Genomanalyse

Vorlesung Genomanalyse vom 17.1.2012

Ingo Bulla
Institut für Mathematik und Informatik
Universität Greifswald
References

• "Biological Sequence Analysis", Durbin, Eddy, Krogh, Mitchison, Cambridge University Press, 1998
• "Inferring phylogenies", Felsenstein, Sinauer, 2004
**Phylogeny**

**Definition (Phylogeny)**
Phylogeny is the study of evolutionary relatedness among various groups of organisms (e.g., species or populations).

**Main concept**
Phylogenetic tree
Types of phylogenetic trees

Categories

- rooted / unrooted
- with / without distances
Example application

Facts about HIV / AIDS

- HIV developed in Africa in the early 20th century
- AIDS was first reported in 1981 in the USA

Question

How did HIV get to the USA?
# Parsimony

## Principle
Find tree explaining observed sequences with minimal number of mutations

## Properties
- simple and fast
- all mutations have unit cost
- neglects lots of information (in particular, branch lengths)

## Tasks
- compute costs for given tree $T$ (Fitch algorithm)
- find tree with minimal cost (by searching through all trees)

## Assumptions
- sequences are aligned, gapless and have the same length
- sites evolve independently
Minimum parsimony problem

**Definition**

Given

- a rooted tree $T$ with characters from an alphabet $\Sigma$ attached to its leaves,

find

- a labelling of the internal nodes of $T$ with characters

such that

- the overall number of character changes along the edges of $T$ is minimized.
Fitch algorithm

Bottom-up phase

Assign to each node a set of potential characters.

- For each leaf node $i$ set: $R_i := \{x_i\}$, with $x_i$ the character at leaf $i$.
- For each internal node $i$ with children $j, k$, set: $R_i := \begin{cases} \quad R_j \cap R_k, & \text{if } R_j \cap R_k \neq \emptyset \\ R_j \cup R_k, & \text{otherwise} \end{cases}$
Fitch algorithm

Top-down phase

Pick a character for each internal node.

- For the root, pick $x_{root} \in R_{root}$
- For each internal node $i$ with parent $j$, set $x_i := \begin{cases} x_j, & \text{if } x_j \in R_i \\ \text{some } x \in R_i, & \text{otherwise} \end{cases}$
Fitch algorithm

Top-down phase

Pick a character for each internal node.

- For the root, pick $x_{root} \in R_{root}$
- For each internal node $i$ with parent $j$, set $x_i := \begin{cases} x_j, & \text{if } x_j \in R_i \\ \text{some } x \in R_i, & \text{otherwise} \end{cases}$
Generalization

Definition
Let $S(x, y)$ be the costs for a mutation from $x \in \Sigma$ to $y \in \Sigma$.

Simple Parsimony
$S(x, x) = 0$ for $x \in \Sigma$
$S(x, y) = 1$ for $x, y \in \Sigma, x \neq y$

Weighted Parsimony
$S(x, x) = 0$ for $x \in \Sigma$
$S(x, y) > 0$ for $x, y \in \Sigma, x \neq y$
### Find optimal tree

#### Heuristic methods
- Subtree pruning and regrafting
- Branch-swopping
- Iterative construction
- Clustering (UPGMA)

#### Method finding the optimum
- Branch and Bound
Subtree pruning and regrafting

Steps

1. Construct an initial tree
Subtree pruning and regrafting

Steps

1. Construct an initial tree
2. Prune a random subtree
## Subtree pruning and regrafting

### Steps

1. Construct an initial tree
2. Prune a random subtree
3. Regraft the subtree at a random edge
4. If cost of trees decreased, accept new tree, otherwise revise change
5. Go back to 2.
Subtree pruning and regrafting

Steps

1. Construct an initial tree
2. Prune a random subtree
3. Regraft the subtree at a random edge

1 2 3 4 5
Subtree pruning and regrafting

Steps

1. Construct an initial tree
2. Prune a random subtree
3. Regraft the subtree at a random edge
4. If cost of trees decreased, accept new tree, otherwise revise change
5. Go back to 2.
Branch-swopping

Steps

1. Construct an initial tree
Branch-swopping

Steps

1. Construct an initial tree
2. Prune two random subtree

Diagram:

```
  1
 /|
/  |
  2
 /
/ |
  3
/
/ |
  4
 /
/  |
  5
 /
/  |
  6
 /
/  |
  7
```

Steps:
1. Construct an initial tree
2. Prune two random subtree
3. Switch the two subtrees
4. Regraft them
5. If cost of trees decreased, accept new tree, otherwise revise change
6. Go back to 2.
Branch-swopping

Steps

1. Construct an initial tree
2. Prune two random subtree
3. Switch the two subtrees
Branch-swopping

Steps

1. Construct an initial tree
2. Prune two random subtree
3. Switch the two subtrees
Branch-swopping

Steps

1. Construct an initial tree
2. Prune two random subtree
3. Switch the two subtrees
4. Regraft them
**Branch-swopping**

**Steps**

1. Construct an initial tree
2. Prune two random subtree
3. Switch the two subtrees
4. Regraft them
5. If cost of trees decreased, accept new tree, otherwise revise change
6. Go back to 2.
Iterative construction

Steps

1. Construct a tree of the first 3 sequences
2. Attach the next sequence at the edge resulting in the lowest cost
Iterative construction

Steps

1. Construct a tree of the first 3 sequences
2. Attach the next sequence at the edge resulting in the lowest cost
## Iterative construction

### Steps

1. Construct a tree of the first 3 sequences
2. Attach the next sequence at the edge resulting in the lowest cost
3. If sequences are left, go back to 2.

![Tree diagram](image)
Branch and Bound

**Notation**
Let $n$ sequences $\{x^1, \ldots, x^n\}$ be given.

**Idea**
- Enumerate all unrooted trees by an array

$$[i_3][i_5][i_7] \cdots [i_{2n-5}],$$

with $i_k \in \{1, \ldots, k\}$ ('milometer').
Branch and Bound

Correspondance array → unrooted tree

Example: \([i_3][i_5] \ldots = [2][1] \ldots\)

- Take unrooted tree \(T_3\) composed of \(x^1, x^2, x^3\)
- Label the edges of \(T_3\) with 1, 2, 3
- Add an edge for \(x^4\) on edge labelled \(i_3\)
- Label the newly generated edges with 4 and 5
- Add an edge for \(x^5\) on edge labelled \(i_5\)
- Label the newly generated edges with 6 and 7
- (proceed until \(x^n\))
Branch and Bound

Let $n$ sequences $\{x^1, \ldots, x^n\}$ be given.

**Idea**

- Enumerate all unrooted trees by an array
  
  $$[i_3][i_5][i_7] \ldots [i_{2n-5}],$$

  with $i_k \in \{1, \ldots, k\}$ (‘milometer’).

- If a partial tree $T_k$ composed of $x^1, \ldots, x^k$ achieves higher cost than the best complete tree so far, discard all complete trees obtained by attaching $x^{k+1}, \ldots, x^n$ to $T_k$. 

# Branch and bound

## Example (Milometer)

<p>| | | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7</td>
</tr>
</tbody>
</table>

1. **Branch and bound**

2. **Example (Milometer)**

   - **3 4 3** ...... **7 0 0 0 0**
   - **3 4 3** ...... **7 1 0 0 0**
   - **3 4 3** ...... **8 0 0 0 0**
Clustering (UPGMA)

**UPGMA**
Unweighted pair group method using arithmetic averages

**Idea**
Cluster sequences, at each stage
- amalgamating two clusters
and
- creating a new node on a tree
at the same time.

**Given**
Measure for distance between two sequences $x^i$ and $x^j$

**Example**
- fraction $f$ of sites $u$ where $x^i_u$ and $x^j_u$ differ
- Jukes-Cantor distance $d_{ij} = -\frac{3}{2} \log(1 - 4 \frac{f}{3}) \ (\rightarrow \infty \ for \ random \ sequences)$

**Definition**
The distance $d_{ij}$ between two clusters $C_i$ and $C_j$ is given by

$$d_{ij} = \frac{1}{|C_i||C_j|} \sum_{p \ in \ C_i} \sum_{q \ in \ C_j} dpq.$$
**Algorithm UPGMA**

### Initialization
- Assign each sequence \( i \) to its own cluster \( C_i \).
- Define one leaf of \( T \) for each sequence, place it at height zero.

### Iteration
- Determine the two clusters \( i \) and \( j \) for which \( d_{ij} \) is minimal.
- Define a new cluster \( k \) by \( C_k = C_i \cup C_j \).
- Define a node \( k \) with daughter nodes \( i \) and \( j \), place it at height \( \frac{1}{2} d_{ij} \).
- Add \( k \) to the current clusters and remove clusters \( i \) and \( j \).
Algorithm UPGMA

**Initialization**
- Assign each sequence $i$ to its own cluster $C_i$.
- Define one leaf of $T$ for each sequence, place it at height zero.

**Iteration**
- Determine the two clusters $i$ and $j$ for which $d_{ij}$ is minimal.
- Define a new cluster $k$ by $C_k = C_i \cup C_j$.
- Define a node $k$ with daughter nodes $i$ and $j$, place it at height $\frac{1}{2}d_{ij}$.
- Add $k$ to the current clusters and remove clusters $i$ and $j$. 
Algorithm UPGMA

Initialization

- Assign each sequence $i$ to its own cluster $C_i$.
- Define one leaf of $T$ for each sequence, place it at height zero.

Iteration

- Determine the two clusters $i$ and $j$ for which $d_{ij}$ is minimal.
- Define a new cluster $k$ by $C_k = C_i \cup C_j$.
- Define a node $k$ with daughter nodes $i$ and $j$, place it at height $\frac{1}{2}d_{ij}$.
- Add $k$ to the current clusters and remove clusters $i$ and $j$. 
Algorithm UPGMA

**Initialization**
- Assign each sequence \( i \) to its own cluster \( C_i \).
- Define one leaf of \( T \) for each sequence, place it at height zero.

**Iteration**
- Determine the two clusters \( i \) and \( j \) for which \( d_{ij} \) is minimal.
- Define a new cluster \( k \) by \( C_k = C_i \cup C_j \).
- Define a node \( k \) with daughter nodes \( i \) and \( j \), place it at height \( \frac{1}{2} d_{ij} \).
- Add \( k \) to the current clusters and remove clusters \( i \) and \( j \).

**Termination**
- When only two clusters \( i \) and \( j \) remain, place the root at height \( \frac{1}{2} d_{ij} \).
## Bootstrap (phylogeny)

### Aim

Estimate amount of uncertainty about an inferred tree
**Bootstrap (real-valued distribution)**

**Given**
Sample $x_1, \ldots, x_n$ from unknown real-valued distribution $D$

**Aim**
Estimate expected value $\theta$ of $D$

**Approach**
- Sampling from empirical distribution given by $x_1, \ldots, x_n$ with replacement ($m$ samples of length $n$) $\rightarrow$ $m$ bootstrap replicates
- Estimate $\theta$ from $m$ bootstrap replicates $\rightarrow$ estimates $\theta_1^*, \ldots, \theta_m^*$
- Analyze $\theta_1^*, \ldots, \theta_m^*$
Example (real-valued distribution)

**Approach**

- Sampling \( m \) times from empirical distribution given by \( x_1, \ldots, x_n \) with replacement → bootstrap replicates
- Estimate \( \theta \) from \( m \) bootstrap replicates → estimates \( \theta_1^*, \ldots, \theta_m^* \)
- Analyze \( \theta_1^*, \ldots, \theta_m^* \)

Unknown true value of \( \theta \)

Sampling

Unknown true distribution

Bootstrap replicate #1

Empirical distribution of sample

Bootstrap replicate #2

Bootstrap replicate #3

Distribution of estimates of \( \theta \)
## Bootstrap (phylogeny)

### Aim
Estimate amount of uncertainty about an inferred tree

### Given
Gapless alignment of \( k \) sequences of length \( n \)

### Assumption
Different sites of alignment evolve independently

### Approach
- Sampling from columns of original alignment with replacement (\( m \) samples with \( n \) columns each) \( \rightarrow m \) alignments \( A_1, \ldots, A_m \) of same size as original one
- Infer most plausible tree for each alignment \( A_1, \ldots, A_m \) \( \rightarrow m \) trees \( T_1, \ldots, T_m \)
- Analyze \( T_1, \ldots, T_m \)
Bootstrap (phylogeny)

**Approach**

- Sampling from columns of original alignment with replacement ($m$ samples with $n$ columns each) $\rightarrow m$ alignments $A_1, \ldots, A_m$ of same size as original one
- Infer most plausible tree for each alignment $A_1, \ldots, A_m$ $\rightarrow m$ trees $T_1, \ldots, T_m$
- Analyze $T_1, \ldots, T_m$
Limits of parsimony

Evaluation Method

- Generate tree
- Simulate mutation process (probabilistic model)
  - Choose position of root (arbitrarily)
  - Generate sequence at root
  - Simulate mutation process from root to each leaf
- Reconstruct tree using parsimony, based on simulated leaf sequences
Limits of parsimony

Setup

- Original tree: 4 leaves, 2 short edges, 2 long edges
- 3 possible tree topologies: $T_1$ (original), $T_2$, $T_3$
- Simulate sequences of length $N$ 1000 times

Results

<table>
<thead>
<tr>
<th>N</th>
<th>$T_1$</th>
<th>$T_2$</th>
<th>$T_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>396</td>
<td>378</td>
<td>224</td>
</tr>
<tr>
<td>100</td>
<td>405</td>
<td>515</td>
<td>79</td>
</tr>
<tr>
<td>500</td>
<td>404</td>
<td>594</td>
<td>2</td>
</tr>
<tr>
<td>2000</td>
<td>353</td>
<td>646</td>
<td>0</td>
</tr>
</tbody>
</table>
Limits of parsimony

Results

<table>
<thead>
<tr>
<th>N</th>
<th>T₁</th>
<th>T₂</th>
<th>T₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>396</td>
<td>378</td>
<td>224</td>
</tr>
<tr>
<td>100</td>
<td>405</td>
<td>515</td>
<td>79</td>
</tr>
<tr>
<td>500</td>
<td>404</td>
<td>594</td>
<td>2</td>
</tr>
<tr>
<td>2000</td>
<td>353</td>
<td>646</td>
<td>0</td>
</tr>
</tbody>
</table>