High-Performance Algorithm Engineering for Computational Phylogenetics

Bernard M.E. Moret

moret@cs.unm.edu

Department of Computer Science

University of New Mexico

Albuquerque, NM 87131

Collaborators and Sponsors

National Science Foundation ITR 00-81404

NSF/Alliance supercomputer access

Equipment grant from Sun Microsystems Inc. (14-processor E4500)

In collaboration with

- David A. Bader, Electrical and Computer Engineering, University of New Mexico
- Tandy Warnow, Computer Sciences, University of Texas at Austin

Overview

- Algorithm Engineering
- High-Performance Algorithm Engineering
- Phylogenies
- Computational Phylogenetics
- An Example: Gene-Order Phylogenies
  - Breakpoint phylogeny
  - Inversion and other genomic distance measures
  - GRAPPA: a high-performance software tool for reconstructing phylogenies from gene-order data

Algorithm Engineering

- The process required to transform a pencil-and-paper algorithm into an efficient, robust, and easily usable implementation.

Algorithm Engineering

- The process required to transform a pencil-and-paper algorithm into an efficient, robust, and easily usable implementation.
- Tools come from software engineering, algorithm design, computer architecture, etc.

Algorithm Engineering

- The process required to transform a pencil-and-paper algorithm into an efficient, robust, and easily usable implementation.
- Tools come from software engineering, algorithm design, computer architecture, etc.
- Main focus is experimentation.

Algorithm Engineering (cont'd)

- Yearly workshop in Europe: *Workshop on Algorithm Engineering (WAE)*, starting in 1997.
- Yearly workshop in the US: *Workshop on Algorithm Engineering and Experiments (ALENEX)*, started in 1999.
- Main journal is the *ACM Journal of Experimental Algorithmics (JEA)*.

High-Performance Algo. Engineering

Running time and quality of solutions as the paramount goal.

Includes parallelism (both shared-memory and message-passing), but most impact comes from refining the serial part of the code.

Cache-aware programming is a key to performance with high-performance machines, which have deep memory hierarchies.

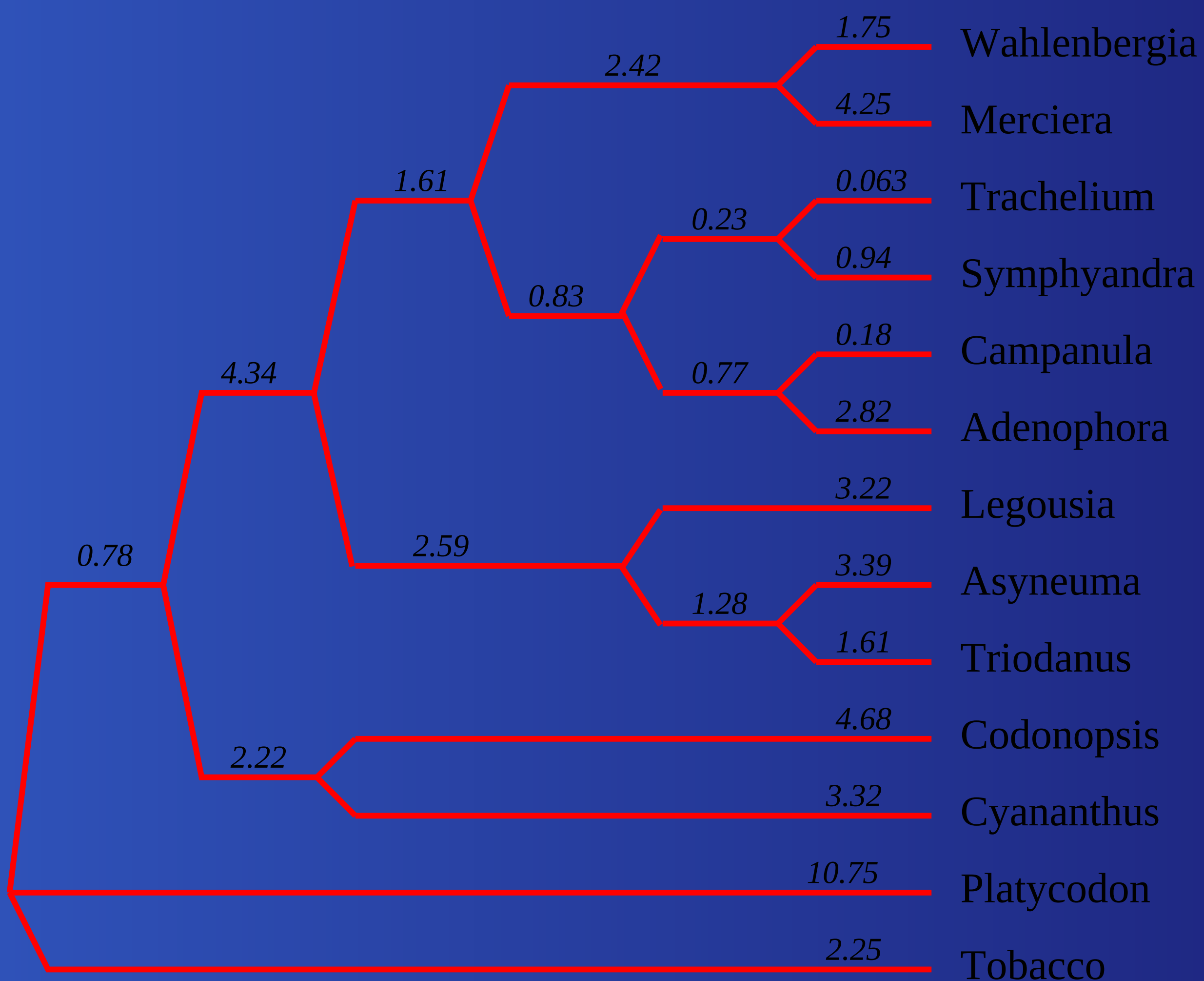
Phylogenies

A phylogeny is a reconstruction of the evolutionary history of a collection of organisms; it usually takes the form of a tree.

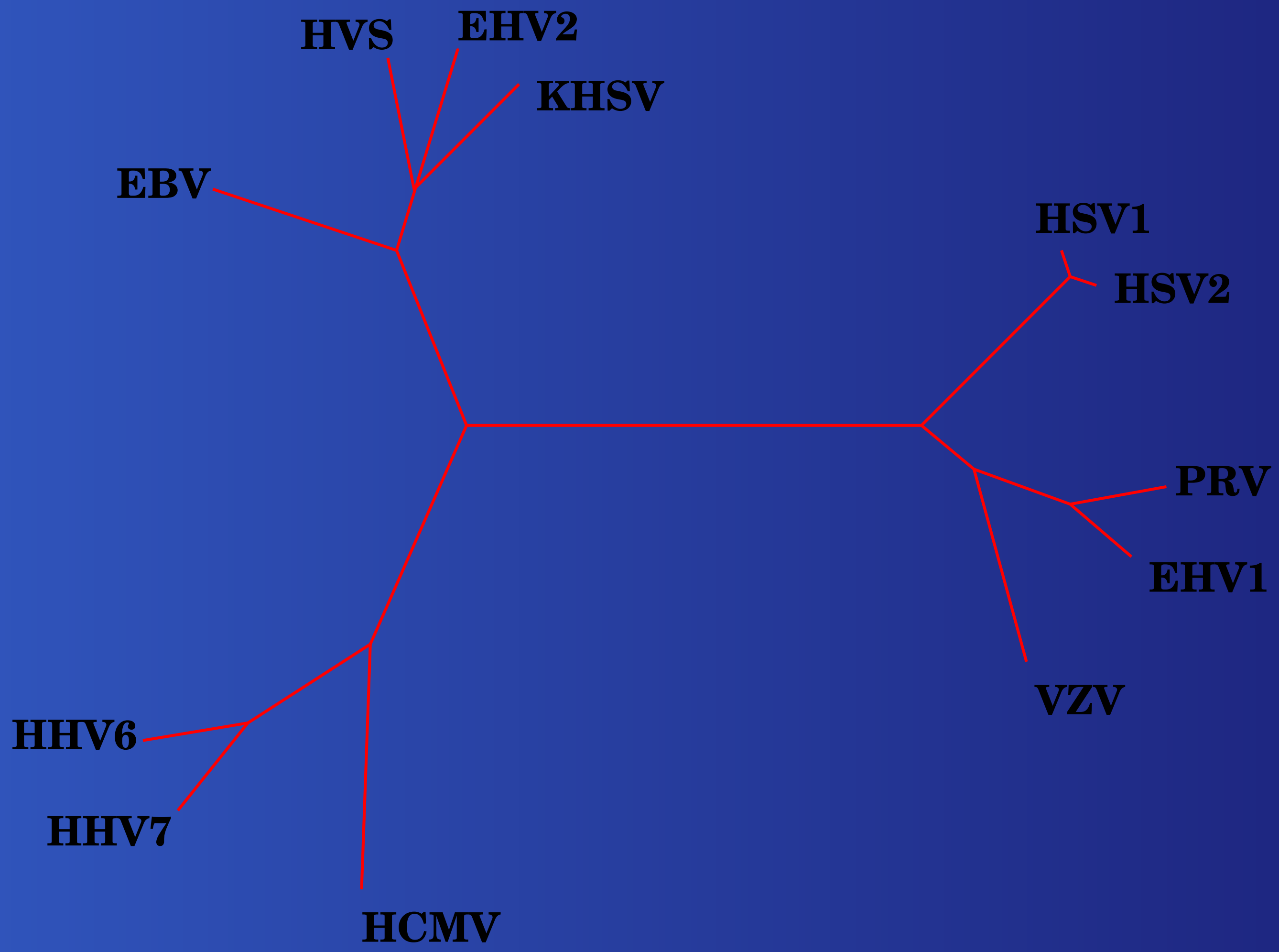
Modern organisms are placed at the leaves and ancestral organisms occupy internal nodes.

The edges of the tree denote evolutionary relationships.

12 Species of *Campanulaceae*



Herpes Viruses that Affect Humans



Phylogenies (cont'd)

Reconstructing phylogenies is a major component of modern research programs in many areas of biology and medicine:

- *pharmaceutical research for drug discovery*
- *understanding rapidly mutating viruses (such as HIV)*
- *designing genetically enhanced organisms*
- *explaining and predicting gene expression*
- *most centrally, understanding genomic evolution*

Computational Phylogenetics

- Is extremely computation-intensive.

  some pharmaceutical companies have been analyzing the same dataset for over two years on farms of > 200 processors.

Computational Phylogenetics

- Is extremely computation-intensive.

  some pharmaceutical companies have been analyzing the same dataset for over two years on farms of > 200 processors.
- Sequence data (RNA, DNA, aminoacid, and protein) has been used for over 20 years and is well understood, but methods do not scale up.

Computational Phylogenetics

- Is extremely computation-intensive.

  some pharmaceutical companies have been analyzing the same dataset for over two years on farms of > 200 processors.
- Sequence data (RNA, DNA, aminoacid, and protein) has been used for over 20 years and is well understood, but methods do not scale up.
- Genomic data (gene order and content of whole genomes) provides information on deep relationships, but is *much harder* to analyze than sequence data.

Computational Phylogenetics

- Is extremely computation-intensive.

  some pharmaceutical companies have been analyzing the same dataset for over two years on farms of > 200 processors.
- Sequence data (RNA, DNA, aminoacid, and protein) has been used for over 20 years and is well understood, but methods do not scale up.
- Genomic data (gene order and content of whole genomes) provides information on deep relationships, but is *much harder* to analyze than sequence data.
- Using mixed data remains uncharted territory.

Computational Phylogenetics (cont'd)

Methods developed by algorithm designers are rarely used by biologists: optimization criteria are chosen for algorithmic reasons more than biological ones.

Methods used by biologists are typically *ad hoc* and offer no guarantees: parameters are set with little understanding of their effects on efficiency or quality.

Getting the two groups to work together requires an atmosphere of mutual respect for each group's research goals and methodologies.

An Example: the Bluebell Family

Jansen's group at UT Austin provided full gene sequences for the chloroplasts of 12 species of Campanulaceae (Bluebells), plus tobacco.

A chloroplast is a semi-independent organism that lives within plant cells and allows them to photosynthesize.

Chloroplasts have a single chromosome with about 120 genes.

Optimization target: reconstruct the phylogeny with the least total number of genomic changes.

An application of Occam's razor; biologists call this the principle of parsimony.

The Bluebell Family (cont'd)

We reimplemented a tool due to D. Sankoff and M. Blanchette using algorithm engineering.

Results: *a speed-up by* **three to four orders of magnitude** *in the serial part of the code and a total speed-up by over* **one million** *when run on the 512-processor* Los Lobos *supercluster at UNM.*

Reasons: *cache-awareness, detailed code optimization, better combinatorial optimization, better bounding, and parallelization.*

Breakpoint Analysis: An Overview

An iterative improvement procedure:

Initially label all internal nodes with gene orders

Repeat

For each internal node v, with neighbors A, B, and C, do

Solve the *MPB* on A, B, C to yield label m

If relabelling v with m improves the score of T, then do it

until no internal node can be relabelled

MPB: Median Problem for Breakpoints

Given 3 gene orders, represented as 3 signed permutations π_1, π_2, and π_3, find a 4th permutation π_m that minimizes the sum of the distances

$$d(\pi_1, \pi_m) + d(\pi_1, \pi_m) + d(\pi_1, \pi_m)$$

where each distance is the number of *breakpoints*, i.e., the number of adjacencies present in one permutation but not in the other.

MPB: an example

Let the (circular) permutations be

1	-2	4	3
1	2	-3	-4
2	-3	-4	-1

A possible median is -1 2 -3 -4, with cost 5

d((1 -2 4 3) , (-1 2 -3 -4)) = 3

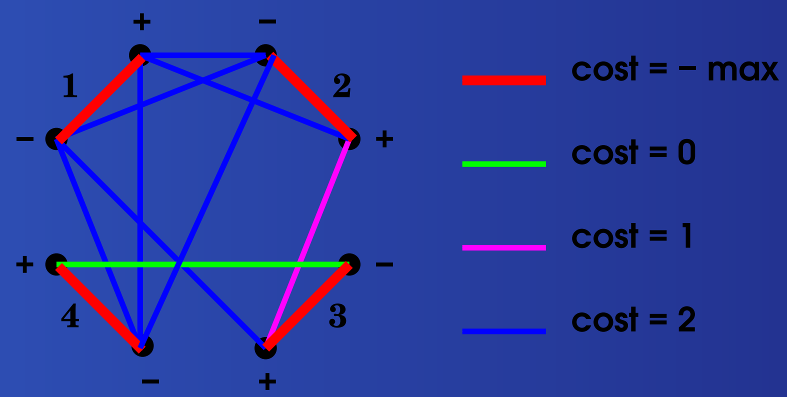
d((1 2 -3 -4) , (-1 2 -3 -4)) = 2

d((2 -3 -4 -1) , (-1 2 -3 -4)) = 0

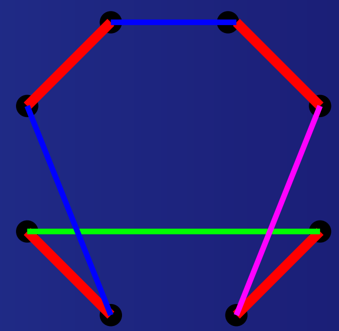
MPB (cont’d)

Sankoff showed to to convert this problem to the Travelling Salesperson Problem.

+1 −2 +4 +3
+1 +2 −3 −4
+2 −3 −4 −1



edges not shown have cost = 3



an optimal solution corresponding to genome +1 +2 −3 −4

Adjacency A B becomes an edge from A to −B

The cost of an edge A −B is the number of genomes that do NOT have the adjacency A B

Re-Engineering: Coding Aspects

- Memory Use

  GRAPPA uses static allocation and has a working set size of 500KB, compared to 12MB for BPAnalysis

Re-Engineering: Coding Aspects

- Memory Use

  GRAPPA uses static allocation and has a working set size of 500KB, compared to 12MB for BPAnalysis

- Cache Awareness

  GRAPPA minimizes pointer dereferencing, has hand-unrolled loops, and re-uses allocated storage

Re-Engineering: Coding Aspects

- Memory Use

  GRAPPA uses static allocation and has a working set size of 500KB, compared to 12MB for BPAnalysis
- Cache Awareness

  GRAPPA minimizes pointer dereferencing, has hand-unrolled loops, and re-uses allocated storage
- Profiling

  Identifies bottlenecks to balance the computation

Re-Engineering: Algorithmic Aspects

- Taking Advantage of Special Structures
  The TSP has only 2 non-trivial edges (cost 1 and 2)

Re-Engineering: Algorithmic Aspects

- Taking Advantage of Special Structures

  The TSP has only 2 non-trivial edges (cost 1 and 2)

- Using all Available Information

  TSP bounds can use full local legality test

Re-Engineering: Algorithmic Aspects

- Taking Advantage of Special Structures

  The TSP has only 2 non-trivial edges (cost 1 and 2)
- Using all Available Information

  TSP bounds can use full local legality test
- Lower Bound for Each Tree

  Triangle inequality implies that a tour of the leaves is at most twice the cost of any tree

Re-Engineering: Parallel Aspects

- Efficient Tree Generation

  Avoid multi-precision arithmetic, allow generation from any count with variable gap – provides parallel generation and also sampling of search space

Re-Engineering: Parallel Aspects

- Efficient Tree Generation

  Avoid multi-precision arithmetic, allow generation from any count with variable gap – provides parallel generation and also sampling of search space

- (current) Portable MPI Implementation

  Exploits “embarrassing” parallelism (each processor handles a fraction of the trees)

Re-Engineering: Parallel Aspects

- Efficient Tree Generation

  Avoid multi-precision arithmetic, allow generation from any count with variable gap – provides parallel generation and also sampling of search space
- (current) Portable MPI Implementation

  Exploits “embarrassing” parallelism (each processor handles a fraction of the trees)
- (future) Hybrid Mode Implementation

  Exploits shared-memory parallelism at each node for combinatorial optimization

Algorithmic Benefits

- In re-engineering an algorithm, one comes to understand it better than anyone else—and so one often finds algorithmic improvements.

Algorithmic Benefits

- In re-engineering an algorithm, one comes to understand it better than anyone else—and so one often finds algorithmic improvements.
- We developed the first true linear-time algorithm for computing the inversion distance between two signed permutations.

Algorithmic Benefits

- In re-engineering an algorithm, one comes to understand it better than anyone else—and so one often finds algorithmic improvements.
- We developed the first true linear-time algorithm for computing the inversion distance between two signed permutations.
- We developed the first family of fast-converging phylogeny reconstruction algorithms for sequence data.

Impact in Computational Biology

- Much faster implementations can alter the practice of research in biology and medicine.
  Reducing the time of an analysis from 2 years down to a day makes an enormous difference in the pace and cost of drug discovery and development.

Impact in Computational Biology

- Much faster implementations can alter the practice of research in biology and medicine.
  Reducing the time of an analysis from 2 years down to a day makes an enormous difference in the pace and cost of drug discovery and development.
- Fast and accurate analysis software enables researchers to pursue more leads, develop better intuition on small datasets, and form new conjectures about biological mechanisms.
  Even when the software does not scale up to "industrial-strength".