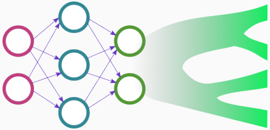
PhyloFormer and phylogenetic reconstruction with deep neural networks



Luca Nesterenko[*], **Luc Blassel**[*], Philippe Veber, Bastien Boussau[†], Laurent Jacob[†]

LISN Bioinfo Seminar - November 21st, 2024

Preamble - Who am I?

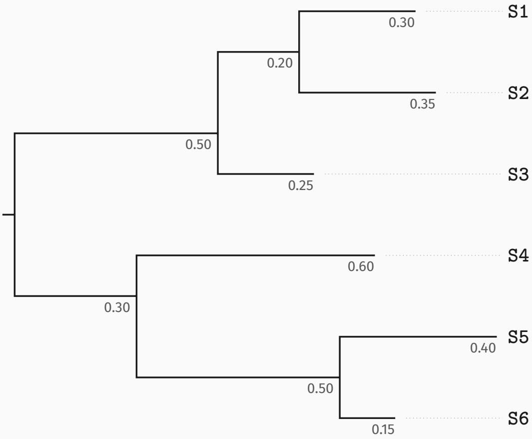
- Studied **biology** at AgroParisTech and **machine-learning** at Dauphine
- Did my **PhD.** at Institut Pasteur 2020-2022, working on:
    1. **Drug resistance** detection in **HIV** with O. Gascuel
    2. Improving long **read-mapping** with R. Chikhi
- Since March 2023, post-doc with Laurent Jacob at LCQB:
  Deep-learning for phylogenetic inference

Blassel, Zhukova, et al. 2021; Blassel, Tostevin, et al. 2021; Blassel, Medvedev, et al. 2022

What is Phylogenetic Inference ?

Context - Phylogenetic inference

S1 TQPRLPTC
S2 TQPSVPKC
S3 TGVPVPAC
S4 TLGRSPSC
S5 TQPRAKTC
S6 TQPRIPAC



Goal: describe **evolutionary-history** of MSA

Context - Why do phylgenetic inference ?

Phylogenetic inference is a **base-task** essential in many **downstream** analyses:

Epidemiology:	Track viral spread and evolution
Virology:	Identify recombination events
Biochemistry:	Identify functional constraints on proteins
Ecology:	Characterize biodiversity
...	

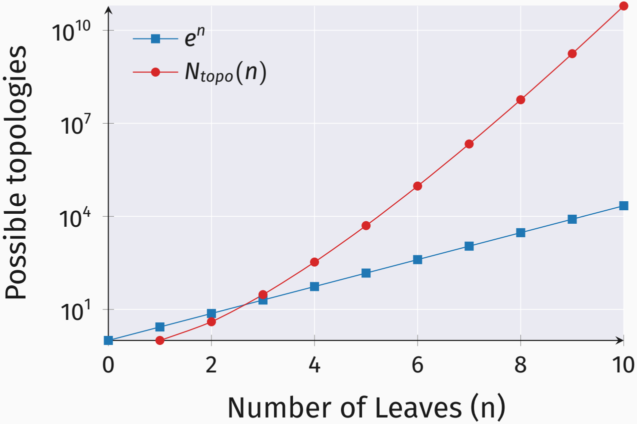
Hadfield et al. 2018; Nelson et al. 2008; Harms and Thornton 2013; Perez-Lamarque et al. 2022

Context - The problem with phylogenetic inference

1. Phylogenies are **hard**!
2. **Super-exponential** tree space

$$N_{topo}(n) = \frac{(2n)!}{(n+1)!}$$

Felsenstein 2004



Context - Likelihood-based tree reconstruction (1)

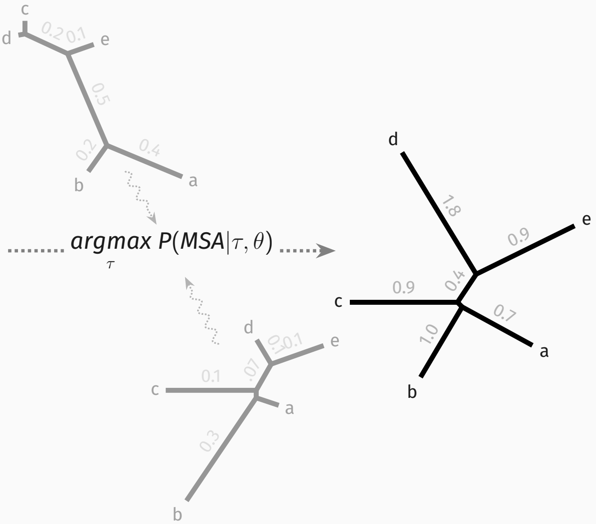
a TQPRLPTC

b TQPSVPKC

c TGVPVPAC

d TLGRSPSC

e TQPRAKTC



$$\underset{\tau}{\mathit{argmax}}\ P(MSA|\tau,\theta)$$

$\tau =$ Phylogeny

$\theta =$ Model

Context - Likelihood-based tree reconstruction (2)

Pros:

- These methods are **accurate**
- The **whole MSA** is considered in $P(MSA|\tau, \theta)$
- With **Bayesian** methods you quantify **uncertainty**

Cons:

- These methods are **slow**
  1. **Computing** the likelihood is **costly**
  2. We have to **explore** the tree-space with **topological** moves
- We are **limited** to models where $P(MSA|\tau, \theta)$ is **computeable**

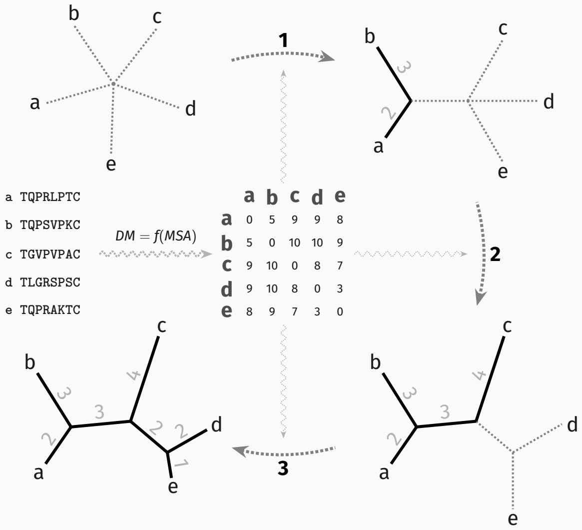
Felsenstein 1993; Kleinman et al. 2010

Context - Distance-based tree reconstruction (1)

$$\tau = f(DM)$$

$$DM = \{d(i,j), (i,j) \in MSA\}$$

we **choose** *d*



a TQPRLPTC

b TQPSVPKC

c TGVPVPAC

d TLGRSPSC

e TQPRAKTC

$DM = f(MSA)$

	a	b	c	d	e
a	0	5	9	9	8
b	5	0	10	10	9
c	9	10	0	8	7
d	9	10	8	0	3
e	8	9	7	3	0

Context - Distance-based tree reconstruction (2)

Pros:

- These methods are **fast**: $\mathcal{O}(n^2)$ to $\mathcal{O}(n^3)$
- These methods are **statistically consistent**
- **Guaranteed** to infer the **true** tree if $Err(DM) \leq \varepsilon$
- Many **variants**: NJ, BioNJ, FastME, ...

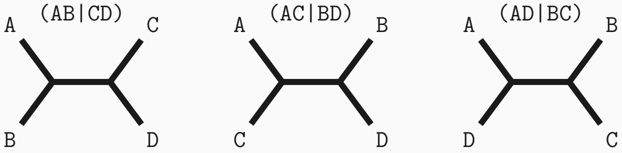
Cons:

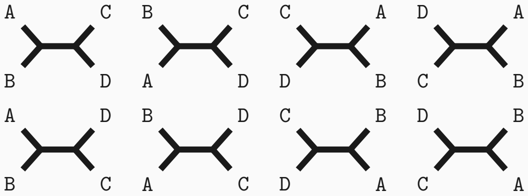
- These methods are **innacurate** compared to **ML**
- We **ignore** information when computing $d(i,j), (i,j) \in MSA$

Gascuel and Steel 2016; Guindon and Gascuel 2003; Saitou and Nei 1987; Gascuel 1997; Lefort et al. 2015

Machine Learning for Phylogenetic Inference

Background - Quartet methods





adapted from Tang et al. 2024

- Unrooted **quartet** has **3 unique** topologies
- An n tree **topology** is **uniquely** represented by its set of ${}_nC_4$ **quartets**
- **Likelihood-based** methods exist to infer **trees** from **quartets**

Bandelt and Dress 1986; Strimmer and Von Haeseler 1996

Background - Quartet classifier networks

Classifier that predicts the **quartet topology** given a set of **4 sequences**

Pros:

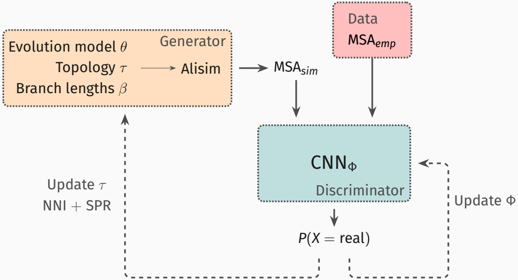
- **Adaptable** to different tree sizes
- **Simple** training and **loss**
- Deal with **equivalent** quartets:
    - Data **augmentation**
    - Network **architecture**

Cons:

- **Scaling**: $Quartets(n) = \binom{n}{4}$
- **Poor** performance in settings with **long branches** and **short sequences**

Suvorov et al. 2019; Zou et al. 2020; Tang et al. 2024; Zaharias et al. 2022

Background - GANs



Smith and Hahn 2023

- Given a **real MSA** M
- **Generator** builds a tree T and **simulates an MSA** M'
- **Discriminator** learns to differentiate M and M' with **CNN**
- **Train** D with **backprop** and G with **topological moves** on tree
- Need to **train** model for **each** inferred **tree**

Background - Learning to explore tree-space

Topology search

- **Exploring** the whole **topology** space is too **expensive**
- **Heuristic** topological **moves**:

SPR	Subtree Prune Regraft
NNI	Nearest Neighbour Interchange

- At **each step** select moves by **best** $\mathcal{L}$

Azouri, Granit, et al. 2024; Azouri, Abadi, et al. 2021

Background - Learning to explore tree-space

Topology search

- **Exploring** the whole **topology** space is too **expensive**
- **Heuristic** topological **moves**:

SPR	Subtree Prune Regraft
NNI	Nearest Neighbour Interchange

- At **each step** select moves by **best** $\mathcal{L}$

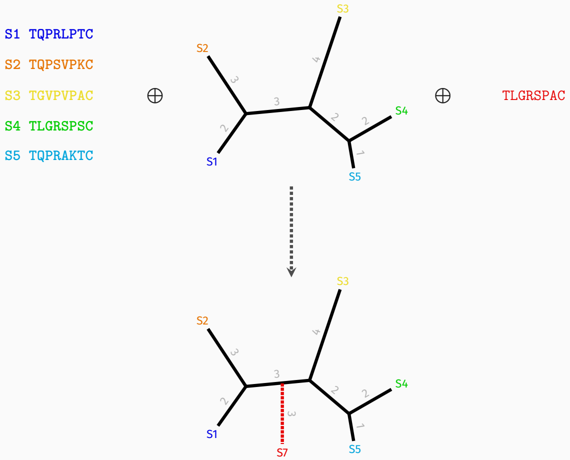
ML-guided topology search

- Train models to **predict** the next **best move**
- **Rank SPRs** by predicted likelihood with **RF regressor**
- This was **also** done with **NNs** in an **RL** setting
- Once trained **speeds** up **likelihood** methods

Azouri, Granit, et al. 2024; Azouri, Abadi, et al. 2021

Background - Phylogenetic placement

- Given a **tree**,an **MSA** and a **sequence**,
  what is the **best spot** to add a **new branch** ?
- **ML** and **distance** based **methods** for placement
- Better **scaling** than full **tree search**



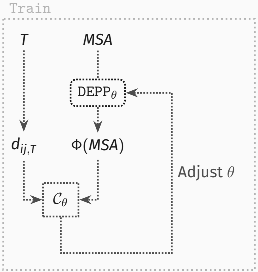
Price et al. 2010; Nguyen et al. 2015; Lefort et al. 2015

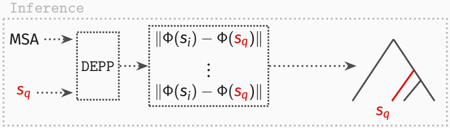
Background - DEPP

- With a **Backbone tree** T and **MSA**
- **Minimize** $\mathcal{C}$ w.r.t Φ :

$$\mathcal{C} = \sum_{(i,j)} \frac{1}{d_{ij,T}} \left(\|\Phi(s_i) - \Phi(s_j)\|_2 - \sqrt{d_{ij,T}} \right)^2$$

- **Embed** new **sequence** $\Phi(s_{new})$
- Get **distances** from $\Phi(s_{new})$ to others and place it
- Use **distance-based** placement method `APPLES`





Balaban et al. 2020; Jiang et al. 2022

Background - DEPP the good and the bad

The good:

- Small **simple** architecture: $\texttt{Conv} \times 3 + \texttt{FCN}$
- **Easy** to **train**
- **Scales** to backbones of $\approx 10^4$ tips
- **Successful** application on **microbial** tree-of-life

The less good:

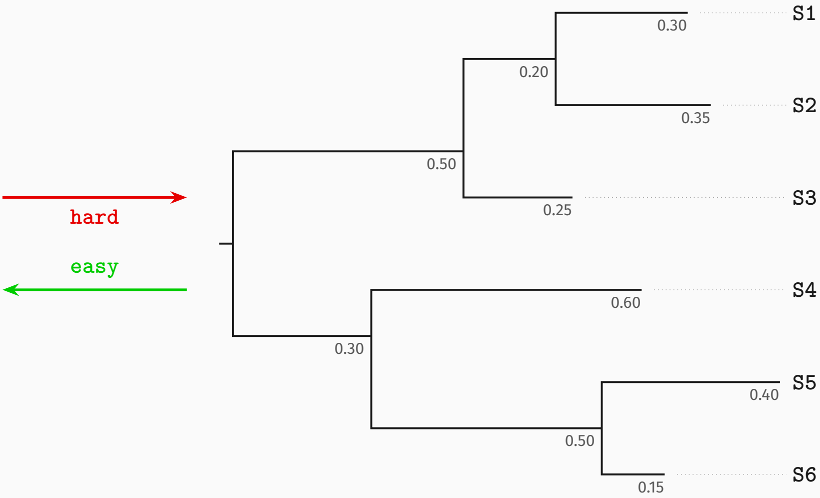
- This is a **simpler** problem that what **we want** to solve
- Need to **train** for **every** backbone **tree** and **MSA**

Jiang et al. 2022

Likelihood-free phylogenetic inference with PhyloFormer

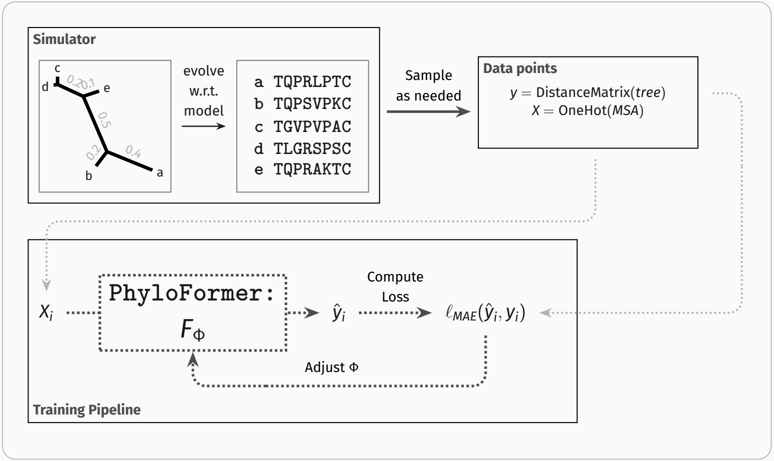
Method - Likelihood-free inference, motivation

```
S1 TQPRLPTC
S2 TQPSVPKC
S3 TGVPVPAC
S4 TLGRSPSC
S5 TQPRAKTC
S6 TQPRIPAC
```



Method - Amortized likelihood-free inference

Training time



Simulator

evolve w.r.t. model

a	`TQPRLPTC`
b	`TQPSVPKC`
c	`TGVPVPAC`
d	`TLGRSPSC`
e	`TQPRAKTC`

Sample as needed

Data points

$y = \text{DistanceMatrix}(tree)$

$X = \text{OneHot}(MSA)$

X_i

`PhyloFormer:` F_Φ

$\hat{y}_i$

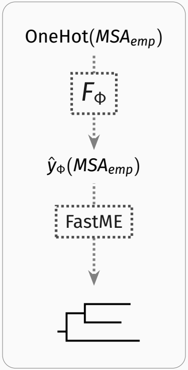
Compute Loss

$\ell_{MAE}(\hat{y}_i, y_i)$

Adjust Φ

Training Pipeline

Inference time



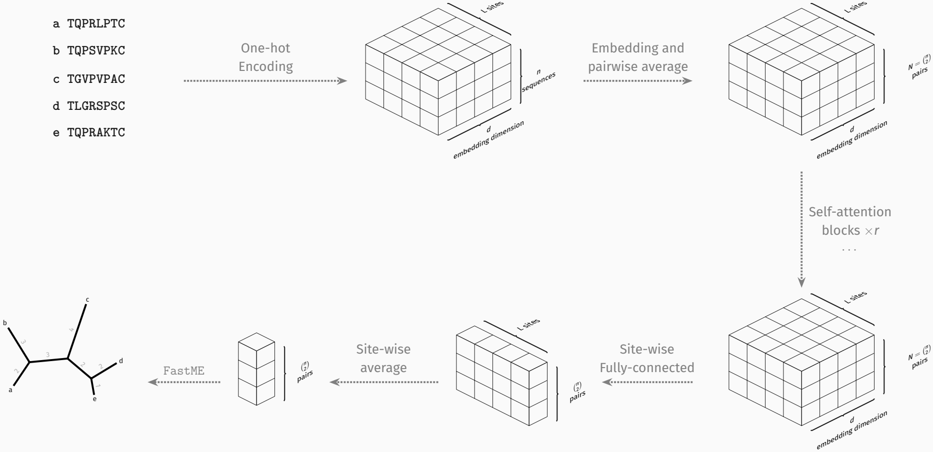
$\text{OneHot}(MSA_{emp})$

F_Φ

$\hat{y}_\Phi(MSA_{emp})$

FastME

Method - PhyloFormer overview



a TQPRLPTC

b TQPSVPKC

c TGVPVPAC

d TLGRSPSC

e TQPRAKTC

One-hot Encoding

Embedding and pairwise average

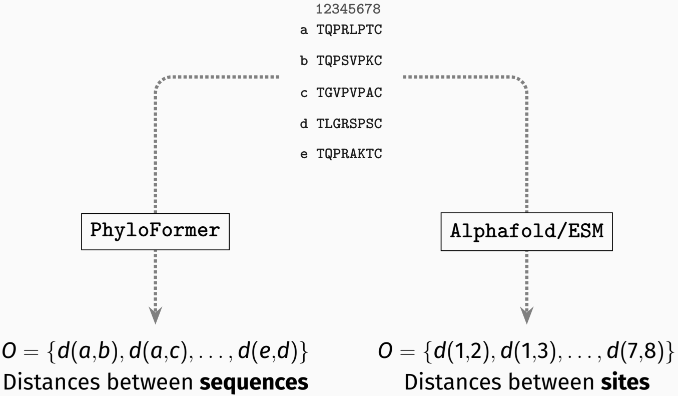
Self-attention blocks $\times r$ ...

Site-wise Fully-connected

Site-wise average

FastME

Method - Similarity with structure prediction



```
  12345678
a TQPRLPTC
b TQPSVPKC
c TGVPVPAC
d TLGRSPSC
e TQPRAKTC
```

PhyloFormer

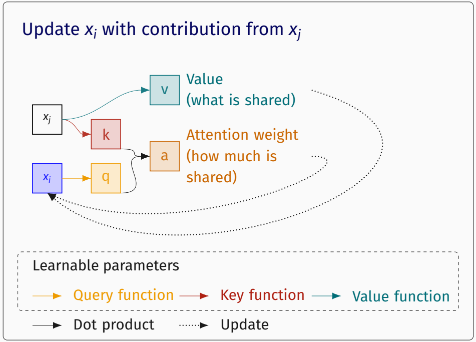
Alphafold/ESM

$O = \{d(a,b), d(a,c), \ldots, d(e,d)\}$
Distances between **sequences**

$O = \{d(1,2), d(1,3), \ldots, d(7,8)\}$
Distances between **sites**

Jumper et al. 2021; Rao et al. 2021

Method - Why self-attention



Update x_i with contribution from x_j

Value
(what is shared)

Attention weight
(how much is shared)

Learnable parameters

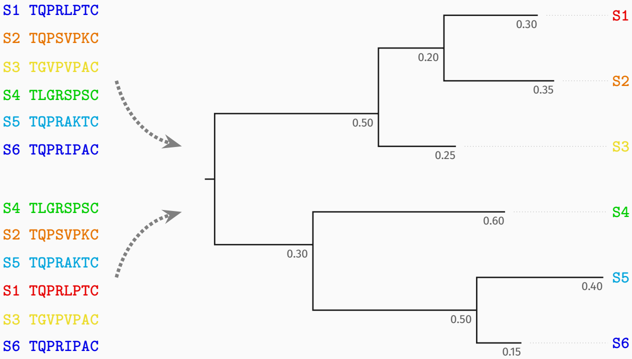
Query function
Key function
Value function

Dot product
Update

- Represents **elements** in a set as a **weighted sum** of all **elements** *(including itself)*
- Parametrized by **learnable weights**
- Yields a **context-aware** and **learnable** representation
- Applies to sets **regardless of cardinality**

Vaswani et al. 2017

Method - Ensuring invariance & equivariance



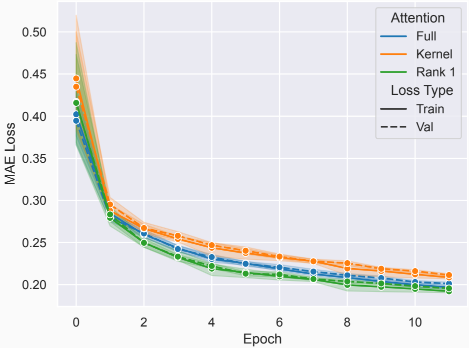
```
S1 TQPRLPTC
S2 TQPSVPKC
S3 TGVPVPAC
S4 TLGRSPSC
S5 TQPRAKTC
S6 TQPRIPAC
```

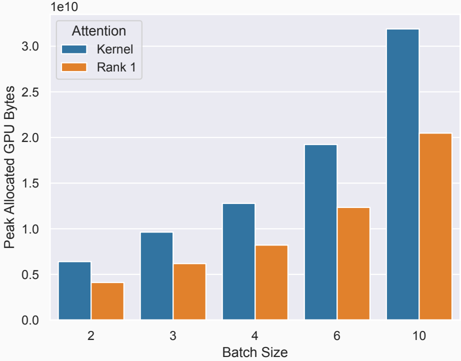
```
S4 TLGRSPSC
S2 TQPSVPKC
S5 TQPRAKTC
S1 TQPRLPTC
S3 TGVPVPAC
S6 TQPRIPAC
```

Self-Attention is already permutation **equivariant**!

Site-wise average ensures **invariance** w.r.t. sites.

Method - Custom Rank 1 attention is better

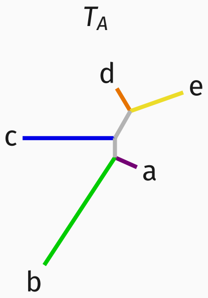




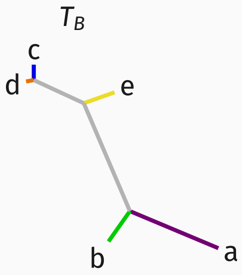
Vaswani et al. 2017; Katharopoulos et al. 2020

How does PhyloFormer perform ?

Results - How do we measure performance ?



$$A = \{(a|bcde), (b|acde), \dots\}$$
$$B = \{(a|bcde), (b|acde), \dots\}$$

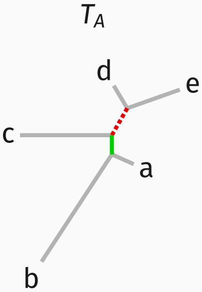


$$RF_{\text{norm}}(T_A, T_B) = (|A| + |B|)^{-1}(|A \cup B| - |A \cap B|)$$

$$KF(T_A, T_B)^2 = \sum_{e \in A \cap B} (w_{(e,A)} - w_{(e,B)})^2 + \sum_{e \in A \setminus B} w^2_{(e,A)} + \sum_{e \in B \setminus A} w^2_{(e,B)}$$

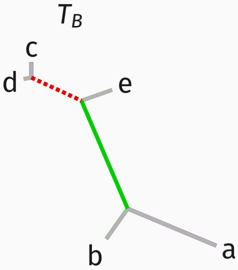
$$\ell_{MAE}(T_A, T_B) = {}_nC_2{}^{-1} \sum_{\{i,j\} \in T_A} |d(i,j,T_a) - d(i,j,T_B)|$$

Results - How do we measure performance ?



$$A = \{(ab|cde), (de|abc)\}$$

$$B = \{(ab|cde), (dc|abe)\}$$

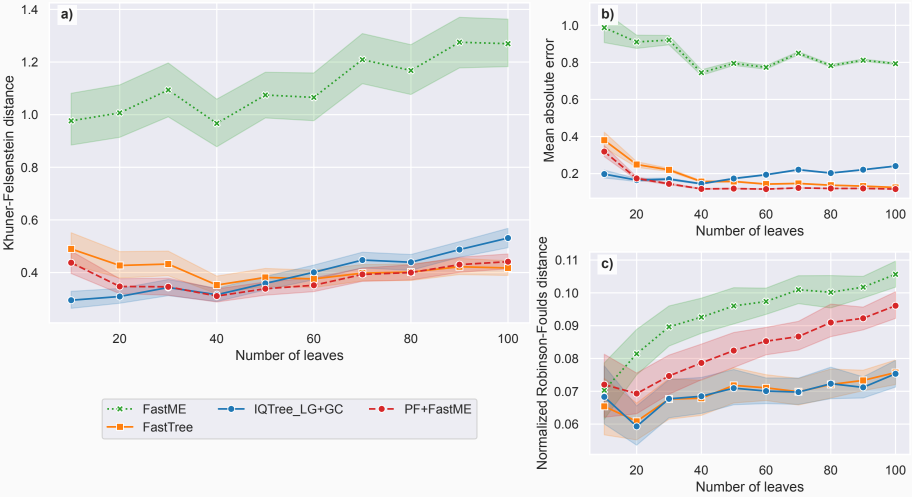


$$RF_{\text{norm}}(T_A, T_B) = (|A| + |B|)^{-1}(|A \cup B| - |A \cap B|)$$

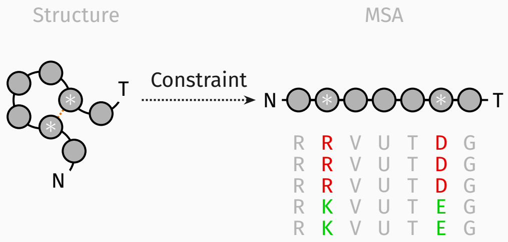
$$KF(T_A, T_B)^2 = \sum_{e \in A \cap B} (w_{(e,A)} - w_{(e,B)})^2 + \sum_{e \in A \setminus B} w^2_{(e,A)} + \sum_{e \in B \setminus A} w^2_{(e,B)}$$

$$\ell_{MAE}(T_A, T_B) = {}_nC_2{}^{-1} \sum_{\{i,j\} \in T_A} |d(i, j, T_a) - d(i, j, T_B)|$$

Results - Under LG+GC model, PF performs on par with ML



Results - What about more complex models ? - CherryML



adapted from Bittrich et al. 2019

- We **simulate** 250 pairs of **adjacent co-evolving sites**
- We use a 400 × 400 substitution **matrix** to describe residue **co-evolution**, from **CherryML**
- Most **ML** methods would consider **sites independent**

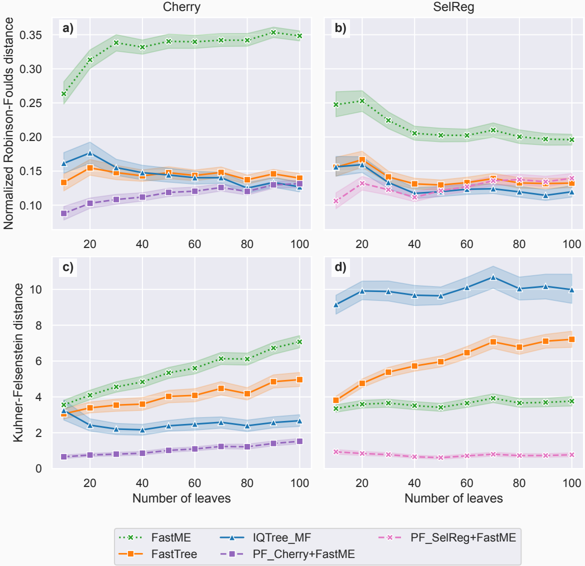
Prillo et al. 2023

Results - What about more complex models ? - SelReg

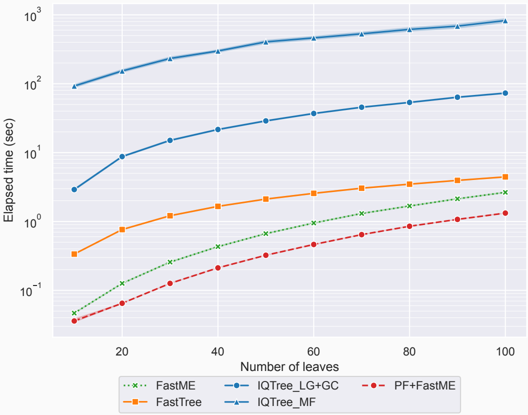
- Sites are: **positively**, **negatively** or **neutrally** selected
- **Codon** model with a 61×61 matrix
- 263 **empirical** amino-acid **profiles**
- **ML-Inferrable** with mixture-models but **expensive**

Duchemin et al. 2023; Halpern and Bruno 1998; Tamuri and Reis 2021

Results - Under complex models, PF performs well



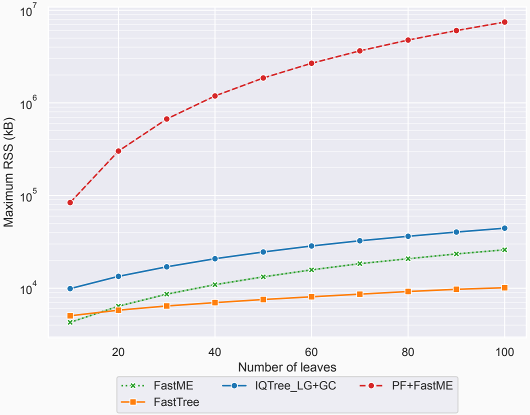
Results - Inference speed



- **PhyloFormer** is the **fastest** method[a]
- PhyloFormer is even **faster than FastME** on its own
- Inference **speed** is **independent** from model **complexity**

[a] using a GPU & more memory

Results - Memory consumption



- **PhyloFormer** has the **highest memory** footprint *(by far)*
- Even **more-so** at **training-time**
- However, **PhyloFormer** can be run on **CPU**

PhyloFormer - Take Home messages

Take-Home

1. On the standard **LG model**, PhyloFormer performs on **par with ML** methods
   - **Especially** w.r.t. pairwise **distances**
   - **Less** w.r.t pure **topology**
2. It is **easily adaptable** to **complex models**, where computing the **likelihood** is **impossible**
3. Once trained, it is the **fastest method**[1]

---

[1] Provided you have a GPU and a lot of trees to infer...

Improving PhyloFormer

We heard the good, what about the bad ?

1. **Embedding** sequence **pairs** scales in $\mathcal{O}(n^2)$ in **time** and **memory**
   $\Rightarrow$ **Hard** to scale to **large MSAs** and/or **long sequences**
2. We have **no guarantees** that the predicted **distances** are **tree-like**
   $\Rightarrow$ Is predicting **distances** and **trees** really **equivalent here** ?
3. **Not model-agnostic**, we perform implicit **model selection** at simulation-time
4. PhyloFormer is **dependant on the MSA** input, by **aligning** we are already introducing **bias**

Extensions - Linear scaling in *n*

- Basic **idea**: wait until **last-minute** to lift up into **pair-space**
- Use **Axial Attention** to learn MSA-aware **sequence embeddings**
- Compute **parametrized** pairwise **embedding-distances**:
  - Either **euclidean distances** between embeddings
  - Or with **symmetric bilinear form**
- **Related** work in `NeuroSeed`, e.g. **edit-distance** approximation with **hyperbolic** sequence **embeddings**

Corso et al. 2021; Vienne et al. 2012; Layer and Rhodes 2017

Extension - Learning embedding distances

Given **sequence-embeddings** $\Phi(S)$ of shape $(n \times d)$

$$E = \Phi(S)W_{euc}^{\top} + b_{euc}$$
$$O = PairwiseEuclidean(E)$$

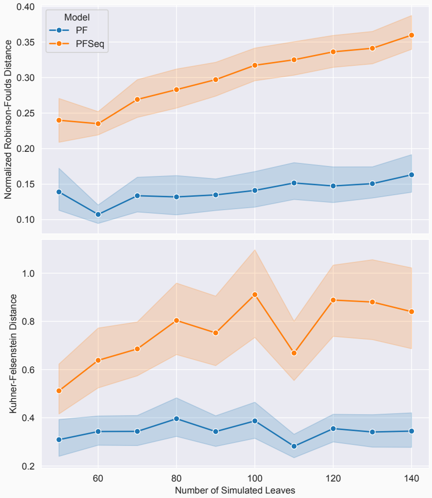
- Parametrized by **weights** W_{euc} and **bias** b_{euc}
- **Tree distance** $d_{ij,T}$ is **not euclidean**, $\sqrt{d_{ij,T}}$ is though

$$O = SoftPlus(\Phi(S)^{\top}W_{bil}\Phi(S) + b_{bil})$$
$$W_{bil} = W_{bil}^{\top}$$

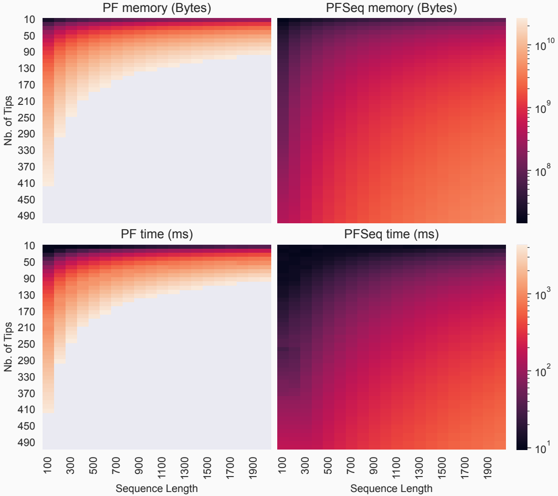
- Parametrized by **weights** W_{bil} and **bias** b_{bil}
- W_{θ} ensures that the distance **matrix** is **symmetric**
- *SoftPlus* ensures that **distances** are **positive**

Extensions - How does PFSeq perform ?

- **Training** PFSeq well is **harder** than with **PhyloFormer**
- Maybe the **pairwise** information is **harder to extract** from sequence-embeddings ?
- **Similar** train **loss** values but **differing test** performance
- Train **longer** and with **more data** ?



Extensions - PFseq scales Much better than PhyloFormer

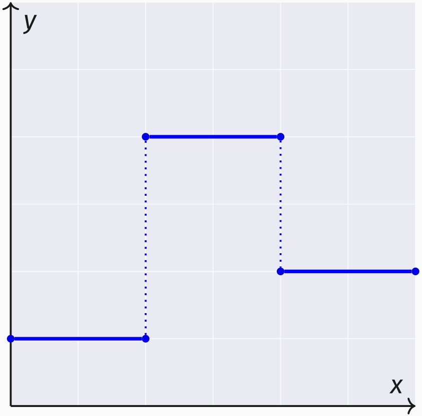


Extensions - End-to-end phylogenetic inference

- **Theoretically**, estimating **distances** or a **tree** are **equivalent** tasks
- **Practically**, not so much ...
- Can we **constrain** the output to **tree-like distances**?
- By adding a **NJ** step **after PhyloFormer** we can **output trees** directly
- **Problem**: NJ is **iterative** and **discrete**, i.e. **not great** for learning

Extensions - Estimating gradients through discrete operations

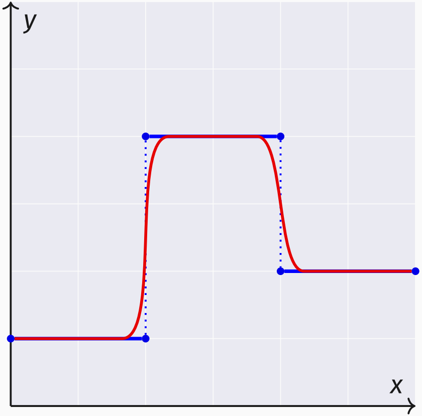
- **Discrete** implies **non-differentiable**
- **Non-differentiability** of a **finite** number of points is **not** always a **problem**: e.g. **ReLU**
- The **problem** is **piecewise constance**: $\nabla f(x) = 0; \forall x$



Berthet et al. 2020; Jang et al. 2017

Extensions - Estimating gradients through discrete operations

- **Discrete** implies **non-differentiable**
- **Non-differentiability** of a **finite** number of points is **not** always a **problem**: e.g. **ReLU**
- The **problem** is **piecewise constance**: $\nabla f(x) = 0; \forall x$
- **Smooth** out f and ∇f:
  - A **perturbation** approach
  - **Straight-through** trick



Berthet et al. 2020; Jang et al. 2017

Extensions - The Gumbel softmax straight-through “trick“

- Useful when you **need** to have **discrete steps** in your algorithm e.g:
  - **Pairwise** sequence **alignment**: average *softmax* values at each DP cell
    $\Rightarrow$ ***No need** for discrete decision*
  - **Neighbour-Joining**: You have to **merge nodes** to **advance** in the algorithm
    $\Rightarrow$ ***Need** to take discrete decision*
- Given O a **discrete** operation with corresponding **“soft”** version $\mathcal{O}$:
  (e.g. $O =$ argmax and $\mathcal{O} =$ softmax)

$$\text{forward:}\ \ X \mapsto O(X)$$
$$\text{backward:}\ \ \nabla_\theta O(X) \approx \nabla_\theta \mathcal{O}(X)$$

```
out = (
  hard(in) - soft(in)
).detach() + soft(in)
```

Jang et al. 2017

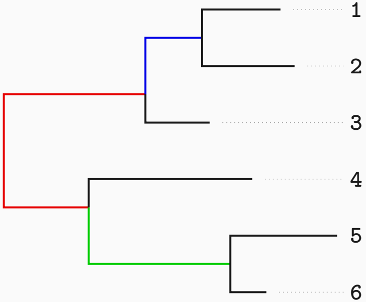
Extensions - Differentiable Neighbour-Joining

We have managed to build a `PyTorch` implementation of **NJ** that:

- Runs on the **GPU**
- Is **differentiable** w.r.t. model parameters because:
  - We use the **straight-through** trick to **approximate gradients** through merge-operations
  - We **avoid** any indexing and in-place operations that **might break** the computational **graph**

Extensions - Towards a topological loss function, splits

- **Tree-topologies** are uniquely **represented** by a set of **leaf-splits**
- With a leaf-**ordering** and a tie-breaking **rule** we have a **unique matrix** representation
- I.e we have a **topological target**



$$\begin{matrix} \{(1,2) \mid (3,4,5,6)\} \\ \{(1,2,3) \mid (4,5,6)\} \\ \{(1,2,3,4) \mid (5,6)\} \end{matrix} \Leftrightarrow \begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

Extensions - Topological Loss functions

Let S and $\bar{S}$ be **row-normalized** split matrices, S_j the j^{th} row of S

RF as a loss

$$RF(S,\bar{S}) = \sum_{i,j} \mathbb{1}_{(S\bar{S}^\top)_{ij}=1}$$

$$RF(S,\bar{S}) \approx \sum_{i,j} (S\bar{S}^\top)_{ij}^p ; p \gg 1$$

$\mathbb{1}_{(S\bar{S}^\top)=1}$ is **discrete,**
$x^p ; p \gg 1$ **not** very **stable**

Relaxing RF

$$\begin{aligned} RF_{ish}(S,\bar{S}) &= \max_{\pi\in\Pi} \sum_{j=1}^{n-3} S_j^\top (\bar{S}_\pi)_j \\ &= \max_{A\in P} \operatorname{tr}(S\bar{S}^\top A) \\ &= \max_{A\in conv(P)} \operatorname{tr}(S\bar{S}^\top A) + \varepsilon H(A) \end{aligned}$$

Optimal Transport formulation of an **RF-like** topological **loss**

Extensions - Topoformer preliminary results

Bad news:

- **Learning** with a topological loss is **hard**
- The **RF approximation** with x^p is too **close to discrete** version
- The **OT** RF-like loss is **not behaving** as we wish

Good news:

- We can **recover** a **distance** matrix **easily** from NJ's output
- We can use L_1 loss as with **PhyloFormer**, but **guaranteed** that d_{ij} is **tree-like**
- This is more **promising** and training curves look **better**

Wrapping things up

Global Conclusion

- **Phyloformer** enables **phylogenetic** inference from **start** to *almost* **finish**
- This is the **first deep-learning method** that does so
- Enables **likelihood-free** phylogenetic inference, paving the way for **complex models**

- **Soon** it will *(hopefully)*
  1. **Scale linearly** with the number of sequences
  2. Be truly **end-to-end** and produce trees
- **Active** work done to **extend PhyloFormer** functionality to other tasks

Other related work *(mostly not mine)*

Exciting work being done in the **team** and with **collaborators**

- Estimating **epidemiological parameters** from MSAs directly

  V. Garot, L. Jacob, S. Alizon and A. Zhukova
- Quantifying **selection** at each MSA sites

  L. Nesterenko, C. West and B. Boussau
- Inferring **phylogenies** under **Potts models**

  P. Barrat-Charlaix, L. Jacob and I
- Estimating **ecological parameters** on trees using GNNs

  A. Leroy, H. Morlon and L. Jacob
- Detecting **Ghost lineages** an gene tree **reconciliation** with GNNs

  E. Marsot, B. Boussau, D. de Vienne and L. Jacob

Obligatory self-promotion slide

- The *updated* **PhyloFormer preprint** is out ⇒
- **Help me** build a **phylogenetics** crate in **Rust**:
  lucblassel/phylotree-rs
- You can find the **slides** here:
  lucblassel.com/files/slides_lisn_2024.pdf

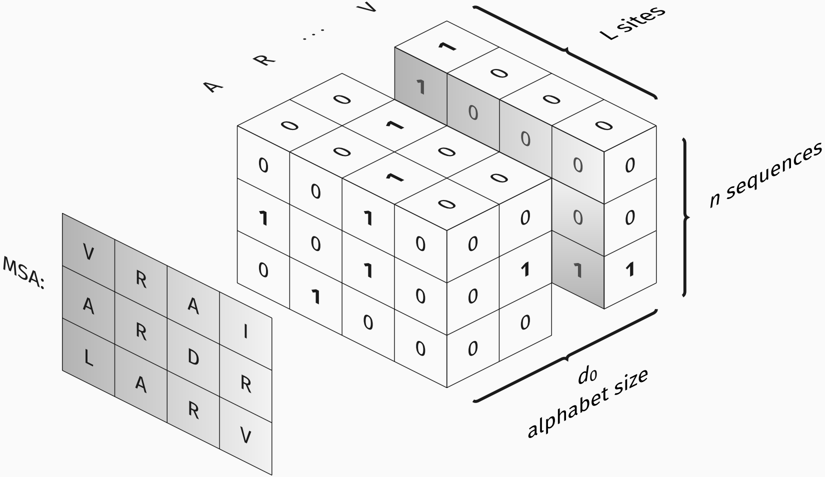
bioRxiv
THE PREPRINT SERVER FOR BIOLOGY

Thanks to:

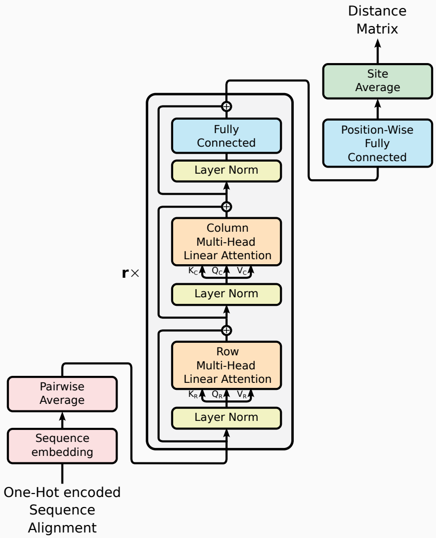
- Luca Nesterenko
- Laurent Jacob
- Bastien Boussau
- Philippe Veber
- Martin Ruffel
- Dexiong Chen
- Johanna Trost

Special thanks to Jean-Zay for all the GPUs!

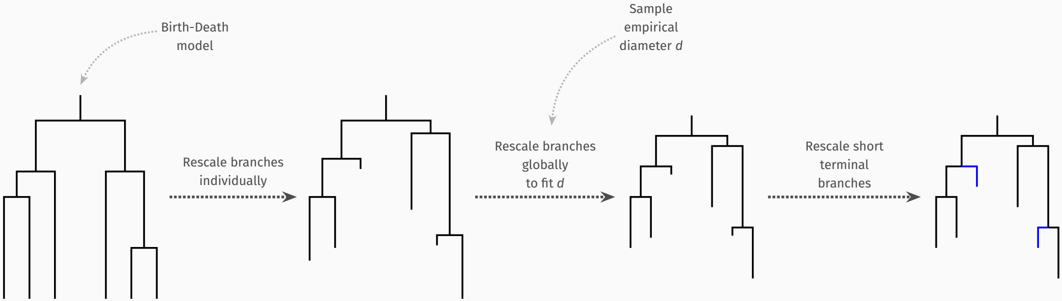
Additional Methods - Data Encoding



Additional Methods - Network Architecture



Additional Methods - Tree simulation



Birth-Death model

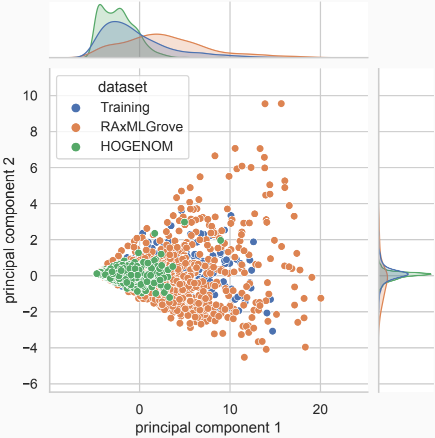
Rescale branches individually

Sample empirical diameter *d*

Rescale branches globally to fit *d*

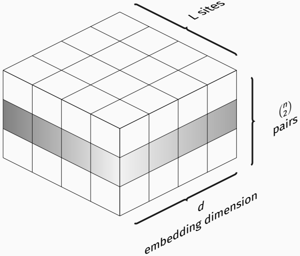
Rescale short terminal branches

Additional Methods - Realistic tree distribution



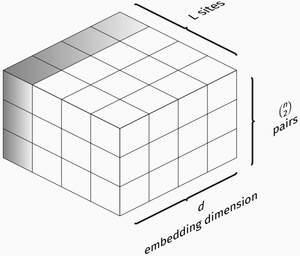
Additional Methods - Axial self-attention[2]

Row attention



Update sites in a pair by **looking** at all **other sites** in the same **pair**

Column attention



Update sites in a pair by **looking** at the **same site** in all **other pairs**

---

[2] Ho et al. 2019; Rao et al. 2021.

Additional Methods - Custom Rank1 Attention

Scaled dot-product

$$Z = softmax\left(\frac{QK^\top}{\sqrt{d_k}}\right)V$$

Linear Kernel Attention

$$z_i = \frac{\tilde{\phi}(q_i)^\top \sum_{j=1}^{M} \tilde{\phi}(k_j)v_j}{\tilde{\phi}(q_i)^\top \sum_{h=1}^{M} \tilde{\phi}(k_h)}$$

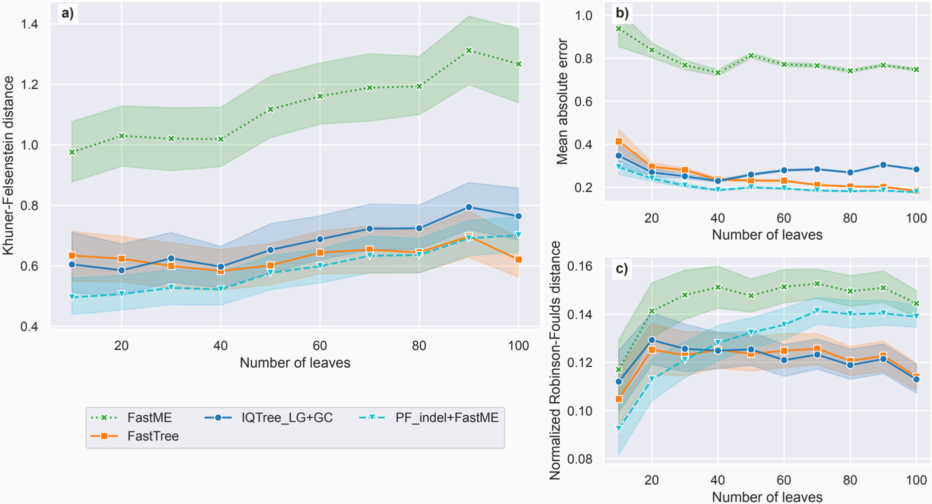
Our Rank-1 Attention

$$\tilde{\phi}(x) = \begin{cases} x+1, & \texttt{if } x > 0 \\ \exp\{(x)\} & \texttt{if } x \leq 0, \end{cases}$$

$$z_i' = \frac{\tilde{\phi}(q_i)}{M^{-1}\sum_{g=1}^{M} \tilde{\phi}(q_g)} \cdot \frac{\sum_{j=1}^{M} \tilde{\phi}(k_j)v_j}{\sum_{h=1}^{M} \tilde{\phi}(k_h)}$$

Vaswani et al. 2017; Katharopoulos et al. 2020

Additional Results - Indel Model

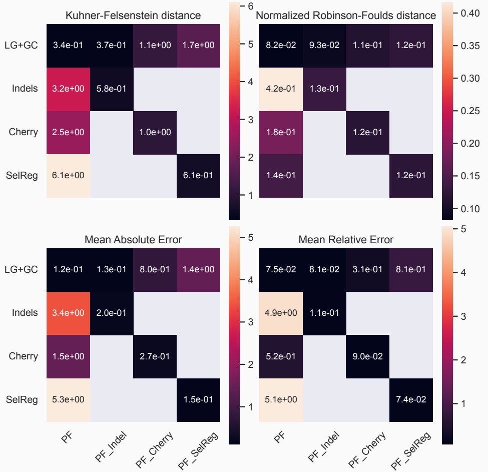


Additional Results - PF captures co-evolution out of the box

Network on **test dataset**:	$\text{PF}_{\text{Cherry}}$ on **Cherry**	PF on **Cherry**	$\text{PF}_{\text{Cherry}}$ on **LG**	PF on **LG**
$a =$ co-evolution attentions	0.256	0.255	0.120	0.135
$b =$ other attentions	0.098	0.115	0.121	0.136
Ratio a/b	4.424	3.408	0.999	0.995
auto-attentions	0.579	0.535	0.542	0.523

- **PF** Already assigns **high-attention** values to **co-evolving** site pairs
- $\textbf{PF}_{\textbf{Cherry}}$ likely **exploits** this signal for **better performance**
- PF does this **without** needing **positional encoding**

Additional Results - PF performs model-based inference



Kuhner-Felsenstein distance

	PF	PF_Indel	PF_Cherry	PF_SelReg
LG+GC	3.4e-01	3.7e-01	1.1e+00	1.7e+00
Indels	3.2e+00	5.8e-01		
Cherry	2.5e+00		1.0e+00	
SelReg	6.1e+00			6.1e-01

Normalized Robinson-Foulds distance

	PF	PF_Indel	PF_Cherry	PF_SelReg
LG+GC	8.2e-02	9.3e-02	1.1e-01	1.2e-01
Indels	4.2e-01	1.3e-01		
Cherry	1.8e-01		1.2e-01	
SelReg	1.4e-01			1.2e-01

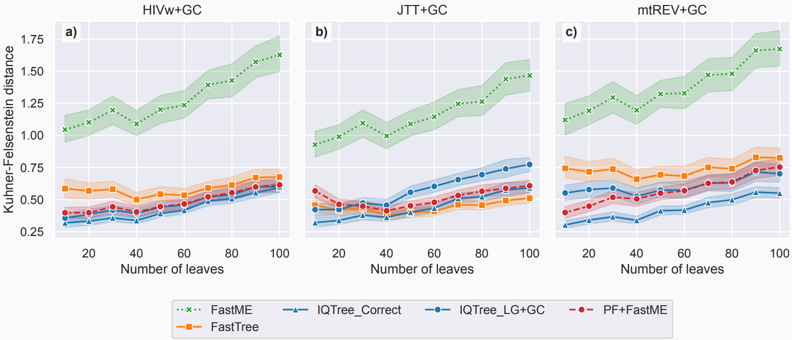
Mean Absolute Error

	PF	PF_Indel	PF_Cherry	PF_SelReg
LG+GC	1.2e-01	1.3e-01	8.0e-01	1.4e+00
Indels	3.4e+00	2.0e-01		
Cherry	1.5e+00		2.7e-01	
SelReg	5.3e+00			1.5e-01

Mean Relative Error

	PF	PF_Indel	PF_Cherry	PF_SelReg
LG+GC	7.5e-02	8.1e-02	3.1e-01	8.1e-01
Indels	4.9e+00	1.1e-01		
Cherry	5.2e-01		9.0e-02	
SelReg	5.1e+00			7.4e-02

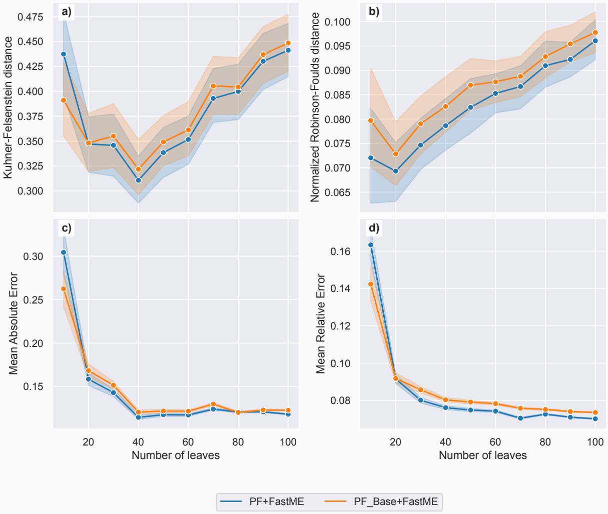
Additional Results - PhyloFormer is likelihood-free not model-free



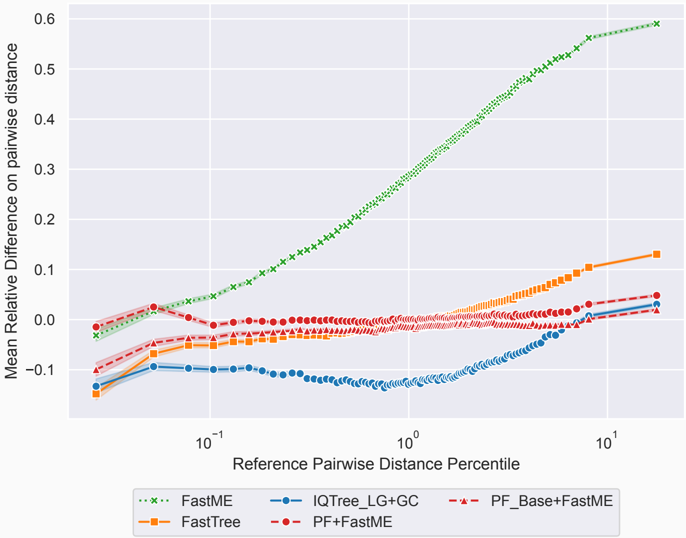
Substitution **models** chosen to be **far** from **LG**

Minh et al. 2021; Norn et al. 2021

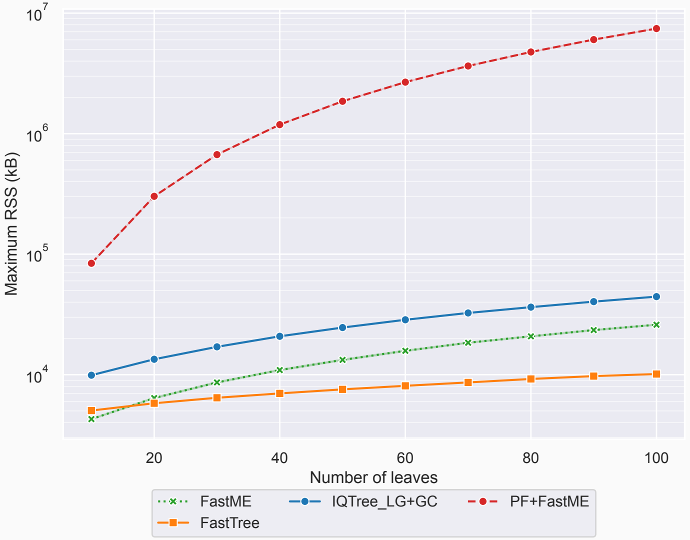
Additional Results - Fine tuning with MRE loss



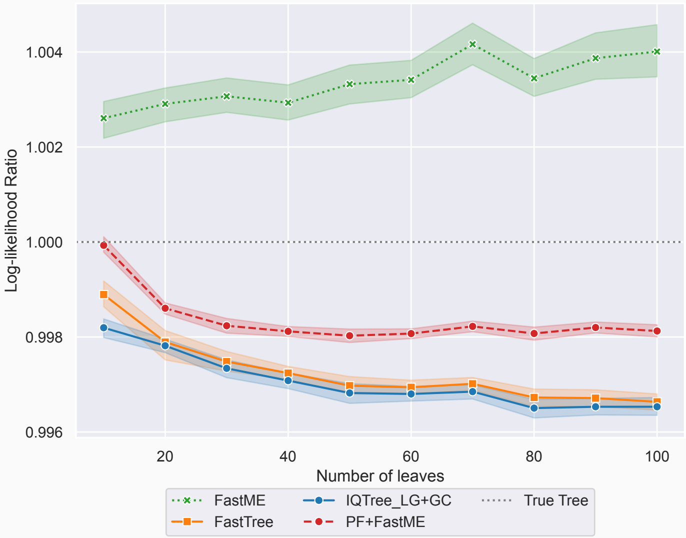
Additional Results - Mean relative error



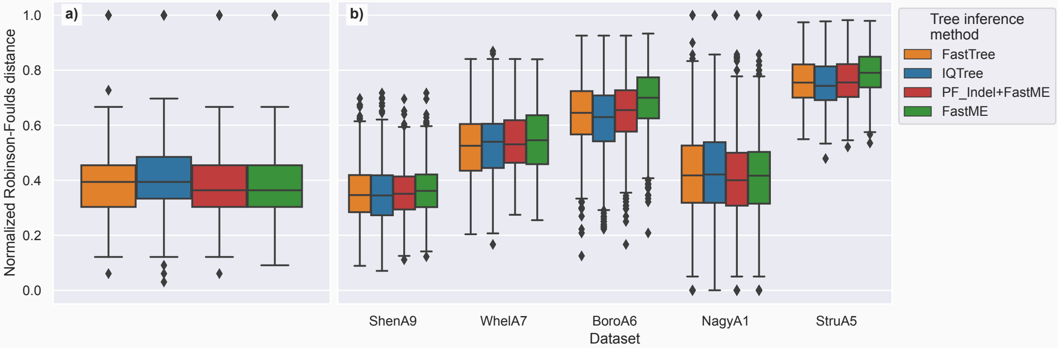
Additional Results - Memory requirements



Additional Results - PhyloFormer outputs likely trees



Additional Results - PhyloFormer is similar to SoTA on empirical data



Additional Results - Training runs

Network Name	Starting Network	Batch Size	Dataset Size	Model of evolution	Effective number of Steps/Epochs	GPUs used	Target learning rate	Target schedule steps	Selected checkpoint step	Loss Function
PF_{Base}	Initialized network	4	170k	LG+GC	145.18k/20.5	6×A100	10^{-3}	213.2k	144k	MAE
PF	PF_{Base}	4	224k	LG+GC	40.3k/4.32	6×A100	10^{-4}	66k	40,3k	MRE
PF_{Indel}	PF_{Base}	1	55k	LG+GC+indels	240k/17.45	4×V100	10^{-3}	240k	136.5k	MAE
$\text{PF}_{\text{Cherry}}$	PF_{Base}	4	1M	Cherry	30k/0.72	6×A100	10^{-3}	66k	18k	MAE
$\text{PF}_{\text{SelReg}}$	PF_{Base}	4	1M	SelReg	66k/1.58	6×A100	10^{-3}	66k	66k	MAE

References

Azouri, D., S. Abadi, et al. (2021). ***Harnessing machine learning to guide phylogenetic-tree search algorithms.*** In: *Nature communications* 12.1, p. 1983.

Azouri, D., O. Granit, et al. (2024). ***The tree reconstruction game: phylogenetic reconstruction using reinforcement learning.*** In: *Molecular Biology and Evolution* 41.6.

Balaban, M. et al. (2020). ***APPLES: Scalable Distance-Based Phylogenetic Placement with or without Alignments.*** In: *Systematic Biology* 69.3, pp. 566–578.

Bandelt, H.-J. and A. Dress (1986). ***Reconstructing the shape of a tree from observed dissimilarity data.*** In: *Advances in Applied Mathematics* 7.3, pp. 309–343.

Berthet, Q. et al. (2020). ***Learning with Differentiable Pertubed Optimizers.*** In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., pp. 9508–9519.

Bittrich, S. et al. (2019). ***StructureDistiller: Structural relevance scoring identifies the most informative entries of a contact map.*** In: *Scientific reports* 9.1, p. 18517.

Blassel, L., P. Medvedev, et al. (2022). ***Mapping-friendly sequence reductions: Going beyond homopolymer compression.*** In: *Iscience* 25.11.

Blassel, L., A. Tostevin, et al. (2021). ***Using machine learning and big data to explore the drug resistance landscape in HIV.*** In: *PLOS Computational Biology* 17.8, e1008873.

Blassel, L., A. Zhukova, et al. (2021). ***Drug resistance mutations in HIV: new bioinformatics approaches and challenges.*** In: *Current opinion in virology* 51, pp. 56–64.

Corso, G. et al. (2021). ***Neural distance embeddings for biological sequences.*** In: *Advances in Neural Information Processing Systems* 34, pp. 18539–18551.

Duchemin, L. et al. (2023). ***Evaluation of methods to detect shifts in directional selection at the genome scale.*** In: *Molecular Biology and Evolution* 40.2, msac247.

Felsenstein, J. (1993). ***PHYLIP (phylogeny inference package), version 3.5 c.*** Joseph Felsenstein.

— (2004). ***Inferring phylogenies.*** Vol. 2. Sinauer associates Sunderland, MA.

Gascuel, O. (1997). ***BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data.*** eng. In: *Molecular Biology and Evolution* 14.7, pp. 685–695.

Gascuel, O. and M. Steel (2016). ***A 'stochastic safety radius' for distance-based tree reconstruction.*** In: *Algorithmica* 74, pp. 1386–1403.

Guindon, S. and O. Gascuel (2003). ***A Simple, Fast, and Accurate Algorithm to Estimate Large Phylogenies by Maximum Likelihood.*** In: *Systematic Biology* 52.5, pp. 696–704.

Hadfield, J. et al. (2018). ***Nextstrain: real-time tracking of pathogen evolution.*** In: *Bioinformatics* 34.23, pp. 4121–4123.

Halpern, A. L. and W. J. Bruno (1998). ***Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies.*** In: *Molecular biology and evolution* 15.7, pp. 910–917.

Harms, M. J. and J. W. Thornton (2013). ***Evolutionary biochemistry: revealing the historical and physical causes of protein properties.*** In: *Nature reviews. Genetics* 14.8, pp. 559–571.

Ho, J. et al. (2019). ***Axial Attention in Multidimensional Transformers.*** In: *CoRR* abs/1912.12180.

Jang, E. et al. (2017). ***Categorical Reparameterization with Gumbel-Softmax.*** In: arXiv:1611.01144.

Jiang, Y. et al. (2022). ***DEPP: Deep Learning Enables Extending Species Trees using Single Genes.*** In: *Systematic Biology*, syac031.

Jumper, J. et al. (2021). ***Highly accurate protein structure prediction with AlphaFold.*** In: *Nature* 596.7873, pp. 583–589.

Katharopoulos, A. et al. (2020). ***Transformers are rnns: Fast autoregressive transformers with linear attention.*** PMLR.

Kleinman, C. L. et al. (2010). ***Statistical Potentials for Improved Structurally Constrained Evolutionary Models.*** In: *Molecular Biology and Evolution* 27.7, pp. 1546–1560.

Layer, M. and J. A. Rhodes (2017). ***Phylogenetic trees and Euclidean embeddings.*** In: *Journal of mathematical biology* 74, pp. 99–111.

Lefort, V. et al. (2015). ***FastME 2.0: a comprehensive, accurate, and fast distance-based phylogeny inference program.*** In: *Molecular biology and evolution* 32.10, pp. 2798–2800.

Minh, B. Q. et al. (2021). ***QMaker: Fast and Accurate Method to Estimate Empirical Models of Protein Evolution.*** In: *Systematic Biology* 70.5, pp. 1046–1060.

Nelson, M. I. et al. (2008). ***Multiple Reassortment Events in the Evolutionary History of H1N1 Influenza A Virus Since 1918.*** In: *PLoS Pathogens* 4.2, e1000012.

Nguyen, L.-T. et al. (2015). ***IQ-TREE: a fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies.*** In: *Molecular biology and evolution* 32.1, pp. 268–274.

Norn, C. et al. (2021). ***A thermodynamic model of protein structure evolution explains empirical amino acid substitution matrices.*** In: *Protein Science* 30.10, pp. 2057–2068.

Perez-Lamarque, B. et al. (2022). ***Analysing diversification dynamics using barcoding data: The case of an obligate mycorrhizal symbiont.*** eng. In: *Molecular Ecology* 31.12, pp. 3496–3512.

Price, M. N. et al. (2010). ***FastTree 2–approximately maximum-likelihood trees for large alignments.*** In: *PloS one* 5.3, e9490.

Prillo, S. et al. (2023). ***CherryML: scalable maximum likelihood estimation of phylogenetic models.*** In: *Nature methods* 20.8, pp. 1232–1236.

Rao, R. M. et al. (2021). ***MSA Transformer.*** In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, pp. 8844–8856.

Saitou, N. and M. Nei (1987). ***The neighbor-joining method: a new method for reconstructing phylogenetic trees.*** In: *Molecular biology and evolution* 4.4, pp. 406–425.

Smith, M. L. and M. W. Hahn (2023). ***Phylogenetic inference using generative adversarial networks.*** In: *Bioinformatics* 39.9, btad543.

Strimmer, K. and A. Von Haeseler (1996). ***Quartet puzzling: a quartet maximum-likelihood method for reconstructing tree topologies.*** In: *Molecular biology and evolution* 13.7, pp. 964–969.

Suvorov, A. et al. (2019). ***Accurate Inference of Tree Topologies from Multiple Sequence Alignments Using Deep Learning.*** In: *Systematic Biology* 69.2, pp. 221–233.

Tamuri, A. U. and M. d. Reis (2021). ***A mutation-selection model of protein evolution under persistent positive selection.*** en. In: *bioRxiv*. Publisher: Cold Spring Harbor Laboratory Section: New Results, p. 2021.05.18.444611.

Tang, X. et al. (2024). ***Novel symmetry-preserving neural network model for phylogenetic inference.*** In: *Bioinformatics Advances* 4.1, vbae022.

Vaswani, A. et al. (2017). ***Attention is all you need.*** In: *Advances in neural information processing systems* 30.

Vienne, D. M. de et al. (2012). ***Phylo-MCOA: a fast and efficient method to detect outlier genes and species in phylogenomics using multiple co-inertia analysis.*** In: *Molecular Biology and Evolution* 29.6, pp. 1587–1598.

Zaharias, P. et al. (2022). ***Re-evaluating Deep Neural Networks for Phylogeny Estimation: The issue of taxon sampling.*** In: *Journal of Computational Biology*.

Zou, Z. et al. (2020). ***Deep residual neural networks resolve quartet molecular phylogenies.*** In: *Molecular biology and evolution* 37.5, pp. 1495–1507.