EPoS: a modular software framework for phylogenetic analysis
Thasso Griebel*, Malte Brinkmeyer and Sebastian Böcker
Faculty of Mathematics and Computer Science, Friedrich-Schiller-University Jena, 07743 Jena, Germany
Received on April 11, 2008; revised on June 20, 2008; accepted on July 14, 2008
Advance Access publication July 16, 2008
Associate Editor: Martin Bishop

ABSTRACT
Summary: Estimating Phylogenies of Species (EPoS) is a modular software framework for phylogenetic analysis, visualization and data management. It provides a plugin-based system that integrates a storage facility, a rich user interface and the ability to easily incorporate new methods, functions and visualizations. EPoS ships with persistent data management, a set of well-known phylogenetic algorithms and a multitude of tree visualization methods and layouts. Implemented algorithms cover distance-based tree construction, consensus trees and various graph-based supertree methods. The rendering system can be customized for, say, different edge and node styles.
Availability: Executables and source code are available under the LGPL license at http://www.bio.informatik.uni-jena.de/epos.
Contact: thasso@minet.uni-jena.de
Supplementary information: The homepage contains tutorials and documentation for both users and programmers who want to develop plugins and extensions.

1 INTRODUCTION
Estimating Phylogenies of Species (EPoS) is a modular software framework for phylogenetic analysis that supports data management, computational methods and visualizations. There exists a wide variety of tools for phylogenetic analysis, but most tools show significant problems regarding usability, data handling and exchange. Algorithmic packages are often command line based and enforce a good understanding of the software environment. On the other hand, visualization tools usually suffer from poor or no support for computational methods. Most programs rest upon their own, unique file formats, which makes data exchange between the programs difficult. Even in a single phylogenetic analysis, a user is required to adopt to a multitude of different interfaces, and has to manually convert data formats.
EPoS fills this gap by combining a powerful graphical user interface (GUI) with a plugin system that allows simple integration of new algorithms, visualizations and data structures. It offers a simple way to incorporate new modules into the framework. In fact, the system itself is built from a set of core modules, which allows extensions in all directions. Limitations only concern the GUI and interaction model. The consistent EPoS GUI is used to manage and store all data and start available computational methods. Thus, the phylogenetic analysis workflow is uncoupled from data and applied methods. EPoS ensures that new computational methods never disrupt existing workflows. Visualizations, on the contrary, can be extended in any direction.

2 VISUALIZATIONS
EPoS contains views for trees, alignments and matrices. The alignment view allows for manual manipulations by modifying gaps, and the comprehensive tree view offers different layouts, colorizations, annotations and export functions. New views for various data types can easily be integrated into the framework, such as new tree layouts. The build-in tree view module focuses on interactive tree analysis and provides functionality to display large trees with several thousand leaves, without loosing the ability to smoothly interact with the view. When comparing two trees side-by-side, interacting with one tree can trigger actions in the second view, such as highlighting the best corresponding node (Munzner et al., 2003), see Section 4 below.

3 DATA MANAGEMENT
To simplify data handling, EPoS creates a persistent workspace that contains all data using a transparent and extendable back-end module. Changes in, say, the visualization of a tree (such as colors or layout) are persistently stored in a tree visualization object. EPoS uses an embedded database as default storage location, but there is no need for the user to manually interact with the database. Data can also be stored on a remote database server.
EPoS’ Application Programming Interface (API) allows data objects to carry private data and supplementary properties. For example, web services can be used to obtain additional information on an object without modifying the objects implementation. This feature can also be used by computational methods that need supplementary information besides tree structure: The Ranked Tree algorithm (Bryant et al., 2004) requires information about divergence dates in the input trees, see Section 4. Such data is simply added to the trees as a supplementary property.

4 METHODS
All computational methods are integrated into a pipeline system. This allows combinations of methods to be executed sequentially, where the data flow is handled automatically by the system. EPoS provides pipelines for different computational methods. It supports distance-based tree reconstruction methods including Neighbor Joining (Saitou and Nei, 1987) and Agglomerative Clustering, consensus construction such as Adams- and N-Consensus and several supertree methods that merge trees with overlapping leaf
sets. EPoS directly supports Aho’s Build (Aho et al., 1981), MinCut (Semple and Steel, 2000), modified MinCut (Page, 2002), Ranked Tree (Bryant et al., 2004) and Ancestral Build (Berry and Semple, 2006) as graph-based supertree algorithms. In addition, we implemented a tree comparison method based on the best corresponding node from (Munzner et al., 2003). This method matches each node from one tree to a corresponding node in the other tree. This is based on the comparison of leaf sets under the compared nodes. We extended the method to handle internal labels and propagate subtree scores upwards. No external software packages have to be installed to use any of these algorithms. New methods can be easily integrated into EPoS, as explained in the web tutorial.

The execution environment is another extendable part within the framework. In this way, EPoS is not limited to the local machine for executing pipelines. In the future, this will allow data and jobs to be moved to other machines or compute grids.

5 CONCLUSION

EPoS provides a scalable and extendable software framework for phylogenetic analysis. EPoS combines computational methods, data visualization tools and data management into one environment. The simplicity of the underlying plugin mechanism allows developers to easily integrate their own tools and algorithm into the framework, and to benefit from method provided by others. The process of connecting algorithms to data and data to visualization is completely covered by the system. Developers do not have to worry about persistency and data integrity. Users can access new computational methods without adapting to a new software environment.

Conflict of Interest: none declared.

REFERENCES


