Running head: TREE TOPOLOGY ESTIMATION USING CNNs

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

Keywords: supervised machine learning, convolutional neuronal network, phylogenetics

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 on simulated data, 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 has been shown to be NP-hard for at least some formulations of the problem (Roch 2006), and 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 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 (reviewed in 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 "training 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 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 or higher-dimensional tensor 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 on simulated data, 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 the 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. In order to construct a training set consisting of trees with a variety of branch length configurations we sampled uniformly from a "branch-length space" (BL-space) that encompasses a wide array 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 shape 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 with $p_{\text{inv}} = 0.5$, the specified branch lengths refer only to the 50% of sites that are free to vary while the remaining sites have branch lengths of zero. 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

104 identical fixed insertion and deletion rates ($\lambda_I = \lambda_D = 0.01$) and drew indel lengths from a Zipf
105 distribution with $a = 1.5$, similar to empirical estimates (Vialle, et al. 2018), and a maximum
106 indel size of 50. For each topology class we simulated either 5×10^4, 1.5×10^5 or 3×10^5 MSAs for
107 training and 1.5×10^4 MSAs for the validation and test datasets. Additionally, we simulated test
108 sets consisting of 3000 trees from each of thirteen regions of the tree-shape space that may
109 potentially cause biased or more erroneous tree topology inference, including the Farris and
110 Felsenstein zones that have long been known to negatively affect performance of classical
111 methods (Swofford, et al. 2001). The simulation parameters for each of these were the same as
112 those stated above with branch lengths drawn from the distributions shown in **Table 1**.
113 Additionally we simulated trees within so called "twisted" Farris zone (McTavish, et al. 2015)
114 with asymmetrical branch lengths as well as trees from different regions that may potentially
115 cause tree estimation to be less accurate. Throughout this paper we denote sister branches or their
116 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
117 the internal branch as B_5. The overall pipeline is summarized in **Figure 1**.

118 The internal branch length (B_5) for the Farris and Felsenstein zones was drawn from
119 $U(0,0.05)$ (**Table 1**). In order to investigate effect of internal branch length on each method's
120 performance in more detail, we also created variants of the Farris and Felsenstein zones, called
121 "extended" where B_5 was drawn from the interval $U(0,0.5)$. Previous simulation studies suggest
122 that methodological biases primarily arise when an internal branch appears to be very short, i.e.
123 internal branch length ≤ 0.05 (Swofford, et al. 2001). However, since these studies followed
124 overly simplified simulation approaches (e.g. MSA simulation under JC model only; Siddall
125 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 as we do in this study.

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 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).

149 We set the filter size for the first convolution operation to 4×1, reasoning that it would
150 potentially capture a phylogenetic signal by striding across the MSA and examining one column
151 of the alignment at a time. The total number of filters was set to 1024 for the first and second
152 convolutional layer and set to 128 for the remaining convolutional layers. We decided upon 1024
153 filters because this number is greater than the number of possible columns in an MSA with four
154 DNA sequences allowing gap characters, and we had decided (arbitrarily) to limit these values to
155 powers of two. Additional network hyperparameters were chosen somewhat arbitrarily after
156 assessing performance of several different network architectures. We also used average-pooling
157 rather than the more common max-pooling operation given that Flagel et al. found this approach
158 to be effective when applied to population genetic alignments (Flagel, et al. 2018). The first
159 average-pooling operation size was set to 1×1 (i.e. no pooling), then all the subsequent
160 convolution steps had equal sizes of 1×2 and pooling steps had equal sizes of 1×4, 1×4, 1×4,
161 1×2, 1×2 and 1×1. We had also experimented with fewer layers but noticed that "deeper" CNN
162 architectures, i.e. with more convolutional layers had better accuracy. In order to avoid possible
163 internal covariate shift and model overfitting we used batch normalization (Ioffe and Szegedy
164 2015) followed by dropout regularization with a rate of 0.2 for each convolution layer as well as
165 for the hidden layer. During training in most of the cases we used a minibatch size of 150 MSAs.
166 The detailed network architectures along with the training time are available in **Supplementary**
167 **Text**. The training procedure was run for a number of iterations determined by the stopping
168 criterion (improvement in validation loss of the current iteration over the best prior iteration <
169 0.001) and patience value (the number of consecutive iterations that must satisfy the stopping
170 criterion before halting training). We observed that the results of our training runs were fairly
171 stable, with a given CNN architecture consistently achieving the same final accuracy on the

172 validation set. To further examine performance during training, we visualized the surface of the
173 loss function and training trajectories across the CNN's parameter space (i.e. the possible values
174 of all weights in the network) as described below. This revealed that there appear to be many
175 readily reachable local minima of the loss function that all have roughly equivalent values
176 (**Supplementary Fig. S2**), lending further support to the notion that training results should be
177 fairly stable, at least for the datasets and architectures examined here.
178
179 *Testing Procedures*
180 We sought to compare the performance of CNN vs. conventional tree inference methods, namely
181 MP, NJ, ML and BI using our simulated test datasets. MP reconstructions were performed using
182 PHYLIP's dnapars program v3.696 (Felsenstein 1989) in the interface provided by SeaView
183 v4.7 (Gouy, et al. 2010). ML analysis was implemented in IQ-TREE v1.6.5 (Nguyen, et al.
184 2015) restricting model selection to only the models that were used to simulate MSAs, i.e. JC,
185 TIM, TIMef, GTR and UNREST (equivalent to the 12.12 model in IQ-TREE) with +I and/or +Γ,
186 where the continuous rate gamma distribution was approximated by discrete gamma with 8
187 categories (G8). For each MSA, ML search was performed exhaustively evaluation each of the
188 three possible unrooted topologies. The NJ trees were generated using the distance function of
189 the best-fit model in IQ-TREE. Note that ML and NJ approaches infer a rooted tree under the
190 UNREST model, thus every such estimated tree was unrooted for further comparisons. Bayesian
191 tree reconstruction was carried out in ExaBayes v1.4.1 (Aberer, et al. 2014) under default
192 parameters starting with a random initial topology. Three independent MCMC chains were run
193 for 10^6 generations each with two coupled chains. Tree support was assessed by standard
194 nonparametric bootstrap (Felsenstein 1985) for MP, NJ, ML and CNN, and by posterior split

195 probability for BI. We also assessed tree support from the CNN by simply using the estimated
196 posterior class membership probabilities emitted by the output layer. Then, for each method we
197 calculated topological accuracy as the number of correctly returned topologies divided by the
198 total number of inferences performed. See **Supplementary Text** for exact commands that were
199 used to infer topologies.
200

201 **Results and Discussion**

202 *CNNs are more Accurate than Standard Methods on Gapped and Ungapped MSAs within BL-*
203 *space*

204 Initially, we trained our CNN on gapped and ungapped training sets of total 15×10^4 MSAs (each
205 with 1000 sites, 5×10^4 MSAs for each of our three possible topologies) and in each case for our
206 initial evaluation we assessed our accuracy on an independent test set generated in the same
207 manner as our training data. We compared the accuracy of our CNN on simulated test data
208 (1.5×10^4 MSAs per topology) to that of several other standard tree estimation methods, namely
209 maximum parsimony (MP), neighbor joining (NJ), maximum likelihood (ML), and Bayesian
210 inference (BI). Strikingly, the CNN was substantially more accurate than the other methods
211 (McNemar test, $P < 1\times10^{-10}$ for each comparison with the CNN) when applied to gapped MSAs
212 (**Fig. 2**). However, this was not the case when testing methods on ungapped MSAs: here
213 accuracy of the CNN (0.765), which was in this case also trained on ungapped MSAs, was nearly
214 identical to the best performing method, BI (accuracy = 0.765, McNemar test, P = 0.918).
215 Nevertheless, the CNN's accuracy improved with increasing amounts of training data and
216 obtained higher accuracy (0.777; **Fig. 2**) than all other methods when trained on either 1.5×10^5 or
217 3×10^5 ungapped MSAs per topology. The difference with the next best method (BI) was not

218 significant (McNemar test, $P = 0.08$) with 1.5×10^5 training examples but significant (McNemar
219 test, $P = 0.013$) with 3×10^5 training examples. Although the results of this initial evaluation
220 within our BL-space are encouraging, we must bear in mind that accuracy can vary within this
221 space. Moreover, our BL-space was constructed in part to expose CNN to many branch-length
222 configurations including some extreme configurations which one may not expect to commonly
223 encounter in real data sets. Thus in the next section we test each method on a number of more
224 narrowly defined tree branch-length configurations in order to gain a more detailed picture of our
225 method's strengths and weaknesses.

226 We also investigated the relationship between several simulation parameters and
227 inference accuracy. When sequences are very closely related, the MSA may consist largely of
228 invariant sites, which are phylogenetically uninformative. One of the parameters of our
229 simulations was the fraction of sites that were not allowed to vary (p_{inv}; see Methods). We found
230 that for each method, misclassified MSAs from the gapped test set had a larger value of p_{inv} on
231 average (Wilcoxon rank sum test, $P < 1\times10^{-10}$ for each standard method and $P = 6.135\times10^{-5}$ for
232 the CNN), but this difference is far less pronounced for the CNN (**Fig. 3a**). However, in the case
233 of the ungapped test set, MSAs misclassified by all methods including the CNN showed a similar
234 bias toward larger values p_{inv} on average (Wilcoxon rank sum test, $P < 1\times10^{-10}$ for each method;
235 **Fig. 3b**). The gamma shape parameter, α, which governs mutation rate heterogeneity across the
236 MSA does not impact the CNN's ability to infer the correct topology (Wilcoxon rank sum test, P
237 $= 0.91$) whereas for all other methods misclassified MSAs had elevated values of α (Wilcoxon
238 rank sum test, $P < 1\times10^{-5}$ for all standard methods; **Fig. 3c**). Again, in the case of ungapped data
239 each method showed a similar bias, with larger α in misclassified versus correctly classified
240 examples (Wilcoxon rank sum test, $P < 1\times10^{-5}$ for all methods; **Fig. 3d**). Together these results

241 suggest that CNNs are more robust to higher values of p_{inv} and α than traditional methods for
242 gapped MSAs. Additionally, we note that the DNA substitution model does not affect any
243 method's performance (Fisher exact test, $P > 0.05$ for all methods).

244 Overall these results suggest that our CNN can effectively extract phylogenetic signal.
245 This is especially so for the CNN trained on gapped MSAs, which easily outcompeted the other
246 methods, all of which treat gaps as missing data. Despite their potential to provide unambiguous
247 phylogenetic information (Rokas and Holland 2000; Dessimoz and Gil 2010) due to reduced
248 homoplasy relative to substitutions (Ashkenazy, et al. 2014), approaches incorporating indels are
249 controversial and not generally utilized in practice (Ogden and Rosenberg 2007; Warnow 2012).
250 When our CNN is trained on gapped data it learns to detect phylogenetic signal from both indels
251 and substitutions present in an MSA. To further examine this property of the gapped MSA CNN,
252 we simulated MSAs of length 1000 using trees with branch lengths drawn from $U(0,0.5)$ with no
253 substitutions (i.e. $p_{inv} = 1$) allowing only indels. Under these conditions all standard methods fail
254 to achieve consistency (Warnow 2012; Truszkowski and Goldman 2016), and MP, NJ, ML and
255 BI all had accuracy of ~0.33 or lower—no better than random guessing. By contrast, the CNN's
256 accuracy on this set was 0.927. To test a scenario towards the opposite extreme wherein every
257 site is informative, we simulated MSAs of length 1000 allowing all sites to vary ($p_{inv} = 0$) under
258 the JC model, including indels, and using trees with all branch lengths drawn from $U(0,0.5)$.
259 Under these conditions all methods exhibited very high accuracy (MP: 0.962, NJ: 0.976, ML:
260 0.975, BI: 0.978 and CNN: 0.959).

261 These results further suggest that our CNN, when trained on gapped MSAs, is in
262 principle able to incorporate indel information into its phylogenetic inference. Because the CNN
263 implicitly constructs substitution and indel process models by examining the simulated training

264 data, alignment error could impact inference in practice. This could be handled in one of two
265 ways: 1) by incorporating alignment errors into the training set (e.g. by running a sequence
266 aligner on training data prior to training), or 2) judging the reliability of indels within the MSA
267 and masking particular sites or omitting entire MSAs appearing to be unreliable (e.g. following
268 Castresana 2000; Misof and Misof 2009; Ashkenazy, et al. 2014; Fujimoto, et al. 2016). Another
269 potential problem is model misspecification: accuracy could suffer if indel rates/lengths differ
270 dramatically between simulated training data and real datasets. Future work will have to
271 investigate strategies to train a network that is robust to such misspecification, perhaps by
272 incorporating a variety of indel models into training; additional steps to train neural networks to
273 be insensitive to the proclivities of empirical data may also be necessary (discussed in
274 Concluding Remarks below).
275
276 *Topology Estimation Under Branch Length-Heterogeneity*
277 There exist several regions in the branch-length parameter space where common methods fail to
278 produce the correct topology. The bias known as long branch attraction (LBA) is the inability of
279 MP (Felsenstein 1978a) and possibly Bayesian (Kolaczkowski and Thornton 2009; Susko 2015)
280 methods to estimate the correct topology in presence of long nonadjacent branches (i.e. the
281 Felsenstein zone; Huelsenbeck and Hillis 1993); these methods tend to erroneously place such
282 long branches together and thus favor an incorrect topology (**Table 1**; **Figs. 4c and 5c**). Another
283 region in the branch-length space, called the Farris zone (Siddall 1998), causes ML methods to
284 exhibit reduced accuracy on topologies that have adjacent long branches and causes MP to
285 correctly infer trees more often as a result of LBA which in this case biases it toward the correct
286 topology (**Table 1**; **Figs. 4a and 5a**); we note that this bias may diminish when using longer

287 alignments due to ML's property of consistency (Swofford, et al. 2001). Additionally, we
288 assessed the performance of all methods in the asymmetrical ("twisted") Farris zone (**Table 1**;
289 **Figs. 4b and 5b**), where a single incorrect topology is favored by NJ if the corrected distance
290 function is convex (McTavish, et al. 2015). We also examined trees with a short internal branch
291 (≤ 0.05) but external branches lengths varying along the interval [0,0.5] (**Table 1**; **Figs. 4d and**
292 **5d**).

293 We primarily focus on these four regions of known bias (**Figs. 4 and 5**) but also
294 examined several other regions of potential bias on test sets of gapped (**Table 1**; **Supplementary**
295 **Fig. S3**) and ungapped (**Table 1; Supplementary Fig. S4**) MSAs. The accuracy estimates for
296 each region are summarized in **Table 2**. In most of these regions, our CNN exhibited accuracy
297 roughly matching or exceeding that of the best performing traditional method, whereas each
298 other method failed catastrophically in at least one of these regions (**Figs. 4 and 5,**
299 **Supplementary Figs. S3, S4**). The two exceptions where the CNN exhibited markedly reduced
300 accuracy were the Felsenstein zone (**Figs. 4 and 5; Table 2**) and the "extra-short branches"
301 region where all branch lengths range vary uniformly along [0, 0.01] (**Table 2; Supplementary**
302 **Figs. S3 and S4**). Specifically, in the Felsenstein zone the CNN's accuracy is 0.464 (gapped)
303 and 0.461 (ungapped), above the random expectation of 1/3, and well below that of ML (0.723
304 (gapped) and 0.719 (ungapped); **Figs. 4c and 5c**). However, CNN's accuracy could be notably
305 improved by adding additional MSAs simulated from the Felsenstein zone or the "extra-short
306 branches" region: performing additional training iterations on 1000 MSAs per topology from
307 these regions boosted the CNN's accuracy to 0.753 for gapped MSAs and to 0.771 for ungapped
308 MSAs for the Felsenstein zone, and to 0.882 (gapped) and 0.904 (ungapped) for the "extra-short
309 branches" region respectively (**Table 2**). This suggests that it is possible to fix a known "blind

spot" of the CNN by simply simulating additional relevant training data and performing more iterations of the training algorithm. The CNN shows superior accuracy in each of the remaining regions of potential bias (**Supplementary Fig. S3**) for gapped MSAs compared to all other methods. Overall, on ungapped data, the CNN's performance roughly matched that of BI across zones (**Supplementary Fig. S4**), with the exception of the regions mentioned above. These two methods both exhibit acceptable but not exceptional accuracy in each of these different zones (**Table 2**), perhaps with the exception of the Felsenstein zone where all methods other than ML have low accuracy (≤ 0.57 versus 0.719 for ML), unless the CNN is provided with additional training data as described above.

Additionally, we tested performance of all methods on shorter gapped and ungapped MSAs of length 500 within each region specified in **Table 2** (see **Supplementary Table S1** for accuracy estimates). Overall, accuracies were lower for all methods, but with qualitatively similar rankings across methods. Further, we showed that accuracy increases with increasing MSA size for the CNN which is potentially indicative of statistical consistency (**Supplementary Table S2 and Supplementary Text**), though we do not prove this here.

Although we find that overall the CNN performs quite well on the test sets examined above, it is important to stress that it a method's overall utility is not fully captured by a single accuracy estimate from one test set, or even a suite of test sets as we have done here. Each of the test sets we examined is a subspace of the BL-space that we defined in the **Supplementary Text**, and there are an infinite number of such sets that one could examine. Moreover, partitioning this space into a number of equally sized sets small enough to comprehensively assess how performance varies across the entire BL-space is a monumental task. For example, if one defined a BL-space and discretized it into 1% increments for all five branches, one would need to

333 simulate 100^5 datasets with all possible branch length combinations in order to traverse the entire
334 parameter space. The unfeasibility of such an experiment was previously acknowledged by
335 Huelsenbeck (1995), thus we cannot rule out the possibility that our CNN's performance is
336 compromised in certain regions of parameter space especially if examples from such regions
337 were underrepresented during training phase. Indeed this appears be the cause of our initially
338 poor performance on the Felsenstein and the "extra-short branches" zones, which was rectified
339 through targeted additional training.
340
341 *Analysis of Phylogenetic Reliability Measures*
342 Testing phylogenetic hypotheses usually involves calculation of some reliability measures such
343 as bootstrap support (BS) for MP, NJ and ML or posterior probabilities (PP) in the context of BI.
344 Despite the fact that their direct comparison is somewhat problematic due to their fundamental
345 differences, one can ask which measures are more prone to support false tree topologies
346 (Douady, et al. 2003; Huelsenbeck and Rannala 2004). For our CNN we used two confidence
347 measures: bootstrap support for the chosen topology (obtained by applying the trained CNN to
348 each MSA in a set of bootstrap replicates), and the class membership posterior probability (CPP)
349 estimates produced automatically by the CNN which can also be interpreted as a measure of
350 topological reliability. We find that high support (BS > 0.95, or PP > 0.95 in the case of BI) is
351 occasionally assigned to incorrectly classified trees by several methods: 0.052 of
352 misclassifications by MP, 0.05 by NJ, 0.003 by ML, 0.042 by BI, and 0.038 by the CNN for
353 gapped MSAs (**Fig. 6a**) and 0.056 by MP, 0.054 by NJ, 0.0036 by ML, 0.048 by BI, and 0.034
354 by the CNN for ungapped MSAs (**Fig. 6b**). Thus, a very high reliability measure cutoff may be
355 necessary in order to filter out positively misleading cases. On the other hand, the CPP estimates

356 show a larger disparity between scores assigned to correctly vs. incorrectly classified topologies
357 than do other measures (**Fig. 6**), and only 0.0004 and 0.0038 of incorrectly classified gapped and
358 ungapped MSAs respectively having a CPP above 0.95. In other words, the CNN can avoid
359 incorrect classifications even when imposing a relatively relaxed posterior probability cutoff, and
360 thus simultaneously recover a large fraction of correctly classified topologies while producing
361 very few misclassifications. This is further illustrated by precision-recall curves
362 (**Supplementary Fig. S5**), which show that compared to other methods the CNN achieves high
363 sensitivity while retaining a high positive predictive value on gapped alignments. Moreover, for
364 our four-taxon topology trees the interpretation of CPP is straightforward: the most unreliable
365 tree estimates receive CPP of ~1/3, which is equivalent to randomly picking one of the three
366 possible topologies.
367
368 *Concluding Remarks*
369 We used a convolutional neural network to infer unrooted quartet topologies through
370 classification. Our results demonstrate that CNNs perform this task quite well and have a number
371 of desirable qualities. First and foremost, under many scenarios our CNN resolves quartets with
372 higher accuracy than existing methods (**Table 2**). The CNN is also relatively insensitive to
373 regions of the tree branch-length heterogeneity (e.g. the Felsenstein and Farris zones) that are
374 known to confound certain methods, perhaps because the training set included trees with a
375 variety of branch-length configurations (**see Supplementary Text**). Another advantage of using
376 CNNs is that they produce posterior probability estimates for each tree topology; these posterior
377 probabilities are estimated during the classification process and thus require no additional
378 computation. Also, we have shown that our CNN's accuracy can be further improved by adding

379 more training datasets, albeit most likely with diminishing returns (**Fig. 2**). Although training on
380 large datasets can impose a computational burden up front, this training only needs to be
381 completed once, after which the CNN can be used to infer tree topologies very rapidly: the CNN
382 can infer the topologies of 3000 MSAs of length 1000 in ~5 seconds using an NVIDIA Tesla
383 V100-SXM2 GPU). Finally, CNNs have the advantage of being able to incorporate indels into
384 their classification in a straightforward manner—in principle one need only include them in the
385 simulated training data—which can improve accuracy by supplying more information to the
386 classifier. However, future efforts will have to examine the robustness of inference to indel
387 model misspecification, and the extent to which robustness can be improved by training under a
388 mixture of indel rates and length distributions. More generally, this problem underscores the
389 importance of considering ways in which empirical data may not match simulated data, and the
390 consequences of such mismatches for inference. This problem is particularly challenging, as one
391 may not always be able to predict the manner in which empirical data may differ substantially
392 from model-based simulations, but could potentially be addressed by incorporating empirical
393 data as well into the training process. For example, the CNN could be trained on simulated
394 MSAs, and then a data set consisting of real MSAs for which there is strong consensus around
395 the true topology can be supplied to the CNN for additional training iterations. Thus, through this
396 transfer learning procedure (Pan and Yang 2010), the CNN would first learn to accurately infer
397 trees from simulated MSA, and then be tuned to recognize some of the idiosyncrasies of real
398 MSAs.

399 Our efforts thus suggest that there is great potential for the application of deep learning
400 techniques to inferring trees, which is central to questions in both phylogenetics (species-tree
401 inference) and population genomics (inferring ancestral recombination graphs along a

402 recombining chromosome (Rasmussen, et al. 2014). Indeed, using the entire MSA as input
403 would be most useful on recombining sequences—otherwise it is possible to collapse the MSA
404 down to a vector of statistics (e.g. the number of occurrences of each distinct possible MSA
405 column, if the number of sequences is small) that would sufficiently capture all of the relevant
406 information from the input. Deep learning methods acting directly on the MSA could also be
407 applied to additional phylogenetic problems, such as substitution model selection or tree
408 inference from protein sequence data.

409 Although our findings are encouraging, additional advances are required for our approach
410 to match the practical utility of conventional methods. First, we have only investigated the ability
411 of a CNN to produce the correct tree topology, but not the associated branch lengths. Artificial
412 neural networks are capable of estimating real-valued parameters in addition to discrete class
413 labels, so future work should investigate the possibility of using deep learning for inferring
414 branch lengths in addition to the topology. Another limitation of our current approach is that it is
415 limited to a small number of sequences/taxa. Future extensions could use quartet amalgamation
416 methods (e.g. von Haeseler and Strimmer 1996; Reaz, et al. 2014). Although such methods have
417 not matched the accuracy of standard approaches such as ML in the past, because our method is
418 able to resolve quartet topologies with greater accuracy and better calibrated confidence scores, it
419 may be worth investigating whether quartet amalgamation combined via our CNN could
420 compete with standard methods when applied to larger numbers of sequences. Alternatively, it
421 may be possible to devise neural networks capable of directly predicting a tree topology. The
422 latter could in principle be accomplished by treating the values of the final hidden layer of the
423 network as entries in a distance matrix and using an output layer that builds a tree from this
424 matrix (e.g. using NJ), and adopting a tree dissimilarity measure such as RF distance as the

425 objective function to be minimized. However, this approach would also have a number of
426 practical challenges that would have to be overcome (e.g. how to train networks that can handle
427 different numbers of input sequences). One possible solution to this problem is to train a number
428 of CNNs each designed to handle a fixed number of sequences, and then when given an input
429 MSA select the CNN corresponding to the closest number of sequences greater than or equal to
430 that of the input. The size of then input can then be increased by adding duplicates of one row of
431 the MSA until the desired number of sequences is reached. After inferring the topology for this
432 input, the duplicate nodes can be pruned from the tree to produce the final topology.

433 Finally, in order to move from inferring gene trees to inferring species trees, our approach
434 will have to be extended to handle tree discordance due to variation in evolutionary rates across
435 loci, incomplete lineage sorting (Maddison and Knowles 2006; Degnan and Rosenberg 2009)
436 and gene flow (Maddison 1997). Supervised learning approaches may prove to be well suited for
437 this question, as arbitrarily complicated phylogenetic models combining multiple sources of
438 discordance can readily be incorporated into the training process via simulation. Thus, while
439 additional practical developments are necessary, applications of deep learning to problems in tree
440 inference and phylogenomics are likely to be a fruitful endeavor.

441

442 **Supplementary Material**

443 Additional supplementary material is available at Dryad (DOI: XXX); R and Python scripts can
444 be found at GitHub (https://github.com/SchriderLab/Tree_learning), also archived on Zenodo
445 (DOI: 10.5281/zenodo.3246669). Training, validation and test MSA data sets are deposited on
446 Figshare (https://doi.org/10.6084/m9.figshare.8279618.v2).

447

Funding

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

Acknowledgements

We thank Matthew Hahn for valuable feedback on the manuscript, and Seth Bybee and Ariella Gladstein for helpful suggestions. We also thank Rob Lanfear and one anonymous reviewer for their comments.

Literature Cited

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.

471 Degnan JH, Rosenberg NA. 2009. Gene tree discordance, phylogenetic inference and the
472 multispecies coalescent. Trends in ecology & evolution, 24:332-340.
473 Dessimoz C, Gil M. 2010. Phylogenetic assessment of alignments reveals neglected tree signal in
474 gaps. Genome biology, 11:R37.
475 Douady CJ, Delsuc F, Boucher Y, Doolittle WF, Douzery EJP. 2003. Comparison of Bayesian
476 and Maximum Likelihood Bootstrap Measures of Phylogenetic Reliability. Molecular
477 biology and evolution, 20:248-254.
478 Farris JS. 1970. Methods for Computing Wagner Trees. Syst Zool, 19:83-92.
479 Felsenstein J. 1978a. Cases in Which Parsimony or Compatibility Methods Will Be Positively
480 Misleading. Syst Zool, 27:401-410.
481 Felsenstein J. 1978b. The Number of Evolutionary Trees. Syst Zool, 27:27-33.
482 Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach.
483 Journal of molecular evolution, 17:368-376.
484 Felsenstein J. 1985. Confidence-Limits on Phylogenies - an Approach Using the Bootstrap.
485 Evolution; international journal of organic evolution, 39:783-791.
486 Felsenstein J. 1989. PHYLIP - Phylogeny Inference Package (Version 3.2). Cladistics, 5:164-
487 166.
488 Fitch WM. 1971. Toward Defining the Course of Evolution: Minimum Change for a Specific
489 Tree Topology. Syst Zool, 20:406-416.
490 Flagel L, Brandvain Y, Schrider DR. 2018. The Unreasonable Effectiveness of Convolutional
491 Neural Networks in Population Genetic Inference. Molecular biology and evolution, 36:220-
492 238.

493 Fletcher W, Yang Z. 2009. INDELible: a flexible simulator of biological sequence evolution.
494 Molecular biology and evolution, 26:1879-1888.
495 Fujimoto MS, Suvorov A, Jensen NO, Clement MJ, Bybee SM. 2016. Detecting false positive
496 sequence homology: a machine learning approach. BMC bioinformatics, 17:101.
497 Goodfellow I, Bengio Y, Courville A. 2016. Deep Learning. Adapt Comput Mach Le:1-775.
498 Gouy M, Guindon S, Gascuel O. 2010. SeaView version 4: A multiplatform graphical user
499 interface for sequence alignment and phylogenetic tree building. Molecular biology and
500 evolution, 27:221-224.
501 Hellmuth M, Wieseke N, Lechner M, Lenhof H-P, Middendorf M, Stadler PF. 2015.
502 Phylogenomics with paralogs. Proceedings of the National Academy of Sciences, 112:2058.
503 Hohl M, Ragan MA. 2007. Is multiple-sequence alignment required for accurate inference of
504 phylogeny? Syst Biol, 56:206-221.
505 Huelsenbeck JP. 1995. Performance of Phylogenetic Methods in Simulation. Syst Biol, 44:17-48.
506 Huelsenbeck JP, Hillis DM. 1993. Success of Phylogenetic Methods in the Four-Taxon Case.
507 Syst Biol, 42:247-264.
508 Huelsenbeck JP, Rannala B. 2004. Frequentist Properties of Bayesian Posterior Probabilities of
509 Phylogenetic Trees Under Simple and Complex Substitution Models. Syst Biol, 53:904-913.
510 Huelsenbeck JP, Ronquist F. 2001. MRBAYES: Bayesian inference of phylogenetic trees.
511 Bioinformatics, 17:754-755.
512 Ioffe S, Szegedy C. 2015. Batch Normalization: Accelerating Deep Network Training by
513 Reducing Internal Covariate Shift. eprint arXiv:1502.03167:arXiv:1502.03167.
514 Jukes TH, Cantor CR. 1969. CHAPTER 24 - Evolution of Protein Molecules. In: Munro HN
515 editor. Mammalian Protein Metabolism, Academic Press, p. 21-132.

516 Kingma DP, Ba J. 2014. Adam: A Method for Stochastic Optimization. eprint
517 arXiv:1412.6980:arXiv:1412.6980.
518 Kolaczkowski B, Thornton JW. 2009. Long-branch attraction bias and inconsistency in Bayesian
519 phylogenetics. PloS one, 4:e7891.
520 Koonin EV. 2005. Orthologs, paralogs, and evolutionary genomics. Annual review of genetics,
521 39:309-338.
522 Krizhevsky A, Sutskever I, E. Hinton G. 2012. ImageNet Classification with Deep
523 Convolutional Neural Networks.
524 Lecun Y, Bottou L, Bengio Y, Haffner P. 1998. Gradient-based learning applied to document
525 recognition. Proceedings of the IEEE, 86:2278-2324.
526 Li S, Pearl DK, Doss H. 2000. Phylogenetic Tree Construction Using Markov Chain Monte
527 Carlo. Journal of the American Statistical Association, 95:493-508.
528 Maddison WP. 1997. Gene Trees in Species Trees. Syst Biol, 46:523-536.
529 Maddison WP, Knowles LL. 2006. Inferring Phylogeny Despite Incomplete Lineage Sorting.
530 Syst Biol, 55:21-30.
531 McTavish EJ, Steel M, Holder MT. 2015. Twisted trees and inconsistency of tree estimation
532 when gaps are treated as missing data – The impact of model mis-specification in distance
533 corrections. Molecular phylogenetics and evolution, 93:289-295.
534 Misof B, Misof K. 2009. A Monte Carlo approach successfully identifies randomness in multiple
535 sequence alignments: a more objective means of data exclusion. Syst Biol, 58:21-34.
536 Nair V, E. Hinton G. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines
537 Vinod Nair.

538 Nguyen LT, Schmidt HA, von Haeseler A, Minh BQ. 2015. IQ-TREE: a fast and effective
539 stochastic algorithm for estimating maximum-likelihood phylogenies. Molecular biology
540 and evolution, 32:268-274.
541 Ogden TH, Rosenberg MS. 2007. How should gaps be treated in parsimony? A comparison of
542 approaches using simulation. Molecular phylogenetics and evolution, 42:817-826.
543 Pan SJ, Yang Q. 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and
544 Data Engineering, 22:1345-1359.
545 Posada D. 2003. Using MODELTEST and PAUP* to Select a Model of Nucleotide Substitution.
546 Current Protocols in Bioinformatics, 00:6.5.1-6.5.14.
547 Rannala B, Yang Z. 1996. Probability distribution of molecular evolutionary trees: a new method
548 of phylogenetic inference. Journal of molecular evolution, 43:304-311.
549 Rasmussen MD, Hubisz MJ, Gronau I, Siepel A. 2014. Genome-Wide Inference of Ancestral
550 Recombination Graphs. PLoS genetics, 10:e1004342.
551 Reaz R, Bayzid MS, Rahman MS. 2014. Accurate phylogenetic tree reconstruction from
552 quartets: a heuristic approach. PloS one, 9:e104008-e104008.
553 Roch S. 2006. A short proof that phylogenetic tree reconstruction by maximum likelihood is
554 hard. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3:92-94.
555 Rokas A, Holland PWH. 2000. Rare genomic changes as a tool for phylogenetics. Trends in
556 ecology & evolution, 15:454-459.
557 Saitou N, Nei M. 1987. The neighbor-joining method: a new method for reconstructing
558 phylogenetic trees. Molecular biology and evolution, 4:406-425.
559 Schrider DR, Kern AD. 2018. Supervised Machine Learning for Population Genetics: A New
560 Paradigm. Trends in genetics : TIG, 34:301-312.

561 Siddall ME. 1998. Success of Parsimony in the Four-Taxon Case: Long-Branch Repulsion by
562 Likelihood in the Farris Zone. Cladistics, 14:209-220.
563 Susko E. 2015. Bayesian long branch attraction bias and corrections. Syst Biol, 64:243-255.
564 Swofford DL, Waddell PJ, Huelsenbeck JP, Foster PG, Lewis PO, Rogers JS. 2001. Bias in
565 Phylogenetic Estimation and Its Relevance to the Choice between Parsimony and
566 Likelihood Methods. Syst Biol, 50:525-539.
567 Tarca AL, Carey VJ, Chen XW, Romero R, Draghici S. 2007. Machine learning and its
568 applications to biology. PLoS computational biology, 3:e116.
569 Tavaré S. 1986. Some Probabilistic and Statistical Problems in the Analysis of DNA Sequences.
570 American Mathematical Society: Lectures on Mathematics in the Life Sciences, Amer
571 Mathematical Society, p. 57-86.
572 Truszkowski J, Goldman N. 2016. Maximum Likelihood Phylogenetic Inference is Consistent on
573 Multiple Sequence Alignments, with or without Gaps. Syst Biol, 65:328-333.
574 Vialle RA, Tamuri AU, Goldman N. 2018. Alignment Modulates Ancestral Sequence
575 Reconstruction Accuracy. Molecular biology and evolution, 35:1783-1797.
576 von Haeseler A, Strimmer K. 1996. Quartet Puzzling: A Quartet Maximum-Likelihood Method
577 for Reconstructing Tree Topologies. Molecular biology and evolution, 13:964-964.
578 Warnow T. 2012. Standard maximum likelihood analyses of alignments with gaps can be
579 statistically inconsistent. PLoS currents, 4:RRN1308.
580 Yang Z. 1994. Yang Z.. Estimating the pattern of nucleotide substitution. J Mol Evol 39: 105-
581 111.
582

Figures

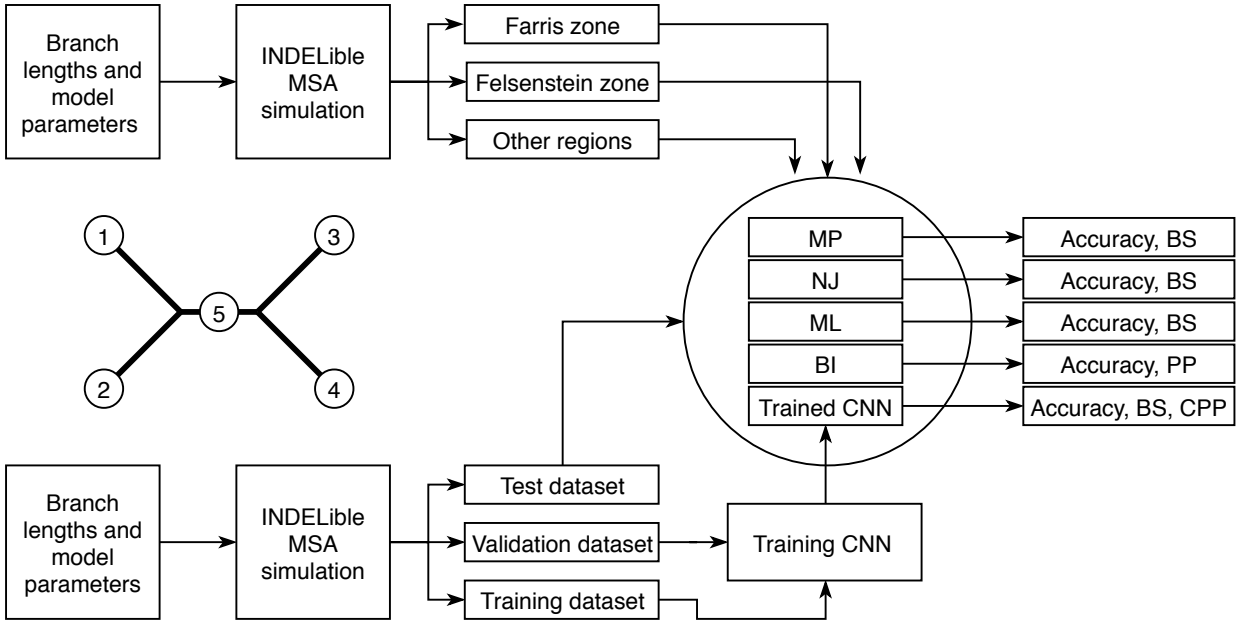

Figure 1: Summary of the simulation and testing procedures. The top branch of the diagram represents the steps that were used to simulate datasets with various branch length configurations (**Table 1**) to test each method's performance. The bottom branch of the diagram represents the steps that were used to simulate training and validation datasets to train the CNN, and the initial 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.

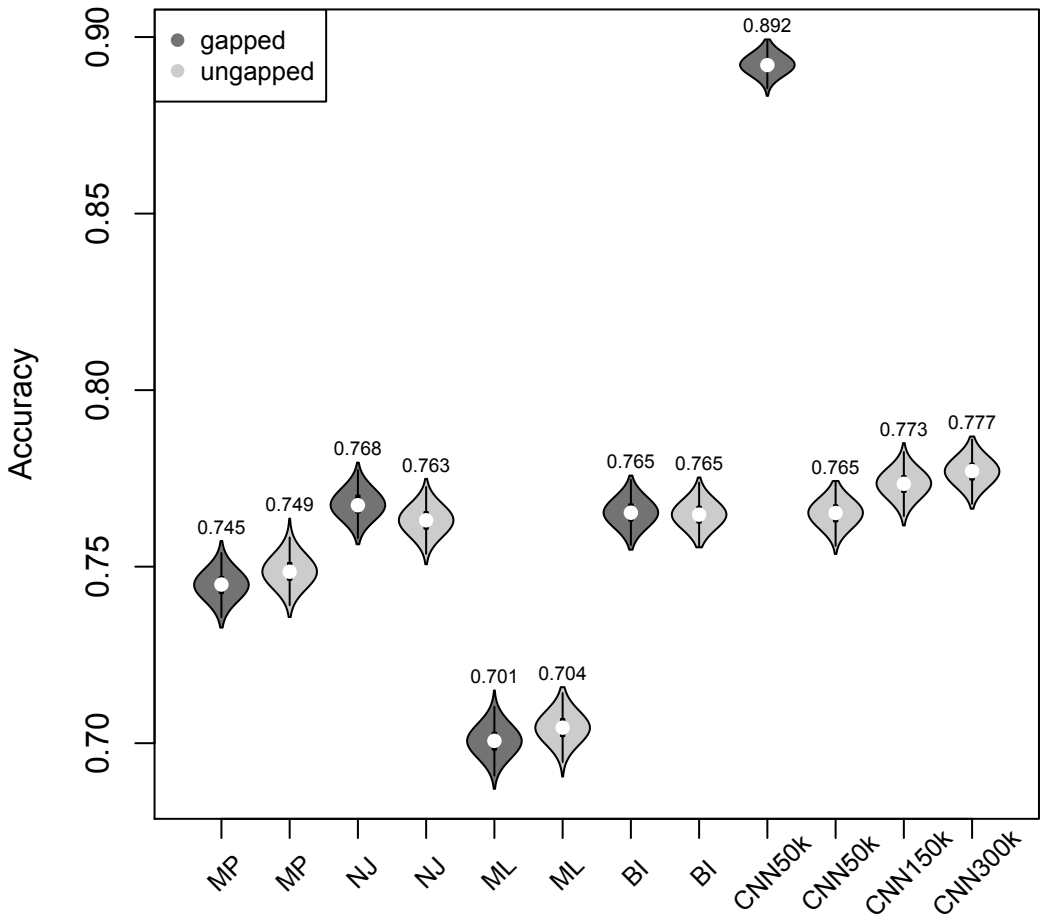

Figure 2: Accuracy of tree topology inference within the test set. The violin plots show the accuracy for different tree estimation methods achieved on gapped and ungapped (15000 MSAs each) datasets. The distribution shown in each violin plot is obtained by bootstrap resampling from the test dataset 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. The number after CNN (i.e. 50k, 150k or 300k) indicates number of MSAs per topology that were used to in training. MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network.

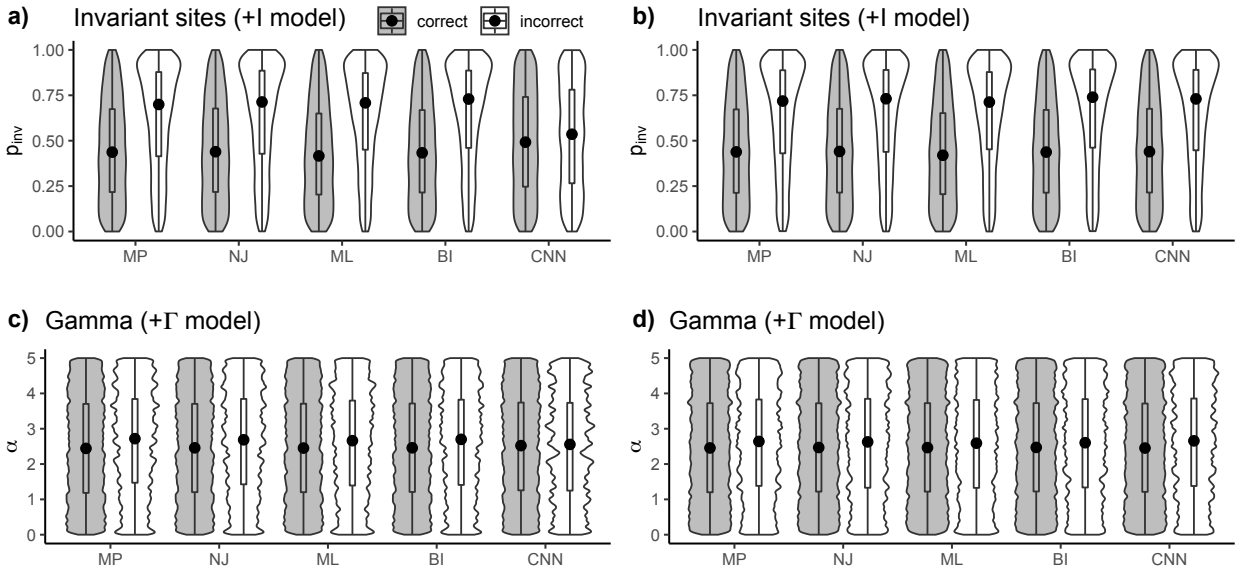

Figure 3: Differences in parameter values between correct and incorrect classifications for each method. The distribution of the parameter governing the fraction of invariable sites (p_{inv}) for correctly and incorrectly inferred trees for a) gapped and b) ungapped MSAs. The distribution of gamma rate parameter (α) for correctly and incorrectly inferred trees for c) gapped and d) ungapped MSAs. The black dots of each violin plot indicate median of the distribution. These results were obtained using 15000 simulated MSAs from BL-space. MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network.

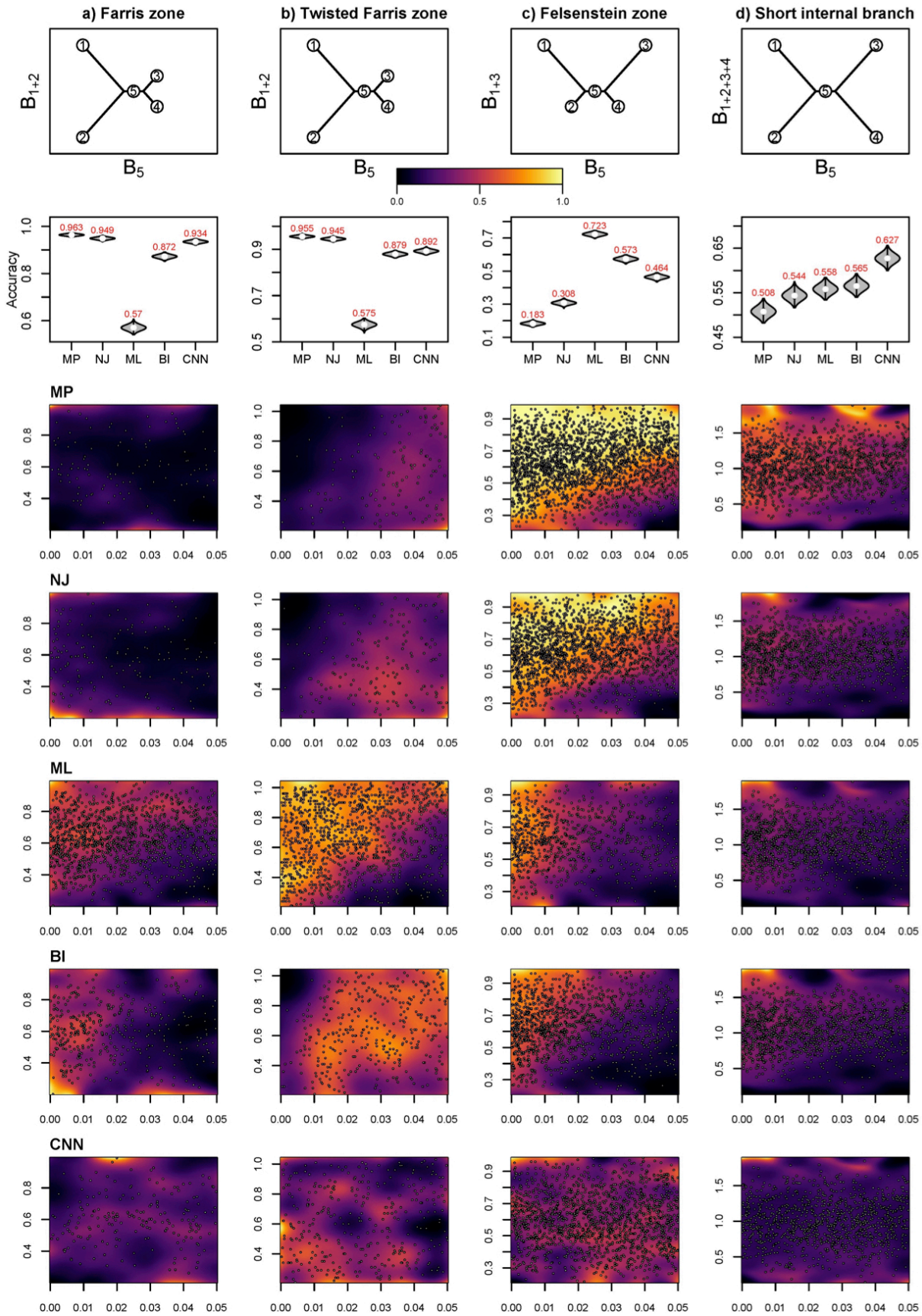
a) Farris zone
b) Twisted Farris zone
c) Felsenstein zone
d) Short internal branch
B1+2
B1+3
B1+2+3+4
B5
Accuracy
0.963
0.949
0.872
0.934
0.57
0.955
0.945
0.879
0.892
0.575
0.183
0.308
0.723
0.573
0.464
0.508
0.544
0.558
0.565
0.627
MP
NJ
ML
BI
CNN

Figure 4: Accuracy within regions with elevated branch-length heterogeneity for gapped MSAs. The four columns summarize results in a) the Farris zone, b) the twisted Farris zone, c) the Felsenstein zone and d) 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 (gray 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. The color scale shows the estimated density of the incorrectly inferred topologies at each point in the branch-length parameter space, with brighter colors representing a greater density of errors. MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network.

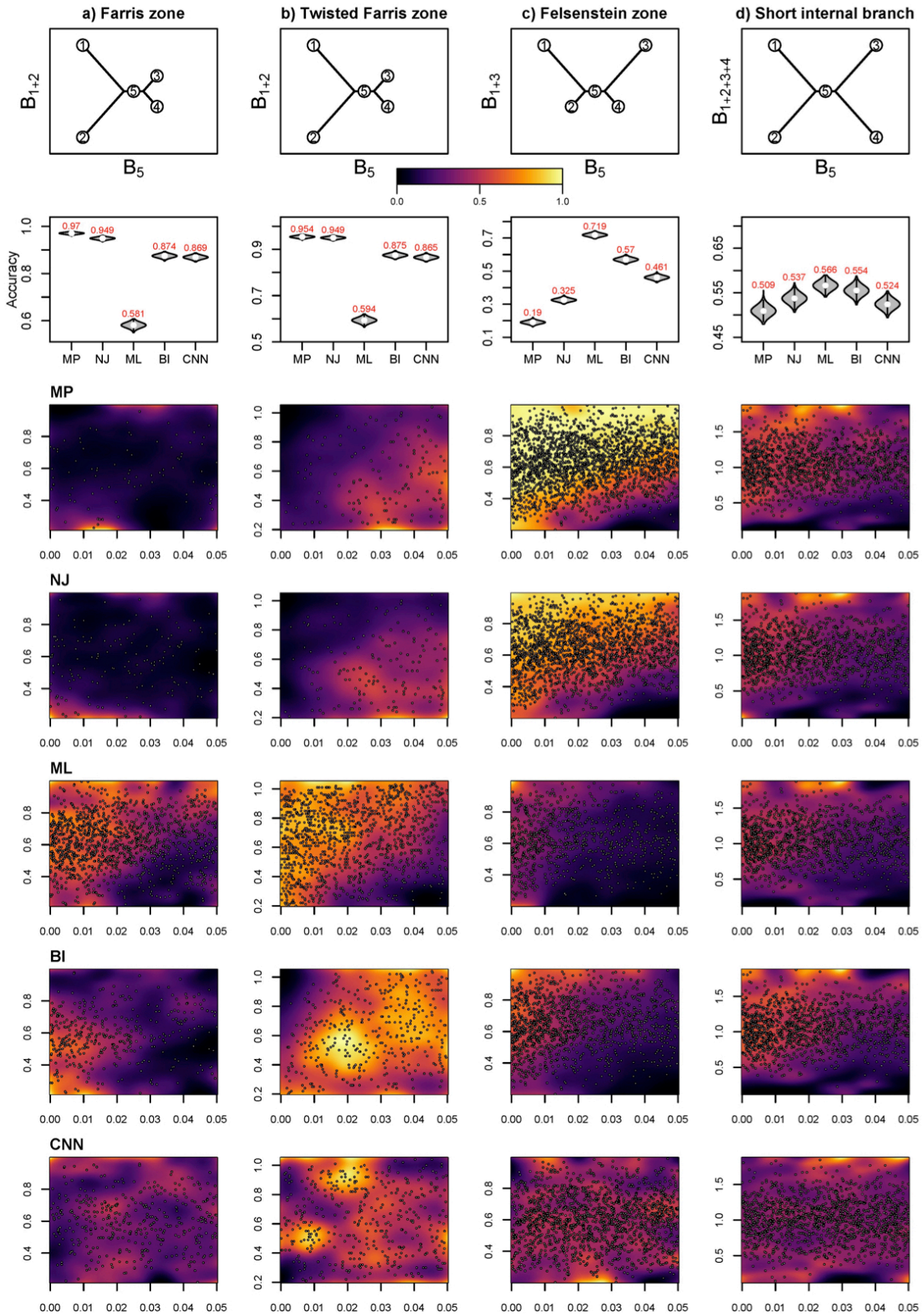
a) Farris zone
b) Twisted Farris zone
c) Felsenstein zone
d) Short internal branch
B1+2
B1+3
B1+2+3+4
B5
Accuracy
0.97
0.949
0.874
0.869
0.581
0.954
0.949
0.875
0.865
0.594
0.719
0.57
0.461
0.325
0.19
0.509
0.537
0.566
0.554
0.524
MP
NJ
ML
BI
CNN

Figure 5: Accuracy within regions with elevated branch-length heterogeneity for ungapped MSAs. The four columns summarize results in a) the Farris zone, b) the twisted Farris zone, c) the Felsenstein zone and d) 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 (gray 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. The color scale shows the estimated density of the incorrectly inferred topologies at each point in the branch-length parameter space, with brighter colors representing a greater density of errors. MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network.

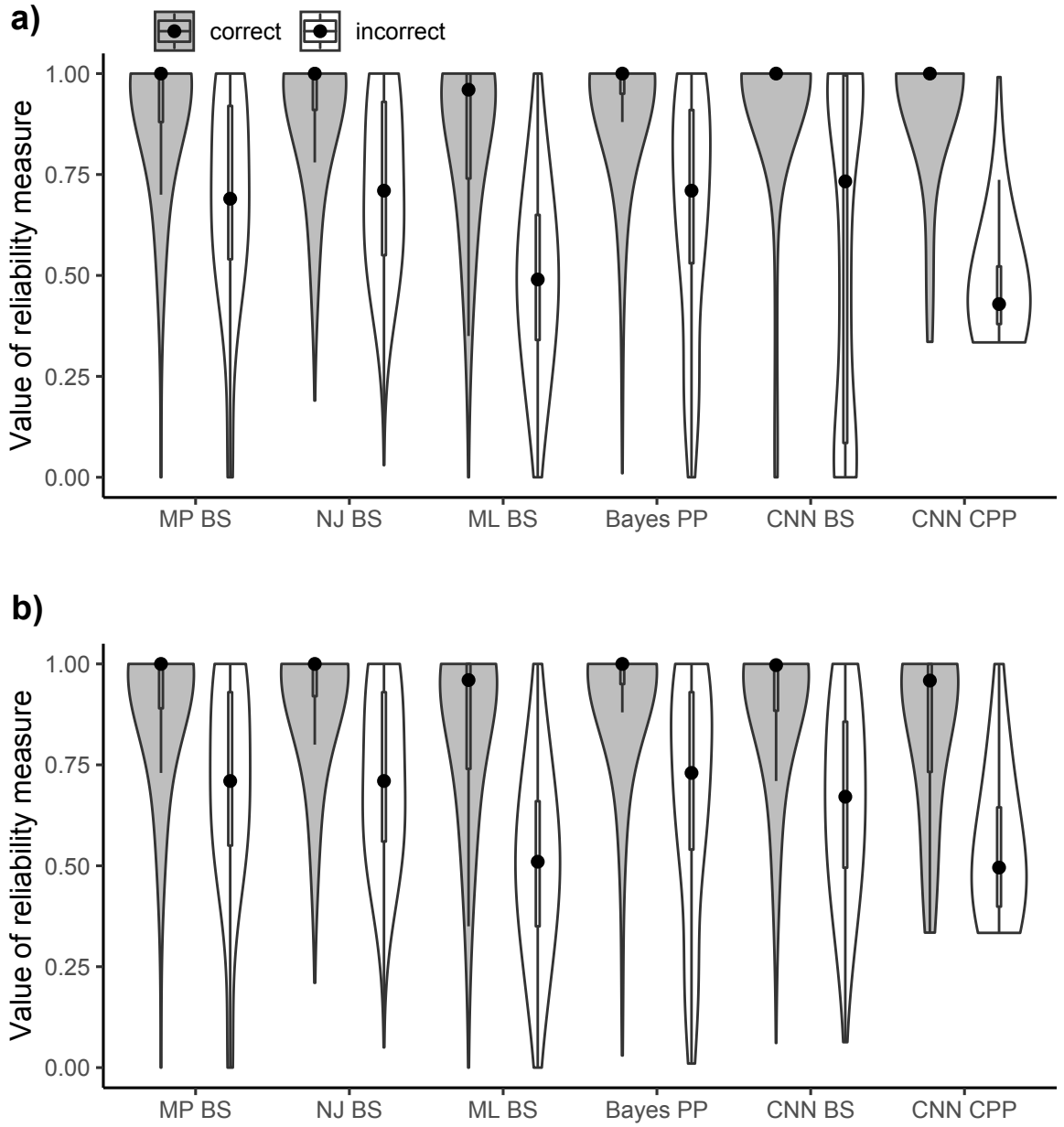

Figure 6: Comparisons of reliability measures among tree inference methods on gapped MSAs. The violin plots represent distribution of reliability measures (BS = bootstrap, PP = Bayesian posterior probability, CPP = CNN topology posterior probability) for correctly and incorrectly inferred topologies. MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network.

Tables

Table 1: Branch-length distributions for test sets designed to assess accuracy and 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 the internal branch. *TExp* denotes truncated exponential distribution. *Beta* denotes beta distribution. A mixture of beta components with equal weights (i.e. $w_1 = w_2 = w_3 = 1/3$) was used to generate the space from which samples were drawn as described in the **Supplementary Text**. *U* denotes uniform distribution. B_1+B_5 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
BL-space	Sampled from w_1*Beta*(α=β=0.1)+w_2*Beta*(α=β=0.5)+w_3*Beta*(α=β=1) as described in **Supplementary Text**				
Truncated exponential	All *TExp*(10,0,0.5)				
Farris zone	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.05)
Twisted Farris zone	*U*(0.1,0.5)	B_1+B_5	B_5	$2B_5$	*U*(0,0.05)
"Extended" Farris zone	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.5)
Felsenstein zone	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0,0.05)
"Extended" Felsenstein zone	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0,0.5)
Long branches	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0,0.5)
Extra-long branches	*U*(0.5,1)	*U*(0.5,1)	*U*(0.5,1)	*U*(0.5,1)	*U*(0,1)
Single long branch	*U*(0.1,0.5)	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.5)
Short branches	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.05)	*U*(0,0.5)
Extra-short branches	All *U*(0,0.01)				
Single short branch	*U*(0,0.05)	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0.1,0.5)	*U*(0,0.5)
Short internal branch	*U*(0, 0.5)	*U*(0, 0.5)	*U*(0, 0.5)	*U*(0, 0.5)	*U*(0,0.05)

Table 2: Performance accuracy of different topology estimation methods on simulated MSAs of length 1000. Accuracy estimates obtained from 15000 (5000 per unrooted topology) sampled from BL-space and from several additional test sets each consisting of 3000 MSAs (1000 per unrooted topology). The accuracy of CNN can be substantially improved by providing additional MSAs for training simulated from specific regions. We performed this additional training for the Felsenstein and extra-short branches regions and show the CNN's improved accuracy in parentheses next to the original accuracy. MP=maximum parsimony, NJ=neighbor-joining, ML=maximum likelihood, BI=Bayesian inference, CNN=convolutional neuronal network.

Region	Accuracy									
	Gapped					Ungapped				
	MP	NJ	ML	BI	CNN	MP	NJ	ML	BI	CNN
BL-space	0.75	0.77	0.7	0.77	0.89	0.75	0.76	0.7	0.77	0.77
Truncated exponential	0.86	0.88	0.88	0.89	0.89	0.85	0.87	0.88	0.88	0.86
Farris zone	0.96	0.95	0.57	0.87	0.93	0.97	0.95	0.58	0.87	0.87
Twisted Farris zone	0.96	0.95	0.58	0.88	0.89	0.95	0.95	0.59	0.88	0.87
"Extended" Farris zone	0.97	0.97	0.85	0.96	0.99	0.97	0.97	0.85	0.96	0.96
Felsenstein zone	0.18	0.31	0.72	0.57	0.46 (0.75)	0.19	0.33	0.72	0.57	0.46 (0.77)
"Extended" Felsenstein zone	0.8	0.83	0.89	0.86	0.92	0.8	0.83	0.9	0.88	0.85
Long branches	0.8	0.83	0.82	0.81	0.95	0.81	0.83	0.82	0.83	0.82
Extra-long branches	0.66	0.68	0.66	0.67	0.96	0.69	0.71	0.69	0.7	0.69
Single long branch	0.96	0.97	0.92	0.97	0.98	0.96	0.96	0.92	0.96	0.95
Short branches	0.98	0.99	0.98	0.98	0.99	0.99	0.99	0.98	0.99	0.99
Extra-short branches	0.89	0.91	0.9	0.9	0.74 (0.88)	0.89	0.91	0.9	0.9	0.86 (0.9)
Single short branch	0.85	0.85	0.83	0.86	0.95	0.85	0.86	0.84	0.86	0.85
Short internal branch	0.51	0.54	0.56	0.57	0.63	0.51	0.54	0.57	0.55	0.52