Multiple Sequence Alignment

Leif Wickland

CS580 Computational Science

Spring 2005

What is sequence alignment?

- Sequence alignment is an positioning of two or more strings of elements in order to highlight similarity
- For example, the following is a sequence alignment
  - QUICKFOX**--**JUMPED
  - QUICKFOX**ES**JUMPED
  - QUICKFOX**EN**DUMPED
- Dashes are often used to indicate gaps

What is sequence alignment?

- In computational biology, the elements in sequences are usually proteins or DNA
- Types of sequence alignment
  - Pairwise alignment
    - Aligning two sequences at once
  - Multiple alignment
    - Aligning many sequences

What is sequence alignment?

- Types of sequence alignment
  - Global alignment
    - Aligns entire sequences using all elements in the sequences
    - Matched subsequences must appear in the same order in all sequences
  - Local alignment
    - Aligns a region of a sequence with a region of another sequence
    - Matched subsequences may appear in different orders and may overlap

Why use sequence alignment?

- Find similarity among proteins to group them into families
  - Can be used to infer protein structure
  - Can be used to find distant relations
- Used as an aid to the analysis of evolutionary history

How is sequence alignment done?

- Standard practice is to use dynamic programming
  - Use an element match fitness table to score the alignment as you go
    - E.g., PAM250, BLOSUM62
  - Apply score penalties for insertions or deletions
  - Produces a mathematically optimal alignment
  - Good approach for pairwise
  - Too hard for multiple
    - O(n^2) comparisons at each position

How is sequence alignment done?

- Typically multiple sequence alignment is desired
- Progressive alignment
  - Performs repeated, progressively larger alignments between pairs of sets of sequences
  - Both of the approaches I looked at use it

How is sequence alignment done?

- If you're going to align the sequences iteratively, How do you pair up the sequences?
  - A common heuristic
    - Exploit the fact that similar sequences are usually evolutionarily related
    - Pair the sequences for alignment in order of decreasing evolutionary relationship
    - Works well for similar sequences; poorly for divergent

Problems with progressive alignment

- If an early alignment is later determined to be partially incorrect, it cannot be fixed
  - Thus the correctness of the order of the pairwise alignments is essential
  - Described as the “local minimum” or “greedy” problem
  - T-Coffee attempts to address this problem

Problems with progressive alignment

- Difficult to choose the right alignment parameters
  - Parameters
    - Match fitness table
    - Gap penalties
      - Usually one penalty for opening a gap and another for extending
    - Getting these right is critical for divergent sequences
  - ClustalW's improvement is to adjust these parameters dynamically

ClustalW Algorithm

1. Align all pairs of sequences to produce a triangular matrix of distances
2. Calculate a guide tree from the distance matrix
3. Progressively align sequences in the order suggested by the guide tree

ClustalW Algorithm

- Step 1: All pairs of sequences are aligned to produce a triangular matrix of distances
  - Use dynamic programming with customizable parameters to align
  - Distance is computed as the percentage of non-identity residues between the pair of sequences
    - Gaps are excluded

ClustalW Algorithm

- Step 2: A guide tree is calculated from the distance matrix
  - Uses neighbor-joining algorithm to produce the tree
    - See Figure 1 on page 4675 of ClustalW paper
    - A type of bottom-up clustering
    - Starts with a star topology
    - Joins edges of “closest” nodes in order to minimize total branch length
    - Inserts a node as the parent of the joined nodes
    - Continues until the root has only 2 children
    - http://www.icp.ucl.ac.be/~opperd/private/neighbor.html

ClustalW Algorithm

- Step 2: A guide tree is calculated from the distance matrix
  - The tree produces a weight for each sequence that represents its similarity to all other sequences
    - A smaller number indicates greater similarity

ClustalW Algorithm

- Step 3: Sequences are progressively aligned in the order suggested by the guide tree
  - Perform a pairwise alignment at a pair of connected leaf nodes
  - At an internal node, perform a pairwise alignment between the alignment or sequence from each branch
    - See Figure 2 on page 4676 of ClustalW paper
    - E.g., if branch A had an alignment of 2 sequences and branch B had an alignment of 4 sequences then the value of each position would be the average of 8 comparisons

ClustalW Algorithm

- Step 3: Sequences are progressively aligned in the order suggested by the guide tree
  - Position values are weighted based on the values from the guide tree

ClustalW Parameter Adjustments

- In protein alignments, gaps do not occur with equal probability at all positions
  - Gaps are more likely to occur “between the major secondary structural elements of alpha-helices and beta-strands than within”
  - Thus, ClustalW imposes a location-adjusted gap opening penalty

ClustalW Parameter Adjustments

- In protein alignments, short stretches (5+) of hydrophilic residues usually indicate a loop or random coil
  - ClustalW imposes a reduced gap opening penalty in these cases
- Gaps are usually at least 8 residues long
  - Gap opening penalty increased for gap < 8 long

ClustalW Parameter Adjustments

- Some element match fitness tables favor exact matches and very conservative substitution
- Other matrices are more lenient
  - Give less importance to exact matches
  - Better for greater evolutionary distances
- ClustalW selects a fitness table based on estimated sequence divergence

ClustalW Results

- In cases of relatively similar sequences, they claim to produce alignments that are “difficult to improve by eye”
  - Similar means at least 35% identity
- Claim to have nearly correctly aligned 60 divergent (as low as 12% identity) sequences

T-Coffee Goals

- Make better decisions early in the progressive phase
- Combine the strengths of local alignment and global alignment
  - ClustalW produces a global alignment

T-Coffee Algorithm

1. Produce a primary library of alignments
2. Combine related data from all alignments
3. Perform progressive alignment

T-Coffee Algorithm

- Produce a primary library of alignments
  - Library consists of one or more scored alignments for each pair of sequences
  - Score is calculated as the percent identity between the alignments

T-Coffee Algorithm

- Combine related data from all alignments
  - Data are combined at a residue pair level
    - Add together the weights of any alignments which support that a residue pair is correctly aligned
    - Incorporates data from all alignments into the scores of every alignment
  - The result is called the extended library

T-Coffee Algorithm

- Perform progressive alignment
  - Doesn't use a normal match fitness table and a gap penalty
  - Uses the weights from the extended library as a match fitness table and uses zero gap penalty
  - Use of the extended library means that early alignments are informed by all alignments