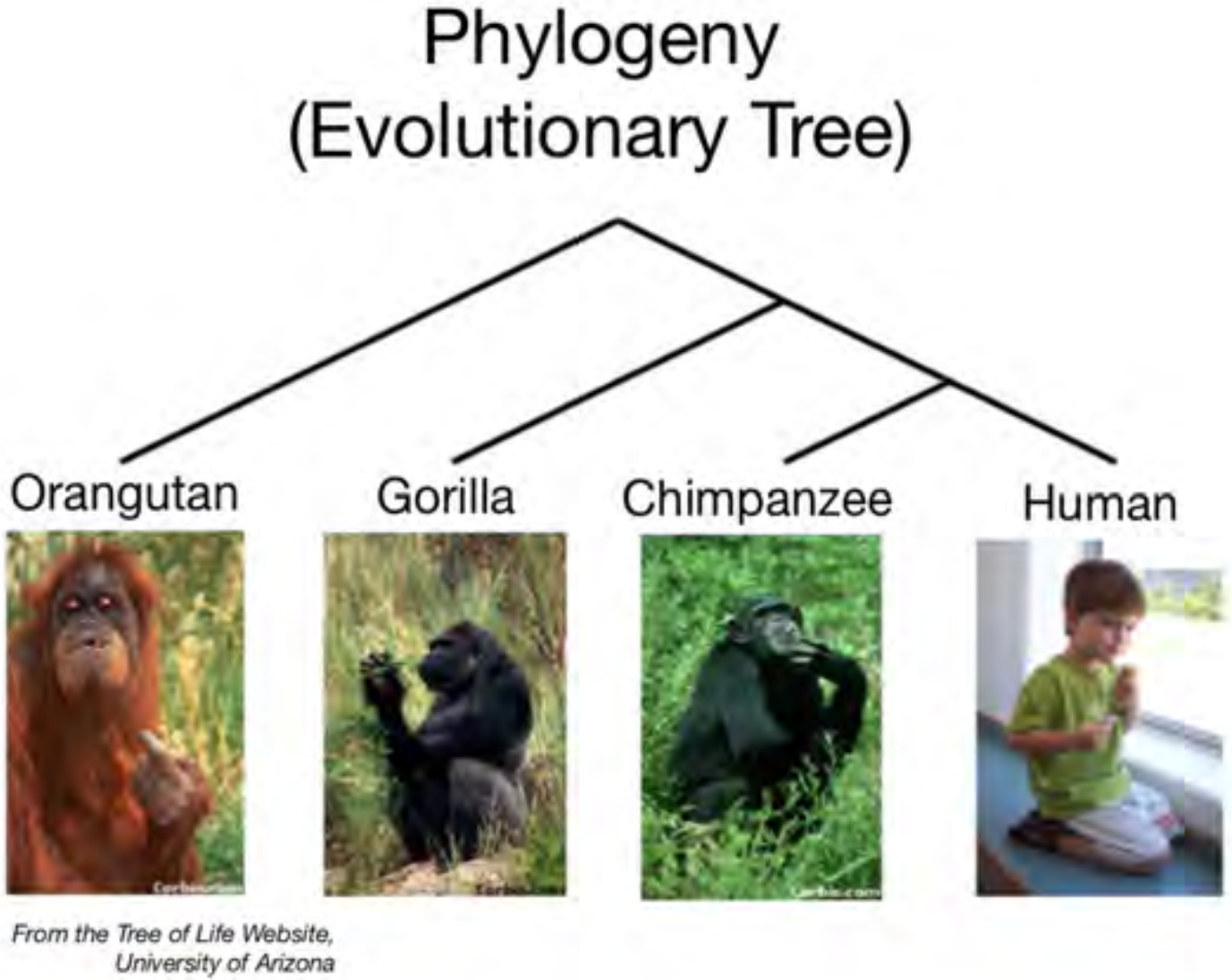
Designing parallel algorithms for constructing large phylogenetic trees on Blue Waters

Erin Molloy
University of Illinois at Urbana Champaign

General Allocation (PI: Tandy Warnow)
Exploratory Allocation (PI: Bill Gropp)

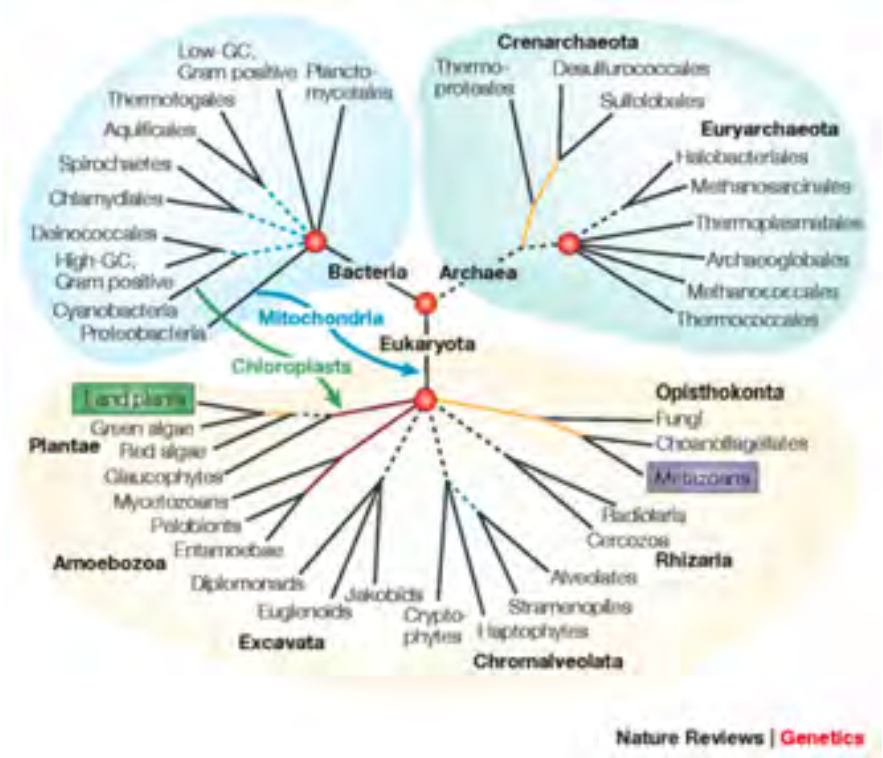
NCSA Blue Waters Symposium
June 5, 2018

Phylogeny


Phylogeny
(Evolutionary Tree)
Orangutan
Gorilla
Chimpanzee
Human
From the Tree of Life Website,
University of Arizona

Tree of Life and Downstream Applications

"Nothing in biology makes sense except in the light of evolution"
–Dobhzansky (1973)

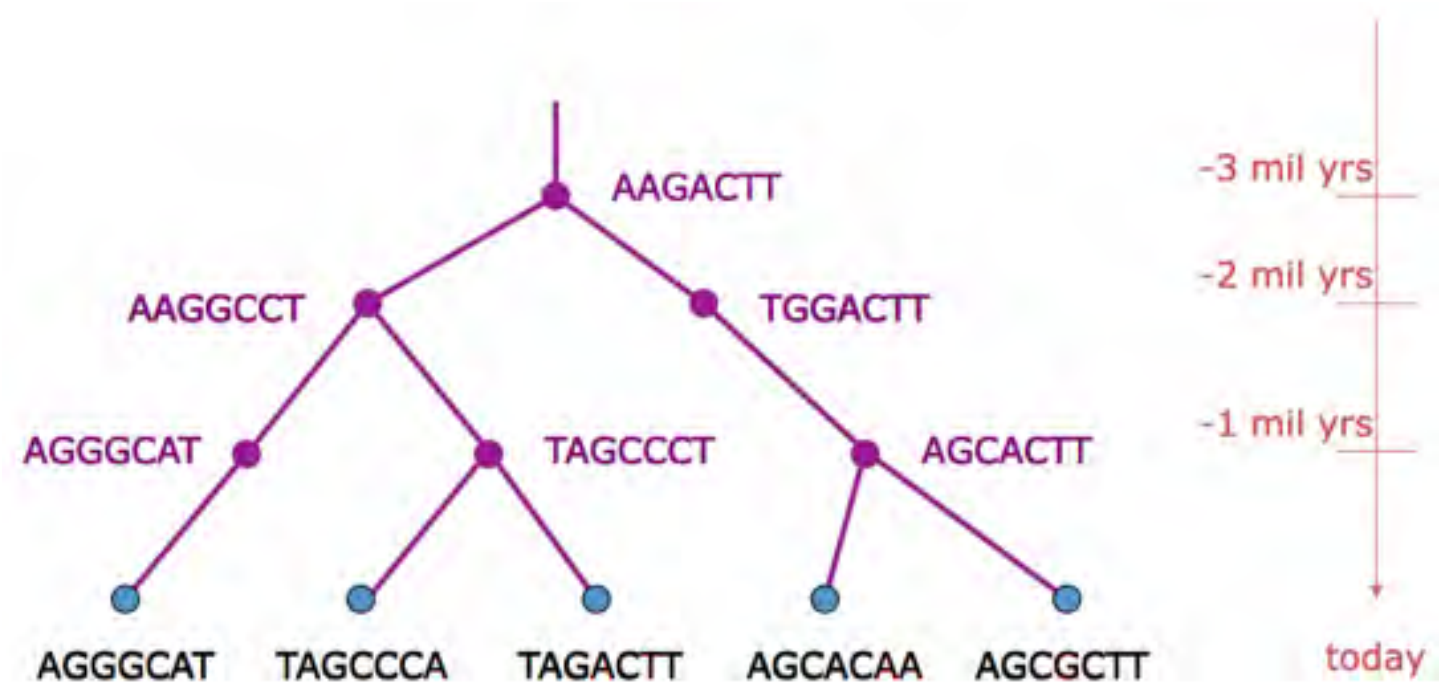


- Protein structure and function prediction
- Population genetics
- Human migrations
- Metagenomics
- Infectious Disease
- Biodiversity

Outline

- Phylogeny estimation pipeline
- Some standard approaches and their limitations
- Our new approach to ultra-large phylogeny estimation on Blue Waters
- Comparison of our approach to existing methods
- Future work

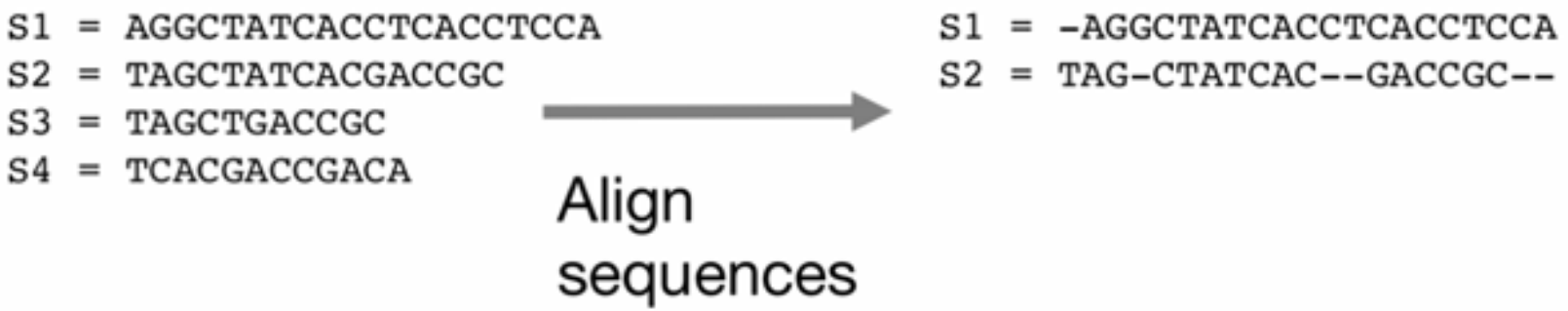
DNA Sequence Evolution



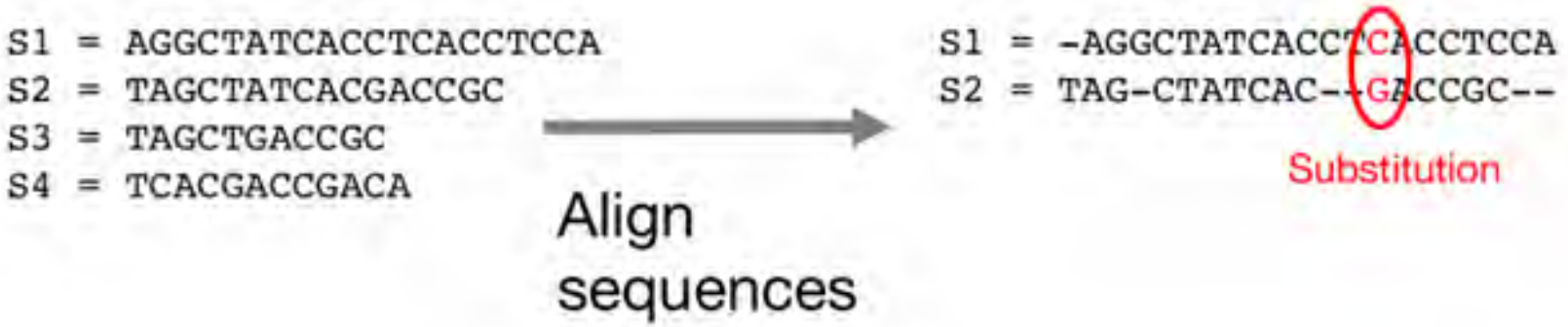
Phylogeny Estimation Pipelines

```
S1 = AGGCTATCACCTCACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```

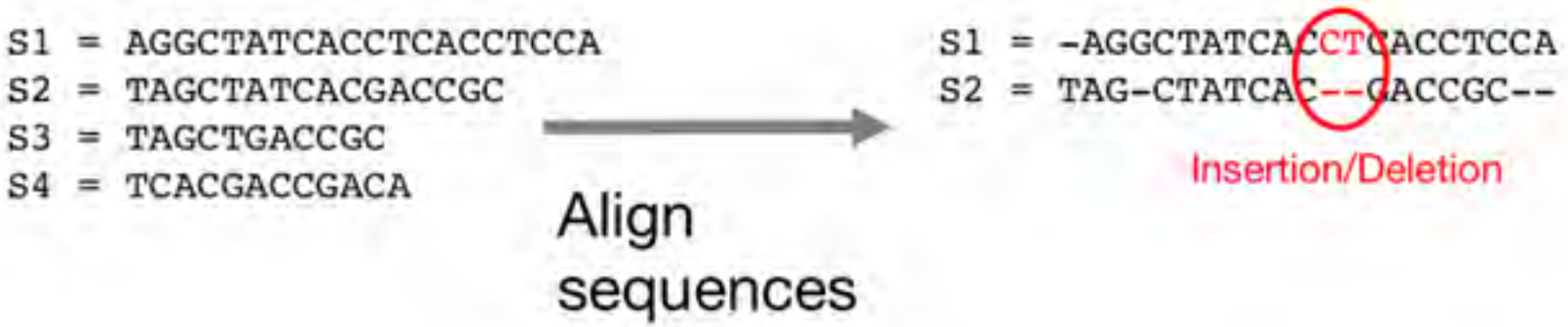
Phylogeny Estimation Pipelines



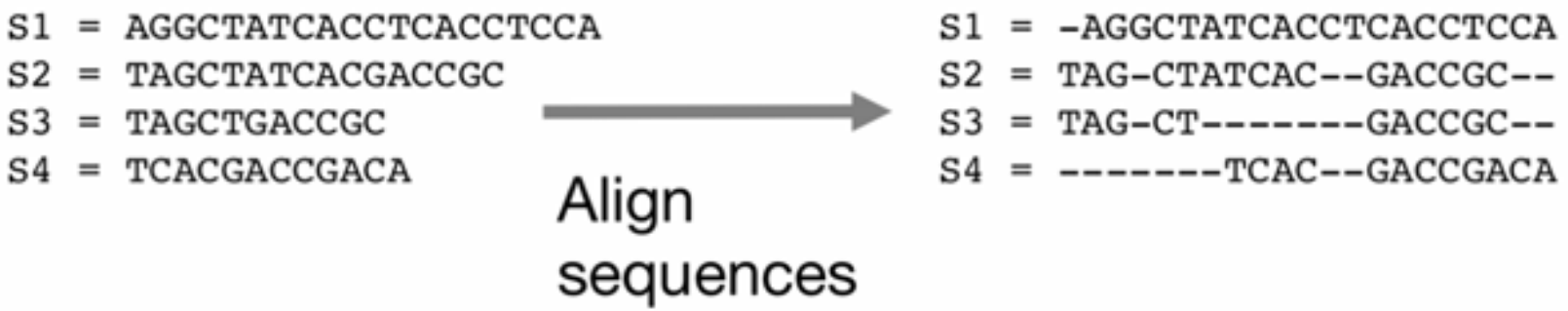
Phylogeny Estimation Pipelines



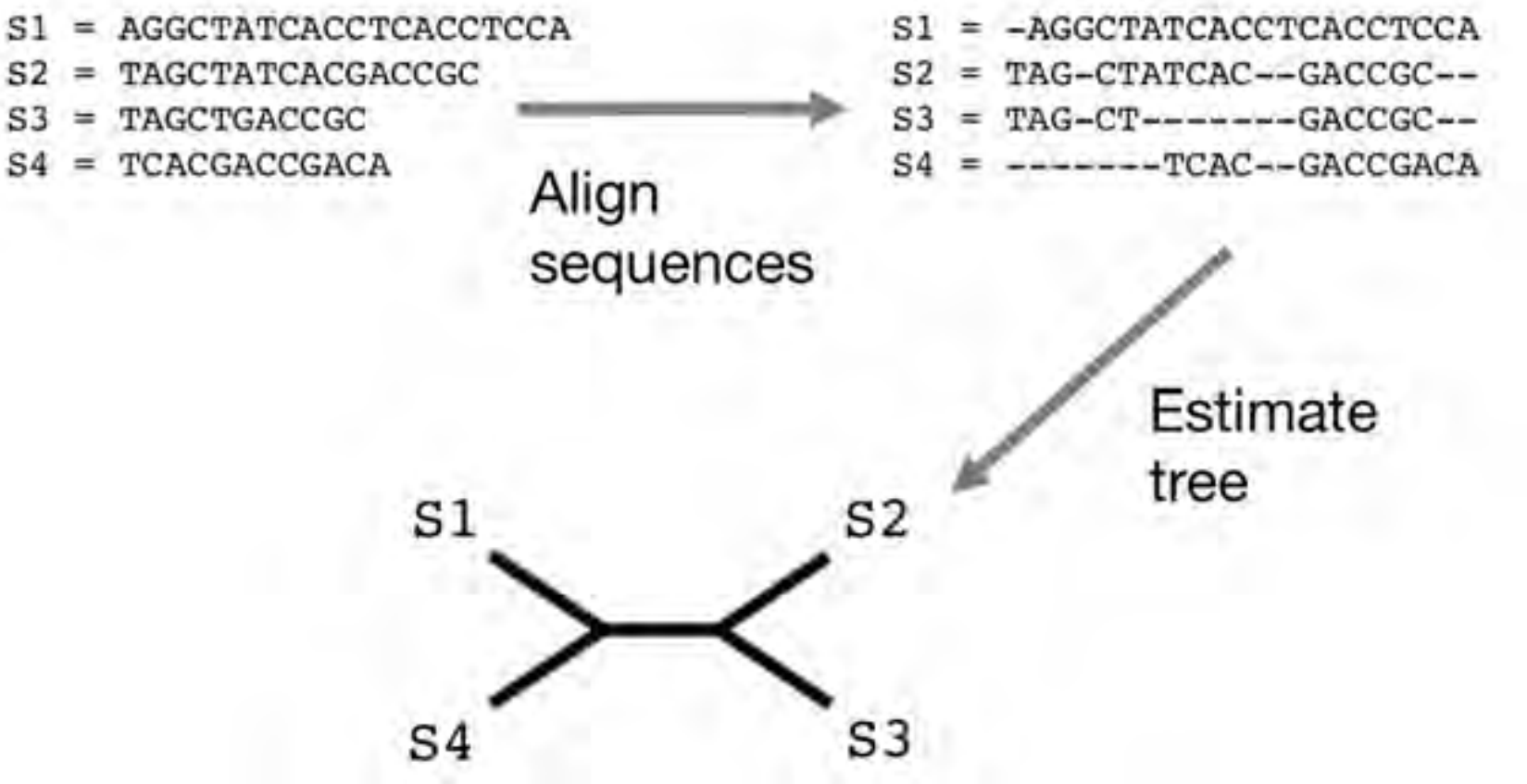
Phylogeny Estimation Pipelines



Phylogeny Estimation Pipelines



Phylogeny Estimation Pipelines



Distance Methods

- **Input:** A matrix D such that $D[i, j]$ indicates the distance between sequence i and sequence j
  - Use multiple sequence alignment to compute pairwise distances
  - Use **alignment-free** method to compute pairwise distances in an embarrassingly parallel fashion
- **Output:** A tree with branch lengths

Distance methods use **polynomial time**.

Maximum Likelihood (ML) Tree Estimation

- **Input:** A multiple sequence alignment
- **Output:** A model tree (topology and other numerical parameters) that maximizes likelihood, that is, the probability of observing the multiple sequence alignment given a model tree

The ML tree estimation problem is **NP-hard**.

[Felsenstein,1981; Roch, 2006]

Maximum Likelihood (ML) Tree Estimation

- **Input:** A multiple sequence alignment
- **Output:** A model tree (topology and other numerical parameters) that maximizes likelihood or probability of observing the multiple sequence alignment given a model tree

The ML tree estimation problem is **NP-hard**.

[Felsenstein,1981; Roch, 2006]

ML Heuristic are typically **more accurate than distance methods**, especially under some challenging model conditions.

ML Tree Estimation: N versus L

A multiple sequence alignment is an $N \times L$ matrix.

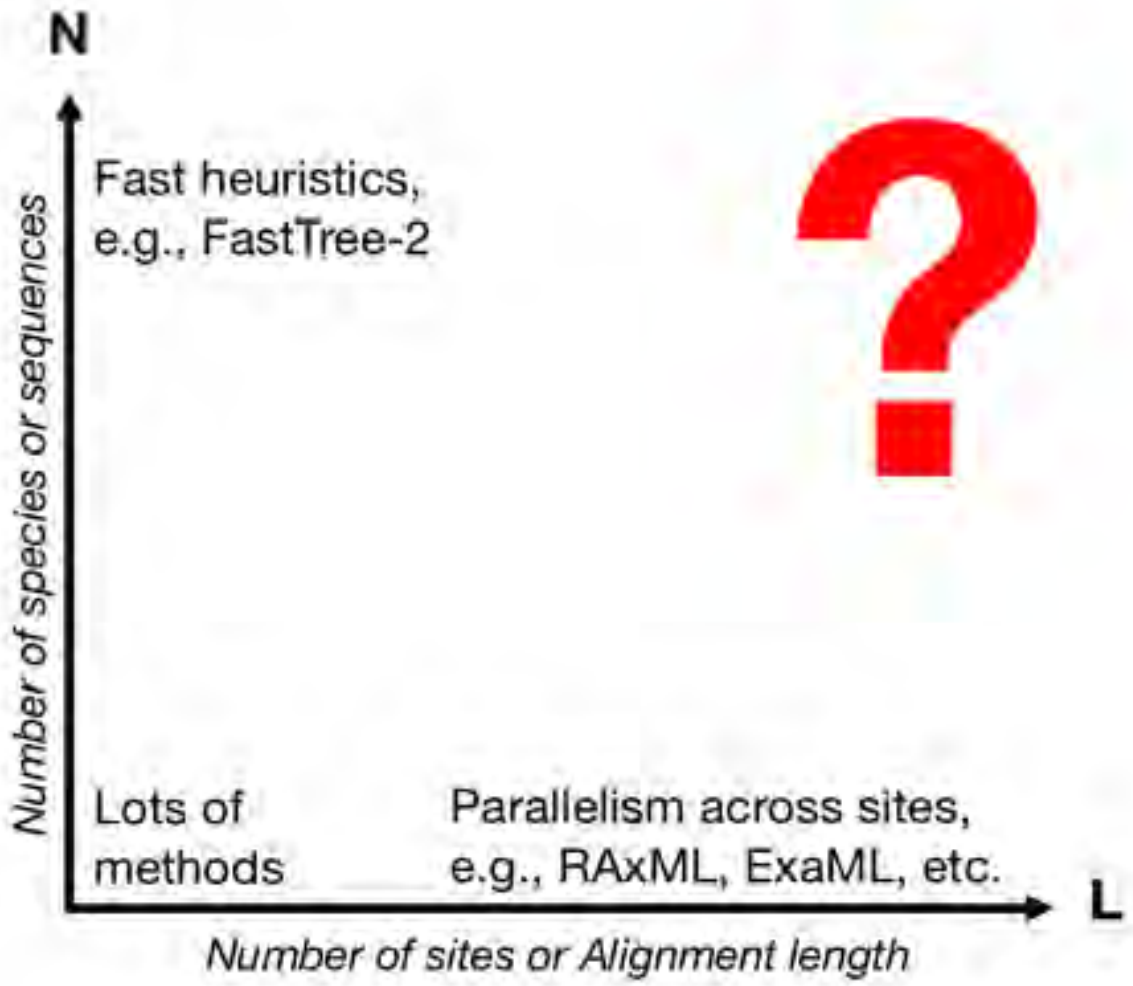
Number of sites or alignment length, L

- Thousands of sites for single gene analysis
- Millions of sites for multi-gene analysis
- Many analyses can be parallelized across sites
  - Because likelihood is computed on each site independently

Number of species or sequences, N

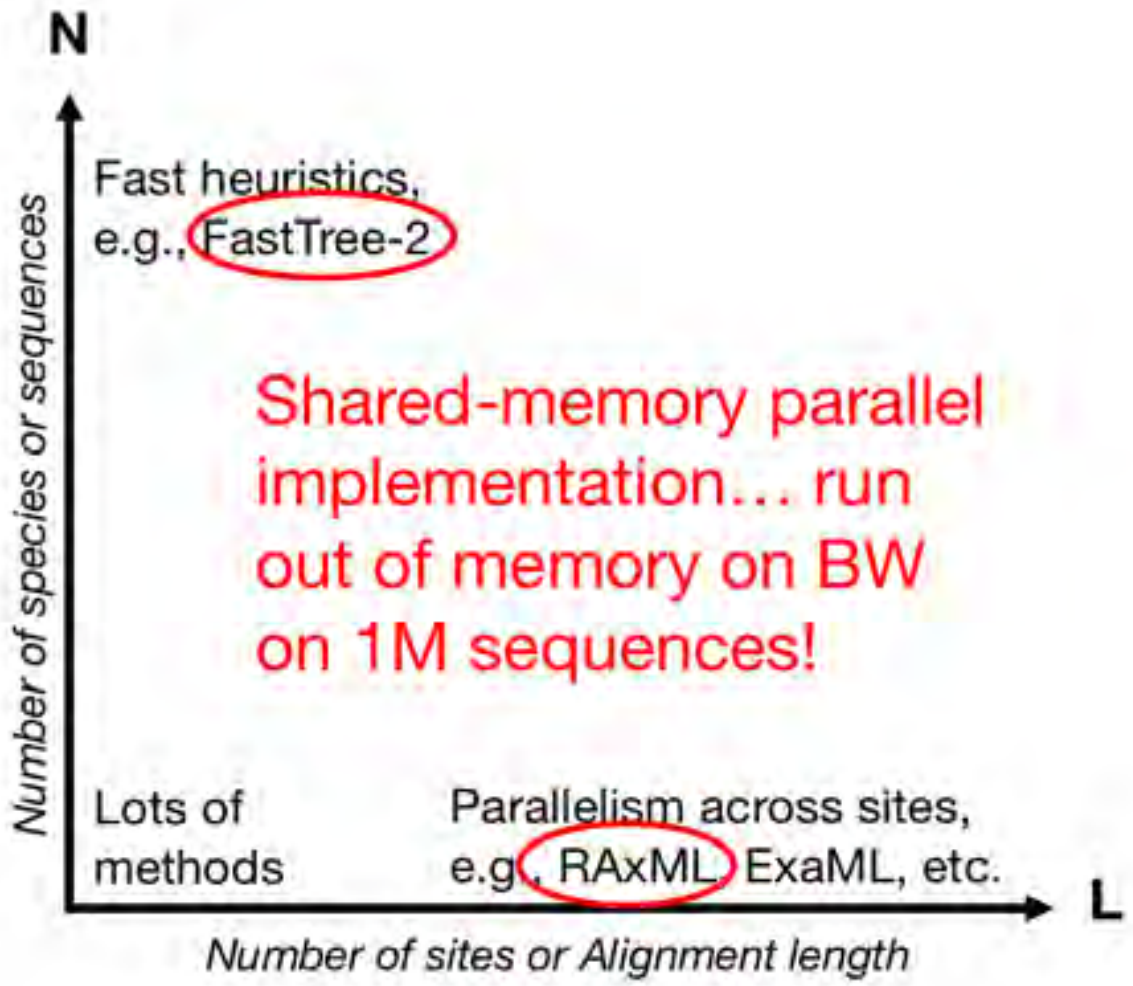
- Many datasets have thousands of species
- The Tree of Life will have millions
- Number of tree topologies on N leaves is $(2N - 5)!!$
- Parallelism is more complicated

ML Tree Estimation: N versus L



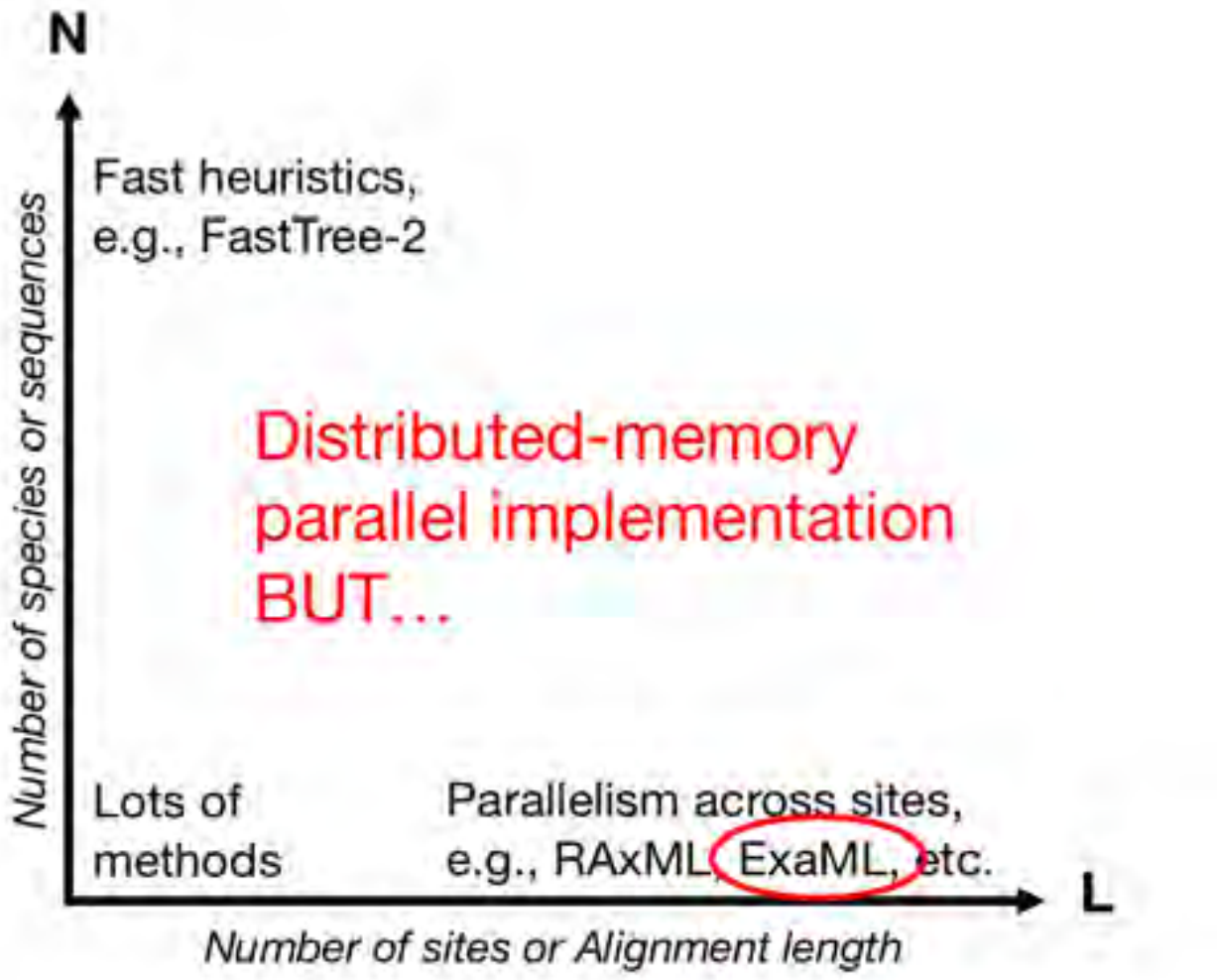
[Stamatakis, 2006; Price et al., 2010; Kozlov et al., 2015]

ML Tree Estimation: N versus L



[Stamatakis, 2006; Price et al., 2010; Kozlov et al., 2015]

ML Tree Estimation: N versus L



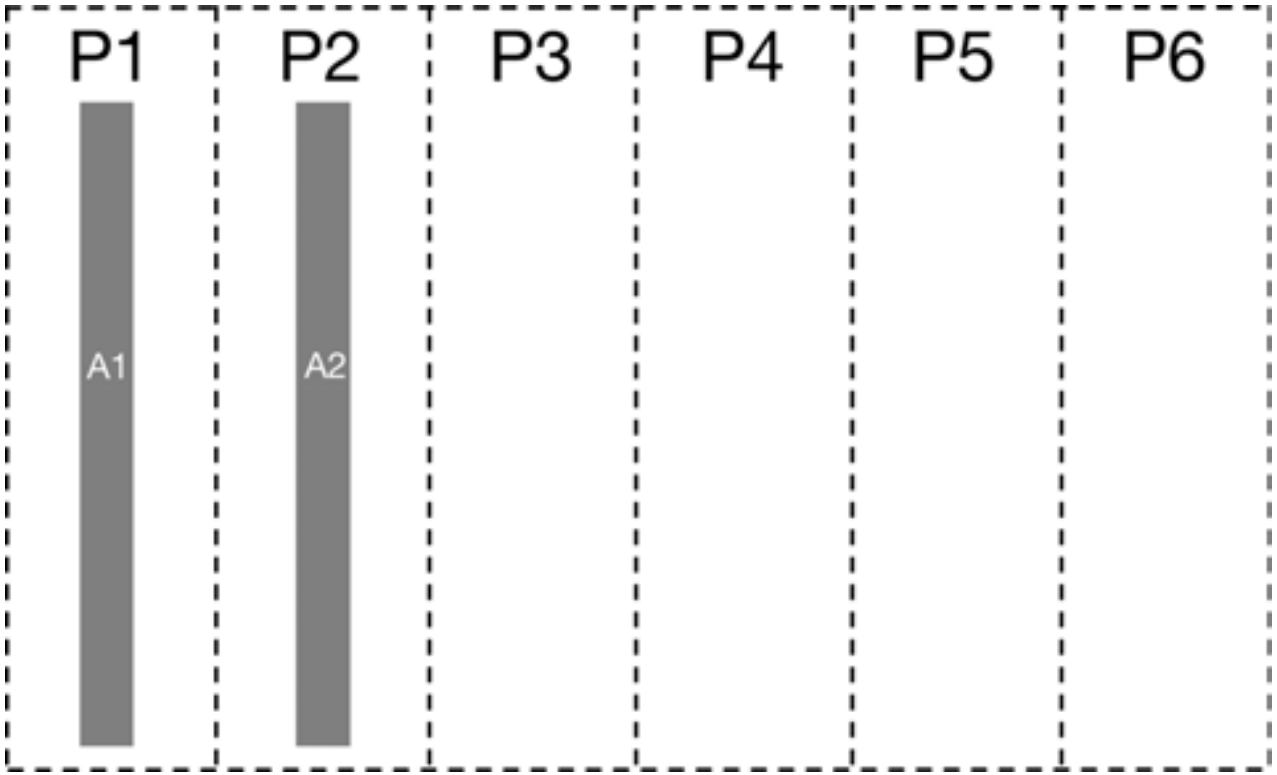
[Stamatakis, 2006; Price et al., 2010; Kozlov et al., 2015]

ML Tree Estimation: N versus L

ExaML [Kozlov et al., 2015] can "be used for analyzing datasets with 10-20 genes and up to 55,000 taxa, but scalability will be limited to at most 100 cores".

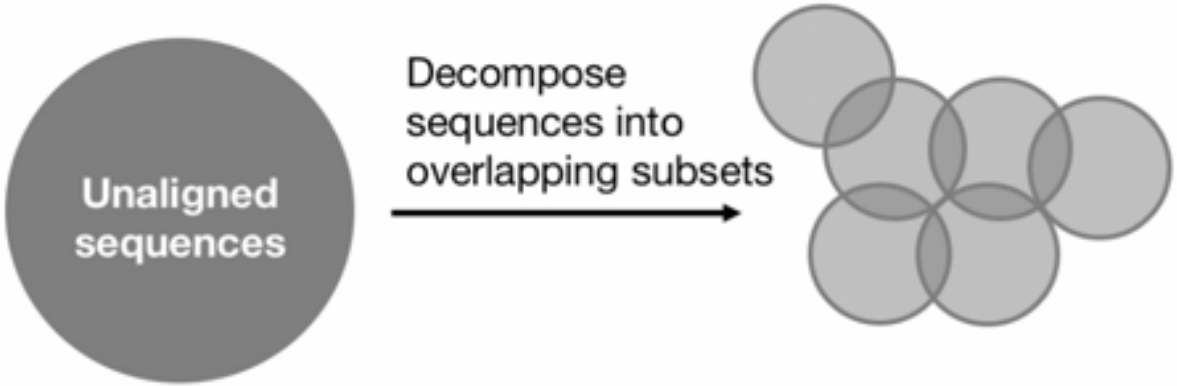
ML Tree Estimation: N versus L

ML Tree Estimation: N versus L

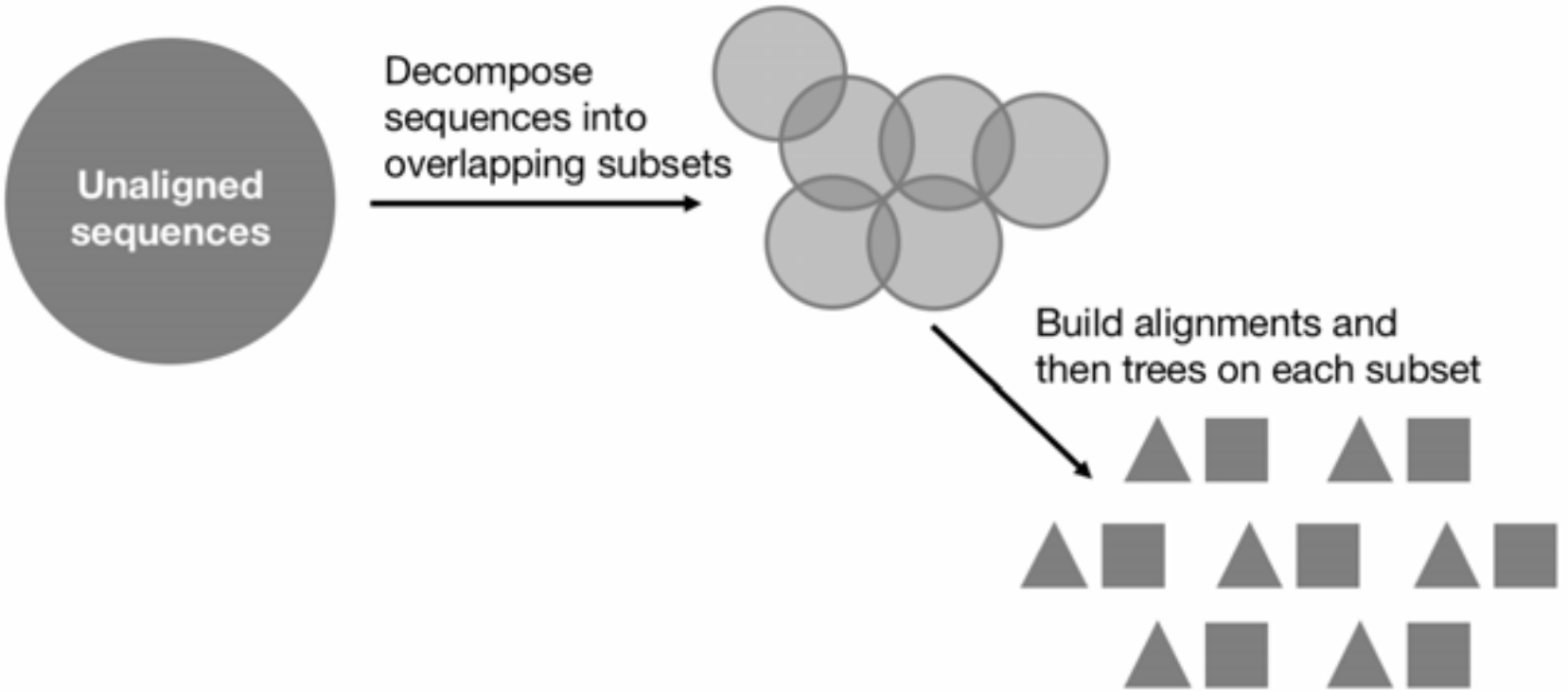


Divide-and-Conquer with DACTAL [Nelesen et al., 2012]

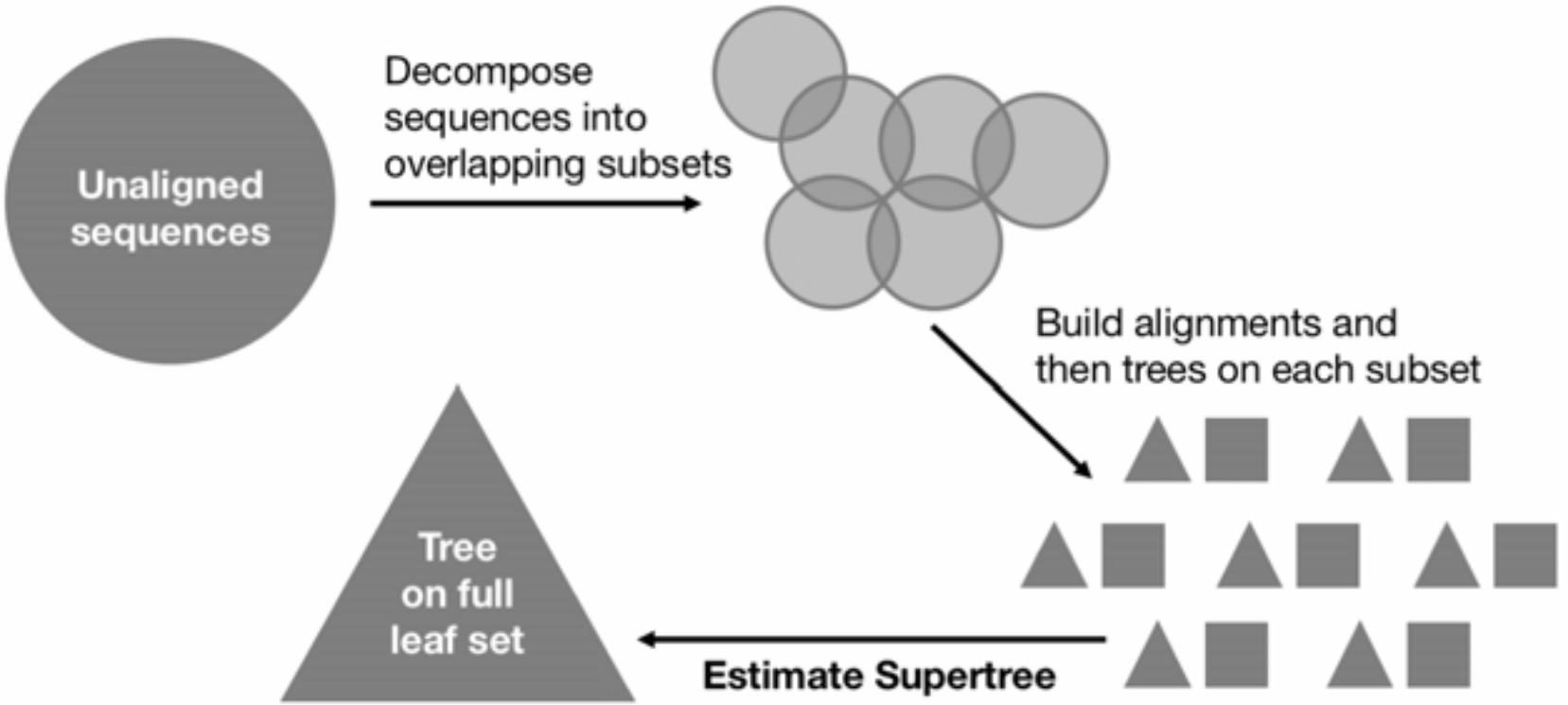
Divide-and-Conquer with DACTAL [Nelesen et al., 2012]



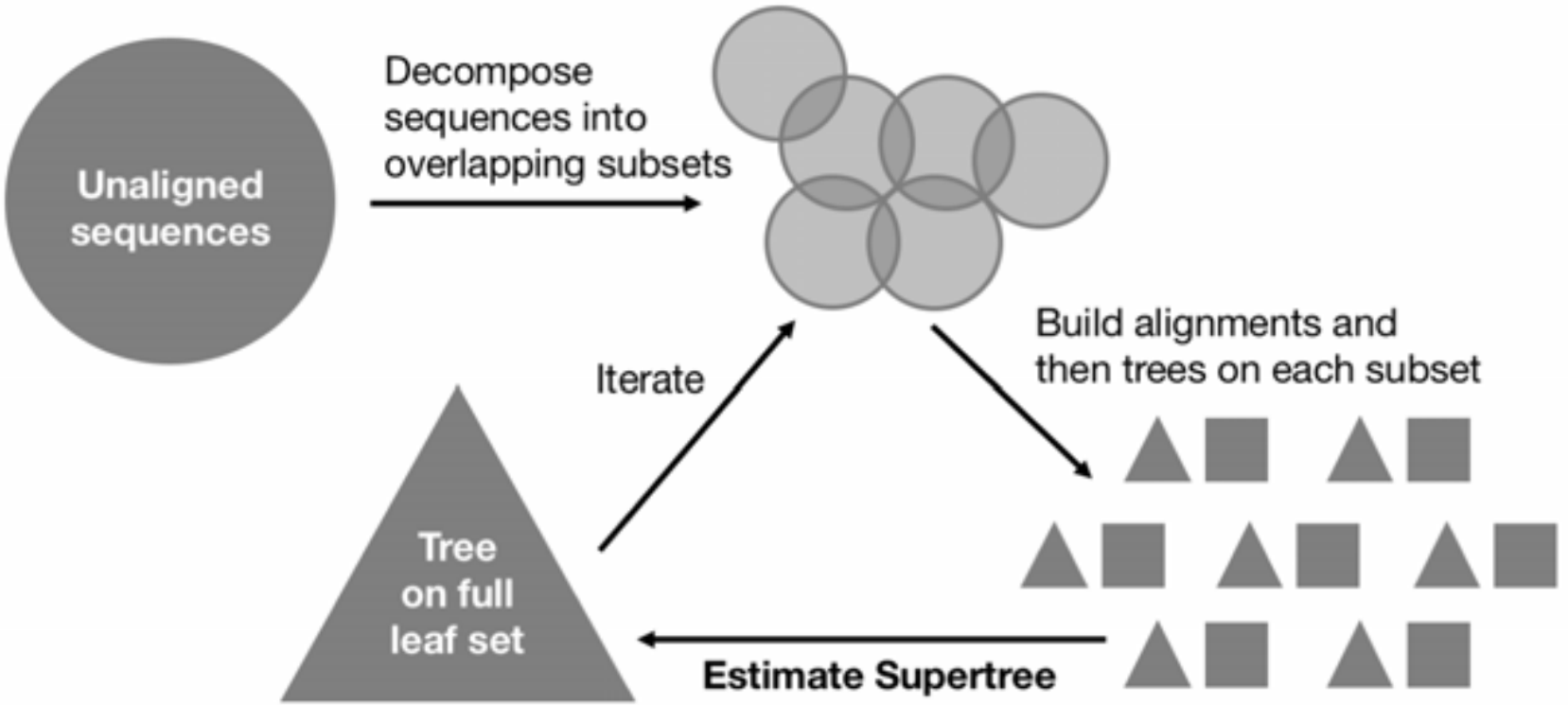
Divide-and-Conquer with DACTAL [Nelesen et al., 2012]



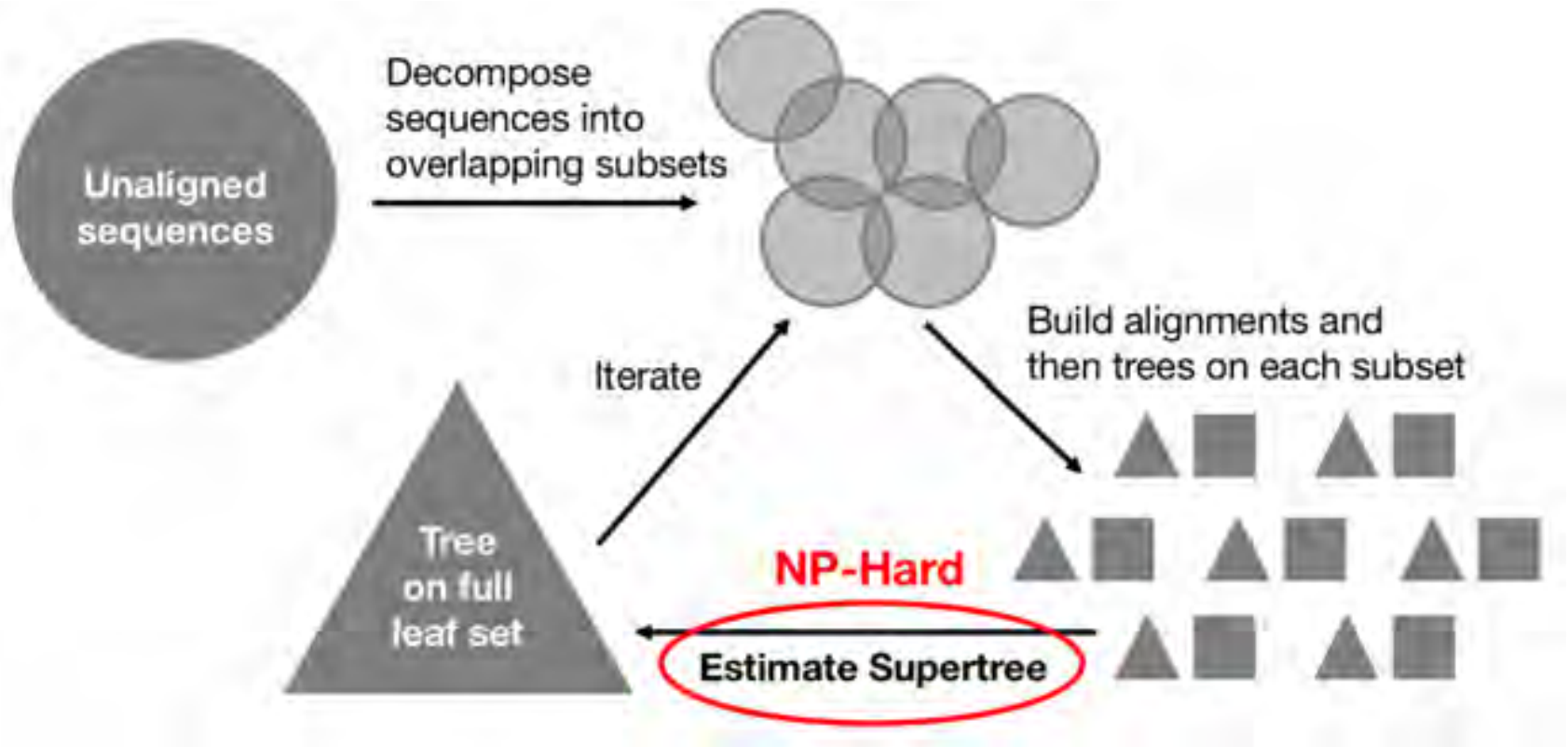
Divide-and-Conquer with DACTAL [Nelesen et al., 2012]



Divide-and-Conquer with DACTAL [Nelesen et al., 2012]



Divide-and-Conquer with DACTAL [Nelesen et al., 2012]



[Steel, 1992; Jiang et al., 2001; Bansal et al., 2010]

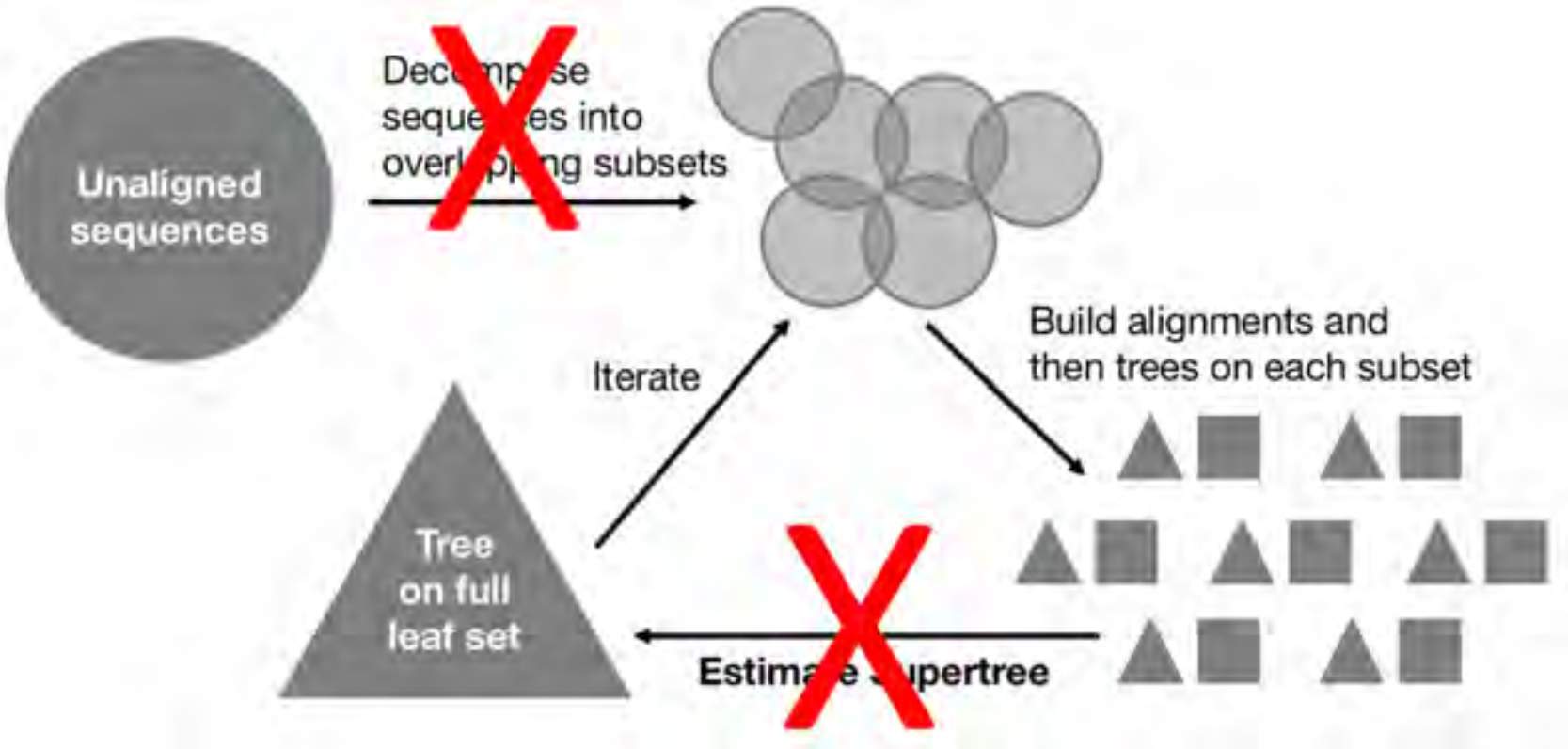
Our Design Goals

If we want to build the Tree of Life (millions of sequences!) using Blue Waters, then we need to design an algorithm that can utilize a very large numbers of (not high memory) processors.

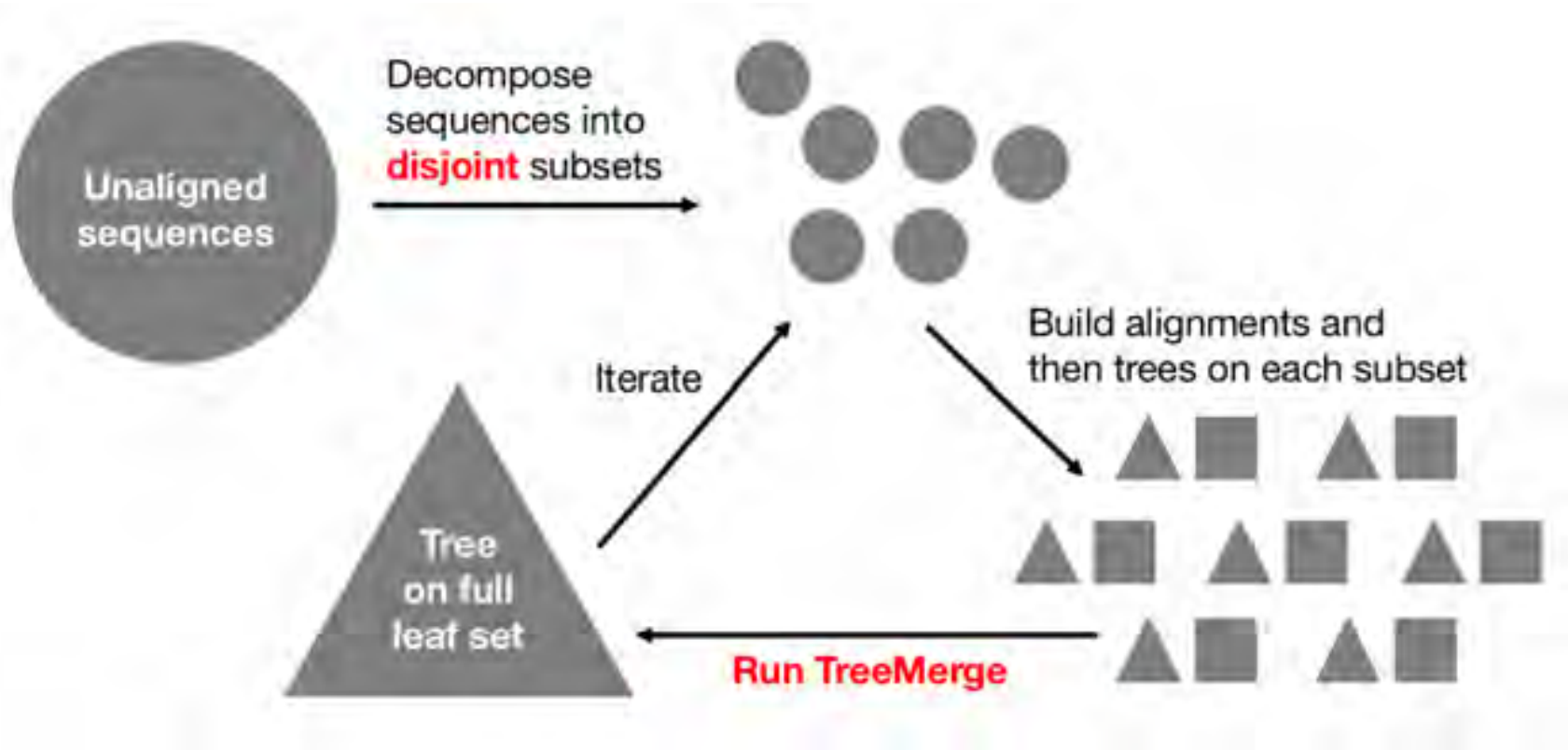
Based on our observations, it would be good to avoid estimating

1. Multiple sequence alignment on the full set of sequences
2. ML tree on the full multiple sequence alignment
3. Supertree

Divide-and-Conquer with DACTAL [Nelesen et al., 2012]

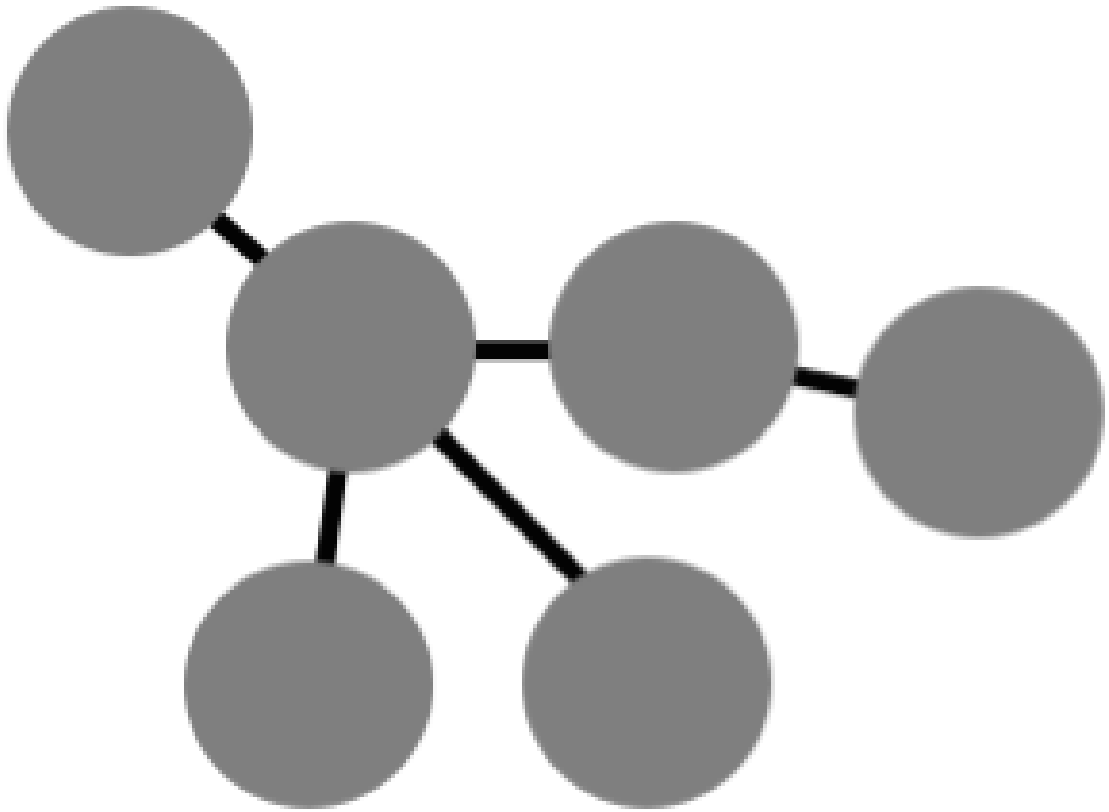


TERADACTAL Algorithm



TreeMerge: Step 1

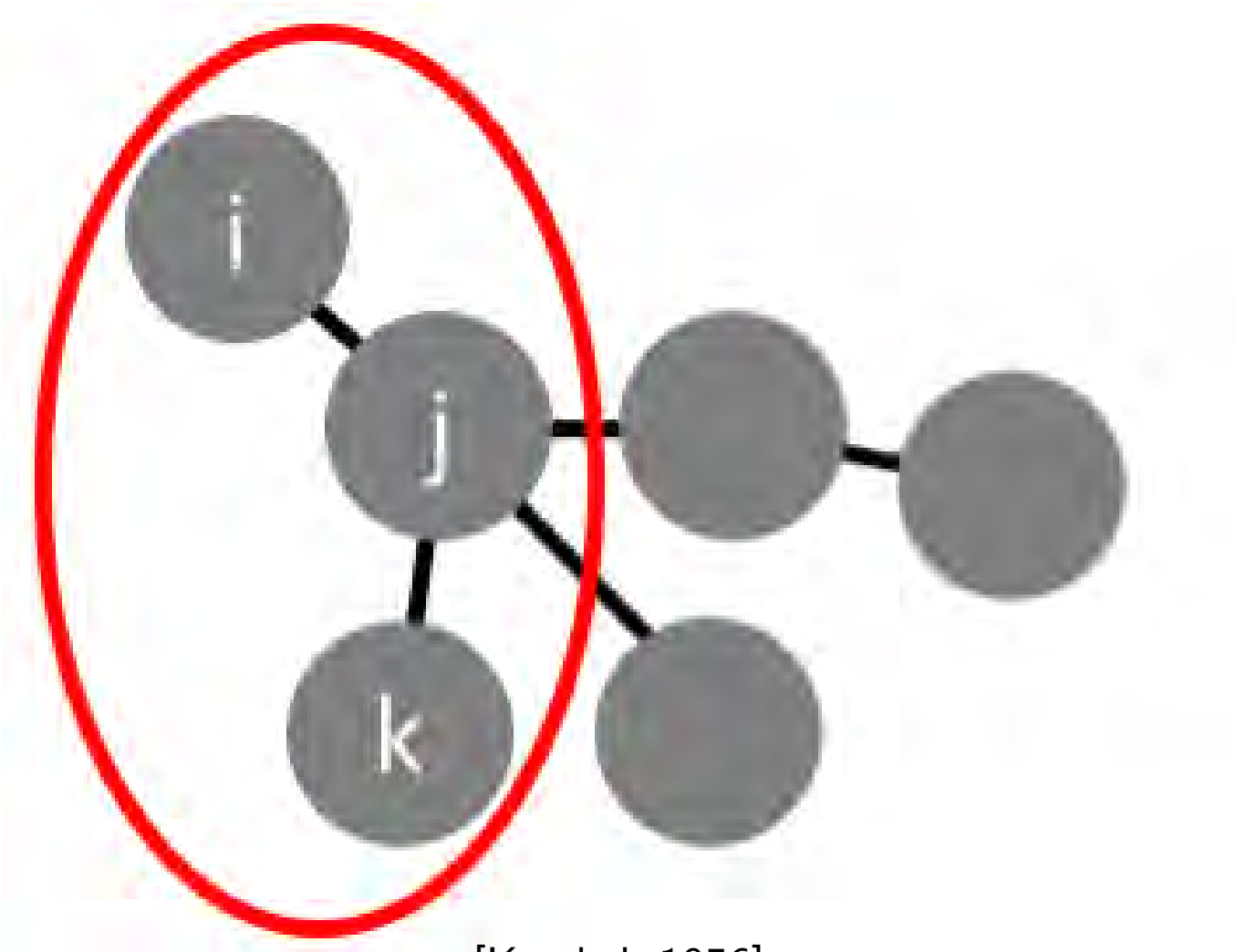
Create a minimum spanning tree on the disjoint subsets.



[Kruskal, 1956]

TreeMerge: Step 1

Create a minimum spanning tree on the disjoint subsets.



[Kruskal, 1956]

TreeMerge: Step 2

Merge trees T_i and T_j into tree T_{ij} such that $T_{ij}|_{L_i} = T_i$ and $T_{ij}|_{L_j} = T_j$ (multiple solutions).

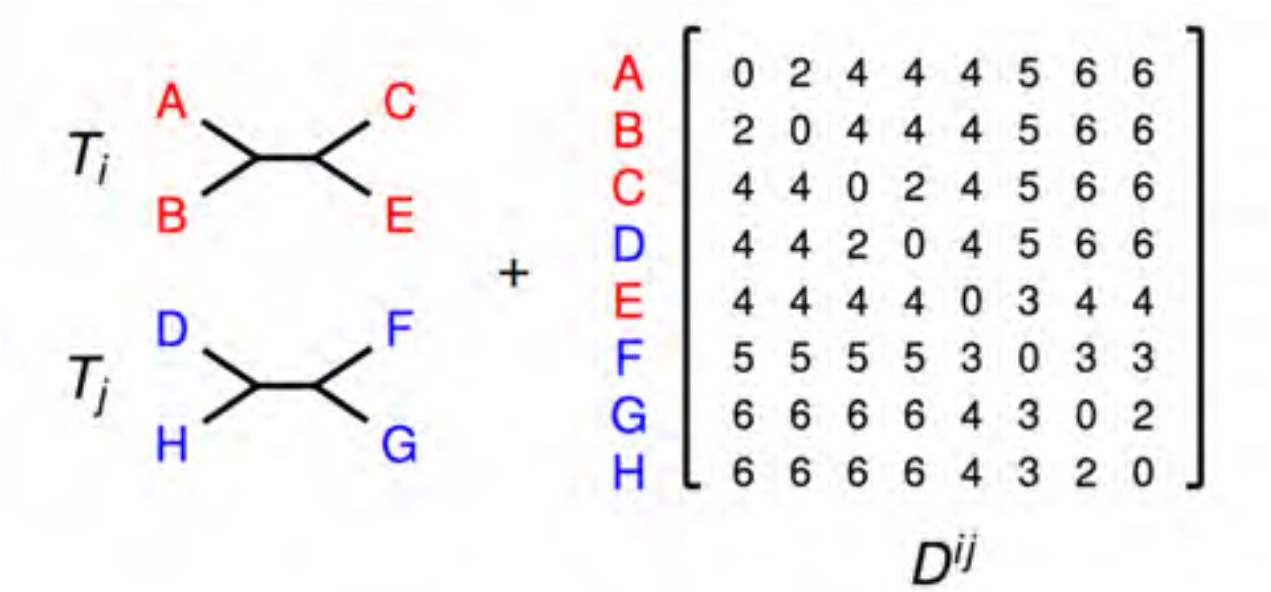
1. Merge two alignments A_i and A_j into A_{ij} using an existing technique (e.g., OPAL [Wheeler and Kececioglu, 2007]).
2. Compute distance matrix D^{ij} from merged alignment A_{ij}.
3. Run NJMerge – our variant of Neighbor-Joining [Saitou and Nei, 1987] that takes both a distance matrix and a set of constraint trees as input.

TreeMerge: Step 2

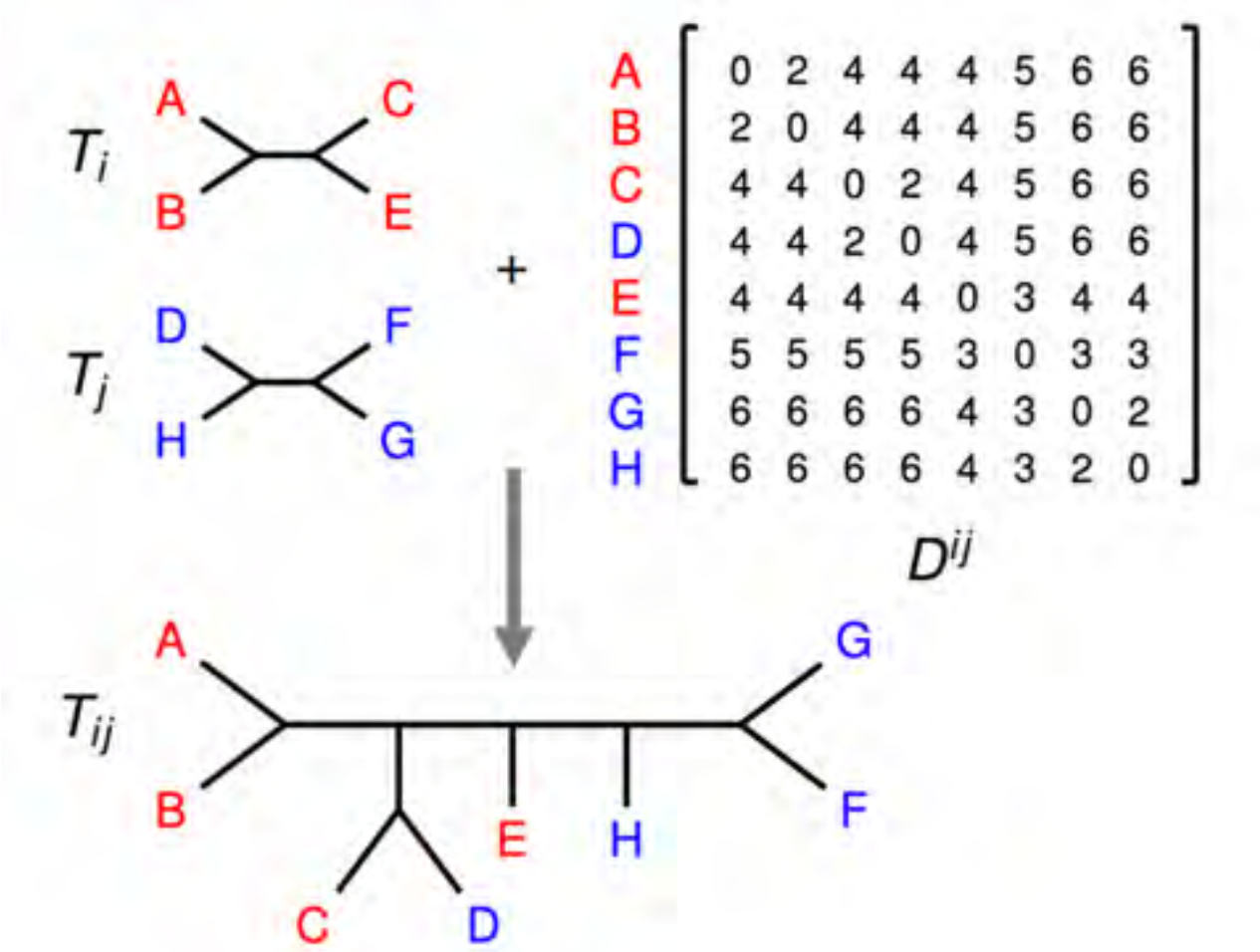
Merge trees T_i and T_j into tree T_{ij} such that $T_{ij}|_{L_i} = T_i$ and $T_{ij}|_{L_j} = T_j$ (multiple solutions).

1. Merge two alignments A_i and A_j into A_{ij} using an existing technique (e.g., OPAL [Wheeler and Kececioglu, 2007]).
2. Compute distance matrix D^{ij} from merged alignment A_{ij}.
3. Run NJMerge – our variant of Neighbor-Joining [Saitou and Nei, 1987] that takes both a distance matrix and a set of constraint trees as input.

NJMerge: Constrained Neighbor-Joining

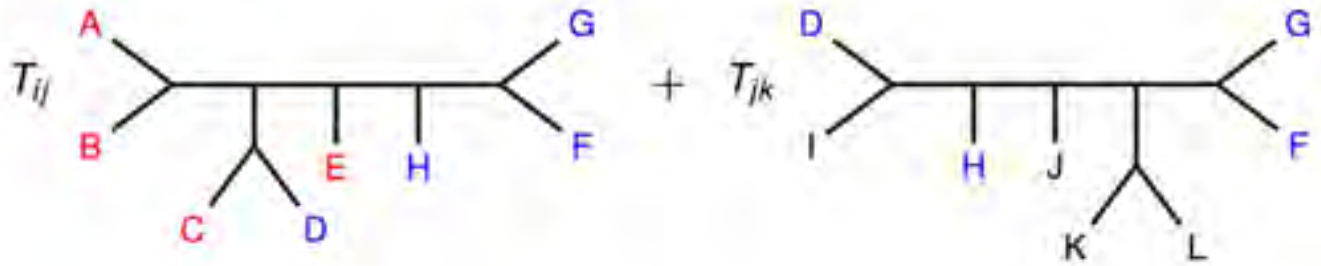


NJMerge: Constrained Neighbor-Joining



TreeMerge: Step 3

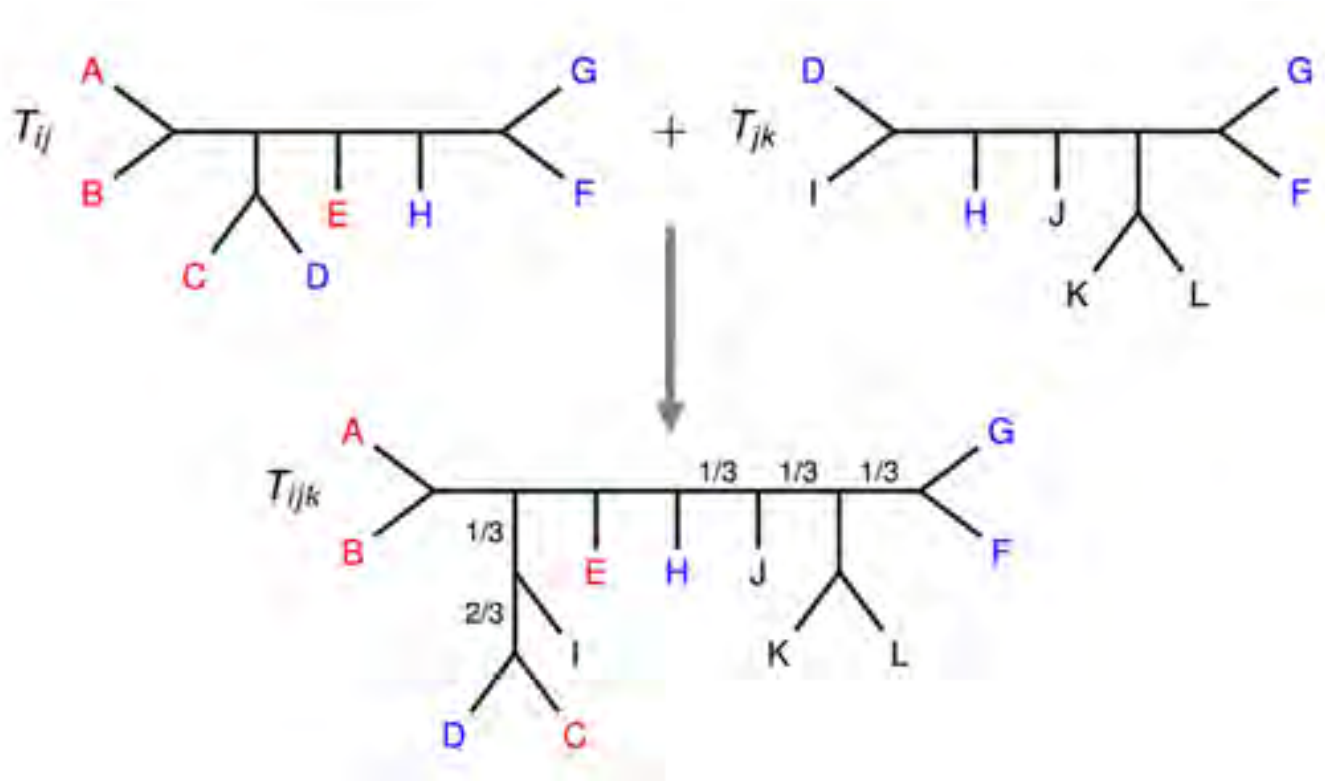
Combine pairs of merged trees using branch lengths, e.g., trees T_{ij} and T_{jk} are combined through T_j (blue).



NOTE: Unlabeled branches have length one for simplicity.

TreeMerge: Step 3

Combine pairs of merged trees using branch lengths, e.g., trees T_{ij} and T_{jk} are combined through T_j (blue).



NOTE: Unlabeled branches have length one for simplicity.

Advantages

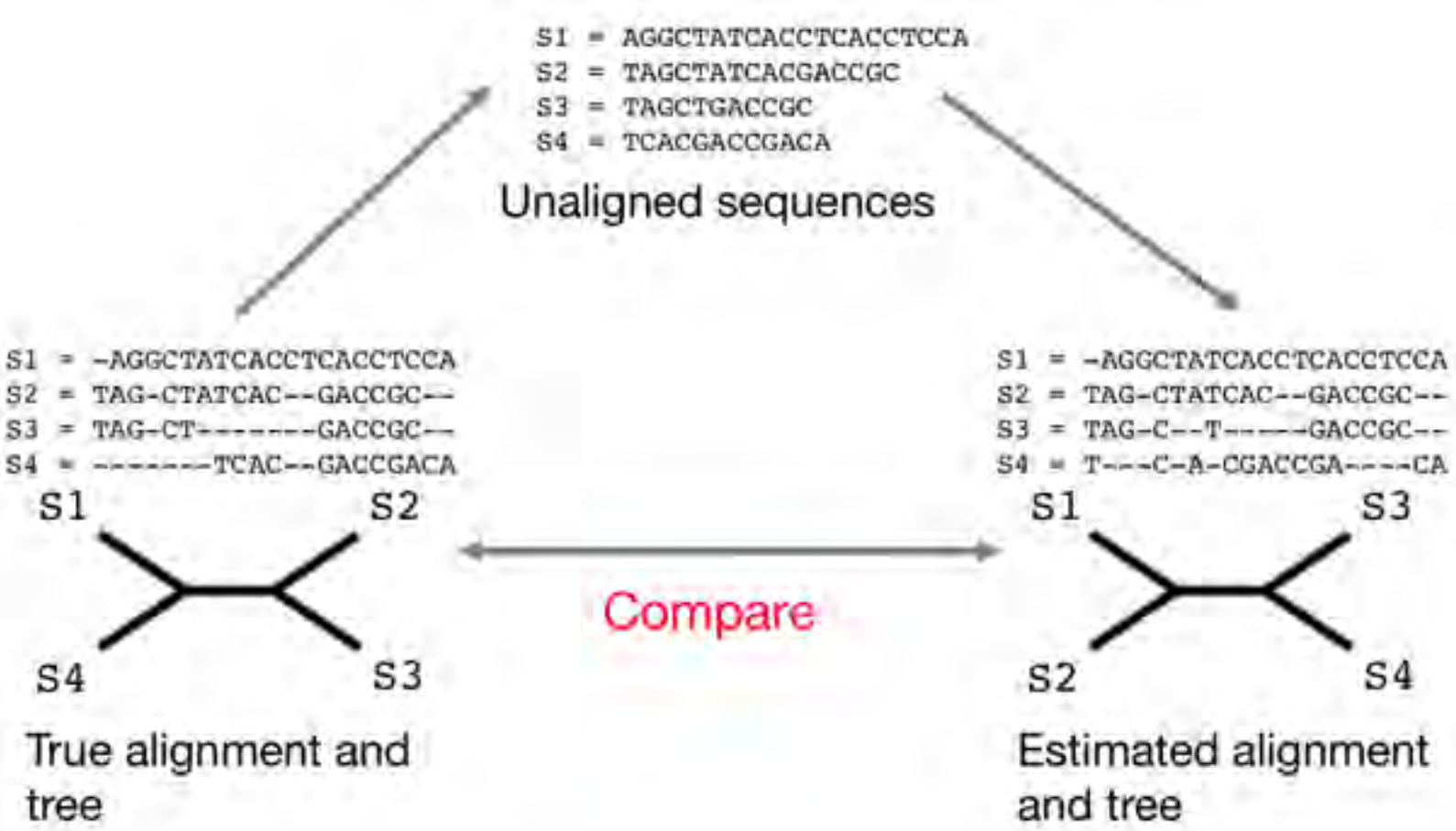
TERADACTAL

- No multiple sequence alignment estimation on the full dataset
- No Maximum Likelihood tree estimation on the full dataset
- No supertree estimation

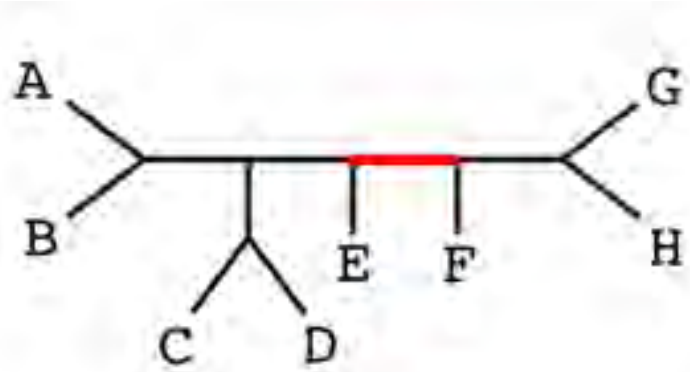
TreeMerge

- Polynomial Time
- Parallel
  - (Step 2) Pairs of alignments and trees can be merged in an embarrassingly parallel fashion.
  - (Step 3) Merged tree pairs can be combined in parallel, as long as they do not share edges in the minimum spanning tree.

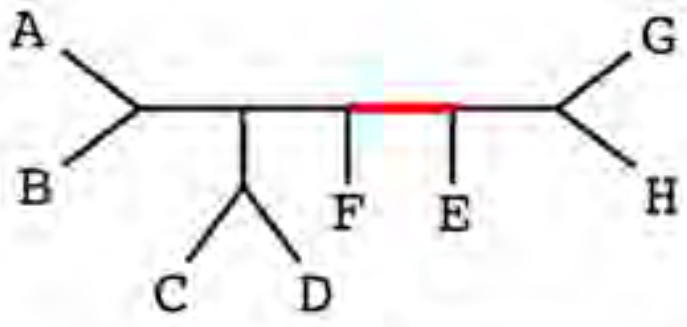
Simulation Studies



Quantifying Tree Error



True tree



Estimated tree

Simulation Study

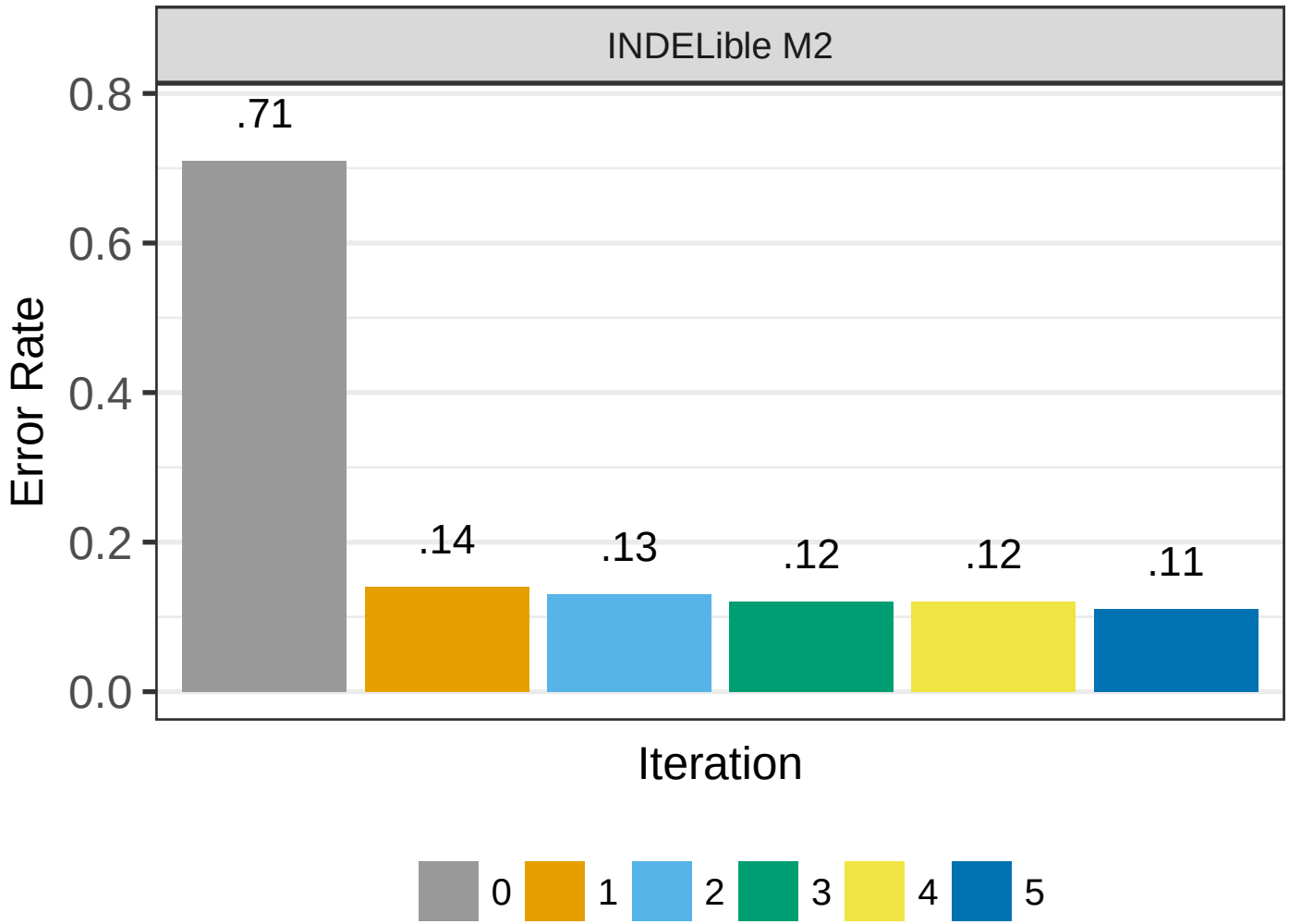
Compare TERADACTAL to

- 2 alignment-free methods
- 3 multiple sequence alignment methods (2 shown)
- 2 distance methods (1 shown)
- 2 Maximum Likelihood methods (1 Shown)

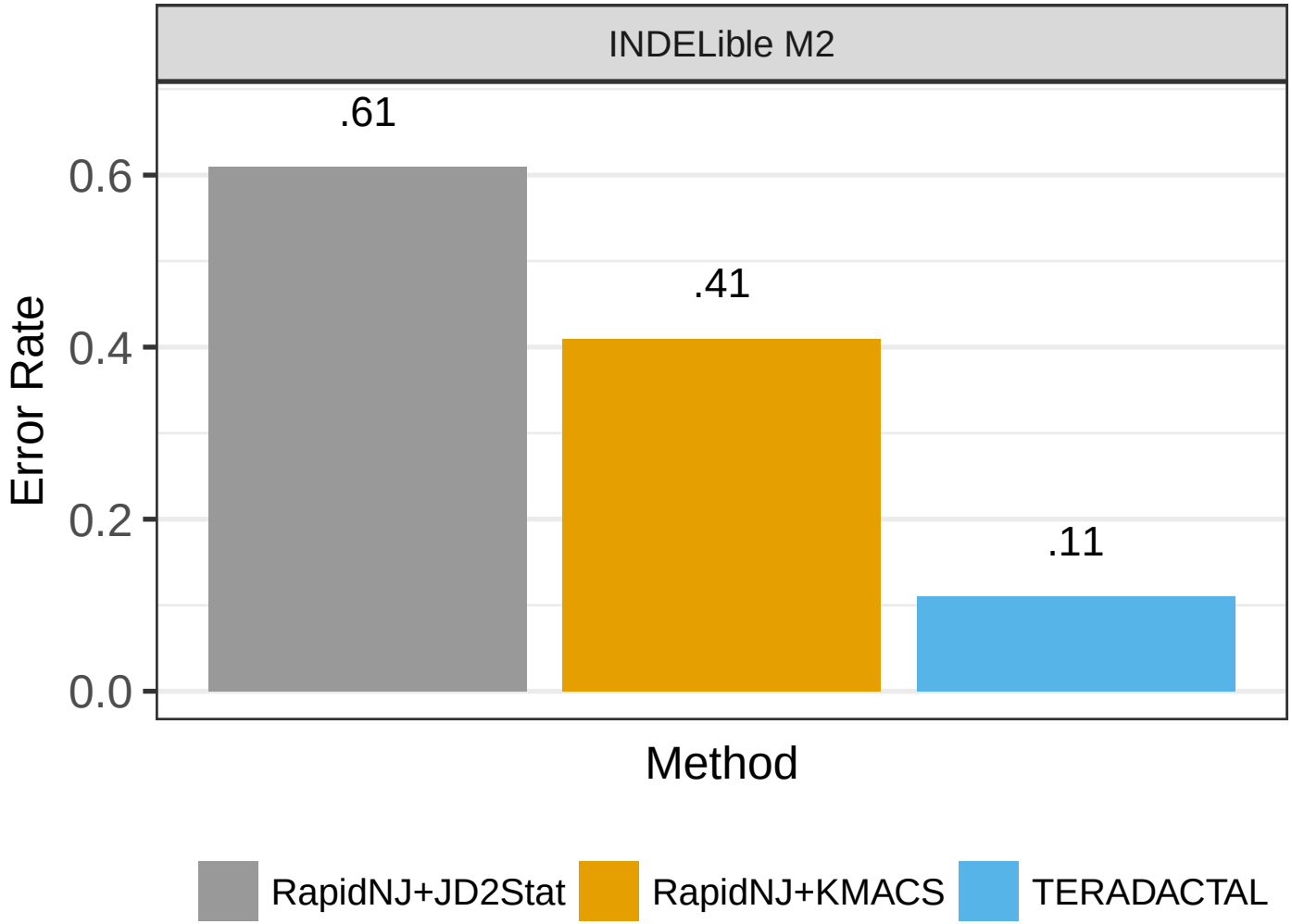
on simulated datasets from Mirarab et al., 2015.

- 10,000 sequences
- 4 model conditions each with 10 replicate datasets (1 shown)

TERADACTAL Iterations

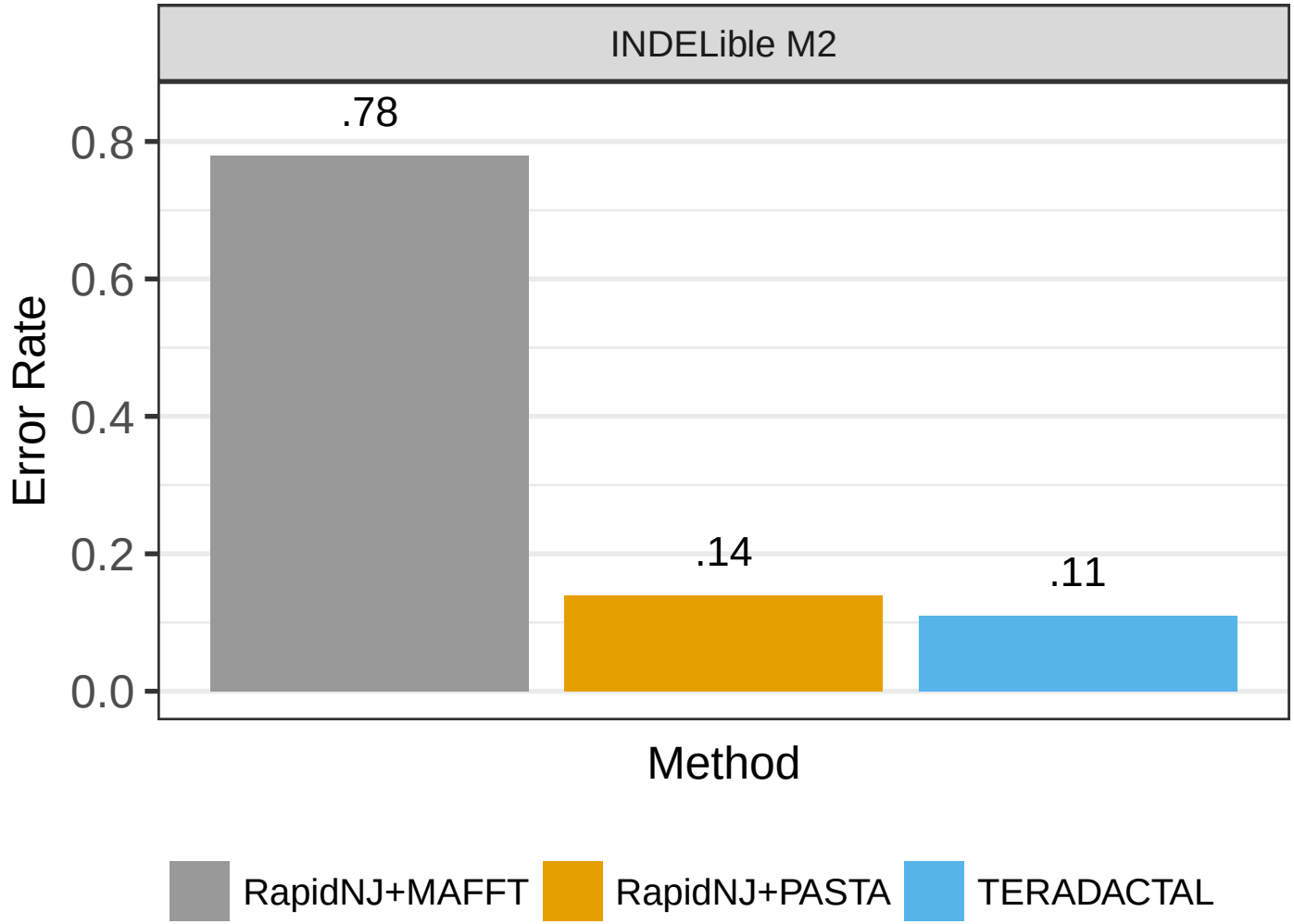


TERADACTAL versus Alignment-Free Methods



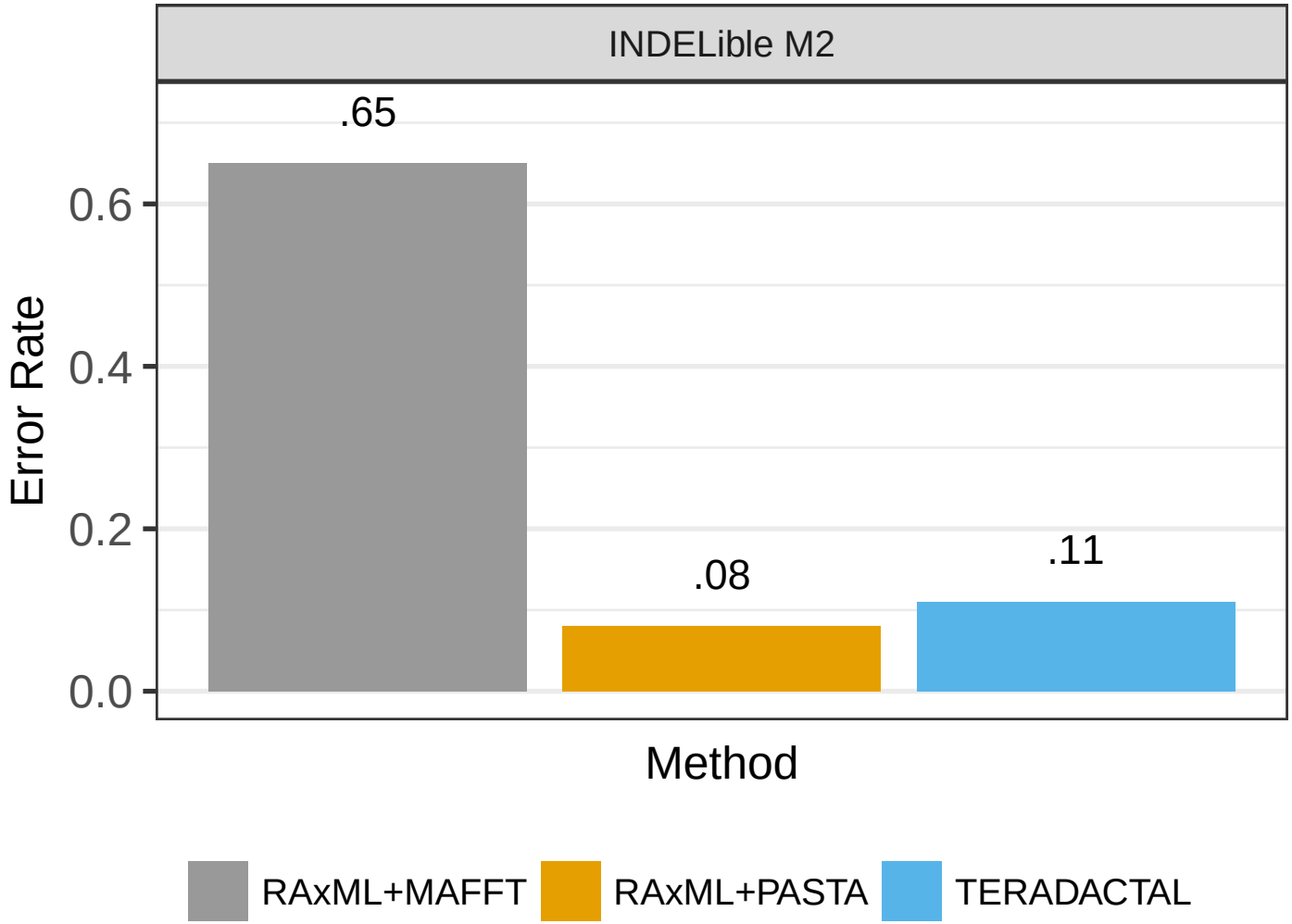
[Simonsen et al., 2008; Chan et al., 2014; Leimeister and Morgenstern, 2014]

TERADACTAL versus Two-Phase Methods (NJ)



[Simonsen et al., 2008; Katoh and Standley, 2013; Mirarab et al., 2015]

TERADACTAL versus Two-Phase Methods (ML)



[Stamatakis, 2006; Katoh and Standley, 2013; Mirarab et al., 2015]

Conclusions

We designed, prototyped, and tested a method that achieves similar error rates to the leading two-phase phylogeny estimation methods but is highly parallel and avoids

1. Multiple sequence alignment estimation on the full dataset
2. Maximum likelihood tree estimation on the full dataset
3. Supertree estimation

Future Work

Scale out to one million sequences.

Why Blue Waters

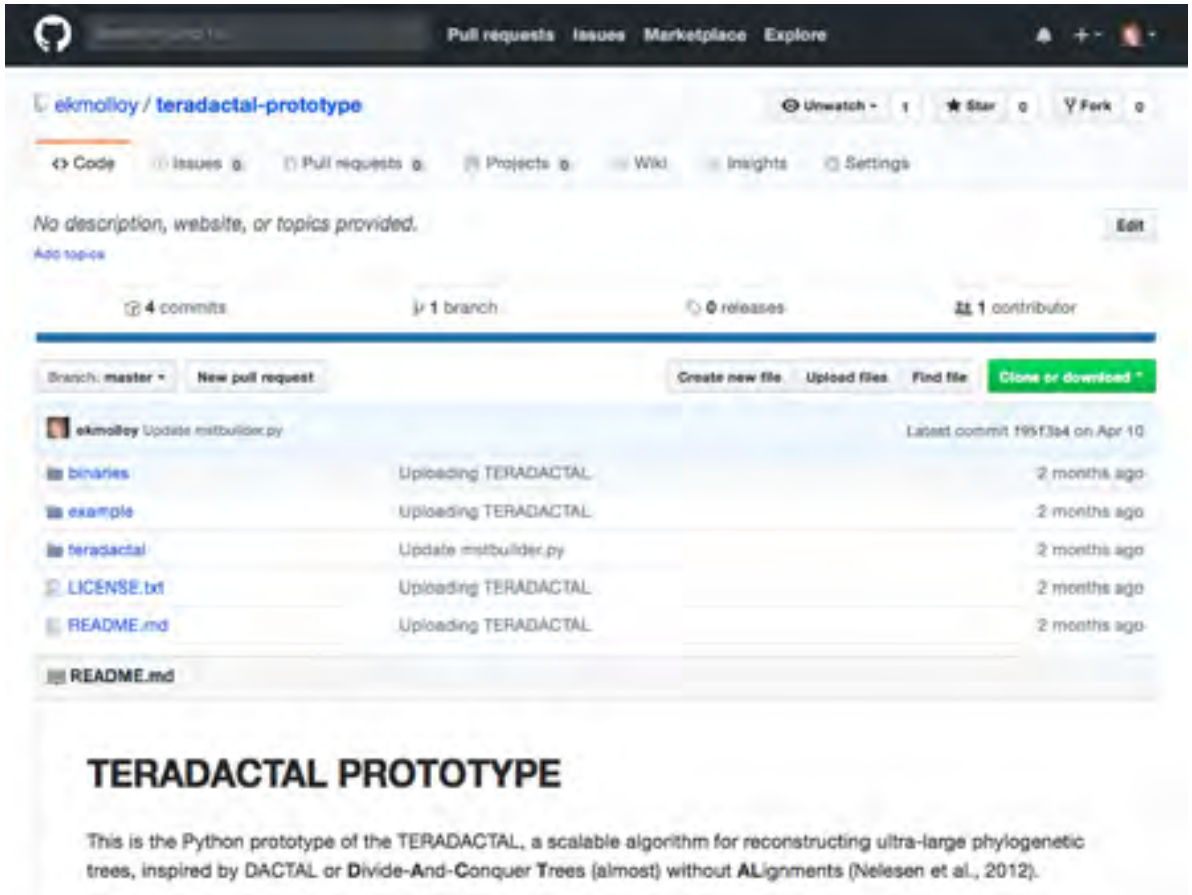
Blue Waters was used to

- Demonstrate that codes (e.g., FastTree-2, PASTA, RAxML) could not run on Blue Waters on datasets with 1 million sequences (Run out of memory!)
- Simulation study completed in < 1 month but would have required > 1 year using our 4 campus cluster nodes

Research Products

Paper under review at the 17th European Conference on Computational Biology (ECCB 2018).

Github: **github.com/ekmolloy/teradactal-prototype**



Other Research Products from General Allocation

- Nute, Saleh & Warnow Benchmarking statistical multiple sequence alignment. Under review at *Systematic Biology*.
- Nute, Chou, Molloy & Warnow (2018). The Performance of Coalescent-Based Species Tree Estimation Methods under Models of Missing Data. *BMC Genomics* 19(Suppl 5):286.
- Vachaspati & Warnow (2018). SVDquest: Improving SVDquartets species tree estimation using exact optimization within a constrained search space. *Mol Phyl Evol* 124:122-136. Github: **github.com/pranjalv123/SVDquest**
- Vachaspati & Warnow (2018). SIESTA: Enhancing searches for optimal supertrees and species trees. *BMC Genomics* 19(Suppl 5):252. Github: **github.com/pranjalv123/SIESTA**

Acknowledgements

This work was supported by the National Science Foundation

- Blue Waters Sustained-Petascale Computing Project (OCI-0725070 and ACI-1238993)
  - General Allocation (PI: Warnow)
  - Exploratory Allocation (PI: Gropp)
- Graduate Research Fellowship Program (DGE-1144245)
- Graph-Theoretic Algorithms to Improve Phylogenomic Analyses (CCF-1535977)

and the Ira & Debra Cohen Graduate Fellowship in Computer Science.

References

- Felsenstein, J. (1981). Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of Molecular Evolution* 17(6):368–376.
- Roch, S. (2006). A short proof that phylogenetic tree reconstruction by maximum likelihood is hard. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 3(1):92–94.
- Stamatakis, A. (2006). RAxML-VI-HPC: maximum likelihood-based phylogenetic analyses with thousands of taxa and mixed models. *Bioinformatics* 22(21):2688–2690.
- Price, M.N., P. S. Dehal, and A. P. Arkin. (2010). FastTree 2 – Approximately Maximum Likelihood Trees for Large Alignments. *PLoS ONE* 5(3):1–10.

References

- Kozlov, A.M., A.J. Aberer, and A. Stamatakis. (2015). ExaML version 3: a tool for phylogenomic analyses on supercomputers. *Bioinformatics* 31.
- Nelesen, S., et al., (2012). DACTAL: Divide-And-Conquer Trees (almost) without Alignments. *Bioinformatics* 28(12):i274–i282.
- Steel, M. (1992). The complexity of reconstructing trees from qualitative characters and subtrees. *Journal of Classification* 9:91-116.
- Jiang, T., P. Kearney, and M. Li. (2001). A polynomial time approximation scheme for inferring evolutationay trees from quartet topologies and its application. *SIAM Journal on Computing* 30(6):1942–1961.

References

- Bansal, M.S., et al., (2010). Robinson-Foulds Supertrees. *Algorithms for Molecular Biology* 5(1).
- Kruskal, J. B. (1956). On the shortest spanning subtree of a graph and the traveling salesman problem". *Proceedings of the American Mathematical Society* 7:48–50.
- Saitou, N. and M. Nei. (1987). The neighbor-joining method: a new method for reconstruction of phylogenetic trees. *Molecular Biology and Evolution* 4: 406–425.
- Mirarab, S., et al., (2015). PASTA: Ultra-Large Multiple Sequence Alignment for Nucleotide and Amino-Acid Sequences. *Journal of Computational Biology* 22(5):377–386.

References

- Wheeler, T.J. and J. D. Kececioglu. (2007). Multiple alignment by aligning alignments. *Bioinformatics* 23(13):i559.
- Chan, C.X. et al., (2014). Inferring phylogenies of evolving sequences without multiple sequence alignment. *Scientific Reports* 4:6504.
- Leimeister, C.-A. and B. Morgenstern. (2014). kmacs: the k-Mismatch Average Common Substring Approach to alignment-free sequence comparison. *Bioinformatics* 30(14):2000–2008.
- Simonsen, M., T. Mailund, and C.N.S. Pedersen. (2008) "Rapid Neighbour-Joining." *Algorithms in Bioinformatics* 5251:113–122.
- Katoh, K. and D.M. Standley. (2013). MAFFT Multiple Sequence Alignment Software Version 7: Improvements in Performance and Usability. *Molecular Biology and Evolution* 30(4):772.