Learning from an Artificial Neural Network in Phylogenetics

Alina F. Leuchtenberger and Arndt von Haeseler

***Abstract*—We show that an iterative ansatz of deep learning and human intelligence guided simplification may lead to surprisingly simple solutions for a difficult problem in phylogenetics.**

Distinguishing Farris and Felsenstein trees is a longstanding problem in phylogenetic tree reconstruction. The Artificial Neural Network F-zoneNN solves this problem for 4-taxon alignments evolved under the Jukes-Cantor model. It distinguishes between Farris and Felsenstein trees, but owing to its complexity, lacks transparency in its mechanism of discernment. Based on the simplification of F-zoneNN and alignment properties we constructed the function FarFelDiscerner. In contrast to F-zoneNN, FarFelDiscerner's decision process is understandable. Moreover, FarFelDiscerner is way simpler than F-zoneNN.

Despite its simplicity this function infers the tree-type almost perfectly on noise-free data, and also performs well on simulated noisy alignments of finite length. We applied FarFelDiscerner to the historical Holometabola alignments where it places Strepsiptera with beetles, concordant with the current scientific view.

***Index Terms*—Artificial neural networks, ANN simplification, phylogenetics, Felsenstein zone, Farris zone, LBA, LBR**

I. Introduction

Artificial Neural Networks (ANNs) are powerful learning methods performing classifications, pattern recognition tasks and more (see [1], [2] and references therein). Fundamentally, they are mathematical functions whose parameters are fitted such that these functions yield the desired output [3]. Recently ANNs have been applied in the field of phylogenetic inference [4]–[10], the process of reconstructing phylogenetic trees depicting the evolutionary history of contemporary taxa, based on e.g. an alignment of their DNA sequences.

Over the years phylogenetics has benefited from increasingly complex mathematical models as well as a dramatic increase of available sequencing data (see e.g. [11]–[14]). While phylogenetic inference is in general performing well, it is computationally expensive and sometimes produces misleading trees (see e.g. [15]).

Alina F. Leuchtenberger is with the Center for Integrative Bioinformatics Vienna, Max Perutz Labs, Vienna Biocenter Campus (VBC), Dr.-Bohr-Gasse 9, 1030, Vienna, Austria, the Medical University of Vienna, Center for Medical Biochemistry, Dr.-Bohr-Gasse 9, 1030, Vienna, Austria and the Vienna Biocenter PhD Program, a Doctoral School of the University of Vienna and the Medical University of Vienna, A-1030 Vienna, Austria
E-mail: alina.leuchtenberger@univie.ac.at

Arndt von Haeseler is with the Center for Integrative Bioinformatics Vienna, Max Perutz Labs, University of Vienna and Medical University of Vienna, Dr. Bohr-Gasse 9, 1030 Vienna, Austria
and the Bioinformatics and Computational Biology, Faculty of Computer Science, University of Vienna, Vienna, Austria
E-mail: arndt.von.haeseler@univie.ac.at

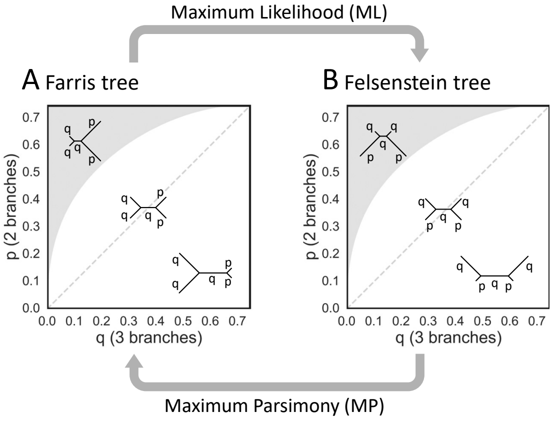


Fig. 1. Farris (A) and Felsenstein trees (B) for varying probabilities (p, q) of observing a nucleotide substitution along a branch. The grey region in both plots represents the Felsenstein zone. While MP tends for the grey region towards reconstructing a Farris tree although the alignments evolved under a Felsenstein tree, ML tends for the grey region towards reconstructing a Felsenstein tree although the alignments evolved under a Farris tree.

Farris and Felsenstein trees are two tree types where incorrect reconstructions can take place. That is to say, for any alignment, that evolved under a Farris/Felsenstein tree a wrong tree is inferred. More specifically, Farris/Felsenstein trees are unrooted 4-taxon trees, i.e. trees with four external nodes each representing one of the four contemporary taxa, and two internal nodes representing their ancestors (Fig. 1). In this paper we focus on the simplest form of Farris and Felsenstein trees where the branches connecting the taxa have only two different branch lengths p and q. These branch lengths are measured by the probability of observing a nucleotide substitution along the respective branch and are therefore indicating the evolutionary distance between the taxa. For both tree types the internal branch (the branch between the internal nodes) and two external branches (branches between an external and an internal node) have length q while the other two external branches have length p.

Farris and Felsenstein trees differ in their pairing of the branches (Fig. 1): Farris trees pair the two branches with length p with each other (see Fig. 1A), while Felsenstein trees pair each branch with length p with a branch of length q (see Fig. 1B).

We consider all trees where p and q range between 0 and 0.75 (see Fig. 1).

If p and q are from the grey region in Figure 1 then an alignment evolved under a Felsenstein tree will be incorrectly reconstructed under the reconstruction method Maximum Par-

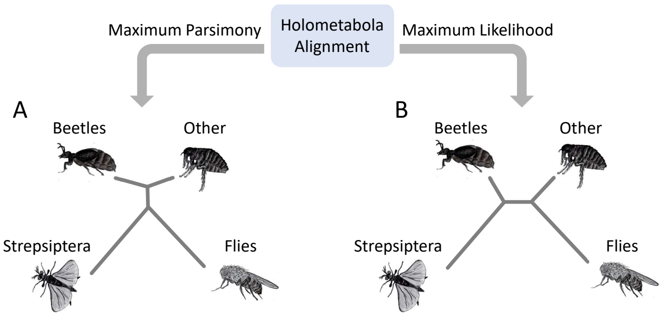


Fig. 2. The two debated placements of Strepsiptera: Maximum Parsimony places Strepsiptera as sister to flies (A) and Maximum Likelihood places Strepsiptera as sister to beetles (B).

simony (MP). The resulting tree will look like a Farris tree. On the other hand for an alignment evolved under a Farris tree with p and q being from the grey region in Figure 1 the reconstruction method Maximum Likelihood (ML) will reconstruct a tree similar to a Felsenstein tree.

The incorrect reconstruction of Farris trees has been coined long branch attraction (LBA) by Felsenstein in 1978 [15] and was further explored by e.g. Hendy and Penny in 1989 [16]. The branch length parameter space where LBA leads to incorrect tree reconstruction was coined the Felsenstein zone [17] (see Fig. 1, grey regions). The zone depends only on the values of p and q, and is independent of the alignment length. More precisely, MP is statistically inconsistent in the Felsenstein zone, that is MP will incorrectly reconstruct a Farris tree for infinitely long alignments.

The second phenomenon was later described as long branch repulsion (LBR), which biases Maximum Likelihood (ML) towards reconstructing a Felsenstein tree (Fig. 1B) for finite alignments which evolved under a Farris tree (Fig. 1A) [18], [19]. The parameter space where LBR takes place was called the Farris zone [19]. Unlike MP, ML is statistically consistent. The Farris zone depends on the alignment length. If the alignment length approaches infinity the Farris zone will disappear and ML will reconstruct the correct tree in spite of the effect of LBR [20].

However, biological alignments have finite lengths. If ML and MP yield discordant results, it is unclear whether LBR or LBA takes place and which method is correct. A well-studied empirical example is the reconstruction of the Holometabola phylogeny based on alignments of Strepsiptera, flies, beetles and other Holometabola [21], [22]. While MP reconstructs a Farris tree which pairs Strepsiptera with flies (Fig. 2A), ML reconstructs a Felsenstein tree as it places Strepsiptera with the beetles and distant to the flies (Fig. 2B). For a long time there was lively discussion about which of these trees is correct.

The disputed placement of Strepsiptera is only one example of the problem of distinguishing Farris and Felsenstein trees, many other examples exist and reliably distinguishing Farris and Felsenstein trees remains an important open problem in phylogenetics [15], [18], [19].

To resolve such discordant placements, Leuchtenberger et al. [6] developed the ANN F-zoneNN for four-taxon alignments evolved under the Jukes-Cantor model [23]. F-zoneNN infers whether a given alignment stems from a Farris or a Felsenstein tree. Thus, it classifies the unknown tree type rather than reconstructing a tree and uses this classification to select the appropriate tree reconstruction method. This strategy reduces the danger of artefacts [6].

The ANN F-zoneNN is a feed-forward neural network composing 9 linear and 9 non-linear functions with more than 1.2 million parameters. Consequently, F-zoneNN acts as a black-box and we do not understand how it distinguishes between the two tree-types.

To counteract the lack of interpretability, in recent years ANN interpretation methods like LIME [24], Partial Dependence Plots [25] or the Activation Maximisation proposed by Erhan et al. [26] were developed. However, they only provide incomplete descriptions of the decision process of an ANN and so ANNs like F-zoneNN remain by and large black-boxes. This lack of interpretability is for many applications not problematic as the user is more interested in the output of an ANN. Still, an interpretable method is desirable as it does not only solve a task, but also provides theoretical insights.

In the following, we simplify F-zoneNN by taking suitable properties of alignments into account. We combine the capability of ANNs with the rich theory on phylogenetic inference and construct the simple function FarFelDiscerner which successfully distinguishes between Farris and Felsenstein trees. FarFelDiscerner allows us to understand how the alignments evolved under Farris trees and those evolved under Felsenstein trees differ from each other. Moreover, FarFelDiscerner, applied to the empirical Holometabola alignments [21], places the Strepsiptera with respect to flies and beetles concordant to the current scientific view.

The outline of this paper is as follows. We will first describe the ANN F-zoneNN [6] in detail. Next, we will simplify F-zoneNN and define equivalence classes based on this simplification. From these equivalence classes we then construct the function FarFelDiscerner and evaluate it on simulated alignments as well as on the empirical Holometabola alignments [21].

II. Background: F-zoneNN

The ANN F-zoneNN [6] whose simplification is the subject of this paper infers whether a four-taxon alignment evolved under a Farris or a Felsenstein tree. A four-taxon alignment $(A_{ij})_{i=1,\ldots,4;j=1,\ldots,n}$ displays in each row i the DNA sequence of a taxon, and in every column j one of the n alignment sites. Each alignment can be succinctly described by the frequencies of its site-patterns i.e. the frequencies of the $4^4 = 256$ possible arrangements of 4 nucleotides on the 4 positions of each alignment site: $AAAA$, $AAAC$, $AAAG$, ..., $TTTT$.

An ANN benefits parameter-wise from a less complex input, thus F-zoneNN uses an input which describes the alignments succinctly. To further reduce the 256 site-patterns, it is taken into account that the sequences evolved under the Jukes-Cantor model [23] and therefore that the nucleotide frequencies and mutation rates are uniform. Due to these assumptions the probability of a site-pattern is only influenced by which sequences show the same nucleotide in the pattern and which do not. For example $AAAC$ and $TTTG$ occur with the same probability as both describe sites where all but the last position show the same nucleotide. Thus, the 256 site-patterns collapse to 15 distinct site-patterns. Each site-pattern

$$s \in \{1234, 1|234, 2|134, 3|124, 4|123, 12|34, 13|24, 14|23,\\ 1|2|34, 1|3|24, 1|4|23, 2|3|14, 2|4|13, 3|4|12, 1|2|3|4\}$$

where 1, 2, 3 and 4 represent the row indices of the alignment and $|$ indicates that the sequences with row indices to the left and right of $|$ have different nucleotides. Thus, $1|2|34$ represents alignment columns where the 1st, 2nd and 3rd row contain mutually different nucleotides and the 4th row contains the same nucleotide as the 3rd row. The site-patterns $1|234$, $2|134$, $3|124$ and $4|123$ represent alignment columns where one sequence has another nucleotide than the other three. Their frequencies indicate how different a sequence is from the others and thus, reflect the branch lengths leading to the corresponding taxa within the tree. Similarly, the frequencies of $12|34$, $13|24$ and $14|23$ measure how different two sequence pairs are from each other and can indicate the existence or length of an internal branch between the corresponding taxa pairs.

$f(s)$ denotes the relative frequency of site-pattern s for an alignment and $\mathbf{f} = (f(1234), ..., f(1|2|3|4)) \in [0,1]^{15}$ is the vector of all site-pattern frequencies of this alignment. Thus, the elements of $\mathbf{f}$ always sum up to 1. $\mathbf{f}$ serves as input for F-zoneNN and is processed through 9 layers. Each layer l takes the data from the previous layer (except for layer 1 which takes $\mathbf{f}$ as input) and applies a linear affine function A_l and a nonlinear function σ_l to it. We denote the dimension of the output of layer l with n_l.

The linear function of layer l, $A_l(x) = W_l \cdot x + b_l$ for $x \in \mathbb{R}^{n_{l-1}}$, includes a weight matrix $W_l \in \mathbb{R}^{n_l \times n_{l-1}}$ and an offset vector $b_l \in \mathbb{R}^{n_l}$. In contrast, the nonlinear functions act on the elements of the vectors and do not alter the vector dimensions. For the first 8 layers σ_l is a Rectified Linear Unit (ReLU) and for the last layer a sigmoid function:

$$\sigma_l(x) = max(0, x) \text{ for } l = 1, ..., 8, \quad \sigma_9(x) = \frac{1}{1 + e^{-x}}.$$

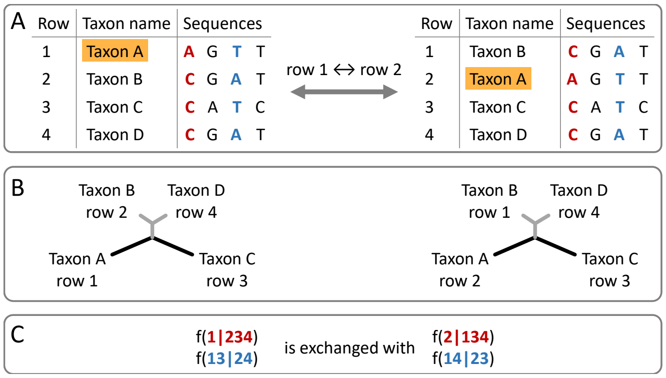
Thus, F-zoneNN is the composition of 9 linear affine and 9 non-linear functions:

$$\begin{aligned} &\text{F-zoneNN}: \ [0,1]^{15} \to [0,1], \\ &\text{F-zoneNN}(\mathbf{f}) = \sigma_9 \circ A_9 \circ ... \circ \sigma_2 \circ A_2 \circ \sigma_1 \circ A_1(\mathbf{f}). \end{aligned} \tag{1}$$

If F-zoneNN's output is ≥ 0.5, F-zoneNN infers a Farris tree and otherwise it infers a Felsenstein tree.

The training and testing of F-zoneNN took place on four-taxon alignments simulated under the Jukes-Cantor model [23] for Farris and Felsenstein trees with varying branch length parameters p and q [6]. Thus, for each training and each testing alignment the correct tree type is known.

The 1.2 million parameters of the weight matrices W_l and the offsets b_l were optimised on training data. This huge number of trainable parameters together with F-zoneNN's complex structure make it impossible to understand what drives F-zoneNN's decisions. Therefore, we will reduce the complexity of F-zoneNN in the next section.



Row	Taxon name	Sequences
1	Taxon A	A G T T
2	Taxon B	C G A T
3	Taxon C	C A T C
4	Taxon D	C G A T

Row	Taxon name	Sequences
1	Taxon B	C G A T
2	Taxon A	A G T T
3	Taxon C	C A T C
4	Taxon D	C G A T

Fig. 3. Row permutation of a sample alignment (A), the corresponding trees with taxon labelled by name and alignment row number of their sequences (B) as well as site-patterns whose frequencies are exchanged due to the swapping of row 1 and 2 (C).

III. Simplification of F-zoneNN

We diminish the complexity of F-zoneNN by studying the ANN itself as well as incorporating knowledge on phylogenetic inference.

As a first step we gradually reduced the number of layers. A network with 3 layers still achieved a sufficient performance whereas reducing to only 2 layers resulted in a much lower performance. Therefore, we decided to use a network with 3 layers. Then we removed the activation function of the last layer, all offsets b_l for $l \in \{1, 2, 3\}$ and replaced the ReLU activation functions with square functions (see Section 1 in the supplemental material) resulting in the function

$$\text{F-zonePoly}(\mathbf{f}) = W_3 \cdot (W_2 \cdot (W_1 \cdot \mathbf{f})^2)^2. \tag{2}$$

As opposed to F-zoneNN (1) which outputs a value in $[0, 1]$ F-zonePoly (2) assumes real numbers and infers a Farris tree if the output ≥ 0.

F-zonePoly (2) can be rewritten as a polynomial of degree four. Each term is a product of one of the

$$\binom{15 + 4 - 1}{4} = \frac{(15 + 4 - 1)!}{4! \cdot (15 - 1)!} = 3060$$

possible combinations of 4 out of the 15 site-pattern frequencies with repetitions [27]. For the reminder of the paper a combination is a multiset with cardinality four, where the elements do not necessarily have multiplicity one. These terms are multiplied by the unknown coefficients c_k with $k \in \{1, ..., 3060\}$ which are functions of entries of the weight matrices W_l for $l \in \{1, 2, 3\}$. Thus, we obtain:

$$\begin{aligned} \text{F-zonePoly}: \ & [0,1]^{15} \to \mathbb{R}, \\ \text{F-zonePoly}(\mathbf{f}) =& c_1 \cdot f(1234)^4 \\ &+ c_2 \cdot f(1234)^3 \cdot f(1|234) \\ &+ ... + c_{3060} \cdot f(1|2|3|4)^4. \end{aligned} \tag{3}$$

To further simplify F-zonePoly, we note that 24 permutations of the four rows of an alignment are possible.

The tree is invariant under such a row permutation as a row permutation does not change the relation of the sequences to each other, it only changes their order within the alignment (see Fig. 3A,B). However, the site-pattern frequencies $\mathbf{f}$ are

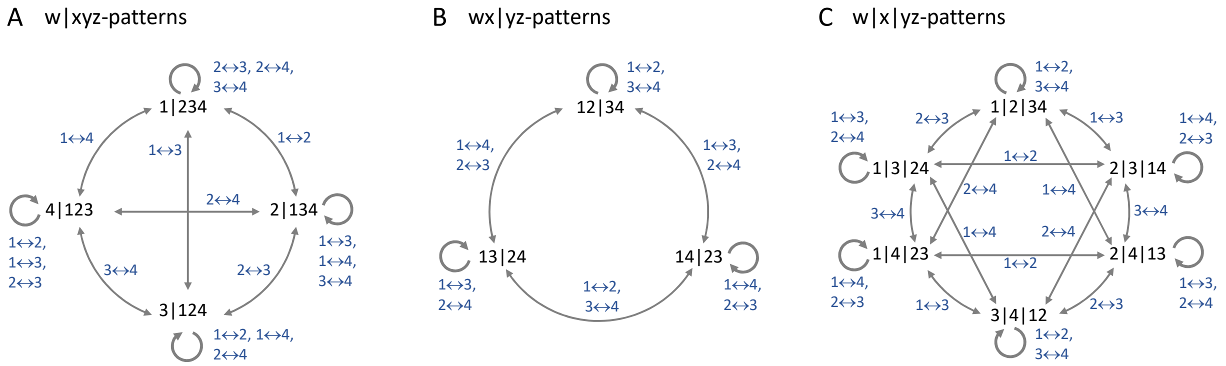


Fig. 4. The $w|xyz$- (A), $wx|yz$- (B) and $w|x|yz$-patterns (C). An edge connects two patterns if a swap of rows (blue) exists to transform one pattern into the other. Note, that any combination of swaps reflects a possible row permutation and so each $w|xyz$- (A), $wx|yz$- (B) and $w|x|yz$-pattern (C) can be transformed to any pattern of the same form.

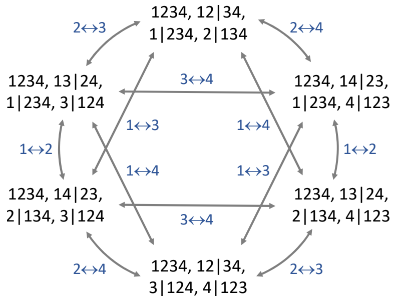


Fig. 5. Site-pattern combinations which are transformed to each other under row permutations. Grey arrows between two combinations indicate that a swap of two rows transforms one combination to the other.

not invariant under row permutations. If for example the 1st and 2nd row of an alignment are swapped the sites with pattern $1|234$ are changed to sites with pattern $2|134$ and vice versa such that $f(1|234)$ and $f(2|134)$ are exchanged (see Fig. 3C). Thus, permutations of rows lead to a permutation of $\mathbf{f}$ and to possibly different values of the terms in F-zonePoly (3). To ensure that F-zonePoly produces the same values for different permutations of the same alignment, there are several strategies. We could train F-zonePoly by considering all permutations as it was done for F-zoneNN [6] or we could transform F-zonePoly's architecture such that it is invariant under row permutations (see e.g. [7]). However, we construct in the following invariant input features for F-zonePoly.

Generally, each $w|xyz$-pattern (a site-pattern of form $w|xyz$ where $w, x, y, z \in \{1, 2, 3, 4\}$ are pairwise distinct) can be transformed to any other $w|xyz$-pattern by a row permutation (see Fig. 4A). Similarly, each $wx|yz$-pattern and each $w|x|yz$-pattern can be transformed to any other $wx|yz$-pattern and $w|x|yz$-pattern, respectively (see Fig. 4B,C). Only the patterns 1234 and $1|2|3|4$ do not change under row permutations.

As row permutations transform site-patterns, they also transform all 3060 combinations. If, for example, the 2nd and 4th sequence in an alignment are swapped, the combination $\{1234, 12|34, 1|234, 2|134\}$ is transformed to the combination $\{1234, 14|23, 1|234, 4|123\}$.

To take advantage of the transformations induced by the row permutations, we call two combinations equivalent if a row permutation transforms them into each other. This way we partition the 3060 combinations into 269 equivalence classes (equivalence_classes.tsv in the online supplemental material). For example the combinations $\{1234, 12|34, 1|234, 2|134\}$ and $\{1234, 14|23, 1|234, 4|123\}$ belong to the same equivalence class comprising six combinations (see Fig. 5). $[e]$ denotes a representation of the equivalence class containing combination e.

With this notation we can simplify F-zonePoly (3) by summing all site-pattern frequency products whose combinations are in the same equivalence class. So the 6 site-pattern frequency products of the combinations depicted in Fig 5 are summed, i.e. the associated coefficients in F-zonePoly are equal. Thus, we don't need to train 269 coefficients and therefore, the number of terms of F-zonePoly is reduced by 91%. However, a polynomial with 269 coefficients is still complex.

To achieve a further reduction, we performed a variant of backward elimination (see Section 2 in the supplemental material). After 11 iterations this process converged to 21 equivalence classes (column equis_after_iterative_elimination in the supplemental file equivalence_classes.tsv).

Subsequently, we trained a network including these 21 equivalence classes. This network is linear in the 21 sums of all site-pattern frequency products whose combinations are part of the same of the 21 equivalence classes. Next, we eliminated the equivalence classes in this network one by one. In each step the equivalence is removed which reduces the networks accuracy the least. E_1, E_2, E_4 and E_5 (see Table I) were the last four equivalence classes that were removed. Thus, E_1, E_2, E_4 and E_5 are important for the networks accuracy.

Interestingly, $E_1 = [\{1234, 13|24, 2|134, 2|134\}]$ and $E_2 = [\{1234, 13|24, 2|134, 4|134\}]$ look very similar. Indeed, the combination $\{1234, 13|24, 2|134, 2|134\}$ in E_1 and the combination $\{1234, 13|24, 2|134, 4|134\}$ in E_2 differ only by their respective last site-pattern, which is a $w|xyz$-pattern. This holds true for all combinations of E_1 and E_2. If we look up the 269 equivalence classes, E_3 (see Table I) shows the same structure as E_1 and E_2.

TABLE I
THE SIX FINAL EQUIVALENCE CLASSES OBTAINED TOGETHER WITH THEIR CARDINALITY AND A DESCRIPTION OF THE PATTERNS OF EACH COMBINATION OF THE RESPECTIVE EQUIVALENCE CLASS.

Equivalence class E	$\lvert E\rvert$	Patterns of the combination
$E_1 = [\{1234, 13\lvert 24, 2\lvert 134, 2\lvert 134\}]$	12	the $wxyz$-pattern, one of the three $wx\lvert yz$-patterns and one of the four $w\lvert xyz$-patterns with multiplicity two
$E_2 = [\{1234, 13\lvert 24, 2\lvert 134, 4\lvert 134\}]$ (see Fig. 5)	6	the $wxyz$-pattern, one of the three $wx\lvert yz$-patterns and two $w\lvert xyz$-patterns which isolate taxa that are paired in the $wx\lvert yz$-pattern (of which there are two choices for each $wx\lvert yz$-pattern)
$E_3 = [\{1234, 13\lvert 24, 2\lvert 134, 3\lvert 134\}]$	12	the $wxyz$-pattern, one of the three $wx\lvert yz$-patterns and two $w\lvert xyz$-patterns which isolate taxa that are not paired in the $wx\lvert yz$-pattern (of which there are four choices for each $wx\lvert yz$-pattern)
$E_4 = [\{4\lvert 123, 4\lvert 123, 3\lvert 124, 4\lvert 123\}]$	12	one of the four $w\lvert xyz$-patterns with multiplicity three and one of the remaining three $w\lvert xyz$-patterns
$E_5 = [\{4\lvert 123, 4\lvert 123, 3\lvert 124, 3\lvert 123\}]$	6	two different of the four $w\lvert xyz$-patterns, both with multiplicity two
$E_6 = [\{4\lvert 123, 4\lvert 123, 3\lvert 124, 2\lvert 123\}]$	12	three different of the four $w\lvert xyz$-patterns, one pattern with multiplicity two and two patterns with multiplicity one

The combinations of E_4 and E_5 have the same relation to each other: For each combination in E_4, there is a combination in E_5 (and vice versa) such that the two combinations differ only in one $w|xyz$-pattern. And again, we find a third equivalence class with the same structure: E_6 (see Table I).

After an extensive analysis we found six equivalence classes E_1, E_2, E_3, E_4, E_5 and E_6 that are important to infer if an alignment evolved under a Farris or a Felsenstein tree. In the following, we will therefore focus on these classes and how we can use them to distinguish between the two tree types.

IV. From Equivalence Classes To FarFelDiscerner

In this section we construct the simple function FarFelDiscerner based on E_1, E_2, E_3, E_4, E_5 and E_6. We start by analysing these equivalence classes. We do this with respect to the expected site-pattern frequencies of Farris and Felsenstein trees with varying branch length parameters p and q [28].

While F-zonePoly (3) works on site-pattern frequency products which are straightforward to compute, FarFelDiscerner operates on equivalence classes containing multiple combinations, each of them providing one site-pattern frequency product. To get a unique value for each equivalence class, we compute the maximal product of E

$$max(E) = max\{\Pi_{s\in e} f(s) | e \in E\}$$

and the argmax

$$argmax(E) = argmax\{\Pi_{s\in e} f(s) | e \in E\}.$$

The $argmax(E)$ hints on which site-patterns are large. With the information which site-pattern is large among same structured site-patterns we can infer properties of the tree. If, for example, $1|234$ is the maximal $w|xyz$-frequency, then the branch leading to the taxon of the first row is probably comparatively long as the expected frequency of a $w|xyz$-pattern is larger the longer the branch leading to the taxon of row w.

We first analyse E_4, E_5 and E_6 containing only $w|xyz$-patterns. Their argmax are those combining the most frequent $w|xyz$-patterns.

For simplicity, we call the maximal $w|xyz$-frequency i_1, the second largest i_2, the third largest i_3 and the smallest i_4. Then the maxima of E_4, E_5 and E_6 are

$$\begin{aligned} max(E_4) &= i_1^3 \cdot i_2, \\ max(E_5) &= i_1^2 \cdot i_2^2, \\ max(E_6) &= i_1^2 \cdot i_2 \cdot i_3. \end{aligned}$$

The longer the branch of a taxon w the larger the $w|xyz$-frequency. Farris and Felsenstein trees include two long external branches and two comparatively short external branches. Therefore, i_1 and i_2 are the expected frequencies of the patterns isolating the long-branched taxa, while i_3 is the frequency of a pattern isolating a comparatively short branch. Thus,

$$\frac{max(E_5)}{max(E_4)} = \frac{i_2}{i_1}$$

describes the ratio of the frequencies of the $w|xyz$-patterns isolating the long-branched taxa. Here, this ratio is 1 as we assume that the two long branches are equal. On the other hand,

$$\frac{max(E_6)}{max(E_4)} = \frac{i_3}{i_1}$$

describes the ratio of the frequencies of a $w|xyz$-pattern isolating a short-branched taxon (i_3) and that isolating a long-branched taxon (i_1).

As a result the difference of the two quotients

$$\frac{max(E_5)}{max(E_4)} - \frac{max(E_6)}{max(E_4)} = \frac{i_2}{i_1} - \frac{i_3}{i_1} \tag{4}$$

increases as the difference in lengths of the long and short branches increases. Thus, equation (4) measures the differences of the external branch lengths, but can not distinguish between tree-types.

Now, we show that the remaining equivalence classes E_1, E_2 and E_3 can distinguish between the tree-types. These equivalence classes combine $w|xyz$- and $wx|yz$-patterns. Therefore, $argmax(E_1)$, $argmax(E_2)$ and $argmax(E_3)$ depend on which branches are long and which branches are paired. Thus, we examine the most frequent $w|xyz$- and $wx|yz$-patterns (Fig. 6A,E) and the patterns in $argmax(E_1)$ (Fig. 6B,F), $argmax(E_2)$ (Fig. 6C,G) and $argmax(E_3)$ (Fig. 6D,H).

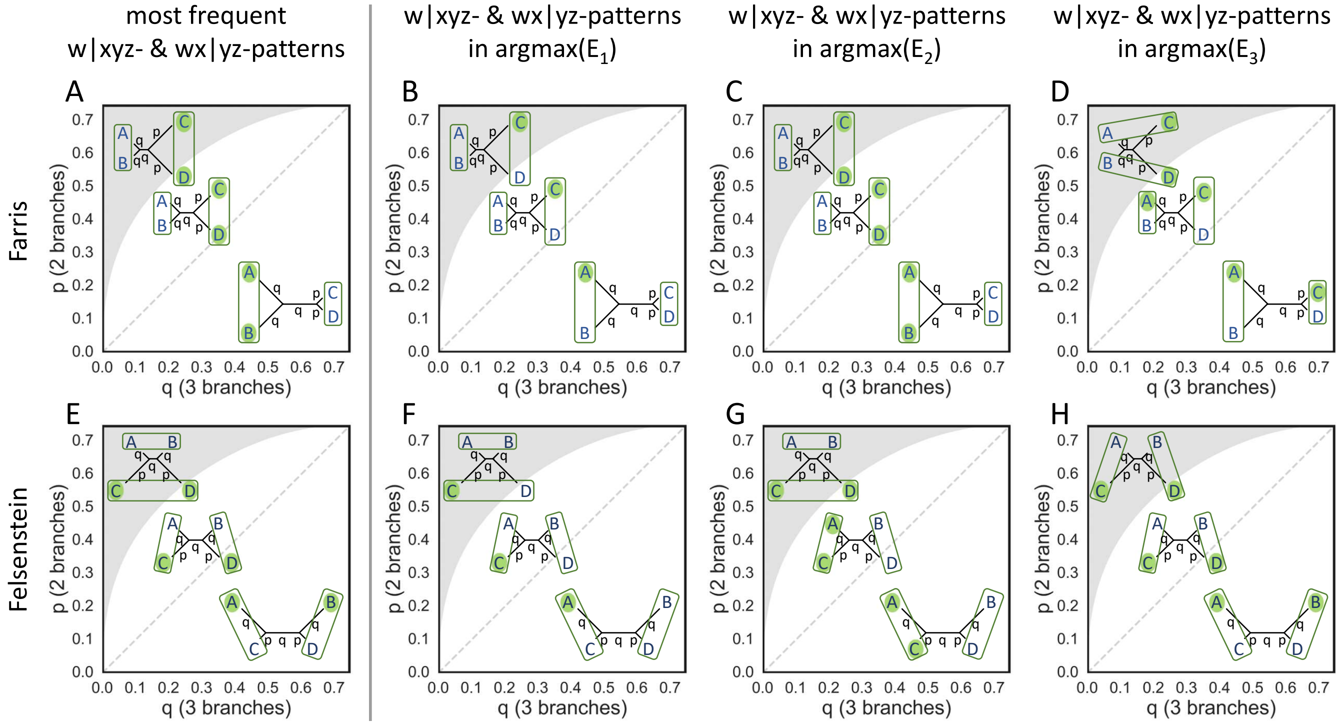


Fig. 6. The plot (A-H) depict three Farris (A-D) and Felsenstein (E-H) trees with various branch lengths. Taxa highlighted green are those isolated in those $w|xyz$-patterns which are most frequent (A,E), which are part of $argmax(E_1)$ (B,F), $argmax(E_2)$ (C,G) or $argmax(E_3)$ (D,H). The green rectangles indicate which taxa are paired in the $wx|yz$-pattern which is most frequent (A,E), part of $argmax(E_1)$ (B,F), $argmax(E_2)$ (C,G) or $argmax(E_3)$ (D,H). In each plot the grey region represents the Felsenstein zone and along the dashed line $p = q$.

To simplify the subsequent notation, we name the taxa of the trees A, B, C and D, the row indices of their sequences we call a, b, c and d, respectively, where $a, b, c, d \in \{1, 2, 3, 4\}$ are pairwise different. While the branches leading to taxa C and D have length p, those leading to A and B and the internal branch have length q. The Farris trees pair taxa with equally long branches i.e. taxon A with taxon B and taxon C with taxon D (Fig. 6A-D). In contrast the Felsenstein trees pair taxon A with taxon C and taxon B with taxon D (Fig. 6E-H). To treat Farris and Felsenstein trees as distinct groups, we assume $p \neq q$ in the subsequent analysis.

$E_1 = [\{1234, 13|24, 2|134, 2|134\}]$ includes all combinations of a $w|xyz$-pattern and a $wx|yz$-pattern. Thus, $argmax(E_1)$ combines the most frequent $w|xyz$-pattern with the most frequent $wx|yz$-pattern (cf. Fig. 6 A,B,E,F). For Farris and Felsenstein trees the $w|xyz$-patterns with the largest expected frequency are the patterns isolating the taxa with the longest branches: $a|bcd$ and $b|acd$ if $q > p$ and $c|abd$ and $d|abc$ if $p > q$ (Fig. 6A,E). Thus, one of these patterns is also the $w|xyz$-pattern of $argmax(E_1)$ (Fig. 6B,F). The most frequent $wx|yz$-pattern and so the $wx|yz$-pattern of $argmax(E_1)$ is for each Farris tree the pattern $ab|cd$ which corresponds to the taxa pairing in the tree (Fig. 6A,B). For Felsenstein trees outside the Felsenstein zone the most frequent $wx|yz$-pattern and thus the $wx|yz$-pattern of $argmax(E_1)$ is $ac|bd$ which again corresponds to the taxa pairing in the tree (Fig. 6E,F, white region). However, for Felsenstein trees in the Felsenstein zone the most frequent $wx|yz$-pattern does not reflect the taxa pairing in the tree (Fig. 6E, grey region). Instead, the most frequent $wx|yz$-pattern and so the $wx|yz$-pattern of $argmax(E_1)$ for Felsenstein trees in the Felsenstein zone is $ab|cd$ which pairs the two long branches (Fig. 6E,F, grey region).

Therefore, we can infer the pairing and the long branches of a tree from $argmax(E_1)$ based on expected site-pattern frequencies unless this tree is a Felsenstein tree in the Felsenstein zone. For these trees the $wx|yz$-pattern of $argmax(E_1)$ does not hint on the taxa pairing in the tree, but on a pairing of the two long branches. This mismatch also causes MP to reconstruct a Farris tree for alignments evolved under a Felsenstein tree in the Felsenstein zone [15].

The combinations of $E_2 = [\{1234, 13|24, 2|134, 4|123\}]$ include two $w|xyz$-patterns isolating those taxa that are paired in the $wx|yz$-pattern of the combination. Thus, not all combinations of two $w|xyz$-patterns and one $wx|yz$ are included in E_2. Still the combination of the most frequent $w|xyz$- and $wx|yz$-patterns of Farris trees are in E_2 as their most frequent $w|xyz$-patterns isolate those taxa which are paired in their most frequent $wx|yz$-pattern (Fig. 6A,C). The same holds for Felsenstein trees in the Felsenstein zone (Fig. 6E,G, grey region). However, unlike for Farris trees, for Felsenstein trees in the Felsenstein zone the most frequent $wx|yz$-pattern does not reflect the taxa pairing in the tree (Fig. 6E, grey region).

For Felsenstein trees outside the Felsenstein zone the two most frequent $w|xyz$-patterns isolate taxa which are separated in the most frequent $wx|yz$-pattern (Fig. 6E, white region). Thus, the combination of the two most frequent $w|xyz$-patterns and the most frequent $wx|yz$-pattern is not in E_2.

Instead, for most Felsenstein trees outside the Felsenstein zone $argmax(E_2)$ combines the most frequent $wx|yz$-pattern, $ac|bd$, with one of the two most frequent $w|xyz$-patterns and one of the two less frequent $w|xyz$-patterns (Fig. 6E,G, white region). There are also Felsenstein trees outside but close to the Felsenstein zone whose $argmax(E_2)$ combines the most frequent $w|xyz$-pattern, $c|abd$ and $d|abc$, with $ab|cd$ which is less frequent than $ac|bd$.

Therefore, $argmax(E_2)$ reveals the tree structure only for Farris trees. For Felsenstein trees either the taxa pairing or the long branches are not correctly indicated.

The combinations in $E_3 = [\{1234, 13|24, 2|134, 3|124\}]$ include two $w|xyz$-patterns isolating those taxa that are separated in the $wx|yz$-pattern of the combination. Thus, the combination of the two most frequent $w|xyz$-patterns with the most frequent $wx|yz$-pattern of a Farris tree is not included in E_3. For Farris trees in the Felsenstein zone $argmax(E_3)$ combines the two most frequent $w|xyz$-patterns with the pattern $ac|bd$ pairing taxa which are separated in the Farris tree (Fig. 6A,D, grey region). The same holds for Farris trees outside but close to the Felsenstein zone. However, for most Farris trees outside the Felsenstein zone $argmax(E_3)$ combines the most frequent $wx|yz$-pattern, $ab|cd$, with one of the most frequent and one of the less frequent $w|xyz$-patterns (Fig. 6A,D, white region). In contrast, for Felsenstein trees outside the Felsenstein zone $argmax(E_3)$ combines the two most frequent $w|xyz$-patterns with the most frequent $wx|yz$-pattern which corresponds to the taxa pairing in the tree (Fig. 6E,H, white region). For Felsenstein trees in the Felsenstein zone $argmax(E_3)$ combines $ac|bd$, $c|abd$ and $d|abc$ (Fig. 6H, grey region). This reflects the tree structure although $ac|bd$ is not the most frequent $wx|yz$-pattern (Fig. 6E,H, grey region). Thus, $argmax(E_3)$ only reveals the tree structure for expected site-pattern frequencies of Felsenstein trees.

Altogether, we can infer the tree structure from $argmax(E_1)$ as long as the expected site-pattern frequencies are not of a Felsenstein tree in the Felsenstein zone.

From $argmax(E_2)$ and $argmax(E_3)$ one of the two tree-types each is inferred correctly. Thus, $argmax(E_2)$ and $argmax(E_3)$ do not help to distinguish the tree-types.

However, with the maximal products of E_1, E_2 and E_3 we can distinguish the tree-types outside the Felsenstein zone. For Farris trees $max(E_2) = max(E_1) > max(E_3)$ as $argmax(E_1)$ and $argmax(E_2)$ combine the most frequent $w|xyz$- and $wx|yz$-patterns, but $argmax(E_3)$ does not. The same principle applies for Felsenstein trees in the Felsenstein zone. For Felsenstein trees outside the Felsenstein zone $max(E_3) = max(E_1) > max(E_2)$ holds, as $argmax(E_2)$ does not combine the most frequent $w|xyz$- and $wx|yz$-patterns, but $argmax(E_1)$ and $argmax(E_3)$ do.

Thus,

$$\text{NaiveDiscerner}(\mathbf{f}) = \frac{max(E_2)}{max(E_1)} - \frac{max(E_3)}{max(E_1)}$$

$$\begin{cases} = 1 - \frac{max(E_3)}{max(E_1)} > 0 \text{ for Farris trees} \\ = 1 - \frac{max(E_3)}{max(E_1)} > 0 \text{ for Felsenstein trees in Fel. zone} \\ = \frac{max(E_2)}{max(E_1)} - 1 < 0 \text{ for Felsenstein trees outside Fel. zone.} \end{cases} \tag{5}$$

If we infer a Farris tree for expected site-pattern frequencies with NaiveDiscerner($\mathbf{f}$) ≥ 0 and a Felsenstein tree otherwise, then we infer the correct tree-type outside the Felsenstein zone. However, in the Felsenstein zone a Farris tree is inferred. Therefore, the tree-type inference with NaiveDiscerner($\mathbf{f}$) mimics MP.

To infer the correct tree-type of a Felsenstein tree in the Felsenstein zone, we need to reduce the value of NaiveDiscerner($\mathbf{f}$). We do this by involving the maximal products of E_4, E_5 and E_6: The term

$$\frac{max(E_5)}{max(E_4)} - \frac{max(E_6)}{max(E_4)}$$

is larger the more different the branch length parameters p and q are (see equation (4)). Thus, this difference is particularly large for trees in the Felsenstein zone.

When incorporating this correction in NaiveDiscerner (5) we get the function:

$$\begin{aligned} &\text{F-zoneRatio}(\mathbf{f}) \\ &= \text{NaiveDiscerner}(\mathbf{f}) - c\left(\frac{\max(E_5)}{\max(E_4)} - \frac{\max(E_6)}{\max(E_4)}\right) \\ &= \left(\frac{\max(E_2)}{\max(E_1)} - \frac{\max(E_3)}{\max(E_1)}\right) - c\left(\frac{\max(E_5)}{\max(E_4)} - \frac{\max(E_6)}{\max(E_4)}\right) \end{aligned} \tag{6}$$

with the coefficient $c > 0$. F-zoneRatio infers a Farris tree if F-zoneRatio($\mathbf{f}$) ≥ 0 and a Felsenstein tree otherwise.

The larger c the more trees are inferred as Felsenstein trees. We want c to be large enough to infer as many Felsenstein trees as possible correctly, but at the same time c needs to be small enough to not incorrectly infer Farris trees as Felsenstein trees. F-zoneRatio correctly infers all trees outside the Felsenstein zone if and only if $c \in (0, 0.9356)$; all Farris trees in the Felsenstein zone if only if $c \leq 0.9286$; and all Felsenstein trees in the Felsenstein zone if and only if $c > 1$ (Theorem 3.1, 3.2 and 3.3, online supplemental material).

Thus, there is no c such that F-zoneRatio (6) correctly infers the tree-type of all trees. However, with $c = 0.9286$ the tree-types of all Farris trees are inferred correctly as well as all Felsenstein trees outside the Felsenstein zone and many Felsenstein trees in the Felsenstein zone.

Therefore, we finally define

$$\begin{aligned} &\text{FarFelDiscerner}(\mathbf{f}) \\ &= \text{NaiveDiscerner}(\mathbf{f}) - 0.9286\left(\frac{\max(E_5)}{\max(E_4)} - \frac{\max(E_6)}{\max(E_4)}\right) \\ &= \frac{\max(E_2)}{\max(E_1)} - \frac{\max(E_3)}{\max(E_1)} - 0.9286\left(\frac{\max(E_5)}{\max(E_4)} - \frac{\max(E_6)}{\max(E_4)}\right) \end{aligned} \tag{7}$$

which infers a Farris tree if the FarFelDiscerner($\mathbf{f}$) ≥ 0 and a Felsenstein tree otherwise.

According to the previous analysis of the equivalence classes FarFelDiscerner bases its inference on whether the most frequent $wx|yz$-pattern pairs long branches with each other (Farris tree) or with a short branch (Felsenstein tree). However, in contrast to MP, FarFelDiscerner corrects this inference in case the long branches are much longer than the short branches.

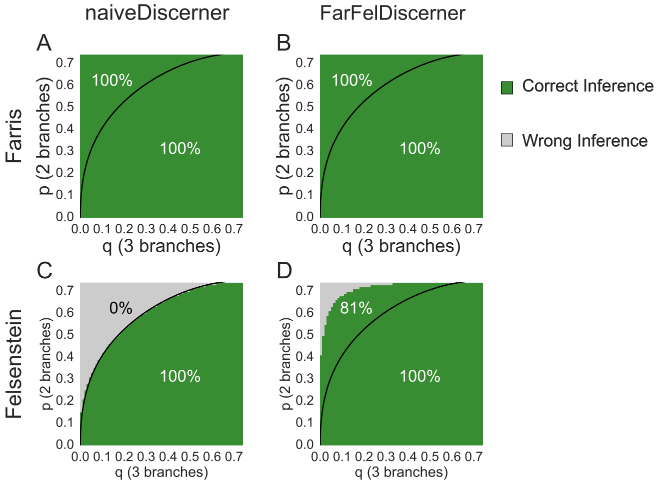


Fig. 7. Tree-type inference on expected site-pattern frequencies of Farris (A,B) and Felsenstein (C,D) trees with varying branch lengths. We infer a Farris tree if the first term of FarFelDiscerner, NaiveDiscerner, (A,C) or FarFelDiscerner itself (B,D) are ≥ 0 and a Felsenstein tree otherwise. The correctness of the inference is indicated by the colour. In each plot the region above the curve represents the Felsenstein zone. The values above and below the curve represent the average accuracy in and outside the Felsenstein zone, respectively.

V. Application of FarFelDiscerner

A. Expected Site-Pattern Frequencies

We numerically evaluated the accuracy of FarFelDiscerner on the expected site-pattern frequencies of Farris and Felsenstein trees whose branch length parameters p and q are independently varied between 0.01 and 0.74 with a step size of 0.01. A tree with $p = q$ is both a Farris and a Felsenstein tree (see Fig. 1 dashed lines). To account for this property, we also count the inference of a Felsenstein tree with $p = q$ as correct if FarFelDiscerner outputs 0 (output rounded to 10th decimal place).

FarFelDiscerner infers the correct tree-type for 97.5% of the expected site-pattern frequencies (see Table II). While the NaiveDiscerner (5) infers all Farris trees and Felsenstein trees outside the Felsenstein zone correctly, it is always wrong for Felsenstein trees in the Felsenstein zone (see Fig. 7A,C). FarFelDiscerner which is the sum of the NaiveDiscerner (5) and the correction term

$$-0.9286\left(\frac{\max(E_5)}{\max(E_4)} - \frac{\max(E_6)}{\max(E_4)}\right)$$

also infers all Farris trees and all Felsenstein trees outside the Felsenstein zone correctly (see Fig. 7B, D below the curve). Moreover, FarFelDiscerner infers 81.3% of Felsenstein trees in the Felsenstein zone correctly (see Fig. 7D above the curve).

As a result FarFelDiscerner achieves a 6% greater accuracy than the original ANN F-zoneNN [6] (see Tab. II) in the Felsenstein zone and on average. Therefore, the simple function FarFelDiscerner outperforms the more complex function F-zoneNN if we use the expected site-pattern frequencies as input.

TABLE II
Accuracy of FarFelDiscerner and F-zoneNN on expected site-pattern frequencies of all Farris and Felsenstein trees and those in the Felsenstein zone.

Method	Data	Farris Trees	Felsenstein Trees	Average Accuracy
FarFelDiscerner	All	100.0%	95.1%	97.5%
F-zoneNN	All	87.8%	94.2%	91.0%
FarFelDiscerner	Fel. zone	100.0%	81.3%	90.6%
F-zoneNN	Fel. zone	96.4%	71.0%	83.7%

TABLE III
Accuracy of FarFelDiscerner and F-zoneNN on all alignments of the test data stemming from Farris or Felsenstein trees and on those alignments of trees in the Felsenstein zone.

Method	Data	Farris Trees	Felsenstein Trees	Average Accuracy
FarFelDiscerner	All	78.9%	84.4%	81.7%
F-zoneNN	All	83.0%	91.8%	87.4%
FarFelDiscerner	Fel. zone	72.7%	70.9%	71.8%
F-zoneNN	Fel. zone	86.6%	68.6%	77.6%

B. Simulated Test Data

We also evaluated FarFelDiscerner on the test alignments of F-zoneNN [6]. These test alignments were simulated under Farris and Felsenstein trees with various branch lengths such that the correct tree type is known. Each of the alignments has a length of 1,000 base-pairs (bp), i.e. has 1,000 sites.

Whereas FarFelDiscerner performs better than F-zoneNN on the expected site-pattern frequencies, this is not the case on the test alignments of F-zoneNN [6]. On these alignments F-zoneNN achieves a 6% larger accuracy than FarFelDiscerner (see Table III).

The reason is F-zoneNN's more complex structure in comparison to FarFelDiscerner and that it was already trained on alignments of finite length. While F-zoneNN learned to distinguish noise and true signal, FarFelDiscerner relies on certain properties of expected site-pattern frequencies which are not always true for finite alignments. For example, $f(w|xyz) > f(x|wyz)$ always holds for expected site-pattern frequencies of a tree whose branch leading to w is longer than that leading to x. This is not necessarily the case for finite alignments of such a tree as the noise can influence the site-pattern frequencies such that $f(w|xyz) < f(x|wyz)$. Nevertheless, FarFelDiscerner achieves an average accuracy of more than 80% on the simulated test alignments (Table III).

FarFelDiscerner infers the type of tree under which an alignment evolved, it is not able to reconstruct a tree. However, we can involve FarFelDiscerner in the tree reconstruction by defining a Mixed Strategy: If FarFelDiscerner infers a Farris tree for an alignment, we use MP for the tree reconstruction and if FarFelDiscerner infers a Felsenstein tree, we use ML.

We compared the accuracies of the Mixed Strategy using FarFelDiscerner with several tree reconstruction methods on the test alignments of length 1,000 bp (see Table IV). These tree reconstruction methods are the four methods evaluated in Leuchtenberger et al. [6]: the classical reconstruction methods ML and MP, the Mixed Strategy using F-zoneNN (which works analogously to the Mixed Strategy using FarFelDis-

TABLE IV
Accuracy of the Mixed Strategy (Mix) using FarFelDiscerner, that using F-zoneNN, of nogap300k [5], ML and MP on the test data. The accuracy is given for alignments stemming from Farris, from Felsenstein trees and for all alignments. For each set-up also the accuracy of the alignments stemming from trees in the Felsenstein zone is given.

Method	Data	Farris Trees	Felsenstein Trees	Average Accuracy
Mix FarFelDiscerner	All	91.7%	84.9%	88.3%
Mix F-zoneNN	All	94.0%	84.9%	89.4%
nogap300k	All	97.5%	81.1%	89.3%
ML	All	80.1%	87.0%	83.6%
MP	All	97.5%	75.0%	86.2%
Mix FarFelDiscerner	Fel. zone	86.7%	59.1%	72.9%
Mix F-zoneNN	Fel. zone	90.1%	59.5%	74.8%
nogap300k	Fel. zone	98.5%	33.1%	65.8%
ML	Fel. zone	67.9%	71.2%	69.6%
MP	Fel. zone	99.8%	8.0%	53.9%

TABLE V
Accuracy of FarFelDiscerner and StrepsipteraNN on the test data of StrepsipteraNN.

Method	Farris Trees	Felsenstein Trees	Average Accuracy
FarFelDiscerner	78.5%	91.5%	85.0%
StrepsipteraNN	90.9%	87.3%	89.1%

cerner) as well as nogap300k (an ANN trained by Suvorov et al. [5] to infer quartet trees).

In comparison to the Mixed Strategy using F-zoneNN, the accuracy of the Mixed Strategy using FarFelDiscerner is 1% lower on average and 2% lower in the Felsenstein zone (Table IV). Thus, the tree reconstruction accuracies of the Mixed Strategies are more similar to each other than the tree-type inference accuracies of F-zoneNN and FarFelDiscerner are similar.

The ANN of Suvorov et al. [5], nogap300k, performs on average 1% better than the Mixed Strategy of FarFelDiscerner, but 7% worse in the Felsenstein zone (Table IV). The accuracies of ML and MP are, respectively, 2% and 5% lower than the accuracy of the Mixed Strategy using FarFelDiscerner (Table IV).

Taken together, the Mixed Strategy using the relatively simple function FarFelDiscerner can keep up with the black-box-like ANNs on alignments evolved under the Jukes-Cantor model [23] and outperforms the classical tree reconstruction methods ML and MP.

Previously, we evaluated FarFelDiscerner on alignments evolved on trees with two branch length parameters under the Jukes-Cantor model [23]. To evaluate FarFelDiscerner's performance under more general circumstances, we next apply it to resolve the placement of Strepsiptera with respect to flies and beetles (see Fig. 2).

C. Empirical Holometabola Alignments

1) Background: The placement of Strepsiptera with respect to flies and beetles (see Fig. 2) was one of the earliest discussed cases of LBA [21], [22]. The original sequence data gathered by Carmean and Crespi [21] consists of 18S ribosomal DNA sequences of Strepsiptera, Coleoptera (beetles), Diptera (flies) and other Holometabola. From these Holometabola sequences 24 quartet-alignments were formed i.e. alignments of four sequences: Each quartet-alignment includes the Strepsiptera sequence, one of two beetle sequences, one of two fly sequences and one of six sequences of other Holometabola. In contrast to other sequences considered in this paper, the Holometabola sequences most likely didn't evolve under the Jukes-Cantor model [23] or on a tree with only two different branch lengths.

To resolve the placement of Strepsiptera, Leuchtenberger et al. [6] trained the ANN, StrepsipteraNN, on simulated alignments resembling the original 24 quartet-alignments. The input of StrepsipteraNN are the frequencies of all 256 site-patterns instead of the 15 collapsed site-pattern frequencies, as it can not be assumed that the Holometabola sequences evolved under the Jukes-Cantor model [23]. The greater complexity of the input data also affects the total number of parameters in StrepsipteraNN. With around 5.9 million parameters, StrepsipteraNN has more than four times as many parameters as F-zoneNN.

2) Evaluation of FarFelDiscerner on Holometabola Alignments: We identified the equivalence classes of FarFelDiscerner by simplifying F-zoneNN, an ANN trained and designed on alignments evolved under a Jukes-Cantor model [23] and on trees with only two different branch lengths. However, FarFelDiscerner's decision process is based on properties of the expected site-pattern frequencies which are mostly not specific to those assumptions. We therefore applied FarFelDiscerner without further adaption to the test data of StrepsipteraNN i.e. the alignments simulated to resemble the Holometabola sequences gathered by Carmean and Crespi [21].

On this test data FarFelDiscerner performs on average with an accuracy of 85% only 4% worse than StrepsipteraNN (Table V). This is especially striking as StrepsipteraNN (in contrast to FarFelDiscerner) was specifically trained to resolve the placement of Strepsiptera.

As FarFelDiscerner successfully distinguishes between Farris and Felsenstein trees based on simulated alignments, we also applied it to the 24 empirical quartet alignments of Holometabola sequences.

For all 24 quartets FarFelDiscerner < 0 and thus infers a Felsenstein tree. This is only due to the correction term

$$-0.9286\left(\frac{\max(E_5)}{\max(E_4)}-\frac{\max(E_6)}{\max(E_4)}\right)$$

of FarFelDiscerner. The other part of FarFelDiscerner, the NaiveDiscerner (5), is greater than 0 as the most frequent $wx|yz$-pattern of all quartets induces a pairing of the long-branched taxa, Strepsiptera and flies. Therefore, FarFelDiscerner suggests that all quartet alignments stem from Felsenstein trees in the Felsenstein zone and places Strepsiptera with the beetles and distant to the flies (cf. Fig. 2B).

This result is concordant with that of StrepsipteraNN (inferred a Felsenstein tree for 99.7% of the quartets [6]) as well as with the current scientific view (see [29], [30]). Altogether, this implies that FarFelDiscerner is not specific for the Jukes-Cantor model [23] or for trees with only two different branch lengths, but that it can also be applied to alignments evolved under a more complex model of evolution and/or on trees with multiple different branch lengths.

VI. Conclusion

In this study we constructed the simple function FarFelDiscerner (7) which distinguishes Farris from Felsenstein trees. Starting with the complex ANN F-zoneNN [6] we iteratively reconstructed a simple function FarFelDiscerner. Based on F-zoneNN we reduced the problem to a fourth-degree polynomial. However, without F-zoneNN we would never have had the idea that a fourth-degree polynomial can be used to distinguish Farris and Felsenstein trees. Only the desire to simplify the structure of F-zoneNN to the point where the underlying function is human understandable led to the construction of the fourth-degree polynomial F-zonePoly (3). And without the polynomial structure of F-zonePoly, in turn, we would not have identified the equivalence classes of site-pattern combinations underlying FarFelDiscerner. Thus, only the simplification of F-zoneNN enabled us to construct the simple function FarFelDiscerner.

This simple function is essentially using six equivalence classes and their maximal products to relate $w|xyz$- and $wx|yz$-patterns and thus which branches are long and which are paired. This way FarFelDiscerner can distinguish between alignments stemming from trees pairing the long branches (Farris trees) and those pairing long with rather short branches (Felsenstein trees).

On the expected site-pattern frequencies FarFelDiscerner achieves an almost perfect accuracy in distinguishing the tree types. Moreover, FarFelDiscerner can also distinguish Farris and Felsenstein trees based on finite alignments simulated under the Jukes-Cantor model [23]. A Mixed Strategy choosing the tree reconstruction method based on FarFelDiscerner's output is even more accurate than ML or MP and similar to the significantly more complex ANNs F-zoneNN and nogap300k.

Further, we demonstrated that the mechanism of FarFelDiscerner works also for the empirical Holometabola alignments whose sequences evolved naturally rather than according to a theoretical substitution model.

Therefore, we showed that the construction, training and subsequent simplification of an ANN led to a simple function performing the same task as the ANN but whose decisions are understandable.

We are aware of the fact, that F-zoneNN is a rather simple feed-forward neural network acting on data with specific properties. The simplification process is not directly transferable to other ANNs. Nevertheless, based on our results, we believe that a simplification can also work for other ANNs. A simplification is especially promising if there is a lot of knowledge available regarding the ANN task and the structure of the input data. As shown in our case a successful simplification, even if laborious, is worthwhile as it can provide valuable new insights. Without using F-zoneNN as a starting point we would not have identified FarFelDiscerner.

Acknowledgments

This work was supported by the Austrian Science Fund (FWF DOC 32-B28). A.v.H. also thanks the Medical University of Vienna and the University of Vienna for their support. We thank Christiane Elgert and Stephen Crotty for helpful discussions and comments. The authors declare they have no competing interests.

References

[1] N. Sapoval, A. Aghazadeh, M. G. Nute, D. A. Antunes, A. Balaji, R. Baraniuk, C. J. Barberan, R. Dannenfelser, C. Dun, M. Edrisi, R. A. L. Elworth, B. Kille, A. Kyrillidis, L. Nakhleh, C. R. Wolfe, Z. Yan, V. Yao, and T. J. Treangen, "Current progress and open challenges for applying deep learning across the biosciences," *Nature Communications*, vol. 13, no. 1, p. 1728, 2022.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org.

[3] M. A. Nielsen, *Neural Networks and Deep Learning*. Determination Press, 2015, neuralnetworksanddeeplearning.com.

[4] Z. Zou, H. Zhang, Y. Guan, and J. Zhang, "Deep residual neural networks resolve quartet molecular phylogenies," *Mol. Biol. Evol.*, vol. 37, no. 5, pp. 1495–1507, 2020.

[5] A. Suvorov, J. Hochuli, and D. R. Schrider, "Accurate inference of tree topologies from multiple sequence alignments using deep learning," *Syst. Biol.*, vol. 69, no. 2, pp. 221–233, 2020.

[6] A. F. Leuchtenberger, S. M. Crotty, T. Drucks, H. A. Schmidt, S. Burgstaller-Muehlbacher, and A. von Haeseler, "Distinguishing felsenstein zone from farris zone using neural networks," *Mol. Biol. Evol.*, vol. 37, no. 12, pp. 3632–3641, 2020.

[7] C. R. Solís-Lemus, S.-A. Yang, and L. Zepeda-Núñez, "Accurate phylogenetic inference with a symmetry-preserving neural network model," 2023, arXiv:2201.04663.

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

[9] S. Burgstaller-Muehlbacher, S. M. Crotty, H. A. Schmidt, F. Reden, T. Drucks, and A. von Haeseler, "Modelrevelator: Fast phylogenetic model estimation via deep learning," *Molecular Phylogenetics and Evolution*, vol. 188, p. 107905, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1055790323002051

[10] M. L. Smith and M. W. Hahn, "Phylogenetic inference using generative adversarial networks," *Bioinformatics*, vol. 39, no. 9, p. btad543, 09 2023. [Online]. Available: https://doi.org/10.1093/bioinformatics/btad543

[11] M. N. Price, P. S. Dehal, and A. P. Arkin, "Fasttree: Computing large minimum evolution trees with profiles instead of a distance matrix," *Molecular Biology and Evolution*, vol. 26, no. 7, pp. 1641–1650, 2009.

[12] A. Stamatakis, "Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies," *Bioinformatics*, vol. 30, no. 9, pp. 1312–1313, 2014.

[13] L.-T. Nguyen, H. A. Schmidt, A. von Haeseler, and B. Q. Minh, "IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies," *Mol. Biol. Evol.*, vol. 32, no. 1, pp. 268–274, 2015.

[14] S. Kalyaanamoorthy, B. Q. Minh, T. K. F. Wong, A. von Haeseler, and L. S. Jermiin, "ModelFinder: fast model selection for accurate phylogenetic estimates," *Nat. Methods*, vol. 14, no. 6, pp. 587–589, 2017.

[15] J. Felsenstein, "Cases in which parsimony or compatibility methods will be positively misleading," *Syst. Biol.*, vol. 27, no. 4, pp. 401–410, 1978.

[16] M. D. Hendy and D. Penny, "A framework for the quantitative study of evolutionary trees," *Syst. Zool.*, vol. 38, no. 4, p. 297, 1989.

[17] J. P. Huelsenbeck and D. M. Hillis, "Success of phylogenetic methods in the four-taxon case," *Syst. Biol.*, vol. 42, no. 3, pp. 247–264, 1993.

[18] P. J. Waddell, "Statistical methods of phylogenetic analysis: including hadamard conjugations, logdet transforms and maximum likelihood," Ph.D. dissertation, Massey University, Palmerston North, New Zealand, 1995.

[19] M. E. Siddall, "Success of parsimony in the four-taxon case: Long-Branch repulsion by likelihood in the farris zone," *Cladistics*, vol. 14, no. 3, pp. 209–220, 1998.

[20] Z. Yang, "Statistical properties of the maximum likelihood method of phylogenetic estimation and comparison with distance matrix methods," *Syst. Biol.*, vol. 43, no. 3, pp. 329–342, 1994.

[21] D. Carmean and B. J. Crespi, “Do long branches attract flies?” *Nature*, vol. 373, no. 6516, p. 666, 1995.

[22] J. P. Huelsenbeck, “Is the felsenstein zone a fly trap?” *Syst. Biol.*, vol. 46, no. 1, pp. 69–74, 1997.

[23] T. H. Jukes and C. R. Cantor, “Evolution of protein molecules,” in *Mammalian Protein Metabolism*, H. N. Munro, Ed. New York, USA: Academic Press, 1969, pp. 21–132.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, “Why should I trust you?: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, San Francisco, CA, USA, 2016, pp. 1135–1144.

[25] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[26] D. Erhan, Y. Bengio, A. Courville, and P. Vincent, “Visualizing higher-layer features of a deep network,” *Technical Report, Univeristé de Montréal*, vol. 1341, 2009.

[27] N. I. Vilenkin, *Combinatorics*. New York, USA: Academic Press, 1971.

[28] J. Felsenstein, *Inferring phylogenies*. Sunderland, MA, USA: Sinauer Associates, Inc., 2004.

[29] O. Niehuis, G. Hartig, S. Grath, H. Pohl, J. Lehmann, H. Tafer, A. Donath, V. Krauss, C. Eisenhardt, J. Hertel, M. Petersen, C. Mayer, K. Meusemann, R. S. Peters, P. F. Stadler, R. G. Beutel, E. Bornberg-Bauer, D. D. McKenna, and B. Misof, “Genomic and morphological evidence converge to resolve the enigma of strepsiptera,” *Curr. Biol.*, vol. 22, no. 14, pp. 1309–1313, 2012.

[30] B. Boussau, Z. Walton, J. A. Delgado, F. Collantes, L. Beani, I. J. Stewart, S. A. Cameron, J. B. Whitfield, J. S. Johnston, P. W. H. Holland, D. Bachtrog, J. Kathirithamby, and J. P. Huelsenbeck, “Strepsiptera, phylogenomics and the long branch attraction problem,” *PLoS One*, vol. 9, no. 10, pp. 1–9, 2014.

Alina F. Leuchtenberger received the M.Sc. degree in mathematics from the University of Vienna, Austria, in 2018. She is working toward the PhD degree at the Center for Integrative Bioinformatics Vienna (CIBIV), Max Perutz Labs, University of Vienna and Medical University of Vienna. Her research is focused on Artificial Neural Networks in Phylogenetics and their interpretability.

Arndt von Haeseler studied biology and mathematics and received the PhD degree in mathematics at the University of Bielefeld. Then, he did his postdoc at the University of Southern California (Los Angeles) with Mike Waterman. Subsequently, he was an assistant professor at the University of Munich, an associate professor at the Max Planck Institute for evolutionary anthropology, and a full professor for bioinformatics at the University of Düsseldorf. He was appointed scientific director of the Center for Integrative Bioinformatics Vienna (CIBIV) in September 2005 and professor for bioinformatics. His research interests are phylogenetic tree reconstruction, modeling evolution, population genetics, algorithms for bioinformatics, and biodiversity.