Reconstruction of ancestral protein sequences using autoregressive generative models

Pierre Barrat-Charlaix,* Matteo De Leonardis, and Andrea Pagnani

DISAT, Politecnico di Torino, Italy

(Dated:)

Abstract

abstract

I. INTRODUCTION

II. RESULTS

A. Autoregressive model of sequence evolution

Models of evolution commonly used in phylogenetics rely on the assumptions that sequence positions evolve independently and that evolution at each position i follows a continuous time Markov chain (CTMC) parametrized by a substitution rate matrix $\mathbf{Q}^i$. Matrix $\mathbf{Q}^i$ is of dimensions $q \times q$ where $q = 4$ for DNA, 20 for amino acids or 64 for codon models. The probability of observing a change from state a to state b during evolutionary time t is then given by $P(b|a,t) = \left(e^{t\mathbf{Q}^i}\right)_{ab}$.

If the model is time-reversible, it is a general property of CTMCs that the substitution rate matrix can be written as

$$\mathbf{Q} = \mathbf{H} \cdot \mathbf{\Pi} = \mathbf{H} \cdot \begin{pmatrix} \pi_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \pi_q \end{pmatrix}, \tag{1}$$

where $\mathbf{H}$ is symmetric and $\mathbf{\Pi}$ is diagonal with entries that sum to 1 [**?**]. The two matrices have simple interpretations. On the first hand, $\mathbf{\Pi}$ fixes the long-term equilibrium frequencies, that is $P(b|a,t) \xrightarrow[t\to\infty]{} \pi_b$. On the other, $\mathbf{H}$ influences the dynamics of the Markov chain but does not change the equilibrium distribution. Most commonly, $\mathbf{\Pi}$ is considered to be independent of the sequence position i, while $\mathbf{H}$ can be multiplied by position-dependent rates in order to model the different variability of different sites [**? ? ?**].

In order to incorporate constraints coming from a protein's structure and function into the evolutionary model, we develop a family-specific model of protein sequence evolution based on the the autoregressive generative model ArDCA [**?**]. ArDCA models the diversity

* Correspondance to: PBC, DISAT pierre.barratcharlaix@polito.it

of sequences in a protein family using a set of learned conditional probabilities. In practice, the model assigns a probability to any sequence $\mathbf{a} = \{a_1, \ldots, a_L\}$ of L amino acids:

$$P^{AR}(\mathbf{a}) = \prod_{i=1}^{L} p_i(a_i|a_{<i}), \tag{2}$$

where the product runs over positions in the sequence and $a_{<i} = a_1, \ldots, a_{i-1}$ represents all amino acids before position i. Functions p_i represent the probability according to the model to observe state a_i in position i, given that the previous amino acids were $a_1, \ldots, a_{i-1}$. Their precise functional form is given in the methods section. They are learned using the aligned sequences of members of the family. In actual implementations, the order in which the product in Eq. 2 is performed is not the natural $(1, \ldots, L)$ but rather an order where positions are sorted by increasing variability. This does not significantly effect the model we present below, and we keep the notation of Eq. 2 for simplicity.

It has been shown in [**?**] that the generative capacities of ArDCA are comparable to that of state of the art models such as bmDCA [**?**]. This means that a set of sequences sampled from the probability in Eq. 2 is statistically hard to distinguish from the natural sequences used in training or, in other words, that the model can be used to sample new artificial homologs of a protein family. Generative capacities of a protein model come from its ability to represent epistasis, that is the relation between the effect of a mutation and sequence context in which it occurs. Here, epistasis is modeled through the conditional probabilities p_i: the distribution of amino acids at position i depends on the states of the previous positions $1, \ldots i-1$.

We take advantage of the autoregressive architecture to define the following evolution model. Given two amino acid sequences $\mathbf{a}$ and $\mathbf{b}$, we propose

$$P(\mathbf{b}|\mathbf{a}, t) = \prod_{i=1}^{L} q_i(b_i|a_i, b_{<i}, t), \tag{3}$$

where the conditional propagator q_i is defined as

$$q_i(b_i|a_i, b_{<i}, t) = \left(e^{t\cdot Q^i(b_{<i})}\right)_{a_i,b_i}, \quad Q^i(b_{<i}) = \mathbf{H} \cdot \begin{pmatrix} p_i(1|b_{<i}) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & p_i(q|b_{<i}) \end{pmatrix}. \tag{4}$$

According to these equations, evolution for each position i follows a standard CTMC. However, we use the decomposition of Eq. 1 to set the equilibrium frequency at i to $p_i(b|b_{<i})$. In other words, we consider that position i evolves in the context of $b_1, \ldots, b_{i-1}$, and that its dynamics are constrained by its long term frequency given by the autoregressive model. An important consequence of this choice is that our evolution model will converge at long times to the generative distribution P^{AR}:

$$q_i(b_i|a_i, b_{<i}, t) \xrightarrow[t\to\infty]{} p_i(b_i|b_{<i}), \quad P(\mathbf{b}|\mathbf{a}, t) \xrightarrow[t\to\infty]{} P^{AR}(\mathbf{b}). \tag{5}$$

We argue here that such a property is essential to build a realistic protein sequence evolution model, particularly when considering evolution over a relatively long time frame. Note that to converge to a generative distribution, accurate modeling of epistasis is required. Using site-specific frequencies would not be sufficient, as the effect of mutations in a protein sequence typically depends on the context [**?**]. The technique proposed here allows us to represent epistasis through the context dependent probabilities p_i, while still considering each sequence position one at a time.

Interestingly, we note that the model in Eq. 3 is not time reversible, although context dependent site propagators in Eq. 4 are reversible. We show in the SM that this is mainly an artifact of the autoregressive nature of the model coupled with epistasis. Using non-time reversible evolutionary models is uncommon in the field, but this is mainly due to practical considerations and there are no fundamental reasons for evolution itself to be reversible [**?**]. In practice, this means that algorithms using this model have to be adapted accordingly.

We underline that this approach has important differences with standard models of evolution used in phylogenenetics. In phylogenetic reconstruction, the tree and the sequence evolution model are usually inferred at the same time and from the same data. The number of parameters of the evolution model is then kept low to reduce the risk of overfitting, for instance by using site specific rates to account for variable and conserved sites. Methods that introduce more complex models such as site specific frequencies do so by jointly inferring the parameters and the tree, leading to relatively complex algorithms [**?** **?**].

Here instead, parameters of the generative model in Eq. 2 are learned from a protein family, *i.e.* a set of diverged homologous protein sequences. While it is true that these sequences share a common evolutionary history and cannot be considered as independent

samples, common learning procedures only account for this in a very crude way [**?** **?**]. Despite this, it appears that the generative properties of such models are not strongly affected by ignoring the phylogeny [**?** **?**]. This allows us to proceed in two steps: first construct the model from data while ignoring phylogeny, and then only use it for phylogenetic inference tasks.

An advantage of this approach is that once the model of Eq. 2 is inferred, the propagator in Eq. 3 comes "for free" as no additional parameters are required. Importantly, our model does not use site specific substitution rates. Indeed, it has been shown that these can be seen as emergent properties when using more complex evolution models such as the one presented here [**?**]. However, a disadvantage is that the technique is only applicable to a given protein family at a time, and requires the existence of an appropriate training set for the model.

B. Ancestral sequence reconstruction

We apply our evolutionary model to the task of ancestral sequence reconstruction (ASR). The goal of ASR is the following: given a set of extant sequences with a shared evolutionary history and the corresponding phylogenetic tree, is it possible to reconstruct the sequences of extinct ancestors at the internal nodes of the tree? Along with the autoregressive evolutionary model described above, we thus need two inputs to perform ASR: a known phylogenetic tree, and the multiple sequence alignment of the leaf sequences. The length of the aligned sequences has to exactly correspond to that of the autoregressive model.

To reconstruct ancestral sequences using the autoregressive model, we proceed as follows:

i for sequence position $i = 1$, use the evolution model defined by the equilibrium frequencies p_1 to reconstruct a state a_1^n at each internal node n of the tree;

ii iterating through subsequent positions $i > 1$: reconstruct state a_i^n at each internal node n using the model defined in Eq. 4, with the context $a_{<i}^n$ having been already reconstructed in the previous iterations.

It is important to note that when any position $i > 1$ is reconstructed, the context at different internal nodes of the tree may differ. For a branch joining two nodes (n, m) of the tree, the evolution model will thus differ if we go down or up the branch: in one case the context at node n must be used, in the other case the context at node m. This is a consequence of the

time-irreversibility of the model. For this reason, we use a variant of Felsenstein's pruning algorithm that is adapted to irreversible models [**?**]. This comes at no computational cost.

Note that this method is adapted to both maximum likelihood and Bayesian inference. In the ML case, each iteration reconstructs the most probable residue is at a position i given the already reconstructed context. In the Bayesian case, residue a_i^n is instead sampled from the posterior distribution.

In any realistic application, the phylogenetic tree has to be reconstructed from the aligned sequences. In principle, a consistent approach would use the same evolutionary model for tree inference and ASR. However, our model does not allow us to reconstruct the tree. Therefore, in any realistic application, the tree is reconstructed using an evolutionary model that typically will differ from ours.

To reduce issues related to this evolutionary model discrepancy, we adopt the following strategy: our ASR method blindly trusts the topology of the input tree, but recomputes the branch length using the sequences. As explained in the Methods, there is no direct way to optimize branch length with the autoregressive model. For simplicity, we use profile model with position-specific amino acid frequencies for this task. This provides a relatively accurate estimate of the branch lengths, as shown in Figure S1.

C. Results on simulated data

Setup. There are two difficulties when evaluating the capacity of a model to perform ASR. The first is that in the case of biological data, the real phylogeny and ancestral sequences are usually not known. As a consequence, one must rely on simulated data to measure the quality of reconstruction. The second is that the reconstruction of an ancestral sequence is always uncertain, as evolutionary models are typically stochastic. The uncertainty becomes higher for nodes that are remote from the leaves. This means that it is only possible to make a statistical assessment about the quality of a reconstruction.

To test our approach, we adopt the following setup. We first generate phylogenetic trees by sampling from a coalescent process. We decide to use Yule's coalescent instead of the more common Kingman. The latter tends to produce a large majority of internal nodes in close vicinity to the leaves with the others separated by very long branches, resulting in a trivial reconstruction for most nodes and a very hard one for the deep nodes. Yule's

coalescent generates a more even distribution of node depths, allowing us to better evaluate reconstruction quality, see Supplementary Material and Figure S2. For each tree, we simulate the evolution of sequences using a model that we refer to as "evolver" to obtain two multiple sequence alignments, one for the leaves and one for the internal nodes of the tree. We then reconstruct internal nodes using the desired approach by using the leaf alignment and the tree topology as input data.

We will consider two kinds of evolver models: *(i)* the same autoregressive model that we will then use for reconstruction, which is an ideal case and *(ii)* an evolutionary model based on a Metropolis sampling of a Potts model. These two evolvers come from models trained on actual protein families: we use evolvers based on the PF00072 response regulator family for results of the main text, and show results for three other families (PF00014, PF00076 and PF00595) in the Supplementary Material . It is important to note that the approach that we propose only makes sense when considering the evolution of a precise protein family, on which the model in Eq. 2 is trained. Hence, any evolver model used in our simulations should reproduce at long times the statistics of the considered protein family, *i.e.* it should satisfy Eq. 5. For this reason, we only consider the two evolvers above and do not use more traditional evolutionary models such as an arbitrary GTR on amino-acids [**?**].

For reconstruction, we compare our autoregressive approach to the commonly used IQ-TREE program [**?**]. When supplied with a protein sequence alignment and a tree, IQ-TREE infers a joint substitution rate matrix for all sequence positions, with rates that can differ accross positions. Both methods run on a fixed tree topology, with branch lengths being re-inferred using maximum likelihood (see Methods).

Autoregressive evolver. We first investigate the case of the autoregressive evolver. This setting is of course ideal for our method, as there is perfect coincidence between the model used to generate the data and to perform ASR. We first evaluate the quality of reconstruction by computing the Hamming distance of the real and inferred sequences for each internal node of the simulated phylogenies. The left and central panels of Figure 1 show this Hamming distance as a function of the node depth, that is the distance separating the node from the leaves, and for a maximum likelihood reconstruction. Hamming distance is computed including gap characters in the aligned sequences on the right panel, while they are ignored on the central one. We see that the autoregressive reconstruction clearly outperforms the

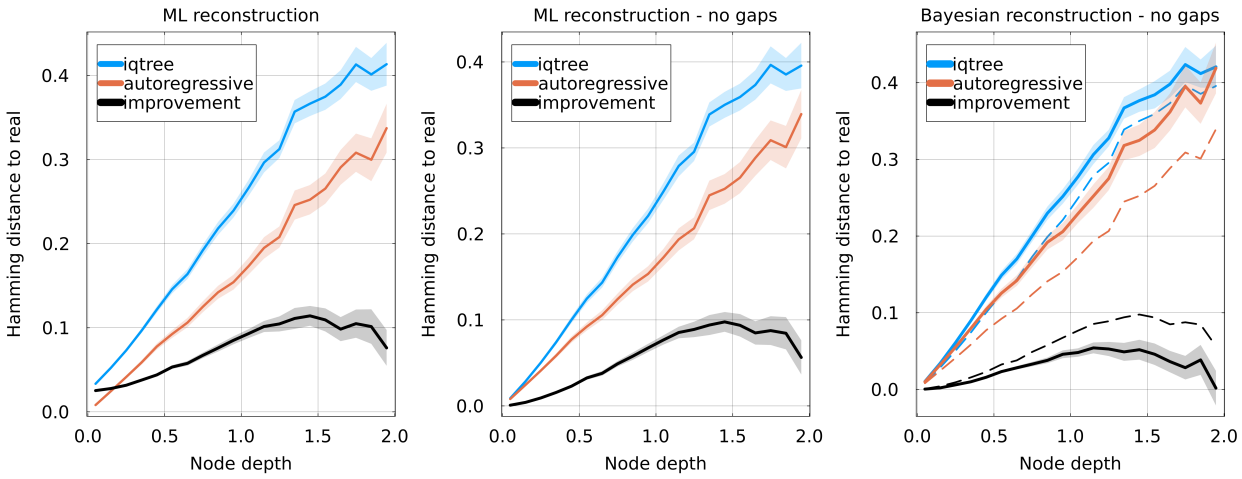

FIG. 1. Hamming distance between reconstructed and real sequences as a function of node depth, using IQ-TREE and our autoregressive approach. The difference between the two methods ("improvement") is shown as a black curve. Estimation of the incertitude is shown as a ribbon. The evolver and reconstruction autoregressive models are learned on the PF00072 family. **Left**: Hamming distance between the full aligned sequences, gaps included, using maximum likelihood reconstruction. **Center**: Hamming distance ignoring gapped positions, using maximum likelihood reconstruction. **Right**: comparison of Bayesian (solid lines) and maximum likelihood (dashed lines) reconstructions, ignoring gaps.

state of the art method: the improvement in Hamming distance increases with node depths, and the distance to the real ancestor drops from ~ 0.4 to ~ 0.3 when using the autoregressive approach.

Interestingly, the performance of IQ-TREE degrades if Hamming distance is computed including gaps, as in the left panel. This is because like other popular methods, IQ-TREE treats gaps in input sequences as unknown amino acids, and reconstructs an ancestral amino acid for gapped positions [**?** **?**]. On the contrary, our autoregressive approach treats gaps as if they were an additional amino acid and will reconstruct ancestral sequences that can contain gaps. This benefits the autoregressive approach as aligned ancetral sequences can in fact contain gaps. This effect is particularly visible at low node depths. However, ignoring the effects of gaps in the Hamming distance also leads to a clear improvement when using the autoregressive approach as shown in the central panel.

The right panel of Figure 1 shows the quality of the reconstruction for Bayesian recon-

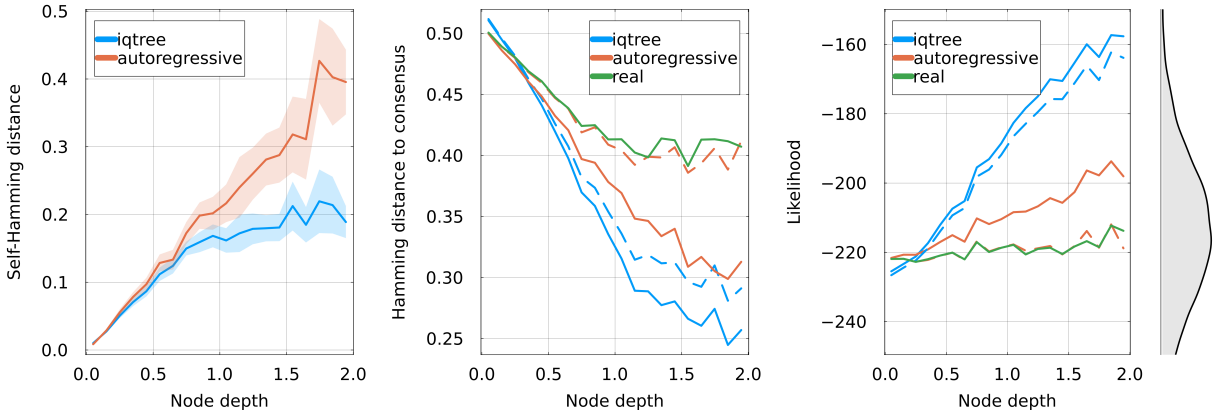

FIG. 2. **Left**: for Bayesian reconstruction, average pairwise Hamming distance among sequences reconstructed for each internal node. This quantifies the diversity of sequences obtained using Bayesian reconstruction. **Center**: Hamming distance between reconstructed sequences and the consensus sequence of the alignment. Solid lines represent maximum likelihood reconstruction or the real internal sequences, and dashed lines Bayesian reconstruction. IQ-TREE appears more biased towards the consensus sequence. **Right**: Log-likelihood of reconstructed and real sequences in the autoregressive model, *i.e.* using the logarithm of Eq. 2. Maximum likelihood methods (orange and blue solid lines) are biased towards more probable sequences. Bayesian autoregressive reconstruction gives sequences that are at the same likelihood level than the real ancestors. The equilibrium distribution of likelihood of sequences generated by Eq. 2 is shown on the right.

struction. In this case, a ensemble of sequences is reconstructed for each internal node, and the metric is the average Hamming distance between this ensemble and the real ancestor. Gaps are again ignored when computing the Hamming distance. We again observe an improvement when using the autoregressive method, of slightly lesser magnitude than in the maximum likelihood case.

Reconstructed sequences. To further analyze the reconstructed sequences, we first look at the diversity of generated ancestors in Bayesian reconstruction. The left panel of Figure 2 shows the average Hamming distance between sequences reconstructed at the same internal node, as a function of depth. For deeper nodes (depth $\gtrsim 1$), the Bayesian autoregressive approach reconstructs a significantly more diverse set of sequences than IQ-TREE: Hamming distance between reconstructions saturates at 0.2 for the latter, while it steadily increases

for the former. Higher diversity can be interpreted as a greater uncertainty concerning the ancestral sequence. However, this must be put in the context of Figure 1: sequences obtained by Bayesian autoregressive reconstruction are more varied but also on average closer to the real ancestor.

The difference in sequence diversity for the two methods is in part explained by the central panel of Figure 2, which shows the Hamming distance between reconstructed ancestors and the consensus sequence of the multiple sequence alignment at the leaves. It appears there that for deep nodes, IQ-TREE reconstructs sequences that are relatively similar to the consensus, with an average distance between the Bayesian reconstruction and the consensus of about 0.3. Contrasting with that, results of the autoregressive method shows less bias towards the consensus with an average distance of 0.4 for deep nodes, in line with the real ancestors. We also note that maximum likelihood sequences for both method are always closer to the consensus than Bayesian ones, which is a known bias of maximum likelihood [**?**].

The bias induced by ignoring the equilibrium distribution of the sequences is also visible in the right panel of Figure 2: it shows the log-likelihood of reconstructed and real ancestral sequences according to the generative model. Note that the log-likelihood here comes from the log-probability of Eq. 2 and can be interpreted as the "quality" of a sequence according to the generative model. It is unrelated to the likelihood computed in Felsenstein's pruning algorithm. Reconstructions with IQ-TREE increase in likelihood when going deeper in the tree, eventually resulting in sequences that are very uncharacteristic of the equilibrium generative distribution as can be seen from the histogram on the right. This effect also happens with the maximum likelihood reconstruction of the autoregressive model, although to a lesser extent. The Bayesian autoregressive reconstruction does not suffer from this bias and reconstructs sequences with a log-likelihood that is similar to that of the real ancestors.

Potts evolver. We then assess the performance of our reconstruction method in the case where the evolver is a Potts model. Potts models are a simple type of generative model and have been used extensively to model protein sequences. They can be use to predict contact in three dimensional structures, effects of mutations, protein-protein interaction partners [**?**]. They can be sampled to generate novel sequences which are statistically similar to natural ones and often functional [**?** **?**]. Additionally, it has recently been shown that they can be used to describe the evolution of protein sequences both qualitatively and quantitatively [**?**].

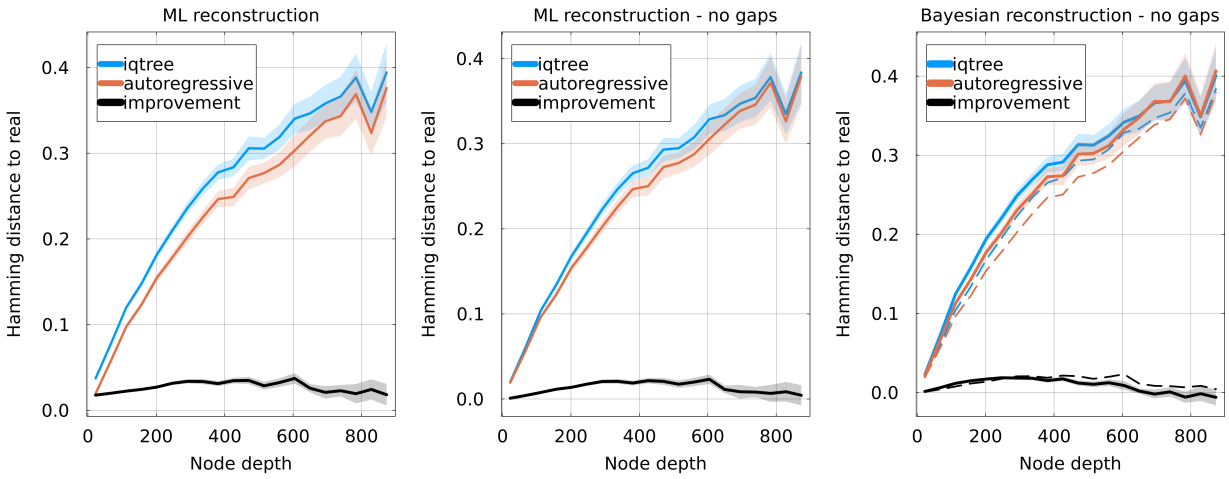

FIG. 3. Analogous to Figure 1, but using a Potts model as the evolver. Hamming distance between reconstructed and real sequences as a function of node depth, using IQ-TREE and our autoregressive approach. The difference between the two methods is shown as a black curve. The evolver and reconstruction autoregressive models are learned on the PF00072 family. **Left**: Hamming distance between the full aligned sequences, gaps included, using maximum likelihood reconstruction. **Center**: Hamming distance ignoring gapped positions, using maximum likelihood reconstruction. **Right**: comparison of Bayesian (solid lines) and maximum likelihood (dashed lines) reconstructions, ignoring gaps.

Potts and autoregressive models both accurately reproduce the statistical properties of protein families. In this sense, it can be considered that they correspond to similar long term generative distributions in the sense of Eq. 5. However, the dynamics of a Potts model are fundamentally different from the ones of usual evolutionary models, including our autoregressive one. First, they are described by a *discrete* time Markov chain, instead of the continuous time used in models based on substitution rate matrices such as in Eq. 1 [**?**]. The discrete time in these dynamics corresponds to attempts at mutation, which can be either accepted or rejected depending on the effect of the mutation according to the model. Secondly, these dynamics naturally give rise to different evolutionary timescales for various sequence positions, as well as interesting qualitative behavior such as the entrenchment of mutations.

To see how this change in dynamics affects our results, we *(i)* sample a large and varied ensemble of sequences from the Potts model and use it to train an autoregressive model, in a

way to guarantee consistent long term distributions between the Potts and autoregressive, and *(ii)* evolve the Potts model along random phylogenies, generating alignments for the leaves and the internal nodes in the same way as above. We then attempt reconstruction of internal nodes using the inferred autoregressive model and IQ-TREE. Figure 3 shows the results of reconstruction, with panels directly comparable to Figure 1. We again see a consistent improvement when using the autoregressive model over IQ-TREE, although of a much smaller amplitude, with an absolute improvement gain in Hamming distance of about 2% for deep internal nodes.

D. Results on experimental evolution data.

We take advantage of recent developments in directed evolution experiments to test our method in a controlled setting. We use the data published in [**?**]: in this work, authors evolved the antibiotic resistant proteins β-lactamase PSE-1 and acetyltransferase AAC6 by submitting them to cycles of mutagenesis and functional selection. Starting from a wild-type protein, they obtained thousands of diverse functional sequences after the directed evolution. An interesting result of this work is that it is possible to recover structural information about the wild-type from the set of evolved sequences.

Here, we use this data as a test setting for ASR: the sequences obtained after directed evolution all derive from a common ancestor, the wild-type, for which we know the amino acid sequence. We can thus reconstruct the wild-type sequence using different ASR methods and compare it to the ground truth. The phylogeny is not known, but given the large populatin size during the experiment and the relatively low number of selection rounds, it is reasonable to approximate it using a star-tree, *i.e.* a tree with a single coalescent event taking place at the root (see Methods). Since the reconstruction task is most interesting when using relatively varied sequences, we decide to use data for the PSE-1 wild-type where 20 cycles of mutagenesis & selection have been performed, resulting in a mean Hamming distance of 12% to the wild-type.

Our ASR procedure is as follows. We randomly pick M amino acid sequences among the proteins evolved from PSE-1 and with 20 cycles of mutagenesis & selection, with $3 \leq M \leq 640$. The total number of sequences at round 20 of directed evolution is much larger, making it computationally hard to use all of them. We then construct a star-like phylongeny and

place the M selected sequences at the leaves, and perform ASR using either IQ-TREE or our autoregressive method which we have trained on an alignment of PSE-1 homologs. We obtain the reconstructed amino acid sequence of the root, which we can then compare to the actual wild-type. As a comparison, and because our approximation of the phylogeny is very simple, we also attempt to reconstruct the root by taking the consensus sequence of the M leaves. We repeat this procedure 100 times for each value of M for a statistical assessment of the different methods.

The results are shown in Figure 4. The left panel shows the average non-normalized Hamming distance to the wild-type as a function of the number of leaves used M. For a low M, all methods understandably make a large number of errors, with a mean Hamming distance larger than 10 for $M = 3$. For a higher M, IQ-TREE and the autoregressive method stabilize to a fixed number of errors: we find a Hamming distance of ~ 4.3 for IQ-TREE and ~ 2.9 for the autoregressive. The consensus curiously reaches a minimum at intermediate M, a fact commented in the Supplementary Material , and saturates at a Hamming distance of 6 when considering all sequences of the round 20. The reconstruction errors are overwhelmingly located at six sequence positions. In the central panel, the fraction of mistakes made at these six positions over the 100 repetitions of $M = 640$ leaves is shown for each method. We observe that there are two positions (129 and 152) where IQ-TREE systematically fails at recovering the wild-type state while the autoregressive model's reconstruction is correct. Inversely, IQ-TREE recovers the wild-type state more often at position 68. The right panel shows the logo of the set of reconstructed sequences at these 6 positions and for each method.

Overall, we see that the reconstruction of the autoregressive model is more accurate. This gain in accuracy comes from the representation of the functional constraints acting on the PSE-1 protein by the generative model, which are inferred separately using an alignment of homologs. The improvement in reconstruction errors is modest, going from an average Hamming distance of 4.3 to 2.9. However, the gain is intrisically limited by the data itself: the evolved sequences have an average Hamming distance of about 12% to the ancestor, which is experimentally challenging but remains small compared to the divergence found in the homologs of PSE-1. For instance, the root-to-tip distance estimated by IQ-TREE and the autoregressive model are respectively 0.13 and 0.15, corresponding to the regime of shallow trees when comparing with Figure 1.

In order

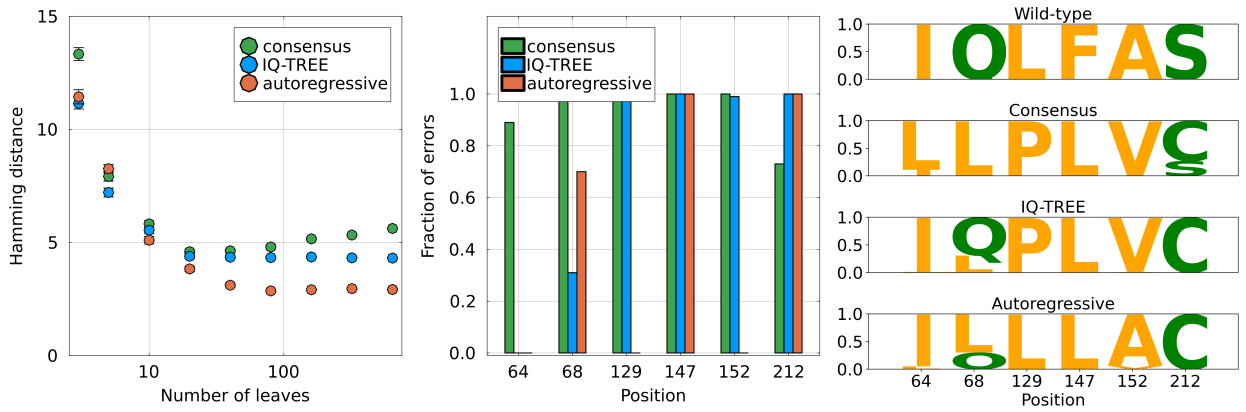

FIG. 4. Reconstruction of the wild-type PSE1 sequence used in [**?**] using sequences from round 20 of the directed evolution. **Left.** Non normalized Hamming distance to the wild-type PSE1 sequence as a function of the number of sequences M used for reconstruction. The fact that the consensus method has a local minimum is discussed in the Supplementary Material . For comparison, the average distance between a leaf sequence and the wild-type is 25. The error bars are computed using the standard deviation obtained from the 100 choices of sequences. **Middle.** For the six sequence positions where most of the reconstruction errors are located, fraction of errors of each method out of 100 independent reconstructions using different sets of $M = 640$ leaves. **Right.** Sequence logo of the reconstructed sequence for the three methods, obtained using 100 independent reconstructions with different sets of $M = 640$ leaves. The logo is only shown for the six positions where most errors are located. For example, all three methods fail 100 times at position 147, reconstructing a leucine L instead of a phenylalanine F.

III. DISCUSSION

IV. METHODS

A. ArDCA

The ArDCA model assigns a probability to any sequence of amino acids of length L given by

$$P^{AR}(\mathbf{a}) = \prod_{i \in \sigma(L)} p_i(a_i | a_{<i}), \tag{6}$$

where $\sigma(L)$ is a permutation of the L first integers and $a_{<i}$ stands for $a_1, \ldots, a_{i-1}$. This

means that the order in which the conditional probabilities p_i are applied is not necessarily the sequence order. The permutation σ is fixed at model inference.

Conditional probabilities p_i are defined as

$$p_i(b|a_{<i}) = \frac{1}{Z_i} \exp\left(\sum_{j<i} J_{ij}(b, a_j) + h_i(b)\right), \tag{7}$$

with the i q-dimensional vectors $J_{i\cdot}$ and h_i are learned parameters. The model is normally trained using a multiple sequence alignment of homologous proteins, *i.e.* a protein family, by finding the parameters J and h that maximize the likelihood of the sequences. It was shown in [**?**] that this specific parametrization captures essential features of the variability of members of a protein family.

By definition, homologous proteins share a joint evolutionary history and cannot be considered as statistically independent. To avoid biases, a reweighting is applied to sequences based on their vicinity to other sequences. This scheme has been showed to substantially increase the performance of such models [**?**].

B. Branch length inference

- how it works generally
- why I can't use AR directly
- how we do it with the profile model
- refer to the SI figure for results

C. Simulations

A simulation is performed as follows. First, a random tree of $n = 100$ leaves is generated from Yule's coalescent. We then normalize its height to a fixed value H that depends on the evolver model used: for the autoregressive model we use $H = 2.0$, while for the Potts model combined with Metropolis steps, we use $H = 8$ sweeps, *i.e.* $H = 8L$ Metropolis steps where L is the length of the sequences.

A root sequence is sampled from the evolver model's equilibrium distribution, and evolution is simulated along each branch independently starting from the root. In the case of the

autoregressive evolve, the dynamics is the one of Eq. 3. In the case of the Potts model, we use a Markov chain with the Metropolis update rule. In this way, we obtain for each repetition a tree and the alignments for internal and leaf nodes. Results presented in this work are obtained by averaging over $M = 100$ such simulations for each protein family.

D. Experimental evolution data

We use the data coming from the experimental evolution of the PSE-1 protein, published in [**?**].

- A word on the wild-type and what it does
- A brief summary of the experiment. What diversity is obtained after 20 rounds? What is the population size?
- what the corresponding pfam is, where we got the sequences (ref.)
- the alignment methodology

It has been noticed in [**?**] that taking into account the transition possibilities between amino acids allowed by the genetic code is important when describing short term evolutionary dynamics with generative models. In our framework, a natural way to include these is by using the symetric matrix $\mathbf{H}$ in the decomposition of Eq. 1. Terms of the $\mathbf{H}$ matrix do not affect the equilibrium distribution of the model, which thus remains generative, but influences the short term dynamics. Here, we simply counted the number of possibilities to transition from any amino acid to any other based on the genetic code, and we constructed the corresponding $\mathbf{H}$ matrix. The diagonal matrix remains given by the equilibrium probabilities of amino acids in the context of the sequence, as given by Eq. 4. We found that this substantially improves the results of the autoregressive reconstruction for the experimental evolution data.

E. iqtree

We run IQ-TREE using the `-asr`

- how I run iqtree + results of model finder

Supplementary Material: Reconstruction of ancestral protein sequences using autoregressive generative models

Appendix A: Autoregressive evolution model

- Show the simplified expression for H=1
- discuss irreversibility, with example

Appendix B: Supplementary figures

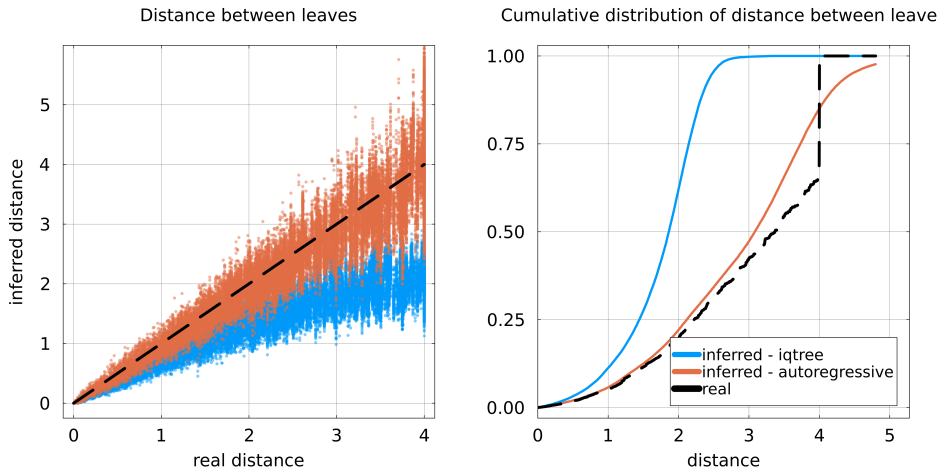

Figure S 1. Quality of branch length inference using data simulated with the autoregressive evolver. Inference is performed using the topology of a tree and leaf sequences generated using the autoregressive evolver. Two techniques are compared: IQ-TREE and the profile model corresponding to the autoregressive evolver. **Left**: inferred distance vs distance in the real trees for every pair of leaves. **Right**: Cumulative distribution of pairwise distance between leaves for the two inference methods and for the real trees. The discontinuity in the curve for the real tree is caused by the ultrametricity and fixed total height of the generated trees.

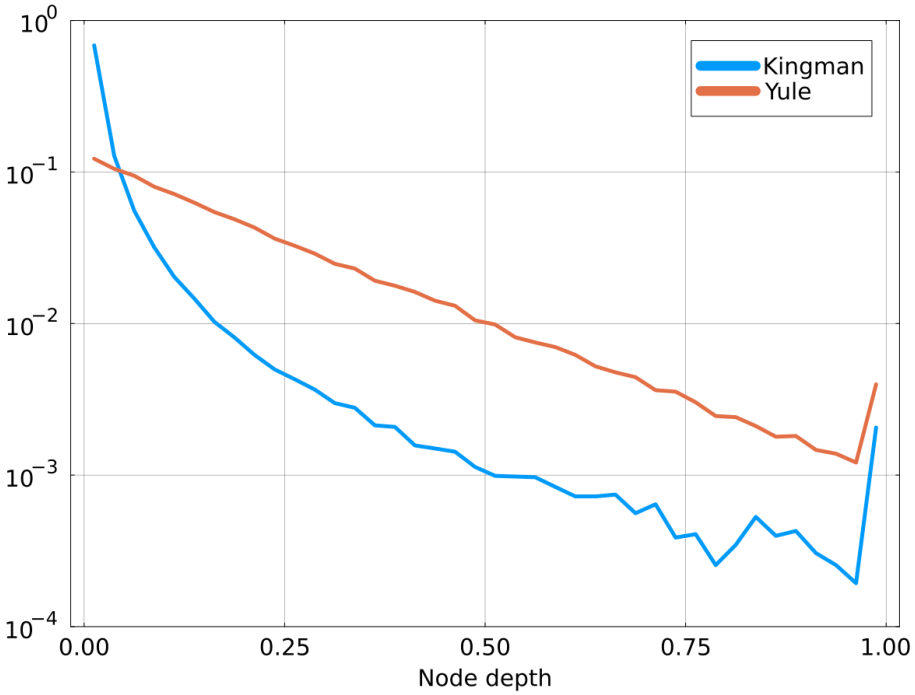

Figure S 2. Distribution of node depth for trees coming from the Kingman and Yule coalescents. Node depth is defined as the distance from a node to the closest leaf. Data is obtained by sampling several trees from each coalescent. Heights of trees are normalized to one. The Kingman process concentrates most of the nodes in close vicinity to the leaves, while the Yule process spreads them more evenly.