Bio++: Efficient Extensible Libraries and Tools for Computational Molecular Evolution

Laurent Gueguen, Sylvain Gaillard, Bastien Boussau, Manolo Gouy, Mathieu Groussin, Nicolas C. Rochette, Thomas Bigot, David Fournier, Fanny Pouyet, Vincent Cahais, et al.

Bio++: Efficient Extensible Libraries and Tools for Computational Molecular Evolution

Laurent Guéguen,[1] Sylvain Gaillard,[2,3,4] Bastien Boussau,[1,5] Manolo Gouy,[1] Mathieu Groussin,[1] Nicolas C. Rochette,[1] Thomas Bigot,[1] David Fournier,[6] Fanny Pouyet,[1] Vincent Cahais,[7] Aurélien Bernard,[7] Céline Scornavacca,[7] Benoît Nabholz,[7] Annabelle Haudry,[1] Loïc Dachary,[8] Nicolas Galtier,[7] Khalid Belkhir,[7] and Julien Y. Dutheil*[,7,9]

[1]Laboratoire de Biométrie et Biologie Evolutive, Université de Lyon, CNRS, INRIA, Villeurbanne, France

[2]INRA, Institut de Recherche en Horticulture et Semences, Angers, France

[3]Agrocampus Ouest, Institut de Recherche en Horticulture et Semences, Angers, France

[4]Institut de Recherche en Horticulture et Semences, Université d'Angers, LUNAM Université, Angers, France

[5]Department of Integrative Biology, University of California, Berkeley

[6]Computational Biology and Data Mining Group, Max-Delbrueck-Center for Molecular Medicine, Berlin, Germany

[7]Institut des Sciences de l'Evolution, Université Montpellier 2, Montpellier, France

[8]12 bd Magenta, Paris, France

[9]Department of Organismic Interactions, Max Planck Institute for Terrestrial Microbiology, Marburg, Germany

***Corresponding author:** E-mail: julien.dutheil@univ-montp2.fr.

Associate editor: Sudhir Kumar

Abstract

Efficient algorithms and programs for the analysis of the ever-growing amount of biological sequence data are strongly needed in the genomics era. The pace at which new data and methodologies are generated calls for the use of pre-existing, optimized—yet extensible—code, typically distributed as libraries or packages. This motivated the Bio++ project, aiming at developing a set of C++ libraries for sequence analysis, phylogenetics, population genetics, and molecular evolution. The main attractiveness of Bio++ is the extensibility and reusability of its components through its object-oriented design, without compromising the computer-efficiency of the underlying methods. We present here the second major release of the libraries, which provides an extended set of classes and methods. These extensions notably provide built-in access to sequence databases and new data structures for handling and manipulating sequences from the omics era, such as multiple genome alignments and sequencing reads libraries. More complex models of sequence evolution, such as mixture models and generic *n*-tuples alphabets, are also included.

Key words: **bioinformatics, models of sequence evolution, phylogeny, C++ libraries.**

The field of molecular evolution has always relied heavily on the use of computers for modeling and analysis (Eck and Dayhoff 1966; Fitch and Margoliash 1967). The need to use computers, and to use them efficiently, is even more pressing now that genome sequence data are accumulating at an increasing pace. In 2006, version 1.0.0 of the Bio++ libraries was published (Dutheil et al. 2006) with the aim to provide a set of flexible, efficient, object-oriented C++ methods for sequence analysis, population genetics, and molecular phylogenetics. Bio++ offers a set of ready-to-use bricks to construct sequence analysis pipelines, develop new complex probabilistic models, run maximum likelihood inference or simulate data, among other possibilities. Since their initial release the libraries have been used in a variety of published works and have enabled the development of new models and tools (for recent examples, see Bérard and Guéguen 2012; Caffrey et al. 2012; Dutheil et al. 2012; Szöllosi et al. 2012; Boussau et al. 2013; Groussin et al. 2013; Scornavacca et al. 2013). As they have been attracting new users and developers, the libraries have been extended to include new analysis tools, and now contain the largest set of models for sequence evolution ever implemented.

New Developments

The initial release of the Bio++ libraries (Dutheil et al. 2006) was followed by several regular updates, and a major new version (Bio++ 2.0.0) was released in 2011. As of January 2013, the current stable version is 2.1.0. Since version 1.0.0, the libraries have extensively developed and new libraries were added to the initial set. These libraries provide new functionalities, mainly dedicated to database access, graphics and graphical user interfaces (GUIs), as well as genomic analysis. The original libraries have also been extended to incorporate new models and analytical tools.

Architecture of the Libraries

Since version 1.0.0, the amount of code in the libraries has more than doubled, reaching a total of more than 700 classes.

Letter

Fast Track

For ease of use, the code is split into several libraries, which can be installed and linked independently, depending on the user's specific needs. Version 2.1.0 contains eight libraries. The "bpp-core" library contains basal classes and interfaces necessary for the development of applications with Bio++. Three other libraries inherited from version 1.0.0 gather tools for sequence analysis (bpp-seq), phylogenetics (bpp-phyl), and population genetics (bpp-popgen). Finally, the following four new libraries were developed:

- bpp-raa, for Remote Acnuc Access, providing classes to query sequence databases
- bpp-seq-omics and bpp-phyl-omics, providing classes for (phylo)genomic analyses
- bpp-qt, providing graphical components based on the Qt library (Blanchette and Summerfield 2008).

Figure 1 shows the dependencies between these libraries. We now briefly describe the recent developments of the original Bio++ components, and the content of the new ones.

Numerical Tools

The models available in Bio++ require numerical routines, which are coded in the core library. Since Bio++ 1.0.0, the collection of available algorithms has been extended (e.g., we added support for numerical derivatives, function reparametrization, and sampling procedures), and the efficiency of existing methods has been further improved. The library provides a fully object-oriented implementation of commonly used routines and algorithms for function minimization and derivation, or matrix calculus. In particular, the library offers a large set of object-oriented, event-driven minimization algorithms for finely tuned optimization of complex functions with numerous parameters, such as likelihood under phylogenetic models. Developing new probabilistic models is now made easier thanks to a larger array of continuous or discretized distributions (Gaussian, exponential, beta, Dirichlet, and any mixture of distributions), as well as standard algorithms for hidden Markov modeling (forward, backward algorithms, and posterior decoding with rescaling to avoid numerical underflow [Durbin et al. 1998]).

Database Access

The bpp-raa library allows network access to several nucleotide and protein sequence databases, both generalist ones (the EMBL sequence library, GenBank, and UniProt) and databases of families of homologous protein-coding genes (e.g., HOGENOM, HOMOLENS, HOVERGEN; Penel et al. 2009). Bpp-raa employs the ACNUC sequence retrieval system (Gouy and Delmotte 2008) to communicate between the library user and a sequence database. The `bpp::RAA` class opens a network connection to a given database, and allows extracting sequences and annotations based on sequence name or accession number, with optional translation to protein. This class also allows building the list of sequences that match a given query, including complex queries that involve logical combinations of criteria (e.g., species name AND/OR/NOT keyword AND/OR/NOT reference AND/OR/NOT previous query). The members of a sequence list can then be extracted for local processing. The `bpp::RaaSpeciesTree` class allows using the taxonomy associated with sequence databases, walking up and down this tree, and finding its nodes by name or numerical taxon ID.

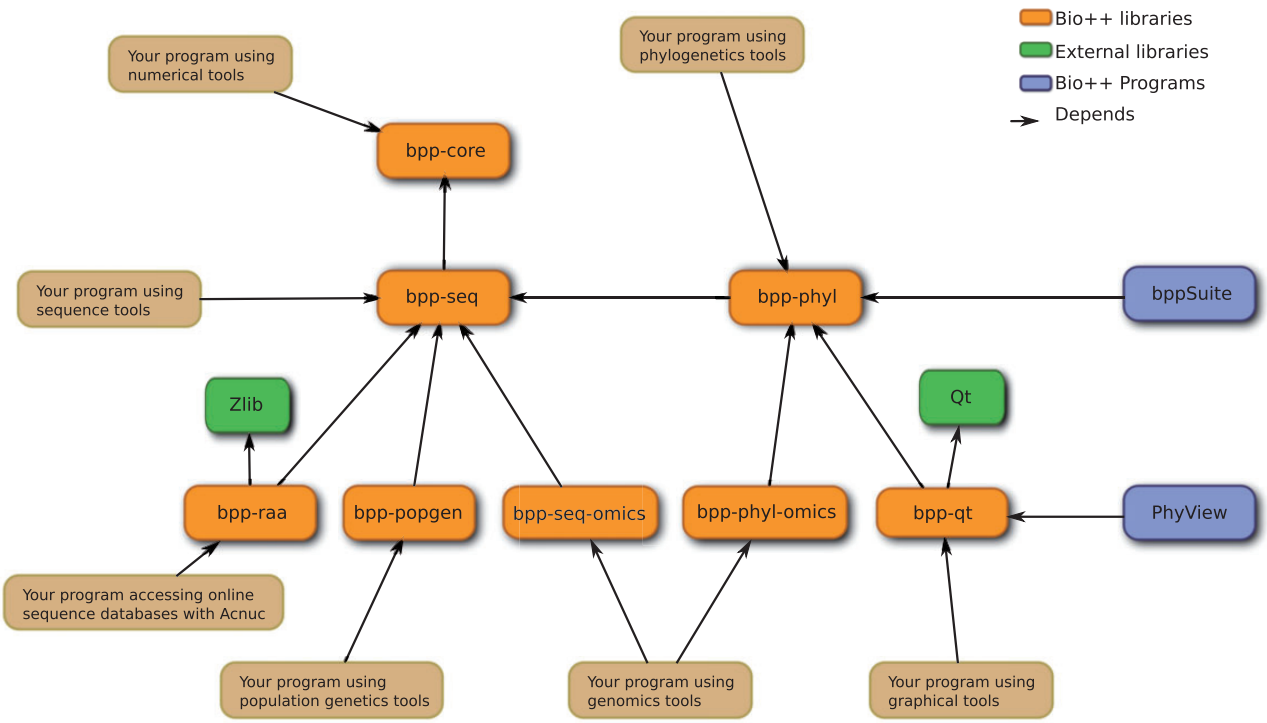

FIG. 1. Dependencies between libraries and programs.

Genomic Tools

The sequence class hierarchy has been extended to cope with the increasing amount of genomic data. These developments follow three main axes: 1) faster handling of sequences, notably via the use of binary coding to allow more efficient comparisons, and rewriting of file parsers, 2) support for subsequences and features, including parsers for GFF and GTF formats, as well as storage and manipulation of meta-data like quality scores, and 3) addition of new file formats, notably those used for (Next Generation) sequencing (Phred, FastQ, and MAF). These new data structures enable a very efficient parsing and filtering of typical genomic data sets. A simple program using a `bpp::SequenceIterator` based on the new `bpp::FastQ` parser and the `bpp::SequenceWithQuality` data structure is able to parse 20 millions paired-ends reads of 100 bp in 20 min on a desktop computer, whereas the same analysis requires more than 1 h and 30 min with an equivalent pipeline built using the (locally installed) Galaxy platform (Hillman-Jackson et al. 2012).

New Models of Sequence Evolution

The first version of Bio++ already supported a large variety of models of sequence evolution for nucleotide or amino acid sequences, later extended with branch-heterogeneous models (Dutheil and Boussau 2008; Groussin et al. 2013). Version 2.1.0 offers in addition a generalized modeling framework that considers *n*-tuples as the evolving units in a sequence. This permits an extensible and flexible implementation of codon models. Specific features of codon models are implemented separately in abstract classes, enabling the development of customized codon models. The currently implemented models notably support 1) position-specific substitution rates; 2) biochemical distances between the encoded amino acids (as in the GY94 model; Goldman and Yang 1994); and 3) preferences between synonymous codons (Yang and Nielsen 2008). Equilibrium frequencies are modeled either in a position-specific manner or at the codon level, with possibility for the user to provide his/her own implementation. The substitution rate is proportional either to the target codon equilibrium frequencies (as in GY94) or to the target nucleotide equilibrium frequencies (as in the MG94 model; Muse and Gaut 1994). This generic implementation unifies the vast majority of models proposed in the literature (Pond and Muse 2005; Wong et al. 2006; Mayrose et al. 2007).

Bio++ 2.0.0 also provides support for mixed models. In these models, a site can "choose" between several models (Yang and Wang 1995; Yang et al. 2000). The resulting compound likelihood for a site is the average of the conditional likelihoods for each model, weighted by their probability distribution. Using this new generic framework, several previously published mixed models have been made available in Bio++, such as codon models M1, M2, M3, M7, and M8 from the widely used codeml program (Yang et al. 2000; Yang 2007) for modeling site-specific selection coefficients or the protein models UL2, UL3, EX2, CAT-C10 to C60 among others (Le, Lartillot, et al. 2008; Le, Gascuel, et al. 2008) for modeling site-specific properties of proteins.

A generic framework has also been implemented for combining branch-heterogeneous models with mixed models. In this framework, it is possible to assign mixed models to a subset of branches. Different mixed models can be assigned to separate branches, in which case a site is allowed to switch between categories of models at nodes, as in the branch-site model of PAML (Zhang et al. 2005) (fig. 2 and supplementary fig. S1, Supplementary Material online). In addition, it is possible to constrain those switches so that particular sets of branches are always in the same category. The current implementation therefore covers a large set of mixed models available in the literature, whilst enabling the development of new ones.

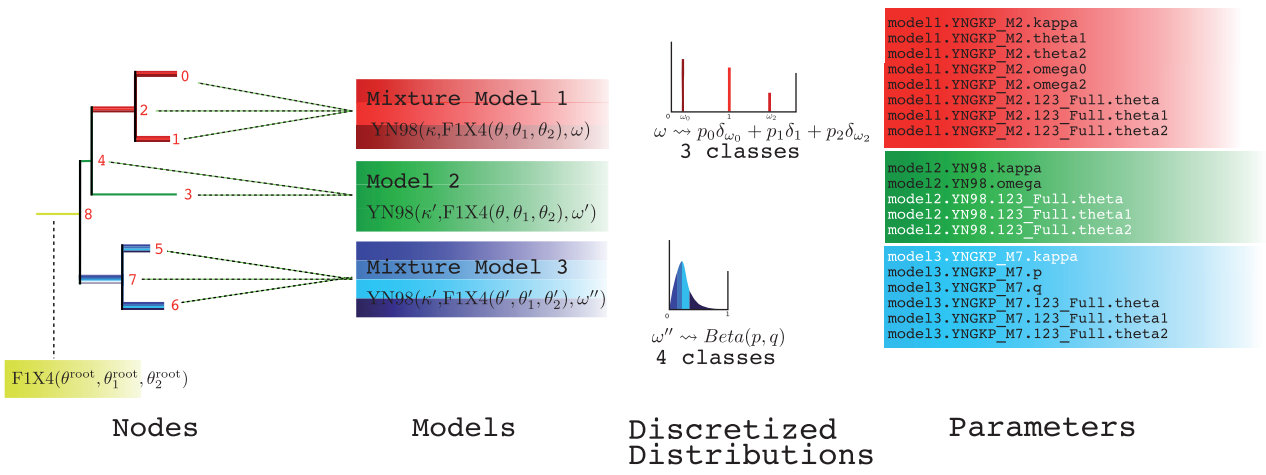

FIG. 2. Non-homogeneous modeling with mixture models. Example of nonstationary and nonhomogeneous modeling of evolution of a codon sequence, using three models (M0, M2, and M7) as defined in Nielsen and Yang (1998) and Yang et al. (2000). On branches 0, 1, and 2, a site can choose between three YN98 models, in which omega can be <1, =1, or >1, with specific probabilities. On branches 5, 6, and 7, a site can choose between four YN98 models, in which omega follows a discretized beta distribution. The equilibrium frequencies of the model on branches 3 and 4 are the same as the ones of the model on branches 0, 1, and 2. The kappa parameter value is the same on branches 3, 4, 5, 6, and 7. In the parameter list, parameters in white are shared between models. Although artificial, this example demonstrates the generality of the modeling framework implemented in Bio++.

An Extended Set of Tools for Molecular Evolution

The large set of models of sequence evolution available in Bio++ can be used in combination with routinely used methods in evolutionary bioinformatics, such as tree-building, population analyses, sequence simulation, and ancestral state reconstruction. Although the libraries are not dedicated to phylogenetic reconstruction per se (for which specialized software exist), they contain building blocks based on published algorithms which can be useful to develop new methods in that field. Such "blocks" include parsimony score and tree likelihood computation (with simple and double recursive algorithms, see Felsenstein 2003), as well as nearest-neighbor interchange topology movements. Distance methods are also available, including neighbor joining (Saitou and Nei 1987) and BioNJ (Gascuel 1997), which are implemented in an object-oriented way. The majority of models implemented can also be used to simulate sequences, including covarion models and nonhomogeneous models, and to reconstruct ancestral sequences using the empirical Bayesian approach (Yang et al. 1995). Population genetics statistics include the computation of a variety of sequence diversity estimators, Tajima's D (Tajima 1989), neutrality index (Rand and Kann 1996) and McDonald and Kreitman's count table for testing of positive selection (McDonald and Kreitman 1991). Since version 1.0.0, a notable addition is the development of generic substitution mapping procedures (Minin and Suchard 2008; Tataru and Hobolth 2011), which can be used to characterize patterns of substitution in a robust and efficient manner (Lemey et al. 2012; Romiguier et al. 2012).

Graphical Tools

Graphical tools have been introduced in version 2.0.0 of the libraries. The bpp-core library provides a generic `bpp::GraphicDevice` class supporting drawing operations such as lines, polygons, and text writing, as well as dedicated interfaces to handle colors and fonts. The bpp-core library includes three implementations of this interface: Scalable Vector Format, LaTeX's Portable graphic Format and the Xfig format, and the bpp-qt library provides an additional implementation based on the Qt graphic library. The bpp-phyl library contains several algorithms for plotting trees on a `bpp::GraphicDevice`, which can therefore be used to save a graphical representation of a tree into a file, or as part of a GUI. Pre-built GUI components for phylogenetic tree browsing are included in the bpp-qt library, and used in the bppPhyView software, a powerful Bio++ based tree editor.

The Bio++ Program Suite and the BppO Language

Several programs developed using the Bio++ libraries are distributed as the Bio++ Program Suite (bppSuite), including the following:

- bppML, which performs maximum likelihood estimation of models of sequence evolution,
- bppSeq, which simulates sequences under a model of sequence evolution,
- bppAncestor, which reconstructs ancestral sequences,
- bppDist, which reconstructs phylogenies based on distance matrices.

They all share a common language for the description of their parameters, notably models of sequence evolution. In Bio++ 2.1.0, this language has a dedicated Application Programming Interface (API) included in the library. It is referred to as the Bio++ Options language, or simply BppO. With BppO, one can easily specify which of the input/output formats, models, frequencies, discrete distributions, to use and perform—depending on the chosen bppSuite program—maximum likelihood estimation of parameters, ancestral sequence reconstruction, sequence simulation, and so forth. Two examples showing how complex models can be specified using the BppO syntax are given in figures 2 and 3 (for codon models) and supplementary figures S1 and S2, Supplementary Material online (for nucleotide models). Programs in bppSuite output their results in a BppO file, which can then be used directly as input for another program. This makes it easy for instance to use a previously fitted model to simulate sequences or reconstruct ancestral sequences. Through the BppO language and BppSuite, a large set of the features of Bio++ are made available to the user without the need for C++ programming.

Availability and Future Directions

The Bio++ libraries are distributed under the CeCILL 2.0 license (compatible with the GNU Public License) at http://bioweb.me/biopp (last accessed June 6, 2013). Source code can be compiled (at least) on any system where the GNU compiler collection is available (including Linux, MacOS, and Windows). Bio++ uses CMake for its configuration (Martin and Hoffman 2010), which facilitates its integration with widely used development environments such as Visual Studio, XCode, CodeBlocks, or Eclipse. Stable versions are released yearly, with precompiled and source packages available for the most common Linux distributions and MacOS. Since 2011, the Bio++ libraries and packages are also directly available from the Debian distribution (and therefore its derivatives such as Ubuntu and Linux Mint). The latest development version of the code can be obtained from a central Git repository.

Bio++ uses unit tests and is checked nightly. The API documentation, generated using the Doxygen program (http://www.doxygen.org, last accessed June 6, 2013), is also updated nightly and made available online to ease the development of new applications. In addition, the Bio++ website features a wiki-based documentation, example programs, a bug tracker and two forums (biopp-help dedicated to getting help with the use of the libraries, with more than 70 members, 160 topics, 890 posts, and biopp-devel for general development discussion, with more than 30 members, 260 topics, 940 posts).

Thanks to its growing community, Bio++ is under continuous development. The strength of Bio++ is its combination of generality and efficiency. Generality is achieved through the

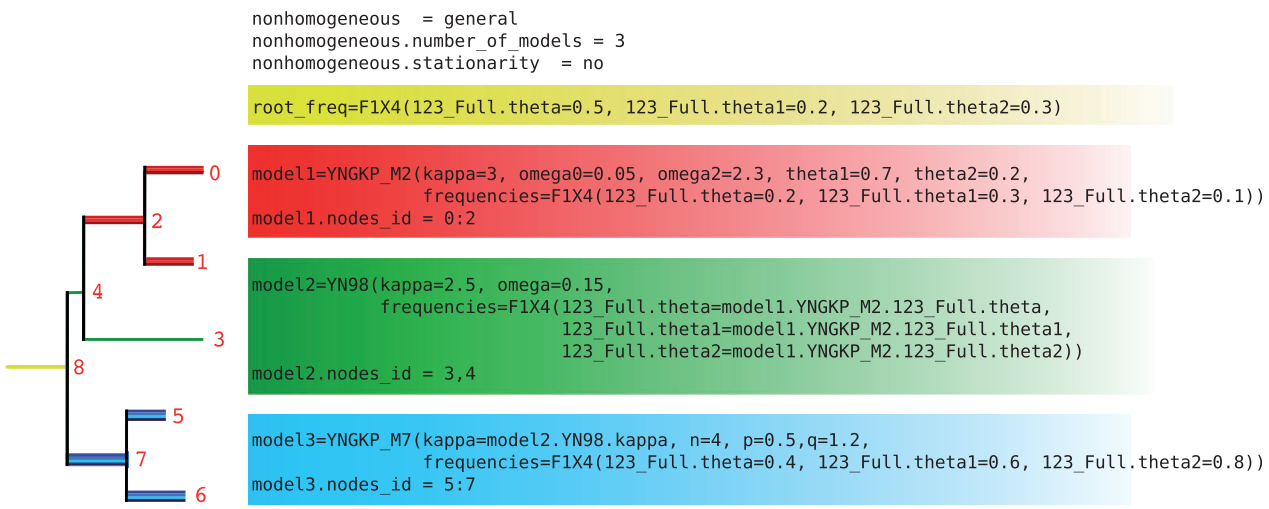

Fig. 3. Syntax of the modeling in the BppO language, using the specific names of models described in the bibliography.

strictly object-oriented design of the library, which eases the development of new models of sequence evolution.

Comparison with other pieces of software shows that the versatility of Bio++ comes at a minimal cost in terms of computer resources (supplementary table S1, Supplementary Material online). For instance, on a nucleotide data set of 79 sequences and 2,353 sites, the BppML program (from the Bio++ program suite) fits a GTR substitution model with 4 gamma-distributed rate classes in 2′05 minutes on a linux desktop machine, using 215 kB of memory. PhyML achieves the same analysis in 1 minute 18 seconds with 390 kB. PAML uses only 28 kB but performs the estimation in 6 minutes 54 seconds. All three programs return the same parameter estimates and likelihood. This efficiency is due to a fine control of memory usage achieved through the classes and tools of the C++ Standard Template Library, as well as the efficient function optimizers implemented in bpp-core. For phylogenetic models, a dedicated modified Newton–Raphson algorithm is used, based on an initial idea from Felsenstein's phylip package, further improved in the NHML software, and re-implemented in an object-oriented manner in Bio++. Programs developed with Bio++ are therefore well fitted for data analyses typically achieved by their C-coded, non-library-based counterparts. Further improving the performances of the libraries is one of the next challenges that the Bio++ developers are currently pursuing, notably by pushing the limit of numerical underflow and developing support for parallelization, to handle increasingly large data sets.

Supplementary Material

Supplementary figures S1 and S2 and table S1 are available at *Molecular Biology and Evolution* online (http://www.mbe.oxfordjournals.org/).

Acknowledgments

The authors thank the users of the Bio++ forums for their continuous feedback helping them to improve the code of the library, as well as the reviewers for their constructive comments on an earlier version of this manuscript. This work has been partially funded by the Agence Nationale de la Recherche ANCESTROME project (ANR-10-BINF-01-01 and ANR-10-BINF-01-02) and the European Research Council PopPhyl project (ERC 232971). This publication is the contribution no. 2013-059 of the Institut des Sciences de l'Evolution de Montpellier (ISE-M).

References

Bérard J, Guéguen L. 2012. Accurate estimation of substitution rates with neighbor-dependent models in a phylogenetic context. *Syst Biol.* 61:510–521.

Blanchette J, Summerfield M. 2008. C++ GUI programming with Qt 4, 2nd ed. Upper Saddle River (NJ): Prentice Hall.

Boussau B, Szöllosi GJ, Duret L, Gouy M, Tannier E, Daubin V. 2013. Genome-scale coestimation of species and gene trees. *Genome Res.* 23:323–330.

Caffrey BE, Williams TA, Jiang X, Toft C, Hokamp K, Fares MA. 2012. Proteome-wide analysis of functional divergence in bacteria: exploring a host of ecological adaptations. *PLoS One* 7:e35659.

Durbin R, Eddy SR, Krogh A, Mitchison G. 1998. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge (UK): Cambridge University Press.

Dutheil J, Boussau B. 2008. Non-homogeneous models of sequence evolution in the Bio++ suite of libraries and programs. *BMC Evol Biol.* 8:255.

Dutheil J, Gaillard S, Bazin E, Glémin S, Ranwez V, Galtier N, Belkhir K. 2006. Bio++: a set of C++ libraries for sequence analysis, phylogenetics, molecular evolution and population genetics. *BMC Bioinformatics* 7:188.

Dutheil JY, Galtier N, Romiguier J, Douzery EJP, Ranwez V, Boussau B. 2012. Efficient selection of branch-specific models of sequence evolution. *Mol Biol Evol.* 29:1861–1874.

Eck RV, Dayhoff MO. 1966. Evolution of the structure of ferredoxin based on living relics of primitive amino acid sequences. *Science* 152:363–366.

Felsenstein J. 2003. Inferring phylogenies, 2nd ed. Sunderland (MA): Sinauer Associates.

Fitch WM, Margoliash E. 1967. Construction of phylogenetic trees. *Science* 155:279–284.

Gascuel O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol.* 14: 685–695.

Goldman N, Yang Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Mol Biol Evol.* 11: 725–736.

Gouy M, Delmotte S. 2008. Remote access to ACNUC nucleotide and protein sequence databases at PBIL. *Biochimie* 90:555–562.

Groussin M, Boussau B, Gouy M. Forthcoming 2013. A branch-heterogeneous model of protein evolution for efficient inference of ancestral sequences. *Syst Biol.*

Hillman-Jackson J, Clements D, Blankenberg D, Taylor J, Nekrutenko A. 2012. Using galaxy to perform large-scale interactive data analyses. *Curr Protoc Bioinformatics.* Chapter 10:Unit10.5.

Le SQ, Gascuel O, Lartillot N. 2008. Empirical profile mixture models for phylogenetic reconstruction. *Bioinformatics* 24:2317–2323.

Le SQ, Lartillot N, Gascuel O. 2008. Phylogenetic mixture models for proteins. *Philos Trans R Soc Lond B Biol Sci.* 363:3965–3976.

Lemey P, Minin VN, Bielejec F, Kosakovsky Pond SL, Suchard MA. 2012. A counting renaissance: combining stochastic mapping and empirical Bayes to quickly detect amino acid sites under positive selection. *Bioinformatics* 28:3248–3256.

Martin K, Hoffman B. 2010. Mastering CMake: a cross-platform build system Version 5. Villeurbanne (France): Kitware.

Mayrose I, Doron-Faigenboim A, Bacharach E, Pupko T. 2007. Towards realistic codon models: among site variability and dependency of synonymous and non-synonymous rates. *Bioinformatics* 23: i319–i327.

McDonald JH, Kreitman M. 1991. Adaptive protein evolution at the Adh locus in *Drosophila*. *Nature* 351:652–654.

Minin VN, Suchard MA. 2008. Fast, accurate and simulation-free stochastic mapping. *Philos Trans R Soc Lond B Biol Sci.* 363: 3985–3995.

Muse SV, Gaut BS. 1994. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Mol Biol Evol.* 11: 715–724.

Nielsen R, Yang Z. 1998. Likelihood models for detecting positively selected amino acid sites and applications to the HIV-1 envelope gene. *Genetics* 148:929–936.

Penel S, Arigon A-M, Dufayard J-F, Sertier A-S, Daubin V, Duret L, Gouy M, Perrière G. 2009. Databases of homologous gene families for comparative genomics. *BMC Bioinformatics* 10(6 Suppl):S3.

Pond SK, Muse SV. 2005. Site-to-site variation of synonymous substitution rates. *Mol Biol Evol.* 22:2375–2385.

Rand DM, Kann LM. 1996. Excess amino acid polymorphism in mitochondrial DNA: contrasts among genes from *Drosophila*, mice, and humans. *Mol Biol Evol.* 13:735–748.

Romiguier J, Figuet E, Galtier N, Douzery EJP, Boussau B, Dutheil JY, Ranwez V. 2012. Fast and robust characterization of time-heterogeneous sequence evolutionary processes using substitution mapping. *PLoS One* 7:e33852.

Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol.* 4:406–425.

Scornavacca C, Paprotny W, Berry V, Ranwez V. 2013. Representing a set of reconciliations in a compact way. *J Bioinform Comput Biol.* 11:1250025.

Szöllosi GJ, Boussau B, Abby SS, Tannier E, Daubin V. 2012. Phylogenetic modeling of lateral gene transfer reconstructs the pattern and relative timing of speciations. *Proc Natl Acad Sci U S A.* 109: 17513–17518.

Tajima F. 1989. Statistical method for testing the neutral mutation hypothesis by DNA polymorphism. *Genetics* 123:585–595.

Tataru P, Hobolth A. 2011. Comparison of methods for calculating conditional expectations of sufficient statistics for continuous time Markov chains. *BMC Bioinformatics* 12:465.

Wong WSW, Sainudiin R, Nielsen R. 2006. Identification of physico-chemical selective pressure on protein encoding nucleotide sequences. *BMC Bioinformatics* 7:148.

Yang Z. 2007. PAML 4: phylogenetic analysis by maximum likelihood. *Mol Biol Evol.* 24:1586–1591.

Yang Z, Kumar S, Nei M. 1995. A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics* 141:1641–1650.

Yang Z, Nielsen R. 2008. Mutation-selection models of codon substitution and their use to estimate selective strengths on codon usage. *Mol Biol Evol.* 25:568–579.

Yang Z, Nielsen R, Goldman N, Pedersen AM. 2000. Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics* 155:431–449.

Yang Z, Wang T. 1995. Mixed model analysis of DNA sequence evolution. *Biometrics* 51:552–561.

Zhang J, Nielsen R, Yang Z. 2005. Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Mol Biol Evol.* 22:2472–2479.