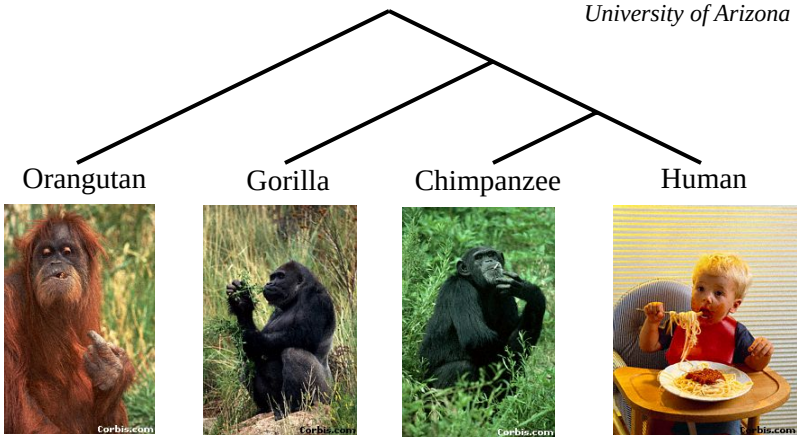
Challenges in constructing very large evolutionary trees

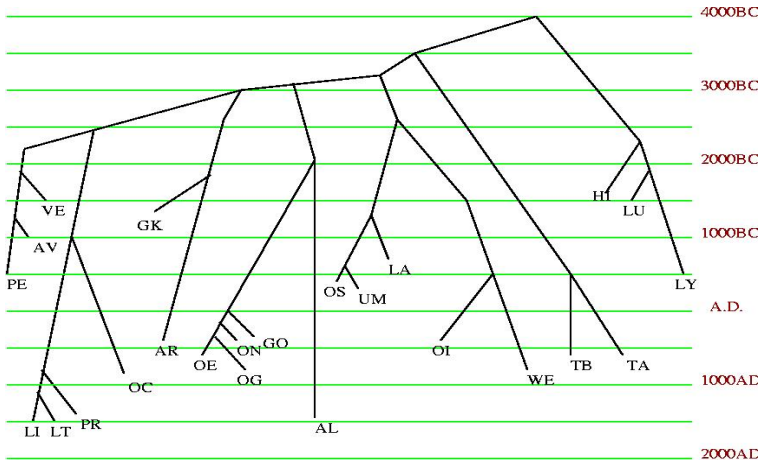
Tandy Warnow
Radcliffe Institute for Advanced Study
University of Texas at Austin

Phylogeny

From the Tree of the Life Website, University of Arizona



Ringe-Warnow Phylogenetic Tree of Indo-European



Major methods for phylogeny reconstruction

- Biology: Polynomial time methods (good enough for small datasets), and local search heuristics for NP-hard optimization problems
- Linguistics: exact algorithms for NP-hard optimization problems

Evolution informs about everything in biology

- Big genome sequencing projects just produce data -- so what?
- Evolutionary history relates all organisms and genes, and helps us understand and predict
  - interactions between genes (genetic networks)
  - drug design
  - predicting functions of genes
  - influenza vaccine development
  - origins and spread of disease
  - origins and migrations of humans

Main research foci

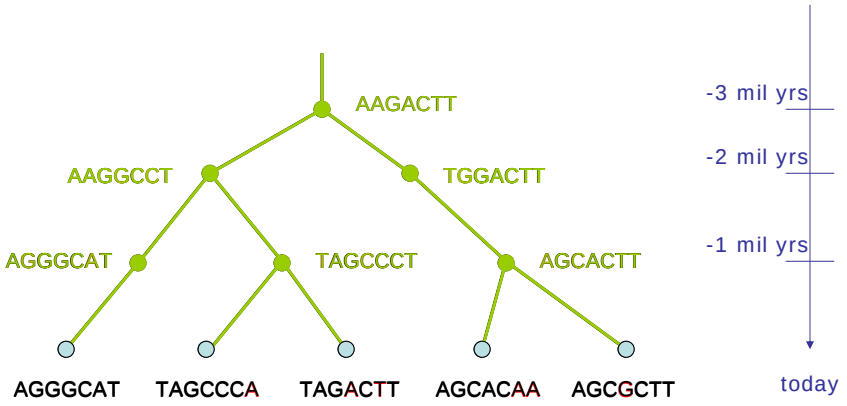
- Solving maximum parsimony and maximum likelihood more effectively
- “Fast converging methods”
- Gene order and content phylogeny
- Reticulate evolution
- Phylogenetic multiple sequence alignment

Gene Order/Content Phylogeny

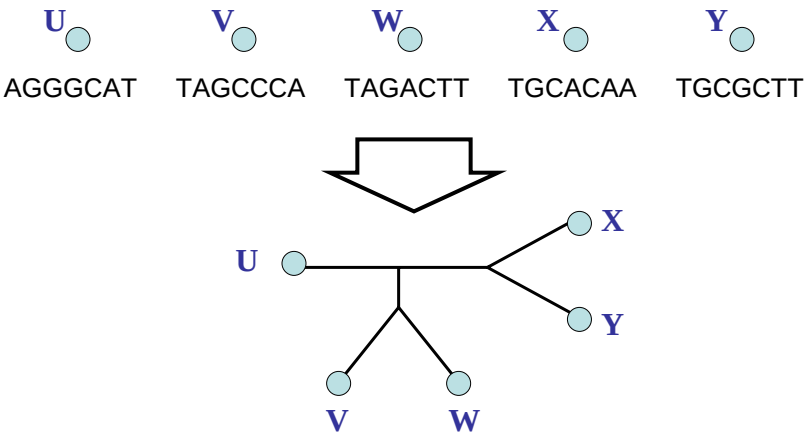
- Group leader: Bernard Moret
- Software: (1) simulating genome evolution on trees (2) GRAPPA: Genome Rearrangement Analysis using Parsimony and other Phylogenetic Algorithms
- Currently limited to equal content genomes
- Ongoing research: handling unequal gene content

Reticulate Evolution

DNA Sequence Evolution



Molecular Systematics



Basic challenges in molecular phylogenetics

- Most favored approaches attempt to solve hard optimization problems such as maximum parsimony and maximum likelihood - *can we design better methods?*
- DNA sequence evolution may be too "noisy" - *perhaps we need new types of data?*
- Many equally good solutions for a given dataset - *how can we figure out "truth"?*
- Not all evolution is tree-like - *how can we detect and infer reticulate evolution*?

Some of our projects

- Divide-and-conquer strategies for maximum parsimony and maximum likelihood
- Using "rare genomic changes" for deep evolution
- Consensus/clustering methods for sets of optimal trees
- Detection and reconstruction of reticulate evolution

(All projects are joint with biologists and computer scientists at various universities, and are part of the new ITR grant)

Coping with NP-hard problems

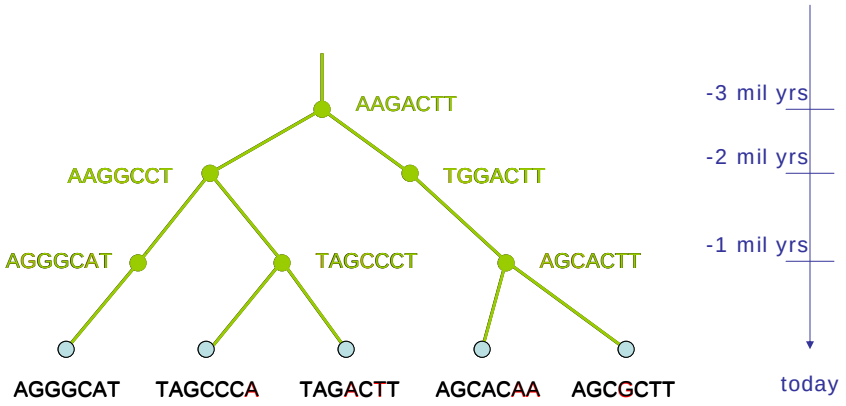
Since NP-hard problems may not be solvable in polynomial time, the options are:

- Solve the problem exactly (but use lots of time on some inputs)
- Use heuristics which may not solve the problem exactly (and which might be computationally expensive, anyway)

General comments for NP-hard optimization problems

- Getting exact solutions may not be possible for some problems on some inputs, without spending a great deal of time.
- You may not know when you have an optimal solution, if you use a heuristic.
- Sometimes exact solutions may not be necessary, and approximate solutions may suffice. (But this may not be true for biology.)

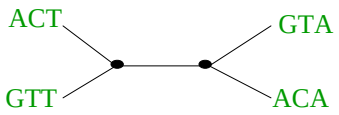
DNA Sequence Evolution

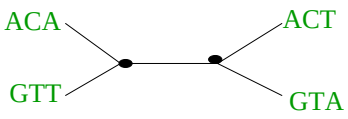


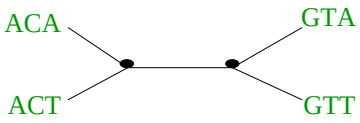
Major phylogeny reconstruction methods

- In biology: mostly hill-climbing heuristics that attempt to solve NP-hard optimization problems (maximum parsimony or maximum likelihood)
- In historical linguistics: much less is established, but an exact solution to an NP-hard problem looks very promising.

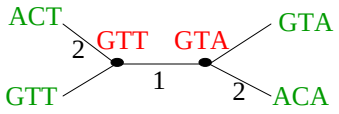
Maximum Parsimony



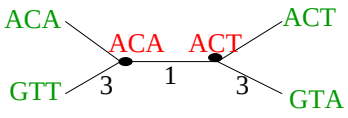




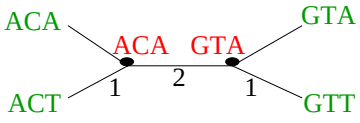
Maximum Parsimony



MP score = 5



MP score = 7

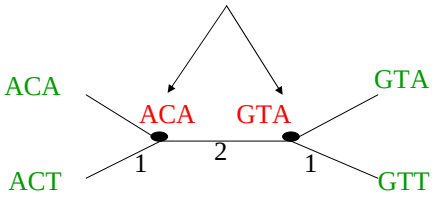


MP score = 4

Optimal MP tree

Maximum Parsimony: computational complexity

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



MP score = 4

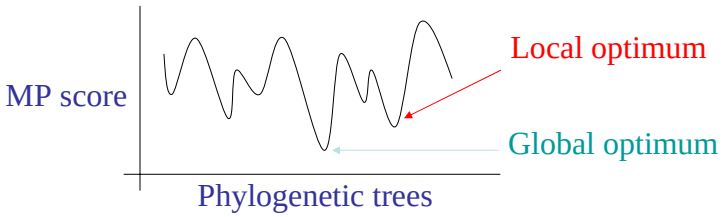
Finding the optimal MP tree is **NP-hard**

Maximum Parsimony

- Given a set S of strings of the same length over a fixed alphabet, find a tree T leaf-labelled by S and with all internal nodes labelled by strings of the same length over the same alphabet which minimizes the sum of the edge lengths.
- Motivation: seeks to minimize the total number of point mutations needed to explain the data
- NP-hard

Solving MP (maximum parsimony) and ML (maximum likelihood)

- Why are MP and ML hard? The search space is huge -- there are (2n-5)!! trees, it is easy to get stuck in local optima, and there can be many optimal trees.
- Why try to solve MP or ML? Our experimental studies show that polynomial time algorithms don't do as well as MP or ML when trees are big and have high rates of evolution.
- Why solve MP and ML well? Because trees can change in biologically significant ways with small changes in objective criterion. (Open problem!)



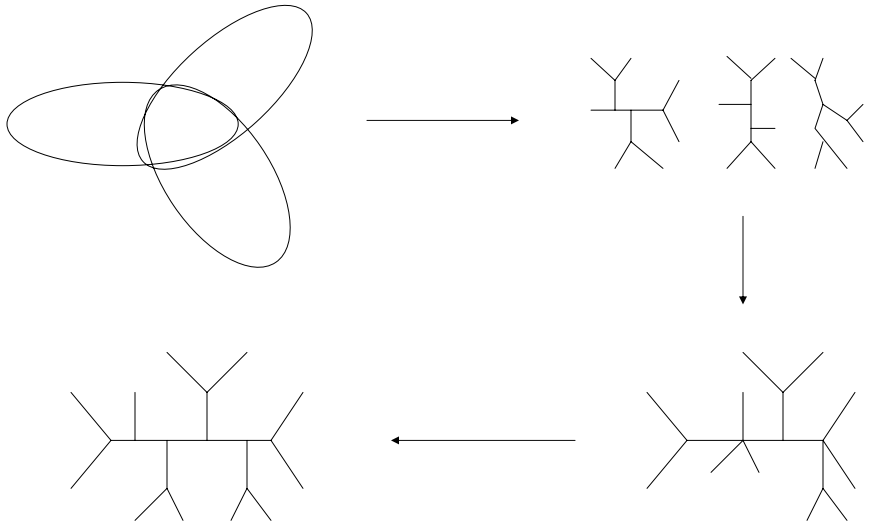
Using divide-and-conquer for MP and ML

- Conjecture: better (more accurate) solutions will be found in less time, if we analyze a small number of smaller subsets and then combine solutions
- Need:
  - 1. techniques for decomposing datasets,
  - 2. base methods for subproblems, and
  - 3. techniques for combining subtrees

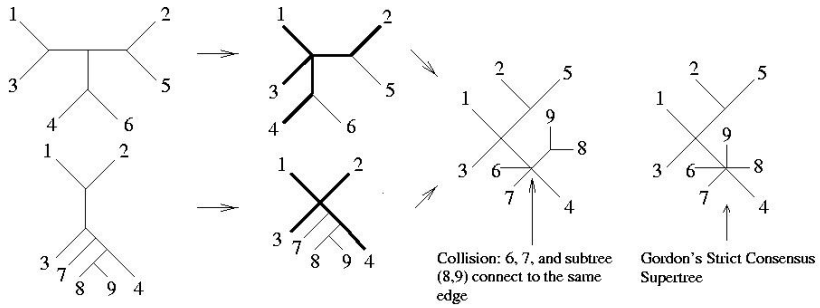
Comparison between TBR and the Ratchet

- Quite dramatic differences -- the Ratchet finds better trees than the best ways of running TBR branch-swapping, on all our datasets
- Even the Ratchet can take too long on some datasets!Ochoterena dataset: 834 DNA sequences

The DCM3 technique for speeding up MP/ML searches



Strict Consensus Merger (SCM)



DCM3-boosting a base method

1. Decompose the dataset into smaller, overlapping subsets, using DCM3
2. Construct phylogenetic trees on the subsets using a base method
3. Merge the subtrees into a single tree using the Strict Consensus Merger
4. Use PAUP* constrained search to refine the resultant tree

What we found

- I-DCM3-TBR is much faster than TBR on all the datasets we examined
- I-DCM3-Ratchet is better than the Ratchet, but by less (depends on dataset)
- I-DCM3-ML improves upon ML using PAUP* ML searches (by a huge amount)

What we found

- DCM3-TBR is much faster than TBR on all the datasets we examined
- DCM3-Ratchet is better than the Ratchet, but by less (depends on dataset)
- DCM3-ML improves upon ML using PAUP* ML searches (by a huge amount)

New technique: Iterative DCM3

Repeat:

1. Apply base method for a specified number of iterations.
2. Obtain a DCM3-decomposition based upon the current best tree (the "guide tree").
3. Apply base method to subproblems, and merge subtrees using the strict consensus merger.
4. Refine the tree.

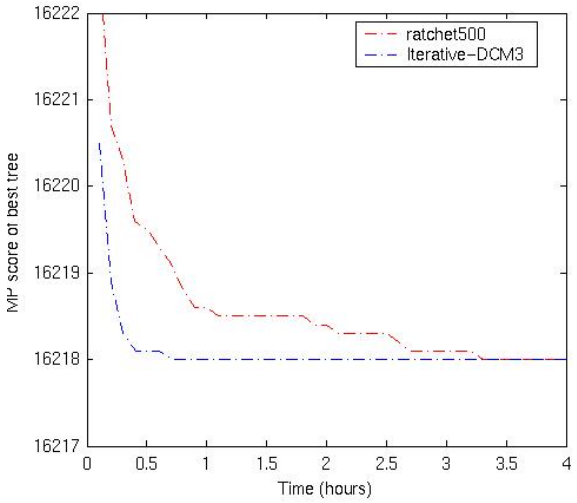
Variants we have examined:

I-DCM3(TBR) and I-DCM3(Ratchet).

Popular heuristics

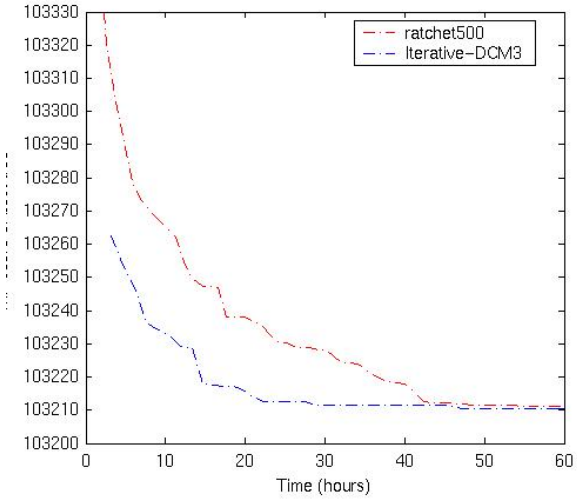
- **PAUP*4.0 hill-climbing heuristics:**
  - **Phase 1: do greedy insertions, with limited TBR,** to get good starting trees
  - **Phase 2**: **do TBR branch swapping on the best trees** obtained in phase I.
- **Ratchet:**
  - Do standard TBR hillclimbing until stuck in local optima.
  - Then **reweight characters** and do TBR hill-climbing to get out of local optima.
  - Go back to original character set, and repeat.

rbcL500 dataset: 500 DNA sequences



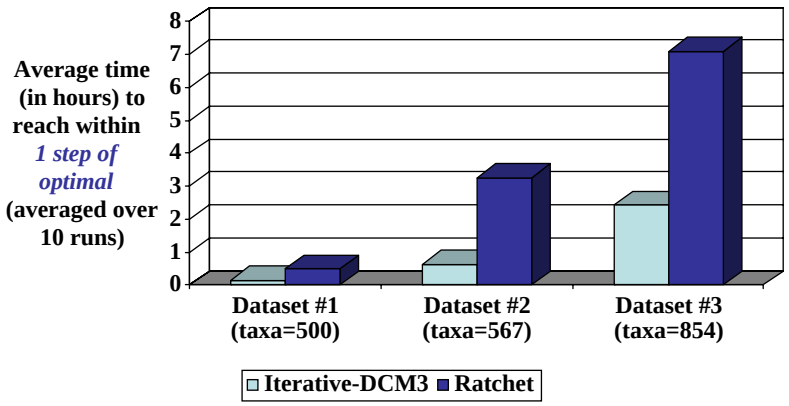
All 10 runs of Iterative-DCM3 find trees with current best score within75 minutes, whereas Ratchet takes at least 3 hours

Gutell dataset: 854 rRNA sequences

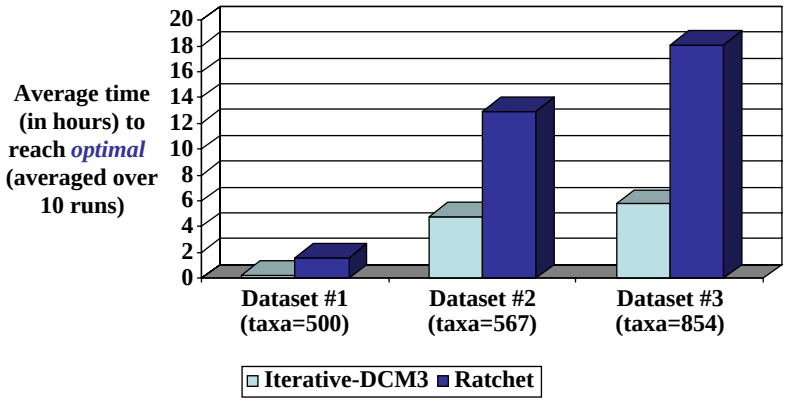


Iterative-DCM3 trials find trees of MP score 103210 in 30 hours, whereas ratchet500 trials take 45 hours to find trees of same score

Iterative-DCM3 vs Ratchet



Iterative-DCM3 vs Ratchet



Conclusions

- I-DCM3 finds trees with MP scores at least as good as Ratchet at every point in time (within first few hours, I-DCM3 is always better)
- On all datasets I-DCM3 finds good MP trees *very quickly*
- Improvements over TBR-based analyses even better

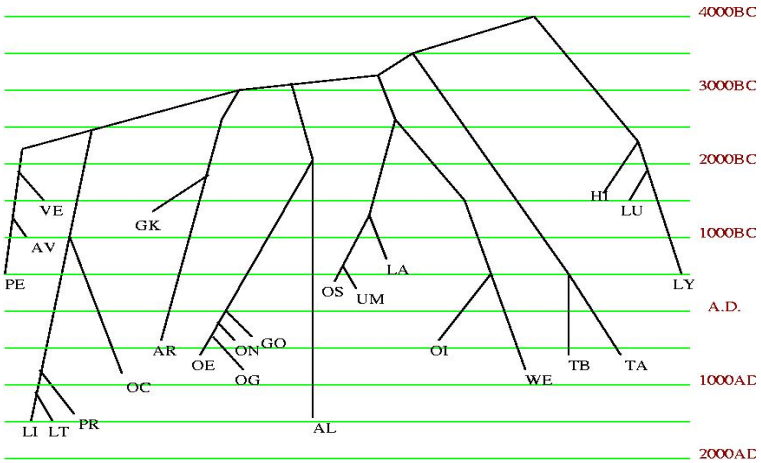
Ongoing research projects

- ML/MP: Getting better (faster and more accurate) divide-and-conquer strategies, and determining just how well we really need to analyze biomolecular datasets
- Analyzing whole genomes using gene order and content data
- Reticulate evolution inference

Comments

- Developing heuristics with good performance takes mathematical insights, but may not involve proofs. Even so, it's really important.
- Extracting information from the set of optimal (and near-optimal) solutions is a major open problem.
- Other types of data (gene orders, morphology) present novel challenges.
- Reticulate evolution detection and reconstruction is a major open problem.

Ringe-Warnow Phylogenetic Tree of Indo-European



Historical Linguistic Data

- A character is a function that maps a set of languages, *L*, to a set of states.
- Three kinds of characters:
  - Phonological (sound changes)
  - Lexical (meanings based on a wordlist)
  - Morphological (grammatical features)

Cognate Classes

- Two words w_1 and w_2 are in the same cognate class, if they evolved from the same word through sound changes.
- French “champ” and Italian “champo” are both descendants of Latin “campus”; thus the two words belong to the same cognate class.
- Spanish “mucho” and English “much” are not in the same cognate class.

Phylogenies of Languages

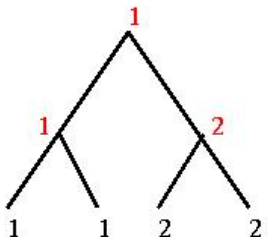
- Languages evolve over time, just as biological species do (geographic and other separations induce changes that over time make different dialects incomprehensible -- and new languages appear)
- The result can be modelled as a rooted tree
- The interesting thing is that many characteristics of languages evolve without back mutation or parallel evolution -- so a "perfect phylogeny" is possible!

Perfect Phylogeny

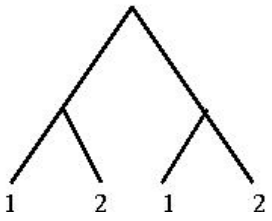
- A phylogeny T for a set S of taxa is a perfect phylogeny if each state of each character occupies a subtree (no character has back-mutations or parallel evolution)

"Homoplasy-Free" Evolution (perfect phylogenies)

YES



NO



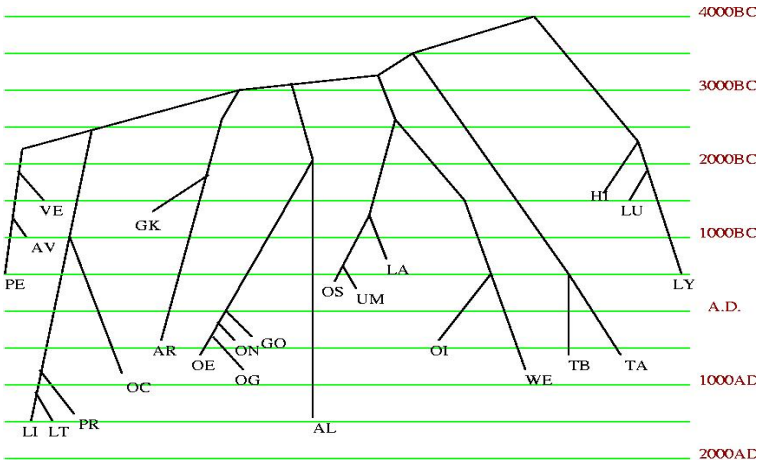
The Perfect Phylogeny Problem

- Given a set S of taxa (species, languages, etc.) determine if a perfect phylogeny T exists for S.
- The problem of determining whether a perfect phylogeny exists is NP-hard (McMorris *et al.* 1994, Steel 1991).

The Indo-European (IE) Dataset

- 24 languages
- 22 phonological characters, 15 morphological characters, and 333 lexical characters
- Total number of working characters is 390 (multiple character coding, and parallel development)
- A phylogenetic tree *T* on the IE dataset (Ringe, Taylor and Warnow)
- *T* is compatible with all but 22 characters: 16 (18) monomorphic and 6 polymorphic
- *Resolves most of the significant controversies in Indo-European evolution; shows however that Germanic is a problem (not treelike)*

Phylogenetic Tree of the IE Dataset



Acknowledgements

- Funding: NSF, the David and Lucile Packard Foundation, and the Radcliffe Institute for Advanced Study

- Collaborators: Bernard Moret and Tiffani Williams (UNM CS), Donald Ringe (Penn Linguistics

- Students: Usman Roshan and Luay Nakhleh (UT-Austin)

Phylolab, U. Texas

Please visit us at
http://www.cs.utexas.edu/users/phylo/