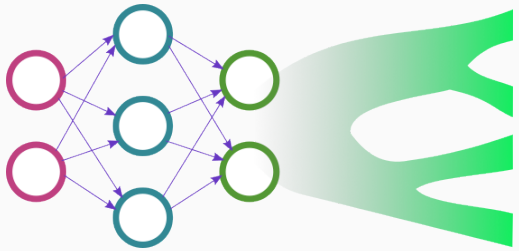
Likelihood-free inference of phylogenetic tree posterior distributions

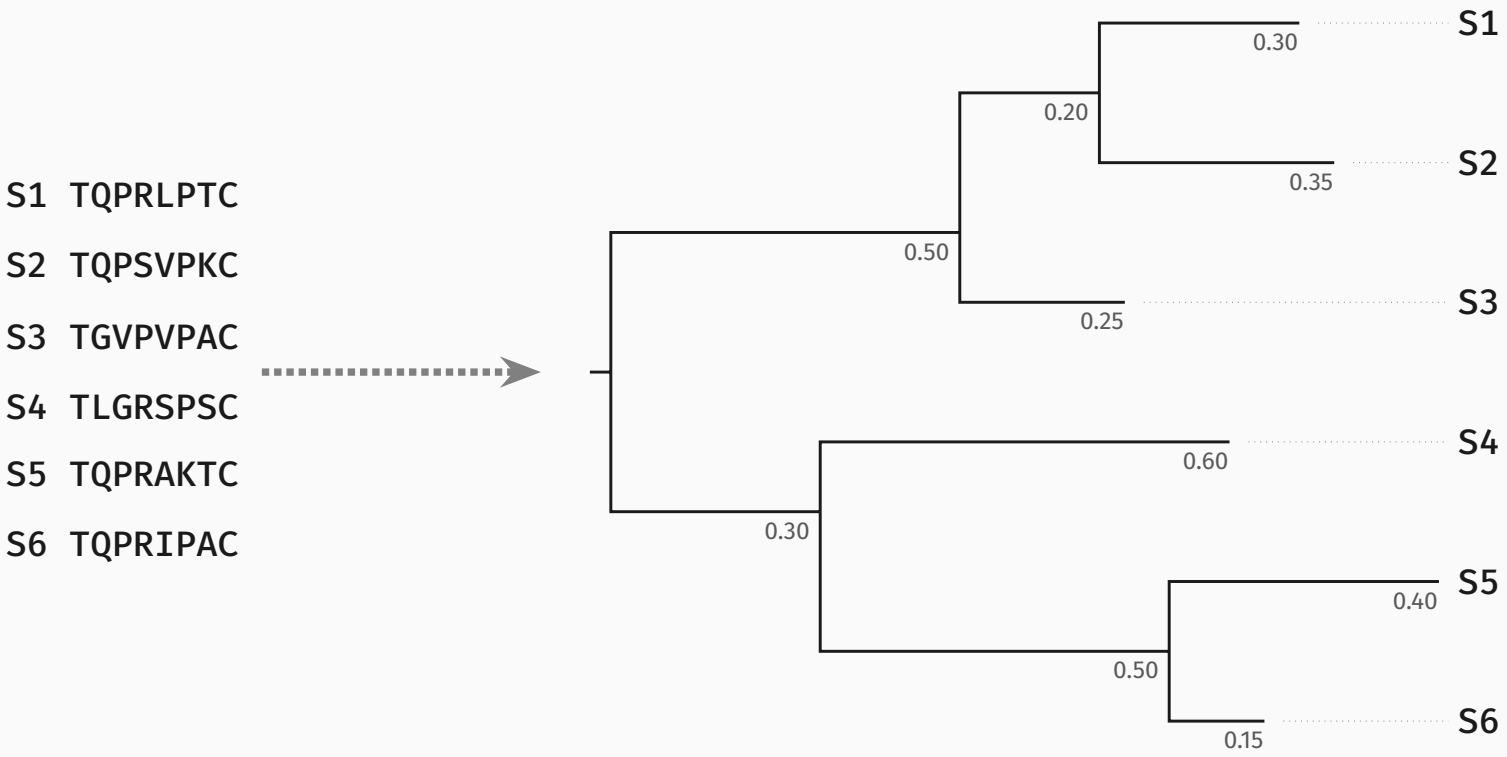


Luc Blassel, Nicolas Lartillot, Bastien Boussau, Laurent Jacob

LEGEND 2025 - Dec. 11th

CQSB SORBONNE UNIVERSITÉ LBBE cnrs

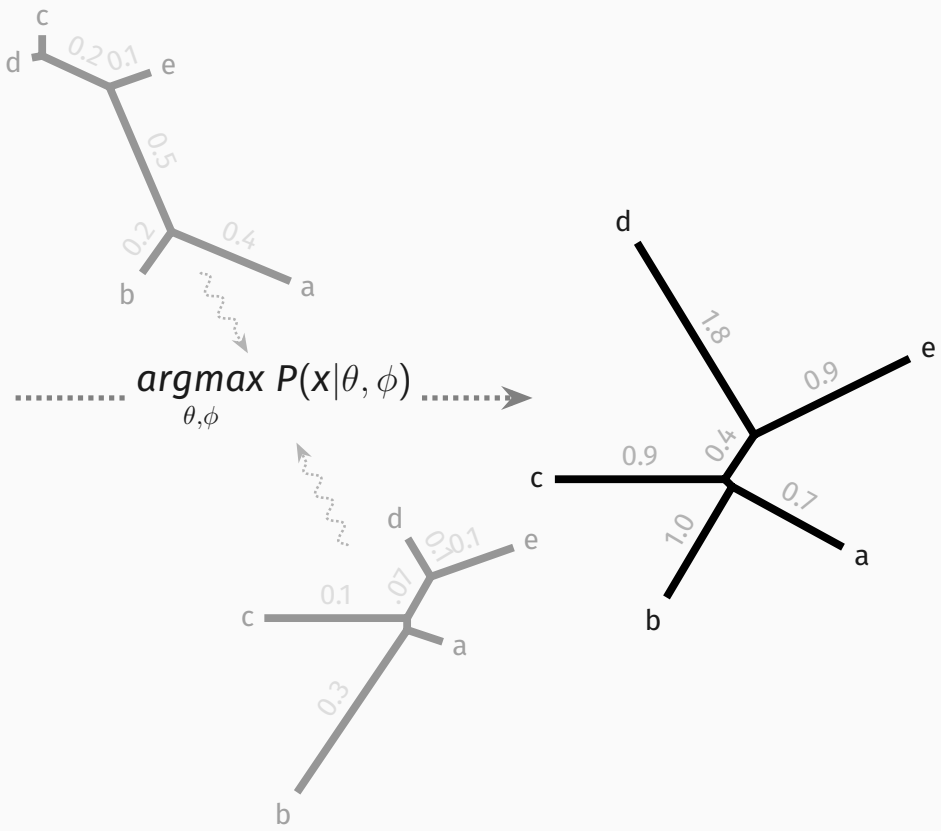
Context - Phylogenetic inference



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

Context - Likelihood-based tree reconstruction

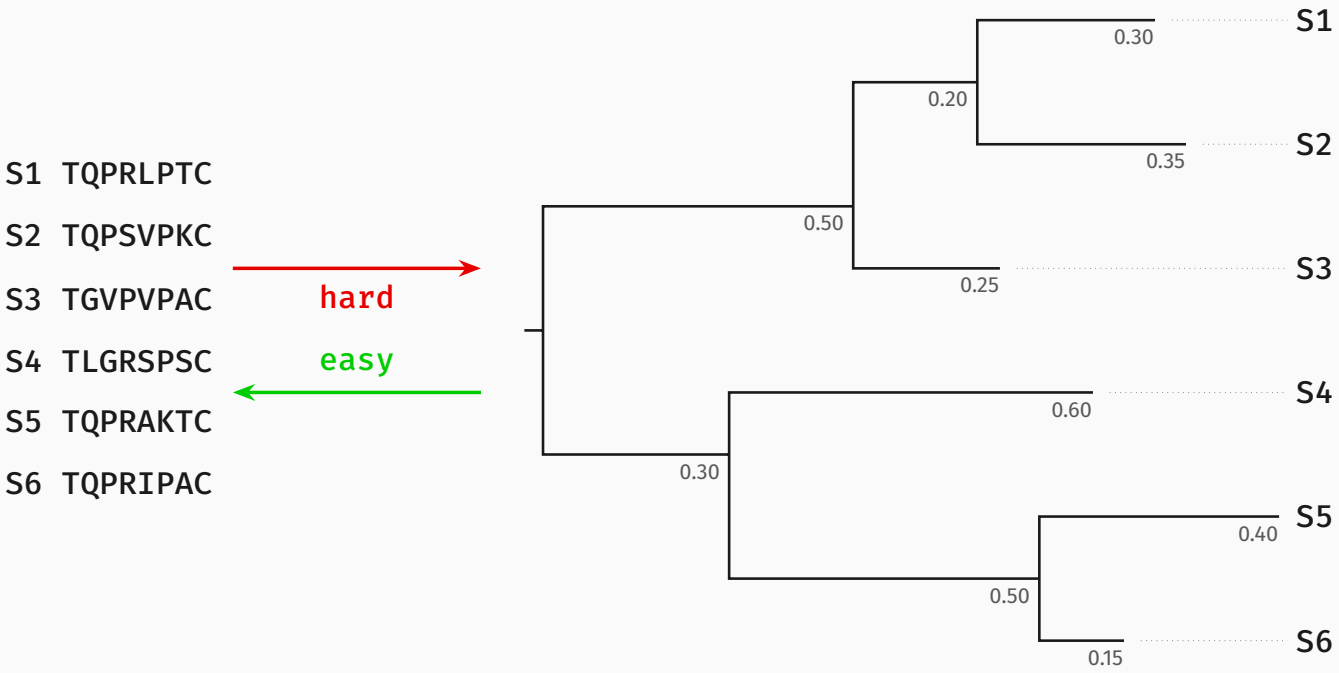
a TQPRLPTC
b TQPSVPKC
c TGVPVPAC
d TLGRSPSC
e TQPRAKTC



- **accurate** but **slow**
- $P(x|\theta,\phi)$ must be **computable**

x : MSA, $\theta = (\tau, \ell)$: Phylogenetic tree, ϕ : Evolution model Felsenstein 1993; Kleinman et al. 2010

Context - Simulation-based/Likelihood-free inference



- We can simulate many (tree, MSA) pairs
- Can we **learn** the mapping **from MSA to tree**?

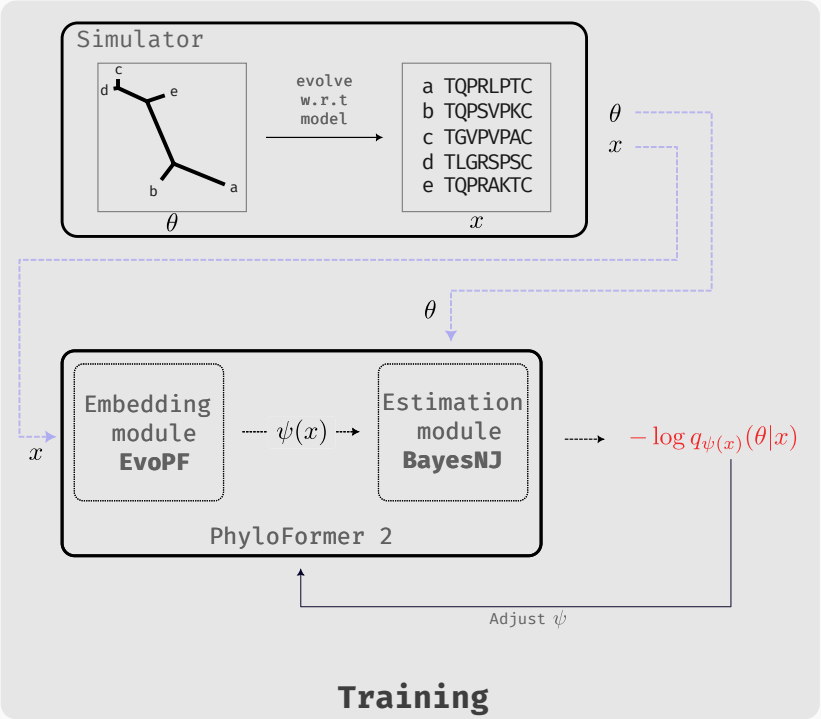
How do we do end-to-end phylogenetic inference?

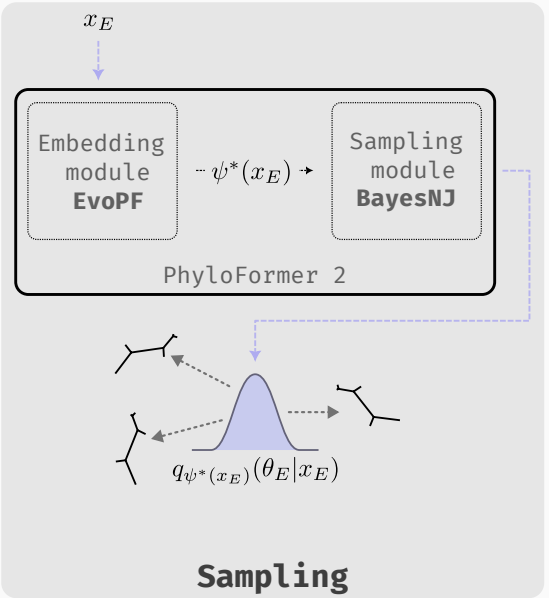
Methods - Neural posterior estimation (NPE)

- Given a **probabilistic model** $p(x|\theta)$ with some prior $p(\theta)$
- We want to **estimate the posterior**: $p(\theta|x)$
- We build $q_\psi(\theta|x)$ a **family** of distributions **parametrized** by ψ (our NN)
- We find $q_{\psi^*} = \underset{\psi}{\text{argmin}}\ \mathbb{E}_{p(x)}[KL(q_\psi(\theta|x)||p(\theta|x)]$
- In practice we **maximize** $\mathbb{E}_{p(x,\theta)}[\log q_{\psi(x)}(\theta|x)]$ by **sampling** from $p(x, \theta)$
- The two are **formally equivalent**[1]

[1] Radev et al. 2020 $\quad x$: MSA, $\quad \theta = (\tau, \ell)$: Phylogenetic tree, $\quad \psi(x)$: NN applied to x

Methods - Phyloformer 2 and NPE





- During **training** find $\psi^* = \underset{\psi}{\operatorname{argmin}} \; -\sum_i \log q_{\psi(x_i)}(\theta_i|x_i)$
- At **inference** time **sample** from: $q_{\psi^*(x_E)}(\theta_E|x_E)$

Methods - The EvoPF module

the EvoPF module is an **adaptation** of the **EvoFormer** module from **AlphaFold2**. The tasks are **transpositions** of each other:

given input MSA ($n \times r$)

EvoFormer represent $r \times r$ relationships between sites

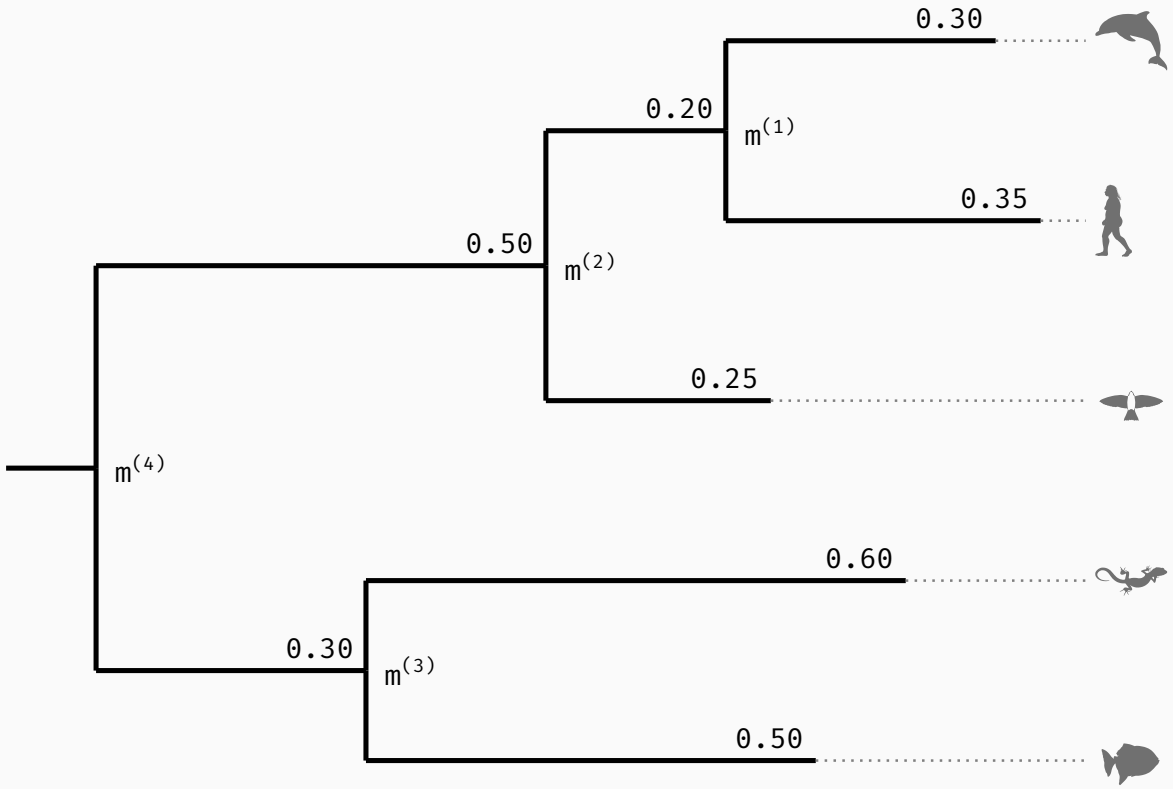
EvoPF represent $n \times n$ relationships between sequences

More expressive than MSA transformer
More lightweight than our **first attempt**, PhyloFormer

Jumper et al. 2021; Rao et al. 2021

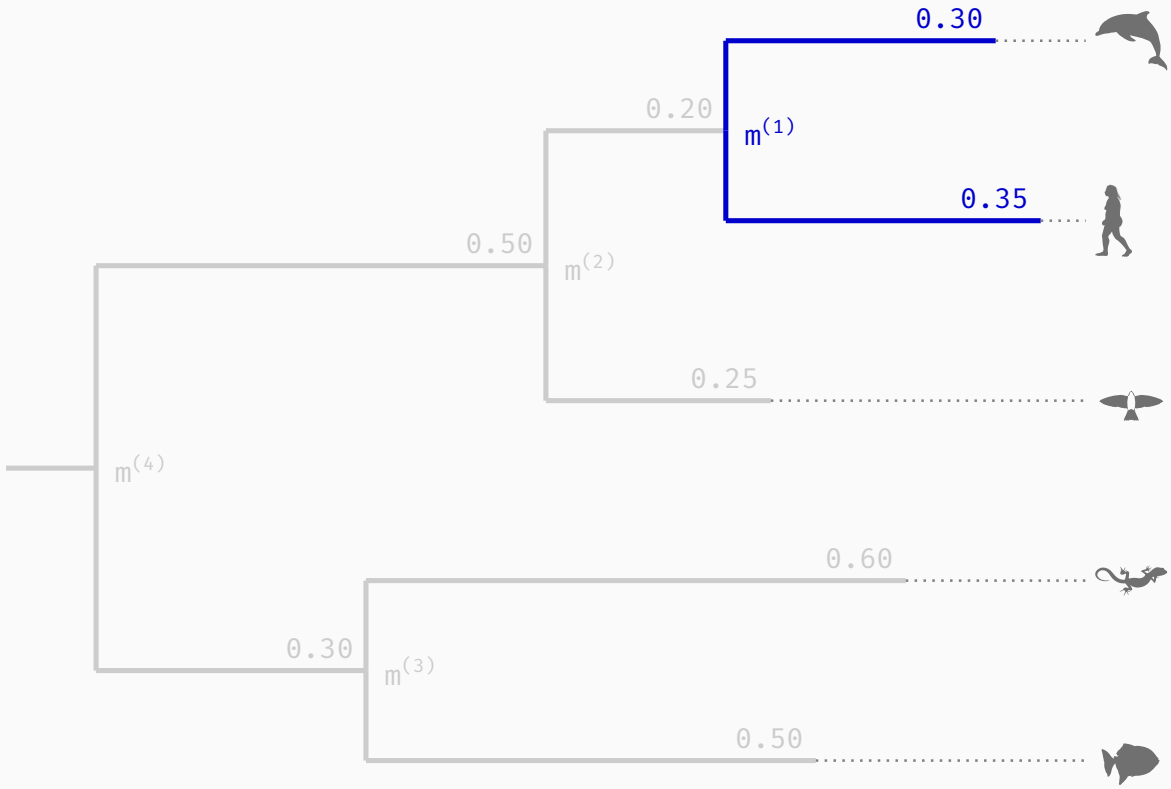
Methods - Trees are series of merges

We want to describe the following tree:



Methods - Trees are series of merges

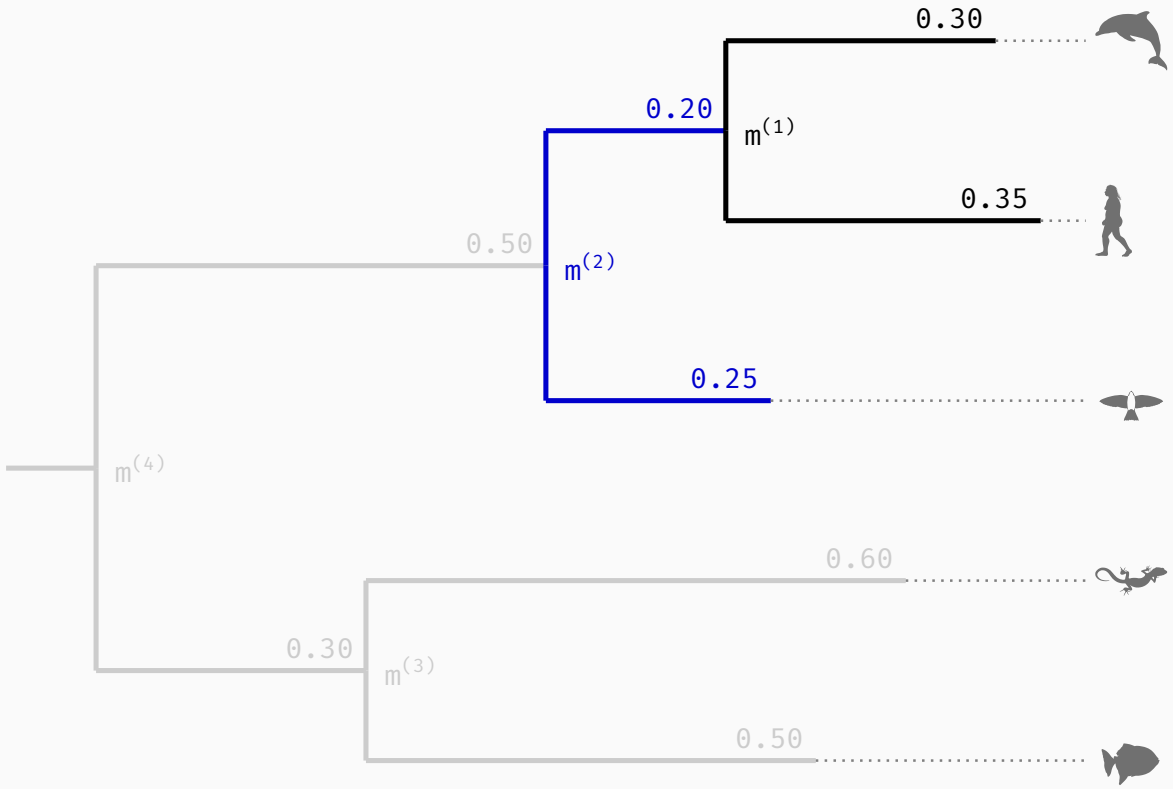
Iteratively merge shortest cherry:



$$\tau = \{m^{(1)} \qquad \}$$

Methods - Trees are series of merges

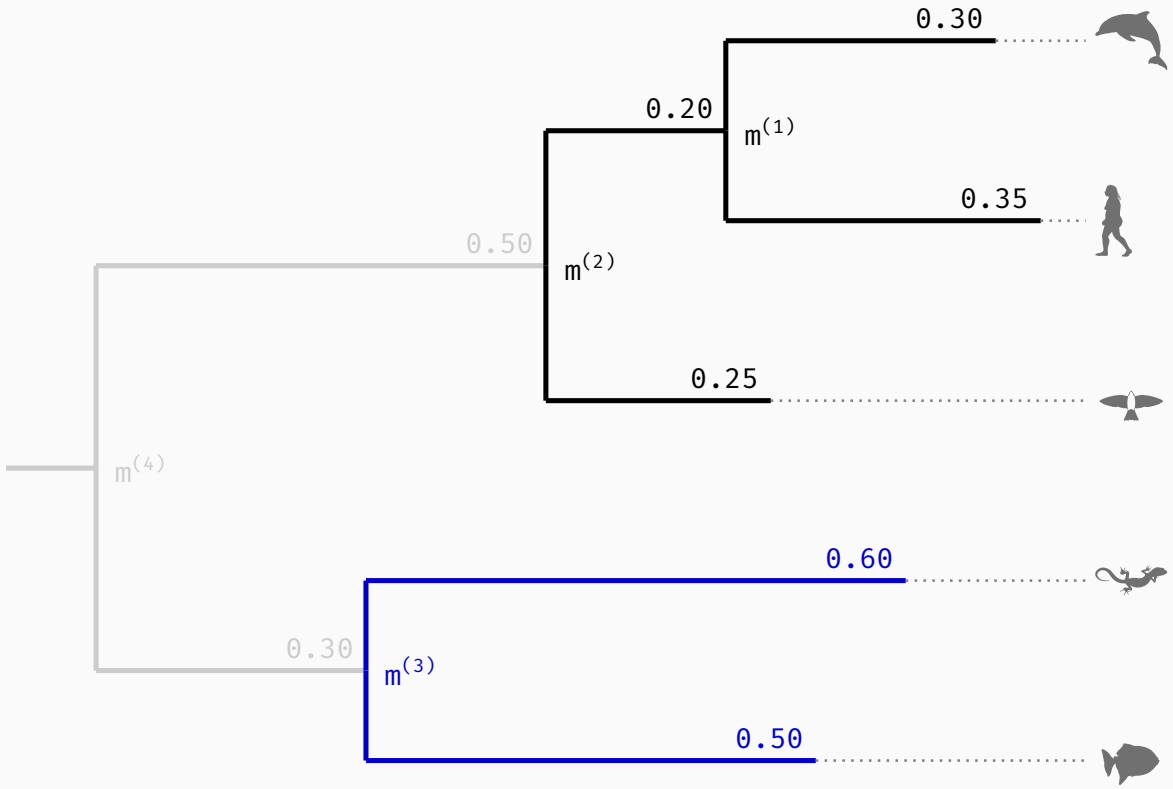
Iteratively merge shortest cherry:



$$\tau = \{m^{(1)}, m^{(2)} \qquad \}$$

Methods - Trees are series of merges

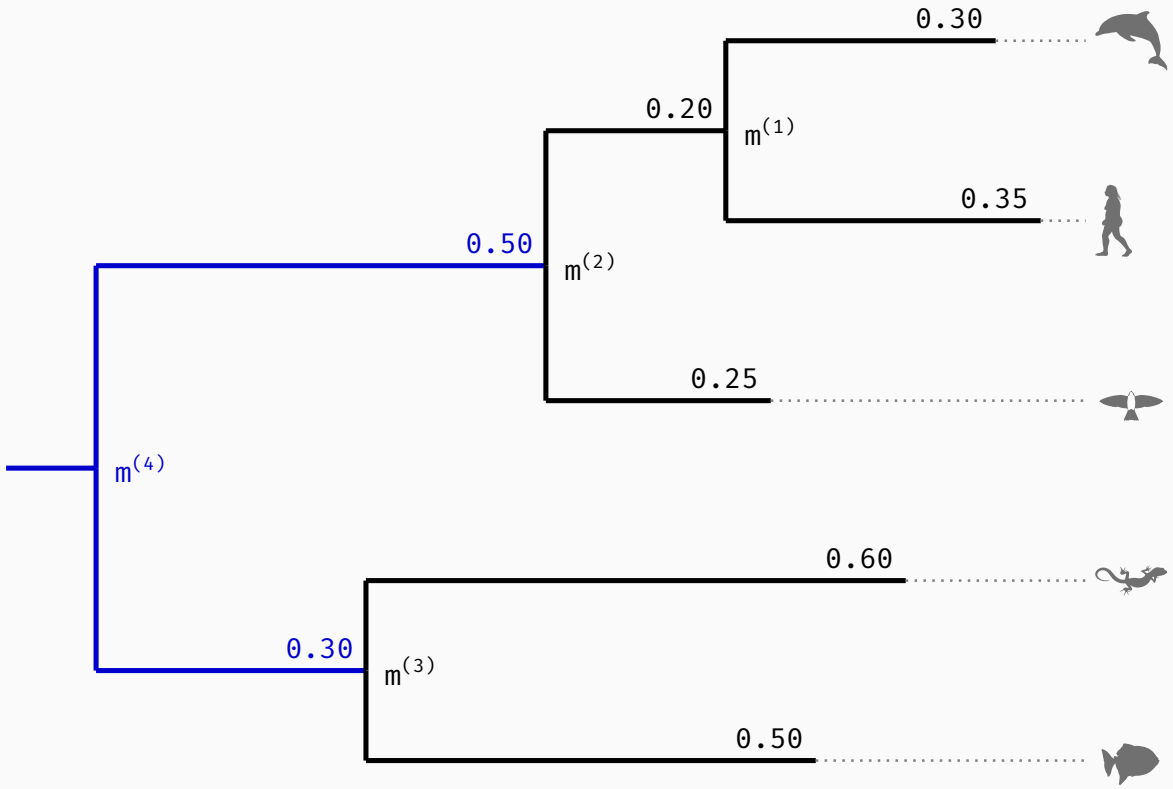
Iteratively merge shortest cherry:



$$\tau = \{m^{(1)}, m^{(2)}, m^{(3)} \quad \}$$

Methods - Trees are series of merges

Iteratively merge shortest cherry:



$$\tau = \{m^{(1)}, m^{(2)}, m^{(3)}, m^{(4)}\}$$

Methods - Approximating the posterior with BayesNJ (1)

- **Tree** is an **ordered set** of merges: $\theta : \{m^{(1)}, \ldots, m^{(N-1)}\}$
- We **factorize** $q_{\psi(x)}(\theta|x)$ as the product of successive merge probabilities:

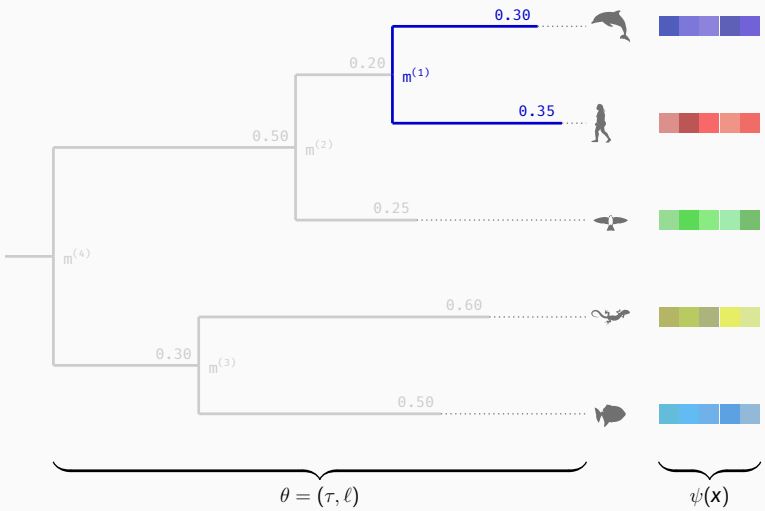
$$q_{\psi(x)}(\theta|x) = \prod_{k=1}^{N-1} q_m(m^{(k)}|m^{(<k)}) q_\ell(\ell^{(k)}|m^{(\leq k)})$$

- **Merge** probabilities have **2 components**:

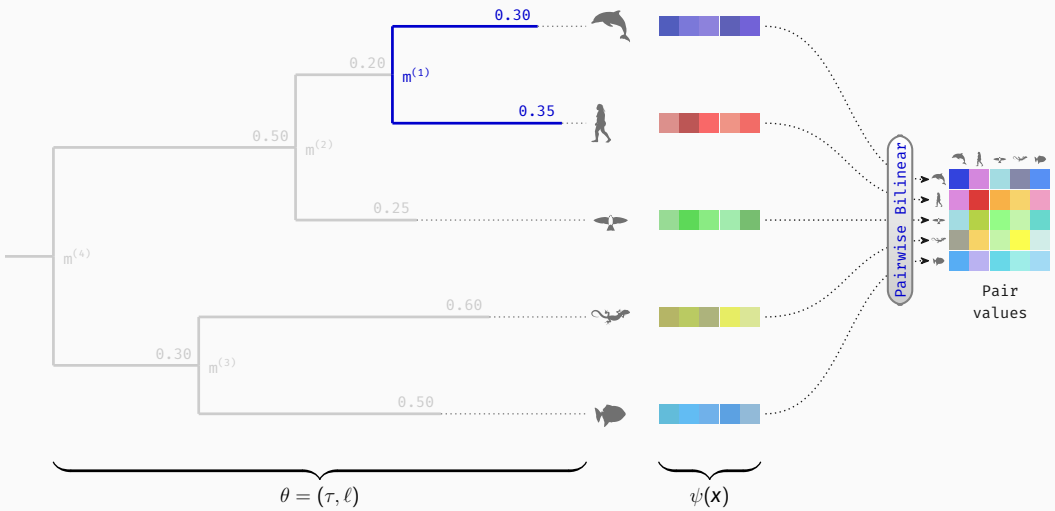
topological: $q_m(m^{(k)}|m^{(<k)})$

branch-length: $q_\ell(\ell^{(k)}|m^{(\leq k)})$

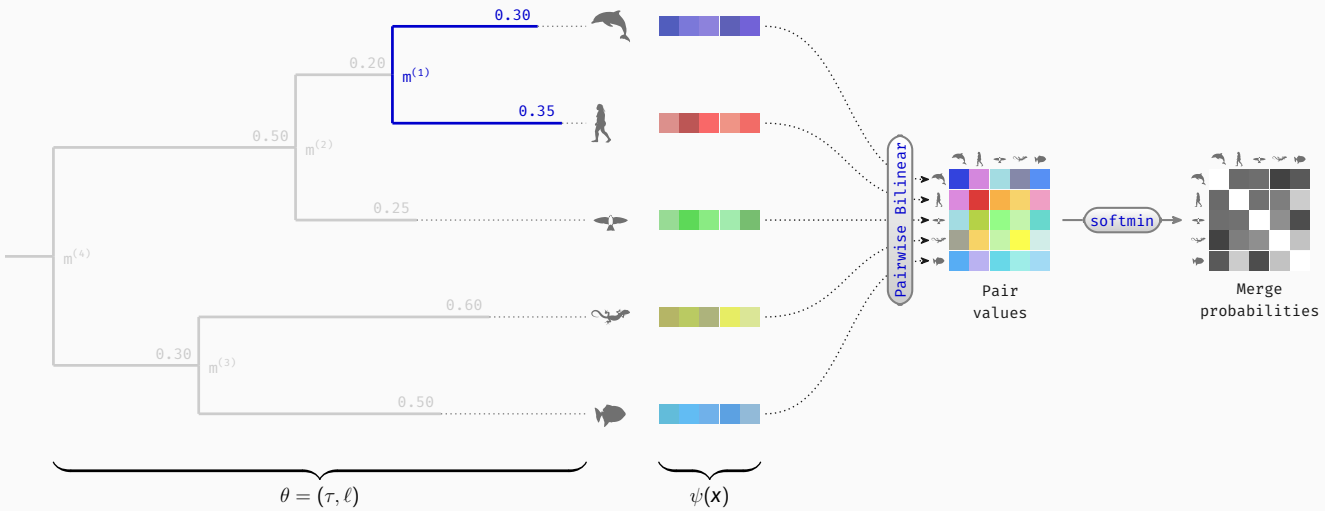
Methods - Approximating the posterior with BayesNJ (2)



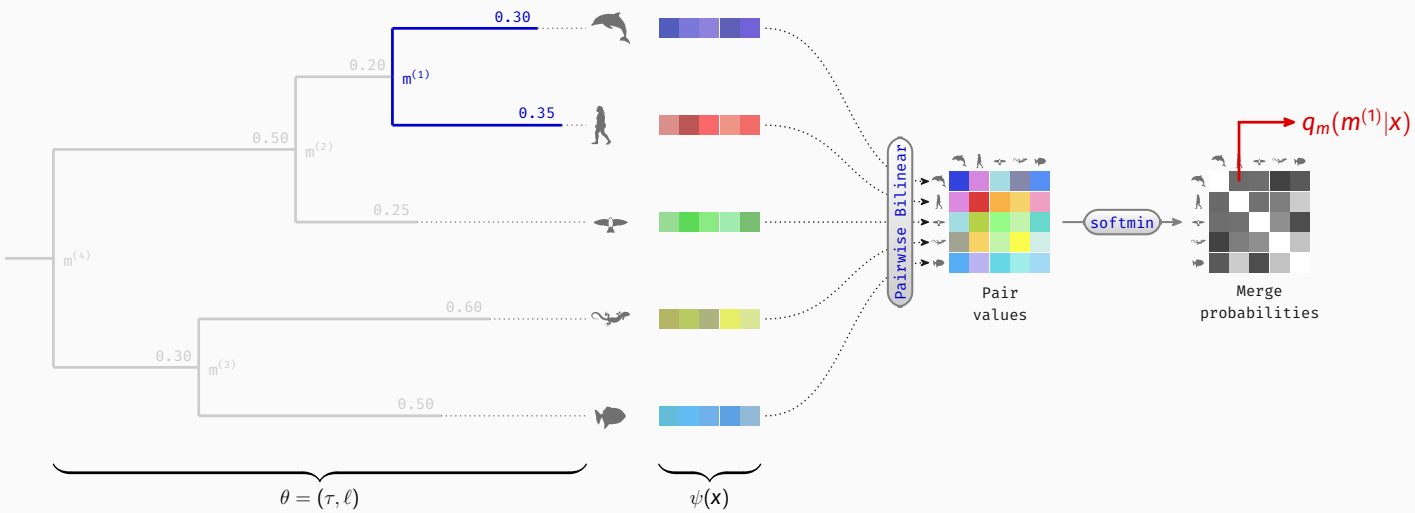
Methods - Approximating the posterior with BayesNJ (2)



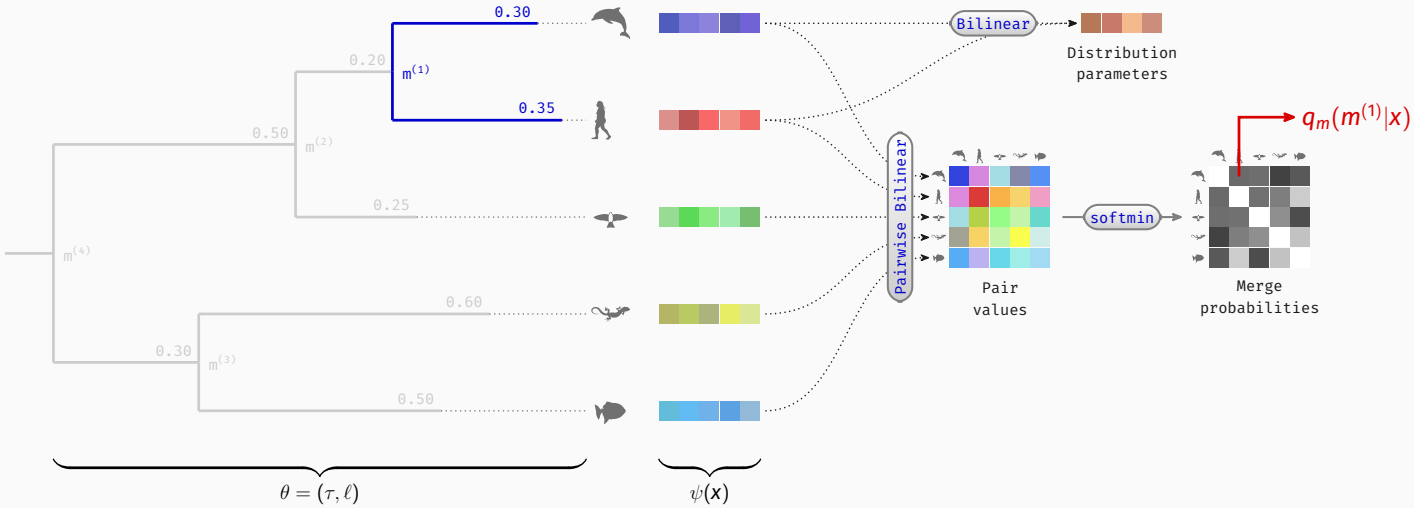
Methods - Approximating the posterior with BayesNJ (2)



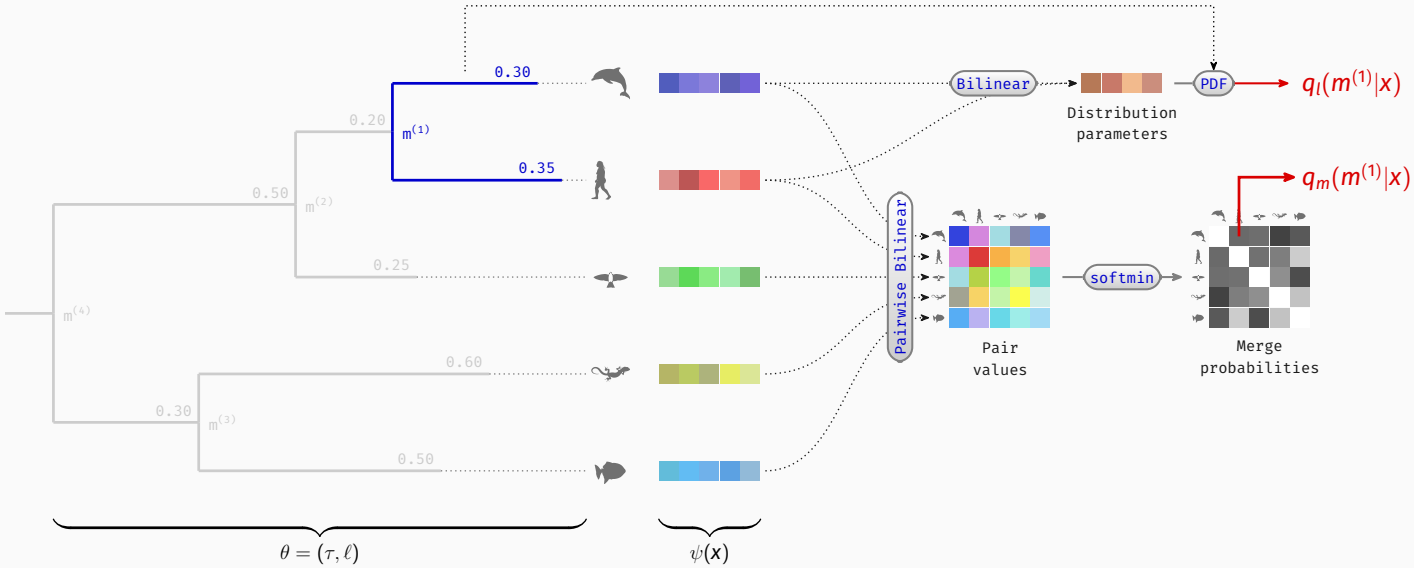
Methods - Approximating the posterior with BayesNJ (2)



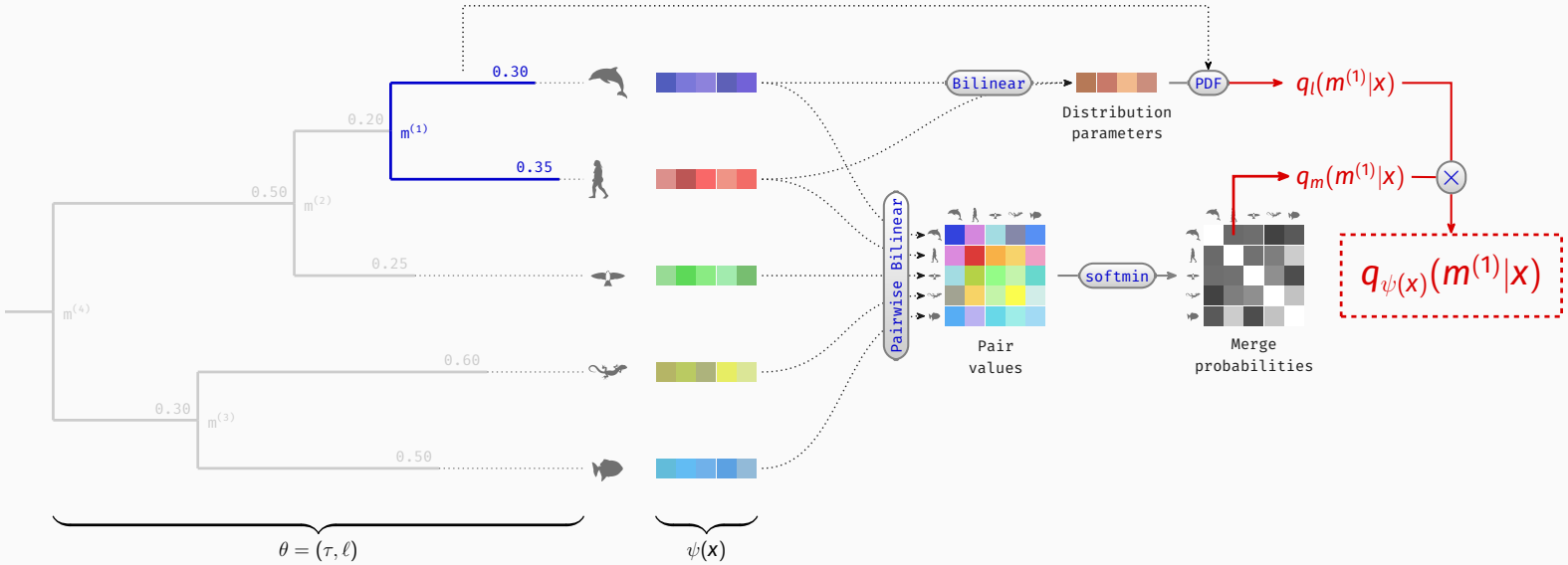
Methods - Approximating the posterior with BayesNJ (2)



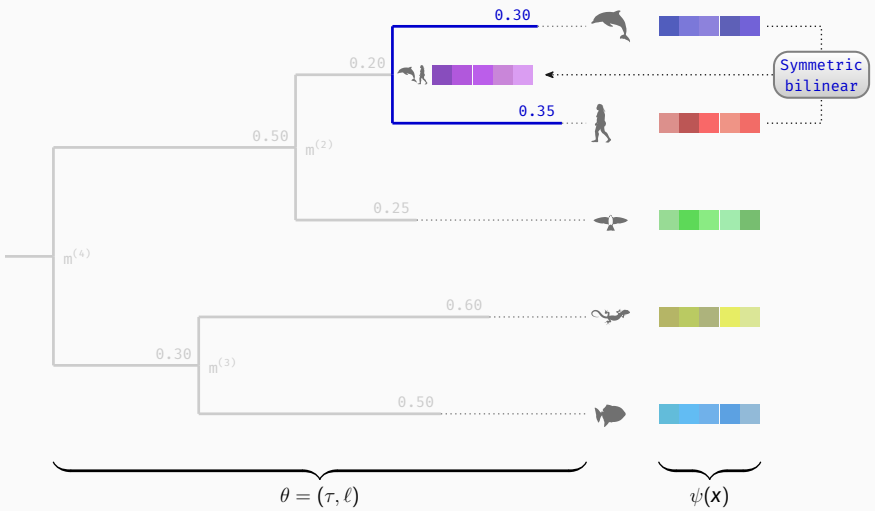
Methods - Approximating the posterior with BayesNJ (2)



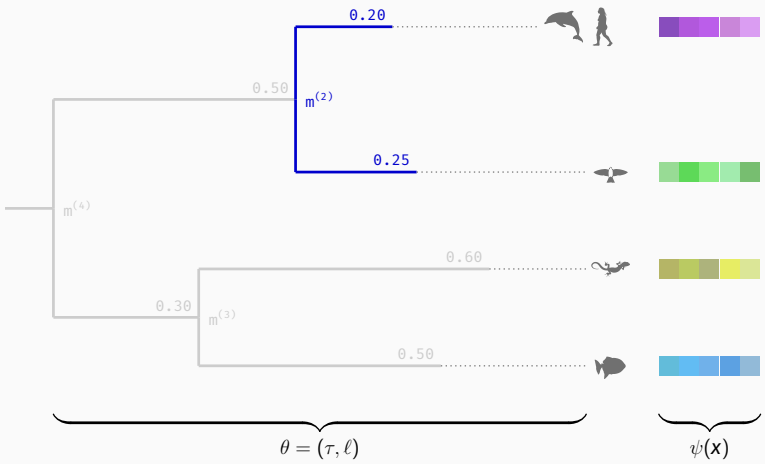
Methods - Approximating the posterior with BayesNJ (2)



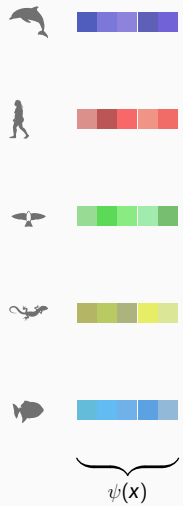
Methods - Approximating the posterior with BayesNJ (2)



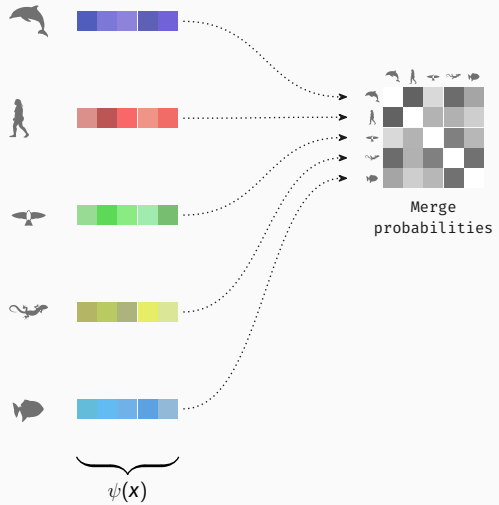
Methods - Approximating the posterior with BayesNJ (2)



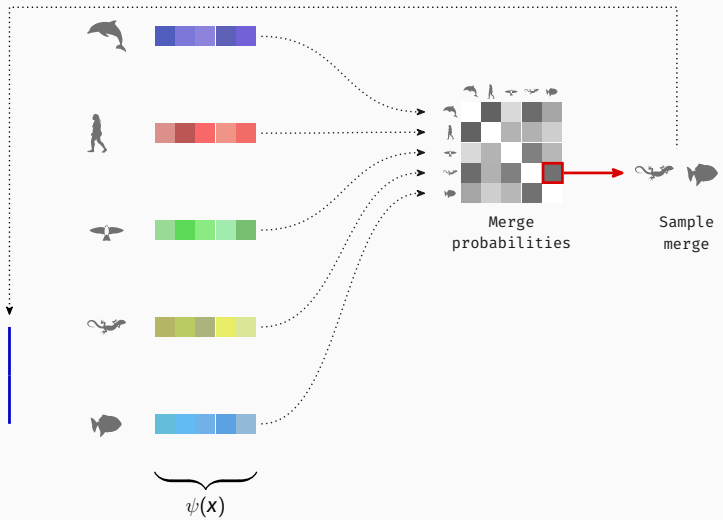
Methods - Sampling from the posterior with BayesNJ



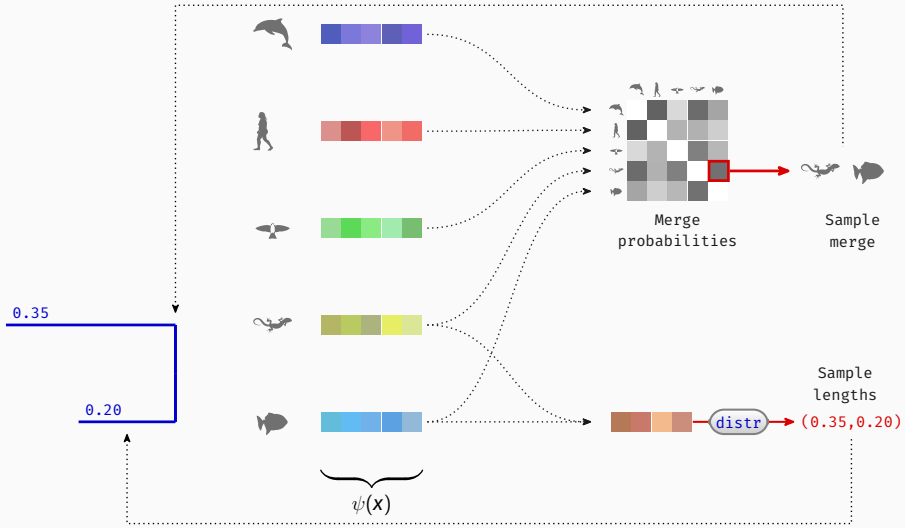
Methods - Sampling from the posterior with BayesNJ



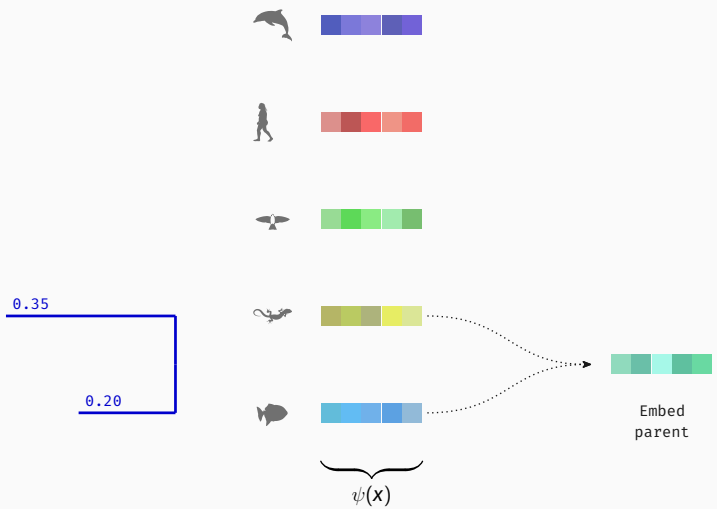
Methods - Sampling from the posterior with BayesNJ



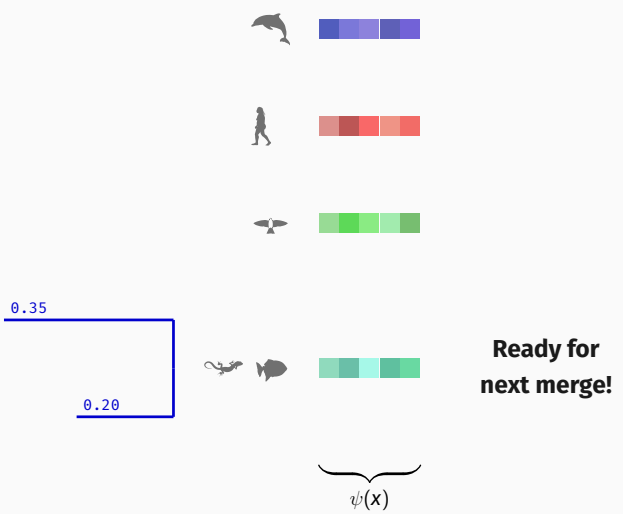
Methods - Sampling from the posterior with BayesNJ



Methods - Sampling from the posterior with BayesNJ



Methods - Sampling from the posterior with BayesNJ

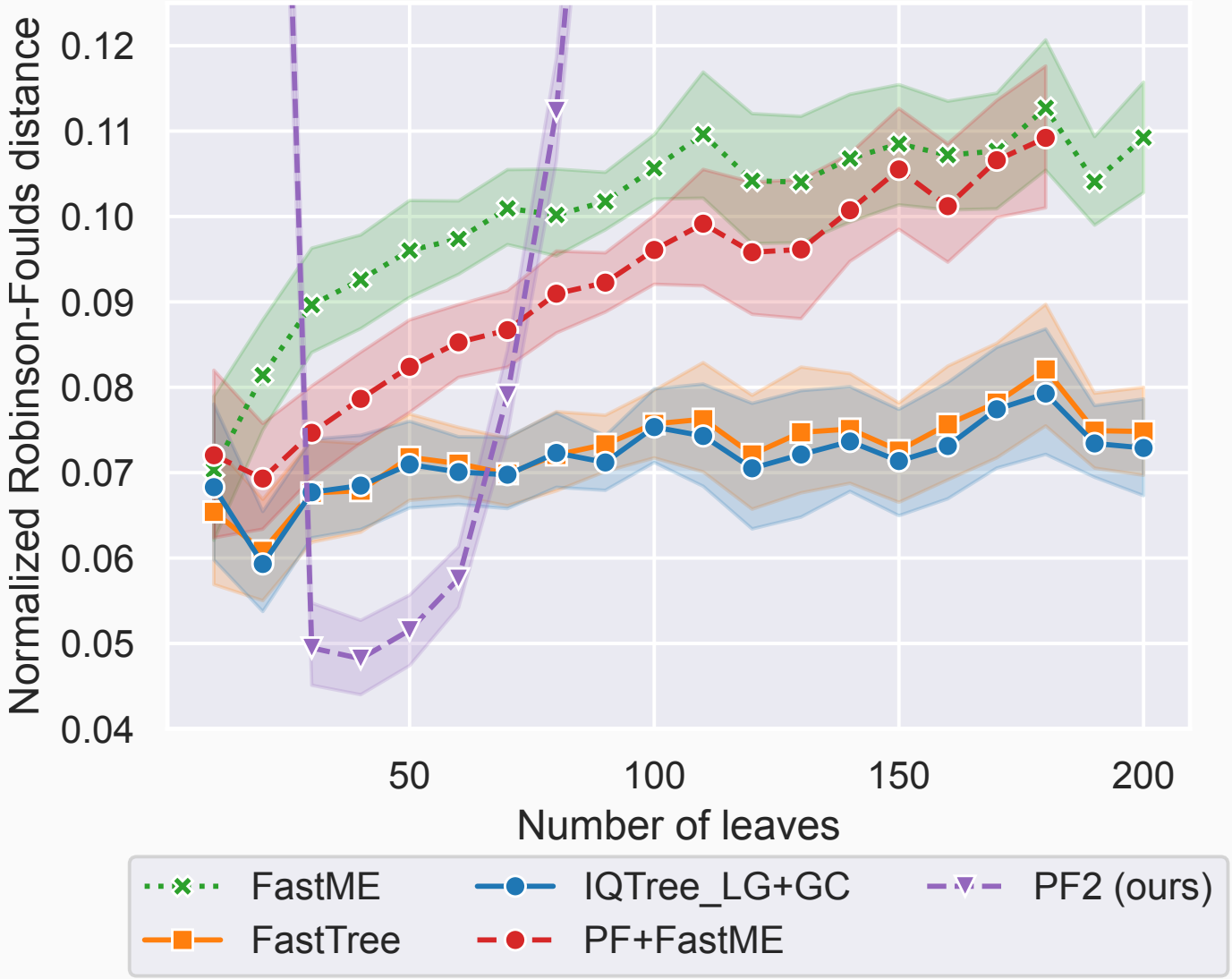


How well does Phyloformer 2 work?

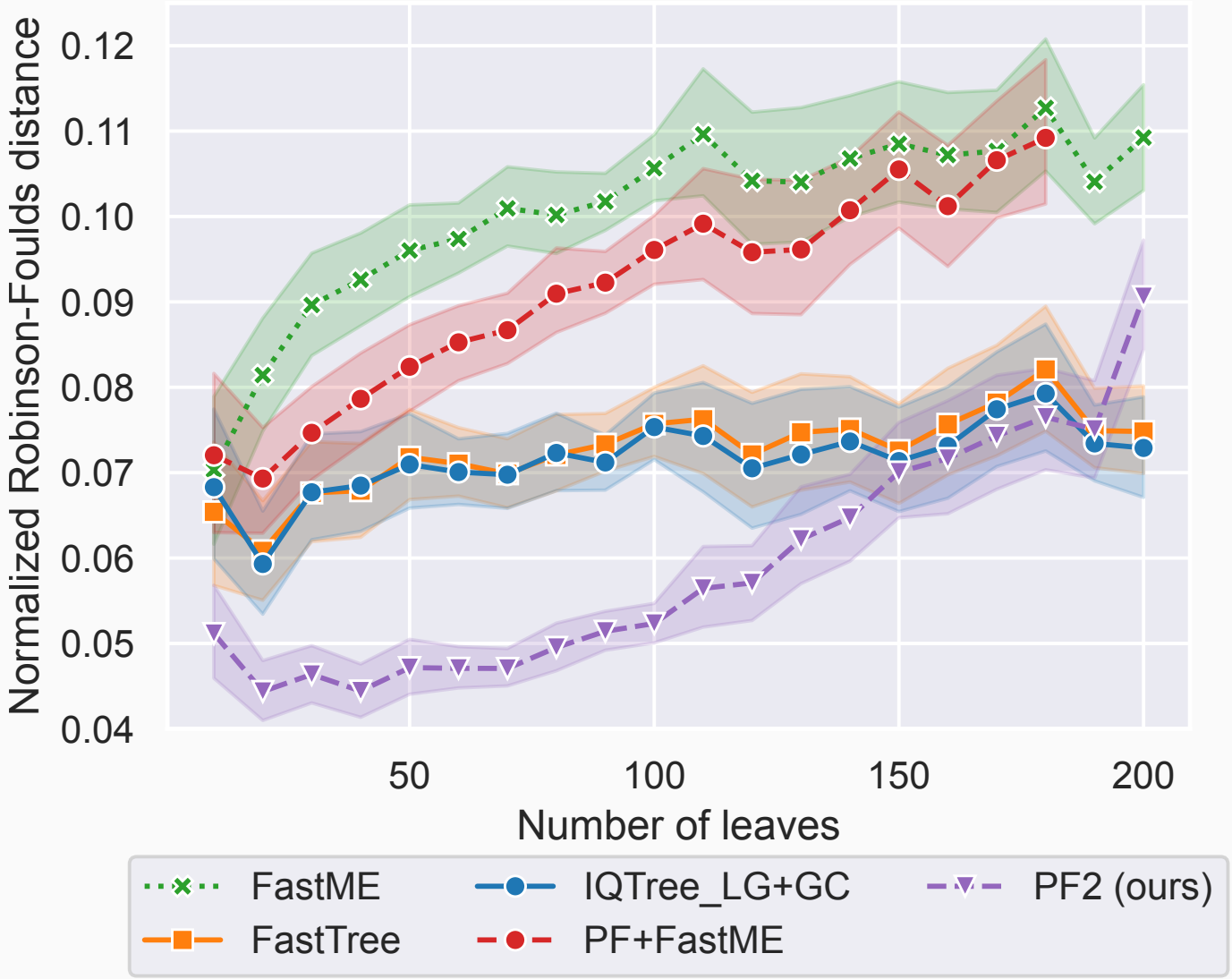
Results - Tractable-likelihood model (1)

- We train PF2 on MSAs simulated under **LG** + Γ8 rates + **indels**
  1. **Pre-train** on 8M **50-tip** trees
  2. **Fine-tune** on **[10-160]-tip** trees (20k trees per size)
- We **test** PF2 on data **simulated** under **similar conditions**
- LG has a **tractable likelihood**, this is the **best-case** for **likelihood** methods

Results - Tractable-likelihood model (2)



Results - Tractable-likelihood model (2)



Fine tuning helps **a lot**

Results - Intractable-likelihood model (1)

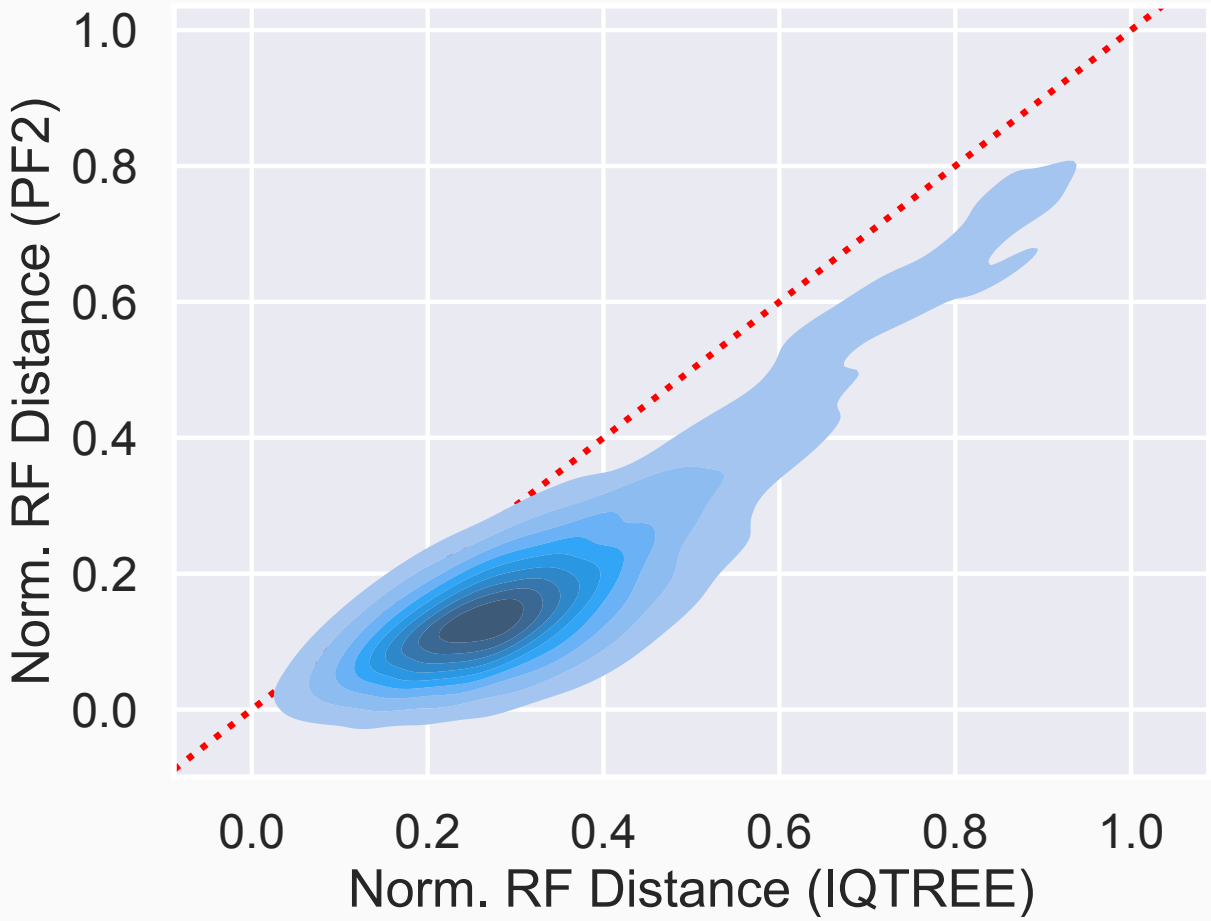
How well does this extend to **more complex** models ?

- Simulate **50-tip** trees under **same prior** as LG training
- Simulate **MSAs** along those trees under **Potts model** fit on PF00072 response-regulator PFAM family
- **Test** on under **same priors**
- **No other methods** to infer trees **under Potts** model, but still compare to **IQTree+ModelFinder**

Work done with N. Sauvage and P. Barrat-Charlaix

Pagnani and Barrat-Charlaix 2025

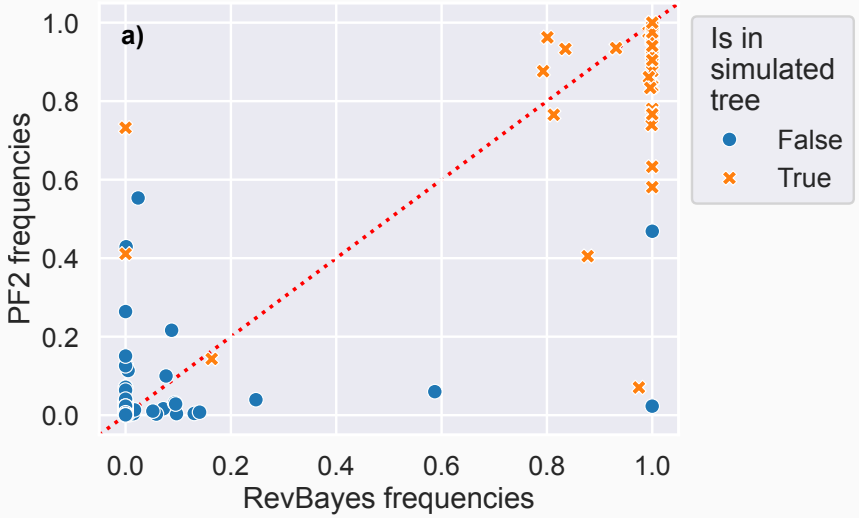
Results - Intractable-likelihood model (2)



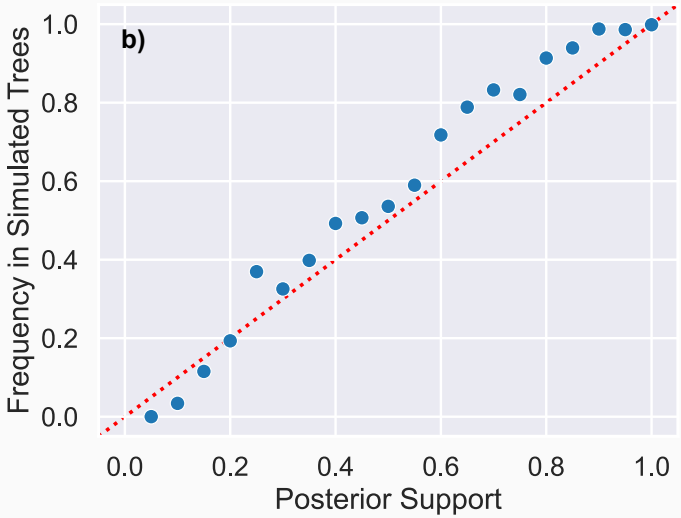
Consistently better topology prediction for **PF2**

Results - Phyloformer 2's posterior

Compared to RevBayes



Simulation-based Calibration

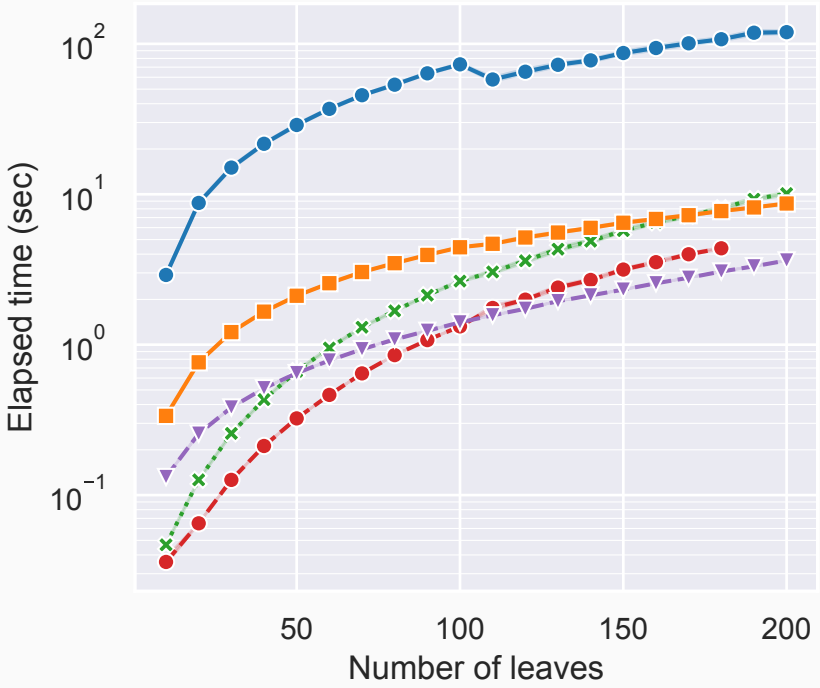


- Overall RevBayes **agrees** with PF2
- PF2 has a **smoother** distribution
- PF2 posterior are **generally well calibrated**

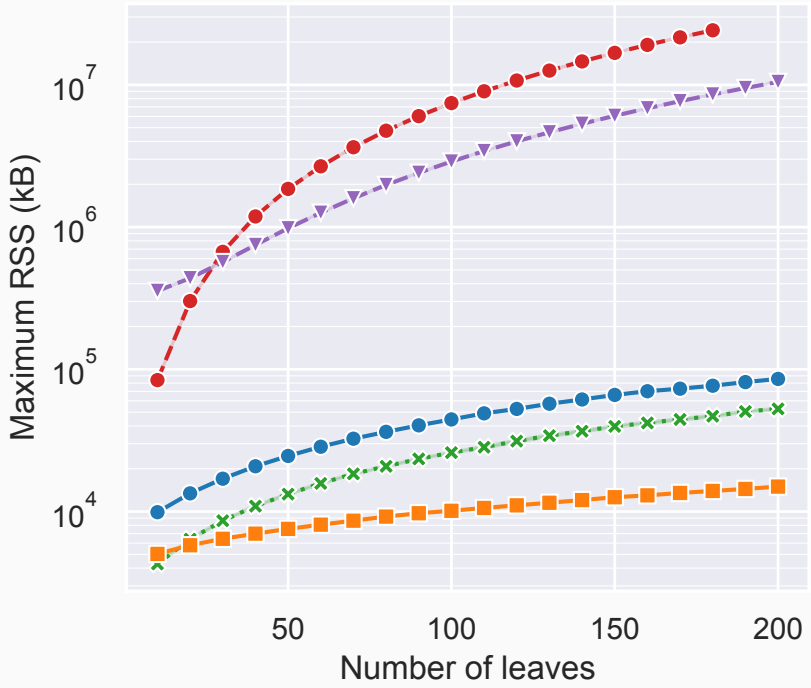
Höhna et al. 2016

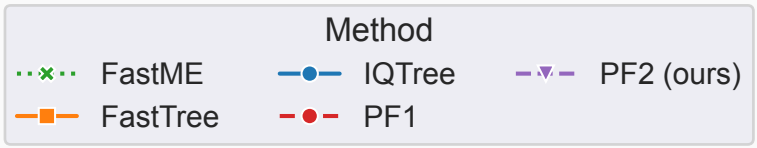
Results - What about scalability ?

Execution time



Memory usage[1]





[1] With 2× bigger sequence embeddings, and 4× bigger pair embeddings...

Where do we go from here ?

Conclusion - Take home messages

1. PF2 enables **end-to-end** phylogenetic **posterior estimation**
2. Under **tractable likelihood** it **outperforms** all other methods
3. **Performs well** Under models with **no likelihood**
4. PF2 can also produce well **calibrated posterior samples**
5. PF2 is **amortized** and allows for **fast inference**

Preprint →

Conclusion - Improving Phlyoformer 2

- **Improve Scalability** of PF2 either through:
  - More **efficient encoders**$^{(W.I.P)}$
  - **Heuristics** to build **larger trees**: e.g. supertrees
- Detect **Out-of-distribution** data and assess **prediction uncertainty**
- Explore more **flexible** tree-distribution **representations**
- Extend PF2 to additional **complex models**: e.g. structural models with **epistasis**

Wohlwend et al. 2025; Wang et al. 2025; Warnow 2018; Jiang et al. 2024
Gal and Ghahramani 2016; Lakshminarayanan et al. 2017; Latrille et al. 2021

Thank you all!

P. Barrat-Charlaix

B. Boussau

Q. Chaung

V. Garot

L. Jacob

N. Lartillot

A. Leroy

L. Nesterenko

N. Sauvage

P. Veber

CQSB

SORBONNE UNIVERSITÉ

LBBE

cnrs

anr

References

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

Gal, Y. and Z. Ghahramani (2016). ***Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning.*** In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by M. F. Balcan and K. Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, pp. 1050–1059.

Höhna, S. et al. (2016). ***RevBayes: Bayesian Phylogenetic Inference Using Graphical Models and an Interactive Model-Specification Language.*** In: *Systematic Biology* 65.4, pp. 726–736.

Jiang, Y. et al. (2024). ***Scaling DEPP phylogenetic placement to ultra-large reference trees: a tree-aware ensemble approach.*** en. In: *Bioinformatics* 40.6.

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

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

Lakshminarayanan, B. et al. (2017). ***Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles.*** In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc.

Latrille, T. et al. (2021). ***Inferring Long-Term Effective Population Size with Mutation–Selection Models.*** In: *Molecular Biology and Evolution* 38.10, pp. 4573–4587.

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.

Nesterenko, L. et al. (2025). ***Phyloformer: Fast, Accurate, and Versatile Phylogenetic Reconstruction with Deep Neural Networks.*** In: *Molecular Biology and Evolution* 42.4, msaf051.

Pagnani, A. and P. Barrat-Charlaix (2025). ***Generative continuous time model reveals epistatic signatures in protein evolution.***

Radev, S. T. et al. (2020). ***BayesFlow: Learning complex stochastic models with invertible neural networks.*** In: *IEEE transactions on neural networks and learning systems* 33.4, pp. 1452–1466.

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.

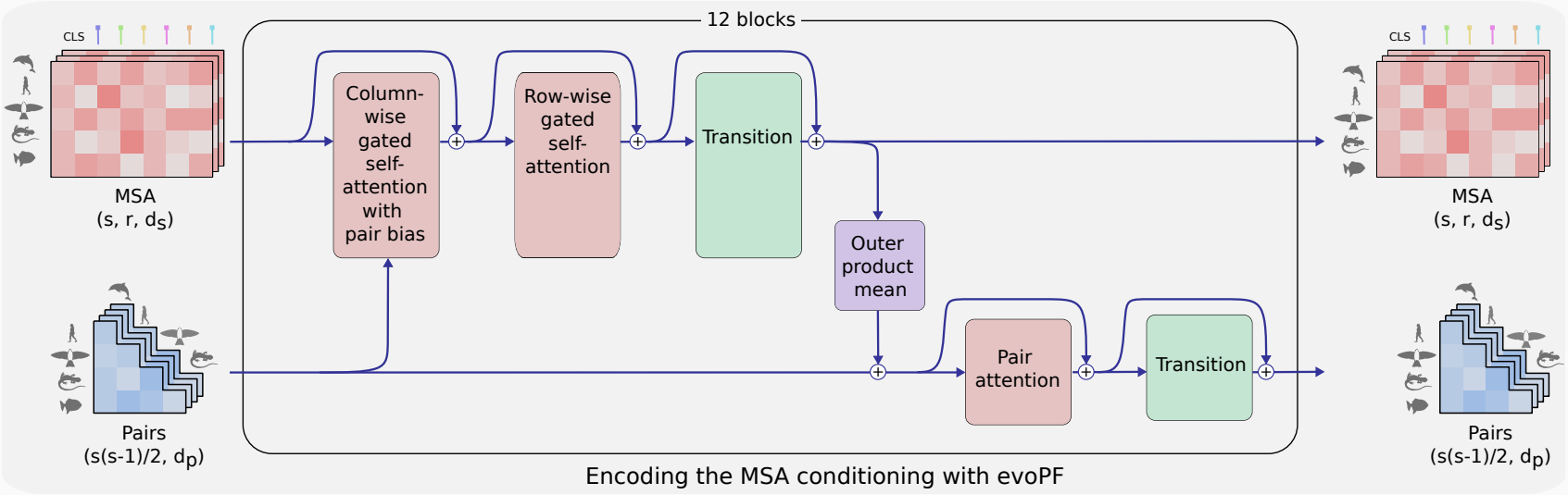
Wang, Y. et al. (2025). ***SimpleFold: Folding Proteins is Simpler than You Think*.**

Warnow, T. (2018). ***Supertree Construction: Opportunities and Challenges*.**

Wohlwend, J. et al. (2025). ***MiniFold: Simple, Fast, and Accurate Protein Structure Prediction*.** In: *Transactions on Machine Learning Research*. Featured Certification.

Supplementary Material

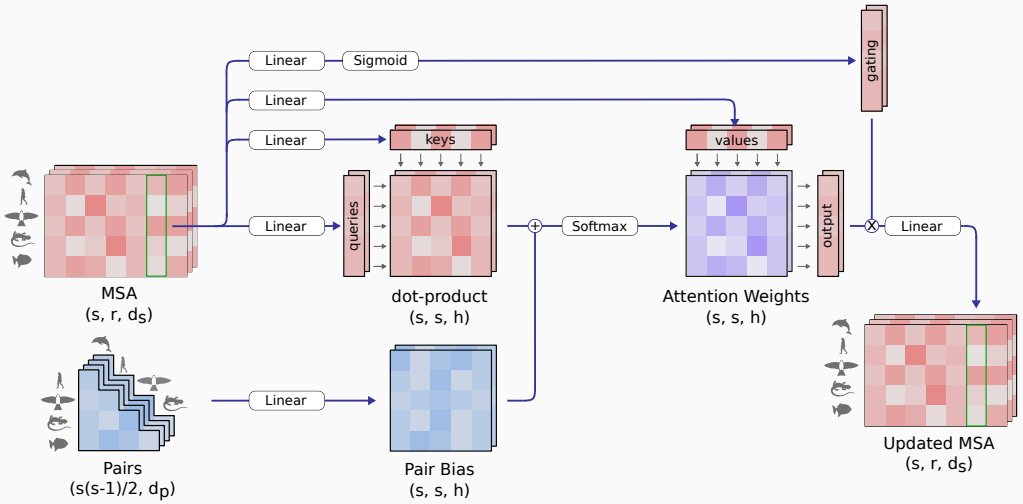
Sup. Methods - EvoPF architecture



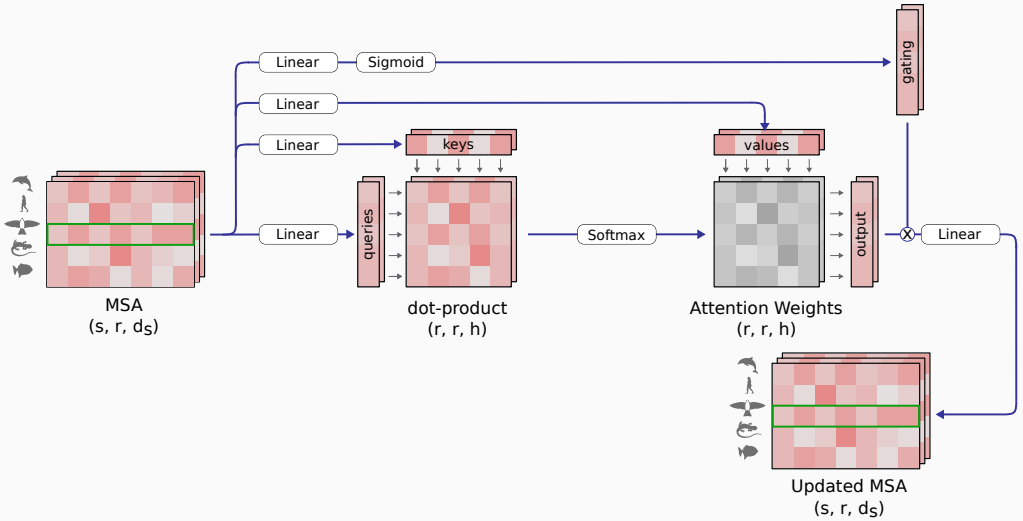
Encoding the MSA conditioning with evoPF

Figure inspired by Jumper et al. 2021

Sup. Methods - EvoPF, the MSA stack

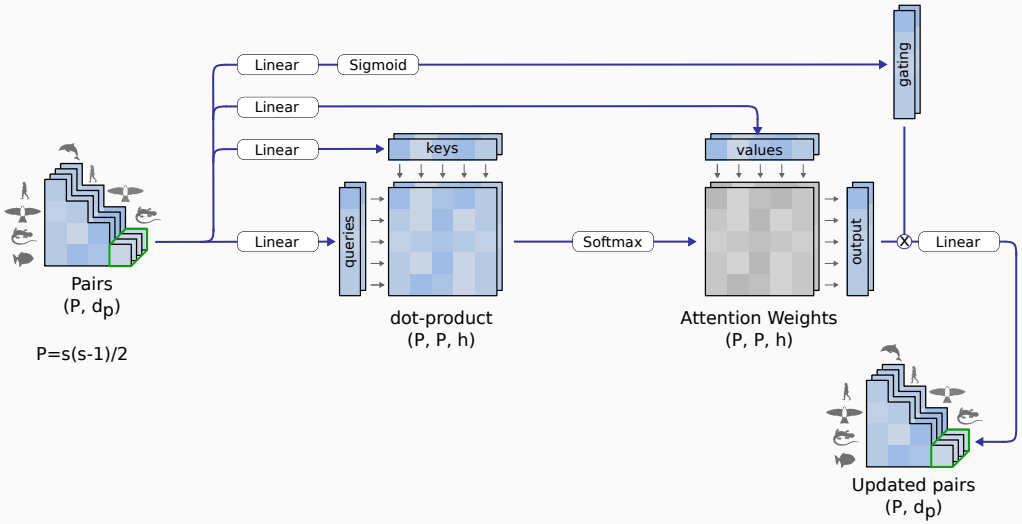


Column-wise attention with pair-bias

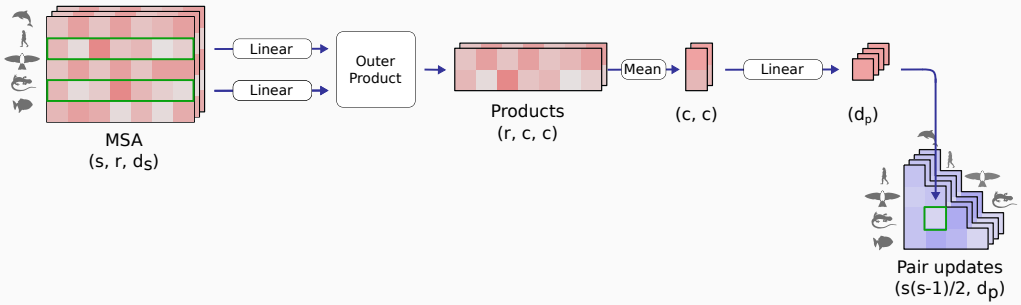


Row-wise attention

Sup. Methods - EvoPF, the pair stack

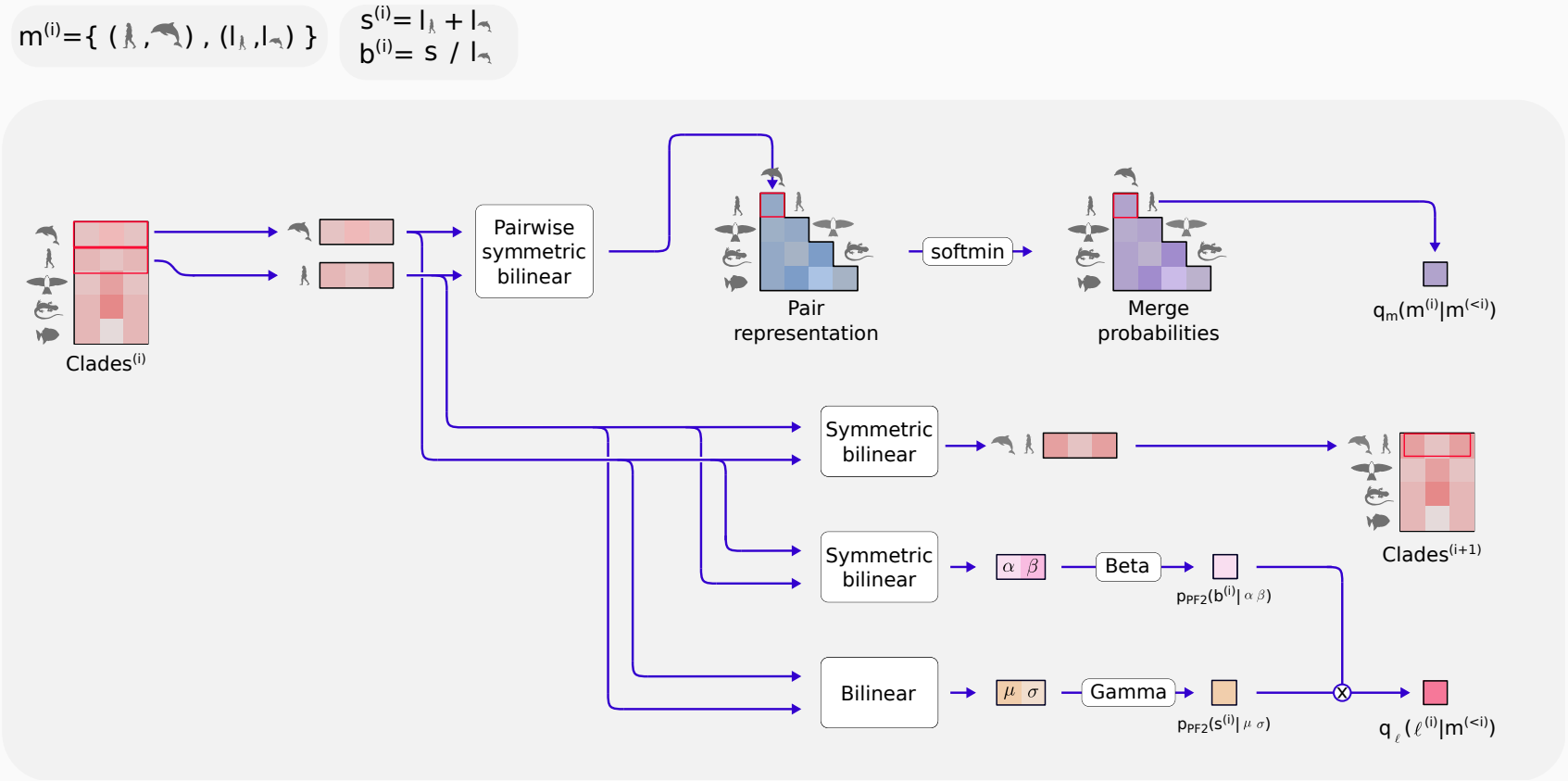


Pair attention



Outer product mean

Sup. Methods - The BayesNJ Module



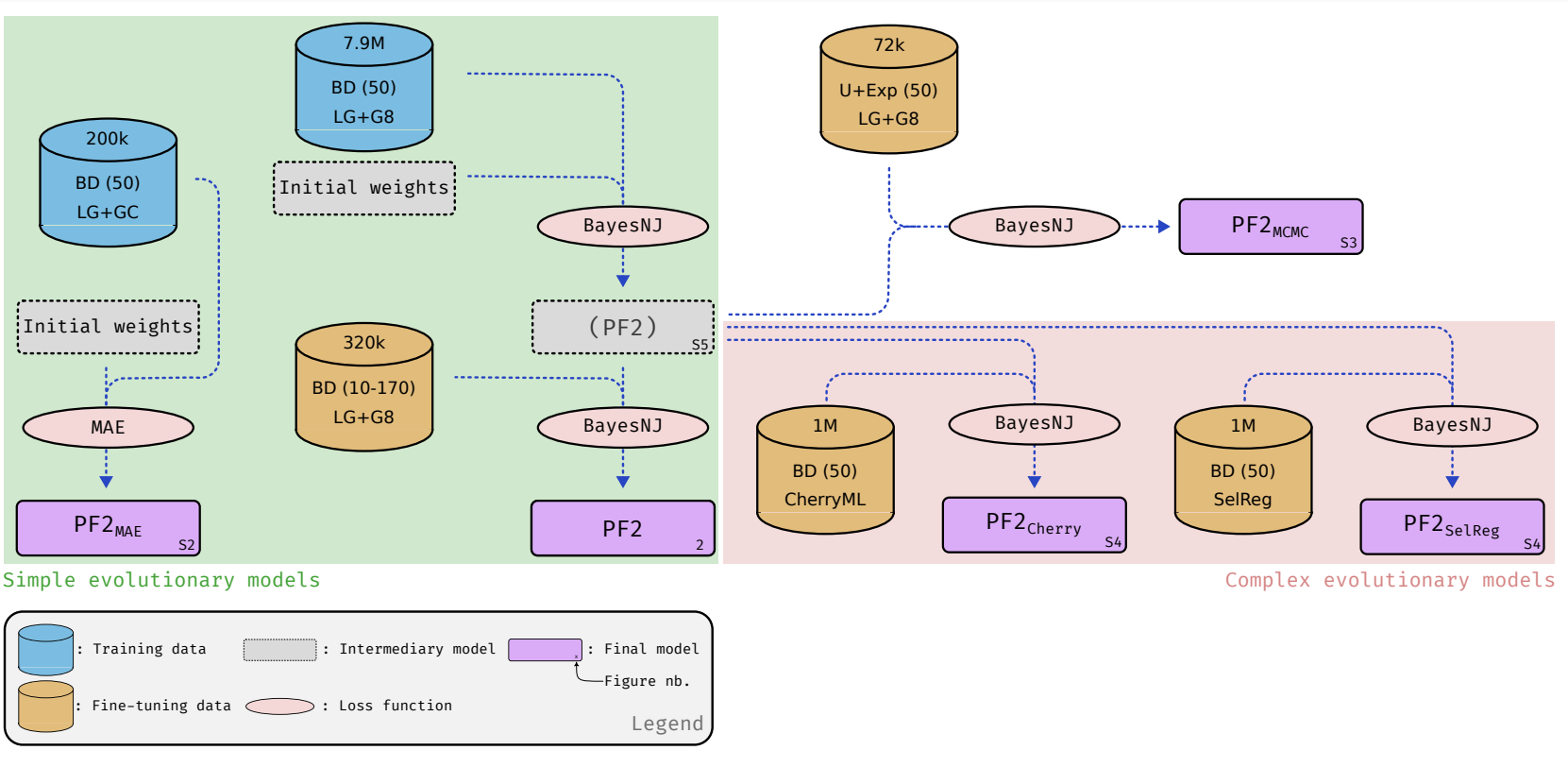
Sup. Methods - Ensuring the merge order is unique

Ensuring a **unique order** on merges ensures that we **define a distribution**. It also keeps **training** and **sampling** comparable [1]

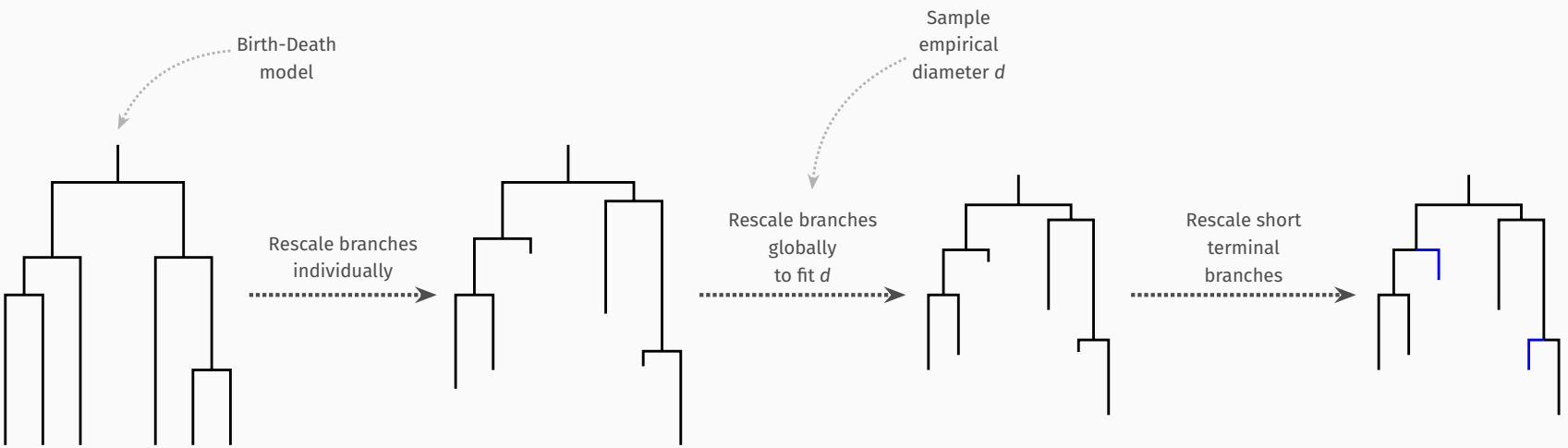
- On a given tree τ always **merge** the **shortest** available **cherry**
- When **sampling**, add **constraints**:
  1. Start with a $N \times N$ constraints matrix $M_{ij} = 0$
  2. At iteration k sample merge $m^{(k)} = (i, j)$ and cherry length $s^{(k)} = M_{ij} + X$
  3. **Update constraints** for cherries **available** when sampling $m^{(k)}$: $M'_{ij} = max(M_{ij}, s^{(k)}) \quad M'_{ui} = 0$
- During evaluation compute $p_{PF2}(s^{(k)} - M_{ij} | m^{(\leq k)})$

[1] Which is not the same if we use the NJ merge order

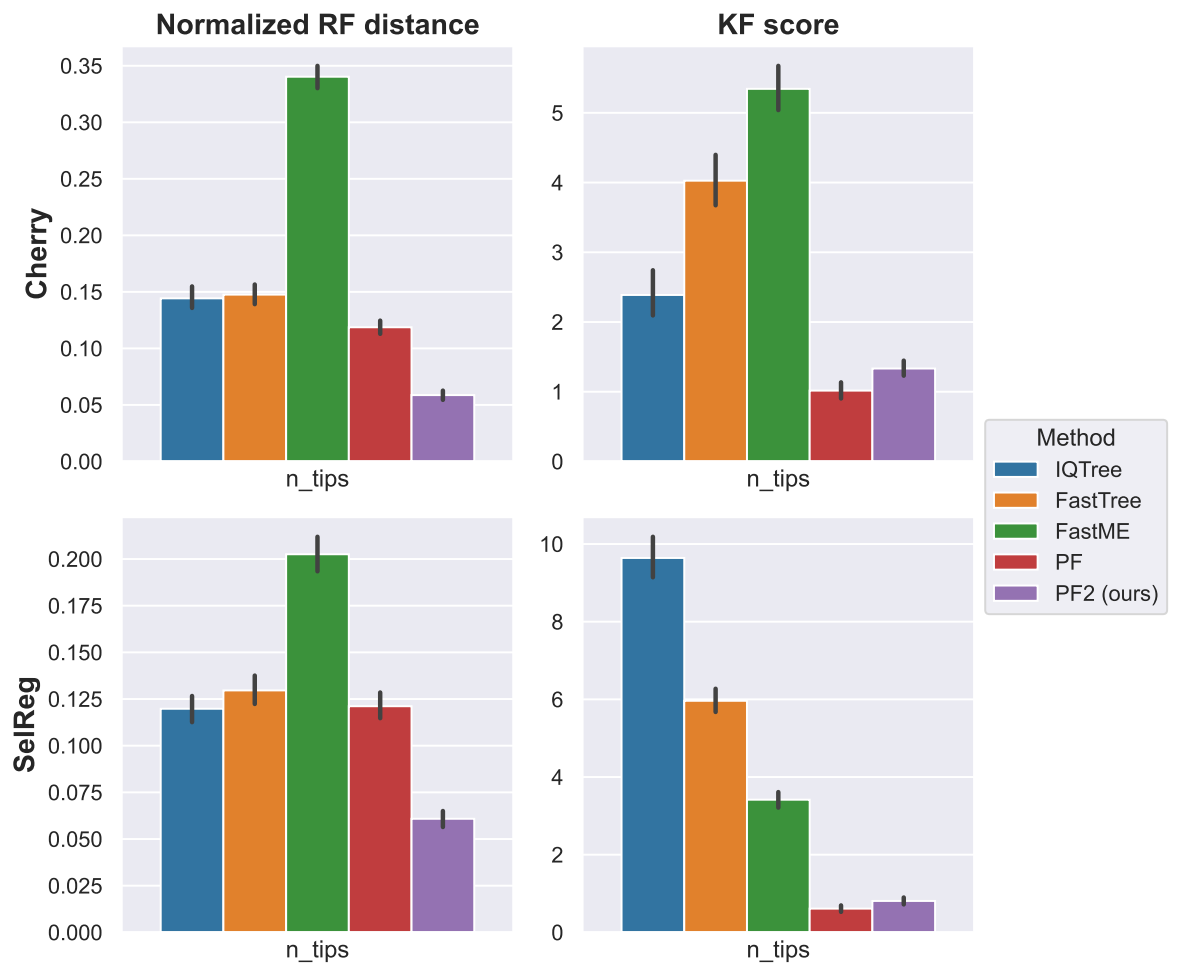
Sup. Methods - Training runs



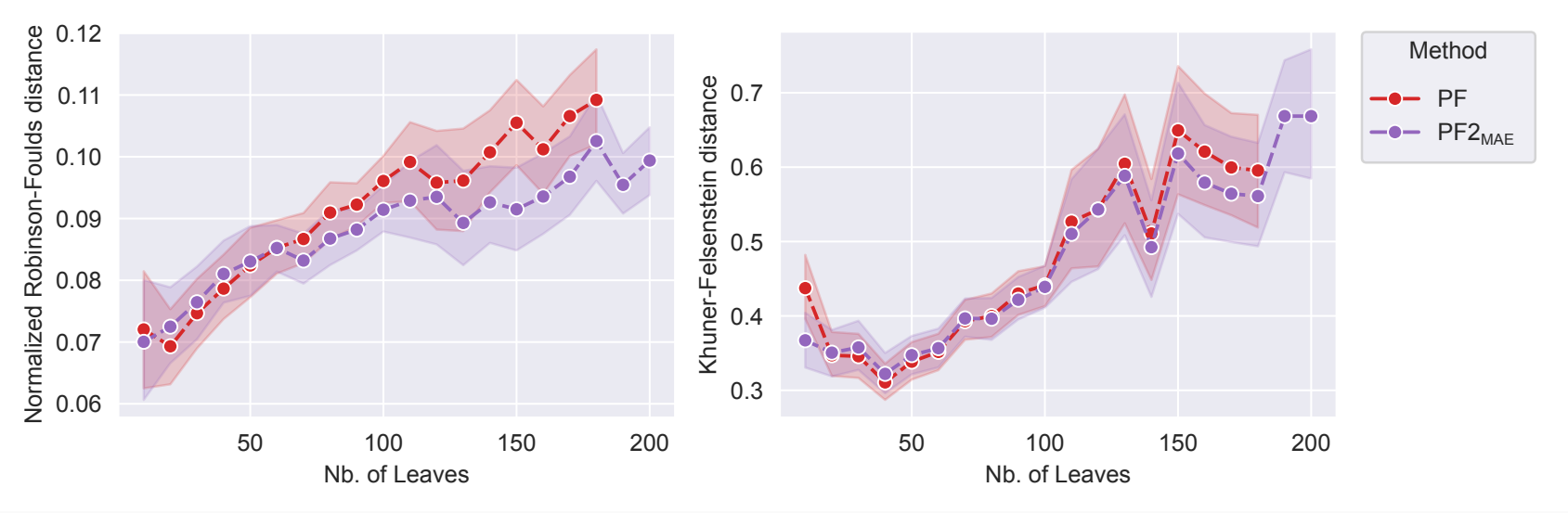
Sup. Methods - Tree simulation procedure



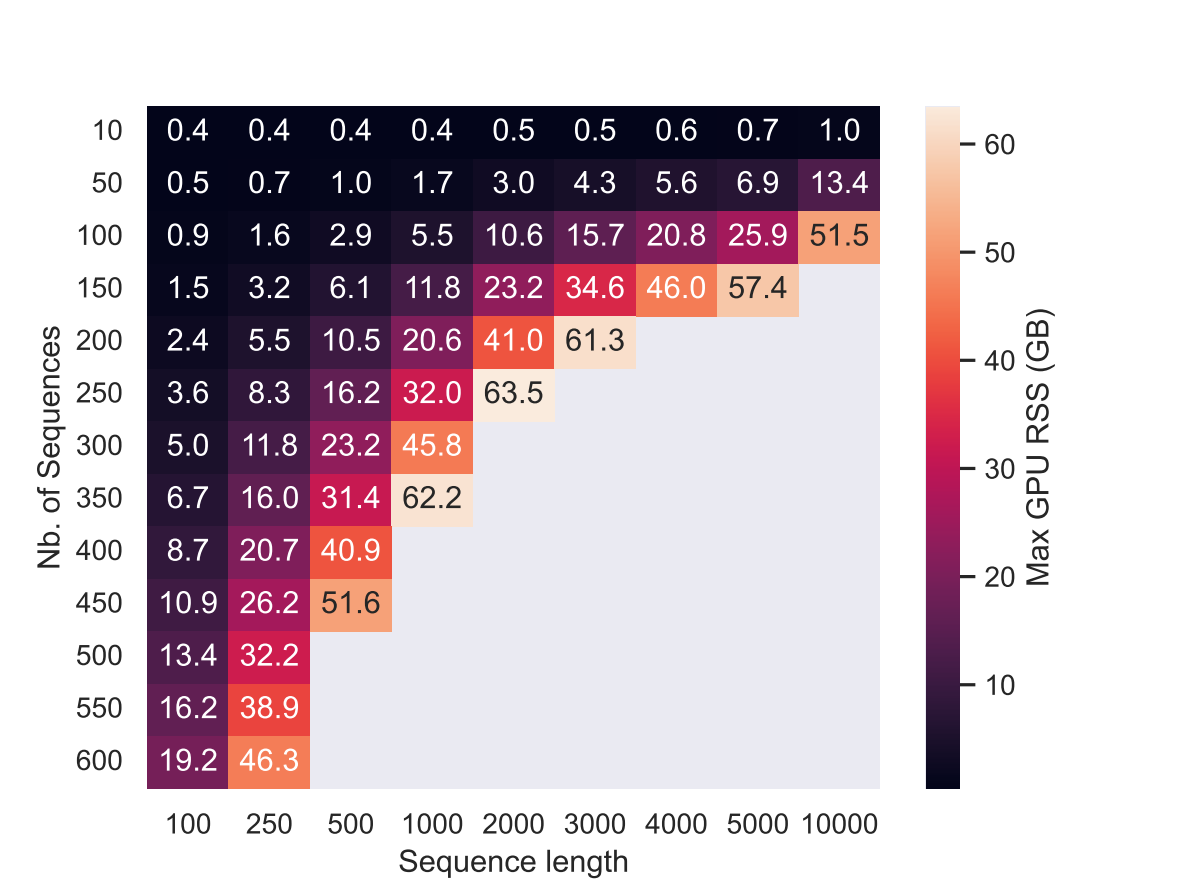
Sup. Results - PF2 performs well on complex models



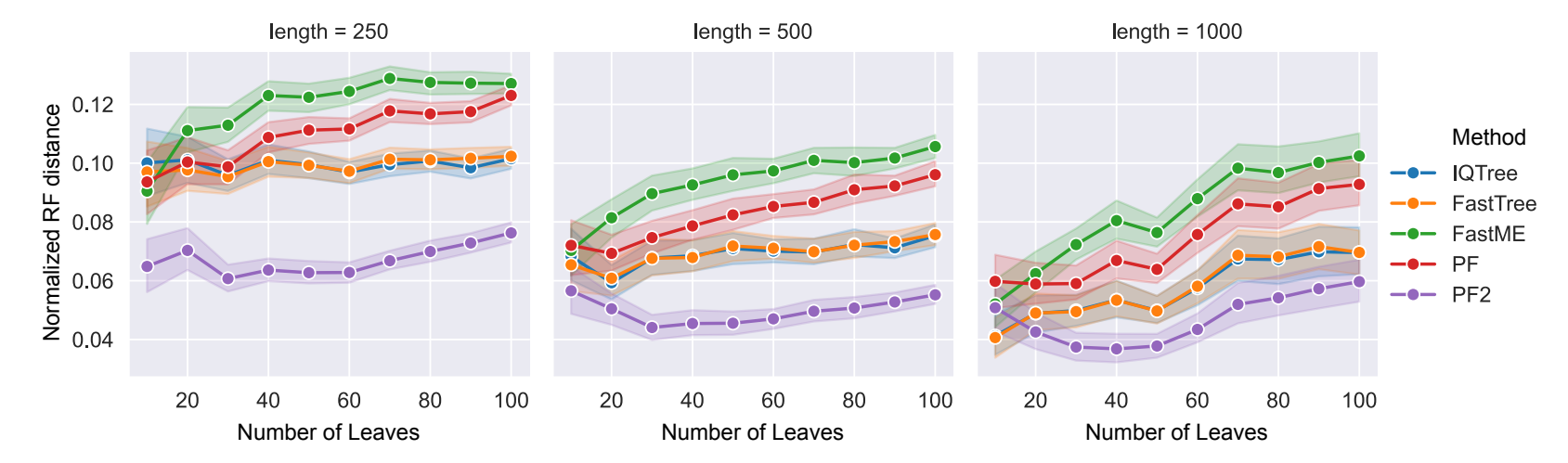
Sup. Results - Ablation study



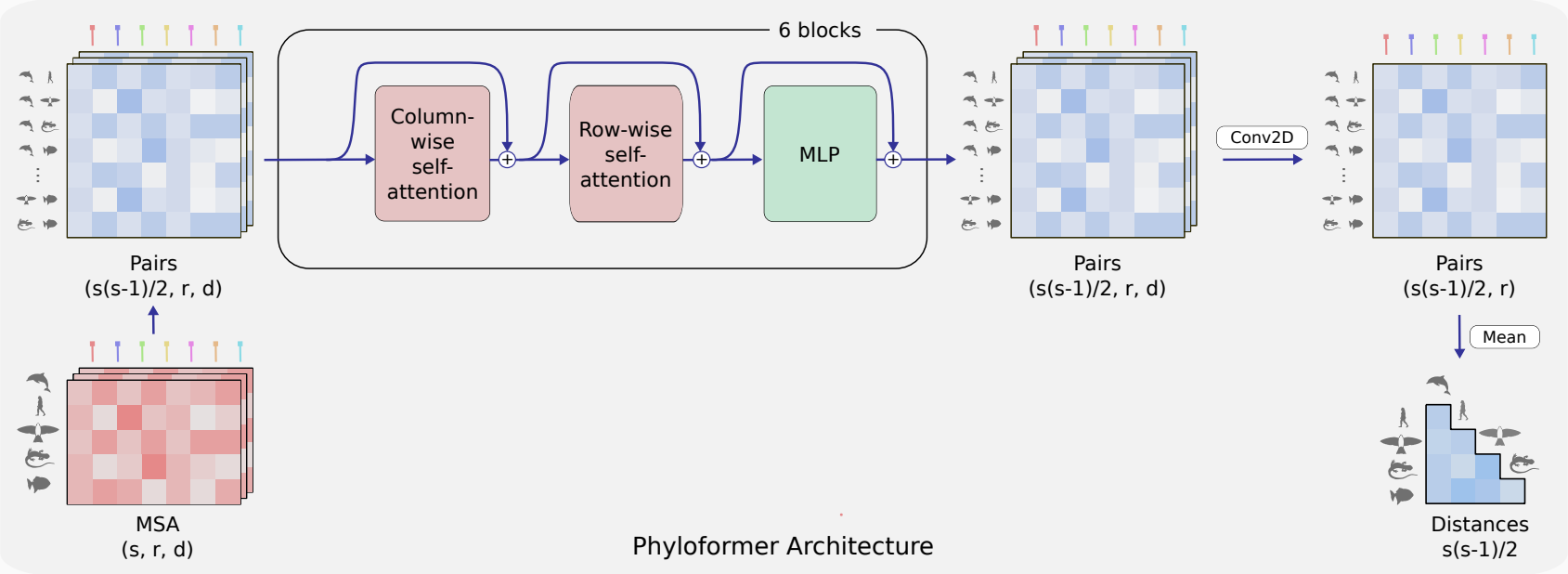
Sup. Results - Memory scaling



Sup. Results - Sensitivity to MSA length



Sup. Context - PF1 architecture



- Input an **MSA**, get a **Distance matrix**
- Feed Distance matrix to **FastME** to get **tree**

Nesterenko et al. 2025; Lefort et al. 2015

Sup. Context - PF1 performance