Evolutionary Trees

Challenges in Computational Biology

DNA
Genome Assembly
Regulatory motif discovery
Gene Finding
Evolutionary Theory
Sequence alignment
Database lookup
RNA transcript
Inference
Protein network analysis
Emerging network properties

Main Topic: Evolutionary Trees

(Solved ?)Questions
• Panda
  – Bear or raccoon?
• Out of Africa
  – mitochondrial evolution story?
• Human evolution
  – Did we ever meet Neanderthal?
• Primate evolution
  – Are we chimp-like or gorilla-like?
• Vertebrate evolution
  – How did complex body plans arise?
• Recent evolution
  – What genes are under selection?

Inferring Phylogenies
Trees can be inferred by several criteria:
– Morphology of the organisms
– Sequence comparison

Example:
Kangaroo: ACASTGACCCCCAACAGT
Elephant: ACASTGACCTAACACAGT
Dog: CCTGTGACGTAACAAACGA
Mouse: CCTGTGACGTAGCAAACGA
Human: CCTGTGACGTAGCAAACGA

Evolutionary Mechanisms
• Types of mutations
  – Single substitution: A to C, G or T, etc.
  – Deletion: 1 bp ... chromosomes (aneuploidy)
  – Duplication: as above (often at tandem repeats)
  – Inversion: ABCDEFG to ABCDEF
  – Translocation: ABCD & WXYZ to ABYX & WXCD
  – Insertion: ABCD ΙΝΞΕΡΤ CD
  – Recombination: ABCDEFGH to ABDEFGH
    ⇒ ABCDEFGH ΛÆ ABDEFGH
    ⇒ ABCDEFGH ΛÆ ABCDEFGH
Tree construction

• Basic principle:
  Degree of sequence difference is “proportional” to
  length of independent sequence evolution

• Basic approach:
  – Choose the gene(s)
  – Align the sequences
  – Ignore the parts which contain many gaps
  – Reconstruct the tree based on the number/type of
    substitutions

Approaches

• Distance-based methods
• Parsimony
• Probabilistic models

Distance based methods

• For any aligned sequences $x, y$, need to defined a
  distance $D(x, y)$
• One possible definition:
  $D(x, y) = \text{number of positions } u \text{ where } x[u] \neq y[u]$ (Hamming distance)
• Other scoring methods (Lectures 2, 3)
• For now we assume some $D(x, y)$
• For sequences $x^1 \ldots x^n$, use $D_{ij}$ to denote distance
  between $x^i, x^j$

A simple clustering method for building tree

• UPGMA (unweighted pair group method using
  arithmetic averages)
• Essentially, the Average-Link method from Lecture 8:
  – Given two disjoint clusters $X, Y$:
    $D(X,Y) = \frac{1}{2} \sum_{i \in X, j \in Y} D_{ij}$
  – Keep merging the closest clusters

Algorithm: UPGMA

• Initialization:
  – Assign each $x_i$ into its own cluster $C_i$
  – Define one leaf per sequence, height 0

• Iteration:
  – Find two clusters s.t. $D(C_i, C_j)$ is min
  – Let $C_k = C_i \cup C_j$
  – Define node connecting $C_i, C_j$, place it at
    height $D / 2$
  – Delete $C_i, C_j$

• Termination:
  When two clusters $C_k, C_l$ remain, place
  root at height $D / 2$

Ultrametric Distances & UPGMA

• What can we say about this method?
• It generates trees with very special property:
  – All distances from root to leaves are equal
  – Molecular clock assumption - time is constant for all species
• Distance metric which can be represented in this way is
called an ultrametric
Ultrametrics

- Property: for any 3 points with indices i, j, k
  - Two distances are equal, and the third one is smaller
  - (Almost) equivalently
    \[ D_{ij} \leq \max(D_{ik}, D_{kj}) \]
- UPGMA will reconstruct the proper tree if D is an ultrametric (for binary tree)

Weakness of UPGMA

- Molecular clock assumption: implies time is constant for all species
- However, certain species (e.g., mouse, rat) evolve much faster
- Example where UPGMA messes up:

Fixes

- Neighbor-joining method
  - Similar to UPGMA
  - Guarantees correct reconstruction of the distance metric is induced by a tree
- Parsimony
  - So far we used the distance information
  - We can use the sequences as well!

Parsimony

- Find the tree that explains the observed sequences with a minimal number of substitutions (possibly weighted)
  - Each internal node is labeled by a sequence
  - The cost of each edge is the Hamming distance between labels
  - Want to minimize the total sum of edge costs
- Two computational sub-problems:
  1. Find the internal labels minimizing parsimony cost of a given tree (easy)
  2. Search through all tree topologies (hard)

“Small” Parsimony Problem

- Given:
  - the tree and sequences in the leaves
- Goal: find internal labels that minimize sum of the costs
- Observation: can focus on one character at a time!
- In fact, we will solve the problem for general distance \( D(\cdot) \) between characters

Algorithm

- Dynamic programming:
  - For a node \( v \) and label \( L \), define \( \text{Cost}(v, L) \) to be the minimum cost of \( v \)'s sub-tree when \( v \) is labeled with \( L \)
  - Use recursion
    \[
    \text{Cost}(v, L_v) = \min_{u_1} D(L_{u_1}, L_v) + \text{Cost}(u_1, L_{u_1}) + \min_{u_2} D(L_{u_2}, L_v) + \text{Cost}(u_1, L_{u_2})
    \]
  - Time \( O(nk) \), where
    - \( n \): number of nodes
    - \( k \): alphabet size (4 for \{A,G,T,C\})
Uniform costs

- For all leaves $k$ set $R_k = \{L_k\}$
- For all non-leaves:
  - Let $i, j$ be the children of $k$
  - Set $R_k = R_i \cap R_j$ if intersection is nonempty
  - Set $R_k = R_i \cup R_j$ and $C += 1$, if intersection is empty