LETTERS

jModelTest: Phylogenetic Model Averaging

David Posada
Departamento de Genética, Bioquímica e Inmunología, Facultad de Biología, Universidad de Vigo, Vigo, Spain

jModelTest is a new program for the statistical selection of models of nucleotide substitution based on “Phyml” (Guindon and Gascuel 2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. Syst Biol. 52:696–704.). It implements 5 different selection strategies, including “hierarchical and dynamical likelihood ratio tests,” the “Akaike information criterion,” the “Bayesian information criterion,” and a “decision-theoretic performance-based” approach. This program also calculates the relative importance and model-averaged estimates of substitution parameters, including a model-averaged estimate of the phylogeny. jModelTest is written in Java and runs under Mac OSX, Windows, and Unix systems with a Java Runtime Environment installed. The program, including documentation, can be freely downloaded from the software section at http://darwin.uvigo.es.

Introduction

Models of nucleotide substitution allow for the calculation of probabilities of change between nucleotides along the branches of a phylogenetic tree. The use of a particular substitution model may change the outcome of the phylogenetic analysis (e.g., Buckley 2002; Buckley and Cunningham 2002; Lemmon and Moriarty 2004), and statistical model selection has become an essential step for the estimation of phylogenies from DNA sequence alignments. In-depth reviews about model selection in phylogenetics are available elsewhere (Johnson and Omland 2003; Posada and Buckley 2004; Sullivan and Joyce 2005). Indeed, the performance of different model selection strategies has been the subject of active research (Posada 2001; Posada and Crandall 2001; Pol 2004; Abdo et al. 2005; Alfaro and Huelsenbeck 2006).

Several programs already exist for the statistical selection of models of nucleotide substitution (e.g., Nylander 2004; Keane et al. 2006). Among these, Modeltest (Posada and Crandall 1998) has been one of the most popular. This note describes a new program called jModelTest that supersedes Modeltest in several aspects. jModelTest allows for the definition of restricted sets of candidate models (table 1), implements customizable “hierarchical likelihood ratio tests” (hLRTs) (Frati et al. 1997; Huelsenbeck and Crandall 1997; Sullivan et al. 1997) and “dynamic likelihood ratio tests” (dLRTs) (Posada and Crandall 2001), provides a rank of models according to the “Akaike Information Criterion” (AIC) (Akaike 1973), to the “Bayesian Information Criterion” (BIC) (Schwarz 1978) or to a “decision-theoretic performance-based” approach (DT) (Minin et al. 2003) (table 2), calculates the relative importance of every parameter, and computes model-averaged estimates of these, including a model-averaged estimate of the tree topology (Posada and Buckley 2004).

Model Selection with jModelTest

jModelTest is essentially a front-end to a computational pipeline that takes advantage of existing programs for running different tasks. Basically, this pipeline (fig. 1) includes:

- “ReadSeq” (Gilbert 2007): for conversion among different DNA sequence alignment formats.
- “Phyml” (Guindon and Gascuel 2003): for the likelihood calculations, including estimates of model parameters and trees.
- “Consense” (from the PHYLIP package) (Felsenstein 2005): to calculate weighted and strict consensus trees representing model-averaged phylogenies.

Likelihood Calculations

Likelihood calculations, including model parameters and tree estimates, are carried out with Phyml (Guindon and Gascuel 2003). The tree topology used in these calculations can be the same across models (fixed) or optimized for each one. Fixed tree topologies can be estimated with the BIONJ algorithm (Gascuel 1997) upon JC distances (Jukes and Cantor 1969) or user-defined. Alternatively, a BIONJ or an ML tree can be estimated under each model. In all cases, branch lengths are estimated and counted as parameters.

Custom Set of Models

Currently, there are 11 different nucleotide substitution schemes implemented in jModelTest, which combined with equal or unequal base frequencies (+F), a proportion of invariable sites (+I), and rate variation among sites (+G), result in 88 distinct models (table 1). The program offers the possibility of defining to a reasonable extent which models are included in the candidate set.

Sequential Likelihood Ratio Tests

A series of likelihood ratio tests (LRTs) can be implemented under a particular hierarchy (hLRTs), in which the user can specify their order, and whether parameters are added (forward selection) or removed (backward selection). Alternatively, the order of the LRTs can be set dynamically (dLRTs) (Posada and Crandall 2001), by comparing the current model with the one that is one hypothesis away and provides the largest increase (under forward selection) or smallest decrease (under backward
select) in likelihood. The hLRTs and dLRTs will be available only if the likelihood scores were calculated upon a fixed topology, due to the nesting requirement of the \( \chi^2 \) approximation.

Information Criteria

The program implements 3 different information criteria: the AIC (Akaike 1973), the BIC (Schwarz 1978), and a performance-based approach based on decision theory (DT) (Minin et al. 2003). Under the AIC framework, there is also the possibility of using a corrected version for small samples (AICc) (Sugiura 1978; Hurvich and Tsai 1989), instead of the standard AIC. In this case, sample size has to be specified, which by default is approximated as the number of sites in the alignment (note that the sample size of an alignment is presently an unknown quantity).

### Table 1

**Substitution Models Available in jModelTest**

<table>
<thead>
<tr>
<th>Model(^a)–(^c)</th>
<th>Free Parameters</th>
<th>Base Frequencies</th>
<th>Substitution Rates</th>
<th>Substitution Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>JC</td>
<td>( k )</td>
<td>Equal</td>
<td>AC = AG = AT = CG = CT = GT</td>
<td>000000</td>
</tr>
<tr>
<td>F81</td>
<td>( k + 3 )</td>
<td>Unequal</td>
<td>AC = AG = AT = CG = CT = GT</td>
<td>000000</td>
</tr>
<tr>
<td>K80</td>
<td>( k + 1 )</td>
<td>Equal</td>
<td>AC = AT = CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>HKY</td>
<td>( k + 1 )</td>
<td>Equal</td>
<td>AC = AT = CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TrNe</td>
<td>( k + 2 )</td>
<td>Equal</td>
<td>AC = AT = CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TPM1</td>
<td>( k + 5 )</td>
<td>Unequal</td>
<td>AC = AT, CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TPM1u</td>
<td>( k + 5 )</td>
<td>Equal</td>
<td>AC = AT, CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TPM2</td>
<td>( k + 2 )</td>
<td>Unequal</td>
<td>AC = AT, CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TPM2u</td>
<td>( k + 2 )</td>
<td>Equal</td>
<td>AC = AT, CG = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TPM3</td>
<td>( k + 2 )</td>
<td>Equal</td>
<td>AC = CG, AT = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TPM3u</td>
<td>( k + 5 )</td>
<td>Equal</td>
<td>AC = CG, AT = GT, AG = CT</td>
<td>012210</td>
</tr>
<tr>
<td>TIM1e</td>
<td>( k + 3 )</td>
<td>Equal</td>
<td>AC = GT, AT = CG, AG, CT</td>
<td>012230</td>
</tr>
<tr>
<td>TIM1</td>
<td>( k + 6 )</td>
<td>Unequal</td>
<td>AC = GT, AT = CG, AG, CT</td>
<td>012230</td>
</tr>
<tr>
<td>TIM2</td>
<td>( k + 3 )</td>
<td>Equal</td>
<td>AC = AT, CG = GT, AG, CT</td>
<td>012230</td>
</tr>
<tr>
<td>TIM2e</td>
<td>( k + 6 )</td>
<td>Equal</td>
<td>AC = AT, CG = GT, AG, CT</td>
<td>012230</td>
</tr>
<tr>
<td>TIM3</td>
<td>( k + 6 )</td>
<td>Unequal</td>
<td>AC = AT, CG = GT, AG, CT</td>
<td>012230</td>
</tr>
<tr>
<td>TIM3u</td>
<td>( k + 5 )</td>
<td>Equal</td>
<td>AC = CG, AT = GT, AG = CT</td>
<td>012230</td>
</tr>
<tr>
<td>TVM</td>
<td>( k + 4 )</td>
<td>Equal</td>
<td>AC, AT, CG, GT, AG = CT</td>
<td>012314</td>
</tr>
<tr>
<td>TVM</td>
<td>( k + 7 )</td>
<td>Unequal</td>
<td>AC, AT, CG, GT, AG = CT</td>
<td>012314</td>
</tr>
<tr>
<td>SYM</td>
<td>( k + 5 )</td>
<td>Equal</td>
<td>AC, AG, AT, CG, CT, GT</td>
<td>012345</td>
</tr>
<tr>
<td>GTR</td>
<td>( k + 8 )</td>
<td>Unequal</td>
<td>AC, AG, AT, CG, CT, GT</td>
<td>012345</td>
</tr>
</tbody>
</table>

### Model Selection Strategies Implemented in jModelTest

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Hierarchical Likelihood Ratio Tests</th>
<th>Dynamical Likelihood Ratio Tests</th>
<th>Akaic Information Criterion</th>
<th>Bayesian Information Criterion</th>
<th>Performance-Based Selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Base tree</td>
<td>hLRTs</td>
<td>dLRTs</td>
<td>AIC</td>
<td>BIC</td>
<td>DT</td>
</tr>
<tr>
<td>Nesting</td>
<td>Fixed</td>
<td>Fixed</td>
<td>Fixed, optimized</td>
<td>Fixed, optimized</td>
<td>Fixed, optimized</td>
</tr>
<tr>
<td>Selection</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Simultaneous</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Parameter</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Model</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

\( ^a \) DT weights are simply the rescaled reciprocal DT scores. This is a gross implementation very likely to change.
model-averaged estimates of these, using all the models in the candidate set, or a fraction included in a particular CI (see Posada and Buckley 2004).

Model-Averaged Phylogenies

jModelTest is able to compute an average estimate of the tree topology by building a consensus of the maximum likelihood (ML) trees for every model in the candidate set, weighting them with their model weights (AIC, BIC, or DT) (fig. 2). Indeed, this option is only available when the tree topology has been optimized for every model. The consensus tree is constructed using the Consense program from the PHYLIP package (Felsenstein 2005).

Software Platform and Availability

jModelTest is written in Java and can be started in any operating system with a Java Runtime Environment (see http://www.java.com). However, jModelTest uses other programs for different tasks, and these have been compiled for Mac OS X, Windows XP, and Linux. The package, including installation instructions, documentation, executables, and example data, is distributed free of charge for academic use from the software section at http://darwin.uvigo.es.

Conclusions

Model selection is an important issue in statistical phylogenetics, around which some questions still remain open (Kelchner and Thomas 2007). jModelTest addresses some of these, providing an increased flexibility for the user to explore the data and the role of the substitution model on the estimation of phylogenetic trees.

![Model-averaged tree of HIV-1 pol sequences](https://example.com/fig2.png)

**FIG. 2.**—Model-averaged tree of HIV-1 pol sequences. The topology shown is the consensus of 88 ML tree topologies, one for every model, weighted according to the AIC weights. The numbers on the branches represent uncertainty due to model selection. In this case, clades (AJ), (AIC), and (HG) are supported by the best and fourth best AIC models (GTR + G, AIC weight = 0.83; TIM3 + G, AIC weight = 0.01; respectively) and others, but not by the second or third best AIC models (GTR + I + G, AIC weight = 0.15; GTR + I, AIC weight = 0.01; respectively).
Acknowledgments

I want to thank a number of users of Modeltest that had made numerous comments and suggestions through the years. Special thanks to Stephane Guindon for his generous help with Phyml and to John Huelsenbeck for suggesting the stochastic calculation of CIs. I want to acknowledge Sudhir Kumar for inviting me to present the latest advances in Modeltest at the 2006 SMBE annual meeting, which finally prompted the completion of jModelTest.

Literature Cited


Sudhir Kumar, Associate Editor

Accepted April 2, 2008