# Statistical stuff: models, methods, and performance issues

CS 598

Algorithmic Computational Genomics

# Today's Class

- Phylogeny as statistical estimation problem
- Stochastic models of evolution
- Distance-based estimation

# Phylogeny estimation as a statistical inverse problem

# Estimation of evolutionary trees as a statistical inverse problem

- We can consider characters as properties that evolve down trees.
- We observe the character states at the leaves, but the internal nodes of the tree also have states.
- The challenge is to estimate the tree from the properties of the taxa at the leaves. This is enabled by characterizing the evolutionary process as accurately as we can.

# Performance criteria

- Running time.
- Space.
- Statistical performance issues (e.g., statistical consistency and sequence length requirements)
- “Topological accuracy” with respect to the underlying ***true tree.*** Typically studied in simulation.
- Accuracy with respect to a mathematical score (e.g. tree length or likelihood score) on real data.

# Statistical models

- Simple example: coin tosses.
- Suppose your coin has probability **p** of turning up heads, and you want to estimate **p**. How do you do this?

# Estimating p

- Toss coin repeatedly
- Let your estimate **q** be the fraction of the time you get a head

- Obvious observation: *q will approach p* as the number of coin tosses increases
- This algorithm is a *statistically consistent* estimator of p. That is, your error |q-p| goes to 0 (with high probability) as the number of coin tosses increases.

# Another estimation problem

- Suppose your coin is biased either towards heads or tails (so that p is not 1/2).
- How do you determine which type of coin you have?

- Same algorithm, but say “heads” if q>1/2, and “tails” if q<1/2. For large enough number of coin tosses, ***your answer will be correct with high probability***.

# Markov models of character evolution down trees

- The character might be binary, indicating absence or presence of some property at each node in the tree.
- The character might be multi-state, taking on one of a specific set of possible states. Typical examples in biology: the nucleotide in a particular position within a multiple sequence alignment.
- A probabilistic model of character evolution describes a random process by which a character changes state on each edge of the tree. Thus it consists of a tree T and associated parameters that determine these probabilities.
- The "Markov" property assumes that the state a character attains at a node v is determined only by the state at the immediate ancestor of v, and not also by states before then.

# Binary characters

- Simplest type of character: presence (1) or absence (0).
- How do we model the presence or absence of a property?

# Simplest model of binary character evolution: **Cavender-Farris**

- For each edge **e**, there is a probability **p(e)** of the property "changing state" (going from 0 to 1, or vice-versa), with $0<p(e)<0.5$ (to ensure that CF trees are identifiable).
- Every position evolves under the same process, independently of the others.

# Statistical models of evolution

- Instead of directly estimating the tree, we try to estimate the process itself.
- For example, we try to estimate the probability that two leaves will have different states for a random character.

# Cavender-Farris pattern probabilities

- Let x and y denote nodes in the tree, and $p_{xy}$ denote the probability that x and y exhibit different states.
- Theorem: Let $p_i$ be the substitution probability for edge $e_i$, and let x and y be connected by path $e_1e_2e_3\ldots e_k$. Then

$$1-2p_{xy} = (1-2p_1)(1-2p_2)\ldots(1-2p_k)$$

# And then take logarithms

- The theorem gave us:

$$1-2p_{xy} = (1-2p_1)(1-2p_2)\ldots(1-2p_k)$$

- If we take logarithms, we obtain

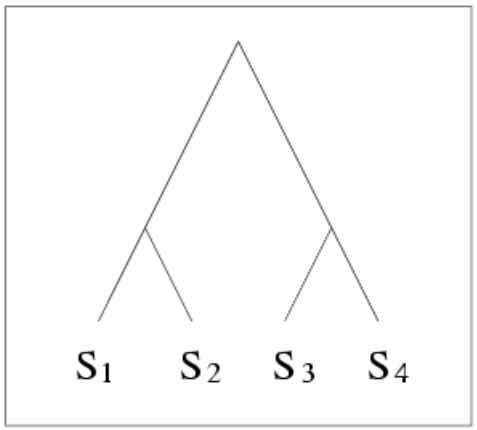
$$\ln(1-2p_{xy}) = \ln(1-2p_1) + \ln(1-2p_2)+\ldots+\ln(1-2p_k)$$

- Since these probabilities lie between 0 and 0.5, these logarithms are all negative. So let's multiply by -1 to get positive numbers.

# An additive matrix!

- Consider a matrix $D(x,y) = -\ln(1-2p_{xy})$
- This matrix is additive!
- Can we estimate this additive matrix from what we observe at the leaves of the tree?
- Key issue: how to estimate $p_{xy.}$
- (Recall how to estimate the probability of a head…)

# Distance-based Methods




TRUE TREE

| | |
|---|---|
| $S_1$ | ACAATTAGAAC |
| $S_2$ | ACCCTTAGAAC |
| $S_3$ | ACCATTCCAAC |
| $S_4$ | ACCAGACCAAC |

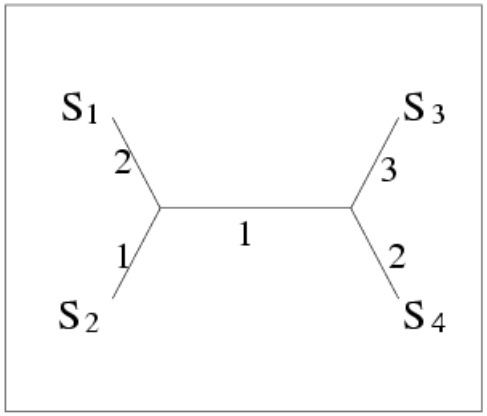
DNA SEQUENCES

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

# Estimating CF distances

- Consider

  $d_{ij}$= -1/2 ln(1-2H(i,j)/k),

  where k is the number of characters, and H(i,j) is the Hamming distance between sequences $s_i$ and $s_j$.

- Theorem: as k increases,

  $d_{ij}$ converges to $D_{ij}$ = -1/2 ln(1-2$p_{ij}$),

  which is an additive matrix.

# CF tree estimation

- Step 1: Compute Hamming distances
- Step 2: Correct the Hamming distances, using the CF distance calculation
- Step 3: Use distance-based method (neighbor joining, naïve quartet method, etc.)

# Four Point Method

- Task: Given 4x4 dissimilarity matrix, compute a tree on four leaves
- Solution: Compute the three pairwise sums, and take the split ij|kl that gives the minimum!
- When is this guaranteed accurate?

# Error tolerance for FPM

- Suppose every pairwise distance is estimated well enough (within f/2, for f the minimum length of any edge).
- Then the Four Point Method returns the correct tree (i.e., ij+kl remains the minimum)

# Naïve Quartet Method

- Compute the tree on each quartet using the four-point condition
- Merge them into a tree on the entire set if they are compatible:
  - Find a sibling pair A,B
  - Recurse on S-{A}
  - If S-{A} has a tree T, insert A into T by making A a sibling to B, and return the tree

# Error tolerance for NQM

- Suppose every pairwise distance is estimated well enough (within f/2, for f the minimum length of any edge).
- Then the Four Point Method returns the correct tree on every quartet.
- And so all quartet trees are compatible, and NQM returns the true tree.

# In other words:

- The NQM method is statistically consistent methods for estimating Cavender-Farris trees!

- Plus it is polynomial time!

# DNA substitution models

- Every edge has a substitution probability
- The model also allows 4x4 substitution matrices on the edges:
  - Simplest model: Jukes-Cantor (JC) assumes that all substitutions are equiprobable
  - General Time Reversible (GTR) Model: one 4x4 substitution matrix for all edges
  - General Markov (GM) model: different 4x4 matrices allowed on each edge

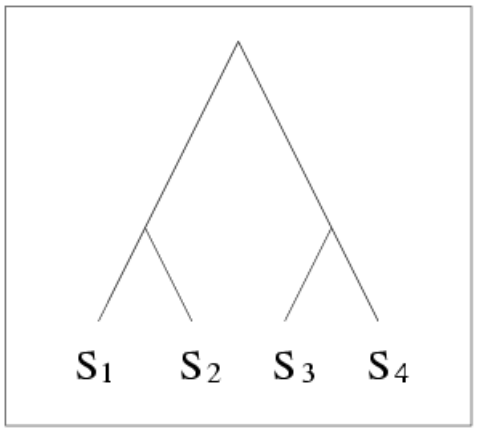
# Jukes-Cantor DNA model

- Character states are A,C,T,G (nucleotides).
- All substitutions have equal probability.
- On each edge **e**, there is a value **p(e)** indicating the probability of change from one nucleotide to another on the edge, with $0<p(e)<0.75$ (to ensure that JC trees are identifiable).
- The state (nucleotide) at the root is random (all nucleotides occur with equal probability).
- All the positions in the sequence evolve identically and independently.

# Jukes-Cantor distances

- $D_{ij}$ = -3/4 ln(1-4/3 H(i,j)/k)) where k is the sequence length
- These distances converge to an additive matrix, just like with Cavender-Farris distances

# Distance-based Methods




TRUE TREE

| | |
|---|---|
| $S_1$ | ACAATTAGAAC |
| $S_2$ | ACCCTTAGAAC |
| $S_3$ | ACCATTCCAAC |
| $S_4$ | ACCAGACCAAC |

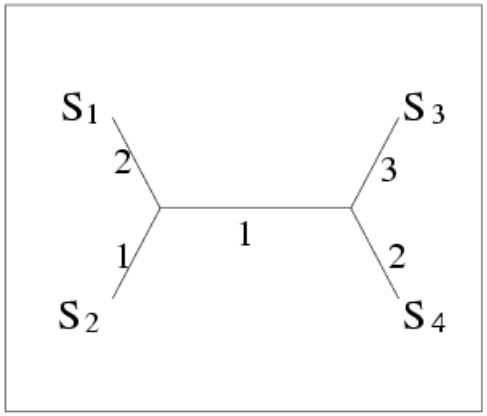
DNA SEQUENCES

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

# Other statistically consistent methods

- Maximum Likelihood
- Bayesian MCMC methods
- Distance-based methods (like Neighbor Joining and the Naïve Quartet Method)

But not maximum parsimony, not maximum compatibility, and not UPGMA (a distance-based method)

# UPGMA

While |S|>2:

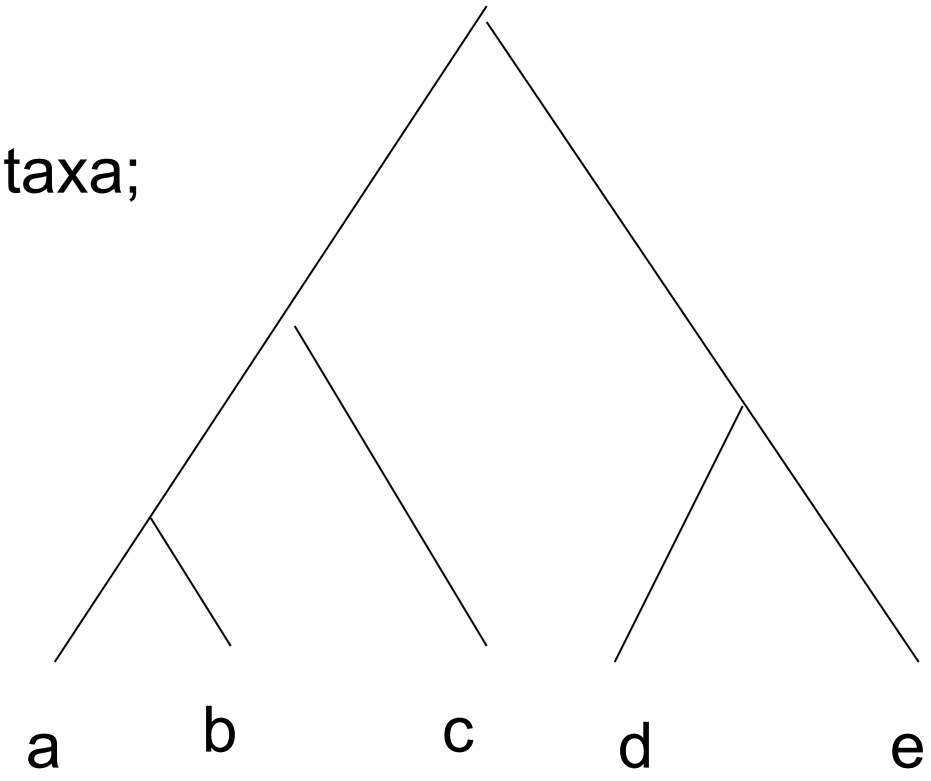
find pair x,y of closest taxa;

delete x

Recurse on S-{x}

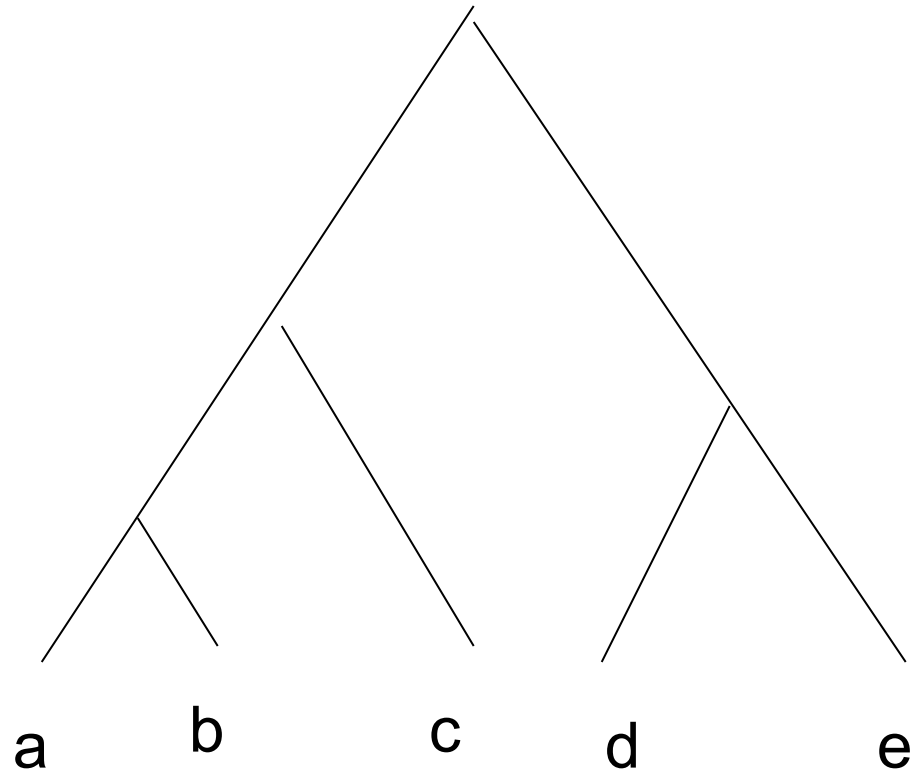
Insert y as sibling to x

Return tree



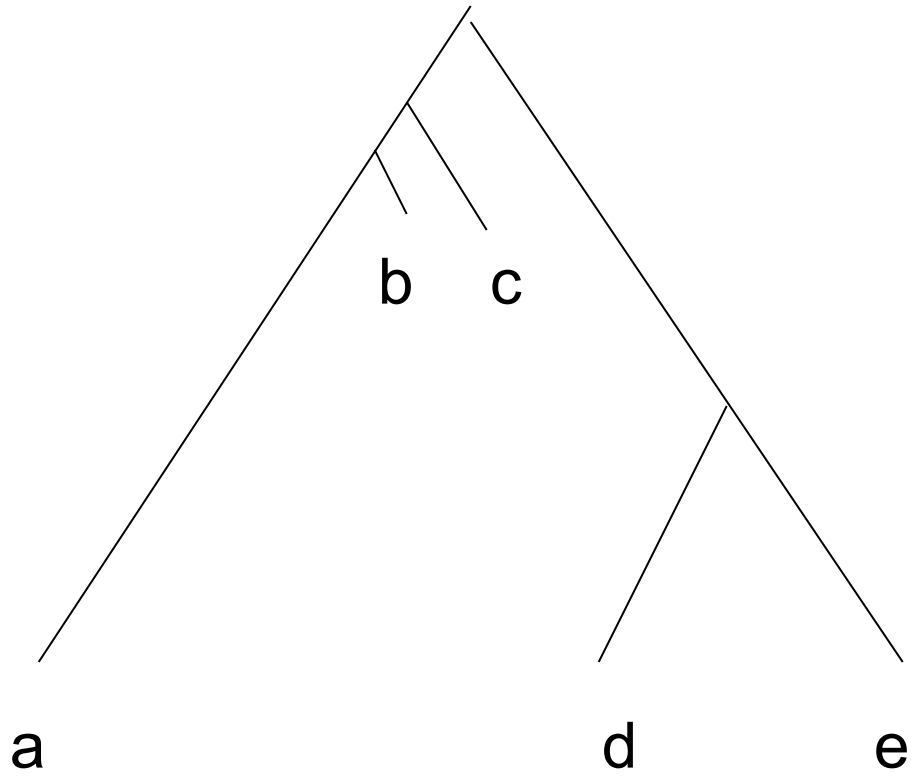
# UPGMA

Works when evolution is "clocklike"



# UPGMA

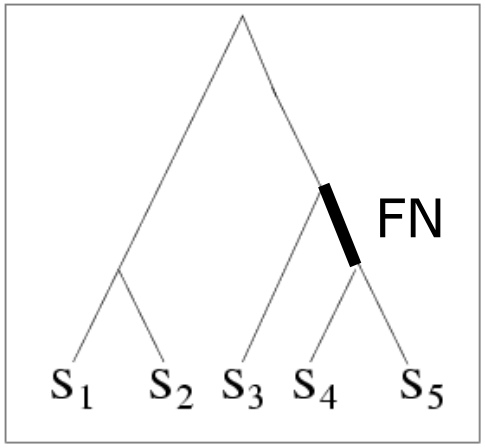
Fails to produce true tree if evolution deviates too much from a clock!



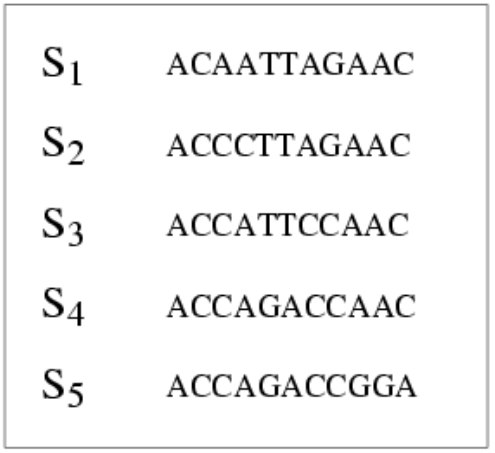
# Better distance-based methods

- Neighbor Joining
- Minimum Evolution
- Weighted Neighbor Joining
- Bio-NJ
- DCM-NJ
- And others

# Quantifying Error



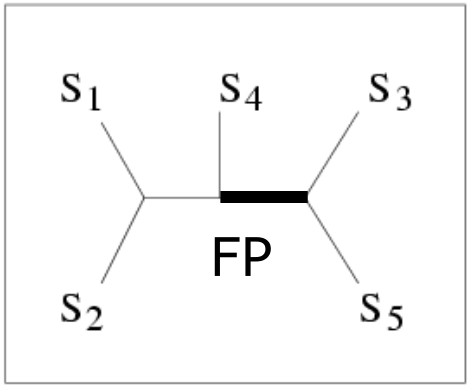

TRUE TREE




DNA SEQUENCES

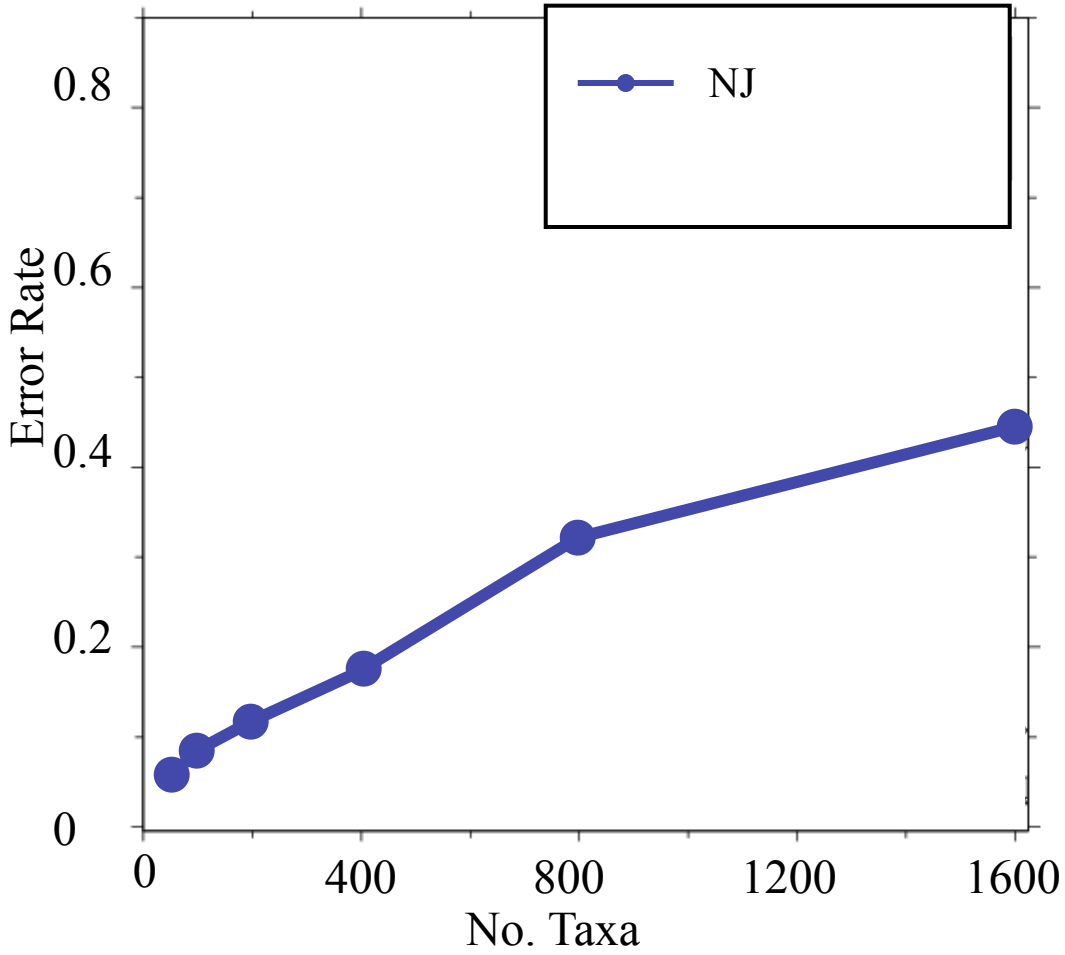
FN: false negative
(missing edge)
FP: false positive
(incorrect edge)

50% error rate




INFERRED TREE

# Neighbor joining has poor performance on large diameter trees *[Nakhleh et al. ISMB 2001]*




**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.

# Statistical Methods

- Many statistical models for biomolecular sequence evolution (JC, K2P, GTR, GM, plus lots more)
- Maximum Likelihood and Bayesian Estimation are the two basic statistical approaches to phylogeny estimation
- MrBayes is a popular Bayesian methods (but there are others)
- RAxML, FastTree-2, IQtree, are among the most accurate ML methods for large datasets, but there are others
- Issues: running time, memory, and models…

# Maximum Likelihood

- Input: sequence data S,
- Output: the model tree (tree T and parameters theta) s.t. Pr(S|T,theta) is maximized.

NP-hard.

Important in practice.

Good heuristics!

But what does it mean?

# Computing the probability of the data

- Given a model tree (with all the parameters set) and character data at the leaves, you can compute the probability of the data.
- Small trees can be done by hand.
- Large examples are computationally intensive - but still **polynomial** time (using an algorithmic trick).

# Cavender-Farris model calculations

- Consider an unrooted tree with topology ((a,b),(c,d)) with p(e)=0.1 for all edges.

- What is the probability of all leaves having state 0?

We show the brute-force technique.

# Brute-force calculation

Let E and F be the two internal nodes in the tree ((A,B), (C,D)).

Then Pr(A=B=C=D=0) =

- Pr(A=B=C=D=0|E=F=0) +
- Pr(A=B=C=D=0|E=1, F=0) +
- Pr(A=B=C=D=0|E=0, F=1) +
- Pr(A=B=C=D=0|E=F=1)

The notation “Pr(X|Y)” denotes the probability of X given Y.

# Calculation, cont.

Technique:

- Set one leaf to be the root
- Set the internal nodes to have some specific assignment of states (e.g., all 1)
- Compute the probability of that specific pattern
- Add up all the values you get, across all the ways of assigning states to internal nodes

# Calculation, cont.

Calculating Pr(A=B=C=D=0|E=F=0)

- There are 5 edges, and thus no change on any edge.
- Since p(e)=0.1, then the probability of no change is 0.9. So the probability of this pattern, given that the root is a particular leaf and has value 0, is $(0.9)^5$.
- Then we multiply by 0.5 (the probability of the root A having state 0).
- So the probability is $(0.5)x\ (0.9)^5$.

# Maximum likelihood under Cavender-Farris

- Given a set S of binary sequences, find the Cavender-Farris model tree (tree topology and edge parameters) that maximizes the probability of producing the input data S.

ML, if solved exactly, is statistically consistent under Cavender-Farris (and under the DNA sequence models, and more complex models as well).

The problem is that **ML is hard to solve.**

# “Solving ML”

- Technique 1: compute the probability of the data under each model tree, and return the best solution.
- Problem: Exponentially many trees on n sequences, and infinitely many ways of setting the parameters on each of these trees!

# “Solving ML”

- Technique 2: For each of the tree topologies, find the best parameter settings.
- Problem: Exponentially many trees on n sequences, and calculating the best setting of the parameters on any given tree is hard!

Even so, there are hill-climbing heuristics for both of these calculations (finding parameter settings, and finding trees).

# Bayesian analyses

- Algorithm is a **random walk** through space of all possible model trees (trees with substitution matrices on edges, etc.).
- From your current model tree, you perturb the tree topology and numerical parameters to obtain a new model tree.
- Compute the probability of the data (character states at the leaves) for the new model tree.
  - If the probability increases, accept the new model tree.
  - If the probability is lower, then accept with some probability (that depends upon the algorithm design and the new probability).
- Run for a long time…

# Bayesian estimation

After the random walk has been run for a very long time…

- Gather a random sample of the trees you visit
- Return:
  - Statistics about the random sample (e.g., how many trees have a particular bipartition), OR
  - Consensus tree of the random sample, OR
  - The tree that is visited most frequently

Bayesian methods, if run *long enough*, are statistically consistent methods (the tree that appears the most often will be the true tree with high probability).

**MrBayes** is standard software for Bayesian analyses in biology.

# Phylogeny estimation statistical issues

- Is the phylogeny estimation method statistically consistent under the given model?
- How much data does the method need need to produce a correct tree?
- Is the method robust to model violations?
- *Is the character evolution model reasonable?*