SMS: Smart Model Selection in PhyML
Vincent Lefort, Jean-Emmanuel Longueville, Olivier Gascuel

To cite this version:
Vincent Lefort, Jean-Emmanuel Longueville, Olivier Gascuel. SMS: Smart Model Selection in PhyML. Molecular Biology and Evolution, Oxford University Press (OUP), 2017, 34 (9), pp.2422-2424. 10.1093/molbev/msx149. lirmm-01794206

HAL Id: lirmm-01794206
https://hal-lirmm.ccsd.cnrs.fr/lirmm-01794206
Submitted on 13 Mar 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License
Brief Communication

SMS: Smart Model Selection in PhyML

Vincent Lefort, Jean-Emmanuel Longueville, and Olivier Gascuel*
1Institut de Biologie Computationnelle, LIRMM, UMR 5506 - CNRS et Université de Montpellier, Montpellier, France
2Unité de Bioinformatique Evolutive, C3BI, USR 3756 - Institut Pasteur et CNRS, Paris, France
*Corresponding author: E-mail: olivier.gascuel@pasteur.fr.
Associate editor: Tal Pupko

Abstract

Model selection using likelihood-based criteria (e.g., AIC) is one of the first steps in phylogenetic analysis. One must select both a substitution matrix and a model for rates across sites. A simple method is to test all combinations and select the best one. We describe heuristics to avoid these extensive calculations. Runtime is divided by \( \sim 2 \) with results remaining nearly the same, and the method performs well compared with ProtTest and jModelTest2. Our software, “Smart Model Selection” (SMS), is implemented in the PhyML environment and available using two interfaces: command-line (to be integrated in pipelines) and a web server (http://www.atgc-montpellier.fr/phyml-sms/).

Key words: model selection, heuristic procedure, AIC and BIC criteria, web server, PhyML.

Current phylogenetic programs provide users with a wide variety of models to represent both the variability of rates across sites (RAS) and the substitution process. With proteins, a large number of substitution matrices have been inferred for various protein types (e.g., membrane and mitochondrial) and origins (e.g., mammals and viruses). To select among these many models, statistical criteria (e.g., AIC [Akaike 1973] and BIC [Schwarz 1978]) are used to find the best likelihood/model-complexity tradeoff. A simple, standard approach is to test all models and then select the best one. This forms the basis of widely used, user-friendly software programs such as ProtTest for proteins (Abascal et al. 2005).

Here, we introduce a new software tool to achieve this task: SMS, which stands for “Smart Model Selection.” This tool is very simple to use, as SMS is fully integrated into the PhyML web server (fig. 1a and b; Guindon et al. 2010). SMS can also be used as a standalone application and is freely available for download (http://www.atgc-montpellier.fr/sms/). SMS uses heuristic strategies to avoid testing all models and options. These strategies are partly inspired by Posada and Crandall (1998) and Darriba et al. (2012). Notably, the latter proposed a fast method called “model filtering” to focus on the most promising substitution matrices for DNA, whereas our heuristic for proteins also ranks the matrices based on their proximity to the data being analyzed. Moreover, SMS simplifies some calculations to save computing time. This is especially relevant in a pipeline context for running extensive phylogenetic analyses, for example, to study protein families. Below, we summarize the main features of SMS and its performance compared with the exhaustive approach, as well as to jModelTest2 (Darriba et al. 2012) and ProtTest. Complete details on algorithms, benchmark data sets, and comparison results are available in Supplementary Material.

With proteins, all substitution matrices available in PhyML are also available in SMS (fig. 1c; 17 matrices). Moreover, users can add their own matrices. All matrices can be used with the option +F (amino-acid frequencies are estimated from the data) and −F (preestimated frequencies). SMS only has two options to model RAS: +\( \Gamma \) (gamma distribution) and +\( \Gamma + I \) (one class of invariant sites is added). Extensive comparisons (supplementary table S4, Supplementary Material online) with 500 representative protein data sets showed that the +I option alone is rarely selected (1/500 with AIC, 4/500 with BIC), and the same holds for the −\( \Gamma + I \) or “none” option (3/500 with AIC, 4/500 with BIC). Protein multiple sequence alignments (MSAs) usually have few constant sites (median proportion in our data sets \( \approx 3\% \)), and we expect a high variability of site rates caused by the variability of functional and structural constraints acting along protein sequences. These results and choices are thus biologically consistent. SMS has a total of 17 (matrices) \( \times 2 \oplus F \oplus F \) options systematically, the matrices are ranked based on the similarity of the amino-acid frequencies in the data and those preestimated in the matrix; iii) SMS selects the best “decoration” (i.e., RAS and +F−F options) to avoid computing both +F and −F options systematically, the matrices are ranked based on the similarity of the amino-acid frequencies in the data and those preestimated in the matrix; iv) SMS selects the best substitution matrix and +F−F option; to avoid computing both +F and −F options systematically, the matrices are ranked based on the similarity of the amino-acid frequencies in the data and those preestimated in the matrix; v) SMS selects the best “decoration” (i.e., RAS and +F−F options) for the best matrix. The gain in computing time is explained by the fact that, for most substitution matrices, SMS performs only 1 or 2 likelihood evaluations per matrix (1.75 on average, corresponding to different decorations), compared with four for the exhaustive approach, which evaluates all decorations for all matrices.

© The Author 2017. Published by Oxford University Press on behalf of the Society for Molecular Biology and Evolution. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/4.0/), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com


Downloaded from https://academic.oup.com/molevol/article-abstract/34/9/2422/3788860 by guest on 13 March 2019
Computations with DNA are simpler than with proteins, as today’s MSAs are most often large enough for GTR to be best compared to other substitution matrices. Moreover, the simplest matrices are not satisfactory because they do not account for the transition/transversion ratio and/or unequal base frequencies. Experiments with 500 representative MSAs confirmed these hypotheses, and are congruent with the large-scale study of (Arbiza et al. 2011). With AIC, GTR is best for 343/500 MSAs, whereas JC69, K80, and F81 are all best with 9/500 MSAs only (supplementary table S3, Supplementary Material online). However, with BIC, K80 is best for 48/500 MSAs. SMS thus uses four substitution matrices: GTR, TN93, HKY85, and K80, which are combined with +I, +F, +1, and “none” (all four RAS options are useful, supplementary table S3, Supplementary Material online), that is, a total of 4 x 4 = 16 models. On average, SMS computes the likelihood value of ~6 models with AIC and 7.5 with BIC, thus dividing the computing time by ~2 as compared to the exhaustive approach using the same models. Based on the user’s selected criterion (AIC/BIC), the basic principle in SMS as follows: i) using a BioNJ tree topology, SMS estimates the branch lengths and model parameters for GTR and the four RAS options; ii) using the “most promising” RAS option with GTR, SMS selects the best matrix in a stepwise manner: SMS compares GTR and TN93; if GTR is better, then SMS stops and keeps GTR; otherwise, SMS compares HKY85 to TN93, and so on (remember that GTR, TN93, HKY85, and K80 are nested); iii) SMS selects the best RAS option for the best matrix. This simple approach, combined with a relatively small set of models, makes SMS nearly as fast as jModelTest2.
the sets of models are more different than with proteins, average AIC/BIC difference is in favor of SMS. With DNA, SMS finds a better model than ProtTest in some cases; when the models differ (35/500 MSAs), ProtTest finds the same model in most cases (73/120); when the models differ (120 and 192 MSAs) SMS and ProtTest differ for 120 and 192 MSAs and the average AIC/BIC difference is clearly in favor of SMS. ProtTest finds a better model than jModelTest2 in 75% of the cases, whereas the gain in AIC/BIC difference is clearly in favor of SMS. The computing time gains of SMS with proteins are quite substantial in practice (supplementary fig. S1, Supplementary Material online). For example, ProtTest requires more than 100 h to process the largest MSA (1,151 taxa and 798 sites), whereas SMS requires ~20 h using the same computer.

Supplementary Material

Supplementary data are available at Molecular Biology and Evolution online.

Acknowledgment

This research was supported by the Institut Français de Bioinformatique (RENABI-IFB, Investissements d’Avenir).

References


Table 1. Method Comparison with 500 DNA, and 500 Protein Representative MSAs.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Data</th>
<th>Criterion</th>
<th>Same Model</th>
<th>SMS Better</th>
<th>SMS Worse</th>
<th>Δ AIC &amp; Δ BIC per taxon per site</th>
<th># PhyML Runs SMS/other</th>
<th>Speed Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMS versus Exhaustive</td>
<td>DNA</td>
<td>AIC</td>
<td>468</td>
<td>na</td>
<td>14</td>
<td>4.6 x 10^-5</td>
<td>6.1/16</td>
<td>1.9–2.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIC</td>
<td>476</td>
<td>na</td>
<td>24</td>
<td>8.0 x 10^-5</td>
<td>7.5/16</td>
<td>1.7–1.9</td>
</tr>
<tr>
<td>SMS versus Exhaustive</td>
<td>Protein</td>
<td>AIC</td>
<td>494</td>
<td>na</td>
<td>6</td>
<td>3.7 x 10^-3</td>
<td>29.3/68</td>
<td>2.2–2.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIC</td>
<td>497</td>
<td>na</td>
<td>3</td>
<td>3.8 x 10^-3</td>
<td>30.2/68</td>
<td>2.1–2.0</td>
</tr>
<tr>
<td>SMS versus jModelTest2</td>
<td>DNA</td>
<td>AIC</td>
<td>380</td>
<td>85</td>
<td>35</td>
<td>-2.5 x 10^-5</td>
<td>6.1/78</td>
<td>1.1–0.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIC</td>
<td>308</td>
<td>151</td>
<td>41</td>
<td>-1.1 x 10^-4</td>
<td>7.5/78</td>
<td>0.9–0.8</td>
</tr>
<tr>
<td>SMS versus jModelTest2</td>
<td>Protein</td>
<td>AIC</td>
<td>465</td>
<td>14</td>
<td>21</td>
<td>-8.9 x 10^-4</td>
<td>29.3/120</td>
<td>3.7–3.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BIC</td>
<td>465</td>
<td>12</td>
<td>23</td>
<td>-7.5 x 10^-4</td>
<td>30.2/120</td>
<td>3.5–3.2</td>
</tr>
</tbody>
</table>

Note.—The “Exhaustive” approach uses the same set of models as SMS and evaluates all of them. “Same model”: number of times (among 500 MSAs) where both methods return the same model; “SMS better”: number of times where the model returned by SMS has a lower AIC/BIC value; “SMS worse”: number of times where the model returned by SMS has a higher AIC/BIC value; “Δ AIC and Δ BIC per taxon per site”: when both models were different, we computed the difference in AIC/BIC per taxon per site, and averaged the results over all MSAs showing a model difference (a negative/positive value means that SMS’s model is better/worse in terms of AIC/BIC); # PhyML runs: number of PhyML runs for one method versus the other; “Speed increase”: for each MSA, we computed the computing time ratio of the method being compared with respect to SMS (e.g., 2 means that SMS is twice as fast), with the column displaying: i) the median value among the 500 speedup ratios for all MSAs, ii) the median value for the 50 largest MSAs (number of sites x number of taxa; see supplementary fig. S1, Supplementary Material online for additional computing time results with large MSAs).