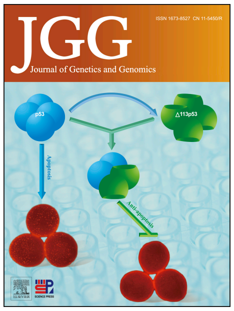


A critical evaluation of deep-learning based phylogenetic inference programs using simulated data sets

Yixiao Zhu, Yonglin Li, Chuhao Li, Xing-Xing Shen, Xiaofan Zhou

PII: S1673-8527(25)00021-9

DOI: https://doi.org/10.1016/j.jgg.2025.01.006

Reference: JGG 1457

To appear in: *Journal of Genetics and Genomics*

Received Date: 2 November 2024

Revised Date: 8 January 2025

Accepted Date: 9 January 2025

A critical evaluation of deep-learning based phylogenetic inference programs using simulated data sets

KEYWORDS

molecular phylogenetics, deep learning, neural network, maximum likelihood

Main text

Inferring phylogenetic trees from molecular sequences is a cornerstone of evolutionary biology. Many standard phylogenetic methods (such as maximum-likelihood) rely on explicit models of sequence evolution and thus often suffer from model misspecification or inadequacy. The on-rising deep learning (DL) techniques offer a powerful alternative. Deep learning employs multi-layered artificial neural networks to progressively transform input data into more abstract and complex representations. DL methods can autonomously uncover meaningful patterns from data, thereby bypassing potential biases introduced by predefined features (Franklin, 2005; Murphy, 2012). Recent efforts have aimed to apply deep neural networks (DNNs) to phylogenetics, with a growing number of applications in tree reconstruction (Suvorov et al., 2020; Zou et al., 2020; Nesterenko et al., 2022; Smith and Hahn, 2023; Wang et al., 2023;), substitution model selection (Abadi et al., 2020; Burgstaller-Muehlbacher et al., 2023) and diversification rate inference (Voznica et al., 2022; Lajaaiti et al., 2023; Lambert et al., 2023). In phylogenetic tree reconstruction, PhyDL (Zou et al., 2020) and Tree_learning (Suvorov et al., 2020) are two notable DNN-based programs designed to infer unrooted quartet trees directly from alignments of four amino acid (AA) and DNA sequences, respectively. These two DNN programs offer pre-built models for immediate analysis and the flexibility to train new models on user-defined data sets, with benchmark tests showing performance comparable to or exceeding traditional phylogenetic methods. However, DNNs encounter challenges as well. It is well known that the effectiveness of a machine-learning algorithm heavily depends on the input-data representation (Alzubaidi et al., 2021). Both PhyDL and Tree_learning are supervised learning methods that need to be trained; however, in molecular phylogenetics, simulation under explicit models of sequence evolution is the only realistic source of training data. Therefore, while DNNs can outperform traditional phylogenetic methods on benchmarks primarily consisting of simulated data (Leuchtenberger et al., 2020), their performance might be compromised on biological data, highlighting the need to understand the robustness of DL-based phylogenetic methods when applied to out-of-distribution data. A recent study suggests that DNNs struggle to match existing methods on datasets with branch-length and sequence-length settings that differ significantly from those in the DNN training data (Zaharias et al., 2022). In this study, we critically evaluated PhyDL and Tree_learning using simulated data, highlighting critical constraints in current deep learning applications in molecular phylogenetics and proposing suggestions to reduce the risk of inaccurate inferences in practical use.

To investigate the strengths and weaknesses of PhyDL and Tree_learning, we first designed a test to evaluate the performance of pre-built models provided by PhyDL and Tree_learning, which are likely to be used out-of-the-box by the community (Fig. 1A). Here, the test data sets were simulated under conditions deliberately selected to avoid those well covered in the data used to train existing PhyDL and Tree_learning models.

PhyDL comes with three sets of pre-built DNN models, namely DNN1, DNN2, and

DNN3, differing in the simulation settings (e.g., heterogeneity level and branch length distribution) of their training data. All these DNN models were trained with the long-branch attraction (LBA) condition—also known as the Felsenstein zone—considered, but relatively few long-branch repulsion (LBR) trees—those in the Farris zone—were included in their training data (Table S1). These DNN models showed comparable or superior performance than maximum-likelihood (ML) methods and other traditional phylogenetic methods on data simulated from LBA-susceptible trees (Zou et al., 2020). We first followed the LBA benchmark design from Zou et al. (2020) to evaluate the DNN models on data sets simulated under LBA/LBR conditions (Figs.S1–S5; Supplementary File Text S1). To further examine the performance of DNN models, we used datasets containing AA alignments simulated with progressively complex models (LG+F+Γ, LG+C20+F+Γ, and LG+C60+F+Γ) (Wang et al., 2018) based on LBA and LBR trees (Fig. 1B). We also analyzed these data sets using the ML phylogenetic program IQ-TREE for comparison. For data simulated under LBA condition, none of the three PhyDL models had an accuracy above 50%, while all ML phylogenetic models performed substantially better than DNN models (Figs. 1C, S6). On LBR data sets, the accuracies were 100% for DNN1 and DNN2 but nearly 0% for DNN3, while the accuracies of ML models were between 65.00% and 99.97%. Additionally, we investigated an unexpected performance of DNN3 regarding tree type, noting a high frequency of “incorrect tree – other” on LBA data and “incorrect tree – LBR-I” on LBR data (Figs. 1B, 1C, S7; Supplementary File Text S2). Furthermore, our investigation of the performance of DNN models during their training processes revealed that DNN3 is more vulnerable to model fluctuations during training compared to DNN1 and DNN2 (Fig. S8; Supplementary File Text S3). Overall, our results suggest that the DNN models provided by PhyDL are less accurate than ML phylogenetic models on LBA data.

We then employed the approach developed by Trost et al. (2023) to quantify the disparity between our test data and the pre-built DNN training data. In brief, a Gradient Boosted Trees (GBT) classifier was trained on one data set (e.g., the DNN1 training data) and then applied on another (e.g., our LG+F+Γ LBA test data) to calculate a balanced accuracy (BACC) (Brodersen et al., 2010) value (0 to 1.0, higher values indicate greater differences) which reflects the difference between the two data sets (Materials and Methods in Supplementary Text). As a result, the GBT analyses accurately distinguished each of our test datasets from the training data of pre-built DNN models (with BACC values above 0.99), indicating substantial differences between our test data and the original training data (Table S2; Fig. S9).

In Suvorov et al. (2020), the Tree_learning CNN model trained on gapped data performed much better than traditional phylogenetic methods on gapped alignments, likely because it can extract additional phylogenetic signals from gaps (Suvorov et al., 2020). Specifically, gaps in the training and test data were all simulated by INDELible, and the phylogenetic signals carried by these indel gaps are expected to match the underlying phylogenies. However, real data often contain random gaps (e.g., due to incomplete genome assemblies, partial gene models, or errors in multiple sequence alignments) that may add noise to phylogenetic analyses. To investigate whether the inclusion of random gaps might impact the performance of pre-built CNN models, we first simulated an ungapped data set (NOGAP.ori) and a gapped data set (INDEL.ori) following the procedures of Suvorov et al., and then created two additional data sets, NOGAP.extragaps and INDEL.extragaps, by introducing random gaps into the first two data sets, respectively (Fig. 1D). We applied the CNN model trained on ungapped data (referred to as "CNN.NOGAP.Ori") on NOGAP.ori, and the model trained on gapped data (referred to as "CNN.INDEL.Ori") on the three data sets with gaps. For comparison, we analyzed the data using IQ-TREE under two modes, including "IQ-TREE.Standard", where gaps are treated as missing data with no information, and "IQ-TREE.Recoded", where gaps are recognized as the fifth character in addition to A, T, C, and G. Our evaluation of IQ-TREE and Tree-learning models on NOGAP.ori yielded similar results to those reported by Suvorov et al. (Fig. S10; Supplementary File Text S4). On INDEL.ori, which includes only indel gaps, CNN.INDEL.Ori and IQ-TREE.Recoded achieved much higher accuracy compared to their performance on NOGAP.ori, while the accuracy of IQ-TREE.Standard remained unchanged. However, after random gaps were introduced into the test data, CNN.INDEL.Ori became substantially less accurate on NOGAP.extragaps and INDEL.extragaps, while the two IQ-TREE models had nearly the same accuracies (Fig. 1E). Additionally, we also tested CNN.NOGAP.Ori, CNN.INDEL.Ori and IQ-TREE models across various branch-length combinations (Fig. S11; Supplementary File Text S5). Our results indicated that the inclusion of random noisy gaps in our test data impaired the performance of existing Tree_learning models, rendering them less accurate than IQ-TREE. CNN models trained on indel gaps likely misinterpreted random gaps as informative characters, extracting misleading signals as a result.

In addition to offering pre-built models, both PhyDL and Tree_learning allow users to train new models using custom data. Therefore, we tested if the performance of PhyDL and Tree_learning on difficult data sets could be improved by targeted training using data simulated under the same challenging conditions, either independently or in conjunction with the original training data (Fig. 1F). Importantly, we examined the performance of the new models under both target and non-target conditions to better understand the outcome of this model optimization strategy.

We first examined if targeted training can produce PhyDL models with better accuracies under LBA/LBR conditions. We simulated additional LBA and LBR data sets under LG+C20+F+Γ. These data sets were used to train new DNN models, including DNN_LBA10K (trained on 10,000 LBA alignments), DNN_LBR10K (trained on 10,000 LBR alignments), and DNN_60K (training on 30,000 LBA and 30,000 LBR alignments). Additionally, we trained DNN_160K using the DNN_60K data along with 100,000 alignments simulated similarly to the original DNN3 training data. These new DNN models were applied on the same test data in our first test (Figs. 1G, S12). DNN_LBA10K demonstrated significantly improved performance on LBA data (accuracy exceeding 95%), but showed notable bias when applied to LBR data (Figs. 1G, S12). A similar trend was observed with DNN_LBR10K, which made accurate inferences under LBR conditions, but its accuracy dropped on LBA data. We also found that adding more simulated alignments from a denser sampling of branch length combinations did not improve the performance of DNN_LBA10K and DNN_LBR10K (Fig. S13). DNN_60K and DNN_160K demonstrated a more balanced performance across LBA and LBR conditions, performing between DNN_LBA10K and DNN_LBR10K on both types of test data (Figs. 1G, S12). Notably, DNN_160K performed substantially better than DNN3 on our test data, and its accuracy on the original DNN3 test data ("testing3_mixed") was still close to that of DNN3 itself (Table S3). Unlike DNN3, errors made by all new DNN models were mostly of the expected "incorrect tree – LBA" on LBA data sets, and distributed more evenly between two types of incorrect trees on LBR data sets (Fig. 1G).

For Tree_learning, we trained two new CNN models, CNN.NOGAP.Extragaps and CNN.INDEL.Extragaps, on data sets simulated under the NOGAP.extragaps and INDEL.extragaps schemes, respectively, and tested their performance on NOGAP and INDEL data sets with or without random gaps (Fig. 1H). Generally, the best-performing model for each data set was the one whose training data were simulated in the same way as the test data. CNN.INDEL.Extragaps had considerably higher accuracy than CNN.INDEL.Ori on both NOGAP.extragaps (63.43% vs. 38.57%) and INDEL.extragaps (84.54% vs. 70.16%) (Fig. 1H; Supplementary File Text S6). We further enhanced the performance of CNN.INDEL.Ori on random gaps by conducting additional training with alignments simulated under the INDEL.extragaps scheme. The fine-tuned model (CNN.Fine-tuned) demonstrated significantly higher accuracy than the original CNN.INDEL.Ori model on NOGAP.extragaps (68.65% vs. 38.57%) and INDEL.extragaps (84.83% vs. 70.16%), while maintaining nearly identical performance to CNN.INDEL.Ori on the ungapped dataset NOGAP.ori (69.42% vs. 69.51%) and exhibiting slightly reduced accuracy on INDEL.ori (85.89% vs. 88.17%) (Fig. 1H). Additionally, we tested if targeted training can produce Tree-learning models with better performance under LBA/LBR conditions (TableS4; Supplementary File Text S7). Our results indicate that our targeted optimization effort has successfully enhanced the model’s capability to handle random gaps, albeit with a slight compromise on its performance on phylogenetically informative indels.

In conclusion, our critical evaluation of PhyDL and Tree_learning provides practical evidence that ML methods generally outperformed DNN programs, especially when data properties were unfamiliar to the pre-built DNN models. While DNN performance can be enhanced by training new models tailored to these specific conditions, this comes at the cost of reduced generalizability. Additionally, several challenges must be addressed before DL-based phylogenetic methods can compete with traditional approaches: first, existing DL methods like PhyDL and Tree_learning can only infer quartet trees instead of full phylogenies (in cases of more than four sequences); second, DL methods need to demonstrate their ability to learn patterns from empirical MSAs;

third, few DL methods can successfully infer branch lengths (Supplementary File Text S8).

Based on our results, we recommend assessing the differences between training and test data prior to conducting tree inference to avoid potential pitfalls in phylogenetic reconstruction with DNN programs (Fig. 1I). Our examination of the difference between the pre-built DNN training data and our test data with GBT classifier may serve as an example (Table S2). Overall, our evaluation provides valuable insights for the future development of DNN-based phylogenetic methods and offers practical guidance for their application.

Data availability

All gene alignments and gene trees are available on the figshare repository (https://doi.org/10.6084/m9.figshare.23617767 – please note that this link will become active upon publication). For access during the peer review process, please use the private link (https://figshare.com/s/2c806c4ab1ebc43472c6).

Conflict of interest

The authors declare no competing financial interests.

Acknowledgments

We thank the members of the Shen lab for constructive feedback. We also thank the editor and reviewers for their constructive suggestions to improve our manuscript. X.X.S. was supported by the National Key R&D Program of China (2022YFD1401600), the National Science Foundation for Distinguished Young Scholars of Zhejiang Province (LR23C140001), and the Fundamental Research Funds for the Central Universities (226-2023-00021). X.Z. was supported by the Key-Area Research and Development Program of Guangdong Province (2018B020205003 and 2020B0202090001).

References

Abadi, S., Avram, O., Rosset, S., Pupko, T., Mayrose, I., 2020. ModelTeller: model selection for optimal phylogenetic reconstruction using machine learning. Mol. Biol. Evol. 37, 3338–3352.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8, 53.

Brodersen, K.H., Ong, C.S., Stephan, K.E., Buhmann, J.M., 2010. The balanced accuracy and its posterior distribution. Presented at the 2010 20th international conference on pattern recognition, IEEE, pp. 3121–3124.

Burgstaller-Muehlbacher, S., Crotty, S.M., Schmidt, H.A., Reden, F., Drucks, T., von Haeseler, A., 2023. ModelRevelator: Fast phylogenetic model estimation via deep learning. Mol. Phylogenet. Evol. 188, 107905.

Franklin, J., 2005. The elements of statistical learning: data mining, inference and prediction. Math. Intell. 27, 83–85.

Lajaaiti, I., Lambert, S., Voznica, J., Morlon, H., Hartig, F., 2023. A comparison of deep learning architectures for inferring parameters of diversification models from extant phylogenies. bioRxiv 2023.03.03.530992.

Lambert, S., Voznica, J., Morlon, H., 2023. Deep learning from phylogenies for diversification analyses. Syst. Biol. 72, 1262–1279.

Leuchtenberger, A.F., Crotty, S.M., Drucks, T., Schmidt, H.A., Burgstaller-Muehlbacher, S., 2020. Distinguishing felsenstein zone from farris zone using neural networks. Mol. Biol. Evol. 37, 3632–3641.

Murphy, K.P., 2012. Machine learning: a probabilistic perspective. MIT press.

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

Smith, M.L., Hahn, M.W., 2023. Phylogenetic inference using generative adversarial networks. Bioinformatics 39, btad543.

Suvorov, A., Hochuli, J., Schrider, D.R., 2020. Accurate inference of tree topologies from multiple sequence alignments using deep learning. Syst. Biol. 69, 221–233.

Trost, J., Haag, J., Hohler, D., Nesterenko, L., Jacob, L., Stamatakis, A., Boussau, B., n.d. Simulations of sequence evolution: how (un)realistic they really are and why. bioRxiv 2023.07.11.548509.

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

Wang, H.-C., Minh, B.Q., Susko, E., Roger, A.J., 2018. Modeling site heterogeneity with posterior mean site frequency profiles accelerates accurate phylogenomic estimation. Syst. Biol. 67, 216–235.

Wang, Z., Sun, J., Gao, Y., Xue, Y., Zhang, Y., Li, K., Zhang, W., Zhang, C., Zu, J., Zhang, L., 2023. Fusang: a framework for phylogenetic tree inference via deep learning. Nucleic Acids Res. 51, 10909–10923.

Zaharias, P., Grosshauser, M., Warnow, T., 2022. Re-evaluating deep neural networks for phylogeny estimation: the issue of taxon sampling. J. Comput. Biol. 29, 74–89.

Zou, Z., Zhang, H., Guan, Y., Zhang, J., 2020. Deep residual neural networks resolve quartet molecular phylogenies. Mol. Biol. Evol. 37, 1495–1507.

Yixiao Zhu

College of Agriculture and Biotechnology and Centre for Evolutionary & Organismal Biology, Zhejiang University, Hangzhou 310058, China

Yonglin Li, Chuhao Li

Guangdong Laboratory for Lingnan Modern Agriculture, Guangdong Province Key Laboratory of Microbial Signals and Disease Control, Integrative Microbiology Research Centre, South China Agricultural University, Guangzhou 510642, China

Xing-Xing Shen*

College of Agriculture and Biotechnology and Centre for Evolutionary & Organismal Biology, Zhejiang University, Hangzhou 310058, China

Xiaofan Zhou*

Guangdong Laboratory for Lingnan Modern Agriculture, Guangdong Province Key Laboratory of Microbial Signals and Disease Control, Integrative Microbiology Research Centre, South China Agricultural University, Guangzhou 510642, China

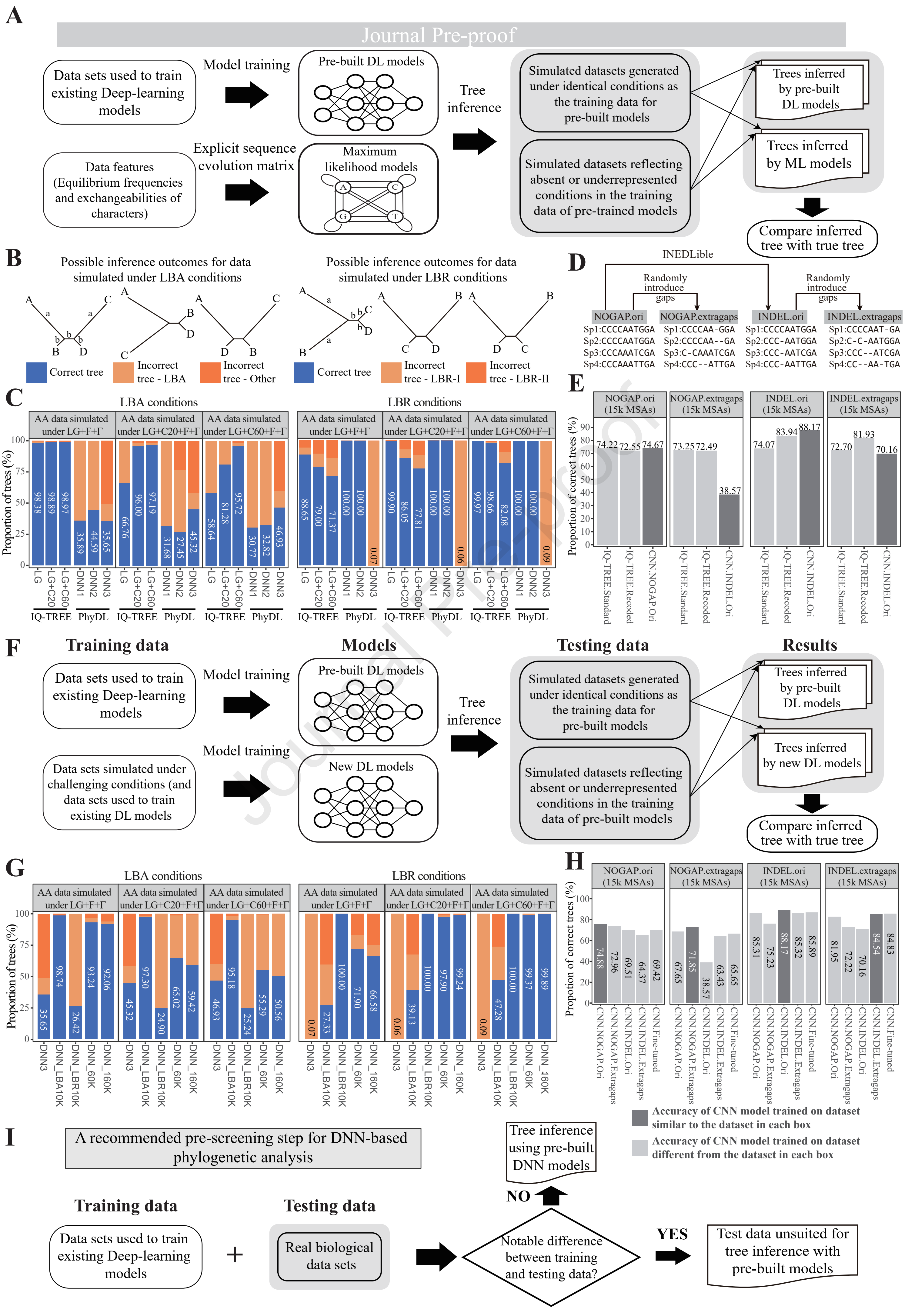
* Corresponding authors

E-mail addresses: xingxingshen@zju.edu.cn (X.-X. Shen);

xiaofan_zhou@scau.edu.cn (X. Zhou)

Figure Legends

Fig. 1. Evaluation of deep learning-based phylogenetic inference programs on simulated datasets. **A**: Schematics of performance evaluations for pre-built models conducted in this study. **B**: Illustrations of the three possible inference outcomes for a four-sequence AA alignment under LBA or LBR conditions, as inferred by IQ-TREE and PhyDL models. **C**: Proportions of different types of trees inferred by IQ-TREE and PhyDL models from test data sets simulated under LBA or LBR conditions. **D**: Schematics of the procedures for simulating the four distinct DNA test datasets used for tree inference with various IQ-TREE and Tree_learning models. **E**: Proportions of correctly inferred trees for various IQ-TREE and Tree_learning models on four simulated test datasets. **F**: Schematics of the performance evaluations for custom-trained models conducted in this study. **G**: Performance of optimized PhyDL models on simulated protein sequence alignments across various branch length combinations. **H**: Performance of new Tree_learning models optimized for the presence of random gaps on simulated DNA sequence alignments. **I**: Schematics of a potential solution to mitigate risks arising from differences between training and testing data.



A
Data sets used to train existing Deep-learning models
Model training
Pre-built DL models
Data features (Equilibrium frequencies and exchangeabilities of characters)
Explicit sequence evolution matrix
Maximum likelihood models
A
C
G
T
Tree inference
Simulated datasets generated under identical conditions as the training data for pre-built models
Simulated datasets reflecting absent or underrepresented conditions in the training data of pre-trained models
Trees inferred by pre-built DL models
Trees inferred by ML models
Compare inferred tree with true tree
B
Possible inference outcomes for data simulated under LBA conditions
Possible inference outcomes for data simulated under LBR conditions
A
B
C
D
a
b
Correct tree
Incorrect tree - LBA
Incorrect tree - Other
Correct tree
Incorrect tree - LBR-I
Incorrect tree - LBR-II
C
LBA conditions
LBR conditions
AA data simulated under LG+F+Γ
AA data simulated under LG+C20+F+Γ
AA data simulated under LG+C60+F+Γ
Proportion of trees (%)
100
75
50
25
0
98.38
98.89
98.97
35.89
44.59
35.65
66.76
96.00
97.19
31.68
27.45
45.32
58.64
81.28
95.72
30.77
32.82
46.93
88.65
79.00
71.37
100.00
100.00
0.07
99.90
86.05
77.81
100.00
100.00
0.06
99.97
98.66
82.08
100.00
100.00
0.09
LG
LG+C20
LG+C60
DNN1
DNN2
DNN3
IQ-TREE
PhyDL
D
INEDLible
Randomly introduce gaps
NOGAP.ori
NOGAP.extragaps
INDEL.ori
INDEL.extragaps
Sp1:CCCCAATGGA
Sp2:CCCCAATGGA
Sp3:CCCAAATCGA
Sp4:CCCAAATTGA
Sp1:CCCCAA-GGA
Sp2:CCCCAA--GA
Sp3:C-CAAATCGA
Sp4:CCC--ATTGA
Sp1:CCCCAATGGA
Sp2:CCC-AATGGA
Sp3:CCC-AATCGA
Sp4:CCC-AATTGA
Sp1:CCCCAAT-GA
Sp2:C-C-AATGGA
Sp3:CCC--ATCGA
Sp4:CC--AA-TGA
E
NOGAP.ori (15k MSAs)
NOGAP.extragaps (15k MSAs)
INDEL.ori (15k MSAs)
INDEL.extragaps (15k MSAs)
Proportion of correct trees (%)
90
80
70
60
50
40
30
20
10
0
74.22
72.55
74.67
73.25
72.49
38.57
74.07
83.94
88.17
72.70
81.93
70.16
IQ-TREE.Standard
IQ-TREE.Recoded
CNN.NOGAP.Ori
CNN.INDEL.Ori
F
Training data
Models
Testing data
Results
Data sets used to train existing Deep-learning models
Model training
Pre-built DL models
Data sets simulated under challenging conditions (and data sets used to train existing DL models
Model training
New DL models
Tree inference
Simulated datasets generated under identical conditions as the training data for pre-built models
Simulated datasets reflecting absent or underrepresented conditions in the training data of pre-built models
Trees inferred by pre-built DL models
Trees inferred by new DL models
Compare inferred tree with true tree
G
LBA conditions
LBR conditions
AA data simulated under LG+F+Γ
AA data simulated under LG+C20+F+Γ
AA data simulated under LG+C60+F+Γ
Proportion of trees (%)
100
75
50
25
0
35.65
98.74
26.42
93.24
92.06
45.32
97.30
24.90
65.02
59.42
46.93
95.18
25.24
55.29
50.56
0.07
27.33
100.00
71.90
66.58
0.06
39.13
100.00
97.90
99.24
0.09
47.28
100.00
99.37
99.89
DNN3
DNN_LBA10K
DNN_LBR10K
DNN_60K
DNN_160K
H
NOGAP.ori (15k MSAs)
NOGAP.extragaps (15k MSAs)
INDEL.ori (15k MSAs)
INDEL.extragaps (15k MSAs)
Propotion of correct trees (%)
100
80
60
40
20
0
74.88
72.96
69.51
64.37
69.42
67.65
71.85
38.57
63.43
65.65
85.31
75.23
88.17
85.32
85.89
81.95
72.22
70.16
84.54
84.83
CNN.NOGAP.Ori
CNN.NOGAP.Extragaps
CNN.INDEL.Ori
CNN.INDEL.Extragaps
CNN.Fine-tuned
Accuracy of CNN model trained on dataset similar to the dataset in each box
Accuracy of CNN model trained on dataset different from the dataset in each box
I
A recommended pre-screening step for DNN-based phylogenetic analysis
Training data
Testing data
Data sets used to train existing Deep-learning models
+
Real biological data sets
Notable difference between training and testing data?
NO
Tree inference using pre-built DNN models
YES
Test data unsuited for tree inference with pre-built models