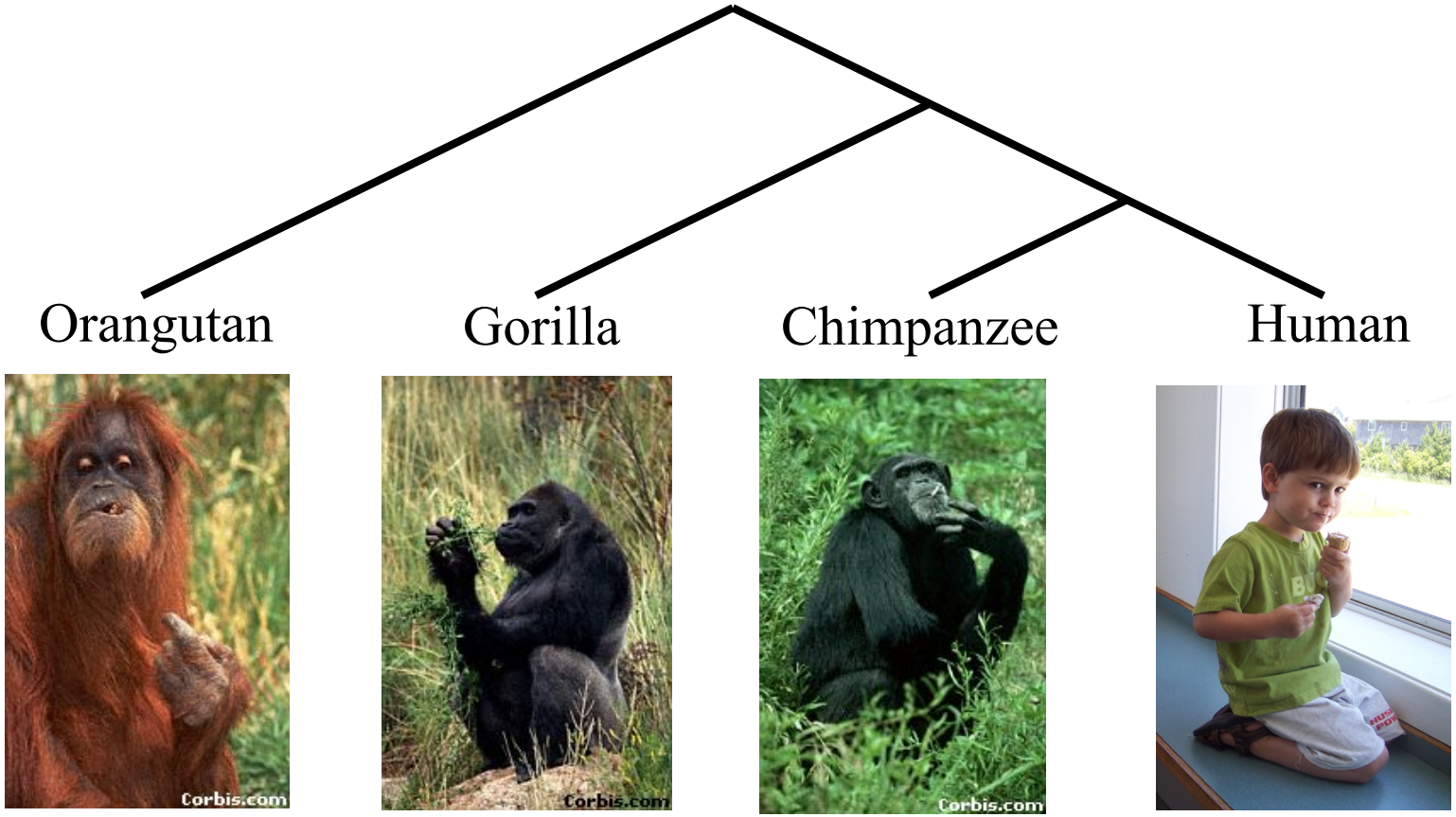
Recent Breakthroughs in Mathematical and Computational Phylogenetics

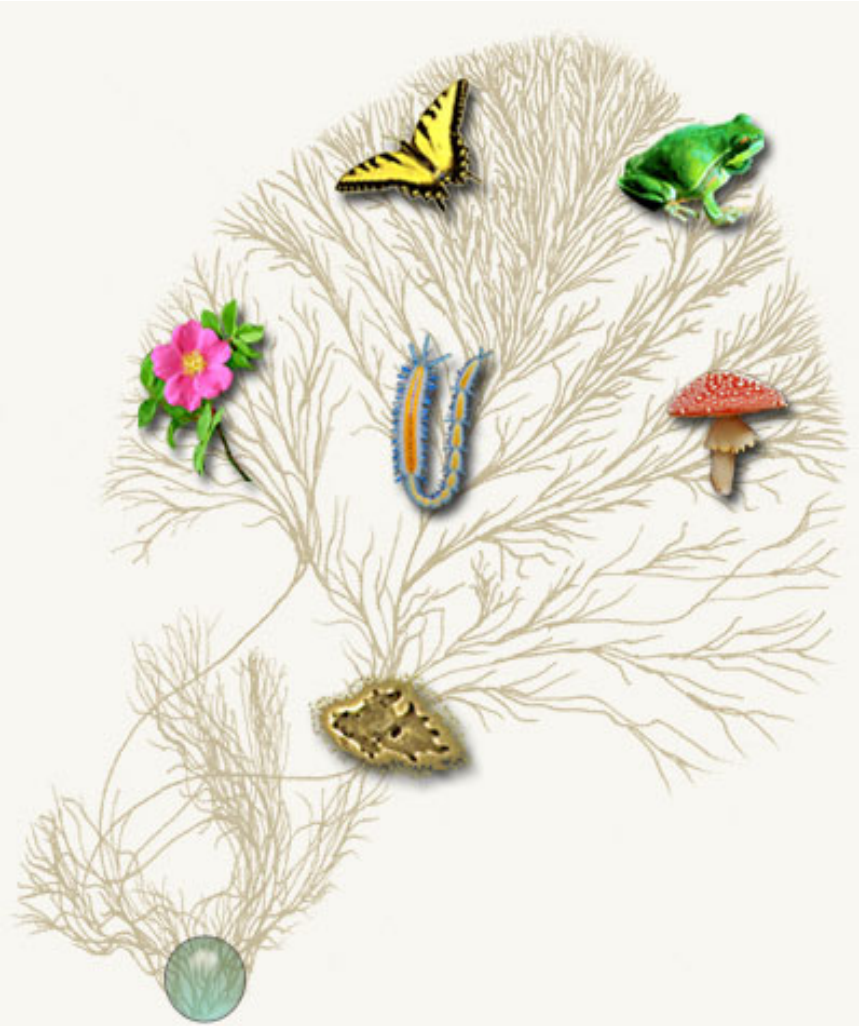
Tandy Warnow
Department of Computer Science
The University of Texas at Austin

Phylogeny (evolutionary tree)



From the Tree of the Life Website,
University of Arizona

How did life evolve on earth?



- **Courtesy of the Tree of Life project**

An international effort to understand how life evolved on earth

Biomedical applications: drug design, protein structure and function prediction, biodiversity.

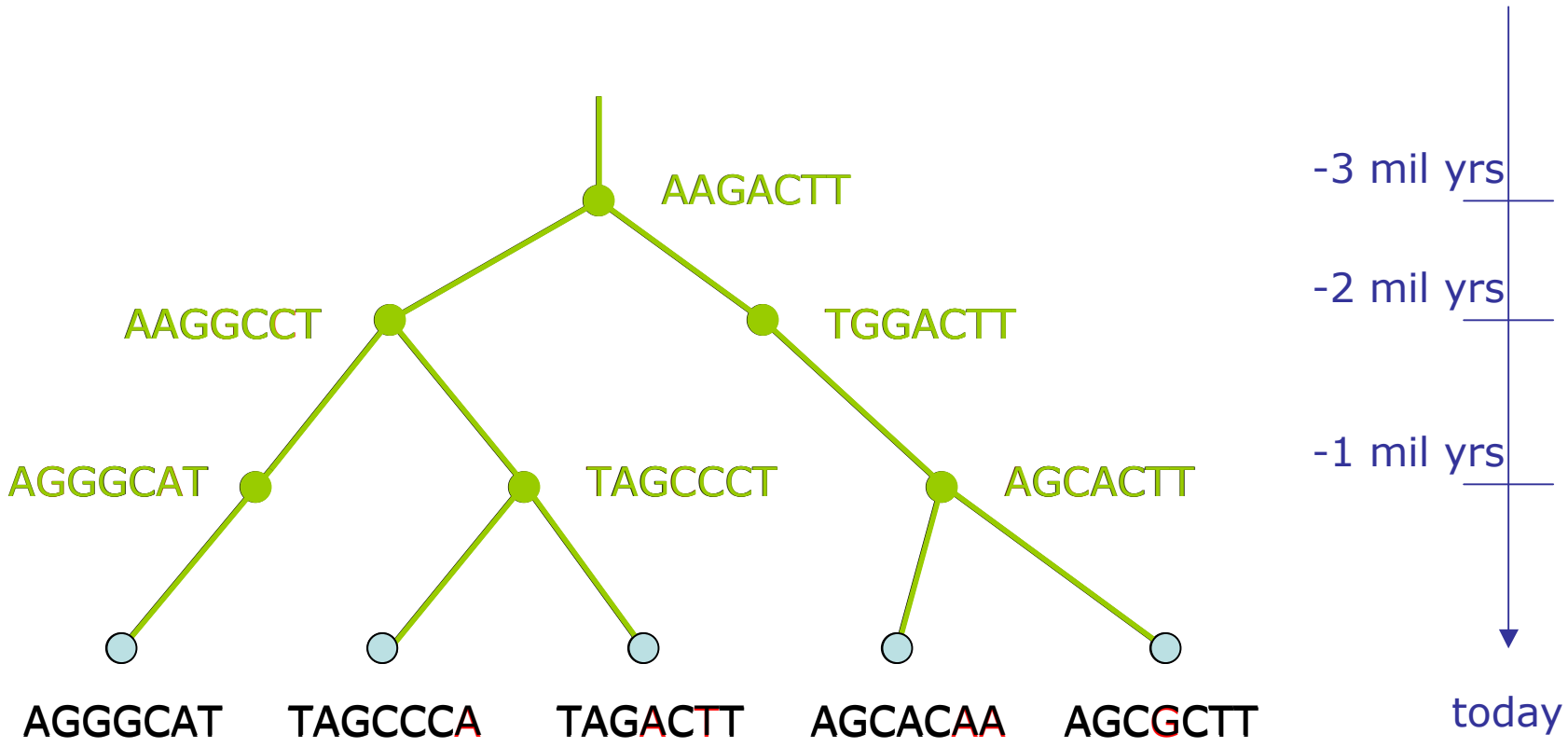
Today's talk:

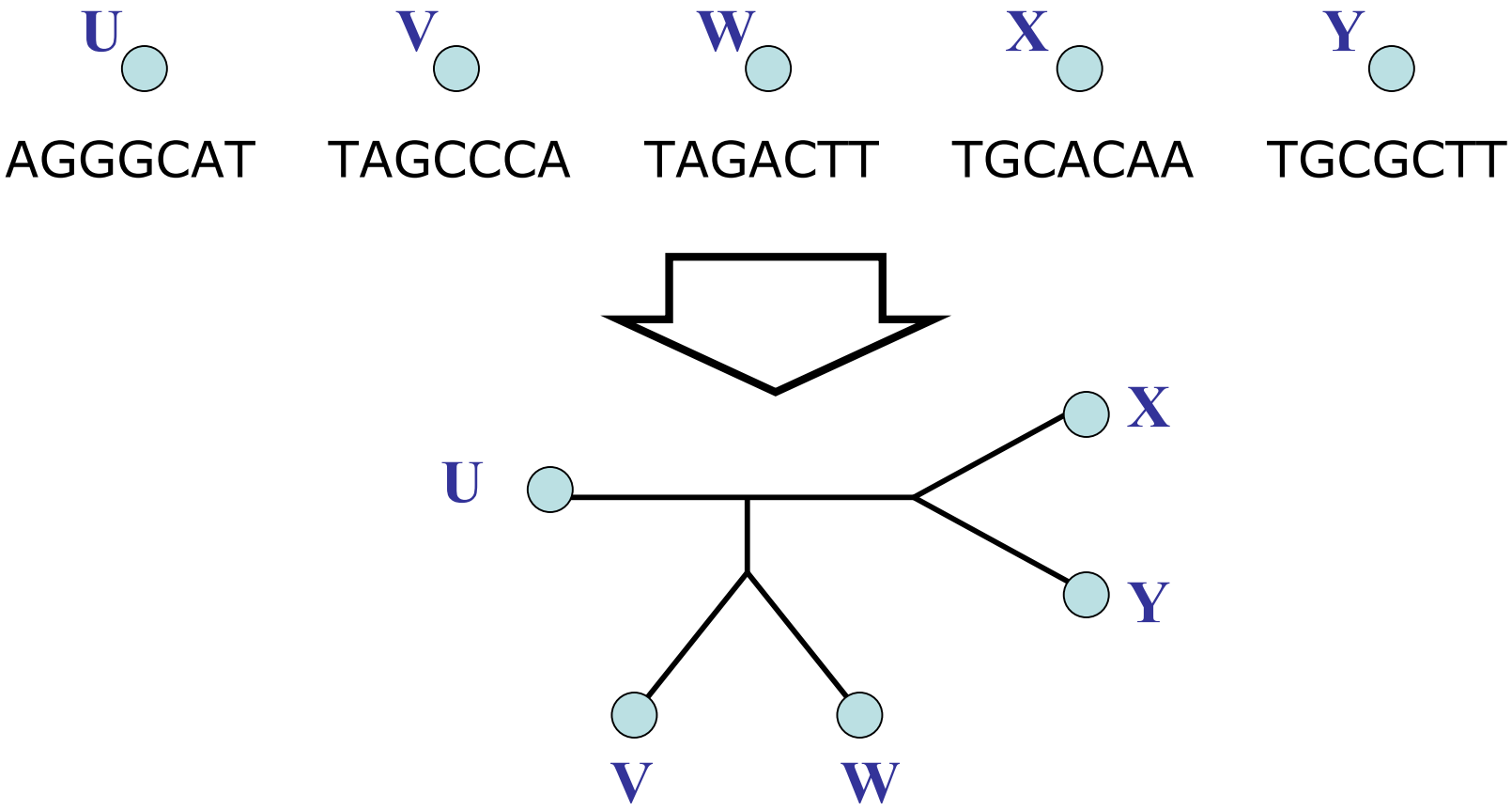
some theory, some empirical performance

- When true alignment is known: methods that are absolute fast-converging (1997 to present)
- Estimating trees in the presence of insertions and deletions:
  - SATé: Liu et al., Science 2009, and Systematic Biology, in press), and
  - DACTAL: Nelesen et al., in preparation

Part 1: Absolute Fast Convergence

DNA Sequence Evolution




U
V
W
X
Y
AGGGCAT
TAGCCCA
TAGACTT
TGCACAA
TGCGCTT
X
U
Y
V
W

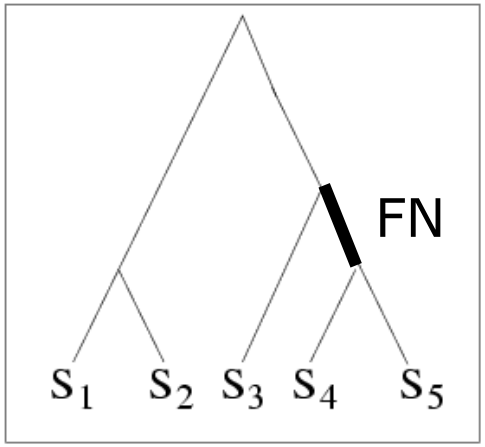
Markov Model of Site Evolution

Simplest (Jukes-Cantor):

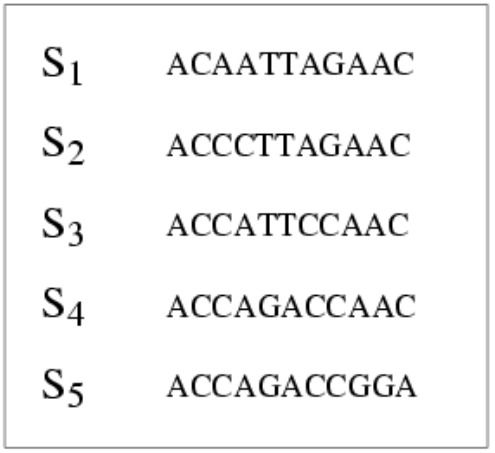
- The model tree T is binary and has substitution probabilities p(e) on each edge e.
- The state at the root is randomly drawn from {A,C,T,G} (nucleotides)
- If a site (position) changes on an edge, it changes with equal probability to each of the remaining states.
- The evolutionary process is Markovian.

More complex models (such as the General Markov model) are also considered, often with little change to the theory.

Quantifying Error



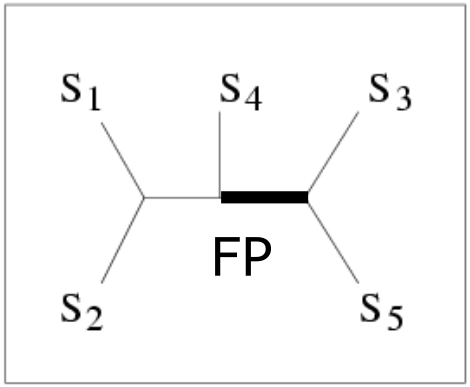
TRUE TREE



DNA SEQUENCES

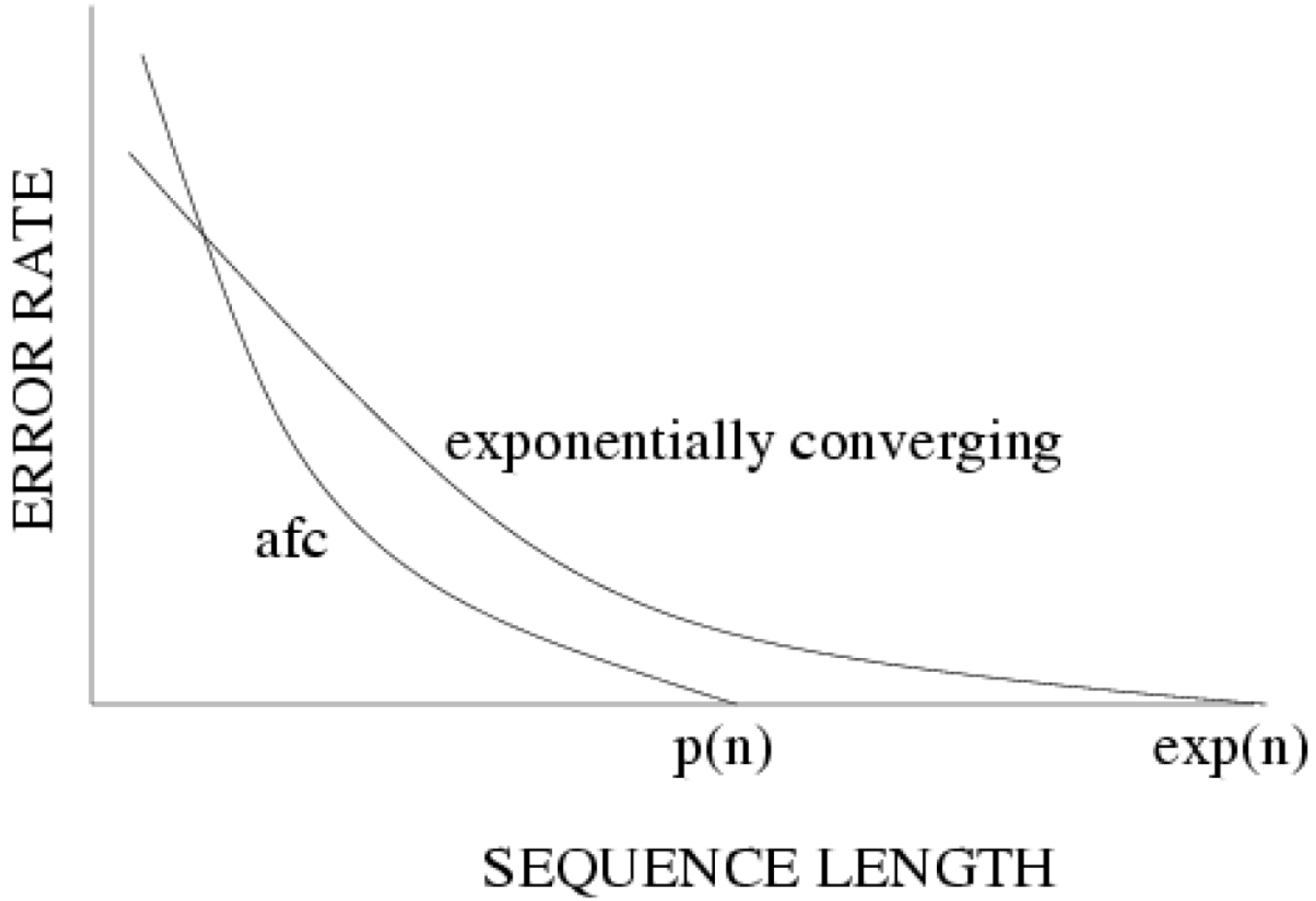
FN: false negative
(missing edge)
FP: false positive
(incorrect edge)

50% error rate



INFERRED TREE

Statistical consistency, exponential convergence, and absolute fast convergence (afc)



"Convergence rate" or sequence length requirement

The sequence length (number of sites) that a phylogeny reconstruction method M needs to reconstruct the true tree with probability at least $1-\varepsilon$ depends on

- M (the method)
- ε
- f = min p(e),
- g = max p(e), and
- n, the number of leaves

We fix everything but **n**.

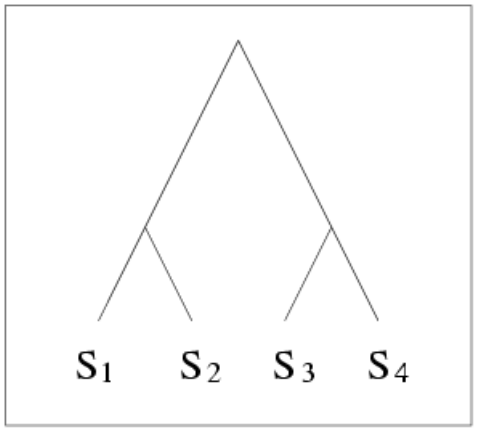
Afc methods

A method M is “absolute fast converging”, or afc, if for all positive f, g, and ε, there is a polynomial p(n) s.t. Pr(M(S)=T) > 1- ε, when S is a set of sequences generated on T of length at least p(n).

Notes:

1. The polynomial p(n) will depend upon M, f, g, and ε.
2. The method M is not “told” the values of f and g.

Distance-based estimation



TRUE TREE

S_1	ACAATTAGAAC
S_2	ACCCTTAGAAC
S_3	ACCATTCCAAC
S_4	ACCAGACCAAC

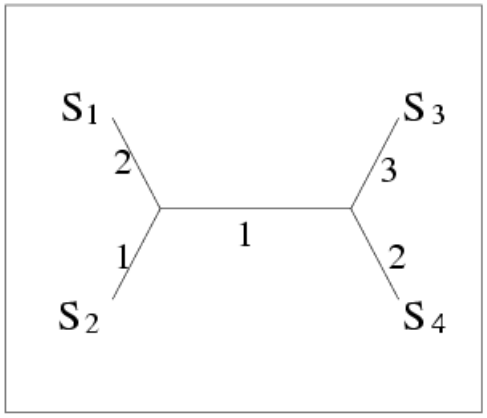
DNA SEQUENCES

STATISTICAL ESTIMATION OF PAIRWISE DISTANCES

	S_1	S_2	S_3	S_4
S_1	0	3	6	5
S_2		0	5	4
S_3			0	5
S_4				0

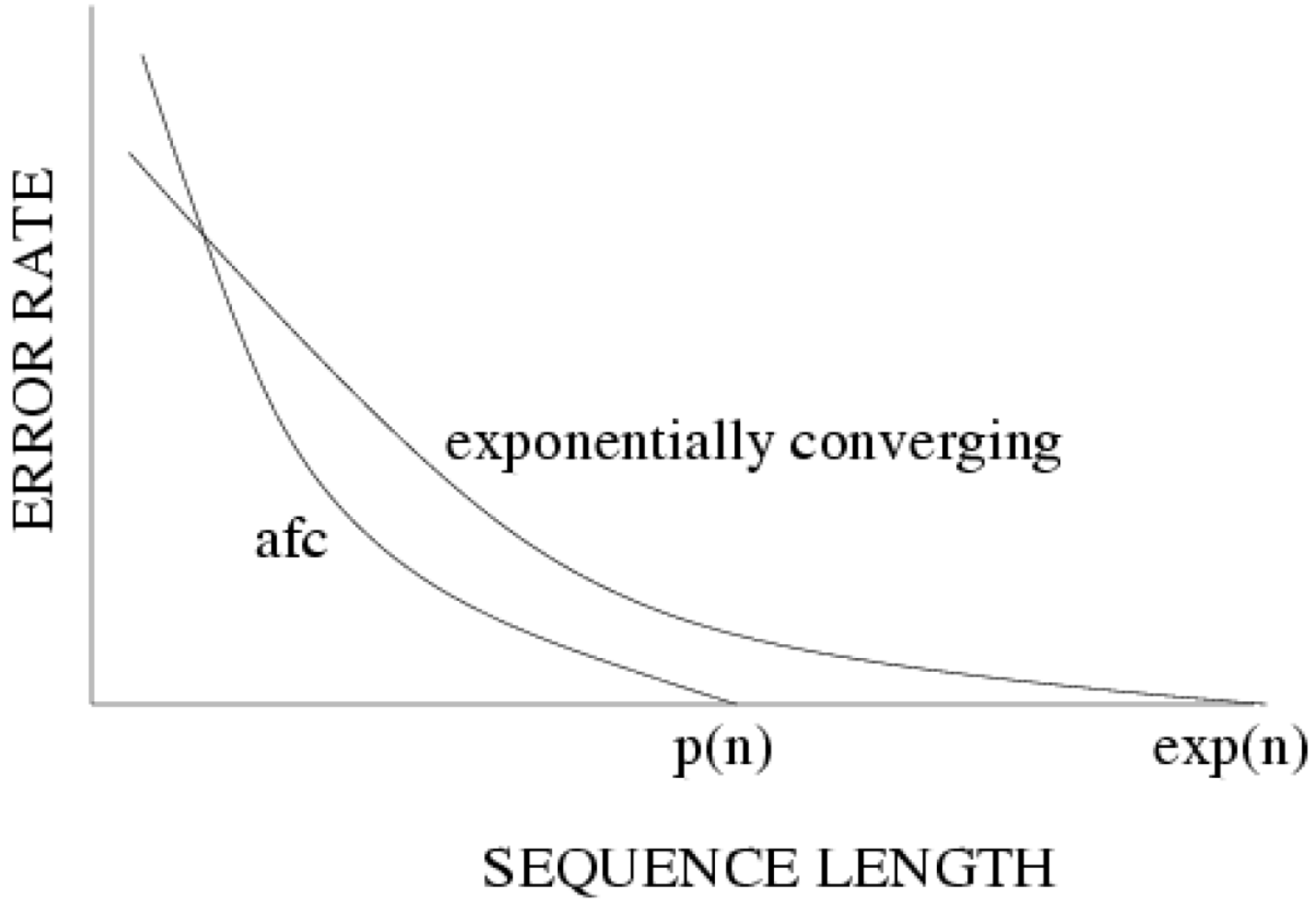
DISTANCE MATRIX

METHODS SUCH AS NEIGHBOR JOINING



INFERRED TREE

Are distance-based methods statistically consistent?
And if so, what are their sequence length requirements?

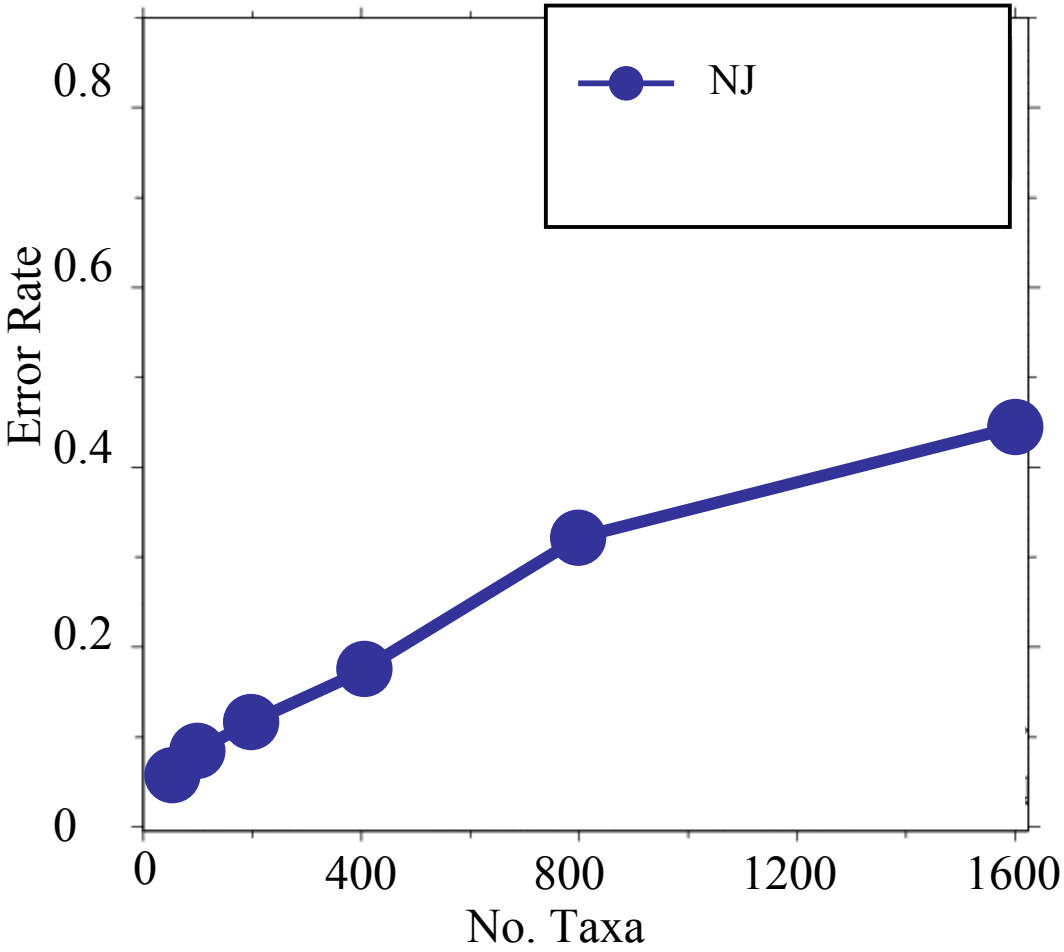


Theorem (Erdos et al., Atteson): Neighbor joining (and some other methods) will return the true tree w.h.p. provided sequence lengths are exponential in the evolutionary diameter of the tree.

Sketch of proof:

- NJ (and other distance methods) guaranteed correct if *all* entries in the estimated distance matrix have sufficiently low error.
- Estimations of large distances require long sequences to have low error w.h.p.

Performance on large diameter trees



Simulation study based upon fixed edge lengths, K2P model of evolution, sequence lengths fixed to 1000 nucleotides.

Error rates reflect proportion of incorrect edges in inferred trees.

[Nakhleh et al. ISMB 2001]

Designing an afc method

- You often don’t need the entire distance matrix to get the true tree (think of the caterpillar tree)

Designing an afc method

- You often don’t need the entire distance matrix to get the true tree (think of the caterpillar tree)
- The problem is you don’t know which entries have sufficiently low error, and which ones are needed to determine the tree.

Designing an afc method

- You often don’t need the entire distance matrix to get the true tree (think of the caterpillar tree)
- The problem is you don’t know which entries have sufficiently low error, and which ones are needed to determine the tree.
- But you can guess!

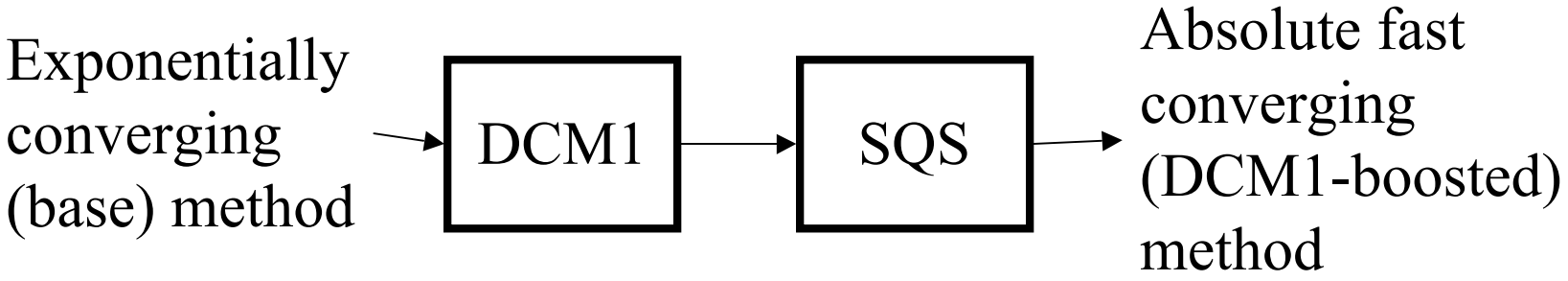
Fast converging methods (and related work)

- 1997: Erdos, Steel, Szekely, and Warnow (ICALP).
- 1999: Erdos, Steel, Szekely, and Warnow (RSA, TCS); Huson, Nettles and Warnow (J. Comp Bio.)
- 2001: Warnow, St. John, and Moret (SODA); Cryan, Goldberg, and Goldberg (SICOMP); Csuros and Kao (SODA); Nakhleh, St. John, Roshan, Sun, and Warnow (ISMB)
- 2002: Csuros (J. Comp. Bio.)
- 2006: Daskalakis, Mossel, Roch (STOC), Daskalakis, Hill, Jaffe, Mihaescu, Mossel, and Rao (RECOMB)
- 2007: Mossel (IEEE TCBB)
- 2008: Gronau, Moran and Snir (SODA)
- 2010: Roch (Science)

and others

DCM1-boosting:

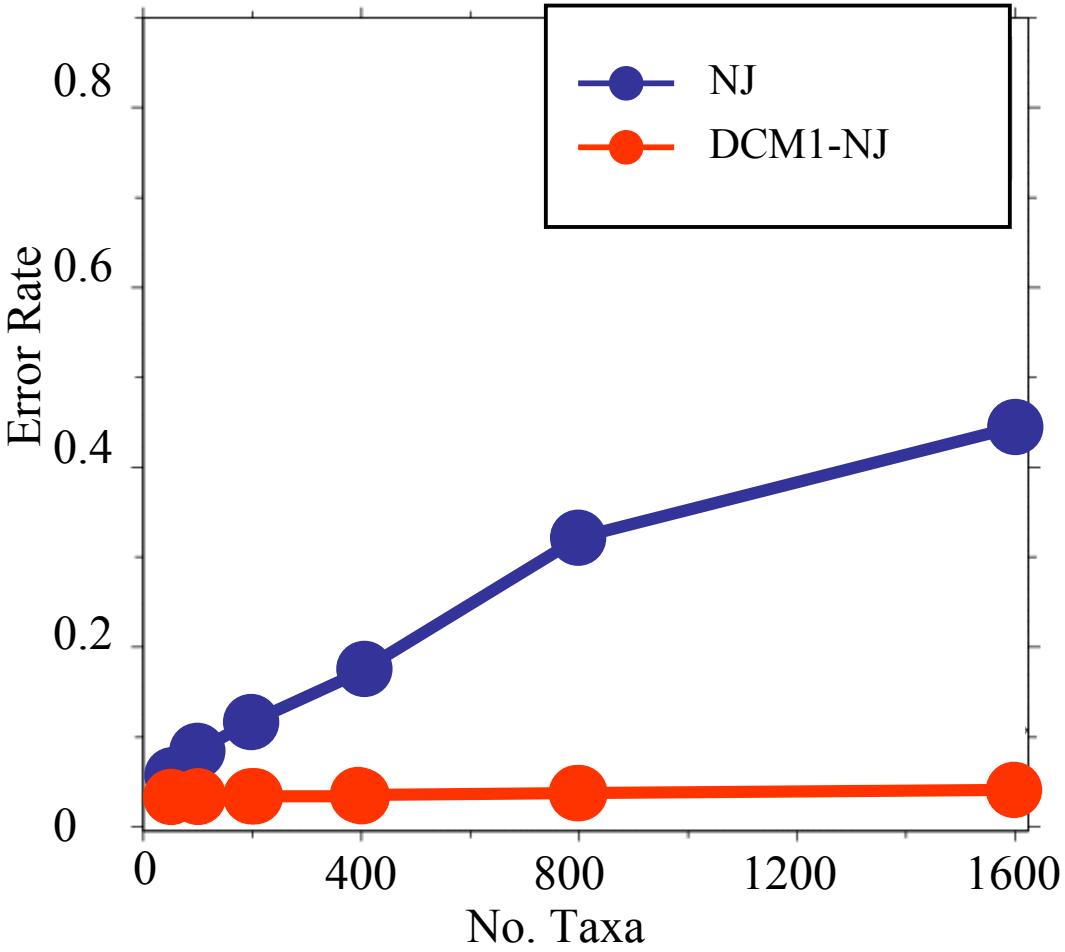
Warnow, St. John, and Moret, SODA 2001



- The DCM1 phase produces a collection of trees (one for each threshold), and the SQS phase picks the “best” tree.
- For a given threshold, the base method is used to construct trees on small subsets (defined by the threshold) of the taxa. These small trees are then combined into a tree on the full set of taxa.

DCM1-boosting distance-based methods

[Nakhleh et al. ISMB 2001]



Theorem (Warnow et al., SODA 2001): DCM1-NJ converges to the true tree from polynomial length sequences

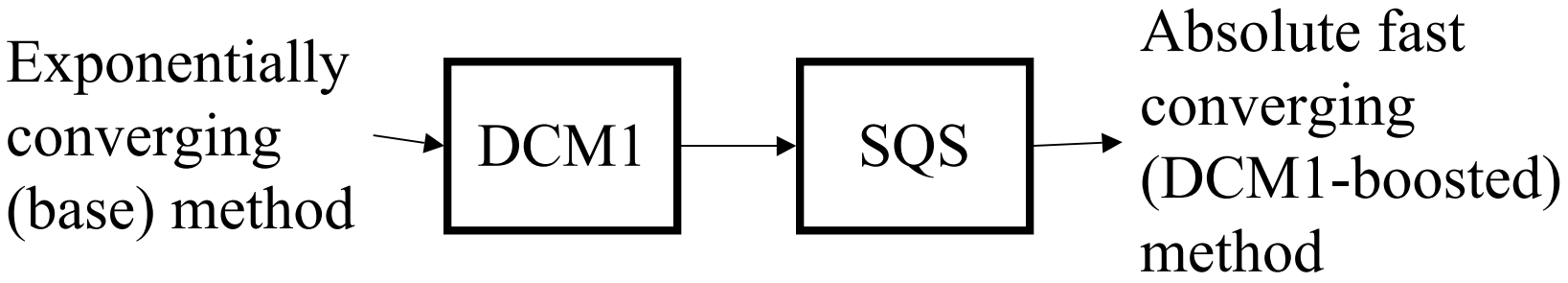
DCM1-NJ+SQS

- Theorem 1: For all f,g,ε, there is a polynomial p(n) such that given sequences of length at least p(n), then with probability at least 1- ε, the DCM1-phase produces a set containing the true tree.

- Theorem 2: For all f, g, ε, there is a polynomial p(n) such that given sequences of length at least p(n), then with probability at least 1- ε, if the set contains the true tree, then the SQS phase selects the true tree.

DCM1-boosting:

Warnow, St. John, and Moret, SODA 2001



- The DCM1 phase produces a collection of trees (one for each threshold), and the SQS phase picks the "best" tree.
- *How to compute a tree for a given threshold:*
  - *Handwaving description*: erase all the entries in the distance matrix above that threshold, and compute a tree from the remaining entries using the "base" method.
  - *The real technique* uses **chordal graph** decompositions.

Chordal (triangulated) graphs

- A graph is chordal iff it has no simple induced cycles of at least four vertices.

- Every chordal graph has at most n maximal cliques, and the *Maxclique* decomposition can be found in polynomial time.

DCM1

Given distance matrix for the species:

1. Define a triangulated (i.e. chordal) graph so that its vertices correspond to the input taxa

2. Compute the max clique decomposition of the graph, thus defining a decomposition of the taxa into overlapping subsets.

3. Compute tree on each max clique using the "base method".

4. Merge the subtrees into a single tree on the full set of taxa.

DCM1 Decompositions

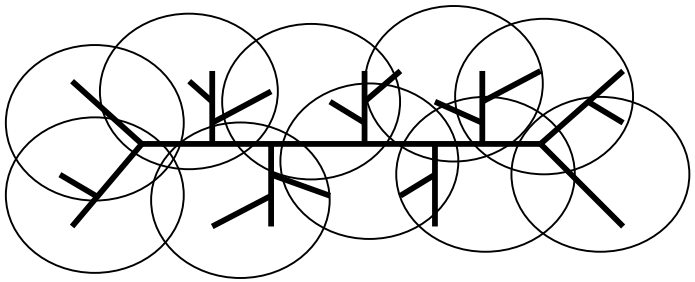
Input: Set S of sequences, distance matrix d, threshold value $q \in \{d_{ij}\}$

1. Compute threshold graph

$$G_q = (V, E), V = S, E = \{(i, j) : d(i, j) \leq q\}$$

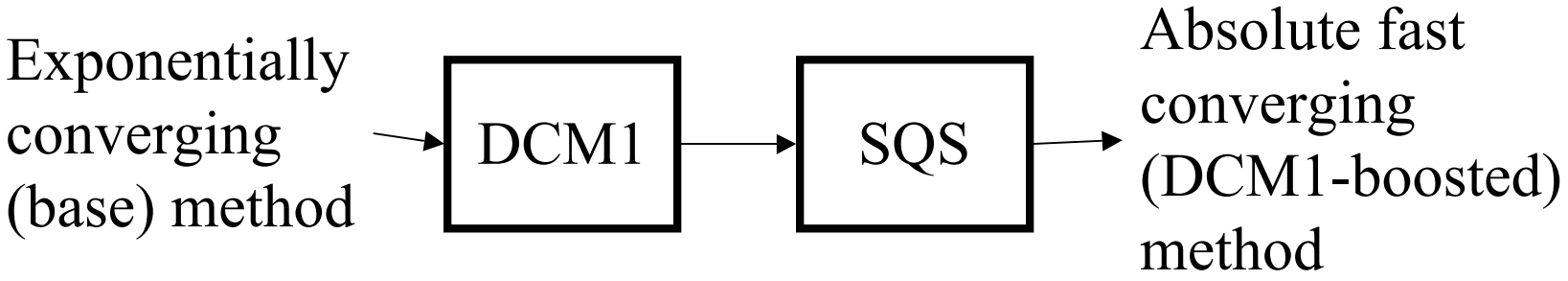
2. Perform minimum weight triangulation (note: if d is an additive matrix, then the threshold graph is provably triangulated).

DCM1 decomposition : Compute maximal cliques



DCM1-boosting:

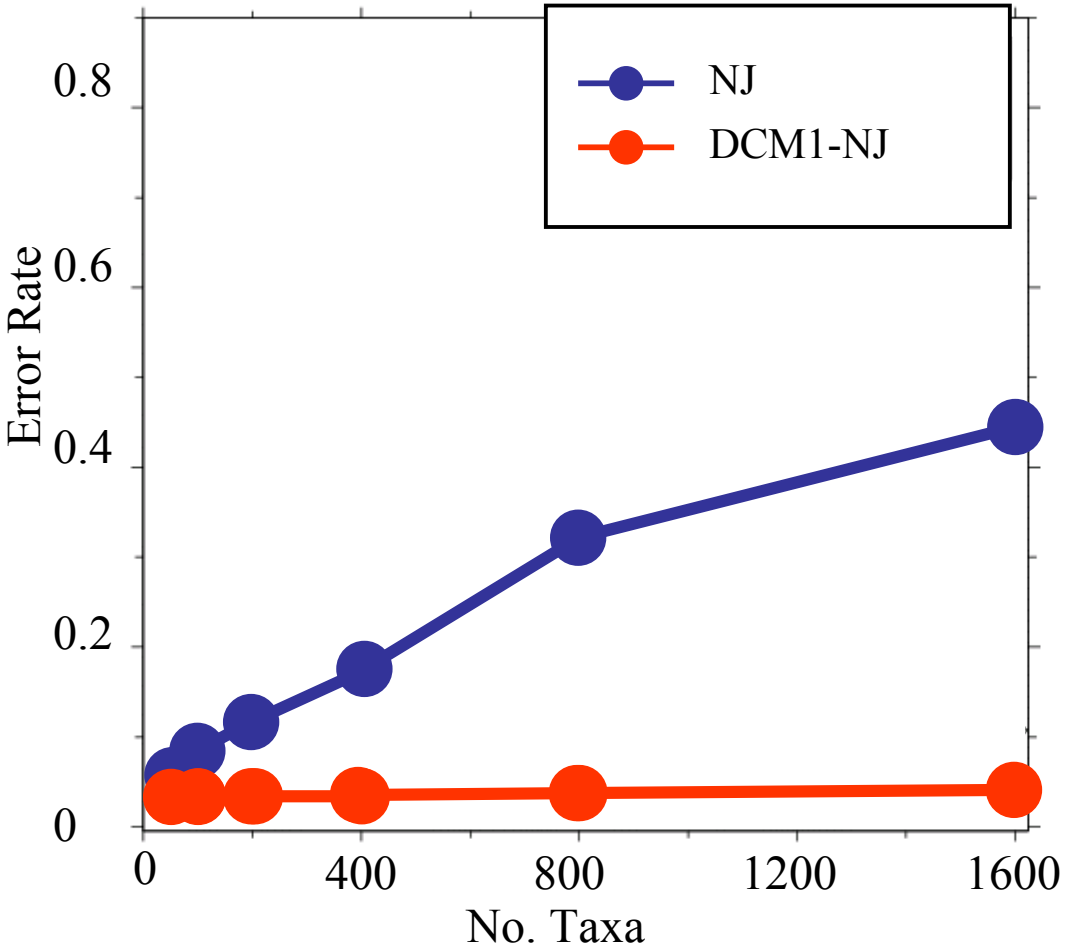
Warnow, St. John, and Moret, SODA 2001



- The DCM1 phase produces a collection of trees (one for each threshold), and the SQS phase picks the “best” tree.
- For a given threshold, the base method is used to construct trees on small subsets (defined by the threshold) of the taxa. These small trees are then combined into a tree on the full set of taxa.

DCM1-boosting distance-based methods

[Nakhleh et al. ISMB 2001]



Theorem (Warnow et al., SODA 2001): DCM1-NJ converges to the true tree from polynomial length sequences.

Many other afc methods, but none (so far) outperform NJ in practice.

Summary and Open Questions

DCM-NJ has better accuracy than NJ

DCM-boosting of other distance-based method also produces very big improvements in accuracy

Other afc methods have been developed with even better theoretical performance

Roch and collaborators have established a threshold for branch lengths, below which logarithmic sequence lengths can suffice for accuracy

Still to be developed: other afc methods with improved empirical performance compared to NJ and other methods

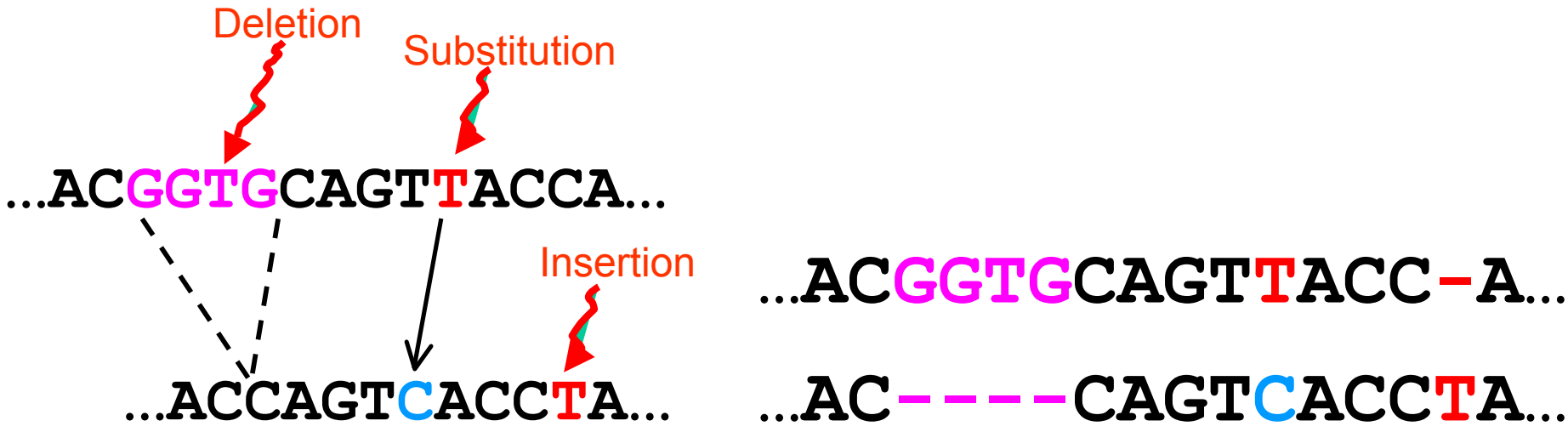
Biggest open problem: sequence length requirement for maximum likelihood (though see Szekely and Steel's work)

What about more complex models?

These results only apply when sequences evolve under these nice substitution-only models.

What can we say about estimating trees when sequences evolve with insertions and deletions ("indels")?

Part II: Estimating trees in the presence of Indels (insertions and deletions)



The true multiple alignment

- Reflects historical substitution, insertion, and deletion events
- Defined using transitive closure of pairwise alignments computed on edges of the true tree

Input: unaligned sequences

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

Phase 1: Multiple Sequence Alignment

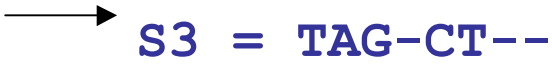
```
S1 = AGGCTATCACCTGACCTCCA
S2 = TAGCTATCACGACCGC
S3 = TAGCTGACCGC
S4 = TCACGACCGACA
```



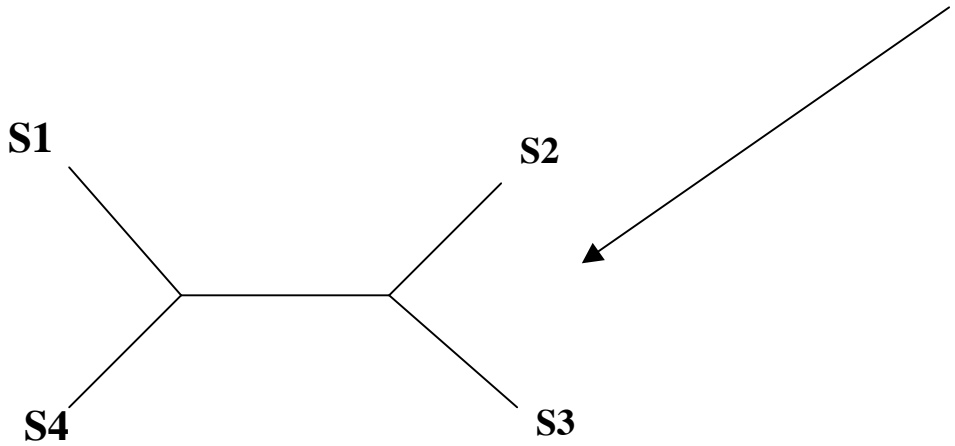
```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-CT-------GACCGC--
S4 = -------TCAC--GACCGACA
```

Phase 2: Construct tree

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



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



Many methods

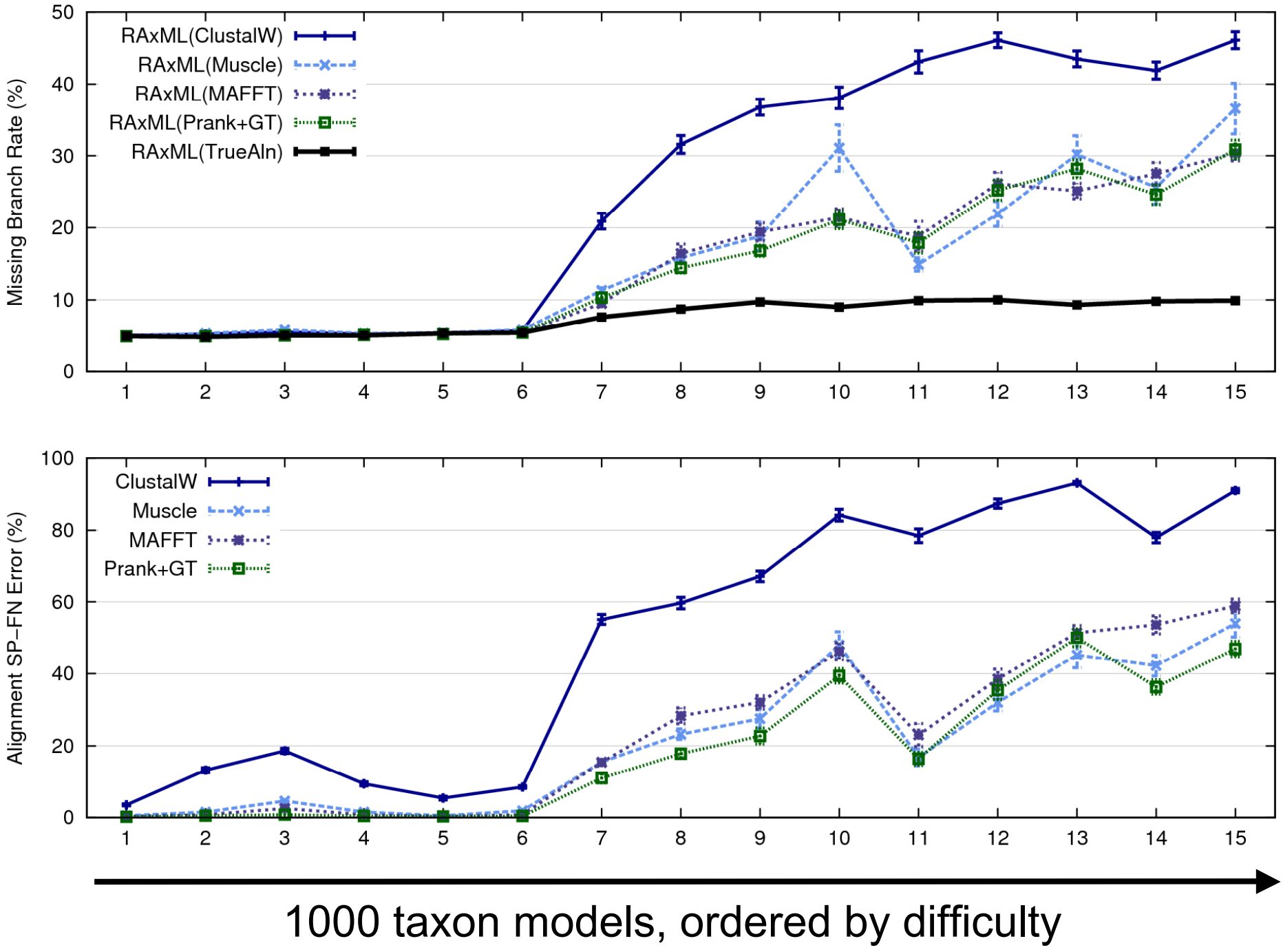
Alignment methods

- Clustal
- POY (and POY*)
- Probcons (and Probtree)
- MAFFT
- Prank
- Muscle
- Di-align
- T-Coffee
- Opal
- *FSA (new method)*
- *Infernal (new method)*
- Etc.

Phylogeny methods

- Bayesian MCMC
- Maximum parsimony
- Maximum likelihood
- Neighbor joining
- FastME
- UPGMA
- Quartet puzzling
- Etc.

***RAxML**: best heuristic for large-scale ML optimization*


Missing Branch Rate (%)
RAxML(ClustalW)
RAxML(Muscle)
RAxML(MAFFT)
RAxML(Prank+GT)
RAxML(TrueAln)
Alignment SP–FN Error (%)
ClustalW
Muscle
MAFFT
Prank+GT
1000 taxon models, ordered by difficulty

Problems with the two-phase approach

- Current alignment methods fail to return reasonable alignments on large datasets with high rates of indels and substitutions.
- Manual alignment is time consuming and subjective.
- *Systematists discard potentially useful markers* if they are difficult to align.

This issues seriously impact large-scale phylogeny estimation (and Tree of Life projects)

Co-estimation methods

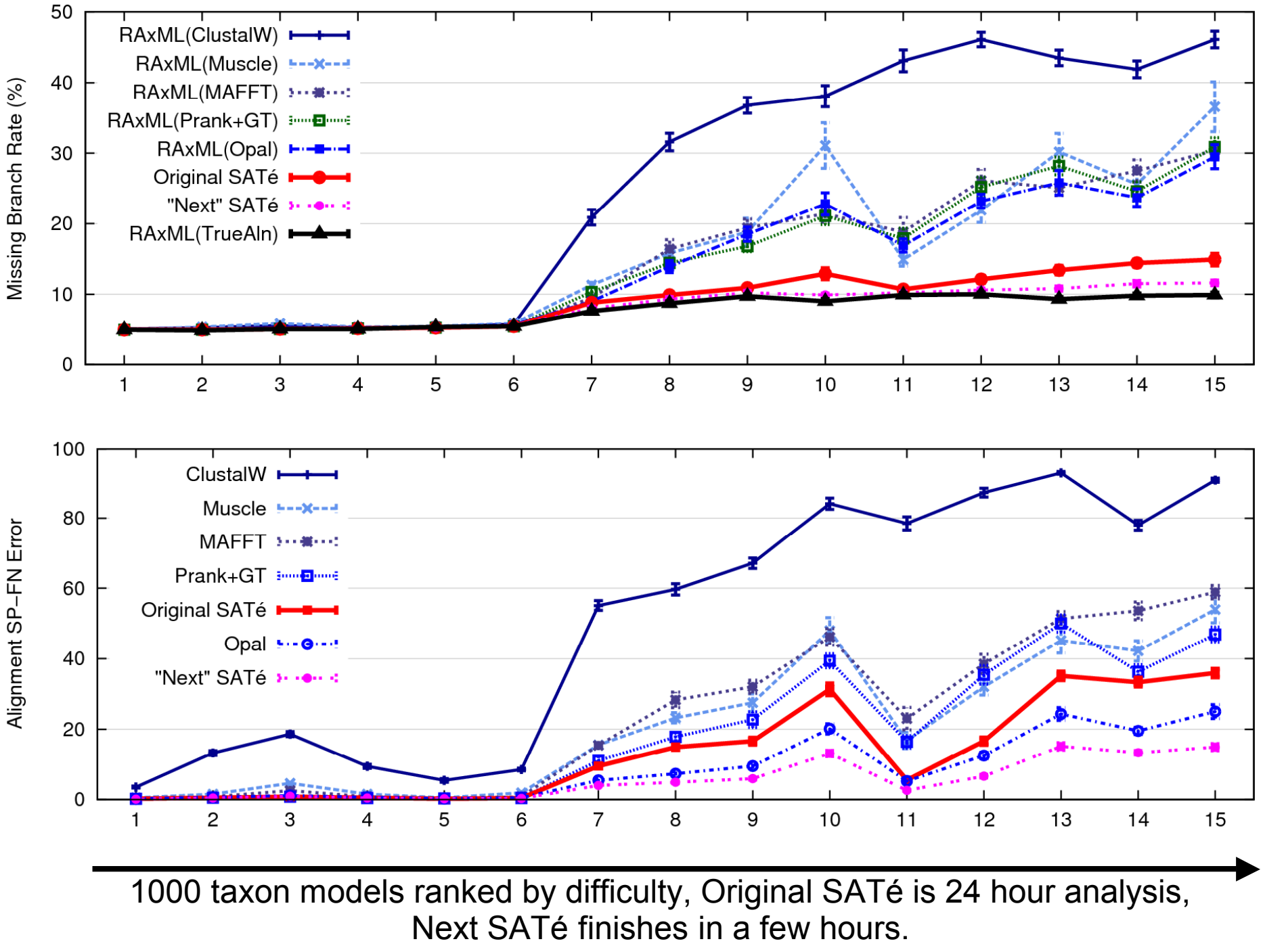
- Statistical methods (e.g., BAliPhy, StatAlign, Alifritz, and others) have been developed, but all are extremely computationally intensive (either unable to analyze datasets with 100 sequences, or using at least a week).

- Steiner Tree approaches based upon edit distances (e.g., POY) are sometimes used, but these have poor topological accuracy and are also computationally intensive.

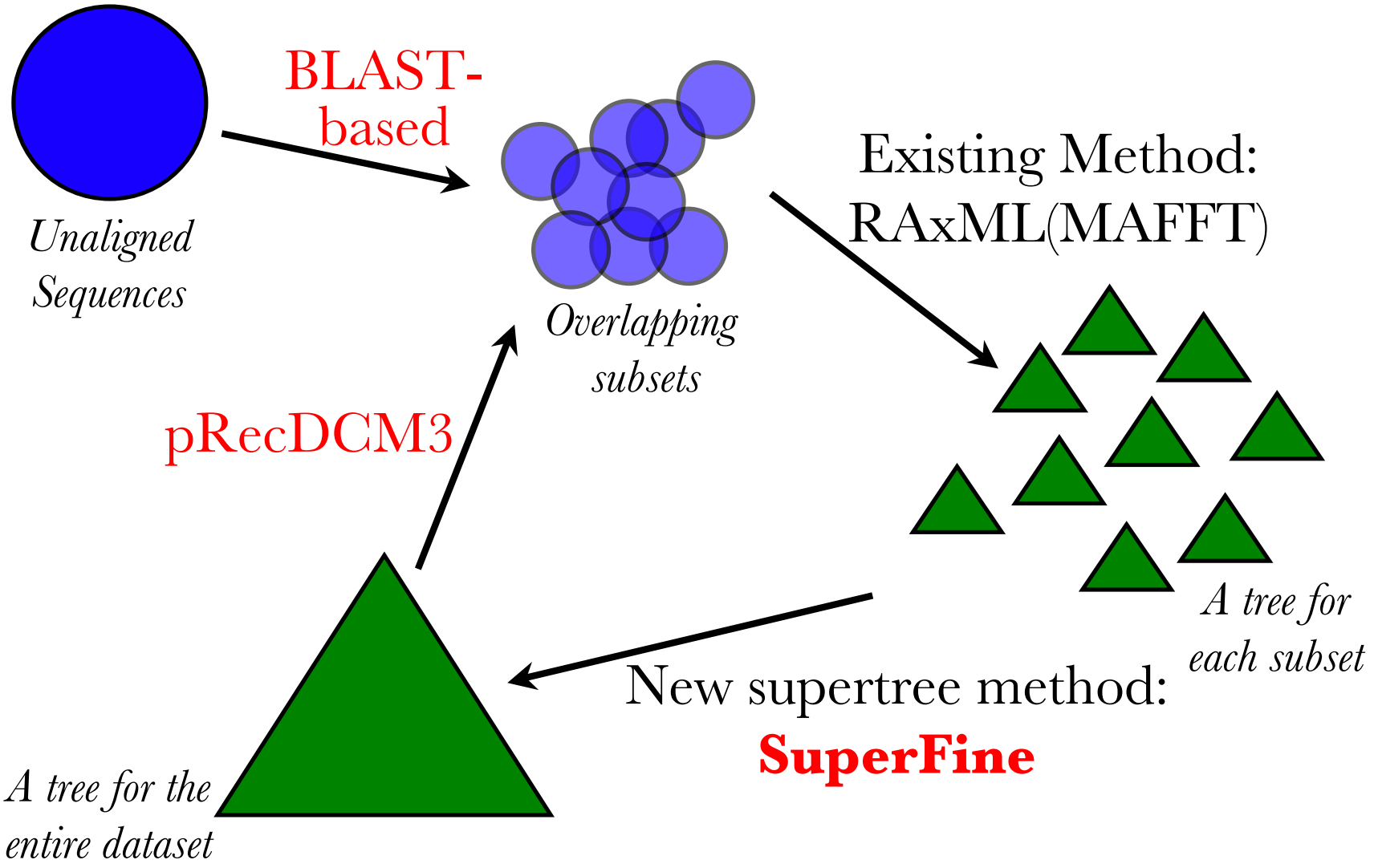
SATé

Liu, Nelesen, Raghavan, Linder, and Warnow, *Science*, 19 June 2009, pp. 1561-1564.

- Kansas SATé software developers: Mark Holder and Jiaye Yu
- Downloadable software for various platforms
- Easy-to-use GUI
- http://phylo.bio.ku.edu/software/sate/sate.html


Missing Branch Rate (%)
RAxML(ClustalW)
RAxML(Muscle)
RAxML(MAFFT)
RAxML(Prank+GT)
RAxML(Opal)
Original SATé
"Next" SATé
RAxML(TrueAln)
Alignment SP–FN Error
ClustalW
Muscle
MAFFT
Prank+GT
Original SATé
Opal
"Next" SATé
1000 taxon models ranked by difficulty, Original SATé is 24 hour analysis,
Next SATé finishes in a few hours.

DACTAL


BLAST-based
Unaligned Sequences
Overlapping subsets
Existing Method: RAxML(MAFFT)
A tree for each subset
New supertree method:
SuperFine
pRecDCM3
A tree for the entire dataset

Average of Three Largest CRW Datasets

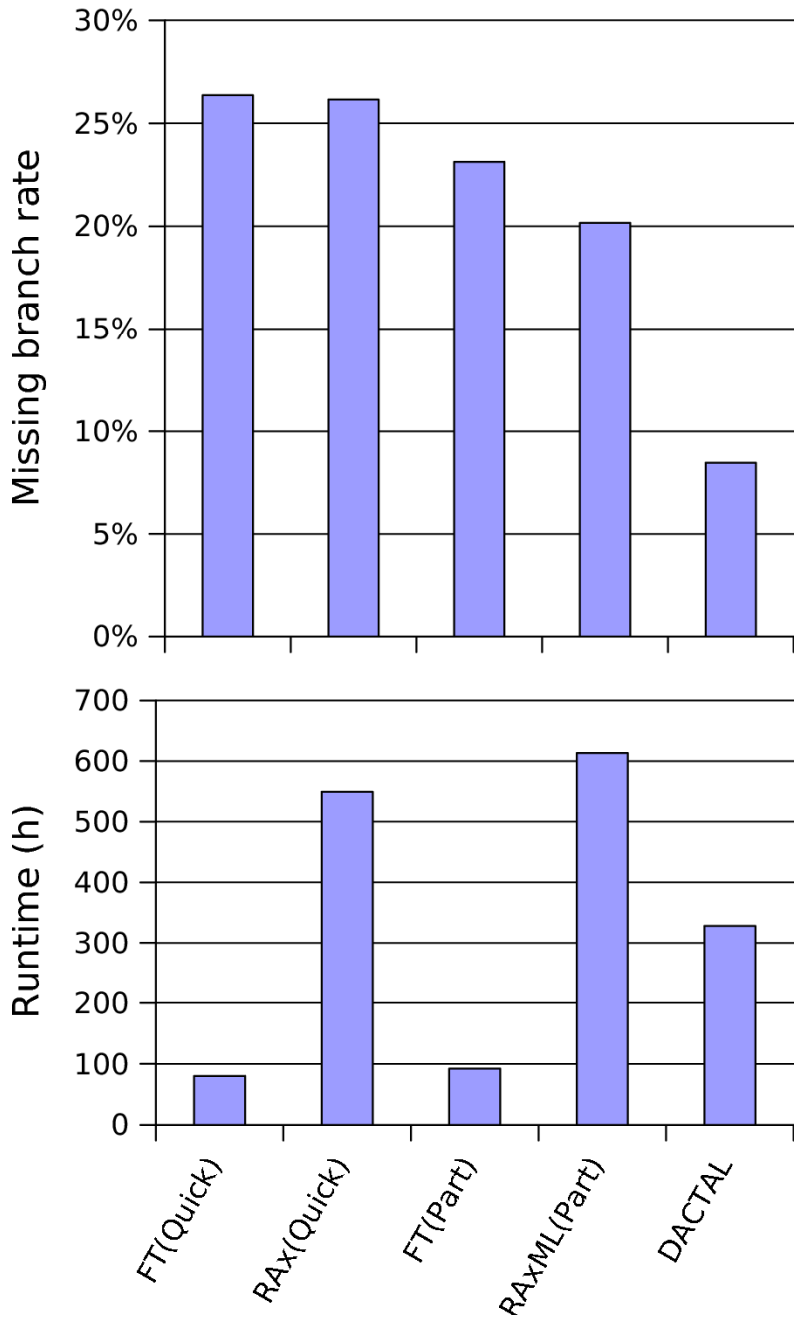
Datasets with curated alignments based upon secondary structure with 6323 to 27,643 sequences (16S.B.ALL, 16S.T, and16S.3).

Reference trees are 75% RAxML bootstrap trees

DACTAL run with at most 5 iterations from FastTree(PartTree)

Observations:

- Quicktree and PartTree the only alignment methods that run on all three datasets
- DACTAL is robust to starting tree (same final accuracy results from worse starting trees)



Observations

- SATé and DACTAL outperform two-phase methods with respect to topological accuracy on large, hard-to-align datasets.
- DACTAL outperforms SATé on the largest datasets, and can analyze datasets that SATé cannot.
- We do not have any theoretical explanation for why these methods perform well.

Implications

- We need new methods for very large phylogenetic analyses.

- Don’t throw out data that look hard to analyze - design new methods!

Implications, continued

- Divide-and-conquer methods can greatly improve the accuracy and speed of phylogeny and alignment estimation.
- Theoretical performance doesn't predict empirical performance.
- Many open questions result from considering phylogeny estimation with indels.

Some open questions

- What is the sequence length requirement for maximum likelihood?
- Are trees identifiable under models including “long gaps”?
- Why do SATé and DACTAL perform well?
- Under standard implementations of ML, gaps are treated as missing data: what are the consequences?

Projects in my lab

- Co-estimation of alignments and trees
- Supertree methods
- Comparative genomics: whole genome phylogeny using gene order and content
- Estimating species trees from gene trees
- Reticulate phylogeny detection and estimation
- Faster maximum likelihood methods
- Datamining sets of trees
- Computational historical linguistics

Acknowledgments

- The John P. Simon Guggenheim Foundation
- The David and Lucile Packard Foundation
- The Radcliffe Institute for Advanced Study
- Microsoft Research New England
- National Science Foundation
- Collaborators: Peter Erdos, Mark Holder, Randy Linder, Kevin Liu, Bernard Moret, Luay Nakhleh, Serita Nelesen, Sindhu Raghavan, Usman Roshan, Jerry Sun, Rahul Suri, Shel Swenson, Alexis Stamatakis, Mike Steel, Katherine St. John, Laszlo Szekely, Li-San Wang, and Jiaye Yu.