

PTC: An Interactive Tool for Phylogenetic Tree Construction

Chen Yang and Sami Khuri
Department of Computer Science
San Jose State University
cherryyang@yahoo.com
khuri@cs.sjsu.edu

Abstract

A phylogenetic tree represents the evolutionary history of a group of organisms. In this work, we introduce a novel interactive tool for constructing phylogenetic trees, Phylogenetic Tree Construction package. The package supports four well-known algorithms, Unweighted Pair Group Method using Arithmetic average, Neighbor Joining, Fitch Margoliash, and Maximum Parsimony.

1. Introduction

A phylogenetic tree represents the evolutionary history of a group of organisms. Constructing phylogenetic trees is a crucial step for biologists to find out how today's species are related to one another in terms of common ancestors. Numerous computer tools have been developed to construct such trees, such as PHYLIP and PAUP.

In this work, we introduce a novel interactive tool for constructing phylogenetic trees, Phylogenetic Tree Construction package (PTC). PTC currently supports four well-known algorithms, Unweighted Pair Group Method using Arithmetic Average (UPGMA), Neighbor Joining (NJ), Fitch Margoliash (FM), and Maximum Parsimony (MP). The reason behind our project is a lack of interactivity in existing tools. The existing packages, in contrast to our tool, only visualize the resulting phylogenetic tree. Furthermore, the interaction with these packages occurs only at the beginning, before the program's execution. We strongly believe that interactive tree construction can be extremely valuable to bioinformaticians. Through the interaction with PTC, users can gain a deeper understanding of the algorithms. We provide the capability to edit input data, view tree construction step-by-step, and compare two consecutive states of the algorithm. Trees are dynamically drawn and can be

resized by the user. PTC is implemented in Java and can be extended to include additional algorithms.

2. Phylogenetic Tree Construction Package

The algorithms implemented in PTC use both, character and distance-based techniques for building phylogenetic trees. The character-based techniques use the individual substitutions among the sequences, while distance-based algorithms first calculate the pairwise distances between the sequences and then construct trees.

UPGMA is a sequential clustering algorithm. It works by clustering the sequences, at each stage amalgamating two Operational Taxonomy Units (OTUs) and at the same time creating a new node in the tree. UPGMA produces a rooted tree. The construction is bottom-up, from the leaves to the root node. UPGMA implicitly assumes the existence of an ultrametric tree: the total branch lengths from the root to any leaf are all equal to one another. In other words, there is a "molecular clock", which ticks at a constant pace, and all the observed species are at an equal number of ticks from the root; the same evolution rate applies to all branches, which is often not the case. The Neighbor Joining algorithm attempts to correct the shortcoming of UPGMA. NJ is a heuristic greedy algorithm. It begins with a star tree and at each stage two closest neighbors are joined into one new node, thus reducing the number of OTUs by 1. The process repeats until there are two OTUs left. Unlike UPGMA, which chooses the neighbors with minimum distance, NJ chooses the neighbors, which minimize the sum of branch lengths at each stage. It is fast and well suited for datasets of substantial size and also for the post-processing step of bootstrap analysis. It is especially suitable when the rate of evolution of the separate lineages under consideration varies. When the branch lengths of trees of known topology are allowed to vary in a manner that simulates varying levels of

evolutionary change, NJ is the most reliable method in predicting the correct tree.

UPGMA and NJ assume the additive property. Unfortunately, in real life, distances are rarely additive. In this case, we try to find the trees, which best fit the distance data. Thus, we try to optimize an objective function that quantifies the degree of "distortion" between the final tree path lengths and the observed distances. The Fitch Margoliash algorithm assumes that the expected error is proportional to the square root of the observed distances. The disadvantage is that FM requires longer execution time than UPGMA and NJ.

PTC also supports the character-based Maximum Parsimony algorithm. The principle of maximum parsimony wants us to look for trees that give the smallest number of evolutionary changes among the OTUs. An evolutionary change is the transformation from one character state to another, such as one DNA base to another, or the loss or gain of a restricted site, or the absence or presence of morphological features. MP allows the use of all known evolutionary information in tree building. It produces numerous unrooted, "most parsimonious trees".

Although all these algorithms have been implemented in different tree building programs, PTC is a novel tool, which not only constructs phylogenetic trees, but also visualizes the steps of each algorithm. The bioninformatics educators can use our package to explain the underlying tree building algorithms and the researchers can use our package to examine the trees produced by different methods.

As can be seen in Figure 1, the user interface of PTC consists of six main components: a menu, an input data area, buttons, an intermediate result area, an instruction area, and a phylogenetic tree area. For FM and MP, when one or more trees can be identified, some additional windows will be popped up to show the best trees. The key features of PTC are:

1. Allow users to run and trace the UPGMA, NJ, FM, and MP algorithms.
2. Visualize the steps of building the trees.
3. Allow users to choose different input data.
4. Can go back to previous state of the algorithms.
5. Provide feedback by using pop-up questions.
6. Can run on many hardware platforms.
7. Allow future developers to easily incorporate new algorithms into PTC.

3. Conclusion

We developed the PTC package using the principle of Object Oriented Design and guidelines for developing effective algorithm visualization. PTC is developed in Java, which has rich methods for designing user interfaces. The input of PTC is flexible. It can be read from a file, typed directly into the table, or randomly generated. The controls of PTC are menus and buttons, which are simple and easy to use. The instructions tell users how the trees are constructed in each step. The previous state window allows the user to compare two consecutive states of the algorithm. Pop-up questions are generated to provide instant feedback to the user. Different colors and shapes are used to differentiate between interior nodes and leaves.

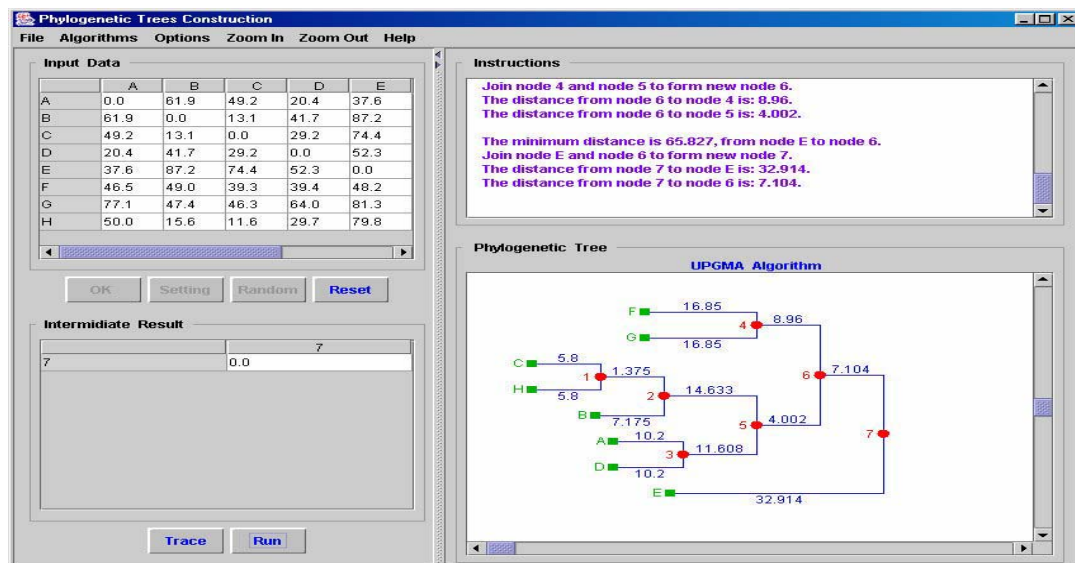


Figure 1: The User Interface of the PTC Package