BIOINFORMATICS APPLICATIONS NOTE Vol. 21 no. 5 2005, pages 676–679
doi:10.1093/bioinformatics/bti079

Phylogenetics

HyPhy: hypothesis testing using phylogenies

Sergei L. Kosakovsky Pond[1], Simon D. W. Frost[1] and Spencer V. Muse[2,*]

[1]Antiviral Research Center, University of California San Diego, San Diego, CA 92103, USA and
[2]Bioinformatics Research Center, North Carolina State University, Raleigh NC 27695-7566, USA

Received on May 12, 2004; revised on September 14, 2004; accepted on October 1, 2004
Advance Access publication October 27, 2004

ABSTRACT

Summary: The *HyPhy* package is designed to provide a flexible and unified platform for carrying out likelihood-based analyses on multiple alignments of molecular sequence data, with the emphasis on studies of rates and patterns of sequence evolution.

Availability: http://www.hyphy.org

Contact: muse@stat.ncsu.edu

Supplementary information: *HyPhy* documentation and tutorials are available at http://www.hyphy.org

1 INTRODUCTION

Research problems in molecular evolution, though wide-reaching in their goals, can be separated into two somewhat disjoint classes: studies of evolutionary history (phylogenetics), and studies of processes that govern molecular evolution. Of course, each of these two categories encompasses many different types of questions, and many investigations require studies of both phylogeny and evolutionary process. Software for molecular evolution is focused disproportionately on addressing issues of phylogenetic reconstruction, with a number of outstanding comprehensive packages to choose from. On the contrary, software for addressing questions about the evolutionary process tends to take the form of stand-alone programs that answer only one or two quite specific problems. There are a few exceptions, including PAML (Yang, 1997), Mr Bayes (http://morphbank.ebc.uu.se/mrbayes/info.php) and library-oriented projects such as PAL (Drummond and Strimmer, 2001) and PyEvolve (Butterfield *et al*., 2004). *HyPhy*was developed as a unified platform for designing, running and interpreting likelihood-based analyses of sequence alignments, with emphasis on modeling the evolutionary process and ease of use.

The *HyPhy*package serves a number of purposes and a number of audiences. It includes (1) a simple menu-based interface to many standard methods of molecular evolutionary analysis; (2) a high-level programming language (*HyPhy*Batch Language, HBL) that allows users to implement and distribute new methods of sequence analysis or to modify existing methods to satisfy the novelties of their data and analysis objectives; (3) a graphical user interface (GUI) that enables users to access much of the power of the programming language without the time investment of learning the language itself.

Unlike most other software packages for sequence analysis, *HyPhy*was designed to allow users to develop new methods of analysis quickly and easily, as opposed to simply offering a friendly interface to collections of existing methods. Also unlike most other packages, it was designed with multi-gene data sets in mind. Since the system was crafted in this style from the outset, *HyPhy*is uniquely positioned to address the needs of a wide variety of researchers as they seek to analyze comparative genomic data sets of ever-growing size and complexity.

2 METHODS AND DESIGN

2.1 Standard analyses

The following is a list of point-and-click prepackaged analyses currently included in *HyPhy*.

2.1.1 Model fitting Given a character alignment, a tree, and a model, obtain maximum likelihood parameter estimates, assess parameter estimation variability by asymptotic normality, profile likelihood, or bootstrap, and reconstruct ancestral character states using an efficient maximum likelihood joint reconstruction method.

2.1.2 Model selection An all-in-one implementation and extension of Model-Test (Posada and Crandall, 1998) for selection of nucleotide models; novel exhaustive schemes for selecting the best fitting nucleotide rate matrix and nucleotide bias corrections for codon substitution models.

2.1.3 Molecular clock Test for the presence of a molecular clock on all or some model parameters (e.g. only on synonymous rates). Clocks can be global or local (applied only to a subtree). Statistical *P*-values can be calculated via the bootstrap.

2.1.4 Phylogenetic reconstruction Distance matrix construction, cluster analysis, and neighbor joining, including information-based distance measures for unaligned sequences (Li *et al*., 2001). Maximum likelihood reconstruction (with or without a topological constraint) under any substitution model using exhaustive search, sequential addition, or star decomposition. NNI and SPR branch swapping are available.

2.1.5 Positive selection Various likelihood and approximate methods for detecting natural selection operating on alignment sites and lineages. Many of the methods are new and in the process of being published. The procedures allow for potentially complex patterns of substitution rate variation, including simultaneous variation of synonymous and non-synonymous substitution rates across sites. The methods can test for differential selection between populations and parts of the tree, and they can be run quickly on clusters of computers. A parallelized implementation of popular methods described in Yang *et al*. (2000) is also included. A public website implementing many of the methods can be accessed at http://www.datamonkey.org.

2.1.6 Relative rate tests A very general implementation of likelihood-based relative rate test methods (e.g. Muse and Weir, 1992). Any subset of the model parameters can be tested (for instance, only the synonymous or the non-synonymous substitution rates). Allows for automatic exhaustive comparisons of all pairs of taxa in an alignment and optional bootstrap *P*-values.

*To whom correspondence should be addressed.

© The Author 2004. Published by Oxford University Press. All rights reserved. For Permissions, please email: journals.permissions@oupjournals.org

Downloaded from https://academic.oup.com/bioinformatics/article/21/5/676/220389 by guest on 13 March 2024

2.1.7 Relative ratio tests An implementation of the relative ratio test methods (Muse and Gaut, 1997). Built-in tools for exhaustive comparisons of all pairs of taxa in a set of alignments and optional bootstrap P-values.

2.1.8 Miscellany KH tests (Hasegawa and Kishino, 1994); fitness amino acid models (Dimmic *et al*., 2000); sliding window analyses; estimation of site-by-site substitution rates similar to Olsen *et al*. (1994, http://geta.life.uiuc.edu/~gary/programs/DNArates.html).

2.1.9 Data tools Various tools for managing and manipulating sequence alignments.

The list of standard analyses is being continuously expanded, and advanced package users can customize existing analyses or write new ones.

2.2 Graphical user interface

The *HyPhy*GUI enables even casual users of *HyPhy*to design complex data analyses, run them, process the results, and create publication-quality graphics. The package includes a feature-rich data viewer and processor, a tree viewer and editor, a graphical model design component, a hypothesis design and testing module, a charting and numerical data analysis module, a built-in help system, and a full-featured text console. All modules support printing and data import and export. Most components provide easy means for the user to expand their functionality. Sample screen shots for some of the components are shown in Figure 1. The *HyPhy*documentation page (http://www.hyphy.org/docs) includes numerous examples and a tutorial on to how to use the GUI.

2.3 The 'hello world' of HBL

The HBL programming language allows for rapid implementation of new molecular evolutionary analyses, and is illustrated by the simple HBL program below. The following nine lines of code read an alignment of four sequences, define the F81 model of sequence evolution (Felsenstein, 1981), define a tree topology for the four taxa, find maximum likelihood estimates of all model parameters, and print the results:

```
DataSet myData = ReadDataFile ("../data/demo.seq");
             /* load an alignment from file*/
DataSetFilter myFilter = CreateFilter (myData,1);
             /* choose the sites to analyze (in
                this case, all of them) */
HarvestFrequencies (obsFreqs, myFilter, 1, 1, 1);
             /* compute character frequencies */
F81RateMatrix =  {{* ,mu,mu,mu}
                  {mu,* ,mu,mu}
                  {mu,mu,* ,mu}
                  {mu,mu,mu,* }};
Model F81 = (F81RateMatrix, obsFreqs);
             /* define the model of substitution */
Tree myTree = ((a,b),c,d);
             /* specify a phylogenetic tree */
LikelihoodFunction theLikFun = (myFilter, myTree);
Optimize (paramValues, theLikFun);
             /* define and maximize the
                likelihood function */
fprintf  (stdout, theLikFun);
             /* output the results */
```

This tutorial file and a complete explanation of the syntax are included with the distribution of *HyPhy*For more details, tutorials and examples related to HBL, please refer to http://www.hyphy.org/docs/

2.4 *HyPhy*core

The foundation of *HyPhy*consists of the following major modules.

2.4.1 Data reader and filter This module reads, writes and converts sequence alignments (nucleotide, amino acid, codon or custom alphabets with custom genetic codes) in a variety of file formats, including PHYLIP, NEXUS and FASTA. The file format and standard alphabets (nucleotide or amino acid) are detected automatically. The module allows the user to select an arbitrary subset of any data file for subsequent analysis and to perform merging operations on data sets, including matching alignment sites or sequences to regular expressions. The module provides a flexible mechanism for tabulating observed frequencies of characters or groups of characters (e.g. codons or dinucleotides) from a data set.

2.4.2 Expression parser This module includes ~100 functions and operations for processing and evaluating expressions of numbers, matrices and strings (e.g. $x := y(w + z)$). Examples include the cumulative distribution functions for common distributions, standard mathematical and logical operators, and many matrix operations. *HyPhy*also includes a growing function toolkit, including numerical integration of arbitrary expressions, root finding, maximization of user-defined functions and expression simplification. The *HyPhy*parser can handle numerical, string, matrix and associative array data types. The parser is used to impose constraints on model parameters, define rate matrices and process results. It is also heavily relied upon by the HBL interpreter and many other modules.

2.4.3 HBL interpreter This module converts scripts written in the *HyPhy*batch language into appropriate commands in the computational core. The language also provides flow control (conditional and looping statements and user-defined functions), user interaction (prompts), and window display options.

2.4.4 Tree parser The tree parser builds bifurcating or multifurcating phylogenetic trees from standard 'Newick' strings and assigns evolutionary models to tree branches. It has the ability to assign different models to different tree branches. *HyPhy*can test rooted and unrooted trees for equality or inclusion, match tree patterns (e.g. to check whether certain sequences are monophyletic), find common subtrees and forests among trees, and reroot trees.

2.4.5 Likelihood function module This module transparently constructs a wide range of likelihood functions, determines the number of parameters in a function, parses constraints on parameters, and performs appropriate non-linear optimization to obtain maximum likelihood parameter estimates, with applicable parallel enhancements. An efficient implementation of likelihood evaluations (Kosakovsky Pond and Muse, 2004) is included. *HyPhy*can perform fitting, inference and simulation on any character data, using any Markov model of character substitution. The modeling capabilities include (1) processes with multiple discrete rates variation across both sites and branches, with a large user-extensible collection of rate distributions, (2) flexible means to define mixtures of models, (3) hidden Markov models, (4) the ability to combine heterogeneous data, such as interspersed coding (codon) and non-coding (nucleotide) sequences, into a single analysis, (5) a rich collection of predefined models for nucleotide, codon and amino acid data.

3 IMPLEMENTATION

*HyPhy*is written in ANSI C++ and includes full-featured native GUIs for Mac OS (Carbon) and the Windows platform (Win32), and precompiled binaries are available. A command line version of the package can be built from the distributed source on any platform that is supported by the GCC family of compilers and includes POSIX compliant system libraries. *HyPhy*includes several open source libraries that are compiled into the binaries: GNU regular expressions library (http://ftp.gnu.org/pub/gnu/regex/regex-0.12.tar.gz), SQLite database engine (http://www.sqlite.org) and a Mersenne twister random number generator (http://www.math.keio.ac. jp/matumoto/emt.html).

Downloaded from https://academic.oup.com/bioinformatics/article/21/5/676/220389 by guest on 13 March 2024

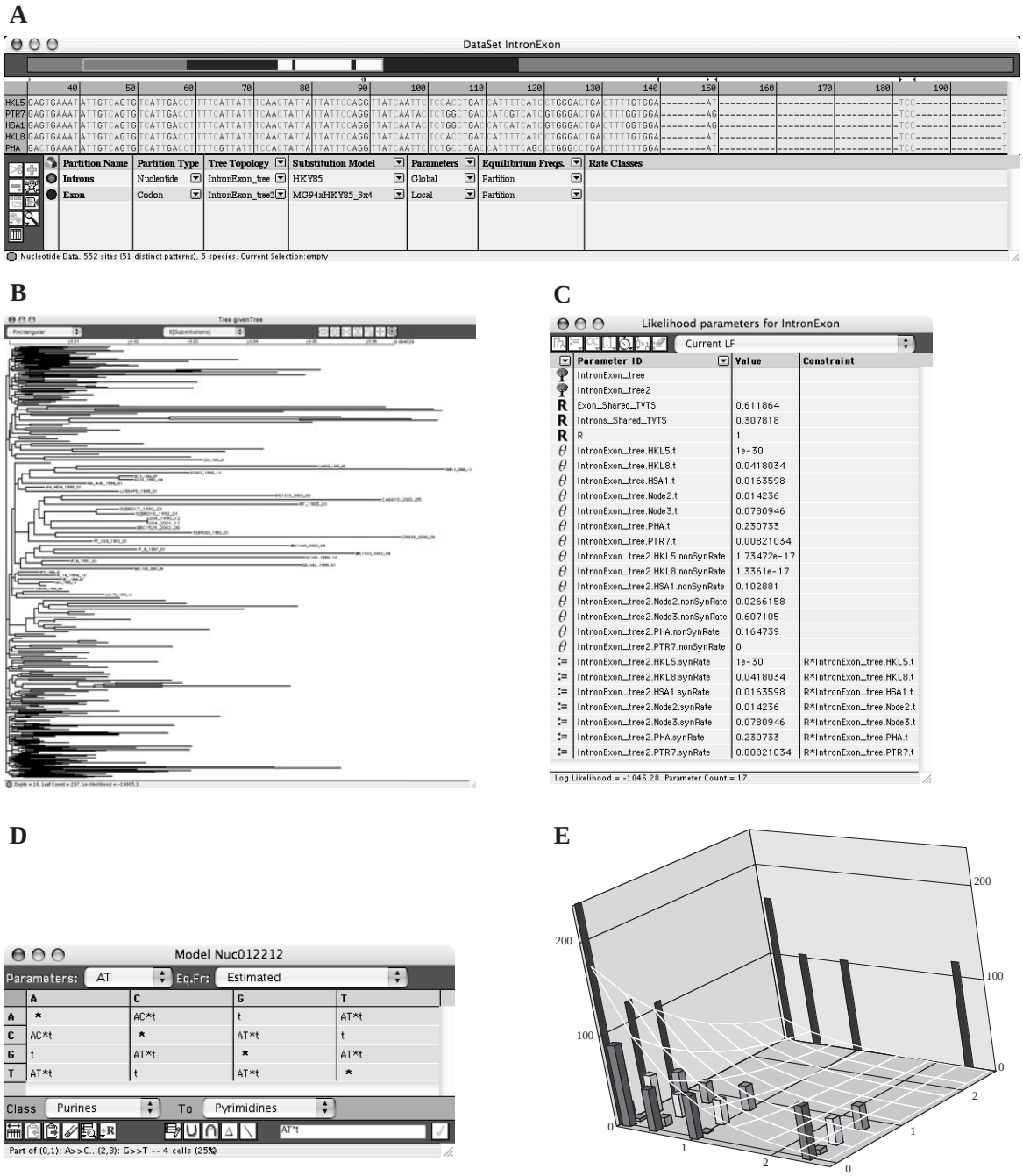

Fig. 1. Graphical components of *HyPhy*. (**A**) Data viewer used to set-up an analysis with mixed (intron-exon) data. (**B**) Tree viewer using a fish-eye projection to explore a large tree. (**C**) Model parameter viewer used to set up a relative ratio constraint between partitions. (**D**) Model editor used to define a custom nucleotide substitution model. (**E**) A 3D chart of inferred counts of sites in different synonymous and non-synonymous rate classes in a data set, compared with the expected continuous surface.

*HyPhy*transparently supports shared memory multiprocessing on systems that include a POSIX threads library by distributing likelihood function calculations across processors, and it can use MPI-distributed environments via the batch language. Many of the standard analyses are programmed to take advantage of MPI clusters when possible.

ACKNOWLEDGEMENTS

This work was supported in part by grants from the NSF (DBI-0096033 and DEB-9996118 to SVM), NIH (5 R01 AI47745 and 5 U01 AI43638 Supp), the University of California Universitywide AIDS Research Program (IS02-SD-701), and by a University of

Downloaded from https://academic.oup.com/bioinformatics/article/21/5/676/220389 by guest on 13 March 2024

California, San Diego Center for AIDS Research/NIAID Developmental Award to S.D.W.F. (grant no. 2 P30 AI36214). We thank two anonymous reviewers for useful comments on earlier versions of this manuscript.

REFERENCES

Butterfield,A., Vedagiri,V., Lang,E., Lawrence,C., Wakefield,M., Isaev,A. and Huttley,G. (2004) Pyevolve: a toolkit for statistical modelling of molecular evolution. BMC BIOINFORMATICS, **5**, 1.

Dimmic,M., Mindell,D. and Goldstein,R. (2000) Modeling evolution at the protein level using an adjustable amino acid fitness model. In *Pacific Symposium on Biocomputing*, Honolulu.

Drummond,A. and Strimmer,K. (2001) PAL: an object-oriented programming library for molecular evolution and phylogenetics. *Bioinformatics*, **17**, 662–663.

Felsenstein,J. (1981) Evolutionary trees from DNA-sequences—a maximum-likelihood approach. *J. Mol. Evol.*, **17**, 368–376.

Hasegawa,M. and Kishino,H. (1994) Accuracies of the simple methods for estimating the bootstrap probability of a maximum-likelihood tree. *Mol. Biol. Evol.*, **11**, 142–145.

Kosakovsky Pond,S. and Muse,S. (2004) Column sorting: rapid calculation of the phylogenetic likelihood function. *Systematic Biology* **53**, 685–692.

Li,M., Badger,J., Chen,X., Kwong,S., Kearny,P. and Zhang,H. (2001) An information-based sequence distance and its application to whole mitochondrial genome phylogeny. *Bioinformatics*, **17**, 149–154.

Muse,S.V. and Gaut,B.S. (1997) Comparing patterns of nucleotide substitution rates among chloroplast loci using the relative ratio test. *Genetics*, **146**, 393–399.

Muse,S.V. and Weir,B.S. (1992) Testing for equality of evolutionary rates. *Genetics*, **132**, 269–276.

Olsen,G.J., Pracht,S. and Overbeek,R. (1994) *DNArates*.

Posada,D. and Crandall,K. (1998) Modeltest: testing the model of DNA substitution. *Bioinformatics*, **14**, 817–818.

Yang,Z.H. (1997) PAML: a program package for phylogenetic analysis by maximum likelihood. *Comput. Appl. Biosci.*, **13**, 555–556.

Yang,Z.H., Nielsen,R., Goldman,N. and Pedersen,A.M.K. (2000) Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics*, **155**, 431–449.

Downloaded from https://academic.oup.com/bioinformatics/article/21/5/676/220389 by guest on 13 March 2024