GRAPPA

A High-Performance Computational Tool For Phylogeny Reconstruction From Gene-Order Data

Bernard M.E. Moret

Department of Computer Science

University of New Mexico

Albuquerque, NM 87131

Collaborators and Sponsors

In collaboration with

- David A. Bader, *Electrical and Computer Engineering, University of New Mexico*
- Tandy Warnow, *Computer Sciences, University of Texas at Austin*
- Stacia K. Wyman, *Computer Sciences, University of Texas at Austin*
- Mi Yan, *Electrical and Computer Engineering, University of New Mexico*

Overview

- Gene-Order Data
- Difficulties in Using Gene-Order Data
- Breakpoint Analysis
- GRAPPA: Breakpoint and Inversion Analyses
- Speed!
- Capabilities and Extensions

Gene-Order Data

- Another source of phylogenetic signal

Gene-Order Data

- Another source of phylogenetic signal
- Slow evolution, so good for deep nodes

Gene-Order Data

- Another source of phylogenetic signal
- Slow evolution, so good for deep nodes
- Inversions, transpositions, duplications, etc.

Gene-Order Data

- Another source of phylogenetic signal
- Slow evolution, so good for deep nodes
- Inversions, transpositions, duplications, etc.
- Inversion is the main mechanism in organellar genomes

Phylogenies from Gene Orders

Distances are hard to compute!

Inversion distances can be computed in linear time (Bader/Moret/Yan), but others can only be approximated.

Models are primitive!

The Nadeau-Taylor model is the most basic possible. No formal ML approach has been suggested.

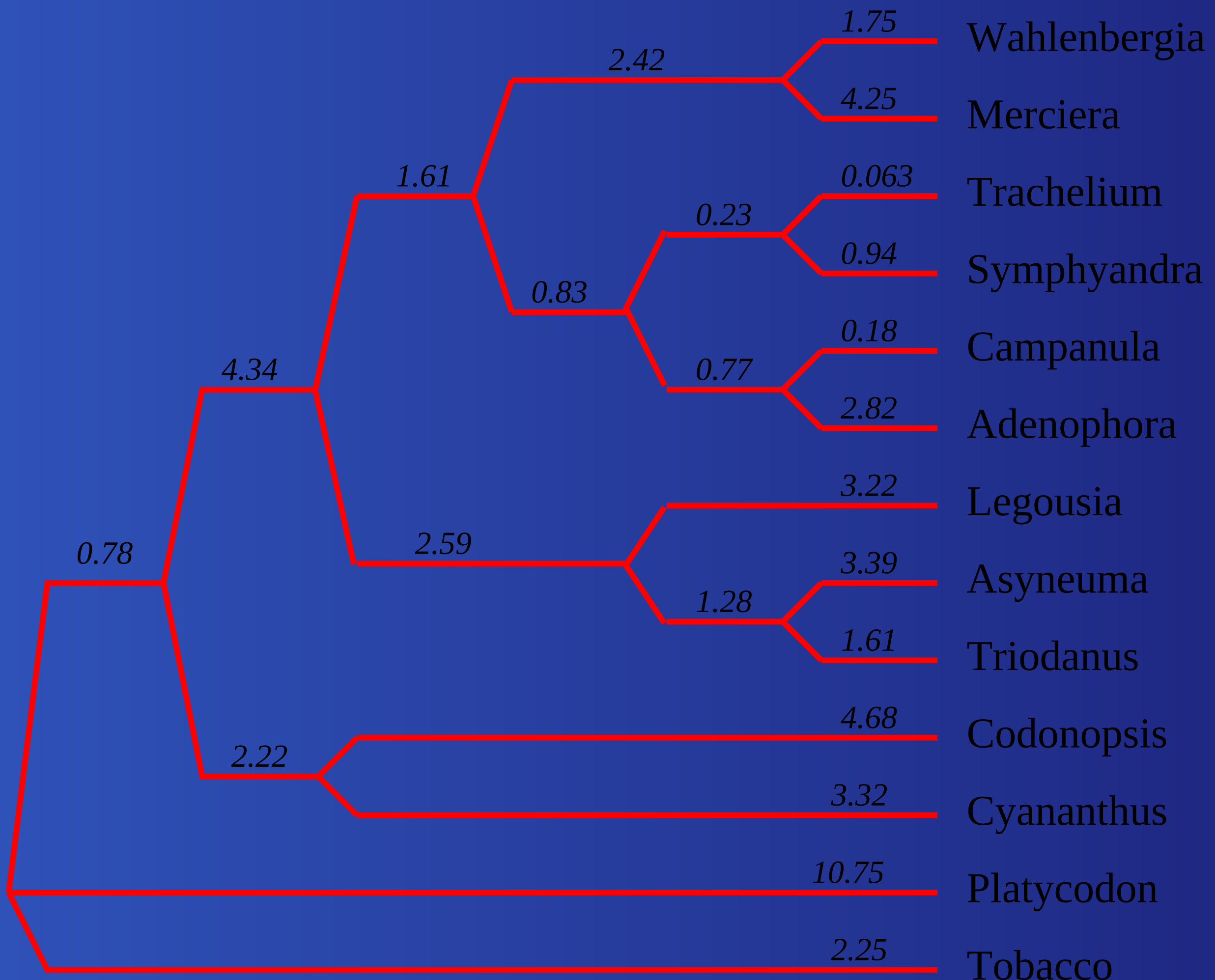
An Example: the Bluebell Family

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

These chloroplasts have a single circular chromosome with about 120 genes.

Target: reconstruct the most parsimonious phylogeny. i.e., the tree that minimizes the total evolutionary distance.

12 Species of *Campanulaceae*



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 Breakpoint

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.*

We can similarly define the Median Problem for Inversions *or other distance measures.*

MPB: an example

Let the (circular) permutations be

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

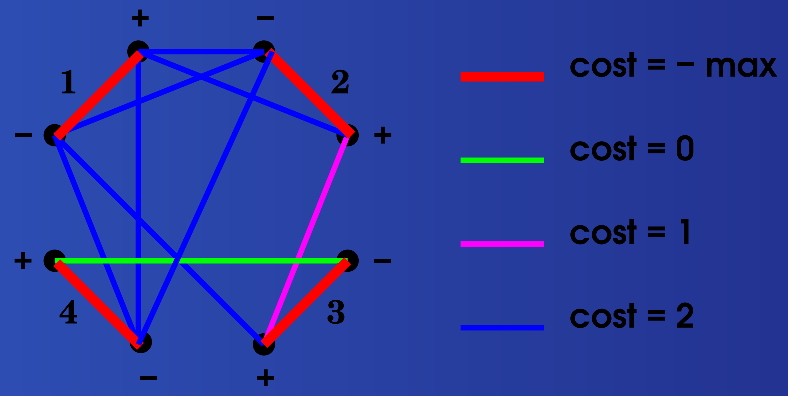
A possible median is -1 2 -3 -4, with cost 5

```
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
```

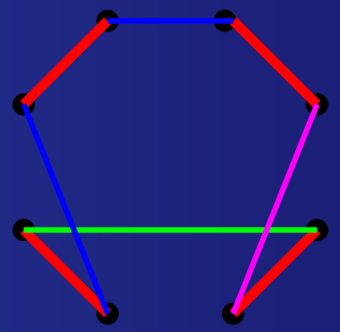
MPB (cont'd)

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

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.

Speed!

- Current release (1.03) runs from 2,000 to 10,000 times faster than the original tool, while also giving more capabilities.

Speed!

- Current release (1.03) runs from 2,000 to 10,000 times faster than the original tool, while also giving more capabilities.
- Research version (1.1) runs from 10,000 to 500,000 times faster than the original tool, thanks to much better bounding.

Speed!

- Current release (1.03) runs from 2,000 to 10,000 times faster than the original tool, while also giving more capabilities.
- Research version (1.1) runs from 10,000 to 500,000 times faster than the original tool, thanks to much better bounding.
- The 13-genome *Campanulaceae* now takes a few hours on a laptop (instead of a few centuries).

Speed!

- Current release (1.03) runs from 2,000 to 10,000 times faster than the original tool, while also giving more capabilities.
- Research version (1.1) runs from 10,000 to 500,000 times faster than the original tool, thanks to much better bounding.
- The 13-genome *Campanulaceae* now takes a few hours on a laptop (instead of a few centuries).
- Speedup on Los Lobos is over 100,000,000!

Capabilities (Version 1.03)

- Can use either breakpoint or inversion distance

Capabilities (Version 1.03)

- Can use either breakpoint or inversion distance
- Can use a variety of TSP solvers (including the CONCORDE solvers)

Capabilities (Version 1.03)

- Can use either breakpoint or inversion distance
- Can use a variety of TSP solvers (including the CONCORDE solvers)
- Includes sampling mechanism (every kth tree) for larger problems

Capabilities (Version 1.03)

- Can use either breakpoint or inversion distance
- Can use a variety of TSP solvers (including the CONCORDE solvers)
- Includes sampling mechanism (every kth tree) for larger problems
- Can use any constraint tree

Capabilities (Version 1.03)

- Can use either breakpoint or inversion distance
- Can use a variety of TSP solvers (including the CONCORDE solvers)
- Includes sampling mechanism (every kth tree) for larger problems
- Can use any constraint tree
- Runs under Linux, Solaris, FreeBSD, and Windows, using various compilers

Capabilities (Version 1.03)

- Can use either breakpoint or inversion distance
- Can use a variety of TSP solvers (including the CONCORDE solvers)
- Includes sampling mechanism (every kth tree) for larger problems
- Can use any constraint tree
- Runs under Linux, Solaris, FreeBSD, and Windows, using various compilers
- Already parallelized to run under MPI

Additional Capabilities (Version 1.1)

- Includes both exact breakpoint median and exact inversion median, generalizable trivially to mixed distances

Additional Capabilities (Version 1.1)

- Includes both exact breakpoint median and exact inversion median, generalizable trivially to mixed distances
- Includes EDE distance correction

Additional Capabilities (Version 1.1)

- Includes both exact breakpoint median and exact inversion median, generalizable trivially to mixed distances
- Includes EDE distance correction
- Computes greedy insertion trees under perfect elimination ordering (often better than NJ trees)

Extensions

- Computation of MPBE and DCM-* trees (see other talks by our group)

Extensions

- Computation of MPBE and DCM-* trees (see other talks by our group)
- Handling unequal gene content due to duplication (inverted repeat, others)

Extensions

- Computation of MPBE and DCM-* trees (see other talks by our group)
- Handling unequal gene content due to duplication (inverted repeat, others)
- Stronger upper bounds to enable a true branch-and-bound exploration

Extensions

- Computation of MPBE and DCM-* trees (see other talks by our group)
- Handling unequal gene content due to duplication (inverted repeat, others)
- Stronger upper bounds to enable a true branch-and-bound exploration
- More efficient and completely general median solver

Conclusions

- GRAPPA is a flexible and efficient tool to examine all possible tree topologies for a collection of genomes with equal gene content, but variable gene order.

Conclusions

- GRAPPA is a flexible and efficient tool to examine all possible tree topologies for a collection of genomes with equal gene content, but variable gene order.
- GRAPPA has confirmed that gene orders have a significant phylogenetic signal (Campanulaceae, other datasets).

Conclusions

- GRAPPA is a flexible and efficient tool to examine all possible tree topologies for a collection of genomes with equal gene content, but variable gene order.
- GRAPPA has confirmed that gene orders have a significant phylogenetic signal (Campanulaceae, other datasets).
- GRAPPA is parsimony-based, which may not be the best criterion (even with distance corrections such as EDE) when distantly-related species are included.