Phylograph: Real-time Interactive Visualization of Phylogenetic Searches

Jesse Mecham, Mark Clement, Quinn Snell, Keith Crandall

Department of Computer Science, Brigham Young University

Abstract

Phylogenetic analysis is an important tool in human epidemiology, viral transmissions and systematics. As larger data sets are assembled, it becomes more and more difficult to analyze this sequenced data using existing algorithms. The *Phylograph* tool provides a way for users and algorithm developers to visualize the phylogenetic search space and an algorithm's search strategy. It also allows users to guide the application into areas of the search space where better solutions may be found. Phylograph will improve our understanding of phylogenetic algorithms and provide increased control and efficiency for phylogenetic applications.

Keywords: phylogenetics, visualization, interactive

1. Introduction

Phylogenetic analysis has become an integral part of many biological research programs. These include such diverse areas as human epidemiology [1,2], viral transmission [3], biogeography [4], and systematics [5]. With the advent of high speed sequencing equipment, an increasingly large volume of sequence data is becoming available. Scientists should be able to take advantage of this data and also of the research that others have performed. For example, when a new virus is detected, it should be possible to build a phylogenetic tree (an evolutionary history) containing all related viruses and the unknown variety in order to answer questions such as:

- Where did this virus come from?
- When did this virus arrive in the human population?
- What are the related species from which we might derive ideas about appropriate antibodies for testing and remedies for treatment?
- Has this virus been genetically modified through natural or human induced recombinant technology?
- How is this virus evolving and what genetic changes occurred to allow it to successfully enter the human population?

Unfortunately, this kind of phylogenetic search is currently computationally infeasible for large data sets. The time it takes to perform a complete search using maximum likelihood exceeds several months with even a small number of taxa. In the case of the SARS epidemic, and others like it, treatment information must be available in days or at most a few weeks in order for appropriate action to be taken. Much of the problem comes from the culture and software design of most phylogenetic software packages [6, 7, 8].

As the size of phylogenetic data sets increases, the number valid trees increases according to the following formula where *n* is the number of sequences.

$$\prod_{k=1}^{n-2} (2k-1)$$

For 50 sequences, there are 3×10^{74} trees in the search space and since phylogenetic search is an NP-COMPLETE problem, there is little hope of finding an exact algorithm to solve it in a reasonable amount of time. The size of the search space also makes it difficult to visualize the problem and to understand why existing heuristics may be inefficient in their search.

Phylograph provides a novel visualization of the search space that provides feedback to researchers in the following ways:

- Algorithm Analysis. Through understanding where an algorithm is searching, application developers can compare existing strategies and develop new ones.
- Computation Steering. Given a graphical representation of the search space with an indication of where the application has

searched so far, the user can use his domain specific knowledge to guide the application into regions where better trees may be found. A user may also notice that there are regions that have not been searched sufficiently. Phylograph allows the user to guide the application to search in these regions.

- Accuracy. Phylograph can determine how thoroughly the search space has been examined and can then estimate how accurate the current tree is based on the coverage of the search space.

1.1. Related Work

In the past few years there has been an increasing interest in visualizing tree searches. Much of this work has been focused on attaching a heuristic to the trees returned (Robinson-Foulds, neighbor interchange distance, or tree bisection and reconnection distance) and then using a method such as multi-dimensional scaling or principal component analysis to graph the diversity [10, 11]. These methods allow the researcher to form some concept of various relationships within the trees that have been searched, such as observing various islanding effects.

While these methods do offer some advantages, they all rely upon some type of heuristic to estimate the topology they have covered. This may in some instances lead the researcher to believe he has covered most of the space, when in fact, he has not. Additionally, since all of these methods rely upon a relative inter-tree metric, there is no one-to-one mapping of the search space. This means that while the researcher can watch the search taking place, there is no way to guide the search by hand to move to a new location.

2. Phylograph Details

Several decisions were made as phylograph was developed in order to create a architecture where the goals of algorithm analysis, computational steering and accuracy analysis could be met.

2.1. Axes Selection

We decided to base the search space on properties of the tree rather than a relational metric. By assigning each tree a shape, permutation and tree score within the scope of a particular granularity, it is possible to construct a three dimensional space which can be visualized (See Section 3 for examples). Each tree below is represented by a dot assigned a specific shape, score, and permutation.

Shape: Let T be set of all valid bifurcating unrooted trees that contain N terminal nodes and E edges. For any tree t in T, we can root the tree by choosing any node n in N and defining that as the root node. Next we arrange tree by descending through tree t and evaluating all nodes beginning at node n. If n_x is an inner node, evaluate the height h of both child nodes (c_1 and c_2) of n_x. If the $h_1 > h_2$, swap c_1 and c_2. The result is a tree which is arranged with the deepest inner nodes on the right. It now becomes possible to assign a unique shape S to any tree t. Tree t is then encoded with a binary scheme b in which every inner node is assigned a '0' and every terminal node is assigned a '1' while parsing the tree from node n in a preorder traversal fashion. By using b, it is possible to assign a canonical order and tree shape index to any tree shape.

Permutation: Let X be the set of all taxa in tree t. Let P be the ordered set of all permutations on X. By reading the bottom, terminal nodes of the rooted tree t in a left to right fashion, it is possible to assign a permutation p to t.

Score: Parsimony score assigned to individual tree.

Granularity: While it is possible to plot every unique tree as defined by shape and permutation, it is impractical because of limitations in human perception. Therefore, using the graphing tool, it is possible to assign various levels of granularity to a graphing model, where only small portions of the tree shape or permutation are mapped. This essentially assigns every tree into a "bucket" where it resides with other trees of similar shape or permutation. The granularity of the bucket can be increased (thus allowing a researcher to zoom into an area to investigate a particular region), but this comes at the expense of loosing visibility on area outside the spectrum of investigation.

3. Results

In order to demonstrate some of the capabilities of Phylograph, a subset of the Zilla data set was analyzed using the Soda [12] phylogenetic package. Both TBR and Ratchet algorithm searches were run while Phylograph recorded all of the intermediate trees generated and displayed them using gnuplot. The

program was run until it finished at approximately 400,000 trees (Fig. 1 – Fig. 8).

Since the plotting is done in real-time, it is possible to follow the progress of the search. This allows the researcher to gain some understanding into how the algorithm is functioning exploring the space. In the graphs below, it is observed that the search travels downwards along a narrow range of permutations (Fig. 6 and Fig. 7 marked by arrows). This suggests that at the set granularity, the search is making greater changes to tree shape than tree permutation, yet still jumps periodically to a completely new permutation.

Another important property visible from the graph is the ability to see where the search is spending most of its computational time. In Fig. 5 and Fig. 6 it becomes apparent that the Ratchet algorithm moves to a much lower score (2600 vs. 4500) more quickly than the TBR search and then spends more time searching in regions with better trees. This kind of data could be useful to a researcher who is trying to determine why one algorithm outperforms another.

By selecting a coordinate in the graph, it is possible to generate a tree which can be fed back into the search algorithm and used as a starting point to continue the search. This allows the researcher the ability to utilize his own expertise by using the mouse to manipulate the search. Fig. 7 illustrates how a user might be able to interactively guide the computation by hand.

3.1. Future Work

The effects of granularity on the graph have yet to be explored fully. While there is a canonical ordering for permutations and shape, further work must be performed to select an ordering that groups similar trees along the axes.

4. Conclusions

As the size of phylogenetic data sets increases, the amount of space that can be searched rapidly decreases. This places an increasing responsibility on the researcher to determine how long to search and estimate the quality of the search. Using the phylograph tool, a researcher can infer a number of properties about a search that may help in analyzing the effectiveness of a particular algorithm. By allowing interactive guidance, the expertise of the researcher is introduced into the search, allowing the researcher to manipulate algorithms in a unique way. It is hoped that this tool will help researchers have greater confidence in their results and a greater feeling of control in their phylogenetic searches.

5. References

[1] Clark, A. *et al.* (1998) Haplotype structure and population genetic inferences from nucleotide-sequence variation in juman lipoprotein lipase. *American Journal of Human Genetics,* 63:595-612

[2] Sing, C., Haviland, M., Zerba, K., Templeton, A. (1992) Application of cladistics to the analysis of genotype-phenotype relationships. *European Journal of Epidemiology,* 8:3-9

[3] Crandall, K., (1996) Multiple interspecies transmissions of human and simian T-cell leukemia/lymphoma virus type I sequences. *Molecular Biology and Evolution,* 13:115-131

[4] DeSalle, R., (1995) Molecular approaches to biogeographic analysis of Hawaiian Drosophilidae. *Hawaiian Biogeography* (ed. by W.L. Wagner and V.A. Funk) Smithsonian Institute Press, pages 72-89

[5] Hillis, D., Miritz, C., Mable, B., (1996) Molecular Systematics, Sinauer Assc. Sunderland.

[6] Swofford, D., (1993) PAUP: Phylogenetic Analysis Using Parsimony, Washington DC: Smithsonian Institution, http://paup.csit.fsu.edu

[7] Goloboff, P., (1997) NONA, Available via FTP with registration from Willi Hennig Society.

[8] Felsenstein, J. 2002. PHYLIP, version 3.6. Department of Genome Sciences, University of Washington.

[9] Maddison, W. P. and D.R. Maddison. 2004. Mesquite: a modular system for evolutionary analysis. Version 1.05 http://mesquiteproject.org

[10] Amenta, N. and Klingner, J. Case Study: Visualizing Sets of Evolutionary Trees. *Proceedings of the IEEE Symposium on Information Visualization 2002 (InfoVIs 2002).* pp. 71-74, 2002.

[11] Montealegre I. and St. John, K. Visualizing Restricted Landscapes of Phylogenetic Trees. http://www.cs.utexas.edu/users/amenta/pubs/treeviz.pdf

[12] Tewk, K., Snell, Q., and Sneadon, D. (2004) Soda, Department of Computer Science, Brigham Young University. http://phylo.byu.edu/soda/

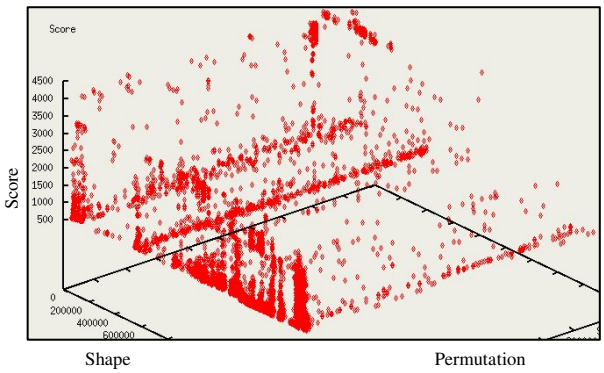

Fig. 1: (TBR) Score vs. shape vs. permutation

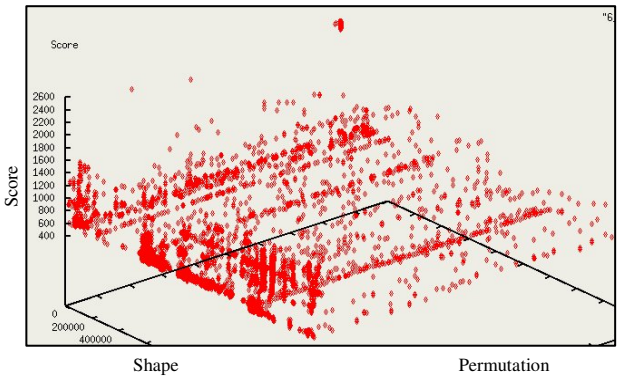

Fig. 2: (Ratchet) Score vs. shape vs. permutation

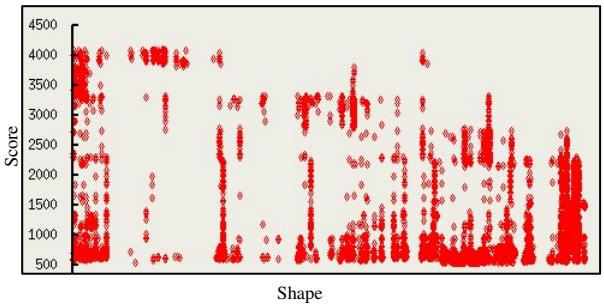

Fig. 3: (TBR) Shape vs. score

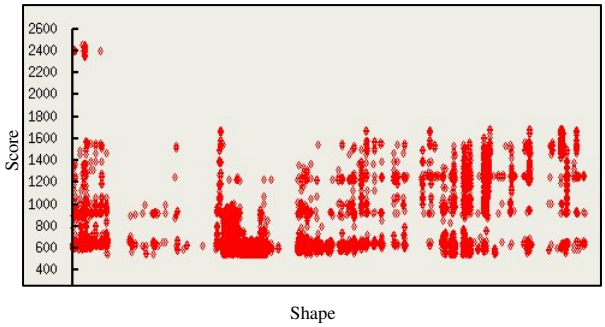

Fig. 4: (Ratchet) Shape vs. score

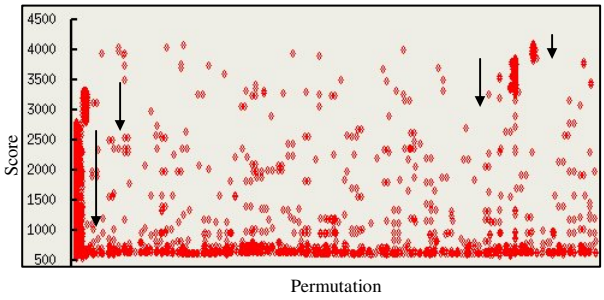

Fig. 5: (TBR) Permutation vs. score - arrows show search progress

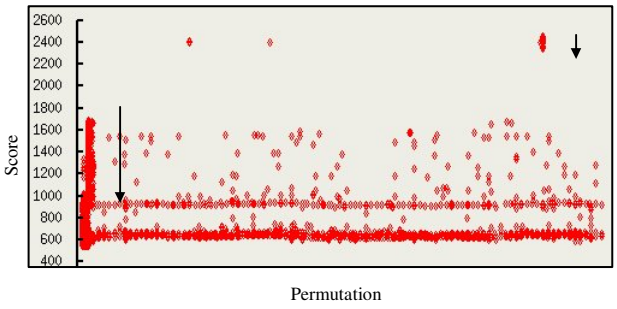

Fig 6: (Ratchet) Permutation vs. score - arrows show search progress

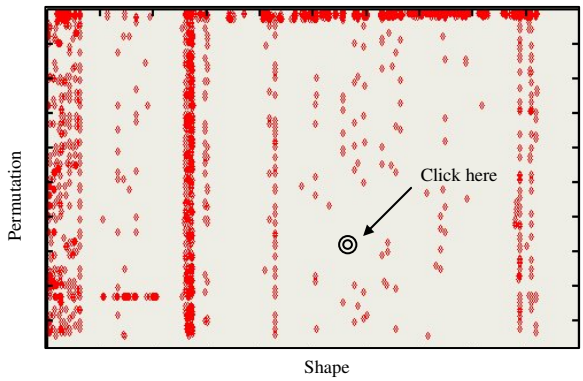

Fig. 7: (TBR) Permutation vs. shape – shows location where user could select to continue search from

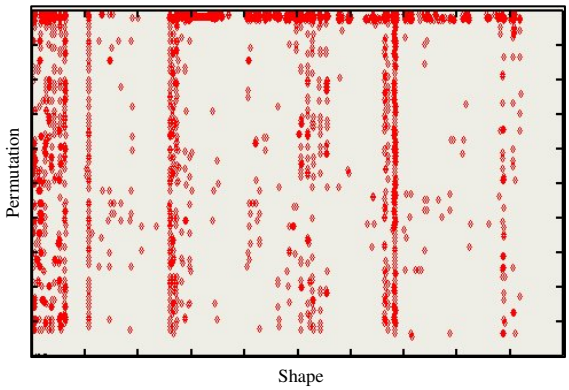

Fig. 8: (Ratchet) Permutation vs. shape