Characterization of selective pressures acting on protein sites with Deep Learning

Estelle Bergiron[1], Luca Nesterenko[1], Julien Barnier[1], Philippe Veber[1], and Bastien Boussau[1,*]

[1]Univ Lyon, Univ Lyon 1, CNRS, VetAgro Sup, UMR5558, Laboratoire de Biométrie et Biologie Evolutive, F-69100, Villeurbanne, France

April 22, 2026

Abstract

It is often useful, in the field of molecular evolution, to identify the selective pressures acting on a particular site of a protein to better understand its function. This is typically done with likelihood-based approaches applied to codon sequences in a phylogenetic context. However, these approaches are computationally costly. Here we adapt a linear transformer neural network architecture, which has been shown to be able to reconstruct accurate pairwise distances from sequence alignments, to identify selective pressures acting on individual amino acid sites. We design different versions of the architecture and train and test them on simulations. We compare the results of one of our best models to state-of-the-art likelihood-based methods and find that it outperforms it when it is applied to data that resemble its training data, but that it performs less well when applied to datasets that do not resemble the ones the model has been trained on. In all cases, our approach operates at a fraction of the computational cost of likelihood-based methods. These results suggest that such a neural network architecture can compare very favorably to state-of-the-art approaches to characterize selection pressures acting on coding sequences, but that it must be trained on datasets representative of empirical data.

1 Introduction

Deep learning approaches have been very successful in a range of scientific areas, and notably in genomics and biochemistry (Avsec et al., 2021; Jumper et al., 2021; Brandes et al., 2023). In recent years, they have begun to be applied in phylogenetics (Mo et al., 2024; Braichenko et al., 2025). First, they were used to reconstruct phylogenies with four taxa (e.g., Suvorov et al., 2019; Zou et al.,

2020) but since then new models have been designed to work with arbitrary numbers of sequences (Nesterenko et al., 2022, 2025; Zhang et al., 2025; Blassel et al., 2026). In particular, the Phyloformer architecture (Nesterenko et al., 2025) was shown to be able to provide accurate estimates of the phylogenetic history of a simulated sequence alignment. Specifically, it performed very well with models of sequence evolution that include co-evolution, but its accuracy decreased as the number of sequences increased. Phyloformer is a deep neural network that takes in input an amino acid sequence alignment and infers a distance matrix, from which a phylogenetic tree can be obtained via existing methods. The architecture is based on several transformer blocks (Vaswani et al., 2017), implementing a custom linear attention mechanism that takes into consideration correlations between sites and between sequences to yield the pairwise distance estimates.

We reasoned that such a network could provide estimates of selective regimes operating on protein sites, instead of estimates of distances between sequences. Inference of selective pressures acting on individual sites is usually based on the d_N/d_S ratio between synonymous and non-synonymous codon substitution rates (Goldman and Yang, 1994; Muse and Gaut, 1994). When at a site $d_N/d_S < 1$, the ratio is indicative of negative or purifying selection; when $d_N/d_S \approx 1$, of neutral evolution; when $d_N/d_S > 1$, of positive selection. Methods that estimate site-wise d_N/d_S are computationally expensive and could be advantageously replaced by a deep-learning approach. In support of this idea, Silvestro et al. (2024) have reported highly accurate substitution rates inferred from sequence alignments using a custom-designed recurrent neural network. Further, West et al. (2025) developed a convolutional neural network that can estimate whether a gene alignment with 8 sequences is under positive selection or not. These works demonstrate that deep learning approaches could provide useful alternatives to likelihood-based approaches for inference of selection.

In this paper, we report our efforts to build upon the Phyloformer architecture to classify protein sites as under neutral evolution or negative or positive selection. We train our network, SelRegAA, on simulations of protein alignments. We choose to simulate sequence evolution in the Mutation-Selection framework (Yang and Nielsen, 2008; Rodrigue et al., 2010), in which each codon evolves according to a DNA substitution matrix along with a vector of amino acid finess values. In this model, synonymous and non-synonymous substitutions are the outcome of the mutation-selection process, where different non-synonymous substitutions may be selected differentially. This is in contrast to standard d_N/d_S models (Goldman and Yang, 1994; Muse and Gaut, 1994), which cannot distinguish between accessible non-synonymous substitutions. In the mutation-selection framework, positive selection is simulated with the Persistent Positive Selection model of Tamuri and dos Reis (2021), which enforces that the fittest amino acid is always different from the current state. Overall, our choice to use the Mutation-Selection framework is motivated by its finer description of the process of coding sequence evolution.

We investigate the performance of our network, its sensitivity to the parameters of the simulation, and its statistical calibration by modifying its structure

and size. We compare its performance to standard likelihood-based approaches, operating in the d_N/d_S or the MutSel frameworks. When simulations match the data used to train the network, we find that SelRegAA performs with an accuracy superior to standard approaches, while being orders of magnitude faster. However, the performance of SelRegAA decreases if it is applied to data that does not look like the data it has been trained on, achieving results similar to standard approaches. These results underline the importance of model and parameters choice when training a neural network on simulations. We conclude from these results that, in the future, a linear transformer network, operating on codon sequences, and trained on data representative of the range of alignments observed empirically, could perform very well, at a fraction of the cost of likelihood-based approaches.

2 Methods

The code for building, training, and testing the neural network is available here: `https://gitlab.in2p3.fr/deelogeny/wp4/selregaa`.

2.1 Neural network architecture

Our neural network is based on Phyloformer (Nesterenko et al., 2025), with two main differences (fig. 1). First, we changed the output of the network. The original one outputs a distance matrix between sequences whereas in this setting we need site-wise annotations. Therefore, SelRegAA outputs a vector of 3 class probabilities per site of the alignment. The loss is computed as the cross-entropy between the output class probabilities and the true site-wise selection regime. Second, SelRegAA performs attention between sequences whereas Phyloformer performs attention between pairs of sequences.

The input to the network is a one-hot encoding of an amino acid sequence alignment. Each amino acid in each sequence is thus transformed into a vector of size 21, corresponding to 20 amino acids, plus a gap character. These vectors are then converted into vectors of size 32, corresponding to the embedding dimension (see Supplementary Fig. 1). The value of these embedding vectors is updated as they traverse the network and its 6 attention blocks (Vaswani et al., 2017). At the end, the embedding vectors are converted into vectors of size 3, averaged per site and across sequences. These vectors of size 3 per site are used as predictors for neutral evolution, negative selection, and persistent positive selection.

2.2 Training algorithm

We trained our neural network with the AdamW optimizer with a learning rate of 0.001. We also used a scheduler with a 5 epochs warmup period followed by an exponential decay with a parameter γ of 0.99. The network was trained for a maximum of 200 epochs with an early stopping criterion, halting training if

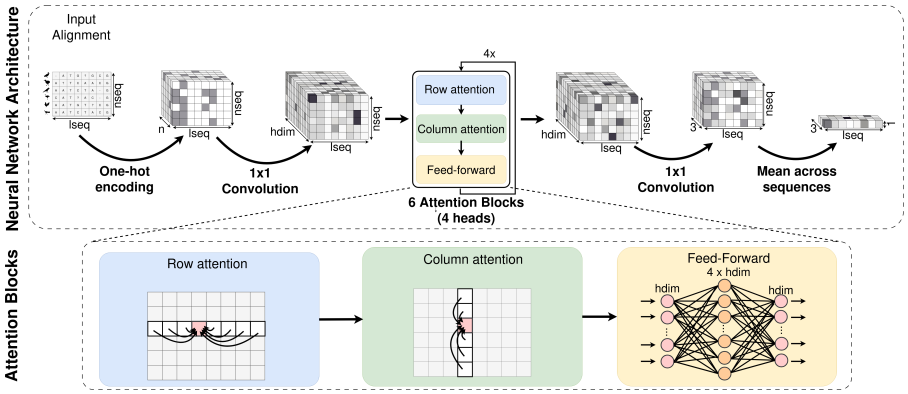

Figure 1: Structure of the base network. Top row: general architecture, from amino acid sequence encoding (left) to site-wise prediction of selective regimes (right). Bottom row: description of an attention block with 4 heads. lseq: number of sites in the alignment. nseq: number of sequences in the alignment. n: vector size after one-hot encoding (here, 21, for the 20 amino acids plus the gap character).

the validation loss value did not go down for 5 consecutive epochs. The network was trained with 10000 alignments, 90% of which for training, and 10% for validation.

2.3 Simulation protocol

Our simulation involves first simulating trees, and then simulating alignments along the trees. Trees were simulated as in Nesterenko et al. (2025). Briefly, ultrametric tree topologies were simulated using a Birth-Death process. Branch lengths were then altered to introduce heterogeneities between root-to-tip distances, and the trees were rescaled to match empirical tree diameters observed in the HOGENOM database (Penel et al., 2009) and the RaxML-Grove database (Höhler et al., 2021). Then, codon alignments were simulated along the phylogenies using the sequence simulator Pastek (https://gitlab.in2p3.fr/pveber/pastek; tag=selregaa), using the `multiselreg` option. With this option, Pastek takes as arguments an effective population size, proportions of sites under negative or positive selection and neutral evolution, parameters of the nucleotide mutation matrix and of the insertion-deletion process, and alignment size. We use uniform echangeability and equilibrium frequencies for the nucleotide mutation matrix (*i.e.*, the Jukes-Cantor model (Jukes and Cantor, 1969)). For each alignment, an effective population size parameter is drawn from a uniform distribution between 0.5 and 5, and applied to all sites. For each alignment, the proportions of sites (in percents) under neutral evolution and under positive selection are both drawn from a Poisson distribution of rate 5 while the remaining sites are simulated under negative selection. For each alignment, gaps are simulated with a deletion rate drawn from $0.05, 0.1, 0.5$ with weights $0.15, 0.7, 0.15$, and according to the TKF91 model (Thorne et al., 1991). An insertion rate equal

to 0 was chosen to avoid the burden of having to choose a selective regime for sites that appear later than at the root. Pastek implements the Gillespie algorithm (Gillespie, 1977) to simulate sequences, and works as follows. For each site, Pastek randomly draws a selective regime category. All sites are simulated according to the Mutation-Selection model (Yang and Nielsen, 2008; Rodrigue et al., 2010). Neutral sites are simulated by assuming a flat amino acid profile, where all amino acids have the same fitness. Negatively selected sites are simulated with an amino acid profile drawn from 263 empirically obtained profiles (Duchemin et al., 2023). Positively selected sites are simulated by drawing an amino acid profile as for negatively selected sites, but with an additional parameter Z_k to model persistent escape from the current amino acid (Tamuri and dos Reis, 2021). This parameter tends to increase the rate of substitutions away from the current state, irrespective of the current state. In our simulations, we used $Z_k = 10$, generating strong persistent positive selection; in their analysis of 3 gene alignments, Tamuri and dos Reis (2021) found that the Z values could reach values of more than 15, although rarely. To test for the sensitivity of the network to training parameters (see section 3.3.3), we also used another set of amino acid profiles, estimated from mammalian protein alignments (Latrille et al., 2024). From this set, we randomly picked two samples of 250000 profiles (`profiles0` and `profiles1`) from 1000 alignments. Pastek can produce codon alignments and the corresponding amino acid alignments. The amino acid alignments were used to train and test the neural networks and to run the PPS test (Tamuri and dos Reis, 2021), and the codon alignments were used with CODEML (Yang, 2007) and Godon (Davydov et al., 2017). The resulting alignments are 250 sites long and contain 50 sequences.

2.4 Comparison between SelRegAA and likelihood-based methods

To compare results and computing time, we applied CODEML, from PaML version 4.10.6, to 10 different alignments. Two sets of 10 alignments were produced, corresponding to two different amino-acid profiles, the default 263 profiles from Duchemin et al. (2023) (`default263`) and one of our two sets of profiles estimated from mammalian alignments (`profiles0`).

Running CODEML CODEML was applied first under the M2 model (`NSSites = 2`), then under the M8 model (`NSSites = 8`). Site-wise prediction of selection regimes was made using the Bayes Empirical Bayes method Yang et al. (2005). Data, configuration and results of each CODEML evaluation are available in the `CODEML` folder of this paper git repository.

Running Godon Godon was applied under the M8 model to the same set of alignments. Sites under positive selection were identified using the Bayes Empirical Bayes approach.

Running the PPS test The persistent positive selection test was performed as described by Tamuri et al., using swmutsel. First, mutation parameters (nucleotide frequencies and transition/transversion ratio) and branch lengths were first estimated for each alignment using CODEML. Then, a null mutation-selection model, without persistent positive selection, was fitted using swmutsel, with the mutation parameters and branch lengths fixed. Site-specific amino acid fitness profiles were estimated with Dirichlet regularization ($\alpha = 0.01$). An alternative model with persistent positive selection was fitted at each site using swmutsel with Dirichlet regularization ($\alpha = 0.01$) and an exponential penalty applied to the PPS parameter ($\lambda = 0.001$). A likelihood ratio was then calculated for each site. Statistical significance was assessed using parametric bootstraping (Cox test): sequence alignments were simulated under the null model using swmutsel, and the PPS model was refitted to obtain an empirical null distribution of the likelihood ratio. P-values were computed from this distribution. Multiple testing correction was then performed separately for each alignment using the Benjamini-Hochberg procedure, and a significance threshold of 0.05 was applied.

2.5 Estimation of computing time

Networks were trained on one V100 GPU. To compare computing times, CODEML, Godon and our base model were run on a cluster of computers, containing Intel Haswell, Broadwell, Skylake and AMD Epyc processors, using one CPU per run. As a result, computations did not all run on the exact same architecture and cannot be compared accurately. However, the orders of magnitude differences in running times that we witnessed would likely remain if measured on a standardized configuration.

2.6 Performance metrics

To evaluate the performance of our networks, we report precisions, recalls and F_1 scores. The precision for a class is the proportion of correctly predicted sites among all the sites predicted by the network to belong to that class. The recall for a class is the proportion of correctly predicted sites among all the sites in the dataset belonging to that class. The F_1 score for a class is the harmonic mean of the precision and recall scores for that class. It ranges between 0 and 1.0, with 1.0 for a perfect prediction with a recall and a precision of 1.0. As a global performance metric we use a global F_1 score as an average of the F_1 scores of each class.

3 Results

Detailed results for each experiment are available at `https://deelogeny.pages.in2p3.fr/wp4/selregaa/`.

3.1 Network architecture and training

In this section we compare the performances of our base model to variants of it, where the training or the architecture of the network were changed.

3.1.1 Training parameters

Our base model was trained on 10000 alignments. We performed 4 independent trainings of our base network architecture to evaluate the reproducibility of our training. Supplementary Table 1 shows that all replicate trainings resulted in very similar performances, with F_1 scores ranging between 0.765 (model base 2) and 0.774 (model base 1). Models 3 and 4 obtained F_1 scores of 0.770. In the rest of the manuscript, we will use model base 1 as our default model, and will name it SelRegAA. Due to our stopping criterion, training times varied, with model base 1's training taking 1h43 minutes for 69 epochs, model base 2's training taking 54 minutes for 38 epochs, model base 3 1h10 for 45 epochs, and model base 4 1h47 for 49 epochs. This experiment suggests that differences of around 0.01 in F_1 scores can be observed between replicates. In terms of precision and recall, the largest differences are observed for classifying sites under positive selection, reaching around 0.04.

We explored the network's performance when it is trained on a larger number of sequences, i.e. 50000 alignments instead of 10000. After training on the larger dataset, the network obtains an F_1 score of 0.793, better than the 0.774 of base model 1/SelRegAA, both in validation. We also evaluated these two models on unseen test data (1000 alignments, 50 tips and 250 sites per alignment). The network trained on the larger dataset achieved an F1 score of 0.788, and SelRegAA obtained a score of 0.769.

We note that both precision and recall are improved for detecting positively selected sites with the larger training dataset size (Supp. Table 2). In the following tests, we train the variants of our network on 10000 alignments to avoid large computational footprints.

3.1.2 Varying the network architecture

Our base network uses attention between sequences and thus differs from Phyloformer, which works on pairs of sequences to provide accurate estimates of pairwise distances. Attempts at working at the sequence level to reconstruct pairwise distances had been much less accurate (Nesterenko et al., 2022). However, working on pairs of sequences entails a memory footprint that scales quadratically with the number of sequences in the alignment, compared to linearly when working with sequences. We investigated the performance of an architecture operating on pairs of sequences, and compared it to SelRegAA, which works on individual sequences.

Table 1 shows that the pair model performs better than the sequence model, in particular for the detection of positive selection, with a gain in precision of 0.08, and in recall of 0.06. These gains appear meaningful since variation in these metrics between our four replicates (Sup. Table 1) were within a range of

0.04. They likely result from the gain in expressivity associated with considering pairs instead of individual sequences, and not from the number of parameters, because the sequence model has $102,051$ parameters, and the pair model has $68,323$ parameters. The pair model has fewer parameters because only 4 attention blocks were used instead of 6, to allow a batch size of 8 alignments during training. However, training the pair model was 15 times slower than the sequence model, and had a markedly larger memory footprint. In the following, we choose to focus on the sequence model because it has the best potential for scaling up to large datasets, where fast approaches are most needed.

Selection regime	Precision: SelRegAA	Precision: Pairs model	Recall: SelRegAA	Recall: Pairs model
Negative	0.96	0.96	0.99	0.99
Neutral	0.69	0.72	0.42	0.43
Positive	0.87	0.95	0.78	0.84

Table 1: A sequence level model performs less well than a costly pair-level model.

Size of the embedding The base network takes a sequence alignment as input, in which all sites are one-hot encoded into vectors of dimension 21. In turn, these vectors are embedded into vectors of size 32, that then pass through the attention blocks. We investigated the impact of changing the size of the embedding. Supplementary Table 3 shows that the size of the embedding has little influence on the performance of the network. Decreasing it to 16 instead of 32 results in a slightly lower F_1 score of 0.755, and increasing it to 48 results in the same F_1 score of 0.774.

Number of attention blocks and heads We investigated a range of network sizes, by changing the number of attention blocks and attention heads in our network.

Our base model has 6 attention blocks that each combine column and row attention mechanisms. We tested reducing this number to 2 and 4, or increasing it to 8. Supplementary Table 4 shows that the performance of the network is not very sensitive, if at all, to the number of attention blocks.

Inside each axial attention mechanism, we use several heads whose roles are to focus on different characteristics of the data. Our base model has 4 attention heads. We trained alternative models with 2 or 6 attention heads, keeping head size constant (size 8). As a result, embeddings become larger with more heads: the network with 2 heads uses embeddings of size 16, and the network with 6 heads uses embeddings of size 48. Supplementary Table 5 shows that the number of heads affects the performance of the network slightly. We observe a very weak decrease in performance when only 2 heads are used (F_1 score of

0.757 instead of 0.774 for base model 1/SelRegAA), but no improvement when 6 heads are used (F_1 score of 0.776).

The effects of row and column attentions The Phyloformer architecture performs attention across both sites and sequences in each attention block. To investigate the impact of each type of attention on site-wise classification accuracy, we trained two networks in which either attention had been deactivated and one network in which both attention mechanisms had been deactivated. Table 2 shows that both attention mechanisms are important. Attention across sites is important, as removing it degrades the F_1 score from 0.744 for SelRegAA to 0.538. Just removing the attention across sequences is enough to completely destroy the network's ability to predict site categories: the network then predicts that all sites belong to the "negative selection" category, because it is the category with the largest number of sites in the data.

Selection regime	SelRegAA	Both disabled	Across sequences disabled	Across sites disabled
Negative	0.96	0.90	0.90	0.93
Neutral	0.69	0.00	0.00	0.51
Positive	0.87	0.00	0.00	0.68

Precision

Selection regime	SelRegAA	Both disabled	Across sequences disabled	Across sites disabled
Negative	0.99	1.00	1.00	0.99
Neutral	0.42	0.00	0.00	0.08
Positive	0.78	0.00	0.00	0.38

Recall

Table 2: Effect of attention across sequences and across sites on precision and recall. Compared to SelRegAA (left), removing the two axial attention mechanisms degrades performance. When both attention mechanisms are disabled or when only attention across sites is used, the network fails to predict site categories (second and third columns). When only attention across sequences is used, the network's performance is decreased markedly (fourth column).

3.2 Sensitivity to dataset characteristics

3.2.1 Sensitivity to alignment size

Our network was trained on datasets with 50 sequences and 250 sites. We investigated its performance on alignments with larger numbers of sites and sequences. When trained and tested on alignments with 100 sequences instead of 50, we observe that the network's recall increases both for the neutral and positive selection categories, with a better precision for positive selection sites (Table 3).

Selection regime	Precision		Recall	
	100 sequences	SelRegAA	100 sequences	SelRegAA
Negative	0.96	0.96	0.99	0.99
Neutral	0.70	0.69	0.50	0.42
Positive	0.92	0.87	0.83	0.78

Table 3: The trained network has better performance when it is trained and tested on alignments with a larger number of sequences.

When trained on alignments with 400 sites instead of 250, the performance of the network does not improve (Supp. Table 6). A larger number of sites does not result in an improved prediction accuracy per site. Increasing the number of sites in the alignment can only be useful for more accurately accounting for the phylogenetic structure in the data, which does not seem to matter here. Conversely, since the prediction is made at the level of individual sites, a larger number of sequences provides more information per site, resulting in a higher performance. Interestingly, the higher performance obtained on alignments with 100 sequences does not translate to smaller alignments: the network trained on large alignments performs less well on alignments with 50 sequences than the network trained on smaller alignments (data not shown).

3.2.2 Weighting the loss to account for imbalance in the proportions of the categories

The training data was generated with 5% of the sites under neutral evolution, 90% under negative selection, and 5% under persistent positive selection, on average per alignment. This strong imbalance in the proportions of the different categories could lead the network to focus on accurately predicting sites under negative selection, with a poorer performance for the other two minor categories. To investigate the strength of this effect, we trained a network with a weighted loss on the same training data. This weighted loss effectively penalizes errors in the minor categories more than errors in the negative selection category, by weighting errors with the inverse of each class frequency. The weights were 0.378

for negative selection and 6.66 for the other two categories. Table 4 shows that the model trained with weighted loss has better recall in the minor categories and a higher precision in the most frequent category. However, the precision in the minor categories is markedly decreased. Overall, the performance of the weighted model is reduced, with an F_1 score of 0.671 compared to 0.744 for SelRegAA. This is not unexpected as the F_1 score depends on class balance, but in the following we keep our focus on the unweighted model SelRegAA.

Selection regime	Precision		Recall	
	SelRegAA	Weighted loss	SelRegAA	Weighted loss
Negative	0.96	0.99	0.99	0.85
Neutral	0.69	0.26	0.42	0.84
Positive	0.87	0.60	0.78	0.84

Table 4: Precision and recall values for SelRegAA (left), and for a model trained with weighted loss (right). The model trained with weighted loss has better recall in the minor categories, and a higher precision in the most frequent category.

We also trained a network with uniform proportions of the three selection regimes, and found that it performed better than SelRegAA on data simulated with uniform proportions, but less well than SelRegAA on data simulated as in SelRegAA's training data (See Supplementary section 8). Overall, the proportions of selection regimes used for training have a clear effect on the network's ability to predict selection regimes.

3.2.3 Sensitivity to the trees used

The trees in our training and testing datasets have been simulated as in Nesterenko et al. (2025). Briefly, all of these trees have 50 leaves, and have been scaled to match the distribution of tree diameters observed in the HOGENOM database Penel et al. (2009). We investigated the sensitivity of SelRegAA to different tree topologies. We obtained 4 empirical phylogenies used in Duchemin et al. (2023). The `besnard2009` tree is a 79 leaf phylogeny of the Cyperaceae family of flowering plants, `online_rodent` is a 32 leaf phylogeny of rodent species, `orthomam` is a 116 leaf phylogeny of mammalian species, and `rubisco` is a 179 leaf phylogeny of the Amaranthaceae family of flowering plants. The plant phylogenies have been reconstructed from single gene alignments, while the other two phylogenies have been reconstructed from concatenates. Table 5 presents the F_1 scores for a variety of experiments: we ran inference on the empirical trees (keeping their original branch lengths), on the same trees after rescaling branch lengths to match the branch length of trees used during training, and actual simulated trees with the same number of leaves than the empirical trees. Performance on alignments simulated along the empirical trees (column *Empirical tree*) is systematically lower than the performance obtained on validation

data during training (F_1 score on test data of 0.769). Within the empirical trees, we observe that performance is better when the number of sequences is larger, which is consistent with our results obtained above (Table 3). Rescaling the empirical trees to match the tree diameters used in our training simulations improves the performance of the network (column *Scaled empirical tree*). Performance on alignments simulated on new trees with the same number of leaves as the empirical phylogenies, but keeping the scaling as in the training data, results in similar performance (column *Simulated tree*). These results show that both the number of sequences and the scale of the tree have large impacts on the performance of the network, but the topology of the trees, whether they are simulated or empirical, does not seem to matter.

Tree	**Empirical tree**	**Scaled empirical tree**	**Simulated tree**
besnard2009	0.711	0.796	0.793
online_rodent	0.624	0.712	0.729
orthomam	0.615	0.804	0.807
rubisco	0.586	0.834	0.818

Table 5: Performance (F_1 score) of the network on different phylogenies of different lengths. The F_1 score obtained on test data during training is 0.769. When empirical trees are scaled to match the scale of training data, performance improves (column *Scaled empirical tree*). Similarly, performance on new simulated phylogenies with numbers of tips that match the empirical trees, but scale that matches the training data, is also improved (column *Simulated tree*).

3.3 Comparison to standard likelihood-based methods

3.3.1 Performance on data similar to the training data

We compared our network to two d_N/d_S methods, the standard method CODEML (Yang, 2007), and a faster implementation, Godon (Davydov et al., 2017), as well as a method to detect positively selected sites based on the Mutation-Selection framework (Tamuri and dos Reis, 2021). The two d_N/d_S methods are at a disadvantage here, because their model does not fit the model that was used to simulate the data. CODEML and Godon implement the same models, and in our tests obtained the same results. We therefore only show results for CODEML, but discuss run times for both implementations. The persistent positive selection test of Tamuri and dos Reis (Tamuri and dos Reis, 2021) fits the simulation model, and therefore provides an interesting comparison to our network. Tests were run on 10 newly simulated alignments, that the base network had not seen before, but simulated with the same pipeline as the training data.

CODEML was used on codon sequences to predict whether a site is under negative selection, neutral evolution or positive selection with model M2a (Nielsen and Yang, 1998), and was used to predict positive selection only using the M8 model (Yang et al., 2000). In both cases, prediction relied on the Bayes

Empirical Bayes method (Yang et al., 2005). Similarly, the persistent positive selection (PPS) test Tamuri and dos Reis (2021) was also applied to predict sites under positive selection only. Overall, SelRegAA has better results than CODEML's two models (Table 6 ; Supp. Fig. 2), and than the PPS test. In particular, CODEML's precision for the Neutral evolution and Positive selection classes is much lower than SelRegAA's precision. CODEML's recall however is better than SelRegAA's for the Neutral evolution class, but lower for the two other classes. Overall, for a given recall rate, SelRegAA often offers a better precision than CODEML (Supp. Fig. 2). Prediction of positive selection with CODEML's model M8 is slightly better than with model M2, but still dominated by SelRegAA. The PPS test for predicting sites under positive selection has very good recall, but very bad precision. This might be linked to optimization difficulties. Indeed, out of the 2500 sites (10 alignments of 250 sites each), the model did not converge for 55 sites, indicating that computation of the PPS test may be hampered by numerical instabilities. Performance metrics were computed on the remaining 2445 sites.

Selection regime	SelRegAA	CODEML M2	CODEML M8	PPS test
Negative	0.97	0.99	NA	NA
Neutral	0.63	0.12	NA	NA
Positive	0.89	0.37	0.41	0.07

Precision

Selection regime	SelRegAA	CODEML M2	CODEML M8	PPS test
Negative	0.98	0.62	NA	NA
Neutral	0.43	0.79	NA	NA
Positive	0.88	0.78	0.78	0.92

Recall

Table 6: Precision and recall values for our basemodel, CODEML model M2 Bayes Empirical Bayes predictions, CODEML model M8 Bayes Empirical Bayes predictions, and the Persistent Positive Selection test on 10 alignments simulated as during training.

Comparison of computing times We compared the inference times of the base network, the PPS test, CODEML and Godon, a faster implementation of the codon models implemented in CODEML. Godon was used to perform inference with model M8. Prediction of selection regimes for the 10 alignments required about 3h on 10 CPUs with CODEML, and 20 minutes on 10 CPUs with Godon. This difference is consistent with the range of speed ups reported

in Davydov et al. (2017). Computation with the PPS test involves site-wise computations, including 100 replicate computations per individual site to assess significativity. We estimate that running the PPS test for all 10 alignments took the equivalent of more than 700 days of computation, on 10 CPUs. For SelRegAA, inference for all 10 alignments took 5 seconds on a single CPU. The different computations were run on different computers on a cluster; however, the differences in computing time are such that there is no doubt that inference with SelRegAA is orders of magnitude faster than inference with Godon, itself about 10 times faster than CODEML, itself orders of magnitude faster than the PPS test.

3.3.2 Calibration

We investigated whether the probabilities output by SelRegAA and by CODEML model 0 are calibrated. A probability p of an event is calibrated if the event occurs a fraction p of the time. Our network outputs for each site a probability that it is under negative, neutral, or positive selection. CODEML uses Bayes Empirical Bayes estimates to compute probabilities for each site to be in one of the three selective regimes Yang et al. (2005). We used the results on the 10 alignments to compute the frequency with which a prediction is correct and compared it to the probabilities output by the two methods.

Fig. 2 shows that the network's output probabilities are well calibrated on data similar to its training dataset. Although the neutral and positive categories are much less frequent in the training data than the negative selection category, all three categories are predicted with good calibration. In contrast, CODEML's M2a Bayes Empirical Bayes probabilities are not well calibrated in all three categories. CODEML is overly conservative for neutral and positive categories and, in reverse, overestimates the probability that a site is in the negative selection category. The poor calibration of CODEML models M2a is probably because it is measured on Mutation-Selection simulations that do not correspond to model M2a.

3.3.3 Performance on simulations using different amino acid profiles

Our method was trained and tested on simulations using a limited set of 263 experimentally determined amino acid fitness profiles. We tested its performance on alignments simulated from independently-obtained amino acid profiles and compared it to CODEML. We retrieved amino acid profiles estimated in Latrille et al. (2024) and randomly picked two samples of 250000 profiles (`profiles0` and `profiles1`) from 1000 alignments.

Table 7 shows that the performance of the network suffers markedly when tested on alignments simulated with a different set of profiles, with an F_1 score dropping from 0.769 to about 0.575. In particular, the two classes with the smallest numbers of sites are much less well detected, both in terms of precision and recall.

A network trained on alignments simulated with `profiles0` has reduced

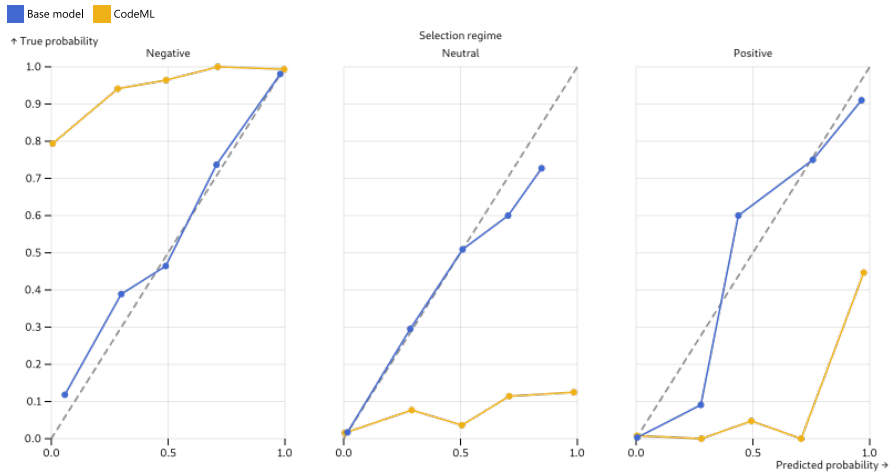

Figure 2: Calibration of SelRegAA and of CODEML. On the x axis, predicted probability that a site is under negative selection (left panel), neutral evolution (middle), or positive selection (right), is compared to the proportion with which a site was indeed in the corresponding selection regime (y axis). A calibrated method is expected to generate points along the y=x regression line.

performance on alignments simulated with the 263 experimentally determined amino acid fitness profiles (F_1 score of 0.676) (Supp. Table 9). However, performance does not vary whether test alignments have been simulated with `profiles0` or `profiles1`, which both were inferred from large numbers of sites in protein alignments from mammals.

Selection regime	Precision			Recall		
	Default 263	Profiles 0	Profiles 1	Default 263	Profiles 0	Profiles 1
Negative	0.96	0.96	0.96	0.99	0.97	0.97
Neutral	0.68	0.52	0.53	0.41	0.15	0.14
Positive	0.87	0.44	0.45	0.78	0.66	0.68

Table 7: Performance of SelRegAA is markedly decreased on alignments simulated with a different set of amino acid profiles.

On 10 alignments simulated with `profiles0`, comparison between CODEML and SelRegAA trained on the 263 default profiles shows that the performance gap between the two models has decreased with F_1 scores of 0.539 and 0.548 respectively (compared to 0.492 and 0.791 on data simulated with the 263 default profiles), although SelRegAA keeps a small edge. The two methods, however,

Selection regime	SelRegAA	CODEML	SelRegAA	CODEML
	Precision		Recall	
Negative	0.96	0.96	0.99	0.84
Neutral	0.59	0.18	0.09	0.74
Positive	0.48	0.53	0.56	0.36

Table 8: Comparison between CODEML and SelRegAA (trained on the default 263 profiles) on alignments simulated with `profiles0`.

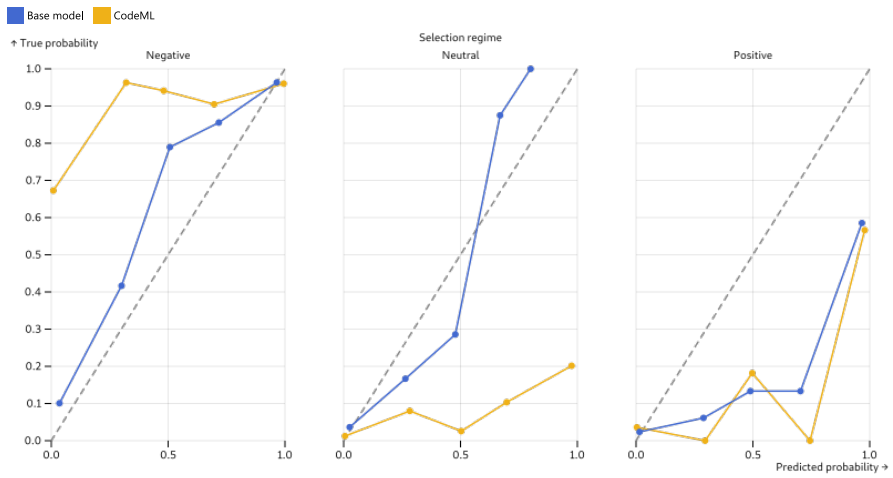

Figure 3: Calibration of CODEML and the basemodel on simulations using amino acid profiles different from the training data

exhibit different behaviours (Table 8). For example, they have opposite positions on the precision-recall trade-off for the neutral category, with SelRegAA showing a bad recall but better precision, while CODEML shows a bad precision with better recall.

This drop in performance for SelRegAA is also seen in the calibration of the probabilities (Fig. 3). In particular, SelRegAA is now overly conservative in its prediction of sites in the positive selection category, with a behaviour that is now similar to CODEML.

Discussion

In this manuscript, we have presented our work to develop and train a neural network to predict selective regimes at amino acid alignment sites. We have characterized how the performance of the network varied with features of the

network architecture and with characteristics of the training data. These data can be used as guidance for training these types of neural networks for site-wise inference on multiple sequence alignments. In particular, we found that varying the embedding size, the number of attention blocks or attention heads had little effect on performance. However, handling pairs of sequences and using larger training datasets improved our results. These two approaches however have shortcomings: training on a larger dataset entails a larger computational footprint, and the model working with pairs of sequences both requires longer training and has a much larger memory footprint than the model working with sequences.

The results obtained by SelRegAA on simulated data are very promising. Comparison with CODEML showed that the network achieved equal or better performance, at a fraction of the computing time. SelregAA's good performance is surprising because, contrary to CODEML, it does not have access to the phylogeny of the sequences, and to synonymous changes, which provide information on the neutral rate of evolution. This suggests that a similar network, trained on codon sequences, could achieve even higher performance. However we believe SelregAA's good performance is the result of (i) a mismatch between the model used to simulate test data and the model underlying CODEML and (ii) the fact that the network performs Bayesian inference and so benefits from the information contained in the prior. Regarding (i), one particularity of our simulations is that they rely on the mutation-selection model, not on a d_N/d_S model. This could cause a decreased performance of CODEML, as the simulated data does not fit its underlying model. However, the mutation-selection model appears to be more realistic than d_N/d_S models (Jones et al., 2019; Rodrigue et al., 2020). Our results could therefore be indicative of expected performances on empirical data. Further, the PPS test (Tamuri and dos Reis, 2021), which relies on the Mutation-Selection framework, also obtained results that were inferior to SelRegAA. In addition, the probabilities output by the network were better calibrated than the Bayes Empirical Bayes estimates provided by CODEML (Yang et al., 2005). However, the results depended on whether the test data matched the training data. While SelRegAA obtained much better results than CODEML on test data that matched its training, performance was reduced when the test data was simulated with different amino acid profiles. This result is illustrative of (ii): the fact that training the network on simulated data amounts to teaching it to perform Bayesian inference (Nesterenko et al., 2025). For this reason, the neural network operates with a correctly specified prior, which gives it an advantage over CODEML.

Our results obtained with different amino acid profiles, with different phylogenies, or with different proportions of selection regimes, show that the performance of the model is very dependent on the match between training and testing data. Once again, we suspect this is because the training of a neural network on simulations amounts to building a model that performs Bayesian inference (Nesterenko et al., 2025). The resulting network therefore encapsulates a prior distribution over all parameters of the simulation model, which can result in altered performance if there are mismatches between the model that

generated the training data and the model that generated the test data. This means that when analyzing empirical data, much attention must be devoted to generating realistic data for the training. Here we have anchored the tree diameters on an empirical tree distribution, and the amino acid profiles on empirical data. Unfortunately however, it is certain that the empirical distributions that we have used do not cover the whole range of parameter values that would be needed to fit different types of alignments. Alignments from different groups of species will differ in their size, their evolutionary depth, their GC content, or the selective pressures that have applied on them. Training a network that could perform equally well on any type of biological alignment is an important challenge. Perhaps a feasible approach would be to train several networks to cover different areas of the parameter space, and develop a method to choose the most relevant network depending on characteristics of the alignment under study. Another solution would be to train a network *de novo* for a given dataset. Our results show that the training of our network on a GPU plus the inference on 10 alignments on a single CPU took about the same amount of time as running CODEML on these 10 alignments on 10 CPUs. When inference needs to be performed at the genomic scale, for thousands of gene alignments, training de novo, on simulations that resemble the empirical data, provides a lower footprint than using CODEML. Our open source code can be deployed for training or fine-tuning a model on new simulations. However, choosing parameter values that generate data sets that are similar to empirical data remains an area of research (see, e.g., for gap patterns, (Ashkenazy et al., 2017)).

This work further suggests that similar neural networks could be used to predict other features of amino acid sites. For instance, a network could be used to predict sites undergoing shifts in the direction of selection associated to phenotype changes (Duchemin et al., 2023). To do so, the network should take as additional input phenotypic states at each leaf of the tree, or a phylogenetic tree annotated with phenotypic states at each node. Beyond site-wise prediction, similar networks could also predict model parameters that apply to an entire alignment, such as a matrix of nucleotide substitutions, or an average d_N/d_S. Because the transformer architecture at the heart of our model can capture dependencies between sites of the alignment, one could also design a model to learn parameters of models that capture co-evolution between sites, such as the Direct Coupling Analysis model (Morcos et al., 2011).

Conclusion

In this work, we have shown that a neural network trained on simulated amino acid sequences could predict selection pressures acting on individual protein sites. It performed better than a state-of-the-art method across datasets that resemble its training data, but can perform less well when there is a difference between training and test data. Looking ahead, this shows that a similar network working on codon sequences, and trained on data that resembles empirical data, could markedly outperform maximum likelihood approaches in speed and

accuracy to characterize selective pressures acting on sites. Further, it paves the way for the inference of other aspects of molecular evolution.

Acknowledgements

This work was funded by the Agence Nationale de la Recherche (PIECES ANR-20-CE45-0017, and Deelogeny ANR-23-CE45-0027). This work was performed using the computing facilities of the CC LBBE/PRABI and of the CC IN2P3.

References

H. Ashkenazy, E. Levy Karin, Z. Mertens, R. A. Cartwright, and T. Pupko. Spartaabc: a web server to simulate sequences with indel parameters inferred using an approximate bayesian computation algorithm. *Nucleic Acids Research*, 45(W1):W453–W457, 07 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx322. URL https://doi.org/10.1093/nar/gkx322.

Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *Nature Methods*, 18(10):1196–1203, Oct. 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01252-x. URL http://www.nature.com/articles/s41592-021-01252-x. Number: 10 Publisher: Nature Publishing Group.

L. Blassel, N. Sauvage, P. Barrat-Charlaix, B. Boussau, N. Lartillot, and L. Jacob. Likelihood-free inference of phylogenetic tree posterior distributions, 2026. URL https://arxiv.org/abs/2510.12976.

S. Braichenko, R. Borges, and C. Kosiol. Phylogenetic Methods Meet Deep Learning. *Genome Biology and Evolution*, 17(10):evaf177, Sept. 2025. ISSN 1759-6653. doi: 10.1093/gbe/evaf177.

N. Brandes, G. Goldman, C. H. Wang, C. J. Ye, and V. Ntranos. Genome-wide prediction of disease variant effects with a deep protein language model. *Nature Genetics*, 55(9):1512–1522, Sept. 2023. ISSN 1546-1718. doi: 10.1038/s41588-023-01465-0. URL http://www.nature.com/articles/s41588-023-01465-0. Publisher: Nature Publishing Group.

I. I. Davydov, M. Robinson-Rechavi, and N. Salamin. State aggregation for fast likelihood computations in molecular evolution. *Bioinformatics*, 33(3): 354–362, Feb. 2017. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw632. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5408795/.

L. Duchemin, V. Lanore, P. Veber, and B. Boussau. Evaluation of Methods to Detect Shifts in Directional Selection at the Genome Scale. *Molecular Biology and Evolution*, 40(2):msac247, Feb. 2023. ISSN 1537-1719. doi: 10.1093/molbev/msac247. URL https://doi.org/10.1093/molbev/msac247.

D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977. doi: 10.1021/j100540a008. URL https://doi.org/10.1021/j100540a008.

N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular Biology and Evolution*, 11(5):725–736, Sept. 1994. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040153. URL https://doi.org/10.1093/oxfordjournals.molbev.a040153.

D. Höhler, W. Pfeiffer, V. Ioannidis, H. Stockinger, and A. Stamatakis. RAxML Grove: an empirical phylogenetic tree database. *Bioinformatics*, 38(6): 1741–1742, Dec. 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab863. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8896636/.

C. T. Jones, E. Susko, and J. P. Bielawski. Looking for Darwin in Genomic Sequences: Validity and Success Depends on the Relationship Between Model and Data. In M. Anisimova, editor, *Evolutionary Genomics: Statistical and Computational Methods*, Methods in Molecular Biology, pages 399–426. Springer, New York, NY, 2019. ISBN 978-1-4939-9074-0. doi: 10.1007/978-1-4939-9074-0_13. URL https://doi.org/10.1007/978-1-4939-9074-0_13.

T. H. Jukes and C. R. Cantor. Evolution of Protein Molecules. In *Mammalian Protein Metabolism*, pages 21–132. Elsevier, 1969. ISBN 978-1-4832-3211-9. doi: 10.1016/B978-1-4832-3211-9.50009-7. URL https://linkinghub.elsevier.com/retrieve/pii/B9781483232119500097.

J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, Aug. 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.

T. Latrille, J. Joseph, D. A. Hartasánchez, and N. Salamin. Estimating the proportion of beneficial mutations that are not adaptive in mammals. *PLOS Genetics*, 20(12):e1011536, Dec. 2024. ISSN 1553-7404. doi: 10.1371/journal.pgen.1011536. URL https://journals.plos.org/plosgenetics/article?id=10.1371/journal.pgen.1011536. Publisher: Public Library of Science.

Y. K. Mo, M. W. Hahn, and M. L. Smith. Applications of Machine Learning in Phylogenetics. *Molecular Phylogenetics and Evolution*, page 108066, Mar. 2024. ISSN 1055-7903. doi: 10.1016/j.ympev.2024.108066. URL https://www.sciencedirect.com/science/article/pii/S1055790324000587.

F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, Dec. 2011. doi: 10.1073/pnas.1111471108. URL http://www.pnas.org/doi/10.1073/pnas.1111471108. Publisher: Proceedings of the National Academy of Sciences.

S. V. Muse and B. S. Gaut. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the

chloroplast genome. *Molecular Biology and Evolution*, 11(5):715–24, Sept. 1994. URL http://mbe.oxfordjournals.org/cgi/reprint/11/5/715.

L. Nesterenko, B. Boussau, and L. Jacob. Phyloformer: towards fast and accurate phylogeny estimation with self-attention networks. *bioRxiv*, June 2022. doi: 10.1101/2022.06.24.496975. URL http://biorxiv.org/lookup/doi/10.1101/2022.06.24.496975.

L. Nesterenko, L. Blassel, P. Veber, B. Boussau, and L. Jacob. Phyloformer: Fast, Accurate, and Versatile Phylogenetic Reconstruction with Deep Neural Networks. *Molecular Biology and Evolution*, 42(4):msaf051, Mar. 2025. ISSN 0737-4038. doi: 10.1093/molbev/msaf051. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11965795/.

R. Nielsen and Z. Yang. Likelihood models for detecting positively selected amino acid sites and applications to the hiv-1 envelope gene. *Genetics*, 148 (3):929–936, 03 1998. ISSN 1943-2631. doi: 10.1093/genetics/148.3.929. URL https://doi.org/10.1093/genetics/148.3.929.

S. Penel, A.-M. Arigon, J.-F. Dufayard, A.-S. Sertier, V. Daubin, L. Duret, M. Gouy, and G. Perrière. Databases of homologous gene families for comparative genomics. *BMC Bioinformatics*, 10(Suppl 6):S3, Jan. 2009. doi: 10.1186/1471-2105-10-S6-S3. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2697650/.

N. Rodrigue, H. Philippe, and N. Lartillot. Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proceedings of the National Academy of Sciences*, 107(10):4629–4634, Mar. 2010. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.0910915107. URL http://www.pnas.org/content/107/10/4629.

N. Rodrigue, T. Latrille, and N. Lartillot. A Bayesian Mutation–Selection Framework for Detecting Site-Specific Adaptive Evolution in Protein-Coding Genes. *Molecular Biology and Evolution*, 38(3):1199–1208, Oct. 2020. ISSN 0737-4038. doi: 10.1093/molbev/msaa265. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7947879/.

D. Silvestro, T. Latrille, and N. Salamin. Toward a semi-supervised learning approach to phylogenetic estimation. *Systematic Biology*, 73(5):789–806, 06 2024. ISSN 1063-5157. doi: 10.1093/sysbio/syae029. URL https://doi.org/10.1093/sysbio/syae029.

A. Suvorov, J. Hochuli, and D. R. Schrider. Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning. *Systematic Biology*, 69(2):221–233, Sept. 2019. ISSN 1063-5157. doi: 10.1093/sysbio/syz060. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8204903/.

A. U. Tamuri and M. dos Reis. A mutation–selection model of protein evolution under persistent positive selection. *Molecular Biology and Evolution*, 39(1): msab309, 10 2021. ISSN 1537-1719. doi: 10.1093/molbev/msab309. URL https://doi.org/10.1093/molbev/msab309.

J. L. Thorne, H. Kishino, and J. Felsenstein. An evolutionary model for maximum likelihood alignment of DNA sequences. *Journal of Molecular Evolution*, 33(2):114–124, Aug. 1991. ISSN 1432-1432. doi: 10.1007/BF02193625. URL https://doi.org/10.1007/BF02193625.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin. Attention is All you Need. *Advances in Neural Information Processing Systems*, 2017. doi: https://doi.org/10.48550/arXiv.1706.03762. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

C. West, C. R. Walker, S. Arasti, V. Vasilev, X. Xu, N. De Maio, and N. Goldman. Detecting interspecific positive selection using convolutional neural networks. *Molecular Biology and Evolution*, 42(7):msaf154, 06 2025. ISSN 1537-1719. doi: 10.1093/molbev/msaf154. URL https://doi.org/10.1093/molbev/msaf154.

Z. Yang. PAML 4: phylogenetic analysis by maximum likelihood. *Molecular Biology and Evolution*, 24(8):1586–1591, Aug. 2007. ISSN 0737-4038. doi: 10.1093/molbev/msm088.

Z. Yang and R. Nielsen. Mutation-Selection Models of Codon Substitution and Their Use to Estimate Selective Strengths on Codon Usage. *Molecular Biology and Evolution*, 25(3):568–579, Mar. 2008. ISSN 0737-4038. doi: 10.1093/molbev/msm284. URL http://academic.oup.com/mbe/article/25/3/568/1060947.

Z. Yang, R. Nielsen, N. Goldman, and A.-M. K. Pedersen. Codon-Substitution Models for Heterogeneous Selection Pressure at Amino Acid Sites. *Genetics*, 155(1):431–449, May 2000. ISSN 0016-6731, 1943-2631. URL http://www.genetics.org/content/155/1/431.

Z. Yang, W. S. Wong, and R. Nielsen. Bayes Empirical Bayes Inference of Amino Acid Sites Under Positive Selection. *Molecular Biology and Evolution*, 22(4):1107–1118, Apr. 2005. ISSN 0737-4038. doi: 10.1093/molbev/msi097. URL https://doi.org/10.1093/molbev/msi097.

X. Zhang, S. Ding, C. Yu, J. Zhao, and D. Bu. Accurate and Efficient Phylogenetic Inference through End-To-End Deep Learning. *Molecular Biology and Evolution*, 42(11):msaf260, Nov. 2025. ISSN 1537-1719. doi: 10.1093/molbev/msaf260. URL https://doi.org/10.1093/molbev/msaf260.

Z. Zou, H. Zhang, Y. Guan, and J. Zhang. Deep Residual Neural Networks Resolve Quartet Molecular Phylogenies. *Molecular Biology and Evolution*,

37(5):1495–1507, May 2020. ISSN 0737-4038. doi: 10.1093/molbev/msz307. URL https://doi.org/10.1093/molbev/msz307.