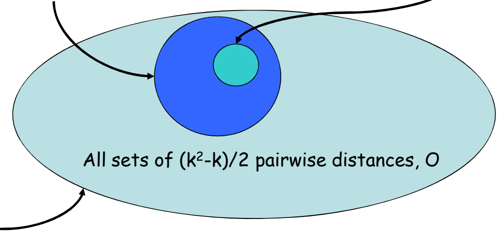
Additive distances

- The distances fit a unique unrooted tree
- A greedy algorithm finds this tree in polynomial time.

Ultrametric distances

- The distances fit a unique tree where all leaves are equidistant from the root.
- A greedy algorithm finds this tree in polynomial time.



Distances are not additive

- Use heuristic search
- Infer branch lengths and topology by optimizing an objective function

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.

When *O[i,j]* is not additive, use heuristic search

Let *t* be the initial tree (generate randomly, use NJ)

Do {

t' = *modify(t)*

Estimate branch lengths for *t'*

If *score(t')* is better than *score(t)*

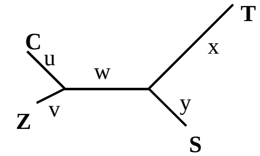
t = *t'*

} until convergence

How to modify *t*?

- Nearest neighbor interchange
- Subtree pruning and regrafting
- Tree bisection and reconnection

Estimating branch lengths: Least Squares



Select u,v,w,x,y to minimize Q,

where $Q(O,T)=\Sigma_i \Sigma_{j \neq i} w_{ij} (O_{ii} - T_{ii})^2$

$W_{ij} = 1$	*Unweighted least squares*
$W_{ij} = (1 / O_{ii})^2$	*Fitch Margoliash*

- *A solution can be found with standard least squares methods*
- *Exact fast algorithms that exploit the structure of the tree have been developed to estimate branch lengths*

Bryant and Waddell, 1998

Gascuel, 1997

Scoring the tree

1. Least Squares: select the tree that minimizes Q
   - *Score (t) = Q(O,T)*

2. Minimum evolution: select the tree that minimizes the sum of the branch lengths
   - *Score(t) = u+v+w+x+y*