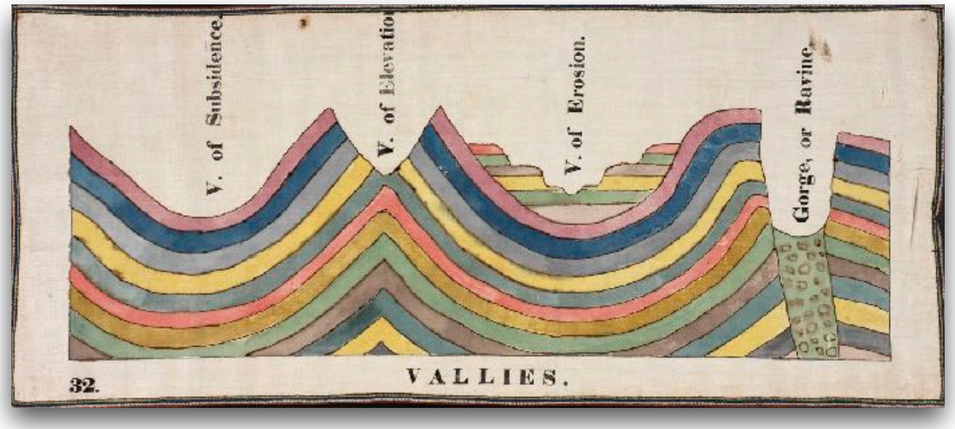
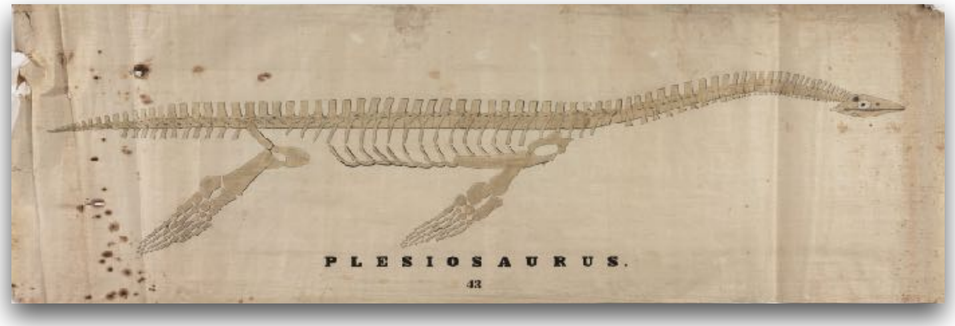




distance-based
versus
charcter-based
algorythms

Distance-based methods

Distance-based phylogenetic trees are based on the **total number of evolutionary changes** between pairs of sequences.

Starting from the alignment, these methods look at all possible pairs of the aligned sequences and count how many characters are different at each position. These pairwise differences are represented in a **distance matrix**.

These methods are now best used for **exploratory analysis** of large datasets before conducting more intensive tree building using character-based methods.

Two distance-based methods are commonly used:

- **Unweighted Pair Group Method with Arithmetic Mean (UPGMA):** provides rooted and ultrametric trees. It assumes a constant molecular clock, meaning sequences evolve at a uniform rate over time.

- **Neighbor joining (NJ):** it does not assume a molecular clock, meaning it allows different lineages to evolve at different rates, and produce unrooted trees.

UPGMA

Step 1: create a distance matrix

The first step is to calculate a pairwise distance matrix between sequences using a metric like p-distance, Jukes-Cantor or Kimura …

… more on these two in the following lessons 👀

Taxon	A	B	C	D
A	0	5	9	9
B	5	0	10	10
C	9	10	0	8
D	9	10	8	0

Step 2: identify the closest pair

Find the smallest distance in the matrix. In this case, A & B have the shortest distance (5), so they are clustered first.

Taxon	A	B	C	D
A	0	5	9	9
B	5	0	10	10
C	9	10	0	8
D	9	10	8	0

Step 3: compute a new distance matrix

The new cluster (A,B) must have its distance recalculated to all other taxa. The distance to each taxon is the average of the distances from A and B:

$$d\,(A,B) - C = d(A,C) + d(B,C) \,/\, 2$$

Taxon	(A,B)	C	D
(A,B)	0	9,5	9,5
C	9,5	0	8
D	9,5	8	0

Step 4: repeat the process of step 2 & 3

2. Find the next smallest distance and cluster pairs.
3. Recalculate new distances.

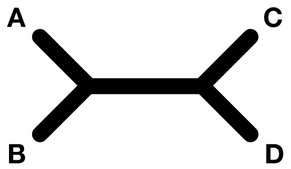
Taxon	(A,B)	(C,D)
(A,B)	0	9,5
(C,D)	9,5	0

Step 5: final merge

The last two clusters, (A,B) and (C,D), are joined,
and the tree is complete.

((A,B),(C,D))

Does in remind you of anything?



PS: there is also WPGMA where the W stands for *weighted*. Basically it weights distances by cluster size, leading to more balanced trees.

NJ

Step 1: create a distance matrix

The same as with UPGMA!

They are both distance based methods in the end! 😜

Taxon	A	B	C	D	E	F	G	H
A	0	4	8	8	9	10	10	10
B	4	0	8	8	9	10	10	10
C	8	8	0	2	5	6	7	7
D	8	8	2	0	5	6	7	7
E	9	9	5	5	0	3	4	4
F	10	10	6	6	3	0	2	2
G	10	10	7	7	4	2	0	1
H	10	10	7	7	4	2	1	0

Step 2: compute net divergence (Q-matrix)

Instead of selecting directly the closest pair (as in UPGMA), NJ calculates a **Q-matrix**, based on net divergence of each taxon from all others (**R**).

Taxon	A	B	C	D	E	F	G	H
A	0	4	8	8	9	10	10	10
B	4	0	8	8	9	10	10	10
C	8	8	0	2	5	6	7	7
D	8	8	2	0	5	6	7	7
E	9	9	5	5	0	3	4	4
F	10	10	6	6	3	0	2	2
G	10	10	7	7	4	2	0	1
H	10	10	7	7	4	2	1	0

Step 2: compute net divergence (Q-matrix)

Step 2.1: Compute R(i) for Each Taxon For each taxon i, using: $R(i) = \sum_{k=1}^{n} d(i,k)$

R(A)=(0+4+8+8+9+10+10+10)=59
R(B)=(4+0+8+8+9+10+10+10)=59
...
R(G)=(10+10+7+7+4+2+1)=41
R(H)=(10+10+7+7+4+2+1)=41

Taxon	A	B	C	D	E	F	G	H
A	0	4	8	8	9	10	10	10
B	4	0	8	8	9	10	10	10
C	8	8	0	2	5	6	7	7
D	8	8	2	0	5	6	7	7
E	9	9	5	5	0	3	4	4
F	10	10	6	6	3	0	2	2
G	10	10	7	7	4	2	0	1
H	10	10	7	7	4	2	1	0

Step 2: compute net divergence (Q-matrix)

Step 2.2: the Q-value for each pair of taxa (i,j) is calculated as: $Q(i,j) = (n-2)d(i,j) - R(i) - R(j)$

- n is the total number of taxa.
- $d(i,j)$ is the distance between taxa *i* and *j*.
- $R(i)$ and $R(j)$ are the net divergences for taxa *i* and *j*.

Example: $Q_{(A,B)} = (8 - 2) \times 4 - 59 - 59 = -94$

Taxon	A	B	C	D	E	F	G	H
A		-94.0	-54.0	-54.0	-44.0	-38.0	-40.0	-40.0
B	-94.0		-54.0	-54.0	-44.0	-38.0	-40.0	-40.0
C	-54.0	-54.0		-74.0	-52.0	-46.0	-42.0	-42.0
D	-54.0	-54.0	-74.0		-52.0	-46.0	-42.0	-42.0
E	-44.0	-44.0	-52.0	-52.0		-60.0	-56.0	-56.0
F	-38.0	-38.0	-46.0	-46.0	-60.0		-68.0	-68.0
G	-40.0	-40.0	-42.0	-42.0	-56.0	-68.0		-76.0
H	-40.0	-40.0	-42.0	-42.0	-56.0	-68.0	-76.0	

Step 3: merge the closest pair and compute new distances

Step 3.1: Merge the pair which has the highest Q-value in the Q-matrix.

The closest pair is made by A and B which have to be merged into a new cluster.

Taxon	A	B	C	D	E	F	G	H
A		-94.0	-54.0	-54.0	-44.0	-38.0	-40.0	-40.0
B	-94.0		-54.0	-54.0	-44.0	-38.0	-40.0	-40.0
C	-54.0	-54.0		-74.0	-52.0	-46.0	-42.0	-42.0
D	-54.0	-54.0	-74.0		-52.0	-46.0	-42.0	-42.0
E	-44.0	-44.0	-52.0	-52.0		-60.0	-56.0	-56.0
F	-38.0	-38.0	-46.0	-46.0	-60.0		-68.0	-68.0
G	-40.0	-40.0	-42.0	-42.0	-56.0	-68.0		-76.0
H	-40.0	-40.0	-42.0	-42.0	-56.0	-68.0	-76.0	

Step 3: merge the closest pair and compute new distances

Step 3.2: compute new distances between the new cluster U and the remaining taxa:

To do so use the formula: $d(U,X) = \frac{1}{2}[d(A,X) + d(B,X) - d(A,B)]$

In this formula:

- $d_{(A,B)}$ is the original distance between *A* and *B*
- $d_{(A,X)}$ and $d_{(B,X)}$ are distances from *A* and *B* to any other taxon *X*

Taxon	(A,B)	C	D	E	F	G	H
(A,B)	0	6	6	7	8	8	8
C	6	0	2	5	6	7	7
D	6	2	0	5	6	7	7
E	7	5	5	0	3	4	4
F	8	6	6	3	0	2	2
G	8	7	7	4	2	0	1
H	8	7	7	4	2	1	0

Step 4: compute the branch lengths of merged nodes

The branch lengths from the newly created node *U* to its child nodes *A* and *B* are computed as:

$$u_A = \frac{1}{2}\left[d(A,B) + \frac{R(A) - R(B)}{n-2}\right] \qquad \text{and} \qquad u_B = d(A,B) - u_A$$

Example for A and B:

$$u_A = \frac{1}{2}\left[4 + \frac{59 - 59}{8-2}\right] = \frac{1}{2}(4) = 2 \qquad \text{and} \qquad u_B = 4 - 2 = 2$$

Step 5: repeat until all taxa are clustered

5.1 - Compute R for each taxa and a Q-matrix with the updated distances.

5.2 - Select next closest pair and merge, compute new distance matrix.

5.3 - calculate branchlengths

Continue until all taxa are merged into a NJ tree!

Key properties of UPGMA

✅ simple & fast
✅ produces a rooted tree
❌ assumes a constant molecular clock
❌ sensitive to unequal evolutionary rates

Key properties of NJ

✅ simple & fast
✅ produces an unrooted tree
✅ does not assume clock-like molecular evolution
❌ less accurate than Maximum Likelihood (ML) or Bayesian Inference (BI)

Character-based methods

Character-based methods compare all sequences by **considering one character** (nucleotides or amino acids) in the alignment **at a time**.

As character-based methods incorporate **evolutionary models** making these methods more accurate than distance-based methods.

These methods construct multiple trees that are evaluated and ranked, resulting in selection of the best tree. Character-based methods include:

- Maximum Parsimony - **MP**
- Maximum Likelihood - **ML**
- Bayesian Inference - **BI**

PARSIMONY

Maximum Parsimony (MP) is a character-based phylogenetic method that seeks to reconstruct the tree with **the fewest evolutionary changes** (substitutions) necessary to explain the observed data.

✅ Minimizes total character changes.

✅ Does not require an explicit evolutionary model, but posses an implicit one

❌ Computationally expensive for large datasets - the number of trees grows exponentially!

❌ Sensitive to homoplasy (convergent evolution or reversals).

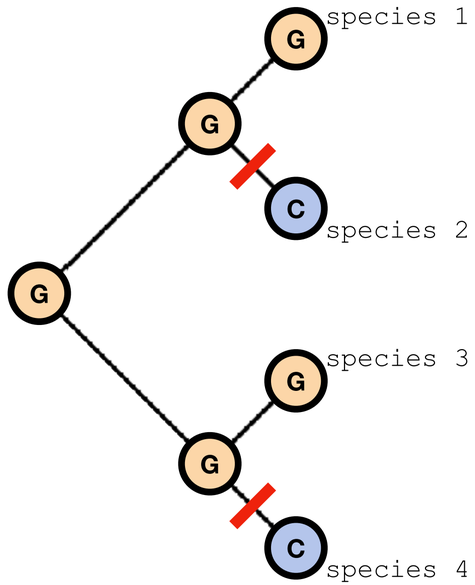
❌ Does not correct for multiple substitutions (it has no proper statistical model).

```
                1 2 3 4 5 6 7 8
species 1       A G G A C C G A
species 2       A C T T T C G G
species 3       A G G A C C T T
species 4       A C G T T C T T
species 5       A G G G C C G C
```

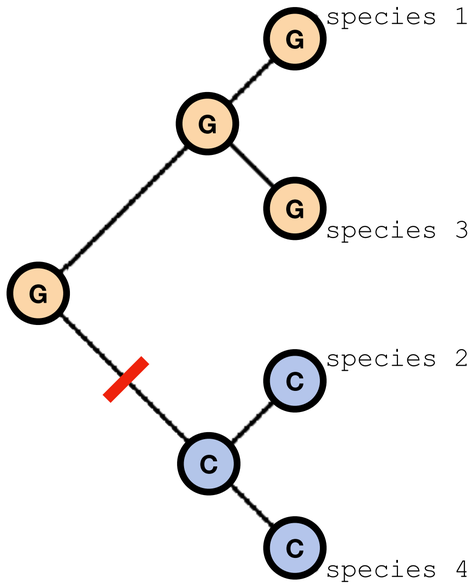
Parsimony uses only **informative sites** (columns), which:
• contains at least two different character states
• each state appears in at least two taxa

1 - not informative, has only one character state (**invariant**)
2 - **informative**, has two character states and each is present in more than one taxa
3 - not informative, has two character states but one is present in just one taxa (**singleton**)
4 - **informative**, has three character states and two is present in more than one taxa
5 - ...
6 - ...
7 - ...
8 - ...

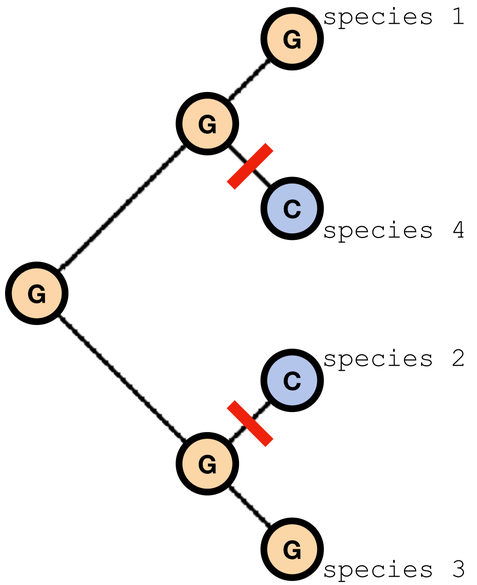
```
          1 2 3 4 5 6 7 8
species 1 A G G A C C G A
species 2 A C T T T C G G
species 3 A G G A C C T T
species 4 A C G T T C T T
            *
```



SCORE = 2



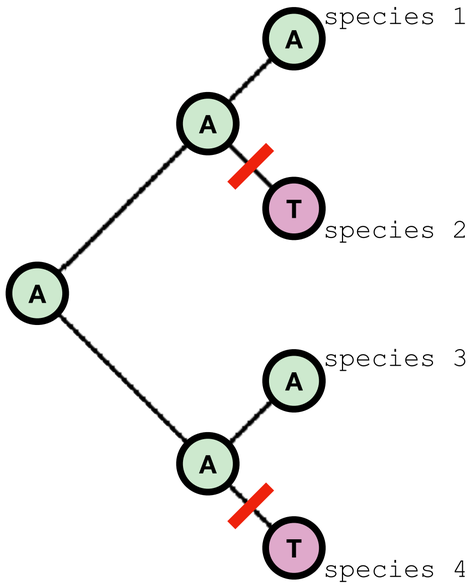
SCORE = 1



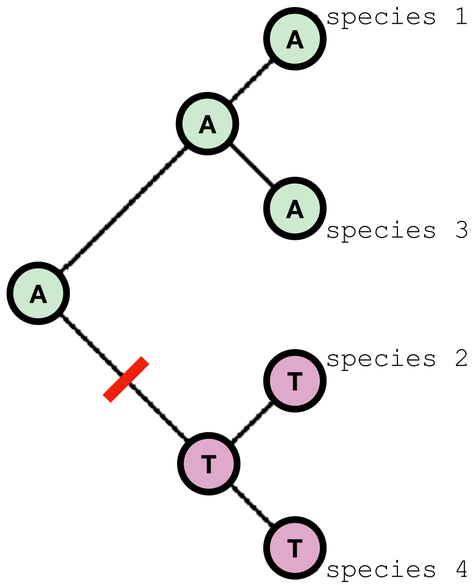
SCORE = 2

```
          1 2 3 4 5 6 7 8
species 1 A G G A C C G A
species 2 A C T T T C G G
species 3 A G G A C C T T
species 4 A C G T T C T T
            *
```

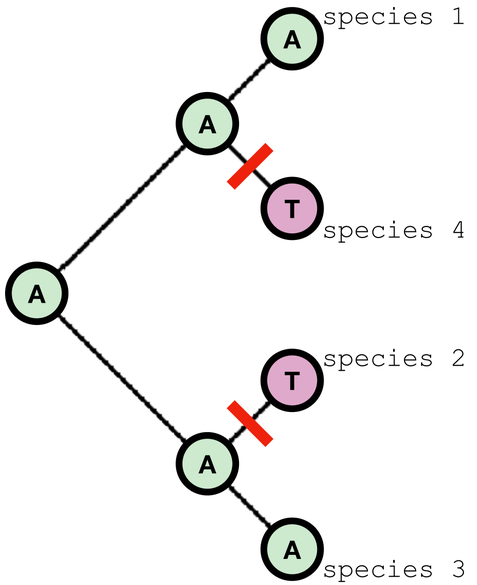
... site 4 is congruent with site 2!



SCORE = 2



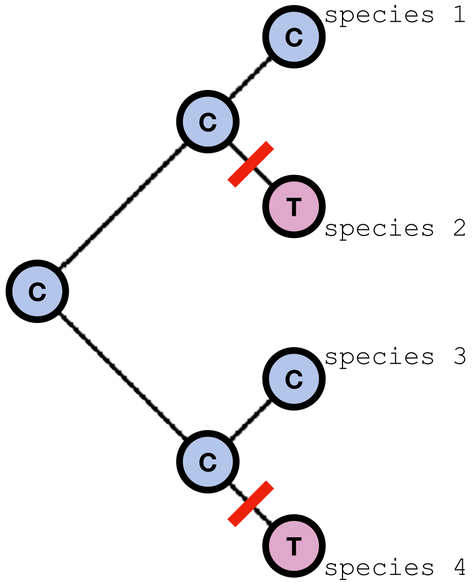
SCORE = 1



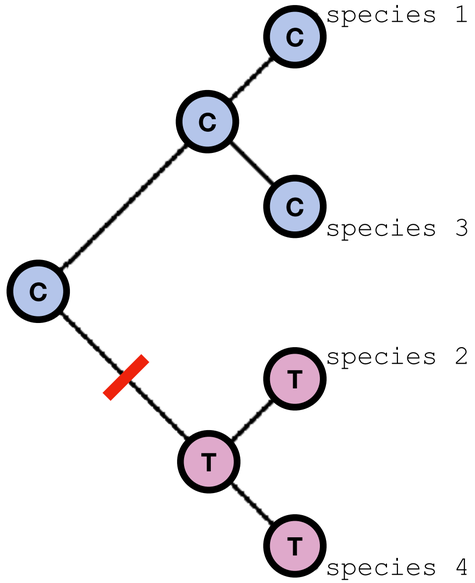
SCORE = 2

```
                1 2 3 4 5 6 7 8
species 1       A G G A C C G A
species 2       A C T T T C G G
species 3       A G G A C C T T
species 4       A C G T T C T T
                        *
```

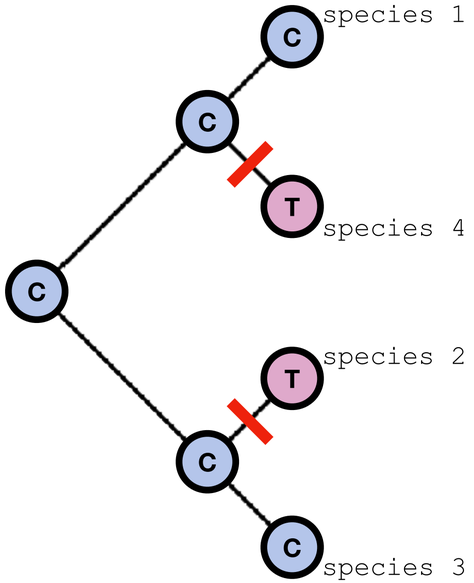
... site 5 is congruent with site 2!



SCORE = 2



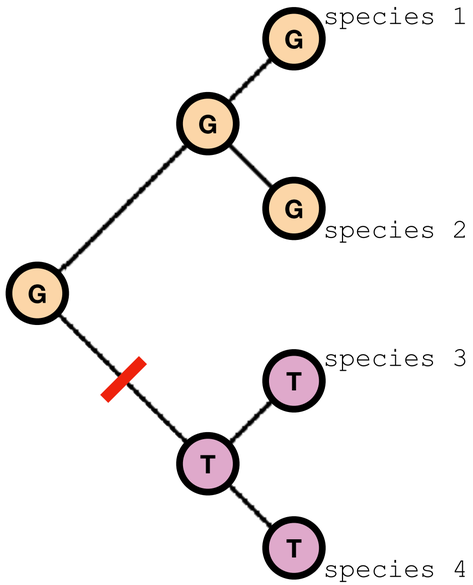
SCORE = 1



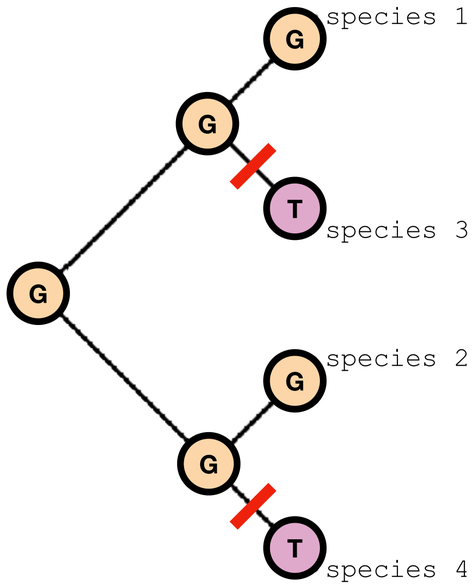
SCORE = 2

```
                1 2 3 4 5 6 7 8
species 1       A G G A C C G A
species 2       A C T T T C G G
species 3       A G G A C C T T
species 4       A C G T T C T T
                        *
```

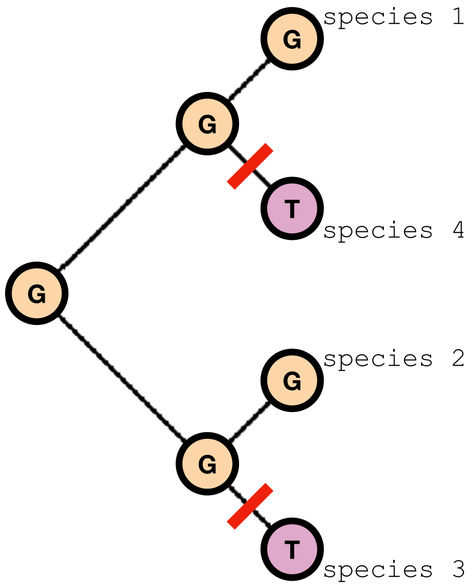
… site 7 is incongruent with site 2!



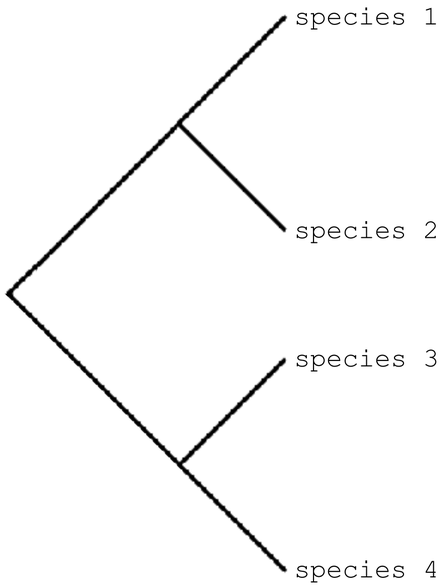
SCORE = 1

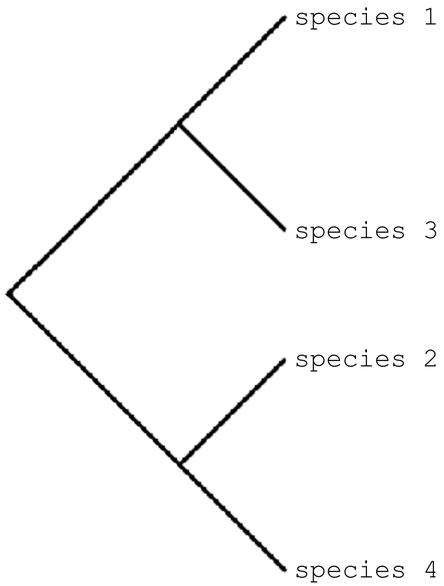


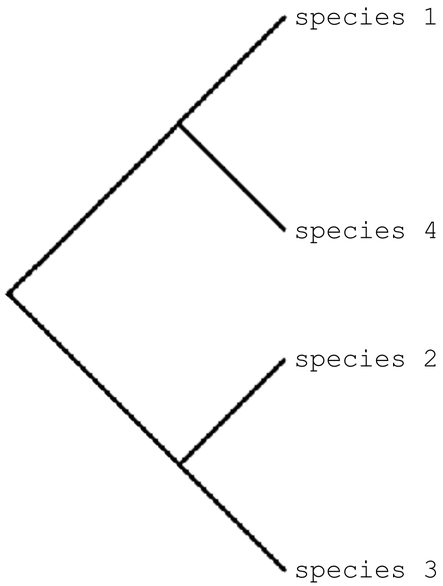
SCORE = 2



SCORE = 2







charachter (site)	topology 1	topology 2	topology 3
2	2	1	2
4	2	1	2
5	2	1	2
7	1	2	1
total	7	**5**	7

Here we could score all the trees, but we learned it is impossible …

…. how can we explore treespace without having to score each tree? …

… using **tree rearrangement methods** ❤️

Nearest Neighbour Interchange (NNI) is a local search method that makes small, localized changes to the tree. It is the simplest and fastest tree rearrangement algorithm!

How NNI Works:

1. Take an internal branch (bifurcation) in the tree.
2. Consider the four subtrees around this branch.
3. Swap neighboring subtrees to create two alternative topologies.
4. Evaluate the new topologies using the given scoring function.
5. Evaluate the new tree and accept if it improves the score - or in some probabilistic approaches, accept worse trees to avoid local optima!

NNI is fast but may get stuck in local optima since it only explores a small number of alternative trees at each step.

Subtree Pruning and Regrafting (SPR) is a more extensive tree rearrangement method than NNI. It moves entire subtrees rather than just swapping immediate neighbors.

How SPR Works:

1. Prune (cut) a subtree from the main tree.
2. Regraft (reattach) the subtree at a different position.
3. Evaluate the new tree and accept if it improves the score.

Tree Bisection and Reconnection (TBR) is the most aggressive of these tree rearrangement methods and explores a much larger space of possible tree topologies. It has the lesser chance of local optima!

How TBR Works:

1. Cut the tree into two large subtrees by breaking an internal branch.
2. Reconnect the two subtrees by joining any two branches.
3. Evaluate the new tree and accept if it improves the score.

FINISH