Phylogenetic Reconstruction from Gene-Order Data

Bernard M.E. Moret

`compbio.unm.edu`

Department of Computer Science
University of New Mexico

Acknowledgments

- Close Collaborators:
  - at UNM: David Bader
  - at UT Austin: Tandy Warnow, Robert Jansen, Randy Linder
- Postdocs and Students:
  - Postdocs: Tanya Berger-Wolf, Tiffani Williams
  - Students: Diana Aranda, Zak Betz, Joel Earnest-DeYoung, Tao Liu, Mark Marron, Jijun Tang, Krister Swenson, Anna Tholse, Laura Waymire
- Support:
  - National Science Foundation
  - Alfred P. Sloan Foundation
  - IBM Corporation

Overview

- Phylogenies and Phylogenetic Data
- Gene-Order Data vs. Sequence Data
- Phylogenetic Reconstruction
- Computing with Gene-Order Data
- Reconstruction from Gene-Order Data
- Experimentation in Phylogeny
- Some Experimental Results
- Open Problems and Conclusion

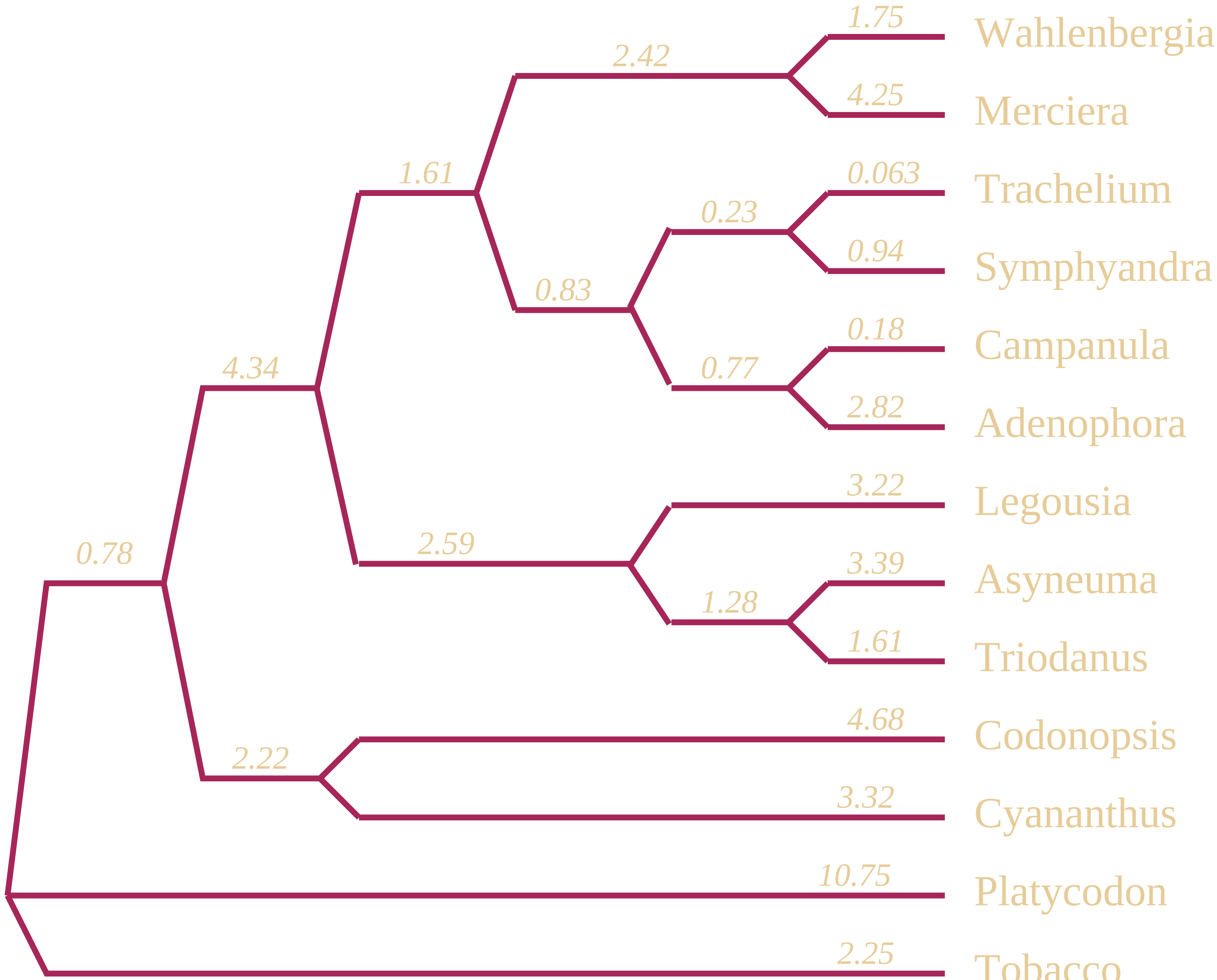
Phylogenies

A phylogeny is a reconstruction of the evolutionary history of a collection of organisms.

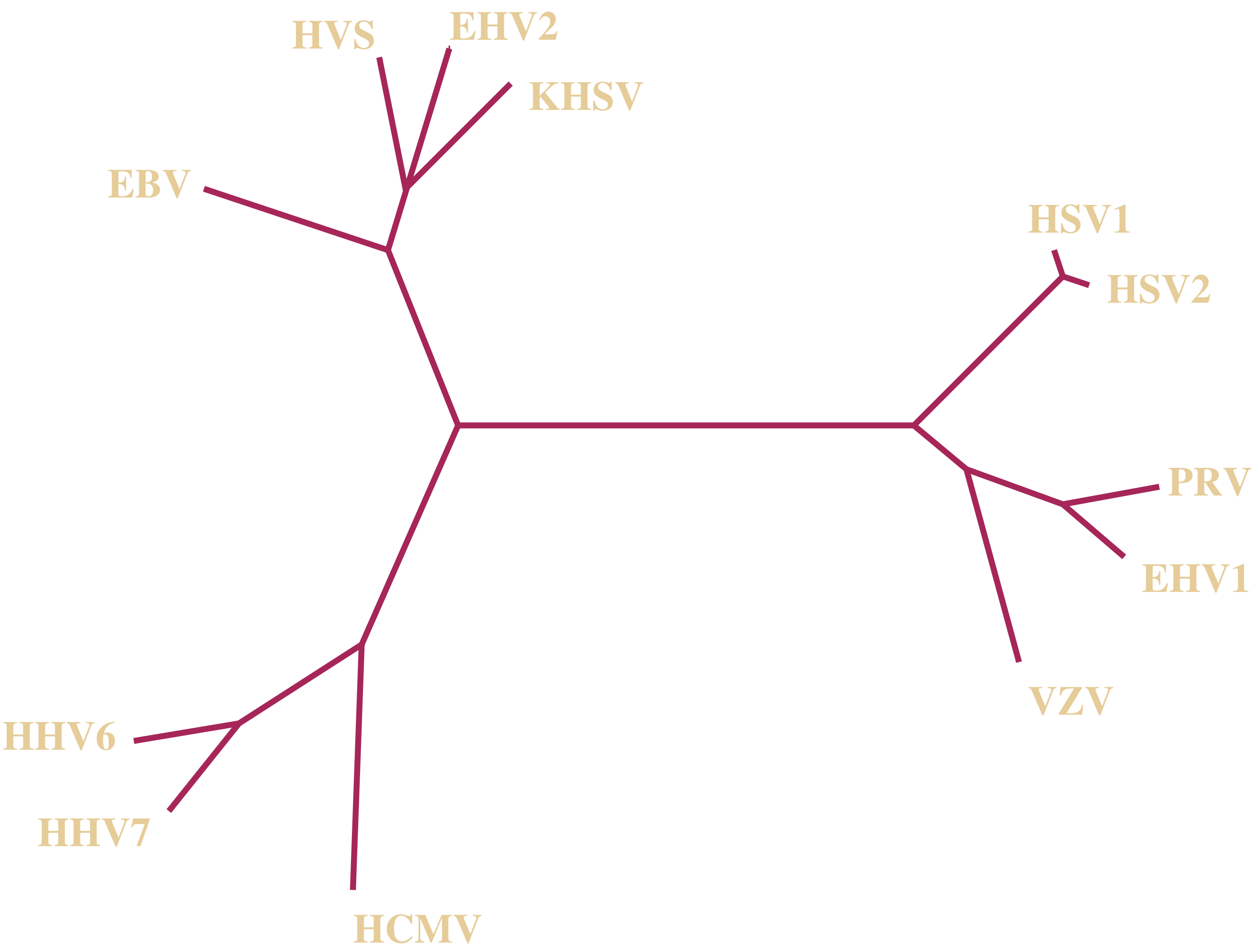
It usually takes the form of a tree.

- Modern organisms are placed at the leaves.
- Edges denote evolutionary relationships.
- “Species” correspond to edge-disjoint paths.

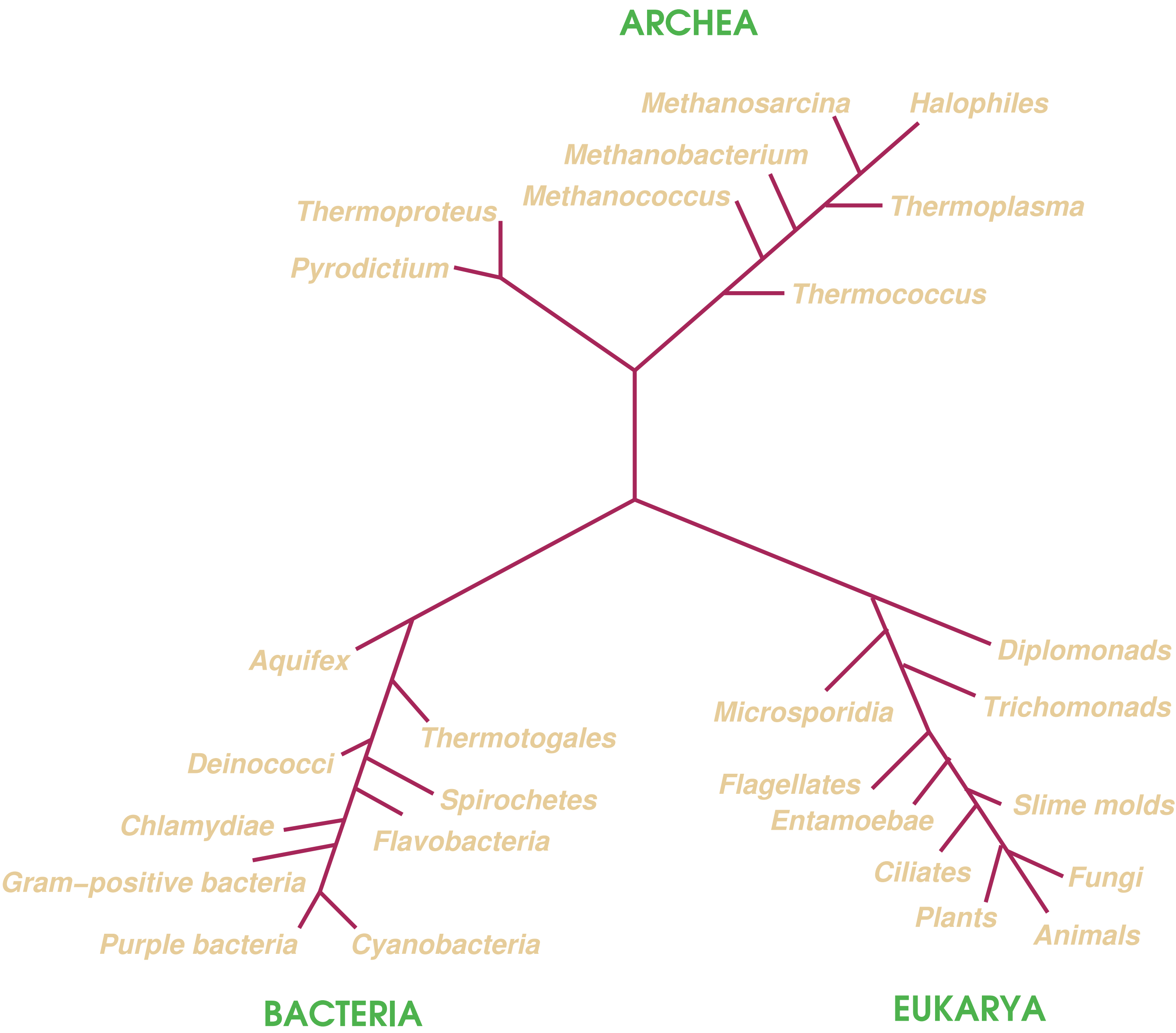
12 Species of Campanulaceae



Herpes Viruses that Affect Humans



The Tree of Life



Phylogenetic Data

- **All kinds of data have been used: behavioral, morphological, metabolic, etc.**
- **Predominant choice is molecular data.**
- **Two main kinds of molecular data:**
  - **sequence data**
    (DNA sequence on genes)
  - **gene-order data**
    (gene sequence on chromosomes)
- **Data elements that can assume new values (according to the model) independently of others are called *characters*.**

Sequence Data

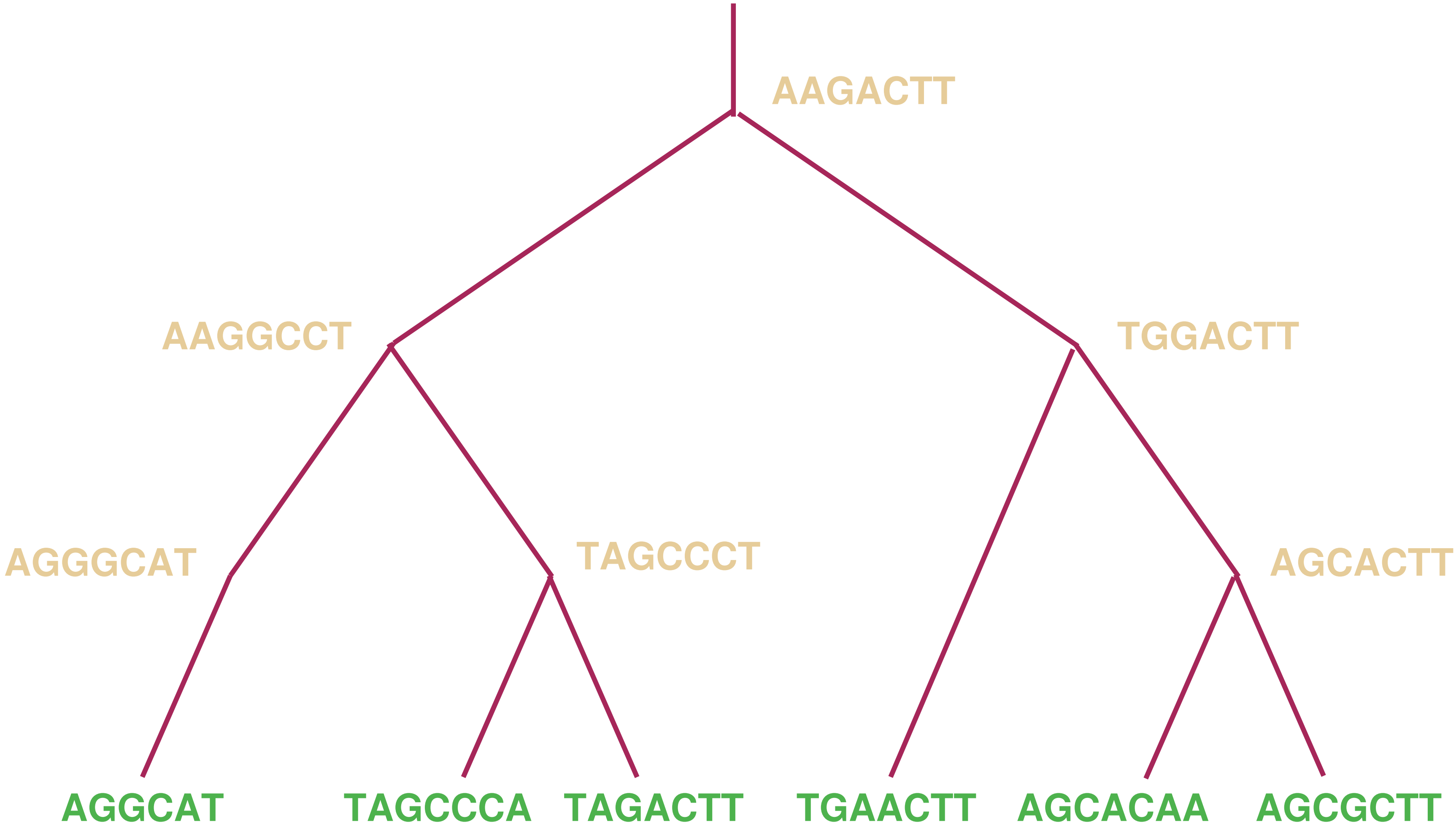
Typically the DNA sequence of a few genes.

Characters are individual positions in the string and can assume four states.

Evolves through point mutations, insertions (incl. duplications), and deletions.

- Find homologous genes across all organisms.
- Align gene sequences for the entire set (to identify gaps – insertions and deletions – and point mutations).
- Decide whether to use a single gene for each analysis or to combine the data.
- Lengths limited by size of genes
  (typically several hundred base pairs)

Sequence Data: Illustration



Sequence Data: Attributes

- **Advantages:**
  - Large amounts of data available.
  - Accepted models of sequence evolution.
  - Models and objective functions provide a reasonable computational framework.
- **Problems:**
  - Sequencing errors (down to $\sim$1%).
  - Fast evolution restricts use to a few million years.
  - Gene evolution need not be identical to organism evolution.
  - Multiple alignments are not well solved.
  - Reconstruction methods do not scale well (in terms of accuracy and running time).

Gene-Order Data

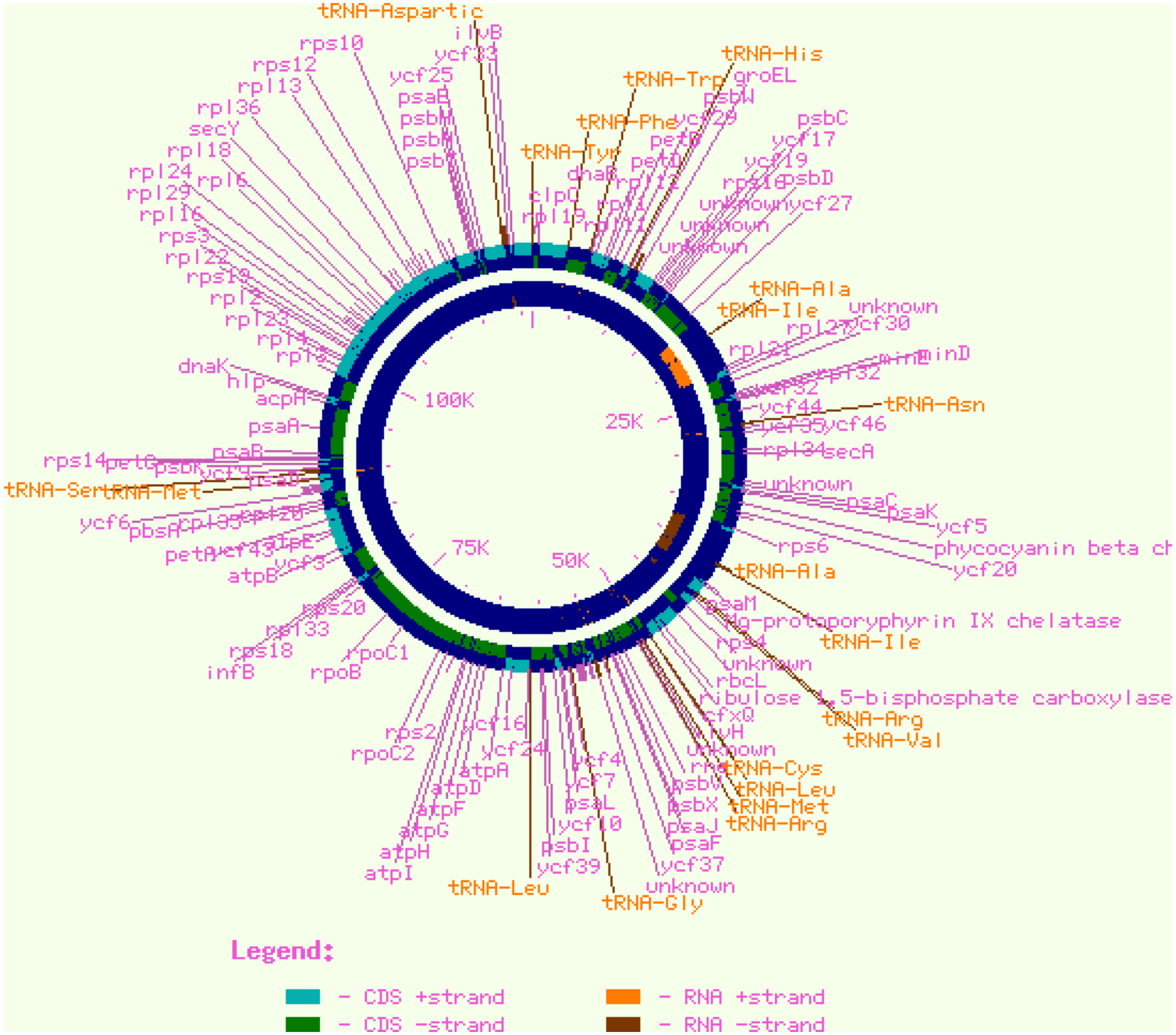
The ordered sequence of genes on one or more chromosomes.

Entire gene-order is a single character, which can assume a huge number of states.

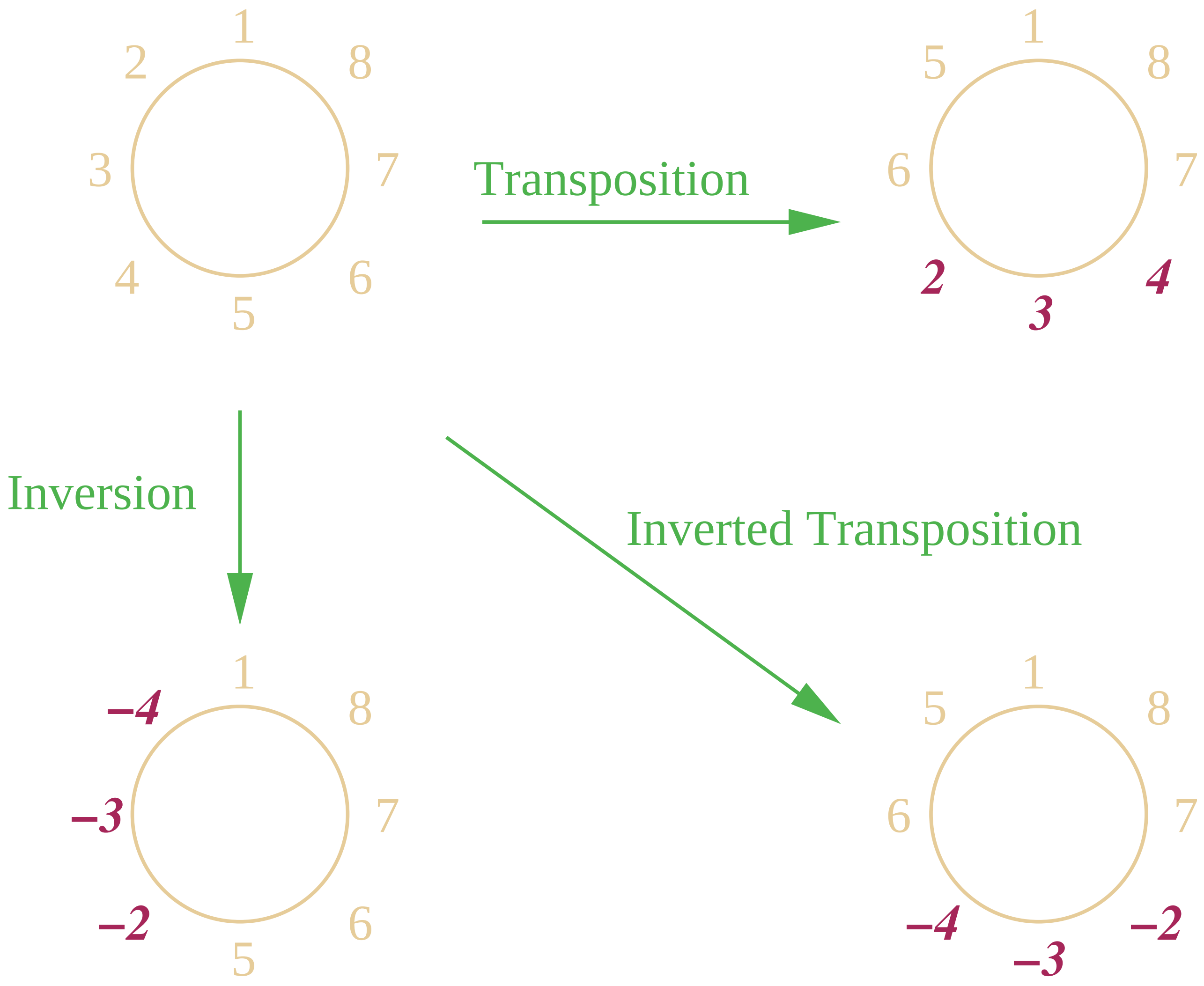
Evolves through inversions, insertions (incl. duplications), and deletions; also transpositions (in mitochondria) and translocations (between chromosomes).

- Identify homologous genes, including duplications.
- Refine rearrangement model for collection of organisms (e.g., handle bacterial operons or eukaryotic exons explicitly).

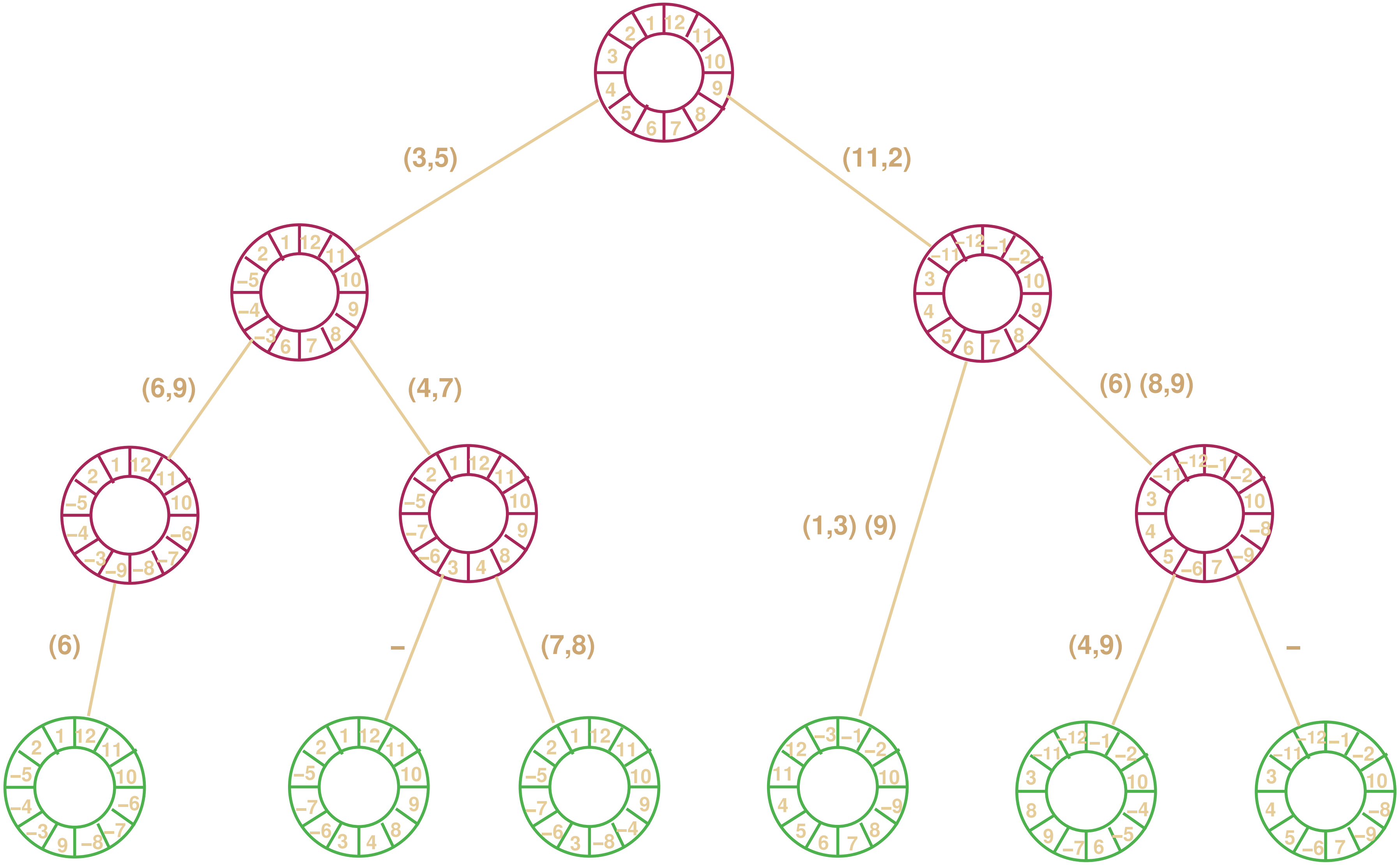
Gene-Order: Guillardia Chloroplast



Gene-Order Data: Rearrangements



Gene-Order Data: Inversion-Only



Gene-Order Data: Attributes

- **Advantages:**
  - Low error rate (depends on recognizing homologies).
  - No gene tree/species tree problem.
  - Rare evolutionary events and unlikely to cause “silent" changes—so can go back hundreds of millions years.
- **Problems:**
  - Mathematics *much more complex* than for sequence data.
  - Models of evolution not well characterized.
  - Very limited data (mostly organelles).
  - Possibly insufficient discrimination among recently evolved organisms.

Existing Gene-Order Data

- Mitochondria for plants and animals ($\sim$200):
  - single circular chromosome with $\sim$40 genes
  - mostly transpositions and insertions/deletions
- Chloroplasts for plants and animals ($\sim$100):
  - single circular chromosome with $\sim$150 genes
  - mostly inversions and insertions/deletions
  - includes a single large inverted repeat
- Bacterial genomes ($\sim$50):
  - from a few hundred to several thousand genes
  - complex evolution, many lateral transfers
- Also 9 nuclear genomes and $\sim$100 phages.
  - Nuclear genomes can be very large and far more complex than organellar or bacterial genomes.
  - Phages are viruses, with very simple genomes, but complex evolutionary events.

Gene-Order Data vs. Sequence Data

	Sequence	Gene-Order
evolution	fast	slow
errors	significant	negligible
data type	a few genes	whole genome
data quantity	abundant	sparse
models	good	primitive
computation	easy	hard

Phylogenetic Reconstruction

Three categories of methods:

- Distance-based methods, such as neighbor-joining.
- Parsimony-based methods (such as implemented in PAUP*, Phylip, Mega, TNT, etc.)
- Likelihood-based methods (including Bayesian methods, such as implemented in PAUP*, Phylip, FastDNAML, MrBayes, GAML, etc.)

In addition:

- Meta-methods (quartet-based methods, disk-covering method) decompose the data into smaller subsets, construct trees on those subsets, and use the resulting trees to build a tree for the entire dataset.

Evolutionary Distances

- **True evolutionary distance**:
  the actual number of permitted evolutionary events that took place to transform one datum into the other.
- **Edit distance**:
  the minimum number of permitted evolutionary events that can transform one datum into the other.
- **Expected true evolutionary distance**:
  obtained from the edit distance by correcting for the known (model or experiments) statistical relationship between true and edit distances.

Distance-Based Methods

- Use edit or expected true evolutionary distances.
- Usually run in *low polynomial time*.
- Reconstruct *only topologies*:
  no ancestral data.
- Prototype is Neighbor-Joining; BioNJ and Weighbor are two improvements.
- NJ is optimal on additive distances (where the distance along a path in the true tree equals the pairwise distance in the matrix).
- NJ is statistically consistent (produces the true tree with probability 1 as the sequence length goes to infinity).

Parsimony-Based Methods

- Aim to minimize total *number of character changes* (which can be weighted to reflect statistical evidence).
- Assume that characters are *independent*.
- Reconstruct *ancestral data*.
- Are known not to be statistically consistent with sequence data (but examples are fairly contrived).
- Finding most parsimonious tree is computational very expensive (NP-hard).
- Optimal solutions limited to sizes around 30; heuristic solutions appear fairly good to sizes of 500.

Likelihood-Based Methods

- Are based on a specific model of evolution and *must estimate all model parameters*.
- Produce *likelihood estimate* (prior or posterior conditional) for each tree.
- Are statistically consistent.
- Reconstruct *only topologies*.
- Are prone to numerical problems:
  likelihood of typical tree on 20 items is around 10^{-21}; on 50 items, around 10^{-75}; . . .
- Are presumably NP-hard; even scoring one tree is very expensive.
- Optimal solutions limited to sizes below 10; heuristic solutions appear fairly good to sizes of 100.

Meta-Methods

General Principle:

decompose the dataset into smaller, overlapping subsets, reconstruct trees for the subsets (by some base method), and combine the results into a tree for the entire dataset.

- Quartet-based methods: use all possible smallest subsets (a quartet is a set of 4 genomes); include Q*, Tree-Puzzle, Quartet-Cleaning.
  Slow and inherently *inaccurate* regardless of base method—consistently surpassed by NJ.
- Disk-covering method (DCM): set up a graph from the distance matrix, find cliques in the graph, use cliques for decomposition.

  High-powered machinery *succeeds* very well, especially when tree is imbalanced.

Computing with Gene-Order Data

- Distances
- Evolutionary models and distance corrections
- Reconstructing ancestral genomes
- The median problem

Distances for Gene-Order Data

- Breakpoint [Sankoff et al., 1998]:
  Distance counting the number of altered adjacencies for identical gene content; linear time.
- INV [Bader, Moret, Yan, 2001]:
  Edit distance (inversions) for identical gene content; linear time.
- INV-DEL [El-Mabrouk, 2000]:
  Edit distance (inversions and insertions/deletions, but no duplications); doable in linear time [Liu, Moret, 2003].
- ALL [Marron, Swenson, Moret, 2003]:
  Estimated edit distance (inversions, insertions/deletions, duplications).
- CLUSTER [Bergeron, Stoye, 2003]:
  Estimate based on number and lengths of conserved gene clusters (no duplications).

Breakpoint Distance

The number of adjacencies present in one genome, but not the other.

```
G1=(1   2   3   4   5   6   7   8)

G2=(1   2  -5  -4  -3   6   7   8)
```

Inversion Distance

Given two signed gene orders of equal content, compute the inversion-only edit distance.

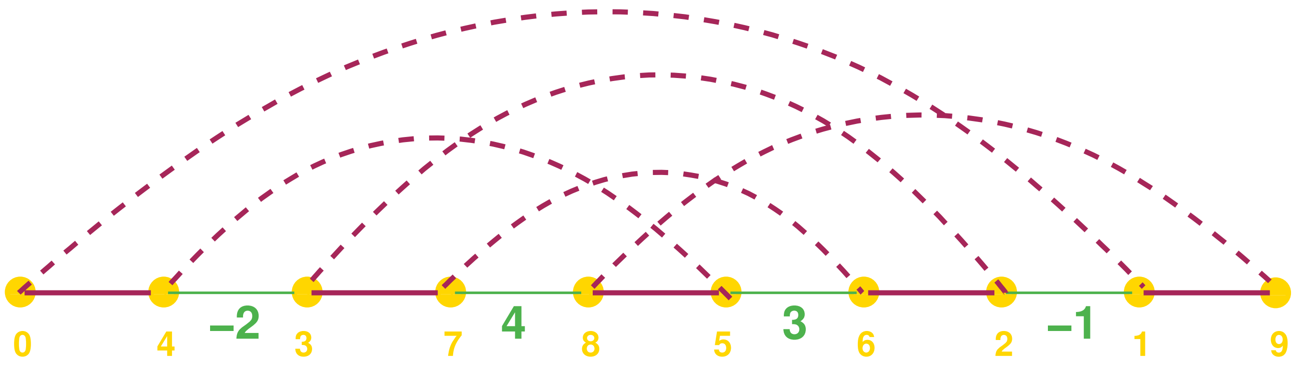
- The problem is NP-hard for *unsigned* permutations.
- Finding the shortest sequence of inversions can be done in polynomial time (Hannenhalli and Pevzner—very difficult result).
- Running time improved by Berman and Hannenhalli, then by Kaplan, Shamir and Tarjan: $O(dn)$, where d is the distance and n the number of genes.
- Theory simplified (with new algorithm, faster in practice) by Bergeron.
- Distance computation reduced to linear time by Bader, Moret, and Yan.

Inversion Distance

Algorithm is based on the breakpoint graph.

Assume one permutation is identity; each green edge is a single gene.

Solid red edges denote existing adjacencies and *dashed red* edges denote desired adjacencies.



Inversion distance is

n - #cycles + #hurdles + (fortress)

Gene-Order Distances in General

The signed gene orders may include duplications and need not have identical gene content.

- Extension to translocations (transpositions between chromosomes) by Hannenhalli and Pevzner. Implemented by Tesler as GRIMM.
- Extension to inversions and deletions (no duplications allowed) by El-Mabrouk. Linear-time distance computation by Liu and Moret.
- Heuristic for duplications by Sankoff (exemplars).
- Heuristic for unequal gene content by Bourque.
- Bounded approximation by Marron, Swenson, and Moret.

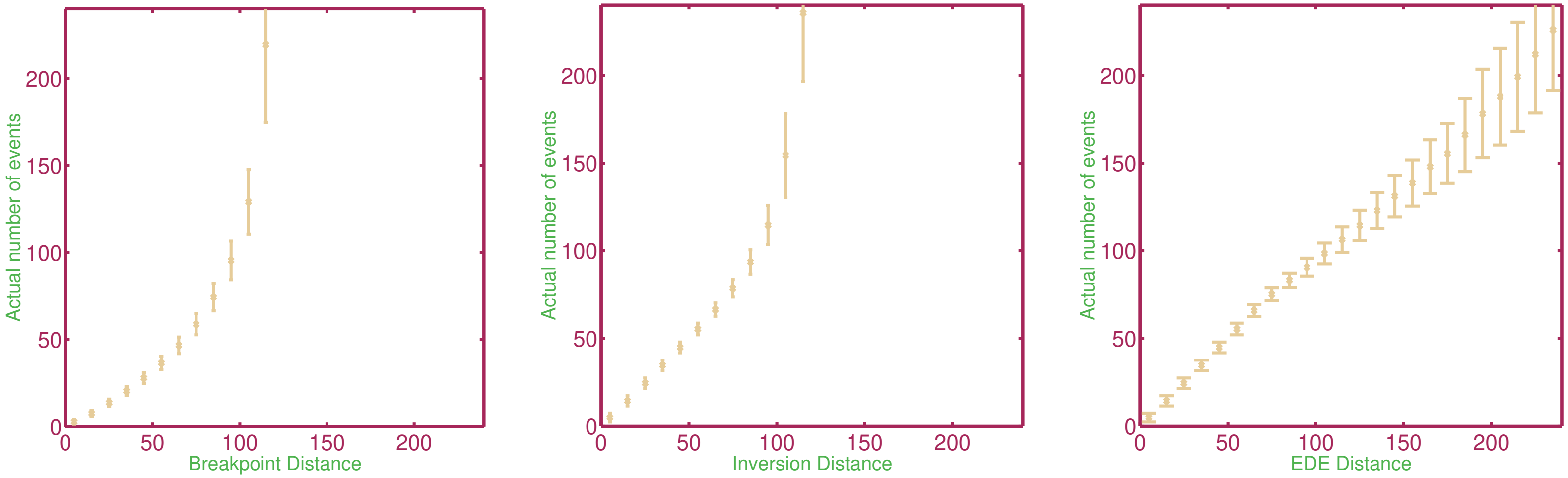
Evolutionary Models for Gene Order

- Strong biological evidence for inversions in chloroplasts.
- Strong computational evidence for transpositions in mitochondria.
- Computational evidence (Sankoff, El-Mabrouk) for short inversions in bacterial genomes (preserve gene clusters).
- Hard radiation plus repair mechanisms could create more complex rearrangements.
- Respective probabilities unknown.

Distance Corrections for Gene Order

- Assume a distribution of events and compute relationship between number of events and edit distance.
- Wang and Warnow gave an exact derivation for correcting the breakpoint distance into an expected number of inversions (IEBP).
- Moret, Tang, Wang, and Warnow gave an empirical derivation for correcting the inversion edit distance into an expected number of inversions (EDE).
- EDE correction substantially improves the performance of both distance-based and parsimony-based reconstruction methods.

EDE Distance Correction



breakpoint distance

EDE corrected distance

inversion distance

Vertical axis is actual number of inversions.
Note increased standard deviation of EDE at larger distances.

Reconstructing Ancestral Genomes

Goal: Reconstruct a signed gene order at each internal node in the tree to minimize sum of edge distances.

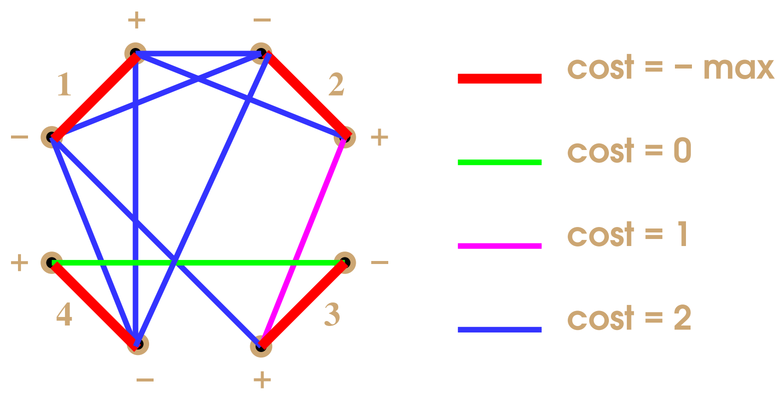
Problem is NP-hard even for just three leaves and simplest of distances (breakpoint, plain inversion)!

This is the median problem for signed genomes: given three genomes, produce a new genome that will minimize the sum of the distances from it to the other three.

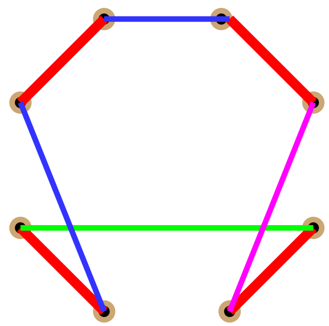
Median Problem for Breakpoints

Sankoff showed to to convert this problem to the Travelling Salesperson Problem when all three have identical gene content.

+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

Median Problem for Inversions

No simple formulation in terms of a standard optimization problem.

- Exact solutions given by Siepel and Moret and by Caprara for identical gene content; work well for distances to median of 0–15 inversions.
- Various heuristics proposed by Bourque and Pevzner and others.
- Siepel and Moret showed that the inversion median is preferable to the breakpoint median (fewer ties).

Median with Deletions/Duplications

Tang and Moret showed it can be solved exactly for small numbers of deletions and duplications.

- Assume no change is reversed and that changes are independent and of low probability.
- Gene content of median can then be determined by preferring one event (e.g., one insertion) over two concurrent events (e.g., two deletions).
- Knowing gene content, all combinations of duplications or insertion locations can be examined—choice grows exponentially, but is manageable for organellar genomes.
- Accuracy is very good (less than 5% error).

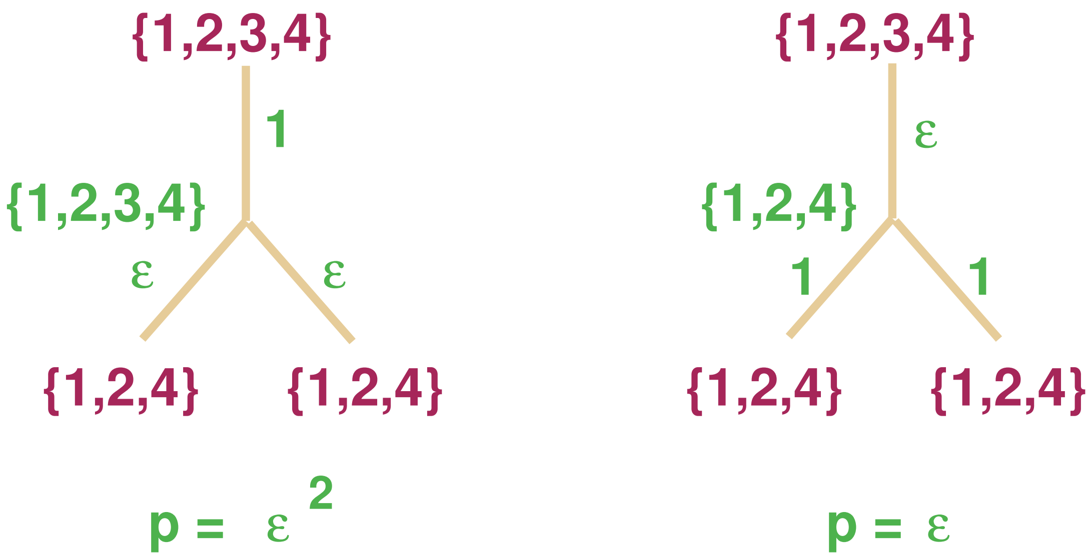
Gene Content of Median

Assumptions:

Probability of a deletion is ε (very small).

Probability of no change is $\sim$1.

Example:



Reconstruction from Gene-Order Data

- **Distance methods**
  - NJ and Weighbor with corrected distances
  - Combining with a DCM booster
- **Parsimony-based methods**
  - Encoding approaches: MPBE, MPME
  - Direct approaches: BPAnalysis, GRAPPA, MGR, DCM-GRAPPA
- **Likelihood-based methods**

Distance Methods

Neighbor-joining and Weighbor using BP, INV, IEBP, and EDE distances.

Moret, Wang, and Warnow showed that NJ-EDE or Weighbor-EDE are clearly better than other combinations:

- Low error rates up to a few hundred genomes.
- Robust against various models of genome rearrangements.
- Suffer when the tree diameter is large (some large pairwise distances).

Improved by using DCM boosting.

The DCM Boosting Approach

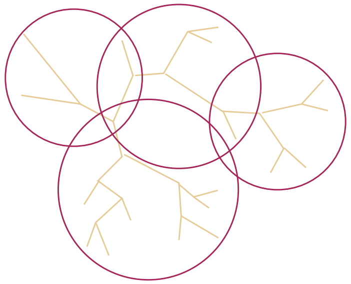
A family of divide-and-conquer methods (Warnow and colleagues).

1. Compute the distance matrix.
2. Threshold the distance matrix (eliminate entries above threshold).
3. Create corresponding graph and triangulate it.
4. Find maximum cliques (polynomial time in triangulated graphs).
5. Create disks (data subsets) with prescribed overlap.
6. Invoke phylogenetic base method on each disk.
7. Assemble final tree from disk trees through consensus and refinement.

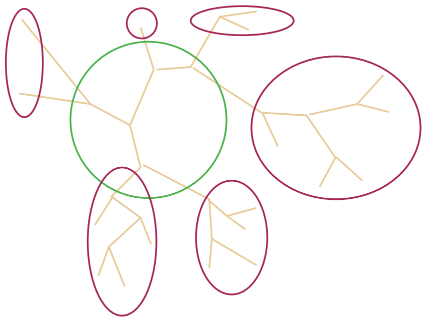
DCM Methods

Two varieties so far (3rd in development):

- DCM-1: Each clique is a disk, so disk diameter minimized.



- DCM-2: Compute graph separator; separator plus graph component is a disk, so all disks share same subset.



DCM-Boosted Distance Methods

- DCM-1-NJ (with an MP last step) beats NJ and greedy MP on sequence data and is robust against size, rate, and other model variations (Moret/Nakhleh/Roshan/Wang/Warnow).
- DCM-1-GRAPPA scales gracefully from the limit of 15 genomes for GRAPPA to at least 1,000 genomes (Tang and Moret).
- DCM-3-MP does better than PAUP* MP or ratchet MP on large real datasets (new results, Warnow lab).

Encoding Methods: MPBE

Idea: encode the signed permutation into a sequence, then use sequence methods.

MPBE: Maximum Parsimony on Breakpoint Encodings

(Cosner, Jansen, Moret, Raubeson, Wang, Warnow, and Wyman)

- New characters will be all gene adjacencies present in the data.
- If m such adjacencies are present, then produce strings of m binary characters:
  index the adjacencies arbitrarily and,
  for each genome, code presence or absence
  of that adjacency with a 1 or 0.
- Use an MP method to reconstruct a tree.

Slower than NJ-EDE or Weighbor-EDE, cannot describe new adjacencies, not all binary strings are valid codes.

Encoding Methods: MPME

Idea: preserve more information than MBPE.

MPME: Maximum Parsimony on Multistate Encodings

(Wang, Jansen, Moret, Raubeson, and Warnow)

- New characters correspond to possible signed genes, so *2n* characters; each character represents a signed gene and so can assume one of *2n* states.
- Site *i* in the sequence takes the value of the gene immediately following gene *i* in the genome (order is reversed for negative signs).

(1,-4,-3,-2) –> (-4,3,4,-1,2,1,-2,-3)

Clearly better than MPBE, but slower and number of character states quickly exceeds limits of popular MP software.

Direct Approaches: BPAnalysis

(due to Sankoff and Blanchette)

Initially label all internal nodes with gene orders

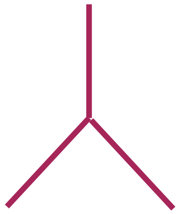
Repeat

- **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 tree score, then do it**

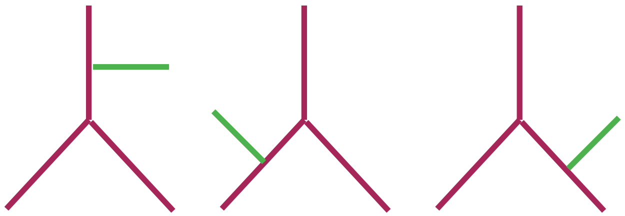
until no internal node can be relabelled

The Number of Trees on N Genomes

- 3 genomes: **1 tree**



- 4 genomes: **3 trees**



- 5 genomes: **15 trees**
- 13 genomes: **13.5 billion trees**
- *n* genomes: **$(2n\text{-}5)!!$ trees**

  $(2n\text{-}5)!! = (2n\text{-}5)*(2n\text{-}7)*\ldots*5*3$

GRAPPA

Genome **R**earrangements **A**nalysis under **P**arsimony & other **P**hylogenetic **A**lgorithms

- Began as a reimplementation of **`BPAnalysis`**.
- Current version runs up to one billion times faster than **`BPAnalysis`**, thanks to *algorithmic engineering*.
- Limit: every added taxon multiplies the running time by twice the number of taxa.
  So 13 taxa take 20 mins, 15 taxa two weeks, 16 taxa a year, 20 taxa over 2 million years, and . . .

Algorithm Engineering: Why?

- **Research Tools:** run many experiments to test hypotheses and for discovery
- **Production Tools:** obviously, save on resources
- **Make It Possible:** a million-fold speedup allows one to solve in a week what would otherwise have been infeasible (requiring millennia)

Algorithm Engineering: What?

- good asymptotic performance
- low proportionality constants
- fast running times on important real-world datasets
- robust performance across a variety of data
- robust performance across a variety of platforms
- scalability to faster platforms and larger datasets

Algorithm Engineering: How?

- Optimize low-level data structures (multiple parallel arrays, maintenance vs. recomputation)
- Optimize low-level algorithmic details (e.g., upper and lower bounds)
- Optimize low-level coding (hand-unrolling loops, keeping local variables in registers)
- Reduce memory footprint (the entire code might fit in cache)
- Maximize locality of reference (the memory hierarchy can cause differentials of 100:1)
- Repeatedly rebalance execution profile (eliminate bottlenecks)

MGR

Multichromosomal Genome Rearrangement (Bourque and Pevzner)

- Uses the median approach, but does not solve it exactly.
- Can handle multiple chromosomes (translocations) using Hannenhalli-Pevzner's results.
- Scaling unknown—published results to date limited to small datasets.
- Accuracy second only to GRAPPA.

DCM-GRAPPA

Our extension to GRAPPA to scale it to large datasets (Tang and Moret).

- Scales gracefully to at least 1,000 genomes (less than 2 days of computation).
- Retains accuracy of GRAPPA: error rates on 1,000-genome datasets are consistently below 3%.
- Uses the DCM-1 approach—may do even better with forthcoming DCM-3.

DCM-GRAPPA: Details

- **Compute pairwise distances**
- **Check all possible threshold values**
- **For each threshold value**
  - Discard values above threshold
  - Create graph from reduced distance matrix
  - Triangulate the graph
  - Find maximum cliques (disks) in the graph
  - Run GRAPPA (or recursive DCM-GRAPPA) on the disks
  - Merge the resulting trees

Likelihood Approaches

- Only one to date: an MCMC method (Larget, Kadane, and Simon)
- Relies on two distinct moves through tree space.
- Essentially untested (just two small datasets used in report).
- Promising results, but unknown running time.
- Code not available.

Testing Algorithms

How to choose test sets?

- Biological datasets test performance where it matters, but can be used only for ranking, are too few to permit quantitative evaluations, and are often hard ot obtain.
  Good for anecdotal reports and “reality checks."
- Simulated datasets enable absolute evaluations of solution quality and can be generated in arbitrarily large numbers.
  Only way to obtain valid characterizations.

Experimental Algorithmics

How to test algorithms empirically to obtain reliable characterization of performance.

Can use experience in physical sciences, but must adapt to specific circumstances of computer-based experimentation:

- Algorithm analysis uses asymptotic characterizations—cannot be inferred from finite data.
- Easy to alter experiments and re-run whole series—leads to drift and biases.
- Controlling variables is just as hard in computational experiments as in the natural sciences—even reproducibility is a challenge.

Experimental Algorithmics: Caveat

At the *First Workshop on Algorithms in Bioinformatics*, Alberto Caprara remarked

- a theoretician solves interesting problems with no practical use whatsoever

whereas

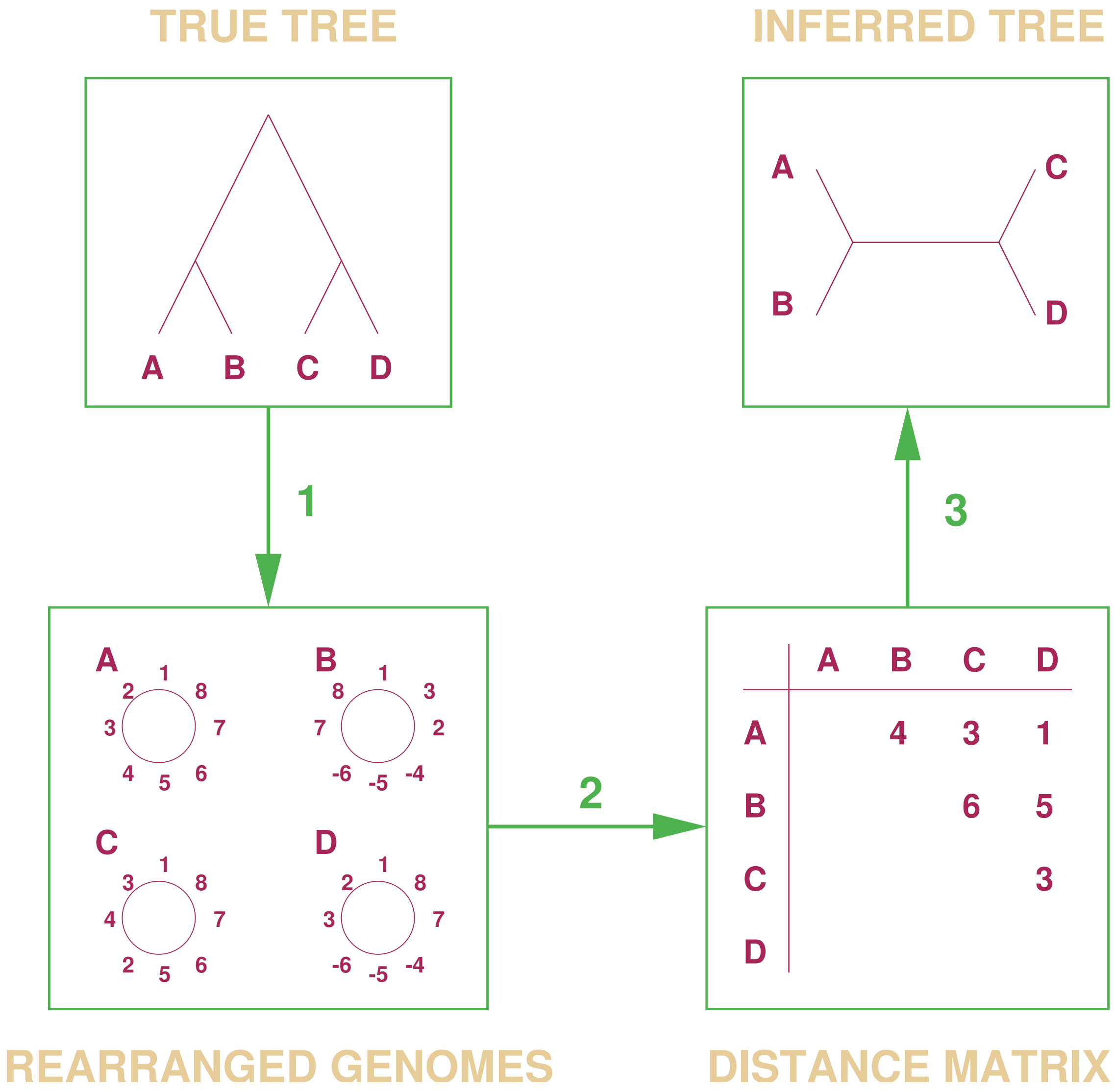
- an experimental algorithmist solves problems of significant practical use, but only tests algorithms on randomly generated instances

The types of instances generated in an experimental study make the difference between useful research and wasting a lot of (students' and CPUs') cycles.

Phylogenetic Considerations

- Tree shape plays a *very large role*.
  Challenge: the shape of the true trees is unknown; moreover, the shape depends on the selection of genomes—e.g., a single genus vs. a sampling of an entire kingdom.
- The evolutionary model is important.
  Challenge: devise an evolutionary model with few parameters that is easily manipulated analytically and computationally and that produces realistic data. (It would also be pleasant if it made biological sense. . .)
- Test a large range of parameters and use many runs for each setting to estimate variance.
  Challenge: even the simplest of models induces a huge parameter space—sampling it requires smoothness guarantees, but NP-hard search spaces lack smoothness.

Typical Simulation Setup



Tree Topologies: Popular Models

- Birth-Death: start with root branch; in any Δt interval, there is probability p of speciation along any of the current branches.
  Biologically motivated; trees are well balanced—too well balanced compared to published phylogenies.

  All distance-based methods do very well on BD trees; DCM decompositions are poor, but not needed.
- Uniform Random: all tree topologies equally likely.
  No biological process; trees are fairly imbalanced—too imbalanced compared to published phylogenies.

  NJ does very poorly on uniform trees; DCM-NJ significantly better.

Tree Topologies: New Models

- Aldous' β-Splitting: parameter can be set to produce anything from a ladder (caterpillar), through uniform, birth-death, to perfect balance.
  No biological process; single parameter cannot localize structure.

  Recommended setting ($\beta = -1$) matches balance of published phylogenies, but algorithmic behavior does not match biological datasets.

- Heard's Multi-Parameter: variable evolutionary rates, inheritable speciation traits, punctuated and gradual evolution.
  Strong biological motivation; too many parameters?

  Subtle differences cause large changes in performance of distance- and parsimony-based methods.

Generating Genome Rearrangements

Questions:

- What rearrangement events? are they weighted?
- Are there forbidden regions (e.g., no inversion across a centromere)? preferred regions or endpoints (hotspots)?
- Relative cost of insertions, duplications, and deletions?
- How to test for robustness?

The Parameter Space

Clearly too large for good sampling:

- tree topologies
- evolutionary models
- evolutionary rates
- tree sizes
- genome sizes
- multiple chromosomes

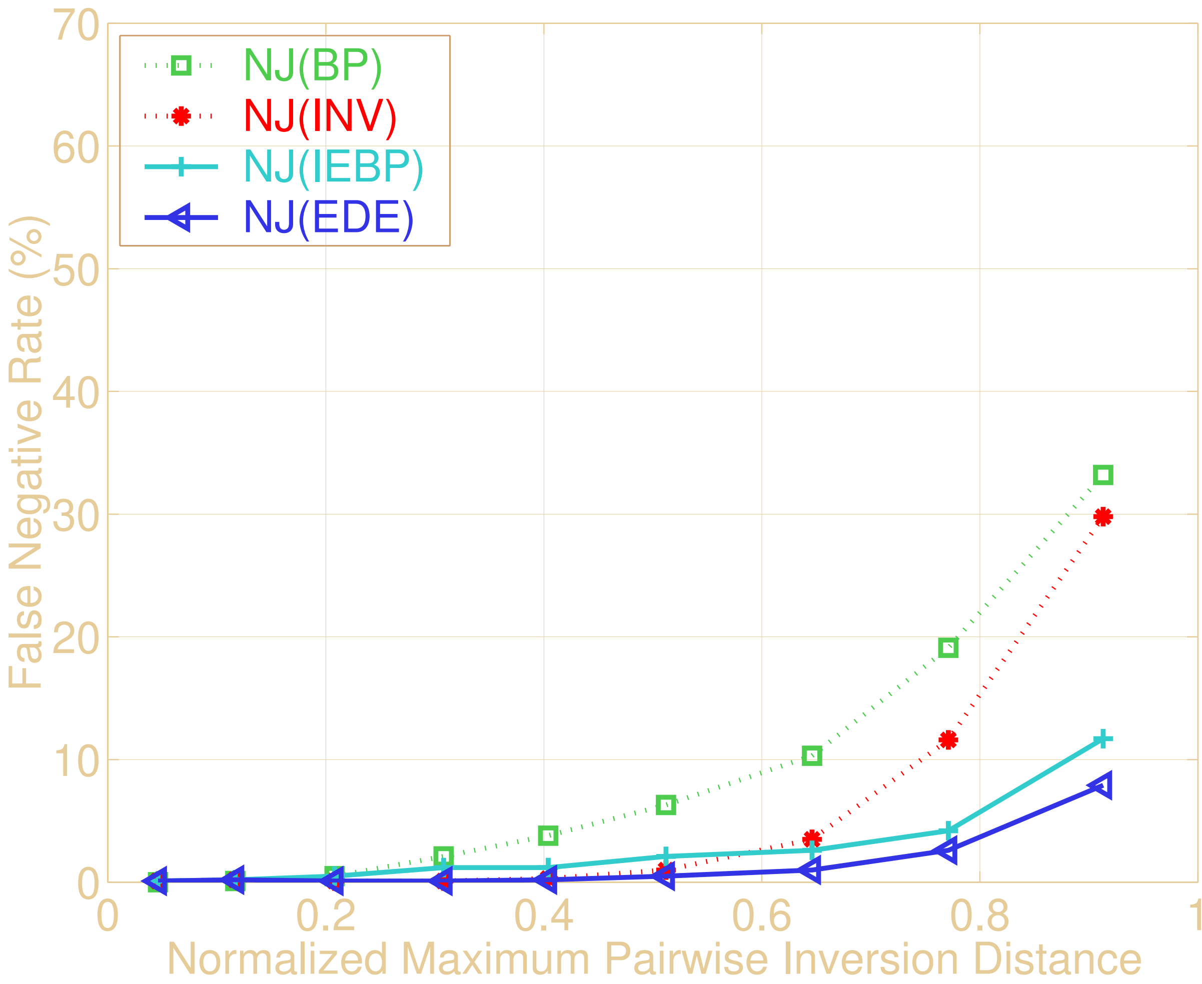
Need to focus testing on appropriate subspace.

Some Experimental Results

- Distance-Based Methods (with and without correction)
- GRAPPA
- DCM-GRAPPA

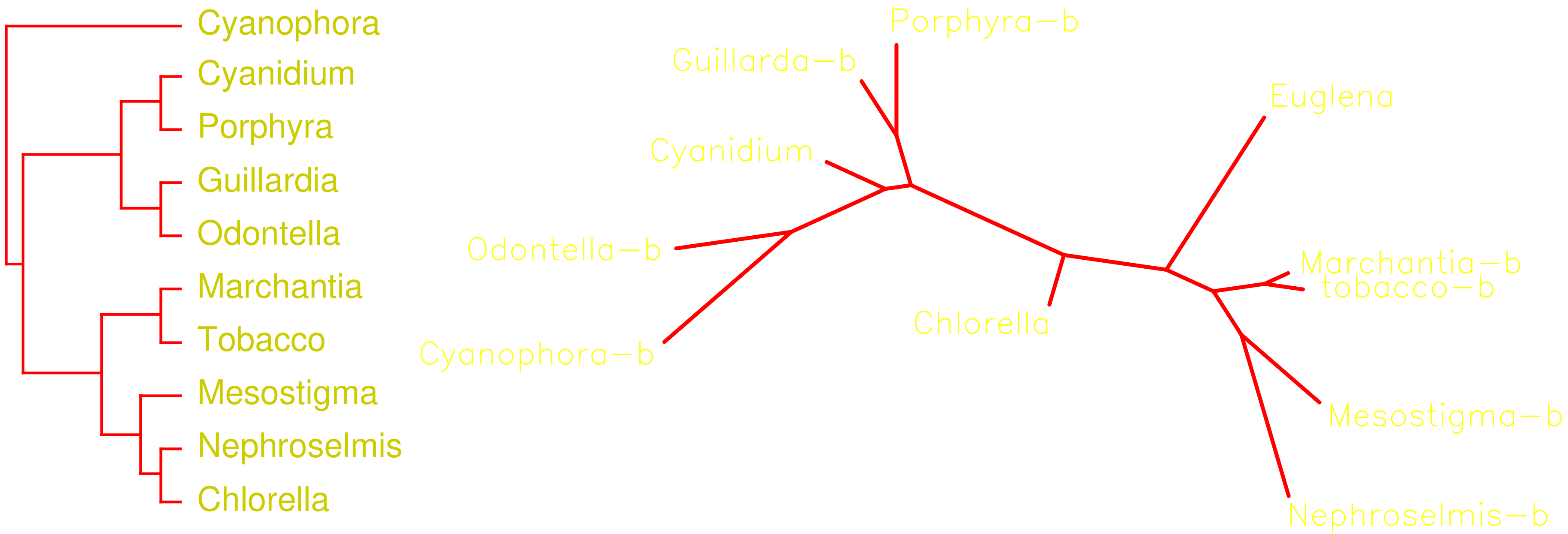
Results: Distance Methods

inversion/transposition/inverted transposition: 1:1:1 ratio
120 genes per genome, 10-20-40-80-160 genomes



Results: GRAPPA

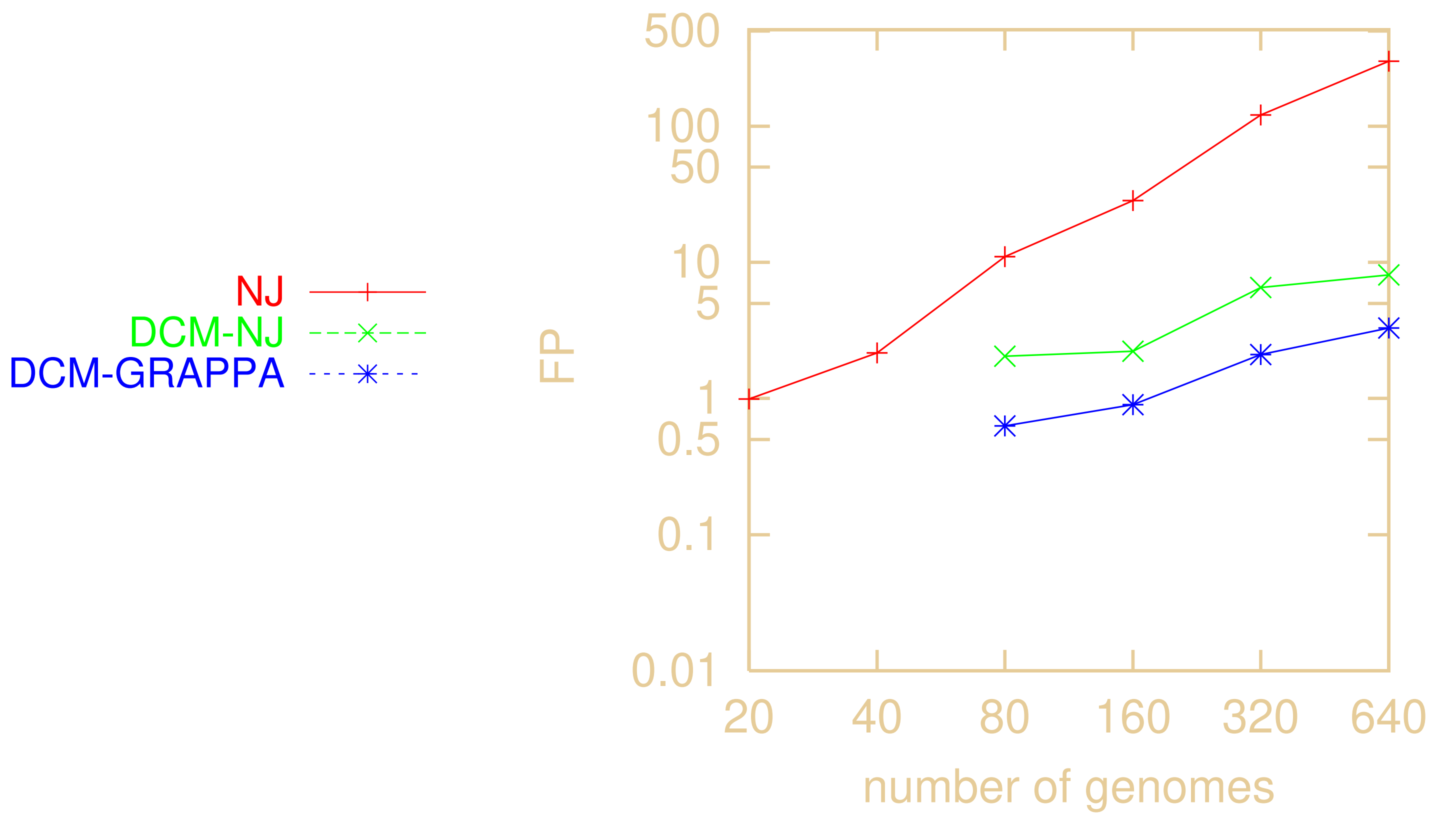
Biological dataset: 11 plants cpDNA genomes, 37 genes
Note: inverted repeat not used here



Left: expected tree (placement of *Euglena* unknown)
Right: GRAPPA tree

Results: DCM-GRAPPA

inversion-only evolution, expected edge length 4
100 genes per genome, 20-40-80-160-320-640 genomes



Shown is total number of edges in error (log/log scale)

Some Open Problems

- Tree models
- Evolutionary models
- Extensions of Hannenhalli-Pevzner theory to handle
  - transpositions and inversions
  - length-dependent rearrangements
  - position-dependent rearrangements
  - duplications
- Good combinatorial formulation of the median problem for inversions and for more general cases.
- Tighter bounds on tree scores.
- Extensions to phylogenetic networks (!)

Conclusions

- Gene-order data carries very good phylogenetic information—much better than sequence data!
- Current algorithmic approaches scale to significant sizes (1,000 for DCM-GRAPPA)—comparable to the best achievable with sequence data and with better results.
- Current approaches remain unable to handle unequal gene content with duplications, but major progress has been made over the last 5 years.
- Data availability is increasing rapidly for organellar genomes, slowly for nuclear genomes, and remains very limited compared to sequence data.

compbio.unm.edu

Laboratory for
High-Performance Algorithm Engineering
and Computational Molecular Biology

Includes all publications by our lab, GRAPPA source files, email addresses, and links to our main collaborators.