PPOSS Talk, Nov 2, 2023

Tandy Warnow
University of Illinois at Urbana-Champaign
Phylogeny (evolutionary tree)

From the Tree of the Life Website,
University of Arizona
Phylogenomics

Phylogeny + genomics = genome-scale phylogeny estimation
Estimating the Tree of Life

Basic Biology:
How did life evolve?

Applications of phylogenies to:
protein structure and function
population genetics
human migrations
metagenomics

Figure from
https://en.wikipedia.org/wiki/Common_descent
Estimating the Tree of Life

Large datasets!
- Millions of species
- Thousands of genes

NP-hard optimization problems
- Exact solutions infeasible
- Approximation algorithms
- Heuristics
- Multiple optima

High Performance Computing:
- Necessary
- But not sufficient

Figure from https://en.wikipedia.org/wiki/Common_descent
Goal of this PPoSS project (wrt Phylogeny)

• Phylogeny estimation can be seen as a statistical estimation problem.

• We want fast and accurate methods that are scalable to large datasets (thousands to hundreds of thousands of species, and genome-scale data).

• We also want these methods to have statistical guarantees (provably statistically consistent).

• The basic technique we will use is divide-and-conquer: using the best methods on subsets.
“Boosters”, or “Meta-Methods”

• Meta-methods use divide-and-conquer and iteration (or other techniques) to “boost” the performance of base methods (phylogeny reconstruction, alignment estimation, etc)
Today’s *Fast* Intro to Phylogenetics Research

- Models of evolution, identifiability, statistical consistency
- Trees, additive matrices, and chordal graphs
- Divide-and-conquer phylogeny estimation: overlapping vs disjoint subsets
- Genome-scale phylogeny:
  - Incomplete lineage sorting and species tree estimation under the Multi-Species Coalescent model (MSC)
  - ASTRAL: non-parametric accurate and statistically consistent species tree estimation under the MSC
  - TreeMerge/GTM: scaling species tree methods to large datasets
Phyogenomic Pipeline

• Assemble and annotate genomes (e.g., determine orthologs)
• Compute multiple sequence alignments of individual loci
• Construct gene trees
• Construct species tree
• Perform post-tree analyses (e.g., estimate dates, infer selection, etc.)
Phylogenomic Pipeline

• Assemble and annotate genomes (e.g., determine orthologs)
• Compute multiple sequence alignments of individual loci
• Construct gene trees
• Construct species tree
• Perform post-tree analyses (e.g., estimate dates, infer selection, etc.)
DNA Sequence Evolution (Idealized)
Phylogeny Problem

AGGGCAT
TAGCCCA
TAGACTT
TGCACAA
TGCGCTT

U
V
W
X
Y

U
V
W
X
Y
Markov Models of Sequence Evolution

The different sites are assumed to evolve \textit{i.i.d.} down the model tree

Simplest site evolution model (Jukes-Cantor, 1969):

- The model tree $T$ is binary and has substitution probabilities $p(e)$ on each edge $e$, with $0 < p(e) < 3/4$.
- The state at the root is randomly drawn from \{A,C,T,G\} (nucleotides)
- If a site (position) changes on an edge, it changes with equal probability to each of the remaining states.
- The evolutionary process is Markovian.

More complex models (such as the Generalized Time Reversible model) are also considered, often with little change to the theory.
FN: false negative
(missing edge)
FP: false positive
(incorrect edge)

50% error rate
Is method M statistically consistent under model G?

Amount of data generated under model G and then given to method M as input

Error in species tree inferred by method M

Question answered by mathematical proof
Questions

• Is the model tree identifiable?

• Which estimation methods are statistically consistent under this model?

• How much data does the method need to estimate the model tree correctly (with high probability)?

• What are the computational issues?
Answers for Gene Tree Evolution?

• We know a lot about which site evolution models are identifiable, and which methods are statistically consistent.
  • Maximum Likelihood statistically consistent, but NP-hard (good heuristics)
  • Distance-based methods also statistically consistent and typically polynomial time, but generally less accurate than maximum likelihood

• We know a little bit about the sample complexity (i.e. sequence length requirements) for standard methods.
  • Maximum likelihood has optimal sample complexity, standard distance-based methods do not

*Take home message:* maximum likelihood preferred, even though hard to find good solutions
Genome-scale data?
Phylogeny + genomics = genome-scale phylogeny estimation
Gene tree discordance

Incomplete Lineage Sorting (ILS) is a dominant cause of gene tree heterogeneity
Gene Trees inside the Species Tree (Multi-Species Coalescent)

Gorilla and Orangutan are not siblings in the species tree, but they are in the gene tree.
Gene Trees inside the Species Tree (Multi-Species Coalescent)

Gorilla and Orangutan are not siblings in the species tree, but they are in the gene tree.

Deep coalescence = INCOMPLETE LINEAGE SORTING (ILS): gene tree can be different from the species tree.
1KP: Thousand Transcriptome Project

- 103 plant transcriptomes, 400-800 single copy “genes”
- Next phase will be much bigger
- Wickett, Mirarab et al., *PNAS* 2014

**Major Challenge:**
- Massive gene tree heterogeneity consistent with ILS
Avian Phylogenomics Project

Erich Jarvis, HHMI
MTP Gilbert, Copenhagen
Guojie Zhang, BGI
Siavash Mirarab, Texas
Tandy Warnow, Texas and UIUC

• Approx. 50 species, whole genomes
• 14,000 loci
• Multi-national team (100+ investigators)
• 8 papers published in special issue of Science 2014

Major challenge:
• Massive gene tree heterogeneity consistent with ILS.
Hierarchical Model: MSC+GTR

- Multi-locus data, generated by a hierarchical model
  - Species tree generates gene trees under Multi-Species Coalescent (MSC)
  - Gene trees generate sequences under the Generalized Time Reversible (GTR) model
Gene evolution model

Sequence evolution model

ACTGCACACCG
ACTGC-CCCCG
AATGC-CCCCG
-CTGCACACGG
CTGAGCATCG
CTGAGC-TCG
ATGAGC-TC-
CTGAC-CACGG
AGCAGCATCGTG
AGCAGC-TCGTG
AGCAGC-TC-TG
C-TA-CACGGTG
CAGGCACGCACGAA
AGC-CACGC-CATA
ATGGCACGC-C-TA
AGCTAC-CACGGAT

Species tree

Gorilla  Human  Chimp  Orangutan
Gene evolution model

Species tree

Gene tree

Sequence evolution model

Sequence data (Alignments)

ACTGACACCCG
ACTGACCCCCG
AATGACCCCCG
-CTGACACCCG

CTGACATCG
CTGAGTCG
ATGAGTC-
CTGACACCGG

AGCAGCATCGTG
AGCAGTCGTG
AGCAGTC-TG
AGCTAC-CACGGAT

CAGGCACGCACGAA
AGC-CACGC-CATA
ATGGCACGC-C-TA
AGCTAC-CACGGAT
Step 1: infer gene trees (traditional methods)

ACTGCACACCG
ACTGC-CCCCCG
AATGC-CCCCCG
-CTGCACACGG

CTGAGCATCG
CTGAGC-TCG
ATGAGC-TC-
CTGA-CAC-G

AGCAGCATCGTG
AGCAGC-TCGTG
AGCAGC-TC-TG
C-TA-CACGGTG

CAGGCACGCACGAA
AGC-CACGC-CATA
ATGGCACGC-C-TA
AGCTAC-CACGGAT

Step 2: infer species trees

Gene tree
Gene tree
Gene tree
Gene tree
Suppose we can estimate all the gene trees correctly.

Can we estimate the species trees from lots of true gene trees?

**Step 1: infer gene trees (traditional methods)**

- ACTGCACACCG
- ACTGC-CCCCCG
- AATGC-CCCCC
- -CTGCACACCG
- CTGAGCATCG
- CTGAGC-TCG
- ATGAGC-TC-
- CTGA-CAC-G
- AGCAGCATCGTG
- AGCAGC-TCGTG
- AGCAGC-TC-TG
- C-TA-CACGGTG
- CAGGCACGCACGAA
- AGC-CACGC-CATA
- ATGGCACGC-C-TA
- AGCTAC-CACGGAT

**Step 2: infer species trees**
How to estimate a 4-leaf species tree

Theorem (Allman et al.): Under the multi-species coalescent model, for any four taxa A, B, C, and D, the most probable unrooted gene tree on \{A, B, C, D\} is identical to the unrooted species tree induced on \{A, B, C, D\}.
How to estimate a 4-leaf species tree

Theorem (Allman et al.): Under the multi-species coalescent model, for any four taxa A, B, C, and D, the most probable unrooted gene tree on \{A,B,C,D\} is identical to the unrooted species tree induced on \{A,B,C,D\}.
Species tree estimation from unrooted gene trees

Corollary: Under the multi-species coalescent model, the species tree is identifiable from the gene tree distribution
Species tree estimation from unrooted gene trees

Corollary: Under the multi-species coalescent model, the species tree is identifiable from the gene tree distribution

Proof: For every four species, select most frequently observed tree as the species tree. Then combine quartet trees!
ASTRAL
[Mirarab, et al., ECCB/Bioinformatics, 2014]

• Optimization Problem (NP-Hard):

Find the species tree with the maximum number of induced quartet trees shared with the collection of input gene trees

\[ \text{Score}(T) = \sum_{t \in \mathcal{T}} |Q(T) \cap Q(t)| \]

• Theorem: Statistically consistent under the multi-species coalescent model when solved exactly
ASTRAL

• Statistically consistent under the MSC, and runs in polynomial time
• Solves constrained version of the NP-hard Maximum Quartet Support problem using dynamic programming
  • Input: Gene trees and set X of allowed bipartitions
  • Output: Species tree T that maximizes the quartet support criterion, subject to drawing its bipartitions from the set X
Tree accuracy when varying the number of species

1000 genes, “medium” levels of recent ILS
ASTRAL on biological datasets

- 1KP: **103** plant species, 400-800 genes
- Yang, et al. **96** Caryophyllales species, 1122 genes
- Dentinger, et al. **39** mushroom species, 208 genes
- Giarla and Esselstyn. **19** Philippine shrew species, 1112 genes
- Laumer, et al. **40** flatworm species, 516 genes
- Grover, et al. **8** cotton species, 52 genes
- Hosner, Braun, and Kimball. **28** quail species, 11 genes
- Simmons and Gatesy. **47** angiosperm species, 310 genes
- Prum et al. **198** avian species, 259 genes
ASTRAL – great, but...

• The good: ASTRAL is
  • increasingly used in practice
  • statistically consistent given true gene trees
  • sometimes more accurate than concatenation, but impacted by gene tree estimation error
  • very fast for many datasets (faster than concatenation)

• The bad: ASTRAL can fail to complete on large enough datasets within reasonable time frames (days of computation)
The alternatives are worse

• Concatenation Analyses (e.g., using RAxML):
  • most commonly used method, not statistically consistent, sometimes more accurate than summary methods
  • computationally intensive (e.g., 250 CPU years for the Avian Phylogenomics project with only 48 species) and do not scale to large numbers of species

• Co-estimation of gene trees and species trees: too expensive

• Other statistically consistent methods: not as accurate as ASTRAL
Goal of this PPoSS project (wrt Phylogeny)

• Phylogeny estimation can be seen as a statistical estimation problem.
• We want fast and accurate methods that are scalable to large datasets (thousands to hundreds of thousands of species, and genome-scale data).
• We also want these methods to have statistical guarantees (provably statistically consistent).
• The basic technique we will use is divide-and-conquer: using the best methods on subsets.
Divide-and-conquer using Chordal Graphs

- A matrix is additive if it equals path lengths in an edge-weighted tree
- Distances calculated in phylogenetics (from sequence data) converge to additive matrices, as the sequence length increases
- If we threshold an additive matrix, we obtain a chordal graph: one that has no simple cycles of size four or larger
- Chordal graphs have lovely properties
  - Can list all maximal cliques in polynomial time
  - Minimum vertex separators are maximal cliques
  - Can obtain decompositions into overlapping subsets, and employ in divide-and-conquer strategies
Divide-and-conquer using Chordal Graphs

• Chordal graphs have lovely properties
  • Can list all maximal cliques in polynomial time
  • Minimum vertex separators are maximal cliques
  • Can obtain decompositions into overlapping subsets, and employ in divide-and-conquer strategies
• If we do this, we need methods that combine overlapping subset trees, i.e., “supertree” methods
• These approaches have not been as scalable as needed.
Decompose species set into pairwise disjoint subsets.

Build a tree on each subset.

Compute tree on entire set of species using “Disjoint Tree Merger” method.

Auxiliary Info (e.g., distance matrix)

Note: use most accurate method on subsets, and treat as absolute constraints.
Divide-and-Conquer using Disjoint Tree Mergers

Decompose species set into pairwise disjoint subsets.

Full species set

Auxiliary Info (e.g., distance matrix)

Tree on full species set

Compute tree on entire set of species using “Disjoint Tree Merger” method

Build a tree on each subset

TreeMerge or GTM

ASTRAL

Note: use most accurate method on subsets, and treat as absolute constraints
Divide-and-Conquer using Disjoint Tree Mergers

Decompose species set into pairwise disjoint subsets.

Build a tree on each subset.

Compute tree on entire set of species using "Disjoint Tree Merger" method.

Theorem: TreeMerge+ASTRAL and GTM+ASTRAL are both statistically consistent and polynomial time.

Note: use most accurate method on subsets, and treat as absolute constraints.
Guide Tree Merger

- Smirnov and Warnow, RECOMB-Comparative Genomics
- Guide Tree Merger (GTM): Another Disjoint Tree Merger method... unlike TreeMerge, it does not allow blending
- Github site: https://github.com/vlasmirnov/GTM

Algorithmic strategy:
- divide species set into disjoint subsets,
- compute species trees on the subsets using selected species tree method, and
- connect subset trees by adding edges (no blending!), so as to minimize distance to the given guide tree (polynomial time!)
ASTRAL+GTM: better than ASTRAL!
### Table 4  Comparison of average runtime (seconds) of NJst-ASTRAL-GTM and ASTRAL for high ILS conditions with introns on 1000 species

<table>
<thead>
<tr>
<th></th>
<th>NJst-ASTRAL-GTM</th>
<th>ASTRAL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>10 Genes (n=18)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Pre-GTM</td>
<td>97.4</td>
<td>n.a.</td>
</tr>
<tr>
<td>-ASTRAL</td>
<td>n.a.</td>
<td>8,617.0</td>
</tr>
<tr>
<td>-GTM</td>
<td>0.4</td>
<td>n.a.</td>
</tr>
<tr>
<td>-Total</td>
<td>97.8</td>
<td>8,656.0</td>
</tr>
<tr>
<td><strong>25 Genes (n=20)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Pre-GTM</td>
<td>174.7</td>
<td>n.a.</td>
</tr>
<tr>
<td>-ASTRAL</td>
<td>n.a.</td>
<td>5,441.4</td>
</tr>
<tr>
<td>-GTM</td>
<td>0.4</td>
<td>n.a.</td>
</tr>
<tr>
<td>-Total</td>
<td>175.1</td>
<td>5,539.4</td>
</tr>
<tr>
<td><strong>1000 Genes (n=16)</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-Pre-GTM</td>
<td>7,948.9</td>
<td>n.a.</td>
</tr>
<tr>
<td>-ASTRAL</td>
<td>n.a.</td>
<td>149,145.9</td>
</tr>
<tr>
<td>-GTM</td>
<td>0.4</td>
<td>n.a.</td>
</tr>
<tr>
<td>-Total</td>
<td>7,949.3</td>
<td>153,045.9</td>
</tr>
</tbody>
</table>
Main competing approaches

<table>
<thead>
<tr>
<th>Species</th>
<th>gene 1</th>
<th>gene 2</th>
<th>...</th>
<th>gene k</th>
</tr>
</thead>
</table>

- Concatenation
- Analyze separately
- Summary Method
Traditional approach: concatenation

- Statistically **inconsistent** and can even be positively misleading (proved for unpartitioned maximum likelihood) [Roch and Steel, Theo. Pop. Gen., 2014]

The GTM pipeline used with RAxML on subsets matches accuracy with RAxML, but is much faster.

<table>
<thead>
<tr>
<th>Table 6</th>
<th>Average runtime (seconds) of FastTree-RAxML-GTM (GTM(RAxML)) and RAxML on 1000-species exon datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GTM(RAxML)</td>
</tr>
<tr>
<td><strong>Low ILS 10 Genes (n=19)</strong></td>
<td></td>
</tr>
<tr>
<td>- FastTree</td>
<td>279.6</td>
</tr>
<tr>
<td>- RAxML subtrees</td>
<td>831.3</td>
</tr>
<tr>
<td>- GTM</td>
<td>0.4</td>
</tr>
<tr>
<td>- Total</td>
<td>1,111.3</td>
</tr>
<tr>
<td><strong>Low ILS 25 Genes (n=10)</strong></td>
<td></td>
</tr>
<tr>
<td>- FastTree</td>
<td>686.3</td>
</tr>
<tr>
<td>- RAxML subtrees</td>
<td>1,460.6</td>
</tr>
<tr>
<td>- GTM</td>
<td>0.4</td>
</tr>
<tr>
<td>- Total</td>
<td>2,147.3</td>
</tr>
<tr>
<td><strong>High ILS 10 Genes (n=12)</strong></td>
<td></td>
</tr>
<tr>
<td>- FastTree</td>
<td>283.7</td>
</tr>
<tr>
<td>- RAxML subtrees</td>
<td>637.5</td>
</tr>
<tr>
<td>- GTM</td>
<td>0.4</td>
</tr>
<tr>
<td>- Total</td>
<td>921.6</td>
</tr>
<tr>
<td><strong>High ILS 25 Genes (n=20)</strong></td>
<td></td>
</tr>
<tr>
<td>- FastTree</td>
<td>731.5</td>
</tr>
<tr>
<td>- RAxML subtrees</td>
<td>1,363.1</td>
</tr>
<tr>
<td>- GTM</td>
<td>0.4</td>
</tr>
<tr>
<td>- Total</td>
<td>2,095</td>
</tr>
</tbody>
</table>

The value for n is the number of replicates being compared, i.e., where a RAxML tree is available.
Summary about phylogenetic tree estimation

• The best tree estimation methods are computationally intensive, and tree-space grows exponentially.

• The Divide-and-Conquer pipelines we are developing (especially GTM) maintain statistical consistency, maintain or improve accuracy and are much faster.

• In addition, they naturally enable parallel implementations.
What about Community Detection?

- I also work on community detection in large networks, largely in the context of Scientometrics.
- Our recent paper (Park et al., Complex Networks 2023) addresses failure of standard community detection methods (aka clustering methods) to produce well-connected clusters.
- See [https://tandy.cs.illinois.edu/bibliometrics.html](https://tandy.cs.illinois.edu/bibliometrics.html) for papers.
## Networks we studied

<table>
<thead>
<tr>
<th>network</th>
<th>nodes</th>
<th>edges</th>
<th>avg_deg</th>
<th>ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open Citations</td>
<td>75,025,194</td>
<td>1,363,605,603</td>
<td>36.35</td>
<td>(17)</td>
</tr>
<tr>
<td>CEN</td>
<td>13,989,436</td>
<td>92,051,051</td>
<td>13.16</td>
<td>(35)</td>
</tr>
<tr>
<td>cit_hepph</td>
<td>34,546</td>
<td>420,877</td>
<td>24.37</td>
<td>(36)</td>
</tr>
<tr>
<td>cit_patents</td>
<td>3,774,768</td>
<td>16,518,947</td>
<td>8.75</td>
<td>(36)</td>
</tr>
<tr>
<td>orkut</td>
<td>3,072,441</td>
<td>117,185,083</td>
<td>76.28</td>
<td>(37)</td>
</tr>
<tr>
<td>wiki_talk</td>
<td>2,394,385</td>
<td>4,659,565</td>
<td>3.89</td>
<td>(38)</td>
</tr>
<tr>
<td>wiki_topcats</td>
<td>1,791,489</td>
<td>25,444,207</td>
<td>28.41</td>
<td>(39)</td>
</tr>
</tbody>
</table>

Table 1: Summary statistics for networks used in this study. Average degree is the average of the node degrees across the network.
Many small edge cuts in Leiden clusters on real-world networks

Leiden optimizing the Constant Potts Model (CPM) or modularity (mod)

Results using other clustering methods are similar

Figure 1: Node coverage, connectivity, and size distribution of clusters generated by Leiden optimizing either CPM or modularity on the Open Citations network (75,025,194 nodes). Con-
The Connectivity Modifier (CM) Pipeline

**Defaults used in our study:** all clusters have min cuts above \( \log_{10} n \), where \( n \) is cluster size, and have size at least 11

---

**Figure 3: Connectivity Modifier Pipeline Schematic.** The four-stage pipeline depends on user-
Impact of the Connectivity Modifier

CPM clustering is impacted by the resolution parameter: small values give high node coverage, but many of these clusters are poorly connected (even trees).

Modularity-optimization is similar to CPM with a small resolution parameter.

Using CM reduces node coverage.

Figure 4: Reduction in node coverage after CM treatment of Leiden clusters. The Open Citations (left panel) and CEN (right panel) networks were clustered using the Leiden algorithm under CPM at five different resolution values or modularity. Node coverage (defined as the percentage of nodes in cluster of size at least 2) was computed for Leiden clusters * (lime green), Leiden clusters with trees and/or clusters of size 10 or less filtered out * (soft orange), and after CM treatment of filtered clusters * (desaturated blue).
The CM pipeline improves accuracy

Results for NMI accuracy on LFR networks. Results for other criteria are similar.
Summary

• The tendency for standard clustering methods to have poorly connected clusters (or else have low node coverage) is striking.
• CM ensures that all returned clusters are well-connected, according to the user specified bound
• CM improves accuracy on LFR networks
• But after CM, there is a drop in node coverage that can be large.
• How do we explain the drop in node coverage?
  • Perhaps not the case that the entire network is covered by communities?