Phylogenetic inference

After this lecture, you can...

... discuss (dis-) advantages of different information types for making trees in light of phylogenetic resolution and noise

... explain four different tree-building algorithms (UPGMA, NJ, MP, ML) and list their (dis-) advantages

... convert between cladograms and Newick tree format

... root unrooted trees in four ways

... identify informative positions in an alignment

... calculate the number of possible trees with $n$ leaves

... explain bootstrapping/jackknifing and their assumptions

... interpret branch support values
Characters used for phylogenetic inference

• Phenotypic characters
  – Infinite number of features
  – Subjective choice
  – Value can depend on observation (etc.)

• Sequence (protein/DNA)
  – Gene/genome is finite
  – Objective choice
  – A sequence is absolute

Phylogenetic resolution

• Use variable regions to compare closely related sequences
  – Badly conserved sequences contain too much noise to resolve distant relationships

• Use conserved regions to compare distantly related sequences
  – Highly conserved sequences contain too little information to resolve close relationships
Trimming alignments

- Conserved regions can be more confidently aligned than variable regions
- Variable regions can add noise to an alignment
- To solve this, badly aligned regions can be trimmed before further analysis

Gblocks eliminates poorly aligned positions and divergent regions of a DNA or protein alignment

Types of tree-building algorithms

- Distance-based approaches
  - Fastest programs for making phylogenetic trees
  - Unweighted Pair Group Method with Arithmetic mean (UPGMA)
  - Neighbor Joining (NJ)
- Maximum parsimony (MP) approaches
  - Assume the minimal number of changes or evolutionary events
- Maximum likelihood (ML) approaches
  - Depend on an explicit model of evolution
  - Considered the most reliable way to infer phylogenies
**Phylogenies based on distance matrices**

Multiple sequence alignment

Evolutionary distance matrix

Phylogenetic tree

**Calculate evolutionary divergence**

Jukes-Cantor correction

Cluster: UPGMA, Neighbor Joining

**Evolutionary distances**

- Sequence (dis-)similarity represents evolutionary distance
  - Use similarity quantification methods from last week’s lectures
- But: evolutionary distance does not correlate 1:1 with sequence alignment score
  - Because mutations at the same position in the sequence become increasingly likely
  - So we have to correct for that: \( d = \frac{3}{4} \ln(1 - \frac{4}{3}D) \)
UPGMA algorithm

Phylogenetic trees can be written as a bracket-notation – Also known as Newick tree format:

```plaintext
```

- Multifurcating branches could be included in the Newick tree format
- For example if the `(B,F,G)` node has very low support

```plaintext
(((A:4.0,D:4.0):4.25,(B:0.5,F:0.5):7.75,G:8.25):6.25,C:14.5):2.5,E:17.0);
```
**Placement of the root**

- The last thing that is added is the root
- UPGMA assumes that the molecular clock holds
  - All tips have equal distance to the root (this is called ultrametric)

- You can tell if the tree is rooted or unrooted by looking at the number of most basal groups:
  - With root: 2 most basal groups
  - Unrooted: 3 most basal groups

```
```

**Question**

1. The numbers indicated in the tree above are branch lengths.
   a) What is a common unit of branch length in molecular phylogenies?
   b) Assume that the molecular clock holds. Fill in the missing branch lengths.
   c) What algorithm was used for building this phylogenetic tree?
   d) What are $d_{AB}$ and $d_{CD}$?
   e) Write this tree in Newick tree format with branch lengths.

2. Research has revealed that the molecular clock does not hold for the lineage leading to C. If $d_{BC} = 6$, what is the distance between C and its last common ancestor with A, B, D, and E?
1. Assume that the molecular clock holds
   a) Mutations per sequence site
   b) Missing branch lengths are now included in the tree above
   c) Unweighted Pair Group Method with Arithmetic mean (UPGMA)
   d) $d_{AB} = 4$, $d_{CD} = 7$
   e) $(( (A:2, B:2) :1, (D:1.5, E:1.5) :1.5) :0.5, C:3.5)$

2. If $d_{BC}$ is 6, then the branch length of C to its last common ancestor with A, B, D, and E is: $6 - 2 - 1 - 0.5 = 2.5$ (in stead of 3.5)

1. The following bracket-notations are also correct (rotated branches):
   e) $(( (A:2, B:2) :1, (D:1.5, E:1.5) :1.5) :0.5, C:3.5)$
   e) $(C:3.5, ( (D:1.5, E:1.5) :1.5, (B:2, A:2) :1) :0.5)$
   e) $(C:3.5, ((B:2,A:2):1, (D:1.5, E:1.5) :1.5) :0.5)$
   e) $(C:3.5, ((A:2,B:2) :1, (E:1.5, D:1.5) :1.5) :0.5)$

*Et cetera*
Non-uniform molecular clock

- Ultrametric algorithms only work if the clock runs at the same speed in all branches
  - All distances to the root are equal
- This is often not the case:

Species A (fast evolving)
Species B (slow evolving)
Species C (fast evolving)
Species D (slow evolving)

Unequal rates of evolution are the rule

- Neighbor Joining (NJ) is designed to account for a non-uniform molecular clock
### Neighbor joining algorithm

- Before we know the tree, the distances between all nodes are represented as a star

1. From a regular distance matrix $d_{ij}$ to a modified distance matrix $e_{ij}$ between all nodes ($N = 5$)

   ![Modified Distance Matrix](image)

   $d_{ij}$  $U_i = \Sigma d_{ij}$  $e_{ij}$  $e_0 = d_{ij} \left( \frac{U_i + U_j}{N-2} \right)$

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>26</td>
<td>29</td>
<td>22</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>B</td>
<td>29</td>
<td>26</td>
<td>22</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>C</td>
<td>22</td>
<td>22</td>
<td>26</td>
<td>33</td>
<td>30</td>
</tr>
<tr>
<td>D</td>
<td>33</td>
<td>33</td>
<td>33</td>
<td>26</td>
<td>29</td>
</tr>
<tr>
<td>E</td>
<td>30</td>
<td>30</td>
<td>30</td>
<td>29</td>
<td>26</td>
</tr>
</tbody>
</table>

2. Join the two closest neighbors in the modified matrix via a new node $W$, and calculate their distances to $W$

   ![Joining Neighbors](image)

   \[d_{Wij} = \frac{d_{ij} + (U_i - U_j)}{(N - 2)}\]

   \[d_{Wj} = \frac{d_{ij} + (U_i - U_j)}{(N - 2)}\]

3. Calculate distances between new node $W$ and all remaining nodes $i$ (example for $i = A$)

   ![Calculating Distances](image)

   \[d_{W} = \frac{d_{ij} + d_{ik} + d_{kj}}{2}\]

   \[d_{W} = \frac{9 + 5}{2} = 7\]

4. Iterate steps 1, 2, and 3 until all neighbors are joined into a single tree

   ![Iterate Steps](image)

- **NJ accounts for different rates of evolution**
  - Evolutionary distances between all nodes stored in branch lengths
Rooting trees using an outgroup
• Distantly related species or gene (homolog)

• Gene duplication event

Rooting trees using prior knowledge
• You can also root trees using your prior knowledge
  – “I know that the root lies between X and Y”
Rooting trees using midpoint rooting

- If all else fails, you can also assume that the root lies halfway between the most distant tips
  - UPGMA takes this approach

\[
\frac{(4 + 1 + 2 + 1)}{2} = 4
\]

Phylogenies based on models of evolution

- Evolutionary events happen on the branches of the tree
- Their probabilities are quantified in a model of evolution
- Some algorithms search all possible trees for the one that best agrees with a given model of evolution
  - Some trees explain the data (= the alignment) better than others
  - Maximum parsimony (MP): the tree that assumes the fewest evolutionary events on the branches to explain the alignment
    - This is a very simplistic model of evolution
  - Maximum likelihood (ML): gives all evolutionary events a probability, finds the tree where total probability is optimized
    - ML can incorporate very sophisticated models of evolution
Maximum parsimony (MP)

• MP example for an alignment in 5 species:

<table>
<thead>
<tr>
<th>Species</th>
<th>Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chimpanzee</td>
<td>AACT T</td>
</tr>
<tr>
<td>Gibbon</td>
<td>AACT T</td>
</tr>
<tr>
<td>Gorilla</td>
<td>AACT T</td>
</tr>
<tr>
<td>Human</td>
<td>AACT T</td>
</tr>
<tr>
<td>Orangutan</td>
<td>AACT T</td>
</tr>
</tbody>
</table>

• Draw all possible trees for the sequences/species present in your multiple alignment

• For each tree, identify where the mutations have taken place
  – Make parsimony assumption: minimum number of required mutations (*)

Maximum parsimony (MP)

• The MP tree has the minimum number of required mutations
  – It is the simplest explanation of the alignment
  – Informative positions contain ≥2 different characters ≥2 × each
Maximum likelihood (ML)

- The simplest explanation (MP) is not always the most likely
  - Some types of mutations are more likely than others
- ML scans all possible trees for the tree that optimizes the probabilities of different events on all the branches
- Models of evolution in ML can be very sophisticated, scoring the likelihood of evolutionary events like:
  - Substitutions between different amino acids
    - E.g. the BLOSUM matrix quantifies the likelihood of all substitutions
  - Insertion or deletion events and their lengths
    - E.g. fixed, linear, or affine gap penalties
  - Site-specific mutation rates
    - E.g. identify fast/slow evolving positions in alignment
    - Give low likelihood to mutations at highly conserved positions
  - Faster/slower evolving lineages in the tree
    - E.g. use the alignment to identify lineages where mutations are more likely
    - Make mutations on those branches “cheaper”
  - Dependency between certain alignment positions (?)

Drawing all possible trees? Really?

How many trees are there?

- For 5 tips, the number of rooted/unrooted trees is: 105 / 15
- For 10 tips, the number of rooted/unrooted trees is: 34,459,425 / 2,027,025
- For 15 tips, the number of rooted/unrooted trees is: 213,458,046,676,875 / 7,905,853,580,625

- # unrooted trees \( N_u \): \( (2n - 5)!! = (2n - 5) \times (2n - 7) \times ... \times 1 \)
- # rooted trees \( N_r \): \( (2n - 3)!! = (2n - 3) \times (2n - 5) \times (2n - 7) \times ... \times 1 \)
- Note: the root is an additional node in the tree: \( N_r(n) = N_u(n + 1) \)

- Fast algorithms use heuristics
  - Heuristic search of “tree-space”
    - Start with a tree (e.g. NJ or random tree)
    - Change a bit (e.g. swap two branches)
    - Accept changes if likelihood goes up
  - So only a subset of all trees need to be evaluated

Searching “tree space”
Which phylogeny approach to use?

• A phylogenetic tree is a hypothesis of the evolutionary history
  – Just like for multiple sequence alignments, the tree is an algorithm's best guess based on the given model of evolution

• Distance trees are fast and give you a quick idea of the tree
  – NJ better than UPGMA because unequal rates of evolution are common

• MP is rarely used to infer molecular phylogenies
  – Trees are not good, the model of evolution is too simplistic

• ML trees are slower but tend to give the most reliable trees

• If you know the true branching order for some lineages, you can test which tree building algorithm best recovers them

Garbage in = garbage out!

• Remember: any sequences will be aligned by an alignment program
  – Even of the sequences are not homologous, and thus the “multiple sequence alignment” is meaningless

• Similarly, any “alignment” will be turned into a “tree” by a phylogeny program
  – Even if the sequences are not homologous or if they are very badly aligned, and thus the “phylogenetic tree” is meaningless

• Solutions:
  – Check your multiple sequence alignment carefully before making a phylogenetic tree
  – If the tree shows unexpected branching order, think twice about your methods before interpreting it as a true biological event
Phylogenetic inconsistencies

- The phylogenies of different genes from the same genomes can be inconsistent

- This can be the result of:
  - Evolution of the gene is different than the evolution of the genome
    - Horizontal gene transfer
    - Unrecognized paralogy
  - Technical issues
    - Bad model of evolution
    - Bad alignments
    - Bad phylogenies
  - Biological noise
    - Mutational saturation: multiple mutations at the same sequence site
    - Different rates of evolution in different lineages (inconsistent molecular clock)

Branch support

- Support values show you how reliable a branching split is

- How do we calculate these values?
  - Bootstrap and jackknife statistics
    - These are a type of permutation statistic
  - Posterior probabilities
    - Based on a model of evolution

Support*

- 100%
- 50%
- 1%

*more about this now

http://epidemic.bio.ed.ac.uk/how_to_read_a_phylogeny
Statistics

- We use statistics to assess confidence in an experiment
  - We repeat the experiment $N$ times and test how robust the result is
  - How much variation is there in the result?

Bootstrapping

- We have already used all the data to create the phylogenetic tree, so how can we get information about the confidence of the nodes in the tree?
- Bootstrap re-sampling or bootstrapping:
  - *Bootstrap* (verb): “To pull oneself up by one's bootstraps” means “to better oneself without external help”
  - A multiple alignment consists of many observations (i.e. the positions or columns)
  - From these many observations, we can randomly re-sample new datasets
    - Create new phylogenies from the re-sampled alignments
    - Are the branchings in the tree are robust in these new datasets?
Like many alignment algorithms, bootstrapping/jackknifing assume that all positions in the alignment evolve independently (which is not true).

Bootstrap re-sampling

- Randomly re-sample columns (positions) from a multiple alignment
  - The number of sampled columns is identical to the number of positions in the original alignment
  - Sampling is done with replacement, so some positions can be sampled multiple times, while other positions are never chosen
- Calculate a bootstrap tree based on the randomly re-sampled alignment
  - Use the same phylogenetic approach
- Repeat this e.g. 100-1,000 times
  - This means making 100-1,000 trees
- For each branch in the original tree, in what percentage of the bootstrap trees was it correctly recovered?
  - This is the bootstrap support of the branch
Jackknife re-sampling

- Randomly select a percentage of the columns (positions) from a multiple alignment
  - For example, 50% of the positions in the original alignment
  - Sampling is done without replacement
- Calculate a jackknife tree based on the randomly re-sampled alignment
  - Use the same phylogenetic approach
- Repeat this e.g. 100-1,000 times
  - This means making 100-1,000 trees
- For each branch in the original tree, in what percentage of the jackknife trees was it correctly recovered?
  - This is the jackknife support of the branch

Interpreting bootstrap values

- Any branch with 100% support is certain
  - This means that the species within it were always found together as a cluster
  - No other sequences belong to that cluster
- Exam question:
  - Where might *C. elegans* CDKH be found in at least one of the bootstrap trees?
  - At 15 ★ positions!
Exam question

- What is the maximum number of bootstrap trees where Zebrafish_1A and Stickleback_1A could have formed a clade of two leaves?
- What is the maximum number of bootstrap trees where any clawed frog sequence could have formed a clade of two leaves with any stickleback sequence?