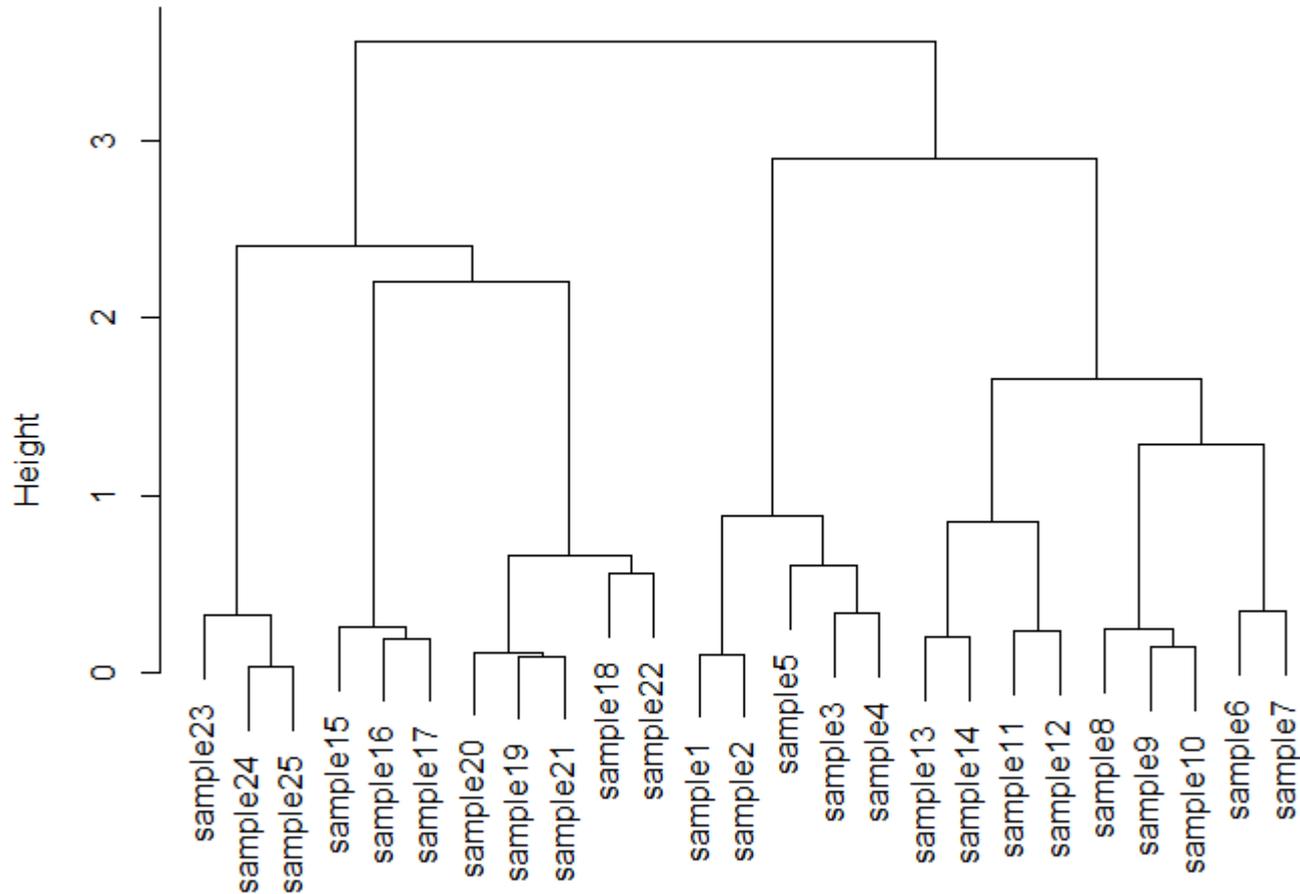


Cluster Analysis

S

{Copyright note: this lecture is partly based on lecture materials of T. Olszewski, Texas A & M, and Michal Kowalewski, U. Florida)

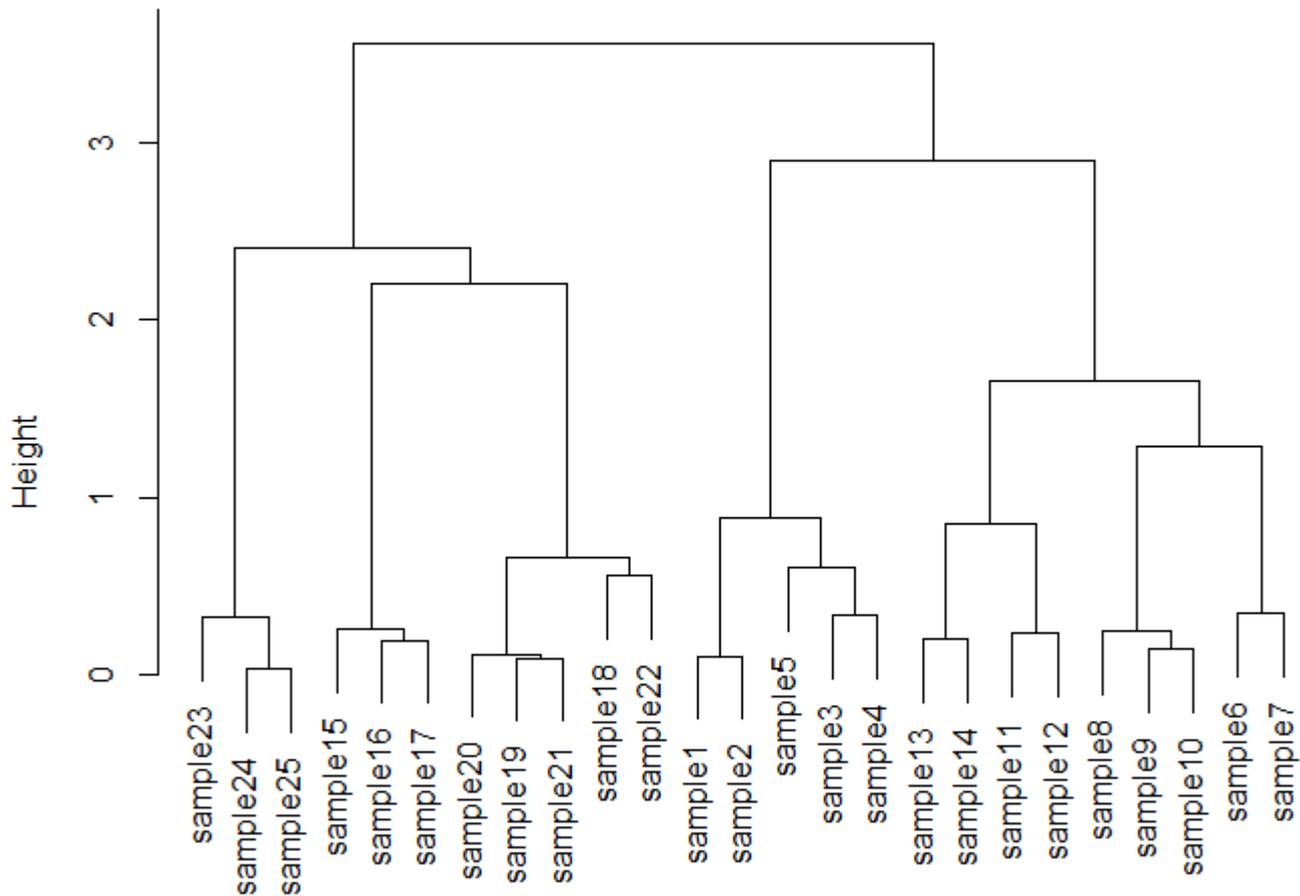
An explorative technique for identifying groups and subgroups in a multivariate dataset, based on a given distance or similarity measure. – Hammer and Harper Paleontological Data Analysis



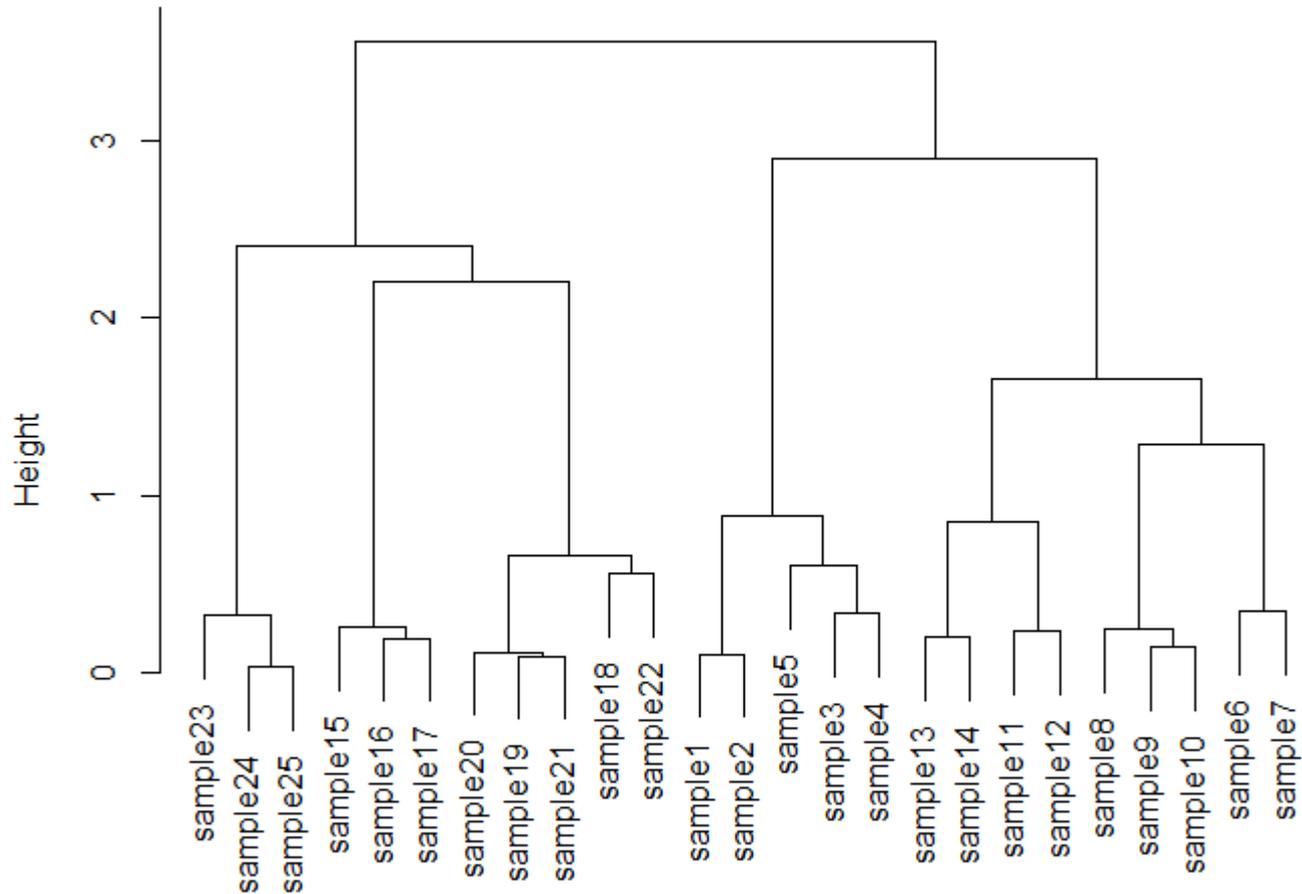
Cluster Analysis

{Copyright note: this lecture is partly based on lecture materials of T. Olszewski, Texas A & M}

Purpose: Classify observations or variables into interpretable categories (i.e., *clusters*)



Dendrogram: A branching diagram that hierarchically nests objects into increasingly more inclusive groups; degree of similarity is depicted by length of branch; ordering axis prevents branches from crossing but is otherwise arbitrary



How to convert distance matrix to clusters?

Types of clustering

- **Divisive versus Agglomerative**
- **Hierarchical** (connectivity based) **versus Non-Hierarchical** (centroid-, density-, other-based)

The classic approach is agglomerative and hierarchical

Hierarchical clustering:

- Divisive (or partitioning) Clustering: **Top-down**
- Agglomerative Clustering: **Bottom-up**
 - Most commonly used
 - Iterative
 - **Not a statistical test!** (though one could be applied... e.g. Similarity Profile Analysis (SIMPROF))

Agglomerative Clustering

Steps:

1) SEARCH. Start with a similarity matrix (ALL agglomerative clustering methods start at this point); find the cell with the highest similarity value (or lowest dissimilarity value) and link that pair of objects (i.e., form a cluster); note that if there is more than one pair with equal similarity, link the first pair found (IMPORTANT: this means that the order of objects in the matrix can influence the outcome, particularly for large data sets with lots of duplicate values!).

Agglomerative Clustering

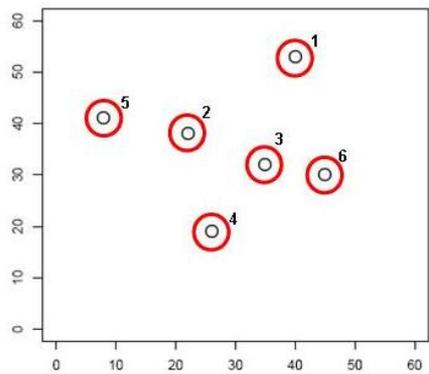
Steps:

- 1) SEARCH. Start with a similarity matrix (ALL agglomerative clustering methods start at this point); find the cell with the highest similarity value (or lowest dissimilarity value) and link that pair of objects (i.e., form a cluster); note that if there is more than one pair with equal similarity, link the first pair found (IMPORTANT: this means that the order of objects in the matrix can influence the outcome, particularly for large data sets with lots of duplicate values!).
- 2) REDUCE. Recalculate similarities, treating the clusters as new objects; how clusters are treated differs among different algorithms.

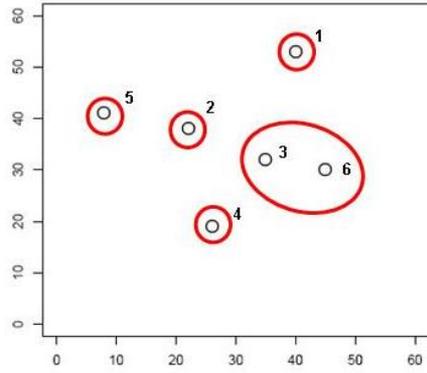
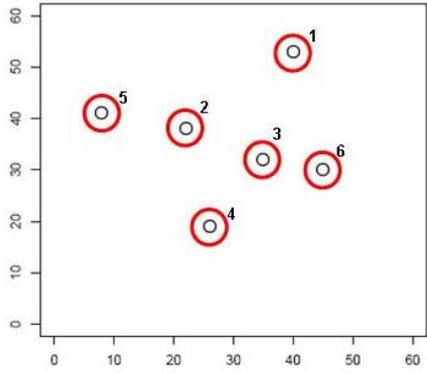
Agglomerative Clustering

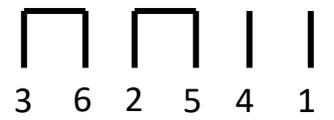
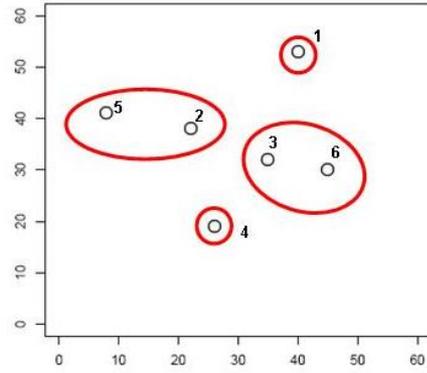
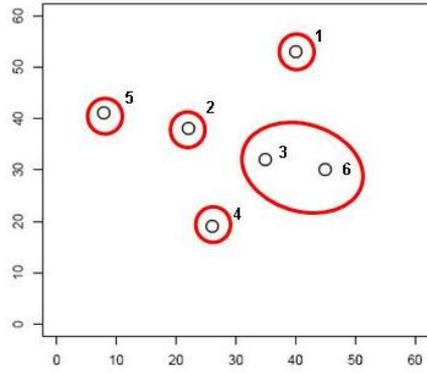
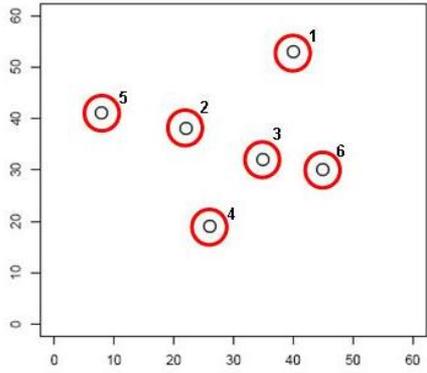
Steps:

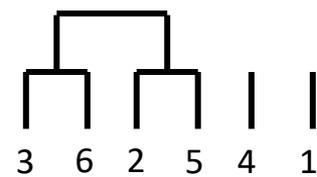
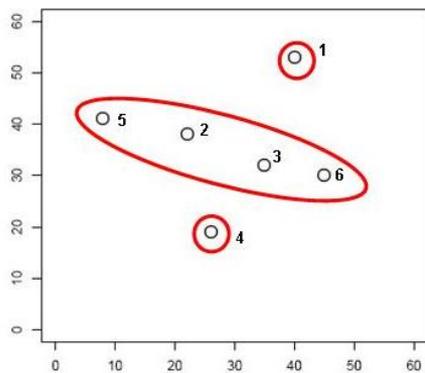
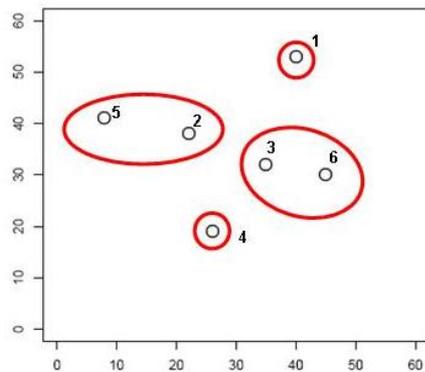
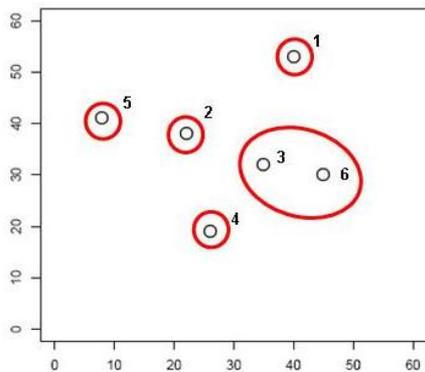
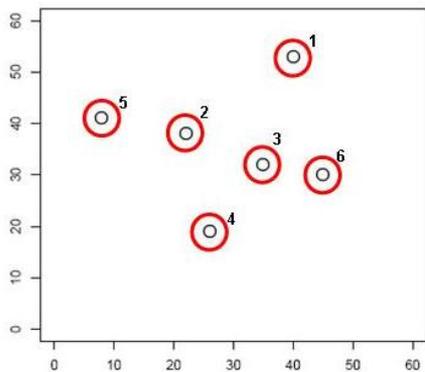
- 1) SEARCH. Start with a similarity matrix (ALL agglomerative clustering methods start at this point); find the cell with the highest similarity value (or lowest dissimilarity value) and link that pair of objects (i.e., form a cluster); note that if there is more than one pair with equal similarity, link the first pair found (IMPORTANT: this means that the order of objects in the matrix can influence the outcome, particularly for large data sets with lots of duplicate values!).
- 2) REDUCE. Recalculate similarities, treating the clusters as new objects; how clusters are treated differs among different algorithms.
- 3) REPEAT until all objects are related to one another.

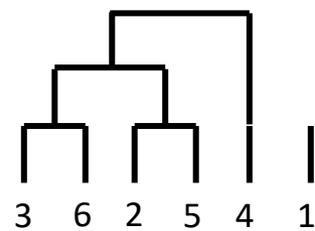
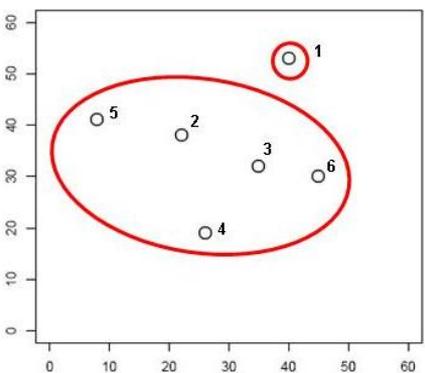
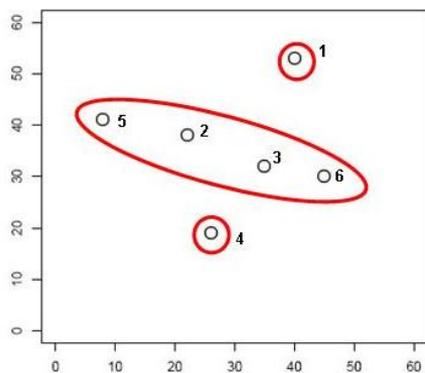
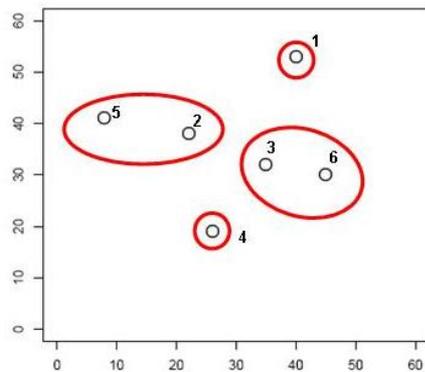
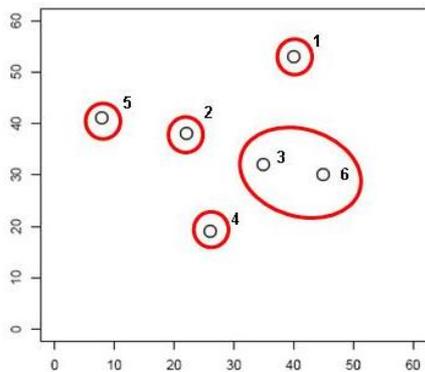
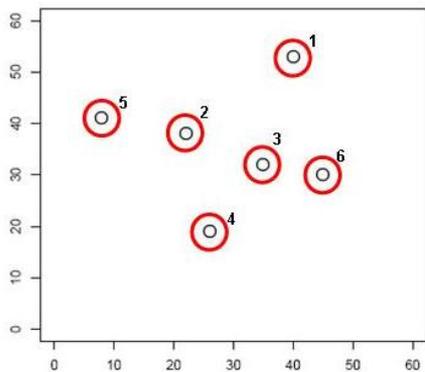


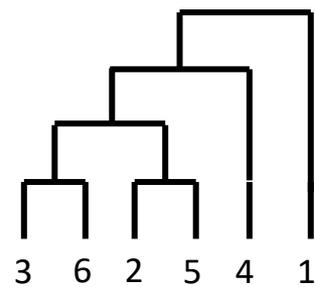
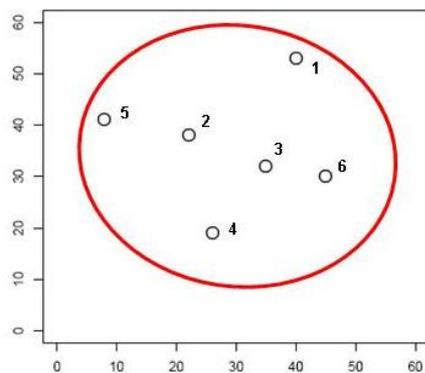
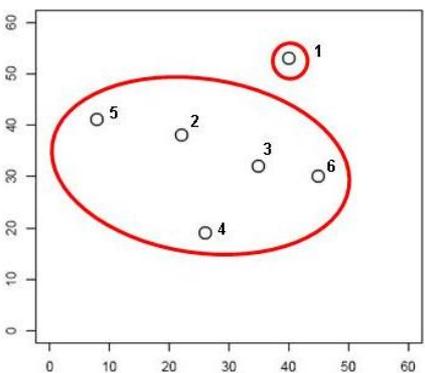
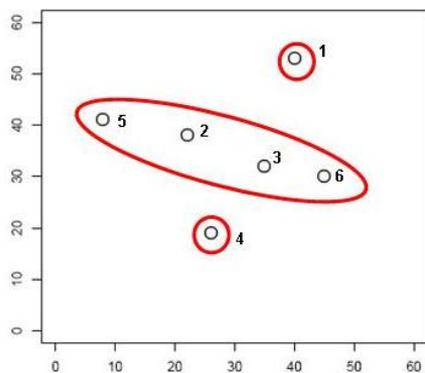
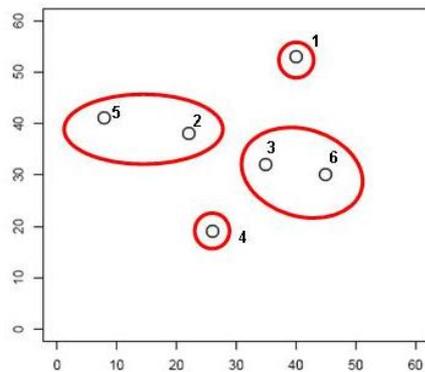
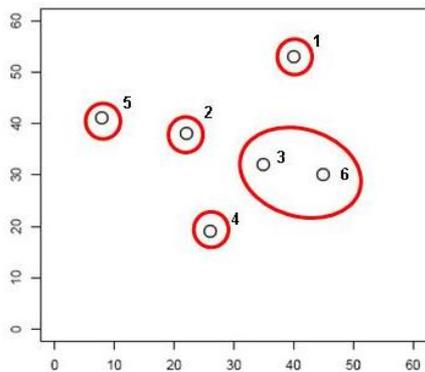
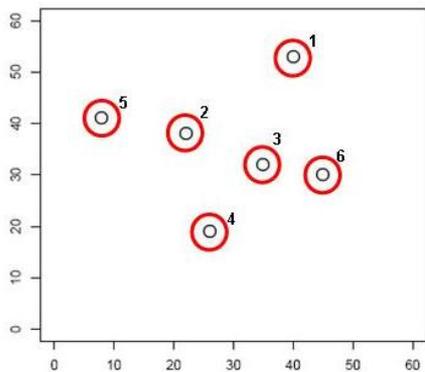
3 6 2 5 4 1

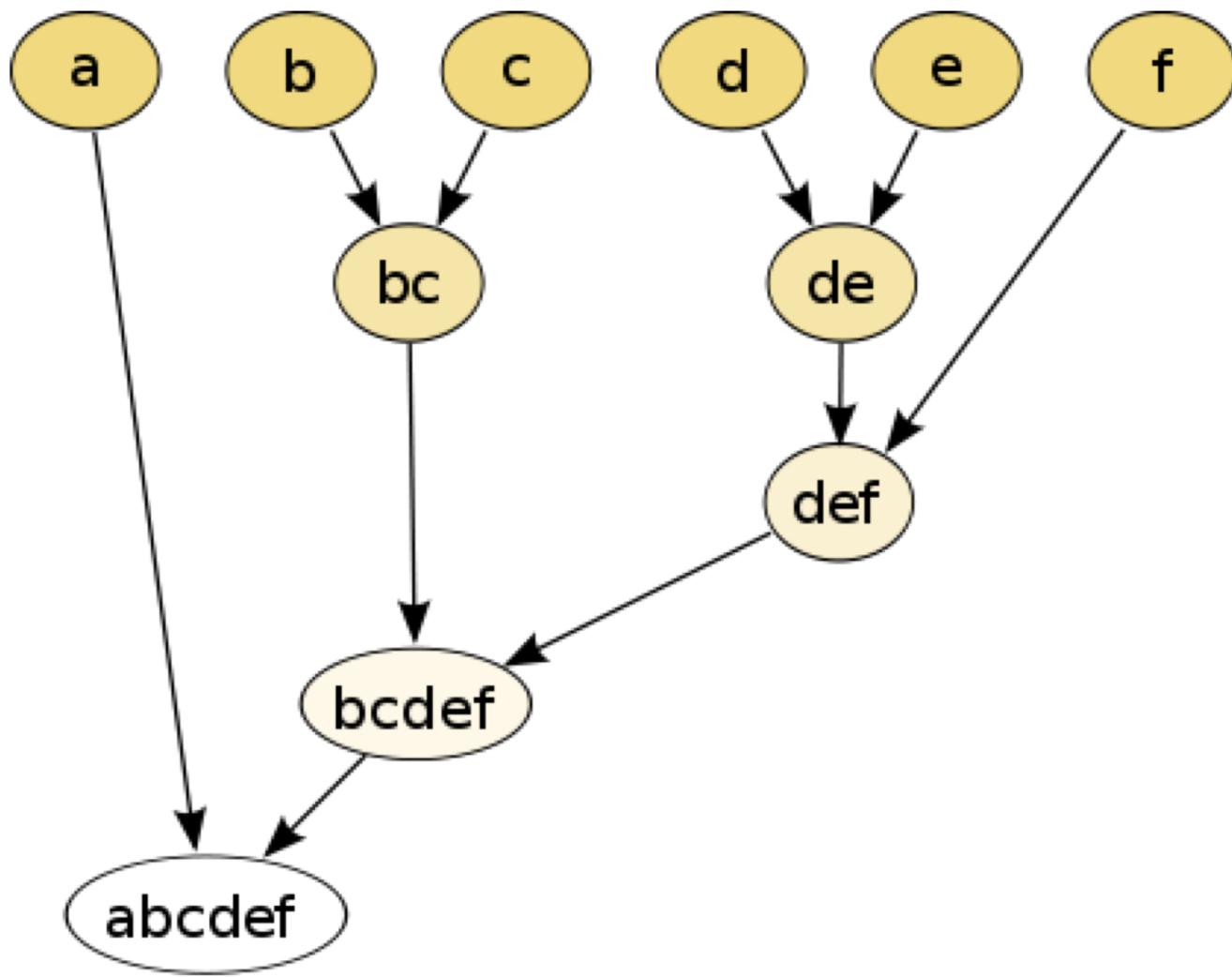












Agglomerative Linking Algorithms (How do we determine groups?)

Agglomerative Linking Algorithms (How do we determine groups?)

- With singleton clusters, linkage straightforward
 - Greatest similarity or Least distance

Agglomerative Linking Algorithms (How do we determine groups?)

- With singleton clusters, linkage straightforward
 - Greatest similarity or Least distance
- For multi-observation clusters: Linkage algorithm used will often *significantly* affect the resulting clusters

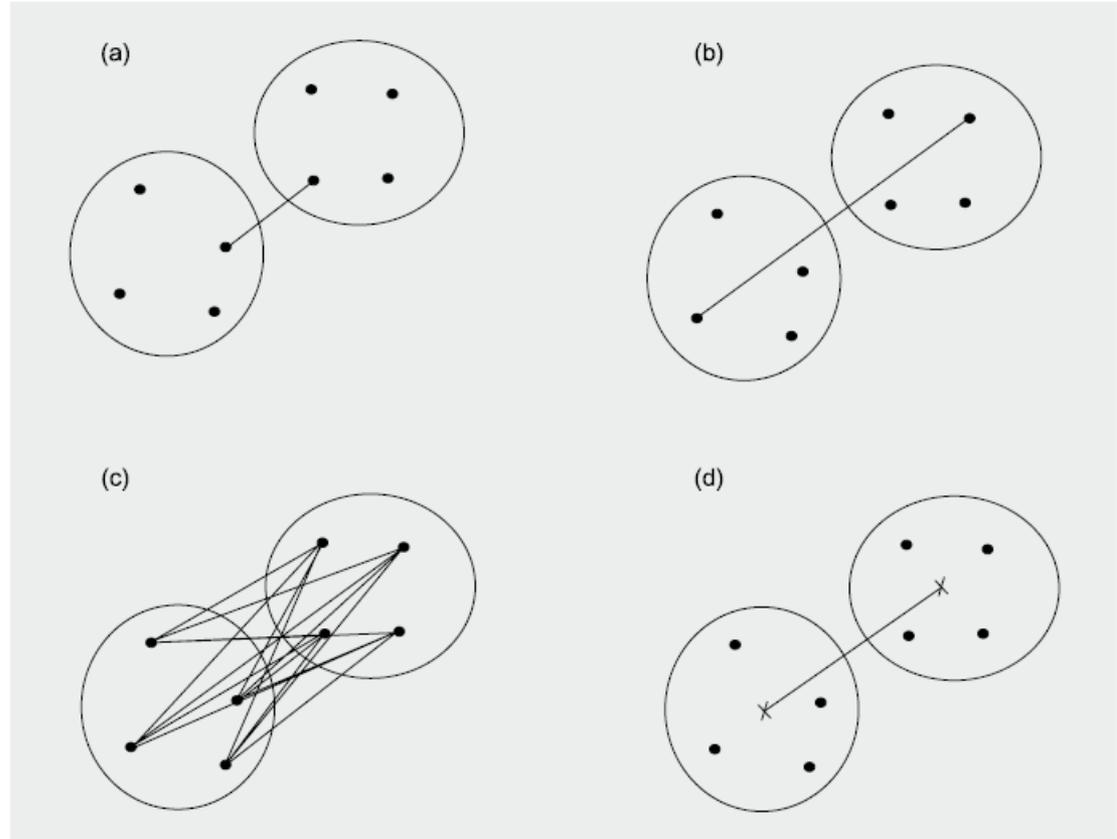
Agglomerative Linking Algorithms (How do we determine groups?)

- With singleton clusters, linkage straightforward
 - Greatest similarity or Least distance
- For multi-observation clusters: Linkage algorithm used will often *significantly* effect the resulting clusters
- No consensus on the “best” algorithm

Agglomerative Linking Algorithms (How do we determine groups?)

Types of Algorithms

- Nearest Neighbor
- Farthest Neighbor
- Average Linkage
 - Unweighted or weighted
- Centroid Linkage
 - Unweighted or weighted
- Ward's



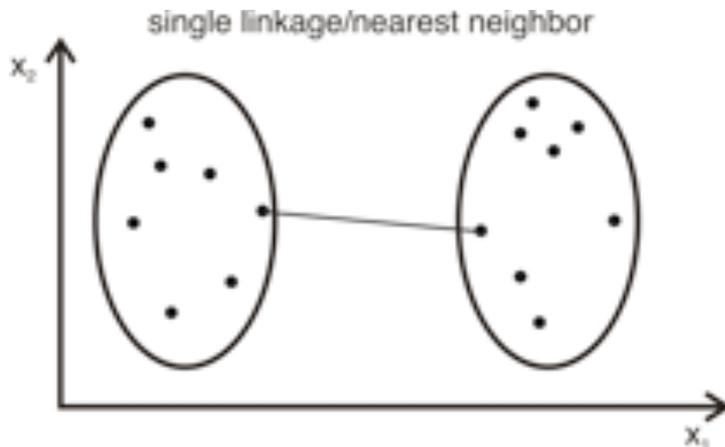
Agglomerative Linking Algorithms

Nearest Neighbor or Single Linkage Clustering. Similarity between two clusters equals the maximum similarity (minimum dissimilarity) between any two members of the clusters:

$$S(AB),C = \max(SAC, SBC)$$

$$S(AB),(CD) = \max(SAC, SAD, SBC, SBD)$$

Note: This algorithm tends to produce “chaining” (i.e., apparent addition of each object in a dendrogram, one by one); normally, this would suggest a gradient structure in the data, but single linkage clustering can produce it artifactually.



Agglomerative Linking Algorithms

Nearest Neighbor or Single Linkage Clustering. Similarity between two clusters equals the maximum similarity (minimum dissimilarity) between any two members of the clusters:

$$S(AB),C = \max(SAC, SBC)$$

$$S(AB),(CD) = \max(SAC, SAD, SBC, SBD)$$

Note: This algorithm tends to produce “chaining” (i.e., apparent addition of each object in a dendrogram, one by one); normally, this would suggest a gradient structure in the data, but single linkage clustering can produce it artifactually.

Single linkage clustering:

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

(D,F) and E distance = **1.00**

because:

(D, E) = 1.00

(F, E) = 1.12

*Note: This is a distance (not similarity) matrix. We want to minimize distance, maximize similarity

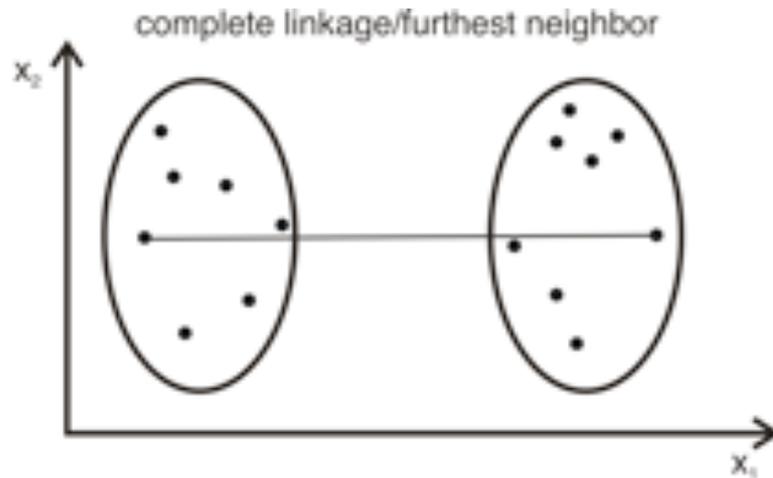
Agglomerative Linking Algorithms

Farthest Neighbor or Complete Linkage Clustering. Similarity between two clusters equals the minimum similarity (maximum dissimilarity) between any two members of the clusters:

$$S(AB),C = \min(SAC, SBC)$$

$$S(AB),(CD) = \min(SAC, SAD, SBC, SBD)$$

Note: Use this rule to recalculate similarity values in the matrix (in the REDUCING step), but continue to SEARCH for the greatest similarity values. This algorithm tends to produce very clear groups – i.e., by using minimum values, it tends to underestimate similarity between recognized clusters.



Agglomerative Linking Algorithms

Farthest Neighbor or Complete Linkage Clustering. Similarity between two clusters equals the minimum similarity (maximum dissimilarity) between any two members of the clusters:

$$S(AB),C = \min(SAC, SBC)$$

$$S(AB),(CD) = \min(SAC, SAD, SBC, SBD)$$

Note: Use this rule to recalculate similarity values in the matrix (in the REDUCING step), but continue to SEARCH for the greatest similarity values. This algorithm tends to produce very clear groups – i.e., by using minimum values, it tends to underestimate similarity between recognized clusters.

Complete linkage clustering:

(D,F) and E distance = **1.12**

because:

(D, E) = 1.00

(F, E) = 1.12

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

*Note: This is a distance (not similarity) matrix. We want to minimize distance, maximize similarity

Agglomerative Linking Algorithms

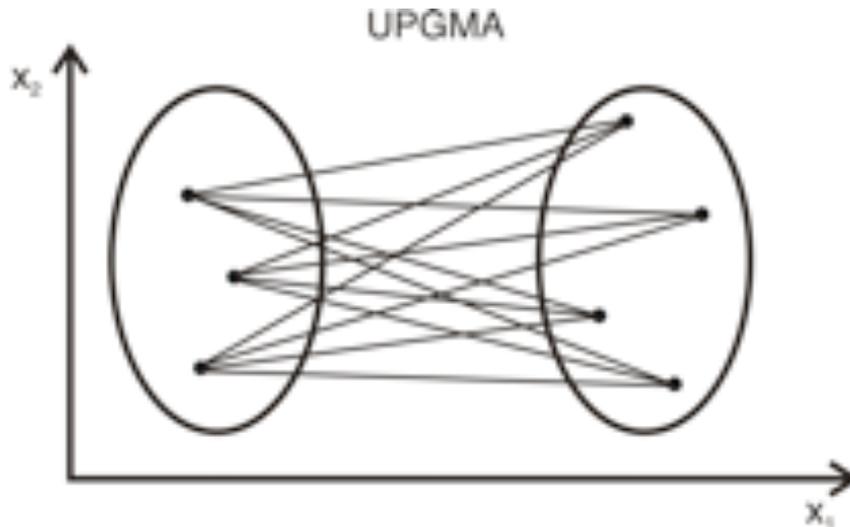
Unweighted Pair Group Method with Arithmetic Averaging (UPGMA). Similarity between two clusters equals the mean similarity between all possible pair-group combinations:

$$S(AB),C = (SAC + SBC)/2$$

$$S(AB),(CD) = (SAC + SAD + SBC + SBD)/4$$

$$SE,(C,(AB)) = (SAE + SBE + SCE)/3$$

Note: Degree of clustering is intermediate between single and complete linkage. Clusters with more samples exert greater influence on the similarity of the new cluster with all other objects.



Agglomerative Linking Algorithms

Unweighted Pair Group Method with Arithmetic Averaging (UPGMA). Similarity between two clusters equals the mean similarity between all possible pair-group combinations:

$$S(AB),C = (SAC + SBC)/2$$

$$S(AB),(CD) = (SAC + SAD + SBC + SBD)/4$$

$$SE,(C,(AB)) = (SAE + SBE + SCE)/3$$

Note: Degree of clustering is intermediate between single and complete linkage. Clusters with more samples exert greater influence on the similarity of the new cluster with all other objects.

UPGMA clustering:

((D,F), E) and C = 2.05

because:

$$(D, C) = 2.24$$

$$(F, C) = 2.50$$

$$(E, C) = 1.41$$

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

*Note: This is a distance (not similarity) matrix. We want to minimize distance, maximize similarity

Agglomerative Linking Algorithms

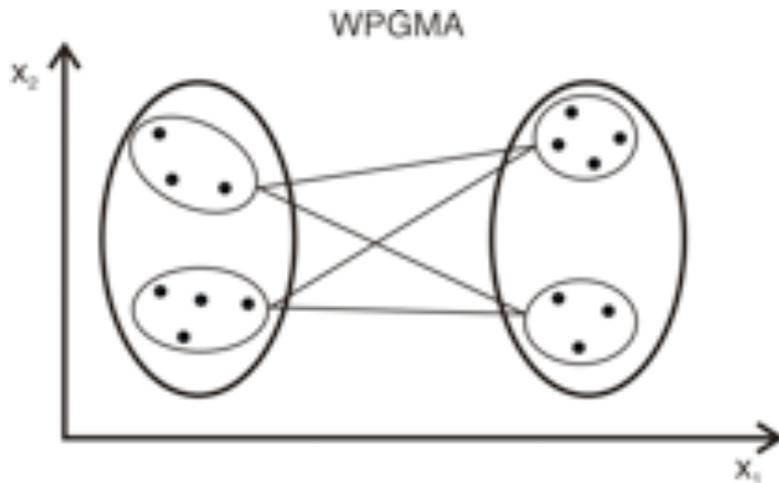
Weighted Pair Group Method with Arithmetic Averaging (WPGMA). Similarity between two clusters equals the mean similarity of previously existing clusters when they are grouped (average always involves only 2 terms and does not weight clusters by their size; i.e., when linking a cluster containing 20 samples and another containing 2 samples, the 2-sample cluster is treated as equal to the 20-sample cluster):

$$S(AB),C = (SAC + SBC)/2$$

$$S(AB),(CD) = [\frac{1}{2}(SAC + SAD) + \frac{1}{2}(SBC + SBD)]/2 = [SA,(CD) + SB,(CD)]/2$$

$$SE,(C,(AB)) = [\frac{1}{2}(SAE + SBE) + SCE]/2 = (SE,(AB) + SCE)/2$$

Note: The first two cases are identical to UPGMA, but the third effectively downweights the members of the larger cluster (AB) so that each cluster carries the same influence on the mean similarity regardless of the number of samples it includes.



Agglomerative Linking Algorithms

Weighted Pair Group Method with Arithmetic Averaging (WPGMA). Similarity between two clusters equals the mean similarity of previously existing clusters when they are grouped (average always involves only 2 terms and does not weight clusters by their size; i.e., when linking a cluster containing 20 samples and another containing 2 samples, the 2-sample cluster is treated as equal to the 20-sample cluster):

$$S(AB),C = (SAC + SBC)/2$$

$$S(AB),(CD) = [\frac{1}{2}(SAC + SAD) + \frac{1}{2}(SBC + SBD)]/2 = [SA,(CD) + SB,(CD)]/2$$

$$SE,(C,(AB)) = [\frac{1}{2}(SAE + SBE) + SCE]/2 = (SE,(AB) + SCE)/2$$

Note: The first two cases are identical to UPGMA, but the third effectively downweights the members of the larger cluster (AB) so that each cluster carries the same influence on the mean similarity regardless of the number of samples it includes.

WPGMA clustering:

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

$((D,F), E)$ and $C = \mathbf{1.89}$

because:

$((D, F), C) = 2.37$ (avg of 2.50 and 2.24)

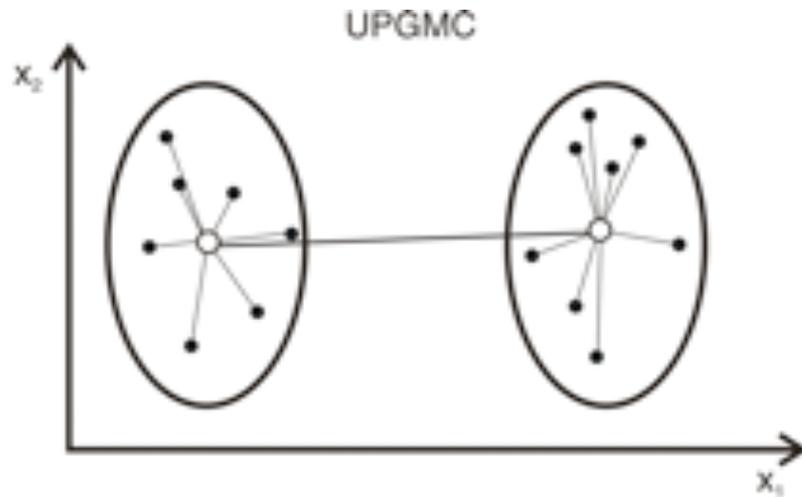
$(E, C) = 1.41$

*Note: This is a distance (not similarity) matrix. We want to minimize distance, maximize similarity

Agglomerative Linking Algorithms

Unweighted Pair Group Method with Centroid Averaging (UPGMC). Similarity between two clusters equals their similarities as composite objects (i.e., the sums of all their component samples):

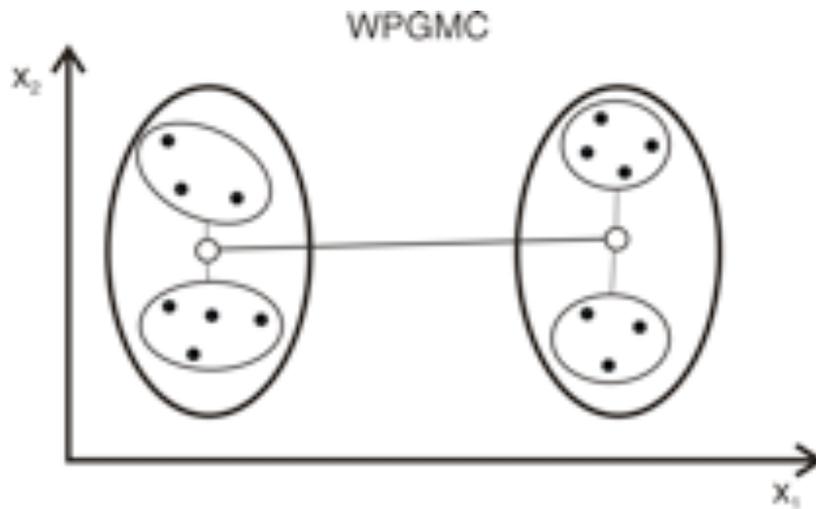
$$x_{(AB),Ci} = \frac{x_{Ai} + x_{Bi} + x_{Ci}}{3}$$



Agglomerative Linking Algorithms

Weighted Pair Group Method with Centroid Averaging (WPGMC). Similarity between two clusters equals their similarities as composite objects, but determining their composite composition using only the last two objects (samples or clusters) to be joined in each cluster:

$$x_{(AB),Ci} = \frac{\frac{(x_{Ai} + x_{Bi})}{2} + x_{Ci}}{2} = \frac{(x_{Ai} + x_{Bi})}{4} + \frac{x_{Ci}}{2}$$

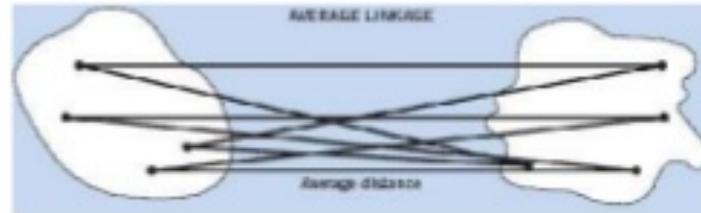


Agglomerative Linking Algorithms

Ward's Method or Minimum Variance Clustering or Orloci's Sum of Squares. Similarity is calculated as in UPGMA or UPGMC but clusters are created by grouping objects to minimize the variance (or inertia/entropy/information) of the similarities between all member objects within a cluster:

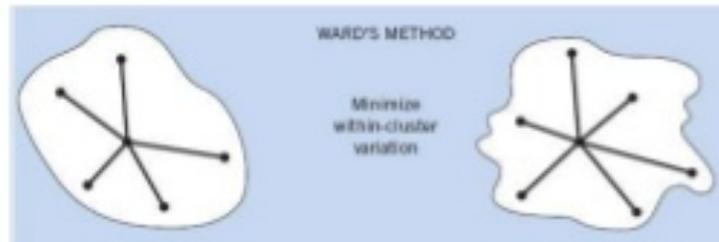
- **Average Linkage**

- Clustering criterion based on the average distance



- **Ward's Method**

- Based on the loss of information resulting from grouping of the objects into clusters (minimize within cluster variation)



Agglomerative Linking Algorithms

Ward's Method or Minimum Variance Clustering or Orloci's Sum of Squares. Similarity is calculated as in UPGMA or UPGMC but clusters are created by grouping objects to minimize the variance (or inertia/entropy/information) of the similarities between all member objects within a cluster:

3 clusters: ABC, DE, F; which two should be clustered next? Pick the two that produce the minimum variance (= mean of all S 's in that cluster):

$$V_{(ABC)(DE)} = \frac{(S_{AD} - \bar{S})^2 + (S_{AE} - \bar{S})^2 + (S_{BD} - \bar{S})^2 + (S_{BE} - \bar{S})^2 + (S_{CD} - \bar{S})^2 + (S_{CE} - \bar{S})^2}{6}$$

$$V_{(ABC)F} = \frac{(S_{AF} - \bar{S})^2 + (S_{BF} - \bar{S})^2 + (S_{CF} - \bar{S})^2}{3}$$

$$V_{(DE)F} = \frac{(S_{DF} - \bar{S})^2 + (S_{EF} - \bar{S})^2}{2}$$

Agglomerative Clustering Algorithms

Limitations of Agglomerative Cluster Analysis:

Agglomerative Clustering Algorithms

Limitations of Agglomerative Cluster Analysis:

- Imposes hierarchical structure on data, whether real or not
 - Even a completely uniform similarity matrix will produce clusters

Agglomerative Clustering Algorithms

Limitations of Agglomerative Cluster Analysis:

- Imposes hierarchical structure on data, whether real or not
 - Even a completely uniform similarity matrix will produce clusters
- It does not depict data with multiple, independent underlying controls well
 - Not good for visualizing e.g. lithology, bathymetry, oxygen levels, etc.

Agglomerative Clustering Algorithms

Limitations of Agglomerative Cluster Analysis:

- Imposes hierarchical structure on data, whether real or not
 - Even a completely uniform similarity matrix will produce clusters
- It does not depict data with multiple, independent underlying controls well
 - Not good for visualizing e.g. lithology, bathymetry, oxygen levels, etc.
- Because based on algorithms rather than analytical solutions, solutions can be non-unique

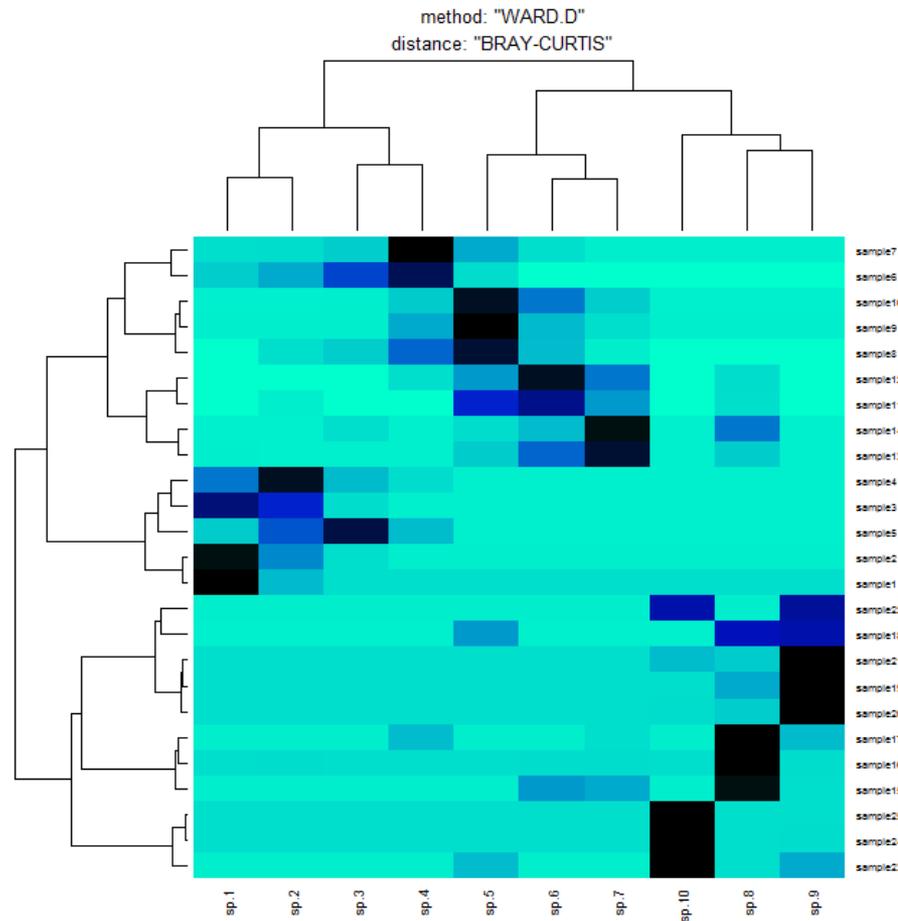
Agglomerative Clustering Algorithms

Limitations of Agglomerative Cluster Analysis:

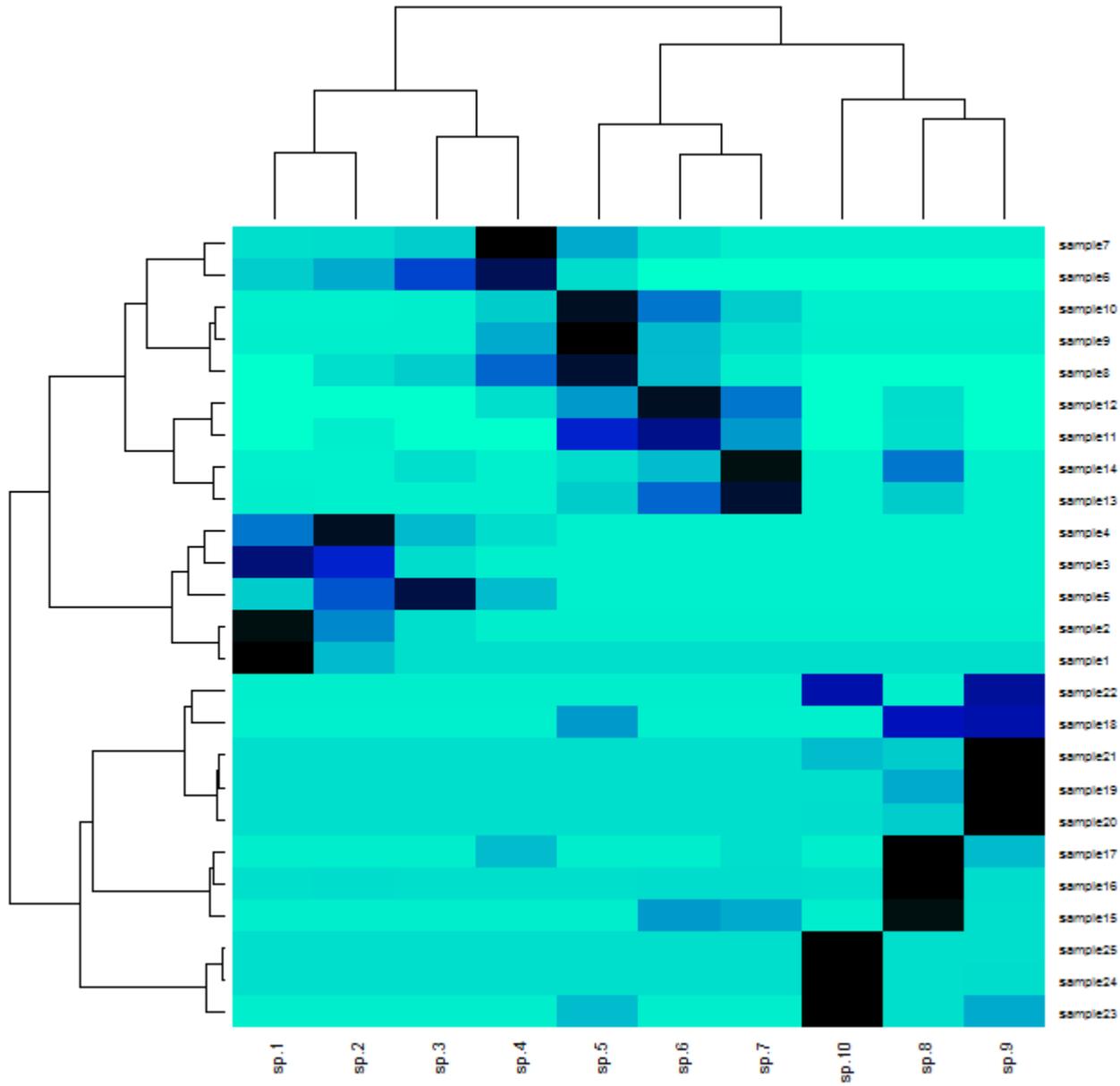
- Imposes hierarchical structure on data, whether real or not
 - Even a completely uniform similarity matrix will produce clusters
- It does not depict data with multiple, independent underlying controls well
 - Not good for visualizing e.g. lithology, bathymetry, oxygen levels, etc.
- Because based on algorithms rather than analytical solutions, solutions can be non-unique
- Linkage algorithm used significantly effects topology of clusters

Two-way Cluster Analysis

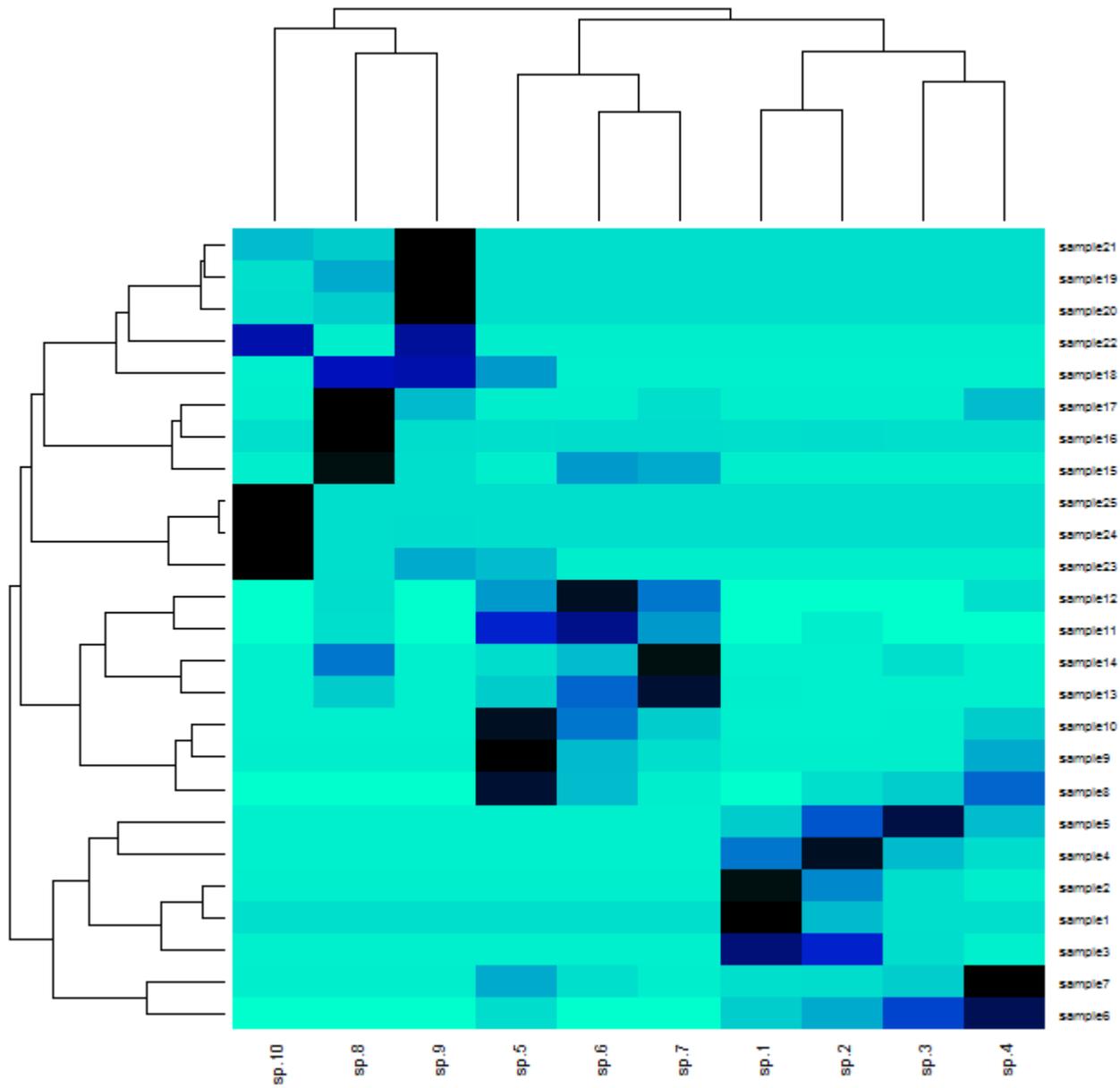
Use ordering axis of sample and taxon analyses of the same data to reorganize the rows and columns of the original data matrix – a very effective way of understanding which taxa are related to which cluster of samples. Remember that the purpose of the ordering axis in cluster analysis is to keep branches from crossing, not to show a statistically justified gradient, so links can be “reflected” and “spun”. In this sense, two-way cluster analysis is arbitrary, but it is nevertheless a very effective way of relating taxa and samples and incorporating data into



method: "WARD.D"
distance: "BRAY-CURTIS"



method: "UPGMA"
distance: "BRAY-CURTIS"



Cophenetic or Matrix Correlation

- Cluster analysis inherently discards some information from the original similarity matrix. So, **how well does a dendrogram represents the original matrix?**

Cophenetic Correlation

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

Cophenetic or Matrix Correlation

- Cluster analysis inherently discards some information from the original similarity matrix. So, **how well does a dendrogram represents the original matrix?**
- Each pair of objects has a similarity value in the original similarity matrix and a depicted similarity in the dendrogram (the linkage level of the pair). For each pair, these values can be cross-plotted, with dendrogram similarity on one axis and measured similarity on the other.

Cophenetic Correlation

Original distance matrix

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

Cophenetic (dendrogram) distance matrix

Dist	A	B	C	D	E	F
A						
B	0.71					
C	2.50	2.50				
D	2.50	2.50	1.41			
E	2.50	2.50	1.41	1.00		
F	2.50	2.50	1.41	0.50	1.00	

*Using single linkage (nearest neighbor) clustering

Cophenetic or Matrix Correlation

- Cluster analysis inherently discards some information from the original similarity matrix. So, **how well does a dendrogram represents the original matrix?**
- Each pair of objects has a similarity value in the original similarity matrix and a depicted similarity in the dendrogram (the linkage level of the pair). For each pair, these values can be cross-plotted, with dendrogram similarity on one axis and measured similarity on the other.
- The degree of correlation between the two (quantified using r) is a measure of how well the clustered pattern retained the underlying information. Note that cluster analysis generally is good at linking very similar objects but loses its ability to accurately depict patterns at lower levels of similarity

Cophenetic Correlation

Original distance matrix

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

Distance	CP
0.71	0.71
5.66	2.50
3.61	2.50
4.24	2.50
3.20	2.50
4.95	2.50
2.92	2.50
3.54	2.50
2.50	2.50
2.24	1.41
1.41	1.41
2.50	1.41
1.00	1.00
0.50	0.50
1.12	1.00

Cophenetic (dendrogram) distance matrix

Dist	A	B	C	D	E	F
A						
B	0.71					
C	2.50	2.50				
D	2.50	2.50	1.41			
E	2.50	2.50	1.41	1.00		
F	2.50	2.50	1.41	0.50	1.00	

*Using single linkage (nearest neighbor) clustering

Cophenetic Correlation

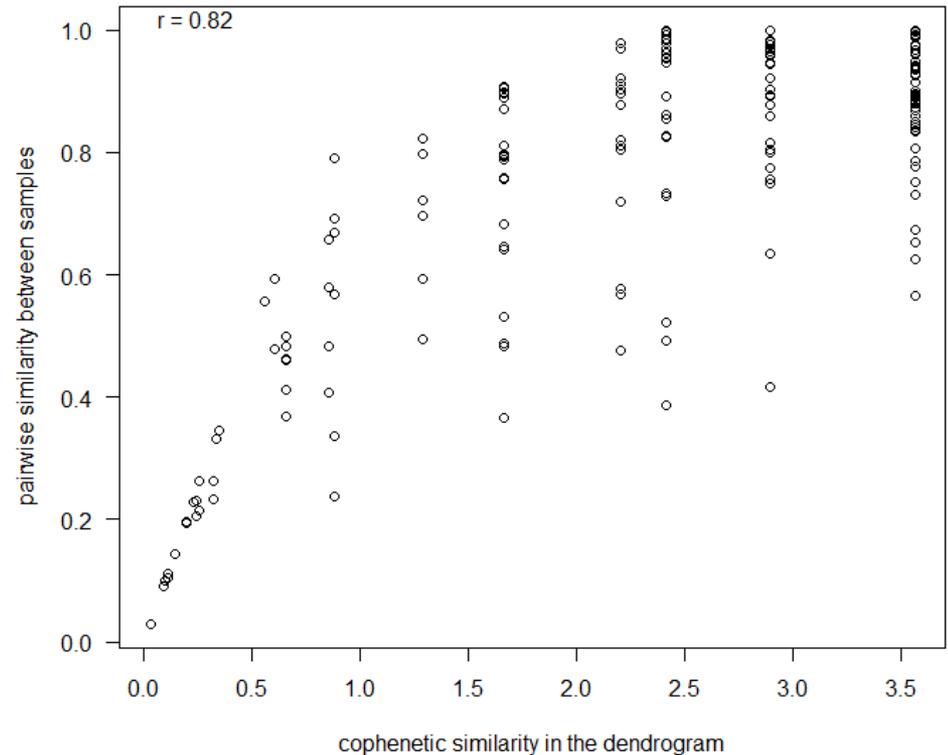
Original distance matrix

Dist	A	B	C	D	E	F
A						
B	0.71					
C	5.66	4.95				
D	3.61	2.92	2.24			
E	4.24	3.54	1.41	1.00		
F	3.20	2.50	2.50	0.50	1.12	

Cophenetic (dendrogram) distance matrix

Dist	A	B	C	D	E	F
A						
B	0.71					
C	2.50	2.50				
D	2.50	2.50	1.41			
E	2.50	2.50	1.41	1.00		
F	2.50	2.50	1.41	0.50	1.00	

Cophenetic Coefficient



*Using single linkage clustering

Non-hierarchical clustering:

One common type: partitioning algorithms

- example: k-means clustering
- User defined # of clusters
- Iterative
- Non-hierarchical

Partition Clustering

Partitioning clustering starts with all objects and divides them into groups

Aim: Given n objects (observations) in a p -dimensional space (variables), determine a partition of the objects into K groups, or clusters, such that the objects within each cluster are more similar to one another than to objects in the other clusters. Each cluster should have the smallest *inertia* or *entropy* or *variance* possible.

Note: K is determined a priori by the user, although multiple K values can be tried to find an optimal fit

Partition Clustering

Partitioning clustering starts with all objects and divides them into groups

Aim: Given n objects (observations) in a p -dimensional space (variables), determine a partition of the objects into K groups, or clusters, such that the objects within each cluster are more similar to one another than to objects in the other clusters. Each cluster should have the smallest *inertia* or *entropy* or *variance* possible.

Note: K is determined a priori by the user, although multiple K values can be tried to find an optimal fit

K-means Algorithm

1. Divide the objects into K sets, either randomly or using some external information
2. Calculate the centroid of each data set
3. Reassign all objects to the nearest centroid
4. Recalculate new centroids based on new groupings
5. Repeat until clusters stabilize (membership no longer changes and centroids don't move)

The traditional form of this algorithm minimizes the total error sum of squares (TESS):

$$E_K^2 = \sum_{i=1}^K \sum_{x_j \in S_i} (x_j - \mu_i)^2$$

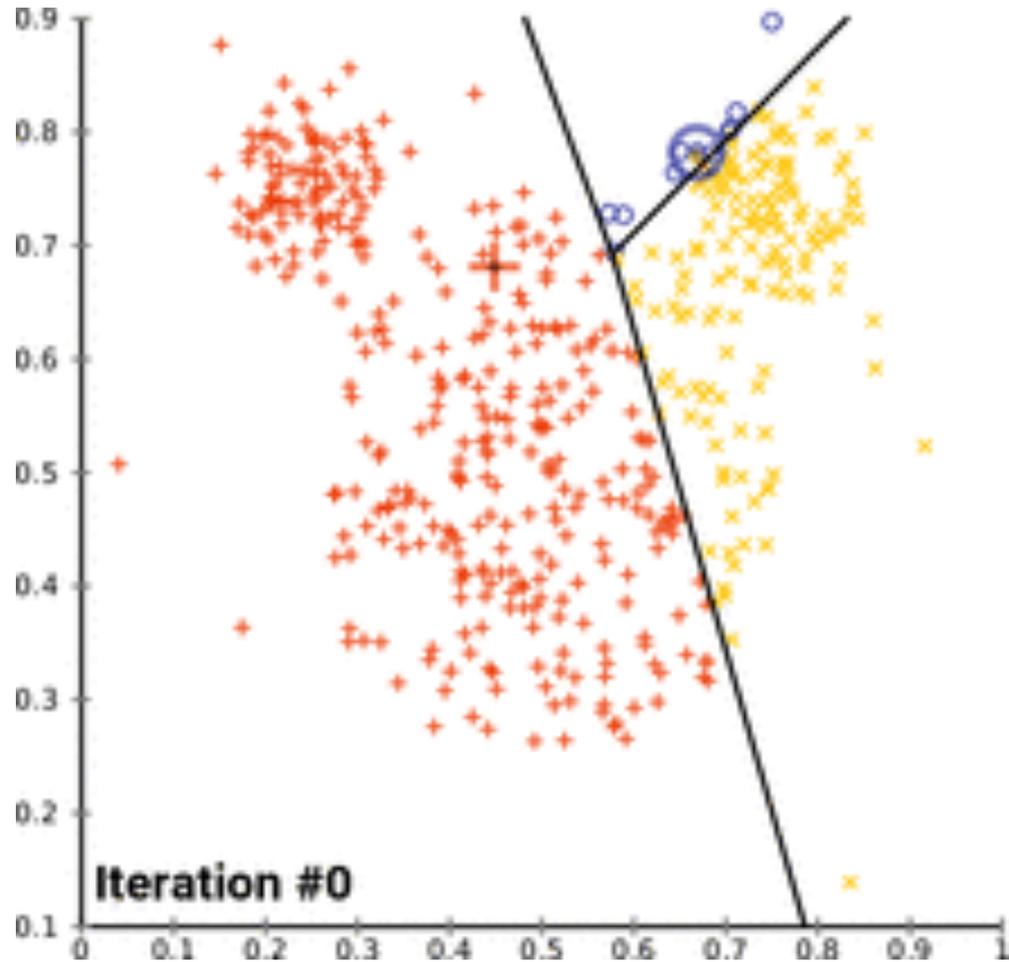
K = number of clusters

S_i = set of objects in cluster i ($i = 1 \dots K$)

x_j = value of variable j for object x

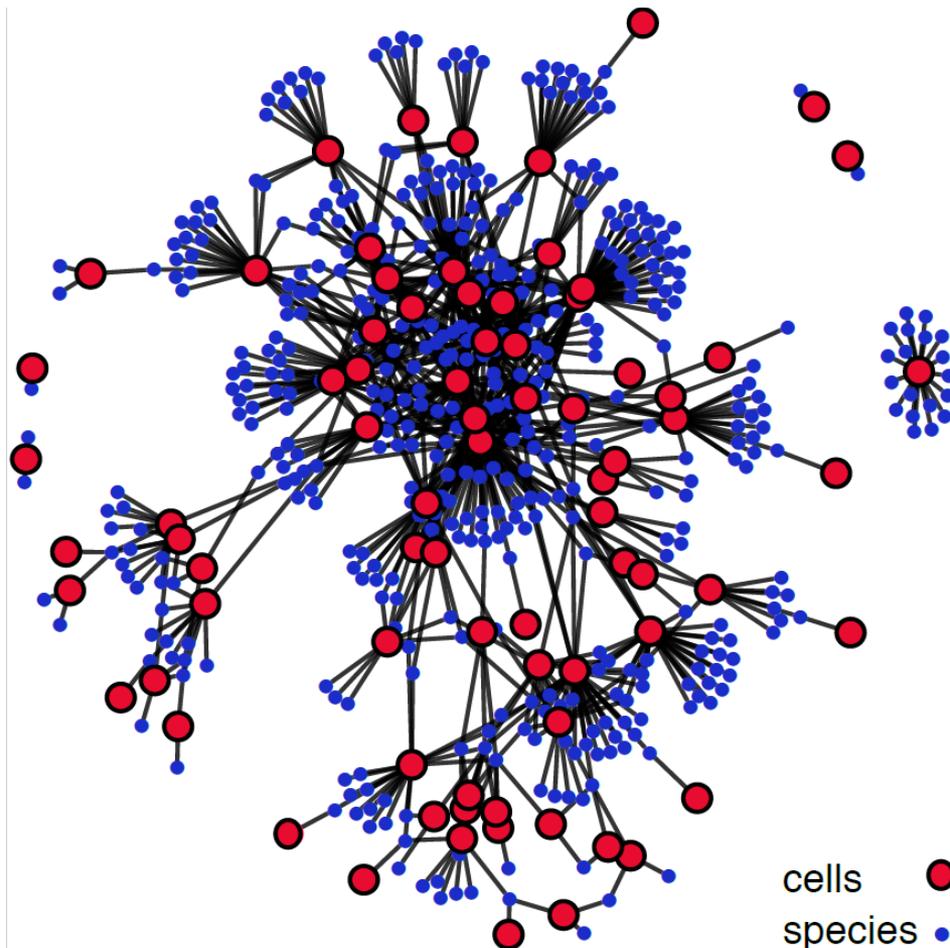
m_{ij} = mean of variable j in cluster i

K-means algorithm



$k = 3$ Goal: **minimizes the total error sum of squares**

Divisive versus Agglomerative and Hierarchical versus Non-Hierarchical Methods – The classical approach is agglomerative and hierarchical; modern clustering techniques for very large data sets (e.g., genetic networks) are based on information theory and are typically neither hierarchical (i.e., cannot be depicted as dendrograms) nor agglomerative; rather, they are based on network linkage patterns



Example of a biogeographic network

Rojas et al. 2017, *Geology*

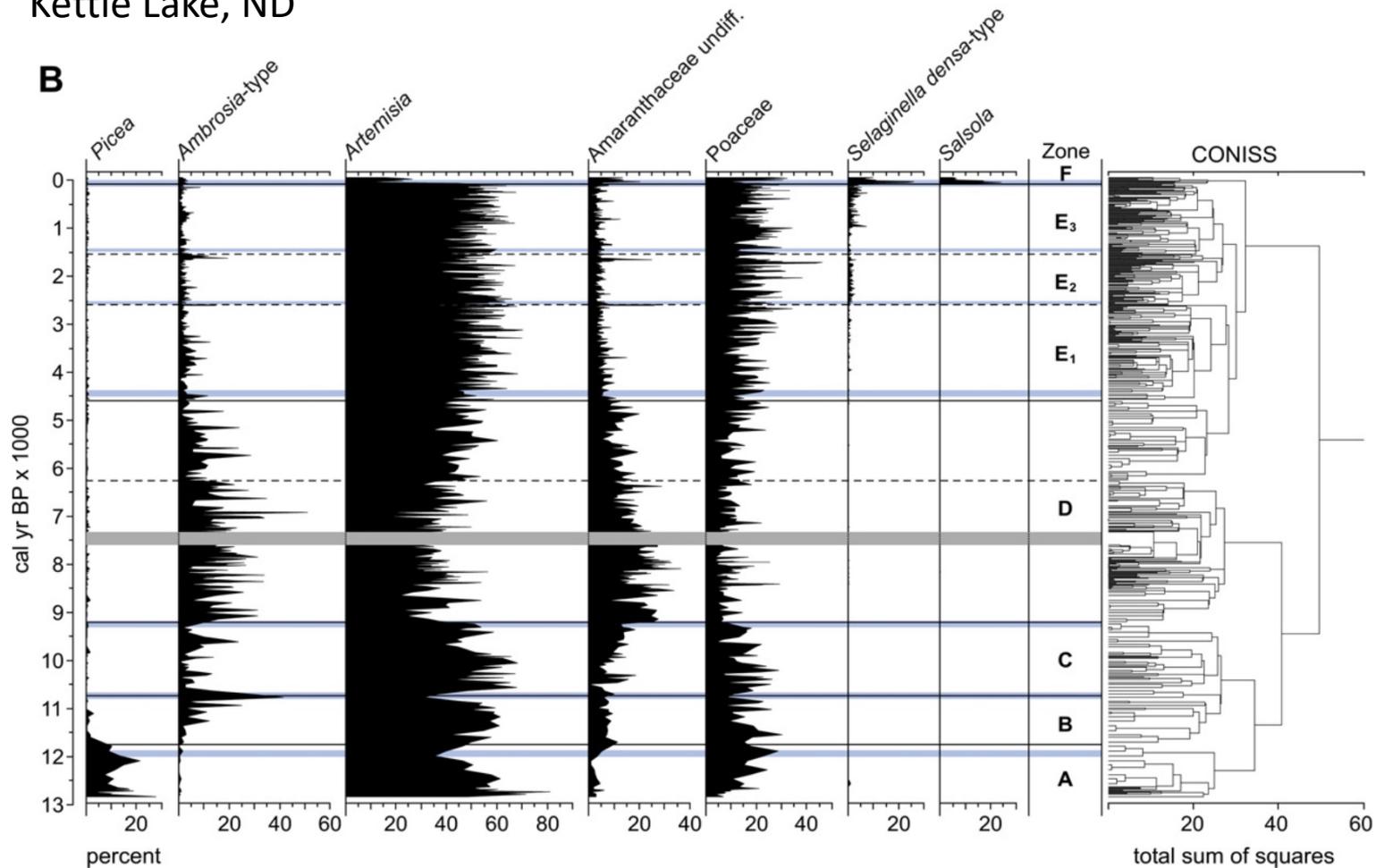
Constrained clustering

Typically, constrained clustering incorporates either a set of must-link constraints, cannot-link constraints, or both.

In the usual unconstrained analysis, the entire dissimilarity matrix is searched at each stage for the next link/cluster. In the constrained analysis, only stratigraphically adjacent values/clusters are considered.

Example, constrained clustering

Kettle Lake, ND



R Commands

`cophenetic{stats}` - computes the cophenetic distances for a hierarchical clustering.

`hclust{stats}` - hierarchical cluster analysis on a set of dissimilarities and methods for analyzing it.

`rect.hcluster{stats}` - draws rectangles around the branches of a dendrogram highlighting the corresponding clusters.

`identify.hlcust{stats}` - reads the position of the graphics pointer when the (first) mouse button is pressed. It then cuts the tree at the vertical position of the pointer and highlights the cluster containing the horizontal position of the pointer.

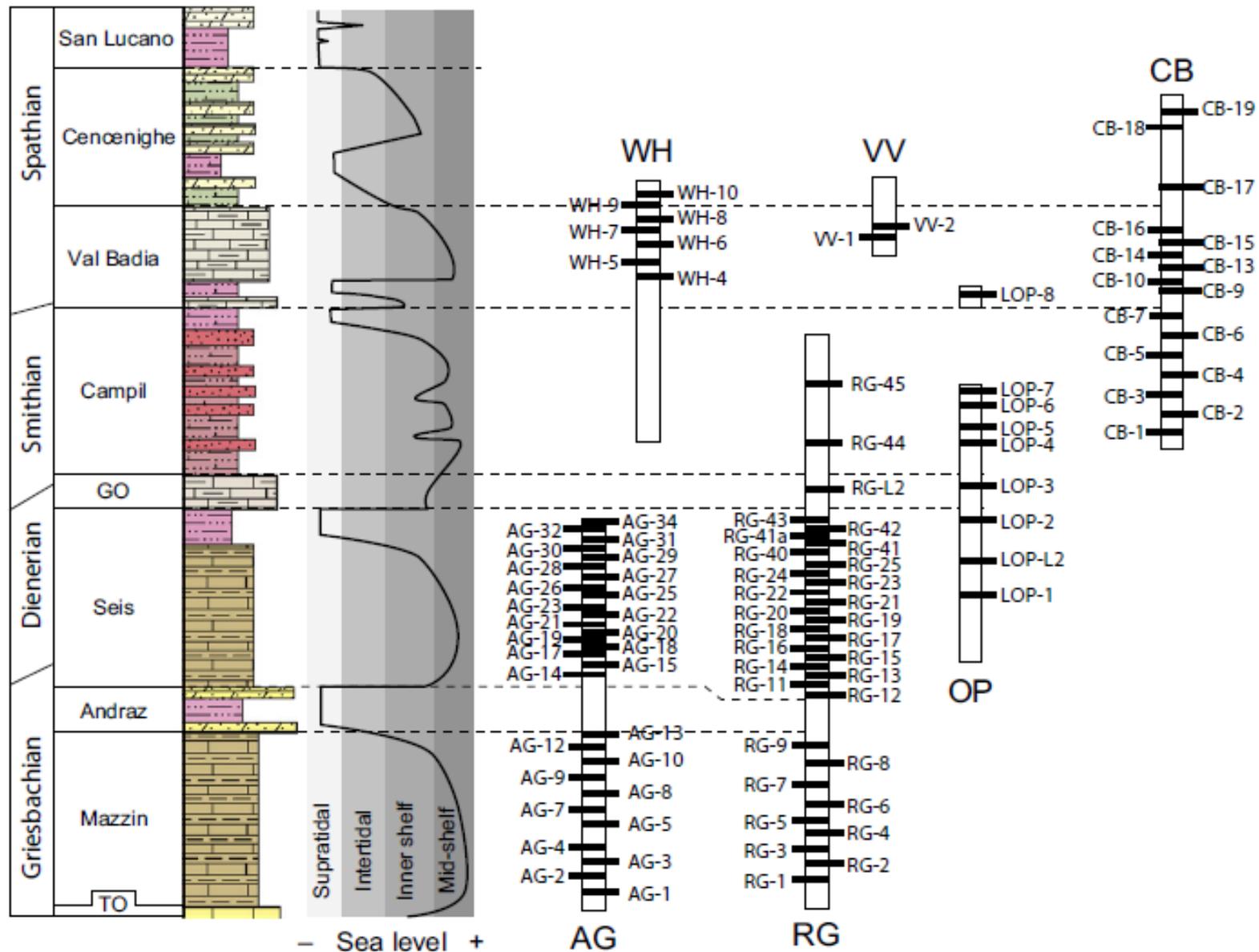
`heatmap{stats}` - produces a false color image (basically `image(t(x))`) with a dendrogram added to the left side and to the top. Reordering of the rows and columns according to some set of values (row or column means) within the restrictions imposed by the dendrogram is carried out. This command can be readily modified to accommodate any dissimilarity metric and any clustering algorithm to produce a 2-way cluster analysis.

`kmeans{stats}` - perform k-means clustering on a data matrix.

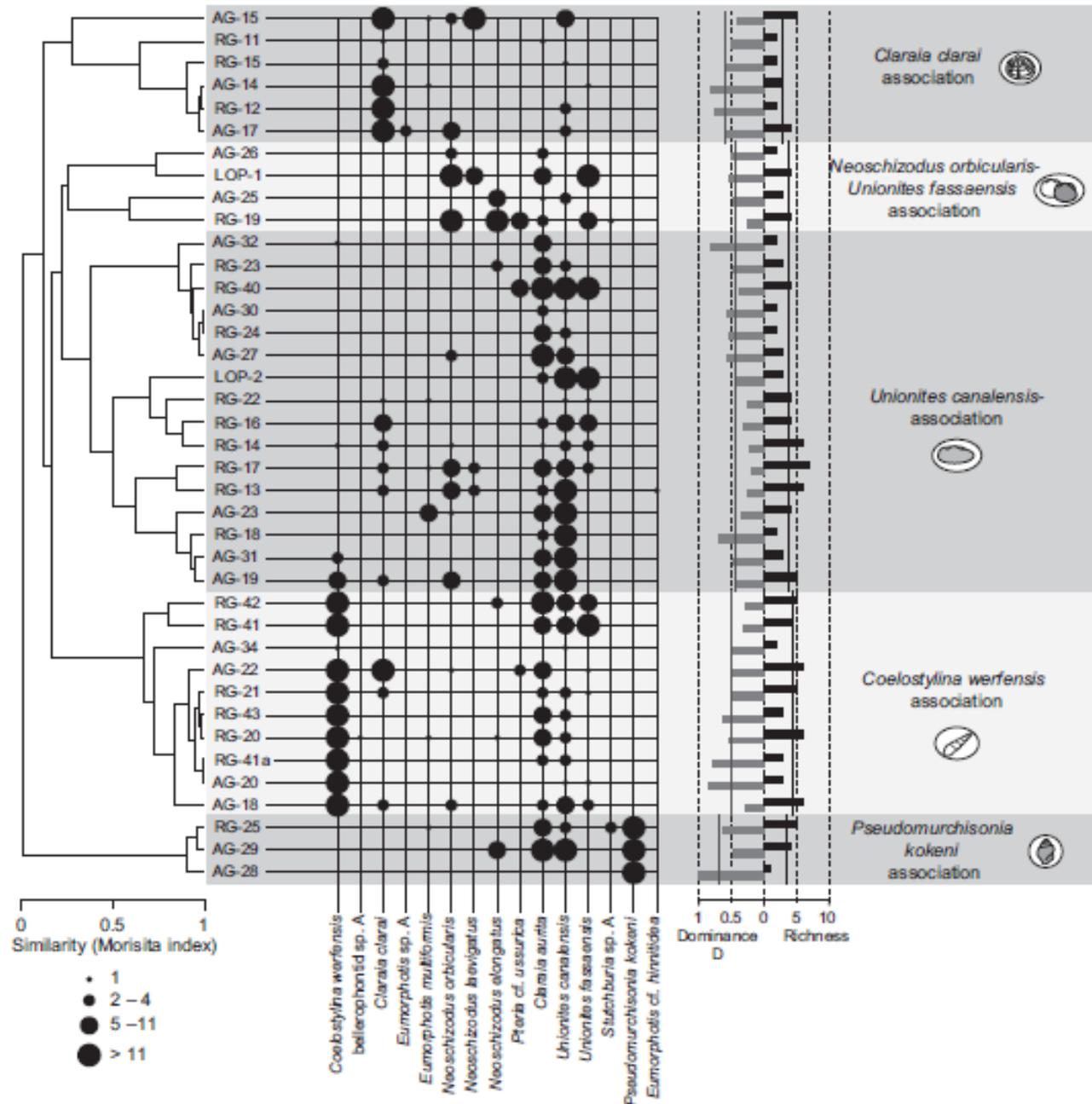
Also, see the `{cluster}` and `{rioja}` packages.

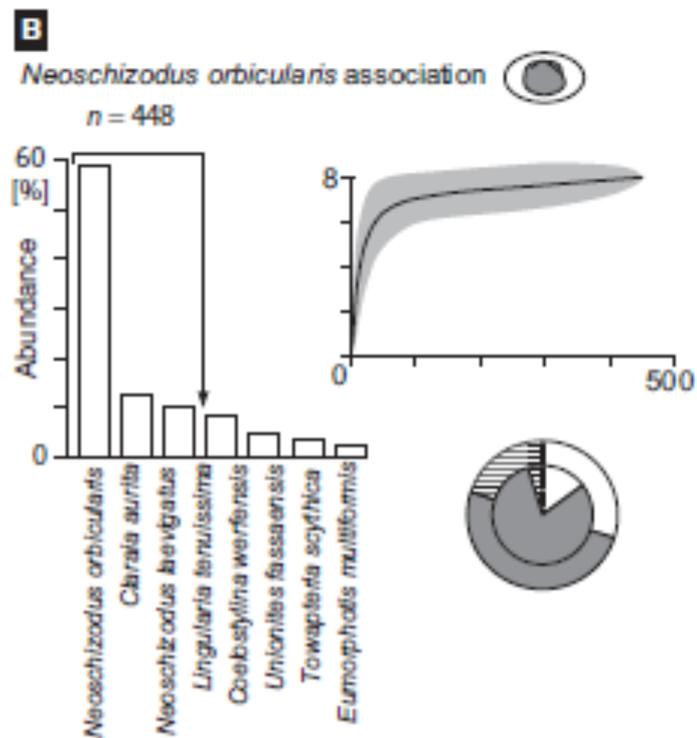
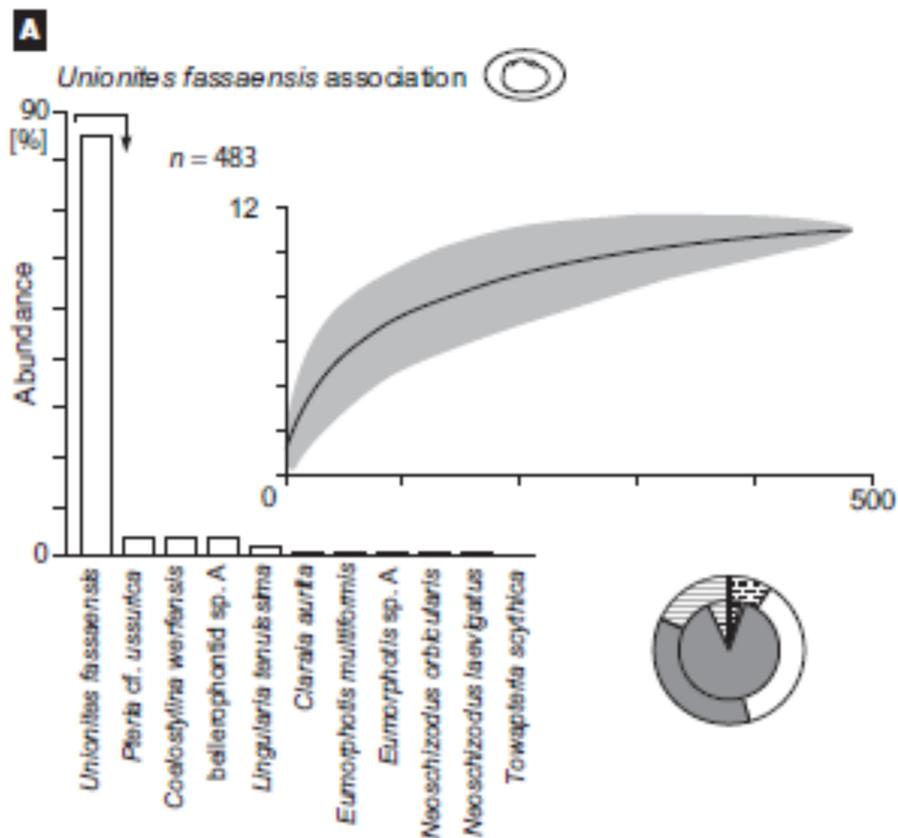
References for Cluster Analysis and Entropy in Ecology

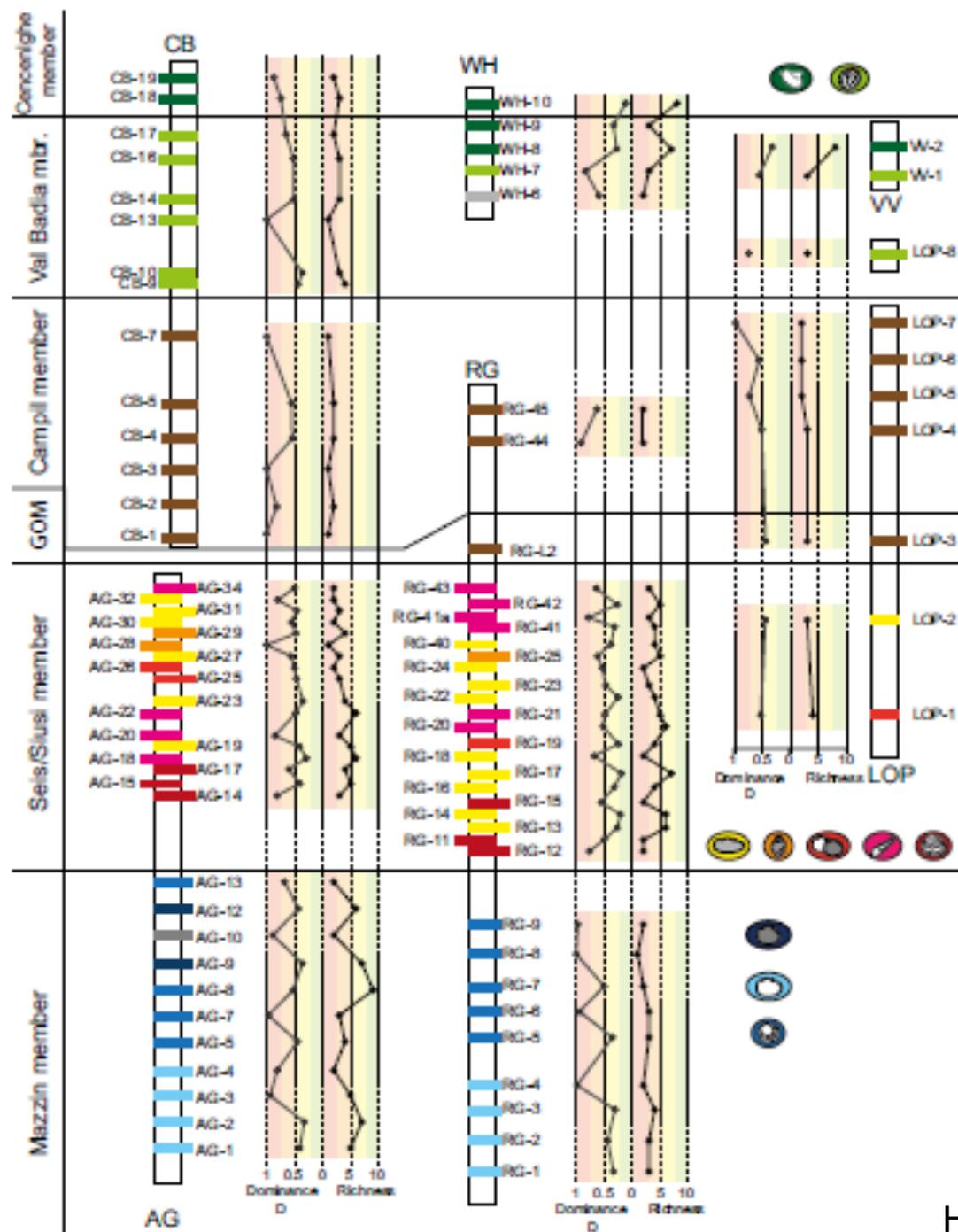
- Girvan, M. and Newman, M.E.J., 2002, Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, v. 99, p. 7821-7826.
- Hopcroft, J., Khan, O., Kulis, B., and Selman, B., 2004, Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences*, v. 101, p. 5249-5253.
- Kaufman, L. and Rousseeuw, P.J., 1990, *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York.
- Mantel, N., 1967, The detection of disease clustering and a generalized regression approach. *Cancer Research*, v. 27, p. 209-220.
- Morris, S.A. and Yen, G.G., 2004, Crossmaps: visualization of overlapping relationships in collections of journal papers. *Proceedings of the National Academy of Sciences*, v. 101, p. 5291-5296. (An independent re-invention of 2-way cluster analysis.)
- Palla, G., Derenyi, I., Farkas, I., and Vicsek, T., 2005, Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, v. 435, p. 814-818.
- Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D., 2004, Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, v. 101, p. 2658-2663.
- Slonim, N., Atwal, G.S., Tkačik, G., and Bialek, W., 2005, Information-based clustering. *Proceedings of the National Academy of Sciences*, v. 102, p. 18297-18302.
- Sneath, P.H.A. and Sokal, R.R., 1973, *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W.H. Freeman and Co., 573 p. (A classic – the basis for virtually all commonly used phenetic and paleoecological clustering methods.)



Marl-limestone alternations
 Marly limestone
 Limestone
 Siltstone
 Marlstone
 Sandstone
 Sandy dolomite







Script used to generate the figure on the previous slide

```
library(stats)
library(vegan)

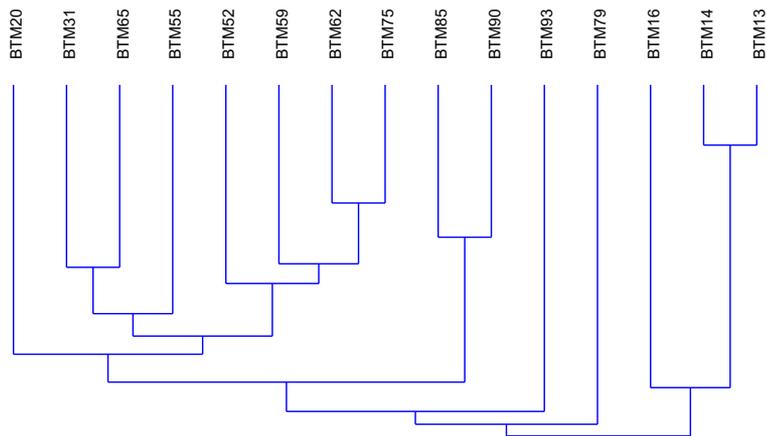
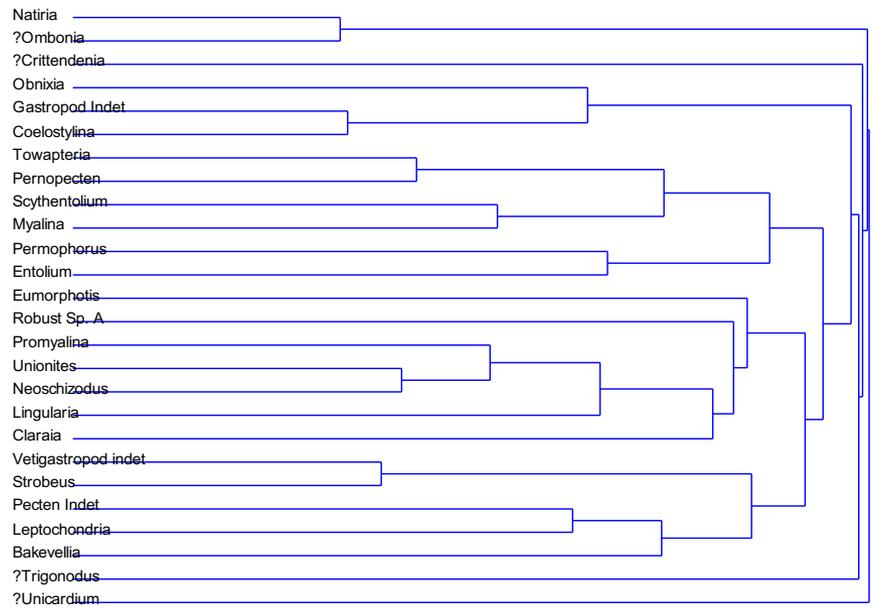
x <- read.csv("dummyorddata.csv") # an example of a simple multivariate dataset
x1 <- as.matrix(x[,-1])           # store as a matrix of numerical variables of interest
x1 <- wisconsin(x1)               # standardized/transform data as appropriate
rownames(x1) <- x[,1]            # store sample labels as rownames

myblue <- c("#00FFCC", "#00EFCC", "#00EECC", "#00DECC", "#00DDCC", "#00CDCC", "#00CCCC", "#00BCCC", "#00BBCC", "#00AACC",
"#0099CC", "#0088CC", "#0087CC", "#0077CC", "#0076CC", "#0066CC", "#0065CC", "#0055CC", "#0054CC", "#0044CC", "#0043CC", "#0033CC",
"#0032CC", "#0022CC", "#0021CC", "#0011CC", "#0011BB", "#0011AB", "#0010AA", "#001099", "#001088", "#001077", "#001066", "#001055", "#
001044", "#001033", "#001022", "#001011", "black")

rd <- vegdist(x1, 'bray')
rc <- hclust(rd, method='ward.D')
cd <- vegdist(t(x1), 'bray')
cc <- hclust(cd, method='ward.D')
op <- par(oma=c(0,0,3,0))

heatmap(x1, Rowv=as.dendrogram(rc), Colv=as.dendrogram(cc), cexRow=0.5, cexCol=0.7,
        col=myblue, margins=c(4,4))
mtext('method: "WARD.D"', side=3, adj=0.5, cex=0.8, line=6)
mtext('distance: "BRAY-CURTIS"', side=3, adj=0.5, cex=0.8, line=5)
par(op)
```

0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
5	3	0	0	0	0	0	0	0	0	0	0	89	62	83
2	0	0	0	10	0	1	0	0	3	0	0	361	0	0
2	0	0	0	0	0	0	3	0	0	0	0	184	0	0
0	0	4	0	0	0	4	0	0	0	0	0	0	0	0
0	0	5	0	0	0	0	1	0	0	0	0	0	0	0
0	0	0	0	1	9	5	0	1	0	0	0	0	0	0
0	1	3	0	0	6	2	5	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0	0	3	2	0	0
0	0	0	0	0	0	3	6	0	0	4	0	0	0	0
10	16	11	23	79	26	41	44	1	0	0	347	25	4	0
11	75	0	0	0	3	0	0	0	0	0	0	0	0	0
8	10	37	17	8	9	0	0	2	0	0	0	4	0	0
5	22	21	21	1	15	2	4	13	9	103	3	0	0	0
1	10	22	7	0	8	1	0	4	3	40	0	7	0	0
119	29	40	1	21	16	12	13	11	4	2	2	19	8	8
0	5	1	49	0	2	4	0	0	0	0	0	0	0	0
0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
0	0	0	0	8	0	0	0	0	1	0	0	0	0	0
9	0	0	9	10	0	0	0	0	0	0	0	0	1	0
3	0	0	7	3	0	2	25	0	0	0	0	0	0	0
7	0	1	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	0	0



Bray-Curtis Similarity
Single linkage