Hierarchical clustering: start

Watch video "Hierarchical Clustering - Fun and Easy Machine Learning" (10min) by Augmented Startups

https://www.youtube.com/watch?v=EUQY3hL38cw

(link in MyCourses)

Questions:

- Which linkage metric to use?
- What kinds of clusters can you find?
- Is there anything equivalent to large data?

Generic agglomerative hierarchical clustering

given D = intercluster distance ("linkage metric")

Initialize distance matrix $\mathbf{M}$
Repeat until termination:

1. pick closest pair of clusters C_i and C_j ($D(C_i, C_j)$ minimal)
2. merge clusters: $C_{ij} = C_i \cup C_j$
3. update $\mathbf{M}$
   - remove rows and cols of C_i and C_j
   - add a new row and col for C_{ij} + their entries (distances to C_{ij})

Famous linkage metrics

Single	$\min_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} \{d(\mathbf{x_1}, \mathbf{x_2})\}$	elongated, straggly, also concentric clusters
Complete	$\max_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} \{d(\mathbf{x_1}, \mathbf{x_2})\}$	small, compact, hyper-spherical, equal-sized
Average	$\frac{\sum_{\mathbf{x_1} \in C_1, \mathbf{x_2} \in C_2} d(\mathbf{x_1}, \mathbf{x_2})}{\lvert C_1 \rvert \lvert C_2 \rvert}$	quite compact; allows different sizes and densities
Minimum variance (Ward)	$SSE(C_1 \cup C_2) - SSE(C_1)$ $-SSE(C_2)$	compact, quite well-separated, hyperspherical; not elongated or very different sized
Distance of centroids	$d(\mathbf{c_1}, \mathbf{c_2})$	hyperspherical, equal-sized; not elongated

Famous linkage metrics

- linkage metric has a strong effect on results!
- **Warning**: most linkage metrics are sensitive to data order! ⇒ results may change if you shuffle data
- single linkage is not, but it is prone to “chaining effect”

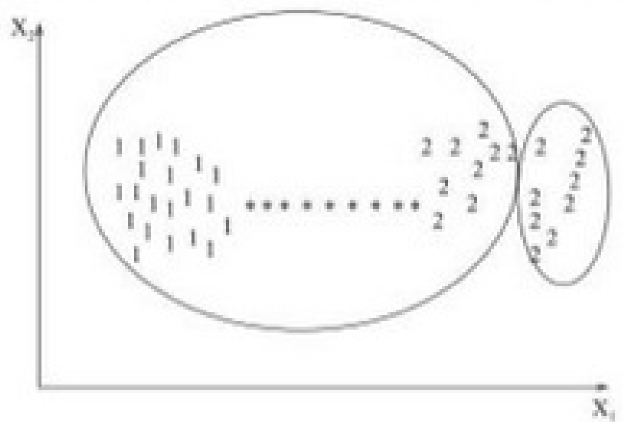


Figure 12. A single-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

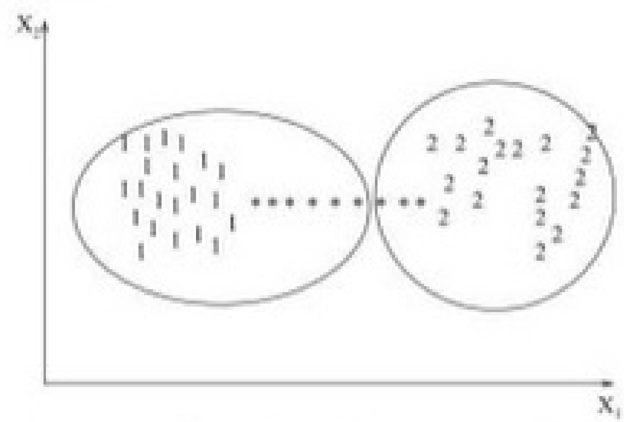


Figure 13. A complete-link clustering of a pattern set containing two classes (1 and 2) connected by a chain of noisy patterns (*).

image source https://www.slideshare.net/KalpaGunaratna/incremental-concpetual-clustering-reading-group-discussion

Example (Old Faithful data)

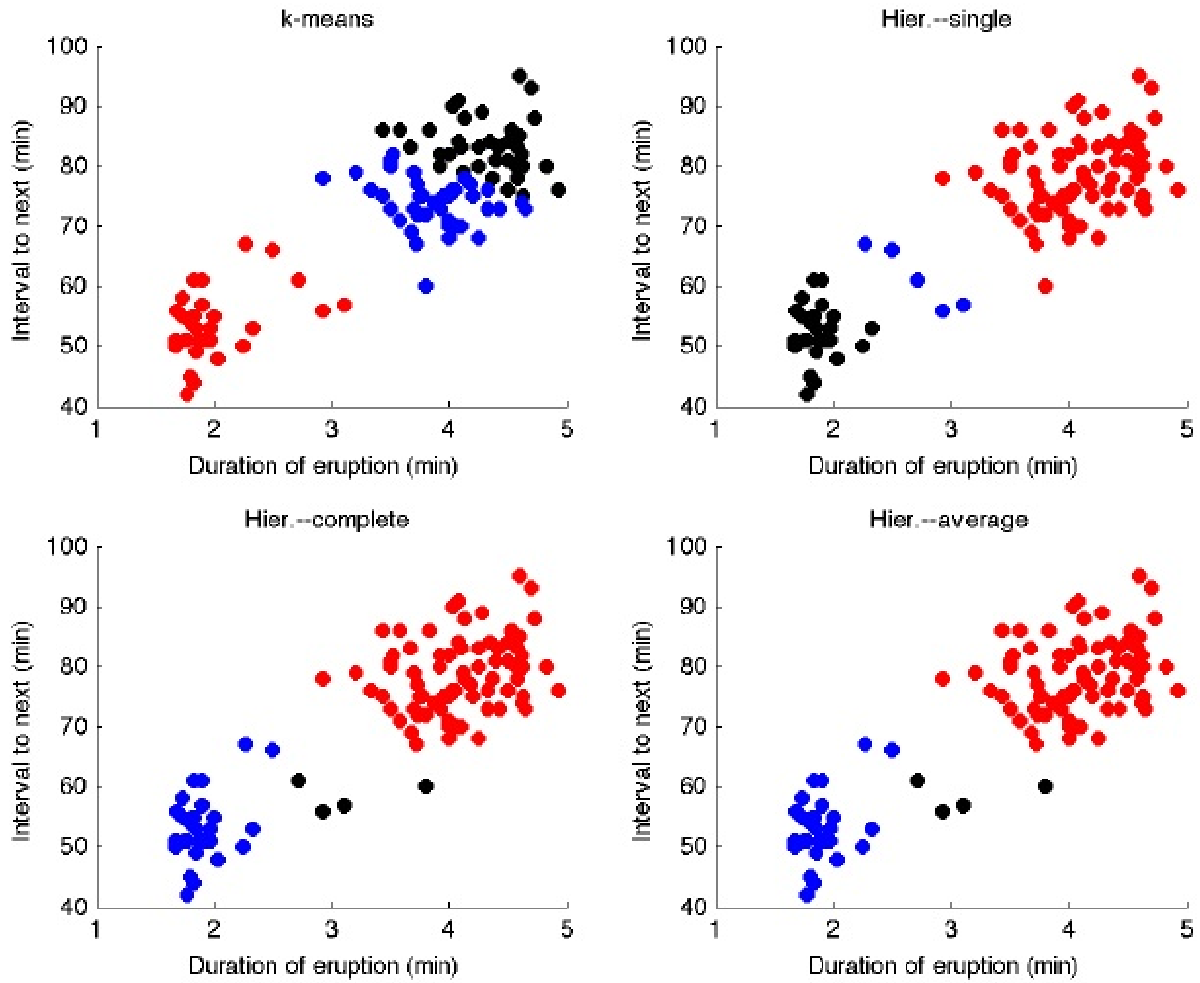


image source: Sungkyu Jung (2013) slides on clustering (STAT2221, Univ. of Pittsburgh)
`https://www.stat.pitt.edu/sungkyu/course/2221Fall13/lec7_clustering.pdf`

Connection to graph theory

Single linkage is related to **connected components** and complete linkage to **cliques**

Let $e_1, \ldots, e_m$ be edges of complete distance graph with weights $d_1 < d_2 < \ldots < d_m$. ($m = \frac{n(n-1)}{2}$, n data points)

Single linkage

1. Initialize: Create graph $\mathbf{G}$ without edges
   - i.e., n connected components and all data points in their own clusters
2. Repeat until one connected component
   - add new edge e_i with smallest d_i to $\mathbf{G}$
   - form clusters from connected components of $\mathbf{G}$

Connection to graph theory

Complete linkage

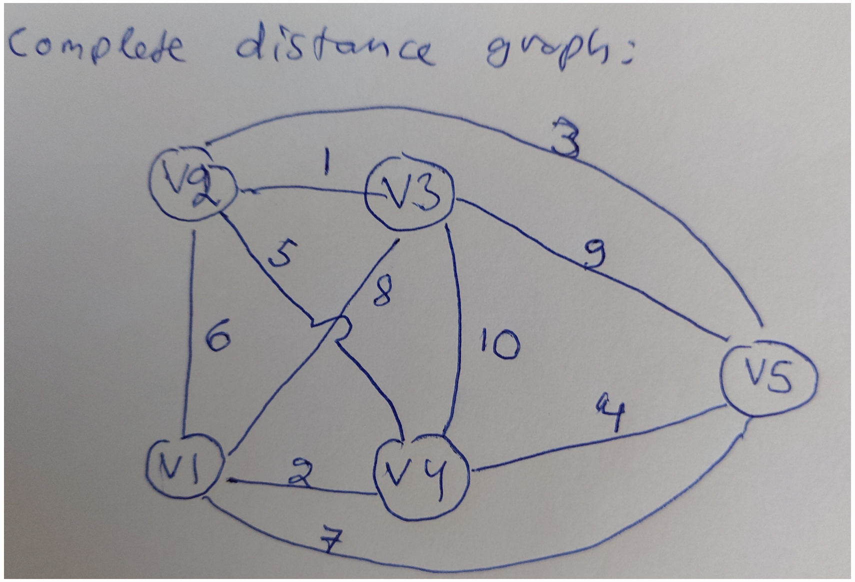
1. Initialize: Create graph $\mathbf{G}$ without edges
   - all data points in their own clusters
2. Repeat until $\mathbf{G}$ complete
   - add new edge e_i with smallest d_i to $\mathbf{G}$
   - if two of the current clusters form a clique in $\mathbf{G}$, merge them

Note: You are not allowed to break existing clusters, even if you would find alternative cliques

Task: graph-based single linkage clustering

Distance matrix:

	v1	v2	v3	v4	v5
v1	0	6	8	2	7
v2	6	0	1	5	3
v3	8	1	0	10	9
v4	2	5	10	0	4
v5	7	3	9	4	0



Add edges in the increasing order of weights $(2, 3, \ldots, 7)$. What are the corresponding single linkage clusters? Complete linkage clusters? (hometask)

Task

*Extra: single linkage clustering from minimum spanning trees

Begin from complete distance graph $\mathbf{G}$ and search its minimum spanning tree (MST)

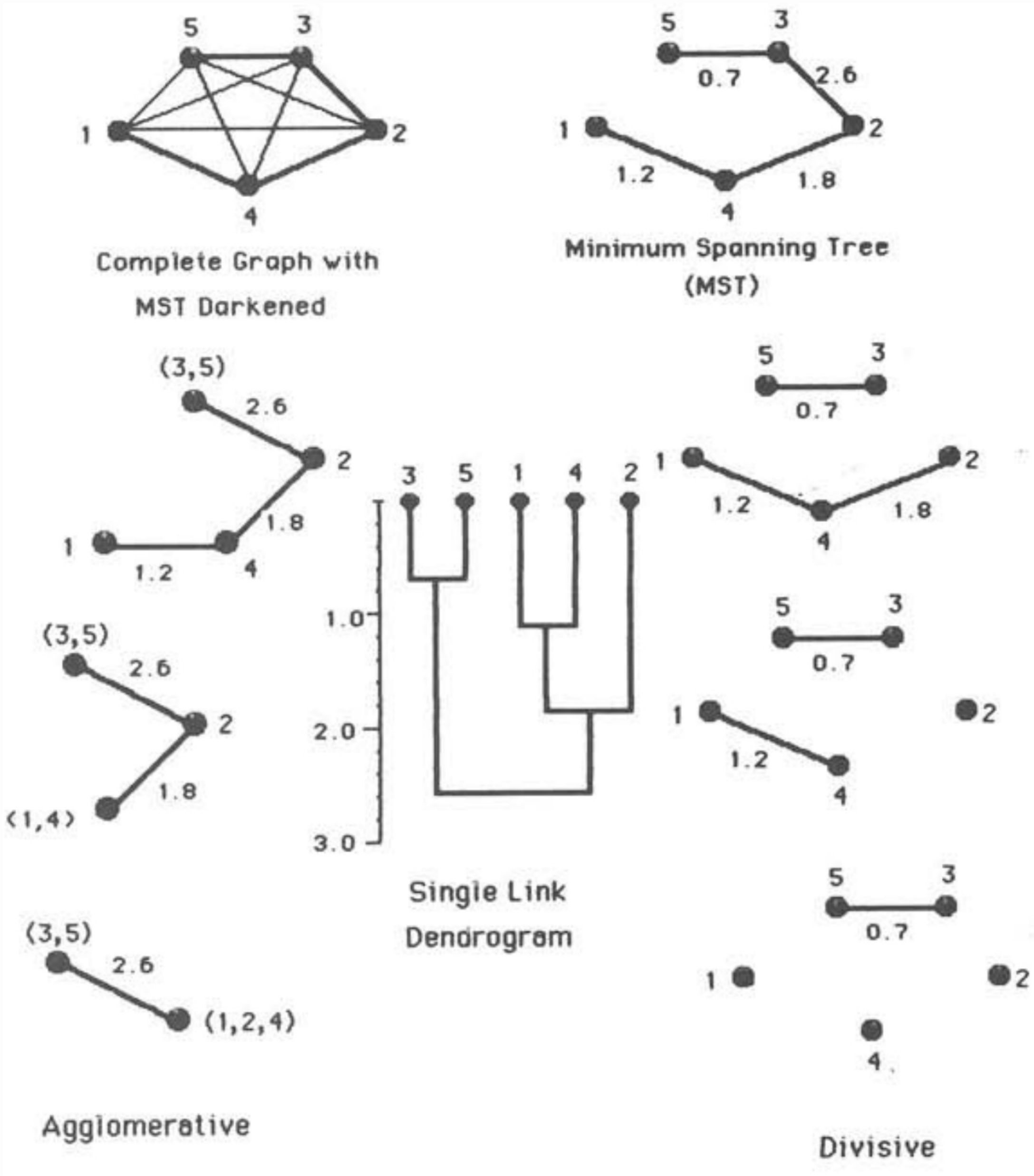
Repeat until all objects belong to one cluster:

1. Merge two clusters that are connected in the MST and have the smallest edge weight
2. Set the edge weight as ∞

Notes:

- The same can be done in a divisive manner: cut the MST edges in the descending order by weight.
- If there are no proximity ties, the result is the same as normal single likage clustering

*Example (Jain and Dubes 1988, Fig 3.6)



Agglomerative or divisive?

- Agglomerative = bottom-up
  - cheaper and easier to implement
  - still slow, at least $\mathcal{O}(n^2)$
  - early decisions based on local patterns, cannot cancel later
- Divisive = top-down
  - often better quality clustering
    $\Leftarrow$ large clusters created early, based on global distribution
  - fastest $\mathcal{O}(n^2 \log(n))$

Bisecting K-means

Idea: combine divisive hierarchical and K-means.
Given K and q=number of iterations

1. Initialization: put all data points into one cluster
2. Repeat until K clusters:
   - choose cluster C to split (with largest SSE)
   - split C q times with 2-means
   - keep the best split (two new clusters)

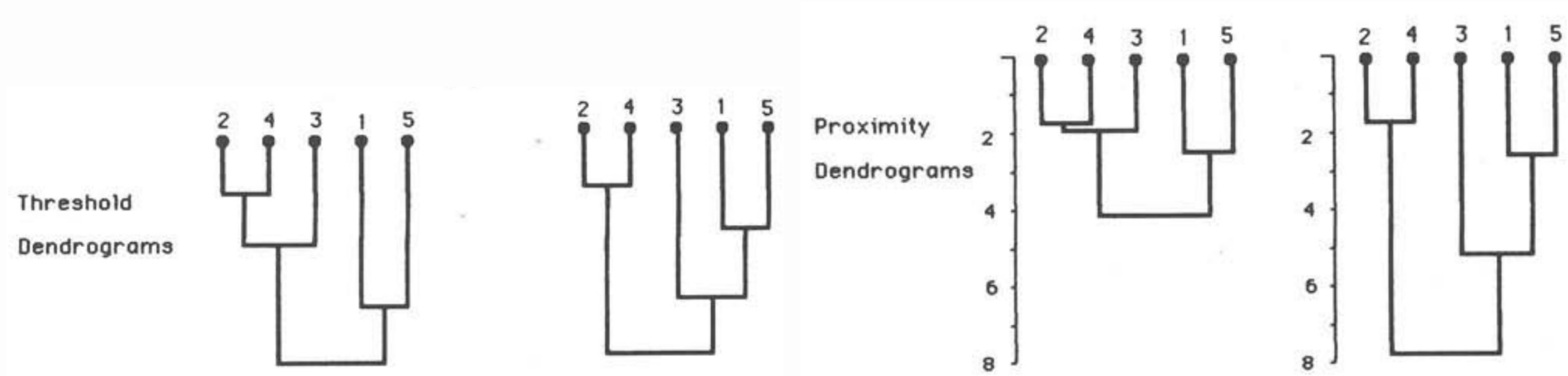
\+ efficient (like K-means)

\+ good results (comparable to hierarchical)

On dendrograms

- **Threshold dendrograms**: in which order the clusters were formed
- **Proximity dendograms**: at which proximities they were formed

$$
\begin{array}{c|cccc}
 & x_2 & x_3 & x_4 & x_5 \\
\hline
x_1 & 5.8 & 4.2 & 6.9 & 2.6 \\
x_2 & & 6.7 & 1.7 & 7.2 \\
x_3 & & & 1.9 & 5.6 \\
x_4 & & & & 7.6
\end{array}
$$



single complete single complete

image source Jain & Dubes 1988 Fig 3.5

Dendrogram example

Here real (biological) classes of data points are shown under the dendrogram

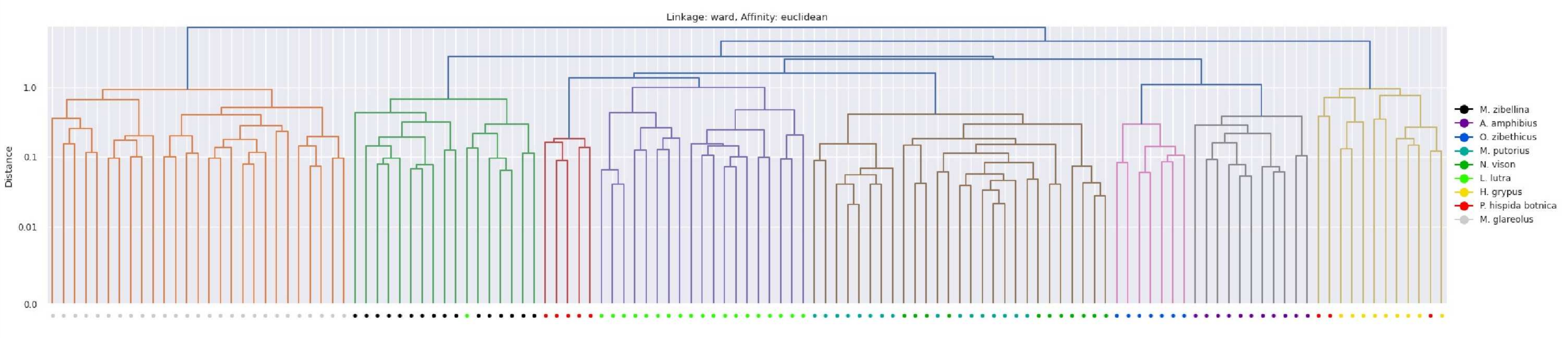


image author Lehtiniemi 2020

Another dendrogram example

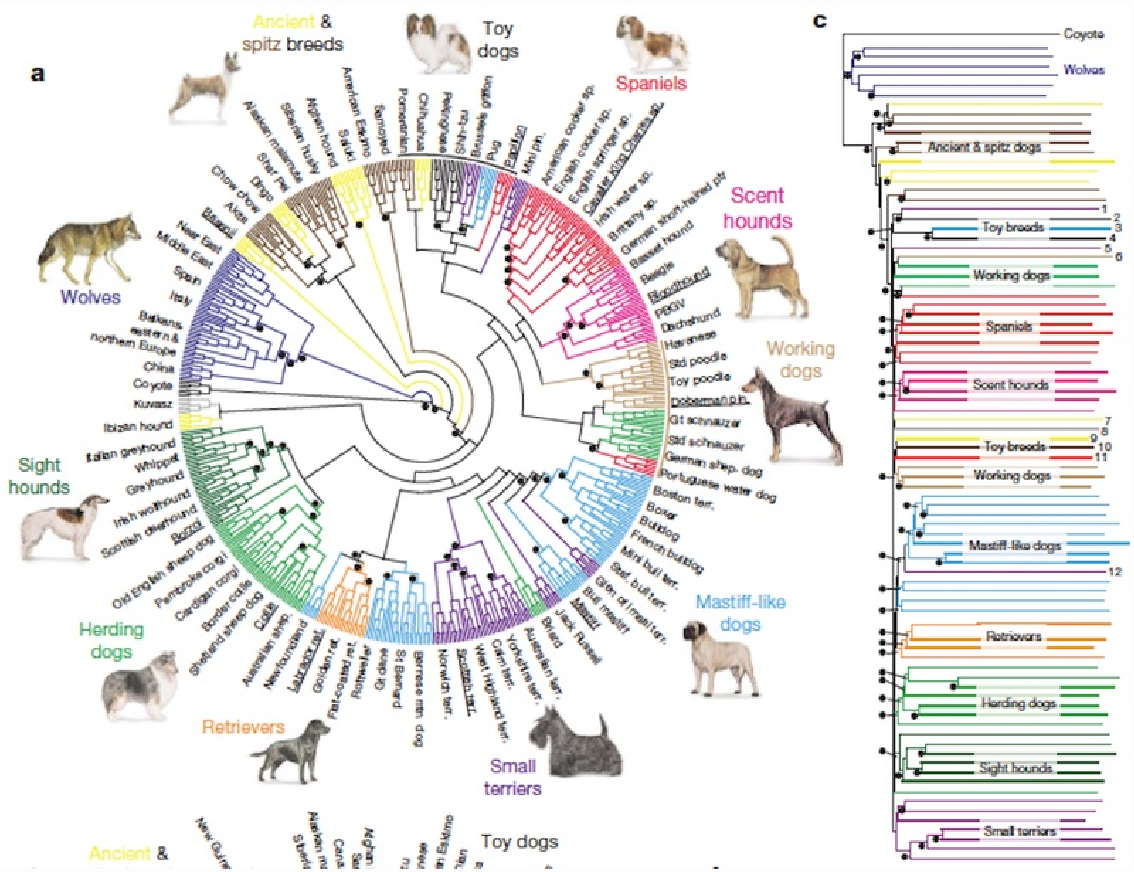


image source
https://www.instituteofcaninebiology.org/how-to-read-a-dendrogram.html

Summary

- useful information on clustering structure
  - dendrograms!
- linkage metrics have a strong effect
  - Beware: most metrics sensitive to data order!
- connections to graph theory (single $\leftrightarrow$ connected components, complete $\leftrightarrow$ cliques)
- inefficient for really large data (at least $O(n^2)$ time and space)

Voluntary task: Fill a summary table!

method	data type	cluster type	benefits	drawbacks
K-representatives				
K-means				
K-medoids				
...				
Hierarchical				
singe-link				
...				
Graph-based				
Density-based				
Probabilistic				

Further reading

- Gan, Ma, Wu: Data clustering – theory, algorithms, and applications. SIAM 2007.
- Jain and Dubes: Algorithms for clustering data. Prentice-Hall 1988.