

Note: Supplementary information is available at <http://www.nature.com/doi/funder/10.1038/nmeth.2106>.

ACKNOWLEDGMENTS

The authors thank M.H. Tatham and M. Trost for comments on the manuscript. This work was supported by a Sir Henry Wellcome Fellowship awarded by the Wellcome Trust (sponsor reference 088957/Z/09/Z, to I.M.).

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Ivan Matic¹, Ivan Ahel² & Ronald T Hay¹

¹Wellcome Trust Centre for Gene Regulation and Expression, College of Life Sciences, University of Dundee, Scotland, UK. ²Cancer Research UK, Paterson Institute for Cancer Research, University of Manchester, Manchester, UK.
e-mail: i.matic@dundee.ac.uk

1. Anonymous. *Nat. Methods* **9**, 419 (2012).
2. Hengel, S.M. & Goodlett, D.R. *Int. J. Mass Spectrom.* **312**, 114–121 (2012).
3. Laing, S., Koch-Nolte, F., Haag, F. & Buck, F. *J. Proteomics* **75**, 169–176 (2011).
4. Huttlin, E.L. *et al. Cell* **143**, 1174–1189 (2010).
5. Laing, S., Unger, M., Koch-Nolte, F. & Haag, F. *Amino Acids* **41**, 257–269 (2011).

jModelTest 2: more models, new heuristics and parallel computing

To the Editor: The statistical selection of best-fit models of nucleotide substitution is routine in the phylogenetic analysis of DNA sequence alignments¹. With the advent of next-generation sequencing technologies, most researchers are moving from phylogenetics to phylogenomics, in which large sequence alignments typically include hundreds or thousands of loci. Phylogenetic resources therefore need to be adapted to a high-performance computing paradigm so as to allow demanding analyses at the genomic level. Here we introduce jModelTest 2, a program for nucleotide-substitution model selection that incorporates more models, new heuristics, efficient technical optimizations and parallel computing.

jModelTest 2 includes important features not present in the previous versions^{2,3} (Supplementary Table 1). We expanded the set of candidate models from 88 to 1,624, and we implemented two heuristics for model selection: a greedy, hill-climbing hierarchical clustering approach (Supplementary Note 1) and a filtering algorithm based on similarity among parameter estimates (Supplementary Note 2).

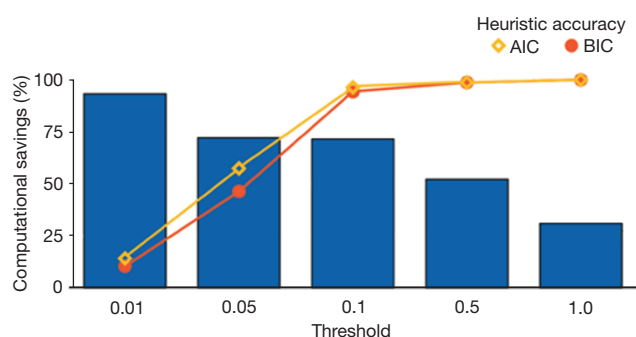


Figure 1 | Benchmarking of the filtering heuristic in jModelTest 2. The threshold of the filtering heuristic (Supplementary Note 2) is directly correlated with the probability of finding the true best-fit model (heuristic accuracy) and inversely related to the number of models for which we avoided the likelihood calculation (computational savings). AIC, Akaike information criterion⁵; BIC, Bayesian information criterion.

jModelTest 2 is written in Java, and it can run on Windows, Macintosh and Linux platforms. Source code and binaries are freely available from <https://code.google.com/p/jmodeltest2/>. The package includes detailed documentation and examples, and a discussion group is available at <https://groups.google.com/forum/#!forum/jmodeltest/>.

We evaluated the accuracy of jModelTest 2 using 10,000 data sets simulated under a large variety of conditions (Supplementary Note 3). Using the Bayesian information criterion⁴ for model selection, jModelTest 2 identified the generating model 89% of the time (Supplementary Table 2); in the remaining cases, jModelTest 2 selected a model similar to the generating one. Accordingly, model-averaged estimates of model parameters were highly precise (Supplementary Table 3). In these simulations, the two selection heuristics that we developed were accurate and efficient. Using the hierarchical clustering heuristic, we found the same best-fit model as the full search 95% of the time. With the similarity filtering approach, we reduced the number of models evaluated by 60% on average and found the global best-fit model 99% of the time (Fig. 1 and Supplementary Note 2).

jModelTest 2 can be executed in high-performance computing environments as (i) a desktop version with a user-friendly interface for multicore processors, (ii) a cluster version that distributes the computational load among nodes, and (iii) as a hybrid version that can take advantage of a cluster of multicore nodes. An experimental study with real and simulated data sets showed remarkable computational speedups compared to previous versions (Supplementary Note 4). For example, the hybrid approach executed on the Amazon EC2 cloud with 256 processes was 182–211 times faster. For relatively large alignments (138 sequences and 10,693 sites), this could be equivalent to a reduction of the running time from nearly 8 days to around 1 hour.

Note: Supplementary information is available at <http://www.nature.com/doi/funder/10.1038/nmeth.2109>.

ACKNOWLEDGMENTS

This work was financially supported by the European Research Council (ERC-2007-Stg 203161-PHYGENOM to D.P.), Spanish Ministry of Science and Education (BFU2009-08611 to D.P.) and Xunta de Galicia (Galician Thematic Networks RGB 2010/90 to D.P. and GHPC2 2010/53 to R.D.).

COMPETING FINANCIAL INTERESTS

The authors declare no competing financial interests.

Diego Darriba^{1,2}, Guillermo L Taboada², Ramón Doallo² & David Posada¹

¹Department of Biochemistry, Genetics and Immunology, University of Vigo, Vigo, Spain. ²Computer Architecture Group, University of A Coruña, A Coruña, Spain.
e-mail: dposada@uvigo.es

1. Sullivan, J. & Joyce, P. *Annu. Rev. Ecol. Evol. Syst.* **36**, 445–466 (2005).
2. Posada, D. & Crandall, K.A. *Bioinformatics* **14**, 817–818 (1998).
3. Posada, D. *Mol. Biol. Evol.* **25**, 1253–1256 (2008).
4. Schwarz, G. *Ann. Stat.* **6**, 461–464 (1978).
5. Akaike, H. *IEEE Trans. Automat. Contr.* **19**, 716–723 (1974).

CircadiOmic: integrating circadian genomics, transcriptomics, proteomics and metabolomics

To the Editor: Circadian rhythms govern a large array of physiological and metabolic functions¹. It is critical to decode circadian oscillations by integrating multiple ‘omic’ approaches. Circadian genomic and