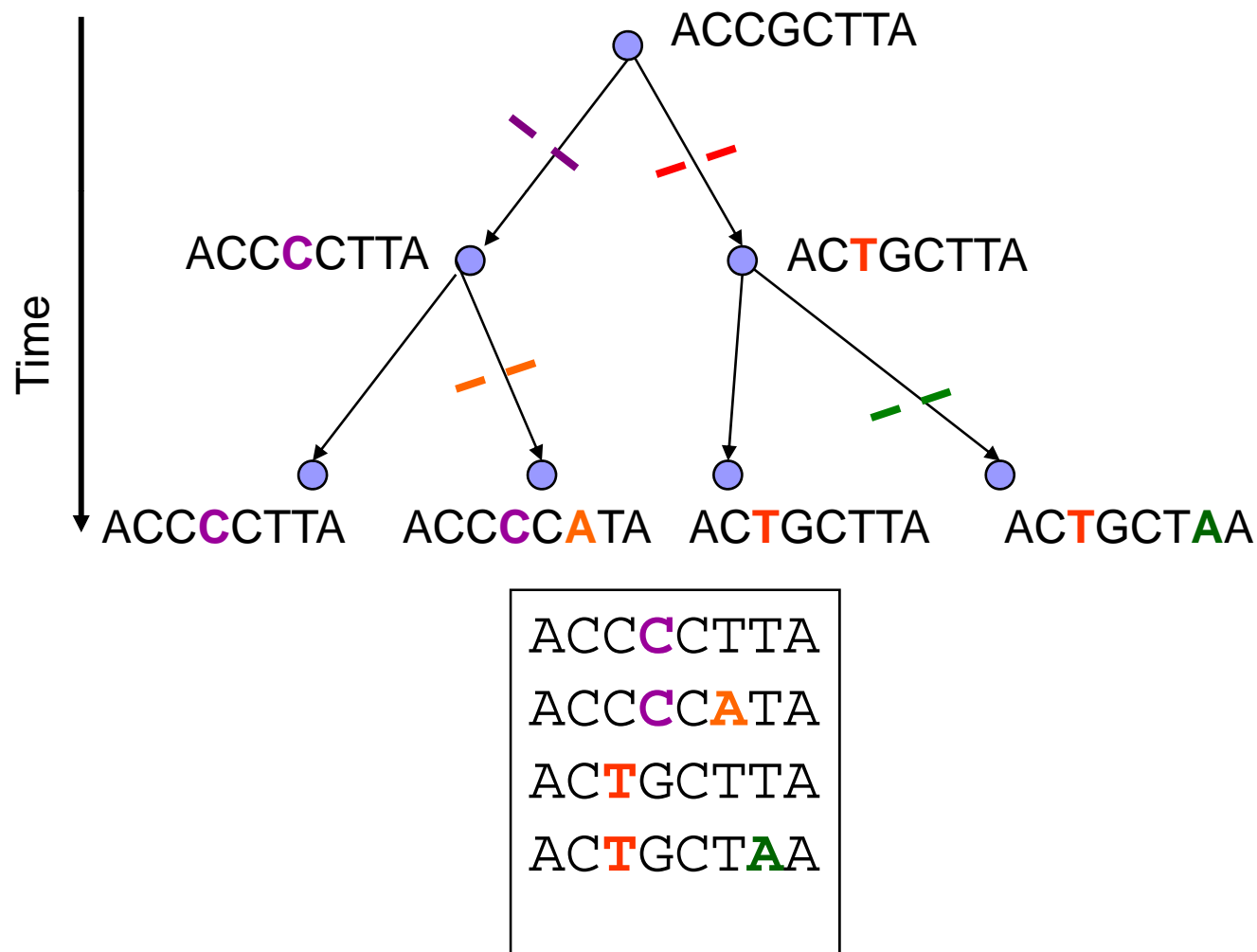




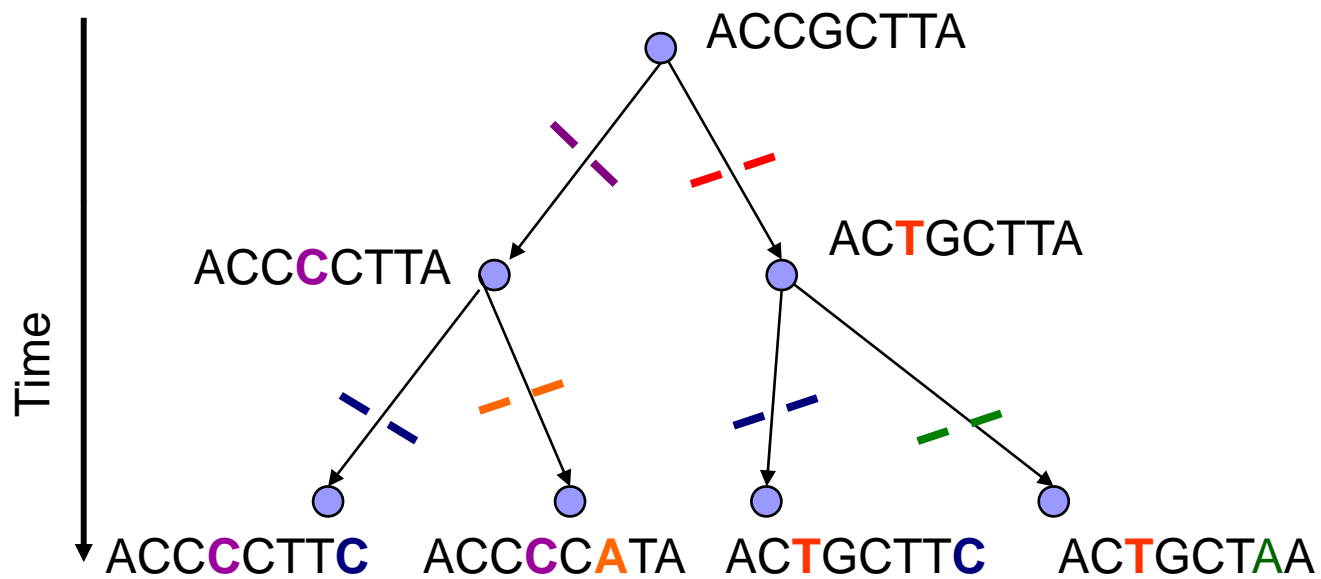
Parsimony analyses of sequence data

Pete Lockhart
Barbara Holland

When patterns can be mapped onto a tree



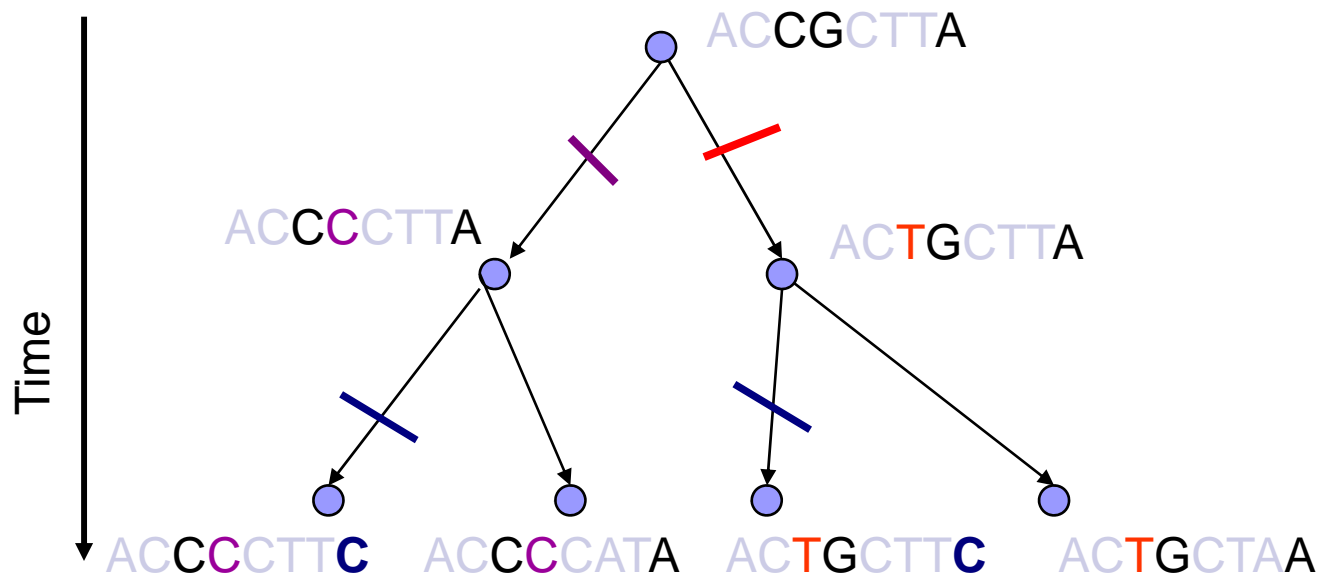
when not



ACCCCTTC
ACCCATA
ACTGCTTC
ACTGCTAA

Homoplasy

- When we have two or more characters that can't possibly fit on the same tree without requiring one character to undergo a parallel change or reversal it is called **homoplasy**.





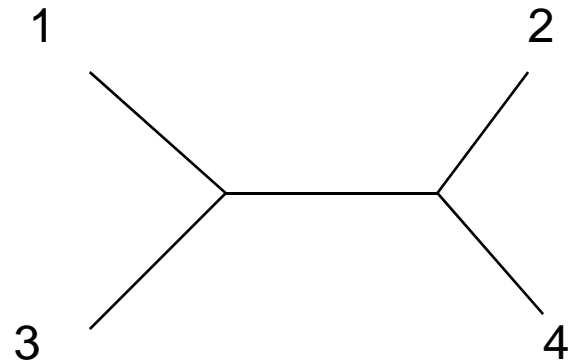
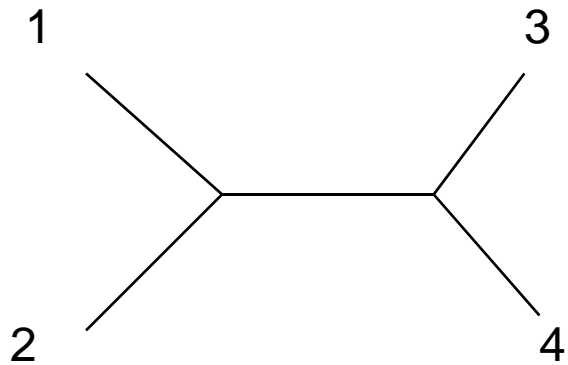
How can we choose the best tree?

- To decide which tree is best we can use an optimality criterion.
- Parsimony is one such criterion.
- It chooses the tree which requires the fewest mutations to explain the data.
- The **Principle of Parsimony** is the general scientific principle that accepts the simplest of two explanations as preferable.



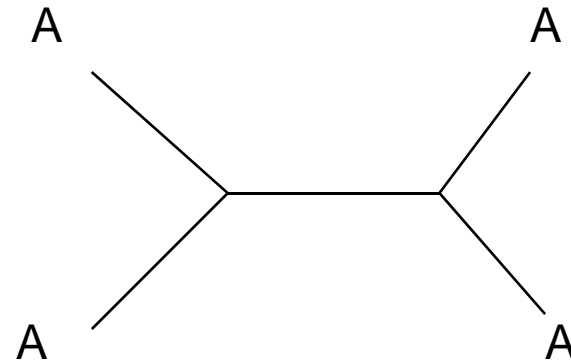
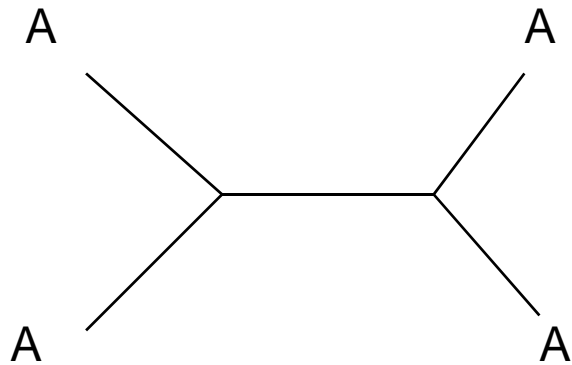
S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A

(1, 2), (3, 4)
(1, 3), (2, 4)



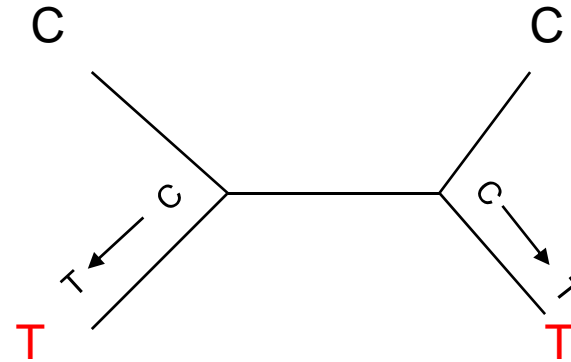
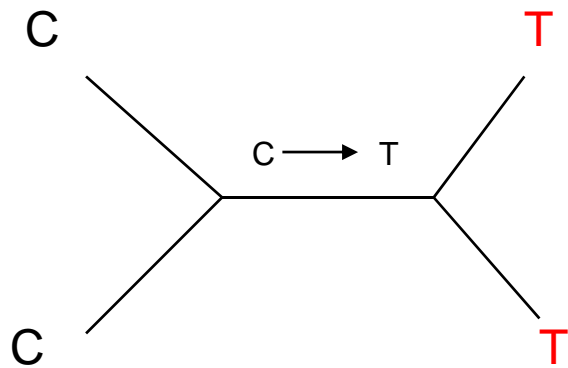


S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A
(1, 2), (3, 4)	0
(1, 3), (2, 4)	0



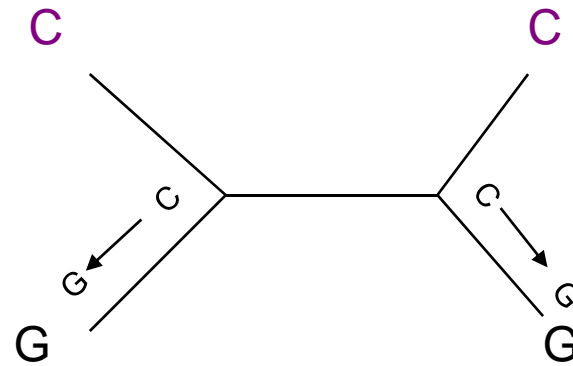
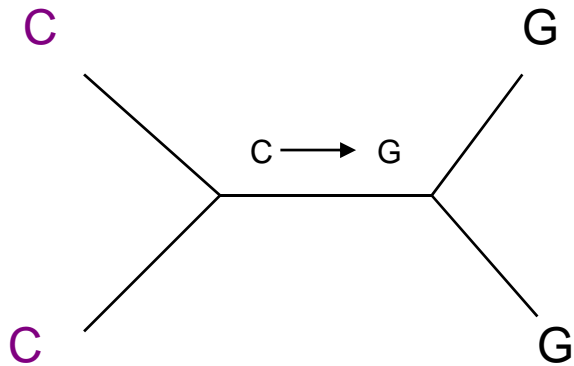


S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A
(1, 2), (3, 4)	001
(1, 3), (2, 4)	002



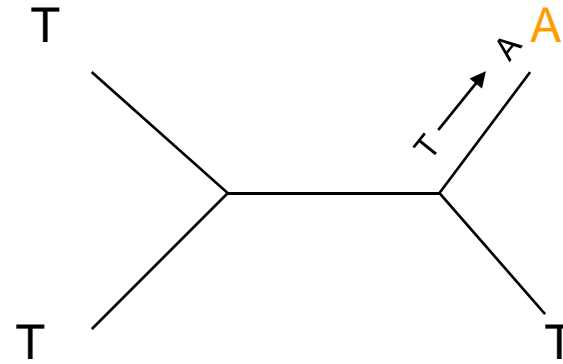
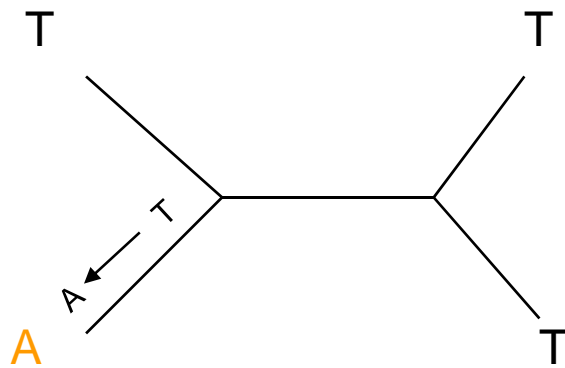


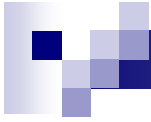
S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A
(1, 2), (3, 4)	0011
(1, 3), (2, 4)	0022



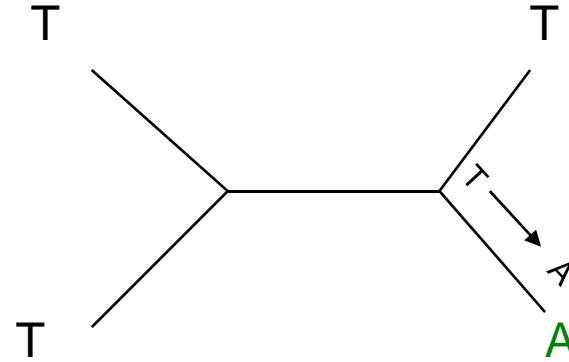
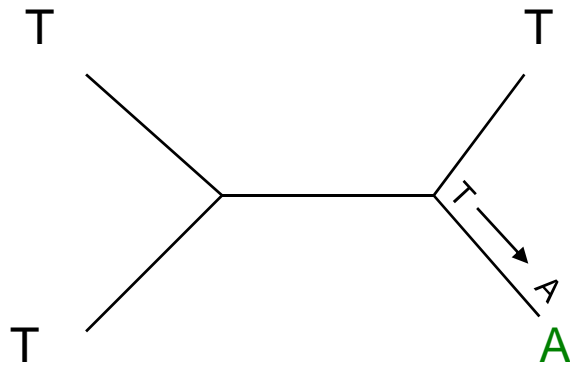


S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A
(1, 2), (3, 4)	001101
(1, 3), (2, 4)	002201

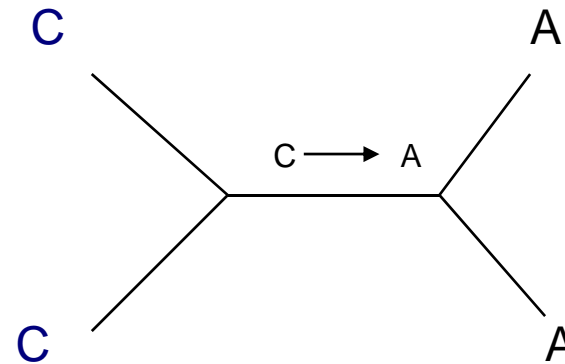
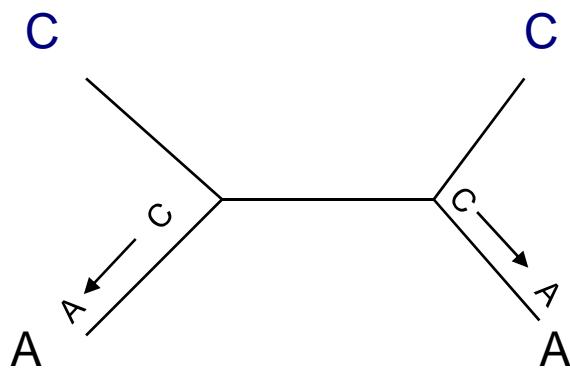




S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A
(1, 2), (3, 4)	0011011
(1, 3), (2, 4)	0022011



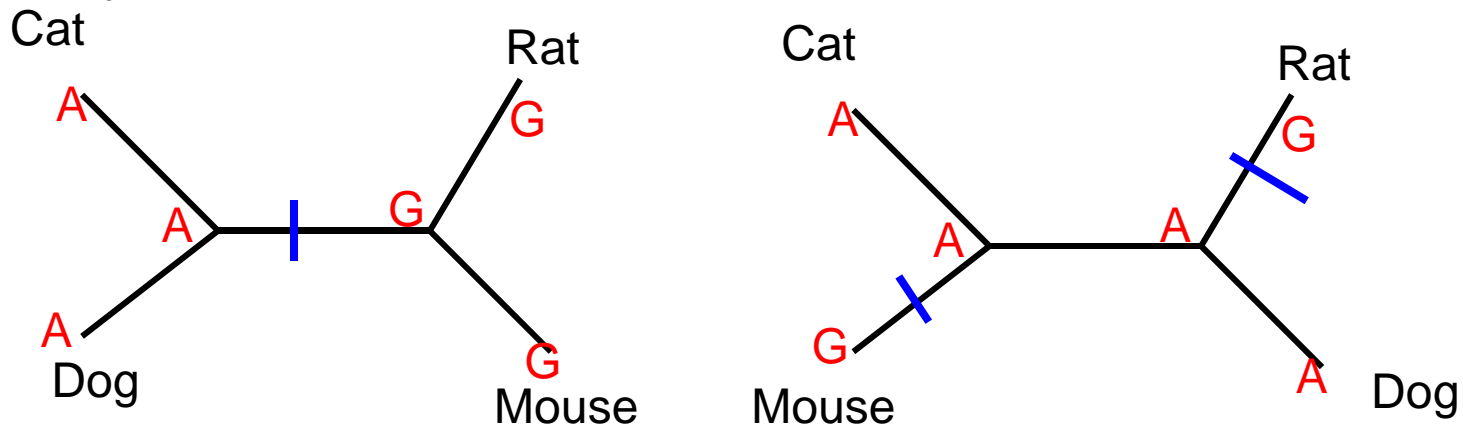
S1	ACC C CTTC
S2	ACC C A T A
S3	AC T GCTTC
S4	AC T GCT A A
(1, 2), (3, 4)	00110112 6
(1, 3), (2, 4)	00220111 7



According to the parsimony optimality criterion we should prefer the tree (1,2),(3,4) over the tree (1,3),(2,4) as it requires the fewest mutations.

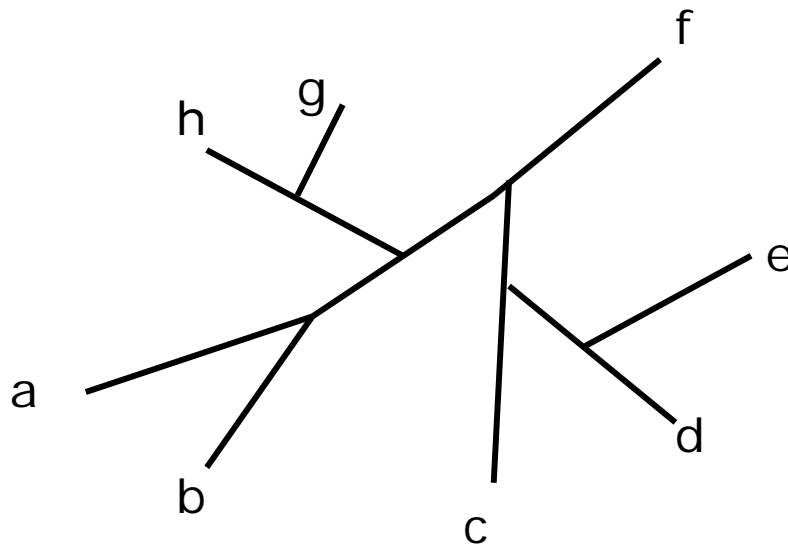
Maximum Parsimony

- The parsimony criterion tries to **minimise the number of mutations required to explain the data**
- The “Small Parsimony Problem” is to compute the number of mutations required on a given tree.
- For small examples it is straightforward to see how many mutations are needed



The Fitch algorithm

- For larger examples we need an algorithm to solve the small parsimony problem

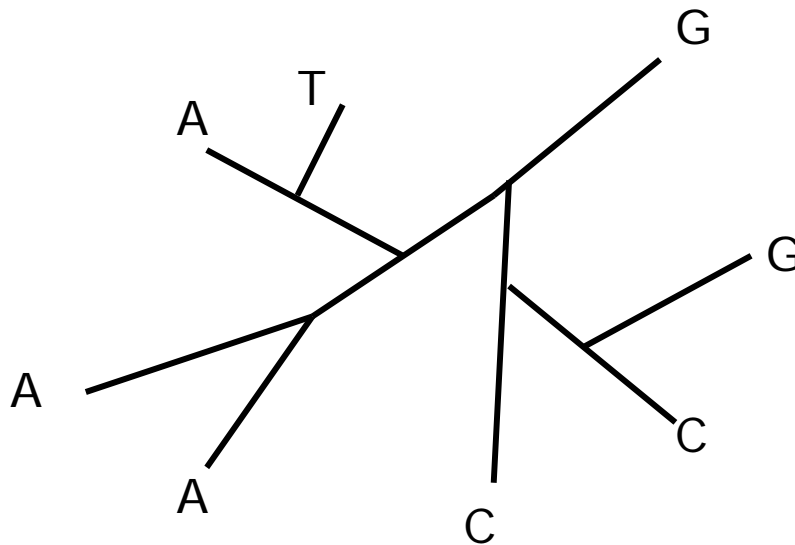


Site

a	A
b	A
c	C
d	C
e	G
f	G
g	T
h	A

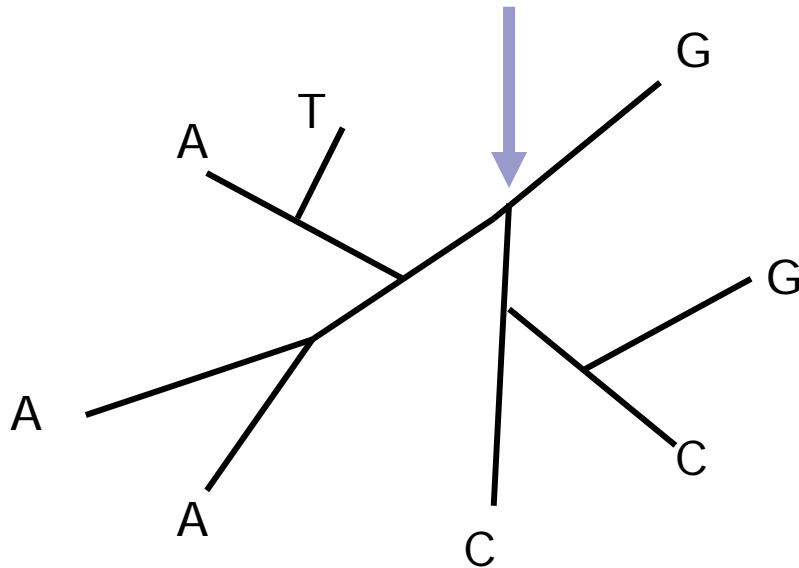
The Fitch algorithm

- Label the tips of the tree with the observed sequence at the site



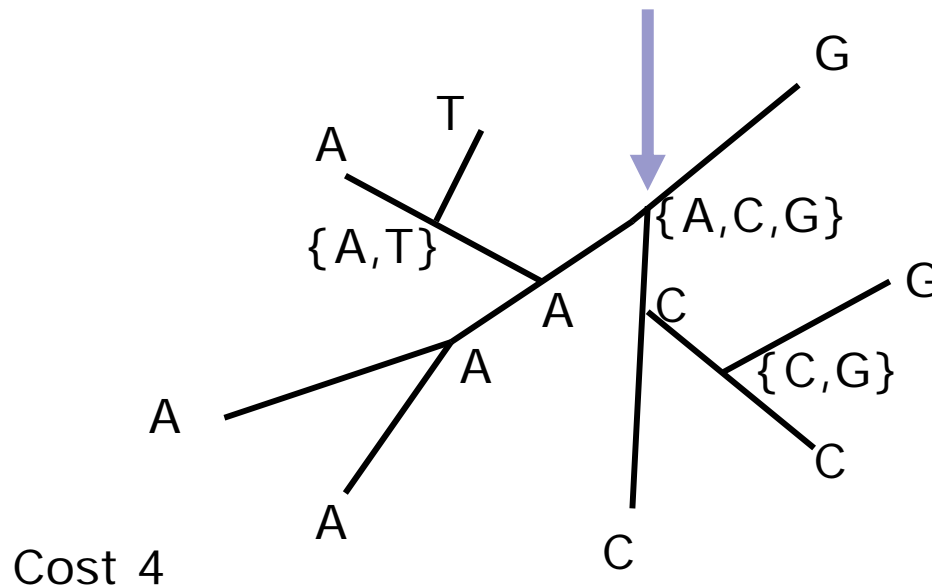
The Fitch algorithm

- Pick an arbitrary root to work towards



The Fitch algorithm

- Work from the tips of the tree towards the root. Label each node with the intersection of the states of its child nodes.
- If the intersection is empty label the node with the union and add one to the cost





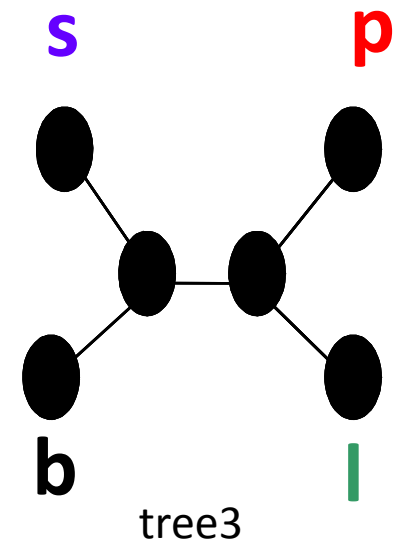
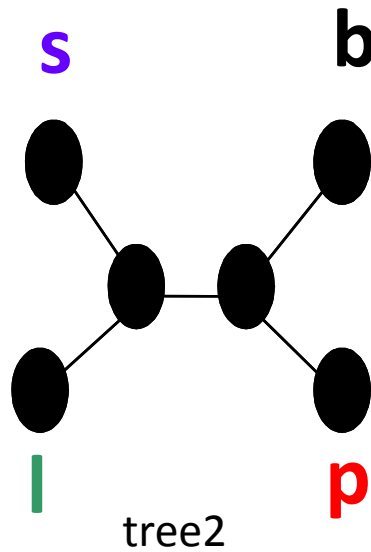
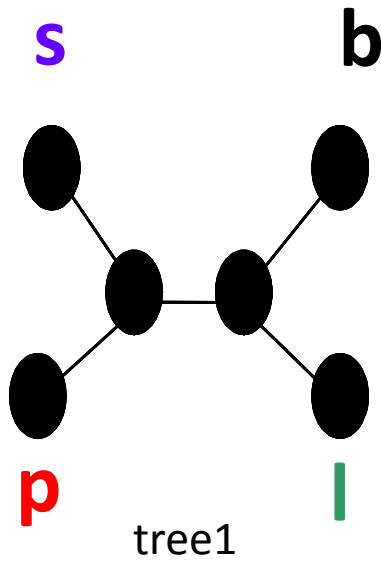
Fitch continued...

- The Fitch algorithm also has a second phase that allocates states to the internal nodes but it does not affect the cost.
- To find the Fitch cost of an alignment for a particular tree we just sum the Fitch costs of all the sites.



slime mould	ATAAA
politician	ATAGC
lawyer	GAAAC
best friend	GATGA

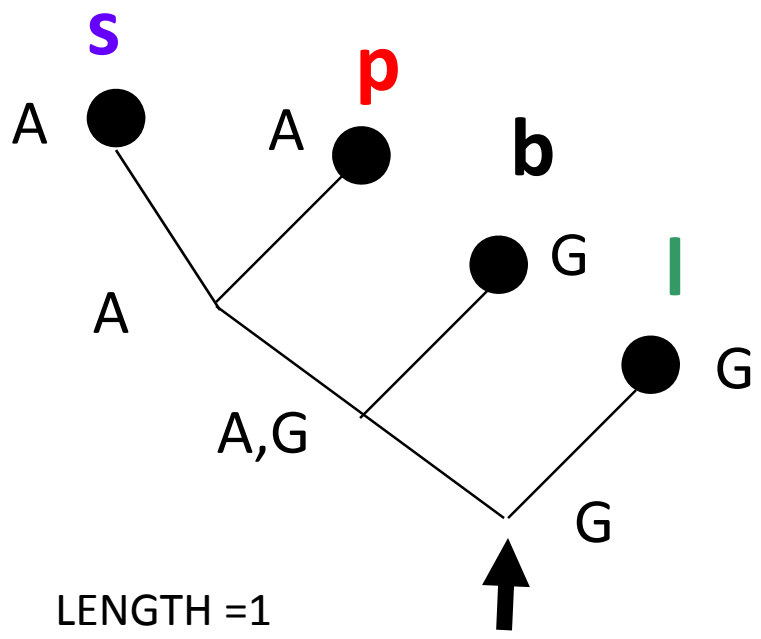
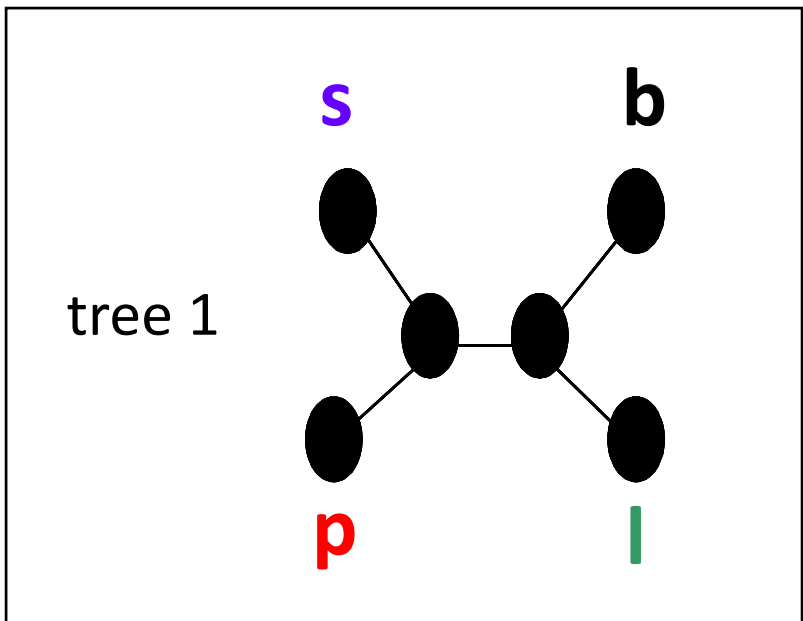
Which bifurcating tree best explains the data?



column 1 (site pattern 1)

slime mould	A
politician	A
lawyer	G
best friend	G

tree 1 arbitrarily
rooted on branch
leading to lawyer



column 2 (site pattern 2)

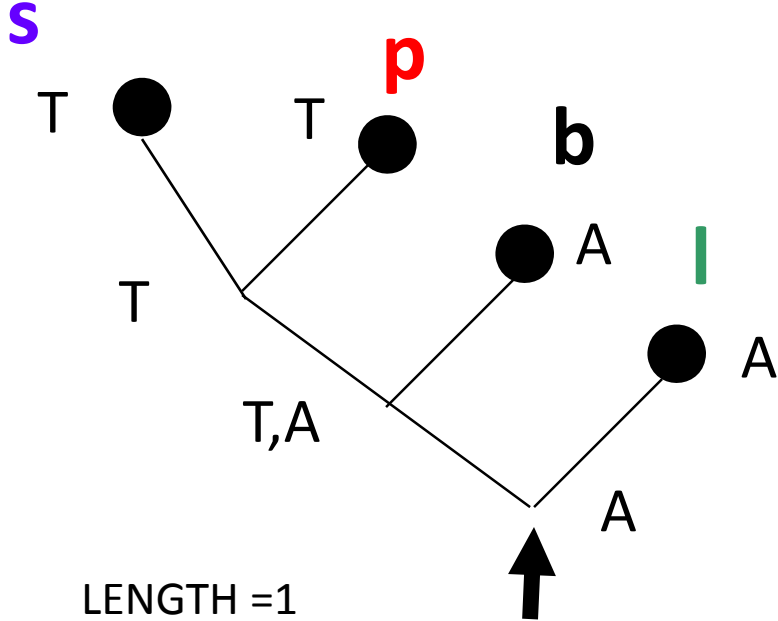
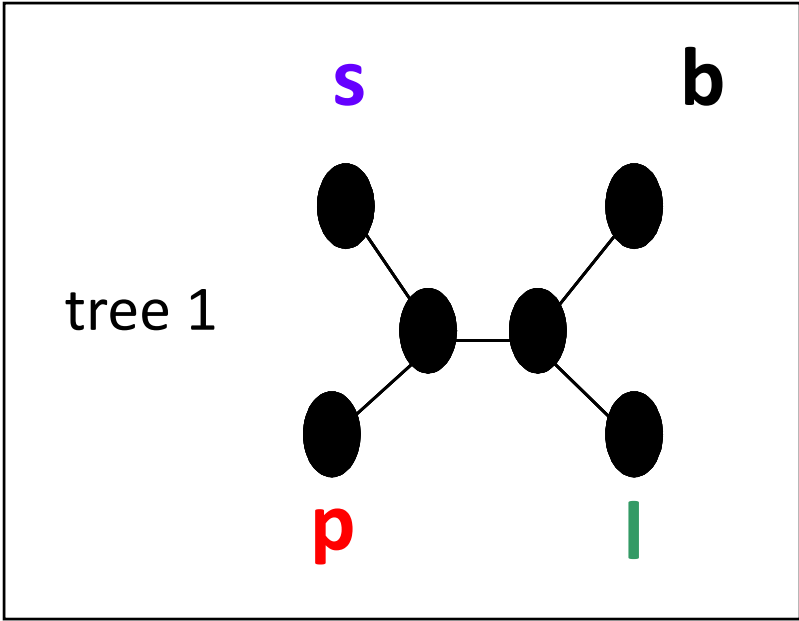
slime mould T

politician T

lawyer A

best friend A

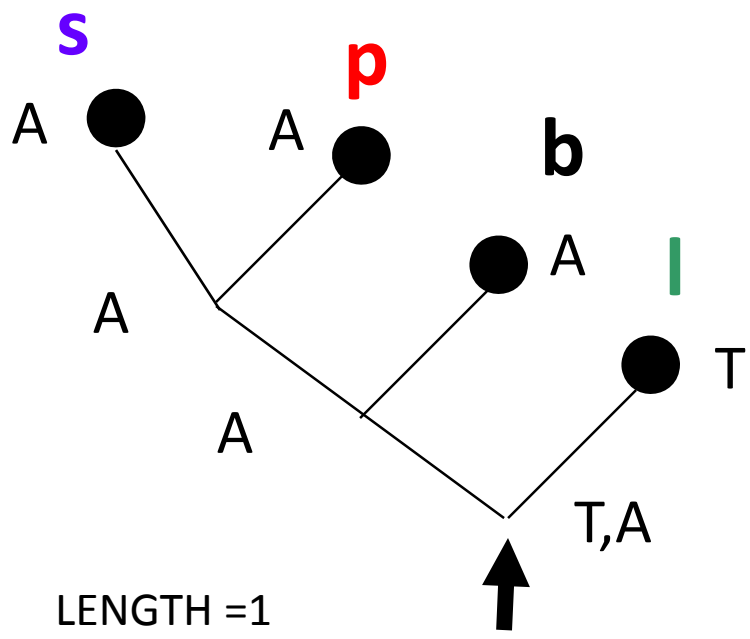
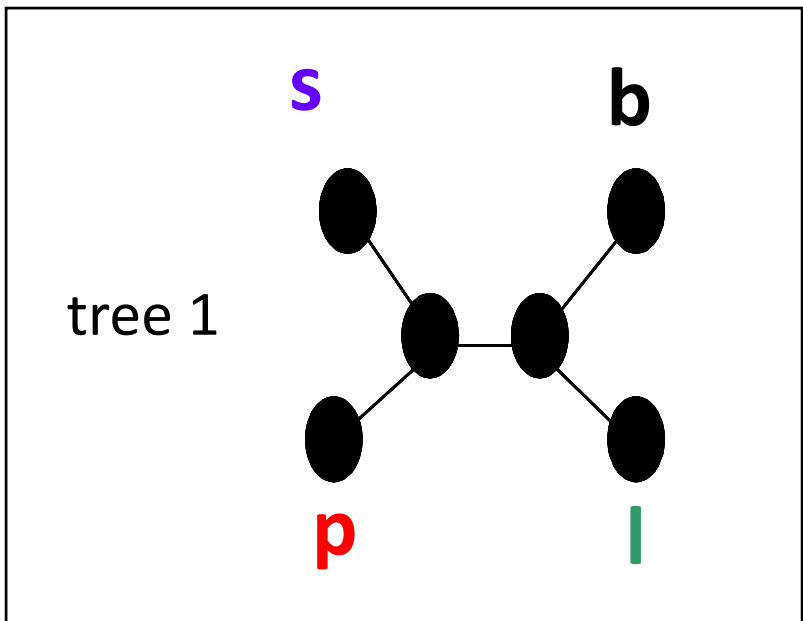
tree 1 arbitrarily
rooted on branch
leading to lawyer



column 3 (site pattern 3)

slime mould	A
politician	A
lawyer	A
best friend	T

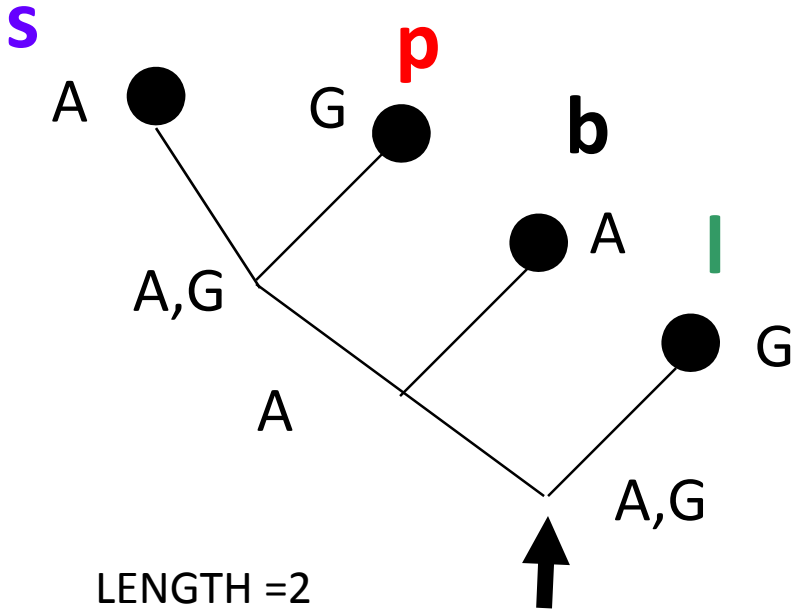
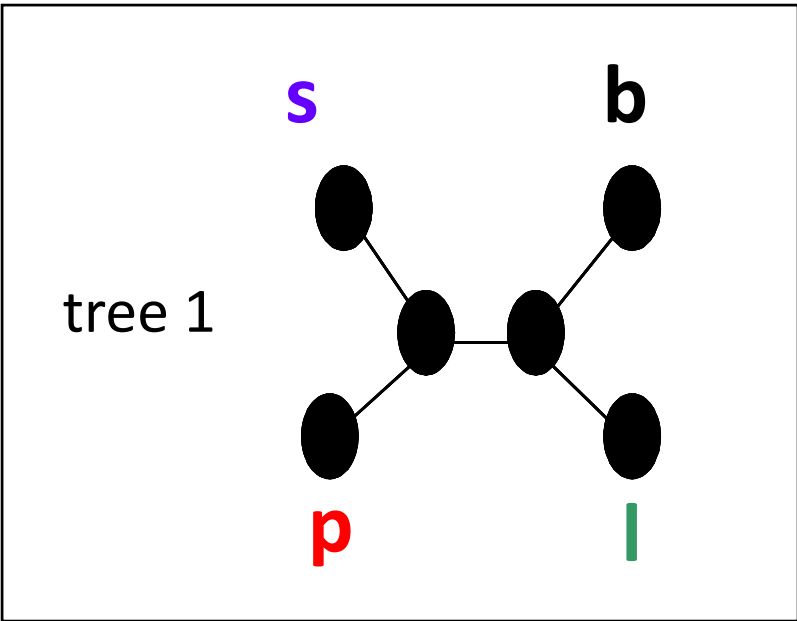
tree 1 arbitrarily
rooted on branch
leading to lawyer



column 4 (site pattern 4)

slime mould	A
politician	G
lawyer	A
best friend	G

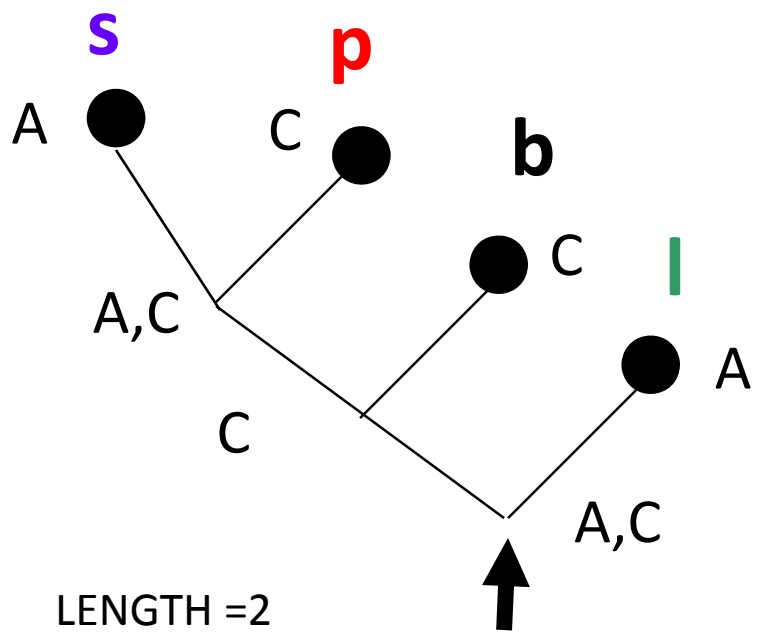
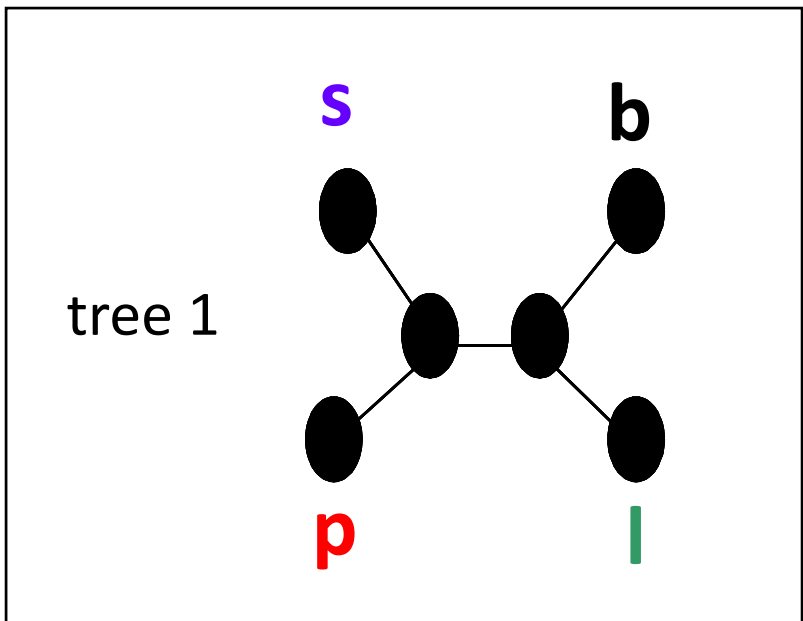
tree 1 arbitrarily rooted on branch leading to lawyer



column 5 (site pattern 5)

slime mould	A
politician	C
lawyer	C
best friend	A

tree 1 arbitrarily rooted on branch leading to lawyer





The most parsimonious tree

- length of data on tree 1 = $1+1+1+2+2=7$
- length of data on tree 2 = $2+2+1+1+2=8$
- length of data on tree 3 = $2+2+1+2+1=8$

- The tree with shortest length = the most parsimonious tree = tree 1 (length 7)



Tricks to increase efficiency

- Ignore constant sites
 - These must cost 0 on any tree
- Ignore parsimony uninformative sites
 - Unless there are at least two states that occur at least twice the site will have the same score on any tree.
- Groups sites with the same pattern
 - E.g. the site ACCGTA will have the same score as the site CTTAGC as will any site with the pattern $wxyzw$



Sankoff algorithm

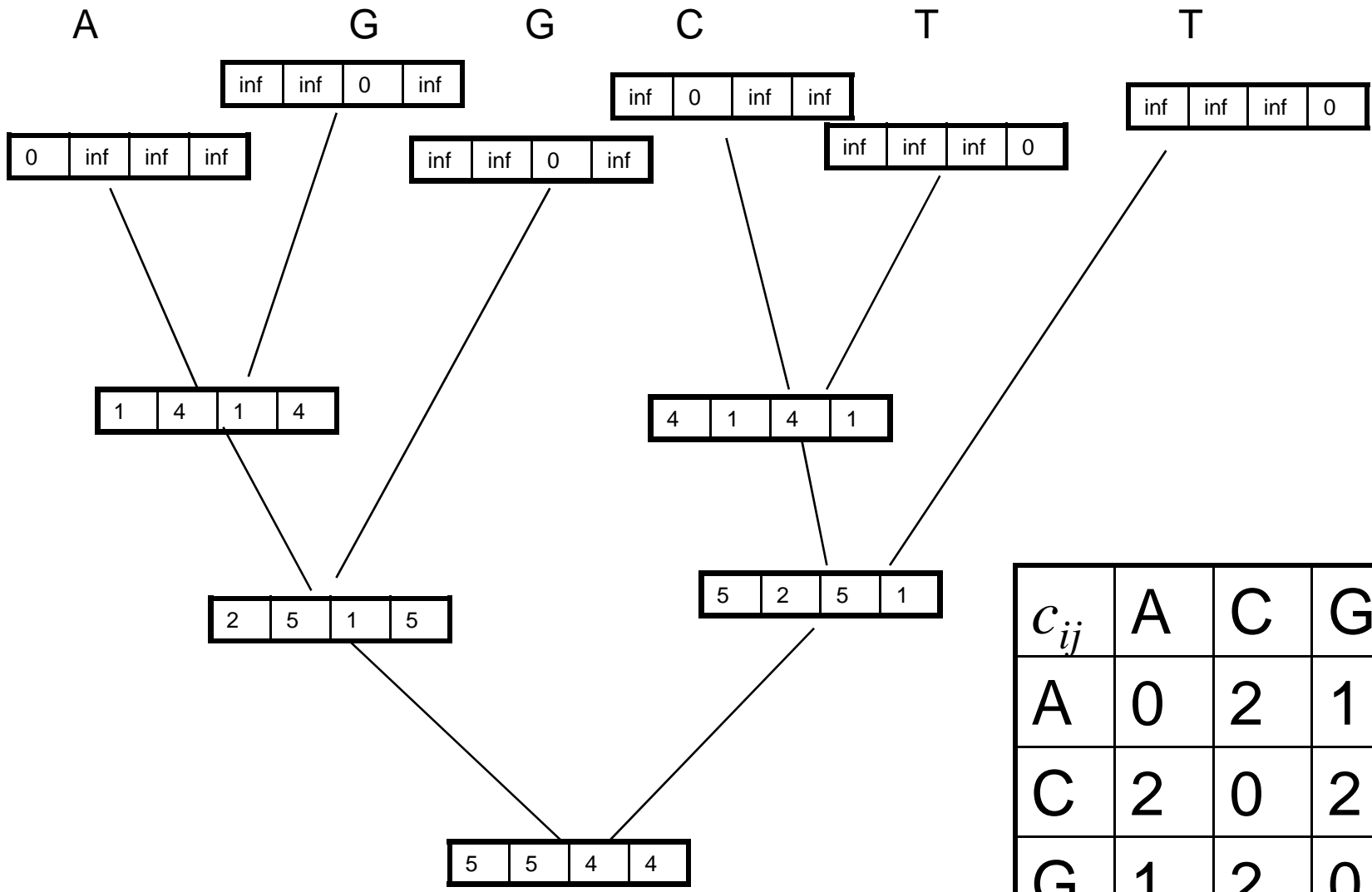
- More general than the Fitch algorithm.
- Assumes we have a table of costs c_{ij} for all possible changes between states i and j
- For each node k in the tree we compute $S_k(i)$ the minimal cost given that node k is assigned state i .
- In particular we can compute the minimum value over i for $S_{root}(i)$ which will be the total cost of the character on the tree.



Sankoff algorithm continued

- The value of $S(i)$ is easy to compute for the tips of the tree, we assign cost 0 if the observed state is i and otherwise assign cost infinity.
- A dynamic programming approach is used to compute the score for an ancestral node a given that the scores of its descendent nodes l and r (left and right) are known.

$$S_a(i) = \min_j [c_{ij} + S_l(j)] + \min_k [c_{ik} + S_r(k)]$$



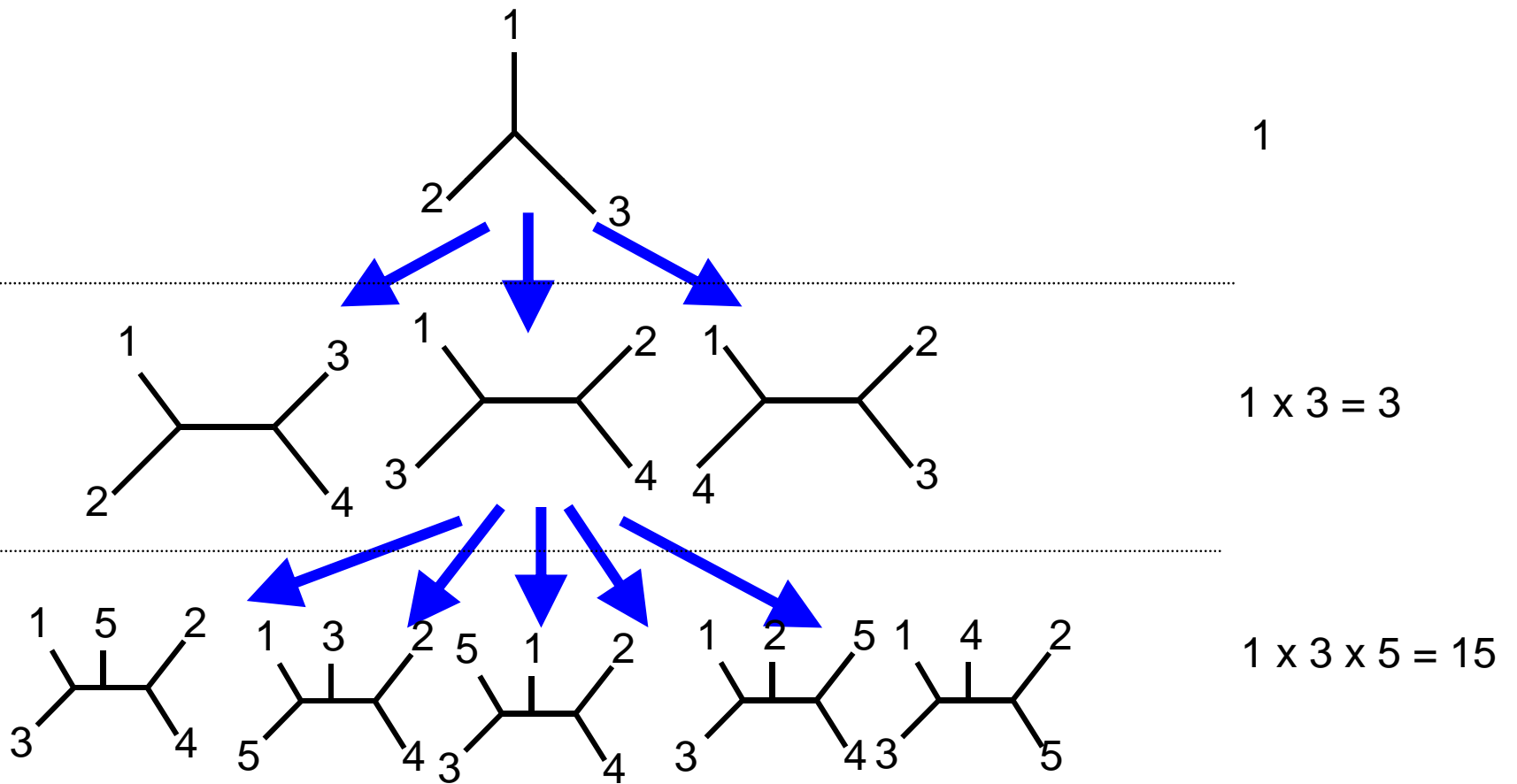
c_{ij}	A	C	G	T
A	0	2	1	2
C	2	0	2	1
G	1	2	0	2
T	2	1	2	0



The “large parsimony problem”

- The small parsimony problem – to find the score of a given tree - can be solved in linear time in the size of the tree.
- The large parsimony problem is to find the tree with minimum score.
- It is known to be NP-Hard.

Counting trees



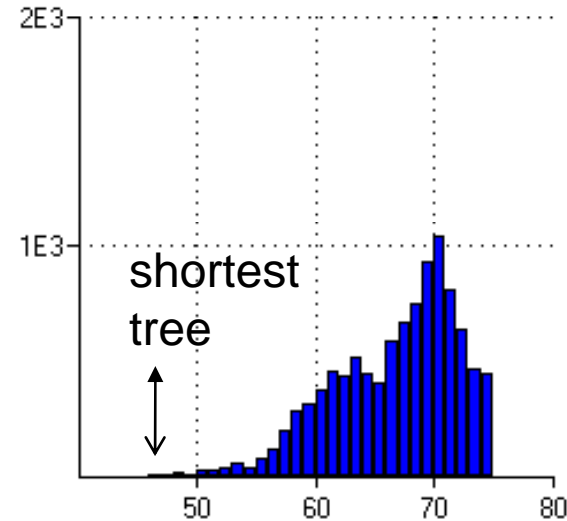
How many trees are there?

#species	#unrooted binary tip-labelled trees
4	3
5	$3*5=15$
6	$3*5*7=105$
7	$3*5*7*9=945$
10	2,027,025
20	$2.2*10^{20}$
n	$(2n-5)!!$

An exact search for the best tree, where each tree is evaluated according to some optimality criterion such as parsimony quickly becomes intractable as the number of species increases

Search strategies

- Exact search
 - possible for small n only
- Branch and Bound
 - up to ~20 taxa
- Local Search - Heuristics
 - pick a good starting tree and use moves within a “neighbourhood” to find a better tree.
- Meta-heuristics
 - Genetic algorithms
 - Simulated annealing
 - The ratchet

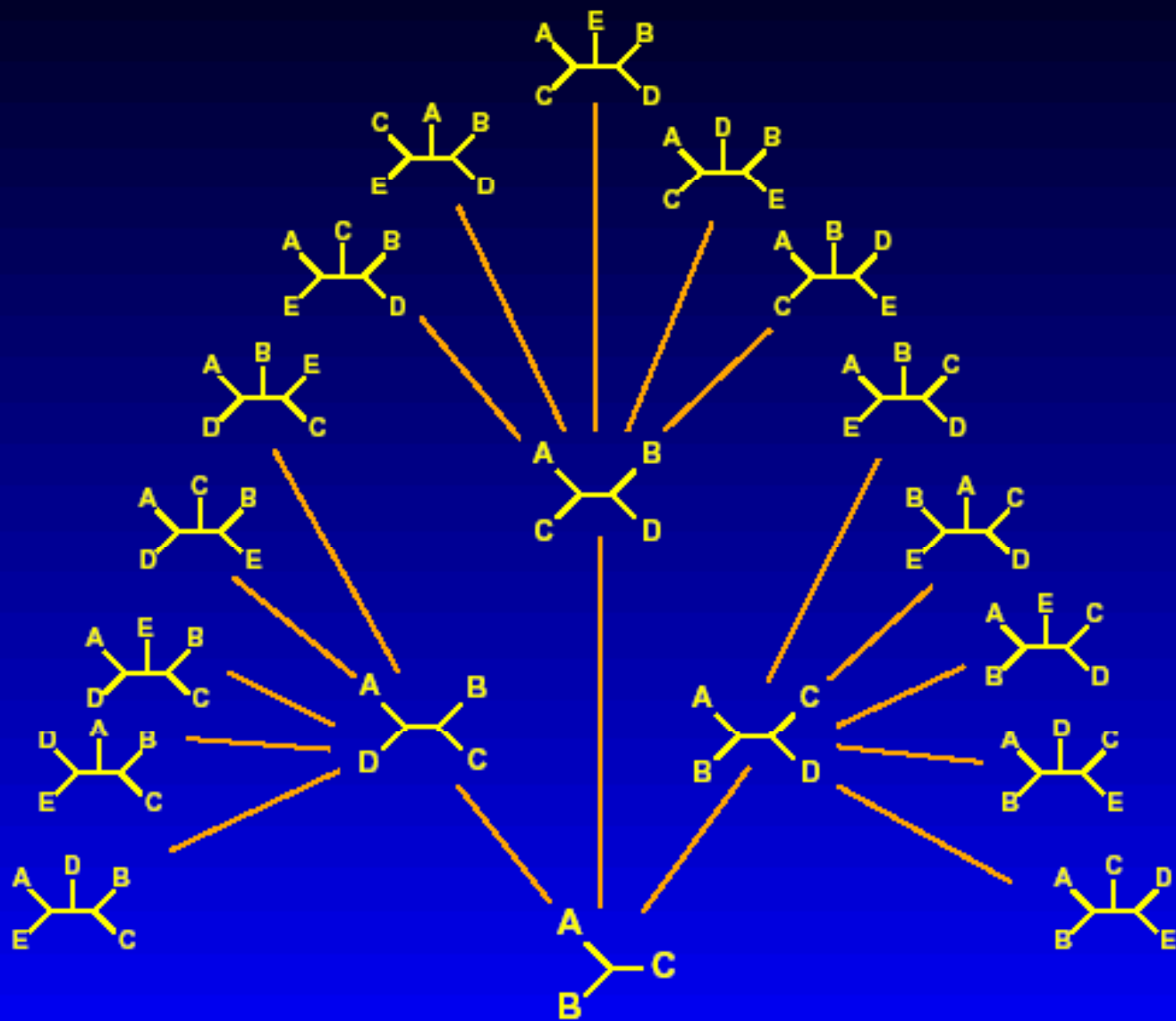




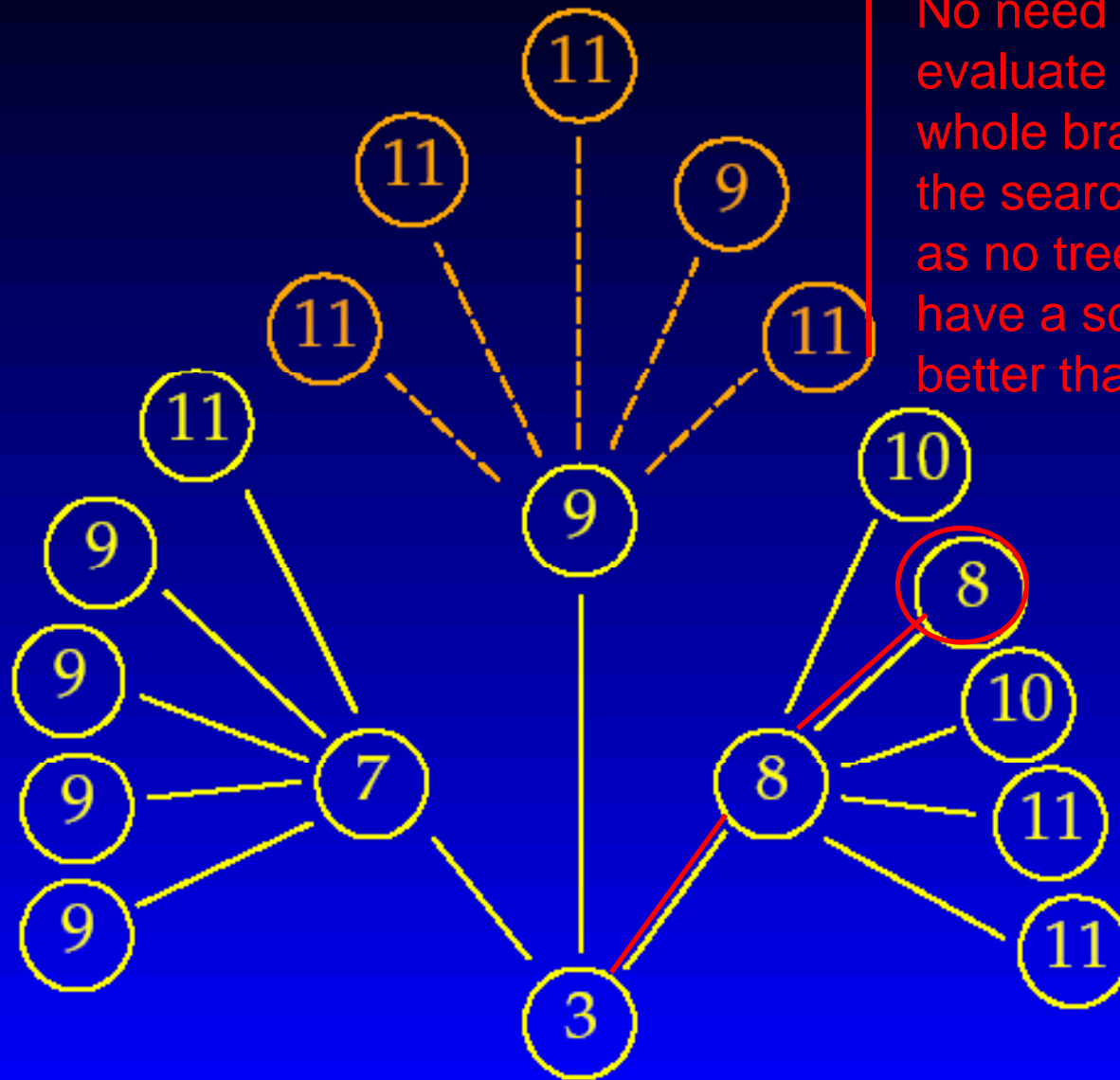
Exact searches

- for small number of taxa ($n \leq 12$) it is possible to compute the score of every tree
- Branch and Bound searches also guarantee to find the optimal solution but use some clever rules to avoid having to check all trees. They may be effective for up to 25 taxa.

Search tree of trees



same, with parsimony scores in place of trees



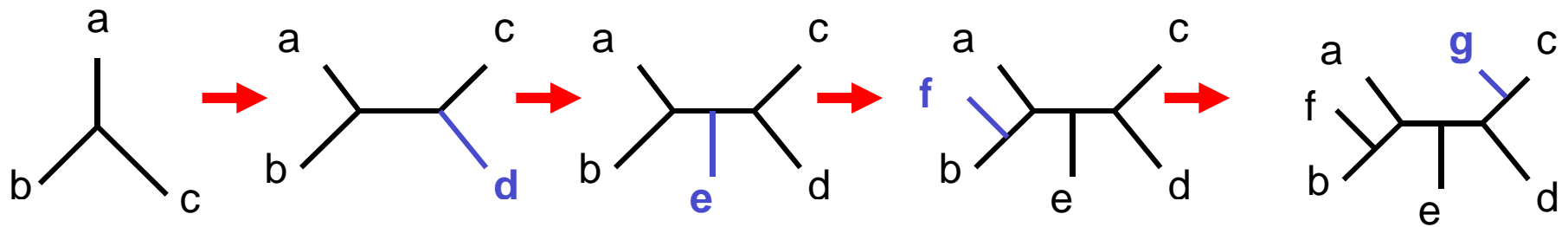
No need to evaluate this whole branch of the search tree, as no tree can have a score better than 9



Searching tree space

- For more than 25 taxa it is usually necessary to use *heuristic* methods.
- When a problem is too hard to solve exactly we use methods that will give a good solution even if we cannot guarantee it will be the best solution.
 - Start from a good (or perhaps random) subtree
 - Add taxa in a stepwise fashion, placing new taxa on the branch that gives the best score

Stepwise addition



Place the new taxon on the branch that gives a subtree with the best score

The problem of local optima and finding the best tree

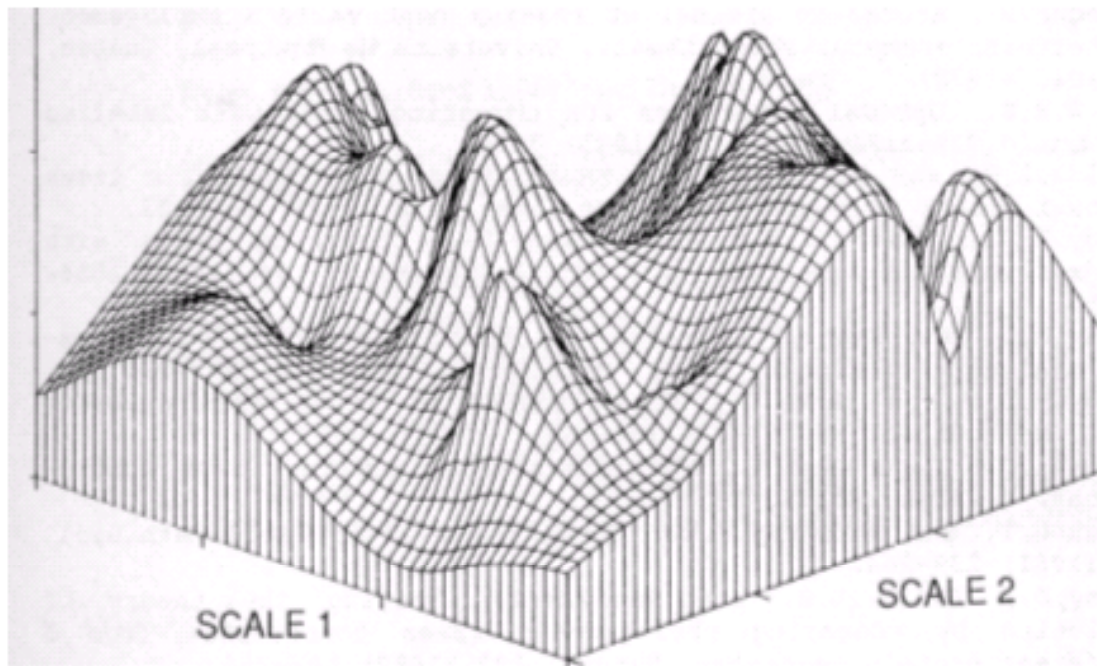


Figure 4.12. The problem of local optima. This figure is derived from the similarity of trees that are close to optimal. In some parts of the space of trees it is possible to ‘hill-climb’ straight to the **global optimum**. In other regions of tree space, hill-climbing only leads to a **local optimum**.

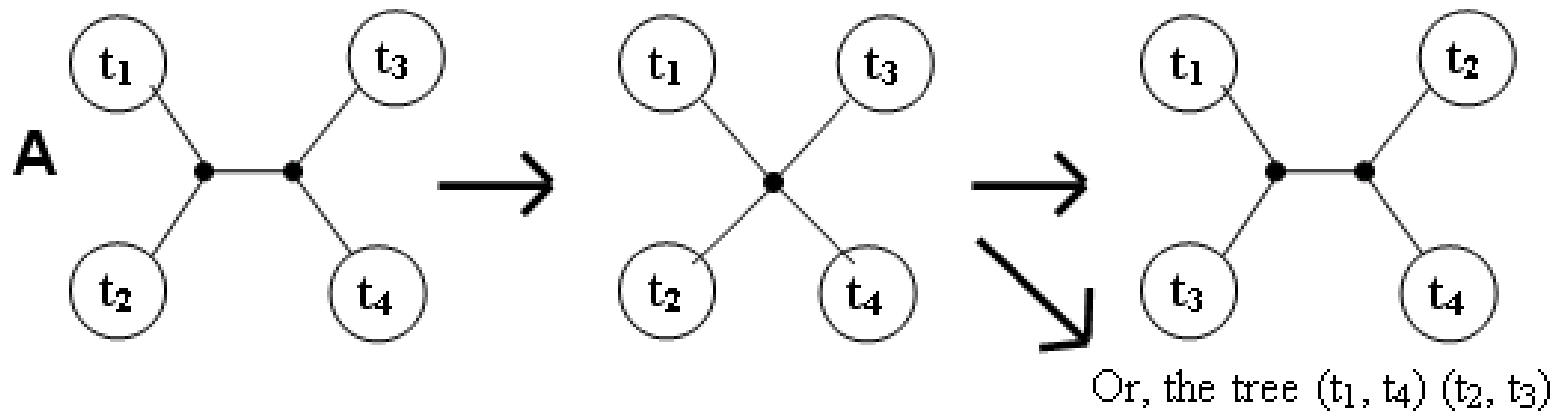


To avoid local optima

- Start from a good (or perhaps random) tree
- Alter the tree using some “move” and check if the new tree has a better score
- Keep moving to trees with better scores until no improving move can be found

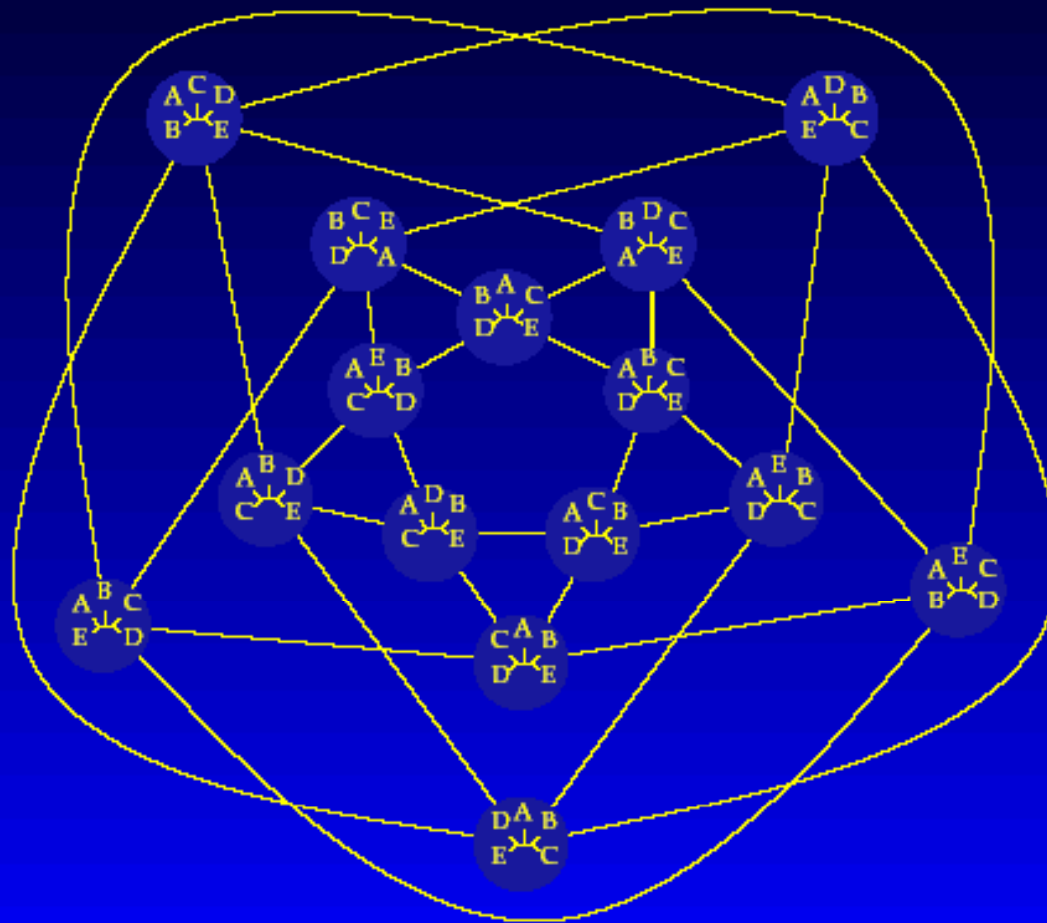
Moving between trees

Crossover – nearest neighbor interchange (nni)



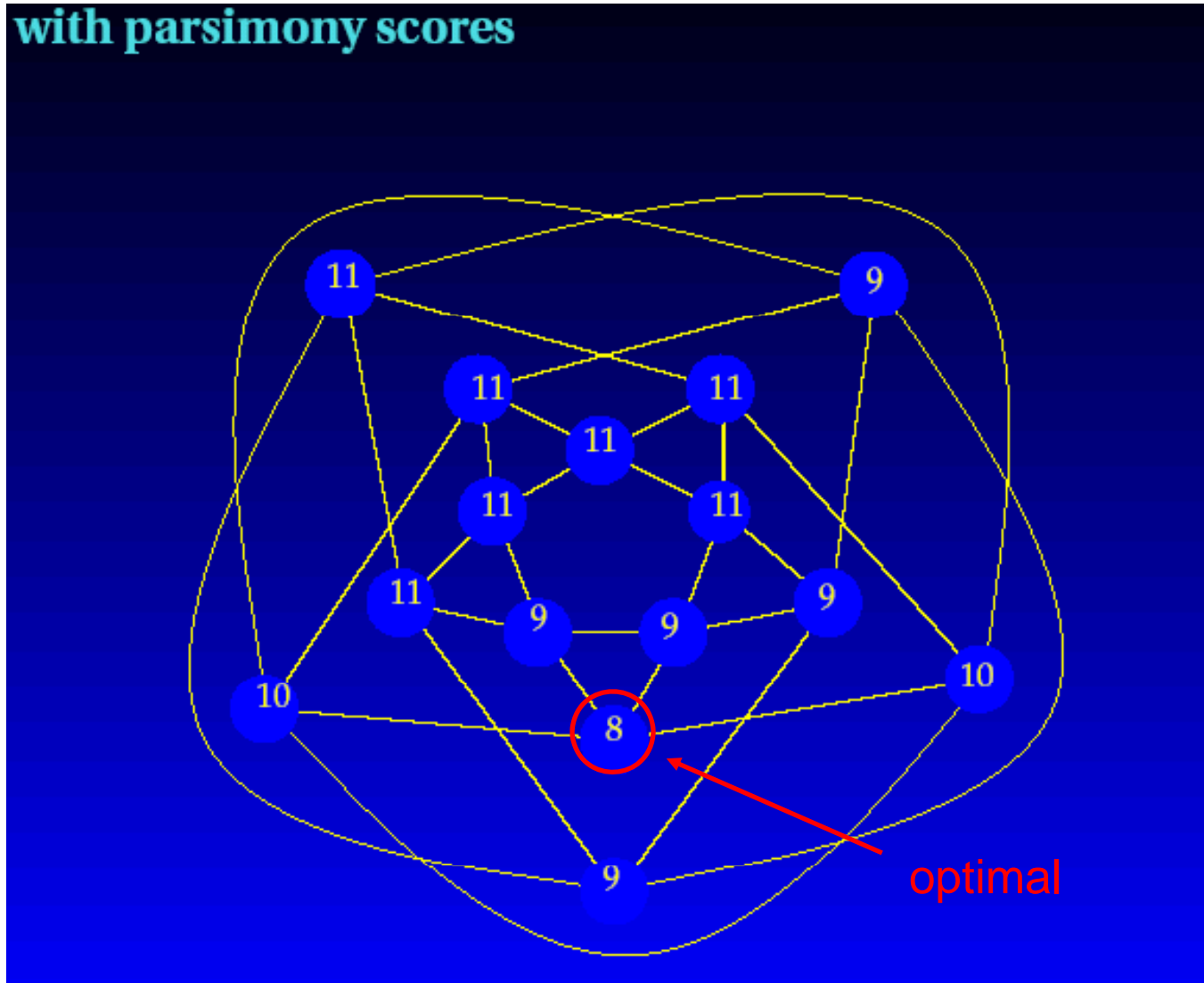
Moving around in tree-space

all 15 trees, connected by NNIs

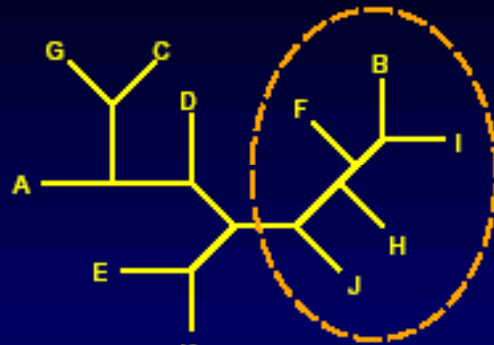


Moving around in tree-space

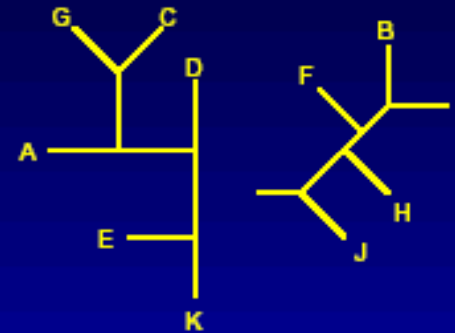
with parsimony scores



Subtree pruning and regrafting (SPR) rearrangement



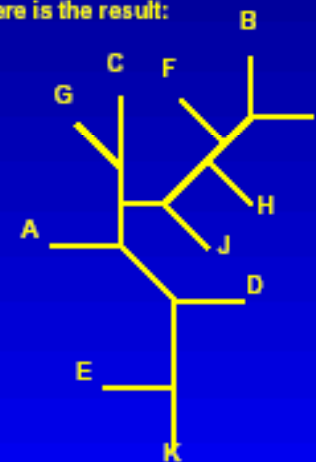
Break a branch, remove a subtree



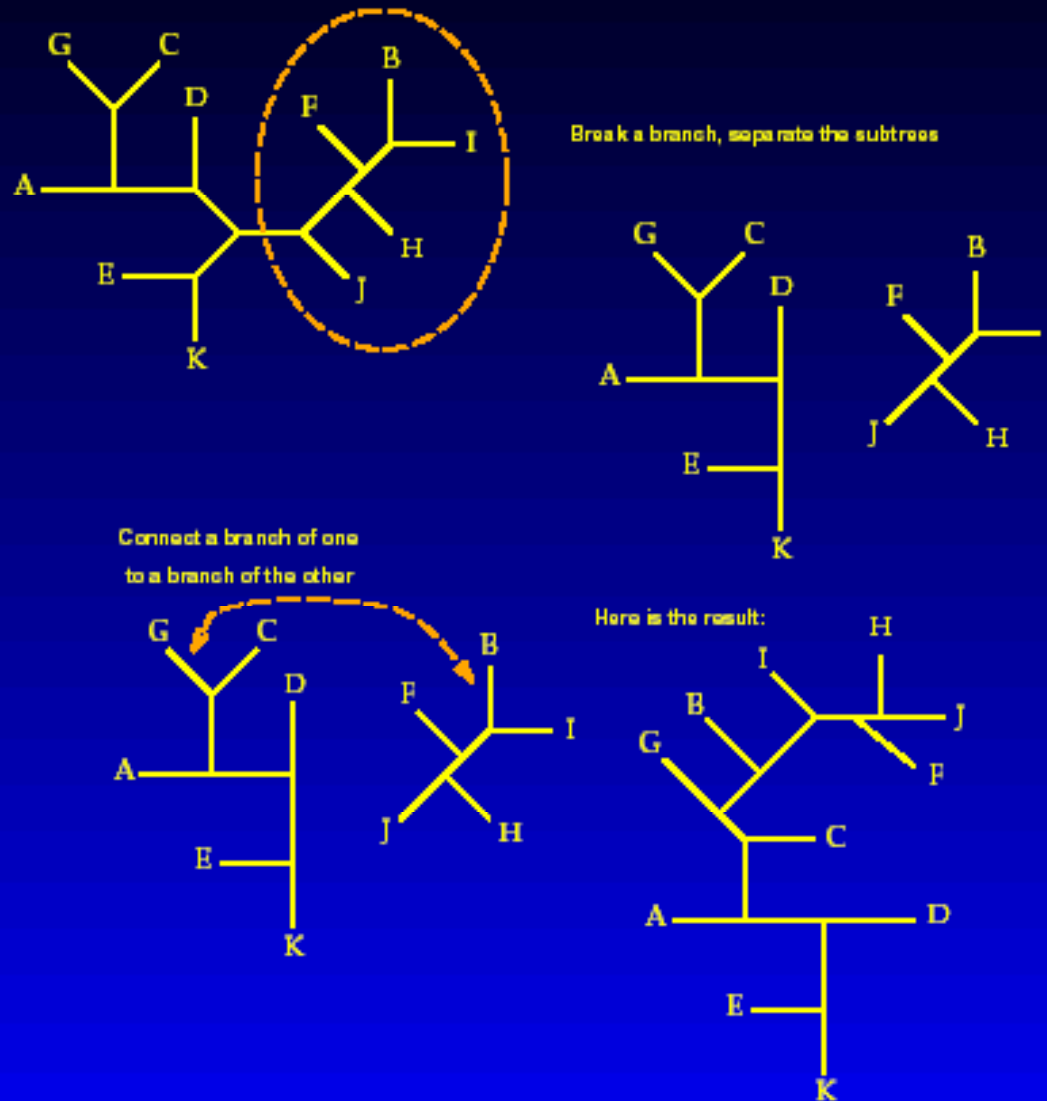
Add it in, attaching it to one (*) of the other branches

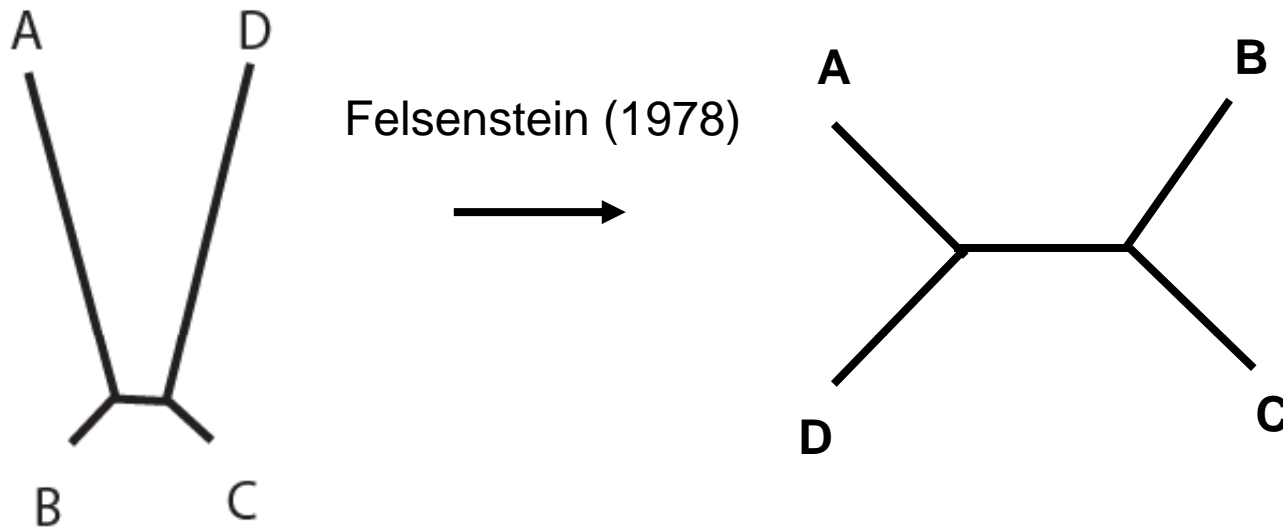


Here is the result:

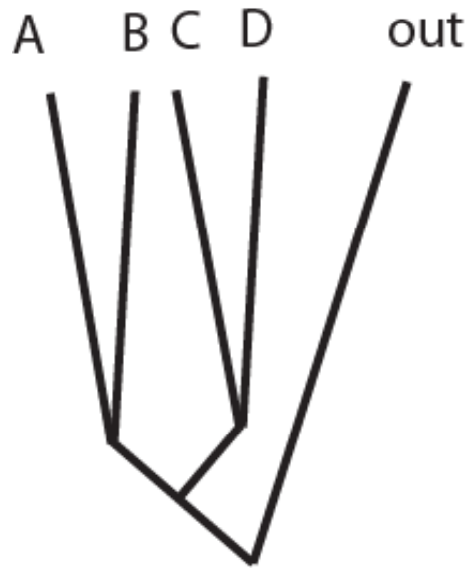


Tree bisection and reconnection (TBR) rearrangement

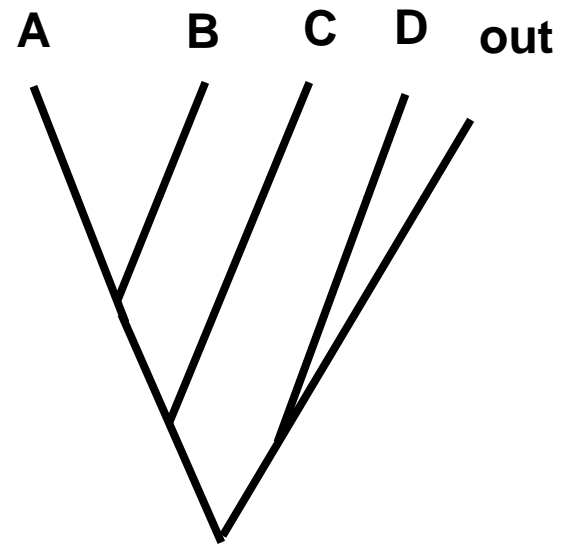




Species A	AGCTAGCTAGCTAGCT
Species B	AAAAAAAAAAAAAAAAAA
Species C	AAAAAAAAAAAAAAAAAA
Species D	AAAAGGGGCCCTTTT



Hendy and
Penny (1989)





Review

- Real data sets usually contain homoplasy – characters that do not agree.
- To choose the best tree we can use an optimality criterion – this is a way of giving each possible tree a score.
- Parsimony is one example of an optimality criterion – it prefers the tree that requires the fewest mutations to explain the data.
- For datasets with >20 taxa we cannot check the score of every tree as there are too many.
- Instead we use heuristic searches to explore tree space and find locally optimal trees.
- Parsimony analysis has low reconstruction accuracy in some situations