# Biology 644

**Old Title:** Bioinformatics for Molecular Biologists

**Potential New Title:** Integrated Bioinformatics Using R for Both Wet and Dry Scientists

# Optimal Pairwise Alignment Methods

- Find the "best alignment" between 2 sequences with lengths n and m, respectively
- "Best alignment" is very dependent upon the substitution matrix and gap penalties
- The Global Alignment Problem tries to find the path between vertices (1,1) and (n,m) in the edit/alignment graph with the best alignment score.
  - The Needleman–Wunsch algorithm
- The Local Alignment Problem tries to find the subpath (among all possible subpaths) between any arbitrary vertices (i,j) and (i', j') in the edit graph with the best alignments score .
  - The Smith–Waterman algorithm
- The Overlap Alignment Problem tries to find the path between vertices (1,j) or (i,1) and (n,j') or (i',m) in the edit/alignment graph with the best alignment score. (arbitrary i,j,i',j')
  - The Overlap algorithm (combines Needleman–Wunsch and Smith–Waterman algorithm)
- These methods use dynamic programming with a recursive "divide-and conquer" strategy to fine the optimal path in the alignment matrix

# Dynamic Programming

- "Divide-and Conquer" strategy
  - Breaks the problem down into smaller sub-problems
    1. Solve the smaller sub-problems optimally
    2. Use the sub-problem solutions to construct the optimal solution to the original problem
- Can be applied to problems that consist of overlapping sub-problems
  - Traveling salesman problem
  - Pairwise Sequence Alignments
    - Global Alignment (Needleman-Wunsch)
    - Local Alignment (Smith-Waterman)

## Global Alignment: Needleman-Wunsch

- Global alignment forces the alignment of both entire sequences
- Guaranteed to find the optimal global alignment(s) of two sequences
  - Even if the two sequences are completely unrelated
  - Because of this, often a threshold score is used to accept a global alignment as significant
- "Optimal" means best scoring according to the substitution matrix and gap penalties you choose
- Used primarily when aligning sequences from two related homologs
- Uses a Dynamic Programming Matrix to find the optimal global alignment score and a “traceback” to find the positions/nucleotides/residues in the alignment path
- Slower than heuristic methods

# Needleman-Wunsch DP Matrix

- Breaks the alignment matrix problem down into smaller sub-problems

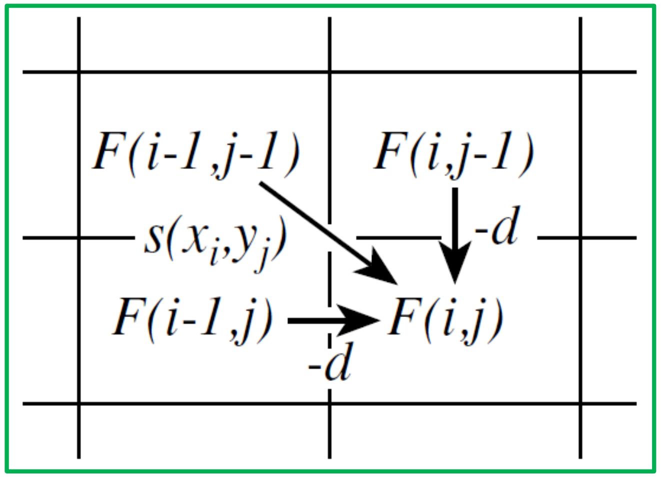
  1. Solve each smaller sub-matrix optimally

  2. Use the sub-matrix solutions to construct the optimal matrix that can be used to find the optimal alignment

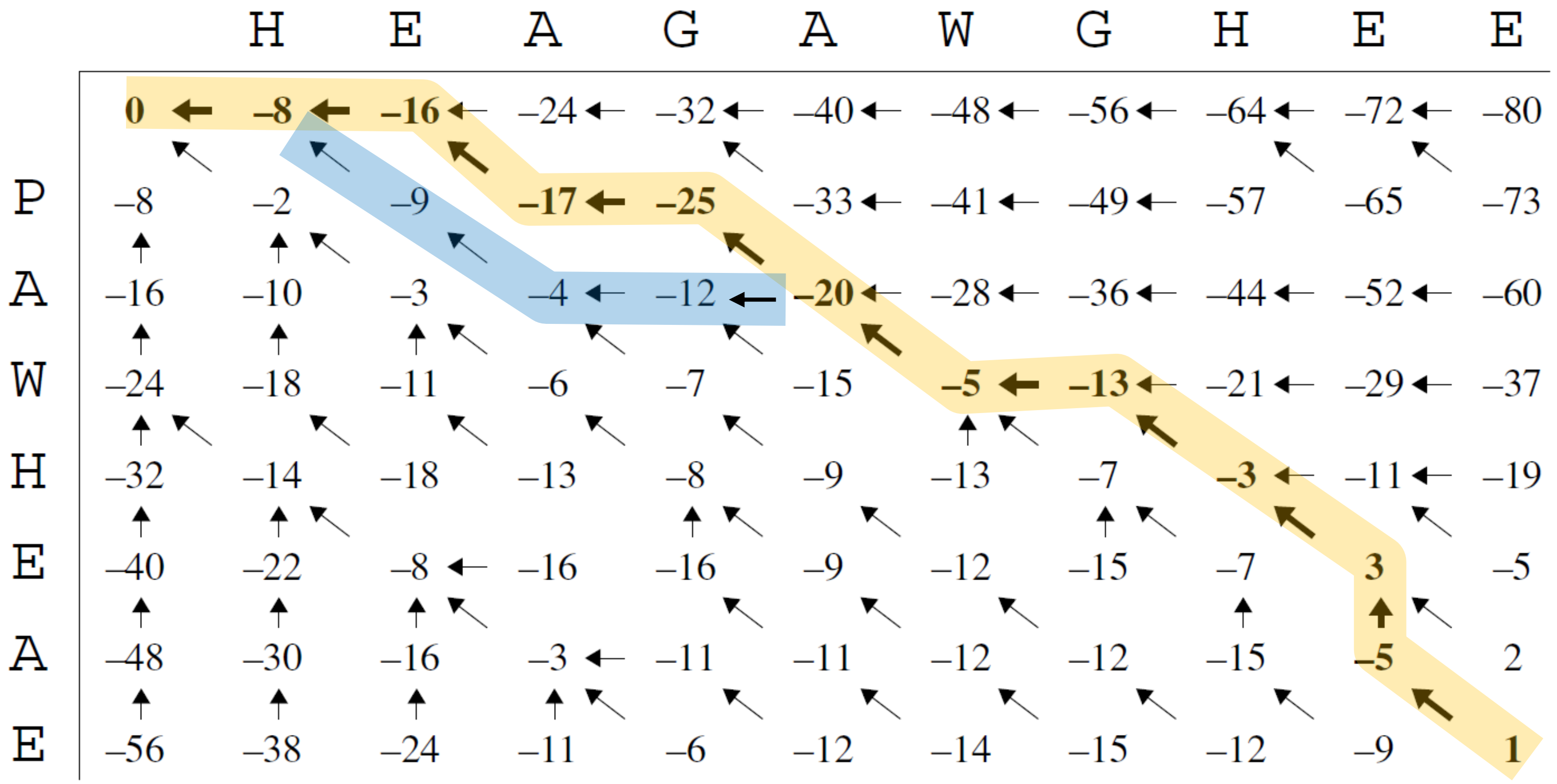
```
HEAGAW      HEAG-AW     HEAGA-W
HEAGYW      HEAGY-W     HEAG-YW
```

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i,y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

*F(column, row)*



# Global Dynamic Programming Matrix

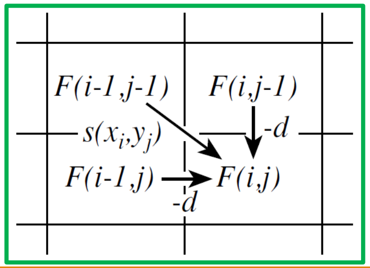


| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | –8 | –16 | –24 | –32 | –40 | –48 | –56 | –64 | –72 | –80 |
| P | –8 | –2 | –9 | –17 | –25 | –33 | –41 | –49 | –57 | –65 | –73 |
| A | –16 | –10 | –3 | –4 | –12 | –20 | –28 | –36 | –44 | –52 | –60 |
| 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 |

$$F(i,j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j), \\ F(i-1, j) - d, \\ F(i, j-1) - d. \end{cases}$$

## 2 Optimal Alignments

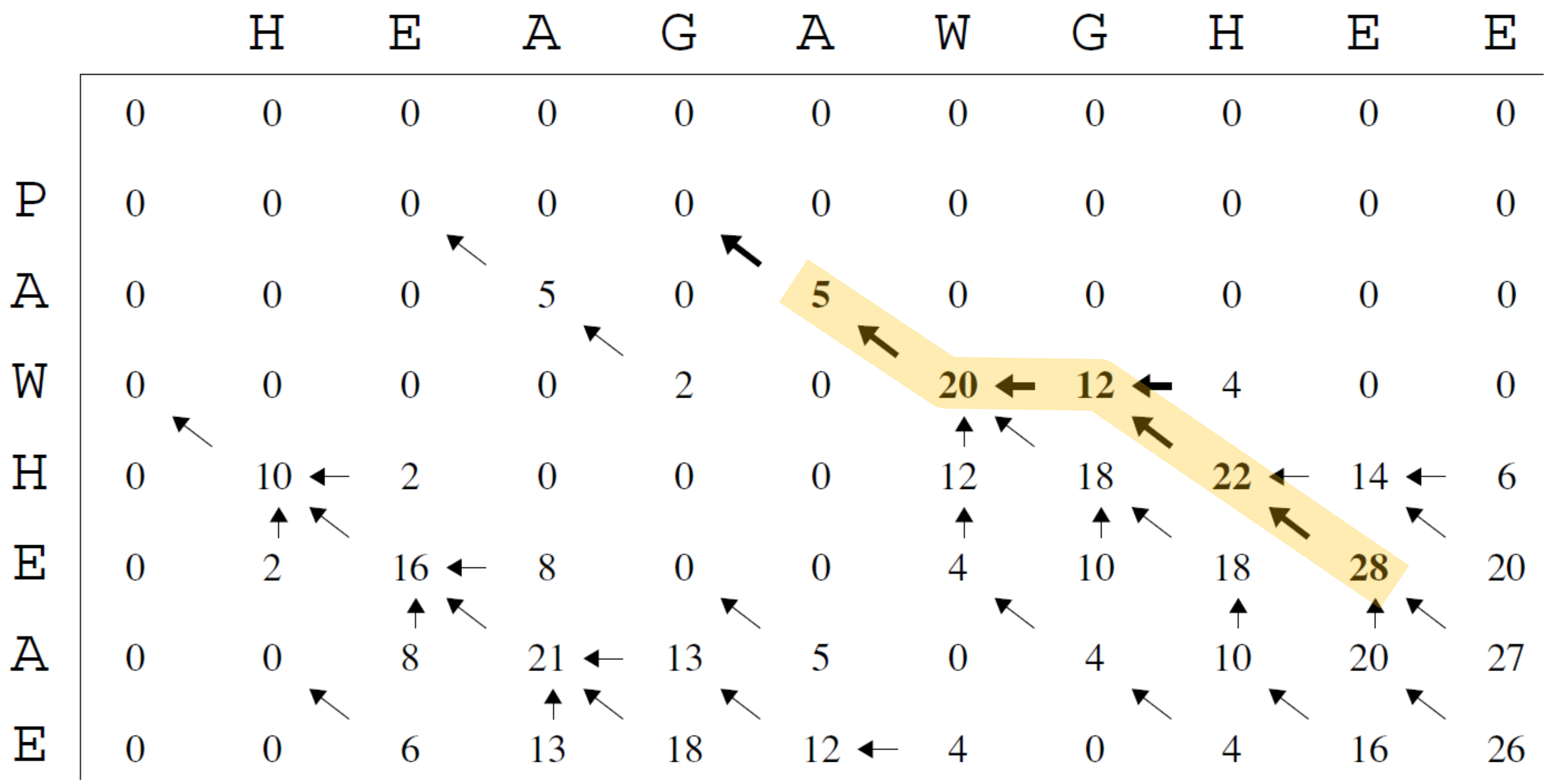
```
HEAGAWGHE-E      HEAGAWGHE-E
--P-AW-HEAE      -PA--W-HEAE
```



## Local Alignment: Smith-Waterman

- Similar to Needleman-Wunsch but with some modifications:
  - Whenever the score of the optimal sub-alignment is less than zero, it is terminated (the matrix element is set to 0)
  - The traceback starts from the highest-scoring element instead of at the lower right corner since you can to start a new alignment anywhere
  - The traceback is stopped as soon as a zero is encountered
- Guaranteed to find the optimal local alignment(s) of two sequences
  - Even if the two sequences are completely unrelated
  - Because of this, often a threshold is used to accept a local alignment as significant
    - P-Value (Score) or E-Value (Score)

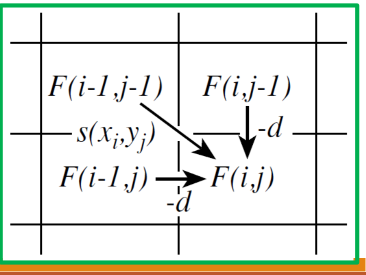
# Local Dynamic Programming Matrix



| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 5 | 0 | **5** | 0 | 0 | 0 | 0 | 0 |
| W | 0 | 0 | 0 | 0 | 2 | 0 | **20** | **12** | 4 | 0 | 0 |
| H | 0 | 10 | 2 | 0 | 0 | 0 | 12 | 18 | **22** | 14 | 6 |
| E | 0 | 2 | 16 | 8 | 0 | 0 | 4 | 10 | 18 | **28** | 20 |
| A | 0 | 0 | 8 | 21 | 13 | 5 | 0 | 4 | 10 | 20 | 27 |
| E | 0 | 0 | 6 | 13 | 18 | 12 | 4 | 0 | 4 | 16 | 26 |

$$F(i,j) = \max \begin{cases} 0, \\ F(i-1,j-1) + s(x_i,y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$

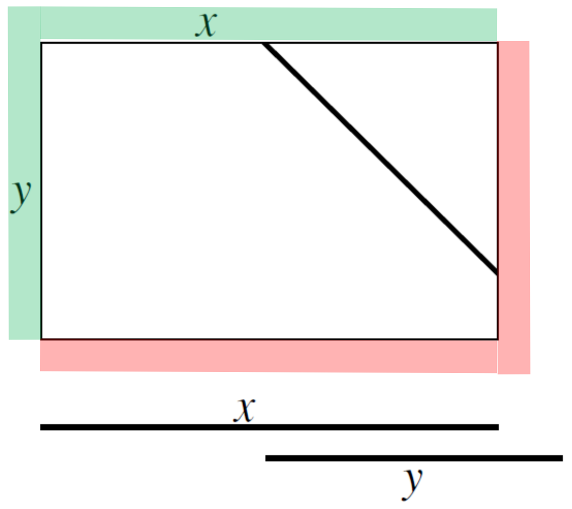
```
AWGHE
AW-HE
```

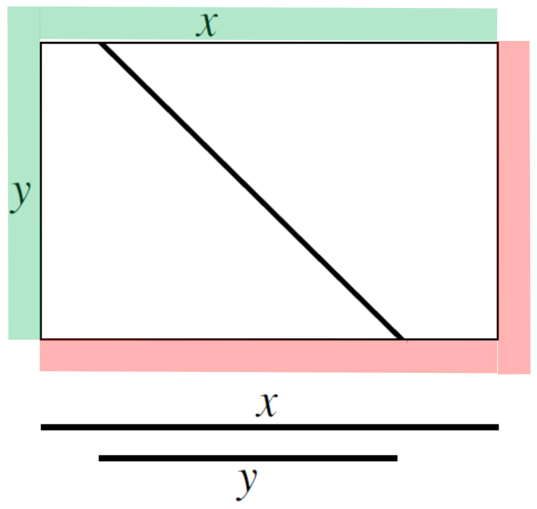


# Overlap Alignment: Modified Needleman-Wunsch

- Similar to Needleman-Wunsch but with some modifications:
    - The match must start on the top or left border of the matrix, and finish on the right or bottom border.
    - The initialization equations are therefore that $F(i, 0) = 0$ for $i = 1,...,n$; and $F(0, j) = 0$ for $j = 1,...,m$
    - The traceback starts from the maximum point on the right or bottom edge and continues until the top or left edge is reached
- Guaranteed to find the optimal overlap alignment(s) of two sequences
    - Even if the two sequences are completely unrelated
    - Because of this, just like in the case of global and local aligning, often a threshold is used to accept an overlap alignment as significant
        - P-Value (Score) or E-Value (Score)

# Overlap Dynamic Programming Matrix

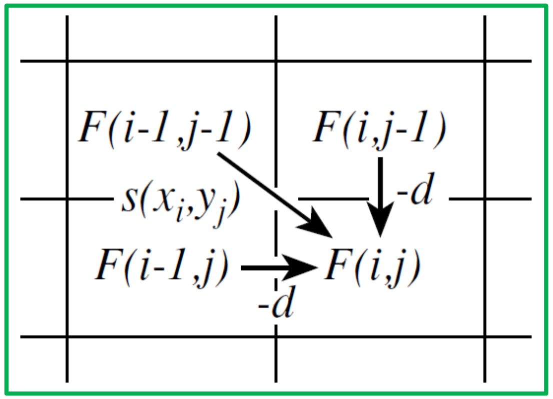




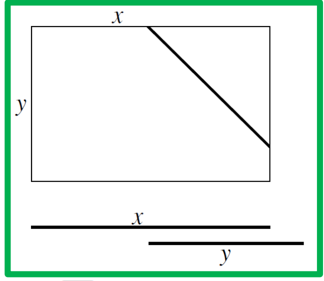
🟩 Start Edge

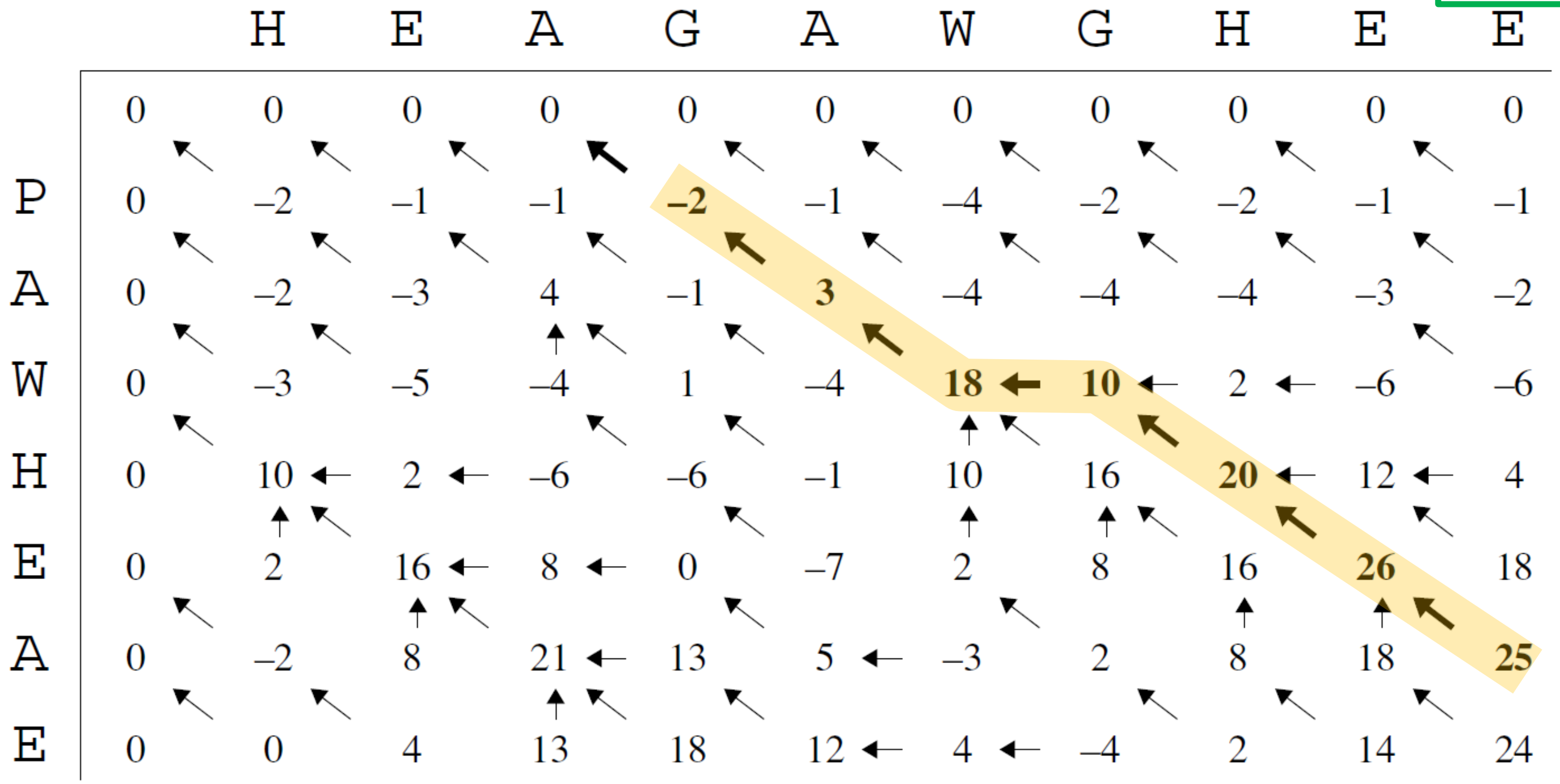
🟥 Finish Edge

$$F(i,j) = \max \begin{cases} F(i-1,j-1)+s(x_i,y_j), \\ F(i-1,j)-d, \\ F(i,j-1)-d. \end{cases}$$



# Overlap Dynamic Programming Matrix

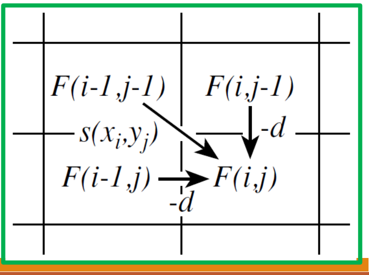




| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | –2 | –1 | –1 | **–2** | –1 | –4 | –2 | –2 | –1 | –1 |
| A | 0 | –2 | –3 | 4 | –1 | **3** | –4 | –4 | –4 | –3 | –2 |
| W | 0 | –3 | –5 | –4 | 1 | –4 | **18** | **10** | 2 | –6 | –6 |
| H | 0 | 10 | 2 | –6 | –6 | –1 | 10 | 16 | **20** | 12 | 4 |
| E | 0 | 2 | 16 | 8 | 0 | –7 | 2 | 8 | 16 | **26** | 18 |
| A | 0 | –2 | 8 | 21 | 13 | 5 | –3 | 2 | 8 | 18 | **25** |
| E | 0 | 0 | 4 | 13 | 18 | 12 | 4 | –4 | 2 | 14 | 24 |

```
GAWGHEE
PAW-HEA
```

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + s(x_i,y_j), \\ F(i-1,j) - d, \\ F(i,j-1) - d. \end{cases}$$



# BLAST: Basic Local Alignment Search Tool

- Heuristic method: Guaranteed to give a good alignment fast (but not necessarily the optimal one)
- Used to find statistically significant local alignments between a query and a large database of known genes from many model species
- Many versions (protein-protein, nucleotide-nucleotide,nucleotide-protein …)
- Widely used and very useful – it's good to know the algorithm
- Pairwise alignment by dynamic programming requires computing an $L_1$ x $L_2$ matrix, where $L_1$ and $L_2$ are the sequence lengths and takes a long time.
  - The speedup used by BLAST (and other heuristic algorithms):
    - reduce the size of this matrix by using fast methods to find "diagonals", also called "gapless high-scoring segment pairs (HSPs)"
    - and then extend and join them together to find good local alignments

# BLAST Overview

## 6 Steps

1. Filtering of low complexity regions from the query sequence (optional)
2. Compile list of relevant words in the query sequence
3. Scan database sequences to find hits to the words in the query sequence
4. Extend hits to High-scoring Segment Pairs (HSPs)
5. Calculate E-values for significant hits
6. Sort and Smith-Waterman Align the best scoring HSPs

# BLAST Step 1: Filtering (Optional)

- Filtering of low complexity regions from the query sequence
- Some sequences contain low complexity regions
    - Map to too many regions in too many genomes
    - Give rise to many random hits
    - Sometimes are repetitive elements
- Filter out by replacing with Xs

# BLAST Step 2: Compiling Similar Word List

- Typically, word length L=3 (protein) or L=11 (nucleotides)
- Find all words of length L in query sequence looking at each start position
- For each position in query sequence,
  - Compare to all possible length L words to find similar words
  - remove dissimilar words below threshold T (usually 11)
  - Limited to ~50 similar words per start position

```
APLSADQASLVKSTWAQVRNSEVEILAAVFTAYPDIQARF...
APL
 PLS
  LSA...  ⟶
```

Word list for position 1 `APL`: **`API`**`,APC,APS,APT,APE`…
Remove words *w* with score(`APL`,w) < T

---

```
GAGTTCCTGGCCATGCTCAATGCTCGATCGGCCTATAG...
GAGTTCCTGGC
 AGTTCCTGGCC
  GTTCCTGGCCA...  ⟶
```

# BLAST Step 3: Scan Database of Words

- Store words for each position in efficient search tree (often a suffix tree)
- Quickly Scan each sequence in a huge database
  - Often the “database” resides in memory in efficient search trees on high-end servers with Terabytes of RAM
- Record exact word hits

Found a hit!

```
GTGGAGGACAACTCCTGGCCATGCTCACGGAGCCAAGTGGAGA
            TCCTGGCCATG
```

# BLAST Step 4: Join and Extend Hits

## BLAST2

- Find hits on same diagonal with distance ≤ A
  - Connect them creating an ungapped alignment

- Extend hits using gaps, matches and mismatches
  - Extension continues until the score falls below the maximum score yet attained minus some value X

- Joined and extended Hits are a called High-scoring Segment Pairs (HSPs)

## Original BLAST

- Extend all single word hits (higher T needed)

Extend hit

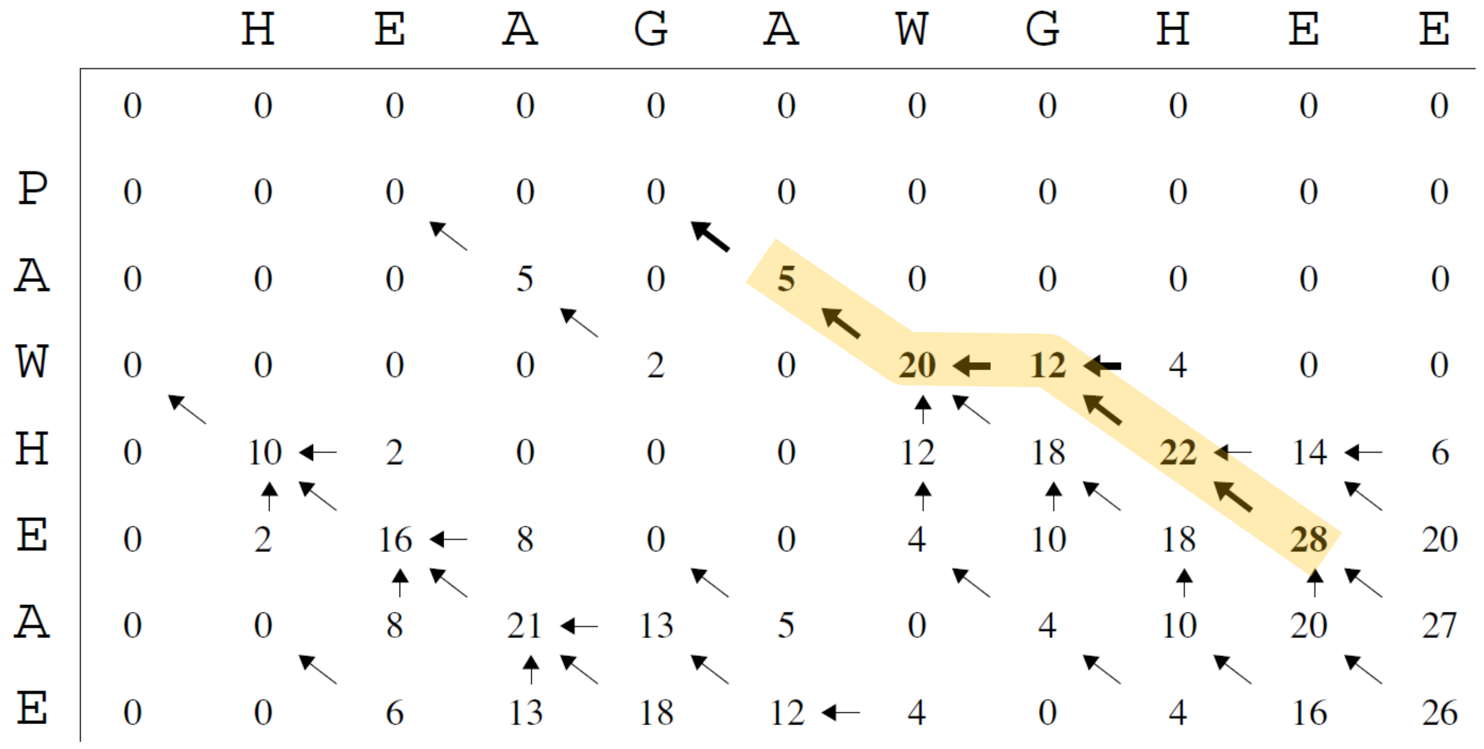
```
Query:          GAGTTCCTGGCCATGCTCA
DB Hit: GTGGAGGACAACTCCTGGCCATGCTCACGGAGCCAAGTGGAGA
Hit:        <------- TCCTGGCCATG -------->
```

# BLAST Step 5: Calculate E-scores and P-scores

- Compile list of HSPs with scores > T
- Let x be the score of a HSP:
    - P-value (Probability): probability that a database yields by pure chance at least one alignment with same or higher score: $P = \text{Prob}(\text{score} \geq x)$
    - E-value (Expected value): number of unrelated database sequences expected to yield same or higher score by pure chance: $E \approx -\ln(1 - P)$
- The E-value describes the number of HSPs one can expect to see just by chance when searching a database of a particular size.
    - Decreases exponentially with increasing score S between two sequences
    - Essentially describes the random background noise that exists for matches between sequences
    - A convenient way to create a significance threshold for reporting results
        - E-value threshold: E-value < 1e-10 for nucleotide searches; < 1e-4 for protein searches
- When E-value < 0.01, P-values and E-value are nearly identical.

# BLAST Step 6: Optimally Align Best HSPs

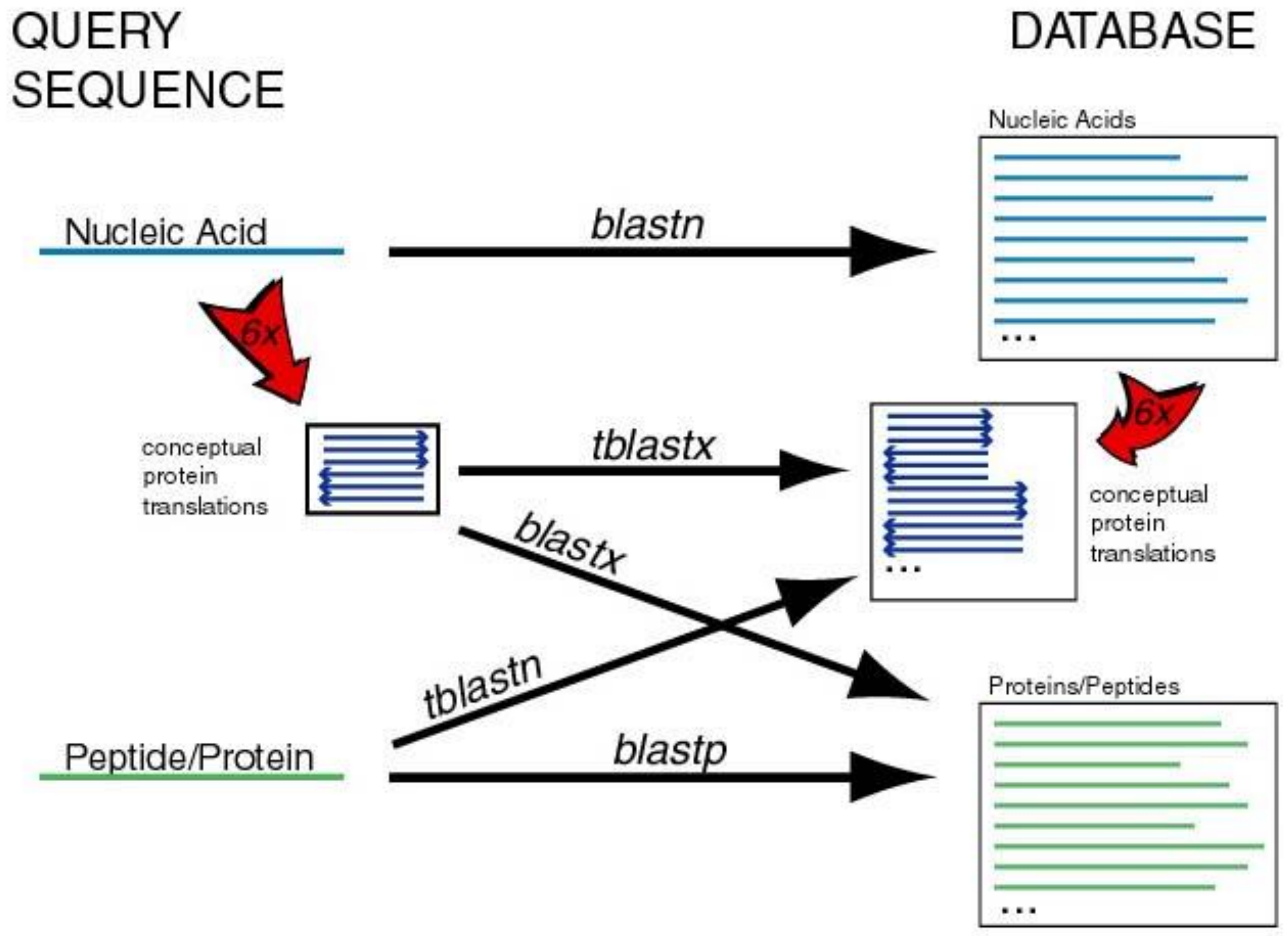
- **Sort the HSPs by E-value**
- **Smith-Waterman Alignment the top HSPs**



| | | H | E | A | G | A | W | G | H | E | E |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| P | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 0 | 0 | 5 | 0 | **5** | 0 | 0 | 0 | 0 | 0 |
| W | 0 | 0 | 0 | 0 | 2 | 0 | **20** | **12** | 4 | 0 | 0 |
| H | 0 | 10 | 2 | 0 | 0 | 0 | 12 | 18 | **22** | 14 | 6 |
| E | 0 | 2 | 16 | 8 | 0 | 0 | 4 | 10 | 18 | **28** | 20 |
| A | 0 | 0 | 8 | 21 | 13 | 5 | 0 | 4 | 10 | 20 | 27 |
| E | 0 | 0 | 6 | 13 | 18 | 12 | 4 | 0 | 4 | 16 | 26 |

```
AWGHE
AW-HE
```

# BLAST: Basic Local Alignment Search Tool



## Recommended Substitution Matrices for BLAST

| Query length | Amino Acid Substitution matrix | (O,E) Gap Penalties |
|---|---|---|
| < 35 | PAM-30 | (9,1) |
| 35 – 50 | PAM-70 | (10,1) |
| 50 – 85 | BLOSUM-80 | (10,1) |
| > 85 | BLOSUM-62 | (10,1) |

# Local Vs Global Alignment

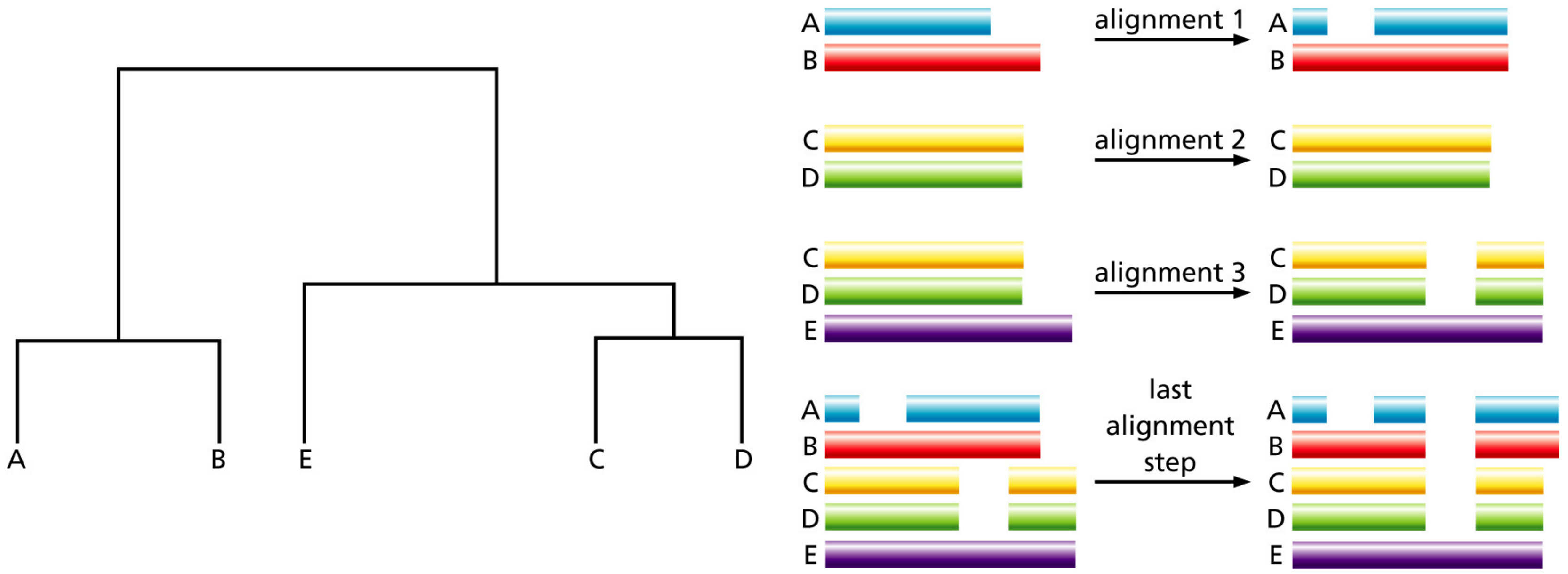
- Global alignment forces the alignment to cover both entire sequences
- Generally, local alignment is used for performing database searches
    - In most cases, you are interested in knowing if any parts of your sequence looks like any parts of any other sequences (e.g. - protein domain search)
    - Although protein domains can move around and switch positions inside homologous gene sequences, local alignment allows you to still find them
- Global alignment is good for related homologs where the protein domains have not switched positions
    - Otherwise, local alignment is better:

```
--T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || |  | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C
```

```
                  tccCAGTTATGTCAGggacacgagcatgcagagac
                     ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

## CLUSTALW, MUSCLE methods

**1. Compute alignment scores between all sequence pairs**
**2. Take highest scoring pair and create their consensus sequence**
**3. Take next highest scoring pair and create their consensus sequence**
**….**
**4. Repeat until all sequences are merged into a single consensus.**



## Alignment Caveats

- Dynamic Programming alignment programs always find the optimal alignment of two sequences
  - Even if the two sequences are completely unrelated
  - "Optimal" means best scoring according to the substitution matrix and gap penalties you use
- Heuristic methods are much faster but not guaranteed to find the best alignment
- For both types, the following underlying assumptions are generally wrong:
  - The frequency of substitution is not the same at all positions
  - The frequencies of insertions and deletions are also not the same at all positions
  - Affine gap penalties do not properly model indel events

# Public Sequence Databases

- NCBI Gene Bank
  - http://ncbi.nih.gov
  - contains many sub-databases
- Protein Data Bank
  - http://www.rcsb.org
  - contains protein structures
- SwissProt
  - http://www.expasy.org/sprot/
  - contains annotated protein sequences
- Prosite
  - http://kr.expasy.org/prosite
  - contains motifs of protein active sites