Chapter 11

Phylogenetic Tree Construction Methods and Programs

Overview

1. Introduction
2. Introduction to Biological Databases
3. Pairwise Sequence Alignment
4. Database Similarity Searching
5. Multiple Sequence Alignment
6. Profiles and Hidden Markov Models
7. Protein Motifs and Domain Prediction
8. Gene Prediction
9. Promoter and Regulatory Element Prediction
10. Phylogenetics Basics
11. Phylogenetic Tree Construction Methods and Programs
12. Protein Structure Basics
13. Protein Structure Visualization, Comparison and Classification
14. Protein Secondary Structure Prediction
15. Protein Tertiary Structure Prediction
16. RNA Structure Prediction
17. Genome Mapping, Assembly and Comparison
18. Functional Genomics
19. Proteomics

Tree-building methods: two main categories

1. **Distance-based methods**

   A multiple sequence alignment (MSA) is first reduced to a distance matrix, which represents how (dis)similar each possible sequence pair is; the evolutionary tree is then constructed from the distances

2. **Character-based methods**

   Each position in the sequence alignment is analysed separately, as an "independently evolving unit" that undergoes random substitutions at a certain rate

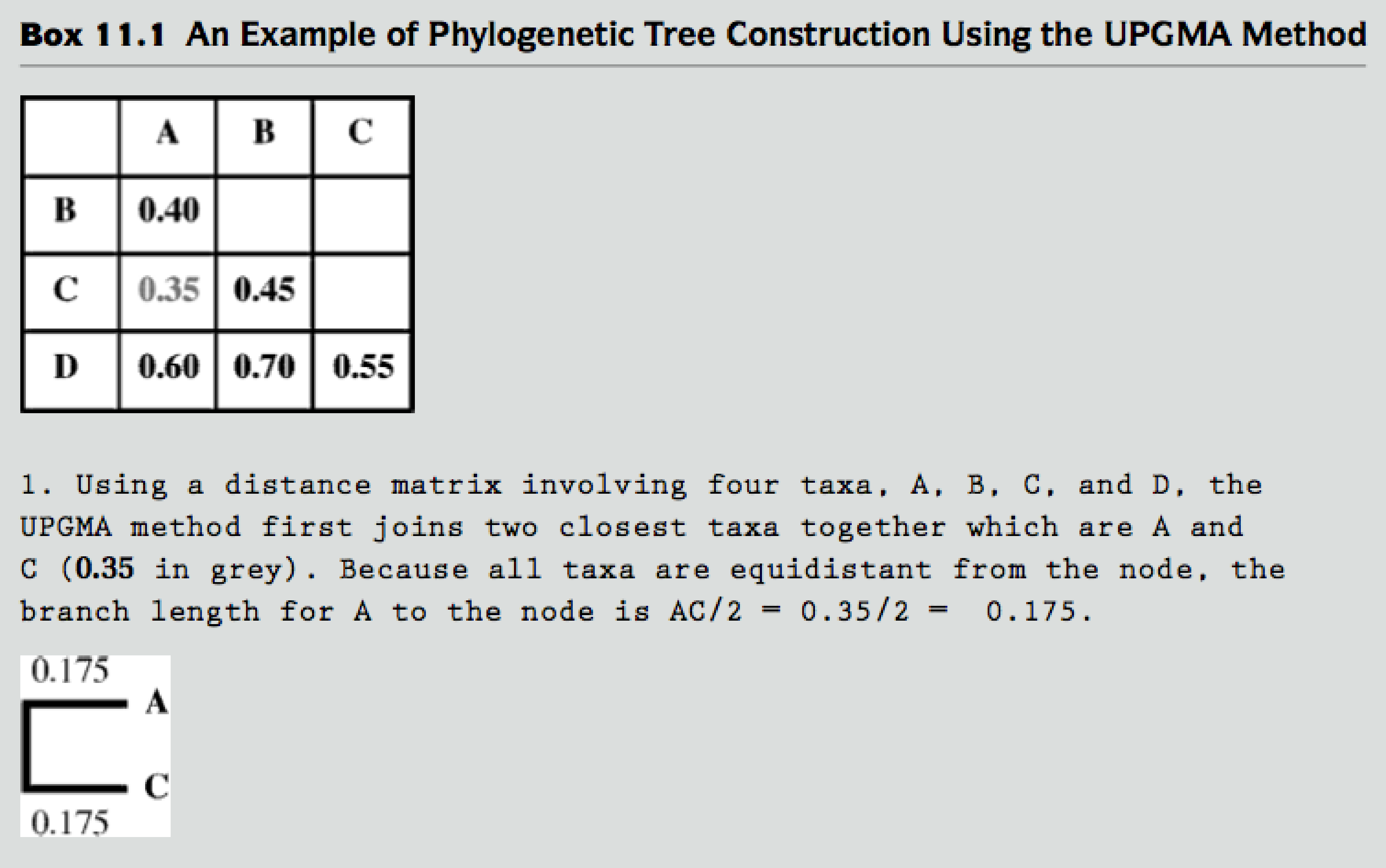
Distance-based methods

- Some kind of "evolutionary distance" between all individual pairs of taxa is computed (simplest distance: fraction of sequence positions that are non-identical)
- Statistical models may be applied to correct for homoplasy (multiple mutations, back-mutations, *etc.*)
- Phylogenetic tree construction from the matrix of pairwise distances:
    - Clustering-based methods: iteratively link the most similar sequence pairs to form a tree
    - Optimality-based methods: compare many/all alternative tree topologies and find "the best"

A simple clustering-based method: UPGMA

Unweighted Pair Group Method using Arithmetic average (UPGMA):

- Starts by grouping the two taxa with the smallest pairwise distance and placing a node at the midpoint between them
- A new "reduced" distance matrix is created in which the new cluster is treated as single taxon
- Iterations are continued until all taxa are placed on the tree
- Last taxon added is considered the outgroup, producing a rooted tree



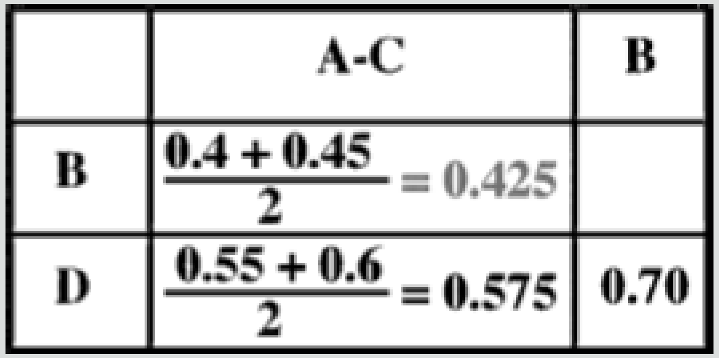
Box 11.1 An Example of Phylogenetic Tree Construction Using the UPGMA Method

	A	B	C
B	0.40		
C	0.35	0.45	
D	0.60	0.70	0.55

1. Using a distance matrix involving four taxa, A, B, C, and D, the UPGMA method first joins two closest taxa together which are A and C (**0.35** in grey). Because all taxa are equidistant from the node, the branch length for A to the node is AC/2 = 0.35/2 = 0.175.

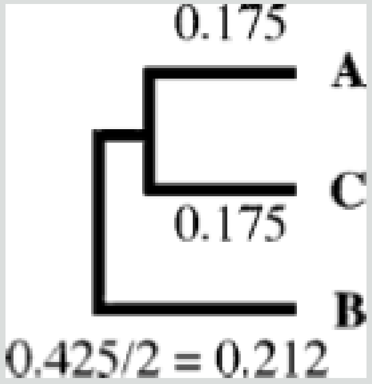
0.175
A
C
0.175

2. Because A and C are joined into a cluster, they are treated as one new composite taxon, which is used to create a reduced matrix. The distance of A-C cluster to every other taxa is one half of a taxon to A and C, respectively. That means that the distance of B to A-C is (AB + BC)/2; and that of D to A-C is (AD + CD)/2.



	A-C	B
B	$\frac{0.4 + 0.45}{2} = 0.425$	
D	$\frac{0.55 + 0.6}{2} = 0.575$	0.70

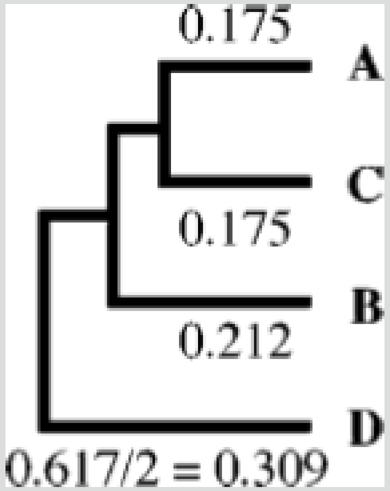
3. In the newly reduced-distance matrix, the smallest distance is between B and A-C (in grey), which allows the grouping of B and A-C to create a three-taxon cluster. The branch length for the B is one half of B to the A-C cluster.

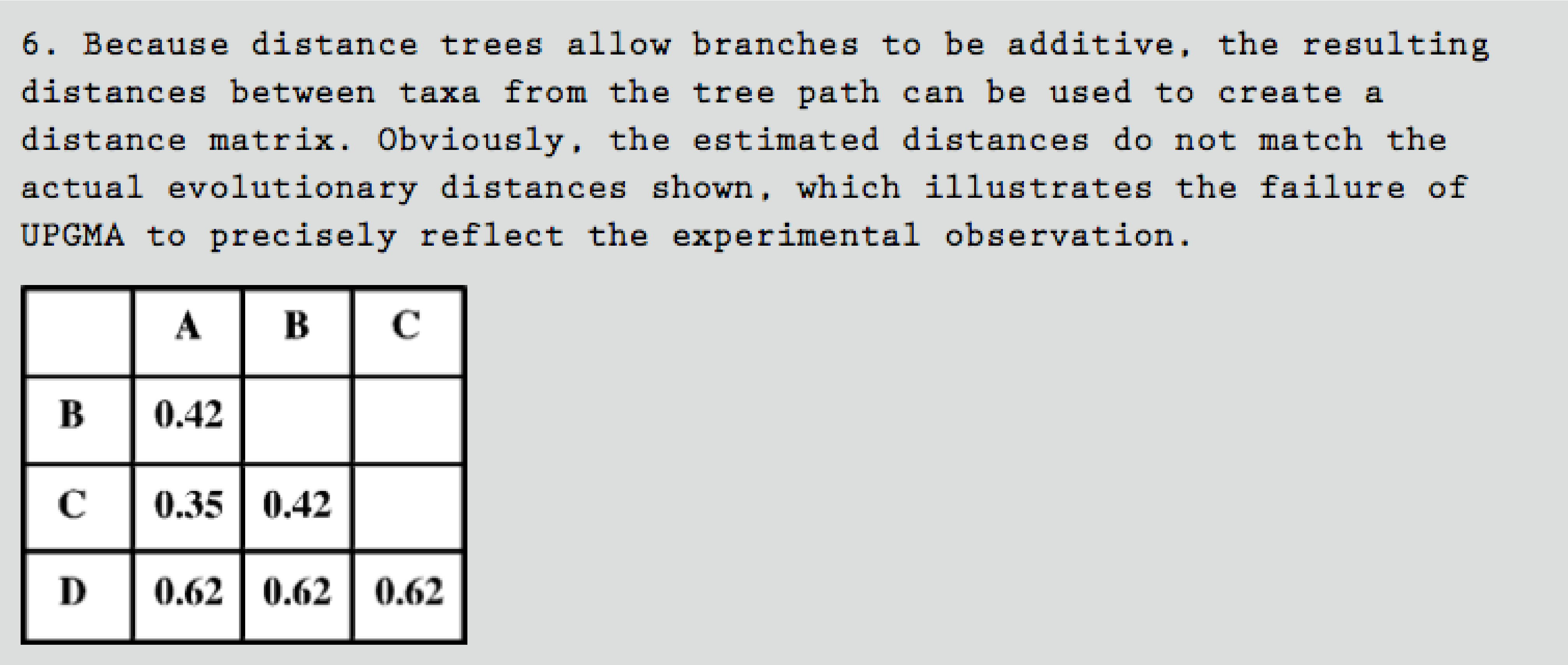


4. When B and A-C are grouped and treated as a single taxon, this allows the matrix to reduce further into only two taxa, D and B-A-C. The distance of D to the composite taxon is the average of D to every single component which is (BD + AD + CD)/3.

	B-A-C
D	$\frac{0.7 + 0.6 + 0.55}{3} = 0.617$

5. D is the last branch to add to the tree, whose branch length is one half of D to B-A-C.





6. Because distance trees allow branches to be additive, the resulting distances between taxa from the tree path can be used to create a distance matrix. Obviously, the estimated distances do not match the actual evolutionary distances shown, which illustrates the failure of UPGMA to precisely reflect the experimental observation.

	A	B	C
B	0.42		
C	0.35	0.42	
D	0.62	0.62	0.62

Properties and limitations of UPGMA

- Assumes that all taxa evolve at a constant rate so that they are equally distant from the root
- Implies a molecular clock is in effect
- Real data rarely meet this assumption, thus UPGMA often produces erroneous tree topologies
- Major advantage: UPGMA is extremely fast (the computations are very simple)

Clustering-based methods: NJ

Principle of the Neighbour Joining (NJ) method:

- As with UPGMA, the tree is built by stepwise reduction of a distance matrix
- However: NJ does not assume the taxa to be equidistant from the root, *i.e.* no molecular clock assumption is made
- NJ attempts to correct for unequal evolutionary rates by also taking into account distances to all other taxa when calculating a pairwise distance
- The idea is that if certain taxa are very different from ***everything else***, they must have evolved faster, therefore their true evolutionary distances are likely to be smaller

Clustering-based methods: NJ

Instead of a simple pairwise distance measure, NJ uses "converted" (*i.e.* corrected) distances:

- Converted distance between A and B:
  $d'_{AB} = d_{AB} – 1/2 \times (r_A + r_B)$
- The r-values are a measure of the distance to everything else:
  $r_i = \Sigma d_{ij}$ (= sum of distances of i to all other taxa)

Clustering-based methods: NJ

Branch lengths in the tree are also corrected for differences in evolution speed, using "transformed" r-values, r'_i:

$r'_i = r_i / (n - 2)$

used to determine the distance of an individual taxon to the nearest node:

$d_{AU} = (d_{AB} + (r'_A - r'_B)) / 2$

U … node formed by A and B

Clustering-based methods: NJ

- NJ starts with completely unresolved "star" tree (with all taxa directly connected to a central node) and progressively decomposes it by selecting pairs of taxa = *star decomposition*
- Therefore, NJ produces unrooted trees

Clustering-based methods: generalised NJ

Generalised Neighbor Joining:

- Neighbor Joining generates only one tree and does not test other tree topologies that may actually be better
- Generalised Neighbor Joining generates multiple NJ trees with different initial taxon groupings
- Tree that best fits the actual evolutionary distances is selected from the pool of NJ trees
- Better chance of finding the best tree

Optimality-based methods

Optimality-based methods compare all possible tree topologies and select "the best":

Two types of optimality-based method, based on different evaluation criteria to determine what is "best":

- Fitch-Margoliash
- Minimum evolution

Optimality-based methods: Fitch-Margoliash (FM)

- Tries to achieve minimal deviation between the distances calculated in the overall branches in the tree and the distances in the original dataset
- Uses a least-squares evaluation criterion:

$$E = \sum_{\mathrm{i}=1}^{\mathrm{T}-1} \sum_{\mathrm{j}=\mathrm{j}+1}^{\mathrm{T}} \frac{(d_{\mathrm{ij}} - p_{\mathrm{ij}})^2}{{d_{\mathrm{ij}}}^2}$$

E … error of the estimated tree
T … number of taxa
d_{ij} … original pairwise distance
p_{ij} … corresponding tree branch length

Optimality-based methods: Minimum Evolution (ME)

- Looks for the tree with a minimum overall branch length
- In other words: ME looks for the simplest tree (smallest distances, *i.e.* lowest number of mutations) that still explains the data
- Optimality criterion: minimise S, with

  $S = \Sigma b_i$
- May slightly outperform the least-squares-based Fitch-Margoliash method

Clustering-based *vs* optimality-based methods

Clustering-based methods:

- Computationally fast
- No guarantee that the best tree is found
- Most popular, very often used as a "quick-and-dirty" approach

Exhaustive tree-searching (optimality-based):

- Better overall accuracy
- Computational cost prohibitive for larger number of taxa (> 10-20)
- Tends to be inferior to (also computationally expensive) character-based methods

Character-based methods

- Also called *discrete* methods
- Involve a detailed analysis of mutational events, rather than (inevitably simplistic!) distance measures
- Evolutionary dynamics of each character (nucleotide or amino acid) can be studied
- Likely ancestral sequences can be inferred
- Most popular approaches:
  - Maximum parsimony (MP)
  - Maximum likelihood (ML)

Maximum Parsimony (MP)

Based on "Occam's razor" (William of Occam 1288 – 1347)

- Assumes that the simplest explanation is the correct one (or at least the one we ought to prefer)
- Simplest explanation requires the fewest assumptions
- Choosing the simplest model helps to "shave off“ variables that are not really necessary to explain the phenomenon

Maximum Parsimony (MP): the principle

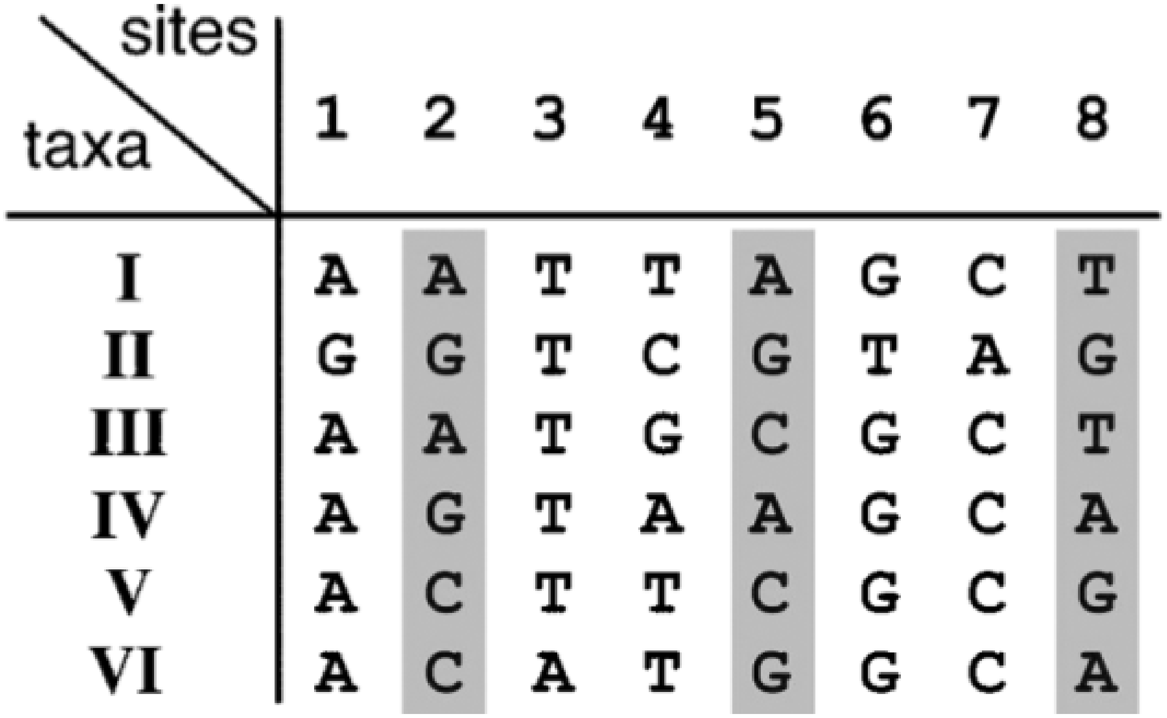
- MP chooses the tree that has the fewest evolutionary changes, in other words: the shortest overall branch lengths
- This choice is justified by the fact that evolutionary changes are relatively rare within a fairly short time frame, *i.e.* going from one node to the next
- The principle is very similar to that of the Minimum Evolution (ME) method, but the MP is character-based rather than distance-based

Maximum Parsimony (MP): informative sites

- We need to investigate all possible tree topologies and reconstruct ancestral sequences that require the smallest number of changes to evolve to the current sequences: this is a combinatorial nightmare!

- To save computing time, only a small number of sites that have the richest phylogenetic information are used in tree determination (= "informative sites")
  - Informative sites: sites that have at least two different kinds of characters, each occurring at least twice
  - Non-informative sites: constant sites or sites that have changes occurring only once, or no changes at all

Example of informative *vs* non-informative sites



sites taxa	1	2	3	4	5	6	7	8
I	A	A	T	T	A	G	C	T
II	G	G	T	C	G	T	A	G
III	A	A	T	G	C	G	C	T
IV	A	G	T	A	A	G	C	A
V	A	C	T	T	C	G	C	G
VI	A	C	A	T	G	G	C	A

Figure 11.1: Example of identification of informative sites that are used in parsimony analysis. Sites 2, 5, and 8 (*grey boxes*) are informative sites. Other sites are noninformative sites, which are either constant or having characters occurring only once.

Maximum Parsimony (MP): the algorithm

1. Calculate the minimum number of substitutions at each informative site for a given tree topology
2. Calculate the sum of minimum number of changes over all informative sites
3. Repeat 1. and 2. for every possible tree topology
4. Choose the tree that has the smallest total number of changes

Finding the minimal number of substitutions

How to reconstruct ancestral nodes that result in the smallest number of substitutions in an entire tree:

1. Going from the leaves to the internal nodes and ultimately to the common root, at each node a list of all possible characters is assembled, assuming that (at least) one of the two branches will have inherited the ancestral character (parsimony!)
2. Then, going from the root towards the leaves, the character that results in the smallest number of substitutions in the tree is chosen from the list at each node

Note: sometimes several solutions might be equally good

Maximum Parsimony (MP)

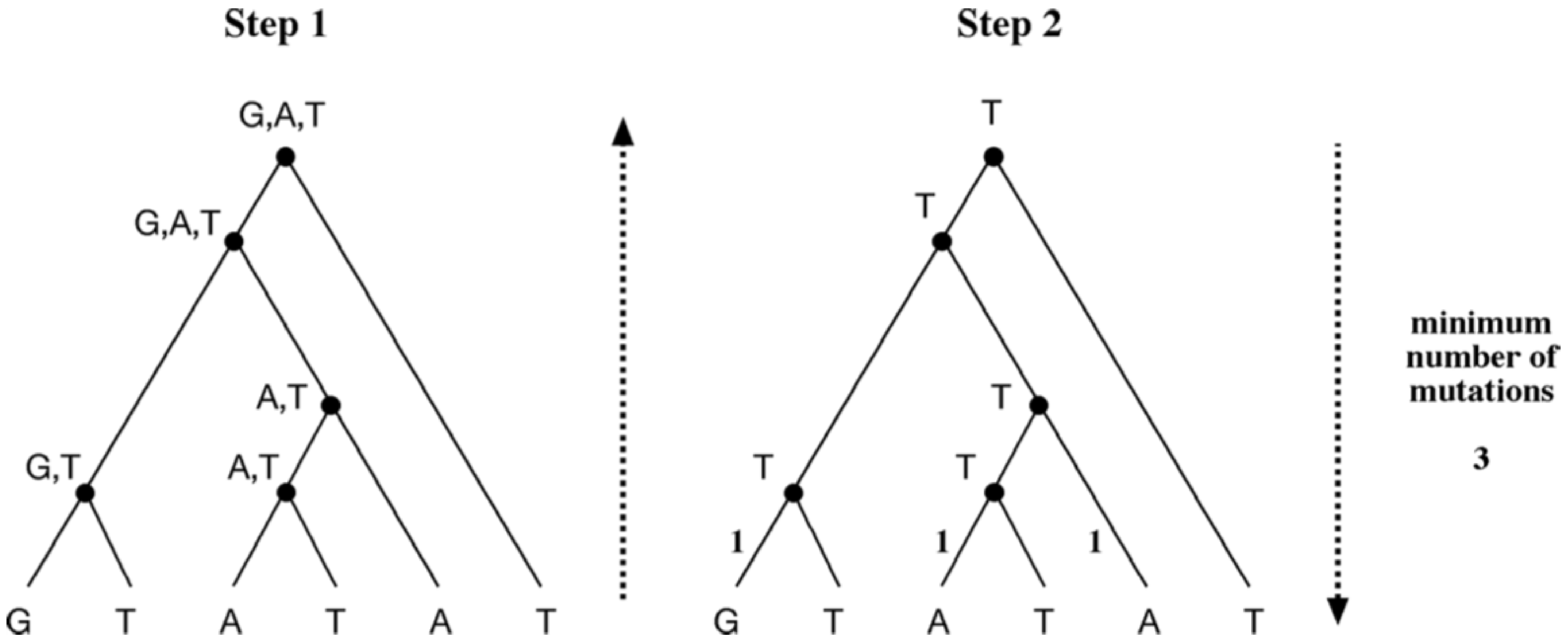


Figure 11.2: Using parsimony to infer ancestral characters at internal nodes involves a two-step procedure. The first step involves going from the leaves to the root and counting all possible ancestral characters at the internal nodes. The second step goes from the root to the leaves and assigns ancestral characters that involve minimum number of mutations. In this example, the total number of mutations is three if T is at the root, whereas other possible character states increase that number.

Weighted parsimony

- The standard parsimony method treats all mutations as equivalent
- However, some mutations are less frequent than others, *e.g.* transversions *vs* transitions in DNA/RNA and certain amino acid substitutions *vs* others (as statistically described by substitution matrices)
- Weighting scheme can take into account the likelihood of different kinds of mutations in determining the most probable node sequences

Tree searching

- Parsimony method is an inherently exhaustive approach, *i.e.* all possible tree topologies have to be investigated
- Starts from a three taxa unrooted tree, for which only one topology is possible
- A fourth taxon is added, producing three possible topologies, and so on
- This brute-force approach only works with relatively few sequences, as it is computationally too demanding for more than ~ 10 taxa

Exhaustive tree-searching: the problem

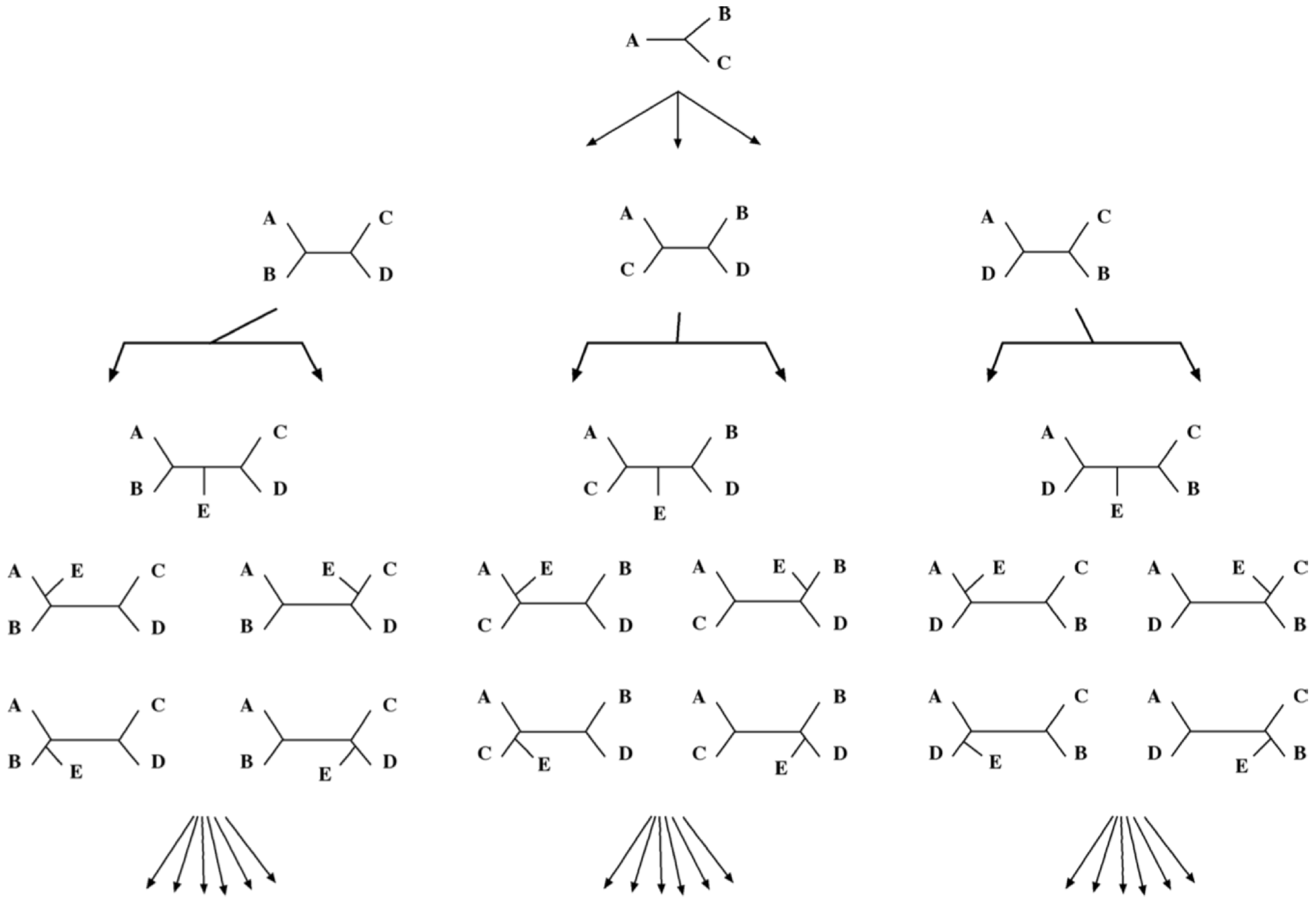


Figure 11.4: Schematic of exhaustive tree construction in the MP procedure. The tree starts with three taxa with one topology. One taxon is then added at a time in an progressive manner, during which the total branch lengths of all possible topologies are calculated.

Simplifying the tree search

Simplification measures have to be introduced to reduce the complexity of the search whenever the number of taxa becomes prohibitively large (*i.e.* almost always)

For instance, the Branch-and-bound algorithm:

- Builds a distance tree for all taxa involved using either NJ or UPGMA
- Uses the minimum number of substitutions for this tree as <u>upper limit</u> for the number of allowed sequence variations (maximally parsimonious tree must be equal or better than the distance-based tree)
- Search is abandoned in directions that exceed the limit

Branch-and-bound algorithm

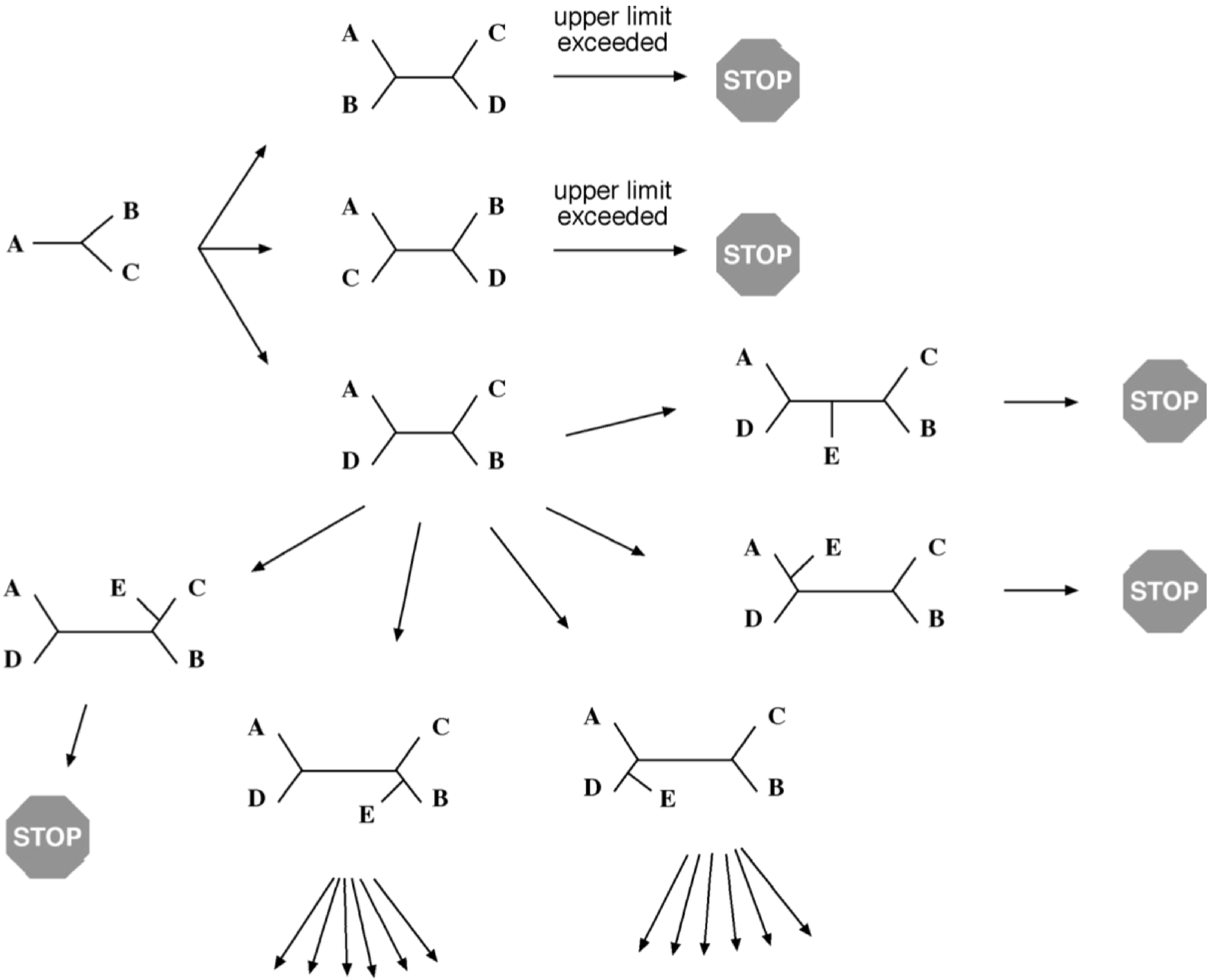


Figure 11.5: Schematic illustration of the branch-and-bound algorithm. Tree building starts with a step-wise addition of taxa of all possible topologies. Whenever the total branch length for a given topology exceeds the upper bound, the tree search in that direction stops, thereby reducing the total computing time.

Heuristic tree-searching methods

When the number of taxa exceeds 20, even the branch-and-bound method becomes computationally unfeasible

A heuristic tree search has to be used:

- Only a small subset of all possible trees is examined
- Quick-and-dirty initial tree by NJ or UPGMA
- Subsequently modified by cutting (*pruning*) a branch and *regrafting* it to other parts of the tree
- If the new tree turns out to be better, it is used as a starting point for another round of rearrangement

Heuristic tree search using branch swapping

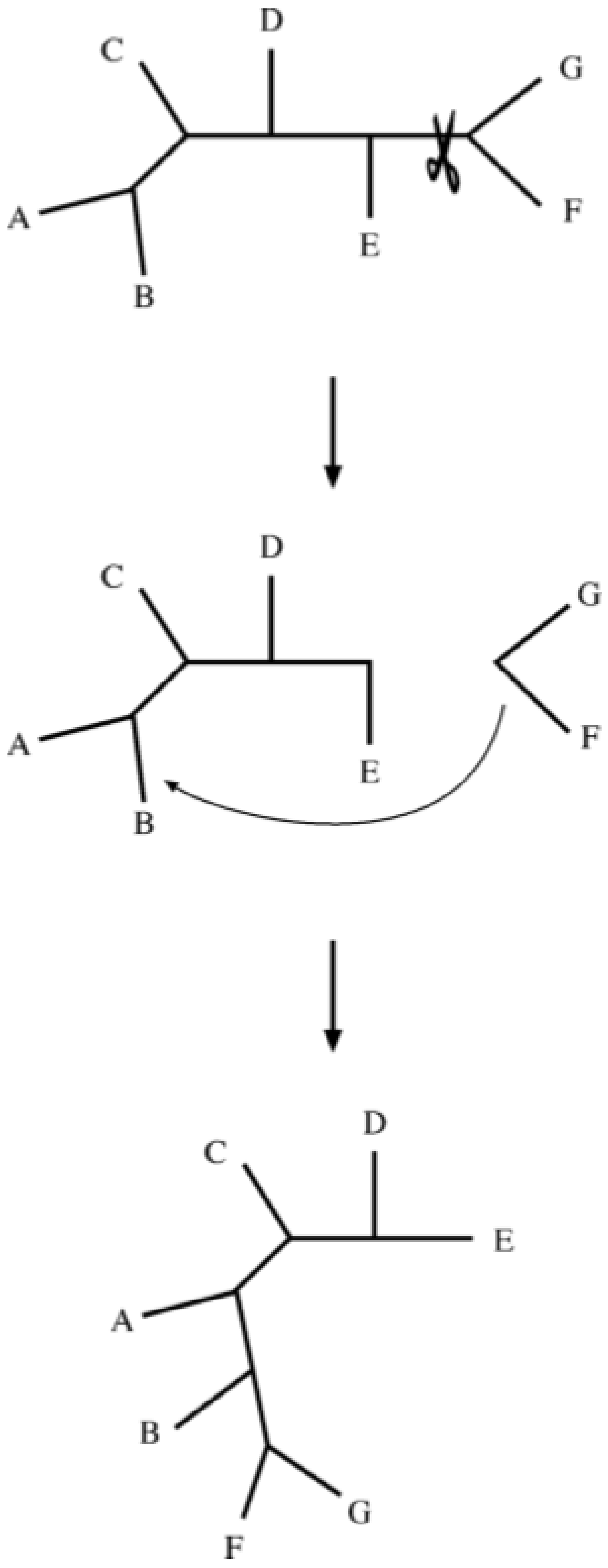


Figure 11.6: Schematic representation of a typical branch swapping process in which a branch is cut and moved to another part of the tree, generating a new topology.

Pros and cons of heuristic tree searching

- Much faster than exhaustive search
- However: no guarantee that heuristic searches will find the most parsimonious tree
- Tree rearrangement tends to focus on a local area and stalls when a local branch length minimum is reached (the *greedy algorithm* problem)

Advantages of Maximum Parsimony (MP)

- Relatively simple and intuitive
- Provides evolutionary information about the sequence characters and reconstructs ancestral sequences at the nodes
- Tends to produce more accurate trees than distance-based methods, especially when sequence divergence is low

Disadvantages of Maximum Parsimony (MP)

- Extremely slow
- Original parsimony assumption may not always hold, particularly when sequence divergence is high
- Estimation of branch length may be inaccurate because Maximum Parsimony does not correct for (or even consider) homoplasy, *i.e.* multiple substitutions
- Very sensitive to "long-branch attraction" (LBA) artifacts

Long-Branch Attraction (LBA)

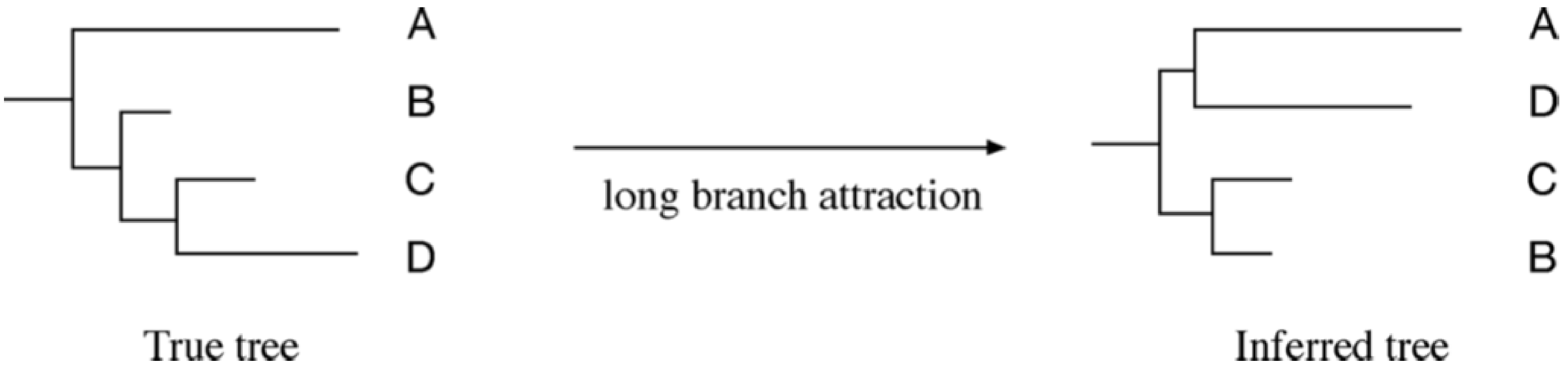


Figure 11.7: The LBA artifact showing taxa A and D are artifactually clustered during phylogenetic construction.

The principle of Maximum Likelihood (ML)

- In scientific research, there is often no direct way to evaluate the chance that a model that we have made to explain our experimental observations is actually correct; this is unfortunate, because that is precisely what we would like to know!
- On the other hand, it may very well be possible to calculate the likelihood that a given model (assumed to be correct) would lead to our observations!
- The idea behind ML: the model that has the highest chance of (re)producing our observations is presumably the best one we can come up with

Phylogenetic analysis using Maximum Likelihood (ML)

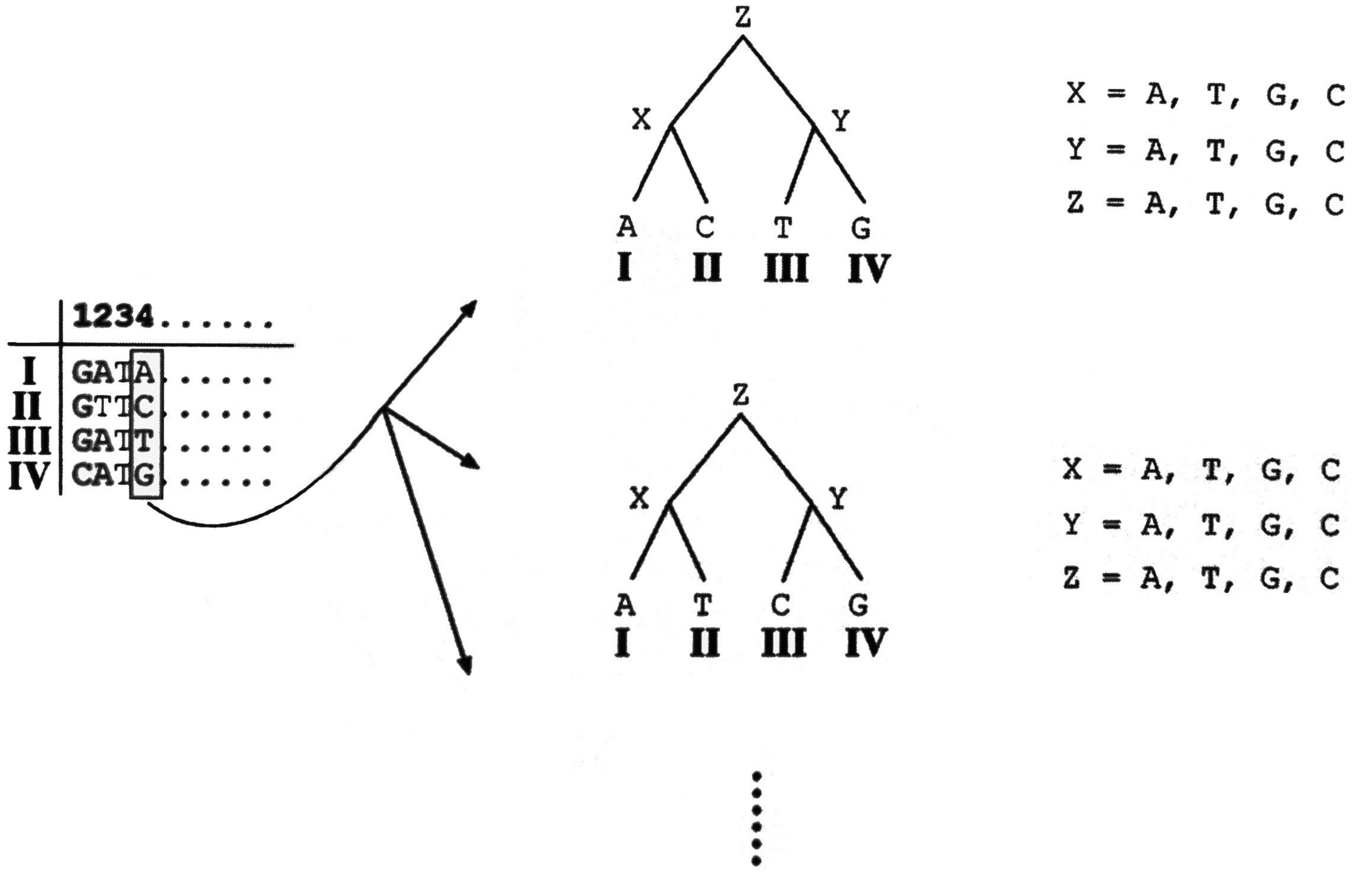
- Like MP, this approach exhaustively evaluates all possible tree topologies to find the best, but according to a maximum likelihood criterion
- A substitution model (such as an amino acid substitution probability matrix) is used to calculate the total likelihood that a given tree leads to the observed data (*i.e.* the modern-day sequences)
- May also include parameters to account for rate variations across sites
- The tree with the highest probability of accounting for the modern-day sequences is then chosen

Phylogenetic analysis using Maximum Likelihood (ML)

- For a particular site, the probability of a tree is the product of the probabilities of all bifurcation events
- It is computationally more convenient to express all probability values as log likelihood values, as multiplication often results in very small numbers
- After logarithmic conversion the likelihood score for the tree is the sum of the log likelihood values for every bifurcation event in the tree

Phylogenetic analysis using Maximum Likelihood (ML)

The ML approach requires evaluating all possible tree topologies and all possible node characters!



```
    | 1234......
I   | GATA......
II  | GTTC......
III | GATT......
IV  | CATG......
```

X = A, T, G, C

Y = A, T, G, C

Z = A, T, G, C

X = A, T, G, C

Y = A, T, G, C

Z = A, T, G, C

The Maximum Likelihood (ML) method: pros and cons

- Extremely time-consuming (worst of all methods!): even with modest numbers of taxa (10-20), heuristic approaches have to be used (like with MP)

- Accuracy (of course) depends on the substitution model

- However: ML is arguably the most rigorous approach mathematically speaking

- Often the method of choice nowadays, usually in combination with heuristic tree-search methods such as NJ plus pruning-and-regrafting to keep the computations manageable

Phylogenetic tree evaluation: bootstrapping

The principle of bootstrapping:

- Assess robustness of the original tree by repeatedly calculating trees from "perturbed" MSAs that do not contain all of the original positions
- A robust phylogenetic relationship will have enough information in the remaining positions to support the evolutionary relationships
- If perturbations lead to different trees, the original phylogeny may depend on noise rather than on a true evolutionary signal

Bootstrapping: the procedure

- A new MSA is generated by randomly selecting columns (positions) from the original MSA
- The new MSA has the same number of columns as the original one
- Columns from the original MSA may be selected multiple times, just once, or not at all
- A tree is generated from the new ("perturbed") MSA

Bootstrapping: the procedure

- The bootstrapping process is repeated 100 to 1 000 times
- All the bootstrapped trees may be combined into a consensus tree based on majority rule
- Nodes are labeled with the percentage of appearance
- Result provides a measure for evaluating the confidence levels of the tree topology
- Bootstrap values of 70% (empirically) correspond to 95% statistical confidence that a node is real

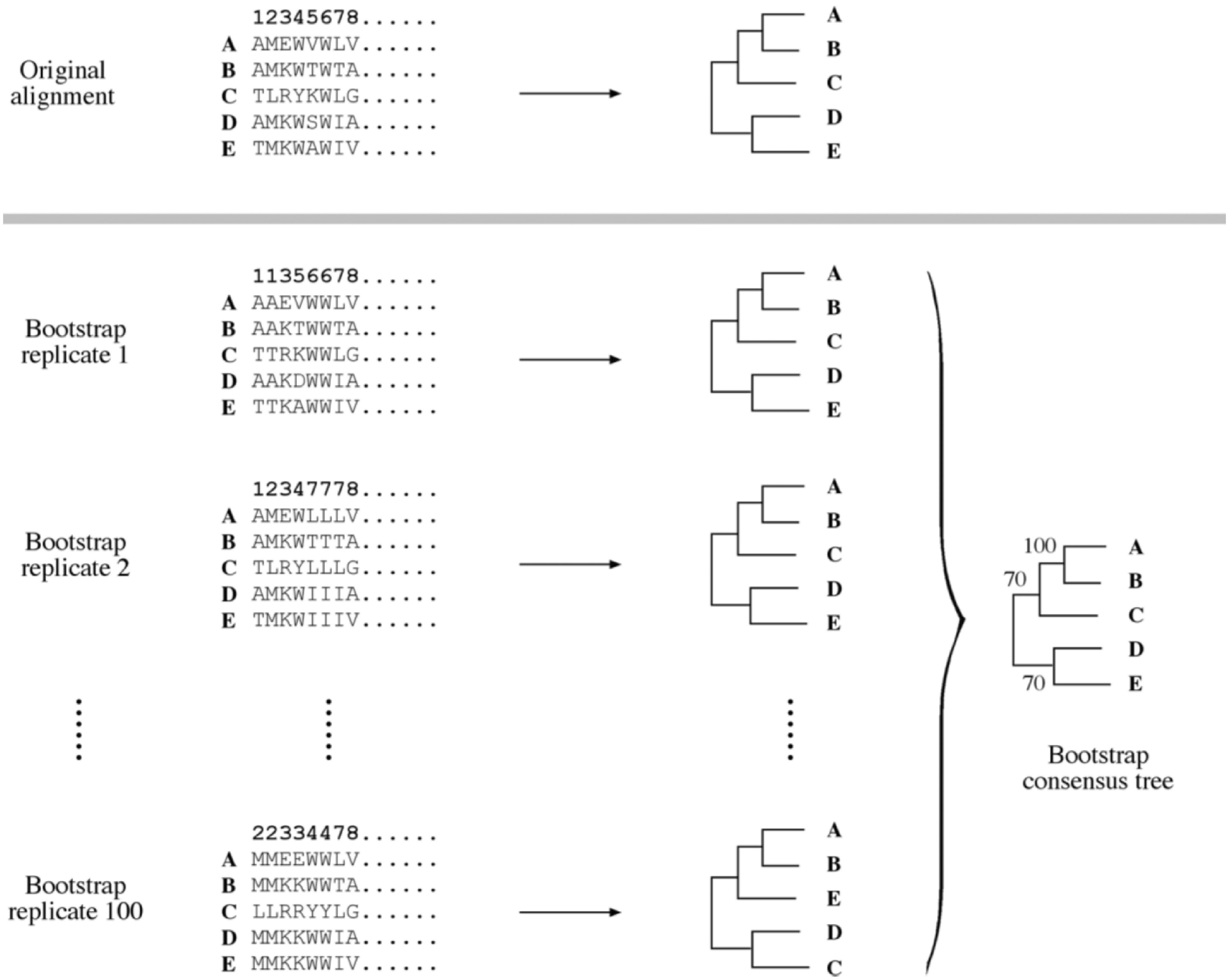


Figure 11.10: Schematic representation of a bootstrap analysis showing the original alignment and modified replicates in which certain sites are randomly replaced with other existing sites. The resulting altered replicates are used to building trees for statistical analysis at each node.

A variant of bootstrapping: jackknifing

- Half of the sites in the MSA are randomly deleted, creating an MSA half as long as the original
- As with bootstrapping, each new MSA is subjected to phylogenetic tree construction and results compared
- Advantages:
  - No sites are duplicated
  - Computing time is much shorter because of sorter sequences

Bootstrapping: caveats

- Strictly speaking, bootstrapping does not assess the accuracy of a tree, but only indicates consistency and stability
- Large number of bootstrap resampling steps are needed to achieve meaningful results (500 to 1 000 are recommended), which is computationally expensive

Statistical tests

Various statistical tests to determine if one tree is significantly better than another in terms of parsimony or likelihood have been developed:

- Kishino-Hasegawa for MP trees
- Shimodaira-Hasegawa for ML trees

Phylogenetic programs

Numerous servers and downloadable programs are available, most of them free of charge

A few examples:

- NGPhylogeny (completely automated web server, https://ngphylogeny.fr)
- Phylip (downloadable program suite, https://phylipweb.github.io/phylip)
- MEGA (downloadable package with GUI, https://www.megasoftware.net)
- MrBayes (downloadable program suite, https://nbisweden.github.io/MrBayes)