Accurate inference of tree topologies from multiple sequence alignments using deep learning

Anton Suvorov[1*], Joshua Hochuli[2] and Daniel R. Schrider[1]

[1]Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA

[2]Biological & Biomedical Sciences Program, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA

*Corresponding author
antony.suvorov@med.unc.edu

Abstract

Reconstructing the phylogenetic relationships between species is one of the most formidable tasks in evolutionary biology. Multiple methods exist to reconstruct phylogenetic trees, each with their own strengths and weaknesses. Both simulation and empirical studies have identified several "zones" of parameter space where accuracy of some methods can plummet, even for four-taxon trees. Further, some methods can have undesirable statistical properties such as statistical inconsistency and/or the tendency to be positively misleading (i.e. assert strong support for the incorrect tree topology). Recently, deep learning techniques have made inroads on a number of both new and longstanding problems in biological research. Here we designed a deep convolutional neural network (CNN) to infer quartet topologies from multiple sequence alignments. This CNN can readily be trained to make inferences using both gapped and ungapped data. We show that our approach is highly accurate, often outperforming traditional methods, and is remarkably robust to bias-inducing regions of parameter space such as the Felsenstein zone and the Farris zone. We also demonstrate that the confidence scores produced by our CNN can more accurately assess support for the chosen topology than bootstrap and posterior probability scores from traditional methods. While numerous practical challenges remain, these findings suggest that deep learning approaches such as ours have the potential to produce more accurate phylogenetic inferences.

Introduction

Reconstructing the historical relationships among biological sequences remains one of the most important challenges in evolutionary biology. In order to reconstruct a tree it is first necessary to acquire orthologous (Koonin 2005) and/or paralogous (Hellmuth, et al. 2015) sequences from a sample of species or populations. Generally, the next step involves producing a multiple sequence alignment (MSA) in order to compare orthologous sequence characters, although in recent years a number of methods that bypass explicit orthology and alignment searches have been proposed (Hohl and Ragan 2007; Bonham-Carter, et al. 2013). Finally, any one of a plethora of tree reconstruction methods can be employed to infer a tree from an MSA. Determining the topology of a tree is an NP-hard problem (Roch 2006), as the space of possible solutions grows double-factorially with increasing numbers of sequences (Felsenstein 1978b). We are thus limited to methods that utilize various heuristics and/or make assumptions about the

process of sequence evolution in order to efficiently traverse tree space, including maximum parsimony (MP: Farris 1970; Fitch 1971), distance-based approaches such as neighbor joining (NJ: Saitou and Nei 1987), maximum-likelihood (ML: Felsenstein 1981), and Bayesian inference (BI: Rannala and Yang 1996; Li, et al. 2000; Huelsenbeck and Ronquist 2001). Although none of these methods are guaranteed to produce a global solution (i.e. the true tree topology), one can evaluate and compare their effectiveness.

Supervised machine learning algorithms represent an alternative framework for drawing inferences from data, including sequence data. These methods have been successfully applied to various areas of biological research (Tarca, et al. 2007), and have recently been introduced to evolutionary biology—population genetics in particular (Schrider and Kern 2018). Briefly, given some multivariate observation $\boldsymbol{x}$ that is associated with a response variable $\boldsymbol{y}$, supervised machine learning seeks to create a function $f(\boldsymbol{x})$ that predicts $\boldsymbol{y}$. This is achieved through a process called training, wherein a set of observations with known response variables are examined by an algorithm that creates and tunes the function f to minimize the disparity between $f(\boldsymbol{x})$ and $\boldsymbol{y}$ on this training set. Typically, $\boldsymbol{x}$ is summarized by a vector of values, or features. Convolutional neural networks (CNNs: Lecun, et al. 1998), a class of deep learning algorithms (Goodfellow, et al. 2016), are able to learn to extract informative features from data arranged in a matrix in order to make a prediction; thus CNNs do not require a predefined feature vector. CNNs have proved extremely effective in analyzing image data (Krizhevsky, et al. 2012) and have recently been shown to achieve impressive accuracy on a number of tasks in population genetic inference when applied to MSAs (Chan, et al. 2018; Flagel, et al. 2018). CNNs and other deep learning methods may thus prove to make sizeable gains in a number of inference tasks involving biological sequence data.

Here we propose a novel approach applying CNNs to phylogenetic inference. We show that CNNs can be trained to extract phylogenetic signal from an MSA and use it to accurately reconstruct unrooted gene tree topologies (i.e. here our $\boldsymbol{y}$ is a tree topology) when given an alignment of four sequences. We adopt a classification approach, wherein the CNN is trained to discriminate among discrete data classes which in this case correspond to the three possible tree topologies. We find that our CNN approach is highly accurate, can naturally include indel information, is fairly robust to classical biases affecting tree reconstruction and has several other benefits over existing methods. Below we describe our methodology and the CNN's

performance in detail before concluding with a brief discussion of limitations, potential extensions,and outstanding practical challenges for adapting deep neural networks to phylogenomics.

Methods

MSA Simulations

For 4-taxon case there exist exactly three possible unrooted tree topologies: ((A,B),(C,D)), ((A,C),(B,D)) and ((A,D),(B,C)) in Newick notation. We simulated an equal number of MSAs from each of these topologies using INDELible (Fletcher and Yang 2009), which allows for the inclusion of indels. The MSA length was set to 1000 sites unless stated otherwise. Trees with various branch length configurations were sampled uniformly from a "branch-length space" (BL-space) that encompasses a wide variety of branch lengths and nodal rotations (see **Supplementary Text and Supplementary Fig. S1)**. Tree branch lengths are expressed in expected number of substitutions per site.

With the tree topology and branch lengths in hand, sequences were generated under a randomly assigned Markov model of nucleotide substitution using INDELible. We used five nested models of increasing complexity: JC (Jukes and Cantor 1969), TIM, TIMef (Posada 2003), GTR (Tavaré 1986) and UNREST (Yang 1994) accommodating rate heterogeneity along the sequence by drawing the rate parameter of a continuous gamma distribution (+Γ) from $U(0,4)$. We also allowed the proportion of invariable sites (p_{inv}) in an MSA to vary, drawing this parameter (+I) from $U(0,1)$. We note that the average expected substitutions per site (i.e. the branch length) is specified for variable sites only, not for the entire MSA. For example, when simulating an MSA of length 1000 and $p_{\text{inv}} = 0.5$, the specified branch lengths refer only to the 50% of sites that are free to vary. Substitution rate parameters and nucleotide frequencies were drawn from $U(0,3)$ and a four-category Dirichlet distribution ($\alpha = 30$), respectively. Finally, when including indels in the simulation we used identical fixed insertion and deletion rates ($\lambda_{\text{I}} = \lambda_{\text{D}} = 0.01$) and drew indel lengths from a Zipf distribution with $a = 1.5$, similar to empirical estimates (Vialle, et al. 2018), and a maximum indel size of 50. For each topology class we simulated either 5×10^4, 1.5×10^5 or 3×10^5 MSAs for training and 2×10^4 MSAs for validation and test datasets. Additionally, we simulated test sets consisting of 3000 trees from each of seven regions of the tree-shape space that may potentially cause biased tree topology inference,

including the Farris and Felsenstein zones that have long been known to negatively affect performance of classical methods (Swofford, et al. 2001). The parameters for each of these were similar to those stated above with branch lengths drawn from the distributions shown in **Table 1**. Additionally we simulated trees within so called "twisted" Farris zone (McTavish, et al. 2015) with asymmetrical branch lengths as well as within region where B_5 is short. Throughout this paper we denote sister branches or their branch lengths on the left side of the tree as B_1 and B_2, those on the right side as B_3 and B_4, and the internal branch as B_5 The overall pipeline is summarized in **Figure 1**.

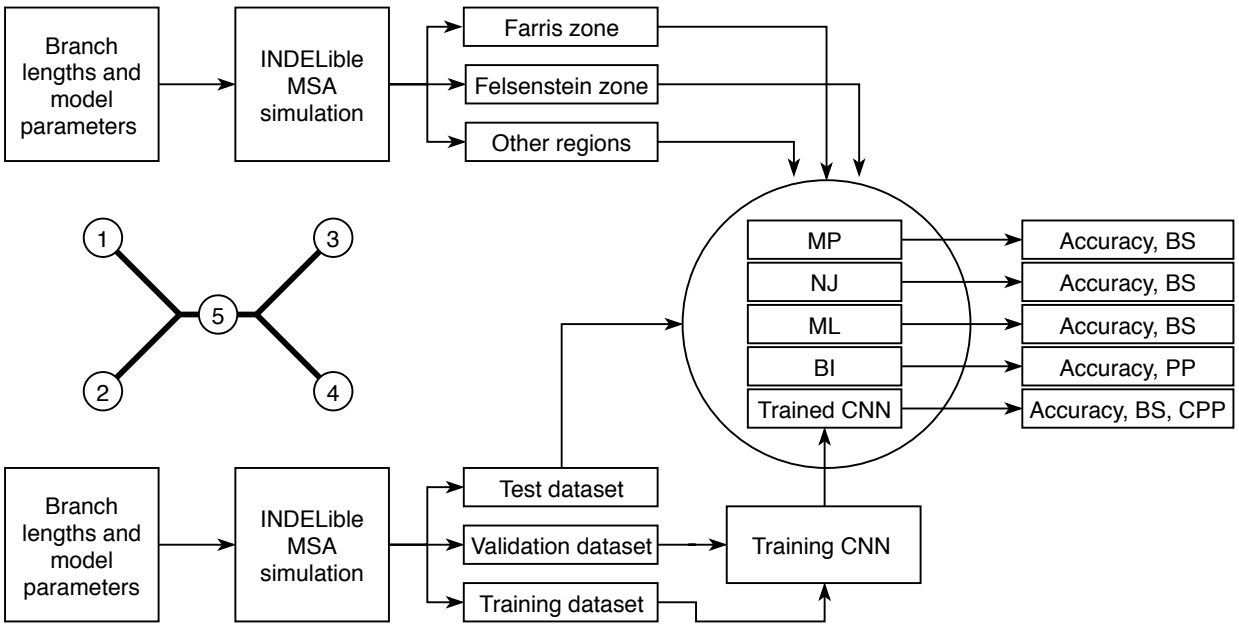

Figure 1: Summary of the simulation and testing procedures. The top branch of the diagram represents the steps that were used to simulate various datasets with heterogeneous branch lengths (**Table 1** and **Supplementary Table S1**) to test methods' performance. The bottom branch of the diagram represents the steps that were used to simulate training and validation datasets to train the CNN. The test dataset was used to estimate accuracy for each method. The training, validation and test datasets were simulated using the procedure described in **Supplementary Text**. MSA=multiple sequence alignment, MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network, BS=bootstrap, PP=posterior probabilities, CPP=class membership posterior probability. The tree shows the branch labelling convention that was used throughout the paper, where numbers in circles correspond to branches B_1, B_2, B_3, B_4 and B_5.

Table 1: Branch length distributions for test sets designed to assess the presence of bias in topology inference. B_1 and B_2 and B_3 and B_4 denote sister branches on the left and right sides of a tree respectively whereas B_5 denotes internal branch. $U(0.5,1)+U(0,0.05)$ is a sum of two uniform random variables which follows the Irwin-Hall distribution.

Region	B_1	B_2	B_3	B_4	B_5
Farris zone	$U(0.5,1)$	$U(0.5,1)$	$U(0,0.05)$	$U(0,0.05)$	$U(0,0.05)$
Twisted Farris zone	$U(0.5,1)$	$U(0.5,1)+U(0,0.05)$	$U(0,0.1)$	$U(0,0.05)$	$U(0,0.05)$
Felsenstein zone	$U(0.5,1)$	$U(0,0.05)$	$U(0.5,1)$	$U(0,0.05)$	$U(0,0.05)$
Short internal branch	$U(0,1)$	$U(0,1)$	$U(0,1)$	$U(0,1)$	$U(0,0.05)$

The internal branch length (B_3) for these zones was drawn from $U(0,0.05)$ (Table 1). In order to investigate effect of internal branch length on each method's performance in more detail, we also created variants of the Farris and Felsenstein zones where B_3 was drawn the interval $U(0,1)$ (**Supplementary Table S1**). For instance, previous simulation studies suggest that methodological biases primarily arise when an internal branch appears to be very short, i.e. internal branch length ≤0.05 (Swofford, et al. 2001). However, since these studies followed overly simplified simulation approaches (e.g. MSA simulation under JC model only Siddall 1998), this conclusion may not hold for MSA datasets generated using more complex simulation strategies that incorporate various substitution models, rate heterogeneity, varying proportions of invariant sites and indels.

CNN Architecture and Training

This section describes the structure of our neural network and training procedure, and thus many of the concepts herein may be foreign to some readers. We refer those interested in an accessible overview of CNNs in the context of sequence alignments to Flagel, et al. (2018). We implemented our CNN in Python using version 2.2.4 of the Keras API (https://keras.io/) with TensorFlow 1.12.0 (Abadi, et al. 2016) as the backend. To extract phylogenetically informative features from an MSA we used a 1D convolution-pooling strategy. We represented each MSA as a matrix with rows corresponding to sequences and columns to sites in the alignment, and with each character in the alignment encoded as a number ('A':0, 'T':1, 'C':2, 'G':3, '-':4), and encoded the MSA's associated topology encoding as a numeric class label. Since indels result in MSAs of varying lengths and our CNN requires all input matrices to have equal size, shorter alignments were extended by "padding" the MSA with additional columns wherein each value was arbitrarily set to -15 until the MSA's length matched that of the longest alignment in our simulated data. Our CNN consisted of eight convolution layers with ReLU activation (Nair and

E. Hinton 2010) and each followed by an average-pooling step, one 1024-node fully connected layer with ReLU activation and an output layer with three nodes corresponding to the three possible topology classes. We used softmax as the activation function for our output layer in order to assign a class membership probability to each of the three classes. We used categorical cross-entropy as our loss (i.e. error) function. During training we used the adaptive moment estimation (Adam) optimizer (Kingma and Ba 2014).

We set the filter size for the first convolution operation to 4×1, reasoning that it would potentially capture a phylogenetic signal by striding across the MSA and examining one column of the alignment at a time. The total number of filters was set to 1024 for the first and second convolutional layer and set to 128 for the remaining convolutional layers. The first average-pooling operation size was set to 1×1 (i.e. no pooling), then all the subsequent convolution steps had equal sizes of 1×2 and pooling steps had equal sizes of 1×4, 1×4, 1×4, 1×2, 1×2 and 1×1. In order to avoid possible internal covariate shift and model overfitting we used batch normalization (Ioffe and Szegedy 2015) followed by dropout regularization with a rate of 0.2 for each convolution layer as well as for the hidden layer. During training we used a minibatch size of 150 MSAs. The training procedure was run for a number of iterations determined by the stopping criterion (validation loss difference between the best prior iteration and the current iteration < 0.001) and patience value (the number of consecutive iterations that must satisfy the stopping criterion before halting training). We observed that the results of our training runs were fairly stable, with a given CNN architecture consistently achieving the same final accuracy on the validation set. To further examine performance during training, we visualized the surface of the loss function and training trajectories across the CNN's parameter space (i.e. the possible values of all weights in the network) as described below. This revealed that there appear to be many readily reachable local minima of the loss function that all have roughly equivalent values (**Supplementary Fig. S2**), lending further support to the notion that training results should be fairly stable, at least for the datasets and architectures examined here.

Testing Procedures

We sought to compare the performance of CNN vs. conventional tree inference methods, namely MP, NJ, ML and BI using our simulated test datasets. MP reconstructions were performed using PHYLIP's dnapars program v3.696 (Felsenstein 1989) the interface provided by SeaView v4.7

(Gouy, et al. 2010). ML analysis was implemented in IQ-TREE v1.6.5 (Nguyen, et al. 2015) restricting model selection to only the models that were used to simulate MSAs, i.e. JC, TIM, TIMef, GTR and UNREST with +I and/or +Γ, where the continuous rate gamma distribution was approximated by discrete gamma with 8 categories (G8). For each MSA, ML search was run independently 100 times; along with the inferred tree, the likelihood scores from the best ML run were recorded for the likelihood-mapping analysis (Strimmer and von Haeseler 1997). The NJ trees were generated using the distance function of the best-fit model in IQ-TREE. Note that ML and NJ approaches infer a rooted tree under the UNREST model, thus every such estimated tree was unrooted for further comparisons. Bayesian tree reconstruction was carried out in ExaBayes v1.4.1 (Aberer, et al. 2014) under default parameters starting with a random initial topology. Three independent MCMC chains were run for 10^6 generations each with two coupled chains. Tree support was assessed by standard nonparametric bootstrap (Felsenstein 1985) for MP, NJ, ML and CNN, and by posterior split probability for BI. We also assessed tree support from the CNN by simply using the estimated posterior class membership probabilities emitted by the output layer. Then, for each method we calculated topological accuracy as the number of correctly returned topologies divided by the total number of inferences performed.

Results and Discussion

CNNs are more Accurate than Standard Methods on Gapped and Ungapped MSAs

Initially, we trained our CNN on gapped and ungapped training sets of 5×10^4 MSAs (each with 1000 sites) for each of our three possible topologies, and in each case evaluated our accuracy on an independent test set. We compared the accuracy of our CNN on simulated test data (3×10^4 MSAs per topology) to that of several other standard tree estimation methods, namely maximum parsimony (MP), neighbor joining (NJ), maximum likelihood (ML), and Bayesian inference (BI). Strikingly, the CNN was substantially more accurate than the other methods (McNemar test, $P < 1\times10^{-10}$ for each comparison with the CNN) when applied to gapped MSAs (**Figure 2**). However, this was not the case when testing methods on ungapped MSAs: here accuracy of the CNN (0.77), which was in this case also trained on ungapped MSAs, was marginally yet significantly smaller than that of the best performing method, NJ (accuracy = 0.776, McNemar test, $P < 1\times10^{-10}$). Nevertheless, the CNN's accuracy improved with increasing amounts of

training data and obtained higher accuracy (0.778; **Figure 2**) than all other methods when trained on 3×10^5 ungapped MSAs per topology, though the difference with the next best method (NJ) was not significant (McNemar test, $P = 0.428$).

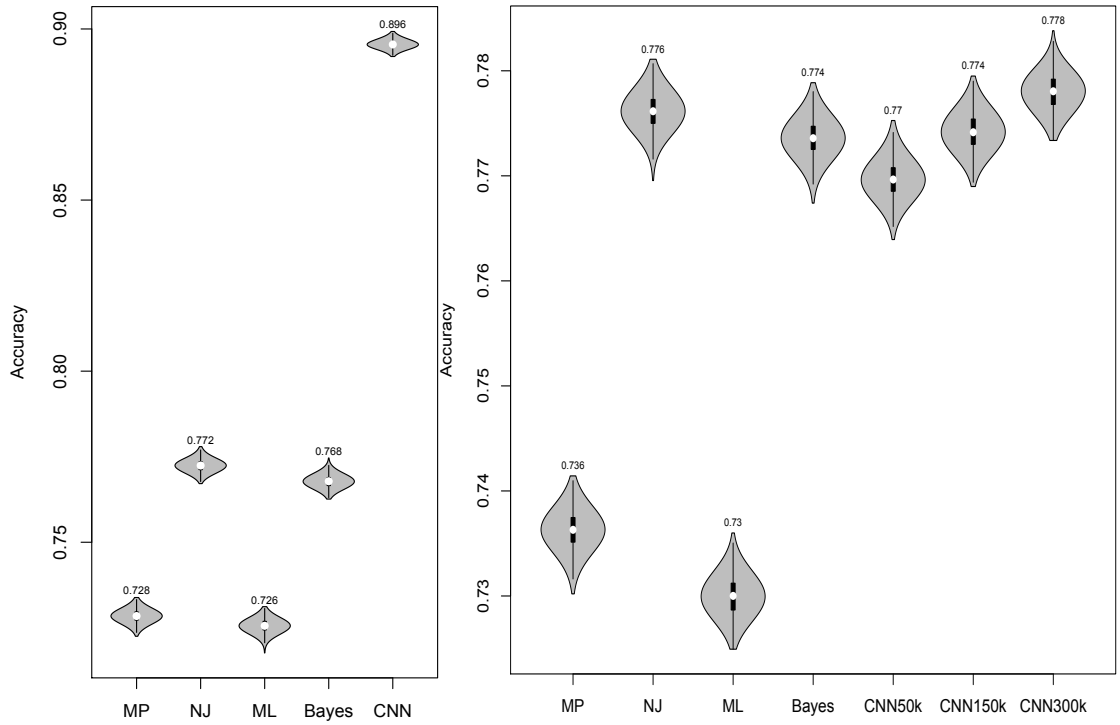

Figure 2: Accuracy of tree topology inference. The violin plots show the accuracy for different tree estimation methods achieved on gapped (left) and ungapped (right) MSA test datasets. The distribution shown in each violin plot is obtained by bootstrap resampling from the test set and calculating accuracy on each bootstrap replicate. The white dots represent the accuracy point estimate (across the complete test set) whose value is indicated above each violin.

We also investigated the relationship between several simulation parameters and inference accuracy. When sequences are very closely related, the MSA may consist largely of invariant sites, which are phylogenetically uninformative. One of the parameters of our simulations was the fraction of sites that were not allowed to vary (p_{inv}; see Methods). We found that for each method, misclassified MSAs from the gapped test set had a larger value of p_{inv} on average (Wilcoxon rank sum test, $P < 1\times10^{-10}$ for each standard method and $P = 1\times10^{-6}$ for the CNN), but this difference is far less pronounced for the CNN (**Figure 3A**). The gamma rate parameter, α, which governs mutation rate heterogeneity across the MSA does not impact the CNN's ability to infer the correct topology (Wilcoxon rank sum test, $P = 0.934$) whereas for all

other methods misclassified MSAs had elevated values of α (Wilcoxon rank sum test, $P < 1\times10^{-10}$ for all standard methods; **Figure 3B**). Similar results hold for ungapped MSAs (**Supplementary Figure S3**). Together these results suggest that CNN's are more robust to higher values of p_{inv} and α than traditional methods. Additionally, we note that the DNA substitution model does not affect any method's performance (Fisher exact test, $P > 0.05$ for all methods).

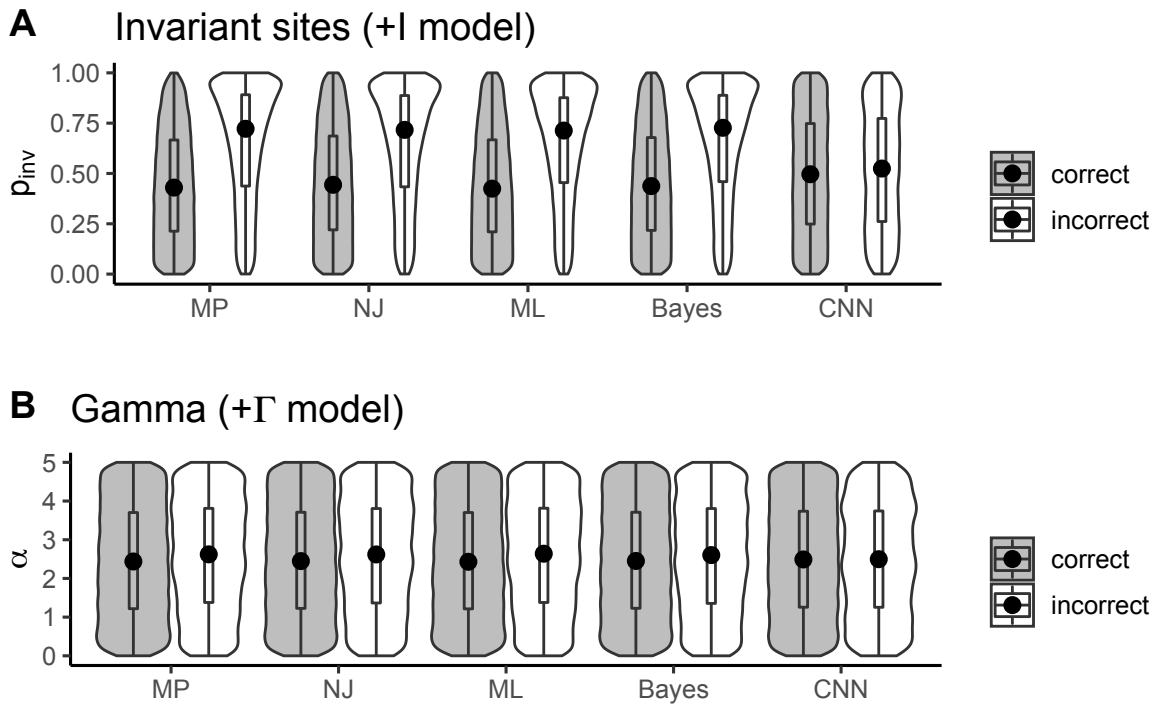

Figure 3: Differences in parameter values between correct and incorrect classifications for each method. A) The distribution of the parameter governing the fraction of invariable sites (p_{inv}) for correctly and incorrectly inferred trees. B) The distribution of gamma rate parameter (α) for correctly and incorrectly inferred trees. The black dots of each violin plot indicate median of the distribution.

Together these results suggest that our CNN can effectively extract phylogenetic signal. This is especially the case for the CNN trained on gapped alignments, which easily outcompeted the other methods, all of which treat gaps as missing data. Despite their potential to provide unambiguous phylogenetic information (Rokas and Holland 2000; Dessimoz and Gil 2010) due to reduced homoplasy relative to substitutions (Ashkenazy, et al. 2014), approaches incorporating indels are controversial and not generally utilized in practice (Ogden and Rosenberg 2007; Warnow 2012). When our CNN is trained on gapped data it learns to detect phylogenetic signal from both indels and substitutions present in an MSA. To further examine

this property of the gapped-MSA CNN, we simulated MSAs of length 1000 with no substitutions allowing only indels. Under these conditions all standard methods fail to achieve consistency (Warnow 2012; Truszkowski and Goldman 2016), and MP, NJ, and BI all had accuracy of 0.33 or lower—no better than random guessing. By contrast, the CNN's accuracy on this set was 0.965, and ML's was 0.94, which is surprising given that this method may not be consistent under this scenario (Warnow 2012).

Together these results show that our CNN, when trained on gapped MSAs, is in principle able to incorporate indel information into its phylogenetic inference. Because the CNN implicitly constructs substitution and indel process models by examining the simulated training data, alignment error could impact inference in practice. This could be handled in one of two ways: 1) by incorporating alignment errors into the training set (e.g. by running a sequence aligner on training data prior to training), or 2) judging the reliability of indels within the MSA and masking particular sites or omitting entire MSAs appearing to be unreliable (e.g. following Castresana 2000; Misof and Misof 2009; Ashkenazy, et al. 2014; Fujimoto, et al. 2016). Another potential problem is model misspecification: accuracy could suffer if indel rates/lengths differ dramatically between simulated training data and real datasets. Future work will have to investigate strategies to train a network that is robust to such misspecification, perhaps by incorporating a variety of indel models into training.

Topology Estimation Under Branch Length-Heterogeneity

There exist several regions in the branch-length parameter space where common methods fail to produce the correct topology. The bias known as long branch attraction (LBA) is the inability of MP (Felsenstein 1978a) and possibly Bayesian (Kolaczkowski and Thornton 2009; Susko 2015) methods to estimate the correct topology in presence of long nonadjacent branches (i.e. the Felsenstein zone; Figure C; Huelsenbeck and Hillis 1993); these methods tend to erroneously place such long branches together and thus favor the incorrect topology. Another region in the branch-length space, called the Farris zone (Figure A; Siddall 1998), causes ML methods to fail to correctly infer topologies that have adjacent long branches; we note that this bias may diminish when using longer alignments due to ML's property of consistency (Swofford, et al. 2001). Additionally, we assessed the performance of all methods in the asymmetrical ("twisted") Farris zone (Table 1; **Figure 4B**), where a single incorrect topology is favored if the distance

function is convex (McTavish, et al. 2015). Lastly, we examined trees with a short internal branch (≤ 0.05) but external branches lengths varying along the interval [0,1] (Table 1; **Figure 4D**).

We primarily focused on these four regions of known bias and also examined several other regions of potential bias on test sets of gapped (**Figure 4 and Supplementary Fig. S4**) and ungapped (**Supplementary Figs. S5**, **S6**) MSAs. In most of these regions, our CNN trained and tested on gapped alignments exhibited roughly matching or exceeding that of the best traditional method, whereas each other method failed catastrophically in at least one of these zones (**Figure 4**). The one exception was the Felsenstein zone, where our CNN's accuracy is 35.3%, slightly above the random expectation of 1/3, and second to ML (47.6%; **Figure 4**); however, the CNN still substantially outperformed the remaining methods in this case (16.1% accuracy for BI and ≤3% for both MP and NJ). The CNN shows superior accuracy in each of the remaining regions of potential bias (**Supplementary Table S1**; **Supplementary Fig. S4**) than the other methods. Generally, for gapped MSAs the CNN's performance mostly depends on the length of internal branch (B_5), with no misclassified topologies for trees with $B_5 > 0.4$ (**Supplementary Fig. S4**) whereas other methods may incorrectly infer topologies across the entire rage of $0 \leq B_5 \leq 1$. Our analysis also revealed that although for the Ferris and Felsenstein zones the internal branch length is typically defined as 0.05 substitutions per site or less (Swofford, et al. 2001), we find that in some cases elevated error rates may hold for longer internal branches as well ("extended" Farris and Felsenstein zones in **Supplementary Figs. S4**, **S6**)

On ungapped data, the CNN's performance roughly matched that of BI across zones. These two methods both exhibit acceptable but not exceptional accuracy in each of these different zones (**Supplementary Fig. S6**), perhaps with the exception of the Felsenstein zone where all methods other than ML have very low accuracy (≤15.5% versus 47.0% for ML). Further, we note that performance could potentially be improved by providing more training examples within troublesome regions of the branch length parameter space.

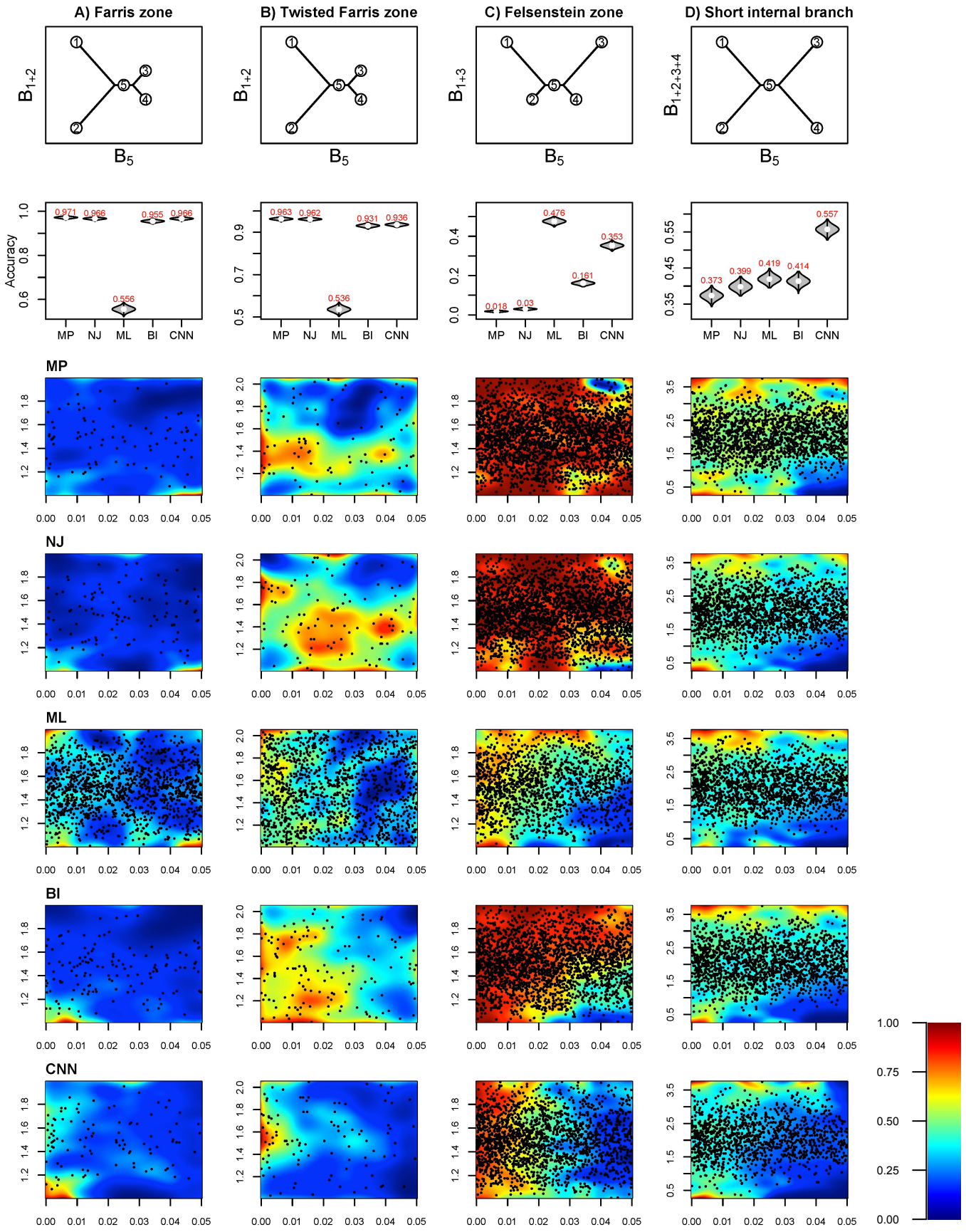
A) Farris zone
B) Twisted Farris zone
C) Felsenstein zone
D) Short internal branch
B1+2
B5
B1+3
B1+2+3+4
Accuracy
0.971
0.966
0.955
0.966
0.556
0.963
0.962
0.931
0.936
0.536
0.476
0.353
0.161
0.018
0.03
0.557
0.373
0.399
0.419
0.414
MP
NJ
ML
BI
CNN
1.00
0.75
0.50
0.25
0.00

Figure 4: Accuracy within regions with elevated branch-length heterogeneity for gapped MSAs. The four columns summarize results in the Farris zone, the twisted Farris zone, the Felsenstein zone and trees with short internal branches. The top row illustrates these tree branch length configurations, and the second row shows accuracy of each method's performance in each zone (with violins representing accuracy on bootstrap replicates from the test set). Panels three through seven show the locations (black dots) and densities (heat map) of incorrectly inferred topologies for each method plotted in two-dimensional space with B5 length on the *x*-axis and B_1+B_2 (labeled as "B_{1+2}"), B_1+B_2 ("B_{1+2}"), B_1+B_3 ("B_{1+3}"), B_1+B_2+B_3+B_4 ("$B_{1+2+3+4}$") on the *y*-axis for Farris, twisted Farris, Felsenstein and short internal branch regions respectively. See Table 1 for branch length simulation details.

Classification Accuracy Increases with Alignment Length

Statistical consistency is defined as the ability of a method, M, to return the true parameter value, θ (e.g. tree topology), with probability approaching unity when amount of data (D) increases (with alignment length measured in nucleotides, n). More formally, a method is consistent when:

$$\lim_{n\to\infty} P(M(D_n) = \theta) = 1$$

Statistical consistency has been considered as one of the major criteria for a good tree reconstruction method (Felsenstein 1978a; Goldman 1990; Penny, et al. 1992). Consistency of machine learning models has been only formally proven for a multilayer perceptron regression estimator with a particular network architecture (Mielniczuk and Tyrcha 1993). Generally, it is not trivial to analytically derive the global theoretical guarantees for neural networks due to their complexity and flexibility as exemplified by their large number of hyperparameters (e.g. different network architectures, activation and loss functions and optimization algorithms), which can be tuned to accommodate a certain learning problem. Additional factors such as sizes of both the training dataset and the input data (in our case number of sequences in the MSA and the MSA's length) may also affect the degree of consistency. Here we considered three consistent methods (provided the correct substitution model is used) of conventional tree estimation: NJ (Saitou and Nei 1987), ML (Felsenstein 1981; Truszkowski and Goldman 2016) and BI (Steel, et al. 2013; Warnow 2018) as well as MP as an example of an inconsistent (Felsenstein 1978a) method under standard Markov substitution models. Because empirical datasets are always finite in length, the rate (Fu 1975) at which $P(M(D_n) = \theta)$ increases with n is perhaps more crucial in practice.

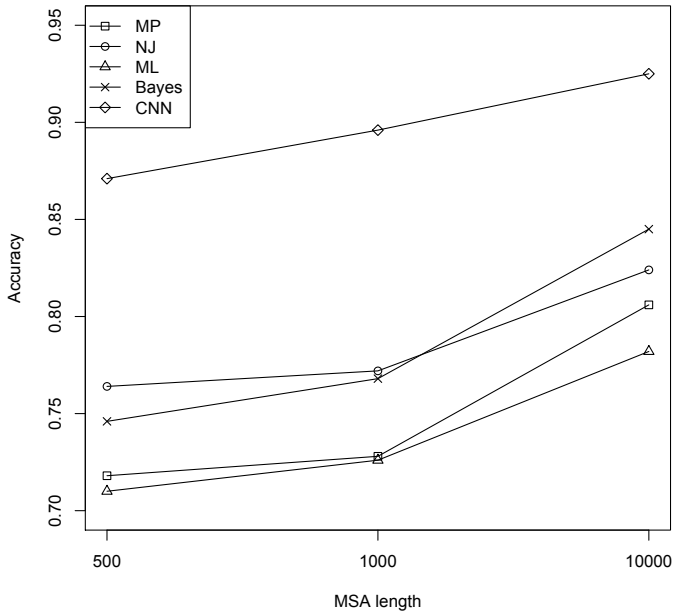

Figure 5: Estimation accuracy on gapped MSAs of various lengths.

We tested our CNN and the four standard methods on simulated sets of MSAs of increasing length. Our results show that for both gapped MSAs each method's accuracy increases with MSA length, and that the CNN outperforms other methods for each MSA length (**Figure 5**). Thus, although we do not prove that CNNs are consistent, these empirical results are highly encouraging and imply that our approach has great potential for practical applications. Indeed, in some instances accuracy on shorter MSAs may be a more crucial quality than provable consistency because asymptotic properties of consistent methods may not be exhibited for smaller datasets (Yang 2014).

Analysis of Phylogenetic Reliability Measures

Testing phylogenetic hypotheses usually involves calculation of some reliability measures such as bootstrap support (BS) for MP, NJ and ML or posterior probabilities (PP) in the context of BI. Despite the fact that their direct comparison is somewhat problematic due to their fundamental differences, one can estimate which measures are more prone to support false tree topologies (Douady, et al. 2003; Huelsenbeck and Rannala 2004). For our CNN we used two confidence measures: bootstrap support for the chosen topology (obtained by applying the trained CNN to each MSA in a set of bootstrap replicates), and also used the class membership posterior probability (CPP) estimates produced by the CNN as an additional measure of topological reliability. We find that high support (BS > 0.95, or PP > 0.95 in the case of BI) is occasionally assigned to incorrectly classified trees by several methods: 5.7% of misclassifications by MP, 5.2% by NJ, 1.1% by ML, 4.7% by BI, and 3.7% by the CNN (all numbers for gapped MSAs; **Figure 6**). Thus, a very high reliability measure cutoff is necessary in order to filter out

positively misleading cases. On the other hand, the CPP estimates show a large gap between scores assigned to correctly vs. incorrectly classified topologies, with the latter being significantly lower, and only 0.1% of incorrectly classified gapped MSAs having a CPP above 0.95. In other words, the CNN can avoid incorrect classifications even when imposing a relatively relaxed posterior probability cutoff, and thus simultaneously recover a large fraction of correctly classified topologies while producing very few misclassifications. This is further illustrated by precision-recall curves (**Supplementary Fig. S7**), which show that compared to other methods the CNN achieves high sensitivity while retaining a high positive predictive value on gapped alignments. Similarly, on ungapped data, the CNN performs better in this respect than the more traditional methods (**Supplementary Fig. S8**). Moreover, for our four-taxon topology trees the interpretation of CPP is straightforward: the most unreliable tree estimates receive CPP of 1/3, which is equivalent to randomly picking one of the three possible topologies.

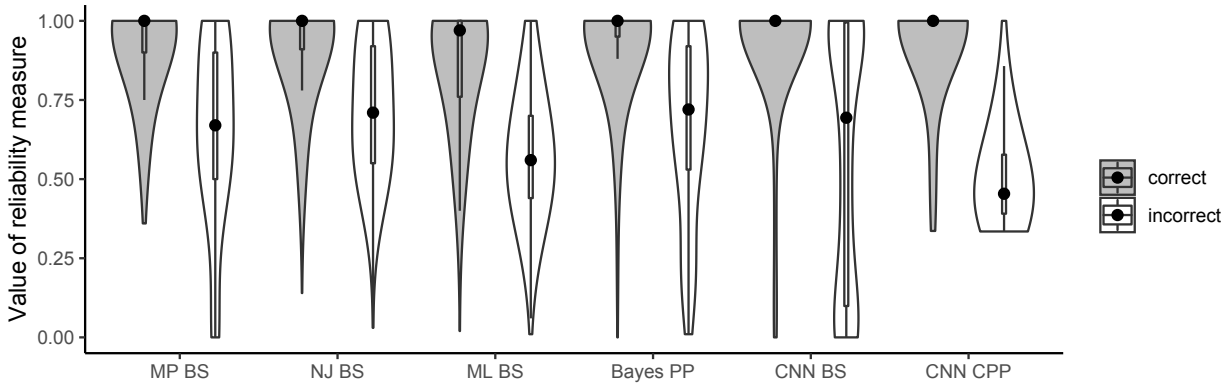

Figure 6: Comparisons of reliability measures among tree inference methods on gapped MSAs. The violin plots represent distribution of reliability measures (BS = bootstrap, PP = Bayesian posterior probability, CPP = CNN topology posterior probability) for correctly and incorrectly inferred topologies.

Visualizing Phylogenetic Value of MSAs using the CNN's Class Membership Posterior Probabilities

The likelihood-mapping method (Strimmer and von Haeseler 1997) is one of the most commonly used methods to visualize the phylogenetic information contained in an MSA and to test certain sister relationships between different lineages or taxonomic groups. This method takes as an input an MSA quartet and using ML it computes a likelihood score (L) for each of the three possible unrooted topologies. The resultant likelihood scores are then converted to posterior probabilities (P_i) according to Bayes theorem, i.e.

$$P_i = \frac{L_i}{L_1+L_2+L_3}.$$

The probability vector **P** = [P_1, P_2, P_3] can be plotted onto a 2D simplex to visualize affinity of a quartet to each of the three topologies T_1, T_2 or T_3. Conveniently, the final output layer of our CNN generates a vector of posterior probabilities (CPPs) that can be readily utilized in the same manner. We compared the distributions of the CPPs to ML's likelihood-mappings using gapped test data (**Figure 8, Supplementary Fig. S9**) and ungapped (**Supplementary Fig. S10**). Generally, the CPPs exhibit less density near the center of the simplex that corresponds to ambiguous quartets; this is especially so for the CNN using gapped MSAs. Thus, our CNN's class membership probabilities can be used to visualize the phylogenetic information in an MSA in a manner that is analogous to likelihood-mapping, but with minimal additional effort as these probabilities are computed internally as part of the classification process.

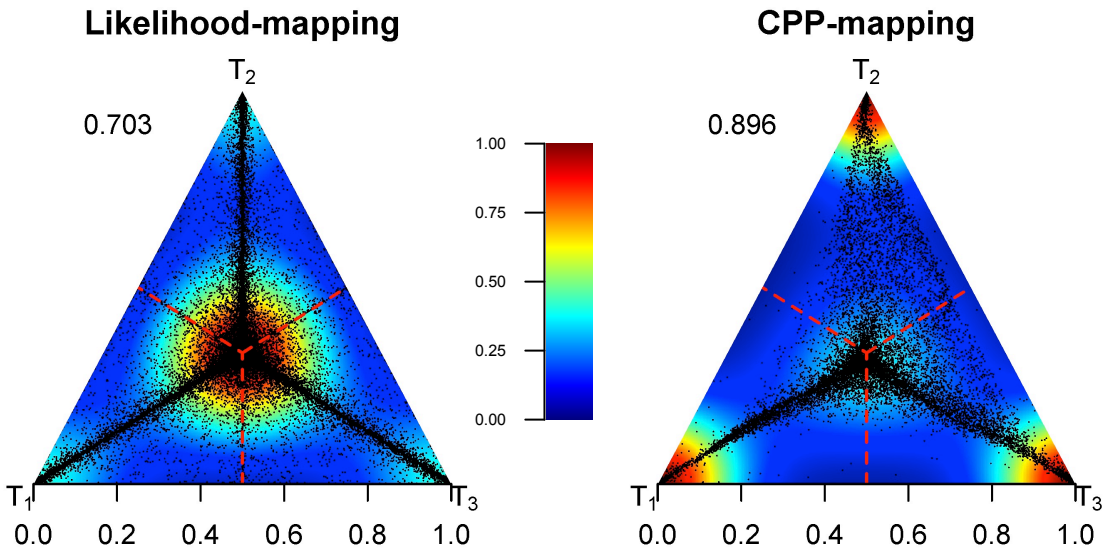

Figure 8: Distribution of likelihood-mapping probabilities and CPP-mapping within simplexes. The simplexes correspond to likelihood (left) and CNN posterior probability (right) mappings calculated on a test set of 60,000 gapped MSAs of length 1000. Accuracy is shown on the left side of simplexes. Each black dot represents the vector of posterior probability estimates for an individual MSA. The background heatmap shows estimated kernel densities of mapped probability vectors in simplex space.

Red dashed lines mark basins of attraction (i.e. regions where the probability of a topology T_i is largest) for the three topologies T_1, T_2 and T_3. In the center all of the topologies are equally likely, i.e. $P_1=P_2=P_3=1/3$.

Concluding remarks

We used a convolutional neural network to infer unrooted quartet topologies through classification. Our results demonstrate that CNNs work and have a number of desirable qualities. First and foremost, our CNN resolves quartets with higher accuracy than existing methods. This is in part due to the fact that our CNN was successfully trained to be relatively insensitive to regions of the tree branch-length heterogeneity (e.g. the Felsenstein and Farris Zones) that are known to confound certain methods. Another advantage of using CNNs is that they produce posterior probability estimates for each tree topology; this has the added benefit of making likelihood-mapping efficient and straightforward. Also, we have shown that our CNN's accuracy can be further improved by adding more training datasets, albeit most likely with diminishing returns (**Figure 2**). Although training on large datasets can impose a computational burden up front, this training only needs to be completed once, after which the CNN can be used to infer tree topologies very rapidly. Finally, CNNs have the advantage of being able to incorporate indels into their classification in a straightforward manner—one need only include them in the simulated training data—which can improve accuracy by supplying more information to the classifier. Future efforts will have to examine the robustness of inference to indel model misspecification, and the extent to which robustness can be improved by training under a mixture of indel rates and length distributions.

Our efforts thus suggest that there is great potential for the application of deep learning techniques to inferring trees, which is central to questions in both phylogenetics (species-tree inference) and population genomics (inferring ancestral recombination graphs along a recombining chromosome (Rasmussen, et al. 2014). Indeed, using the entire MSA as input would be most useful on recombining sequences—otherwise it is possible to collapse the alignment down to a vector of statistics (e.g. the number of times each distinct possible alignment column was observed, if the number of sequences is small) that would sufficiently capture all of the relevant information from the input. Deep learning methods acting directly on the alignment could also be applied to additional phylogenetic problems, such as substitution model selection or tree inference from protein sequence data.

Although our findings are encouraging, additional advances are required for our approach to match the practical utility of conventional methods. First, we have only investigated the ability of a CNN to produce the correct tree topology, but not the associated branch lengths. Artificial neural networks are capable of estimating real-valued parameters in addition to discrete class labels, so future work should investigate the possibility of using deep learning for inferring branch lengths in addition to the topology. Another limitation of our current approach is that it is limited to a small number of sequences/taxa. Future extensions could use quartet amalgamation methods (e.g. von Haeseler and Strimmer 1996; Reaz, et al. 2014), or perhaps even devise neural networks capable of directly predicting a tree topology. The latter could in principle be accomplished by treating the values of the final hidden layer of the network as entries in a distance matrix and using an output layer that builds a tree from this matrix (e.g. using NJ), and adopting a tree dissimilarity measure such as RF distance as the objective function to be minimized. Finally, in order to move from inferring gene trees to inferring species trees, our approach will have to be extended to handle tree discordance due to variation in evolutionary rates across loci, incomplete lineage sorting (Maddison and Knowles 2006; Degnan and Rosenberg 2009) and gene flow (Maddison 1997). Supervised learning approaches may prove to be well suited for this question, as arbitrarily complicated phylogenetic models combining multiple sources of discordance can readily be incorporated into the training process via simulation. Thus, while additional practical developments are necessary, applications of deep learning to problems in tree inference and phylogenomics are likely to be a fruitful endeavor.

Acknowledgements

This work was funded by the National Institutes of Health under award number R00HG008696.

References

Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, Corrado GS, Davis A, Dean J, Devin M*, et al.* 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. eprint arXiv:1603.04467:arXiv:1603.04467.

Aberer AJ, Kobert K, Stamatakis A. 2014. ExaBayes: massively parallel bayesian tree inference for the whole-genome era. Molecular biology and evolution, 31:2553-2556.

Ashkenazy H, Cohen O, Pupko T, Huchon D. 2014. Indel reliability in indel-based phylogenetic inference. Genome biology and evolution, 6:3199-3209.

Bonham-Carter O, Steele J, Bastola D. 2013. Alignment-free genetic sequence comparisons: a review of recent approaches by word analysis. Briefings in bioinformatics.

Castresana J. 2000. Selection of conserved blocks from multiple alignments for their use in phylogenetic analysis. Molecular biology and evolution, 17:540-552.

Chan J, Perrone V, Spence JP, Jenkins PA, Mathieson S, Song YS. 2018. A Likelihood-Free Inference Framework for Population Genetic Data using Exchangeable Neural Networks. bioRxiv.

Degnan JH, Rosenberg NA. 2009. Gene tree discordance, phylogenetic inference and the multispecies coalescent. Trends in ecology & evolution, 24:332-340.

Dessimoz C, Gil M. 2010. Phylogenetic assessment of alignments reveals neglected tree signal in gaps. Genome biology, 11:R37.

Douady CJ, Delsuc F, Boucher Y, Doolittle WF, Douzery EJP. 2003. Comparison of Bayesian and Maximum Likelihood Bootstrap Measures of Phylogenetic Reliability. Molecular biology and evolution, 20:248-254.

Farris JS. 1970. Methods for Computing Wagner Trees. Syst Zool, 19:83-92.

Felsenstein J. 1978a. Cases in Which Parsimony or Compatibility Methods Will Be Positively Misleading. Syst Zool, 27:401-410.

Felsenstein J. 1978b. The Number of Evolutionary Trees. Syst Zool, 27:27-33.

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. Journal of molecular evolution, 17:368-376.

Felsenstein J. 1985. Confidence-Limits on Phylogenies - an Approach Using the Bootstrap. Evolution; international journal of organic evolution, 39:783-791.

Felsenstein J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics, 5:164-166.

Fitch WM. 1971. Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. Syst Zool, 20:406-416.

Flagel L, Brandvain YJ, Schrider DR. 2018. The Unreasonable Effectiveness of Convolutional Neural Networks in Population Genetic Inference. bioRxiv.

Fletcher W, Yang Z. 2009. INDELible: a flexible simulator of biological sequence evolution. Molecular biology and evolution, 26:1879-1888.

Fu J. 1975. The Rate of Convergence of Consistent Point Estimators.

Fujimoto MS, Suvorov A, Jensen NO, Clement MJ, Bybee SM. 2016. Detecting false positive sequence homology: a machine learning approach. BMC bioinformatics, 17:101.

Goldman N. 1990. Maximum-Likelihood Inference of Phylogenetic Trees, with Special Reference to a Poisson-Process Model of DNA Substitution and to Parsimony Analyses. Syst Zool, 39:345-361.

Goodfellow I, Bengio Y, Courville A. 2016. Deep Learning. Adapt Comput Mach Le:1-775.

Gouy M, Guindon S, Gascuel O. 2010. SeaView version 4: A multiplatform graphical user interface for sequence alignment and phylogenetic tree building. Molecular biology and evolution, 27:221-224.

Hellmuth M, Wieseke N, Lechner M, Lenhof H-P, Middendorf M, Stadler PF. 2015. Phylogenomics with paralogs. Proceedings of the National Academy of Sciences, 112:2058.

Hohl M, Ragan MA. 2007. Is multiple-sequence alignment required for accurate inference of phylogeny? Syst Biol, 56:206-221.

Huelsenbeck JP, Hillis DM. 1993. Success of Phylogenetic Methods in the Four-Taxon Case. Syst Biol, 42:247-264.

Huelsenbeck JP, Rannala B. 2004. Frequentist Properties of Bayesian Posterior Probabilities of Phylogenetic Trees Under Simple and Complex Substitution Models. Syst Biol, 53:904-913.

Huelsenbeck JP, Ronquist F. 2001. MRBAYES: Bayesian inference of phylogenetic trees. Bioinformatics, 17:754-755.

Ioffe S, Szegedy C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. eprint arXiv:1502.03167:arXiv:1502.03167.

Jukes TH, Cantor CR. 1969. CHAPTER 24 - Evolution of Protein Molecules. In: Munro HN editor. Mammalian Protein Metabolism, Academic Press, p. 21-132.

Kingma DP, Ba J. 2014. Adam: A Method for Stochastic Optimization. eprint arXiv:1412.6980:arXiv:1412.6980.

Kolaczkowski B, Thornton JW. 2009. Long-branch attraction bias and inconsistency in Bayesian phylogenetics. PloS one, 4:e7891.

Koonin EV. 2005. Orthologs, paralogs, and evolutionary genomics. Annual review of genetics, 39:309-338.

Krizhevsky A, Sutskever I, E. Hinton G. 2012. ImageNet Classification with Deep Convolutional Neural Networks.

Lecun Y, Bottou L, Bengio Y, Haffner P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86:2278-2324.

Li S, Pearl DK, Doss H. 2000. Phylogenetic Tree Construction Using Markov Chain Monte Carlo. Journal of the American Statistical Association, 95:493-508.

Maddison WP. 1997. Gene Trees in Species Trees. Syst Biol, 46:523-536.

Maddison WP, Knowles LL. 2006. Inferring Phylogeny Despite Incomplete Lineage Sorting. Syst Biol, 55:21-30.

McTavish EJ, Steel M, Holder MT. 2015. Twisted trees and inconsistency of tree estimation when gaps are treated as missing data – The impact of model mis-specification in distance corrections. Molecular phylogenetics and evolution, 93:289-295.

Mielniczuk J, Tyrcha J. 1993. Consistency of Multilayer Perceptron Regression-Estimators. Neural Networks, 6:1019-1022.

Misof B, Misof K. 2009. A Monte Carlo approach successfully identifies randomness in multiple sequence alignments: a more objective means of data exclusion. Syst Biol, 58:21-34.

Nair V, E. Hinton G. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines Vinod Nair.

Nguyen LT, Schmidt HA, von Haeseler A, Minh BQ. 2015. IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. Molecular biology and evolution, 32:268-274.

Ogden TH, Rosenberg MS. 2007. How should gaps be treated in parsimony? A comparison of approaches using simulation. Molecular phylogenetics and evolution, 42:817-826.

Penny D, Hendy MD, Steel MA. 1992. Progress with methods for constructing evolutionary trees. Trends in ecology & evolution, 7:73-79.

Posada D. 2003. Using MODELTEST and PAUP* to Select a Model of Nucleotide Substitution. Current Protocols in Bioinformatics, 00:6.5.1-6.5.14.

Rannala B, Yang Z. 1996. Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. Journal of molecular evolution, 43:304-311.

Rasmussen MD, Hubisz MJ, Gronau I, Siepel A. 2014. Genome-Wide Inference of Ancestral Recombination Graphs. PLoS genetics, 10:e1004342.

Reaz R, Bayzid MS, Rahman MS. 2014. Accurate phylogenetic tree reconstruction from quartets: a heuristic approach. PloS one, 9:e104008-e104008.

Roch S. 2006. A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3:92-94.

Rokas A, Holland PWH. 2000. Rare genomic changes as a tool for phylogenetics. Trends in ecology & evolution, 15:454-459.

Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Molecular biology and evolution, 4:406-425.

Schrider DR, Kern AD. 2018. Supervised Machine Learning for Population Genetics: A New Paradigm. Trends in genetics : TIG, 34:301-312.

Siddall ME. 1998. Success of Parsimony in the Four-Taxon Case: Long-Branch Repulsion by Likelihood in the Farris Zone. Cladistics, 14:209-220.

Sober E. 1988. Reconstructing the past : parsimony, evolution, and inference. Cambridge, Mass., MIT Press.

Steel M, Linz S, Huson DH, Sanderson MJ. 2013. Identifying a species tree subject to random lateral gene transfer. Journal of theoretical biology, 322:81-93.

Strimmer K, von Haeseler A. 1997. Likelihood-mapping: a simple method to visualize phylogenetic content of a sequence alignment. P Natl Acad Sci USA, 94:6815-6819.

Susko E. 2015. Bayesian long branch attraction bias and corrections. Syst Biol, 64:243-255.

Swofford DL, Waddell PJ, Huelsenbeck JP, Foster PG, Lewis PO, Rogers JS. 2001. Bias in Phylogenetic Estimation and Its Relevance to the Choice between Parsimony and Likelihood Methods. Syst Biol, 50:525-539.

Tarca AL, Carey VJ, Chen XW, Romero R, Draghici S. 2007. Machine learning and its applications to biology. PLoS computational biology, 3:e116.

Tavaré S. 1986. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences. American Mathematical Society: Lectures on Mathematics in the Life Sciences, Amer Mathematical Society, p. 57-86.

Truszkowski J, Goldman N. 2016. Maximum Likelihood Phylogenetic Inference is Consistent on Multiple Sequence Alignments, with or without Gaps. Syst Biol, 65:328-333.

Vialle RA, Tamuri AU, Goldman N. 2018. Alignment Modulates Ancestral Sequence Reconstruction Accuracy. Molecular biology and evolution, 35:1783-1797.

von Haeseler A, Strimmer K. 1996. Quartet Puzzling: A Quartet Maximum-Likelihood Method for Reconstructing Tree Topologies. Molecular biology and evolution, 13:964-964.

Warnow T. 2012. Standard maximum likelihood analyses of alignments with gaps can be statistically inconsistent. PLoS currents, 4:RRN1308.

Warnow T. 2018. Computational phylogenetics : an introduction to designing methods for phylogeny estimation. Cambridge University Press.

Yang Z. 1994. Yang Z.. Estimating the pattern of nucleotide substitution. J Mol Evol 39: 105-111.

Yang Z. 2014. Molecular Evolution: A Statistical Approach. Molecular Evolution: A Statistical Approach:1-492.