RAxML vs. FastTree: A Comparison of Two Maximum Likelihood Phylogeny Estimation Methods

Mia Schoening
RAxML vs. FastTree

- **RAxML**
  - Implements standard SPR-based hill-climbing algorithm
  - Main method used for large-scale ML phylogeny estimation
  - Computational requirements limit number of sequences and sites

- **FastTree**
  - Uses combination of Neighbor-Joining, Minimum Evolution, and ML-based NNI rearrangement methods
  - Can handle alignments up to 1 million sequences
  - On large alignments, orders of magnitude faster than RAxML and PhyML
RAxML vs. FastTree

- **RAxML**
  - Implements standard SPR-based hill-climbing algorithm
  - Main method used for large-scale ML phylogeny estimation
  - Computational requirements limit number of sequences and sites

- **FastTree**
  - Uses combination of Neighbor-Joining, Minimum Evolution, and ML-based NNI rearrangement methods
  - Can handle alignments up to 1 million sequences
  - On large alignments, orders of magnitude faster than RAxML and PhyML

When is it necessary to use RAxML over FastTree?
Model Tree

- Sequence Type: DNA
- Number of Sequences: 1001
- Sequence Length: 5000 sites
Generating Model Tree Set

- **Branch Length**
  - Small: x10
  - Moderate: x50
  - Large: x100

- **Number of Leaves**
  - Dense: 1001 Taxa
  - Sparse: 50 Taxa
Generating Model Tree Set

- **Branch Length**
  - Small: x10
  - Moderate: x50
  - Large: x100

- **Number of Leaves**
  - Dense: 1001 Taxa
  - Sparse: 50 Taxa

1. Small-Sparse
2. Small-Dense
3. Moderate-Sparse
4. Moderate-Dense
5. Large-Sparse
6. Large-Dense
Software Versions

RAxML Version 8.2.12

FastTree Version 2.1.10
Running Time Analysis

Sparse Trees

<table>
<thead>
<tr>
<th></th>
<th>FastTree</th>
<th>RAxML</th>
</tr>
</thead>
<tbody>
<tr>
<td>small_sparse</td>
<td>12.93</td>
<td>179.11</td>
</tr>
<tr>
<td>moderate_sparse</td>
<td>12.96</td>
<td>325.83</td>
</tr>
<tr>
<td>large_sparse</td>
<td>13.49</td>
<td>255.41</td>
</tr>
</tbody>
</table>

Dense Trees

<table>
<thead>
<tr>
<th></th>
<th>FastTree</th>
<th>RAxML</th>
</tr>
</thead>
<tbody>
<tr>
<td>small_dense</td>
<td>312</td>
<td>312</td>
</tr>
<tr>
<td>moderate_dense</td>
<td>333.84</td>
<td>2829.01</td>
</tr>
<tr>
<td>large_dense</td>
<td>350.73</td>
<td>33060.13</td>
</tr>
</tbody>
</table>
## ML Scores (as log likelihoods)

<table>
<thead>
<tr>
<th></th>
<th>Small_Sparse</th>
<th>Small_Dense</th>
<th>Moderate_Sparse</th>
<th>Moderate_Dense</th>
<th>Large_Sparse</th>
<th>Large_Dense</th>
</tr>
</thead>
<tbody>
<tr>
<td>FastTree</td>
<td>-210,099</td>
<td>-3,387,957</td>
<td>-242,980</td>
<td>-4,324,461</td>
<td>-253,999</td>
<td>-4,571,306</td>
</tr>
<tr>
<td>RAxML</td>
<td>-207,591</td>
<td>-3,366,916</td>
<td>-240,882</td>
<td>-4,277,753</td>
<td>-252,064</td>
<td>-4,519,003</td>
</tr>
</tbody>
</table>
Tree Topology Analysis
Discussion

- RAxML consistently outperforms FastTree with respect to ML scores and tree topology, but at the cost of longer running times.
- Both methods had higher RF error values on sparse datasets than dense datasets.
- As edge lengths increased, RF error values increased for both methods.
References


