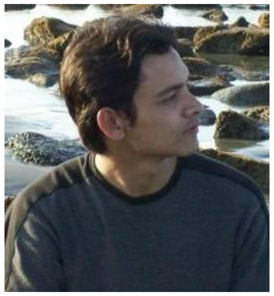
Software for Scientists

Tandy Warnow

Department of Computer Science

University of Texas at Austin

Warnow Laboratory



PhD students: Siavash Mirarab[1], Nam Nguyen, and Md. S. Bayzid[2]
Undergrad: Keerthana Kumar
Lab Website: http://www.cs.utexas.edu/users/phylo

Funding: Guggenheim Foundation, Packard Foundation, NSF, Microsoft Research New England, David Bruton Jr. Centennial Professorship, and TACC (Texas Advanced Computing Center)

[1]HHMI International Predoctoral Fellow, [2]Fulbright Predoctoral Fellow

Main Points

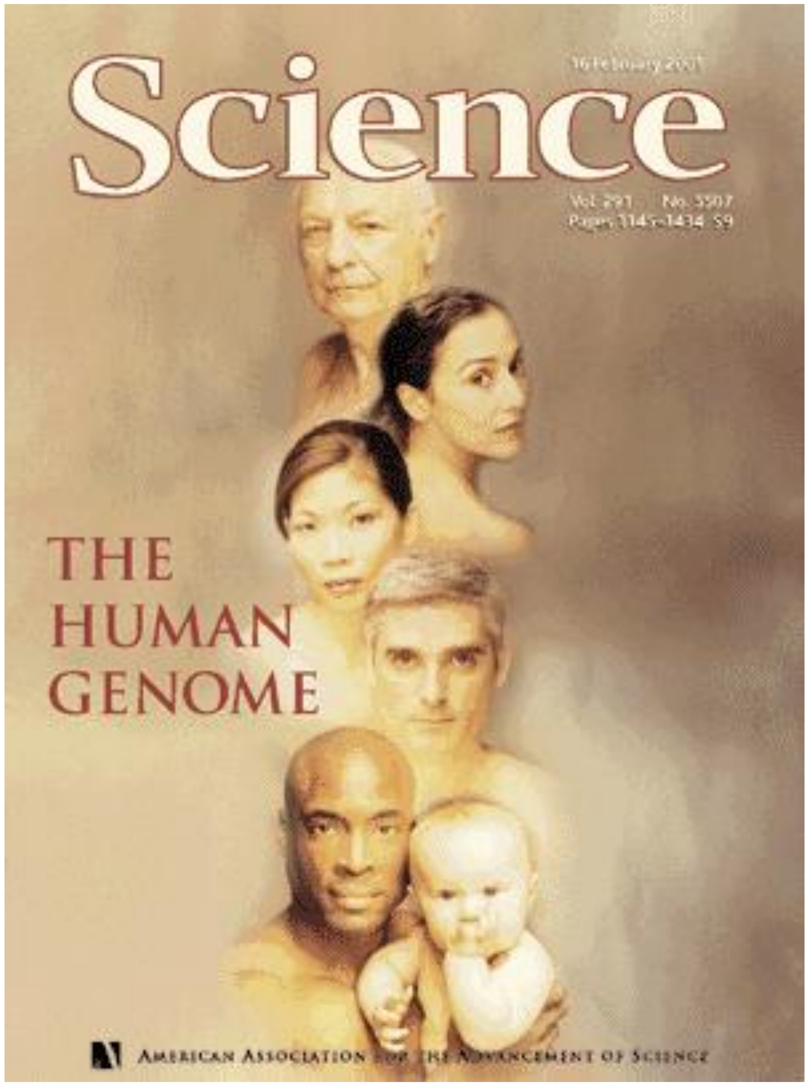
- Computer scientists develop algorithms and software to make it possible for scientists to get improved accuracy in their analyses.
- These algorithms involve creative strategies, including divide-and-conquer, iteration, and randomization.
- Extensive simulations and data analyses are part of the evaluation process!

BIG DATA
A REVOLUTION THAT WILL TRANSFORM HOW WE LIVE, WORK, AND THINK
VIKTOR MAYER-SCHÖNBERGER
KENNETH CUKIER

data
Hadoop
database
compression
databases
mobile
Data
terabytes
time
tools
store
now
query
big
Twitter
NoSQL
SQL
big-data
new
support
storage
processing
information
analysis
analyses
example
column-store

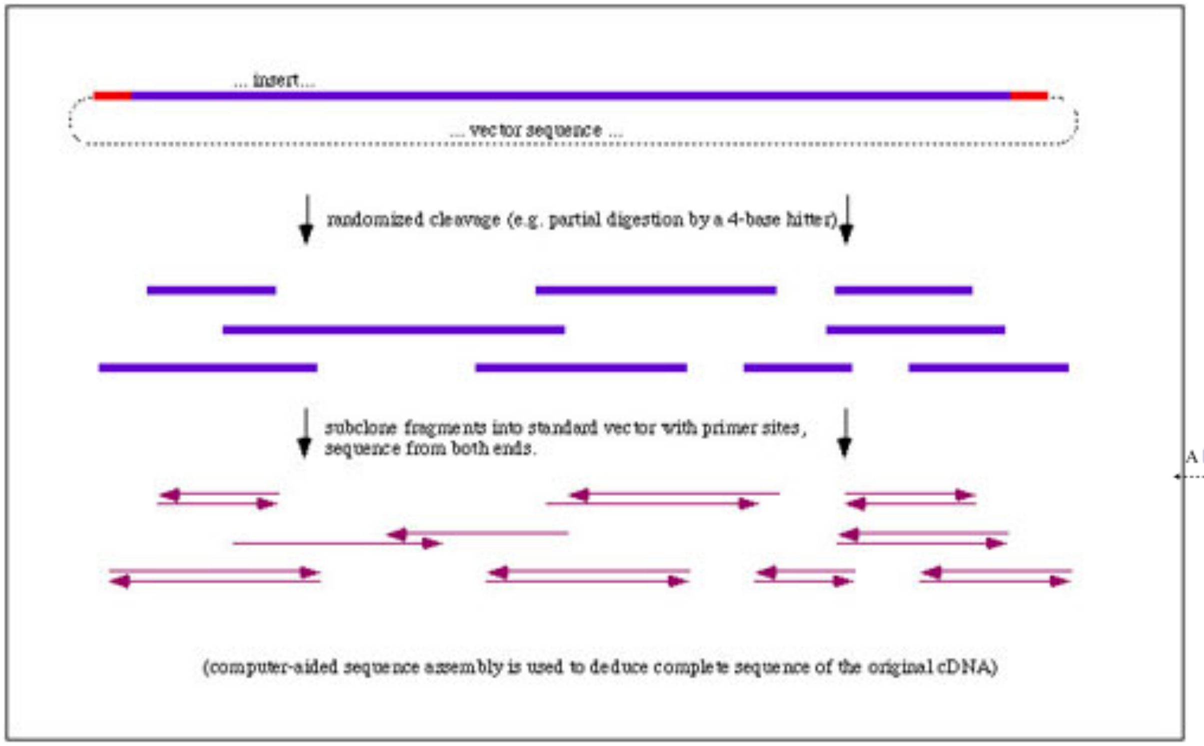
IT COMES FROM EVERYWHERE. IT KNOWS ALL.
ACCORDING TO THE BOOK OF WIKIPEDIA, ITS NAME IS "BIG DATA."
BIG DATA
BUT IF WE ACCEPT BIG DATA IN OUR SERVERS, WE WILL BE SAVED FROM BANKRUPTCY.
LET US PAY.
$

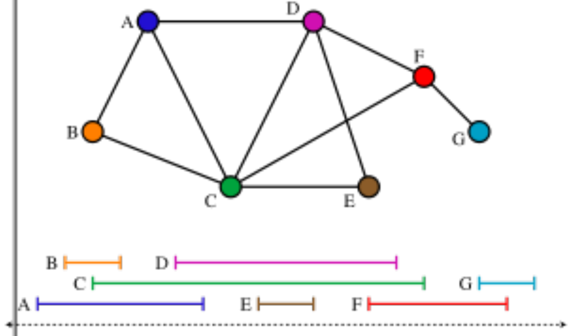
BigData for Biology: Genomics



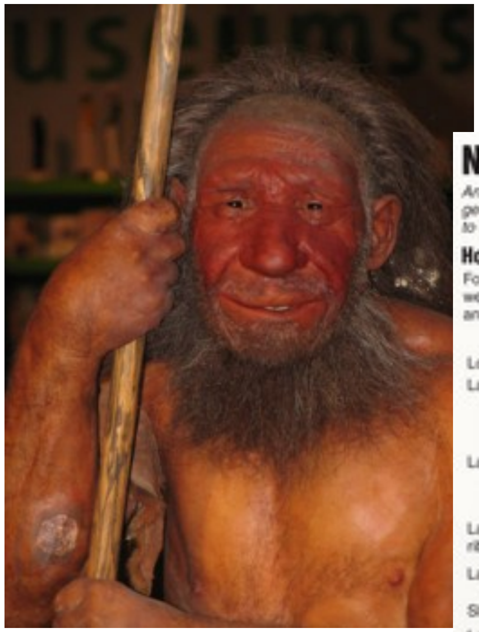
Whole Genome Sequencing:

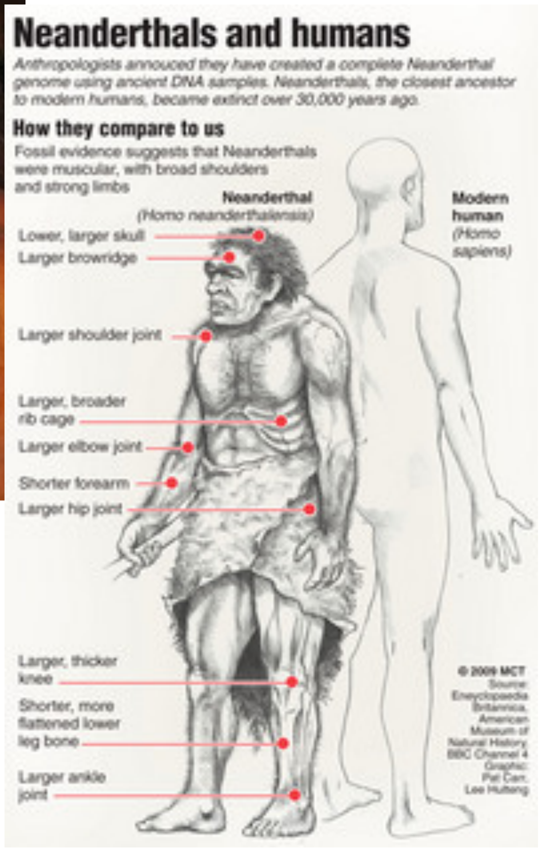
Graph Algorithms and Combinatorial Optimization!

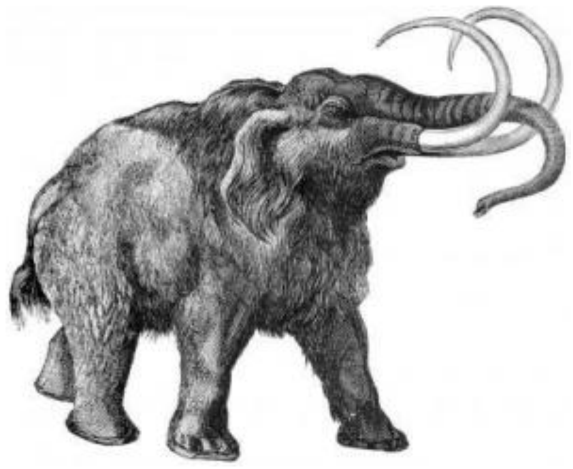

... insert...
... vector sequence ...
randomized cleavage (e.g. partial digestion by a 4-base hitter)
subclone fragments into standard vector with primer sites, sequence from both ends.
(computer-aided sequence assembly is used to deduce complete sequence of the original cDNA)



Other Genome Projects! (Neandertals, Wooly Mammoths, and more ordinary creatures…)

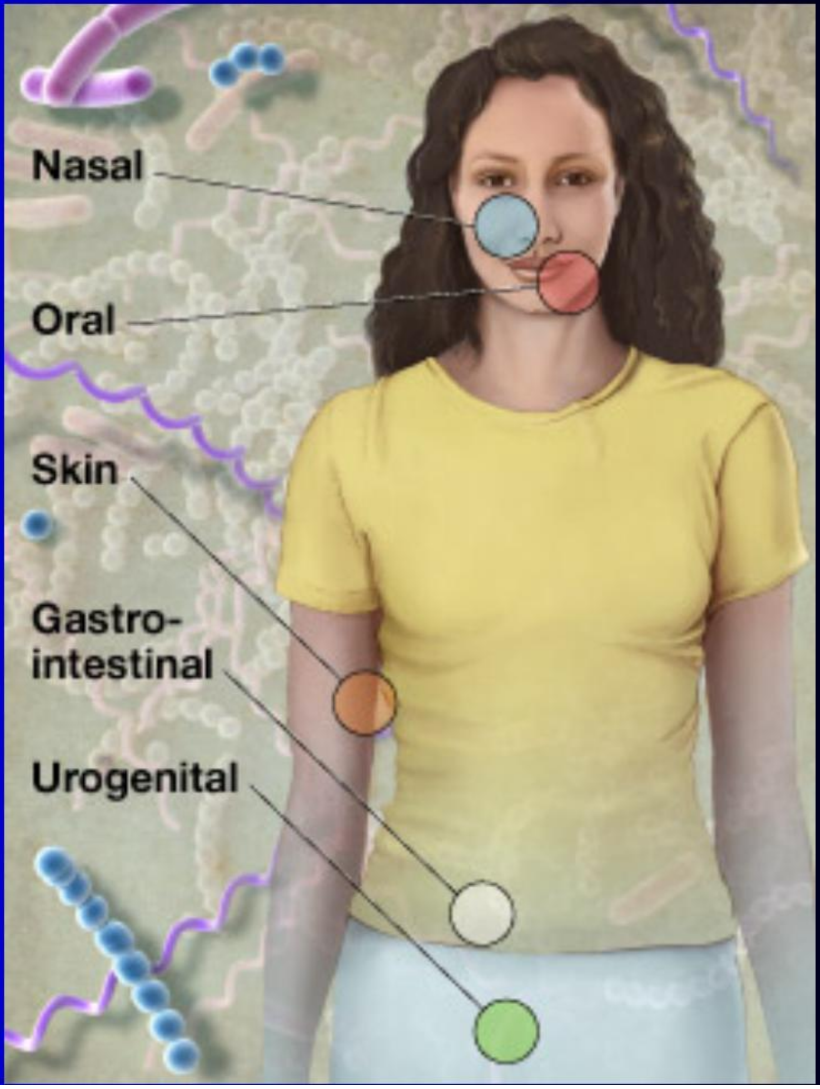



Neanderthals and humans
Anthropologists annouced they have created a complete Neanderthal genome using ancient DNA samples. Neanderthals, the closest ancestor to modern humans, became extinct over 30,000 years ago.
How they compare to us
Fossil evidence suggests that Neanderthals were muscular, with broad shoulders and strong limbs
Neanderthal (Homo neanderthalensis)
Modern human (Homo sapiens)
Lower, larger skull
Larger browridge
Larger shoulder joint
Larger, broader rib cage
Larger elbow joint
Shorter forearm
Larger hip joint
Larger, thicker knee
Shorter, more flattened lower leg bone
Larger ankle joint
© 2009 MCT
Source: Encyclopaedia Britannica, American Museum of Natural History, BBC Channel 4
Graphic: Pat Carr, Lee Hulteng

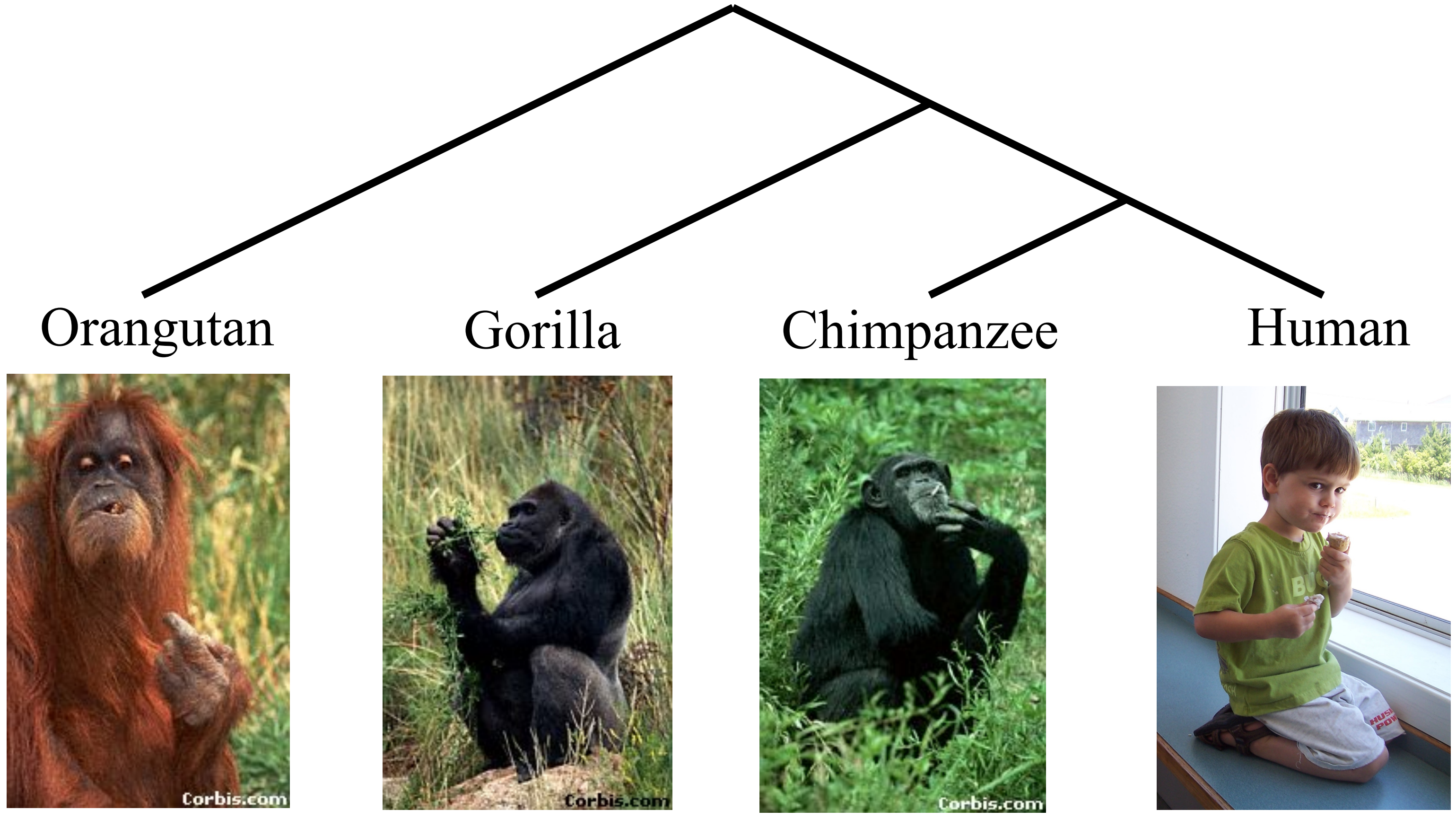


The NIH Human Microbiome Project

**25,000 human genes,
1,000,000 bacterial genes**

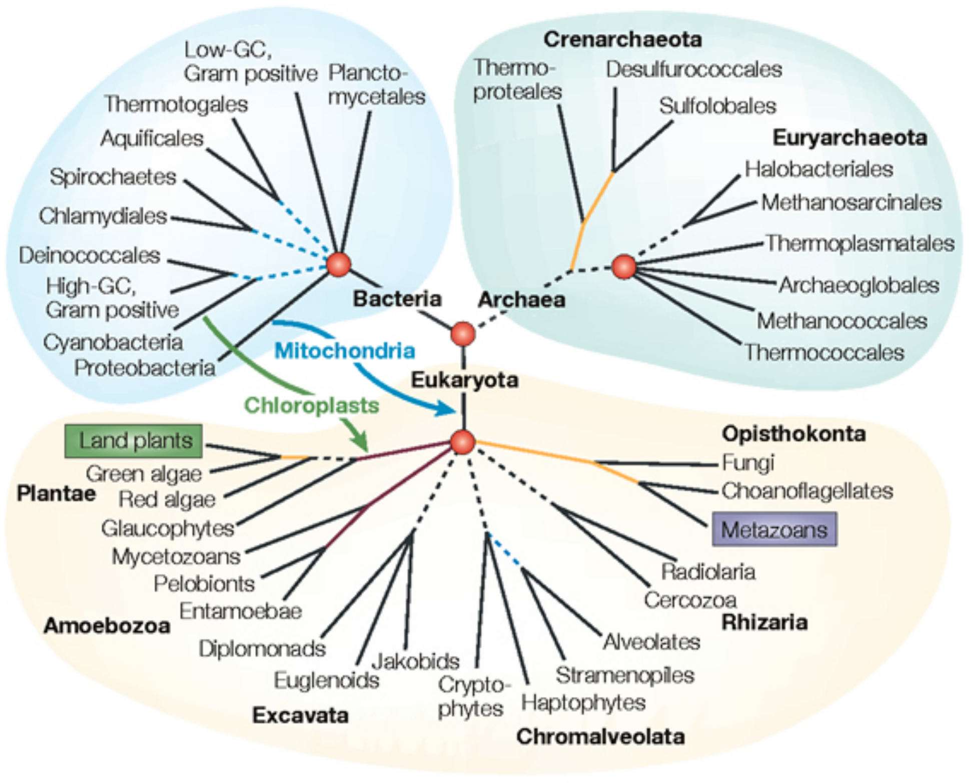


Phylogeny (evolutionary tree)



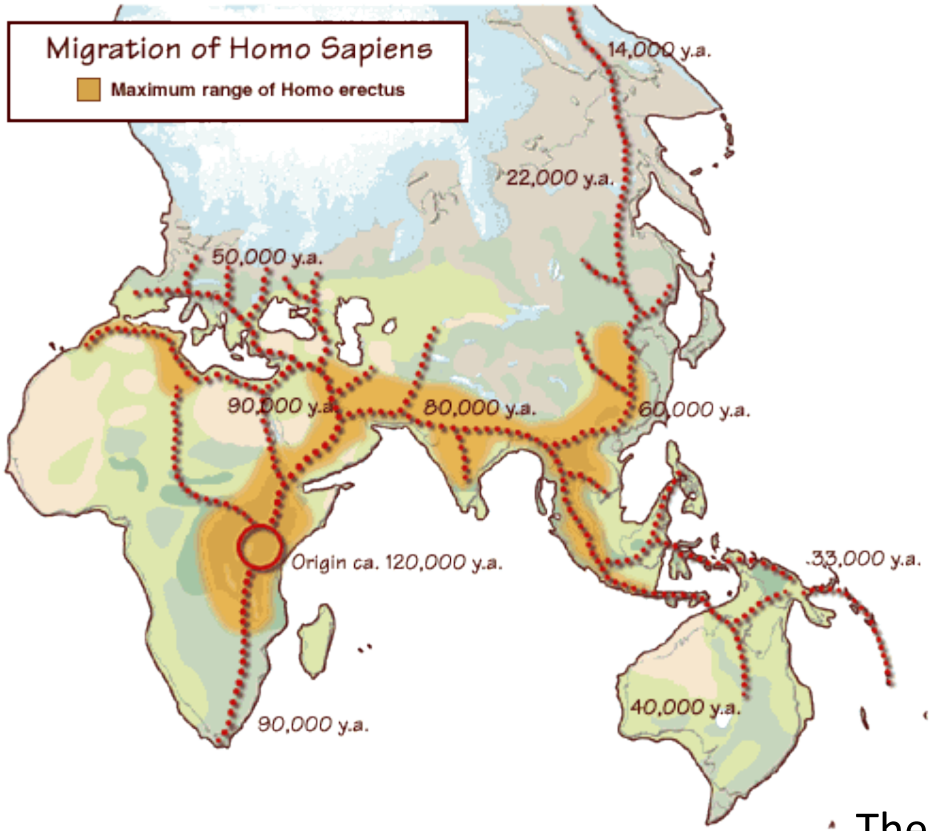
From the Tree of the Life Website,
University of Arizona

Assembling the Tree of Life



Nature Reviews | Genetics

Where did humans come from, and how did they move throughout the globe?



- The 1000 Genome Project: using human genetic variation to better treat diseases

Goal: Constructing a Tree of Life

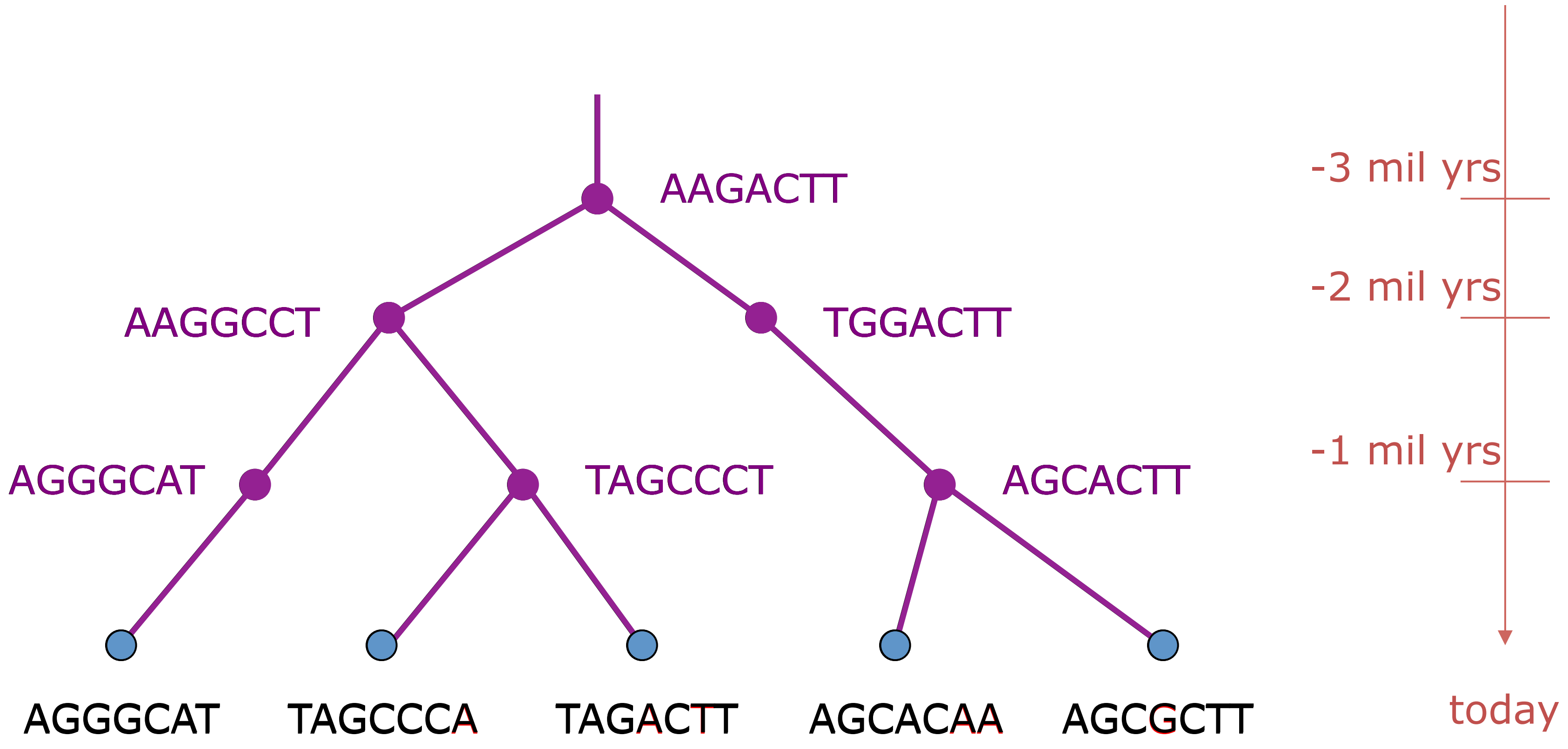
- Why do we want to do this?
- What are the computer science challenges?
- What kinds of techniques help us do this well?

Why? Because!

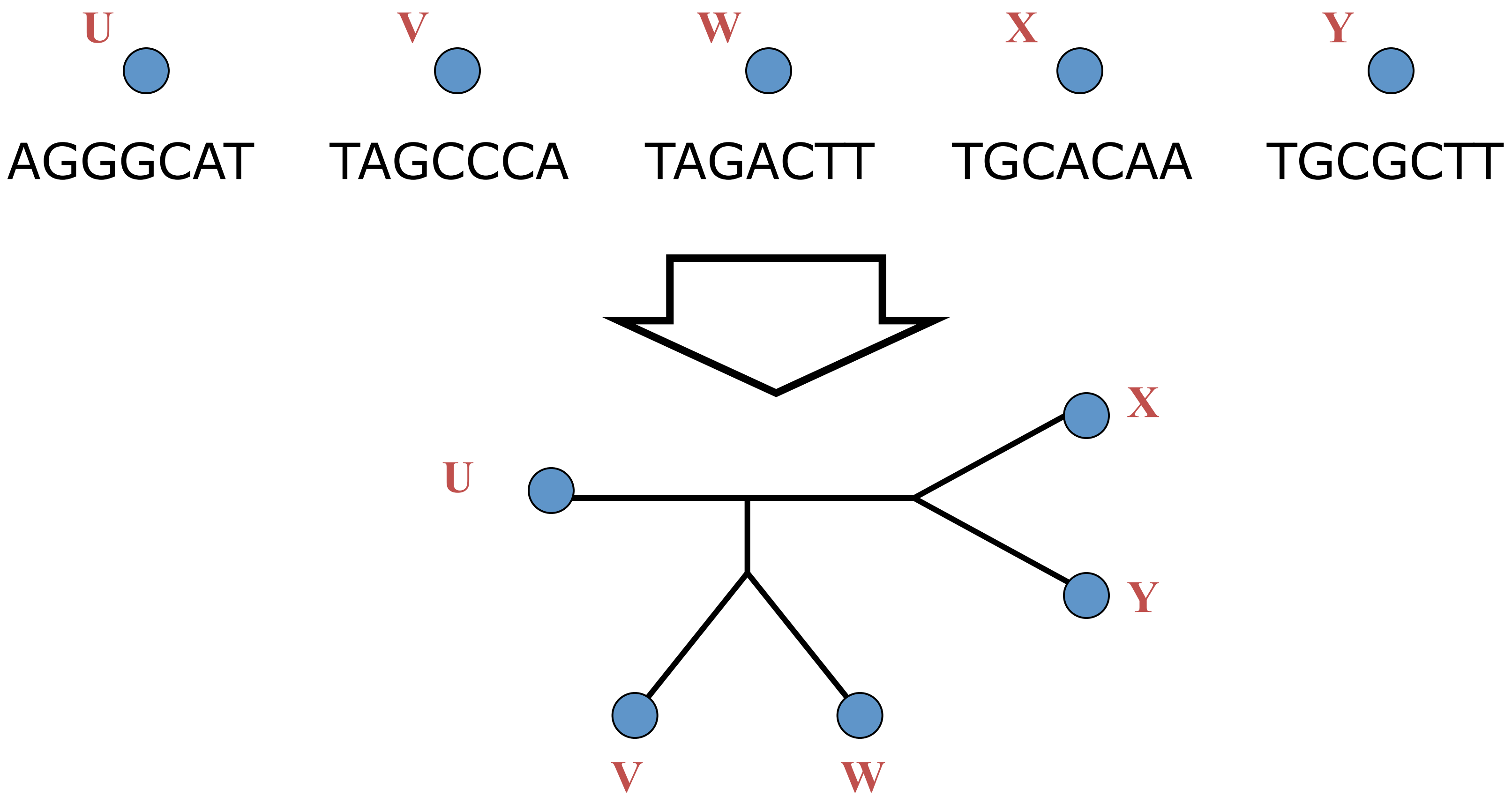
- Evolutionary history relates all organisms and genes, and helps us understand and predict
  - interactions between genes (genetic networks)
  - drug design
  - predicting functions of genes
  - influenza vaccine development
  - origins and spread of disease
  - origins and migrations of humans

"Nothing in biology makes sense except in the light of evolution"
Dobzhansky

DNA Sequence Evolution



Phylogeny Problem



Performance criteria

- Running time.
- Space.
- "Topological accuracy" with respect to the underlying ***true tree or true alignment.*** Typically studied in simulation or using mathematical theory.
- Accuracy with respect to a particular criterion (e.g. maximum likelihood score), on real data.

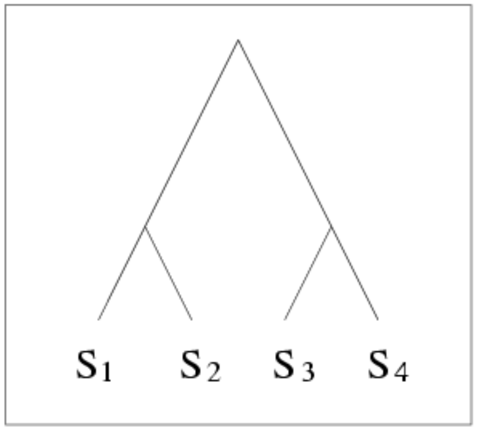
Tree Estimation Methods

- Bayesian MCMC -- very very slow to converge
- NP-hard problems:
  - Maximum parsimony
  - Maximum likelihood
- Polynomial time methods:
  - Neighbor joining
  - FastME
  - UPGMA
  - Quartet puzzling

Tree Estimation Methods

- Bayesian MCMC -- very very slow to converge
- NP-hard problems:
  - Maximum parsimony
  - Maximum likelihood
- Polynomial time methods:
  - Neighbor joining
  - FastME
  - UPGMA
  - Quartet puzzling

Distance-based estimation



TRUE TREE

S_1	ACAATTAGAAC
S_2	ACCCTTAGAAC
S_3	ACCATTCCAAC
S_4	ACCAGACCAAC

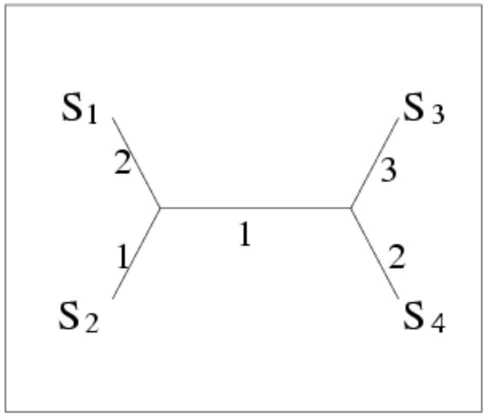
DNA SEQUENCES

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

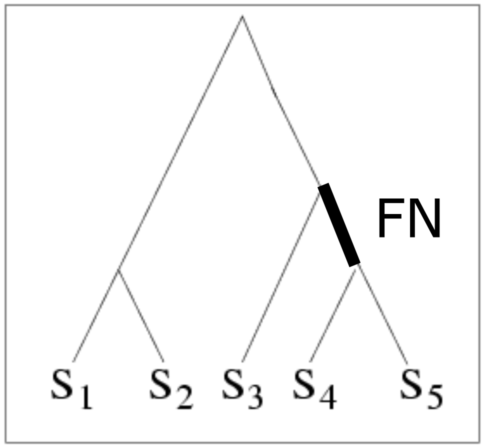
DISTANCE MATRIX

METHODS SUCH AS NEIGHBOR JOINING



INFERRED TREE

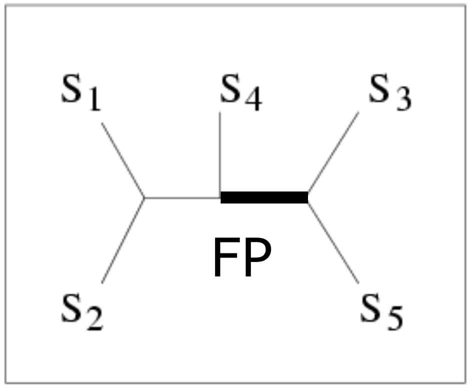
Quantifying Error



TRUE TREE

S_1	ACAATTAGAAC
S_2	ACCCTTAGAAC
S_3	ACCATTCCAAC
S_4	ACCAGACCAAC
S_5	ACCAGACCGGA

DNA SEQUENCES

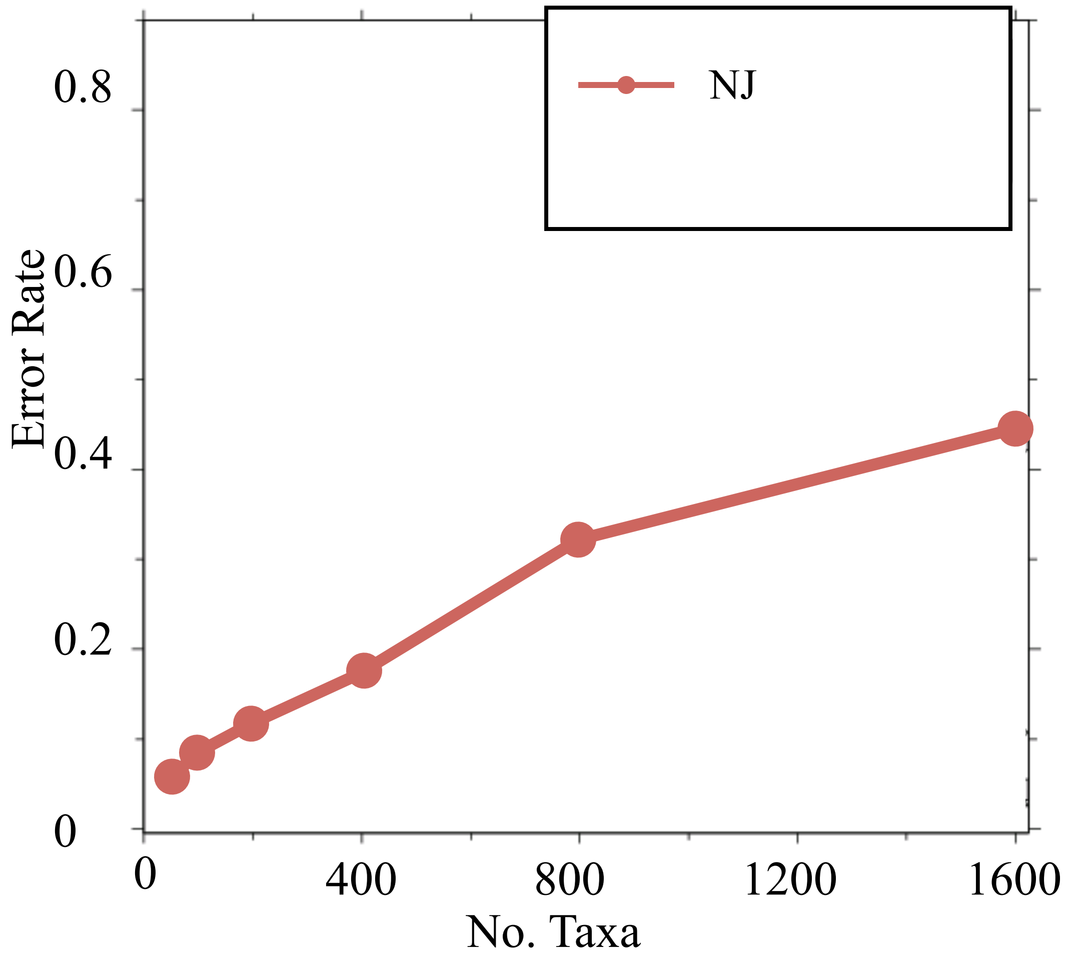


INFERRED TREE

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

50% error rate

Polynomial time methods typically have high error rates on large trees



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

Neighbor joining method for tree estimation.

Error rates reflect proportion of incorrect edges in inferred trees.

[Nakhleh et al. ISMB 2001]

Tree Estimation Methods

- Bayesian MCMC -- very very slow to converge
- NP-hard problems:
  - Maximum parsimony
  - Maximum likelihood
- Polynomial time methods:
  - Neighbor joining
  - FastME
  - UPGMA
  - Quartet puzzling

A polynomial-time problem

- **2-colorability**: Given graph G = (V,E), determine if we can assign colors **red** and **blue** to the vertices of G so that no edge connects vertices of the same color.

A polynomial-time problem

- **2-colorability**: Given graph G = (V,E), determine if we can assign colors **red** and **blue** to the vertices of G so that no edge connects vertices of the same color.

- Greedy Algorithm. Start with one vertex and make it **red**, and then make all its neighbors **blue**, and keep going. If you succeed in coloring the graph without making two nodes of the same color adjacent, the graph can be 2-colored.

What about this?

- **3-colorability:** Given graph G, determine if we can assign **red, blue,** and **green** to the vertices in G so that no edge connects vertices of the same color.

- Some decision problems can be solved in polynomial time:
  - Can graph G be 2-colored?
  - Does graph G have a Eulerian tour?
- Some decision problems *seem* to not be solvable in polynomial time:
  - Can graph G be 3-colored?
  - Does graph G have a Hamiltonian cycle?

What about this?

- **3-colorability:** Given graph G, determine if we can assign **red,** **blue,** and **green** to the vertices in G so that no edge connects vertices of the same color.

- This problem is provably NP-hard. What does this mean?

P vs. NP, continued

- The “big” question in theoretical computer science is:
  - Is it possible to solve an NP-hard problem in polynomial time?
- If the answer is “yes”, then **all** NP-hard problems can be solved in polynomial time, so **P=NP**.   This is generally not believed.

Coping with NP-hard problems

Since NP-hard problems may not be solvable in polynomial time, the options are:

- Solve the problem *exactly* (but use lots of time on some inputs)
- Use *heuristics* which may not solve the problem exactly (and which might be computationally expensive, anyway)

Solving NP-hard problems exactly is … unlikely

- Number of (unrooted) binary trees on *n* leaves is (2n-5)!!
- If each tree on **1000** taxa could be analyzed in **0.001** seconds, we would find the best tree in

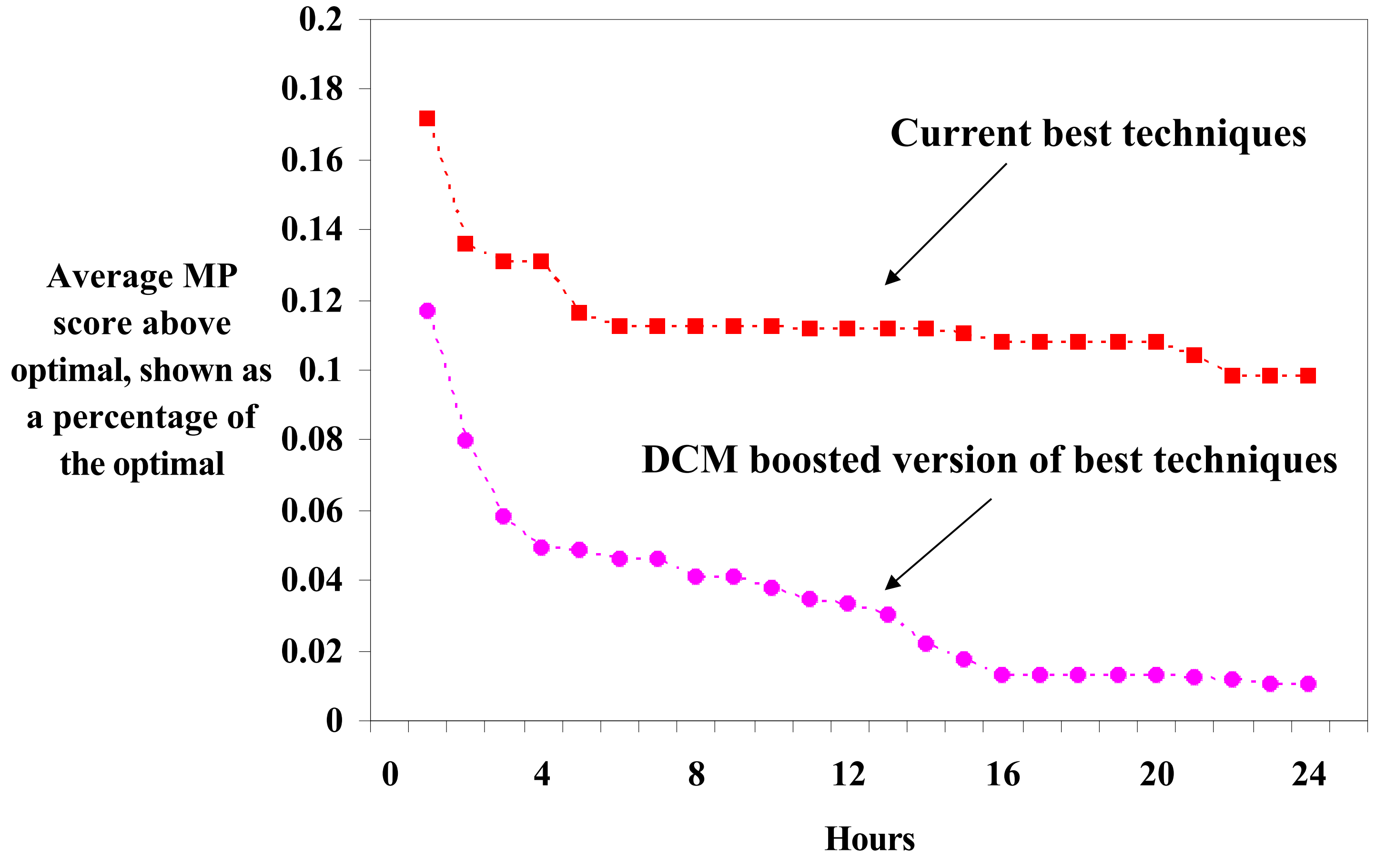
  2890 millennia

#leaves	#trees
4	3
5	15
6	105
7	945
8	10395
9	135135
10	2027025
20	2.2×10^{20}
100	4.5×10^{190}
1000	2.7×10^{2900}

Tree Estimation Methods

- Bayesian MCMC -- very very slow to converge
- NP-hard problems:
  - Maximum parsimony
  - Maximum likelihood
- Polynomial time methods:
  - Neighbor joining
  - FastME
  - UPGMA
  - Quartet puzzling

NP-hard optimization problems: better accuracy, but slow to find good solutions



Comparison of TNT to Rec-I-DCM3(TNT) on one large dataset

NP-hard problems in Biology

- Optimization problems in biology are almost all NP-hard, and heuristics may run for months (or years!!) before finding *local optima*.

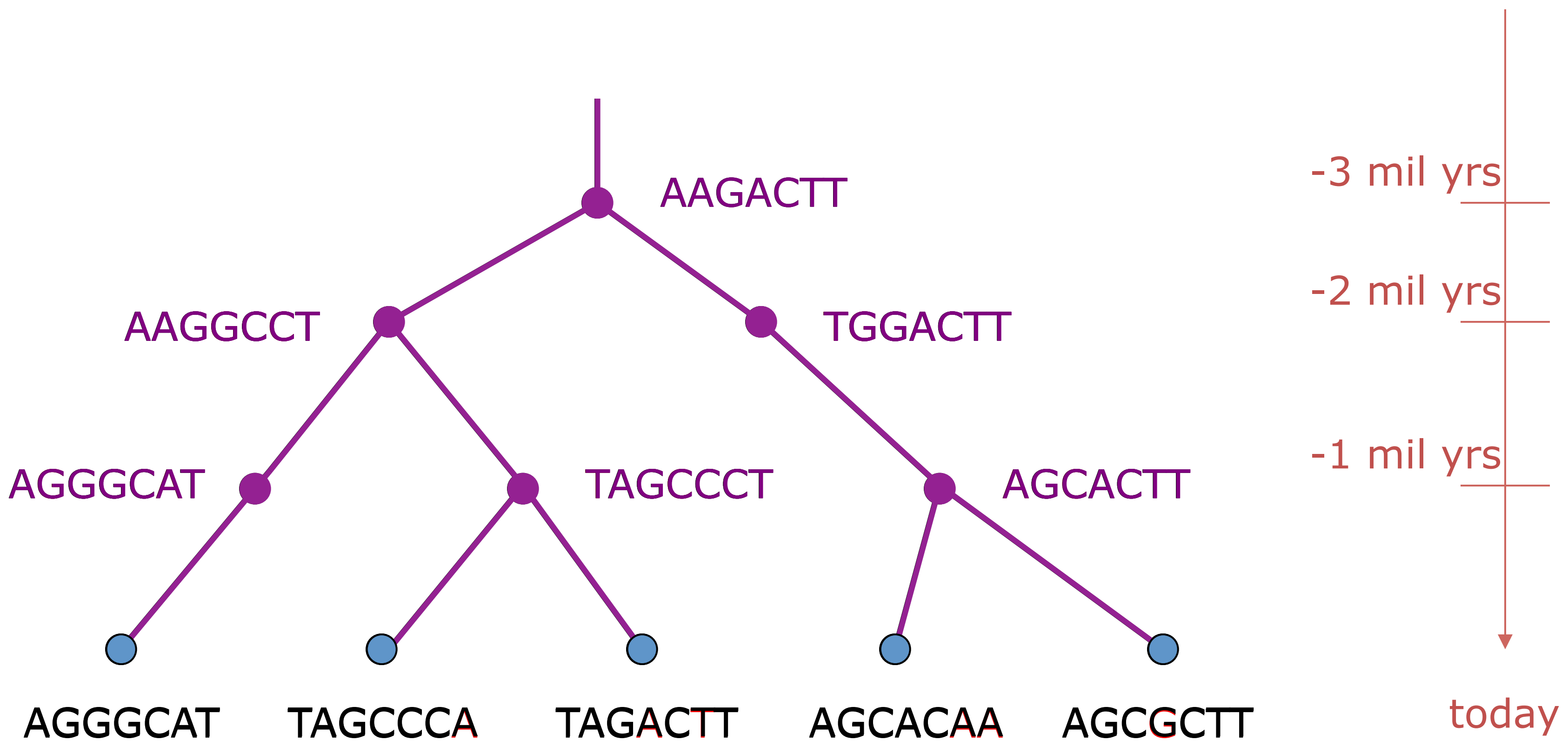
- The challenge here is to find better heuristics, since exact solutions are very unlikely to ever be achievable on large datasets.

Challenges

Tree estimation:

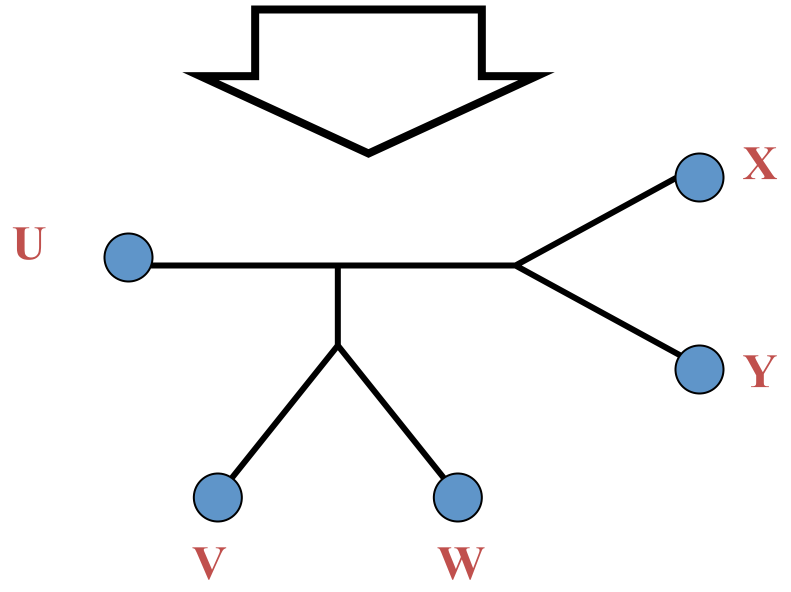
- The most accurate methods are heuristics for NP-hard problems, but these may not run efficiently (or find good solutions) on big datasets.
- Polynomial time methods have poor accuracy on large datasets

DNA Sequence Evolution

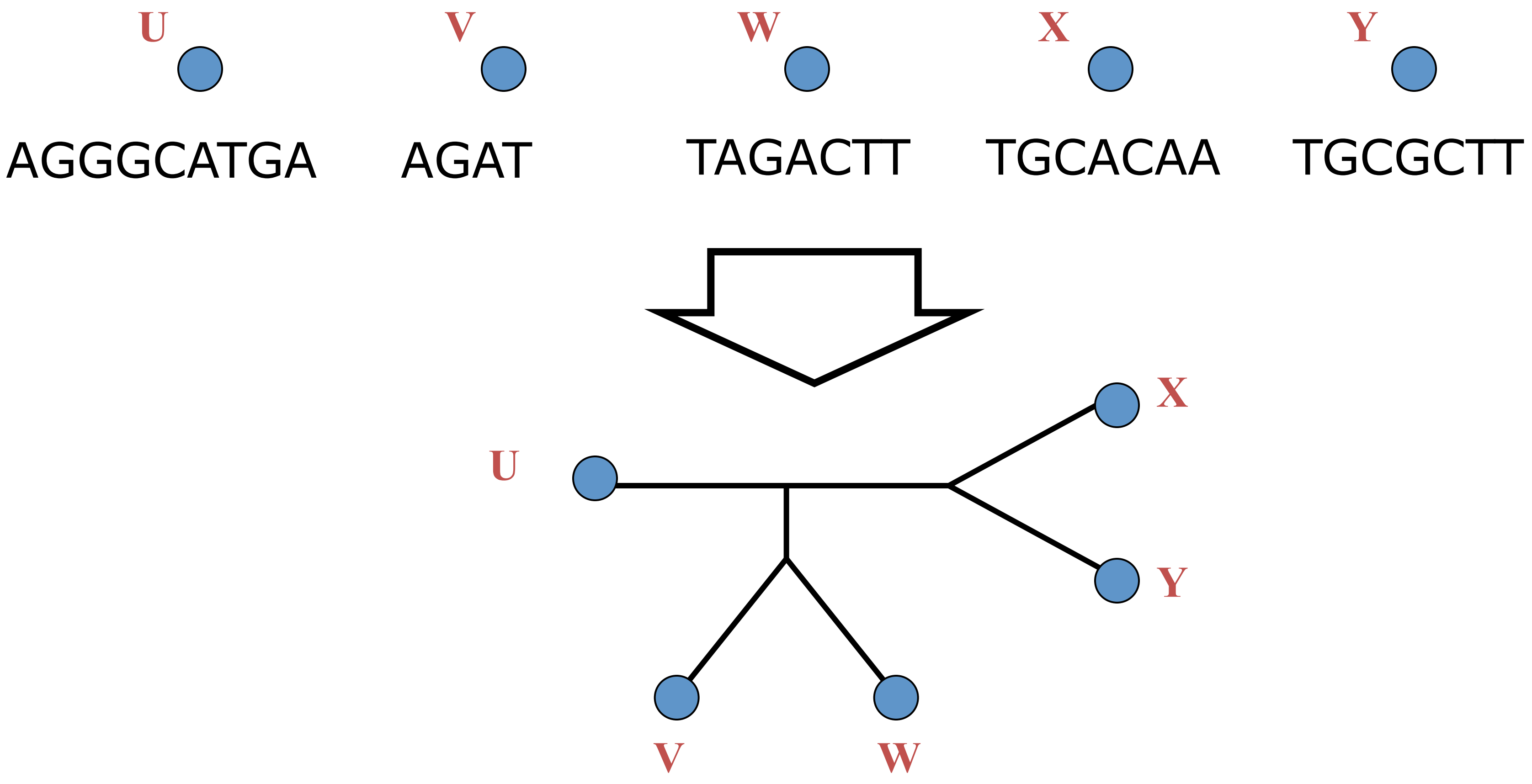


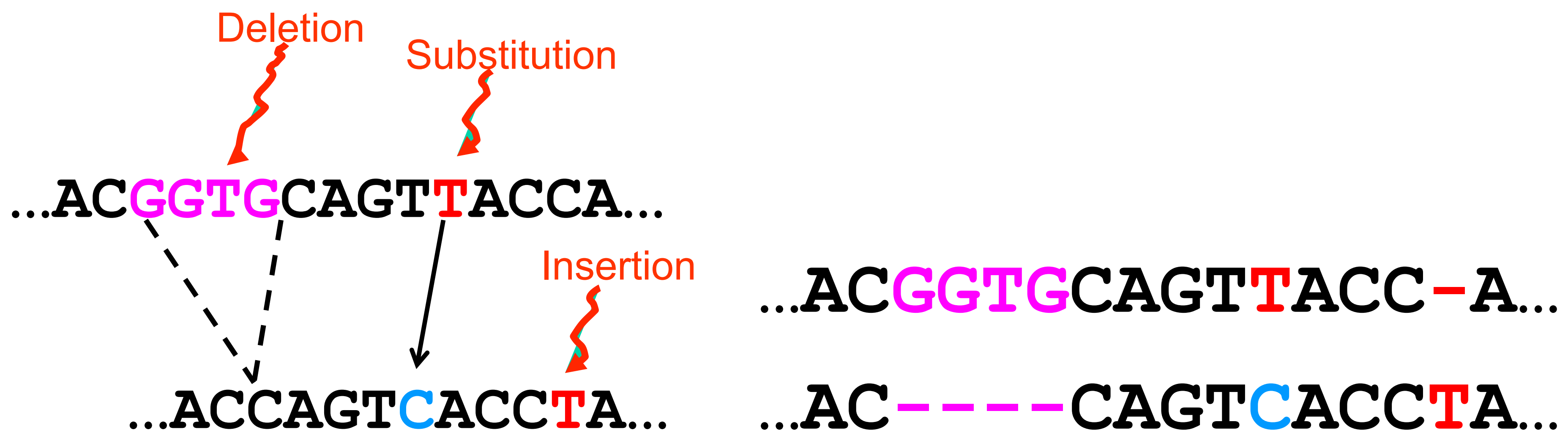
Phylogeny Problem

U	V	W	X	Y
AGGGCAT	TAGCCCA	TAGACTT	TGCACAA	TGCGCTT



The “real” problem!





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
```

$\longrightarrow$

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

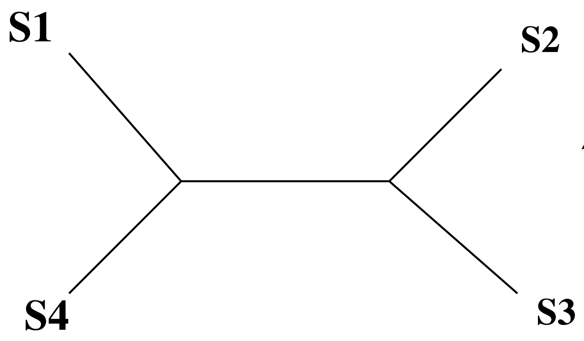
Phase 2: Construct tree

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

$\longrightarrow$

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

$\swarrow$



Two-phase estimation

Alignment methods

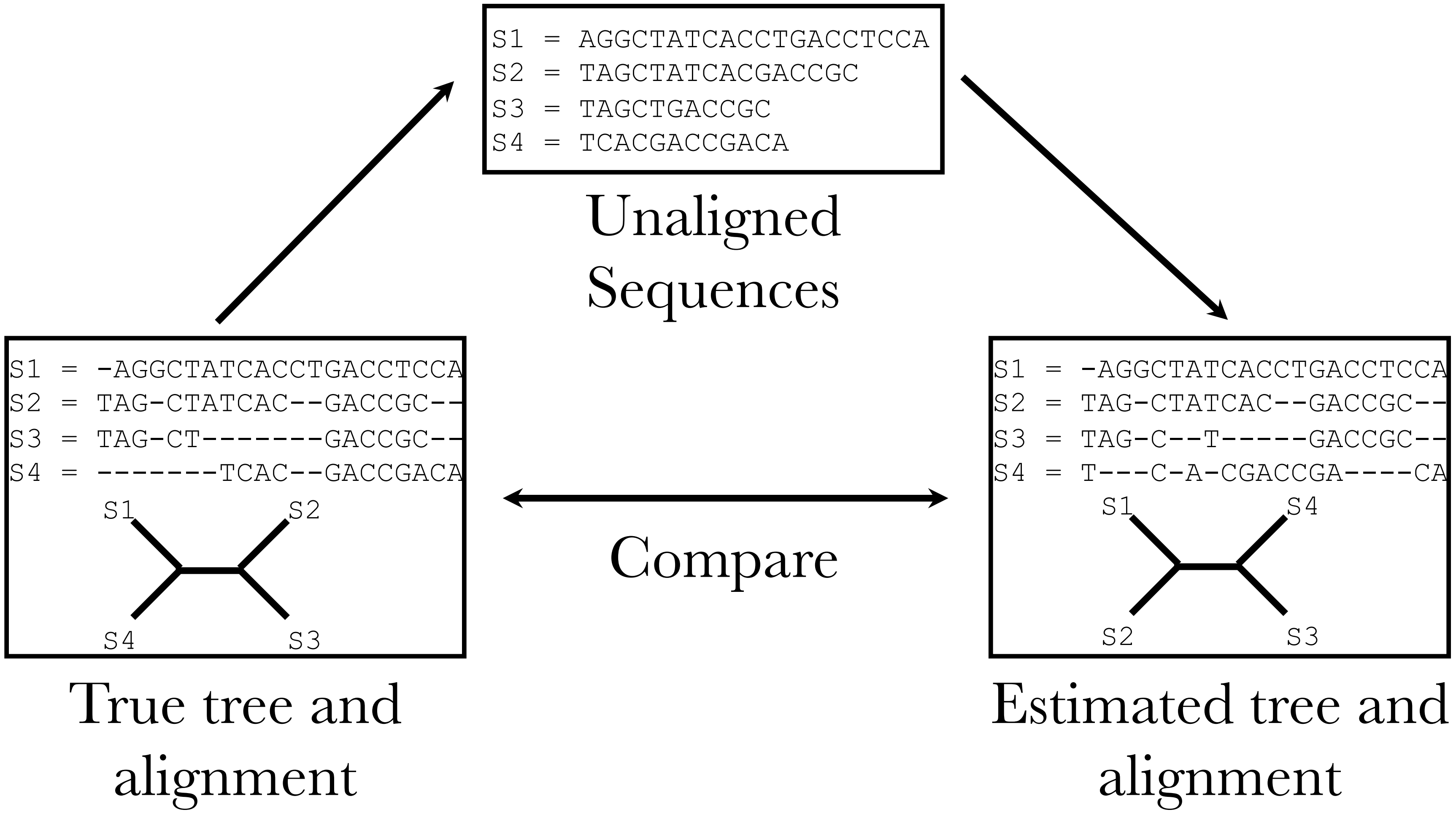
- Clustal
- POY (and POY*)
- Probcons (and Probtree)
- Probalign
- MAFFT
- Muscle
- Di-align
- T-Coffee
- Prank (PNAS 2005, Science 2008)
- Opal (ISMB and Bioinf. 2007)
- *FSA (PLoS Comp. Bio. 2009)*
- *Infernal (Bioinf. 2009)*
- Etc.

Phylogeny methods

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

RAxML*: heuristic for large-scale ML optimization*

Simulation Studies



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

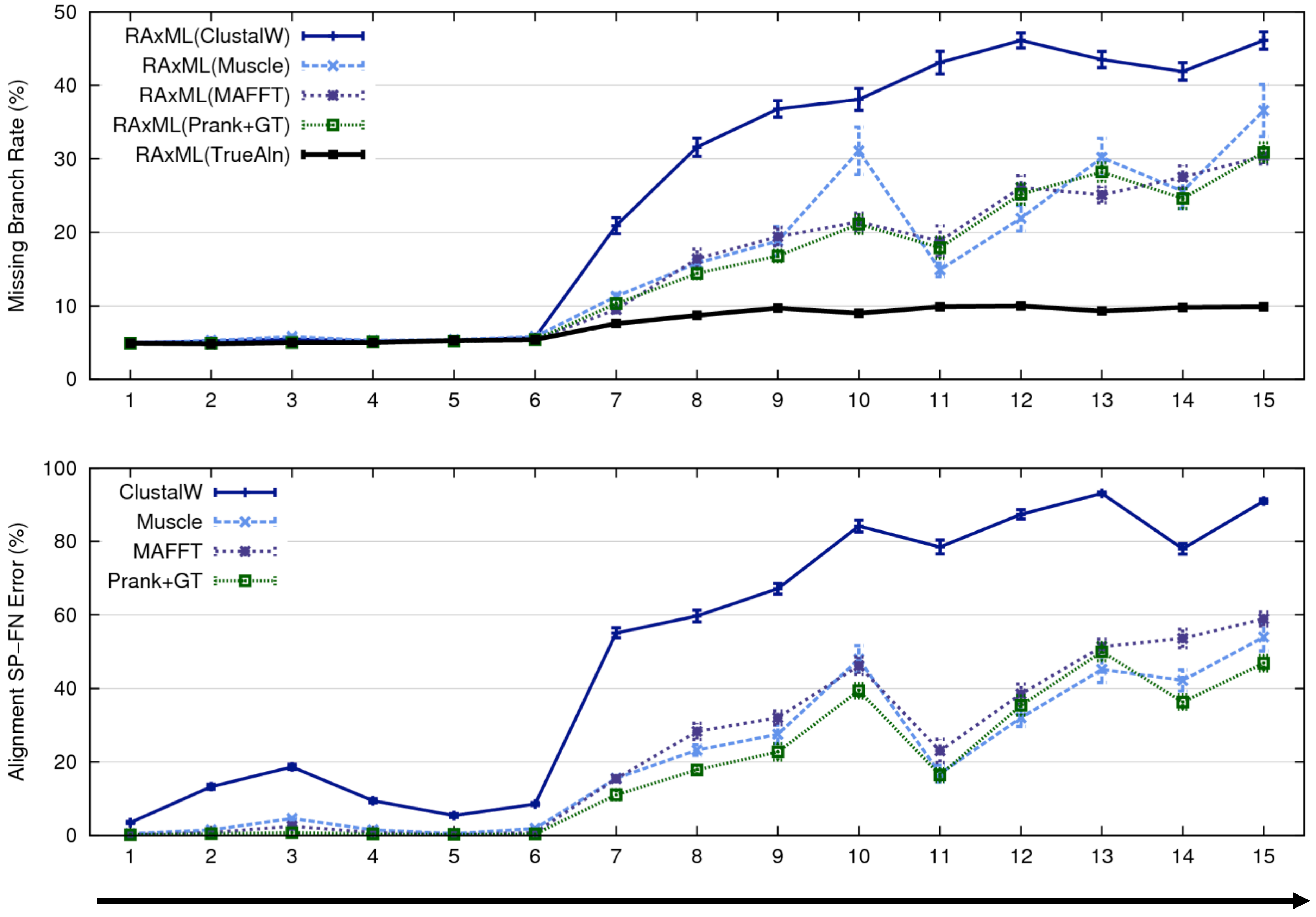
Unaligned Sequences

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

True tree and alignment

```
S1 = -AGGCTATCACCTGACCTCCA
S2 = TAG-CTATCAC--GACCGC--
S3 = TAG-C--T-----GACCGC--
S4 = T---C-A-CGACCGA----CA
```

Estimated tree and alignment



1000 taxon models, ordered by difficulty (Liu et al., 2009)

Problems

- Large datasets with high rates of evolution are hard to align accurately, and phylogeny estimation methods produce poor trees when alignments are poor.
- Many phylogeny estimation methods have poor accuracy on large datasets (even if given correct alignments)
- *Potentially useful genes are often discarded* if they are difficult to align.

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

Computational Phylogenetics

Courtesy of the Tree of Life project

Current methods can use months to estimate trees on 1000 DNA sequences

Our objective:

More accurate trees and alignments on 500,000 sequences in under a week

We prove theorems using graph theory and probability theory, and our algorithms are studied on real and simulated data.

(Some of) our Methods

- SATé, PASTA, and UPP: large, very large, and ultra-large multiple sequence alignment
- DACTAL and DCM: ultra-large tree estimation

- Techniques:
  - Divide-and-conquer
  - Iteration
  - Hidden Markov Models
  - Graph Theory

Divide-and-Conquer

- Divide-and-conquer is a basic algorithmic trick for solving problems!
- Basic idea: divide a dataset into two or more sets, solve the problem on each set, and then combine solutions.
- Example: MergeSort to sort a list of k integers!

Divide-and-Conquer

- MergeSort solves the “Sorting Problem”: given a list of integers, put them into increasing order (smallest to largest).

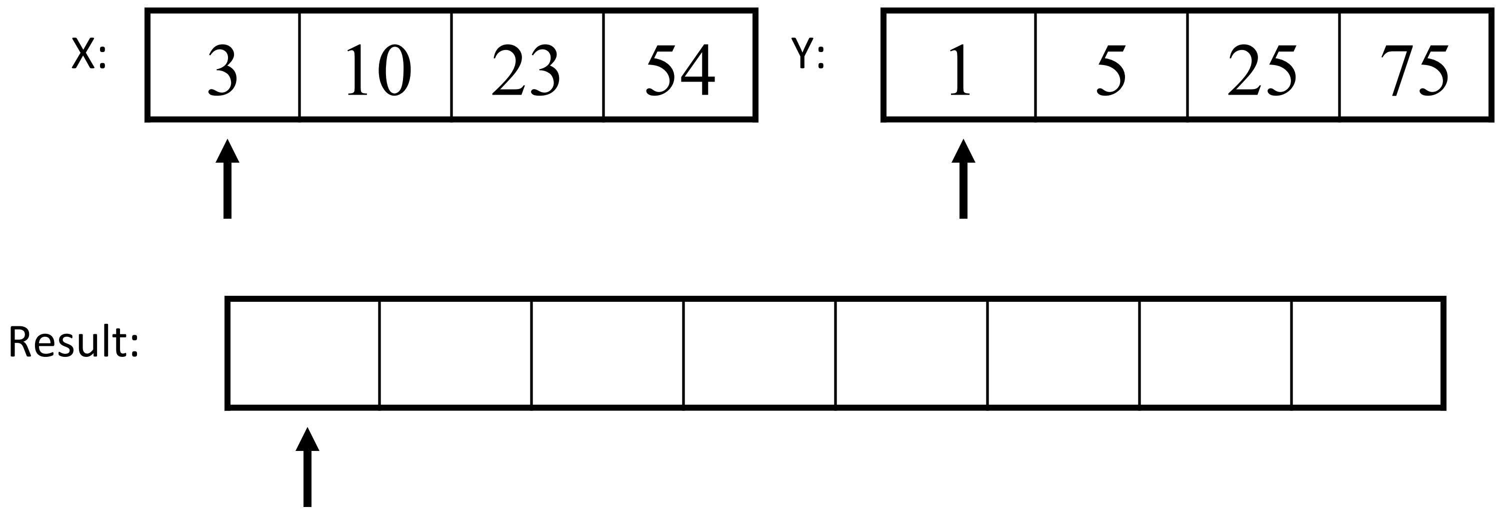
- This can be done by recursively dividing the unsorted list in half, applying MergeSort to each side, and then merging the right and left back together.

Merge Sort Algorithm

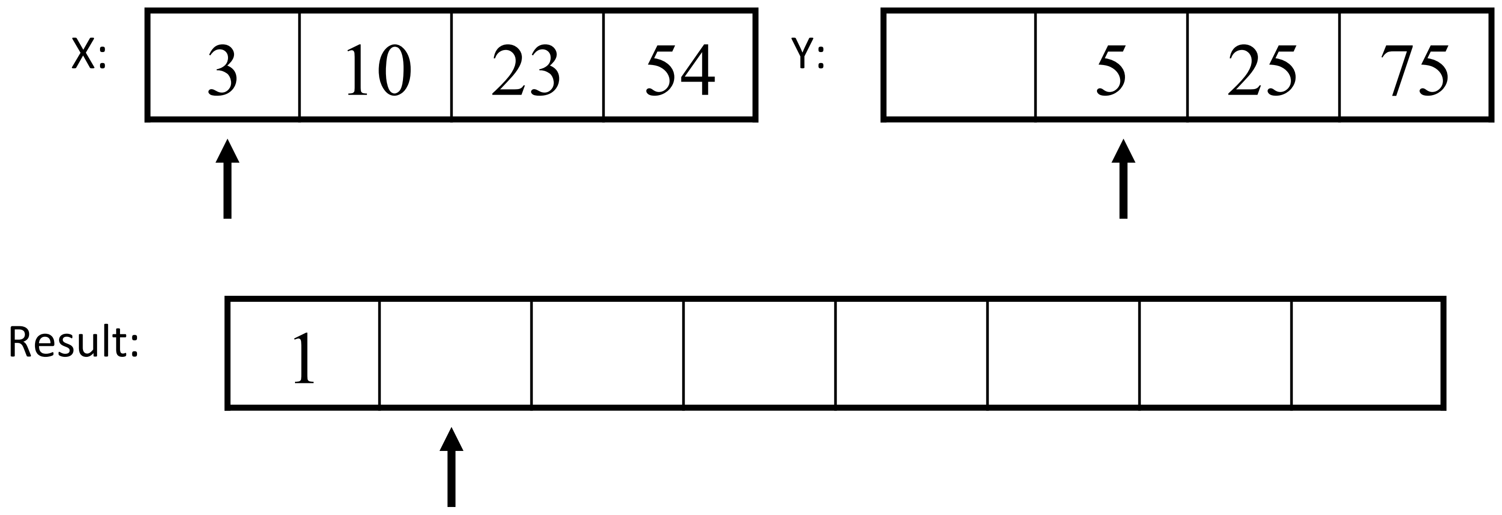
Given a list L with a length k:

- If k == 1 → the list is sorted
- Else:
  - Merge Sort the left side (0 through k/2)
  - Merge Sort the right side (k/2+1 thru k)
  - Merge the right side with the left side

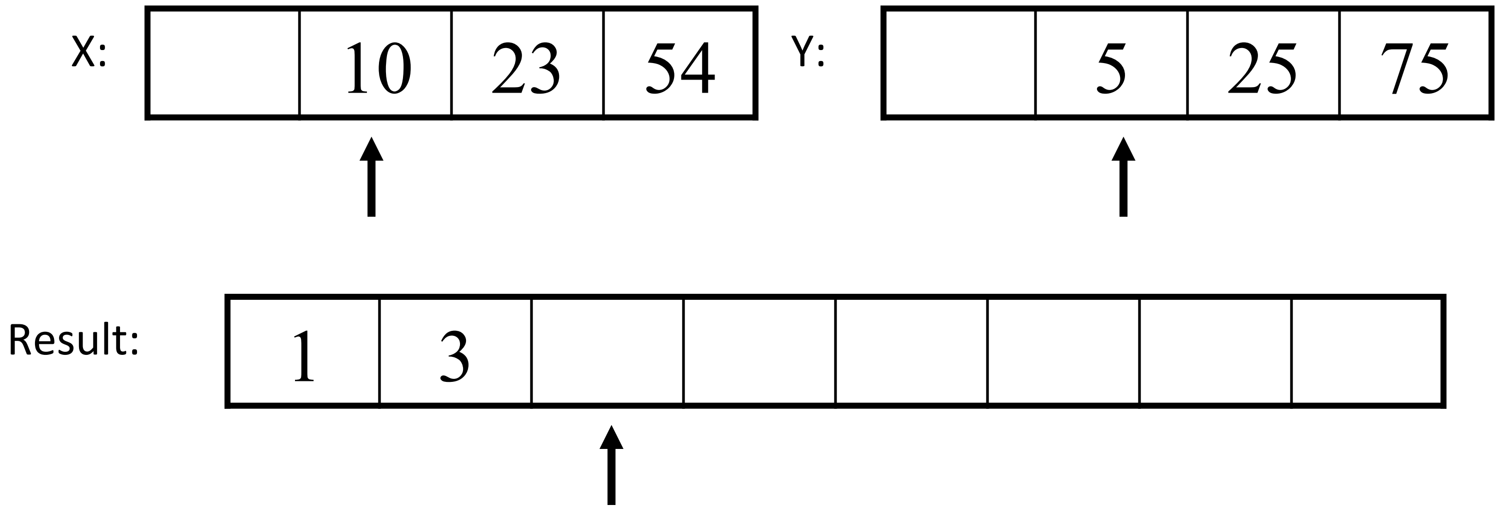
Merging two sorted lists



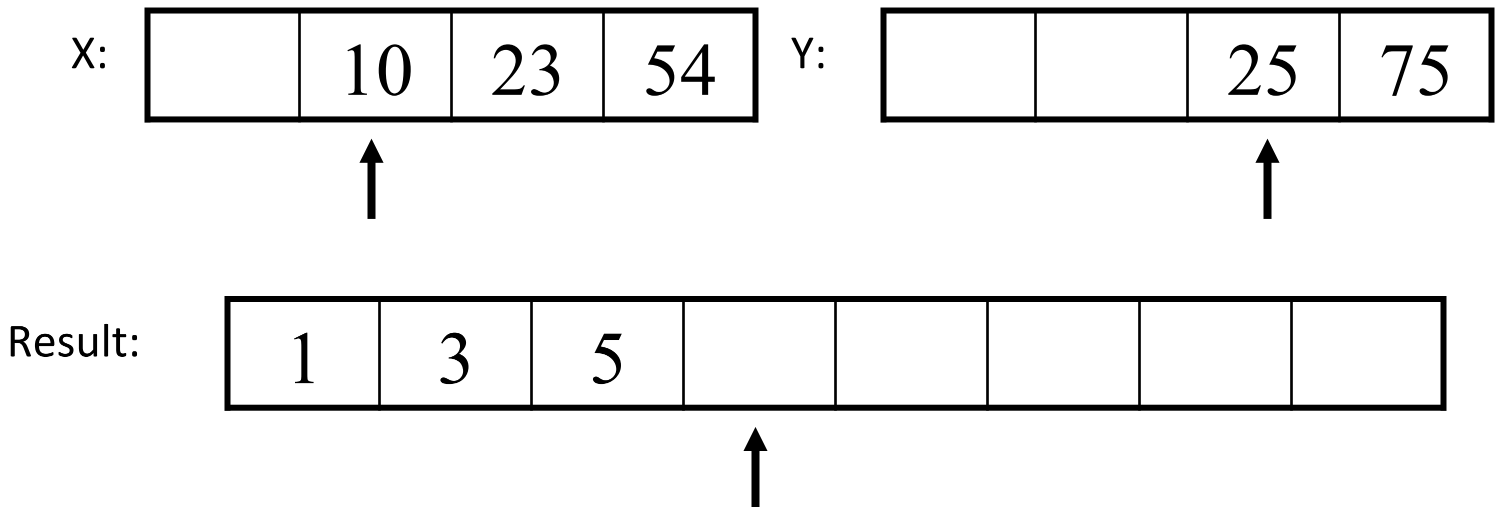
Merging (cont.)



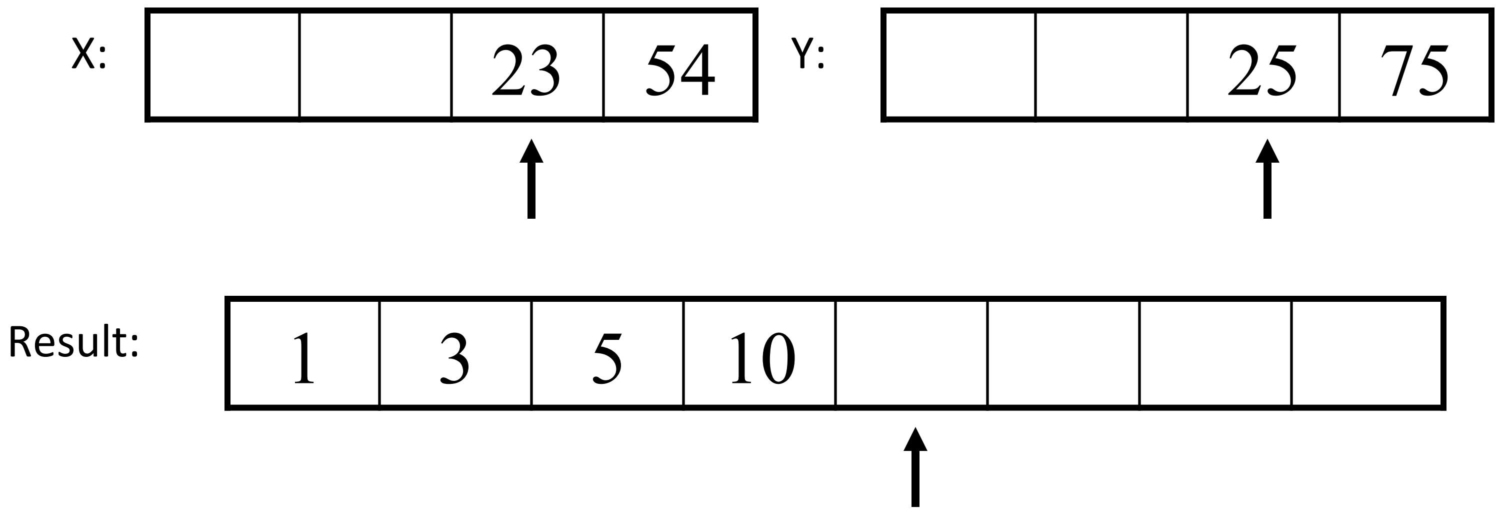
Merging (cont.)



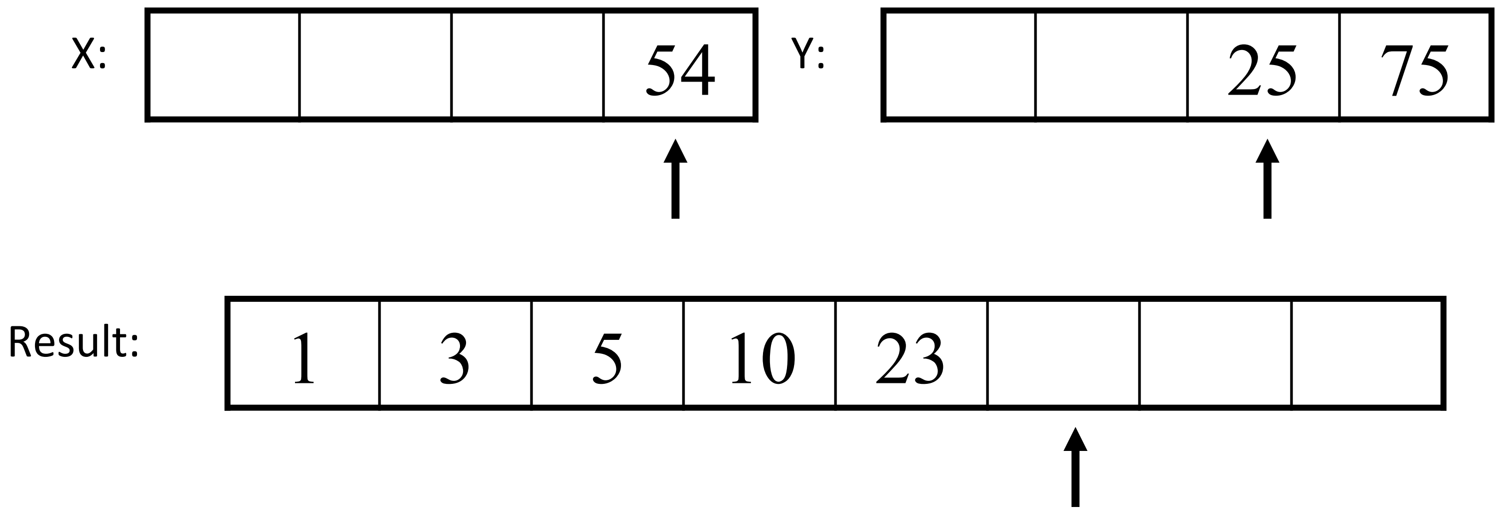
Merging (cont.)



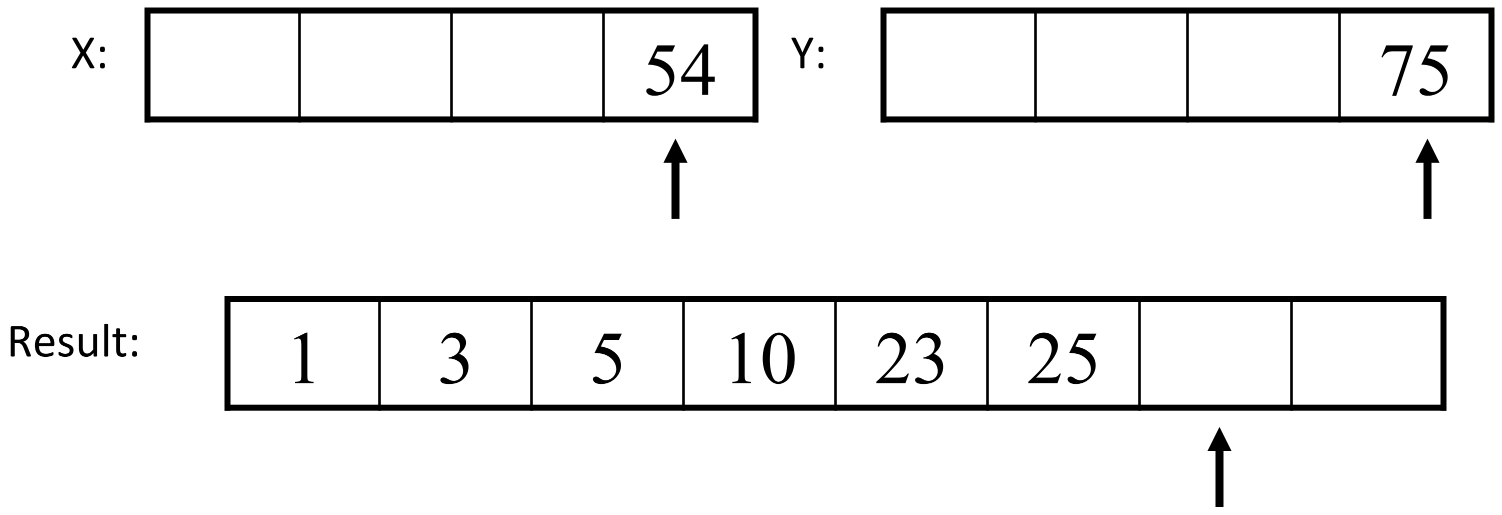
Merging (cont.)



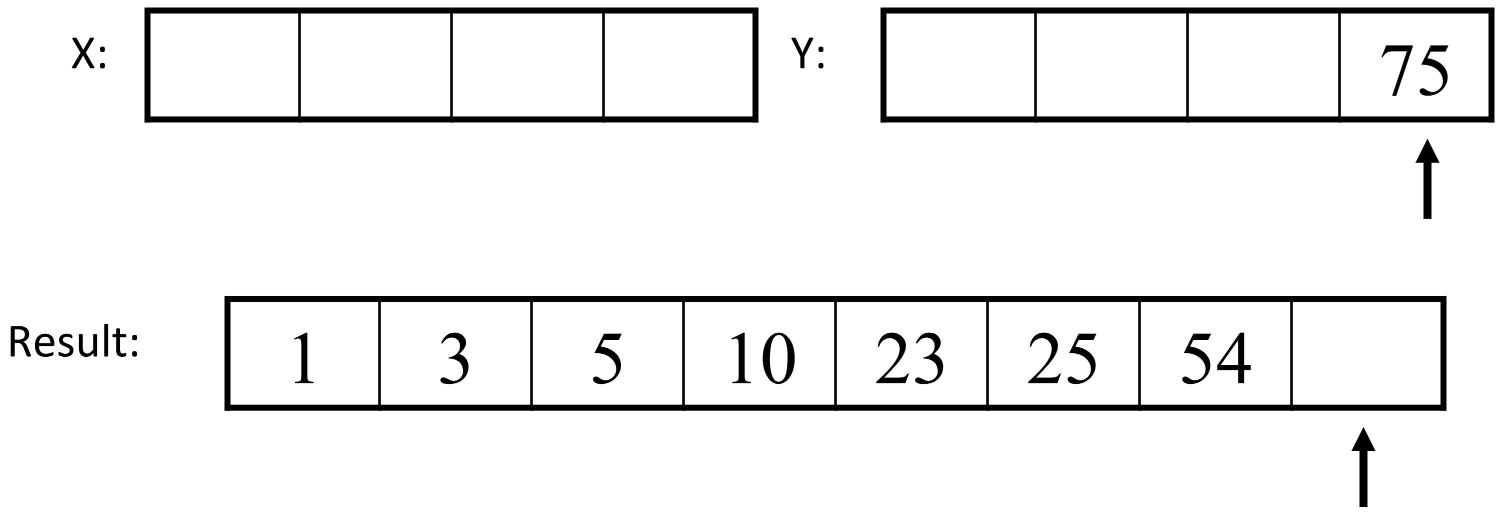
Merging (cont.)



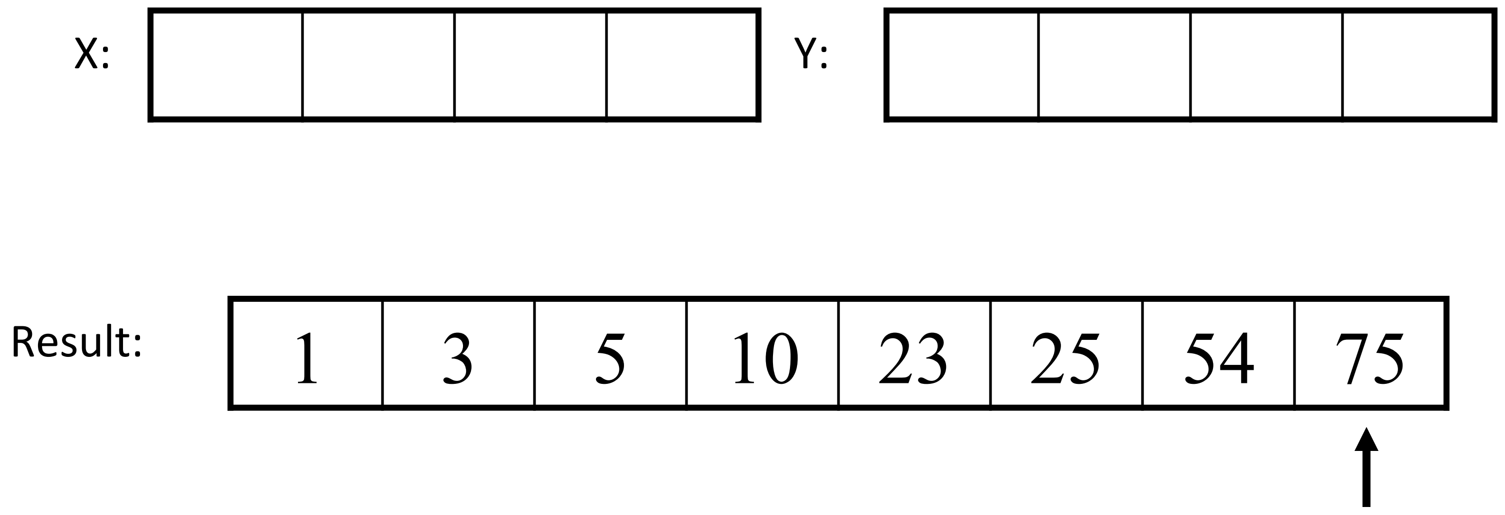
Merging (cont.)



Merging (cont.)



Merging (cont.)



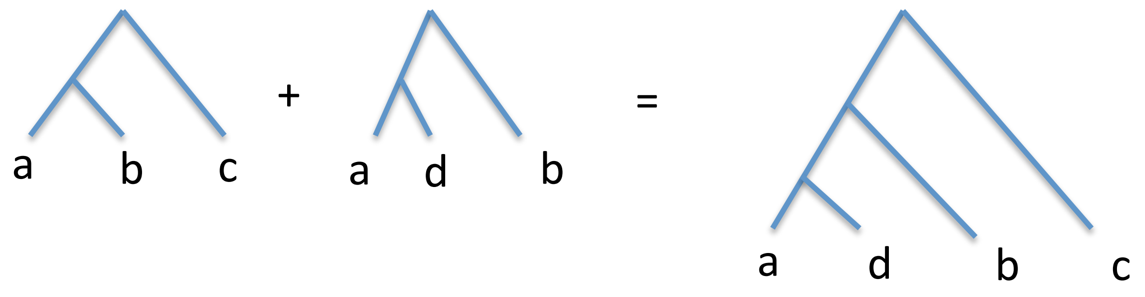
Divide-and-Conquer for Tree Estimation

- Basic idea: divide a dataset into two or more overlapping subsets, construct trees on each set, and then combine trees!

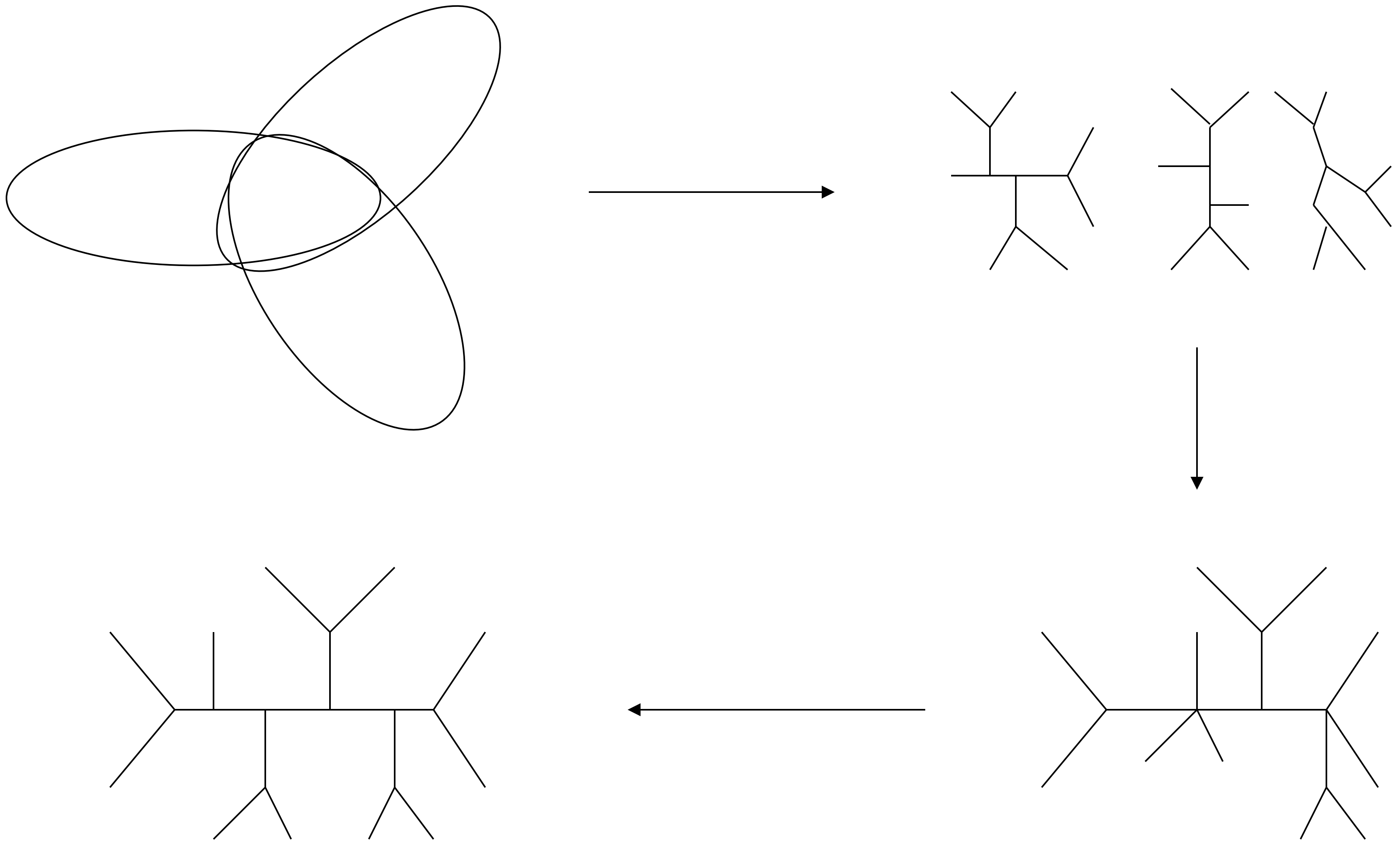
- Somewhat easy to figure out how to do this on rooted trees, but a bit harder on unrooted trees.

Constructing trees from smaller trees

- Rooted trees: If they agree, it's easy! (Not so easy if they disagree)
- Unrooted trees: NP-hard, even if they agree!



DCMs: Divide-and-conquer for improving phylogeny reconstruction



Four Boosters

- DCM1: improves accuracy of distance-based methods using divide-and-conquer and chordal graph theory
- DACTAL: uses divide-and-conquer plus iteration to get a very large tree without needing a multiple sequence alignment

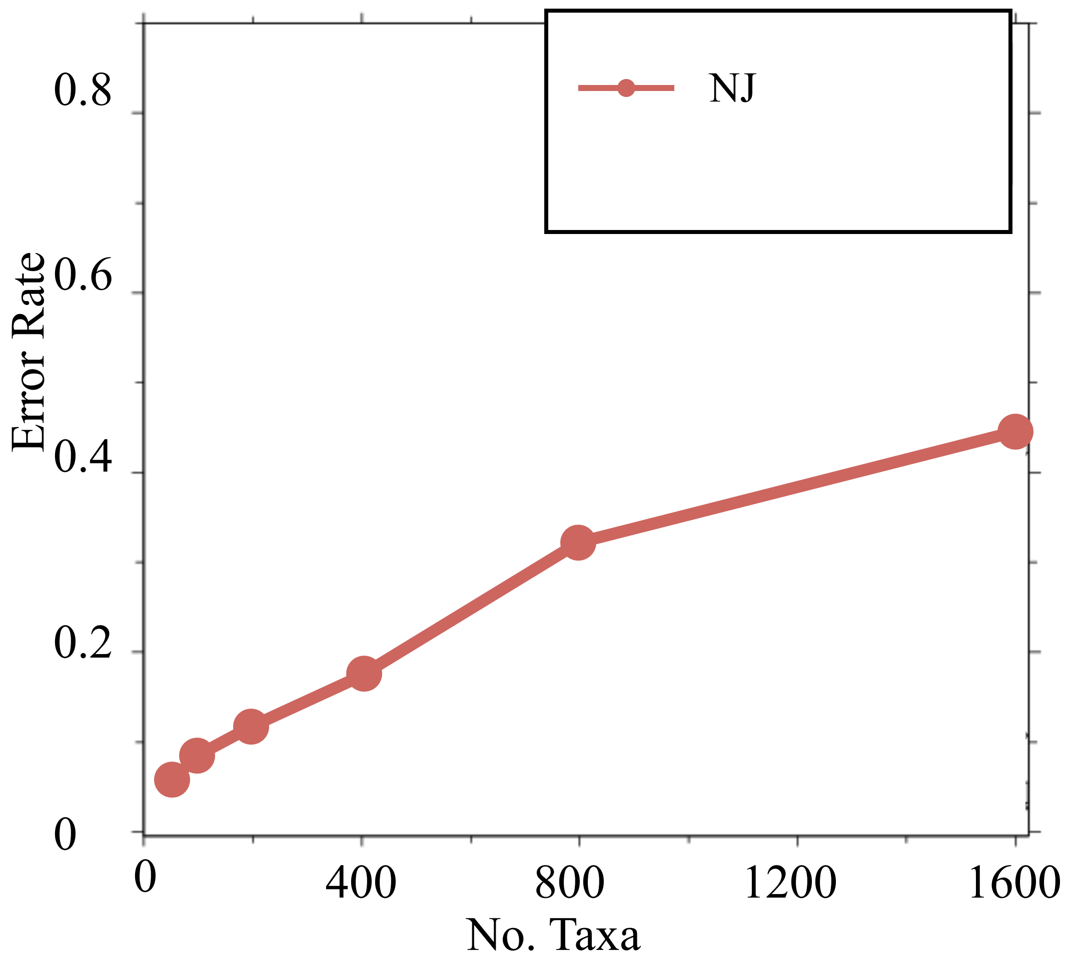
- SATé: uses divide-and-conquer plus iteration to co-estimate the multiple sequence alignment and a tree
- UPP: uses divide-and-conquer, randomization, and Hidden Markov Models to obtain ultra-large alignments

First example: DCM1

DCM = “Disk Covering Method”

- Theory: Warnow, St. John, and Moret, SODA 2001,
- Practice:Nakhleh et al., ISMB 2001

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]

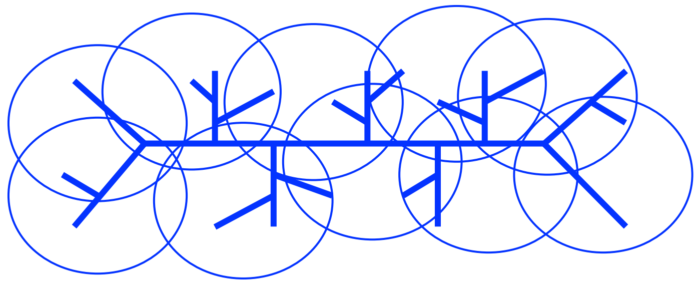
DCM1 Decompositions

Input: Distance matrix (distances between species) and threshold q.

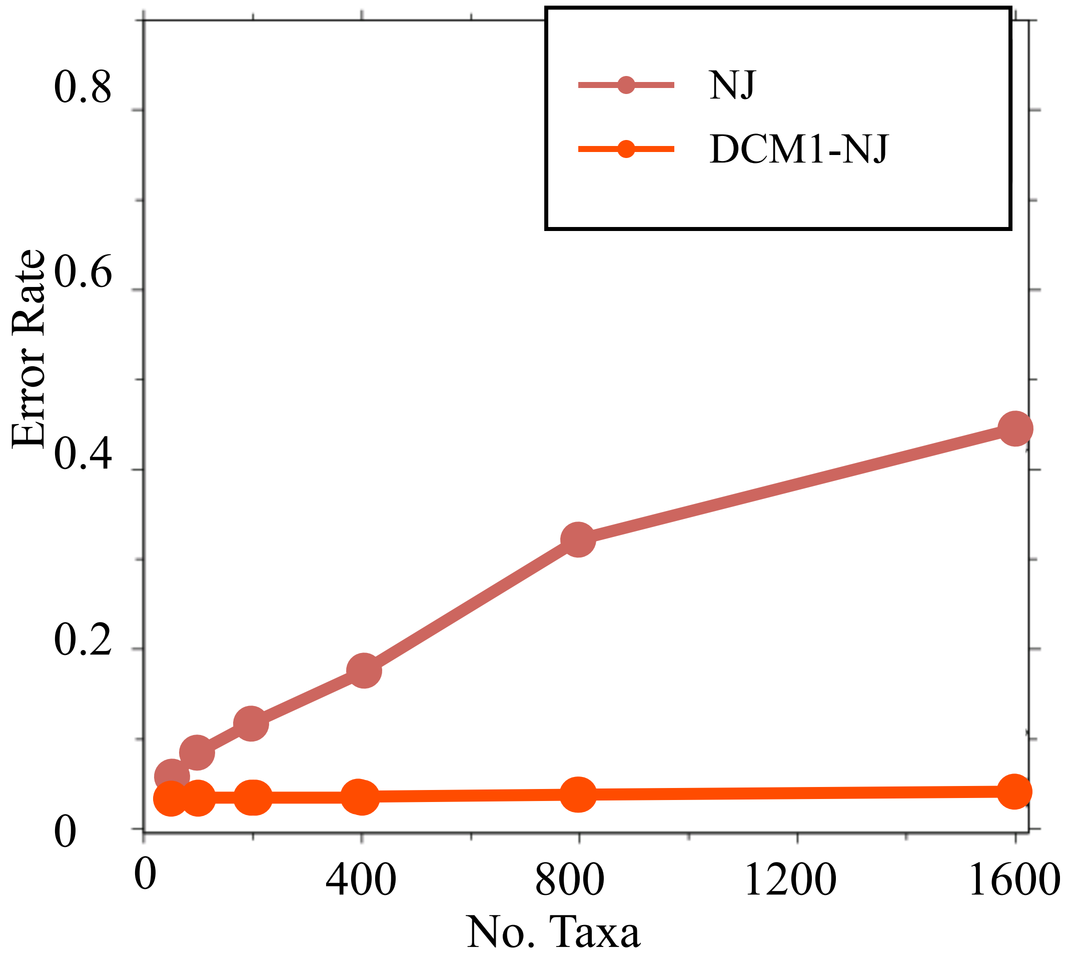
Output: Division of the set of species into *overlapping subsets*.

Technique: Compute maximal cliques in a “Triangulated Threshold Graph”.

Looks like moving a disk across a tree!



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

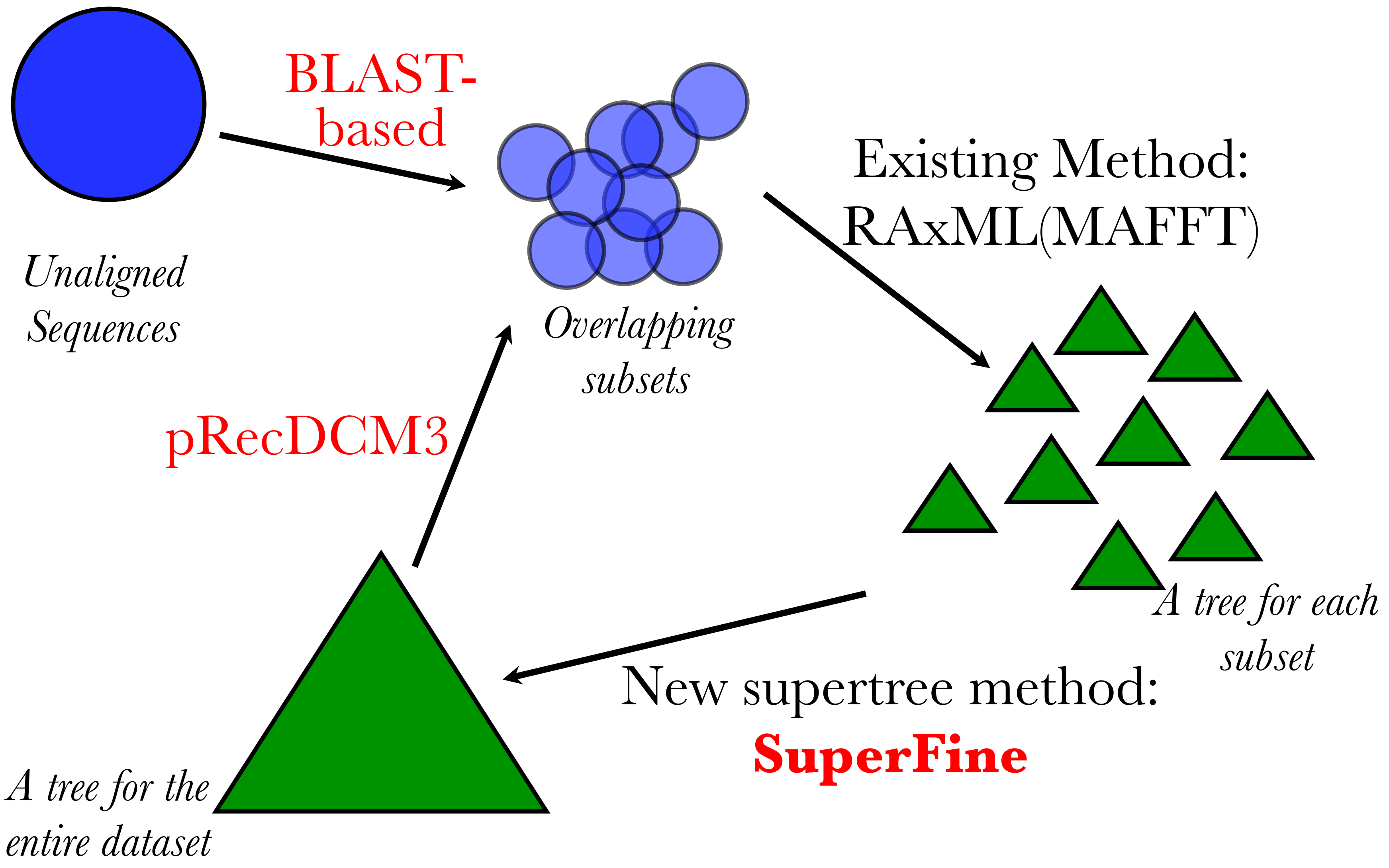
Second Example: DACTAL

(Divide-And-Conquer Trees (Almost) without alignments)

- Input: set S of unaligned sequences
- Output: tree on S (but no alignment)

Nelesen, Liu, Wang, Linder, and Warnow, ISMB and Bioinformatics 2012

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 3 Largest CRW Datasets

CRW: Comparative RNA database,

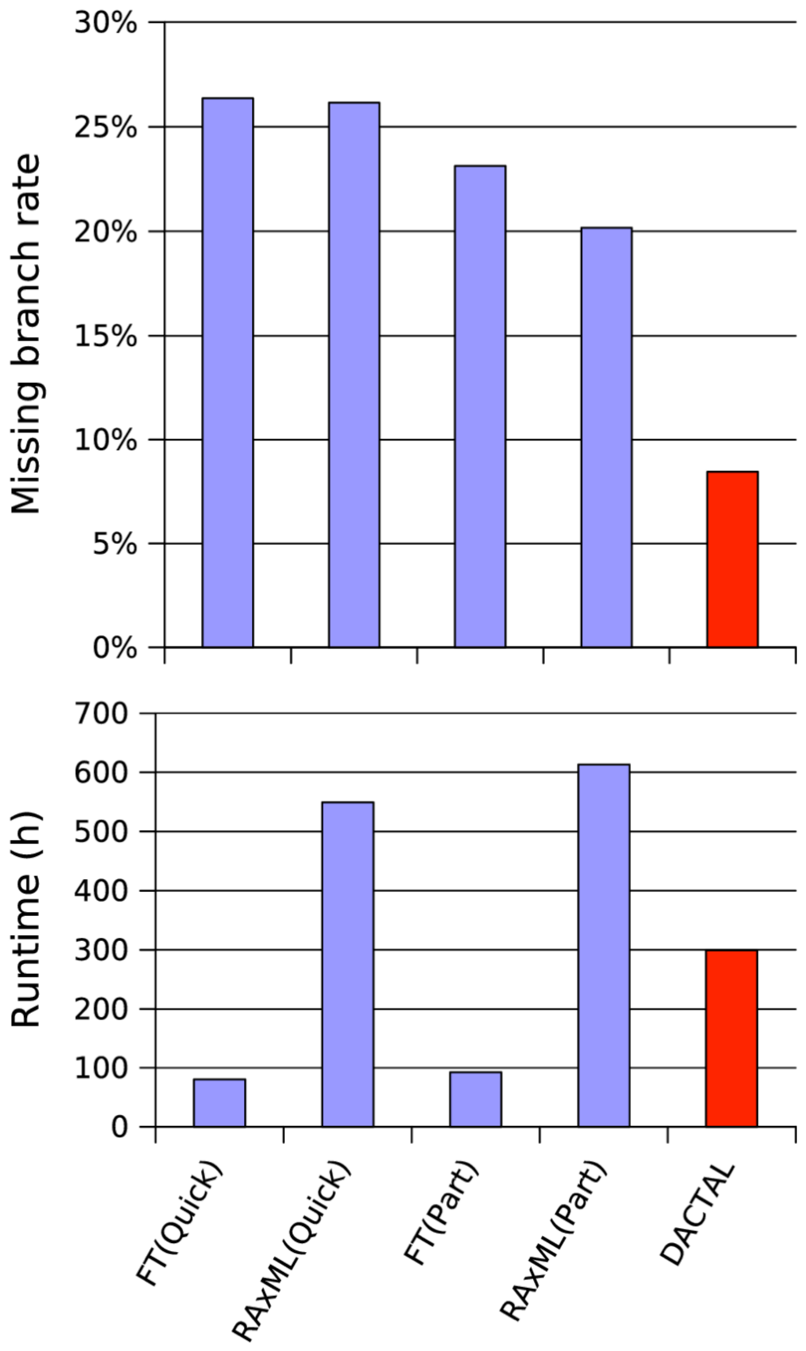
Three 16S datasets with 6,323 to 27,643 sequences

Reference alignments based on secondary structure

Reference trees are 75% RAxML bootstrap trees

DACTAL (shown in red) run for 5 iterations starting from FT(Part)

FastTree (FT) and RAxML are ML methods



Divide-and-Conquer for Alignment

- SATé: uses <u>divide-and-conquer </u>plus iteration to co-estimate the multiple sequence alignment and a tree
- UPP: <u>uses divide-and-conquer</u>, randomization, and Hidden Markov Models to obtain ultra-large alignments

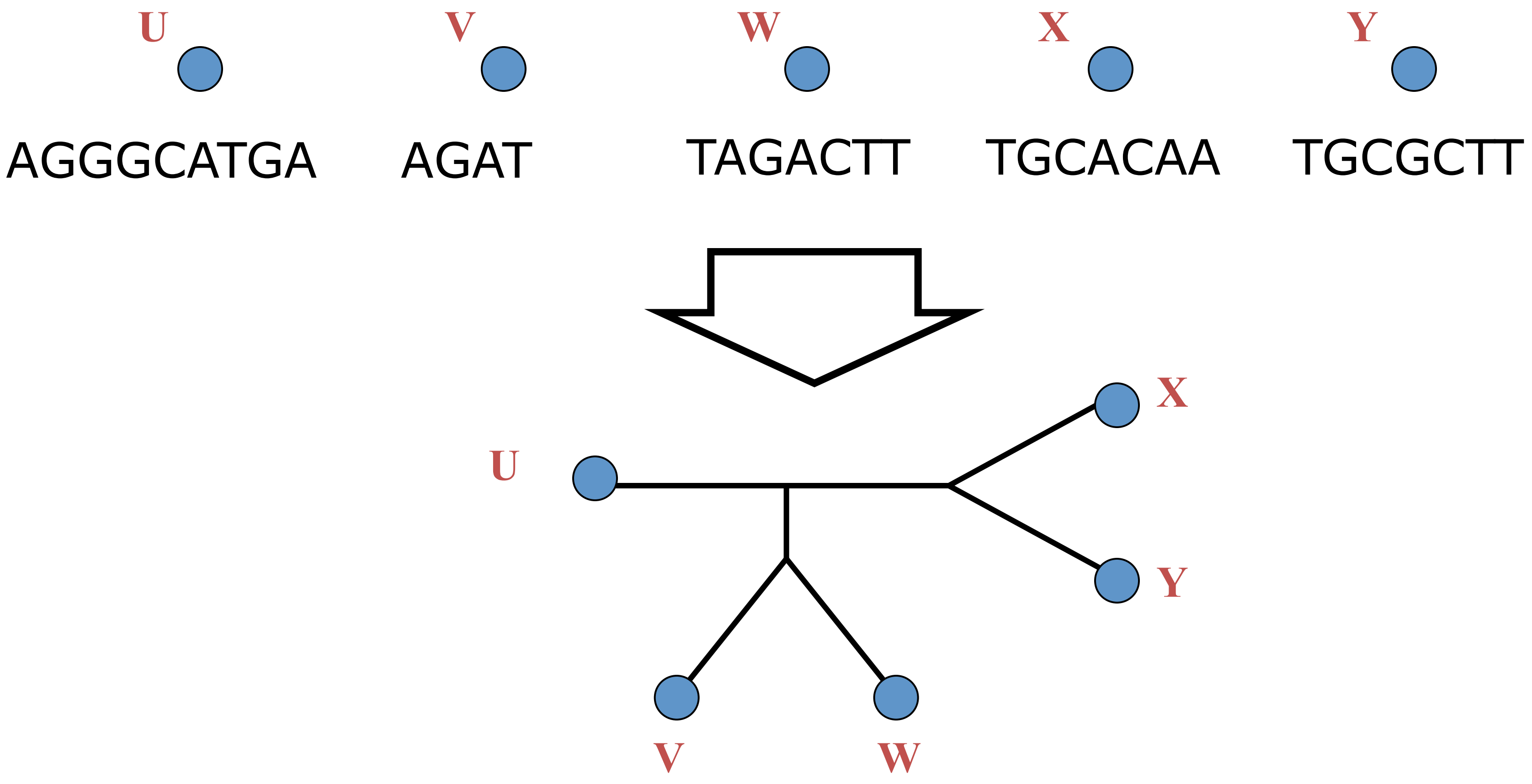
Example 3: SATé

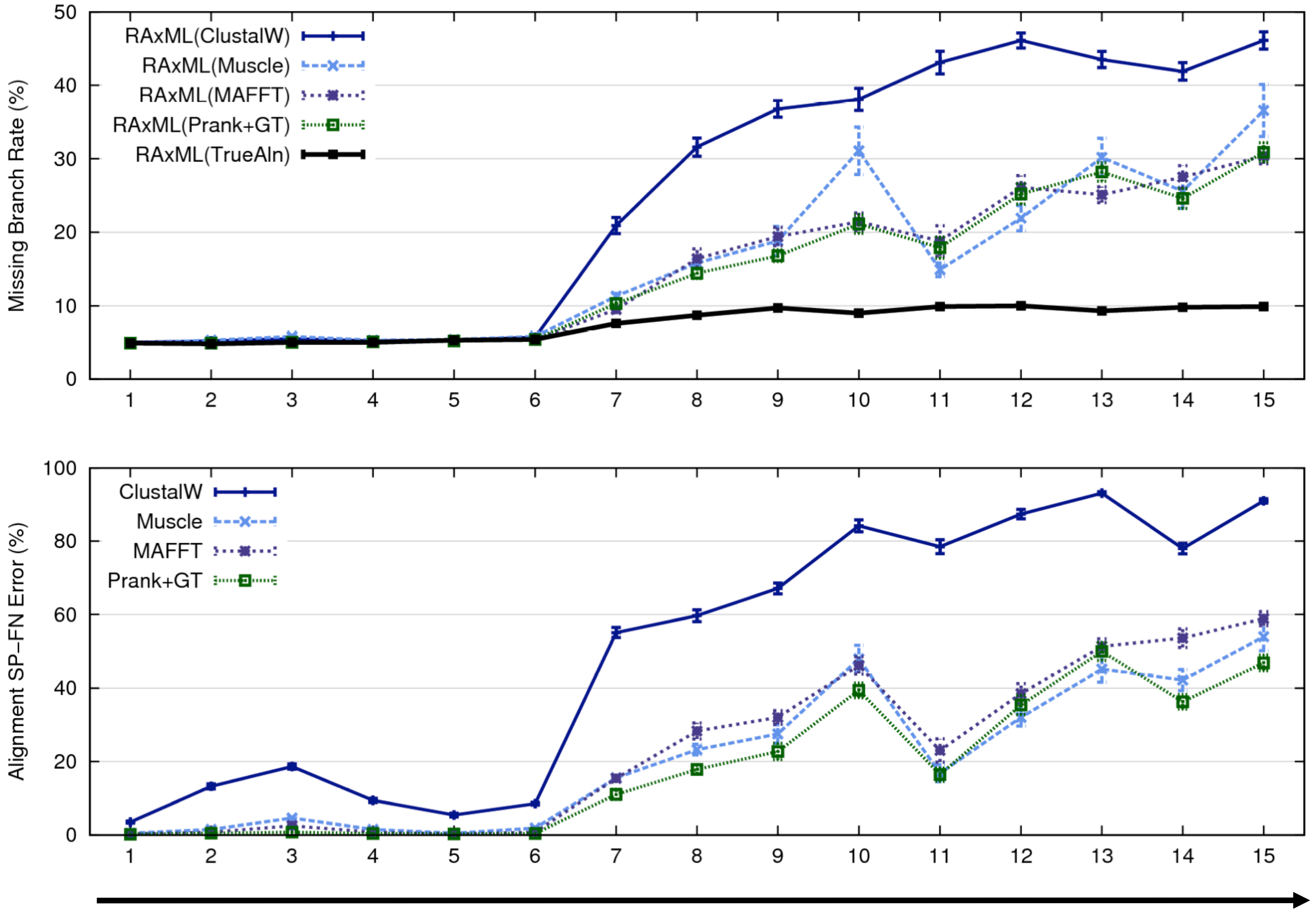
SATé: Simultaneous Alignment and Tree Estimation

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

Liu et al., Systematic Biology 2012

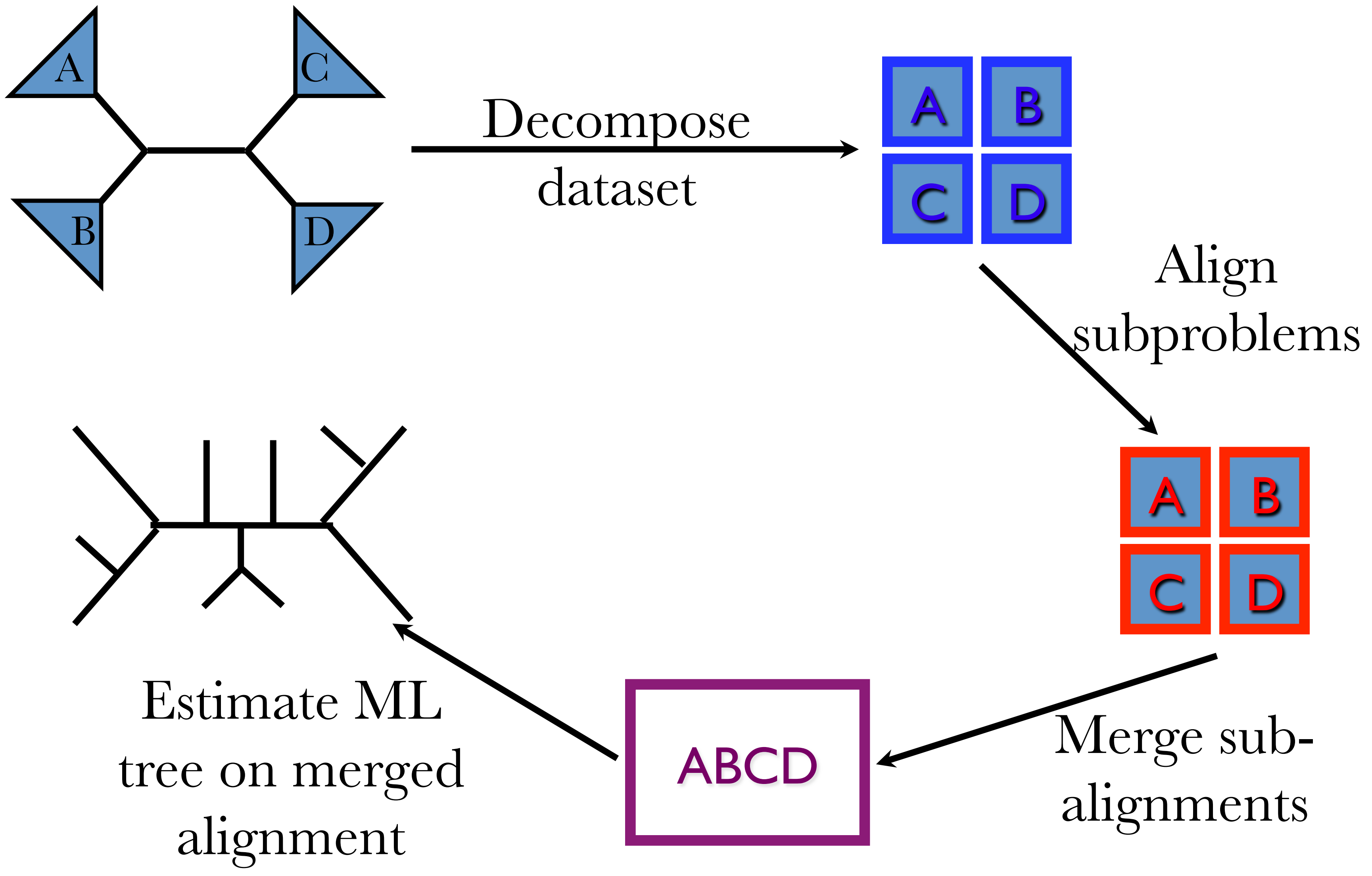
Public software distribution (open source) through Mark Holder's group at the University of Kansas


U
V
W
X
Y
AGGGCATGA
AGAT
TAGACTT
TGCACAA
TGCGCTT
U
X
Y
V
W



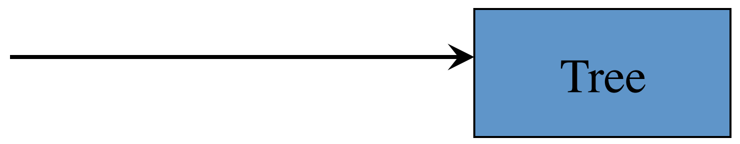
1000 taxon models, ordered by difficulty (Liu et al., 2009)

Re-aligning on a tree

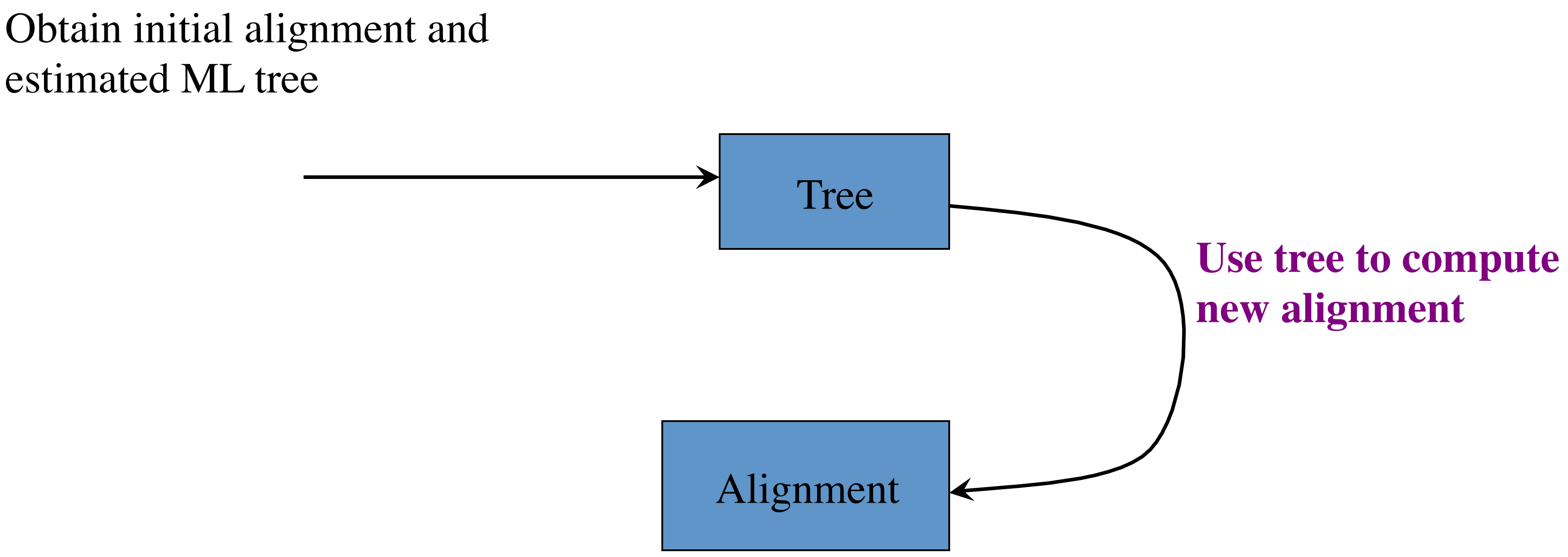


SATé Algorithm

Obtain initial alignment and
estimated ML tree



SATé Algorithm

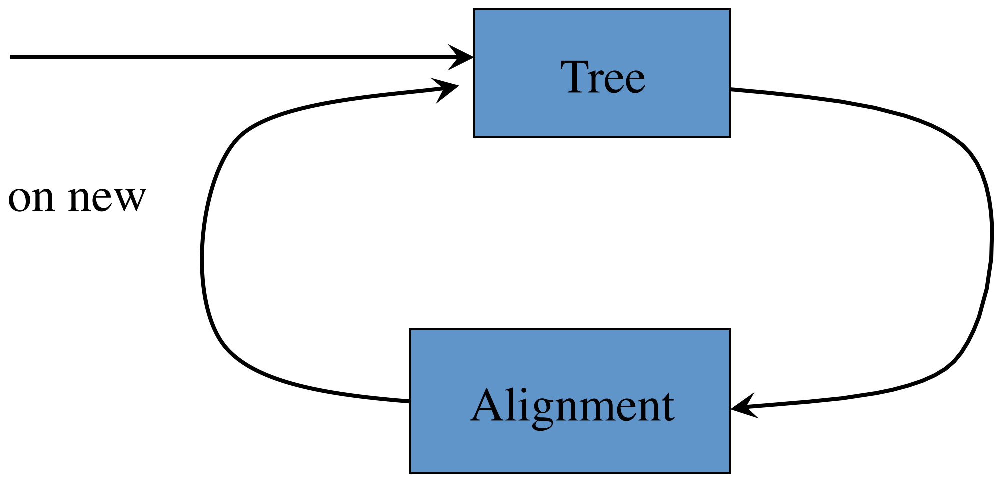

Obtain initial alignment and estimated ML tree
Tree
Use tree to compute new alignment
Alignment

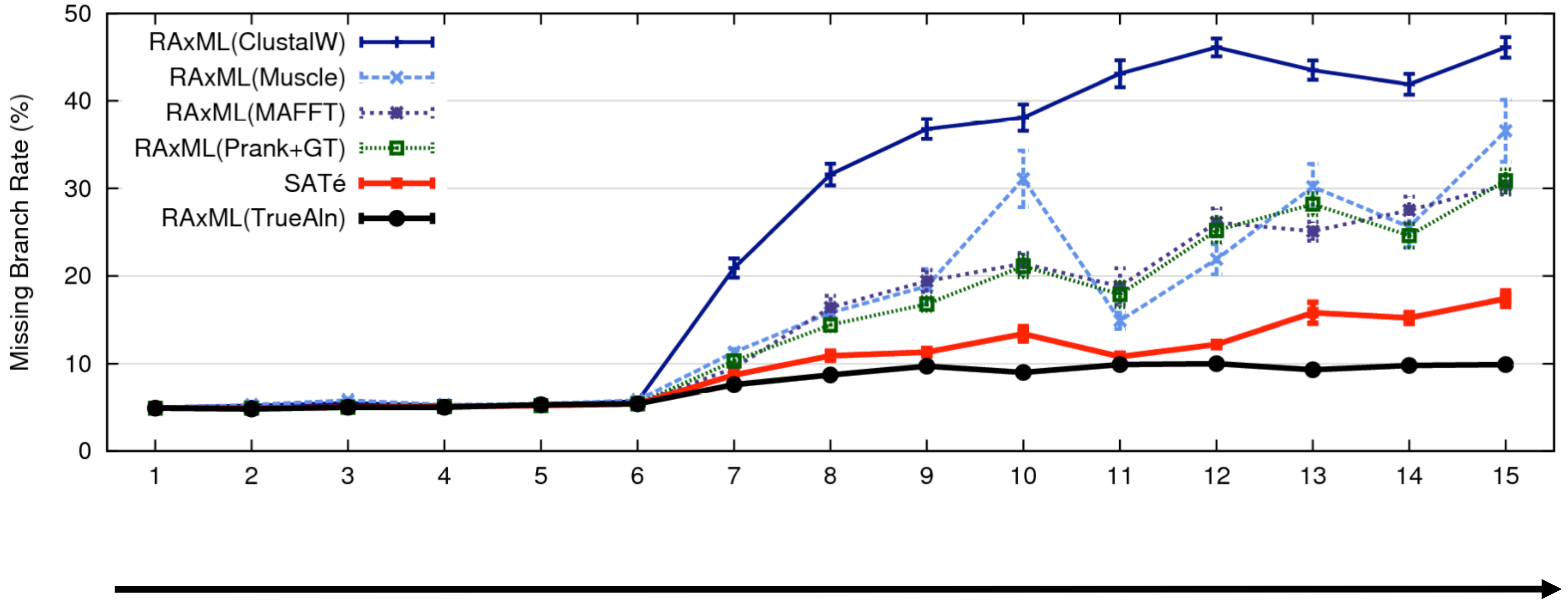
SATé Algorithm

Obtain initial alignment and estimated ML tree

Estimate ML tree on new alignment

Use tree to compute new alignment

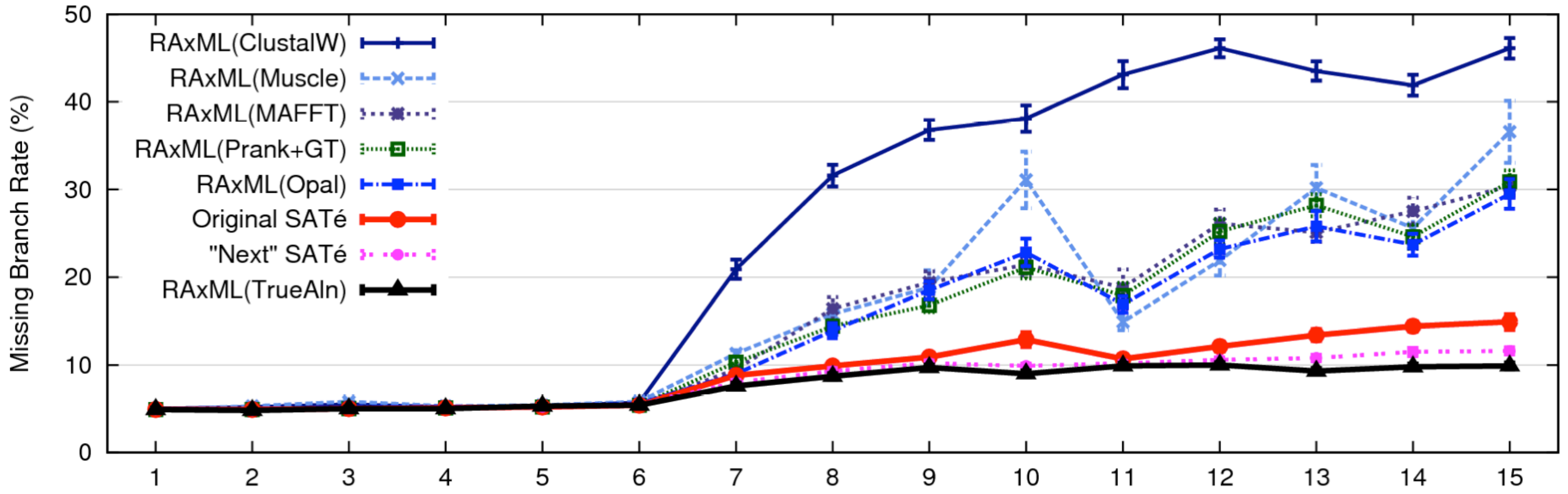


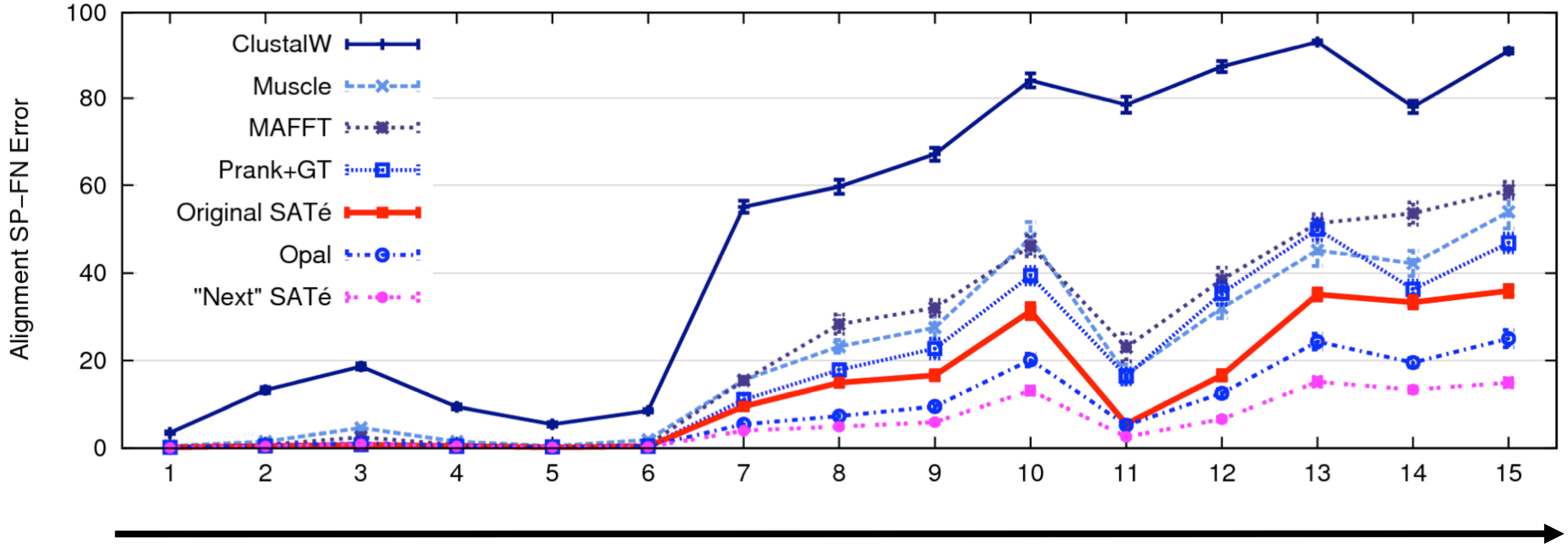


1000 taxon models, ordered by difficulty

24 hour SATé analysis, on desktop machines

(Similar improvements for biological datasets)





1000 taxon models ranked by difficulty

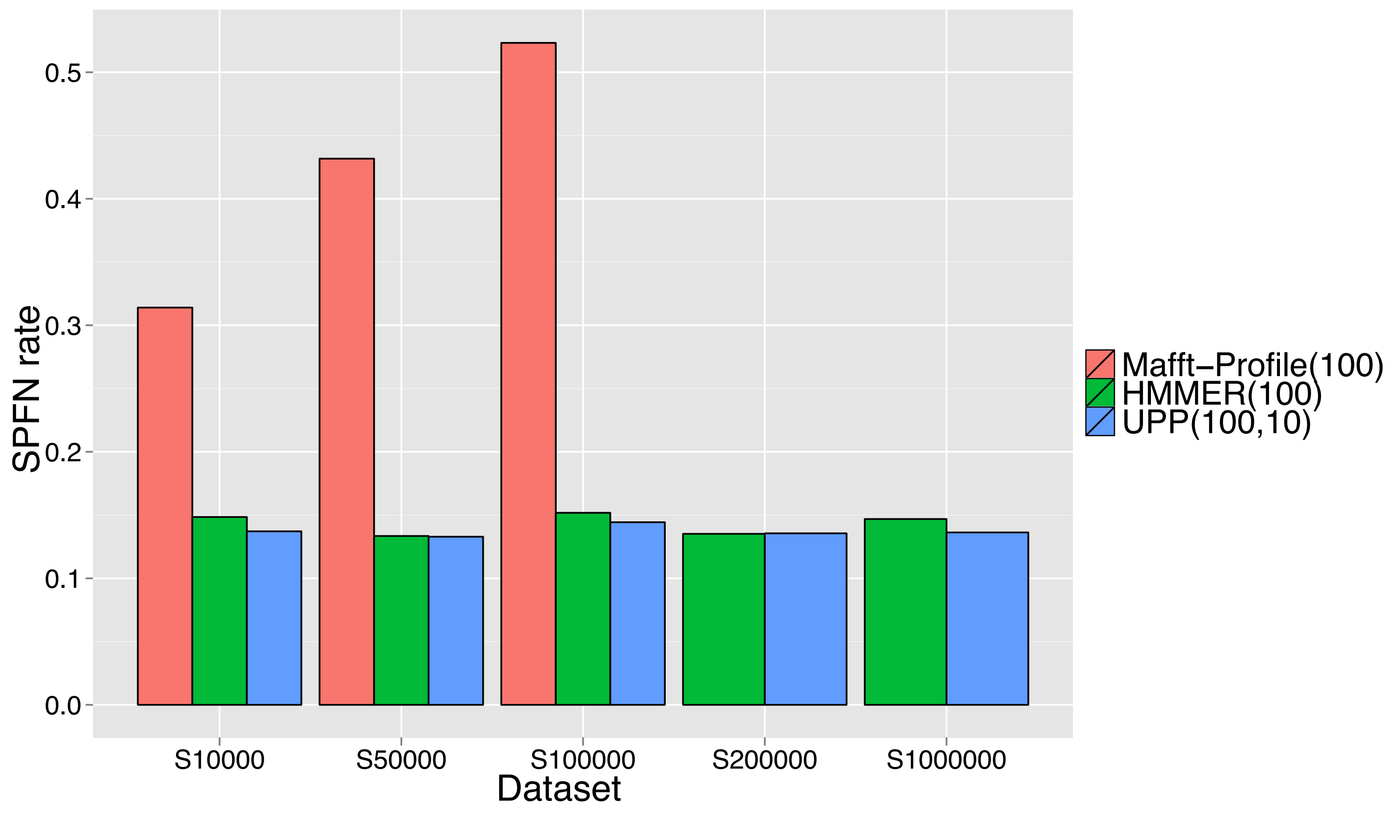
Example 4: UPP

- UPP: Ultra-large alignments using SEPP
- Authors: Nam Nguyen, Siavash Mirarab, and Tandy Warnow

Basic idea:

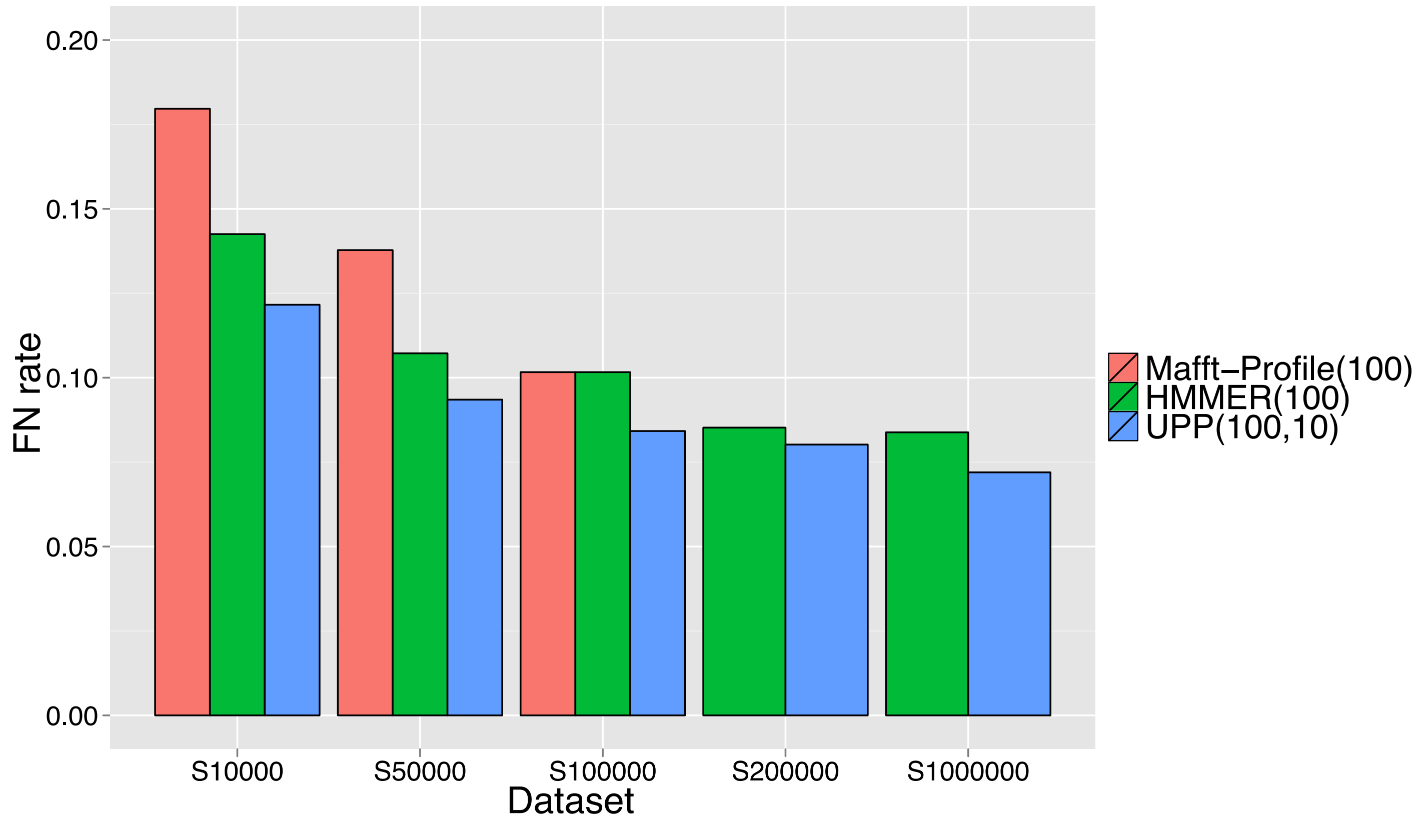
- estimate an alignment on a small random subset of the sequence dataset
- add all the remaining sequences into the small alignment (one by one, independently), using multiple Hidden Markov Models

UPP vs. HMMER vs. MAFFT (alignment error)



MAFFT-profile alignment strategy not as accurate as UPP(100,10) or UPP(100,100).

UPP vs. HMMER vs. MAFFT (tree error)



ML on UPP(100,10) and UPP(100,100) alignments both produce produce better trees than MAFFT.
Decomposition into a family of HMMs improves resultant trees.

Four Boosters

- DCM1: improves accuracy of distance-based methods using divide-and-conquer and chordal graph theory
- DACTAL: uses divide-and-conquer plus iteration to get a very large tree without needing a multiple sequence alignment
- SATé: uses divide-and-conquer plus iteration to co-estimate the multiple sequence alignment and a tree
- UPP: uses divide-and-conquer, randomization, and Hidden Markov Models to obtain ultra-large alignments

“Boosters”, or “Meta-Methods”

- Meta-methods use divide-and-conquer and iteration (or other techniques) to “boost” the performance of base methods (phylogeny reconstruction, alignment estimation, etc)

$$\begin{tikzcd}\text{Base method M} \arrow[r] & \boxed{\text{Meta-method}} \arrow[r] & \text{M*}\end{tikzcd}$$

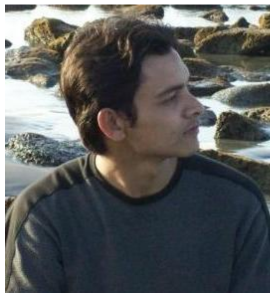
Summary

- Standard alignment and phylogeny estimation methods do not provide adequate accuracy on large datasets.
- When markers tend to yield poor alignments and trees, don't throw out the data *– get a better method!*

Summary

- Computer scientists develop algorithms and software to make it possible for scientists to get improved accuracy in their analyses.
- These algorithms involve creative strategies, including divide-and-conquer, iteration, and randomization.
- Extensive simulations and data analyses are part of the evaluation process!

Warnow Laboratory



PhD students: Siavash Mirarab[1], Nam Nguyen, and Md. S. Bayzid[2]
Undergrad: Keerthana Kumar
Lab Website: http://www.cs.utexas.edu/users/phylo

Funding: Guggenheim Foundation, Packard Foundation, NSF, Microsoft Research New England, David Bruton Jr. Centennial Professorship, and TACC (Texas Advanced Computing Center)

[1]HHMI International Predoctoral Fellow, [2]Fulbright Predoctoral Fellow

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