



Genome Analysis (Integrative Bioinformatics & Genomics) 2008

Lecture 9

Pattern recognition and phylogeny

Patterns

Some are easy some are
not

- Knitting patterns
- Cooking recipes
- Pictures (dot plots)
- Colour patterns
- Maps

*In 2D and 3D humans are hard to be beat by a
computational pattern recognition technique,
but humans are not so consistent*

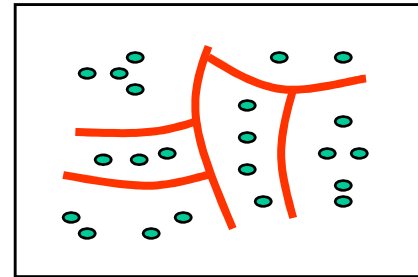
Example of algorithm reuse:

Data clustering

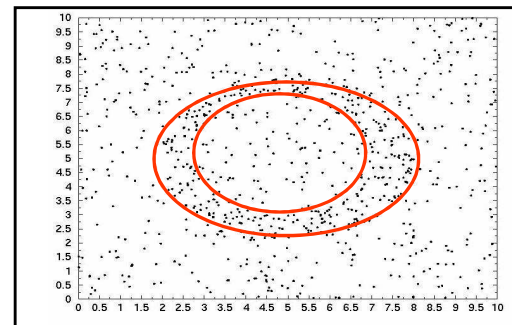
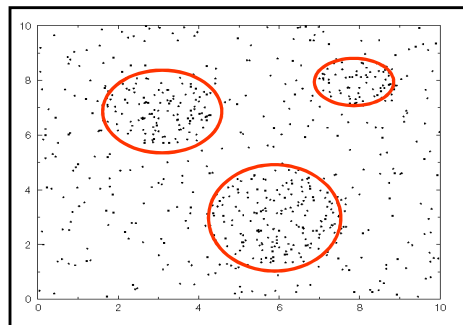
- Many biological data analysis problems can be formulated as clustering problems
 - microarray gene expression data analysis
 - identification of regulatory binding sites (similarly, splice junction sites, translation start sites,)
 - (yeast) two-hybrid data analysis (experimental technique for inference of protein complexes)
 - phylogenetic tree clustering (for inference of horizontally transferred genes)
 - protein domain identification
 - identification of structural motifs
 - prediction reliability assessment of protein structures
 - NMR peak assignments

Data Clustering Problems

- **Clustering**: partition a data set into clusters so that data points of the same cluster are “similar” and points of different clusters are “dissimilar”

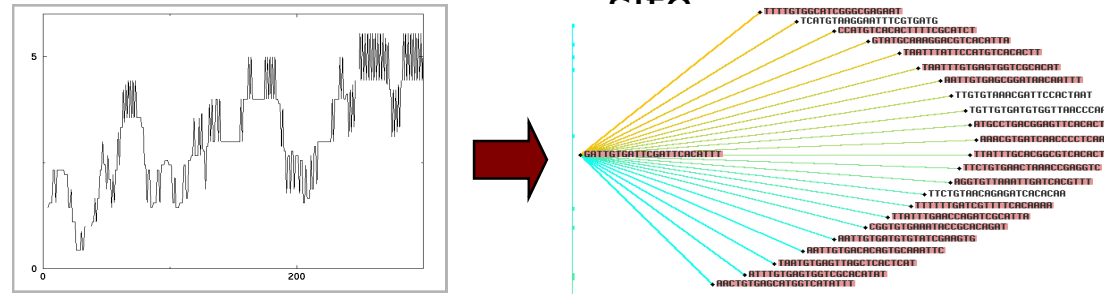


- **Cluster identification** -- identifying clusters with significantly different features than the background



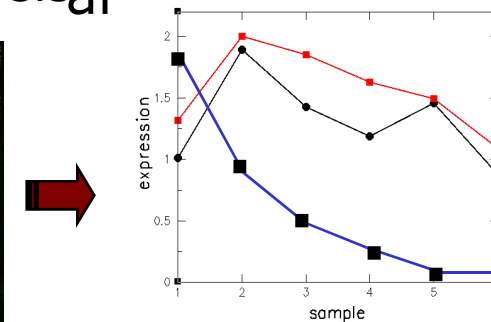
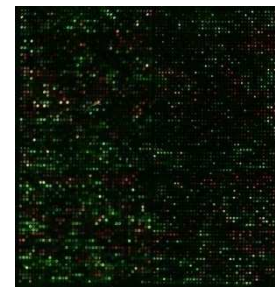
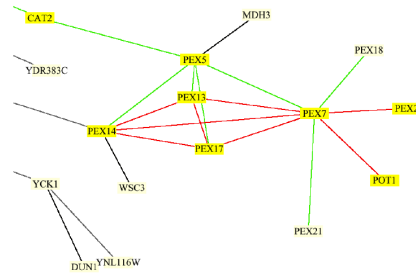
Application Examples

- Regulatory binding site identification: CRP (CAP) binding



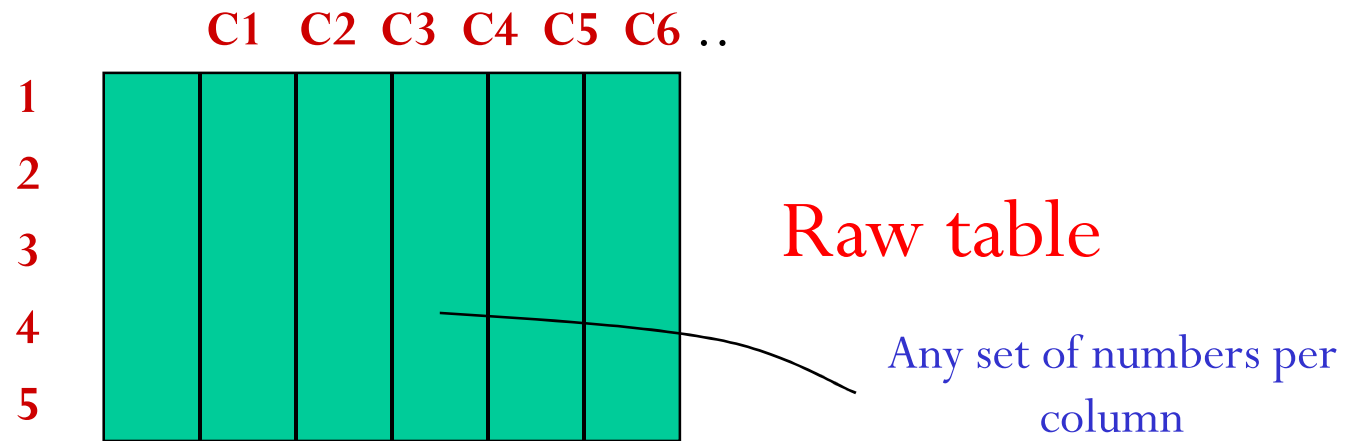
Gene expression data

- Two hybrid data analysis



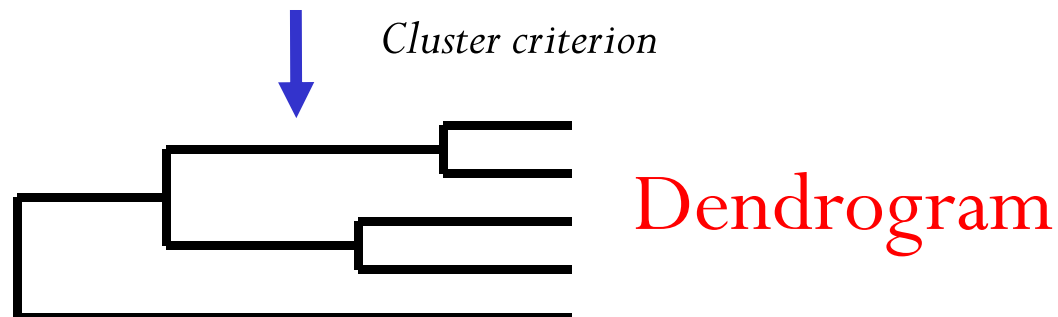
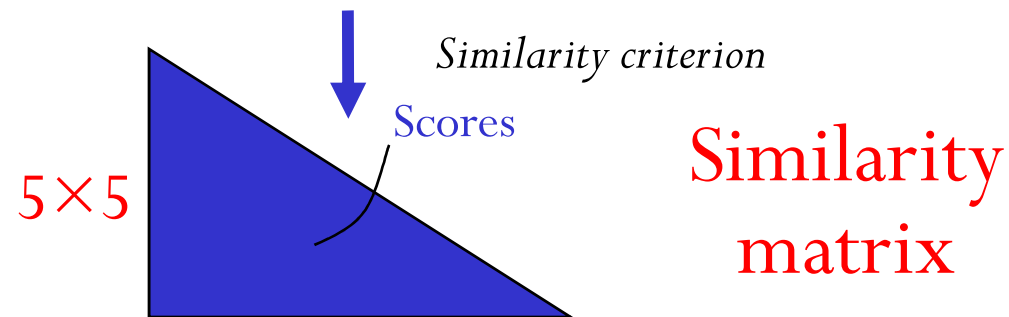
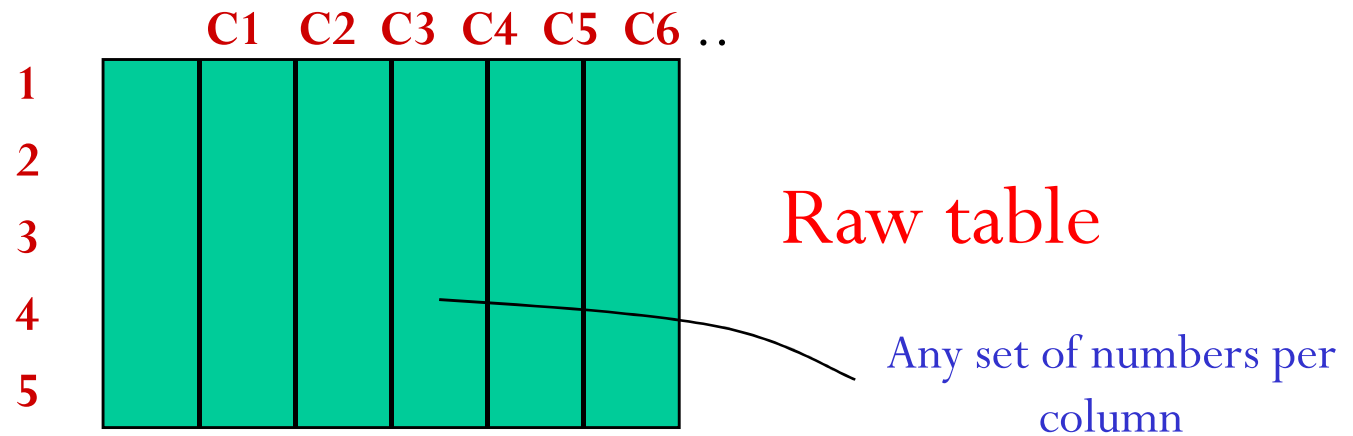
These problems are all solvable by a clustering algorithm

Multivariate statistics – Cluster analysis



- Multi-dimensional problems
- Objects can be viewed as a cloud of points in a multidimensional space
- Need ways to group the data

Multivariate statistics – Cluster analysis



Comparing sequences

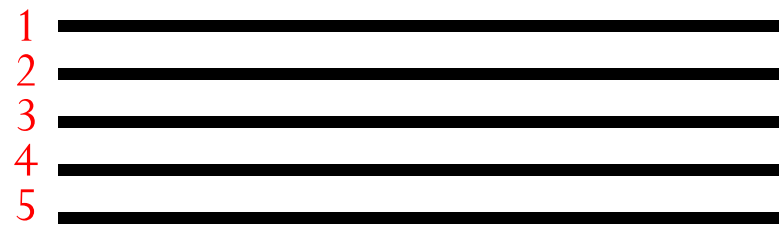
- Similarity Score -

Many properties can be used:

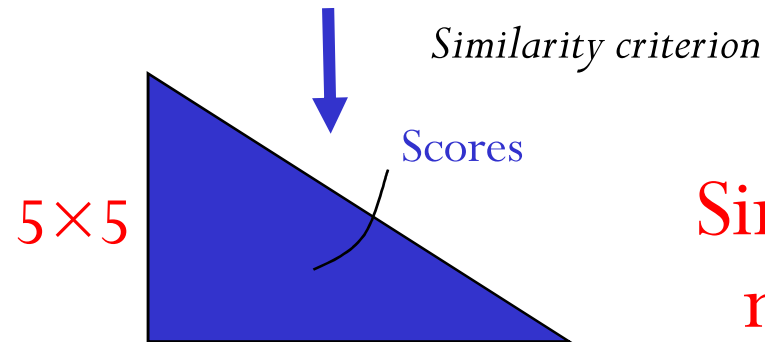
- Nucleotide or amino acid composition
- Isoelectric point
- Molecular weight
- Morphological characters
- But: molecular evolution through sequence alignment

Multivariate statistics – Cluster analysis

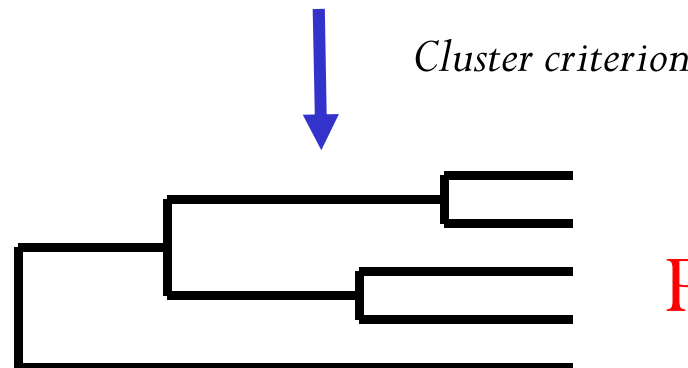
Now for sequences



Multiple sequence
alignment



Similarity
matrix



Phylogenetic tree

Lactate dehydrogenase multiple alignment

Human	-KITVVGVGAVGMACAISILMKDLADELALVDVIEDKLKGEMMDLQHGSFLRTPKIVSGKDYNVTANSKLVIIITAGARQ
Chicken	-KISVVGVGAVGMACAISILMKDLADELTLDVVEDKLKGEMMDLQHGSFLKTPKITSGKDYSVTAHSKLVIVTAGARQ
Dogfish	-KITVVGVGAVGMACAISILMKDLADEVALVDVMDKLGEMMDLQHGSFLHTAKIVSGKDYSVSAGSKLVVITAGARQ
Lamprey	SKVTIVGVGQVGMAAAISVLLRDLADELALVDVVEDRLKGEMMDLLHGSFLKTAKIVADKDYSVTAGSRLVVVITAGARQ
Barley	TKISVIGAGNVGMAIAQTILTQNLADEIALVDALPDKLRGEALDLQHAAAFLLPRVRI -SGTDAAVTKNSDLVIVTAGARQ
Maizey casei	-KVILVGDGAVGSSYAYAMVLQGIAQEIGIVDIFKDKTKGDAIDLSNALPFTSPKKIYSA -EYSDAKDADLVVITAGAPQ
Bacillus	TKVSVIGAGNVGMAIAQTILTRDLADEIALVDAVPDKLRGEMDLQHAAAFLLPRTRLVSGTDMSVTRGSDLVIVTAGARQ
Lacto__ste	-RVVVIGAGFVGASYVFALMNQGIADIEVLIDANESKAIGDAMDFNHGKVFAPKPVDIWHGDYDDCRDADLVVICAGANQ
Lacto_plant	QKVVLVGDGAVGSSYAFAMAQQGIAEEFVIVDVVKDRTKGDALDLEDAQAFTAPKKIYSG -EYSDCKDADLVVITAGAPQ
Therma_mari	MKIGIVGLGRVGSSTAFALLMKGFAREMVLIDVKKRAEGDALDLIHGTPFTRRANIYAG -DYADLKGSDVVIVAAGVPQ
Bifido	-KLAVIGAGAVGSTLAFAAAQRGIAREIVLEDIAKERVEAEVLDMQHGSSEFYPTVSIDGSDDPEICRDADMVVITAGPRQ
Thermus_aqua	MKVGIVGSGFVGSATAYALVLQGVAREVVLVDLDRKLAQAHAEDILHATPFAHPVWVRSGW -YEDLEGARVVIVAAGVAQ
Mycoplasma	-KIALIGAGNVGNSFLYAAMNQGLASEYGIIDINPDFADGNAFDGEDASASLPFPISVSRYEYKDLKDADFIVITAGRPQ

Distance Matrix

		1	2	3	4	5	6	7	8	9	10	11	12	13
1	Human	0.000	0.112	0.128	0.202	0.378	0.346	0.530	0.551	0.512	0.524	0.528	0.635	0.637
2	Chicken	0.112	0.000	0.155	0.214	0.382	0.348	0.538	0.569	0.516	0.524	0.524	0.631	0.651
3	Dogfish	0.128	0.155	0.000	0.196	0.389	0.337	0.522	0.567	0.516	0.512	0.524	0.600	0.655
4	Lamprey	0.202	0.214	0.196	0.000	0.426	0.356	0.553	0.589	0.544	0.503	0.544	0.616	0.669
5	Barley	0.378	0.382	0.389	0.426	0.000	0.171	0.536	0.565	0.526	0.547	0.516	0.629	0.575
6	Maizey	0.346	0.348	0.337	0.356	0.171	0.000	0.557	0.563	0.538	0.555	0.518	0.643	0.587
7	Lacto_casei	0.530	0.538	0.522	0.553	0.536	0.557	0.000	0.518	0.208	0.445	0.561	0.526	0.501
8	Bacillus_stea	0.551	0.569	0.567	0.589	0.565	0.563	0.518	0.000	0.477	0.536	0.536	0.598	0.495
9	Lacto_plant	0.512	0.516	0.516	0.544	0.526	0.538	0.208	0.477	0.000	0.433	0.489	0.563	0.485
10	Therma_mari	0.524	0.524	0.512	0.503	0.547	0.555	0.445	0.536	0.433	0.000	0.532	0.405	0.598
11	Bifido	0.528	0.524	0.524	0.544	0.516	0.518	0.561	0.536	0.489	0.532	0.000	0.604	0.614
12	Thermus_aqua	0.635	0.631	0.600	0.616	0.629	0.643	0.526	0.598	0.563	0.405	0.604	0.000	0.641
13	Mycoplasma	0.637	0.651	0.655	0.669	0.575	0.587	0.501	0.495	0.485	0.598	0.614	0.641	0.000

How can you see that this is a distance matrix?

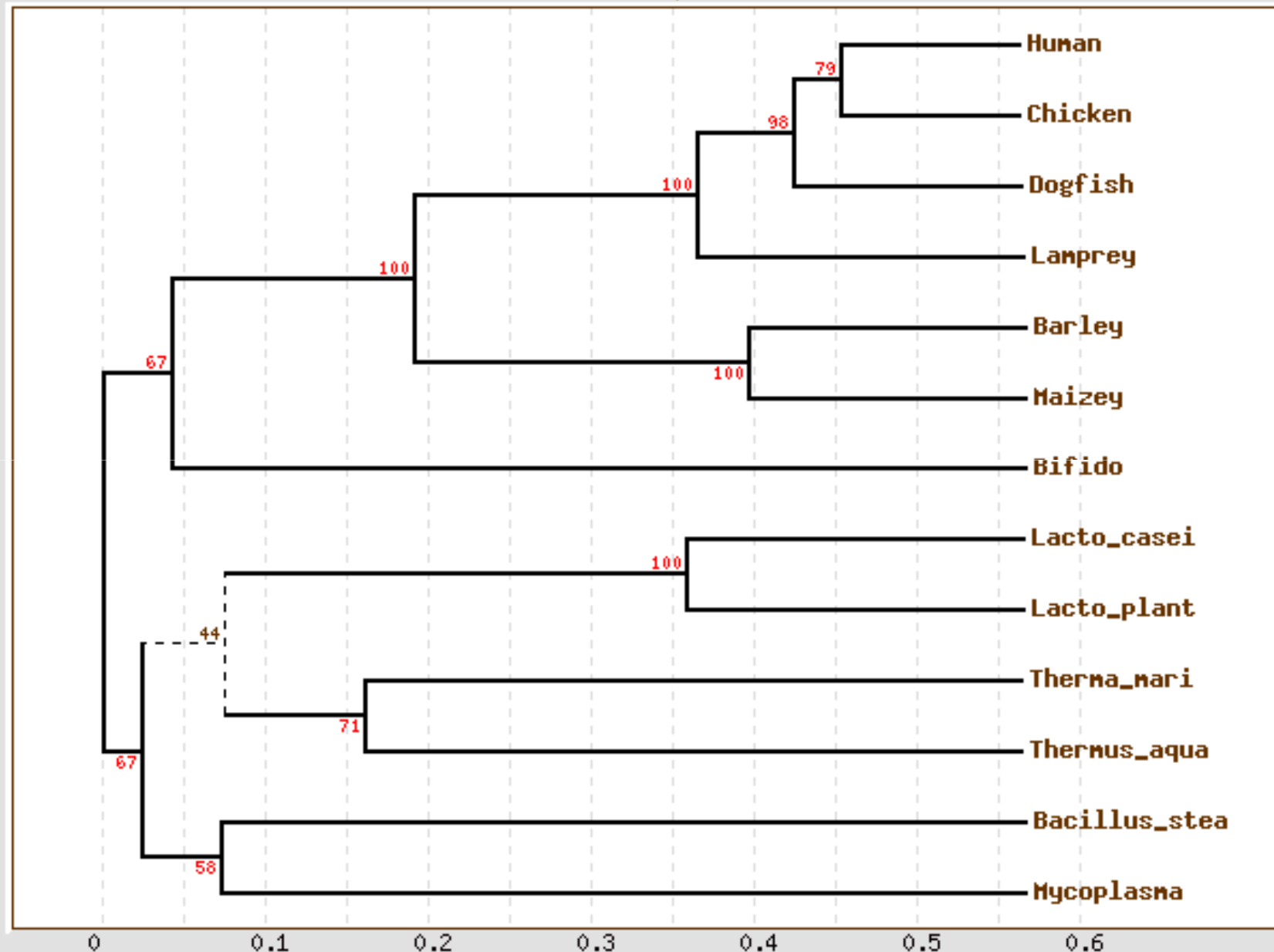
Cluster algorithm

PHYLOGENETIC TREE

with bootstrap values

Phylogram

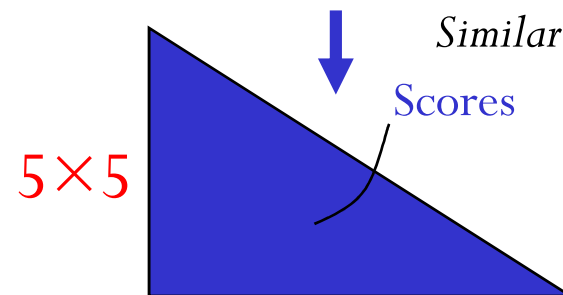
tr16424_1u.png



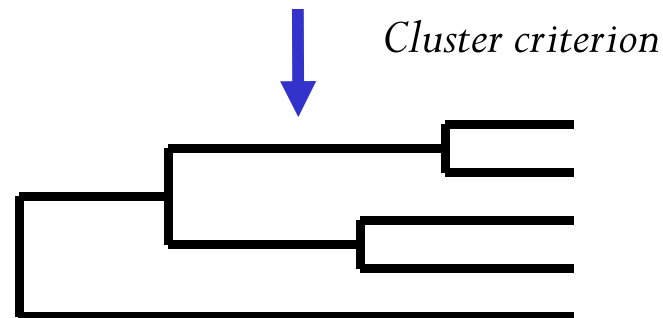
Multivariate statistics – Cluster analysis

	C1	C2	C3	C4	C5	C6	..
1							
2							
3							
4							
5							

Data table



Similarity matrix



Dendrogram/tree

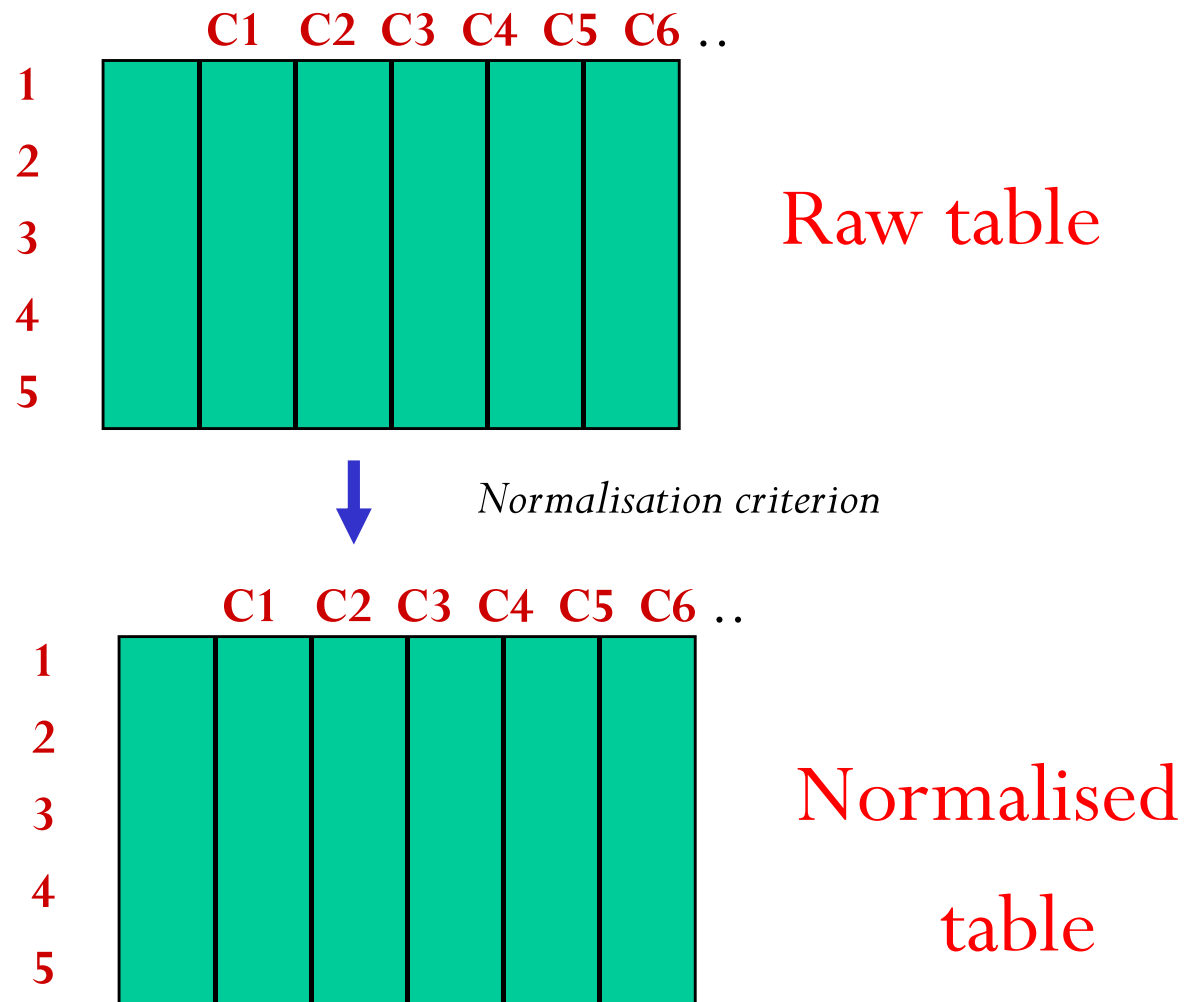
Multivariate statistics – Cluster analysis

Why do it?

- Finding a true typology
- Model fitting
- Prediction based on groups
- Hypothesis testing
- Data exploration
- Data reduction
- Hypothesis generation

But you can never prove a classification/typology!

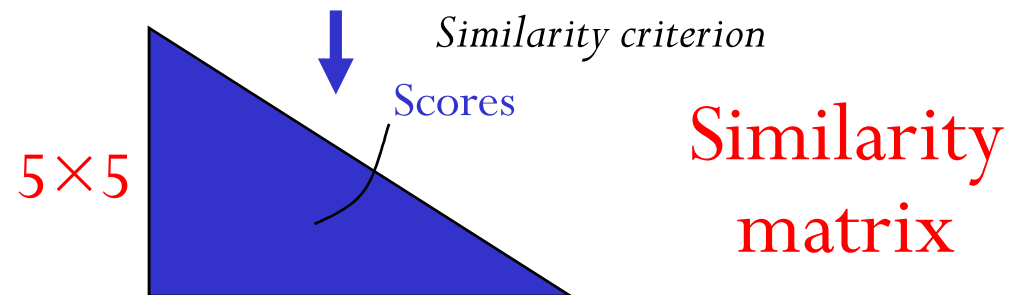
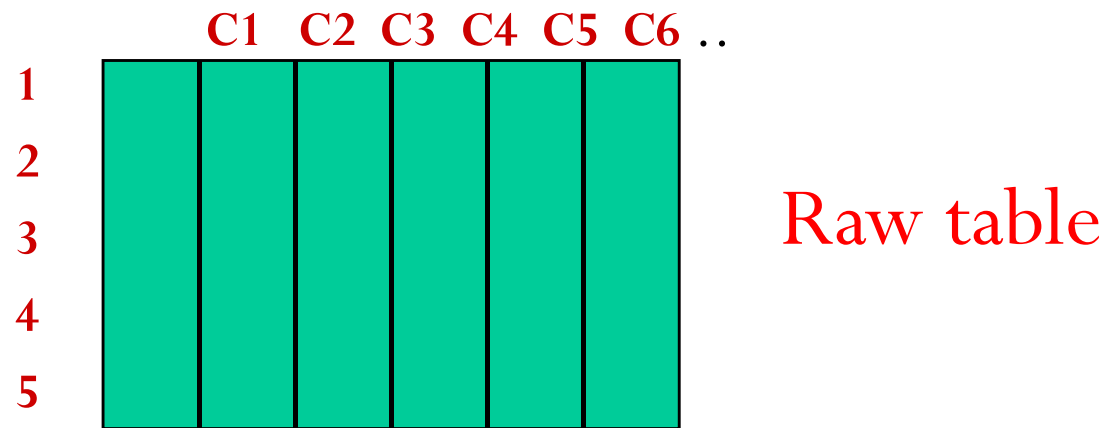
Cluster analysis – data normalisation/weighting



Column normalisation $x/\max$

Column range normalise $(x-\min)/(\max-\min)$

Cluster analysis – (dis)similarity matrix



$$D_{i,j} = (\sum_k |x_{ik} - x_{jk}|^r)^{1/r} \quad \text{Minkowski metrics}$$

$r = 2$ Euclidean distance

$r = 1$ City block distance

(dis)similarity matrix

$$D_{i,j} = (\sum_k |x_{ik} - x_{jk}|^r)^{1/r} \quad \text{Minkowski metrics}$$

$r = 2$ Euclidean distance

$r = 1$ City block distance

EXAMPLE:

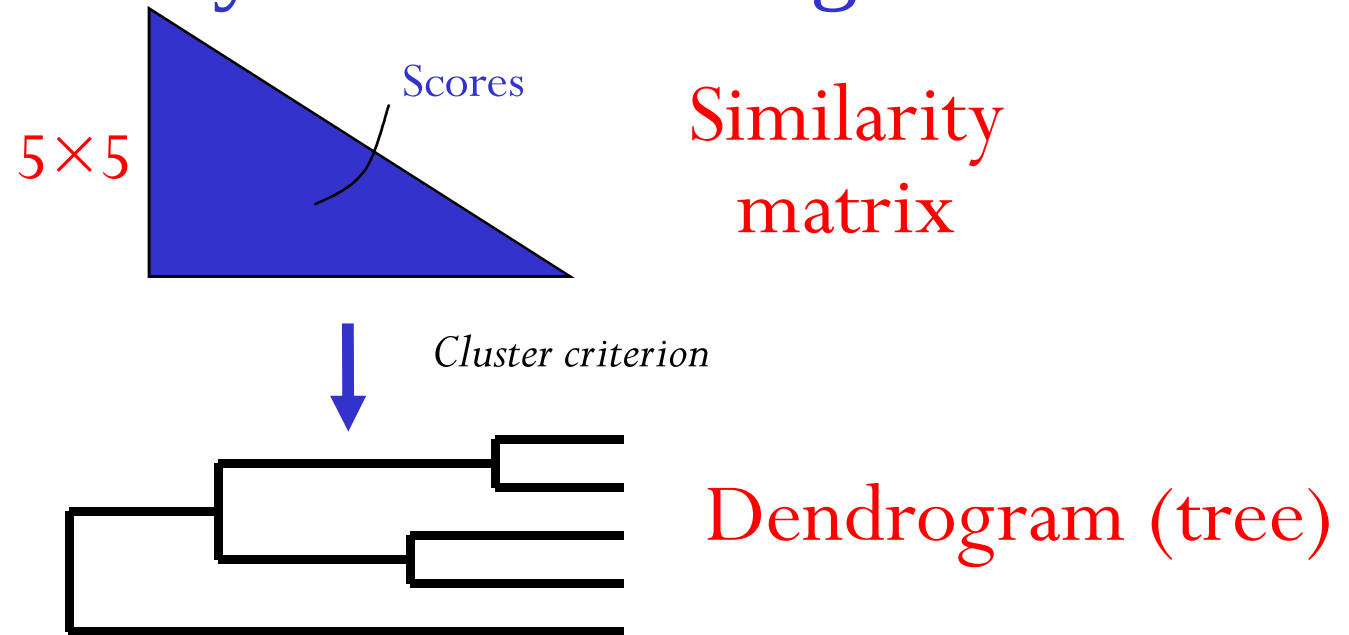
	length	height	width
Cow1	11	7	3
Cow 2	7	4	5

Note: The differences between the two rows are 4, 3, and -2, indicated by blue brackets and numbers in the original image.

$$\text{Euclidean dist.} = \sqrt{4^2 + 3^2 + (-2)^2} = \sqrt{29} = 5.39$$

$$\text{City Block dist.} = |4| + |3| + |-2| = 9$$

Cluster analysis – Clustering criteria



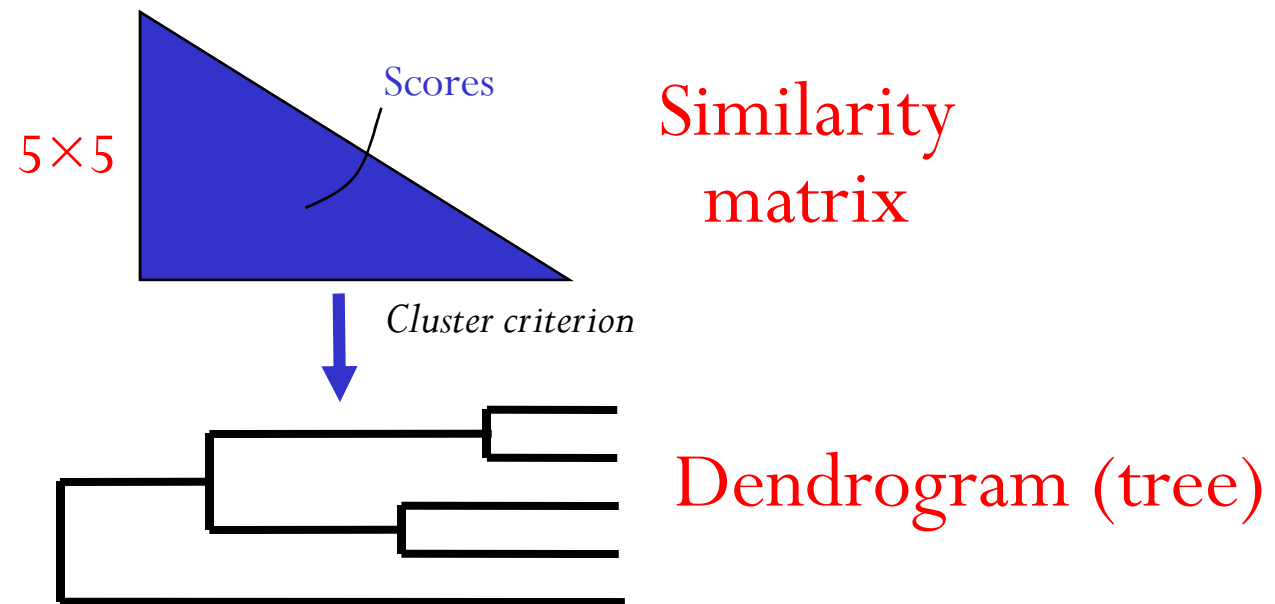
Single linkage - Nearest neighbour

Complete linkage – Furthest neighbour

Group averaging – UPGMA

Neighbour joining – global measure, used to make a
Phylogenetic Tree

Cluster analysis – Clustering criteria



Four different clustering criteria:

Single linkage - Nearest neighbour

Complete linkage – Furthest neighbour

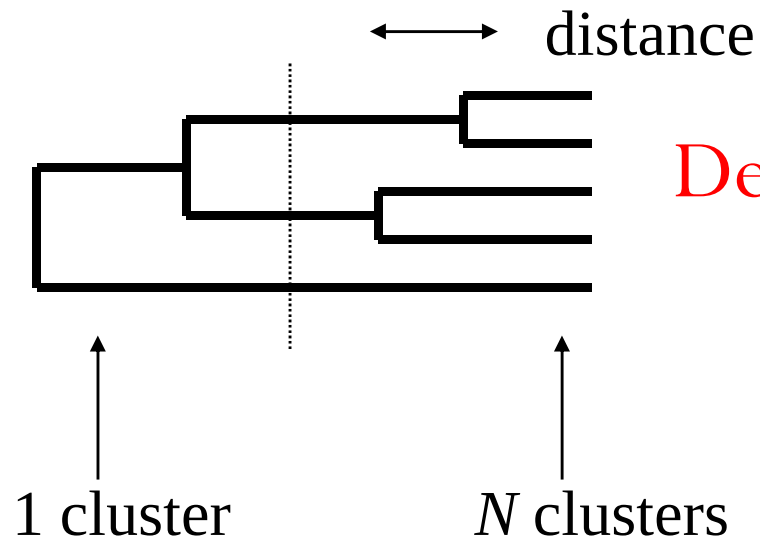
Group averaging – UPGMA

Neighbour joining (global measure)

Note: these are all *agglomerative* cluster techniques; i.e. they proceed by merging clusters as opposed to techniques that are *divisive* and proceed by cutting clusters

Cluster analysis – Clustering criteria

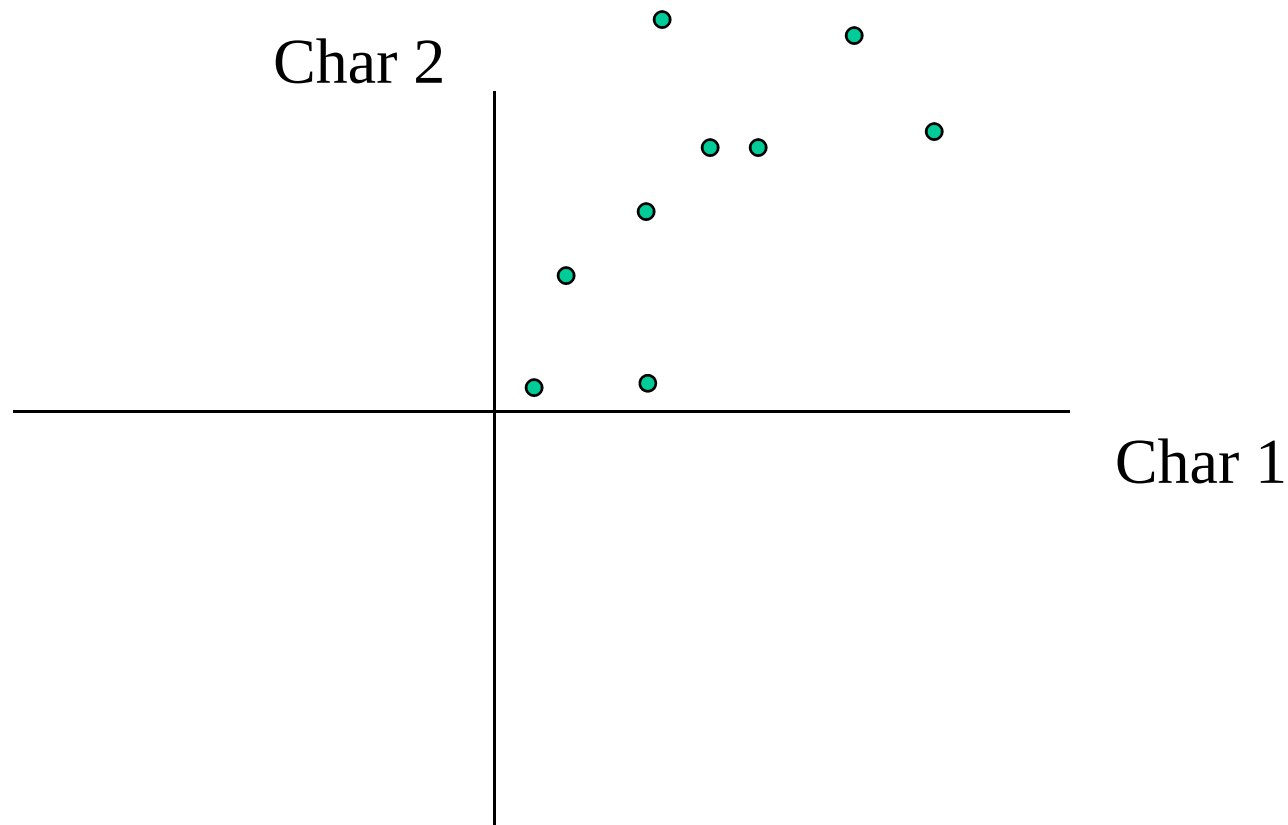
1. Start with N clusters of 1 object each
2. Apply clustering distance criterion iteratively until you have 1 cluster of N objects
3. Most interesting clustering somewhere in between



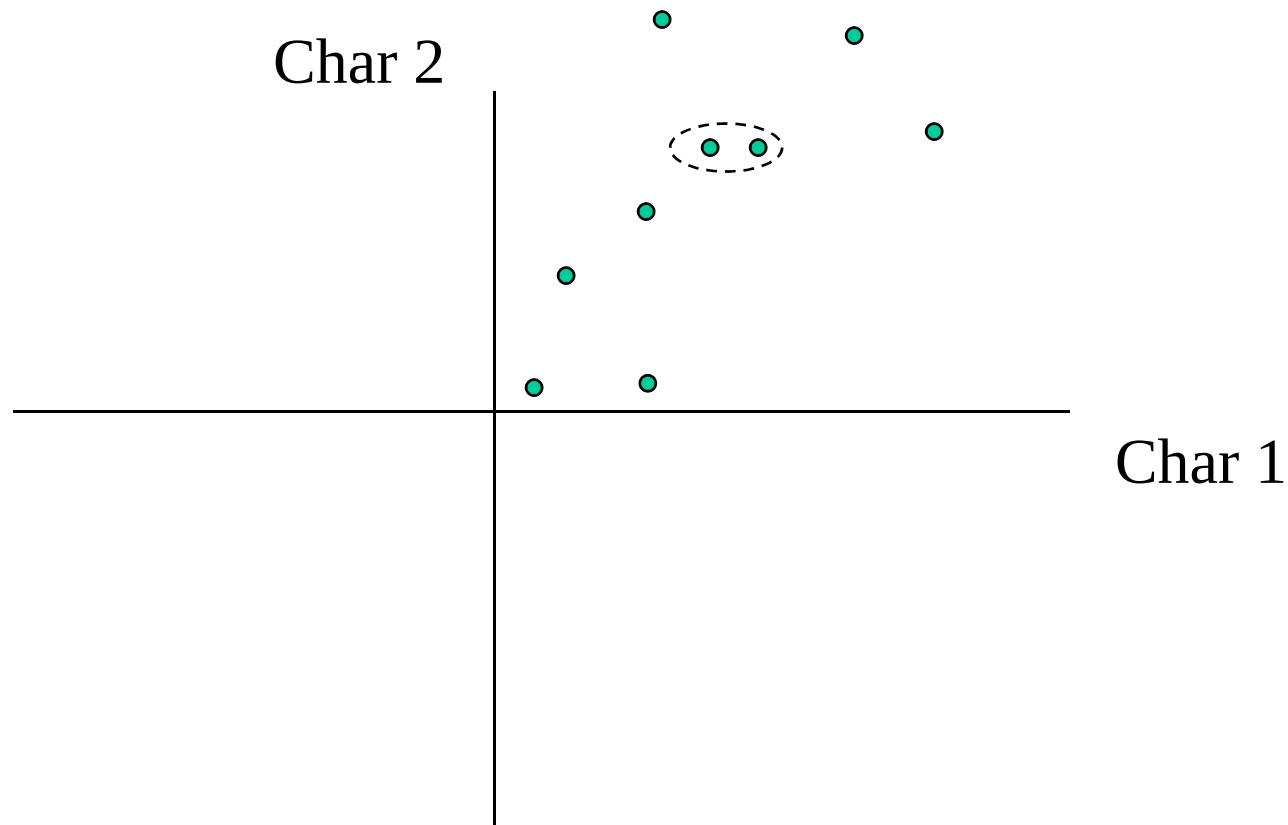
Dendrogram (tree)

Note: a dendrogram can be rotated along branch points (like mobile in baby room) -- distances between objects are defined along branches

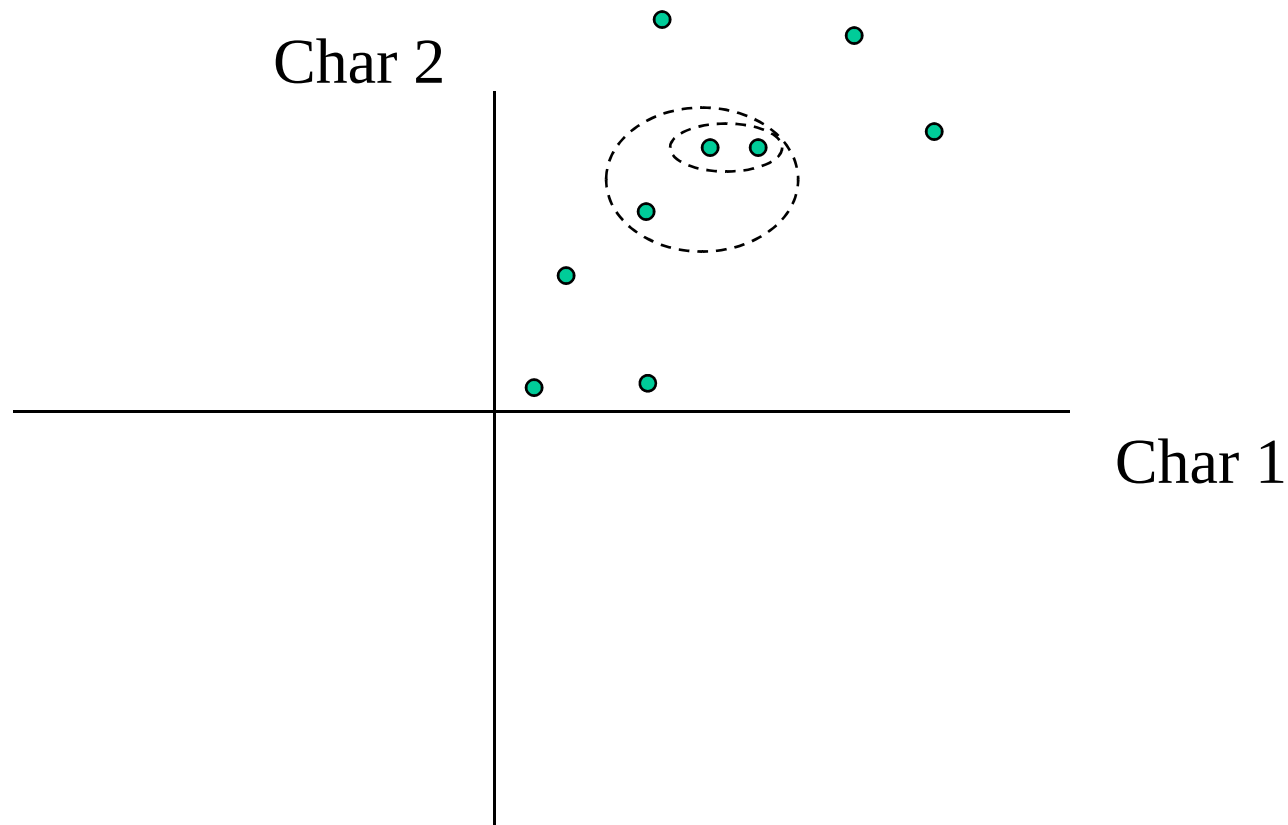
Single linkage clustering (nearest neighbour)



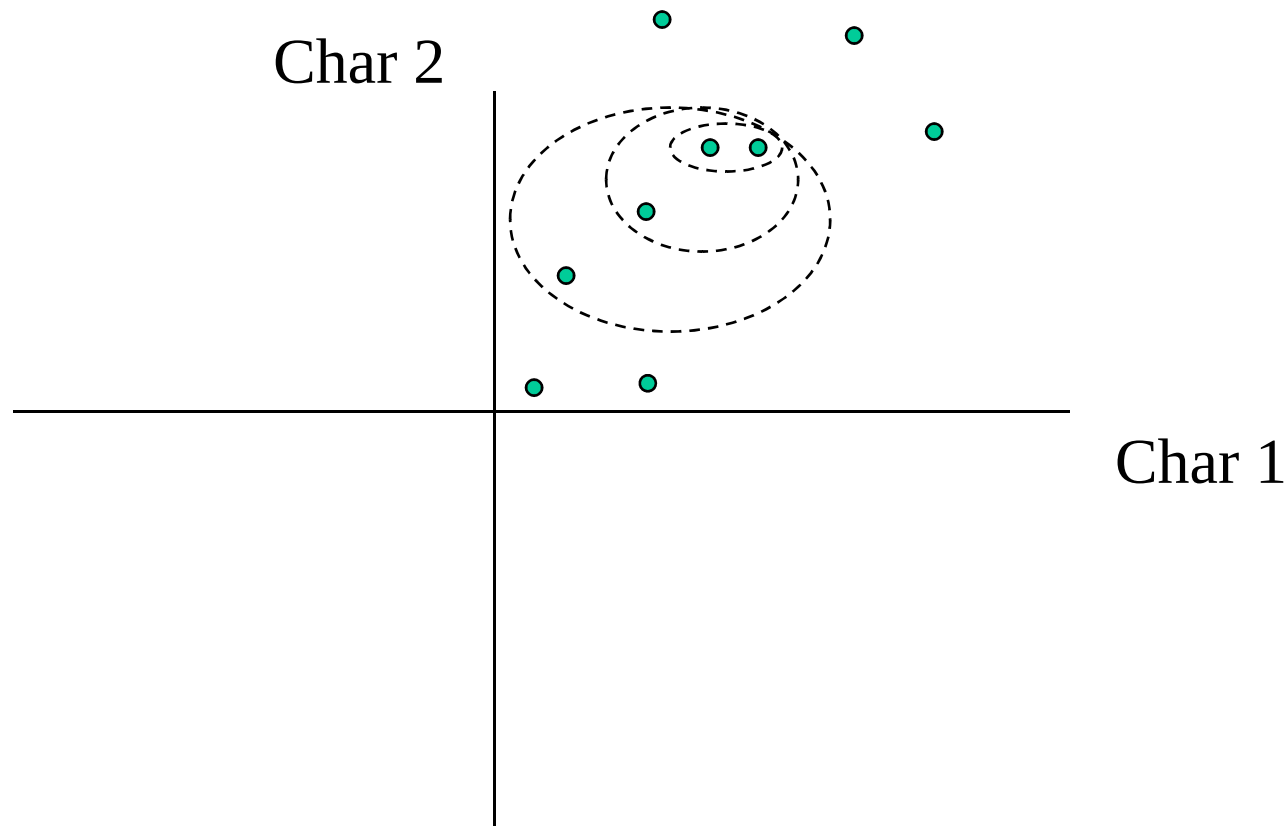
Single linkage clustering (nearest neighbour)



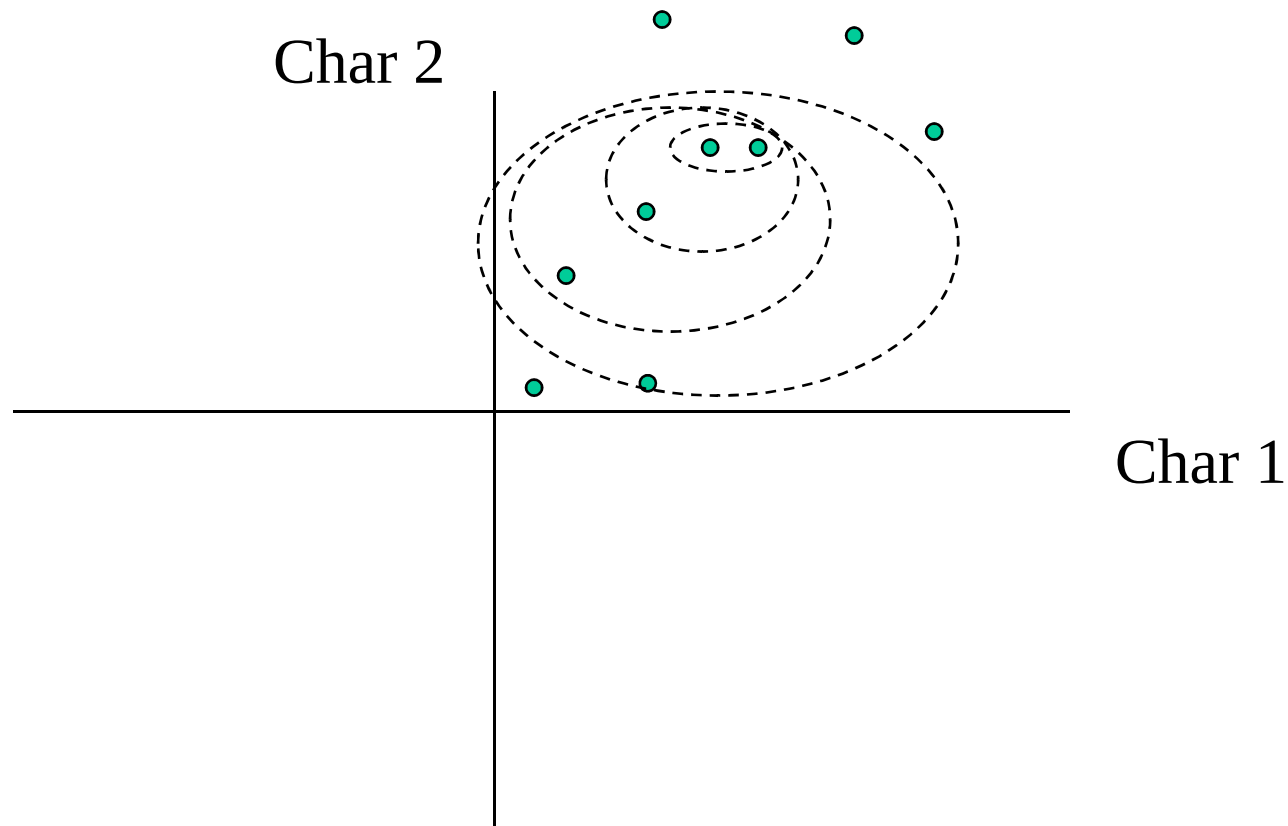
Single linkage clustering (nearest neighbour)



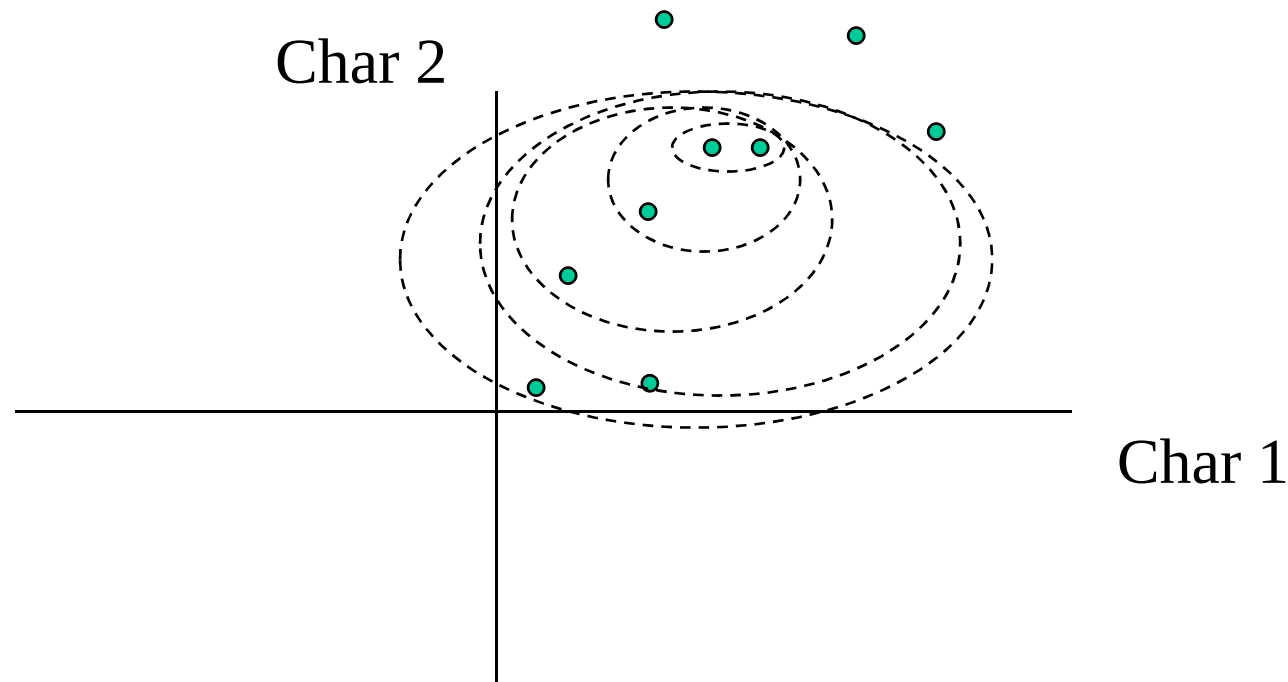
Single linkage clustering (nearest neighbour)



Single linkage clustering (nearest neighbour)

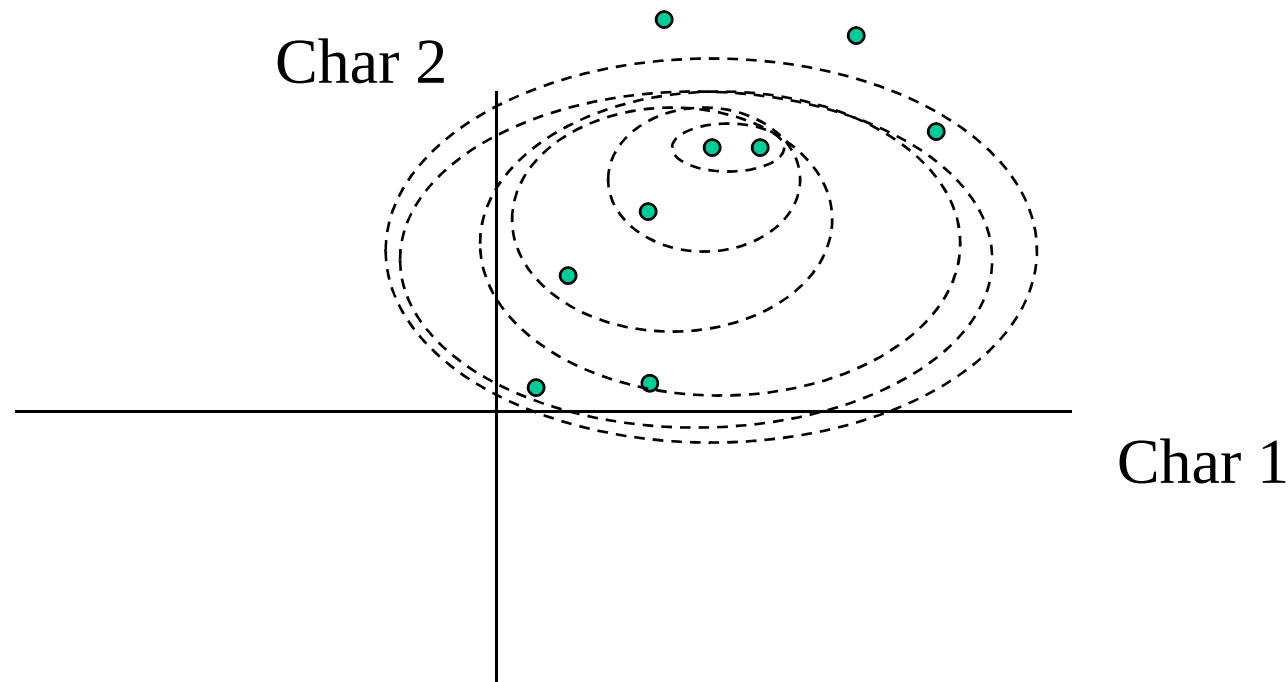


Single linkage clustering (nearest neighbour)



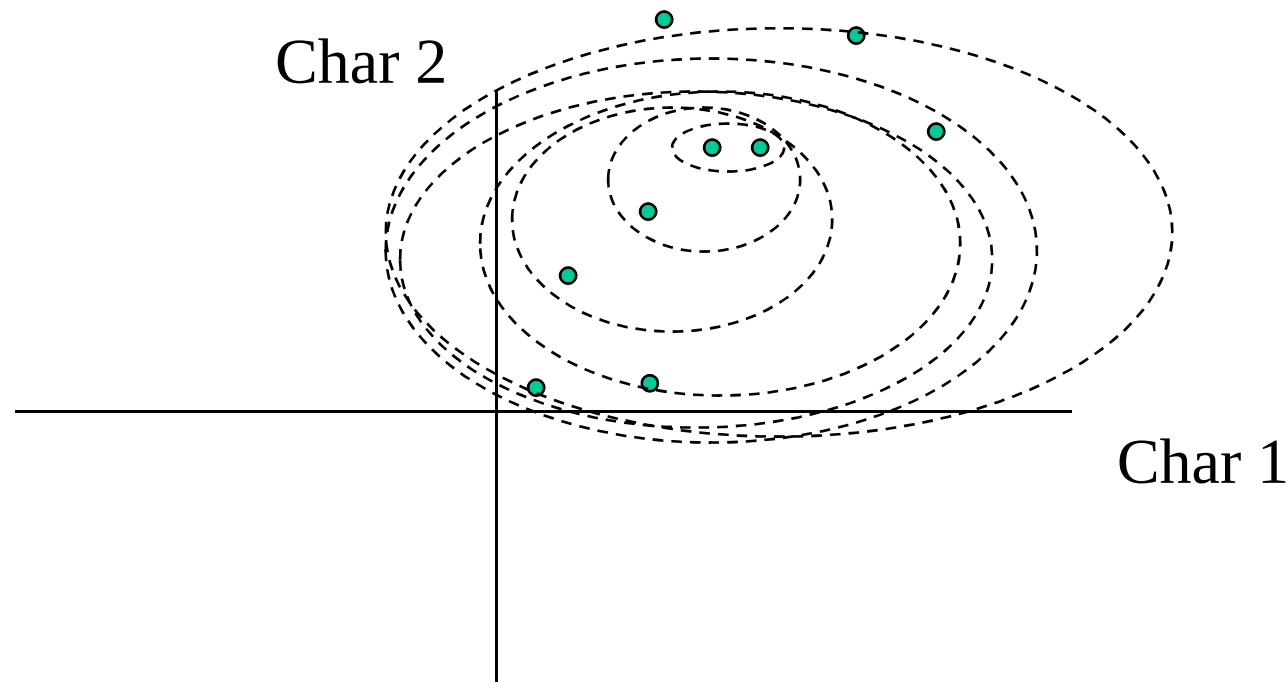
Distance from point to cluster is defined as the smallest distance between that point and any point in the cluster

Single linkage clustering (nearest neighbour)



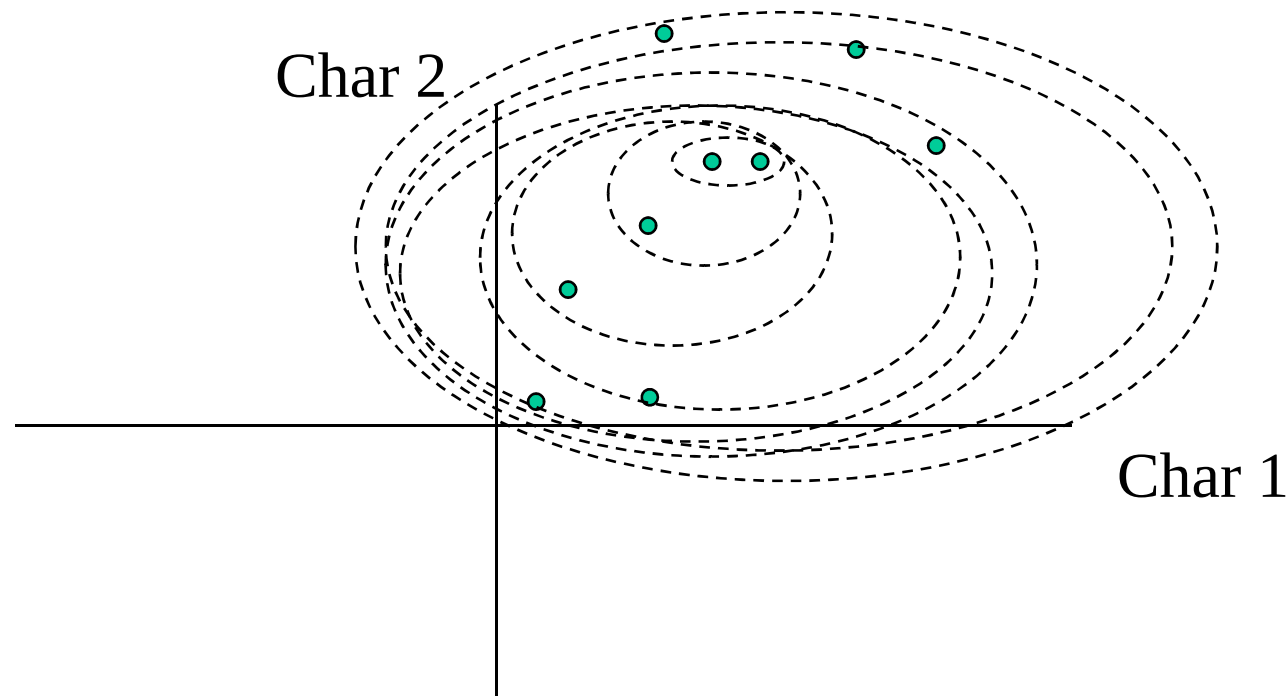
Distance from point to cluster is defined as the smallest distance between that point and any point in the cluster

Single linkage clustering (nearest neighbour)



Distance from point to cluster is defined as the smallest distance between that point and any point in the cluster

Single linkage clustering (nearest neighbour)

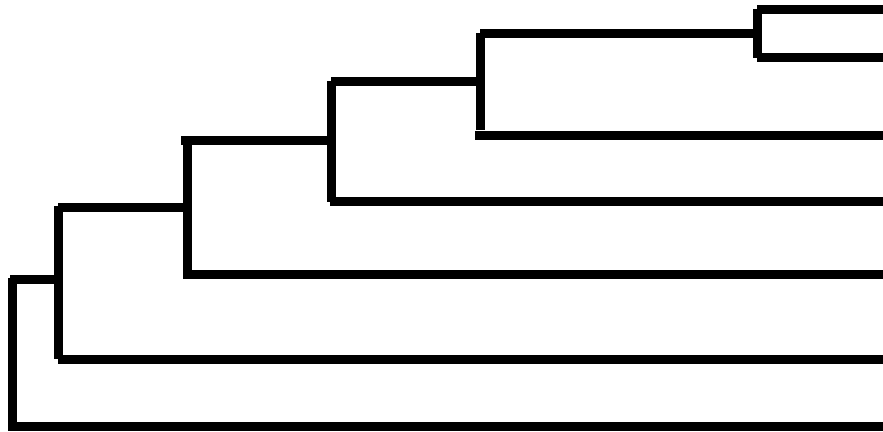


Distance from point to cluster is defined as the smallest distance between that point and any point in the cluster

Single linkage clustering (nearest neighbour)

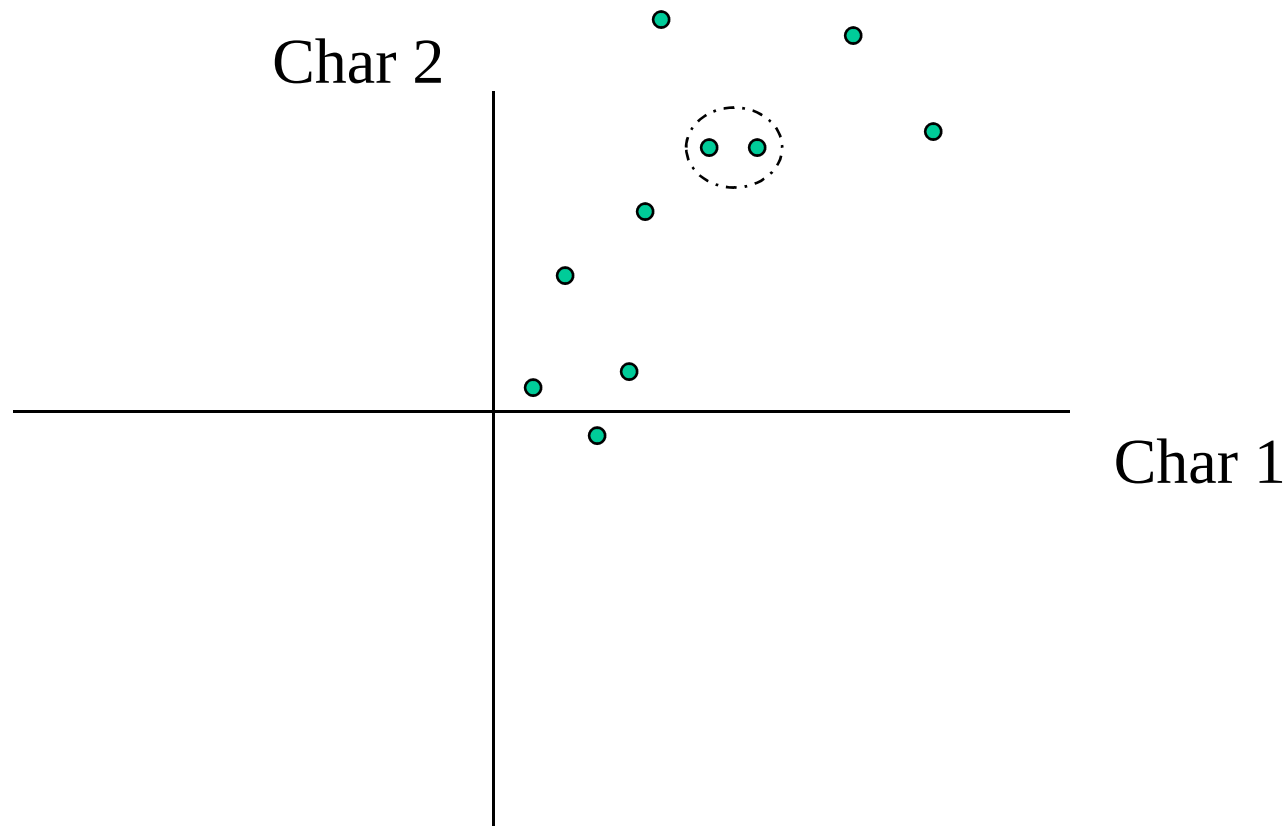
Let C_i and C_j be two disjoint clusters:

$$d_{i,j} = \text{Min}(d_{p,q}), \text{ where } p \in C_i \text{ and } q \in C_j$$

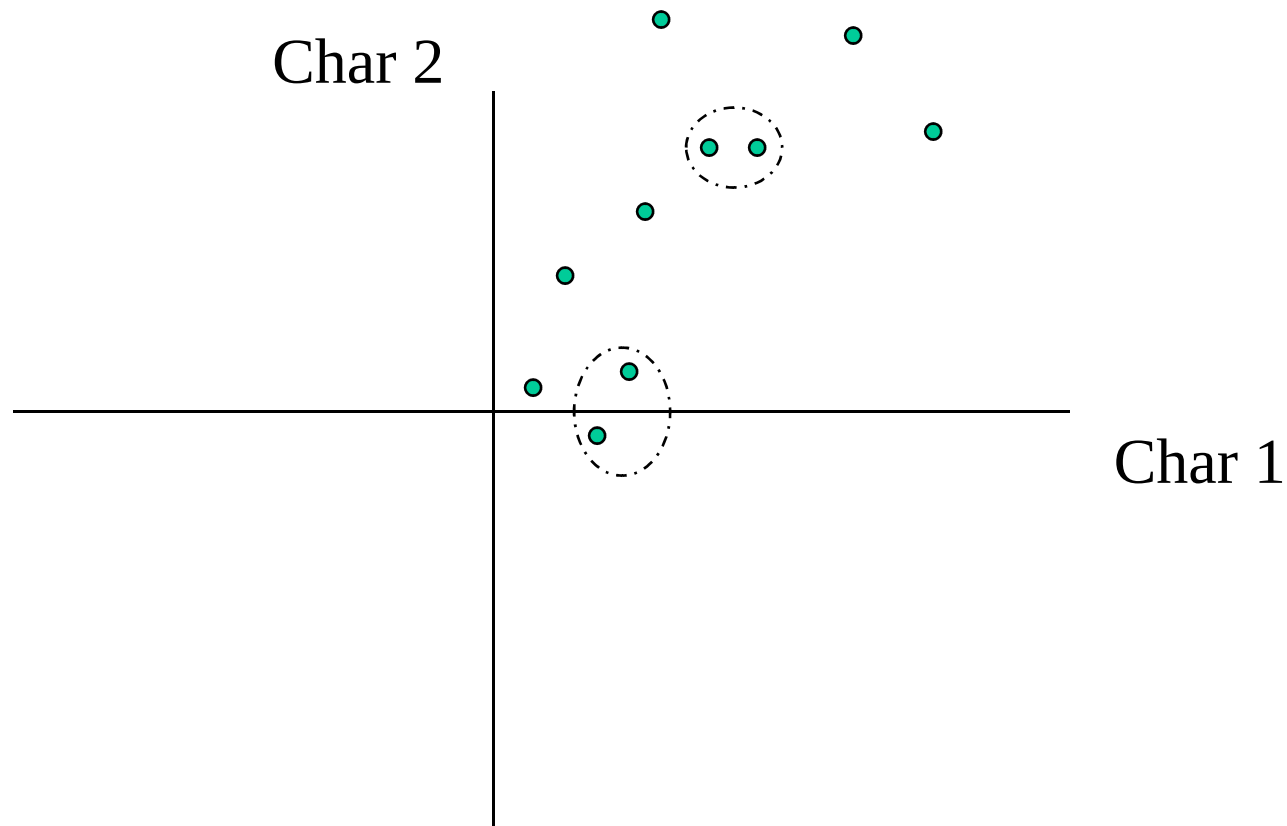


Single linkage dendrograms typically show **chaining** behaviour (i.e., all the time a single object is added to existing cluster)

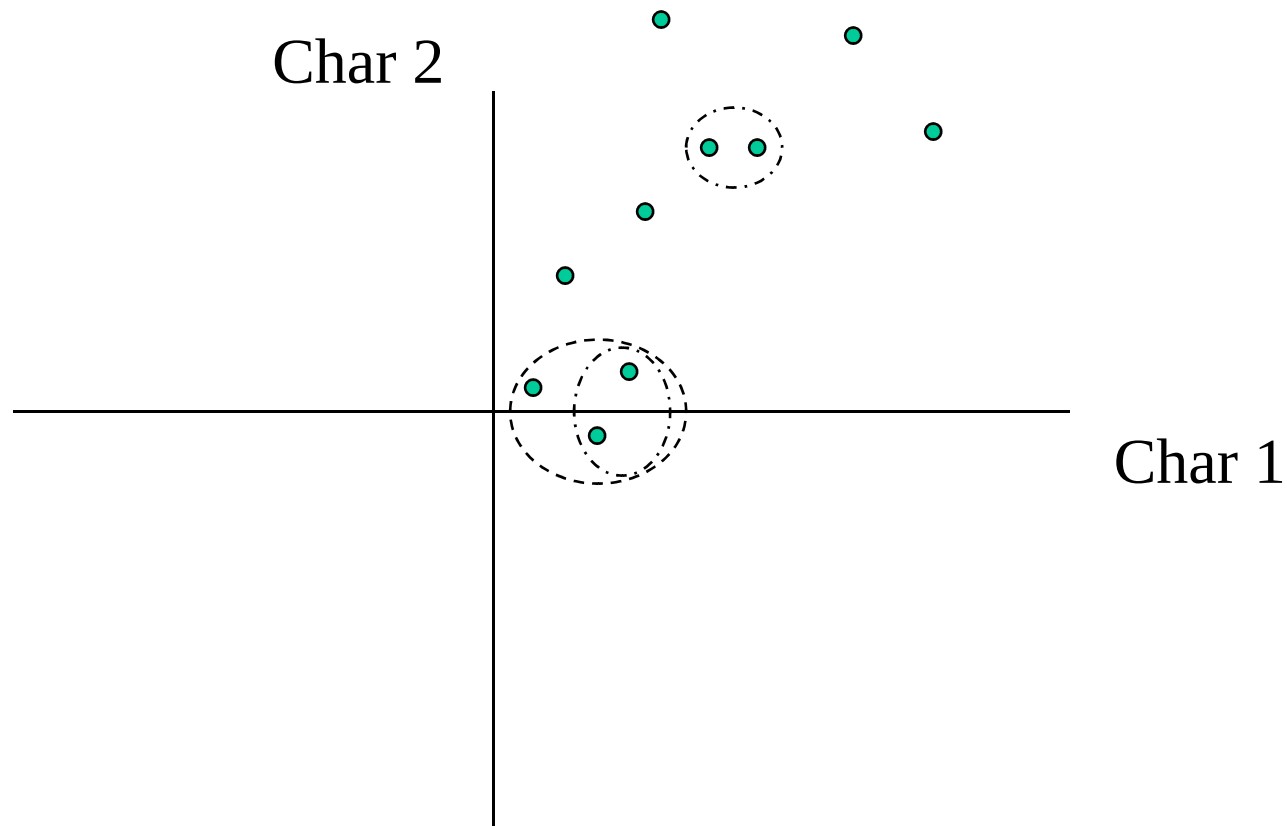
Complete linkage clustering (furthest neighbour)



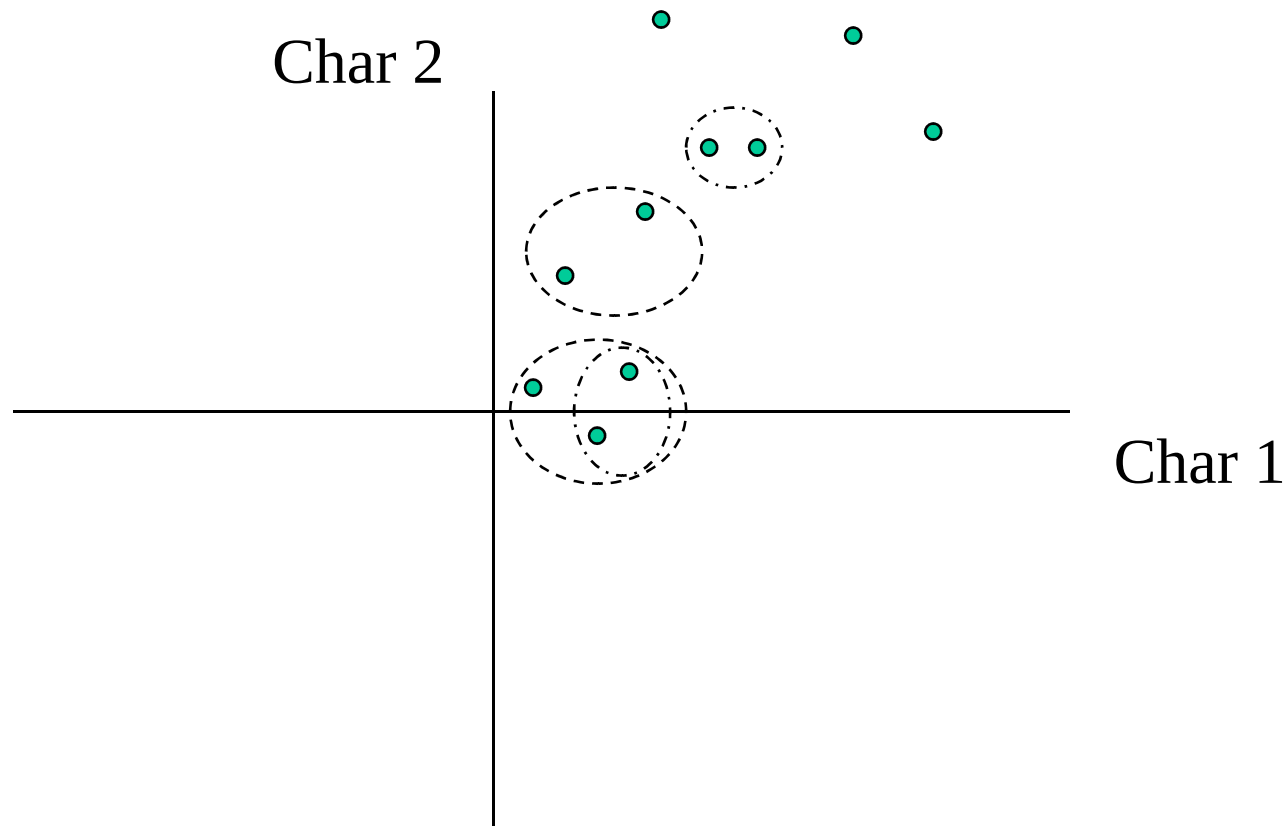
Complete linkage clustering (furthest neighbour)



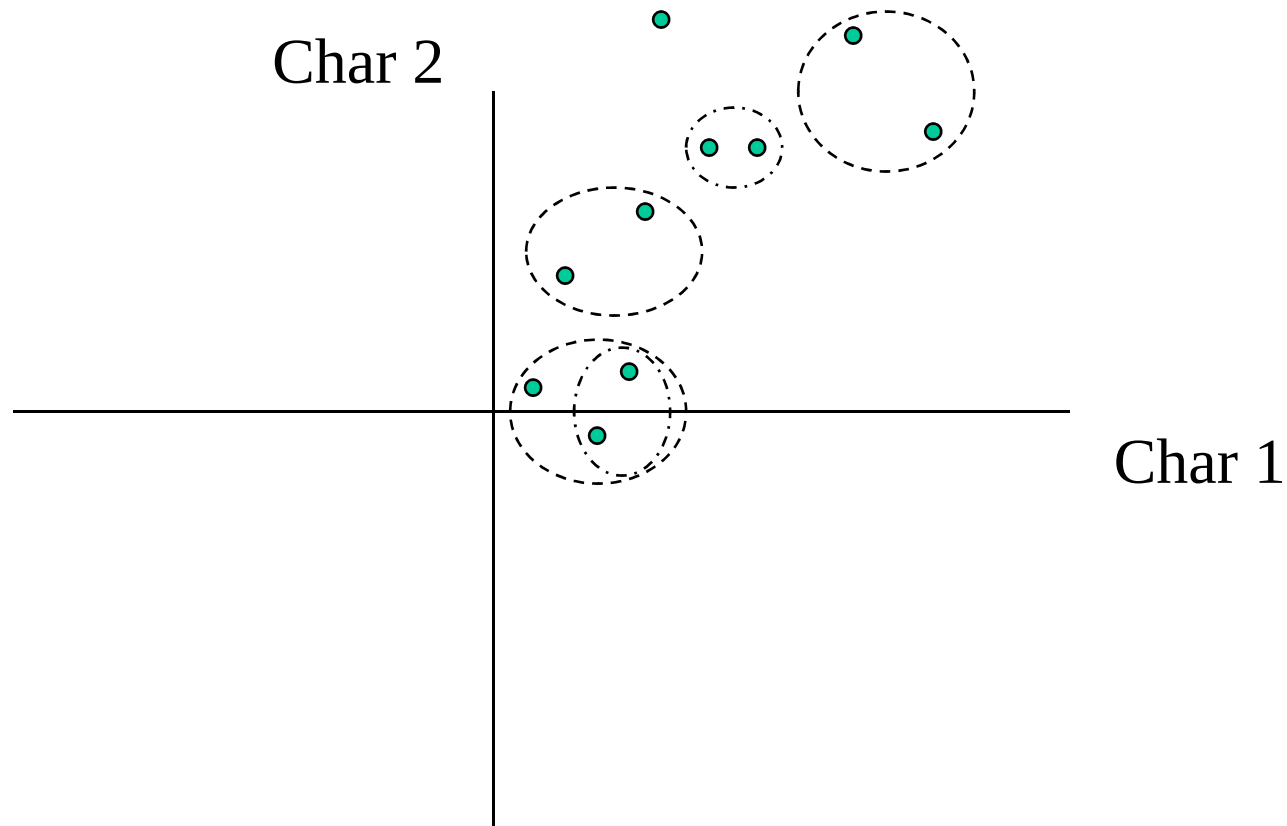
Complete linkage clustering (furthest neighbour)



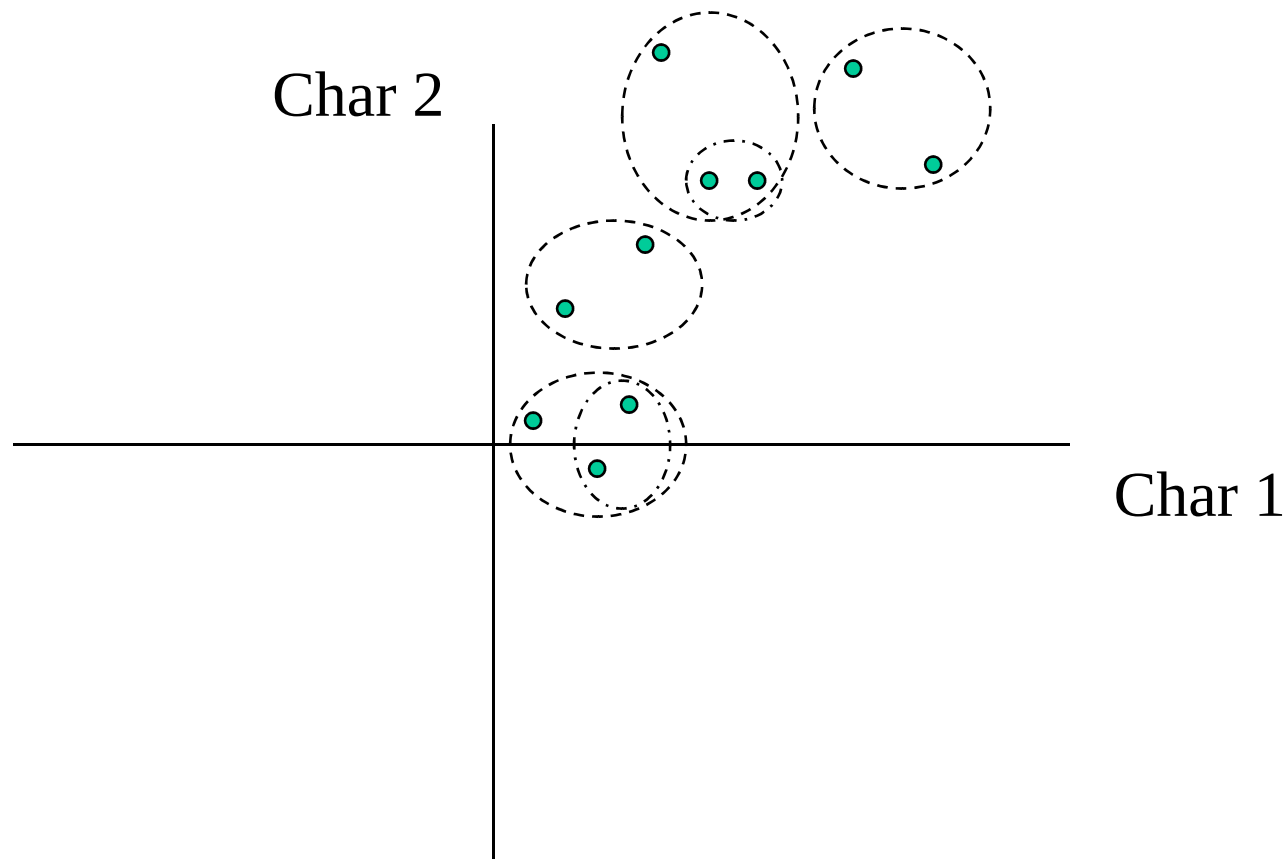
Complete linkage clustering (furthest neighbour)



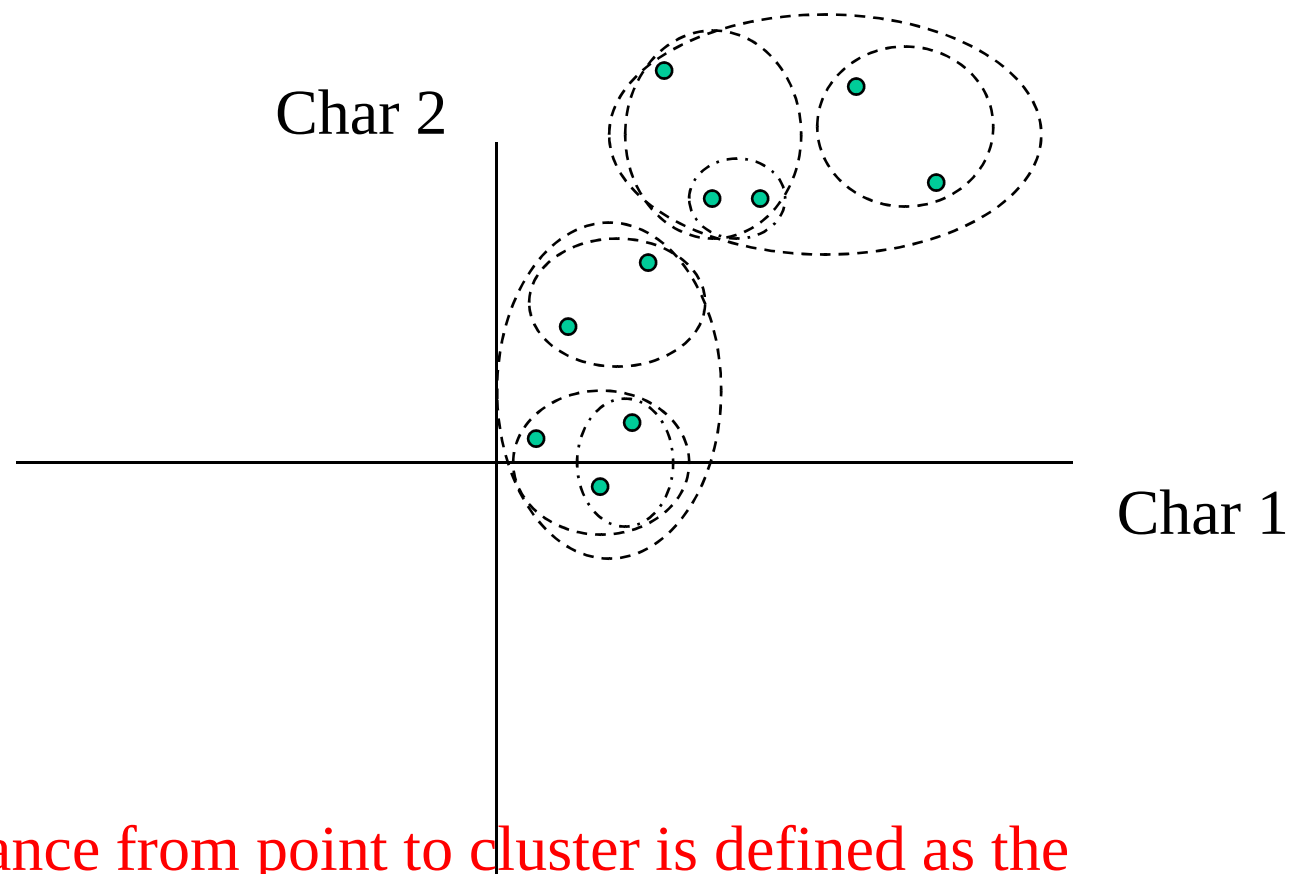
Complete linkage clustering (furthest neighbour)



Complete linkage clustering (furthest neighbour)

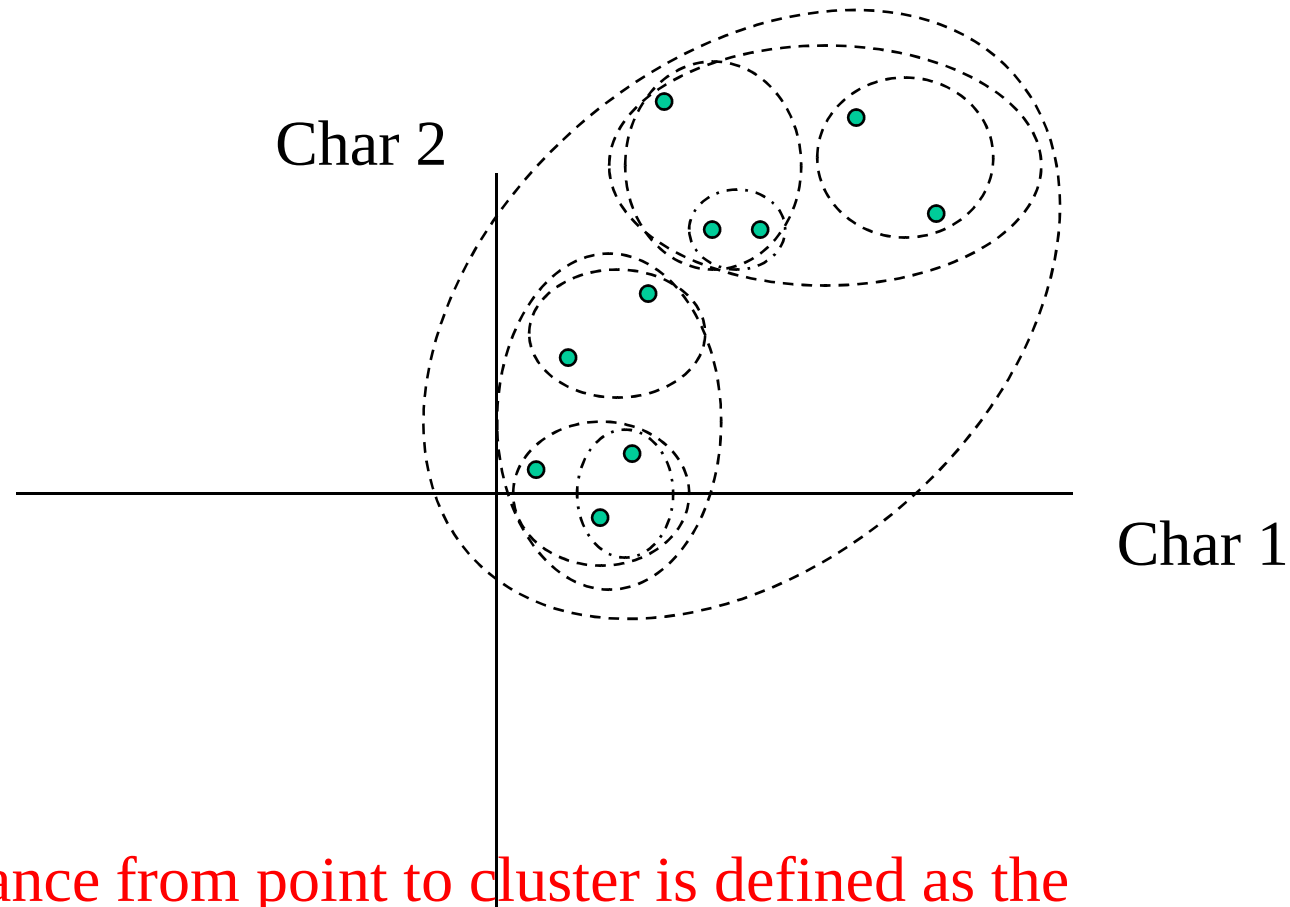


Complete linkage clustering (furthest neighbour)



Distance from point to cluster is defined as the largest distance between that point and any point in the cluster

Complete linkage clustering (furthest neighbour)

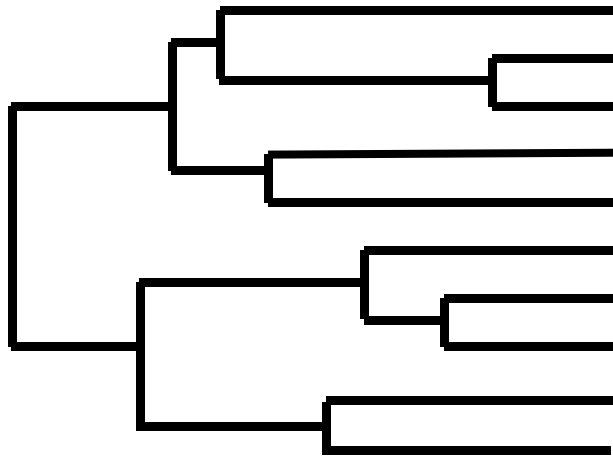


Distance from point to cluster is defined as the largest distance between that point and any point in the cluster

Complete linkage clustering (furthest neighbour)

Let C_i and C_j be two disjoint clusters:

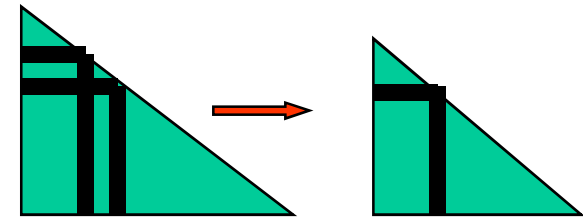
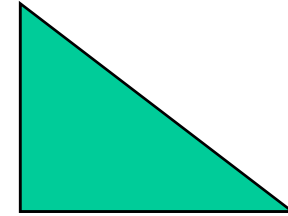
$$d_{i,j} = \text{Max}(d_{p,q}), \text{ where } p \in C_i \text{ and } q \in C_j$$



More 'structured' clusters than with
single linkage clustering

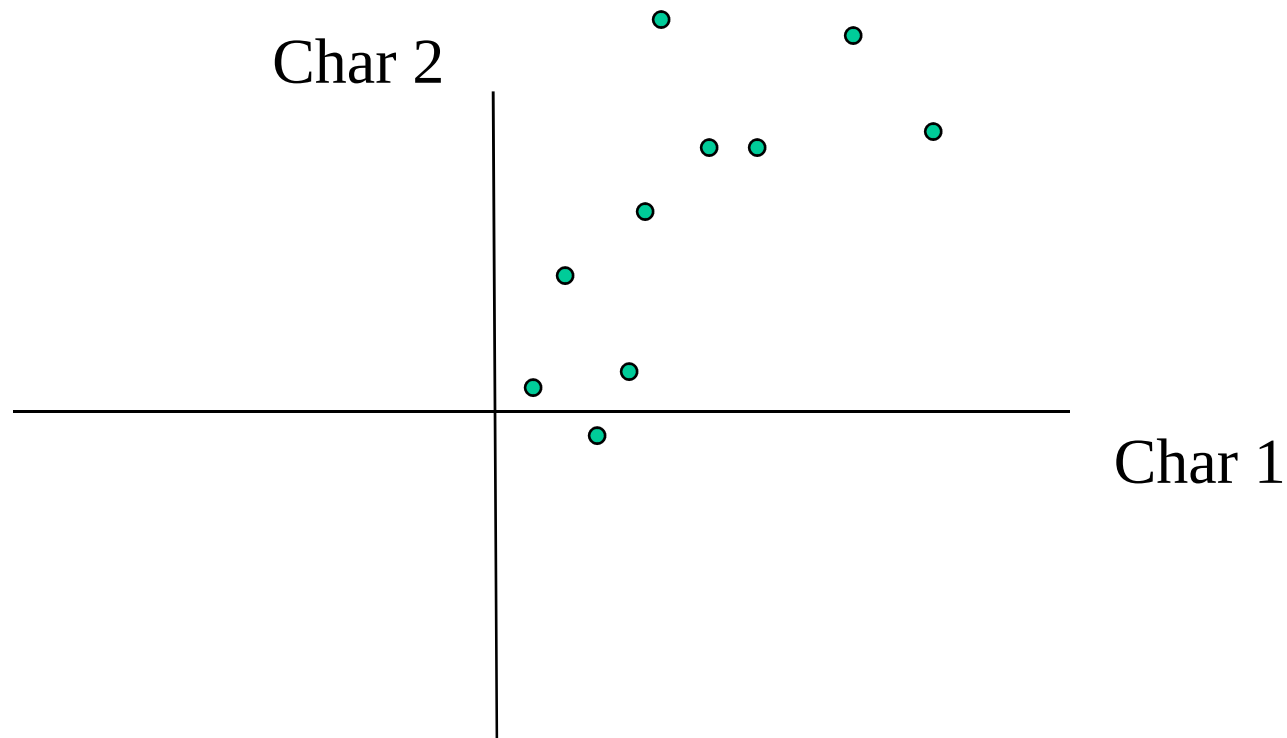
Clustering algorithm

1. Initialise (dis)similarity matrix
2. Take two points with smallest distance as first cluster (later, points can be clusters)
3. Merge corresponding rows/columns in (dis)similarity matrix
4. Repeat steps 2. and 3.
using appropriate cluster
measure when you need to calculate new
point-to-cluster or cluster-to-cluster
distances until last two clusters are
merged



Average linkage clustering

(Unweighted Pair Group Mean Averaging -UPGMA)

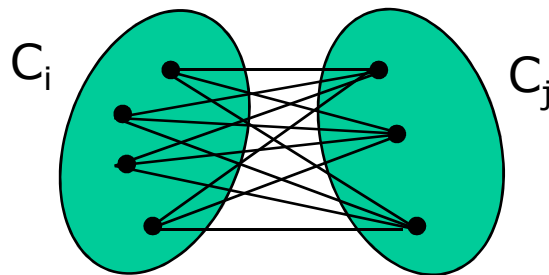


Distance from cluster to cluster is defined as the average distance over all within-cluster distances

UPGMA

Let C_i and C_j be two disjoint clusters:

$$d_{i,j} = \frac{1}{|C_i| \times |C_j|} \sum_p \sum_q d_{p,q}, \text{ where } p \in C_i \text{ and } q \in C_j$$

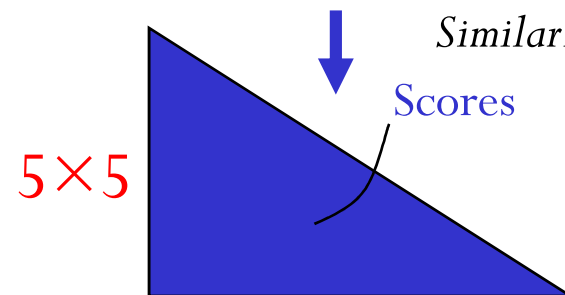


In words: calculate the average over all pairwise inter-cluster distances

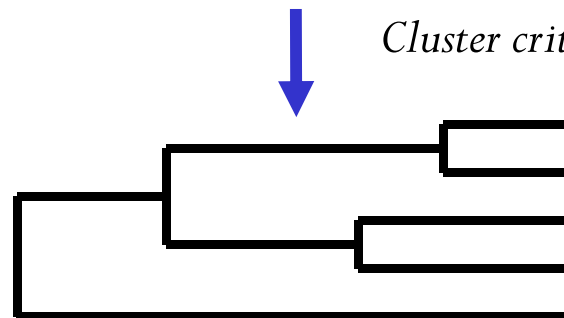
Multivariate statistics – Cluster analysis

	C1	C2	C3	C4	C5	C6	..
1							
2							
3							
4							
5							

Data table

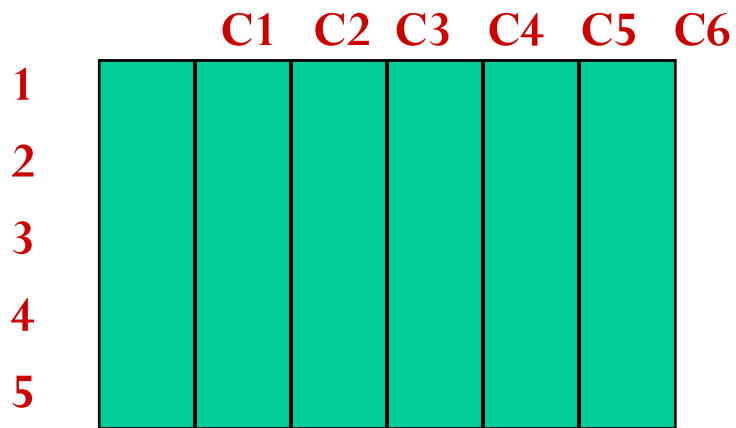


Similarity
matrix

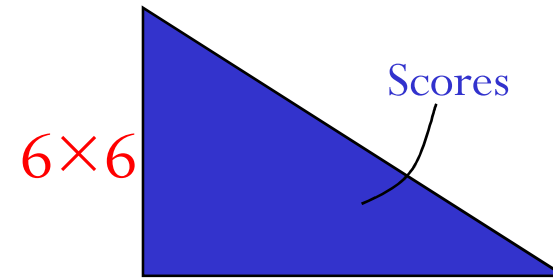


Phylogenetic tree

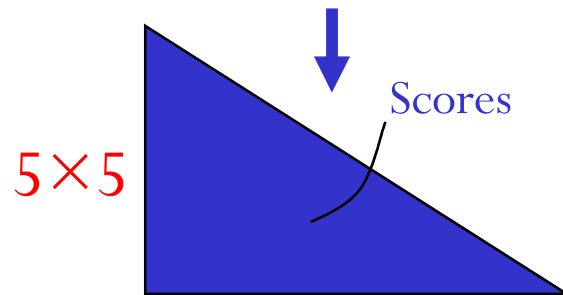
Multivariate statistics – Cluster analysis



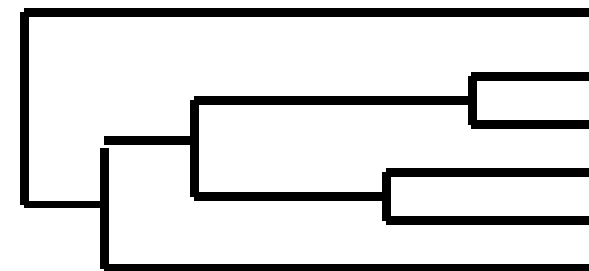
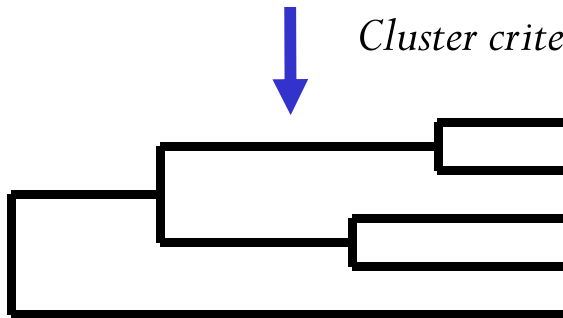
*Similarity
criterion*



Cluster criterion

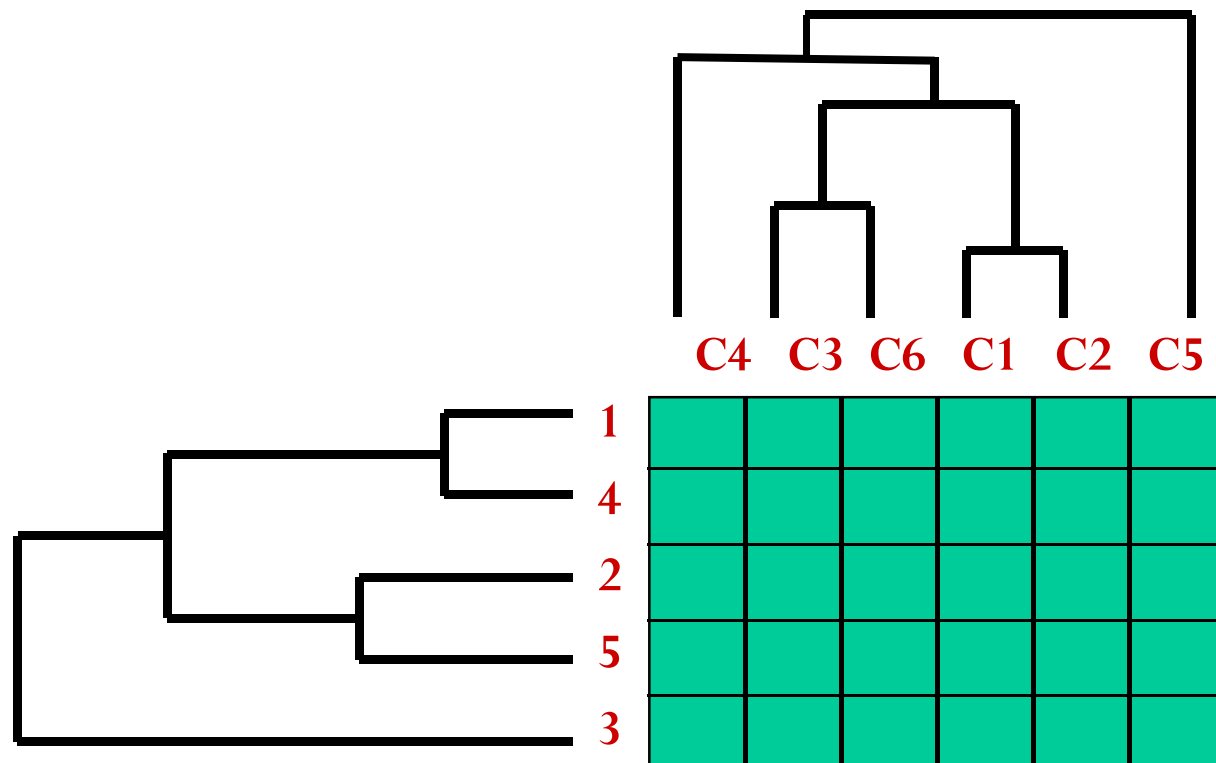


Cluster criterion



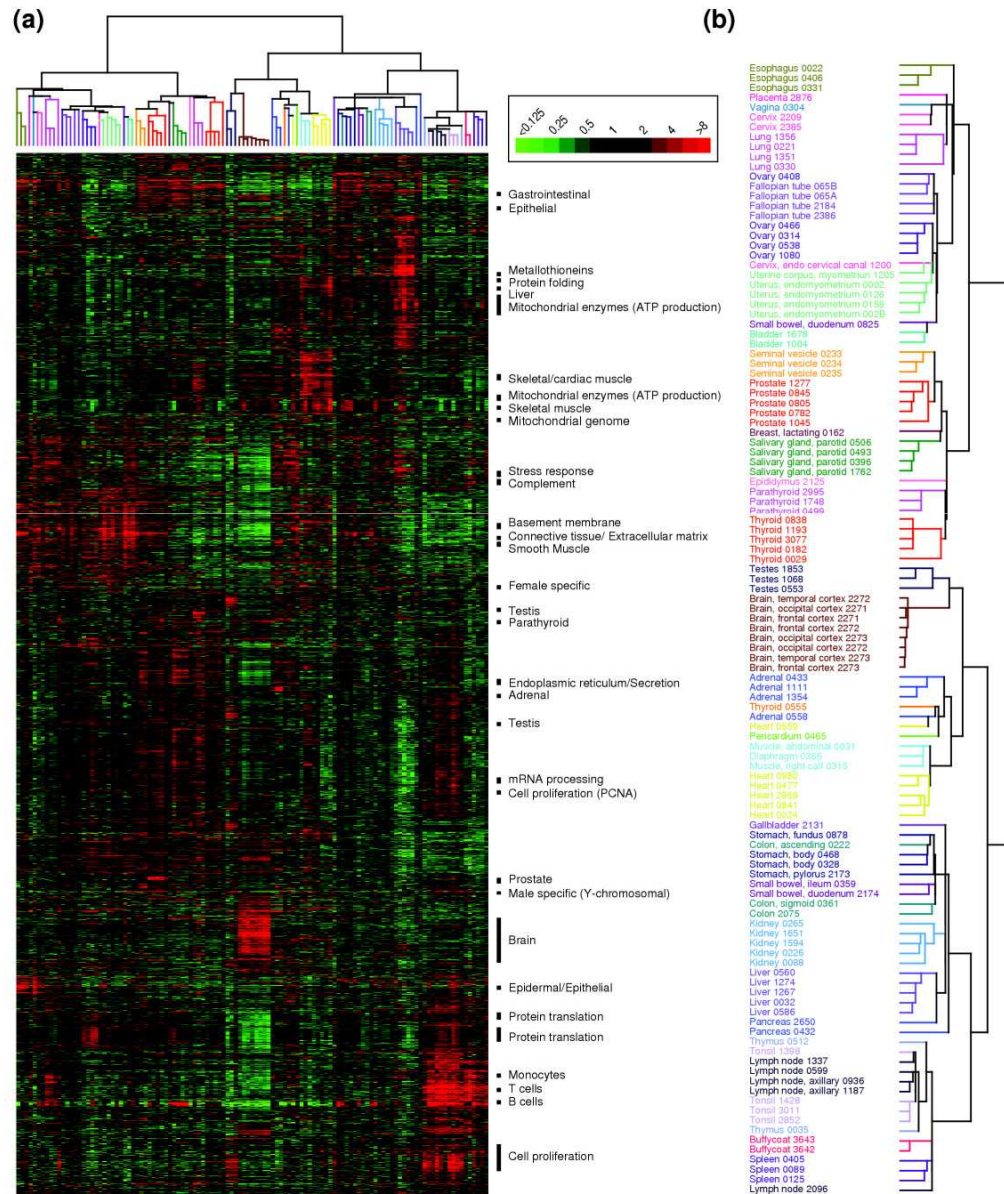
*Make two-way ordered
table using dendrograms*

Multivariate statistics – Two-way cluster analysis

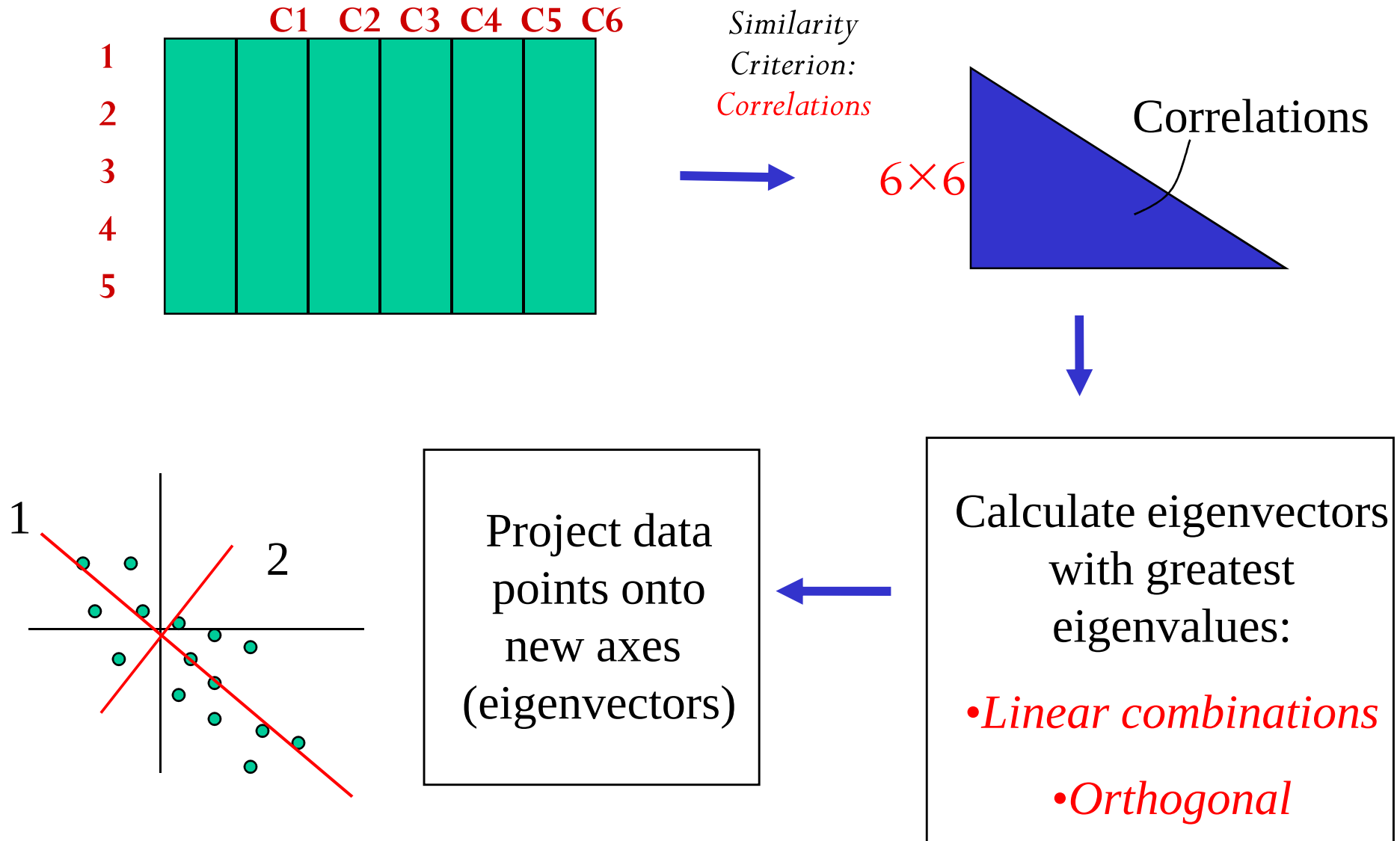


*Make two-way (rows, columns) ordered table using dendrograms;
This shows 'blocks' of numbers that are similar*

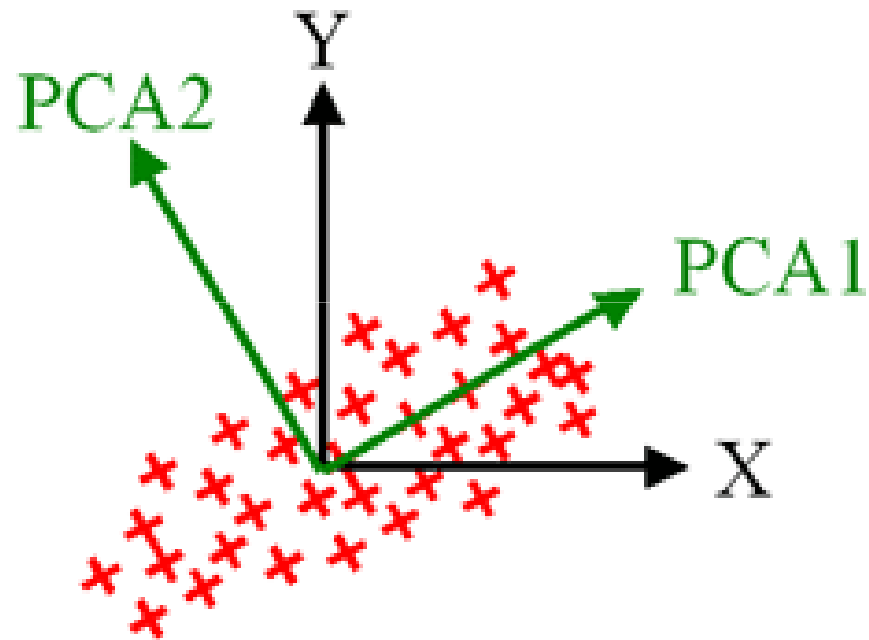
Multivariate statistics – Two-way cluster analysis



Multivariate statistics – Principal Component Analysis (PCA)



Multivariate statistics – Principal Component Analysis (PCA)



Evolution/Phylogeny methods

Centre for Integrative Bioinformatics VU
vrije Universiteit amsterdam

Bioinformatics

“Nothing in Biology makes sense except in the light of evolution” (Theodosius Dobzhansky (1900-1975))



“Nothing in bioinformatics makes sense except in the light of Biology”

Evolution

- Most of bioinformatics is comparative biology
- Comparative biology is based upon evolutionary relationships between compared entities
- Evolutionary relationships are normally depicted in a *phylogenetic tree*

Where can phylogeny be used

- For example, finding out about orthology *versus* paralogy
- Predicting secondary structure of RNA
- Predicting protein-protein interaction
- Studying host-parasite relationships
- Mapping cell-bound receptors onto their binding ligands
- Multiple sequence alignment (e.g. Clustal)

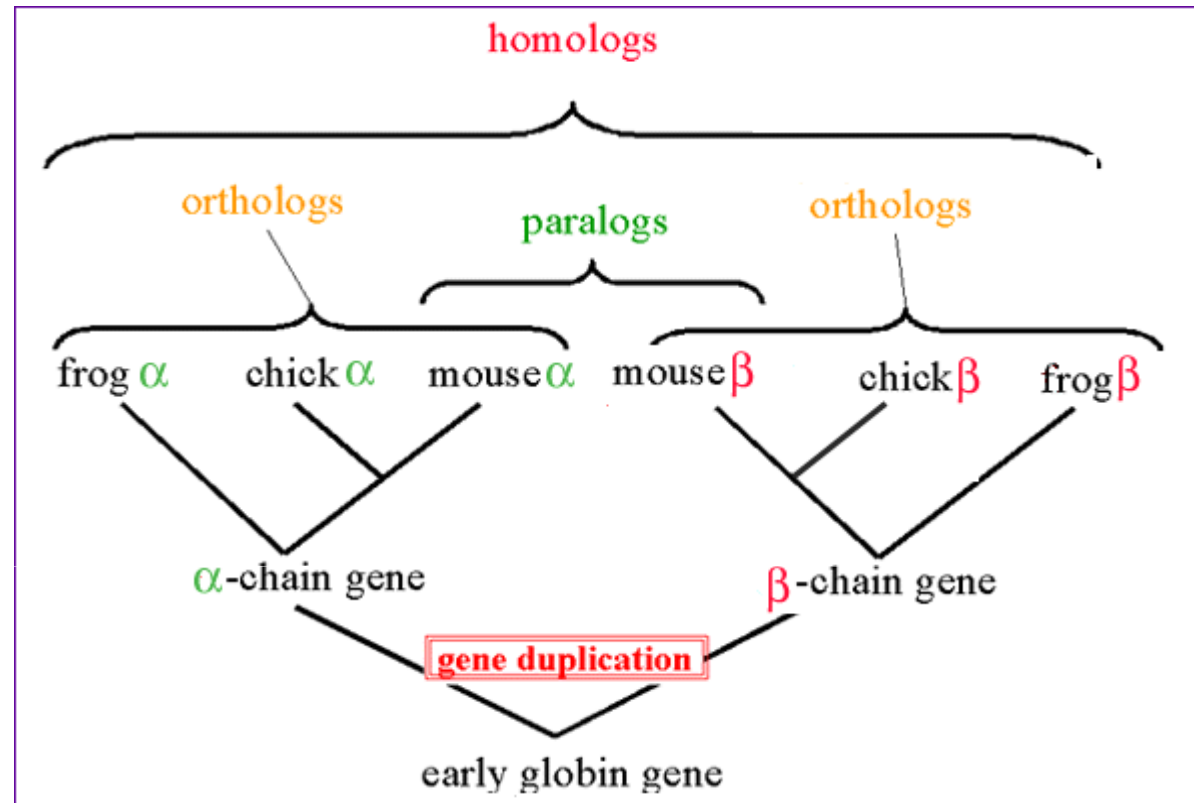
DNA evolution

- Gene nucleotide substitutions can be synonymous (i.e. not changing the encoded amino acid) or nonsynonymous (i.e. changing the a.a.).
- Rates of evolution vary tremendously among protein-coding genes. Molecular evolutionary studies have revealed an ~ 1000 -fold range of nonsynonymous substitution rates (Li and Graur 1991).
- The strength of negative (purifying) selection is thought to be the most important factor in determining the rate of evolution for the protein-coding regions of a gene (Kimura 1983; Ohta 1992; Li 1997).

DNA evolution

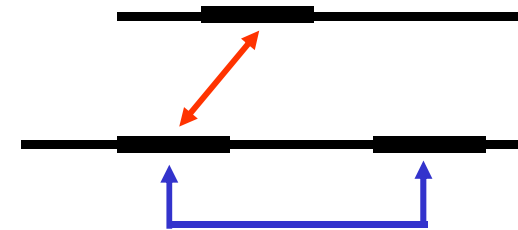
- “Essential” and “nonessential” are classic molecular genetic designations relating to organismal fitness.
 - A gene is considered to be essential if a knock-out results in (conditional) lethality or infertility.
 - Nonessential genes are those for which knock-outs yield viable and fertile individuals.
- Given the role of purifying selection in determining evolutionary rates, the greater levels of purifying selection on essential genes leads to a lower rate of evolution relative to that of nonessential genes
- This leads to the observation: “What is important is conserved”.

Reminder -- Orthology/paralogy



Orthologous genes are homologous (corresponding) genes in different species

Paralogous genes are homologous genes within the same species (genome)



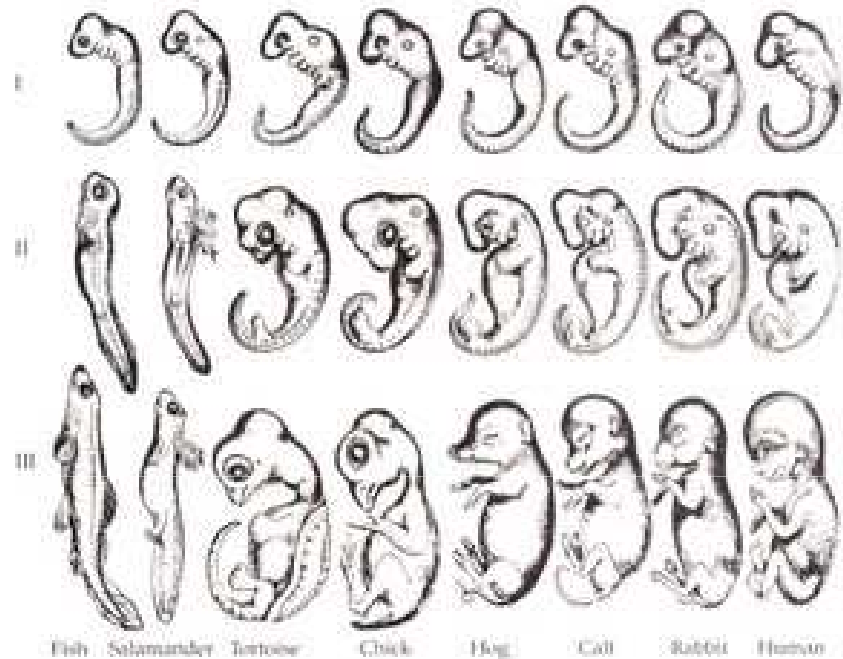
Old Dogma – Recapitulation Theory (1866)

Ernst Haeckel:

“Ontogeny recapitulates
phylogeny”

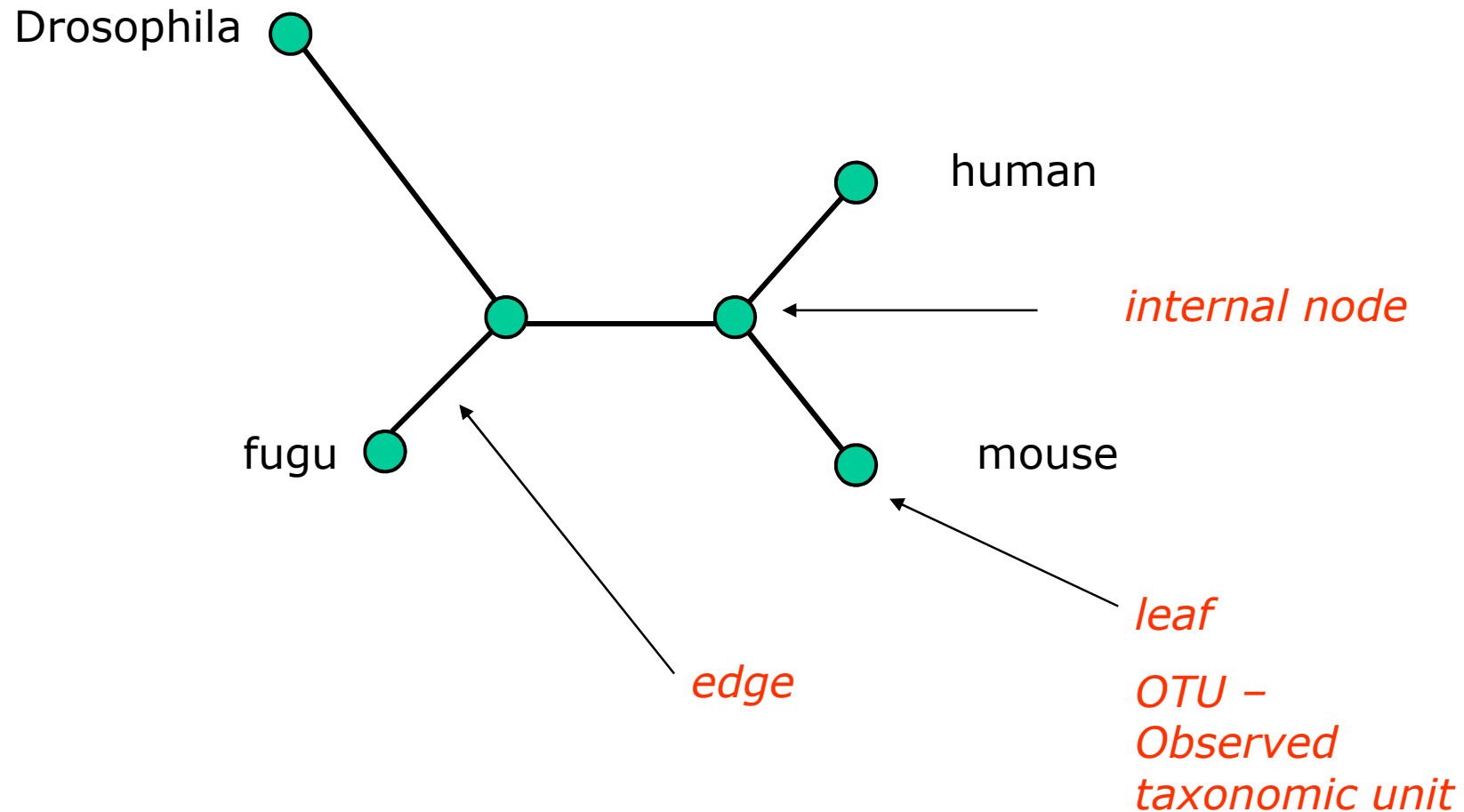
- Ontogeny is the development of the embryo of a given species;
- phylogeny is the evolutionary history of a species

http://en.wikipedia.org/wiki/Recapitulation_theory

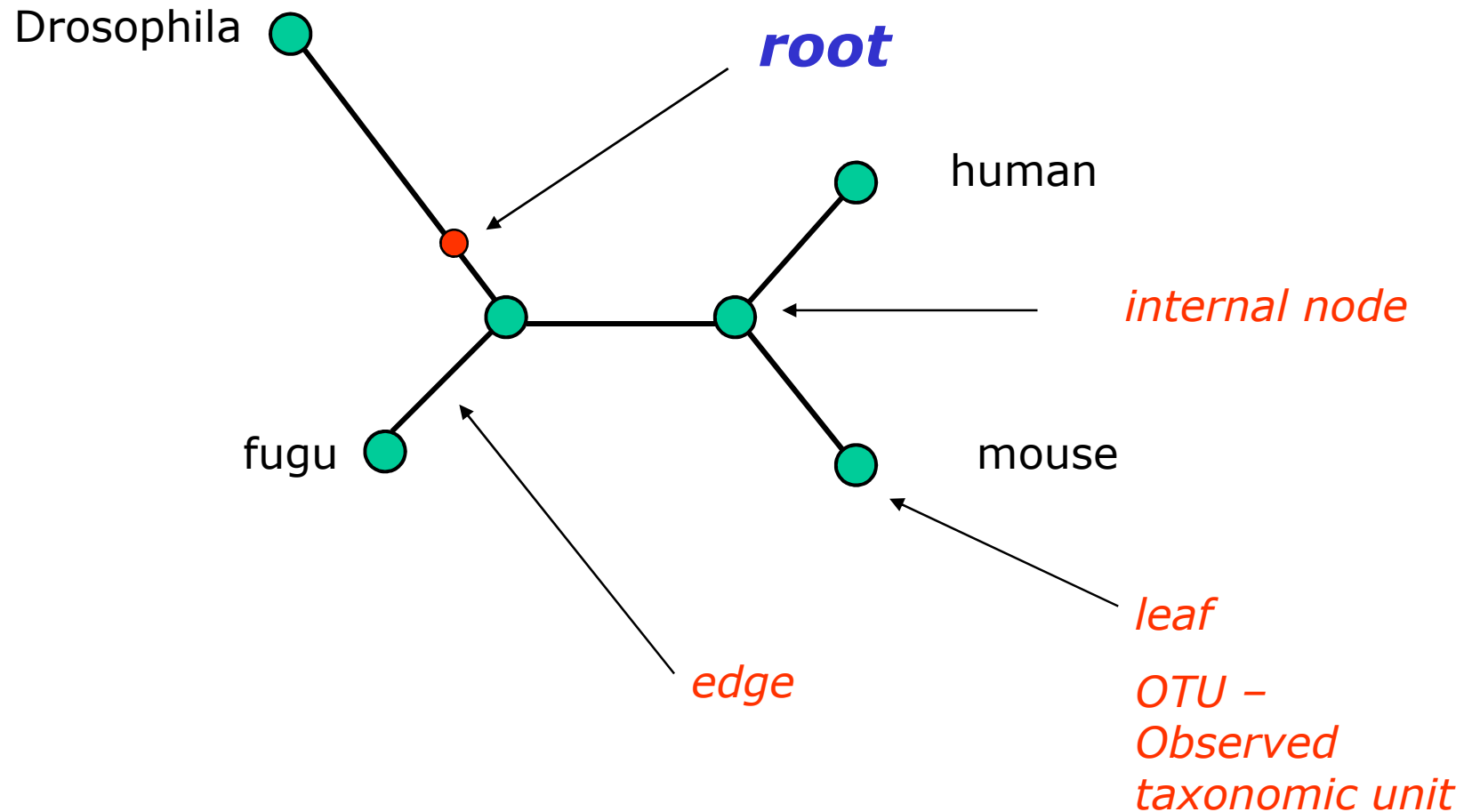


Haeckels drawing in support of his theory: For example, the human embryo with gill slits in the neck was believed by Haeckel to not only signify a fishlike ancestor, but it represented a total fishlike stage in development. However, gill slits are not the same as gills and are *not functional*.

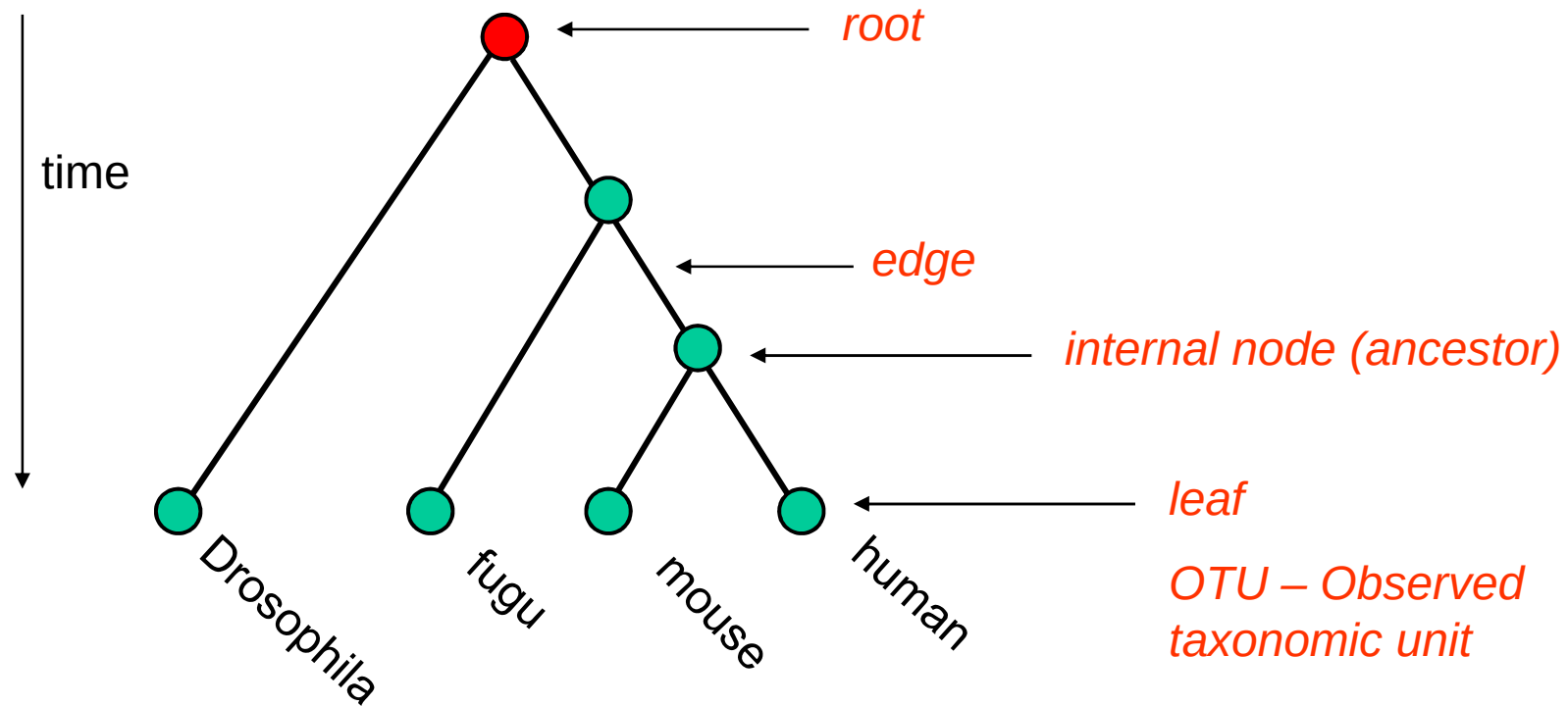
Phylogenetic tree (unrooted)



Phylogenetic tree (unrooted)

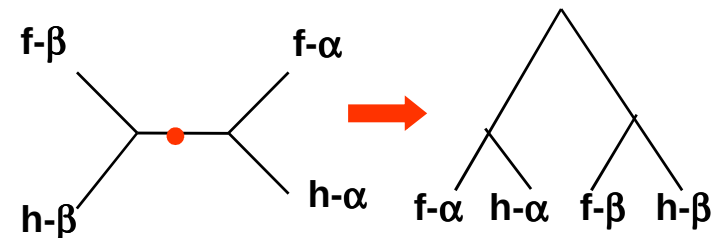
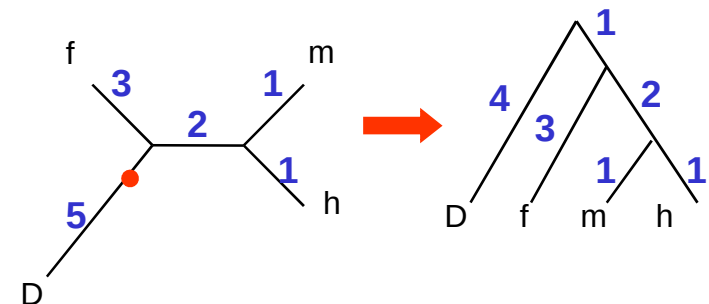
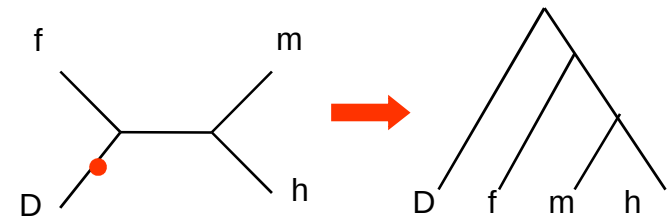


Phylogenetic tree (rooted)



How to root a tree

- Outgroup – place root between distant sequence and rest group
- Midpoint – place root at midpoint of longest path (sum of branches between any two OTUs)
- Gene duplication – place root between paralogous gene copies



Combinatoric explosion

$$\text{Number of unrooted trees} = \frac{(2n-5)!}{2^{n-3}(n-3)!}$$

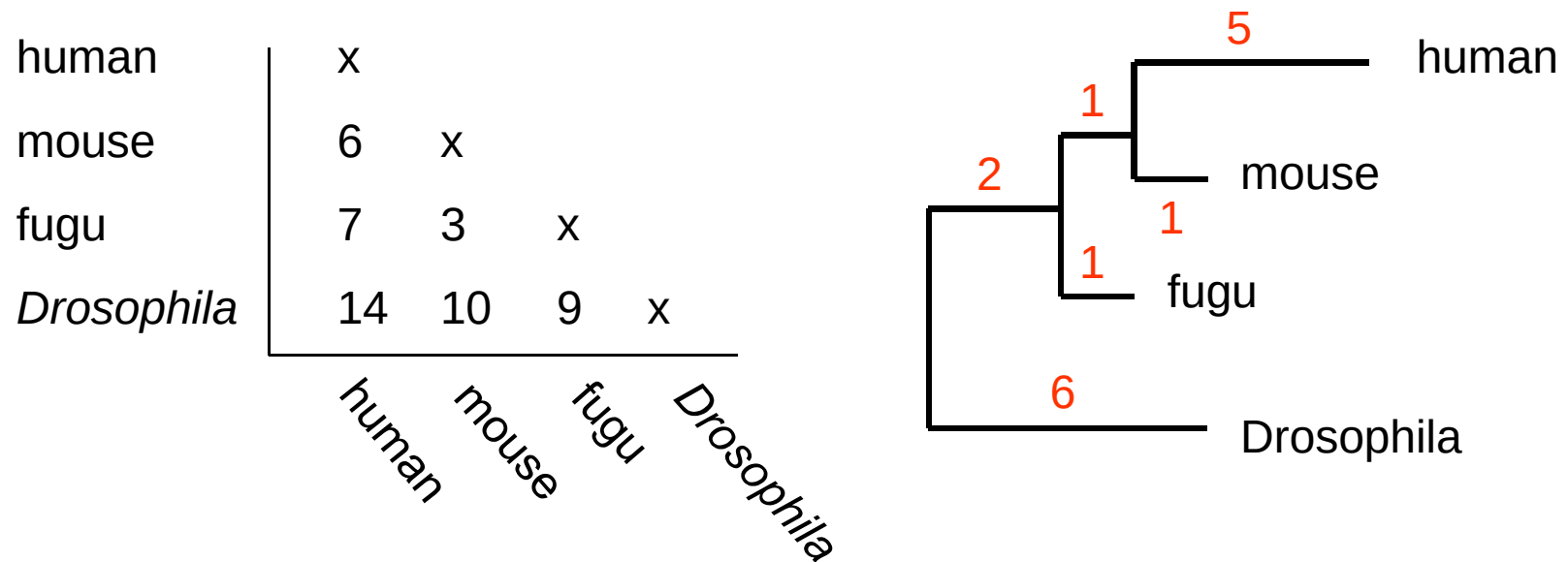
$$\text{Number of rooted trees} = \frac{(2n-3)!}{2^{n-2}(n-2)!}$$

Combinatoric explosion

# sequences	# unrooted trees	# rooted trees
2	1	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10,395
8	10,395	135,135
9	135,135	2,027,025
10	2,027,025	34,459,425

Tree distances

Evolutionary (sequence distance) = sequence dissimilarity



Note that with evolutionary methods for generating trees you get distances between objects by walking from one to the other.

Phylogeny methods

1. **Distance based** – pairwise distances (input is distance matrix)
2. **Parsimony** – fewest number of evolutionary events (mutations) – relatively often fails to reconstruct correct phylogeny, but methods have improved recently
3. **Maximum likelihood** – $L = \Pr[Data|Tree]$ – most flexible class of methods - user-specified evolutionary methods can be used

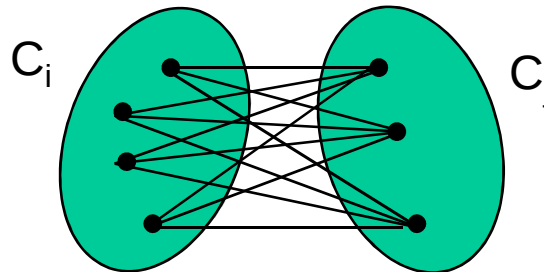
Similarity criterion for phylogeny

- A number of methods (e.g. ClustalW) use sequence identity with Kimura (1983) correction:
Corrected $K = -\ln(1.0 - K - K^2/5.0)$, where K is percentage divergence (expressed as sequence identity difference) corresponding to two aligned sequences (often only taking the gap-less alignment columns into account)
- There are various models to correct for the fact that the true rate of evolution cannot be observed through nucleotide (or amino acid) exchange patterns (e.g. back mutations)
- Saturation level is ~94% changed sequences, higher real mutations are no longer observable

Distance based --UPGMA

Let C_i and C_j be two disjoint clusters:

$$d_{i,j} = \frac{1}{|C_i| \times |C_j|} \sum_p \sum_q d_{p,q}, \text{ where } p \in C_i \text{ and } q \in C_j$$



In words: calculate the average over all pairwise inter-cluster distances

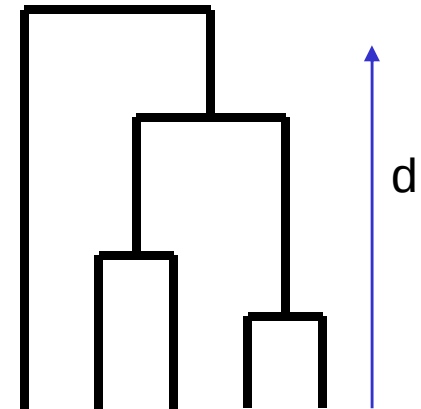
Clustering algorithm: UPGMA

Initialisation:

- Fill distance matrix with pairwise distances
- Start with N clusters of 1 element each

Iteration:

1. Merge cluster C_i and C_j for which d_{ij} is minimal
2. Place internal node connecting C_i and C_j at height $d_{ij}/2$
3. Delete C_i and C_j (keep internal node)



Termination:

- When two clusters i, j remain, place root of tree at height $d_{ij}/2$

Ultrametric Distances

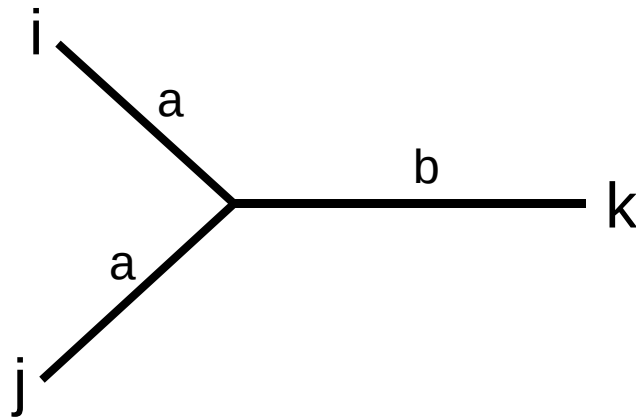
- A tree T in a metric space (M, d) where d is **ultrametric** has the following property: there is a way to place a root on T so that for all nodes in M , their distance to the root is the same. Such T is referred to as a **uniform molecular clock tree**.
- (M, d) is ultrametric if for every set of three elements $i, j, k \in M$, two of the distances coincide and are greater than or equal to the third one (see next slide).
- UPGMA is guaranteed to build correct tree if distances are ultrametric. But it fails if not!

Ultrametric Distances

Given three leaves, two distances are equal while a third is smaller:

$$d(i,j) \leq d(i,k) = d(j,k)$$

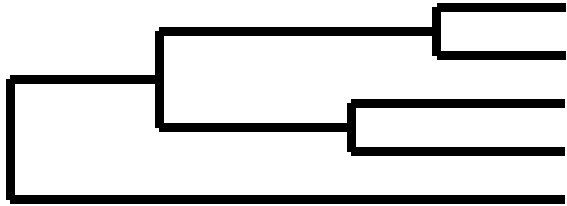
$$a+a \leq a+b = a+b$$



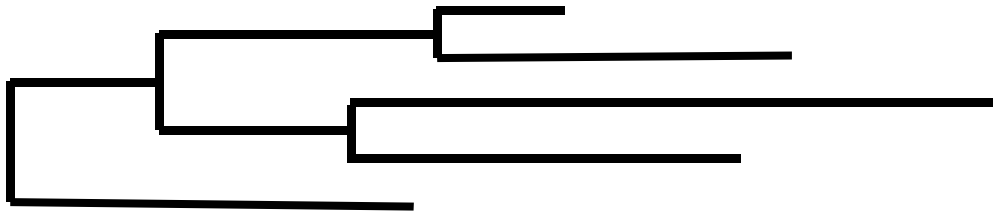
nodes i and j are at same evolutionary distance from k – dendrogram will therefore have ‘aligned’ leafs; i.e. they are all at same distance from root

No need to memorise formula

Evolutionary clock speeds



Uniform clock: **Ultrametric distances** lead to identical distances from root to leaves



Non-uniform evolutionary clock: leaves have different distances to the root -- an important property is that of **additive trees**. These are trees where the distance between any pair of leaves is the sum of the lengths of edges connecting them. Such trees obey the so-called **4-point condition** (next slide).

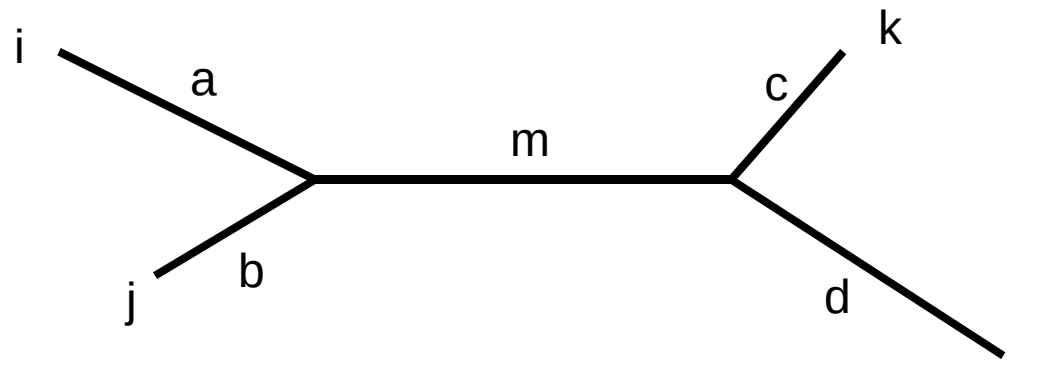
Additive trees

All distances satisfy 4-point condition:

For all leaves i, j, k, l :

$$d(i, j) + d(k, l) \leq d(i, k) + d(j, l) = d(i, l) + d(j, k)$$

$$(a+b) + (c+d) \leq (a+m+c) + (b+m+d) = (a+m+d) + (b+m+c)$$



Result: all pairwise distances obtained by traversing the tree

No need to memorise formula

Additive trees

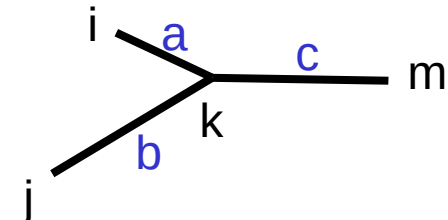
In **additive** trees, the distance between any pair of leaves is the sum of lengths of edges connecting them

Given a set of additive distances: a unique tree T can be constructed:

- For two neighbouring leaves i, j with common parent k , place parent node k at a distance from any node m with

$$d(k, m) = \frac{1}{2} (d(i, m) + d(j, m) - d(i, j))$$

$$c = \frac{1}{2} ((a+c) + (b+c) - (a+b))$$



No need to memorise formula

Ultrametric/Additive distances

If d is ultrametric then d is additive

If d is additive it does not follow that d is ultrametric

Can you prove the first statement?

Distance based -Neighbour joining (Saitou and Nei, 1987)

- Widely used method to cluster DNA or protein sequences
- **Global** measure – keeps total branch length minimal, tends to produce a tree with minimal total branch length (concept of minimal evolution)
- Agglomerative algorithm
- Leads to **unrooted** tree

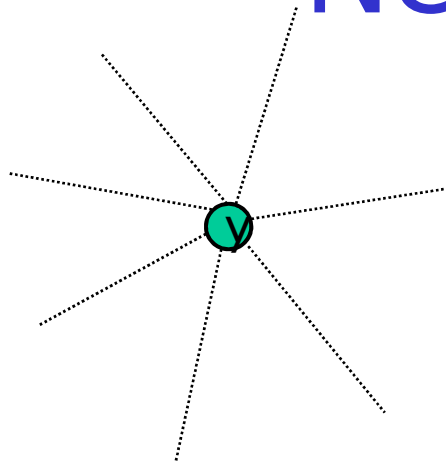
Neighbour-Joining (Cont.)

- Guaranteed to produce correct tree if distances are **additive**
- May even produce good tree if distances are not additive
- At each step, join two nodes such that total tree distances are minimal (whereby the number of nodes is decreased by 1)

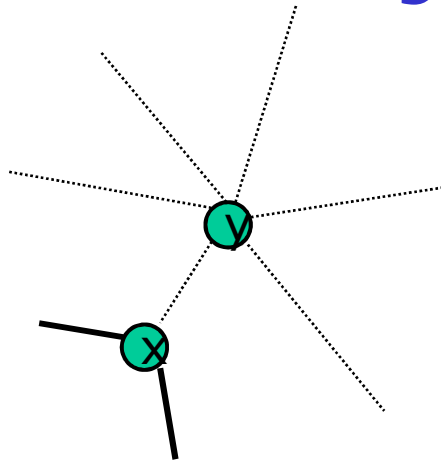
Neighbour-Joining

- Contrary to UPGMA, NJ does not assume taxa to be equidistant from the root
- NJ corrects for unequal evolutionary rates between sequences by using a conversion step
- This conversion step requires the calculation of **converted (corrected) distances, r-values** (r_i) and **transformed r values** (r'_i), where $r_i = \sum d_{ij}$ and $r'_i = r_i / (n-2)$, with n each time the number of (remaining) nodes in the tree
- Procedure:
 - NJ begins with an unresolved **star tree** by joining all taxa onto a single node
 - Progressively, the tree is decomposed (**star decomposition**), by selecting each time the taxa with the shortest corrected distance, until all internal nodes are resolved

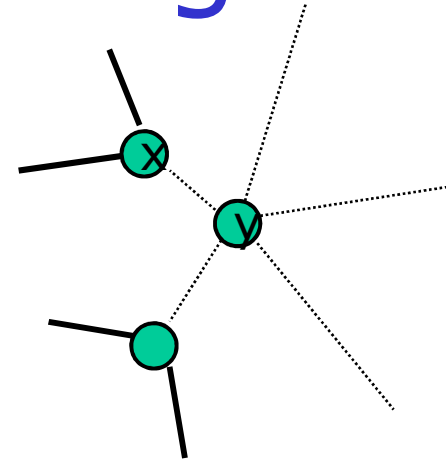
Neighbour joining



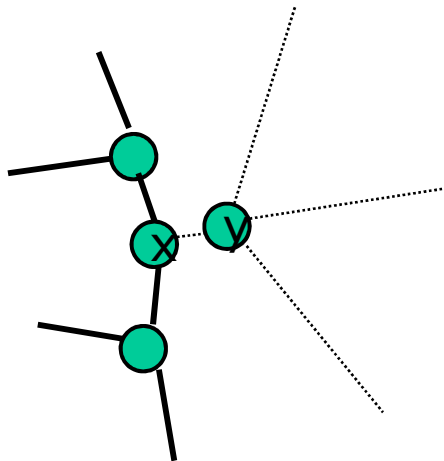
(a)



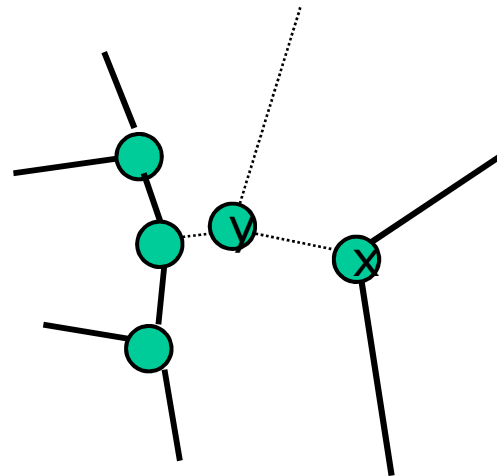
(b)



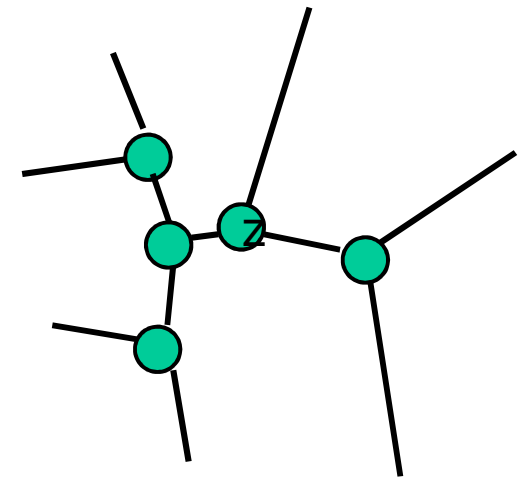
(c)



(d)



(e)



(f)

At each step all possible 'neighbour joinings' are checked and the one corresponding to the minimal total tree length (calculated by adding all branch lengths) is taken.

Neighbour joining – ‘correcting’ distances

Finding neighbouring leaves:

Define

$$d'_{ij} = d_{ij} - \frac{1}{2} (r_i + r_j) \quad [d'_{ij} \text{ is corrected distance}]$$

Where

$$r_i = \sum_k d_{ik} \quad \text{and}$$

$$r'_i = \frac{1}{|L| - 2} \sum_k d_{ik} \quad [|L| \text{ is current number of nodes}]$$

Total tree length D_{ij} is minimal iff i and j are neighbours

No need to memorise

Algorithm: Neighbour joining

Initialisation:

- Define T to be set of leaf nodes, one per sequence
- Let $L = T$

Iteration:

- Pick i, j (neighbours) such that $d'_{i,j}$ is minimal (minimal total tree length) *[this does not mean that the OTU-pair with smallest uncorrected distance is selected!]*
- Define new ancestral node k , and set $d_{km} = \frac{1}{2} (d_{im} + d_{jm} - d_{ij})$ for all $m \in L$
- Add k to T , with edges of length $d_{ik} = \frac{1}{2} (d_{ij} + r'_i - r'_j)$
- Remove i, j from L ; Add k to L

Termination:

- When L consists of two nodes i, j and the edge between them of length d_{ij}

No need to memorise, but know how NJ works intuitively

Algorithm: Neighbour joining

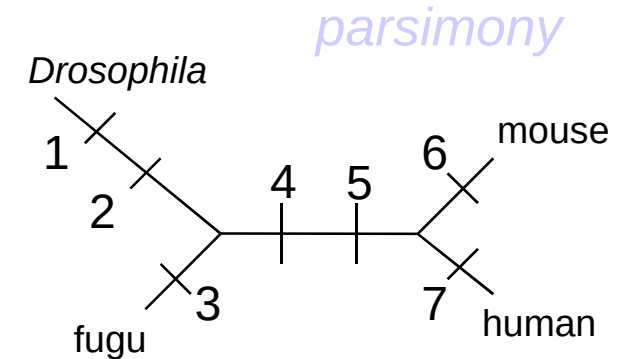
NJ algorithm in words:

1. Make star tree with 'fake' distances (we need these to be able to calculate total branch length)
2. Check all $n(n-1)/2$ possible pairs and join the pair that leads to smallest total branch length. You do this for each pair by calculating the real branch lengths from the pair to the common ancestor node (which is created here – 'y' in the preceding slide) and from the latter node to the tree
3. Select the pair that leads to the smallest total branch length (by adding up real and 'fake' distances). Record and then delete the pair and their two branches to the ancestral node, but keep the new ancestral node. The tree is now 1 one node smaller than before.
4. Go to 2, unless you are done and have a complete tree with all real branch lengths (recorded in preceding step)

Parsimony & Distance

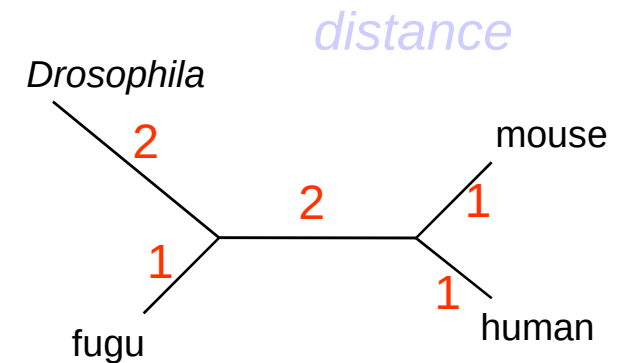
Sequences

	1	2	3	4	5	6	7
<i>Drosophila</i>	t	t	a	t	t	a	a
fugu	a	a	t	t	t	a	a
mouse	a	a	a	a	a	t	a
human	a	a	a	a	a	a	t



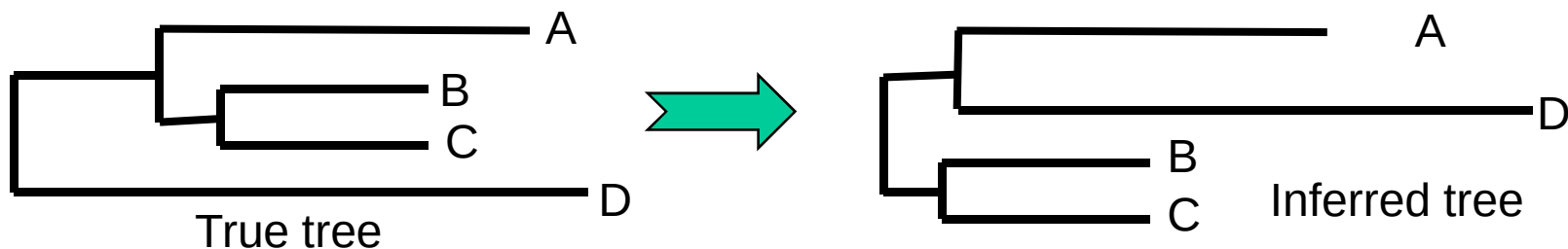
human	x			
mouse	2	x		
fugu	4	4	x	
<i>Drosophila</i>	5	5	3	x

human mouse fugu *Drosophila*



Problem: Long Branch Attraction (LBA)

- Particular problem associated with parsimony methods
- Rapidly evolving taxa are placed together in a tree regardless of their true position
- Partly due to assumption in parsimony that all lineages evolve at the same rate
- This means that also UPGMA suffers from LBA
- Some evidence exists that also implicates NJ



Maximum likelihood

Pioneered by Joe Felsenstein

- If *data*=alignment, *hypothesis* = tree, and under a given *evolutionary model*, maximum likelihood selects the *hypothesis* (tree) that maximises the observed *data*
- A statistical (Bayesian) way of looking at this is that the tree with the largest *posterior probability* is calculated based on the *prior probabilities*; i.e. the evolutionary model (or observations).
- Extremely time consuming method
- We also can test the relative fit to the tree of different models (Huelsenbeck & Rannala, 1997)

Maximum likelihood

Methods to calculate ML tree:

- **Phylip** (<http://evolution.genetics.washington.edu/phylip.html>)
- **Paup** (<http://paup.csit.fsu.edu/index.html>)
- **MrBayes** (<http://mrbayes.csit.fsu.edu/index.php>)

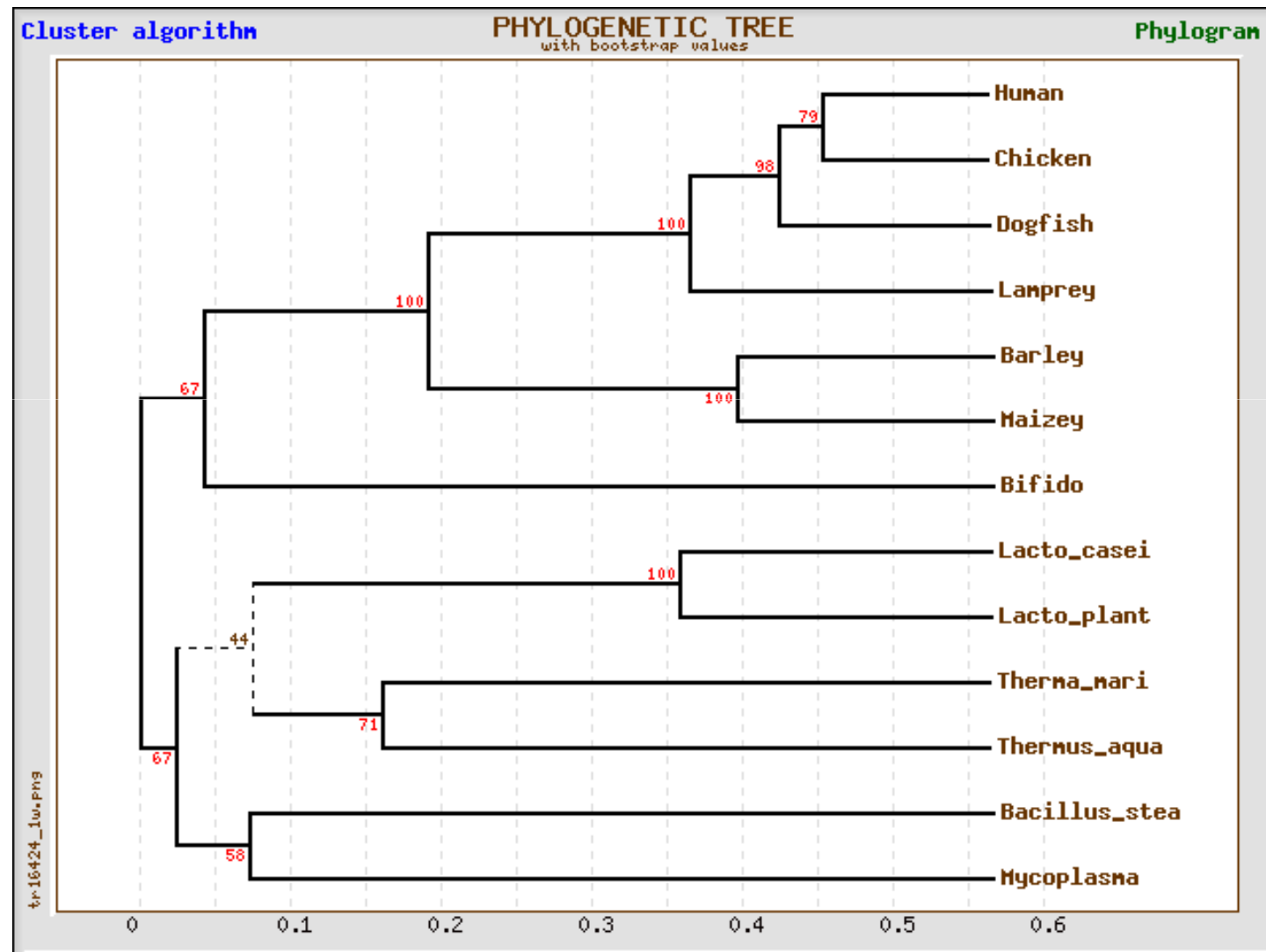
Method to analyse phylogenetic tree with ML:

- **PAML** (<http://abacus.gene.ucl.ac.uk/software/paml.htm>)
The strength of PAML is its collection of sophisticated substitution models to analyse trees.
- Programs such as PAML can test the relative fit to the tree of different models (Huelsenbeck & Rannala, 1997)

Maximum likelihood

- A number of ML tree packages (e.g. Phylip, PAML) contain tree algorithms that include the assumption of a uniform molecular clock as well as algorithms that don't
- These can both be run on a given tree, after which the results can be used to estimate the probability of a uniform clock.

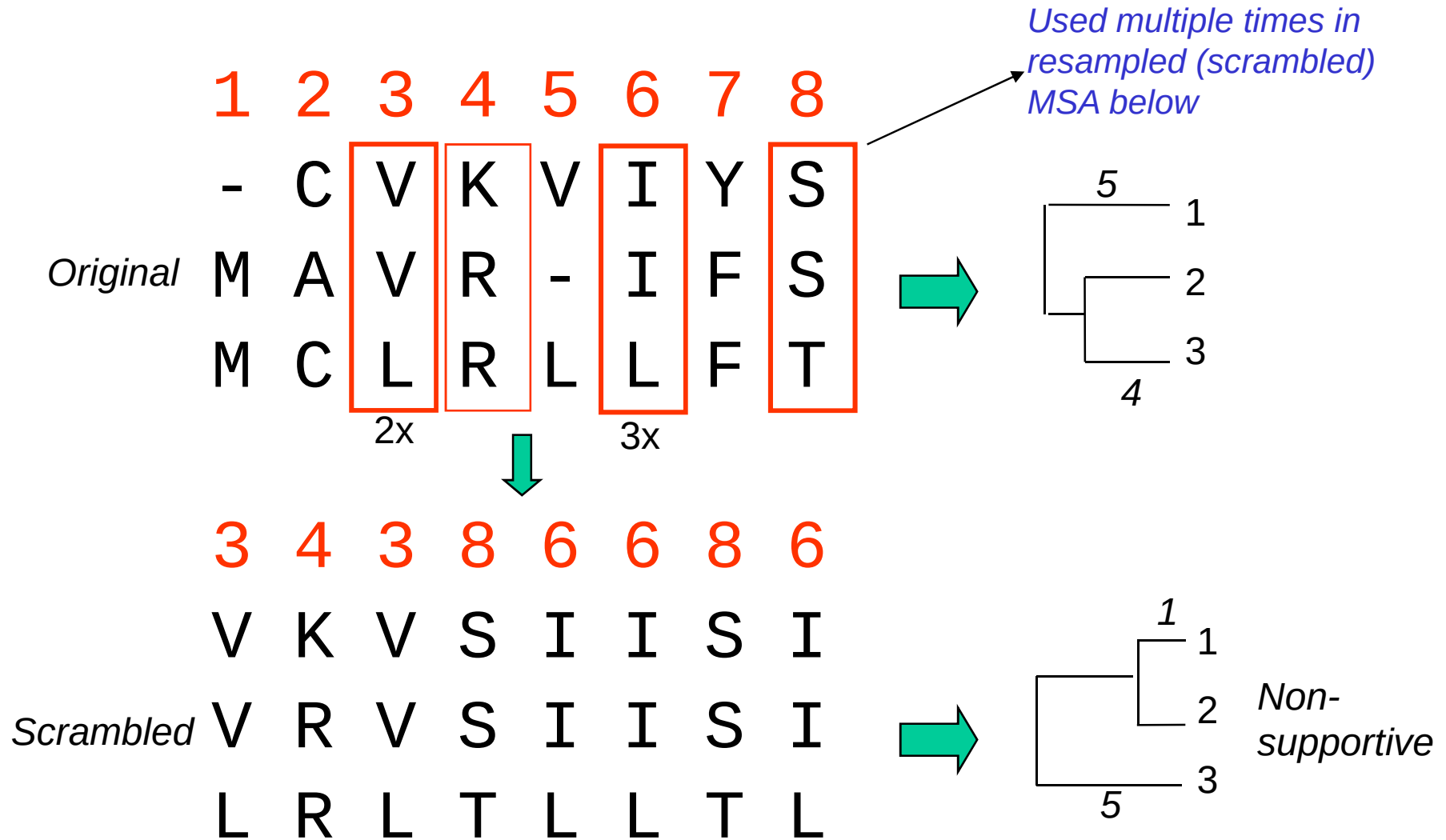
How to assess confidence in tree



How to assess confidence in tree

- Distance method – *bootstrap*:
 - Select multiple alignment columns *with replacement* (scramble the MSA)
 - Recalculate tree
 - Compare branches with original (target) tree
 - Repeat 100-1000 times, so calculate 100-1000 different trees
 - How often is branching (point between 3 nodes) preserved for each internal node in these 100-1000 trees?
 - Bootstrapping uses resampling of the data

The Bootstrap -- example



Only boxed alignment columns are randomly selected in this example

Some versatile phylogeny software packages

- MrBayes
- Paup
- Phylip



MrBayes: Bayesian Inference of Phylogeny

- MrBayes is a program for the Bayesian estimation of phylogeny.
- Bayesian inference of phylogeny is based upon a quantity called the posterior probability distribution of trees, which is the probability of a tree conditioned on the observations.
- The conditioning is accomplished using Bayes's theorem. The posterior probability distribution of trees is impossible to calculate analytically; instead, MrBayes uses a simulation technique called Markov chain Monte Carlo (or MCMC) to approximate the posterior probabilities of trees.
- The program takes as input a character matrix in a NEXUS file format. The output is several files with the parameters that were sampled by the MCMC algorithm. MrBayes can summarize the information in these files for the user.

No need to memorise



MrBayes: Bayesian Inference of Phylogeny

MrBayes program features include:

- A common command-line interface for Macintosh, Windows, and UNIX operating systems;
- Extensive help available via the command line;
- Ability to analyze nucleotide, amino acid, restriction site, and morphological data;
- Mixing of data types, such as molecular and morphological characters, in a single analysis;
- A general method for assigning parameters across data partitions;
- An abundance of evolutionary models, including 4 X 4, doublet, and codon models for nucleotide data and many of the standard rate matrices for amino acid data;
- Estimation of positively selected sites in a fully hierarchical Bayes framework;
- The ability to spread jobs over a cluster of computers using MPI (for Macintosh and UNIX environments only).

No need to memorise

PAUP

PAUP* Version 4

...tools for inferring
and interpreting
phylogenetic trees

Analyze

- Molecular sequences
- Morphological data
- Other data types

Using

- Maximum likelihood
- Parsimony
- Distance methods



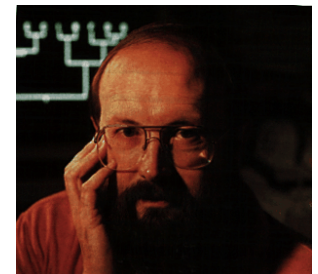


Phylip – by Joe Felsenstein

Phylip programs by type of data

- DNA sequences
- Protein sequences
- Restriction sites
- Distance matrices
- Gene frequencies
- Quantitative characters
- Discrete characters
- tree plotting, consensus trees, tree distances and tree manipulation

<http://evolution.genetics.washington.edu/phylip.html>



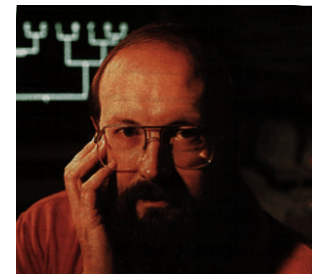


Phylip – by Joe Felsenstein

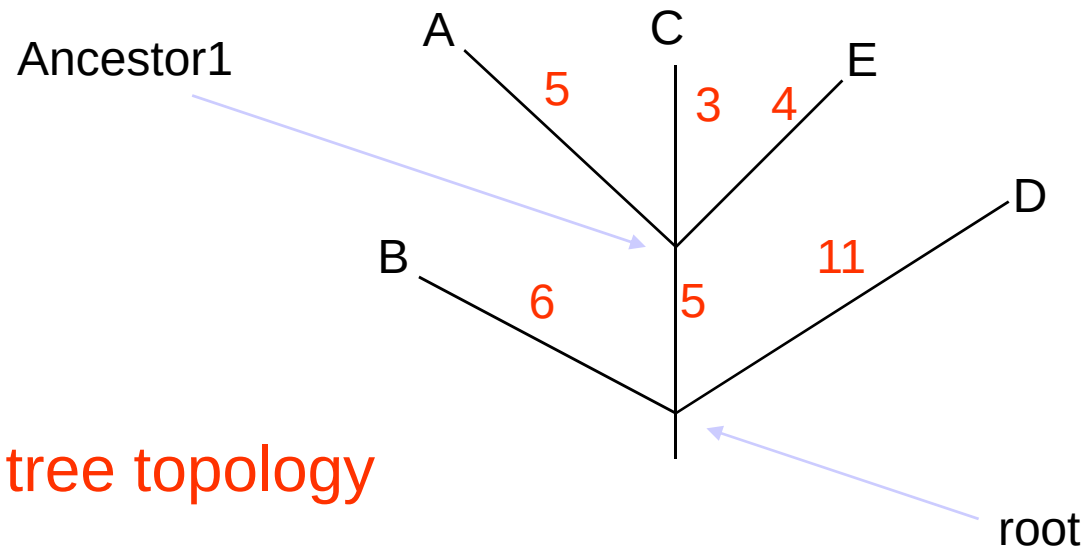
Phylip programs by type of algorithm

- Heuristic tree search
- Branch-and-bound tree search
- Interactive tree manipulation
- Plotting trees, consensus trees, tree distances
- Converting data, making distances or bootstrap replicates

<http://evolution.genetics.washington.edu/phylip.html>



The Newick tree format



`(B,(A,C,E),D);` -- tree topology

`(B:6.0,(A:5.0,C:3.0,E:4.0):5.0,D:11.0);` -- with branch lengths

`(B:6.0,(A:5.0,C:3.0,E:4.0)Ancestor1:5.0,D:11.0)Root;`
-- with branch lengths and ancestral node names

Distance methods: fastest

- Clustering criterion using a distance matrix
- Distance matrix filled with alignment scores (sequence identity, alignment scores, E-values, etc.)
- Cluster criterion

Kimura's correction for protein sequences (1983)

This method is used for proteins only. Gaps are ignored and only exact matches and mismatches contribute to the match score.

Distances get 'stretched' to correct for back mutations

$$S = m/npos,$$

Where m is the number of exact matches and $npos$ the number of positions scored

$$D = 1 - S$$

$$\text{Corrected distance} = -\ln(1 - D - 0.2D^2) \quad (\text{see also earlier slide})$$

Reference:

M. Kimura, The Neutral Theory of Molecular Evolution, Camb. Uni. Press, Camb., 1983.

Sequence similarity criteria for phylogeny

- In addition to the Kimura correction, there are various models to correct for the fact that the true rate of evolution cannot be observed through nucleotide (or amino acid) exchange patterns (e.g. due to back mutations).
- Saturation level is ~94%, higher real mutations are no longer observable

A widely used protocol to infer a phylogenetic tree

- Make an MSA
- Take only gapless positions and calculate pairwise sequence distances using Kimura correction
- Fill distance matrix with corrected distances
- Calculate a phylogenetic tree using Neighbour Joining (NJ)

Phylogeny disclaimer

- With all of the phylogenetic methods, you calculate one tree out of very many alternatives.
- Only one tree can be correct and depict evolution accurately.
- Incorrect trees will often lead to 'more interesting' phylogenies, e.g. the whale originated from the fruit fly etc.

Take home messages

- Rooted/unrooted trees, how to root a tree
- Make sure you can do the UPGMA algorithm and understand the basic steps of the NJ algorithm
- Understand the three basic classes of phylogenetic methods: distance-based, parsimony and maximum likelihood
- Make sure you understand bootstrapping (to assess confidence in tree splits)