Computing Triplet and Quartet Distances Between Trees

Gerth Stølting Brodal, Morten Kragelund Holt, Jens Johansen
Aarhus University

Rolf Fagerberg
University of Southern Denmark

Thomas Mailund, Christian N. S. Pedersen, Andreas Sand
Aarhus University, Bioinformatics Research Center

Work presented at SODA 2013 and ALENEX 2014
Outline

- Evolutionary trees
  - rooted vs. unrooted, binary vs. arbitrary degree
- Tree distances
  - Robinson-Foulds, triplet, quartet
- Results and previous work
  - triplet, quartet distances
- Algorithms
  - triplet (quartet)
- Experimental results (ALENEX 2014)
Rooted Evolutionary Tree

Bonobo  Chimpanzee  Human  Neanderthal  Gorilla  Orangutan
Dominant modern approach to study evolution is from DNA analysis
Constructing Evolutionary Trees – Binary or Arbitrary Degrees?

Sequence data

1
2
3
...
 n

Distance matrix

1 2 3 ... n

1 0 0 ... 0
2 0 0 ... 0
3 0 0 ... 0
... ... ... ...
n 0 0 0 ... 0

Binary trees
(despite no evidence in distance data)

Neighbor Joining
Saitou, Nei 1987
[ O(n^3) Saitou, Nei 1987 ]

Arbitrary degree
(compromise; good support for all edges)

Refined Buneman Trees
Moulton, Steel 1999
[ O(n^3) Brodal et al. 2003 ]

Arbitrary degrees
(strong support for all edges; few branches)

Buneman Trees
Buneman 1971
[ O(n^3) Berry, Bryan 1999 ]
Data Analysis vs Expert Trees – Binary vs Arbitrary Degrees?

*Cultural Phylogenetics of the Tupi Language Family in Lowland South America.*

Neighbor Joining on linguistic data

Linguistic expert classification (Aryon Rodrigues)
Evolutionary Tree Comparison

Robinson-Foulds distance = \# non-common splits = 2 + 1 = 3


[Day 1985] \( O(n) \) time algorithm using 2 x DFS + radix sort
Robinson-Foulds Distance (unrooted trees)


\[
\text{RF-dist}(T_1, T_2) = 4 + 5 = 9
\]

\[
\text{RF-dist}(T_1 \backslash \{8\}, T_2 \backslash \{8\}) = 0
\]

Robinson-Foulds very sensitive to outliers
Quartet Distance (unrooted trees)


Consider all $\binom{n}{4}$ quartets, i.e. topologies of subsets of 4 leaves \{i,j,k,l\}

- resolved: $ij|kl$
- unresolved: $ijkl$

(only non-binary trees)

Quartet-dist($T_1$, $T_2$) = $\binom{n}{4}$ - # common quartets = 5 - 1 = 4
Consider all \( \binom{n}{3} \) triplets, i.e. topologies of subsets of 3 leaves \{i,j,k\}

\[ \text{Resolved: } k \mid ij \quad \text{Unresolved: } ijk \]

(only non-binary trees)

\[
\begin{array}{c|c|c}
\text{Triplet} & T_1 & T_2 \\
\hline
\{1,2,3\} & 2 & 13 \\
\{1,2,4\} & 1 & 24 \\
\{1,2,5\} & 1 & 25 \\
\{1,3,4\} & 4 & 13 \\
\{1,3,5\} & 5 & 13 \\
\{1,4,5\} & 1 & 45 \\
\{2,3,4\} & 3 & 24 \\
\{2,3,5\} & 3 & 25 \\
\{2,4,5\} & 5 & 24 \\
\{3,4,5\} & 3 & 45 \\
\end{array}
\]

\[
\text{Triplet-dist}(T_1, T_2) = \binom{n}{3} - \# \text{common triplets} = 10 - 5 = 5
\]
## Computational Results

<table>
<thead>
<tr>
<th></th>
<th>Rooted Triplet distance</th>
<th>Unrooted Quartet distance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O($n^2$)</td>
<td></td>
<td>O($n^3$)</td>
</tr>
<tr>
<td>O($n \cdot \log^2 n$)</td>
<td></td>
<td>O($n^2$)</td>
</tr>
<tr>
<td>O($n \cdot \log n$)</td>
<td></td>
<td>O($n \cdot \log^2 n$)</td>
</tr>
<tr>
<td>CPQ 1996</td>
<td></td>
<td>SPMBF 2007</td>
</tr>
<tr>
<td>SBFPM 2013</td>
<td></td>
<td>D 1985</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BTKL 2000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BFP 2001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BFP 2003</td>
</tr>
<tr>
<td><strong>Arbitrary degrees</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O($n^2$)</td>
<td></td>
<td>O($d^9 \cdot n \cdot \log n$)</td>
</tr>
<tr>
<td>O($n \cdot \log n$)</td>
<td></td>
<td>O($n^2 \cdot 688$)</td>
</tr>
<tr>
<td>O($n \cdot \log n$)</td>
<td></td>
<td>O($d \cdot n \cdot \log n$)</td>
</tr>
<tr>
<td>BDF 2011</td>
<td></td>
<td>SPMBF 2007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>NKMP 2011</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[SODA 2013]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[ALENEX 2014]</td>
</tr>
</tbody>
</table>
Distance Computation

\[ \text{Triplet-dist}(T_1, T_2) = B + C + D = \binom{n}{3} - A - E \]

Sufficient to compute \( A \) and \( E \)

\( D + E \) and \( C + E \) unresolved in one tree

(For binary trees \( C \), \( D \) and \( E \) are all zero)
Parameterized Triplet & Quartet Distances

\[ B + \alpha \cdot (C + D), \quad 0 \leq \alpha \leq 1 \]

BDF 2011 \( O(n^2) \) for triplet, NKMP 2011 \( O(n^{2.688}) \) for quartet
[SODA 2013/ALENEX 2014] \( O(n \cdot \log n) \) and \( O(d \cdot n \cdot \log n) \), respectively
Counting Unresolved Triplets in One Tree

\[ \sum \sum_{i<j<k} n_i \cdot n_j \cdot n_k \]

Triplet anchored at \( v \)

Computable in \( O(n) \) time using DFS + dynamic programming

Quartets (root tree arbitrary)

\[ \sum_{v} \left( \sum_{i<j<k<l} n_i \cdot n_j \cdot n_k \cdot n_l + \left( n - \sum_{l} n_l \right) \sum_{i<j<k} n_i \cdot n_j \cdot n_k \right) \]

Quartet anchored at \( v \)
Counting Agreeing Triplets (Basic Idea)

\[ \sum_{v \in T_1} \sum_{w \in T_2} \sum_c \sum_{1 \leq i \leq d} \binom{n_i^c}{2} (n^w - n^c - n_i^w + n_i^c) \]

\[ \sum_{1 \leq i \leq d} n_i^w \]
Efficient Computation

Limit recolorings in $T_1$ (and $T_2$) to $O(n \cdot \log n)$

Reduce recoloring cost in $T_2$ to $O(n \cdot \log^2 n)$

Reduce recoloring cost in $T_2$ from $O(n \cdot \log^2 n)$ to $O(n \cdot \log n)$

- Contract $T_2$ and reconstruct $H(T_2)$ during recursion
Counting Agreeing Triplets (II)

node in $H(T_2)$ = component composition in $T_2$

Contribution to agreeing triplets at node in $H(T_2)$

$$\sum_{1 \leq i \leq d} n_i \binom{C_1}{i} \cdot n_{i^*} \binom{C_2}{i} + \sum_{1 \leq i \leq d} \binom{n_i \binom{C_1}{i}}{2} (n_{i^*} \binom{C_2}{i} - n_i \binom{C_2}{i}) + \sum_{1 \leq i \leq d} (n_{i^*} \binom{C_1}{i} - n_i \binom{C_1}{i}) n_{(ii)} \binom{C_2}{i}$$
From $O(n \cdot \log^2 n)$ to $O(n \cdot \log n)$

Compressed version of $T_2$ of size $O(n_v)$

Update $O(1)$ counters for all colors through node $H(\overline{T}_2)$

Colored path lengths

$$\sum_{2 \leq i \leq d} \log \left( \frac{|\overline{T}_2|}{n_i} \right) = \sum_{2 \leq i \leq d} n_i \cdot \log \frac{n_v}{n_i}$$

Total cost for updating counters

$$\sum_{\text{leaf } l \in T_1} \sum_{\text{ancestor } a^{(j)} \text{ not heavy child} \text{ child}} \log \frac{n a^{(j+1)}}{n a^{(j)}} = n \cdot \log n$$
Counting Quartets...

- Root $T_1$ and $T_2$ arbitrary
- Keep up to $7d^2 + 97d + 29$ different counters per node in $H(T_2)...$

Bottleneck in computing disagreeing resolved-resolved quartets

$\sum_{1 \leq i < d} \sum_{i < j \leq d} n_{(ij)} \cdot G_1 \cdot n_{(ij)} \cdot G_2$

double-sum $\Rightarrow$ factor $d$ time
Distance Computation

\[ \text{Triplet-dist}(T_1, T_2) = B + C + D = \binom{n}{3} - A - E \]

Resolved

- **A : Agree**
  - \(i\), \(j\), \(k\)

- **B : Disagree**
  - \(i\), \(j\), \(k\)
  - \(j\), \(i\), \(k\)
  - \(j\), \(k\), \(i\)
  - \(k\), \(i\), \(j\)

Unresolved

- **C**
  - \(i\), \(j\), \(k\)

- **D**
  - \(i\), \(j\), \(k\)
  - \(i\), \(k\), \(j\)
  - \(j\), \(i\), \(k\)
  - \(k\), \(i\), \(j\)

- **E**
  - \(i\), \(j\), \(k\)

O(\(n\cdot\log n\)) triplets & quartets

Sufficient to compute **A and E**
**ALENEX 2014: Implementation**  
(M.Sc. thesis Morten Kragelund Holt and Jens Johansen)

<table>
<thead>
<tr>
<th></th>
<th>Binary time counters</th>
<th>Arbitrary degree time counters</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Triplet</strong></td>
<td>O(n log n) 6</td>
<td>O(n log n) 4d+2</td>
</tr>
<tr>
<td><strong>Quartet</strong></td>
<td>O(n log n) 40</td>
<td>O(max(d₁, d₂) n log n) 2d² + 79d + 22 (B, with T₁↔T₂)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>O(min(d₁, d₂) n log n) 7d² + 97d + 29 (B, no swap)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>d² + 12d + 12 (E, no swap)</td>
</tr>
</tbody>
</table>

Worst-case #counters per node in HDT(T₂)

- First implementation for triplets for arbitrary degree
- Space usage ≈10 KB per node for quartet (binary trees)  
  - can handle ≈ 1,000,000 leaves
- 64 bit integers, except 128 bit integers for values > n³  
  - quartet distance of up to ≈ 2,000,000 leaves
Experimental Results
Quartet Distance – Binary Trees

- [ALENEX 2014] are the first \(O(n \cdot \log n)\) implementations
- MP 2004 overhead from working with polynomials
Experimental Results

Quartet Distance – High Degree Trees

- $d = 256$
- $d = 1024$

[ALENEX 2014] are the first $n \cdot \text{poly}(\log n, d)$ implementation
Experimental Results

Triplet Distance – Binary Trees

- [ALENEX 2014] are the first $O(n \cdot \log n)$ implementation
- SBFPM 2013 only binary trees, no contractions
Experimental Results

Triplet Distance – High Degree Trees

- [ALENEX 2014] first implementation
- Triplet distance appears hardest for binary trees
### Summary

<table>
<thead>
<tr>
<th>Rooted Triplet distance</th>
<th>Unrooted Quartet distance</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary</strong></td>
<td></td>
</tr>
<tr>
<td>(O(n^2))</td>
<td>(O(n^3))</td>
</tr>
<tr>
<td>(O(n \cdot \log^2 n))</td>
<td>(O(n^2))</td>
</tr>
<tr>
<td>(O(n \cdot \log n))</td>
<td>(O(n \cdot \log n))</td>
</tr>
<tr>
<td><strong>Arbitrary degrees</strong></td>
<td></td>
</tr>
<tr>
<td>(O(n^2))</td>
<td>(O(d^9 \cdot n \cdot \log n))</td>
</tr>
<tr>
<td>(O(n \cdot \log n))</td>
<td>(O(n^2 \cdot 688))</td>
</tr>
<tr>
<td>(\star)</td>
<td>(\star)</td>
</tr>
</tbody>
</table>

\(d = \text{minimal degree of any node in } T_1 \text{ and } T_2\)

\(\star = \text{fastest implementation for large } n\)
References

- **On the Scalability of Computing Triplet and Quartet Distances.**

- **Algorithms for Computing the Triplet and Quartet Distances for Binary and General Trees.**
  Biology - Special Issue on Developments in Bioinformatic Algorithms, 2013.

- **A practical O(n log^2 n) time algorithm for computing the triplet distance on binary trees.**

- **Efficient Algorithms for Computing the Triplet and Quartet Distance Between Trees of Arbitrary Degree.**
  G.S. Brodal, R. Fagerberg, C.N.S. Pedersen, T. Mailund, A. Sand.
  SODA 2013.

- **A sub-cubic time algorithm for computing the quartet distance between two general trees.**
  Algorithms in Molecular Biology 2011.

- **Computing the Quartet Distance Between Evolutionary Trees of Bounded Degree.**

- **QDist - Quartet Distance between Evolutionary Trees.**

- **Computing the Quartet Distance Between Evolutionary Trees in Time O(n log n).**

- **Computing the Quartet Distance Between Evolutionary Trees in Time O(n log^2 n).**