Ancestral protein sequence reconstruction using a tree-structured Ornstein-Uhlenbeck variational autoencoder

Anonymous authors
Paper under double-blind review

Abstract

We introduce a deep generative model for representation learning of biological sequences that, unlike existing models, explicitly represents the evolutionary process. The model makes use of a tree-structured Ornstein-Uhlenbeck process, obtained from a given phylogenetic tree, as an informative prior for a variational autoencoder. We show the model performs well on the task of ancestral sequence reconstruction of single protein families. Our results and ablation studies indicate that the explicit representation of evolution using a suitable tree-structured prior has the potential to improve representation learning of biological sequences considerably. Finally, we briefly discuss extensions of the model to genomic-scale data sets and the case of a latent phylogenetic tree.

1 Introduction

Representation learning of biological sequences is important for data exploration and downstream tasks such as protein design (Detlefsen et al., 2020; Alley et al., 2019). Deep generative models such as variational autoencoders (VAEs) (Kingma & Welling, 2013; 2019) have been especially useful for this purpose (Riesselman et al., 2018; Greener et al., 2018). However, current models do not take evolutionary information fully into account, i.e., by relating the sequences belonging to a protein family in a phylogenetic tree and incorporating parameterized evolutionary models (Durbin et al., 1998). To address this problem, we replace the standard multivariate Gaussian prior of a conventional VAE with a tree-structured prior that takes into account a given evolutionary tree. We propose a prior based on the Ornstein-Uhlenbeck Gaussian process on a tree (Hansen, 1997; Jones & Moriarty, 2013). We apply the model to a classic problem in phylogenetics, namely the inference of ancestral sequences.

Ancestral sequence reconstruction (ASR), i.e., the inference of ancestral sequences given their descendants or leaf sequences (Pauling et al., 1963; Yang et al., 1995; Koshi & Goldstein, 1996; Joy et al., 2016; Hochberg & Thornton, 2017; Selberg et al., 2021), has important applications including protein engineering (Cole & Gaucher, 2011; Spence et al., 2021), modeling tumour evolution (El-Kebir et al., 2015), evaluating virus diversity and vaccine design (Gaschen et al., 2002), understanding drug mechanisms (Wilson et al., 2015) and reconstructing ancient proteins *in vitro* (Chang et al., 2002; Wilson et al., 2015; Hochberg & Thornton, 2017).

As input, we assume a set of n_S known, aligned leaf sequences and their phylogenetic tree. The task we want to address is the inference of the $n_A = n_S - 1$ unknown, ancestral sequences (Joy et al., 2016). We show that our probabilistic model, called Draupnir, is about on par with or better than the accuracy of established ASR methods for a standard experimentally-derived data set (Alieva et al., 2008; Randall et al., 2016) and several simulated data sets.

The paper is organised as follows. In Background, we briefly discuss evolution of biological sequences, ancestral sequence reconstruction and the tree-structured Ornstein-Uhlenbeck process. In Related Work, we discuss deep generative models of biological sequences. In Methods, we describe the Draupnir model, the inference of ancestral sequences and

the setup of the benchmarking experiments. In Results, we discuss the quality of the latent representations, compare the accuracy of Draupnir with state-of-the-art phylogenetic methods for ASR, and present the results of ablation experiments. We end with a brief discussion of future work, including extending the method to genomic-scale data sets and the case of a latent phylogenetic tree.

2 Background

2.1 Protein sequences and evolution

Biological molecules such as proteins and nucleic acids (DNA, RNA) can be characterised by sequences of characters from an alphabet of size n_C, where typically $n_C = 21$ for proteins and $n_C = 5$ for nucleic acids (Durbin et al., 1998). These alphabets include one character that represents a gap, which is useful in aligning related sequences in a multiple sequence alignment (MSA). In the course of evolution, mutations arise that cause changes in these sequences, including character substitutions, deletions and insertions. A set of n_S known, homologous, extant sequences and their evolutionary relationships are naturally represented as n_S leaf nodes in a binary tree or phylogeny, where the n_A internal or ancestral nodes represent unknown, ancestral sequences (Joy et al., 2016). Among the internal nodes, the root node is the most ancient node.

Edges between two nodes in the tree are labelled by positive real numbers that represent the time difference or the amount of change between them. Such a labelled binary tree naturally defines a $(n_S + n_A) \times (n_S + n_A)$ matrix containing the pairwise distances between all nodes, called the *patristic distance matrix*, $\mathbf{T}$. In the context of biological sequences, the field of phylogenetics is concerned with the inference of the tree topology, the labels of the tree's edges and the composition of the ancestral sequences, making use of methods based on heuristics (such as maximum parsimony) or probabilistic, evolutionary models (Joy et al., 2016).

2.2 Ancestral protein reconstruction

The ASR problem amounts to inferring the composition of the n_A ancestral sequences from the n_S extant sequences, making use of a tractable model of evolution (Joy et al., 2016). Typically, the phylogenetic tree that relates the sequences is assumed known. Standard methods to do this typically assume independent (factorized) evolution of the characters in the sequence, which is a computationally convenient but unrealistic assumption. For example, in proteins, amino acids are involved in an intricate 3-dimensional network of interactions that can lead to strong dependencies between amino acids far part in the sequence. This phenomenon is called *epistasis* (Hochberg & Thornton, 2017), which requires coevolutionary models that go beyond the factorized assumption. Nonetheless, it has been possible to infer ancestral sequences and subsequently resurrect functional ancient, ancestral proteins *in vitro* (Hochberg & Thornton, 2017). The aim of this work is to go beyond the assumption of independent, factorized evolution by using a model of evolution that features continuous, latent vector representations of the protein sequences. This allows us to formulate the ASR problem in the context of a deep generative model.

2.3 The Ornstein-Uhlenbeck process on a phylogenetic tree

Typically, ASR of biological sequences is done using factorised evolutionary models that represent substitutions, insertions and deletions of the discrete characters in the sequences (Joy et al., 2016). In contrast, Draupnir aims to model the evolution of latent, continuous representations or underlying traits of the sequences. A simple diffusive process allowing for an equilibrium distribution is the Ornstein-Uhlenbeck (OU) process (Hansen, 1997; Jones & Moriarty, 2013). As the OU process is a Gaussian process, it has a Gaussian equilibrium distribution, as well as Gaussian marginal distributions.

We use an OU process on a phylogenetic tree (TOU process) (Hansen, 1997; Jones & Moriarty, 2013) to put the latent representations under the control of a parameterized

evolutionary model. Apart from the mean, which for our purposes can be assumed to be zero, the TOU process has three parameters: the variation unattributable to the phylogeny or the intensity of specific variation σ_n, the characteristic length scale of the evolutionary dynamics, λ and the intensity of inherited variation, σ_f. The covariance function for the corresponding multivariate Gaussian distribution is then given by (Hadjipantelis et al., 2012; Jones & Moriarty, 2013),

$$\Sigma_{k,l} = \sigma_f^2 \exp\left(\frac{-T_{k,l}}{\lambda}\right) + \sigma_n^2 \delta_{k,l} \tag{1}$$

where $T_{k,l}$ is the patristic distance between nodes k and l in the tree, and the Kronecker delta $\delta_{k,l} = 1$ if $k = l$, and 0 otherwise.

The TOU process and related diffusive processes on trees are well-established evolutionary models that have been used to model the evolution of continuous traits, such as body mass or length (Joy et al., 2016). For example, Lartillot (2014) proposes a phylogenetic Kalman filter for ancestral trait reconstruction of low-dimensional, continuous traits; Tolkoff et al. (2018) propose phylogenetic factor analysis, in which a latent variable under the control of a small number independent univariate Brownian diffusion processes is related to observed traits through a loading matrix; Horta et al. (2021) use a multivariate TOU process and Markov chain Monte Carlo to model both continuous traits and sequences of discrete characters. To represent the latter, they make use of a pairwise Potts model.

3 Related work

3.1 Representation learning of biological sequences

A VAE (Kingma & Welling, 2013; 2019) is a probabilistic, generative model featuring latent vectors or representations, $\{\mathbf{z}\}_{n=1}^N$, that are independently sampled from a prior distribution, $\mathbf{z}_n \sim \pi(\mathbf{z}_n)$. The latent vectors are passed to a neural network (the *decoder*) with parameters θ, leading to a likelihood, $\mathbf{x}_n \sim p_\theta(\mathbf{x}_n \mid \mathrm{NN}_\theta(\mathbf{z}_n))$, for the data, $\{\mathbf{x}\}_{n=1}^N$. The prior is typically a standard multivariate Gaussian distribution, but other priors have been used, such as distributions on the Poincaré ball to recover hierarchical structures (Mathieu et al., 2019). The posterior distribution $p(\mathbf{z}_n \mid \mathbf{x}_n)$ is intractable, but can be approximated with a variational distribution or *guide*, $q_\phi(\mathbf{z}_n \mid \mathrm{NN}_\phi(\mathbf{x}_n))$, involving a second neural network (the *encoder*). Point estimates of the parameters θ and ϕ are obtained by maximizing the *evidence lower bound* (ELBO),

$$\mathcal{L}_{\theta,\phi}(\mathbf{x}) = \mathbb{E}_q\left[\log\left(\frac{p_\theta(\mathbf{x},\mathbf{z})}{q_\phi(\mathbf{z} \mid \mathbf{x})}\right)\right],$$

using stochastic gradient ascent (Hoffman et al., 2013).

VAEs are increasingly used for representation learning of biological sequences (Detlefsen et al., 2020). Riesselman et al. (2018) use a VAE with biologically motivated priors to evaluate the stability of mutants and to explore new regions of sequence space. Greener et al. (2018) use autoencoders to design metal-binding proteins and novel protein folds. Ding et al. (2019) show that the latent representations obtained with a VAE can capture evolutionary relationships between sequences. The above models do not represent the phylogenetic tree explicitly, but typically aim to condition on some evolutionary information by training on pre-computed MSAs - an approach that has been called *evo-tuning* (Rao et al., 2019; Detlefsen et al., 2020). Hawkins-Hooker et al. (2021) use a VAE with a convolutional encoder and decoder, combining upsampling and autoregression, without relying on a MSA.

The above models assume that the latent vectors factor independently, which is computationally convenient but unrealistic if the sequences are related to each other in a phylogeny. A more realistic approach thus uses a prior $\pi(\{\mathbf{z}\}_{n=1}^N \mid \tau, \kappa)\pi(\kappa)$ that conditions the latent vectors on a given phylogenetic tree, τ, and an evolutionary model with latent parameters, κ. Because the latent vectors do not factor independently anymore, mini-batch training can include the sequences but not the latent vectors, which limits the possible size

of the data sets. Nonetheless, we show here that such a model is both computationally tractable and practically useful for realistic data sets concerning single protein families.

4 Methods

4.1 The Draupnir model

The pseudocode of the Draupnir model is given in Algorithm 1; Figure 1 shows the corresponding graphical model. A summary of the variables, their dimensions and the notation is given in Tables 1 and 2 in Appendix A.1.

As inputs, we assume (a) a set of n_S aligned sequences, each with length n_L, organized in the $n_S \times n_L$ matrix, $\mathbf{S}$, and (b) information on their phylogenetic tree in the form of their $(n_S + n_A) \times (n_S + n_A)$ patristic distance matrix, $\mathbf{T}$.

The latent matrix $\mathbf{Z}$ of the model is a matrix with n_S rows (one for each leaf sequence) and n_Z columns, where n_Z is the size of the latent representation of the sequences. In all experiments, $n_Z = 30$. Each column of $\mathbf{Z}$ (with n_S elements) is sampled from a univariate OU process on the phylogenetic tree, representing the evolution of a hidden trait underlying the sequences along the tree. Each row of $\mathbf{Z}$ corresponds to the latent vector of a standard VAE. However, unlike a standard VAE, the latent vectors do not factorize independently.

For each of the n_Z TOU processes, the parameters of the TOU process, corresponding to κ in Section 3.1, are sampled from a suitable prior distribution. The TOU process has three parameters: σ_f, λ and σ_n. As we use n_Z tree OU processes (one for each column of $\mathbf{Z}$), we need to sample n_Z sets of these three parameters.

For each of the three parameters, $\sigma_f, \lambda, \sigma_n$, we sample a hyperparameter $(\alpha_1, \alpha_2, \alpha_3)$ from a half-normal distribution with scale parameter equal to one. These hyperparameters serve as scale parameter for the half-normal priors over σ_f, λ and σ_n. Given the parameters of the TOU process obtained from the prior described above, an $n_S \times n_S$ covariance matrix can be calculated based on the patristic distance matrix of the leaves, $\mathbf{T}^{(S,S)}$. We need one such covariance matrix for each of the n_Z columns of the matrix of latent representations, $\mathbf{Z}$. The element k, l, with $k, l \in 1, ..., n_S$, of covariance matrix h, with $h \in 1, ..., n_Z$, is given by (Hadjipantelis et al., 2012; Jones & Moriarty, 2013),

$$C_{h,k,l} = \sigma^2_{f,h} \exp(-T_{k,l}/\lambda_h) + \sigma^2_{n,h} \delta_{k,l}. \tag{2}$$

As decoder, we use a bidirectional gated recurrent unit (GRU, Cho et al. (2014)) with length equal to the alignment length, n_L. The input at each position i of the GRU for sequence $\mathbf{S}_{k,:}$ is a concatenated vector, consisting of the latent vector $\mathbf{Z}_{k,:}$ representing sequence k, and the BLOSUM embedding $\mathbf{E}_{i,:}$, which is the result of applying a fully connected neural network, $\mathrm{NN}^{(1)}_\theta$, to the BLOSUM vector $\mathbf{V}_{i,:}$ describing the amino acid preferences at position i in the MSA (see Section 4.2). For each of the n_S sequences and for each position i, the GRU states are mapped to a logit vector that specifies the probabilities of the n_C characters using another fully connected neural network, $\mathrm{NN}^{(2)}_\theta$. The architecture of the networks is shown in Appendix A.2.

4.2 BLOSUM embeddings

A BLOSUM matrix $\mathbf{B}$ is an $n_C \times n_C$ substitution matrix used for sequence alignment, where each row contains the log-odds scores of replacing a given character with any of the other characters (Henikoff & Henikoff, 1992). Each position (column) in the MSA is represented by the average of the BLOSUM vectors of the characters in that column (see Algorithm 2). The averaged BLOSUM vectors only need to be precomputed once. In the model, the BLOSUM vectors are processed into BLOSUM embeddings by a neural network to provide position-specific information on the MSA, while the latent variables provide sequence-specific information (see Algorithm 2).

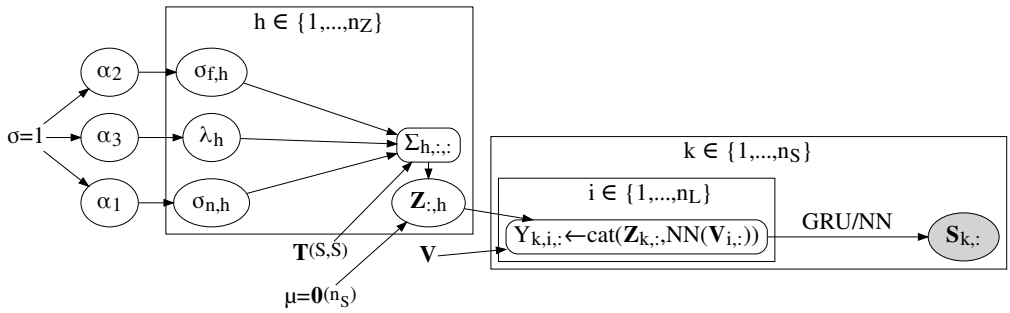


Figure 1: Draupnir as a graphical model. For notation and information on variables and their dimensions see Tables 1 and 2 in Appendix A.1 and A.2. Random variables are shown as ellipses, while deterministic quantities are shown as rounded boxes and observed random variables are shown as shaded ellipses. Parameters of priors and other given quantities are shown without boxes. The model contains three plates, respectively corresponding to the number of dimensions of the leaf sequence-specific latent vector $\mathbf{Z}_{\mathbf{k},:}$ (n_Z=30), the number of leaf sequences (n_S) and the alignment length (n_L). "cat" indicates the concatenation of two vectors. α are the hyperparameters and $\sigma_n, \sigma_f, \lambda$ are the parameters of the TOU processes. $\mathbf{\Sigma}_{h,:,:}$ is the covariance matrix that is used to sample component h of the n_S latent vectors from a multivariate Gaussian distribution (with mean $\mathbf{0}^{(n_S)}$). $\mathbf{T}^{(S,S)}$ is the patristic distance matrix containing the distances between the leaf sequences. $\mathbf{Y}_{k,:}$ is the input vector for the GRU that produces the likelihood parameters for leaf sequence k, $\mathbf{S}_{k,:}$. $\mathbf{V}$ represents the MSA as an $n_L \times n_C$ matrix of averaged BLOSUM vectors. NN denotes a fully connected neural network.

4.3 Model implementation and training

Draupnir was implemented in the deep probabilistic programming language Pyro (Bingham et al., 2019) and trained using stochastic variational inference with Pyro's *AutoDelta* guide, optimizing the ELBO (Kingma & Welling, 2019) and resulting in maximum a posteriori (MAP) estimates for all parameters. We use Adam (Kingma & Ba, 2014) as the optimizer using the default values. From the MAP estimates, we can sample the latent representations of the ancestral nodes (see Section 4.4). These latent representations are then subsequently decoded to their respective ancestral sequences. We also use a custom guide to calculate a variational posterior (see Appendix A.6). Training details can be found in Appendix A.4. All programs were executed on an Intel(R) Xeon(R) Gold 6136 CPU @ 3.00GHz machine with a Quadro RTX 6000 GPU.

4.4 Inference of the ancestral sequences

In this section, for ease of notation, let $\mathbf{z} \equiv \mathbf{Z}^{(A,S)}_{:,h}$ denote one of the $h \in \{1, \ldots, n_z\}$ columns of the latent representation matrix for both ancestral and leaf sequences, $\mathbf{Z}^{(A,S)}$. First, note that $\mathbf{z}$ can be partitioned as $\mathbf{z} = (\mathbf{z}^{(S)}, \mathbf{z}^{(A)})$, where $\mathbf{z}^{(S)}$ and $\mathbf{z}^{(A)}$ denote the latent representations of the leaf and ancestral sequences, respectively. The prior $p(\mathbf{z})$ is a multivariate Gaussian distribution with parameters,

$$\mu = \begin{pmatrix} \mu^{(S)} \\ \mu^{(A)} \end{pmatrix} = \begin{pmatrix} \mathbf{0}^{(n_Z)} \\ \mathbf{0}^{(n_A)} \end{pmatrix}, \mathbf{\Sigma} = \begin{pmatrix} \mathbf{\Sigma}^{(S,S)} & \mathbf{\Sigma}^{(S,A)} \\ \mathbf{\Sigma}^{(A,S)} & \mathbf{\Sigma}^{(A,A)} \end{pmatrix}, \Lambda = \begin{pmatrix} \Lambda^{(S,S)} & \Lambda^{(S,A)} \\ \Lambda^{(A,S)} & \Lambda^{(A,S)} \end{pmatrix},$$

where $\Lambda \equiv \mathbf{\Sigma}^{-1}$. The covariance matrices $\mathbf{\Sigma}^{(S,S)}, \mathbf{\Sigma}^{(A,A)}, \mathbf{\Sigma}^{(A,S)} \equiv \left(\mathbf{\Sigma}^{(S,A)}\right)^T$ are respectively obtained from the distance matrices concerning distances within the leaves, within the ancestors and between the ancestors and the leaves, $\mathbf{T}^{(S,S)}, \mathbf{T}^{(A,A)}$ and $\mathbf{T}^{(A,S)}$, and the TOU process parameters (see Eq. 2).

Algorithm 1 The Draupnir model

Require: Multiple sequence alignment $\mathbf{S}$, patristic distance matrix $\mathbf{T}$

for $j \in \{1, 2, 3\}$ **do** ▷ Hyperpriors over the TOU process parameters
  $\alpha_j \sim \mathcal{HN}(1)$

for $h \in \{1, ..., n_Z\}$ **do** ▷ Priors over the parameters of the n_Z TOU processes
  $\sigma_{f,h} \sim \mathcal{HN}(\alpha_0)$
  $\sigma_{n,h} \sim \mathcal{HN}(\alpha_1)$
  $\lambda_h \sim \mathcal{HN}(\alpha_2)$

for $h \in \{1, ..., n_Z\}$ **do** ▷ Kernels for the n_Z TOU processes
  for $k, l \in \{1, ..., n_S\}$ **do**
    $C_{h,k,l} \leftarrow \sigma_{f,h}^2 \exp(-T_{k,l}^{(S,S)}/\lambda_h) + \sigma_{n,h}^2 \delta_{k,l}$

for $h \in [1, \ldots, n_Z]$ **do** ▷ Prior over tree-strucured latent matrix $\mathbf{Z}$
  $\mathbf{Z}_{:,h} \sim \mathcal{MVN}(\mathbf{0}^{(n_S)}, \mathbf{C}_{h,:,:})$

for $i \in [1, \ldots, n_L]$ **do** ▷ BLOSUM embeddings
  $\mathbf{E}_{i,:} \leftarrow \mathrm{NN}_\theta^{(1)}(\mathbf{V}_{i,:})$

for $k \in [1, \ldots, n_S]$ **do** ▷ Input vector $\mathbf{Y}$ for GRU
  for $i \in [1, \ldots, n_L]$ **do**
    $\mathbf{Y}_{k,i,:} \leftarrow \mathrm{cat}(\mathbf{Z}_{k,:}, \mathbf{E}_{i,:})$ ▷ Concatenate sequence- and position-specific vectors

for $k \in [1, \ldots, n_S]$ **do** ▷ Likelihood parameters (logits) $\mathbf{L}$ from GRU
  $\mathbf{H}_{k,:,:} \leftarrow \mathrm{GRU}_\theta(\mathbf{Y}_{k,:,:})$ ▷ Bidirectional GRU states
  for $i \in [1, \ldots, n_L]$ **do**
    $\mathbf{L}_{k,i,:} \leftarrow \mathrm{NN}_\theta^{(2)}(\mathbf{H}_{k,i,:})$
    $S_{k,i} \sim \mathrm{Categorical}(\mathbf{L}_{k,i,:})$ ▷ Likelihood at position i in sequence k

Algorithm 2 Pre-computation of averaged BLOSUM vectors

Require: Multiple sequence alignment $\mathbf{S}$

$\mathbf{V} \leftarrow \mathbf{0}^{(n_L \times n_c)}$ ▷ Initialize BLOSUM weighted average $\mathbf{V}$
for $i \in [1, \ldots, n_L]$ **do** ▷ Position in sequence alignment
  for $k \in [1, \ldots, n_S]$ **do** ▷ Index of leaf sequence
    $r \leftarrow S_{k,i}$ ▷ Character at position i in leaf sequence k
    $\mathbf{V}_{i,:} \leftarrow \mathbf{V}_{i,:} + \mathbf{B}_{r,:}$ ▷ Add BLOSUM vector corresponding to the amino acid
  $\mathbf{V}_{\mathbf{i},:} \leftarrow \frac{1}{n_S} \mathbf{V}_{\mathbf{i},:}$ ▷ Average

As $p(\mathbf{z})$ is a multivariate Gaussian distribution, we can easily obtain the conditional distribution of $\mathbf{z}^{(A)}$ given $\mathbf{z}^{(S)}$ as follows (see Bishop (2006), page 689),

$$p(\mathbf{z}) = \mathcal{MVN}\left(\mathbf{z} \mid \mathbf{\Sigma}\right) \Rightarrow p\left(\mathbf{z}^{(A)} \mid \mathbf{z}^{(S)}\right) = \mathcal{MVN}\left(\mathbf{z}^{(A)} \mid \mu_{A|S}, \left(\Lambda^{(A,A)}\right)^{-1}\right),$$

with

$$\mu_{A|S} = \mu_A - \left(\Lambda^{(A,A)}\right)^{-1} \Lambda^{(A,S)} \left(\mathbf{z}^{(S)} - \mu^{(S)}\right) = -\left(\Lambda^{(A,A)}\right)^{-1} \Lambda^{(A,S)} \mathbf{z}^{(S)}.$$

As $\mu_{A|S}$ corresponds to the MAP of $\mathbf{z}^{(A)}$ for any given values of $\mathbf{z}^{(S)}$ and the TOU process parameters when the ancestral sequences are not observed, the MAP estimate of the latent

representation of the ancestral sequences is given by,

$$\mathbf{z}^{(A),\mathrm{MAP}} = -\left(\Lambda^{(A,A)}\right)^{-1} \Lambda^{(A,S)} \mathbf{z}^{(S),\mathrm{MAP}}.$$

In addition, a Gaussian approximation of the posterior of $\mathbf{z}^{(A)}$ based on the MAP estimates is given by,

$$\mathbf{z}^{(A)} \sim \mathcal{MVN}\left(\mathbf{z}^{(A)} \mid \mathbf{z}^{(A),\mathrm{MAP}}, \left(\Lambda^{(A,A)}\right)^{-1}\right). \tag{3}$$

The reconstructed ancestral sequences are subsequently obtained by applying the GRU decoder to the MAP estimates of their latent representations, as explained in Section 4.1.

4.5 Benchmarking

In order to assess the accuracy of the ASR we benchmark Draupnir against state-of-the-art phylogenetic methods. We selected methods that perform ASR using a given tree topology and given patristic distances, including one Bayesian method (PhyloBayes, Lartillot et al. (2013)) and three maximum likelihood based methods (PAML, Yang (2007); FastML, Ashkenazy et al. (2012); and IQTree, Nguyen et al. (2015)). We apply the ASR methods to both protein sequences, and to their corresponding DNA sequences followed by subsequent translation to protein sequences. For Draupnir, we use the protein sequences, which is the harder problem. We use nine data sets with different numbers of leaves (from 19 to 800), different alignment lengths (from 63 to 477) and with or without gaps (8 and 1, respectively). The data sets include six simulated data sets generated using the software EvolveAGene (Hall, 2016) and three data sets with experimentally determined ancestral sequences. Note that the simulated data sets were obtained from factored evolution models. We included a large data set with 800 leaves. For prediction with Draupnir, we either use the most likely sequence, sample sequences from the MAP estimates using Equation 3 (we report the average identity of 50 samples)) or sample sequences from the variational posterior using an amortised guide (Appendix A.6). Details on the data sets and training can be found in Appendix A.3 and A.4, respectively.

5 Results

5.1 Latent representations and coevolution

In order to inspect the quality of the latent representations of the ancestral and leaf sequences, we use the β-lactamase family with 32 leaf sequences. We visualize t-SNE projections (Van der Maaten & Hinton, 2008) and compare the results with the structure of the phylogenetic tree (see Figure 2). The result indicates that the latent representations represent the structure of the tree and its different clades (subtrees) well, indicating that the TOU process performs well as an informative prior on evolutionary relationships. In Appendix A.6, we show how marginalizing over the latent representations allows Draupnir to model coevolution among sites.

Figure 2: Results for the β-lactamase family with 32 leaves. *Left*: t-SNE projection of the latent representations of the ancestral and leaf nodes. *Right*: The phylogenetic tree. Both plots are coloured according to clade membership.

5.2 Benchmarking Results

In Figure 3, we compare the accuracy of Draupnir with state-of-the-art ASR methods by plotting the average percent identity between the ancestral sequences as reconstructed by Draupnir and the true sequences. The true ancestral sequences were either experimentally determined or simulated. The description and origin of the data sets can be found in Appendix A.3. Tables with benchmark results are shown in Appendix A.5.

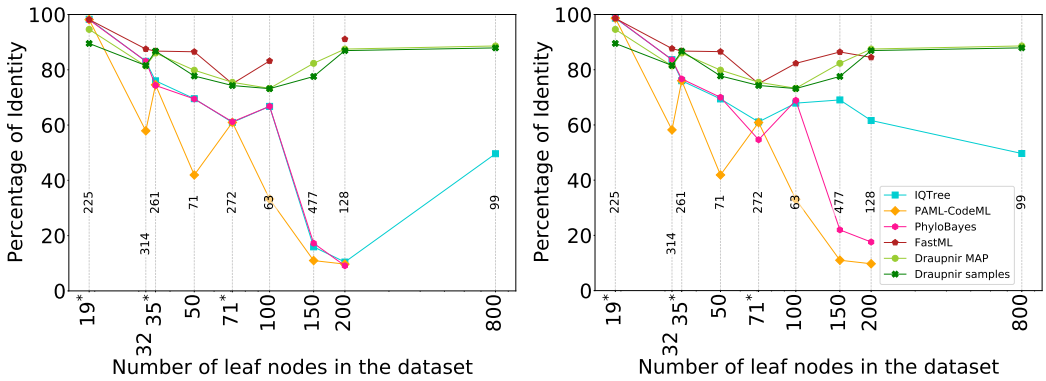
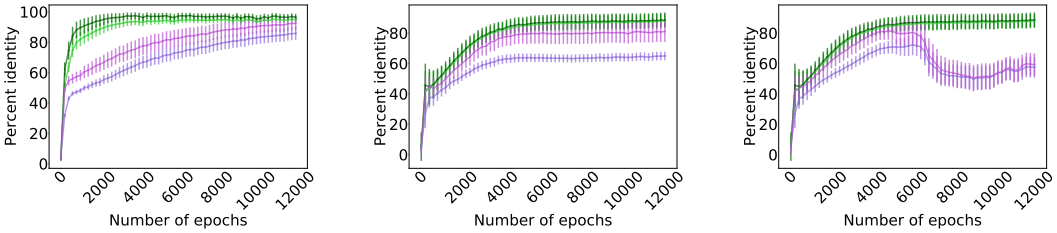


Figure 3: Comparison of the average percentage identity (y-axis) between predicted and true ancestral sequences for Draupnir and ASR methods for different number of leaves (x-axis; experimental data sets are indicated with an asterisk). Missing points indicate that the ASR method failed to produce results on the given hardware. The alignment size is shown on the dotted lines. *Left*: Comparison with ASR methods using protein sequences. *Right*: Comparison using DNA sequences, subsequently translated to protein sequences. The standard deviations can be found in Appendix A.5. For the benchmarking settings refer to Appendix A.8

5.3 Ablation studies

In the first ablation study, we investigate the influence of the BLOSUM embeddings by removing them as input to the GRU. Overall, the absence of the BLOSUM embeddings slows down convergence and sometimes make the learning process unstable, but ultimately does not strongly affect accuracy (see Figure 5).

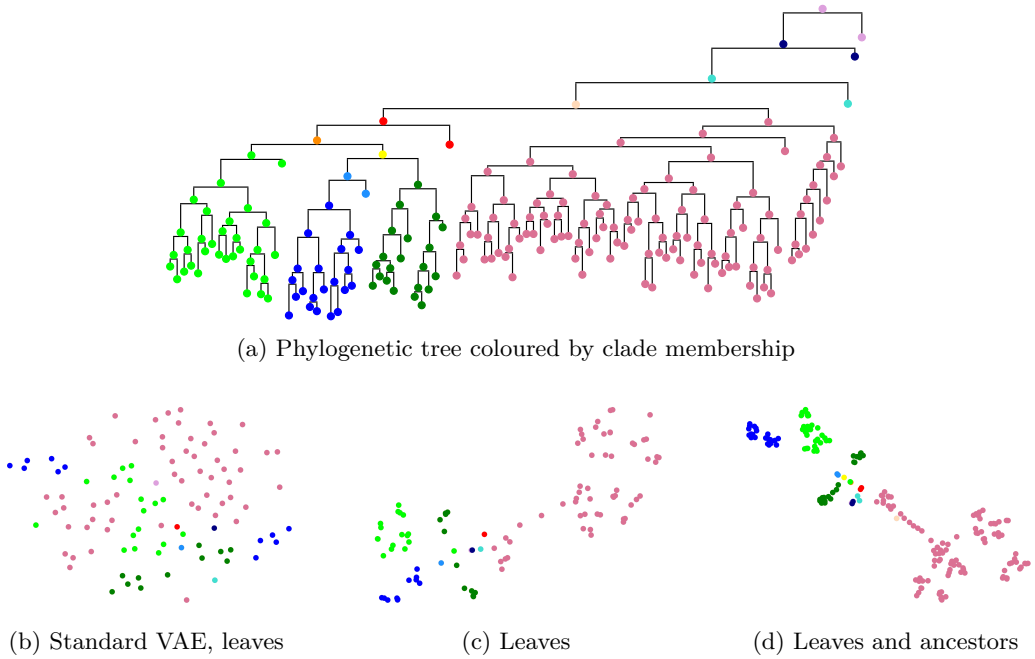


(a) Leaves, stable case (b) Ancestors, stable case (c) Ancestors, unstable case.

Figure 5: BLOSUM embedding ablation study for the 800 leaves data set. For every 100 training epochs, the average percent identity and standard deviation is plotted for all leaves (training set) or ancestors (test set), respectively. The results obtained with the BLOSUM embedding are shown in dark green (MAP) and light green (sampled, see Equation 3). The results without the BLOSUM embeddings are shown in pink (MAP) and purple (sampled).

In the second ablation study, we investigate the influence of the tree-structured prior by comparing with a standard VAE with a Gaussian prior. We do this by using diagonal unit covariance matrices for each of the n_Z columns of the latent matrix $\mathbf{Z}$. The rest of the model was identical. We then compare the latent representations obtained for the leaf nodes. The

results (see Figure 7) indicate that the standard VAE is not capable of reconstructing the evolutionary relationships well: sequences belonging to the same clade often end up far apart in latent space. This indicates that the influence of the tree-structures prior is substantial. A quantitative analysis of this ablation study can be found in Appendix A.8.



(a) Phylogenetic tree coloured by clade membership

(b) Standard VAE, leaves

(c) Leaves

(d) Leaves and ancestors

Figure 7: Bottom: t-SNE projections of the latent representations for the SRC-Kinase SH3 domain with 100 leaves, obtained from a standard VAE (left) and Draupnir (center and right), colored by clade. Note that only the latter model can be used to infer the latent representations of the ancestral sequences. Top: the corresponding tree.

6 Discussion and future work

Draupnir demonstrates the potential value of incorporating evolutionary information and evolutionary models explicitly in deep generative models for representation learning of biological sequences. We point out that it is possible to extend the model with additional information beyond sequences, for example backbone angles describing protein structure (Golden et al., 2017) or measurements of protein stability. In future work, extending the model to genomic-size data can be done using inducing points for Gaussian processes, as explored in Jazbec et al. (2021) and Vikram et al. (2019). The case of a latent phylogenetic tree can be addressed using a coalescent point process prior (Lambert & Stadler, 2013; Vikram et al., 2019). Finally, in the large data case, the current simple network architectures can be improved with more expressive compositions such as an MSA transformer (Rao et al., 2021) or a deconvolutional model for sequences (Hawkins-Hooker et al., 2021). Finally, we point out that the learned parameters of the TOU processes might offer interpretable information on the evolutionary process.

References

Naila O Alieva, Karen A Konzen, Steven F Field, Ella A Meleshkevitch, Marguerite E Hunt, Victor Beltran-Ramirez, David J Miller, Jörg Wiedenmann, Anya Salih, and Mikhail V Matz. Diversity and evolution of coral fluorescent proteins. *PLoS ONE*, 3(7):e2680, 2008.

Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.

Haim Ashkenazy, Osnat Penn, Adi Doron-Faigenboim, Ofir Cohen, Gina Cannarozzi, Oren Zomer, and Tal Pupko. FastML: a web server for probabilistic reconstruction of ancestral sequences. *Nucleic acids research*, (W1):W580–W584, 2012.

Ahmet Bakan, Anindita Dutta, Wenzhi Mao, Ying Liu, Chakra Chennubhotla, Timothy R Lezon, and Ivet Bahar. Evol and prody for bridging protein sequence evolution and structural dynamics. *Bioinformatics*, 30:2681–2683, 2014.

Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep universal probabilistic programming. *The journal of machine learning research*, 20(1):973–978, 2019.

Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

Belinda SW Chang, Karolina Jönsson, Manija A Kazmi, Michael J Donoghue, and Thomas P Sakmar. Recreating a functional ancestral archosaur visual pigment. *Molecular biology and evolution*, (9):1483–1489, 2002.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

Megan F Cole and Eric A Gaucher. Utilizing natural diversity to evolve protein function: applications towards thermostability. *Current opinion in chemical biology*, (3):399–406, 2011.

Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. What is a meaningful representation of protein sequences? *arXiv preprint arXiv:2012.02679*, 2020.

Xinqiang Ding, Zhengting Zou, and Charles L Brooks III. Deciphering protein evolution and fitness landscapes with latent space models. *Nature communications*, (1):1–13, 2019.

Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

Mohammed El-Kebir, Layla Oesper, Hannah Acheson-Field, and Benjamin J Raphael. Reconstruction of clonal trees and tumor composition from multi-sample sequencing data. *Bioinformatics*, (12):i62–i70, 2015.

Brian Gaschen, Jesse Taylor, Karina Yusim, Brian Foley, Feng Gao, Dorothy Lang, Vladimir Novitsky, Barton Haynes, Beatrice H Hahn, Tanmoy Bhattacharya, et al. Diversity considerations in HIV-1 vaccine selection. *Science*, (5577):2354–2360, 2002.

Michael Golden, Eduardo García-Portugués, Michael Sørensen, Kanti V Mardia, Thomas Hamelryck, and Jotun Hein. A generative angular model of protein structure evolution. *Molecular biology and evolution*, 34(8):2085–2100, 2017.

Joe G Greener, Lewis Moffat, and David T Jones. Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, pp. 1–12, 2018.

Pantelis Z Hadjipantelis, Nick S Jones, John Moriarty, David Springate, and Christopher G Knight. Ancestral inference from functional data: statistical methods and numerical examples. *arXiv preprint arXiv:1208.0628*, 2012.

Barry G Hall. Comparison of the accuracies of several phylogenetic methods using protein and DNA sequences. *Molecular biology and evolution*, 22(3):792–802, 2005.

Barry G Hall. Effects of sequence diversity and recombination on the accuracy of phylogenetic trees estimated by kSNP. *Cladistics*, (1):90–99, 2016.

Thomas F Hansen. Stabilizing selection and the comparative analysis of adaptation. *Evolution*, 51(5):1341–1351, 1997.

Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLoS computational biology*, (2):e1008736, 2021.

Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the national academy of sciences*, (22):10915–10919, 1992.

Georg KA Hochberg and Joseph W Thornton. Reconstructing ancient proteins to understand the causes of structure and function. *Annual Review of Biophysics*, pp. 247–269, 2017.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of machine learning research*, 14(5), 2013.

Edwin Rodríguez Horta, Alejandro Lage-Castellanos, and Roberto Mulet. Ancestral sequence reconstruction for co-evolutionary models. *arXiv preprint arXiv:2108.03801*, 2021.

Metod Jazbec, Matt Ashman, Vincent Fortuin, Michael Pearce, Stephan Mandt, and Gunnar Rätsch. Scalable Gaussian process variational autoencoders. In *International Conference on Artificial Intelligence and Statistics*, pp. 3511–3519. PMLR, 2021.

Nick S Jones and John Moriarty. Evolutionary inference for function-valued traits: Gaussian process regression on phylogenies. *Journal of the Royal Society Interface*, 10(78):20120616, 2013.

Jeffrey B Joy, Richard H Liang, Rosemary M McCloskey, T Nguyen, and Art FY Poon. Ancestral reconstruction. *PLoS computational biology*, 12(7):e1004763, 2016.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Diederik P Kingma and Max Welling. An introduction to variational autoencoders. *Foundations and trends in machine learning*, 12:307–392, 2019.

Jeffrey M Koshi and Richard A Goldstein. Probabilistic reconstruction of ancestral protein sequences. *Journal of molecular evolution*, 42(2):313–320, 1996.

Amaury Lambert and Tanja Stadler. Birth–death models and coalescent point processes: The shape and probability of reconstructed phylogenies. *Theoretical population biology*, 90:113–128, 2013.

Nicolas Lartillot. A phylogenetic Kalman filter for ancestral trait reconstruction using molecular data. *Bioinformatics*, pp. 488–496, 2014.

Nicolas Lartillot, Nicolas Rodrigue, Daniel Stubbs, and Jacques Richer. PhyloBayes MPI: Phylogenetic reconstruction with infinite mixtures of profiles in a parallel environment. *Systematic biology*, (4):611–615, 2013.

Emile Mathieu, Charline Le Lan, Chris J Maddison, Ryota Tomioka, and Yee Whye Teh. Continuous hierarchical representations with Poincaré variational auto-encoders. *arXiv preprint arXiv:1901.06033*, 2019.

Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108:E1293–E1301, 2011.

Lam-Tung Nguyen, Heiko A Schmidt, Arndt Von Haeseler, and Bui Quang Minh. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular biology and evolution*, 32(1):268–274, 2015.

Linus Pauling, Emile Zuckerkandl, T Henriksen, and R Lövstad. Chemical paleogenetics. *Acta chemica Scandinavica*, 17:S9–S16, 1963.

Ryan N Randall, Caelan E Radford, Kelsey A Roof, Divya K Natarajan, and Eric A Gaucher. An experimental phylogeny to benchmark ancestral sequence reconstruction. *Nature communications*, 7(1):1–6, 2016.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with TAPE. *Advances in neural information processing systems*, 32:9689, 2019.

Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA transformer. *bioRxiv*, 2021.

Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

Avery GA Selberg, Eric A Gaucher, and David A Liberles. Ancestral sequence reconstruction: From chemical paleogenetics to maximum likelihood algorithms and beyond. *Journal of Molecular Evolution*, (3):157–164, 2021.

Matthew A Spence, Joe A Kaczmarski, Jake W Saunders, and Colin J Jackson. Ancestral sequence reconstruction for protein engineers. *Current Opinion in Structural Biology*, pp. 131–141, 2021.

Max R Tolkoff, Michael E Alfaro, Guy Baele, Philippe Lemey, and Marc A Suchard. Phylogenetic factor analysis. *Systematic biology*, (3):384–399, 2018.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, (11), 2008.

Sharad Vikram, Matthew D Hoffman, and Matthew J Johnson. The LORACs prior for VAEs: Letting the trees speak for the data. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 3292–3301. PMLR, 2019.

C Wilson, RV Agafonov, M Hoemberger, S Kutter, A Zorba, J Halpin, V Buosi, R Otten, D Waterman, DL Theobald, et al. Using ancient protein kinases to unravel a modern cancer drug's mechanism. *Science*, 347(6224):882–886, 2015.

Ziheng Yang. PAML 4: Phylogenetic analysis by maximum likelihood. *Molecular biology and evolution*, 24(8):1586–1591, 2007.

Ziheng Yang, Sudhir Kumar, and Masatoshi Nei. A new method of inference of ancestral nucleotide and amino acid sequences. *Genetics*, 141(4):1641–1650, 1995.

A Appendix

A.1 Notation and variables

Name	**Description**
n_Z	Number of TOU processes, length of latent vector (30)
n_L	Alignment length
n_S	Number of leaf sequences
n_A	Number of ancestral sequences ($n_A = n_S - 1$)
n_C	Number of character types
$\mathbf{S}$	Sequence alignment matrix of leaf sequences
$\mathbf{Z}^{(S)} \equiv \mathbf{Z}$	Matrix of latent representations of the leaf sequences Note: We use $\mathbf{Z}$ for notational convenience where possible.
$\mathbf{Z}^{(A)}$	Matrix of latent representations of the ancestral sequences
$\mathbf{Z}^{(A,S)}$	Matrix of latent representations of the leaf and ancestral sequences
$\mathbf{T}$	Patristic distance matrix (given)
$\mathbf{T}^{(S,S)}$	Patristic distance submatrix of distances between leaf sequences (given)
$\mathbf{T}^{(A,A)}$	Patristic distance submatrix of distances between ancestral sequences (given)
$\mathbf{T}^{(A,S)}$	Patristic distance submatrix of distances between leaf and ancestral sequences (given)
$\mathbf{B}$	BLOSUM matrix (given)
α	Vector of OU hyperprior parameters
$\sigma_{\mathbf{f}}$	Vector of intensities of inherited variation (TOU process)
$\sigma_{\mathbf{n}}$	Vector of intensities of specific variation (TOU process)
λ	Vector of characteristic lenght-scales (TOU process)
$\mathbf{C}$	Tensor of n_Z TOU process covariance matrices
$\mathbf{V}$	Matrix of weighted BLOSUM vectors
$\mathbf{E}$	Matrix of BLOSUM embeddings in the model
$\mathbf{F}$	Tensor of BLOSUM embeddings in the guide
$\mathbf{Y}$	Input tensor to GRU
$\mathbf{H}$	State tensor of GRU
$\mathbf{L}$	Tensor of logits of the n_C sequence characters
θ	Parameters of the neural networks and the GRU
$\mathbf{0}^{(d)}$	d-dimensional vector of zeros
$\mathbf{0}^{(m \times n)}$	$(m \times n)$-dimensional matrix of zeros
GRU	Gated recurrent unit
NN	Fully connected neural network
$\mathcal{HN}$	Half-normal distribution
$\mathcal{MVN}$	Multivariate Gaussian distribution
cat	Concatenation of two vectors.

Table 1: Variables and notation used for the Draupnir model.

Name	**Dimensions**
α	3
$\sigma_{\mathbf{f}},\ \sigma_{\mathbf{n}},\ \lambda$	n_Z
$\mathbf{T}$	$(n_S + n_A) \times (n_S + n_A)$
$\mathbf{T}^{(A,A)}$	$n_A \times n_A$
$\mathbf{T}^{(S,S)}$	$n_S \times n_S$
$\mathbf{T}^{(A,S)}$	$n_A \times n_S$
$\mathbf{C}$	$n_Z \times n_S \times n_S$
$\mathbf{Z}^{(S)} \equiv \mathbf{Z}$	$n_S \times n_Z$
$\mathbf{Z}^{(A)}$	$n_A \times n_Z$
$\mathbf{Z}^{(A,S)}$	$(n_S + n_A) \times n_Z$
$\mathbf{Y}$	$n_S \times n_L \times (n_Z + n_C)$
$\mathbf{E}$	$n_L \times n_C$
$\mathbf{F}$	$n_S \times n_L \times n_C$
$\mathbf{H}$	$n_S \times n_L \times 60$
$\mathbf{L}$	$n_S \times n_L \times n_C$
$\mathbf{B}$	$n_C \times n_C$
$\mathbf{S}$	$n_S \times n_L$
$\mathbf{V}$	$n_L \times n_C$

Table 2: Dimensions of variables

A.2 Draupnir settings

Neural network architecture The Draupnir model contains three neural networks (see Algorithm 1, Figure 1 and Figure 8): a fully connected network, $\mathrm{NN}_\theta^{(1)}$, that maps the pre-computed BLOSUM vectors, $\mathbf{V}$, to BLOSUM embeddings, $\mathbf{E}$; a bidirectional GRU_θ (Cho et al., 2014) with a single layer that takes as input the BLOSUM embeddings and the latent vector of the $k-$th leaf sequence, $\mathbf{Z}_{k,:}$, and a second fully connected network, $\mathrm{NN}_\theta^{(2)}$, that maps the GRU states to the logit vectors of the sequence characters. The dimensionality of the state of the bidirectional GRU is 2×60.

In the guide, we re-use the neural network architectures described above: $\mathrm{NN}_\phi^{(2)}$ and GRU_ϕ are identical to $\mathrm{NN}_\theta^{(2)}$ and GRU_θ, respectively, except that the output size of $\mathrm{NN}_\phi^{(2)}$ is $(2 \times n_Z)$ instead of n_C, corresponding to the length of the mean vector and standard deviation vector of the latent representation. $\mathrm{NN}_\phi^{(1)}$ is identical to $\mathrm{NN}_\theta^{(1)}$.

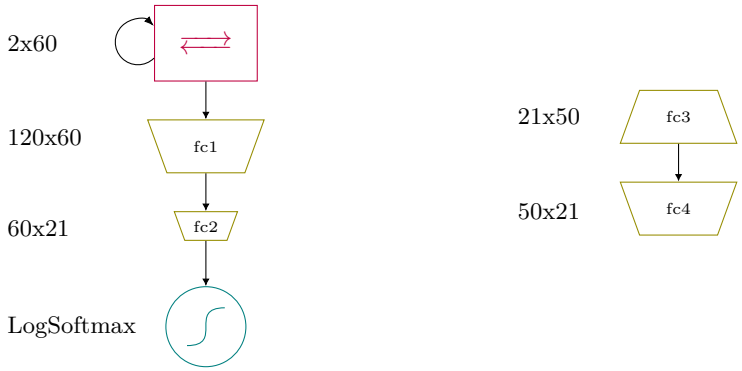


Figure 8: Architectures and dimensions of the neural networks used in Draupnir. *Left*: Architecture of the bidirectional GRU_θ (red) and $\mathrm{NN}_\theta^{(2)}$. *Right*: Architecture of $\mathrm{NN}_\theta^{(1)}$. "fc" indicates a fully connected layer.

Additional Draupnir settings We chose BLOSUM62 as the base substitution matrix, except for the CFP datasets (71 and 35 leaves) where we use PAM70 due to the presence of special amino acids.

A.3 Data sets

Table 3: Descriptions of the nine data sets. All data sets contain insertions and deletions (gaps), except the one in cursive (top), which only contains substitutions. Data sets labelled with an asterisk only contain the sequence of the root node.

Dataset	Number of leaves	Alignment length	Source
Datasets with experimentally determined ancestral sequences			
Randall's Coral fluorescent proteins (CFP) benchmark	19	225	Randall et al. (2016)
*Coral fluorescent proteins (CFP) Faviina subclade	35	261	*allfav* root node sequence from Alieva et al. (2008)
*Coral fluorescent proteins (CFP) subclade	71	272	*allcor* root node sequence from Alieva et al. (2008)
EvolveAGene4 (Hall, 2005) simulations			
Simulation β-Lactamase	32	314	GenBank accession no. AF309824
Simulation Calcitonin	50	71	NBCI CCDS7820.1
Simulation SRC-Kinase SH3 domain	100	63	GenBank BC011566.1
Simulation Sirtuin	150	477	NBCI CCDS44412.1
Simulation SRC-Kinase SH3 domain	200	128	GenBank BC011566.1
Simulation SRC-Kinase SH3 domain	800	99	GenBank BC011566.1
Leaves only data set			
PF00400	125	138	PFAM family no. PF00400

A.4 Training

Table 4: Training settings and running times (Intel(R) Xeon(R) Gold 6136 CPU @ 3.00GHz, Quadro RTX 6000 GPU).

Dataset	**Epochs**	**Model parameters**	**Running times**
Datasets with experimentally determined ancestral sequences			
Randall's Coral fluorescent proteins (CFP) benchmark, 19 leaves	16600	52055	1 day, 58 min
Coral fluorescent proteins (CFP) Faviina subclade, 35 leaves	23000	55181	2 days, 4 h, 21 min
Coral fluorescent proteins (CFP) clade, 71 leaves	23000	52535	1 day, 19 h, 16 min
EvolveAGene4 (Hall, 2005) simulations			
Simulation β-Lactamase, 32 leaves	15000	52445	19h, 7min
Simulation Calcitonin, 50 leaves	18700	52985	1 day, 10 h, 28 min
Simulation SRC-Kinase SH3 domain, 100 leaves	21600	54485	2 days, 6 h, 22 min
Simulation Sirtuin, 150 leaves	20000	55985	1 day, 11 min
Simulation SRC-Kinase SH3 domain, 200 leaves	22000	57485	2 days, 14 h,7 min
Simulation SRC-Kinase SH3 domain, 800 leaves	12000	141005	22 h 48 min
Leaves only data set			
PF00400, 125 leaves - Marginal	15100	55235	18 h 48 min
PF00400, 125 leaves - Variational	7700	137487	18 h 49 min

A.5 Benchmarking tables

Table 5: Benchmarking results: Protein sequences. The table shows the average and the standard deviation of the percent identity across the predicted ancestral sequences and their corresponding true sequences. "Not available" indicates the ASR method failed to produce results for that data set on the given hardware. The results for the standard ASR methods were obtained using the amino acid sequences.

	Number of leaves	Alignment length	Draupnir MAP	Draupnir samples	PAML-CodeML	PhyloBayes	FastML	IQTree
Randall's Coral fluorescent proteins (CFP)	19	225	94.61±1.59	89.52±2.76	98.14±1.3	98.09±1.03	98.17±1.31	98.27±1.07
Coral fluorescent proteins (CFP) Faviina subclade	35	261	86.05±0.97	86.74±0.98	74.56±1.14	74.33±0.86	86.76±1.27	76.01±1.16
Coral fluorescent proteins (CFP) clade	71	272	75.47±1.19	74.30±1.11	60.96±0.8	61.17±0.85	74.94±1.07	60.96±0.8
Simulation Beta-Lactamase	32	314	81.38±6.95	81.55±6.77	57.91±22.39	83.12±6.13	87.52±6.28	83.07±6.01
Simulation Calcitonin, 50	50	71	79.85±8.75	77.75±8.45	41.94±21.21	69.45±8.88	86.52±6.07	69.54±8.83
Simulation SRC-Kinase SH3 domain, 100	100	63	73.27±10.45	73.17±9.49	33.24±17.33	66.78±9.35	83.21±8.71	66.76±9.17
Simulation Sirtuin SIRT1, 150	150	477	82.34±4.48	77.57±5.19	10.97±5.59	17.24±1.09	Not available	15.93±1.33
Simulation SRC-Kinase SH3 domain, 200	200	128	87.50±5.09	86.94±4.81	9.69±7.99	9.10±1.64	91.09±4.54	10.44±2.71
Simulation SRC-Kinase SH3 domain, 800	800	99	88.62±4.97	87.92±4.43	Not available	Not available	Not available	49.63±3.6

Table 6: Benchmarking results: DNA sequences. The table shows the average and the standard deviation of the percent identity across the predicted ancestral sequences and their corresponding true sequences. "Not available" indicates the ASR method failed to produce results for that data set on the given hardware. The results for the standard ASR methods were obtained using the DNA sequences.

	Number of leaves	Alignment length	Draupnir MAP	Draupnir samples	PAML-CodeML	PhyloBayes	FastML	IQTree
Randall's Coral fluorescent proteins (CFP)	19	225	94.61±1.59	89.52±2.76	98.69±0.82	98.82±0.83	98.59±0.77	98.69±0.76
Coral fluorescent proteins (CFP) Faviina subclade	35	261	86.05±0.97	86.74±0.98	76.01±1.16	76.67±1.13	86.76±1.27	76.01±1.16
Coral fluorescent proteins (CFP) clade	71	272	75.47±1.19	74.30±1.11	60.88±0.85	54.65±0.71	74.94±1.07	61.10±0.79
Simulation Beta-Lactamase	32	314	81.38±6.95	81.55±6.77	58.21±22.68	83.61±6.15	87.66±6.3	83.65±6.13
Simulation Calcitonin, 50	50	71	79.85±8.75	77.75±8.45	41.88±21.14	70.01±8.92	86.55±6.29	69.42±8.89
Simulation SRC-Kinase SH3 domain, 100	100	63	73.27±10.45	73.17±9.49	33.06±17.56	68.90±9.24	82.31±9.9	67.90±9.31
Simulation Sirtuin SIRT1, 150	150	477	82.34±4.48	77.57±5.19	11.01±5.67	22.00±2.09	86.43±11.19	69.06±0.77
Simulation SRC-Kinase SH3 domain, 200	200	128	87.50±5.09	86.94±4.81	9.72±8.17	17.60±2.53	84.48±8.5	61.62±2.71
Simulation SRC-Kinase SH3 domain, 800	800	99	88.62±4.97	87.92±4.43	Not available	Not available	Not available	49.66±3.68

A.6 Coevolution analysis

Conventional ASR methods use models that assume factorized evolution (Horta et al., 2021), that is, they assume that each site evolves independently of all other sites in the sequence. In reality, some sites are coupled in evolution, resulting in dependencies between sites. This phenomenon is called epistasis (Hochberg & Thornton, 2017). Draupnir is a model that goes beyond factorized evolution, and thus potentially models coevolving sites. Here, we analyze to what extend this is indeed the case.

In order to evaluate modelling of coevolution, we make use of direct coupling analysis (DCA) (Morcos et al., 2011). DCA identifies coevolving pairs of residues that directly influence each other by calculating a quantity called Direct Information (DI), which is obtained by fitting a Markov random field. We calculated the DI using *ProDy* (Bakan et al. (2014)). As data set, we used 125 sequences from the WB40 domain of the PF00400 family from the PFAM data base.

The DIs of the leaf sequences serve as ground truth and are compared with sequences sampled from Draupnir at the root node in three different ways (see below). If coevolution is at least partially modelled, the DIs of the leaf sequences will be similar (but not completely identical) to the DIs of the sampled sequences at the root node. We sample sequences from Draupnir using three different methods.

The first method (MAP) simply uses the MAP estimates of the probability vectors at each position:

$$\mathbf{a} \sim \prod_{i=1}^{n_L} p\left(a_i \mid \boldsymbol{\theta}_i^{(\mathrm{MAP})}\right),$$

where $\mathbf{a}$ is an ancestral sequence and $\boldsymbol{\theta}_i^{(\mathrm{MAP})}$ is the MAP estimate of the amino acid probability vector at position i for that sequence. This baseline results in independent

sites. Picking the most likely amino acid at each position according to the above expression corresponds to "Draupnir MAP" in Section 5.2.

The second method (Marginal) makes use of the MAP estimates of the leaf representations but marginalizes over the ancestral representations (see Section 4.4):

$$\mathbf{a} \sim \int \left(\prod_{i=1}^{n_L} p\left(a_i \mid \left[\boldsymbol{\theta}\left(\mathbf{z}^{(a)}\right)\right]_i\right) \right) p(\mathbf{z}^{(a)} \mid \mathbf{Z}^{(\mathrm{MAP})}) \mathrm{d}\mathbf{z}^{(a)},$$

where the probability vectors $\boldsymbol{\theta}_i$ are obtained from a GRU applied to the latent representation $\mathbf{z}^{(a)}$ of the ancestral sequence $\mathbf{a}$ (see Algorithm 1). The last factor involves conditional Gaussian distributions (see Equation 3 and Section 4.4). Integrating over $\mathbf{z}^{(a)}$ introduces correlations along the sequence. Therefore, this method is in principle capable to model some coevolution as we integrate over the latent representations of the ancestors (while using a point estimate for the latent representations of the leaves). This corresponds to "Draupnir samples" in Section 5.2.

In the third method (Variational), we make use of a guide $q_\phi(\mathbf{Z} \mid \mathbf{S})$ to obtain a variational posterior over the latent representations $\mathbf{Z}$ (see Algorithm 3):

$$\mathbf{a} \sim \int \left(\prod_{i=1}^{n_L} p\left(a_i \mid \left[\boldsymbol{\theta}\left(\mathbf{z}^{(a)}\right)\right]_i\right) \right) p(\mathbf{z}^{(a)} \mid \mathbf{Z}) q_\phi(\mathbf{Z} \mid \mathbf{S}) \mathrm{d}\mathbf{z}^{(a)} \mathrm{d}\mathbf{Z}.$$

In this case, we marginalize over the latent representations of both leaves and ancestors. This method should capture the coevolutionary signal to a greater extent than the second method.

The results are shown in Figure 9. As expected the third method, (Variational) does best, while the first method (MAP) does worst. The Variational method improves considerably upon the Marginal method, indicating that replacing the MAP estimate with a variational distribution for the latent representations of the leaves has significant impact. The results indicate that Draupnir indeed to a significant extent can capture coevolutionary information.

Algorithm 3 Architecture of the guide. We use point estimates for the hyperparameters and parameters of the TOU processes, and multivariate diagonal Gaussian distributions for the latent representations. An asterisk indicates the value of a point estimate. $\delta(.)$ is the Dirac delta function.

Require: Multiple sequence alignment $\mathbf{S}$

for $j \in \{1, 2, 3\}$ **do** ▷ Point estimates of TOU hyperprior parameters
  $\alpha_j \sim \delta(\alpha_j^*)$

for $h \in \{1, ..., n_Z\}$ **do** ▷ Point estimates of the parameters of the n_Z TOU processes
  $\sigma_{f,h} \sim \delta(\sigma_{f,h}^*)$
  $\sigma_{n,h} \sim \delta(\sigma_{n,h}^*)$
  $\lambda_h \sim \delta(\lambda_h^*)$

for $k \in [1, \ldots, n_S]$ **do**
  for $i \in [1, \ldots, n_L]$ **do**
    $r \leftarrow S_{k,i}$ ▷ Amino acid at position i in leaf sequence k
    $\mathbf{F}_{k,i,:} \leftarrow \mathrm{NN}_{\phi}^{(1)}(\mathbf{B}_{r,:})$ ▷ BLOSUM embedding

for $k \in [1, \ldots, n_S]$ **do**
  $\mathbf{H}_{k,:,:} \leftarrow \mathrm{GRU}_{\phi}(\mathbf{F}_{k,:,:})$ ▷ Bidirectional GRU states
  $\mathbf{m}, \mathbf{c} \leftarrow \mathrm{NN}_{\phi}^{(2)}(\mathbf{H}_{k,-1,:})$ ▷ Mean and standard deviations of $\mathbf{Z}_{k,:}$
  $\mathbf{Z}_{k,:} \sim \mathcal{MVN}(\mathbf{m}, \mathrm{diag}(\mathbf{c}))$

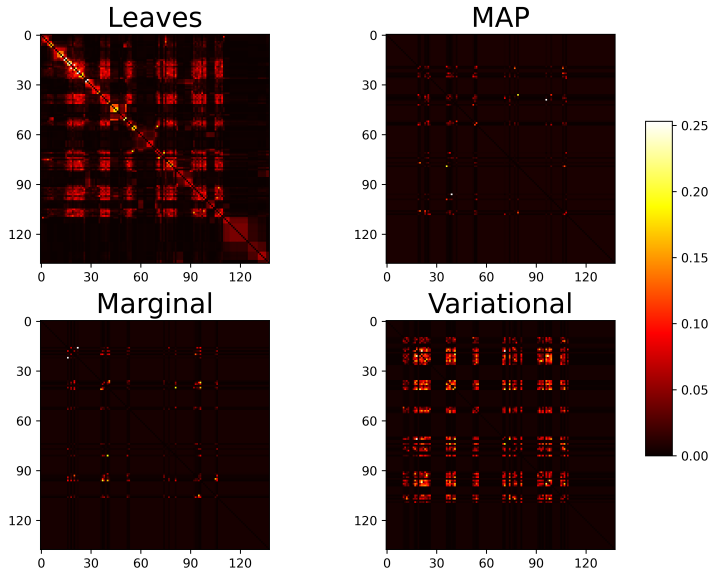


Figure 9: DI values for all position pairs of the WB40 data set obtained from the leaf sequences (upper left) and sequences sampled at the root node (other plots) using the MAP, Marginal and Variational methods. We sampled 125 root sequences, which is equal to the number of leaves. The correlation coefficients between the DIs of the leaves and the sampled sequences are 0.11 (MAP), 0.15 (Marginal) and 0.53 (Variational).

A.7 Quantitative analysis of the latent space representations

In order to compare the standard VAE with Draupnir in a quantitative way (see Figure 7), we analyze the correlation between (a) the Euclidean distances between the latent representations of the leaves and (b) the corresponding branch lengths in the phylogenetic tree for both models. The results are shown in Figure 10.

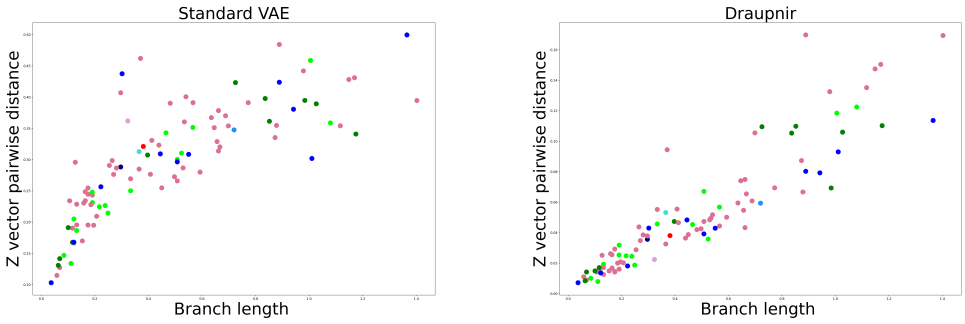


Figure 10: Comparison of the Euclidean distances between the latent representations of the leaves (y-axis) and the corresponding branch lengths in the phylogenetic tree (x-axis). We use the same color scheme as in Figure 2. We traverse the tree in level order and assign the colour of the clade of the first leaf. (*Left*) The standard VAE. The correlation coefficient is 0.79; the Spearman correlation coefficient is 0.85. (*Right*) Draupnir. The correlation coefficient is 0.91; the Spearman correlation coefficient is 0.945.

A.8 Benchmark settings

PAML-CodeML settings PAML-CodeML (provided by Biopython 1.78) was used with the following settings:

verbose	2 (includes detailed information of the posterior probabilities per node)
runmode	0 (utilize given tree)
seqtype	2 (amino acids)
clock	0 (no molecular clock, genes are evolving at different rates)
aaDist	0
getSE	0
RateAncestor	1
aaRateFile	WAG
method	1
model	2 (more dn/ds (selection coefficient) ratios per branch)
fix_alpha	0 (estimate gamma)
alpha	0.5 (initial alpha value for gamma distribution)
fix_blength	0 (keep given branch lengths)

PhyloBayes 4.1 settings

pd -s -f -T *treefile* -cat -gtr -d *alignmentfile chainname*

In the case of PhyloBayes, as recommended, we run 2 Markov chains until the convergence criteria are met. The recommended convergence criteria are minimum effective sample size above 300 and *max diff* among the chains below 0.1. Both for evaluation of the convergence of the chains and for sampling the ancestral sequences we use 100 samples.

FastML v3.11 settings

perl fastml –MSA-file *alignmentfile* –seqType *aa* or *nuc* –SubMatrix *WAG or GTR* –indelReconstruction *ML* –Tree *treefile*

IQ-Tree v2.0.3 settings

For IQ-Tree (multicore version 2.0.3), model choice and settings are automatically optimized by using the options "-m TEST" and "-nt AUTO".

iqtree -s *alignmentfile* -m *TEST* -asr -te *treefile* -nt *AUTO*