

01/11/05

Evolution/Phylogeny

Bioinformatics Course
Algorithms for Genomes

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 (and thus evolution)”

Content

- **Evolution**
 - requirements
 - negative/positive selection on genes (e.g. Ka/Ks)
 - gene conversion
 - homology/paralogy/orthology (operational definition 'bi-directional best hit')
- **Clustering**
 - single linkage
 - complete linkage
- **Phylogenetic trees**
 - ultrametric distance (uniform molecular clock)
 - additive trees (4-point condition)
 - UPGMA algorithm
 - NJ algorithm
 - bootstrapping

Darwinian Evolution

What is needed:

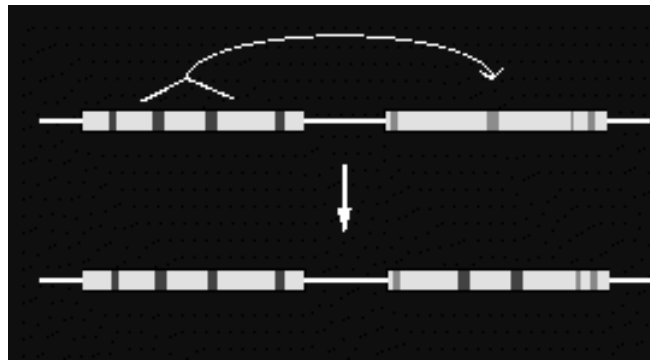
2. Template (DNA)
3. Copying mechanism
(meiosis/fertilisation)
4. Variation (e.g. resulting from copying errors, gene conversion, crossing over, genetic drift, etc.)
5. Selection

Gene conversion

- The transfer of DNA sequences between two homologous genes, most often by unequal crossing over during meiosis
- Can be a mechanism for mutation if the transfer of material disrupts the coding sequence of the gene or if the transferred material itself contains one or more mutations

Gene conversion

- Gene conversion can influence the evolution of gene families, having the capacity to generate both diversity and homogeneity.
- Example of a intrachromosomal gene conversion event:



- The potential evolutionary significance of gene conversion is directly related to its frequency in the germ line. While meiotic inter- and intrachromosomal gene conversion is frequent in fungal systems, it has hitherto been considered impractical in mammals. However, meiotic gene conversion has recently been measured as a significant recombination process in mice.

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).

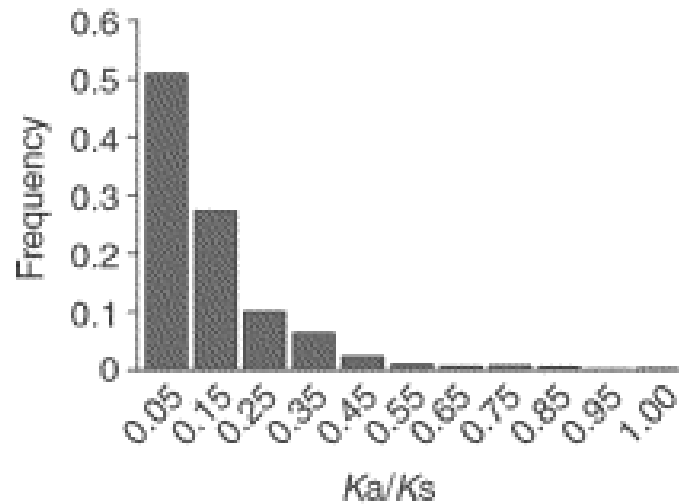
DNA evolution

- “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.

Ka/Ks Ratios

- **Ks** is defined as the number of synonymous nucleotide substitutions per synonymous site
- **Ka** is defined as the number of nonsynonymous nucleotide substitutions per nonsynonymous site
- The **Ka/Ks** ratio is used to estimate the type of selection exerted on a given gene or DNA fragment
- Need orthologous nucleotide sequence alignments
- Observe nucleotide substitution patterns at given sites and correct numbers using, for example, the Pamilo-Bianchi-Li method (Li 1993; Pamilo and Bianchi 1993).
- Correction is needed because of the following:
Consider the codons specifying aspartic acid and lysine: both start AA, lysine ends A or G, and aspartic acid ends T or C. So, if the rate at which C changes to T is higher from the rate that C changes to G or A (as is often the case), then more of the changes at the third position will be synonymous than might be expected. Many of the methods to calculate *Ka* and *Ks* differ in the way they make the correction needed to take account of this bias.

Ka/Ks ratios



TRENDS in Genetics

The frequency of different values of Ka/Ks for 835 mouse–rat orthologous genes. Figures on the x axis represent the middle figure of each bin; that is, the 0.05 bin collects data from 0 to 0.1

Ka/Ks ratios

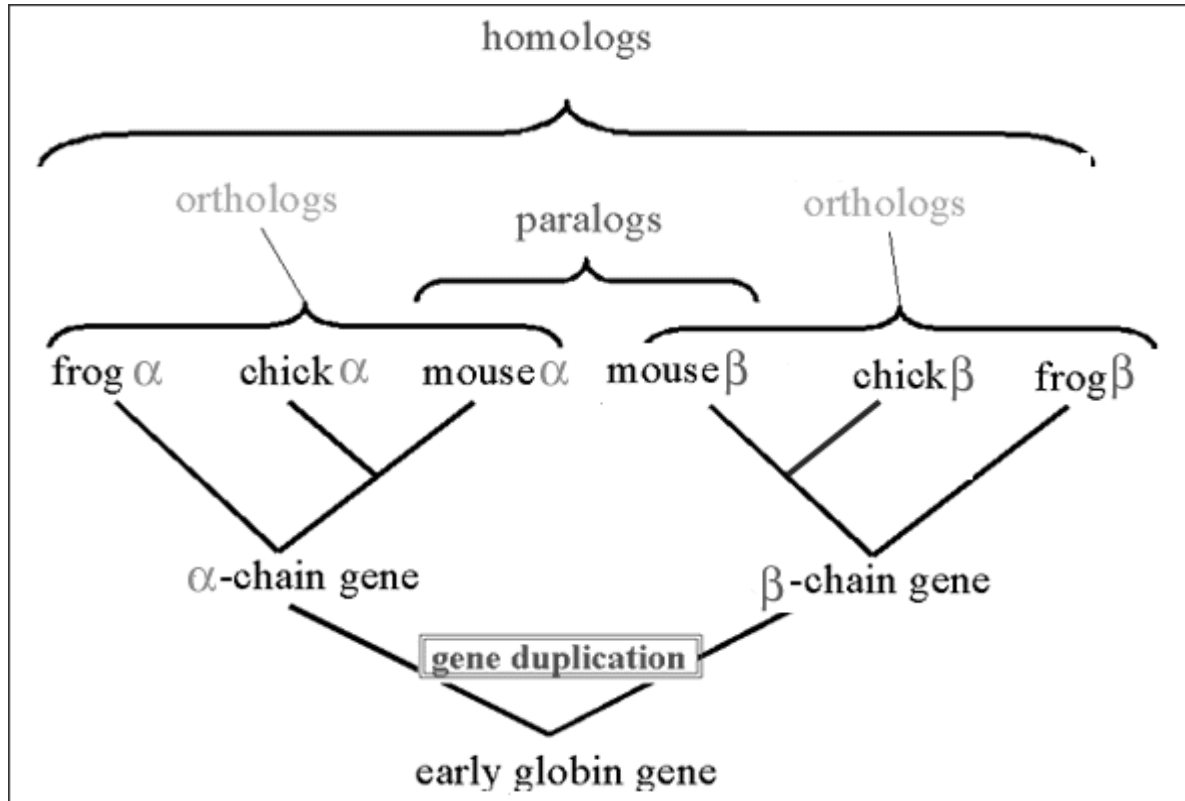
Three types of selection:

2. Negative (purifying) selection $\in Ka/Ks < 1$

3. Neutral selection (Kimura) $\in Ka/Ks \sim 1$

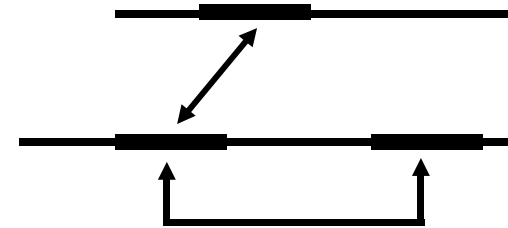
4. Positive selection $\in Ka/Ks > 1$

Orthology/paralogy



Orthologous genes are homologous (corresponding) genes in different species

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



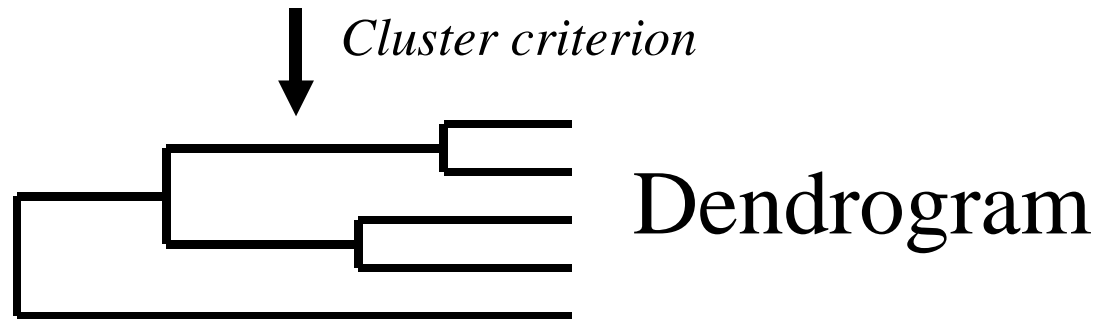
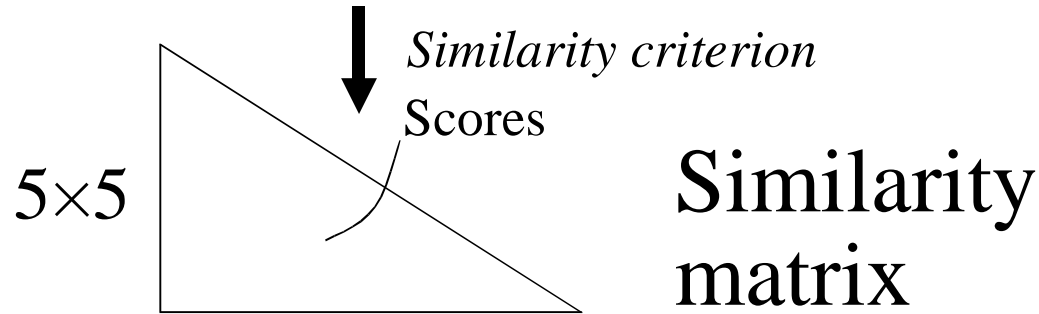
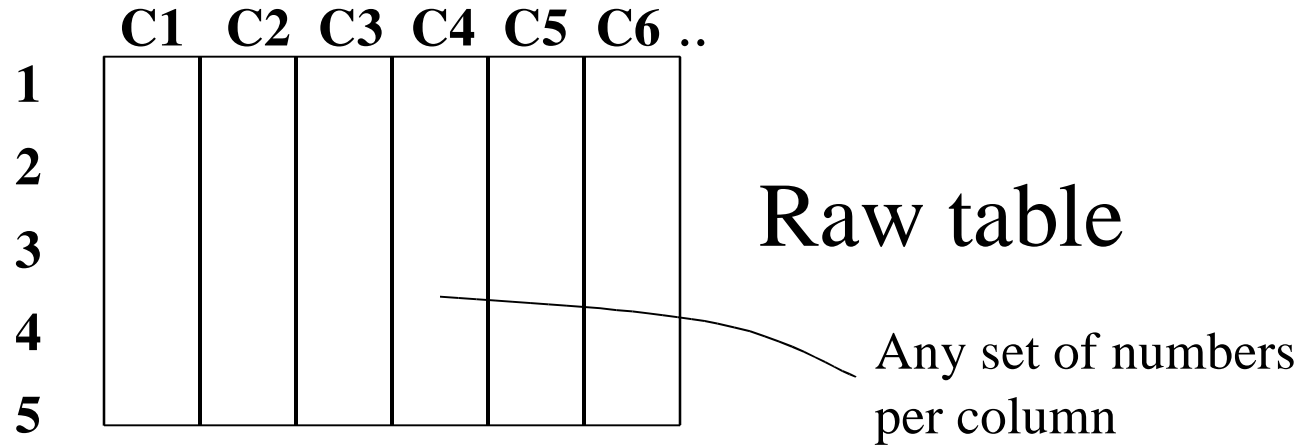
Orthology/paralogy

Operational definition of orthology:

Bi-directional best hit:

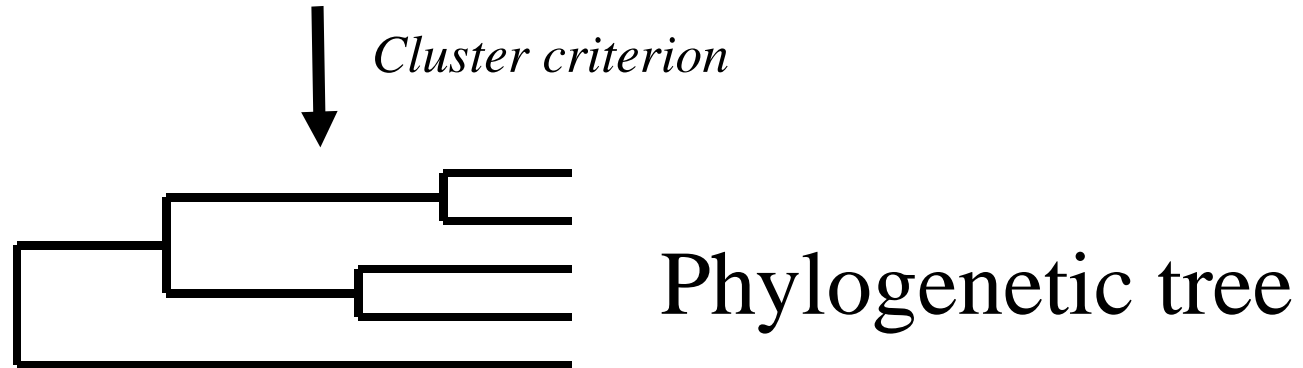
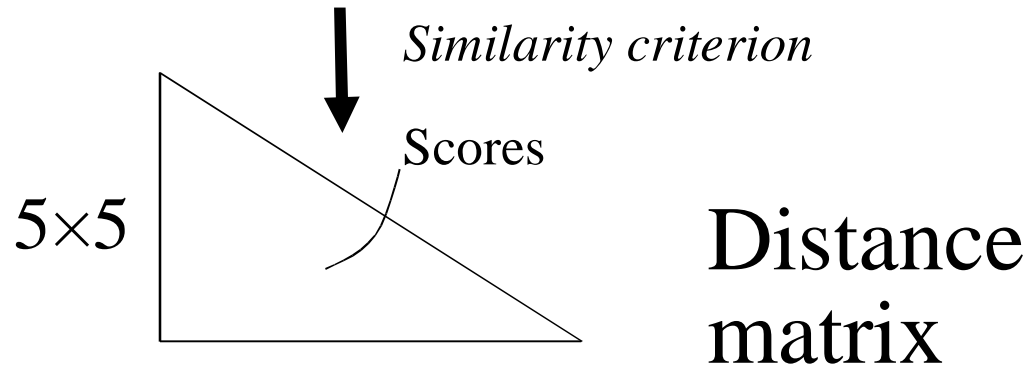
- Blast gene A in genome 1 against genome 2: gene B is best hit
 - Blast gene B against genome 1: if gene A is best hit
- $\in A$ and B are orthologous

Multivariate statistics – Cluster analysis



Multivariate statistics

Producing a Phylogenetic tree from sequences



Similarity criterion for phylogeny

- ClustalW: uses sequence identity with Kimura (1983) correction:
Corrected $K = -\ln(1.0 - K - K^2/5.0)$, where K is percentage divergence corresponding to two aligned sequences
- 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%, higher real mutations are no longer observable

Lactate dehydrogenase multiple alignment

Human	-KITVVGVGAVGMACAISILMKDLADELALVDVIEDKCLKGEMMDLQHGSFLFLRTPKIVSGKDYNVTANSKLVIIITAGARQ
Chicken	-KISVVGVGAVGMACAISILMKDLADELTLVDVVEDKCLKGEMMDLQHGSFLFKTPKITSGKDYSVTAHSKLVIVITAGARQ
Dogfish	-KITVVGVGAVGMACAISILMKDLADEVALVDVMEDKCLKGEMMDLQHGSFLFLHTAKIVSGKDYSVSAGSKLVVITAGARQ
Lamprey	SKVTIVGQVGMAAAISVLLRDLADELALVDVVEDRLKKGEMMDLLHGSFLFKTAKIVADKDYSVTAGSRLVVVITAGARQ
Barley	TKISVIGAGNVGMAIAQTIQTILTQNLADEIALVDALPDKLRGEALDLQHAAAFLLPRVRI-SGTDAAVTKNSDLVIVITAGARQ
Maizey casei	-KVILVGDGAVGSSYAYAMVLQGIQAEIGIVDIFKDKTKGDAIDL SNALPFTSPKKIYSA-EYSDAKDADLVVITAGAPQ
Bacillus	TKVSVIGAGNVGMAIAQTIILTRDLADEIALVDAVPDKLRGEMDLQHAAAFLLPRTRLVSGTDM SVTRGSDLVIVITAGARQ
Lacto__ste	-RVVIGAGFVGASYVFALMNQGIAD EIVLIDANESKAIGDAMDFNHGKVFAPKPVDIWHGDYDDCRDADLVVICAGANQ
Lacto_plant	QKVVLVGDGAVGSSYAFAMAQQGIAEEFVIVDVVKDRTKGDALDLEDAQAFTAPKKIYSG-EYSDCKDADLVVITAGAPQ
Therma_mari	MKIGIVGLGRVGSSTAFALLMKGFAREMVLIDVDKKRAEGDALDLIHGTPFTRRANIYAG-DYADLKGSDVVIVAAGVPQ
Bifido	-KLAIVIGAGAVGSTLAFAAAQRGIAREIVLEDIAKERVEAEVLDMQHGS SFYPTV SIDGSDDPEICRDADMVVITAGPRQ
Thermus_aqua	MKVGIVGSGFVGSATAYALVLQGVAREVVLVDLDRKLAQAHAEDILHATPFAHPVWVRSGW-YEDLEGARVVIVAAGVAQ
Mycoplasma	-KIALIGAGNVGNSFLYAAMNQGLASEYGIIDINPDFADGNAFDFEDASASLPFPISVSRYEYKDLKDAFDIVITAGRPQ

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

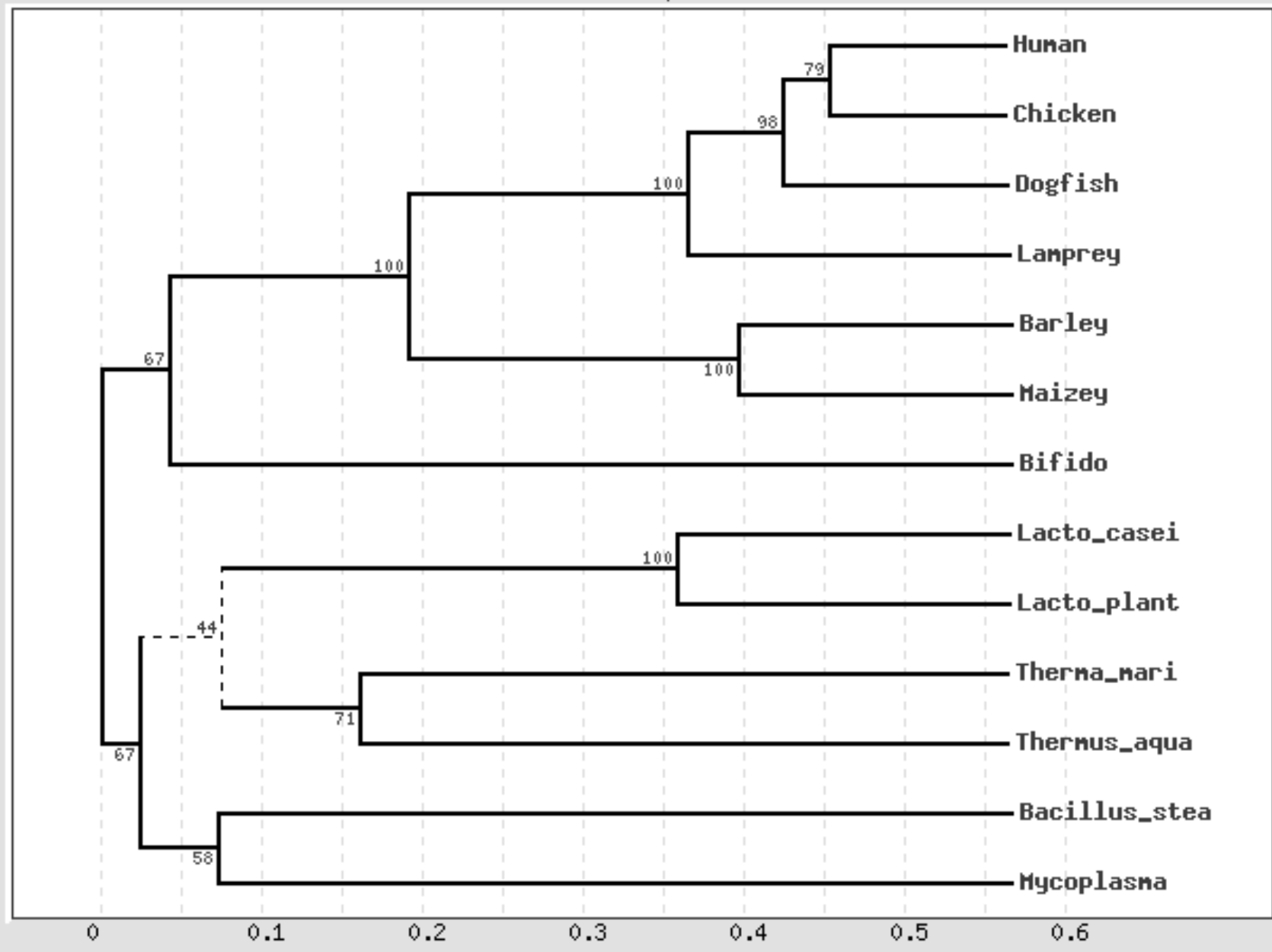
How can you see that this is a distance matrix?

Cluster algorithm

PHYLOGENETIC TREE

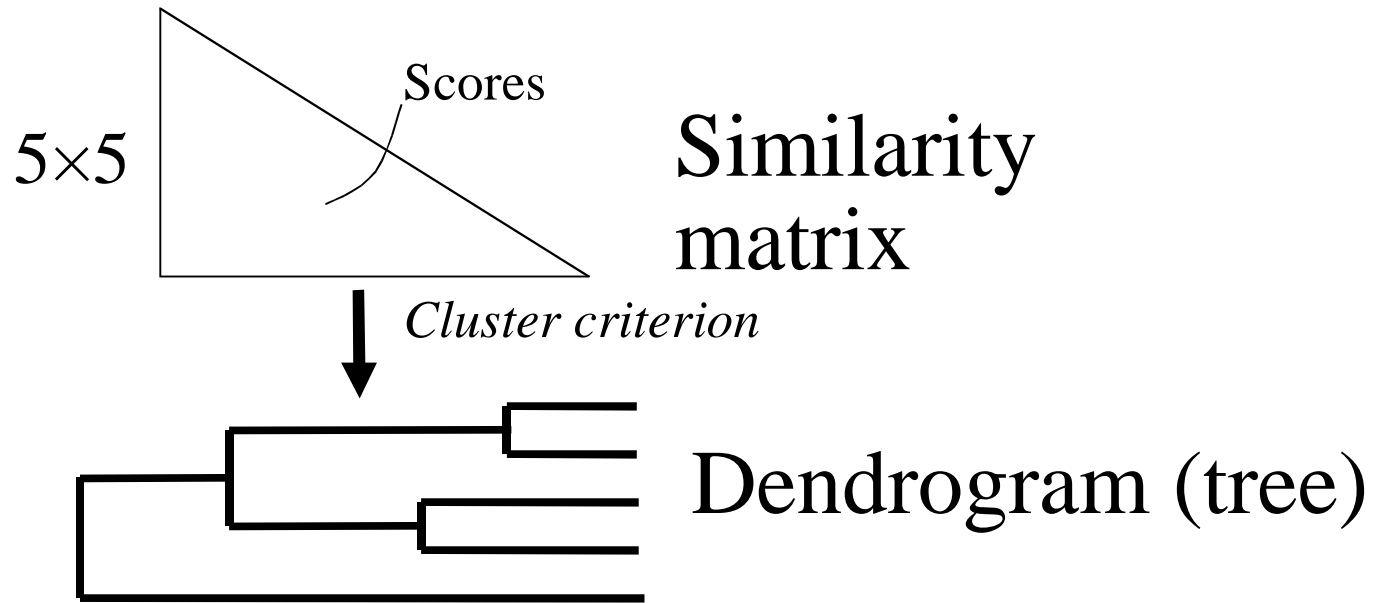
with bootstrap values

Phylogram



tr16424_1w.png

Cluster analysis – Clustering criteria



Four different clustering criteria:

Single linkage - Nearest neighbour

Complete linkage – Furthest neighbour

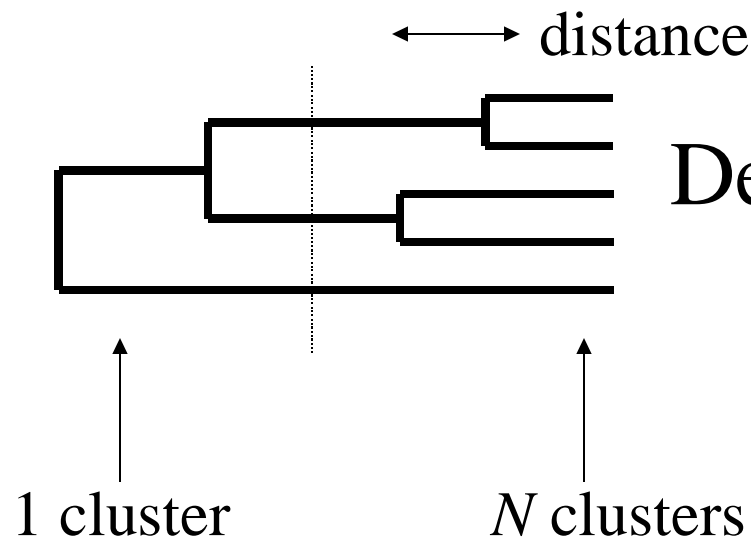
Group averaging – UPGMA

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

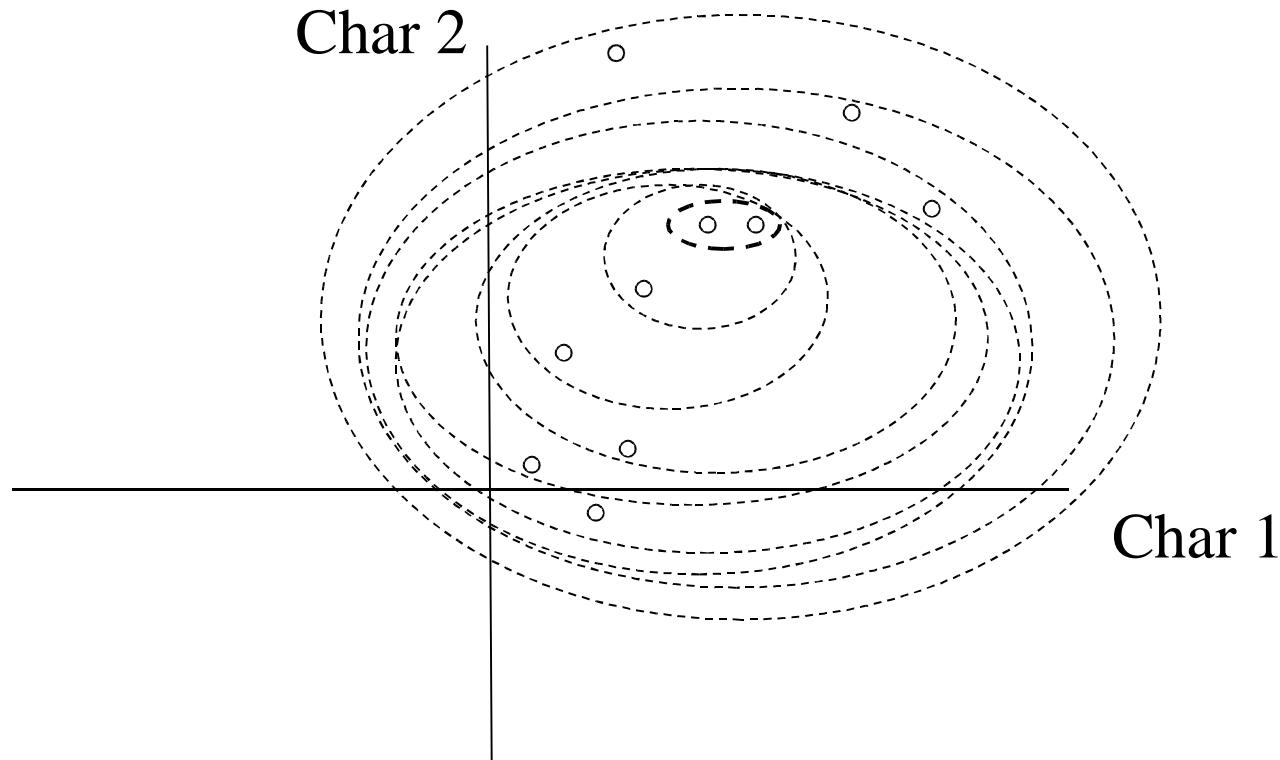
General agglomerative cluster protocol

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



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)

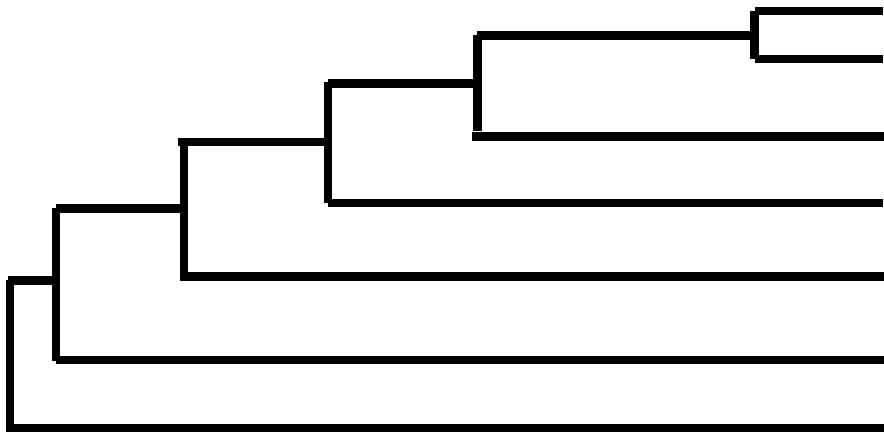


Distance from point to cluster is defined as the smallest distance between that point and any point in the cluster

Single linkage clustering (nearest neighbour)

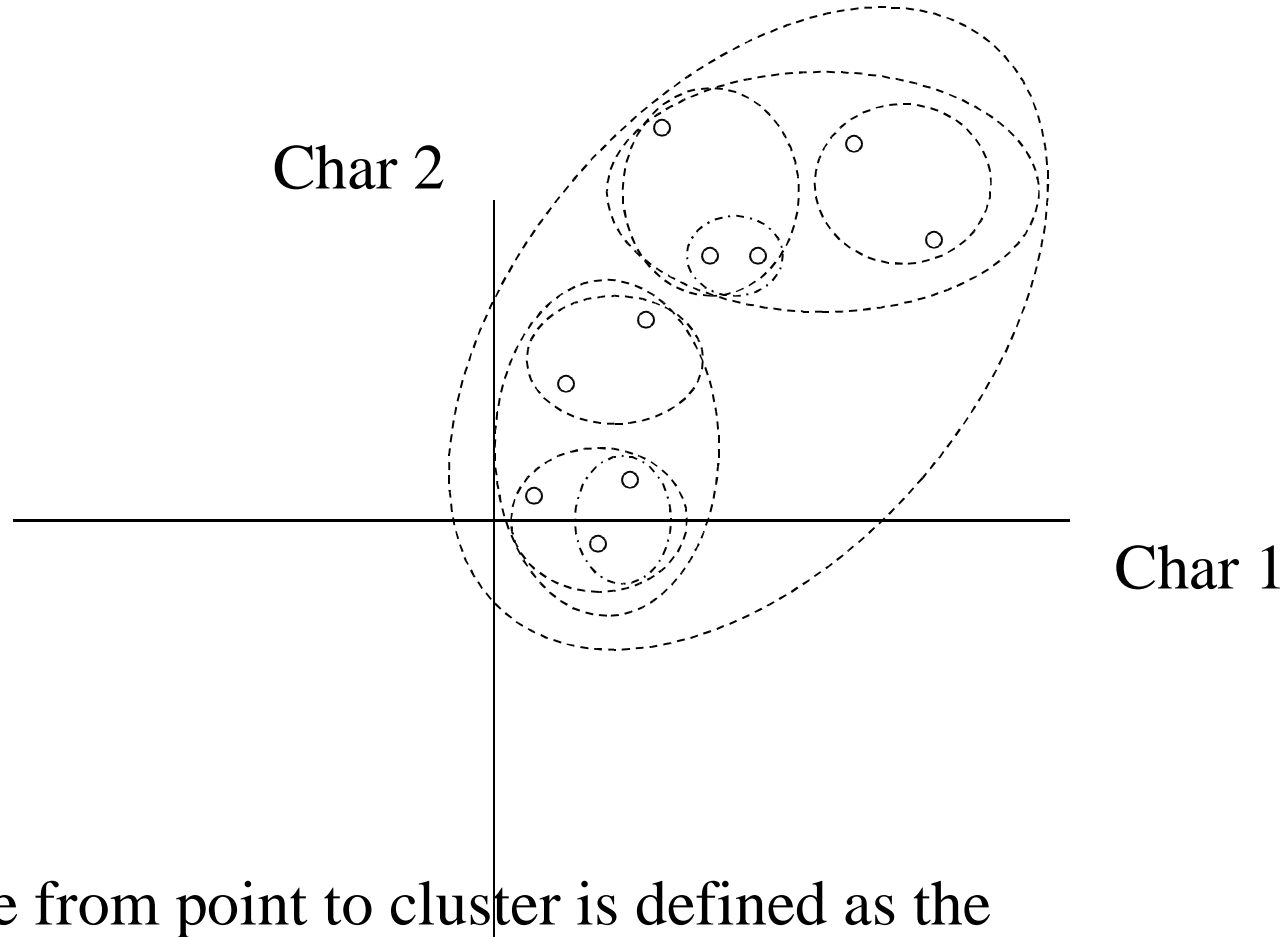
Let C_i and C_j be two disjoint clusters:

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



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)

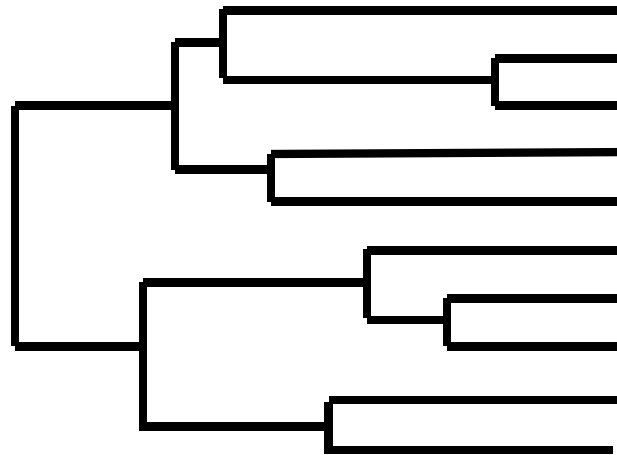


Distance from point to cluster is defined as the largest distance between that point and any point in the cluster

Complete linkage clustering (furthest neighbour)

Let C_i and C_j be two disjoint clusters:

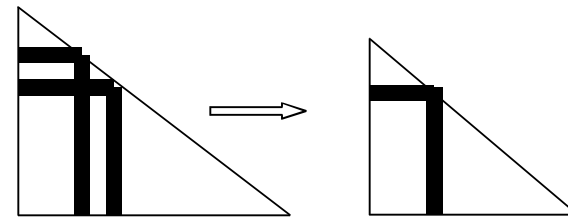
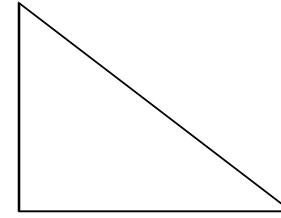
$$d_{i,j} = \text{Max}(d_{p,q}), \text{ where } p \in C_i \text{ and } q \in C_j$$



More 'structured' clusters than with single linkage clustering

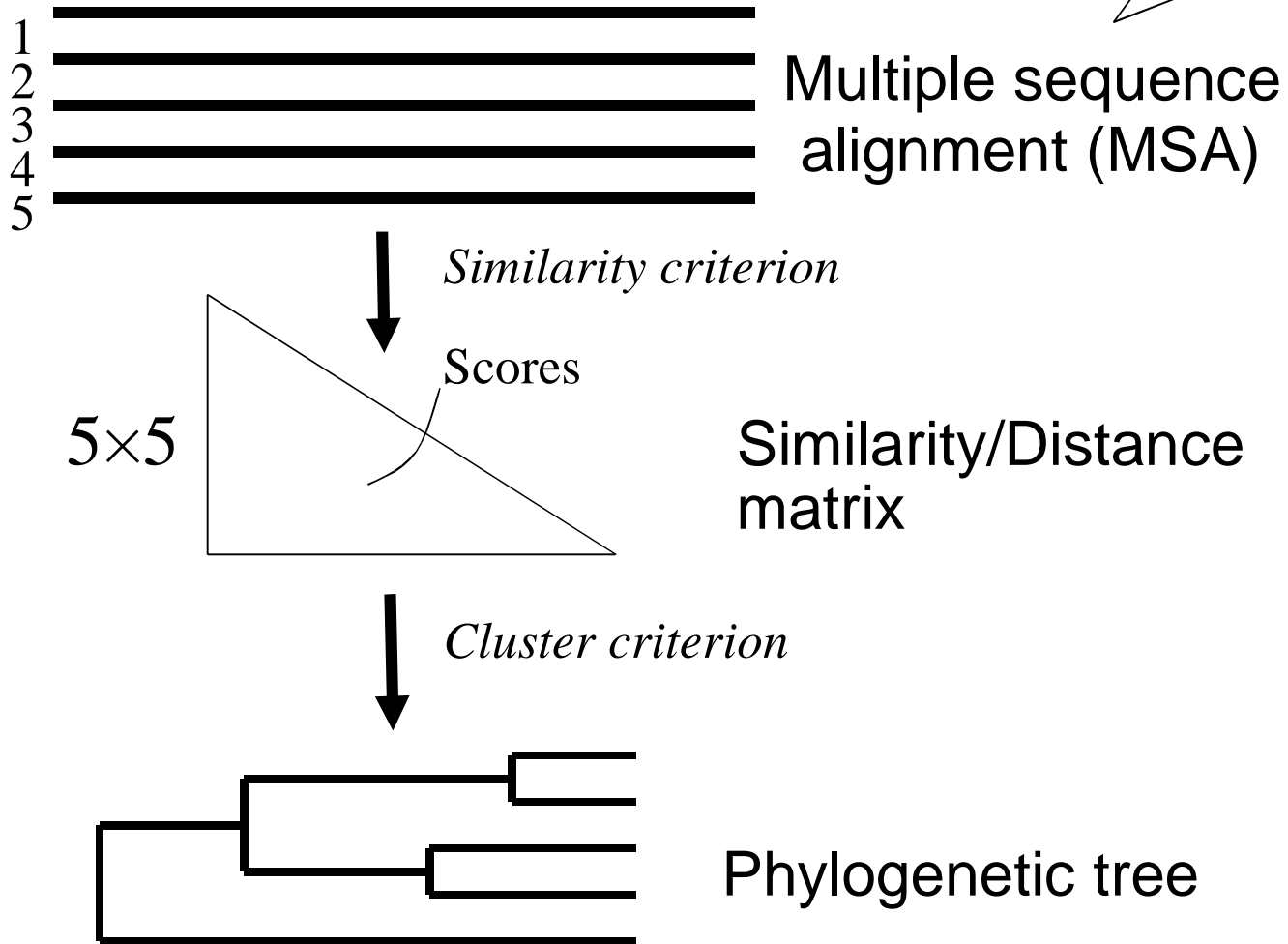
Clustering algorithm

1. Initialise (dis)similarity matrix
2. Take two points with smallest distance as first cluster
3. Merge corresponding rows/columns in (dis)similarity matrix
4. Repeat steps 2. and 3. using appropriate cluster measure until last two clusters are merged

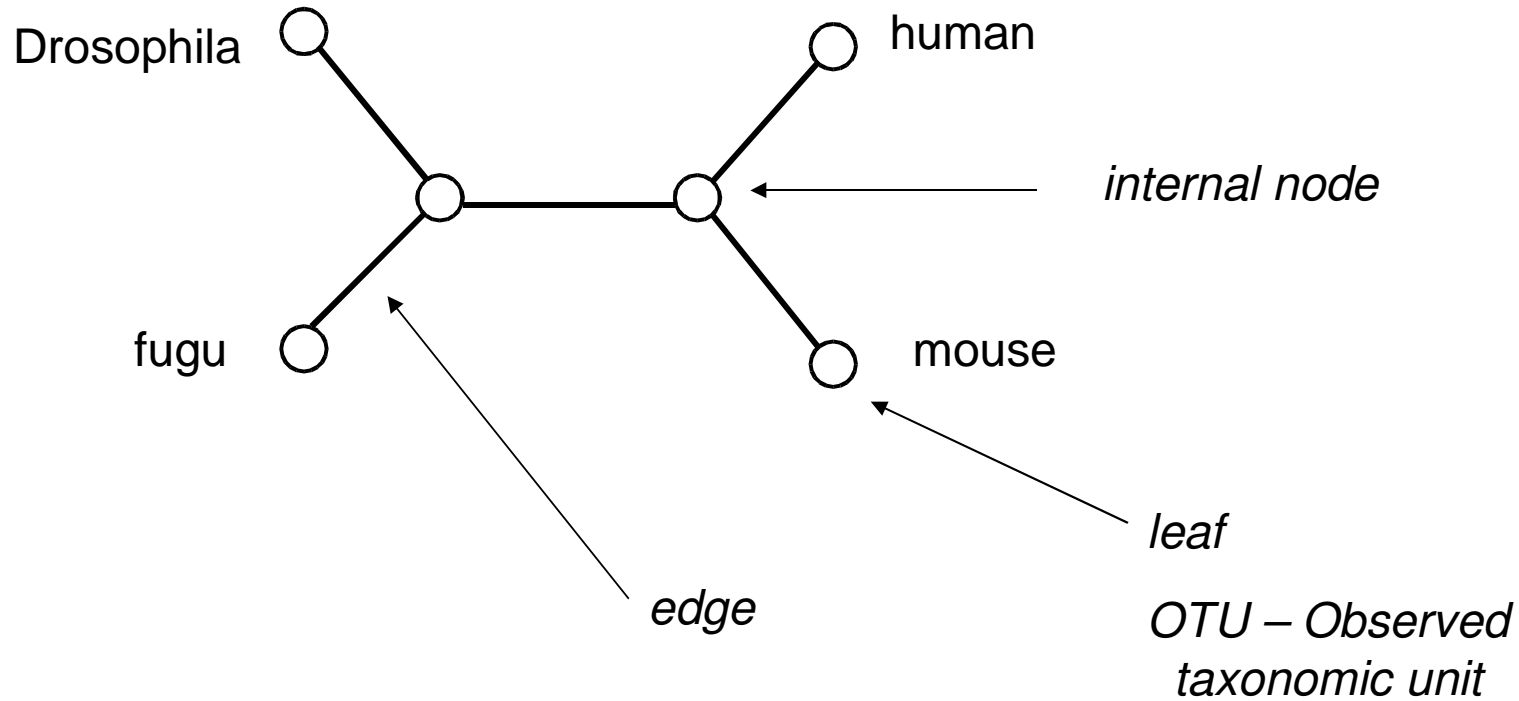


Phylogenetic trees

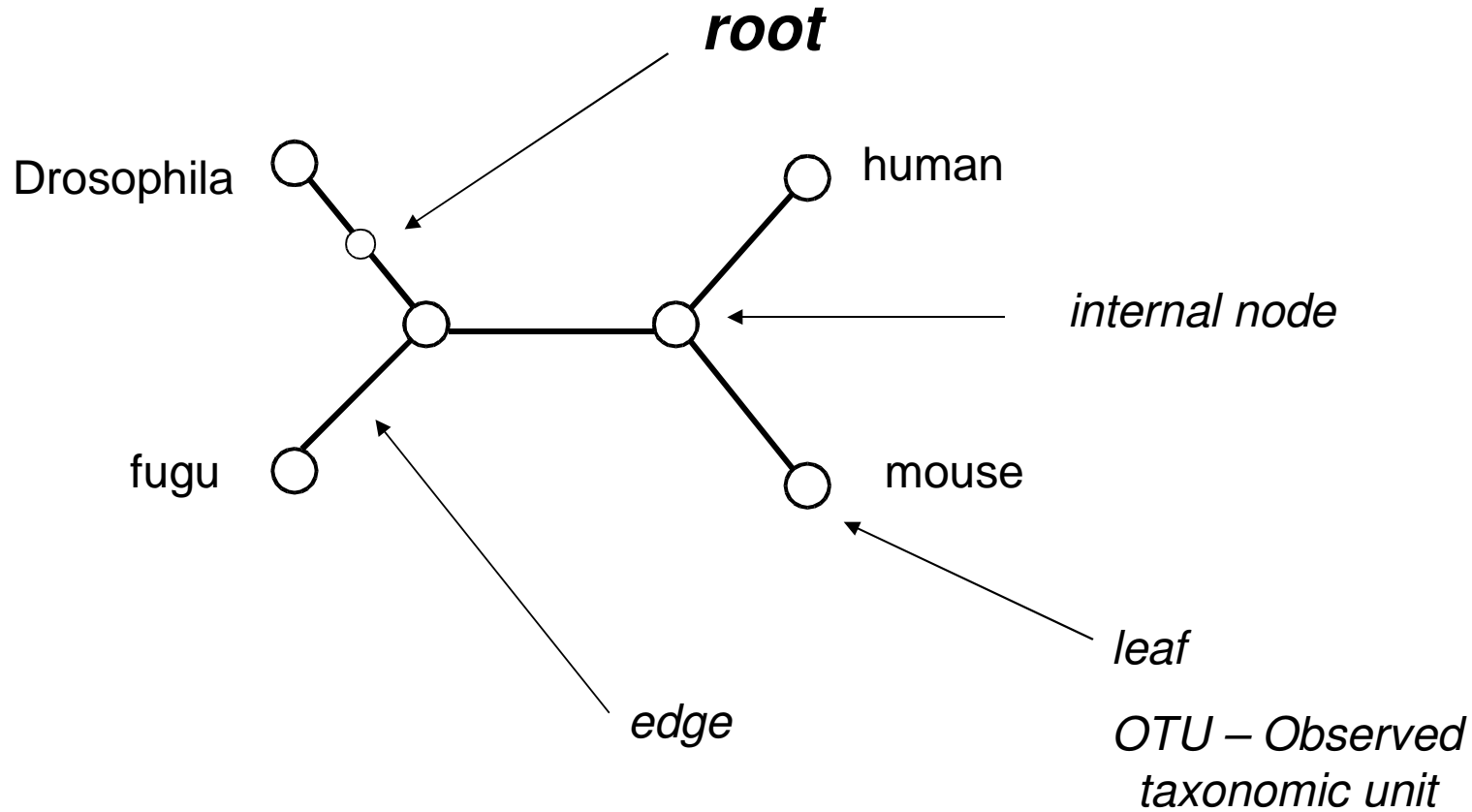
*MSA quality is crucial
for obtaining correct
phylogenetic tree*



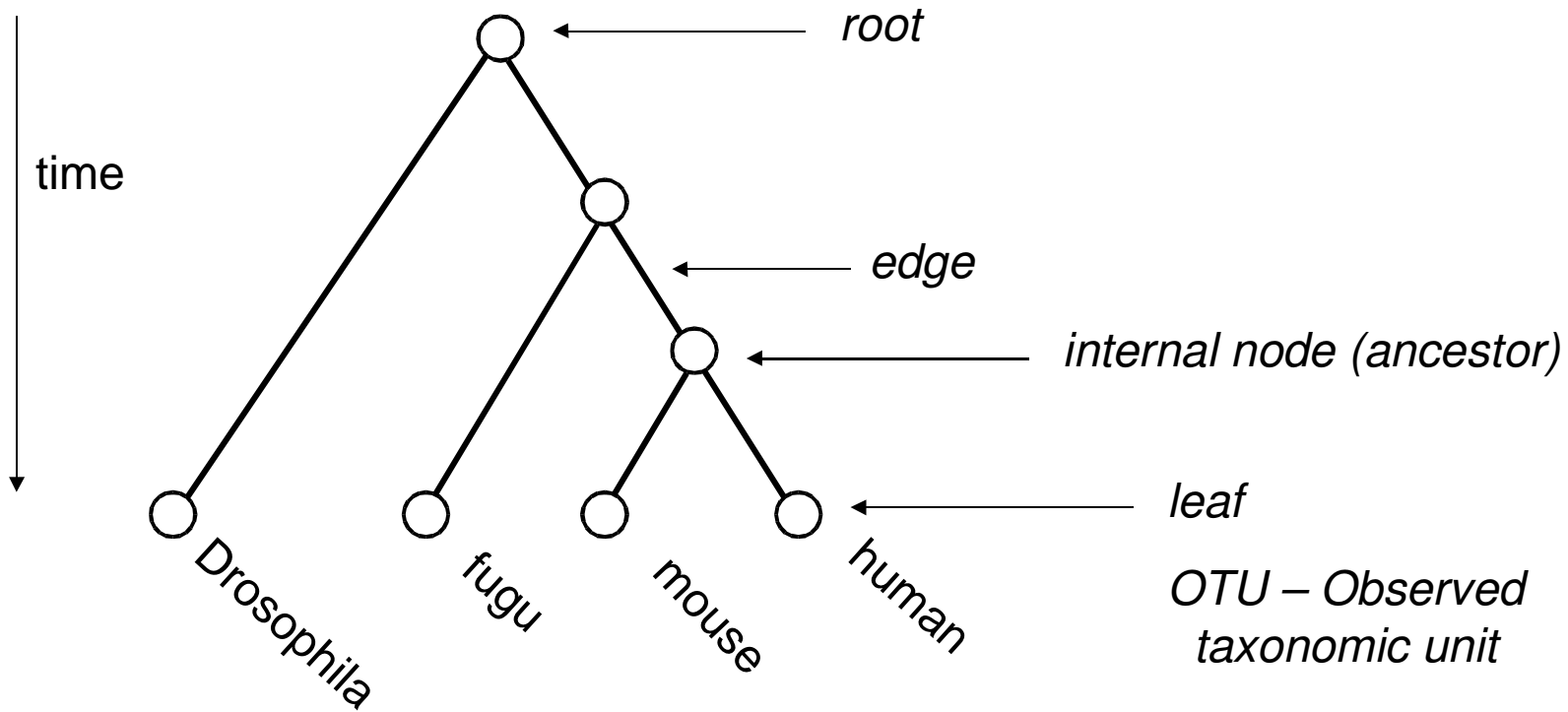
Phylogenetic tree (unrooted)



Phylogenetic tree (unrooted)

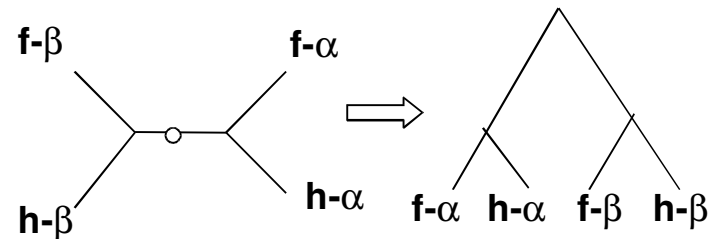
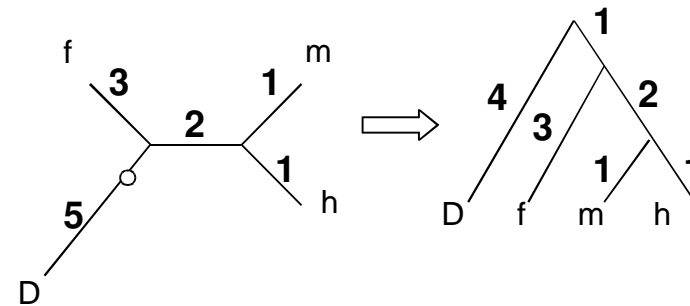
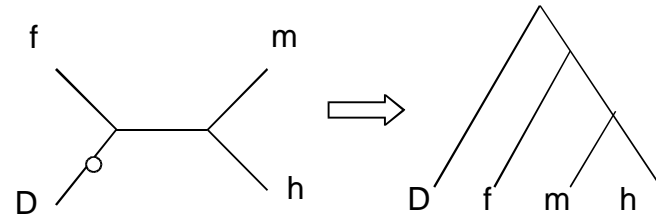


Phylogenetic tree (rooted)



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 (see earlier globin example)



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

A simple clustering method for building phylogenetic trees

Unweighted Pair Group Method using Arithmetic Averages (UPGMA)

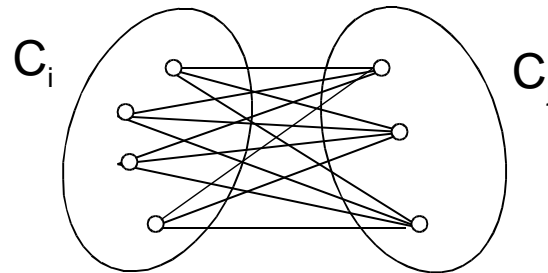
Sneath and Sokal (1973)

UPGMA

Let C_i and C_j be two disjoint clusters:

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

number of
points in
cluster



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

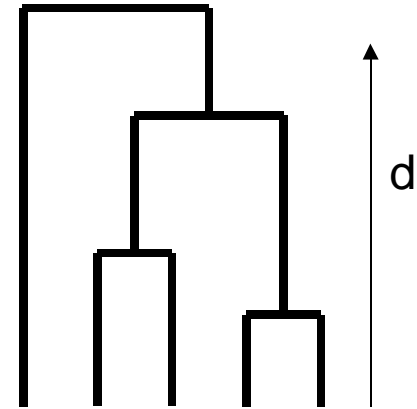
Clustering algorithm: UPGMA

Initialisation:

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

Iteration:

6. Merge cluster C_i and C_j for which d_{ij} is minimal
7. Place internal node connecting C_i and C_j at height $d_{ij}/2$
8. 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

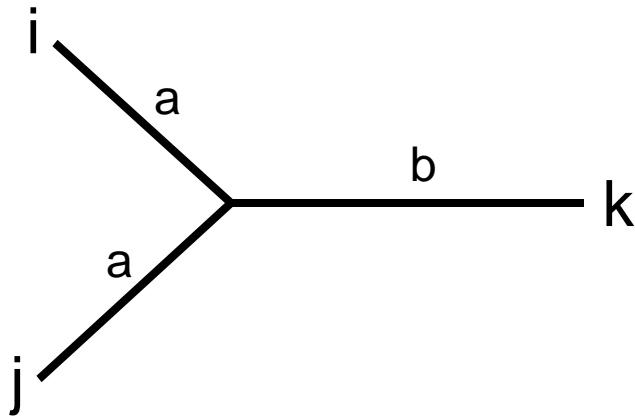
- A tree T in a metric space (M, d) where d is **ultrametric** has the following property: there is a way to place a root on T so that for all nodes in M , their distance to the root is the same. Such T is referred to as a **uniform molecular clock tree**.
- (M, d) is ultrametric if for every set of 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:

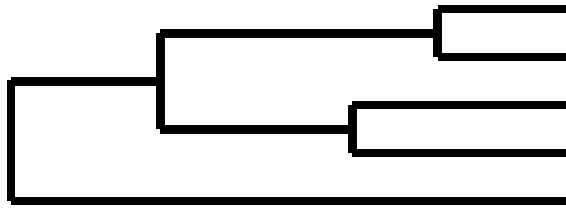
$$d(i,j) \leq d(i,k) = d(j,k)$$

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

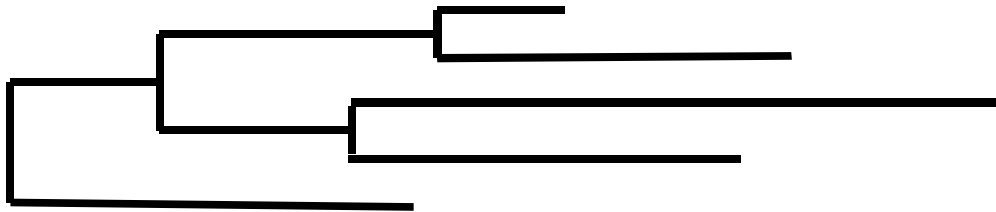


nodes i and j are at same evolutionary distance from k – dendrogram will therefore have ‘aligned’ leafs; i.e. they are all at same distance from root

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).

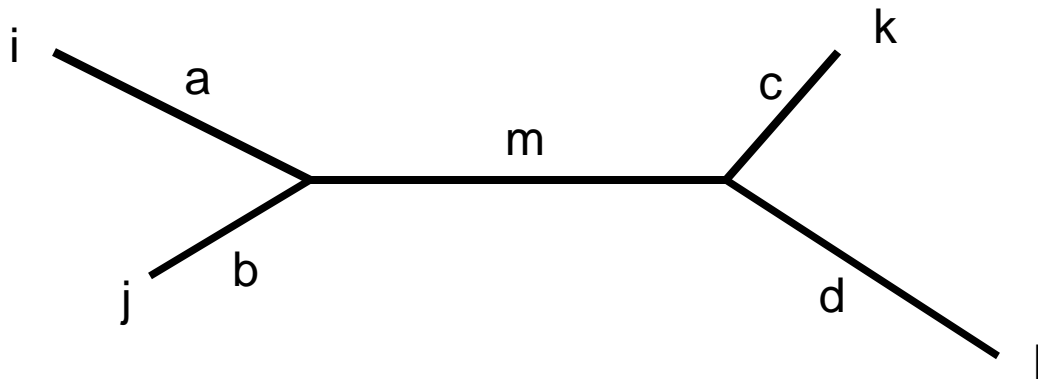
Additive trees

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

Additive trees

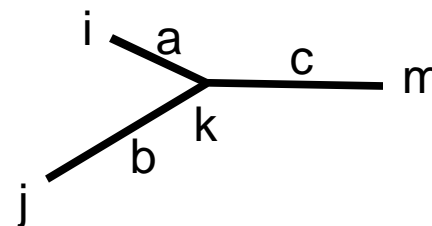
In **additive** trees, the distance between any pair of leaves is the sum of lengths of edges connecting them

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

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

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

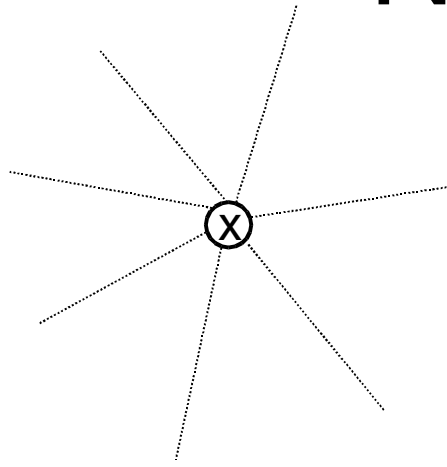


d is ultrametric $\implies d$ additive

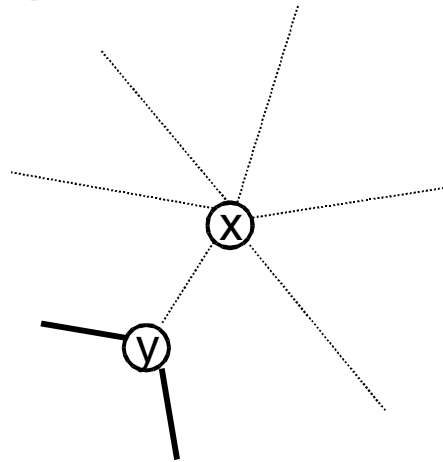
Neighbour-joining (Saitou and Nei, 1987)

- Guaranteed to produce correct tree if distances are additive
- May even produce good tree if distances are not additive
- Global measure – keeps total branch length minimal
- At each step, join two nodes such that distances are minimal (criterion of minimal evolution)
- Agglomerative algorithm
- Leads to **unrooted** tree

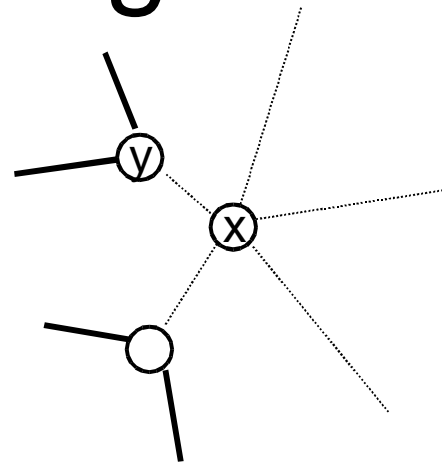
Neighbour joining



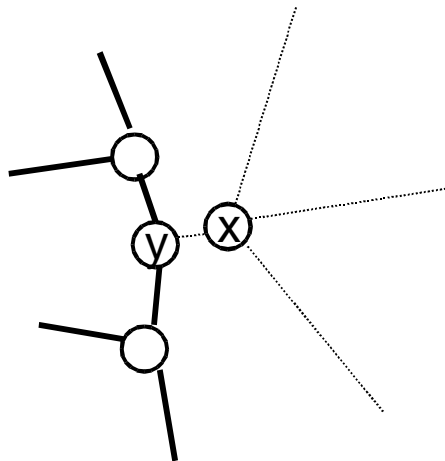
(a)



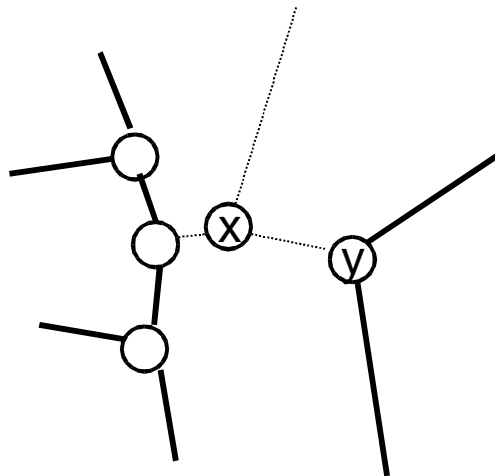
(b)



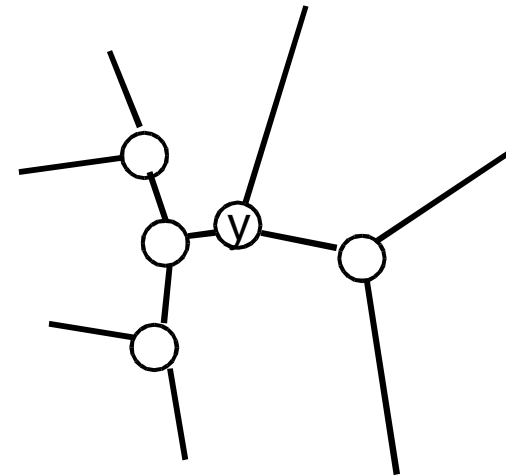
(c)



(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

Finding neighbouring leaves:

Define

$$D_{ij} = d_{ij} - (r_i + r_j)$$

Where

$$r_i = \frac{1}{|L| - 2} \sum_k d_{ik}$$

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

Proof in Durbin book, p. 189

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)
- Define new 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}

Evolution

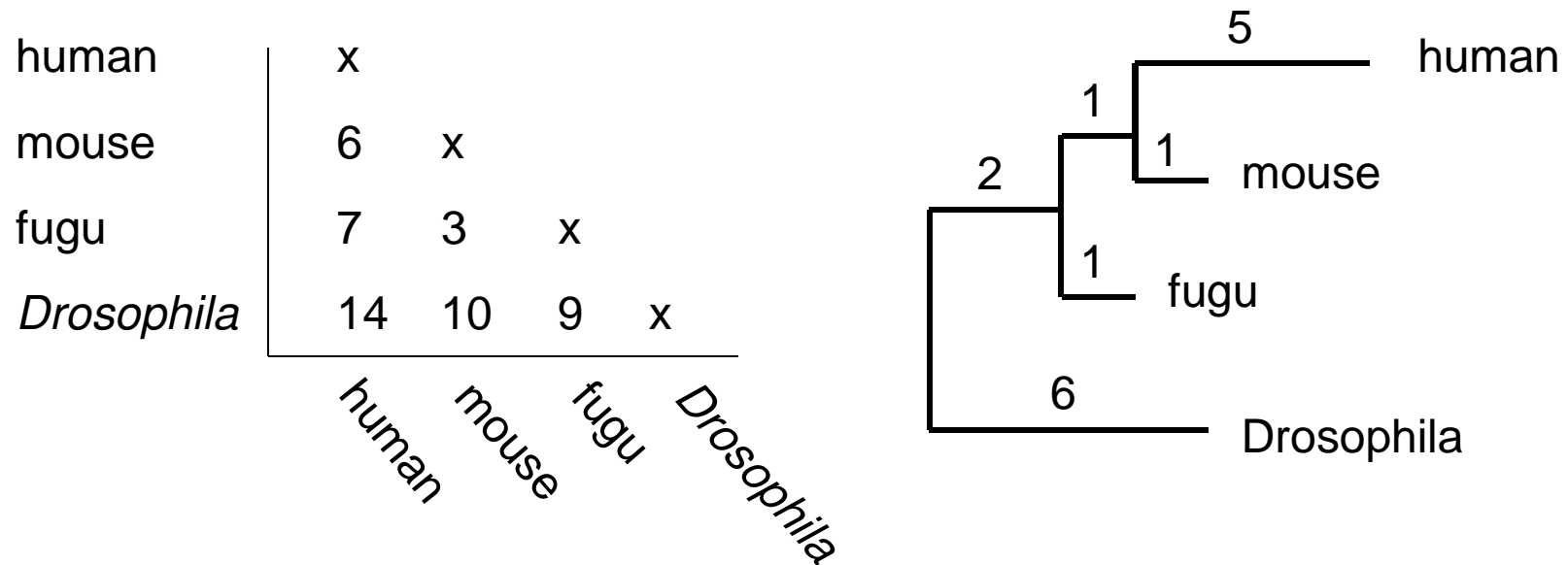
- Most of bioinformatics is comparative biology
- Comparative biology is based upon evolutionary relationships between compared entities
- Evolutionary relationships are normally depicted in a *phylogenetic tree*

Where can phylogeny be used

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

Tree distances

Evolutionary (sequence distance) = sequence dissimilarity



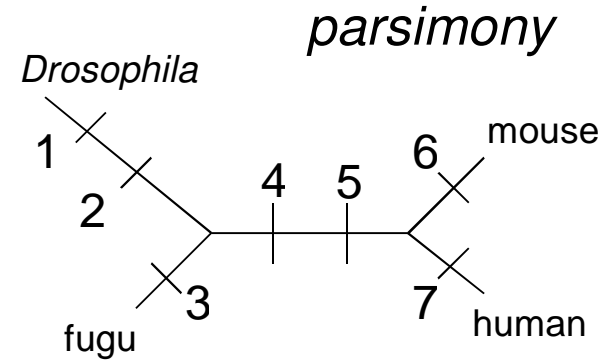
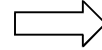
Three main classes of phylogenetic methods

- Distance based
 - uses pairwise distances (see earlier slides)
 - fastest approach
- Parsimony
 - fewest number of evolutionary events (mutations)
 - attempts to construct maximum parsimony tree
- Maximum likelihood
 - $L = \Pr[Data|Tree]$
 - can use more elaborate and detailed evolutionary models

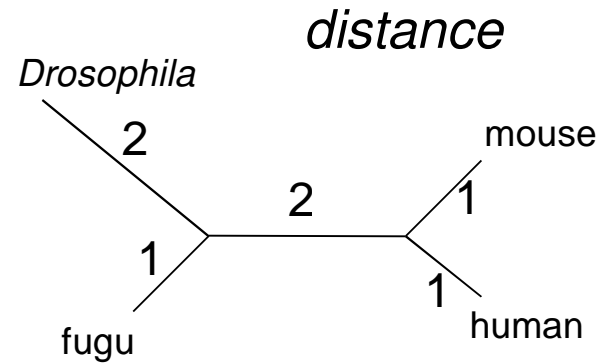
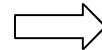
Parsimony & Distance

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



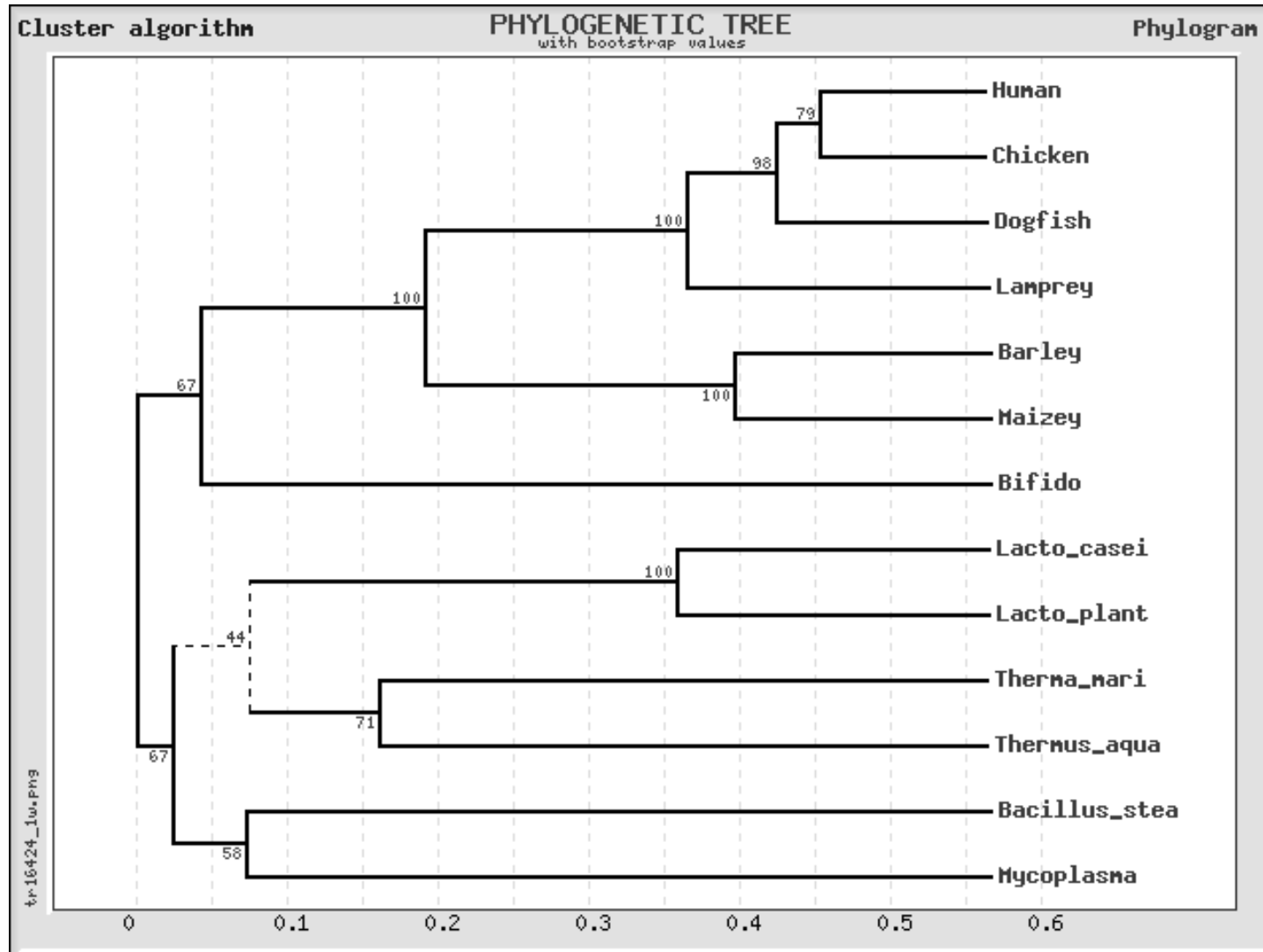
human	x			
mouse	2	x		
fugu	4	4	x	
<i>Drosophila</i>	5	5	3	x
	human	mouse	fugu	<i>Drosophila</i>



Maximum likelihood

- If *data*=alignment, *hypothesis* = tree, and under a given *evolutionary model*, maximum likelihood selects the *hypothesis* (tree) that maximises the observed *data*
- Extremely time consuming method
- We also can test the relative fit to the tree of different models (Huelsenbeck & Rannala, 1997)

How to assess confidence in tree



How to assess confidence in tree

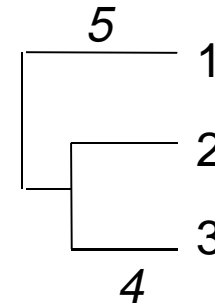
- Distance method – *bootstrap*:
 - Select multiple alignment columns *with replacement*
 - Recalculate tree
 - Compare branches with original (target) tree
 - Repeat 100-1000 times, so calculate 100-1000 different trees
 - How often is branching (point between 3 nodes) preserved for each internal node?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

The Bootstrap -- example

Original

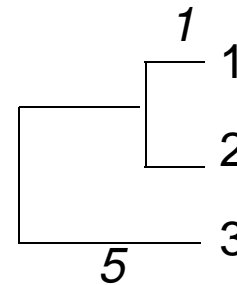
1	2	3	4	5	6	7	8
-	C	V	K	V	I	Y	S
M	A	V	R	-	I	F	S
M	C	L	R	L	L	F	T

2x
3x



Scrambled

3	4	3	8	6	6	8	6
V	K	V	S	I	I	S	I
V	R	V	S	I	I	S	I
L	R	L	T	L	L	T	L



Non-supportive