Phylogeny

- Phylogenetic Trees, Maximum Parsimony, Bootstrapping
- Trees from Distances, Clustering, Neighbor Joining
- Probabilistic methods, Rate matrices
- Models of Sequence Evolution, Maximum Likelihood Trees
- Genome evolution

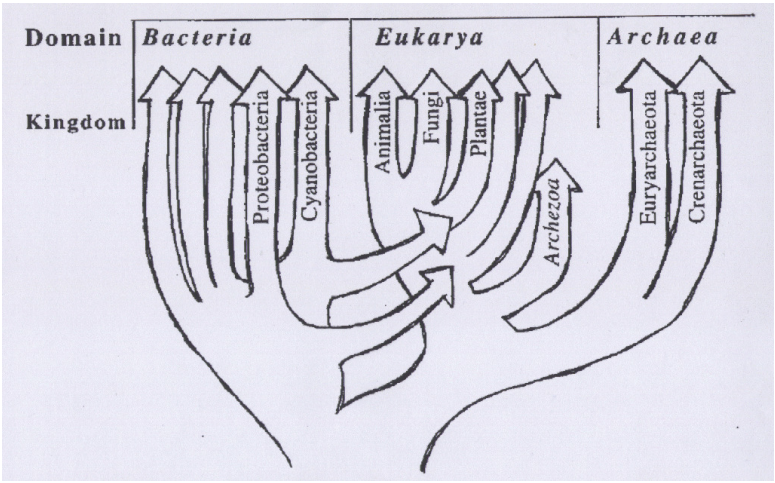
Phylogeny (2)

Recommende sources

- Dan Graur, Weng-Hsiun Li, Fundamentals of Molecular Evolution, Sinauer Associates
- D.W. Mount. Bioinformatics: Sequences and Genome analysis, 2001.
- R. Durbin, S. Eddy, A. Krogh & G. Mitchison, Biological sequence analysis, Cambridge, 1998
- J. Felsenstein: Inferring phylogenies. Sinauer Associates, 2004

Phylogeny, the tree of life

Essential molecular mechanisms as replication and gene expression were found to be similar among the organisms studied so far. This favors the idea that all present day living organisms have evolved from a common ancestor. The relationship is called *phylogeny* and is represented by a *phylogenetic tree*.



(Figure from Doolittle, Science 284, 1999)

Reconstructing molecular phylogenies

Zuckerkandl and Pauling put forward to reconstruct the course of evolution by means of molecular with huge numbers of characters.

E. Zuckerkandl and L. Pauling (1962), Molecular disease, evolution and genetic heterogeneity, In Horizons in Biochemistry, ed. M. Marsha and B. Pullman, Academic Press, pp. 189–225.

```
Euglena_anabaena      AGAATCTGGGTTTGATTCCGGAGAGGGAGCCTGAGAGACGGCTA
Khawkinea_quartana    AGAATCAGGGTTCGATTCCGGAGAGGGAGCCTGAGAAACGGCTA
Phacus_splendens      AGAATTCGGGTTCGATTCCGGAGAGGGAGCCTGAGAAACGGCTA
Phacus_oscillans      AGAATCAGGGTTCGATTCCGGAGAGGGAGCCTGAGAGACGGCTA
Astasia_longa         AGAATCAGGGTTCGATTTCGGAGAGGGAGCCTCAGAGACGGCTA
Lepocinclis_ovum      AGAATCAGGGTTCGATTCCGGAGAGGGAGCCTGAGAGACGGCTA
Peranema_trichophorum AGAATTAGGGTTTGATTCTGGAGAGGGAGCCTGAGAAATGGCTA
```

(A window of an alignment of small subunit rRNA genes)

Molecular Phylogeny (2)

- Traditional methods used morphological characteristics
- Advantages of molecular sequences
  - DNA and amino acid sequences are strictly heritable units
  - Unambigious description of molecular characters and character states
  - Homology assessment is (relatively) easy
  - Easier amenability to mathematical modeling and quantitative analysis
  - Distant evolutionary relationships may be revealed
  - Huge amounts of data available

Molecular Phylogeny (3)

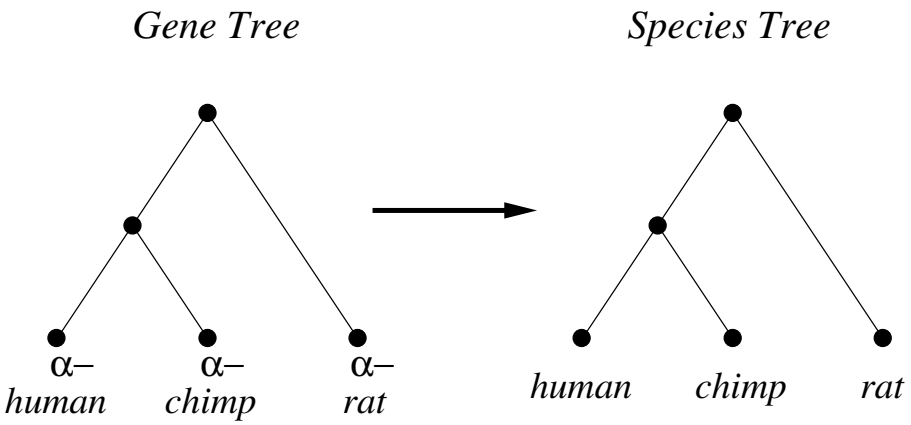
Reconstructed molecular phylogenies are used to

- gain insights into molecular evolution
- predict gene functions
- detect various regimes of selective pressures (pharmacology)
- study the evolution of viruses (epidemiology)

Gene Tree and Species Tree

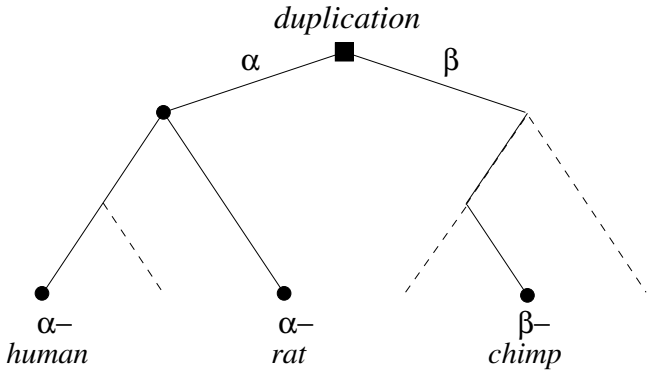
The traditional objective of a phylogenetic tree is to represent the evolutionary relationship between species. In molecular phylogeny, an alignment of homologous genes usually is input into the tree reconstruction. The phylogeny of the species can be transferred from the gene tree, if the genes are *orthologous*.

Consider the evolution of alpha–hemoglobins in human, chimp and rat:



Gene Tree and Species Tree (2)

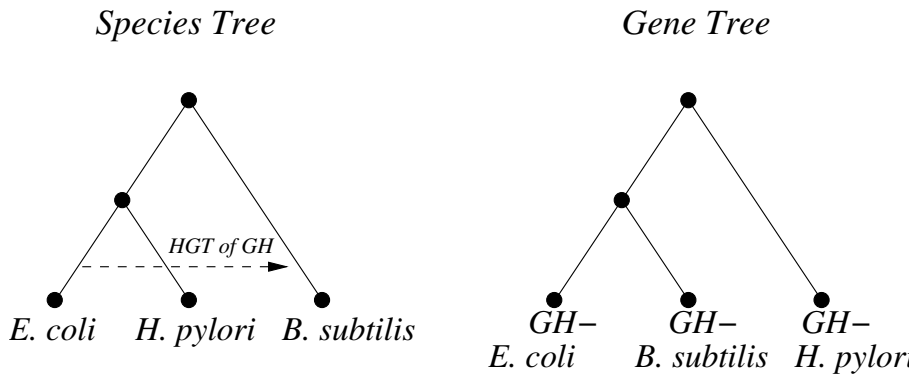
- *Homologous genes* have evolved from a common ancestor
- *Orthologous genes* have evolved from a common ancestor by a speciation event
- *Paralogous genes* have evolved from a common ancestor by a duplication event



β–chimp is paralogous to α–rat (and to α–human) since the least common ancestor of the two genes corresponds to a duplication event.

Gene Tree and Species Tree (3)

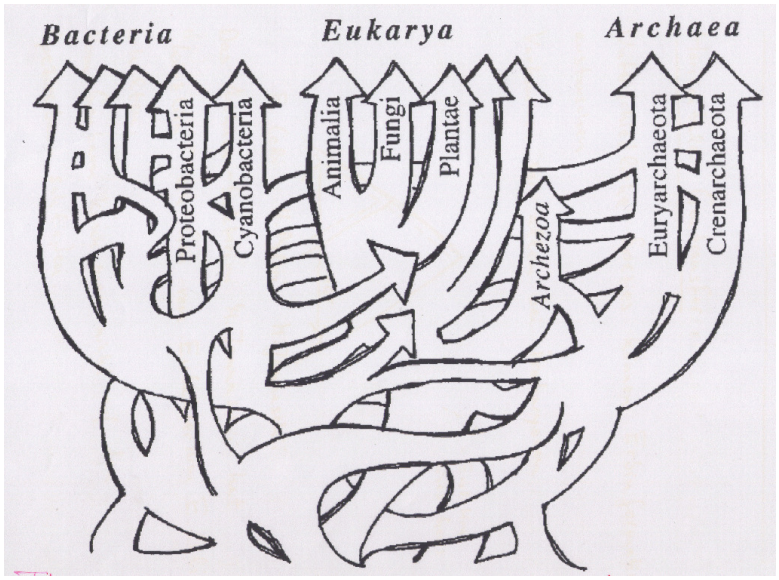
Species may exchange heretic information. This mainly occurs in Prokaryotes and is called *Horizontal Gene Transfer (HGT)* . Consider that a B. subtilis strain recently obtained the gene encoding Glyclosyl Hydrolase (GH) from an E. coli strain.



The gene tree and the species tree are incongruent and it is not possible to infer the species phylogeny based on the gene tree for Glycosyl Hydrolase.

Gene Tree and Species Tree (4)

There is no unique universal phylogenetic tree.

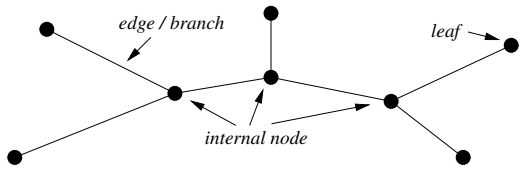


(figure taken from Doolittle, Science 284, 1999)

Basic notions on trees, graphs and trees

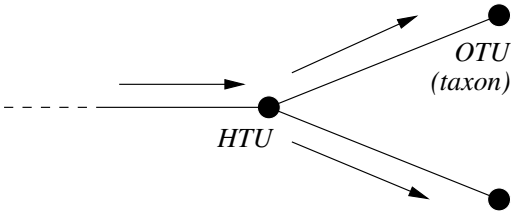
- A *graph* is a pair $G = (V, E)$ consisting of a set of *nodes* (or *vertices*) and a set $E \subseteq V \times V$ of *edges* (or *branches*) that connect nodes. The *degree* of a node $v \in V$ is the number of edges incident to v.
- A *path* is a sequence of nodes $v_1, v_2, \ldots, v_n$ where v_i and v_{i+1} are connected by an edge for all $i = 1, \ldots, n-1$.
- A *cycle* is a simple path in which the first and last vertex are the same. A graph without cycles is called *acyclic*.

- A *tree* is an acyclic graph. Any two nodes of the tree are connected by a unique simple path. A *binary tree* is a tree where the nodes have degree 3 (*iternal nodes*) or degree 1 (*leaves*).



Basic notions, bifurcations

- Phylogeny reconstruction in general yields binary trees. We think of the evolution of a species proceeding in time and speciation events splitting an ancestral species into two new ones. This is reflected by *bifurcations* in the binary tree.



- Internal nodes correspond to hypothetical ancestors. In phylogeny, they are referred to as *HTUs* (hypothetical taxonomic units). Leaves are called *taxa* or *OTUs* (operational taxonomic units). Phylogeny is reconstructed for a set of taxa, which e.g. are given as genes or proteins.

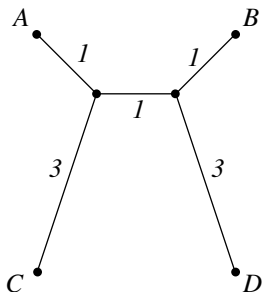
Taxa and clades

Taxa (sing. *taxon*) can be considered groups of organisms that receive a formal taxonic name. Clades are internal nodes of the species trees and comprise a set of taxa/OTUs. Taxa are considered

- *monophyletic* if all have a single ancestor and all its descendents appear in this taxon.
- *polyphyletic* if they have mixed ancestry
- *paraphyletic* if they have a *least common ancestor* that includes other taxa

Basic notions, weighted trees

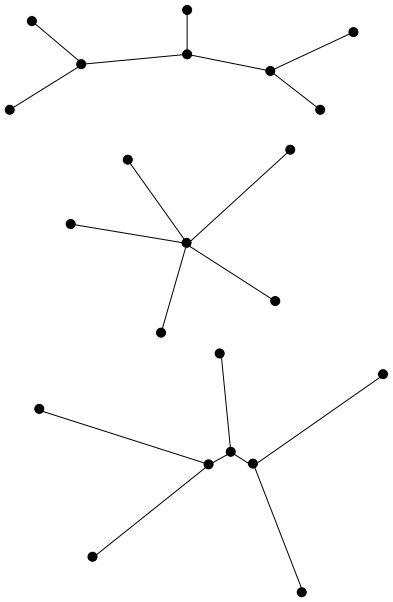
Reconstructed phylogenetic trees are *weighted trees*. That is, each edge is assigned an edge length. Edge lengths represent mutation events which are supposed to have occured on the evolutionary path.



Differences in edge lenghts in the above tree reflect the fact, that the rates at which mutations accumulate in the sequences vary among the lineages to the taxa. This may be due to biological or environmental reasons.

Basic notions, binary tree vs. star tree

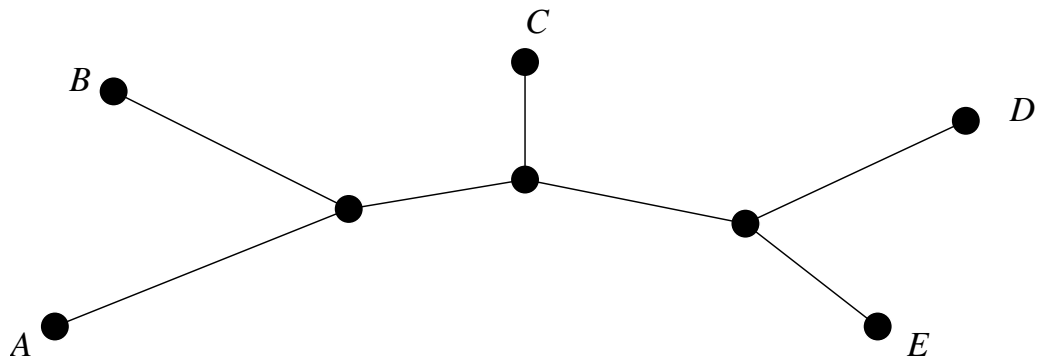
- Binary trees are said to be *fully resolved*. They do not exhibit multifurcations.
- A star tree only has one internal node with a multifurcation (unresolved node, *polytomy*). It is not resolved at all and provides no information about phylogenetic relationships.
- Reconstruction of phylogenies on data with a weak phylogenetic signal sometimes yields fully resolved trees which look starlike.



Basic notions, rooting

Given a binary unrooted tree, one does not know wether an internal node is the ancestor or the descendant of its neighboring internal node.

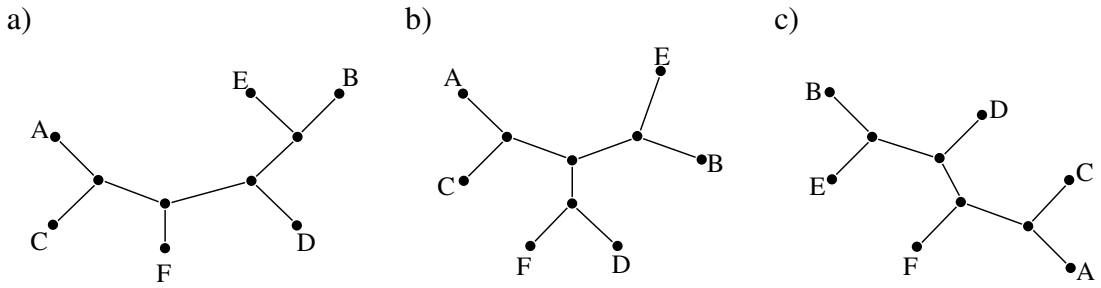
Sometimes it is possible to obtain external information that a certain taxon is more distantly related to the other taxa than the other ones among themselves. Such a taxon is called *outgroup*. Adding a root node to the edge to the outgroup then allows interpreting bifurcations with respect to time.



Note, that an outgroup being too distantly related may lead to incorrect tree reconstructions.

Basic notions, topology and splits

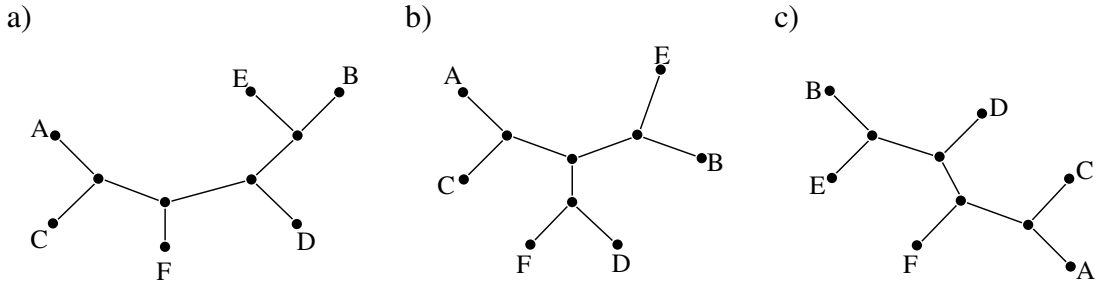
- Two trees showing the same branching pattern are said to have the same *tree topology*.



- The trees shown in a) and c) have the same topology whereas the topology of the tree in b) is different.

Basic notions, topology and splits (2)

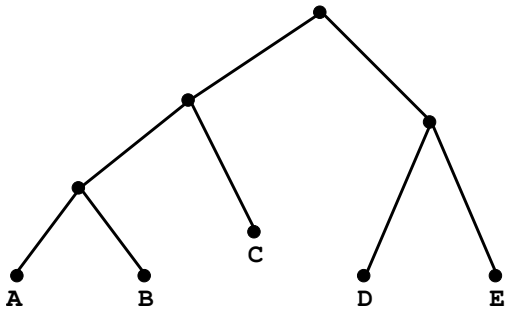
- A *split* (*bipartition*) at an edge partitions the set of taxa into two disjoint sets.



- A split at an edge is phylogenetically informative, if the edge is not connected to a leaf. For the tree in b) the splits $(AC||FDEB), (FD||ACEB), (EB||ACFD)$ are phylogenetically informative.
- The topology of a binary tree is given by its set of phylogenetically informative splits.
- The number of different splits between two trees is a *metric* and can be used to measure the distance between trees (Robinson–Foulds distance)

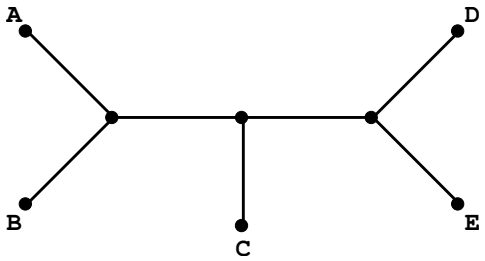
Basic notions, Newick format

Electronically, trees are usually held in a readable text file in the *Newick format*.




```
(((A,B),C),(D,E))
```

The root is represented by the outmost parenthesis. There are many ways to represent unrooted trees.




```
(((A,B),C),(D,E))
((A,B),(C,(D,E)))
(A,(B,(C,(D,E))))
((A,B),C,(D,E))
...
```

Weighted trees

Reconstructed phylogenetic trees normally are *weighted trees*. Each edge is assigned an edge length, representing mutation events which are supposed to have occured on the evolutionary path.rates at which mutations accumulate in the sequences vary among the lineages to the taxa.

In Newick-format, edges are given preceeded by a *colon*, e.g. `((A:3,B:2):3,(C:2,D:3))`

Methods for phylogeny reconstruction ...

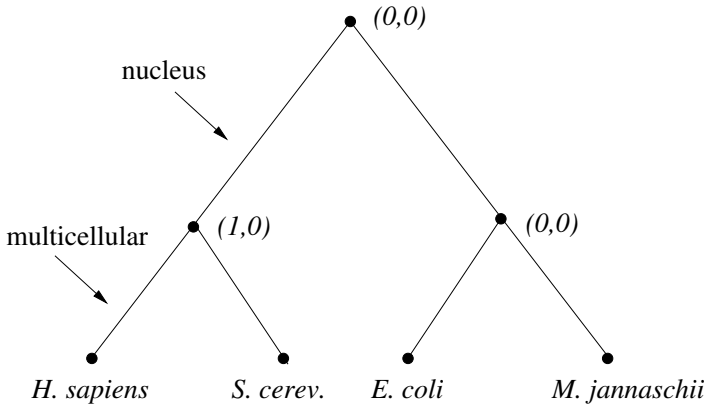
... are classified according to their input data.

1. **Character based methods** take as input a *character state matrix*. Examples for characters are 'number of extremities', 'existence of a backbone, 'nucleotide at a site in a molecular sequence', ...
    - Maximum Parsimony
        - Unweighted (Fitch)
        - Weighted (Sankoff)
    - Probabilistic (Maximum Likelihood (Felsenstein))

2. **Distance based methods** take as input a *distance matrix*, which is obtained by measuring the dissimilarity or the evolutionary distance between the taxa.
   - Hierarchical clustering (UPGMA)
   - Neighbor Joining
   - Least Squares (Fitch–Margoliash)

Character state matrix

	nucleus	multicellular
E. coli	0	0
M. jannaschii	0	0
S. cerevisiae	1	0
H. sapiens	1	1



This tree in accordance with *Ockham's razor*: The best hypothesis is the one requiring the smallest number of assumptions.

Character state matrix (2)

An alignment is a *character state matrix*. The characters are the sites of the alignment, the character states are the nucleotides a taxa holds at a site.

```
Euglena_anabaena      AGAATCTGGGTTTGATTCCGGAGAGGGAGCCTGAGAGACGGCTA
Khawkinea_quartana    AGAATCAGGGTTCGATTCCGGAGAGGGAGCCTGAGAAACGGCTA
Phacus_splendens      AGAATTCGGGTTCGATTCCGGAGAGGGAGCCTGAGAAACGGCTA
Phacus_oscillans      AGAATCAGGGTTCGATTCCGGAGAGGGAGCCTGAGAGACGGCTA
Astasia_longa         AGAATCAGGGTTCGATTTCGGAGAGGGAGCCTCAGAGACGGCTA
Lepocinclis_ovum      AGAATCAGGGTTCGATTCCGGAGAGGGAGCCTGAGAGACGGCTA
Peranema_trichophorum AGAATTAGGGTTTGATTCTGGAGAGGGAGCCTGAGAAATGGCTA
```

The characters, i.e. the alignment columns, are treated (or modeled) independently of each other.

Maximum Parsimony

According to Ockham's razor, *Maximum Parsimony* identifies a tree which can be explained by a minimum number of substitution events.

	1	2	3	4	5	6
a	T	T	G	A	A	T
b	G	T	G	G	C	C
c	C	T	G	A	C	C
d	A	T	T	G	A	T

Consider the above alignment. There are three tree topologies for the four taxa. For each tree topology, we place the sequences of the taxa at its leaves. We are ignorant about sequences at internal nodes (HTUs). But we assign sequences to internal nodes, such that the number of substitutions along the edges which are required to describe the transition from one sequence to another in the tree gets minimal. Among the three topologies, the one(s) which can be explained by the smallest number of substitution events is (are) the *Maximum Parsimony Tree(s).*

(Obtain Maximum Parsimony Tree for above alignment at blackboard. Which sites are phylogenetically informative ?).

Maximum Parsimony, tree length

The *tree length* l_i of a tree T_i is the minimal number of substitutions which is necessary to explain the tree when assigning sequences to internal nodes. In order to identify the Maximum Parsimony Tree, we applied the following procedure:

- For each tree topology T_i
  - $l_i \leftarrow 0$
  - For each alignment column j
    - Assign nucleotides to internal nodes in T_i such that the number of substitutions s_{ij} along the edges is minimal
    - $l_i \leftarrow l_i + s_{ij}$

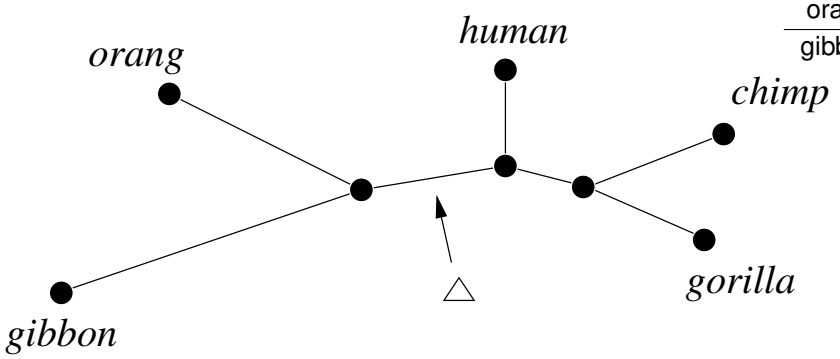
A tree T_i with the smallest tree length l_i is a Maximum Parsimony Tree.

Maximum Parsimony, Fitch algorithm

The algorithm to compute s_{ij} efficiently was proposed by Fitch (1971).

Given a set of taxa, a tree topology i and an alignment column j

human	A
chimp	A
gorilla	C
orang	C
gibbon	G



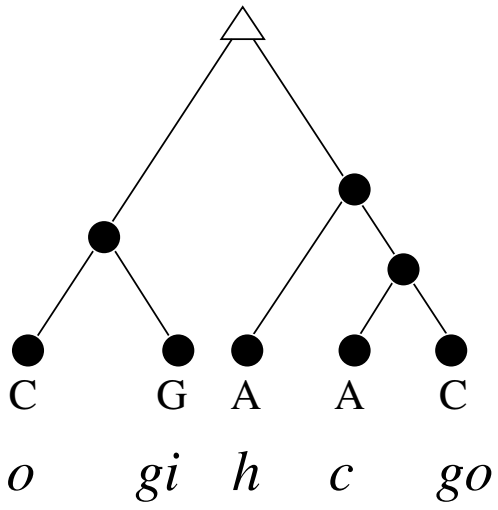
1.) Add a root node to any edge

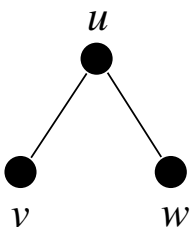
Maximum Parsimony, Fitch algorithm (2)

2.) **bottom-up-pass**

The rooted tree is traversed from the leaves to the root. According to the following rule, sets of nucleotides (character states) are assigned to internal nodes. Say, u is the ancestor of v and w and $\mathcal{U}, \mathcal{V}, \mathcal{W}$ are the respective sets of nucleotides, then set

$$\mathcal{U} = \begin{cases} \mathcal{V} \cup \mathcal{W}, & \text{if } \mathcal{V} \cap \mathcal{W} = \emptyset \\ \mathcal{U} \cap \mathcal{W}, & \text{else} \end{cases}$$





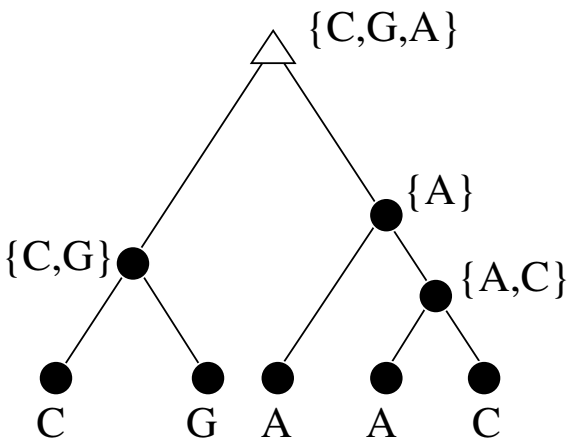
Maximum Parsimony, Fitch algorithm (3)

3.) **top-down-pass**

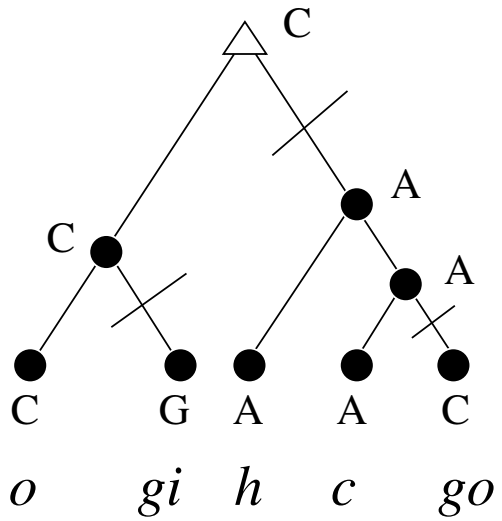
The tree is traversed from the root node to the leaves and the internal nodes are assigned nucleotides according to the following rules

- Assign the root node any nucleotide x out of its set of states $\mathcal{U}_{root}$.
- Assign the child v of parent u the nucleotide

$$\begin{cases} x, & \text{if } x \in \mathcal{U} \\ \text{any nucleotide}, & \text{else} \end{cases}$$



Maximum Parsimony, Fitch algorithm (4)



For the given topology i and the alignment column j, the number of substitutions in the tree is $s_{ij} = 3$. The time complexity of the Fitch algorithm is $O(n)$ where n is the number of taxa.

(Is there a tree topology with fewer substitutions for this column? Consider the taxas' set of character states.)

Weighted Parsimony, Sankoff algorithm

Sankoff (1975) suggests a Dynamic Programming algorithm to compute s_{ij}. The Sankoff algorithm is more general than the Fitch algorithm. For example, it allows to score different changes differently. Further, it is possible to apply Sankoff's algorithm to trees with multifurcations (polytomies) at internal nodes. The algorithm traverses the tree bottom-up from the leaves to the root in a way such that when a node is processed, all its children have already been processed. Each node is assigned to a map with all possible character states λ_i as keys and the tree length l_i of the subtree rooted at this node when assigning it to λ_i as entries. A leaf's map contains the leafs character state as the only key with entry 0.

Weighted Parsimony, Sankoff algorithm (2)

Initialisation: Root the tree T at any internal edge.
Recursion: $\forall v \in T$ compute $l_i(v)$
If v is a leaf node, set $\lambda_i(v) = 0$ if λ_i equals the observed state, else ∞.
If v is an internal node in T . Let $\lambda_i(v)$ be the i-th state of node v and $l_i(v)$ be the length of the subtree rooted at v when assigning λ_i to v.
foreach $\lambda_i(v)$, do

$$l_i(v) = \sum_{w \text{ child of } v} \min_j \{ l_j(w) + C(\lambda_j(w), \lambda_i(v)) \}$$

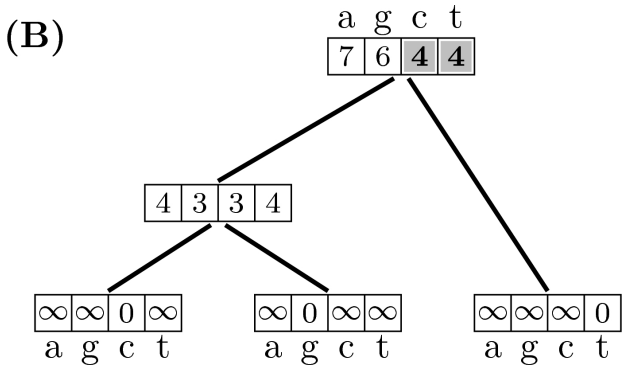
where $C(\lambda_j(w), \lambda_i(v))$ is the cost function for transitions.
Termination The minimal entry in the roots map then is the parsimony tree length.

Example for Sankoff

(A)

C_{ij}	a	g	c	t
a	0	1	3	3
g	1	0	3	3
c	3	3	0	1
t	3	3	1	0

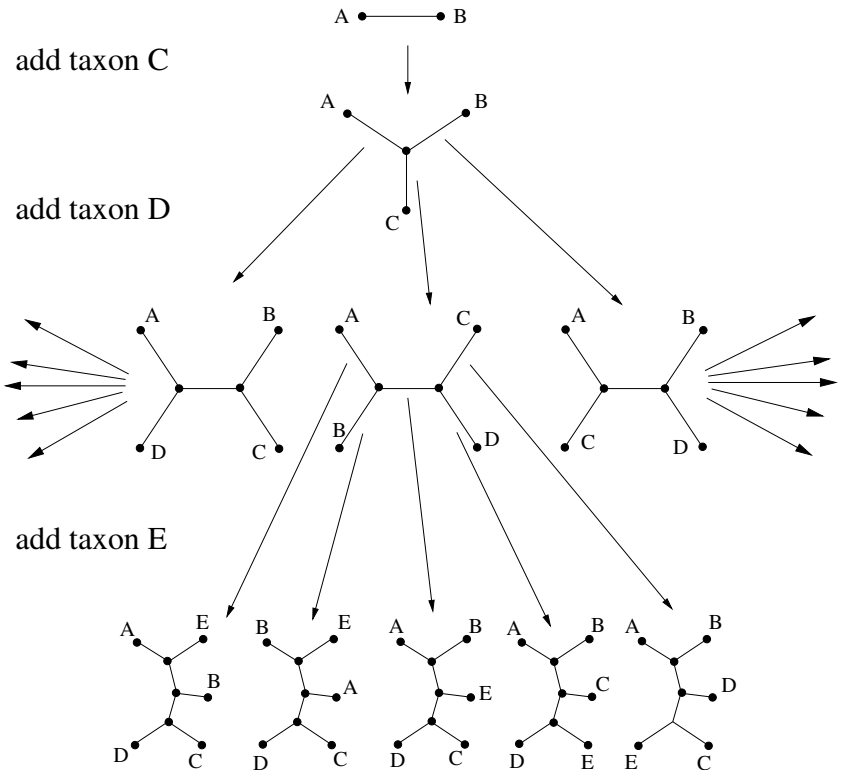
(B)



From Clemente et al. BMC Bioinformatics (2009)

The number of binary trees given n taxa

All possible binary tree topologies for n taxa can be enumerated by the following procedure: We start with a tree containing any two taxa and subsequently add the other taxa to the tree by inserting internal nodes and edges to the taxa.
(Derive formula as exercise or at the blackboard. Hint: the number of edges of a binary tree with n leaves equals $2n-3$)



The number of binary trees given n taxa (2)

The numbers of different unrooted and rooted binary tree topologies U_n and R_n are

$$U_n = \prod_{i=3}^{n}(2i-5), \qquad R_n = \prod_{i=3}^{n+1}(2i-5)$$

where n is the number of taxa.

n	U_n	R_n
2	1	1
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10395
8	10395	135135
9	135135	2027025
10	2027025	34459425

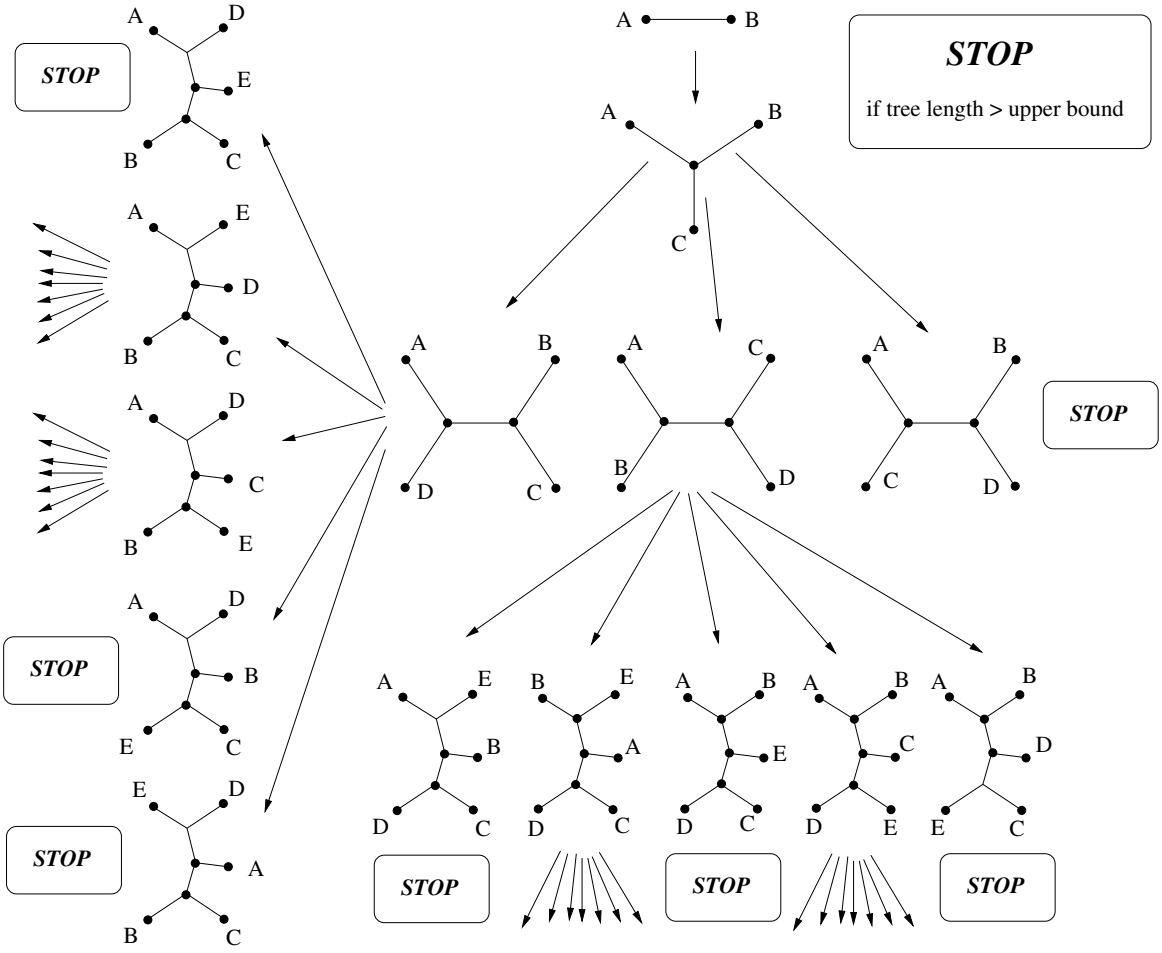
R_{20} = 8 200 794 532 637 891 559 375

Maximum Parsimony, Branch and Bound

The identification of a Maximum Parsimony Tree requires checking the tree lengths for all tree topologies. Recall that the number of tree topologies rapidly increases with the number of taxa. However, for treelike data, the application of a *branch and bound*–strategy (Hendy and Penny 1982) drastically reduces the tree search space and exact solutions for 20 or more taxa are obtained in manageable time.

Concept:

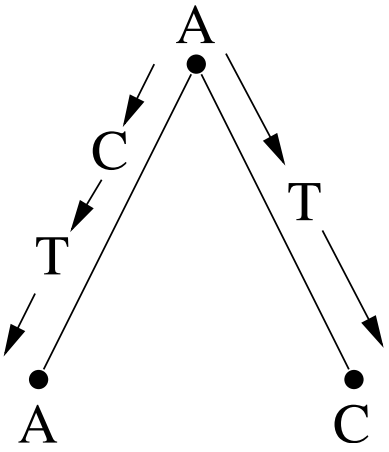
- Obtain an upper bound for the tree length (e.g. by Neighbor Joining)
- Construct all tree topologies by consecutively adding edges and taxa (see above)
- If the tree length of an intermediate tree is larger than the upper bound, the corresponding searching the corresponding subtree is halted.



Maximum Parsimony, miscellaneous

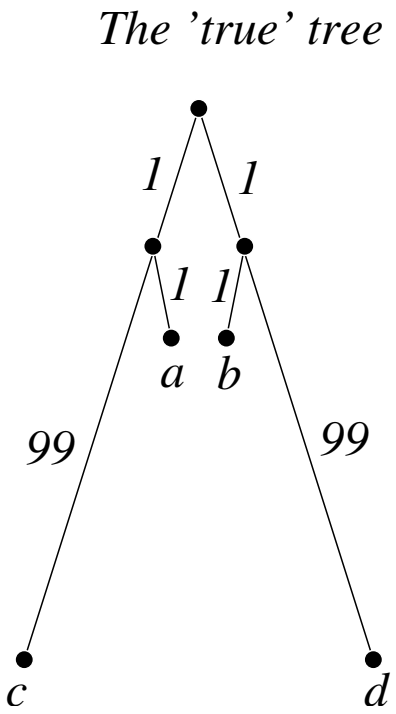
- If branch and bound methods are too slow, heuristic searches are used. Usually an initial tree is obtained, e.g. by Neighbor Joining, and this tree is rearranged (*Branch swapping*).
- Transitions occur more often than transversions. *Weighted parsimony* therefore assigns a larger 'substitution weight' to transversions.

- Maximum Parsimony is widely used. However, Maximum Parsimony does not take into account that the observed character states of taxa being neighbors in the tree may have been multiply mutated. Maximum Parsimony therefore should only be applied to closely related sequences where the chance that multiple substitutions occured is small.



Inconsistency of Maximum Parsimony

An estimation method is consistent, if it approaches the true value of the quantity estimated as more and more amounts of data are available. When reconstructing phylogenies, the estimated quantities are the edge lengths of the tree and the tree topology. Assume, we know that the four sequences representing taxa a, b, c, d have evolved according to the tree shown on the right. The edges to a and b are short whereas the edge lengths to c and d are much larger. In other words, rates of evolution for taxa c and d are relatively high compared to the rates at which taxa a and b evolved.



Inconsistency of Maximum Parsimony (2)

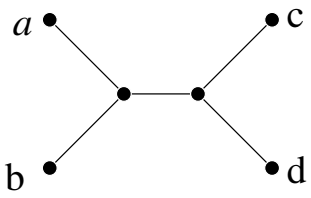
The following sequence family was generated by REFORM (Random Evolutionary FORests MOdel, see `http://www.molgen.mpg.de/ rahmann/`). The root sequence was drawn from the uniform distribution of nucleotides, and the sequences were simulated according to the Jukes–Cantor model (see below) and the tree shown above, where the edge lengths correspond to the expected number of substitutions per 100 sites.

```
a  ATAAAGAGAAATGAGGACTACCCCAGACAAAATACTTAGTCATTAGAGGATGCACGAGAG  |60
b  ATAAAGCGAAAGGAGGAGTACCCCAGACAAAATACTCAGTCATTAGAGGCTGCACGAGAG  |60
c  AGCAAGAACTCGTCACCCTGCCACACACACAAAGCTGTATCGACCAACAAATGTCAAGAA  |60
d  ATAATGTGATTGGGGCTGCGGGGCACTGGACATTCTTCGCCCGCAACTCCAGCACGAGCA  |60
   *  * *   * *   *   *   ***     *  **    *  * *    *   * ** *  |21
            i     i   i     i                i i   ii        i   | 9
s - sites where nucleotides in sequences a and b differ
* - sites with identical nucleotides in c and d
i - phylogenetically informative sites
```

Inconsistency of Maximum Parsimony (3)

Sequences *a* and *b* are well conserved. Only four susbstitutions have accumulated in the sequences on their evolutionary paths. On the other hand, sequences *c* and *d* are very divergent. But even for two random sequences we'd expect $\frac{1}{4}$ of the sites to show up the same nucleotide. In the alignment there are 21 sites holding the same nucleotide in *c* and *d*. With respect to a Maximum Parsimony reconstruction, these sites become the phylogenetically informative ones given that the nucleotides between *a* and *b* are still conserved but different from the nucleotides in *c* and *d*. In the alignment there are 9 informative sites, but only one of them favours the correct topology. The Maximum Parsimony tree therefore has the wrong topology.

Maximum Parsimony tree

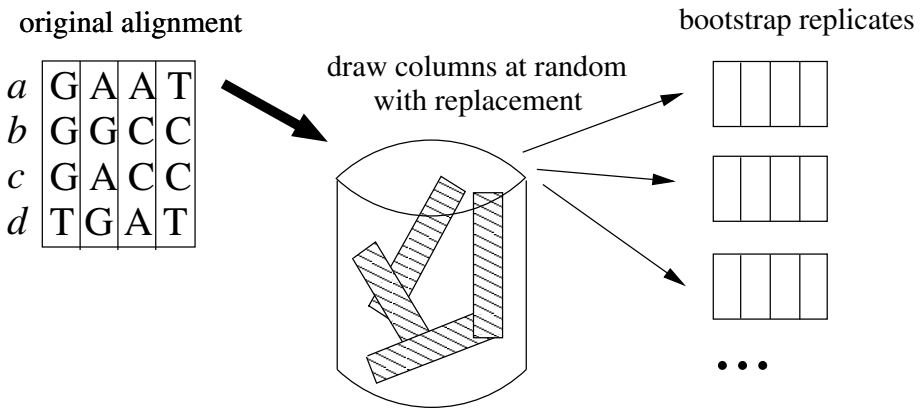


This effect is called *long branch attraction*. A ML estimation finds the correct topology.

Bootstrapping

Non–parametric bootstrapping is the most commonly used method to obtain a quantity telling us something about the uncertainty of tree reconstructions (Felsenstein 1983).

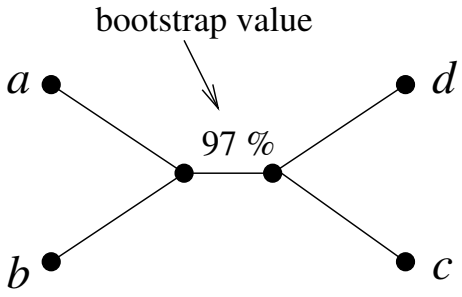
Procedure: The observations or (the sample) we use to estimation the phylogeny are the columns of a multiple alignment. The order of the columns is irrelevant for the outcome of the tree estimate. In one bootstrap simulation step, a new alignment or *bootstrap replicate* is generated by drawing columns from the original alignment at random with replacement. This is repeated until the bootstrap replicate contains as many columns as the original alignment.



Bootstrapping (2)

In this way, n bootstrap replicates are obtained where typical values for n range from $100 - 1000$. The tree estimation is applied to all bootstrap replicates in turn. This yields n bootstrap trees each coming with a set of splits. *Bootstrap values* (or the *bootstrap support*) correspond to the relative frequency at which a split of the tree (estimated on the original alignment) occured in the bootstrap replicates.

The bootstrap applied to the above alignment (section Incosistency of MP) yields the following result for the wrongly reconstructed tree:



Bootstrap values do not provide probabilities for the splits being the correct ones.

Bootstrapping (3)

Given the tree estimation method, the data is 'disturbed' by sampling from the sample and the stability of the outcomes of the method is checked. Bootstrapping provides no information about the reliability of the method.

"... this is not a test of how accurate your tree is; it only gives information about the stability of the tree topology (the branching order), and it helps assess wether the sequence data is adequate to validate the topology." (Berry and Gascuel, 1996)

... given the method.

Trees from distances

Distance based tree building methods rely on a distance measure between sequences resulting in a distance matrix. Distance measures usually take a multiple alignment of the sequences as input. After the distance measure is performed sequence information is not used any more. This is in contrast to character based tree building methods which consider each column of a multiple sequence alignment as a character and which assess the nucleotides or amino acid residues at those sites (the character states) directly.

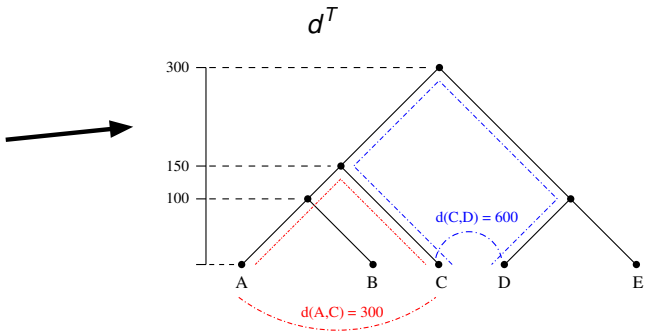
The idea when using distance based tree building methods is that knowledge of the "true evolutionary distances" between homologous sequences should enable us to reconstruct their evolutionary history.

Trees from distances (2)

Suppose the evolutionary distances between members of a taxa set $\{A, B, C, D, E\}$ are given by the following distance matrix

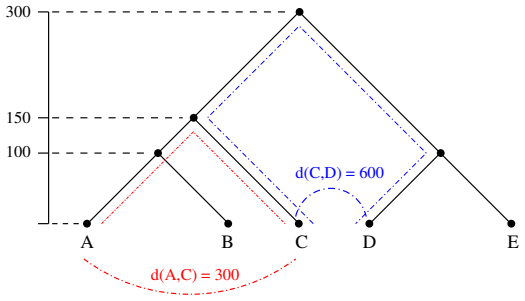
d^M

	A	B	C	D	E
A	0	200	300	600	600
B		0	300	600	600
C			0	600	600
D				0	200
E					0



d^M is given and a tree T with its *path metric* d^T is searched. T is reconstructed algorithmically or by fitting d^T to d^M. For the above tree we see that $d^T = d^M$.

Trees from distances (3)



A tree like this is called a *dendrogram*. The nodes are ranked on the basis of their relative distance to the root on the horizontal axis. The amount of evolution which has accumulated in A and C since divergence from their common ancestor is 150. In other words, the evolutionary distance from A (and C) to the common ancestor of A and C is 150. In general, the sum of edge weights along the path between two nodes corresponds to the evolutionary distance between the two nodes. Deriving distances between leaves is done by summing up edge weights along the path between the leaves. Distances derived in this way from a tree form the *path metric* d^T of the tree.

Basic definitions, metric

Definition: A *metric* on a set of objects O is given by an assignment of a real number d_{ij} (a distance) to each pair $i, j \in O$, where d_{ij} fulfills the following requirements:

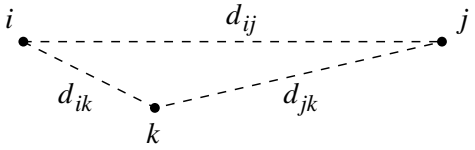
(i) $d_{ij} > 0 \quad for\ i \neq j$

(ii) $d_{ij} = 0 \quad for\ i = j$

(iii) $d_{ij} = d_{ji} \quad \forall\ i, j \in O$

(iv) $d_{ij} \leq d_{ik} + d_{kj} \quad \forall\ i, j, k \in O$

The latter requirement is called the *triangle inequality*

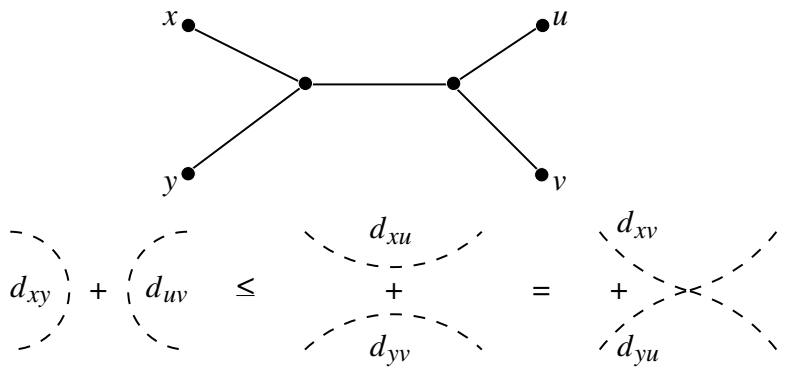


Basic definitions, additive metric (2)

Let d be a metric on O. d is an *additive metric* if it satisfies the *four point condition* (Bunemann 1971).

Four point condition: d is an *additive metric* on O, if any four elements from O can be named x, y, u and v such that

$$d_{xy} + d_{uv} \ \leq\ d_{xu} + d_{yv} \ =\ d_{xv} + d_{yu}.$$



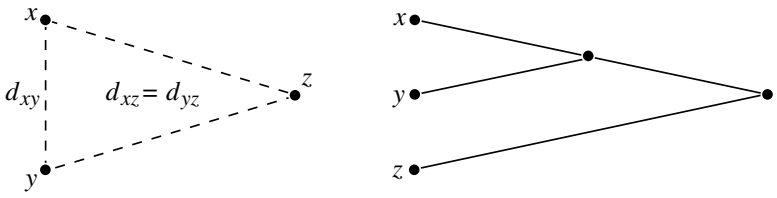
The four point condition is a strengthened version of the triangle inequality. It implies that the path metric of a tree is an additive metric.

Basic definitions, ultrametric (3)

d is an *ultrametric* if it satisfies the *three point condition*.

Three point condition: d is an *ultrametric* on O, if any three elements from O can be named x, y, z such that

$$d_{xy} \leq d_{xz} = d_{yz}.$$



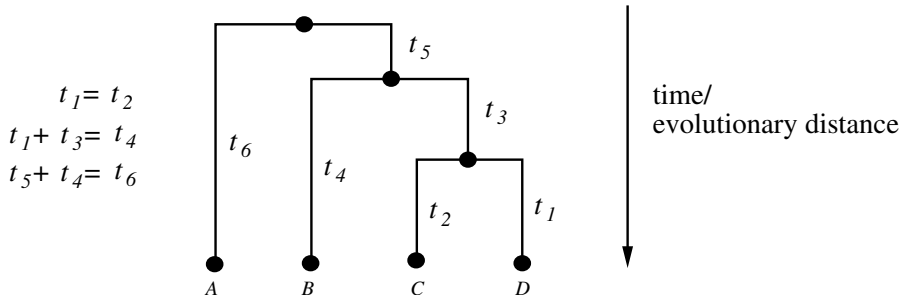
This is an even stronger version of the triangle inequality.

If d is an ultrametric, it is an additive metric.

If d is an additive metric, it is a metric.

Ultrametric trees

A weighted tree is called an *ultrametric tree* if it can be rooted in such a way that the distances from the root to each leaf are equal.



There is a clear interpretation inherent to ultrametric trees: Sequences have evolved from a common ancestor at constant rate (molecular clock hypothesis).

The path metric of an ultrametric tree is an ultrametric. Conversely, if distances d^M between a set of objects form an ultrametric, there is one ultrametric tree T corresponding to the distance measure, that is $d^T = d^M$. Given an ultrametric, this ultrametric tree can easily be reconstructed by one of the agglomerative clustering procedures described below.

UPGMA (Unweighted pair group method using arithmetic averages

Given a set of objects O with n elements and distances $d_{i,j},\ i,j \in O$, initially each object is assigned a singleton cluster. Then the algorithm proceeds as follows:

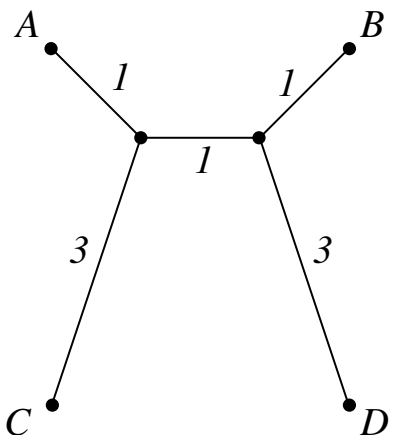
While there is more than one cluster left, do:

1. Find the pair (i, j) with the smallest distance d_{ij} and create a new cluster u that joins clusters i and j.
2. Define the *height* (i.e. distance from leaves) of u to be $l_{ij} := d_{ij}/2$
3. Compute the distance d_{ku} of u to any other cluster: $d_{ku} := \frac{n_i d_{ki} + n_j d_{kj}}{n_i + n_j}$ where n_i is the number of elements in cluster i. d_{ku} is the arithmetic average of the original distances of all elements in k and all elements in u.
4. Remove i, j from the list of objects

UPGMA (2)

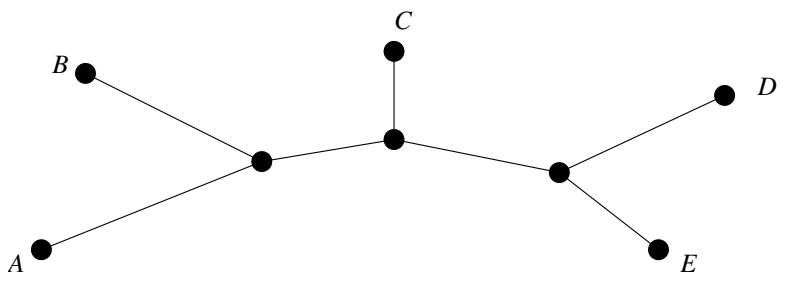
UPGMA was originally developed for phenetics, i.e. for constructing phenograms reflecting phenotypic similarities rather than evolutionary distances.

If the assumption of an approximately constant rate of evolution among the lineages does not hold, UPGMA fails to find the correct topology. Consider that taxa evolved according to the below tree:



Additive trees

An *additive tree* is a weighted binary tree. Rates of evolution vary among species, among gene families, among sites in molecular sequences and generally in the course of evolution. *Additive trees* do not presume a constant evolutionary rate nor do they make any assumption about the rooting and therefore reflect our ignorance as to where the common ancestor lies. Given an additive metric there is exactly one tree topology that allows for realization of an additive tree.



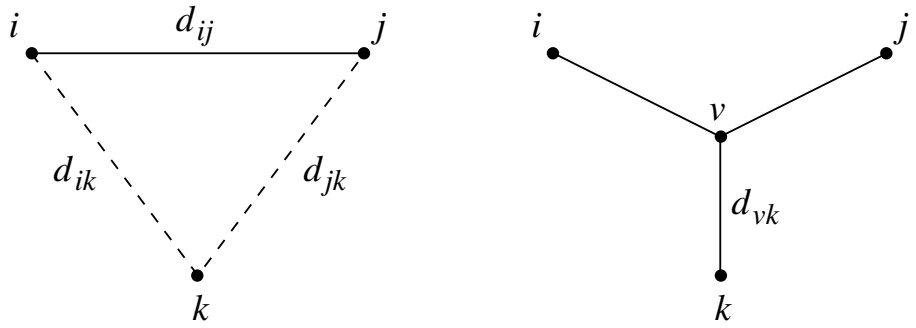
Exact reconstruction of additive trees

An additive metric can be represented as a unique additive tree which can be reconstructed in time complexity $O(n^2)$ (Waterman, Smith, Singh, Beyer, 1977).

The algorithm successively inserts objects into intermediate trees until no objects are left to insert. It makes use of the following rationale:

Given an intermediate tree T' containing leaf i and leaf j, one tests if one can insert an edge connecting leaf k to the intermediate tree along the path connecting i and j. Denote the node connecting i, j and k as v and the weight of the edge being inserted as d_{vk}.

Exact reconstruction of additive trees (2)



$$d_{ik} + d_{jk} = d_{iv} + d_{vk} + d_{jv} + d_{vk} = 2 \cdot d_{vk} + d_{ij}$$

$$d_{vk} = \frac{1}{2}(d_{ik} + d_{jk} - d_{ij})$$

$$\begin{aligned} d_{iv} &= d_{ik} - d_{vk} \\ d_{jv} &= d_{jk} - d_{vk} \end{aligned}$$

(Example tree reconstruction at blackboard.)

Neighbor Joining

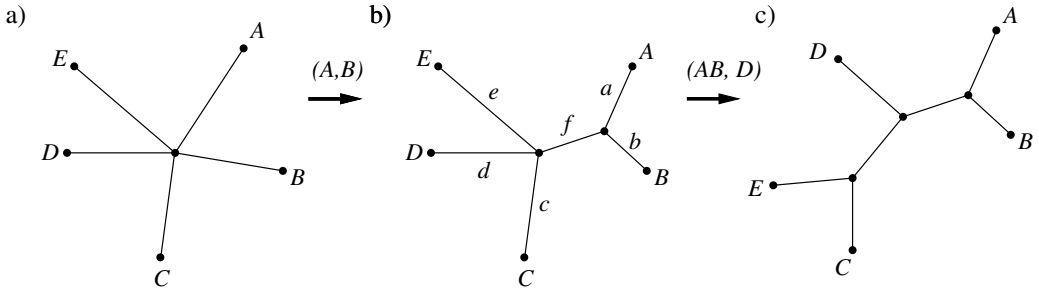
Since distance measures on real data never provide an additive metric, the above algorithm to exactly reconstruct the additive tree given an additive metric is not applicable to real data. The *neighbor joining* method (Saitou, Nei, 1987) is similar to cluster analysis in some ways. The individual taxa are iteratively grouped together, forming larger and larger clusters of taxa. In contrast to UPGMA, neighbor joining does not assume a molecular clock, but it assumes that observed distances are close to an additive metric. Given an additive metric, the neighbor joining method identifies the correct tree and it also correctly reconstructs trees if additivity only holds approximately.

Definition: Two taxa are *neighbors* in a tree if the path between them contains only one node.

As neighbor relationships of nodes in a binary tree uniquely define the tree topology, successively identifying neighbors is a way to reconstrut the tree. The time complexity of the Neighbor Joining algorithm is $O(n^3)$ given n taxa.

Neighbor Joining (2)

The concept to identify neighbors is the following: A star tree is decomposed



such that the tree length is minimized in each step. Consider the above star tree with N leaves shown in a). The star tree corresponds to the assumption that there is no clustering of taxa. In general there is a clustering of taxa and if so, the overall tree length (the sum of all branch lengths) S_F of the true tree or the final NJ tree (see c)) is smaller than the overall tree length of the star tree S_0.

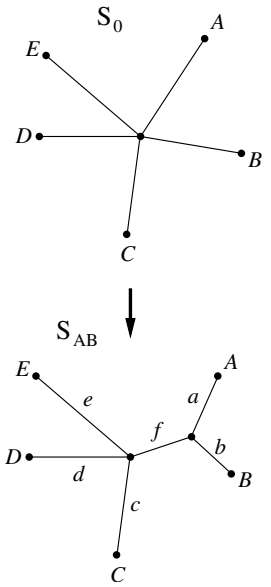
Neighbor Joining (3)

The tree length of the tree with resolved neighbors A and B is

$$S_{ij} = \sum_{\substack{k=1 \\ k\neq i,j}}^{N} \frac{d_{ki} + d_{kj}}{2(N-2)} + \frac{d_{ij}}{2} + \sum_{\substack{k<l \\ k,l\neq i,j}}^{N} \frac{d_{kl}}{N-2}$$

where N is the number of taxa.
Computation of S_{AB} yields

$$\begin{aligned} S_{AB} &= (3a+3b+6f+2c+2d+2e)\cdot\frac{1}{6} \\ &+ \frac{a+b}{2} + (2c+2d+2e)\cdot\frac{1}{3} \\ &= a+b+f+c+d+e \end{aligned}$$



Neighbor Joining (4)

Theorem: Given an additive tree T. O is the set of leaves of T. Values of S_{ij} are computed by means of the path metric d^T. Then $m, n \in O$ are neighbors in T, if $S_{mn} \leq S_{ij} \quad \forall\, i, j \in O$.

Thus, computation of S_{ij} for all pairs of taxa allows identifying neighbors given additive distances.

The neighbors are combined into one composite taxon and the procedure is repeated.

We rewrite S_{ij}:

$$\begin{aligned} S_{ij} &= \frac{1}{2(N-2)} \left(2 \cdot \sum_{\substack{k<l \\ k,l \neq i,j}}^{N} d_{kl} + \sum_{\substack{k=1 \\ k \neq i,j}}^{N} (d_{ki} + d_{kj}) \right) + \frac{d_{ij}}{2} \\ &= \frac{1}{2(N-2)} \left(2 \cdot \sum_{i<j}^{N} d_{ij} - r_i - r_j \right) + \frac{d_{ij}}{2} \end{aligned}$$

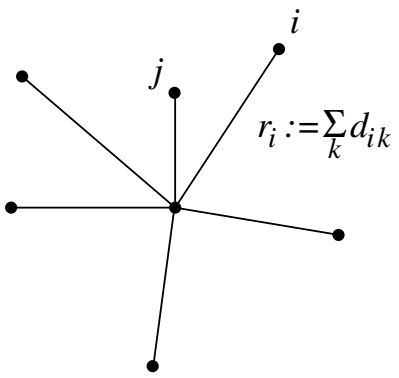
with $r_i := \sum_{k=1}^{N} d_{ik}$.

Neighbor Joining (5)

Since the sum $\sum_{i<j}^{N} d_{ij}$ is the same for all pairs of i and j, we can replace S_{ij} by

$$M_{ij} := d_{ij} - \frac{r_i + r_j}{N-2}$$

for the purpose of easier computation of relative values of S_{ij}.
r_i is also called *net divergence*.
$\frac{r_i+r_j}{N-2}$ holds averaged distances of i and j to all other leaves. Thus, if i and j were neighbors in evolution and i or j evolved fast such that d_{ij} is large, $\frac{r_i+r_j}{N-2}$ is also large and M_{ij} gets small.



Neighbor Joining (6)

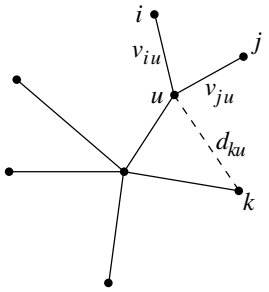
Algorithm: Given distances d_{ij} between members of a set O of N objects. Represent the objects as terminal nodes in a starlike tree:

1. For each terminal node i compute
$$r_i := \sum_{k=1}^{N} d_{ik}.$$
2. For all pairs of terminal nodes (i, j) compute
$$M_{ij} := d_{ij} - \frac{r_i + r_j}{N-2}.$$
Let (i, j) be a pair with minimal value M_{ij} for $i \neq j$.
3. Join nodes i and j into a new terminal node u. The branch lengths from u to i and j are:
$$v_{iu} = \frac{d_{ij}}{2} + \frac{r_i - r_j}{2N-4} \quad \text{and} \quad v_{ju} = d_{ij} - v_{iu}.$$

Neighbor Joining (7)

4. Obtain the distances from u to another terminal node k by

$$d_{ku} = \frac{d_{ik} + d_{jk} - d_{ij}}{2}.$$



5. Delete i and j from the set of objects. If there are more than two clusters left, continue with Step 1

Neighbor Joining (8)

- NJ is fast ($O(n^3)$) and therefore it is suited to be applied to large data sets
- takes rate differences into account
- makes use of distance measure and its model

- result is one tree ($\rightarrow$ Bootstrapping)
- reduction of sequence information
- no objective function

Least Squares on Distances

The problem addressed in reconstructing trees on distances is to find a tree T with path metric d^T on measured distances d^M. This problem can be divided into identifying the topology and reconstructing the edge lengths. Neighbor Joining solves the problem algorithmically and all at once.

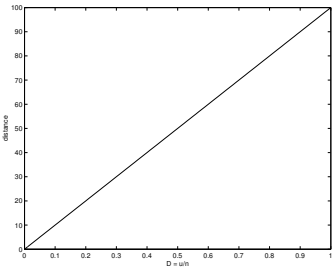
Given a tree topology, Fitch and Margoliash (1967) apply an objective function to fit d^T to d^M. They define the disagreement between a tree and the distance measure by the sum of squared weighted differences in distances:

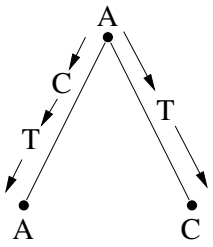
$$E := \sum_{i<j} |d_{ij}^T - d_{ij}^M|^2 \cdot \frac{1}{(d_{ij}^M)^2}$$

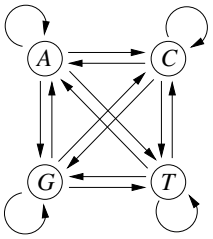
The weights take into account relative uncertanties in the distance measures and may be adapted. d^T is obtained by minimizing E.

How to obtain distances?

We are given a multiple alignment and want to obtain pairwise evolutionary distances in units of substitutions per 100 sites presumably having occured on the evolutionary paths. A simple estimator for a pairwise distance is the fraction of observed mismatches u in the alignment and the alignment length n, $D = \frac{100 \cdot u}{n}$. Clearly, the linear estimator does not give us the quantity of interest. The issue of multiple substitutions was already discussed in the Maximum Parsimony section. The effect of long branch attraction was demonstrated by a tree which was simulated acc according to a tree and the probabilistic Jukes–Cantor model. In the following probabilistic modeling of sequence evolution are discussed and it is shown how to estimate evolutionary distances and phylogenetic trees in a Maximum Likelihood framework.

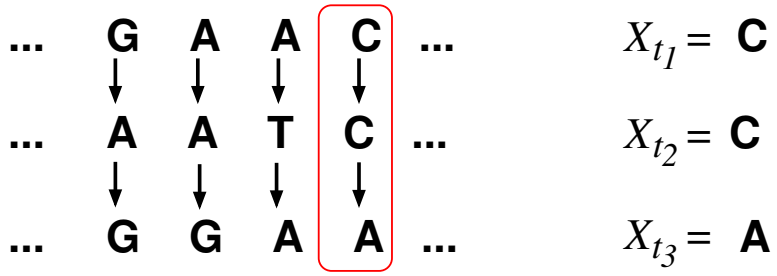






The Markov model of sequence evolution

Sequence evolution is modeled by a *Markov process* acting independently on the sites of the sequence.

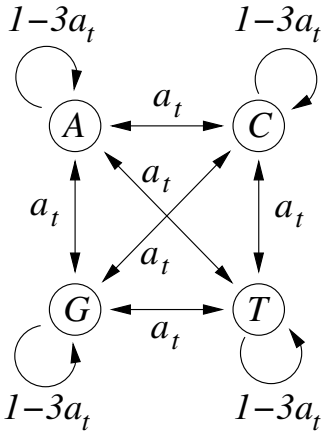


A (time–homogeneous) *Markov process* is a sequence of random variables $(X_t)_{t \geq 0}$ given by a triple $\left(\mathcal{A}, \pi^0, P(t)\right)$, where $\mathcal{A} = \{1, ..., n\}$ is the set of states (nucleotides or amino acid residues) (X_t) takes, π^0 is the initial distribution of states ($\pi_i^0 = Pr[X_0 = i]$) and $P(t)$ is a $n \times n$ matrix of transition probabilities satisfying

$P_{ij}(t) = Pr[X_{t+s} = j | X_s = i] = Pr[X_t = j | X_0 = i]$

The transition probability matrix

For nucleotides, the simplest model is the *Jukes–Cantor–model* (1969). The set of states comprises the nucleotides ($\mathcal{A} = \{1,2,3,4\}$), the distribution of nucleotides is assumed to be uniform ($\pi = (\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4})$) and the probabilities that any nucleotide is substituted by another any other nucleotide are equal.
Thus, the *transition probability matrix* of the Jukes–Cantor model has the form



$$P(t) = \begin{pmatrix} 1-3a_t & a_t & a_t & a_t \\ a_t & 1-3a_t & a_t & a_t \\ a_t & a_t & 1-3a_t & a_t \\ a_t & a_t & a_t & 1-3a_t \end{pmatrix}$$

How is $P(t)$ computed ?

The transition probability matrix (2)

The transition probability matrix $P(t)$ is a stochastic matrix and has the following properties:

- $P(0) = I$, I - identity matrix,
- $P_{ij}(t) \geq 0$ and $\sum_j P_{ij}(t) = 1$,
- $P(s+t) = P(s)P(t)$

The latter equation is called *Chapman–Kolmogorov equation*. E.g. think of $\mathcal{A} = \{1,2,3,4\}$ and the process being in state 1 reaching state t in time $s+t$. The transition probability $P_{14}(s+t)$ is

$$\begin{aligned} Pr[X_{s+t} = 4 | X_0 = 1] = \; & Pr[X_s = 1 | X_0 = 1] \cdot Pr[X_{s+t} = 4 | X_s = 1] \\ & + Pr[X_s = 2 | X_0 = 1] \cdot Pr[X_{s+t} = 4 | X_s = 2] \\ & + Pr[X_s = 3 | X_0 = 1] \cdot Pr[X_{s+t} = 4 | X_s = 3] \\ & + Pr[X_s = 4 | X_0 = 1] \cdot Pr[X_{s+t} = 4 | X_s = 4] \\ = \; & \sum_{k \in \mathcal{A}} P_{1k}(s) P_{k4}(t) \end{aligned}$$

The rate matrix

We assume that the probability transition matrix $P(t)$ of a time continuous Markov chain is continuous and differentiable at any $t > 0$. I.e. the limit

$$Q := \lim_{t \searrow 0} \frac{P(t) - I}{t}$$

exists. Q is known as the *rate matrix* or the *generator* of the Markov chain. For very small time periods $h > 0$, transition probabilities are approximated by

$$\begin{aligned} P(h) &\approx I + hQ \\ P_{ij}(h) &\approx Q_{ij} \cdot h, \qquad i \neq j. \end{aligned}$$

From the last equation we see, that the entries of Q may be interpreted as substitution rate.

The rate matrix (2)

From the Chapman-Kolmogorov equation we get

$$\begin{aligned} \frac{d}{dt}P(t) &= \lim_{h \searrow 0} \frac{P(t+h) - P(t)}{h} \\ &= \lim_{h \searrow 0} \frac{P(t)P(h) - P(t)I}{h} \\ &= P(t) \lim_{h \searrow 0} \frac{P(h) - P(0)}{h} \\ \frac{d}{dt}P(t) &= P(t)Q = QP(t). \end{aligned}$$

Under the initial condition $P(0) = I$ the differential equation can be solved and yields (as in the one–dimensional case)

$$P(t) = \exp(tQ) = \sum_{k=0}^{\infty} \frac{Q^k t^k}{k!}.$$

Transition probabilities for any $t > 0$ are computed from the matrix Q.

The rate matrix (3)

Recall, that for very small h we have $P(h) \approx I + hQ$.

Q has the following properties:

- $P_{ij}(h) \geq 0 \;\Rightarrow\; Q_{ij} \geq 0 \quad \text{for} \quad i \neq j$
- $\sum_j P_{ij}(h) = 1 \text{ and } Q_{ij} \geq 0,\ i \neq j \;\Rightarrow\; Q_{ii} \leq 0$

- $1 = \sum_j P_{ij}(h) = 1 + \sum_j Q_{ij} \Rightarrow \sum_j Q_{ij} = 0,\ Q_{ii} = -\sum_{j \neq i} Q_{ij}$

The rate matrix (4)

The rate matrix of the Jukes–Cantor model is

$$Q = \begin{pmatrix} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha \end{pmatrix}.$$

where $\alpha \geq 0$.

Due to the simple structure of Q, $\exp(tQ)$ can be calculated explicitly. The transition probability matrix is

$$P(t) = \begin{pmatrix} 1-3a_t & a_t & a_t & a_t \\ a_t & 1-3a_t & a_t & a_t \\ a_t & a_t & 1-3a_t & a_t \\ a_t & a_t & a_t & 1-3a_t \end{pmatrix},$$

where

$$a_t = \frac{1 - \exp(-4\alpha t)}{4}$$

Stationarity and reversibility

A (probability) distribution of states π is called *stationary*, if the probability to observe a certain state remains the same for all time points, i.e.

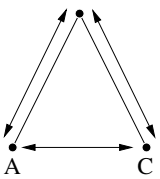
$$\pi_j = \sum_{i \in \mathcal{A}} \pi_i P_{ij}(t) \quad \text{and} \quad \pi = \pi P(t) \quad \forall\ t.$$

For very small h and stationarity,

$$\pi_j = \sum_i \pi_i P_{ij}(h) = \pi_j + \sum_i \pi_i Q_{ij} \quad \Rightarrow \quad \sum_i \pi_i Q_{ij} = 0 \quad \text{and} \quad \pi Q = 0.$$

That is, if we are given Q, we are given all model parameters.

In general, π is not uniform. Modeling the evolution of state i reaching state j in time t by the same process as state j reaching state i in time t is called *reversibility*:
$\pi_i P_{ij}(t) = \pi_j P_{ji}(t)$ (*detailed balance*)



The stationary distribution of a Markov process which obeyes detailed balance is also called the *equilibrium distribution*. When tree estimation under a reversible model is concerned, reversibility implies that we are ignorant about the root position.

Calibrating the rate matrix

If we are given Q, we are given all model parameters. All of the above equations hold, if Q is multiplied with a constant. Explicit model parameters fix the speed of the process. In other words: The model parameters hold substitution rates. And rates hold the information how many substitutions per time unit one expects.

The rate matrix is calibrated to *PEM (percent of expected mutations)*–units. 1 PEM is the time (or evolutionary distance) where one substitution event per 100 sites is expected to have occured.

Given Q, one expects $E = \sum_i \pi_i \sum_{j \neq i} Q_{ij} = -\sum_i \pi_i Q_{ii}$ substitution events per time unit.

The Jukes–Cantor rate matrix Q is calibrated to PEM–units by setting $E = \frac{1}{100} \Leftrightarrow -4 \cdot \frac{1}{4} \cdot -3\alpha = \frac{1}{100} \Leftrightarrow \alpha = \frac{1}{300}$.

Evolutionary Markov Process (X_t) with stationary distribution π (π–EMP)

Müller, Rahmann and Vingron have summarized the properties of a Markov process being suited to describe the substitution process at a site of a molecular sequence. A π–EMP has the following properties:

- (X_t) is time homogeneous.
  $P_{ij}(t) = \text{Prob}[X_{s+t} = j | X_s = i] = \text{Prob}[X_t = j | X_0 = i]$.
- (X_t) is stationary w.r.t. π.
  $\pi_j = \sum_i \pi_i P_{ij}(t),\ \pi = \pi P(t)\ \ \forall\ t$.
- (X_t) is reversible. $\pi_i P_{ij}(t) = \pi_j P_{ji}(t)$.
- (X_t) is calibrated to 1 PEM, the evolutionary distance where one substitution event per 100 sites is expected to have occured.

The Jukes–Cantor correction

The linear estimator $D = \frac{100 \cdot u}{n}$ (u-mismatches, n- sequence length) for the distance between two DNA sequences is ignorant about the putative occurence of multiple substitutions. The Jukes–Cantor correction provides a formula for the evolutionary distance d of two DNA sequences, i.e. d holds the number of substitutions which are expected to have occured per 100 sites given the observed mismatches. (d is measured in time units of the Markov process).

The probability p to observe that a nucleotide is not substituted after time t is

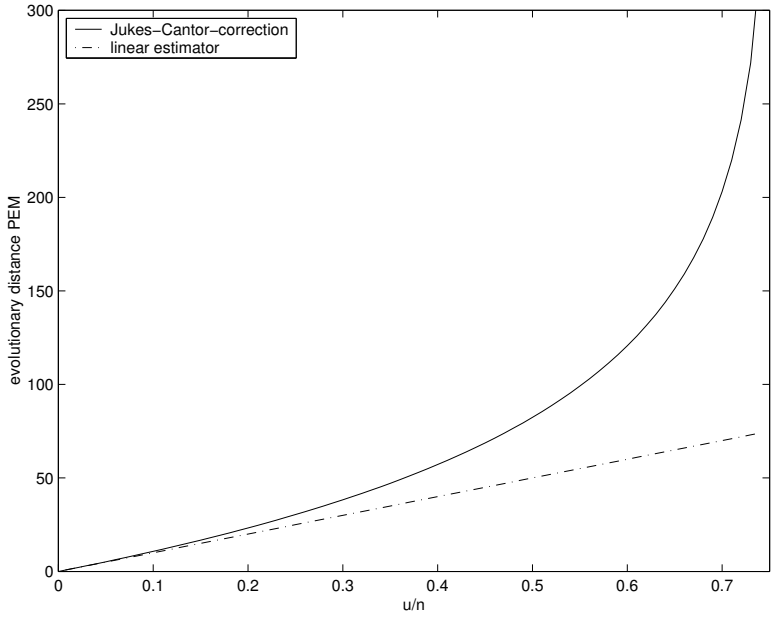
$$p = \sum_i \pi_i P_{ii}(t) = 4 \cdot \tfrac{1}{4}(1 - 3a_t) = 1 - \tfrac{3}{4}(1 - \exp(-4\alpha t)) = \tfrac{1+3\exp(-4\alpha t)}{4}$$

There are u mismatches among n sites. That is, we observe $p = 1 - \frac{u}{n}$. Calibration to PEM–units and setting $t = d$ yields

$$1 - \frac{u}{n} = \frac{1 + 3\exp(-4d/300)}{4}$$

Jukes–Cantor correction (2)

$$d = -\frac{300}{4} \ln\left(1 - \frac{4u}{3n}\right) \text{ PEM}$$



If $\frac{u}{n} \geq 0.75$, d is not defined.

More models

The assumptions of the Jukes–Cantor model for the evolution of a DNA sequence are simplistic. Each substitution of any nucleotide to any other nucleotide is supposed to have the same chance and the distribution of nucleotides π is assumed to be uniform.

Transitions ($A \leftrightarrow G$ and $C \leftrightarrow T$) are more frequently observed than transversions. The *Kimura 2-parameter model* takes that into account by introducing different rates for transitions and transversions and the *Felsenstein 81 model* incorporates non–uniform nucleotide distributions.

A parametrization of the rate matrix is not suited when modeling protein evolution. The 20 amino acid residues come with a variety of replacement frequencies. Dayhoff (1978) estimated the famous 1-step (PAM) transition matrix of a time–discrete Markov chain on alignments of closely related sequences. Müller and Vingron (2000) estimate their model on alignments of varying degree of divergence. They present an iterative approach cycling between estimating evolutionary distances of sequences in an alignment and updating the current rate matrix Q.

Maximum Likelihood (ML) estimation of evolutionary distances

ML estimation maximizes $Pr(\text{data}|\text{parameters})$, the probability of the data given the parameters. The probabilities of all sets of data must add up to one. When holding the data constant and varying in parameters, this doesn't hold and $Pr(\text{data}|\text{parameters})$ is called the *likelihood*.

Consider the following alignment:

A	G	C
A	L	A

The probability of observing the alignment $\mathcal{D}$ (the data) with distance t given the Markov model $\mathcal{M}$ is

$$\Pr(\mathcal{D}|t,\mathcal{M}) = \pi_A P_{AA}(t) \cdot \pi_G P_{GL}(t) \cdot \pi_R P_{CA}(t).$$

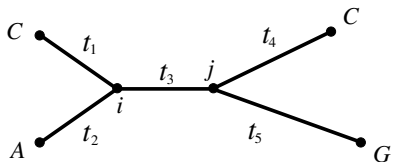
Consider $\Pr(\mathcal{D}|t,\mathcal{M})$ as likelihood function depending on distance t:

$$\log \mathcal{L}(t) = \log \Pr(\mathcal{D}|t,\mathcal{M})$$

The evolutionary distance is estimated as distance $\hat{t}$ maximizing the likelihood function.

Maximum likelihood trees

Consider one site of an alignment holding the states $A, C, G, T \in \mathcal{A}$. Think of a given tree topology $\mathcal{T}$. This allows labeling the edges of the tree in the below figure with the corresponding edge lengths $t_1, ..., t_5$ and labeling the leaves with states at its leaves.



First assume that states at internal nodes $i, j \in \mathcal{A}$ are known. We choose one of the nodes as root node, e.g. the one with state i. The probability $Pr(ACGTij)$ to observe all states in the above tree is obtained by evolving the state of the root node according to the tree under the model:

$$Pr(ACGTij) = \pi_i P_{iC}(t_1) P_{iA}(t_2) P_{ij}(t3) P_{jC}(t_4) P_{jG}(t_5)$$

Note that due to the reversibility of the model $Pr(ACGTij)$ does not depend on the choice of the root node.

Maximum Likelihood trees (2)

Due to our ignorance about states at internal nodes, we sum over each state at the internal nodes and obtain the probability $Pr(\mathcal{D}_{site}|t_1,...,t_5,\mathcal{T},\mathcal{M})$ to observe the site given $\mathcal{T}$, edge lengths $t_1, ..., t_5$ and the model $\mathcal{M}$:

$$Pr(\mathcal{D}_{site}|t_1,...,t_5,\mathcal{T},\mathcal{M}) = \sum_{i\in\mathcal{A}} \pi_i P_{iC}(t_1) P_{iA}(t_2) \sum_{j\in\mathcal{A}} P_{ij}(t3) P_{jC}(t_4) P_{jG}(t_5)$$

Sites are modeled independently of each other and the likelihood to observe the alignment $\mathcal{D}$ therefore becomes

$$\mathcal{L}(t_1,...,t_5,\mathcal{T}) = Pr(\mathcal{D}|t_1,...,t_5,\mathcal{T},\mathcal{M}) = \prod_{sites} Pr(D_{site}|t_1,...,t_5,\mathcal{T},\mathcal{M})$$

The *Maximum Likelihood tree* is the one with toplology $\hat{\mathcal{T}}$ and edge lengths $\hat{t}_1, ..., \hat{t}_5$ maximizing $\mathcal{L}$.

Maximum Likelihood trees (3)

- Sometimes the model parameters are also varied when optimizing the likelihood, that is $\mathcal{L} = \mathcal{L}(t_1, ..., t_n, \mathcal{T}, \mathcal{M})$.
- For ease of computation *log* $\mathcal{L}$ instead of $\mathcal{L}$ is maximized.
- The likelihood of a site is efficiently computed by recursion (Felsenstein 1981). A rooted tree topology is traversed from the leaves to the root. Each internal node k is assigned a conditional likelihood $\mathcal{L}_{s,k}$ as the likelihood of the subtree rooted at k, given that node k has state (nucleotide) $s \in \mathcal{A}$.

Maximum likelihood trees (4)

- ML estimation is a powerful concept to infer phylogenies. ML tree estimation has proven to be robust even when (evolutionary) model assumptions are violated.
- ML is computationally the most expensive tree reconstruction method.
- A very fast and widely used heuristic to reduce the tree search space is *Quartet Puzzling* (Strimmer, v. Haeseler 1996, see also `http://www.tree-puzzle.de/`). The optimal tree for all subsets of sequences consisting only of four sequences (=quartet) is computed. Subsequently, the quartet trees are combined into a larger tree for all sequences.