BIOINFORMATICS APPLICATIONS NOTE Vol. 22 no. 22 2006, pages 2823–2824
doi:10.1093/bioinformatics/btl478

Phylogenetics

Clearcut: a fast implementation of relaxed neighbor joining

Luke Sheneman*, Jason Evans and James A. Foster
Department of Biological Sciences, University of Idaho, Moscow, ID, USA

Received on June 8, 2006; revised on September 5, 2006; accepted on September 6, 2006
Advance Access publication September 18, 2006
Associate Editor: Keith A Crandall

ABSTRACT

Summary: Clearcut is an open source implementation for the relaxed neighbor joining (RNJ) algorithm. While traditional neighbor joining (NJ) remains a popular method for distance-based phylogenetic tree reconstruction, it suffers from a O(N^3) time complexity, where *N* represents the number of taxa in the input. Due to this steep asymptotic time complexity, NJ cannot reasonably handle very large datasets. In contrast, RNJ realizes a typical-case time complexity on the order of N^2logN without any significant qualitative difference in output. RNJ is particularly useful when inferring a very large tree or a large number of trees. In addition, RNJ retains the desirable property that it will always reconstruct the true tree given a matrix of additive pairwise distances. Clearcut implements RNJ as a C program, which takes either a set of aligned sequences or a pre-computed distance matrix as input and produces a phylogenetic tree. Alternatively, Clearcut can reconstruct phylogenies using an extremely fast standard NJ implementation.

Availability: Clearcut source code is available for download at: http://bioinformatics.hungry.com/clearcut

Contact: sheneman@hungry.com

Supplementary information: http://bioinformatics.hungry.com/clearcut

1 INTRODUCTION

Scientists need to infer increasingly large phylogenies. Neighbor joining (NJ) (Saitou and Nei, 1987; Studier and Keppler, 1988) is a popular phylogeny construction algorithm which clusters taxa according to estimated pairwise evolutionary distances. While NJ is largely considered to be a fast algorithm, it cannot efficiently reconstruct extremely large phylogenies. Relaxed neighbor joining (RNJ) (Evans *et al*., 2006) is a very fast variation of NJ which scales better to larger datasets. Both RNJ and NJ share the desirable theoretical property of recovering the true tree if the distance matrix is purely additive (Waterman *et al*., 1977). In the more common case where distances are non-additive, RNJ produces results with negligible differences from those produced by NJ (Evans *et al*., 2006).

Specifically, NJ requires time in O(N^3) for inputs with N taxa (Studier and Keppler, 1988). RNJ requires approximately N^2logN time for typical inputs, though in rare worst case scenarios it degenerates to the same asymptotic runtime as NJ. Thus, RNJ allows users to process larger inputs in less time than NJ, or to bootstrap more trees in the same amount of time.

As the name implies, NJ works by starting with a star-shaped tree and iteratively joining 'neighboring' nodes until a bifurcating tree is constructed. At each step, traditional NJ searches the entire distance matrix and identifies and joins the pair of nodes with the global minimum transformed distance. In contrast, RNJ opportunistically joins any two neighboring nodes immediately after it is determined that the nodes are closer to each other than any other node in the distance matrix. It is not required that the candidate nodes be the closest of all nodes remaining in the matrix. In this sense, our algorithm relaxes the requirement of exhaustively searching the distance matrix at each step to find the closest two nodes to join.

This article announces the availability of Clearcut, which implements both RNJ and a highly optimized version of NJ.

2 METHODS

Clearcut is a small C program that compiles and runs under most UNIX variants, and has been explicitly tested on Linux, FreeBSD, MacOS X and Solaris. It is entirely a text-based program and takes all arguments on the command-line. The source code for Clearcut is freely distributed under the BSD license.

Clearcut implements both relaxed and traditional NJ. It is capable of taking input either in the form of a pre-computed pairwise distance matrix or a set of aligned sequences in FASTA format. When presented with an alignment, Clearcut will compute pairwise distances by first determining the percent identity between all sequence pairs. Optionally, compensation for multiple hits is possible by applying either a Jukes-Cantor correction (Jukes and Cantor, 1969) or Kimura correction (Kimura, 1980) to the pairwise distances. These optional distance corrections can be applied to either DNA or amino acid sequences.

Both NJ and RNJ are sensitive to the order in which distances are input and the order in which nodes are joined. Command-line options allow Clearcut to randomly reorder taxa to mitigate stochastic bias resulting from the original order in which taxa are presented in the input. A similar Clearcut option controls whether attempts to join nodes are done randomly or in a strictly deterministic order. Attempting to join randomly selected nodes can reduce systematic bias in some cases, while it is faster to attempt to join nodes in a deterministic order.

Since RNJ is a non-deterministic algorithm, Clearcut optionally allows the user to quickly generate any number of distinct, equally valid RNJ trees from the same non-additive distance matrix.

3 RESULTS

We compared Clearcut to several popular traditional NJ implementations including PHYLIP Neighbor (Felsenstein, 2004), QuickTree (Howe *et al*., 2002) and QuickJoin (Mailund *et al*., 2004). Our comparison used both simulated sequences and biologically-derived sequences.

For the simulated dataset, we artificially constructed trees of different sizes, which were representative of the two extreme

*To whom correspondence should be addressed.

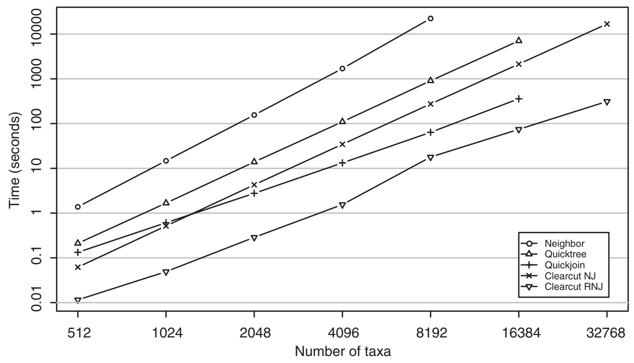

Fig. 1. Speed tests between Clearcut and other NJ programs on simulated distance data demonstrate that Clearcut is dramatically faster for different input sizes. Note the logarithmic scale used on both axes.

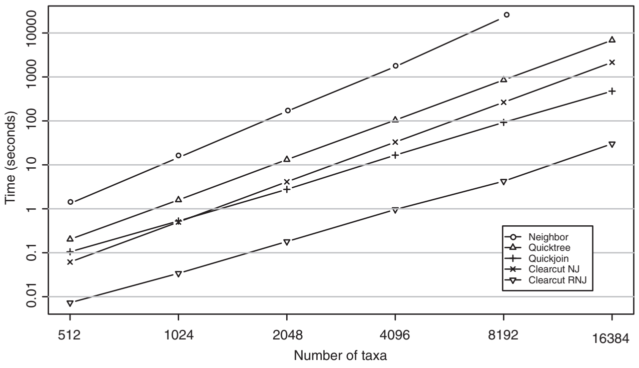

Fig. 2. Speed comparisons using data derived from real rRNA sequences.

tree shapes: maximally deep (pectinate) and maximally shallow (perfect). We stochastically assigned gamma-distributed branch lengths to each branch and then used the simulated tree to construct a purely additive distance matrix.

For the biological sequences, we constructed datasets of various sizes by sampling aligned bacterial rRNA sequences without replacement from RDP-II, the Ribosomal Database Project (Cole *et al.*, 2005). We then used Clearcut itself to generate the distance matrices.

Compared to existing NJ programs, Clearcut's RNJ implementation reconstructed phylogenies in a fraction of the time for all tested tree shapes and sizes as shown in Figures 1 and 2. Clearcut outperformed other implementations by as much as two or three orders of magnitude. Quickjoin, the second fastest NJ implementation, was unable to handle our largest inputs due to its extremely large memory requirements.

Due to rigorous implementation optimizations, especially with respect to cache locality, even Clearcut's traditional NJ implementation is extremely fast.

4 FUTURE ENHANCEMENTS

Future versions of Clearcut will allow users to bootstrap RNJ trees by sampling with replacement from the provided distance matrix. Clearcut will then construct a majority-rule consensus tree with nodal-support values. The labeled consensus tree will be output in Graphviz (Ellson *et al.*, 2003) format.

Future versions of Clearcut will initially compile into a C library before linking into an executable front-end. This will allow Clearcut to be directly embedded and used inside other programs.

ACKNOWLEDGEMENTS

L.S. was partially funded by NIH P20 RR16454 from the INBRE Program of the National Center for Research Resources. J.E. and J.A.F. were partially funded by NIH NCRR 1P20 RR16448.

Conflict of Interest: none declared.

REFERENCES

Cole,J.R. *et al.* (2005) The ribosomal database project (RDP-II): sequences and tools for high-throughput rRNA analysis. *Nucleic Acids Res.*, **33**, D294–D296.

Ellson,J. *et al.* (2003) Graphviz and dynagraph—static and dynamic graph drawing tools. In Junger,M. and Mutzel,P. (eds), *Graph Drawing Software*. Springer-Verlag, Berlin, Heidleberg, New York, pp. 127–148.

Evans,J. *et al.* (2006) Relaxed neighbor-joining: a fast distance-based phylogenetic tree construction method. *J. Mol. Evol.*, **62**, 785–792.

Felsenstein,J. (2004) PHYLIP (phylogeny inference package) version 3.6. Distributed by the author, Department of Genome Sciences, University of Washington, Seattle.

Howe,K. *et al.* (2002) QuickTree: building huge neighbour-joining trees of protein sequences. *Bioinformatics*, **18**, 1546–1547.

Jukes,T.H. and Cantor,C.R. (1969) Evolution of protein molecules. In Munro,H.N. (ed.), *Mammalian Protein Metabolism, chapter 24*. Academic Press, NY, Vol. III, pp. 21–132.

Kimura,M. (1980) A simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. *J. Mol. Evol.*, **16**, 111–120.

Mailund,T. and Pedersen,C.N.S. (2004) QuickJoin—fast neighbour-joining tree reconstruction. *Bioinformatics*, **20**, 3261–3262.

Saitou,N. and Nei,N. (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, **4**, 406–425.

Studier,J.A. and Keppler,K.J. (1988) A note on the neighbor-joining algorithm of Saitou and Nei. *Mol. Biol. Evol.*, **5**, 729–731.

Waterman,M.S. *et al.* (1977) Additive evolutionary trees. *J. Theor. Biol.*, **64**, 199–213.