Fast Phylogenetic Approaches

Xiaofan Zhou
Integrative Microbiology Research Centre
South China Agriculture University
Why is tree inference so difficult?

• Too many trees to look at

• Too many calculations to do
Too many trees

No. of binary unrooted trees with \( n \) tips:
\[
= 1 \times 3 \times 5 \cdots \times (2n - 5)
\]
\[
= (2n - 3)!/(2^{(n-2)}(n - 2)!)\]

<table>
<thead>
<tr>
<th>Tips</th>
<th>Binary unrooted trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>15</td>
</tr>
<tr>
<td>10</td>
<td>2,027,025</td>
</tr>
<tr>
<td>20</td>
<td>2.22 \times 10^{18}</td>
</tr>
<tr>
<td>30</td>
<td>8.69 \times 10^{36}</td>
</tr>
<tr>
<td>40</td>
<td>1.31 \times 10^{55}</td>
</tr>
<tr>
<td>50</td>
<td>2.84 \times 10^{74}</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Sunway Taihu-Light**
World’s fastest and largest supercomputer
Peak Flops: 125.4 \times 10^{15}

\(~3.31 \times 10^{21} \) billion years
Too many calculations

• Branch length estimation

• Model parameter optimization

• ......
Fast phylogenetic approaches

• Too many trees to look at
  • Heuristic search of the tree space

• Too many calculations to do
  • Approximate estimation of branch length

• Other techniques for fast phylogenetics
Heuristic tree search

Starting tree → Tree topology modification and evaluation → Final tree

If a better tree is found
Nearest Neighbor Interchange (NNI)
Nearest Neighbor Interchange (NNI)

Total NNI: $2(n - 3)$
Subtree Pruning and Re-grafting (SPR)

Total SPR: $4(n - 3)(n - 2)$
Unique tree: 15
SPR as a chain of NNI
Heuristic search of tree space
Heuristic tree search
Heuristic tree search
Approximate likelihood calculation

- Global optimization vs. local optimization
Approximate likelihood calculation

- Global optimization vs. local optimization

- Exhaustive optimization vs. approximate optimization
  - Diminished return from extra efforts
  - Subsequent topological changes can invalidate extra efforts
Other techniques for fast phylogenetics

• GAMMA vs CAT

• Fast approaches for node support

• Parallelization

• Memory saving
Outline

• Popular fast phylogenetic programs
  • FastTree, RAxML, PhyML, IQ-TREE
  • Main algorithm and development

• Empirical evaluation using recent phylogenomic datasets
**NJ – Neighbor Joining**

Pairwise distance matrix: $D$

For each tip, calculate:

$$u_i = \sum_{k:k\neq i}^n D_{ik}/(n - 2)$$

Identify the pair of tips $i$ and $j$ that minimize the NJ criterion:

$$Q = D_{ij} - u_i - u_j$$

Join $i$ and $j$, replace with a new node $(ij)$, and update $D$:

$$D_{(ij), k} = (D_{ik} + D_{jk} - D_{ij})/2$$

Repeat until fully resolved
What is NJ optimizing?

- NJ is a greedy algorithm optimizing “balance” tree length:

\[ L = \sum_{i,j} 2^{1-p_{ij}} D_{ij} \]

Pairwise distance matrix: \( D \)

For each tip, calculate:

\[ u_i = \sum_{k: k \neq i}^n D_{ik}/(n - 2) \]

Identify the pair of tips \( i \) and \( j \) that minimize the NJ criterion:

\[ Q = D_{ij} - u_i - u_j \]

Join \( i \) and \( j \), replace with a new node \( (ij) \), and update \( D \):

\[ D_{(ij), k} = (D_{ik} + D_{jk} - D_{ij})/2 \]

Repeat until fully resolved.
What is NJ optimizing?

• Balanced Minimum Evolution
• FastME:
  → stepwise addition starting tree
  → NNI
  → SPR (FastME 2)

Pairwise distance matrix: $D$

For each tip, calculate:
$$u_i = \sum_{k:k \neq i}^{n} \frac{D_{ik}}{n - 2}$$

Identify the pair of tips $i$ and $j$ that minimize the NJ criterion:
$$Q = D_{ij} - u_i - u_j$$

Join $i$ and $j$, replace with a new node $(ij)$, and update $D$:
$$D_{(ij), k} = \frac{(D_{ik} + D_{jk} - D_{ij})}{2}$$

Repeat until fully resolved
Variants of NJ

• BIONJ/WEIHBOR
  • Take variance/co-variance into consideration

• Relaxed Neighbor Joining
  • Join the pair of neighbors firstly found, instead of the one that minimizing $Q$

• Fast Neighbor Joining
  • Look for a subset of the pairs
FastTree

• Sequence profile instead of distance matrix
  • Reduce memory requirements
  • Use an average profile of all active nodes to calculate the NJ criterion, instead of actually doing all pairwise computing

• Three main heuristics
  • Top-hits: neighbors of neighbors are also likely to be neighbors
  • Fast neighbor joining: remembering the best join candidate for each node
  • Relaxed neighbor joining: hill-climbing search for best join

\[
\begin{array}{cccccc}
A & C & G & T & A \\
A & - & C & T & A \\
C & A & G & T & A \\
C & A & G & G & A \\
\end{array}
\]

\[
\begin{array}{cccccc}
A & 0.5 & 0.66 & 0 & 0 & 1 \\
T & 0 & 0 & 0 & 0.75 & 0 \\
C & 0.5 & 0.33 & 0.25 & 0 & 0 \\
G & 0 & 0 & 0.75 & 0.25 & 0 \\
- & 0 & 0.25 & 0 & 0 & 0 \\
\end{array}
\]
FastTree2

- ME SPR
  - Extends SPR only along the best path up to length of 10
  - Only two rounds of SPR for each subtree
FastTree2

• ME SPR
  • Extends SPR only along the best path up to length of 10
  • Only two rounds of SPR for each subtree

• ML NNI
  • $2\log_2 N$ quick rounds + 1 final thorough rounds
  • Heuristics:
    • Branch-length estimation
      • Skip the topology if significantly worse than current tree
    • Star-topology test
      • Pick the winner when one of the quartet tree is significantly better than a star topology
    • Subtree skipping
      • Skip subtrees without significant improvement in recent two rounds
FastTree2

NJ

ME NNI

ME SPR

ML NNI

Final tree

Alternate until reach the round max
## FastTree: performance

<table>
<thead>
<tr>
<th></th>
<th>16S rRNA</th>
<th>16S rRNA</th>
<th>7 COGs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of sequences</td>
<td>4,114</td>
<td>6,718</td>
<td>2,500</td>
</tr>
<tr>
<td>RAxML 7's Log Likelihood</td>
<td>-325,581</td>
<td>-481,259</td>
<td>-1,238,666</td>
</tr>
<tr>
<td>FastTree 2's Log Likelihood</td>
<td>-328,062</td>
<td>-493,841</td>
<td>-1,240,916</td>
</tr>
<tr>
<td>Difference</td>
<td>2,481</td>
<td>12,582</td>
<td>2,251</td>
</tr>
<tr>
<td>Well-supported RAxML splits (bootstrap &gt; 0.9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total in RAxML tree</td>
<td>851</td>
<td>1,124</td>
<td>-</td>
</tr>
<tr>
<td>Found by FastTree</td>
<td>837</td>
<td>1,075</td>
<td>-</td>
</tr>
<tr>
<td>Weakly-supported RAxML splits (bootstrap 0.8–0.9)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total in RAxML tree</td>
<td>265</td>
<td>419</td>
<td>-</td>
</tr>
<tr>
<td>Found by FastTree</td>
<td>250</td>
<td>365</td>
<td>-</td>
</tr>
<tr>
<td>Locally-supported RAxML splits (SH &gt; 0.95)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Total in RAxML tree</td>
<td>1,336</td>
<td>1,927</td>
<td>1,018</td>
</tr>
<tr>
<td>Found by FastTree</td>
<td>1,033</td>
<td>1,319</td>
<td>889</td>
</tr>
</tbody>
</table>

*Table 3. Comparison of RAxML and FastTree’s log likelihoods, and the agreement of FastTree with RAxML’s well-supported splits, for large genuine alignments.*

FastTree: performance

Table 4. Running time and memory usage on genuine alignments.

<table>
<thead>
<tr>
<th>Alignment</th>
<th>Distinct</th>
<th>FastTree 2.0.0</th>
<th>RAXML 7 Hours</th>
<th>PhyML 3 Hours</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sequences</td>
<td>Positions</td>
<td>Model</td>
<td>Hours</td>
</tr>
<tr>
<td>16S rRNA, subsets</td>
<td>500</td>
<td>1,287 nt.</td>
<td>GTR</td>
<td>0.02</td>
</tr>
<tr>
<td>COGs, subsets</td>
<td>500</td>
<td>65–1,009 a.a.</td>
<td>JTT</td>
<td>0.02</td>
</tr>
<tr>
<td>COGs, subsets</td>
<td>2,500</td>
<td>197–384 a.a.</td>
<td>JTT</td>
<td>0.11</td>
</tr>
<tr>
<td>Efflux permeases</td>
<td>8,362</td>
<td>394 a.a.</td>
<td>JTT</td>
<td>0.25</td>
</tr>
<tr>
<td>16S rRNAs, families</td>
<td>15,011</td>
<td>1,287 nt.</td>
<td>GTR</td>
<td>0.66</td>
</tr>
<tr>
<td>ABC transporters</td>
<td>39,092</td>
<td>214 a.a.</td>
<td>JTT</td>
<td>1.02</td>
</tr>
<tr>
<td>16S rRNAs, all</td>
<td>237,882</td>
<td>1,287 nt.</td>
<td>JC</td>
<td>21.8</td>
</tr>
</tbody>
</table>
PHYLIP – PHYLogeny Inference Package

Three species starting tree

Sequential addition of the next species

NNI

SPR

Final tree

Repeat until the tree is comprehensive
fastDNAml

• Algorithm largely the same as PHYLIP

• Less exhaustive branch-length optimization
  • Only optimize the three branches relevant to the sequence addition

• Lazy SPR
  • Only consider re-grafting on branches at most $r$ node away from the pruning position
RAxML - Randomized AxML

- Parsimony starting tree:
  - Parsimony is connected with ML
  - Speed and randomization!

- Lazy SPR
  - Only pre-scoring during one SPR iteration
  - SPRs leading to better scores are immediately implied
  - Dynamic adjustment of Lazy SPR radius
SPR iteration

SPR cycle for a subtree:

1. make rearrangement and pre-scoring;
2. if got better score, store the tree in “bestLSR” and “best20List” (if applicable);

At the end, accept the best SPR in “bestLSR” and update the current best tree (if applicable).

Thorough optimization for “best20List”

Better tree?

Yes

Increase $r_i/r_u$ by 5

$r_{up} + 5 \leq r_{max}$?

Yes

END

No

No
RAxML: improvements

• Optimized data structure
  • rearrangement descriptor
  • (pruned position, re-grafting position, $l_1$, $l_2$, $l_3$, $l_4$)
RAxML: improvements

• Optimized data structure

• Adaptive determination of LSR radius
  • SPR on the RAP starting tree with different $\tau$ value
  • Choose the smallest $\tau$ giving rise to the best score
RAxML: improvements

- Optimized data structure
- Adaptive determination of LSR radius
- “Subtree skipping”
  - $lh_{cutoff}$ determined dynamically during SPR cycles

\[ all(t') - ll(t) < lh_{cutoff} \]

~2.5x speed up
SPR iteration

Determine the best \( r \) on the RAP tree

SPR iteration

1. make rearrangement and pre-scoring;
2. if a re-grafting lead to substantially worse score, skip all the entire clade below that branch;
3. if got better score, store the tree in “bestLSR” and “best20List” (if applicable);

At the end, accept the best SPR in “bestLSR”, update the current best tree and “best20List” (if applicable);

Thorough optimization for “best20List”

Better tree?

Yes

Yes

\( r_{up} + 5 \leq r_{max} \)?

No

END

Yes

No

Increase \( r/ru \) by 5

RAP starting tree

subtree 1

subtree 2

......

......

SUBTREE 1

SUBTREE 2

......

......

SPR cycle for a subtree:

For every branch within \( r_{l}/ru \) radius of the pruned location:
1. make rearrangement and pre-scoring;
2. if a re-grafting lead to substantially worse score, skip all the entire clade below that branch;
3. if got better score, store the tree in “bestLSR” and “best20List” (if applicable);

At the end, accept the best SPR in “bestLSR”, update the current best tree and “best20List” (if applicable);
PhyML

• Single internal branch re-optimization
PhyML

• Single internal branch re-optimization

• Simultaneous topology modifications
  • Rank all NNI candidates by LLS
  • Remove conflicting candidates
  • Apply the top $\lambda$ fraction simultaneously
  • If get worse score, lower $\lambda$ by half; keep going until the best one
PhyML

BIONJ starting tree

Find NNI candidates

Yes

Rank candidates, remove conflicts,

Apply top $\lambda$ fraction

Improvement?

Yes

No

Lower $\lambda$ by half

END
PhyML: performance

---

**Real data**

<table>
<thead>
<tr>
<th>Method</th>
<th>218 taxa (4,182 bp)</th>
<th>500 taxa (1,428 bp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNADIST+ NJ/BIONJ</td>
<td>50 sec</td>
<td>2 min, 19 sec</td>
</tr>
<tr>
<td>DNADIST+ Neighbor</td>
<td>4 min, 52 sec</td>
<td>58 min, 40 sec</td>
</tr>
<tr>
<td>DNAPARS</td>
<td>4 min, 4 sec</td>
<td>13 min, 12 sec</td>
</tr>
<tr>
<td>PAUP*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PAUP*+ NJ</td>
<td>10 hr, 50 min</td>
<td></td>
</tr>
<tr>
<td>MrBayes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fastDNAm1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NJML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MetaPIGA</td>
<td>4 hr, 45 min</td>
<td>9 hr, 4 min</td>
</tr>
<tr>
<td>MetaPIGA+ NJ</td>
<td>1 hr, 40 min</td>
<td>3 hr</td>
</tr>
<tr>
<td>PHYML</td>
<td>8 min, 13 sec (15)</td>
<td>11 min, 59 sec (13)</td>
</tr>
</tbody>
</table>

---

**Simulations**

<table>
<thead>
<tr>
<th>Method</th>
<th>40 taxa (500 bp)</th>
<th>100 taxa (500 bp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNADIST+ NJ/BIONJ</td>
<td>0.3 sec</td>
<td>2.3 sec</td>
</tr>
<tr>
<td>DNADIST+ Neighbor</td>
<td>1.5 sec</td>
<td>22 sec</td>
</tr>
<tr>
<td>DNAPARS</td>
<td>0.5 sec</td>
<td>6 sec</td>
</tr>
<tr>
<td>PAUP*</td>
<td>3 min, 21 sec</td>
<td>1 hr, 4 min</td>
</tr>
<tr>
<td>PAUP*+ NJ</td>
<td>1 min, 10 sec</td>
<td>22 min</td>
</tr>
<tr>
<td>MrBayes</td>
<td>2 min, 6 sec</td>
<td>32 min, 37 sec</td>
</tr>
<tr>
<td>fastDNAm1</td>
<td>1 min, 13 sec</td>
<td>26 min, 31 sec</td>
</tr>
<tr>
<td>NJML</td>
<td>15 sec</td>
<td>6 min, 4 sec</td>
</tr>
<tr>
<td>MetaPIGA</td>
<td>21 sec</td>
<td>3 min, 27 sec</td>
</tr>
<tr>
<td>MetaPIGA+ NJ</td>
<td>6 sec</td>
<td>23 sec</td>
</tr>
<tr>
<td>PHYML</td>
<td>2.7 sec (6.4)</td>
<td>12 sec (8.3)</td>
</tr>
</tbody>
</table>

PhyML-SPR

- Filter candidate SPR moves based on distance criterion
- Quick estimation of branch-length using distance-based approach

Hordijk and Gascuel (2005) Bioinformatics

[Diagram of phylogenetic tree with labels A, B, C, D, E and connections between them]
PhyML 3

• Use parsimony score instead of distance to filter SPR candidates

• Alternated SPR and NNI searches
  • NNI optimizes all five relevant branches instead of one
SPR iteration

Filter SPR candidates with parsimony score, and for each survived candidate:
1. Optimize branch length at pruned position;
2. Evaluate candidate by likelihood score;
3. If no improvements, optimize the three relevant branches and re-evaluate;
4. Update best tree and best score (if applicable)

At the end, accept the best tree and score (if applicable).
PhyML 3: performance

<table>
<thead>
<tr>
<th>TABLE 3. Comparison of log-likelihoods on 50 DNA and 50 protein medium-size data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Av. LogLk rank</strong></td>
</tr>
<tr>
<td>DNA</td>
</tr>
<tr>
<td>PhyML 2.4.5</td>
</tr>
<tr>
<td>PhyML 3.0 NNI</td>
</tr>
<tr>
<td>PhyML 3.0 SPR</td>
</tr>
<tr>
<td>PhyML 3.0 BEST</td>
</tr>
<tr>
<td>PhyML 3.0 RAND</td>
</tr>
<tr>
<td>RAxML</td>
</tr>
<tr>
<td>Protein</td>
</tr>
<tr>
<td>PhyML 2.4.5</td>
</tr>
<tr>
<td>PhyML 3.0 NNI</td>
</tr>
<tr>
<td>PhyML 3.0 SPR</td>
</tr>
<tr>
<td>PhyML 3.0 BEST</td>
</tr>
<tr>
<td>PhyML 3.0 RAND</td>
</tr>
<tr>
<td>RAxML</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TABLE 4. Comparison of log-likelihoods on 10 DNA and 10 protein large data sets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Av. LogLk rank</strong></td>
</tr>
<tr>
<td>DNA</td>
</tr>
<tr>
<td>PhyML 2.4.5</td>
</tr>
<tr>
<td>PhyML 3.0 NNI</td>
</tr>
<tr>
<td>PhyML 3.0 SPR</td>
</tr>
<tr>
<td>RAxML</td>
</tr>
<tr>
<td>Protein</td>
</tr>
<tr>
<td>PhyML 2.4.5</td>
</tr>
<tr>
<td>PhyML 3.0 NNI</td>
</tr>
<tr>
<td>PhyML 3.0 SPR</td>
</tr>
<tr>
<td>RAxML</td>
</tr>
</tbody>
</table>

Quartet Puzzling

• Building tree from small pieces
  • 40 sequences
    \(1.31 \times 10^{51}\) binary unrooted trees
    931,390 quartets

• Stepwise addition based on quartet trees
PUZZLE

Quartet puzzling:
For each possible combination of three sequences \((a, b, c)\) in tree:
- Identify the quartet tree (e.g. \((ab|cx)\))
- Add penalty to branches connecting \(a\) and \(b\)

Insert \(x\) to the branch with lowest penalty

ML tree for all quartets
Randomize the order of sequences
Start with the first quartet tree

Repeat many times

All (local) best trees
IQPNNI - Important quartet puzzling + NNI

- Important quartet

Important quartet puzzling:

For each important quartet in tree:
- Build quartet tree with NJ (e.g. \((ab|cx)\))
- Add score of 1 to all branches in the subtree represented by \(c\)

Insert \(x\) to the branch with highest score

BIONJ tree, followed by NNI

Delete seqs with probability \(P_{del}\)

1\(^{st}\) del seq

2\(^{nd}\) del seq

\(\ldots\)

last del seq

NNI

All (local) best trees

Repeat until the number of iteration exceeds a threshold
Escape from local optima

Tree likelihood score vs. Tree space

- Stochastic perturbation
- NNI hill-climbing
Ratchet

Tree likelihood score

Tree space

Re-weighting of sites

Remove the weighting
**IQ-TREE**

- A pool of starting trees
- A pool of candidate trees
- NNI- instead of IQP-based perturbation
- Simultaneous NNI modifications
  - Reduced NNI neighborhood

---

Escape from local optima: IQ-TREE

Stochastic perturbation
IQ-TREE: performance

Other techniques for fast phylogenetics

- GAMMA vs CAT
GAMMA vs CAT

• GAMMA
  • model rate heterogeneity among sites using the gamma distribution
  • each site has certain probability belonging to each rate category

• CAT
  • assign sites into fixed number of rate categories
  • each site belongs to a specific rate category
<table>
<thead>
<tr>
<th>Dataset</th>
<th>$T(\Gamma)/T({\text{CAT}})$</th>
<th>$T(\Gamma'')/T({\text{CAT+R}})$</th>
<th>$l_p(\Gamma)/l_p({\text{CAT}})$</th>
<th>$l_p(\Gamma')/l_p({\text{CAT+R}})$</th>
<th>$RF(\Gamma', {\text{CAT}})$</th>
<th>$RF(\Gamma', {\text{CAT+R}})$</th>
<th>$\alpha$</th>
<th># pat</th>
</tr>
</thead>
<tbody>
<tr>
<td>73, OLA F</td>
<td>4.177018</td>
<td>2.779553</td>
<td>0.999959</td>
<td>0.999959</td>
<td>0.008392</td>
<td>0.005594</td>
<td>1.180</td>
<td>1,196</td>
</tr>
<tr>
<td>74, OLA F</td>
<td>3.456238</td>
<td>2.420959</td>
<td>0.999963</td>
<td>0.999963</td>
<td>0.004837</td>
<td>0.004571</td>
<td>0.575</td>
<td>578</td>
</tr>
<tr>
<td>104, OLA F</td>
<td>2.971896</td>
<td>1.465502</td>
<td>0.999616</td>
<td>1.000293</td>
<td>0.113659</td>
<td>0.098049</td>
<td>0.329</td>
<td>581</td>
</tr>
<tr>
<td>128, OLA F</td>
<td>8.728934</td>
<td>4.362863</td>
<td>1.000026</td>
<td>1.000268</td>
<td>0.16996</td>
<td>0.16996</td>
<td>3.166</td>
<td>2,985</td>
</tr>
<tr>
<td>144, OLA F</td>
<td>4.335371</td>
<td>2.23404</td>
<td>0.999983</td>
<td>1.000107</td>
<td>0.055789</td>
<td>0.055088</td>
<td>0.825</td>
<td>1,254</td>
</tr>
<tr>
<td>178, OLA F</td>
<td>4.742052</td>
<td>2.397997</td>
<td>0.999998</td>
<td>1.000133</td>
<td>0.026346</td>
<td>0.026062</td>
<td>0.634</td>
<td>1,150</td>
</tr>
<tr>
<td>180, OLA F</td>
<td>3.261044</td>
<td>2.300603</td>
<td>0.999608</td>
<td>1.000112</td>
<td>0.048179</td>
<td>0.046499</td>
<td>0.454</td>
<td>924</td>
</tr>
<tr>
<td>101, SC</td>
<td>8.697863</td>
<td>4.081393</td>
<td>0.999791</td>
<td>0.999873</td>
<td>0.09492</td>
<td>0.084925</td>
<td>0.417</td>
<td>1,630</td>
</tr>
<tr>
<td>150, SC</td>
<td>4.212270</td>
<td>2.630621</td>
<td>0.999555</td>
<td>1.00037</td>
<td>0.00404</td>
<td>0.003232</td>
<td>0.433</td>
<td>1,130</td>
</tr>
<tr>
<td>150, ARB</td>
<td>6.935958</td>
<td>4.125580</td>
<td>1.000019</td>
<td>1.00032</td>
<td>0.013805</td>
<td>0.014478</td>
<td>0.562</td>
<td>2,137</td>
</tr>
<tr>
<td>193, VINH</td>
<td>2.541966</td>
<td>1.822700</td>
<td>0.999929</td>
<td>1.00007</td>
<td>0.117755</td>
<td>0.112272</td>
<td>1.313</td>
<td>459</td>
</tr>
<tr>
<td>200, ARB</td>
<td>7.359741</td>
<td>3.981281</td>
<td>1.00068</td>
<td>1.00089</td>
<td>0.036272</td>
<td>0.034257</td>
<td>0.534</td>
<td>2,253</td>
</tr>
<tr>
<td>218, RDP H</td>
<td>5.890610</td>
<td>2.320172</td>
<td>0.998824</td>
<td>1.00018</td>
<td>0.120092</td>
<td>0.103695</td>
<td>0.545</td>
<td>1,847</td>
</tr>
<tr>
<td>250, ARB</td>
<td>7.076141</td>
<td>3.817160</td>
<td>1.00027</td>
<td>1.00076</td>
<td>0.022394</td>
<td>0.028974</td>
<td>0.550</td>
<td>2,350</td>
</tr>
<tr>
<td>500, ARB</td>
<td>7.378079</td>
<td>3.243040</td>
<td>1.000112</td>
<td>1.000207</td>
<td>0.057573</td>
<td>0.050351</td>
<td>0.579</td>
<td>2,751</td>
</tr>
<tr>
<td>500, ZILLA</td>
<td>4.150663</td>
<td>3.014160</td>
<td>1.000160</td>
<td>1.003023</td>
<td>0.054162</td>
<td>0.048947</td>
<td>0.494</td>
<td>1,193</td>
</tr>
<tr>
<td>715, CHUCK</td>
<td>4.663363</td>
<td>2.297917</td>
<td>0.999991</td>
<td>1.000146</td>
<td>0.043868</td>
<td>0.039804</td>
<td>0.842</td>
<td>1,231</td>
</tr>
<tr>
<td>1000, ARB</td>
<td>8.151405</td>
<td>2.894259</td>
<td>1.001454</td>
<td>1.001549</td>
<td>0.051377</td>
<td>0.048072</td>
<td>0.552</td>
<td>3,364</td>
</tr>
<tr>
<td>1663, ARB</td>
<td>4.893710</td>
<td>1.827999</td>
<td>1.000221</td>
<td>1.000320</td>
<td>0.087571</td>
<td>0.084487</td>
<td>0.621</td>
<td>1,576</td>
</tr>
<tr>
<td>Averages</td>
<td>5.450585</td>
<td>2.843387</td>
<td>1.000037</td>
<td>1.000183</td>
<td>0.053958</td>
<td>0.050539</td>
<td>0.770</td>
<td>1,609</td>
</tr>
</tbody>
</table>
Other techniques for fast phylogenetics

• GAMMA vs CAT

• Fast approaches for node support
Standard Bootstrap

Original data matrix $\rightarrow$ Standard tree inference $ightarrow$ Tree

Bootstrap rep. data matrix 0 $\rightarrow$ Bootstrap rep. tree 0

Bootstrap rep. data matrix 1 $\rightarrow$ Bootstrap rep. tree 1

... $\rightarrow$ ...

Bootstrap rep. data matrix $n$ $\rightarrow$ Bootstrap rep. tree $n$
Rapid bootstrap (RAxML)

Original data matrix

Bootstrap rep. data matrix 0

Bootstrap rep. data matrix 1

Bootstrap rep. data matrix 10

MP tree from original data

Bootstrap rep. tree 0

Bootstrap rep. tree 1

Bootstrap rep. tree 10

Tree

Additional shortcuts:

• LSR radius randomly chosen between 5 and 15;
• 2 iterations of LSR;
• More aggressive subtree skipping;
• Thorough optimization for best 5 instead of 20;
Local branch support

1. Approximate likelihood-ratio test (aLRT):
   - \(aLRT \leftarrow 2(l_1 - \max(l_2, l_3)) < 2(l_1 - l_0)\)

2. SH-aLRT:
   - aLRT with RELL bootstrap re-sampling
   - Support value = \(\text{count}(aLRT > aLRT^*)\)

3. Approximate Bayes (aBayes):
   - \(\Pr(T_1|D) = \Pr(D|T_1) \Pr(T_1)/\sum_{i=1}^{3} \Pr(D|T_i) \Pr(T_i)\)
Local branch support: performance

Ultra-fast bootstrap (IQ-TREE)

Original data matrix

IQ-TREE iterative search algorithm

Bootstrap rep. site index 0

Bootstrap rep. site index 1

Bootstrap rep. site index n

Resampling with replacement

For every tree $T$ with $l(T) > l_{\text{min}}$:

per-site log likelihood values

Compare with current $l(T')$, update $T'$ and $l(T')$ if applicable

Bootstrap rep. tree $T_0 / l(T'_0)$

Bootstrap rep. tree $T_1 / l(T'_1)$

Bootstrap rep. tree $T_n / l(T'_n)$
UFBS: performance

Other techniques for fast phylogenetics

• GAMMA vs CAT

• Fast approaches for node support

• Parallelization
Parallelization

• Multi-threading and MPI

• Parallel tree searches:
  • MPI: RAxML, PhyML, TREE-PUZZLE/pIQPNNI/IQ-TREE

• Likelihood calculation
  • RAxML/IQ-TREE
Other techniques for fast phylogenetics

• GAMMA vs CAT

• Fast approaches for node support

• Parallelization

• Memory saving
Memory saving

- Subtree equality vector (SEV) for gappy data set
Recommendations:

• Multiple searches using distinct starting trees

• More thorough searches by tuning key parameters
  • FastTree: more thorough NNI (“-mlacc”, “-slownni”) / no. of ME SPR (“-spr”)
  • RAxML: Lazy SPR radius (“-c”) / old hill-climbing strategy (“-f o”)
  • IQ-TREE: length of search (“-nstop”) / perturbation strength (“-pers”)

• More than one phylogenetic approaches
NJ

BIONJ

FastME
**FastTree**


**RAxML/ExaML**


**PhyML**


**TREE-PUZZLE/IQPNNI/IQ-TREE**


**CAT/FreeRates model**


**Fast approaches for support values**


Thanks!