Phyloformer: towards fast and accurate phylogeny estimation with self-attention networks

Laurent Jacob

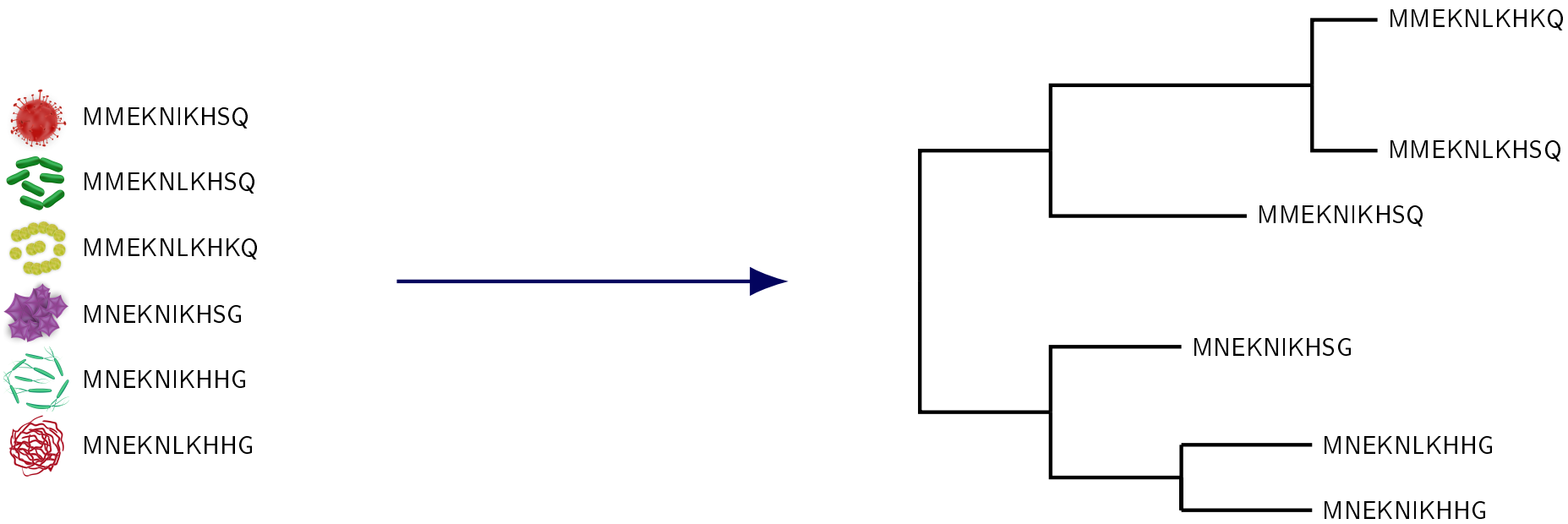
BeVAS workshop, April 18th 2023

Acknowledgements

Luca Nesterenko

Bastien Boussau

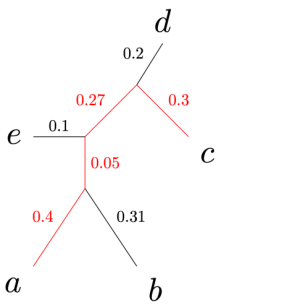
Molecular phylogenetics



Reconstruct the evolutionary history of homologous sequences

Two main paradigms

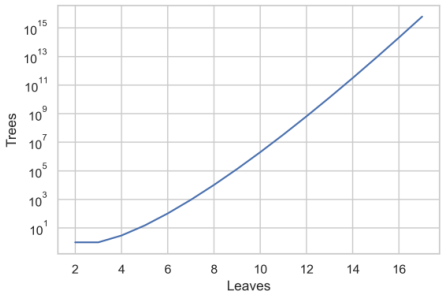
Distance methods: fast but inaccurate



$d_{ac} = 0.4 + 0.05 + 0.27 + 0.3$

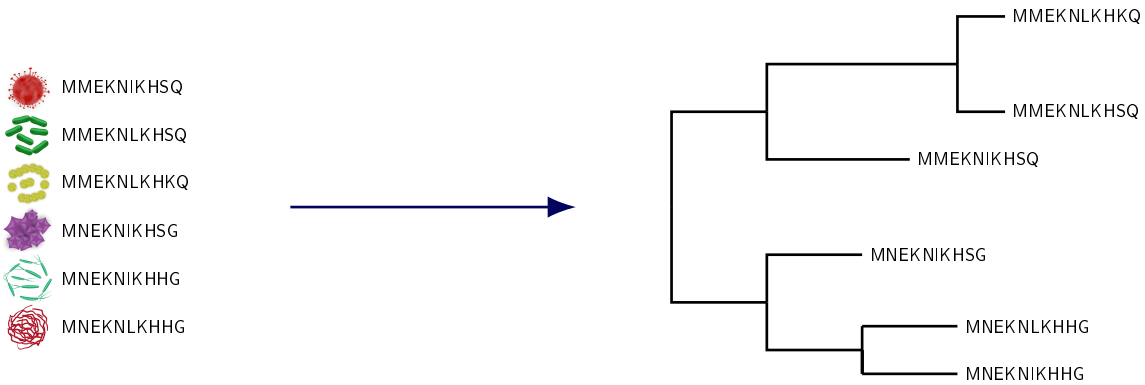
- Start from pairwise distances between sequences.
- Fast, guaranteed to recover the right tree given the right distances...
- ...but distance estimates are often inaccurate, leading to poor reconstruction.

Maximum likelihood: accurate but slow



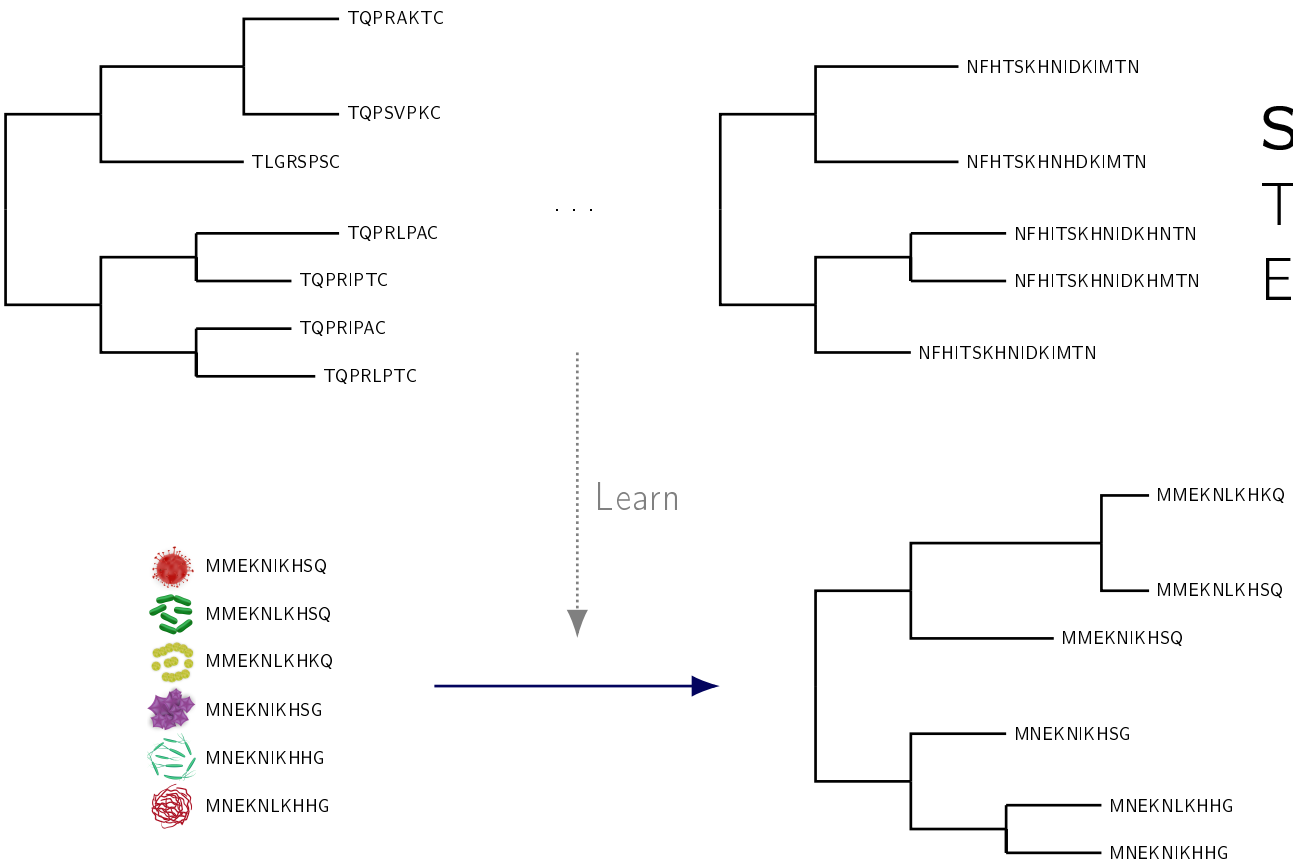
- Given a probabilistic sequence evolution model, find the tree making the whole set of sequences most likely.
- State of the art accuracy, but explores a huge tree space.
- Relies on strong simplifying assumptions.

Deep learning for molecular evolution



A new paradigm for phylogenetic reconstruction:
learn a function **predicting the tree from homologous sequence**.

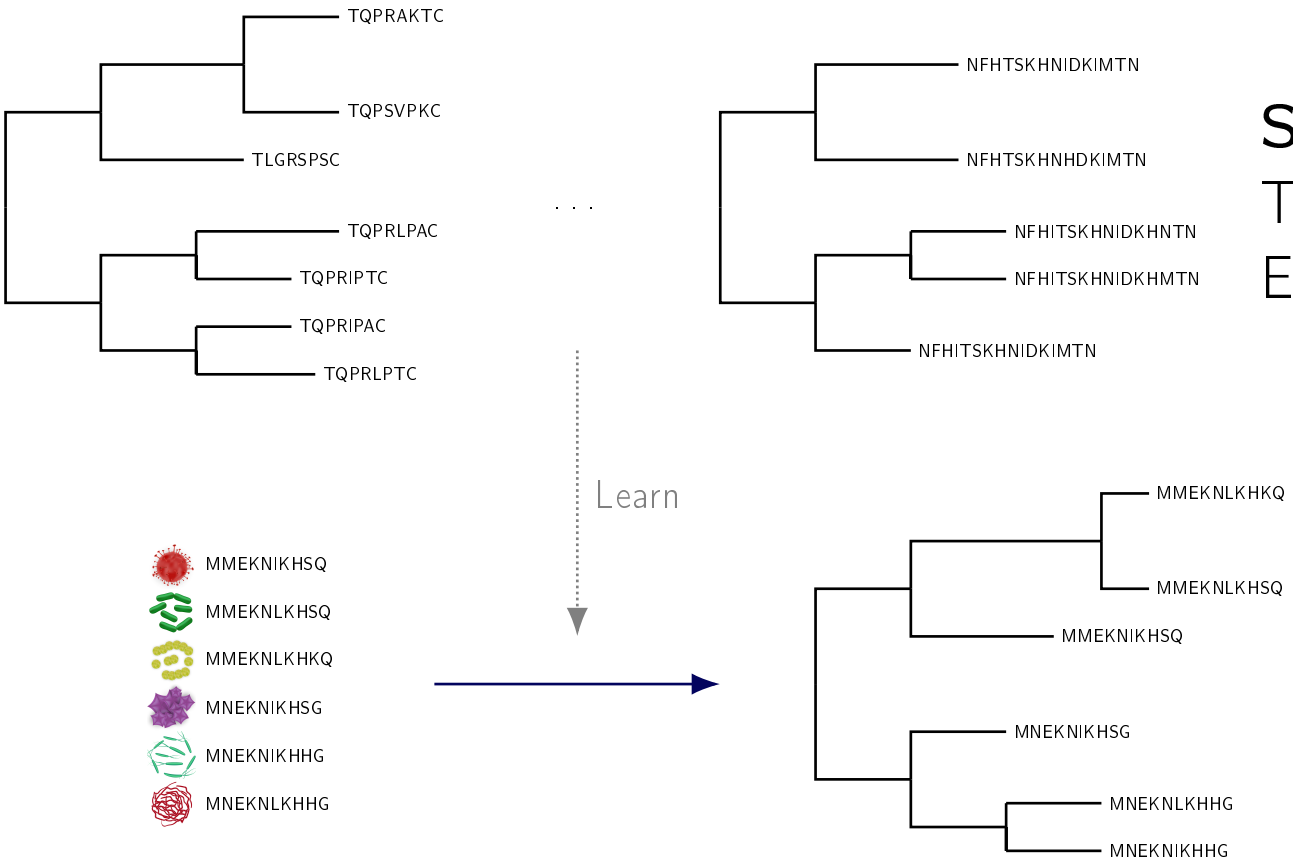
Deep learning for molecular evolution



Simulate examples of:
Trees
Evolved sequences

A new paradigm for phylogenetic reconstruction:
learn a function **predicting the tree from homologous sequence**.

Deep learning for molecular evolution



Simulate examples of:
Trees
Evolved sequences

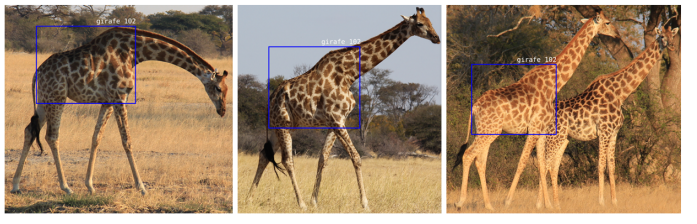
A new paradigm for phylogenetic reconstruction:
learn a function **predicting the tree from homologous sequence**.

Motivation: faster and/or dealing with more complex models.

But how is it ok to learn from simulated data?

An unusual setting for supervised learning

- Usually: perform induction from real-world data.



- Here: we have access to a **forward** process

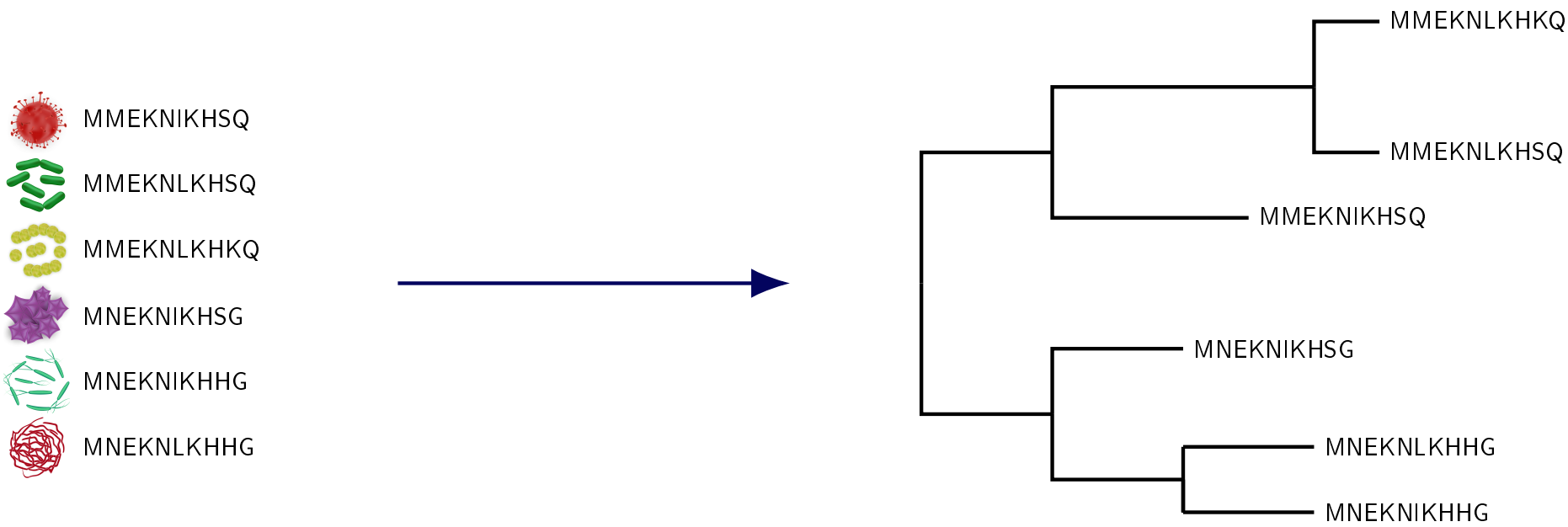
$$\text{Tree} \xrightarrow{\text{Fast simulation}} \text{Homologous sequences}$$

and will use supervised learning to **reverse** it.

Intuition: conceptually not so different from maximum likelihood

- Likelihood optimization is too expensive.
- Instead, we learn a map from the input to a solution.

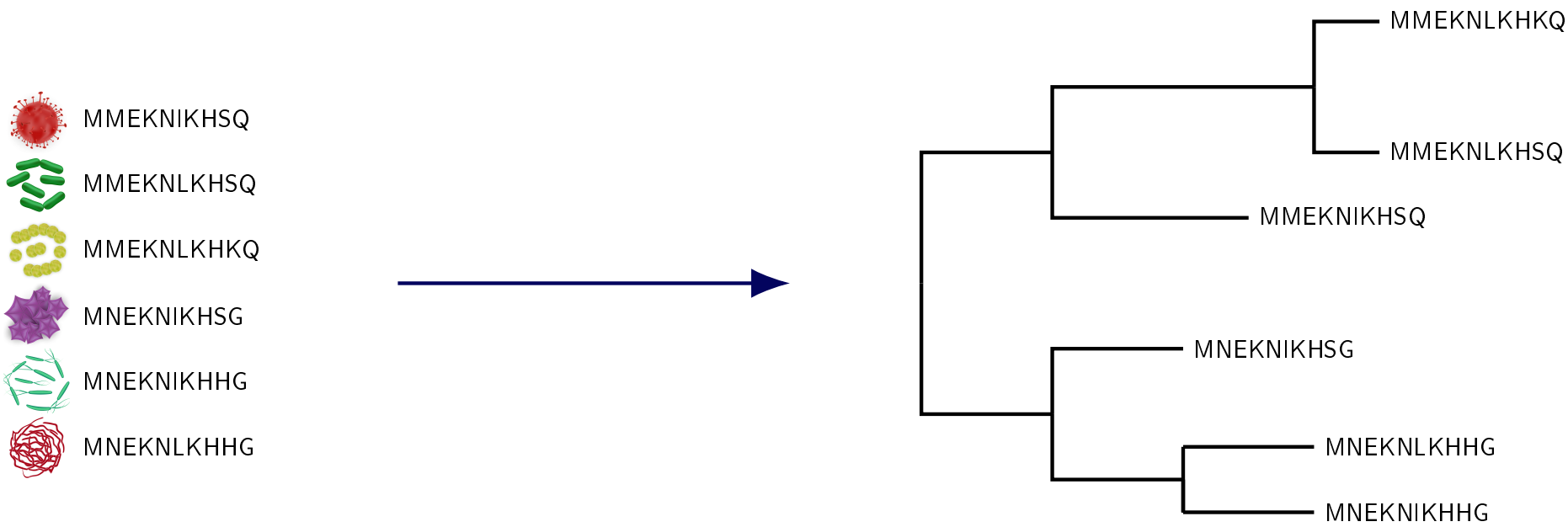
Concrete issues



We need a learnable function that:

- outputs a phylogenetic tree.
- takes as input a set of homologous sequences,

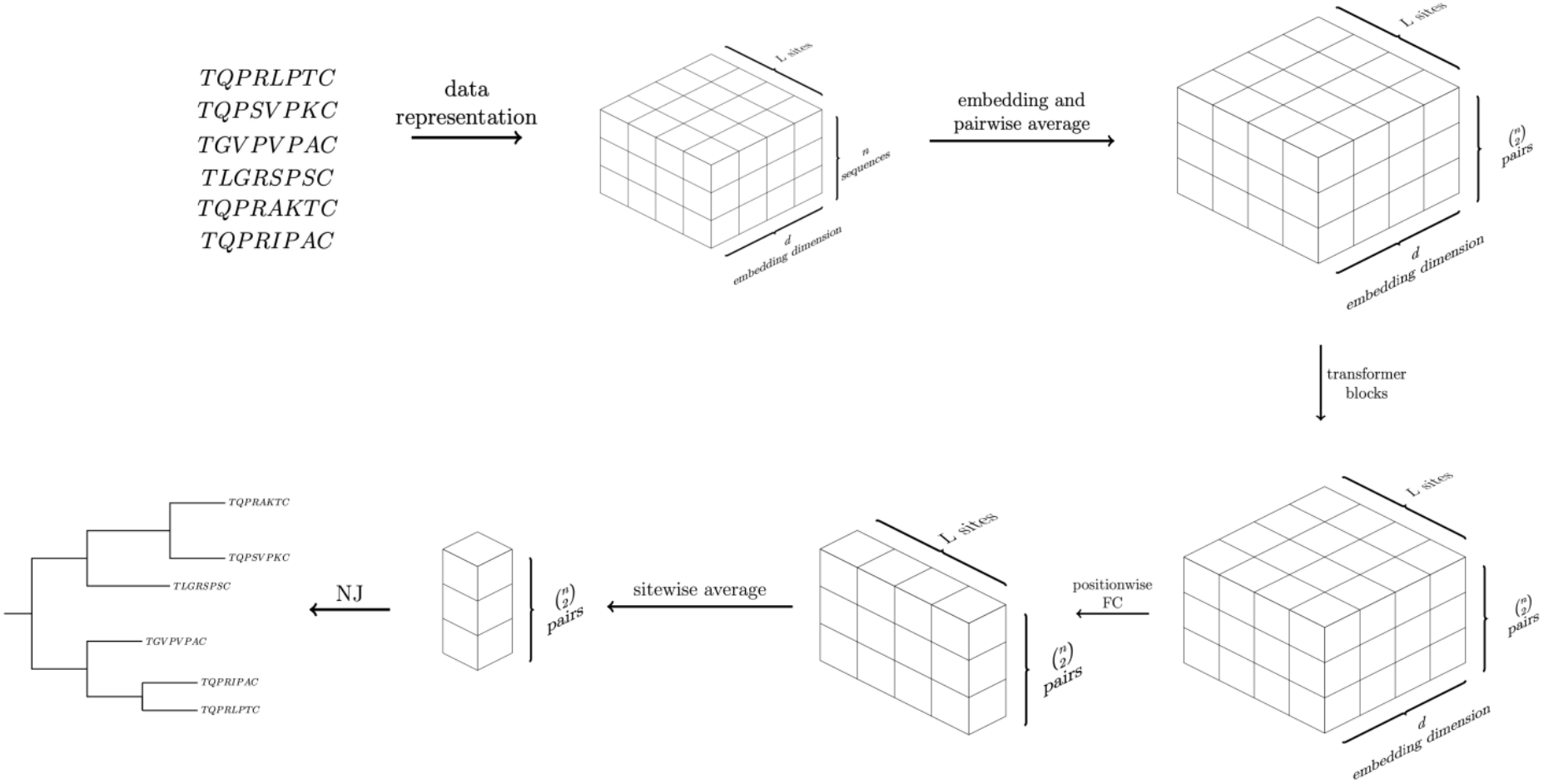
Concrete issues



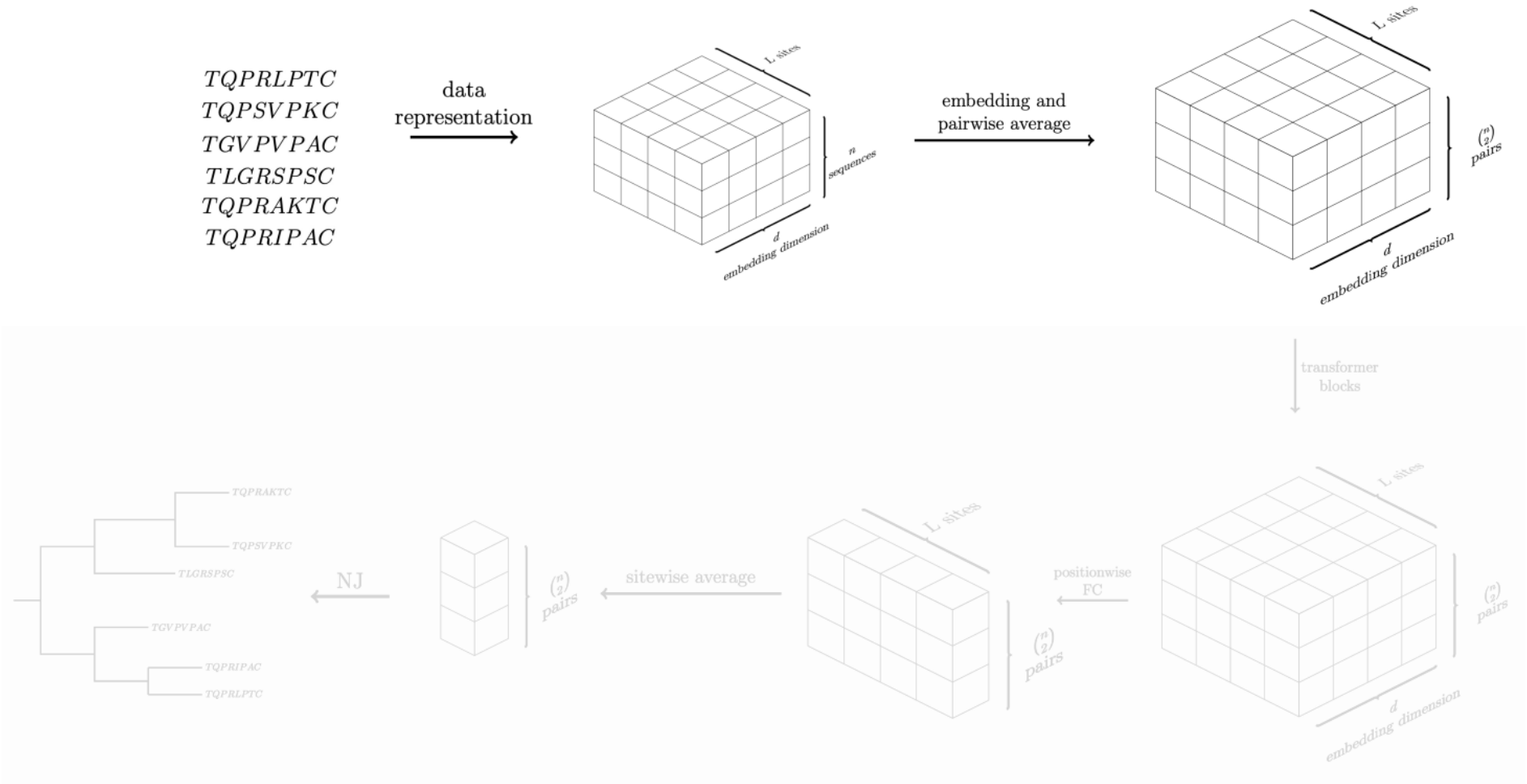
We need a learnable function that:

- outputs a phylogenetic tree.
  $\rightarrow$ **use evolutionary distances as a proxy.**
- takes as input a set of homologous sequences,
  $\rightarrow$ **use self-attention** (dual to contact prediction).

Phyloformer overview



Phyloformer overview

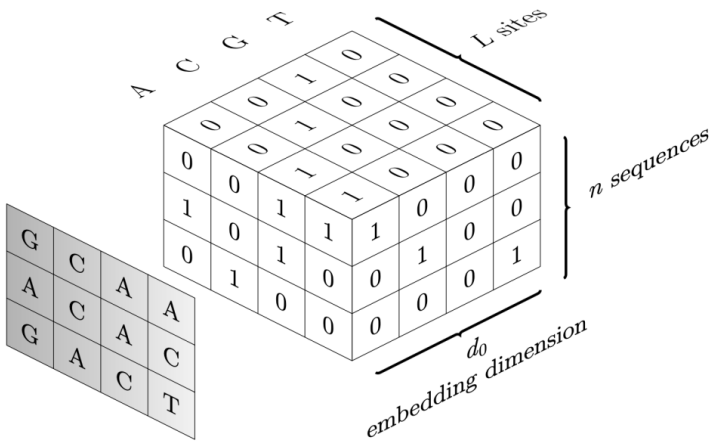


One-hot encoding for aligned sequences

A single sequence:

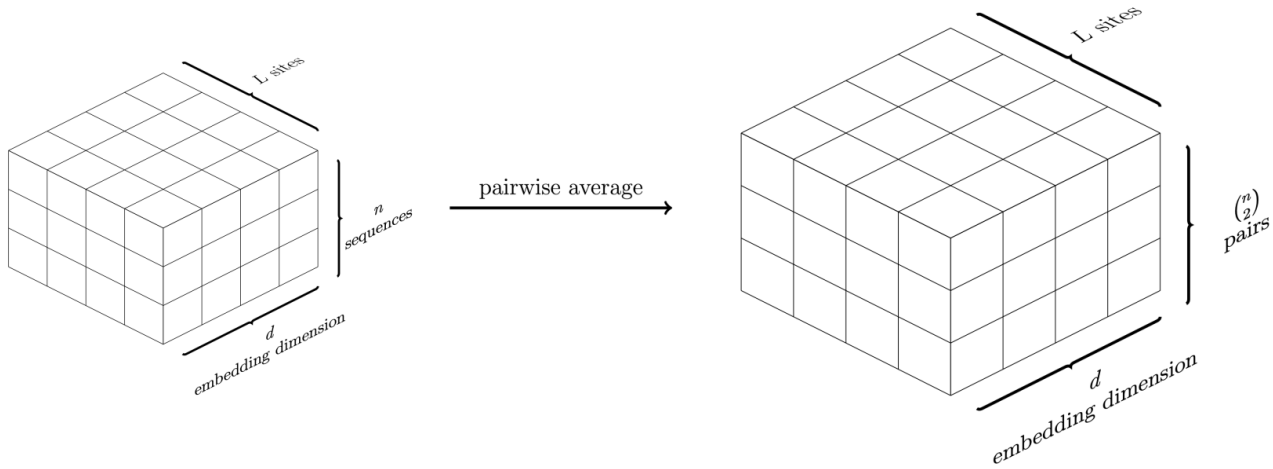
	A	A	C	G	T	...
A	1	1	0	0	0	...
C	0	0	1	0	0	...
T	0	0	0	0	1	...
G	0	0	0	1	0	...

A set of aligned sequences:



Our alphabet is actually {A, R, N, D, ..., Y, V, X, -} so $d_0 = 22$.

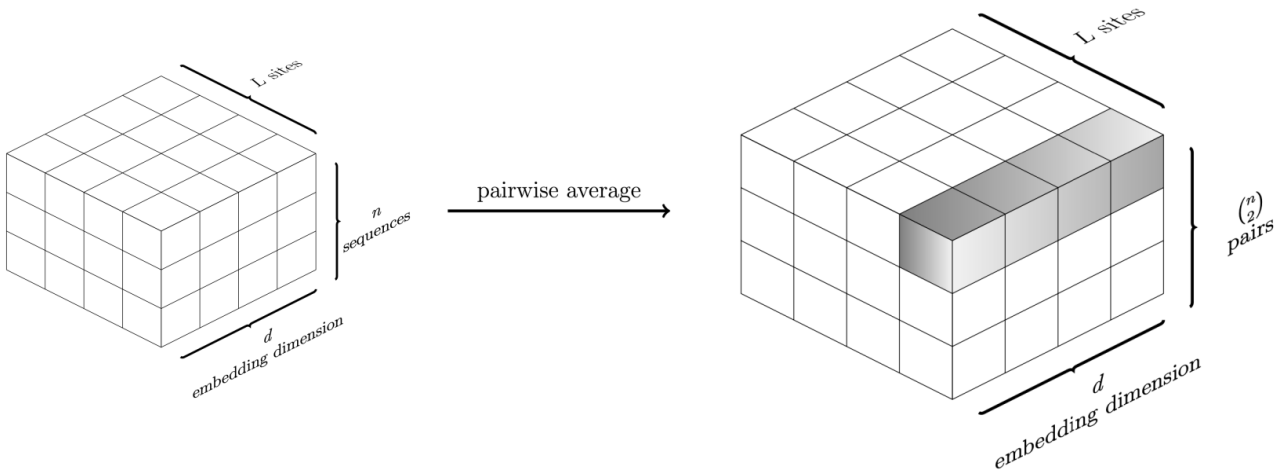
Encoding **pairs** of aligned sequences



- We choose to work on pairs of sequences (predict distance for each).
- We represent each pair by simply averaging over sequences.

	A	A	C	G	T	...
	A	T	C	C	T	...
A	1	0.5	0	0	0	...
C	0	0	1	0.5	0	...
T	0	0.5	0	0	1	...
G	0	0	0	0.5	0	...

Encoding **pairs** of aligned sequences



- We choose to work on pairs of sequences (predict distance for each).
- We represent each pair by simply averaging over sequences.

	A	A	C	G	T	...
	A	T	C	C	T	...
A	1	0.5	0	0	0	...
C	0	0	1	0.5	0	...
T	0	0.5	0	0	1	...
G	0	0	0	0.5	0	...

- We now have a set of $\binom{n}{2} \times L$ amino acids encoded as $\mathbb{R}^{d=22}$ vectors.

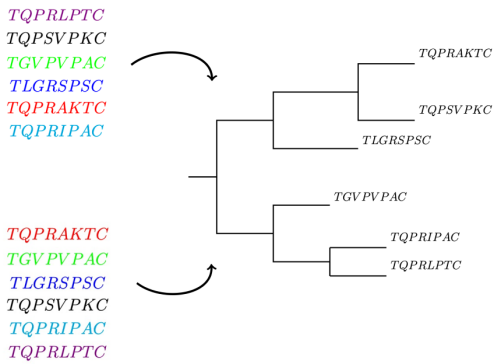
Permutation invariance and equivariance

If we permute input sequences:

- output distances should follow the same permutation (equivariance):

$$f(\pi((s_1, s_2), \ldots, (s_{n-1}, s_n))) = \pi(f((s_1, s_2), \ldots, (s_{n-1}, s_n))).$$

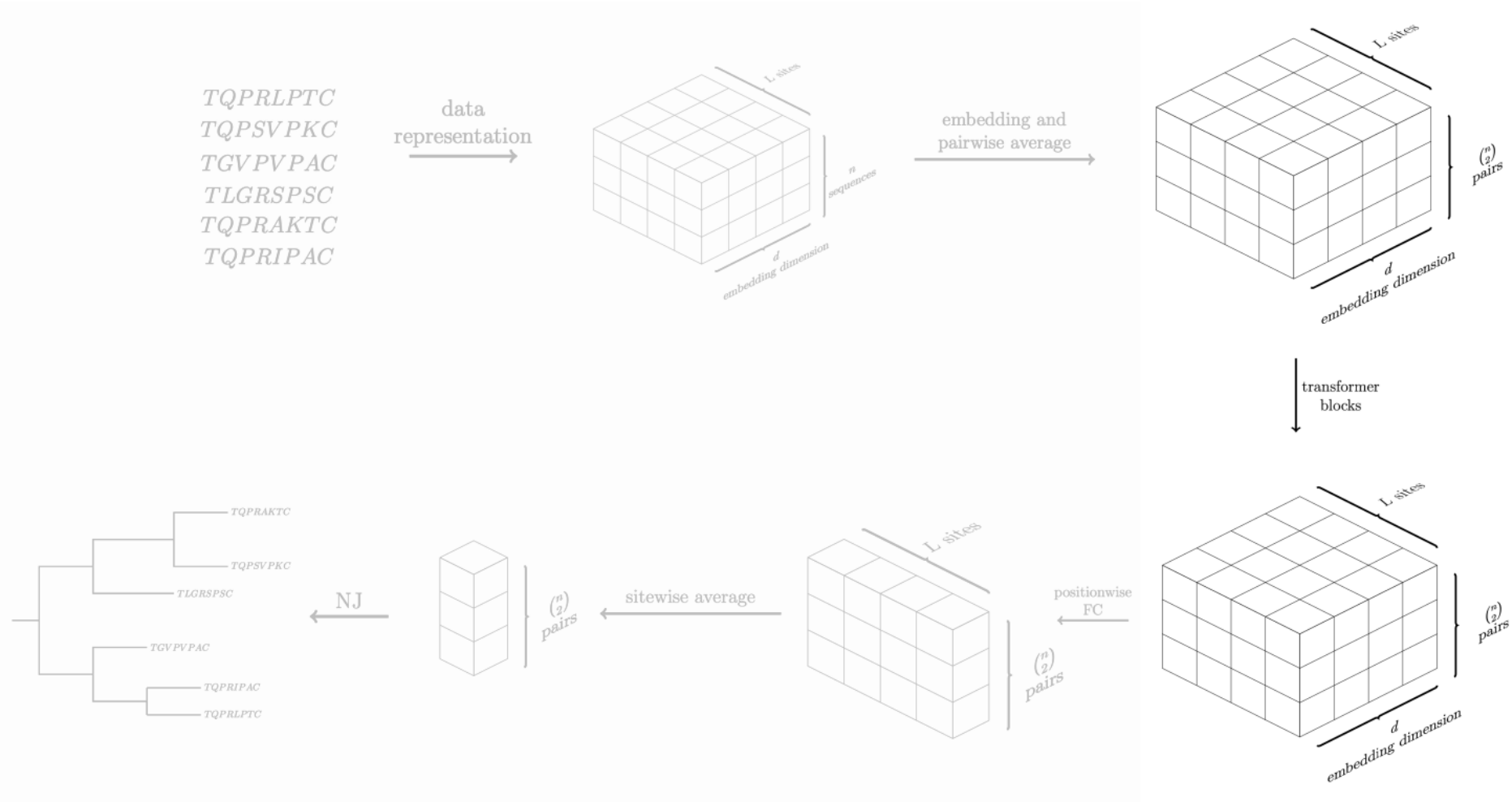
- output tree should be the same (invariance):



Issues:

- This has **no reason to be true in general** (*e.g.* linear function).
- Need to retain some expressivity.
  E.g. average provides invariance but discards a lot of information.

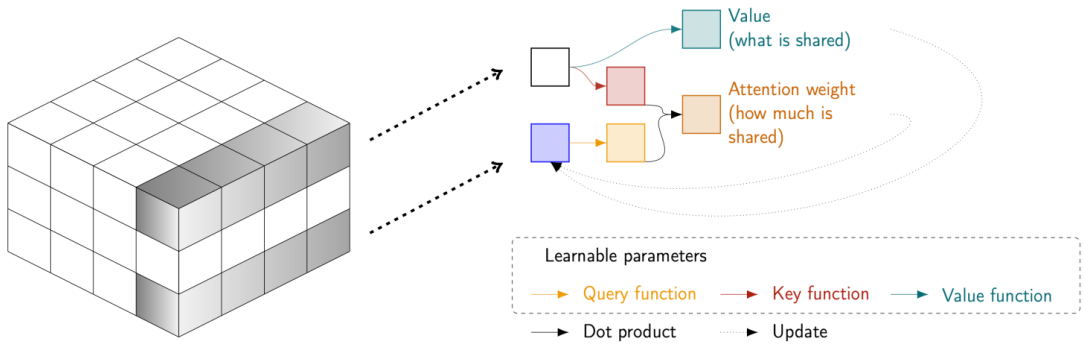
Expressive, permutation-equivariant functions with MSA Transformers



Self-attention in a nutshell

General idea

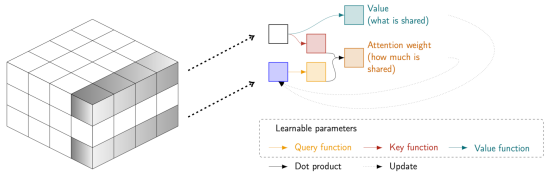
- Takes as input an unordered set.
- Updates each element as a linear combination of all of them.
- Output is a new representation of the same set. Iterate.



Updates

- Update relies on three learnable functions: Query, Key, Value.
- Query(a) and Key(b) determine the weight w_{ab} of b in the update of a.
- a is replaced by $\sum_e w_{ae}$Value(e)

The effect of self-attention



"Attention is all you need" (Transformer paper, Vaswani *et al.* 2017)

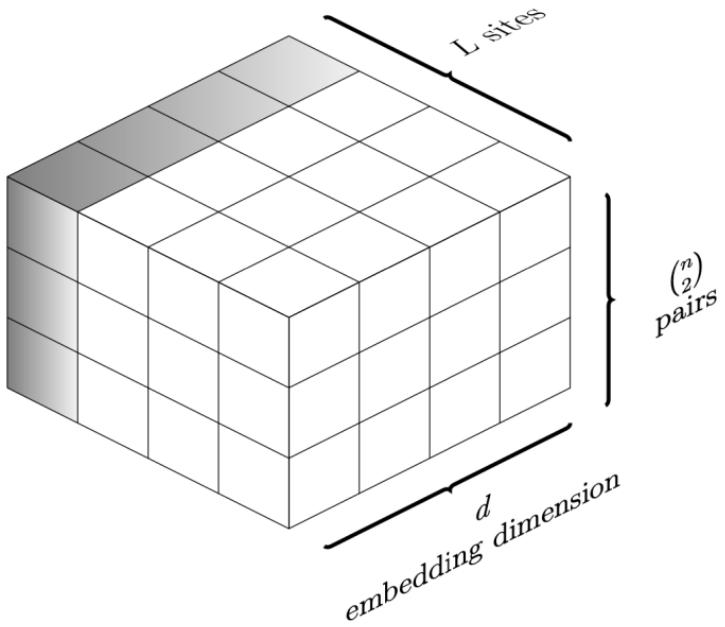
- Query and Key provide **attention weights**: how much a should pay attention to b in its update.
- Major impact in the ML literature.

Back to our issues

- All three functions act on elements: provides equivariance, modularity to any cardinal.
- Starts from independant representations, enhanced by information from all other pairs at each iteration
  $\rightarrow$ **Iteratively builds a set-aware representation for each pair.**

Axial attention

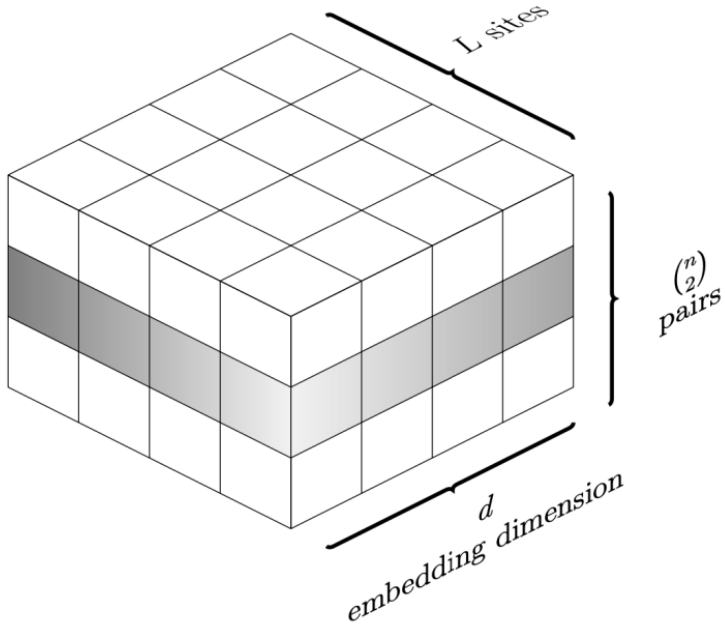
- We need equivariance both across pairs and sites.
- Alternate between column- and row-wise attention.



For each site, update each pair using all others.

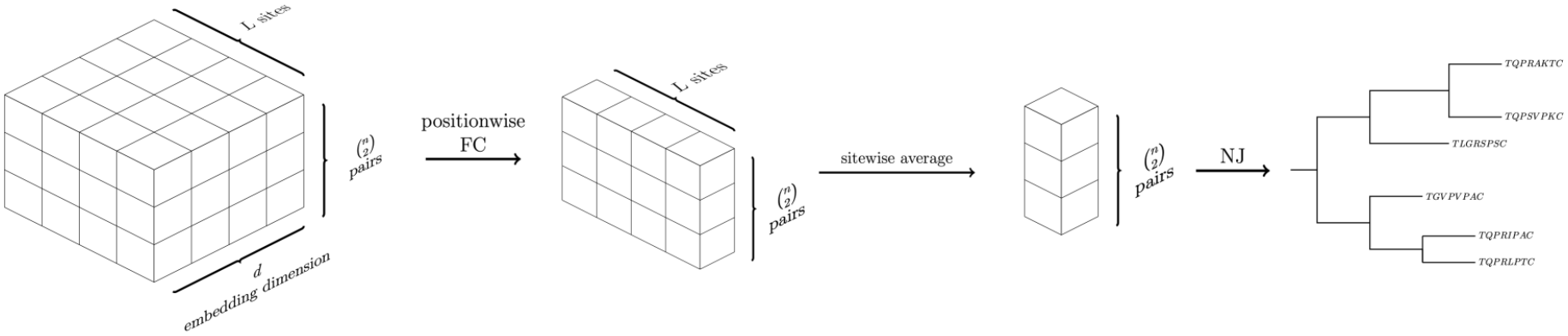
Axial attention

- We need equivariance both across pairs and sites.
- Alternate between column- and row-wise attention.



For each pair, update each site using all others.

Final steps (after the transformer blocks)

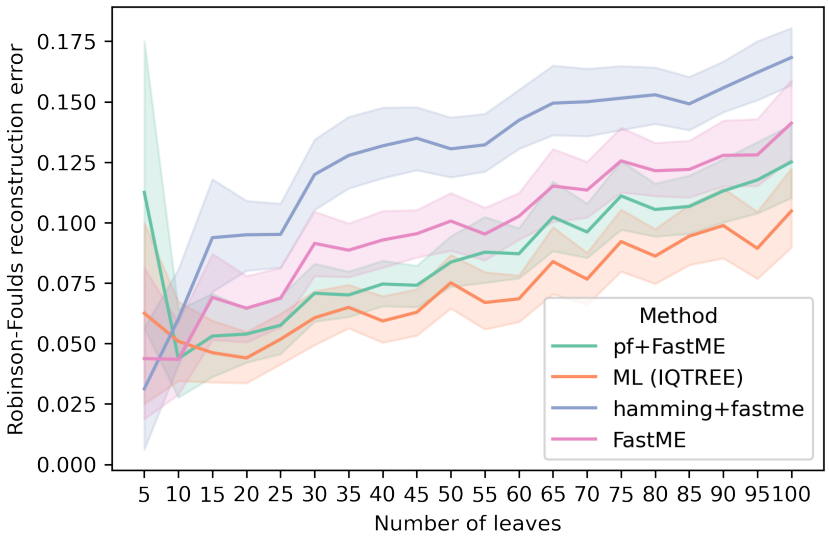


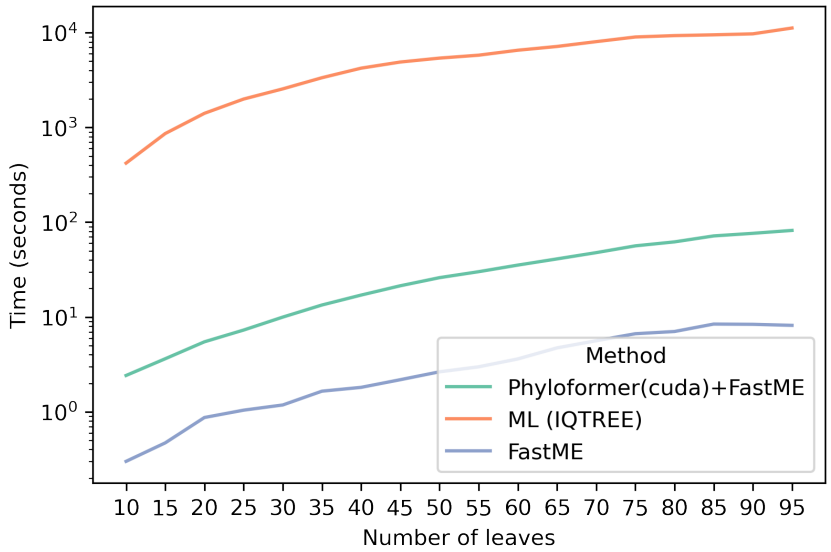
- Pool across sites to obtain a single value per pair.
- Loss function happens at this level:
  compare to true distance on simulated data, backpropagate.
- Representation is optimized to yield good distance estimates.
- Then use a distance method to build the tree (not end-to-end).

Results: a trade-off between distance and ML

Setting

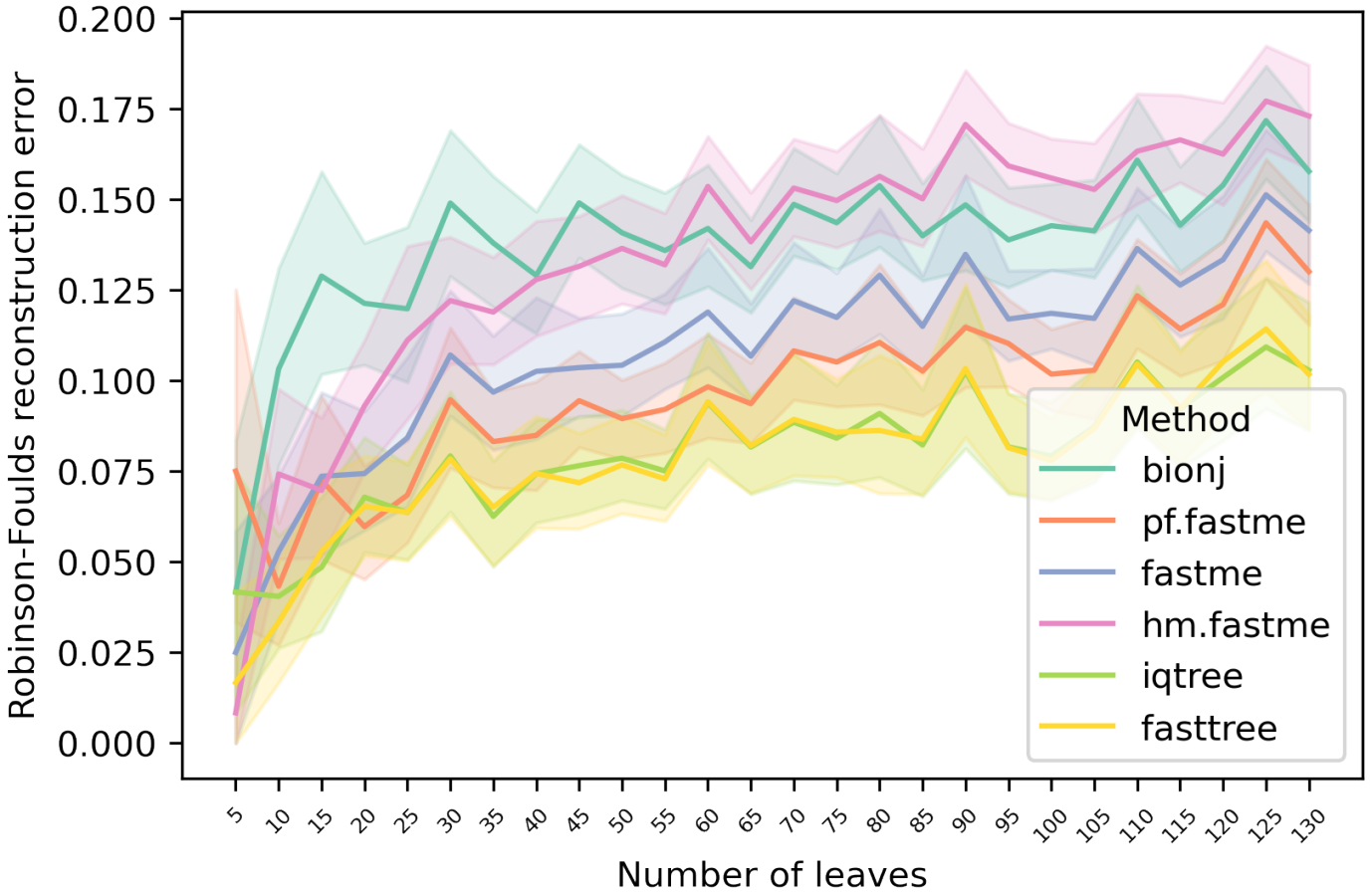
- Train on $200,000$ alignments of $n = 20$, $L = 200$.
- BD-generated trees, AliSim+(LG-GC) MSAs.





- Intermediate performance between distance- and ML-based methods.
- 100x faster than ML, 10x slower than distance.
- Much more memory intensive ($\sim$3.5Gb for 100 leaves).

The current setting may be too easy



- FastTree is actually as good as IQTREE on current simulations.
- Not easy to find a hard but realistic setting.
- Phyloformer is still $\times 2$ faster than FastTree.

Discussion

Summary

- Exploit self-attention to predict evolutionary distances among homologous sequences.
- Currently: intermediate trade-off between likelihood and distance methods.

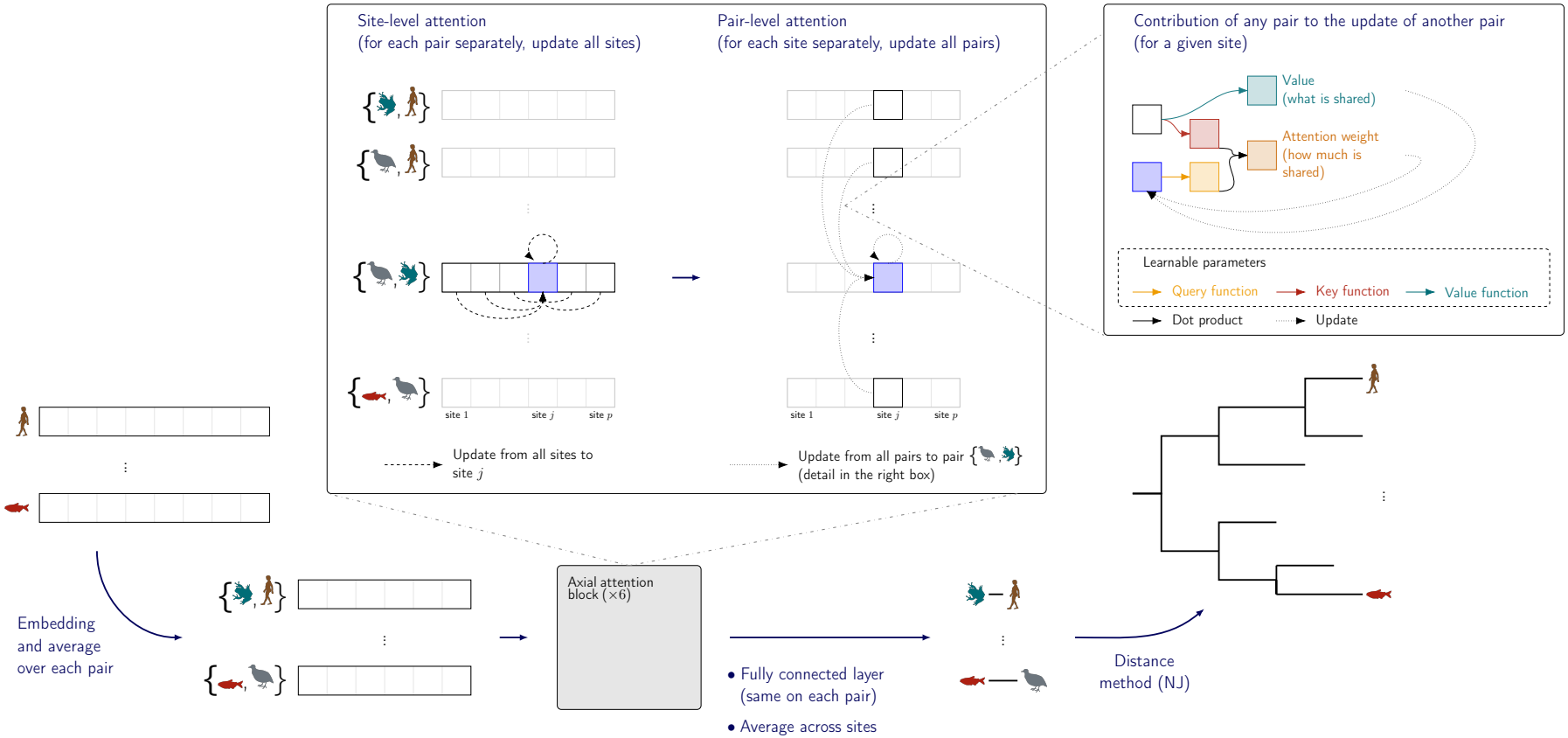
Current priorities

- Robustness: model, training data.
- Uncertainty assessment.

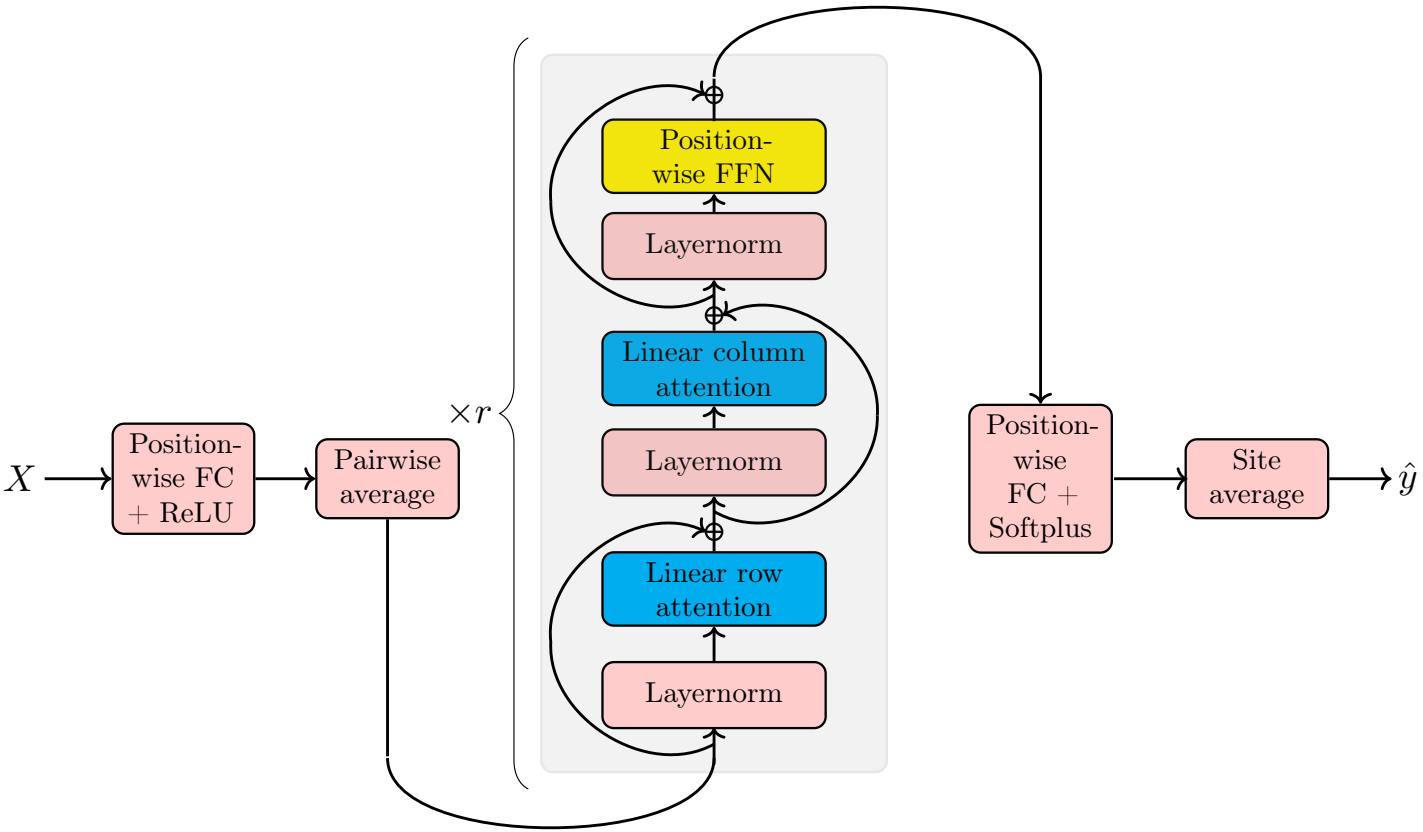
Future work

- Accuracy, scalability.
- More complex evolution models, indels.
- Related problems: reconciliation, diversification, phylodynamics...

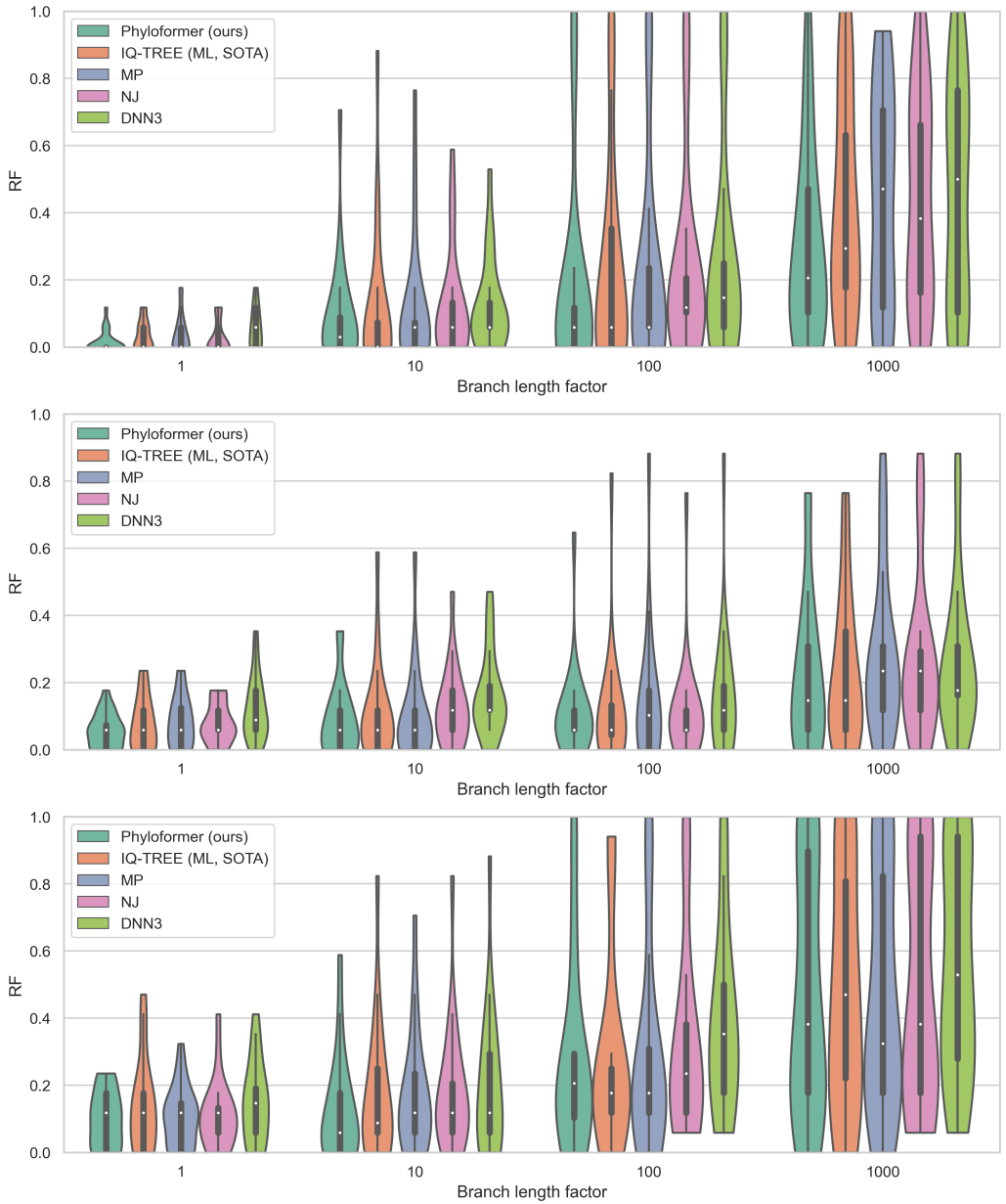
Thanks!


Site-level attention
(for each pair separately, update all sites)
Pair-level attention
(for each site separately, update all pairs)
site 1
site j
site p
Update from all sites to site j
Update from all pairs to pair
(detail in the right box)
Contribution of any pair to the update of another pair
(for a given site)
Value
(what is shared)
Attention weight
(how much is shared)
Learnable parameters
Query function
Key function
Value function
Dot product
Update
Embedding and average over each pair
Axial attention block (×6)
• Fully connected layer (same on each pair)
• Average across sites
Distance method (NJ)

Architecture

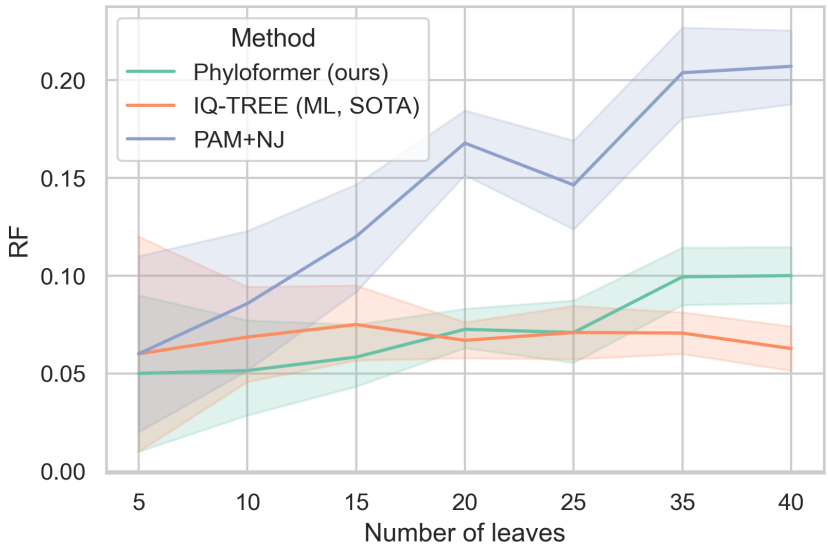


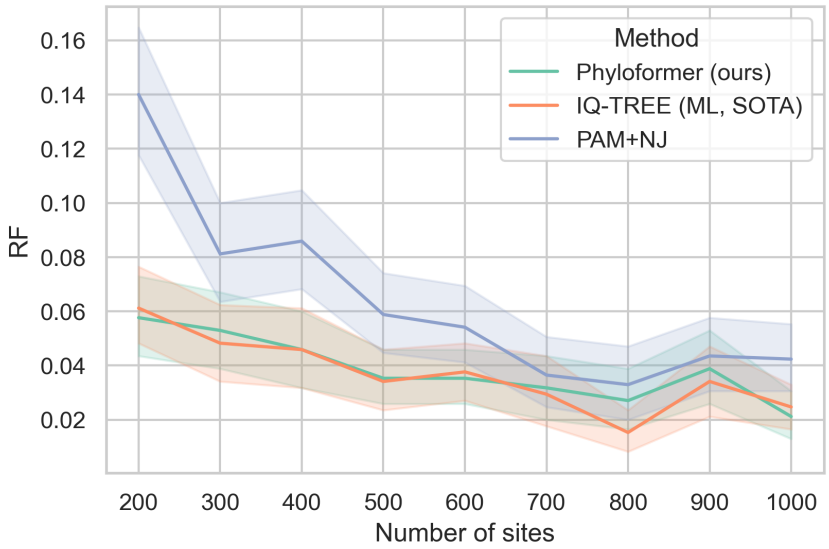
Evosimz model



- Complex model: 9 different substitution matrices, heterogeneities across sites and branches.
- 12 different parameters combinations. Phyloformer trained only on the easiest.
- Best performances across all methods on 9 out of the 12 datasets.

WAG model





Training on PAM, testing on WAG: same trend.