Supplementary Information

Nature Methods

jModelTest 2: more models, new heuristics and parallel computing

Diego Darriba, Guillermo L. Taboada, Ramón Doallo and David Posada

Supplementary Table 1 | New features in jModelTest 2. jModelTest 2 implements a number of new features that facilitate model selection among more models and for large data sets.

New feature	Description
1. Exhaustive GTR submodels	All the 203 different partitions of the GTR rate matrix[1] can be included in the candidate set of models. When combined with rate variation (+I,+G, +I+G) and equal/unequal base frequencies the total number of possible models is $203 \times 8 = 1624$.
2. Hill-climbing hierarchical clustering	Calculating the likelihood score for a large number of models can be extremely time-consuming. This hill-climbing algorithm implements a hierarchical clustering to search for the best-fit models within the full set of 1624 models, but optimizing at most 288 models while maintaining model selection accuracy.
3. Heuristic filtering	Heuristic reduction of the candidate models set based on a similarity filtering threshold among the GTR rates and the estimates of among-site rate variation.
4. Absolute model fit	Information criterion distances can be calculated for the best-fit model against the unconstrained multinomial model (based on site pattern frequencies)[2]. This is computed by default when the alignment does not contain missing data/ambiguities, but can also be approximated otherwise.
5. High Performance Computing	Model selection can be executed in parallel in multicore desktop machines and in HPC clusters achieving large speedups.
6. Topological summaries	Tree topologies supported by the different candidate models are summarized in the html log, including confidence intervals constructed from cumulative models weights, plus Robinson-Foulds[3] and Euclidean distances to the best-fit model tree.
7. Alignment sample size	The alignment sample size used for the AICc and BIC frameworks can be calculated according to alignment length (L) as before, but also as the number of variable sites, $L \times$ the number of sequences (N), Shannon entropy and Normalized Shannon entropy multiplied by $N \times L$.
8. User-friendly HTML log	The results of the model selection can be displayed in html format including maximum likelihood trees derived from each model and linked to http://www.phylowidget.org[4] for graphical depiction.

Nature Methods: doi:10.1038/nmeth.2109

Supplementary Table 2 | Model selection accuracy. Model selection accuracy was defined as the number of times the best-fit model selected by jModelTest 2 was the generating model. In case these models differed, we kept track of which components of the generating model were identified correctly (base frequencies, partition, rate variation among sites). In this table we show the model selection accuracy (%) across 10,000 data sets. *Num Params* refers to the mean number of parameters of the best-fit models. *Full Model* refers to the number of times the exact generating model was selected as the best-fit model. *Partition* refers to the number of times the structure of the R-matrix was correctly identified. *Rate Variation* refers to the number of times the rate variation parameter combinations (+I, +G, +I+G) were correctly identified.

Criterion	Num Params	Full Model	Partition	Rate Variation
AIC	5.62	62.36	70.64	93.11
BIC	4.99	89.34	89.87	99.29
DT	4.99	89.30	89.94	99.27

Nature Methods: doi:10.1038/nmeth.2109

Supplementary Table 3 | Mean square errors for model averaged estimates. To obtain the MSEs, and because the generating model and the best-fit model can differ, we did not consider every case for every parameter. For the base frequencies, transition/transversion ratio and R-matrix we considered all cases (see **Supplementary Note 3**). For the proportion of invariable sites in +I models (*p-invI*) we considered only cases where the generating model was M (*p-inv*=0) or M+I (*p-inv* = simulated). For the proportion of invariable sites in +I+G models (*p-invIG*) we considered only cases where the generating model was M+I+G (*p-inv* = simulated). For the alpha shape of the gamma rate variation among sites in M+G models (*alphaG*) we considered only cases where the generating model was M+G (*alpha* = simulated). For the alpha shape of the gamma rate variation among sites in M+I+G models (*alphaIG*) we considered only cases where the generating model was M+I+G (*p-inv* = simulated).

Parameter	MSE (AIC)	MSE (BIC)
fA	0.01	0.01
fC	0.01	0.01
fG	0.01	0.01
fT	0.01	0.01
titv	0.88	0.75
Ra	3.93	2.46
Rb	13.46	12.03
Rc	6.12	10.95
Rd	4.92	3.26
Re	6.16	5.38
p-invI	0.83	0.83
p-invIG	0.02	0.02
alphaG	0.09	0.09
alphaIG	0.14	0.14

Nature Methods: doi:10.1038/nmeth.2109

Supplementary Note 1 | Hill-climbing hierarchical clustering algorithm. Models of DNA substitution are defined by a rate matrix R, which describes the rate at which nucleotides of one type change into another type (e.g., r_{ij} for $i \neq j$), is the rate at which base *i* goes to base *j*. Because the models are most of the time assumed time reversible for tractability, this rate matrix is in practice always symmetrical. Therefore, we can define the rate matrix just in terms of the upper triangular matrix as a vector of 6 rates (r_{AC}, r_{AG}, r_{AT}, r_{CG}, r_{CT}, r_{GT}). Note that we assume all rates are relative to r_{GT}, which is set to 1.0. We can reduce the number of free parameters further forcing several of these rates to be the same. For example, we could assume the same rate for the two types of transitions (r_{AG} = r_{CT}). An easy way to label these partitions (i.e., the set of constraints) is indicating with 6 digits which rates are forced to be identical. For example, the JC[5] model has the partition 000000, where all the 6 rates among the 4 nucleotides are identical, while the GTR or SYM[6] models have the partition 012345, where all 6 rates are different. How many partitions are in between? The number of ways in which we can subdivide n elements into (non-empty) groups is given by the Bell numbers, $B(n)$, which in turn is the sum from $k = 1$ to $k = n$ of the number of ways to partition a set of n elements into k (non-empty) groups, which is given by the Stirling numbers of the second kind, $S(n,k)$:

$$B(n)=\sum_{k=0}^{n} S\left(n,k\right)=\sum_{k=0}^{n}\frac{1}{k!}\sum_{j=0}^{k}\left(\text{-}1\right)^{k\text{-}j}\binom{k}{j}j^{n}$$

$$S\left(n,k\right)=\sum_{j=0}^{k}\left(-1\right)^{k-j}\frac{j^{n-1}}{\left(j-1\right)!\left(k-j\right)!}=\frac{1}{k!}\sum_{j=0}^{k}\left(-1\right)^{k-j}\binom{k}{j}j^{n}$$

Accordingly, the *R*-matrix can be partitioned in $B(3) = 203$ different ways. For each *R*-matrix we can build 8 different models depending on the rate variation parameters (i.e., +I, +G and +I+G models) and whether we considered equal/unequal frequencies. Therefore, the total number of possible time reversible models is $203 \times 8 = 1624$. Indeed, the exhaustive computation of so many models is only feasible for small alignments or when computer power is not a problem. To alleviate this situation, we have implemented a simple, greedy hill climbing heuristic to search for the best-fit model in large candidate sets of models (up to 1624) without evaluating all of them (i.e., avoiding an *exhaustive* search). The algorithm is as follows:

1. Start with $n = 6$ and $k = 6$. There is only a single partition (012345) that fits this condition.
2. Select the best-fit model, $M_{current\_best}$, according to the chosen information-theoretic criterion (AIC, AICc or BIC).
3. Set $k = k - 1$.
4. Define a new set of models by exploring all possible merges of two groups into a single group.
5. Select the best-fit model, M_{best_merge}, from this set.

Nature Methods: doi:10.1038/nmeth.2109

6. If $M_{current\_best}$ has a better AIC/AICc/BIC score than M_{best_merge}, stop, otherwise continue
7. Set $M_{current\_best} = M_{best\_merge}$
8. Update the number of elements ($n = n - 1$).
9. Repeat from step 3 until the algorithm finds a local maxima or $k = 1$.

Note that in fact we are travelling in diagonal through the Stirling numbers of the second kind pyramid (**Fig. S1**), evaluating models only at those stages where $k = (n\text{-}1)$ for $n = 1...6$.

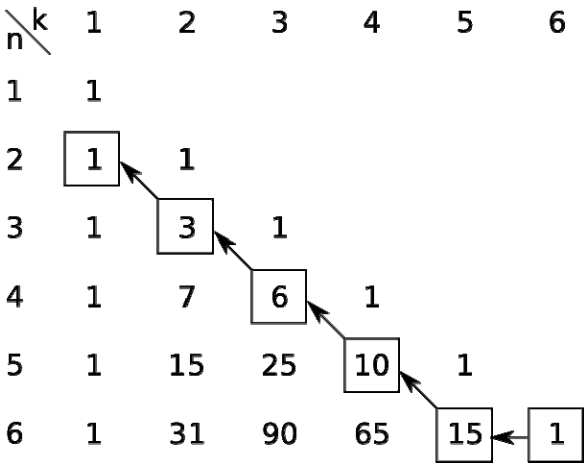

Figure S1 | Stirling numbers of the second kind. The rows sum to the n^{th} Bell number (e.g., 203 for $n = 6$). The squared numbers represent the stages of our hierarchical clustering algorithm. As long as our algorithm moves forward, the number of elements and groups are reduced by one until there is a single group.

In this heuristic the number of model partitions evaluated goes from a minimum of 1 to a maximum of 36 (i.e., up to 36 x 8 = 288 different models). Because, as any heuristic, this algorithm can get stuck in local optima, we have evaluated its performance analyzing 2,000 simulated alignments (as in **Supplementary Note 3** but considering all possible 203 *R*-matrices). In this case, our heuristic finds the same model as the exhaustive search 95% of the time. Note that a very similar heuristic to find model partitions across multigene data sets has just been developed[7].

Nature Methods: doi:10.1038/nmeth.2109

Supplementary Note 2 | Heuristic filtering. For very large alignments, even the computation of the likelihood of the 88 standard models implemented in the previous version of jModelTest can take a long time. We have developed a second heuristic to find the best-fit model without evaluating all candidate models. The basic idea is that model selection can be somewhat predicted from the most complex model. Our strategy attempts to significantly reduce the number of candidate models evaluated paying attention to the GTR+I+G rate matrix and the base frequencies estimates. For example, if the maximum likelihood estimates of the transition and transversion rates are very different and the likelihood score is low enough (so we expect noticeable likelihood differences among models), one could obviate the evaluation of a simple model like JC. We perform the filtering process in three main steps: (1) look at the rate matrix, where different enough rates will imply that models with equal rates will be excluded; (2) look at the base frequencies, where different enough frequencies will imply that models with equal base frequencies will be excluded; and (3) look at the among-site rate variation, where small *p-inv* or *alpha* estimates will imply that only site-homogeneous models will be considered.

To decide what can be considered different enough we defined a *filtering threshold* (δ). A higher δ means a larger model set and a smaller probability if getting trapped in local optima (i.e., the model selected is not the best one according to the selection criterion). On the other hand, a lower δmeans more possibilities of selecting the optimal model but less computational load. Although accurate among-site rate variation filtering should be presumably implemented from the GTR+I or GTR+G models, we prefer to use a single model (GTR+I+G) for every dataset. Because these two rate variation parameters (*alpha* and *pinv*) try to model the same thing, a proportion of invariable sites could be theoretically 'converted' into gamma rate variation in the +I+G model. Therefore, we use the two thresholds just described for excluding models at this step. Once the filtering is completed, we will obtain a set of excluded models that it is not necessary to optimize. The whole heuristic is as follows:

1. Optimize the GTR+I+G model, obtaining maximum likelihood estimates of the *R*-matrix (to facilitate notation r_{AC}, r_{AG}, r_{AT}, r_{CG}, r_{CT} and r_{GT} will be referred here as r_1, r_2, r_3, r_4, r_5 and r_6, respectively), base frequencies (π_1, π_2, π_3 and π_4), the proportion of invariant sites (*pinv* = ρ) and the alpha shape of the gamma distribution (*alpha* = α) estimates.
2. Set up a filtering threshold $\delta \in \Re > 0$
3. If the standard deviation of the rates r_i is high enough ($\sigma > 1.0$) standardize them to a *z*-score: $z_i = (r_i - \bar{r}_i) / \sigma_{r_i}$
4. Calculate the pairwise differences between every pair of rates $D_{ij} = \left| z_i - z_j \right|$
5. If $D_{ij} > \delta$, all models with equal *i* and *j* rates will be ignored during the selection process. One special case is for the transition rates (r_2 and r_5) , where the rates are known to be higher than the transversion rates. Therefore in this case we use a more stringent threshold, $D_{25} > 2\delta$.

Nature Methods: doi:10.1038/nmeth.2109

6. Check whether $\frac{\min(\pi_1,\pi_2,\pi_3,\pi_4)}{\max(\pi_1,\pi_2,\pi_3,\pi_4)} < (1-\delta)$
7. In this case, all models with equal frequencies are ignored.
8. Define a gamma shape threshold $\alpha_{min} \in \Re$ (α_m should be big enough, e.g., $\alpha_m = 50$).
9. If $\alpha < \alpha_{min}$ filter out all +G and +I+G models from the candidate set
10. Let $\rho_{min} \in \Re$ and $\alpha_{max} \in \Re$, where ρ_{min} is the minimum ρ and α_{max} the maximum α. Then +I models will be excluded if $\rho < \rho_{min}$ and $\alpha > \alpha_{max}$. Note that α_{max} is not expected to be as big as α_{min}, but the higher it is, the less probably is to exclude the best-fit model.

In addition, our empirical analysis also showed that the effectiveness of this heuristic depends on the 'complexity' of the input data, which is reflected in the likelihood score. For simpler data sets the likelihood is smaller, and the number of parameters become more decisive, favoring the selection of simpler models. However, in this case the reduction of the candidate models set is also less important since the execution times will also be smaller. **Fig. S2** shows the heuristic accuracy as a function of the likelihood score across 4,000 alignments sampled from the 10,000 simulated.

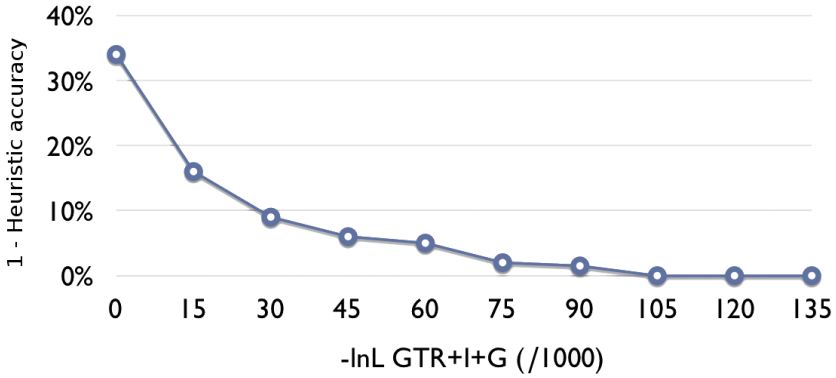

Figure S2 | Likelihood score and heuristic performance. The figure shows the percentage of best-fit models identified during the exhaustive search wrongly filtered out (1 - heuristic accuracy) from the candidate model set in function of the likelihood of the GTR+I+G model for a fixed filtering threshold.

Since it is difficult for the user to estimate *a priori* the likelihood of the models, we aimed to find some kind of general "threshold tuning" that depends on the likelihood score of the GTR+I+G model and guarantees a similar trade-off between the accuracy of the heuristic and the computational savings to that depicted in **Fig. 1** independently of the particular data set. Using logarithmic interpolation we arrived to the following function:

$$f(x) = t\frac{-5\ln(x-1)+6\ln(151)-1\ln(1)}{\ln(151)-\ln(1)}$$

where t is the user-defined filtering threshold and x is the -lnL of the GTR+I+G model for the specific data set divided by 1000.

Nature Methods: doi:10.1038/nmeth.2109

Supplementary Note 3 | Simulations from prior distributions. We simulated 10,000 nucleotide sequence alignments with 40 sequences and 2500 bp each. In order to consider a variety of simulation scenarios we first sampled model parameters and random trees from different statistical distributions using R[8]. Then, we used Seq-Gen[9] to simulate the DNA sequences accordingly and analyzed them with jModelTest 2. We considered 4 model families: without rate variation (M), with a proportion of invariable sites (M+I), with gamma rate variation among sites (+G), and with both a proportion of invariable sites and gamma rate variation among sites (+I+G). The simulation pipeline was repeated 2,500 times for each model family in turn:

1. Select at random one of the 22 possible models implemented in jModelTest 2 for the given family according to a Uniform distribution U(0,21).

2. Assign parameter values according to the model predefined structure:
    2.1. The base frequencies (ACGT) are set to 0.25 or sampled from a Dirichlet distribution D(1.0,1.0,1.0,1.0).
    2.2. The transition/Transversion rate comes from a Gamma distribution G(2,1) truncated between 2 and 10
    2.3. The R-matrix parameters are sampled from a Dirichlet distribution D(6,16,2,8,20,4) scaled with the last rate.
    2.4. The gamma shape for rate variation among sites comes from an Exponential distribution E(2) truncated between 0.5 and 5.
    2.5. The proportion of invariable sites is sampled from a Beta distribution B(1,3) truncated between 0.2 and 0.8.

3. Generate a random non-ultrametric rooted tree:
    3.1. We used the function *rtree* from the ape R package[10] (this works splitting randomly the edges) with branches according to a Exponential distribution E(1,10).
    3.2. Total tree length was scaled so the tree length was uniformly distributed in the U[2, 12] interval.

4. Simulate a sequence alignment using the parameter values sampled and the simulated tree using SeqGen[9].

5. Analyze this dataset using jModelTest 2: select the best-fit model under the AIC, BIC and DT criteria and obtain model-averaged parameter estimates.

Nature Methods: doi:10.1038/nmeth.2109

Supplementary Note 4 | Speed-up benchmark on real and simulated datasets. We analyzed several real data sets in order to benchmark the speed-ups obtained with jModelTest 2:

Dataset	Organism	Genes	NumSeq	Length	Reference
A	HIV-1	polimerase	8	3009	http://www.hiv.lanl.gov/
B	HIV-1	whole genome	138	10693	http://www.hiv.lanl.gov/
C	Yeast	106 genes	8	127060	Rokas et al. (2003)
D	simulated	--	40	500	Guindon and Gascuel (2003)
E	simulated	--	100	500	Guindon and Gascuel (2003)

Datasets A and B are trimmed alignments initially downloaded from http://www.hiv.lanl.gov/. Dataset C was provided by Antonis Rokas[11]. Datasets D and E are two alignments already used for benchmarking Phyml_ENREF_11[12], and can be downloaded from http://www.atgc-montpellier.fr/phyml/datasets.php.

The threaded version of jModelTest 2 was executed on CESGA's SVG nodes with 2 AMD Opteron Processors 6174@2.2GHz (2x12 cores, hence 24 cores) and 32GB memory. The MPI-based version of jModelTest 2 was executed on 8 Xeon nodes with 2 Intel Xeon E5420@2.50GHz per node (2x4 cores, hence 8 cores per node) and 16GB memory. These nodes are interconnected via 10 Gigabit Ethernet. The hybrid multithread/MPI-based implementation was executed in a public cloud infrastructure, in 32 Amazon EC2 cluster compute instances (23 GB memory, 33.5 EC2 Compute Units and Linux OS), with two Intel Xeon X5570 quad-core Nehalem processors each instance, hence 8 cores per virtual machine. These systems are interconnected via 10 Gigabit Ethernet, which is the differential characteristic of this resource. In fact, this EC2 instance type has been specifically designed for HPC applications and other demanding latency-bound applications. According to Amazon one EC2 Compute Unit provides the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

In the shared memory architecture with 24 cores, the scalability of the multithreaded implementation was almost linear with up to 8 threads, but also scaled well with 24 threads (**Fig. S3a**). In a cluster –distributed memory– the MPI-based application scaled well up to 32 processes, especially for the largest data sets (**Fig. S3b**). Here, the fact that some models can be optimized much more faster than others –especially when they do not include rate variation among sites–, posed a theoretical limit to the scalability. This problem was circumvented when we implemented a hybrid multithread/MPI-based approach –shared and distributed memory–, executed on Amazon EC2 cloud, which resulted in speedups of 182-211 with 256 processes even for the most complex cases (**Fig. S3c**). For relatively large alignments here (e.g., 138 sequences and 10,693 sites) this could be equivalent to a reduction of the running time from near 8 days to around 1 hour.

Nature Methods: doi:10.1038/nmeth.2109

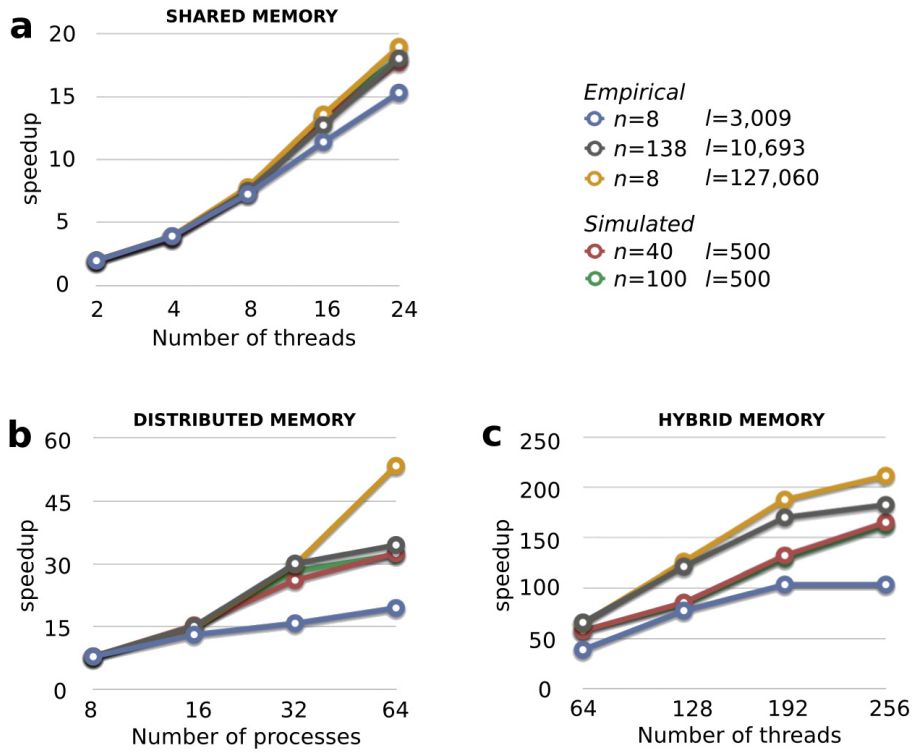

Figure S3. Scalability of jModelTest 2 with real and simulated data. The x-axis represents the number of parallel processes used in executions, and the y-axis represents the speedup regarding the sequential execution. *n* is the number of taxa and *l* is the alignment length.

In addition, we ran additional simulated datasets with up to 100 taxa and 10,000 sites on a different testbed. With shared memory (**Fig. S4a**), jModelTest 2 showed an almost linear speedup up to 8 threads (1 per core). When enabling hyperthreading (12 or 16 threads on 8 physical cores) the scaling-up was less pronounced. With distributed memory (**Fig. S4b**), the scalability was even better, reaching some saturation with 64 processes, mainly due to the serialized execution of each model optimization and the workload imbalance between models. As the 22 +G models represent around 80% of the total execution time, it is expected a theoretical limit 40X speedup in most cases. The hybrid memory version brings a more fine grain parallelism, and therefore overcomes the previous scalability limit (**Fig. S4c**). Those tests with higher computational load reached up to 130X speedup, while the lightest ones showed reduced efficiency due to the low computational load per process, making the parallel overhead (i.e., the cost of communications and synchronization) larger in relative terms. In this experiment, shared memory speedups were obtained in an 8-core node with Hyper-Threading technology (i.e., running up to 16 threads on 8 physical cores). The distributed memory version was executed on 8 Xeon nodes with 2 Intel Xeon E5420@2.50GHz per node (2 × 4 cores, hence 8 cores per node) and 16GB memory. These nodes are interconnected via 10 Gigabit Ethernet. The Hybrid memory version was executed on 32 nodes with the same features.

Nature Methods: doi:10.1038/nmeth.2109

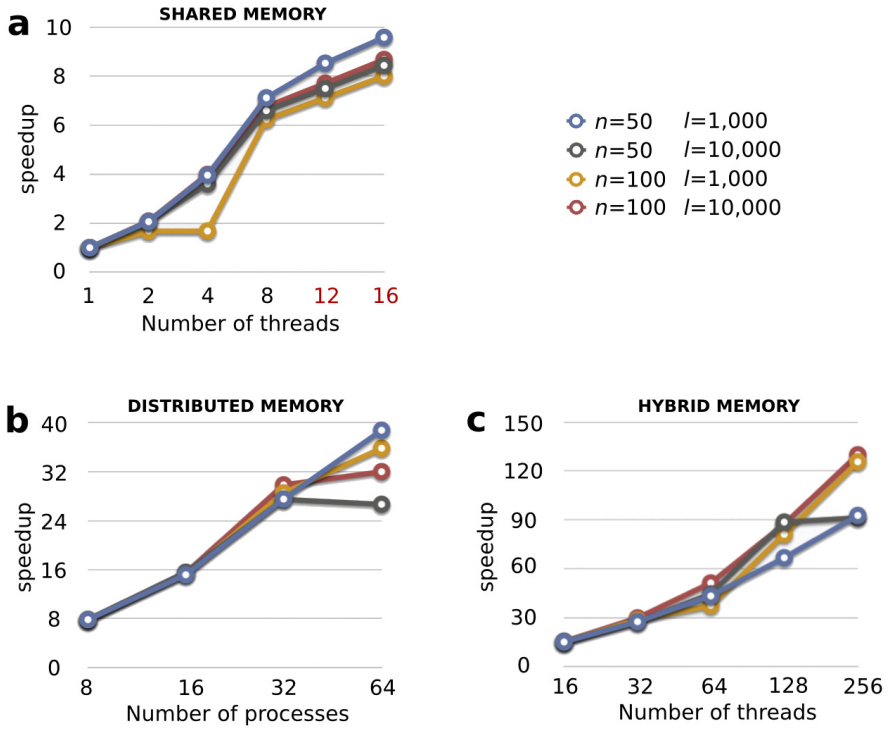

Figure S4. Scalability of jModelTest with simulated data. The speedups reported are for (a) Shared memory (red numbers in the x-axis indicate hyperthreading), (b) Distributed memory and (c) Hybrid memory version. *n* is number of taxa and *l* is the alignment length.

Nature Methods: doi:10.1038/nmeth.2109

References

1. Tavaré, S., in *Some mathematical questions in biology - DNA sequence analysis*, edited by R. M. Miura (American Mathematical Society, Providence, RI, 1986), Vol. 17, pp. 57-86.
2. Goldman, N., *J. Mol. Evol.* **36**, 182-198 (1993).
3. Robinson, D. F. & Foulds, L. R., *Math. Biosci.* **53**, 131-147 (1981).
4. Jordan, G. E. & Piel, W. H., *Bioinformatics* **24**, 1641-1642 (2008).
5. Jukes, T. H. & Cantor, C. R., in *Mammalian Protein Metabolism*, edited by H. M. Munro (Academic Press, New York, NY, 1969), pp. 21-132.
6. Zharkikh, A., *J. Mol. Evol.* **39**, 315-329 (1994).
7. Lanfear, R., Calcott, B., Ho, S. Y. & Guindon, S., *Mol. Biol. Evol.* **29**, 1695-1701 (2012).
8. R Development Core Team, *R Foundation for Statistical Computing* (2008).
9. Rambaut, A. & Grassly, N. C., *Comput. Appl. Biosciences* **13**, 235-238 (1997).
10. Paradis, E., Claude, J. & Strimmer, K., *Bioinformatics* **20**, 289-290 (2004).
11. Rokas, A., Williams, B. L., King, N. & Carroll, S. B., *Nature* **425**, 798-804 (2003).
12. Guindon, S. & Gascuel, O., *Syst. Biol.* **52**, 696-704 (2003).

Nature Methods: doi:10.1038/nmeth.2109