Supertree Estimation

Tandy Warnow

February 26, 2017
Today’s material:

- Review of Tree Compatibility (Chapter 3)
- Intro to supertree methods (Chapter 7)
Species tree estimation

Multiple challenges:

1. NP-hard problems on big datasets
2. Heterogeneity (different trees for different parts of genome)

Supertree estimation addresses the first problem, but the second problem is more challenging!
Supertree estimation

- Input: set \( \mathcal{T} \) of trees on subsets of \( S \), species set
- Output: tree \( T \) on full set \( S \), optimizing some criterion
Supertree optimization problems

- Find $T$ minimizing the Robinson-Foulds distance to the trees in $\mathcal{T}$
- Find $T$ minimizing the quartet distance to the trees in $\mathcal{T}$
- Represent every tree in $\mathcal{T}$ with a 0, 1-matrix (one column for every edge), concatenate the matrices, and then solve maximum parsimony
- Represent every tree in $\mathcal{T}$ with a 0, 1-matrix (one column for every edge), concatenate the matrices, and then solve CFN maximum likelihood
- Compute a matrix $M$ of average leaf-to-leaf distances between species, and find an additive matrix close to $M$ (minimizing some criterion)

All these problems are NP-hard.
Tree Compatibility

A set $\mathcal{T}$ of trees is said to be compatible if there is a supertree that induces each tree in $\mathcal{T}$.

**Tree Compatibility problem**: are the trees in $\mathcal{T}$ compatible?

- If all the trees are rooted, then the problem can be solved using Aho, Sagiv, Szymanski, and Ullman (Section 3.3 from textbook) in polynomial time.
- If the trees are unrooted, the problem is NP-hard (Section 3.5 from textbook).

Because unrooted tree compatibility is NP-hard, it is trivial to show that most optimization problems for supertree construction from unrooted source trees are NP-hard.

But they are still hard for supertree construction from rooted source trees!
Popular Supertree “methods”

- MRP: Matrix Representation with Parsimony
- MRL: Matrix Representation with Likelihood
- Robinson-Foulds Supertrees

All of these are NP-hard problems, so heuristics are used to find “good” solutions.
SuperFine: a supertree method “booster”

SuperFine (Swenson et al. 2012) is a meta-method for improving the speed and accuracy of supertree methods:

- Compute a constraint tree using the Strict Consensus Merger
- For each polytomy $v$ (node of degree $d > 3$):
  - Compute an encoding of the source trees into trees with at most $d$ leaves
  - Run preferred supertree method on the new source trees, obtaining $t(v)$
  - Refine polytomy $v$ with the computed supertree $t(v)$
Phase 1: Strict Consensus Merger (SCM)
Phase 2: Refining the SCM tree using MRP
SuperFine+MRP vs MRP, CA-ML, and other methods
Robinson-Foulds Supertrees

Finding the supertree $T$ that minimizes the Robinson-Foulds (RF) distance to the source trees is the Robinson-Foulds Supertree problem.

MulRF (Chaudhary et al., 2014) and PluMiST (Kupczok, 2011) are two methods for Robinson-Foulds Supertrees.

Robinson-Foulds Supertrees are (sort of) approximations to the Maximum Likelihood Supertree (Steel and Rodrigo, 2008) problem (see Bryant and Steel 2009 for more).
Constrained optimization

Bipartition-constrained Robinson-Foulds Supertree Problem:

- **Input:** Set $\mathcal{T}$ of source trees and set $X$ of bipartitions on $S$
- **Output:** Tree $T$ on $S$ that draws its bipartition set $C(T)$ from $X$, and that minimizes the RF distance to $T$ among all such supertrees

FastRFS (Vachaspati and Warnow, Bioinformatics 2016) solves this problem in polynomial time, using dynamic programming.
The CPL dataset has 2228 species, and so represents the largest and most difficult dataset; MulRF and PluMiST could not complete on it.
<table>
<thead>
<tr>
<th>Method</th>
<th>500</th>
<th>500</th>
<th>500</th>
<th>500</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaffold %</td>
<td>20</td>
<td>50</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td># Replicates</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>ASTRAL</td>
<td>15.3</td>
<td>14.8</td>
<td>12.7</td>
<td>11.2</td>
</tr>
<tr>
<td>ASTRAL-enhanced</td>
<td>14.8</td>
<td>14.1</td>
<td>12.6</td>
<td>11.2</td>
</tr>
<tr>
<td>ASTRID</td>
<td>26.0</td>
<td>50.1</td>
<td>45.4</td>
<td><strong>10.5</strong></td>
</tr>
<tr>
<td>MRL</td>
<td>15.4</td>
<td>14.3</td>
<td>12.1</td>
<td>11.2</td>
</tr>
<tr>
<td>MulRF</td>
<td>46.9</td>
<td>40.3</td>
<td>27.4</td>
<td>12.6</td>
</tr>
<tr>
<td>PluMiST</td>
<td>35.4</td>
<td>29.5</td>
<td>22.4</td>
<td>10.9</td>
</tr>
<tr>
<td>FastRFS-basic</td>
<td>14.5</td>
<td>14.3</td>
<td>12.4</td>
<td>11.1</td>
</tr>
<tr>
<td>FastRFS-enhanced</td>
<td><strong>14.3</strong></td>
<td><strong>13.9</strong></td>
<td><strong>12.0</strong></td>
<td>10.8</td>
</tr>
</tbody>
</table>

**Table**: Average supertree topology estimation error on simulated datasets.
Table 2. Supertree topology estimation error on simulated datasets, measured using the Robinson-Foulds error rate, expressed as a percentage

<table>
<thead>
<tr>
<th>Method</th>
<th>100</th>
<th>100</th>
<th>100</th>
<th>100</th>
<th>500</th>
<th>500</th>
<th>500</th>
<th>1000</th>
<th>1000</th>
<th>1000</th>
<th>1000</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scaffold %</td>
<td>20</td>
<td>30</td>
<td>75</td>
<td>100</td>
<td>20</td>
<td>50</td>
<td>75</td>
<td>100</td>
<td>20</td>
<td>50</td>
<td>75</td>
<td>100</td>
</tr>
<tr>
<td># Replicates</td>
<td>9</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>ASTRAL</td>
<td>11.7</td>
<td>14.0</td>
<td>11.6</td>
<td>10.0</td>
<td>15.3</td>
<td>14.8</td>
<td>12.7</td>
<td>11.2</td>
<td>16.9</td>
<td>15.7</td>
<td>13.6</td>
<td>11.6</td>
</tr>
<tr>
<td>ASTRAL-enhanced</td>
<td>11.8</td>
<td>13.1</td>
<td>11.5</td>
<td>10.0</td>
<td>14.8</td>
<td>14.1</td>
<td>12.6</td>
<td>11.2</td>
<td>16.3</td>
<td>15.1</td>
<td>13.5</td>
<td>11.6</td>
</tr>
<tr>
<td>ASTRID</td>
<td>15.8</td>
<td>18.7</td>
<td>17.1</td>
<td>9.6</td>
<td>26.0</td>
<td>50.1</td>
<td>45.4</td>
<td>10.5</td>
<td>35.6</td>
<td>58.1</td>
<td>52.0</td>
<td>11.2</td>
</tr>
<tr>
<td>MRL</td>
<td>13.6</td>
<td>13.6</td>
<td>11.2</td>
<td>10.8</td>
<td>15.4</td>
<td>14.3</td>
<td>12.1</td>
<td>11.2</td>
<td>17.4</td>
<td>15.1</td>
<td>13.5</td>
<td>12.2</td>
</tr>
<tr>
<td>MulRF</td>
<td>22.1</td>
<td>26.0</td>
<td>15.3</td>
<td>9.3</td>
<td>46.9</td>
<td>40.3</td>
<td>27.4</td>
<td>12.6</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>PluMiST</td>
<td>25.9</td>
<td>16.6</td>
<td>11.5</td>
<td>9.3</td>
<td>35.4</td>
<td>29.5</td>
<td>22.4</td>
<td>10.9</td>
<td>–</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>FastRFS-basic</td>
<td>13.5</td>
<td>14.3</td>
<td>10.5</td>
<td>9.1</td>
<td>14.5</td>
<td>14.3</td>
<td>12.4</td>
<td>11.1</td>
<td>17.3</td>
<td>15.6</td>
<td>13.5</td>
<td>12.0</td>
</tr>
<tr>
<td>FastRFS-enhanced</td>
<td>13.5</td>
<td>13.4</td>
<td>10.6</td>
<td>9.3</td>
<td>14.3</td>
<td>13.9</td>
<td>12.0</td>
<td>10.8</td>
<td>16.7</td>
<td>15.1</td>
<td>13.4</td>
<td>11.8</td>
</tr>
</tbody>
</table>

The best result for each model condition is boldfaced. No results are shown for PluMiST or MulRF on the 1000-taxon simulated datasets due to running time limitations for these methods. Results are averaged over the completed replicates.

![Fig. 2](https://academic.oup.com/bioinformatics/article-abstract/33/5/631/2525708)

Figure 2 shows running times (in seconds) on biological data of supertree methods. MulRF and PluMiST could not be run on the CPL dataset, due to its large size; hence no values are shown for those methods on that dataset.

The remaining methods (MRL, the two ASTRAL versions and the two FastRFS versions) were fairly close in accuracy. However, MRL was never more accurate than FastRFS-enhanced, and was only the top performing method for one model condition (where it tied with FastRFS-enhanced). ASTRAL-enhanced was more accurate than ASTRAL on 8 conditions, tied on 1 condition, and less accurate on 3 conditions. FastRFS-enhanced was more accurate than FastRFS-basic on 9 model conditions, tied on 1 condition, and worse on 2 conditions. FastRFS-enhanced was more accurate than ASTRAL-enhanced on 8 of the 12 model conditions, tied on 1 condition and worse on 3 conditions.

FastRFS-enhanced was the top performing method on 5 of the 12 model conditions; the next best performing method was ASTRAL-enhanced, which was the top performing method in 3 of the 12 model conditions. Thus, overall FastRFS-enhanced provided the best accuracy of the tested supertree methods. These results, and especially the pairwise comparisons, suggest that optimizing the Robinson-Foulds Supertree criterion (minimize RF distance) is better than optimizing the ASTRAL criterion (minimize quartet distance) for supertree estimation, and that adding bipartitions from MRL (and from ASTRID if its internode distance matrix is complete) also tends to improve accuracy.

3.5 Running time

Figure 2 shows running times on the biological datasets. MulRF and PluMiST took the most time, each typically requiring hours where FastRFS-basic, MRL and ASTRAL completed in well under a minute (and sometimes in just a few seconds). MRL and FastRFS-enhanced were the next most computationally intensive, but were sometimes fast, and finally ASTRAL, ASTRID and FastRFS-basic were the fastest, often completing in just seconds. As an example, the running times on the largest dataset on which all the methods completed (THPL, with 538 taxa) showed substantial differences between methods: PluMiST used 86 400 s (i.e. 24 h), MulRF used 29 160 s (i.e. 8.1 h), FastRFS-enhanced used 615 s (just over 10 min), MRL used 575 s (i.e. just under 10 min), and ASTRID, ASTRAL and FastRFS-basic used under 20 s.

The size of X impacts the running time for FastRFS, and ranged from 1155 to 20 233 for FastRFS-basic and from 2485 to 48 313 for FastRFS-enhanced. The most computationally intensive dataset for FastRFS-enhanced is the CPL dataset, which maximizes both the number of taxa and \( |X| \); however, FastRFS-enhanced completed on this dataset in 3282 s (i.e. under an hour). The majority of the time for FastRFS-enhanced is spent computing the MRL tree; the other parts of the analysis (i.e. computing the ASTRID matrix and potentially the ASTRID tree, computing the constraint set from ASTRAL, and running the DP algorithm) takes very little time (typically less than a minute).

ASTRID’s running time was highly variable, but the running time is high only for large datasets with missing entries in the distance matrix. The reason is that when the matrix has missing entries, ASTRID must use BIONJ* (which takes \( \Theta(n^3) \) time) instead of FastME (which takes \( \Theta(n^2) \) time). For example, ASTRID used about 6 h on the CPL dataset (the only biological dataset with these missing entries), but completed in just seconds on all the other datasets.

4 Conclusions

Supertree estimation is a basic bioinformatics challenge that is necessary for the construction of large phylogenies as well as for enabling statistical phylogeny estimation methods to be applied to large
Summary about supertree methods

- Supertree methods are essential techniques for large-scale phylogeny estimation, in part because divide-and-conquer is necessary.
- New approaches provide good accuracy but are limited to relatively small datasets.
- SuperFine can help, but only when the constraint tree computed by the Strict Consensus Merger is not too unresolved.
- New supertree methods are needed!
- Potential directions: distance-based supertree methods, constrained approaches, others?