Heuristic Algorithms for the Protein Model Assignment Problem

J. Hauser[1], K. Kobert[1], F. Izquierdo-Carrasco[1], K. Meusemann[3], B. Misof[3], M. Gertz[2], and A. Stamatakis[1]

[1] Heidelberg Institute for Theoretical Studies, Heidelberg, Germany
[2] Heidelberg University, Institute of Computer Science, Heidelberg, Germany
[3] Zentrum für molekulare Biodiversitätsforschung, Zoologisches Forschungsmuseum Alexander Koenig, Bonn, Germany

Abstract. Assigning an optimal combination of empirical amino acid substitution models (e.g., WAG, LG, MTART) to partitioned multi-gene datasets when branch lengths across partitions are linked, is suspected to be an NP-hard problem. Given p partitions and the approximately 20 empirical protein models that are available, one needs to compute the log likelihood score of 20^p possible model-to-partition assignments for obtaining the optimal assignment.

Initially, we show that protein model assignment (PMA) matters for empirical datasets in the sense that different (optimal versus suboptimal) PMAs can yield distinct final tree topologies when tree searches are conducted using RAxML.

In addition, we introduce and test several heuristics for finding near-optimal PMAs and present generally applicable techniques for reducing the execution times of these heuristics. We show that our heuristics can find PMAs with better log likelihood scores on a fixed, reasonable tree topology than the naïve approach to the PMA, which ignores the fact that branch lengths are linked across partitions. By re-analyzing a large empirical dataset, we show that phylogenies inferred under a PMA calculated by our heuristics have a different topology than trees inferred under a naïvely calculated PMA; these differences also induce distinct biological conclusions. The heuristics have been implemented and are available in a proof-of-concept version of RAxML.

Keywords: phylogenetic inference, maximum likelihood, model assignment, protein data

1 Introduction

An important task in phylogenetics consists in computing the (maximum) likelihood score on a given tree topology. Typically, the logarithm of the likelihood is computed for numerical reasons. Throughout the paper, we use likelihood and log likelihood as synonyms. The likelihood score represents the probability of observing the data (a set of aligned molecular sequences), given a strictly bifurcating unrooted tree. A statistical model of evolution is required to specify how the observed data (e.g., an alignment of amino acid sequences) was generated by the given topology, that is, the model provides transition rates between possible states (e.g., amino acid characters).

For DNA data, a general time reversible substitution model [1] is typically being used, which requires a direct maximum likelihood estimate of the transition rates. For amino acid data, this is mostly not considered, because it may result in over-parametrizing the model (DNA has 5 rates, protein data has 189 transition rates). Therefore, a plethora of empirical protein substitution models such as MTART [2], WAG [3], and LG [4], have been derived from large collections of real-world protein alignments. Some of these models are intended for general use (e.g., WAG and LG) and some have been optimized for specific organisms (e.g., the MTART model for *Arthropoda*).

Selecting an appropriate empirical protein substitution model for the data at hand represents an important and generally non-trivial task. This is because using an inappropriate model that does not fit the data well, can lead to erroneous phylogenetic estimates (see, e.g., [5] or [6]).

Here, we consider the case of protein model assignment for partitioned (different sets of sites evolve under distinct evolutionary models) multi-gene amino acid sequence alignments. Note that, determining an appropriate partitioning scheme is also a non-trivial problem (e.g., [7]) but outside the scope of this paper. Therefore, we assume that an appropriate partitioning scheme is given. We denote this task as *protein model assignment (PMA) problem*. Given a fixed, reasonable (i.e., non-random) tree we want to assign the best-fit empirical protein substitution model to each partition such that the overall likelihood is maximized. Note that, using the optimal (with respect to the likelihood score) PMA does not increase the number of parameters in the model. Hence, over-fitting the data is not an issue and we can directly obtain the optimal PMA by finding the assignment that maximizes the likelihood. However, finding the optimal PMA is challenging if we assume that branch lengths are shared across partitions, that is, partitions are linked via a joint branch length estimate.

Using a joint branch length estimate across partitions is important because it drastically reduces the number of free parameters in the model. The number of inner branches in a strictly binary unrooted tree is $2n - 3$, where n is the number of taxa. Thus, each set of independent branch lengths that is estimated increases the number of model parameters by $2n - 3$. Therefore, joint branch length estimates can be deployed to avoid over-parametrizing the model.

Simply calculating the maximum likelihood score for all possible PMAs on a fixed, reasonable (i.e., non-random) tree, for p partitions and the approximately 20 available protein substitution models, is computationally prohibitive because of the exponential number (20^p) of possible assignments. We have already developed a proof (preprint available at http://www.exelixis-lab.org/Exelixis-RRDR-2012-9.pdf) that shows that the PMA problem is NP-hard. Here, we introduce and evaluate three heuristic strategies for computing 'good' PMAs for partitioned protein alignments under joint branch length estimates.

For small problem instances with $p := 3$ partitions (extracted from publicly available real datasets [8] and [9]) we observed substantial differences in final RAxML-based tree topologies inferred under the optimal PMA obtained from the exhaustive algorithm and suboptimal PMAs obtained via a naïve approach that is currently being used for determining the PMA. On simulated datasets, which generally tend to exhibit stronger signal (see, e.g., [10]), we did not observe that the PMA has an impact on final tree topologies, presumably because simulated data tend to be 'too perfect'. As we show here, finding a 'good' PMA is important for empirical analyses of real biological data because it changes the results, that is, the final tree topologies. Our heuristic PMA search strategies consistently find better PMAs, with respect to the likelihood score (without increasing the number of parameters in the model!) than the commonly used naïve heuristics that disregard the fact that partitions are linked via the branch lengths.

The remainder of this paper is organized as follows: In Section 2 we briefly review related work on the general problem of protein model selection. In Section 3 we introduce our heuristics and computational shortcuts for reducing the computational burden of computing likelihood scores for candidate PMAs. In Section 4 we discuss the experimental setup and provide experimental results. We conclude in Section 5 and discuss directions of future work.

2 Related Work

To the best of our knowledge, this paper and the paper addressing the NP-hardness proof are the first to identify and address the PMA problem.

Hence, we will briefly review work on the protein model selection methods in phylogenetics. There exists an extensive literature on methods for selecting models of nucleotide or amino acid substitution (see [11] for an in-depth review).

Initially, model testing pipelines applied likelihood ratio tests for selecting the best fit model. However, these tests require the models to be nested, which is not always the case. Therefore, tests relying on the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC), that do not require the models to be nested, have recently gained momentum

One of the most widely used tools for selecting protein models is ProtTest [12]. Another, fairly similar tool, for protein model selection is Aminosan [13].

As stated above, none of the existing pipelines address the PMA problem. Keep in mind that, PMA is essentially not a model selection problem, but an optimization problem because the number of model parameters is constant for all 20^p possible PMAs. As such, computing a 'good' PMA (finding the optimal PMA is NP-hard!) for partitions that are linked via a joint branch length estimate forms part of the general model selection process that is implemented by the above tools.

3 Heuristics

For all heuristics described here, we assume that a reasonable (i.e., non-random) tree is given. Such a tree can be obtained by executing a neighbor joining or parsimony tree search. It is broadly accepted that using a fixed, parsimony or neighbor joining tree for estimating model parameters is sufficient to obtain accurate parameter estimates [14]. Hence, given such a reasonable fixed tree *and* a data partitioning scheme with p data partitions, our goal is to find the PMA that maximizes the likelihood. This PMA can then be used for a subsequent maximum likelihood (ML) tree search using, for instance, RAxML.

Initially, we briefly describe the *naïve heuristics* that represent a simple and straight-forward approach to obtain a somewhat reasonable PMA. The naïve heuristics simply ignore the fact that partitions are linked via branch lengths and determine the best-scoring protein substitution model independently for each partition (by looping over the protein models) using a per-partition branch length estimate. The PMA obtained by this naïve approach can be used as initial seed for the search algorithms presented in Sections 3.3 and 3.4 to accelerate convergence.

If the number of partitions is small (e.g., $p := 3$) one can also perform an *exhaustive search* by computing the maximum likelihood scores for all possible $20^3 = 8000$ PMAs to obtain the global maximum, that is, the exact solution.

In our heuristics, we want to explore an as large as possible fraction of the search space by evaluating as many candidate PMAs as possible. However, computing the likelihood on candidate PMAs is expensive, because model parameters such as the α shape parameter of the Γ model of rate heterogeneity [15] and the joint branch lengths need to be re-optimized for each PMA. Therefore, we initially discuss some general computational shortcuts to reduce the computational cost of calculating likelihood scores for candidate PMAs.

3.1 Accelerating the Evaluation of Candidate PMAs

In the course of the searches we need to compute the maximum likelihood score for a large number of candidate PMAs. This entails fully re-optimizing all model parameters such as the branch lengths and the α shape parameter for each new PMA from scratch, that is, from some initial default values for α and the branch lengths. These parameters are optimized via standard numerical optimization procedures such as Brent's algorithm (α) and the Newton-Raphson procedure (branch lengths). Instead of re-optimizing all parameters from scratch, we can, re-use the parameter values of the current PMA i as initial values for optimizing the parameters and scoring a new PMA $i + 1$. This will generally be faster, because the parameter estimates (especially the α parameter) for assignment i will not differ substantially from those of assignment $i + 1$. The differences in model parameter estimates between PMAs i and $i + 1$ are also small because in the heuristics presented below, we only change the protein model of one partition at a time to obtain PMA $i + 1$ from PMA i. Hence, the numerical optimization routines will require less iterations to converge because the initial parameter values are 'good'. In our tests, this modification only yielded minimal deviations in likelihood scores (less than 0.5 log likelihood units) while improving execution times by a factor of 2.8 on average (see [16] for details, available at http://www.exelixis-lab.org/pubs/JoergHauserMasterThesis.pdf).

The second approach to reducing execution times of candidate PMAs strives to avoid evaluating candidate PMAs that are not promising. In other words, given a PMA that needs to be scored by computing its maximum likelihood score, we deploy an inexpensive pre-scoring criterion to determine whether or not it is worth to evaluate this PMA. To pre-score PMAs we use the per-partition likelihood scores for each partition and each protein substitution model that can be computed using the naïve approach outlined above. These scores, albeit obtained under a per-partition branch length estimate, can be used to pre-score candidate PMAs because of a strong correlation between the overall (across all partitions) likelihood scores under a joint branch length estimate and the likelihood scores under a per-partition branch length estimate. In Figure 1 we depict the full (left y-axis) and approximate (right y-axis) likelihood scores for 100 random PMAs on a dataset with 50 partitions and 50 taxa that was subsampled from the real biological dataset [8] used in Section 4.

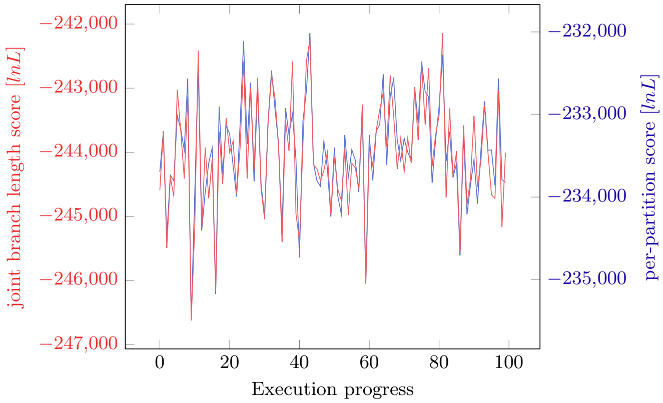

Fig. 1. Full likelihoods and approximate likelihoods for 100 random PMAs on a real biological dataset.

Because of this strong correlation, the per-partition likelihood scores as obtained under a per-partition branch length estimate can be used to omit the evaluation of candidate PMAs that do not appear to be promising. For details on computing the threshold for deciding which candidate PMA evaluations to skip, please refer to [16]. By using this technique we were able to accelerate the heuristics by a factor of 1.5 to 2.

3.2 Greedy Partition Addition Strategy

The greedy partition addition heuristics represent a constructive approach that gradually extends the alignment by adding one partition (and model) at a time. We start with the first partition and determine and fix the best protein substitution model for this partition. Then, we add the second partition and compute the likelihood scores for all 20 possible protein model assignments to this second partition while keeping the model for the first partition fixed. Once we have determined the best protein model for the second partition, we fix the model for the second partition as well. Thereafter, we add the third partition and compute the likelihood scores for all possible 20 model assignments to this third partition while keeping the models for the first and second partition fixed. Note that, the per-partition likelihood scores are re-computed for *all* partitions each time a new model is assigned to the new partition that is being added because the joint branch lengths are re-estimated for the entire alignment.

We continue extending the alignment (and PMA) in this way until all partitions have been added to the alignment. For this algorithm, we need to evaluate $p * 20$ candidate PMAs, where p is the number of partitions. Note that, the final PMA obtained by applying this strategy can be different depending on the order by which we add partitions. Therefore, we have implemented a fixed partition addition order by sorting the partitions by their length in terms of number of sites and adding them in descending order (longest partition first). We chose to optimize the model for the longest partition first because the longest partition typically contributes most to the overall likelihood score of the full alignment. However, this had no notable effect on performance of the heuristics with respect to the final likelihood scores of the best PMA that was found [16].

3.3 Steepest Ascent Strategy

The steepest ascent approach implements a classic neighborhood-based hill climbing strategy. Given some initial PMA, which can either be a random assignment, the result of the naïve heuristics, or the assignment computed by the greedy addition strategy (see above), we proceed as follows: We evaluate the likelihood scores of all PMAs that differ by one model-to-partition assignment from the current assignment. In other words, we explore a neighborhood of size 1. We need to calculate the likelihood scores of $(20 - 1) * p$ PMAs to explore the size 1 neighborhood of the current assignment (when not using the pre-scoring approach). Once all $19 * p$ scores have been calculated, we select the PMA that yields the largest likelihood improvement. We then explore the neighborhood of this new assignment. If there does not exist a PMA in the size 1 neighborhood that further improves the likelihood, we have reached a local optimum and the algorithm terminates.

3.4 Simulated Annealing Strategy

We also implemented a simulated annealing algorithm because of its ability to navigate out of local maxima [17].

We can initialize the PMA for the simulated annealing strategy either at random or with the result of the naïve heuristics. As for the steepest ascent algorithm, we explore the size 1 neighborhood of the current PMA. There are nonetheless some fundamental differences. We iteratively evaluate the neighboring assignments of the current PMA and compute their corresponding likelihood scores.

For each neighboring assignment that is evaluated, we carry out an acceptance/rejection step. Thus, if the likelihood of the candidate PMA is better than that of the current PMA, we accept it immediately and use it as current assignment (in analogy to a greedy hill climbing strategy). If the likelihood of the candidate PMA is worse than that of the current PMA we need to decide whether to accept a backward step or not. We accept a PMA that decreases the likelihood if $r < e^{-\frac{l-l'}{T_k}}$, where r is a uniform random number in $[0;1]$, l is the likelihood of the current assignment, and l' the likelihood of the candidate PMA. Finally, T_k is the temperature of the annealing process at iteration k (evaluation of the kth PMA). This procedure is also known as *Metropolis criterion* (see [18]). We implemented a standard cooling schedule $T_k = \lfloor T_0 \beta^k \rfloor$, where T_0 is the starting temperature and $\beta \in [0;1]$ represents a parameter that needs to be tuned. We empirically determined a setting of $\beta := 0.992$ (see [16] for details). The simulated annealing process terminates at iteration n when $T_n = 0$ *and* when a PMA is generated that has a worse likelihood score than the currently best one.

4 Performance Assessment

The modified RAxML code, the test datasets, as well as the wrapper scripts (greedy algorithm) are available at http://exelixis-lab.org/joerg/pma.tar.gz.

4.1 Experimental Setup

We implemented the three search strategies outlined in Sections 3.2 through 3.4 as well as the naïve and exhaustive search algorithms in the standard RAxML version [19] and via wrapper scripts. We also used RAxML to compute Robinson-Foulds distances [20] between trees. Computational experiments were performed on our institutional cluster, which is equipped with 50 48-core AMD Magny-Cours nodes (equipped with 128GB RAM each) and connected via Infiniband.

4.2 Test-Datasets

We used real (empirical) *and* simulated datasets to test (i) whether a good PMA 'matters' with respect to the final tree topology and (ii) to evaluate our heuristic search strategies. We used three partitioned real-world data sets from two studies [8, 21] that encompass data from all three domains of life. The properties of the datasets are summarized in Table 4.2.

We simulated datasets using INDELible [22] on random ’true’ tree shapes with 40 taxa that were generated via a R script provided by David Posada (included in the on-line data archive) and 2, 4, 8, 16, 32, 64, and 128 partitions, respectively. Partition lengths were randomly generated and ranged between 300 and 500 sites. Protein substitution models to simulate the data along the tree for each partition were also assigned at random.

Domain	# taxa	# partitions	length	reference
Eukaryotes	117	129	37,476	[8]
Bacteria	992	56	20,609	[21]
Archaea	86	68	17,639	[21]

4.3 Results

Does the PMA matter? Initially, we address the question whether obtaining a good PMA actually matters, that is, if it alters the final tree topology when applying a standard RAxML maximum likelihood tree search. For this purpose, we randomly sub-sampled 50 datasets containing three partitions and 50 taxa from each of the three real-world datasets listed in Table 4.2. We thereby generated a total of 150 small real-world test datasets. For each sub-sampled alignment we then computed a PMA using the naïve algorithm and the exhaustive algorithm to obtain the globally optimal PMA. Note that, running the exhaustive algorithm on more than 3 ($20^3 = 8000$ distinct possible PMAs) partitions was computationally not feasible. Model assignments differed for 86 out of the 150 alignments. Thus, the naïve approach yields suboptimal PMAs for more than half of the datasets. For those 86 datasets where the PMAs differed we executed 10 standard RAxML tree searches (staring from distinct randomized addition order parsimony trees) per dataset to obtain the best-known ML tree under the naïve and optimal PMA. We only obtained topologically identical ML trees for 14% of the 86 datasets. The average topological RF-distance between the trees inferred under the naïve and the optimal assignment was 9%. As expected the naïve PMA never yielded a final tree with a better likelihood than the optimal PMA. Hence, on real data, investing computational effort to finding a ’good’ PMA is important, because it has a noticable impact on the structure of the final tree topology.

We then also calculated the PMAs using the steepest ascent heuristics on these 150 datasets. The inferred PMAs differed from the optimal PMA obtained by the exhaustive algorithm for only 10 out of 150 datasets (7%). Furthermore, the best-known ML trees inferred on those 10 datasets showed an average RF-distance of only 3%. We conclude that, (i) the steepest ascent heuristics are able to infer the optimal PMA in the majority of the cases and (ii) when the heuristics yield a suboptimal PMA, the inferred PMA nonetheless induces a substantially smaller topological error than the naïve PMA.

For the simulated datasets, we inferred ML trees using a random PMA, the naïve assignment, and the known, true PMA under which the data was generated. Thereafter, we calculated the RF distances between the ML trees inferred using the random PMA and the true PMA as well as the RF-distance between the trees inferred under the naïve PMA and the true, known PMA. We found differences in RF distance to be negligible in both cases (random and naïve PMAs) on simulated data. We suspect that this is due to the fact that simulated data tends to be more perfect than real data [10].

The important finding is that determining a PMA that fits the data well has a substantial impact on real word data analyses.

Performance of Heuristics: To assess the relative performance and quality of the three heuristic strategies we propose, we sub-sampled 15 datasets containing 50 taxa and 50 partitions from each of the three real-world datasets. This was done to reduce the computational burden of these analyses.

We intend to determine which strategy performs best with respect to execution times and result quality which we quantify as the maximum likelihood score of the respective PMAs. Note that, the number of free model parameters is identical for all candidate PMAs, hence a likelihood-based comparison of PMAs is meaningful. As a reference, we used the likelihood score and the execution time required by the naïve heuristics. The simulated annealing and steepest ascent algorithms were seeded with the PMAs obtained from the naïve heuristics. These two search strategies were also seeded with a random seed, but performed worse (results not shown).

We summarize the results in Figure 2. The figure contains average execution times in seconds and average score improvements in *log* likelihood units over the 15 test datasets for the three PMA heuristics we propose. The execution times displayed for the simulated annealing and steepest ascent strategy include the execution time of the naïve algorithm whose assignment is used as a seed. For *all* 15 test datasets, we were able to find a PMA with a better likelihood than obtained via the naïve algorithm on the the same, fixed, reasonable tree topology. Overall, the steepest ascent algorithm performs best with respect to execution times *and* result quality.

Re-Analysis of a Biological Dataset: We inferred ML trees and bootstrap support values on the main empirical dataset used in [8] with (i) the PMA as used in the original study (WAG assigned to all partitions; denoted as `allWAG`) and (ii) the PMA as obtained from the steepest ascent heuristics (denoted as `optimized`). The relative RF distance between the resulting best-known ML trees was 8%. Hence, an optimized PMA can change the shape of final tree topologies as well as the biological conclusions which we discuss in the following.

The most conspicuous difference between the two trees is the position of the bristle tail (*Lepismachilis ysignata*, a wingless insect of the *Archaeognatha* insect order), which belongs to the primarily wingless *hexapods*. Insects in the *Archaeognatha* order are typically assumed to be a sister group (neighboring

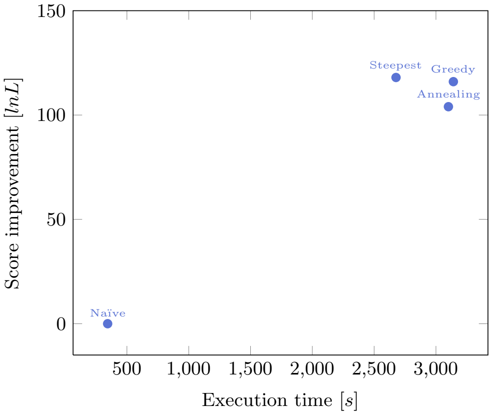

Fig. 2. Execution times in seconds of the three strategies and average improvement in terms of *log* likelihood units over the PMA obtained from the naïve approach

subtree) of the so-called *Dicondylia* that include all winged insects (the so-called *Pterygota*). Therefore, the phylogenetic position of the bristle tail within the winged insects in the `allWAG` analysis is rather implausible, since it also shows low bootstrap support. Moreover, its position in the `optimized` phylogeny received strong bootstrap support. Its phylogenetic position as a sister group of the winged insects as obtained from the `optimized` analysis has also been observed in prior studies based on molecular and morphological data [8, 23, 24].

Another notable difference is the placement of *Locusta migratoria* from the order *Orthoptera*. *Orthoptera* (grasshoppers, crickets, weta, and locusts) are commonly assumed to form a monophyletic clade (be located in a single, distinct subtree). Hence, the placement of *Locusta migratoria* is more plausible in the `allWAG` analysis in which *Orthoptera* are monophyletic. However, its position in the `optimized` tree only received moderate bootstrap support, such that it is difficult to draw conclusions regarding its placement based on the dataset at hand. Note that, the phylogenetic position of *Locusta migratoria* is generally considered difficult and hard-to-resolve [8]. The placement of *Locusta migratoria* highly depends on the dataset being used [25]. There is some evidence that *Locusta migratoria* might be a so-called rogue taxon [26].

Overall, from a biological perspective, the tree obtained via the `optimized` tree inference has to be favored. Furthermore, our re-analysis shows that biologically meaningful differences can be observed when inferring trees under an appropriately optimized PMA.

5 Conclusion and Future Work

We addressed the problem of assigning empirical protein substitution models to partitioned datasets that are analyzed under a joint branch length estimate across partitions. This paper is the first paper addressing this problem empirically. We show that obtaining a 'good' PMA (with respect to the likelihood score) matters on empirical datasets, because tree inferences under a naïve PMA can yield a topologically and biologically different phylogeny with worse likelihood scores than inferences under the optimal PMA. We specifically use the term 'good' PMA because finding the optimal PMA is NP-hard. While we can compute the globally optimal assignment for datasets with three partitions via an exhaustive search, finding a 'good' PMA on datasets with more partitions requires heuristic search strategies. We introduce, make available, and test three 'classic' search strategies for combinatorial optimization problems and adapt them to the problem at hand. We show that all three strategies can produce PMAs with better likelihood scores than the naïve search on all test data sets. Moreover, we presented two techniques for reducing the computational cost of our heuristics.

On a large biological dataset [8], we demonstrate that investing computational effort to optimize the PMA is important because it has an impact on the final tree topology as inferred with RAxML and on the biological interpretation of the tree.

We are currently integrating the steepest ascent strategy that performed best in our experiments into the standard RAxML version. Moreover, we also intend to parallelize the heuristic strategies by using a hybrid MPI/PThreads approach. In this setting, the evaluation of candidate PMAs can be distributed among MPI processes that conduct the likelihood calculations in parallel using the fine-grain PThreads parallelization of the phylogenetic likelihood function in RAxML.

References

1. Tavaré, S.: Some probabilistic and statistical problems in the analysis of DNA sequences. Some mathematical questions in biology-DNA sequence analysis **17** (1986) 57–86
2. Abascal, F., Posada, D., Zardoya, R.: Mtart: a new model of amino acid replacement for arthropoda. Mol. Biol. Evol. **24**(1) (2007) 1–5
3. Whelan, S., Goldman, N.: A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. Mol. Biol. Evol. **18**(5) (2001) 691–699
4. Le, S., Gascuel, O.: An improved general amino acid replacement matrix. Mol. Biol. Evol. **25**(7) (2008) 1307–1320
5. Sullivan, J., Swofford, D.: Are guinea pigs rodents? The importance of adequate models in molecular phylogenetics. J. Mamm. Evol. **4**(2) (1997) 77–86
6. Keane, T., Creevey, C., Pentony, M., Naughton, T., Mclnerney, J.: Assessment of methods for amino acid matrix selection and their use on empirical data shows that ad hoc assumptions for choice of matrix are not justified. BMC Evol. Biol. **6**(1) (2006) 29
7. Lanfear, R., Calcott, B., Ho, S., Guindon, S.: Partitionfinder: combined selection of partitioning schemes and substitution models for phylogenetic analyses. Mol. Biol. Evol. **29**(6) (2012) 1695–1701

8. Meusemann, K., von Reumont, B., Simon, S., Roeding, F., Strauss, S., Kück, P., Ebersberger, I., Walzl, M., Pass, G., Breuers, S., et al.: A phylogenomic approach to resolve the arthropod tree of life. Mol. Biology Evol. **27**(11) (2010) 2451–2464
9. Yutin, N., Puigbò, P., Koonin, E., Wolf, Y.: Phylogenomics of Prokaryotic Ribosomal Proteins. PloS ONE **7**(5) (2012)
10. Stamatakis, A., Ludwig, T., Meier, H.: RAxML-III: A Fast Program for Maximum Likelihood-based Inference of Large Phylogenetic Trees. Bioinformatics **21(4)** (2005) 456–463
11. Posada, D. In: Selection of Phylogenetic Models of Molecular Evolution. John Wiley & Sons, Ltd (2001)
12. Abascal, F., Zardoya, R., Posada, D.: Prottest: selection of best-fit models of protein evolution. Bioinformatics **21**(9) (2005) 2104–2105
13. TANABE, A.: Kakusan4 and aminosan: two programs for comparing nonpartitioned, proportional and separate models for combined molecular phylogenetic analyses of multilocus sequence data. Mol. Ecol. Resources **11**(5) (2011) 914–921
14. Yang, Z.: Among-site rate variation and its impact on phylogenetic analyses. Trends Ecol. & Evol. **11**(9) (1996) 367–372
15. Yang, Z.: Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites. J. Mol. Evol. **39** (1994) 306–314
16. Hauser, J.: Algorithms for Model Assignment in Multi-Gene Phylogenetics. Master's thesis, Ruprecht-Karls University Heidelberg (2012)
17. Kirkpatrick, S., Gelatt, C., Vecchi, M.: Optimization by simulated annealing. Science **220**(4598) (1983) 671
18. Aarts, E., Laarhoven, P.: Simulated annealing: an introduction. Stat. Neerland. **43**(1) (1989) 31–52
19. Stamatakis, A.: RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. Bioinformatics **22**(21) (2006) 2688–2690
20. Robinson, D., Foulds, L.: Comparison of phylogenetic trees. Math. Biosci. **53**(1-2) (1981) 131–147
21. Yutin, N., Puigbò, P., Koonin, E., Wolf, Y.: Phylogenomics of Prokaryotic Ribosomal Proteins. PloS ONE **7**(5) (2012) e36972
22. Fletcher, W., Yang, Z.: Indelible: a flexible simulator of biological sequence evolution. Mol. Biol. Evol. **26**(8) (2009) 1879–1888
23. Grimaldi, D.: 400 million years on six legs: On the origin and early evolution of Hexapoda. Arthropod Struct. & Dev. **39**(2) (2010) 191–203
24. Trautwein, M., Wiegmann, B., Beutel, R., Kjer, K., Yeates, D.: Advances in insect phylogeny at the dawn of the postgenomic era. Ann. R. Entomol. **57** (2012) 449–468
25. Letsch, H., Meusemann, K., Wipfler, B., Schütte, K., Beutel, R., Misof, B.: Insect phylogenomics: results, problems and the impact of matrix composition. Proc. Royal Soc. B **279**(1741) (2012) 3282–3290
26. von Reumont, B., Jenner, R., Wills, M., DellAmpio, E., Pass, G., Ebersberger, I., Meyer, B., Koenemann, S., Iliffe, T., Stamatakis, A., et al.: Pancrustacean phylogeny in the light of new phylogenomic data: support for Remipedia as the possible sister group of Hexapoda. Mol. Biol. Evol. **29**(3) (2012) 1031–1045