Categories of tree reconstruction methods

	Parsimony	Distance	Maximum likelihood estimation	Bayesian methods
Character data	x		x	x
Pairwise distances		x		

Distance-based methods

- Obtaining a distance matrix from an alignment and correcting for multiple substitutions — Nov 29th
- Fitting distances to a tree
  - Conditions for obtaining an exact fit — Dec 1st
    - Additive distances
    - Ultrametric distances
  - Greedy algorithms — Dec 6th
    - 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

Last Tuesday, we discussed distance correction using the Jukes Cantor model.

Briefly...

Calculating distances from MSAs

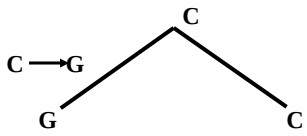
The distance between *taxon i* and *taxon j* is the distance of the pairwise alignment *induced by the MSA.*

```
(H)  AC_TCAT
(R)  A_GTCAT
(P)  ACGTCCT
(C)  ACCAGAT
```

	Rabbit	Pig	Chicken
Human	4	5	8
Rabbit	0	5	11
Pig		0	11

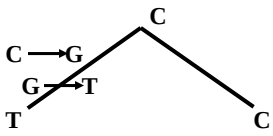
d(x,y)=3
d(x,_)=2

Substitution patterns



Single substitution:
- 1 change, 1 difference

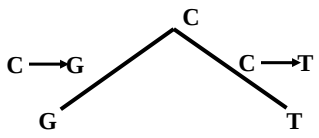
```
...G...
...C...
```



Multiple substitution:
- 2 changes, 1 difference

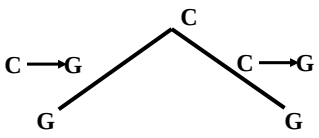
```
...T...
...C...
```

Substitution patterns



Coincidental substitution:
- 2 changes, 1 difference

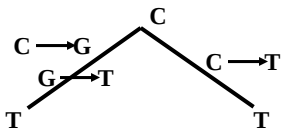
```
...G...
...T...
```



Parallel substitution:
- 2 changes, no difference

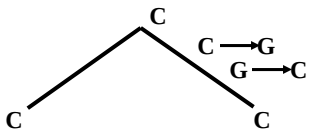
```
...G...
...G...
```

Substitution patterns



Convergent substitution:
- 3 changes, no difference

```
...T...
...T...
```



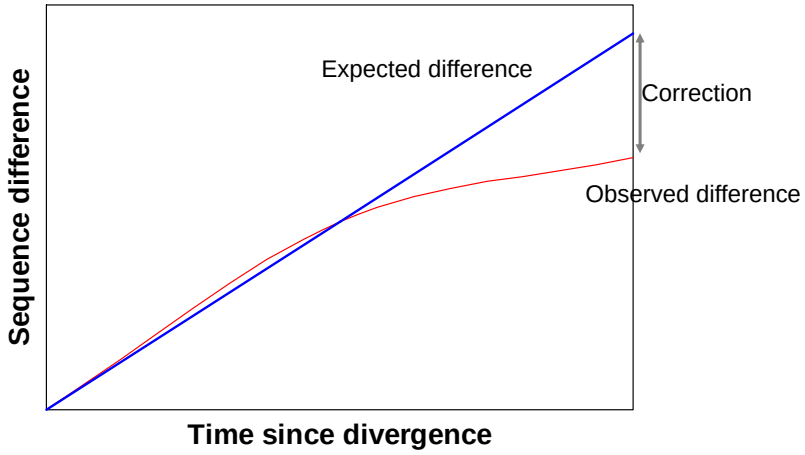
Back substitution:
- 2 changes, no difference

```
...C...
...C...
```

How distance matrices are obtained

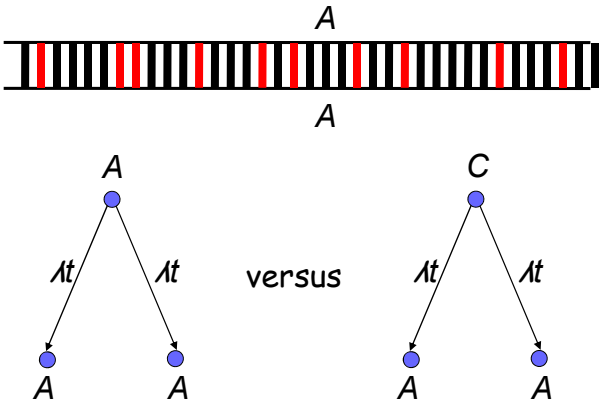
Given sequences from *k* taxa
- Construct a multiple sequence alignment
- Determine pairwise distance from each pair of taxa *using the MSA*
- Correct for multiple substitutions

Correcting for multiple substitutions



Correcting for multiple substitutions

Given *m* mismatches in a PW alignment of length *n*, estimate the actual number of substitutions



Correcting for multiple substitutions

Given *m* mismatches in a PW alignment of length *n*, estimate the actual number of substitutions

Note that:

p' = m/n is an estimator for the underlying probability of a mismatch, *p = P(mismatch)*

The number of substitutions is *2λt*

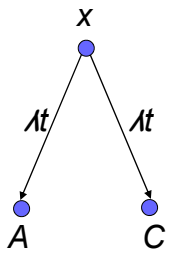
Strategy

Propose a Markov model of substitution

Derive p=f(*λt*)

$\lambda t = f^{-1}(p)$

$E[subs/site] = 2 f^{-1}(p)$



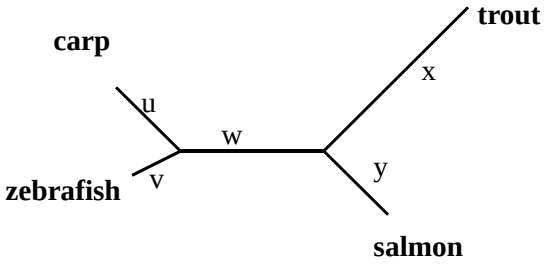
Distance-based methods

- Obtaining a distance matrix from an alignment and correcting for multiple substitutions
- Fitting distances to a tree
  - ➢ Conditions for obtaining an exact fit
    - ➢ Additive distances
    - Ultrametric distances
  - – Greedy algorithms
    - UPGMA
    - NeighborJoining

Match distance matrix to branch lengths

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

Observed distances



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

Observed distances

$u + v = 3$

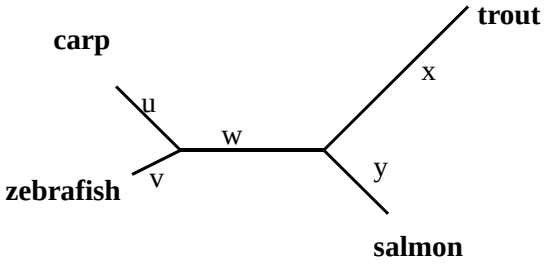
$u + w + y = 7$

$u + w + x = 9$

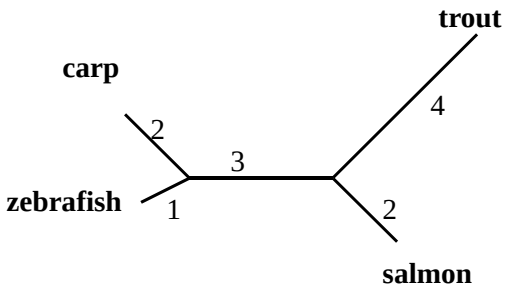
$v + w + y = 6$

$v + w + x = 8$

$x + y = 6$

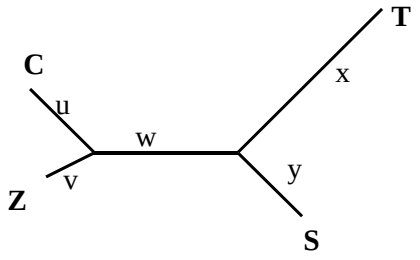


	Carp	Zebrafish	Salmon	Trout
Carp	0	3	7	9
Zebrafish		0	6	8
Salmon			0	6
Trout				0



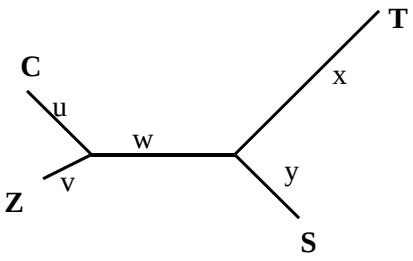
Can every matrix be fitted to a tree?

	C	Z	S	T
C	0	2	3	3
Z		0	4	3
S			0	2
T				0



Can every matrix be fitted to a tree? No!

	C	Z	S	T
C	0	2	3	3
Z		0	4	3
S			0	2
T				0



$u + v = 2$

$u + w + y = 3$

$u + w + x = 3$

$v + w + y = 4$

$v + w + x = 3$

$x + y = 2$

$u + v + 2w + x + y = 7$

$u + v + 2w + x + y = 6$

Additive Matrices:

	C	Z	S	T
C	0	2	3	3
Z		0	4	3
S			0	2
T				0

A matrix can be fitted to a tree,
if and only if the equations

u + v = 2
u + w + y = 3
u + w + x = 3
v + w + y = 4
v + w + x = 3
x + y = 2

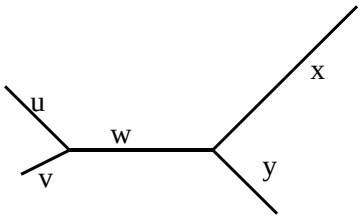
have a solution.

A matrix is *additive* if and only if it satisfies the four point condition.

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

Four point condition:

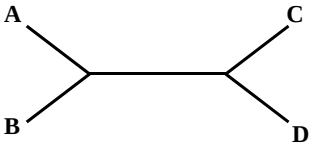
AB+CD <= max(AC+BD,AD+BC)
AC+BD <= max(AB+CD,AD+BC)
AD+BC <= max(AC+BD,AB+CD)



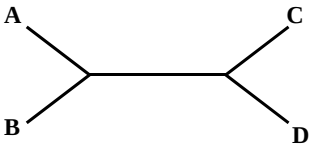
The four-point condition: a test for *additivity…*

AB+CD <= max(AC+BD,AD+BC)
AC+BD <= max(AB+CD,AD+BC)
AD+BC <= max(AC+BD,AB+CD)

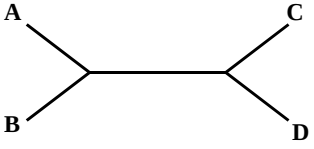
AB + CD



AC + BD



AD + BC



AB+CD <= max(AC+BD,AD+BC)

AC+BD <= max(AB+CD,AD+BC)

AD+BC <= max(AC+BD,AB+CD)

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

Does this matrix satisfy the four point condition?

AB+CD <= max(AC+BD,AD+BC)

AC+BD <= max(AB+CD,AD+BC)

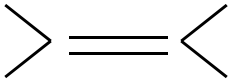
AD+BC <= max(AC+BD,AB+CD)

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

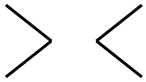
Does this matrix satisfy the four point condition?

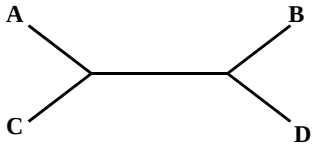
The four-point condition also gives the topology:

AB + CD

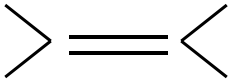


AC + BD





AD + BC



The matrix is additive

The four point condition holds for all quartets in *t* :

AB+CD <= max(AC+BD,AD+BC)

AC+BD <= max(AB+CD,AD+BC)

AD+BC <= max(AC+BD,AB+CD)

The equations

Equivalent statements

```
u + v = AB        v + w + y = BC
u + w + y = AC    v + w + x = BD
u + w + x = AD    x + y = CD
```

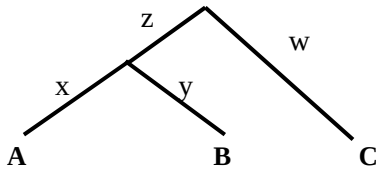
have a solution.

The topology and branch lengths are uniquely determined.

Distance-based methods

- Obtaining a distance matrix from an alignment and correcting for multiple substitutions
- Fitting distances to a tree
  - Conditions for obtaining an exact fit
    - Additive distances
    - Ultrametric distances
  - Greedy algorithms
    - UPGMA
    - NeighborJoining

Ultrametric distances



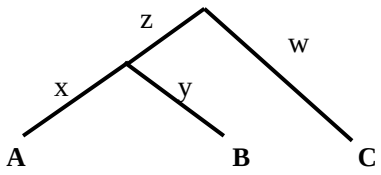
Consider

- 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



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

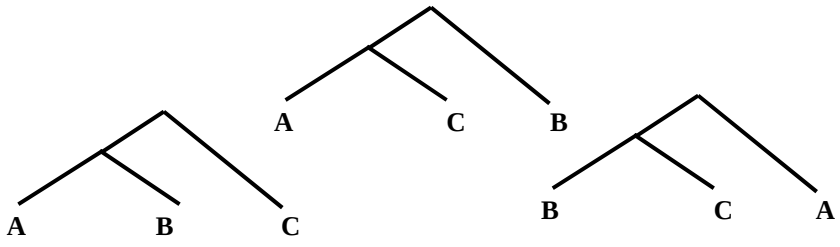
For every triple, {A,B,C} in *T*

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

We know the matrix

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

We don't know the tree topology



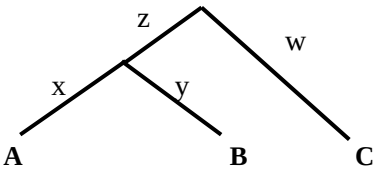
Is the matrix ultrametric?

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*

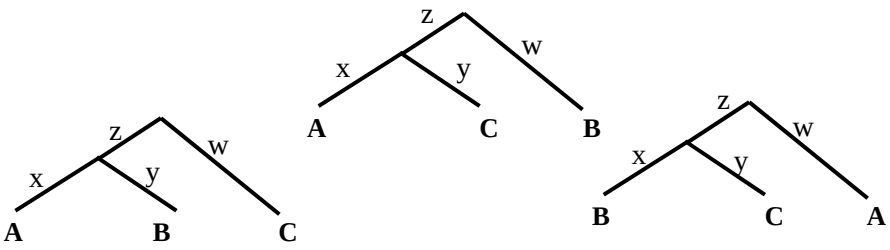
- *AB ≤ max(AC,BC)*
- *AC ≤ max(AB,BC)*
- *BC ≤ max(AC,BC)*

Three point condition an example

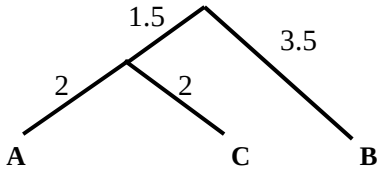
For every triple, {A,B,C} in *T*

- *AB ≤ max(AC,BC)*
- *AC ≤ max(AB,BC)*
- *BC ≤ 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*

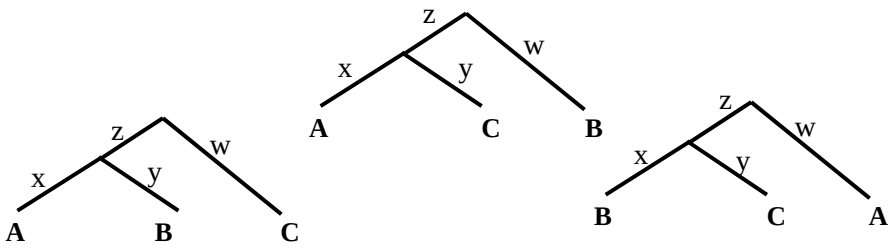
- *AB ≤ max(AC,BC)*
- *AC ≤ max(AB,BC)*
- *BC ≤ max(AC,BC)*

Three point condition an example

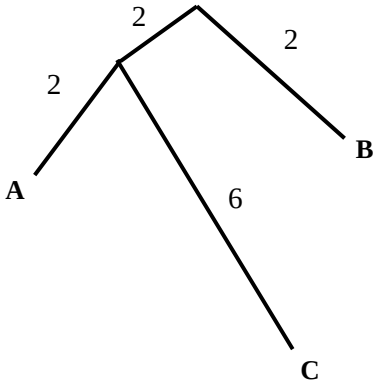
For every triple, {A,B,C} in *T*

- *AB ≤ max(AC,BC)*
- *AC ≤ max(AB,BC)*
- *BC ≤ 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 n*ot 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.