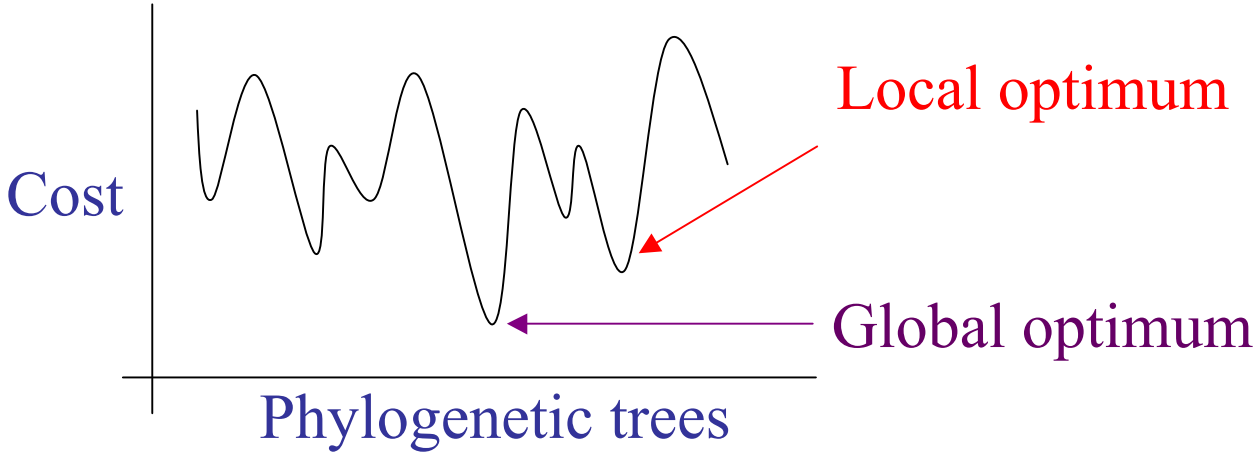
394C: Algorithms for Computational Biology

Tandy Warnow

Jan 25, 2012

Phylogenetic reconstruction methods

1. Hill-climbing heuristics for hard optimization criteria (Maximum Parsimony and Maximum Likelihood)



2. Polynomial time distance-based methods: UPGMA, Neighbor Joining, FastME, Weighbor, etc.

Performance criteria

- Running time.
- Space.
- Statistical performance issues (e.g., statistical consistency) with respect to a Markov model of evolution.
- "Topological accuracy" with respect to the underlying ***true tree.*** Typically studied in simulation.
- Accuracy with respect to a particular criterion (e.g. tree length or likelihood score), on real data.

How can we infer evolution?

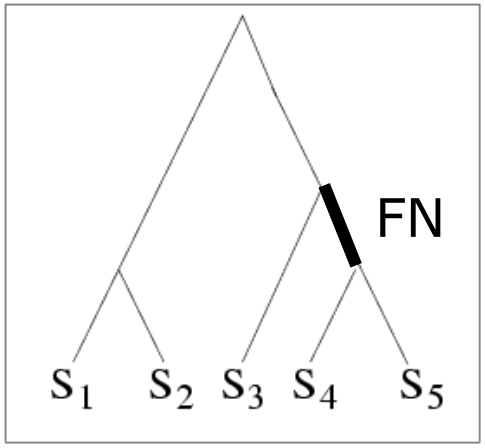
While there are more than two sequences, DO

- Find the “closest” pair of sequences and make them siblings
- Replace the pair by a single sequence

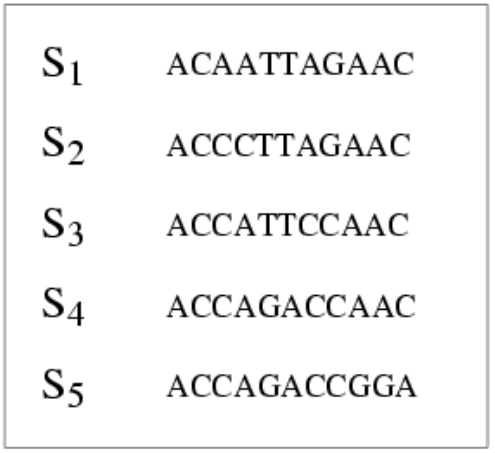
That was called “UPGMA”

- Advantages: UPGMA is polynomial time and works well under the “strong molecular clock” hypothesis.
- Disadvantages: UPGMA does not work well in simulations, perhaps because the molecular clock hypothesis does not generally apply.
- Other polynomial time methods, also distance-based, work better. One of the best of these is Neighbor Joining.

Quantifying Error



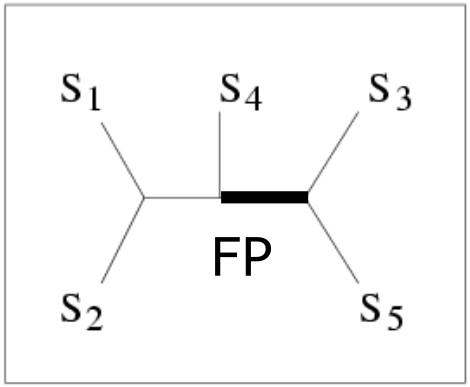
TRUE TREE



DNA SEQUENCES

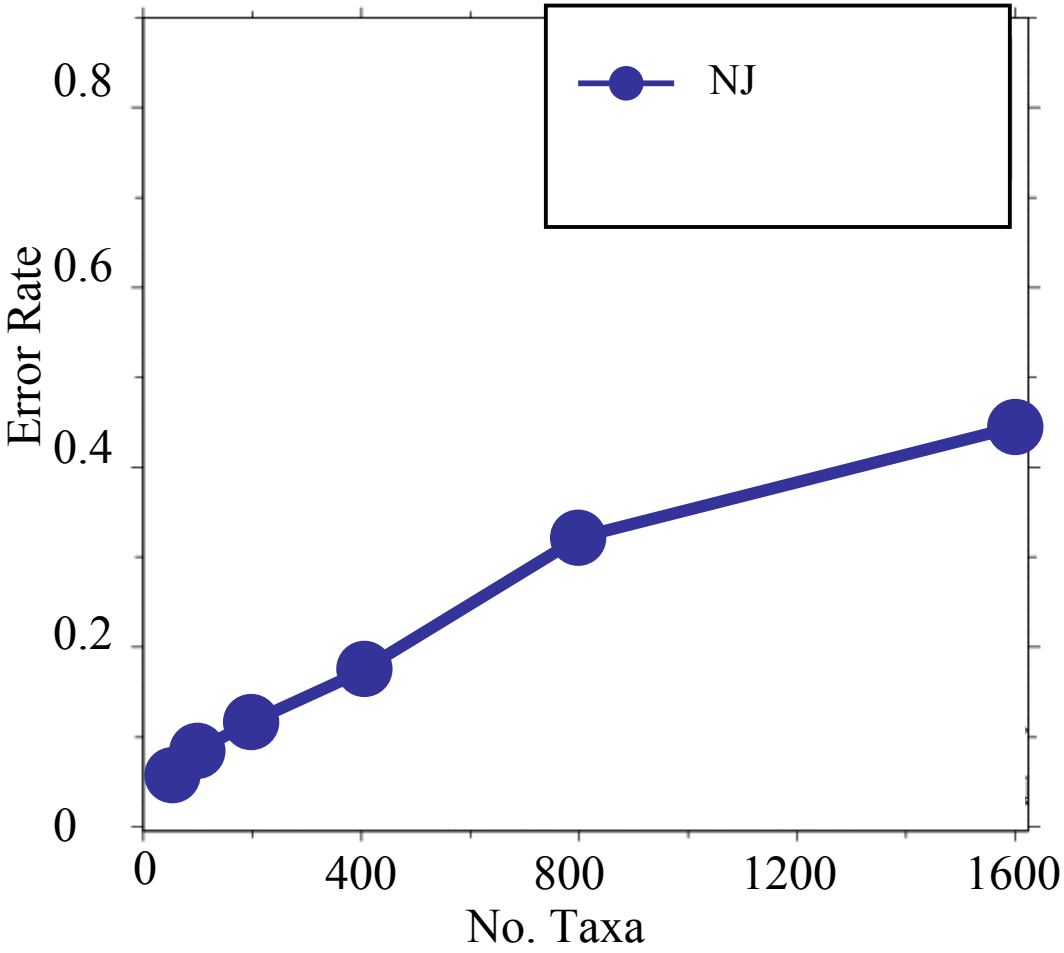
FN: false negative
(missing edge)
FP: false positive
(incorrect edge)

50% error rate



INFERRED TREE

Neighbor joining has poor performance on large diameter trees *[Nakhleh et al. ISMB 2001]*



Simulation study based upon fixed edge lengths, K2P model of evolution, sequence lengths fixed to 1000 nucleotides.

Error rates reflect proportion of incorrect edges in inferred trees.

- Other standard polynomial time methods don't improve substantially on NJ (and have the same problem with large diameter datasets).
- What about trying to "solve" maximum parsimony or maximum likelihood?

Maximum Parsimony

- **Input**: Set *S* of *n* aligned sequences of length k
- **Output**:
  - A phylogenetic tree T leaf-labeled by sequences in *S*
  - additional sequences of length *k* labeling the internal nodes of *T*

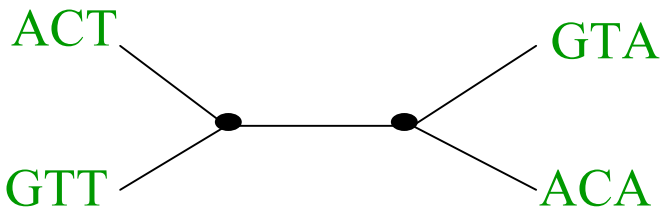
such that $\sum_{(i,j)\in E(T)} H(i,j)$

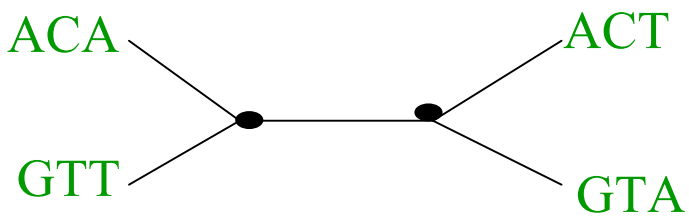
is minimized, where H(i,j) denotes the Hamming distance between sequences at nodes i and j

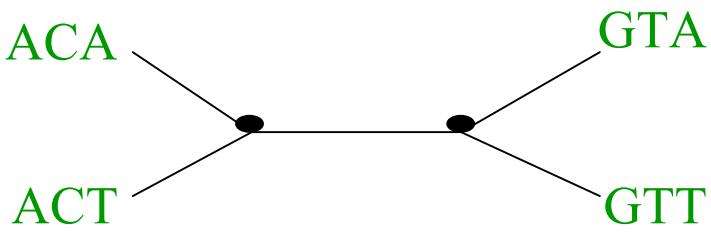
Maximum parsimony (example)

- **Input**: Four sequences
  - ACT
  - ACA
  - GTT
  - GTA
- **Question**: which of the three trees has the best MP scores?

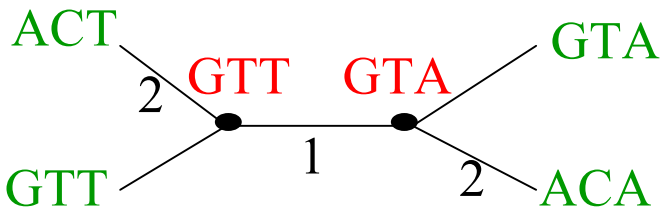
Maximum Parsimony



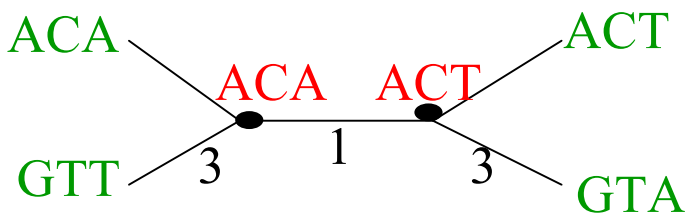




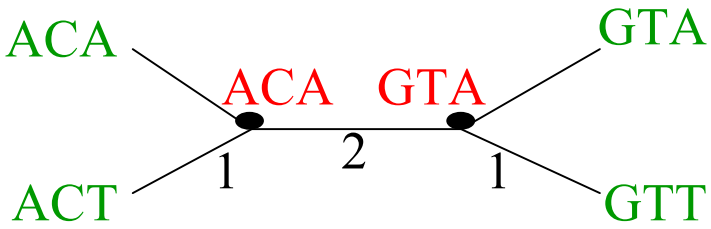
Maximum Parsimony



MP score = 5



MP score = 7

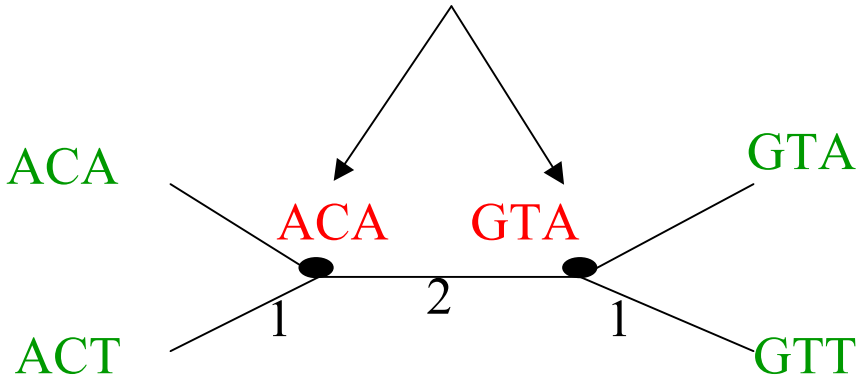


MP score = 4

Optimal MP tree

Maximum Parsimony: computational complexity

Optimal labeling can be computed in linear time O(nk)



MP score = 4

Finding the optimal MP tree is **NP-hard**

Dynamic Programming Algorithm for fixed tree MP

Single site solution for input tree T.

Root tree T at some internal node. Now, for every node v in T and every possible letter X, compute

Cost(v,X) := optimal cost of subtree of T rooted at v, given that we label v by X.

Base case: easy

General case?

DP algorithm (con't)

- Cost(v,X) =

  $\min_Y\{Cost(v_1,Y)+cost(X,Y)\}$ +
  $\min_Y\{Cost(v_2,Y)+cost(X,Y)\}$

where v_1 and v_2 are the children of v, and Y ranges over the possible "states", and cost(X,Y) is an arbitrary cost function.

DP algorithm (con't)

We compute Cost(v,X) for every node v and every state X, from the "bottom up".

The optimal cost is

$\min_X\{\text{Cost}(\text{root},X)\}$

Running time? Accuracy?

How to extend to many sites?

Special case when cost(X,Y)=Hamming(X,Y)?

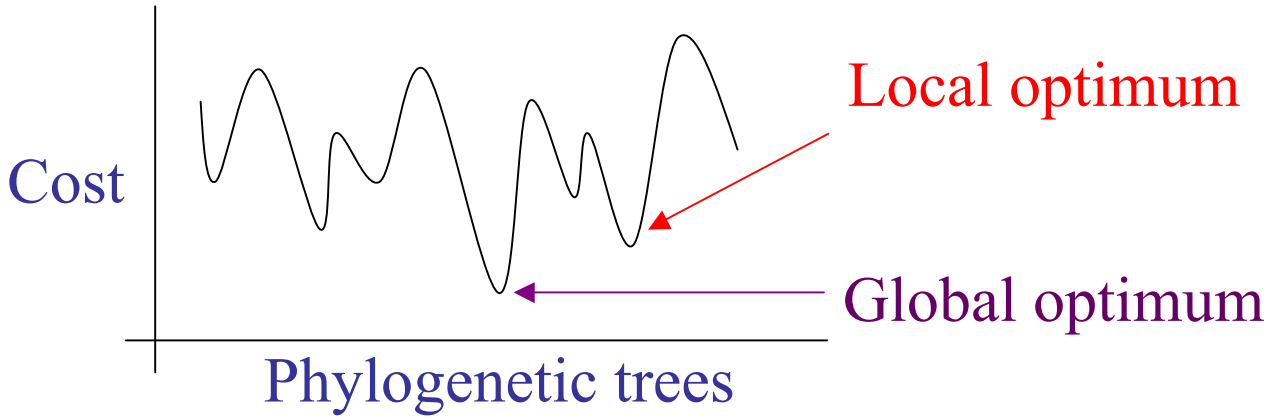
Solving NP-hard problems exactly is … unlikely

- Number of (unrooted) binary trees on *n* leaves is (2n-5)!!
- If each tree on **1000** taxa could be analyzed in **0.001** seconds, we would find the best tree in **2890 millennia**

#leaves	#trees
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025
20	2.2×10^{20}
100	4.5×10^{190}
1000	2.7×10^{2900}

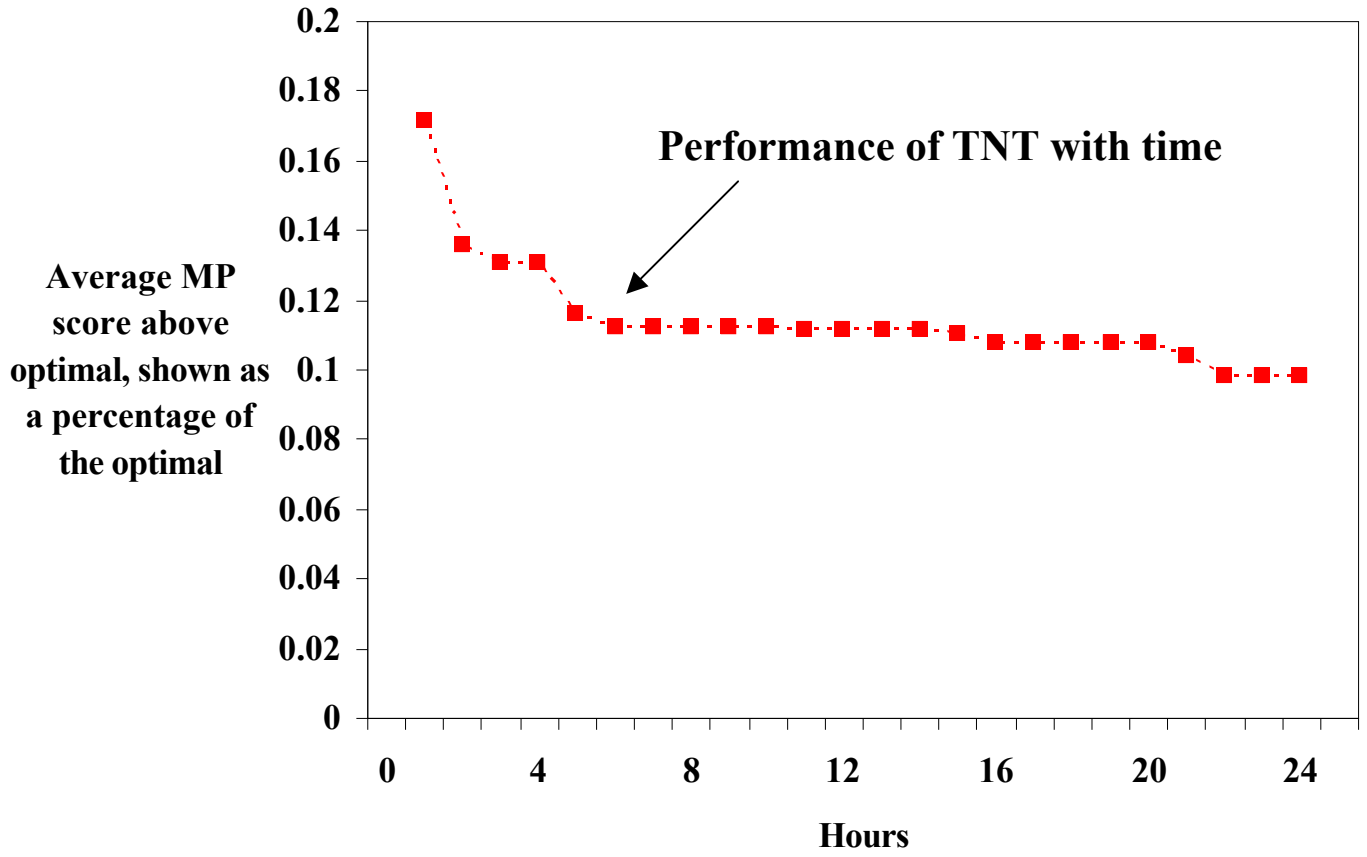
Approaches for "solving" MP/ML

1. Hill-climbing heuristics (which can get stuck in local optima)
2. Randomized algorithms for getting out of local optima
3. Approximation algorithms for MP (based upon Steiner Tree approximation algorithms).



Problems with current techniques for MP

Shown here is the performance of a heuristic maximum parsimony analysis on a real dataset of almost 14,000 sequences. ("Optimal" here means best score to date, using any method for any amount of time.) Acceptable error is below 0.01%.



Observations

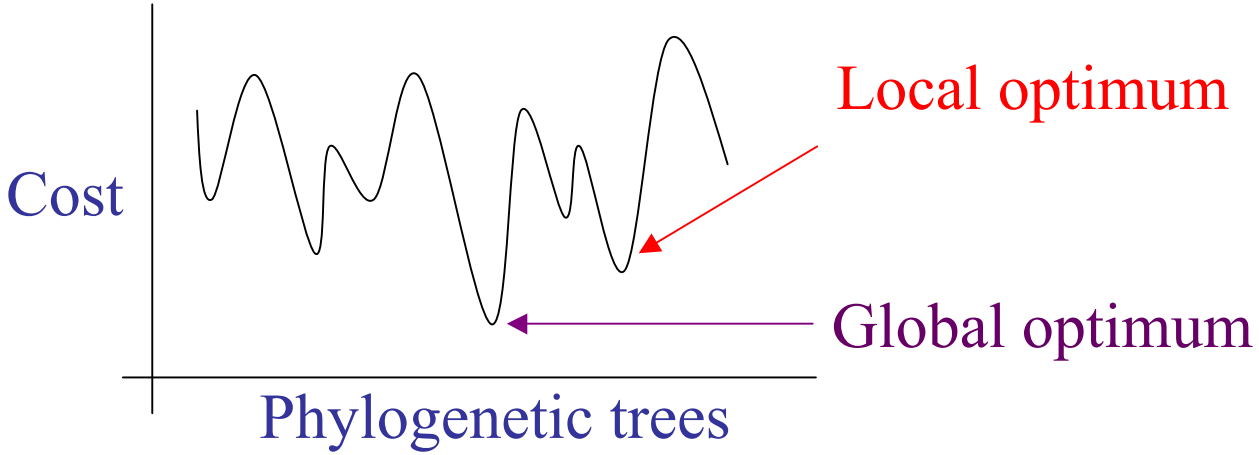
- The best MP heuristics cannot get acceptably good solutions within 24 hours on most of these large datasets.
- Datasets of these sizes may need months (or years) of further analysis to reach reasonable solutions.
- Apparent convergence can be misleading.

What happens after the analysis?

- The result of a phylogenetic analysis is often thousands (or tens of thousands) of equally good trees. What to do?
- Biologists use consensus methods, as well as other techniques, to try to infer what is likely to be the characteristics of the "true tree". Current techniques lack sufficient power.

Phylogenetic reconstruction methods

1. Hill-climbing heuristics for hard optimization criteria (Maximum Parsimony and Maximum Likelihood)



2. Polynomial time distance-based methods: UPGMA, Neighbor Joining, FastME, Weighbor, etc.

Supertree methods

- Input: collection of trees (generally unrooted) on subsets of the taxa
- Output: tree on the entire set of taxa

Basic questions:

- is the set of input trees compatible?
- can we find a tree satisfying a maximum number of input trees?

Triplet-based methods

- Triplet Compatibility: does a tree exist that satisfies all the input triplets? If so, find it. Polynomial time solvable!
- Aho, Sagiv, Szymanski, and Ullman algorithm (works on any input)

Quartet-based methods

- Quartet Compatibility: does there exist a tree compatible with all the input quartet trees? If so, find it. (NP-hard)
- Maximum Quartet Compatibility: find a tree satisfying a maximum number of quartet trees (NP-hard)
- Naïve Quartet Method solves Quartet Compatibility (must have a tree on every quartet)

Real data

- Cannot reliably obtain accurate rooted triplets
- Cannot reliably obtain accurate quartet trees
- All input trees will have some error
- “Supertree” methods need to be able to handle error in the input trees

Supertree methods

- Input: collection of trees (generally unrooted) on subsets of the taxa
- Output: tree on the entire set of taxa

Basic questions:

- is the set of input trees compatible?
- can we find a tree satisfying a maximum number of input trees?

Tree compatibility

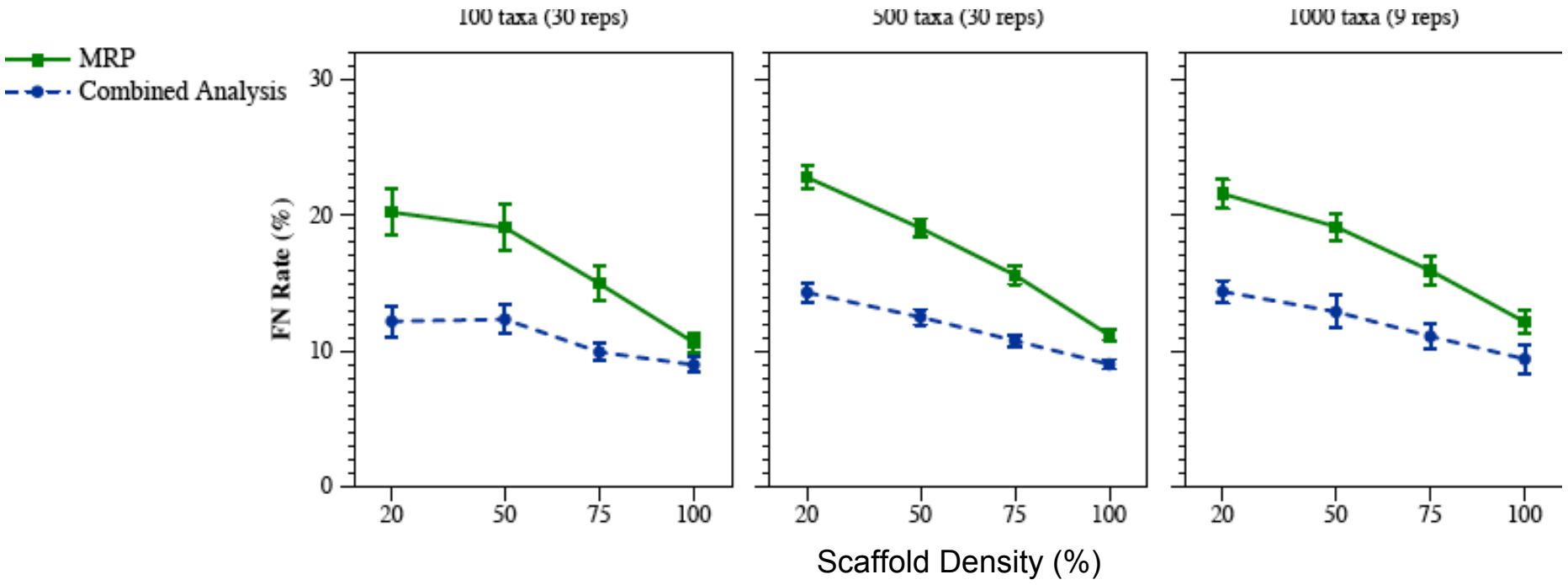
- Unrooted trees: NP-hard
- Rooted trees: Polynomial

But rooted trees are even harder to get exactly correct than unrooted trees!

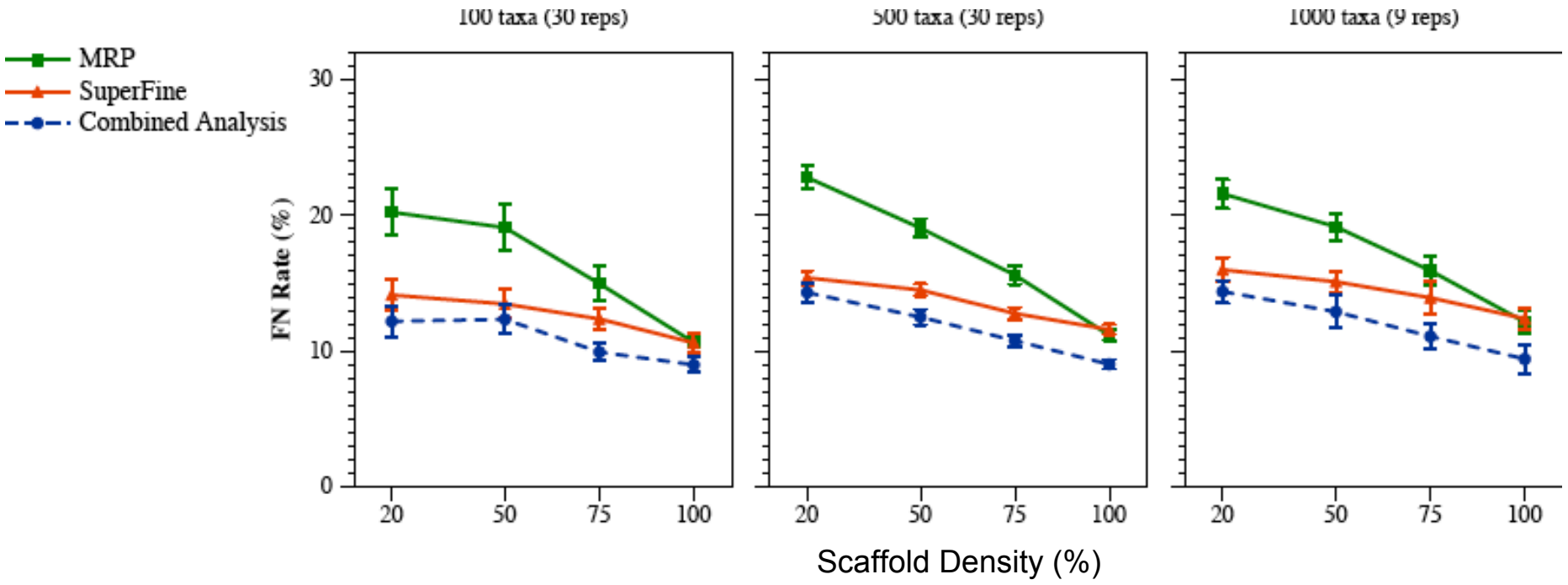
MRP

- Matrix Representation with Parsimony
- Encode each input source tree as a matrix with entries from {0,1,?}, and run maximum parsimony
- Solves “tree compatibility” exactly!

False Negative Rate

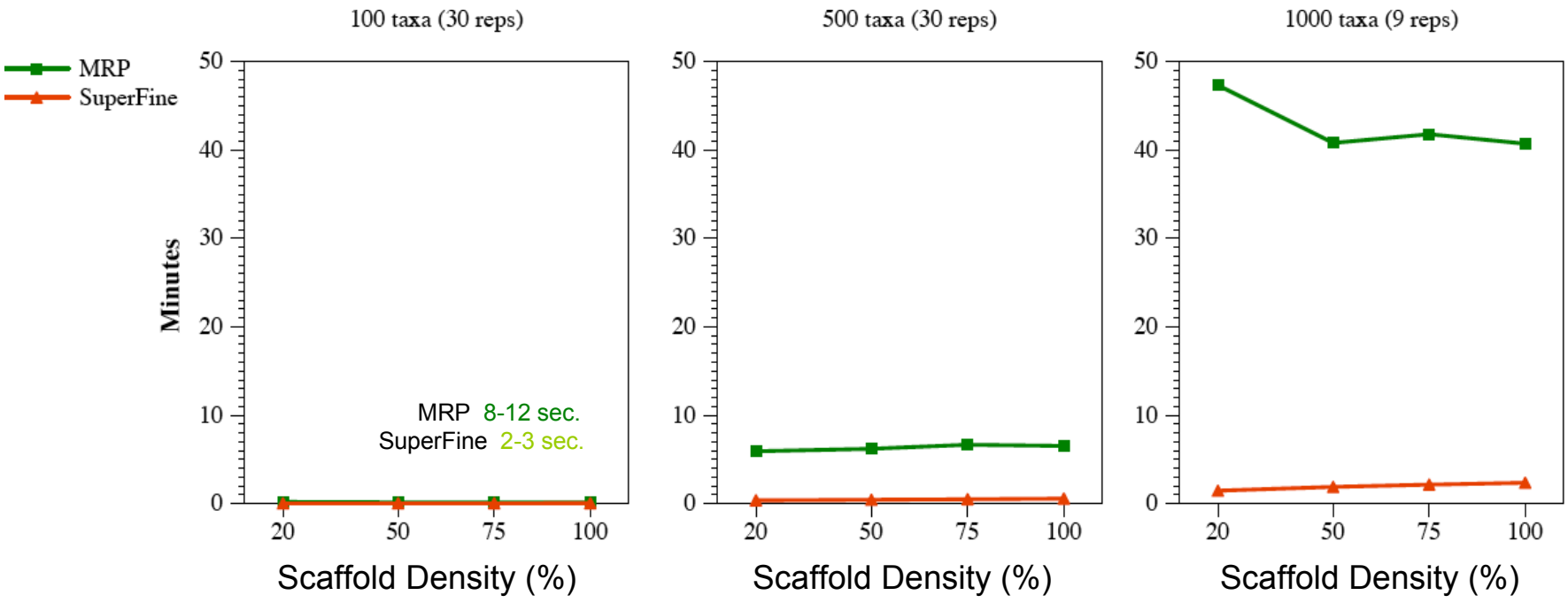


False Negative Rate



Running Time

SuperFine vs. MRP



Observations

- SuperFine is much more accurate than MRP, with comparable performance only when the scaffold density is 100%
- SuperFine is almost as accurate as CA-ML
- SuperFine is extremely fast

SuperFine

- Swenson et al. 2012, Systematic Biology. Paper #100 at

http://www.cs.utexas.edu/users/tandy/papers.html