Computing and Updating Large Alignments and Trees

Tandy Warnow

Support: NSF grant - #2006069
Sandia National Laboratories LDRD

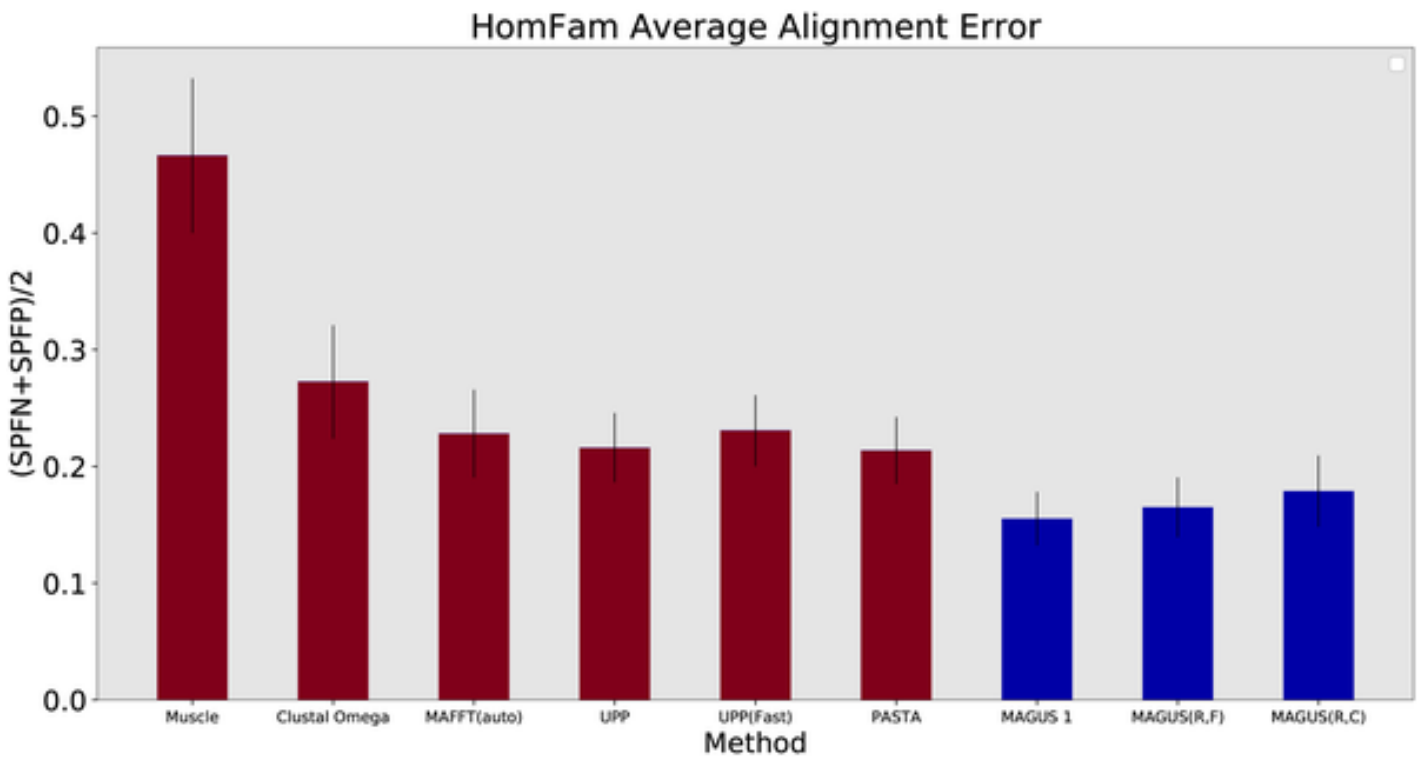
Overview – Large alignments and trees

- Constructing MSA or tree:
  - Datasets can be very large (hundreds of thousands of sequences, even 1,000,000).
  - Some sequences are very short, or perhaps very long
  - Species trees instead of gene trees
  - Uncertain homology
- Updating an alignment or tree
  - Add sequences to an alignment or tree instead of recomputing from scratch

Our methods for computing very large alignments or trees

- Methods for computing very large alignments
  - SATé, PASTA, and MAGUS: best when there is limited sequence length heterogeneity
  - WITCH, WITCH-ng, and HMMerge: best when there is sequence length heterogeneity (especially lots of very short sequences)
- Methods for computing very large maximum likelihood trees
  - GTM pipelines (Park et al.) – improves RAxML, IQ-TREE2

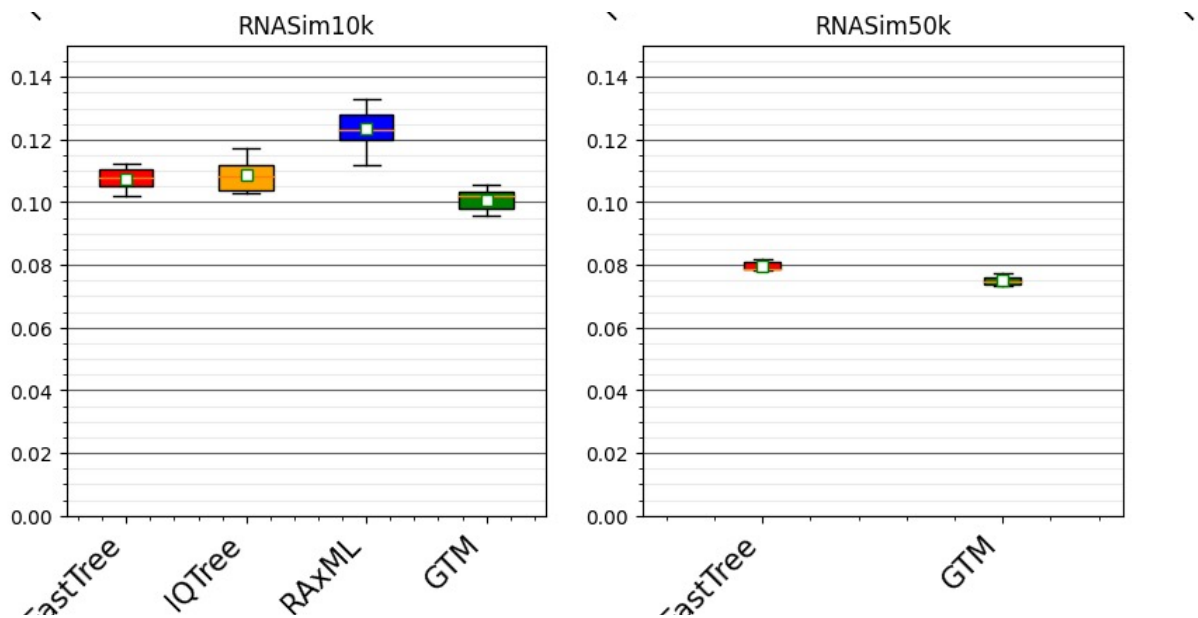
MAGUS – Highly Accurate Multiple Sequence Alignment for large datasets



Smirnov V (2021) Recursive MAGUS: Scalable and accurate multiple sequence alignment. PLOS Computational Biology 17(10): e1008950. https://doi.org/10.1371/journal.pcbi.1008950

https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1008950

GTM pipelines: Improving Large-scale Maximum Likelihood Tree Search



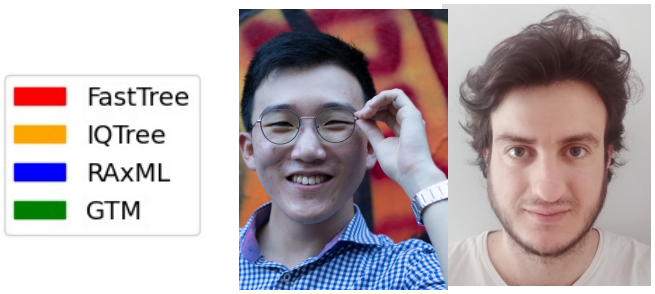


Figure 2 from "Disjoint Tree Mergers for Large-Scale Maximum Likelihood Tree Estimation", Park, Zaharias, and Warnow, *Algorithms 2021*

Trends
- On RNASim10k: GTM most accurate topology
- On RNASim50K:
  - IQTree failed
  - RAxML had nearly 100% error
  - GTM most accurate

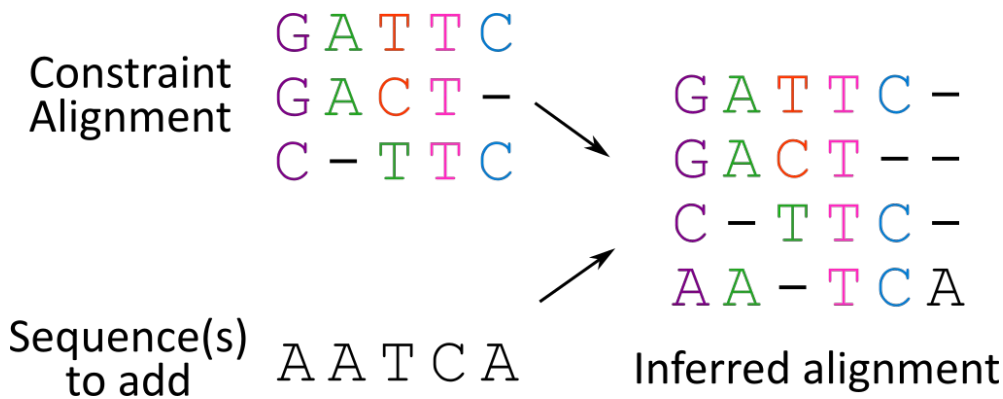
Methods for updating very large alignments or trees

- Methods for adding sequences to alignments
  - MAFFT-add and MAFFT-linsi-add (no HMM)
  - UPP-add, WITCH-ng-add (HMM-based)
  - EMMA (Chengze Shen et al.)

- Methods for adding sequences to very large trees (phylogenetic placement)
  - pplacer (Matsen et al., 2010) and EPA-ng (Barbera et al, 2019)
  - SCAMPP: scaling pplacer to large trees (Wedell et al., 2022)
  - BSCAMPP: scaling EPA-ng to large trees (Wedell et al., 2023)

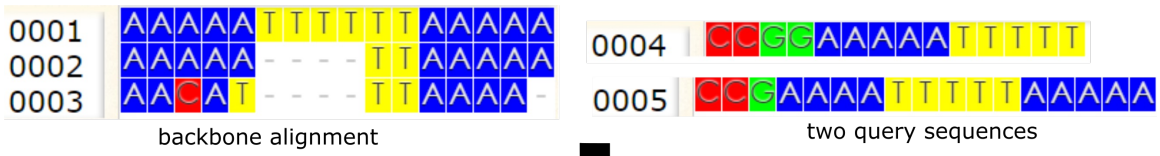
Intro to EMMA: Adding sequences into a constraint alignment

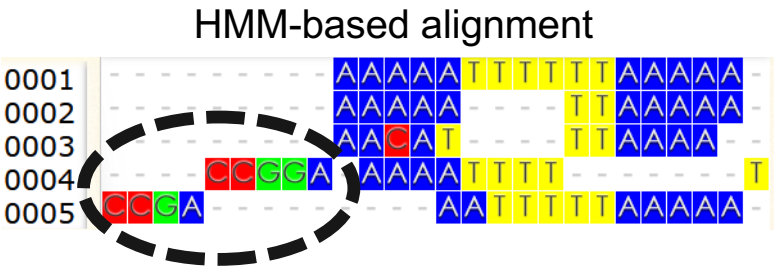
- Adding sequences to a **constraint alignment** useful for:
  1. De novo alignment
  2. Updating an existing alignment

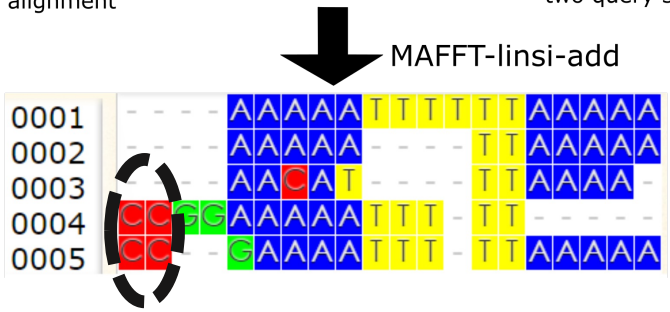


the constraint alignment is preserved

MAFFT-linsi --add vs. HMM-based methods

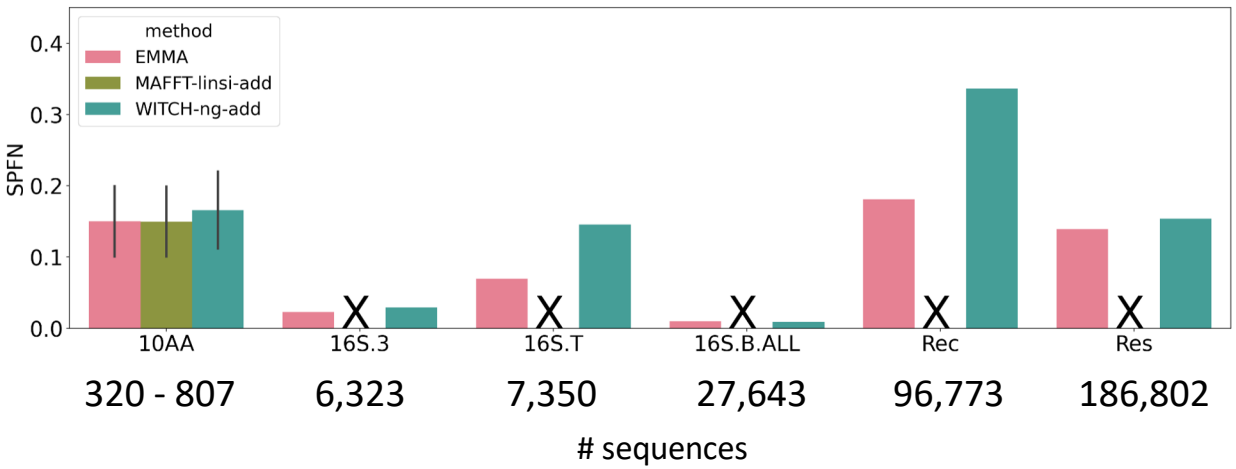






- **HMM-based** methods (UPP, WITCH, etc.):
  - Can only find homologous pairs to the **existing columns in the constraint alignment**.
- MAFFT-linsi --add can find the two homologous pairs “C-C” in the first two columns.

EMMA - experimental results - large random constraint



- **Large random constraint**
- Both NT and AA datasets
- "MAFFT-add had poor accuracy, and is omitted
- X" - failed to run

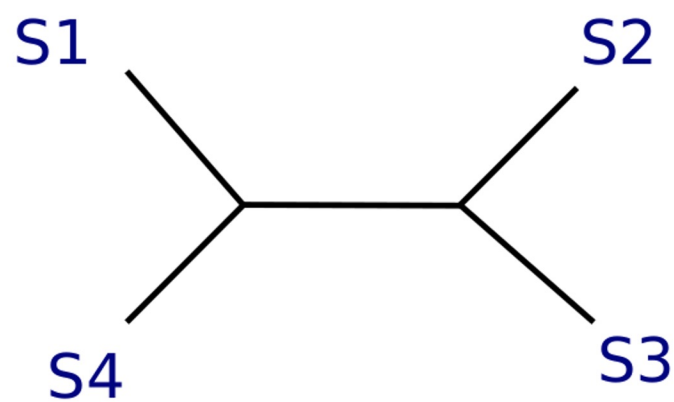
Observations:

- EMMA is the most accurate method, and can scale to the largest dataset with 186,802 AA sequences
- MAFFT-linsi-add had high accuracy but only completed on one dataset

Intro to SCAMPP and BSCAMPP

Phylogenetic placement problem: *Given a query sequence and multiple sequence alignment, determine the placement into an existing 'backbone' tree.*

```
S1  = -AGGCTATCACCTGACCTCCA-AA
S2  = TAG-CTATCAC--GACCGC--GCA
S3  = TAG-CT-------GACCGC--GCT
S4  = TAC----TCAC--GACCGACAGCT
Q1  = -------T-A--AAAC--------
```

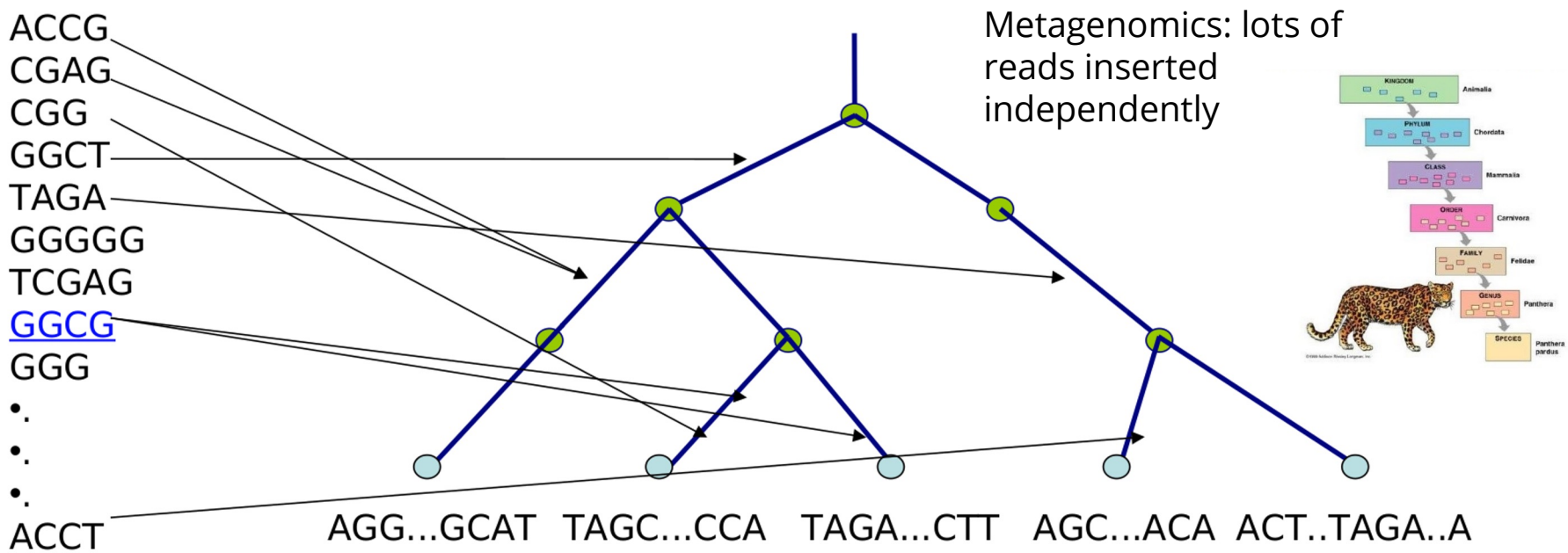


Placement into a taxonomy of full-length sequences

Fragmentary sequences from some gene

Full-length sequences for same gene, and an alignment and a tree

Metagenomics: lots of reads inserted independently



Leading Methods for Phylogenetic Placement

Maximum likelihood methods (expensive to run):

- **pplacer** (Matsen et al., 2010)
- **EPA-ng** (Barbera et al., 2019)

Distance-based methods:

- **APPLES-2** (Balaban et al., 2021).

Parsimony-based methods:

- **UShER** (Turakhia et al., 2021)

Alignment-free methods:

- **App-SpaM** (Blanke et al., 2021)
- **RAPPAS** (Linard et al., 2019)

pplacer and EPA-ng limited to small backbone trees

EPA-ng scales sublinearly with number of queries

Our goal: Scaling ML phylogenetic placement methods to large trees and many queries

SCAMPP Framework (Wedell et al., TCBB 2022)

Designed to allow existing phylogenetic placement methods use larger backbone trees.

Used with specified phylogenetic placement method (e.g., pplacer)

Input: Backbone tree with branch lengths, alignment and **set of aligned query sequences**, and a subtree size.

For each query sequence:

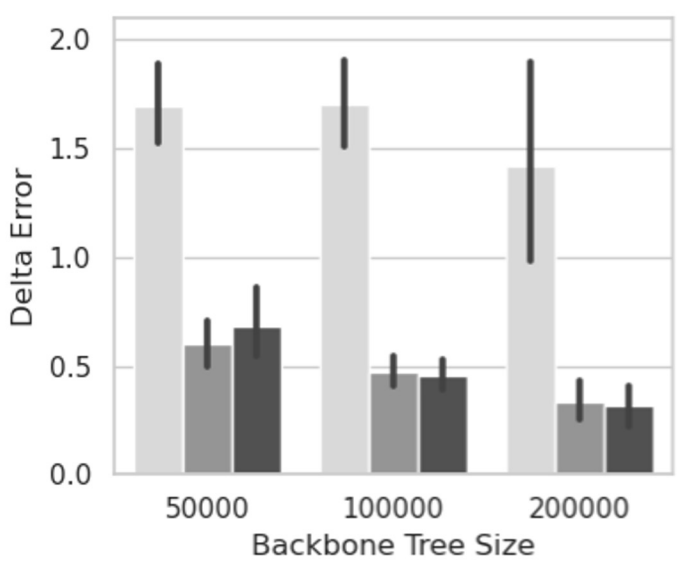
Stage 1- Extract placement subtree from backbone tree

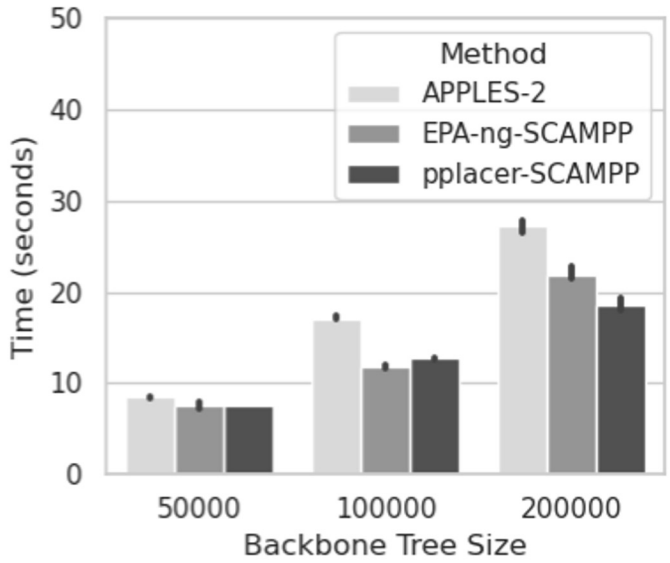
Stage 2 - Use pplacer to find edge in placement subtree and location and distal length along placement edge.

Stage 3 - Find edge in backbone tree using branch lengths.

SCAMPP Results

SCAMPP implemented with EPA-ng and pplacer show better accuracy and competitive runtime than APPLES-2 for **SINGLE QUERY SEQUENCE PLACEMENT.**

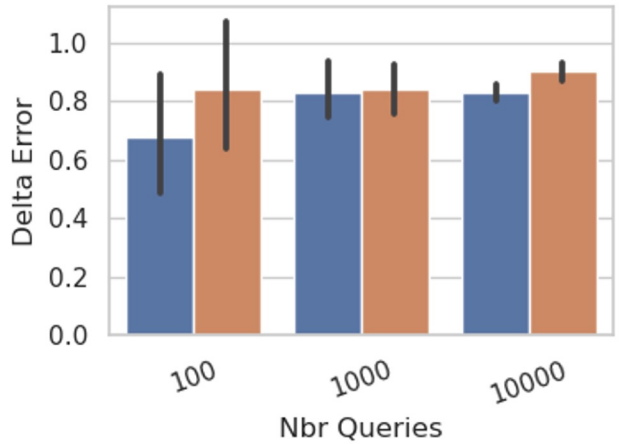


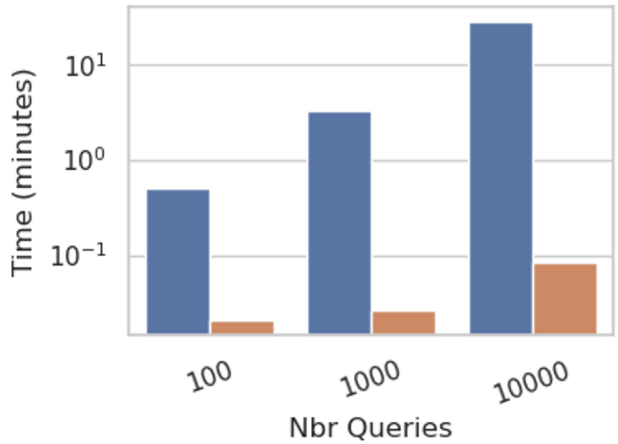


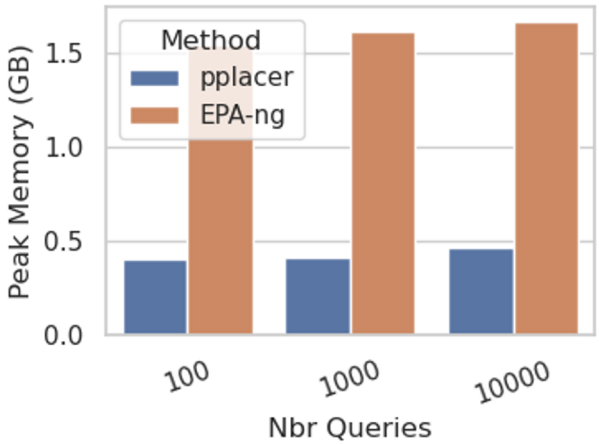
Detla Error: increase in topological error produced by adding query.

EPA-ng Scales the Number of Queries

*EPA-ng's runtime scales **sublinearly** with respect to the number of queries.*







This motivates BSCAMPP

Batch-SCAMPP initial results

Table 3 Testing Data Results for Method Comparison on RNASim (50,000 sequences in the backbone tree with 10,000 fragmentary query sequences).

	RNASim		
Method	Delta Error	Runtime (minutes)	Memory (GB)
BSCAMPP(e)	0.50	7.2	3.0
SCAMPP(e)	0.51	466.0	1.2
SCAMPP(p)	0.46	1421.3	0.2

BATCH-SCAMPP (2023) is a modified version of SCAMPP that is designed for use with EPA-ng, which scales sublinearly with number of query sequences, but cannot place into large trees

BSCAMPP(e) vs. Alignment Free

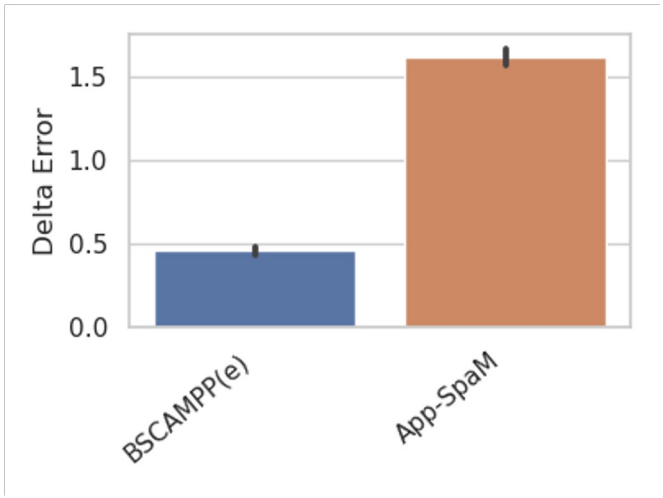
We show App-SpaM. RAPPAS failed due to memory.

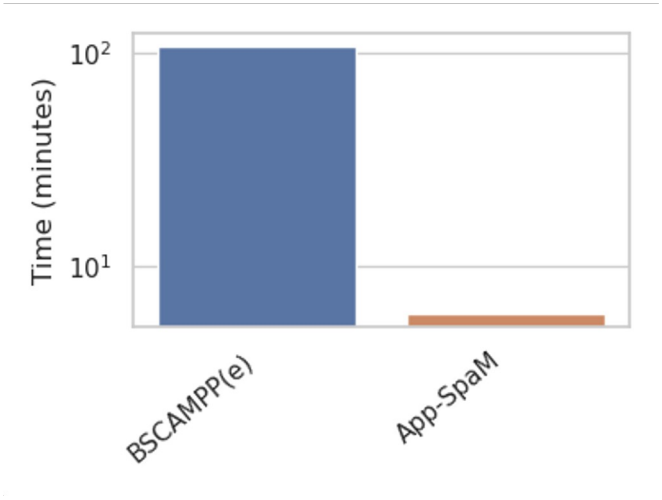
BSCAMPP uses estimated alignments using UPP.

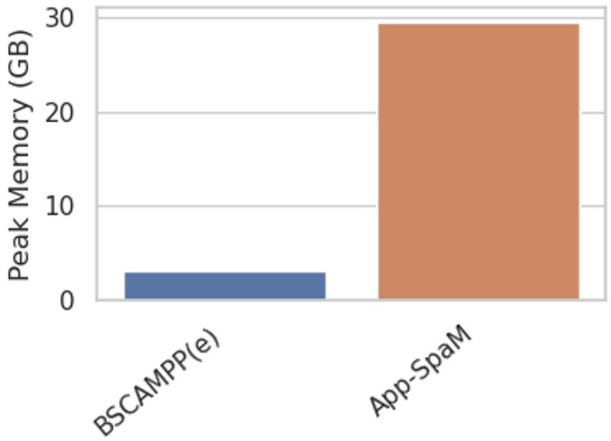
BSCAMPP runtime is largely estimating alignment.

RNASim 50K (training clade)

- Fragmentary queries (~150nt)
- 50,000 leaf backbone tree
- 10,000 query sequences







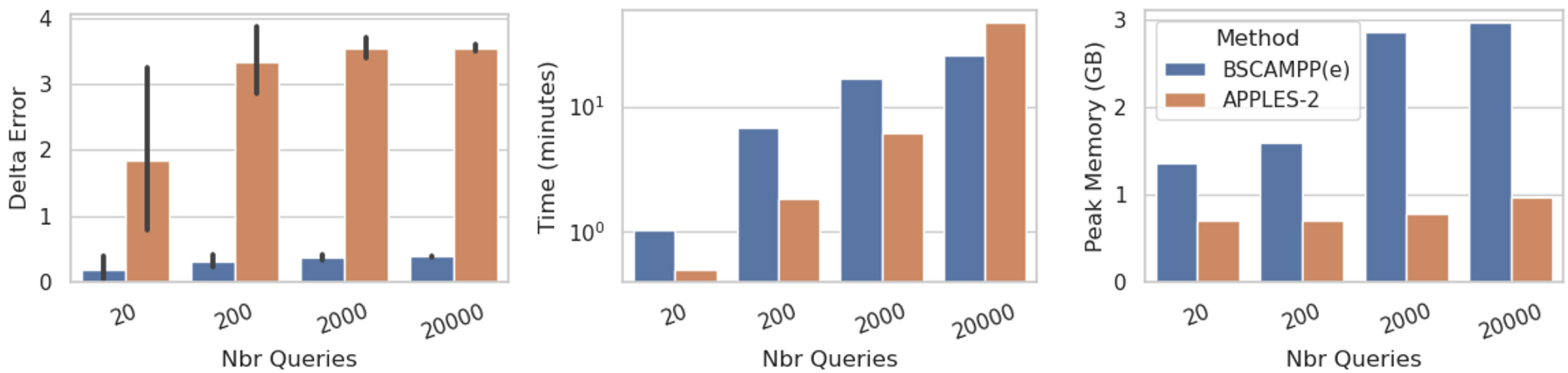
BSCAMPP(e) vs. APPLES-2

APPLES-2 is very fast and has low memory requirement, but has much higher placement error than Batch-SCAMPP(EPA-ng)

BSCAMPP(EPA-ng) runtime is sublinear with number of query sequences

RNASim 180K (testing clade)

- Fragmentary queries (~150nt)
- 180,000 leaf backbone tree
- Up to 20,000 query sequences



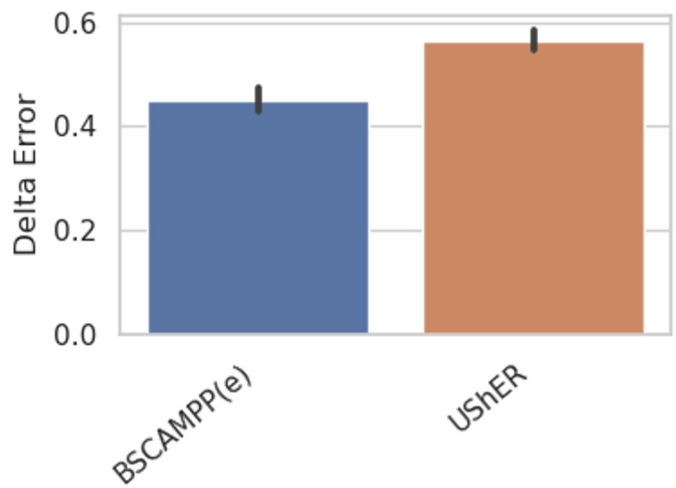
BSCAMPP(e) vs. UShER

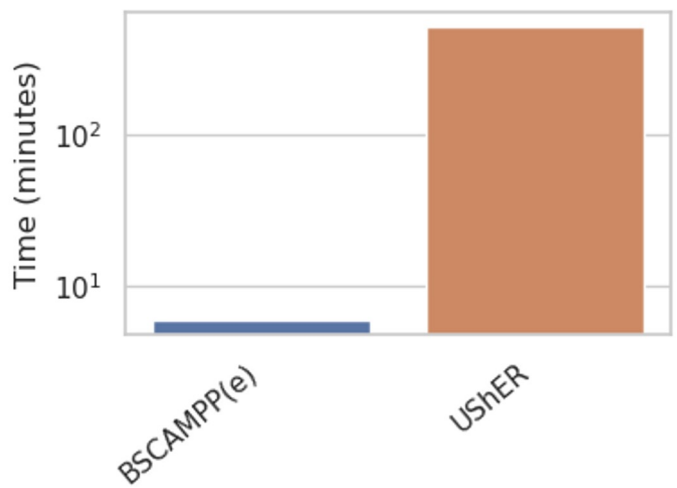
Using true alignments.

Memory usage less than 4GB for both methods.

RNASim 50K (training clade)

- Fragmentary queries (~150nt)
- 50,000 leaf backbone tree
- 10,000 query sequences





More UShER vs. BSCAMPP(e) Changing rate of evolution

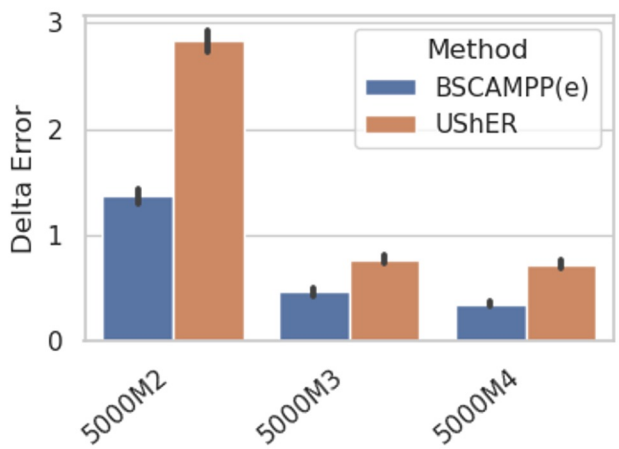
UShER still worse than BSCAMPP with lower rates of evolution

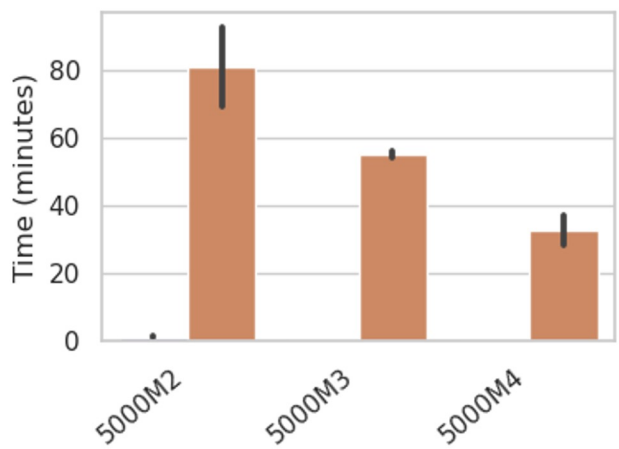
Placing 10,000 fragments onto 4,000 leaf backbone trees

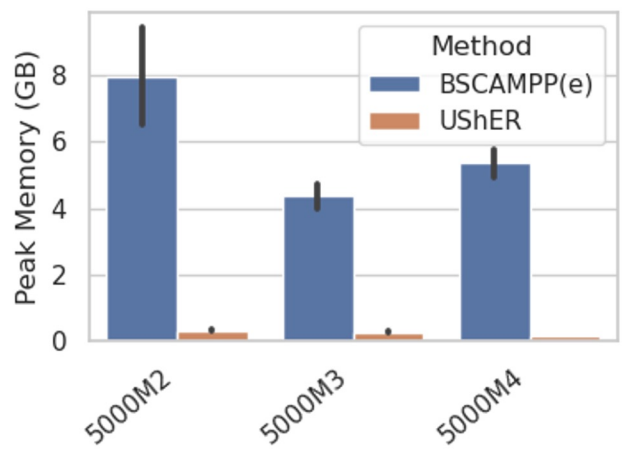
5000M2: high rate

5000M3: medium rate

5000M4: low rate



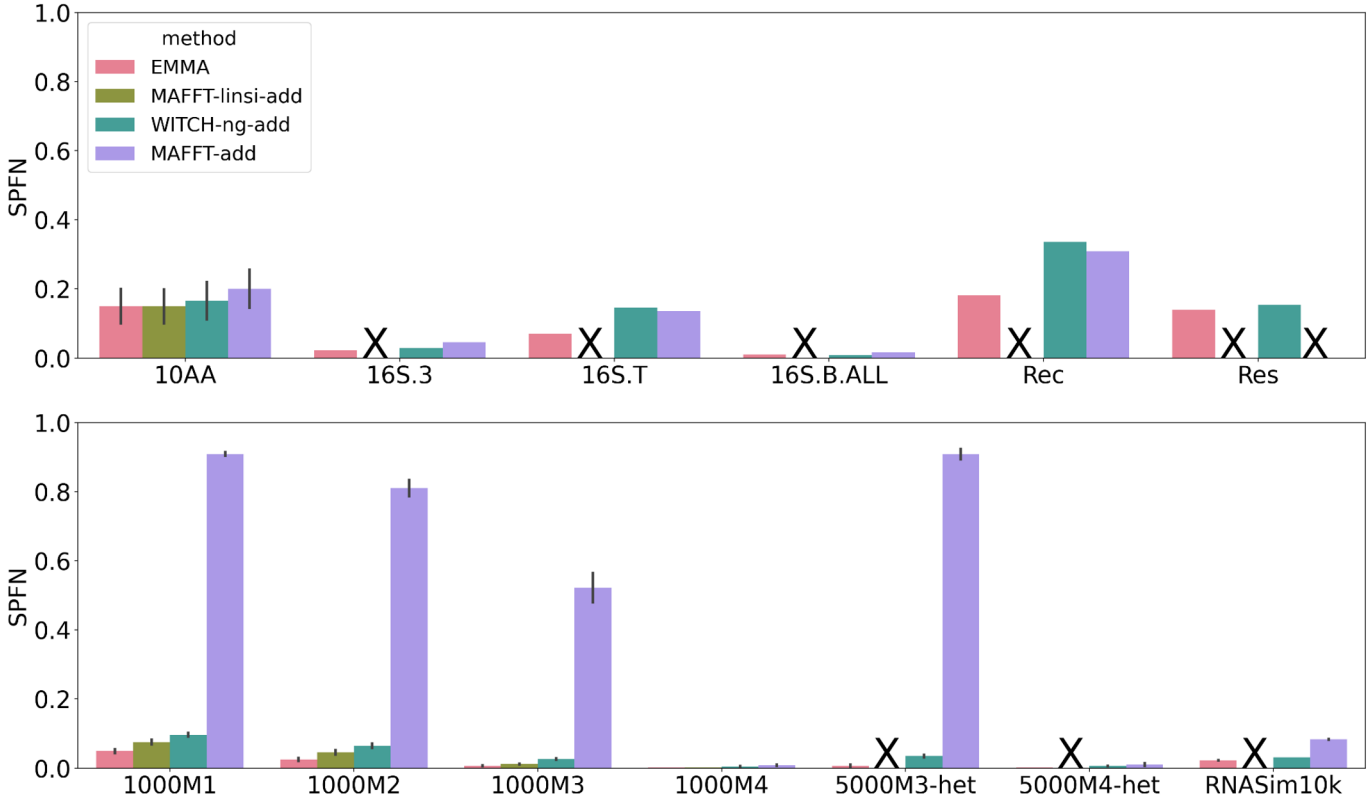




Summary

- New approaches to constructing and updating large-scale alignment and tree estimation, with outstanding accuracy
- All software available in open-source form on github
- We are looking for collaborations

Backup slides - results with MAFFT --add



Large random constraint