Divide and Conquer Helps Model-based Alignments

Siavash Mirarab
Department of Computer Science
University of Texas at Austin

This talk ...

- Topics (not in that exact order!):
  - Phylogenetic Placement Problem
  - Metagenomics
  - Hidden Mordov Models and their application to sequence search and alignment
  - SEPP
  - UPP

Phylogenetic Reconstruction

Start from unaligned sequences

Align all the unaligned sequences together to get a Multiple Sequence Alignment (MSA)

Build a phylogeny based on the MSA

unaligned sequences

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

Multiple Sequence Alignment

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

→

```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```

What is an alignment anyway?

(parent)

...ACGGTGCAGTTACCA...

Deletion

Substitution

Insertion

...ACCAGTTACCC...

(brother)

...ACGGTGCAGTTTACCTA...

(sister)

...ACGGTGCAGTTACCTA...

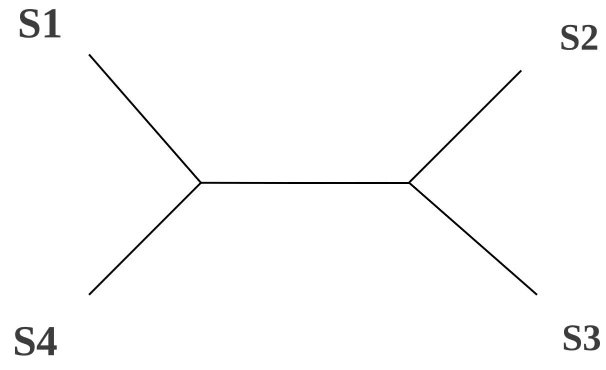
...AC----CAGTTACC.C...

The true multiple alignment reflects historical substitution, insertion, and deletion

Construct tree

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

```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```



Phylogenetic Reconstruction

Start from unaligned sequences

Align all the unaligned sequences together to get a Multiple Sequence Alignment (MSA)

Build a phylogeny based on the MSA

Phylogenetic Placement

Input:

A *backbone* alignment and tree

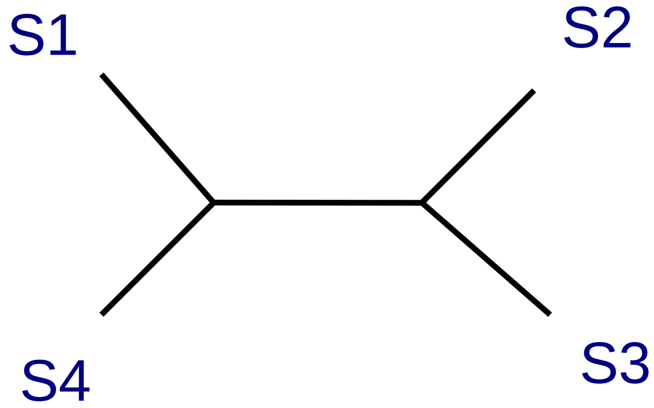
A set of *query* sequences

Goal:

Place query sequences on the backbone tree to optimize a criterion of interest

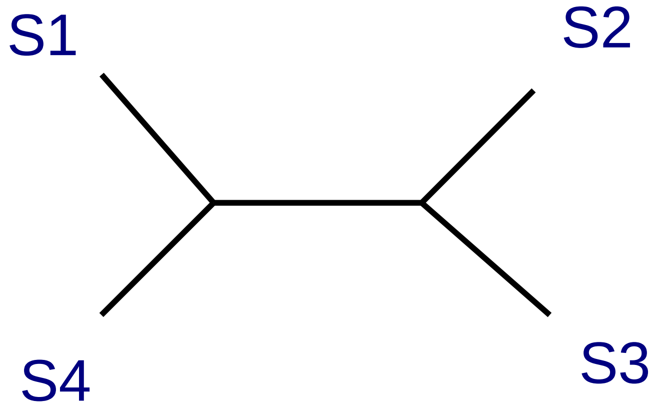
Input

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



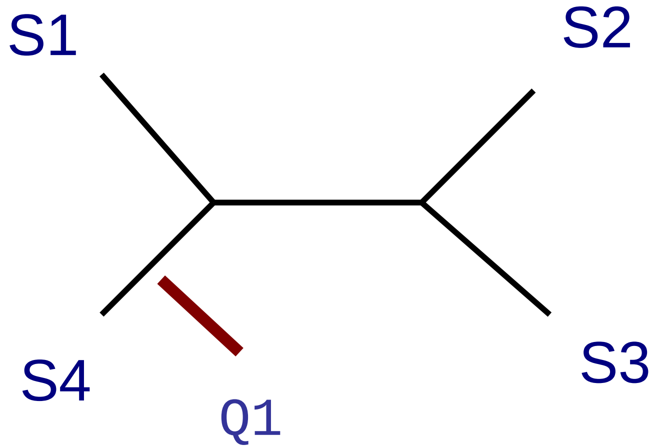
Align Query Sequence

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



Place Sequence

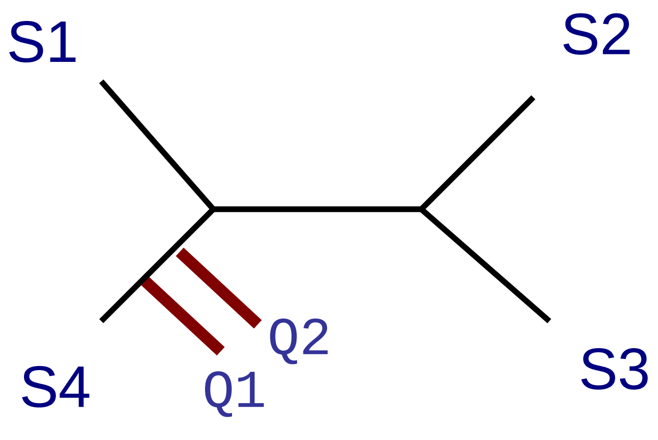
```
S1  = -AGGCTATCACCTGACCTCCA-AA
S2  = TAG-CTATCAC--GACCGC--GCA
S3  = TAG-CT-------GACCGC--GCT
S4  = TAC----TCAC--GACCGACAGCT
Q1  = -------T-A--AAAC--------
```



Phylogenetic Placement

- Addition of each sequence is independent of the other sequences
  - Thus, running time is linear in the number of query sequences

- The relation between added sequences is not inferred



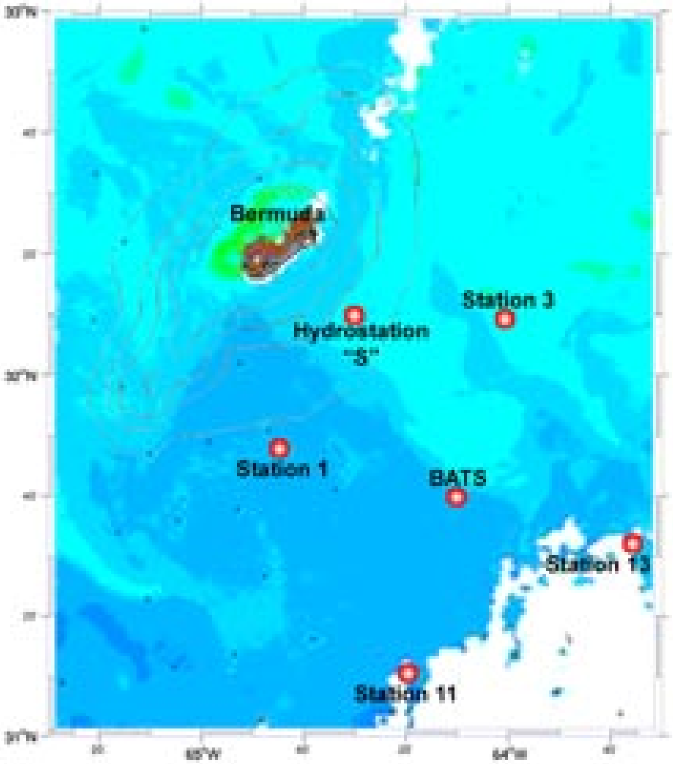
Applications of Phylogenetic Placement

- Starting trees for search algorithms
- Rogue Taxa Detection
- Contamination Detection
- **Metagenomics**

Metagenomics:

Venter et al., Exploring the Sargasso Sea:

Scientists Discover One Million New Genes in Ocean Microbes



Metagenomic data analysis

Direct Sampling from environment

Metagenomic analyses using NGS sequencing technology results in unknown species an short fragmentary reads

Taxon identification: given short sequences, identify the species for each fragment

Applications: Human Microbiome

Issues: accuracy and speed

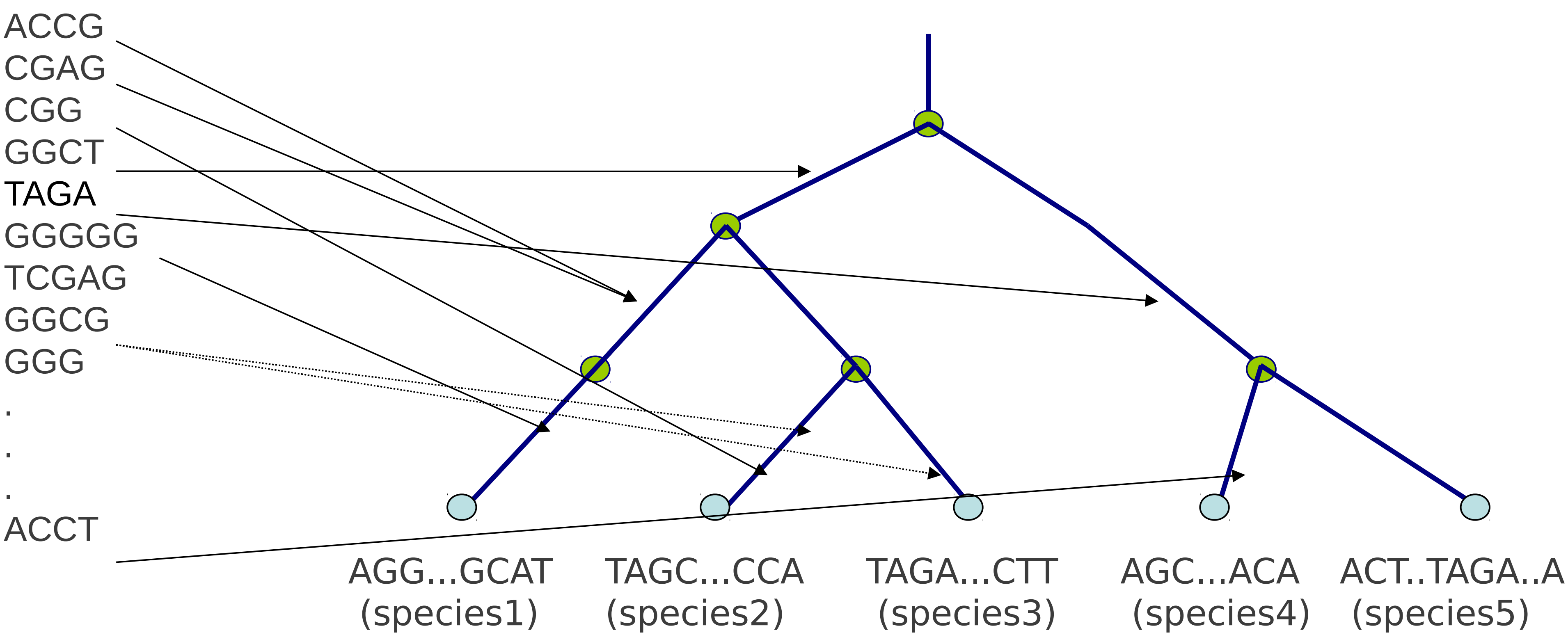
Phylogenetic Placement

Fragmentary Unknown Reads:

(60-200 bp long)

Known Full length Sequences,
a *reference* alignment and tree

(500-10,000 bp long)



Phylogenetic Placement

Align each query sequence to backbone alignment

- HMMER: using Hiden Markov Models
- PaPaRa

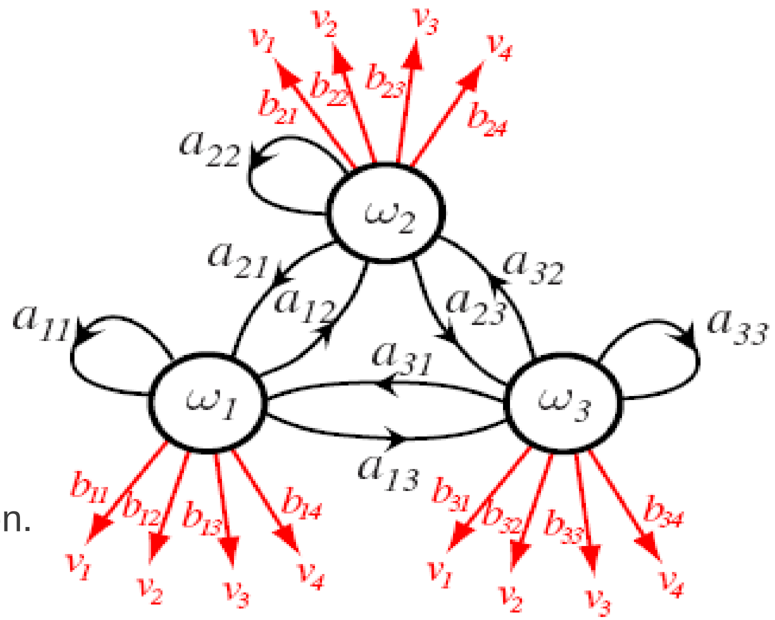
Place each query sequence into backbone tree, using extended alignment

- pplacer: Maximum Likelihood

Hidden Markov Models

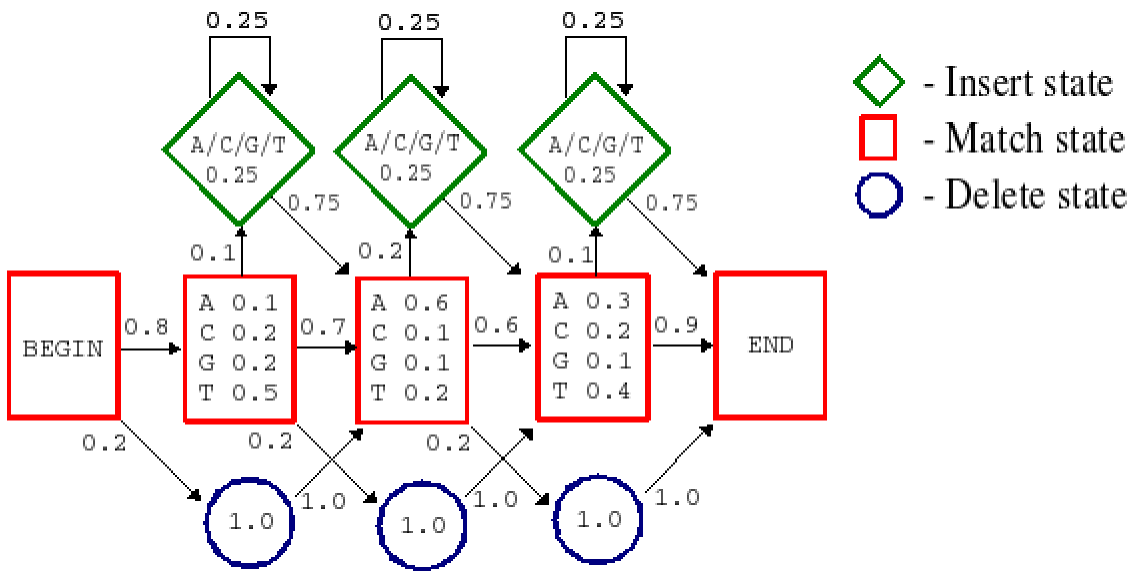
- Probabilistic modeling of processes that typically produce a sequence of observations. Examples: speech, DNA
- A state transition system
- Markov Property: the state of the process at step t only depends on step t-1
- State transitions are “hidden”
- Each state emits an

  observable output



Richard O. Duda, Peter E. Hart, and David G. Stork, Pattern Classification.

HMM Example: DNA Sequence

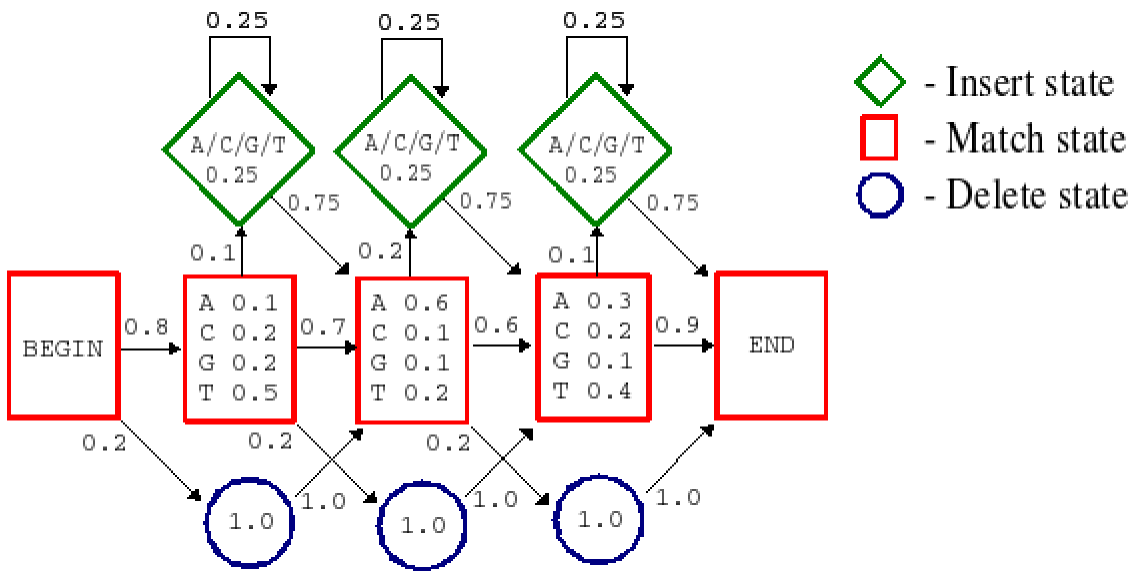


http://www.bioinfo.ifm.liu.se/edu/TFTB29/HT2012/assignment3.html

- AAA.A..
- C.A.A..
- C.-CA..
- G.A.C..
- T.ATG..
- G.C.CCC
- T.A.-..
- -.T.T..
- T.-.-..
- T.G.T..
- T.T.T..
- -.A.T..

- **Problem 1:** given a model and observed data, find the probability of a observed data
- **Problem 2:** given a model and observed data, find the most likely state transition
- **Problem 3:** given a set of observations, build a model that best explains the data

HMM Example: DNA Sequence



http://www.bioinfo.ifm.liu.se/edu/TFTB29/HT2012/assignment3.html

- AAA.A..
- C.A.A..
- C.-CA..
- G.A.C..
- T.ATG..
- G.C.CCC
- T.A.-..
- -.T.T..
- T.-.-..
- T.G.T..
- T.T.T..
- -.A.T..

- **Problem 1:** Find the probability that a sequence is related to another set (e.g. a gene)
- **Problem 2:** Align a new sequence to a set of aligned sequences, presented as a HMM
- **Problem 3:** Represent a set of aligned sequences as a HMM

Phylogenetic Placement

Align each query sequence to backbone alignment

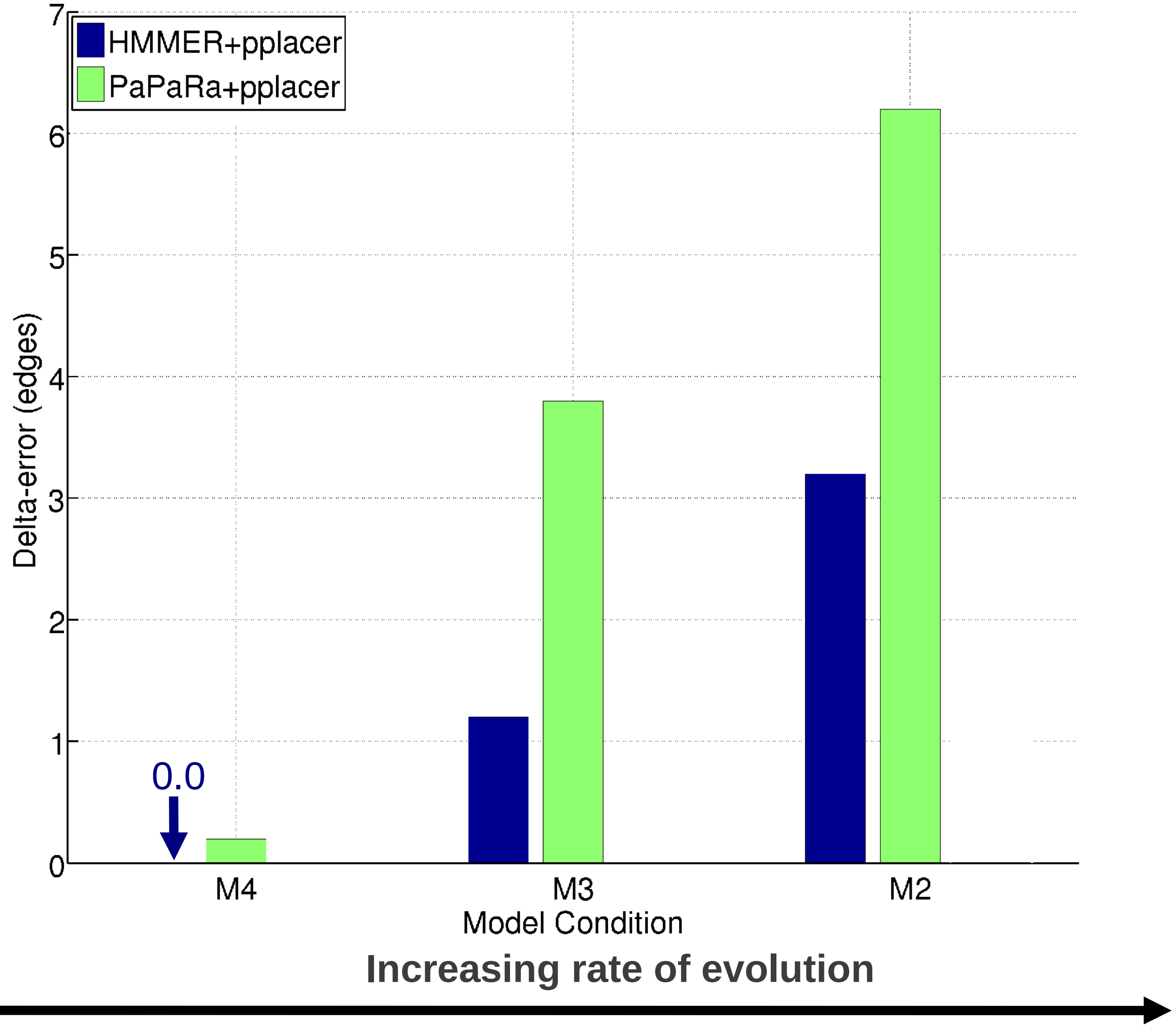
- HMMER: using Hiden Markov Models

- PaPaRa

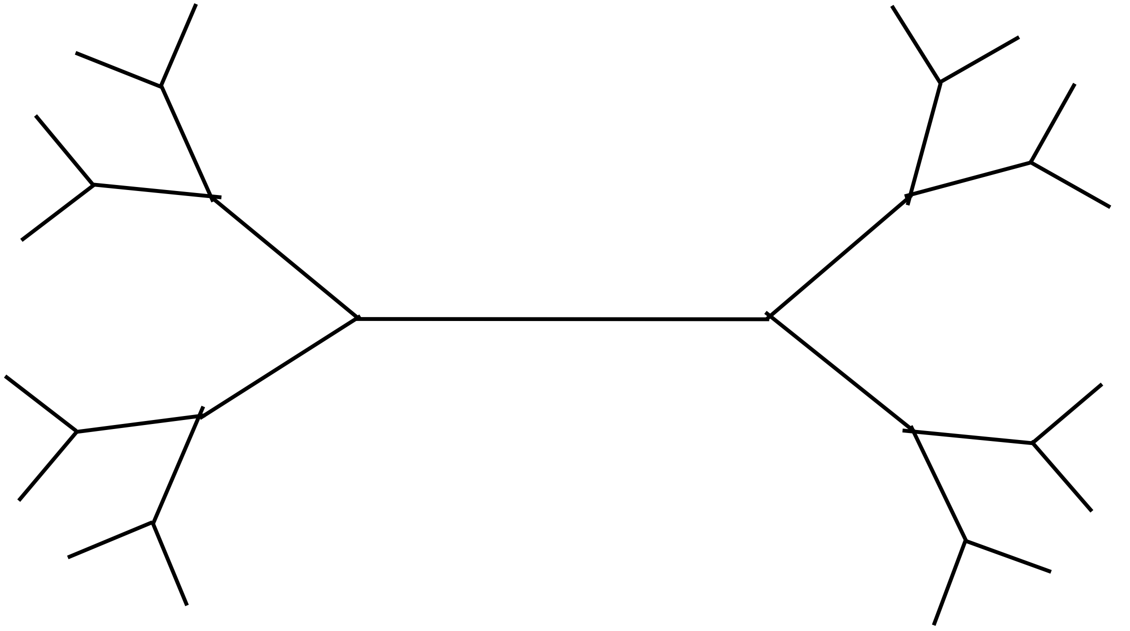
Place each query sequence into backbone tree, using extended alignment

- pplacer: Maximum Likelihood

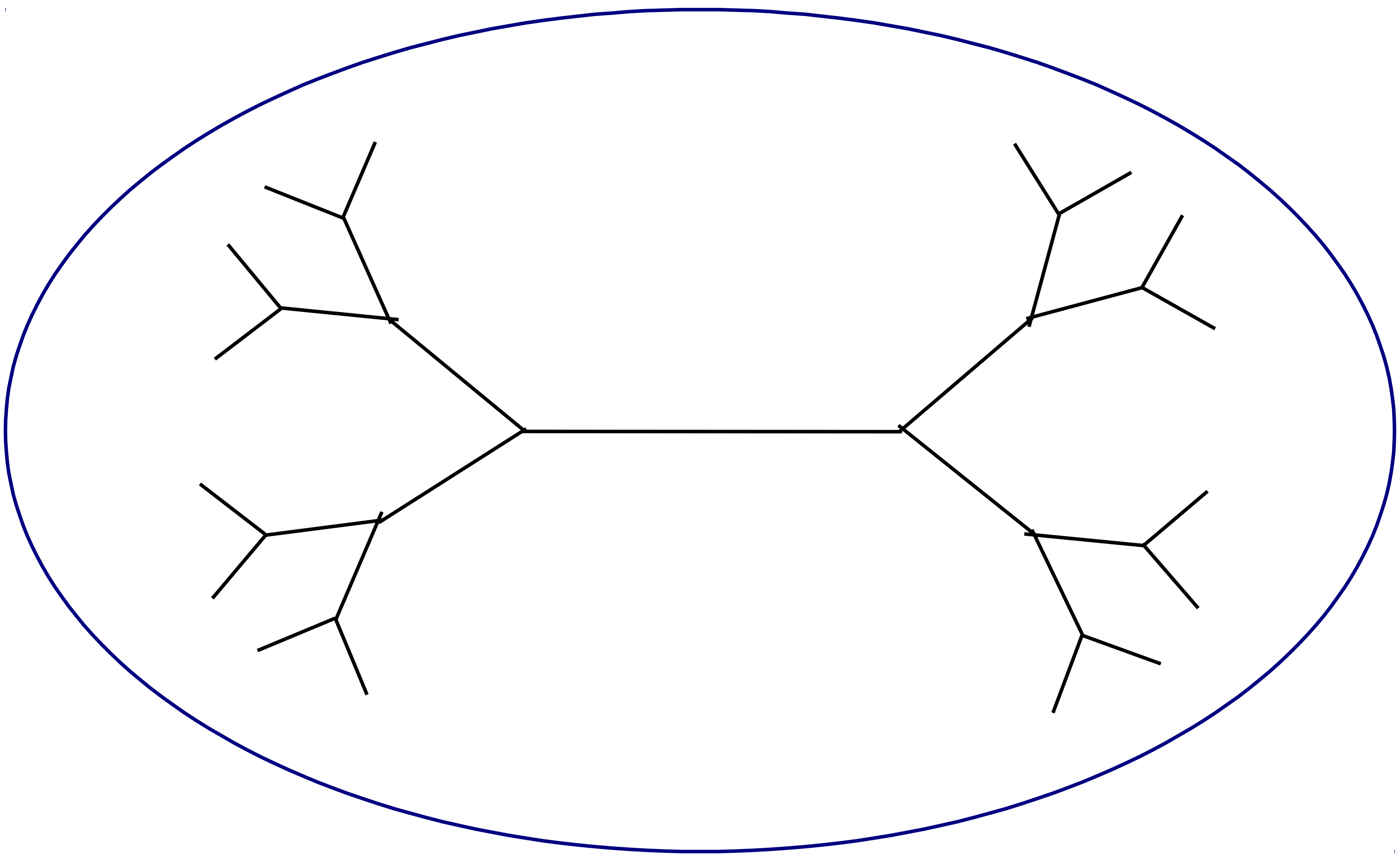
Performance of Existing Tools



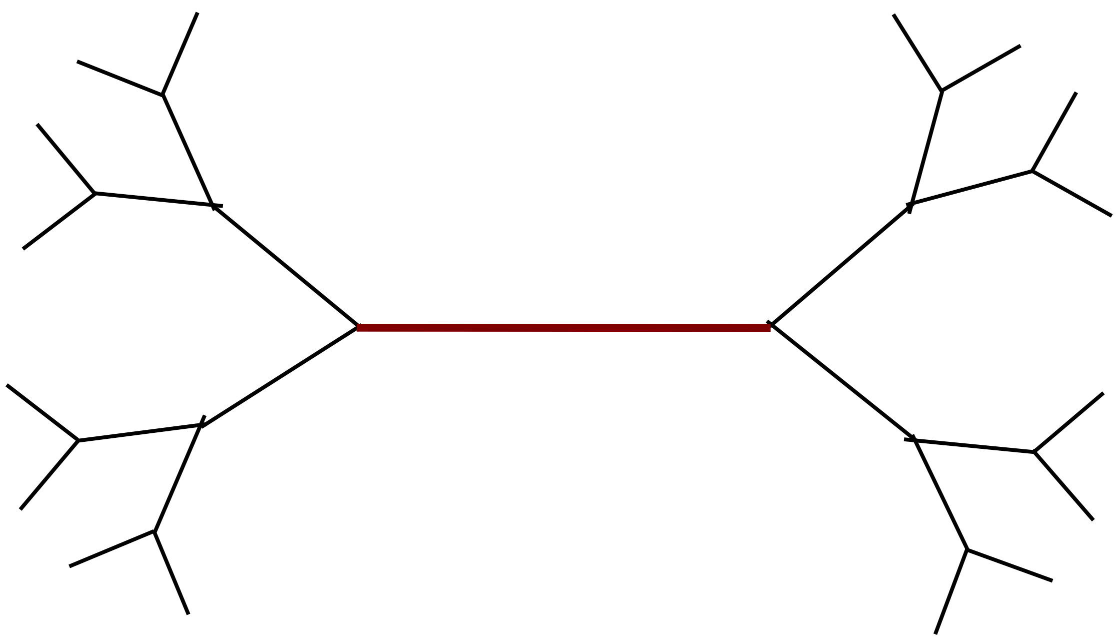
Insight



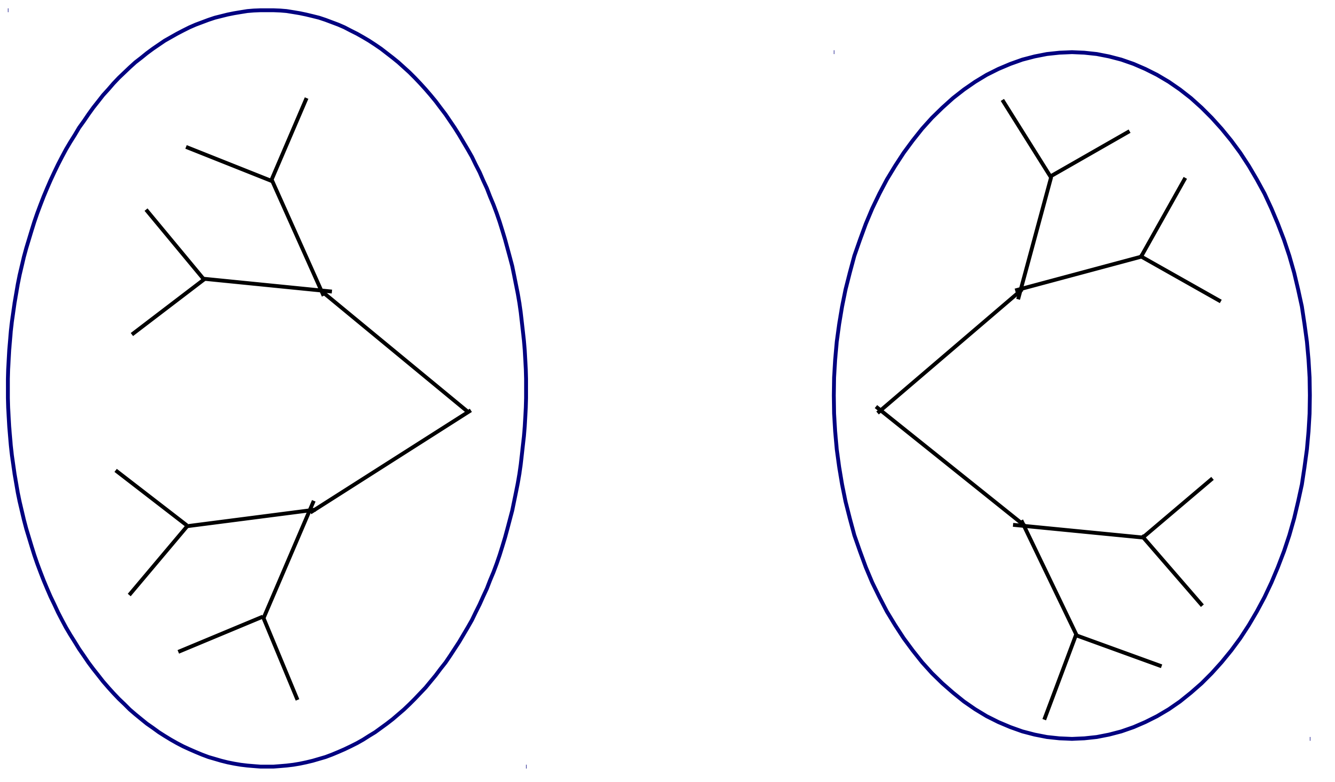
Insight



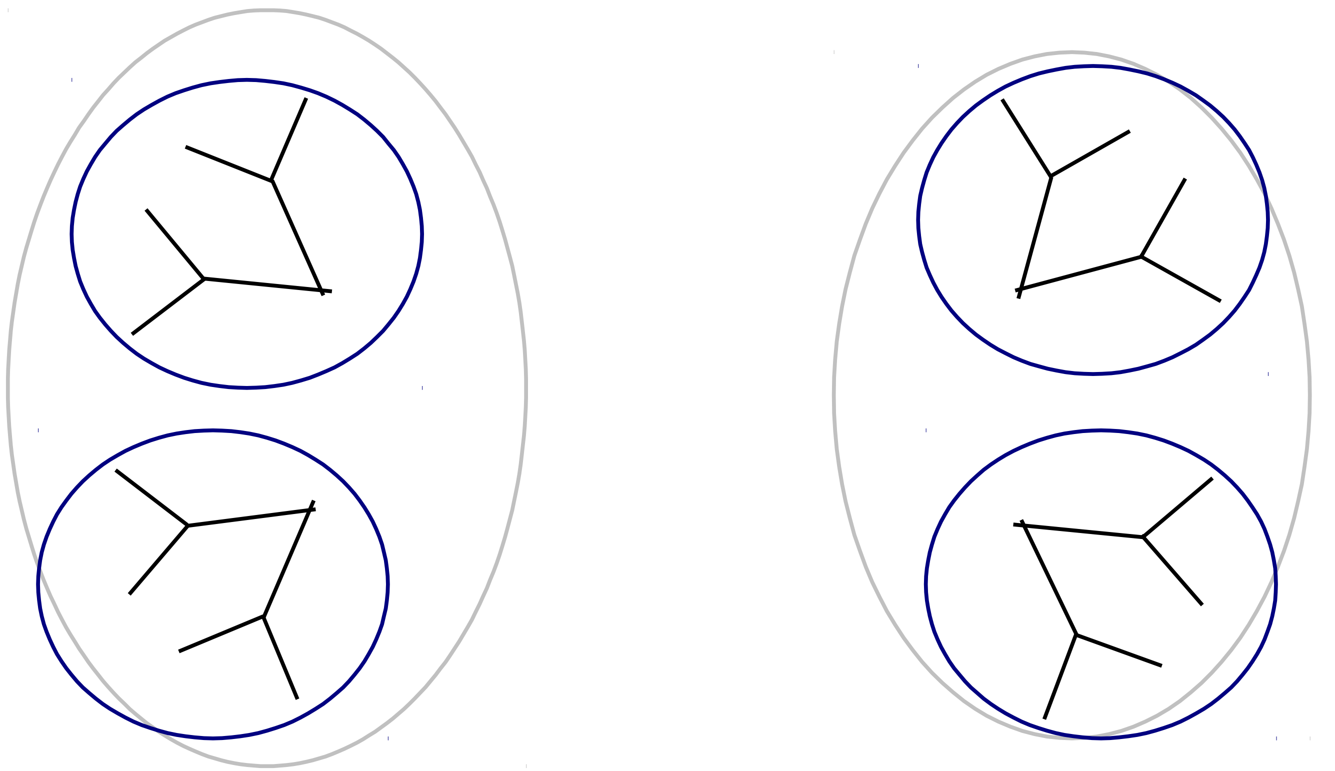
Insight



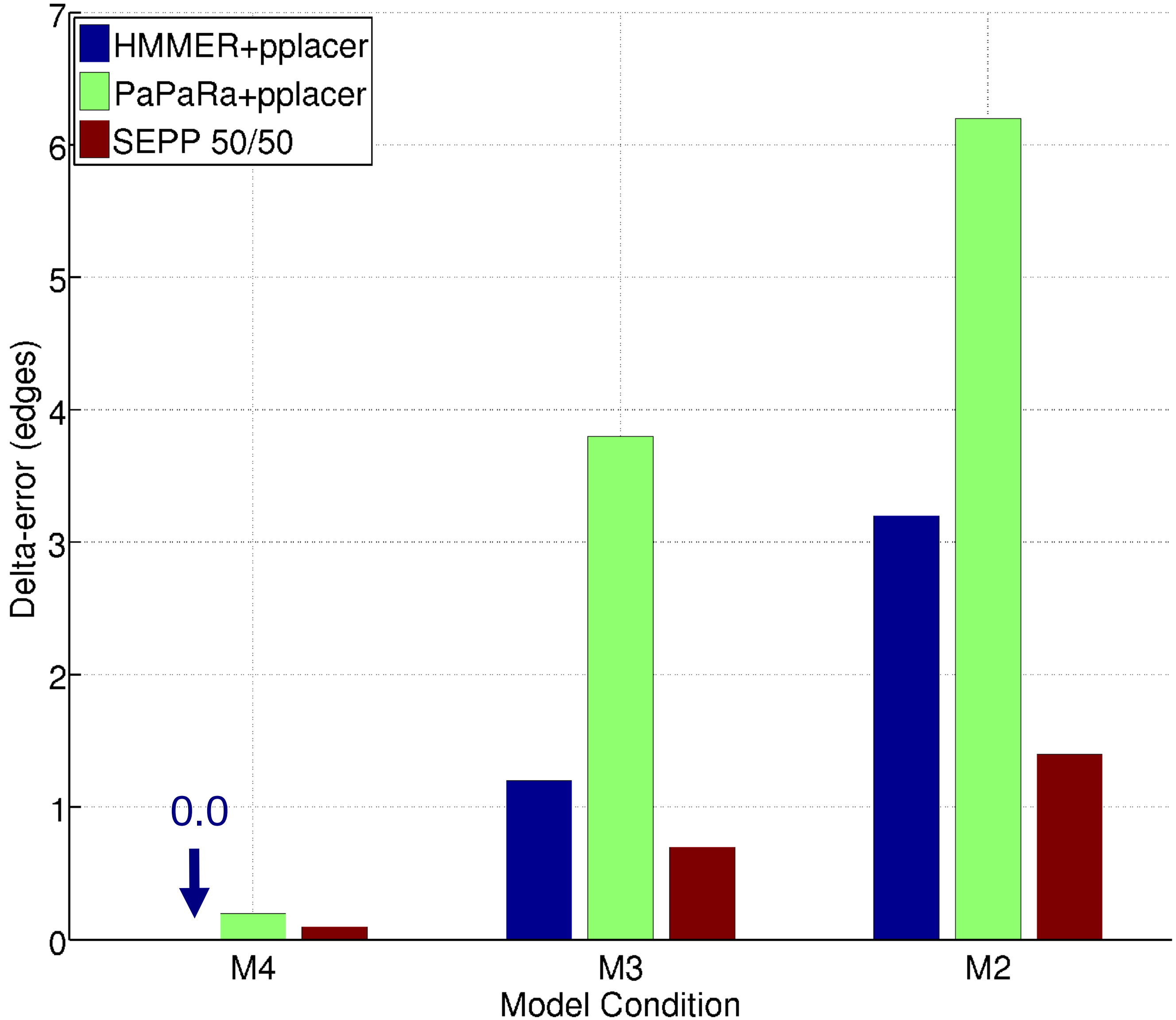
Insight



Insight

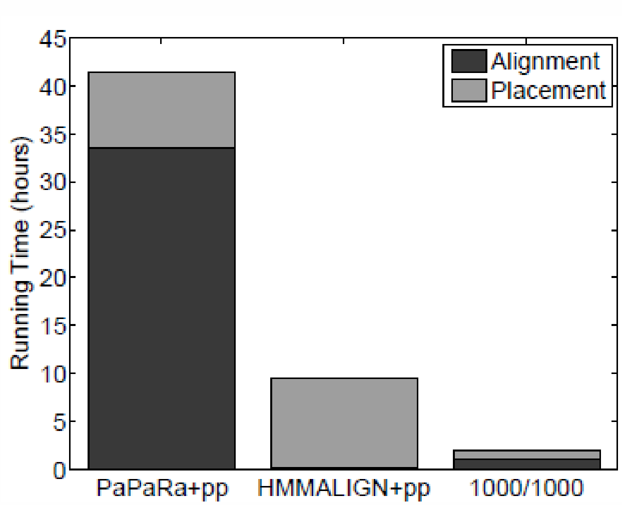


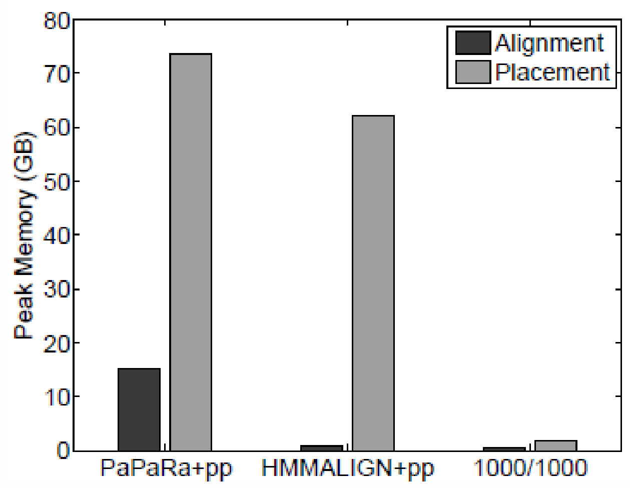
SEPP (10%-rule) on simulated data

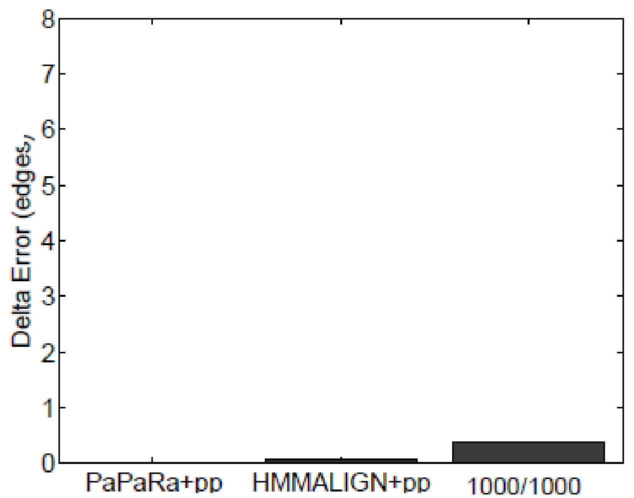


Increasing rate of evolution

SEPP on Biological Data







16S.B.ALL dataset, 13k curated backbone tree, 13k total fragments

For 1 million fragments:

PaPaRa+pplacer: ~133 days

HMMALIGN+pplacer: ~30 days

SEPP 1000/1000: ~6 days

Part II: UPP
(Ultra-large alignment using SEPP[1])

Objective: highly accurate multiple sequence alignments and trees on ultra-large datasets

Authors: Nam Nguyen, Siavash Mirarab, and Tandy Warnow

In preparation – expected submission Fall 2013

[1] SEPP: SATe-enabled phylogenetic placement, Nguyen, Mirarab, and Warnow, PSB 2012

UPP: basic idea

Input: set S of unaligned sequences

Output: alignment on S

- Select random subset X of S
- Estimate “backbone” alignment A and tree T on X
- Independently align each sequence in S-X to A
- Use transitivity to produce multiple sequence alignment A* for entire set S

Input: Unaligned Sequences

```
S1  = AGGCTATCACCTGACCTCCAAT
S2  = TAGCTATCACGACCGCGCT
S3  = TAGCTGACCGCGCT
S4  = TACTCACGACCGACAGCT
S5  = TAGGTACAACCTAGATC
S6  = AGATACGTCGACATATC
```

Step 1: Pick random subset (backbone)

```
S1   = AGGCTATCACCTGACCTCCAAT
S2   = TAGCTATCACGACCGCGCT
S3   = TAGCTGACCGCGCT
S4   = TACTCACGACCGACAGCT
S5   = TAGGTACAACCTAGATC
S6   = AGATACGTCGACATATC
```

Step 2: Compute backbone alignment

```
S1  = -AGGCTATCACCTGACCTCCA-AT
S2  = TAG-CTATCAC--GACCGC--GCT
S3  = TAG-CT-------GACCGC--GCT
S4  = TAC----TCAC—-GACCGACAGCT
S5  = TAGGTAAAACCTAGATC
S6  = AGATAAAACTACATATC
```

Step 3: Align each remaining sequence to backbone

First we add S5 to the backbone alignment

```
S1  = -AGGCTATCACCTGACCTCCA-AT-
S2  = TAG-CTATCAC--GACCGC--GCT-
S3  = TAG-CT-------GACCGC—-GCT-
S4  = TAC----TCAC--GACCGACAGCT-
S5  = TAGG---T-A—CAA-CCTA--GATC
```

Step 3: Align each remaining sequence to backbone

Then we add S6 to the backbone alignment

```
S1  = -AGGCTATCACCTGACCTCCA-AT-
S2  = TAG-CTATCAC--GACCGC--GCT-
S3  = TAG-CT-------GACCGC--GCT-
S4  = TAC----TCAC—-GACCGACAGCT-
S6  = -AG---AT-A-CGTC--GACATATC
```

Step 4: Use transitivity to obtain MSA on entire set

```
S1  = -AGGCTATCACCTGACCTCCA-AT--
S2  = TAG-CTATCAC--GACCGC--GCT--
S3  = TAG-CT-------GACCGC--GCT--
S4  = TAC----TCAC--GACCGACAGCT--
S5  = TAGG---T-A—CAA-CCTA--GATC-
S6  = -AG---AT-A-CGTC--GACATAT-C
```

UPP: details

Input: set S of unaligned sequences

Output: alignment on S

- Select random subset X of S
- Estimate “backbone” alignment A and tree T on X
- Independently align each sequence in S-X to A
- Use transitivity to produce multiple sequence alignment A* for entire set S

How to align sequences to a backbone alignment?

Standard machine learning technique: Build HMM (Hidden Markov Model) for backbone alignment, and use it to align remaining sequences

HMMER (Sean Eddy, HHMI) leading software for this purpose

Using HMMER

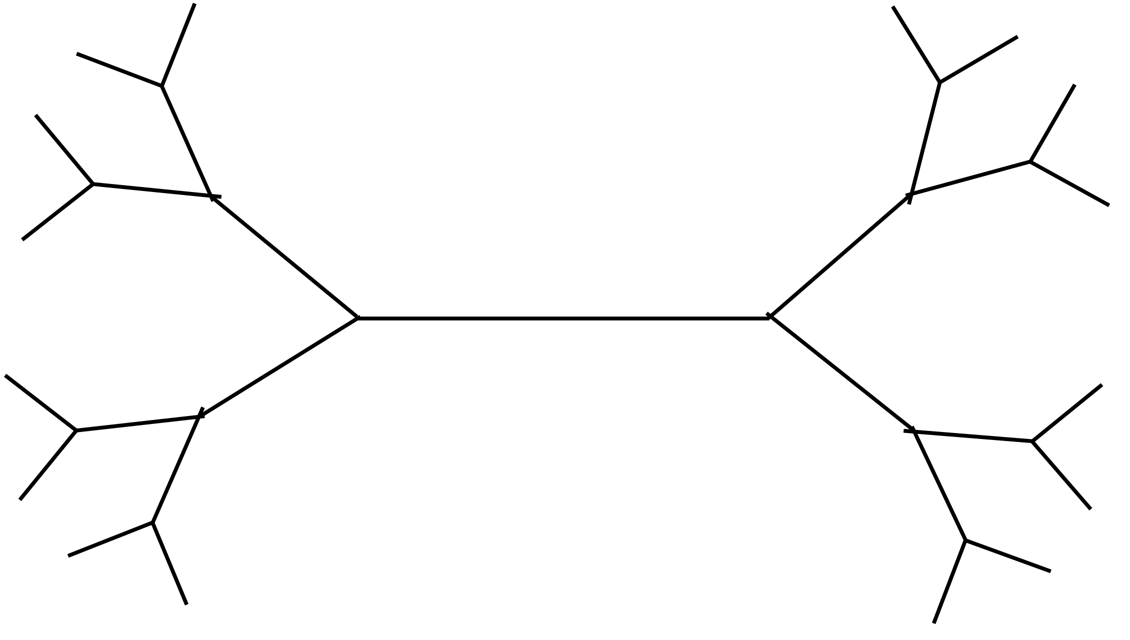
Using HMMER works well…

Using HMMER

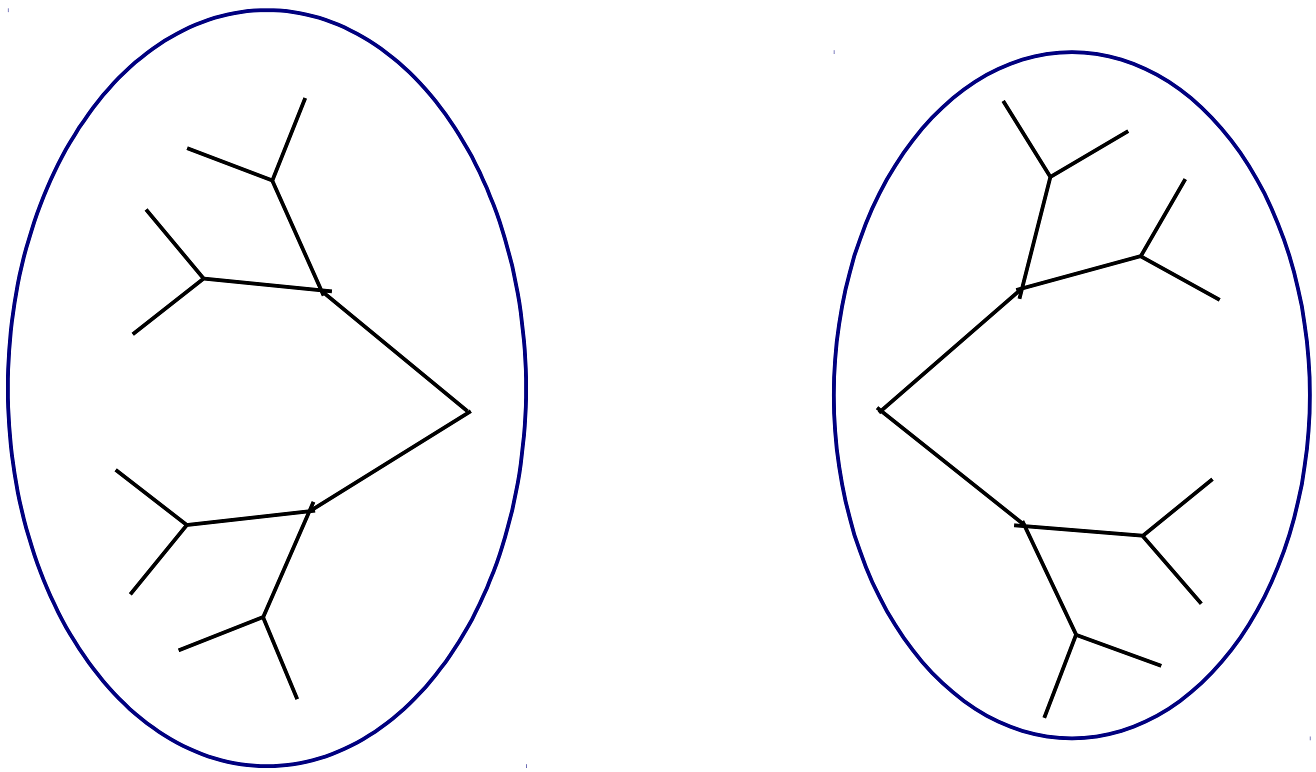
Using HMMER works well…except when the dataset is big!

Using HMMER to add sequences to an existing alignment

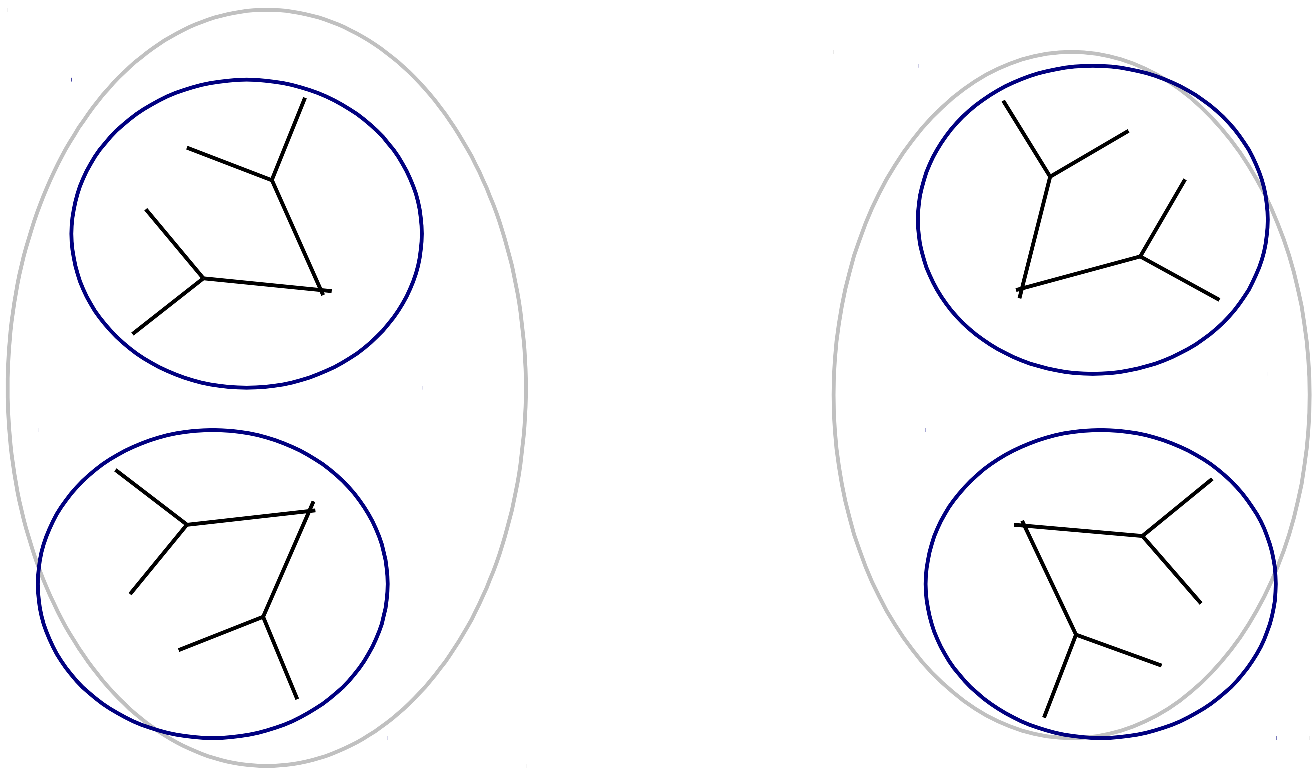
1) build one HMM for the backbone alignment
2) Align sequences to the HMM, and insert into backbone alignment



Or 2 HMMs?



Or 4 HMMs?



UPP(x,y)

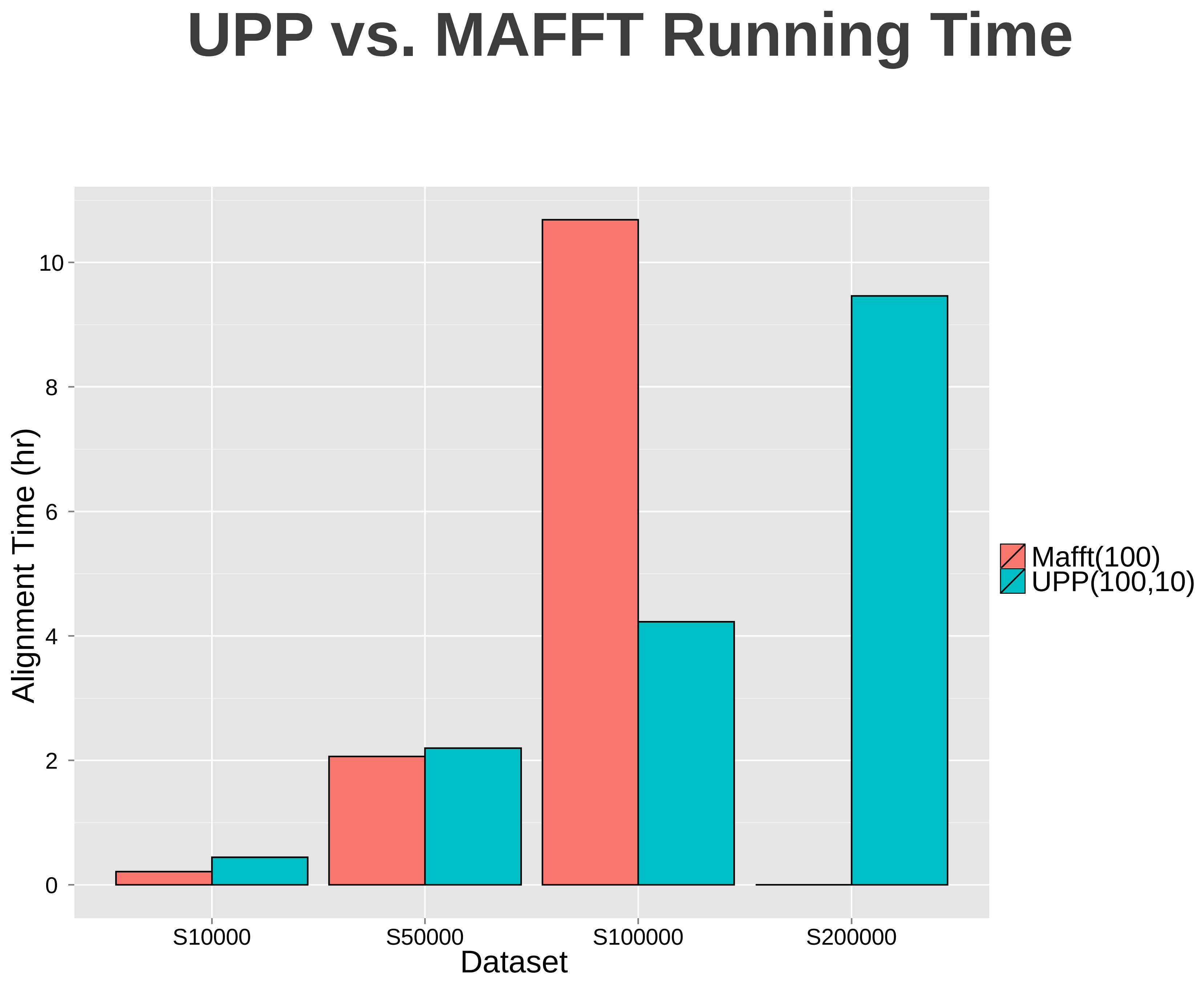
- Pick random subset X of size **x**
- Compute alignment A and tree T on X
- Use SATé decomposition on T to partition X into small “alignment subsets” of at most **y** sequences
- Build HMM on each alignment subset using HMMBUILD
- For each sequence s in S-X,
  - Use HMMALIGN to produce alignment of s to each subset alignment and note the score of each alignment.
  - Pick the subset alignment that has the best score, and align s to that subset alignment.
  - Use transitivity to align s to the backbone alignment.

UPP design

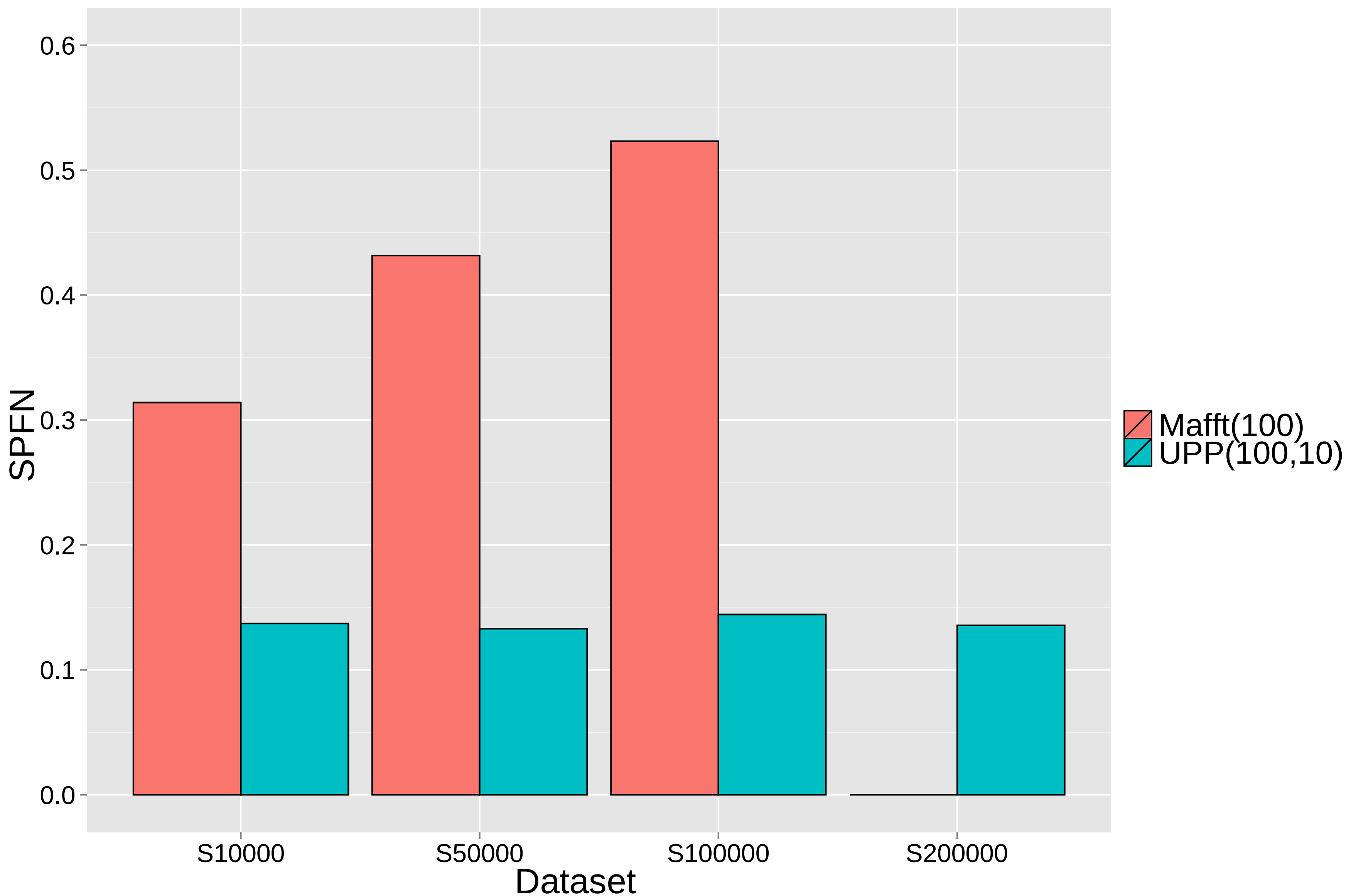
- Size of backbone matters – small backbones are sufficient for most datasets (except for ones with very high rates of evolution). Random backbones are fine.
- Number of HMMs matters, and depends on the rate of evolution and number of taxa.
- Backbone alignment and tree matter; we use SATé.

Evaluation of UPP

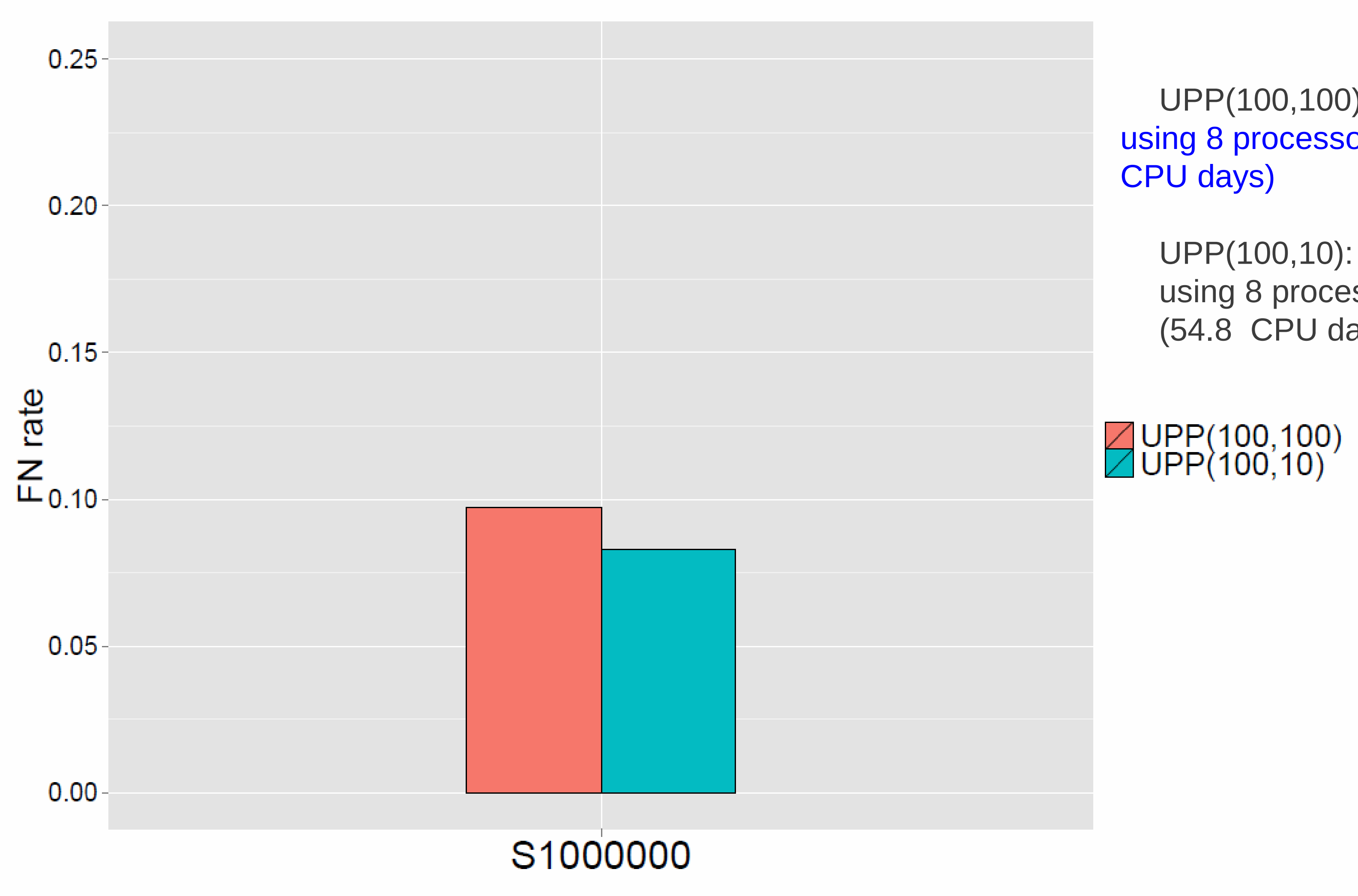
- Simulated Datasets: 1,000 to 1,000,000 sequences (RNASim, Junhyong Kim Penn)
- Biological datasets with reference alignments (Gutell's CRW data with up to 28,000 sequences)
- Criteria: Alignment error (SP-FN and SP-FP), tree error, and time


UPP vs. MAFFT Running Time
Alignment Time (hr)
0
2
4
6
8
10
S10000
S50000
S100000
S200000
Dataset
Mafft(100)
UPP(100,10)

UPP vs. MAFFT Alignment Error



One Million Sequences: Tree Error



UPP(100,100): 1.6 days using 8 processors (5.7 CPU days)

UPP(100,10): 7 days using 8 processors (54.8 CPU days)

Note improvement obtained by using UPP decomposition

UPP performance

- Speed: UPP is very fast, parallelizable, and scalable.
- UPP vs. standard MSA methods: UPP is more accurate on large datasets (with 1000+ taxa), and trees on UPP alignments are more accurate than trees on standard alignments.
- UPP vs. SATé: UPP is much faster and can analyze much larger datasets; UPP has about the same alignment accuracy, but produces slightly less accurate trees.

More Fundamental Questions

- Data partitioning for model estimation;

  Trade-off between:
  - Larger number of more specific models estimated based on less data
  - Fewer models, each less specific, but each estimated based on more data
- Related to a host of theoretical issues, such as
  - model fit
  - Information content
- Can Decomposition be incorporated into the model?

Conclusion

- It can pay off to decompose your observations into subsets and building models on these subsets
  - Decomposition needs to make each subset more homogeneous
  - The search problem morphs into n searches
- Iterative addition of sequences to a backbone is a useful strategy, if done with care