MP, ML, AML Reconstruction of Phylogenetic Trees: A Status Report

Benny Chor

School of Computer Science
Tel-Aviv University
Phylogenetic Reconstruction

• Input: A set of $n$ aligned sequences (genes, proteins) from $n$ species,
Phylogenetic Reconstruction

• Input: A set of $n$ aligned sequences (genes, proteins) from $n$ species,
• Goal: Reconstruct the tree which best explains the evolutionary history of this gene/protein.
Phylogenetic Reconstruction

- Input: A set of $n$ aligned sequences (genes, proteins) from $n$ species,
- Goal: Reconstruct the tree which best explains the evolutionary history of this gene/protein.
- Tree reconstruction is still a challenge today.
Phylogenetic Reconstruction

- **Input:** A set of $n$ aligned sequences (genes, proteins) from $n$ species,
- **Goal:** Reconstruct the tree which best explains the evolutionary history of this gene/protein.
- Tree reconstruction is still a challenge today.
- Many concrete questions are still unresolved (e.g. mammalian evolutionary tree).
Phylogenetic Reconstruction

- **Input**: A set of $n$ aligned sequences (genes, proteins) from $n$ species,
- **Goal**: Reconstruct the tree which best explains the evolutionary history of this gene/protein.
- Tree reconstruction is still a challenge today.
- Many concrete questions are still unresolved (e.g. mammalian evolutionary tree).
- Most realistic formulations of the problem, which take errors into account, give rise to hard computational problems.
Popular Methods

- Distance based methods:
Popular Methods

- Distance based methods:
  - UPGMA
Popular Methods

- Distance based methods:
  - UPGMA
  - Neighbor Joining.
Popular Methods

- Distance based methods:
  - UPGMA
  - Neighbor Joining.
  - Buneman trees.
Popular Methods

• Distance based methods:
  • UPGMA
  • Neighbor Joining.
  • Buneman trees.

• Character Based Methods:
Popular Methods

- Distance based methods:
  - UPGMA
  - Neighbor Joining.
  - Buneman trees.

- Character Based Methods:
  - Maximum Parsimony.
Popular Methods

• Distance based methods:
  • UPGMA
  • Neighbor Joining.
  • Buneman trees.

• Character Based Methods:
  • Maximum Parsimony.
  • Maximum Likelihood.
Popular Methods

• Distance based methods:
  • UPGMA
  • Neighbor Joining.
  • Buneman trees.

• Character Based Methods:
  • Maximum Parsimony.
  • Maximum Likelihood.

• Additional Methods:
Popular Methods

• Distance based methods:
  • UPGMA
  • Neighbor Joining.
  • Buneman trees.

• Character Based Methods:
  • Maximum Parsimony.
  • Maximum Likelihood.

• Additional Methods:
  • Quartets Based.
Popular Methods

• Distance based methods:
  • UPGMA
  • Neighbor Joining.
  • Buneman trees.

• Character Based Methods:
  • Maximum Parsimony.
  • Maximum Likelihood.

• Additional Methods:
  • Quartets Based.
  • Disc Covering.
Talk Outline

• Maximum likelihood (ML).
Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
- Existence of multiple maxima.
Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
- Existence of multiple maxima.
- Computation complexity: Maximum likelihood vs. maximum parsimony (MP).
Talk Outline

- Maximum likelihood (ML).
- The likelihood surface.
- Existence of multiple maxima.
- Computation complexity: Maximum likelihood vs. maximum parsimony (MP).
- Ancestral maximum likelihood (AML) and its computational complexity.
Maximum Likelihood

- **Input**: A set of \( n \) observed sequences and an underlying substitution model.
Maximum Likelihood

- **Input:** A set of $n$ observed sequences and an underlying substitution model.
- **Desired Output:** The weighted tree $T$ that maximizes the likelihood of the data.
Maximum Likelihood

• **Input:** A set of $n$ observed sequences and an underlying substitution model.

• **Desired Output:** The weighted tree $T$ that maximizes the likelihood of the data.

• **Likelihood** of a data: The conditional probability of producing the data, given the model parameters.
Maximum Likelihood

- **Input:** A set of $n$ observed sequences and an underlying substitution model.
- **Desired Output:** The weighted tree $T$ that maximizes the likelihood of the data.
- **Likelihood** of a data: The conditional probability of producing the data, given the model parameters.
- Likelihood is a common optimization criteria in numerous settings, including phylogenetic (Felsenstein 1981).
2–State Substitution Model

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XYYYYYYYYYXYY XY XY YX XY X</td>
</tr>
<tr>
<td>2</td>
<td>XYYYYYYYYYYXX YX YX YX YX X</td>
</tr>
<tr>
<td>3</td>
<td>XYYYYYYYYYXY YX XY XY XY X</td>
</tr>
<tr>
<td>4</td>
<td>XYYYYYYYYYYXY YX XY XY YX Y</td>
</tr>
</tbody>
</table>

- Just two characters states, X and Y.
2–State Substitution Model

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XXXXXXXXXYYYY YXY XY YY XY X</td>
</tr>
<tr>
<td>2</td>
<td>XXXXXXXXYYYY YYX YX YY YX X</td>
</tr>
<tr>
<td>3</td>
<td>XXXXXXXXYYYY YYX XY XY XY X</td>
</tr>
<tr>
<td>4</td>
<td>XXXXXXXXYYYY YYX XY XY YY Y</td>
</tr>
</tbody>
</table>

- Just two characters states, X and Y.
- Transitions between states are symmetric.
2–State Substitution Model

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XXXXXXXXXXXYYY  XXY  XY  YY  XY  X</td>
</tr>
<tr>
<td>2</td>
<td>XXXXXXXXXXXYYY  YYX  YX  YY  YY  X</td>
</tr>
<tr>
<td>3</td>
<td>XXXXXXXXXXXYYY  YYX  XY  YY  YY  X</td>
</tr>
<tr>
<td>4</td>
<td>XXXXXXXXXXXYYY  YYX  XY  YY  YY  Y</td>
</tr>
</tbody>
</table>

- Just **two** characters states, **X** and **Y**.
- Transitions between states are symmetric.
- Equal rates across sites.
2–State Substitution Model

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XXXXXXXXXYYYY XYXY XY YX XY X</td>
</tr>
<tr>
<td>2</td>
<td>XXXXXXXXXYYY YYX YX YY YX X</td>
</tr>
<tr>
<td>3</td>
<td>XXXXXXXXXYYY YYX XY XY XY X</td>
</tr>
<tr>
<td>4</td>
<td>XXXXXXXXXYYY YYX XY XY YY Y</td>
</tr>
</tbody>
</table>

- Just two characters states, X and Y.
- Transitions between states are symmetric.
- Equal rates across sites.
- Every column induces a pattern.
# 2–State Substitution Model

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XXXXXXXXXXXY  YYY  XXY  XY  YX  XY  X</td>
</tr>
<tr>
<td>2</td>
<td>XXXXXXXXXXXY  YYY  YYX  YX  YX  YX  X</td>
</tr>
<tr>
<td>3</td>
<td>XXXXXXXXXXXY  YYY  YYX  XY  XY  XY  X</td>
</tr>
<tr>
<td>4</td>
<td>XXXXXXXXXXXY  YYY  YYX  XY  XY  YX  Y</td>
</tr>
</tbody>
</table>

- Just **two** characters states, X and Y.
- Transitions between states are symmetric.
- Equal rates across sites.
- Every column induces a pattern.
- **Remark:** A simple model, yet very powerful.
Neyman 2–State Substitution Model

For each edge $e$ of a tree $T$, the edge weight $p_e$ represents the probability of having different states at the two ends of $e$. 
A Very Simple Example

Four species \((n = 4)\), just one site \((c = 1)\)

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>X</td>
</tr>
<tr>
<td>2</td>
<td>X</td>
</tr>
<tr>
<td>3</td>
<td>Y</td>
</tr>
<tr>
<td>4</td>
<td>Y</td>
</tr>
</tbody>
</table>

Analyze the natural tree \((12)(34)\)
Computing the Likelihood

Each unknown state (?) can assume **one of two** possibilities, X or Y. For example, the assignment

\[
(1) \quad \text{X} \quad \text{Y (3)} \\
\text{X} \quad \text{Y} \\
\text{Y} \quad \text{Y (4)}
\]

contributes \((1 - p_1) \cdot (1 - p_2) \cdot p_{12} \cdot (1 - p_3) \cdot (1 - p_{123})\).

The likelihood is the sum of this

\[ + \text{ three similar expressions} \ldots \]
Computing the Likelihood (2)

\[ L(\text{data} \mid T, \text{edge parameters}) \]
\[ \triangleq \sum_{\text{internal assignments}} \prod_{\text{edges}} p^{d_e}(1 - p)^{\ell - d_e}. \]

Each \( d_e \) is number of unequal sites along edge \( e \). It depends on the internal assignment \( a \), and input pattern \( t \) at two ends of the edge.
Computing the Likelihood (2)

\[ L(\text{data} \mid T, \text{edge parameters}) \]
\[ \triangleq \sum_{\text{internal assignments}} \prod_{\text{edges}} p^{d_e} (1 - p)^{\ell - d_e}. \]

Each \( d_e \) is number of unequal sites along edge \( e \). It depends on the internal assignment \( a \), and input pattern \( t \) at two ends of the edge.

- A well defined objective function to maximize.
- Termed average likelihood by Penny and Steel.
- Widely used in practice.
Three Likelihood Versions

- **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree & edge weights})$. 
Three Likelihood Versions

- **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree \\& edge weights})$.

- **Small Likelihood**: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.
Three Likelihood Versions

• **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data} | \text{tree} & \text{edge weights})$.

• **Small Likelihood**: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.

• **Tiny Likelihood**: Given observed data & a tree & edge weights, find the likelihood.
Three Likelihood Versions

• **Big Likelihood**: Given the sequence data, find a tree and edge weights that maximize $L(\text{data}|\text{tree \\& edge weights})$.

• **Small Likelihood**: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.

• **Tiny Likelihood**: Given observed data & a tree & edge weights, find the likelihood.

• Tiny likelihood can be efficiently computed using dynamic programming (Felsenstein, 1981).
Hill Climbing / Small Likelihood

• Typical approach to small likelihood, used in practice:
• Typical approach to small likelihood, used in practice:
• Start at some initial point with edge weights $p$. 
Hill Climbing / Small Likelihood

• Typical approach to small likelihood, used in practice:
  • Start at some initial point with edge weights $p$.
  • Apply hill climbing to the likelihood function, till reaching a maximum.
The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a single local and global maximum in the parameter space.
The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a single local and global maximum in the parameter space.
- Fukami and Tateno (89), Tillier (94): For any tree, the ML point will be unique.
The Likelihood Surface

- For hill climbing to be guaranteed to find the maximum, there must be a **single local and global maximum** in the parameter space.
- Fukami and Tateno (89), Tillier (94): For any tree, the ML point will be **unique**.
- Steel (94): Proofs are erroneous - A simple but pathological **counter example** (multiple maxima on the wrong tree).
The Likelihood Surface

• For hill climbing to be guaranteed to find the maximum, there must be a **single local and global maximum** in the parameter space.

• Fukami and Tateno (89), Tillier (94): For any tree, the ML point will be **unique**.

• Steel (94): Proofs are erroneous - A simple but pathological **counter example** (multiple maxima on the **wrong tree**).

• (94–present): Hill climbing techniques still used. Steel’s counter example is considered too “biologically unrealistic” to warrant concern.
The Likelihood Surface (cont.)

• Rogers and Swofford (99): Simulation Study
The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
  - Data is simulated on a tree.
The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
  - Data is simulated on a tree.
  - Multiple optima are rare...
The Likelihood Surface (cont.)

- Rogers and Swofford (99): Simulation Study
  - Data is simulated on a tree.
  - Multiple optima are rare...
  - ...especially on the correct tree.
The Likelihood Surface (cont.)

• Rogers and Swofford (99): Simulation Study
  • Data is simulated on a tree.
  • Multiple optima are rare...
  • ...especially on the correct tree.

• Goal here: Investigate the problem analytically (joint work with Hendy, Holland, Penny).
Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables).
- Constrained optimization.
Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.
- Systems of polynomial equations.
Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.
- Systems of polynomial equations.
- Analytical solution: very hard in general, even for four taxa.
Maximizing Likelihood on Trees

Tools used

- Hadamard conjugation (Hendy and Penny 93).
- Splits and sequence spectra (change of variables)
- Constrained optimization.
- Systems of polynomial equations.
- Analytical solution: very hard in general, even for four taxa.
- Employing computer algebra and algebraic geometry tools.
# Example: Conservative Data, Two Very Different ML Trees

<table>
<thead>
<tr>
<th>species</th>
<th>observed data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>XXXXXXXXYYY YYY XY YX XY X</td>
</tr>
<tr>
<td>2</td>
<td>XXXXXXXYY YXX YX YY YX X</td>
</tr>
<tr>
<td>3</td>
<td>XXXXXXXYY YXX XY YY XY X</td>
</tr>
<tr>
<td>4</td>
<td>XXXXXXXY YXX XY YY YX Y</td>
</tr>
</tbody>
</table>
Example: Conservative Data, Two Very Different ML Trees
Small Likelihood & Multiple Maxima

- **Small Likelihood** (reminder): Given observed data & a tree, but **not** the edge weights, find the edge weights that maximize the likelihood.
Small Likelihood & Multiple Maxima

• Small Likelihood (reminder): Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.

• Multiple ML points for general case imply small likelihood cannot be solved by hill climbing.
Small Likelihood & Multiple Maxima

- **Small Likelihood** (reminder): Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.

- Multiple ML points for **general case** imply small likelihood cannot be solved by hill climbing.

- Not clear if small likelihood has efficient (worst case) solutions.
Maximum Parsimony (MP)

- **Big Parsimony**: Given the sequence data, find a tree and assignment of sequences to internal nodes that minimizes the number of changes across all edges.
Maximum **Parsimony (MP)**

- **Big Parsimony**: Given the sequence data, find a tree and assignment of sequences to internal nodes that minimizes the number of changes across all edges.

- **Small Parsimony**: Given the sequence data and a tree, find internal assignment(s) that minimizes total number of changes.
Maximum **Parsimony** (MP)

- **Big Parsimony**: Given the sequence data, find a tree and assignment of sequences to internal nodes that minimizes the number of changes across all edges.

- **Small Parsimony**: Given the sequence data and a tree, find internal assignment(s) that minimizes total number of changes.

- MP considered by practitioners easier than ML. Indeed small parsimony has efficient algorithms (Fitch 1971, Sankoff and Cedergren 1983).
Complexity of Reconstruction

Both MP and ML have well-defined objective functions

$\implies$ Reconstruction is a computational problem.
Complexity of Reconstruction

Both MP and ML have well-defined objective functions

⇒ Reconstruction is a computational problem.

Number of trees over \( n \) leaves is exponential in \( n \)

⇒ Cannot exhaustively search all trees.
Complexity: Small MP vs. ML

- Small parsimony is in P.
Complexity: Small MP vs. ML

• Small parsimony is in P.
• Small likelihood – unknown.
Complexity: Big MP vs. ML

Is ML Computationally Intractable?

- **Big MP** known for almost 20 years to be computationally intractable [Day *et al.*, 1986, reduction from vertex cover].

- No such result has been found for **Big ML** to date (2004).

- Tuffley and Steel (1997): Relations between likelihood and parsimony.

- Addario-Berry *et al.* (2003): **Big Ancestral ML** is hard.

- Still, no cigar (and not even close).
Ancestral ML (AML)

- A tree reconstruction method that is “in between” ML and MP.
Ancestral ML (AML)

- A tree reconstruction method that is “in between” ML and MP.
- The goal is to simultaneously find edge weights and assignment of sequences to internal nodes so that the likelihood of the data, given the tree parameters, is maximized.
Ancestral ML (AML)

- A tree reconstruction method that is “in between” ML and MP.
- The goal is to simultaneously find edge weights and assignment of sequences to internal nodes so that the likelihood of the data, given the tree parameters, is maximized.
- AML is widely used in evolutionary studies.
Ancestral ML (AML)

- A tree reconstruction method that is “in between” ML and MP.
- The goal is to simultaneously find edge weights and assignment of sequences to internal nodes so that the likelihood of the data, given the tree parameters, is maximized.
- AML is widely used in evolutionary studies.
- Also termed joint reconstruction of ancestral sequences.
Ancestral ML (AML)

- A tree reconstruction method that is “in between” ML and MP.
- The goal is to simultaneously find edge weights and assignment of sequences to internal nodes so that the likelihood of the data, given the tree parameters, is maximized.
- AML is widely used in evolutionary studies.
- Also termed joint reconstruction of ancestral sequences.
- AML computes the likelihood contribution resulting from best assignment to internal nodes, while “regular ML” sums up over all assignments.
Two AML Versions

- **Big AML**: Given the sequence data, find a tree, assignment to internal nodes, and edge weights that maximize the likelihood of the data.
Two AML Versions

- **Big AML**: Given the sequence data, find a tree, assignment to internal nodes, and edge weights that maximize the likelihood of the data.

- **Small AML**: Given observed data, a tree and edge weights, but not the internal assignment, find the assignment that maximize the likelihood.
Two AML Versions

- **Big AML**: Given the sequence data, find a tree, assignment to internal nodes, and edge weights that maximize the likelihood of the data.

- **Small AML**: Given observed data, a tree and edge weights, but not the internal assignment, find the assignment that maximize the likelihood.

- **PPSG 2000**: A poly time, dynamic programming algorithm for small AML.
Two AML Versions

• **Big AML**: Given the sequence data, find a tree, assignment to internal nodes, and edge weights that maximize the likelihood of the data.

• **Small AML**: Given observed data, a tree and edge weights, but not the internal assignment, find the assignment that maximize the likelihood.

• **PPSG 2000**: A poly time, dynamic programming algorithm for small AML.

• **Remark**: Version where tree is given but no edge weights or assignment is still open.
Two AML Versions

- **Big AML**: Given the sequence data, find a tree, assignment to internal nodes, and edge weights that maximize the likelihood of the data.
- **Small AML**: Given observed data, a tree and edge weights, but not the internal assignment, find the assignment that maximize the likelihood.
- **PPSG 2000**: A poly time, dynamic programming algorithm for small AML.
- **Remark**: Version where tree is given but no edge weights or assignment is still open.
- **ACHLPW 2003**: Big AML is NP-hard.
Useful AML Observation

• Given sequence data, a tree, and assignment to internal nodes.
Useful AML Observation

- Given sequence data, a tree, and assignment to internal nodes.
- The edge weights that maximize the likelihood of the data equal $d_e/k$. 
Useful AML Observation

• Given sequence data, a tree, and assignment to internal nodes.
• The edge weights that maximize the likelihood of the data equal $d_e/k$.
• Where $d_e$ equals the number of changes across edge $e$, and $k$ is the common sequence length.
AML, Reformulated

Previous observation implies

- **Input:** A set $S$ of $n$ binary sequences, each of length
AML, Reformulated

Previous observation implies

- **Input:** A set \( S \) of \( n \) binary sequences, each of length
- **Goal:** Find a tree \( T \) with \( n \) leaves, an assignment \( p : E(T) \rightarrow [0, 1] \) of edge probabilities, and a labelling \( \lambda : V(T) \rightarrow \{0, 1\}^k \) of the vertices such that
AML, Reformulated

Previous observation implies

- **Input:** A set $S$ of $n$ binary sequences, each of length
- **Goal:** Find a tree $T$ with $n$ leaves, an assignment $p : E(T) \rightarrow [0, 1]$ of edge probabilities, and a labelling $\lambda : V(T) \rightarrow \{0, 1\}^k$ of the vertices such that
  1. The $n$ labels of the leaves are exactly the sequences from $S$. 
AML, Reformulated

Previous observation implies

- **Input:** A set \( S \) of \( n \) binary sequences, each of length
- **Goal:** Find a tree \( T \) with \( n \) leaves, an assignment \( p : E(T) \rightarrow [0, 1] \) of edge probabilities, and a labelling \( \lambda : V(T) \rightarrow \{0, 1\}^k \) of the vertices such that

1. The \( n \) labels of the leaves are exactly the sequences from \( S \).
2. the sum of all “edge entropies”
   \[ \sum_{e \in E(T)} H\left(d_e/k\right) \] is minimized.
AML, Reformulated

Previous observation implies

- **Input:** A set $S$ of $n$ binary sequences, each of length

- **Goal:** Find a tree $T$ with $n$ leaves, an assignment $p : E(T) \rightarrow [0, 1]$ of edge probabilities, and a labelling $\lambda : V(T) \rightarrow \{0, 1\}^k$ of the vertices such that
  
  1. The $n$ labels of the leaves are exactly the sequences from $S$.
  2. the sum of all “edge entropies” \( \sum_{e \in E(T)} H(d_e/k) \) is minimized.

- \( H(p) = -p \log(p) - (1 - p) \log(1 - p) \) is the binary entropy function.
AML vs. MP

Optimization criteria

- **Input:** A set $S$ of $n$ binary sequences, each of length $k$. 
AML vs. MP

Optimization criteria

• **Input:** A set $S$ of $n$ binary sequences, each of length $k$.

• AML: Minimize the sum of all “edge entropies”

\[
\sum_{e \in E(T)} H \left( \frac{d_e}{k} \right).
\]
AML vs. MP

Optimization criteria

- **Input:** A set $S$ of $n$ binary sequences, each of length $k$.
- **AML:** Minimize the sum of all “edge entropies”
  $$\sum_{e \in E(T)} H \left( \frac{d_e}{k} \right).$$
- **MP:** Minimize the sum of all “edge differences”
  $$\sum_{e \in E(T)} \frac{d_e}{k}.$$
AML vs. MP

Optimization criteria

- **Input**: A set $S$ of $n$ binary sequences, each of length $k$.
- **AML**: Minimize the sum of all “edge entropies”
  $$\sum_{e \in E(T)} H \left( \frac{d_e}{k} \right).$$
- **MP**: Minimize the sum of all “edge differences”
  $$\sum_{e \in E(T)} \frac{d_e}{k}.$$  
- Can think of the two problems as attempting to minimize different edge weights (functions of $d_e$).
NP hardness of AML: Ideas

• MP was shown NP-hard by Day, Johnson, Sankoff using reduction from vertex cover (VC).
NP hardness of AML: Ideas

- MP was shown NP-hard by Day, Johnson, Sankoff using reduction from vertex cover (VC).
- Analogy of AML and MP optimization criteria suggests using similar approach.
NP hardness of AML: Ideas

• MP was shown NP-hard by Day, Johnson, Sankoff using reduction from vertex cover (VC).
• Analogy of AML and MP optimization criteria suggests using similar approach.
• Reduction from VC indeed identical.
**NP hardness of AML: Ideas**

- MP was shown NP-hard by Day, Johnson, Sankoff using reduction from vertex cover (VC).
- Analogy of AML and MP optimization criteria suggests using similar approach.
- Reduction from VC indeed identical.
- Proof substantially more involved as entropy $H(d_e/k)$ is not as “well behaved” as plain edge differences $d_e/k$. 
Open Problems as of 2004

• Hardness proof for big AML as a stepping stone for big ML?
Open Problems as of 2004

• Hardness proof for big AML as a stepping stone for big ML?
• Is small ML in poly-time?
Our Major Question

Is ML Computationally Intractable?

- **MP** known for almost 20 years to be 
  *computationally intractable* [Day *et al.*, 1986, 
  translation from *vertex cover*].

- No such result has been found for **ML** to date.

- Tuffley and Steel (1997): Relations between 
  likelihood and parsimony.


- Still, no cigar (and *not even close*).
Is ML Computationally Intractable?

- Still, no cigar (and not even close).
- Particularly frustrating in light of intuition among practitioners that ML is harder than MP.
- Maybe some slick and efficient ML algorithm lurks out there, waiting to be discovered?
Is ML Computationally Intractable?

- Still, no cigar (and not even close).
- Particularly frustrating in light of intuition among practitioners that ML is harder than MP.
- Maybe some slick and efficient ML algorithm lurks out there, waiting to be discovered?

CT2005:
ML is computationally hard (NP complete) $\implies$ No such algorithm exists (unless P=NP).
Intractability Proof: The Big Picture

Efficiently translate vertex cover (VC) to ML.
Intractability Proof: The Big Picture

Efficiently translate vertex cover (VC) to ML.
Intractability Proof: The Big Picture

Efficiently translate vertex cover (VC) to ML.

“Translation” means

- **Small** cover $\implies$ **Large** likelihood.
- **Large** cover $\implies$ **Small** likelihood.
Vertex Cover in Graphs

Given a graph \((V, E)\)

- find a **small** set of vertices \(C\)
- such that for each edge in the graph,
- \(C\) contains at least one endpoint.
Vertex Cover in Graphs

Given a graph \((V, E)\)
- find a **small** set of vertices \(C\)
- such that for each edge in the graph, \(C\) contains at least one endpoint.

(figure from www.cc.ioc.ee/jus/gtglossary/assets/vertex_cover.gif)
Well Known: Vertex Cover is Intractable

The decision version of this problem (does $G$ has a cover of size $\leq c$) is computationally intractable.
Well Known: Vertex Cover is Intractable

The decision version of this problem (does $G$ has a cover of size $\leq c$) is computationally intractable.

(figure from http://wwwbrauer.in.tum.de/gruppen/theorie/hard/vc1.png)
Maximum Likelihood: Decision Problem

Input: A set of equi length binary sequences $S_1, S_2, \ldots, S_m$, and a real number, $D$. 

Maximum Likelihood: Decision Problem

**Input:** A set of equi length binary sequences $S_1, S_2, \ldots, S_m$, and a real number, $D$.

**Question:** Is there a tree $T$ and edge lengths such that
\[ \log_2 L(S_1, S_2, \ldots, S_m \mid T, \text{edge parameters}) > D \]
Maximum Likelihood: Decision Problem

Input: A set of equi length binary sequences $S_1, S_2, \ldots, S_m$, and a real number, $D$.

Question: Is there a tree $T$ and edge lengths such that
$\log_2 L(S_1, S_2, \ldots, S_m \mid T, \text{edge parameters}) > D$?

Notice: Yes/No question.
Translating VC to ML

The following graph, with 5 vertices and 6 edges

1 -- 3 -- 2
|     |     |
4     5

Translates to
Translating VC to ML

The following graph, with 5 vertices and 6 edges

Translates to

A set with 7 binary sequences, each of length 5:

00000 10100 10010 01100
01001 00110 00101
Translating VC to ML

- If \( G \) has \( n \) vertices and \( m \) edges, will construct \( m + 1 \) binary sequences, each of length \( n \).
- One sequence is all zeroes.
- For every edge \((i, j) \in E\), have the sequence:

\[
\begin{align*}
&00 \ldots 00 \quad 1 \quad 00 \ldots 00 \quad 1 \quad 00 \ldots 00 \\
&i-1 \quad j-i-1 \quad n-j \quad n
\end{align*}
\]
Relation between likelihood and parsimony (Tuffley and Steel)

\[
L(S|T) \equiv Pr(S|p^*, T) \geq 2^{-\log(k_c) \cdot \text{pars}(S,T) - C^d}
\]

\[
L(S|T) \equiv Pr(S|p^*, T) \leq 2^{-\log(k_c) \cdot \text{pars}(S,T) - C^u}
\]

\[C^u\text{ and } C^d\text{ are sub quadratic functions of the size of } |V(T)|, \text{pars}(S,T), \text{ and } k - k_c.\]
Relation between likelihood and parsimony (Tuffley and Steel)

\[ L(S|T) \equiv Pr(S|p^*, T) \geq 2^{-\log(k_c) \cdot \text{pars}(S,T) - C^d} \]

\[ L(S|T) \equiv Pr(S|p^*, T) \leq 2^{-\log(k_c) \cdot \text{pars}(S,T) - C^u} \]

\( C^u \) and \( C^d \) are sub quadratic functions of the size of \( |V(T)| \), \( \text{pars}(S,T) \), and \( k - k_c \).

**Conclusion**: if \( (k - k_c) = O(|V(T)|) \) (as our case) then

\[ L(S|T) = O(\text{pars}(S,T) \cdot \log(n)) + O(|V(T)|^2) \]

\[ + O(\text{pars}(S,T) \log(\text{pars}(S,T))) \]
Canonical Trees: Definition

1. Tree has an internal node (called the “root”) with 0 length edge to the all zero leaf.
2. All leaves are at distance one or two from the root.
3. Subtrees of distance two leaves contains one, two, or three leaves. All sequences in a subtree with two or three leaves share a “1” in same position.
**Canonical Trees: Definition**

1. Tree has an internal node (called the “root”) with 0 length edge to the all zero leaf.
2. All leaves are at distance one or two from the root.
3. Subtrees of distance two leaves contains one, two, or three leaves. All sequences in a subtree with two or three leaves share a “1” in same position.

---

Yes

No
Canonical Trees and Vertex Covers

[Day, 1986]: A canonical tree with degree $d$ at root exists

$\iff$

$G$ has a cover of size $d$. 
Canonical Trees and Likelihood

Now establish relationship between degree of root of canonical trees and likelihood.
Canonical Trees and Likelihood

Now establish relationship between degree of root of canonical trees and likelihood.

Given a canonical tree \( T \) with \( m + 1 \) leaves labelled by sequences from \( S \). Let \( d \) denote the degree of the root. Then for the optimal edge lengths \( p^* \),

\[
\log(Pr(S \mid T, p^*)) = -(m + d) \cdot \log n + \theta(n).
\]
Canonical Trees and Likelihood

Now establish relationship between degree of root of canonical trees and likelihood.

Given a canonical tree $T$ with $m + 1$ leaves labelled by sequences from $S$. Let $d$ denote the degree of the root. Then for the optimal edge lengths $p^*$,

$$\log(Pr(S \mid T, p^*)) = -(m + d) \cdot \log n + \theta(n).$$

So as $n \to \infty$,

$$\frac{-\log(L)}{(m + d) \log(n)} \to 1.$$
Do We Have the Desired Proof?

\[- \frac{\log(L)}{(m + d) \log(n)} \to_{n \to \infty} 1.\]

Seems to imply

- **Small** cover \(\implies\) **Large** likelihood.
- **Large** cover \(\implies\) **Small** likelihood.
Do We Have the Desired Proof?

\[ -\log(L) \to \frac{(m + d) \log(n)}{\log(n)} \to n \to \infty 1. \]

Seems to imply

- Small cover \implies Large likelihood.
- Large cover \implies Small likelihood.

Take it easy. There is a problem here. What we actually showed is a reduction from VC to ML of canonical trees.
Do We Have the Desired Proof?

\[
- \log(L) \over (m + d) \log(n) \rightarrow_{n \rightarrow \infty} 1.
\]

Seems to imply

- **Small** cover \(\implies\) **Large** likelihood.
- **Large** cover \(\implies\) **Small** likelihood.

Take it easy. There is a **problem** here. What we actually showed is a reduction from VC to **ML** of **canonical trees**.

But **ML** tree **need not** be canonical!
All Is Lost?
All Is Lost?

Not necessarily.
All Is Lost?

- Not necessarily.
- Starting from any ML tree (arbitrary shape).
All Is Lost?

- Not necessarily.
- Starting from any ML tree (arbitrary shape).
- Carry out a sequence of gentle modification.
All Is Lost?

- Not necessarily.
- Starting from any ML tree (arbitrary shape).
- Carry out a sequence of gentle modification.
- Each modification may decrease log likelihood.
All Is Lost?

- Not necessarily.
- Starting from any ML tree (arbitrary shape).
- Carry out a sequence of gentle modification.
- Each modification may decrease log likelihood.
- But only by a little ($B \log n$).
All Is Lost?

- Not necessarily.
- Starting from any ML tree (arbitrary shape).
- Carry out a sequence of gentle modification.
- Each modification may decrease log likelihood.
- But only by a little ($B \log n$).
- Number of modification small enough.
All Is Lost?

- Not necessarily.
- Starting from any ML tree (arbitrary shape).
- Carry out a sequence of gentle modification.
- Each modification may decrease log likelihood.
- But only by a little ($B \log n$).
- Number of modification small enough.
- So accumulated loss in log likelihood is small.
Hardness Conclusion

- Maximum likelihood of phylogenetic tree is **computationally intractable**
- No magic bullet!
Open Problems and Further Research

- Four states characters & beyond. √
Open Problems and Further Research

- Four states characters & beyond. ✓
- Hardness of ML approximation. ✓
Open Problems and Further Research

- Four states characters & beyond. ✓
- Hardness of ML approximation. ✓
- ML hardness under molecular clock. ✓
Open Problems and Further Research

- Four states characters & beyond. ✓
- Hardness of ML approximation. ✓
- ML hardness under molecular clock. ✓
- Efficient approximation algorithms.
Open Problems and Further Research

- Four states characters & beyond. ✓
- Hardness of ML approximation. ✓
- ML hardness under molecular clock. ✓
- Efficient approximation algorithms.
- Efficient algorithms for small likelihood: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.
Open Problems and Further Research

- Four states characters & beyond. ✓
- Hardness of ML approximation. ✓
- ML hardness under molecular clock. ✓
- Efficient approximation algorithms.
- Efficient algorithms for small likelihood: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.
- Regions of input parameters where ML can be efficiently solved.
Open Problems and Further Research

- Four states characters & beyond. ✓
- Hardness of ML approximation. ✓
- ML hardness under molecular clock. ✓
- Efficient approximation algorithms.
- Efficient algorithms for small likelihood: Given observed data & a tree, but not the edge weights, find the edge weights that maximize the likelihood.
- Regions of input parameters where ML can be efficiently solved.