Toward Inferring Ancestral States and Evolutionary Parameters using a Variational Generative Model for Multiple Sequence Alignments

Amine M. Remita [1] **Abdoulaye Baniré Diallo** [1]

Abstract

Most evolutionary-oriented deep generative models do not explicitly consider the underlying evolutionary dynamics of biological sequences as it is performed within the Bayesian phylogenetic inference framework. In this study, we propose a method for a deep variational Bayesian generative model (EvoVGM) that jointly approximates the true posterior of local evolutionary parameters and generates sequence alignments. Moreover, it is instantiated and tuned for continuous-time Markov chain substitution models such as JC69, K80 and GTR. We train the model via a low-variance stochastic estimator and a gradient ascent algorithm. Here, we analyze the consistency and effectiveness of EvoVGM on synthetic sequence alignments simulated with several evolutionary scenarios and different sizes.

1. Introduction

In systematics and evolutionary biology, probabilistic evolutionary models are extensively used to study unseen and complex historical events affecting the genomes of a set of taxa during a period of time (i.e., recombination, horizontal gene transfer and selective pressure). Their ability to detect evolutionary events and measure their parameters using biological sequences has enabled valuable applications in population genetics (Kern & Haussler, 2010), medicine (Yuan et al., 2015) and epidemiology (Faria et al., 2014; Dudas et al., 2017). These models allow the estimation of probabilities of certain types of mutations such as substitutions (Jukes & Cantor, 1969; Tavaré et al., 1986), indels (Diallo et al., 2007) and genome rearrangements (Sankoff & Blanchette, 1999). Main approaches supporting evolutionary studies, such as phylogenetics, implement evolutionary models with Markovian properties (Tavaré et al., 1986). Typically, evolutionary parameters of these models are jointly represented with different types of high-dimensional variables (discrete and continuous), inducing a computationally intractable joint posterior. Bayesian phylogenetic approaches provide methods to efficiently approximate the intractable joint posterior and quantify the uncertainty in the estimation of the parameters (Yang & Rannala, 1997; Huelsenbeck & Ronquist, 2001). They mainly implement random-walk Markov Chain Monte Carlo (MCMC) algorithms, which can converge to an accurate posterior but with a considerable cost. Furthermore, they are prone to limitations due to the complexity of the posterior (Whidden & Matsen IV, 2015), their dependence on initialization and proposal distribution parameters, and their sensitivity to the prior distributions (Huelsenbeck et al., 2002). Recently, variational inference (VI) has sparked interest in phylogenetics as a robust alternative to approximate the intractable posterior by relying on fast optimization methods (Dang & Kishino, 2019; Fourment & Darling, 2019; Zhang & Matsen IV, 2019; Zhang, 2020). VI finds an optimal candidate from a space of tractable distributions that minimizes the Kullback-Leibler (KL) divergence to the exact posterior (Jordan et al., 1999; Blei et al., 2017). It inherently bounds the intractable marginal likelihood of the observed data. Moreover, VI is also used in building deep generative models (Kingma & Welling, 2014; Rezende et al., 2014). However, contrary to Bayesian phylogenetic inference frameworks, most evolutionary-oriented deep generative models do not explicitly consider the underlying evolutionary dynamics of the biological sequences (Riesselman et al., 2018; Lim & Blanchette, 2020; Weinstein & Marks, 2021).

Here, we propose EvoVGM, a deep variational generative model that simultaneously estimates local evolutionary parameters and generates nucleotide sequence data. Like phylogenetic inference, we explicitly integrate a continuous-time Markov chain substitution model into the generative model. The model is trained in an unsupervised manner following the evolutionary model constraints.

[1]Université du Québec à Montréal.
Correspondence to: <remita.amine@courrier.uqam.ca>.

The 2022 ICML Workshop on Computational Biology. Baltimore, Maryland, USA, 2022.

2. Background

2.1. Notation

The observed data $\mathbf{X}$ is an alignment of M character sequences with length N, where $\mathbf{X} \in \mathcal{A}^{M \times N}$. In our case, the alphabet of characters $\mathcal{A} = \{A, G, C, T\}$ is a set of nucleotides. x_n^m is the character in the m^{th} sequence (x^m) and at the n^{th} site (x_n) of the alignment. Here, we assume that each alignment $\mathbf{X}$ has a hidden ancestral state sequence $a \in \mathcal{A}^N$. We take the hypothesis that each ancestral state a_n has evolved independently from the other states $\{a_i; i \neq n\}$ to an extant character x_n^m over an evolutionary time expressed as a branch length t and following a substitution model defined by a set of parameters ψ. In a Bayesian framework, we seek representations allowing to model uncertainty on the quantity and the composition of different entities. We consider the observable characters ($\mathbf{x}_n^m$) and the ancestral states ($\mathbf{a}_n$) as random variables (noted in bold, unlike scalar values) and represent them by categorical distributions over $\mathcal{A}$. Also, branch lengths ($\mathbf{t}^m$) and substitution model parameters ($\boldsymbol{\psi}$) will be modelled as random variables and will be represented by suitable distributions.

2.2. Markov Chain Models of Character Substitution

The evolution of a character is measured by the number of hidden substitutions that undergoes over time. To estimate this quantity, we assume that the process of evolution follows a continuous-time Markov chain model whose states belong to $\mathcal{A}$. The model is parameterized by a rate matrix $\mathbf{Q}$ and relative frequencies π of characters at equilibrium. Each element of the matrix q_{ij} ($i \neq j$) defines the instantaneous substitution rate of character i changing into character j. The diagonal elements q_{ii} are set up in a way that each row sums to 0. $\mathbf{Q}$ is scaled by a factor μ, so that the time t will be measured in the expected number of substitutions per site and the average rate of substitution at equilibrium will be 1. We use time-reversible Markov chain models assuming the amount of changes from one character to another is the same in both ways. For nucleotide substitution time-reversible models, the equation of $\mathbf{Q}$ is

$$\mathbf{Q} = \begin{pmatrix} \cdot & a\pi_G & b\pi_C & c\pi_T \\ a\pi_A & \cdot & d\pi_C & e\pi_T \\ b\pi_A & d\pi_G & \cdot & f\pi_T \\ c\pi_A & e\pi_G & f\pi_C & \cdot \end{pmatrix} \mu,$$

where $a, b, c, d, e,$ and f are the set of relative substitution rate parameters ρ, and $\pi_A + \pi_G + \pi_C + \pi_T = 1$ are the relative frequencies π. Once $\mathbf{Q}$ is estimated we can compute the probability transition matrix $\mathbf{P}$ over an evolutionary time t as $\mathbf{P}(t) = \exp(\mathbf{Q}\, t)$. The matrix exponential is computed using spectral decomposition of $\mathbf{Q}$ as it is reversible (see (Lemey et al., 2009) and (Yang, 2014) for more details).

Several substitution models could be generated depending on the constraints placed on the set of parameters $\psi = \{\rho, \pi\}$. The simplest model is JC69 with equal substitution rates and uniform relative frequencies (Jukes & Cantor, 1969). The K80 model defines uniform frequencies like JC69, but it differentiates between the two types of substitution rates corresponding to transitions ($\alpha = a = f$) and transversions $\beta = b = c = d = e$ (Kimura, 1980). Usually, K80 is parameterized by the transition/transversion rate ratio $\kappa = \alpha/\beta$. Finally, the general time-reversible (GTR) model sets all the parameters ψ free (Tavaré et al., 1986; Yang, 1994).

2.3. Evolutionary Posterior

Along with $\mathbf{a}$ and $\mathbf{t}$ variables, we consider the parameters of the Markov chain model $\boldsymbol{\psi}$ as latent (hidden) variables to be inferred from the observed data $\mathbf{X}$. Assuming an independent evolution of the sites in an alignment (Felsenstein, 1981), the marginal likelihood of the data $\mathbf{X}$ factorizes into $p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n)$. The inference of the latent variables for each site x_n requires the computation of the evolutionary joint posterior $p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}|\mathbf{x}_n)$. The evolutionary posterior is calculated according to Bayes' theorem:

$$p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}|\mathbf{x}_n) = \frac{p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})}{p(\mathbf{x}_n)}, \tag{1}$$

which exposes the joint density of the observable variable and the latent variables $p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})$, and the marginal likelihood $p(\mathbf{x}_n)$. The former is factorized as a product of the joint prior density of the latent variables $p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})$ and the likelihood $p(\mathbf{x}_n|\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})$. The latter is calculated by marginalizing over the values of all the latent variables as $\iiint p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})\, p(\mathbf{x}_n|\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})\, d\mathbf{a}_n\, d\mathbf{t}\, d\boldsymbol{\psi}$. The computation of the evolutionary joint posterior density is computationally intractable as it depends on the evaluation of $p(\mathbf{x}_n)$, which is intractable due to the integrals in its marginalization. We show in the next section strategies to determine each term in the equation 1.

3. Proposed Evolutionary Model

In this section, we describe a deep variational generative model that simultaneously estimates local evolutionary biological parameters and generates nucleotide sequence data. Similar to deep variational-based generative models (Kingma & Welling, 2014; Rezende et al., 2014), the proposed model architecture consists of two main sub-models: 1) a set of deep variational encoders that infers the parameters of evolutionary-latent-variable distributions and allows sampling, and 2) a generating model that computes probability transition matrices from sampled latent variables and generates a distribution of sequence alignments from reconstructed ancestral states (see Figure 1).

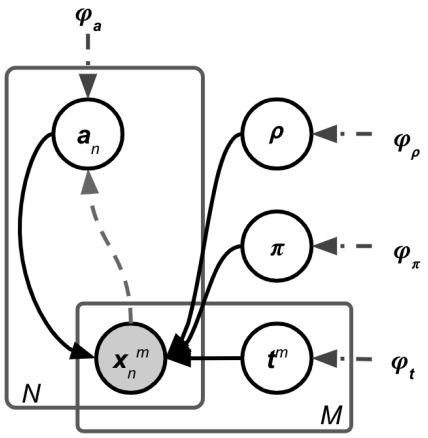


Figure 1: Graphical illustration of the inference (dashed gray lines) and the generation (solid lines) processes of the GTR-based variational generative model. Gray circles represent the observed variables. Blank circles represent the latent variables. $\{\varphi_{\mathbf{a}}, \varphi_{\mathbf{t}}, \varphi_{\boldsymbol{\rho}}, \varphi_{\boldsymbol{\pi}}\}$ is the set of hyper-parameters of the prior densities.

3.1. Variational Inference of the Joint Posterior

We use mean-field variational inference to approximate the true joint posterior probability distribution by a new probability distribution $q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}|\mathbf{x}_n)$ (Jordan et al., 1999; Kingma & Welling, 2014; Rezende et al., 2014). We model each latent variable by an independent approximate distribution whose parameters will be inferred using a non-linear transformation either of $\mathbf{x}_n$, or an independent, fixed random noise $\boldsymbol{\zeta}$. The non-linear transformations are implemented using deep neural networks (NeuralNet) parameterized by a set of independent and adaptable variational parameters $\phi = \{\phi_{\mathbf{a}}, \phi_{\mathbf{t}}, \phi_{\boldsymbol{\psi}}\}$.

For each sequence x^m, we infer and sample an evolutionary time variable $\mathbf{t}^m$. We model its approximate density $q_{\phi_{\mathbf{t}}}(\mathbf{t}^m)$ using a gamma distribution to ensure the positiveness of the samples. The parameters of the distribution (shape and rate) are produced by a non-linear transformation on uniform noise $\boldsymbol{\zeta}_{\mathbf{t}}$ as follows:

$$q_{\phi_{\mathbf{t}}}(\mathbf{t}^m) = \text{Gamma}(\mathbf{t}^m;\, \text{NeuralNet}(\boldsymbol{\zeta}_{\mathbf{t}};\, \phi_{\mathbf{t}})).$$

Next, we infer and sample the latent variables of the Markov chain model parameters $\boldsymbol{\psi}$ with independent approximate densities $q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi})$. The JC69 model does not have any free parameters to be estimated, so $\boldsymbol{\psi} = \emptyset$. For the K80 model, we infer the latent variable of the transition/transversion rate ratio ($\boldsymbol{\kappa}$) using a gamma-based approximate distribution ($q_{\phi_{\boldsymbol{\kappa}}}(\boldsymbol{\kappa})$) to ensure the positiveness of the samples. Its local parameters are produced by a neural network on uniform noise $\boldsymbol{\zeta}_{\boldsymbol{\kappa}}$ as follows:

$$q_{\phi_{\boldsymbol{\kappa}}}(\boldsymbol{\kappa}) = \text{Gamma}(\boldsymbol{\kappa};\, \text{NeuralNet}(\boldsymbol{\zeta}_{\boldsymbol{\kappa}};\, \phi_{\boldsymbol{\kappa}})).$$

In the case of the GTR model, we model the variational densities of the substitution rate parameters ($\boldsymbol{\rho}$) and the relative frequencies ($\boldsymbol{\pi}$) using Dirichlet distributions. This ensures that the sum of the sampled values is equal to one. Their concentrations are generated by a set of independent neural networks on uniform noises $\boldsymbol{\zeta}_{\boldsymbol{\rho}}$ and $\boldsymbol{\zeta}_{\boldsymbol{\pi}}$, respectively:

$$q_{\phi_{\boldsymbol{\rho}}}(\boldsymbol{\rho}) = \text{Dirichlet}(\boldsymbol{\rho}; \text{NeuralNet}(\boldsymbol{\zeta}_{\boldsymbol{\rho}};\, \phi_{\boldsymbol{\rho}})),$$
$$q_{\phi_{\boldsymbol{\pi}}}(\boldsymbol{\pi}) = \text{Dirichlet}(\boldsymbol{\pi}; \text{NeuralNet}(\boldsymbol{\zeta}_{\boldsymbol{\pi}};\, \phi_{\boldsymbol{\pi}})).$$

Lastly, for each site x_n, an ancestral variable $\mathbf{a}_n$ is inferred and sampled with an approximate density $q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \,|\, \mathbf{x}_n)$ represented by a categorical distribution over the $(|\mathcal{A}| - 1)$-simplex as follows:

$$q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \,|\, \mathbf{x}_n) = \text{Categorical}(\mathbf{a}_n; \text{NeuralNet}(\mathbf{x}_n;\, \phi_{\mathbf{a}})).$$

We apply a non-linear transformation on $\mathbf{x}_n$ to produce the local parameters of $q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \,|\, \mathbf{x}_n)$, which are a set of $|\mathcal{A}|$ probabilities that sum to one. Using a mean-field variational inference approach, the approximate joint posterior factorizes into:

$$q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}|\mathbf{x}_n) = q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \,|\, \mathbf{x}_n) \prod_{m=1}^{M} q_{\phi_{\mathbf{t}}}(\mathbf{t}^m)\, q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi}). \tag{2}$$

3.2. Generating Model Computation

The generating model is represented by the joint density $p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}) = p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})\, p(\mathbf{x}_n|\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})$, which is parameterized only by the local latent variables. We use independent prior densities for the latent variables, so $p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}) = p(\mathbf{a})\, p(\mathbf{t})\, p(\boldsymbol{\psi})$. To ease the computation, we apply for each prior density the same distribution type as its corresponding approximate posterior density and determine its hyper-parameters φ. Moreover, for each nucleotide x_n^m, we use the probability transition matrix $\mathbf{P}(\mathbf{t}^m)$ to define the likelihood function, which is the probability of evolving a character $\mathbf{a}_n$ into $\mathbf{x}_n^m$ during a time $\mathbf{t}^m$, as:

$$\begin{aligned} \widehat{\mathbf{x}}_n^m &= \mathbf{a}_n \times \mathbf{P}(\mathbf{t}^m; \boldsymbol{\psi}), \\ p(\mathbf{x}_n^m|\mathbf{a}_n, \mathbf{t}^m, \boldsymbol{\psi}) &= \text{Categorical}(\mathbf{x}_n^m; \widehat{\mathbf{x}}_n^m). \end{aligned} \tag{3}$$

The likelihood of a site x_n is computed following a pre-order traversal. We call it a top-down likelihood since it includes the sampled ancestral states in its estimation. It is different from the likelihood computed in a phylogeny, which is based on a post-order traversal (Felsenstein, 1981) and does not include sampled ancestral states. Finally, the joint density is

$$p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}) = p(\mathbf{a})\, p(\mathbf{t})\, p(\boldsymbol{\psi}) \prod_{m=1}^{M} p(\mathbf{x}_n^m|\mathbf{a}_n, \mathbf{t}^m, \boldsymbol{\psi}). \tag{4}$$

Table 1: Log likelihood estimates of EvoVGM models using validation alignments of five sequences with a length of 5000 bp. JC69, K80 and GTR substitution models were used to simulate training and validation alignments. The estimates are computed and averaged from fitting and running the models ten times.

	JC69		K80		GTR	
	MEAN	STD	MEAN	STD	MEAN	STD
ACTUAL	-17249.830		-17024.340		-15818.739	
EVOVGM_JC69	-17209.913	142.128	-17287.278	185.441	-16491.810	125.664
EVOVGM_K80	-17203.100	151.758	-17007.724	133.294	-16495.530	175.024
EVOVGM_GTR	-17204.540	121.459	-17014.296	151.457	-15540.730	126.673

3.3. Stochastic Estimator and Learning Algorithm

Variational inference allows us to form a lower bound on the marginal likelihood of each site x_n as $\log p(\mathbf{x}_n) \geq \mathcal{L}_n(\phi, \mathbf{x}_n)$, where $\mathcal{L}_n$ is the evidence lower bound (ELBO) (Jordan et al., 1999; Blei et al., 2017). Putting together equations 1, 2 and 4, we can derive the equation of the multi-sample estimator of the EvoVGM model as follows:

$$\mathcal{L}_n(\phi, \mathbf{x}_n) = \left(\frac{1}{L} \sum_{l=1}^{L} \sum_{m=1}^{M} \log p(\mathbf{x}_n^m | \mathbf{a}_n^l, \mathbf{t}^{m,l}, \boldsymbol{\psi}^l) \right)$$
$$-\alpha_{\text{KL}} \Bigg(\text{KL}(q_{\phi_\mathbf{a}}(\mathbf{a}_n \,|\, \mathbf{x}_n) \parallel p(\mathbf{a})) + \sum_{m=1}^{M} \text{KL}(q_{\phi_\mathbf{t}}(\mathbf{t}^m) \parallel p(\mathbf{t})) + \text{KL}(q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi}) \parallel p(\boldsymbol{\psi})) \Bigg), \quad (5)$$

where L is the sampling size, $\text{KL}(\cdot \parallel \cdot)$ is the Kullback–Leibler divergence, and α_{KL} is a regularization coefficient (see the development of this equation in **??**). This estimator is computationally tractable because it is independent of the direct evaluation of the true joint posterior. To maximize the ELBO and learn the global variational parameters ϕ, EvoVGM estimates and backpropagates the gradients for the whole data $\mathbf{X}$ using the reparameterization trick (Kingma & Welling, 2014) and a gradient ascent optimizer. The algorithm of EvoVGM is detailed in Algorithm 1. It is implemented in Pytorch (Paszke et al., 2019) and its open-source code is available at https://github.com/maremita/evoVGM.

4. Experiments

The evaluation of the proposed Bayesian variational method to estimate evolutionary parameters and generate sequence alignments is oriented towards assessing its consistency, effectiveness, and the understanding of its behavior during the training using simulated sequence alignments.

We used Pyvolve (Spielman & Wilke, 2015) to simulate the evolution of different sequence alignments with a site-wise homogeneity model and a combination of substitution models (JC69, K80 and GTR), the number of sequences (3, 4 and 5) and alignment lengths (100 bp, 1000 bp and 5000 bp). A site-wise homogeneity model evolves sequences from a root sequence with the same substitution model over lineages and with the same branch lengths for nucleotides. The sequence alignments used in the training step of the EvoVGM models were simulated with different random seeds from those used in the validation step but with the same array of evolutionary parameters.

First, we evaluated and compared three variants of the EvoVGM model, each one implemented with a different Markov chain substitution model: EvoVGM_JC69, EvoVGM_K80, EvoVGM_GTR. Each model was fit ten times to the same sequence alignment using a different weight initialization. Table 1 and Figure A.1 show the results of the models trained and evaluated with alignments of five sequences of length 5000 base pairs (bp). All models converge to values closer to or higher than the actual log likelihood of the data, which is calculated with equation 3 using the known ancestral sequences and evolutionary parameters. To assess the consistency and the effectiveness of the models, we calculated the Euclidean distance and the Pearson correlation coefficient between the estimated and actual values of the evolutionary parameters. Mostly, parameter estimates improve when the number of sequences is higher and the alignments are longer. All three models approximated the branch lengths even when trained with datasets simulated with a different substitution model (Tables A.1, A.1, A.1, and others not shown here). For small datasets, EvoVGM_GTR estimates better relative frequencies than substitution rates. However, as the datasets get larger, the approximations of the substitution rates get better (Tables 2 and 3).

Lastly, we assessed the effect of different hyper-parameters (α_{KL}, the size of the hidden layers of the neural networks of the encoders, the sample size, and the learning rate) on the behavior and performance of the three models. Each model was fit ten times on the same alignment of five 5000-bp sequences simulated using its respective substitution model. The results are highlighted in Figures A.2, A.3 and A.4. In

Table 2: Euclidean distance (DIST) and Pearson correlation coefficient (CORR) between actual and estimated substitution rates by EvoVGM_GTR. The GTR substitution model was used to simulate training and validation alignments.

$N \rightarrow$	100		1000		5000	
M	DIST	CORR	DIST	CORR	DIST	CORR
3	0.621	0.103	0.177	0.668	0.129	0.784
4	0.305	0.472	0.114	0.864	0.036	0.985
5	0.206	0.652	0.053	0.968	0.012	0.998

Table 3: Euclidean distance (DIST) and Pearson correlation coefficient (CORR) between actual and estimated relative frequencies by EvoVGM_GTR. The GTR substitution model was used to simulate training and validation alignments.

$N \rightarrow$	100		1000		5000	
M	DIST	CORR	DIST	CORR	DIST	CORR
3	0.190	0.941	0.084	0.991	0.095	0.992
4	0.125	0.891	0.090	0.996	0.050	0.999
5	0.176	0.775	0.022	1.000	0.033	0.999

general, models converge faster when the α_{KL} coefficient is lower, and the number of hidden layers and the learning rate are larger. The sample size does not affect the overall convergence. However, a small sample size induces a substantial variance in the estimator.

5. Conclusion

In this work, we show that a deep variational Bayesian generative method could constitute a feasible option to approximate the true parameters of an evolutionary model and generate the associated sequence alignment. The implementation of this method, EvoVGM, estimates the branch lengths, the ancestral states, and the substitution model parameters from a multiple sequence alignment. We assessed its consistency and effectiveness using sequence alignments simulated with different sizes. In general, the EvoVGM model needs a few thousand iterations to converge. It tends to be accurate with low variance in estimating the evolutionary parameters using fine-tuned hyper-parameters. Moreover, it provides an effective way of estimating the parameters for different substitution models such as JC69, K80, and GTR. The generalization to other models like HKY is also straightforward. For future work, many extensions could be explored to improve the EvoVGM model, such as considering a prior tree topology, investigating the influence of the priors on inference, and allowing parameter heterogeneity across sites and lineages.

References

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

Dang, T. and Kishino, H. Stochastic Variational Inference for Bayesian Phylogenetics: A Case of CAT Model. *Molecular Biology and Evolution*, 36(4):825–833, apr 2019. ISSN 0737-4038.

Diallo, A. B., Makarenkov, V., and Blanchette, M. Exact and heuristic algorithms for the indel maximum likelihood problem. *Journal of Computational Biology*, 14(4):446–461, 2007.

Dudas, G., Carvalho, L. M., Bedford, T., Tatem, A. J., Baele, G., Faria, N. R., Park, D. J., Ladner, J. T., Arias, A., Asogun, D., et al. Virus genomes reveal factors that spread and sustained the ebola epidemic. *Nature*, 544 (7650):309–315, 2017.

Faria, N. R., Rambaut, A., Suchard, M. A., Baele, G., Bedford, T., Ward, M. J., Tatem, A. J., Sousa, J. D., Arinaminpathy, N., Pépin, J., et al. The early spread and epidemic ignition of hiv-1 in human populations. *science*, 346(6205):56–61, 2014.

Felsenstein, J. Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981.

Fourment, M. and Darling, A. E. Evaluating probabilistic programming and fast variational Bayesian inference in phylogenetics. *PeerJ*, 7(12):e8272, dec 2019. ISSN 2167-8359.

Huelsenbeck, J. P. and Ronquist, F. MRBAYES: Bayesian inference of phylogenetic trees. *Bioinformatics*, 17(8): 754–755, 2001.

Huelsenbeck, J. P., Larget, B., Miller, R. E., and Ronquist, F. Potential applications and pitfalls of Bayesian inference of phylogeny. *Systematic Biology*, 51(5):673–688, 2002. ISSN 10635157.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183–233, 1999.

Jukes, T. H. and Cantor, C. R. Evolution of protein molecules. In Munro, H. H. (ed.), *Mammalian protein metabolism*, volume III, pp. 21–132. Academic Press, New York, 1969.

Kern, A. D. and Haussler, D. A population genetic hidden markov model for detecting genomic regions under selection. *Molecular biology and evolution*, 27(7):1673–1685, 2010.

Kimura, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of molecular evolution*, 16 (2):111–120, 1980.

Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. In *Proceedings of the International Conference on Learning Representations*, dec 2014.

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

Lim, D. and Blanchette, M. EvoLSTM: context-dependent models of sequence evolution using a sequence-to-sequence LSTM. *Bioinformatics*, 36(Supplement_1): i353–i361, jul 2020. ISSN 1367-4803.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. *31st International Conference on Machine Learning, ICML 2014*, 4:3057–3070, 2014.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10):816–822, 2018. ISSN 15487105.

Sankoff, D. and Blanchette, M. Probability models for genome rearrangement and linear invariants for phylogenetic inference. In *Proceedings of the third annual international conference on Computational molecular biology*, pp. 302–309, 1999.

Spielman, S. J. and Wilke, C. O. Pyvolve: A flexible python module for simulating sequences along phylogenies. *PLOS ONE*, 10(9):1–7, 09 2015.

Tavaré, S. et al. Some probabilistic and statistical problems in the analysis of dna sequences. *Lectures on mathematics in the life sciences*, 17(2):57–86, 1986.

Weinstein, E. N. and Marks, D. A structured observation distribution for generative biological sequence prediction and forecasting. In *International Conference on Machine Learning*, pp. 11068–11079. PMLR, 2021.

Whidden, C. and Matsen IV, F. A. Quantifying mcmc exploration of phylogenetic tree space. *Systematic biology*, 64(3):472–491, 2015.

Yang, Z. Estimating the pattern of nucleotide substitution. *Journal of molecular evolution*, 39(1):105–111, 1994.

Yang, Z. *Molecular evolution: a statistical approach*. Oxford University Press, 2014.

Yang, Z. and Rannala, B. Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo method. *Molecular biology and evolution*, 14(7):717–724, 1997.

Yuan, K., Sakoparnig, T., Markowetz, F., and Beerenwinkel, N. Bitphylogeny: a probabilistic framework for reconstructing intra-tumor phylogenies. *Genome biology*, 16 (1):1–16, 2015.

Zhang, C. Improved Variational Bayesian Phylogenetic Inference with Normalizing Flows. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 18760–18771. Curran Associates, Inc., 2020.

Zhang, C. and Matsen IV, F. A. Variational Bayesian Phylogenetic Inference. In *International Conference on Learning Representations*, 2019.

A. Appendix

A.1. Development of the ELBO $\mathcal{L}(\phi, \mathbf{X})$

$$
\begin{aligned}
\log p(\mathbf{X}) &= \sum_{n=1}^{N} \log p(\mathbf{x}_n) \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \left[\log \frac{p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})}{p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \left[\log \frac{p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})}{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \frac{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)}{p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_\phi} \left[\log \frac{p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})}{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \right] + \mathbb{E}_{q_\phi} \left[\log \frac{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)}{p(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \right] \\
&= \underbrace{\sum_{n=1}^{N} \mathcal{L}_n(\phi, \mathbf{x}_n)} + \sum_{n=1}^{N} \mathtt{KL} \left(q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n) \parallel p(\mathbf{a}_n, \mathbf{t}_n, \boldsymbol{\psi} \mid \mathbf{x}_n) \right) \\
&\geq \qquad \mathcal{L}(\phi, \mathbf{X}).
\end{aligned}
$$

$$
\begin{aligned}
\mathcal{L}(\phi, \mathbf{X}) &= \sum_{n=1}^{N} \mathcal{L}_n(\phi, \mathbf{x}_n) \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_\phi} \left[\log \frac{p(\mathbf{x}_n, \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi})}{q_\phi(\mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi} \mid \mathbf{x}_n)} \right] \\
&= \sum_{n=1}^{N} \mathbb{E}_{q_\phi} \Big[\log p(\mathbf{x}_n | \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}) + \log p(\mathbf{a}) + \log p(\mathbf{t}) + \log p(\boldsymbol{\psi}) \\
&\qquad - \log q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \mid \mathbf{x}_n) - \log q_{\phi_{\mathbf{t}}}(\mathbf{t}) - \log q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi}) \Big] \\
&= -N \left(\mathbb{E}_{q_\phi} \left[\log p(\boldsymbol{\psi}) - \log q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi}) \right] + \mathbb{E}_{q_\phi} \left[\log p(\mathbf{t}) - \log q_{\phi_{\mathbf{t}}}(\mathbf{t}) \right] \right) \\
&\quad + \sum_{n=1}^{N} \mathbb{E}_{q_\phi} \left[\log p(\mathbf{x}_n | \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}) \right] + \mathbb{E}_{q_\phi} \left[\log p(\mathbf{a}) - \log q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \mid \mathbf{x}_n) \right] \\
&= -N \left(\mathtt{KL}(q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi}) \parallel p(\boldsymbol{\psi})) + \mathtt{KL}(q_{\phi_{\mathbf{t}}}(\mathbf{t}) \parallel p(\mathbf{t})) \right) \\
&\quad + \sum_{n=1}^{N} \mathbb{E}_{q_\phi} \left[\log p(\mathbf{x}_n | \mathbf{a}_n, \mathbf{t}, \boldsymbol{\psi}) \right] - \mathtt{KL}(q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \mid \mathbf{x}_n) \parallel p(\mathbf{a})) \\
&= -N \left(\mathtt{KL}(q_{\phi_{\boldsymbol{\psi}}}(\boldsymbol{\psi}) \parallel p(\boldsymbol{\psi})) + \sum_{m=1}^{M} \mathtt{KL}(q_{\phi_{\mathbf{t}}}(\mathbf{t}^m) \parallel p(\mathbf{t})) \right) \\
&\quad + \sum_{n=1}^{N} \left(\frac{1}{L} \sum_{l=1}^{L} \sum_{m=1}^{M} \log p(\mathbf{x}_n^m | \mathbf{a}_n^l, \mathbf{t}^{m,l}, \boldsymbol{\psi}^l) \right) - \mathtt{KL}(q_{\phi_{\mathbf{a}}}(\mathbf{a}_n \mid \mathbf{x}_n) \parallel p(\mathbf{a})).
\end{aligned}
$$

A.2. Learning algorithm for EvoVGM model

Algorithm 1 Learning algorithm for **EvoVGM**

Input: Alignment $\mathbf{X}$ of M sequences with length N
$\phi_{\mathbf{a}}, \phi_{\mathbf{t}}, \phi_{\boldsymbol{\psi}} \leftarrow$ initialize global variational parameters
for $i \in [1...max\_iter]$ **do**
  $\mathbf{t}^m \leftarrow$ Sample $M \times L$ branch latent variables ($\phi_{\mathbf{t}}$)
  $\boldsymbol{\psi} \leftarrow$ Sample L evolutionary latent variables ($\phi_{\boldsymbol{\psi}}$)
  $\mathbf{P}^m \leftarrow$ Compute $M \times L$ probability transition matrices ($\mathbf{t}^m, \boldsymbol{\psi}$)
  for $n \in [1...N]$ **do**
    $\mathbf{a}_n \leftarrow$ Sample L ancestor latent variable ($\mathbf{x}_n; \phi_{\mathbf{a}}$)
    $\hat{\mathbf{x}}_n \leftarrow$ Generate $M \times L$ nucleotides ($\mathbf{a}_n, \mathbf{P}^m$)
    $\mathcal{L}_n \leftarrow$ Compute ELBO according to the equation A.1
    $\mathcal{L} \mathrel{+}= \mathcal{L}_n$
  end for
  $\boldsymbol{g} \leftarrow$ Compute gradients of total ELBO ($\mathcal{L}$)
  $\phi_{\mathbf{a}}, \phi_{\mathbf{t}}, \phi_{\boldsymbol{\psi}} \leftarrow$ Update parameters ($\boldsymbol{g}$) with gradient ascent optimizer
end for

A.3. Supplemental results

A.3.1. Assessment of EvoVGM models on different datasets

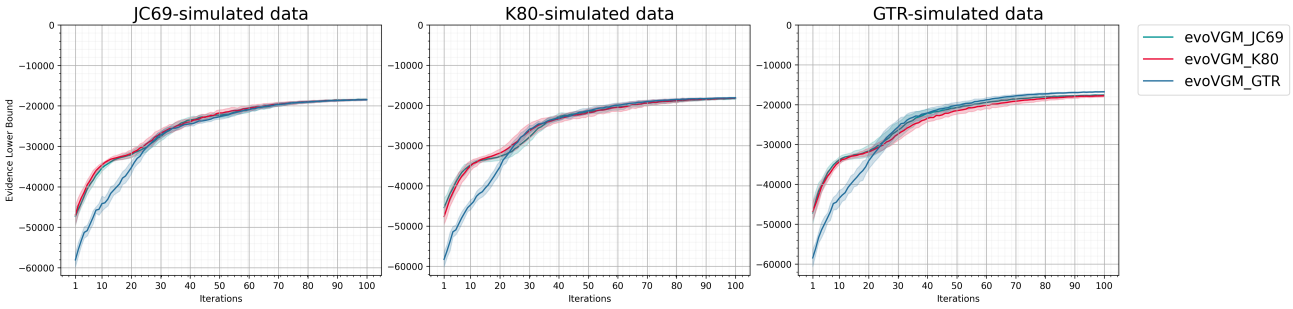


Figure A.1: Evidence lower bound (ELBO) of the EvoVGM models. The models were trained with alignments of five 5000-bp sequences which were simulated with three different substitution models (JC69, K80 and GTR). We show the trend of the ELBO over the first 100 iterations of the training.

Table A.1: Euclidean distance (DIST) and Pearson correlation coefficient (CORR) between original and estimated branch lengths by EvoVGM_JC69. Rows correspond to the number of sequences. Columns correspond to the alignment length. Datasets simulated with JC69 substitution model.

$N \rightarrow$	100		1000		5000	
M	DIST	CORR	DIST	CORR	DIST	CORR
3	0.129	0.969	0.069	0.982	0.143	0.982
4	0.166	0.938	0.065	0.997	0.079	0.997
5	0.179	0.841	0.096	0.993	0.076	0.990

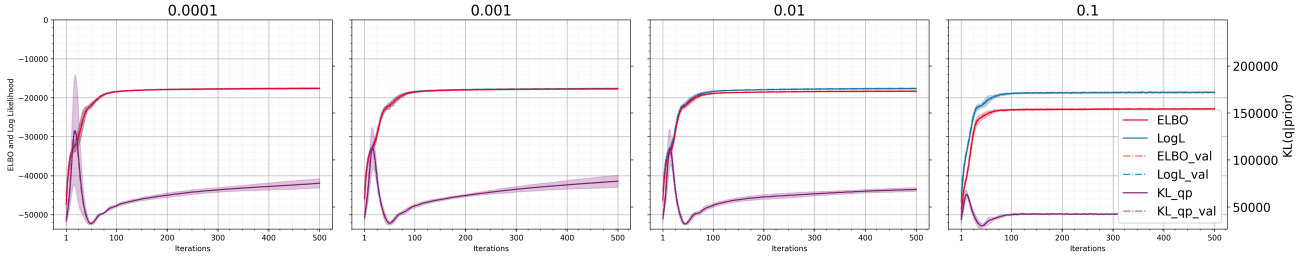
Table A.1: Euclidean distance (DIST) and Pearson correlation coefficient (CORR) between original and estimated branch lengths by EvoVGM_K80. Rows correspond to the number of sequences. Columns correspond to the alignment length. Datasets simulated with K80 substitution model.

$N \rightarrow$	100		1000		5000	
M	DIST	CORR	DIST	CORR	DIST	CORR
3	0.133	0.948	0.171	0.975	0.074	0.975
4	0.184	0.855	0.093	0.996	0.049	0.992
5	0.180	0.835	0.062	0.990	0.073	0.999

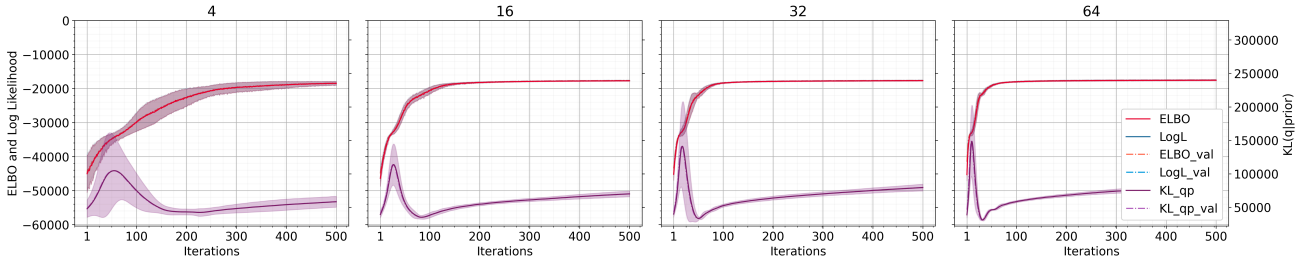
Table A.1: Euclidean distance (DIST) and Pearson correlation coefficient (CORR) between actual and estimated branch lengths by EvoVGM_GTR. Rows correspond to the number of sequences. Columns correspond to the alignment length. Datasets simulated with GTR substitution model.

$N \rightarrow$	100		1000		5000	
M	DIST	CORR	DIST	CORR	DIST	CORR
3	0.300	0.984	0.090	0.980	0.097	0.986
4	0.076	0.994	0.086	0.998	0.085	0.992
5	0.081	0.986	0.069	0.995	0.116	0.962

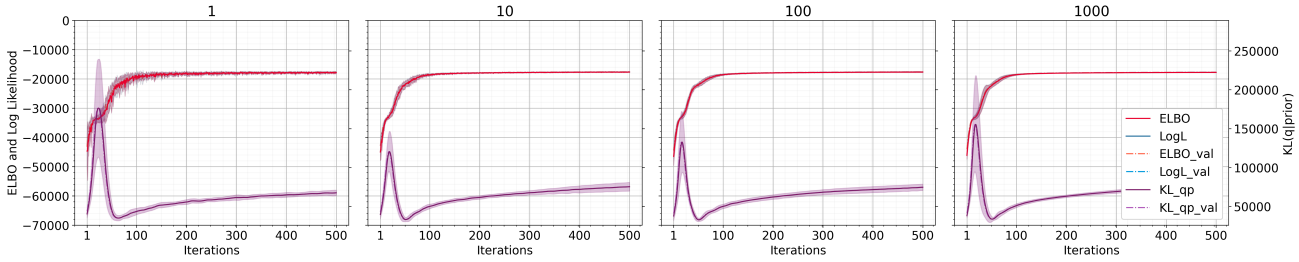
A.3.2. HYPER-PARAMETERS EVALUATION OF EVOVGM MODELS



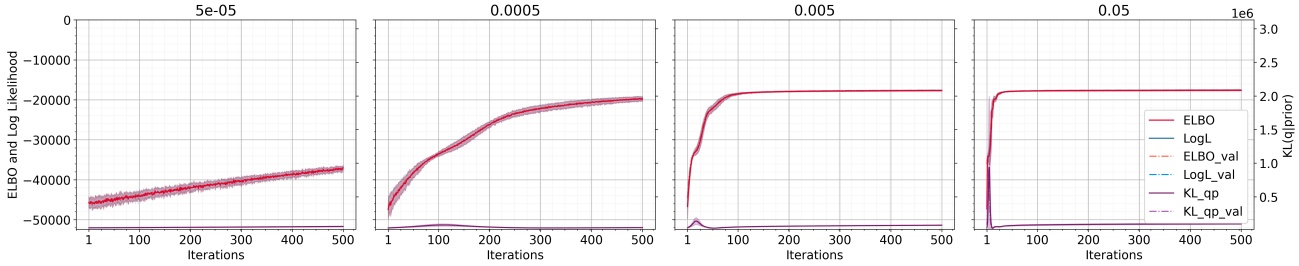
(a) α_{KL}



(b) Hidden size

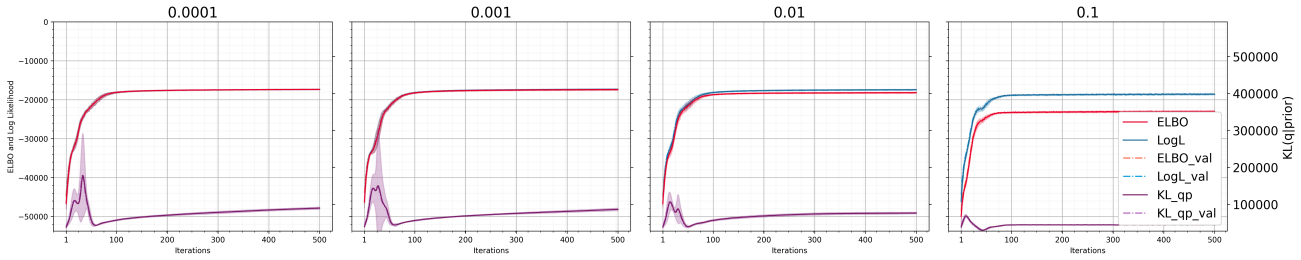


(c) Sample size

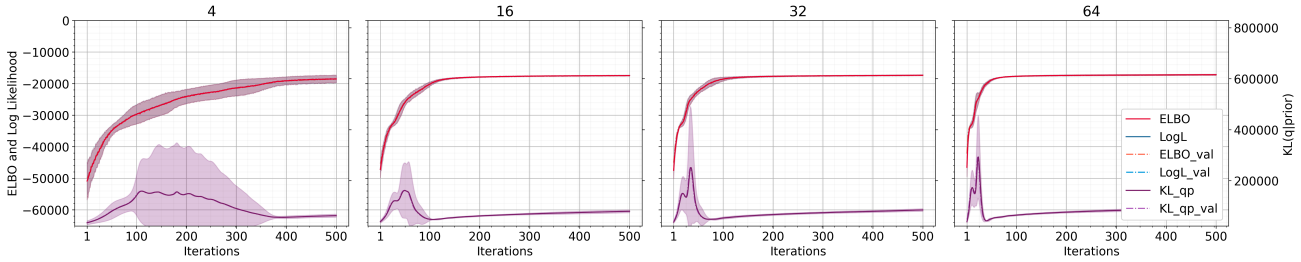


(d) Learning rate

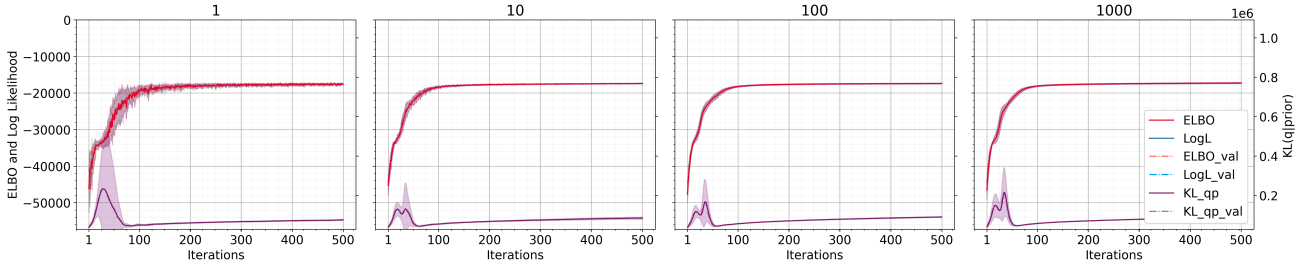
Figure A.2: Performance of the EvoVGM_JC69 model for multiple settings of (α_{KL}, the size of the hidden layers of the neural networks of the encoders, the sample size and the learning rate. The JC69 substitution model was used to simulate training and validation alignments of five sequences with a length of 5000 bp. We show the trend of the performance over the first 500 iterations of the training.



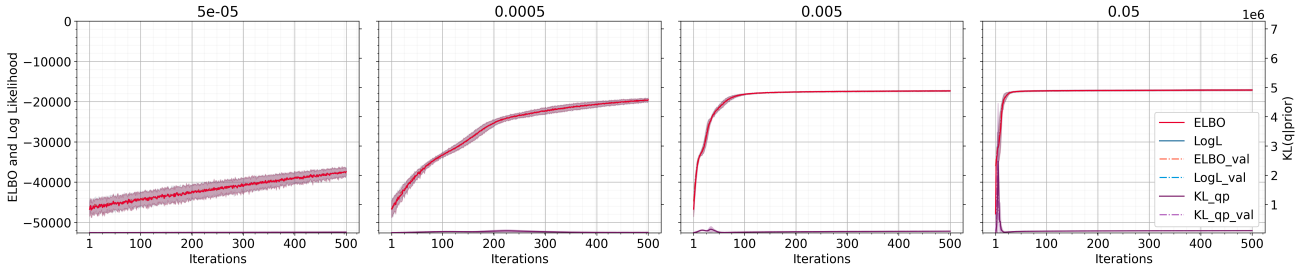
(a) α_{KL}



(b) Hidden size

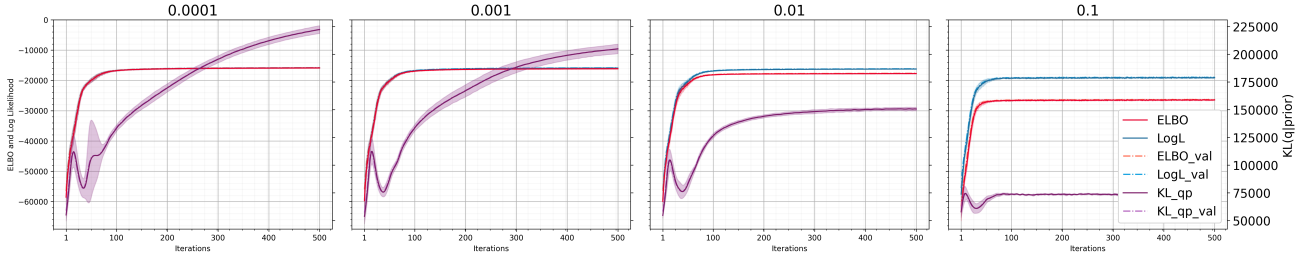


(c) Sample size

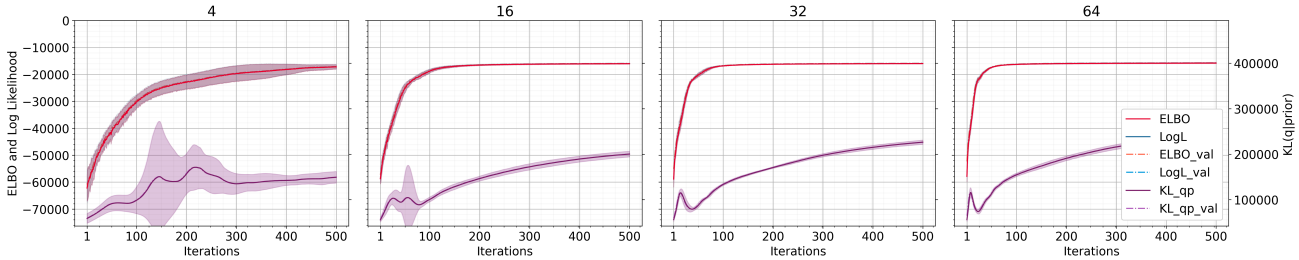


(d) Learning rate

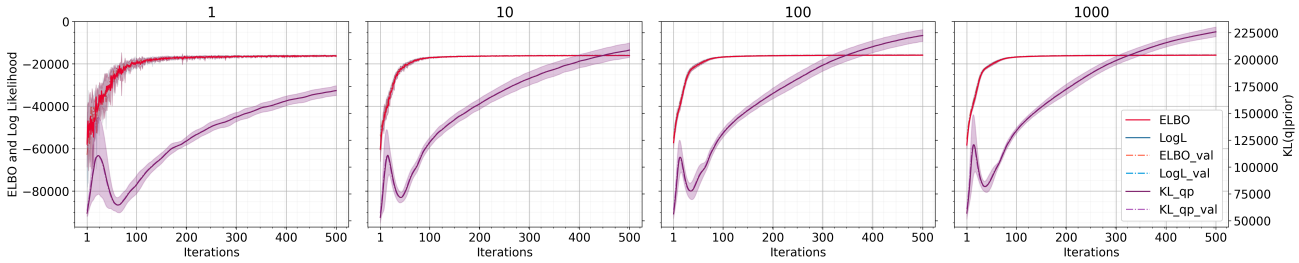
Figure A.3: Performance of the EvoVGM_K80 model for multiple settings of (α_{KL}, the size of the hidden layers of the neural networks of the encoders, the sample size and the learning rate. The K80 substitution model was used to simulate training and validation alignments of five sequences with a length of 5000 bp. We show the trend of the performance over the first 500 iterations of the training.



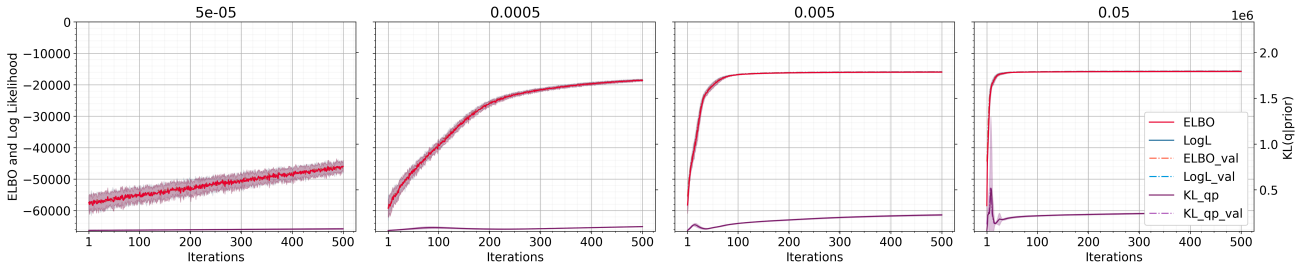
(a) α_{KL}



(b) Hidden size



(c) Sample size



(d) Learning rate

Figure A.4: Performance of the EvoVGM_GTR model for multiple settings of (α_{KL}, the size of the hidden layers of the neural networks of the encoders, the sample size and the learning rate. The GTR substitution model was used to simulate training and validation alignments of five sequences with a length of 5000 bp. We show the trend of the performance over the first 500 iterations of the training.