Phylogenetic Models of Rate Heterogeneity: A High Performance Computing Perspective

Alexandros Stamatakis

Institute of Computer Science, Foundation for Research and Technology-Hellas
P.O. Box 1385, Heraklion, Crete, GR-71110 Greece
stamatak@ics.forth.gr

Abstract

Inference of phylogenetic trees using the maximum likelihood (ML) method is NP-hard. Furthermore, the computation of the likelihood function for huge trees of more than 1,000 organisms is computationally intensive due to a large amount of floating point operations and high memory consumption. Within this context, the present paper compares two competing mathematical models that account for evolutionary rate heterogeneity: the Γ *and CAT models. The intention of this paper is to show that—from a purely empirical point of view—CAT can be used instead of* Γ*. The main advantage of CAT over* Γ *consists in significantly lower memory consumption and faster inference times. An experimental study using RAxML has been performed on 19 real-world datasets comprising 73 up to 1,663 DNA sequences. Results show that CAT is on average 5.5 times faster than* Γ *and—surprisingly enough—also yields trees with slightly superior* **Γ likelihood values**. *The usage of the CAT model decreases the amount of average L2 and L3 cache misses by factor 8.55.*

1. Introduction

Phylogenetic trees are used to represent the evolutionary history of a set of n organisms (also called taxa). A multiple alignment of a small region of their DNA or protein sequences can be used as input for the computation of phylogenies. In a computational context phylogenetic trees are usually strictly bifurcating unrooted trees. The organisms of the alignment are located at the tips and the inner nodes represent extinct common ancestors. The branches of the tree represent the time which was required for the mutation of one species into another—new—one. The inference of phylogenies with computational methods has many important applications in medical and biological research, such as e.g. drug discovery and conservation biology (see [1] for a summary). Due to the rapid growth of available sequence data and the constant improvement of multiple alignment methods it has now become feasible to compute large trees which comprise more than 1,000 organisms. The computation of the tree-of-life containing representatives of all living beings on earth is one of the *grand challenges* in Bioinformatics.

The fundamental algorithmic problem computational phylogeny faces consists in the immense amount of potential tree topologies. This number grows exponentially with the number of sequences n, e.g. for $n = 50$ organisms there already exist $2.84 * 10^{76}$ alternative topologies; a number almost as large as the number of atoms in the universe ($\approx 10^{80}$). In fact, it has already been demonstrated that finding the optimal tree under the *maximum likelihood* (ML) criterion is NP-hard [4]. However, despite the algorithmic complexity and the high computational cost of the ML function, significant progress has been achieved with the release of fast and accurate sequential and parallel programs such as e.g. PHYML [8], IQPNNI [16], MetaPIGA [13], TreeFinder [11], GAML [2], TREE-PUZZLE [22] and RAxML [23]. Typically, these programs allow for inference of 1,000 taxon trees on a single CPU in reasonable times. Since the main focus of program development has rightfully been on search algorithms, technical issues such as memory efficiency and manual optimization of the source code have been neglected. Despite the distinct statistical approach, programs for Bayesian inference such as MrBayes [10] heavily rely on the frequent evaluation of the likelihood function and therefore share the same technical problems.

Therefore, a *paradigm shift* towards technical implementation issues in ML program development is required to enable inference of larger phylogenetic trees in the future. For example, a recent critical review of the RAxML source code from a technical perspective lead to some simple technical optimizations which yield run time improvements of factor 1.66 on 1,000 taxa up to 67 on 25,000 taxa. For the 25,000 taxon case RAxML already required 2GB of main memory under the simple HKY85 [9] model of nucleotide substitution *without* rate heterogeneity. Thus, if the Γ model with quadruple memory requirements was to be used on this large alignment, such an analysis would not be feasible. Within the context of continuous data accumulation, increasing memory shortage *and* the growing discrepancy between CPU and memory access speeds the present paper intends to analyze whether the Γ model of rate heterogeneity can be replaced by the significantly faster and less memory-intensive CAT model from a purely empirical perspective. The basic objective is to determine if comparable trees—with respect to their **Γ likelihood score** and topological similarity—can be obtained by using CAT on typical real world biological data. **It can not be emphasized enough, that all final tree topologies in this paper, whether inferred under Γ or CAT, are compared on the basis of their Γ likelihood values to ensure a fair comparison using the objective function and to account for the preferred model in the phylogenetics community which currently is the Γ model.** The basic question asked in this paper is if a significantly faster search under the CAT model can yield final tree topologies with similar Γ likelihood values as a search under Γ. This is then placed into a high performance computing perspective, since CAT appears to be the only technically feasible solution (duo to reduced memory requirements) to incorporate rate heterogeneity into analyses of huge trees.

The remainder of this paper is organized as follows: In Section 2 the computationally relevant mathematical background of the Γ and CAT models for ML-based tree inference is outlined. In addition, the relatively few existing alternative implementations (except that of RAxML) of the CAT model are mentioned. Section 3 describes the algorithm which is used in RAxML to optimize and categorize the per-site individual rates. Furthermore, a dedicated algorithm for the refinement under Γ (R_Γ algorithm) is briefly described which allows for additional refinement of final trees obtained with the CAT model under the Γ model. The results of the large experimental study on 19 real-world alignments and a performance comparison with the popular IQPNNI and PHYML programs is provided in Section 4. Moreover, the memory efficiency of the competing models is analyzed in terms of L2 and L3 cache misses. Section 5 provides a conclusion and addresses issues of current and future work.

2. Models of Rate Heterogeneity

As already mentioned there exist two basic methods to account for rate heterogeneity among sites (alignment columns): individual per-site evolutionary rates and the Γ model [26] of rate heterogeneity. The main reason that the model of per-site evolutionary sites is not often used, is based mainly on statistical concerns that optimizing the evolutionary rate for each individual site might lead to over-parameterizing and thus over-fitting the data [26]. However, the effect of overestimation can be alleviated by using a fixed number $c \ll m$ of rate categories (CAT model), where m is the number of distinct patterns (columns) in the alignment. In this case every individually optimized evolutionary rate r_i where $i = 1, ..., m$ has to be mapped to one of the rate categories ρ_j where $j = 0, ..., c-1$. In addition, the individual values of the ρ_j have initially to be determined by taking into account the distribution of r_i values. This categorization—hence the name CAT—of c rate categories also has a computational advantage over using m individual rates: A relatively large number of exponentials has to be computed *per individual rate* in order to obtain the respective transition probability matrix. The number of invocations of the compute-intensive `exp()` function is reduced to a constant c which typically ranges between 25–100 in comparison to the full alignment length m which typically ranges from 500–2,000 (see Table 1). To the best of the author's knowledge, models of per-site evolutionary rates are currently only implemented in RAxML, IQPNNI, PHYLIP [7], and fastDNAml [17].

However, in IQPNNI the per-site rates are not categorized and only the likelihood of the intermediate and final trees is evaluated using this model. This means that the actual tree search is conducted using the plain model without rate heterogeneity (B.Q. Minh, personal communication). Therefore, IQPNNI has only been executed using the Γ model of rate heterogeneity for the performance comparison in Table 2. It is worth noting though, that individual rates r_i in IQPNNI are optimized using a different approach [15] than in RAxML.

The approach implemented in fastDNAml and PHYLIP is very similar to the CAT model in RAxML. The main difference is that in fastDNAml the number of individual rate categories is fixed to a maximum of 36. and that a separate program (DNArates [18]) is required to compute and categorize the evolution-

ary rates based on a *fixed* tree. Thus, in contrast to RAxML, fastDNAml does not offer the possibility to re-adapt the parameters of the CAT model to a changing topology during the inference process. Unfortunately, the process of rate optimization and categorization in DNArates has never been published.

The need to account for rate heterogeneity in ML-based analyses is broadly accepted in the community. Moreover, Biologists often have to incorporate the Γ model into their studies in order obtain publishable results because there also exists strong biological evidence for rate variation among sites. It has been demonstrated [26], that ML inference under the assumption of rate homogeneity can lead to erroneous results if rates vary among sites. The intention of this paper is not to argue against Γ in a statistical sense but to empirically determine if CAT can be used as a vehicle to circumvent the complexity of Γ.

CAT versus Γ: The HPC Perspective: The goal of this paragraph is to provide a notion of the amount of memory space and arithmetic operations required to compute the ML score for a tree topology under CAT and Γ. The seminal paper by Felsenstein [6] and the chapter by Swofford *et al.* [24] provide detailed descriptions of the mathematical background. Within the HPC context the focus is on the memory space and amount of floating point operations required by CAT and Γ. In most ML implementations the execution time is largely dominated by two basic operations: the computation of the likelihood vectors (also called partial likelihoods arrays) and the optimization of branch lengths. Those operations typically require $\geq 90\%$ of execution time (e.g. for a dataset with 150 sequences: 92.72% of total execution time in PHYML and 92.89% in RAxML-VI). Thus, an acceleration of these functions on a technical level is crucial. The number n (n: number of taxa) and the length of the likelihood vectors m (m: number of distinct patterns/columns in the alignment), dominate the memory consumption of typical ML implementations. Thus, the overall memory consumption is of $O(n * m)$.

After a change of the tree topology the likelihood of the tree is computed by filling in the likelihood vectors affected by the change bottom-up. To understand how the individual likelihood vectors are updated with given branch lengths `bq`, `br` consider a subtree rooted at node `p` with immediate descendants `r` and `q` and likelihood vectors `p[]`, `q[]`, and `r[]` respectively. When the likelihood vectors `q[]` and `r[]` have been computed the entries of `p[]` can be calculated—in an extremely simplified manner—as outlined by the pseudo-code below:

```
for(i = 0; i < m; i++)
   p[i]=f(g(q[i],bq),g(r[i], br));
```

where `f()` is a simple function, i.e. requires just a few FLOPs, to combine the values of `g(q[i], bq)` and `g(r[i], br)`. The `g()` function however is more computationally intensive since it contains the evaluation of the transition probabilities. The parameters `bq` and `br` represent the branch lengths. The above pseudocode represents the main computational load of a typical ML implementation. Note that, the optimization of branch lengths is very similar with respect to the loop structure and therefore omitted at this point.

Accommodating CAT: Given the optimized rates per site and given the rate categorization (see Section 3) one can easily modify the basic `for`-loop to accommodate the CAT model. At each iteration of the loop the evolutionary rate `r` of the current position `i` has to be determined. This is performed by looking up the respective rate category of position `i` in `category[]`. Note that, $0 \leq$ `cat` $< c$ where c is the maximum number of rate categories specified by the user (default in RAxML-VI is 50 rate categories). The `rate[]` vector is pre-computed before the main `for`-loop mainly in order to avoid redundant invocations of the `exp()` function. In order to account for distinct per-site rates the `g()` function now also depends on the rate category of the site. The pseudocode for updating the likelihood vectors with rate categories is indicated below:

```
for(i = 0; i < m; i++)
  {
   cat = category[i];
   r = rate[cat];
   p[i] = f(g(q[i],bq,r), g(r[i],br,r));
  }
```

As the above pseudo-code clearly shows, the additional computational effort required by the CAT model consists in the pre-computation of the `rate[]` vector (which is not shown here) and accessing the `category[]` and `rate[]` arrays in the main `for`-loop. The additional memory required is an array of c double values for `rate[]` and an array of m integer values for `category[]`.

Accommodating Γ: The computationally more complex form of dealing with heterogeneous rates, due to the fact that significantly more memory *and* floating point operations are required (typically factor 4), consists in using either discrete or continuous stochastic models for the rate distribution *at each* site. In this case every site has a certain probability of evolving at

any rate contained in a given probability distribution. For example a concrete distribution of the likelihood for one site is obtained by summing over all products of likelihoods for the discrete rates times the probability from the distribution. In the continuous case likelihoods must be integrated over the entire probability distribution.

The most common and most broadly used distribution types are the continuous [25] and discrete [26] Γ distribution. The actual form of the Γ function is determined by the α shape parameter which is optimized based on the likelihood. Small α values ($\alpha = 0.1$) stand for high rate heterogeneity and large values ($\alpha = 5.0$) for low rate heterogeneity. For computational reasons, in most ML programs the default is to use a Γ distribution with 4 discrete rates. This represents an acceptable trade–off between inference time, memory consumption and accuracy. The implementation of the Γ model in RAxML also uses 4 discrete rates which in addition have been hard-coded in the main `for`-loops and manually optimized to ensure a fair comparison with the highly optimized implementation of the CAT model.

Given the *four* individual rates from the discrete Γ distribution `r0,...,r3` *now four* individual likelihood entries `p[i].g0,...,p[i].g3` at each site `i` have to be updated as indicated below:

```
for(i = 0; i < m; i++)
  {
    p[i].g0 = f(g(q[i], bq, r0), g(r[i], br, r0));
    p[i].g1 = f(g(q[i], bq, r1), g(r[i], br, r1));
    p[i].g2 = f(g(q[i], bq, r2), g(r[i], br, r2));
    p[i].g3 = f(g(q[i], bq, r3), g(r[i], br, r3));
  }
```

The above pseudocode clearly shows why using Γ is almost prohibitive for computing huge trees: Per iteration of the `for`-loop it requires almost the quadruple number of floating point operations. Since memory shortage currently represents one of the main obstacles for inference of larger phylogenies the quadruple memory space required by Γ is even more problematic. Therefore, with respect to the HPC perspective and inference of huge trees, the CAT model should, and in the 25,000 taxon case must, be preferred over the Γ model due to the significantly inferior inference times and memory consumption.

3. Algorithms

Rate Category Optimization & Classification: The per-site evolutionary rates of a tree in RAxML are obtained by maximizing their individual per-site likelihood values. The lowest possible rate has been limited to 0.0001 to avoid numerical problems. The largest possible rate is arbitrary. The per-site rates are optimized using a Brent-like iterative method [19]. Provided the optimized individual rates $r_i, i = 1, ..., m$ they have to be mapped to the respective rate categories $\rho_j, j = 0, ..., c-1$. RAxML uses a very simple procedure to categorize rates which is mainly based on the per-site likelihood contributions $l(r_i)$. Initially, all individual rates $r_i, r_j, i \neq j$ for which $abs(r_i - r_j) < 0.001$ are stored in an intermediate set of rate categories $\sigma_k, k \leq m$ and their partial likelihoods are summed up and stored in $l(\sigma_k)$, e.g. $l(\sigma_k) := l(r_i) + l(r_j)$ if $abs(r_i - r_j) < 0.001$. Thereafter, the list of σ_k is sorted with respect to the partial likelihood values $l(\sigma_k)$ in descending order such that those rates σ_k which contribute most the overall likelihood are at the front of the sorted list. The first c entries of the sorted σ_k list will become the rate categories ρ_j for the inference process, i.e. $\rho_0 := \sigma_0, ..., \rho_{c-1} := \sigma_{c-1}$. Finally, an individual site i is then assigned the rate category $k \leq c$ which minimizes $abs(\rho_k - r_i)$. It is important to note, that the ϵ parameter for the optimization of per-site evolutionary rates is lowered progressively. It is lowered at each invocation of the rate category optimization function, which is executed after each iteration of the RAxML tree search algorithm.

The R_Γ Algorithm: The R_Γ (read Refinement under Γ) optimization option is a modified version of the standard hill-climbing search algorithm of RAxML as described in [23]. R_Γ optimization can be invoked by specifying `-f h` in the command line of RAxML. The purpose of this search algorithm is to further optimize trees under Γ that have initially been computed under CAT. For example, initial trees can be inferred using CAT and then further be refined using Γ with the R_Γ search algorithm. One main difference with respect to the standard search algorithm is that the maximum rearrangement setting is limited to a distance of 5. In addition, the analytical optimization of branch lengths (for details see [23]) which is mainly intended for the initial optimization phase where many improved trees are encountered, is omitted. Thus, the lengths of branches adjacent to the insertion points of subtrees are optimized thoroughly during the entire optimization process. The limitation of the rearrangement distance and the choice to optimize branches thoroughly are due to the assumption that refinement under a distinct, though related, model of substitution will not result in dramatic changes of the tree topology. This is of course a very questionable and perhaps dangerous assumption, but works well in the concrete case: a refinement of trees inferred with CAT under Γ (see

next Section). The combined inference process of trees under CAT with a subsequent refinement using R_Γ will henceforth be called CAT+R_Γ.

4. Experimental Setup & Results

Test Data: In order to conduct experiments, a relatively large number of 19 real-world alignments from various sources has been used. The alignments comprising 150, 200, 250, 500, 1,000, and 1,665 taxa (150_ARB,...,1663_ARB) have been extracted from the ARB small subunit ribosomal ribonucleic acid (ssu rRNA) database [14]. Those alignments contain organisms from the domains Eukarya, Bacteria and Archaea. In addition, the 101 and 150 sequence data sets (101_SC, 150_SC, available at www.indiana.edu/~rac/hpc/fastDNAml) are used. Furthermore, two well-known real data sets comprising 218 and 500 sequences (218_RDPII, 500_ZILLA) were included into the test set. In particular, the 500_ZILLA alignment has been studied extensively under the parsimony criterion [3]. A 193-taxon data set 193_VINH is also included which has been used by Vinh *et al.* [12] to assess performance of the PhyNav program. A set of 7 alignments comprising 73, 74, 104, 128, 144, 178, and 180 mitochondrial DNA sequences of mammals (73_Olaf,...,180_Olaf) have kindly been provided by Olaf Bininda-Emonds from the Technische Universität München. Finally, an alignment of 715 archaeal 16s sequences (715_CHUCK) has been obtained from the Pace Laboratory at the University of Colorado at Boulder. It is important to note that, except for the 128_OLAF dataset, it was not possible to obtain real-world alignments with a Γ shape parameter $\alpha \leq 0.3$ or $\alpha \geq 1.4$. The general consensus of responses from Biologists was that the above range of α values is typical for real alignment data (personal communications). In a paper on bacterial phylogenies Dalevi *et al.* make the same observation [5].

Test Platforms: PHYML (v2.4.4), RAxML (now available as RAxML-VI (v1.0)), and IQPNNI (v3.0.b1) have been compiled using `gcc-3.3.3`. All performance tests where executed on the Infiniband-cluster at the Technische Universität München, equipped with 36 2.4GHz Quad-Opteron nodes and 8GB of main memory per node. In order to measure the amount of cache misses in RAxML an Intel 1.3GHz Itanium processor was used, since hardware counters can easily be obtained on this architecture using `pfmon`.

Experimental Setup: For each alignment, a set of randomized parsimony starting trees was generated with RAxML (see [23]). For the smaller alignments ($\leq$ 250 sequences) 10 distinct starting trees per dataset were generated and for the larger ones ($>$ 250 sequences) 5 trees. In order to keep the inference times within reasonable limits the HKY85 model of nucleotide substitution was used. On each dataset/starting tree combination a RAxML search was executed using the HKY85+Γ model, the HKY85+CAT model with $c = 25$ rate categories and the HKY85+CAT model with $c = 50$ rate categories. Moreover, the final HKY85+CAT trees were refined using the R_Γ algorithm. **To ensure a fair comparison the final Γ log likelihood values l_Γ were computed for all trees obtained under HKY85+CAT. Therefore, all likelihood-based comparisons in this study refer to Γ likelihood values!** This evaluation of final topologies was performed using the RAxML model and branch length optimization option `-f e`, i.e. the topology itself was not altered. The topological Robinson-Foulds distances [21] were also computed using the respective final trees of each individual run with Γ, CAT, and CAT+R_Γ. In order to determine the number of L2 and L3 cache misses RAxML was executed *once* per datasets on 8 alignments (see Table 3) with *one* fixed starting tree under the HKY85+Γ and HKY85+CAT ($c = 25$) models.

PHYML and IQPNNI have been executed once on each alignment under the HKY85+Γ model. The programs have only been executed once per dataset, since they use a deterministic neighbor joining starting tree. Note that, there exist subtle differences in the numerical implementation of the likelihood function between PHYML and IQPNNI and RAxML, especially with respect to scaling very small likelihood values. To this end, the Γ likelihood values of all final tree topologies have been computed with RAxML (`-f e` command line option) to ensure a fair comparison of scores. PHYML is a purely deterministic program and the inference process terminated in all cases. In IQPNNI the number of iterations of the search algorithm can be set to arbitrary values. Therefore, the number of iterations of the IQPNNI algorithm was set to approximately match the inference times of RAxML (execution time ratios between 0.7 and 1.5, see Table 2).

Results: As already mentioned the main argument against using the Γ model is the high memory consumption and the high computational cost. To emphasize this point Figure 1 depicts the development of the Γ likelihood values over time under the HKY85+CAT model with $c = 25$ (RAxML-CAT) and under the HKY85+Γ model (RAxML-CAT) for 715_CHUCK on

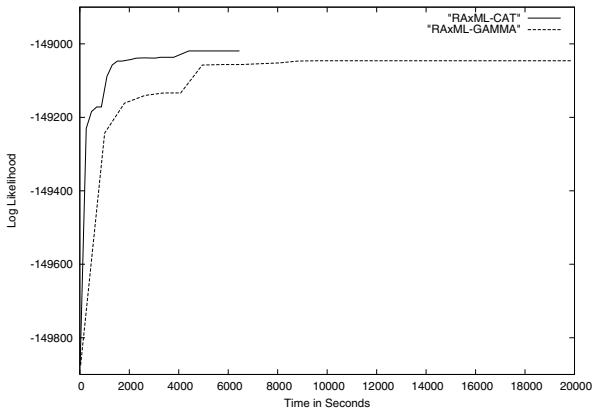

Figure 1. Γ Log Likelihood development over time under the CAT and Γ models

the same starting tree.

Since the Γ model represents the by far more established method all trees obtained with CAT have been re-evaluated under Γ in order to match the perspective of the prevailing opinion in the phylogenetics community. Due to the fact that, on average the results under CAT with $c = 50$ were worse then under $c = 25$, most probably due to a more prominent effect of over-fitting the data, only results with $c = 25$ are depicted. The results for $c = 50$ are available on-line at www.ics.foth.gr/˜stamatak. Table 1 indicates the average RAxML execution time improvements of CAT over Γ (column: T(Γ)/T(CAT)). In addition, the average execution time improvements of CAT+R_Γ compared to inferences under Γ were determined (column: T(Γ)/T(CAT+R_Γ)). Furthermore, the ratios of the final Γ likelihood values $l_\Gamma(\Gamma)/l_\Gamma(CAT)$ obtained under Γ and CAT were determined. Thus, if $l_\Gamma(\Gamma)$ / $l_\Gamma(CAT) < 1.0$ the inference under Γ yielded better final Γ likelihood values and if $l_\Gamma(\Gamma)$ / $l_\Gamma(CAT) > 1.0$ the inference under CAT yielded better final Γ likelihood values. An analogous ratio has been computed to compare CAT+R_Γ with the "pure" Γ inference values (column: $l_\Gamma(\Gamma)/l_\Gamma$(CAT+$R_\Gamma$)). Columns RF(Γ,CAT) and RF(Γ,CAT+R_Γ) indicate the average Robinson-Foulds distance between the final tree topologies obtained under CAT and Γ as well as CAT+R_Γ and Γ respectively. Finally, column α indicates the values of the Γ shape parameter per dataset and colum # pat the number of distinct patterns in each alignment which corresponds to m (length of the computationally intensive `for`-loops). The average values for all parameters determined over all datasets and all starting trees are provided in the bottom line of Table 1. Note that, the apparently small likelihood differences in Table 2 *are significant* since they have been calculated based on the *log* likelihood values and therefore indicate the difference in log likelihood units.

Table 2 provides a performance comparison of the averages obtained with RAxML under CAT ($c = 25$) with IQPNNI and PHYML under Γ. Columns T(P)/T(R) and T(I)/T(R) indicate the execution time ratio of T(PHYML)/T(RAxML) and T(IQPNNI)/T(RAxML) respectively. In addition, columns $l_\Gamma(P)/l_\Gamma(R)$ and $l_\Gamma(I)/l_\Gamma(R)$ depict the ratio of the Γ likelihood values of the final trees l_Γ(PHYML)/l_Γ(RAxML)and l_Γ(IQPNNI)/l_Γ(RAxML).

Dataset	T(P)/T(R)	$l_\Gamma(P)/l_\Gamma(R)$	T(I)/T(R)	$l_\Gamma(I)/l_\Gamma(R)$
73_OLAF	0.84	1.000097	1.07	1.000077
74_OLAF	1.25	0.999948	1.23	0.999948
104_OLAF	0.79	1.000235	1.44	1.000684
128_OLAF	0.99	1.000041	0.92	0.999752
144_OLAF	1.43	1.000435	1.03	1.000706
178_OLAF	0.77	0.999954	0.80	0.999797
180_OLAF	0.37	1.002741	1.10	1.001716
101_SC	1.03	1.002887	1.20	1.001444
150_SC	0.90	1.003077	1.03	1.002178
150_ARB	2.17	1.000338	1.59	1.000444
193_VINH	0.36	1.000965	0.91	1.000542
200_ARB	1.48	1.000601	0.96	1.000265
218_RDPII	0.72	1.001665	0.91	1.001024
250_ARB	1.01	1.000267	0.82	1.000250
500_ARB	0.35	1.003701	0.76	1.001822
500_ZILLA	0.31	1.001872	0.72	1.000869
715_CHUCK	0.57	1.006013	0.76	1.003195
1000_ARB	0.65	1.002969	1.22	1.002700
1663_ARB	0.23	1.003419	0.82	1.003063
Averages	0.85	1.001644	1.02	1.001078

Table 2. Performance comparison of RAxML-CAT with IQPNNI and PHYML

Table 3 indicates the ratios L2(Γ)/L2(CAT) and L3(Γ)/L3(CAT) of the number of L2 and L3 cache misses RAxML produced under the CAT and Γ models for a representative subset of alignments. Column T(Γ)/T(CAT) indicates the execution time ratios. It is important to emphasize that these results deviate from the results in Table 1 since they have been obtained on a distinct CPU architecture and are *not average values* over multiple starting trees.

Dataset	L2(Γ)/L2(CAT)	L3(Γ)/L3(CAT)	T(Γ)/T(CAT)
150_SC	10.76	6.88	3.55
150_ARB	5.83	5.84	3.62
193_V	10.90	8.19	2.82
200_ARB	6.32	10.02	5.13
218_RDPII	5.57	6.98	3.31
250_ARB	7.12	10.37	4.94
500_ZILLA	6.83	4.30	2.94
715_V	6.33	4.26	2.73
Averages	7.46	7.11	3.63

Table 3. L2 and L3 Cache misses for Γ and CAT

Discussion: Modeling processes we know little about, such as evolution, can lead to controversial dis-

Dataset	T(Γ)/T(CAT)	T(Γ)/T(CAT+R_Γ)	l_Γ(Γ)/l_Γ(CAT)	l_Γ(Γ)/l_Γ(CAT+R_Γ)	RF(Γ,CAT)	RF(Γ,CAT+R_Γ)	α	# pat
73_OLAF	4.177018	2.779953	0.999959	0.999997	0.008392	0.005594	1.180	1,196
74_OLAF	3.456038	2.429559	0.999963	0.999963	0.029371	0.029371	0.575	578
104_OLAF	2.971896	1.465592	0.999616	1.000293	0.113659	0.098049	0.329	581
128_OLAF	8.728934	4.362863	1.000026	1.000268	0.016996	0.016996	3.166	2,985
144_OLAF	4.353371	2.233404	0.999983	1.000107	0.055789	0.055088	0.825	1,254
178_OLAF	4.742052	2.397997	0.999998	1.000183	0.026346	0.026062	0.634	1,150
180_OLAF	3.261044	2.300603	0.999608	1.000112	0.048179	0.046499	0.454	924
101_SC	8.607863	4.081393	0.999791	0.999873	0.098492	0.084925	0.417	1,630
150_SC	4.212270	2.630621	0.999955	1.000037	0.040404	0.032323	0.433	1,130
150_ARB	6.935958	4.125580	1.000019	1.000032	0.013805	0.014478	0.562	2,137
193_VINH	2.541966	1.822700	0.999929	1.000007	0.117755	0.112272	1.313	459
200_ARB	7.359741	3.981281	1.000068	1.000089	0.036272	0.034257	0.534	2,253
218_RDPII	5.890610	2.320172	0.999824	1.000018	0.120092	0.103695	0.545	1,847
250_ARB	7.076141	3.817160	1.000027	1.000076	0.032394	0.028974	0.580	2,330
500_ARB	7.378079	3.243040	1.000112	1.000207	0.057573	0.050351	0.579	2,751
500_ZILLA	4.156063	3.014160	1.000160	1.000203	0.054162	0.048947	0.494	1,193
715_CHUCK	4.663363	2.297917	0.999991	1.000146	0.043868	0.039804	0.842	1,231
1000_ARB	8.151405	2.894259	1.001454	1.001549	0.051377	0.048072	0.552	3,364
1663_ARB	4.897310	1.827990	1.000221	1.000320	0.087571	0.084487	0.621	1,576
Averages	5.450585	2.843487	1.000037	1.000183	0.055395	0.050539	0.770	1,609

Table 1. Comparison of final Γ and CAT trees computed with RAxML

cussions and views. Thus, an empirical approach which takes into account the biological and medical insights which can be achieved using competing models should be adopted. This also holds for modeling rate heterogeneity in ML programs. The general necessity of incorporating rate heterogeneity has been empirically deduced and accepted. In order to account for the prevailing views trees inferred via CAT have been evaluated under Γ to demonstrate that CAT can find topologies with equally good and even partially better likelihood values than Γ under *exactly the same* search algorithm. Thus, CAT can be used as a workaround for the computationally intense Γ model in a large number of practical cases. In addition, the R_Γ search option has been incorporated into RAxML to offer a trade-off between the currently widely preferred Γ model and the speed of CAT. The CAT+R_Γ algorithm still yields an average reduction in execution times by a factor of 2.8. The maximum average deviations in Γ likelihood scores range between −0.000392 and +0.001454 for CAT and −0.000127 and +0.001549 for CAT+R_Γ respectively (see Table 1). The score-differences in those cases where CAT yields inferior average likelihoods than Γ are acceptable. Moreover, on average over all datasets both CAT and CAT+R_Γ yield better Γ likelihood scores at significantly lower inference times. Despite the relatively small average deviations of likelihood scores the topological distance between trees obtained by Γ and CAT is in some cases relatively large (104_OLAF, 101_SC, 193_VINH, 218_RDPII, 1663_ARB). Due to the more exhaustive search algorithm and the highly optimized implementation of the likelihood functions, both for Γ and CAT, RAxML clearly out-competes other fast and popular ML programs such as IQPNNI and PHYML within the same amount of execution time. For PHYML the maximum score deviations range from −0.000052 to +0.006013 and for IQPNNI from −0.000248 to +0.003195. It is important to note, that IQPNNI and PHYML only out-competed RAxML under CAT without refinement on 3 of the smaller datasets where considerations regarding inference times and memory consumption are less important. Simulated data has not been used in the present study because there is no means to generate simulated alignment data with Seq-Gen [20] under the CAT model as opposed to the Γ model. Therefore, such an analysis would *a priori* be biased in favor of Γ. From the HPC perspective, the arguments in favor of CAT are evident: The inference times are significantly better for CAT and CAT+R_Γ due to the lower number of L2 and L3 cache misses (see Table 3) and the smaller number of floating point operations. In addition, analyses of huge trees with several thousands of taxa, can be conducted with a quadruple (for the most common implementation of Γ using 4 distinct rate categories) alignment size under CAT.

5. Conclusion & Future Work

To the best of the author's knowledge this paper provides the first comparative study of the Γ and CAT models of rate heterogeneity on typical real-world alignment data under the same search algorithm. In addition, the first detailed description of a simple rate categorization algorithm is given. The current release of RAxML-VI which is available as open source code at www.ics.foth.gr/~stamatak also incorporates the R_Γ algorithm. It is important to note that the results in this paper have been obtained using the old and significantly slower hill-climbing search algorithm of RAxML-V. The final Γ likelihood values of trees inferred under CAT and CAT+R_Γ are in many cases better than those computed under Γ and only slightly worse in the remaining cases. Moreover, RAxML under CAT (without R_Γ) out-competes other popular and

fast ML programs under Γ on all large datasets where HPC-related considerations are important. However, a method to mathematically determine a "good" number of rate categories c for a specific alignment under some statistical criterion is desirable. Nonetheless, **based on the Γ likelihood scores**, it has been demonstrated, that CAT can be used as a replacement for Γ. Moreover, CAT currently represents the only computationally feasible solution to accommodate rate heterogeneity in the analyses of huge trees, given that an analysis of 25,000 taxa with RAxML-VI currently requires 2GB of main memory.

Acknowledgments

The author is grateful to Charles Robertson and Olaf Bininda-Emonds for providing some of the real-world datasets for this study. Special thanks go to Daniele Catanzaro for the very helpful comments on this manuscript.

References

[1] D. Bader, B. Moret, and L. Vawter. Industrial applications of high-performance computing for phylogeny reconstruction. In *Proc. of SPIE ITCom*, volume 4528, pages 159–168, 2001.

[2] M. Brauer, M. Holder, L. Dries, D. Zwickl, P. Lewis, and D. Hillis. Genetic algorithms and parallel processing in maximum-likelihood phylogeny inference. *Molecular Biology and Evolution*, 19:1717–1726, 2002.

[3] M. W. Chase and et al. Phylogenetics of seed plants: An analysis of nucleotide sequences, from the plastid gene rbcl. *Annals of the Missouri Botanical Garden*, 80:528–580, 1993.

[4] B. Chor and T. Tuller. Maximum likelihood of evolutionary trees is hard. In *Proc. of RECOMB05*, 2005.

[5] D. Dalevi, P. Hugenholtz, and L. Blackall. A multiple-outgroup approach to resolving divisionlevel phylogenetic relationships using 16s rdna data. *Int. J. of Syst. and Evol. Microbiol.*, 51:385–391, 2001.

[6] J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376, 1981.

[7] J. Felsenstein. Phylip (phylogeny inference package) version 3.6, 2004. Distributed by the author. Department of Genome Sciences, University of Washington, Seattle.

[8] S. Guindon and O. Gascuel. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst. Biol.*, 52(5):696–704, 2003.

[9] M. Hasegawa, H. Kishino, and T. Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial dna. *J. Mol. Evol.*, 22:160–174, 1985.

[10] J. Huelsenbeck and F. Ronquist. Mrbayes: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17:754–755, 2001.

[11] G. Jobb, A. Haeseler, and K. Strimmer. Treefinder: A powerful graphical analysis environment for molecular phylogenetics. *BMC Evolutionary Biology*, 4, 2004.

[12] S. Le, H. Schmidt, and A. Haeseler. Phynav: A novel approach to reconstruct large phylogenies. In *Proc. of GfKl conference*, 2004.

[13] A. Lemmon and M. Milinkovitch. The metapopulation genetic algorithm: An efficient solution for the problem of large phylogeny estimation. *Proc. of the Nat. Acad. of Sci.*, 99:10516–10521, 2001.

[14] W. Ludwig and et al. Arb: A software environment for sequence data. *Nucl. Acids Res.*, 32:1363–1371, 2004.

[15] S. Meyer and A. v. Haeseler. Identifying site-specific substitution rates. *Mol. Biol. Evol.*, 20:182–189, 2003.

[16] B. Minh, L. Vinh, A. Haeseler, and H. Schmidt. piqpnni - parallel reconstruction of large maximum likelihood phylogenies. *Bioinformatics*, 2005.

[17] G. Olsen, H. Matsuda, R. Hagstrom, and R. Overbeek. fastdnaml: A tool for construction of phylogenetic trees of dna sequences using maximum likelihood. *Comput. Appl. Biosci*, 20:41–48, 1994.

[18] G. Olsen, S. Pracht, and R. Overbeek. Dnarates distribution. unpublished. geta.life.uiuc.edu/~gary/programs/DNArates.html.

[19] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[20] A. Rambaut and N. C. Grassly. Seq-gen: An application for the monte carlo simulation of dna sequence evolution along phylogenetic trees. *Comp. Appl. Biosc.*, 13:235–238, 1997.

[21] D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53:131–147, 1981.

[22] H. Schmidt, K. Strimmer, M. Vingron, and A. Haeseler. Tree-puzzle: maximum likelihood phylogenetic analysis using quartets and parallel computing. *Bioinformatics*, 18:502–504, 2002.

[23] A. Stamatakis, T. Ludwig, and H. Meier. Raxml-iii: A fast program for maximum likelihood-based inference of large phylogenetic trees. *Bioinformatics*, 21(4):456–463, 2005.

[24] D. L. Swofford and G. J. Olsen. Phylogeny reconstruction. In D. Hillis and C. Moritz, editors, *Molecular Systematics*, chapter 11, pages 411–501. Sinauer Ass. Inc., Sunderland, Massachusetts, USA, 1990.

[25] Z. Yang. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites. *J. Mol. Evol.*, 39:306–314, 1994.

[26] Z. Yang. Among-site rate variation and its impact on phylogenetic analyses. *Trends Ecol. Evol.*, 11:367–372, 1996.