Joseph Felsenstein’s Tree Package

Ian Sammis

Department of Mathematics, UC Berkeley
Basic Ideas

• Phylip (unlike SPLITSTREE or polymake) assumes the input metric represents a tree, and that any deviations from a tree metric are the results of data error.

• Attempts to infer a “best” tree from the data provided.
Two algorithms

- Phylip presents two conceptually different algorithms to construct the tree:
  - “Optimization-based methods” (Fitch & Margioulsh ’67, Cavalli-Sforza & Edwards ’67)
  - Neighbor-joining (Saitou & Nei ’87)
Optimization-based

- Attempts to find a tree minimizing:

\[
\sum_{i} \sum_{j} n_{ij} \frac{(D_{ij} - d_{ij})^2}{D_{ij}^p}
\]

Adds elements to tree in the position that gives the smallest average discrepancy.

Very slow—\(O(n^4)\)
Neighbor-joining

- Uses a “plausible” algorithm to build a tree by separating “close” OTUs (Operational Taxonomic Units) from an initial star configuration.

- Much more efficient than optimization.

- Difficult to analyze (Dr. Felsenstein claims it corresponds to $P=0$, but that this is difficult to prove).
Running Phylip

- Three executables:
  - FITCH—Optimization; no molecular clock
  - KITCH—Optimization; molecular clock included
  - NEIGHBOR—Neighbor-joining and UPGMA (UPGMA assumes a molecular clock).
Running Phylip

- Relatively primitive input-output
- Distance software only a small portion of Phylip (also contains routines to infer distances from sequences).
- Tricky to run in batch mode.

On to the demo!