PSODA: Better Tasting and Less Filling Than PAUP

Hyrum Carroll
Computer Science Department
Brigham Young University
Provo, Utah 84602, USA
hdc@cs.byu.edu

Mark Ebbert
Biology Department and
Computer Science Department
Brigham Young University
Provo, Utah 84602, USA
marktwe@byu.net

Mark Clement and Quinn Snell
Computer Science Department
Brigham Young University
Provo, Utah 84602, USA
{clement,snell}@cs.byu.edu

***Abstract*—PSODA is an open-source phylogenetic search application that implements traditional parsimony and likelihood search techniques as well as advanced search algorithms. PSODA is compatible with PAUP and the search algorithms are competitive with those in PAUP. PSODA also adds a basic scripting language to the PAUP block, making it possible to easily create advanced meta-searches. Additionally, PSODA provides a user-friendly GUI with real-time graphing visualizations and phylogeny viewer, and a multiple sequence alignment algorithm PSODA is freely available from the PSODA web site: http://csl.cs.byu.edu/psoda.**

I. Introduction

A high quality phylogeny, or evolutionary tree, is important to accurately determine the relationships between species. Phylogenies have been in use for over a hundred years and software has been employed to produce better phylogenies for a couple of decades. These applications are among the most frequently cited papers in the field of bioinformatics with over 10,000 citations for both PHYLIP [1] and PAUP* [2] (according to scholar.google.com). Although many, many people use these applications, they have not been thoroughly maintained in the past several years. Furthermore, most existing phylogenetic reconstruction packages either have a licensing fee and are not extendible to experiment with new algorithms and methods or have serious performance limitations. This work presents an open source phylogenetic search package that is free to use and has performance comparable to PAUP.

II. Related Works

Due to the advantages of using computer algorithms to perform phylogenetic reconstruction, several software packages have emerged (e.g., PHYLIP [1], PAUP* [2] and TNT [3]). PHYLIP was first released in 1980 by Joe Felsenstein and is one of the first programs to perform Maximum Likelihood (ML) searches. It is an open source package that focuses on ML, but also allows for Maximum Parsimony (MP) searches.

PAUP* (Phylogenetic Analysis Using Parsimony *and other methods) is believed to be the most widely used phylogenetic search program. It is both feature-rich (analysis in both MP and ML) and robust. Unfortunately, it is proprietary software and requires a licensing fee to use.

Another phylogenetic search package is TNT (Tree analysis using New Technology), written by Pablo Goloboff, Steve Farris, and Kevin Nixon. TNT performs parsimony searches remarkably fast using several heuristic searches. TNT is also proprietary and requires a licensing fee to use.

While several phylogenetic reconstruction packages exist, they are either too slow for analysis on medium to large data sets and / or are proprietary.

III. Features

Analyzing phylogenetic trees currently requires the use of several different programs. PSODA bridges the gap and brings the many features necessary to analyze phylogenetic trees together in one package. Some of the features PSODA provides are:

- Cost
- Open-Source Code
- Performance
- Models of Analysis
- Graphical User Interface
- Cross-platform architecture
- Input format
- Multiple Sequence Alignment
- PsodaScript

A. Cost

First of all, PSODA does not require a fee or subscription to use. There is no trial period which expires–it is simply free. Not requiring a fee or subscription can be considered as a variable length trial period for researchers to use the program. During this time, they can verify that PSODA meets their needs. The lack of a cost also allows all organizations (under-funded or not) to perform phylogenetic analysis.

B. Open-Source Code

PSODA uses open-source code licensed under the GNU General Public License, Version 2 (see http://www.gnu.org/licenses/old-licenses/gpl-2.0.html). This license allows others not only to collaborate and make improvements to the package, but also to extend it and perform algorithmic experiments with a stable code foundation. We envision many researchers finally being able to implement, in code, concepts that they've envisioned, but have not implemented due to the hurdles of developing a fast and reliable foundation of code. Example modifications include a different enumeration of topologies

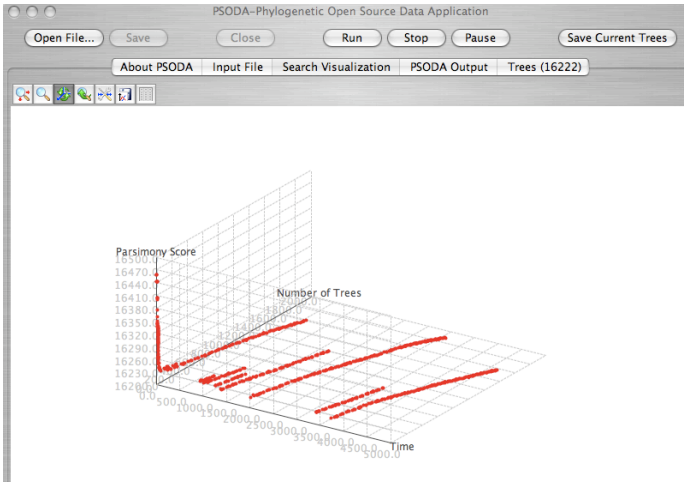

Fig. 1. PSODA GUI illustrating the search results (tree scores and number of trees over time).

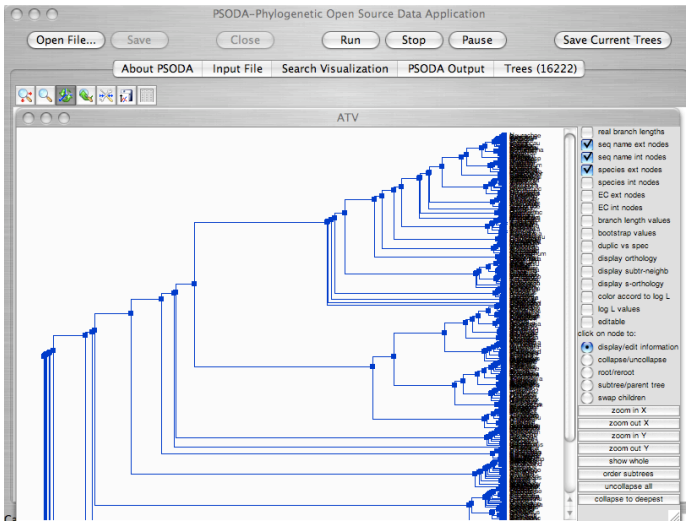

Fig. 2. PSODA GUI using ATV [8] to display a phylogeny.

during a TBR search and integrating both Maximum Likelihood and Maximum Parsimony into a single search algorithm (see [4]). To implement the latter would require minor changes and additions to the existing code.

C. Performance

The performance of a phylogenetic search application is usually measured by the phylogeny scores that it achieves over time. PSODA has comparable performance to other phylogenetic search packages in terms of the trees scores obtained over time. For a more detailed treatment of PSODA's performance, see the Results section.

D. Models of analysis

The two main models of phylogenetic analysis are Maximum Parsimony [5] and Maximum Likelihood [6]. MP has been used for phylogeny analysis longer than ML. It is based on the same principles as Occam's razor – the simplest solution is the best solution. Joe Felsenstein fathered the ML movement when he discovered inconsistencies with MP when long branches are present in a phylogeny. ML uses different models of evolution. PSODA allows users to perform both Maximum Parsimony and Maximum Likelihood searches. For ML searches, PSODA uses the F84 model [7].

E. Graphical User Interface

PSODA's graphical user interface (GUI) (see Figures 1 and 2) is an important part of making PSODA user-friendly and portable. All of the other programs used in searching tree space have only a command-line interface, or the GUI that is available only works on an older operating systems. Many users prefer the option of being able to interact with a program using the GUI on a variety operating system. To met this need, PSODA's GUI uses the Java programming language, giving it portability without further installations of libraries.

There are several features offered in PSODA's GUI in order to facilitate many of the tasks required to run and analyze datasets. Three of the most valuable features in the GUI are: data format conversion, a 3D visualization of the progress of the search and an individual phylogeny viewer. Each of these features are discussed in this section.

1) Data Format Conversion: Several formats currently exist for genetic and phylogeny data such as PHYLIP [1], Clustal [9], MEGA [10], NEXUS [11], and FASTA [12]; however, none of the standard programs accept all of these formats, nor do they perform all types of analyses common for phylogenetic tree. To do so requires using multiple programs and multiple file formats. The process of converting from one format to another can be difficult for many users, so DataConvert (David McClellan, http://biology.byu.edu/faculty/dam83/cdm/), which is capable of converting each file format to any other, is included in PSODA. When converting the formats, DataConvert offers the option of interleaving the sequences or leaving them discrete. Also, if there is an entire directory of files that needs to be converted, DataConvert can convert all of those files instead of requiring the user to convert each file individually.

2) 3D Search Visualization: While tree space visualization is a current topic of research, PSODA provides a 3D graph of the progress of the search (see Figure 1). The default graph's axes are elapsed time, tree score and the number of trees found of the score. The graph is updated in real-time. Such a visualization can provide insights into the progress of the search and when enough searching is enough. Since PSODA is open source, it is possible for others to contribute by defining new dimensions to better map the phylogenetic tree space.

3) Phylogeny Viewer: To analyze a specific tree for biological accuracy it is necessary to view it. While there are several tree viewers available, other phylogeny search applications do not integrate a viewer into their program. To view the saved trees from a search often requires converting to a new format specific to the tree viewer of choice. Integrated into PSODA is ATV (A Tree Viewer) [8]. ATV is a powerful tree

```
BEGIN PSODA;
    hsearch (start=stepwise, nreps=5);

    while (true)
        hsearch (start=current);
        align (guidetree=best);
    endwhile;
end;
```

Fig. 3. An example PSODA block using PsodaScript: iterations of phylogeny search and multiple sequence alignment. NOTE: PSODA also recognizes BEGIN PAUP to start the block.

viewer written in Java, and therefore provides the same level of portability enjoyed by PSODA. Among the most useful features of ATV are its ability to view trees with a large number of taxa, view branch lengths and zoom in and out.

F. Cross-platform architecture

PSODA has been carefully designed to run on the most popular operating systems. Executable binaries of PSODA for Mac OS X, Linux and Windows operating systems are available from the PSODA website, http://csl.cs.byu.edu/psoda. Additionally, the source code is also available for contribution and modification.

G. Input format

PSODA uses the NEXUS format for inputting sequences, trees and commands. The format is familiar to many researchers, and there is a wealth of supporting tools that exist to convert existing data and create new NEXUS files. Additionally, PSODA allows auxiliary input beyond the NEXUS format to handle features not present in other phylogenetic search applications. An example of this is unaligned data. Currently, the NEXUS format does not support unaligned data (unless the sequences are all the same length). PSODA uses unaligned data to perform a progressive multiple sequence alignment.

H. Multiple Sequence Alignment

While most phylogenetic search applications only handle previously aligned data sets, PSODA also can perform a progressive multiple sequence alignment [13] of unaligned data. Given a guide tree, PSODA traverses the tree, aligning the most closely related sequences first (the leaves of the tree). After those sequences are aligned, it aligns alignments of sequences. This continues and results in an alignment of all the sequences. PSODA uses the Needleman-Wunsch algorithm [14] to perform the alignments. Including an alignment algorithm in a phylogenetic search application allows for interesting combinations of alignment and phylogeny search to be efficiently combined.

I. PsodaScript

All of the standard search programs available require specific settings, which become numerous, in order to accomplish normal and specialized searches; but even then it is not always possible to produce certain types of searches. PSODA has a built-in language that allows users to create searches of their own design [15]. For example, with the PSODA language you can develop a specialized search that iterates between phylogeny searches and multiple sequence alignments [16], [17] as shown in Figure 3.

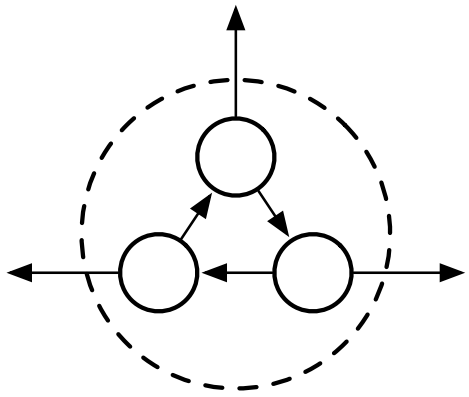

Fig. 4. A set of QNode objects used to represent a vertex in a phylogeny.

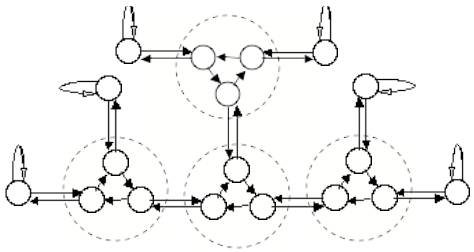

Fig. 5. PSODA's internal representation of a phylogeny.

IV. Implementation Details

PSODA represents phylogenies internally as unrooted. Each vertex of the tree consists of three QNode objects (see Figure 4). Each of these objects has an internal pointer to other QNodes in the vertex, and an external pointer to a QNode object. The direction of the parent and children of this vertex depends on how the tree is viewed. Each external pointer can point to a parent or a child in the tree structure. To make a tree, the external pointers from two vertices are connected. This structure allows for flexible traversal and optimizations. Single QNode objects represent leaves of the tree. This representation of trees is similar to that of PHYLIP.

To create an unrooted phylogeny, multiple QNodes are linked together via their external pointers. Figure 5 illustrates how several QNode objects are connected together to represent a phylogeny such as ((A,B),(C,D),(E,F)). Phylogenies of virtually any size can be represented using this architecture.

PSODA executes both Maximum Parsimony and Maximum Likelihood searches using Tree Bisection and Reconnection (TBR). TBR works by removing a branch of a phylogeny, thereby creating two subtrees. The first subtree is re-connected at various points to the other subtree. Each internal branch from the first subtree is re-connected at every possible location

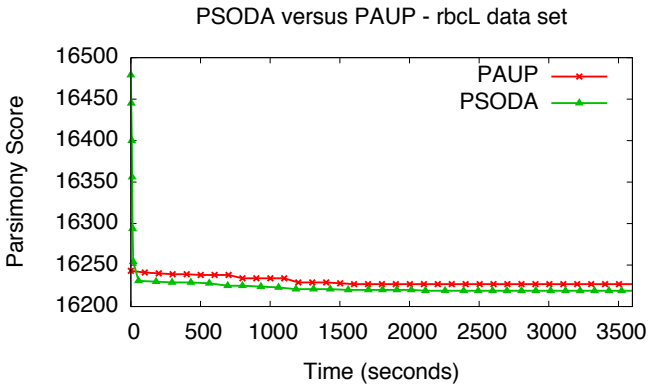

Fig. 6. Performance results, in terms of best parsimony score found over time, for a parsimony search using PSODA and PAUP* on the 500 taxa seed plant rbcL data set [25]. Note: lower parsimony score is better.

in the second subtree. If the two subtrees have N_1 and N_2 species, then there are $(2N_1 - 3)(2N_2 - 3)$ possible rearrangements (including the original one) [18]. Using TBR to search for phylogenies also facilitates such heuristics as the ratchet [19] and similar derivatives.

Using three QNodes to represent nodes of a phylogeny facilitates the implementation of several optimizations [20], [21], [22], [23], [24]. For example, during parsimony searches in PSODA, tree rearrangements are evaluated by their *views* (see [21] and [24]). Evaluating the views of the new phylogeny found by TBR allows the phylogeny to be scored by just summing the parsimony score of the two views and the new join point. The overhead of this optimization is pre-processing the phylogeny before a search and storing the parsimony scores for each subtree. Pre-processing the phylogeny takes slightly longer than completely scoring the phylogeny, allowing this optimization to remarkably speed up TBR searches.

As mentioned in the Features section, the GUI is written in JNI and Java for portability and the underpinnings are written in object-oriented C++.

V. Results

PSODA performs tree searches comparable to other phylogenetic search packages. The results presented here were run at the Ira and Mary Lou Fulton Supercomputing Laboratory at Brigham Young University. Each node of the computers used has two Dual-core Intel Xeon EM64T processors (2.6GHz) and 8 GB of memory. The data set, *rcbL* [25], is comprised of 500 plant seed taxa, each with a length of 759 sites. It is the most studied data set in systematics. Figure 6 illustrates the results of running a TBR search with PSODA and PAUP on the rcbL data set. While PAUP initially has better performance (the first 20 seconds), PSODA quickly catches up and surpasses PAUP. The best parsimony score found by PAUP is 16,227 (after 1,507 seconds). PSODA finds a phylogeny with a better (of 16,226) after only 653 seconds. Furthermore, it is interesting to note that the best parsimony tree score published for this data set is 16,218 [19], and PSODA achieved 16,219 (after 2,033 seconds) with simpler methods than those used elsewhere.

VI. Conclusion

PSODA is an open-source phylogeny reconstruction package made freely available to the public. It implements traditional search algorithms for Maximum Parsimony and Maximum Likelihood as well as more advanced search techniques. It also provides a user-friendly GUI, and is available for more operating systems. The input format is compatible with PAUP. Furthermore, PSODA's performance is comparable with PAUP. Finally, PSODA has several features unique to itself, such as integrated graphing visualizations, a multiple sequence alignment algorithm and a scripting language.

Acknowledgments

We would like to thank the beta testers of PSODA for their insightful comments. This material is based upon work supported by the National Science Foundation under Grant No. 0120718.

References

[1] J. Felsenstein, "PHYLIP (Phylogeny Inference Package) version 3.57 c," *Department of Genetics, University of Washington, Seattle*, 1995.
[2] D. L. Swofford, *PAUP*. Phylogenetic Analysis Using Parsimony (* and Other Methods). Version 4*. Sunderland, Massachusetts: Sinauer Associates, 2003.
[3] P. Goloboff, S. Farris, and K. Nixon, "TNT: Tree analysis using new technology," http://www.cladistics.com/webtnt.html, 2001.
[4] K. Sundberg, T. O'Connor, H. Carroll, M. Clement, and Q. Snell, "Using parsimony to guide maximum likelihood searches," in press.
[5] J. Camin and R. Sokal, "A Method for Deducing Branching Sequences in Phylogeny," *Evolution*, vol. 19, no. 3, pp. 311–326, 1965.
[6] J. Felsenstein, "Evolutionary Trees from DNA Sequences: A Maximum Likelihood Approach," *Journal of Molecular Evolution*, vol. 17, no. 6, pp. 368–376, 1981.
[7] H. Kishino and M. Hasegawa, "Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data, and the branching order in hominoidea," *Journal of Molecular Evolution*, vol. 29, no. 2, pp. 170–179, 1989.
[8] C. M. Zmasek and S. R. Eddy, "ATV: display and manipulation of annotated phylogenetic trees," *Bioinformatics*, vol. 17, no. 4, pp. 383–384, 2001.
[9] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, pp. 4673–4680, 1994.
[10] S. Kumar, K. Tamura, and M. Nei, "MEGA: Molecular Evolutionary Genetics Analysis software for microcomputers," *Bioinformatics*, vol. 10, pp. 189–91, 1994.
[11] D. R. Maddison, D. L. Swofford, and W. P. Maddison, "NEXUS: An Extensible File Format for Systematic Information," *Systematic Biology*, vol. 46, no. 4, pp. 590–621, 1997.
[12] D. J. Lipman and W. R. Pearson, "Rapid and sensitive protein similarity searches," *Science*, vol. 227, no. 4693, p. 1435, 1985.
[13] D.-F. Feng and R. F. Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees," *Journal of Molecular Evolution*, vol. 25, no. 4, pp. 351–360, 1987.
[14] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.
[15] J. L. Krein, A. R. Teichert, H. D. Carroll, M. J. Clement, and Q. O. Snell, "PsodaScript: Applying Advanced Language Constructs to Open-source Phylogenetic Search," October 2007, in press.
[16] O. Gotoh, "Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments," *J. Mol. Biol*, vol. 264, no. 4, pp. 823–838, 1996.

[17] P. G. Ridge, H. D. Carroll, D. Sneddon, M. J. Clement, and Q. O. Snell, "Large Grain Size Stochastic Optimization Alignment," in *Proceedings of the Sixth IEEE Symposium on BionInformatics and BioEngineering (BIBE'06)*. IEEE Computer Society Washington, DC, USA, 2006, pp. 127–134, ridge and Carroll are equally contributing authors.
[18] J. Felsenstein, *Inferring Phylogenies*. Sunderland, MA: Sinauer Associates, 2004.
[19] K. Nixon, "The parsimony ratchet, a new method for rapid parsimony analysis," *Cladistics*, vol. 15, pp. 407–414, 1999.
[20] P. A. Goloboff, "Character optimization and calculation of tree lengths," *Cladistics*, vol. 9, pp. 433–436, 1993.
[21] D. Sankoff, Y. Abel, and J. Hein, "A tree · a window · a hill; generalization of nearest-neighbor interchange in phylogenetic optimization," *Journal of Classification*, vol. 11, no. 2, pp. 209–232, 1994.
[22] D. Gladstein, "Efficient Incremental Character Optimization," *Cladistics*, vol. 13, no. 1-2, pp. 21–26, 1997.
[23] F. Ronquist, "Fast Fitch-Parsimony Algorithms for Large Data Sets," *Cladistics*, vol. 14, no. 4, pp. 387–400, 1998.
[24] P. A. Goloboff, "Analyzing large datasets in reasonable times: Solutions for composite optima," *Cladistics*, vol. 15, pp. 415–428, 1999.
[25] M. Chase, D. Soltis, R. Olmstead, D. Morgan, D. Les, B. Mishler, M. Duvall, R. Price, H. Hills, Y. Qiu *et al.*, "Phylogenetics of Seed Plants: An Analysis of Nucleotide Sequences from the Plastid Gene rbcL," *Annals of the Missouri Botanical Garden*, vol. 80, no. 3, pp. 528–580, 1993.