

Evolutionary Tree Reconstruction

Given observations of similarities or differences between k species, find the hypothesis (tree) that best explains the data with respect to some criterion:

- Maximum parsimony (character data)
- **Minimum evolution (distance data)**
- Maximum Likelihood (character data)

Assumptions:

- Selection dominates
- Mutations are rare
- No multiple substitutions

Parsimony:

Character data
Find the tree that requires the fewest changes to explain the data

Assumptions:

- Neutral mutation dominates
- Multiple substitutions occur

Minimum Evolution:

Distance data
Find the tree that best fits the pairwise distances between taxa

Maximum Likelihood:

Character data
Find the most likely tree

Distance-based methods

- How to obtain a distance matrix
- Correcting for multiple substitutions
- Fitting distances to a tree
 - Conditions for obtaining an exact fit
 - Ultrametric distances
 - Additive distances
 - Minimum Evolution: finding the tree with the best fit
 - Greedy algorithms
 - UPGMA
 - NeighborJoining

How distance matrices are obtained

Given sequences from k taxa

- Construct a multiple sequence alignment
- Determine pairwise distance from each pair of taxa *using the MSA*
- Correct for multiple substitutions

Multiple Sequence Alignment

```

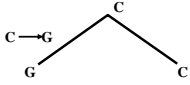
~~~~~ALTEKQEALLKQSWEVLKQNI PAHSLR LRFAL I IEAA...
~~~~~MALTEKQEALLKQSWEVLKQNI PAHSLR LRFAL I IEAA...
~~~~~MALTEKQEALLKQSWEVLKQNI P GHSLR LRFAL I IEAA...
~~~~~EALLKQSWEVLKQNI P GHSLC LRFAL I IEAA...
  
```

The distance between *taxon i* and *taxon j* is the distance of the pairwise alignment induced by the MSA.

Distance-based methods

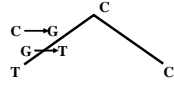
- How to obtain a distance matrix
- Correcting for multiple substitutions
- Fitting distances to a tree
 - Conditions for obtaining an exact fit
 - Ultrametric distances
 - Additive distances
 - Minimum Evolution: finding the tree with the best fit
 - Greedy algorithms
 - UPGMA
 - NeighborJoining

Substitution patterns



Single substitution:
- 1 change, 1 difference

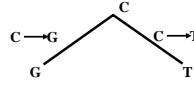
...G...
...C...



Multiple substitution:
- 2 changes, 1 difference

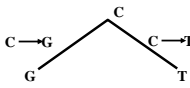
...T...
...C...

Substitution patterns



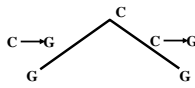
Coincidental substitution:

Substitution patterns



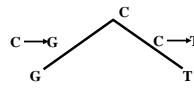
Coincidental substitution:
- 2 changes, 1 difference

...G...
...T...



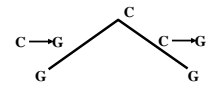
Parallel substitution:

Substitution patterns



Coincidental substitution:
- 2 changes, 1 difference

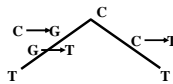
...G...
...T...



Parallel substitution:
- 2 changes, no difference

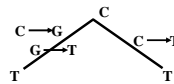
...G...
...G...

Substitution patterns



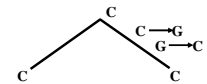
Convergent substitution:

Substitution patterns



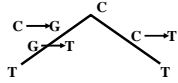
Convergent substitution:
- 3 changes, no difference

...T...
...T...



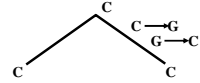
Back substitution:

Substitution patterns



Convergent substitution:
– 3 changes, no difference

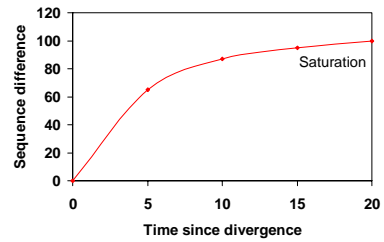
...T...
...T...



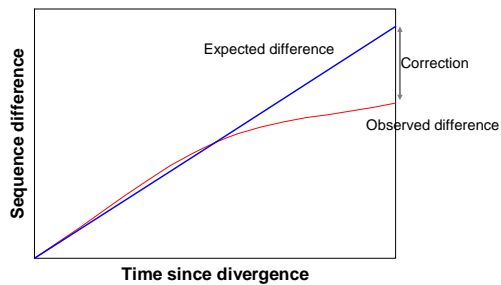
Back substitution:
– 2 changes, no difference

...C...
...C...

Multiple substitutions



Correcting for multiple substitutions

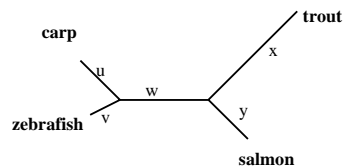


Distance-based methods

- How to obtain a distance matrix
- Correcting for multiple substitutions
- Fitting distances to a tree
 - Conditions for obtaining an exact fit
 - Additive distances
 - Ultrametric distances
 - Minimum Evolution: finding the tree with the best fit
 - Greedy algorithms
 - UPGMA
 - NeighborJoining

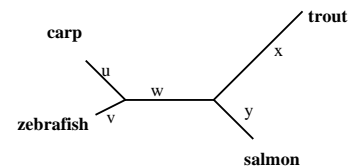
Match distance matrix to branch lengths

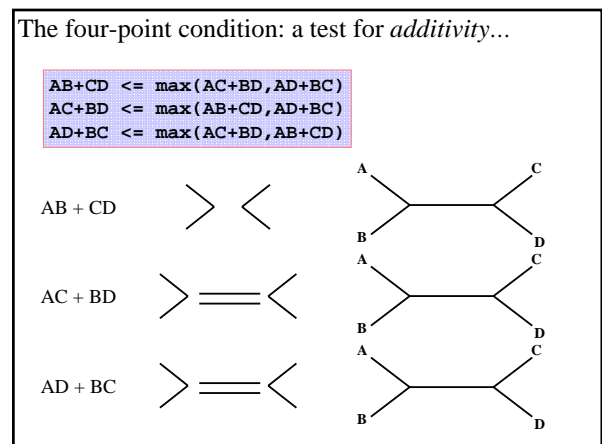
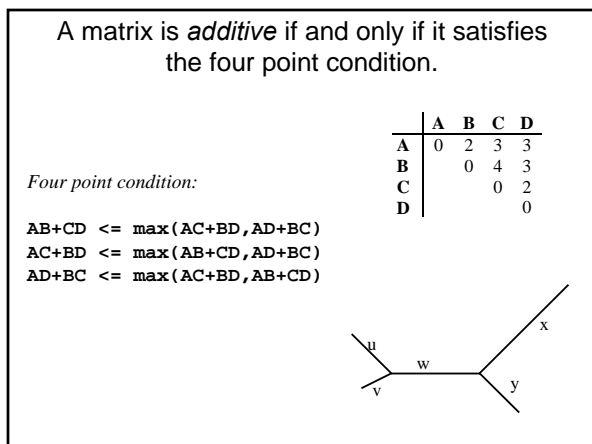
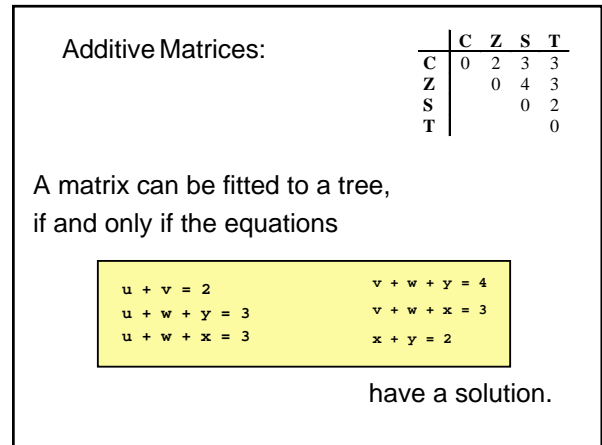
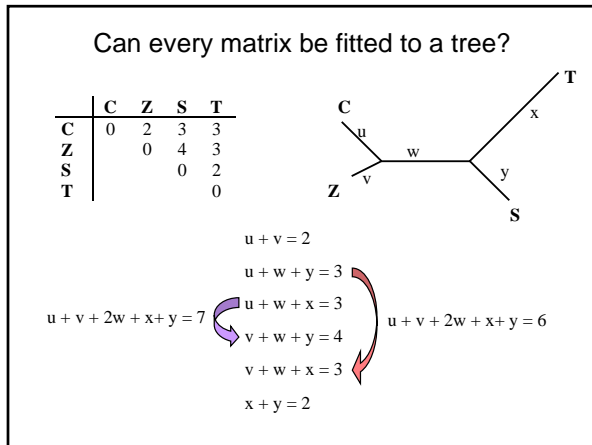
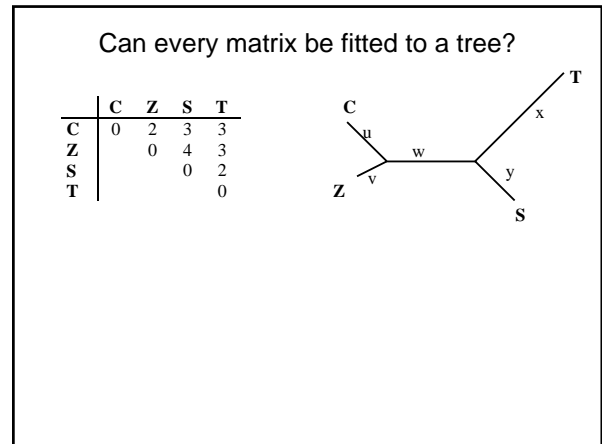
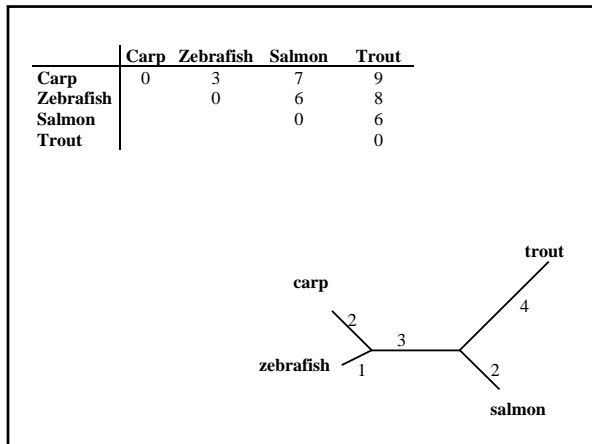
	Carp	Zebrafish	Salmon	Trout	
Carp	0	3	7	9	Observed distances
Zebrafish		0	6	8	
Salmon			0	6	
Trout				0	



	Carp	Zebrafish	Salmon	Trout	
Carp	0	3	7	9	Observed distances
Zebrafish		0	6	8	
Salmon			0	6	
Trout				0	

$$\begin{aligned}
 u + v &= 3 \\
 u + w + y &= 7 \\
 u + w + x &= 9 \\
 v + w + y &= 6 \\
 v + w + x &= 8 \\
 x + y &= 6
 \end{aligned}$$





	A	B	C	D
A	0	2	3	3
B		0	4	3
C			0	2
D				0

$AB+CD \leq \max(AC+BD, AD+BC)$
 $AC+BD \leq \max(AB+CD, AD+BC)$
 $AD+BC \leq \max(AC+BD, AB+CD)$

Does this matrix satisfy the four point condition?

	A	B	C	D
A	0	3	9	7
B		0	8	6
C			0	6
D				0

$AB+CD \leq \max(AC+BD, AD+BC)$
 $AC+BD \leq \max(AB+CD, AD+BC)$
 $AD+BC \leq \max(AC+BD, AB+CD)$

Does this matrix satisfy the four point condition?

We know the matrix

	Carp	Salmon	Zebrafish	Trout
Carp	0	6	3	7
Salmon		0	7	7
Zebrafish			0	8
Trout				0

We don't know the tree topology

	Carp	Salmon	Zebrafish	Trout
Carp	0	6	3	7
Salmon		0	7	7
Zebrafish			0	8
Trout				0

$AB+CD \leq \max(AC+BD, AD+BC)$
 $AC+BD \leq \max(AB+CD, AD+BC)$
 $AD+BC \leq \max(AC+BD, AB+CD)$

The four-point condition also gives the topology:

$AB + CD \quad \rangle \! = \! \langle$
 $AC + BD \quad \rangle \! < \! \langle$
 $AD + BC \quad \rangle \! = \! \langle$

- The matrix is additive
- The four point condition holds:
 - $AB+CD \leq \max(AC+BD, AD+BC)$
 - $AC+BD \leq \max(AB+CD, AD+BC)$
 - $AD+BC \leq \max(AC+BD, AB+CD)$
- The equations
 - $u + v = AB$
 - $u + w + y = AC$
 - $u + w + x = AD$
 - $v + w + y = BC$
 - $v + w + x = BD$
 - $x + y = CD$

Equivalent statements

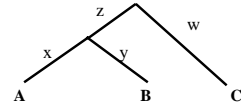
have a solution.

- The topology and branch lengths are uniquely determined.

Distance-based methods

- How to obtain a distance matrix
- Correcting for multiple substitutions
- Fitting distances to a tree
 - Conditions for obtaining an exact fit
 - Additive distances
 - **Ultrametric distances**
 - Minimum Evolution: finding the tree with the best fit
 - Heuristics
 - UPGMA
 - NeighborJoining

Ultrametric distances



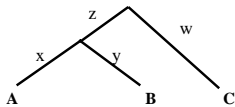
Consider

- a rooted tree with
- constant mutation rate on all branches (molecular clock)

Note:

1. Same distance from the root to every leaf
2. $D[A,B] < D[A,C] = D[B,C]$
3. $x+y < x+z+w = y+z+w$

Three point condition



$$x+y < x+z+w = y+z+w$$

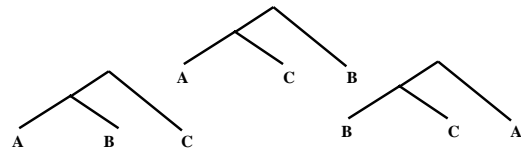
For every triple, $\{A,B,C\}$ in T

- $AB \leq \max(AC, BC)$
- $AC \leq \max(AB, BC)$
- $BC \leq \max(AC, BC)$

We know the matrix

	A	B	C
A	0	2	3
B		0	4
C			0

We don't know the tree topology



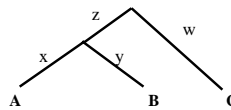
Is the matrix ultrametric?

Equivalent statements

A matrix

- is ultrametric
- satisfies the three point condition
- fits a rooted tree with equal distances from the root to all leaves
- mutation rates are the same in all lineages.

Three point condition an example



	A	B	C
A	0	7	4
B		0	7

For every triple, $\{A,B,C\}$ in T

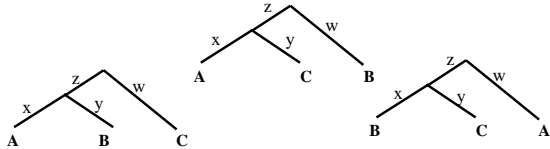
- $AB \leq \max(AC, BC)$
- $AC \leq \max(AB, BC)$
- $BC \leq \max(AC, BC)$

Three point condition an example

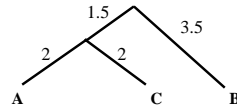
For every triple, $\{A,B,C\}$ in T

- $AB \leq \max(AC,BC)$
- $AC \leq \max(AB,BC)$
- $BC \leq \max(AC,BC)$

	A	B	C
A	0	7	4
B		0	7



Three point condition



	A	B	C
A	0	7	4
B		0	7

For every triple, $\{A,B,C\}$ in T

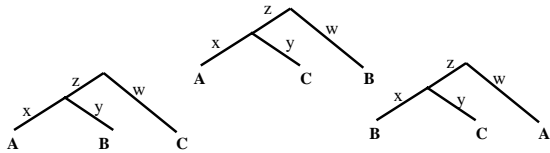
- $AB \leq \max(AC,BC)$
- $AC \leq \max(AB,BC)$
- $BC \leq \max(AC,BC)$

Three point condition an example

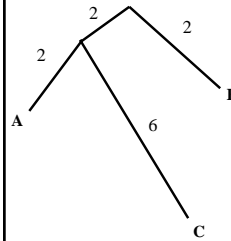
For every triple, $\{A,B,C\}$ in T

- $AB \leq \max(AC,BC)$
- $AC \leq \max(AB,BC)$
- $BC \leq \max(AC,BC)$

	A	B	C
A	0	6	8
B		0	10



Three point condition another example



	A	B	C
A	0	6	8
B		0	10

All ultrametric matrices fit rooted trees
but not all rooted trees are ultrametric.
If the matrix is not ultrametric,
the closest pair may not be neighbors

Summary

- A matrix is *additive* if it satisfies the four point condition.
- A tree defines a *tree metric*, $T[i,j]$; *i.e.*, the pairwise distances between all pairs of leaves.
- All tree metrics are additive.
- If a matrix, $O[i,j]$, is additive
 - there exists a unique tree topology with branch lengths such that $T[i,j] = O[i,j]$.
 - This tree can be obtained in polynomial time.
- In real life, observed distance matrix, $O[i,j]$ is never additive.

Summary, cont'd

- A matrix is *ultrametric* if it satisfies the three point condition.
- All ultrametric matrices fit rooted trees.
- Not all rooted tree metrics are ultrametric.
- An ultrametric tree
 - satisfies the molecular clock hypothesis.
 - All distances from the root to a leaf are the same.
 - Its branch lengths are proportional to time.
- For $k > 3$,
 - All ultrametric matrices are additive
 - But, an additive matrix is *not necessarily* ultrametric.

Distance-based methods

- How to obtain a distance matrix
- Correcting for multiple substitutions
- Fitting distances to a tree
 - Conditions for obtaining an exact fit
 - Additive distances
 - Ultrametric distances
 - Minimum Evolution: finding the tree with the best fit
 - Greedy algorithms
 - UPGMA
 - NeighborJoining

How to reconstruct a tree when $O[i,j]$ is not additive.

Exhaustive search:

- Consider all trees and select the tree that is the “closest” fit; *i.e.*, the difference between $O[i,j]$ and $T[i,j]$ is smallest
- Measures of fit: L_α norms

$$\|O, T\|_\alpha = \left(\sum_{i < j} |O[i, j] - T[i, j]|^\alpha \right)^{1/\alpha}, \alpha = 1, 2, \dots$$

$$\|O, T\|_\infty = \max \{ |O[i, j] - T[i, j]| \}$$

Minimum Evolution

Given an observed distance matrix, $O[i,j]$

For every topology, T , with k leaves

Let $D^*(T)$ be the matrix that minimizes

$$L_2 = \|O, T\|_2 = \sqrt{\sum_{i < j} |O[i, j] - T[i, j]|^2}$$

Let $Score(T)$ be the sum of the branch lengths of $D^*(T)$

The best tree is the tree with minimum score.

Distance-based methods

- How to obtain a distance matrix
- Correcting for multiple substitutions
- Fitting distances to a tree
 - Conditions for obtaining an exact fit
 - Additive distances
 - Ultrametric distances
 - Minimum Evolution: finding the tree with the best fit
 - Greedy algorithms
 - UPGMA
 - NeighborJoining

Unweighted Pair Group Method with Arithmetic Means

UPGMA