## Dynamic Programming: The algorithm to find the optimal sequence alignment

As we have seen in the previous lecture to score an alignment we need a substitution table, e.g. BLOSUM62, and a gap penalty, e.g. gap penalty g= -8.

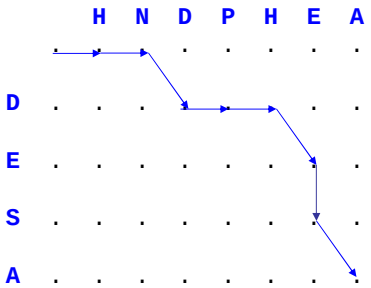
How do we find the alignment with the highest score?

This problem was solved by Needleman and Wunsch using a method that has become known as “Dynamic Programming”.

Needleman and Wunsch, 1970, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol., **48**, 443-453.

## Global Alignment: The Needleman-Wunsch Algorithm

Consider two sequences **HNDPHEA** and **DESA** we can create the following matrix and path:




Create an (m+1)×(n+1) matrix where n=length of 1st sequence (along top) and m=length of 2nd sequence (along left column).

Exercise: Can you relate the path through the matrix above to an alignment?

**HNDPHE-A**
**--D--ESA**

**Global Alignment: The Needleman-Wunsch Algorithm**

The basic operations of match, insert a gap in the upper sequence, and insert a gap in the lower sequence are represented by a ↘, ↓ and →, respectively in the matrix. Any alignment can be represented by a path through this matrix using these elemental moves. Each move will be accompanied by the corresponding score, a gap penalty score for ↓ and →, and a substitution score for ↘.

Note sliding one sequence relative to other is equivalent to inserting gaps at beginning of the sequence which means moving along the top row or left most column.

Object is to find the path with the maximum score.

Solved by building up the optimal alignment between two sequences from the optimal alignment of smaller subsequences.

**Global Alignment: The Needleman-Wunsch Algorithm**

The basic operations of match, insert a gap in the upper sequence, and insert a gap in the lower sequence are represented by a ↘, ↓ and →, respectively in the matrix. Any alignment can be represented by a path through this matrix using these elemental moves. Each move will be accompanied by the corresponding score, a gap penalty score for ↓ and →, and a substitution score for ↘.

Note sliding one sequence relative to other is equivalent to inserting gaps at beginning of the sequence which means moving along the top row or left most column.

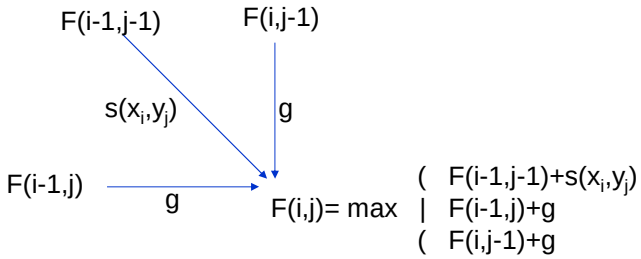
Object is to find the path with the maximum score.

Solved by building up the optimal alignment between two sequences from the optimal alignment of smaller subsequences.

## Global Alignment: The Needleman-Wunsch Algorithm

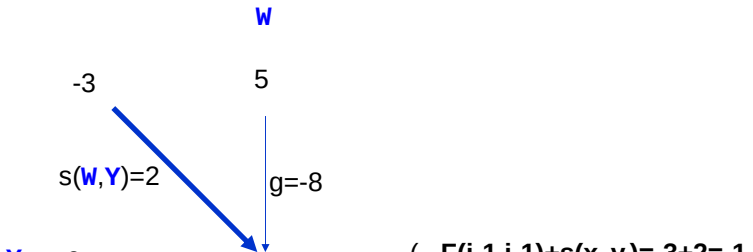
Consider two sequences X and Y, where $x_i$ is the ith residue of sequence X (the upper sequence) and $y_j$ is the jth residue of sequence Y (the lower sequence). Consider the matrix denoted, F, for which the value F(i,j) is the score of the best alignment of the subsequences up to $x_i$ and $y_j$.

If F(i-1,j-1), F(i-1,j) and F(i,j-1) are known then F(i,j) can be calculated as:




## Global Alignment: The Needleman-Wunsch Algorithm

BLOSUM62 and g= -8.

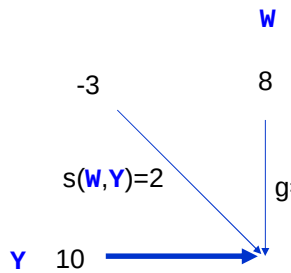



F(i,j)=-1 from diagonal route so **W** and **Y** are matched.

```
...W...
...Y...
```

**Global Alignment: The Needleman-Wunsch Algorithm**

BLOSUM62 and g= -8.



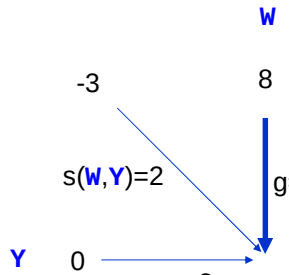

F(i,j)= max ( F(i-1,j-1)+s($x_i$,$y_j$)=-3+2=-1
| **F(i-1,j)+g=10-8=2**
( F(i,j-1)+g=8-8=0

F(i,j)=2 from horizontal route so **W** and **-** are matched.

```
...W...
...-...
```

**Global Alignment: The Needleman-Wunsch Algorithm**

BLOSUM62 and g= -8.




F(i,j)= max ( F(i-1,j-1)+s($x_i$,$y_j$)=-3+2=-1
| F(i-1,j)+g=0-8=-8
( **F(i,j-1)+g=8-8=0**

F(i,j)=0 from vertical route so **-** and **Y** are matched.

```
...-...
...Y...
```

**Global Alignment: The Needleman-Wunsch Algorithm**

So each cell in the matrix has the maximum value from the 3 surrounding cells.

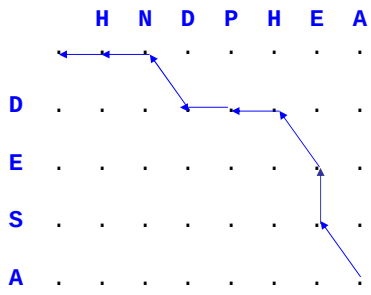
To complete the algorithm we need an initial values. They are F(0,0)=0.

F(i,0)=ig
F(0,j)=jg

The value in final cell is the best score of the global alignment. The alignment itself can be found by tracing back from the final cell. This is known as *traceback*. It involves moving back from the current cell at (i,j) to the previous neighbouring cell (i-1,j), (i,j-1) or (i-1,j-1) from which the score at F(i,j) derived.

**Global Alignment: The Needleman-Wunsch Algorithm**

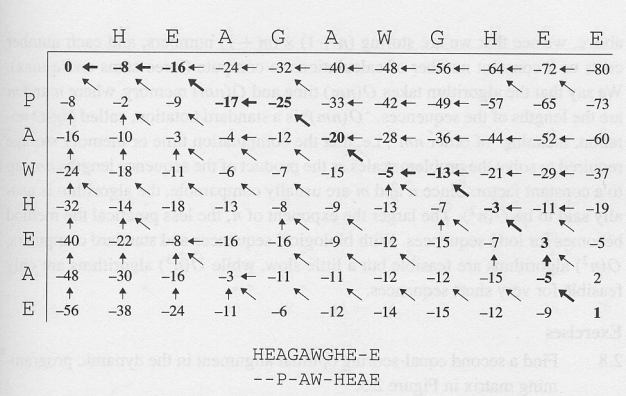
**Traceback**




**HNDPHE-A**
**--D--ESA**

## Global Alignment: The Needleman-Wunsch Algorithm

The figure below shows the “dynamic programming matrix” for the alignment of the sequences HEAGAWGHEE and PAWHEAE using the BLOSUM50 substitution matrix and a gap penalty g=-8.



| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | **0** | **–8** | **–16** | –24 | –32 | –40 | –48 | –56 | –64 | –72 | –80 |
| P | –8 | –2 | –9 | **–17** | **–25** | –33 | –42 | –49 | –57 | –65 | –73 |
| A | –16 | –10 | –3 | –4 | –12 | **–20** | –28 | –36 | –44 | –52 | –60 |
| W | –24 | –18 | –11 | –6 | –7 | –15 | **–5** | **–13** | –21 | –29 | –37 |
| H | –32 | –14 | –18 | –13 | –8 | –9 | –13 | –7 | **–3** | –11 | –19 |
| E | –40 | –22 | –8 | –16 | –16 | –9 | –12 | –15 | –7 | **3** | –5 |
| A | –48 | –30 | –16 | –3 | –11 | –11 | –12 | –12 | –15 | **–5** | 2 |
| E | –56 | –38 | –24 | –11 | –6 | –12 | –14 | –15 | –12 | –9 | **1** |

```
HEAGAWGHE-E
--P-AW-HEAE
```

From: Biological Sequence Analysis, Durbin, Eddy, Krogh, Mitchison, Cambridge.