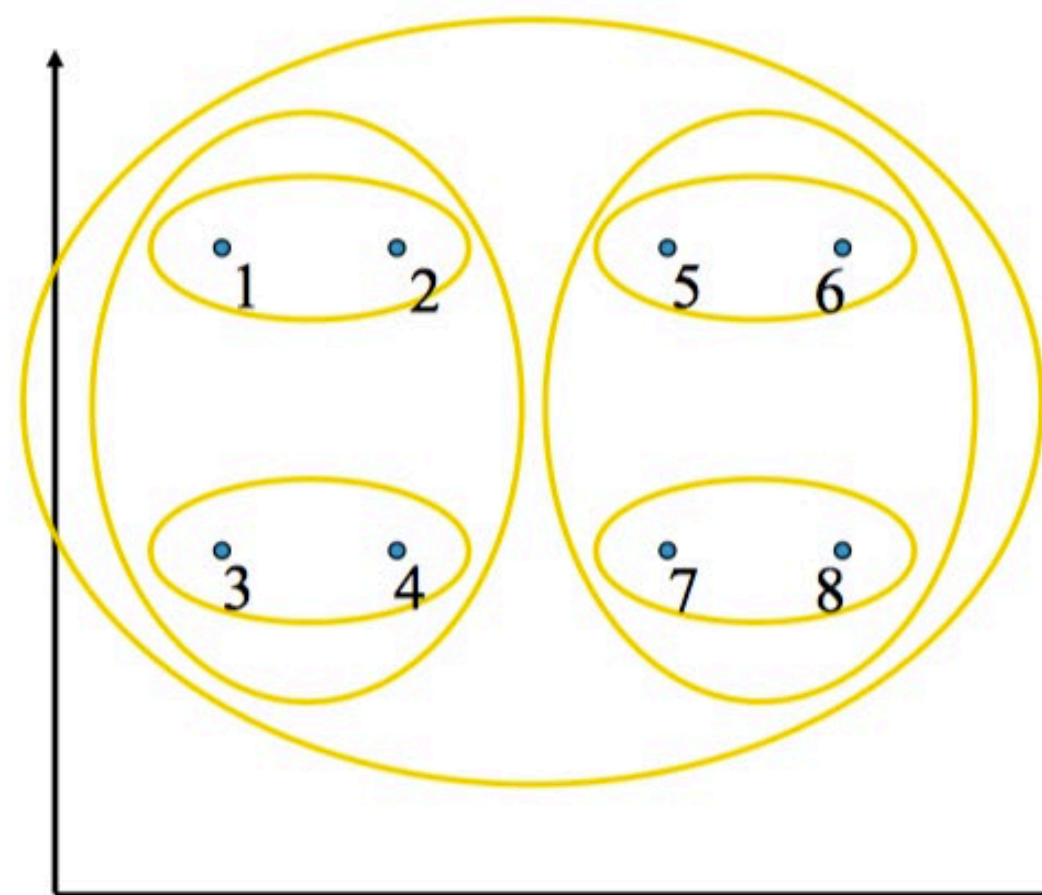
Complete Link clustering (Farthest pair)



## Closest pair (single-link clustering)



## Farthest pair (complete-link clustering)





# Clustering Evaluation

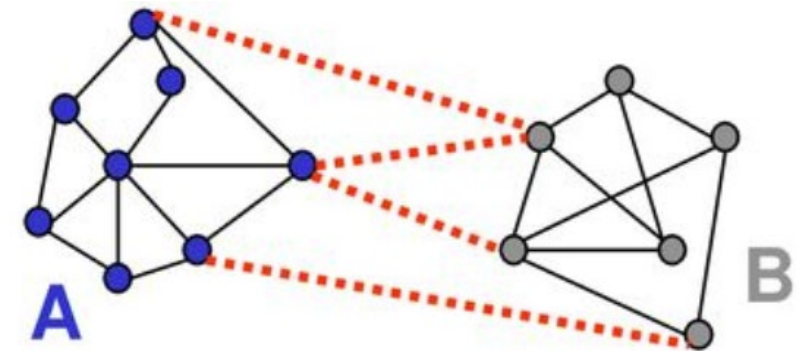
- Internal measures for clustering evaluation
  - Elbow method
  - Silhouette Coefficient
  - Graph-based measures (Beta-CV and Normalized cut)

We want intra-cluster datapoints to be as close as possible to each other and inter-clusters to be as far as possible from each other

# The Beta-CV Measure

- Let  $W$  be the pair-wise distance matrix for all the given points. For any two point sets  $S$  and  $R$ , we define:

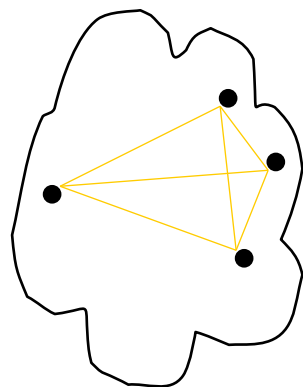
$$W(S, R) = \sum_{\mathbf{x}_i \in S} \sum_{\mathbf{x}_j \in R} w_{ij}$$



The sum of all the intracluster and intercluster weights are given as

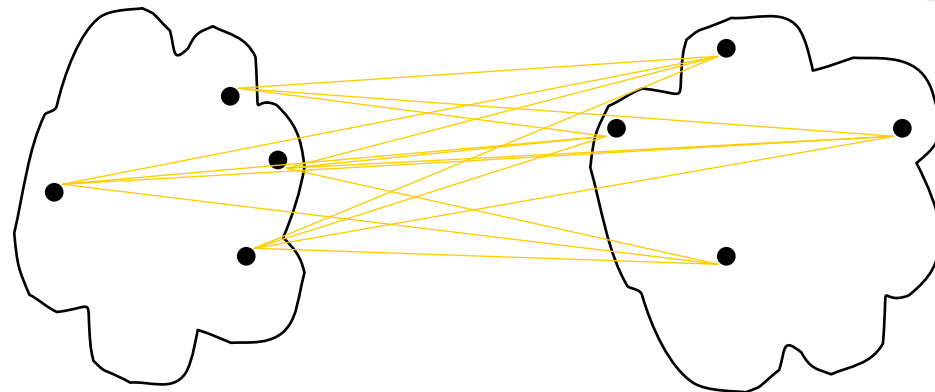
$$W_{in} = \frac{1}{2} \sum_{i=1}^k W(C_i, C_i)$$

The distance of each point is measured two times



cohesion

$$W_{out} = \frac{1}{2} \sum_{i=1}^k W(C_i, \bar{C}_i) = \sum_{i=1}^{k-1} \sum_{j>i} W(C_i, C_j)$$



separation

# The Beta-CV Measure

The number of distinct intracluster and intercluster edges is given as:

$$N_{in} = \sum_{i=1}^k \binom{n_i}{2}$$

$$N_{out} = \sum_{i=1}^{k-1} \sum_{j=i+1}^k n_i \cdot n_j$$

**BetaCV Measure:** The BetaCV measure is the ratio of the mean intracluster distance to the mean intercluster distance:

$$BetaCV = \frac{W_{in}/N_{in}}{W_{out}/N_{out}} = \frac{N_{out}}{N_{in}} \cdot \frac{W_{in}}{W_{out}} = \frac{N_{out}}{N_{in}} \frac{\sum_{i=1}^k W(C_i, C_i)}{\sum_{i=1}^k W(C_i, \overline{C_i})}$$

The smaller the BetaCV ratio, the better the clustering.

# Normalized Cut

**Normalized cut:** 
$$NC = \sum_{i=1}^k \frac{W(C_i, \bar{C}_i)}{\text{vol}(C_i)} = \sum_{i=1}^k \frac{W(C_i, \bar{C}_i)}{W(C_i, V)} = \sum_{i=1}^k \frac{W(C_i, \bar{C}_i)}{W(C_i, C_i) + W(C_i, \bar{C}_i)} = \sum_{i=1}^k \frac{1}{\frac{W(C_i, C_i)}{W(C_i, \bar{C}_i)} + 1}$$

where  $\text{vol}(C_i) = W(C_i, V)$  is the volume of cluster  $C_i$

The higher normalized cut value, the better the clustering

$W(C_i, C_i)$



$W(C_i, \bar{C}_i)$



Intra-cluster distance

Inter-cluster distance