Deep Learning from Phylogenies for Diversification Analyses

Sophia Lambert, Jakub Voznica, Hélène Morlon

Deep Learning from Phylogenies for Diversification Analyses

Lambert Sophia[1†*], Voznica Jakub[2,3,4†*], Morlon Hélène[1]

[1]*Institut de Biologie de l'École Normale Supérieure, École Normale Supérieure, CNRS, INSERM, Université Paris Sciences et Lettres, 75005 Paris, FRANCE;*

[2]*Unité de Bioinformatique Évolutive - Département Biologie computationnelle, Institut Pasteur, Paris, FRANCE;*

[3]*Unité de Biologie Computationnelle, USR 3756 CNRS, Paris, FRANCE;*

[4]*Université Paris Cité, Paris, FRANCE;*

† These authors contributed equally to this work

** corresponding slambert@bio.ens.psl.eu and voznica.jakub@gmail.com*

ABSTRACT

Birth-death models are widely used in combination with species phylogenies to study past diversification dynamics. Current inference approaches typically rely on likelihood-based methods. These methods are not generalizable, as a new likelihood formula must be established each time a new model is proposed; for some models such formula is not even tractable. Deep learning can bring solutions in such situations, as deep neural networks can be trained to learn the relation between simulations and parameter values as a regression problem. In this paper, we adapt a recently developed deep learning method from pathogen phylodynamics to the case of diversification inference, and we extend its applicability to the case of the inference of state-dependent diversification models from phylogenies associated with trait data. We demonstrate the accuracy and time efficiency of the approach for the time constant homogeneous birth-death model and the Binary-State Speciation and Extinction model. Finally, we illustrate the use of the proposed inference machinery by reanalyzing a phylogeny of primates and their associated ecological role as seed dispersers. Deep learning inference provides at least the same accuracy as likelihood-based inference while being faster by several orders of magnitude, offering a promising new inference approach for deployment of future models in the field.

KEYWORDS: convolutional neural networks, birth-death models, deep learning, phylogeny representation, diversification, macroevolution.

INTRODUCTION

Phylogenetic approaches for studying species origination and extinction dynamics over deep time rely on the statistical adjustment of stochastic birth-death models (Kendall 1948) to dated phylogenetic trees representing the evolutionary relatedness of species and the dating of their divergence times (Stadler 2013; Morlon 2014; Harmon 2019). An increasing amount of such phylogenetic trees have become available, and has been accompanied by the complexification of diversification models. These models include homogeneous rate models where speciation and extinction rates are identical across lineages at any given time, and range from simple time-constant (Nee et al. 1994) to time-dependent (Morlon et al. 2011; Stadler 2011; May et al. 2016), environment-dependent (Condamine et al. 2013), and diversity-dependent (Etienne et al. 2012) models. Diversification models also include heterogeneous rate models (Alfaro et al. 2009; Morlon et al. 2011; Rabosky 2014; Höhna et al. 2019; Maliet et al. 2019; Barido-Sottani et al. 2020; Laudanno et al. 2020), with the class of State-dependent Speciation and Extinction (SSE) models that links rate heterogeneity to specific characteristics of the species (Maddison et al. 2007; FitzJohn 2010; Goldberg et al. 2011; Fitzjohn 2012; Beaulieu and O'Meara 2016; Herrera-Alsina et al. 2019; Vasconcelos et al. 2022).

The parameters of interest of these models – mainly speciation and extinction rates – are traditionally inferred using likelihood-based techniques – maximum likelihood or Bayesian inference. Maximum likelihood consists of finding the parameters that maximize the relative probability of observing the data (here the phylogeny, or the phylogeny and associated trait data in the case of SSE models); Bayesian inference uses this likelihood along with *a priori* information on the parameters of interest to explore the *posterior* probability distribution of the parameters given the data. Numerous studies have used these inference approaches to estimate speciation and extinction rates over geological times and across the Tree of Life, to investigate the processes modulating these diversification dynamics (Stadler 2011; Etienne et al. 2012; Pyron and Wiens 2013; Höhna 2014; Rolland et al. 2014; Gubry-Rangin et al. 2015; Rabosky et al. 2018; Condamine et al. 2019; Stone and Wolfe 2021). While powerful,

likelihood-based inference techniques are limited by potential intractability issues for complex diversification models, and by computational cost on increasingly large phylogenetic data (Hinchliff et al. 2015). Indeed, complex models do not always have a closed-form solution and/or a model's likelihood cannot always be evaluated in a reasonable amount of time. This renders the application of likelihood-based methods to complex models and species-rich groups (*e.g.*, Coleoptera, Diatoms or SAR11) difficult. As a result, there are several models of diversification in the literature whose behavior has been studied with simulations but that lack an inference machinery (McPeek 2008; Aristide and Morlon 2019; Hagen et al. 2021). Methods based on Expectation Maximization (EM) algorithms (Dempster et al. 1977; Richter et al. 2020), data augmentation (Maliet and Morlon 2022), or composite likelihoods (Lindsay 1988; Varin et al. 2021)) can overcome some of these limitations (Raynal 2019), yet they still rely on likelihood formulae.

Recently, another approach based on universal Probabilistic Programming Languages (PPL (Kudlicka et al. 2020; Ronquist et al. 2021)) has been developed. The latter has been used in phylogenetics in combination with specific automated inference that do not require having closed-form analytical expressions (or numerical solutions) for the likelihood (Kudlicka et al. 2020; Ronquist et al. 2021), nor designing summary statistics. The inference is performed in a Bayesian framework by approximating posterior probability distributions of model parameters by combining prior distributions with distributions over simulated outcomes of the process conditioned on the observed data (Kudlicka et al. 2020; Ronquist et al. 2021). Off-the-shelf general-purpose PPLs such as STAN or WebPPL are user-friendly, however they do not meet the computational requirements of complex diversification models. Custom phylogenetic PPLs addressing these challenges have been proposed, such as Blang (Bouchard-Côté et al. 2022), TreeFlow (Swanepoel et al. 2022), and TreePPL (Senderov 2023); they are promising but still relatively experimental.

An alternative is the use of likelihood-free inference techniques, such as Approximate Bayesian Computation (ABC (Beaumont et al. 2002; Marin et al. 2012; Sisson et al. 2018)). In its most basic form, ABC relies on generating artificial data by simulating the process of interest along a given parameter range and compressing the data by computing summary statistics on these simulations to enable the comparison with summary statistics computed from the observed biological data of interest. This comparison is done by computing a distance and evaluating if this distance is sufficiently small to accept the simulated data using a chosen tolerance threshold (rejection-based approach). ABC has been useful to fit complex models in various fields, including phylogenetic diversification analyses (Bokma 2010; Janzen et al. 2015; Janzen and Etienne 2016). However, there are several limitations of ABC, including its reliance on the choice of statistics that should summarize the information contained in data in a low number of metrics, the choice of the distance metric to compare the observed and simulated data, and the choice of the tolerance threshold (for approaches to evaluate and minimize the influence of these choices on parameter inference, see *e.g.* Sisson et al. 2007; Beaumont et al. 2009; Blum and François 2010; Del Moral et al. 2012; Blum et al. 2013; Prangle 2017).

Deep learning offers an alternative likelihood-free inference technique. Deep learning (Goodfellow et al. 2016) is a subfield of machine learning where highly flexible statistical learning methods based on neural networks (NNs) are used to learn solutions to regression (such as parameter estimation) or classification (such as model selection) problems. The term ‘deep learning’ is conventionally associated with a deep neural network (*i.e.* a NN with at least two hidden layers in addition to the input and output layers). Here we take a more stringent definition (also used in the field) of a deep neural network that takes raw data as input values and extracts patterns from this low-level representation thus creating its own summary statistics or high-level features, without the need of designing those. Estimating model parameters from empirical data by training a deep neural network to learn model parameter values from simulations is increasingly used in several fields including population genetics (*e.g.* to infer parameters related to the strength and type of natural selection or to time-variation

in population sizes, Sheehan and Song 2016; Sanchez et al. 2020; Avecilla et al. 2022) and phylogenetic reconstruction (*e.g.* to estimate evolutionary distances between sequences, Nesterenko et al. 2022). In macroevolution, an early achievement of using machine learning (Bokma 2006) consisted in training an artificial neural network on the axes of a principal component analysis of simulated phylogenetic branching times to infer speciation and extinction rates. A similar framework was then used for the inference of rates of phenotypic evolution (Bokma 2010). While promising, this approach was not developed further by the community. More recently, Sukumaran et al. (2016) and Skeels et al. (2022) trained a machine learning algorithm to learn the classification of biogeographic trait-dependent dispersal and diversification models, from summary statistics although not attempting the regression task for parameter estimation.

A step forward was recently taken by Voznica et al. (2022), who developed a deep learning approach for the statistical inference of birth-death models from phylogenies in the context of pathogen phylodynamics. The authors developed a tree representation, the Compact Bijective Ladderized Vector (CBLV) tree representation, which applies to non-ultrametric trees representing the evolutionary relationship between pathogen sequences sampled at different dates. The CBLV representation proved to be efficient for model selection and the inference of transmission dynamics when combined with Convolutional Neural Networks (CNN) (LeCun et al. 1998), where it yielded accuracy at least comparable to gold-standard Bayesian approaches. Voznica et al. (2022) also combined an extensive set of Summary Statistics with Feed-Forward Neural Networks (FFNN-SS) that yielded similar results. The CBLV representation was extended by Thompson *et al.* (2023) to include geographic information (*i.e.* the location of each strain) and performed well to predict the origin of outbreaks. To our knowledge, comparable attempts to use deep learning to infer diversification dynamics from phylogenies of extant species do not exist.

Here, we aim to establish a proof of concept that deep NNs can detect information contained in phylogenies of extant species (and potentially associated trait data) about speciation and extinction dynamics (state-dependent or not) and estimate diversification parameters faster and as accurately as likelihood methods. We adapt the approach of Voznica et al. (2022) to birth-death diversification models to infer diversification dynamics from phylogenies of extant species. The ~~phylogenetic~~ input data are different from the ones in Voznica et al. (2022). First, the phylogenetic trees are different as we consider here the dated ultrametric trees of extant species (*i.e.* no fossils). Second, we allow for the possibility to include trait data associated to the tips. We begin by adapting the CBLV tree representation from Voznica et al. (2022) to ultrametric phylogenies and to ultrametric phylogenies with tip state data, in the form of 'Complete Diversity-reordered Vector' (CDV). We then assess the performance of deep learning inference in comparison to maximum likelihood estimation (MLE) by using the simple homogeneous time-constant birth-death (BD) model (Nee et al. 1994), for which a closed-form expression of the likelihood exists, and the binary-state speciation and extinction (BiSSE) model, for which the likelihood is approximated by solving Ordinary Differential Equations (Maddison et al. 2007). Finally, we illustrate the approach by applying our trained neural network for BiSSE to an empirical phylogeny of 273 primates (Fabre et al. 2009, [doi:10.5061/dryad.tdz08kq32]) and their associated interaction type (mutualistic or antagonistic) with plants (Gómez and Verdú 2012). In the discussion, we point out potential next steps for deep NNs and diversification analyses.

METHODS

Our main goal is to develop a compact and exhaustive representation of the raw data (ultrametric phylogenies, with potentially associated tip state data) into a matrix (the CDV), and to test the performance of a CNN combined with this representation for parameter inference (thereafter referred to as the CNN-CDV approach). We train the neural networks to learn – from input data simulated under birth-death diversification processes (*e.g.* the CDV representation of ultrametric phylogenies) – the parameters of this diversification process (*i.e.* mainly speciation and extinction rates, which are the output

of the NN). For the time-homogeneous birth-death model (BD), we compare the performance of CNN-CDV to FFNN combined with a series of summary statistics (FFNN-SS), FFNN combined with the CDV representation (FFNN-CDV), and the maximum likelihood estimation (MLE) approach. For the BiSSE model, we compare the performance of CNN-CDV to FFNN-CDV and MLE. Codes used to perform the simulations, encode the phylogenetic data into the CDV representation, train the neural networks, and use the trained networks on simulated or empirical data for parameter inference, are available on GitHub (https://github.com/JakubVoz/deeptimelearning).

TREE REPRESENTATIONS

Full Tree Representation: Complete Diversity-reordered Vector (CDV)

We adapted the Compact Bijective Ladderized Vector (CBLV) representation used in Voznica et al. (2022), originally intended for non-ultrametric trees, to ultrametric trees and to ultrametric trees with information on tip states. The encoding proceeds in the following steps (**Fig. 1**) : 1] internal node reordering, 2] inorder tree traversal (Cormen 2009) and creation of vector representation, 3] completion to the maximum tree-size in simulations, 4] addition of tree height and sampling probability to the vector representation.

Figure 1: Representation of ultrametric trees with tip state data.

Illustration of the encoding algorithm for 'Complete Diversity-ordered Vector' (CDV) on a tree with 5 tips. ***(a)*** *Ultrametric tree with tip state information (here there are two possible states for each tip, either state 1 or state 2, represented in orange and pink respectively).* ***(b)*** *The tree is reordered following a diversity criterion: for each internal node, the sum of the branch lengths of the descending tree is computed and the internal node with higher sum is rotated on the left.* ***(c)*** *We then create a 2-rows matrix filled by visiting the tree according to a tree inorder traversal algorithm, with tips represented on the top row and internal nodes on the bottom row. Tips are assigned their encoded trait state ('1' for state 1 and '2' for state 2) and internal nodes their distance to the root. We add a first column with tree height.* ***(d)*** *Finally, we complete the matrix with zeroes, so that its size is the one of the largest simulated tree and we add a column with the value of the sampling fraction. The final CDV representation is shown in panel d (middle part, in green). To obtain the representation of an ultrametric tree without tip state data, we do not include the first row on tip state information.*

Before encoding, we rescaled the trees to unit average branch length and the rate parameters (*e.g.* diversification rate(s), turnover rate(s) *etc.*) accordingly. The tree rescaling concerned training, validation and testing sets and needs to be performed for empirical datasets as well. In practice, the tree rescaling allows to apply the trained networks on trees of very different ages and diversification rates without the need to simulate from these rates/ages, pending comparable ratios between parameters as those in the training set. The trained neural networks thus apply to a wider set of trees and parameter ranges than the ones used to simulate the training set.

The criterion used in Voznica et al. (2022) for tree reordering was based on the ladderization, where each internal node is rotated so that the branch supporting the most recent tip is on the left. As this cannot apply to ultrametric trees where all tips are sampled at the same time, we used a diversity criterion: for each internal node we compute the sum of the branch lengths of the descending tree (*i.e.* phylogenetic diversity) and the branch with highest sum is shifted to the left. Next, we perform a tree inorder traversal:

the reordered phylogeny is traversed by recursively starting with the left subtree and for each visited internal node, its distance to the root is added to a vector (Cormen 2009). For trees with tip data, we create a second vector with information on the tip data, while visiting tips during the same traversal. For binary tip data (as obtained by simulating BiSSE for example), we use the values 1 and 2 to distinguish the two states. These two vectors are then combined into a matrix. We then add a first column with the value of the tree height. Here, contrary to what was done in Voznica *et al.* (2022), the tip-to-node distances are not explicitly recorded, as for an ultrametric tree this information is redundant with the (already recorded) distance to the root of internal nodes and tree height. We name this representation a Complete Diversity-ordered Vector (CDV). This representation could be easily extended to account for information on non-binary traits, for example using one hot encoding, which consists in encoding a qualitative variable of x states into x rows with 1 representing the presence of state and 0 its absence. Similarly, multiple traits, including quantitative ones, can be encoded by stacking additional rows to the matrix. Furthermore, we can imagine treating missing trait data by assigning a value of -1 in the CDV representation when the information is lacking.

The CDV representation is bijective (under mild assumptions, for example the absence of concomitant branching events), in the sense that we can unambiguously reconstruct any given tree from its representation, and compact: (x+1)**n* entries for a tree with *n* tips (n-1 values for internal nodes and 1 on tree height) and x informational values on tips (x=0 in the absence of tip state information and x=1 for information on a single trait). The created vector (or matrix when tip state information is available) is completed with zeroes to obtain a representation of the same size as the largest phylogeny in the simulations. In order to account for potential missing extant species in the phylogeny, we add a last column with the value of the sampling fraction, computed as the ratio of the number of species represented in the phylogeny divided by the total number of extant species.

In order to assess whether the CDV representation with tip data allows tip state information to be extracted, we compared results obtained with this representation to those obtained with a less informative

representation (coined CDV-less) where, instead of adding a vector with the tip state information, we only add the number of tips in each state at the end of the vector summarizing the information on internal nodes, tree height and sampling fraction.

Summary statistics representation

We used a set of 97 summary statistics (SS) representing trees (without associated trait data as we do not implement the FFNN-SS for the BiSSE model). The summary statistics were mainly based on those published in Saulnier et al. (2017) and in Voznica et al. (2022) (see the original papers for details) and they comprise:

- 8 summary statistics on tree topology
- 25 summary statistics on branch lengths
- 49 summary statistics on the Lineage-Through-Time (LTT) plot
- 14 summary statistics on consecutive internal branches
- 1 summary statistic on number of tips

We modified several statistics so that they apply to ultrametric trees. Instead of minimal and maximal tree height (the time of first and last sampled tips in non-ultrametric trees), we use the crown age. In Saulnier et al. (2017), there are several summary statistics defined on the LTT plot which consider the maximum number of lineages in the phylogeny of sampled taxa, the time of its occurrence and the slopes of the curves before and after this time. In ultrametric trees, the maximum number of lineages in the LTT always appears at present (when we sample the tips) and thus such division of the LTT plot is not possible. Instead, we divide the LTT plot into three equal parts and we measure the slope for each one, together with the ratios between the first and the second slope and between the second and the third slope. The computing time of these statistics grows linearly with tree size. We added the sampling fraction to these 97 measures, thus resulting in a vector of 98 scalars.

Finally, we reduced and centered the SS by subtracting the mean and scaling to unit variance, using the standard scaler from the scikit-learn package (Pedregosa et al. 2011) fitted to the training set.

NEURAL NETWORKS: ARCHITECTURE AND TRAINING

A NN is organized in neural layers that in turn are organized in neurons (or 'units'). In supervised learning, a NN can be trained to minimize the difference between an expected output (or target) and the predicted one; this difference is measured by the 'loss function'. Here we used the mean absolute error as the loss function, which is often used in machine learning regression for its robustness to outliers (we expect that the mean squared error would yield similar accuracy, performing better on extreme values at the cost of non-extreme ones, as it is more susceptible to outliers (Botchkarev 2019)). A NN contains an input layer (by which the numerical values representing the data are passed to the following layer) and an output layer (in our case a vector with estimates of all the parameters of the birth-death model considered), potentially separated by hidden layers (at least two in the case of deep neural networks).

Feed-Forward Neural Networks (FFNNs) are one of the most basic forms of deep NNs. They are fully connected: for each neural layer, all neurons are connected to the neurons of the previous layer. The connections are characterized by trained bias and weights – *i.e.* real values by which individual inputs of a given neuron are multiplied – as well as an activation function by which the summed input (input values multiplied by weights to which a bias value is added) is transformed. FFNNs typically work best on structured data, such as summary statistics, where the same summary statistics are at exactly the same entry in the input vector. While they can also be used on unstructured data (such as images or CDV), the information is then scattered along the input vector, and extracting this information often requires using many hidden layers, with an associated quadratic increase in the number of parameters to train with the size of the input, making FFNNs potentially less efficient.

Convolutional Neural Networks (CNNs) (LeCun et al. 1998; Krizhevsky et al. 2017) contain a convolutional-pooling part and a fully connected one. The convolutional-pooling part consists of convolutional and pooling layers and outputs a vector used as input of the fully connected part. The convolutional-pooling part aims at learning and extracting repeated patterns in the input, that are then combined for prediction in the fully connected part. Convolutional layers transform their input with several convolutional operations, each one specified by a kernel (or 'filter' or 'feature detector') whose parameters are trained. Each kernel transforms subparts or patches of the input by applying the convolutional operation and outputs a single value for each patch. We set the stride (by how much the kernel moves on the input when traversing it) to 1. A convolutional layer is specified by the number of kernels and their size (the size of their input), and outputs a 'feature map' (intermediate representation of transformed input). Pooling layers transform the resulting feature maps into smaller ones by taking the maximum or average of values subsampled in the map within a given window. CNNs typically work well with raw, low-level (vector and matrix) data such as images, videos or time-series recordings, by learning and extracting repeated patterns or 'features' through trained convolution functions and building from them their own high-level features (such as a set of summary statistics). They do not need pre~~specified~~defined feature input such as summary statistics. Each kernel learns and extracts one pattern in the data that can appear anywhere in the representation. The same accuracy on raw, low-level data can also be obtained with FFNNs, but typically at the cost of larger training sets.

The training of a given network consists in iteratively changing the parameters of the NN (*e.g.* bias, weights) in order to minimize the loss function. This is performed by an optimization algorithm for stochastic gradient descent, here the Adam optimizer (Kingma and Ba 2015). The networks were trained on simulated data, the targets being the parameters of the diversification models (here BD and BiSSE).

Several training 'tricks' were developed for efficient and robust training in practice. We used a training set of 990,000 simulations split into subsets called batches, during the training. After measuring the loss on the whole batch, the trained parameters were updated with the optimizer to minimize the loss.

Splitting the training set into batches of simulations enables to update the trained values more robustly (and moving into 'right direction' with respect to the minimal error). We set the batch size to 8,000 simulations, as preliminary analyses showed that smaller batch sizes (of 1,000) slowed down the training and affected the accuracy. When the network parameters were updated on the whole training set (called an 'epoch'), the training started again passing through the whole training set.

To prevent overfitting, we used the dropout technique on fully-connected layers for both FFNNs and CNNs, which consists of shutting down randomly half of the neurons in the network during the training phase (Srivastava et al. 2014). We set the dropout to 0.5, the rule of thumb value (Srivastava et al. 2014). We also used a technique called early stopping (Bengio 2012): at the end of each epoch, the loss is computed on a validation set (here, we used a validation set of 10,000 simulations), and the training is stopped when the loss on the validation set starts to increase. Typically, during the training of our networks, there will be hundreds of epochs before the training is stopped.

The test set then enables one to measure the true accuracy of the network. In our case it consisted of 500 simulations for the BD model and 10,000 simulations for BiSSE.

For the BD model, we used a CNN architecture on the CDV representation (referred to as CNN-CDV), a FFNN architecture on the CDV representation (FFNN-CDV), and a FFNN architecture on the summary statistics representation (FFNN-SS). For the BiSSE model, we used only the CNN-CDV and the FFNN-CDV, as applying the FFNN-SS would require deploying a new set of summary statistics accounting for tip data.

We used network architectures very close to those used in Voznica et al. (2022). The FFNNs consisted of: i) one input layer (with a number of nodes corresponding to the size of the input data), ii) 4 sequential hidden layers organized in a funnel shape with 64-32-16-8 neurons, and iii) 1 output layer with all the parameter estimates (of size 2 for BD and 4 for BiSSE model respectively). During the exploratory phase of the project, we trained multiple FFNN-CDVs each with an output of size one (for each

parameter), rather than a single FFNN-CDV with a multidimensional output, and this yielded similar accuracy (data not shown). The neurons associated to all layers had an exponential linear activation (Clevert et al. 2015). The only difference with Voznica et al. (2022) is thus the activation function of the output layer.

As for the CNNs, the input layer was reshaped, in the case of CNN-CDV and BiSSE, into a matrix with tips and internal nodes separated into distinct rows (and the sampling probability added at the end of each row). As hidden layers, we used a combination of: i) two 1D convolutional layers of 50 kernels each; the kernels had input size 5*1 (for BD and 5*2 for BiSSE) in the first layer, and 10*1 (for BD and 10*2 for BiSSE) in the second ii) max pooling of size 10, iii) another 1D convolutional layer of 80 kernels of size 10, iv) a GlobalPoolingAverage1D layer, and v) a FFNN of funnel shape (64-32-16-8 neurons) with the same architecture and setting as the NN used for FFNN-SS. The only differences with Voznica et al. (2022) are the activation function of the output layer (set here to the exponential linear unit function; the Rectified Linear Unit – ReLU – could be an interesting alternative to force outputs to be positive) and the size of the 1-D kernels in the first layer: we used 5 while they used 3, which we found to perform slightly better.

We implemented the NNs in Python 3.6 using the Tensorflow 1.5.0 (Abadi et al. 2016), Keras 2.2.4 (Chollet 2015) and scikit-learn 0.19.1 (Pedregosa et al. 2011) libraries.

CONFIDENCE INTERVALS

95% confidence intervals (CI) around parameter values can be computed using the approximated parametric bootstrap method suggested in Voznica et al. (2022). The overall idea is to use the distribution of prediction errors computed on simulations, using simulations that are closest to the data; we use here simulations from the training set. More precisely, to compute CIs for a set of parameter values on phylogeny T, we subset the 20% of the simulations from the training set that are closest in terms of tree size, and then the 20% of these which are closest in terms of sampling fraction. Next, for each parameter

value p predicted from T, we identify the 1,000 simulations from the remaining set with closest true parameter values $R_{CI}=\{r_{i=1,1000}\}$, record the corresponding predicted parameter values $P_{CI}=\{p_{i=1,1000}\}$, and measure the estimation errors as $E_{CI}=\{e_i=p_i-r_i\}$. Finally, we extract the 95% CI around p from the distribution $D=\{p+e_i-m(E_{CI})\}$, where $m(E_{CI})$ is the median of errors (individual points in D that have negative values are set to 0).

MACROEVOLUTIONARY MODELS AND SIMULATIONS

We assessed the performance of the NNs with two widely used models, the simple homogeneous, time constant birth-death (BD) model, and the binary-state speciation and extinction (BiSSE) model.

The Birth-Death (BD) Model

The BD model has a closed-form expression of its likelihood, which allows us to compare the performance of the Deep Learning and MLE approach in a "best case" scenario for MLE. Here, when referring to BD, we mean the homogeneous time-constant birth-death-sampling model (Nee et al. 1994; Yang and Rannala 1997; Stadler 2009), as we consider the possibility that some extant species are not represented in the phylogenies. In this model, new species originate with a constant speciation rate λ and go extinct with a constant extinction rate μ, typically expressed in number of events/lineage/Myr. At present each extant species is sampled with probability f (Bernoulli sampling scheme (Stadler 2009)). We assume f to be fixed (it is given as information in the CDV and SS representations and is not treated as a free parameter), in which case λ and μ are identifiable.

We parametrized our simulations with the turnover ($\varepsilon = \mu / \lambda$) and speciation rate (**Table 1**); the parameter values were sampled uniformly at random within parameter boundaries with standard Latin-hypercube sampling (McKay et al. 1979) using the Python PyDOE package. We used a uniform

distribution to prevent overrepresentation of a particular parameter subspace, as the same trained network is intended to be applied to a variety of empirical data, rather than on a single dataset with associated *a priori* parameter values. We performed the simulations with our own simulator, using a Gillespie algorithm (Gillespie 1977). Each simulation started with one lineage and ended when the number of living species reached $\frac{s}{f}$, where s is the number of tips in the sampled phylogeny. We then sampled s species. We thus conditioned the simulations on the number of tips. We trained the NNs to learn λ and ε.

Table 1: Parameterization of the constant-rate birth-death model with incomplete sampling.

For each parameter, we display its symbol and the range of values used in the simulations by sampling from the uniform distribution (indicated with U) in training, validation and testing sets. Parameters indicated in bold are those that are estimated during the inference.

The Binary State Speciation and Extinction (BiSSE) Model

The BiSSE model is the simplest state-dependent birth-death model: species are characterized by a binary state (1 or 2) which can influence their constant speciation (λ_1 and λ_2) and extinction rates (μ_1 and μ_2). Species can also transition anagenetically from one state to the other (with rates q_{12} and q_{21}). As for the BD model, we can add to this diversification process a Bernoulli sampling scheme at present, which allows analyzing trees with missing extant species. We consider here a simple version of BiSSE with symmetrical transition rates $q=q_{12}=q_{21}$ as well as turnover rate and sampling probabilities at present (ε and f, respectively) shared across species irrespective of their state.

The BiSSE model allows us to illustrate the utility, and test the validity of the CDV representation with tip data. The likelihood of this model can be computed by solving Ordinary Differential Equations (ODEs), and recent efforts have been made to provide an efficient maximum-likelihood inference machinery for this model on large phylogenies (Louca and Pennell 2020).

We parameterized our simulations with the speciation rate associated to state 1 (λ_1), the turnover rate (ε) and the ratios of λ_2 and q_{12} relative to λ_1. We sampled these parameters within a biologically realistic parameter space, given the literature on empirically inferred parameters (*e.g.* (Villarreal and Renner 2013; Williams et al. 2014; Gamisch 2016)) (**Table 2**). The parameter subspace was covered with standard Latin-hypercube sampling (McKay et al. 1979) using the Python PyDOE package. The simulations were performed using the R package castor 1.6.6 (Louca et al. 2018) and were conditioned on number of tips. We trained the NNs to learn λ_1, λ_2, q_{12} and ε.

Table 2: Parameterization of the Binary State Speciation and Extinction model with incomplete sampling.

For each parameter, we display its symbol and the range of values used in the simulations by sampling from the uniform distribution (indicated with U) in training, validation and testing sets. Parameters indicated in bold are those that are estimated during inference. λ_2 *is parameterized with respect to* λ_1*, being at 10% to 100% of its value. The transition rates are also parameterized with respect to* λ_1*, being at 1% to 10% of its value. The corresponding ranges of values are indicated in brackets.*

MAXIMUM LIKELIHOOD ESTIMATION

We compared the NN approaches with MLE. For the constant-rate BD, we used a MLE based on the Nelder–Mead optimization algorithm encoded within our custom function fitMLE_bdRho available at https://github.com/sophia-lambert/UDivEvo/tree/master/R. The function encodes a likelihood formula conditioned on the age of the phylogeny (here $t_0 = t_{crown}$) under a Bernoulli sampling scheme, and parametrized to infer the net diversification rate ($r = \lambda - \mu$) and the turnover rate (ε) as it is easier to maximize the likelihood in this reparametrized likelihood landscape. For the BiSSE model, we used the

MLE deployed under the R packages diversitree 0.9-3 (Fitzjohn 2012) and castor 1.6.6 (Louca and Doebeli 2018), both conditioned on the crown age of the phylogeny, under a Bernoulli sampling scheme and parametrized to infer λ_1, λ_2, μ_1, μ_2, and $q=q_{12}=q_{21}$. Diversitree is the traditional package used for fitting BiSSE; castor was developed more recently, and implements a faster algorithm for computing the likelihood of SSE models on large phylogenies (Louca and Doebeli 2018).

PERFORMANCE ASSESSMENT

Accuracy of Parameter Estimation

To assess the accuracy of parameter estimation, we used 500 simulated test trees for the simple BD model and 10 000 for BiSSE. 17 BiSSE simulations for which castor and/or diversitree outputted an error message or no estimated values (15 for castor, 3 for diversitree with one simulation in common) were excluded from these analyses, resulting in 9,983 test trees.

To avoid over-penalizing the MLE approaches that, contrary to the NNs, do not have constrained parameter ranges, we added similar constraints to the MLE estimates. Indeed, to the exception of λ for constant-rate BD and λ_1 for BiSSE for which NNs can predict values outside of the parameter values initially covered by the simulations due to tree rescaling, the other parameter estimates (ε, λ_2 and $q=q_{12}=q_{21}$) are constrained by the parameter range used for ε, $r_{\lambda 2}$ and r_q in our simulations. We imposed similar constraints to the MLE estimates. For example, if the MLE for ε is above 1, we set it to 1, and if the MLE for λ_2 is lower than $0.1*\lambda_1$, we set it to $0.1*\lambda_1$ (0.1 is the minimum value for $r_{\lambda 2}$ used in our simulations). These constraints 'help' the MLE: by running the MLE without them (as is done in empirical applications), we would obtain larger deviations from the simulated parameters, hence larger error in comparison to the NN approaches. Constraining the MLE provides a fairer (and more conservative) comparison.

To evaluate deviations between the true (~~simulated,~~ or 'target') parameter values and the predicted, we used four different measures:

- *Error* $E_i = \text{predicted}_i - \text{target}_i$
- *Mean absolute error* $MAE = 1/n * \sum_i^n \text{abs}(\text{predicted}_i - \text{target}_i)$
- *Mean relative absolute error* $MRE = 1/n * \sum_i^n \text{abs}(\text{predicted}_i - \text{target}_i)/\text{target}_i$
- *Bias* $B = 1/n * \sum_i^n (\text{predicted}_i - \text{target}_i)$

We also reported the Pearson correlation coefficient between simulated and predicted values, computed with the R package 'stats' ('cor.test' function) version '4.2.1'.

We assessed the influence of tree size on parameter estimation accuracy.

Finally, to evaluate whether deviations from target values obtained with different inference techniques are the same or different, we evaluated the Pearson correlations between the estimation error (E) obtained with each of our inference procedures.

Time Efficiency

We compared the average time of estimation between CNN-CDV and MLE for BiSSE. For the CNN-CDV approach, we reported the average CPU time of encoding a tree (averaged over 1,000,000 trees). The estimation on itself is negligible with respect to the time of encoding. For MLE BiSSE estimation with the castor and diversitree packages, we reported the average CPU time (average over 10,000 test trees, out of which 18 resulted in an error).

EMPIRICAL ILLUSTRATION

Primates can have an antagonistic interaction with plants through herbivory, or a mutualistic one through frugivory and seed dispersal (Gómez and Verdú 2012). To illustrate the application of our deep learning approach, we reanalyzed the dataset of Gómez and Verdú (2012), that categorizes primate species

according to the nature of their interaction with plants (state 1 for a mutualistic interaction, and 2 for an antagonistic one), using the primates phylogeny of Fabre et al. (2009). We started by pruning taxa without information on interaction type (13/273) and rescaled the phylogeny as previously described. Next, we performed some sanity checks to verify that the empirical data fell within the space covered by our BiSSE simulations; if it does not, this means that the model and/or the range of parameters used in the simulations is not well adapted to the empirical data, in which case application of the trained neural network to the data might output meaningless results. For these analyses, we used the set of phylogenies that we simulated under the BiSSE model to produce our test set. First, we checked that each of the summary statistics values calculated on the empirical data fell within the range spanned by the summary statistics values calculated on the simulated phylogenies. Then, we performed a principal component analysis (R package 'FactoMineR', function 'PCA') on the set of summary statistics described above (see Methods) with the addition of four simple summary statistics (101 summary statistics in total) adapted to the inclusion of tip data: the number of tips in each state (1 or 2) and the phylogenetic diversity of each state (R package 'picante', function 'pd'). Finally, we transformed the data into its CDV representation and fed it to the trained CNN-CDV network on the BiSSE model. We used a sampling fraction of 0.68, computed using a global diversity of 381 stable species complexes for primates, following Gómez and Verdú (2012). We computed 95% confidence intervals around CNN-CDV parameter estimates using the approximated parametric bootstrap method described above. We compared the results with those obtained using the classical MLE approach, computing the 95% confidence intervals around MLE estimates with an adaptive sampling of the likelihood surface ('hisse' and 'SupportRegionHiSSE' functions of the R package 'hisse', respectively). We used the 'hisse' package for more direct comparison, as its parametrization (using the turnover rate rather than the extinction rate) is the one we used in our machine learning approach. We ran the inference without hidden states, equal turnover rates between states, and symmetrical state transition rates.

RESULTS

PERFORMANCE ASSESSMENT

The comparison of parameter estimates obtained using deep NNs versus MLE for the BD model shows that CNN-CDV and FFNN-SS are as accurate as MLE, while FFNN-CDV has a lower accuracy, in terms of the error E and mean absolute and relative errors, for the speciation, extinction, net diversification and turnover rates (**Fig. 2**, **Table S1**). The likelihood of BD model has an exact analytical solution. This implies that MLE, together with CNN-CDV and FFNN-SS, are as accurate as one can be. The good performance of FFNN-SS might be explained by the representation of the lineage-through-time plot in the summary statistics, that contains all information available in the tree for homogeneous BD models (Nee et al. 1994). The lower performance of FFNN-CDV was expected given that FFNN is less adapted to unstructured data. The neural networks seem to avoid cases of high negative error on the turnover rate compared to MLE: while this error can reach down to -0.6 with MLE, it never falls behind -0.35 with CNN-CDV and FFNN-SS (**Fig. 2d**). This is probably due to the fact that contrary to MLE, CNN-CDV and FFNN-SS rarely return estimates close to 0 for the turnover rate (**Fig. S1d**). In term of bias, the MLE and deep NNs exhibit similar trends with a negligible positive bias for speciation, extinction, turnover rate and a negligible negative bias for the net diversification rate (**Fig. 2**, **Table S1**).

Figure 2: Comparison of estimation accuracy between pretrained CNN-CDV, FFNN-SS, FFNN-CDV and MLE for the BD model.

Swarm plots representing the distribution of estimation errors (E) across 500 simulations. Each dot represents the error of a single simulation. Estimated parameters were obtained with Convolutional Neural Networks- Complete Diversity-reordered Vector (CNN-CDV, in purple), Feed-Forward Neural Networks- Summary Statistics (FFNN-SS, in orange), Maximum Likelihood Estimation (MLE, in green) and Feed-Forward Neural Networks- Complete Diversity-reordered Vector (FFNN-CDV, in blue) for (a) the speciation, (b) the extinction, (c) the net diversification and (d) the turnover rate. The mean absolute

error (MAE) is displayed under each swarm plot and the bias is represented as a black dot on swarm plots for each model and parameter.

The comparison of parameter estimates obtained using deep learning versus MLE for the BiSSE model confirms that CNN-CDV is at least as accurate as MLE. The only exception is for λ_2 estimates, where MLE implemented in diversitree is slightly more accurate than CNN-CDV in terms of mean absolute error. CNN-CDV is more accurate than the fast MLE algorithm implemented in castor for all parameters (**Fig. 3, Table S2**). The accuracy of the MLE estimation implemented in castor is similar to that of the least reliable FFNN-CDV neural network; it performs slightly better for some parameters (μ_1 and q_{12}) but slightly worse for others (μ_2). The CNN trained on a CDV without the individual tip information (CNN-CDV-less) is much less accurate, indicating that the information on individual tips states is well presented in the CDV representation, and extracted by the CNN (**Table S2**). In term of bias, the MLE and deep NNs exhibit different trends. CNN-CDV displays a negligible negative bias for λ_2, μ_1, μ_2 and q_{12} while MLE displays a negligible positive bias for all parameters (**Fig. 3**, **Table S2**). This negligible bias is not always consistent across the parameter space. In particular, the slight underestimation of the CNN-CDV seems to be pulled by an underestimation of high rates (**Fig. S2**), a part of the parameter space that is less represented in the training and testing sets for some parameters. This is due to the fact some parameters are parameterized with respect to others (**Table 1**, e.g. λ_2 is parameterized with respect to λ_1, being at 10% to 100% of its value, thus maximal possible values reached by λ_2 are relatively rare).

Figure 3: Comparison of estimation accuracy between pretrained CNN-CDV, FFNN-CDV, and MLE obtained with two different inference software for the BiSSE model.

Swarm plots representing the distribution of estimation errors (E) across 9,983 simulations. Estimated parameters were obtained with CNN-CDV (in purple), castor (in dark green), diversitree (in light green) and FFNN-CDV (in blue) for a) the speciation rates 1 and b) 2, c) the extinction rates 1 and d) 2 and e) the transition rates (q_{12}=q_{21}, in our setting). The simulations for which castor or diversitree outputted an

error message (15/10,000 for castor and 3/10,000 for diversitree) were excluded from the comparison. The MAE is displayed under each swarm plot and the bias is represented as a black dot on swarm plots for each model and parameter. For visualization purposes, the upper outliers of the swarm plot are not displayed at their extreme values but instead are put at the boundaries of the chart.

For both the BD and BiSSE model, the accuracy of CNN-CDV increases as the tree size increases (**Fig. 4**). Similar to MLE, a tree size of at least 300 sampled extant species is required for a median relative absolute error of 6% for the speciation rate and 18% for the extinction rate in the case of the BD model (**Fig. S3**). In the case of the BiSSE model, a tree size of at least 380 is required for a median relative absolute error of less than 14% on the speciation rates, 25% on the extinction rates, and 13% on the transition rate.

Figure 4: Effect of tree size on the absolute error of parameter estimates when using CNN-CDV under the BD and BiSSE models.

For each model a) constant-rate birth-death (BD), and b) Binary-State Speciation and Extinction (BiSSE), we display the regression on absolute error for each parameter as a function of tree size for 500 test trees (for BiSSE 500 values are shown instead of 10,000 for visualization purposes). The area around the solid line delimited by the dotted lines represent the 95% confidence interval around the regression.

Under the BD model, we found a strong correlation between the errors (E) obtained for the two best performing deep NNs and MLE (**Fig. 5**). The errors obtained with FFNN-CDV, on the other hand, exhibit a low correlation with errors obtained with all the other methods. The correlations of errors obtained with the best performing deep NN (CNN-CDV) and MLE are also weaker for the BiSSE model (**Fig. 6**), in particular for λ_2 and μ_2.

Figure 5: Correlation between estimation errors obtained for the BD model with CNN-CDV, FFNN-SS, FFNN-CDV and MLE. *Correlation matrix representing the Pearson correlation coefficient of estimation errors (E) across the 500 test set simulations of the BD model for (a) the speciation rate and (b) the extinction rate. The larger and the darker the circle, the stronger the correlation.*

Figure 6: Correlation between estimation errors obtained for the BiSSE model with CNN-CDV, FFNN-CDV, MLE computed with *diversitree*, and MLE computed with *castor*.

Correlation matrix representing the Pearson correlation coefficient of estimation errors (E) across the 9,983 test set simulations of the BiSSE model for a) & b) the speciation rates associated with states 1 and 2, c) & d) the extinction rates associated with states 1 and 2 and e) the transition rates (q_{12}=q_{21}, in our setting). The larger and the darker the circle, the stronger the correlation.

The parameter estimation for the 10,000 BiSSE simulations took 629.5 CPU hours with diversitree, 55.6 CPU hours with castor and 0.4 CPU hours with CNN-CDV, which consisted in encoding the test set into CDV. Once the CNN-CDV is trained, the estimation is thus around 140 times faster than castor and over 1500 times faster than diversitree. Training the network entailed first simulating the training set (1 million trees, 40 CPU hours). The training in itself then took 8 ~~CPU~~ hours with Nvidia Titan X GPUs. Contrary to MLE for which a large number of CPU hours are required for each new empirical analysis, with deep learning the empirical analyses are very fast once the network has been trained. The same pre-trained networks should be applicable to a large variety of empirical trees, thanks to tree rescaling which enables applications to clades of very different ages and speciation rates, and this is why the computational time required to run the simulations of training and validation datasets was not accounted for in the comparison.

EMPIRICAL ILLUSTRATION

The primate phylogeny with associated character state (mutualistic or antagonistic interaction with plants) passed the sanity checks on model adequacy. Indeed, all the summary statistics computed on the empirical data fell within the range spanned by the simulations. Likewise, our PCA analysis on these summary statistics showed the empirical data nested in the simulations space, when considering both PC1 and PC2 (explaining together 69% of the variance of our data, **Fig. 7a**) and PC3 and PC4 (that represent an additional 12 points of explained variance, **Fig. 7b**). The CNN-CDV analyses estimated a speciation rate λ_{mut} of 0.295 [95% CI 0.234-0.375] for primate species with a mutualistic interaction with plants, and λ_{anta} of 0.093 [95% CI 0.046-0.144] for those with an antagonistic interaction. The turnover rate ε was estimated at 0.234 [95% CI 0.100-0.446], and the transition rates q at 0.0089 [95% CI 0.0049-0.0137]. With MLE, we obtained consistent results for λ_{mut} (0.395 [95% CI 0.342-0.452]) and q (0.013 [95% CI 0.0085- 0.018]), with overlapping CIs. This was not the case for λ_{anta} for which estimates obtained with MLE were higher than those obtained with CNN-CDV (0.210 [95% CI 0.167-0.266]), although they remained lower than estimates of λ_{mut}, therefore not changing the qualitative results. Estimates obtained with MLE were also higher than those obtained with CNN-CDV for the turnover rate ε (0.778 [95% CI 0.690- 0.858]). This simple analysis suggests that mutualistic interactions can favor diversification in primates, as found by (Gómez and Verdú 2012), although we do not interpret this result further here, as models with hidden traits should be used to reach more convincing biological conclusions (Beaulieu and O'Meara 2016; Scott 2018). The goal of this empirical analysis is simply to illustrate how the method and trained networks can be used on empirical data.

Figure 7: The primate data falls within the space of our BiSSE simulations.

Coordinates of the empirical data (large dot, in pink) and of the test set simulations (small triangles, turquoise) on the a) PC1 and PC2 axes and b) PC3 and PC4 axes of a principal component analysis performed on 101 summary statistics.

DISCUSSION

We developed, tested and illustrated the use of a deep learning based inference approach for phylogenetic diversification analyses, including the case of trait-dependent diversification. We found that both convolutional neural networks combined with a compact representation of the phylogenetic data into a matrix (the CDV) and feed-forward neural networks combined with summary statistics can reach levels of parameter estimation accuracy comparable to those obtained with the well-established maximum likelihood approach, while being faster by several orders of magnitude.

To demonstrate the potential of deep learning for phylogenetic diversification analyses, we worked with two simple diversification models on which MLE estimates can easily be obtained for comparison (the BD and BiSSE models). We also worked with phylogenies of relatively moderate size (200 to 500 extant species sampled). The real value of deep learning will be to allow rapid inferences for more complex models for which likelihoods are not tractable, and/or for large phylogenies of several thousands of extant species for which computing likelihoods can take a long time. Our analyses on simple models and relatively moderate size demonstrate that it is worth putting effort and computation power into simulating more complex models, generating larger trees, and training neural networks on such simulations. Given our results, we can expect efficient simulators combined with the CDV representation and CNN to provide an accurate likelihood-free estimation method applicable to very large phylogenies. The CDV representation is not model-specific and can easily be enriched with information on both internal nodes and tips. It could be used for example to represent information on species multidimensional traits, geographic distributions, abundances, and genetic diversity. Combined with efficient simulation

models for the evolution of biodiversity (e.g. Hagen et al. 2021), the CNN-CDV deep learning inference approach could help adjusting biologically realistic biodiversity models to multifaceted data for a better understanding of how present-day biodiversity was generated, maintained, and distributed geographically.

When evaluating the correlations of errors across different inference techniques, we showed high correlations for all parameters among CNN-CDV, FFNN-SS and MLE (and lower correlations with FFNN-CDV) under the rate-homogeneous BD model, and correlations that depended on the parameter considered under the rate-heterogeneous BiSSE model. This suggests that the CNN-CDV, FFNN-SS and MLE might capture the same information about phylogenetic branching times (the only informative data in the case of homogeneous birth-death models), but not topology (which becomes informative in the case of heterogeneous models). Under the simple BD model, estimates are highly accurate, and deviations between true and estimated parameter values might be mainly driven by the stochasticity of the process, resulting in a high correlation among errors. The less accurate NN approach (FFNN-CDV) exhibits a lower error correlation in line with its lower accuracy, supporting the hypothesis that the FFNN is not as efficient to extract the appropriate information directly from CDV input. Under the BiSSE model, correlations of errors are especially low for λ_2 and μ_2, which is also reflected in our results on the primate data, where estimates obtained with CNN-CDV and MLE are quite different. These parameters influence phylogenetic imbalance, supporting our interpretation that CNN-CDV may extract different topological features than MLE. The low correlations of errors could also in part be due to the slight difference of performances between the inference techniques for this more complex model.

Besides parameter estimation, diversification models are widely used for model comparison, in order to test alternative hypotheses about how diversification proceeds. This problem can be treated with deep learning as a classification problem, with neural networks trained to predict a class (a model) for simulations obtained with different models, covering large parameter subspaces and combined with *a priori* and *a posteriori* safety checks. This approach has been developed for pathogen phylodynamics in Voznica et al. (2022) and shown to perform as good as the gold standard approaches. In the case of

diversification analyses, an example of an interesting hypothesis to test is whether species in different states have different speciation rates, which can be done by model comparison using the BiSSE model: the approach would be to train a neural network to distinguish data simulated under a model where lineages in different states have different speciation rates from data simulated under a model where they have the same rates. This could be done by using the same CNN-CDV settings we used in the paper for parameter estimation under BiSSE, by replacing the MAE loss function by a categorical function such as the cross entropy function and the exponential linear unit activation function of the output layer by a function such as the softmax (Voznica et al. 2022). Given that speciation rates can be estimated with good accuracy with deep learning, we expect that neural networks will also be able to efficiently distinguish these models given enough differences in speciation rates between states.

We have explored the applicability of CNNs and FFNNs for phylogenetic diversification inference. Other classes of neural networks may also bring accurate solutions for parameter estimation and model selection when combined with simulated datasets. Noticeably, the Graph Neural Networks or Graph Convolutional Networks are neural networks designed for graph data that could be particularly well suited for analyzing phylogenies, encoded as directed graphs. Applications of the GNNs have been developed in the last couple of years across different fields, for instance in protein interaction prediction, drug design or social networks analyses (Zhou et al. 2020). More work is required to assess which network architecture performs best for phylogenetic diversification analyses, which has started to be explored by Lajaaiti et al. (2023). Other alternative tunings of NN hyperparameters can be performed, such as the choice of the loss function. We have chosen here the MAE which favors estimation accuracy for moderate rather than extreme parameter values. This resulted in a slight overall better performance of NNs over MLE, but a tendency for underestimation of values that are less represented in the training sets, such as high values for parameters that are not uniformly distributed in our setting (λ_2, μ_1, μ_2 and q_{12} in the case of BiSSE). Using the MSE instead of the MAE could increase estimation accuracy for extreme

values (Botchkarev 2019), which could be pertinent in the case of strongly uneven representativeness of parameter subspace in the training set.

As the ground truth is unknown, the neural networks are trained on simulations. This raises questions on how robust such approaches are to model misspecification, even though some studies suggest that machine learning approaches might be more robust to model misspecification than other inference approaches (Liang and Jordan 2008; Lee et al. 2010). Recently in the context of pathogen phylogeography, Thompson et al. (2023) showed that CNN combined with an (extended) CBLV representation are as robust as likelihood-based approaches when it comes to model misspecification. We illustrated how some sanity checks can be performed to verify that the empirical data falls within the space of simulated data. If it does not, outputs of the neural networks should not be trusted. As in Voznica et al. (2022), we used summary statistics for these sanity checks. Another approach circumventing the use of summary statistics would consist in using autoencoders (Hinton and Salakhutdinov 2006), often deployed for anomaly detection (Chalapathy and Chawla 2019). The autoencoders are a family of NNs, which are trained to output their input while enforcing dimensionality reduction within their neural layers. The induced reconstruction error, *i.e.* the difference between the input and output, can then be used to check if an empirical data is well represented by simulations. For example, autoencoders could be trained on the CDV representation of phylogenetic data simulated under diversification models, and then applied to empirical phylogenetic data. If the reconstruction error is larger for the empirical data than for the simulations, this indicates departures from the simulated model.

While we showed that the combination of CNN and CDV yielded a fairly accurate inference method, we did not focus on the explainability of our networks, *i.e.* understanding what patterns are automatically extracted from a raw representation or phylogenies. During an exploratory phase of the project, we assessed permutation feature performance (Breiman 2001): we created modified test sets from the original one while permuting the values of one individual column at a time and looked at the impact on the accuracy of prediction on such a modified test set. This approach showed that the perturbation of

the tree height and the sampling fraction impacted the accuracy the most, while a perturbation of the remaining part of the representation (internal nodes and trait information) had minor impact (data not shown). This was in line with our expectations: the information is diluted all along the phylogeny representation. Further investigation of individual kernels with established tools (Selvaraju et al. 2020; Li et al. 2022) may shed light into individual patterns that are important for the inference.

We have considered the task of fitting birth-death diversification models to a fixed phylogeny, assumed to be known. While this is a current practice in the field, a better way to account for phylogenetic uncertainty consists in performing full phylogenetic inference, where the phylogenetic tree is inferred from molecular sequence alignments jointly with the parameters of the diversification process (Bouckaert et al. 2014, 2019). Machine learning is already used to data mine molecular sequence alignments (Yang et al. 2020), and recent progress has been made for phylogenetic reconstruction as well (Suvorov et al. 2020; Zou et al. 2020; Nesterenko et al. 2022; Solis-Lemus et al. 2022). Ultimately, these recent advances could be in the long term combined to train CNNs directly on sequences simulated from a joint process of speciation, extinction, and sequence evolution.

Deep learning is gaining popularity in biology, including in ecology and evolution (Borowiec et al. 2021). It has been used as a likelihood-free approach to fit population genetic models (Flagel et al. 2019) to sequence data, and more recently to fit epidemiological models to pathogen phylogenies (Voznica et al. 2022, Thompson et al. 2023). We have shown that it can also perform well as a likelihood-free approach for fitting diversification models to phylogenies of extant species. More work is needed to establish which data representation and network architecture perform best, to perform statistical inference directly on sequence alignments rather than on fixed phylogenies, and to efficiently train networks for more complex models. We hope that our paper will stimulate research in this direction. Ultimately, this should foster the development of new diversification models that are not limited (or whose design is not biased) by our ability to compute likelihoods.

FUNDING

SL was supported by PSL IRIS Science des données, données de la science and the Fondation pour la Recherche Médicale (FDT202106013269). JV was supported by the Ecole Normale Supérieure Paris-Saclay and by ED Frontières de l'Innovation en Recherche et Education, Programme Bettencourt. HM acknowledges funding from ERC-CoG grant PANDA.

ACKNOWLEDGEMENTS

We thank Viktor Senderov for double-checking our understanding of Probabilistic Programming Languages. We also thank Sally Otto, Wouter van der Bijl and their colleagues for their feedbacks on the manuscript. We thank Quang Tru Huynh for administrating the GPU farm at Institut Pasteur and the INCEPTION program (Investissement d'Avenir grant ANR16-CONV-0005) that financed the GPU farm.

DATA AVAILABILITY

The empirical data and Supplementary Material associated to the paper are available from the Dryad Digital Repository: https://datadryad.org/stash/share/Bv2VGCG6wh5vVTVIFJf6uYBVZj5Dap_w-CxOkLMgOBw (Temporary link) [doi:10.5061/dryad.tdz08kq32].

All codes are available on GitHub, at https://github.com/JakubVoz/deeptimelearning.

REFERENCES

Abadi M., Agarwal A., Barham P., Brevdo E., Chen Z., Citro C., Corrado G.S., Davis A., Dean J., Devin M., Ghemawat S., Goodfellow I., Harp A., Irving G., Isard M., Jia Y., Jozefowicz R., Kaiser L., Kudlur M., Levenberg J., Mane D., Monga R., Moore S., Murray D., Olah C., Schuster M., Shlens J., Steiner B., Sutskever I., Talwar K., Tucker P., Vanhoucke V., Vasudevan V., Viegas F., Vinyals O., Warden P., Wattenberg M., Wicke M., Yu Y., Zheng X. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. arxiv.

Alfaro M.E., Santini F., Brock C., Alamillo H., Dornburg A., Rabosky D.L., Carnevale G., Harmon L.J. 2009. Nine exceptional radiations plus high turnover explain species diversity in jawed vertebrates. Proceedings of the National Academy of Sciences. 106:13410–13414.

Aristide L., Morlon H. 2019. Understanding the effect of competition during evolutionary radiations: an integrated model of phenotypic and species diversification. Ecology Letters. 22:2006–2017.

Avecilla G., Chuong J.N., Li F., Sherlock G., Gresham D., Ram Y. 2022. Neural networks enable efficient and accurate simulation-based inference of evolutionary parameters from adaptation dynamics. PLoS Biol. 20:e3001633.

Barido-Sottani J., Vaughan T.G., Stadler T. 2020. A Multitype Birth–Death Model for Bayesian Inference of Lineage-Specific Birth and Death Rates. Systematic Biology. 69:973–986.

Beaulieu J.M., O'Meara B.C. 2016. Detecting Hidden Diversification Shifts in Models of Trait-Dependent Speciation and Extinction. Syst Biol. 65:583–601.

Beaumont M.A., Cornuet J.-M., Marin J.-M., Robert C.P. 2009. Adaptive approximate Bayesian computation. Biometrika. 96:983–990.

Beaumont M.A., Zhang W., Balding D.J. 2002. Approximate Bayesian Computation in Population Genetics. Genetics. 162:2025–2035.

Bengio Y. 2012. Practical Recommendations for Gradient-Based Training of Deep Architectures. In: Montavon G., Orr G.B., Müller K.-R., editors. Neural Networks: Tricks of the Trade: Second Edition. Berlin, Heidelberg: Springer. p. 437–478.

Blum M.G.B., François O. 2010. Non-linear regression models for Approximate Bayesian Computation. Stat Comput. 20:63–73.

Blum M.G.B., Nunes M.A., Prangle D., Sisson S.A. 2013. A Comparative Review of Dimension Reduction Methods in Approximate Bayesian Computation. Statistical Science. 28:189–208.

Bokma F. 2006. Artificial neural networks can learn to estimate extinction rates from molecular phylogenies. Journal of Theoretical Biology. 243:449–454.

Bokma F. 2010. Time, Species, and Separating Their Effects on Trait Variance in Clades. Systematic Biology. 59:602–607.

Borowiec M.L., Dikow R.B., Frandsen P.B., McKeeken A., Valentini G., White A.E. 2021. Deep learning as a tool for ecology and evolution. Methods in Ecology and Evolution. 13:1640–1660.

Botchkarev A. 2019. A New Typology Design of Performance Metrics to Measure Errors in Machine Learning Regression Algorithms. IJIKM. 14:045–076.

Bouchard-Côté A., Chern K., Cubranic D., Hosseini S., Hume J., Lepur M., Ouyang Z., Sgarbi G. 2022. Blang: Bayesian Declarative Modeling of General Data Structures and Inference via Algorithms Based on Distribution Continua. Journal of Statistical Software. 103:1–98.

Bouckaert R., Heled J., Kühnert D., Vaughan T., Wu C.-H., Xie D., Suchard M.A., Rambaut A., Drummond A.J. 2014. BEAST 2: A Software Platform for Bayesian Evolutionary Analysis. PLoS Comput Biol. 10:e1003537.

Bouckaert R., Vaughan T.G., Barido-Sottani J., Duchêne S., Fourment M., Gavryushkina A., Heled J., Jones G., Kühnert D., De Maio N., Matschiner M., Mendes F.K., Müller N.F., Ogilvie H.A., Du Plessis

L., Popinga A., Rambaut A., Rasmussen D., Siveroni I., Suchard M.A., Wu C.H., Xie D., Zhang C., Stadler T., Drummond A.J. 2019. BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. PLoS Computational Biology. 15:e1006650–e1006650.

Breiman L. 2001. Random Forests. Machine Learning. 45:5–32.

Chalapathy R., Chawla S. 2019. Deep Learning for Anomaly Detection: A Survey. .

Chollet F.K. 2015. Keras: the Python deep learning API. Available from https://keras.io/.

Clevert D.-A., Unterthiner T., Hochreiter S. 2015. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). 4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings.

Condamine F.L., Rolland J., Morlon H. 2013. Macroevolutionary perspectives to environmental change. Ecology Letters. 16:72–85.

Condamine F.L., Rolland J., Morlon H. 2019. Assessing the causes of diversification slowdowns: temperature-dependent and diversity-dependent models receive equivalent support. Ecology Letters. 22:1900–1912.

Cormen T.H. 2009. Introduction to algorithms. Cambridge, Mass: MIT Press.

Del Moral P., Doucet A., Jasra A. 2012. An adaptive sequential Monte Carlo method for approximate Bayesian computation. Stat Comput. 22:1009–1020.

Dempster A.P., Laird N.M., Rubin D.B. 1977. Maximum Likelihood from Incomplete Data Via the EM Algorithm - Dempster - 1977 - Journal of the Royal Statistical Society: Series B (Methodological) - Wiley Online Library. Available from https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.2517-6161.1977.tb01600.x.

Etienne R.S., Haegeman B., Stadler T., Aze T., Pearson P.N., Purvis A., Phillimore A.B. 2012. Diversity-

dependence brings molecular phylogenies closer to agreement with the fossil record. Proc. R. Soc. B. 279:1300–1309.

Fabre P.-. H., Rodrigues A., Douzery E.J.P. 2009. Patterns of macroevolution among Primates inferred from a supermatrix of mitochondrial and nuclear DNA. Molecular Phylogenetics and Evolution. 53:808–825.

FitzJohn R.G. 2010. Quantitative Traits and Diversification. Systematic Biology. 59:619–633.

Fitzjohn R.G. 2012. Diversitree: Comparative phylogenetic analyses of diversification in R. Methods in Ecology and Evolution. 3:1084–1092.

Flagel L., Brandvain Y., Schrider D.R. 2019. The unreasonable effectiveness of convolutional neural networks in population genetic inference. Molecular Biology and Evolution. 36:220–238.

Gamisch A. 2016. Notes on the Statistical Power of the Binary State Speciation and Extinction (BiSSE) Model. Evolutionary Bioinformatics. 12:EBO.S39732.

Gillespie D.T. 1977. Exact stochastic simulation of coupled chemical reactions. Journal of Physical Chemistry. 81:2340–2361.

Goldberg E.E., Lancaster L.T., Ree R.H. 2011. Phylogenetic Inference of Reciprocal Effects between Geographic Range Evolution and Diversification. Systematic Biology. 60:451–465.

Gómez J.M., Verdú M. 2012. Mutualism with Plants Drives Primate Diversification. Systematic Biology. 61:567–577.

Goodfellow I., Bengio Y., Courville A. 2016. Deep Learning. .

Gubry-Rangin C., Kratsch C., Williams T.A., McHardy A.C., Embley T.M., Prosser J.I., Macqueen D.J. 2015. Coupling of diversification and pH adaptation during the evolution of terrestrial Thaumarchaeota. Proc Natl Acad Sci USA. 112:9370–9375.

Hagen O., Flück B., Fopp F., Cabral J.S., Hartig F., Pontarp M., Rangel T.F., Pellissier L. 2021. gen3sis: the general engine for eco-evolutionary simulations on the origins of biodiversity. .

Harmon L.J. 2019. Phylogenetic Comparative Methods - Learning from trees. CC-BY-4.0 license: .

Herrera-Alsina L., van Els P., Etienne R.S. 2019. Detecting the Dependence of Diversification on Multiple Traits from Phylogenetic Trees and Trait Data. Systematic Biology. 68:317–328.

Hinchliff C.E., Smith S.A., Allman J.F., Burleigh J.G., Chaudhary R., Coghill L.M., Crandall K.A., Deng J., Drew B.T., Gazis R., Gude K., Hibbett D.S., Katz L.A., Laughinghouse H.D., McTavish E.J., Midford P.E., Owen C.L., Ree R.H., Rees J.A., Soltis D.E., Williams T., Cranston K.A. 2015. Synthesis of phylogeny and taxonomy into a comprehensive tree of life. Proc Natl Acad Sci USA. 112:12764–12769.

Hinton G.E., Salakhutdinov R.R. 2006. Reducing the Dimensionality of Data with Neural Networks. Science.

Höhna S. 2014. Likelihood Inference of Non-Constant Diversification Rates with Incomplete Taxon Sampling. PLoS ONE. 9:e84184.

Höhna S., Freyman W.A., Nolen Z., Huelsenbeck J.P., May M.R., Moore B.R. 2019. A Bayesian Approach for Estimating Branch-Specific Speciation and Extinction Rates. .

Janzen T., Etienne R.S. 2016. Inferring the role of habitat dynamics in driving diversification: evidence for a species pump in Lake Tanganyika cichlids. .

Janzen T., Höhna S., Etienne R.S. 2015. Approximate Bayesian Computation of diversification rates from molecular phylogenies: introducing a new efficient summary statistic, the nLTT. Methods in Ecology and Evolution. 6:566–575.

Kendall D.G. 1948. On the generalized “birth-and-death” process. Annals of Mathematical Statistics. 19:1–15.

Kingma D.P., Ba J.L. 2015. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings.

Krizhevsky A., Sutskever I., Hinton G.E. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM. 60:84–90.

Kudlicka J., Murray L.M., Ronquist F., Schon T.B. 2020. Probabilistic programming for birth-death models of evolution using an alive particle filter with delayed sampling. .

Lajaaiti I., Lambert S., Voznica J., Morlon H., Hartig F. 2023. A Comparison of Deep Learning Architectures for Inferring Parameters of Diversification Models from Extant Phylogenies. .

Laudanno G., Haegeman B., Rabosky D.L., Etienne R.S. 2020. Detecting Lineage-Specific Shifts in Diversification: A Proper Likelihood Approach. Systematic Biology.:syaa048.

LeCun Y., Bottou L., Bengio Y., Haffner P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE. 86:2278–2323.

Lee B.K., Lessler J., Stuart E.A. 2010. Improving propensity score weighting using machine learning. Statistics in Medicine. 29:337–346.

Li X., Xiong H., Li X., Wu X., Zhang X., Liu J., Bian J., Dou D. 2022. Interpretable deep learning: interpretation, interpretability, trustworthiness, and beyond. Knowl Inf Syst. 64:3197–3234.

Liang P., Jordan M.I. 2008. An asymptotic analysis of generative, discriminative, and pseudolikelihood estimators. Proceedings of the 25th International Conference on Machine Learning.:584–591.

Lindsay B.G. 1988. Composite Likelihood Methods. Contemporary Mathematics. 80:221–239.

Louca S., Doebeli M. 2018. Efficient comparative phylogenetics on large trees. Bioinformatics. 34:1053–1055.

Louca S., Pennell M.W. 2020. A General and Efficient Algorithm for the Likelihood of Diversification

and Discrete-Trait Evolutionary Models. Systematic Biology. 69:545–556.

Louca S., Shih P.M., Pennell M.W., Fischer W.W., Parfrey L.W., Doebeli M. 2018. Bacterial diversification through geological time. Nat Ecol Evol. 2:1458–1467.

Maddison W.P., Midford P.E., Otto S.P. 2007. Estimating a Binary Character's Effect on Speciation and Extinction. Systematic Biology. 56:701–710.

Maliet O., Hartig F., Morlon H. 2019. A model with many small shifts for estimating species-specific diversification rates. Nature Ecology & Evolution. 3:1086–1092.

Maliet O., Morlon H. 2022. Fast and Accurate Estimation of Species-Specific Diversification Rates Using Data Augmentation. Systematic Biology. 71:353–366.

Marin J.-M., Pudlo P., Robert C.P., Ryder R.J. 2012. Approximate Bayesian computational methods. Stat Comput. 22:1167–1180.

May M.R., Höhna S., Moore B.R. 2016. A Bayesian approach for detecting the impact of mass-extinction events on molecular phylogenies when rates of lineage diversification may vary. Methods in Ecology and Evolution. 7:947–959.

McKay M.D., Beckman R.J., Conover W.J. 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics. 21:239–239.

McPeek M.A. 2008. The Ecological Dynamics of Clade Diversification and Community Assembly. The American Naturalist. 172:E270–E284.

Morlon H. 2014. Phylogenetic approaches for studying diversification. Ecology Letters. 17:508–525.

Morlon H., Parsons T.L., Plotkin J.B. 2011. Reconciling molecular phylogenies with the fossil record. Proceedings of the National Academy of Sciences. 108:16327–16332.

Nee S., May R.M., Harvey P.H. 1994. The reconstructed evolutionary process. :7.

Nesterenko L., Boussau B., Jacob L. 2022. Phyloformer: towards fast and accurate phylogeny estimation with self-attention networks. :2022.06.24.496975.

Pedregosa F., Michel V., Grisel OLIVIERGRISEL O., Blondel M., Prettenhofer P., Weiss R., Vanderplas J., Cournapeau D., Pedregosa F., Varoquaux G., Gramfort A., Thirion B., Grisel O., Dubourg V., Passos A., Brucher M., Perrot andÉdouardand M., Duchesnay andÉdouard, Duchesnay EDOUARDDUCHESNAY Fré. 2011. Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. Journal of Machine Learning Research. 12:2825–2830.

Prangle D. 2017. Adapting the ABC Distance Function. Bayesian Analysis. 12:289–309.

Pyron R.A., Wiens J.J. 2013. Large-scale phylogenetic analyses reveal the causes of high tropical amphibian diversity. Proc. R. Soc. B. 280:20131622.

Rabosky D.L. 2014. Automatic Detection of Key Innovations, Rate Shifts, and Diversity-Dependence on Phylogenetic Trees. PLoS ONE. 9:e89543.

Rabosky D.L., Chang J., Title P.O., Cowman P.F., Sallan L., Friedman M., Kaschner K., Garilao C., Near T.J., Coll M., Alfaro M.E. 2018. An inverse latitudinal gradient in speciation rate for marine fishes. Nature. 559:392–395.

Raynal L. 2019. Bayesian statistical inference for intractable likelihood models. .

Richter F., Haegeman B., Etienne R.S., Wit E.C. 2020. Introducing a general class of species diversification models for phylogenetic trees. Statistica Neerlandica. 74:261–274.

Rolland J., Condamine F.L., Jiguet F., Morlon H. 2014. Faster Speciation and Reduced Extinction in the Tropics Contribute to the Mammalian Latitudinal Diversity Gradient. PLoS Biol. 12:e1001775.

Ronquist F., Kudlicka J., Senderov V., Borgström J., Lartillot N., Lundén D., Murray L., Schön T.B., Broman D. 2021. Universal probabilistic programming offers a powerful approach to statistical

phylogenetics. Commun Biol. 4:1–10.

Sanchez T., Cury J., Charpiat G., Jay F. 2020. Deep learning for population size history inference: Design, comparison and combination with approximate Bayesian computation. Molecular Ecology Resources.

Saulnier E., Gascuel O., Alizon S. 2017. Inferring epidemiological parameters from phylogenies using regression-ABC: A comparative study. PLoS Comput Biol. 13:e1005416.

Scott J.E. 2018. Reevaluating cases of trait-dependent diversification in primates. American Journal of Physical Anthropology. 167:244–256.

Selvaraju R.R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D. 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. Int J Comput Vis. 128:336–359.

Senderov V. 2023. TreePPL. Available from http://treeppl.org//.

Sheehan S., Song Y.S. 2016. Deep Learning for Population Genetic Inference. PLoS Comput Biol. 12:e1004845.

Sisson S.A., Fan Y., Beaumont M. 2018. Handbook of Approximate Bayesian Computation. CRC Press.

Sisson S.A., Fan Y., Tanaka M.M. 2007. Sequential Monte Carlo without likelihoods. Proceedings of the National Academy of Sciences. 104:1760–1765.

Skeels A., Bach W., Hagen O., Jetz W., Pellissier L. 2022. Temperature-Dependent Evolutionary Speed Shapes the Evolution of Biodiversity Patterns Across Tetrapod Radiations. Systematic Biology.:syac048.

Solis-Lemus C., Yang S., Zepeda-Nunez L. 2022. Accurate Phylogenetic Inference with a Symmetry-preserving Neural Network Model. .

Srivastava N., Hinton G., Krizhevsky A., Salakhutdinov R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 15:1929–1958.

Stadler T. 2009. On incomplete sampling under birth–death models and connections to the sampling-based coalescent. Journal of Theoretical Biology. 261:58–66.

Stadler T. 2011. Mammalian phylogeny reveals recent diversification rate shifts. Proceedings of the National Academy of Sciences. 108:6187–6192.

Stadler T. 2013. Recovering speciation and extinction dynamics based on phylogenies. Journal of Evolutionary Biology. 26:1203–1219.

Stone B.W., Wolfe A.D. 2021. Asynchronous rates of lineage, phenotype, and niche diversification in a continental-scale adaptive radiation. .

Sukumaran J., Economo E.P., Lacey Knowles L. 2016. Machine Learning Biogeographic Processes from Biotic Patterns: A New Trait-Dependent Dispersal and Diversification Model with Model Choice By Simulation-Trained Discriminant Analysis. Syst Biol. 65:525–545.

Suvorov A., Hochuli J., Schrider D.R. 2020. Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning. Systematic Biology. 69:221–233.

Swanepoel C., Fourment M., Ji X., Nasif H., Suchard M.A., Matsen IV F.A., Drummond A. 2022. TreeFlow: probabilistic programming and automatic differentiation for phylogenetics. .

Thompson A., Liebeskind B., Scully E.J., Landis M. 2023. Deep learning approaches to viral phylogeography are fast and as robust as likelihood methods to model misspecification. .

Varin C., Reid N., Firth D. 2021. AN OVERVIEW OF COMPOSITE LIKELIHOOD METHODS. :39.

Vasconcelos T., O'Meara B.C., Beaulieu J.M. 2022. A flexible method for estimating tip diversification rates across a range of speciation and extinction scenarios. Evolution. 76:1420–1433.

Villarreal J., Renner S.S. 2013. Correlates of monoicy and dioicy in hornworts, the apparent sister group to vascular plants. BMC Evol Biol. 13:239.

Voznica J., Zhukova A., Boskova V., Saulnier E., Lemoine F., Moslonka-Lefebvre M., Gascuel O. 2022. Deep learning from phylogenies to uncover the epidemiological dynamics of outbreaks. Nat Commun. 13:3896.

Williams J.H., Taylor M.L., O'Meara B.C. 2014. Repeated evolution of tricellular (and bicellular) pollen. American Journal of Botany. 101:559–571.

Yang A., Zhang W., Wang J., Yang K., Han Y., Zhang L. 2020. Review on the Application of Machine Learning Algorithms in the Sequence Data Mining of DNA. Frontiers in Bioengineering and Biotechnology. 8:1032–1032.

Yang Z., Rannala B. 1997. Bayesian phylogenetic inference using DNA sequences: a Markov Chain Monte Carlo Method. Molecular Biology and Evolution. 14:717–724.

Zhou J., Cui G., Hu S., Zhang Z., Yang C., Liu Z., Wang L., Li C., Sun M. 2020. Graph neural networks: A review of methods and applications. AI Open. 1:57–81.

Zou Z., Zhang H., Guan Y., Zhang J., Liu L. 2020. Deep Residual Neural Networks Resolve Quartet Molecular Phylogenies. Molecular Biology and Evolution. 37:1495–1507.

Table 1: Parameterization of the constant-rate birth-death model with incomplete sampling.

Parameters	Symbols	Ranges
Turnover rate	**ε**	**U(0.01,1)**
Speciation rate	**λ**	**U(0.01,0.5)**
Tree size	s	U(200, 500)
Sampling fraction	f	U(0.01,1)

Table 2: Parameterization of the Binary State Speciation and Extinction model with incomplete sampling.

Parameters	Symbols	Range
Turnover rate	**ε**	**U(0,1)**
Speciation rate 1	**λ_1**	**U(0.01,1)**
Speciation rate 2 and its ratio to λ_1	**λ_2**, $r_{\lambda 2}$	**[10e-3,1]** U(0.1,1)
Transition rate and its ratio to λ_1	**q_{12}=q_{21}**, r_q	**[10e-4,0.1]** U(0.01,0.1)
Tree size	s	U(200, 500)
Sampling fraction	f	U(0.01,1)

Figure 1

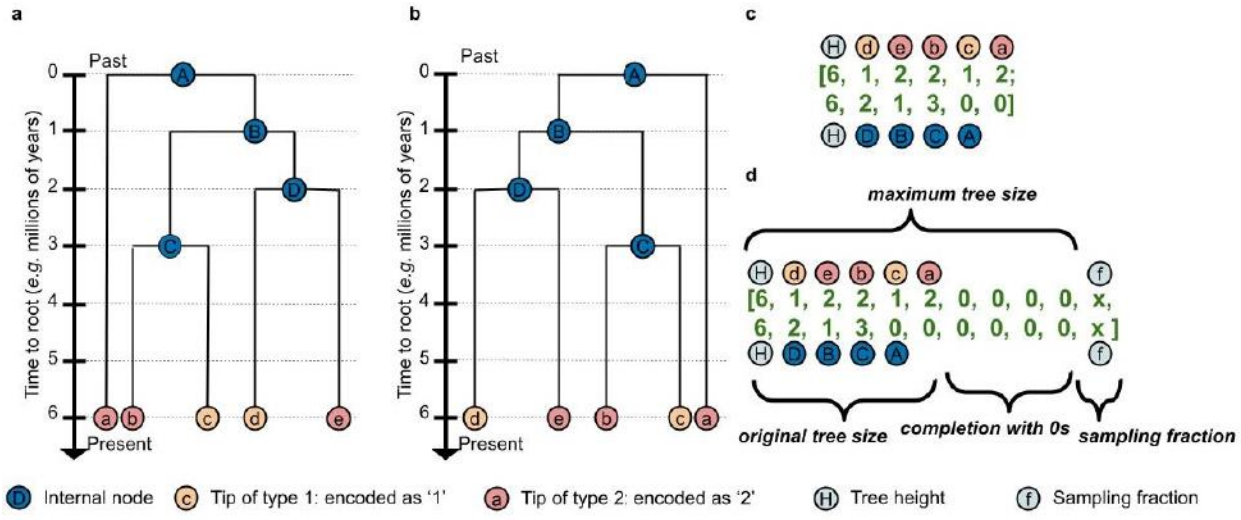

Figure 2

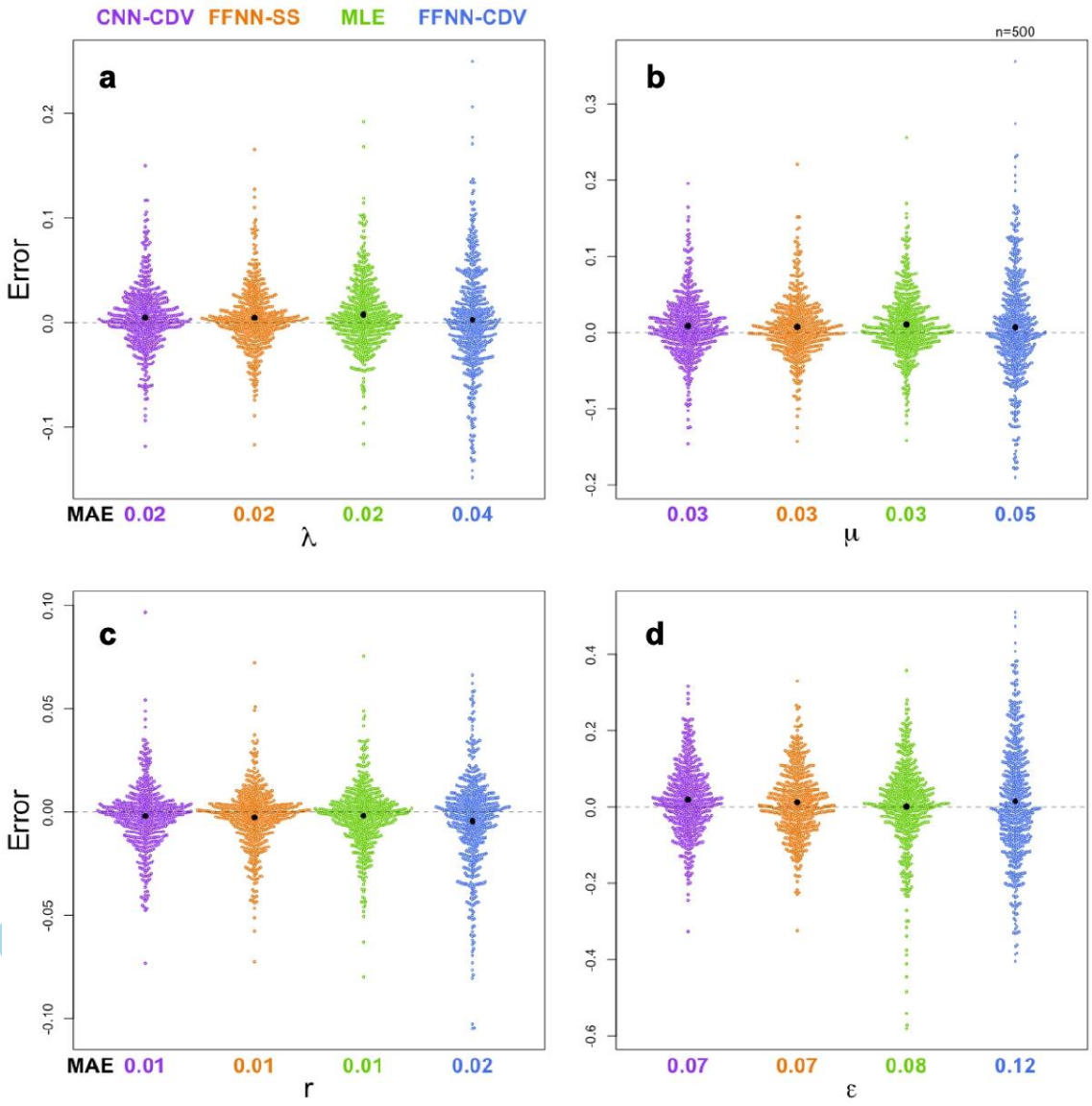

Figure 3

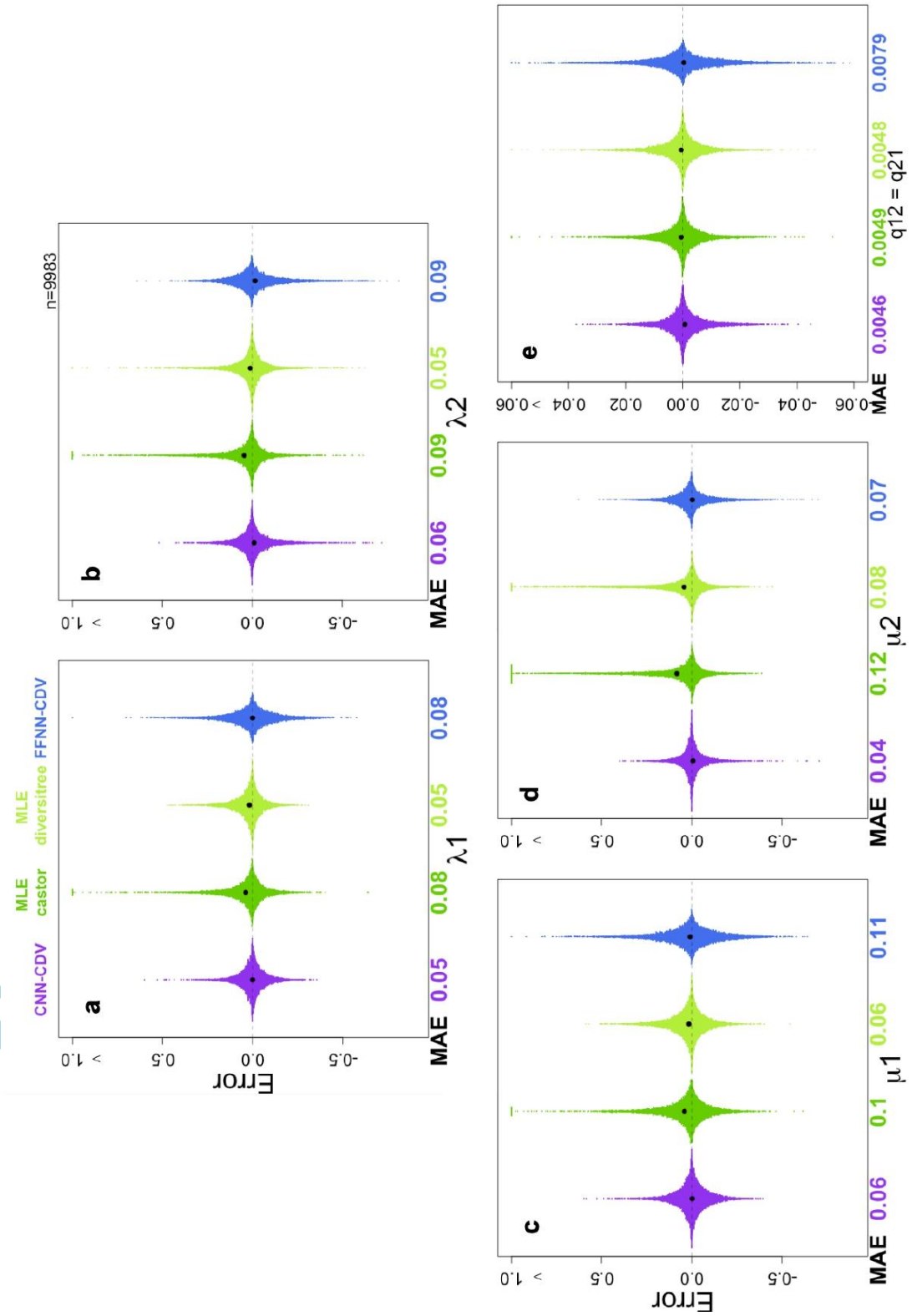

Figure 4

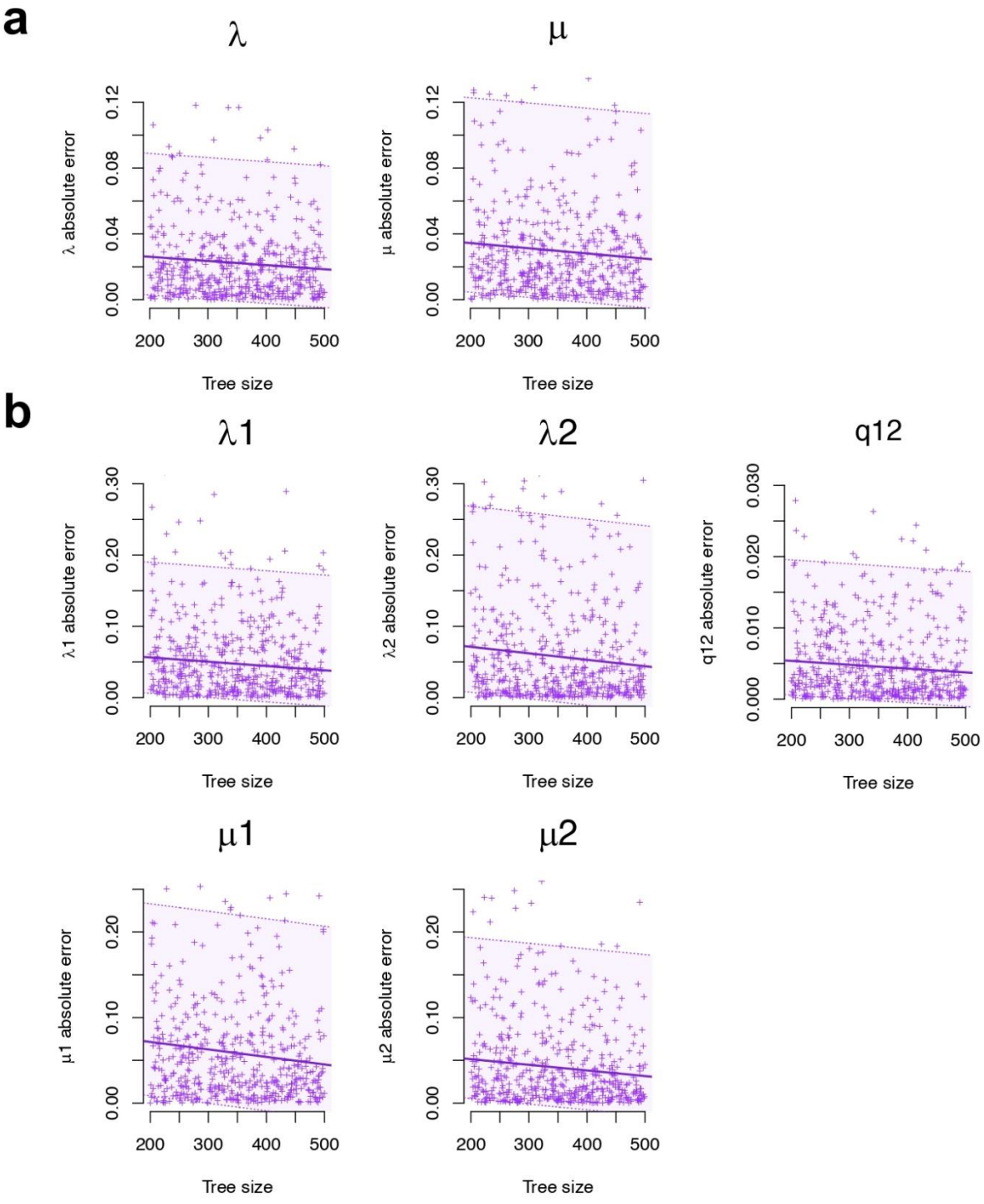

Figure 5

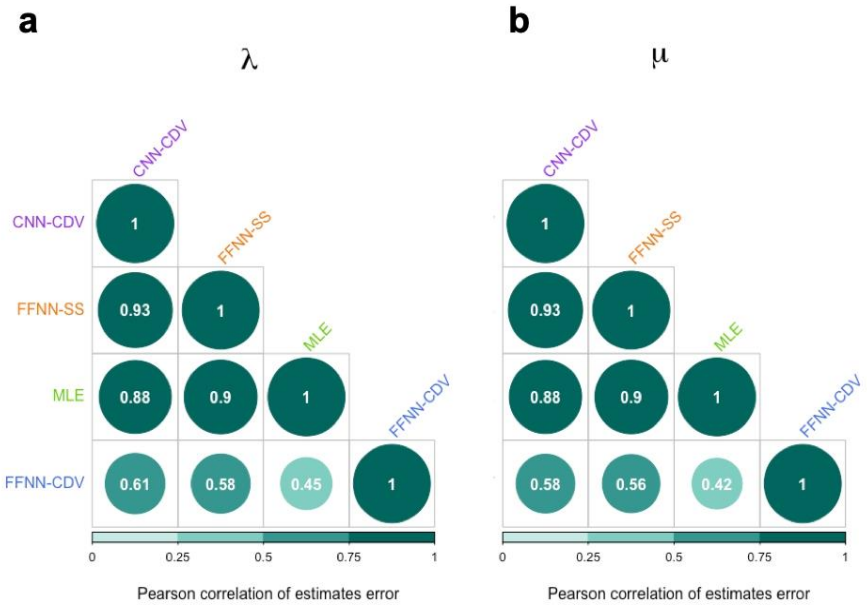

Figure 6

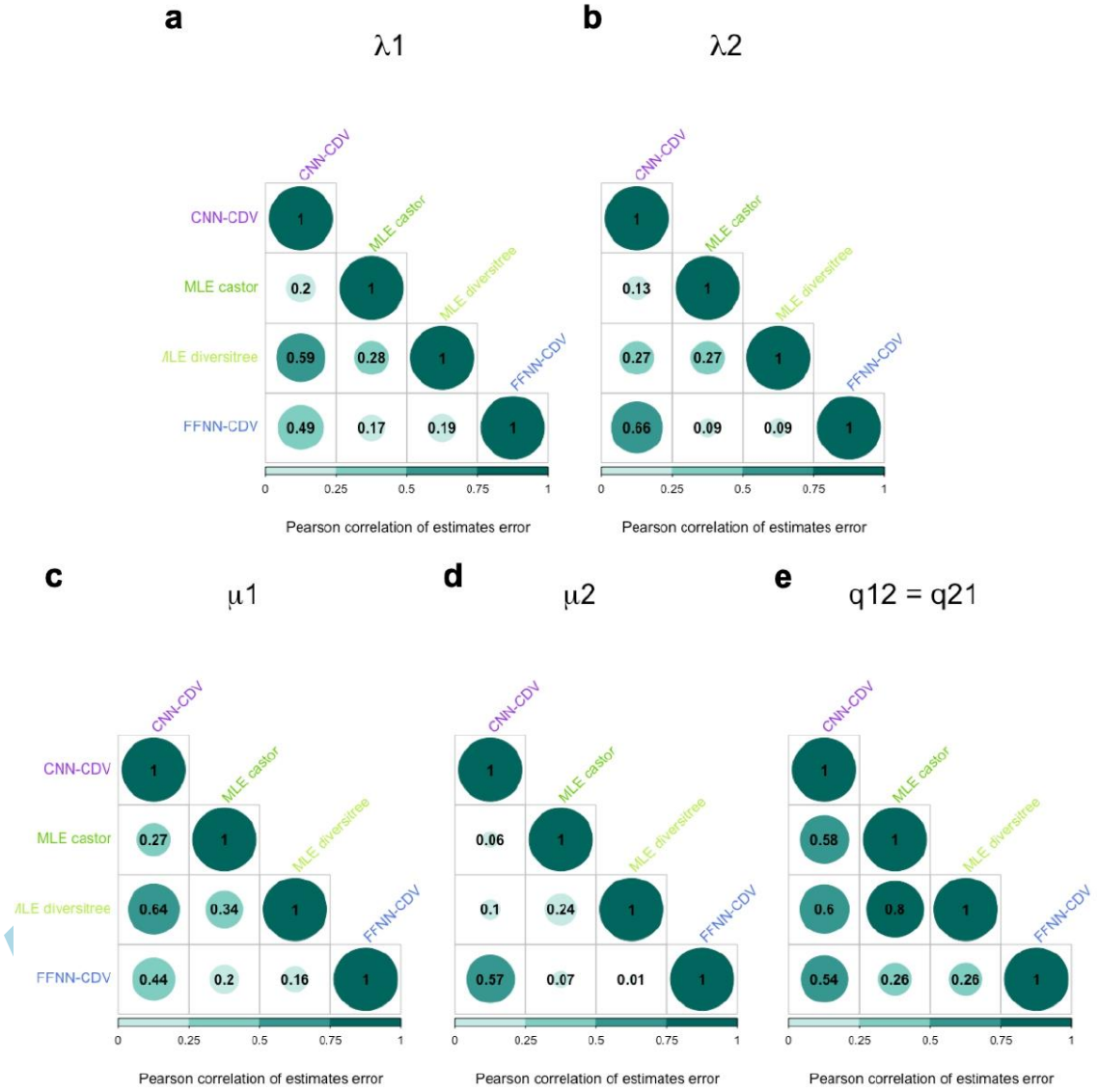

Figure 7