Phylogenetic trees

Searching through tree space

Outline

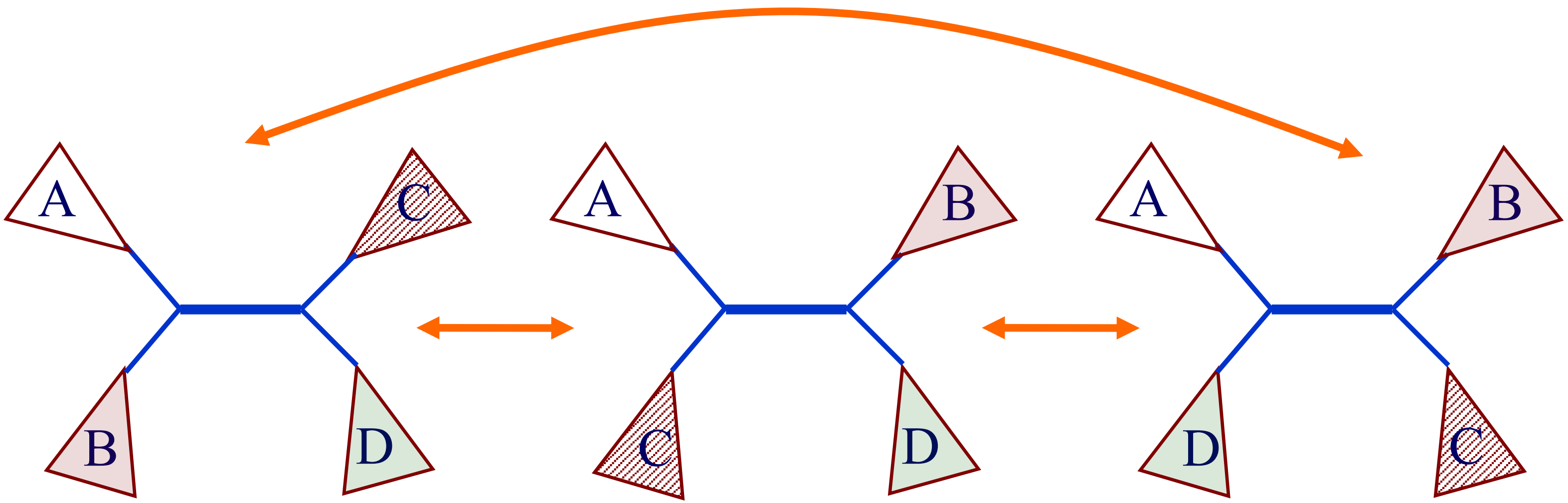
- The second component of a parsimony method for phylogenetic tree estimation: searching through tree space
- A greedy heuristic method
- Exact branch and bound methods

Exploring the Space of Trees

- we've considered how to find the minimum number of changes for a given tree topology
- need some search procedure for exploring the space of tree topologies
  - How do we move from one tree to another?
  - How do we ensure that we fully explore the space of trees?

Heuristic Method: Nearest Neighbor Interchange

- for any internal edge in a tree, there are 3 ways the four subtrees can be grouped
- nearest neighbor interchanges move from one grouping to another

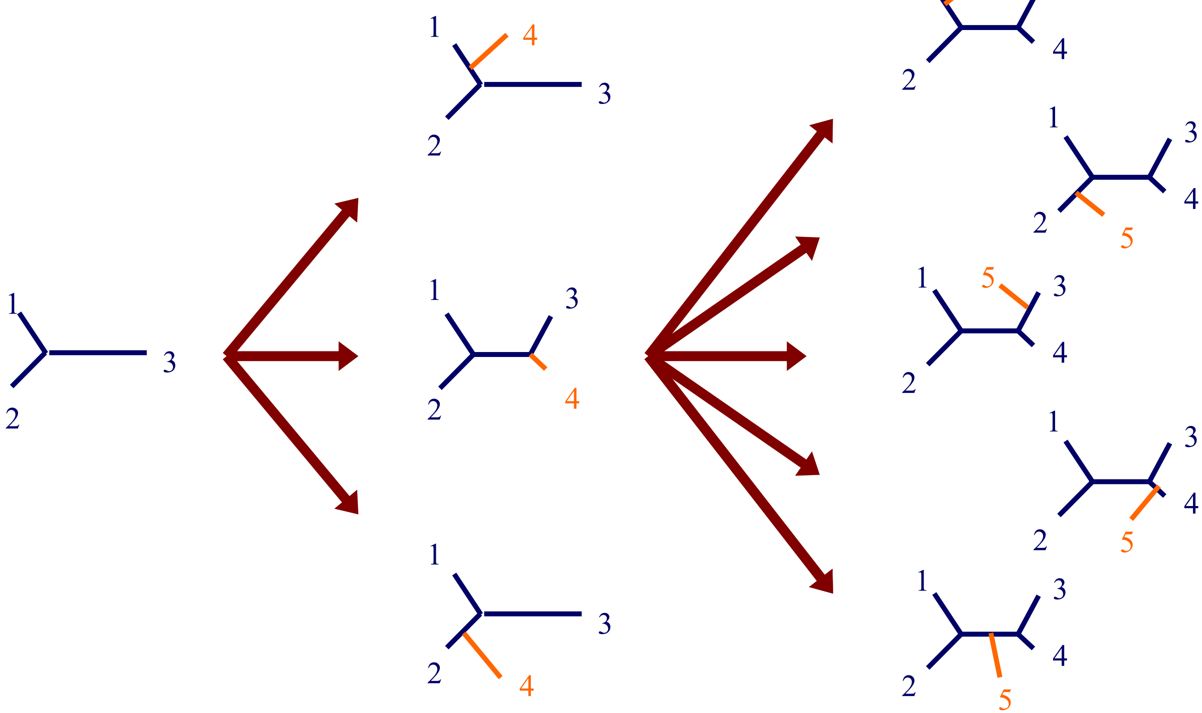


Heuristic Method: Greedy search with Nearest Neighbor Interchange

given: set of leaves *L*
create an initial tree *t* incorporating all leaves in *L*
best-score = parsimony algorithm applied to *t*
repeat
    for each internal edge *e* in *t*
        for each nearest neighbor interchange
            t' ← tree with interchange applied to edge *e* in *t*
            score = parsimony algorithm applied to *t'*
            if *score* < *best-score*
                best-score = *score*
                best-tree = *t'*
    t = *best-tree*
until stopping criteria met (e.g., *best-tree* does not change)

Exact Method: Branch and Bound

- each partial tree represents a set of complete trees
- the parsimony score on a partial tree provides a *lower bound* on the best score in the set



- search by repeatedly selecting the partial tree with the lowest lower bound

Exact Method: Branch and Bound

given: set of leaves L

initialize a queue Q with a partial tree with 3 leaves from L

repeat

  $t \leftarrow$ tree in Q with lowest lower bound

  if t has incorporated all leaves in L

    return t

  else

    create new trees by adding next leaf from L to each branch of t

    compute lower bound for each tree

    put trees in Q sorted by lower bound

Branch and Bound (Alternate Version)

given: set of leaves L

use heuristic method to grow initial tree t'

initialize Q with a partial tree with 3 leaves from L

repeat

  $t \leftarrow$ tree in Q with lowest lower bound

  if t has incorporated all leaves in L

    return t

  else

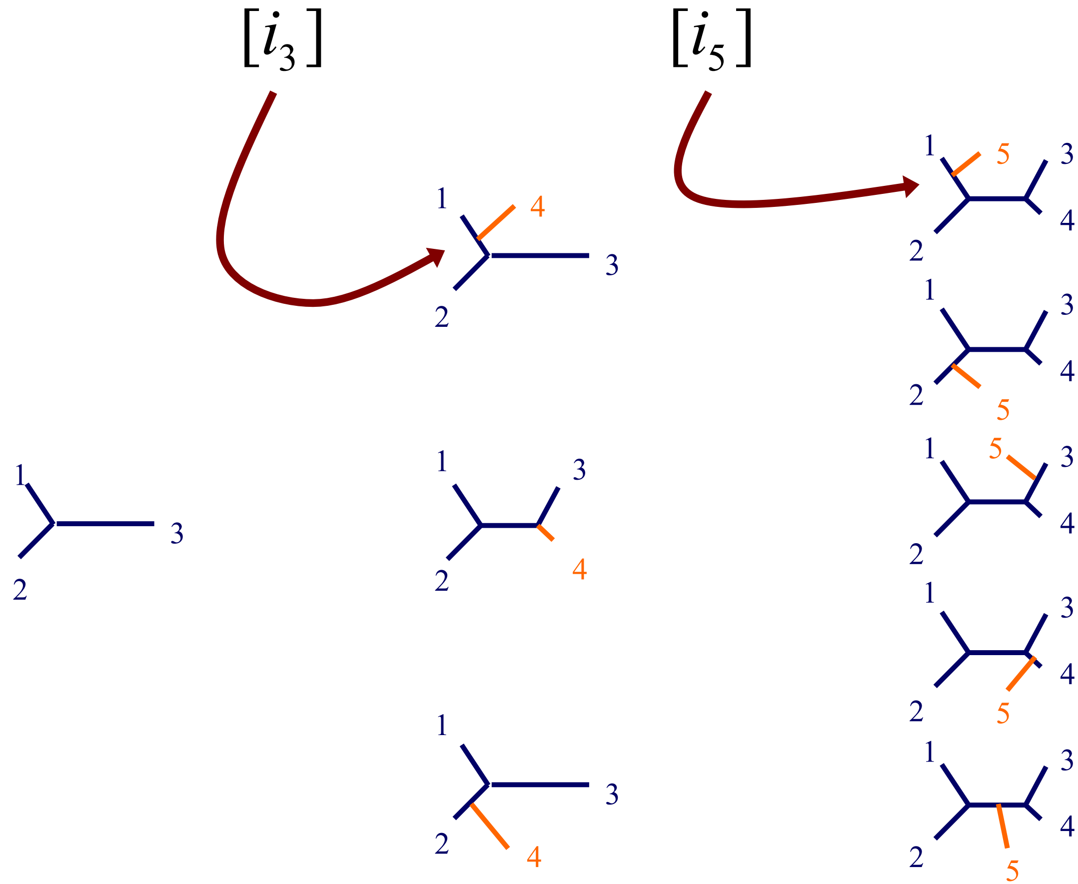
    create new trees by adding next leaf from L to each branch of t

    for each new tree n

      if lower-bound(n) < score(t')

        put n in Q sorted by lower bound

Implementing Branch and Bound (Second Alternate Version)



Implementing Branch and Bound (Second Alternate Version)

- for n sequences, maintain an array of counters

  $$[i_3][i_5][i_7]...[i_{2n-5}]$$

  where i_k takes on values $0\ldots k$
- a complete tree is represented by an assignment of all i_k to non-zero values
- i_k indicates, for a partial tree with k edges, on which edge to add a branch for the next sequence
- $i_k = 0$ indicates a partial tree

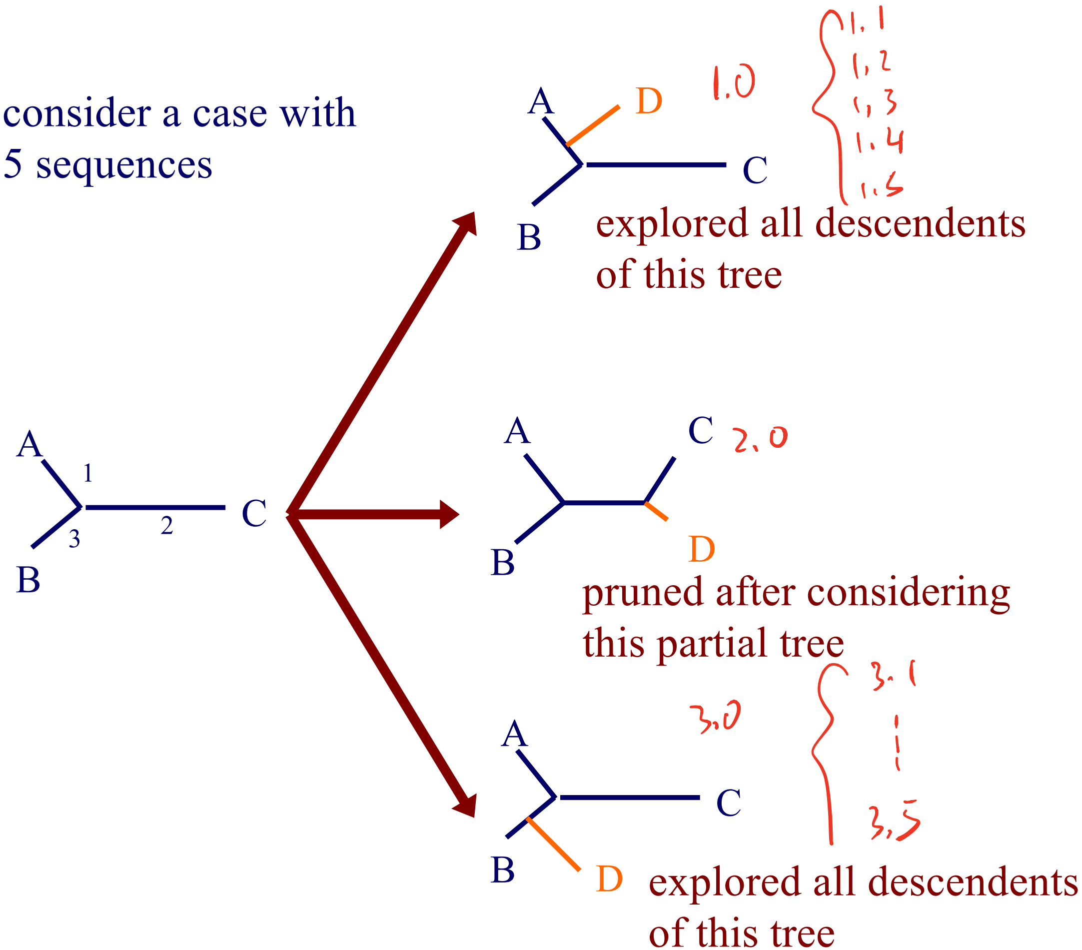
Implementing Branch and Bound

- to search space, roll counters through their allowable numbers (somewhat) like an odometer
  - rightmost counter moves fastest
  - whenever a counter is 0, all counters to the right of it must be 0 also
  - test cost of (partial) tree at each tick of odometer
  - have odometer skip when pruning occurs

$$[i_3][i_5][i_7]\ldots[i_{2n-5}]$$

Implementing Branch and Bound

consider a case with
5 sequences



counters went:

$[i_3]$	$[i_5]$
$\boxed{1}$	$\boxed{0}$
1	1
1	2
1	3
1	4
1	5
2	0
3	0
3	1
3	2
3	3
3	4
3	5

Comments on Branch and Bound

- it is an *exact* search method
  - guaranteed to find optimal solution
- may be much more efficient than exhaustive search
- in the worst case, it is no better
- efficiency depends on
  - the tightness of the lower bound
  - the quality of the initial tree

Rooted or Unrooted Trees for Parsimony?

- we described parsimony calculations in terms of rooted trees
- but we described the search procedures in terms of unrooted trees

- *unweighted parsimony*: minimum cost is independent of where root is located
- *weighted parsimony*: minimum cost is independent of root if substitution cost is a *metric*

triangle inequality

Summary

- Two techniques for searching tree space
  - Heuristic greedy search
  - Exact search via branch and bound
- Branch and bound
  - Removes parts of the search space from consideration via lower bounds