Inference of phylogenetic trees

Jiří Kléma

Department of Computer Science,
Czech Technical University in Prague

Lecture based on Mark Craven's class at University of Wisconsin

http://cw.felk.cvut.cz/wiki/courses/b4m36bin/start

Overview

- Phylogenetic inference: task definition,
- basics from graph theory,
- motivation for phylogenetic analysis
  - example trees,
- three general types of methods
  - distance: find tree that accounts for estimated evolutionary distances,
  - parsimony: find the tree that requires minimum number of changes to explain the data,
  - maximum likelihood: find the tree that maximizes the likelihood of the data.

Phylogenetic inference: task definition

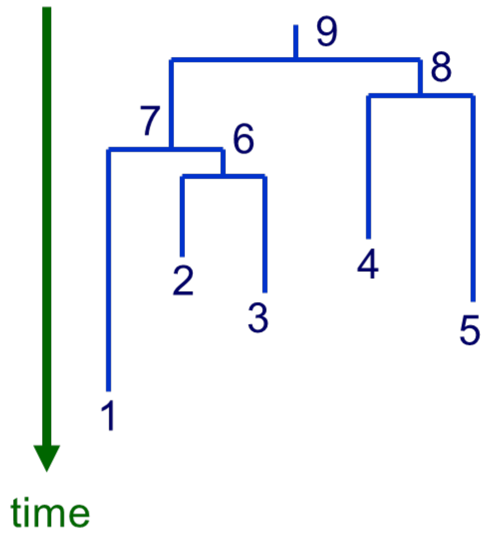
- Given
  - data characterizing a set of species/genes,
  - earlier: morphological data,
  - today: nucleotide sequences or amino acid sequences,
- Do
  - infer a phylogenetic tree that accurately characterizes the evolutionary lineages among the species/genes,
  - phylogenesis = the evolutionary development and diversification of a species or group of organisms,
  - limitations: homoplasy, horizontal gene transfer, etc.

Phylogenetic tree basics

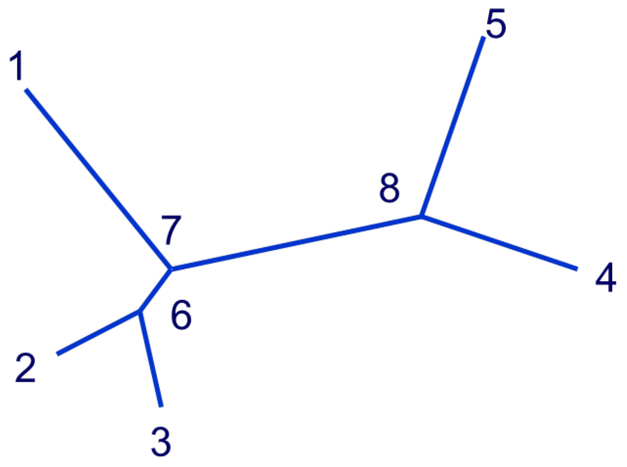
- Tree
  - an undirected graph without cycles,
  - a directed graph whose underlying undirected graph is a tree (often also $\forall v$: indegree(v)$\leq$1 to avoid polytrees with many roots),
- phylogenetic tree
  - leaves = things (genes, species, individuals/strains) being compared,
  - internal nodes = hypothetical ancestral units,
  - **taxon** (taxa plural) = species and broader classifications of organisms,
- rooted and unrooted trees
  - a directed tree has a root
    - the root represents the common ancestor,
    - path from root to a node represents an evolutionary path,
  - an undirected tree is unrooted
    - specifies relationships among things, but not evolutionary paths.

Rooted and unrooted trees

- The role of root
  - an extra node that tells us the direction of evolution,
- the number of possible trees for n leaves (sequences) quickly grows
  - unrooted: $\prod_{i=3}^{n}(2i-5)$,
  - rooted: $(2n-3)\prod_{i=3}^{n}(2i-5)$.



rooted tree



unrooted tree

Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

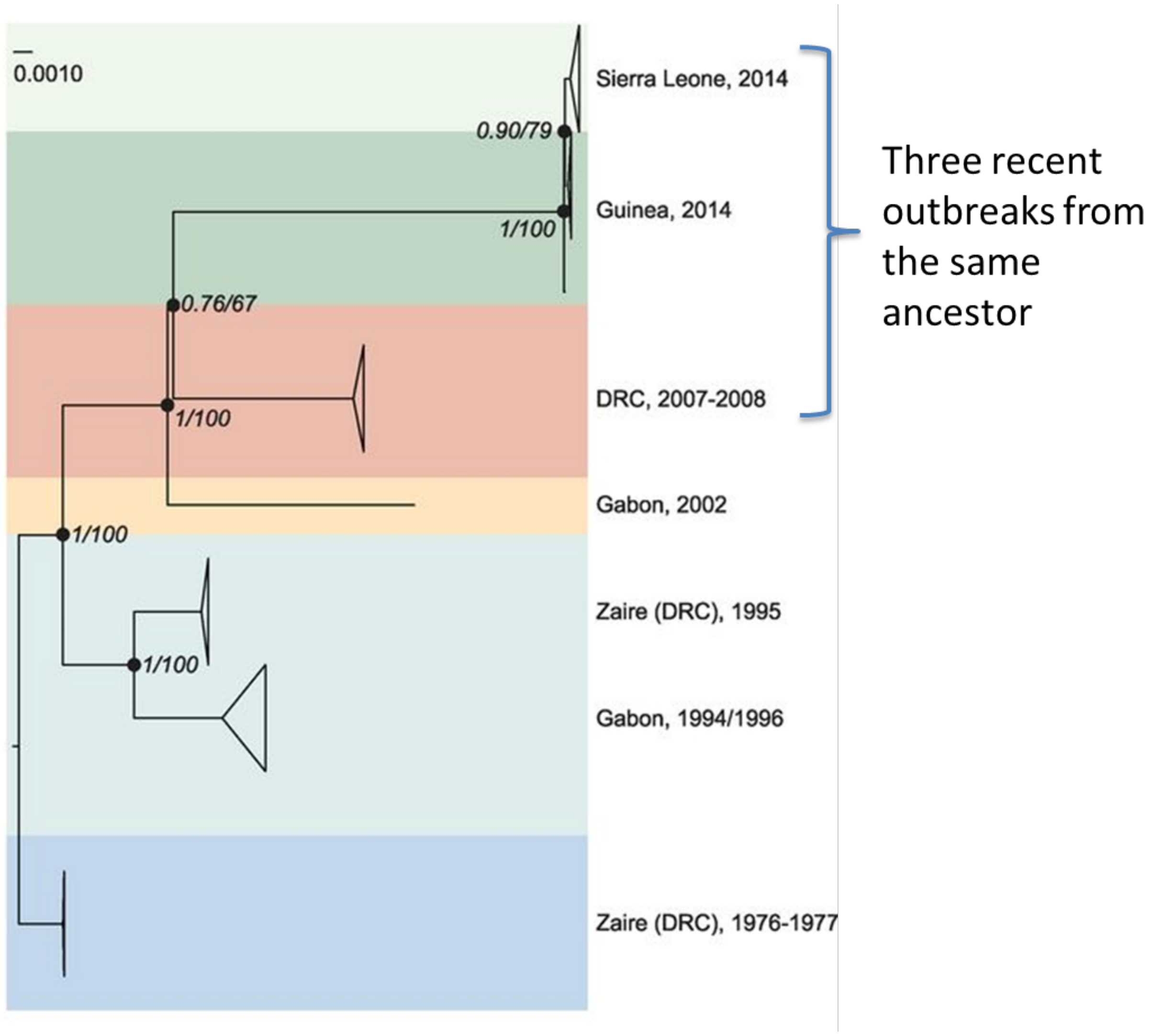
Why construct phylogenetic trees?

- to gain knowledge of biologial diversity from raw data
  - and organize it in a structured (hierarchical) way,
- to understand evolutionary lineage of various species
  - straightforward reconstruction, see tree paths from the root to a leaf,
- to understand how various functions evolved and which loci underlie it
  - may help extract functional (e.g., gene-trait association) signal from genomic data,
- to inform multiple alignments
  - multiple sequence alignments often used to create a phylogenetic tree,
  - the knowledge of phylogeny helps to improve multiple sequence alignments (guide trees),
- to identify what is most conserved/important in some class of sequences
  - those that keep relatively unchanged far back up the phylogenetic tree.

Example tree: tracing the evolution of the Ebola virus

- Ebola virus: a lethal human pathogen
- 2014 Ebola epidemic in Africa
  - until recently the largest case in 1976 (318 cases),
  - outbreak reported in Feb 2014,
  - 11,315 deaths, fatality rate 78%,
- key questions
  - where did the pathogen come from?
  - how is it evolving?
- In a 2014 Science paper
  - whole genome sequence alignment of 99 Ebola virus genomes from 78 patients in Sierra Leone,
  - also three published Guinean samples and 20 genomes from earlier outbreaks.

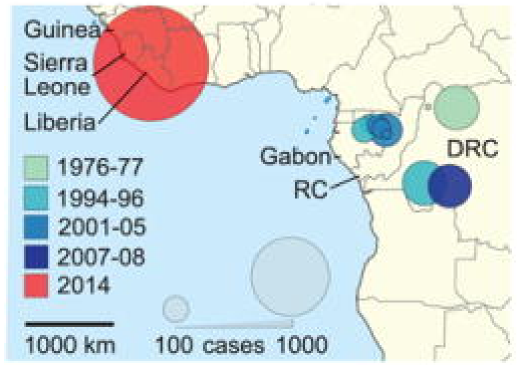
Example tree: tracing the evolution of the Ebola virus



Gire et al., Science 2014.

Example tree: tracing the evolution of the Ebola virus

- Insights gained from sequence comparison [Gire et al., Science 2014]
  - "Genetic similarity across the sequenced 2014 samples suggests **a single transmission from the natural reservoir**, followed by **human-to-human transmission during the outbreak.**",
  - ". . . the Sierra Leone outbreak stemmed from the introduction of two genetically distinct viruses from Guinea around the same time . . . ",
  - ". . . the three most recent outbreaks (2002, 2007, 2014) represent an independent zoonotic event from the same genetically diverse viral population in its natural reservoir . . . ".

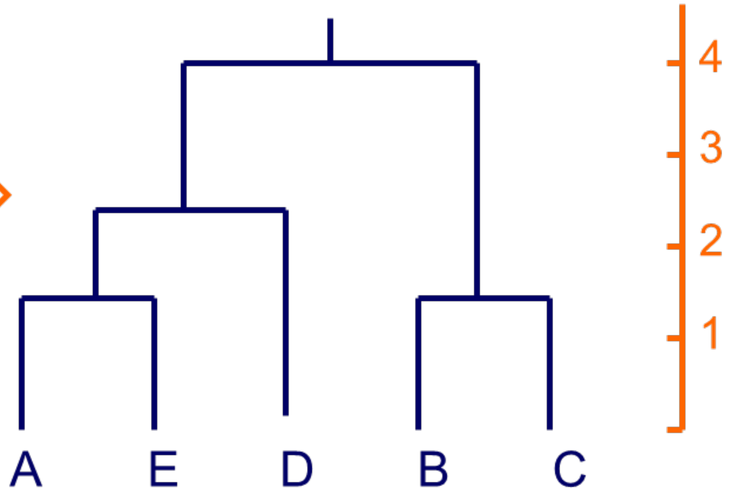


Gire et al., Science 2014.

Distance-based approaches

- given: an $n \times n$ matrix M, where M_{ij} is the distance between taxa i and j,
- do: build an edge-weighted tree such that the distances between leaves i and j correspond to M_{ij}.

	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

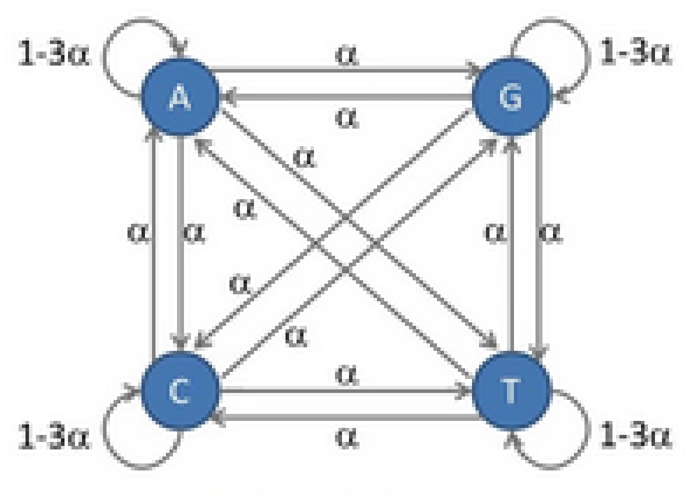
Where do we get distances?

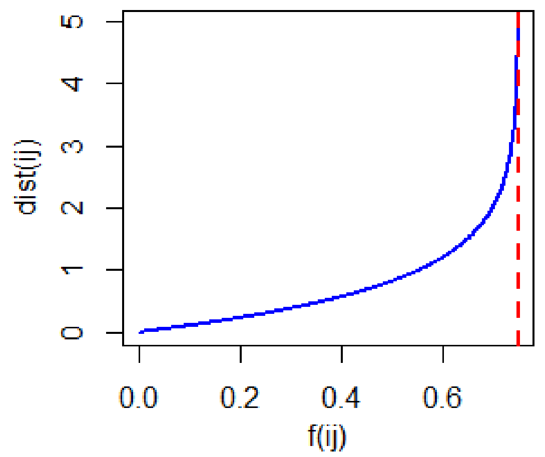
- Commonly obtained from sequence alignments
  - in alignment of sequence i with sequence j, $\text{dist}_{ij} = f_{ij}$

$$f_{ij} = \frac{\#mismatches}{\#matches + \#mismatches}$$

- to correct for multiple substitutions at a single position
  - use Jukes-Cantor model for mutation rates

$$dist_{JC}(i.j) = -\frac{3}{4}\ln\Big(1 - \frac{4}{3}f_{ij}\Big)$$

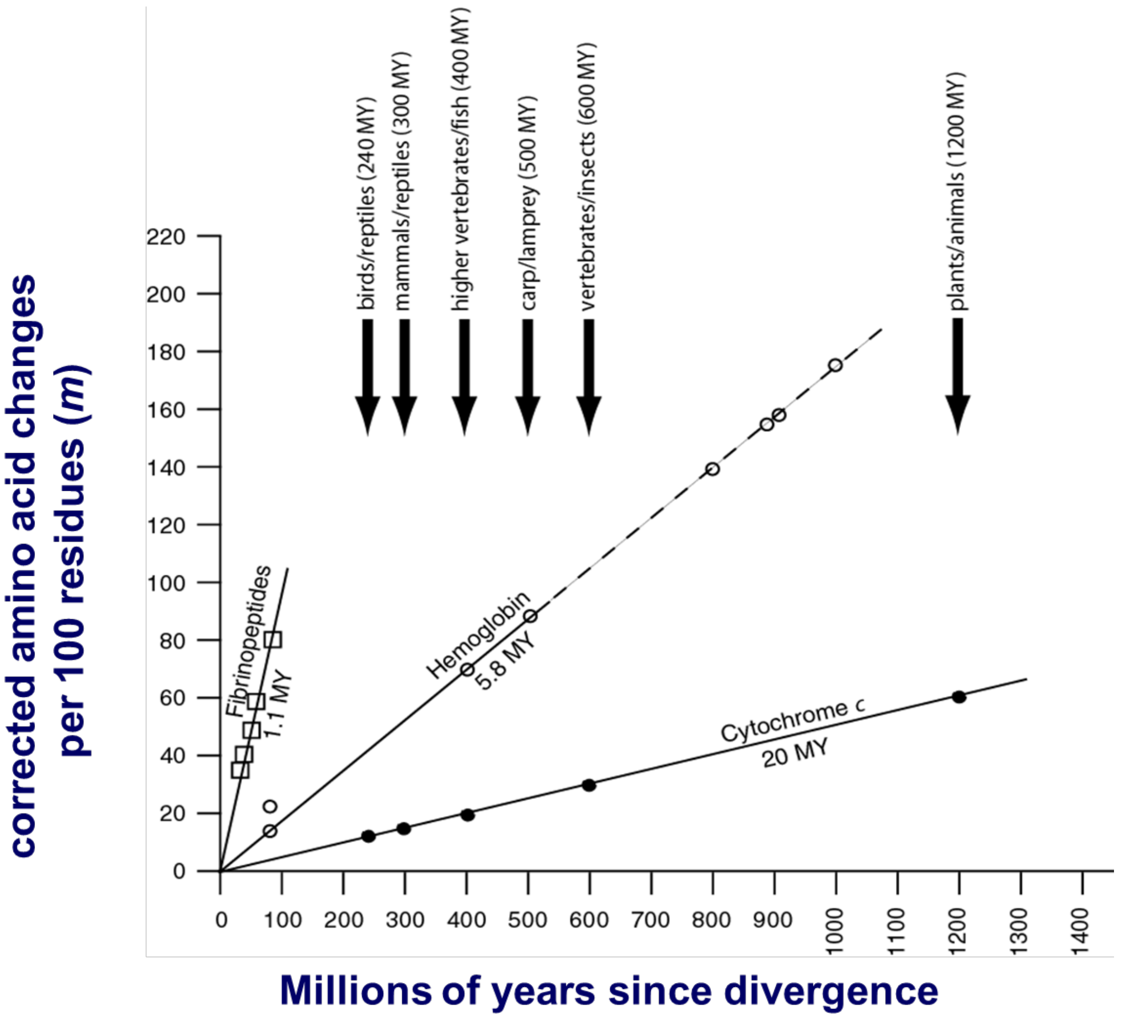




The molecular clock hypothesis and ultrametric data

- In the 1960s, sequence data were accumulated for small, abundant proteins such as globins, cytochromes c, and fibrinopeptides. Some proteins appeared to evolve slowly, while others evolved rapidly.
- Linus Pauling, Emanuel Margoliash and others proposed the hypothesis of a **molecular clock**: For every given protein, the rate of molecular evolution is approximately constant in all evolutionary lineages.
- the molecular clock assumption is not generally true: selection pressures vary across time periods, organisms, genes within an organism, regions within a gene,
- if it does hold, then the data is said to be **ultrametric**
  - this property simplifies construction of rooted phylogenetic trees.

The molecular clock hypothesis



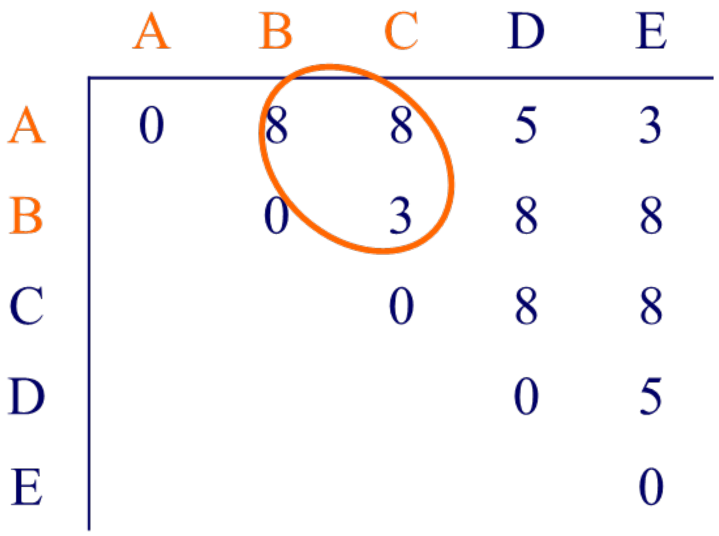
Pevsner: Bioinformatics and Functional Genomics, Wiley, 2009.

Distance metrics

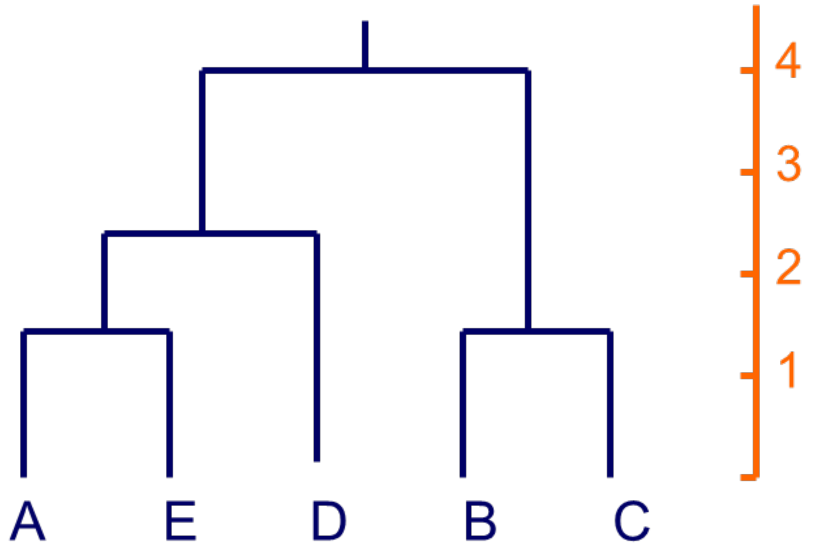
- Properties of distance metrics
  - identity: $dist(x_i, x_i) = 0$,
  - symmetry: $dist(x_i, x_j) = dist(x_j, x_i)$,
  - triangle inequality: $dist(x_i, x_j) \leq dist(x_i, x_k) + dist(x_k, x_j)$,
  - non-negativity: $dist(x_i, x_j) \geq 0$ (follows from the previous properties),
- semimetric if the triangle equality does not hold,
- ultrametric property makes the triangle equality condition stronger
  - ultrametric: $dist(x_i, x_j) \leq max(dist(x_i, x_k), dist(x_k, x_j))$.

The molecular clock hypothesis and ultrametric data

- Ultrametric data
  - for any triplet of sequences, i, j, k, the distances are either all equal, or two are equal and the remaining one is smaller.



	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

The UPGMA method

- Unweighted Pair Group Method using Arithmetic Averages (UPGMA),
- given ultrametric data, UPGMA will reconstruct the tree T that is consistent with the data,
- basic idea
  - iteratively pick two taxa/clusters and merge them.
  - create new node in tree for merged cluster.
- distance d_{ij} between clusters C_i and C_j of taxa is defined as
  - average distance between pairs of taxa from each cluster
  - $d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} d_{pq}$
- given a new cluster C_k formed by merging C_i and C_j,
- we can calculate the distance between C_k and any other cluster C_l as follows
  - $d_{kl} = \frac{d_{il}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$

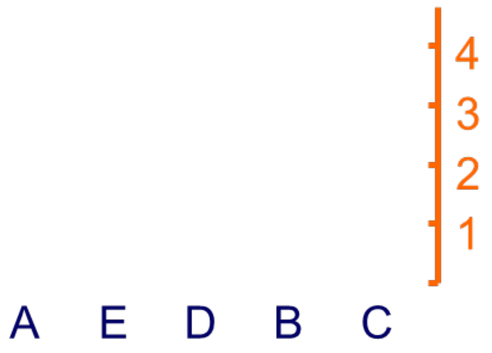
The UPGMA algorithm

- assign each taxon to its own cluster,
- define one leaf for each taxon, place it at height 0,
- while more than two clusters
  - determine two clusters i, j with smallest d_{ij},
  - define a new cluster $C_k = C_i \cup C_j$,
  - define a node k with children i and j, place it at height $d_{ij}/2$,
  - replace clusters i and j with k,
  - compute distance between k and other clusters,
- join last two clusters, i and j, by root at height $d_{ij}/2$.

UPGMA example

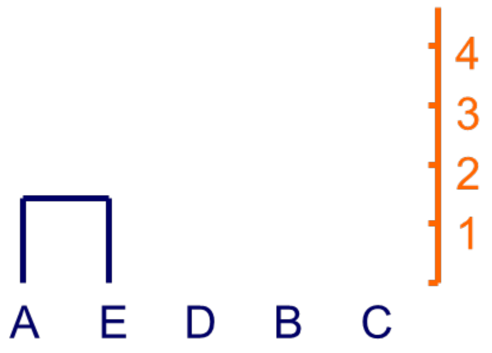
initial state

	A	B	C	D	E
A	0	8	8	5	3
B		0	3	8	8
C			0	8	8
D				0	5
E					0



after one merge

	AE	B	C	D
AE	0	8	8	5
B		0	3	8
C			0	8
D				0

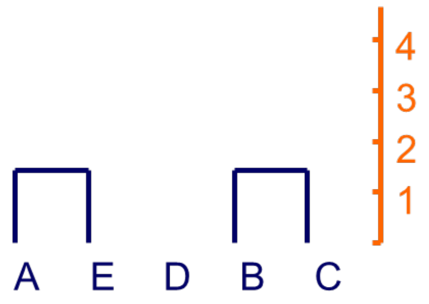


Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

UPGMA example

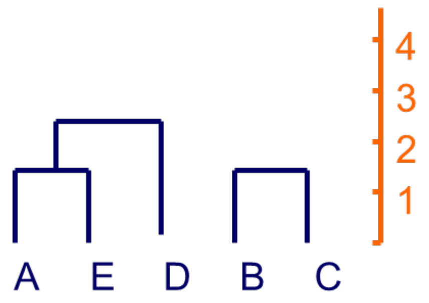
after two merges

	AE	BC	D
AE	0	8	5
BC		0	8
D			0

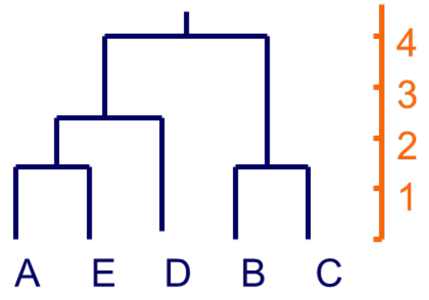


after three merges

	AED	BC
AED	0	8
BC		0



final state



Another distance-based algorithm: neighbor joining

- unlike UPGMA
  - does not make molecular clock assumption,
  - produces unrooted trees,
- it assumes **additivity**
  - distance between a pair of leaves is sum of lengths of edges connecting them,

    $\forall x, y, u, v\,(\text{leaves}) : d(x,y)+d(u,v) \leq max(d(x,u)+d(y,v), d(y,u)+d(x,v))$
- like UPGMA, constructs a tree by iteratively joining subtrees, however
  - the pair of subtrees to be merged on each iteration is selected differently,
  - distances are updated differently after each merge too.

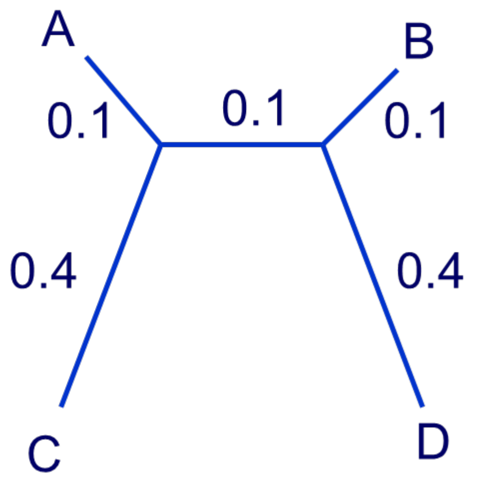
Picking pairs of nodes to join in neighbor joining (NJ)

- at each step, we pick a pair of nodes to join
  - should we pick a pair i and j with minimal distance d_{ij}?
- suppose the real tree below, we aim to pick the first pair of nodes to join
  - wrong decision to join A and B,
  - we need to consider distance of the pair to other leaves too,
  - pick a pair of nodes that minimizes D_{ij}

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

$$r_i = \frac{1}{|L| - 2} \sum_{k \in L} d_{ik}$$

where L is the set of leaves



$\boxed{d_{AB} = 0.3}$

$d_{AC} = 0.5$

$D_{AB} = -1.1$

$\boxed{D_{AC} = -1.2}$

Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

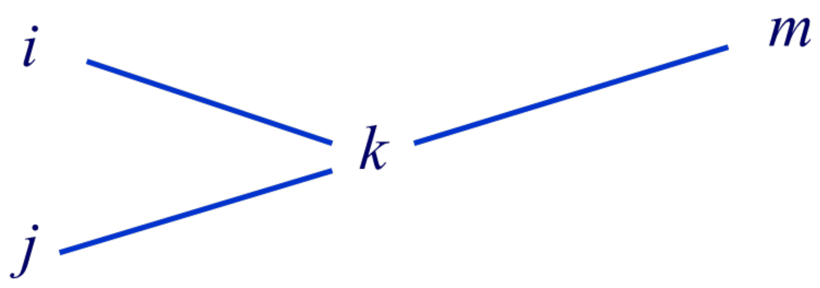
Updating distances in neighbor joining

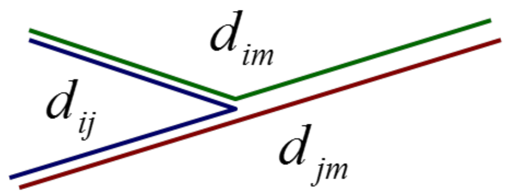
- the joined pair (i, j) will be replaced by a new internal node k,
- its distance to another node m is given by

$$d_{km} = \frac{1}{2}(d_{im} + d_{jm} - d_{ij})$$

- the distance from a leaf to its parent node calculated in the same way

$$d_{ik} = \frac{1}{2}(d_{ij} + d_{im} - d_{jm})$$





Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Updating distances in neighbor joining

- the previous update works perfectly if data are strictly additive,
- if not, a more robust method applies
  - instead of single m, take into account the distance to **all** other leaves

$$d_{ik} = \frac{1}{2}(d_{ij} + r_i - r_j)$$

$$r_i = \frac{1}{|L| - 2} \sum_{m \in L} d_{im}$$

  - where L is the set of leaves.

Neighbor joining algorithm

```
define the tree $T$ = set of leaf nodes
$L$ = $T$
while more than two subtrees in $T$
    pick the pair $i$, $j$ in $L$ with minimal $D_{ij}$
    add to $T$ a new node $k$ joining $i$ and $j$
    determine new distances $d_{ik}$, $d_{jk}$
    determine $d_{km}$ for all other $m$ in $L$
    remove $i$ and $j$ from $L$ and insert $k$
    (treat it like a leaf)
join two last subtrees $i$ and $j$ with edge of length $d_{ij}$
```

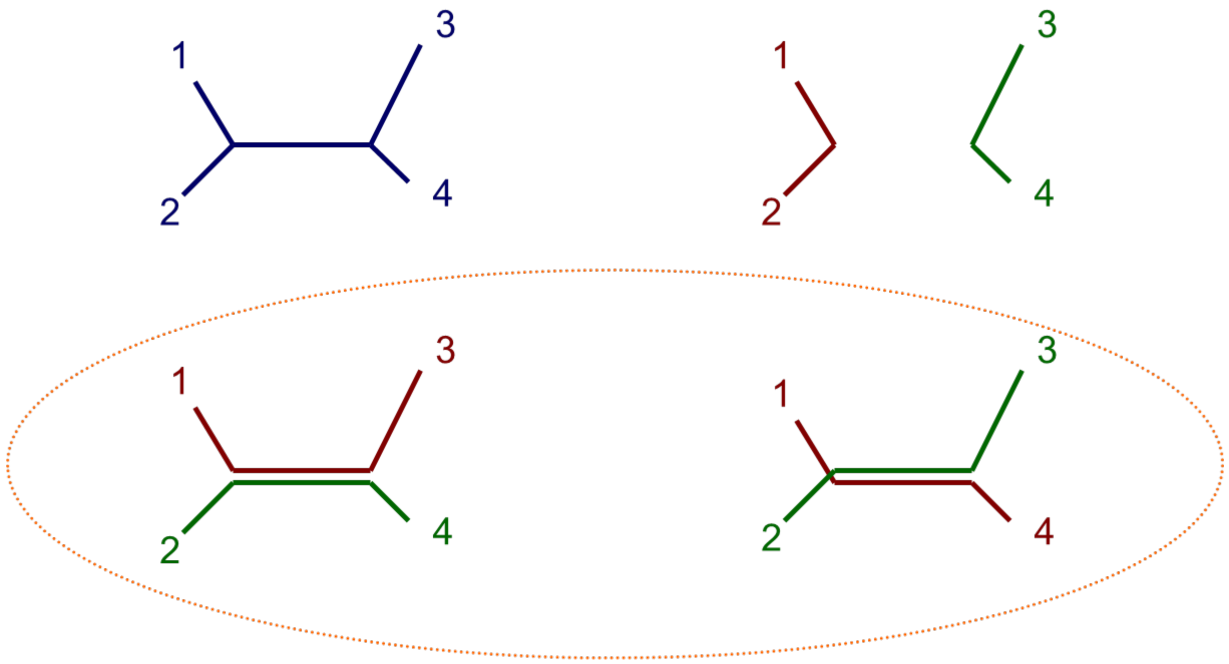
- if the data is additive (and these distances represent real distances), then neighbor joining will identify the correct tree,
- otherwise, the method may not recover the correct tree, but it may still be reasonable heuristics,
- neighbor joining is commonly used.

Testing for additivity

- remember the additivity property

  $\forall i, j, k, l$ (leaves) $: d(i,j) + d(k,l) \leq max(d(i,k) + d(j,l), d(i,l) + d(j,k))$

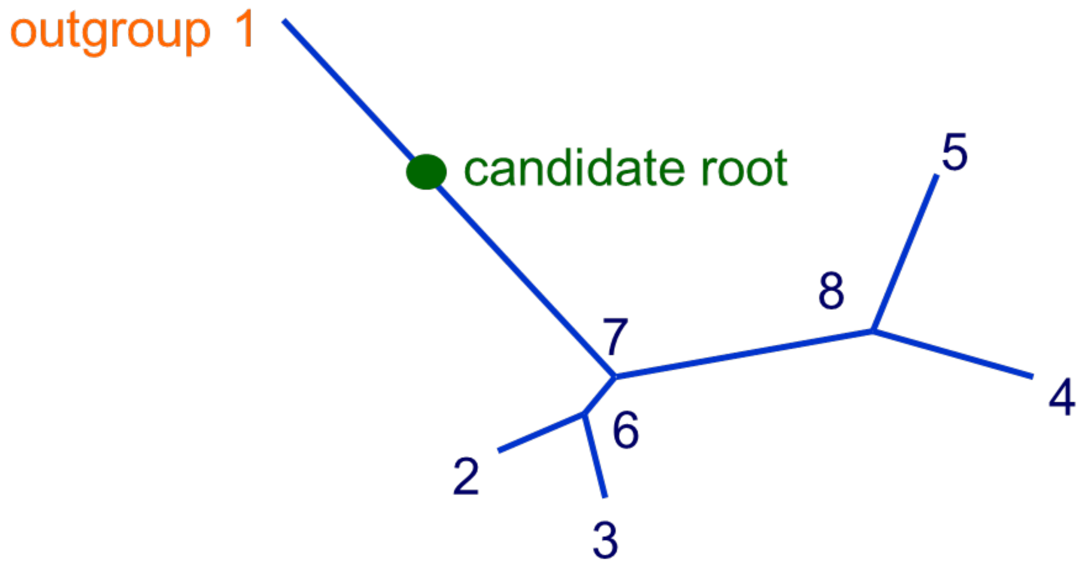
- for every set of four leaves, i, j, k, and l, two of the distances $d_{ij}+d_{kl}$, $d_{ik}+d_{jl}$ and $d_{il}+d_{jk}$ must be equal and not less than the third.



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Rooting trees

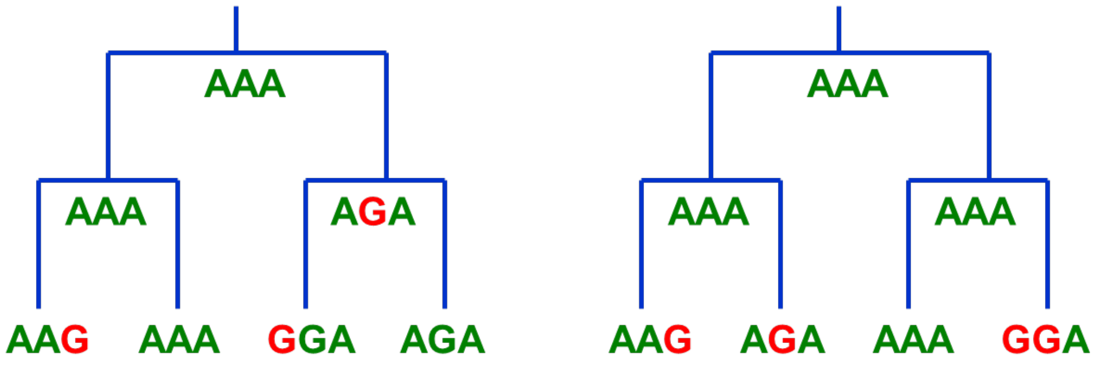
- finding a root in an unrooted tree sometimes accomplished with an **outgroup**,
  - a species known to be (far) more distantly related to remaining species,
- edge joining the outgroup to the rest of the tree is best candidate for root position,
- no outgroup $\rightarrow$ pick the midpoint of the longest chain of consecutive edges.



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Parsimony-based approaches

- parsimony: find the tree that explains the data with minimum changes,
- given: character-based data,
- do: find tree that explains the data with a minimal number of changes,
- focus is on finding the right tree topology, not on estimating branch lengths,
- there are various trees that could explain the phylogeny of the sequences **AAG**, **AAA**, **GGA**, **AGA** including the two below,
- parsimony prefers the first tree because it requires fewer substitution events.



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Parsimony-based approaches

- usually these approaches involve two separate components
  - a procedure to find the minimum number of changes needed to explain the data for a given tree topology
    - we will assume that positions are independent and compute the minimum number of changes for each position separately,
    - at first, we will treat the costs of these changes uniformly,
    - then, we will work with different costs for different changes,
  - a search through the space of trees
    - cannot be exhaustive, too many trees,
    - a heuristic method of nearest neighbor interchange.

Finding minimum number of changes for a given tree

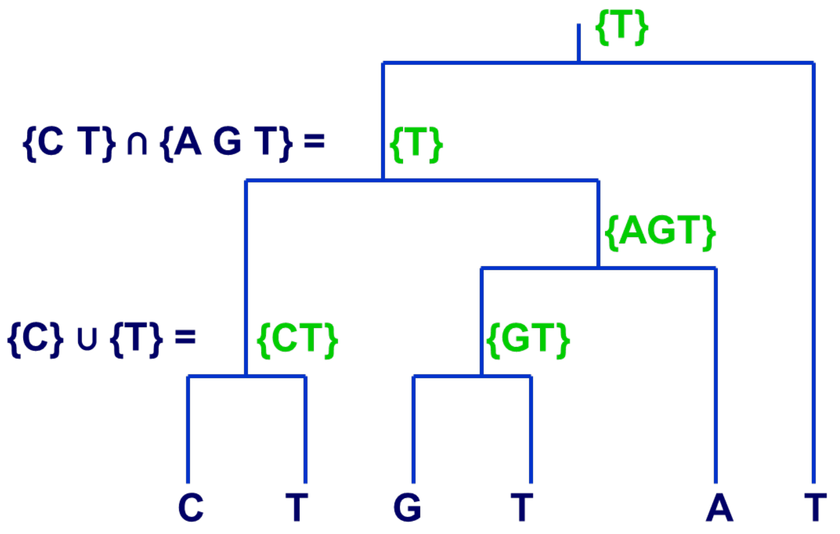
- brute force approach
  - for each possible assignment of states to the internal nodes, calculate the number of changes,
  - report the minimum number of changes found,
  - runtime is $\mathcal{O}(Nk^N)$
    k = number of characters (4 for DNA), N = number of leaves,
- Fitch's two-pass algorithm
  - firstly traverses tree from leaves to root determining set of possible states (e.g. nucleotides) for each internal node,
  - secondly traverses tree from root to leaves picking ancestral states for internal nodes,
  - deals with the uniform costs of changes,
  - finds the best assignment in $\mathcal{O}(Nk)$.

Fitch's algorithm: step 1 = post-order

- do a post-order (from leaves to root) traversal of tree,
- determine possible states R_i of internal node i with children j and k

$$R_i = \begin{cases} R_j \cup R_k, \text{ if } R_j \cap R_k = \emptyset \\ R_j \cap R_k, \text{ otherwise} \end{cases}$$

- this step calculates the number of changes required
  $\#$ of changes $=$ $\#$ union operations.

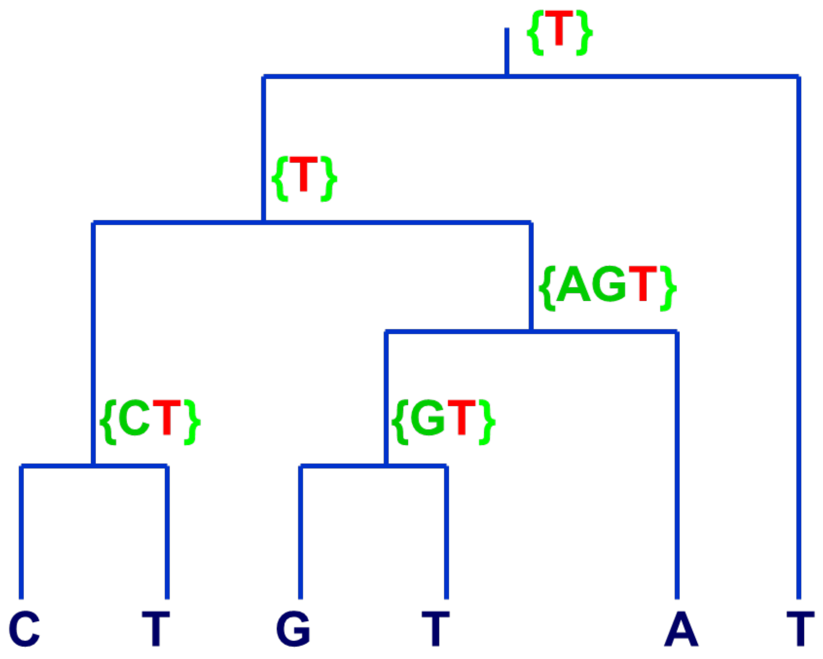


Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Fitch's algorithm: step 2 = pre-order

- do a pre-order (from root to leaves) traversal of tree,
- select state r_j of internal node j with parent i

$$r_j = \begin{cases} r_i, \text{ if } r_i \in R_j \\ \text{arbitrary state} \in R_j, \text{ otherwise} \end{cases}$$



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

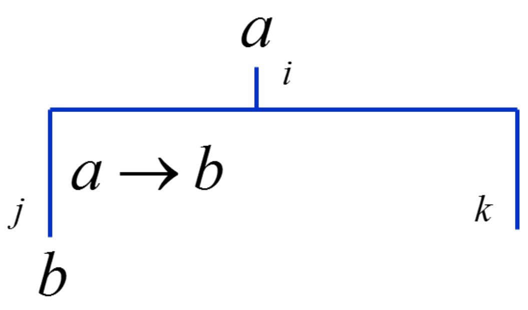
Weighted parsimony

- instead of assuming all state changes are equally likely, use different costs $S(a,b)$ for different changes,
- the modification of Fitch's algorithm proposed by Sankoff & Cedergren,
- 1st post-order step of algorithm propagates costs up through tree,
- the goal is to determine cost $R_i(a)$ of assigning character a to node i,
- for leaves

$$R_i(a) = \begin{cases} 0, & \text{if a is character at leaf,} \\ \infty, & \text{otherwise} \end{cases}$$

- for an internal node i with children j and k

$$R_i(a) = min_b(R_j(b) + S(a,b)) + min_b(R_k(b) + S(a,b))$$

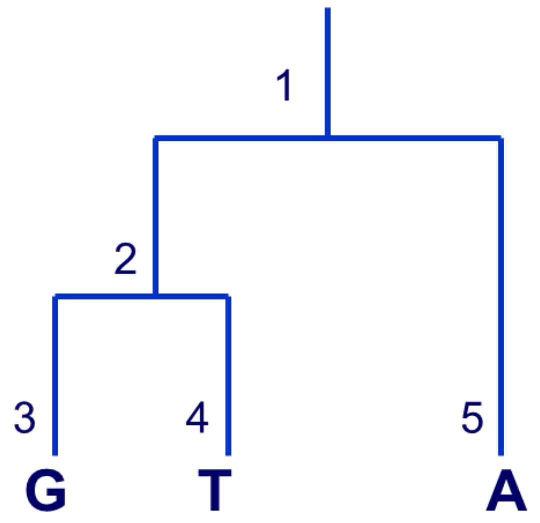


Example: weighted parsimony

$$R_3[A]=\infty, R_3[C]=\infty, R_3[G]=0, R_3[T]=\infty$$

$$R_4[A]=\infty, R_4[C]=\infty, R_4[G]=\infty, R_4[T]=0$$

$$R_2[A]=R_3[G]+S(A,G)+R_4[T]+S(A,T)$$
$$\vdots$$
$$R_2[T]=R_3[G]+S(T,G)+R_4[T]+S(T,T)$$



$$R_5[A]=0, R_5[C]=\infty, R_5[G]=\infty, R_5[T]=\infty$$

$$R_1[A]=\min\left(R_2[A]+S(A,A),\ \ldots\ ,\ R_2[T]+S(A,T)\right)\ +\ R_5[A]+S(A,A)$$
$$\vdots$$
$$R_1[T]=\min\left(R_2[A]+S(T,A),\ \ldots\ ,\ R_2[T]+S(T,T)\right)\ +\ R_5[A]+S(T,A)$$

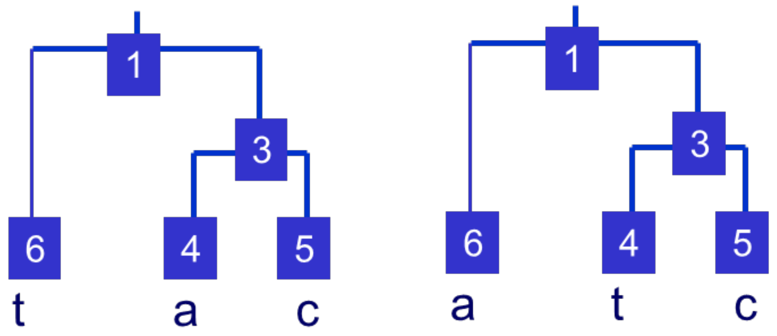
Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Weighted parsimony: step 2 = pre-order

- do a pre-order (from root to leaves) traversal of tree
  - for root node: select minimal cost character,
  - for each internal node: select the character that resulted in the minimum cost explanation of the character selected at the parent,

Example: weighted parsimony

- Consider the two simple phylogenetic trees shown below,
- and the symmetric cost matrix for assessing nucleotide changes,
- the tree on the right has a cost of 0.8,

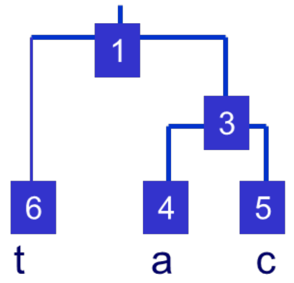


	a	c	g	t
a	0	0.8	0.2	0.9
c	0.8	0	0.7	0.5
g	0.2	0.7	0	0.1
t	0.9	0.5	0.1	0

Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

- show how the weighted parsimony determines the cost of the tree on the left,
- what are the minimal cost characters for the internal nodes in the tree?
- which of the two trees would the maximum parsimony approach prefer?

Example: weighted parsimony



	a	c	g	t
a	0	0.8	0.2	0.9
c	0.8	0	0.7	0.5
g	0.2	0.7	0	0.1
t	0.9	0.5	0.1	0

$$R_3(a) = 0 + 0.8 = 0.8$$
$$R_3(c) = 0.8 + 0 = 0.8$$
$$R_3(g) = 0.2 + 0.7 = 0.9$$
$$R_3(t) = 0.9 + 0.5 = 1.4$$

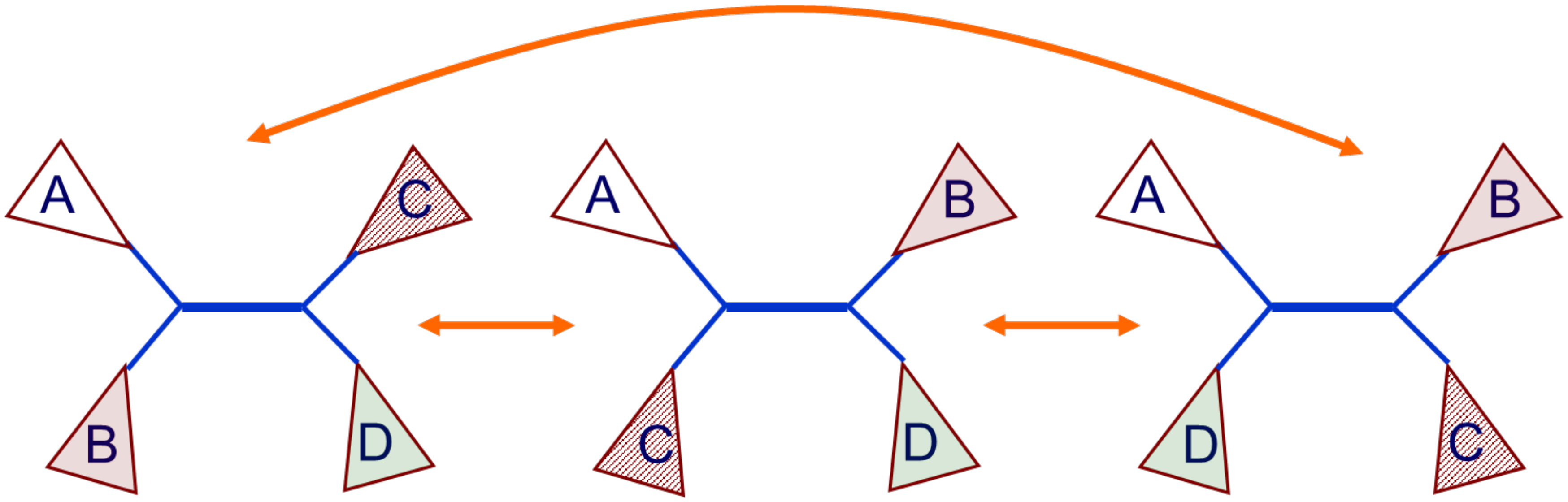
$$R_1(a) = 0.9 + \min\{0.8,\ \ 0.8+0.8,\ \ \boxed{0.2}+0.9,\ \ 0.9+1.4\} = 1.7$$
$$R_1(c) = 0.5 + \min\{0.8+0.8,\ \ 0.8,\ \ 0.7+0.9,\ \ 0.5+1.4\} = 1.3$$
$$R_1(g) = 0.1 + \min\{0.2+0.8,\ \ 0.7+0.8,\ \ 0.9,\ \ 0.1+1.4\} = 1.0$$
$$R_1(t) = 0 + \min\{0.9+0.8,\ \ 0.5+0.8,\ \ 0.1+0.9,\ \ 1.4\} = 1.0$$

Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

- The minimal cost characters for node 1 are either g or t. The tree costs 1.0.
- The minimal cost character for node 3 is g.
- The maximum parsimony approach would prefer the other tree (1.0>0.8).

Exploring the space of trees: nearest neighbor interchange

- For any internal edge in a tree
  - there are 3 ways the four subtrees can be grouped,
- nearest neighbor interchanges move from one grouping to another.
  - it represents the simplest tree structure rearrangement (only local).



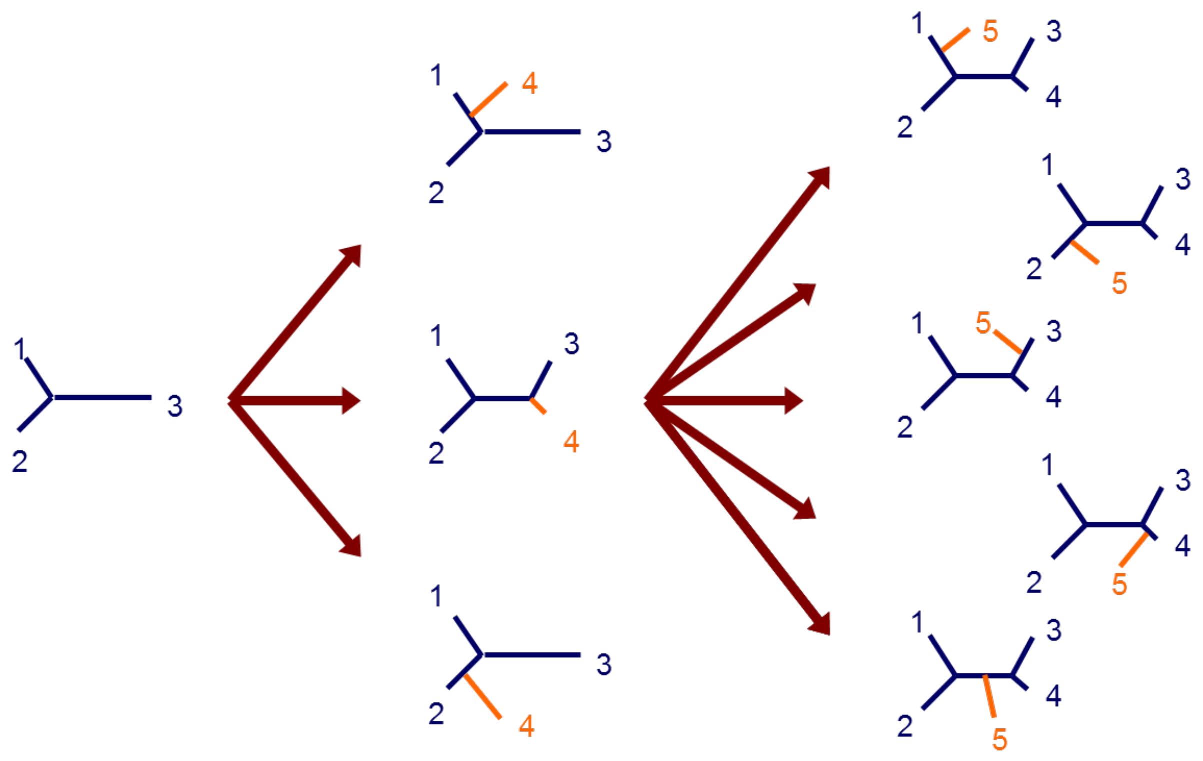
Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Heuristic hill-climbing with nearest neighbor interchange

```
given: set of leaves $L$
create an initial tree $t$ incorporating all leaves in $L$
best-score = parsimony algorithm applied to $t$
repeat
  for each internal edge $e$ in $t$
    for each nearest neighbor interchange
      $t' \leftarrow$ tree with interchange applied to edge $e$ in $t$
      score = parsimony algorithm applied to $t'$
      if score < best-score
        best-score = score
        best-tree = $t'$
  t = best-tree
until stopping criteria met
```

Exact method: branch and bound

- Each partial tree represents a set of complete trees,
- parsimony score on a partial tree = lower bound on the best score in the set,
- search by repeatedly selecting the partial tree with the lowest lower bound.



Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

Exact method: branch and bound

```
given: set of leaves $L$
initialize $Q$ with a partial tree with 3 leaves from $L$
repeat
  $t \leftarrow$ tree in $Q$ with lowest lower bound
  if $t$ has incorporated all leaves in $L$
    return $t$
  else
    create new trees
    (by adding next leaf from $L$ to each branch of $t$)
    compute lower bound for each tree
    put trees in $Q$ sorted by lower bound
```

Exact method: branch and bound (improved version)

```
given: set of leaves $L$
use heuristic method to grow initial tree $t'$
initialize $Q$ with a partial tree with 3 leaves from $L$
repeat
  $t \leftarrow$ tree in $Q$ with lowest lower bound
  if $t$ has incorporated all leaves in $L$
    return $t$
  else
    create new trees
    (by adding next leaf from $L$ to each branch of $t$)
    for each new tree $n$
      if lower-bound(n) < score(t')
      put $n$ in $Q$ sorted by lower bound
```

Comments on parsimony methods

- Branch and bound is a complete search method
  - guaranteed to find optimal solution,
- may be much more efficient than exhaustive search
- in the worst case, it is no better,
- branch and bound efficiency depends on
  - the tightness of the lower bound,
  - the quality of the initial tree,
- we described parsimony calculations in terms of rooted trees
  - but we described the search procedures in terms of unrooted trees,
  - it is not a big problem as
    - unweighted parsimony: minimum cost is independent of where root is located,
    - weighted parsimony: minimum cost is independent of root if substitution cost is a metric.

Probabilistic phylogenetic methods

- A probabilistic alternative to parsimony
  - instead of cost $S(a, b)$ of a substitution occurring along a branch, it uses a probability $P(child = a|parent = b)$,
  - for a given tree, instead of finding a minimal cost assignment to the ancestral nodes, it sums the probabilities of all possible ancestral states,
  - instead of finding a tree with minimum cost, it finds a tree that **maximizes likelihood** (probability of the data given the tree),
- this approach aims to minimize the effect of implicit parsimony simplifications
  - substitution costs are rather arbitrary and the most parsimonious tree critically depends on them,
  - parsimony methods require assignments of character states to the ancestral nodes, the best assignment does not have to be the true one.

Probabilistic model setup

- We observe n sequences $x^1, \dots, x^n$,
- we are given a tree T and want to model the **likelihood** $P(x^1, \dots, x^n | T)$
  - likelihood $=$ probability of observation (sequences) given model (tree),
- for simplicity, consider that our sequences are of length 1 (just one character),
- to generalize to longer sequences, assume independence of each position
  - each column of an ungapped multiple alignment tretaed independently,
  - probability of sequences $=$ product of probability of each position/column,
- the states of internal nodes given by random variables $X^{n+1}, \dots, X^{2n-1}$
  - assume a rooted binary tree,
- the branch lengths will be ignored for the sake of simplicity
  - $P(child = a | parent = b)$ instead of $P(child = a | parent = b, time)$.

Probabilistic model setup

- The probability of any particular configuration of states at all tree nodes

$$P(x^1, \dots, x^{2n-1}|T) = q_{x^{2n-1}} \prod_{i=1}^{2n-2} P(x^i|x^{\alpha(i)})$$

  - $q_{x^{2n-1}}$ is the prior probability of the state of the root node,
  - $\alpha(i)$ is the index of the parent node of node i,
- key assumption
  - state of node i is **conditionally independent** of the states of its ancestors given the state of its parent,
- we only care about the probability of the observed (extant) sequences
- need to marginalize (sum over possible values of ancestral states) to obtain the likelihood

$$P(x^1, \dots, x^n|T) = \sum_{x^{n+1},\dots,x^{2n-1}} q_{x^{2n-1}} \prod_{i=1}^{2n-2} P(x^i|x^{\alpha(i)})$$

Felsenstein's algorithm

- There is an exponential number of terms in the previous likelihood sum!
- dynamic programming to the rescue once again!
- subproblem: $P(L_k|a)$: probability of the leaves below node k, given that the residue at k is a,
- Recurrence:

$$\begin{aligned} P(L_k|a) &= \sum_{b,c} P(b|a)P(L_i|b)P(c|a)P(L_j|c) = \\ &= \sum_{b} P(b|a)P(L_i|b) \sum_{c} P(c|a)P(L_j|c) \end{aligned}$$

  - where i and j are the children nodes of k,
  - b and c represent the states of node i and node j, respectively.

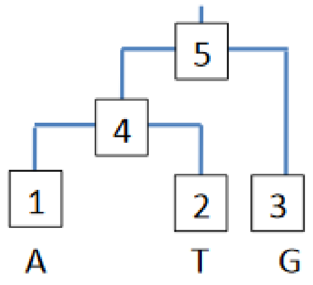
Felsenstein's algorithm

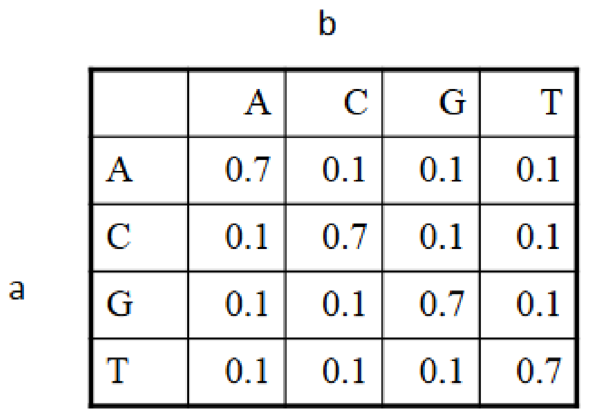
- Initialize: $k = 2n - 1$
- Recursion:
  - if k is a leaf node
$$P(L_k|a) = \begin{cases} 1, \text{ if } a = x^k \\ 0, \text{ otherwise} \end{cases}$$
  - else, compute $P(L_i|a)$ and $P(L_j|a)$ for all a at daughters i and j
$$P(L_k|a) = \sum_b P(b|a)P(L_i|b) \sum_c P(c|a)P(L_j|c)$$
- Termination:
  - likelihood is equal to
$$\sum_a P(L^{2n-1}|a)q_a$$

Concluding remarks on maximum likelihood methods

- Very similar to the weighted parsimony case,
- main differences are at
  - leaf nodes,
  - minimization versus summation for internal nodes,
- can it be used to infer ancestral states as well?
  - instead of summing, we would maximize,
  - as in the parsimony case, we would need to keep track of the maximizing assignment,
- substitution probabilities $P(a|b)$ can be derived from principled mathematical models and/or estimated from data.

What is probability for the following set of residues?





a \ b	A	C	G	T
A	0.7	0.1	0.1	0.1
C	0.1	0.7	0.1	0.1
G	0.1	0.1	0.7	0.1
T	0.1	0.1	0.1	0.7

Assume the above conditional probability matrix P(b|a) for all branches

	A	C	G	T
$P(L_1\|x)$	1	0	0	0
$P(L_2\|x)$	0	0	0	1
$P(L_3\|x)$	0	0	1	0
$P(L_4\|x)$	0.07	0.01	0.01	0.07
$P(L_5\|x)$	0.0058	0.0022	0.0154	0.0058

Marc Craven, BMI/CS 576, www.biostat.wisc.edu/bmi576.

What is probability for the following set of residues?

- In leaf nodes the simple 0/1 rule,
- in internal nodes

$$P(L_4|A) = P(A|A)P(L_1|A)P(T|A)P(L_2|T) = 0.7 \times 1 \times 0.1 \times 1 = 0.07$$

$$P(L_4|C) = P(A|C)P(L_1|A)P(T|C)P(L_2|T) = 0.1 \times 1 \times 0.1 \times 1 = 0.01$$

$\ldots$

  — the other options in the leaf nodes lead to trivial zero probabilities and do not influence the sum,

$$P(L_5|A) = P(G|A)P(L_3|G) \sum_{b \in \{ACGT\}} P(b|A)P(L_4|b) =$$
$$= 0.1 \times 1(0.7 \times 0.07 + 0.1 \times 0.01 + 0.1 \times 0.01 + 0.1 \times 0.07) = 0.0058$$

- the probability of residues given the tree (with the uniform nucleotide priors)

$$\sum_x P(L_5|x)q_x = 0.25(0.0058 + 0.0022 + 0.0154 + 0.0058) = 0.0073$$