VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

gPartition: An efficient alignment partitioning program for genome datasets

Thu Kim Le[1,2], Do Duc Dong[1], Bui Ngoc Thang[1], Diep Thi Hoang[1], Nguyen Phương Thao[3], and Le Sy Vinh[1]

[1]*University of Engineering and Technology, Vietnam National University Hanoi, 144 Xuan Thuy, Cau Giay, 10000 Hanoi, Vietnam*

[2]*Hanoi University of Science and Technology, 1st Dai Co Viet, Hai Ba Trung, 10000 Hanoi, Vietnam*

[3]*Institute of Information Technology, Vietnam Academy of Science and Technology, Hoang Quoc Viet, 10000 Hanoi, Vietnam*

Abstract: Phylogenomics, or evolutionary inference based on genome alignment, is becoming prominent thanks to next-generation sequencing technologies. In model-based phylogenomics, the partition scheme has a significant impact on inference performance, both in terms of log-likelihoods and computation time. Therefore, finding an optimal partition scheme, or partitioning, is critical in a phylogenomic inference pipeline. To accomplish this, one needs to divide the alignment sites into disjoint partitions so that the sites of similar evolutionary models are in the same partition. Computational partitioning is a recent approach of increasing interest due to its capability of modeling the site-rate heterogeneity within a single gene. State-of-the-art computational partitioning methods, such as mPartition or RatePartition, are, however, ineffective on long alignments of millions of sites. In this paper, we introduce gPartition, a new computational partitioning method leveraging both the site rate and the best-fit substitution model. We conducted experiments on recently published alignments to compare gPartition with mPartition and RatePartition. gPartition was orders of magnitude faster than other methods. The AIC score demonstrated that gPartition produced partition schemes that were better or comparable to mPartition. gPartition outperformed RatePartition on all examined alignments. We implemented our proposed method in the gPartition program to help researchers partition genome alignments with millions of sites more efficiently.
Availability: The gPartition program is written in Python and freely available at https://github.com/thulekm/gPartition for both Linux and Windows users.

Keywords: Rate heterogeneity, alignment partitioning method, genome datasets.

1. Introduction

The maximum likelihood (ML) method is one of the most common methods to infer phylogenetic trees from nucleotide alignments [1]. The ML method requires nucleotide substitution models to calculate the likelihood of phylogenies [2], [3]. Determining proper substitution models for an alignment under the study is a challenging task in phylogenetic inferences. This task is becoming more imperative because long or even whole-genome alignments are now easily created thanks to the

advantages of next-generation sequencing technologies.

It is well known that the nucleotide substitution processes vary among sites in the alignment [4], therefore, using one substitution model for all sites might lead to incorrect trees. A number of approaches have been proposed to handle the heterogeneity of evolutionary processes among sites [5]–[9]. Among these, mixture models [7], [9] and alignment partitioning [5], [10], [11] are the most popular approaches in practice. The mixture model approach uses multiple models to describe the evolutionary processes of the alignment. The likelihood of each site in the alignment is estimated as the weighted sum of likelihood values derived with the models. We note that the mixture model approach is computationally expensive because it has to compute likelihood values with different models, and not applicable for long alignments.

An alignment partitioning method classifies sites in the alignment into disjoint subsets (referred to as a partition scheme) such that all sites in a subset are assumed to evolve under the same evolutionary process that might be different between subsets. The simplest alignment partitioning method uses gene boundaries to divide the alignment into a list of loci. However, several studies have shown that partitioning by gene boundaries is not good enough because sites in the same gene might evolve under different evolutionary processes [12]–[14].

The site rate-based partitioning method, notably RatePartition, classifies sites into subsets based on their evolutionary rates such that sites in the same subset have similar evolutionary rates [10], [11]. The evolutionary rates of sites are normally calculated by the TIGER algorithm [12]. The complexity of TIGER increases quadratically with the alignment length, thus inapplicable to long alignments. The assumption that sites with similar evolutionary rates evolve under the same evolutionary process is not biologically realistic. Another pitfall of site rate-based methods is that they group all invariant sites into one subset that might increase the likelihood of trees but lead to biased trees [11], [15].

The model-based partitioning method that employs both evolutionary rates and best-fit substitution models of sites to divide alignments has been proposed to improve the site rate-based method [16]. Although the model-based method is better than both the gene boundary-based and the site rate-based methods in building maximum likelihood trees, its hierarchical top-down partitioning approach is computationally expensive and infeasible for long alignments.

We design gPartition algorithm to overcome the computational burden of the existing methods in order to analyze long or even whole-genome alignments. The gPartition method combines the fastTIGER [17] algorithm to rapidly approximate the evolutionary rates of sites and a new model-based partition scheme to efficiently divide long alignments.

2. Methods

2.1 Likelihood estimation

Let $A = \{A_{xi}\}$ denote an alignment of n nucleotide sequences and l sites where A_{xi} is the nucleotide of sequence x at site i. The ML phylogenetic tree inference searches for an unrooted binary tree T with n leaves (i.e., describing the relationships among n sequences) and a nucleotide substitution model M (i.e., representing substitution rates between nucleotides during the evolution) that maximize the likelihood value $L(T, M; A)$. The model M is typically characterized by a time-homogeneous, time-continuous, and time-reversible Markov process [1], [18].

To avoid computational complexity, we assume that the evolutionary processes among sites are independent. Accordingly, the likelihood value $L(T, M; A)$ can be calculated from individual sites as follows:

$$L(T, M; A) = \prod_{i=1}^{l} L(T, M; A_i) = \prod_{i=1}^{l} P(A_i | T, M)$$

where A_i is the nucleotide data at site i of A. The likelihood value $L(T, M; A_i)$ can be calculated by the conditional probability $P(A_i|T, M)$ of data A_i given tree T and substitution model M.

The alignment partitioning methods divide the l sites of A into k disjoint subsets $\mathbf{P} = (P_1, \ldots, P_k)$ such that all sites of subset $P_i \in \mathbf{P}$ follow the same evolutionary model M_i. Given a partition scheme $\mathbf{P}$, the maximum likelihood (ML) phylogenetic inference determines a tree T together with a set of evolutionary models $\mathbf{M} = (M_1, \ldots, M_k)$ to maximize the conditional probability value $P(\mathbf{P}|T, \mathbf{M})$. Technically, the likelihood value $L(T, \mathbf{M}; \mathbf{P})$ can now be calculated as follows:

$$L(T, \mathbf{M}; \mathbf{P}) = \prod_{i=1}^{k} \prod_{j=1}^{l_i} L(T, M_i; P_{ij})$$

where l_i is the size of subset P_i and P_{ij} is the data at site j^{th} of subset P_i.

The information-theoretic metrics such as the Akaike information criterion (AIC) [19] or the Bayesian information criterion (BIC) [20] are commonly used to compare the fitness among partition schemes because they compromise the likelihood values and the number of free parameters of partition schemes.

2.2 Fast evolutionary rate estimation

The TIGER algorithm

The evolutionary rates of sites provide useful information for classifying sites for both site rate-based and model-based methods. Normally, the site rates are estimated by the TIGER algorithm by analyzing the sequence similarity among sites [12]. Note that the TIGER method does not require any tree so it avoids the tree-bias in estimating the site rates. For each site i, the TIGER method initializes a sequence group G(i) of subgroups, each containing sequences with the same nucleotide at site i. The sequence groups are used to measure the similarity among sites.

The rate r_i for the site i $(i = 1 \ldots l)$ is defined as:

$$r_i = \frac{\sum_{j \neq i} g(i,j)}{n-1}$$

where g(i, j) is "agreement score" between two sequence groups G(i) and G(j) and calculated as follows:

$$g(i,j) = \frac{\sum_{X \in G(j)} a(X, G(i))}{|G(j)|}$$

where $\mathrm{a}(\mathrm{X}, \mathrm{G(i)}) = \begin{cases} 1 \text{ if } \exists \mathrm{Y} \in \mathrm{G(i)} \mid \mathrm{X} \subseteq \mathrm{Y} \\ 0 \text{ otherwise} \end{cases}$

The r_i score ranges from 0 (the fastest rate) to 1 (the slowest rate).

The complexity of the TIGER algorithm is $\mathrm{O(nl^2)}$. It increases quadratically with the length of the alignment and only applicable for alignments with at most several thousand of sites.

The fastTIGER algorithm

We have introduced a rapid algorithm of the so-called fastTIGER to approximately estimate the evolutionary rates of sites by analyzing the similarity between sequences instead of between sites [17]. Intuitively, if two sequences are highly similar, their nucleotides should be the same at slowly evolving sites. Oppositely, if the nucleotides of two highly similar sequences are different at site i, the evolutionary rate r_i at site i should be high. The similarity $d(x, y)$ between two sequences x and y is calculated by the number of sites with the same nucleotides between two sequences. Our fastTIGER

algorithm approximately estimates the site rate r_i as follows:

$$r_i = 1 - \frac{\sum_{A_{xi}=A_{yi}} d(x,y)}{\sum_{x<y} d(x,y)}$$

The rate r_i ranges from 0 (the slowest rate) to 1 (the fastest rate). The complexity of fastTIGER is $O(n^2 l)$ and linear with the length of the alignment.

Consider an alignment A of 3 sequences and 9 sites as in Table 1, the fastTIGER algorithm first calculates the pairwise sequence similarity, i.e., $d(1,2) = 5; d(1,3) = 3; \quad d(2,3) = 6;$ therefore the sum $d(1,2) + d(1,3) + d(2,3) = 14$.

Table 1: An alignment of 3 sequences with 9 sites.

A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
A	A	C	T	G	A	T	C	T
T	G	C	G	G	T	T	C	T
T	G	C	A	A	T	A	C	T

The rates of 9 sites are calculated as follows:

- Site $A_1 = \text{'ATT'}$: sequence 2 and 3 have the same nucleotide T, therefore

$$r_1 = 1 - \frac{d(2,\ 3)}{14} = 1 - \frac{6}{14} = \frac{8}{14}$$

- Site $A_2 = 'AGG'$: sequence 2 and 3 have the same nucleotide G, therefore

$$r_2 = 1 - \frac{d(2,\ 3)}{14} = 1 - \frac{6}{14} = \frac{8}{14}$$

- Site $A_3 = \text{'CCC'}$, Site $A_8 = 'CCC'$, Site $A_9 = \text{'TTT'}$: all sequences have the same nucleotides, therefore

$$r_3 = r_8 = r_9 = 1 - \frac{d(1,2) + d(1,3) + d(1,4)}{14} = 0$$

- Site $A_4 = 'TGA'$: all sequences have different nucleotides, therefore

$$r_4 = 1 - \frac{0}{14} = 0$$

- Site $A_5 = 'GGA'$ and $A_7 = 'TTA'$: sequences 1 and 2 have the same nucleotide, therefore

$$r_5 = r_7 = 1 - \frac{d(1,\ 2)}{14} = 1 - \frac{5}{14} = \frac{9}{14}$$

- Site $A_6 = 'ATT'$: sequence 2 and 3 have the same nucleotide T, therefore

$$r_6 = 1 - \frac{d(2,\ 3)}{14} = \frac{8}{14}$$

2.3 Model-based partitioning method

We have previously proposed a model-based partitioning algorithm, mPartition, to considerably overcome the limitations of the gene-based and site rate-based algorithms [16]. The mPartition method follows a top-down clustering scheme to partition sites. It uses evolutionary rates to classify sites into slow, medium, and fast evolving subsets, then rearranges sites among the subsets based on their best-fit substitution models to increase the likelihood value. The subsets are repeatedly divided into smaller subsets to maximize the likelihood value. The iterative strategy of mPartition for selecting best-fit substitution models and rearranging sites is computationally expensive.

In this paper, we design the gPartition algorithm to handle long alignments. To this end, gPartition employs a clustering algorithm with three key steps: the seed partitioning step divides all sites into seed-subsets, each containing a small number of sites based on the similarity of their site rates; the model unifying step combines seed-subsets with similar substitution models to obtain larger subsets; and finally, the site repartitioning step rearranges the sites among subsets based on their best-fit substitution models to increase the likelihood value. Note that the gPartition algorithm clusters all sites into subsets only once, instead of many times as in the mPartition algorithm. Moreover, the

gPartition method uses fastTIGER instead of TIGER in the mPartition method to estimate the evolutionary rates of sites.

The gPartition method is specifically described as follows (also see Figure 1):

The gPartition method

- *Rate estimation step:* Estimate evolutionary rates for all sites using the fastTIGER algorithm.
- *Seed partitioning step:* Partition all sites into k seed-subsets based on the similarity of site rates such that on average each seed-subset contains about 100 variant sites, i.e., $\mathrm{k} = \mathrm{l_v}/100$ where $\mathrm{l_v}$ is the number of variant sites. Precisely, the variant site i with rate r_i is classified into the $\mathrm{S^{th}}$ subset if $\frac{\mathrm{S\text{-}1}}{\mathrm{k}} \leq \mathrm{r_i} < \frac{\mathrm{S}}{\mathrm{k}}$. If the S^{th} subset contains fewer than 50 sites, merge it with the $(\mathrm{S}+1)^{\mathrm{th}}$ subset to avoid small subsets [16].
- *Model unifying step:* Determine the best substitution models **M** for the seed-subsets using the IQ-TREE software [21]. Two seed-subsets with nearly identical substitution models (i.e., the Pearson correlation is greater than 0.9999) are merged to create larger subsets.
- *Site repartitioning step*: Select from **M** the best-fit substitution model for each variant site. Reclassify variant site i into subset S if its best-fit substitution model is the substitution model of S .

Note that the numbers of invariant sites distributed to subsets are proportional to the corresponding subset likelihood values. This helps overcome the pitfall of grouping all invariant sites into one subset, causing biased trees [16].

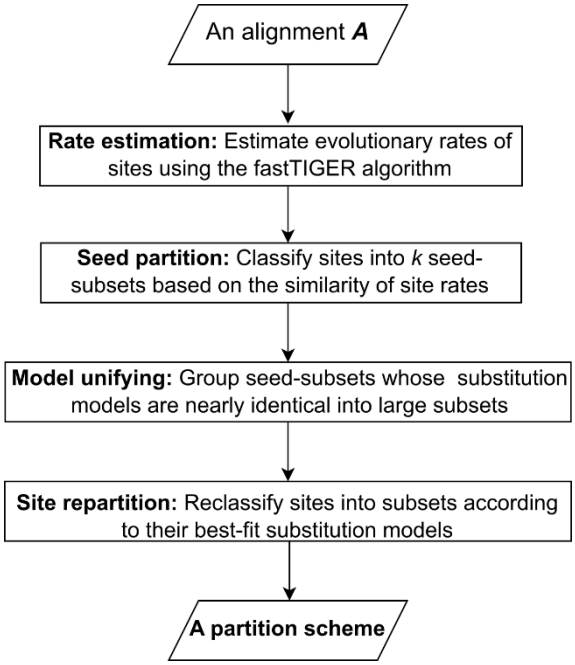

Figure 1 The gPartition algorithm to partition long even whole genome alignments

3. Experiments and results

We examined the performance of gPartition, the site rate-based method RatePartition [11], and the model-based method mPartition on 10 long alignments with lengths ranging from 5998 to 1296042 sites extracted from 10 published genome datasets (Table 2). The partition schemes generated from the methods were used to construct ML trees using the IQ-TREE software. The Akaike information criterion (AIC) scores [19] based on the ML trees were used to compare the partition schemes, i.e., the smaller AIC score indicates the better partition scheme. The running time of the partitioning methods was measured on a workstation with a 2.3 GHz 18-core processor.

The running time of the fastTIGER, TIGER, RatePartition, mPartition, and gPartition algorithms on the test alignments is summarized in Figure 2. It is obvious that fastTIGER is orders of magnitudes faster than the TIGER algorithm. The TIGER algorithm could not

accomplish three large alignments (Aculeata, Neoaves, and Spermatophyta) after three days, while the fastTIGER algorithm required less than 13 minutes for the large alignments. It is obvious that the gPartition method was much faster than the mPartition method, e.g., gPartition took only 3 minutes (0.3 minute from fastTIGER) to divide the Xenopsminutus alignment with 825804 sites while mPartition required 425 minutes (235.2 minutes from TIGER) for the alignment.

Table 2: The alignments used for examining partitioning methods

	Datasets	#sequences	#sites	#Loci	Paper
1	Geometridae.	164	5998	8	(Sihvonen et al. 2011)
2	Pieridae	110	6247	8	(Penz, Devries, and Wahlberg 2012)
3	Osteichthyes	61	19997	61	(Broughton et al. 2013)
4	Vertebrata	110	25919	168	(Fong et al. 2012)
5	Actinopterygii	27	149366	491	(Faircloth et al. 2013)
6	Aculeata	187	183747	807	(Branstetter et al. 2017)
7	Neoaves	33	539526	1541	(McCormack et al. 2013)
8	Phasianidae	18	614159	1501	(Meiklejohn et al. 2016)
9	Xenopsminutus	8	825804	1366	(Smith et al. 2013)
10	Spermatophyta	32	1296042	3924	(Ran et al. 2018)

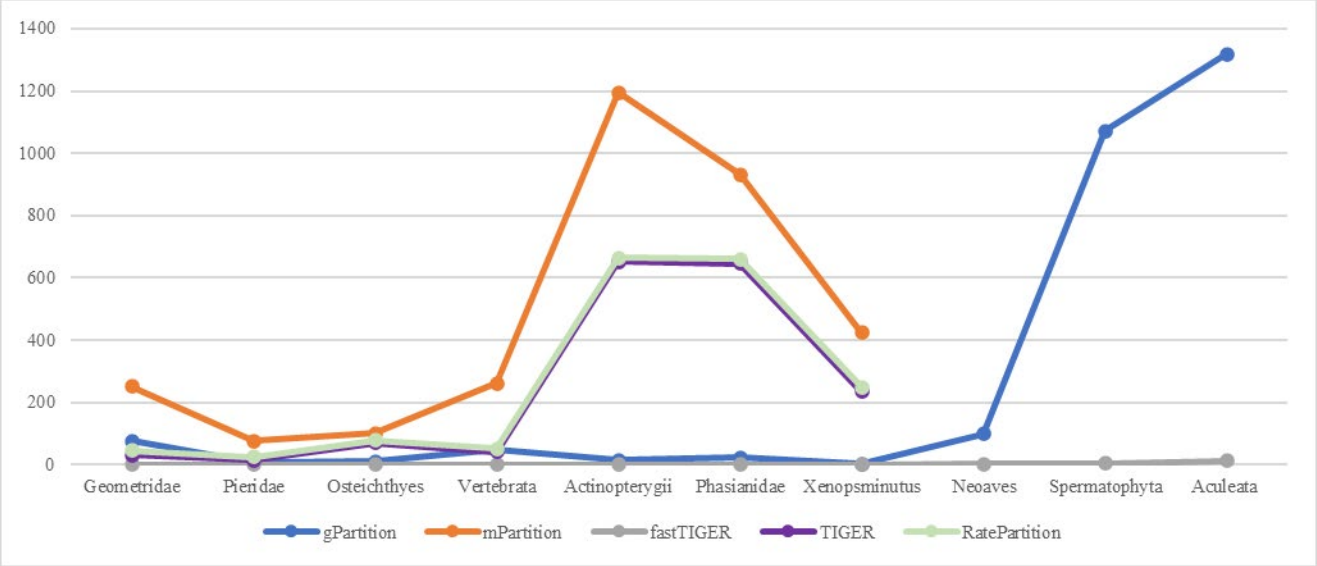

Figure 2 The running times (in minutes) of the site rate estimation methods (i.e., fastTIGER, TIGER algorithms), site rate-based partitioning method (i.e, RatePartition), and model-based partitioning methods (i.e., mPartition and gPartition). As the TIGER method could not analyze the three large datasets (i.e., Aculeata, and Spermatophyta and Neoaves), the running time of TIGER, RatePartition, and mPartition was not available for these datasets.

Table 3 The performance of gPartition, mPartition, and RatePartition with divide factor $d = 4$ methods on 10 alignments. n: the number of sequences; l: the number of sites; NA: Not Available.
The best AIC values are in bold.

Datasets	Size		AIC per site		
	n	l	gPartition	mPartition	RatePartition
Geometridae	164	5998	**60.7**	62.5	62.9
Pieridae	110	6247	**41.3**	42.8	43.4
Osteichthyes	61	19997	**29.3**	31.4	32.0
Vertebrata	110	25919	**20.4**	21.9	21.1
Actinopterygii	27	149366	8.3	**7.8**	9.6
Aculeata	187	183747	**93.3**	NA	NA
Neoaves	33	539526	**4.8**	NA	NA
Phasianidae	18	614159	3.7	**3.4**	3.9
Xenopsminutus	8	825804	2.78	**2.76**	2.81
Spermatophyta	38	1296042	**29.4**	NA	NA

The performance of partitioning algorithms is summarized in **Table 3**. Note that both mPartition and RatePartition methods employed the TIGER algorithm to estimate site rates, therefore, their results were not available for the three large alignments. The AIC scores indicated that gPartition outperformed RatePartition for all alignments. It was better than mPartition on 4 out of 7 alignments and worse on 3 alignments. The average sizes of subsets generated from gPartition and mPartition are 5314 and 4551 sites, respectively.

4. Conclusions

Partitioning alignments into subsets such that sites in the same subset follow similar evolutionary processes is the first crucial step in phylogenetic analyses. The increase in alignment length might add phylogenetic information. Unfortunately, the existing alignment partitioning methods are not designed to analyze long alignments. We described the gPartition algorithm to divide long alignments based on evolutionary rates and best-fit substitution models of sites. The gPartition program was orders of magnitude faster than the existing partitioning programs and was able to partition genome alignments with millions of sites. Our experiments on biological datasets showed that the partition schemes generated from gPartition yielded better species trees in maximum likelihood analyses. We strongly recommend researchers use gPartition to handle the heterogeneity of evolutionary processes among sites in genome analyses.

Acknowledgments

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01.2019.06.

References

[1] P. Lemey, M. Salemi, and A.-M. Vandamme, *The phylogenetic handbook: a practical approach to phylogenetic analysis and hypothesis testing*, 2nd ed. Cambridge University Press, 2009.

[2] C. Blair and R. W. Murphy, “Recent trends in molecular phylogenetic analysis: where to next?,” *J. Hered.*, vol. 102, no. 1, pp. 130–138, 2011.

[3] B. Shapiro, A. Rambaut, and A. J. Drummond, “Choosing appropriate substitution models for the phylogenetic analysis of protein-coding sequences.,” *Molecular biology and evolution*, vol. 23, no. 1. United States, pp. 7–9, Jan. 2006.

[4] Z. Yang, “Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites:

Approximate methods," *J. Mol. Evol.*, vol. 39, no. 3, pp. 306–314, 1994, doi: 10.1007/BF00160154.

[5] R. Lanfear, C. B, S. Y. Ho, and S. Guindon, "PartitionFinder: combined selection of partitioning schemes and substitution models for phylogenetic analyses," *Mol. Biol. Evol.*, vol. 29, pp. 1695–1701, 2012.

[6] N. Lartillot and H. Philippe, "A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process," *Mol Biol Evol*, vol. 21, pp. 1095–1109, 2004.

[7] S. Q. Le, C. C. Dang, and O. Gascuel, "Modeling protein evolution with several amino acid replacement matrices depending on site rates," *Mol Biol Evol*, vol. 29, pp. 2921–36, 2012.

[8] Q. LS, G. O, and L. N, "Empirical profile mixture models for phylogenetic reconstruction," *Bioinformatics*, vol. 24, pp. 2317–23, 2008.

[9] M. Pagel and A. Meade, "A phylogenetic mixture model for detecting pattern-heterogeneity in gene sequence or character-state data," *Syst. Biol.*, vol. 53, pp. 571–581, 2004, doi: 10.1080/10635150490468675.

[10] P. B. Frandsen, B. Calcott, C. Mayer, and R. Lanfear, "Automatic selection of partitioning schemes for phylogenetic analyses using iterative k-means clustering of site rates," *BMC Evol. Biol.*, vol. 15, p. 13, 2015, doi: 10.1186/s12862-015-0283-7.

[11] J. Rota, T. Malm, N. Chazot, C. Peña, and N. Wahlberg, "A simple method for data partitioning based on relative evolutionary rates," *PeerJ*, vol. 6, p. e5498, 2018, doi: 10.7717/peerj.5498.

[12] C. A. Cummins and J. O. McInerney, "A method for inferring the rate of evolution of homologous characters that can potentially improve phylogenetic inference, resolve deep divergence and correct systematic biases," *Syst. Biol.*, vol. 60, pp. 833–844, 2011.

[13] B. Jeremy M and L. Alan R, "The importance of data partitioning and the utility of Bayes factors in Bayesian phylogenetics," *Syst. Biol.*, vol. 56, pp. 643–655, 2007.

[14] J. Nylander, F. Ronquist, J. P. Huelsenbeck, and J. Nieves-Aldrey, "Bayesian phylogenetic analysis of combined data," *Syst Biol*, vol. 53, pp. 47–67, 2004.

[15] S. Baca, E. Toussaint, and K. Miller, "Molecular phylogeny of the aquatic beetle family Noteridae (Coleoptera: Adephaga) with an emphasis on data partitioning strategies," *Mol. Phylogenet. Evol.*, vol. 107, pp. 282–292, 2017.

[16] T. Le Kim and V. Le Sy, "mPartition: A Model-based method for partitioning alignments," *J. Mol. Evol.*, vol. 88, no. 8, pp. 641–652, 2020.

[17] T. Le Kim and V. Le Sy, "fastTIGER: A rapid method for estimating evolutionary rates of sites from large datasets," in *The 13th International Conference on Knowledge and Systems Engineering, November 10-12, 2021, Bangkok, Thailand*, 2021, p. in press.

[18] J. Felsenstein, *Inferring phylogenies*. Sunderland, MA, USA: Sinauer Associates, 2003.

[19] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Automat. Contr.*, vol. 19, pp. 716–723, 1974, doi: 10.1109/TAC.1974.1100705.

[20] G. Schwarz, "Estimating the dimension of a model," *Ann Stat*, vol. 6, pp. 461–464, 1978.

[21] B. Q. Minh *et al.*, "IQ-TREE 2: new models and efficient methods for phylogenetic inference in the genomic era," *Mol. Biol. Evol.*, vol. 37, no. 5, pp. 1530–1534, 2020.