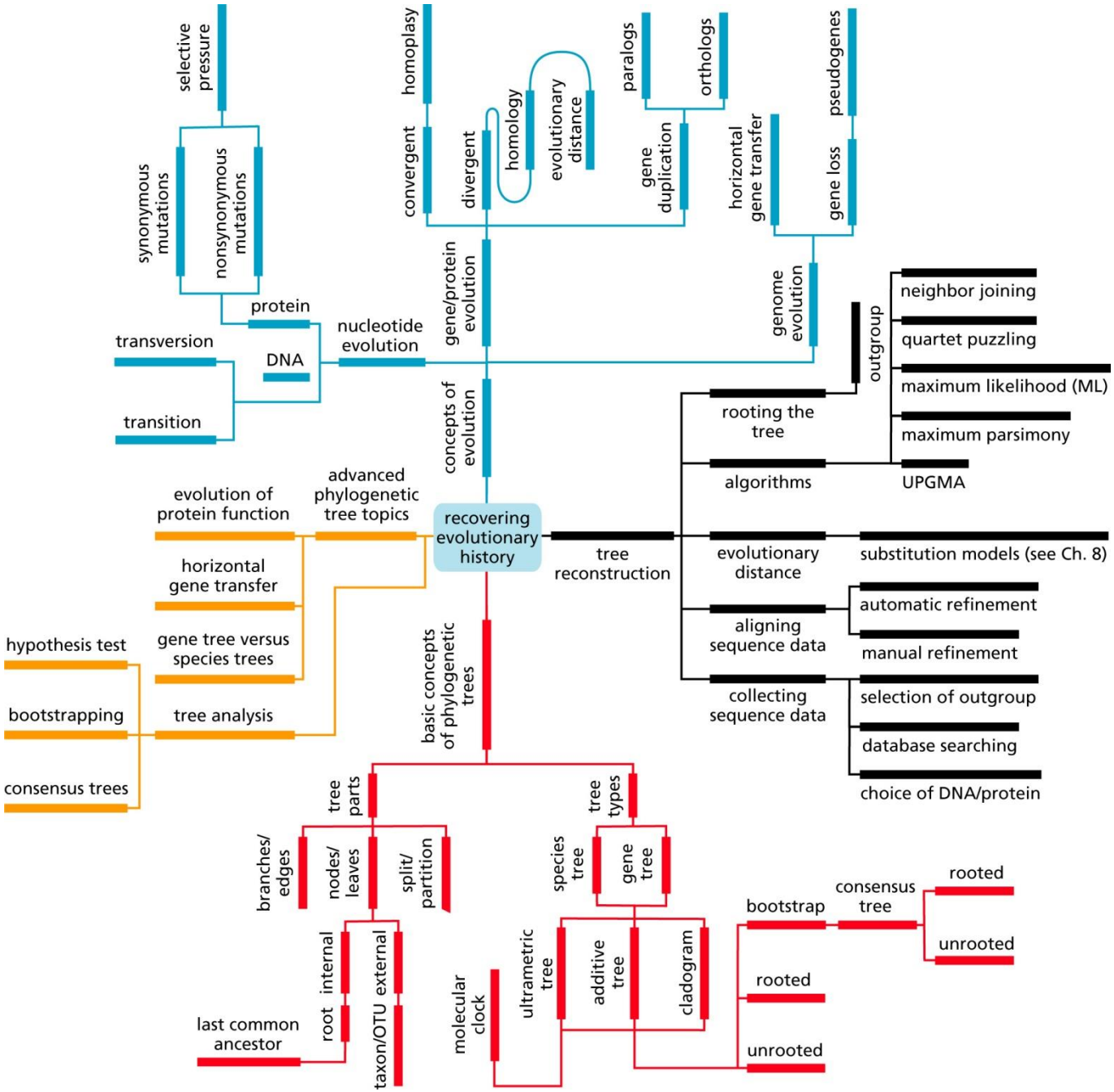
GLOBEX Bioinformatics (Summer 2015)

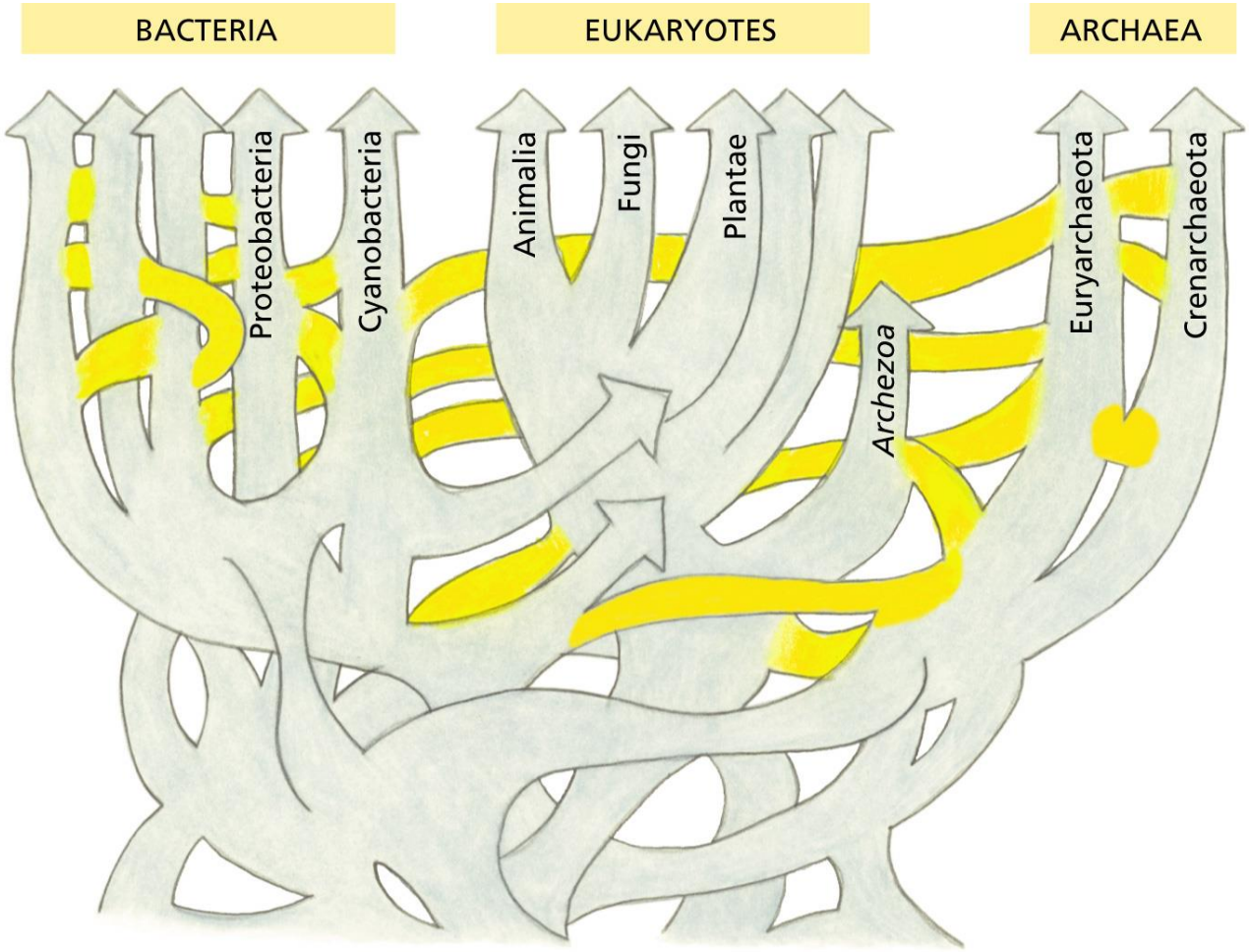
Phylogenetic Trees

- Basic concepts
- Character-based
- Distance-based
- Probability-based

Evolution

- Mutation, selection, Only the Fittest Survive.
- Speciation. At one extreme, a single gene mutation may lead to speciation. [*Nature* **425**(2003)679]
- Phylogeny: evolutionary relation among species, often represented as a tree structure.


selective pressure
synonymous mutations
nonsynonymous mutations
protein
DNA
transversion
transition
nucleotide evolution
homoplasy
convergent
divergent
homology
evolutionary distance
paralogs
orthologs
gene duplication
horizontal gene transfer
pseudogenes
gene loss
gene/protein evolution
genome evolution
concepts of evolution
neighbor joining
quartet puzzling
maximum likelihood (ML)
maximum parsimony
UPGMA
outgroup
rooting the tree
algorithms
evolution of protein function
advanced phylogenetic tree topics
recovering evolutionary history
tree reconstruction
evolutionary distance
substitution models (see Ch. 8)
horizontal gene transfer
gene tree versus species trees
hypothesis test
bootstrapping
consensus trees
tree analysis
aligning sequence data
automatic refinement
manual refinement
collecting sequence data
selection of outgroup
database searching
choice of DNA/protein
basic concepts of phylogenetic trees
tree parts
branches/ edges
nodes/ leaves
split/ partition
root
internal
external
taxon/OTU
last common ancestor
tree types
species tree
gene tree
molecular clock
ultrametric tree
additive tree
cladogram
bootstrap
consensus tree
rooted
unrooted
rooted
unrooted

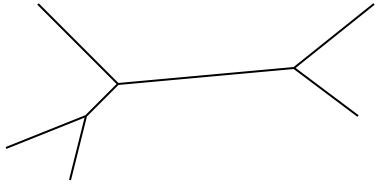

BACTERIA
EUKARYOTES
ARCHAEA
Proteobacteria
Cyanobacteria
Animalia
Fungi
Plantae
Archezoa
Euryarchaeota
Crenarchaeota

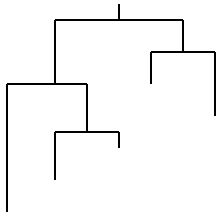
Question: how to infer phylogeny?

- Based on morphological features
- Based on molecular features
  - Gene trees
  - Phylogenetic trees (using 16s rRNA)
  - Criteria for selecting features
    - Ubiquitous
    - Relatively stable
  - Reconciliation between gene trees and species trees
    - Orthology genes

Trees (binary)

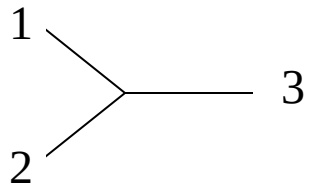
- Unrooted vs rooted

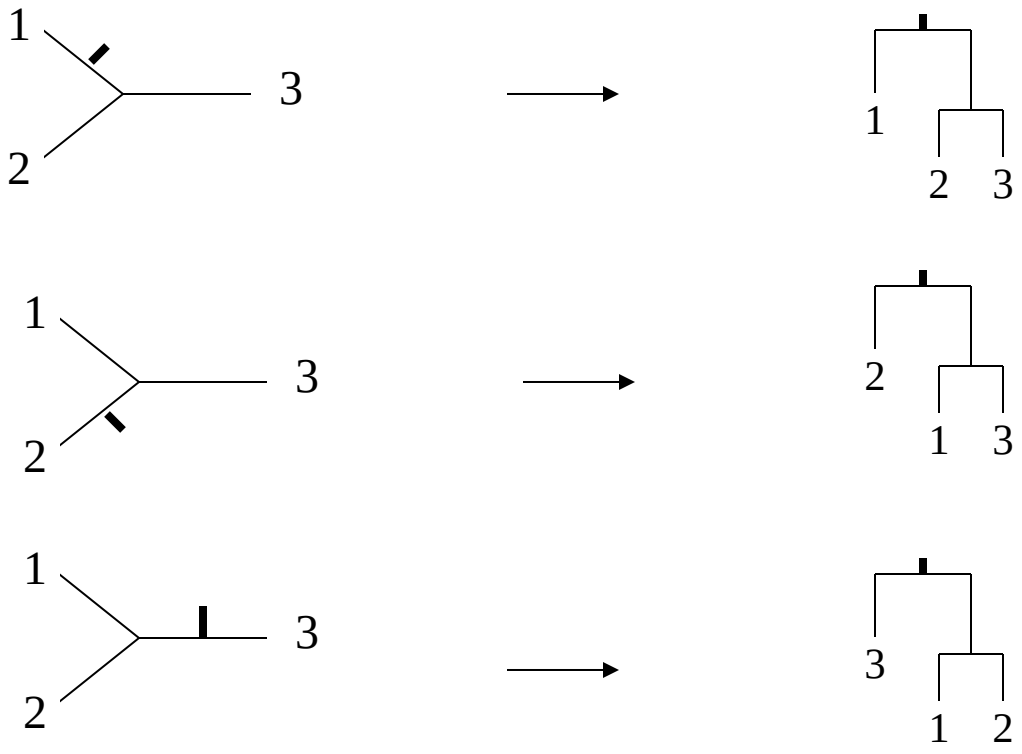




- Leaves versus internal nodes
- For an unrooted binary tree with n (≥2) leaves
  - # of nodes (including leaves) is 2n – 2.
  - # of edges is 2n -3
  - Can lead to 2n-3 rooted trees, by adding a root at any edge.

- For example,





How many different configurations can a tree of n leaves have?

Assume the tree is unrooted.

Grow the tree by adding one leaf at a time

n = 2, there is 1 edge to break.

n = 3, there are 3 edges to break => 3 different configurations

n = 4, there are 5 edges to break => 5 different configurations

…

n = n, there are (2n-3) edges to break => (2n-5)

$1 \cdot 3 \cdot 5 \cdot 7 \cdot \ldots \cdot (2n-3) = (2n-3)!!$

The number of possible configurations as a function of the tree size increases very fast (exponentially).

How to find the best tree?

- Define what “best” means?
  - Character-based: parsimony → minimal number of mutations
  - Distance-based: shorter “distance” means more closely related.
  - Probability-based: best tree gives highest likelihood for the observed (Maximum Likelihood)
- Search for the best
  - Brute-force (search space is huge, exponential in tree size)
  - Heuristics – genetic algorithm,..

Character-based approaches

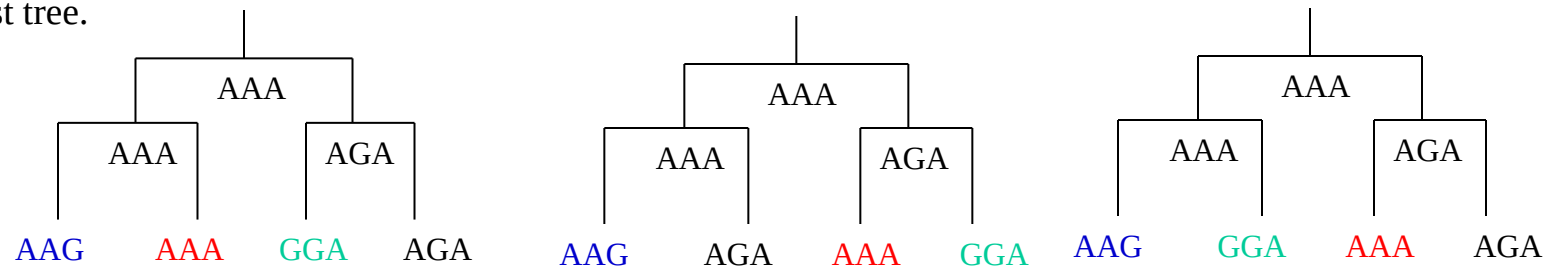
Parsimony

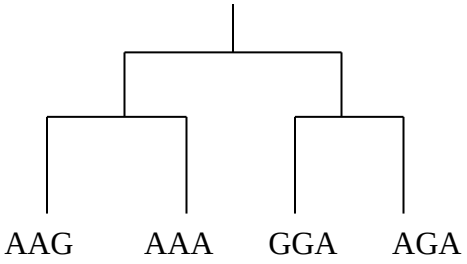
- Based on sequence alignment.
- Assign a cost to a given tree
- Search through the topological (configuration) space of all trees for the best tree: the one that has the lowest cost.

For example, given an alignment of four sequences

AAG
AAA
GGA
AGA

If the number of mutations is used as a measure of cost, then the leftmost tree in the following is the best tree.



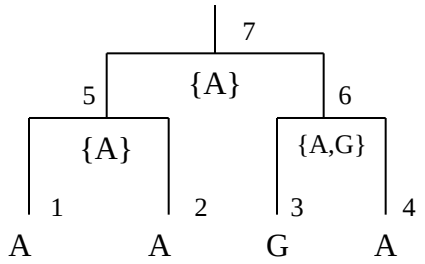


Algorithm: unweighted parsimony [Fitch 1971]
// given an alignment A of n sequences
// each position in A is treated independently
// Tree T with n leaves labeled for each sequence

C = 0; // the total cost

for (u = 1 to |A|) { // u is the position index into the alignment A
initialization:
set C_u = 0 and k = 2n -1 // C_u is the cost and k is the node index
// index starting 1, from left to right, bottom to up

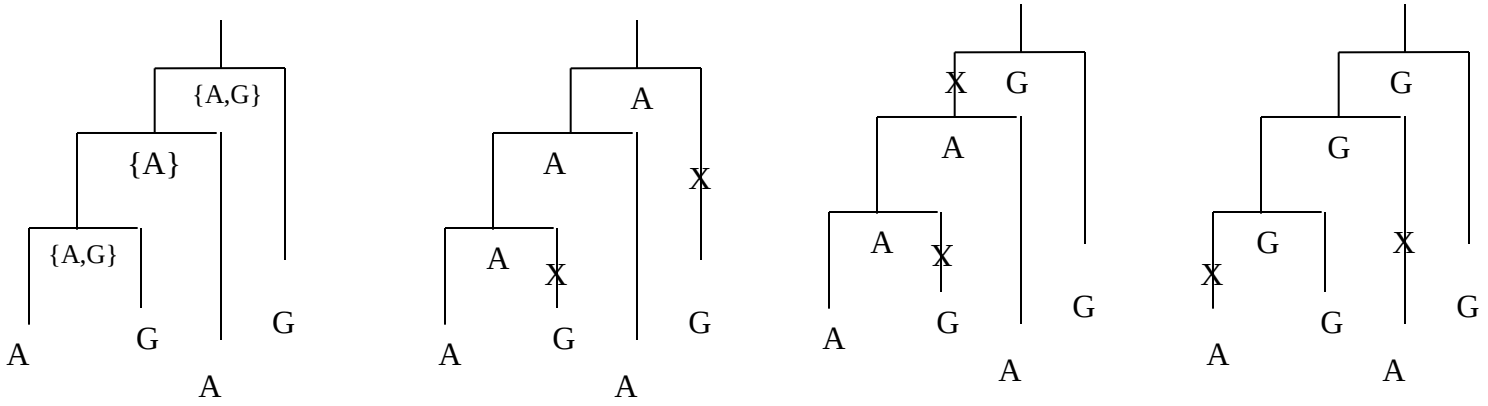
recursion: to obtain the set R_k // contains candidate residues assigning to node k
if k is leaf node:
set $R_k = x_u$ // which is the residue at position u
else
compute R_i, R_j for the daughter nodes i, j of k
if ($R_i \cap R_j$) is not empty:
set $R_k = R_i \cap R_j$
else
set $R_k = R_i \cup R_j$
$C_u = C_u + 1$
termination: $C = C + C_u$
}
minimal cost of tree = C.



Trackback phase:

- Randomly choose a residue from R_{2n-1} (the root) and proceed down the tree.
- if a residue is chosen from the set R_k
  - Choose the same residue from the daughter set R_i if possible, otherwise pick a residue at random from R_i.
  - Choose the same residue from the daughter set R_j if possible, otherwise pick a residue at random from R_j.

For example,



Traceback cannot find this tree, although it is equally optimal as the other two trees.

Algorithm: Weighted parsimony [Sankoff & Cedergren 1983]

// given an alignment A, each position in A is treated independently
// Tree T with the leaves labeled, and a residue substitute score matrix S.

C = 0; // the total cost

for (u = 1 to |A|) { // u is the position index into the alignment A

initialization:

set k = 2n -1 // k is the node index, currently pointing to the root

recursion: Compute $S_k(a)$ // the minimal cost for assigning residue a to node k

if k is leaf node:

if $a = x_u^k$ then $S_k(a) = 0$

else $S_k(a) = \infty$ // cannot substitute a leaf

else // k is not a leaf node

compute $S_i(a)$, $S_j(a)$ for all a at the daughter nodes i, j of k

set $S_k(a) = \min_b [S_i(b) + S(a,b)] + \min_b [S_j(b) + S(a,b)]$

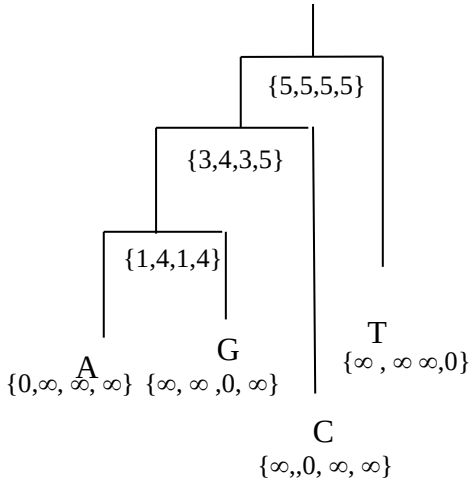
set $l_k(a) = \text{argmin}_b [S_i(b) + S(a,b)]$, $r_k(a) = \text{argmin}_b [S_j(b) + S(a,b)]$. // for traceback

termination:

$C = C + \min_a S_{2n-1}(a)$.

}

minimal cost of tree = C.



	A	C	G	T
A	0	2	1	2
C	2	0	2	1
G	1	2	0	2
T	2	1	2	0

- Both algorithms run in O(nm), where n is number of sequences and m is the sequence length in terms of number of residues.

- Weighted parsimony, when using S(a,a) = 0 for all a and S(a,b) = 1 for all a ≠ b, gives the same cost as that for the traditional parsimony.

- Traceback in weighted parsimony can find assignments that are missed in the traditional unweighted parsimony.

- The cost from the unweighted parsimony is independent of the position for the root node. Therefore, the cost can be computed using unrooted trees.

- The number of trees to search using parsimony grows huge as the number of leaves increases. It is proved that finding the most parsimonious tree is an NP-hard problem.

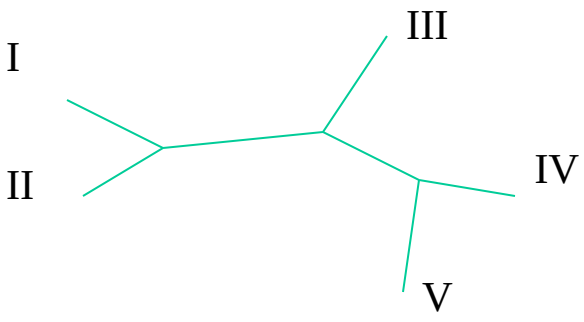
- Branch-and-bound
  - Guarantee to find the optimal tree
  - Worse-case complexity is the same as exhaustive search.

Assessing the trees: the ***bootstrap***

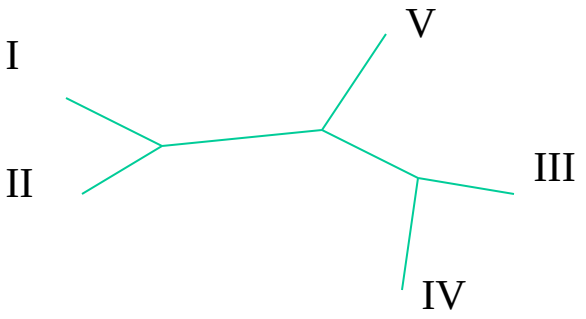
- "Plug-in" sampling with replacement
  - Given an alignment with, say, one hundred columns.
  - Randomly select one column from the original alignment as the first column, and repeat this process until one hundred columns are selected forming a new alignment of one hundred columns.
  - Use this artificially created alignment for parsimony analysis, a new tree is found.
  - Repeat this whole process many times (say 1000).
  - The frequency with which a chosen phylogenetic feature appears is used as a measure of the confidence we have in this feature.

Bootstrap

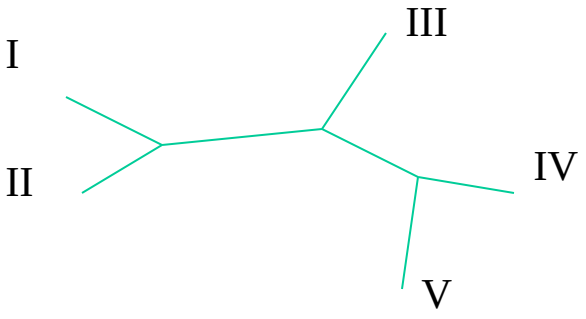
```
     1234567890
I    GGGGGGATCA
II   GGGAGTATCA
III  GGATAGACAT
Iv   GATCATGTAT
V    GTTCATATCT
```

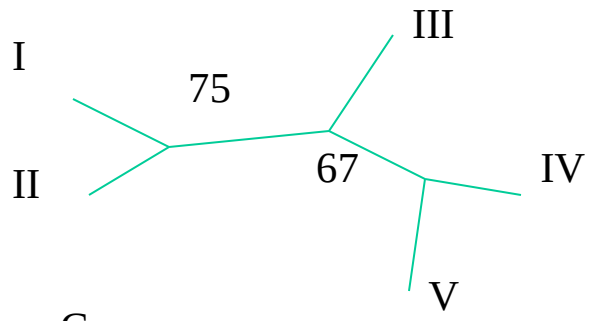



```
     1155569999
I    GGGGGGCCCC
II   GGGGGTCCCC
III  GGAAAGAAAA
Iv   GGAAATAAAA
V    GGAAATCCCC
```



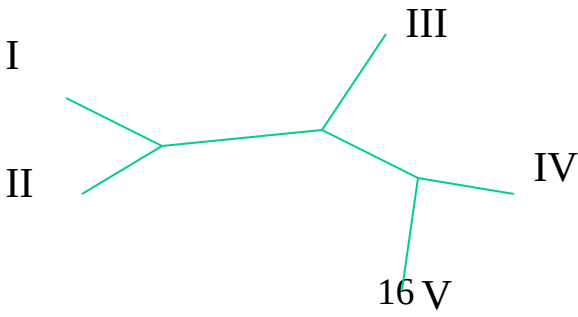

```
     1224555770
I    GGGGGGGAAA
II   GGGAGGGAAA
III  GGATAAAAAT
Iv   GATCAAAGGT
V    GTTCAAAAAT
```





Consensus bootstrap tree

```
     3334667888
I    GGGGGGATTT
II   GGGATTATTT
III  AAATGGACCC
Iv   TTTCTTGTTT
V    TTTCTTATTT
```



Distance-based approaches

UPGMA – unweighted pair group method using arithmetic averages

Distance between two clusters C_i and C_j:

$$d_{ij} = (1/|C_i||C_j|) \sum_{p \in Ci,\, q \in Cj} d_{pq}.$$

Note: it is NOT always possible to interpret pairwise sequence similarity scores as metric distance.

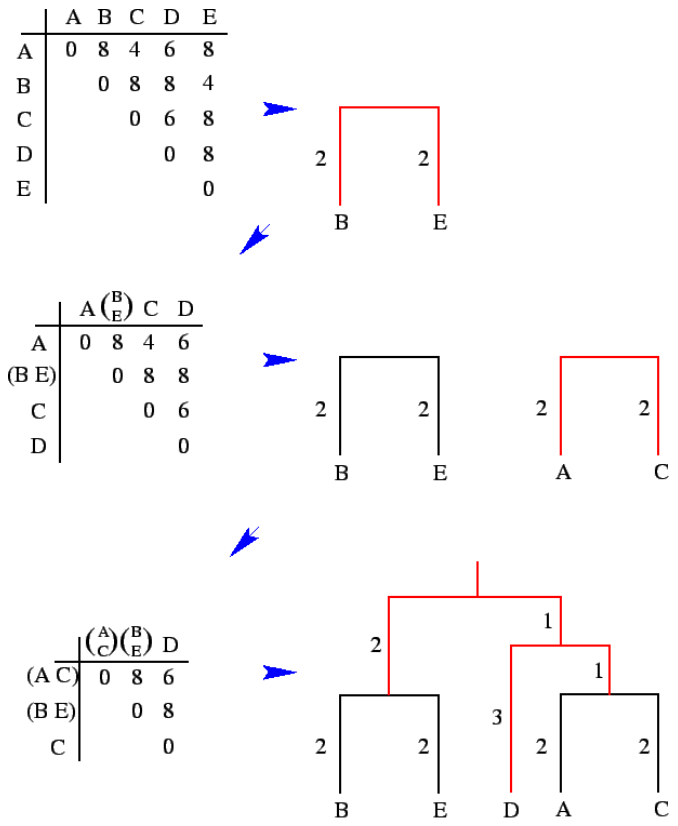


Figure: Construction of an ultrametric tree

Algorithm: UPGMA

Initialization:

- Assign each sequence i to its own cluster C_i
- Define one leaf of T for each sequence, and place at height zero

Iteration:

- Determine the two clusters i, j for which d_{ij} is minimal.
- Define a new cluster k by $C_k = C_i \cup C_j$, and define d_{km} for all m
- Define a node k with daughter noes i and j, and place it at height $d_{ij} / 2$.
- Add k to the current clusters and remove i and j.

Termination:

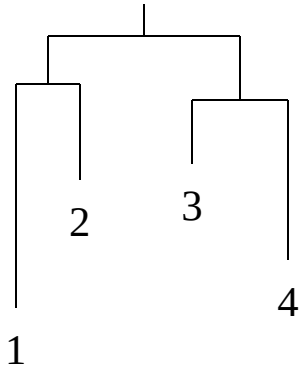
- When only two clusters i, j remain, place the root at height $d_{ij}/ 2$.

Ultrametric: for any triplet (x_i, x_j, x_k), distances d_{ij}, d_{jk}, d_{ki} are either all equal or two are equal and the remaining is smaller.

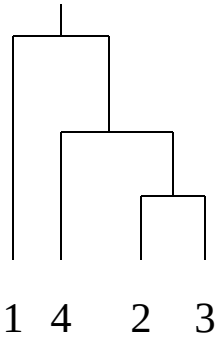
Molecular clock: two siblings evolve at the same constant rate.

Such requirements are often not satisfied, and UPGMA trees then will be not correct.

For example,



Actual tree



Tree reconstructed incorrectly using UPGMA

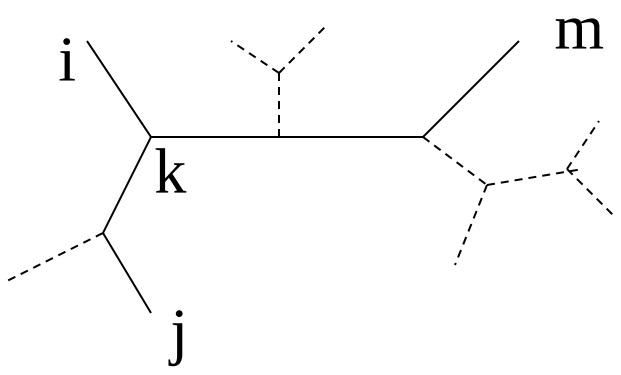
Neighbor-joining:

- Distances are additive.
- Given a pair of leaves, determine if they are neighboring leaves (not necessarily with shortest distance)
- Once we merge a pair of neighboring leaves, how do we compute the distance between this pair (as a whole, called k) and another leaf, called m?

$$\mathbf{\tfrac{1}{2}\,(d_{im} + d_{jm} - d_{ij})}$$

$$= \tfrac{1}{2}\,(d_{ik} + d_{km} + d_{jk} + d_{km} - d_{ik} - d_{jk}\,)$$

$$= \tfrac{1}{2}\,(d_{km} + d_{km}\,) = d_{km}.$$



Without a tree, how can we know that if two leaves are neighbor (when neighbors do not mean shortest distance)?

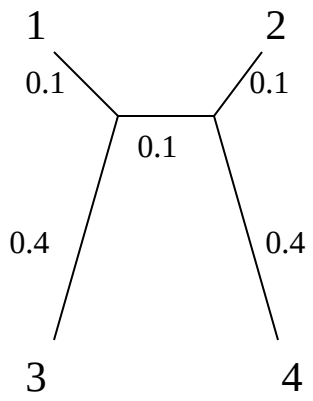
Theorem (Saitou & Nei, 1987**):** For each leaf i, define r_i as

$$r_i = (1/(|L|-2)) \sum_{k \in L} d_{ik},$$

where L stands for the set of leaves.

Then a pair of leaves i and j will be neighboring leaves if $D_{ij} = d_{ij} – (r_i + r_j)$ is minimal.

Example:



$d_{12} = 0.3$ $D_{12} = -1.1$

$d_{13} = 0.5$ $D_{13} = -1.2$

$d_{14} = 0.6$ $D_{14} = -1.1$

$d_{23} = 0.6$ $D_{23} = -1.1$

$d_{24} = 0.5$ $D_{24} = -1.2$

$d_{34} = 0.9$ $D_{34} = -1.1$

$r_1 = 0.7$

$r_2 = 0.7$

$r_3 = 1.0$

$r_4 = 1.0$

Neighbor joining will generate unrooted trees.

Initialization:

define T to be the set of leaf nodes, one for each given sequence, and put L = T

Iteration:

- Pick a pair i, j in L for which D_{ij} is minimal

- Define a new node k and set $d_{km} = ½(d_{im} + d_{jm} – d_{ij})$ for all m in L.

- Add k to T with edges of lengths $d_{ik} = ½ (d_{ij} + r_i – r_j)$, $d_{jk} = d_{ij} – d_{ik}$.

- Remove i and j from L and add k.

Termination:

When L consists of two leaves I and j, add the remaining edge between i and j, with length d_{ij}.

Pros and Cons of distance-based methods

- Easy to implement, and fast to run
- Robust to minor sequence errors
- Distance-based phylogenetic trees do not generate ancestral sequences
- Definition of “distance” may be problematic

Probabilistic Approaches

Evolutionary processes are by nature stochastic.

Baye's rule: P(model|data) = P(data|model) P(model)/P(data)

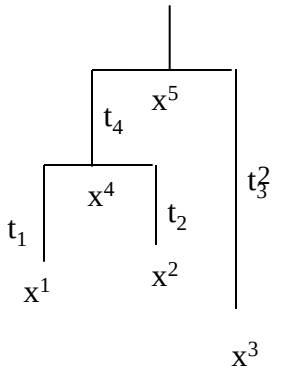
- model includes
  - the evolution theory,
  - a specific phylogenetic tree (topology and edge lengths), and
  - assignment of sequences to the tree leaves.
- Data: a set of sequences that are used to infer phylogeny.

Let P(x|y, t) be the probability in that sequence x is evolved from an ancestral sequence y over an edge of length t.

$P(x^1, x^2, x^3, x^4, x^5|T, t.)$

$= P(x^1| x^4, t_1) P(x^2| x^4, t_2) P(x^3| x^5, t_3) P(x^4| x^5, t_4) P(x^5)$

A shorthand notation $P(x^{\cdot}|T, t.)$ is used where $x^{\cdot}$ stands for a set of sequences, and t. for edge lengths of the tree T.



In general, if we know

- P(x|y, t) for any sequences x, y, and time duration t
- A tree T, and assignment of sequences to tree nodes,

Then we can compute the likelihood for observing the sequences as they are assigned onto the tree leaves.

Q: Given n, the number of leaves, there are (2n-3)!! different trees (plus many different ways to assign length to tree edges), which tree can best interpret the data?

A: The tree that gives the maximum likelihood (ML).

In practice, to implement the ML method, two issues we need to address

1. A model of evolution, which gives the conditional probabilities P(x|y, t)
2. Method to find the maximum likelihood. For this, any optimization method may be utilized, such as
   - Descent gradient
   - Simulated annealing
   - Genetic algorithm

Models of evolution

Independence assumption: mutations occur independently at different positions.

Therefore,

$$P(x|y, t) = \prod_u P(x_u|y_u, t),$$

where $P(x_u|y_u, t)$ is the probability that residue x_u in sequence x mutates to residue y_u in sequence y over time t.

multiplicative assumption:

$$P(b|a, t + \Delta t) = \sum_c P(c|a, t) \cdot P(b|c, \Delta t).$$

For DNA sequences, probabilities for all possible mutations among four nucleotides during a given time period t form a 4 by 4 matrix

$$S(t) = \begin{pmatrix} P(A|A, t) & P(A|C, t) & P(A|G, t) & P(A|T, t) \\ P(C|A, t) & P(C|C, t) & P(C|G, t) & P(C|T, t) \\ P(G|A, t) & P(G|C, t) & P(G|G, t) & P(G|T, t) \\ P(T|A, t) & P(T|C, t) & P(T|G, t) & P(T|T, t) \end{pmatrix}$$

And, we have multiplicative property for these matrices

$$S(t) \cdot S(\Delta t) = S(t + \Delta t).$$

For each P(b|a, t) in the substitution matrix S(t), it is reasonable to assume that no mutation can occur at zero time interval:

- P(a|a, 0) = 1
- P(b|a, 0) = 0.

That is,

$$S(0) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Further, let's assume that P(b|a, Δt) over a infinitesimal interval Δt is proportional to Δt by a constant r, called mutation rate:

$$P(b|a, \Delta t) = r\, \Delta t.$$

Jukes-Cantor model for DNA sequences assumes that all nucleotide mutations have the same rate. That is,

$$S(\Delta t) = \begin{pmatrix} 1\text{-}3r & r & r & r \\ r & 1\text{-}3r & r & r \\ r & r & 1\text{-}3r & r \\ r & r & r & 1\text{-}3r \end{pmatrix} \qquad \Delta t \equiv I + R\, \Delta t$$

Therefore,

$$S(t + \Delta t) = S(t) \cdot S(\Delta t) = S(t) \cdot [I + R\Delta t]$$

$$[S(t + \Delta t) - S(t)] / \Delta t = S(t) \cdot R$$

$$S'(t) = S(t) \cdot R \quad \text{when } \Delta t \rightarrow 0.$$

Solving the differential equations, we have

- $P(a|a, t) = \frac{1}{4} (1+ 3e^{-4rt})$ for $a \in \{A,C, G, T\}$
- $P(b|a, t) = \frac{1}{4} (1- e^{-4rt})$ for $a \neq b$, a and $b \in \{A,C, G, T\}$

In this model, when $t = \infty$, $P(b|a, \infty) = \frac{1}{4}$. That is, the nucleotide equilibrium frequencies are all equal.

Kimura model for DNA sequences assumes different rates for transitions (A↔G and C ↔ T) and transversions (A ↔ T, G ↔ T, A ↔ C, and C ↔ G).

That is,

$$S(\Delta t)= \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{pmatrix} 1-2r-s & r & s & r \\ r & 1-2r-s & r & s \\ s & r & 1-2r-s & r \\ r & s & r & 1-2r-s \end{pmatrix} \end{array} \Delta t \equiv I+R\,\Delta t$$

Similar models are proposed for mutations among amino acids.

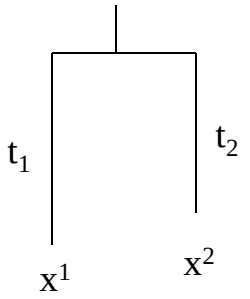
If we were able to quantify the "time" as how many number mutations have occurred, the substitute matrices in those models would correspond to PAM matrices at respective times.

Maximum Likelihood

- The case of two sequences aligned with no gaps

$P(x^1, x^2|T, t_1, t_2) = \prod_{u=1 \text{ to } N} P(x^1_u, x^2_u|T, t_1, t_2)$

- Let x^1 = CCGGCCGCGCG
  x^2 = CGGGCCGCCCG



$$P(C,C| T, t_1, t_2) = P(C|A, t_2) P(C|A, t_1) P(A) + P(C|C, t_2) P(C|C, t_1) P(C) + P(C|G, t_2) P(C|G, t_1) P(G) + P(C|T, t_2) P(C|T, t_1) P(T)$$

$$= \tfrac{1}{4} [3 \times \tfrac{1}{4} (1- e^{-4rt1}) \times \tfrac{1}{4} (1- e^{-4rt2}) + \tfrac{1}{4} (1+3 e^{-4rt1}) \times \tfrac{1}{4} (1+3 e^{-4rt2})]$$

$$= 1/16 (1+3 e^{-4r(t1 +t2)}).$$

$$P(G,G| T, t_1, t_2) = 1/16 (1+ 3 e^{-4r(t1 +t2)}).$$

$$P(C,G| T, t_1, t_2) = 1/16 (1- e^{-4r(t1 +t2)}).$$

Therefore, for an alignment that has n1 identical sites and n2 mutational sites, we have

$$P(x^1, x^2|T, t_1, t_2) = 1/ 16^{n1+n2} \times (1+3 e^{-4r(t1 +t2)})^{n1} \times (1- e^{-4r(t1 +t2)})^{n2},$$

which is a function of edge lengths in tree T.

In general, the probability can be computed by working up the tree from the leaves using post-order traversal. This is done by Felsenstein's algorithm (1981).

Once the probability is available, optimizing the assignments of edge lengths t in the tree amounts to

$$\left.\frac{\partial P}{\partial t}\right|_{t_m} = 0$$

Where t_m is the tree length assignment that maximize the likelihood.

How to optimize tree topology?

- Discrete structure, therefore cannot take derivatives.

Basic strategy for searching the tree space

- A tree generation algorithm that can generate trees
- Assess the likelihood
  - Accept
  - Reject

A genetic algorithm implementation [Matsuda 1998]

Genetic algorithm

Input
- P, the population,
- r: the fraction of population to be replaced,
- f, a fitness,
- ft, the fitness_threshold,
- m: the rate for mutation.

Initialize population (randomly)

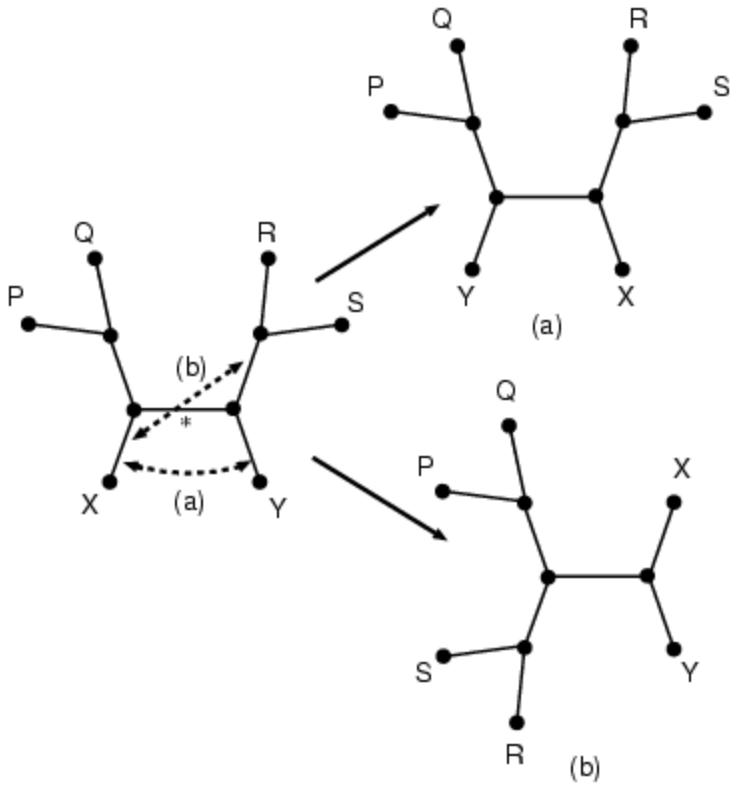
Evaluate: for each h in P, compute Fitness(h)

While [Max_h f(h)] < ft

do

1. Select
2. Crossover
3. Mutate
4. Update P with the new generation Ps
5. Evaluate: f(h) for all h $\in$ P

Return the h in P that has the best fitness

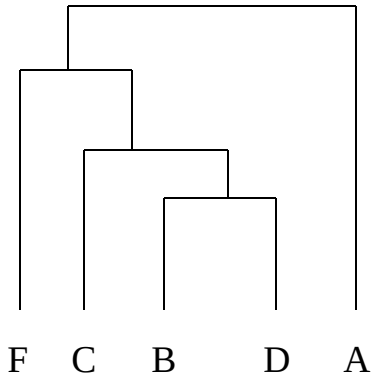


Branch exchange in a phylogenetic tree

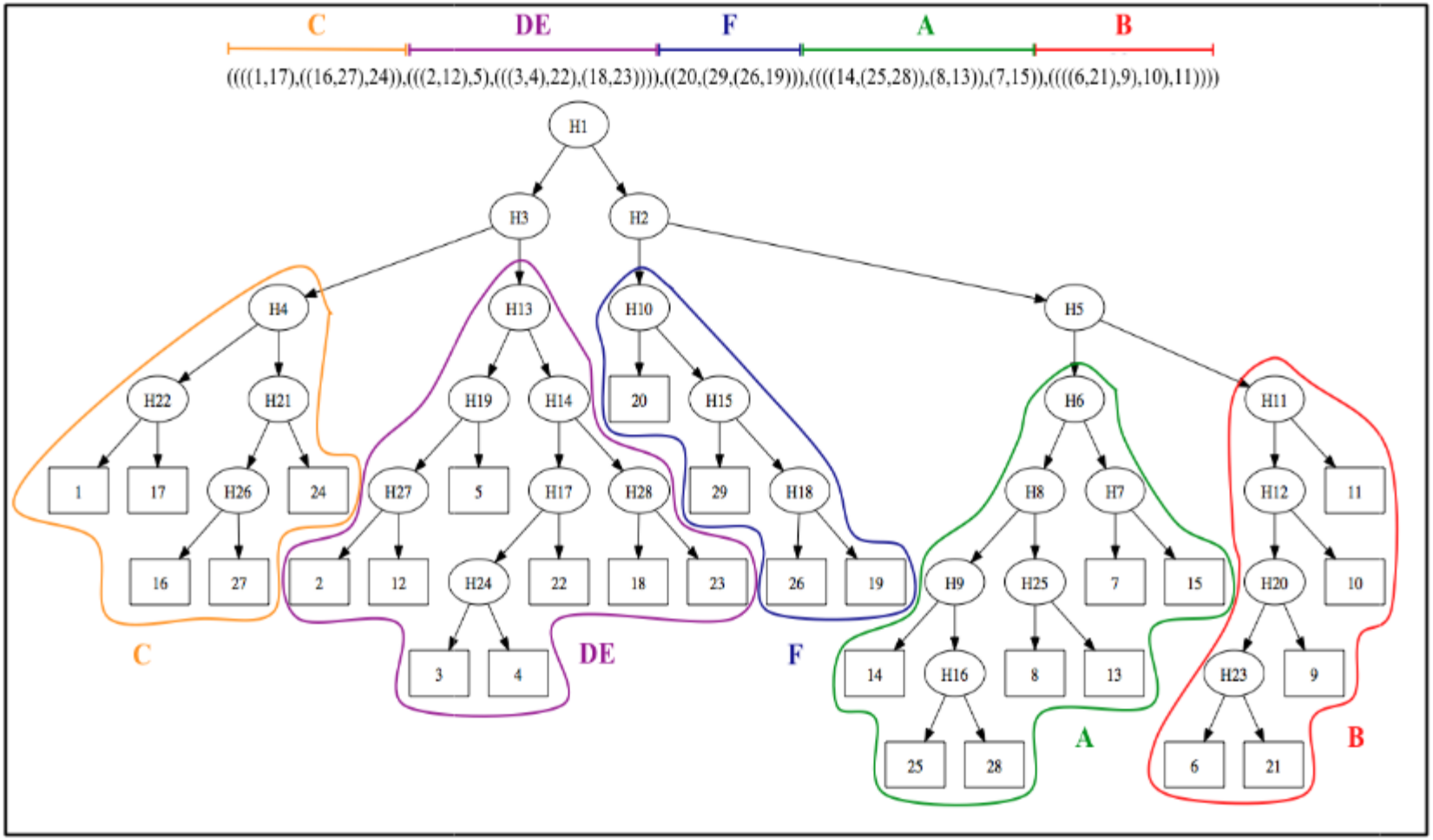
Key components for implementing genetic algorithms

- Representing hypotheses (which are the trees here)
  - New Hampshire format

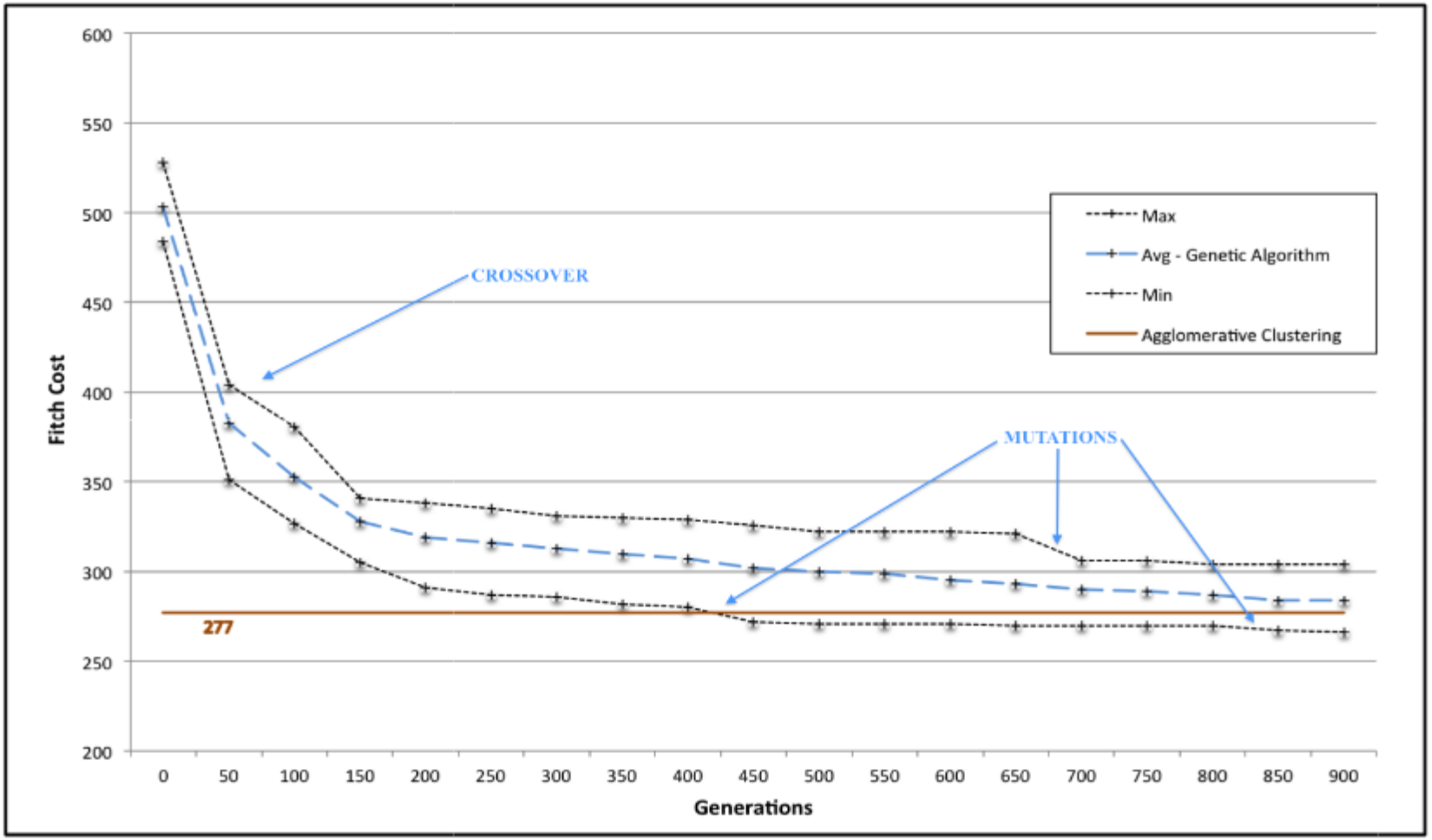
    (A,(F,(C,(B,D))))
- Genetic operators
  - Crossover:
    - Single
    - Two point
    - uniform
  - Mutation: point



- Fitness function: we use the likelihood computed for each tree using



Blanchette, O;Keefe & Benuskova, 2012



Blanchette, O;Keefe & Benuskova, 2012

Software packages and databases for phylogenetic trees

- Phylip by Felsenstein (http://evolution.genetics.washington.edu/phylip.html)
- PAUP (http://paup.csit.fsu.edu/)
- MacClad (http://macclade.org/macclade.html)
- TreeBase (http://www.treebase.org/treebase/)

More advanced topics in phylogenetic analysis

- Different heuristics for sampling the tree space
  - Monte Carlo
  - …
- More realistic evolutionary models
  - With gaps
  - Non-uniform: different rates at different sites
  - …
- Using different data sets and reconciliation
  - Sequences
  - Gene positions -> genome rearrangement [Nadeau & Taylor 1984, PNAS 81:814-818, Pavzner, Sankoff, …]
  - …