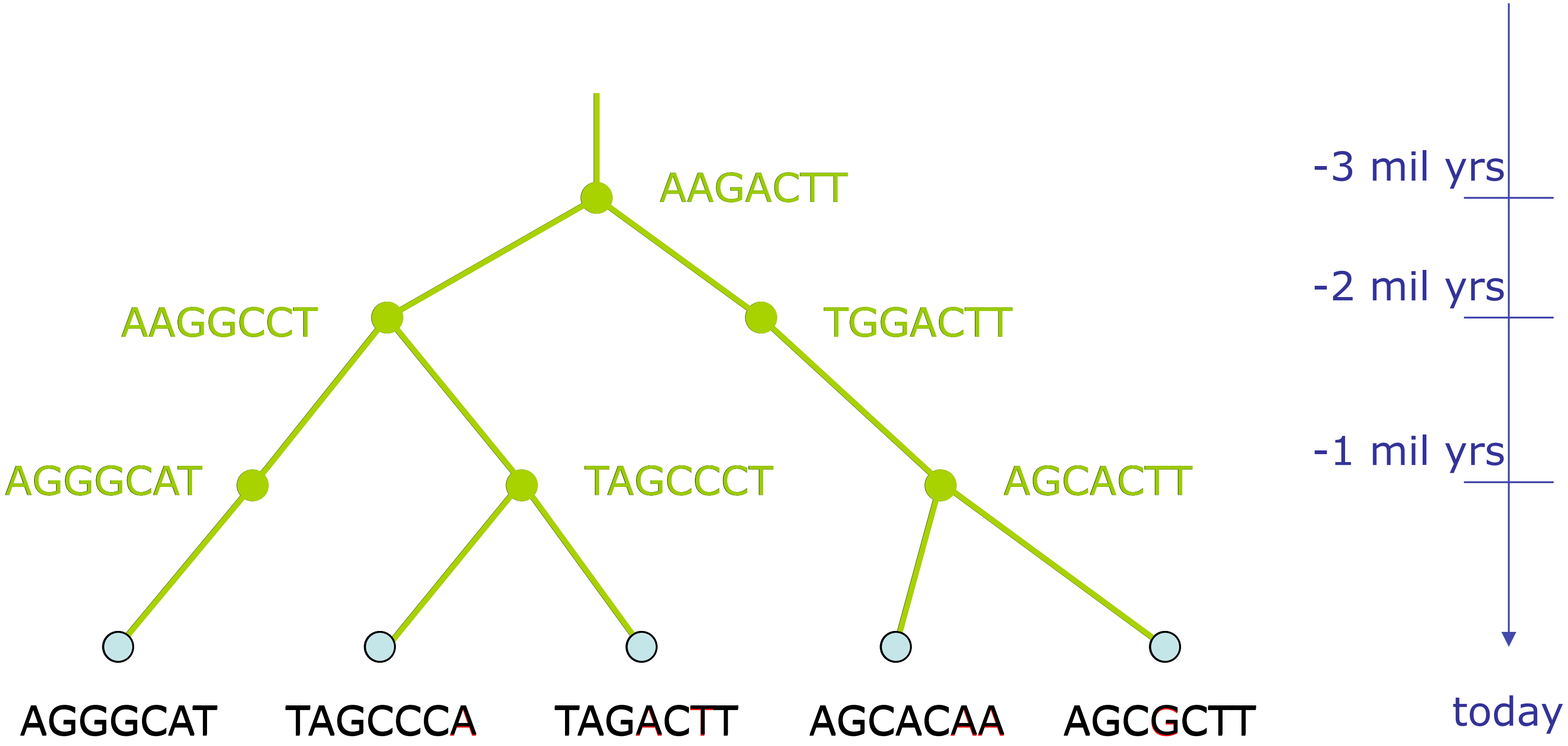
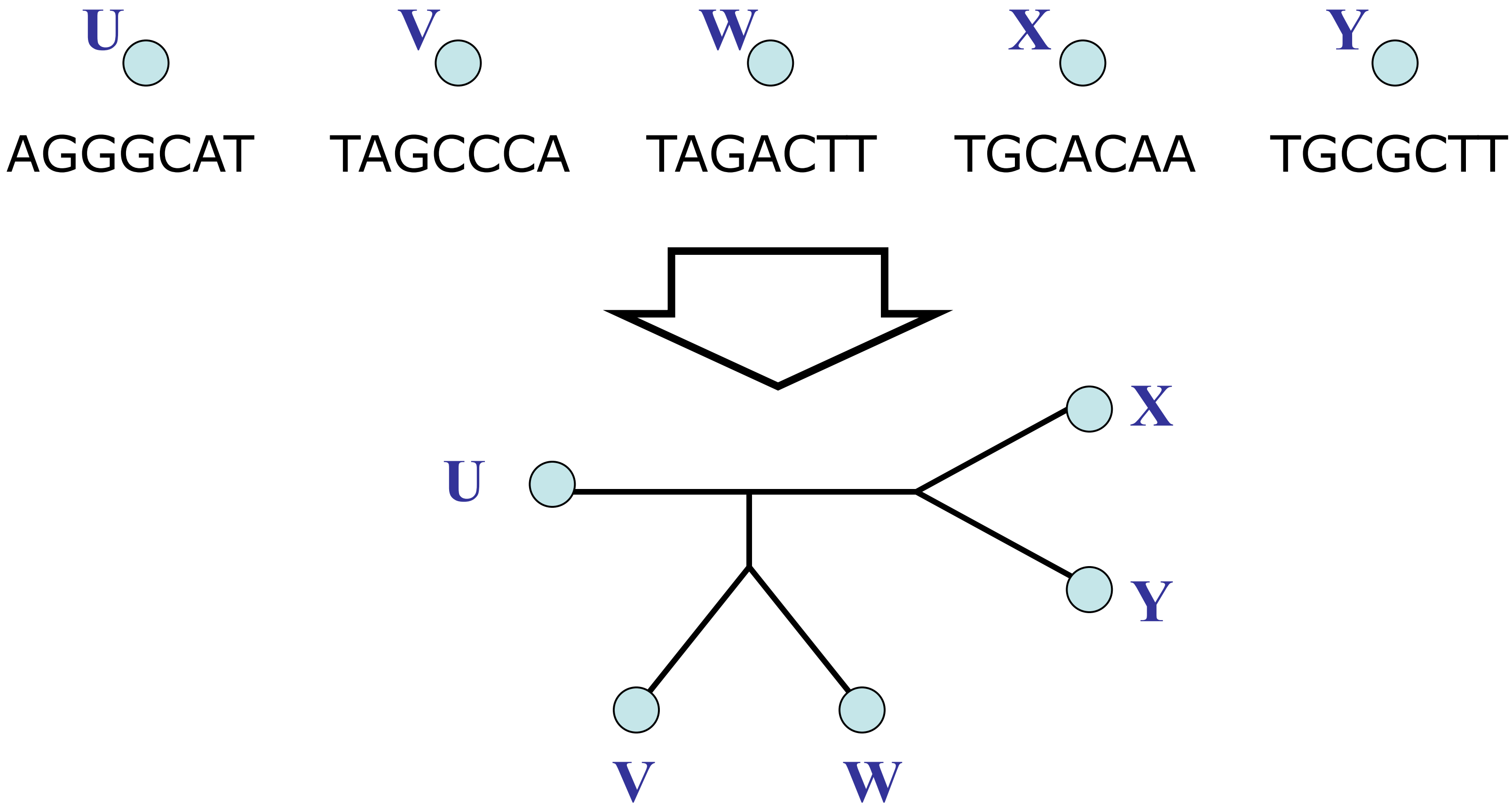
Sequence length requirements

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

Part 1: Absolute Fast Convergence

DNA Sequence Evolution





AGGGCAT TAGCCCA TAGACTT TGCACAA TGCGCTT

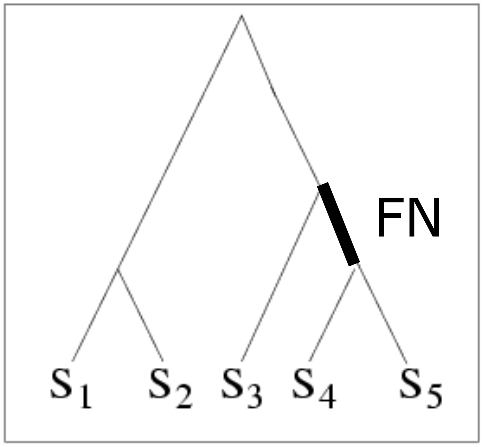
Markov Model of Site Evolution

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



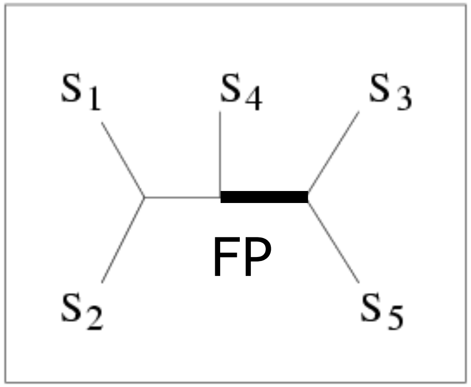
TRUE TREE

S_1	ACAATTAGAAC
S_2	ACCCTTAGAAC
S_3	ACCATTCCAAC
S_4	ACCAGACCAAC
S_5	ACCAGACCGGA

DNA SEQUENCES

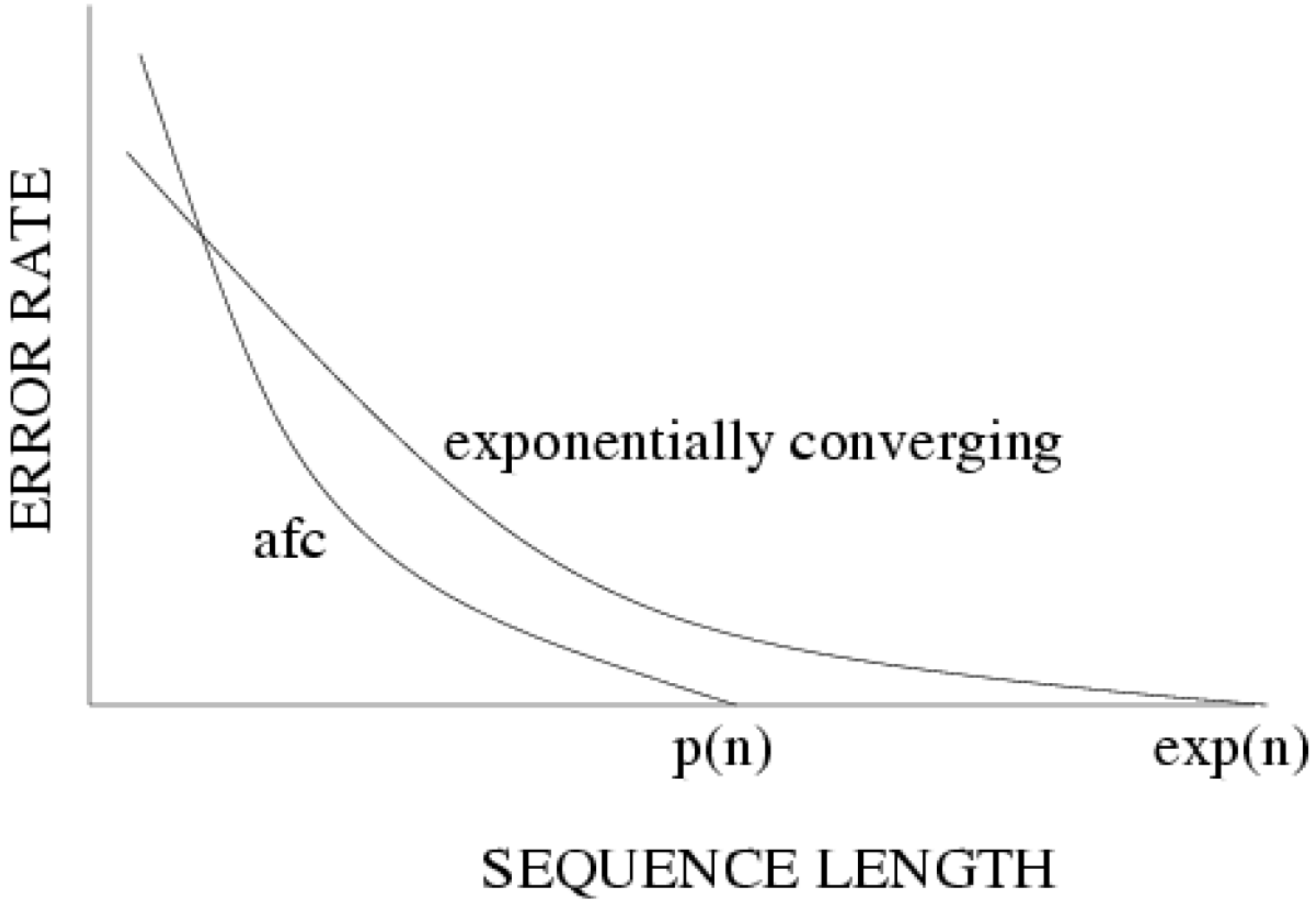
FN: false negative
(missing edge)
FP: false positive
(incorrect edge)

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-ε 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**.

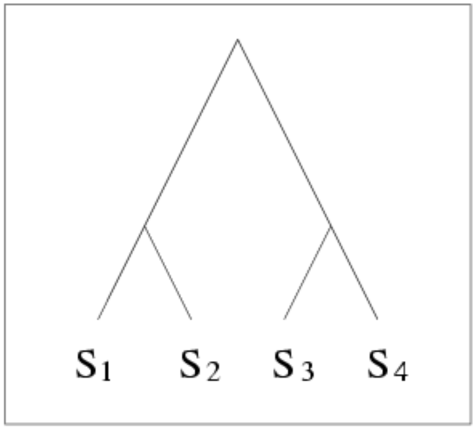
Afc methods

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. $Pr(M(S)=T) > 1- \varepsilon$, 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

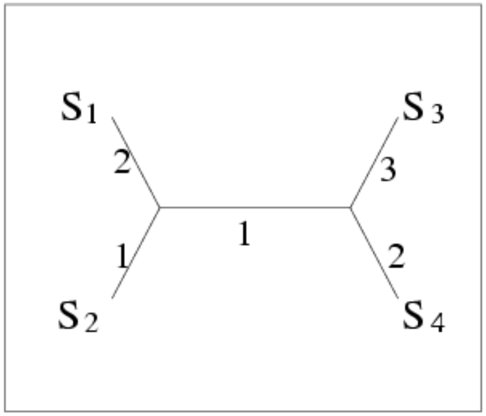
DNA SEQUENCES

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

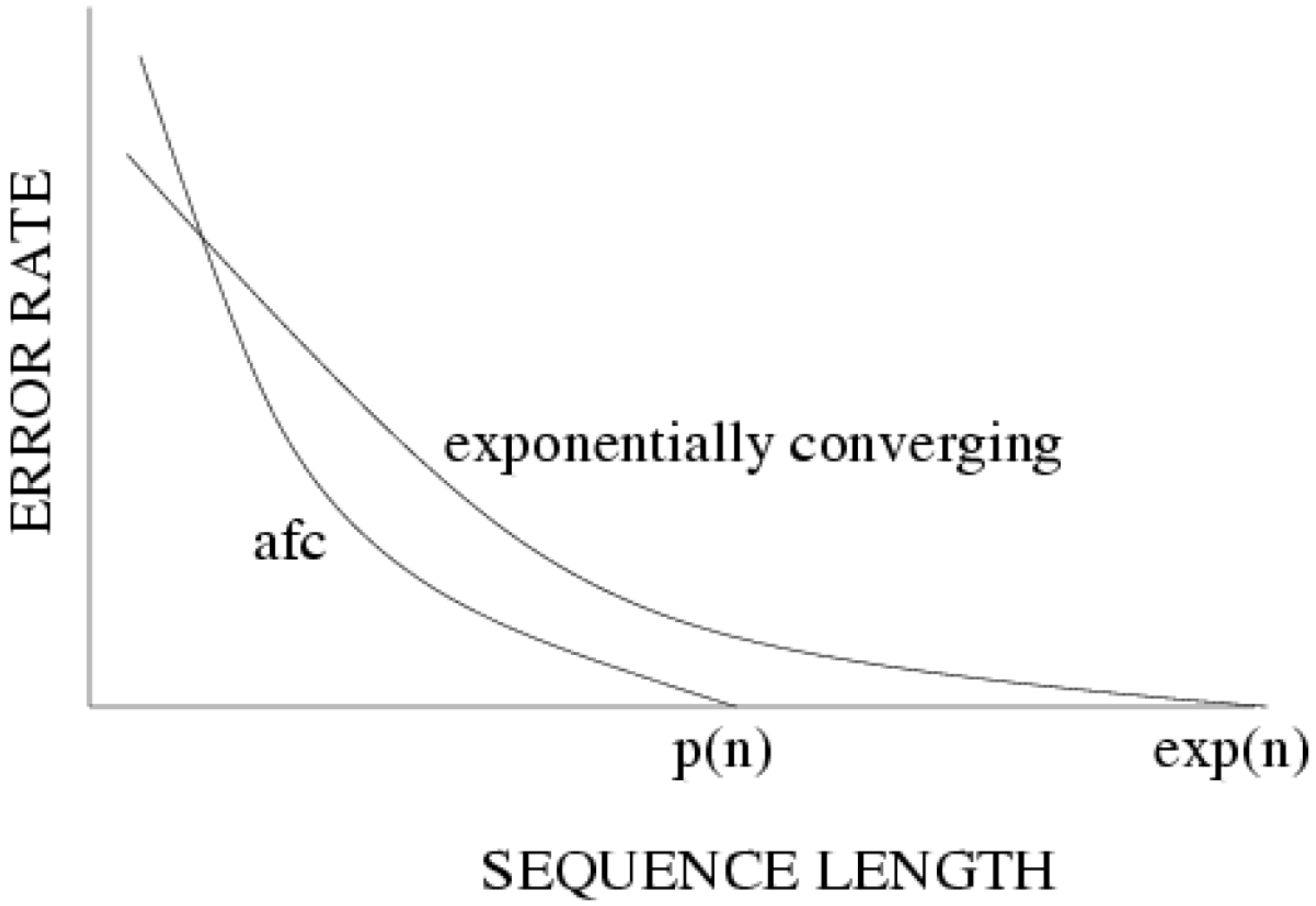
DISTANCE MATRIX

METHODS
SUCH AS
NEIGHBOR
JOINING



INFERRED TREE

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

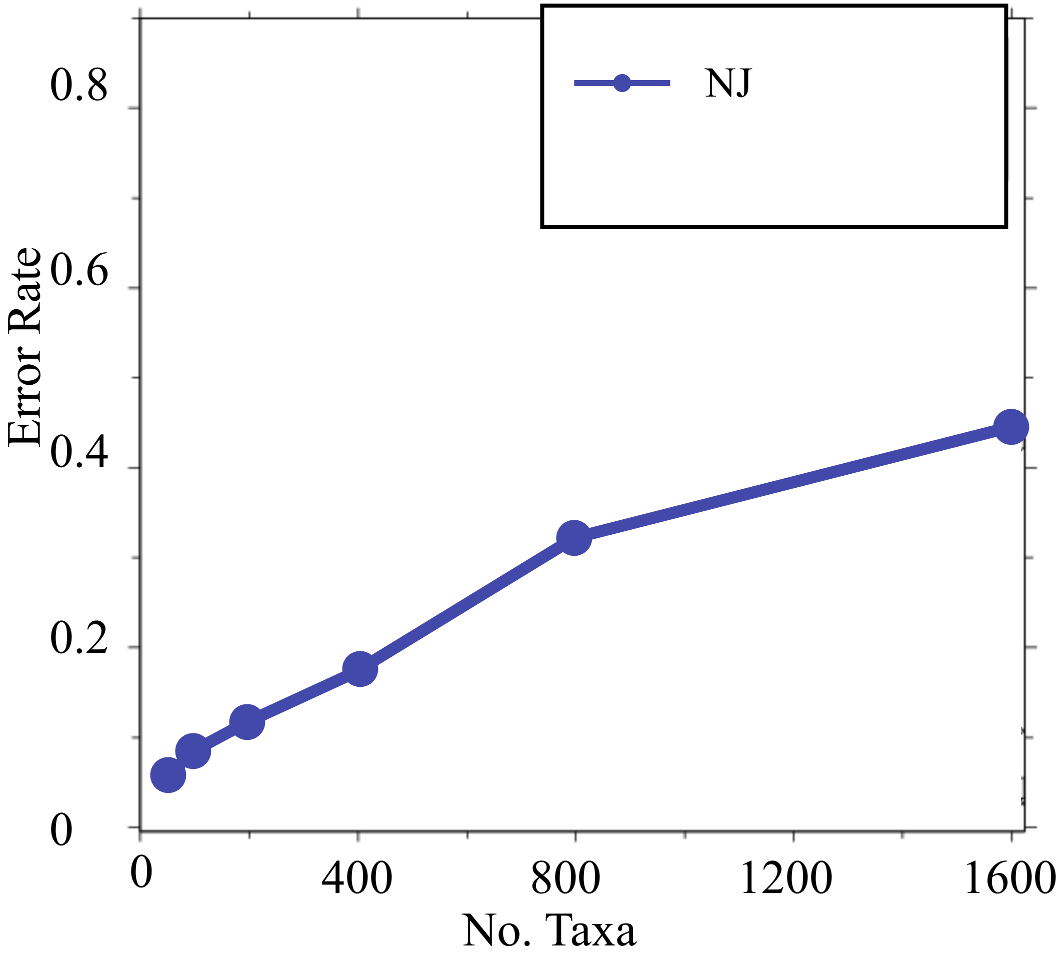


Theorem (Erdos et al., Atteson): Neighbor joining (and some other methods) will return the true tree w.h.p. provided sequence lengths are exponential in the evolutionary diameter of the tree.

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)
- 2013: Roch (in preparation)

and others

II: Short Quartet Methods

- The first “absolute fast converging” methods were based on “short quartets”, which are quartet trees formed by taking the nearest leaf in each subtree around some edge.

- “Nearest” can be based on any branch lengths, including just unit branch lengths.

Short Quartets Define the Tree

- **Theorem:** Let (T,w) be a tree with branch lengths, and let Q be the set of short quartet trees of T. If T’ is some tree on the same leaf set, and Q is a subset of Q(T’), then T=T’.
- Proof: Recall that T=T’ iff Q(T)=Q(T’). Then we will show that the dyadic closure(Q) = Q(T), and the result follows.

Dyadic Closure

- AB|CD + BC|DE defines a tree on A,B,C,D,E, and so implies quartets
  - AB|CE
  - AB|DE
  - AC|DE

- AB|CD + AB|CE => AB|DE

The first short quartet method

Given distance matrix D and threshold q, DO:

- Erase all entries in D that are bigger than q.
- For all quartets i,j,k,l such that all pairwise distances are at most q, use the Four Point Method to compute a tree on i,j,k,l.
- Compute the Dyadic Closure Q of this set of quartet trees.
- If no conflicts occur, then Q = Q(T) for some tree; compute T_q using the Naïve Quartet Method. Else reject q.

The Short Quartet Method

- After you compute T_q for each q in D, see which case is true:
  - All threshold values for q are rejected
  - At least one value is not rejected, and all non-rejected values return the same tree
  - At least two values are not rejected but they return different trees

The Short Quartet Method

- The outcome we want is:
  - At least one value is not rejected, and all non-rejected values return the same tree

- We can prove that this outcome happens with high probability given polynomial length sequences, and that it returns the true tree!
- In other words, the Dyadic Closure Method is absolute fast converging.

Nice, but

- Although the Dyadic Closure method is absolute fast converging, it generally has bad performance: it returns the true tree or no tree, and most often it will return no tree.

- So it has good theory but bad performance, like the Naïve Quartet Method.

DCM1: another afc method

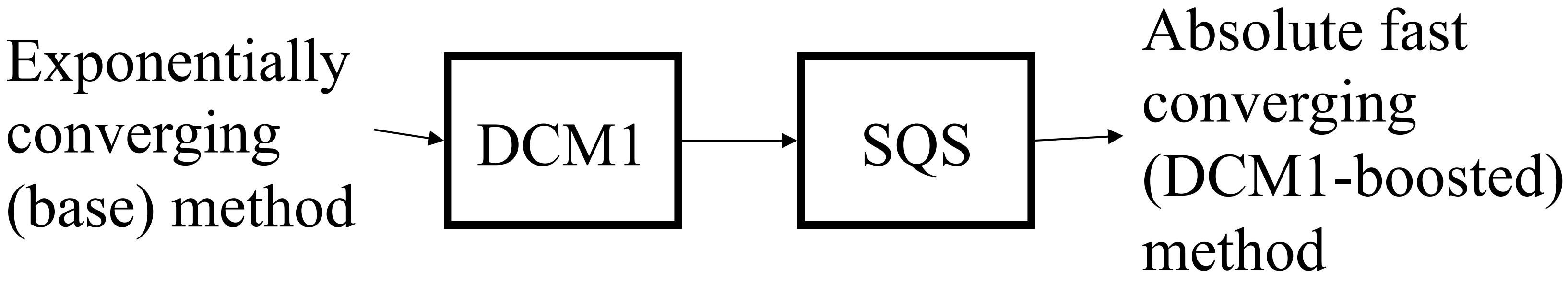
- DCM: disk-covering method

- Idea is to use divide-and-conquer to decompose a dataset into subsets, apply your favored method to construct trees on the subsets, and then combine these trees into a tree on the full dataset.

But, the details matter (see Stendhal)

DCM1-boosting:

Warnow, St. John, and Moret, SODA 2001



Exponentially converging (base) method

DCM1

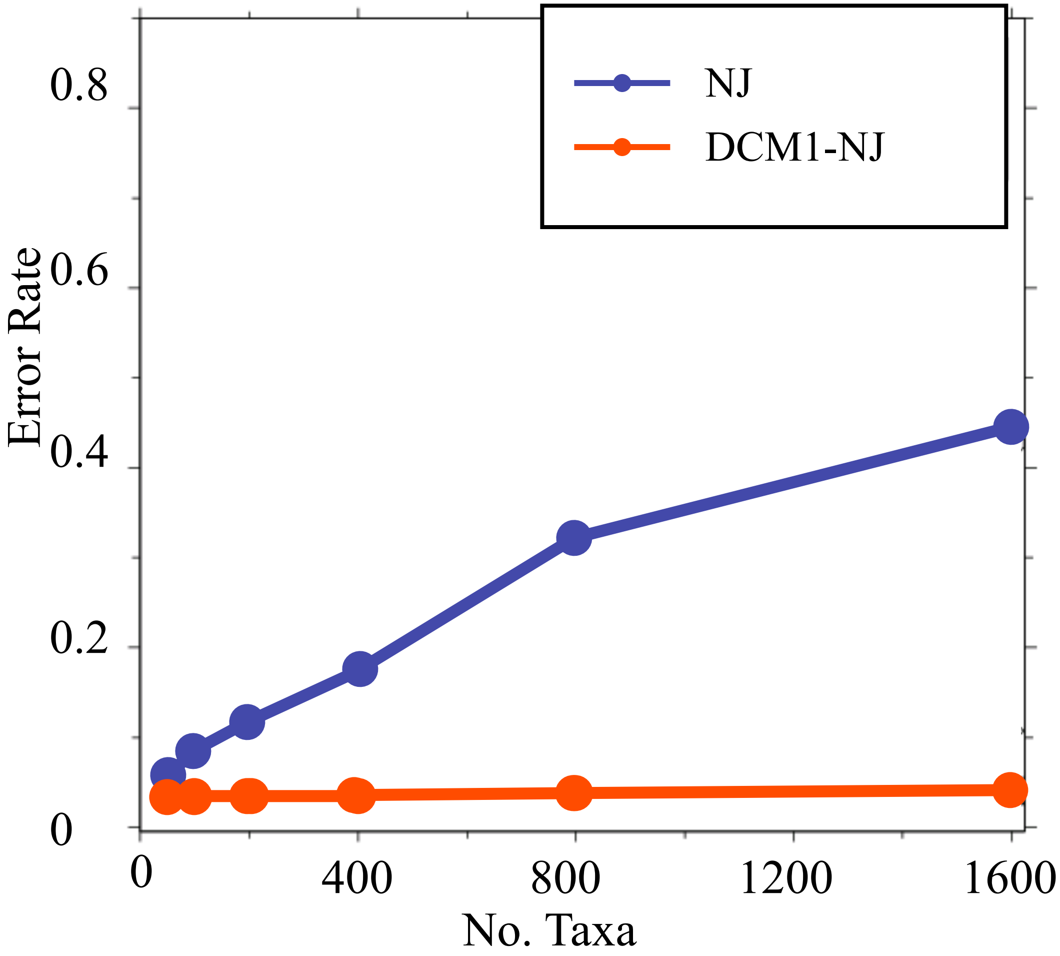
SQS

Absolute fast converging (DCM1-boosted) method

- 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]



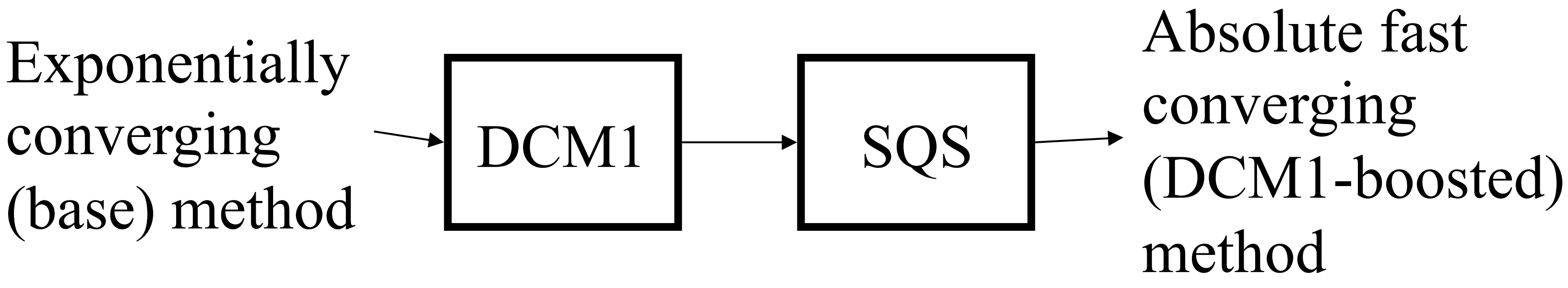
Theorem (Warnow et al., SODA 2001): DCM1-NJ converges to the true tree from polynomial length sequences

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



Exponentially converging (base) method

DCM1

SQS

Absolute fast converging (DCM1-boosted) method

- 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.

More about chordal graphs

- If G is not a clique, then for any pair of vertices a,b that are not adjacent, the minimum vertex separator is a clique

Chordal graphs

- A chordal graph has a perfect elimination scheme (an ordering on the vertices so that for every vertex, the set of neighbors of the vertex that follow it in the ordering form a clique).
- In fact, any graph that has a perfect elimination scheme is chordal!
- Hence we can determine if a graph is chordal using a greedy algorithm.

More about chordal graphs

- A graph is chordal if and only if it is the intersection graph of a set of subtrees of a tree.
- This theorem is why the Perfect Phylogeny Problem and the Triangulating Colored Graphs problem are equivalent.

More about chordal graphs

- If D is an additive distance matrix and q is a positive number, then the Threshold Graph TG(d,q) is chordal, where
  - TG(d,q) has n vertices v1, v2, …, vn
  - and has edges (i,j) if and only if D[i,j] <= q.

- 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

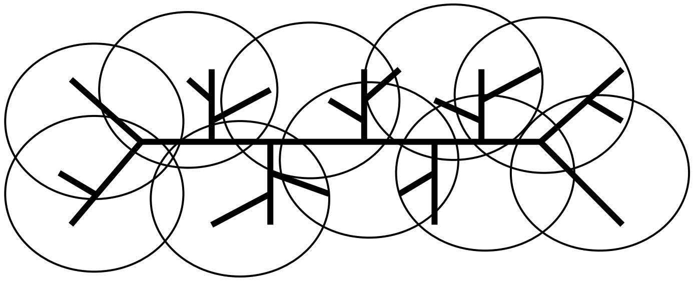
Input: Set S of sequences, distance matrix d, threshold value $q \in \{d_{ij}\}$

1. Compute threshold graph

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

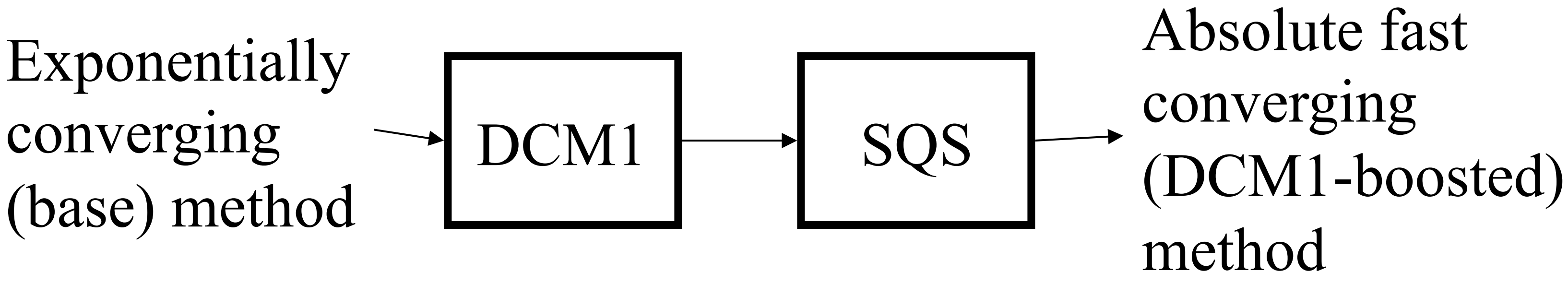
2. Perform minimum weight triangulation (note: if d is an additive matrix, then the threshold graph is provably triangulated).

DCM1 decomposition : Compute maximal cliques



DCM1-boosting:

Warnow, St. John, and Moret, SODA 2001



Exponentially converging (base) method

DCM1

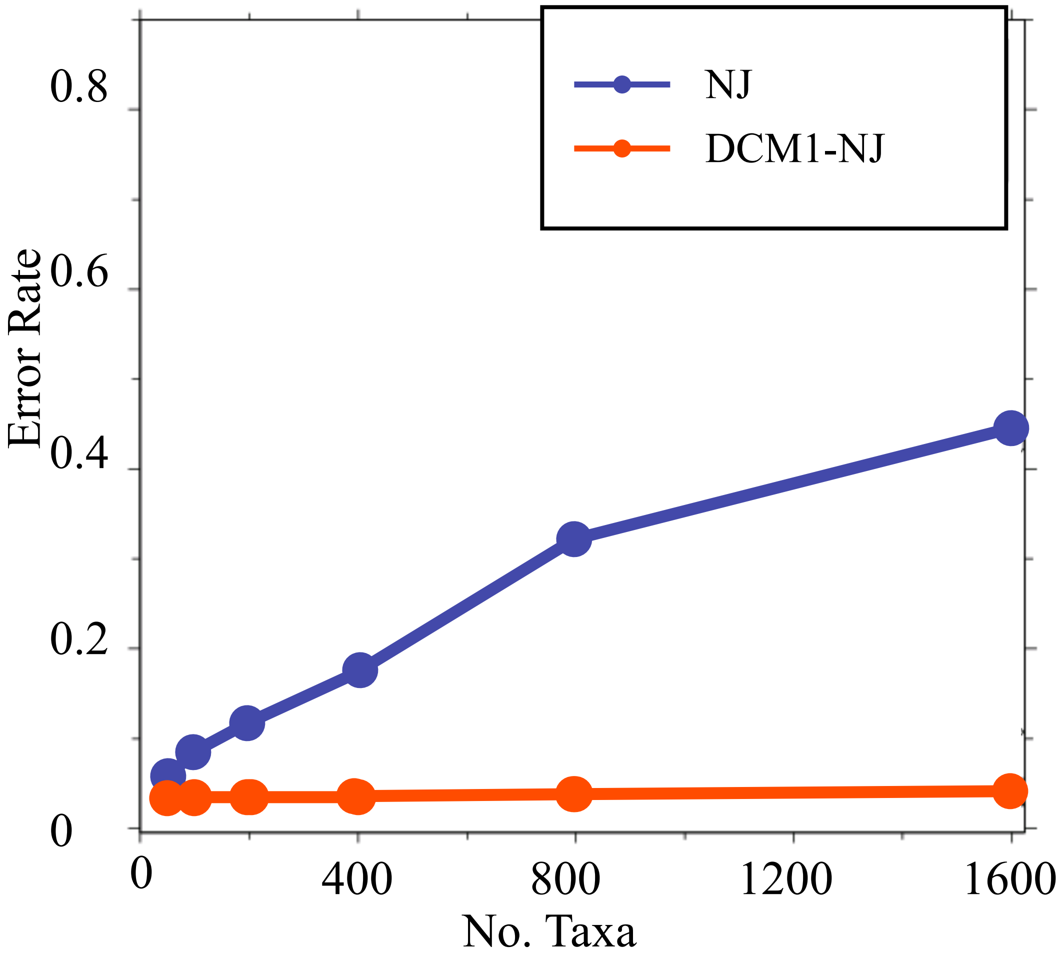
SQS

Absolute fast converging (DCM1-boosted) method

- 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

Sebastien Roch recently proved maximum likelihood is afc

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")?

Some open questions

- 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?