CIPRES: Algorithmic Challenges

Bernard Moret
University of New Mexico

Phylogenetic Reconstruction: Areas of Research and Application Constraints

Overview

- Comparing phylogenetic trees for accuracy
- Better methods for current optimization criteria—accuracy, speed, and *flexibility*
- Relating optimization criteria and actual reconstruction accuracy; new criteria
- Synthesizing an answer from collections of trees (consensus methods)
- Constrained reconstruction
- Statistical assessment of results and other postanalysis tools
- Mathematical foundations for simulation studies; methodological innovations
- Database storage to support efficient queries on arbitrary pieces of complex objects
- Extensions of above to evolutionary networks

Comparing Trees

- Preferred measure is Robinson-Foulds: relative symmetric difference in the two sets of induced bipartitions.
- RF distance treats all bipartitions equally, but accuracy is more important on earlier branches.
- RF distance has a limited range—finer discriminations may be needed.
- No method for comparing inferred ancestral strings (genomes).

Optimization Criteria

- Maximum parsimony (MP): minimize sum of edit distances along the tree edges. Easy to evaluate in most settings.
- Maximum likelihood (ML): maximize probability that the data could have been produced by the chosen tree. Hard to evaluate in all settings.
- Outright optimization limited to $\sim$20 taxa for MP, fewer than $\sim$10 taxa for ML.
- Good heuristics can handle $\sim$500 taxa for MP and $\sim$100 taxa for ML (Bayesian approach).
- Tandy's and my group have extended good MP heuristics to $\sim$10,000 taxa through the Disk-Covering Method.

Optimization Criteria cont'd

- ML:
  Computing the ML score of a fixed tree under a fixed model: currently takes exponential time.
  Numerical problems arise for large dataset: almost all trees have infinitesimal likelihood.
- MP and ML:
  Lack of good bounding (both lower and upper bounds).
  Equivalent of Held-Karp for MP or ML?
- Heuristics are local search with a simple move: break the tree into two pieces and reconnect in a better way.
  Equivalent of Lin-Kernighan for MP or ML?
- Flexibility (and thus biological accuracy) lacking, esp. for MP.

Criteria vs. Goals

- ML and MP are surrogates for the truth: the “true" evolutionary tree almost certainly does not maximize parsimony nor likelihood (models will be too simple).
- Problem: truth cannot be verified, except in simulations or in parts of the tree.
- Opportunity: truth may be approached faster than optimality.
- Tandy’s and my group have results showing we should be able to stop MP heuristics after only 10–20% of current computation with no damage to accuracy.
- Can we devise new criteria that lead to the truth faster?

Consensus Methods

- MP and Bayesian ML methods return many equivalently good trees (can be in the 1,000's). Current codes use a consensus method to produce a single representative tree.
- Can consensus methods be improved? Only one of them (due to Tandy et al.) has good theoretical properties.
- Consensus methods produce one tree for the entire collection; almost certainly better is to produce some small number of trees, e.g., by combining clustering and consensus.

Constrained Reconstruction

- Many parts of the tree are well understood—accepted topologies, etc. Using such constraints is part of existing codes, but is rigid.
- Data other than molecular data are widely available, but lacks good models or are difficult to encode without biases. Can various types of constraints be extracted from them? How could such constraints (positive, negative, partial orders, etc.) be used?

Postanalysis Tools

- Biologists want to assess the reconstruction.
- Crucial tool is statistical validation of results: confidence that the true or optimal tree is within some error bracket around the result.
- Current method is simple “bootstrapping" (jackknifing) and measuring percent of trees that feature the same edge, yielding a bootstrap score for each edge. Primitive and computationally expensive (many runs!).

Simulation Studies

- Current models used to create datasets are serious oversimplifications. Interesting combinatorics to relate, e.g., tree generation with properties of resulting tree distribution (height, balance, etc.)
- Even with simple models, little to no theory exists to determine parameters of statistically useful experiments.

Database Storage

- Even with DNA strings, most queries require *linear search* through the database!
- Problem is that stored objects (strings, trees) are complex and queries need access to arbitrary subpieces of the string or tree.
- BLAST (due to Gene and others) uses clever heuristics to reduce the number of strings searched in most cases; perhaps similar algorithms can be devised for searching trees.
- Better solution: represent the trees so that arbitrary subtrees could easily be accessed on the basis of their properties. (Compare with dynamic data structures for graphs or geometry.)

Extension to Networks

- Two evolutionary events give rise to nontree structures: hybridization and horizontal gene transfer (HGT). In both cases, a node will have two parents rather than one.
- Result is a DAG with constraints (both hybridization and HGT require temporal coexistence, so parents of node cannot be arbitrary).
- Little to no work to date, although problem is clear: hybridization is common in plants and both HGT and hybridization are widespread in bacteria and archea (two of the three main branches of the Tree of Life).

Extension to Networks cont'd

- Many evolutionary events can masquerade as nontree evolution, most importantly duplication of genes and subsequent divergent evolution of duplicates. Distinguishing the two is *reconciliation*; some good theoretical work by Hallett and Lagergren.
- Tandy’s group and mine have devised a method for comparing two evolutionary networks that captures all desired properties and reduces to the standard RF measure when used on trees.
- We also have (very) preliminary results on reconstruction and on designing a simulator.

Conclusions

- My list is surely not exhaustive.
- But it is already long enough to keep many algorithm designers busy!
- Problems vary from foundational work in combinatorics and statistics, through basic algorithm and data structure design, to refining current heuristics.
- Software designers and pro team at SDSC stand ready to help with experimental work, algorithm engineering, parallel implementation, and code integration, as needed.