Sequence Matching: The Needleman Wunsch Algorithm.

General algorithm for sequence comparison

Fundamental principle - to calculate the alignment score S(i, j) you only need to enumerate and score all ways in which one aligned pair can be added to a shorter alignment to produce an alignment of the first *i* residues of seq1 and the first *j* residues of seq2.

All possible pairs are represented by a two-dimensional array, and all possible comparisons are represented by pathways through this array.

Global alignments - i.e., every residue of the two sequences has to participate - therefore it will not detect motif or active site homology alone.

Sequence Matching: The Needleman Wunsch Algorithm.

Three Main Steps:

1. Assign similarity scores:

A numerical value (score) is assigned to every cell in the array depending on similarity/dissimilarity. Similarity scores may be simple, or related to chemical similarities or frequency of observed substitutions.

2. Score pathways through array:

For each cell want to know the maximum possible score for an alignment ending at that point. Cumulative score by adding in a path through the matrix. Searches subrow and subcolumn for the highest score. Gap penalty independent of the length of the gap. The best match is the pathway with the highest score.

Maximum match = largest number of amino acids of one protein that can be matched with those of another protein while allowing for all possible deletions.

3. Construct an alignment.

Sequence Matching: The Needleman Wunsch Algorithm.

Similarity values:

A numerical value is assigned to every cell in the array depending on the similarity/dissimilarity of the two residues.

These may be simple scores of scores based on their structure/function or frequencies of observed substitutions.

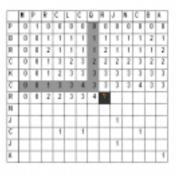
An example with a simple score function: match = 1, no match = 0 is illustrated here:

		P	a.	C	L	C	ų.	R	J.	Ν	C	в	A
F	T	1											
Ð	Г											1	Г
R	t		1					1					Γ
Ċ	T			1		1					1		
ĸ	t												Γ
¢	T	\square		1		1					Π		Г
RN	Г							1					Г
	Г									1			Г
J	T								1				
Ċ	Γ			1		1					1		
J	Г								1				
A.	Г												

Sequence Matching: The Needleman Wunsch Algorithm.

Score pathways through the array:

- For each cell, we want to know the maximum possible score for an alignment ending at that point.
- Searches subrow and subcolumn as shown for the nighest score.
- Add this to the score of the current cell.
- · Proceeds rod by row through the array.



Sequence Matching: The Needleman Wunsch Algorithm.

Constructing an alignment:

- The alignment score is cumulative by adding along a path through the array.
- The best alignment is the highest score, i.e., the maximum match.
- The maximum match will always be somewhere in the outer row or column.
- The alignment is constructed by working backwards from the maximim match.



HP-RCLOOR-JHCBA | || | | | | | | -PERCEC-RHJ-CJA

Sequence Matching: The Needleman Wunsch Algorithm.

Just as in the Smith-Waterman algorithm, we can augment this algorithm with similarity tables and gap penalties.

In our example, we had used a trivial similarity measure: match = 1, no match = 0.

We had also used gap penalty = 0.

In this case,

$$\begin{array}{ll} Hij = \max\{ & H_{i-1,j-1} + s(a_i, \ b_j), \\ & \max\{H_{i+k,j-1} - W_k + s(a_i, b_j)\}, \\ & \max\{H_{i-1,j-r} - W_r + s(a_i, b_j)\} \\ \} \end{array}$$

Tools for Alignment and Matching:

We have seen the Smith-Waterman and the Needleman-Wunsch algorithms for matching sequences.

Using these algorithms, given two sequences of length *m* and *n*, we must compute a table of size *n* x *m*. In each algorithm, each entry in this table takes a constant amount of computation. If this constant is t_{c} then the time taken for one match is

$T = n x m x t_c$

When n and m become large (even for simple proteins, n and m can be tens of thousands), these algorithms can take a long time to execute.

Furthermore, we are typically matching a reference sequence against a large database of sequences. Therefore, we might need to perform a large number of such matches. In this case, approximations to these algorithms might be necessary.