BIOINFORMATICS APPLICATIONS NOTE

Vol. 21 no. 8 2005, pages 1711–1712
doi:10.1093/bioinformatics/bti195

Phylogenetics

RBT—a tool for building refined Buneman trees

Søren Besenbacher*, Thomas Mailund, Lasse Westh-Nielsen and
Christian N. S. Pedersen

Bioinformatics Research Center, Department of Computer Science, University of Aarhus, Høegh-Guldbergs
Gade 10, Building 090, DK-8000 Århus C, Denmark

Received on October 1, 2004; revised on November 22, 2004; accepted on November 26, 2004
Advance Access publication December 7, 2004

ABSTRACT

Summary: We have developed a tool implementing an efficient algorithm for refined Buneman tree reconstruction. The algorithm—which has the same complexity as the neighbour-joining method and the (plain) Buneman tree construction—enables refined Buneman tree reconstruction on large taxa sets.

Availability: The source code for RBT, written in Java, is available under the GNU Public License (GPL) at http://www.birc.dk/Software/RBT

Contact: besen@daimi.au.dk

The evolutionary relationship for a set of species is often represented by a rooted tree, where leaves correspond to the species and the internal nodes to speciation events. Techniques for reconstructing the correct evolutionary tree from information about the label-species have been intensively studied and several different methods have been proposed. Some methods directly construct rooted trees, whereas others construct unrooted trees and rely on out-groups to position the root. The methods also differ on the type of input they accept—e.g. a matrix of distances between all pairs of taxa, or a DNA or amino acid sequence for each taxon—and on the approach to build the tree—e.g. minimizing the total branch length or the difference between pathlengths in the tree and the distance matrix or minimizing the number of mutations necessary to explain the input sequences (for an overview of different methods, see Nei and Kumar, 2000).

A widely used distance-based method is the neighbour-joining method, developed by Saitou and Nei (1987) and refined by Studier and Keppler (1988), which combines efficiency—$O(n^3)$ time usage and $O(n^2)$ memory usage, where n is the number of taxa—with a reasonable accuracy. St John *et al.* (2003) suggest it as a standard against which new phylogenetic methods should be evaluated. The neighbour-joining method, however, always constructs a fully resolved binary tree, which can be misleading since many internal edges potentially will be artefacts of the method rather than be supported by the input data.

The Buneman tree (Buneman, 1971), shown by Berry and Bryant (1999) to be computable in time $O(n^3)$ and space $O(n^2)$, avoids this problem by only resolving edges with strong support from the input data. The Buneman tree is a conservative, but reliable estimate of the true evolutionary tree. Unfortunately, few splits satisfy the very strict requirement needed to be edges in the Buneman tree, thus the Buneman method resolves only few edges. To alleviate this, Moulton and Steel (1999), proposed the refined Buneman tree, that resolves more edges than the Buneman tree but still requires more support from the input data than does the neighbour-joining tree. The first algorithm for computing the refined Buneman tree, by Bryant and Moulton (1999), with runtime $O(n^6)$, was improved by Berry and Bryant (1999) to time $O(n^5)$ and space $O(n^4)$ and later to time $O(n^3)$ and space $O(n^2)$—the same as the neighbour-joining and the Buneman tree construction—by Brodal *et al.* (2003).

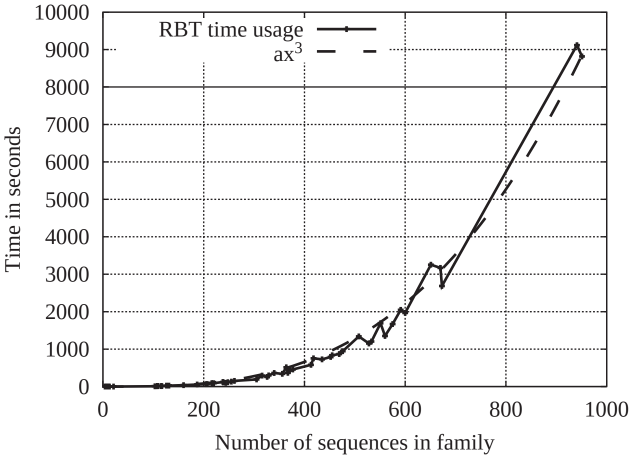

Fig. 1. Time usage of RBT for various sizes of input. The experiments were performed on a standard PC with an Intel Pentium 4 CPU 2.00 GHz and 512 MB RAM running Linux Red Hat 9 with kernel version 2.4.20-31.9. (The fitted coefficient a equals 1.02813×10^{-5}.)

We have implemented this latest algorithm in the tool RBT. The tool is written in Java and should be able to run on any platform for which a Java 1.1.x runtime environment is available. From the specified website, the user can download either an executable jar-file or the entire source code of the program. RBT is a simple command line program that takes a distance matrix in the PHYLIP format as input. Default output from the program is a phylogenetic tree in Newick format, but as an option you can choose to see the resulting tree as a set of splits written as bit-vectors. If you choose to get the output as a set of splits, it is also possible to make the program annotate the splits that, besides being in the refined Buneman tree, also occur in the ordinary Buneman tree. This feature makes it possible to compare

*To whom correspondence should be addressed.

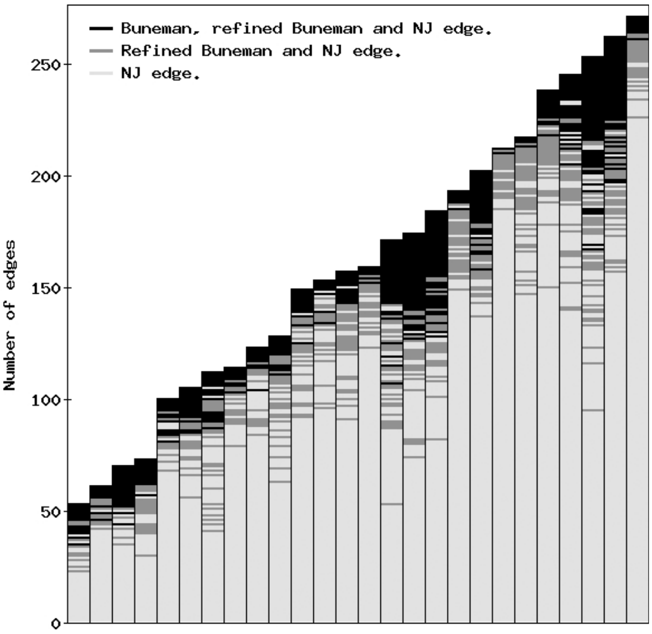

Fig. 2. Comparison of neighbour-joining, refined Buneman and Buneman trees. The figure shows the edges obtained using the three methods, ordered w.r.t. their bootstrap value (obtained using the neighbour-joining method).

the performance of the refined Buneman method and the Buneman method.

To validate that the RBT program obeys the worst case complexity of the underlying algorithm, we used RBT to construct trees from distance matrices obtained from the Pfam database (http://www.sanger.ac.uk/Software/Pfam/). The measured running times are reported in Figure 1. As it can be seen the program runs in time $O(n^3)$ as expected.

With our RBT tool, for the first time it becomes possible to compute refined Buneman trees for realistically sized taxa sets. This makes it possible to compare trees constructed using the refined Buneman method with trees constructed using other methods, e.g. neighbour-joining. It is not within the scope of this application note to present an exhaustive set of such experiments. However, as a preliminary evaluation, we have constructed trees, using the refined Buneman, plain Buneman and neighbour-joining methods, from a set of distance matrices obtained from the Pfam database. For every input (distance matrix), we computed three trees which we compare for shared edges. An edge in the Buneman tree is by definition also an edge in the refined Buneman tree. It is not necessarily the case that an edge in the refined Buneman will also be an edge in the neighbour-joining tree, but it holds for all the trees that we have constructed in this experiment.

Figure 2 visualizes our experimental results. Each vertical bar represents one experiment, i.e. the result of running the three methods on one distance matrix. The size of the vertical bar is proportional to the number of edges in the neighbour-joining tree and one can think of the vertical bar as made of equal-sized boxes each corresponding to one edge in the tree. The boxes (edges) are sorted by the confidence assigned to them in the neighbour-joining method using bootstrapping. Boxes (edges) that are also edges in the Buneman tree are coloured black, and boxes (edges) that are also edges in the refined Buneman tree are coloured dark grey. As evident from the figure, the refined Buneman tree resolves more edges than the plain Buneman tree, while still mainly finding edges with high confidence. Unfortunately, it also appears that, as the input size grows, the refined Buneman method resolves relatively fewer and fewer edges, suffering from the same problem as the plain Buneman tree. Using RBT, we plan to examine this in more detail in future work.

REFERENCES

Berry,V. and Bryant,D. (1999) Faster reliable phylogenetic analysis. In *Proceedings of the RECOMB 1999*, Lyon, France, pp. 59–68.

Brodal,G.S., Fagerberg,R., Östlin,A., Pedersen,C.N.S. and Rao,S.S. (2003) Computing refined buneman trees in cubic time. In *Proceedings of the WABI 2003*, Budapest, Hungary LNCS 2812, pp. 259–270.

Bryant,D. and Moulton,V. (1999) A polynomial algorithm for constructing the refined Buneman tree. *Appl. Math. Lett.*, **12**, 51–56.

Buneman,P. (1971) The recovery of trees from measures of dissimilarity. In: Kendall,D. and Tautu,P. (eds) *Mathematics in Archaeological and Historical Sciences*, Edinburg University Press, Edinburg, pp. 387–395.

Moulton,V. and Steel,M. (1999) Retractions of finite distance functions onto tree metrics. *Discrete Appl. Math.*, **91**, 215–233.

Nei,N. and Kumar,S. (2000) *Molecular Evolution and Phylogenetics*. Oxford University Press, Oxford.

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

St John,K., Warnow,T., Moret,B. and Vawter,L. (2003) Performance study of phylogenetic methods: (unweighted) quartet methods and neighbor-joining. *J. Algorithms*, **48**, 173–193.

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