Species tree inference and update on very large datasets using approximation, randomization, parallelization, and vectorization

Siavash Mirarab

Electrical and Computer Engineering
University of California at San Diego
Phylogenetic reconstruction from data

Gorilla
ACTGCACACCG

Human
ACTGCCCCCG

Chimpanzee
AATGCCCCCG

Orangutan
CTGCACACCG
Phylogenetic reconstruction from data
Phylogenetic reconstruction from data

Gorilla
ACTGCACACCG

Human
ACTGCCCCCG

Chimpanzee
AATGCCCCCG

Orangutan
CTGCACACCG
Phylogenetic reconstruction from data

Gorilla
ACTGCACACCG

Human
ACTGCCCCCG

Chimpanzee
AATGCCCCCG

Orangutan
CTGCACACGG

\[ D \]
Phylogenetic reconstruction from data

Orangutan
ACTGCACACCG

Gorilla
ACTGCACACCG

Human
ACTGCCCCCG

Chimpanzee
AATGCCCCCG

Orangutan
CTGCACACCG

\[ D \]

\[ P(D|T) \]

\[ T \]
Applications: HIV forensic

Scaduto et al., PNAS, 2010
Applications: microbiome

Applications: microbiome


Applications: food safety

Tracking the source of a listeriosis outbreak

Applications: ebola

Fig. 3. Molecular dating of the 2014 outbreak. (A) BEAST dating of the separation of the 2014 lineage from Middle African lineages (SL = Sierra Leone; GN = Guinea; DRC = Democratic Republic of Congo; tMRCA: Sep 2004, 95% HPD: Oct 2002 - May 2006). (B) BEAST dating of the tMRCA of the 2014 West African outbreak (tMRCA: Feb 23, 95% HPD: Jan 27 - Mar 14) and the tMRCA of the Sierra Leone lineages (tMRCA: Apr 23, 95% HPD: Apr 2 - May 13); probability distributions for both 2014 divergence events overlayed below. Posterior support for major nodes is shown.

source: Gire et al., Science, 2014
Sequence data growth

- Rapid growth in the available sequences

**Sequence data growth**

- Rapid growth in the available sequences
- Our focus has shifted to “whole genomes”

More genomic regions

More sequences

A million rows

Billions of columns
Phylogenetic reconstruction: a computational nightmare

• Almost all problems are NP-Hard!

• The search space is huge.
  
  • Focusing on topology alone, there are \((2n-5)!!\) trees with \(n\) leaves

  • We also care about branch lengths: \(\mathbb{R}^{2n-3}\)

• We are interested in \(n\) in \(10^1\ldots10^6\) range
Dealing with uncertainty

- Really no “experimental” way to validate results. Thus, we use computation-heavy procedures to calculate statistical support
Dealing with uncertainty

• Really no “experimental” way to validate results. Thus, we use computation-heavy procedures to calculate statistical support

• Genome evolution is complex. We need complex statistical models for accurate inference
Dealing with uncertainty

• Really no “experimental” way to validate results. Thus, we use computation-heavy procedures to calculate statistical support.

• Genome evolution is complex. We need complex statistical models for accurate inference.

• We need extensive simulations to test methods.
To scale to large datasets ...

- Approximate and heuristic solutions
To scale to large datasets ...

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
To scale to large datasets ...

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.
How about accuracy?

Increased data *can* make problems easier, but …
How about accuracy?

Increased data *can* make problems easier, but …

- Larger datasets often
  - Are used to answer harder problems
  - Allow the use of more complex models
  - Riddled with erroneous data
How about accuracy?

Increased data *can* make problems easier, but …

- Larger datasets often
  - Are used to answer harder problems
  - Allow the use of more complex models
  - Riddled with erroneous data
- Often, methods lose their accuracy on large datasets
Examples of improving scalability
To scale to large datasets

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.
ASTRAL

- Optimization problem (NP-Hard):
  
  Find the median tree among a set of input trees

  Distance between two trees := the number of quartet trees they do not share
ASTRAL

• Optimization problem (NP-Hard):

  Find the median tree among a set of input trees

  Distance between two trees := the number of quartet trees they do not share

• ASTRAL: an exact solution using a dynamic programming algorithm

  • Solves the problem exactly on a constrained search space
Choosing Constraints

- Restricted search space should
  - Not compromise statistical consistency
  - Be large enough that accuracy is not sacrificed
  - Be small for scalability
ASTRAL evolution

- Search space should ideally grow polynomially with the dataset size
  - Tandy talked about ASTRAL-I and ASTRAL-II, which do not such have guarantees
  - ASTRAL-III [Zhang et al., 2018]
    - Guarantees polynomial size search space
    - Increases the speed of the dynamic programming
Changing the number of species ($n$)

Simulations: $n = 200 \text{ to } 10,000$, $k = 1000$ gene trees

Empirical running time seems to increase close to $O(n^{1.8})$

[Zhang et al., 2018]
Changing the number of genes ($k$)

Simulations: $n = 48$ species, $k = 256$ to $16,384$ gene trees

Empirical running time seems to increase close to $O(k^2)$

[Zhang et al., 2018]
To scale to large datasets

• Approximate and heuristic solutions
• Make the problem easier
  • Divide-and-conquer
  • Constrained search
• Develop optimized code.
Profiling ASTRAL-III

Many species

Many genes
Further scaling improvements

• Developed a randomized algorithm for a key step in the dynamic programming.

For a set $X$ of subsets of an alphabet, find:

$$\{(A, B) | A, B \in X, A \cup B \in X, A \cap B = \emptyset\}$$

• Using, Abelian group theory, we can do compute this in time quadratic in $|X|$.

• Has an astronomically low probability of failure
Further scaling improvements

• Developed a randomized algorithm for a key step in the dynamic programming.

For a set $X$ of subsets of an alphabet, find:

$$\{(A, B) \mid A, B \in X, A \cup B \in X, A \cap B = \emptyset\}$$

• Using, Abelian group theory, we can do compute this in time quadratic in $|X|$.

• Has an astronomically low probability of failure

• Implemented the kernel in C++ and added vectorization.
Improved scalability

Many species

Many genes
Parallelize the main step of the dynamic programming (blue)

Many species

Many genes
Scaling + GPU

- Can analyze datasets with 10,000 species and 1000 genes in less than a day given 24 cores and a GPU

- https://github.com/smirarab/ASTRAL/tree/MP-similarity

---

Diagram showing speedup versus CPU cores with and without a GPU for Many genes.
• 10,000 microbial species and 381 genes

• ASTRAL infers the tree in 24 hours (4 GPUs)
To scale to large datasets

- Approximate and heuristic solutions
- Make the problem easier
  - Divide-and-conquer
  - Constrained search
- Develop optimized code.
Statistical support

- Traditional approach: **bootstrapping** each gene, then bootstrapping species tree
- **Slow**: requires bootstrapping all genes (e.g., 100 times slower)
Statistical support

- Traditional approach: bootstrapping each gene, then bootstrapping species tree
  - **Slow**: requires bootstrapping all genes (e.g., 100 times slower)
  - **Inaccurate** and hard to interpret
    [Mirarab et al., Sys bio, 2014; Bayzid et al., PLoS One, 2015]
Local posterior probability

- Quartet frequencies follow a multinomial distribution

\[ m_1 = 80 \quad m_2 = 63 \quad m_3 = 57 \]
Local posterior probability

• Quartet frequencies follow a multinomial distribution

\[ m_1 = 80 \quad m_2 = 63 \quad m_3 = 57 \]

\[ \begin{align*}
\theta_1 & \quad \text{Chimp} \quad \text{Gorilla} \quad \text{Orang.} \\
\theta_2 & \quad \text{Human} \quad \text{Orang.} \\
\theta_3 & \quad \text{Human} \quad \text{Gorilla} \quad \text{Chimp}
\end{align*} \]

• \( P(\text{gene tree seen } m_1/m \text{ times } = \text{species tree}) = P(\theta_1 > 1/3) \)

• Solved analytically: “local posterior probability” (localPP)
Local posterior probability

- Quartet frequencies follow a multinomial distribution

\[
\begin{align*}
m_1 &= 80 \\
m_2 &= 63 \\
m_3 &= 57
\end{align*}
\]

- \( P(\text{gene tree seen } m_1/m \text{ times } = \text{ species tree}) = P(\theta_1 > 1/3) \)

- Solved analytically: “local posterior probability” (localPP)

- \( n > 4 \) leads to an exponentially growing number of cases

- **Approximate** by using averaged all quartet scores, which can be computed in time quadratic in \( n \).
Quartet support vs. localPP

**Background:** How many targets are enough?

The number of loci required for confidently resolving a phylogenetic relationship depends on its “hardness.” Discordant gene trees due to incomplete lineage sorting (ILS) can make branches of a species tree very difficult to resolve. ILS is a function of the branch length and population size. This means that shorter branches or larger population sizes increase ILS, and can make species tree reconstruction more difficult. The co-PI has recently developed a new approach for estimating the support of a branch in a coalescent-based framework based on the proportion of gene trees that support quartet topologies in gene trees \[31\]. This new approach is analogous to finding the probability that a three-sided die is loaded towards each facet by observing outcomes of many tosses. We can also ask the reverse question: if we have an estimate of the degree of bias of a die, how many tosses are needed to confidently find its loaded side. Similarly, this approach \[31\] sheds light on the relationship between the number of genes required to achieve a level of support for a branch of a certain length in coalescent units (Fig. 3). While the co-PI’s method of calculating support, which is implemented in ASTRAL \[32, 33\], gives the basic mathematical framework, more method development is needed (see below).

**Research Approach**

**Specimens**

Suitable specimens of more than 80 species of Sabellidae and >120 Terebelliformia species have been collected from localities around the world in the last 15 years by the PI. Further collecting will be undertaken for some key taxa for a few more transcriptomes and for targeted DNA capture. We also have agreements from other experts in Sabellidae and Terebelliformia to provide other needed ethanol-preserved specimens for targeted capture. We will obtain ~ 50% of the known sabellid diversity and ~25% of terebelliforms for targeted capture sequencing (~500 species total). Our plan is to sample all genera, particularly focusing on their type species. This should allow for the generation of robust phylogenies, from which we will revise the taxonomy. Specimens will be vouchered at the SIO Benthic Invertebrate Collection and biodiversity information curated on the Encyclopedia of Life.

**Sequencing (transcriptomes and targeted capture of DNA)**

A targeted capture approach will be used with an appropriate number of loci (see below) to generate robust phylogenies of Sabellidae and Terebelliformia. Two different sets of targets will be designed from transcriptomes, across Sabellidae and Terebelliformia respectively. We have already generated new transcriptomes for nine sabellids, a serpulid and a fabriciid (bold terminals in Fig. 4), and nine Terebelliformia (bold terminals in Fig. 5) and combined this data with the few publicly available transcriptomes. Based on direct sequencing data already obtained for several genes for Sabellidae and Terebelliformia \[Rouse in prep.; Stiller et al. in prep.\], our sampling arguably spans the extant diversity of each clade. Several more transcriptomes will be needed to ensure the targeted capture methods will be robust. Methods for generating these transcriptomes will follow those previously used by us \[6, 34\].

**How many targets are needed?**

A targeted capture pipeline typically starts from a large number of loci sequenced from a small number of species using genome-wide approaches (e.g., transcriptomics) \[7, 8, 35\]. This initial dataset is then used to select a smaller subset of loci for the targeted capture phase, which involves a larger set of species. To reduce the cost and effort, one would want to minimize the number of loci in the second phase, as long as sufficient loci are selected to confidently resolve relationships of interest. To date the number of loci has been determined in an ad hoc manner, either by the ultra conserved elements discovered \[7, 8\] or limitations of the targeted capture technology \[13\]. Initial analyses of our two datasets demonstrates how the number of

![Graph showing the relationship between quartet frequency and local posterior probability for different numbers of genes.](image)

**Figure 3. Impact of number of genes (colors) and amount of ILS (x-axis) on support (posterior probability) for a branch (y-axis).** Under the coalescent model, for four species separated with an unrooted branch of length \(d\), the probability of a gene tree being identical to the species tree is \(1 - \frac{2}{3}e^{-d}\) (we use this as a measure of ILS). We measure support using local pp \[31\] for various numbers of gene trees (assuming no gene tree estimation error). More ILS (ie., low \(1 - \frac{2}{3}e^{-d}\)) requires more genes to get high support.

- Increased number of genes \(\Rightarrow\) increased support
- Increased discordance \(\Rightarrow\) Reduced support
localPP is more accurate than bootstrapping

Avian simulated dataset (48 taxa, 1000 genes)  
[Sayyari and Mirarab, MBE, 2016]
To scale to large datasets

• Approximate and heuristic solutions

• Make the problem easier
  • Divide-and-conquer
  • Constrained search

• Develop optimized code.
Phylogenetic placement

```
ACCG
CGAG
CGGT
GGCT
TAGA
GGAG
GcTT
•
•
•
ACCT
```

Query sequences

A backbone dataset

- Escherichia coli
- Salmonella enterica
- Salmonella bongori
- Yersinia pestis
- Vibrio cholerae
Phylogenetic placement

Applications:
- Place sequences of unknown origins on a reference tree of known sequences (a major goal in microbiome analyses)
- Update an existing tree quickly without recomputing
Why placement?

- Placement is easier than *de novo* inference
- Placement is usually *sufficient* for downstream applications
  - Sometimes, placement is *more accurate*.
- Placement is *embarrassingly parallel*
SEPP: placement for microbiome data

- Uses divide-and-conquer to align and place on very large backbone trees
- Useful for identifying microbiome data
SEPP: placement for microbiome data

- Uses divide-and-conquer to align and place on very large backbone trees
  - Useful for identifying microbiome data
  - Has better accuracy than \textit{de novo} inference of the tree
    - When inferring from fragmentary data

[Janssen, mSystems, 2018]
Placement algorithms:
State of the art (ML) is memory-hungry

ML
(used inside SEPP)
Placement algorithms: State of the art (ML) is memory-hungry

ML (used inside SEPP)

Minimum least square error on sequence distances (our new method)
Distance-based method is also much faster than ML
The distance-based method is equally or more accurate
The distance-based method is equally or more accurate.
How about placing on species trees?

• Once new data are added to gene trees, we want to update the species tree
How about placing on species trees?

- Once new data are added to gene trees, we want to update the species tree
- INSTRAL: a (worst-case) quadratic-time ASTRAL-like method to update species trees
  - Finds the best place almost always (>99.9%)
How about placing on species trees?

• Once new data are added to gene trees, we want to update the species tree

• INSTRAL: a (worst-case) quadratic-time ASTRAL-like method to update species trees
  • Finds the best place almost always (>99.9%)
  • Sub-quadratic running time in practice

[Rabiee, biorxiv, 2018]
Examples of concerns with accuracy
But how about accuracy?

Increased data *can* make problems easier, but …

- Larger datasets often
  - Are used to answer harder problems
  - Allow the use of more complex models
  - Riddled with erroneous data
- Often, methods lose their accuracy on large datasets
Errors abound in phylogenomic datasets

On the importance of homology in the age of phylogenomics
Mark S. Springer & John Gatesy
Pages 210-228 | Received 10 Jul 2017, Accepted 25 Oct 2017, Published online: 08 Dec 2017

Resolving Difficult Phylogenetic Questions: Why More Sequences Are Not Enough
Hervé Philippe, Henner Brinkmann, Dennis V. Lavrov, D. Timothy J. Littlewood, Michael Manuel, Gert Wörheide, Denis Baurain

Error, signal, and the placement of Ctenophora sister to all other animals
Nathan V. Whelan, Kevin M. Kocot, Leonid L. Moroz, and Kenneth M. Halanych
PNAS published ahead of print April 20, 2015 https://doi.org/10.1073/pnas.1503453112

Phylogenetic analysis at deep timescales: Unreliable gene trees, bypassed hidden support, and the coalescence/concatalescence conundrum
John Gatesy*, Mark S. Springer

The gene tree delusion ★
Mark S. Springer*, John Gatesy*
Errors show as long branches

Gatesy et. al. (2014)
Errors show as long branches

A Gene tree from Mammalian dataset
Song et al, PNAS, 2012
Errors show as long branches

Gatesy et. al. (2014)

A Gene tree from Mammalian dataset

Song et al., PNAS, 2012

Sphagnum lescurii

A Gene tree from 1kp Plants dataset

Wicket et al., PNAS, 2014
If we are to remove 1 taxon
“shrinkable”: $d_0/d_1 \approx 3.5$

If we are to remove 2 taxa
“shrinkable”: $d_1/d_2 \approx 1.1$
An optimization problem

The $k$-shrink problem:

- Given:
  - a tree with $n$ leaves and branch lengths
  - some $1 \leq k \leq n$

- Find:
  - for every $1 \leq i \leq k$:
    - the set of $i$ leaves that should be removed to reduce the tree diameter maximally
An optimization problem

The $k$-shrink problem:

- **Given:**
  - a tree with $n$ leaves and branch lengths
  - some $1 \leq k \leq n$

- **Find:**
  - for every $1 \leq i \leq k$:
    - the set of $i$ leaves that should be removed to reduce the tree diameter maximally

We have a polynomial time solution using dynamic programming

Uyen Mai
What to remove?

Let \( \nu_i = \frac{\text{the diameter after } i-1 \text{ removals}}{\text{the diameter after } i \text{ removals}} \)
What to remove?

![Graph showing removal ratio](image)
Running Time

- k-shrink can be solved in $O(k^2h+n)$ where $h$ = the tree height
- by default, we set $k=O(n^{0.5})$
- Fast: processes a tree of $n=203,452$ leaves with $k=2255$ in 28 mins
How much data?

• In biology, data is not free. Fundamental questions:
  • How much data is enough?
  • What’s the best strategy to obtain data?
How much data?

- In biology, data is not free. Fundamental questions:
  - How much data is enough?
  - What’s the best strategy to obtain data?
- Theoretical answers: sample complexity
  - For ASTRAL, we have bounds on the number of genes needed

  [Shekhar et al, TCBB, 2017]
How much data?

• In biology, data is not free. Fundamental questions:
  • How much data is enough?
  • What’s the best strategy to obtain data?

• Theoretical answers: sample complexity

  • For ASTRAL, we have bounds on the number of genes needed
    [Shekhar et al, TCBB, 2017]

• Practical:

  • For ASTRAL, more genes are better than more or individuals
    [Rabiee and Mirarab, MPE, 2018]

  • For phylogenetic placement, we showed a very low coverage of genome (too little for assembly) is more than enough
    [Sarmashghi, et. al., bioRxiv, 2018]
Code Optimization

• All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

• What’s the best measure of performance?
  • FLOPS?
Code Optimization

• All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

• What’s the best measure of performance?
  • FLOPS?
  • POPS ~ PPPD?
Code Optimization

• All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

• What’s the best measure of performance?
  • FLOPS?
  • POPS ~ PPPD?
  • Programs Optimized Per Student ~ Papers Published Per Dissertation
Code Optimization

• All methods shown here are in python, Java, etc. with little or no optimization (perhaps except ASTRAL)

• What’s the best measure of performance?
  • FLOPS?
  • POPS ~ PPPD?
    • Programs Optimized Per Student ~ Papers Published Per Dissertation
  • Ease of development matters in fast moving fields
More generally …

• Analyzing large datasets requires method development:

• Scalability:
  hardware + software + better algorithms

• Attention to accuracy and how properties of big data (e.g. data noisiness) affect accuracy