Pairwise sequence alignment

Tandy Warnow

December 27, 2016

Pairwise alignment

Tandy Warnow

True Pairwise Alignment

- Suppose X and Y are two sequences, and X evolves into sequence Y via insertions, deletions, and substitutions.
- The true pairwise alignment of X and Y represents this true history.
- Examples:
  - AAT evolves into ACCAT by the insertion of CC
  - ATGA evolves into ATTAG by changing G to T, and then adding G
  - CTAA evolves into CTTAA by inserting a T.

Questions:

1. What are the pairwise alignments?
2. How can we guess at these evolutionary histories (and so pairwise alignment)?

Edit distance

Suppose each event (insertion, deletion, and substitution) costs 1. Can we compute the minimum cost edit transformation between two sequences?

Computing the edit distance

Input: sequences a and b of lengths m and n, respectively.
Output: minimum number of indels and substitutions needed to transform a into b.
A two-dimensional matrix, F[0..m,0..n] is used to hold the edit distance values:

$F(i,j) = d(a[1..i], b[1..j])$ (Definition of what we want)

$F(0,0) = 0$

$F(i,0) = i, i = 1..m$

$F(0,j) = j, j = 1..n$

For $i,j \geq 1$, $F[i,j] = min\{$

$F[i-1,j-1]+$ if a[i]=b[j] then 0 else 1,
$F[i-1,j]+1$,
$F[i,j-1]+1$
$\}$

Needleman-Wunsch minimum edit distance

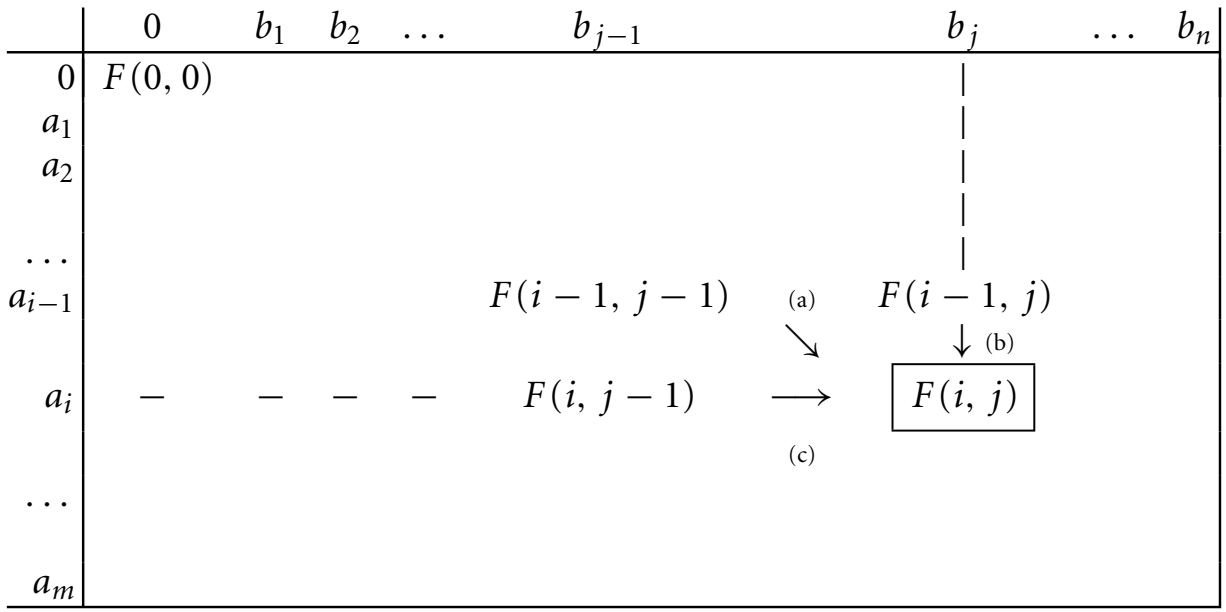

Figure 9.3 (Figure 2.4 in Huson et al. (2010)) The Needleman-Wunch dynamic programming ap-

Dynamic programming algorithm for minimum edit distance

Compute the matrix from the bottom up!
Running time is $O(KL)$ where the first sequence has length K and the second sequence has length L.
Example: $ACAT$ and $CCGT$

Finding the actual transformation - backtracing

To find the actual transformation, use backtracing.
Example: *ACAT* and *CCGT*

Extensions

Minimum cost approaches:

- How would you modify this algorithm if indels cost C and substitutions cost C'?
- How would you modify the edit distance algorithm if a indel of length p has cost $C + C'p$?

Maximize similarity approaches:

- How would you modify the algorithm to maximize score, so matches have value 1 and mismatches and indels each have negative value -1 (i.e., they cost)?
- How would you modify the algorithm to maximize score, where matches and mismatches have scores (possibly negative) that depend on the pair of letters, and indels all have negative scores?
- How would you modify the algorithm if you want a *local alignment* (maximize cut off some prefix and suffix)

More generally,

- How would you align two alignments?

Extending to multiple alignment

Now that we have a way of defining the "cost" of a pairwise alignment, we can extend to a set of three or more sequences in at least two ways:

- Sum-of-pairs score (find MSA to minimize the sum of costs of all induced pairwise scores)
- Treelength (find tree and sequences at all nodes of the tree to minimize the sum of costs on the edges of the tree)

Both problems are NP-hard.

Treelength

Finding the sequences at internal nodes of a fixed tree to find the minimum cost is itself NP-hard.
Methods for Treelength include POY and BeeTLe.
Very computationally intensive (worse than MP) ...and also controversial!

Progressive Alignment

- Given sequences S, find rooted tree (somehow)
- Align sequences from the bottom up. Note this requires the ability to align two alignments.
- Return the alignment defined at the root.