The Journal of Open Source Software

modelestimator v2: inferring amino acid replacement rates from multi-sequence alignments

Ruben Ridderström[1] **and Lars Arvestad**[1,2]

1 Department of Mathematics, Stockholm University, Sweden **2** Swedish e-science Research Centre

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 24 February 2024
Published: unpublished

Summary

Phylogenetic inference is often based on generative probabilistic models, under which one seeks to maximize the likelihood of an estimated tree or estimate a posterior distribution on trees for some given DNA or protein sequence data. The core of most evolution models describe how symbols in the input sequences are replaced. For models of DNA substitution, the few involved parameters can be inferred during tree estimation, but that approach is difficult for protein models due to the many free parameters. However, there is a range of standard models that provide good approximations in many cases, for example Dayhoff (Dayhoff et al., 1978), WAG (Whelan & Goldman, 2001), JTT (Jones et al., 1992), LG (Le & Gascuel, 2008), VT (Müller & Vingron, 2000), and more.

Methods have been developed to estimate "empirical models" – replacement rate matrices directly from peptide multiple sequence alignments (Arvestad, 2006; Holmes & Rubin, 2002; Minh et al., 2021). This paper reports a re-implementation of the `modelestimator` method (Arvestad, 2006).

Improvements

For the re-implementation, we sought improvements in some basic aspects:

1. User friendliness
2. Interoperability
3. Computational efficiency
4. Maintainability

We consider *User friendliness* to include easy deployment and following Unix practices for command-line tools. The new version is written in Python and we therefore use PyPI, the Python Package Index (*PyPI*, n.d.), to provide straightforward software installation. Proper care has been taken to provide a clean command-line user interface, which improves on the previous version of `modelestimator`.

Interoperability has been improved in two ways. Firstly, we get easy parsing of several standard input formats using Biopython (Cock et al., 2009). Secondly, there are command-line options that ensure that the output is suitable for popular phylogenetic inference tools, including MrBayes (Ronquist et al., 2012), IQTREE2 (Nguyen et al., 2015), PhyML (Guindon et al., 2010), RAxML (Stamatakis, 2014), and PAML (Yang, 2007).

Choosing Python for the implementation may not be optimal for *Computational efficiency*, but since NumPy (Oliphant, 2006) is used for core computations, `modelestimator` is fast enough for problem-free interactive usage even on large input alignments.

To ensure *Maintainability*, we have implemented a set of unit tests, put the source code at GitHub, where development and community infrastructure is provided, and implemented continuous integration tests. Furthermore, leaving the old code base for modelestimator (a combination of Perl and Octave) is a step towards maintainability.

We recommend installing modelestimator using PyPI: “pip install modelestimator-v2”.

Statement of need

As shown in (Arvestad, 2006), inferring an amino-acid replacement-model for an alignment often improves the likelihood of a phylogeny inference. Consider estimating a phylogeny for an alignment of some Cres/Testatin homologs [CTES; Frygelius et al. (2010)], with 65 sequences and 441 columns. In the following, we use modelestimator to estimate replacement models for this specific alignment and compare it to standard models and methods using PhyML (Guindon et al., 2010) and IQTREE2 (Nguyen et al., 2015).

PhyML with modelestimator

Running PhyML to infer a phylogenetic tree using default options means using the LG model and empirical equilibrium frequencies:

```
$ phyml -d aa -i CTES.phy
```

This estimation took about 2m40s on a MacBook Pro with an M1 Pro CPU and returns a tree with a log-likelihood of -12418.6. LG might not be the right model, so it is a good idea to consider alternative models. However, we can also try modelestimator and get a model tailored for the present alignment:

```
$ modelestimator -a phyml -f phylip CTES.phy  >  ratematrix.txt
```

Option -a phyml makes sure the output works for PhyML, but the format is the same for IQTREE2. This estimation took about a second on the same computer.

The resulting model can be used by PhyML in a new attempt. Using “--model custom” instructs PhyML to use a user-provided rate matrix and “--aa ratematrix.txt” locates the file containing it.

```
$ phyml -d aa -i CTES.phy  --model custom --aa ratematrix.txt
```

The resulting tree is slightly different and has a log-likelihood of -12206.7, an improvement by about 200. That is a significant improvement for a long-running ML computation given that finding the rate matrix only took a second.

Using IQTREE2, ModelFinder, and modelestimator

IQTREE2 provides a convenient method [ModelFinder; Kalyaanamoorthy et al. (2017)] for automating model search among a large number of published models, including combinations with models for rate heterogeneity. In addition, support for making a statistically sensible model choice (with respect to the number of free parameters in the model), makes IQTREE2 helpful.

Applying IQTREE2 to our alignment, using default options,

```
$ iqtree2 -s CTES.phy
```

compares 1225 model combinations and takes almost 8 minutes using a single CPU core. The JTT+R3 model is chosen based on the Bayesian Information Criterion (BIC), and it allows a tree with a log-likelihood of -12289.5 to be inferred.

Using the model estimated by modelestimator is done like this:

```
$ iqtree2 -s CTES.phy -m ratematrix.txt
```

IQTREE2 now finds a slightly different tree with this specific rate matrix and the log-likelihood increases to -12278.4. The computation takes about 40 seconds, by avoiding ModelFinder, so we have a 12x speedup with a better log-likelihood.

One can also request that IQTREE adds a rate heterogeneity model by specifying the model "ratematrix.txt+I+G4". The same tree is inferred, but the log-likelihood improves to -12208.3.

Using IQTREE2, QMaker, and modelestimator

IQTREE has support for using ML for estimating an empirical model with the QMaker method (Minh et al., 2021). We can apply QMaker with the already inferred tree and note JTT as a good starting point for model search:

```
$ iqtree2 -seed 1 -T 1 -s CTES.phy -te treefile --init-model JTT \
          --model-joint GTR20+FO --prefix GTR20_FO
```

This inference has a better log-likelihood, -12136.2, than what was achieved with `modelestimator`. However, it requires 7 minutes to compute (single core) even though JTT is given as a suitable starting point and the previously inferred tree is used to initialize the search.

Now try the same search with the rate matrix from `modelestimator` as a starting point.

```
$ iqtree2 -seed 1 -T 1 -s CTES.phy -te treefile --init-model ratematrix.txt \
          --model-joint GTR20+FO --prefix GTR20_FO_with_modelestimator
```

This computation takes about a minute and finds a model that calculates the likelihood of the tree to -12135.9. That is, we get a seven-fold speedup due to a better starting point for the parameter search, and the cost is a computation that took us a second.

Conclusion

The reimplemented `modelestimator` is easy to install and run, and speeds up model choice or inference. With enough data, it estimates rate matrices that yield higher likelihoods than standard models.

References

Arvestad, L. (2006). Efficient methods for estimating amino acid replacement rates. *Journal of Molecular Evolution*, *62*(6), 663–673. https://doi.org/10.1007/s00239-004-0113-9

Cock, P. J. A., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B., & Hoon, M. J. L. de. (2009). BioPython: Freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422–1423. https://doi.org/10.1093/bioinformatics/btp163

Dayhoff, M., Schwartz, R., & Orcutt, B. (1978). A model of evolutionary change in proteins. *Atlas of Protein Sequences and Structures*, 345–352.

Frygelius, J., Arvestad, L., Wedell, A., & Töhönen, V. (2010). Evolution and human tissue expression of the cres/testatin subgroup genes, a reproductive tissue specific subgroup of the type 2 cystatins. *Evolution & Development*, *12*(3), 329–342. https://doi.org/10.1111/j.1525-142X.2010.00418.x

Guindon, S., Dufayard, J.-F., Lefort, V., Anisimova, M., Hordijk, W., & Gascuel, O. (2010). New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. *Systematic Biology*, *59*(3), 307–321. https://doi.org/10.1093/sysbio/syq010

Holmes, I., & Rubin, G. M. (2002). An expectation maximization algorithm for training hidden substitution models. *Journal of Molecular Biology*, *317*(5), 753–764. https://doi.org/10.1006/jmbi.2002.5405

Jones, D. T., Taylor, W. R., & Thornton, J. M. (1992). The rapid generation of mutation data matrices from protein sequences. *Computer Applications in the Biosciences : CABIOS*, *8*(3), 275–282. https://doi.org/10.1093/bioinformatics/8.3.275

Kalyaanamoorthy, S., Minh, B. Q., Wong, T. K., Von Haeseler, A., & Jermiin, L. S. (2017). ModelFinder: Fast model selection for accurate phylogenetic estimates. *Nature Methods*, *14*(6), 587–589. https://doi.org/10.1038/nmeth.4285

Le, S. Q., & Gascuel, O. (2008). An improved general amino acid replacement matrix. *Molecular Biology and Evolution*, *25*(7), 1307–1320. https://doi.org/10.1093/molbev/msn067

Minh, B. Q., Dang, C. C., Vinh, L. S., & Lanfear, R. (2021). QMaker: Fast and Accurate Method to Estimate Empirical Models of Protein Evolution. *Systematic Biology*, *70*(5), 1046–1060. https://doi.org/10.1093/sysbio/syab010

Müller, T., & Vingron, M. (2000). Modeling amino acid replacement. *Journal of Computational Biology*, *7*(6), 761–776. https://doi.org/10.1089/10665270050514918

Nguyen, L.-T., Schmidt, H. A., Haeseler, A. von, & Minh, B. Q. (2015). IQ-TREE: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, *32*(1), 268–274. https://doi.org/10.1093/molbev/msu300

Oliphant, T. E. (2006). *Guide to NumPy*. Trelgol Publishing USA.

PyPI: The python package index. (n.d.). pypi.org.

Ronquist, F., Teslenko, M., Mark, P. van der, Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., & Huelsenbeck, J. P. (2012). MrBayes 3.2: Efficient Bayesian Phylogenetic Inference and Model Choice Across a Large Model Space. *Systematic Biology*, *61*(3), 539–542. https://doi.org/10.1093/sysbio/sys029

Stamatakis, A. (2014). RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, *30*(9), 1312–1313. https://doi.org/10.1093/bioinformatics/btu033

Whelan, S., & Goldman, N. (2001). A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular Biology and Evolution*, *18*(5), 691–699. https://doi.org/10.1093/oxfordjournals.molbev.a003851

Yang, Z. (2007). PAML 4: Phylogenetic Analysis by Maximum Likelihood. *Molecular Biology and Evolution*, *24*(8), 1586–1591. https://doi.org/10.1093/molbev/msm088